

# Modeling Statistical phenomena

## Assignment 2

Ali Setareh Kokab (95100491)

April 8, 2021

# Contents

<b>1</b>	<b>Summary</b>	<b>3</b>
<b>2</b>	<b>SIR on lattice</b>	<b>3</b>
2.1	Build a lattice . . . . .	3
2.2	SIR on lattice . . . . .	3
2.3	Results . . . . .	4

## 1 Summary

In this assignment we are going to study network-based SIR model in which infection is transmitted through a network of contacts between individuals. We also try to find the numerical values of the parameters in SIR model's solution (page 23) [1] for weekly deaths from plague in Bombay data. Finally, we review some articles from recent literature about models that has been used for modeling the COVID-19 pandemic.

## 2 SIR on lattice

In this exercise, we make a square lattice and run SIR model on it.

### 2.1 Build a lattice

The network that we consider here is a two dimensional square lattice in which each node is connected with its four nearest neighbors. We also use periodic boundary condition. To make our lattice, we used `lattice.grid_graph` function from Python NetworkX library [2] with periodic parameter set true.

### 2.2 SIR on lattice

To simulate SIR in lattice we have written two main functions. The first function is `init` function which it is responsible to produce the initial condition for us. It gets total number nodes (N) and the number infected nodes (k). In this function for each node we define two attributes which represent the node today condition and tomorrow condition respectively. By condition we mean susceptible (S), Infected (I) and recovered (R). In this function we choose k nodes randomly and set both of their attributes 'i'.

The function `sir` is responsible for time evolution of the lattice. It gets number of time steps of the simulation and probability to get infected (p) and recovered (r). In each time step, we loop through infected nodes and produce four random numbers for each of these nodes. If the random number is less than p, we change the 'tomorrow' attribute of that node from 'S' to 'I'. Moreover, we loop through every nodes that their 'today' attribute is infected and produce a random number for each of them. If this random number is less than r, we change 'tomorrow' attribute of that node from 'I' to 'R'. Eventually, we set 'today' attribute of changed nodes to their 'tomorrow' attribute and go the next time step. This function runs until there are no infected node in the system. This function returns three lists which contains number of the susceptibles, infecteds and recovers in each time step.

As a result of using random numbers in our model, every time that we run our code we will see a different result. In order to be able to draw conclusion from our simulation, we should run our code several times and report the **average** of our outputs.

### 2.3 Results

In fig 1 you see the histogram of the  $R(\infty)$  for  $p=0.25$  and  $r=1$ . For plotting this graph, we have ran our code two thousands time. As you can see, the final number of recovered nodes in most executions is less than ten. This result is expected as the probability of gettign infected is much less than the probaility of getting recovered. By looking at the plot of  $I$  versus time (figure 2), we see that in this scenario we have endemic.

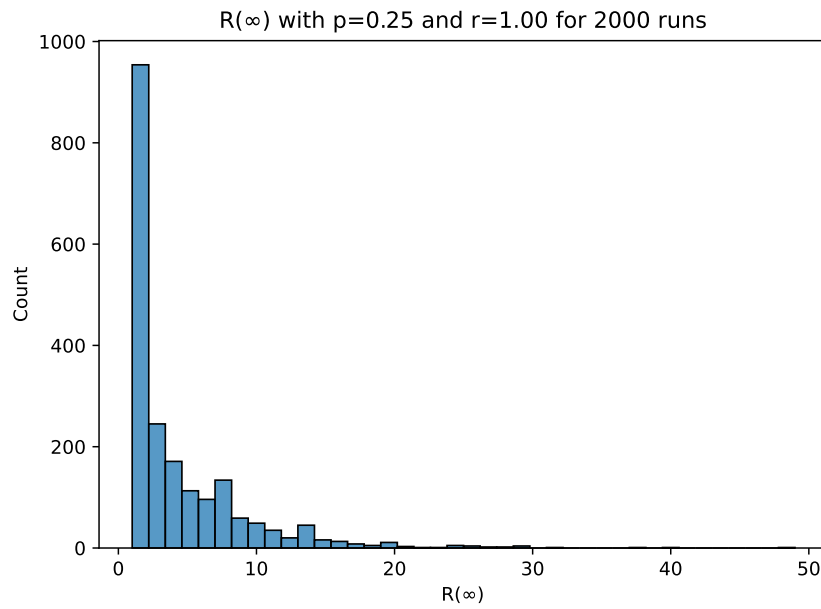


Figure 1: Histogram of  $R(\infty)$

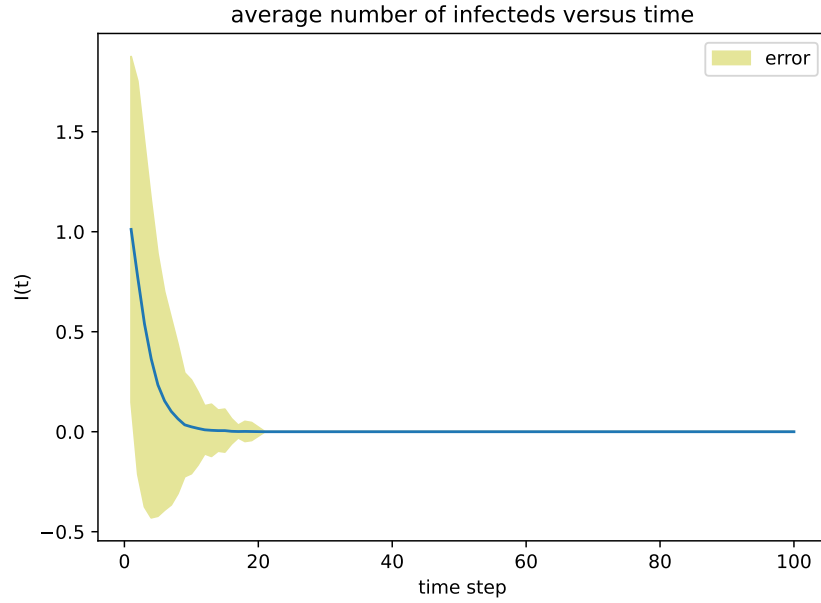


Figure 2: Average number of the infecteds versus time for  $p=0.25$  and  $r=1$ . Each point is averaged over 2000 ensembles. As it is seen, we are facing endemic here.

We run our program for different values of  $p$ . You see the results in figure 3 to 6.

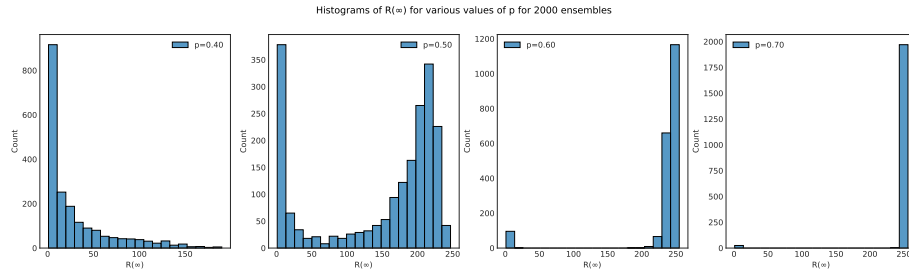


Figure 3: Caption

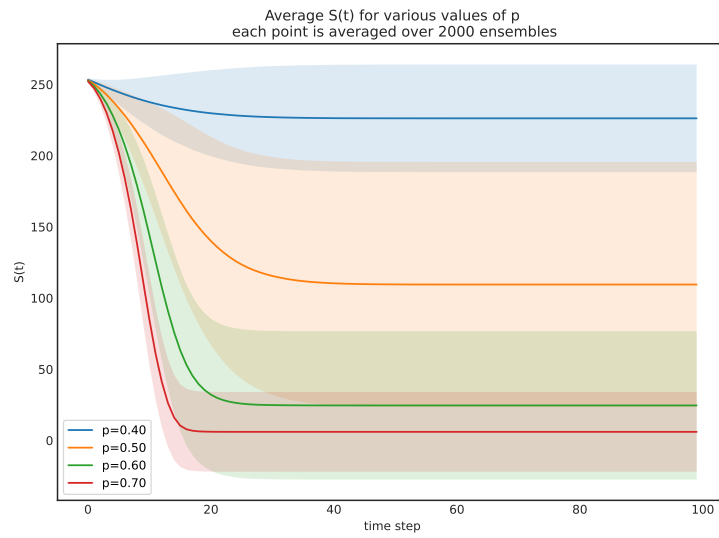


Figure 4: Caption

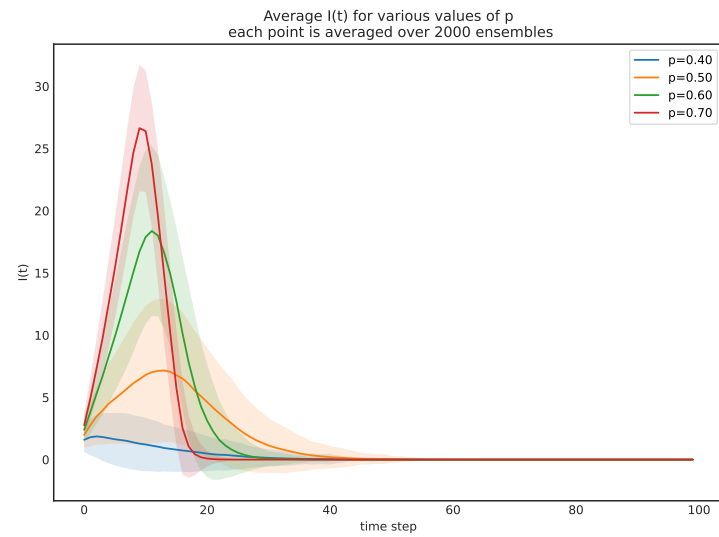


Figure 5: Caption

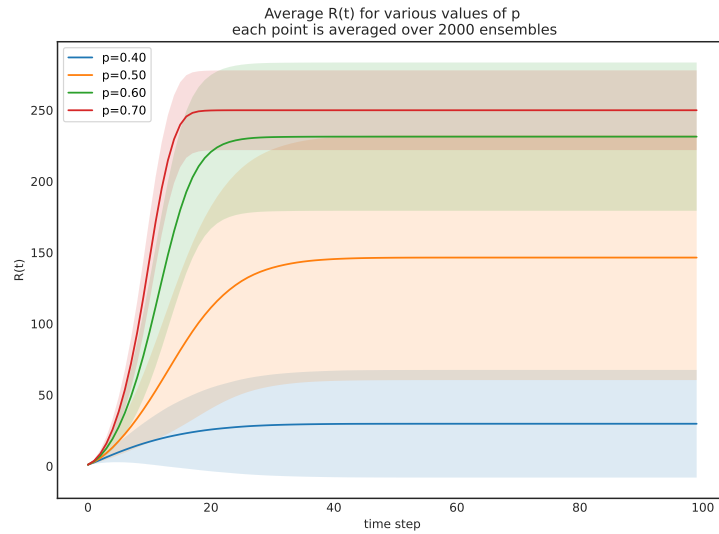


Figure 6: Caption

## References

1. Keeling, M. J. & Rohani, P. *Modeling Infectious Diseases in Humans and Animals* <https://doi.org/10.2307/j.ctvcn4gk0> (Princeton University Press, Sept. 2011).
2. Hagberg, A. A., Schult, D. A. & Swart, P. J. *Exploring Network Structure, Dynamics, and Function using NetworkX* in *Proceedings of the 7th Python in Science Conference* (eds Varoquaux, G., Vaught, T. & Millman, J.) (Pasadena, CA USA, 2008), 11–15.