Infectivity Model

New Mexico Supercomputing Challenge

Final Report

April 5, 2017

Team Number: ATC-3

School Name: Academy For Technology and the Classics

Team Members

Ben Thorp

Teddy Gonzales

Alex Baten

Ben Sheffer

Teacher

Jennifer Hooten

Project Mentors

John Thorp

Executive Summary

Our project is a model of the spread of the Influenza Thread A between two hypothetical neighborhoods. By studying how individual conditions can affect the spread of a disease through two different neighborhoods, we hope to suggest factors that will minimise the spread of Influenza Thread A. We can utilize our data in order 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 (vaccines), and contagion time periods.

We created a computer model in Python, to demonstrate the spread of Influenza between two communities. The model simulates the movement of agents between day and night congregation points. Each run of the simulation takes a number of parameters including:

- The total number of humans
- The number of houses or rooms in each environment
- The percentage of people in communities 1 and 2 that are living healthy lifestyles
- A low or high likelihood to catch the disease (associated with healthy lifestyle)
- The times when the disease starts and stops being contagious in communities with or without health care
- Access to health care in communities 1 and 2
- Percentage of vaccinated humans in communities 1 and 2

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 the model can predict, on a local level, how the disease spreads through different communities in different conditions.

Problem Statement

While discussing potential ideas for our project, we came across Google Trends. In our Computer Science class we read an article that went over how Google Trends could be used as a database for influenza infection rates by viewing the search results for “flu symptoms”. 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, in the future we wish to increase its complexity to develop a program that makes more accurate predictions.

Solution Method

We plan to solve our problem by creating a model that will incorporate the given variables, along with a fairly basic agent-based Python/Mesa model in order to show the spread of the disease between the two different neighborhoods. We will compare how the different demographics will affect the severity of the illness in the two neighborhoods. 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 main problem with this is that the simulation needs to accurately predict what would happen in 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.

Verification and Validation 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. From these infections about 1-4% of people will die, mainly due to pre-existing health conditions. Due to these facts we decided to ignore the lethality of the disease, and focus on it's spread (contagion).

We also found the vaccine rates across the U.S. by age group, with roughly 49% of children 0.5-17 years old having gotten the vaccine, and a whopping 31.7% of adults 18-64 having received a vaccine. We can incorporate this into our program by giving 49% of children a smaller chance of being infected, and 31.7% of adults a smaller chance of being infected. Since vaccinations play a major role in fighting Influenza, we decided to incorporate vaccines into our model. We have also researched and incorporated when you can pass the disease onto others, and how long you are symptomatic. 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 still needs of some more research to further back it up and will be addressed in year two of the project.

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.

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.

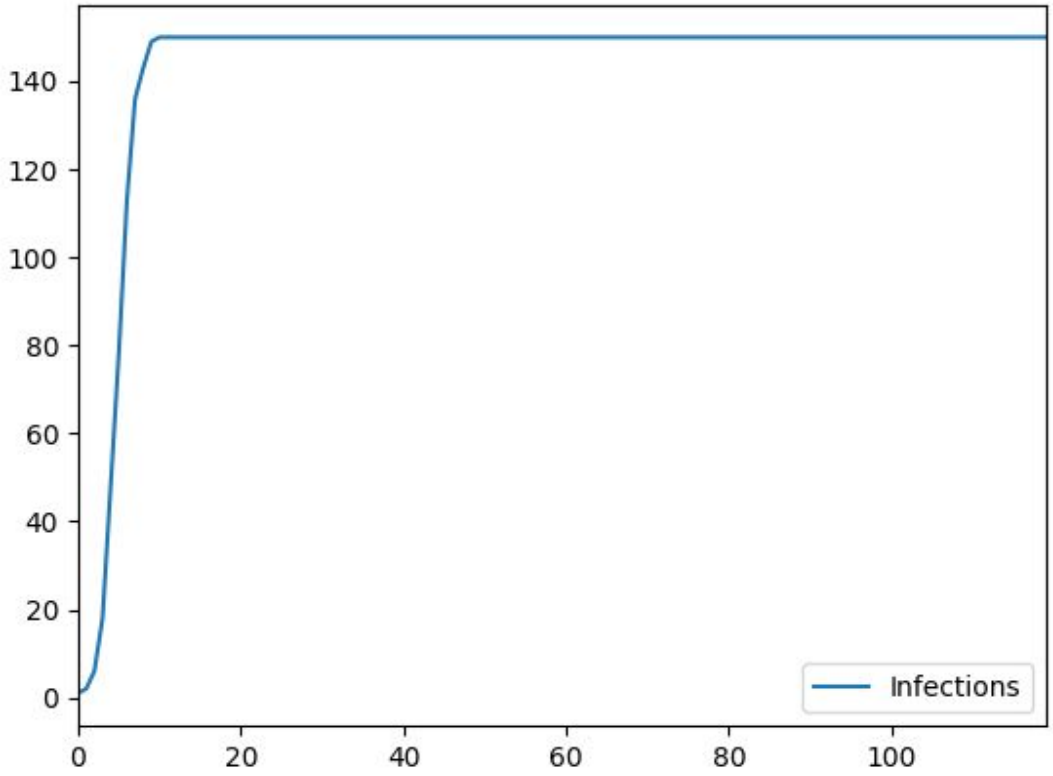


Figure 1: This is a trial that was carried out that involved 150 human agents and they infected others for ~60 days (day/night). There were 10 rooms per household, school, and work and the disease had a 50% (L=0.50) chance to infect others in the same room as them. Once the human agent is infected, they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

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.

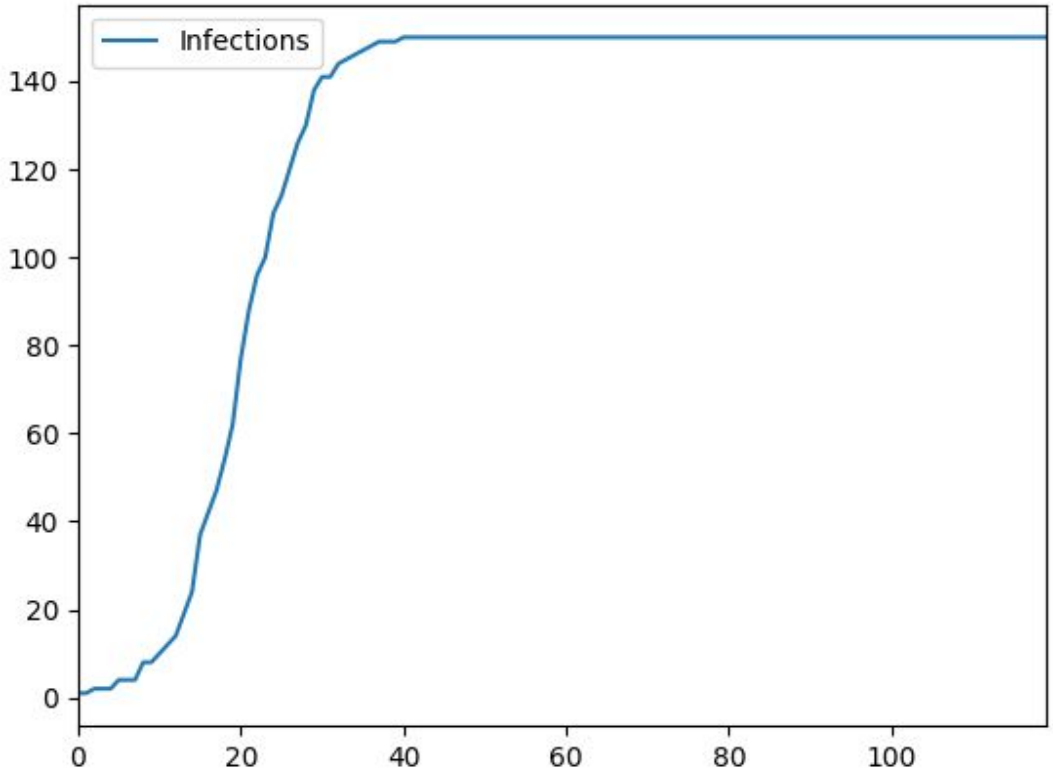


Figure 2: This is a trial that was carried out that involved 150 human agents and they infected others for ~60 days (day/night). There were 10 rooms per household, school, and work and the disease had a 10% (L=0.1) chance to infect others in the same room as them. Once the human agent is infected, they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

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.

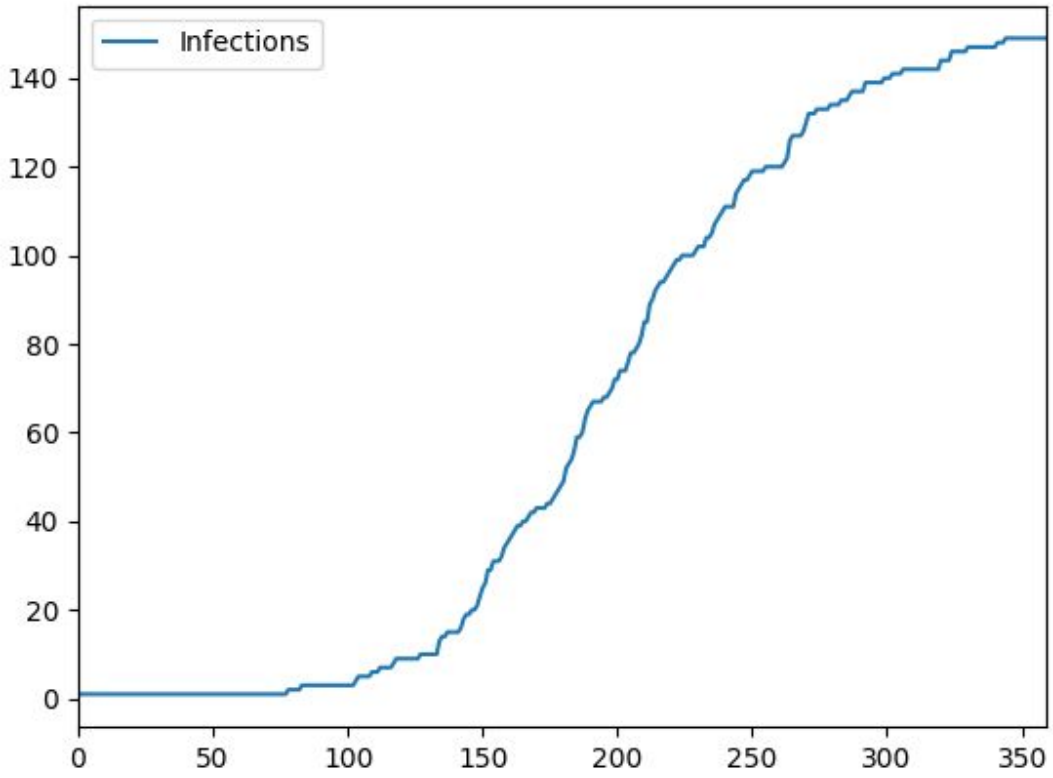


Figure 3: This is a trial that was carried out that involved 150 human agents and they infected others for ~180 days (day/night). There were 10 rooms per household, school, and work and the disease had a 1% (L=0.01) chance to infect others in the same room as them. Once the human agent is infected, they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

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.

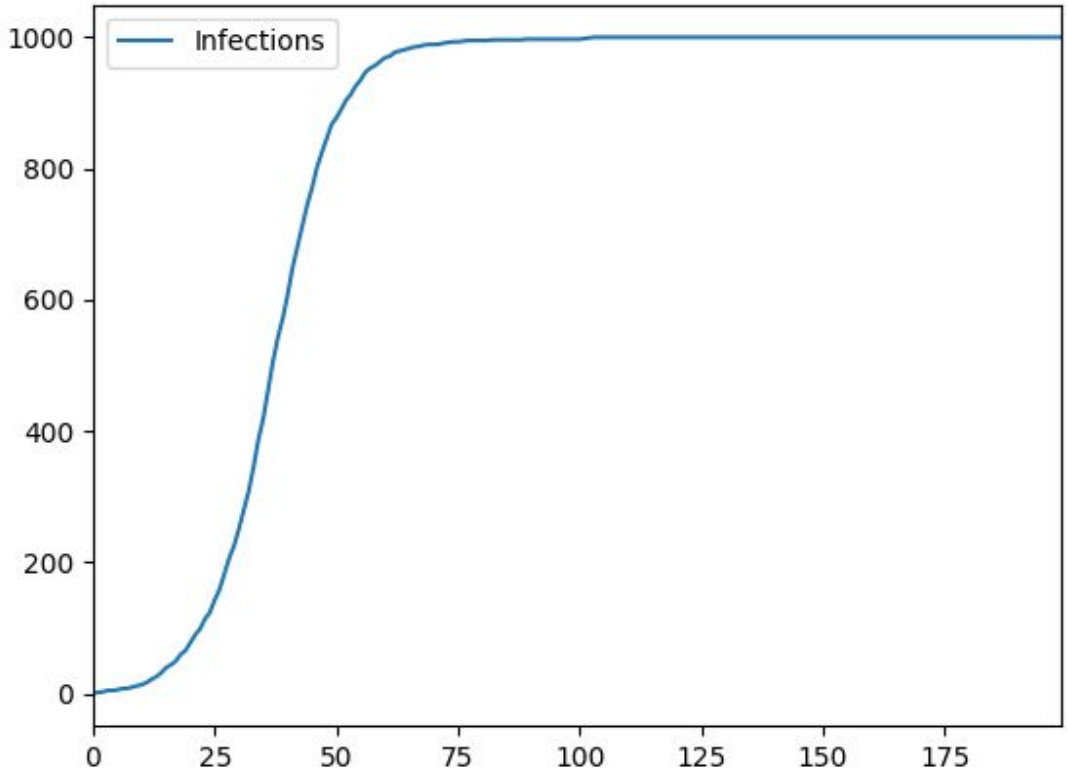


Figure 4: This is a trial that was carried out that involved 1000 human agents and they infected others for ~100 days (day/night). There were 100 rooms per household, school, and work and the disease had a 10% (L=0.1) chance to infect others in the same room as them. Once the human agent is infected,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

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.

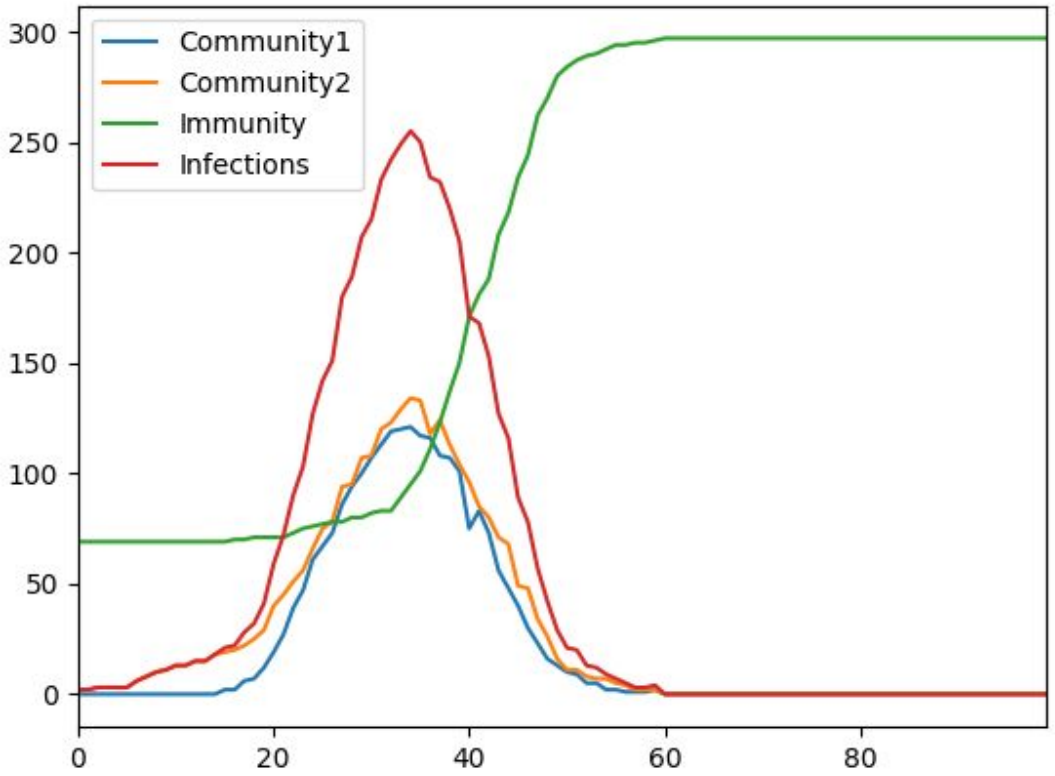


Figure 5: This is a trial that was carried out that involved 300 human agents and they infected others for ~30 days (day/night). There were 30 rooms per household, school, and work and the disease had a 20% (L=0.2) chance to infect others in the same room as them. The humans became contagious after day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

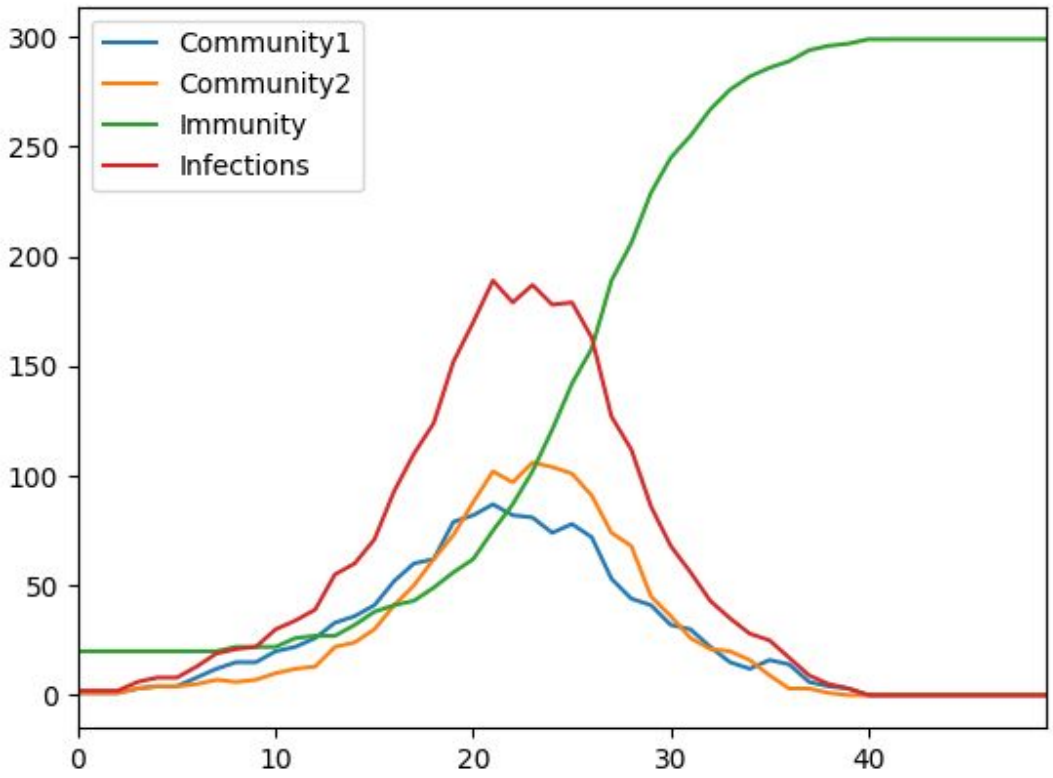


Figure 6: This is a trial that was carried out that involved 300 human agents and they infected others for ~30 days (day/night). There were 30 rooms per household, school, and work and the disease had a 40% (L=0.4) chance to infect others in the same room as them. The humans became contagious after day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

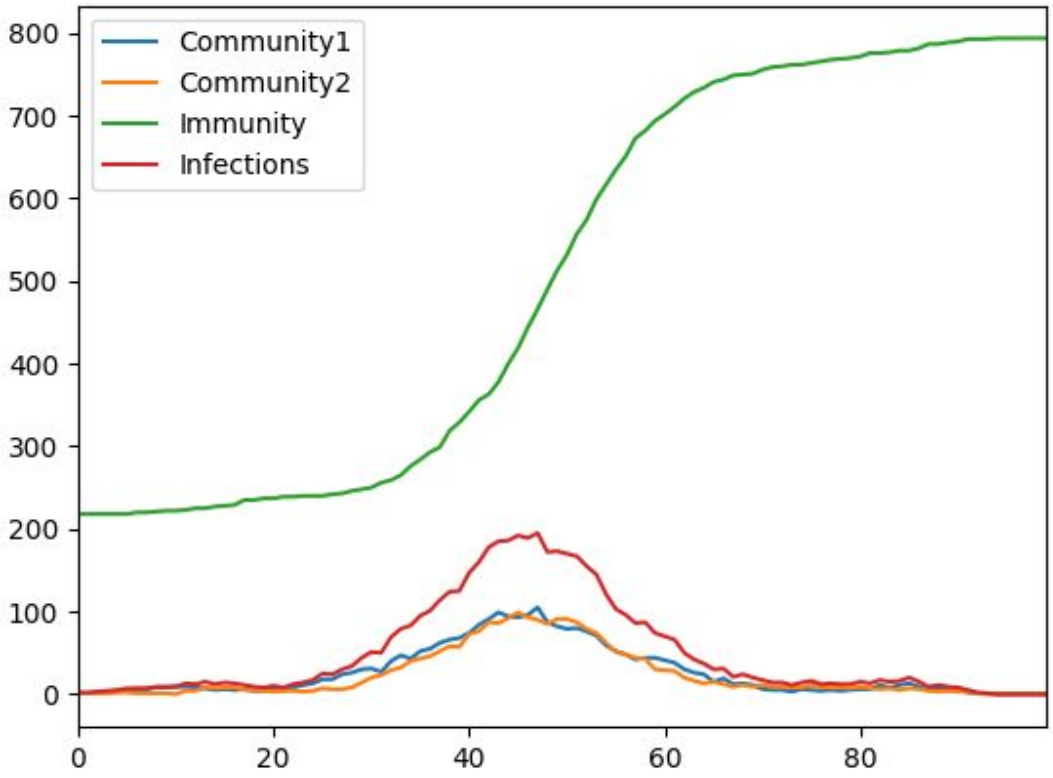


Figure 7: This is a trial that was carried out that involved 1000 human agents and they infected others for ~30 days (day/night). There were 100 rooms per household, school, and work and the disease had a 20% (L=0.2) chance to infect others in the same room as them. The humans became contagious after 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 hopsital or not. Also each community can have a different ratio of people living health lifestyles.

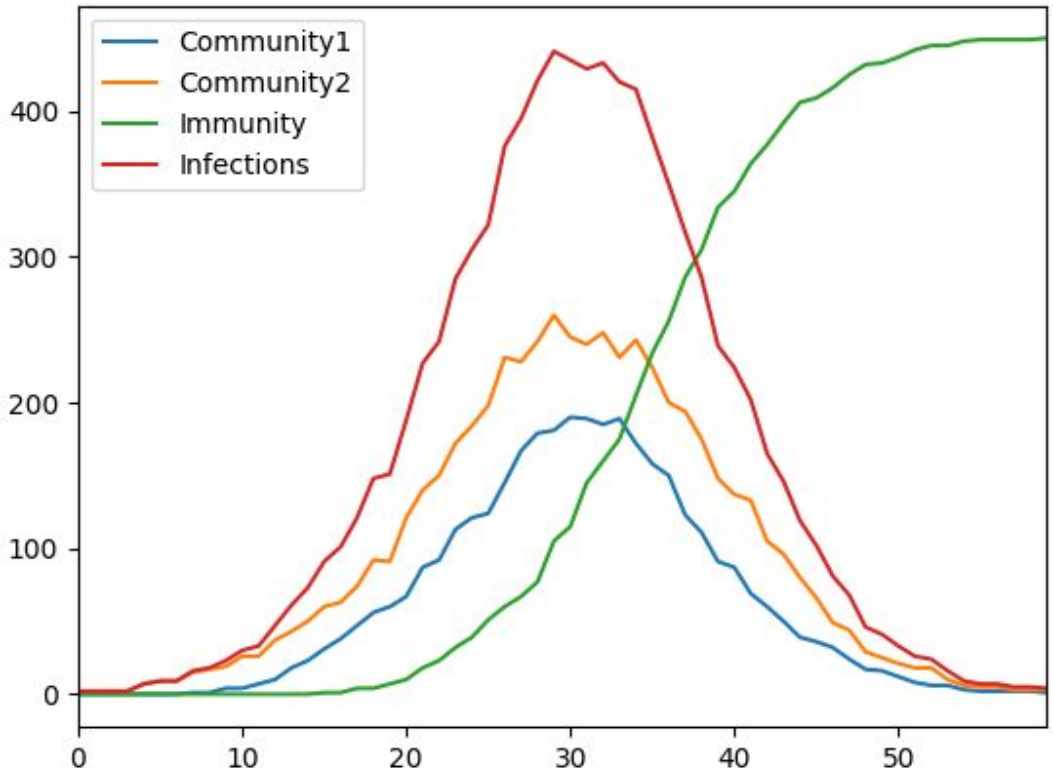


Figure 8: This is a trial that was carried out that involved 1000 human agents and they infected others for ~30 days (day/night). There were 200 rooms per household, school, and work and the disease had two likelihoods. The likelihood for healthier people to catch the disease is 20%. Comparatively the 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

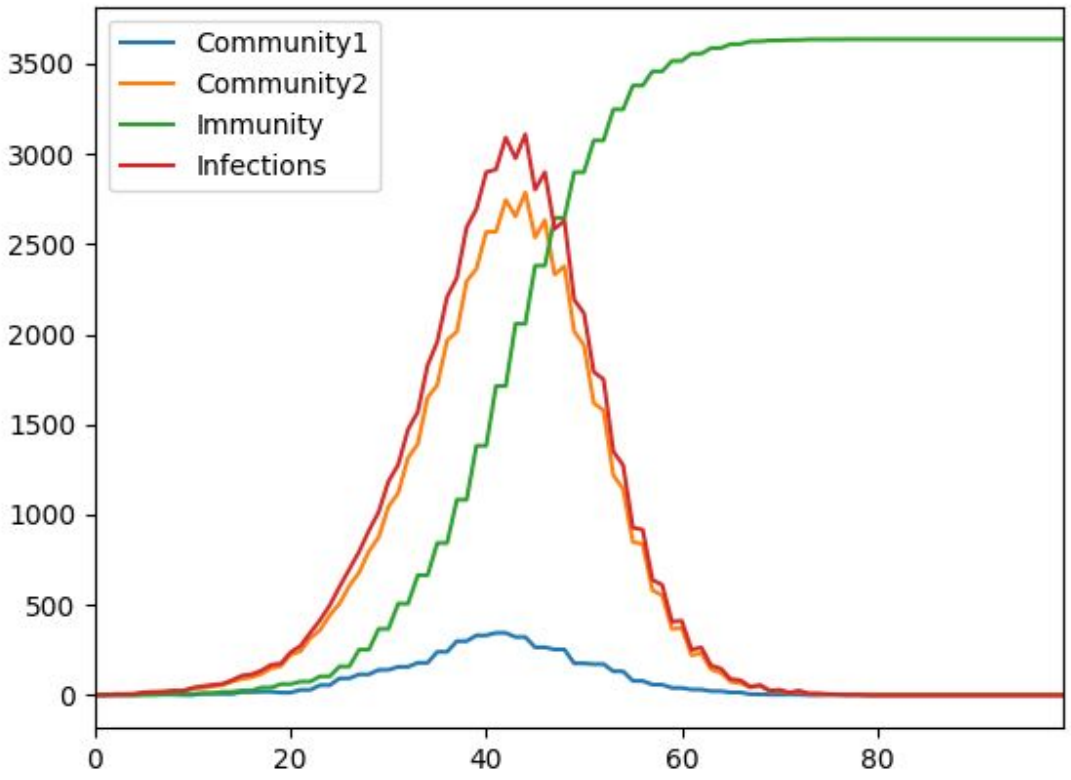


Figure 9: This is a trial that was carried out that involved 10000 human agents and they infected others for ~50 days (day/night). There were 1500 rooms per household, school, and work and the disease had two likelihoods. The likelihood for healthier people to catch the disease is 11%. Comparatively 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.

Conclusion and Achievements

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 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 "rooms density" and see if it does improve conditions (lower number of infections).

Going into the future, we wish to incorporate real world data from the past, change variables to predict a future outbreak, and see what would help the communities the most. As of now, our simulations only provide insight about how the variables affect

the number of infections. To create a reliable model that can accurately predict infection rates, we need to increase our simulation's complexity. Some variables that we may like to add in the future are; age, diet, additional communities, transportation, and greater complexity in the interactions of the "community agents".

Code

Selecting a language

We chose Python because we went on the Santa Fe Tech tour in Socorro and discovered that Python fit our needs. We are using Python 3.6 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

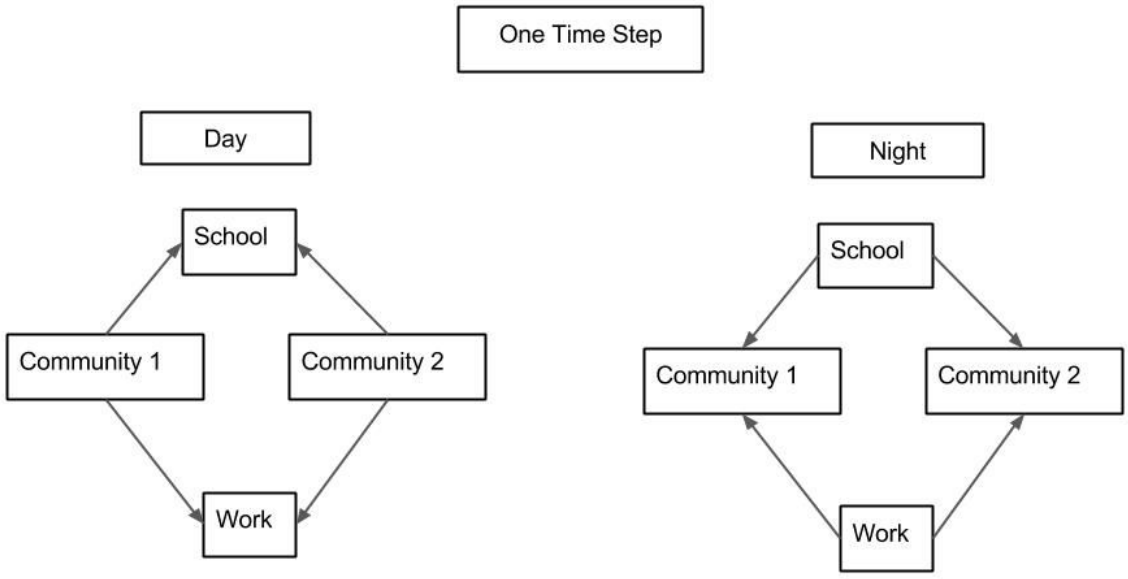
Our program requires us to make a model about the spread of disease. Writing models from scratch in Python is quite difficult, so we decided to look online for premade libraries to make the job easier. What we found was the Mesa framework.

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 a 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.

You can learn more about the Mesa framework at;
http://mesa.readthedocs.io/en/latest/tutorials/intro_tutorial.html

Day and Night for agents

In our model, humans travel between their nighttime communities and their daytime school or work. The communities are divided into households and infected human can infect others within their household. During the day, the school and work are divided into rooms, where inhabitants can infect each other. A time step is comprised of both a day and a night.



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.

Community 1	Community 2	School	Work
House 1	House 1	Classroom 1	Office 1
House 2	House 2	Classroom 2	Office 2
House 3	House 3	Classroom 3	Office 3
House 4	House 4	Classroom 4	Office 4
House 5	House 5	Classroom 5	Office 5

The source code

```
# Project 2017 ATC - Flu Transmission
# Built on Mesa Agent Library from George Mason Univ.
# Written by Ben Thorp
# ATC-3 Ben Thorp, Ben Sheffer, Alex Baten, Teddy Gonzales
# Version 1.1
#   Added start and stop of contagiousness
# Version 1.2
#   Added HealthCareAccess and HealthLifeStyle

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

# 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.community1 is True:
                    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.community1 is False ):
                    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, h,
             hls_com1, hls_com2, lowL, highL,
             con_start, con_end_short, con_end_long,
             com1_hca, com2_hca, vac_com1, vac_com2):
    # 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
    # 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
    # vac_com1 - percentage of vaccinated humans in community1
    # vac_com2 - percentage of vaccinated humans in community2
    self.num_agents = N
    self.grid = MultiGrid(4, h, True)
    self.schedule = RandomActivation(self)
    self.community1_hca = com1_hca
    self.community2_hca = com2_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.healthy_lifestyle_com1 = hls_com1
    self.healthy_lifestyle_com2 = hls_com2
    self.low_likelyhood = lowL
    self.high_likelyhood = highL

    # Create N humans for the model
    for i in range(self.num_agents):
        a = Human(i, self, con_start)
        self.schedule.add(a)
```

```
        # Add the agent to a random grid cell
        if (a.community1 is True):
            x = 0
        else:
            x = 1
        y = random.randrange(self.grid.height)
        self.grid.place_agent(a, (x, y))

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

    # Initialize the software that collects the data each timestep
    self.datacollector = DataCollector(
        model_reporters={"Community2": compute_infections_c2,
                         "Community1": compute_infections_c1,
                         "Immunity": compute_immunity,
                         "Infections": compute_infections,
                         }
        )

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

    self.schedule.step()
    self.timestep+=1
    if (self.timestep % 2 == 0 ):
        # Day time
        self.day=True
    else:
        # Night time
        self.day=False

# run_steps steps the model forword for "steps" (days/nights)
def run_steps(self, steps):
    for i in range(steps*2):
        # Doubling steps makes correct number of day/night 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
        self.community1 = random.choice([True, False])
        self.adult=random.choice([True, False])
        self.household=random.randrange(model.grid.height)
        self.schoolroom=random.randrange(model.grid.height)
        self.workroom=random.randrange(model.grid.height)

        # Initialize disease variables
        # self.likelihood = likelihood
        if self.community1 == True:
            if (random.random() <= model.healthy_lifestyle_com1):
                self.likelihood = model.low_likelyhood
            else:
                self.likelihood = model.high_likelyhood
        else:
            if (random.random() <= model.healthy_lifestyle_com2):
                self.likelihood = model.low_likelyhood
            else:
                self.likelihood = model.high_likelyhood

        self.con_timer = 0
        self.con_start = con_start
```

```
    # change length of disease based on community health care access
    if self.community1 == True:
        if model.community1_hca == True:
            self.con_end = model.con_end_short
        else :
            self.con_end = model.con_end_long
    else :
        if model.community2_hca == True:
            self.con_end = model.con_end_short
        else :
            self.con_end = model.con_end_long

    self.immunity = False
    if self.community1 == True:
        if (random.random() <= model.vaccinated_com1):
            self.immunity = True

    else:
         if (random.random() <= model.vaccinated_com2):
            self.immunity = True

    self.infected = False
    if unique_id==1:
        self.infected=True
    if unique_id==2:
        self.infected=True

def move(self):
    community0_row=0
    community1_row=1
    schoolroom_row=2
    workroom_row=3
    if (self.model.day): # Day
        if (self.adult): # Adults go to work
            # workroom_row - the work environment
            # self.workroom - the room in the work environment
            new_position = (workroom_row, self.workroom)
        else : # Kids go to school
```

```
            new_position = (schoolroom_row, self.schoolroom)
    else : # Night
        if (self.community1): # Assigns the human to its correct community
            new_position = (community0_row, self.household)
        else :
            new_position = (community1_row, self.household)
    self.model.grid.move_agent(self, new_position)

def infect_others(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 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.community1 == True :
                    if self.model.community1_hca == True:
                        self.immunity = True
                else:
                    if self.model.community2_hca == True:
                        self.immunity = True

                self.infected = False
```

Sources:

“Read ‘Understanding the Changing Planet: Strategic Directions for the Geographical Sciences’ at NAP.edu.” *National Academies Press: OpenBook*, www.nap.edu/read/12860/chapter/11. Accessed 2 Apr. 2017.

Weinstein, Robert A., et al. “Transmission of Influenza: Implications for Control in Health Care Settings.” *Clinical Infectious Diseases*, Oxford University Press, 15 Oct. 2003, academic.oup.com/cid/article-lookup/doi/10.1086/378292. Accessed 2 Apr. 2017.

“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.

“ILINet State Activity Indicator Map.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, gis.cdc.gov/grasp/fluview/main.html. Accessed 2 Apr. 2017.

"Influenza." Centers for Disease Control and Prevention. Centers for Disease Control and Prevention, 06 Oct. 2016. Web. 22 Dec. 2016

"Key Facts About Influenza (Flu)." Centers for Disease Control and Prevention. Centers for Disease Control and Prevention, 25 Aug. 2016. Web. 22 Dec. 2016.

"Estimating Seasonal Influenza-Associated Deaths in the United States." Centers for Disease Control and Prevention. Centers for Disease Control and Prevention, 09 Dec. 2016. Web. 22 Dec. 2016.

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

“National Center for Health Statistics.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 8 Dec. 2016, www.cdc.gov/nchs/products/databriefs/db267.htm. Accessed 2 Apr. 2017.

“Influenza (Flu).” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 31 Mar. 2017, www.cdc.gov/flu/weekly/usmap.htm. Accessed 2 Apr. 2017.

Kiersz, Andy. “Here's A Block-By-Block Look At Who's Making How Much Across NYC's 5 Boroughs.” *Business Insider*, Business Insider, 11 Dec. 2014, www.businessinsider.com/new-york-city-income-maps-2014-12. Accessed 2 Apr. 2017.

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. Also we would like to thank The George Mason University for creating the mesa library for Python.

We are also very grateful for the time spent with the teacher and the mentor, Jennifer Hooten and John Thorp.