CSS 458 Final Project

Team 1796

Gabriel Pitzel - Bryan Lin - Oliver Jeremiah Fernandez

Simulating Impact of Epidemic on Population Centers, Resources, and General Populous

[**Contents**](#_g5m5p7syaqfk)

[Introduction 1](#_ggpiboe77bwq)

[Program Files 1](#_ffulpbhk02zb)

[Working Repository 2](#_psbysvxgwyd1)

[User Manual 2](#_47nvf1dt6t68)

[Analysis Report 5](#_l2gfnb648kns)

[Project Contributions 8](#_uzcj1zdbilt)

### 1. Introduction

This is the final writeup for the CSS 458 final project, and serves as the read-me, analysis, user manual and more to our final project submission, an epidemic simulation model.

### 2. Program Files

All required program files are included in the provided .zip. In addition, the complete set of files can be found on the online repository, see section 2.

Program Files:

Config\_And\_Run.py

Driver.py

Agent.py

Events.py

Locations.py

Parameters.py

Resources.py

Sim\_Tools.py

Testing\_Module.py

Visuals.py

For user operation, only Config\_And\_Run.py and Parameters.py are relevant. See Section 4 for usage instruction.

### 3. Working Repository

The repository for the project can be found here:

<https://github.com/IProxyPI/PlagueSim>

### 4. User Manual

There are two relevant files to running the simulation:

Config\_And\_Run.py - Execution config and driver

Parameters.py - World config

**Config\_And\_Run.py:**

Config\_And\_Run.py serves as the driver program with the following utility:

* Configure the simulation parameters
  + Number of executions
  + World presets
  + Simulation time
  + World response
  + Live outputs
  + Analysis
* Creates the driver objects and executes the program

Program execution is done by running Config\_And\_Run.py.

The parameters of the simulation are configured in their respective categories, with the following options:

*simulation\_time - Months*

*print\_interval - Time steps between graph/progress bar prints. -1 to disable*

*world\_factor - Factor to multiply city size by*

*number\_of\_simulations - Number of times the simulation is run, results are averaged*

*world\_preset - Preset for which city is generated and used in the simulation.*

*Available world presets:*

* *"Mini city" ~= Population 1601 \* world\_Factor*
* *"Large city" ~= Population 6568 \* world\_Factor*
* *"Small town" ~= Population 866 \* world\_Factor*
* *"Downtown" ~= Population 4942 \* world\_Factor*

*analysis\_checklist = [ Display real-time graph,*

*Real-time graph cumulative or not*

*Track an agent ]*

*respose\_effects = [ Enforce masks,*

*Enforce vaccine,*

*Enforce isolation ]*

**Parameters.py:**

Parameters.py functions as the config files for the simulation parameters that are less likely to be configured on a run-by-run basis.

* Holds parameters for:
  + Population
  + City generation
  + Structures
  + Disease
  + World response
  + Action effectiveness

These parameters are broken down in the following categories:

* *Disease Parameters:*
* *Infection chance*
* *Contagion period*
* *Infection period*
* *Immunity period*
* *Time before symptoms show*
* *Airborne infection percentage*
* *Contact infection percentage*
* *Protection parameters*
  + *Mask infection reduction*
  + *Hand washing infection reduction*
  + Vaccine infection reduction
* *Population parameters*
  + *Perc stay home if sick*
  + *Perc mask if sick*
  + *Perc will announce if sick*
  + *Perc washes hands if sick*
  + *Perc immune compromised*
  + *Perc anti mask*
  + *Perc anti isolation*
  + *Perc anti vaccine*
  + *Perc asymptomatic*
  + *Chance of announcing if sick*
* *City parameters*
  + *People per household*
  + *Workers per retail*
  + *Workers per recreation*
  + *Workers per hospital*
  + *Workers per office*
  + *Workers per farm*
  + *Hospital capacity*

\* Parameters in which their functionality has not yet been implemented are omitted

Once configured, simply run Config\_And\_Run.py.

There are two spaces for output:

In the console will be simulation progress, as well as numerical resulting data. After all simulations are complete a final analysis will be presented on sim averages. This includes, but is not limited to:

Total infections

Total population

Total deaths

Infections per trait

% of population killed

In plots will be two graphs. An cumulative graph will showcase the given number of each status in our modified SIRS model at any given time. A non-cumulative graph will show new trends for each timestep, in this case being an hour.

Additional information can be toggled, primarily agent tracking.

By toggling Track and agentwithin Config\_And\_Run.py (Fig 1), real time data will be printed during the simulation's runtime on a single tracked agent.

(Fig 1)

*analysis\_checklist = [ Display real-time graph,*

*Real-time graph cumulative or not*

***Track an agent*** *]*

For output information, there are 4 primary categories:

Unsanitary : Either does not mask when sick/exposed, or does not wash hands

Anti-isolation : Will not stay home if sick

Unannouncer : Does not alert others if they are sick

Asymptomatic : Shows no symptoms, unaffected by disease, still can spread

### 5. Analysis Report

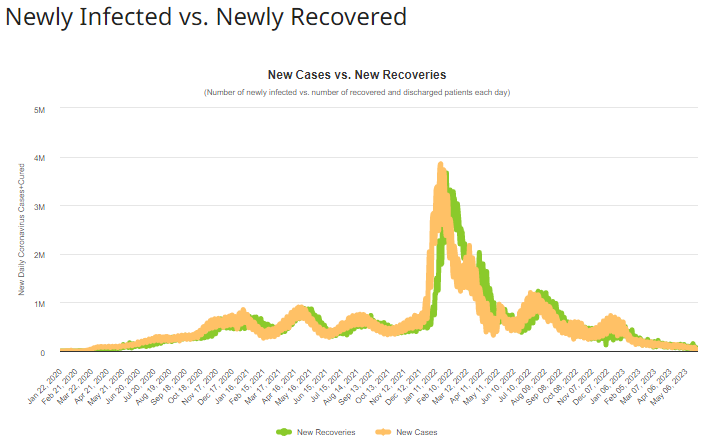
Through the use of the model, we can gather a large amount of data and attempt experimental responses to a variety of different disease archetypes. Due to the complex and massive nature of the real world and flow of population/disease, direct numerical representation is not possible in a simulation of this scale and timeframe. Instead, we have designed a model that simulates and analyzes trends. This allows us to focus on determining the largest threats during an epidemic, as well as testing responses to act upon these threats.

To begin with the analysis:

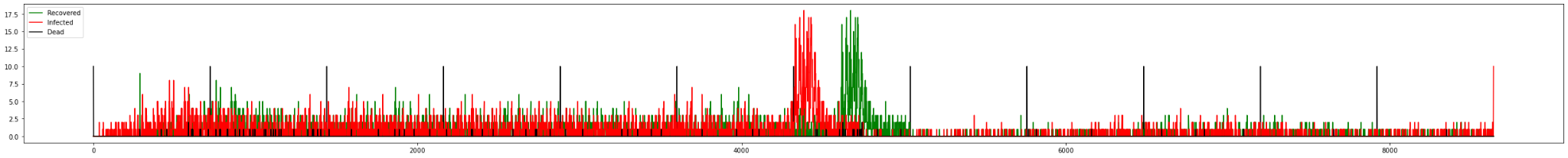
Verification and validation.

To validate the model, we built the model to best replicate the parameters we could find on the recent Covid-19 epidemic. The main data here is the effects of a highly contagious disease that, after an interval of in our case 6 months, drastically increases in its contagiousness. This replicates the appearance of the omicron variant, and due to the sudden increase in numbers, triggers an automated government response within the simulation, enforcing further masking, isolation and self quarantine, and deploying vaccines for the new variant. The Resulting infections immediately drop, then entering a far lower cycle than what was previously run.This trend matches the real world trend data found on the Covid-19 and omicron epidemic, and will be shown below (Fig 2, Fig 3).

(Fig 2 - Covid-19 New cases VS New recoveries)



(Fig 3 - Simulated Covid-19 New cases VS New recoveries)



In addition to graph comparison, internal tests were done to ensure the model replicated logical actions. Primarily:

Over-Infection - Implementing a virus with a too-high infection rate would overinfect, resulting in no remaining susceptible civilians. This would cause the disease to die off.

Infection Waves - The infection behaved in waves as susceptible individuals recovered and lost immunity

Population Density - The rate of spreading was influenced by the density of population, and spread far quicker in denser spaces, such as offices and hospitals.

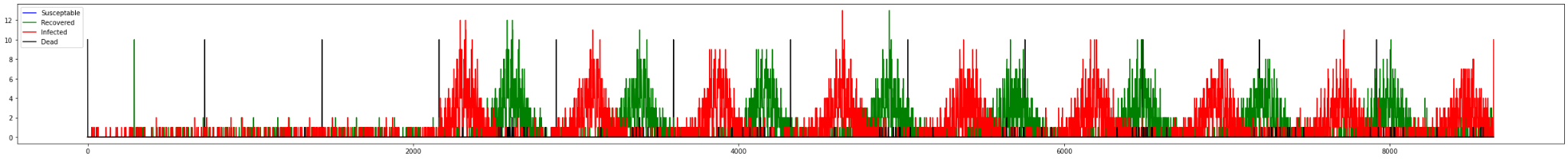
Plans for further analysis exist, such as graphing movement throughout individual cities and neighborhoods, as well as graphing movement between cities with limited contact. Unfortunately, these features were not implemented and this validation was unable to be performed.

To verify the model, a series of trackers were implemented to allow for real time analysis on the functionality of the model, as well as a limited set of unit tests to confirm functionality.

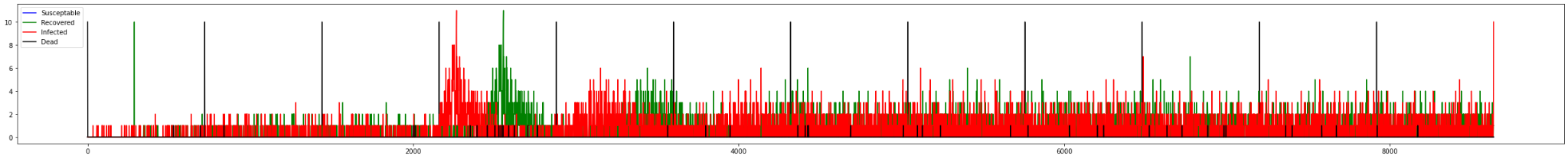
Once in a position in which we were happy with the model and its outputs, we were able to run some experiments to analyze different threats that may occur during an epidemic.

For example, building further off of our Covid-19 model, we altered what responses were put in place with the following results:

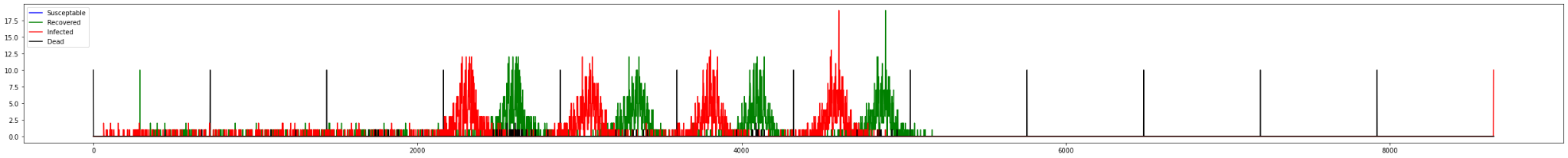
( Fig 4 - Enforced masking )



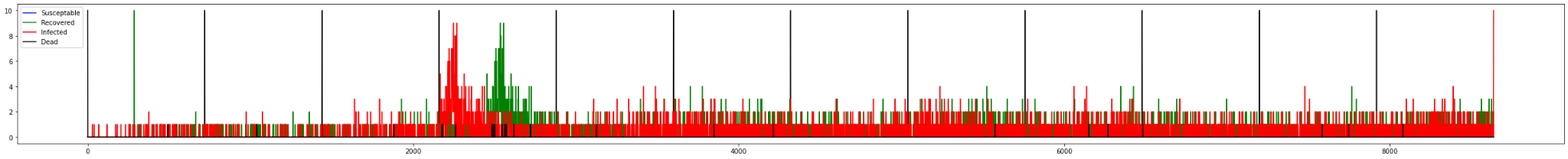
( Fig 5 - Enforced vaccine)



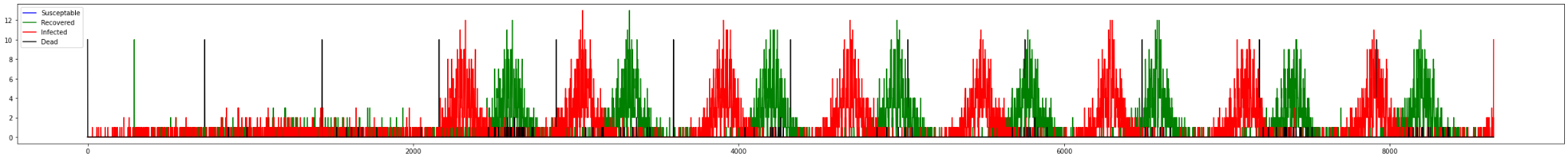
( Fig 6 - Enforced isolation)



( Fig 7 - Enforced everything)



( Fig 8 - Enforced nothing)



\* Each graph is just one recorded simulation of many

This provides insight into the effectiveness of each individual response, usable for benefit-cost analysis. Beyond proving that ‘helpful actions help’, by analyzing the depth of the influence of each factor, and further testing can show how these interact with each other, allowing for a user to determine the best combination of responses to different disease scenarios.

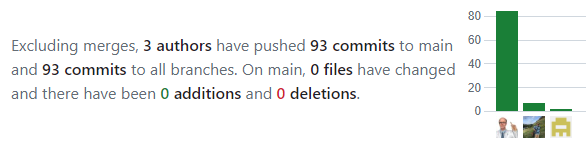
A major aspect that is not considered, however, is the effect on aspects other than the rate of daily infections. Due to the detailed workings of the simulation, with extension more factors can be analyzed using these parameters. In future versions, impact on supply chains will be included, with the expectation to see:

* Are vaccines or masking more effective at preventing food-borne contamination?
* Is enforcing isolation worth the risk to the supply chain? How long must isolation continue before it is safe to resume daily life?
* Will it save more lives to instate an isolation mandate at the cost of slowed vaccine development, or fast-track vaccine development at the cost of further cases now?

Creating the ability to answer these questions with in depth graphs and analysis is the purpose of this model.

### 6. Project Contributions

Commit contributions:



<https://github.com/IProxyPI/PlagueSim/commits?author=IProxyPI&since=2023-05-01&until=2023-05-31>

<https://github.com/IProxyPI/PlagueSim/commits?author=ojfernandez&since=2023-05-01&until=2023-05-31>

<https://github.com/IProxyPI/PlagueSim/commits?author=Bryan253&since=2023-05-01&until=2023-05-31>

Individual Contributions:

Gabriel:

* Model plan
* Model design
* Model implementation
* Model testing/tuning
* Parameter research
* Presentation slides
* Presentation
* Documents

Oliver:

* Model plan
* Analysis
* Verification
* Presentation

Bryan:

* Model plan
* Parameter research
* Presentation