Lattice

July 4, 2024

1 Run a Disease Propagation in a lattice environment

1.1 Lattice lengends

In the latice matrix, we set 0 as a cell that is not populated, 1 as populated, and 2 as infected with the diseases.

1.2 Method

In the initial for loop, we will iterate through the matrix and then use numpy random library to determine if the cell is populated or not. After that, if the cell is populated, we will again use numpy random library to determine if that specific populated cell is infected or not.

1.3 Computing the number of infected and populated cells

To save computing processing time, we will create a list to store the number of infected and populated cells in each generation.

1.4 About Infection and Immunity in the Present Generation

If the cell is infected or recovered in that specific generation, it cant be infected

1.5 About Immunity

A cell has a 5% chance of having permanent immunity after being infected.

[297]:
```
#import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

1.6 Creating a Cell class

In each cell, we will create a class called Cell. We use this to keep track of the state of the cell. We want to include features such as the duration of the infection, the duration of immunity, and the state of the cell, whether it is populated or infected.

1.6.1 Duration of the immunity

The duration of the immunity is different for each cell. The duration ranges from 5-15 generations. ### Duration of the infection The duration of the infection is different for each cell. The duration ranges from 1-10 generations.

[298]:
```
class Cell:
    def __init__(self, state, duration, immunity):
        self.state = state
        self.duration = duration
        self.immunity = immunity
        self.infected_duration = 0
        self.immuned_duration = 0
        self.recently_infected = False # If the agent is infected in the␣
↪present generation, it cant affect any agent in that generation.
        self.recently_recovered = False
        self.perfect_immunity = False
    def check_immunity(self):
        """_summary_
        Checks if the agent is immune.
        If yes, return True.
        If no, return False
        Returns:
            _type_: _description_
        """
        if self.perfect_immunity == True:
            return True
        if self.immuned_duration > 0:
            return True
        else:
            return False
    def check_infection(self):
        """_summary_
        Checks if the agent is infected.
        If yes, return True and subtract 1 from the infected duration. If␣
↪infected_duration<0, set the state to 1 and set the state of␣
↪immuned_duration equal to immunity.
        If no, return False
        Returns:
            _type_: _description_
        """
        if self.state == 2:
            return True
        else:
            return False
    def subtract_duration(self):
        if self.infected_duration > 0:
            self.infected_duration -= 1
```

```
            if self.infected_duration == 0:
                self.recently_recovered = True
        if self.immuned_duration > 0:
            self.immuned_duration -= 1
    def check_state(self):
        self.recently_recovered = False
        self.subtract_duration()
        if (self.infected_duration == 0 and self.state == 2):
            self.state = 1
            self.immuned_duration = self.immunity
            self.recently_recovered = True
            if self.recently_recovered == True:
                if np.random.random() <= 0.05:
                    self.perfect_immunity = True
        self.recently_infected = False
    def just_infected(self):
        self.recently_infected = True
        self.infected_duration = self.duration
    def __str__(self):
        return f"State : {self.state}\nInfection duration:{self.
↪duration}\nImmunity duration:{self.immunity}\nInfected duration:{self.
↪infected_duration}\nImmuned duration:{self.immuned_duration}"
```

[299]:
```
np.random.random() < 0.05
```

[299]:
```
False
```

[300]:
```
N = 100 # a matrix of NxN environment
p0 =0.80 #probability of a certain cell in the NXN environment to have an agent
pt = [x/10 for x in list(range(1,10,1))] #probability of disease propagation,␣
↪from 0.1-0.9 with a step of 0.1
generations = 100
pt1 = pt[0]
```

[301]:
```
lattice = np.empty((N,N), dtype = object)
```

[302]:
```
infected_generation = np.zeros(100) # this gives us the number of infected␣
↪cells in each generation
susceptible_generation = np.zeros(100)
recovered_generation = np.zeros(100)
perfect_immunity_generation = np.zeros(100)
populated_generation = 0 # this gives us the number of populated cells in each␣
↪generation
```

1.7 Populate the lattice

[303]:
```
row_number = 100
col_number = 100
for row in range(row_number):
    for col in range(col_number):
        #make a class for the cell in lattice[row,,coll]
        lattice[row,col] = Cell(state = 0, duration = np.random.randint(5,15),␣
 ↪immunity = np.random.randint(1,10))
        #make a random choice with probability of 1 being 0.8 and probability␣
 ↪of 0 being 0.2
        lattice[row,col].state = np.random.choice([0,1], p = [1-p0,p0])
        if lattice[row,col].state == 1:
            lattice[row,col].state = np.random.choice([1,2], p = [0.99,0.01])
            if lattice[row,col].state == 2:
                lattice[row,col].infected_duration = lattice[row,col].duration
#This is the first generation.
# Count the number of infected cells (state == 2) and populated cells (state !=␣
 ↪0)
infected_generation[0] = sum(cell.state == 2 for row in lattice for cell in row)
# count the number of non_zero cells (state!= 0)
populated_generation = sum(cell.state != 0 for row in lattice for cell in row)
print("The number of infected cells in the first generation is: ",␣
 ↪infected_generation[0])
print("The number of populated cells in the first generation is: ",␣
 ↪populated_generation)
```

```
The number of infected cells in the first generation is:  77.0
The number of populated cells in the first generation is:  8077
```

In the code above, we have created a lattice of 100x100 cells. Each cell has a state, duration of the infection, and duration of the immunity. We can check some cells in the lattice to see the different features.

[304]:
```
print(lattice[0,1])
print(lattice[26,35])
```

```
State : 1
Infection duration:6
Immunity duration:5
Infected duration:0
Immuned duration:0
State : 1
Infection duration:5
Immunity duration:8
Infected duration:0
Immuned duration:0
```

[305]:
```
infected_states = []
for row in lattice:
    for cell in row:
        if cell.state == 2:
            infected_states.append(cell)

print(infected_states[1])
print(infected_states[2])
```

```
State : 2
Infection duration:13
Immunity duration:8
Infected duration:13
Immuned duration:0
State : 2
Infection duration:6
Immunity duration:5
Infected duration:6
Immuned duration:0
```

1.8 Neighbour of a cell

The neighbour of a cell is it's left, right, top, and bottom cell. Meaning that a cell can infect a cell in the left, right, top, bottom, upper right, upper left, bottom right, and bottom left cell.

1.9 Infecting the neighbour of a cell

To infect a neighbour of a cell, first, it needs to check if the cell is populated and not infected. If it is, we will then check if that specific cell is immune or not. Since lattice[row,col], meaning that the

upper neighbor is lattice[row-1,col]

the lower neighbor is lattice[row+1,col]

the left neighbor is lattice[row,col-1]

the right neighbor is lattice[row,col+1].

the upper right neighbor is lattice[row-1,col+1]

the upper left neighbor is lattice[row-1,col-1]

the lower right neighbor is lattice[row+1,col+1]

the lower left neighbor is lattice[row+1,col-1].

Unfortunately, python's way of pass by reference is complicated. That is why we cant use a reusable function to make our code cleaner (or so I know).

1.9.1 Algorithm for infection

The algorithm for infection is as follows: - For each cell in the lattice: - If the cell is infected: - If the upper neighbour is populated and not infected: - If the upper neighbour is not immune:

Change state based on the probability of infection (pt) Do the same for the other neighbours Check the state of each cell in the lattice and then subtract the duration of the immunity and infection.

Do this for each generation.

[306]:
```
for generation in [x+2 for x in list(range(generations-1))]:
    for row in range(row_number):
        for col in range(col_number):
            if lattice[row,col].recently_infected == True:
                continue #if the agent is recently infected, it cant affect any␣
↪agent.
            if lattice[row,col].state == 2:
                #upper neighbor
                if row > 0: #to avoid index error
                    if lattice[row-1,col].state == 1: #upper neighbour
                        if lattice[row-1,col].check_immunity() == False:
                            lattice[row-1,col].state =  np.random.choice([1,2],␣
↪p = [1-pt1,pt1])
                            if lattice[row-1,col].state == 2:
                                lattice[row-1,col].just_infected()

                #lower neighbour
                if row < row_number-1: #to avoid index error
                    if lattice[row+1,col].state == 1: #lower neighbor
                        if lattice[row+1,col].check_immunity() == False:
                            lattice[row+1,col].state =  np.random.choice([1,2],␣
↪p = [1-pt1,pt1])
                            if lattice[row+1,col].state == 2:
                                lattice[row+1,col].just_infected()

                #left neighbor
                if col > 0: #to avoid index error
                    if lattice[row,col-1].state == 1: #left neighbour
                        if lattice[row,col-1].check_immunity() == False:
                            lattice[row,col-1].state =  np.random.choice([1,2],␣
↪p = [1-pt1,pt1])
                            if lattice[row,col-1].state == 2:
                                lattice[row,col-1].just_infected()

                #right neighbor
                if col < col_number-1: #to avoid index error
                    if lattice[row,col+1].state == 1: #right neighbor
                        if lattice[row,col+1].check_immunity() == False:
                            lattice[row,col+1].state =  np.random.choice([1,2],␣
↪p = [1-pt1,pt1])
                            if lattice[row,col+1].state == 2:
                                lattice[row,col+1].just_infected()
                #upper right neighbour
```

```
                if row > 0 and col < col_number-1: #to avoid index error
                    if lattice[row-1,col+1].state == 1: #upper right neighbor
                        if lattice[row-1,col+1].check_immunity() == False:
                            lattice[row-1,col+1].state =  np.random.
↪choice([1,2], p = [1-pt1,pt1])
                            if lattice[row-1,col+1].state == 2:
                                lattice[row-1,col+1].just_infected()
                #upper left neighbour
                if row > 0 and col > 0: #to avoid index error
                    if lattice[row-1,col-1].state == 1: #upper left neighbour
                        if lattice[row-1,col-1].check_immunity() == False:
                            lattice[row-1,col-1].state =  np.random.
↪choice([1,2], p = [1-pt1,pt1])
                            if lattice[row-1,col-1].state == 2:
                                lattice[row-1,col-1].just_infected()
                #lower right neighbour
                if row < row_number-1 and col < col_number-1: #to avoid index␣
↪error
                    if lattice[row+1,col+1].state == 1: #lower right neighbour
                        if lattice[row+1,col+1].check_immunity() == False:
                            lattice[row+1,col+1].state =  np.random.
↪choice([1,2], p = [1-pt1,pt1])
                            if lattice[row+1,col+1].state == 2:
                                lattice[row+1,col+1].just_infected()
                #lower left neighbour
                if row < row_number-1 and col > 0: #to avoid index error
                    if lattice[row+1,col-1].state == 1: #lower left neighbour
                        if lattice[row+1,col-1].check_immunity() == False:
                            lattice[row+1,col-1].state =  np.random.
↪choice([1,2], p = [1-pt1,pt1])
                            if lattice[row+1,col-1].state == 2:
                                lattice[row+1,col-1].just_infected()
    #After the iteration, iterate again for checking the state and removing␣
↪duration
    for row in range(row_number):
        for col in range(col_number):
            lattice[row,col].check_state()

    infected_generation[generation-1] = sum(cell.state == 2 for row in lattice␣
↪for cell in row)
    recovered_generation[generation-1] = sum((cell.immuned_duration > 0 or cell.
↪perfect_immunity == True) for row in lattice for cell in row)
    perfect_immunity_generation[generation-1] = sum(cell.perfect_immunity ==␣
↪True for row in lattice for cell in row)
```

```
    susceptible_generation[generation-1] = sum(cell.state != 0 for row in␣
↪lattice for cell in row) - infected_generation[generation-1] -␣
↪recovered_generation[generation-1]
    print("The number of infected cells in the", generation, "generation is: ",␣
↪infected_generation[generation-1])
    print("The number of susceptible cells in the", generation, "generation is:␣
↪", susceptible_generation[generation-1])
    print("The number of recovered cells in the", generation, "generation is:␣
↪", recovered_generation[generation-1])
    print("The number of perfect immune cells in the", generation, "generation␣
↪is: ", perfect_immunity_generation[generation-1])
```

```
The number of infected cells in the 2 generation is:  126.0
The number of susceptible cells in the 2 generation is:  7951.0
The number of recovered cells in the 2 generation is:  0.0
The number of perfect immune cells in the 2 generation is:  0.0
The number of infected cells in the 3 generation is:  190.0
The number of susceptible cells in the 3 generation is:  7887.0
The number of recovered cells in the 3 generation is:  0.0
The number of perfect immune cells in the 3 generation is:  0.0
The number of infected cells in the 4 generation is:  270.0
The number of susceptible cells in the 4 generation is:  7807.0
The number of recovered cells in the 4 generation is:  0.0
The number of perfect immune cells in the 4 generation is:  0.0
The number of infected cells in the 5 generation is:  379.0
The number of susceptible cells in the 5 generation is:  7698.0
The number of recovered cells in the 5 generation is:  0.0
The number of perfect immune cells in the 5 generation is:  0.0
The number of infected cells in the 6 generation is:  486.0
The number of susceptible cells in the 6 generation is:  7575.0
The number of recovered cells in the 6 generation is:  16.0
The number of perfect immune cells in the 6 generation is:  1.0
The number of infected cells in the 7 generation is:  610.0
The number of susceptible cells in the 7 generation is:  7434.0
The number of recovered cells in the 7 generation is:  33.0
The number of perfect immune cells in the 7 generation is:  1.0
The number of infected cells in the 8 generation is:  771.0
The number of susceptible cells in the 8 generation is:  7254.0
The number of recovered cells in the 8 generation is:  52.0
The number of perfect immune cells in the 8 generation is:  2.0
The number of infected cells in the 9 generation is:  931.0
The number of susceptible cells in the 9 generation is:  7064.0
The number of recovered cells in the 9 generation is:  82.0
The number of perfect immune cells in the 9 generation is:  6.0
The number of infected cells in the 10 generation is:  1093.0
The number of susceptible cells in the 10 generation is:  6869.0
```

```
The number of recovered cells in the 10 generation is:  115.0
The number of perfect immune cells in the 10 generation is:  8.0
The number of infected cells in the 11 generation is:  1265.0
The number of susceptible cells in the 11 generation is:  6635.0
The number of recovered cells in the 11 generation is:  177.0
The number of perfect immune cells in the 11 generation is:  17.0
The number of infected cells in the 12 generation is:  1465.0
The number of susceptible cells in the 12 generation is:  6374.0
The number of recovered cells in the 12 generation is:  238.0
The number of perfect immune cells in the 12 generation is:  18.0
The number of infected cells in the 13 generation is:  1624.0
The number of susceptible cells in the 13 generation is:  6145.0
The number of recovered cells in the 13 generation is:  308.0
The number of perfect immune cells in the 13 generation is:  22.0
The number of infected cells in the 14 generation is:  1791.0
The number of susceptible cells in the 14 generation is:  5914.0
The number of recovered cells in the 14 generation is:  372.0
The number of perfect immune cells in the 14 generation is:  30.0
The number of infected cells in the 15 generation is:  1920.0
The number of susceptible cells in the 15 generation is:  5667.0
The number of recovered cells in the 15 generation is:  490.0
The number of perfect immune cells in the 15 generation is:  36.0
The number of infected cells in the 16 generation is:  2048.0
The number of susceptible cells in the 16 generation is:  5436.0
The number of recovered cells in the 16 generation is:  593.0
The number of perfect immune cells in the 16 generation is:  46.0
The number of infected cells in the 17 generation is:  2223.0
The number of susceptible cells in the 17 generation is:  5172.0
The number of recovered cells in the 17 generation is:  682.0
The number of perfect immune cells in the 17 generation is:  49.0
The number of infected cells in the 18 generation is:  2326.0
The number of susceptible cells in the 18 generation is:  4969.0
The number of recovered cells in the 18 generation is:  782.0
The number of perfect immune cells in the 18 generation is:  57.0
The number of infected cells in the 19 generation is:  2405.0
The number of susceptible cells in the 19 generation is:  4759.0
The number of recovered cells in the 19 generation is:  913.0
The number of perfect immune cells in the 19 generation is:  75.0
The number of infected cells in the 20 generation is:  2485.0
The number of susceptible cells in the 20 generation is:  4591.0
The number of recovered cells in the 20 generation is:  1001.0
The number of perfect immune cells in the 20 generation is:  89.0
The number of infected cells in the 21 generation is:  2589.0
The number of susceptible cells in the 21 generation is:  4384.0
The number of recovered cells in the 21 generation is:  1104.0
The number of perfect immune cells in the 21 generation is:  99.0
The number of infected cells in the 22 generation is:  2680.0
The number of susceptible cells in the 22 generation is:  4184.0
```

```
The number of recovered cells in the 22 generation is:  1213.0
The number of perfect immune cells in the 22 generation is:  108.0
The number of infected cells in the 23 generation is:  2702.0
The number of susceptible cells in the 23 generation is:  4054.0
The number of recovered cells in the 23 generation is:  1321.0
The number of perfect immune cells in the 23 generation is:  126.0
The number of infected cells in the 24 generation is:  2778.0
The number of susceptible cells in the 24 generation is:  3914.0
The number of recovered cells in the 24 generation is:  1385.0
The number of perfect immune cells in the 24 generation is:  136.0
The number of infected cells in the 25 generation is:  2805.0
The number of susceptible cells in the 25 generation is:  3808.0
The number of recovered cells in the 25 generation is:  1464.0
The number of perfect immune cells in the 25 generation is:  155.0
The number of infected cells in the 26 generation is:  2869.0
The number of susceptible cells in the 26 generation is:  3669.0
The number of recovered cells in the 26 generation is:  1539.0
The number of perfect immune cells in the 26 generation is:  171.0
The number of infected cells in the 27 generation is:  2973.0
The number of susceptible cells in the 27 generation is:  3504.0
The number of recovered cells in the 27 generation is:  1600.0
The number of perfect immune cells in the 27 generation is:  185.0
The number of infected cells in the 28 generation is:  3025.0
The number of susceptible cells in the 28 generation is:  3413.0
The number of recovered cells in the 28 generation is:  1639.0
The number of perfect immune cells in the 28 generation is:  201.0
The number of infected cells in the 29 generation is:  3111.0
The number of susceptible cells in the 29 generation is:  3252.0
The number of recovered cells in the 29 generation is:  1714.0
The number of perfect immune cells in the 29 generation is:  215.0
The number of infected cells in the 30 generation is:  3202.0
The number of susceptible cells in the 30 generation is:  3130.0
The number of recovered cells in the 30 generation is:  1745.0
The number of perfect immune cells in the 30 generation is:  221.0
The number of infected cells in the 31 generation is:  3253.0
The number of susceptible cells in the 31 generation is:  3004.0
The number of recovered cells in the 31 generation is:  1820.0
The number of perfect immune cells in the 31 generation is:  239.0
The number of infected cells in the 32 generation is:  3328.0
The number of susceptible cells in the 32 generation is:  2867.0
The number of recovered cells in the 32 generation is:  1882.0
The number of perfect immune cells in the 32 generation is:  255.0
The number of infected cells in the 33 generation is:  3431.0
The number of susceptible cells in the 33 generation is:  2719.0
The number of recovered cells in the 33 generation is:  1927.0
The number of perfect immune cells in the 33 generation is:  278.0
The number of infected cells in the 34 generation is:  3483.0
The number of susceptible cells in the 34 generation is:  2598.0
```

```
The number of recovered cells in the 34 generation is:  1996.0
The number of perfect immune cells in the 34 generation is:  292.0
The number of infected cells in the 35 generation is:  3486.0
The number of susceptible cells in the 35 generation is:  2553.0
The number of recovered cells in the 35 generation is:  2038.0
The number of perfect immune cells in the 35 generation is:  324.0
The number of infected cells in the 36 generation is:  3520.0
The number of susceptible cells in the 36 generation is:  2468.0
The number of recovered cells in the 36 generation is:  2089.0
The number of perfect immune cells in the 36 generation is:  338.0
The number of infected cells in the 37 generation is:  3519.0
The number of susceptible cells in the 37 generation is:  2430.0
The number of recovered cells in the 37 generation is:  2128.0
The number of perfect immune cells in the 37 generation is:  359.0
The number of infected cells in the 38 generation is:  3535.0
The number of susceptible cells in the 38 generation is:  2364.0
The number of recovered cells in the 38 generation is:  2178.0
The number of perfect immune cells in the 38 generation is:  373.0
The number of infected cells in the 39 generation is:  3547.0
The number of susceptible cells in the 39 generation is:  2305.0
The number of recovered cells in the 39 generation is:  2225.0
The number of perfect immune cells in the 39 generation is:  399.0
The number of infected cells in the 40 generation is:  3547.0
The number of susceptible cells in the 40 generation is:  2270.0
The number of recovered cells in the 40 generation is:  2260.0
The number of perfect immune cells in the 40 generation is:  423.0
The number of infected cells in the 41 generation is:  3504.0
The number of susceptible cells in the 41 generation is:  2287.0
The number of recovered cells in the 41 generation is:  2286.0
The number of perfect immune cells in the 41 generation is:  445.0
The number of infected cells in the 42 generation is:  3518.0
The number of susceptible cells in the 42 generation is:  2232.0
The number of recovered cells in the 42 generation is:  2327.0
The number of perfect immune cells in the 42 generation is:  467.0
The number of infected cells in the 43 generation is:  3487.0
The number of susceptible cells in the 43 generation is:  2249.0
The number of recovered cells in the 43 generation is:  2341.0
The number of perfect immune cells in the 43 generation is:  491.0
The number of infected cells in the 44 generation is:  3401.0
The number of susceptible cells in the 44 generation is:  2258.0
The number of recovered cells in the 44 generation is:  2418.0
The number of perfect immune cells in the 44 generation is:  515.0
The number of infected cells in the 45 generation is:  3349.0
The number of susceptible cells in the 45 generation is:  2270.0
The number of recovered cells in the 45 generation is:  2458.0
The number of perfect immune cells in the 45 generation is:  537.0
The number of infected cells in the 46 generation is:  3334.0
The number of susceptible cells in the 46 generation is:  2286.0
```

```
The number of recovered cells in the 46 generation is:  2457.0
The number of perfect immune cells in the 46 generation is:  554.0
The number of infected cells in the 47 generation is:  3312.0
The number of susceptible cells in the 47 generation is:  2245.0
The number of recovered cells in the 47 generation is:  2520.0
The number of perfect immune cells in the 47 generation is:  577.0
The number of infected cells in the 48 generation is:  3264.0
The number of susceptible cells in the 48 generation is:  2239.0
The number of recovered cells in the 48 generation is:  2574.0
The number of perfect immune cells in the 48 generation is:  604.0
The number of infected cells in the 49 generation is:  3273.0
The number of susceptible cells in the 49 generation is:  2227.0
The number of recovered cells in the 49 generation is:  2577.0
The number of perfect immune cells in the 49 generation is:  626.0
The number of infected cells in the 50 generation is:  3286.0
The number of susceptible cells in the 50 generation is:  2197.0
The number of recovered cells in the 50 generation is:  2594.0
The number of perfect immune cells in the 50 generation is:  645.0
The number of infected cells in the 51 generation is:  3243.0
The number of susceptible cells in the 51 generation is:  2284.0
The number of recovered cells in the 51 generation is:  2550.0
The number of perfect immune cells in the 51 generation is:  664.0
The number of infected cells in the 52 generation is:  3197.0
The number of susceptible cells in the 52 generation is:  2305.0
The number of recovered cells in the 52 generation is:  2575.0
The number of perfect immune cells in the 52 generation is:  681.0
The number of infected cells in the 53 generation is:  3177.0
The number of susceptible cells in the 53 generation is:  2332.0
The number of recovered cells in the 53 generation is:  2568.0
The number of perfect immune cells in the 53 generation is:  699.0
The number of infected cells in the 54 generation is:  3185.0
The number of susceptible cells in the 54 generation is:  2325.0
The number of recovered cells in the 54 generation is:  2567.0
The number of perfect immune cells in the 54 generation is:  711.0
The number of infected cells in the 55 generation is:  3214.0
The number of susceptible cells in the 55 generation is:  2309.0
The number of recovered cells in the 55 generation is:  2554.0
The number of perfect immune cells in the 55 generation is:  739.0
The number of infected cells in the 56 generation is:  3223.0
The number of susceptible cells in the 56 generation is:  2265.0
The number of recovered cells in the 56 generation is:  2589.0
The number of perfect immune cells in the 56 generation is:  765.0
The number of infected cells in the 57 generation is:  3227.0
The number of susceptible cells in the 57 generation is:  2275.0
The number of recovered cells in the 57 generation is:  2575.0
The number of perfect immune cells in the 57 generation is:  785.0
The number of infected cells in the 58 generation is:  3225.0
The number of susceptible cells in the 58 generation is:  2292.0
```

```
The number of recovered cells in the 58 generation is:  2560.0
The number of perfect immune cells in the 58 generation is:  800.0
The number of infected cells in the 59 generation is:  3262.0
The number of susceptible cells in the 59 generation is:  2249.0
The number of recovered cells in the 59 generation is:  2566.0
The number of perfect immune cells in the 59 generation is:  817.0
The number of infected cells in the 60 generation is:  3235.0
The number of susceptible cells in the 60 generation is:  2245.0
The number of recovered cells in the 60 generation is:  2597.0
The number of perfect immune cells in the 60 generation is:  844.0
The number of infected cells in the 61 generation is:  3195.0
The number of susceptible cells in the 61 generation is:  2280.0
The number of recovered cells in the 61 generation is:  2602.0
The number of perfect immune cells in the 61 generation is:  863.0
The number of infected cells in the 62 generation is:  3204.0
The number of susceptible cells in the 62 generation is:  2256.0
The number of recovered cells in the 62 generation is:  2617.0
The number of perfect immune cells in the 62 generation is:  879.0
The number of infected cells in the 63 generation is:  3232.0
The number of susceptible cells in the 63 generation is:  2237.0
The number of recovered cells in the 63 generation is:  2608.0
The number of perfect immune cells in the 63 generation is:  897.0
The number of infected cells in the 64 generation is:  3221.0
The number of susceptible cells in the 64 generation is:  2185.0
The number of recovered cells in the 64 generation is:  2671.0
The number of perfect immune cells in the 64 generation is:  917.0
The number of infected cells in the 65 generation is:  3155.0
The number of susceptible cells in the 65 generation is:  2223.0
The number of recovered cells in the 65 generation is:  2699.0
The number of perfect immune cells in the 65 generation is:  935.0
The number of infected cells in the 66 generation is:  3110.0
The number of susceptible cells in the 66 generation is:  2212.0
The number of recovered cells in the 66 generation is:  2755.0
The number of perfect immune cells in the 66 generation is:  962.0
The number of infected cells in the 67 generation is:  3078.0
The number of susceptible cells in the 67 generation is:  2244.0
The number of recovered cells in the 67 generation is:  2755.0
The number of perfect immune cells in the 67 generation is:  982.0
The number of infected cells in the 68 generation is:  3042.0
The number of susceptible cells in the 68 generation is:  2232.0
The number of recovered cells in the 68 generation is:  2803.0
The number of perfect immune cells in the 68 generation is:  1003.0
The number of infected cells in the 69 generation is:  2985.0
The number of susceptible cells in the 69 generation is:  2264.0
The number of recovered cells in the 69 generation is:  2828.0
The number of perfect immune cells in the 69 generation is:  1022.0
The number of infected cells in the 70 generation is:  3001.0
The number of susceptible cells in the 70 generation is:  2273.0
```

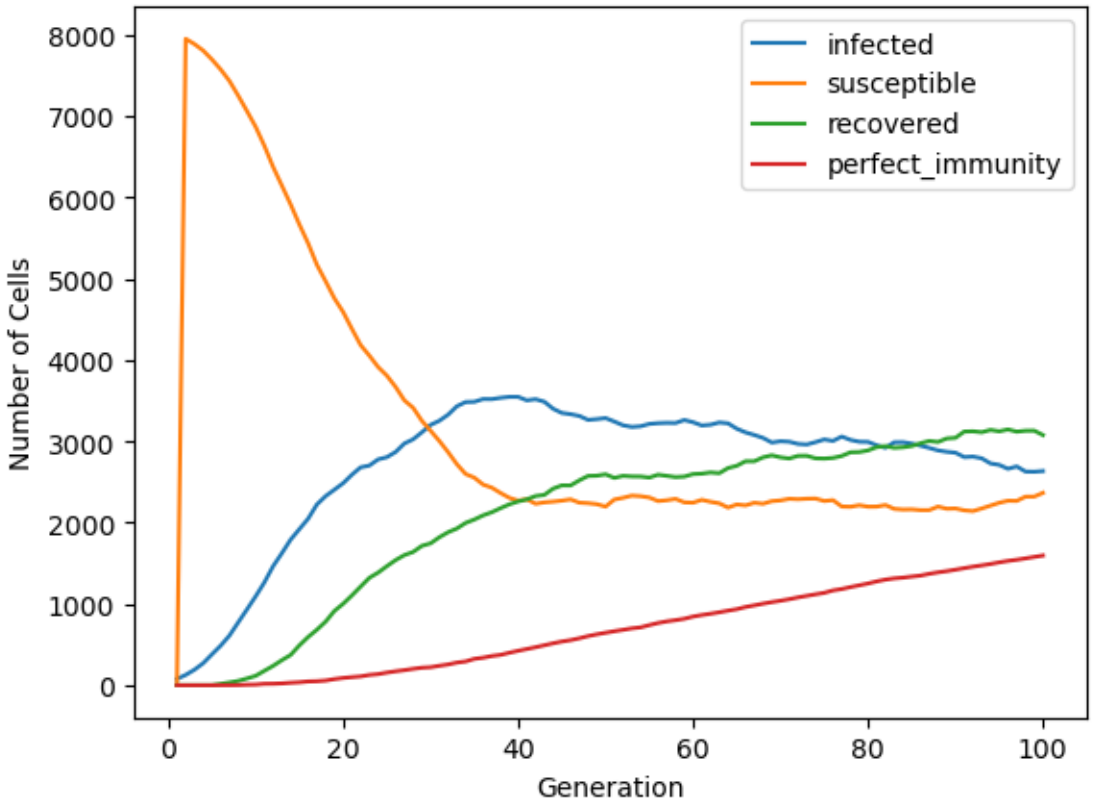
```
The number of recovered cells in the 70 generation is:  2803.0
The number of perfect immune cells in the 70 generation is:  1039.0
The number of infected cells in the 71 generation is:  2995.0
The number of susceptible cells in the 71 generation is:  2293.0
The number of recovered cells in the 71 generation is:  2789.0
The number of perfect immune cells in the 71 generation is:  1058.0
The number of infected cells in the 72 generation is:  2968.0
The number of susceptible cells in the 72 generation is:  2288.0
The number of recovered cells in the 72 generation is:  2821.0
The number of perfect immune cells in the 72 generation is:  1080.0
The number of infected cells in the 73 generation is:  2961.0
The number of susceptible cells in the 73 generation is:  2294.0
The number of recovered cells in the 73 generation is:  2822.0
The number of perfect immune cells in the 73 generation is:  1099.0
The number of infected cells in the 74 generation is:  2990.0
The number of susceptible cells in the 74 generation is:  2296.0
The number of recovered cells in the 74 generation is:  2791.0
The number of perfect immune cells in the 74 generation is:  1118.0
The number of infected cells in the 75 generation is:  3023.0
The number of susceptible cells in the 75 generation is:  2266.0
The number of recovered cells in the 75 generation is:  2788.0
The number of perfect immune cells in the 75 generation is:  1136.0
The number of infected cells in the 76 generation is:  3005.0
The number of susceptible cells in the 76 generation is:  2276.0
The number of recovered cells in the 76 generation is:  2796.0
The number of perfect immune cells in the 76 generation is:  1166.0
The number of infected cells in the 77 generation is:  3059.0
The number of susceptible cells in the 77 generation is:  2198.0
The number of recovered cells in the 77 generation is:  2820.0
The number of perfect immune cells in the 77 generation is:  1183.0
The number of infected cells in the 78 generation is:  3017.0
The number of susceptible cells in the 78 generation is:  2195.0
The number of recovered cells in the 78 generation is:  2865.0
The number of perfect immune cells in the 78 generation is:  1207.0
The number of infected cells in the 79 generation is:  2995.0
The number of susceptible cells in the 79 generation is:  2214.0
The number of recovered cells in the 79 generation is:  2868.0
The number of perfect immune cells in the 79 generation is:  1229.0
The number of infected cells in the 80 generation is:  2993.0
The number of susceptible cells in the 80 generation is:  2196.0
The number of recovered cells in the 80 generation is:  2888.0
The number of perfect immune cells in the 80 generation is:  1251.0
The number of infected cells in the 81 generation is:  2951.0
The number of susceptible cells in the 81 generation is:  2197.0
The number of recovered cells in the 81 generation is:  2929.0
The number of perfect immune cells in the 81 generation is:  1277.0
The number of infected cells in the 82 generation is:  2920.0
The number of susceptible cells in the 82 generation is:  2216.0
```

```
The number of recovered cells in the 82 generation is:  2941.0
The number of perfect immune cells in the 82 generation is:  1299.0
The number of infected cells in the 83 generation is:  2988.0
The number of susceptible cells in the 83 generation is:  2169.0
The number of recovered cells in the 83 generation is:  2920.0
The number of perfect immune cells in the 83 generation is:  1314.0
The number of infected cells in the 84 generation is:  2989.0
The number of susceptible cells in the 84 generation is:  2161.0
The number of recovered cells in the 84 generation is:  2927.0
The number of perfect immune cells in the 84 generation is:  1323.0
The number of infected cells in the 85 generation is:  2972.0
The number of susceptible cells in the 85 generation is:  2163.0
The number of recovered cells in the 85 generation is:  2942.0
The number of perfect immune cells in the 85 generation is:  1336.0
The number of infected cells in the 86 generation is:  2947.0
The number of susceptible cells in the 86 generation is:  2154.0
The number of recovered cells in the 86 generation is:  2976.0
The number of perfect immune cells in the 86 generation is:  1348.0
The number of infected cells in the 87 generation is:  2919.0
The number of susceptible cells in the 87 generation is:  2154.0
The number of recovered cells in the 87 generation is:  3004.0
The number of perfect immune cells in the 87 generation is:  1370.0
The number of infected cells in the 88 generation is:  2888.0
The number of susceptible cells in the 88 generation is:  2197.0
The number of recovered cells in the 88 generation is:  2992.0
The number of perfect immune cells in the 88 generation is:  1388.0
The number of infected cells in the 89 generation is:  2871.0
The number of susceptible cells in the 89 generation is:  2171.0
The number of recovered cells in the 89 generation is:  3035.0
The number of perfect immune cells in the 89 generation is:  1403.0
The number of infected cells in the 90 generation is:  2860.0
The number of susceptible cells in the 90 generation is:  2174.0
The number of recovered cells in the 90 generation is:  3043.0
The number of perfect immune cells in the 90 generation is:  1421.0
The number of infected cells in the 91 generation is:  2806.0
The number of susceptible cells in the 91 generation is:  2151.0
The number of recovered cells in the 91 generation is:  3120.0
The number of perfect immune cells in the 91 generation is:  1440.0
The number of infected cells in the 92 generation is:  2812.0
The number of susceptible cells in the 92 generation is:  2142.0
The number of recovered cells in the 92 generation is:  3123.0
The number of perfect immune cells in the 92 generation is:  1458.0
The number of infected cells in the 93 generation is:  2791.0
The number of susceptible cells in the 93 generation is:  2174.0
The number of recovered cells in the 93 generation is:  3112.0
The number of perfect immune cells in the 93 generation is:  1474.0
The number of infected cells in the 94 generation is:  2728.0
The number of susceptible cells in the 94 generation is:  2209.0
```

```
The number of recovered cells in the 94 generation is:  3140.0
The number of perfect immune cells in the 94 generation is:  1491.0
The number of infected cells in the 95 generation is:  2710.0
The number of susceptible cells in the 95 generation is:  2244.0
The number of recovered cells in the 95 generation is:  3123.0
The number of perfect immune cells in the 95 generation is:  1511.0
The number of infected cells in the 96 generation is:  2662.0
The number of susceptible cells in the 96 generation is:  2269.0
The number of recovered cells in the 96 generation is:  3146.0
The number of perfect immune cells in the 96 generation is:  1529.0
The number of infected cells in the 97 generation is:  2691.0
The number of susceptible cells in the 97 generation is:  2267.0
The number of recovered cells in the 97 generation is:  3119.0
The number of perfect immune cells in the 97 generation is:  1543.0
The number of infected cells in the 98 generation is:  2628.0
The number of susceptible cells in the 98 generation is:  2319.0
The number of recovered cells in the 98 generation is:  3130.0
The number of perfect immune cells in the 98 generation is:  1560.0
The number of infected cells in the 99 generation is:  2626.0
The number of susceptible cells in the 99 generation is:  2320.0
The number of recovered cells in the 99 generation is:  3131.0
The number of perfect immune cells in the 99 generation is:  1578.0
The number of infected cells in the 100 generation is:  2634.0
The number of susceptible cells in the 100 generation is:  2366.0
The number of recovered cells in the 100 generation is:  3077.0
The number of perfect immune cells in the 100 generation is:  1595.0
```

[320]:
```
plt.plot([x+1 for x in list(range(generations))], infected_generation, label =␣
 ↪"infected")
plt.plot([x+1 for x in list(range(generations))], susceptible_generation, label␣
 ↪= "susceptible")
plt.plot([x+1 for x in list(range(generations))], recovered_generation, label =␣
 ↪"recovered")
plt.plot([x+1 for x in list(range(generations))], perfect_immunity_generation,␣
 ↪label = "perfect_immunity")
plt.xlabel("Generation")
plt.ylabel("Number of Cells")
plt.legend()
```

[320]:
```
<matplotlib.legend.Legend at 0x2020bfb7fd0>
```



2 Determining the percentage of infected cells to the total population for probability of transmission of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,0.9

[308]:
```
infected_array = np.empty((len(pt), generations))
susceptible_array = np.empty((len(pt), generations))
recovered_array = np.empty((len(pt), generations))
percentage_infected = np.zeros(len(pt))
```

[309]:
```
for i,prob_infection in enumerate(pt):
    for row in range(row_number):
        for col in range(col_number):
            #make a class for the cell in lattice[row,,coll]
            lattice[row,col] = Cell(state = 0, duration = np.random.
 ↪randint(2,4), immunity = np.random.randint(1,10))
            #make a random choice with probability of 1 being 0.8 and␣
 ↪probability of 0 being 0.2
            lattice[row,col].state = np.random.choice([0,1], p = [1-p0,p0])
            if lattice[row,col].state == 1:
```

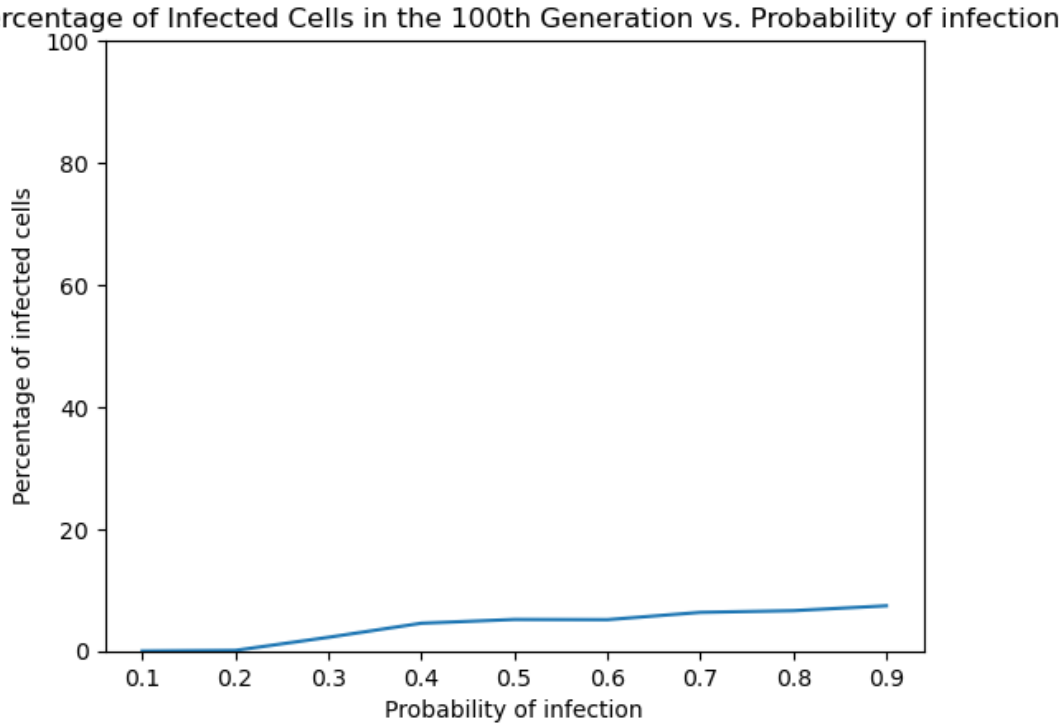
```
                lattice[row,col].state = np.random.choice([1,2], p = [0.99,0.
↪01])
                if lattice[row,col].state == 2:
                    lattice[row,col].infected_duration = lattice[row,col].
↪duration
    #This is the first generation.
    # Count the number of infected cells (state == 2) and populated cells␣
↪(state != 0)
    infected_array[i,0] = sum(cell.state == 2 for row in lattice for cell in␣
↪row)
    # count the number of non_zero cells (state!= 0)
    populated_generation = sum(cell.state != 0 for row in lattice for cell in␣
↪row)
    for generation in [x+2 for x in list(range(generations-1))]:
        for row in range(row_number):
            for col in range(col_number):
                if lattice[row,col].recently_infected == True:
                    continue #if the agent is recently infected, it cant affect␣
↪any agent.
                if lattice[row,col].state == 2:
                    #upper neighbour
                    if row > 0: #to avoid index error
                        if lattice[row-1,col].state == 1: #upper neighbour
                            if lattice[row-1,col].check_immunity() == False:
                                lattice[row-1,col].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row-1,col].state == 2:
                                    lattice[row-1,col].just_infected()

                    #lower neighbour
                    if row < row_number-1: #to avoid index error
                        if lattice[row+1,col].state == 1: #lower neighbour
                            if lattice[row+1,col].check_immunity() == False:
                                lattice[row+1,col].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row+1,col].state == 2:
                                    lattice[row+1,col].just_infected()

                    #left neighbour
                    if col > 0: #to avoid index error
                        if lattice[row,col-1].state == 1: #left neighbour
                            if lattice[row,col-1].check_immunity() == False:
                                lattice[row,col-1].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row,col-1].state == 2:
                                    lattice[row,col-1].just_infected()
```

```
                    #right neighbour
                    if col < col_number-1: #to avoid index error
                        if lattice[row,col+1].state == 1: #right neighbour
                            if lattice[row,col+1].check_immunity() == False:
                                lattice[row,col+1].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row,col+1].state == 2:
                                    lattice[row,col+1].just_infected()
                    #upper left neighbour
                    if row > 0 and col > 0: #to avoid index error
                        if lattice[row-1,col-1].state == 1: #upper left␣
↪neighbour
                            if lattice[row-1,col-1].check_immunity() == False:
                                lattice[row-1,col-1].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row-1,col-1].state == 2:
                                    lattice[row-1,col-1].just_infected()

                    #upper right neighbour
                    if row > 0 and col < col_number-1: #to avoid index error
                        if lattice[row-1,col+1].state == 1: #upper right␣
↪neighbour
                            if lattice[row-1,col+1].check_immunity() == False:
                                lattice[row-1,col+1].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row-1,col+1].state == 2:
                                    lattice[row-1,col+1].just_infected()

                    #lower right neighbour
                    if row < row_number-1 and col < col_number-1: #to avoid␣
↪index error
                        if lattice[row+1,col+1].state == 1: #lower right␣
↪neighbour
                            if lattice[row+1,col+1].check_immunity() == False:
                                lattice[row+1,col+1].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
                                if lattice[row+1,col+1].state == 2:
                                    lattice[row+1,col+1].just_infected()
                    #lower left neighbour
                    if row < row_number-1 and col > 0: #to avoid index error
                        if lattice[row+1,col-1].state == 1: #lower left␣
↪neighbour
                            if lattice[row+1,col-1].check_immunity() == False:
                                lattice[row+1,col-1].state =  np.random.
↪choice([1,2], p = [1-prob_infection,prob_infection])
```

```
                                if lattice[row+1,col-1].state == 2:
                                    lattice[row+1,col-1].just_infected()
        #After the iteration, iterate again for checking the state and removing␣
↪duration
        for row in range(row_number):
            for col in range(col_number):
                lattice[row,col].check_state()
        # Count the number of infected cells (state == 2) and populated cells␣
↪(state != 0)
        infected_array[i,generation-1] = sum(cell.state == 2 for row in lattice␣
↪for cell in row)
    percentage_infected[i] = infected_array[i, generations-1]/sum(cell.state !=␣
↪0 for row in lattice for cell in row) * 100
```

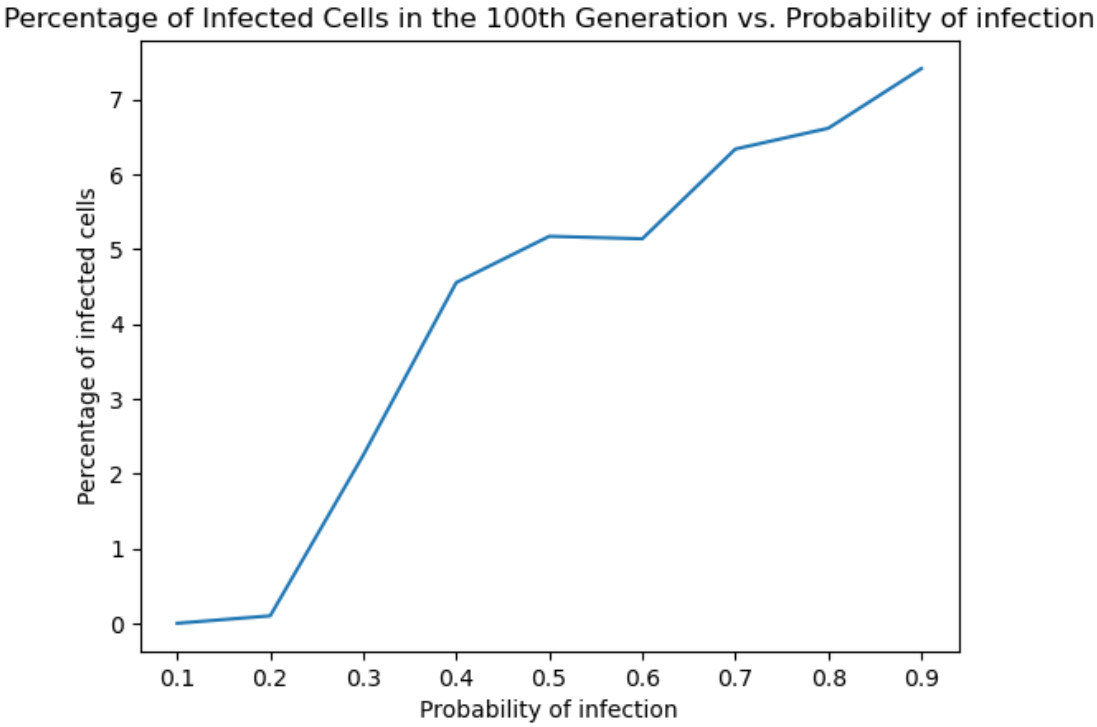
[310]:
```
sns.lineplot(x = pt, y = percentage_infected)
plt.xlabel("Probability of infection")
plt.ylabel("Percentage of infected cells")
plt.ylim(0,100)
plt.title("Percentage of Infected Cells in the 100th Generation vs. Probability␣
 ↪of infection")
```

[310]:
```
Text(0.5, 1.0, 'Percentage of Infected Cells in the 100th Generation vs.
Probability of infection')
```



[311]:
```
sns.lineplot(x = pt, y = percentage_infected)
plt.xlabel("Probability of infection")
plt.ylabel("Percentage of infected cells")
plt.title("Percentage of Infected Cells in the 100th Generation vs. Probability␣
↪of infection")
```

[311]:
```
Text(0.5, 1.0, 'Percentage of Infected Cells in the 100th Generation vs.
Probability of infection')
```



2.1 Visualizing the number of infected cells for different probability of transmission

[312]:
```
infected_array
```

[312]:
```
array([[6.600e+01, 1.170e+02, 1.110e+02, 5.900e+01, 4.000e+01, 2.500e+01,
        1.200e+01, 1.100e+01, 1.000e+01, 9.000e+00, 5.000e+00, 4.000e+00,
        4.000e+00, 2.000e+00, 2.000e+00, 2.000e+00, 1.000e+00, 1.000e+00,
        1.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
```

```
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00],
[7.200e+01, 1.790e+02, 2.400e+02, 2.610e+02, 2.620e+02, 2.690e+02,
 2.450e+02, 2.480e+02, 2.490e+02, 2.360e+02, 2.280e+02, 2.200e+02,
 2.210e+02, 2.120e+02, 1.890e+02, 1.920e+02, 2.130e+02, 2.010e+02,
 2.170e+02, 2.090e+02, 2.130e+02, 2.210e+02, 2.140e+02, 2.100e+02,
 2.100e+02, 1.830e+02, 1.810e+02, 1.650e+02, 1.600e+02, 1.580e+02,
 1.310e+02, 1.350e+02, 1.340e+02, 1.320e+02, 1.310e+02, 1.240e+02,
 1.180e+02, 1.280e+02, 1.130e+02, 1.150e+02, 1.020e+02, 1.090e+02,
 1.050e+02, 1.080e+02, 9.200e+01, 8.100e+01, 7.600e+01, 7.200e+01,
 7.100e+01, 5.600e+01, 5.100e+01, 5.900e+01, 5.900e+01, 5.100e+01,
 5.000e+01, 5.300e+01, 4.900e+01, 4.700e+01, 3.900e+01, 3.700e+01,
 4.100e+01, 4.000e+01, 4.700e+01, 6.400e+01, 6.200e+01, 5.300e+01,
 5.300e+01, 5.200e+01, 4.400e+01, 3.400e+01, 2.900e+01, 3.200e+01,
 3.500e+01, 3.300e+01, 3.500e+01, 3.400e+01, 3.700e+01, 3.300e+01,
 3.400e+01, 3.700e+01, 3.900e+01, 3.500e+01, 2.800e+01, 2.600e+01,
 2.700e+01, 2.700e+01, 2.200e+01, 3.100e+01, 2.700e+01, 1.600e+01,
 1.400e+01, 1.500e+01, 2.000e+01, 1.500e+01, 8.000e+00, 9.000e+00,
 1.100e+01, 1.100e+01, 1.000e+01, 8.000e+00],
[7.100e+01, 2.140e+02, 3.550e+02, 4.300e+02, 5.030e+02, 5.390e+02,
 5.940e+02, 6.310e+02, 6.680e+02, 6.750e+02, 6.250e+02, 6.060e+02,
 5.990e+02, 6.330e+02, 6.810e+02, 6.990e+02, 6.730e+02, 6.740e+02,
 6.780e+02, 6.410e+02, 6.330e+02, 6.360e+02, 6.350e+02, 6.470e+02,
 6.930e+02, 6.640e+02, 7.000e+02, 7.420e+02, 7.180e+02, 6.730e+02,
 6.320e+02, 6.170e+02, 5.810e+02, 6.010e+02, 6.130e+02, 6.050e+02,
 5.950e+02, 5.940e+02, 6.070e+02, 5.620e+02, 5.550e+02, 5.470e+02,
 5.440e+02, 5.400e+02, 5.380e+02, 5.260e+02, 4.480e+02, 4.540e+02,
 4.140e+02, 3.730e+02, 3.670e+02, 3.540e+02, 3.510e+02, 3.530e+02,
 3.930e+02, 4.010e+02, 4.030e+02, 4.080e+02, 4.000e+02, 3.880e+02,
 3.620e+02, 3.640e+02, 3.180e+02, 3.020e+02, 2.940e+02, 2.940e+02,
 2.990e+02, 3.130e+02, 3.080e+02, 3.170e+02, 3.440e+02, 3.190e+02,
 2.990e+02, 2.920e+02, 2.770e+02, 2.680e+02, 2.790e+02, 2.730e+02,
 2.700e+02, 2.660e+02, 2.710e+02, 2.760e+02, 2.530e+02, 2.420e+02,
 2.360e+02, 2.340e+02, 2.210e+02, 2.300e+02, 2.330e+02, 2.420e+02,
 2.120e+02, 2.020e+02, 2.140e+02, 1.800e+02, 1.720e+02, 1.690e+02,
 1.880e+02, 1.940e+02, 1.830e+02, 1.790e+02],
[8.000e+01, 2.840e+02, 5.270e+02, 7.190e+02, 9.040e+02, 1.017e+03,
 1.073e+03, 1.055e+03, 1.058e+03, 9.790e+02, 9.320e+02, 9.110e+02,
 8.290e+02, 7.280e+02, 6.080e+02, 5.630e+02, 6.090e+02, 6.220e+02,
 6.720e+02, 7.430e+02, 7.730e+02, 8.020e+02, 8.620e+02, 9.050e+02,
 9.150e+02, 9.290e+02, 8.750e+02, 8.530e+02, 8.360e+02, 7.800e+02,
```

```
 7.810e+02, 7.810e+02, 7.440e+02, 7.490e+02, 7.580e+02, 7.610e+02,
 7.540e+02, 7.690e+02, 7.430e+02, 7.480e+02, 7.760e+02, 8.050e+02,
 7.730e+02, 7.570e+02, 7.240e+02, 7.000e+02, 6.490e+02, 6.670e+02,
 6.390e+02, 6.350e+02, 6.510e+02, 6.540e+02, 6.510e+02, 6.310e+02,
 6.320e+02, 6.220e+02, 5.960e+02, 5.980e+02, 5.600e+02, 5.660e+02,
 5.430e+02, 5.240e+02, 5.100e+02, 5.460e+02, 5.550e+02, 5.720e+02,
 5.920e+02, 5.750e+02, 5.610e+02, 5.140e+02, 5.010e+02, 5.010e+02,
 4.990e+02, 4.870e+02, 4.440e+02, 4.340e+02, 4.440e+02, 4.350e+02,
 4.730e+02, 4.580e+02, 4.270e+02, 4.370e+02, 4.350e+02, 4.260e+02,
 4.150e+02, 3.910e+02, 3.840e+02, 3.430e+02, 3.350e+02, 3.210e+02,
 2.870e+02, 2.990e+02, 3.240e+02, 3.280e+02, 3.190e+02, 3.060e+02,
 3.020e+02, 3.300e+02, 3.570e+02, 3.620e+02],
[9.500e+01, 3.750e+02, 7.310e+02, 1.033e+03, 1.285e+03, 1.406e+03,
 1.446e+03, 1.424e+03, 1.251e+03, 1.062e+03, 9.060e+02, 7.420e+02,
 6.790e+02, 6.950e+02, 7.150e+02, 8.680e+02, 9.440e+02, 1.031e+03,
 1.186e+03, 1.288e+03, 1.278e+03, 1.263e+03, 1.153e+03, 1.024e+03,
 9.300e+02, 8.980e+02, 7.970e+02, 7.660e+02, 7.890e+02, 8.890e+02,
 9.400e+02, 9.770e+02, 1.020e+03, 1.075e+03, 1.093e+03, 1.036e+03,
 9.540e+02, 9.270e+02, 8.640e+02, 7.650e+02, 7.830e+02, 7.830e+02,
 7.620e+02, 7.730e+02, 8.030e+02, 8.260e+02, 8.540e+02, 8.850e+02,
 8.700e+02, 8.390e+02, 7.740e+02, 7.680e+02, 7.790e+02, 7.650e+02,
 7.310e+02, 7.280e+02, 6.930e+02, 6.240e+02, 6.470e+02, 6.710e+02,
 6.470e+02, 6.670e+02, 6.430e+02, 6.440e+02, 6.340e+02, 6.430e+02,
 6.240e+02, 6.280e+02, 6.220e+02, 6.140e+02, 6.490e+02, 6.500e+02,
 6.450e+02, 6.530e+02, 5.860e+02, 6.110e+02, 5.760e+02, 5.550e+02,
 5.340e+02, 5.460e+02, 5.230e+02, 5.070e+02, 5.200e+02, 4.850e+02,
 4.780e+02, 4.750e+02, 5.120e+02, 4.980e+02, 4.910e+02, 4.370e+02,
 4.220e+02, 4.250e+02, 4.370e+02, 4.210e+02, 4.250e+02, 4.520e+02,
 4.650e+02, 4.160e+02, 3.880e+02, 4.130e+02],
[9.000e+01, 4.310e+02, 8.720e+02, 1.137e+03, 1.339e+03, 1.463e+03,
 1.494e+03, 1.399e+03, 1.213e+03, 1.091e+03, 9.080e+02, 7.140e+02,
 6.390e+02, 6.260e+02, 7.550e+02, 9.690e+02, 1.179e+03, 1.310e+03,
 1.315e+03, 1.311e+03, 1.323e+03, 1.210e+03, 1.161e+03, 1.052e+03,
 9.490e+02, 8.230e+02, 7.760e+02, 8.330e+02, 9.270e+02, 1.030e+03,
 1.151e+03, 1.232e+03, 1.175e+03, 1.182e+03, 1.135e+03, 1.057e+03,
 9.080e+02, 8.070e+02, 7.390e+02, 7.470e+02, 8.200e+02, 8.670e+02,
 8.770e+02, 8.940e+02, 9.480e+02, 1.017e+03, 1.059e+03, 1.103e+03,
 1.045e+03, 9.770e+02, 8.830e+02, 7.750e+02, 6.970e+02, 7.150e+02,
 7.020e+02, 7.540e+02, 7.630e+02, 7.910e+02, 8.590e+02, 9.070e+02,
 9.340e+02, 8.460e+02, 8.370e+02, 7.760e+02, 6.840e+02, 6.740e+02,
 6.810e+02, 6.500e+02, 6.610e+02, 6.630e+02, 7.020e+02, 7.190e+02,
 7.370e+02, 7.620e+02, 7.360e+02, 7.310e+02, 6.990e+02, 6.870e+02,
 6.840e+02, 6.400e+02, 6.040e+02, 5.650e+02, 5.850e+02, 5.610e+02,
 5.760e+02, 5.780e+02, 5.710e+02, 5.580e+02, 5.490e+02, 5.230e+02,
 5.410e+02, 5.480e+02, 5.050e+02, 5.100e+02, 4.840e+02, 4.320e+02,
 4.180e+02, 4.040e+02, 4.200e+02, 4.130e+02],
[1.030e+02, 5.230e+02, 1.081e+03, 1.461e+03, 1.755e+03, 1.715e+03,
```

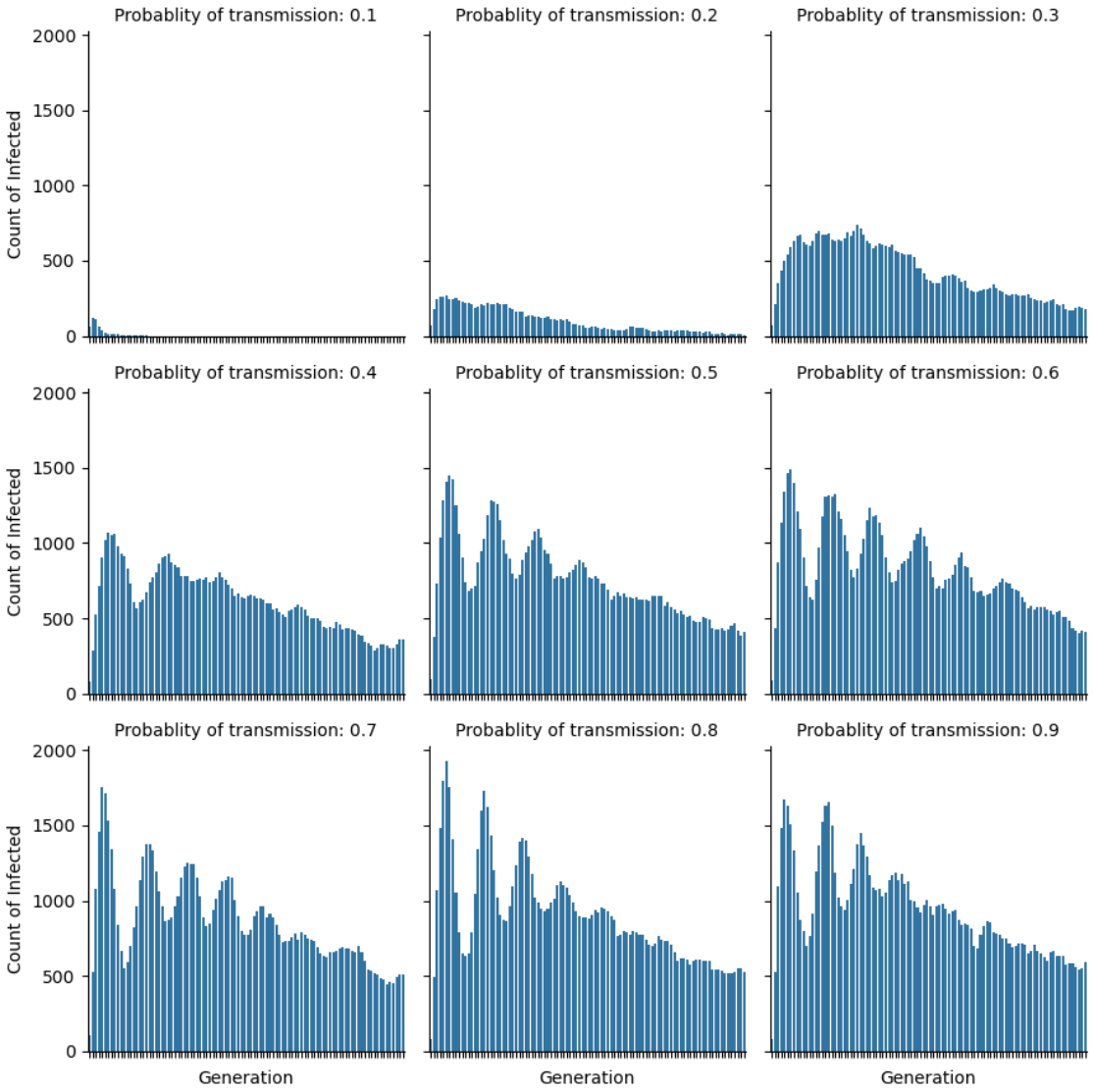
```
 1.535e+03, 1.341e+03, 1.078e+03, 8.380e+02, 6.630e+02, 5.490e+02,
 5.930e+02, 6.960e+02, 8.240e+02, 9.650e+02, 1.137e+03, 1.294e+03,
 1.378e+03, 1.376e+03, 1.332e+03, 1.190e+03, 1.061e+03, 9.630e+02,
 8.620e+02, 8.740e+02, 8.920e+02, 9.630e+02, 1.032e+03, 1.150e+03,
 1.227e+03, 1.253e+03, 1.243e+03, 1.244e+03, 1.156e+03, 1.028e+03,
 8.920e+02, 8.290e+02, 8.470e+02, 9.410e+02, 1.015e+03, 1.070e+03,
 1.124e+03, 1.138e+03, 1.159e+03, 1.153e+03, 1.005e+03, 8.950e+02,
 7.950e+02, 7.720e+02, 7.720e+02, 8.090e+02, 9.000e+02, 9.320e+02,
 9.660e+02, 9.670e+02, 8.900e+02, 9.100e+02, 8.870e+02, 8.380e+02,
 7.730e+02, 7.250e+02, 7.310e+02, 7.300e+02, 7.580e+02, 7.810e+02,
 7.420e+02, 7.900e+02, 7.760e+02, 7.490e+02, 7.440e+02, 7.280e+02,
 6.880e+02, 6.520e+02, 6.360e+02, 6.270e+02, 6.610e+02, 6.540e+02,
 6.670e+02, 6.830e+02, 6.910e+02, 6.820e+02, 6.790e+02, 6.650e+02,
 6.600e+02, 6.960e+02, 6.540e+02, 5.960e+02, 5.460e+02, 5.300e+02,
 5.160e+02, 5.060e+02, 4.870e+02, 4.740e+02, 4.460e+02, 4.610e+02,
 4.500e+02, 4.940e+02, 5.090e+02, 5.090e+02],
[8.300e+01, 4.950e+02, 1.070e+03, 1.484e+03, 1.798e+03, 1.927e+03,
 1.757e+03, 1.407e+03, 1.054e+03, 7.930e+02, 6.470e+02, 6.360e+02,
 6.510e+02, 7.930e+02, 1.044e+03, 1.339e+03, 1.602e+03, 1.733e+03,
 1.620e+03, 1.437e+03, 1.201e+03, 1.019e+03, 9.030e+02, 8.750e+02,
 8.640e+02, 9.600e+02, 1.091e+03, 1.239e+03, 1.389e+03, 1.420e+03,
 1.399e+03, 1.293e+03, 1.180e+03, 1.024e+03, 9.900e+02, 9.490e+02,
 9.340e+02, 9.490e+02, 9.850e+02, 1.014e+03, 1.107e+03, 1.131e+03,
 1.106e+03, 1.089e+03, 1.037e+03, 9.870e+02, 9.340e+02, 8.990e+02,
 8.920e+02, 8.920e+02, 8.830e+02, 9.060e+02, 9.360e+02, 9.240e+02,
 9.530e+02, 9.480e+02, 9.340e+02, 8.960e+02, 8.720e+02, 7.690e+02,
 7.760e+02, 7.970e+02, 7.870e+02, 7.740e+02, 7.950e+02, 7.890e+02,
 7.770e+02, 7.740e+02, 7.440e+02, 7.110e+02, 7.030e+02, 7.170e+02,
 7.640e+02, 7.440e+02, 7.290e+02, 7.310e+02, 7.070e+02, 6.600e+02,
 6.020e+02, 6.160e+02, 6.150e+02, 6.100e+02, 5.790e+02, 5.960e+02,
 6.060e+02, 6.110e+02, 6.020e+02, 6.040e+02, 6.030e+02, 5.400e+02,
 5.390e+02, 5.410e+02, 5.320e+02, 5.150e+02, 5.160e+02, 5.170e+02,
 5.250e+02, 5.520e+02, 5.470e+02, 5.270e+02],
[8.200e+01, 5.250e+02, 1.097e+03, 1.480e+03, 1.671e+03, 1.632e+03,
 1.508e+03, 1.338e+03, 1.050e+03, 8.730e+02, 7.970e+02, 7.020e+02,
 7.670e+02, 9.170e+02, 1.190e+03, 1.370e+03, 1.520e+03, 1.631e+03,
 1.660e+03, 1.503e+03, 1.188e+03, 1.020e+03, 9.630e+02, 9.380e+02,
 1.008e+03, 1.115e+03, 1.213e+03, 1.379e+03, 1.448e+03, 1.366e+03,
 1.297e+03, 1.170e+03, 1.085e+03, 1.074e+03, 1.075e+03, 1.029e+03,
 1.056e+03, 1.136e+03, 1.167e+03, 1.185e+03, 1.136e+03, 1.175e+03,
 1.114e+03, 1.124e+03, 1.002e+03, 9.990e+02, 9.510e+02, 9.220e+02,
 9.720e+02, 1.003e+03, 9.650e+02, 9.060e+02, 9.670e+02, 9.700e+02,
 9.770e+02, 9.470e+02, 9.150e+02, 9.280e+02, 9.400e+02, 8.710e+02,
 8.370e+02, 8.470e+02, 8.430e+02, 8.170e+02, 7.020e+02, 6.860e+02,
 7.740e+02, 8.350e+02, 8.610e+02, 8.550e+02, 7.930e+02, 7.830e+02,
 7.750e+02, 7.510e+02, 7.510e+02, 7.190e+02, 6.920e+02, 7.010e+02,
 7.140e+02, 7.120e+02, 7.060e+02, 6.480e+02, 6.690e+02, 7.040e+02,
```

```
       6.690e+02, 6.530e+02, 6.210e+02, 6.040e+02, 6.600e+02, 6.660e+02,
       6.300e+02, 6.360e+02, 6.290e+02, 5.740e+02, 5.870e+02, 5.830e+02,
       5.560e+02, 5.450e+02, 5.520e+02, 5.900e+02]])
```

[313]:
```
df = pd.DataFrame(infected_array).T
df.index = [generation for generation in range(1,generations+1)]
df.index.name = 'generation'
df.columns = pt
# Melt the DataFrame to long format
df_melted = df.reset_index().melt(id_vars='generation', var_name='probability',␣
 ↪value_name='count')
```

[314]:
```
g = sns.FacetGrid(df_melted, col='probability', col_wrap=3, sharey=True)
g.map(sns.barplot, 'generation', 'count')

# Adjust layout
g.set_titles(col_template='Probablity of transmission: {col_name}')
g.set_axis_labels('Generation', 'Count of Infected')
for ax in g.axes.flat:
    ax.set_xticklabels([])

plt.tight_layout()
plt.show()
```

```
c:\Users\jacob\anaconda3\lib\site-packages\seaborn\axisgrid.py:670: UserWarning:
Using the barplot function without specifying `order` is likely to produce an
incorrect plot.
  warnings.warn(warning)
```


Probablity of transmission: 0.1
Probablity of transmission: 0.2
Probablity of transmission: 0.3
Probablity of transmission: 0.4
Probablity of transmission: 0.5
Probablity of transmission: 0.6
Probablity of transmission: 0.7
Probablity of transmission: 0.8
Probablity of transmission: 0.9
Count of Infected
0
500
1000
1500
2000
Generation