

Program Structures and Algorithms

INFO 6205 FALL 2020

Team Project

Virus Simulation (SARS-CoV-2 vs Ebola)

Team Number: 9

Team Members:

Darasy Reth (Section 2)

Kshitij Verma (section 6)

Vedant Jain (section 6)

Introduction

Our task is to simulate the spread of a virus such as SARS-CoV-2, the pathogen behind COVID-19. The codes are written in Python3 which includes all the backend data analysis as well as function to visualize the simulation. Following things that are considered for building this simulator:

- The K and R factors of the disease
- The social distancing
- The population density
- The usage and effectiveness of the masks
- The prevalence of testing
- Barrier into the subject area using quarantine restriction.
- Immunity (born with immunity or from recover from the disease)

Aim of the Task

- This simple simulator is used to illustrate the effects of various social behaviors on the spread and impact of COVID19 and Ebola.
- The virus isn't just spread from human-to-human contact directly, but with many factors included and vary from one person to another such as ages, health conditions and etc. Our simulator was designed from the ground-up to study environmental transmission.
- We use natural human behaviors for random local movement to simulate how people come into contact with one another and transfer the disease using K and R values as well as social distancing and other factors.
- Unlike the model that use R factor to assume that R number of people will get infected if they stay close to the infectious person., we also use K factor and others to leverage the spread of the disease since not all infectious can infect R number of people in reality.
- We based our parameter values mostly on actual COVID-19 and Ebola statistics provided by the reliable source to study the simulation when applicable. Otherwise, we use the estimate based on our knowledge regarding the parameters for each disease.

Project Details

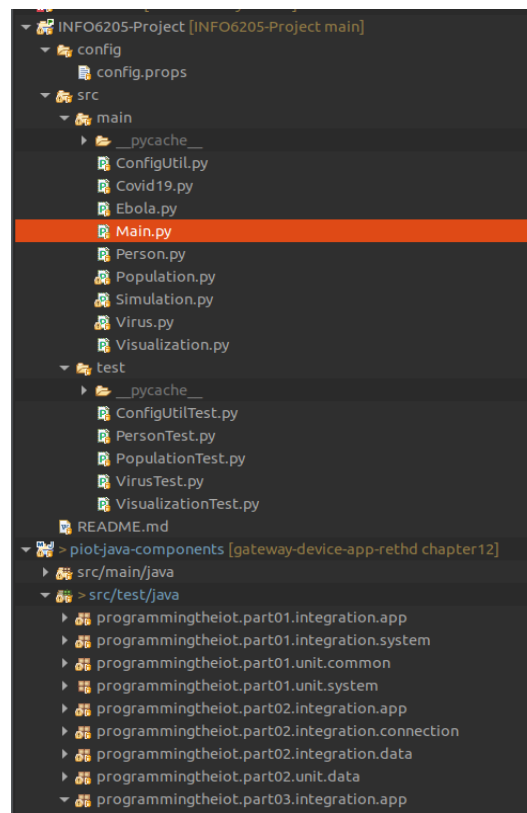
- We have used a simple mathematical model inspired by the idea of SIR model describing the structure of infectious disease but instead of simulating recovered, we will show information statistics of Susceptible, Infectious and Death (Removed).
- This project aims to deliver the outcome of the simulation by using the dataset of Covid-19 and another disease to make a comparison which we used the data of Ebola, which is another big outbreak mainly in Africa.
- **What is SIR?** Just a little background. The SIR model is one of the fundamental epidemiological models that illustrates the dynamics of an infective epidemic given that large population has already been susceptible, infected, and recovered (again, we will study on death/removed).
- In our simulation, assuming that the virus is generally contracted only once, an infected individual either pass away or recovers with an immunity.
- As the disease spreads, the susceptible population are likely to get infected (have never been infected), and the infected individuals tend to be removed (either die or recover).
- It presumes that the total number of the population is a constant number of **N**. Some the infected population will be put into the quarantine zone if this parameter is set to True within the *config.props* file. Otherwise, the infected population will randomly move and infect the surrounding people within the range less than social distance.
- We have considered parameters such as following for making a better simulation model:
 - ✓ Random initial position for each person for modeling.
 - ✓ Death Factor of Covid-19 and Ebola for people from different age groups. **[1]**
 - ✓ Calculated Recovery Rate factor of people from different age groups.
 - ✓ Calculated chances of a person getting affected by the virus on basis of age.
 - ✓ Chances that infected people get tested and put into a quarantine zone.
 - ✓ R factors value based on K factor of both diseases.
 - ✓ Social distancing between each object.
 - ✓ Chance a person wears a mask.
 - ✓ Chances a person has an immunity against the disease.
- The model divides a homogeneous and isolated population into the following three categories: susceptible, infectious, removed.
 - ✓ Susceptible: a group of people who are vulnerable to exposure with infectious people as well as immune population. They can be patient when the infection happens.
 - ✓ Infected: These have already contracted the disease.
 - ✓ Removed: The person who dies from the disease.
- Our model representing a framework describing how the percentage of population in each group can change over time.

Project Implementation

We have implemented this simulation using Python3 with the usage of Object-Oriented Design along with some understanding and estimation using Probability and Statistic in mind. The project is divided into 3 main directories: Config, main, and test. “**Config**” folder contains the configuration file that we use to extract some of the data we collected from various reliable source to support with our disease simulation investigation. “**main**” folder contains all the necessary Python class that are used to simulate the spread of the virus. The functions that include within main directory including **ConfigUtil.py** (configuration class to read data from config.props), **Virus.py** (a class that define a virus related instances and functions such as determine if the person is infected, recovered or dead), **Ebola.py** (a child class of Virus.py), **Covid19.py** (a child class of Virus.py), **Person.py** (a class representation of people instances and random movement), **Population.py** (a class representation of a group of Person.py and contains functions to determine how an infectious person infects others, update every person’s movement and generate various person characteristic necessary to run the simulation such as mask wear and health condition), and **Visualization.py** (a class that helps to visualize the data of the results running the simulation). “**test**” folder contains Unit tests that are used for testing methods for this project.

To run the simulation, run the python class named “Main.py” which runs the simulation for both Covid-19 and Ebola using 300 population over 60 days period (can be adjusted).

The following screenshot representing the simulation project directories:



1) The K and R factors of the disease

R and K factors are considered in modeling the simulator where we have taken R numbers = 3.0 [2] for Covid-19 based on the common value we have seen on different source. And for Ebola R number = 1.5 (where 1.8 in Congo and 1.3 in Uganda) [3]. Since we cannot find any mathematical model to compute R based on K, we have used an estimation based on we have read where when K is low (≈ 0.1) the value of R can get really low (≈ 0) or high (≈ 17) and when the value of K is higher or equal to 1.0, R seems to remain the same. For Covid-19, k is somewhat low around 0.1 [4] while, for Ebola, k is estimated to 1 due to its widespread comparing to Covid-19.

To estimate R, we estimate via

If $k < 1$, $R = R * R^{(1-k)}$

Else, R remains the same

Within population.py

```
141 def updateSurroundNeighbors(self, infectedPerson):
142     """
143     A method to update health status of all the neighbors close to the infected person in less than social distancing range
144     """
145     r = 0
146     r_factor = self.virus.r_factor
147     if self.virus.k_factor < 1:
148         r_factor = random.randint(0, int(r_factor*(r_factor ** (1- self.virus.r_factor))))
149     for p in self.population:
```

2) Social Distancing

We can find out whether people get infected or not, by checking the closest neighbors to infectious person within social distancing range. In this formula, we will use R values which is determined by the rule above. If R number of healthy people are less than social distance range to the infectious person, all R number of people will get infected. Otherwise, R number of healthy people will get infected and the remaining will be randomly set to get infected or not based on their ages, health status and etc. Also, the social distance range is set to be the same between Covid-19 and Ebola as they share the same recommend range. [4] To compute the distance between two people with (x, y) as their location, we simply use Euclidean Distance:

$$\text{Distance} = ((x_1 - x_2)^2 + (y_1 - y_2)^2)^{0.5}$$

(x1, y1): location of person 1, (x2, y2): location of person 2

Within population.py

```
170 def distanceBetweenTwoIndividual(self, personA, personB):
171     """
172     A method to compute the distance between 2 people
173     """
174     A = personA.getLocation()
175     B = personB.getLocation()
176     # Using Euclidean Distance
177     distance = (A - B)**2
178     distance = np.sum(distance)
179     distance = np.sqrt(distance)
180     return distance
```

```

140 def updateSurroundNeighbors(self, infectedPerson):
141     """
142     A method to update health status of all the neighbors close to the infected person in less than social distancing range
143     """
144     r = 0
145     r_factor = self.virus.r_factor
146     if self.virus.k_factor < 1:
147         r_factor = random.randint(0, int(r_factor*(r_factor ** (1- self.virus.r_factor))))
148     for p in self.population:
149         neighbor = self.getPerson(p)
150         if self.distanceBetweenTwoIndividual(infectedPerson, neighbor) < self.virus.getSocialDistancing():
151             if (neighbor.isInfected() == False) and neighbor.isAlive() and (not neighbor.isImmune()):
152                 neighborGetInfected = False
153                 # Apply K and R factor
154                 # If the number of neighbors is less than R, they all get infected
155                 if r < r_factor and (not neighbor.isImmune()):
156                     neighborGetInfected = True
157                 # Else, determine based on their chance of being infected
158             else: neighborGetInfected = self.virus.checkIfInfected(neighbor)
159             # If infected, updated their health status
160             if neighborGetInfected: # If get infected
161                 self.population[p].setHealthStatus(2) # Value 2 = Infected (0: Healthy, 1: Immune, 2: Infected, 3: Death)
162                 self.population[p].setInfectedStatus(True)
163                 self.infectedPopulation[p] = neighbor
164                 self.population[p].incrementDaysOfInfected()
165                 # A random function to determine if the person get tested when found to be infected
166                 self.population[p].setTestedStatus(self.assignGetTestedStatus())
167

```

3) The usage and effectiveness of masks

Another factor that each person object has is the usage of mask. Each person will be assigned whether they use the mask or not based on the estimates and how much it is effective against the disease in general as there are different type of masks in the market and they provide different effectiveness against the disease (ex. N95 can protect up to 95%, while cloth mask might be able to protect for about 65% [6] which is the value we use).

To randomly assign whether people wear mask or not, we use random select with weight bias assign to True or False where the mask usage percentage is weighted to True and 1 – mask usage percentage is weighted on False.

To randomly determine whether people who wear mask get infected or not, we use random select with weight bias assign to True or False where the mask effectiveness percentage is weighted to False and 1 – mask effectiveness percentage is weighted on True.

Within population.py

```

259
260 def assignWearMaskStatus(self):
261     """
262     A random function to determine if the person wear a mask
263     """
264     # |
265     return random.choices([True, False], (self.maskUsage, 1 - self.maskUsage))[0]
266

```

4) The prevalence of testing

Another factor is whether people who are infected get tested or not. We based on this value on the data that we have collected regarding the total number of tests taken and total number of populations. So, if the people who are infected, we will check if he/she also get tested. Similar to the previous estimation, we determine whether people get tested or not based the probability of getting tested using the data from Worldometer. [6]

To randomly assign whether people stay in quarantine zone or not, we use random select with weight bias assign to True or False where quarantine strictness percentage is weighted to True and 1 – quarantine strictness percentage is weighted on False.

Within population.py

```
252
253 def assignGetTestedStatus(self):
254     """
255     A random function to determine if the person get tested when found to be infected
256     """
257     #
258     return random.choices([True, False], (self.numberOfTested, 1 - self.numberOfTested))[0]
259
```

5) Quarantine into the subject area

Another factor is whether people who are infected get tested or not. We based on this value on the data that we have collected regarding the total number of tests taken and total number of populations. So, if the people who are infected and he/she also get tested, we will determine whether they are put into the quarantine zone or not based on the chance that people will stay under quarantine since not everyone will follow the quarantine rule. Similar to the previous estimation, we determine whether people stay in quarantine or not based the probability of staying in quarantine.

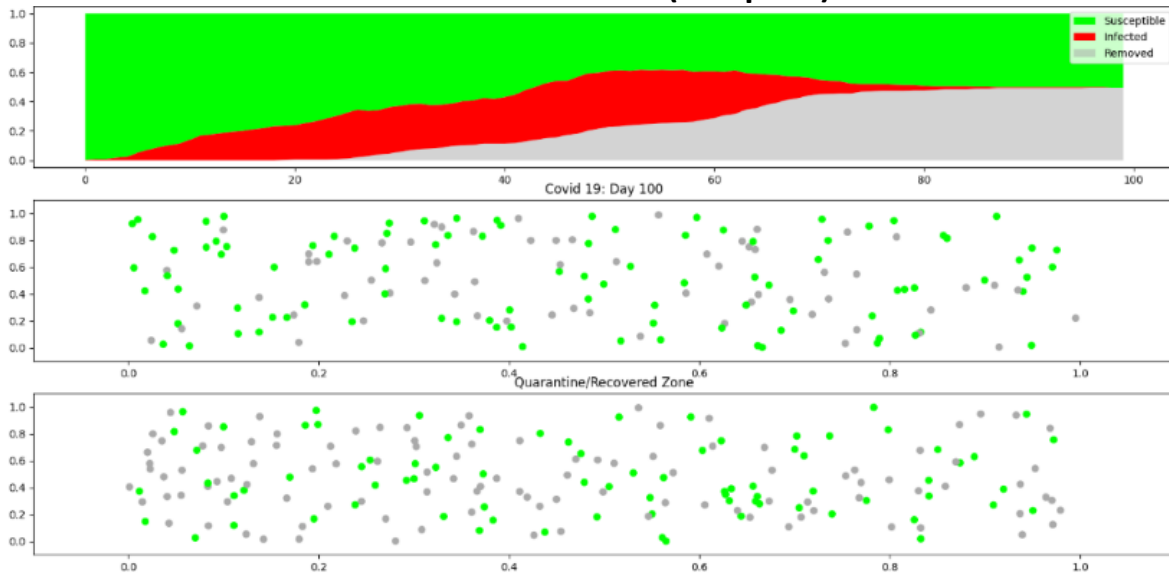
To randomly assign whether people stay in quarantine zone or not, we use random select with weight bias assign to True or False where quarantine strictness percentage is weighted to True and $1 - \text{quarantine strictness percentage}$ is weighted on False.

Within Main.py

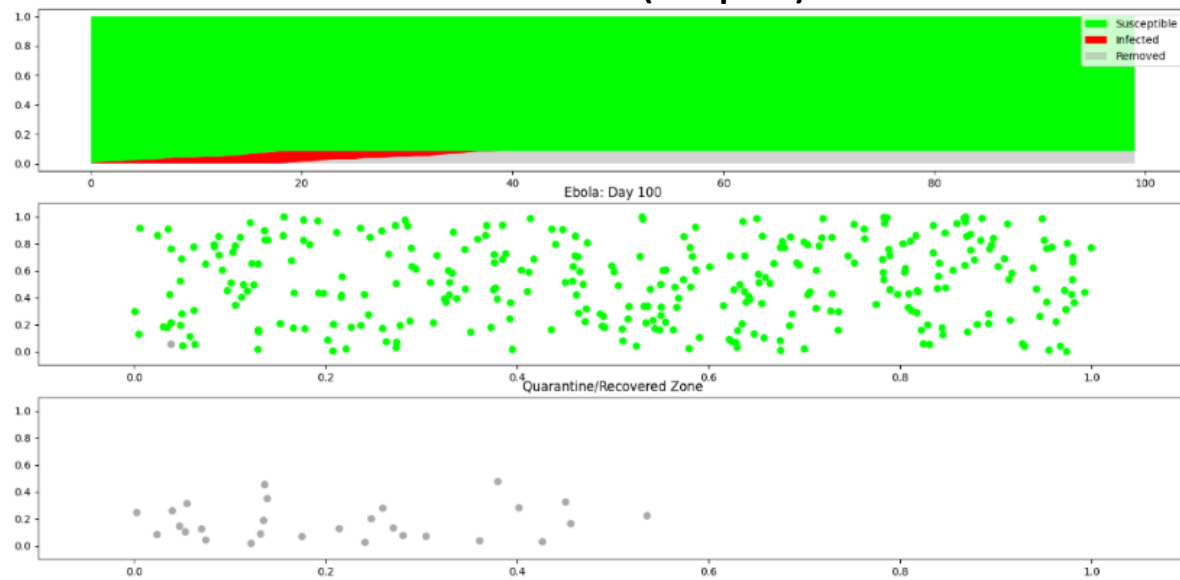
```
55
56     population.updateMap()
57
58     if quarantineRestriction:
59         tempSet = population.getInfectedPopulation().copy()
60         for infectedPerson in tempSet:
61             # Assume 10% chance the person who tested are not put into quarantine zone
62             if (virus.randomBooleanWeightSelect(0.9, 0.1)) and (population.getPerson(infectedPerson).getTestedStatus() == True):
63                 quarantineZone.addPerson(population.getPerson(infectedPerson))
64                 population.removePerson(infectedPerson)
65         quarantineZone.updateMap()
66
67
```


Output

Covid-19 Simulation (Sample 1)



Ebola Simulation (Sample 1)



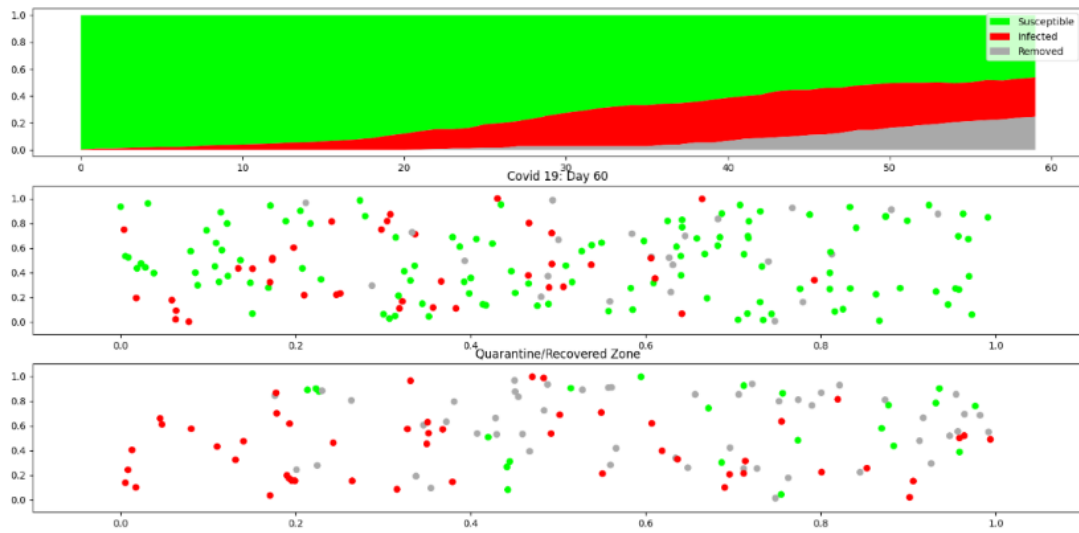
Results of both from console

[illegible]

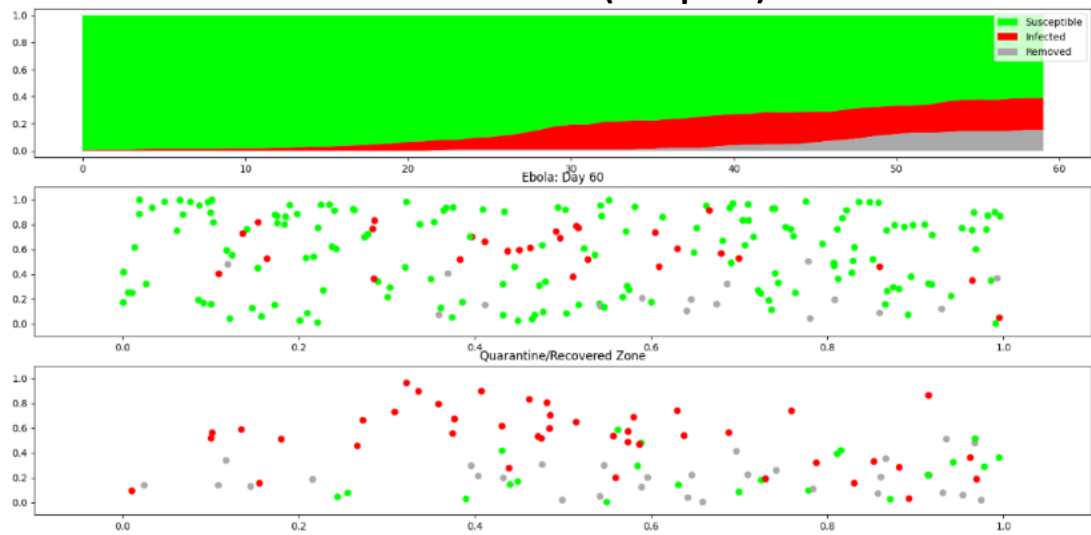
```
<terminated> Main.py [/home/darasy/git/venv/bin/python3.8]
```

[illegible]

Covid-19 Simulation (Sample 2)



Ebola Simulation (Sample 2)



Mathematical Analysis and Evidence

In this simulation, we have selected different computing methods for each factor as they are randomly assigned and unique. We have found that the number of confirmed cases, recovery, dead and healthy population are indeed related to many of the factors listed previously and below such as K and R values, mask usage and quarantine. The analysis and evidence are provided as below which show how each factor affects the results in general if anyone of them is not well followed:

1) The k and R factors of the disease:

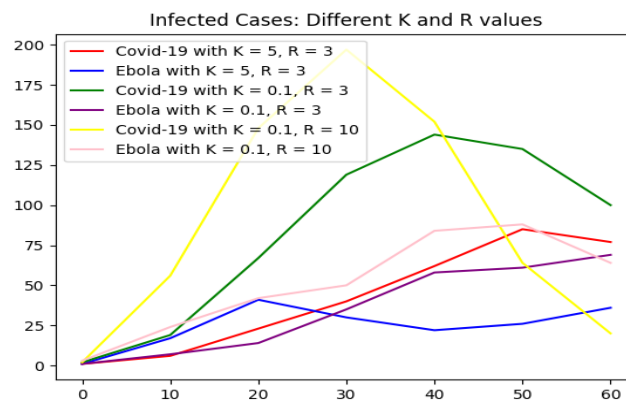
For this factor, R represents the number of people that one infected person can infect R number of people. However, this is not always true as one infected person can affect from nobody to more than R number of people. This is when K factor comes into consideration as this factor can range from very small value such as 0.1 up to infinity. Low K value represents that one infectious person can infect nobody or infect a large group of people while higher value of K means that one infected person can infect R number of people. However, there is no model or real explanation regarding the relation between K and R. Therefore, I have based my computation regarding R using K purely on estimation. From the article [3], Covid-19 have the value of R between 2 and 3 where K is somewhat low around 0.1 which means 1 infected person can infect around 15 other people. Based on these representations, I come up with the estimation using the formula as below:

To estimate R, we estimate via

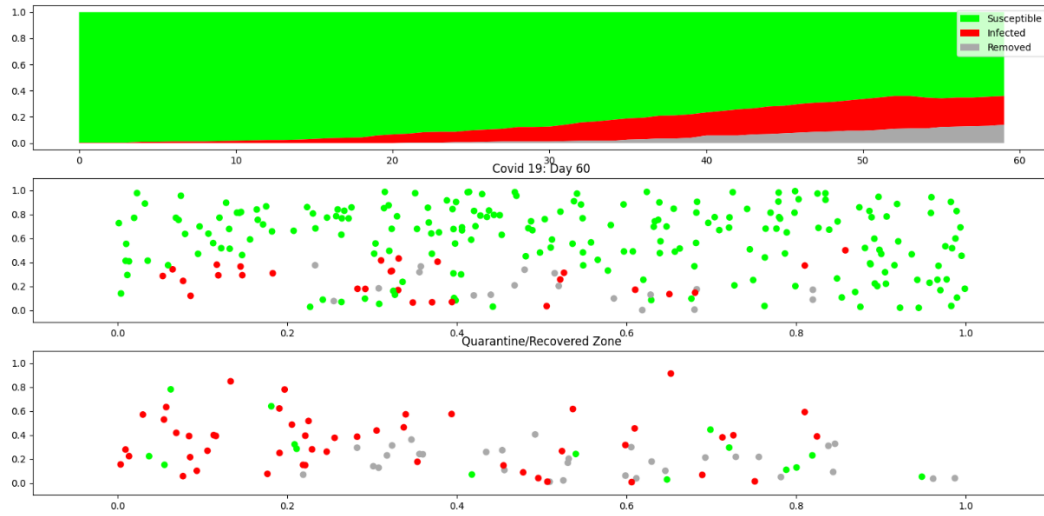
If $k < 1$, $R = R * R^{(1-k)}$

Else, R remains the same

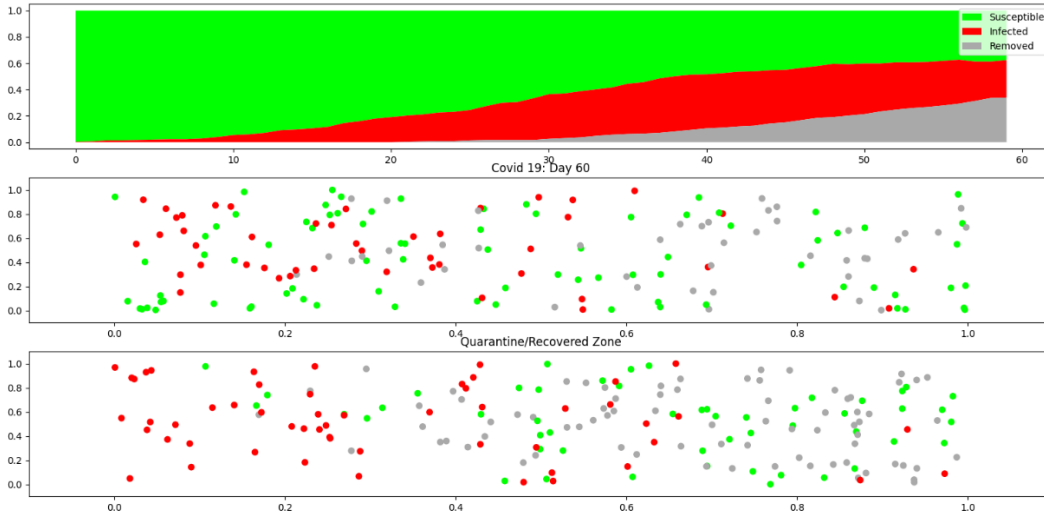
Based on my observation, K and R do have a strong relationship as the value of K and provides a variance to the value R. By that, if the value of K is low (~ 0.1), it will provide a high variance to the value of R. For Covid-19, where R is between 2.0 and 3.0, when $K = 0.1$, it indicates that one infected person can infects either no one or a large group of people such as in the house party or inside the subway train. When K is high, on the other hand, R variance will be low, which means 1 infected person can infect around R surrounding people. So if K is low and R is high, we can expect to see a large infection and death rate. As shown in the graph below, when $K = 5.0$ and $R = 3.0$, the infection and death rate does not grow much comparing to $K = 0.1$ and $R = 3.0$. And if we look at when $K = 0.1$ and $R = 10.0$, as expected, the infected and death rate grow quite fast and high.



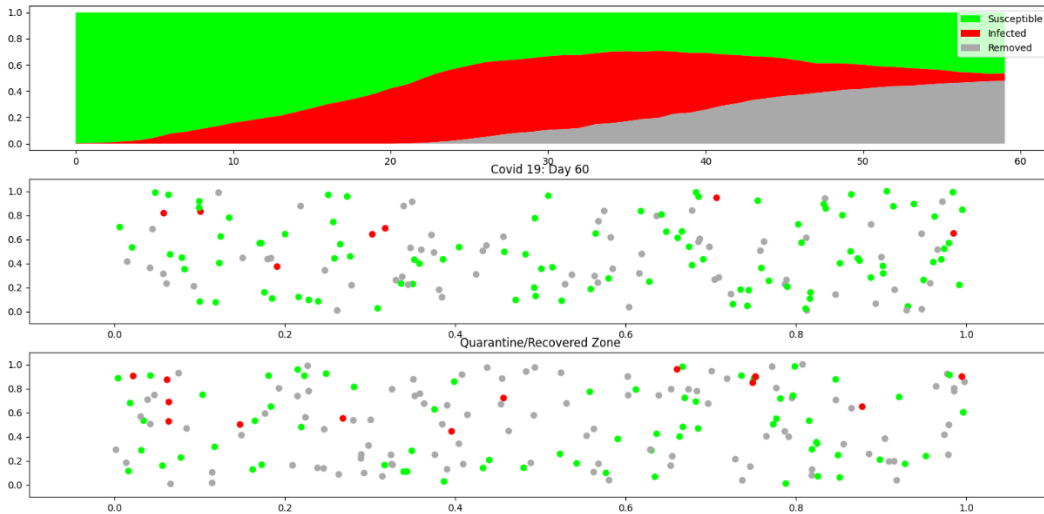
Covid-19 with $K = 5.0$, $R = 3.0$



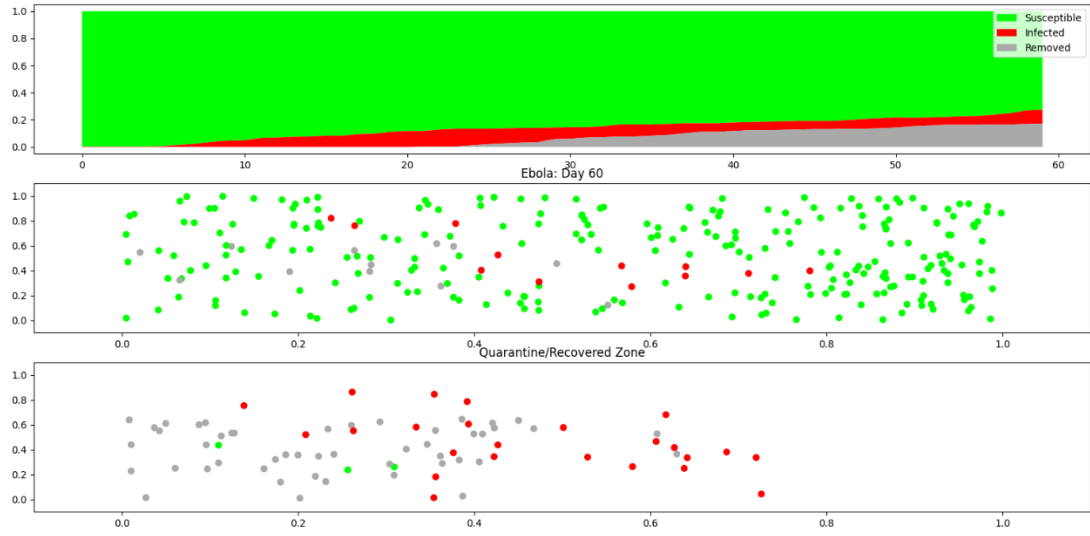
Covid-19 with $K = 0.1$, $R = 3.0$



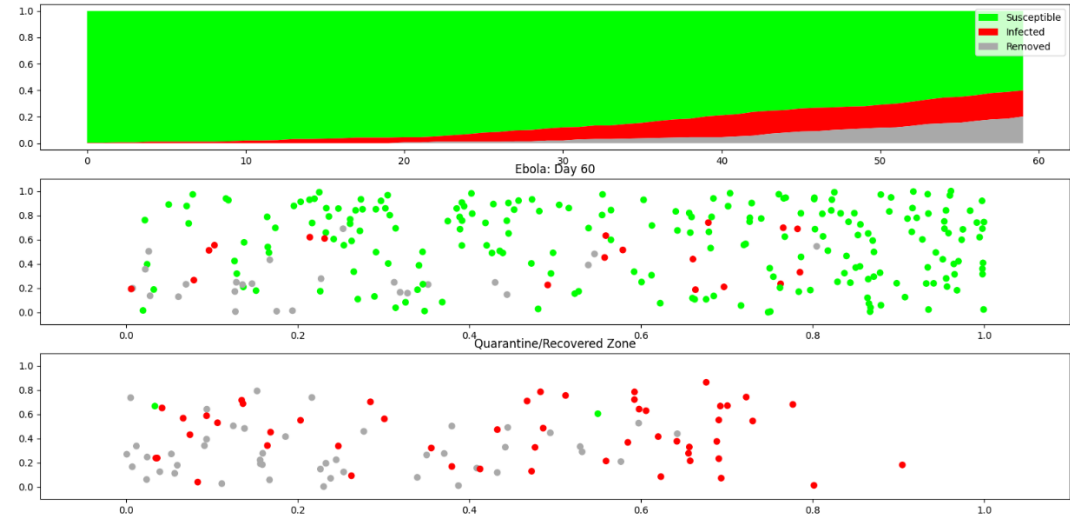
Covid-19 with $K = 0.1$, $R = 10.0$



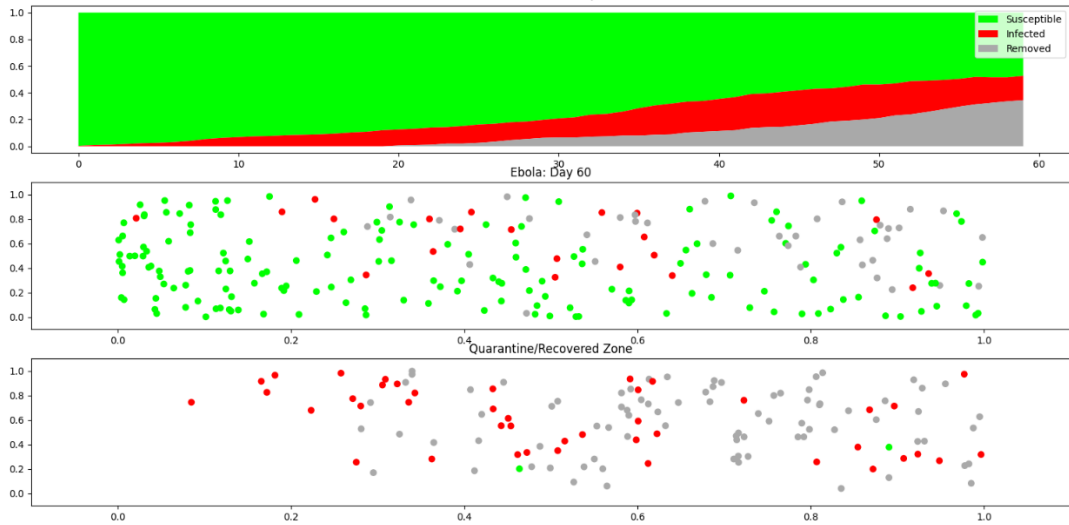
Ebola with $K = 5.0$, $R = 3.0$



Ebola with $K = 0.1$, $R = 3.0$



Ebola with $K = 0.1$, $R = 10.0$



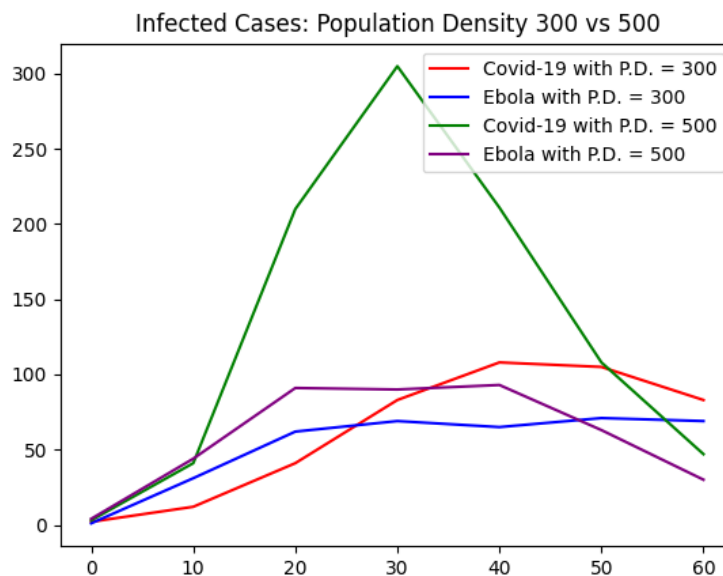
2) Population Density and Social Distancing

Social distancing plays an important role in reducing the infection rate. From what we have studied, within the area that has many populations, the number of infection rate can go up quite high, where the area with less crowds tends to have a lower infection rate. This is because once one person gets infected, it is easy for him/her to spread the disease in a crowded area as there are high chances many people are around the infected patient, where in the less crowded area, such case is less likely to occur. Based on the computation using the Euclidean that we have implemented, we have found that social distancing is really in reducing the spread of both Covid-19 and Ebola diseases. Since they both share the same social distancing range, we use the same threshold for both Covid-19 and Ebola.

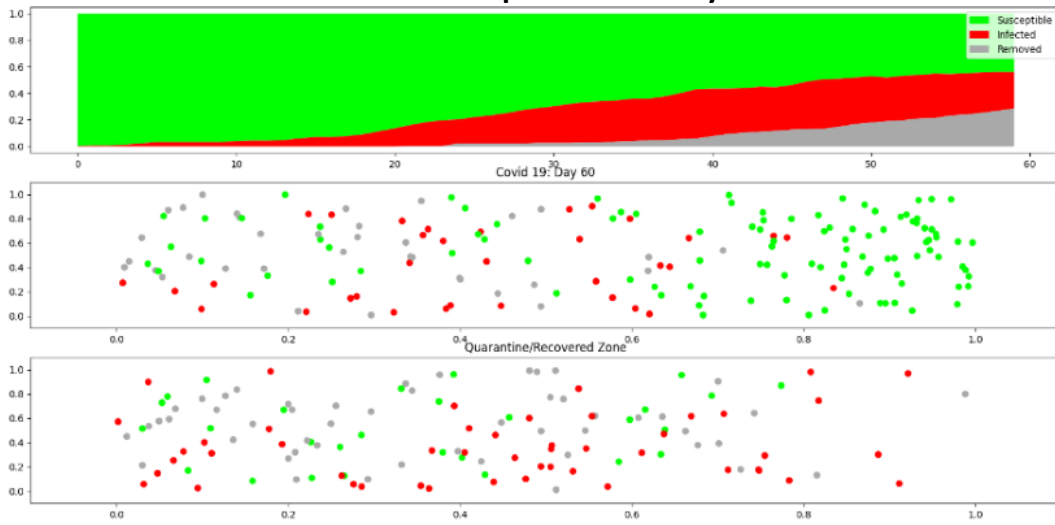
$$\text{Distance} = ((x1 - x2)^2 + (y1 - y2)^2)^{0.5}$$

(x1, y1): location of person 1, (x2, y2): location of person 2

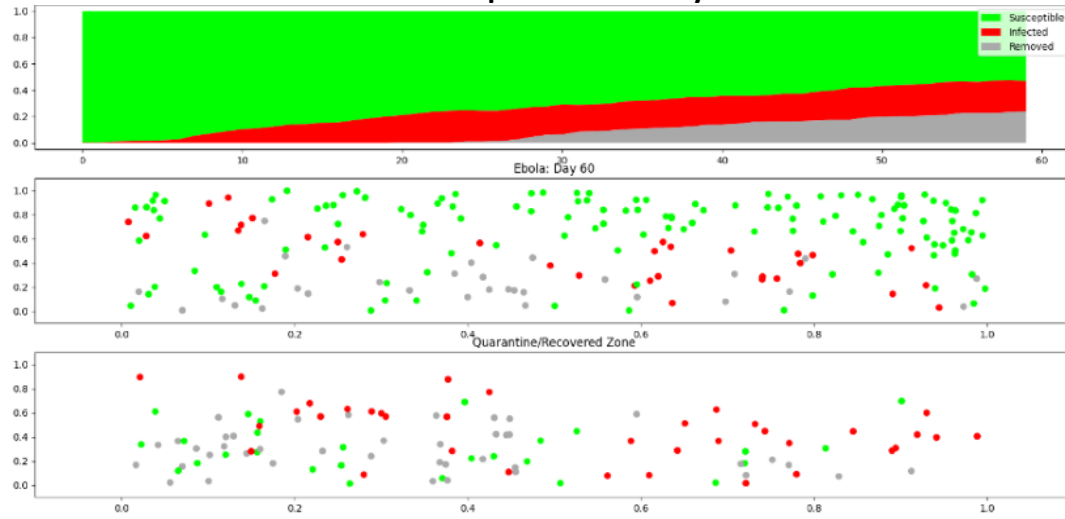
As we can see from the graph below, for Covid-19, when the population density remains around 300, the spike is far below comparing to when the population density = 500. For Ebola with its lower infection rate, while the number does increase, it's not too far different between when population density = 300 to when the population density = 500. This shows that Covid-19 provides a significant impact to the crowded group as it spreads its disease quite fast and too many people who do not maintain social distancing.



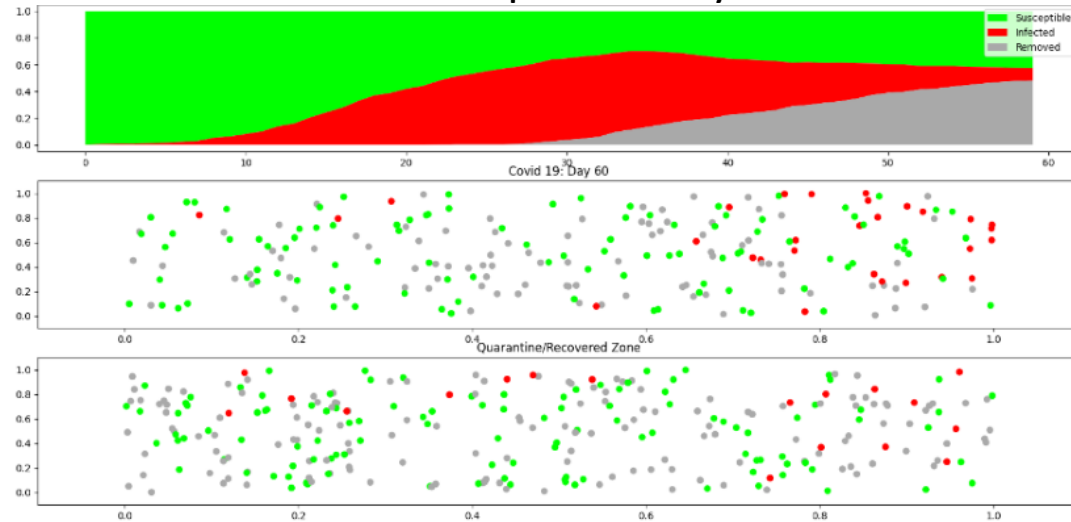
Covid-19 with Population Density = 300



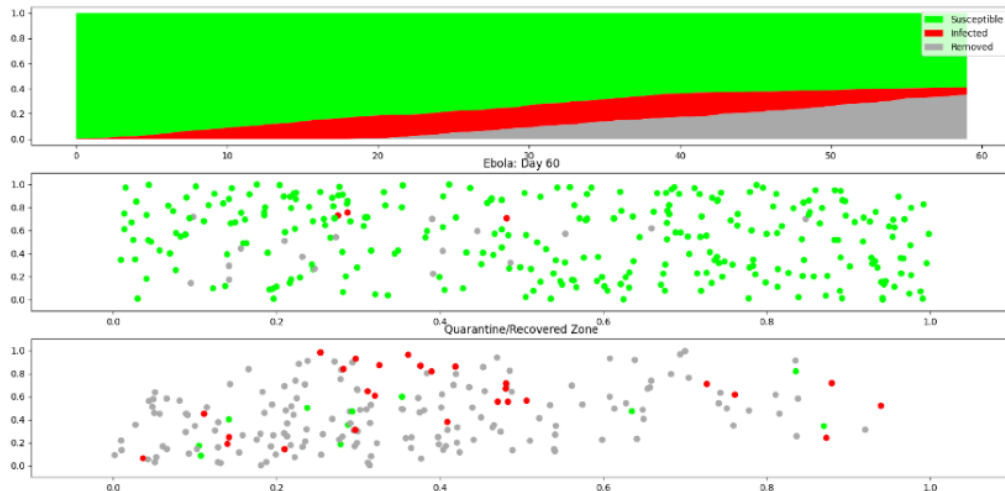
Ebola with Population Density = 300



Covid-19 with Population Density = 500



Ebola with Population Density = 500



3) The usage and effectiveness of masks

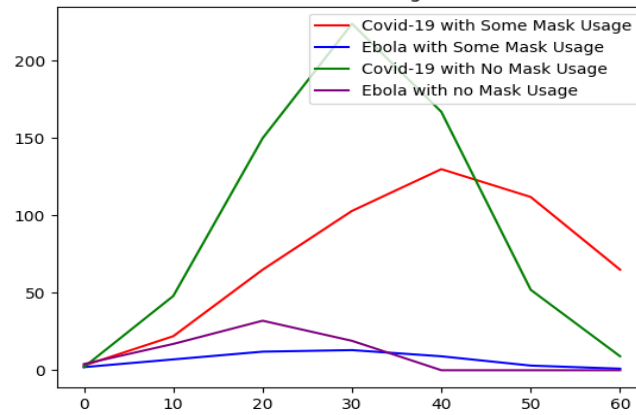
Masking is greatly correlated with lower daily growth or strong reduction from peak growth of COVID-19, mainly R factor. [2] Virus prevention rate is related to type of mask, for instance, N90 masks, prevention rate or efficacy is 90% while N95 masks offer 95% of prevention rate. However, a normal cloth masks are less effective but provide more protection than without wearing mask at all. From our finding, cloth masks provide somewhat 65% protection [6] which is twice safer than when without wearing a mask. With this probability, we have used random method to randomly assign whether an individual get infected or not by assigning weight of 65% to False and 35% to True when wearing, while without wearing mask, the person can get infected if he/she does not have an immune system. This is applied to both Covid-19 and Ebola.

Model to determine if people get infected when close to sick person:

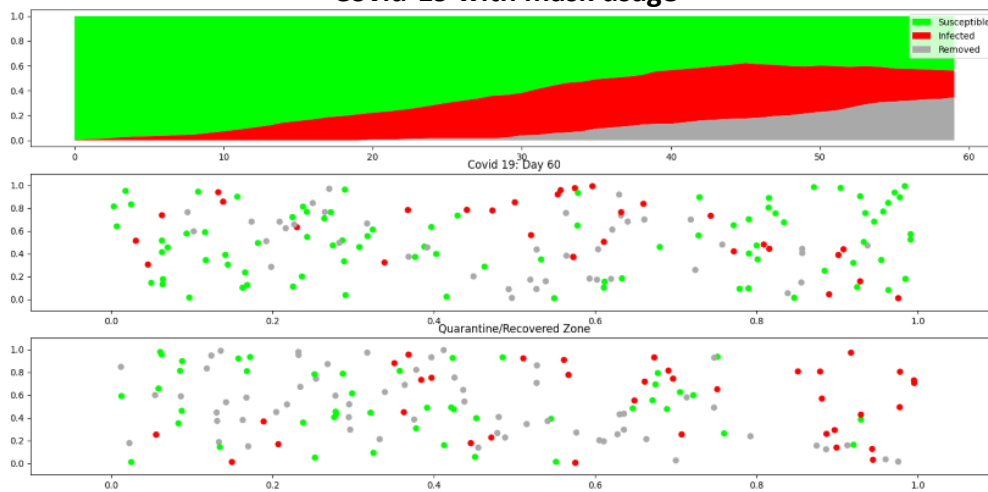
`random.choices([True, False], (0.35, 0.65))`

From what we have seen from the graph below, we can see that, for Covid-19, when everyone does not use mask, the number of infected cases go up high comparing to when some people use the mask. The difference is more than 50% higher when people does not use mask than some population use mask. For Ebola, while the result is not too much different, the number of infected cases goes up when some people wear masks comparing to when nobody use at all. While masks are important for both Covid-19 and Ebola outbreaks to reduce the infected cases, the result indicates that masks are very crucial in Covid-19 outbreak more than in Ebola.

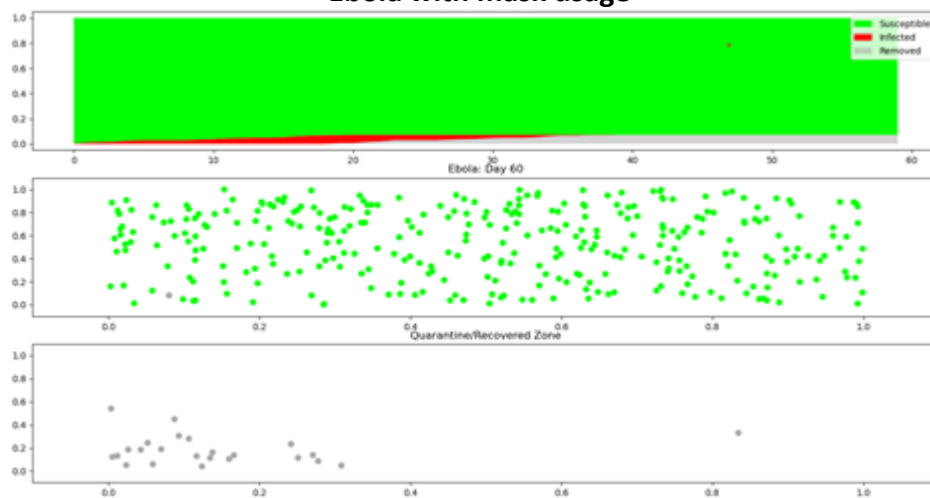
Infected Cases: Some Mask Usage vs No Mask At All



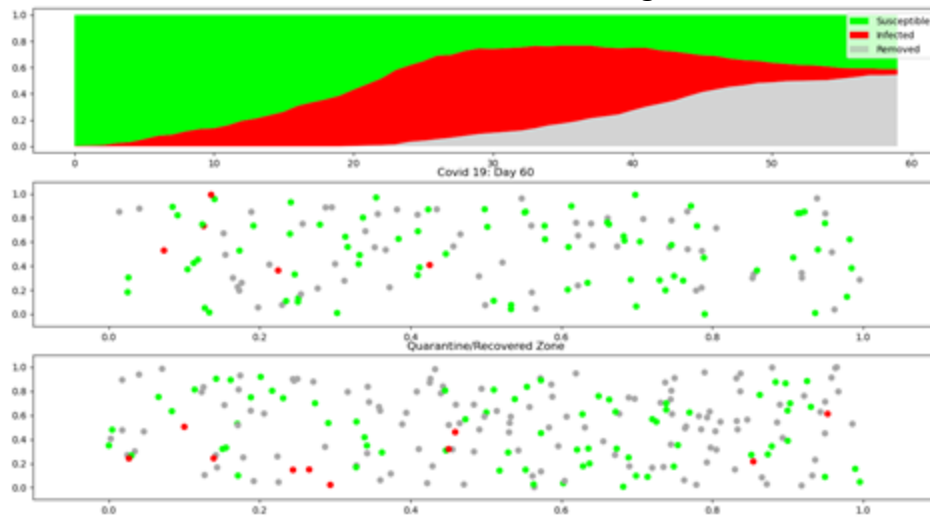
Covid-19 with mask usage



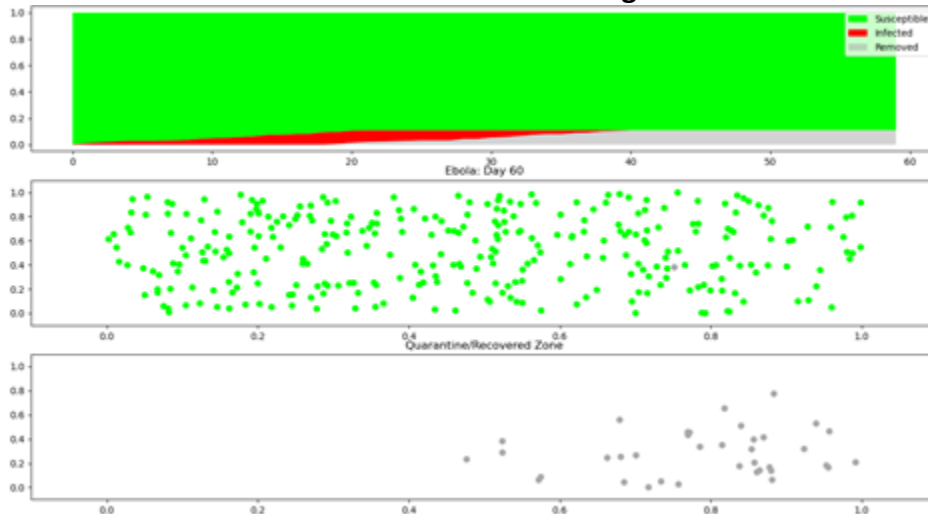
Ebola with mask usage



Covid-19 without mask usage



Ebola without mask usage

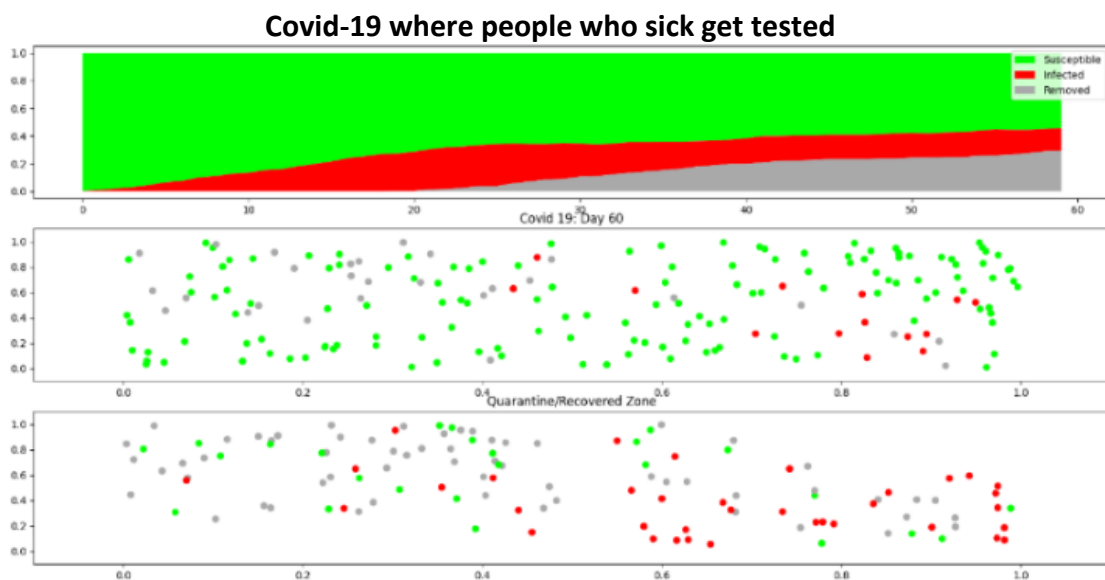
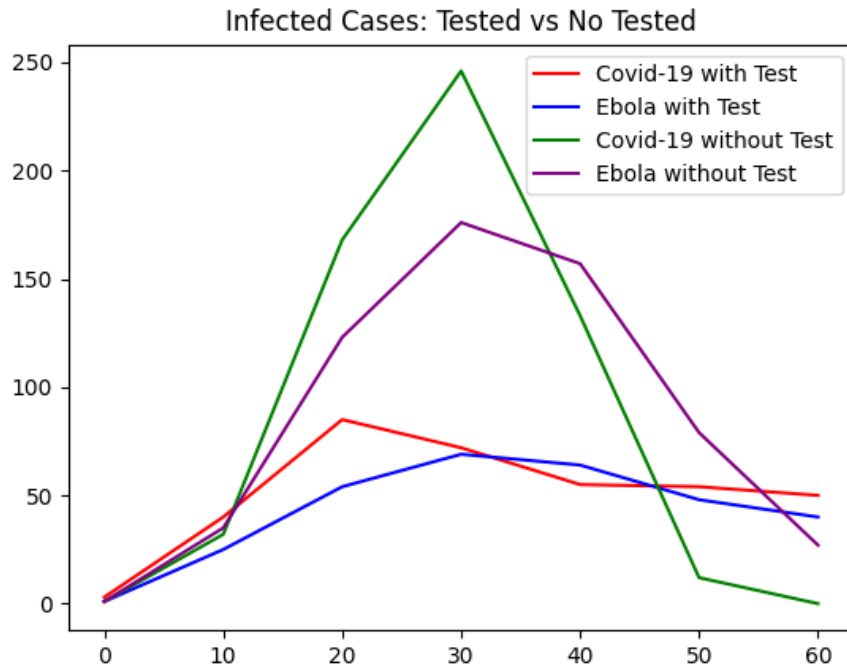


4) The Prevalence of Testing

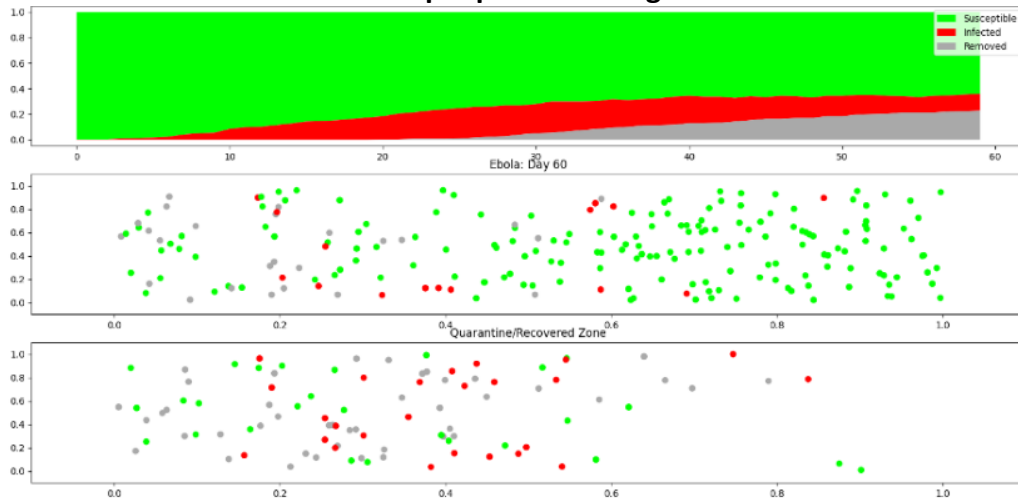
Another factor that we have taken into consideration is whether an individual who are infected with the disease get tested or not. Because if the people are aware that he/she is sick with the disease, it is possible that he/she can stay under quarantine and reduce the spread. Otherwise, the infected person can roam around the area and infect other people which we have seen in many cases. Luckily, we have found the data from Worldometer [7] that provides us with the data as to how many populations have taken the tests that provides us with some estimations to predict whether an infected individual get tested for the virus. From the time we are writing this, there are 205,061,072 tests conducted out of 331,849,210 population which is about 62%. Then, to predict whether an infected individual get tested, we use random select method with weight 62% given to True and the remaining 38% given to False for Covid-19 where for Ebola the number is about 80% for tested and 20% for not tested.

Model to determine if people who sick get tested:
random.choices([True, False], (Weight of Tested, Weight of Tested))

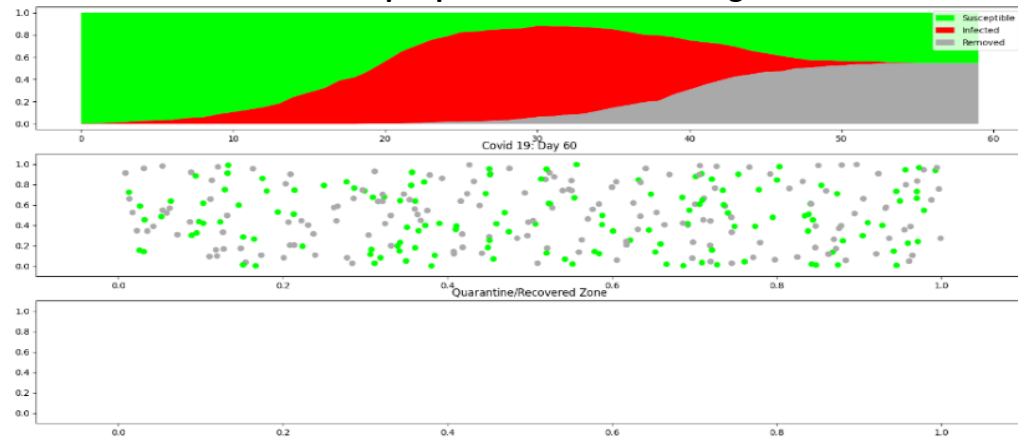
Base on the graph shown below, we can see that testing is quite important. When people get sick and they get tested to know that result, they can alert themselves to stay under quarantine or reduce contact with their beloved ones. Otherwise, as we can the number of cases for both diseases will increase significantly as the patient roaming around the area without knowing they are infected and contagious to the surrounding people. From the graph below, for Covid-19, the number of cases increase more than 200% where for Ebola, the number increases for about 100%.



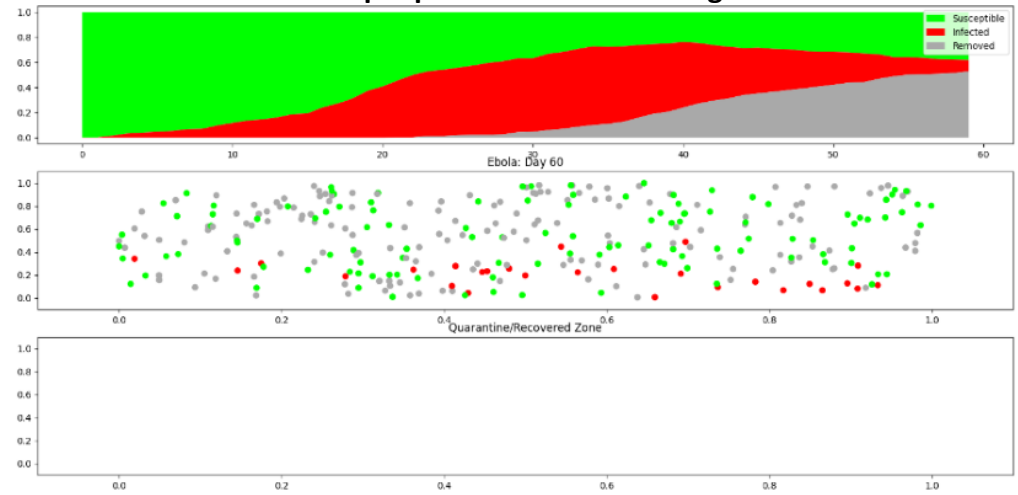
Ebola where people who sick get tested



Covid-19 where people who sick does not get tested



Ebola where people who sick does not get tested



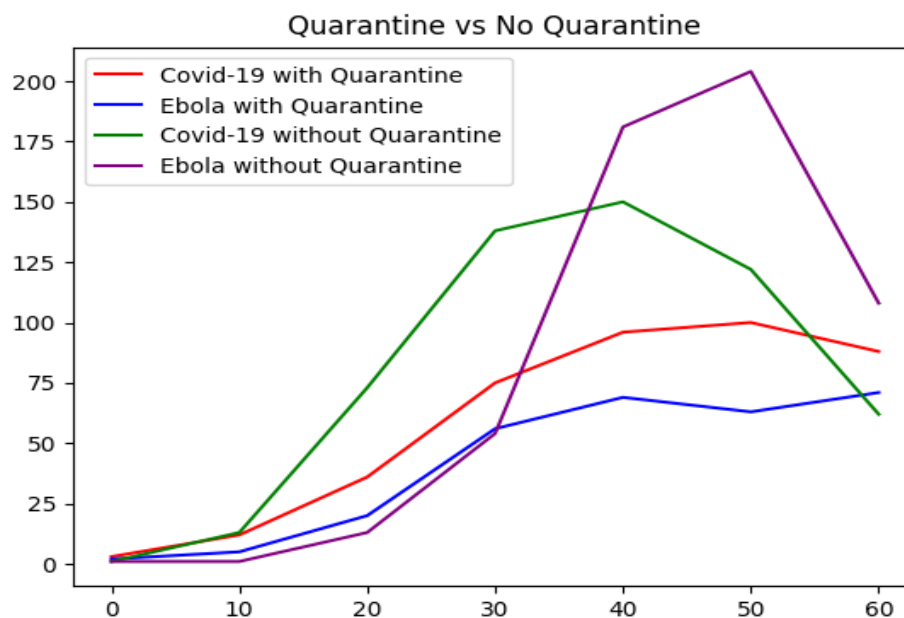
5) Quarantine into the Subject Area

Quarantining can help in reducing number of people getting infected. When someone gets infected, he/she can spread the disease easily via respiratory unintended especially in the close spaces such as subway, elevator, grocery stores, etc. However, with quarantine restriction in effect and if many people follow it well, it can help to reduce the disease significantly as the sick individual do not make any contact with anyone that avoid a chance of the disease transmission. From our model, we have estimated that 85% of people who are infected have been followed the quarantine protocol. [8] With that percentage, we have used random function to determine whether people stay under quarantine or not with the weight of 85% given to True and 15% to False.

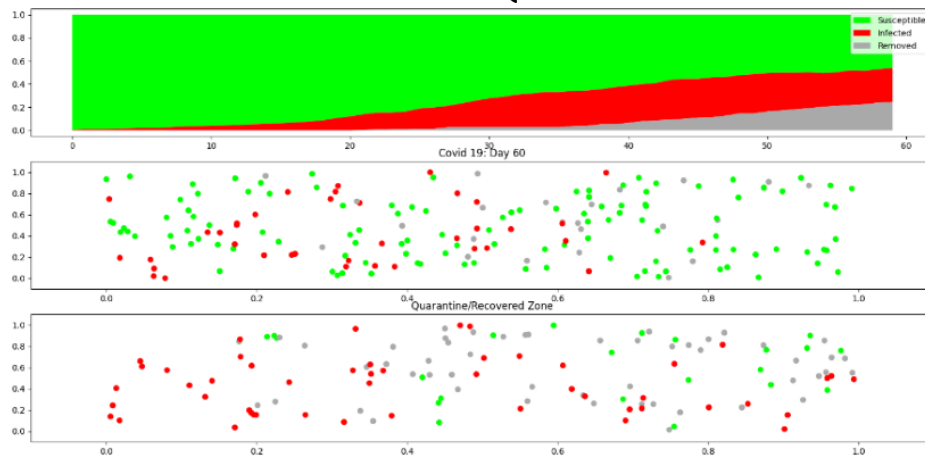
Model to determine if people stay under quarantine:

`random.choices([True, False], (0.85, 15))`

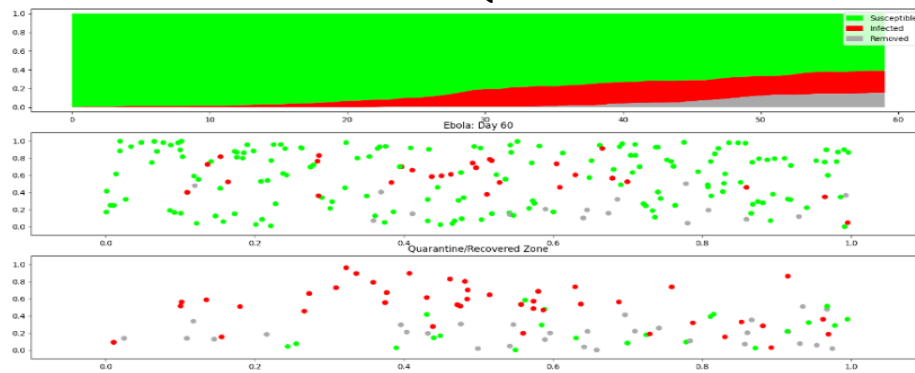
As can be seen from the result we have gotten from our experiments below, when there is no quarantine, the number of infected cases goes quite high comparing to when people follow quarantine rules in some degrees. We can see that for Covid-19, the number rises for about 50% more in their corresponding peaks, while increase more than 100%.



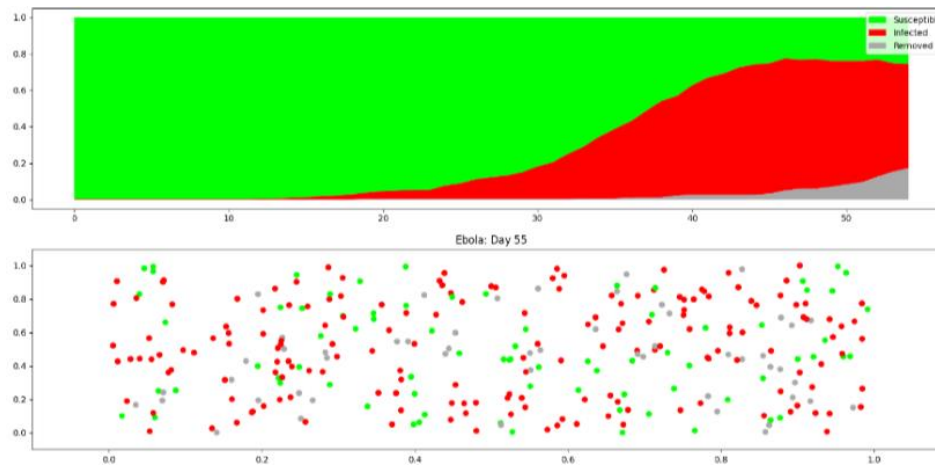
Covid-19 With Quarantine



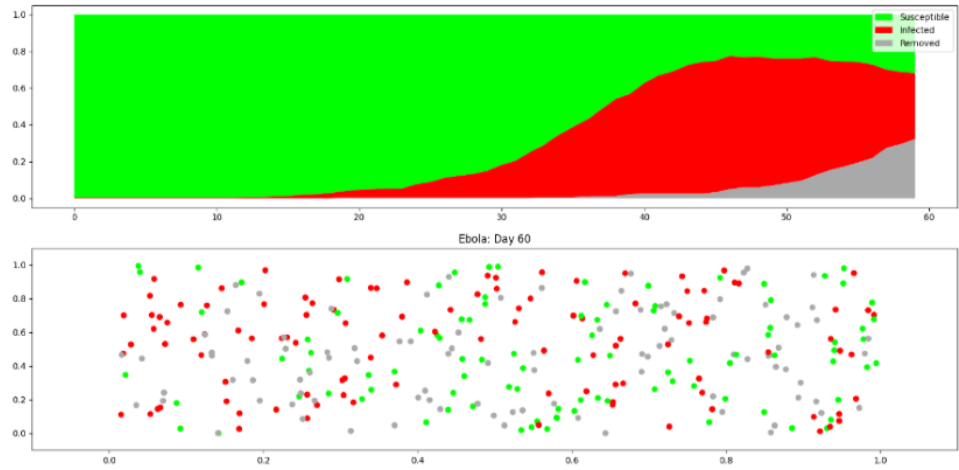
Ebola with Quarantine



Covid-19 without Quarantine



Ebola without Quarantine



Conclusion

- As seen from the output data, it is evident that K is to define it as the how variant R can be on individual basis. K can affect the value of R on individual basis, such as a low value of K means a higher variance in R . A very high K value will mean everyone will pretty much only infect R number of people. R is defined as number of people a single infected person will infect which means a large R value indicates virus getting spread across large number of people.
- It is observed that there is a strong correlation between the population density and the number of Covid-19 cases. We found that population density has a positive effect on increasing the number of cases of COVID-19 virus and Ebola. As population density increases, the number of infections also increases as well. The higher the population density, the closer there is people in public places, which will increase the spread of the current virus. Thus, it is estimated that population density has a positive effect on the spread of Covid-19.
- We observed following scenario's: no mask at all and effectiveness depends upon different quality of masks. Thus, certainly if people choose to wear normal mask, we all can reduce the spread of the respiratory related disease such as Covid-19 and Ebola altogether. Based on our studies, masks do play an important role in reducing the epidemic.
- If people are in always contact with each other, they will infect or be infected by others very easily. On the other hand, if people tend to stay at home which means they hardly contact with others, then they certainly will be infected with less probability. Recent estimates for COVID-19 transmission, we predict that under effective quarantine rules, we can reduce many confirmed cases for both diseases. When used with other public health measures like widespread testing and social distancing, contact tracing is a key strategy for preventing further spread of COVID-19.
- In conclusion, to study about these epidemic diseases, it is important to look at all sides and factors in order to understand how to slow down these outbreaks as in our cases we have experimented on not just R and K factors but also whether an individual wear a mask, and focus on individual's health status, age, whether he/she get tested and stay quarantine. With our simulator, I believe we can provide an insight on how to slow these virus outbreaks. While Ebola outbreak are quite severe, we find that Covid-19 provides more impact to the population comparing to Ebola as it spreads faster to the surrounding people.

Unit Tests and Code Coverage

Test Coverage

The screenshot displays the PyCharm Code Coverage window, which is divided into two main sections: a left sidebar for folder selection and a right pane for the coverage report.

Left Sidebar:

- test** (expanded):
 - PersonTest.py
 - EbolaTest.py
 - VirusTest.py
 - SimulationTest.py
 - Covid19Test.py
 - __pycache__
 - ConfigUtilTest.py
 - PopulationTest.py
 - VisualizationTest.py
- main** (expanded):
 - Person.py
 - ConfigUtil.py
 - Simulation.py
 - Ebola.py
 - Virus.py
 - __pycache__
 - Covid19.py
 - Main.py
 - Population.py
 - Visualization.py

Right Pane:

Enable code coverage for new launches? ☒ Open cov dir

Auto clear on a new launch? ☒ Clear

Auto refresh on new launch? ☒ Refresh

Name	Stmts	Miss	Cover	Missing
Main.py	11	11	0%	1-4, 7, 11-14, 16, 20
Visualization.py	55	-	100%	
ConfigUtil.py	25	2	92%	57, 68
Population.py	209	17	91.9%	103-109, 113-114, 116-117, 142, 202, 293, 299, 338, 3
Simulation.py	75	1	98.7%	67
Ebola.py	5	-	100%	
Virus.py	85	2	97.6%	172-173
Person.py	122	2	98.4%	69, 71
Covid19.py	5	-	100%	
TOTAL	592	35	94.1%	

Choose folder to analyze

type filter text

test (expanded):

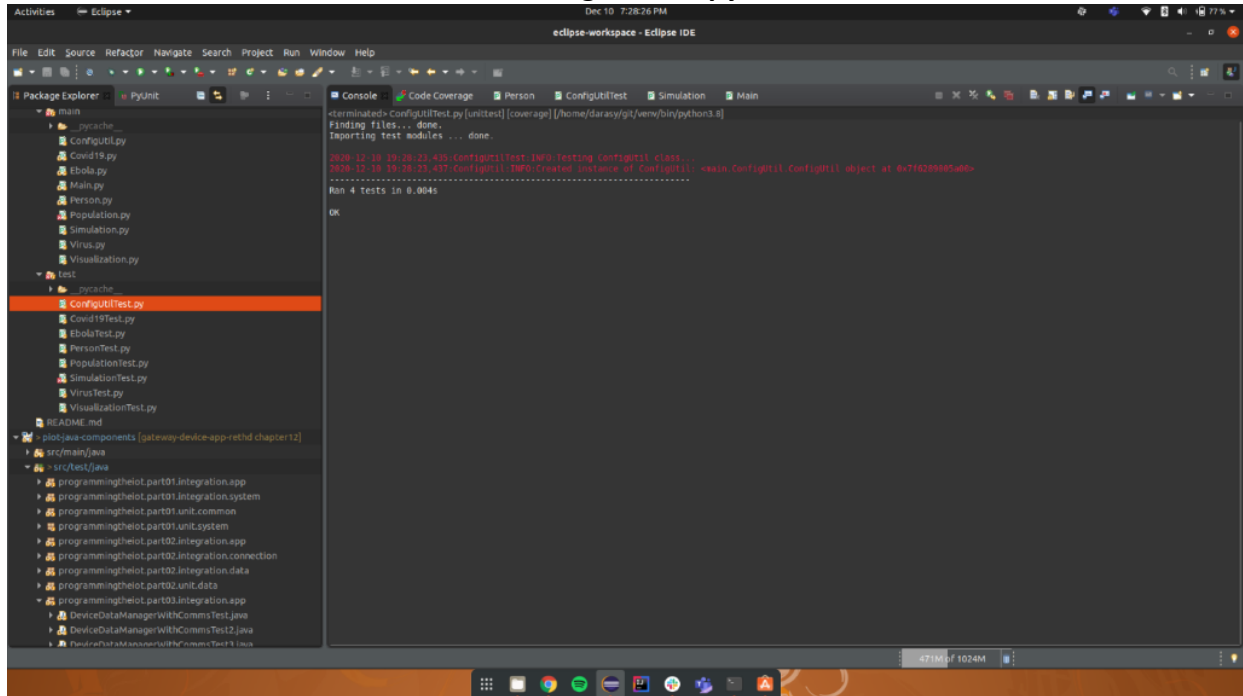
- PersonTest.py
- EbolaTest.py
- VirusTest.py
- SimulationTest.py
- Covid19Test.py
- __pycache__
- ConfigUtilTest.py
- PopulationTest.py
- VisualizationTest.py

main (expanded):

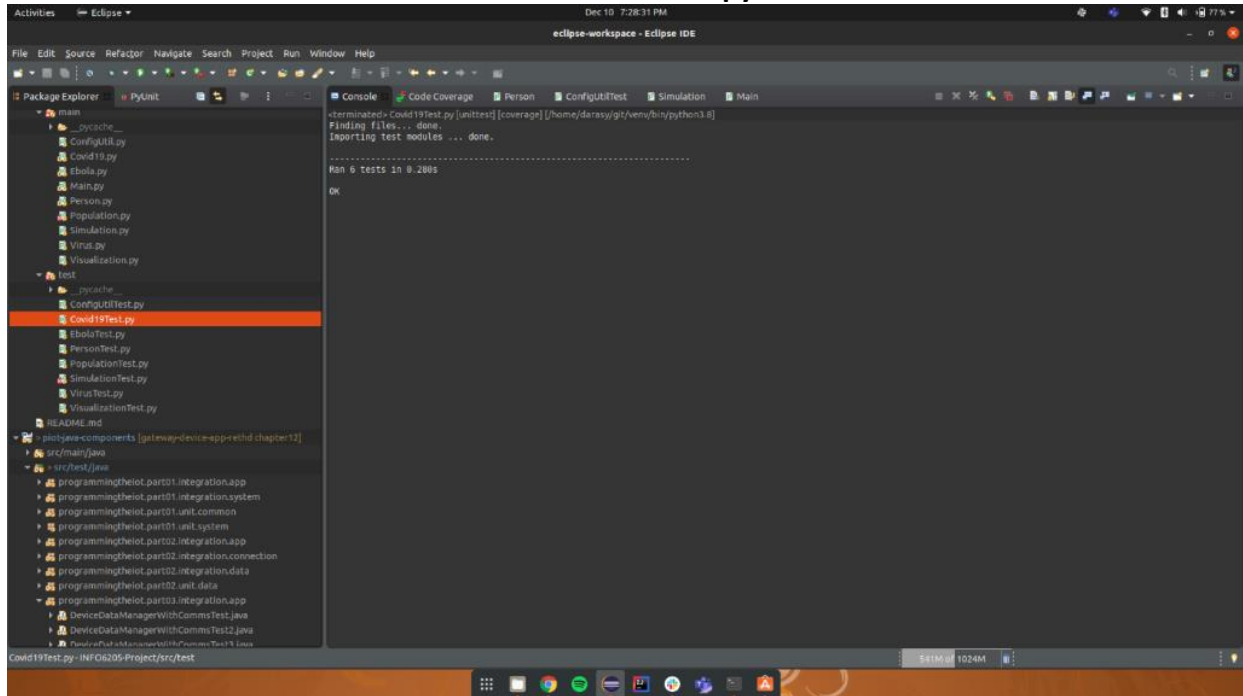
- Person.py
- ConfigUtil.py
- Simulation.py
- Ebola.py
- Virus.py
- __pycache__
- Covid19.py
- Main.py
- Population.py
- Visualization.py

Name	Stmts	Miss	Cover	Missing
VisualizationTest.py	32	-	100%	
ConfigUtilTest.py	68	-	100%	
PopulationTest.py	133	-	100%	
SimulationTest.py	23	2	91.3%	36-37
Covid19Test.py	35	-	100%	
EbolaTest.py	35	-	100%	
VirusTest.py	123	-	100%	
PersonTest.py	81	-	100%	
TOTAL	530	2	99.6%	

ConfigUtilTest.py



Covid19Test.py



EbolaTest.py

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left lists the project structure, with the 'test' folder expanded and 'EbolaTest.py' selected. The Console on the right displays the output of the test run:

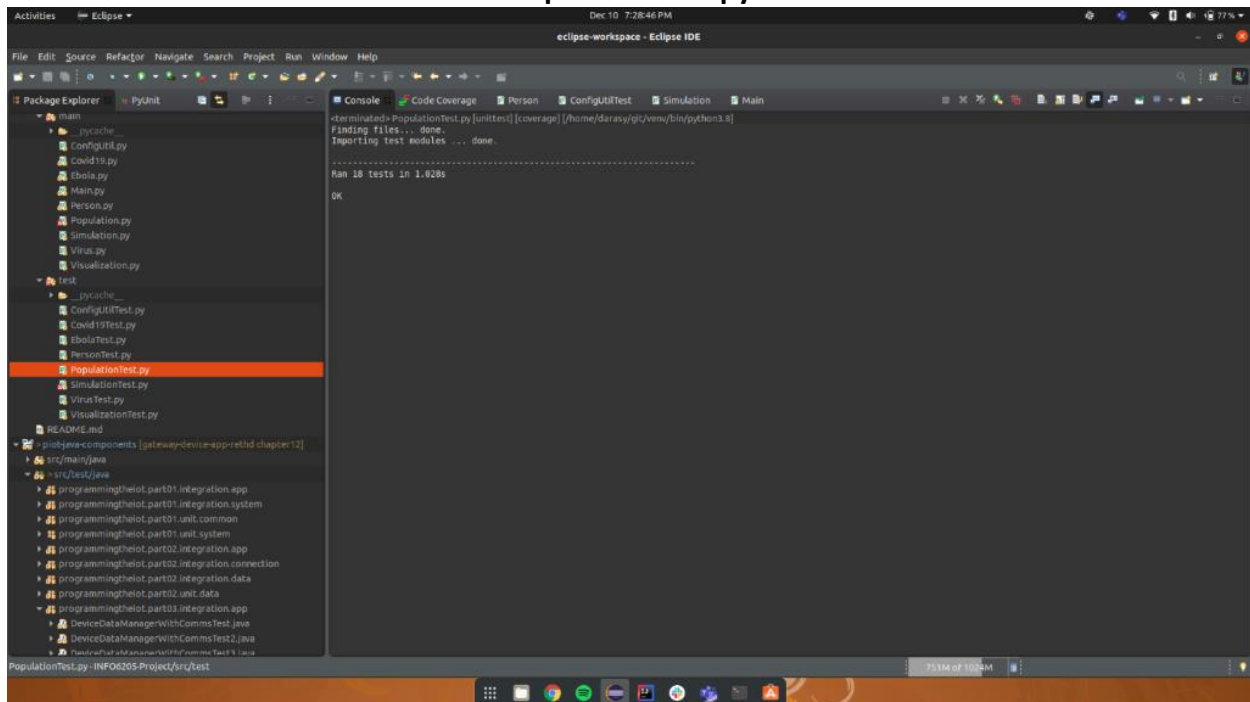
```
<terminated> EbolaTest.py [unittest] [coverage] [/home/darasy/gpt/venv/bin/python3.8]  
Finding files... done.  
Importing test modules ... done.  
-----  
Run 6 tests in 0.278s  
OK
```

PersonTest.py

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left lists the project structure, with the 'test' folder expanded and 'PersonTest.py' selected. The Console on the right displays the output of the test run:

```
<terminated> PersonTest.py [unittest] [coverage] [/home/darasy/gpt/venv/bin/python3.8]  
Finding files... done.  
Importing test modules ... done.  
-----  
2024-12-10 15:28:19.638:PersonTest:INFO:Testing Person class...  
Run 15 tests in 0.003s  
OK
```

PopulationTest.py



```
Activities  Eclipse
Dec 10 7:28:46 PM
eclipse-workspace - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer  PyUnit  Console  Code Coverage  Person  ConfigUtilTest  Simulation  Main

main
├── __pycache__
│   ├── ConfigUtil.py
│   ├── Covid19.py
│   ├── Ebola.py
│   ├── Main.py
│   ├── Person.py
│   ├── Population.py
│   ├── Simulation.py
│   ├── Virus.py
│   └── Visualization.py
├── test
│   ├── __pycache__
│   │   ├── ConfigUtilTest.py
│   │   ├── Covid19Test.py
│   │   ├── EbolaTest.py
│   │   ├── PersonTest.py
│   │   ├── PopulationTest.py
│   │   ├── SimulationTest.py
│   │   ├── VirusTest.py
│   │   └── VisualizationTest.py
│   └── README.md
└── .pio\java-components [gateway@devise-app:rethd chapter12]
    ├── src/main/java
    │   ├── programmingtheiot.part01.integration.app
    │   ├── programmingtheiot.part01.integration.system
    │   ├── programmingtheiot.part01.unit.common
    │   ├── programmingtheiot.part01.unit.system
    │   ├── programmingtheiot.part02.integration.app
    │   ├── programmingtheiot.part02.integration.connection
    │   ├── programmingtheiot.part02.integration.data
    │   ├── programmingtheiot.part02.unit.data
    │   ├── programmingtheiot.part03.integration.app
    │   ├── DeviceDataManagerWithCommsTest1.java
    │   └── DeviceDataManagerWithCommsTest2.java
    └── .pio\java-components\src\test

PopulationTest.py - INFO6205-Project\src\test

7:33AM of 10:41M
```

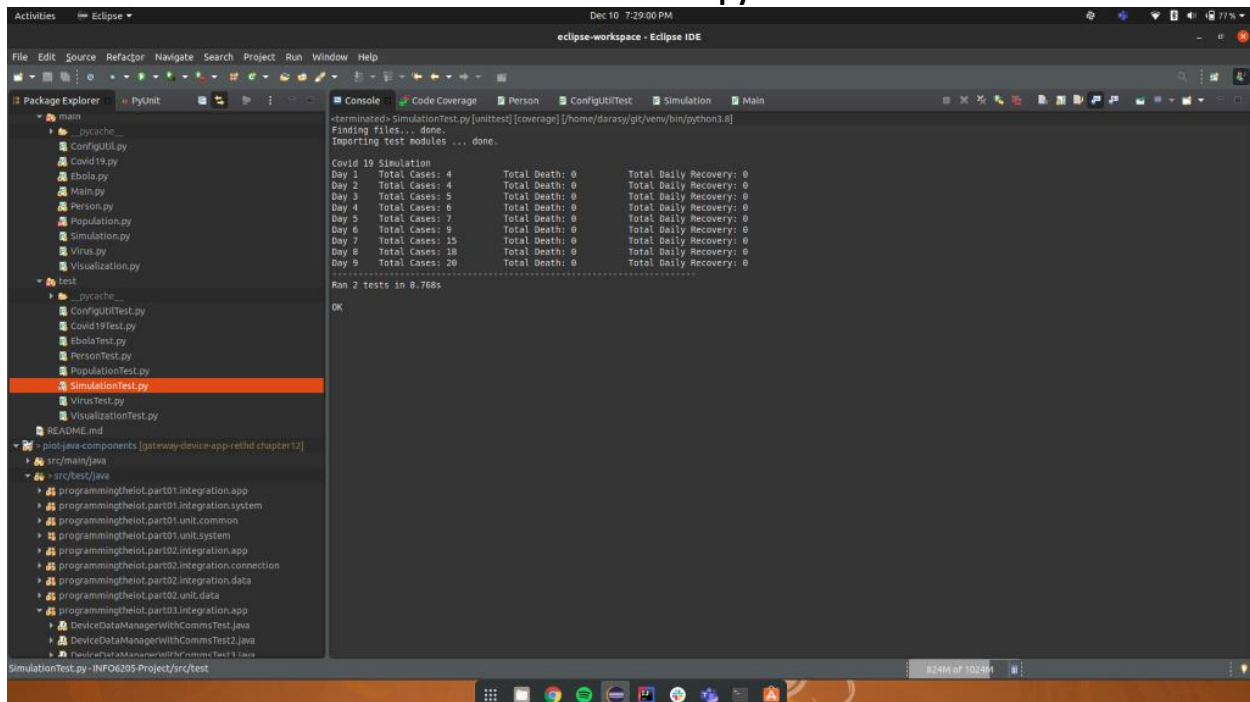
Console Output:

```
<terminated> PopulationTest.py [unittest] [coverage] [/home/darasy/git/venv/bin/python3.8]
Finding files... done.
Importing test modules ... done.

Ran 18 tests in 1.628s

OK
```

SimulationTest.py



```
Activities  Eclipse
Dec 10 7:29:00 PM
eclipse-workspace - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer  PyUnit  Console  Code Coverage  Person  ConfigUtilTest  Simulation  Main

main
├── __pycache__
│   ├── ConfigUtil.py
│   ├── Covid19.py
│   ├── Ebola.py
│   ├── Main.py
│   ├── Person.py
│   ├── Population.py
│   ├── Simulation.py
│   ├── Virus.py
│   └── Visualization.py
├── test
│   ├── __pycache__
│   │   ├── ConfigUtilTest.py
│   │   ├── Covid19Test.py
│   │   ├── EbolaTest.py
│   │   ├── PersonTest.py
│   │   ├── PopulationTest.py
│   │   ├── SimulationTest.py
│   │   ├── VirusTest.py
│   │   └── VisualizationTest.py
│   └── README.md
└── .pio\java-components [gateway@devise-app:rethd chapter12]
    ├── src/main/java
    │   ├── programmingtheiot.part01.integration.app
    │   ├── programmingtheiot.part01.integration.system
    │   ├── programmingtheiot.part01.unit.common
    │   ├── programmingtheiot.part01.unit.system
    │   ├── programmingtheiot.part02.integration.app
    │   ├── programmingtheiot.part02.integration.connection
    │   ├── programmingtheiot.part02.integration.data
    │   ├── programmingtheiot.part02.unit.data
    │   ├── programmingtheiot.part03.integration.app
    │   ├── DeviceDataManagerWithCommsTest1.java
    │   └── DeviceDataManagerWithCommsTest2.java
    └── .pio\java-components\src\test

SimulationTest.py - INFO6205-Project\src\test

8:24M of 10:24M
```

Console Output:

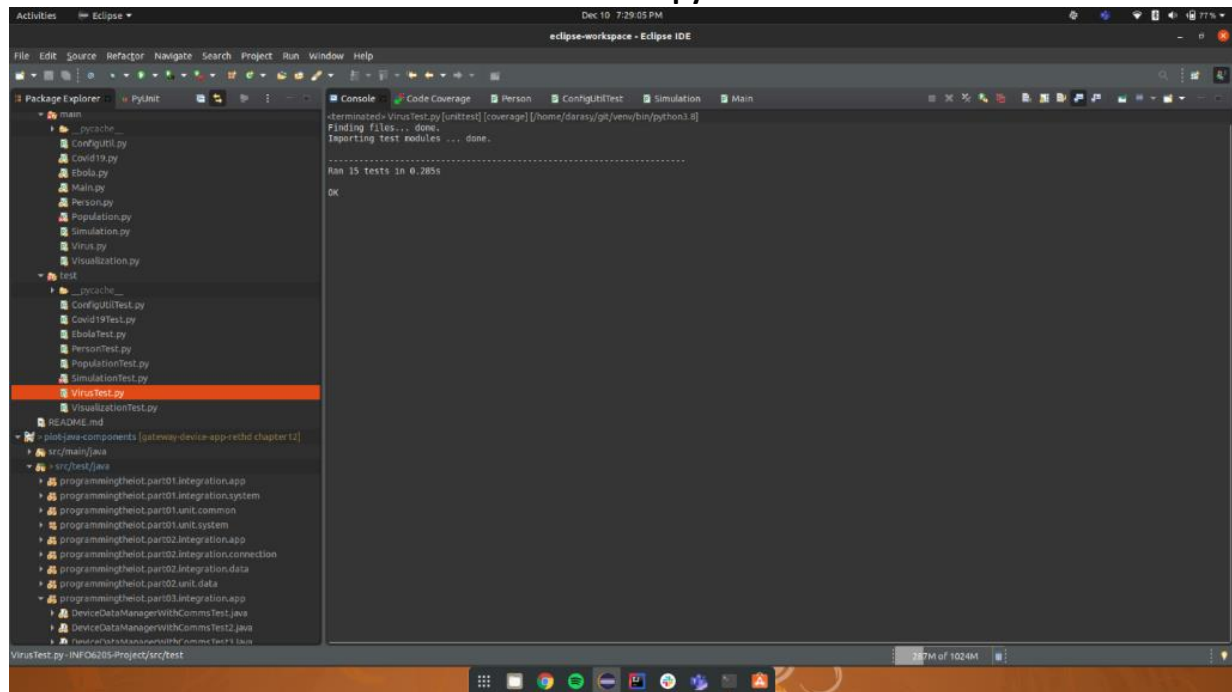
```
<terminated> SimulationTest.py [unittest] [coverage] [/home/darasy/git/venv/bin/python3.8]
Finding files... done.
Importing test modules ... done.

Covid 19 Simulation
Day 1 Total Cases: 4 Total Death: 0 Total Daily Recovery: 0
Day 2 Total Cases: 4 Total Death: 0 Total Daily Recovery: 0
Day 3 Total Cases: 5 Total Death: 0 Total Daily Recovery: 0
Day 4 Total Cases: 6 Total Death: 0 Total Daily Recovery: 0
Day 5 Total Cases: 7 Total Death: 0 Total Daily Recovery: 0
Day 6 Total Cases: 9 Total Death: 0 Total Daily Recovery: 0
Day 7 Total Cases: 15 Total Death: 0 Total Daily Recovery: 0
Day 8 Total Cases: 18 Total Death: 0 Total Daily Recovery: 0
Day 9 Total Cases: 20 Total Death: 0 Total Daily Recovery: 0

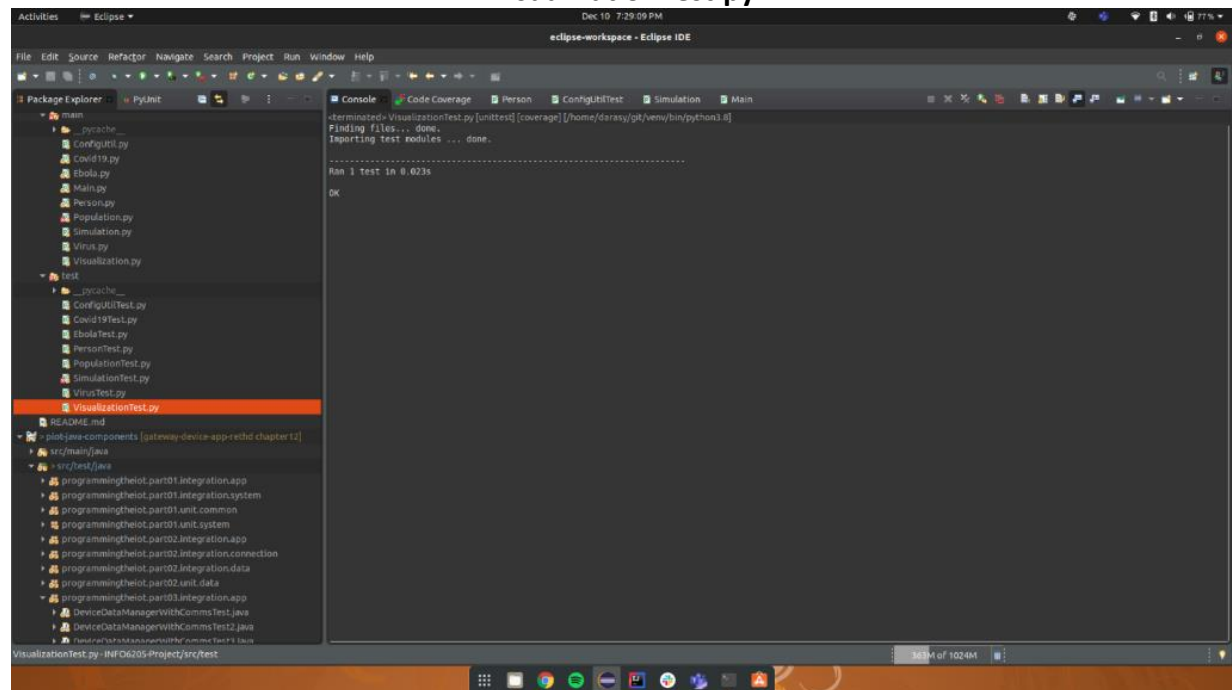
Ran 2 tests in 8.768s

OK
```

VirusTest.py



VisualizationTest.py



References

- [1] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4982163/table/T2/?report=objectonly>
- [2] [https://www.thelancet.com/journals/laninf/article/PIIS1473-3099\(20\)30840-9/fulltext](https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30840-9/fulltext)
- [3] <https://pubmed.ncbi.nlm.nih.gov/15178190>
- [4] <https://ifrcgo.org/ecv-toolkit/action/social-distancing>
- [5] <https://theconversation.com/is-the-k-number-the-new-r-number-what-you-need-to-know-140286>
- [6] <https://www.ucdavis.edu/coronavirus/news/your-mask-cuts-own-risk-65-percent>
- [7] <https://www.worldometers.info/coronavirus>
- [8] <https://thehill.com/changing-america/well-being/medical-advances/491760-3-in-4-americans-say-they-are-self-isolating-in>