Final Project

Abdullah Saad #1043850

University of Guelph CIS*3130

Final Project

Problem Formulation

The coronavirus pathogen causes a contagious, severe, acute respiratory condition. A community outbreak of pneumonic characteristics began in Wuhan, China during January 2019. This viral outbreak was attributed to a novel coronavirus, SARS-CoV-2 also known as COVID-19. China implemented a shutdown of Wuhan and was joined by other countries soon after to suppress the growth of the resulting global pandemic.

History has shown us numerous epidemics and pandemics around the globe, such as the bubonic plague, cholera, smallpox, and various strains of influenza. These differ in their contagiousness and outcomes, so they are passed more or less easily and kill or severely harm more or less people.

With the high contagiousness and severe outcomes that the novel coronavirus has shown the world, many are creating simulations to assess different infection-related scenarios to produce strategies for the future and understanding of the past and present. Given that the infections around the globe have produced a grand variety of data points, simulation models can allow us to educate, plan, and execute as necessary.

In a time where pharmaceutical intervention is yet unable to reduce or control the spread of the virus, the Canadian Federal Government and Health Canada are implementing efforts for non-pharmaceutical interventions (NPIs), by intercepting the transmission of COVID-19. For example, response procedures such as self-isolation and complete lockdown are utilized as NPIs.

This project builds a simulation program in a Python environment for showing how important physical distancing and lockdown procedures can be whilst trying to mitigate the risk of the virus’ spread. Note that COVID-19 infections increase exponentially by default if uninterfered with in a population.

My mathematical modelling and simulation of epidemics will analyze different containment concepts. The goal is to aid efforts in making a prediction of the future of COVID-19, testing NPI scenario effectiveness, thus informing policy corresponding to quarantine, and the limitations placed on schools and the country's economic activity.

The simulation is modelled by using different equations to forecast for the upcoming days. The simulation reproduces the relevant aspects of the real world as precisely as possible by trying to illustrate the degree of contact between people as well as measures to limit the number of contacts to increase or decrease the number of infected leads by implementing various containment strategies. The statistics generated by the simulation are intended to illustrate how important it can be to obey the restriction rules and showing what would happen if we followed such rules, as these measures may reduce the risk of infection.

Methodology

Setting of objectives and overall plan

The simulation will utilize ‘pure python,’ including NumPy, and matplotlib as a visualization tool. It will show the spread of an infection among a certain population using a “billiard ball model”, where each individual is represented as a perfectly elastic 2D cycle, which travels at a constant speed on a straight line. These circles are categorized as “healthy person (green)”, “sick person (red)”, “immune person (blue)”, and “dead person (black)”. This algorithmic modelling method employs a degree of randomness as part of its logic; bits are then the auxiliary input, and the output will be determined by the random bits.

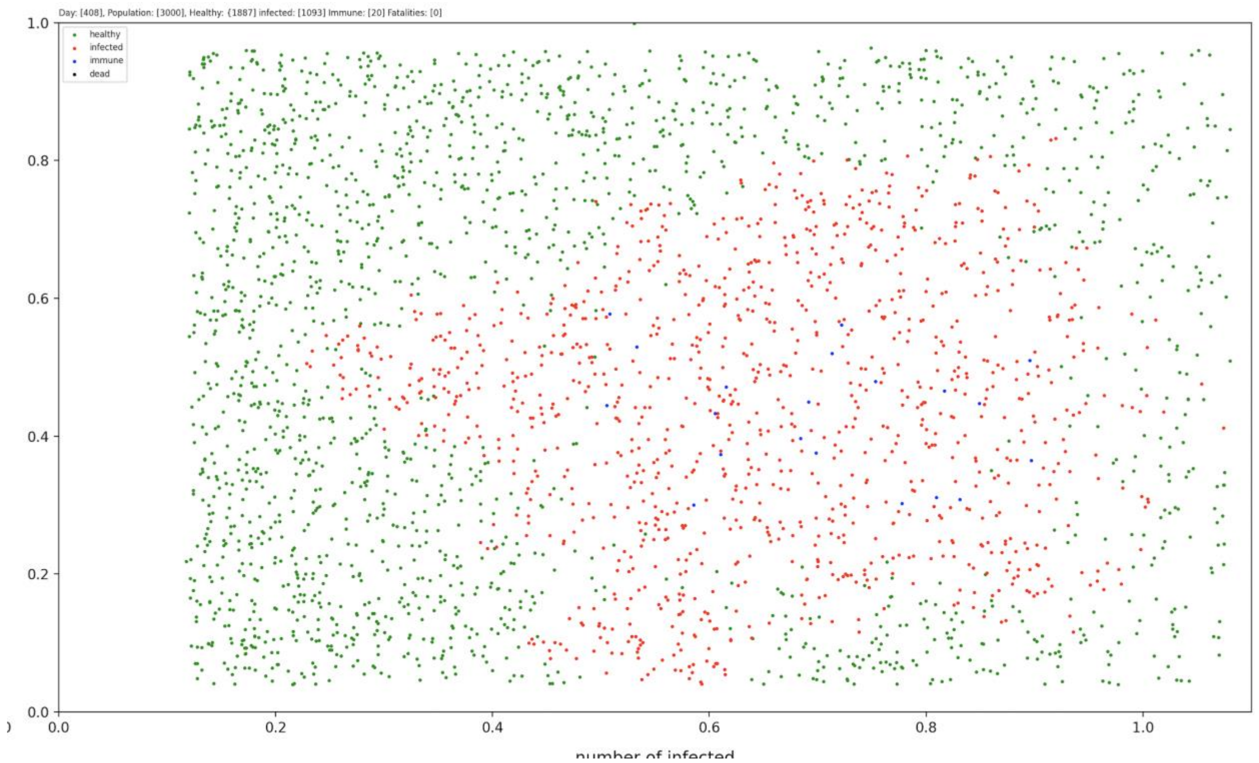
Initializing the population matrix array for each person is a two-dimensional data structure where numbers are arranged into rows and columns. Each column will include ID of elastic 2D cycle, and the location of the cycle in our “billiard ball model”, and the direction of each cycle and different speed dependent on different scenarios and the state of the cycle “0 is healthy, 1 is sick, 2 is immune, 3 is dead.”

Initialize the array matrix by ‘numpy. zeros() function returns a new array element's value as 0’

- Draw samples from a uniform distribution for x and y coordinate by using “numpy.random. uniform (low=0.0, high=1.0, size=None) “
    - Where low is Lower boundary of the output interval. All values generated will be greater than or equal to low.
    - Where high is Upper boundary of the output interval. All values generated will be less than or equal to high.
    - The size will be the population size

- Draw random samples from a normal (Gaussian) distribution for the heading of the cycle and the random speed for each cycle by using **numpy.random.normal(***loc=0.0*, *scale=1.0*, *size=None***)** The normal distributions occurs often in nature.
  - Mean ("center") of the distribution will be 0
  - Standard deviation (spread or "width") of the distribution. Will be 1
  - **Size will be the population size**

The simulation will move the cycle within the bounders and transmits the COVID-19 disease by specific disease range between cycle and use probability by roll a dice to see if the healthy person will be infected by compare the np.randome.random() to the infection chance value , Also, each cycle will save when got infected time for recover and it will roll a die to see if the cycle will die or recover by rolling a dice if will recover or die and it also depends on the age.



The SIR / SIDARTHE model

The SIR model is based on so-called compartments. Which means that the set of all people under consideration is divided into disjoints sets where each letter in SIR stands for a compartment.

In the SIR model, the compartments are

S (t) b = Susceptible (infectious persons),

I (t) b =Infected (infected persons)

R (t) b = Removed (from the disease process “Removed" persons)

T= The totality of all persons

N(t) = I(t) + S(t) + R(t)

- S(t) will represent the susceptible people how could get the disease
- I(t) represent infective people who are currently have the disease and can transmit it to other
- R(t) represent infected people who recovered from the COVID-19
- The population is sum of susceptible + infected + removed.

B(transmission rate)* i(t) *S(t) number of infected people in day t

A (rate of recover) * i(t)

The model makes striking assumption: people are following immune to their disease; there is no different between dying or recovering from the disease, infected people are immediately contagious.

$$\frac{ds}{dt} = -\beta\frac{sI}{N}$$

$$\frac{di}{dt} = -\beta\frac{SI}{N} - \gamma I$$

$$\frac{dr}{dt} = \gamma I$$

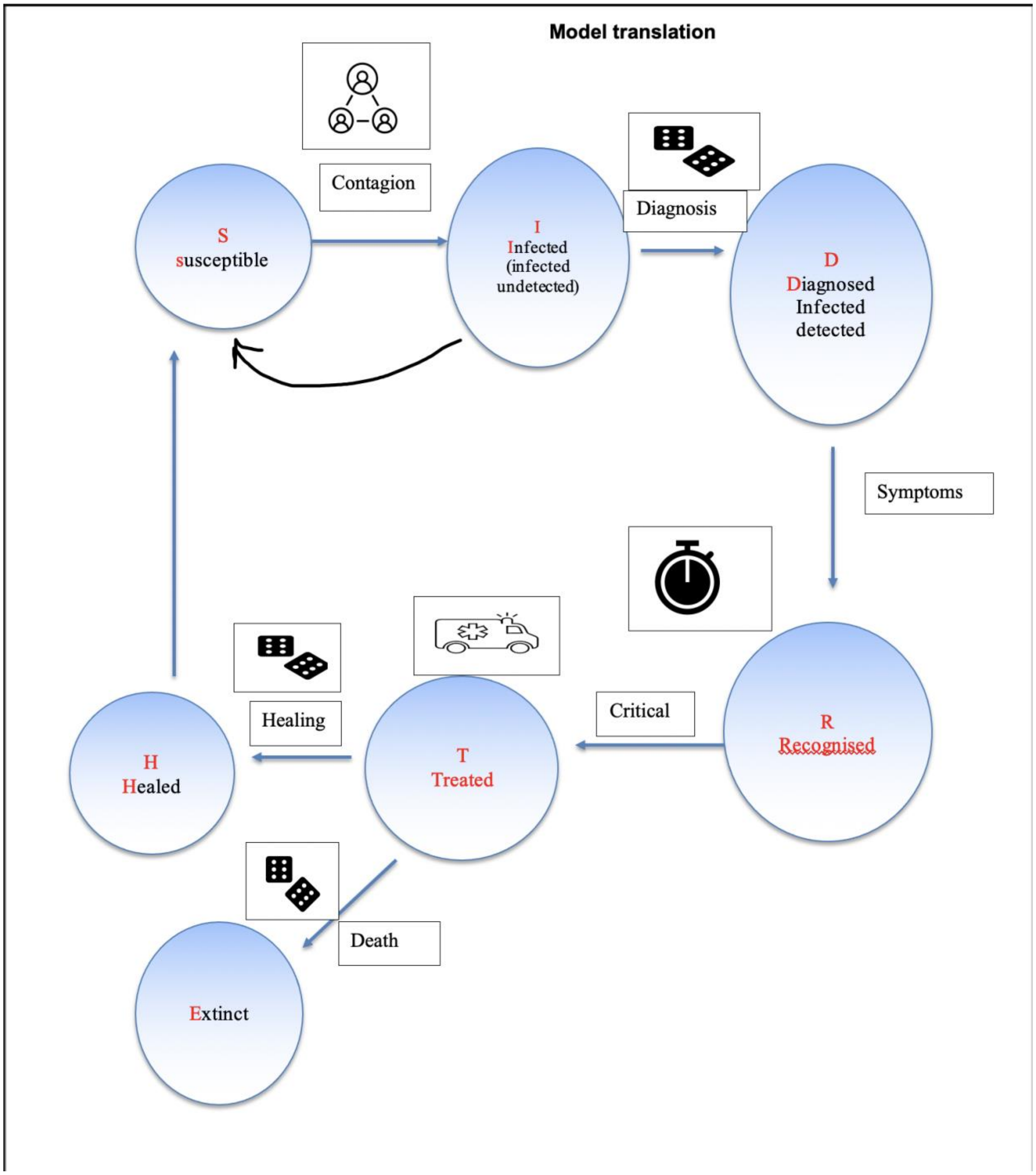
However, this greatly sampled model didn’t help me meet the requirements more. So, I have created a precise mode SIDARTHE model.

The population size is constant, and we can assume the S + I + D + A + R + T + H + E = 1 holds for all Times t.

The simulation consists the following compartments is composed: Susceptible, Infectious, Recovered and Deceased

- In the SIR model, the compartments are
- Susceptible (infectious persons),
- Infected (infected persons)
- Recovered: Removed (from the disease process “Removed" persons)
- Deceased: people dying from the disease

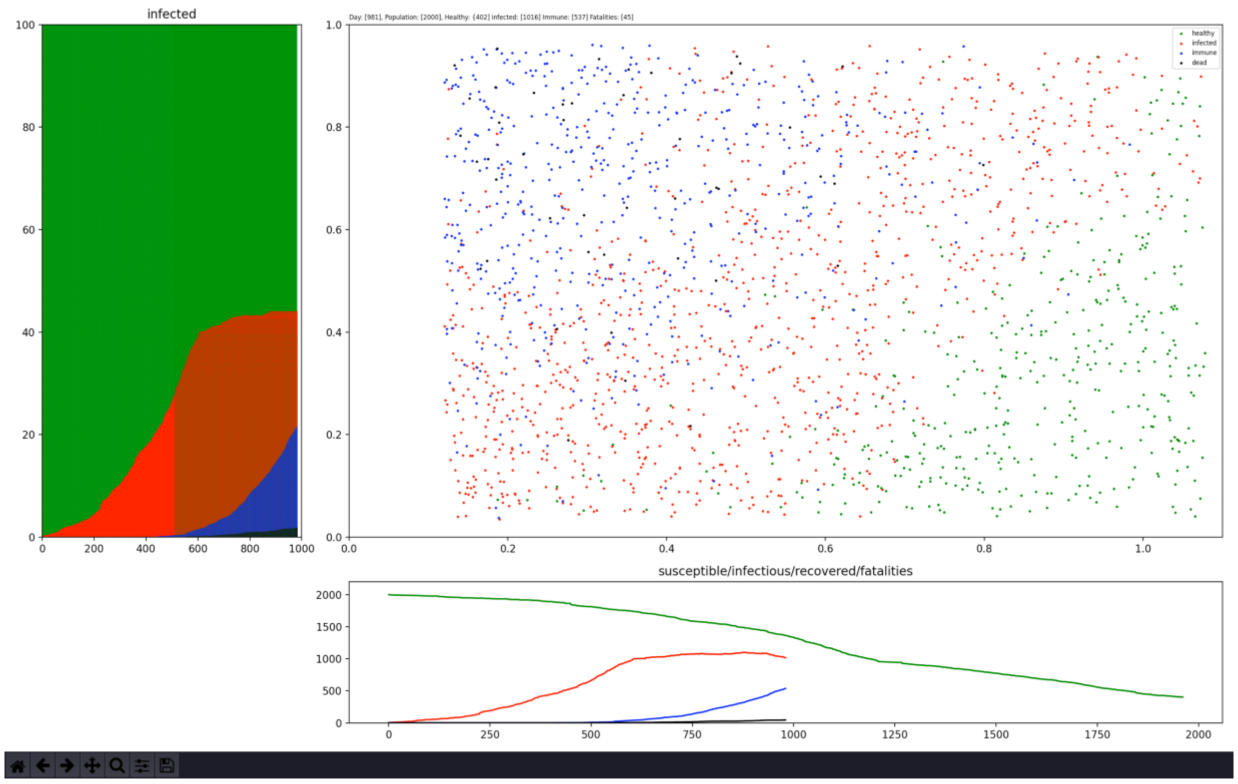
This model based on simulated individuals’ computer -aided modeling and the units have option of making decisions and taking action so our result from macro system level.


Model translation
Contagion
S
susceptible
I
Infected
(infected
undetected)
Diagnosis
D
Diagnosed
Infected
detected
Symptoms
R
Recognised
Critical
T
Treated
Healing
H
Healed
Death
Extinct

Infection Simulation. This simulation will have a small population randomly interacting

with each other. People staying within a bound would have a 5% chance of being infected and 2% chance of dying, as predicted by a uniform distribution, Gaussian distribution, and using the for live simulation funcAnimation in matplotlib.

The user will enter which the first day the disease will enter the simulation and will chase under which scenario the simulation will run under



The simulation run under 1000 day with 2000 population size and result in 1598 infected with 537 recovered and 45 died.

<u>Case of lock down</u>

The user will choose at which percent the lockdown will accrue such as after 75% of population got infected and the percentage of the people will be complying the lockdown rules such as 80% of people will lockdown after the COVID-19 spread in simulation.

<u>Case of self-isolation</u>

Using the continuous random variable and reducing the speed of each cycle and reduction the infection range:

Verification and validation

Steps that verify the model is by changing the scenario of the simulation from normal infection simulation to the lockdown simulation to the self-isolation simulation. Also, the simulation let the user to input which day of the simulation the first case shown and how much percentage of population get infected to run under the lockdown option and provide the chance to choose the percentage of population following the rules and obey the lockdown rules. The simulation provides the self-isolation scenario by reduce the interaction between the cycle.

To verify the model, use data from the past months of COVID-19 infection rates in a population, and apply them in the simulation. They should produce the same results seen in the past. To validate the model for the future in the real world, use the verified model of the past and apply it to predict future patterns. Of course, this would be verified again for the future sample once the future passes, to ensure reliability.

I would like to apply the simulation to the real world by working more on build hospital and housing so when the lockdown 6-8 cycle get together in one house in my bounder and if one didn’t obey the lockdown rules it might spread the Covid-19 in the house and each one of the cycles has already friend’s connection in different houses. Also, thinking about build a campus simulation that track the students ID location and campus and when a covid-19 case confirms in campus, the simulation will predict the students were withing the infection range of the student who got tested positive for the covid-19.

Results

1. Histogram representing infections with the “billiard ball model”.
    - Susceptible (infectious persons),
    - Infected (infected persons)
    - Recovered: Removed (from the disease process “Removed" persons)
    - Deceased: people dying from the disease
2. Line chart with multiples lines representing infections under normal infection simulation.
    - Susceptible (infectious persons)
    - Infected (infected persons)
    - Recovered: Removed (from the disease process “Removed" persons)
    - Deceased: people dying from the disease
    -
3. Histogram representing infections with the “billiard ball model” under lockdown scenario.
    - Susceptible (infectious persons),
    - Infected (infected persons)
    - Recovered: Removed (from the disease process “Removed" persons)
    - Deceased: people dying from the disease

4. Line chart with multiples lines representing infections under lockdown scenario infection simulation.
    - Susceptible (infectious persons)
    - Infected (infected persons)
    - Recovered: Removed (from the disease process “Removed" persons)
    - Deceased: people dying from the disease
5. Histogram representing infections with the “billiard ball model” under self-isolation.
        - Susceptible (infectious persons),
        - Infected (infected persons)
        - Recovered: Removed (from the disease process “Removed" persons)
        - Deceased: people dying from the disease

6. Line chart with multiples lines representing infections under self- isolation.
    - Susceptible (infectious persons)
    - Infected (infected persons)
    - Recovered: Removed (from the disease process “Removed" persons)
    - Deceased: people dying from the disease

References

Currie, C. S., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. S., Robertson, D. A., & Tako, A. A. (2020). How simulation modelling can help reduce the impact of COVID-19. Journal of Simulation, 1-15.

Dieckmann, P., Torgeirsen, K., Qvindesland, S. A., Thomas, L., Bushell, V., & Langli Ersdal, H. (2020). The use of simulation to prepare and improve responses to infectious disease outbreaks like COVID-19: practical tips and resources from Norway, Denmark, and the UK. Advances in Simulation, 5, 1-10.

Stevens, H. (2020, March 14). These simulations show how to flatten the coronavirus growth curve. Retrieved October 31, 2020, from https://www.washingtonpost.com/graphics/2020/world/corona-simulator/