CS51 Assignment 2: Algorithms and Simulation

This assignment has two distinct parts in addition to a reflection: Part 1 requires you to apply your knowledge of algorithmic thinking and optimization and Part 2 allows you to demonstrate your modeling and coding skills by writing a numerical simulation. Material relevant for Part 1 will be covered in class during weeks 3-5, while material relevant for Part 2 will be covered in class during weeks 7 and 8.

You'll notice several "*Optional Challenge*" problems throughout the assignment to challenge yourself. These will only be scored (4 or 5) if they are completed correctly with thorough explanation. If you attempt an optional challenge but do not succeed, you will not be penalized with a low score. Remember that you must include an explanation and interpretation for optional problems to be scored.

This is an individual assignment. We will be checking for similarities among submissions and will take plagiarism seriously.

You must complete all tasks within this pre-formatted Jupyter notebook. Please follow ALL formatting guidelines and the HC Guidelines in the assignment instructions on Forum (near the top and bottom of the instructions respectively).

PART 1: OPTIMIZATION

For this section of the assignment, you will select one of the scenarios below and apply #optimization. You must complete all sections. [#optimization, potentially relevant: #modeling, #algorithms, #variables, #utility, #constraints]

1. *Scenario 1*: To prevent the spread of an infectious disease, a vaccine needs to be distributed as quickly and efficiently as possible to the 15 cities that have had major outbreaks. How can you optimize the route between the cities? For this scenario, you should select cities that are relevant to the disease that you will choose for PART 2. It may be helpful to include a map of these cities (either an existing map or create your own).

2. *Scenario 2*: Suppose that a new virus is starting to spread, and many clinics do not have sophisticated diagnostic tools and must be able to determine whether or not a patient has this dangerous virus based solely on easily measured symptoms. You have been collecting information on symptoms (temperature, WBC count, headache severity, and cough severity) and you need to determine which patients have this new disease and which have only a milder illness. Plots that provide an overview of the data are available here. The data can be accessed at this link (1 = Infected, 0 = Not infected).

1.1 Optimization Problem:

Describe the optimization problem for your scenario: what is the objective function? What are the decision variables? Are there any constraints? Clearly articulate each component so

that it's clear how the objective value would be measured and how the decision variables would impact it (~200 words).

Scenario 1:The objective function is to find the global minima of the distance between the 15 cities in the US with the higest level of Covid - 19 ourbreak. The objective value is the length traveled that is measured in kilometers and is a quantitative discrete variable.

The decision variables are the order in which the cities will be visited, the way we will choose the cities to travel (which algorithm will be used to find the objective value) Objective value is the dependent variable while decision variables are independent variables that affect the value of the output. The distance traveled will vary depending on the route chosen. We want to modify decision variables in such a way that they will help us to reach our objective function.

Depending on the route that will be chosen to visit cities, the distance will either decrease or increase. The choice of algorithm can influence the objective function by approaching the task in different ways thus being able or not being able to find a global minimum rather than local minima. Some algorithms spend more computational power while others spend less and we have to account for that.

Constraints: fixed number of cities that should be visited. The city cannot be visited two times since if it already received the vaccine we do not need to waste resources to visit it for the second time.

1.2 Optimization Technique:

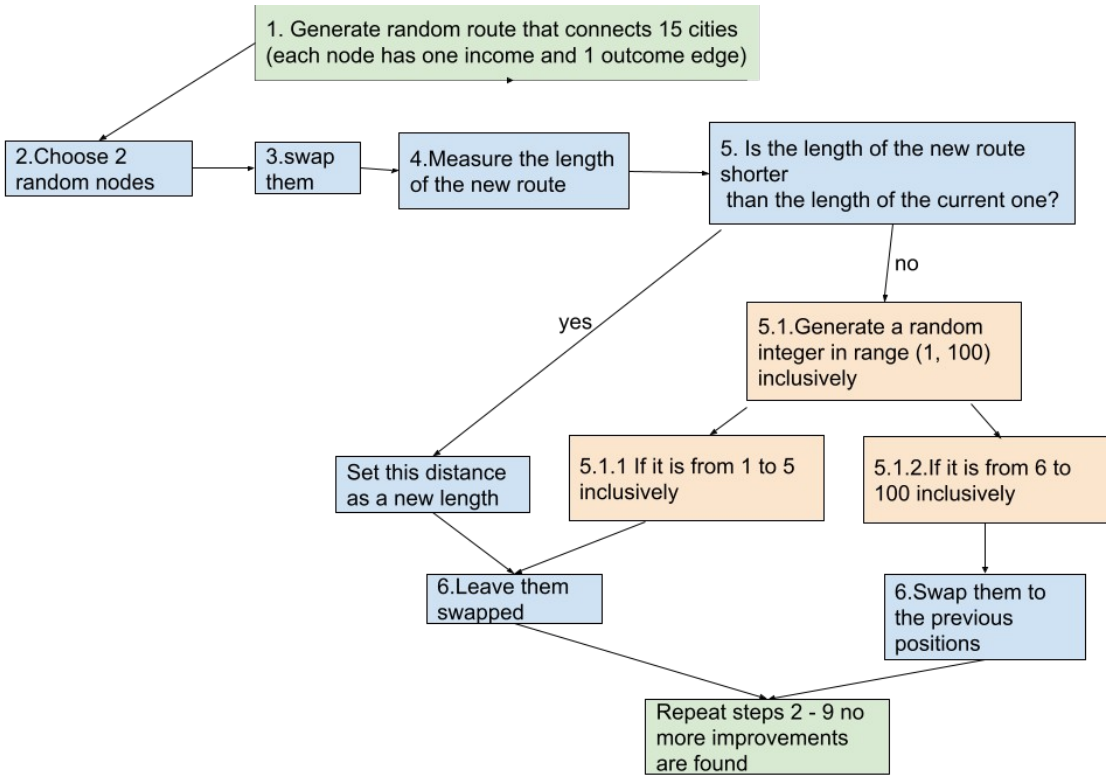
What process can be used to find the optimal solution in your scenario? Identify and describe an existing algorithm that could be used to complete this process, including the inputs, outputs, required steps, and the termination condition. Explain the advantages and limitations of this algorithm. In your explanation, you should address whether your algorithm would lead to the global optimum and you may wish to compare your algorithm with other possible optimization techniques. (~200 words)

- *Optional*: Draw a flowchart to illustrate the process.
- *Optional challenge*: Create a program in Python to implement this optimization process. Please provide a thorough explanation of how the code works, both holistically and using in-line comments. You must also provide at least one "test case" that demonstrates that your code is properly implemented. This test should be something like, "In this case, it's clear that the only possible solution is X; let's check that with our code. Yes, the code outputs X as well, so we can in principle induce that the implementation is correct."

```
I will use 2-opt swap and simulated annealing algorithm.
It swaps 2 cities in a randomly generated route and compares the new
route to the previous one.
If the new total distance is shorter, the current route is updated.
The algorithm builds on the improved route and repeats the steps until
no more improvements are found.
```

Simulated annealing adds randomization of swapping two cities when it does **not** produce a shorter route.
It helps to prevent algorithms from getting stuck **in** suboptimal solutions by pushing them to look **for** other solutions that would have been missed without randomization.
Although this approach does **not** guarantee the best solution, it **is** the best algorithm balancing the computational power needed **for** execution **and** closeness to the **global** minima.
Brute force considers 15! = 1,307,674,368,000 routes which **is** time consuming.
Greedy algorithm will **not** get as close to the **global** maxima as the 2-opt swap.
It fails to find the globally optimal solution because it does **not** consider all the data.
The choice made by a greedy algorithm may depend on choices it has made so far, but it **is not** aware of future choices it could make, therefore it **is not** effective.
Once the greedy algorithm finds the next best solution it stops looking further **and** gets stuck on sub-optimal solutions.

```
from IPython.display import Image
Image(r"C:\Users\tetia\Downloads\Flow chart traveling salesman
CS.png")# replace FILE_NAME_2.2_1.png with your image file
```



PART 2: SIMULATION

The SIR model of the spread of disease is commonly used to help understand how a disease might move through a population. You were introduced to this with the NetLogo agent-based model in NS50 and will review it again in Weeks 7 and 8 of CS51. Check out the class readings to learn about this model.

For this assignment, you will select one disease of your choice to model. Please choose a disease from THIS LIST to investigate. If you would like to select your own disease to model, you may email your professor with the disease and parameter descriptions for approval. You must select an infectious disease (one that is transmitted from person to person through a viral, bacterial, or parasitic agent), not a genetic or environmental disease.

Part 2.1 Numerical Modeling and Simulation

For this part of the assignment, you'll consider the SIR model described by the set of differential equations below, and the numerical simulation in Python via Euler's method.

$$\frac{dS}{dt} = -\frac{b}{N} S(t) I(t)$$

$$\frac{dI}{dt} = \frac{b}{N} S(t) I(t) - k I(t)$$

$$\frac{dR}{dt} = k I(t)$$

2.1.1 Variables and Parameters (~250 words) [#variables]

This section serves to set up an initial analysis of the SIR model.

1. State the disease you selected to model.
2. Identify the relevant **variables** of the model, fully classify what type of variables they are, and explain what they mean in the context of your model. Next, identify appropriate numerical values and units for the **initial values** of the variables. You're encouraged to use empirical data if possible to justify these values. You may also complete a well-reasoned #estimation for any values that are difficult to justify with empirical data. Include APA citations for any external sources used. Note that you can work with population values S, I, R, or proportions, S/N, I/N, R/N, as long as you are consistent.

3. Explain what the relevant **parameters** (b and k) are and what they mean. For your model, identify and justify appropriate numerical values and units for b and k. As above, you may include a well-reasoned estimate using empirical data to support your justification. Further, explain what it would mean for the parameters (b and k) to be smaller or larger. Consider what real-world factors, in the context of the disease you selected, would reduce or increase these parameters.

4. *Optional*: Modify the basic SIR model to add a layer of real-world complexity. A few ideas are listed below. Explain the key features of the extended model, including the modified differential equations and a full description and classification of any new variables and parameters following the steps above.

- Vaccination
- Antibiotic use and/or development of antibiotic resistance
- Variability in population susceptibility (e.g. children and the elderly have different rates of infection compared to young adults).
- Birth and death rates in the population

The disease selected for modeling is COVID-19 N - population parameter. Since for this model, we will consider the situation in the US on the 31st of January N = 330 000 000. I - infected and transmit the disease. I(0) = 13 934 766 (World Health Organization, 2021) S - susceptible. Since there are no people with innate immunity for Covid this variable initial value will be N - I(0) - R(0) = 273691498. R - removed: recovered or died; do not transmit the disease. R(0) is 42 373 736 (World Health Organization, 2021) Variables S, I, R are discrete in their nature since they represent humans. However, here they will be considered continuous. Differential equations describe continuous models. Since our model has the algorithm that is set by the differential equations of Euler's method we need continuous variables to get the best results for interpretation, a smooth curve on the graph. However, when we apple Euler's method variables become discrete again because of the step size which has a specific value that separates our points.

Considering the relationship between the variables, S, R, I all are dependent variables. The independent variable is t which is time represented in days. So, we are deriving S, I, and R as the functions of time.

Parameters for the simulation are:

k = 0.085 - estimation of infectious rate, the ratio of individuals that are transferred from the susceptible population to infectious every day. It is the number of people that an infected person could infect in 1 day.

b = 0.062 - estimation of recovery rate. Fraction of an infected group that recovers every day or probability of 6.2% for a sick person to recover in a given day. This gives the approximation of the infectious period of 16 days.

An increase in b could be influenced by the violation of social distancing and mask norms that serve as a prevention for spreading the Covid-19 disease. A change in k could be

influenced by the change of rate humans produce antibodies that is related to virus mutations and vaccine development.

References: World Health Organization. (2021). United States COVID - Coronavirus Statistics - Worldometer. Wordometer.
https://www.worldometers.info/coronavirus/country/us/

Optional:

I add an additional variable v - vaccination rate.
This variable has been increasing at a steady rate from 62% on 12/31/2021 to 64,9% on 28/02/2022.
So v(0) = 0.62. If step size 1 day, v(59) = 0.649(WHO, 2022). Formula for the equation of v is $v=0.00049t+62$.
Positive slope signifies an increasing vaccination trend. This variable negatively influences infections rate,
so the $dI/dt=bS(t)I(t)-kI(t)$ becomes $dI/dt=(b/v)*(S(t)I(t)-kI(t))$

References:

WHO. (2022). U.S. COVID-19 vaccine tracker: See your state's progress.
Mayo Clinic. https://www.mayoclinic.org/coronavirus-covid-19/vaccine-tracker

2.1.2 Euler's Method Description (~150 words) [#algorithms]

Explain what it means to solve the SIR differential equations and how Euler's method works as an algorithm to achieve a numerical solution via simulation. In your explanation, identify the inputs, outputs, and steps that the algorithm takes, and consider the role of the step size (h) in the algorithm.

The input in the algorithm of Euler's method is the initial values for S, I, R variables. the value of independent variable time(t_end) as the termination condition for the algorithm, h - step-size parameter. The smaller h is the more correct estimation the algorithm can produce since it takes more measurement on a limited scale. However, it should not be too small since it will require significant computational power while will not improve the prediction accuracy a lot.

When we solve the derivative equation for SIR, we get the formula that helps to model the behavior of the variables in the system set by the equation. The output of the model based on Euler's equations is the set of predicted values for each step (smaller stepsize = more data points in the output).

It works like an algorithm since it takes input, converts it to the prediction of the value, and outputs it. The algorithm repeats this prediction S[n+1] based on S[n] on every step on the defined time interval.

2.1.3 Euler's Method Implementation [#algorithms, #dataviz]

Define a function that implements a numerical simulation to derive the implications of your model using Euler's method in Python. Your simulation must generate at least one relevant visualization of the disease dynamics (see required analysis below), including a descriptive figure legend and caption. You may need to adjust the run-time and step size in your simulation to ensure the visualization is maximally informative. Include thorough comments in your code to convey your understanding of the implementation of Euler's method.

```
import numpy as np  # import the libraries needed numpy for
mathematical calulations and array setup, matplotlib for graph
generation
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 15})  #set parametrs for the graph:
update frontsize to 15 and figure size to 8.5
plt.rcParams["figure.figsize"] = [8,5]

def SIR_Euler(b,k,initial_conds): #define function ht will generate
the solution based on the algytythms of Euler differential equations
    t0 = 0 # the strating point for the x-axis, the first day of the
simulation
    t_end = 200 # final time, so t0 and t_end form the range for the
x-axis

    h = 0.1 # step size (time interval)
    steps = int((t_end - t0)/h + 1) # number of steps

    # variables:
    t = np.linspace(t0, t_end, steps) # storing t values in equally
spaced slots over the Y-axis interval(from t0 till t_end)
    S = np.zeros(steps) # for storing S values in the numpy array
willed with zeros initially
    I = np.zeros(steps) # for storing I
    R = np.zeros(steps) #for sorting R

    # initial conditions:
    S[0] = initial_conds[0] #set the position in the list for the
values of initial positions that will be substracted with initial
values.
    I[0] = initial_conds[1]
    R[0] = initial_conds[2]
    N = S[0] + I[0] + R[0] #Number of total population that is the sum
of 3 possible states of variable
    ## Euler's method
    for n in range(steps-1): # range(start, stop, step)
        S[n+1] = S[n] - b*h*S[n]*I[n]/N #Counting the number of
susceptible individuals in the population by applying the Euler's
method formula. This algorithm is the basis of the model
        R[n+1] = R[n] + h*(k*I[n])
```

```
        I[n+1] = I[n] + h*((b*S[n]*I[n])/N - k*I[n]) #Counting the
number of infecteded individuals in the population


    plt.plot(t,S,linewidth=2,label='S(t)') #building line plots for
each varibale
    plt.plot(t,I,linewidth=2,label='I(t)')
    plt.plot(t,R,linewidth=2,label='R(t)')
    plt.xlabel('t [days]') #adding lables, title
    plt.ylabel('S,  I, R')
    plt.title("SIR Model for COVID-19 spread in the US \n for 200 days
from 31.12.21 till 19.06.22")
    plt.legend(loc='best') #specified the place for legend as best
means that the algorithm will position it in the place that has the
most free space avaliable
    plt.show() #show the plot

# parameters:
infection_rate = 0.085 # Beta- Covid infection rate(rate of transition
from S to I)
recovery_rate = 0.062 #K- Covid revocery rate(rate of transition from
I to R)

# initial conditions:
S0 = 2.73691498*(10**9)#intial number of susceptible people
I0 = 1.39347661*(10**8) #initial number of infected people
R0 = 4.2373736*(10**7) #initial number of recovered people
initial_vals = [S0,I0,R0] #here we store the initial conditions that
are specific to this case in the list for initial values. We will
substract the list of initial conditions with initial values. This is
done to make the function adjustable to any set of initial conditions

# call the function to run the simulation
SIR_Euler(b=infection_rate, k=recovery_rate,
initial_conds=initial_vals)
print('Figure 1: The figure illustrates the change of variables S, I,
R on the Y-axis over time(days) on the X-axis. \ninfection rate =
0.085, recovery rate = 0.062\nThe number of susceptible people
decreases as the number of infectious people decreases. \nThe number
of removed people increases as the number of infected people
decreases. ')
#print the captions of the graph, add '\n' to break lines and add
logical structure
```

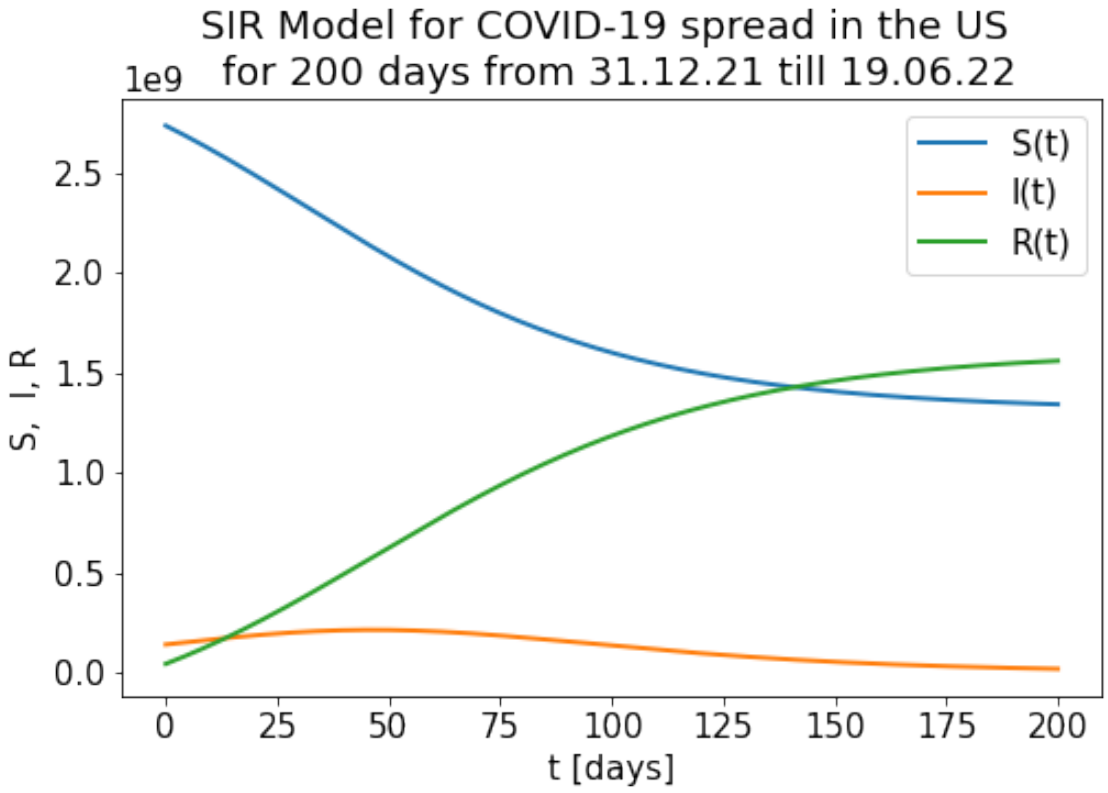


Figure 1: The figure illustrates the change of variables S, I, R on the Y-axis over time(days) on the X-axis.
infection rate = 0.085, recovery rate = 0.062
The number of susceptible people decreases as the number of infectious people decreases.
The number of removed people increases as the number of infected people decreases.

2.1.4 Results and Interpretation (~250 words) [#modeling, #dataviz]

- Interpret the results of the numerical simulation by making reference the output in the visualization(s). To fully interpret the results, you should re-run the simulation above multiple times with varying parameter inputs (b and k) and observe the behavior of your model. Include at least two additional visualizations here to support your answer. Does the behavior align with what you would expect these adjustments to have in reality (given your answer to 2.1.1.3 above)?

- *Optional*: include at least one multidimensional phase space plot and provide a full interpretation of what it shows.

- Explain how useful this model is by considering the following guiding questions: What insights can be gained? How closely do the results match what you'd expect in reality? What are the most notable assumptions of this model and what impact do they have on its usefulness?

```
SIR_Euler(b=0.55, k=recovery_rate, initial_conds=initial_vals)
print('Figure 2: The figure illustrates the change of variables S, I,
R on the Y-axis over time(days) on the X-axis\ninfection rate = 0.55,
recovery rate = 0.062')
```

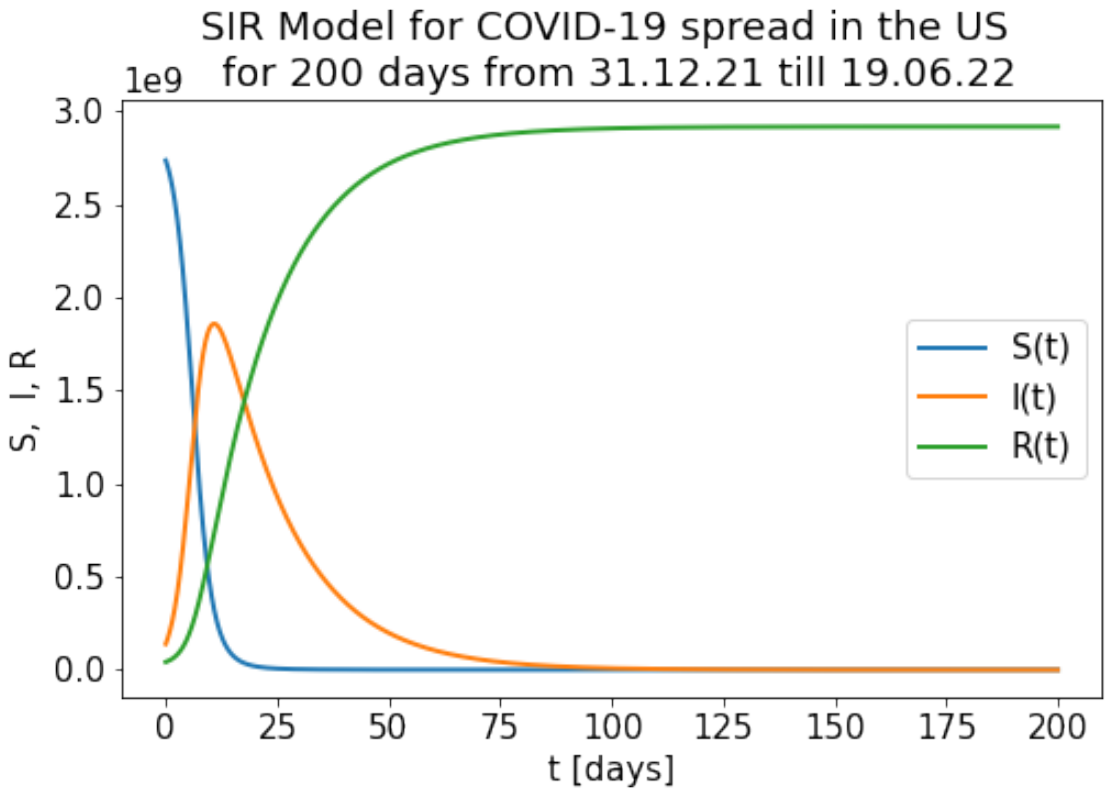


Figure 2: The figure illustrates the change of variables S, I, R on the Y-axis over time(days) on the X-axis
infection rate = 0.55, recovery rate = 0.062

```
SIR_Euler(b=0.55, k=0.01, initial_conds=initial_vals)
print('Figure 3: The figure illustrates the change of variables S, I,
R on the Y-axis over time(days) on the X-axis\ninfection rate = 0.55,
recovery rate = 0.01')
```

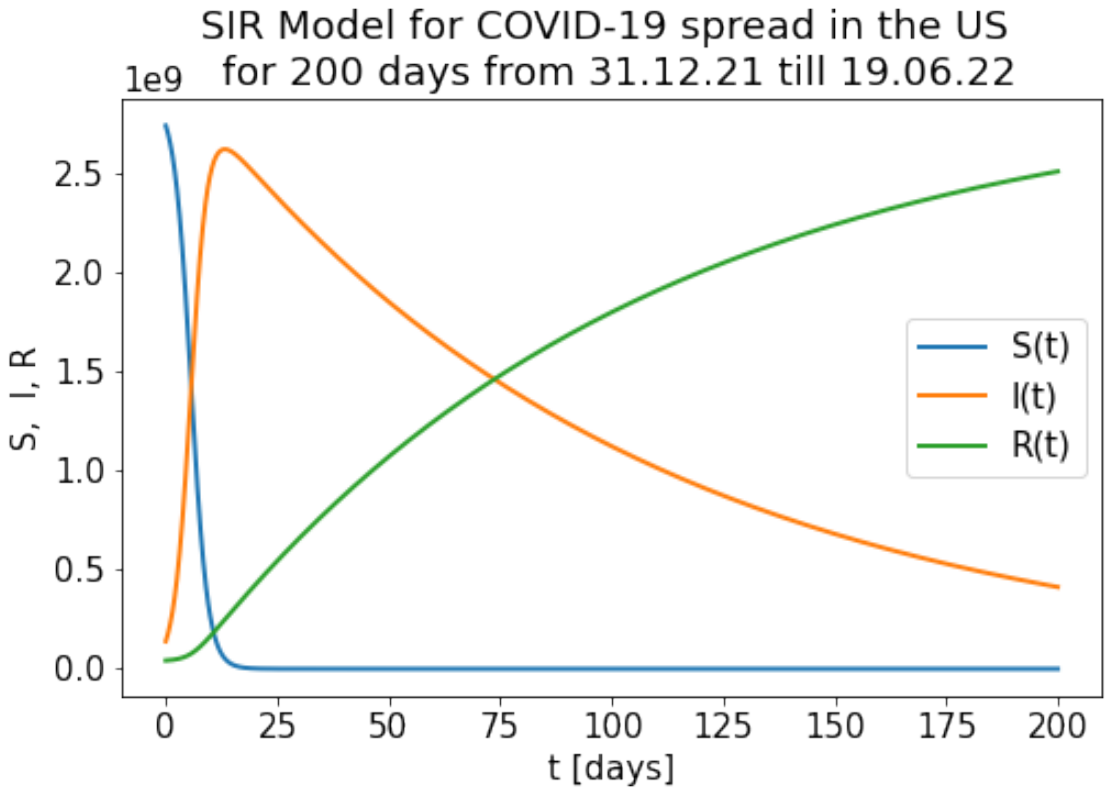


Figure 3: The figure illustrates the change of variables S, I, R on the Y-axis over time(days) on the X-axis
infection rate = 0.55, recovery rate = 0.01

```
SIR_Euler(b=0.05, k=0.01, initial_conds=initial_vals)
print('Figure 4: The figure illustrates the change of variables S, I,
R on the Y-axis over time(days) on the X-axis\ninfection rate = 0.05,
recovery rate = 0.01')
```

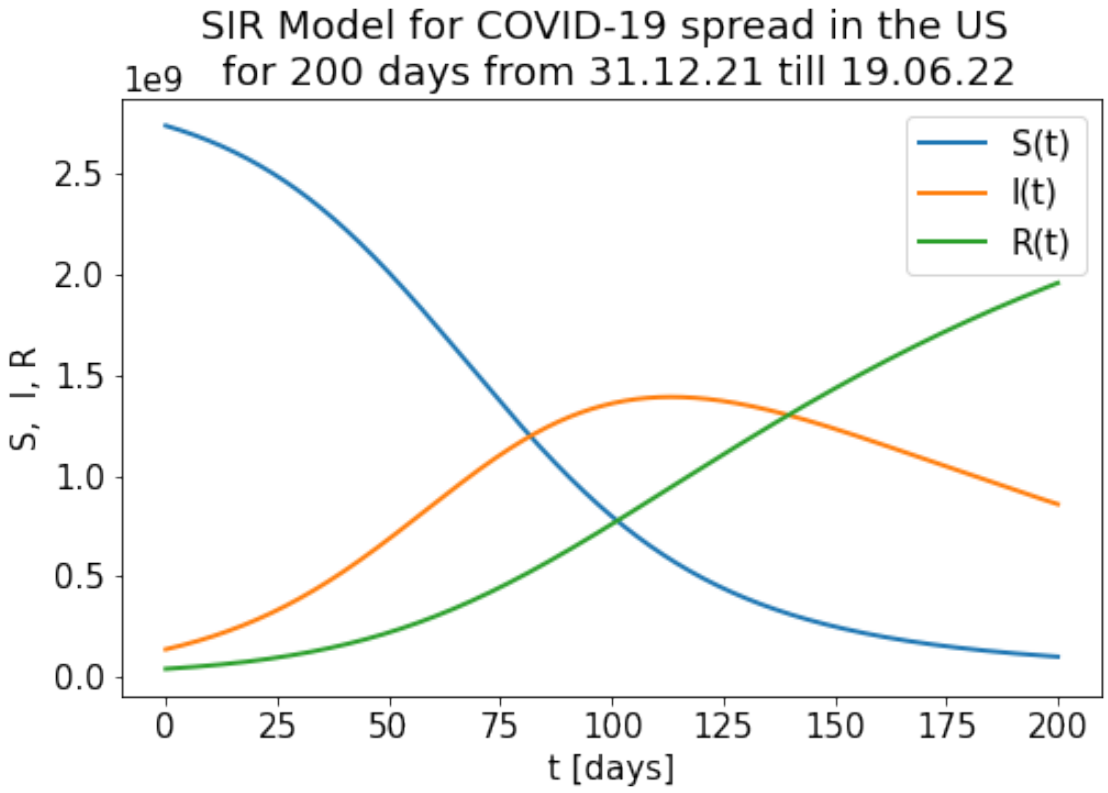


Figure 4: The figure illustrates the change of variables S, I, R on the Y-axis over time(days) on the X-axis
infection rate = 0.05, recovery rate = 0.01

```
SIR_Euler(b=0.55, k=0.5, initial_conds=initial_vals)
print('Figure 5: The figure illustrates the change of variables S, I,
R on the Y-axis over time(days) on the X-axis\ninfection rate = 0.55,
recovery rate = 0.5')
```

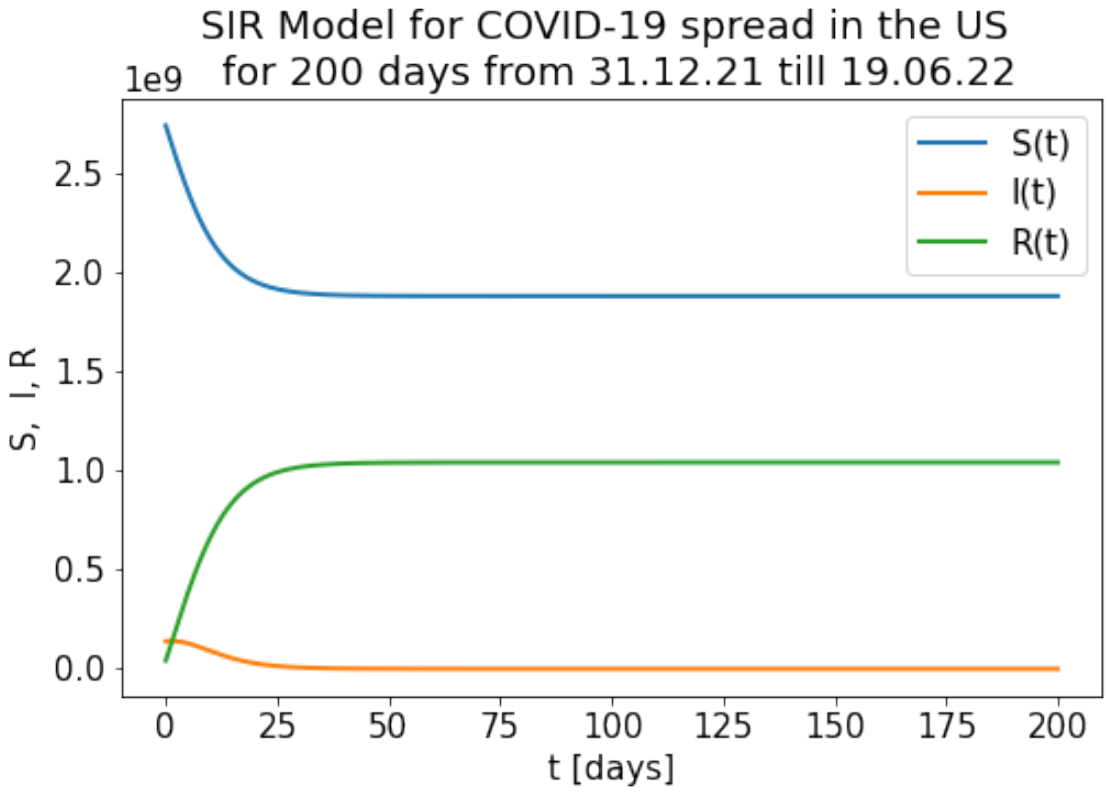


Figure 5: The figure illustrates the change of variables S, I, R on the Y-axis over time(days) on the X-axis
infection rate = 0.55, recovery rate = 0.5

```
SIR_Euler(b=0.01, k=0.1, initial_conds=initial_vals)
print('Figure 5: The figure illustrates the change of variables S, I,
R on the Y-axis over time(days) on the X-axis\ninfection rate = 0.01,
recovery rate = 0.1')
```

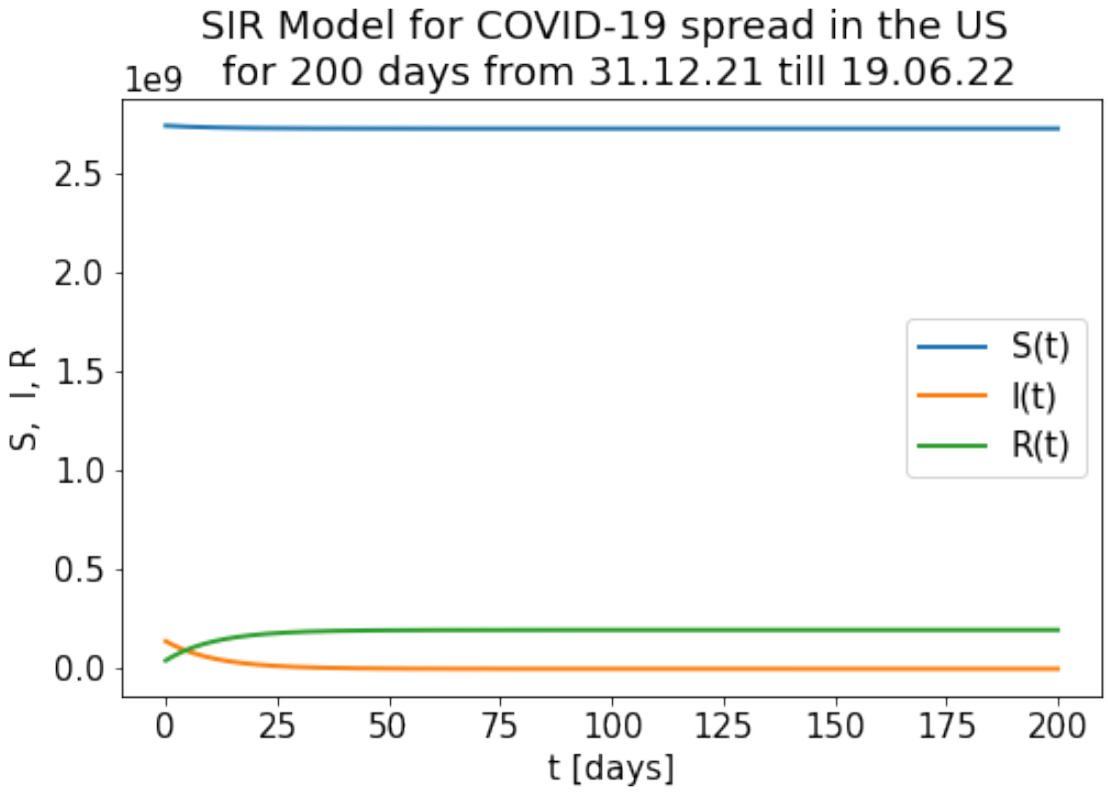


Figure 5: The figure illustrates the change of variables S, I, R on the Y-axis over time(days) on the X-axis
infection rate = 0.01, recovery rate = 0.1

As most people recovered from the disease and got immunity against the virus, R had the largest value by the end of the simulation (Figure 1) As predicted, a higher infectious rate results in a higher number of infected and increases the peak value of the outbreak as I rapidly increases due to the faster disease spread (figure 2). A lower recovery rate prolongs epidemic by affecting durability of the disease and the infectious period, consequently increasing the I value (figure 3). Even though the recovery rate is low, due to the decrease in the infectious rate the increase of I slowed down and the scale of the outbreak was diminished by decreasing the maximum value of I(figure 4). Even if the disease is highly contagious, the epidemic can be avoided if the recovery rate is high. When the time window to infect susceptible decreases so does the probability of the disease spread (figure 5). When both the recovery rate and the infectious rates are low, the spread of the disease is low. Hovever, the removed variable is also low and the largest part of the population is still susceptible(figure 6).

The model's prediction does not match the real data of Covid dynamics in the US after the 31st of December 2021(it did not decrease). Models will always have simplifications based on assumptions since social systems are too complex to track interactions between all the relevant variables. SIR model assumes that the population is homogeneous which is not applicable in the real world since, due to different health conditions, some people are more susceptible to Covid-19.

Part 2.2 Agent-Based Modeling and Simulation (*OPTIONAL*)

This part of the assignment is optional and will only be scored if completed effectively (score of 4 or 5). It is a valuable chance to compare the simulation above with the agent-based simulation implemented in NetLogo.

Note about parameters: this model uses similar variables and parameters as the one above, but the parameters are not defined identically. In particular, the "Infectiousness" parameter in NetLogo is analogous to, but not equivalent to the infection rate in the SIR model. Thus, they should not be set to the same value in both of your simulations. The infection rate in the SIR model already incorporates the interaction rate of individuals, while the NetLogo simulation sets that rate separately. In other words, the "infectiousness" parameter in NetLogo only dictates the probability of infecting someone if they come close enough, but does not take into account how frequent those interactions occur. Be sure to investigate the meaning of the other parameters as well so that you understand how to set them appropriately.

2.2.1 Optional: Simulation Comparison (~250 words) [#modeling]

After fully exploring the NetLogo model and running multiple simulations, summarize how it compares to your Python SIR simulation above. Aim to identify the main similarities, the major differences, and at least advantage for each one. Comment on which you believe to be a more realistic representation of nature, justifying your reasoning.

The main similarities of the SIR differential equation model and NetLogo model are that the variables are the same and the parameters are similar. B in NetLogo is a part of b in SIR since it is only contact rate, while in SIR, it is contact + probability to get infected after the contact. So the input does not vary much in both approaches. Although both models are approaching the same system, they have different approaches and focus on different levels of analysis: SIR is based on the top-down approach. It estimates the mean behavior at a population (global) level, thus modeling populations and not individuals. NetLogo model is based on the bottom-up approach, which works at an individual level. Agents have a set of rules to follow. So, the global behavior in NetLogo is an emergent property that emerges from interactions among agents. This behavior is unpredictable, as it does not follow linear rules (Chiacchio, F at al, 2014). SIR is deterministic and has specified infections and recovery rates, while NetLogo is stochastic and includes probability. So NetLogo will produce a different result each time the model is run, while with SIR, we will get the same output if the input is identical. NetLogo is more useful for analyzing infections spread in the common point of meeting(supermarket) when the social distancing parameter is different. SIR is more helpful for estimating the Covid - 19 for the next month in Ukraine to take preparative measures(here, the information about individuals' interactions will not add significant value to investigating the research question). References: Chiacchio, F., Pennisi, M., Russo, G., Motta, S., & Pappalardo, F. (2014). Agent-based modeling of the immune system: NetLogo, a promising framework. BioMed research international, 2014, 907171. https://doi.org/10.1155/2014/907171

2.2.2 Optional challenge: Your own agent-based simulation [#algorithms, #modeling]:

Create your own agent-based simulation of the disease dynamics for your chosen disease in Python. You may add in real-world complexities as desired (vaccination, antibiotic use and/or development of antibiotic resistance, variability in population susceptibility). Your work needs to be explained in sufficient detail, including citations to any external sources consulted, in order to receive credit.

- One option: a tree graph can be useful in modeling person-to-person interactions.
- Another option: turtles.

```
# Add code to complete the optional implementation
# Add more cells as needed to explain your work
```

REFLECTION

In less than 100 words, explain how this unit has enhanced your view of the power of modeling, algorithms, and simulations to describe the natural world around you.

Knowledge of algorithms and simulations as the key components of modeling is helpful to better understand the processes in the natural world. Modeling shows the best place where the intervention will be the most effective use of resources with the highest impact on the dynamics of the system overall. For example, knowing that decreasing the infectious rate in Covid modeling is enough to decrease the devastative effect of the pandemic is helpful to implement preventative measures like social distancing to artificially reduce the infectious rate.

You're done!

You must upload TWO files:

1. A **PDF** of your entire assignment. This is to be submitted as a separate file, NOT simply inside the zipped folder. Email attachments will not be accepted. We encourage students to follow the tips available in this guide, especially the best practices listed at the end.
2. A **zipped folder** containing the .ipynb file and any other relevant files for running the notebook.