Student: Giang Vu

NUID: 00153797

INFO 6205 – Prof. Robin Hillyard

Final Project Report: Covid-19 Simulation

1. **Introduction about the topic**

   SARS-Cov-2 or being known as COVID-19 has been a huge disease in our new decade. Started from early days of 2020, the virus has been spread out for more than a year without any sign of being stopped anytime soon. The virus has caused detrimental effect for our world, from destroying our economy to changing everyone works and daily activities habits. Although the vaccines has been distributed, the cases numbers still seems to increase with some possible COVID-19 variations. Building a mathematical model for Covid-19 simulation will help us not only to get better prediction of the virus behaviors under different factors/conditions but also can allow the general public to have an easier understanding of Covid-19 and how the vaccine will affect them.

2. **Aim of the project.**

   The aim of this project is to model and simulate the spreading outcome of 2 different viruses, namely Covid-19 and H1N1, given the same initial condition (populations, vaccine effect, hospital capacity, etc.) The simulation model will take into consideration different factors of the virus that will affect the output like population, R_value (basic reproduction number), recovery rate of patient and the intervention of government policies like social distancing, wearing mask and provide vaccination. To simplify the study, I would use data and number of mostly from Massachusetts to run the simulation sample and analyze the affection of both viruses on the state.

3. **Complete project details.**

   The project is written in Python, using Jupyter notebook as environment. All the output of the program is in form of graphs and charts for easier visualization of the results. The code and all graph images are stored inside the deliverable GitHub repository: Final Project Repo for Virus Simulation. Detailed of libraries requirement and programming running instruction can be accessed from the repo Readme file.

   Repository structuring:

   - Covid_simulation.ipynb: main program/code to run the simulation, in Jupyternotebook format.
   - Graphs: contains all .png and HTML output of the charts/graphs in the simulation
   - Unit Test.png: Screen shot unit test for functions used in program.
   - FinalProjectReport_GiangVu.pdf: Documenting report.

   The model formula will take into account these viruses' factors:

   - Basic reproduction value (R): The average number of people who will catch a disease from one contagious person.

- Recovery rate of patient after getting infected.
- Incubation period: Time elapsed between exposure and when symptoms and signs are first apparent.
- Average contact Rate

While there are two main models, one is the base SEIR model ((Susceptible, Exposed, Infectious, and Recovered) model using the virus parameters above, while adjusting the R value to reflect the effectiveness of wearing mask and including the calculation for social distancing. The 2nd model is the base SEIR + consider 3 new factors that are death cases, hospitalization, and potential effect of vaccination coverage. Afterward, I used the 2nd model for scenario simulation to compare 2 viruses (Covid-19 & H1N1). There are 3 scenarios ran for each virus that would be:

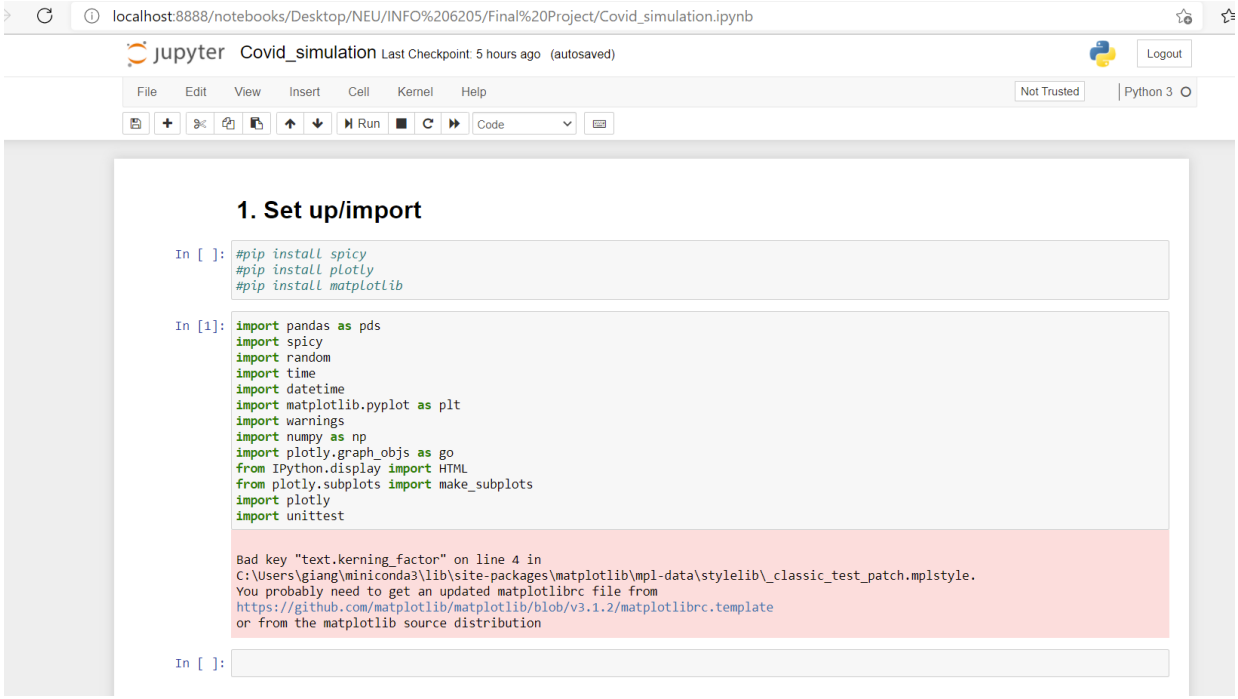
- Increasing social distancing
- Increasing hospital capacity
- Different date of vaccination.

I will go into more details and explanation about the models and scenarios on Section 4 of this report.

4. Implementation - charts, algorithm

4.1 Set up environment:

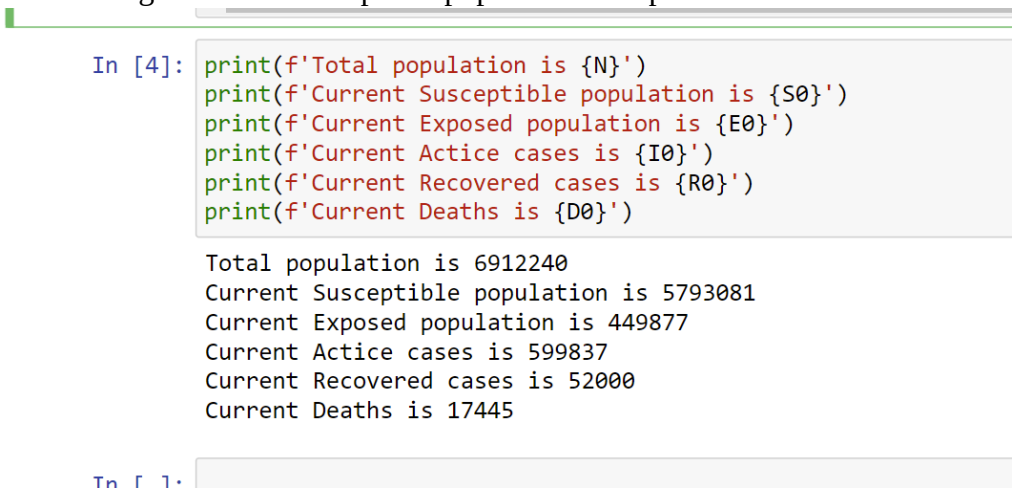
Installing necessary libraries/packages using pip install [package_name] then import [package name]



4.2 Initialized common parameters/data.

As previously mentioned above, some common parameters being used are:

- R Value
- Recovery rate (RT)
- Incubation period (incubation)
- Reporting rate (rr): using China data as that was the only source of country data that is available.
- Initialized MA state data for simulation set up like Population(N), Active cases (I), Confirmed Cases(C), Recovered Cases (R), Exposed (E), Susceptible (S) and Deaths (D). (source: https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_daily_reports_us/04-16-2021.csv)
  Below image is the final output of population set up.



```
In [4]: print(f'Total population is {N}')
        print(f'Current Susceptible population is {S0}')
        print(f'Current Exposed population is {E0}')
        print(f'Current Actice cases is {I0}')
        print(f'Current Recovered cases is {R0}')
        print(f'Current Deaths is {D0}')

        Total population is 6912240
        Current Susceptible population is 5793081
        Current Exposed population is 449877
        Current Actice cases is 599837
        Current Recovered cases is 52000
        Current Deaths is 17445

In [ ]:
```

4.3 Modeling with Formula

4.3.1 Base SEIR Model:

The SEIR model divides the population into 4 groups: Susceptible, people who were never infected or vaccinated; Exposed, people who were exposed to the virus and have not shown any symptoms; Infectious, people who are infectious and Recovered, people who are recovered.

$$N\ (Total\ Population) = S\ (Susceptible) + E\ (Exposed) + I\ (Infectious) + R\ (Recovered)$$

$$\frac{dS}{dt} = -\frac{\beta SN}{I} \quad \text{(Equation 1)}$$

$$\frac{dE}{dt} = \frac{\beta SN}{I} - \sigma E \quad \text{(Equation 2)}$$

$$\frac{dI}{dt} = \sigma E - \gamma I \quad \text{(Equation 3)}$$

$$\frac{dR}{dt} = \gamma I \quad \text{(Equation 4)}$$

The parameters σ, β, and γ are defined as below:

σ = 1/Incubation period, β (*average contact rate*) = $R_0 \times \gamma$

γ (*recovery rate*) = 1/Time from symptom onset to recovery (RT)

To simulate the process, for each person in S, we simulate he/she enter stage E with the probability of β. For person in E, we simulate he will enter I stage with the probability of σ. For person in I, we simulate whether he will enter R stage with the probability of γ. Afterward, new number of E, I, R can be calculated by adding total number of people entering each stage then minus the number of people moving out of those states. Below is the code for above formula:

3.1 SEIR model (base)

N (population) = S(Susceptible) + E(Exposed) + I(Infected)+ R(Recovered)

n [6]:
```
#Virus simulation for 1 day (from day 0 to day 1)

prob_e = beta*I0/N # probability of an individual go from S to E
new_e = np.sum(np.random.rand(S0) < prob_e) #calculate number of new people gettting exposed
S1 = S0 - new_e #new S population after a portion getting exposed
prob_i = 1/incubation # probability of an individual go from E to I
new_i = np.sum(np.random.rand(E0) < prob_i) #calculate number of new people gettting infected
prob_r = gamma # probability of an individual go from I to R
new_r = np.sum(np.random.rand(I0) < prob_r)
E1 = E0 + new_e - new_i
I1 = I0 + new_i - new_r
R1 = R0 + new_r
```

Afterward, I define the function: SEIR_simuation using above formula/calculation but also include another factor for social distancing sd [0,1] where 0 = no social distance and 1 = locked down (no infection)
This will only change the calculation for probability of stage E: prob_e = beta*I*(1-sd)/N (where N is total population).
For the final part of this base model, I take account for effectiveness of mask, which I assume can reduce the spread chance of infected person by 30%, which make our new reproduction number (R_1) = R_0 * 0.07.
Then I ran both model (no mask and with mask) with 0% and 50% social distancing factor.

4.3.2 SEIR + mask + hospitalization + vaccination

Based on the base SEIR model, I considered death and hospitalization factors to make it more realistic. I also include the impact of vaccination on covid19 cases to predict the pandemic situation might improve after vaccines are distributed.

I divided new infectious cases into two types: critically ill patients who have strong symptoms and non-critical ill patient (mild). Then I experimented two types of infectious patients.

For vaccination, I added two vaccine factors, vaccine date and vaccine proportion to the model. At the vaccination date, the S population is decreased by the proportion of vaccination ratio, which lead to less people would get infected.

For model simplicity, I made assumption that the vaccine proportion is 25% or 0.25. Below is the code for new model with calculation including new factors, function SEIRHV_simulation_with_masks:

```
capacity = 15000 #https://www.mass.gov/doc/command-center-hospital-capacity-charts/download
H0 = 0 #hospitalized number
S0 = N - E0 - I0 - R0 - D0 - H0 # S+E+I+H+R+D=N

#simulate all critical ill patient ratios at one time
CR_all = np.random.normal(CR, 0.001, period)

S[0] = S0
E[0] = E0
I_c[0] = np.sum(np.random.rand(I0) < CR_all[0])
I_m[0] = I0 - I_c[0]
H[0] = H0
R[0] = R0
D[0] = D0

rho = 2 #sensitivity factor for controling the hospital factor on different recovery rate

for i in range(1, period):
    prob_e = beta_1*(I_c[i-1]+I_m[i-1])*(1-sd)/N
    new_e = np.sum(np.random.rand(int(S[i-1])) < prob_e)
    S[i] = S[i-1] - new_e
    #after vaccine injection: affect expose rate
    if i == vaccine_date:
        S[i] = (1-vaccine_p) * S[i-1]
        V[i] = vaccine_p * S[i-1]
    prob_i = 1/incubation
    new_i = np.sum(np.random.rand(int(E[i-1])) < prob_i)
    E[i] = E[i-1] + new_e - new_i

    #divide two types of patients
    new_i_c = np.sum(np.random.rand(new_i) < CR_all[i]) #critical patient
    new_i_m = new_i - new_i_c #mild patient

    new_d_hc = np.sum(np.random.rand(max(int(H[i-int(h2d)]),0)) < CDR_h) if i>=int(h2d) else 0 #hospitalized critical death
    new_d_c = np.sum(np.random.rand(max(int(I_c[i-int(s2d)]),0)) < CDR*(1-h_capacity))  if i>=int(s2d) else 0 #non-hospitalized critical death
    new_d_m = np.sum(np.random.rand(max(int(I_m[i-int(s2d)]),0)) < MR*(1-h_capacity))  if i>=int(s2d) else 0 #non-hospitalized mild death

    new_r_hc = np.sum(np.random.rand(max(int(H[i-int(h2r)]),0)) < g_hc*(1+h_capacity)) if i>=int(h2r) else 0  #hospitalized critical recovery
    new_r_c = np.sum(np.random.rand(max(int(I_c[i-int(RT)]),0)) < gamma_c*(1+rho*h_capacity)) if i>=int(RT) else 0  #non-hospitalized critical recovery
    new_r_m = np.sum(np.random.rand(max(int(I_m[i-int(RT)]),0)) < gamma_m*(1+rho*h_capacity)) if i>=int(RT) else 0  #non-hospitalized mild recovery

    D[i] = D[i-1] + new_d_hc + new_d_c + new_d_m

    #check if hospital exceeding capacity
    new_h = np.sum(np.random.rand(max(int(I_c[i-1]),0)) < hospital_rate) if H[i-1]<capacity else 0

    I_m[i] = max(I_m[i-1] + new_i_m - new_r_m - new_d_m, 0)
    I_c[i] = max(I_c[i-1] + new_i_c - new_h - new_r_c - new_d_c, 0)
    H[i] = max(H[i-1] + new_h - new_d_hc - new_r_hc, 0)
    R[i] = R[i-1] + new_r_hc + new_r_c + new_r_m
```

4.4 Compare 2 viruses/ Scenarios Simulation.

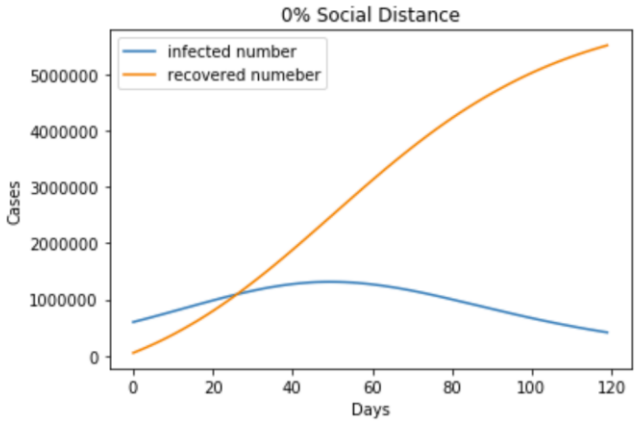
In this part, I compare 2 viruses using 3 different controlling policies: increasing social distancing, increasing hospital capitalization and distribute different vaccine date. The parameter setting for each accordingly would be 0%, 25%, 50% and 75%. Different vaccination date scenarios are vaccinated at 30, 60 and 90 days while simulation running for 120 days total.

Each scenario will have 4 differents sub-scenarios that would be plotting as sub-plots. The whole scenario will be exported into a single html file like Covid_vaccination.html for easier view.

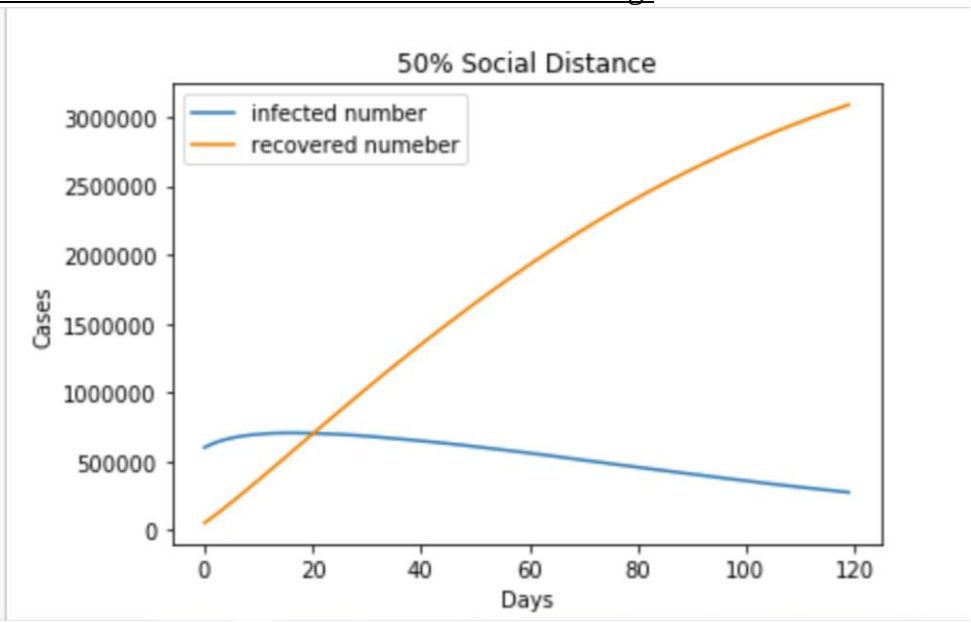
5. Output

Output is based on running the simulation for 120 days.

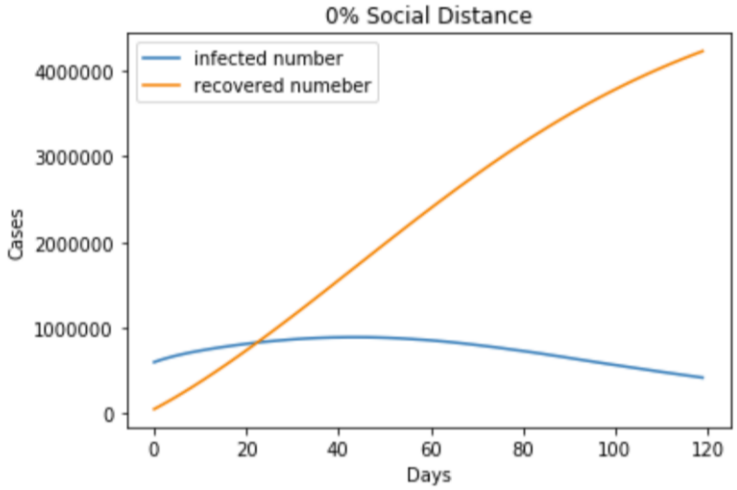
5.1 Base model simulation with no social distancing:



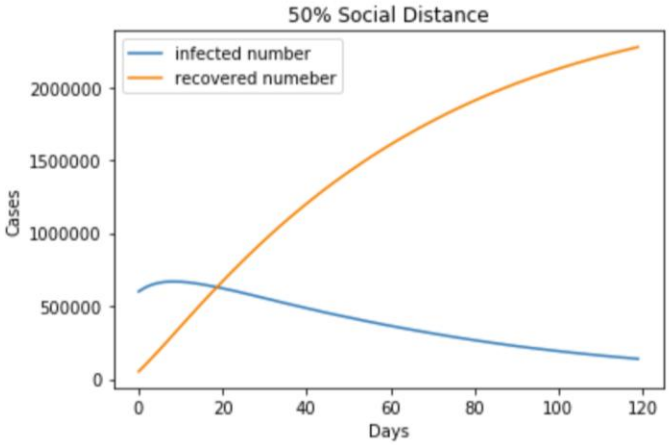
5.2 Base model simulation with 50% social distancing:



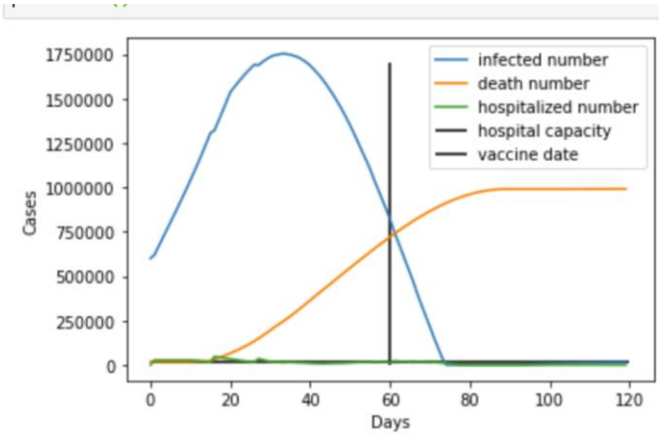
5.3 Base model simulation + using masks with no social distancing:



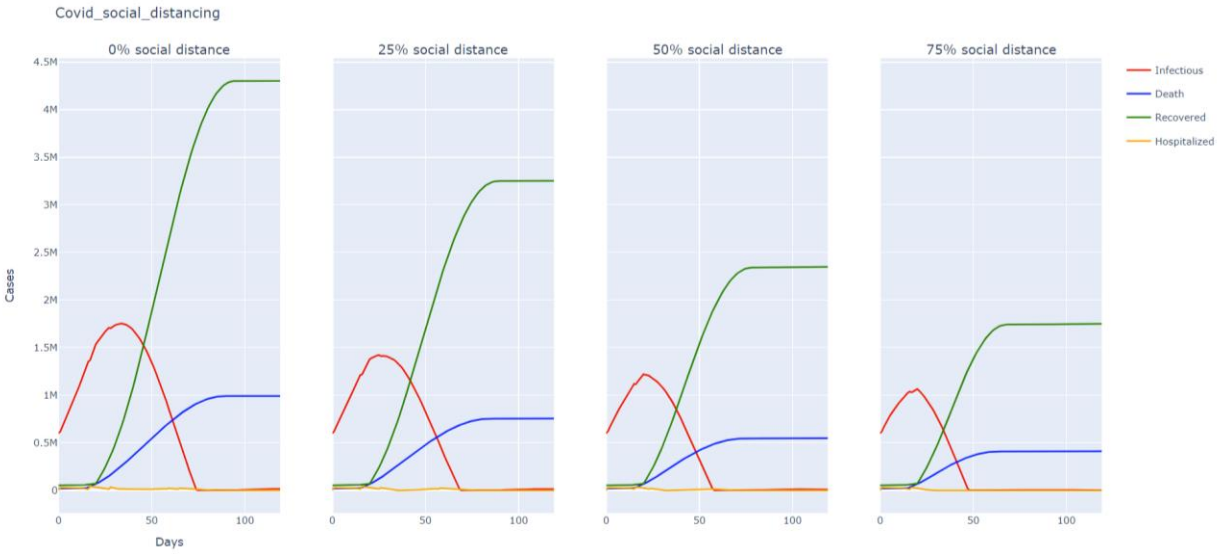
5.4 Base model simulation + using masks with 50% social distancing:

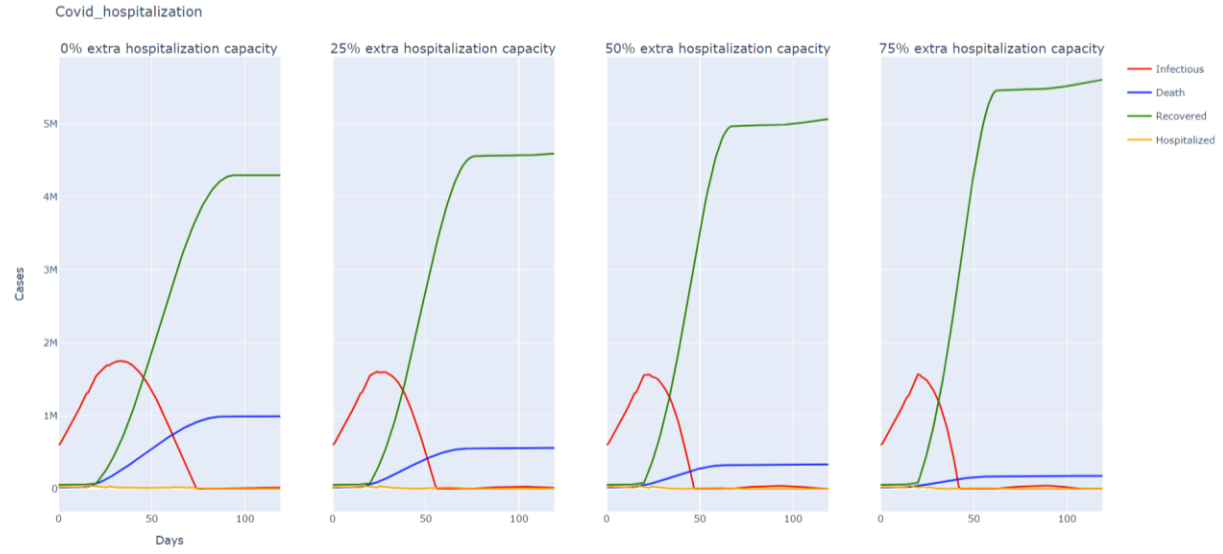


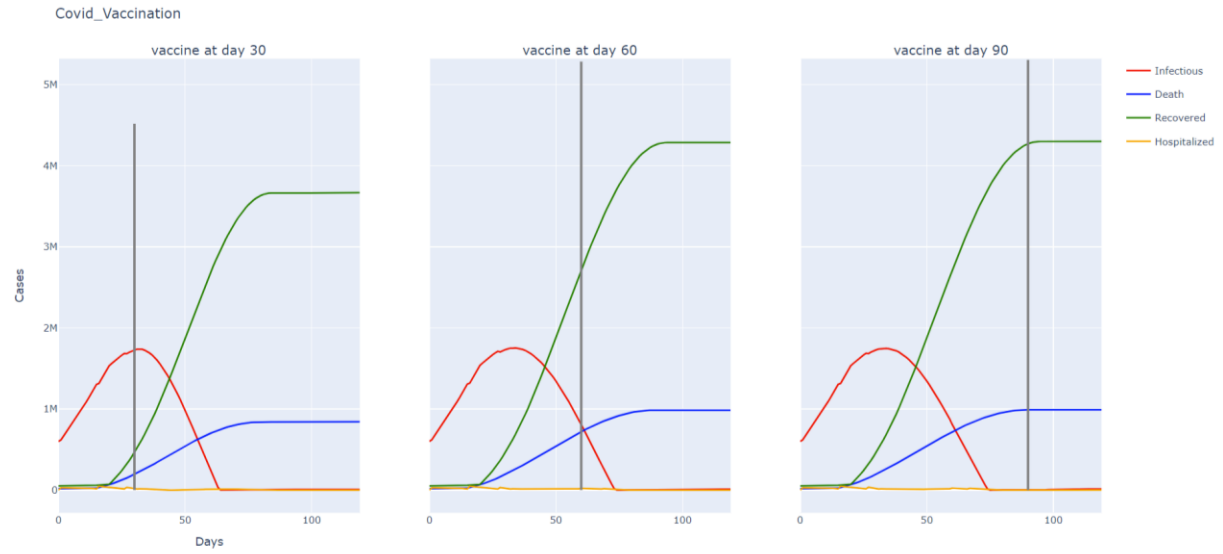
5.5 SEIR + mask + hospitalization + vaccination (0.25 proportion and vaccine date at day 60):



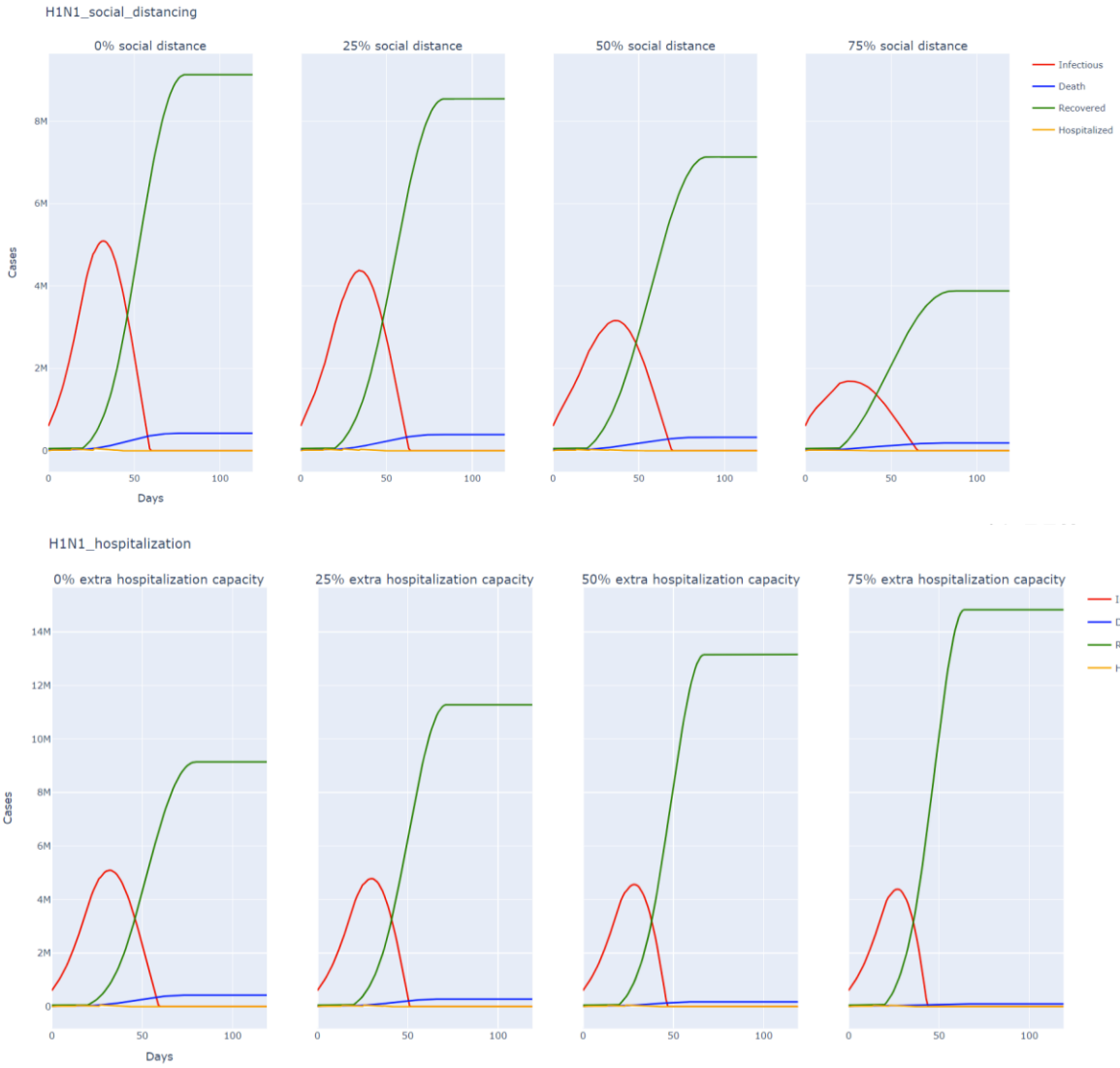
5.6 Covid-19 with 3 Scenarios

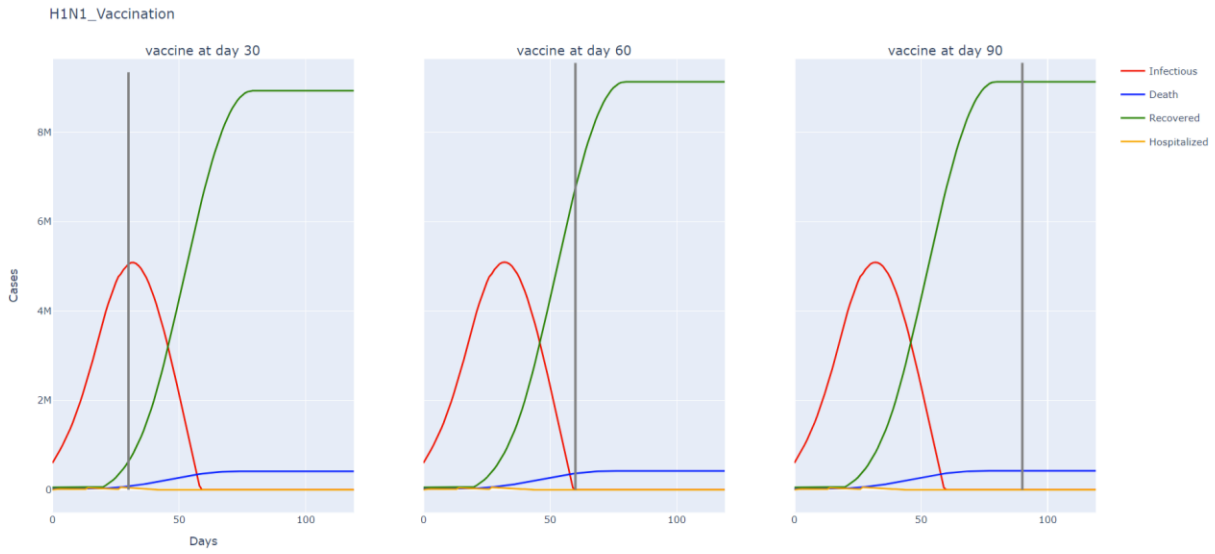

Covid_social_distancing
0% social distance
25% social distance
50% social distance
75% social distance
Infectious
Death
Recovered
Hospitalized
Cases
Days


Covid_hospitalization
0% extra hospitalization capacity
25% extra hospitalization capacity
50% extra hospitalization capacity
75% extra hospitalization capacity
Infectious
Death
Recovered
Hospitalized
Cases
Days


Covid_Vaccination
vaccine at day 30
vaccine at day 60
vaccine at day 90
Infectious
Death
Recovered
Hospitalized
Cases
Days

5.7 H1N1 with 3 Scenarios





6. Mathematical analysis/evidence

6.1 Exponential Growth:

The exponential growth rate indicates the detrimental effect of the disease spread. The growth rate and R value are correlated as with R > 1, infections number will increase exponentially as each person would infect > 1 people. If R value < 1, the virus slowly starts to go down. This may happen due to two reasons: the entire population was infected or due to controlling policies like social distancing, wearing masks or vaccination taken to reduce the number of people each infected person can infect.

6.2 Covid-19: infectious and can be fatal.

From the three policies used for controlling covid-19, we see a stable pattern that the number of recovered cases is much larger than that of the death cases and the infectious cases increase rapidly and reach peak in ~ 35-40 days in all scenarios. To investigate the detail of each policy, we found the following differences:

+ Social distancing seems to be the most effective when it can control the total deaths and slowing down the infection of virus. When increasing the social distancing power from 0% to 75%, we see a drop of deaths from 1 million to just 0.4 million, and no more new cases (as pandemic being controlled) from ~ day 75 to day 45. This proves the great effectiveness of social distancing approach regarding controlling the virus spreading.

+ Vaccination works quite well for controlling the infectious source but not as well as social distancing. We also can clearly see the effectiveness of vaccine is sensitive to injection time. If being used too late, i.e., the 90th day out of 120 days period, the vaccine shows not much effect in controlling the cases since the virus peak has passed earlier. The only significant reduction we could only see when vaccine being injected on day 30. (day 60 and day 90 scenario has mostly same output).

+ Hospitalization capacity expansion looks like the most reliable method to prevent death cases, which reduce the deaths from 1 million to 175,000 deaths when expand it

to 75%. However, the number of infectious cases is still seemed quite higher than social distancing, and the pandemic only seem to be controlled around day 75 (no new case). This can be explained as hospital treatment does not help prevent the infection source but could help speed up the recovery process and stop the possible death. So, we can see a relatively large infectious cases, but a much larger recovered population compared to the other two approaches.

6.3 H1N1: highly infectious but not fatal

From the three policies used for controlling H1N1, we see that clearly the number of death cases is much lower compared to Covid-19, although it being considered as one of the biggest epidemics in history. While the infectious cases seem to be even higher than covid in our simulation. To investigate the detail of each policy, we found the following differences:

+ Social distancing proved that it can slowing down the infection of virus effectively. When increasing the social distancing power from 0% to 75%, we see deaths count of 0.17 million. The peak infectious cases recorded reduce from 5 million to just ~ 1.7 million.

+ Vaccination method does not prove to be really effective here as we could observe no significant reduction in any metrics of deaths, infectious cases or recovered cases. This finding seems really interesting as we should have expected it to prevent the infection of the record in some ways.

+ Hospitalization seems to be the most effective method when it comes to prevent death cases or getting patient recovered quickly, 75% increasing hospital capacity reduce the deaths to < 100,000. The virus proved to be not fatal since most of the cases infected has shown to be recovered sooner or later.

7. Conclusion

We experimented 3 different controlling policies: social distancing, vaccine allocation and extra hospitalization capacity. For both viruses, social distancing seems like our go-to method regarding of our purpose as it is the most optimal way to prevent the spreading of virus infection. However, based on my analysis and observation above, Covid-19 proved to be a lot more detrimental compared to H1N1, as the later one didn't really need vaccination or hospitalization to be controlled effectively. While coronavirus has higher potential of death in case the patient did not receive treatment on time.

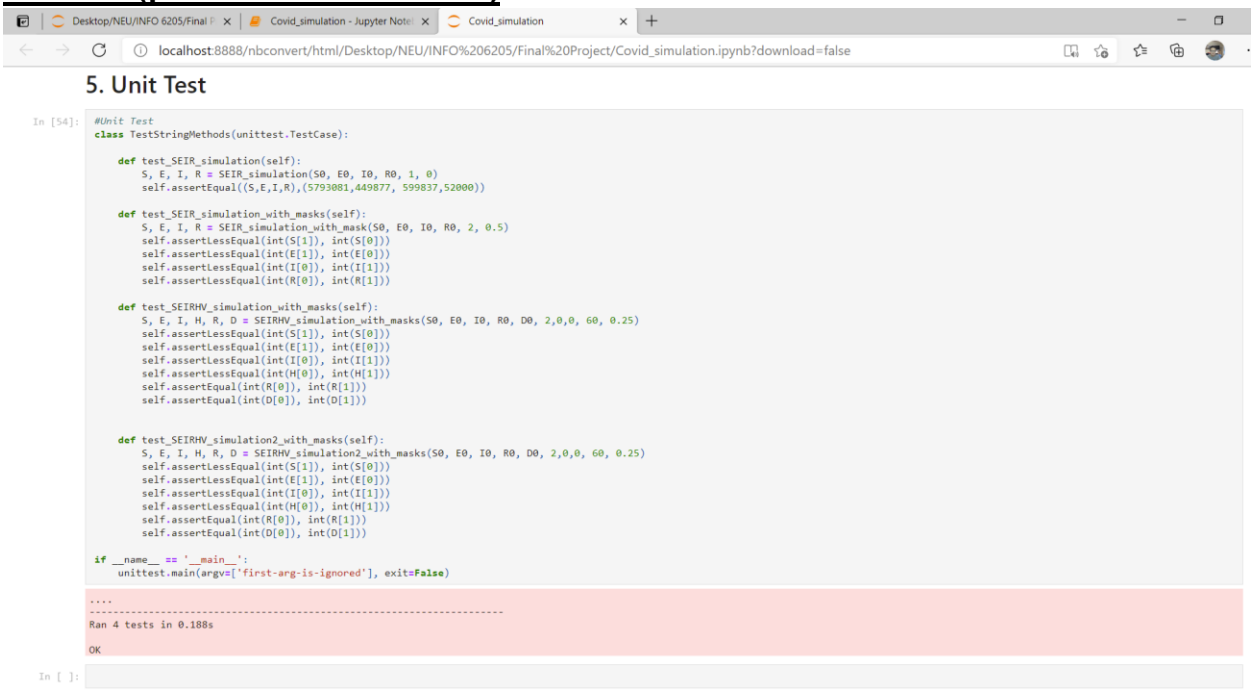
Regarding the mathematical analysis, the optimal policy could be:

- (1) When the virus is highly infectious while not fatal like H1N1, it is more economical to just manage the close contact within the population so that the infection source will decrease rapidly. Since the death rate is quite low, the extra hospitalization capacity for slowing down the mortality rate and enhancing the recovery process tends to be less productive when compared to the social distancing approach.
- (2) On the other side, when the virus is not as infectious but more dangerous like Covid-19, increasing health care capacity to prevent the death possibility and speed up the recovery process of each patient is quite impactful. The relatively low infection

rate compare to H1N1 makes the virus less sensitive to the infection process cut-off from vaccination. Thus, combining social-distancing and adding hospitalization could be the reliable approach for handling the Covid-19 here.

The simulation really helped me understand the optimal strategies when we need to control each virus. Before the simulation, it is quite obvious to think that those three policies will work well equally for controlling any virus with the common idea that the more controlling policies the better. However, government only have quite short, limited time to react to these situations. Hence, it would be beneficial and economical to investigate the virus and understand its key transmission process before making the optimal approach. Specially, they could consider using model simulation to predict the potential scenarios and evaluate the efficiency of different methods, which would save much more time instead of making non-optimal decisions that could cost a lot of resources and people's lives.

8. Unit Test (passed for all 4 functions)



9. References

1. Grant Sanderson. Exponential growth and epidemics, March 8, 2020
2. Adam Kleczkowski. Is the k number the new r number? what you need to know, 2020.
3. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data
4. Carcione, J., Santos, J., Bagaini, C., & Ba, J. (2020, May 15). A simulation of a covid-19 epidemic based on a deterministic seir model.
5. WHO Interim Guidelines. Infection prevention and control of epidemic- and pandemic-prone acute respiratory diseases in health care. 2017.

6. Laura Matrajt and Tiffany Leung. Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease. 2020.