COE 322
Infectious Disease Simulation
Daniel Reta Antonio
UT EID: dr37277

Introduction

The simulation of infectious diseases within populations holds paramount importance in understanding and forecasting the spread, impact, and potential mitigation strategies against various contagious illnesses. In this project, a simulation program was developed to model the spread of infectious diseases within a population, considering factors such as transmission probability, vaccination, and the duration of the disease. This write-up provides an overview of the code developed, its functionalities, and the experiments conducted to analyze disease transmission dynamics.

Code Description

The implemented code comprises a simulation framework that simulates the spread of an infectious disease among a population of individuals. The simulation involves individuals represented as 'persons' within a grid-based environment. These individuals can exist in four distinct health states: SUSCEPTIBLE, SICK, RECOVERED, and VACCINATED. The code simulates the progression of the disease through these states over a specified period. Key features of the code include:

- Population Class: Manages the collection of individuals, assigns coordinates within the grid, and tracks their health status.
- Person Class: Represents individuals with functionalities to infect, vaccinate, progress health states, and infect neighboring individuals based on transmission probability.
- Disease Class: Defines the characteristics of the disease, including its duration and transmission probability.
- Simulation Loop: Runs the simulation for a specified number of days, tracking the spread of the disease within the population.

Exercises (Experiments) and Results

1. Exercise 49.1

Exercise 49.1. Write a `Person` class with methods:

- `status_string()` : returns a description of the person's state as a `string`;
- `one_more_day()` : update the person's status to the next day;
- `infect(s)` : infect a person with a disease, where the disease object

```
Disease s(n);
```

is specified to run for n days.

This exercise implements a Person class with the methods mentioned above in the exercise description. The output is the following, in this case the disease lasts 14 days.

```
On day 1, Jo is sick
On day 2, Jo is sick
On day 3, Jo is sick
On day 4, Jo is sick
On day 5, Jo is sick
On day 6, Jo is sick
On day 7, Jo is sick
On day 8, Jo is sick
On day 9, Jo is sick
On day 10, Jo is sick
On day 11, Jo is sick
On day 12, Jo is sick
On day 13, Jo is sick
On day 14, Jo is sick
On day 15, Jo is recovered
```

2. Exercise 49.2

Exercise 49.2. Add a transmission probability to the `Disease` class, and add a `touch` method to the `Person` class. Design and run some tests.

This exercise implements the feature that a sick person can infect other people. The day that this happens will depend on the transmission probability. The output of this exercise is the following:

```
On day 1, Alice is susceptible
On day 1, Bob is sick (4 days to go)
On day 2, Alice is susceptible
On day 2, Bob is sick (3 days to go)
Alice got infected by Bob
On day 3, Alice is sick (4 days to go)
On day 3, Bob is sick (2 days to go)
On day 4, Alice is sick (3 days to go)
On day 4, Bob is sick (1 days to go)
On day 5, Alice is sick (2 days to go)
On day 5, Bob is recovered
On day 6, Alice is sick (1 days to go)
On day 6, Bob is recovered
Both Alice and Bob have recovered!
```

3. Exercise 49.4

Exercise 49.4. Program a population without infection.

- Write the `Population` class. The constructor takes the number of people:

```
Population population(npeople);
```

- Write a method that infects a number of random people:

```
// /pandemic.cpp
    population.random_infection(fever,initial_infect);
```

- Write a method `count_infected` that counts how many people are infected.
- Write an `one_more_day` method that updates all persons in the population.
- Loop the `one_more_day` method until no people are infected: the `Population::one_more_day` method should apply `Person::one_more_day` to all person in the population.

Write a routine that displays the state of the popular, using for instance: ? for susceptible, + for infected, − for recovered.

This exercise implements a population class using vectors. This program displays the health status of every member of the population everyday. The disease is set to last 5 days, since there we are assuming no contact, only one person gets sick and the disease runs its course by the last day.

```
In day 1 #sick: 1 : ? ? ? ? ? ? ? ? ? +
In day 2 #sick: 1 : ? ? ? ? ? ? ? ? ? +
In day 3 #sick: 1 : ? ? ? ? ? ? ? ? ? +
In day 4 #sick: 1 : ? ? ? ? ? ? ? ? ? +
In day 5 #sick: 1 : ? ? ? ? ? ? ? ? ? +
Disease ran its course by day 5
```

4. Exercise 49.5

> **Exercise 49.5.** Write a simulation where in each step the direct neighbors of an infected person can now get sick themselves.
>
> Run a number of simulations with population sizes and contagion probabilities. Are there cases where people escape getting sick?

Exercise 49.5 differs from the previous exercise because this time, only direct neighbors of an infected person can get sick. After running a number of simulations, there are cases where some people are successful at avoiding getting infected. As it can be seen on the output, the disease does not stop until the last person has recovered. This output presented below, is one of those cases where several people avoided getting infected.

```
In day 1 #sick: 1 : ? ? ? ? ? + ? ? ? ?
In day 2 #sick: 1 : ? ? ? ? ? + ? ? ? ?
In day 3 #sick: 2 : ? ? ? ? + + ? ? ? ?
In day 4 #sick: 2 : ? ? ? ? + + ? ? ? ?
In day 5 #sick: 3 : ? ? ? + + + ? ? ? ?
In day 6 #sick: 2 : ? ? ? + + - ? ? ? ?
In day 7 #sick: 3 : ? ? + + + - ? ? ? ?
In day 8 #sick: 3 : ? + + + - - ? ? ? ?
In day 9 #sick: 3 : ? + + + - - ? ? ? ?
In day 10 #sick: 2 : ? + + - - - ? ? ? ?
In day 11 #sick: 2 : ? + + - - - ? ? ? ?
In day 12 #sick: 2 : + + - - - - ? ? ? ?
In day 13 #sick: 1 : + - - - - - ? ? ? ?
In day 14 #sick: 1 : + - - - - - ? ? ? ?
In day 15 #sick: 1 : + - - - - - ? ? ? ?
In day 16 #sick: 1 : + - - - - - ? ? ? ?
Disease ran its course by day 16
```

5. Exercise 49.6

> **Exercise 49.6.** Incorporate vaccination: read another number representing the percentage of people that has been vaccinated. Choose those members of the population randomly.
>
> Describe the effect of vaccinated people on the spread of the disease. Why is this model unrealistic?

Exercise 49.6 incorporates vaccination status to different members of the population. After running a number of simulations, the effect that vaccinated people have on the spread of the disease is such that of a barrier. As it can be seen on the following sample output, the right-most person is the person that got infected first. The disease spreads to the people directly next to patient zero. However, the fourth to last person is vaccinated and prevents the disease from spreading to the left side of the population.

```
In day 1 #sick: 1 : ? ? ? ? * ? * ? ? +
In day 2 #sick: 2 : ? ? ? ? * ? * ? + +
In day 3 #sick: 3 : ? ? ? ? * ? * + + +
In day 4 #sick: 3 : ? ? ? ? * ? * + + +
In day 5 #sick: 3 : ? ? ? ? * ? * + + +
In day 6 #sick: 2 : ? ? ? ? * ? * + + -
In day 7 #sick: 1 : ? ? ? ? * ? * + - -
Disease ran its course by day 7
```

6. Exercise 49.7

> **Exercise 49.7.** Code the random interactions. Now run a number of simulations varying
>
> - The percentage of people vaccinated, and
> - the chance the disease is transmitted on contact.
>
> Record how long the disease runs through the population. With a fixed number of contacts and probability of transmission, how is this number of function of the percentage that is vaccinated?
>
> Report this function as a table or graph. Make sure you have enough data points for a meaningful conclusion. Use a realistic population size. You can also do multiple runs and report the average, to even out the effect of the random number generator.

This exercise implements random interactions. This means that now to infect people, susceptible people do not necessarily have to be next to infected people. As a general rule, the fewer interactions and the lower the probability of transmission that each person has the longer it takes for the disease to run through the population. The higher the percentage of people that are vaccinated, the sooner it takes the disease to run through the population as there is fewer people to infect.

```
In day 1 #sick: 1 : ? * ? ? ? * * ? + *
In day 2 #sick: 3 : ? * + + ? * * ? + *
In day 3 #sick: 6 : + * + + + * * + + *
In day 4 #sick: 6 : + * + + + * * + + *
In day 5 #sick: 6 : + * + + + * * + + *
In day 6 #sick: 5 : + * + + + * * + - *
In day 7 #sick: 3 : + * - - + * * + - *
Disease ran its course by day 7
Total infected: 6
```