

Research report for the Final Year Project

## “ Epidemic Simulation ”

Under the guidance of

**Dr. H C Vijayalakshmi**

Team Number: 26

Team Members:

USN	Name	Section	Roll No.
01JST17CS052	Eeshaan Achar	C	16
01JST17CS024	Anuj Yadav	C	8
01JST17CS142	Saurav Kumar	C	40
01JST17CS080	Manjunath Badakar	C	22

Signature of Guide  
(Dr. H C Vijayalakshmi)

Signature of HOD  
(Dr. M P Pushpalatha)

**Department of Computer Science and Engineering**  
**2020-21**

## CERTIFICATE

This is to certify that the final year project presentation titled **“Epidemic Simulation”** is presented by **Eeshaan Achar, Anuj Yadav, Saurav Kumar** and **Manhunath Badakar** for the partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science and Engineering, at JSS Science & Technology University, Mysuru during the year 2020-21. It is certified that all corrections / suggestions suggested during the presentation have been incorporated in the report. The report has been approved as it satisfies the academic requirements with respect to the final year project presentation prescribed for the Bachelor of Engineering degree.

**Signature of Guide**

**(Dr. H C Vijayalakshmi)**

**Signature of HOD**

**(Dr. M. P. Pushpalatha)**

**Panel Members**

1. ....

2. ....

**Signature**

.....

.....

**Date:**

**Place:**

## **Declaration**

We, the undersigned, solemnly declare that the report of the research work entitled “Epidemic Simulation”, is based on our own work carried out during the course of our study at JSS Science and Technology University Mysore, under the supervision of Dr. H C Vijayalakshmi. We assert that the statements made and conclusions drawn are an outcome of the research work. We further declare that to the best of our knowledge and belief that the project report.

Eeshaan Achar

Anuj Yadav

Saurav Kumar

Manjunath Badakar

## **Abstract**

This document is the concluding report of our research project entitled “Epidemic Simulation”, and is prepared as per the guidelines issued by the Department of Computer Science and Engineering, JSS Science and Technology University Mysore. In this document, we have covered every important aspect of the research to the best possible extent, and we hope that the anyone reading this would not only understand the research but would also find it worthy of his / her time.

## **Acknowledgement**

This research has been made possible by the kind support of many individuals such as our friends and families, and organizations such as our university. We wish to extend our sincere gratitude to all of them. We would also like to express gratitude to our mentor Dr. H C Vijayalakshmi, as without her guidance in the past as well as in the coming days, this project would not have been feasible. Lastly, we would like to thank our fellow teammates for their time and efforts, without which this project would not have seen the light of the day.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Statement . . . . .	1
1.2	Introduction to Problem Domain . . . . .	1
1.3	Objectives and Scope . . . . .	1
1.3.1	Objectives . . . . .	1
1.3.2	Scope . . . . .	2
1.4	Applications . . . . .	2
1.5	Existing Solutions . . . . .	2
1.6	Proposed Solution . . . . .	3
1.7	Research Timeline . . . . .	3
<b>2</b>	<b>Literature Survey</b>	<b>4</b>
2.1	SARS: The First Pandemic of the 21st Century . . . . .	4
2.2	Strategies for containing an emerging influenza pandemic in Southeast Asia	4
2.3	Mathematical models for devising the optimal Ebola virus disease eradication	5
2.4	Homogeneous and network models in Epidemiology . . . . .	5
2.5	FluTE, a publicly available stochastic influenza Epidemic simulation model	5
2.6	Agent-based simulation on avian influenza in Vietnam . . . . .	6
2.7	Cholera Epidemic in Haiti, 2010 . . . . .	6
2.8	Global seasonal occurrence of MERS-CoV infection . . . . .	6
2.9	The effects of border control and quarantine measures on the spread of COVID-19 . . . . .	7
2.10	A Social Network Model of the COVID-19 Pandemic . . . . .	7
2.11	Simulation of the COVID-19 epidemic on the social network of Slovenia . .	7
<b>3</b>	<b>System Requirements and Analysis</b>	<b>9</b>
3.1	Requirements Collection . . . . .	9
3.2	Software Requirements . . . . .	9
3.2.1	Input Requirements . . . . .	9
3.2.2	Output Requirements . . . . .	9
3.2.3	Other Non-functional Requirements . . . . .	9
3.3	Research Requirements . . . . .	10
<b>4</b>	<b>Tools and technology used</b>	<b>11</b>
4.1	Amazon Web Services . . . . .	11
4.2	Python 3 . . . . .	11
4.3	NetworkX . . . . .	11
4.4	Matplotlib . . . . .	11
4.5	Flask . . . . .	11
4.6	HTML5 . . . . .	11
4.7	CSS3 . . . . .	11
4.8	SASS . . . . .	12
4.9	JavaScript . . . . .	12
4.10	Ubuntu 20 LTS . . . . .	12
4.11	NGINX . . . . .	12
4.12	Gunicorn3 . . . . .	12

<b>5</b>	<b>System Design</b>	<b>13</b>
5.1	High Level Design (HLD) . . . . .	13
5.2	Inputs . . . . .	14
5.3	Outputs . . . . .	15
<b>6</b>	<b>System Implementation</b>	<b>16</b>
6.1	Software Modules . . . . .	16
6.1.1	Initializer Module . . . . .	16
6.1.2	Contaminator module . . . . .	16
6.1.3	Analyzer Module . . . . .	16
6.1.4	Server Module . . . . .	16
6.2	Simulation Boundaries . . . . .	17
<b>7</b>	<b>System Testing and Result Analysis</b>	<b>18</b>
<b>8</b>	<b>Conclusion</b>	<b>43</b>
<b>9</b>	<b>References</b>	<b>44</b>
9.1	Primary . . . . .	44
9.2	Secondary . . . . .	44
<b>A</b>	<b>Research Team</b>	<b>46</b>
<b>B</b>	<b>COs, POs and PSOs mapping for the Project work (CS84P)</b>	<b>47</b>
B.1	Course Outcomes . . . . .	47
B.2	Program Outcomes . . . . .	47
B.3	Program Specific Outcomes . . . . .	48
<b>C</b>	<b>Publication Details</b>	<b>49</b>

# **1 Introduction**

## **1.1 Problem Statement**

An Epidemic is an outbreak of a disease that spreads rapidly and widely. Epidemics such as the Bubonic Plague, Avian flu, H5N1 influenza, H1N1 swine flu, SARS etc. have disturbed human lives for centuries causing massive number of deaths and illnesses among people and animals. As the number of urbanized and mobile population has increased, the possibility of a worldwide pandemic has grown too. The coronavirus pandemic is a testament to the same.

Understanding the ways in which diseases propagate has several benefits such as improving healthcare systems, increasing life spans, and reducing the impact of biological warfare. Thus, it is very important for governmental authorities and healthcare agencies to understand how diseases spread and how to limit this spread by selecting the best mitigation and social strategies. Examples of mitigation strategies include vaccination, antiviral treatment, and household prophylaxis. Social distancing strategies include school closure, quarantine, and isolation.

## **1.2 Introduction to Problem Domain**

Historically, epidemics have been modelled mathematically to understand their dynamics. These models can vary from a few simple equations to complex systems that need to be simulated on supercomputers. The latest advances in high performance computing and computational network science can help epidemiologists develop large-scale high fidelity models of the epidemic spread. These models can help guide public health officials and policy makers in taking appropriate decisions to prevent and control the epidemics.

## **1.3 Objectives and Scope**

Objective is the aim or the final result we wish to achieve by completing a certain process. The scope of a process defines the limitations that we have to face while completing a process. For a successive process, the objectives have to be achieved within the scope.

### **1.3.1 Objectives**

- Develop a simulation software with an intuitive UI to enable anyone that is interested to simulate the disease spreading and understand its dynamics.
- Simulate the spread of several hypothetical (not necessarily) diseases on several different kinds of population using the said software, and obtain useful information such as the rate of spreading, the rate of recovery of victims, the epidemic duration etc.
- Understand and appreciate the effect of various proposed safety measures such as social distancing, hand washing etc. amidst the prevailing coronavirus pandemic, through the simulation of the same.
- Understand the various social network models available in the literature, along with their working and relevance.



- Appreciate the contributions of various research scholars through their published work, and also propose any improvements if found and feasible.

### **1.3.2 Scope**

- The simulation will consist of a small sample of population, expected to be around some thousand nodes. This is unlike the real world which consists of over 7 billion humans beings and trillions of other creatures.
- Unlike the real world where the behaviour of people vastly differs from each other and is highly unpredictable, the nodes in our simulated world will have a much more defined behaviour.
- An epidemic will have varied statistics across varying geography. In this simulation however, these statistics will be fed as parameters and will be constant for the entire population.
- There are hundreds of parameters that govern the biological, chemical, and physical dynamics of a disease and its spread through a population. Simulation of all these is impractical, and so we will be going with a small subset of parameters that we believe are the most definitive of an epidemic.

## **1.4 Applications**

- Identifying the critical parameters that contribute the most to the spread of contagions. In other words we can identify what measures can be the most helpful in containing the spread of disease.
- Spreading awareness among people and the government about the necessary measures to be taken for handling the spread of the epidemic.
- In the event of a future pandemic, we can deduce the various critical parameters according to the situation then, as well as the properties of the new contagion.
- Identify the measures that can be taken to minimise economic losses, while still containing the disease.

## **1.5 Existing Solutions**

We had taken a look at multiple research papers, articles and videos to look for existing solutions to the said problem statement. However, pretty much all these focus on a particular disease and do not generalise the results to other diseases. Moreover, the simulations are done through a script which is either not available to the public or isn't "well written", making it difficult to execute them and reproduce the results or formulate new discoveries. Nevertheless, these did help us shape this research and a detailed study on all the papers we've referred is in the upcoming sections.

## 1.6 Proposed Solution

In this research, we simulate the spread of a disease on a small population and record the results. Based on the disease, and the various measures taken to limit its spread, we would accordingly tweak the parameters and observe the variations in the result. We would also like to summarize the main technical challenges in this field. For the simulation we have used the basic Susceptible-Infected-Removed model, also called the SIR model. This is a type of compartment model where time is divided into periods and each individual is classified to a particular state in each period. The population is partitioned into three classes: Susceptible (S), Infected (I), and Removed (R). A person who becomes infected moves from class S to class I at a rate which depends on the infectiousness of the virus and the prevalence of infection. Infected individuals who recover from the infection or die from it move to class R.

## 1.7 Research Timeline

Following is the timeline of the research. In January, since we had exams, the work had been paused briefly.

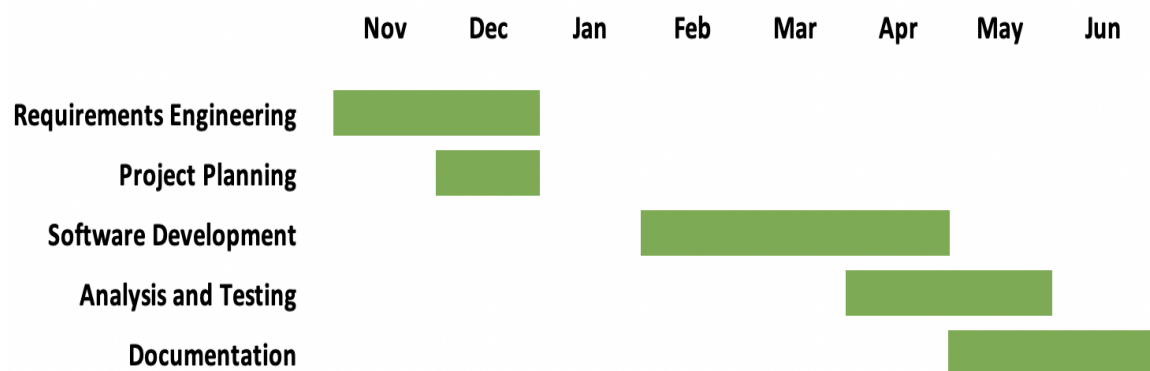


Figure 1: Gantt Chart

## 2 Literature Survey

A literature survey is a way to represent the current knowledge, including any substantive findings as well as theoretical and methodological contributions, on a particular topic. Such a survey helps steer the research work to be carried out in a direction that will make it more useful and / or applicable. Therefore, in this section, we go through some of the research papers that aimed to model the pandemics that plagued the 21st century, and thus helped us get a better understanding of our own problem statement.

### 2.1 SARS: The First Pandemic of the 21st Century

This paper by James D Cherry & Paul Krogstad discusses how after an unprecedented global public health effort, the SARS epidemic was controlled within 7 months of its original occurrence. It also examines the unique pediatric aspects of SARS, and reviews the epidemiology with regards to future epidemics. Some of the relevant information derived from this research was that the incubation period of SARS is 2–10 days, with a median of 4–6 days. The attack rate in children is reported to be less than that of adults, but when consideration is given to the large number of nosocomial cases in original data sets, it appears that children have similar rates as adults. The most striking laboratory finding is absolute lymphopenia. This occurred in nearly all pediatric patients. In one study, 57% percent of children had lymphopenia on presentation, and this frequency rose to 91% . The mean value was  $0.9 \pm 0.7 \times 9/L$ . Other frequently abnormal laboratory findings include thrombocytopenia, elevated lactate dehydrogenase, and creatinine phosphokinase. The research concluded that the most important factor in preventing a future epidemic is sound public health policy and the use of standard infection control procedures.

### 2.2 Strategies for containing an emerging influenza pandemic in South-east Asia

In this paper by Neil M. Ferguson and others, published at the time of the H5N1 influenza, the researchers simulated the spread of epidemics on a model of Thailand having 85 million residents. The model included households, schools, and workplaces. Random contacts in the community associated with day-to-day movement and travel were also modeled. They used a metric known as the basic reproduction number,  $R_0$  to analyze the state of the pandemic.  $R_0$  is defined as the average number of secondary cases generated by a typical primary case in an entirely susceptible population. A disease can spread if  $R_0 > 1$ , but if  $R_0 < 1$ , chains of transmission will inevitably die out. In the simulation, they concluded that for  $R_0 = 1.5$ , the epidemic in the modeled population peaks around day 150 and is mostly over by day 200, at which point 33% of the population has been infected. At  $R_0 = 1.8$ , the epidemic peaks around day 100 and infects about 50% of the population. They further analyzed the effect of social distancing measures, the effect and challenges (such as logistics and supply) of using antiviral prophylaxis. Finally, they concluded that the elimination of a nascent pandemic can best be achieved using a combination of geographically targeted prophylaxis and social distancing measures if the basic  $R_0$  for the virus is below 1.8.

## **2.3 Mathematical models for devising the optimal Ebola virus disease eradication**

In this research by Shuo Jiang and others, a mathematical model was constructed to devise the optimal Ebola virus disease eradication plan by investigating the numerical spread of Ebola and its eradication pathways. By incorporating hospital isolation and application of medication in the model, and analysing their effect on resisting the spread, it demonstrated the second peak of 10,029 total cases. Using the regional spread of EVD with transmission model analysis, it analysed the numbers of new infections through four important transmission paths including household, community, hospital and unsafe funeral. Based on the result of the model, key paths in different situations were found, and some suggestions were proposed to control regional transmission considering all the Ebola characteristics, economic and time optimization, dynamic factors and local condition constraints etc.

## **2.4 Homogeneous and network models in Epidemiology**

An exact contact network model requires knowledge of every individual in a population and every disease-causing contact between individuals (e.g. sneezing in the case of airborne diseases or sexual contact in the case of sexually transmitted diseases). For even small populations, this is typically unfeasible, and thus the researchers worked with approximate networks. There are several techniques for gathering the information needed to build realistic contact network models. They included tracing all infected individuals and their contacts during an outbreak, surveying individuals in populations and using census, social characteristic or other collected data. The researchers also focused exclusively on static networks, i.e. networks in which contacts are assumed to be fixed during the infectious period of an individual. The permanence of contacts captured by static networks offered a more realistic model of human contact behaviour than that by traditional epidemiological models.

## **2.5 FluTE, a publicly available stochastic influenza Epidemic simulation model**

FluTE is an individual-based simulation model of influenza epidemics. In this, researchers described the model's community structure, the natural history of influenza, and simulated interventions. Briefly, all individuals in the model are members of social mixing groups, within which influenza is transmitted by random mixing. The model can simulate several intervention strategies, and these can either change the transmission characteristics of influenza (e.g., vaccination) or change the contact probabilities between individuals (e.g., social distancing). Interventions can occur before the epidemic or in response to an ongoing epidemic.

To help prepare for future influenza seasonal epidemics or pandemics, they developed a new stochastic model of the spread of influenza across a large population. Individuals in this model have realistic social contact networks, and transmission and infections are based on the current state of knowledge of the natural history of influenza. The model has been calibrated so that outcomes are consistent with the 1957/1958 Asian A(H2N2) and 2009 pandemic A(H1N1) influenza viruses. They presented examples of how this model can be

used to study the dynamics of influenza epidemics in the United States and simulate how to mitigate or delay them using pharmaceutical interventions and social distancing measures.

## **2.6 Agent-based simulation on avian influenza in Vietnam**

In this paper, based on the daily reported number of dead poultry due to avian influenza in northern Vietnam in November of 2005, authors used a mathematical model to estimate the basic reproduction number  $R_0$  of the disease. The significant value of the estimated  $R_0$  explained the explosive outbreak in the poultry population. The authors developed an SIR compartment using a combination of EBM and an ABM to recapture the recorded data of the outbreak of avian influenza and to evaluate the efficiency of existing control measures. Their model assumes a totally homogeneous and well-mixed poultry population where the interaction between infected and susceptible individuals is a random process. The work concluded that the infection process of avian influenza in poultry is not significantly affected by external factors. The results inferred that a comprehensive strategy of culling, bio-security control and large-scale vaccination campaign should be taken promptly to keep the disease under control. Advantages of this approach include evaluation of control and vaccination strategies, as well as combining EBM and ABM. An Assumption was that poultry population was considered to be totally homogeneous and well mixed and this is one of the limitations of this approach.

## **2.7 Cholera Epidemic in Haiti, 2010**

In this paper the authors built a compartmental transmission model for the Vibrio Cholera Epidemic in Haiti in 2010 and 2011 and explored potential effects of disease-control strategies. They developed an SIR model with the addition of a water compartment. The water compartment could be contaminated by infected or infectious persons and could in turn infect susceptible persons. The models represented the population of each of Haiti's ten administrative regions. These populations were combined to form a meta-population model, in which disease could spread both within a given region and between regions reflecting the movement of people between different regions. It was found that Cholera spread between two regions is proportional to the population size of both regions and inversely proportional to the square-distance between regional centroids. Analysis of changes in disease dynamics over time suggests that public health interventions have substantially affected this epidemic. A limited vaccine supply provided late in the epidemic was projected to have a modest effect. One major assumption was that cholera could be transmitted through either close contacts or contaminated water. However, waterborne transmission and consumption of food items contaminated with infective water, which are important ways of cholera transmission, were not considered in this simulation.

## **2.8 Global seasonal occurrence of MERS-CoV infection**

This paper aims at investigating the global seasonal occurrence of Middle East Respiratory Syndrome coronavirus (MERS-CoV) outbreaks. The authors obtained the data on the prevalence and occurrence of Middle East Respiratory Syndrome Coronavirus (MERS-CoV) infection from the World Health Organization (WHO) for all the MERS cases reported from the various countries and their allies ministries. The conclusion of this research was that MERS-CoV infection affected 2048 people worldwide; 82% cases were reported

from the Saudi Arabia and 17.96% cases were reported from other countries worldwide. The maximum number of cases 23.14% were reported in the month of June. However, low occurrence of infections were seen in the month of January. The health sectors need of awareness programs to mandate implementation of effective control strategies and stringent compliance with better standards of health and hygiene nationwide. The health officials also need to highlight the seasonal occurrence of MERS-CoV suggesting to take the better preventive measures to minimize the disease burden globally.

## **2.9 The effects of border control and quarantine measures on the spread of COVID-19**

In this research paper the authors have developed an “easy-to-use” mathematical framework extending from a meta-population model embedding city-to-city connections to arrange the dynamics of transmission waves caused by imported (people who have been associated with travel history from an epidemic region), secondary, and others from an outbreak source region when control measures are considered. Using the cumulative number of the secondary cases, the authors try to determine the probability of community spread.

Using the top 10 visiting cities from Wuhan in China as an example, it is demonstrated that the arrival time and the dynamics of the outbreaks at these cities can be successfully predicted under the reproduction number  $R_0 = 2.92$  and incubation period  $T = 5.2$  days. It is also shown in the study that although control measures can gain extra 32.5 and 44.0 days in arrival time through an intensive border control measure and a shorter time to quarantine under a low  $R_0$  (1.4). The study allows to assess the effects of border control and quarantine measures on the emergence and global spread of COVID-19 in a fully connected world using the dynamics of the secondary cases and demonstrated that after the reporting delay was estimated, the dynamics of the outbreaks at connected cities can be successfully reconstructed using both the imported and the secondary cases. The result implies that all the connected (direct or indirect) countries are having a great risk of outbreak and that if the epidemic growth at the source location is high, even a near full-scale border control without proper quarantine measures, will have only limited effects.

## **2.10 A Social Network Model of the COVID-19 Pandemic**

This social network model developed by Pei Jun Zhao, Harvard T.H. Chan School of Public Health, provides a conceptual framework to analyze the COVID-19 pandemic on the level of individuals, that can be used to predict the influence of collective social behavior on overall pandemic trajectories. It shows the importance of social distancing, and the message that to contain the epidemic, every member of the public plays a crucial part in breaking the chain of transmission. The social network model of COVID-19 transmission is expandable and extendable. This paper provided a simplified version using population averages for parameters.

## **2.11 Simulation of the COVID-19 epidemic on the social network of Slovenia**

In this paper by Žiga Zaplotnik and others, the researchers developed a virus transmission model on a simplified social network of Slovenia with 2 million nodes organized into

home / care center clusters. They found that when the epidemic is uncontrolled, the intrinsic uncertainty mostly originates from the uncertainty in virus transmission, while the randomness of the social network has only minor impact of the final size of the epidemic. The latter is in line with a study, where the social network was constructed based on extensive contact survey data, and which reported only minor impact of reshaping the network structure or removing the variance of connection weights on the final size of the epidemic. On the opposite, in the controlled epidemic with low infected population, the randomness of the social network becomes the major source of forecast uncertainty. They also showed that the uncertainty of the forecast and the associated risk is extremely asymmetric (roughly symmetric on a logarithmic axis) with long exponential tails.

## **3 System Requirements and Analysis**

As this is a research-based project, and there are no stakeholders other than us, the researchers, the traditional approach of requirements engineering and SRS documentation isn't applicable here. However, we can try and formalize the goals and objectives of this research in a more systematic way. We can also list out the limited set of requirements for the simulation software.

### **3.1 Requirements Collection**

The requirements gathering and negotiation phases were conducted by virtual meetings of the team members and the mentor in the month of November. Further discussions were carried out by the team members over a period of two weeks and clarity was obtained with respect to the research work. It was decided that in order to model the population in our simulation we will be using a SIR model, as the literature survey revealed that most research on this topic used the same. The different parameters to be considered in the research include disease parameters such as its incubation period, the total time an individual is affected on an average, the mortality rate, the probability and radius of infection, etc. and social parameters such as probability and strictness of quarantine measures, the social distancing maintained, the rate of travel, presence of central locations and communities, etc.

### **3.2 Software Requirements**

#### **3.2.1 Input Requirements**

The inputs must correspond to the various parameters associated with a typical airborne contagious disease. They must also correspond to the various parameters that define a population structure. The specific list of inputs is provided in the upcoming System Design section. Lastly the inputs must be in the form of sliders with appropriate min, max, and step values.

#### **3.2.2 Output Requirements**

The output must be in the form of graphs, specifically the SIR plot, daily new cases plot, total cases plot, and the daily effective reproduction plot. The output must also include statistics such as the number of peak cases, the total number of cases, the total duration of the epidemic, etc.

#### **3.2.3 Other Non-functional Requirements**

The software must have a neat and intuitive user interface (UI). When simulating, the execution must not take more than a few seconds to complete.



### **3.3 Research Requirements**

- The software developed, and the findings must be open sourced. This is so that if anyone in future would like to continue the research, can do so.
- The research must be well documented.
- The software should be written with highest possible code quality so as to make it extendable to incorporate more features when needed
- The research must be completed well within the stipulated time.

## **4 Tools and technology used**

### **4.1 Amazon Web Services**

Amazon Web Services offers cloud web hosting solutions that provide businesses, non-profits, and governmental organizations with low-cost ways to deliver their websites and web applications. We used this as the hosting platform to host the simulation software during the research phase (2nd half) of this project.

### **4.2 Python 3**

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. The complete backend of the software was written with this language.

### **4.3 NetworkX**

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. We used it to generate the graphs needed for simulation

### **4.4 Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

### **4.5 Flask**

Flask is a framework used to develop websites in Python. We used it to convert and serve our simulation script. Flask is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

### **4.6 HTML5**

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

### **4.7 CSS3**

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

## 4.8 SASS

SASS is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets. SassScript is the scripting language itself. Sass consists of two syntaxes. The original syntax, called "the indented syntax," uses a syntax similar to Haml.

## 4.9 JavaScript

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

## 4.10 Ubuntu 20 LTS

We used Ubuntu 20 LTS as the operating system to run the server i.e. AWS EC2 instance. It is a Linux distribution based on Debian and composed mostly of free and open-source software. Ubuntu is officially released in three editions: Desktop, Server, and Core for Internet of things devices and robots. All the editions can run on the computer alone, or in a virtual machine like our EC2.

## 4.11 NGINX

Nginx (pronounced "engine-x") is an open source reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer, HTTP cache, and a web server (origin server). The nginx project started with a strong focus on high concurrency, high performance and low memory usage. It is licensed under the 2-clause BSD-like license and it runs on Linux, BSD variants, Mac OS X, Solaris, AIX, HP-UX, as well as on other \*nix flavors. It also has a proof of concept port for Microsoft Windows.

## 4.12 Gunicorn3

Gunicorn 'Green Unicorn' is a Python WSGI HTTP Server for UNIX. It's a pre-fork worker model ported from Ruby's Unicorn project. The Gunicorn server is broadly compatible with various web frameworks, simply implemented, light on server resource usage, and fairly speedy.

## 5 System Design

In this section, let us discuss the high level design / architecture of the simulation software as well as the inputs and outputs.

### 5.1 High Level Design (HLD)

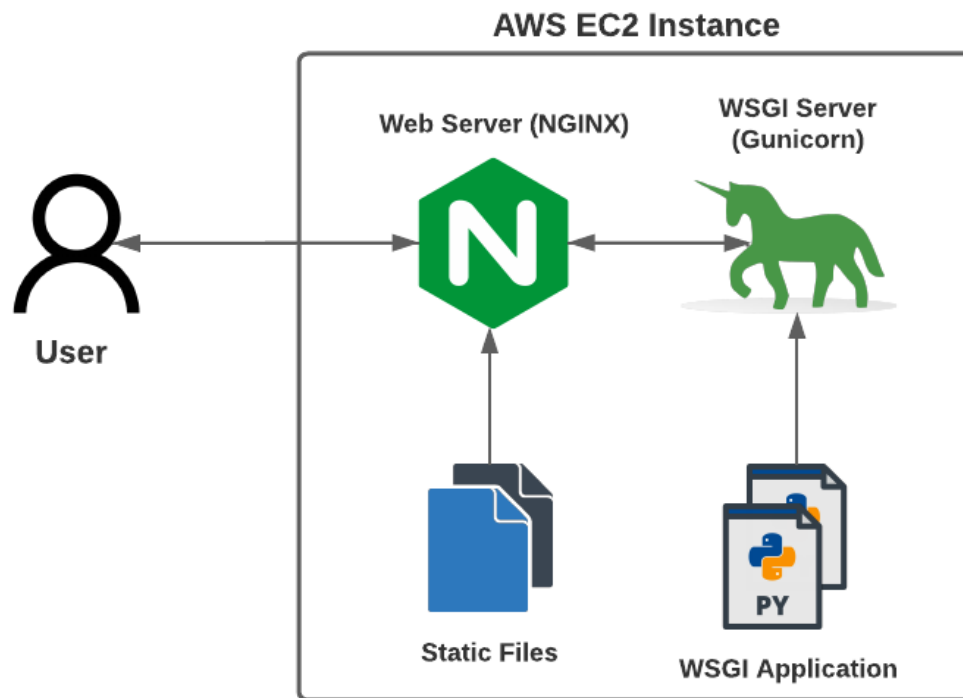


Figure 2: System HLD

The high level end to end design of the simulation software is as shown in the figure. The software is hosted on an AWS EC2 instance, which is an IaaS offering from AWS. On this instance, an NGINX web server exists that listens to HTTP requests (on port 80) by the end users. If the received request is for a static file such as an image, a CSS file, or any such, then NGINX serves the requested file directly. All other requests are routed to Gunicorn, which is a WSGI server i.e. a server capable of running WSGI applications such as the Flask application that we are developing. In this case, NGINX acts as a proxy between the user and the WSGI server. Gunicorn serves this request using the flask application defined in the form of python scripts, and returns the response to NGINX, which will in turn return it to the user. In case one does not have a hosting solution such as AWS and just wants to test the software, then he can use the default development server that comes with Flask instead of Gunicorn and send the HTTP requests locally.

## 5.2 Inputs

Following are the simulation parameters that the software takes as inputs to the simulation

- Population i.e the number of nodes in the graph
- Community centres i.e the minimum number of nodes that need to be connected to at least a tenth of all the nodes
- Average number of friends that an individual has i.e. average degree of nodes
- Quarantine strictness when someone is quarantined
- Quarantine probability when a person is found to be infected
- Total hospital beds available
- Average immunity of the population, with 1 and 0 being 100% and 0% immune to the disease
- Immunity variation\*
- Average social distance (in metres) maintained by the people
- Social distance variation\*
- Average adherence to safety measures by the people in the range 0 and 1
- Adherence to safety measures variation\*
- Average travel probability of the people in the range 0 and 1
- Travel probability variation\*
- Travel ban strictness imposed by the authorities
- Average number of physical contacts between two people in a day
- Number of physical contacts variation\*
- Average duration (in days) the infection affects the infected
- Infection duration variation\*
- Probability (calculated on a daily basis) of infected person dying
- Probability that the virus infects i.e. spreads from one person to another given all favourable conditions
- Infection radius i.e. maximum physical distance (in metres) between two people after which the virus fails to spread
- Probability that an infected person is asymptomatic

\* One half of the difference between max possible value and min possible value of the parameter being considered. This helps in distributing the parameter over the population in a binomial fashion.

### **5.3 Outputs**

The output of the simulation software is a set of statistical graphs (SIR, Daily new cases, and Total recorded cases for now) as well as key statistics (peak active cases and total duration of the epidemic for now) that describe the spread of the virus in the simulated epidemic.

## 6 System Implementation

From the HLD, we identified 4 components as part of this software system. NGINX and Gunicorn are standardized open source components that have been developed extensively and are available for free for anyone to use. The static files are simple CSS and JavaScript files that do not need any detailed explanation. Therefore, in this section, let us discuss the low level implementation of the WSGI application.

### 6.1 Software Modules

#### 6.1.1 Initializer Module

This module is responsible for generating a graph using NetworkX. The graph, whose structure is defined through the given inputs, represents the population of a city, with the nodes and edges representing people and the physical connectedness among them respectively. The graph (as a whole), nodes, and edges contain parameters defined through the input, and these will be used during the simulation. This module also sets up a few other things such as seeding the randomizer, that are necessary for the simulation.

#### 6.1.2 Contaminator module

This module is responsible for simulating the spread of the defined disease over the population. It uses BFS to do so, however with slight changes. Each outer BFS loop represents a day for our population, and so in order to provide each person with the same probability of infecting others, the queue is shuffled each day. Moreover, unlike traditional BFS, here the nodes aren't removed from the queue after processing, instead are retained until it either dies or recovers from the disease. During the simulation it considers the various input factors to progress until the epidemic is over. In the end it returns the SIR statistics of each day.

#### 6.1.3 Analyzer Module

This module is responsible for performing analysis using the SIR statistics obtained from the contaminator module. To do so, it first runs the simulation using the said module. Then, it plots three graphs, one for the SIR statistics, one for the total infections until date, and one for the new infections each day. The latter two are also plotted based on the same SIR statistics. Finally, it stores the graphs as a PNG image and returns the filename, along with the peak active cases of the infection and the total duration of the epidemic during the simulation.

#### 6.1.4 Server Module

This module is responsible for accepting the inputs from the user through the UI, constructing a simulation instance using the constructor module, using the analyzer module to run the simulation and get the results, and then sending the results back to the user in the UI. It is a standard Flask module that converts the Python scripts to a WSGI application.

## 6.2 Simulation Boundaries

Considering the time available as well as our present skills in the field, there were a few compensations we had to make in terms of what and what is not simulated. Some of the key limitations include:

- The various parameters being considered for the simulation are constants. But in reality, we see these parameters varying each day in an unpredictable manner. For example, at the outbreak phase of the disease people follow safety measures more strictly than when the disease has become “normal”. Constant parameters also means that there won’t be any waves of diseases in the simulation
- There are no community clusters in the network structure, instead the network is just one big cluster. This means that we are not simulating isolated families / colonies / communities / cities etc. that could very well be the case in reality
- The simulation has been designed with COVID-19 in mind, and hence it might not be completely applicable to other diseases, such as the ones which spread through water.
- The severity of the disease can vary among people. This also means hospitals can be preferential when it comes to treatments. But in our simulation we are assuming the severity to be the same for everyone, and hence the hospitals of our simulation too, will treat everyone equally
- A hospitalized person would have a lesser probability of dying than someone who is home quarantined because of the medical attention he can receive. We have however not considered this, as, for the SIR statistics it doesn’t really matter



## 7 System Testing and Result Analysis

In this section, let us go through the results of various simulations and understand what parameters matter the most when it comes to the dynamics of a disease propagation. Before that, let us establish a baseline against which we can compare the results. The following diagram is a screenshot of the UI of the software. We can see the various settings used as the baseline.

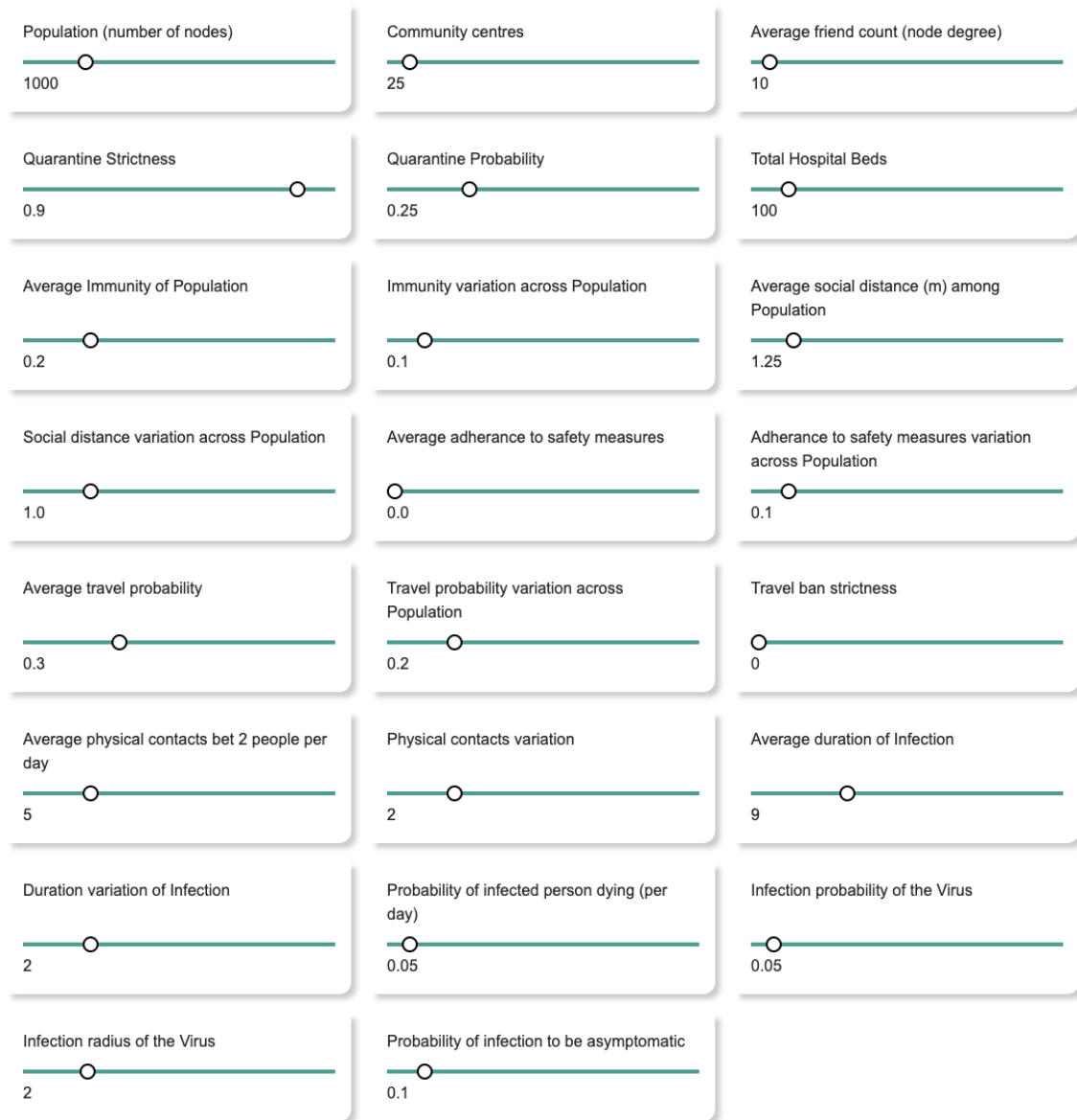


Figure 3: Baseline Inputs

For the baseline settings, the epidemic lasts for 22 days with all 1000 nodes getting infected. 13th day reported the last new case, and after that everyone had been infected. The peak of the SIR curve reached 875, and thus crossed the available hospital beds which is definitely not a good situation. The effective reproduction number had some noise in the beginning, but when the cases start rising it ranged around 1.75 to 2.5 before declining.

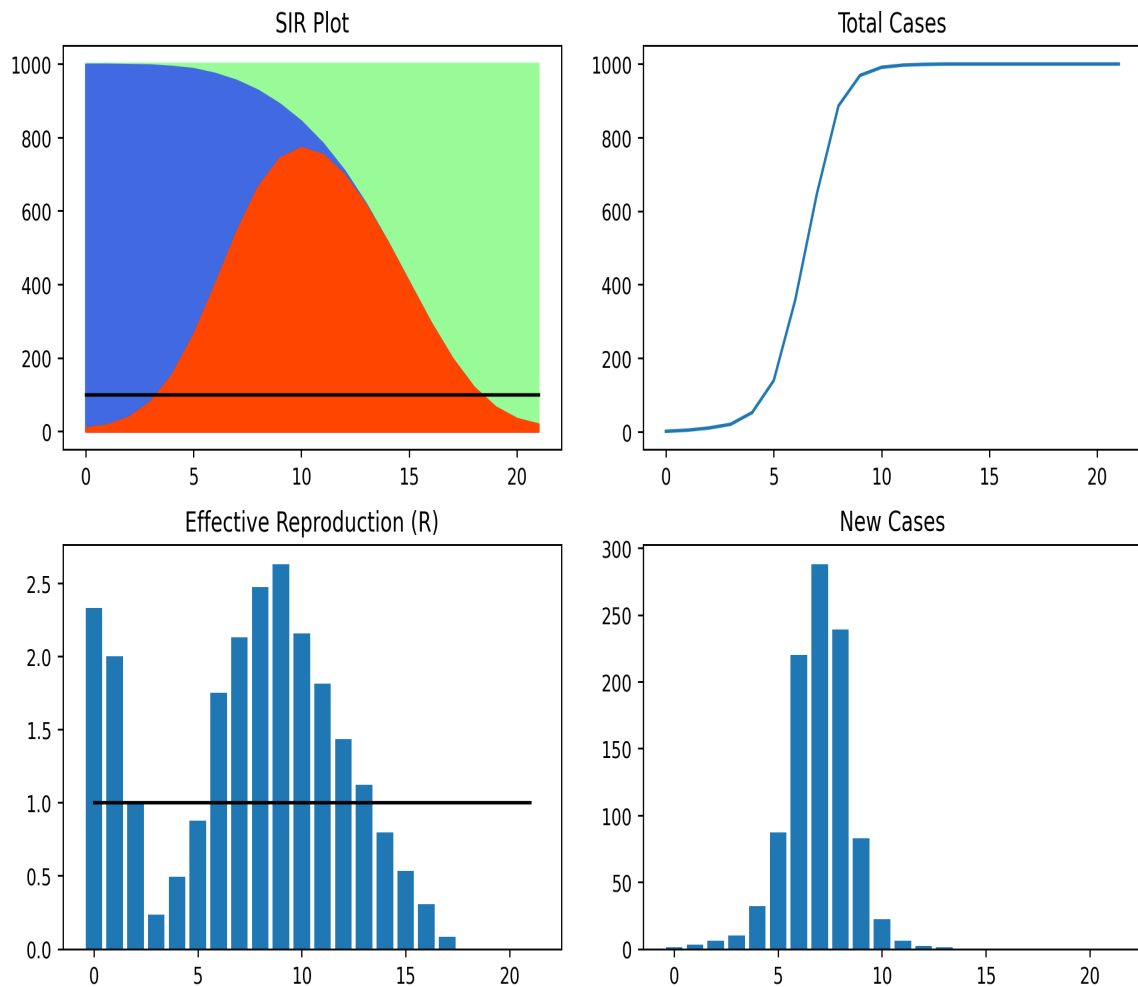


Figure 4: Baseline Statistics

When we increase the “connectedness” of the graph by increasing community centre count to 100 and average friend count to 20, the total duration falls down to 17 days, but the peak cases rise to 940. This was no surprise. We can say that the disease was encouraged to spread rapidly, and hence was over quickly. But of course, this is no solution to the problem

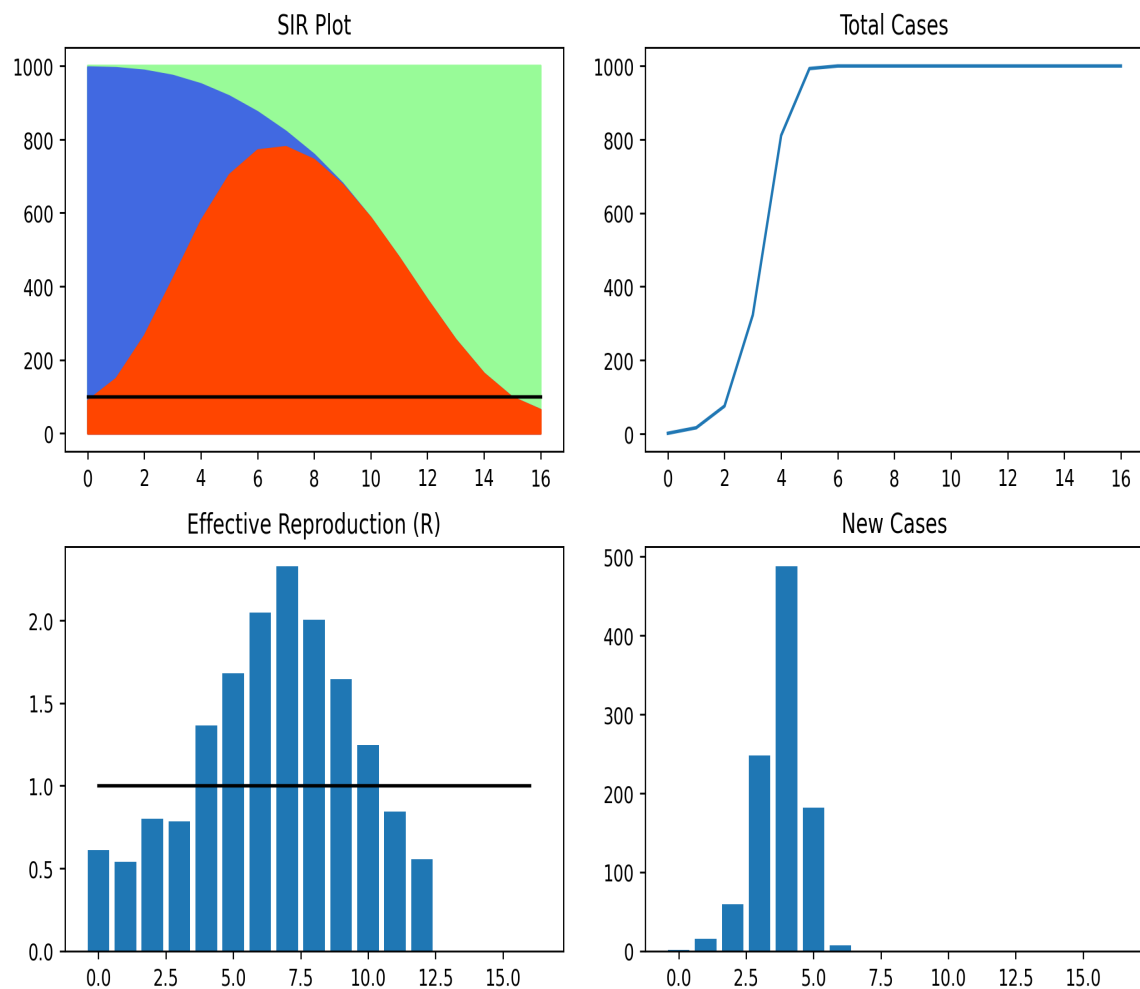


Figure 5: Community Centres = 100, Average Friend Count = 20

When we go to the opposite end and decrease the connectedness of the graph by having only 10 community centres and only 5 friends on average, we once again see the expected results. The peak cases comes down to 681, and total cases down to 977 (i.e. few people were unaffected). We still crossed the number of beds available though. The effective reproduction might seem counter intuitive at first peaking at almost 2.75, unlike the previous case where it didn't even reach 2.5. But on closer inspection, this is because the number of people causing secondary infections are lesser. What matters here is that during the rise in cases, it too was on an uptrend and was above 1, signalling the increase. When the cases peaked out,  $R$  started on a downtrend.

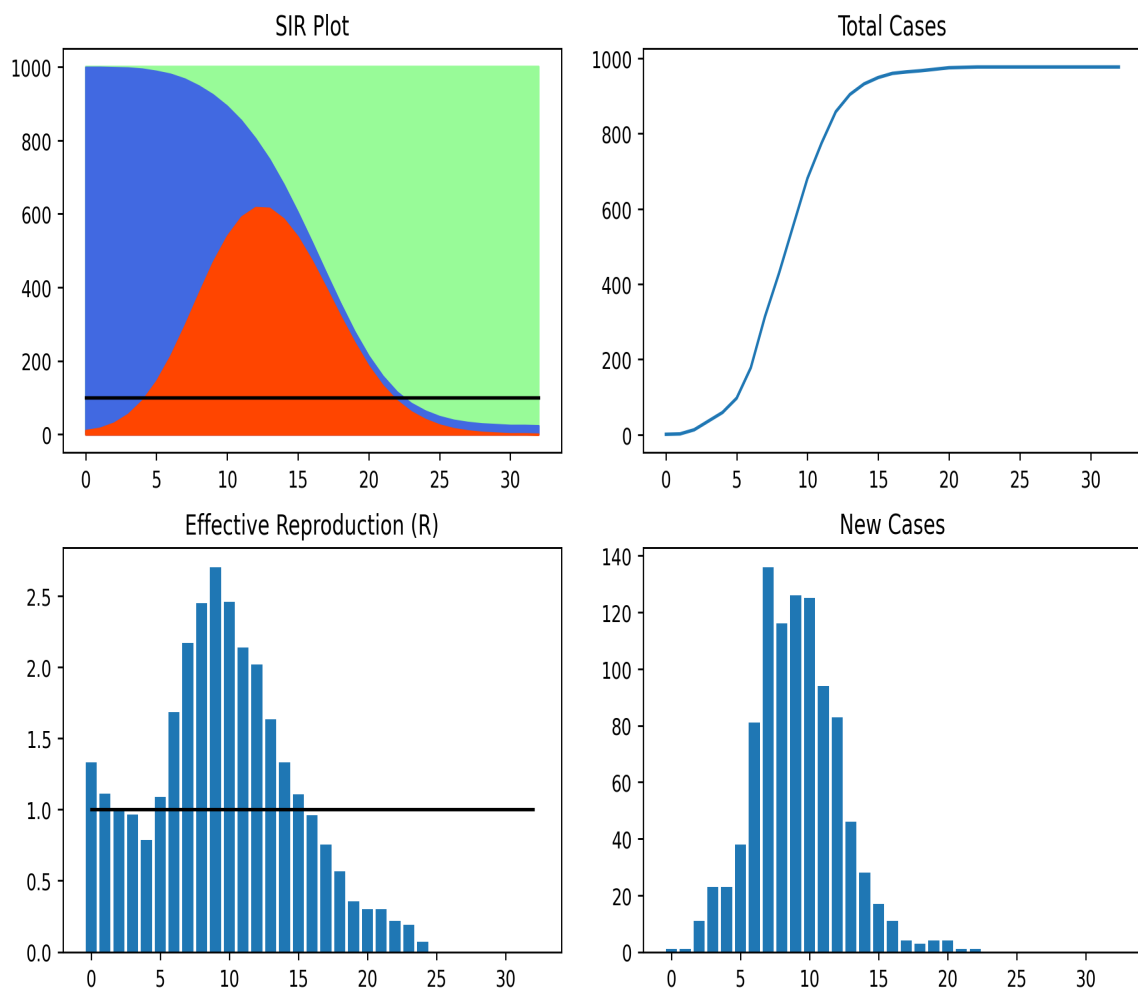


Figure 6: Community Centres = 10, Average Friend Count = 5

Next let us try changing the travel parameters, namely the travel probability and the travel strictness. With travel probability set to 0.9 and travel strictness at 0 i.e. no travel ban, we didn't see much difference from the baseline. The duration was the same and once again everyone was infected. In fact the peak cases slightly decreased, although this can be dismissed as noise. So why didn't the situation get drastically worse with everyone travelling? This was because in our model we had considered the population to be one giant cluster, without any smaller "sub clusters". As a result there was no effect if people travelled or not. It's too late to go back and change the graph structure in code, but that's fine as we've got plenty of other parameters to work with. We also tried going the other extreme with a complete travel ban by setting the travel strictness to 1, and once again we didn't see any major difference.

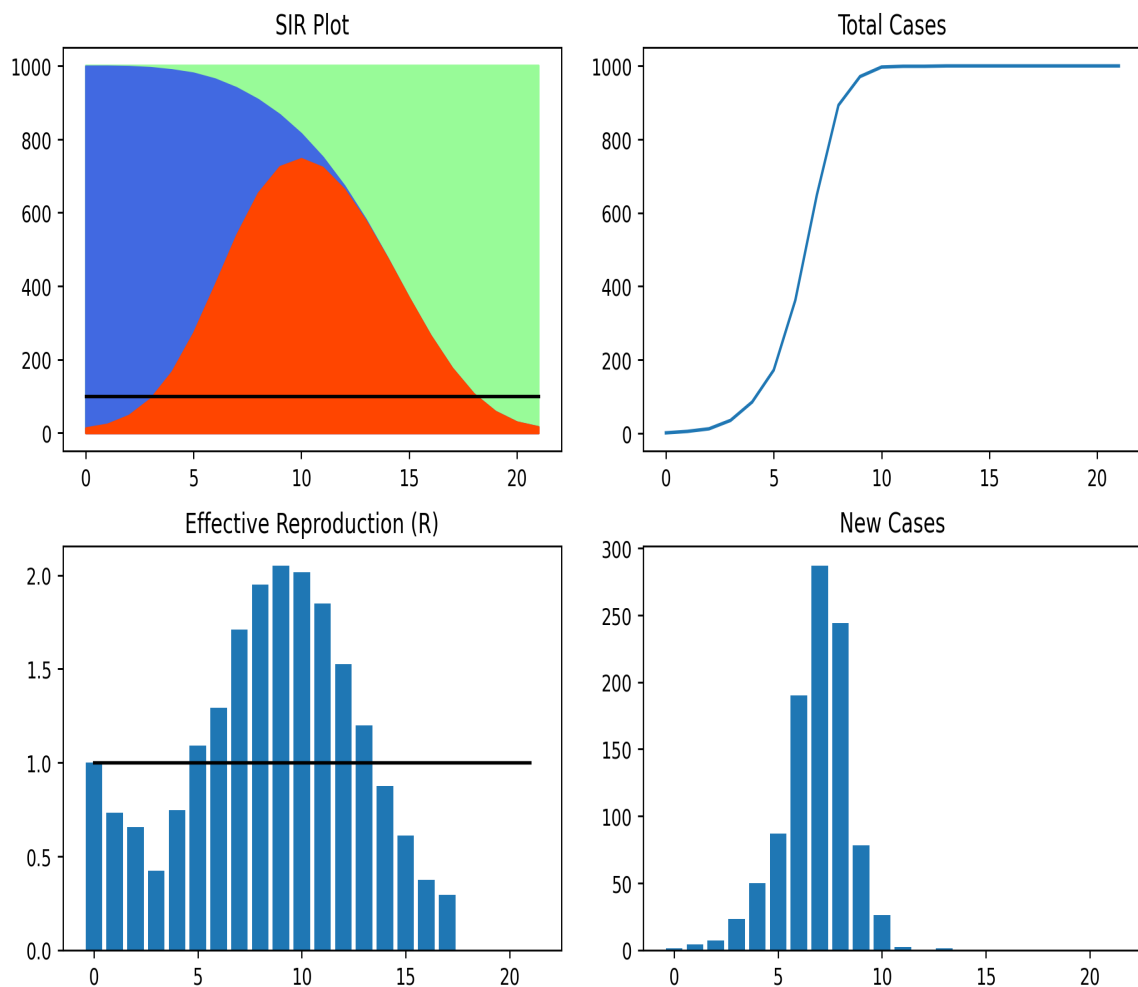


Figure 7: Average travel probability = 0.9

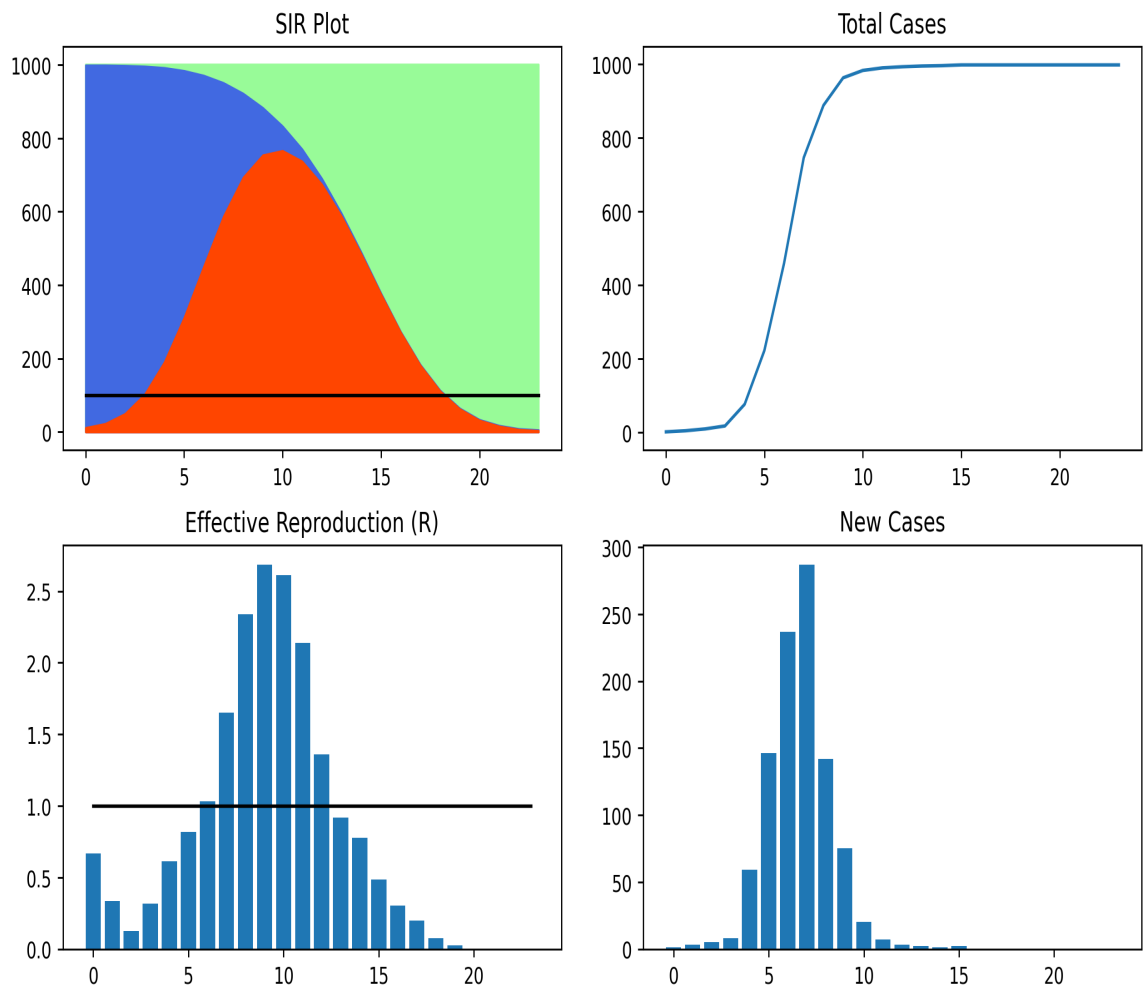


Figure 8: Travel ban strictness = 1

Average friend count in this simulation is the degree of each node in the graph, which shows us how many nodes are connected to a single node. We have taken the base degree as 10 but as we increase this degree to 60 we see certain changes in our graphs. First the slope of the total cases graph increases which represent that the number of nodes infected increases at a faster rate. As can be seen through the new cases graph where we can see how fast all the population is infected.

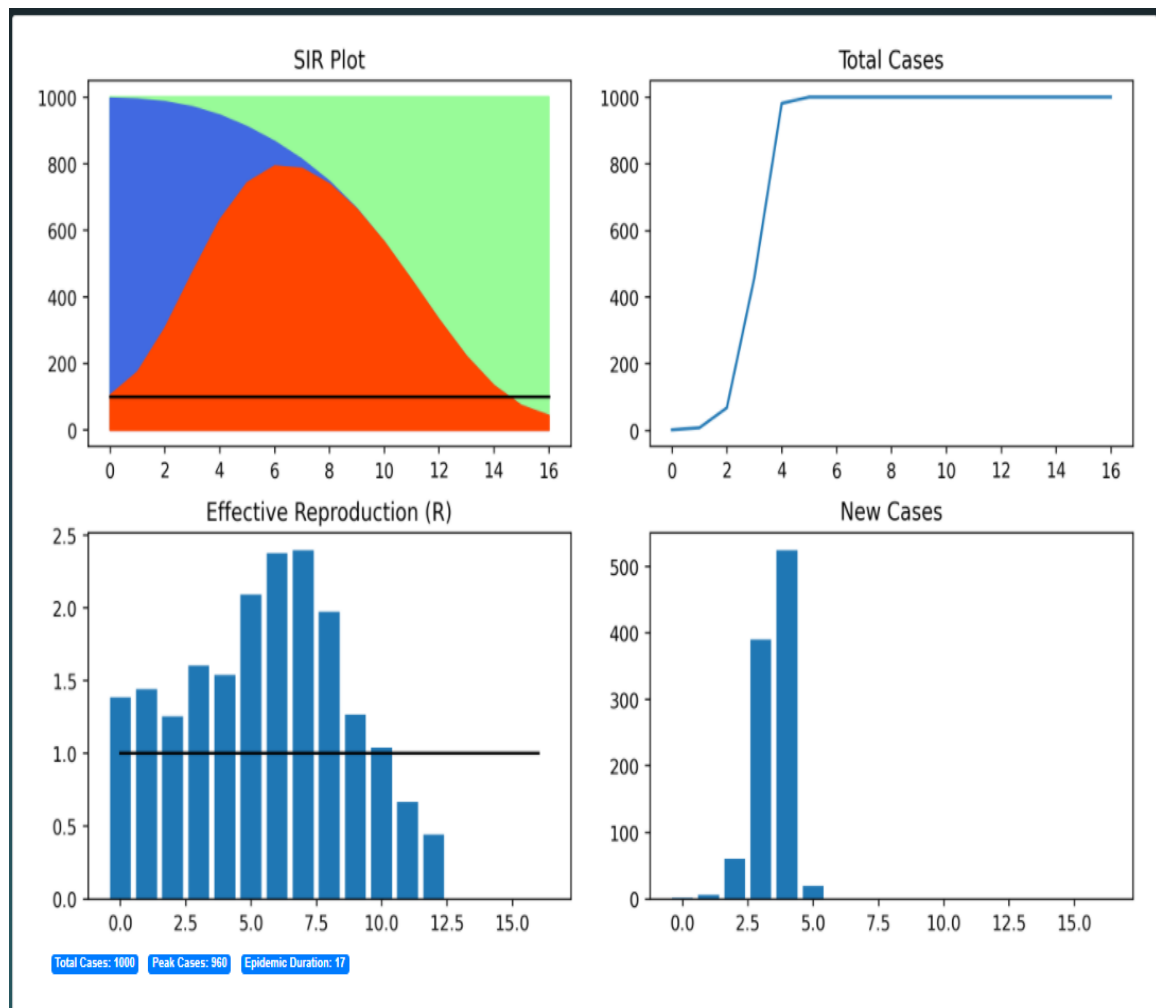


Figure 9: Average friend Count = 60

Average Social distance is the measure in meters between 2 nodes. This is used to determine the physical distance between two nodes which is the sum of average social distance and social distance variation, this physical distance will be one of the parameter to decide whether an adjacent node can be infected or not since if this physical distance is larger than the infection radius the disease cannot be transmitted. From this we can easily concur that if we increase the average social distance between nodes, less nodes will be infected. As is visible in the graph, if we increase the average social distance to 2.2 from our base value of 1.25 we see in the SIR graph that the susceptible population is less than the base graph. Also the new cases graph is more spread out which means more time is taken for spreading of the infection.

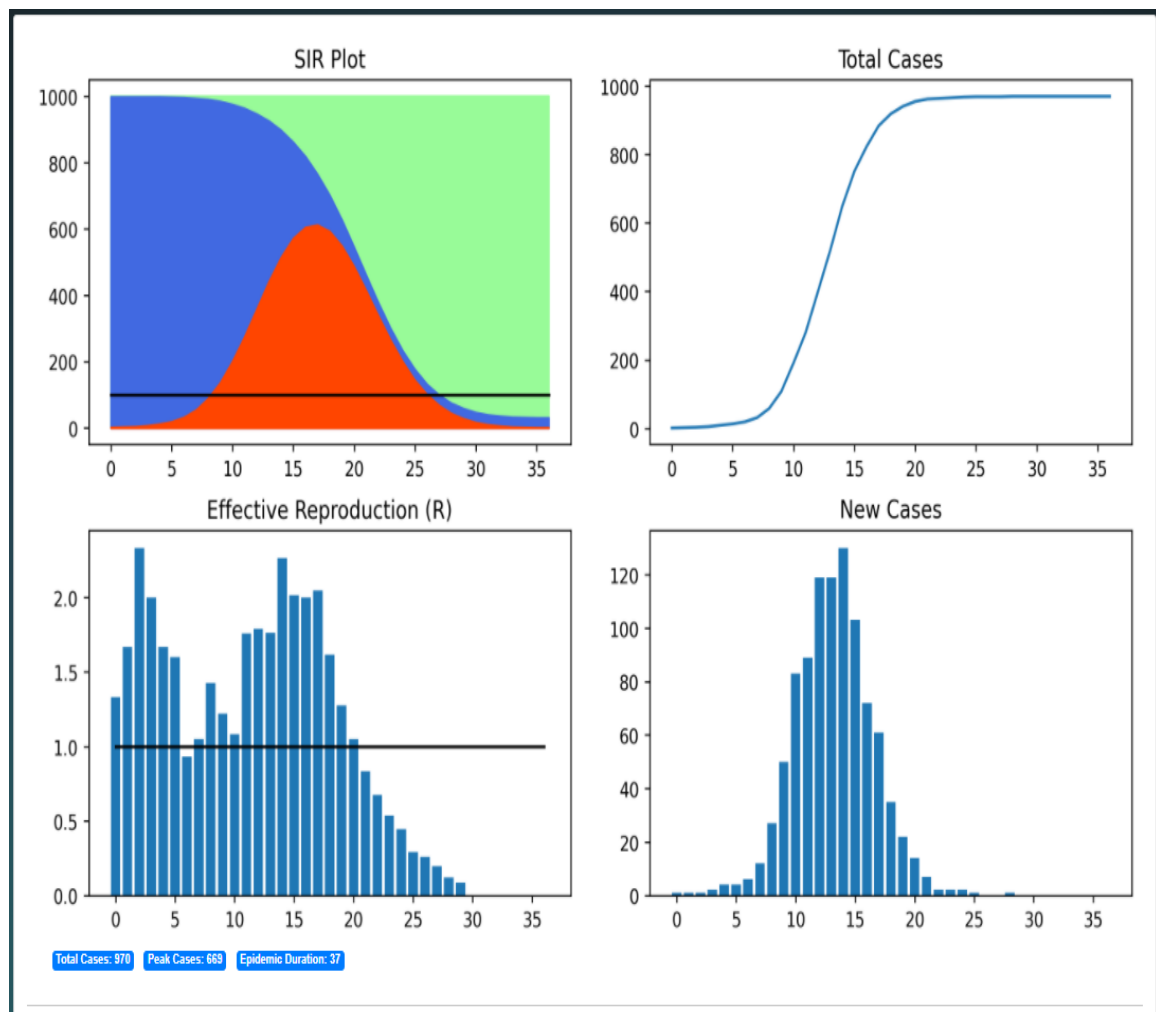


Figure 10: Average Social Distance = 2.2



To see whether contamination between two nodes is possible or not we have to see how many physical contacts a particular node makes. For this we have a parameter called as average physical contact between 2 nodes per day. For these physical contacts we check whether infection is possible between nodes considering other parameters also like immunity, quarantine probability etc. But if we keep other parameters we can say that if the average number of physical contacts increase the infection will spread faster, as is visible in the graph below where the average physical contact has been increased to 20 from base value.

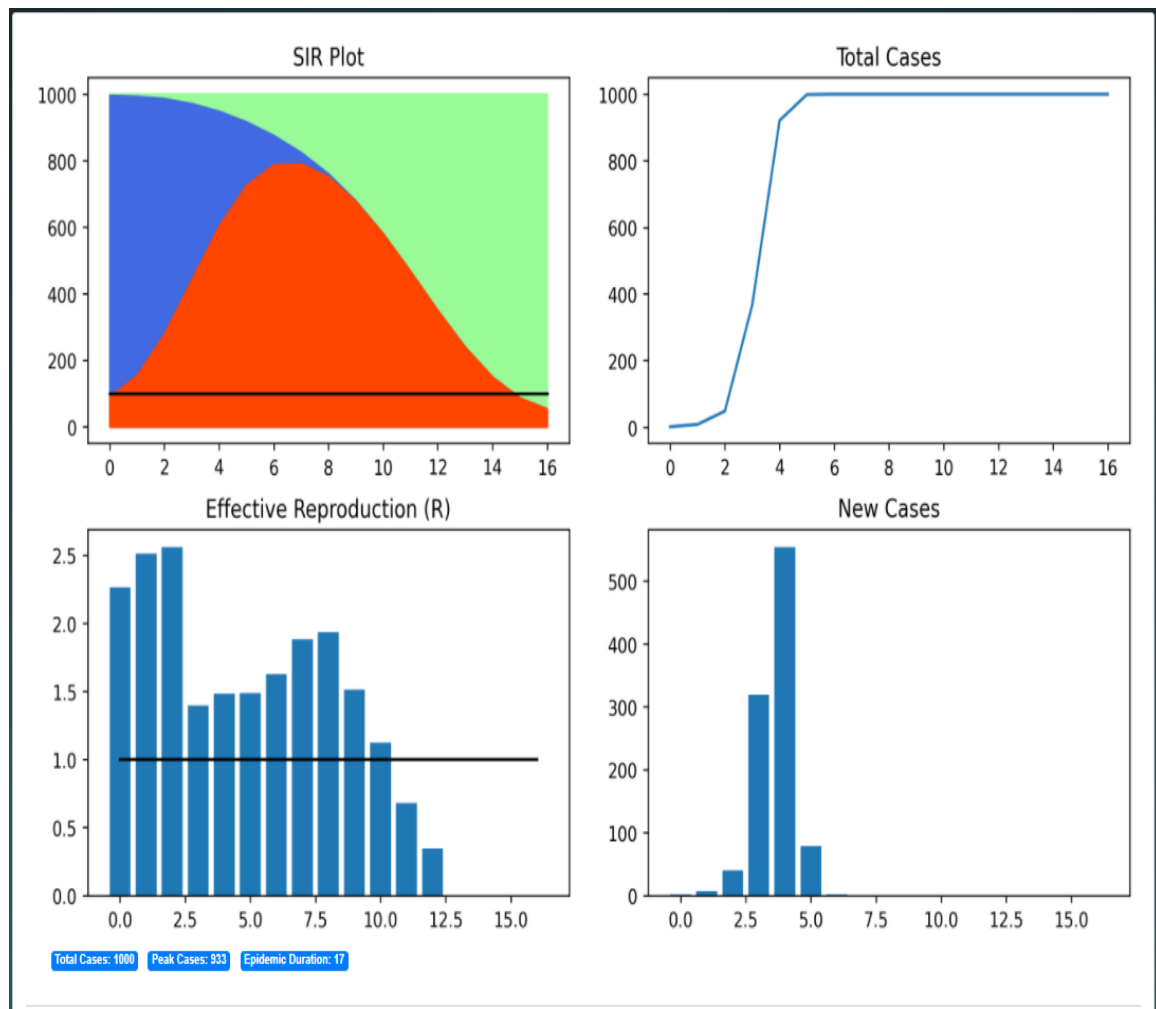


Figure 11: Average number of physical contact between 2 people in a day = 20

As depicted above that increasing the average physical contacts increase the infection rate in the population, but we know that many other factors are also working for determining the infection rate like if we increase the average physical contact to 20 as shown above but with that we increase the average immunity of the population also to 0.8 from base value of 0.2, we see that although more people are coming in contact but since the immunity is increased the chance getting infected gets decreased. In the graph below we can see the new cases graph is more spread out and in the SIR graph also the increase in susceptible cases is not sudden whereas it increases slowly.

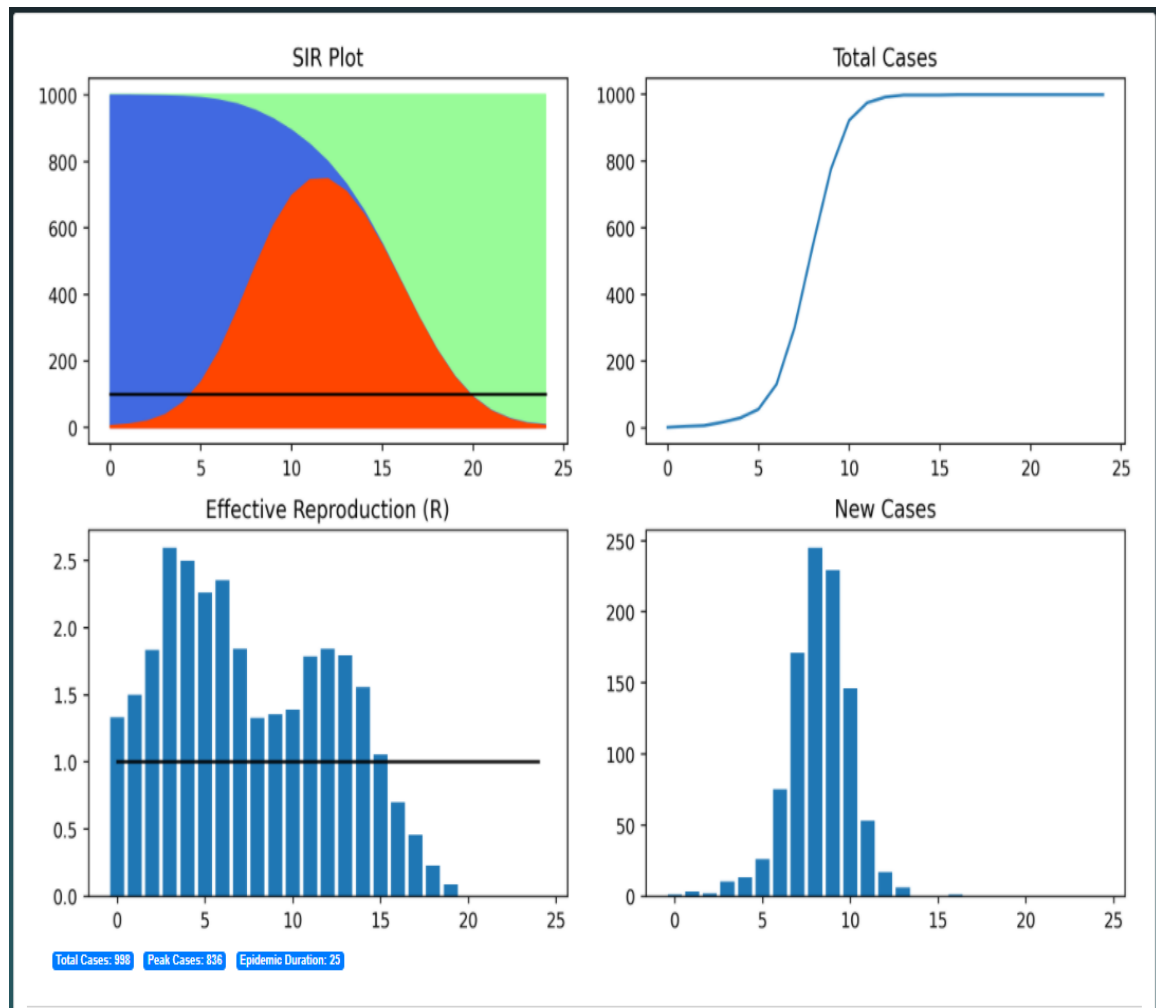


Figure 12: Average number of physical contact between 2 people in a day = 20 average immunity = 0.8

For the baseline, we have kept quarantine strictness high which is quite likely in real scenario. But if we change it from 0.9 to 0.15, we observe that the disease takes more time to eradicate and also the peak number of cases is reached in less time.

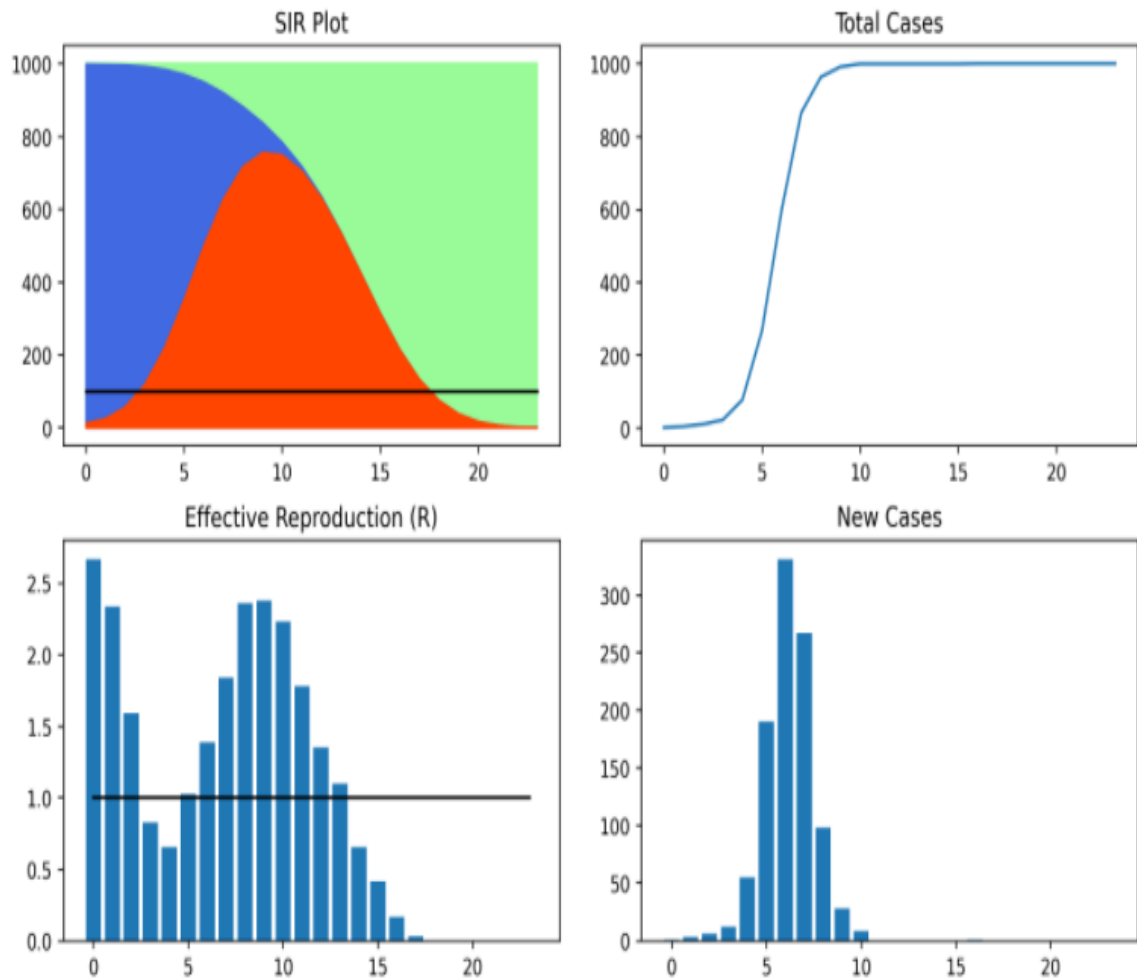


Figure 13: Quarantine strictness = 0.15

Quarantine Probability - It represents the probability of an infected person to be quarantined. For reference we have taken quarantine probability to be 0.2. Changing it to 1, i.e if everyone being affected is quarantined, it is observed that number of peak cases is dropped and a few people are not affected at all. Although it is also observed that it takes more time for the disease to eradicate.

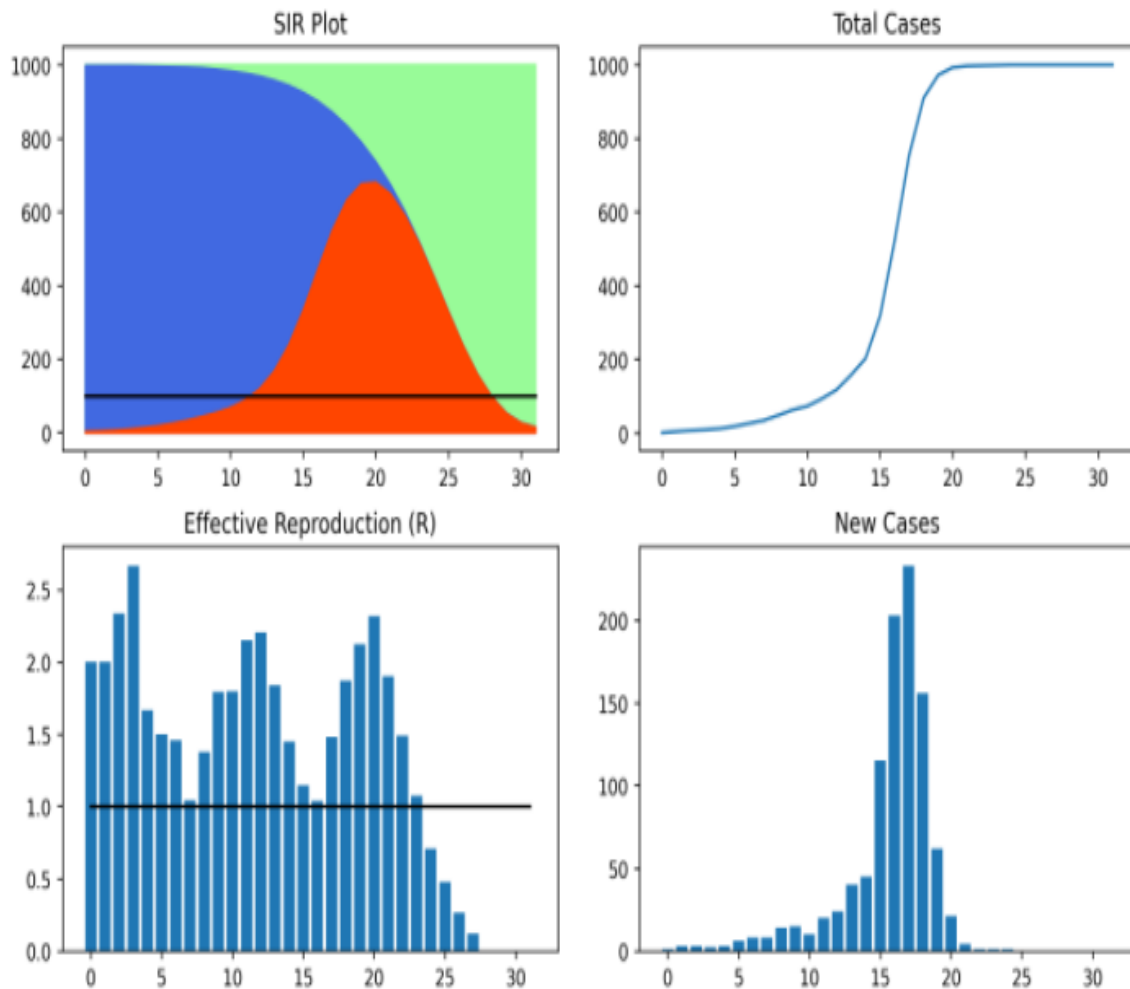


Figure 14: Quarantine probability = 1

On the other hand if quarantine probability is set to 0, it is observed that, the whole population is affected in less duration of time and peak number of cases is also increased.

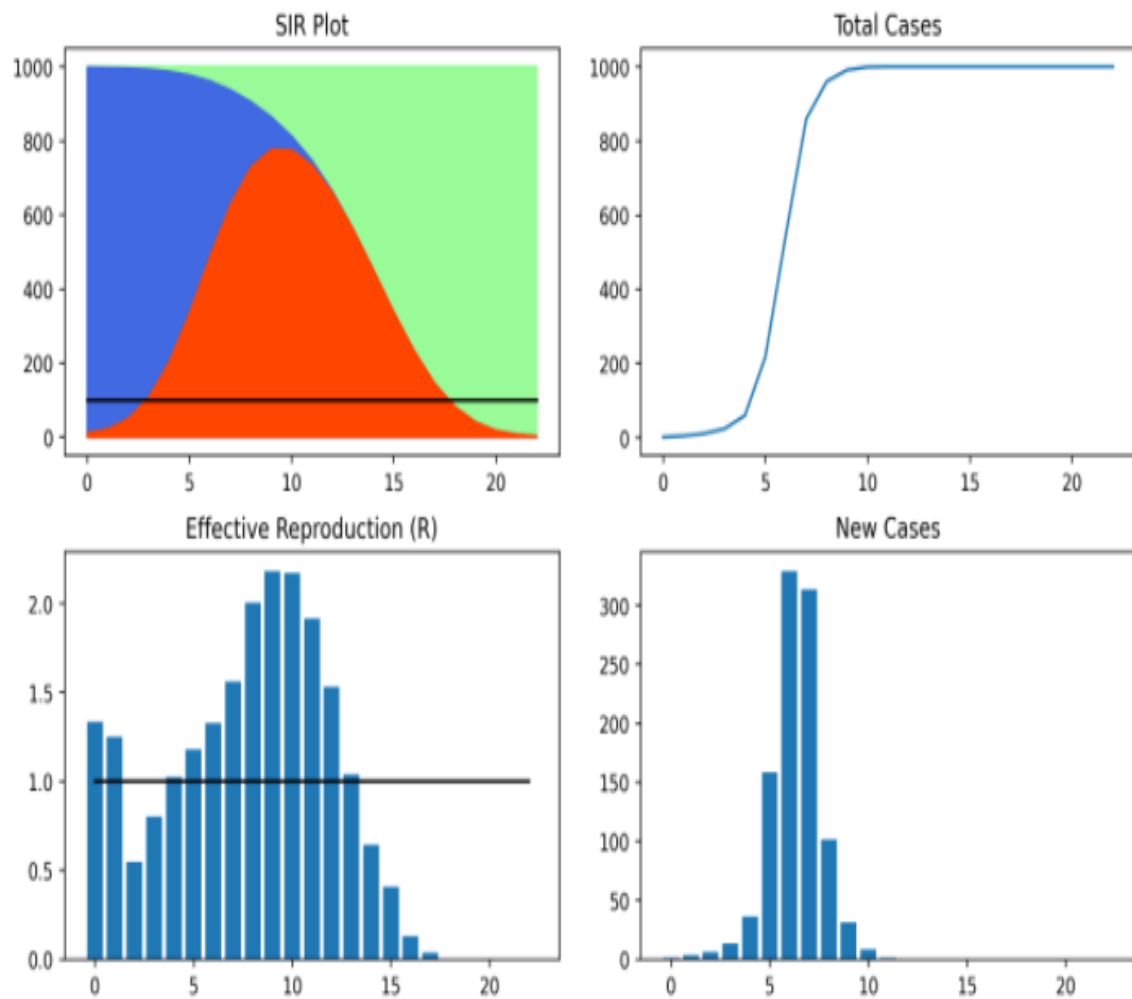


Figure 15: Quarantine probability = 0

As reference we have taken number of available beds (hospital beds) sufficient for 10% of the total population. If it is increased to 80%, it is observed that disease doesn't affect the whole population, but there is no considerable changes in peak, daily new cases etc. This is due to the fact that we have increased the number of hospital beds but the probability of quarantining infected people is still low(0.2). If probability is increased to 0.8, it is observed that a drastic change occurs, the graphs flattens considerably, with less number of daily cases. The disease doesn't affect lot of people and the peak is decreases drastically.

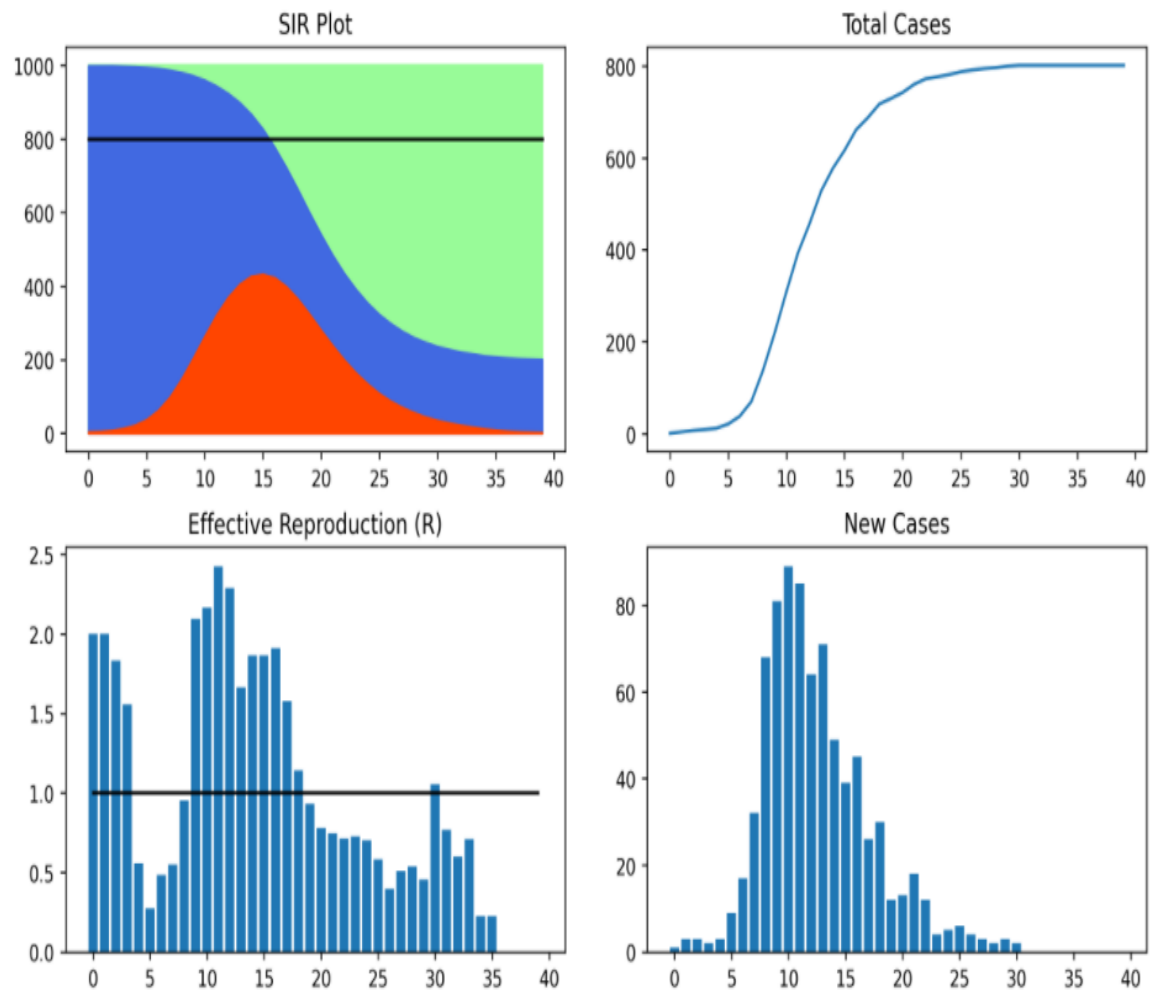


Figure 16: Bed available = 800

Next let's simulate a scenario where the vaccination drive is in full swing causing the immunity to go up, and naturally people are following a less stricter quarantining. No doubt, it is observed that peak cases drops with considerable number of people remaining unaffected, but the disease is prolonged. While it might not look great initially to have the disease prolonged, it also helps make sure most people get the beds they need leading to a lower mortality rate

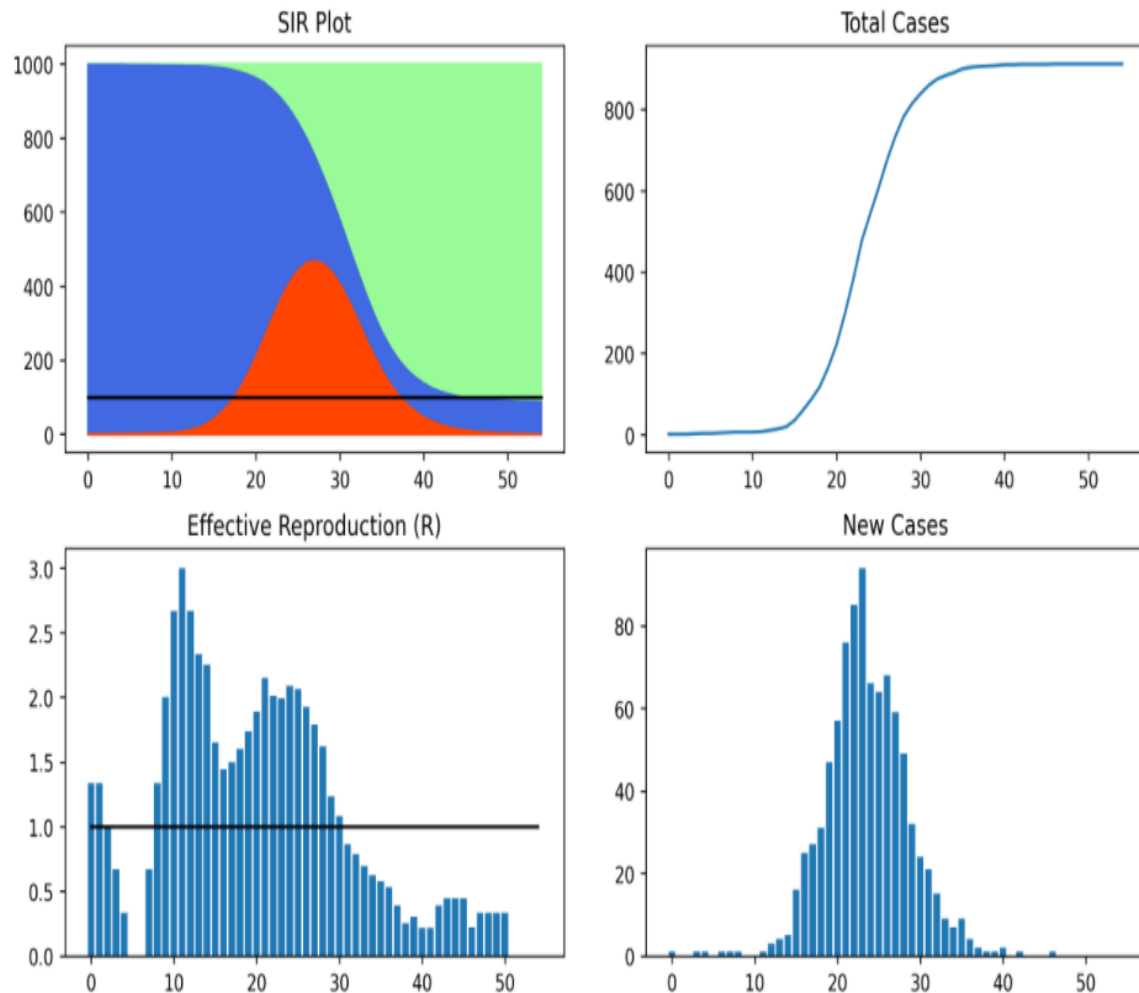


Figure 17: Average immunity = 0.8, Quarantine Strictness = 0.5

For the reference we have taken the adherence to safety measures as 0 i.e. none. Now if we increase it to 0.4, i.e. simulating people wearing masks, sanitizing their hands etc. it is observed that the curve flattens a bit, peak reached is less, and total cases decreases.

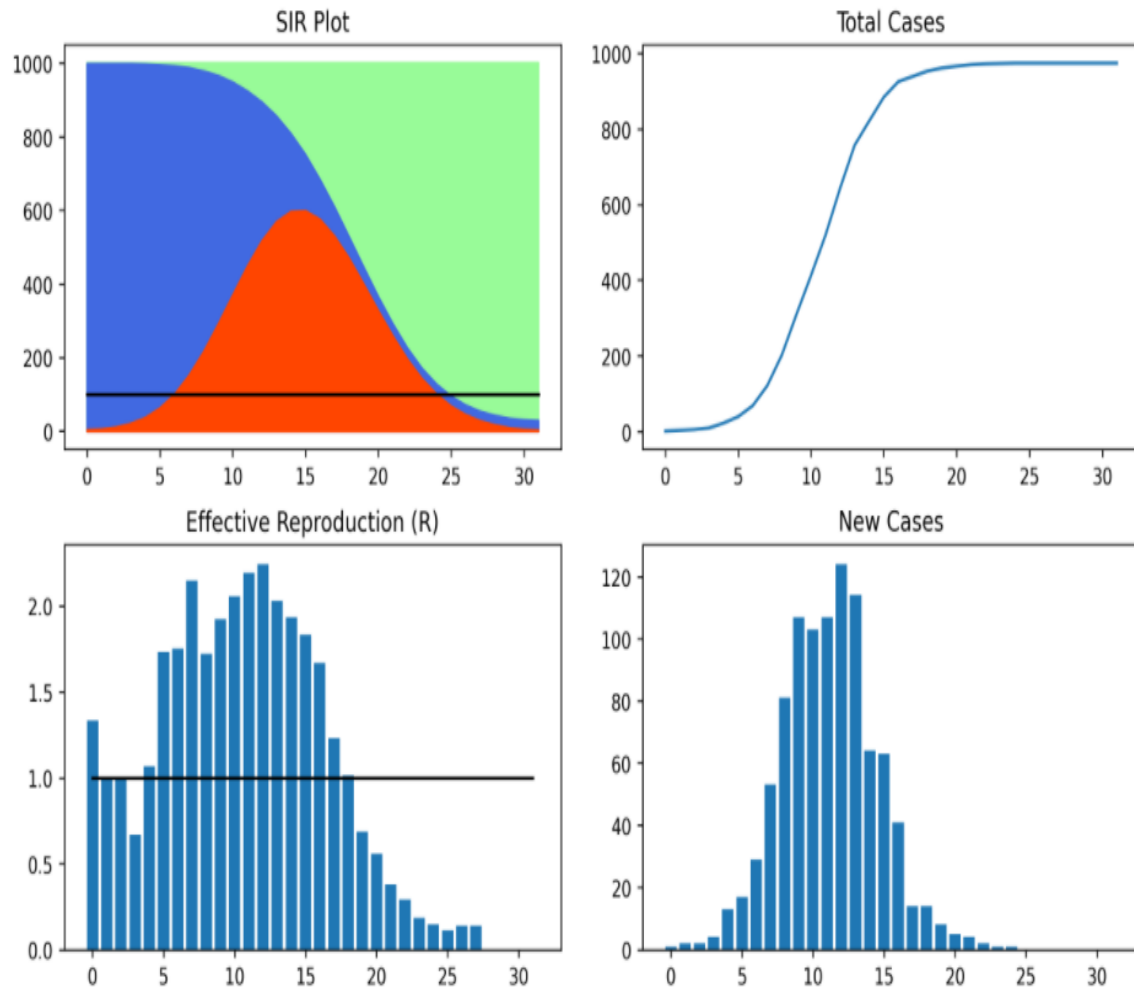


Figure 18: Average adherence to safety measures = 0.4



Now if we further, increase adherence to 0.8 and decrease quarantine strictness to 0.6, the graph gets more flat and there is decrease in peak and total number of overall cases. This goes to show how important following safety measures is!

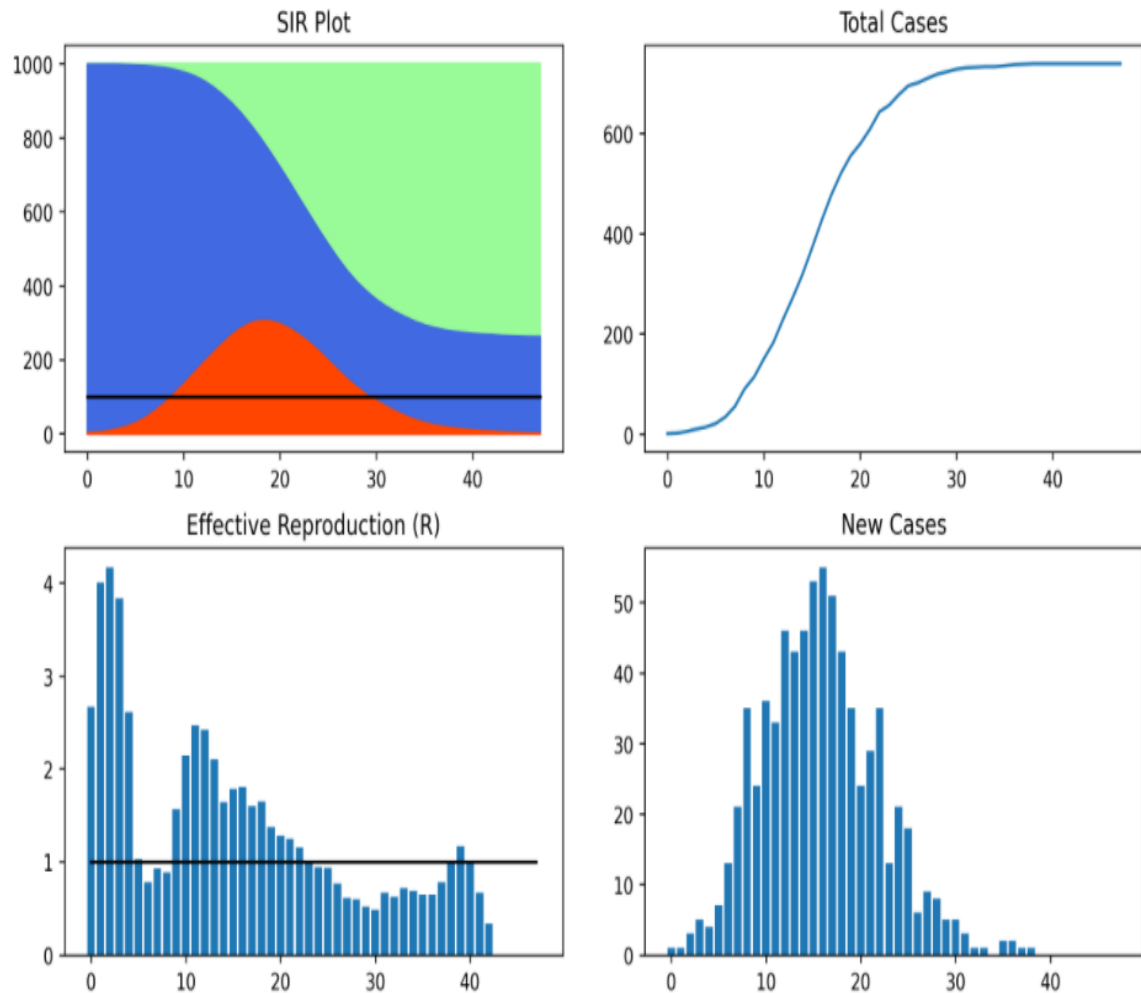


Figure 19: Average adherence to safety measures = 0.8, Quarantine Strictness = 0.6

In the next example we have set the parameters in such a way that a useful observation can be achieved, in case of widespread of the disease and during the peak of the pandemic, emergency field hospitals are set to increase the number of beds to accommodate the high number of active cases (beds available increased to 400, i.e 40% of the population). As we are dealing at the peak of the pandemic, thus quarantine strictness and quarantine probability is also high(0.9).

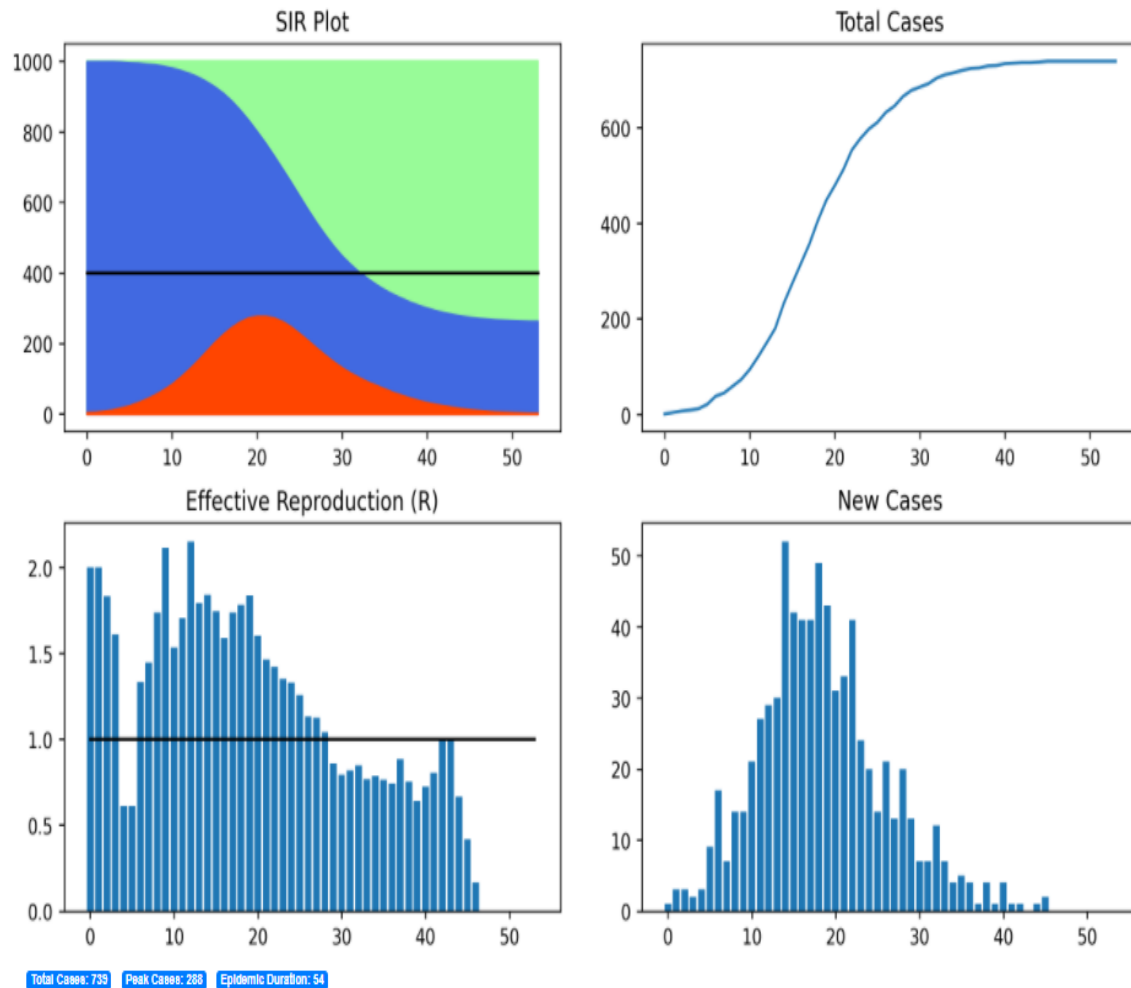


Figure 20: Beds available = 400, Quarantine Strictness = 0.9, Quarantine Probability = 0.9

With these parameters, it is observed that increasing the number of beds available and high strictness and quarantining of people helps into slowing down the spread of the disease, and also it keeps the daily active cases much lower with drastic decrease in peak number of cases.

Now in the coming example it is shown that if quarantine measures are not strictly followed even in case of high quarantine probability and more than enough number of beds available for the population the disease can be widespread and can affect the whole population, to depict this quarantine strictness is set to 0.1, probability to 8 and number of beds available to 800 for a 1000 population. Even though the beds are sufficient for whole population and nearly everyone getting affected is quarantined the disease affect the whole population and peak reached is high.

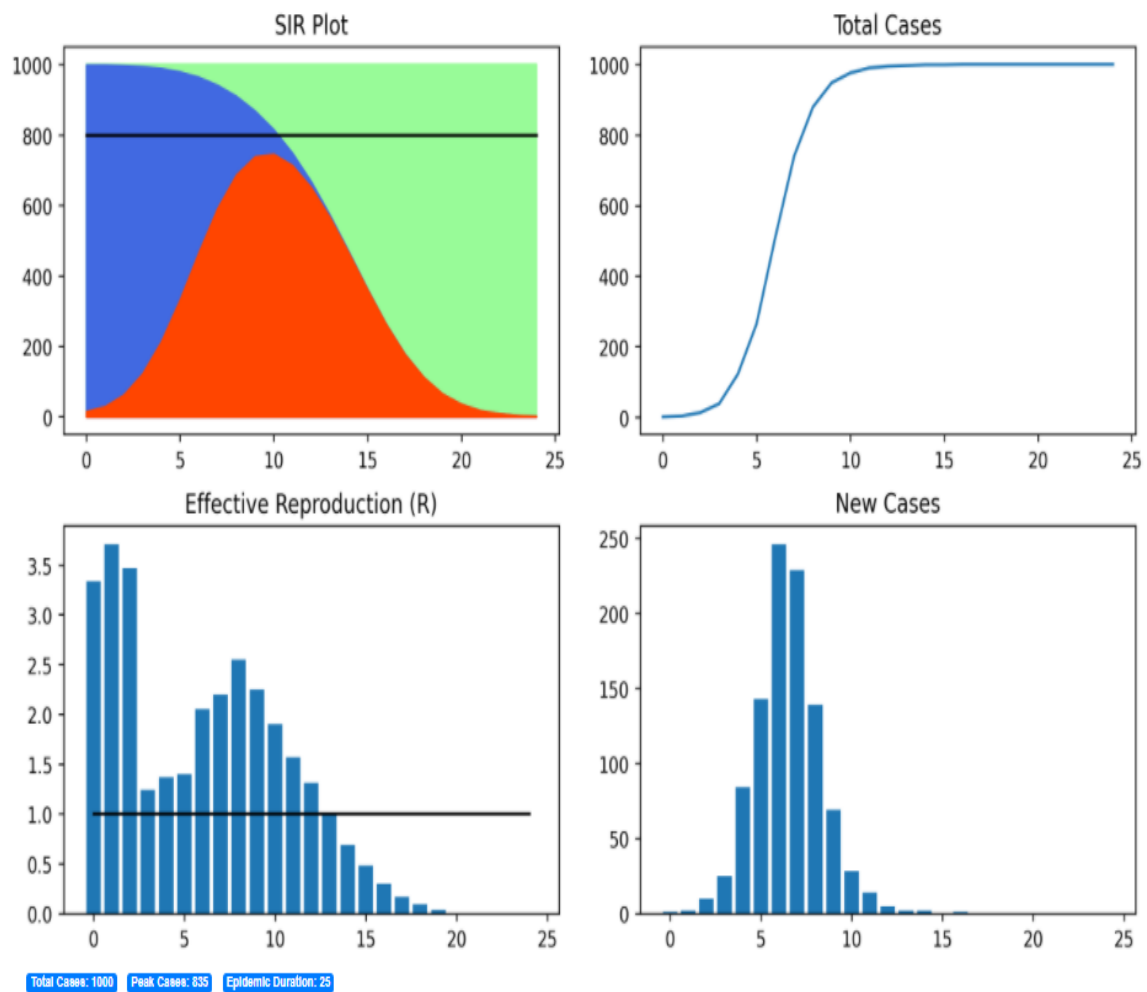


Figure 21: Beds available = 800, Quarantine Strictness = 0.1, Quarantine Probability = 0.8

Average duration of infection - It is the period(in days) that the infection affects the infected. We have taken the base degree for this variable as 9. But as we increase this degree to 25 we see certain changes in our graphs. The total epidemic duration increased to 38, but the peak cases remains nearly the same.

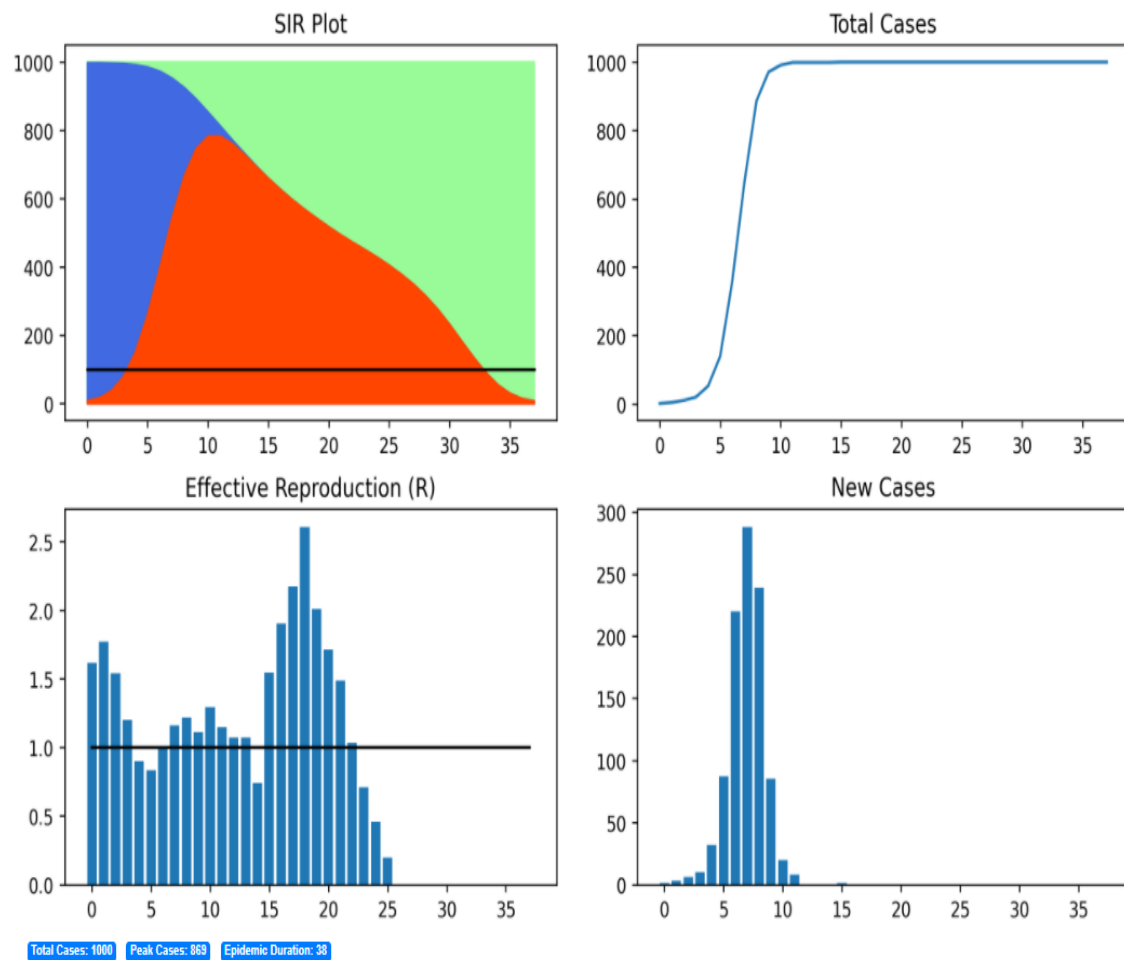


Figure 22: Average duration of infection = 25

This probability is set on a daily basis. We have taken 0.05 as the reference value for this. When we increased it's value to 0.2 it is observed that the total cases falls down to 983 (i.e., 17 people were unaffected) and also a significant decrease in the reproduction rate. This is because the rate of spread will be low when the infected person dies. So the epidemic duration for this increases by 6 more days, as compared to the base graph.

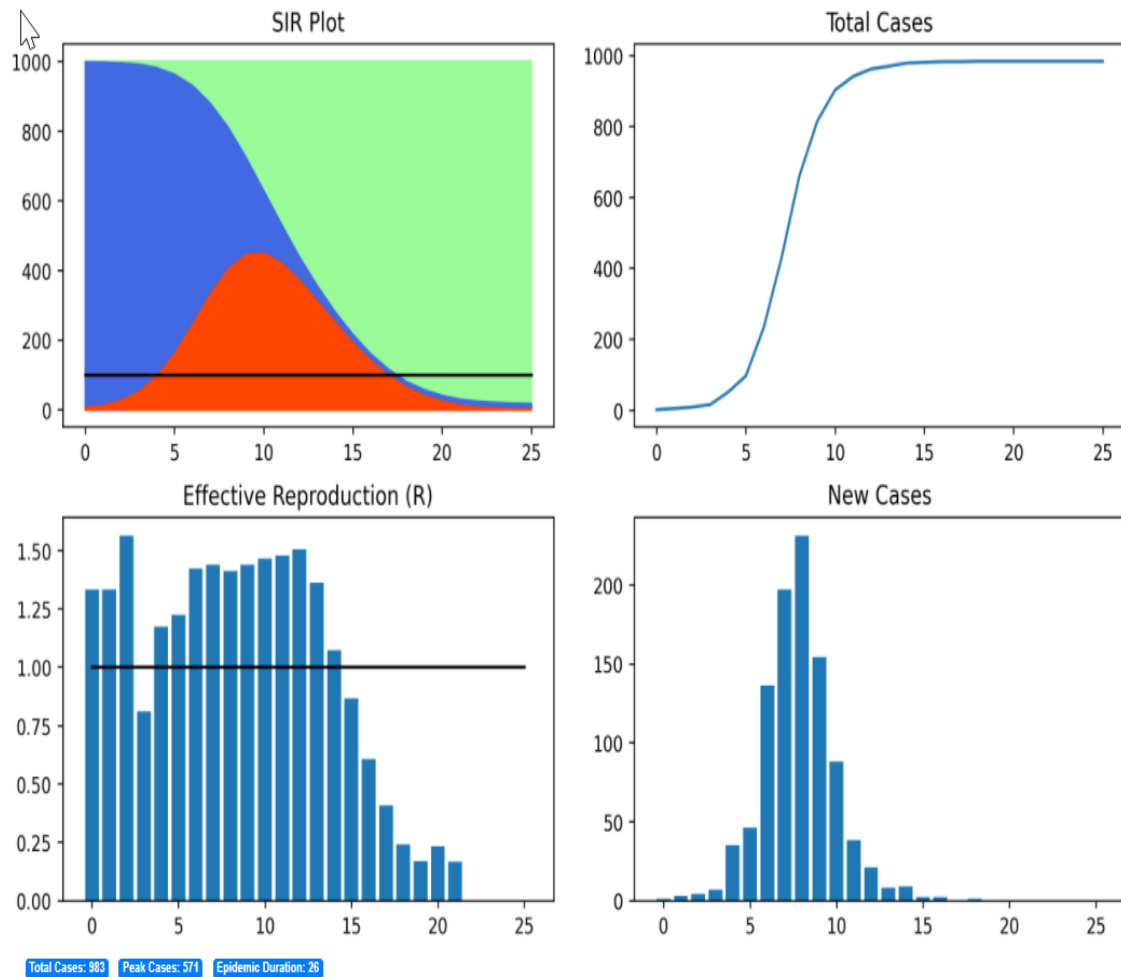


Figure 23: Probability of infected person dying = 0.2

Infection probability of the virus - The following is the probability that the virus infects i.e., spreads from one person to another given all favourable conditions. When we increased the value from 0.05 to 0.35, the total duration falls down to 16 days, but the peak raises to 945. Also the slope of the Total Cases graph increases which represents that the number of nodes infected increases at a faster rate. We can see from the graph that how fast all the population is infected.

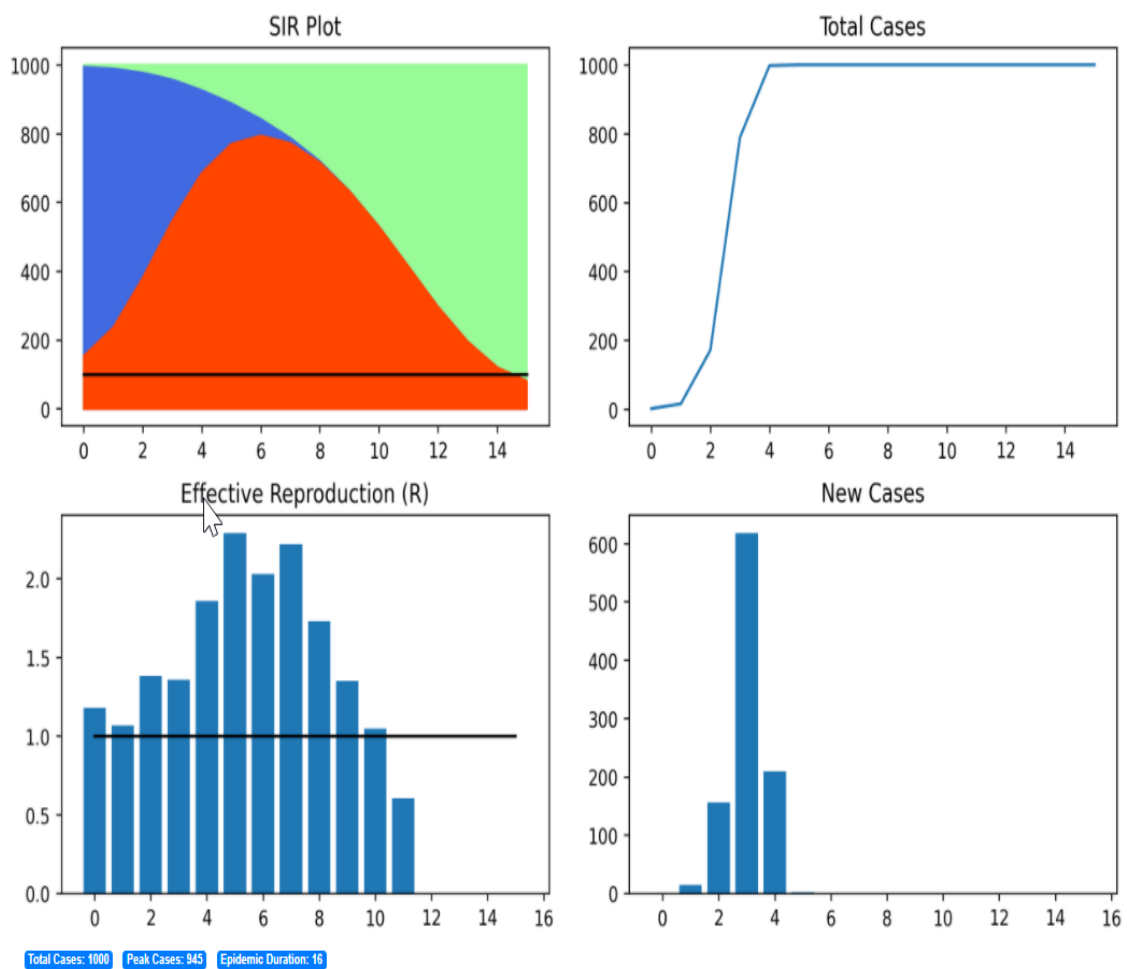


Figure 24: Infection probability of the virus = 0.35

Probability of infection to be asymptomatic - This shows the probable range (0 to 1) of how asymptomatic a infected person is. When we increased it's value from base value 0.1 to 0.6, we didn't see much difference from the baseline except that the Effective reproduction increased a bit at the beginning but decreased eventually. However the duration remained the same and everyone is affected.

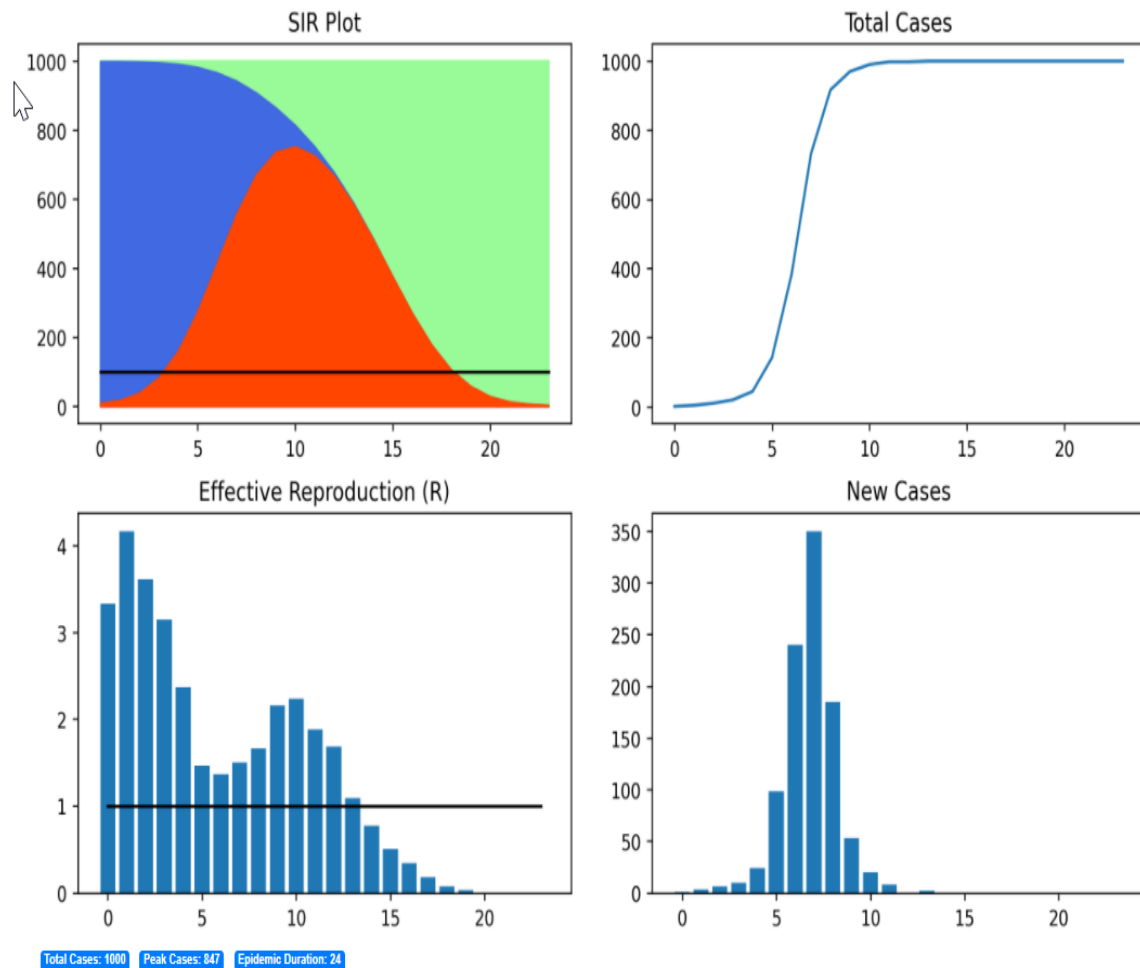


Figure 25: Probability of Infection to be asymptomatic = 0.6

Infection radius of the virus - This is the maximum physical distance(in metres) between two people after which the virus fails to spread. When we changed the radius from 2 to 0.7, the graph gets a little flat and there is decrease in peak and in the total number of overall cases. As we can see in the graph the total cases are just 555 and the peak cases are around 200. Since the infection radius is less, the spreading rate is effectively low in this case. Because of this the duration also increased from 22 days to 68 days.

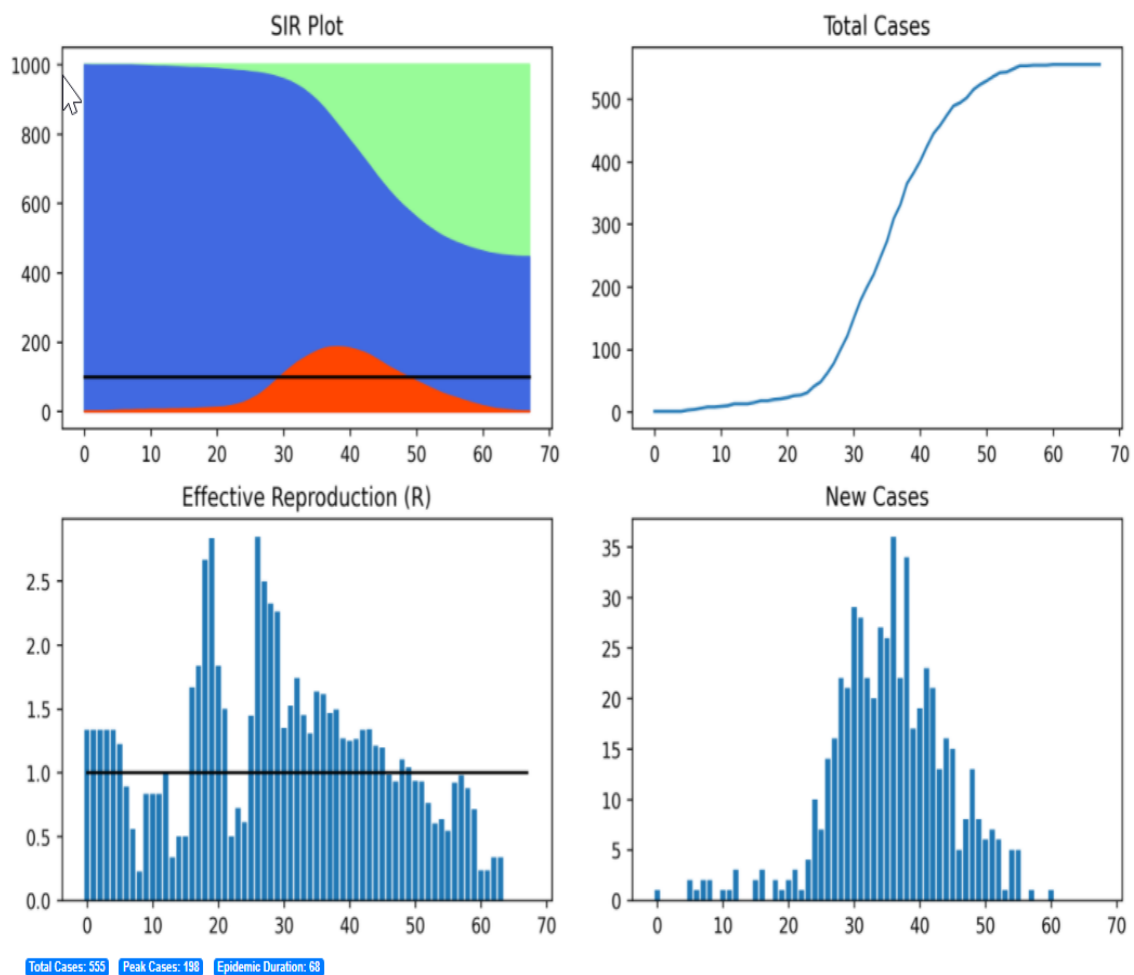


Figure 26: Infection radius of the virus = 0.7



On the other hand when we did the reverse of the above condition i.e., slightly increase in the infection radius from 2 to 5, it is observed that the whole population is affected in less time and peak number of cases also got increased.

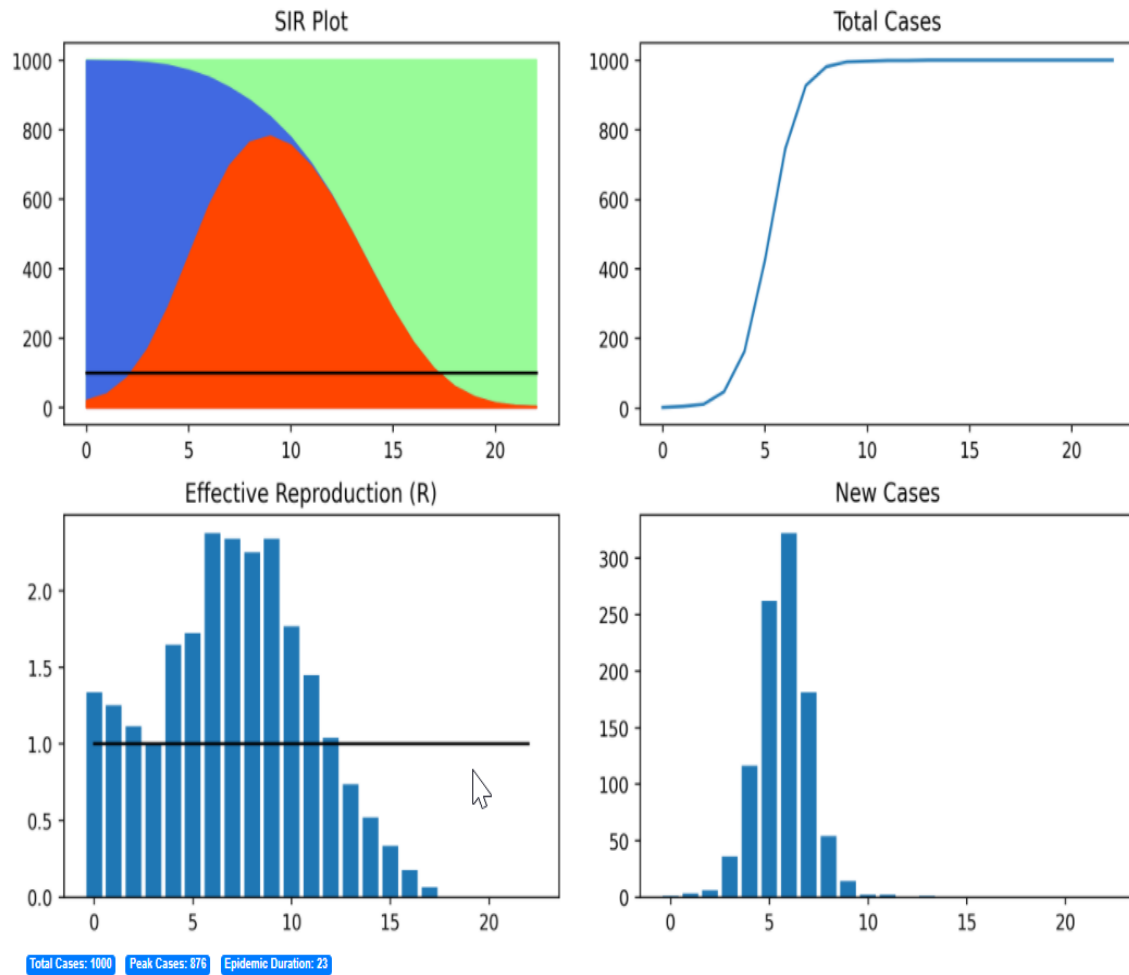


Figure 27: Infection radius of the virus = 5

## 8 Conclusion

Having analysed the simulation results and drawn necessary conclusions, let us now formally wrap everything up. The research started with identifying the problem statement and laying out the requirements. With that done, we then went through the literature available, to gain valuable insights to help us along the way. We also discovered that what we had planned to do, while wasn't unexplored territory, also wasn't done in the way we had planned to. Therefore we finalised the research plan, and began the simulation software development phase. Once that was nearing its end, we started with the testing and analysis phase, and followed it up with the final documentation phase.

This research which had started in the November of 2020, had spanned up to June of 2021. Of course, being busy with college work and jobs, we could find time only during the weekends to do this research. In spite of that, we think we have done a significant work, both in terms of the software development and simulation analysis.

In case one would like continue this research, he / she can find all the documentation including this one, as well as the Python code on GitHub at [github.com/eeshaanachar/epidemic-simulation](https://github.com/eeshaanachar/epidemic-simulation). As a starting point, we suggest remodelling the network structure to include clusters of population and to check the effect of travel parameters, as this was something we couldn't achieve owing to time constraints. From there one can think of newer parameters and / or outputs for the simulation. We also genuinely hope that the research would be carried out in pure interest and not because of another pandemic requiring this research.

## 9 References

### 9.1 Primary

- Sanderson, G. (2020). Simulating an epidemic. Retrieved from <https://www.youtube.com/watch?v=gxAaO2rsdIs>
- Stevens, H. (2020). Why outbreaks like coronavirus spread exponentially, and how to “flatten the curve”. Retrieved from <https://www.washingtonpost.com/graphics/2020/world/corona-simulator>.
- Simler K. (2020). Outbreak. Retrieved from <https://meltingasphalt.com/interactive/outbreak>.
- Iyengar, S. (2019). Social Networks. Retrieved from <https://www.youtube.com/playlist?list=PLyqSpQzTE6M8CLBcLnq-f3vHRH-klC39L>
- Sanderson, G. (2020). Exponential growth and epidemics. Retrieved from <https://www.youtube.com/watch?v=Kas0tIxDvrg>
- Zaplotnik, Z., Gavrić, A., & Medic L. (2020). Simulation of the COVID-19 pandemic on the social network of Slovenia. Retrieved from <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0238090>.

### 9.2 Secondary

- Cherry, J., Krogstad, P. (2004). SARS: The First Pandemic of the 21st Century. Retrieved from <https://doi.org/10.1203/01.PDR.0000129184.87042.FC>
- Ferguson, N., Cummings, D., Cauchemez, S. et al. (2005). Strategies for containing an emerging influenza pandemic in Southeast Asia. Retrieved from <https://doi.org/10.1038/nature04017>
- Jiang, S., Wang, K., Li, C. et al. (2017). Mathematical models for devising the optimal Ebola virus disease eradication. Retrieved from <https://doi.org/10.1186/s12967-017-1224-6>
- Zhao, P. (2020). A Social Network Model of the COVID-19 Pandemic. Retrieved from <https://doi.org/10.1101/2020.03.23.20041798>.
- Bansal, S., Grenfell, B. & Meyers, L. (2007). When individual behaviour matters: homogeneous and network models in epidemiology. Retrieved from <https://doi.org/10.1098/rsif.2007.1100>
- Chao, D., Halloran, E. et al. (2010). FluTE, a Publicly Available Stochastic Influenza Epidemic Simulation Model. Retrieved from <https://doi.org/10.1371/journal.pcbi.1000656>
- Nguyen, D., Deguchi, H. et al. (2010). Agent-based simulation on avian influenza in Vietnam: Basic characteristics of the epidemic and efficiency evaluation of control measures. Retrieved from <https://ieeexplore.ieee.org/abstract/document/5530215>

- Tuite, A., Eisenberg, M. et al. (2011). Cholera Epidemic in Haiti, 2010: Using a Transmission Model to Explain Spatial Spread of Disease and Identify Optimal Control Interventions. Retrieved from [www.researchgate.net/publication/50304455](http://www.researchgate.net/publication/50304455)
- Nassar, M., Meo, S. et al. (2018). Global seasonal occurrence of Middle East Respiratory Syndrome Coronavirus (MERS-CoV) infection. Retrieved from [www.researchgate.net/publication/326041885](http://www.researchgate.net/publication/326041885)
- Hossain, M. et al. (2020). The effects of border control and quarantine measures on the spread of COVID-19. Retrieved from [www.sciencedirect.com/science/article/pii/S1755436520300244](http://www.sciencedirect.com/science/article/pii/S1755436520300244)

## **A Research Team**

**Eeshaan Achar**



**Anuj Yadav**



**Saurav Kumar**



CS Scanned with CamScanner

**Manjunath Badakar**



## **B COs, POs and PSOs mapping for the Project work (CS84P)**

### **B.1 Course Outcomes**

**CO1:** Formulate the problem definition, conduct literature review and apply requirements analysis.

**CO2:** Develop and implement algorithms for solving the problem formulated.

**CO3:** Comprehend, present and defend the results of exhaustive testing and explain the major findings.

### **B.2 Program Outcomes**

**PO1:** Apply knowledge of computing, mathematics, science, and foundational engineering concepts to solve the computer engineering problems.

**PO2:** Identify, formulate and analyze complex engineering problems.

**PO3:** Plan, implement and evaluate a computer-based system to meet desired societal needs such as economic, environmental, political, healthcare and safety within realistic constraints.

**PO4:** Incorporate research methods to design and conduct experiments to investigate real-time problems, to analyze, interpret and provide feasible conclusion.

**PO5:** Propose innovative ideas and solutions using modern tools.

**PO6:** Apply computing knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

**PO7:** Analyze the local and global impact of computing on individuals and organizations for sustainable development.

**PO8:** Adopt ethical principles and uphold the responsibilities and norms of computer engineering practice.

**PO9:** Work effectively as an individual and as a member or leader in diverse teams and in multidisciplinary domains.

**PO10:** Effectively communicate and comprehend.

**PO11:** Demonstrate and apply engineering knowledge and management principles to manage projects in multidisciplinary environments.

**PO12:** Recognize contemporary issues and adapt to technological changes for life-long learning.

### **B.3 Program Specific Outcomes**

**PSO1:** Problem Solving Skills: Ability to apply standard practices and mathematical methodologies to solve computational tasks, model real world problems in the areas of database systems, system software, web technologies and Networking solutions with an appropriate knowledge of Data structures and Algorithms.

**PSO2:** Knowledge of Computer Systems: An understanding of the structure and work- ing of the computer systems with performance study of various computing architectures.


**PSO3:** Successful Career and Entrepreneurship: The ability to get acquaintance with the state of the art software technologies leading to entrepreneurship and higher studies.

**PSO4:** Computing and Research Ability : Ability to use knowledge in various domains to identify research gaps and to provide solution to new ideas leading to innovations.

## C Publication Details

We have consolidated the work done in this research into a research paper, and have submitted it to “THE 28TH ASIA-PACIFIC SOFTWARE ENGINEERING CONFERENCE” (<https://apsec2021.seat.org.tw/>) that was originally planned to take place in Taipei, Taiwan, but now is being held virtually. This conference is an international conference and in their own words, “The Asia-Pacific Software Engineering Conference (APSEC) has long been established as a premier regional conference that brings together researchers and practitioners from academia, industry and government to share the state of the art and the practice of software engineering, and to explore emerging challenges and solutions in software engineering innovation.”


Unfortunately though, the conference dates are 6th to 9th of December, 2021. In spite of that, we decided to go with the said conference as it is an international one, is open to a wide variety of research work in the field of Computer Science and Software Engineering, and importantly is legit unlike most of the paid conferences that promise to get the paper published in a few hours / days. We believe that our work is worthy enough of getting the recognition, if any, through this conference.

 **APSEC 2021 (author)**

[New Submission](#) [Submission 82](#) [APSEC 2021](#) [Conference](#) [News](#) [EasyChair](#)

**APSEC 2021 Submission 82**

**The submission has been saved!**

Submission 82	
Title:	Epidemic Simulation
Paper:	 (Jul 07, 05:45 GMT)
Track:	Technical Research Papers
Author keywords:	academic-research research simulation software-development verification
Topics:	Empirical Software Engineering, Open Source Development, Testing, Verification, and Validation
Abstract:	This research paper analyses the results obtained from the simulation of the spread of epidemics over different types of population. This analysis should help increase awareness on the various measures that a society can take to minimise and mitigate the effects of a pandemic. The simulation was done using a software that was built with Python. While not all parameters supported by the simulation software are discussed in this paper, the ones that showed the most interesting results have been picked.
Submitted:	Jul 07, 05:45 GMT
Last update:	Jul 07, 05:45 GMT

Authors						
first name	last name	email	country	affiliation	Web page	corresponding?
Eeshaan	Achar	eeshaan77@gmail.com	India	JSSSTU		✓
Anuj	Yadav	yadavanuj952@gmail.com	India	JSSSTU		✓
Saurav	Kumar	saurav23999ashu@gmail.com	India	JSSSTU		✓
Manjunath	Badakar	manjubadakar@gmail.com	India	JSSSTU		✓
Vijayalakshmi	H C	vijilakshmi@sjce.ac.in	India	JSSSTU		

Figure 28: Proof of Submission