Infectivity Model

New Mexico Supercomputing Challenge

Final Report

April 4, 2018

Team Number: ATC-3

School Name: Academy for Technology and the Classics

Team Members:

Alex Baten

Ben Thorp

Teddy Gonzales

Teacher:

Jennifer Hooten

Project Mentor:

John Thorp

Executive Summary

Our project is a model of the spread of the Influenza Thread A between several hypothetical neighborhoods. By studying how individual conditions can affect the spread of a disease through several different communities, which were roughly organized using a map found on the City of Albuquerque website[1], we hope to suggest factors that will minimise the spread of Influenza Thread A. We can use our data to show if individual decisions produce a quicker spread through the community. This data can also help us understand, on a local level, how the disease spreads in different conditions. Our model considers countermeasures against disease, such as vaccines and contagion time periods. This model can also be used to compare two or more different variables and their impact on the spread of influenza.

We created a computer model in Python, to demonstrate the spread of Influenza between several communities. The model simulates the movement of agents between day and night congregation points. Each run of the simulation takes a number of parameters that were gathered using Zip Codes located in each district (Appendix A) including the following:

- The total number of humans in each district
- An estimate about the number of houses and rooms in each area
- The times when the disease starts and stops being contagious in communities with or without health care
- Percentage of vaccinated humans in the different communities

Last year we tested the model with a number of simulations that grew in complexity as the model matured. These simulations are presented below. The simulations demonstrate that

---

[1] *City Of Albuquerque-Find Your Counselor*

the model can predict, on a local level, how the disease spreads through different communities in different conditions. This year we included real-world data taken from the 2015 Community Survey taken by the Census Bureau (Appendix B). This meant that most the data would be fixed in terms of the amount of people and the age groups of those people.

Problem Statement

While discussing potential ideas for our project in our first year, we came across Google Trends. In our Computer Science class we read an article that discussed about how Google Trends could be used as a database for influenza infection rates by viewing the search results for “flu symptoms” over time. The problem is that Google Trends can only present data from the past. To address the factors that cause the Influenza to spread, we wanted a computer model that could simulate future situations and provide useful data. Expanding on the idea of simulating future situations, we decided to add variables, such as vaccines, to see how the initial conditions in a community would affect the spread of influenza. Of course our model is only a basic simulation, however, we have built on last year’s model to increase its complexity and develop a program that makes more accurate predictions by incorporating more real world data than we used last year.

This year we originally had the idea to incorporate many new parameters such as: wealth, population of a real-life city, and a formula that would help us calculate the probability of getting the disease. We originally had decided to use San Diego as the city that we would use as the model. However, as we worked on the program, we decided to use Albuquerque as the city that we would model. This is because Albuquerque is more local and applicable to where we live.

Solution Method

We planned to solve our problem by creating a model that incorporates given variables. We did this by using a basic open source agent-based Python/Mesa model in order to show the spread of the disease between different communities. We compared how the different demographics affected the severity of the illness in the different communities. We would also like to explore what factors battle the infections (healthcare, vaccinations, and lifestyle).

Our simulation was created with the intent of understanding influenza so that it can be controlled and combatted. By manipulating different variables, we hope to discover which methods of disease control are the most effective.

The first year we struggled to make the program accurately simulate real life. The immense complexity of the world makes it hard to replicate, so we've performed research to create a generalized model. Its accuracy is limited, but it is useful for tracking trends.

This year we were able to create a more realistic model, that would still be able to detect trends. Although, compared to the real world, it is still a fairly basic model. It neglects socio-economic data, the strength of the immune system, a more complex look at interactions between people, mental health, and the environment that the agents are in

Validation and Verification of the Model

- Validation - Are you building the right thing?
- Verification - Are you building it right?

Validation

Influenza is a virus that infects many during the winter and early spring. With research we have found that the disease has a main "type", Influenza A. This virus usually spreads once a year peaking in February. We have found that 5-20% of the United States will get the virus per year[2]. From these infections about 1-4% of people will die, mainly due to pre-existing health

---

[2] webMD

conditions [2]. Due to these facts we decided to ignore the lethality of the disease, and focus on it's spread (contagion).

The first year we used the nationwide average vaccine rate by age group[3]. In this group, 59.3% of children 0.5-17 years old have received a vaccine, 36.3% of adults 18-64 have received a vaccine, and 63.4% of people 65+ have received a vaccine. We incorporated this data into our program by giving 59.3% of children 0.5-17 years old a smaller chance of being infected, and by giving 36.3% of adults 18-64 a smaller chance of being infected. We excluding people 65 and up from the program. Since vaccinations play a major role in fighting Influenza, we decided to incorporate vaccines into our model. We have also researched and incorporated the amount of time you can pass the disease to other people for. This is important as it serves as a base for a more complex epidemiology model. From our research we have a hypothesis that people from neighborhoods without health care possibly have a higher chance of infection. This is a hypothesis that still requires more research to to back up our inferences.

In the second year we managed to improve upon how the disease is passed from one community to another community. We used a formula created by an group that was also trying to simulate the spread of Influenza[4]. The formula, $P(\text{inf}) = 1 - \Pi k \Pi j \Pi B \in \psi_{Ajkl}$ exp $\{-\lambda_{ij} d_{iAjkl} y_B\}$, when translated into simpler math it is just, 1 - exp$\{ -\lambda_{ij} d_a y_B\}$, with λ_{ij} correlating to a value (Appendix C1) that is dependent on the age of agent whose chance is being calculated, agent A, and of the age of the agent who is coming into contact with agent A, agent B . The variable d_a correlates to the amount of time that agent A was in contact with agent B (Appendix C2). The variable y_b is a binary value, that is 0 if agent B is not sick, and 1 if agent B is sick. This is so that if agent B is not sick, it will not be inputted into the equation that calculates if agent A gets sick. There is one

---

[3] Influenza (flu)

[4] Effectiveness of Interventions to Reduce Contact Rates during a Simulated Influenza Pandemic

$\exp\{ -\lambda_{ij} d_a y_B\}$ for each agent that Agent A comes into contact with which are multiplied all together before being subtracted from one. The application and values that go into the formula for each age group of each district is in Appendix C2. If the agent is infected, it will take 4-16 times steps or 1-4 days after for the agent to display symptoms of Influenza[5] it should be noted that the agent will be contagious during this time.

Verification

The entered data gave simulations that provided useful graphs that displayed the infection trends for the two communities. Trials 6 and 7 (shown below) provide a clear example. In Trial 6, the communities had about 10 people per room and the disease had a 40% (L=0.4) chance to infect others. In Trial 7 the communities also had about 10 people per room but the disease had a lower, 20% (L=0.2), chance to infect others.

Since it was easier to catch the disease in Trial 6, the infection sum shows a turning point later than the infection sum turning point in Trial 7 (Trial 6 shows a turning point of infections at about 22 days and Trial 7 shows a turning point of infections at about 45 days). Because the infection probability in Trial 7 was half of the probability of Trial 6, one would assume that the turning point in infectivity would have been later in Trial 6.

This is a useful simulation in that it shows the connection of the contagiousness parameter and the actual rate of spread of the disease in the model.

---

[5] CDC

First Year Selected Results

Trial 1 - Persistent disease, high infectivity

We started with a basic simulation to show the spread of a disease through a community of 150 agents. Figure 1 shows the number of infected agents versus the number of time steps (Days*2) for this simulation.

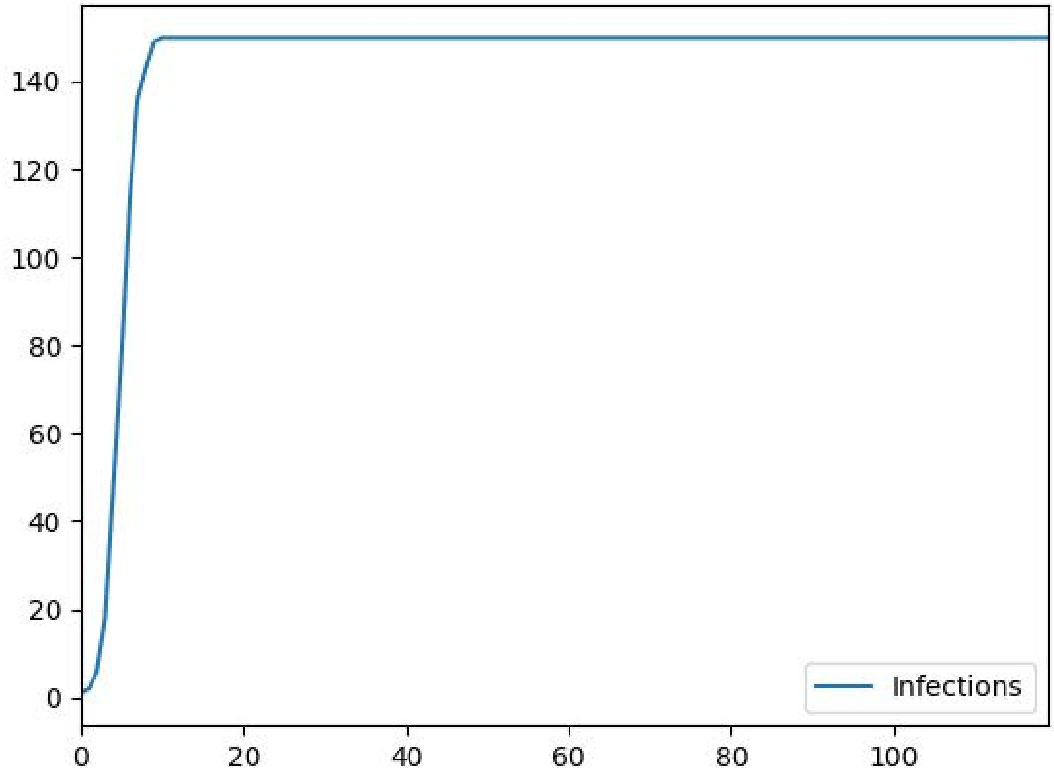


Figure 1: This is a trial that was carried out that involved 150 human agents and they infected others for ~60 days (day/night). There were 10 rooms per household, school, and work and the disease had a 50% (L=0.50) chance to infect others in the same room as them. Once the human agent is infected, they never become uninfected.

Trial 2 - Persistent disease, medium infectivity

Here we made it a little more difficult for people to catch the disease. You can see that it takes longer to spread.

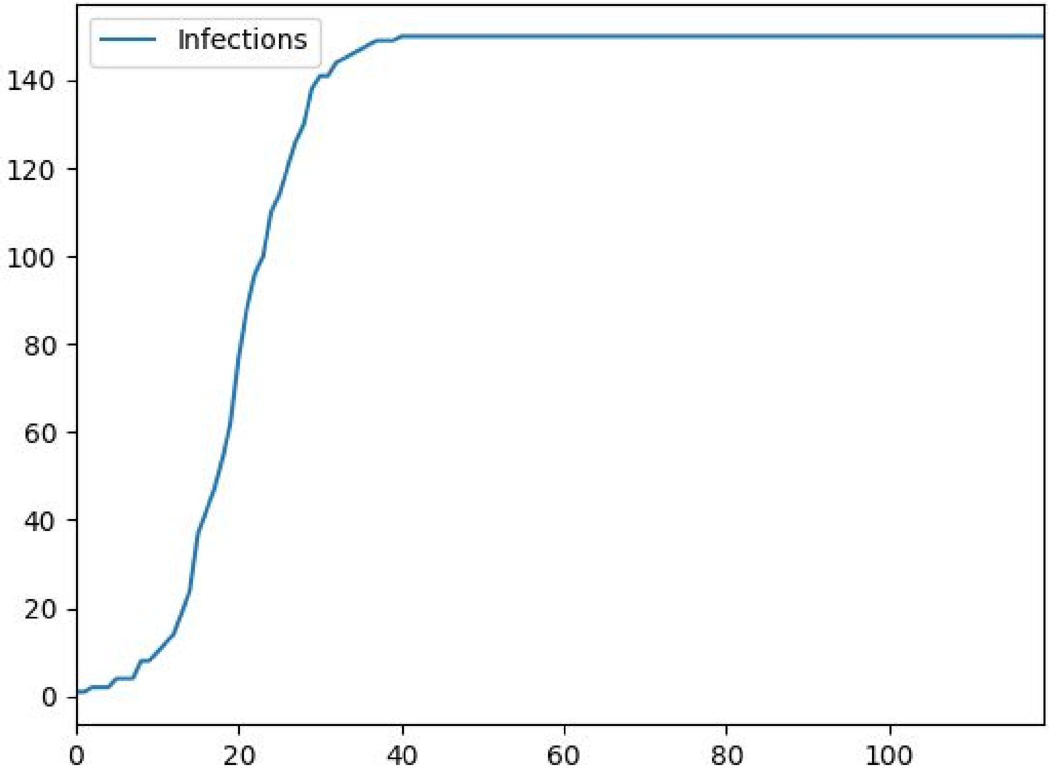


Figure 2: This is a trial that was carried out that involved 150 human agents and they infected others for ~60 days (day/night). There were 10 rooms per household, school, and work and the disease had a 10% (L=0.1) chance to infect others in the same room as them. Once the human agent is infected, they never become uninfected.

Trial 3 - Persistent disease, low infectivity

In Trial 3 we made it even more difficult for people to catch the disease. You can see that it takes much longer to spread to the full population.

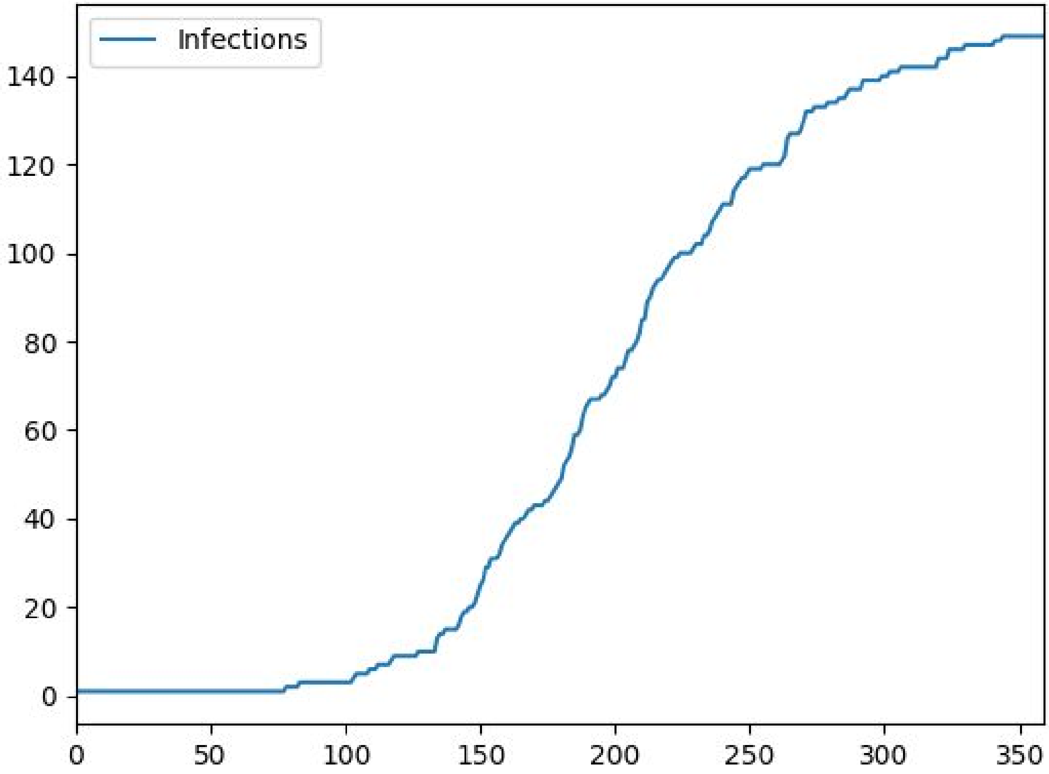


Figure 3: This is a trial that was carried out that involved 150 human agents and they infected others for ~180 days (day/night). There were 10 rooms per household, school, and work and the disease had a 1% (L=0.01) chance to infect others in the same room as them. Once the human agent is infected, they never become uninfected.

Trial 4 - Large Population, Medium infectivity

Now we tried more agents to see of the results would still finish in a reasonable amount of time.

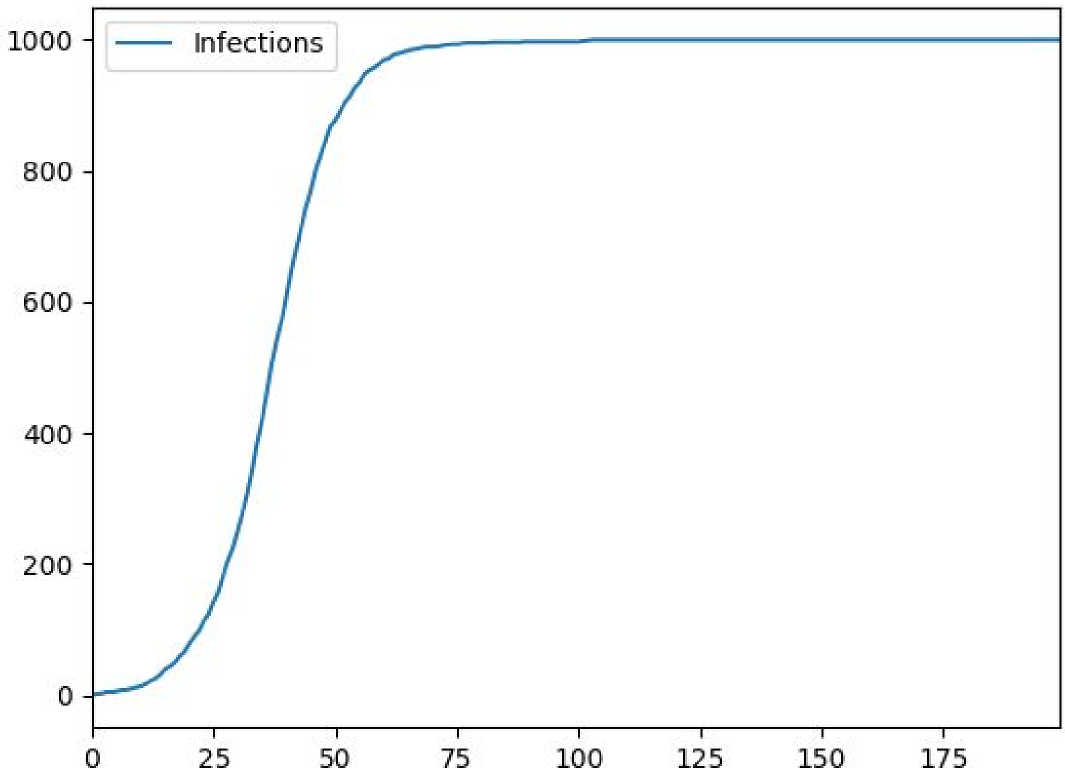


Figure 4: This is a trial that was carried out that involved 1000 human agents and they infected others for ~100 days (day/night). There were 100 rooms per household, school, and work and the disease had a 10% (L=0.1) chance to infect others in the same room as them. Once the human agent is infected, they never become uninfected.

Trail 5: Transmission between communities

In Trial 5, we improved the Python graphics so the infection rate in both communities could be observed. We added a line for the number of people that are immune overall.

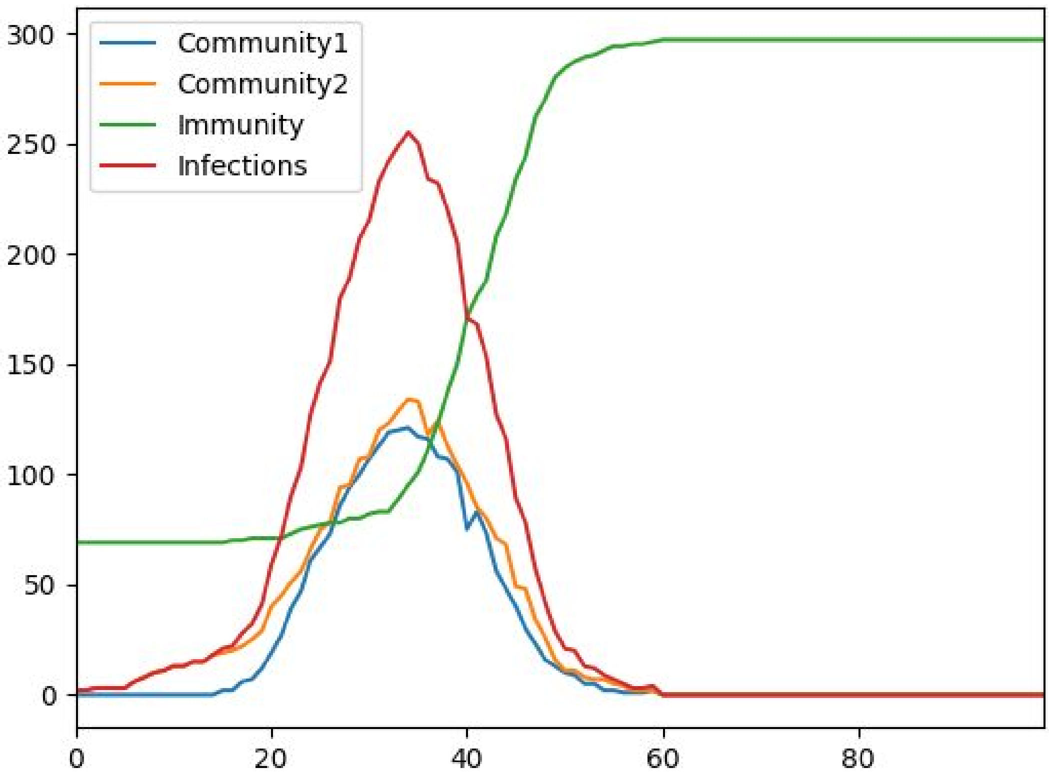


Figure 5: This is a trial that was carried out that involved 300 human agents and they infected others for ~30 days (day/night). There were 30 rooms per household, school, and work and the disease had a 20% (L=0.2) chance to infect others in the same room as them. The humans became contagious after day 2 and became immune after day 15. 20% got a vaccine.

Trial 6: Shortened course of disease but increased contagiousness

Here we studied the effect of increasing the contagiousness but reducing the duration of an individual's illness.

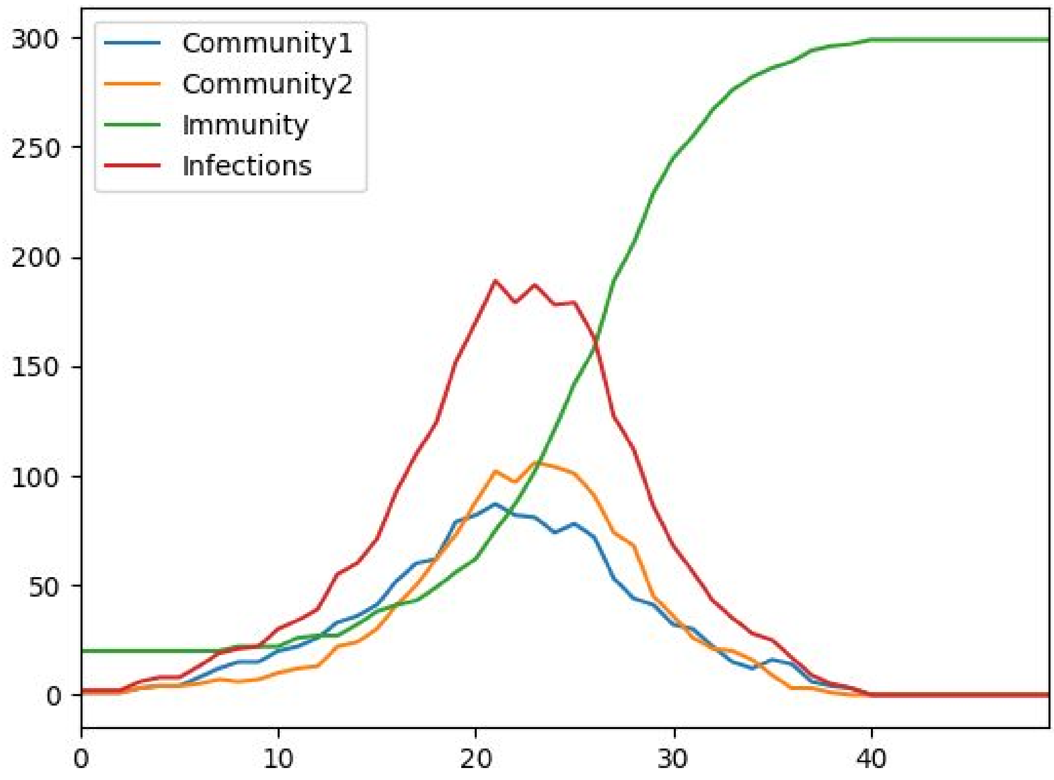


Figure 6: This is a trial that was carried out that involved 300 human agents and they infected others for ~30 days (day/night). There were 30 rooms per household, school, and work and the disease had a 40% (L=0.4) chance to infect others in the same room as them. The humans became contagious after day 3 and became immune after day 7. 5.5% got a vaccine.

Trial 7: Reduced contagiousness.

Here we mainly reduced the contagiousness from Trial 6.

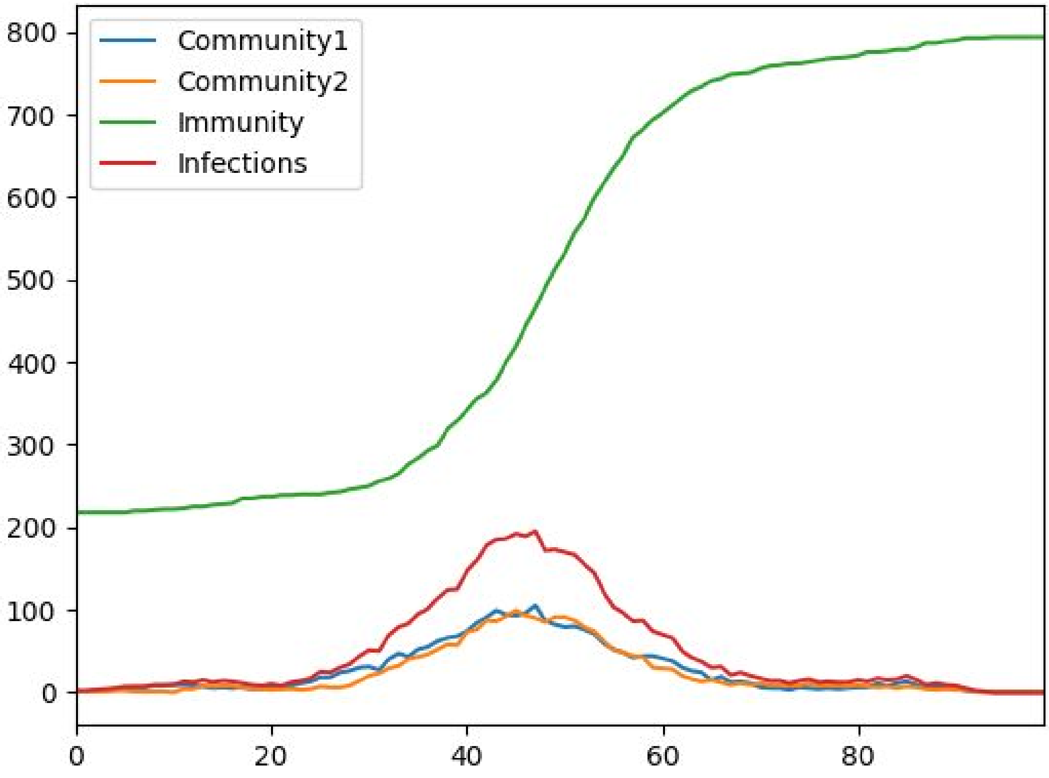


Figure 7: This is a trial that was carried out that involved 1000 human agents and they infected others for ~30 days (day/night). There were 100 rooms per household, school, and work and the disease had a 20% (L=0.2) chance to infect others in the same room as them. The humans became contagious after day 2 and became immune after day 5. 20% got a vaccine.

Trial 8: Introducing lifestyle and community health care parameters

For Trial 8 we rewrote parts of the model to use parameters Ms. Hooten suggested. Communities can now have a hospital or not. Also each community can have a different ratio of people living health lifestyles.

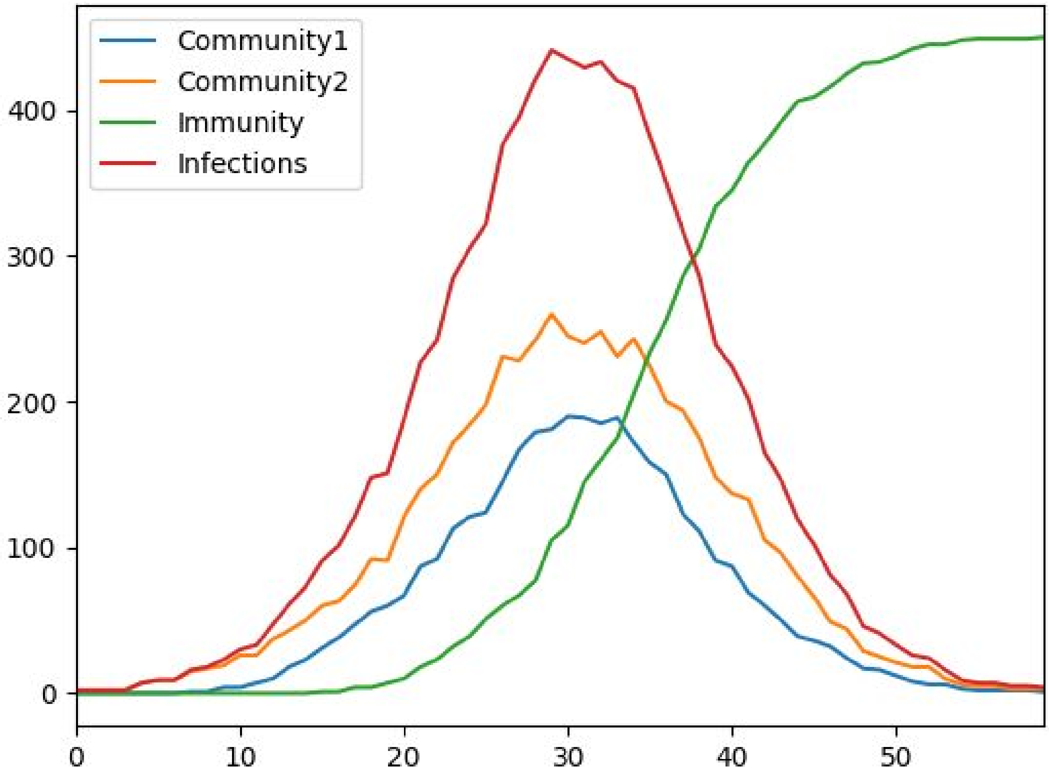


Figure 8: This is a trial that was carried out that involved 1000 human agents and they infected others for ~30 days (day/night). There were 200 rooms per household, school, and work and the disease had two likelihoods. The likelihood for healthier people to catch the disease is 20%. Comparatively the likelihood of catching the disease for unhealthier people is 90%.The humans became contagious after day 3 and became immune if healthy after 7 days. On the

contrary if you are unhealthy it takes 9 days. In first community there is access to healthcare, and in the second community there is no access to healthcare. In the first community .1% were vaccinated and in the second community 1% were vaccinated. It is worthwhile noting that community two picked up the disease faster and took longer to recover and become completely immune.

Trail 9; Tamiflu

We modelled the effects of Tamiflu, a drug that reduces the numbers affected by the influenza virus. It does this by attaching itself to the outside of the virus, preventing the virus from replicating. Studies have looked at the effects of Tamiflu. A article in the Journal of American Medical Association (JAMA) found that Tamiflu reduced contagion too 11% among family members. This simulation uses these statistics.

http://livehealthy.chron.com/still-contagious-after-taking-tamiflu-10334.html

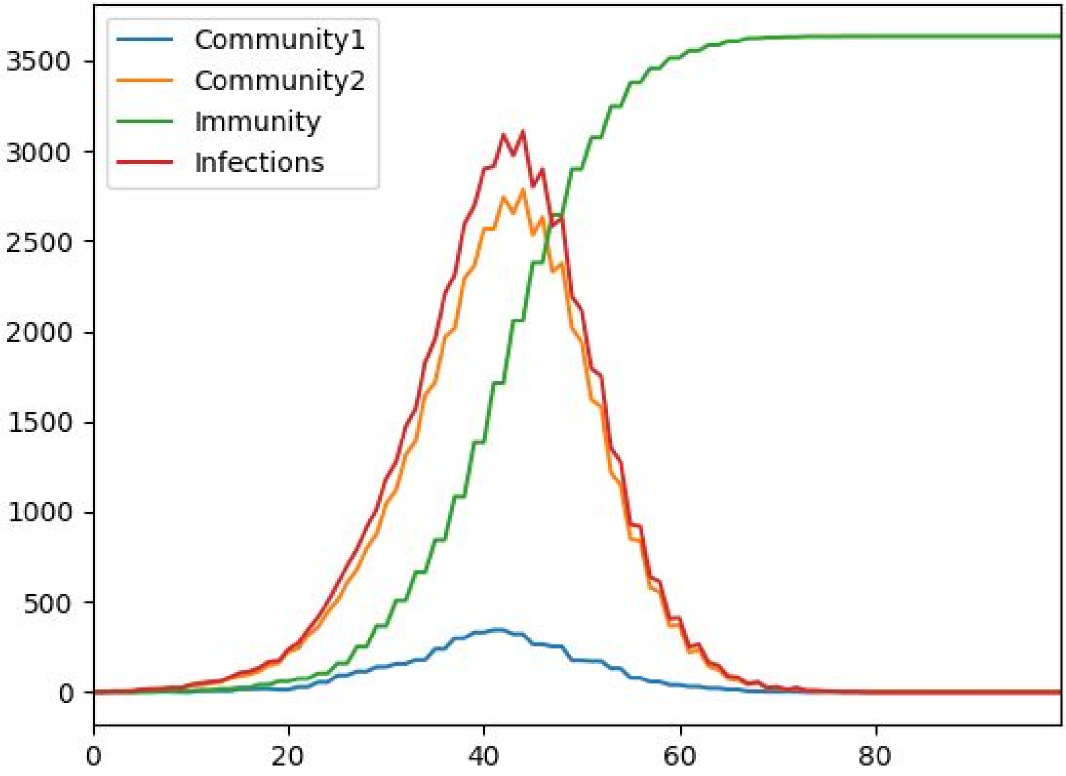


Figure 9: This is a trial that was carried out that involved 10000 human agents and they infected others for ~50 days (day/night). There were 1500 rooms per household, school, and

work and the disease had two likelihoods. The likelihood for healthier people to catch the disease is 11%. Comparatively the likelihood of catching the disease for unhealthier people is 75%.The humans became contagious after day 2 and became immune if healthy after 1 days. On the contrary, if you are unhealthy it takes 10 days. No one was vaccinated in either community. You can see the tamiflu was a success in community 1.

Second Year Selected Results

Trial 1 - Budget Cuts - Community 1 has hospital enabled

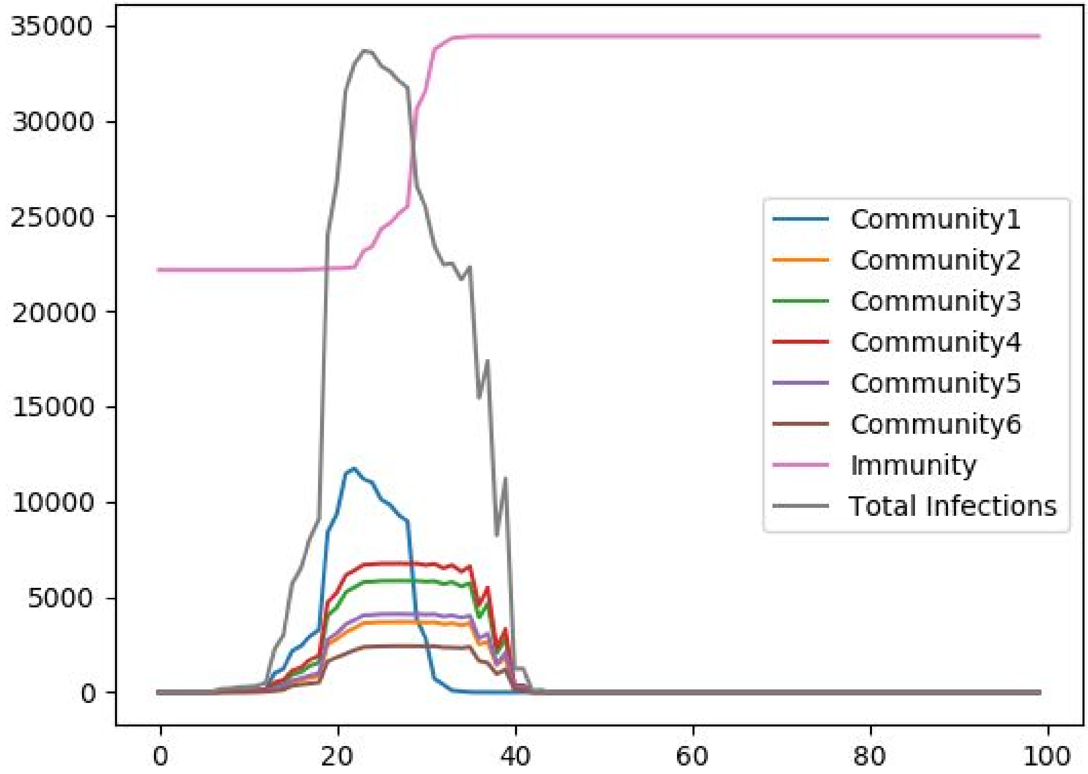


Figure 1: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there was only one hospital in the program in order to figure out which hospital had the biggest impact. In this trial, the

hospital was located in Community One. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 2 - Budget Cuts - Community 2 has hospital enabled

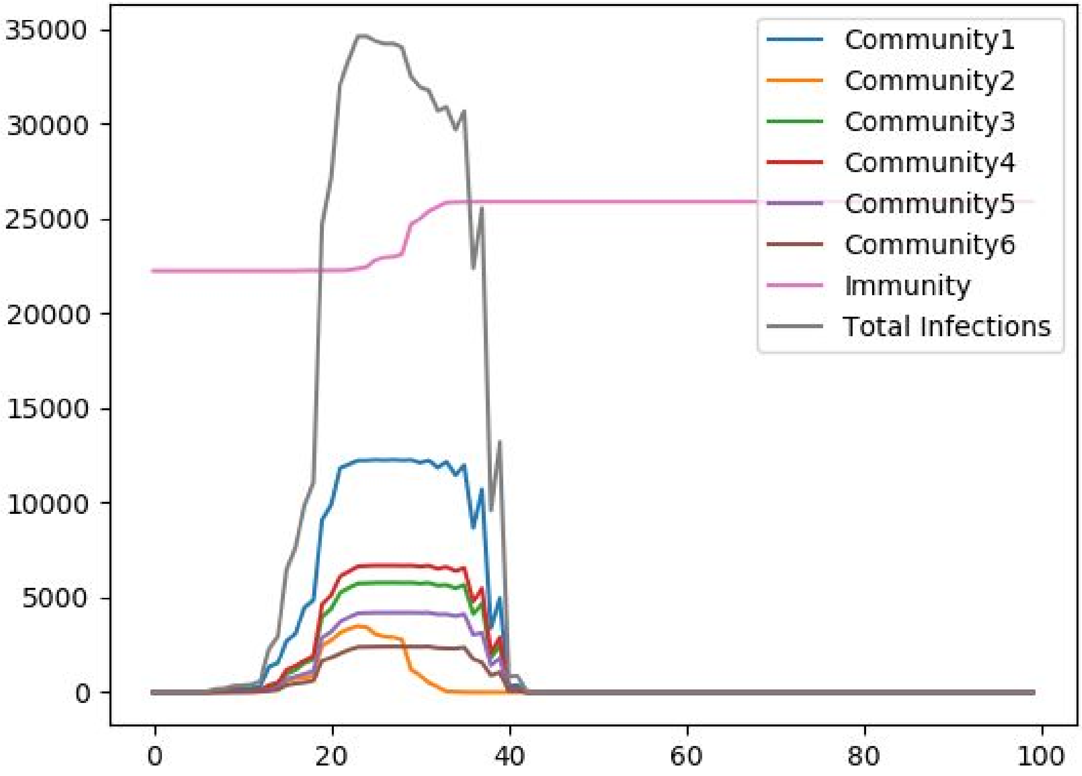


Figure 2: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there was only one hospital in the program in order to figure out which hospital had the biggest impact. In this trial, the hospital was located in Community Two. Each

community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 3 - Budget Cuts - Community 3 has hospital enabled

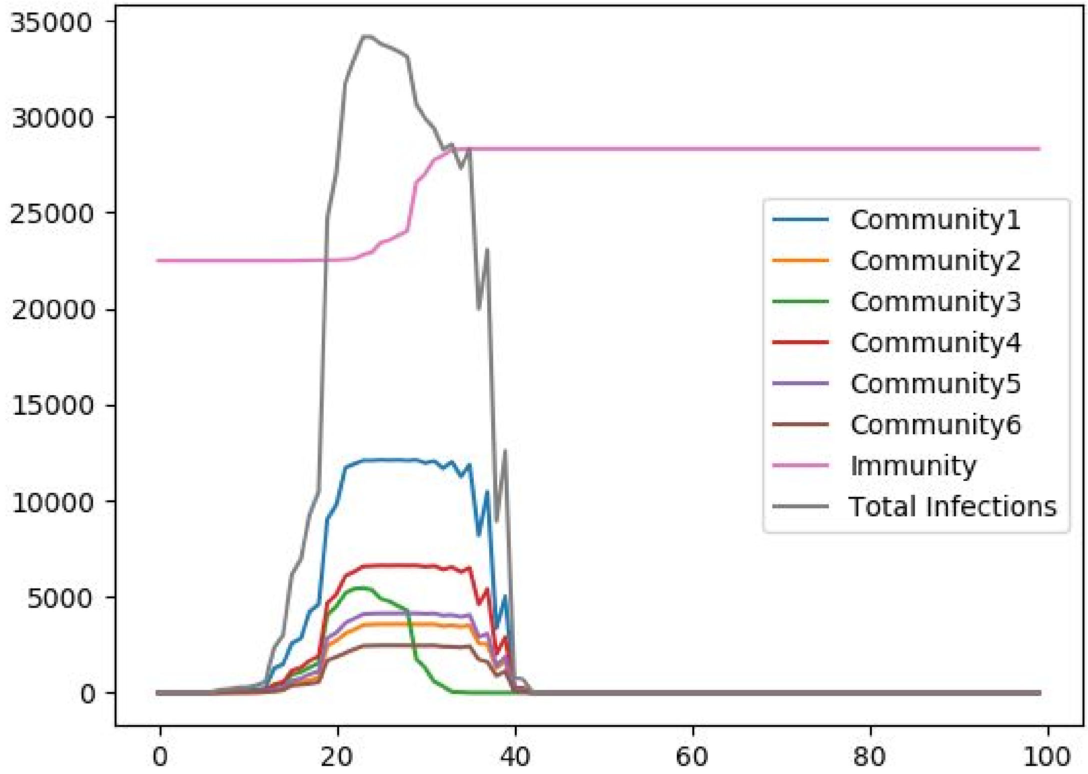


Figure 3: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there was only one hospital in the program in order to figure out which hospital had the biggest impact. In this trial,

the hospital was located in Community Three. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 4 - Budget Cuts - Community 4 has hospital enabled

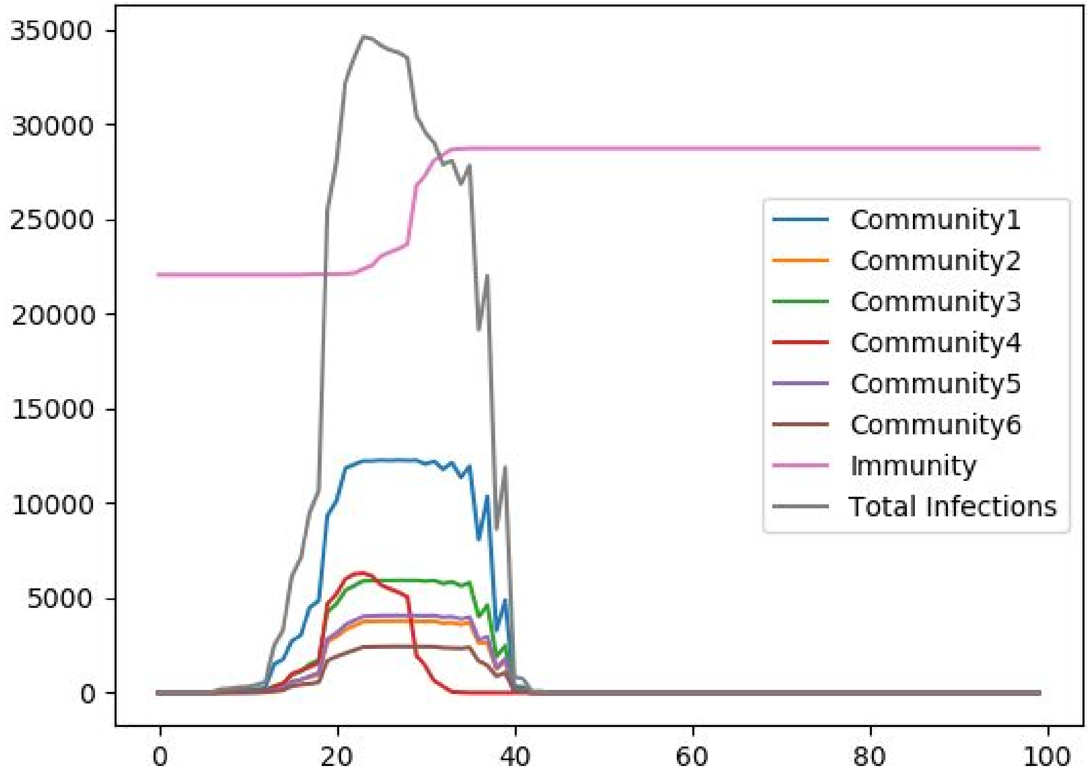


Figure 4: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there was only one hospital in the program in order to figure out which hospital had the biggest impact. In this trial, the hospital was located in Community Four. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 5 - Budget Cuts - Community 5 has hospital enabled

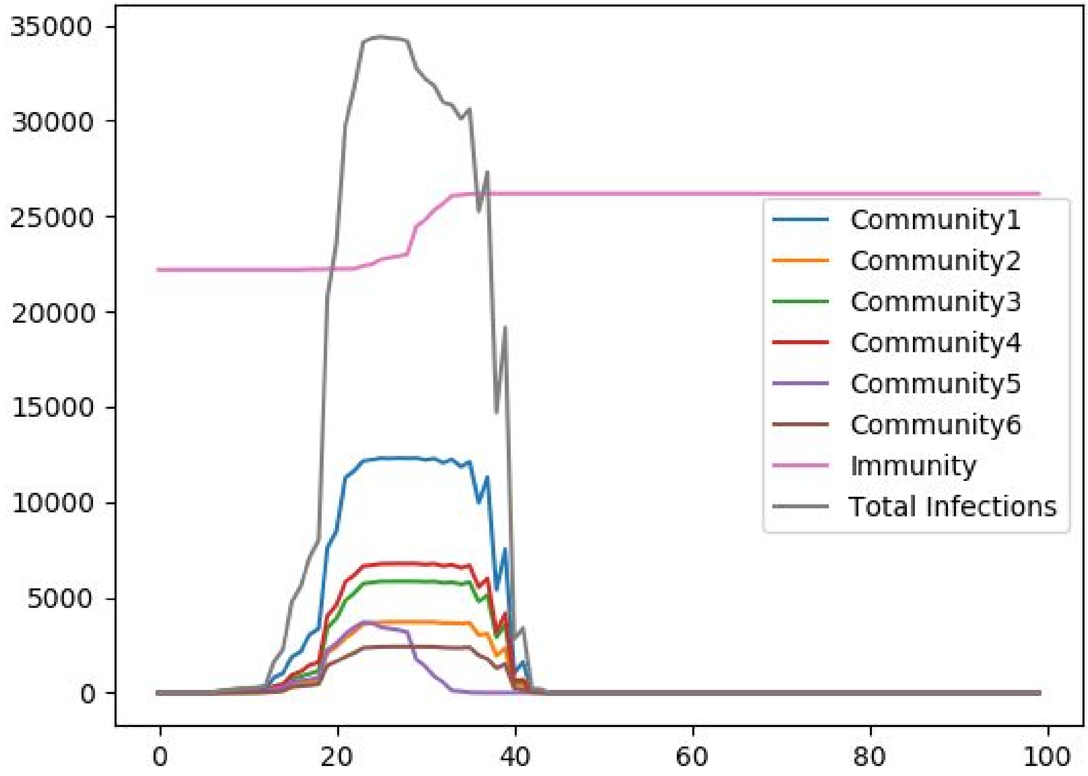


Figure 5: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there was only one hospital in the program in order to figure out which hospital had the biggest impact. In this trial, the hospital was located in Community Five. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 6 - Budget Cuts - Community 6 has hospital enabled

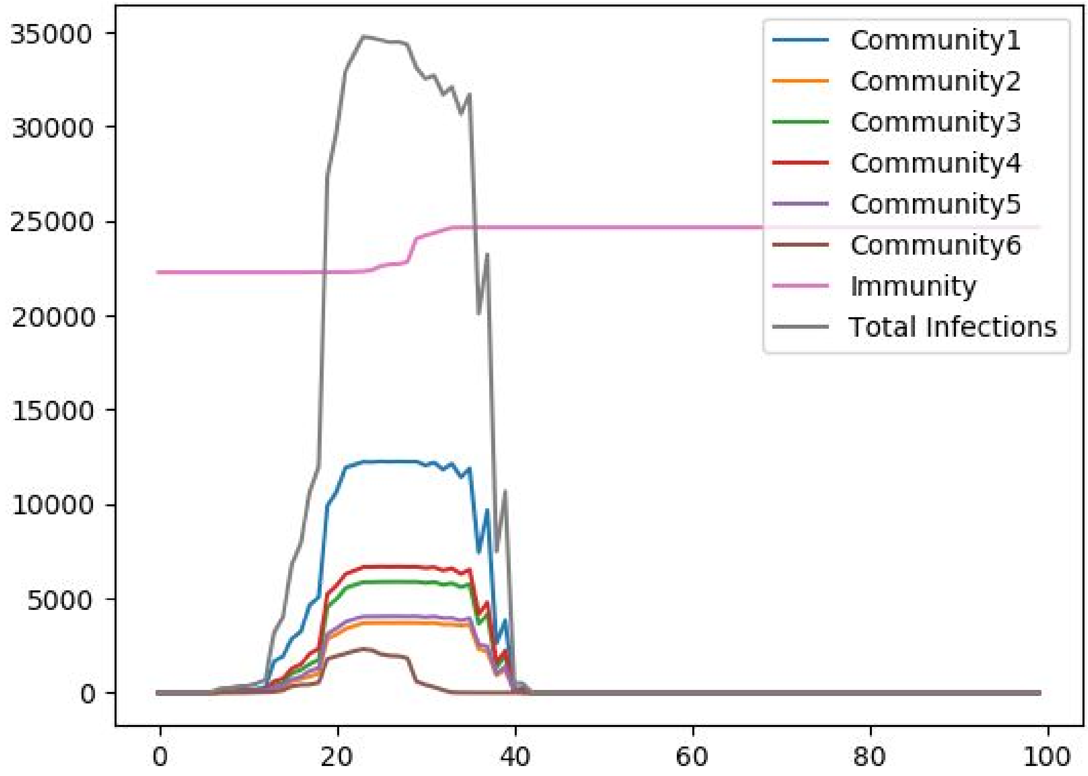


Figure 6: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there was only one hospital in the program in order to figure out which hospital had the biggest impact. In this trial, the hospital was located in Community Six. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 7 - Most Valuable Hospital - Community 1 Hospital Disabled

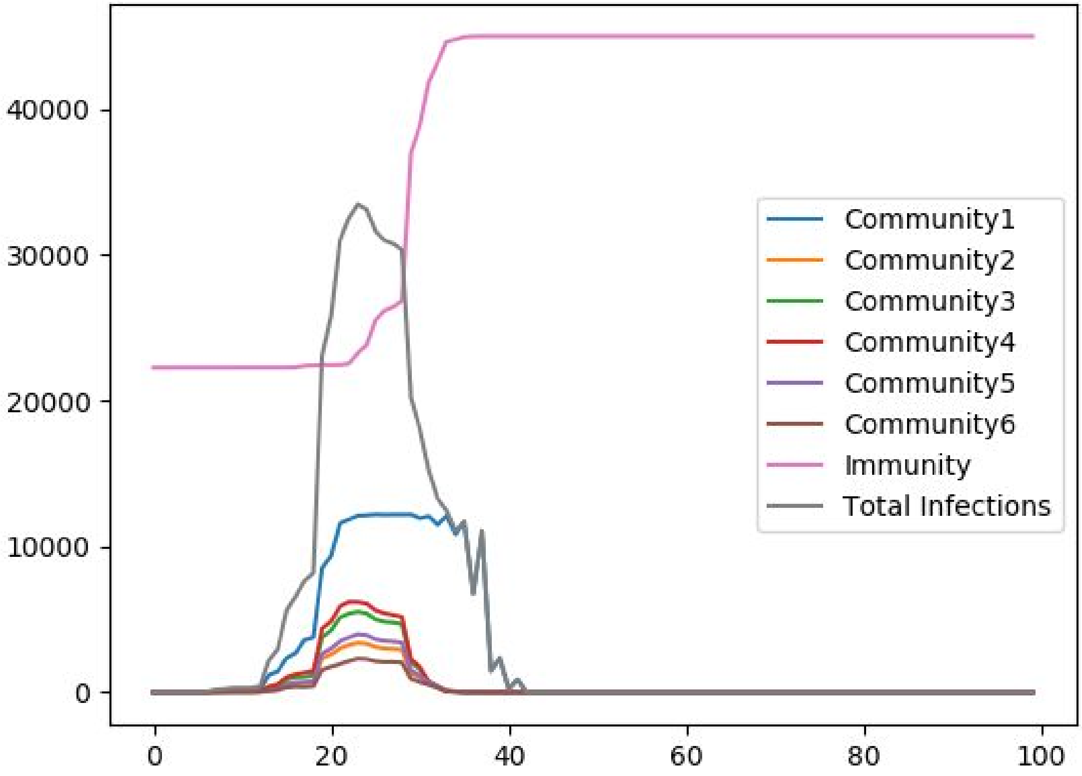


Figure 7: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there would be hospitals in all communities except one in order to figure out which hospital had the impact biggest impact. In this trial, there was no hospital located in Community One. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 8 - Most Valuable Hospital - Community 2 Hospital Disabled

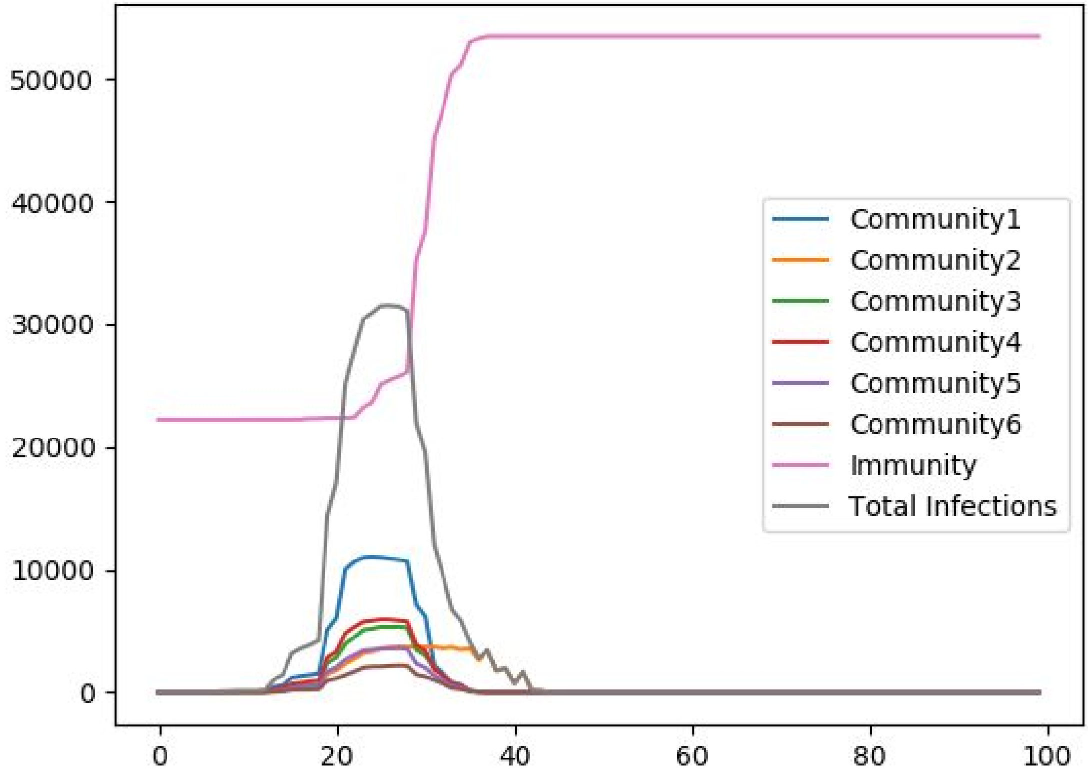


Figure 8: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there would be hospitals in all communities except one in order to figure out which hospital had the impact biggest impact. In this trial, there was no hospital located in Community Two. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 9 - Most Valuable Hospital - Community 3 Hospital Disabled

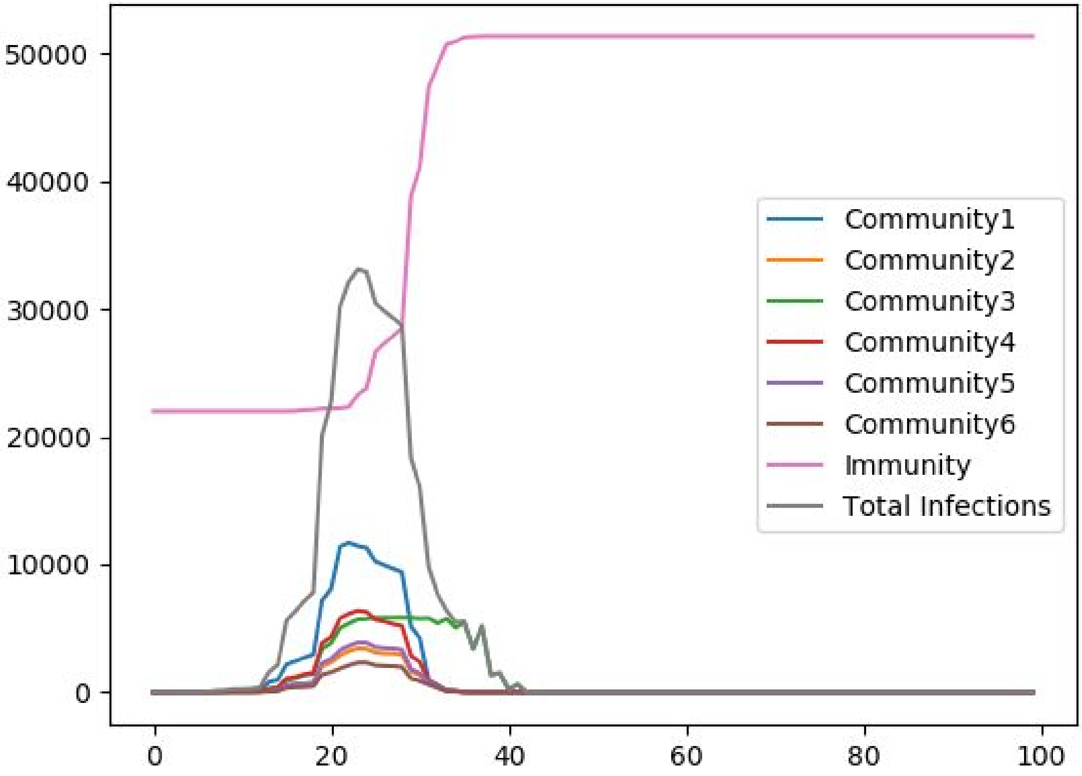


Figure 9: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there would be hospitals in all communities except one in order to figure out which hospital had the impact biggest impact. In this trial, there was no hospital located in Community Three. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 10 - Most Valuable Hospital - Community 4 Hospital Disabled

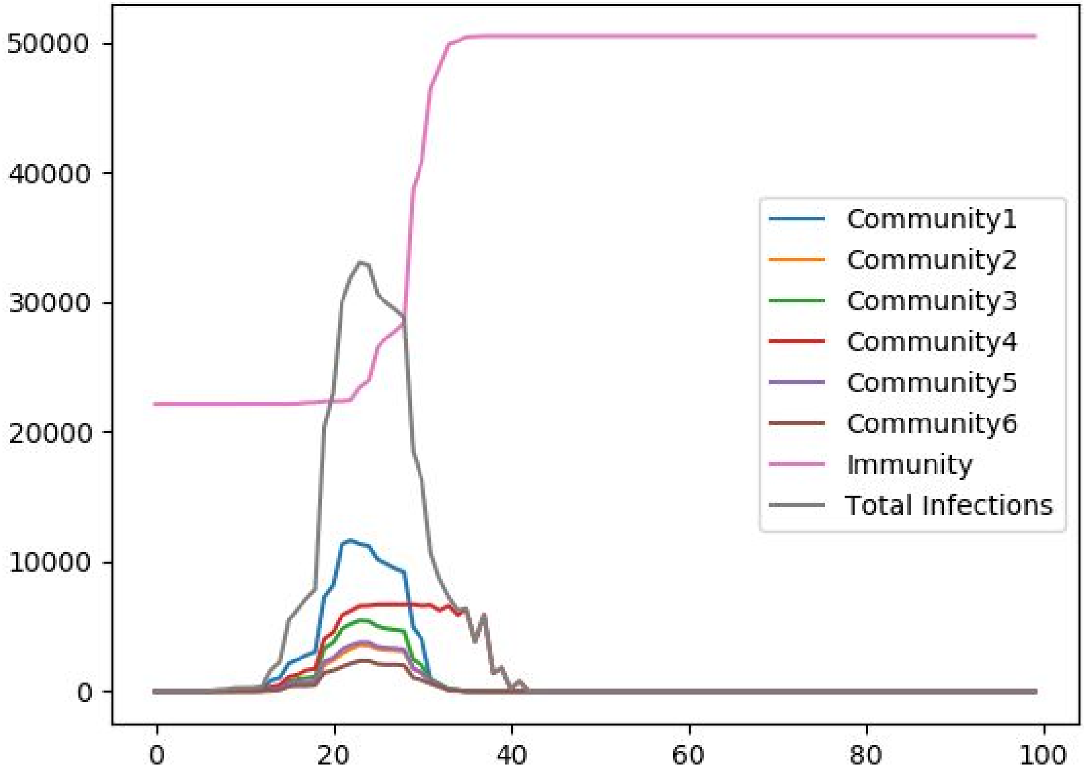


Figure 10: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there would be hospitals in all communities except one in order to figure out which hospital had the impact biggest impact. In this trial, there was no hospital located in Community Four. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation

Trial 11 - Most Valuable Hospital - Community 5 Hospital Disabled

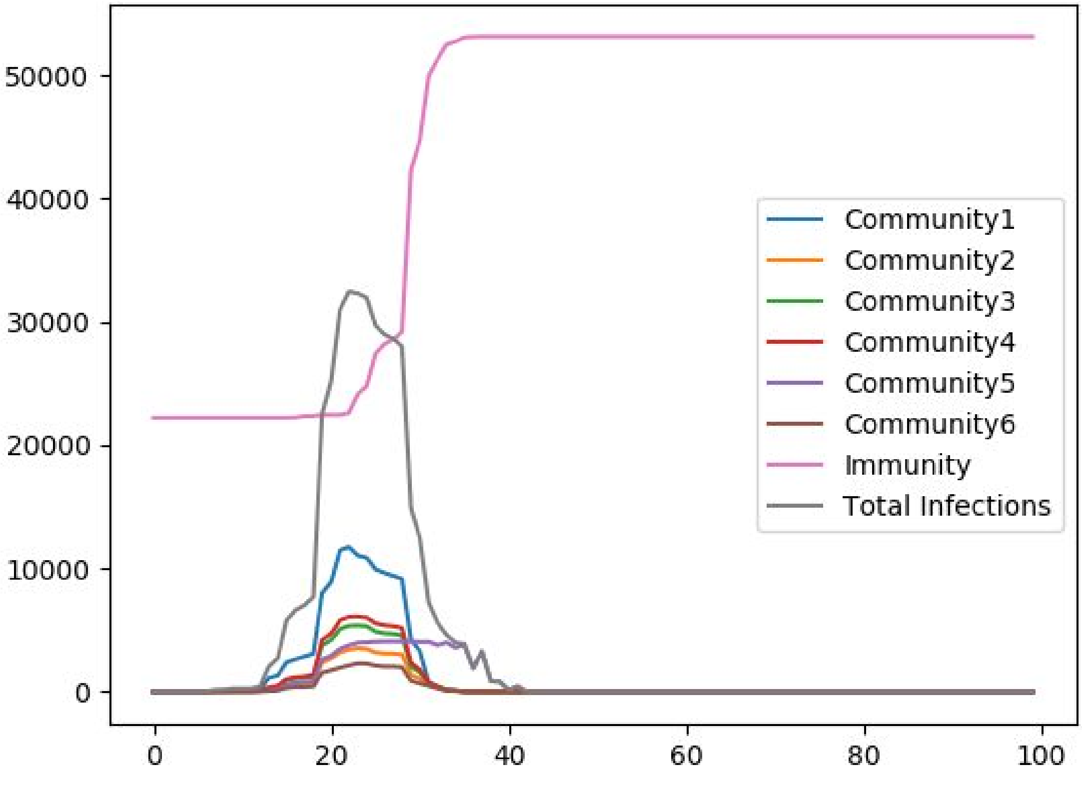


Figure 11: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there would be hospitals in all communities except one in order to figure out which hospital had the impact biggest impact. In this trial, there was no hospital located in Community Five. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation

Trial 12 - Most Valuable Hospital - Community 6 Hospital Disabled

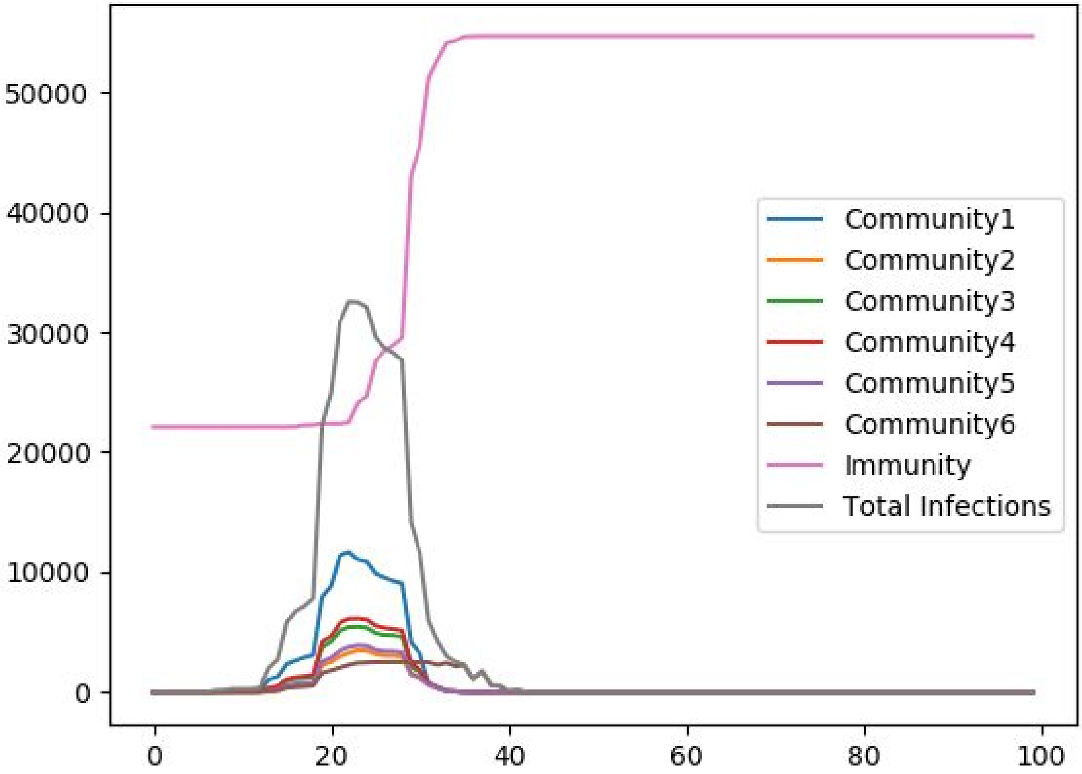


Figure 12: This is a trial that was carried out that had 50,000 human agents who infected each other for ~25 days (day/bus/night). There were 100 rooms per community, school, work and bus route. The disease had a likelihood to catch the disease for healthy people (30%), and a likelihood for unhealthy people (10%). We designed this trial so that there would be hospitals in all communities except one in order to figure out which hospital had the impact biggest impact. In this trial, there was no hospital located in Community Six. Each community was vaccinated on a weighted scale according to the amount of people in the entire simulation.

Trial 13 - Ten Percent Ratio to Real Scale

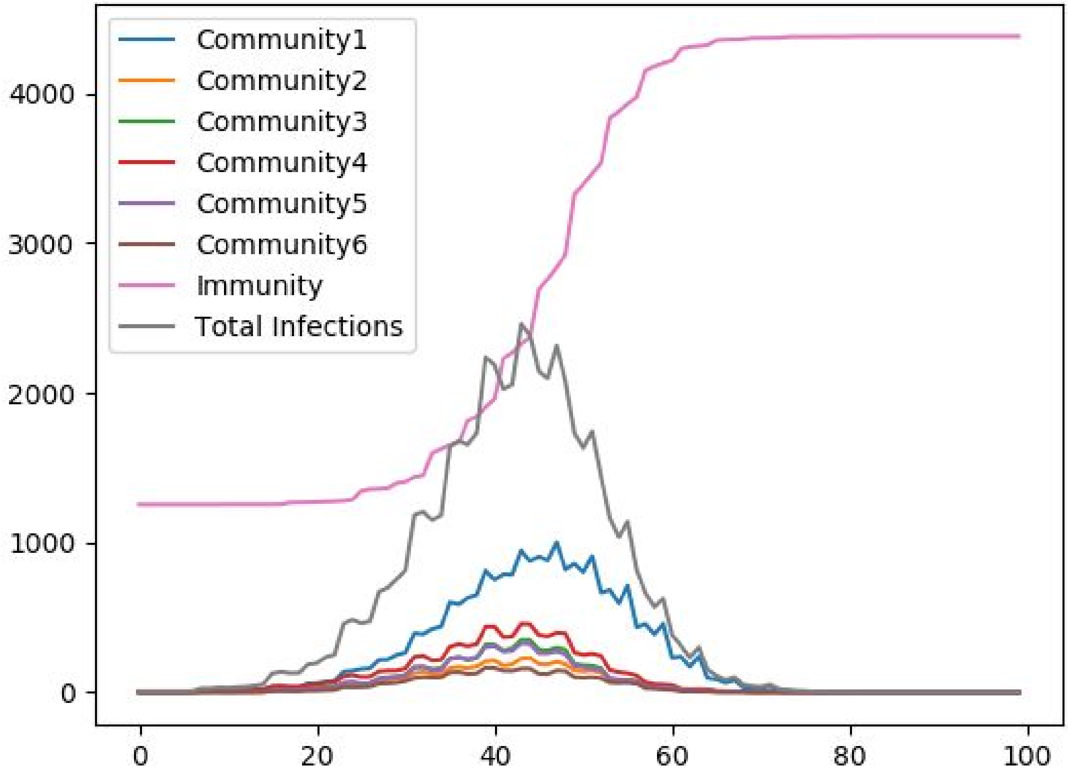
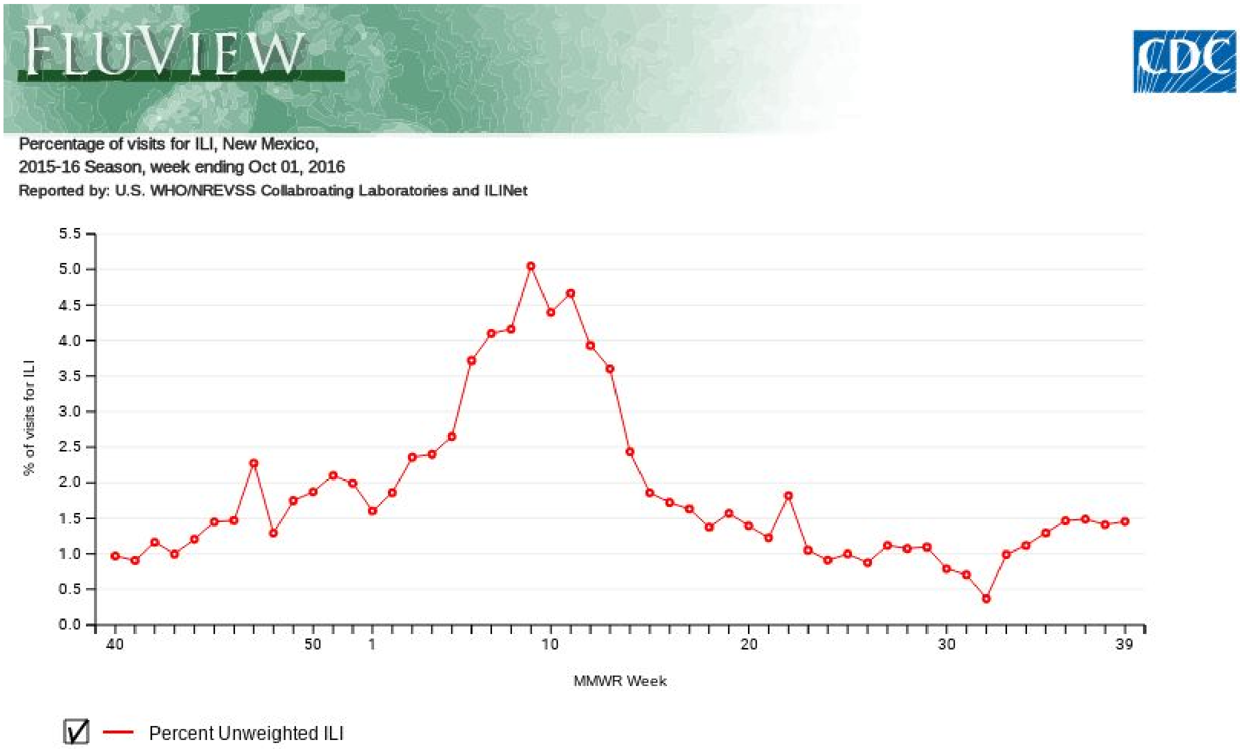


Figure 13 - We ran a full scale model of Albuquerque with 572,022 humans. The code runs through two stages. The first stage includes the setup of the model. The setup took an estimated two and a half hours and when we proceeded to the second stage, or running the interactions between the humans, the program continued to run for over six hours without progressing on to 'Day 1'. We decided that the program was taking too long, so with the same parameters, except with a tenth of the population, we ran another trial.

This is a trial that was carried out that involved 57,202 humans and they infected others for ~25 days (day/night/bus). There were approximately 22,881 rooms per household, 2,288 classrooms for school, 572 rooms for work, and 1,906 bus routes and the disease had two likelihoods. The likelihood for healthier people to catch the disease is 10%. Comparatively the likelihood of catching the disease for unhealthier people is 30%.The humans became contagious after day 6 and became immune if healthy after 9 days. On the contrary, if you are unhealthy it takes 16 days. The vaccination rates were weighted bases on the population of the entire model and the population of that community.

Analysis



[6]

---

[6] FluView

The graph above is the recorded counts of Influenza like Illness (ILI) from the 2015-2016 Flu season in New Mexico. Comparing our graphs to the actually recorded data some similarities should be noticed. One of which is both graphs overall shapes, which resemble a bell curve, even though our model was a hyper-realistic model that only ran for 25 "days". Another similarity is despite the overall bell curve shape both graph had irregularities on a small scale. This is an improvement compared to last year when our graphs were notably more smooth throughout the whole simulation, with few regularities.

In the first 12 graphs from this year , we tested to see how the presence of a hospital would affect the spread of the disease. In the first six graphs it is shown that all communities regardless of if they had a hospital that they all had the progressive upward curve. What separated the communities with hospitals from the countries without hospitals graphically was that the countries without hospitals took longer to completely recover from the flu outbreak, no matter the population size. We then proceeded to do the opposite, and have all of the communities had a hospital except one community, and see how that impacts the epidemic. The results we received were that the communities with hospitals recovered much faster , as we had predicted. While the community without the hospital took longer to recover from the epidemic. This trend continues through the rest of our tests. This reaffirmed our belief that access to hospitals helps alleviate the epidemic and end it faster than if there was no hospitals or healthcare centers.

Conclusion and Achievements

In the first year we started with a basic model, a model with just rooms and the ability to catch the disease. Then after doing some research, we added a vaccine percentage, or the percentage of the population that is immune to the disease. We consulted our teacher, who suggested that we add a variable to describe access to health care, or hospitals, into the communities and a variable for the agent's desire for a healthy lifestyle. Although our project provides only basic simulations, it is useful for comparing trials that gives potentially challenging outcomes. Data that challenges preconceptions introduces new questions that prompts the maturing of our simulation and the introduction of variables that we may have thought to be unnecessary.

Our most significant achievement is the ability to take several variables and obtain a simulation. Each simulation is different and each provides different insight. The example provided in the "validation" shows that comparing two simulations allows us to decide which variable we should change next. If we find that the rate at which people come in contact is more of a factor than vaccines or contagiousness, then we can improve the conditions of the community by lowing the "room's density" and see if it does in fact improve conditions (lower number of infections).

We have successfully added an additional four communities, as well as an age variable and a public transportation variable. We found out, just as we had originally thought, that public transportation does in fact spread the disease significantly faster than if they humans traveled directly from their house to their school or work and vice versa. Age also played a large role in the simulation.

If we continue this project into the future, we intend to add weekends to the program. This will slow the spread of the disease in the school and work settings and on the bus, but allow it to grow at a higher concentration in the communities. Even though the percentage is small, we can also include mortality rates of the flu.

Selecting a language

We chose Python because we went on the Santa Fe Tech tour in Socorro during the 2016/2017 Supercomputing Computing Challenge and discovered that Python fit our needs. We are using Python 3.6.5 for our project. It creates a nice graph when all of the variables are filled. Python was more complicated than was required, but it still performed well.

Choosing Mesa for agents

Our program requires us to make a model about the spread of disease. Writing models from scratch in Python is quite difficult, so we decided to look online for premade libraries to make the job easier. What we found was Mesa, an open-source framework[7].

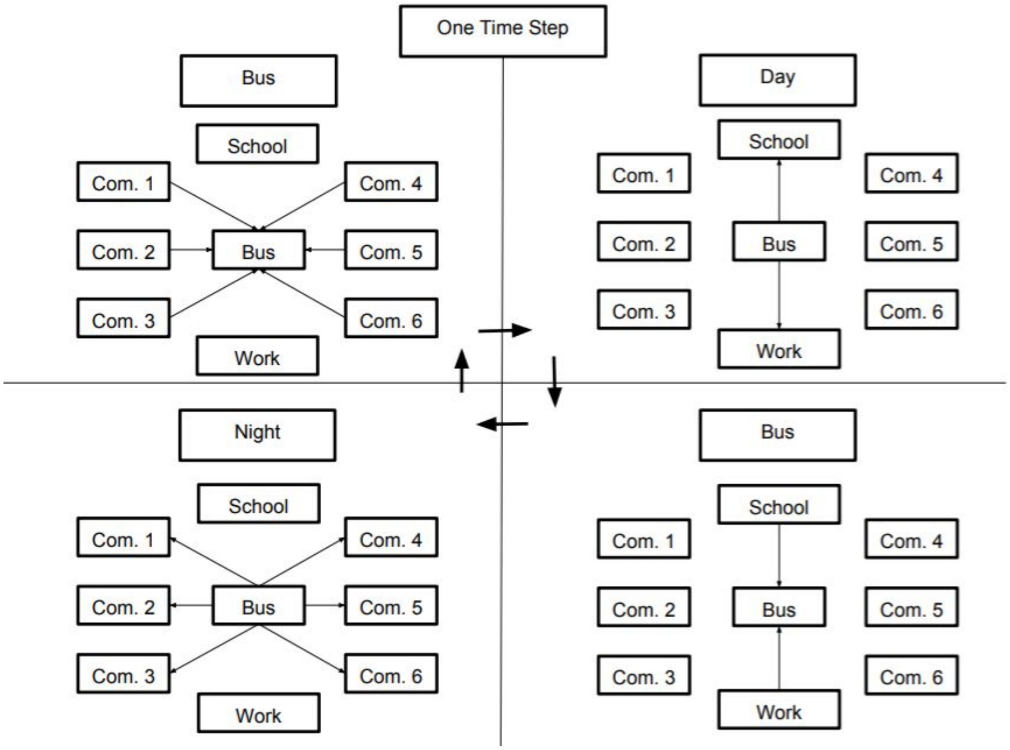
Mesa was first released on June 21, 2015 by the programmers at George Mason University in Virginia. We used a video and a web page to learn how to use it. It is an open-source basic agent-based modeling Framework. Mesa simplifies tasks and allows the user to analyse and visualize their work. Mesa has built-in support for the agents, like spatial grids and agent scheduling.

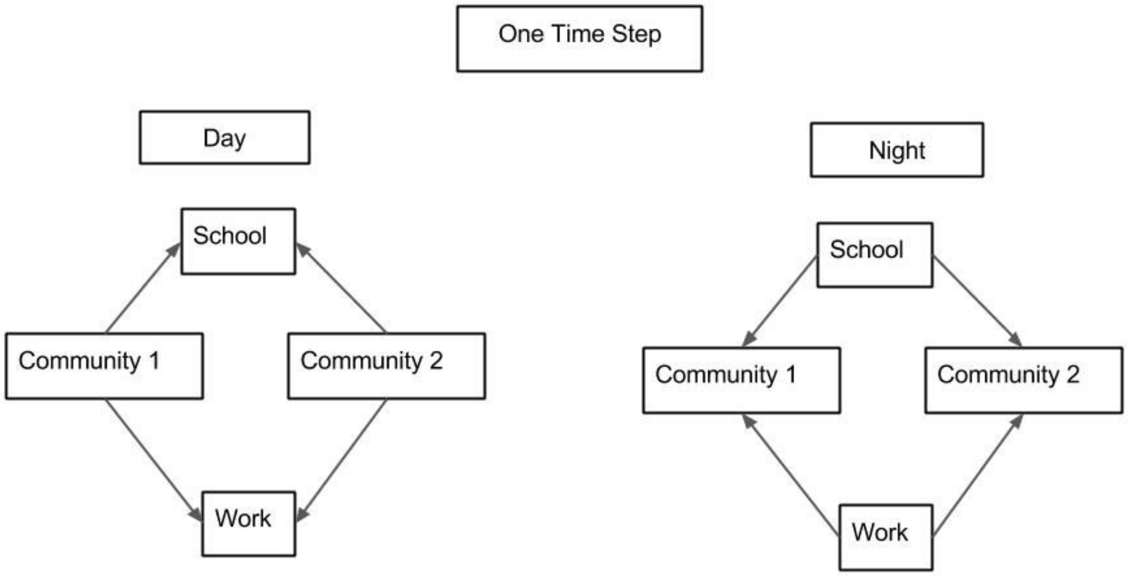
---

[7] http://mesa.readthedocs.io/en/latest/tutorials/intro_tutorial.html

Day and Night for agents

In our model, humans travel between their nighttime communities, their daytime school or work and a bus to and from their community or daytime activity. All humans take the bus to and from their assigned community and their school or work. The communities are divided into households and infected human can infect others within their household. The school and work are divided into rooms, where inhabitants can infect each other. The buses are divided into individual bus routes, when inhabitants can infect each other. A time step is comprised of a day, a bus route, a night, and a second bus route.





The parameters we used are number of humans, number of rooms, infectivity likelihood, contagion start, contagion end, and the percentage of vaccinated humans.

The Big Data Structure

We hold all the agent's placement in a matrix. The height (number of rows) is based on the number of houses or rooms in each community. There are four columns that correspond to the day and night placement of the agents. Each agent has its own concept of its location during the day or night, and will move around the matrix in accordance to the timestep cycle.

Bus	Com. 1	Com. 2	Com. 3	Com. 4	Com. 5	Com. 6	School	Work
Route 1	House 1	House 1	House 1	House 1	House 1	House 1	Classroom 1	Office 1

Route 2	House 2	House 2	House 2	House 2	House 2	House 2	Classroom 2	Office 2
Route 3	House 3	House 3	House 3	House 3	House 3	House 3	Classroom 3	Office 3
Route 4	House 4	House 4	House 4	House 4	House 4	House 4	Classroom 4	Office 4
Route 5	House 5	House 5	House 5	House 5	House 5	House 5	Classroom 5	Office 5

Code

```
# Project 2017-2018 ATC - Flu Transmission
# Built on Mesa Agent Library from George Mason Univ.
# Written by Ben Thorp
# ATC-3 Ben Thorp, Alex Baten, Teddy Gonzales
# Version 1.1
#   Added start and stop of contagiousness
# Version 1.2
#   Added HealthCareAccess and HealthLifeStyle
# Version 1.3
#   Added community3 and buses
# Version 1.4
#   Added community4, community5, community6, and age groups

from mesa import Agent, Model
from mesa.time import RandomActivation
import random
from mesa.space import MultiGrid
from mesa.datacollection import DataCollector
```

```
import matplotlib.pyplot as plt
from math import exp, expm1

# Function that computs the number of infections for graphing
def compute_infections(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in community 1 for graphing
def compute_infections_c1(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                if (human.community is 1):
                    total_inf_count += 1
    return total_inf_count

# Function that computs the number of infections in community 2 for graphing
def compute_infections_c2(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                if ( human.community is 2):
                    total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in community 3 for graphing
def compute_infections_c3(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                if (human.community is 3):
```

```
                    total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in community 4 for graphing
def compute_infections_c4(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                if (human.community is 4):
                    total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in community 5 for graphing
def compute_infections_c5(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                if (human.community is 5):
                    total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in community 6 for graphing
def compute_infections_c6(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                if (human.community is 6):
                    total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in the bus route for graphing
def compute_infections_b1(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
```

```
            total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in the work for graphing
def compute_infections_work(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                total_inf_count += 1
    return total_inf_count

# Function that computes the number of infections in the school for graphing
def compute_infections_school(model):
    total_inf_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.infected:
                total_inf_count += 1
    return total_inf_count

# Function that computs the immunity for everyone for graphing
def compute_immunity(model):
    total_im_count = 0
    for cell in model.grid.coord_iter():
        cell_content, x, y = cell
        for human in cell_content:
            if human.immunity:
                total_im_count += 1
    return total_im_count

class InfectModel(Model):
    """A Mesa Model to simulate the spread of disease through a home and work environment"""

    # __init__ creates the model
    def __init__(self, N,
                 hls_com1, hls_com2, hls_com3, hls_com4, hls_com5, hls_com6, #lowL, highL,
                 con_start, con_end_short, con_end_long,
                 com1_hca, com2_hca, com3_hca, com4_hca,
                 com5_hca, com6_hca, vac_com1, vac_com2,
                 vac_com3, vac_com4, vac_com5, vac_com6):
```

```
# N - the total number of humans
# h - the number of houses or rooms in each environment

# hls_com1 - the percentage of people in community1
#   that are living a healthy lifestyle
# hls_com2 - the percentage of people in communtiy2
#   that are living a healthy lifestyle
# hls_com3 - the percentage of people in communtiy3
#   that are living a healthy lifestyle
# hls_com4 - the percentage of people in communtiy4
#   that are living a healthy lifestyle
# hls_com5 - the percentage of people in communtiy5
#   that are living a healthy lifestyle
# hls_com6 - the percentage of people in communtiy6
#   that are living a healthy lifestyle
# lowL - a low likelyhood to catch the disease (
#   assosiated with healthy lifestyle)
# highL - a high likelyhood to catch the disease (
#   assosiated with healthy lifestlye)

# con_start - when the disease starts to be contagious
# con_end_short - when the disease stops being contagious with health care
# con_end_long - when the disease stops being contagious without health care

# com1_hca - the health care access in community1
# com2_hca - the health care access in community2
# com3_hca - the health care access in community3
# com4_hca - the health care access in community4
# com5_hca - the health care access in community5
# com6_hca - the health care access in community6

# vac_com1 - percentage of vaccinated humans in community1
# vac_com2 - percentage of vaccinated humans in community2
# vac_com3 - percentage of vaccinated humans in community3
# vac_com4 - percentage of vaccinated humans in community4
# vac_com5 - percentage of vaccinated humans in community5
# vac_com6 - percentage of vaccinated humans in community6

self.num_agents = N
self.grid = MultiGrid(9, 78000, True)
self.schedule = RandomActivation(self)

self.community1_hca = com1_hca
```

```
self.community2_hca = com2_hca
self.community3_hca = com3_hca
self.community4_hca = com4_hca
self.community5_hca = com5_hca
self.community6_hca = com6_hca

self.con_end_short = con_end_short
self.con_end_long = con_end_long

self.vaccinated_com1 = vac_com1
self.vaccinated_com2 = vac_com2
self.vaccinated_com3 = vac_com3
self.vaccinated_com4 = vac_com4
self.vaccinated_com5 = vac_com5
self.vaccinated_com6 = vac_com6

self.healthy_lifestyle_com1 = hls_com1
self.healthy_lifestyle_com2 = hls_com2
self.healthy_lifestyle_com3 = hls_com3
self.healthy_lifestyle_com4 = hls_com4
self.healthy_lifestyle_com6 = hls_com5
self.healthy_lifestyle_com5 = hls_com6

#self.low_likelihood = lowL
#self.high_likelihood = highL

# Create N humans for the model
for i in range(self.num_agents):
    if i% 10000 == 0 :
        print(i)
    a = Human(i, self, con_start)
    self.schedule.add(a)

    #y = random.randrange(self.grid.height)
    self.grid.place_agent(a, (a.community, a.household))

    # Add the agent to a random grid cell
    #if (a.community == 1):
    #    x = 0
    #elif (a.community == 3):
    #    x = 4
    #elif (a.community == 4):
    #    x = 6
```

```
        #elif (a.community == 5):
        #    x = 7
        #elif (a.community == 6):
        #    x = 8
        #else:
        #    x = 1



    # Initialize timestep
    self.timestep=0
    #self.day=True
    self.transportation_cycle = 0

    # Initialize the software that collects the data each timestep
    self.datacollector = DataCollector(
        model_reporters={"Community1": compute_infections_c1,
                         "Community2": compute_infections_c2,
                         "Community3": compute_infections_c3,
                         "Community4": compute_infections_c4,
                         "Community5": compute_infections_c5,
                         "Community6": compute_infections_c6,
                         #"Work Place": compute_infections_work,
                         #"School": compute_infections_school,
                         #"Public Transport": compute_infections_b1,
                         "Immunity": compute_immunity,
                         "Total Infections": compute_infections,

                         }
        )

def step(self):
    # Collects the data for this timestep
    self.datacollector.collect(self)

    self.schedule.step()
    self.timestep+=1
    self.transportation_cycle = (self.timestep % 4 )

# run_steps steps the model forword for "steps" (days/nights)
def run_steps(self, steps):
    for i in range(steps*4):
        if i%4 == 0:
```

```
                s='Day: ' + str(i//4)
                print(s)
            # Quadupling steps makes correct number of day/night/bus cycles
            # Because a day/night cycle takes two timesteps
            self.step()
        inf_data = self.datacollector.get_model_vars_dataframe()
        inf_data.plot()
        plt.show()

class Human(Agent):
    """An agent that represents one human in the model"""
    def __init__(self, unique_id, model, con_start):
        # unique_id - the human's id number
        # model - the Mesa simulation class
        # likelihood - the chance that the infection spreads from one human to another
        # con_start - when the disease starts to be contagious

        # Call the Mesa agent setup
        super().__init__(unique_id, model)

        # Initialize the human variables
        tempa = [1, 2, 3, 4, 5, 6]
        com_size_weights = [.34, .10, .14, .22, .13, .07]
        temp = random.choices(tempa, weights=com_size_weights, k=1)
        self.community = temp[0]
        #self.household=random.randrange(model.grid.height)
        tempb = model.num_agents
        tempc = com_size_weights[(self.community) - 1]
        self.household=random.randrange(int(tempb * tempc * 0.3))
        #self.schoolroom=random.randrange(model.grid.height)
        self.schoolroom=random.randrange(int(model.num_agents/25))
        #self.workroom=random.randrange(model.grid.height)
        self.workroom=random.randrange(int(model.num_agents/100))
        #self.bus_number=random.randrange(model.grid.height)
        self.bus_number=random.randrange(int(model.num_agents/30))

        # Initialize the age to match Albuquerque's age profile
        self.age = self.my_age()

        # Initializes the age bracket
        self.ic = [[0.00061,0.00033,0.00080], [0.00053,0.00032,0.00080],
[0.00057,0.00029,0.00102]]
```

```
self.bracket_youth = 0
self.bracket_adult = 1
self.bracket_elderly = 2

if self.age >= 5 and self.age <= 18:
    self.age_bracket = self.bracket_youth

elif self.age >= 19 and self.age <= 65:
    self.age_bracket = self.bracket_adult

elif self.age >= 66 and self.age <= 95:
    self.age_bracket = self.bracket_elderly

else:
    print ("Something went wrong when assigning ages")



# Initialize disease variables
# self.likelihood = likelihood
#tmp_rnd = random.random()
#if self.community == 1:
#    if (tmp_rnd <= model.healthy_lifestyle_com1):
#        self.likelihood = model.low_likelihood
#    else:
#        self.likelihood = model.high_likelihood

#elif self.community == 2:
#    if (tmp_rnd <= model.healthy_lifestyle_com2):
#        self.likelihood = model.low_likelihood
 #   else:
 #       self.likelihood = model.high_likelihood

#elif self.community == 3:
#    if (tmp_rnd <= model.healthy_lifestyle_com3):
#        self.likelihood = model.low_likelihood
#    else:
#        self.likelihood = model.high_likelihood

#elif self.community == 4:
#    if (tmp_rnd <= model.healthy_lifestyle_com4):
#        self.likelihood = model.low_likelihood
#    else:
```

```
        #         self.likelihood = model.high_likelihood

        #elif self.community == 5:
        #     if (tmp_rnd <= model.healthy_lifestyle_com5):
        #         self.likelihood = model.low_likelihood
         #    else:
         #        self.likelihood = model.high_likelihood

      #   elif self.community == 6:
       #      if (tmp_rnd <= model.healthy_lifestyle_com6):
       #          self.likelihood = model.low_likelihood
        #     else:
        #         self.likelihood = model.high_likelihood
        #else:
        #     print ("Something went wrong witht the hls")

        self.con_timer = 0
        self.con_start = con_start

        # change length of disease based on community health care access
        if self.community == 1:
            if model.community1_hca == True:
                self.con_end = model.con_end_short
            else :
                self.con_end = model.con_end_long

        elif self.community == 2:
            if model.community2_hca == True:
                self.con_end = model.con_end_short
            else :
                self.con_end = model.con_end_long

        elif self.community == 3:
            if model.community3_hca == True:
                self.con_end = model.con_end_short
            else :
                self.con_end = model.con_end_long

        elif self.community == 4:
            if model.community4_hca == True:
                self.con_end = model.con_end_short
            else :
                self.con_end = model.con_end_long
```

```
elif self.community == 5:
    if model.community5_hca == True:
        self.con_end = model.con_end_short
    else :
        self.con_end = model.con_end_long

elif self.community == 6:
    if model.community6_hca == True:
        self.con_end = model.con_end_short
    else :
        self.con_end = model.con_end_long

else :
    print ("Something went wrong with assinging the healthcare access")

self.immunity = False
tmp_rnd = random.random()
if self.community == 1:
    if (tmp_rnd  <= model.vaccinated_com1):
        self.immunity = True

elif self.community == 2:
    if (tmp_rnd  <= model.vaccinated_com2):
        self.immunity = True

elif self.community == 3:
    if (tmp_rnd  <= model.vaccinated_com3):
        self.immunity = True

elif self.community == 4:
    if (tmp_rnd  <= model.vaccinated_com4):
        self.immunity = True

elif self.community == 5:
    if (tmp_rnd  <= model.vaccinated_com5):
        self.immunity = True

elif self.community == 6:
    if (tmp_rnd  <= model.vaccinated_com6):
        self.immunity = True

else:
```

```
            print ("Something went wrong with the self.immunity")

        self.infected = False
        if unique_id==1:
            self.infected=True
        if unique_id==2:
            self.infected=True
        if unique_id==3:
            self.infected=True
        if unique_id==4:
            self.infected=True
        if unique_id==5:
            self.infected=True
        if unique_id==6:
            self.infected=True

    def my_age(self):
        # a = random.randint(6, 95)
        p = random.random()
        if p < 0.25:
            a = 12

        elif p < .88:
            a = 45

        else:
            a = 75
        return a

    def move(self):
        # Placement of the communities rows
##          community0_row=1
##          community1_row=2
##          community3_row=3
##          community4_row=4
##          community5_row=5
##          community6_row=6

        # Placement of the day place rows
        #schoolroom_row=7
        #workroom_row=8

        # Placement of the bus
```

```
        #bus_row=0

        # Beginning cycle - Night
        if (self.model.transportation_cycle == 0):
            self.exposure_time = 840

            # Assigns the human to its correct community
            new_position = (self.community, self.household)
##              if (self.community == 1):
##                  new_position = (community0_row, self.household)
##              elif (self.community == 2):
##                  new_position = (community1_row, self.household)
##              elif (self.community == 3):
##                  new_position = (community3_row, self.household)
##              elif (self.community == 4):
##                  new_position = (community4_row, self.household)
##              elif (self.community == 5):
##                  new_position = (community5_row, self.household)
##              elif (self.community == 6):
##                  new_position = (community6_row, self.household)
##              else :
##                  print ('Incorrect assigned community')

        # Morning Commute
        elif (self.model.transportation_cycle == 1):
            self.exposure_time = 60
            new_position = (0, self.bus_number) #bus_row

        # Day
        elif (self.model.transportation_cycle == 2):
            self.exposure_time = 560
            # This if-statement controls where the ages will go during the 'day'
            if (self.age_bracket == self.bracket_adult): # Ages go to work
                # workroom_row - the work environment
                # self.workroom - the room in the work environment
                new_position = (8, self.workroom) #workroom_row
            else : # Kids go to school
                new_position = (7, self.schoolroom) #schoolroom_row

        # Afternoon Commute
        elif (self.model.transportation_cycle == 3):
            self.exposure_time = 60
            new_position = (0, self.bus_number) #bus_row
```

```
        # Error Message
        else :
            print ("Something went wrong with the night/day cyle")

        self.model.grid.move_agent(self, new_position)

##      def infect_others_old(self):
##          cellmates = self.model.grid.get_cell_list_contents([self.pos])
##          if len(cellmates) > 1:
##              for other in cellmates:
##                  if other.immunity is False:
##                      if other.infected is False:
##                          if(random.random() <= self.likelihood):
##                              other.infected = True

    def infect_others(self):
        cellmates = self.model.grid.get_cell_list_contents([self.pos])
        if len(cellmates) > 1:
            n = len(cellmates)
            for other in cellmates:
                if other.immunity is False:
                    if other.infected is False:
                        if(random.random() <= (1-(exp(-(self.ic[self.age_bracket][other.age_bracket] *
self.exposure_time))))):
                            other.infected = True

    def step(self):
        self.move()
        if self.infected:
            self.con_timer +=1
            if (( self.con_timer >= self.con_start) and (self.con_timer <= self.con_end)):
                        self.infect_others()
            else:
               if ( self.con_timer > self.con_end):
                    self.immunity = False
                    if self.community == 1 :
                        if self.model.community1_hca == True:
                            self.immunity = True
                    elif self.community == 2:
                        if self.model.community2_hca == True:
                            self.immunity = True
                    elif self.community == 3:
```

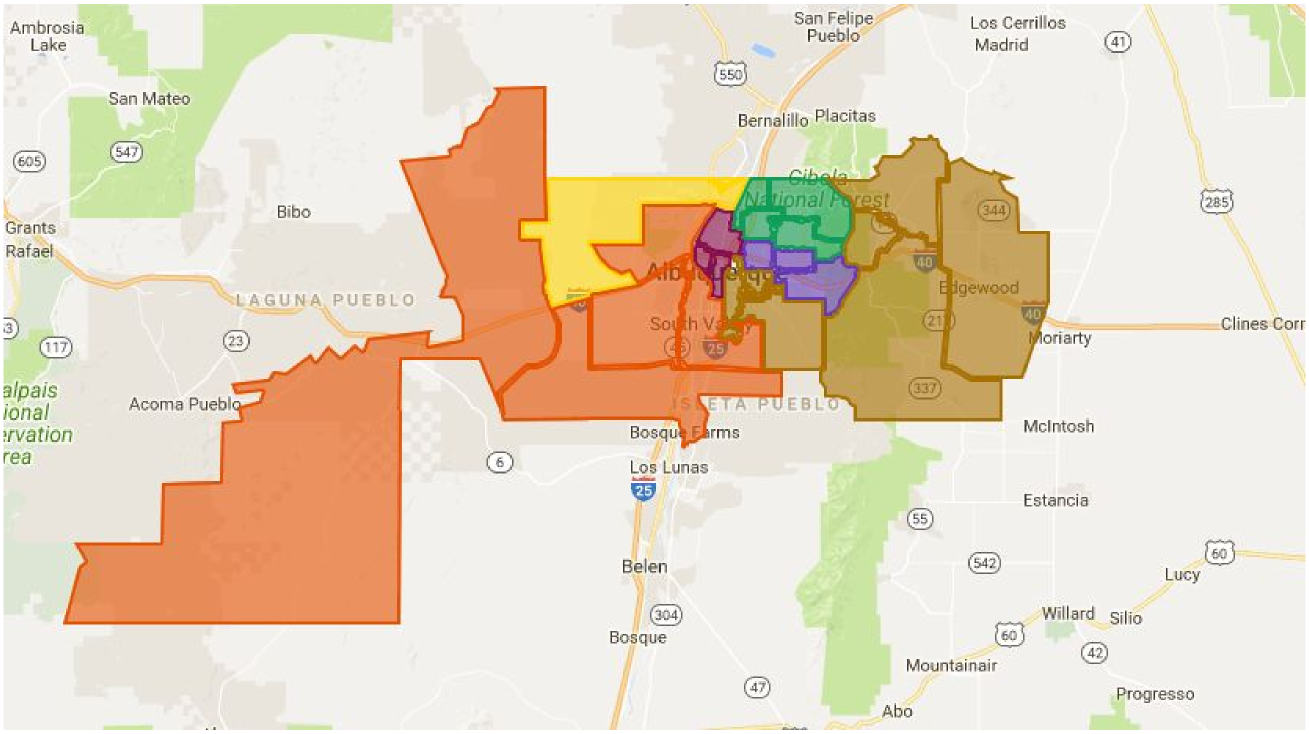
```
            if self.model.community3_hca == True:
                self.immunity = True
        elif self.community == 4:
            if self.model.community4_hca == True:
                self.immunity = True
        elif self.community == 5:
            if self.model.community5_hca == True:
                self.immunity = True
        elif self.community == 6:
            if self.model.community6_hca == True:
                self.immunity = True
        else:
            print ("Incorrect Community ID")

        self.infected = False
```

IDLE Commands

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.



import sys
>>> sys.path.append('E:/Users/Ben/Epi/')
>>> from Epi_Model_4 import *
>>> m1=InfectModel (50000, 100, .1, .2, .3, .4, .5, .6, .4, .5, 6, 14, 20, False, False, False, 
False, False, False, .1, .2, .3, .4, .5, .6)
>>> m1.run_steps(25)
```

Appendix

A[1]



Color	Red	Purple	Brown	Green	Yellow	Blue
District	One	Two	Three	Four	Five	Six

*Created by cross referencing Google Maps with the map of the Albuquerque Districts

B[7][8]

	District One	**District Two**	**District Three**	**District Four**	**District Five**	**District Six**	**Total**
Total Population	191,899	59,909	81,407	124,636	71,943	116,742	586,627
Margin of Error	+/- 3,198	+/- 1,704	+/- 2,238	+/-2,755	+/-1897	+/- 2,469	+/- 5,952
% of Total Population	32.71%	10.21%	13.88%	21.25%	12.26%	19.90%	100.00%
Vaccination Rate[8]	37.06%	35.38%	27.20%	47.03%	45.47%	38.73%	38.48%
Number of Houses	71,493	27,351	41,668	56092	27,690	52,841	249,784
Margin of Error	+/-1,363	+/-518	+/-742	+/-673	+/-458	+/-630	+/-1,933
Pop. Ages 5-19	48,072	11,047	12,949	22,278	17,331	20,291	120,291
Margin of Error	+/- 1,659	+/-850	+/-971	+/-1,073	+/-458	+/-1,091	+/-2,640
Per Room	0.7	0.4	0.3	0.4	0.6	0.4	0.47
Pop. Ages 20-64	123,213	38,564	59,727	78,661	45,845	75,147	382,5993
Margin of Error	+/-2,591	+/-1,358	+/-1,901	+/-2,030	+/-1,560	+/-2,116	+/-4,817
Per Room	1.7	1.4	1.4	1.4	1.7	1.4	1.50

---

[8] Vaccination Record HHS

Pop. Ages 65+	20,614	10,298	8,731	23,697	8,767	21,304	83,113
Margin of Error	+/-870	+/-581	+/-673	+/-1,524	+/-483	+/-483	+/-2128
Per Room	0.3	0.4	0.2	0.4	0.3	0.4	0.33

*Margin of Error was found by using the formula $\text{MOE}_{\text{Sum}} = \sqrt{(MOE_1)^2 + (MOE_2)^2 ...}$[9]

**All data was taken from the American Community Survey (ACS), done by the Census Bureau. All data was taken from the American FactFinder Website, which is in association with the Census Bureau . All of the values are estimates of the population.

C1[4]

Table for values of λ_{ij}

x-axis is Age Group of Susceptible (Agent A)

y-axis is Age Group of Infectious (Agent B)

	Age 5-19	**Age 20-64**	**Age 65+**
Age 5-19	.00061	.00033	.00080
Age 20-64	.00053	.00032	.00080
Age 65+	.00054	.00029	.00102

---

[9] Using Data from ACS

*Should be noted that we removed the 0-4 population in each district due to it not having a purpose in our program

C2[4]

Table for d_a per age group Organized in day,night

x-axis is Age Group of Susceptible (Agent A)

y-axis is Age Group of Infectious (Agent B)

	Age 5-19	**Age 20-64**	**Age 65+**
Age 5-19	480, 120, 840	0, 120, 840	0, 120, 0
Age 20-64	0, 120, 840	480, 120, 840	0,120, 0
Age 65+	0, 120, 0	0, 120, 0	480, 120, 840

*Should be noted that we removed the 0-4 population in each district due to it not having a purpose in our program

**The number before the comma is the time that the Agent A of an age range spends with an Agent B of age during a 8 hour day period. The second number is the time that Agent A of an age range spends with an Agent B of age a on the bus The third number is the time that the Agent A of an age range spends with an Agent B of age a, during a 14 hour night period.

Sources:

[1]

City Of Albuquerque-Find Your Councilor , www.cabq.gov/council/find-your-councilor.

[2]

“WebMD.” *WebMD*, WebMD, www.webmd.com/cold-and-flu/flu-statistics. Accessed 2 Apr. 2017.

[3]

“Influenza (Flu).” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 1 Nov. 2017, www.cdc.gov/flu/fluvaxview/coverage-1516estimates.htm.

[4]

Haber, M. J., Shay, D. K., Davis, X. M., Patel, R., Jin, X., Weintraub, E....Thompson, W. W. (2007). Effectiveness of Interventions to Reduce Contact Rates during a Simulated Influenza Pandemic. *Emerging Infectious Diseases*, *13*(4), 581-589. https://dx.doi.org/10.3201/eid1304.060828.

[5]

“How Flu Spreads.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 12 Sept. 2013, www.cdc.gov/flu/about/disease/spread.htm. Accessed 2 Apr. 2017.

[6]

CDC GIS FluView. (n.d.). Retrieved April 04, 2018, from https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

[7]

Introductory Tutorial¶. (n.d.). Retrieved April 04, 2018, from http://mesa.readthedocs.io/en/latest/tutorials/intro_tutorial.html

[8]

HHS Office of the Secretary, & National Vaccine Program Office. (2017, November 15). Flu Vaccination Claims Rates by Geographic Area. Retrieved April 04, 2018, from https://www.hhs.gov/nvpo/about/resources/interactive-mapping-tool-tracking-flu-vaccinations/index.html

[9]

https://www.census.gov/content/dam/Census/programs-surveys/acs/guidance/training-presentations/20170419_MOE.pdf

Acknowledgements

A large thank you to the Los Alamos National Lab and the judges for making the Challenge possible. We had a great time and spent many hours learning new information. We would also like to thank the George Mason University for creating the mesa library for Python. We are very grateful for the time spent with our teacher and the mentor, Jennifer Hooten and John Thorp, respectively.