W6.2 Applied

Simulate a Pandemic

In this activity, you will write a (very simple!) Python program to simulate the spread of a disease among a group of people over some time period. Namely, the code will simulate the spread of the disease from infected individuals to non-infected individuals with the help of `random` Module over a user-specified number of days.

But first, as a warm up, simulate

- a coin flip and
- a dice roll below

with the help of the `random` Module.

Example 1

```
Tails!
You have rolled: 4
```

Example 2

```
Heads!
You have rolled: 1
```

There is no marking here because the output of the program is random.

The content of this session should not be considered as health advice.

Pandemic Simulation Quiz

Question 1 *Submitted Apr 5th 2022 at 10:36:24 am*

Based on scientific evidence, what factors contribute to the spread of the disease (let us focus on COVID-19 for this instance)? Multiple answers are accepted.

- [x] transmission dynamics of the disease (e.g., person-to-person, airborne etc.)
- [x] population dynamics (i.e., interaction of people, e.g., close contact)
- [ ] colour of your clothes
- [ ] the name of your pet

Question 2 *Submitted Apr 5th 2022 at 10:36:40 am*

Why would we want to write a simulation to model the spread of a disease? Multiple answers are accepted.

- [x] To understand the disease better.
- [x] To be able to make predictions about the effects of the pandemic.
- [x] To test out different public health policies.
- [x] To help policy makers produce (potentially new) effective policies.

Assumptions

As we have seen in the quiz, both *transmission dynamics* and *population dynamics* seem to play an important role in the spread of the disease. We are now going to use this information to make some (very simplistic!) assumptions that will allow us to simulate the spread of the disease among a group people based on their daily contact information with each other over a user specified time period. Namely, we are going to assume that:

1. the disease spreads *randomly* (for simplicity, let us assume with 20% chance) among two people who have been in close contact for a given day,
2. we have access to accurate data on whether two people will be in close contact for a given day,
3. people can only transition from non-infected to infected (i.e., no other transitions are allowed),
4. people's behaviour does not change over the duration of the pandemic, and
5. no other information is relevant to the spread of the disease.

Note that the assumptions listed above are extremely simplistic – but we have to start somewhere :).

Before you begin writing the complete simulation model, we will first ask you to write a small program below that simulates the spread of the disease from an infected person to a non-infected person with the help of `random` Module (i.e., based on the first assumption above).

Your program should output `Infected!` with a probability of 20% and `Not infected!` otherwise.

Simulate the spread of the disease

Now you are ready to simulate the spread of a disease among a group people according to their daily contacts with each other over a user-specified time period based on the assumptions provided previously. Write a program in the Python file named `simple_pandemic_simulation.py` that takes in the number of days the simulation will run for and displays the number of newly infected people for each day of the simulated pandemic.

▶ Expand

Specifically, the code should perform the following tasks.

▶ Expand

Task 1 (✓)

Prompt the user for the number of days the simulation should run and store that information in the variable named `num_days`. When collecting the input, please prompt the user with the following messages:

"Please enter the number of days: "

Task 2 (✓)

Initialise a list named `is_infected` that maps the index of a person from tuple `people` to a Boolean value (i.e., `True` if the individual is infected, `False` otherwise). Initially, every person except one should be non-infected. The person that is infected in the beginning of the pandemic should be randomly chosen. Write a function named `initialise_random_infected` in the imported module `population_information` to perform this initialization.

▶ Expand

Task 3 (✓)

Initialise the close contact information of people using the imported module `population_information` using the defined function `initialise_random_contact_info`. This information, i.e., whether two individuals are close contacts or not, that maps a pair of person indices to a Boolean value (i.e., `True` if the pair of people are close contacts, `False` otherwise) should be stored in the table (i.e., list of lists) named `is_close_contact`.

▶ Expand

Task 4 (✓)

Run the simulation for `num_days` days. For each day, the simulation should:

1. check whether an infected person infects a non-infected person based on assumption 1, and
2. display the number of newly infected people for that day.

Namely in the **beginning** of every day, the code should generate a random number between 0 and 1 for each infected person that comes into close contact with a non-infected person. If the randomly generated number is less than or equal to 0.2, it should update the person as infected at the **end** of that day.

Similarly, the code should display the number of newly infected people at the **end** of that day. When providing the output, please use the following message for day `X` of the simulated pandemic for reporting the number of infected people `Y` on day `X`.

"Day X : Number of newly infected people is Y"

There should be `num_days` lines of output displayed to the user.

> There is no marking here since the output is random and different every time.

Going Further

You have (*with high probability**) written your first simulation code – congratulations! But as we have discussed earlier, we have made many assumptions in order to make our life easy. In this section, we are going to briefly discuss how we can make this simulation more realistic. Some points to consider are listed below – how would you go about changing your code to account for these more realistic views of modelling a pandemic?

- We have assumed that the only allowed transition was from non-infected to infected. In epidemiology modelling, an accepted view is the SIR compartmental model, which allows individuals to go from non-infected (a.k.a., **S**usceptible) to **I**nfected to **R**emoved (i.e., dead or recovered). We will relax this assumption in Week 8!
- We have assumed that the behaviour of people does not change throughout the pandemic. However, whether it is due to a personal choice of each person or a public health policy implemented by a higher authority (e.g., lockdowns), the behaviour of people can change throughout the pandemic (i.e., `is_close_contact` can be dynamic).
- We have assumed that no other information is relevant to the spread of the disease, and this is clearly not true. For example, we know that the disease can spread from other sources (e.g., from touching objects or surfaces that have droplets) which are not 'close contacts'.

**pun intended :)*

Feedback

Question 1

Feedback

What worked best in this lesson?

No response

Question 2

Feedback

What needs improvement most?

No response