



HoP101: Session 5

Completing our simple model

Mrigank Pawagi

Indian Institute of Science

ACM-W Student Chapter, Summer 2023

```
# main.py

import random

N = 10000

# 0 -> susceptible
# 1 -> infected
# 2 -> recovered

# T = 0

state = [0] * N
state[250] = 1
state[500] = 1
state[750] = 1
state[1000] = 1
state[1250] = 1

# If you are susceptible, and have a infected neighbour, you have a 50% chance of being infected
k = 0.5 # chance of infection

# If you are recovered, you have a ___ % of recovery given that it takes 10 days to recover
d = 10 # days to recover
r = 1/d # chance of recovery

# "Average" number of people each individual has contact with per day
c = 10

print(N-5, 5, 0, sep=", ")

T = 1 # Time instance

while T < 100:
    # After 5 days, I put a Lockdown
    if T == 5:
        c = 1

    new_state = state.copy()
```

```

i = 0
while i < N:
    if state[0] == 0:
        # We now analyse its contacts
        # pick 'c' people randomly and count how many of them are infected
        # I want 'c' random elements from the list [0, 1, 2, ..., i - 1, i + 1, i + 2, ..

        pool = list(range(N))
        del pool[i]

        contacts = random.sample(pool, c) # List of indices of neighbours

        number_of_infected = 0
        for neighbour in contacts:
            if state[neighbour] == 1:
                number_of_infected += 1

        # probability of infection
        prob = 1 - (1 - k) ** number_of_infected

        if random.random() < prob:
            new_state[i] = 1

    if state[i] == 1: # Checking for recovery
        if random.random() < r:
            new_state[i] = 2

    i = i + 1

state = new_state.copy()

i = 0
nums = [0, 0, 0]
while i < len(state):
    nums[state[i]] += 1
    i += 1

print(nums[0], nums[1], nums[2], sep=", ")

T = T + 1

```