# 

# COMP 5070 Statistical Programming for Data Science

# COVID-19 Simulation

## Assessable Exercise 1 DUE by 11:00pm (CST), Sunday 22nd August

* This exercise is a part of the continuous assessment that is worth 25% of your overall grade.
* Your code should be submitted as a single .py file using LearnOnline. Do not hardcode any paths on your computer in the code, as I should be able to load and run your code.
* The exercise is out of 100 marks. To obtain the maximum available marks you should aim to:

1. Code the requested program (60%).
2. Use a clear coding style (10%). Code clarity is an important part of your submission. Thus, you should choose meaningful variable names and adopt the use of comments - you don't need to comment every single line, as this will affect readability - however you should aim to comment at least each section of code.
3. Have the code run successfully (10%).
4. Output the information in a presentable manner as decided by yourself (10%).
5. Document code limitations including, but not limited to, the requested functionalities (10%).

This assessable exercise can be openly discussed within the group online and you are welcome to share tips and tricks (not entire programs, however).

Having said that, the ground rules are:

* If you use another person’s code in your file, please note the source and how much of the code is not yours.
* If you submit a program cobbled together by other peoples’ code with no, or little, original input from yourself, you will automatically receive a zero mark. The idea is to develop your own programming style with (or without) the help of others, however any code used should support your approach to how you write the program, not replace your own efforts.

If you’re unsure at any point, you’re welcomed to check with me.

# COVID-19 Simulation

In this exercise you should write a code to simulate COVID-19 infection in the society. Simulation is a fantastic tool to investigate multiple “what-if” scenarios without any risk for real people. Below are rules that should help you building a simulation model:

1. We assume that there are “N\_population” citizens in a population. “N\_population” is an input parameter. During development and testing you can have it small, but your code should be able to handle a reasonably large value for population size.
2. Every citizen has a health state – “healthy, sick, dead” (or you can code it as 0, 1, 2). As we start all citizens are alive and healthy. Obviously, dead citizens cannot become sick, cannot meet anyone and, as a result, cannot infect anyone.
3. To start the pandemic, you randomly mark a small number of citizens as “sick” – there can be a parameter for the number of initially infected citizens. You can start with 1 or 2 initial sick cases.
4. One iteration is one day. During the day, every citizen can meet a random number (say between 0 and 20 inclusive) of randomly selected citizens. You don’t really need to control all citizens as we are interested in meetings of sick people only.
5. Every sick citizen can stay sick and infectious for 10 days, hence you should have some counter for each sick citizen. After 10 days a sick citizen becomes healthy and stops spreading the virus.
6. During the day every sick citizen has a probability to die (mortality rate) from the disease with probability 0.2% (very low probability).
7. If a sick citizen does not die, then they can meet other people as per the rule 4 above and if they meet a healthy person, that person might become sick too and start infecting other people starting from the next day (that is an infection day is day 0 of their sickness). The probability for a citizen becoming sick (infection rate) after a contact is 30% (this is quite high).
8. You should run this simulation for a number of days (iterations) and output each day results: how many sick citizens in population in total, how many people died, how many new infections per day, something else you might find interesting or useful (e.g. R0).
9. You can run simulations for a predefined number of days, say 300 days, or till some natural outcome – all citizens get healthy, or all citizens die.
10. Keep all parameters in the beginning of the code, so you can easily change them without a need to change anything in the code.
11. Try to change parameters of the model – increase mortality rate (more dangerous virus scenario – SARS and MERC had 3% mortality); or decrease infection rate (say, 5% – wearing masks and social distance scenario); or increase the length of sickness period; or reduce the number of possible contacts for all citizens to a range between 0 and 2 (lockdown scenario). Make a brief comment (3-4 lines as comments) on what parameters have the strongest effect on the infection growth.

The main purpose of this exercise is to test your understanding of Python fundamentals: using conditions and loops, storing and changing information in memory by using lists or dictionaries, informative and meaningful outputs by print outs, creating an algorithm and build a code for it, using user-defined functions to make the code more manageable.

For this exercise, students should use only Python fundamentals from weeks 1 to 3. No advanced data structures from the following weeks or OOP are allowed. I understand that this task would be much easier with advanced tools, but I want to see how you use simple tools first.

**Hint on getting a random event for a given probability**

If you know that some event has probability 75%, e.g., coin tossing on head – this is clearly a biased coin; then you generate a random number from a uniform distribution between 0 and 1, e.g., x = random.random(). And then, if x < 0.75 is TRUE we assume coin to be on head; if FALSE we assume the coin is on tail.

Beware of a relationship between percentage and proportion in probability: 75% means 0.75; 0.2% means 0.002.