

**Coronavirus's Countless Outcomes:
Modelling Ordinances and Adherences**

Submitted by: Brendan Sakosits and Ross Bollinger

Supervised by: Dr. Mehdi Owrang

Presented to

Department of Computer Science

In Fulfillment of Requirements

CSC-493 Computer Science Capstone Project

American University

December 5, Fall 2020 Term

Table of Contents:

Abstract	3
Introduction	4
Review	5
Design Requirements/Details of Project	7
Final Implementation	12
Conclusions	24
References	26

Abstract

Government ordinances resulting from the COVID-19 pandemic have brought to bear the public's inconsistent understanding of epidemiology. As a result, many have come to question the efficacy of those ordinances, i.e: quarantining, mask-wearing, and social distancing, which poses a tremendous health risk to the population. This project attempts to solve this problem by modeling the behaviors of contagion spread resulting from these changing ordinances, and providing an accessible and dynamic web application to users interested in developing their intuition for said behaviors.

Guided by relevant literature in agent based contagion modelling and epidemiological modelling generally, we designed and constructed an agent-based SIR model of the spread of COVID-19, which illustrates the effects of varying government ordinances on contagion spread in a local context. We then created a web application that accesses the model and visualises the simulation and its relevant data, with several user inputs that directly contribute to ordinance adherence throughout the simulation.

To evaluate the effectiveness of the model, we ran three separate tests with various alterations to those inputs to see if the model behaves as the literature would suggest. Not only did the model behave as expected, but exceeded expectations in its ability to model complex phenomena, having demonstrated the impact of reactive quarantine mandates on the potential for returning waves of infection.

The epidemiological conclusion is that this agent based SIR model of contagion spread supports the literature's conclusions that government mandates for ordinance adherence slows the spread of contagion, and that the timing of government cessation of mandates for quarantining must be decided on carefully so as to prevent repeated waves of infection. We hope that, through experimentation, users can come to the same conclusions and develop corresponding intuitions that can then educate their future judgements of policy.

Introduction

The population is currently in the midst of the global COVID-19 pandemic, and one of the guidelines for stopping the spread of the virus is the use of face masks. Countries around the world have issued ordinances to (among other actions) wear facial masks when in public areas, with the U.S. being no exception. However, some controversy has come up about the usefulness of ordinances such as these, with statements like “not-that-bad” or “not-that-high” circulating about the death toll and infection rate (Bump 2020).

The CDC has cited several studies that strongly conclude the efficacy of mask-wearing (Leung et al, Ma et al. 2020), so that is not what we are examining. However, the necessity of *mandates* for these ordinances is up for debate in America. Accessibly demonstrating the outcomes of different ordinances and rates of adherence can clear the air and standardize discussion on the subject, enabling more pragmatic personal explorations of the numbers.

For instance, a common point brought up against mandated adherence to mask ordinances is that “virus immunity is obtained by people getting sick and developing antibodies. Masking has made people fear one another. Did we mask for the flu? There is no cure for that either” (Sylte). If that citizen had an accessible web application where they could ask that question and see a prediction -- without having to read scientific jargon or the opinions of pundits -- that would lead to more level-headed discourse. Other questions might be, “If mask adherence were doubled, how would that affect the case count?” or “If nobody wore masks, how quickly would the contagion peak?”

We wish to implement such an accessible web app contagion model, with input fields to input parameters, and options for default parameters based on real data, allowing users to run a model of the coronavirus with whatever hypotheses they may have. This enables users to creatively explore the effects of different ordinances and rates of adherence, and make their own policy judgements from it. In this paper, we will be discussing the development of the model and the web app, the application’s accuracy, uses, and limitations, and how our model differs from others in the field. The policy conclusions of our model will also be explored in the Results section, as well as how those conclusions fit into accepted epidemiological wisdom and public discourse.

Review

When seeking to model behavior in reaction to certain ordinances within an epidemiological phenomena, the primary problems that arise are “*How will we model this epidemiological phenomena?*” and “*How will we model the impacts of different ordinances?*”

The most common and fundamental contagion model one sees in epidemiological literature is the S.I.R model, or Susceptible, Infected, and Recovered model (Froese 2020). Models of this form (some with extra or less considerations, such as “Vaccinated” or “Exposed” as in the SIVS and SEIR models) use generalized variables representing total populations, which change according to observed derivatives over a specified number of iterations to predict the growth, peaking, stalling, or demise of contagions over time (Froese 2020). The simplicity of the SIR-type model makes it perfect for predicting contagion behavior at a macro level, such as the global or regional; however, it does not predict local contagion behavior, such as at the city or county level, and does not adjust for the existence of or changes in government ordinances regarding the contagion (Abou-Ismaïl 2020). Only by extending this base model can we examine the local behavior of the Coronavirus, and more importantly: the change in local behavior due to counter-contagion ordinances.

One approach used by Shilei Zhao and Hua Chen in modelling the outbreak in China was to add a consideration for “Unquarantined” and “Quarantined” populations in the model, rather than simply an “infected” population, thereby extending the middle portion of the SIR model to measure the effects of required quarantine ordinances on contagion spread (Zhao 2020). Although their model produced significant findings, we decided not to use this kind of method as it is incapable of measuring the effects of other ordinances such as masking and distancing.

Another approach, and the one we used, is to extend the model by making it an Agent Based Model (ABM). Unlike the standard SIR models, which see the populations of infected, recovered, etc as numbers with rates of change applied to them, ABMs create independent instances for each unit (e.g: a household or individual) in the model and run a set of simple rules that govern the interactions of these agents, in emulation of their real life counterparts. The aim is to see larger regularities emerge through collective processes (Cuevas 2020, Hoertel et al 2020). This enables the model-maker to create attributes for the population that vary with each instance of that population within the model, representing population diversities such as varied mask and distancing ordinance adherence rates.

The ABM model we took the most inspiration from in applying this schema was Uri Wilensky’s 1997 HIV model in NetLogo. His model takes in the population's tendency to practice abstinence, the population's tendency to use condoms, and the population's tendency to get tested for HIV, among other factors. It then generates an array of agents (in this case, individuals) fitting those specific demographics and applies a set of simple rules related to STD transmission to that array. The emergent behavior then reveals the behavior of the impacts of

individual tendencies (such as condom usage) upon the macro issue of HIV spread. We are applying this same schema (SIR - ABM) to the spread of Coronavirus.

ABM expansions of SIR-type models have been done on Coronavirus before. In the aforementioned article by Erik Cuevas, the author uses his ABM SIR extension to model facility-specific microcosms of Coronavirus contagion spread, as in Hospitals and individual rooms with varying crowdedness and risk-level. Of primary weakness to his schema, regarding our purposes, is the lack of inclusion of a “recovered” or “dead” category due to the very time-limited nature of his model, as well as that time-limited nature itself (Cuevas 2020).

To properly examine the behavior of contagion spread in light of changing government ordinances for quarantine, mask-wearing, and social distancing, we need to synthesize and expand upon these already robust and innovative models.

Regarding default values for parameters used by the model, we also executed a literature review of scholarly research. These are the default values we arrived at, with the relevant sources sourced:

- Probability that an individual wears masks despite no mandate: 59% (Katz 2020)
- Probability that an individual social distances despite no mandate: 64% (Saad 2020)
- Probability of infection for an interaction between individuals where the infected is wearing a mask, but not social distancing: 13% (Chu 2020)
- Probability of infection for an interaction between individuals where the infected is not wearing a mask, but is social distancing: 17% (Chu 2020)
- Probability of infection for an interaction between individuals where the infected is not wearing a mask nor social distancing: 19% (Chu 2020)
- Probability of infection for an interaction between individuals where the infected is both wearing a mask and social distancing: 3% (Chu 2020)
- Average incubation period for virus: 6 days (CDC 2020)
- Average amount of time in “t” units that it takes to recover from infection: 4

Design Requirements/Details of Project

As a general outline, this project consists of a model and an interface for using that model and examining its results. Therefore, this project consists of a backend: the model — and a frontend: the user inputs and visualizations / output graphs. This section is organized accordingly.

Backend Model Design

Framed simply, the backend design is a process of receiving the user inputs from the frontend → initializing arrays of the populations → updating the arrays a specified number of times → updating the graphs and visualizations → exporting these through flask to the front end.

Now that the general course is mapped out, here is a deeper dive into the design schema: Upon receiving the user inputs, they feed into an initialization function that creates the arrays of agents, as well as empty arrays of population counts and derivative data for use by the frontend for easy visualization to users. The array of agents is then fed a user-selected number of times into an “Update” function that asks all infected agents to run an “infect” function — which just calculates the chance of infection given two agent inputs — on all neighbors of the infected person + a randomized collection of other community agents, but only if the infected agent in question is not quarantining. After each iteration of this “Update” function is finished, the population count and derivative arrays are updated with the changes, and “t” is incremented. Finally, all infected agents are iterated through to be given a chance at recovery, turning them into “recovered” agents, rather than “infected” or “susceptible” agents, in accordance with the standard S.I.R framework. Once this is all done, the visualizations and graphs are created, a process that will be discussed separately on the following page as a portion of the front end.

User Input

We wanted to make sure there were as many ways for the user to affect the model and its outcome as possible, so as to make the model as dynamic as possible. In discussing what the user should be able to give the model that allow for more customized results, we landed on 10 different inputs:

- The total time that the model simulates. This variable asks for a number from the user that specifies how many days the model simulates its results from.
- Whether or not the population is in mandated quarantine at the start of the simulation. This is a boolean (shown to the user as a checkbox) that determines whether or not every person in the simulation is isolating themselves from the beginning of the simulation. Our goal with this input and the next one is to demonstrate how drastically the rate of spread is affected by quarantining.

- The percentage of sick people at which mandated quarantine comes into effect. The input takes the proportion of people who have to become infected before the mandated quarantine must take effect if the previous box is not checked.
- The percentage of people who isolate themselves after they realize they are infected. This input asks for a number that determines how many people decide to isolate themselves after they realize they have contracted COVID-19. This input allows the user to experiment with how responsible each person is with how they handle an infection.
- The percentage of people who are asymptomatic. This input is closely related to the amount of people who isolate after realization of infection. It asks for a number that determines how many people who, when infected, do not experience symptoms. If a person does not experience symptoms, they do not isolate because they do not realize they are infected.
- The percentage of people who consistently socially distance. This input takes a number that determines how many people follow social distancing guidelines through keeping at least 6 feet between themselves and others at all times.
- The percentage of people who consistently wear masks. This input takes a number that determines how many people who always follow the mask ordinances correctly. It should be noted that a person can be both socially distancing and wearing masks.
- The amount of time it takes to recover. This input takes a number that determines how many days it takes for an infected person to recover and no longer be able to spread the virus.
- The percent chance of infection while wearing a mask. This input takes a number that determines the likeliness that someone gets infected whilst wearing a mask.
- The percent chance of infection while socially distancing. This input takes a number that determines the likeliness that someone gets infected whilst socially distancing.
- The percent chance of infection while wearing a mask *and* socially distancing. This input takes a number that determines the likeliness that someone gets infected whilst both wearing a mask and socially distancing.

The last three inputs were chosen to be made inputs due to the controversy surrounding the efficacy of ordinances. These three variables are what allow the user to truly test said efficacy. We decided to assign default values to all of these variables, assigning most default values based on CDC data (i.e. the last three inputs, as well as others). We decided on this final list of inputs because they are simple enough to implement into the model, and they allow for a very wide range of variability that users can experiment with.

Visualization

When conceptualizing a way to visualize the model that would show the progression of the spread of the virus firsthand, we played around with a couple of options. However, the priority was always to display each person and their specific state (susceptible, infected, or recovered) through the use of a color. Ultimately, we decided on using a scatter plot to display the model. Each point on the scatter plot would represent a person, and the three colors we decided to use for states are as follows: blue for susceptible, red for infected, and green for recovered. The ultimate goal for this visualization was to display multiple scatter plots of the same model at different time intervals. This would clearly demonstrate the progression of the simulation over time.

As for the other two plots (the total counts of states and the rates of change of those states), we simply needed two line plots. Each line plot has three lines that represent the three states. As such, each line's color would correspond to those states (blue, red, and green, as outlined above).

Feasibility Discussion

As shown in the literature review, there were a number of influential papers to our design. The use of Agent Based Models (ABMs) by authors in epidemiology (such as the aforementioned Wilensky, Cuevas, and Hoertel et al) demonstrated the usefulness of such a tactic and inspired our ABM approach to our problem. Wilensky in specific revealed how effective a simple model can be at illustrating behavior, compared to the complex predictive-use models by the other two authors. Other innovations in the model, such as the use of neighborhoods for infection, were made creatively in response to need, but do have precedents as coding principles elsewhere generally.

More importantly, a great deal of our parameters and results rely heavily on default values for infectivity, risk-factors, recovery times, etc — all of which required extensive research to pin down.

Economic:

Since we are using exclusively free software, the design and development costs of this project are nonexistent. The goal for this project is to have an accessible model for people to use online for free, so it would not cost anything to access or use. The project will also not be marketed, so no costs are of note in that area.

Sustainability:

There would be little to no issues occurring with product sustainability. The only thing that could come to mind is software updates or changing data, but because the project is entirely

online, there would be no difficulty in adapting to outside changes such as these. In fact, this project can easily be updated to include any factors surrounding the pandemic, with future iterations possessing high potential for even better accuracy.

Ethical:

The fact that this model is simulating a very contemporary and hot button issue could cause some ethical problems. People could try to use this application as a perfect simulation to predict the future of our pandemic, and some may try to treat outputs from this model generated from inputs that aren't accurate to real world data as supporting evidence for their own agenda. The best way to solve this problem would be to include some kind of warning that states that results of this model are not perfectly accurate and should not be used as solitary statistical evidence to support any argument. This application is primarily for experimentation and learning purposes.

Political:

The potential political misuse of this model runs in the same vein as the ethical concerns. Because of how contentious the issue of ordinances is (especially in America), the COVID-19 pandemic has become politicized. As such, some may try to use this application as a method of invalidating other models or predictions, when in fact, this model has tremendous limitations and should be viewed in constructive conversation with other epidemiological models, *not* against them.

Final Implementation

For the final implementation of our application, we used Python to program the model itself, python's matplotlib library for visualization, and finally python's Flask library for connecting the model to html code that is able to take inputs and display the images.

Backend Model

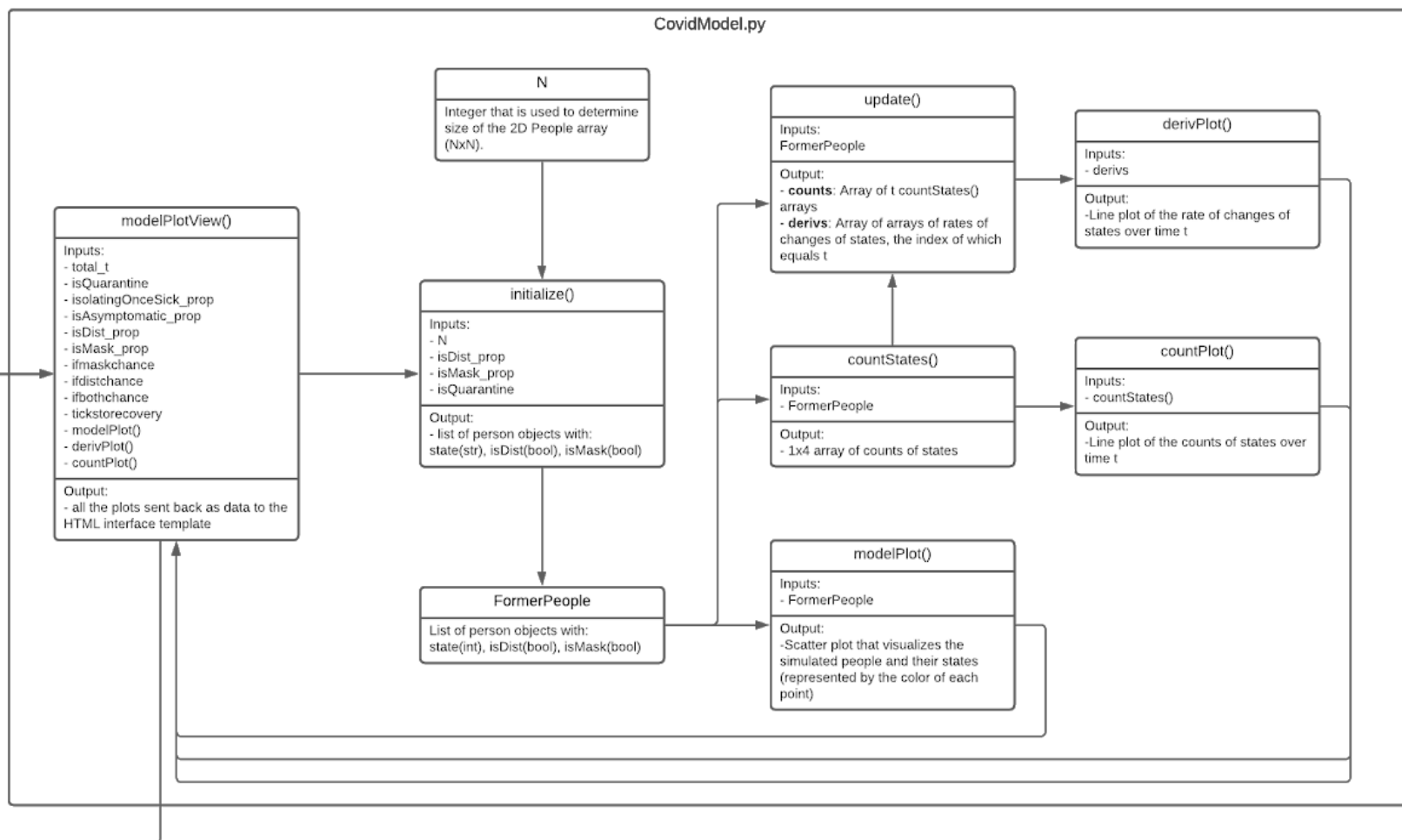


Fig 1. Python Code UML Section

As aforementioned, the general structure of the backend model is:

User inputs from the frontend → initializing the agent arrays → updating the arrays a specified number of times → updating counts[] and derivs[] → sending these to the frontend functions

To start, initialize is a simple function that creates a 2D array of “Susceptible” (state=2) agents, where the agents are an object called “Person” with the following properties:

- State (Susceptible, Infected, or Recovered, represented by 1, 2, and 3)
- isDist (boolean, whether the person is social distancing)
- isMask (boolean, whether the person is wearing masks)
- recoveryT (ticks since infection, initialized to zero)
- isAsymptomatic (boolean, whether the person is asymptomatic)
- isIsolating (boolean, whether the person is voluntarily quarantining, regardless of government mandate for quarantine)

Initialize then selects the middle agent to turn infected as case zero in the simulation, and places the first population count in the counts[] array to be used by the later graphing functions.

```
def initialize(N, isDist_prop, isMask_prop, isQuarantine):
    global t
    for i in range(40):
        for j in range(40):
            FormerPeople[i][j] = Person(1,
                random() <= isDist_prop,
                random() <= isMask_prop,
                0, random() <= isAsymptomatic_prop,
                isQuarantine)
        FormerPeople[20][20].state=2
    counts.append(countStates(FormerPeople))
    t += 1
```

After initialize is done, the update function (below) can take in an array of agents. It makes a copy of this array, to be edited before replacing that input array for the next iteration of update. A version of “x= new x” is used for this. Counts[] is again updated for use later in the model by the frontend functions; this is where it will be updated for the remainder of the model. Finally, in this first portion of “update”, the reactive government mandate for quarantine is checked against the population of infected to establish whether mandatory quarantine must be instituted.

```
def update(FormerPeople):
    global isQuarantine
    global t
    People=copy.deepcopy(FormerPeople)
    counts.append(countStates(People))
    if counts[t][1]/1600>=quarantineProp:
        isQuarantine=True
    else:
        isQuarantine=False
```

After the above image but before the below image, nested for-loops iterate over all agents in the model, and make the infected do the following:

- Updates the infected individuals recovery timeline, and has them recover if they reach the user-specified “ticksToRecovery” value. Default = 4, based on our literature review.
- Run “infect” — a function that calculates the risk of infection between two inputted agents and changes the latter to infected if the chance is below `random.random()` — on all 8 of its neighboring agents in the 2D array, simulating household spread of contagion.
- If the infected individual in question is not voluntarily isolating, and mandatory quarantine is not in effect, the individual also infects between $1x$ and $3x$ ¹ as many individuals as the size of their household (8) anywhere in the array, representing community infection due to non quarantining individuals interacting in restaurants, schools, buses and other public spaces.

Finally, the last section of update, pictured below, simply updates the derivative array for use by the frontend graphing and visualization functions later in the model.

```
templist=[]
if t>0:
    for i in range(3):
        if counts[t-1][i]!=0:
            templist.append(counts[t][i]/counts[t-1][i])
        else:
            templist.append(0)
    derivs.append(templist)
    t+=1
    return(People)
```

¹ This value ($1x$ to $3x$) is the only deeply impactful arbitrary element of the simulation, as we could not find any reliable estimates of the relationship between local and communal infectivity rates for non quarantining individuals, and simply estimated based on what we thought was reasonable. Therefore, this model assumes that the average not-self-isolating individual, during a pandemic, interacts significantly with 1 to $3x$ as many people in their greater community as in their extended household bubble.

Web Application

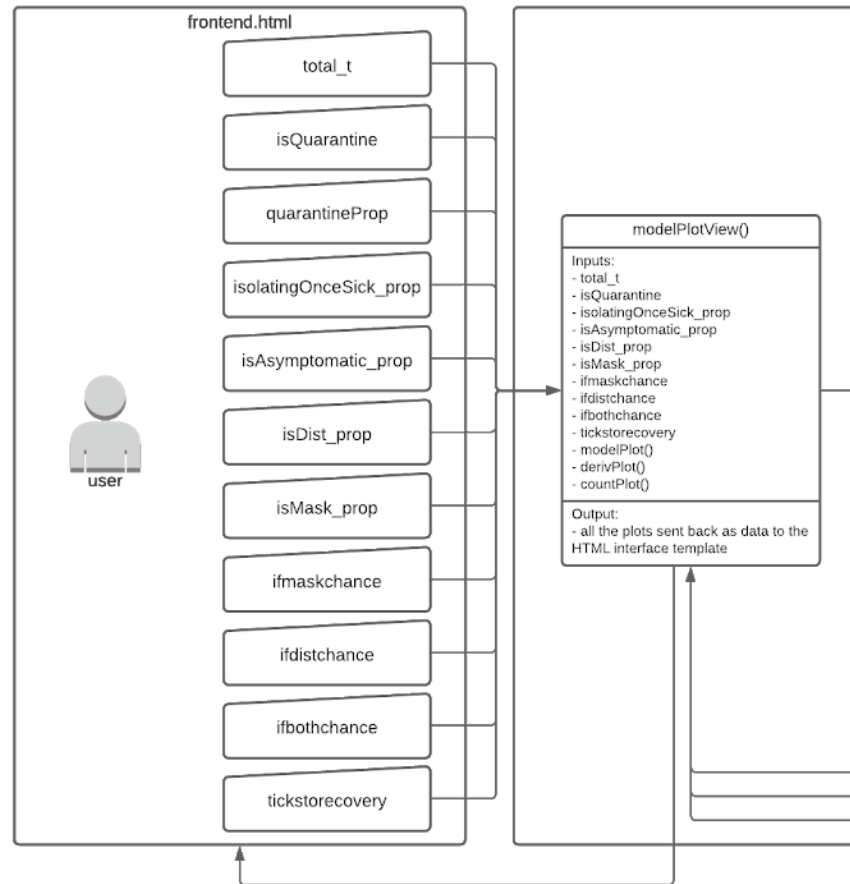


Fig 2. Web Application UML Section

The web interface is a simple html template titled “frontend.html”. It takes a set of inputs mentioned in the Design Requirements section (`total_t`, `isQuarantine`, `quarantineProp`, `isolatingOnceSick_prop`, `isAsymptomatic_prop`, `isDist_prop`, `isMask_prop`, `ifmaskchance`, `ifdistchance`, `ifbothchance`, and `tickstorecovery`). These inputs are then assigned to defined variables in the Python code for use in simulation. This fetching of inputs and assignments is all done by a function called `modelPlotView` in the Python code. This code then sends those new variables into the rest of the code so simulation can actually begin. Once the simulation is done, `modelPlotView()` passes the graphs generated by the model back to the web interface to be displayed by the html code. This is all shown in Fig 2.

```

<!doctype html>
<html>
<head>
<title>Covid Model Web App</title>
</head>
<body>
  <form method="POST" action="/input">
    Total time ticks (1 tick = 6 days): <input type="number" name="timeticks" value="30" required><br>
    Is population in mandated quarantine from the start? <input type="checkbox" name="inQuar"><br>
    At what percent of the population being sick does mandated quarantine come into effect? <input type="number" name="quarantineProp" value="50" required><br>
    Percentage of people voluntarily isolating upon realizing they are sick <input type="number" name="isoProp" value="100" required><br>
    Percentage of infected who are asymptomatic <input type="number" name="asymProp" value="40" required><br>
    Percentage of people who consistently social distance <input type="number" name="socDistProp" value="64" required><br>
    Percentage of people who consistently wear masks <input type="number" name="maskProp" value="59" required><br>
    Percent chance that someone gets infected without wearing a mask or social distancing <input type="number" name="neitherProp" value="19" required><br>
    Percent chance that someone gets infected while wearing a mask <input type="number" name="maskEncProp" value="13" required><br>
    Percent chance that someone gets infected while social distancing <input type="number" name="socDistEncProp" value="17" required><br>
    Percent chance that someone gets infected while both wearing a mask and social distancing <input type="number" name="bothProp" value="3" required><br>
    Time ticks it takes infected to recover <input type="number" name="recTime" value="4" required><br>
    <input type="submit" value="Submit">
  </form>
  
  <br>
  
  <br>
  
  
</body>
</html>

```

Total time ticks (1 tick = 6 days):

Is population in mandated quarantine from the start? ☐

At what percent of the population being sick does mandated quarantine come into effect?

Percentage of people voluntarily isolating upon realizing they are sick

Percentage of infected who are asymptomatic

Percentage of people who consistently social distance

Percentage of people who consistently wear masks

Percent chance that someone gets infected without wearing a mask or social distancing

Percent chance that someone gets infected while wearing a mask

Percent chance that someone gets infected while social distancing

Percent chance that someone gets infected while both wearing a mask and social distancing

Time ticks it takes infected to recover

Fig 3. Frontend.html Code and Output

Fig 3 displays the code written in frontend.html, as well as its output on the actual interface. This is all the code necessary for creating the entire front end of the application. As you can see, it creates a set of input forms, each with text coming before it that explains what the input's value does. Each input form also has a unique name, which is how the modelPlotView() function in Fig 2. gets the value of each input. At the bottom of the code, there are six img elements that have their source labeled as a placeholder variable labeled as "image#". These

variables are each a separate graph image, and they are assigned to a graph generated by the model through modelPlotView. Please take note of the default values shown in the input fields in the second image of Fig 3.

```
@app.route('/input', methods=['GET', 'POST'])
def modelPlotView():

    total_t = int(request.form.get("timeticks"))
    if request.form.get("inQuar"):
        isQuarantine = True
    quarantineProp = int(request.form.get("quarantineProp"))/100
    isolatingOnceSick_prop = int(request.form.get("isoProp"))/100
    isAsymptomatic_prop = int(request.form.get("asymProp"))/100
    isDist_prop = int(request.form.get("socDistProp"))/100
    isMask_prop = int(request.form.get("maskProp"))/100
    ifneitherchance = int(request.form.get("neitherProp"))/100
    ifmaskchance = int(request.form.get("maskEncProp"))/100
    ifdistchance = int(request.form.get("socDistEncProp"))/100
    ifbothchance = int(request.form.get("bothProp"))/100
    tickstorecovery = int(request.form.get("recTime"))

    return render_template('frontend.html',
                           image1=pngImage1B64String,
                           image2=pngImage2B64String,
                           image3=pngImage3B64String,
                           image4=pngImage4B64String,
                           image5=pngImage5B64String,
                           image6=pngImage6B64String)
```

Fig 4. modelPlotView() Applied to the Web Interface

Fig 4. shows the code sections in modelPlotView that apply to outputting onto the web interface. First, we define the route for the web app with the methods 'GET' and 'POST' so that we can actually retrieve the information from the template. Next, we set all the variables already defined in the code to the values inputted in the web interface. After that, the graphs are generated, then encoded into base64 strings. Said base64 strings are then fed back into the frontend template to be displayed on the web interface.

Results

Hypotheses

Concerning the three ordinances we simulated: quarantine, mask-wearing and social distancing, and comparing them to non-quarantine results, our predictions were as follows:

For the quarantine test, we expect the inclusion of a reactive quarantine mandate to dramatically flatten the “infected” population curve, thereby dramatically increasing the end-state total of uninfected. This is because a quarantine mandate dramatically reduces the number of infections per infected person per time unit. As well, we anticipate that the uninfected populations will be more clustered than in any other case, due to our model of quarantine dramatically shortening the distance of interactions (as in, in Quarantine, all interactions are limited to immediate neighbors).

For the mask-wearing and/or social distancing test, we expect the inclusion of a mask-wearing and/or social distancing mandate to effectually flatten the curve, similarly to the quarantine mandate, but with less effect than a quickly-enforced quarantine mandate (i.e: implemented at infected population $\leq 10\%$) with non-mandated masking and distancing. This is because Quarantine changes the entire ability of the virus to spread by limiting total interactions, whereas masking and distancing mandates simply alter the chance of infection per interaction.

For the non-quarantine test with default non-mandate masking and distancing (control), we anticipate a steep peak, immediately tapering as the infected recover, until the vast majority are recovered. This peak should be highest.

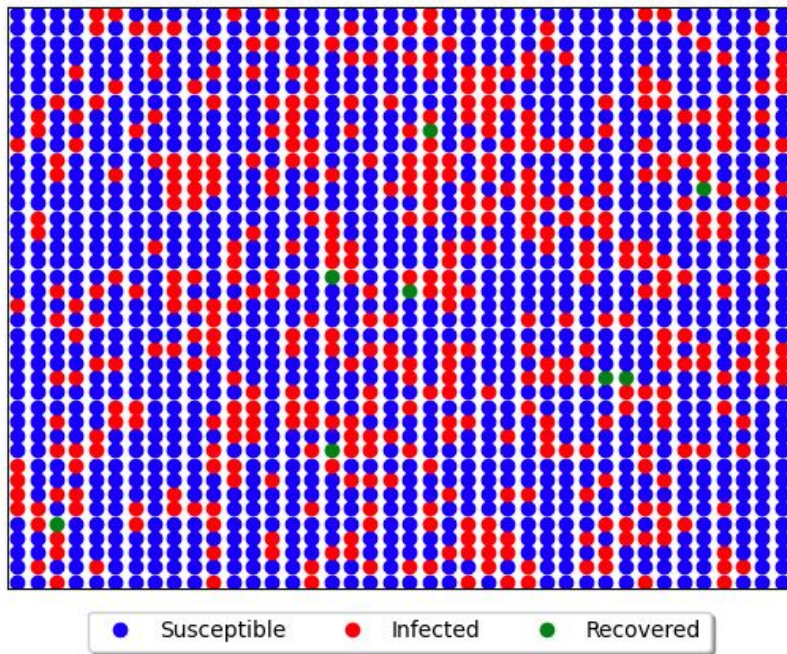
Essentially, if the model is correct and the epidemiological canon is correct as well, we anticipate the inclusion of mandates to decrease total end-case numbers by slowing contagion spread, in alignment with our literature review.

Experimentation

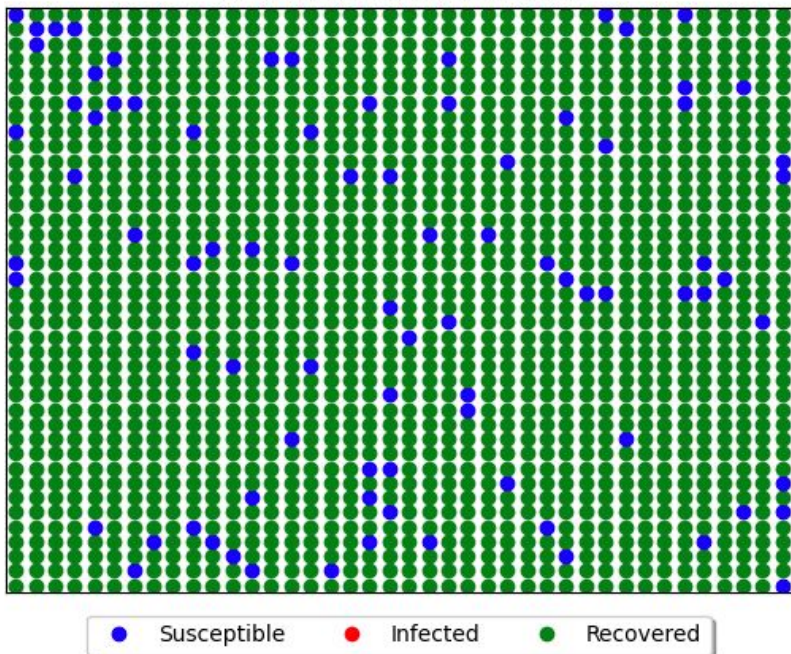
For the results, we decided to do four tests in which we altered only one of the inputs, but kept the rest set to their default values. Test 1 sets the “infected percent before quarantine” to 100 (meaning the government does not mandate quarantine whatsoever), Test 2 sets the “infected percent before quarantine” to 10 (meaning the government mandates quarantine at 10% of population infected, but ceases again when 90% of the population is uninfected), Test 3 is Test 1, but with the percentage of mask wearing people set to 100 (mandated use), and Test 4 sets the percentage of people following both ordinances (mask wearing and socially distancing) to 0. The results are shown below:

Test 1:

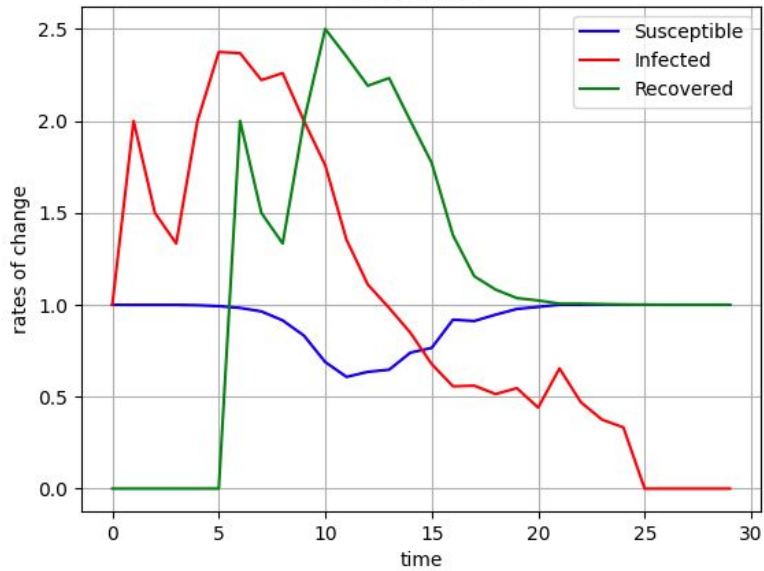
Covid Model State Graph 1/3 Through



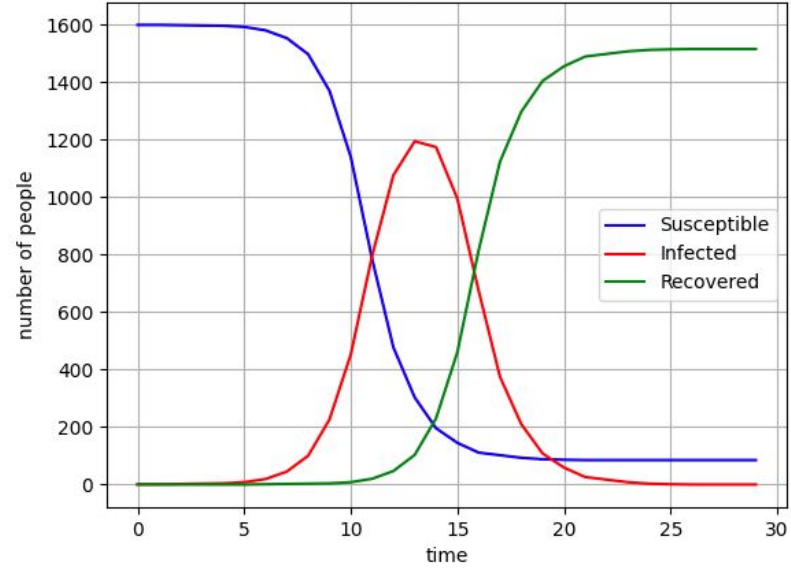
Covid Model State Graph End



Derivatives

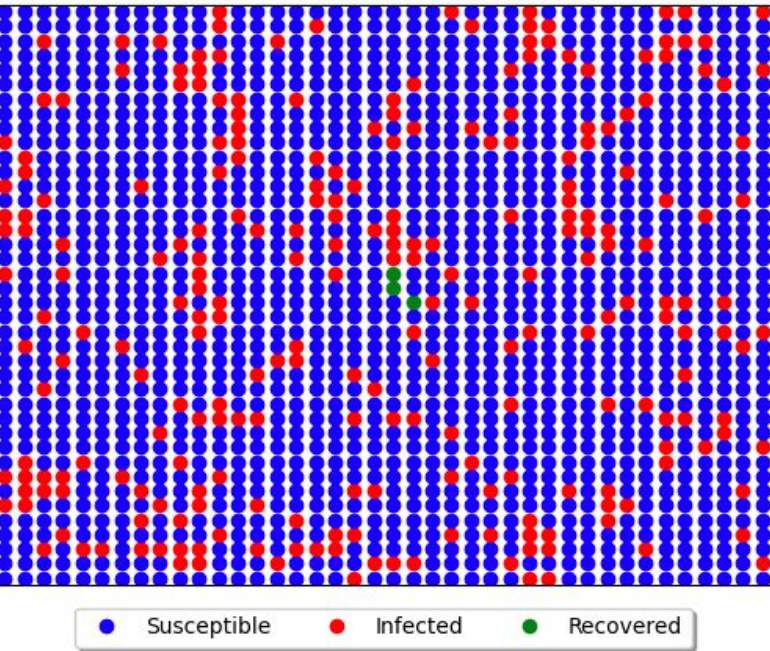


Populations

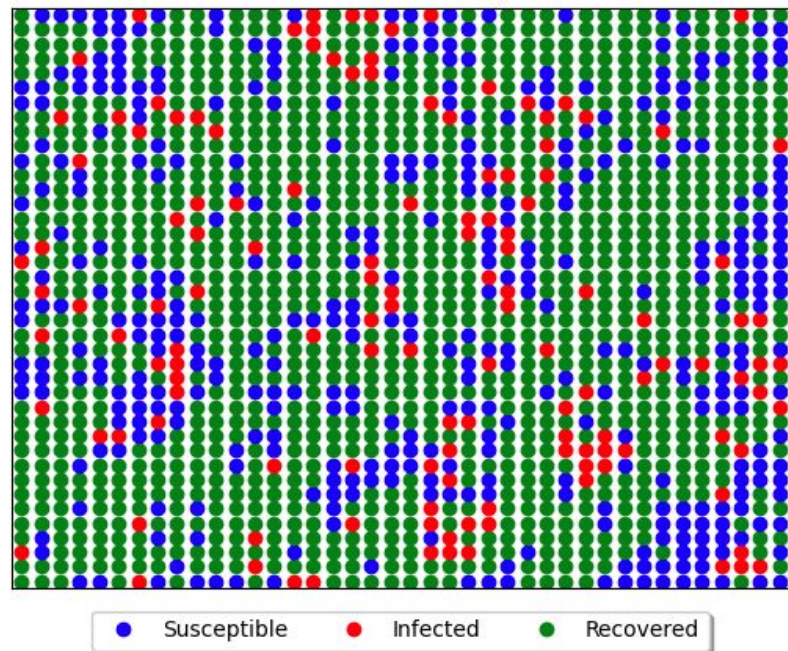


Test 2:

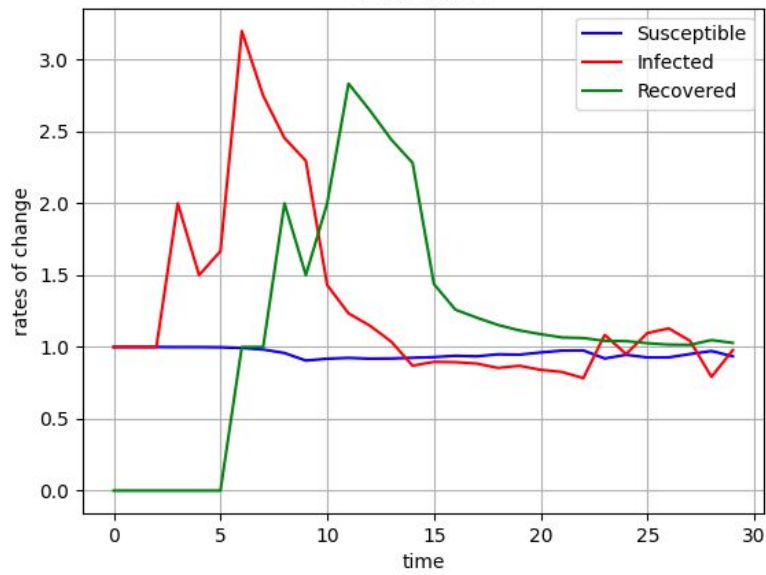
Covid Model State Graph 1/3 Through



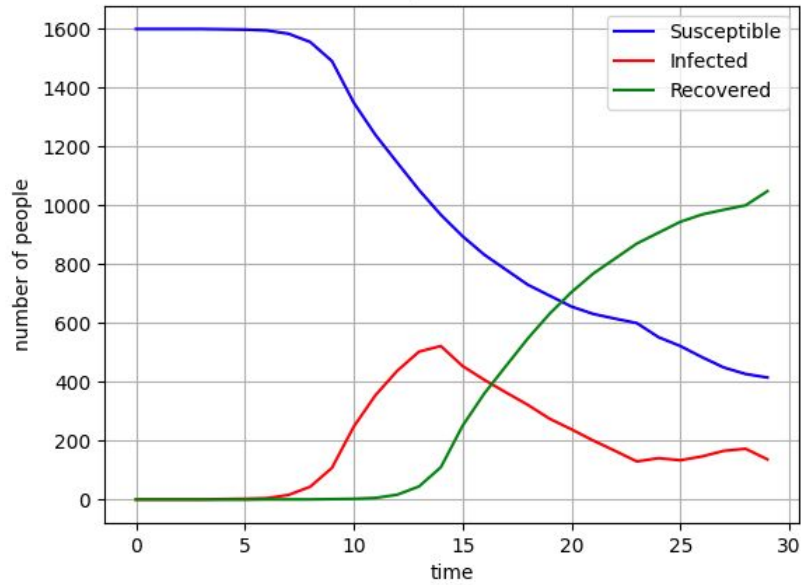
Covid Model State Graph End



Derivatives

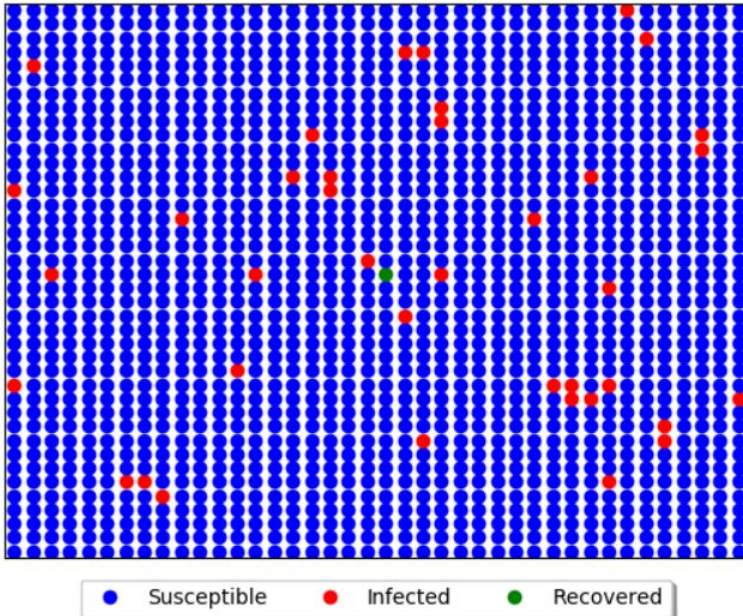


Populations

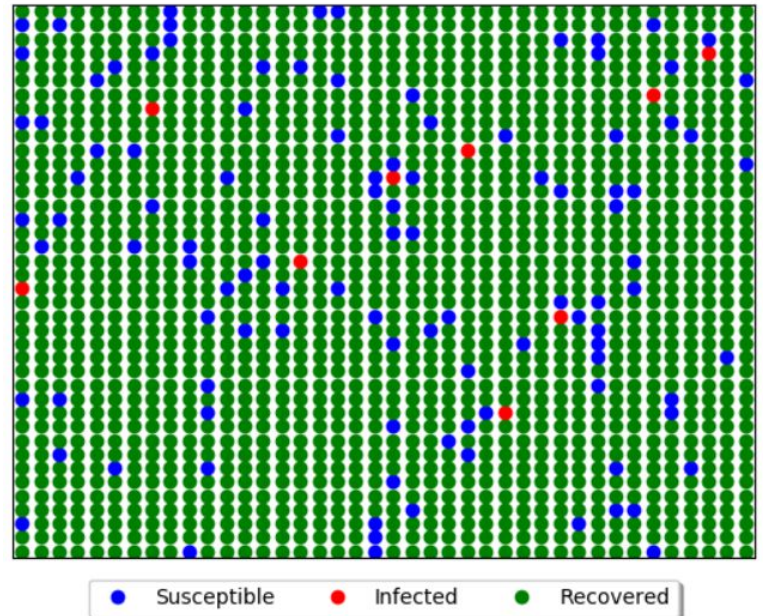


Test 3:

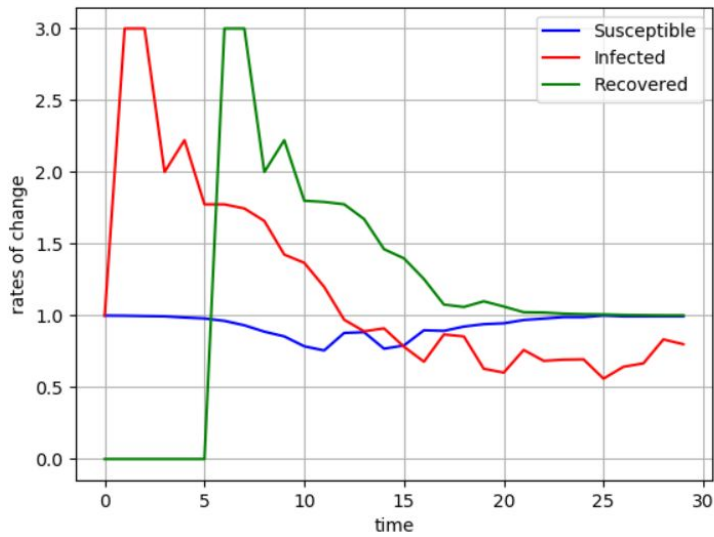
Covid Model State Graph 1/3 Through



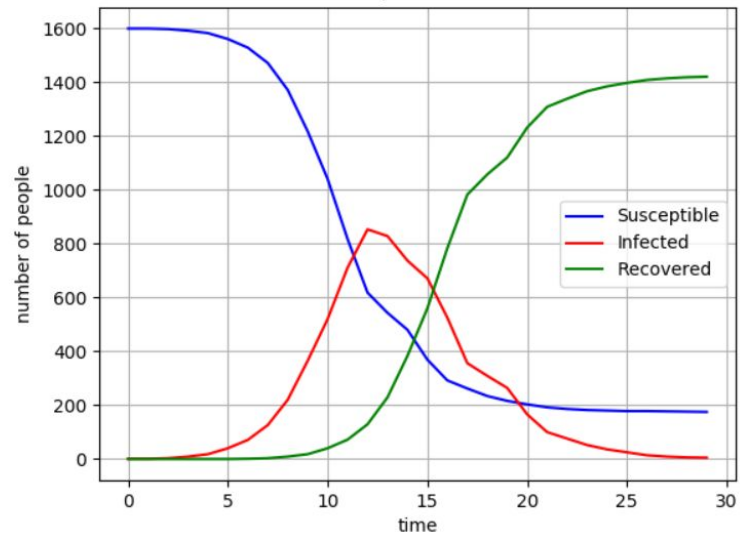
Covid Model State Graph End



Derivatives



Populations



Discussion

All hypotheses were correct regarding the impacts of these mandates. Compared to the results from Test 1 (the control), tests 2 and 3 (quarantine and masking mandates, respectively) both had far smaller peak populations of infected, with the reactive quarantine mandate of 10% more than *halving* the peak. As well, the impact of the mask mandate was significant although less impactful than the quarantine mandate, as predicted, and the quarantine uninfected populations were indeed vastly more clustered than other end-states.

Interestingly, the reactive quarantine mandate (Test 2) unexpectedly correctly predicted wave behavior in the contagion spread. Unlike the unmitigated spread test and mask mandate test, the reactivity of the government mandate to quarantine — which allows the quarantine mandate to drop once infected populations again reached a low number (similar to many American state policies) — led to a second wave of infection that prevented the contagion from dying out in the model as early as it could have. This makes sense, as the premature lifting of a quarantine certainly leads to potential second waves of infection; however we were unsure if this effect would persist from the global-regional level down to the microcosmic scope of this model. Such a behavior is very negatively impactful to long term populations of uninfected, and this specific outcome reveals that ending a quarantine while 10% of a population is still infected is far too early.

In further testing, it appears this effect can persist at least down from 10% to 1% - 2%, demonstrating a resilience to this behavior as well.

These results as a whole strongly suggest that government mandates for masking, distancing, and especially quarantining, all have significant negative effects on the speed of contagion spread through a local population.

Conclusions

The COVID-19 pandemic is the divergence of discourse regarding the efficacies of government mandates. As some people grow to question these mandates, it is the responsibility of the scientific and academic worlds to communicate the importance of these mandates and why they should be adhered to. It is evident, however, that it is easier for people to learn from their own experimentation and research, rather than have someone try and teach them. As such, we wanted our model to be a personalized and comprehensible experience that allows people to test efficacy of ordinances for themselves. We wanted our model to be as accurate as possible, while still allowing for this level of personalization, which we achieved as best we could given the amount of time we had.

In pursuing this problem, we decided first to make a simple SIR model of contagion behavior as it pertains to the 2019 coronavirus and its spread at a local level, and then, in pursuit of emergent behavior relating to ordinances affecting said model, we reimplemented the model within an agent based framework. Once that was done, the issue of accessibility to an audience was approached by creating a web app for the model, wherein users can specify a wide array of parameters to test the results of, in order to develop an intuition for the impacts of ordinances on contagion spread.

This design was very effective at modelling the local contagion behaviors, as validated by the consistency of the results with predictions from the literature. As well, the web app is very effective at promoting the development of understanding of ordinance impacts, through the model's 4 stage visualization and graphs of the populations' sizes and rates of change.

For the most part, as a group, we were relatively on schedule for the duration of the development process. There were times where we did delay some steps of the process, however said delays were anticipated and we returned to the regular schedule fairly quickly. The most significant mistake we made in scheduling throughout this project was scheduling the paper for the final week before being due, when it should have been drafted earlier.

Possible extensions to this model are numerous, but should be approached with caution in any agent based model, as simplicity begets more natural emergent behavior. Heeding that warning, the model can be extended to account for:

- Many other attributes of population such as age or prior health condition;
- Variation in population density throughout the model.
- Government reactivity for the other mandates, i.e: mask-wearing and distancing;
- The relationship of risk-factors to ordinance adherence (the elderly or immunosuppressed are more likely to wear masks, *ceteris paribus*)

Additionally, on a structural level, the model could be expanded to not only include agents, but separately delineated communities of agents called “Bubbles” that generally self isolate as a group rather than as individuals or households, to examine the effect of that phenomena.

On the accessibility side of the project, an important potential subsequent action is the publication of this web app to an SEO-optimized website it’s own, or to port the web app to a mobile device platform.

While there were few scheduling delays, the delays that did occur set us marginally aback. So much so that we may have had enough time to add more extensions and inputs to the model, make the web interface more visually appealing, and deploy the web interface to a publicly accessible online link (currently the project only runs on a local host ip address). The lesson to be learned is that time management can *always* be more efficient, and projects like these are only as good as the amount of time and effort that is put in. While we definitely put a significant amount of effort into this project, there definitely could have been more time dedicated to it as well.

References

- Abou-Ismaïl, Anas. “Compartmental Models of the COVID-19 Pandemic for Physicians and Physician-Scientists.” *SN Comprehensive Clinical Medicine*, Springer International Publishing, 4 June 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7270519/.
- Bump, Philip. *Analysis | Not-That-Bad or Not-That-High: How Advocates of a Return to Normal Misrepresent Coronavirus Deaths*. 10 Apr. 2020, www.washingtonpost.com/politics/2020/04/10/not-that-bad-or-not-that-high-how-advocates-return-normal-misrepresent-coronavirus-deaths/.
- Chu, Derek K, et al. “Physical Distancing, Face Masks, and Eye Protection to Prevent Person-to-Person Transmission of SARS-CoV-2 and COVID-19: a Systematic Review and Meta-Analysis.” *The Lancet*, The Lancet, 1 June 2020, [www.thelancet.com/journals/lancet/article/PIIS0140-6736\(20\)31142-9/fulltext](http://www.thelancet.com/journals/lancet/article/PIIS0140-6736(20)31142-9/fulltext).
- “COVID-19 Pandemic Planning Scenarios.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, www.cdc.gov/coronavirus/2019-ncov/hcp/planning-scenarios.html.
- Cuevas, Erik. “An Agent-Based Model to Evaluate the COVID-19 Transmission Risks in Facilities.” *Computers in Biology and Medicine*, Elsevier Ltd., 20 May 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7237380/.
- Froese, Henri. “Infectious Disease Modelling: Understanding the Models That Are Used to Model Coronavirus.” *Medium*, Towards Data Science, 3 May 2020, towardsdatascience.com/infectious-disease-modelling-part-i-understanding-sir-28d60e29fd6c.
- Katz, Josh, et al. “A Detailed Map of Who Is Wearing Masks in the U.S.” *The New York Times*, The New York Times, 17 July 2020, www.nytimes.com/interactive/2020/07/17/upshot/coronavirus-face-mask-map.html.

Leung, Nancy H. L., et al. *Respiratory Virus Shedding in Exhaled Breath and Efficacy of Face Masks*. 3 Apr. 2020, www.nature.com/articles/s41591-020-0843-2.

MacMillan, C. (2020, June 19). *What Does Recovery From COVID-19 Look Like?*
07 Dec. 2020, <https://www.yalemedicine.org/news/covid-19-recovery>

Saad, Lydia. “Americans Still Social Distancing, but Intensity Is Evolving.” *Gallup.com*, Gallup,
3 Nov. 2020,
news.gallup.com/poll/312206/americans-social-distancing-intensity-evolving.aspx.

Sylte, Author: Allison. *We Asked Our Medical Expert to Respond to the Reasons Why People Don't Wear Masks*. 1 May 2020,
www.9news.com/article/news/health/coronavirus/doctor-responds-to-reasons-for-not-wearing-masks/73-d2c5c731-da4a-4fc0-b8d7-b0195e9abf50

Zhao, Shilei, and Hua Chen. “Modeling the Epidemic Dynamics and Control of COVID-19 Outbreak in China.” *Quantitative Biology (Beijing, China)*, Higher Education Press, 11
Mar. 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7095099/.

