

Simulating the Spread of COVID-19

Aleksey Panas, Anand Singh, Jannate Rahim, Rohit Shetty

December 2021

1 Project Description

COVID-19 is a disease caused by a virus called SARS-CoV-2. This disease emerged around December 2019 causing millions of deaths all across the world. The virus spreads from an infected person to others through respiratory droplets and aerosols. The practice of social distancing and vaccines are factors that helped reduce the spread of the virus. Much data has been collected to analyze the exponential spread of Covid-19, and assist with slowing it . The data sets are crucial in scientific research as they allow us to test or check future insights and will be quite necessary for future long-term study.

The goal of our project is to present a simplified model that will simulate the spread of an epidemic in a given population. With this model, we will try to simulate some real-world conditions and interpret how these conditions affect the spread of an epidemic(Covid-19) in this case.

A ‘curve’ in our context is a graphical representation of the onset of illness among cases associated with an outbreak. The term “Flatten the Curve” means to spread out the rate of infection in order to prevent an overwhelmed health care system and infrastructure. During our research, we will explore many factors that have an effect on the curve. We will also analyze how each of the preventative measures (social distancing, quarantine, vaccinations, etc) and socio-economic factors (population density, access to healthcare, etc) affect this curve.

The question we want to ask is, **given the nature of a particular urban center or town, can we accurately simulate the outcome of a Covid outbreak if it were to happen there** and identify causing factors which lead to an undesirable “curve” outcome?

With gradual improvement of the simulation through feedback from real world data and scientific research, we might get a usable tool for predicting the risk an epidemic poses on regions with particular characteristics. This would allow testing policies in advance, and foreseeing the leading causes of the spread of infection.

2 Relevant Dataset

A relevant dataset that we used for our project comes from the Government of Canada. This dataset holds information about various Toronto neighborhoods and their CoVID data. The dataset is available in many formats, including CSV and JSON, and contains statistics on the population of each neighborhood, such as the percent of elderly people, density of the population, percent young people, and other similar numbers. Alongside this, the dataset shows CoVID case progression, testing practices, and outbreaks. We used the variables `population_density`, `percent_over_65/85`, and neighborhood area, in order to run a simulation with similar parameters as this data. This will then allow us to see a similar covid case rate and possibly be able to conclude that there was a relationship between the variables.

Link to Dataset:

https://open.canada.ca/data/en/dataset/2d86f026-10b4-44ac-a68b-80a9dd5dd390/resource/ab558292-2e62-4b71-944b-aa6c19cc5d41?inner_span=True

3 Computational Overview

Computational Q4: In our simulation, we incorporated several computational models. We successfully created a model to calculate the probability of infection based on 2D geometric movement paths over time. We also created a model for calculating body temperature change in different seasons, to simulate the real-world conditions that affect health and body immunity.

To simulate the navigation of people in the city we implemented a path-finding algorithm. It goes around the vertices of shapes and generates paths to the final destination. This was a very complex model that used recursion trees to keep track of all the possible routes, and works on almost any similar problem domain in path-finding.

We implemented a Sim Manager class that was similar to the one developed in class, which was in charge of managing the entire simulation, running the probability models, and keeping track of everyone in the city and their movements and actions.

Our program has a launcher menu allowing you to launch the simulation and compare results with a real-world data making it interactive and visually attractive. The program also uses pygame to help simply visualize the simulation world with different rectangles representing buildings and created a randomised city grid as well.

We then used pandas and plotly to plot and display data, allowing the users to compare the Covid-statistics with real-world data obtained from our dataset. Our focus was the different neighbourhoods with toronto and the dataset contained information regarding relevant demographics.

We have used several interesting and unique modules to help bring our simulations to life. We used guizero to implement the menu GUI, which is a GUI library built on top of Tkinter. We used pygame to display graphics in a pygame window using the main loop with pygame event handling. We used pandas and NumPy libraries to help with data-wrangling and graphing it with plotly. We also built a geometry module containing path-finding and classes to represent 2D geometry such as points, vectors, rectangles, etc

Add anything else you want

4 Instructions for Obtaining Sets and Running Program

- Download and install all the files from markus
- Check and install all libraries from requirements.txt
- Now download the dataset needed from UTSEND:
 - Claim ID: T6pvNXbx8uEH2Tq5
 - Claim Passcode: 9Br7VmXiYbo8hT2w
- Make sure Toronto_data.csv is in a folder called "datasets." This folder should be in the main directory with main
- in the project directory, make sure to have a folder called "simdata" and a folder called "logs"
- Make that all zipped folders in the main directory are unzipped!
- 1. In the launcher, go to Sim Launcher tab and fill out a desired configuration. Some of these variables currently have no effect on the sim. Make sure city blocks are below 5 on both x and y. Once ready, press Launch
- 2. To view data comparison, go to "Compare Data" and select a cvs file from the simdata folder. You must first run a simulation for a short time to generate this. Once a simulated dataset is chosen from simdata folder, select the caption column as day-number and the values column as case-proportion. Now press run comparison

5 Changes Made to Project Plan

The initial goal of our project was to present a simplified model that would simulate the spread of an epidemic in a given population. We would then try to simulate some real-world conditions and interpret how these conditions would affect the spread of an epidemic(Covid-19) in this case. We did consider reversing the problem approach, that is, instead of making a simulation, we would use data to train a prediction model. We deemed that approach would be a bit too complex. Instead, we stuck with the simulation, keeping in mind that it wouldn't be flawless at first attempt. The simulation will need to be improved over time using surrounding data.

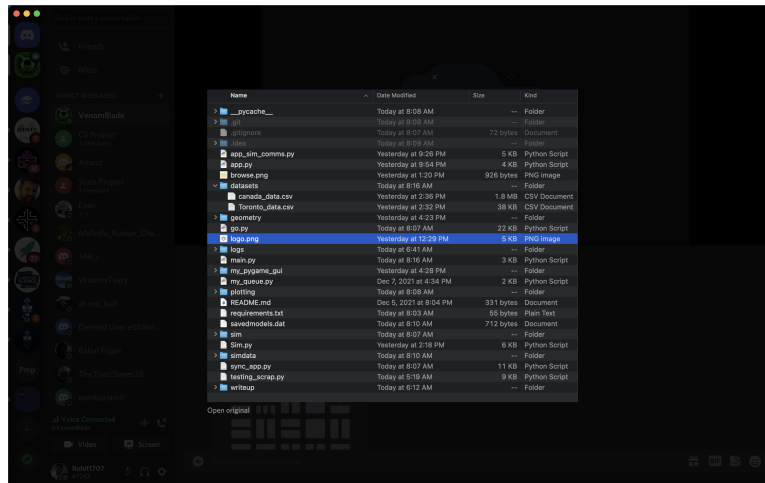


Figure 1: This is an image which shows us the file path for Toronto data sset

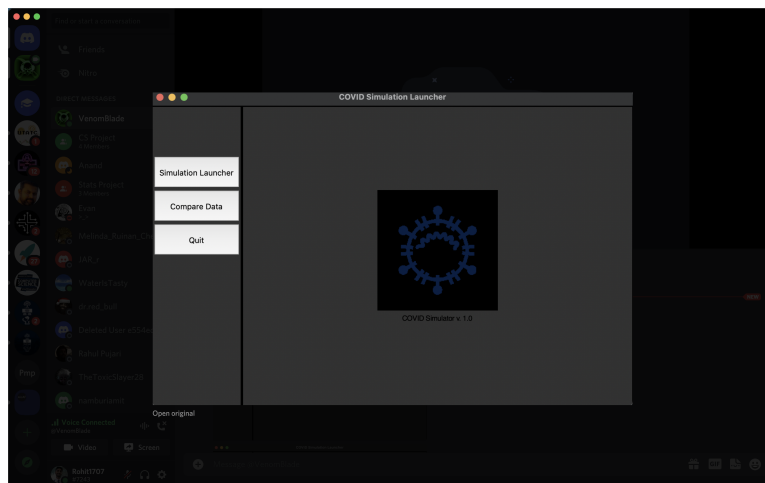


Figure 2: This is an image which shows us the app when main.py is run

6 Discussion

Our simulation obtained works quite well. We successfully incorporated many factors that affect the spread of covid-19 including mask wearing precautions, the affect of different weather and temprature conditions, age demographics and several other factors realted to health, immunity and spread of infectious diseases.

The results we noticed showed us some simple general trends. The wearing of masks and other precautions shows us that the number of cases would decrease in such scenarios. We can clearly observe a desirable/flatter curve, if we take necessary precautions.

We display the simplified results using plotly showing how the proportion of cases changes with time. But, it is quite obvious that we have not been able to completely model the real world, as seen is the graphs obtained while comparing it to real world data. This is expected since there are many real-world socio-economic factors that we simply haven't thought of or can't feasible replicate. In our path-finding algorithm we also assumed social distancing squares instead of circles, because we couldn't develop an algorithm that would work with perfect circles. This is something that would require an understanding of higher level concepts linear algebra.

Overall, we are are very satisfied with our project and even though it didn't produce real-world accurate results, it gives the user a good idea about the outcome of a Covid outbreak given the nature of an urban centre or town. The simulations produced give us a way to visualize the spread of Covid and the graphs help us compare the results to real world data.

7 Works Cited

"The University of Alabama at Birmingham." *UAB*, <http://www.uab.edu/>.

Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, <http://www.cdc.gov/>.

"What Is Coronavirus?" *Johns Hopkins Medicine*, <https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus>.

"These Simulations Show How to Flatten the Coronavirus Growth Curve." *The Washington Post*, WP Company, 14 Mar. 2020, <https://www.washingtonpost.com/graphics/2020/world/corona-simulator/>.

"Toronto Air Pollution and Covid-19 Data by Neighbourhood - Toronto Air Pollution and COVID-19 Data by Neighbourhood." *Open Government Portal* , <https://open.canada.ca/data/en/dataset/2d86f026-10b4-44ac-a68b-80a9dd5dd390/resource/ab558292-2e62-4b71-944b-aa6c19cc5d41?innerspan=True>.

“Pygame Front Page¶.” Pygame Front Page - Pygame v2.1.1 Documentation, <https://www.pygame.org/docs/>.

“Multiprocessing - Process-Based Parallelism¶.” Multiprocessing - Process-Based Parallelism - Python 3.10.1 Documentation, <https://docs.python.org/3/library/multiprocessing.html>.

“Getting Started.” Getting Started - Guizero, <https://lawsie.github.io/guizero/start/>.

“Pandas Documentation¶.” Pandas Documentation - Pandas 1.3.5 Documentation, <https://pandas.pydata.org/docs/>.