Chapter 1
Introduction

Identifying areas with low access to testing and high case burden is necessary to understand risk and allocate resources in the COVID-19 pandemic. Using zip code level data for India, we analyzed testing rates, positivity rates, and proportion positive. A spatial scan statistic identified clusters of high and low testing rates, high positivity rates, and high proportion positive. Boxplots and Pearson correlations determined associations between outcomes, clusters, and contextual factors. Clusters with less testing and low proportion positive tests had higher income, education, and white population, whereas clusters with high testing rates and high proportion positive tests were disproportionately black and without health insurance. Correlations showed inverse associations of white race, education, and income with proportion positive tests, and positive associations with black race, Hispanic ethnicity, and poverty. We recommend testing and health care resources be directed to eastern Brooklyn, which has low testing and high proportion positives.

1.1 COVID-19

Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It was first identified in December 2019 in Wuhan, China, and has resulted in an ongoing pandemic. The first case may be traced back to 17 November 2019. As of 8 June 2020, more than 6.98 million cases have been reported across 188 countries and territories, resulting in more than 401,000 deaths. More than 3.13 million people have recovered.

The virus is primarily spread between people during close contact, most often via small droplets produced by coughing, sneezing, and talking. The droplets usually fall to the ground or onto surfaces rather than travelling through air over long distances. Less commonly, people may become infected by touching a contaminated surface and then touching their face. It is most contagious during the first three days after the onset of symptoms, although spread is possible before symptoms appear, and from people who do not show symptoms.

1.2 Pandemic

The COVID-19 pandemic, also known as the coronavirus pandemic, is an ongoing pandemic of coronavirus disease 2019 (COVID-19) [1], caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The outbreak was first identified in Wuhan, China, in December 2019. The World Health Organization declared the outbreak a Public Health Emergency of International Concern on 30 January, and a pandemic on 11 March.

Coronaviruses are important human and animal pathogens. At the end of 2019, a novel coronavirus was identified as the cause of a cluster of pneumonia cases in Wuhan, a city in the Hubei Province of China. It rapidly spread, resulting in an epidemic throughout China, followed by an increasing number of cases in other countries throughout the world. On 30th January 2020 India recorded its first COVID-19 case in state of Kerala. It was a student who had travel history to China. And till the start of June India has over 200 thousand confirmed cases.

1.3 Problem statement

In this project we have dived deep into ‘What does data say about Covid-19 situation in India?’.

Chapter 2
LITERATURE SURVEY

The literature review took place during the last weeks of May 2020 and focused mainly on the Web of Science (WOS) database, using Scopus and Dial net as support. The topic considered for the selection of articles was the one related to the global pandemic caused by COVID-19 and how it has affected psychologically and motorically children up to 12 years old. The following keywords were used: “COVID-19” and “children” and the Boolean operator “and.” After this first search and considering only the works published in 2020 (since that is when the pandemic occurred), 837 scientific documents were obtained. By restricting the search to only journal articles, the documents were reduced to 576 articles, after which the language filter was applied, selecting only those papers published in English and Spanish, leaving a total of 537. Since the pandemic started in China, the initial search was also done in that language, not finding any related articles. The articles signed by researchers of Chinese nationality are written in English. Finally, the following areas of research were chosen: “Psychology,” “Sociology,” and “Education Educational Research,” finally limiting the search to 48 scientific articles, which make up the sample of this study.

2.1 Algorithms

Research on data mining has led to the formulation of several data mining algorithms. These algorithms can be directly used on a dataset for creating some models or to draw vital conclusions and inferences from that dataset. Some popular data mining algorithms are Decision tree, Naïve Bayes, k-means, artificial neural network etc. [4]

2.1.1 Decision Tree:

A Decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences including chance event outcomes and utility. It is one of the ways to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis to help and identify a strategy that will most likely reach the goal. It is also a popular tool in machine learning.

2.1.2 K-means Algorithm:

K-means creates k groups from a set of given objects so that the members of a group are more similar. Other than specifying the number of clusters, k-means also "learns" the clusters on its own without any information about which cluster a particular observation should belong to. That's why k-means can be called as semi-supervised learning method. K-means is especially effective over large datasets.

2.1.3 The ID3 algorithm (Quinlan86):

It is a Decision tree building algorithm which determines the classification of objects by testing the values of the properties. It builds the tree in a top-down fashion, starting from a set of objects and the specification of properties. At each node of the tree, a property is tested, and the results used to partition the object at that point are set. This process is recursively continued till the set-in a given sub tree is homogeneous with respect to the classification criteria. Then it becomes a leaf node. At each node, information gain is maximized, and entropy is minimized. In simpler words, that property is tested which divides the candidate set in the most homogeneous subsets [5].

2.1.4 Support Vector Machine (SVM):

It is a supervised learning method which classifies data into two classes over a hyper plane. Support vector machine performs a similar task like C4.5 except that it doesn't use Decision trees at all. Support vector machine attempts to maximize the margin (distance between the hyper plane and the two closest data points from each respective class) to decrease any chance of misclassification. Some popular implementations of support vector machine are scikit-learn, MATLAB and of LIBSVM.

Regression is a statistical concept which is used to determine the weight of relationship between one dependent variable (usually denoted by Y) and a series of other changing variables (known as independent variables). Two basic types of regression are linear regression and multiple linear regression. Also, there are several non-linear regression methods that are used for more complicated data analysis [6]

2.2 Python Libraries:

2.2.1 Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license

2.2.2 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.

2.2.3 Plotly:

Plotly provides **online graphing, analytics, and statistics tools for individuals and collaboration**, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.

2.2.4 Cufflinks:

Cufflinks is **another library that connects the Pandas data frame** with Plotly enabling users to create visualizations directly from Pandas. The library binds the power of Plotly with the flexibility of Pandas for easy plotting.

2.2.5 Follium:

Folium is **a Python library** that makes it possible visualize data on an interactive Leaflet map. Resources: Folium Documentation.

Chapter 3

System Requirement Specification

A System Requirements Specification (SRS)) [6] is a document or set of documentation that describes the features and behavior of a system or software application. It includes a variety of elements that attempts to define the intended functionality required.

3.1 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system or its component. Where a function is described as a specification of behavior between inputs and outputs. There are a lot of software requirements specifications included in the functional requirements of the Hospital Management System, which contains various process, namely Registration, Check out, Report Generation, and Database.

3.2 NON-FUNCTIONAL REQUIREMENTS

Software requirement can be non-functional and be a performance requirement. Non-functional requirements are the characteristics or attributes of the system that can judge its operation. There are a lot of software requirements specifications included in the non-functional requirements of the Hospital Management System, which contains various process, namely Security, Performance, Maintainability, and Reliability.

CHAPTER 4

Design

Project design is an early phase of the project where a project's key features, structure, criteria for success, and major deliverables are all planned out. The aim is to develop one or more designs that can be used to achieve the desired project goals.[5]

4.1 Process Logic

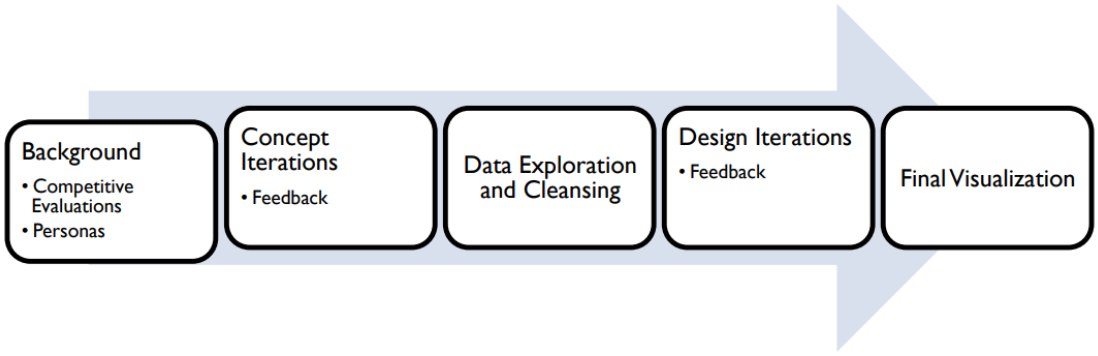

Fig 4.1 Process logic

process provides a framework to extract [8] nontrivial information from data. With the advent of massive storage, increased data collection, and advanced computing paradigms, the data at our disposal are only increasing.

4.2 SPECTRUM OF COVID-19 CASES

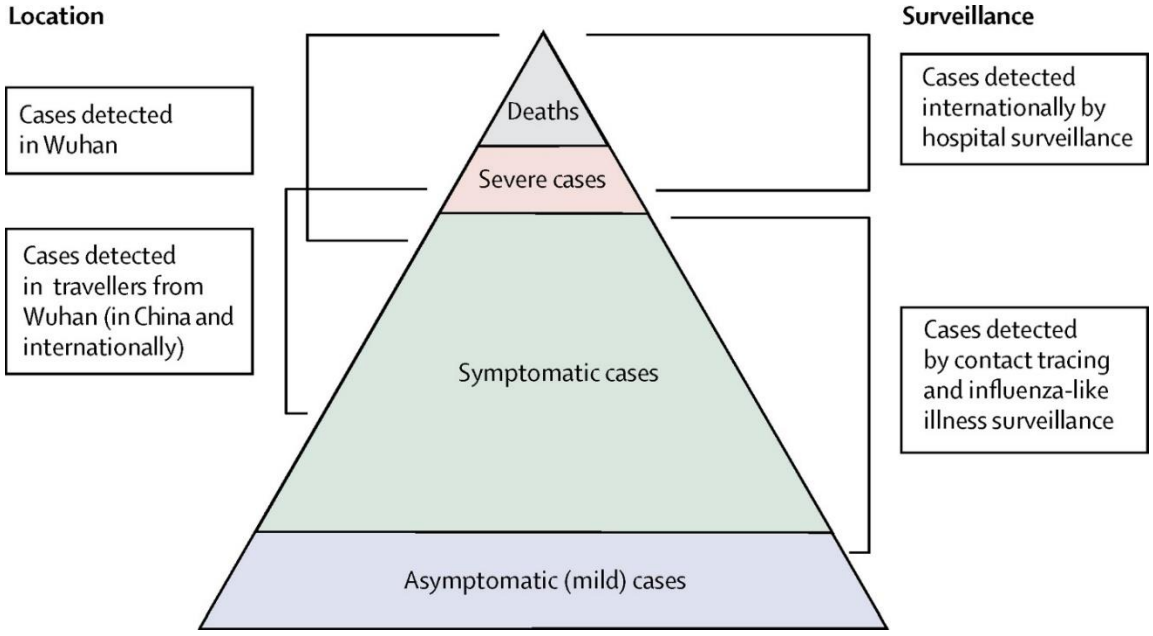

Fig 4.2 Spectrum of Covid-19 cases

Patients with certain underlying comorbidities are at a higher risk of progressing to severe COVID-19. These comorbidities include being aged ≥65 years; having cardiovascular disease, chronic lung disease, sickle cell disease, diabetes, cancer, obesity, or chronic kidney disease; being pregnant; being a cigarette smoker; being a transplant recipient; and receiving immunosuppressive therapy.

4.3 Exploratory data analysis

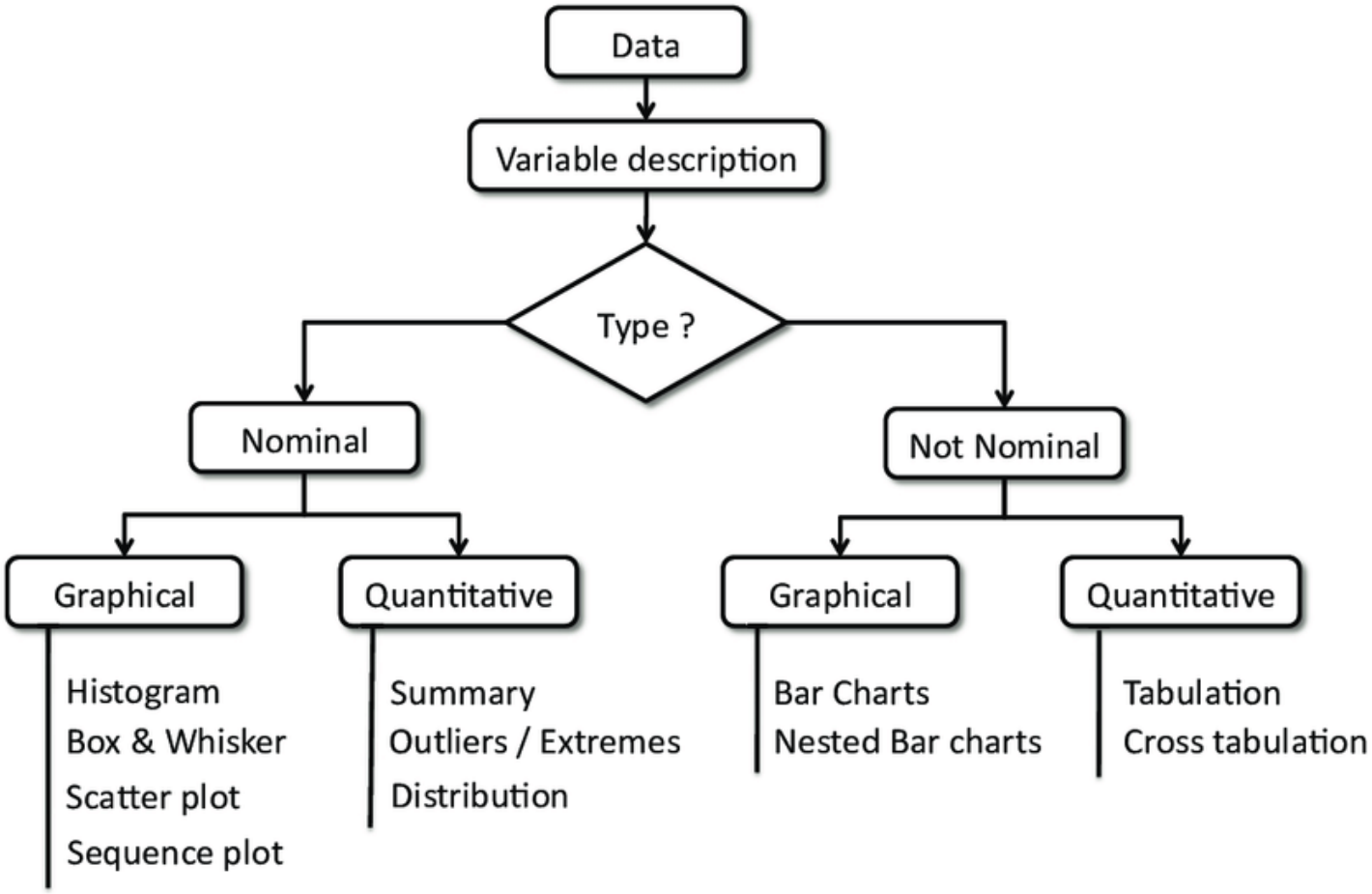

Fig 4.3 Exploratory data analysis

Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or it check assumptions with the help of statistical summary and graphical representations.

CHAPTER 5
Implementation

The Implementation phase of the Project Management Process puts the project into action. The Implementation phase consist of four sub phases: Execution, Monitoring & Control, and Move to Production.[5]

5.1 Source code

#Importing Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
%matplotlib inline

import plotly
import plotly.express as px
import plotly.graph_objects as go
import cufflinks as cf
import plotly.offline as pyo
from plotly.offline import init_notebook_mode,plot,iplot

import folium
pyo.init_notebook_mode(connected=True)
cf.go_offline()
```

#Accessing the Dataset

```
df=pd.read_excel(r"Covid cases in India.xlsx")
df
df['Total Cases']=df['Total Confirmed cases (Indian National)']+df['Total Confirmed cases (
```

```
    Foreign National )']
df
total_cases_overall=df['Total Cases'].sum()
print('The total number of cases till now in India is ',total_cases_overall)
df['Active Cases']=df['Total Cases']-(df['Death']+df['Cured'])
df
Total_Active_Cases=df.groupby('Name of State / UT')['Total
    Cases'].sum().sort_values(ascending=False).to_frame()
Total_Active_Cases
Total_Active_Cases.style.background_gradient(cmap='Reds')
```

################## **Graphical Represantaion**###########

#Pandas vis

```
df.plot(kind='bar',x='Name of State / UT',y='Total Cases')
plt.show()
```

#Plotly

```
df.iplot(kind='bar',x='Name of State / UT',y='Total Cases')
```

#Matplotlib vis

```
plt.figure(figsize=(15,4))
plt.bar(df['Name of State / UT'],df['Total Cases'])
plt.xticks(rotation=90)
plt.show()
px.bar(df,x='Name of State / UT',y='Total Cases')
plt.figure(figsize=(15,4))
df.plot(kind='scatter',x='Name of State / UT',y='Total Cases')
plt.xticks(rotation=90)
plt.show()
plt.scatter(df['Name of State / UT'],df['Total Cases'])
plt.xticks(rotation=90)
plt.show()
df.iplot(kind='scatter',x='Name of State / UT',y='Total Cases',mode='markers+lines',title='My
    Graph',xTitle='Name of State / UT',yTitle='Total Cases',colors='red',size=20)
```

```
px.scatter(df,x='Name of State / UT',y='Total Cases')
```

 # Total Cases using Matplotlib

```
fig=plt.figure(figsize=(20,10),dpi=200)
axes=fig.add_axes([0,0,1,1])
axes.bar(df['Name of State / UT'],df['Total Cases'])
axes.set_title("Total Cases in India")
axes.set_xlabel("Name of State / UT")
axes.set_ylabel("Total Cases")
plt.xticks(rotation=90)
plt.show()
```

#plotly

```
fig=go.Figure()
fig.add_trace(go.Bar(x=df['Name of State / UT'],y=df['Total Cases']))
fig.update_layout(title='Total Cases in India',xaxis=dict(title='Name of State / UT'),yaxis=dict(title='Total Cases'))
Indian_Cord=pd.read_excel(r"Indian Coordinates.xlsx")
Indian_Cord
df_full=pd.merge(Indian_Cord,df,on='Name of State / UT')
map=folium.Map(location=[20,70],zoom_start=4,tiles='Stamenterrain')
for lat,long,value, name in zip(df_full['Latitude'],df_full['Longitude'],df_full['Total Cases'],df_full['Name of State / UT']):
folium.CircleMarker([lat,long],radius=value*0.8,popup=('<strong>State</strong>: '+str(name).capitalize()+'<br>"<strong>Total Cases</strong>: ' + str(value)+ '<br>'),color='red',fill_color='red',fill_opacity=0.3).add_to(map)
map
```

 # How corona virus is rising Globally

```
dbd_India=pd.read_excel(r"per_day_cases.xlsx",parse_dates=True,sheet_name="India")
dbd_Italy=pd.read_excel(r"per_day_cases.xlsx",parse_dates=True,sheet_name="Italy")
dbd_Korea=pd.read_excel(r"per_day_cases.xlsx",parse_dates=True,sheet_name="Korea")
```

```
dbd_Wuhan=pd.read_excel(r"per_day_cases.xlsx",parse_dates=True,sheet_name="Wuhan")
dbd_India
```

#Matplotlib

```
fig=plt.figure(figsize=(10,5),dpi=200)
axes=fig.add_axes([0.1,0.1,0.8,0.8])

axes.bar(dbd_India["Date"],dbd_India["Total Cases"],color='blue')
axes.set_xlabel("Date")
axes.set_ylabel("Total Cases")

axes.set_title("Confirmed cases in India")
plt.show()
```

#plotly Express

```
fig=px.bar(dbd_India,x="Date",y="Total Cases",color='Total Cases',title='Confirmed cases in
    India')
fig.show()
fig=px.bar(dbd_Italy,x="Date",y="Total Cases",color='Total Cases',title='Confirmed cases in
    Italy')
fig.show()

fig=px.bar(dbd_Korea,x="Date",y="Total Cases",color='Total Cases',title='Confirmed cases
    in Korea')
fig.show()

fig=px.bar(dbd_Wuhan,x="Date",y="Total Cases",color='Total Cases',title='Confirmed cases
    in Wuhan')
fig.show()
fig=plt.figure(figsize=(10,5),dpi=200)
axes=fig.add_axes([0.1,0.1,0.8,0.8])
```

```
axes.plot(dbd_India["Date"],dbd_India["Total Cases"],color='blue',marker='*')
axes.set_xlabel("Date")
axes.set_ylabel("Total Cases")
axes.set_title("Confirmed cases in India")
plt.show()
```

#plotly Express

```
fig=px.scatter(dbd_India,x="Date",y="Total Cases",color='Total Cases',title='Confirmed
    cases in India')
fig.show()
```

#Plotly

```
dbd_India.iplot(kind='scatter',x='Date',y='Total Cases',mode='lines+markers')
fig=go.Figure()
fig.add_trace(go.Scatter(x=dbd_India['Date'],y=dbd_India['Total
    Cases'],mode='lines+markers'))
#Subplots using Bar Graph
from plotly.subplots import make_subplots
fig=make_subplots(
rows=2,cols=2,
specs=[[{"secondary_y":True},{"secondary_y":True}],[{"secondary_y":True},{"secondary_
    y":True}]],
subplot_titles=("S.Korea","Italy","India","Wuhan"))
```

```
fig.add_trace(go.Bar(x=dbd_Korea['Date'],y=dbd_Korea['Total Cases'],
marker=dict(color=dbd_Korea['Total Cases'],coloraxis="coloraxis")),1,1)

fig.add_trace(go.Bar(x=dbd_Italy['Date'],y=dbd_Italy['Total Cases'],
marker=dict(color=dbd_Italy['Total Cases'],coloraxis="coloraxis")),1,2)
```

```
fig.add_trace(go.Bar(x=dbd_India['Date'],y=dbd_India['Total Cases'],
marker=dict(color=dbd_India['Total Cases'],coloraxis="coloraxis")),2,1)

fig.add_trace(go.Bar(x=dbd_Wuhan['Date'],y=dbd_Wuhan['Total Cases'],
marker=dict(color=dbd_Wuhan['Total Cases'],coloraxis="coloraxis")),2,2)



fig.update_layout(coloraxis=dict(colorscale='Bluered_r'),showlegend=False,title_text="Total
    Cases in 4 Countries")

fig.update_layout(plot_bgcolor='rgb(230,230,230)')
fig=make_subplots(
rows=2,cols=2,
specs=[[{"secondary_y":True},{"secondary_y":True}],[{"secondary_y":True},{"secondary_
    y":True}]],
subplot_titles=("S.Korea","Italy","India","Wuhan"))

fig.add_trace(go.Scatter(x=dbd_Korea['Date'],y=dbd_Korea['Total Cases'],
marker=dict(color=dbd_Korea['Total Cases'],coloraxis="coloraxis")),1,1)

fig.add_trace(go.Scatter(x=dbd_Italy['Date'],y=dbd_Italy['Total Cases'],
marker=dict(color=dbd_Italy['Total Cases'],coloraxis="coloraxis")),1,2)

fig.add_trace(go.Scatter(x=dbd_India['Date'],y=dbd_India['Total Cases'],
marker=dict(color=dbd_India['Total Cases'],coloraxis="coloraxis")),2,1)

fig.add_trace(go.Scatter(x=dbd_Wuhan['Date'],y=dbd_Wuhan['Total Cases'],
marker=dict(color=dbd_Wuhan['Total Cases'],coloraxis="coloraxis")),2,2)



fig.update_layout(coloraxis=dict(colorscale='Bluered_r'),showlegend=False,title_text="Total
    Cases in 4 Countries")
```

```
fig.update_layout(plot_bgcolor='rgb(230,230,230)')
```

#World Coronavirus

```
df=pd.read_csv(r'covid_19_data.csv',parse_dates=['Last Update'])
df.rename(columns={'ObservationDate':'Date','Country/Region':'Country'},inplace=True)
df

df.query('Country=="UK"')
df.groupby('Date').sum()
confirmed=df.groupby('Date').sum()['Confirmed'].reset_index()
death=df.groupby('Date').sum()['Deaths'].reset_index()
rec=df.groupby('Date').sum()['Recovered'].reset_index()


fig=go.Figure()
fig.add_trace(go.Scatter(x=confirmed['Date'],y=confirmed['Confirmed'],mode='lines+markers',name='Confirmed',line=dict(color='blue',width=2)))

fig.add_trace(go.Scatter(x=death['Date'],y=death['Deaths'],mode='lines+markers',name='Deaths',line=dict(color='red',width=2)))
fig.add_trace(go.Scatter(x=rec['Date'],y=rec['Recovered'],mode='lines+markers',name='Recovered',line=dict(color='green',width=2)))
df_confirmed=pd.read_csv(r'time_series_covid_19_confirmed.csv')
df_confirmed.rename(columns={'Country/Region':'Country'},inplace=True)
df_latlong=pd.merge(df,df_confirmed,on=['Country','Province/State'])
df_latlong
fig=px.density_mapbox(df_latlong,lat="Lat",lon="Long",hover_name="Province/State",hover_data=["Confirmed","Deaths","Recovered"],animation_frame="Date",color_continuous_scale="Portland",radius=7,zoom=0,height=700)
fig.update_layout(title='Worldwide Corona Virus Cases')
fig.update_layout(mapbox_style="open-street-map",mapbox_center_lon=0)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
```

5.2 Special Libraries used

5.2.1 Plotly and Cufflinks:

i. Plotly is an open-source and browser-based graphing library which facilitates interactive plotting. The library is available for several programming languages such as Python, R, MATLAB, Perl, Julia, Arduino, and REST, among others.[2]

ii. Cufflinks is another library that connects the Pandas data frame with Plotly enabling users to create visualizations directly from Pandas. The library binds the power of Plotly with the flexibility of Pandas for easy plotting.

5.2.2 plotly.offline:

i. Plotly allows you to generate graphs offline and save them in local machine. The **plotly.offline.plot()** function creates a standalone HTML that is saved locally and opened inside your web browser.

ii. Use **plotly.offline.iplot()** when working offline in a **Jupyter Notebook** to display the plot in the notebook.

5.2.3 Folium:

Folium is a **Python library used for visualizing geospatial data**. It is easy to use and yet a powerful library. Folium is a Python wrapper for Leaflet. js which is a leading open-source JavaScript library for plotting interactive maps. ... js and the simplicity of Python, which makes it an excellent tool for plotting maps.

5.3 Snapshots

i.Extracting data

Out[4]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	1	Andhra Pradesh	12	0	1	0
1	2	Chhattisgarh	6	0	0	0
2	3	Delhi	38	1	6	1
3	4	Gujarat	43	0	0	3
4	5	Haryana	16	14	11	0
5	6	Himachal Pradesh	4	0	0	1
6	7	Karnataka	20	0	3	2
7	8	Kerala	131	7	11	0
8	9	Madhya Pradesh	23	0	0	1
9	10	Maharashtra	144	3	15	4
10	11	Odisha	3	0	0	0
11	12	Puducherry	1	0	0	0
12	13	Punjab	29	0	0	1
13	14	Rajasthan	41	2	3	0
14	15	Tamil Nadu	32	3	1	1
15	16	Telengana	34	11	1	0
16	17	Chandigarh	7	0	0	0
17	18	Jammu and Kashmir	18	0	1	1
18	19	Ladakh	13	0	0	0
19	20	Uttar Pradesh	42	1	11	0
20	21	Uttarakhand	4	0	0	0
21	22	West Bengal	11	0	0	1
22	23	Bihar	7	0	0	1
23	24	Mizoram	1	0	0	0
24	25	Goa	6	0	0	0
25	26	Manipur	1	0	0	0

Fig 5.3 Snapshot (Extracting data)

ii. Color gradient

Out[12]:

	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total Cases	Active Cases
0	Andhra Pradesh	12	0	1	0	12	11
1	Chhattisgarh	6	0	0	0	6	6
2	Delhi	38	1	6	1	39	32
3	Gujarat	43	0	0	3	43	40
4	Haryana	16	14	11	0	30	19
5	Himachal Pradesh	4	0	0	1	4	3
6	Karnataka	20	0	3	2	20	15
7	Kerala	131	7	11	0	138	127
8	Madhya Pradesh	23	0	0	1	23	22
9	Maharashtra	144	3	15	4	147	128
10	Odisha	3	0	0	0	3	3
11	Puducherry	1	0	0	0	1	1
12	Punjab	29	0	0	1	29	28
13	Rajasthan	41	2	3	0	43	40
14	Tamil Nadu	32	3	1	1	35	33
15	Telengana	34	11	1	0	45	44
16	Chandigarh	7	0	0	0	7	7
17	Jammu and Kashmir	18	0	1	1	18	16
18	Ladakh	13	0	0	0	13	13
19	Uttar Pradesh	42	1	11	0	43	32
20	Uttarakhand	4	0	0	0	4	4
21	West Bengal	11	0	0	1	11	10
22	Bihar	7	0	0	1	7	6
23	Mizoram	1	0	0	0	1	1
24	Goa	6	0	0	0	6	6
25	Manipur	1	0	0	0	1	1

Fig 5.3 Snapshot (Color gradient)

iii.Map

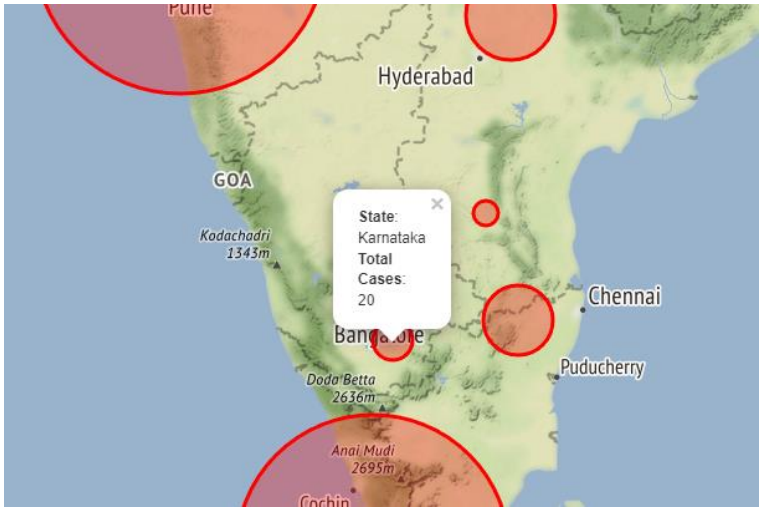

Fig 5.3 Snapshot (Map)

iv.Bar graph

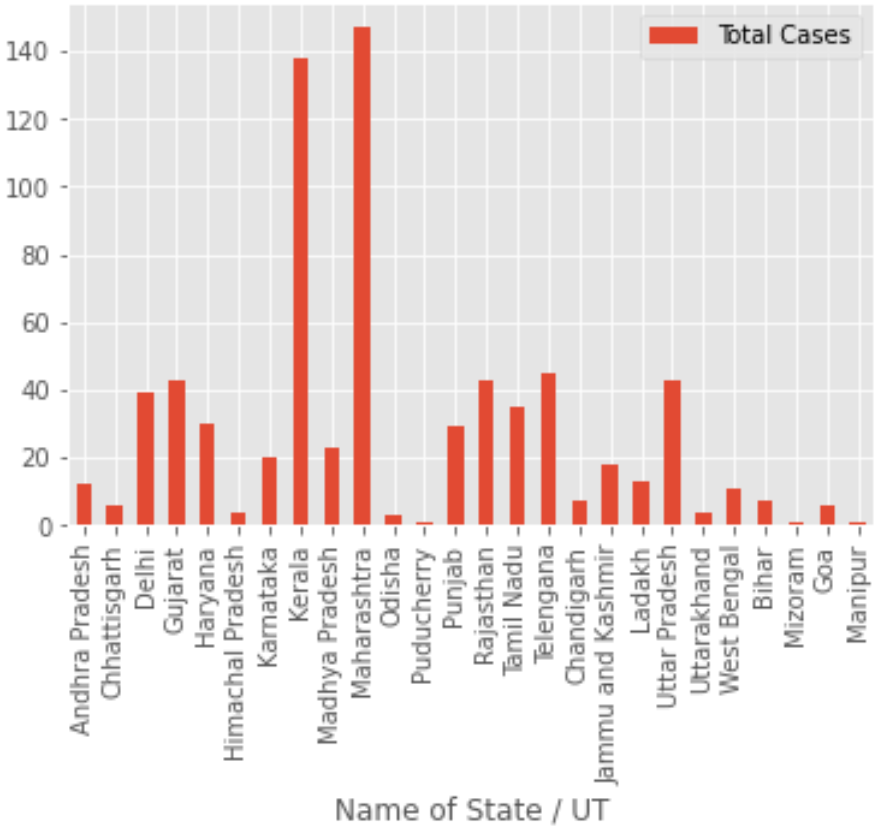

Fig 5.3 Snapshot (Bar graph)

v.Plot graph

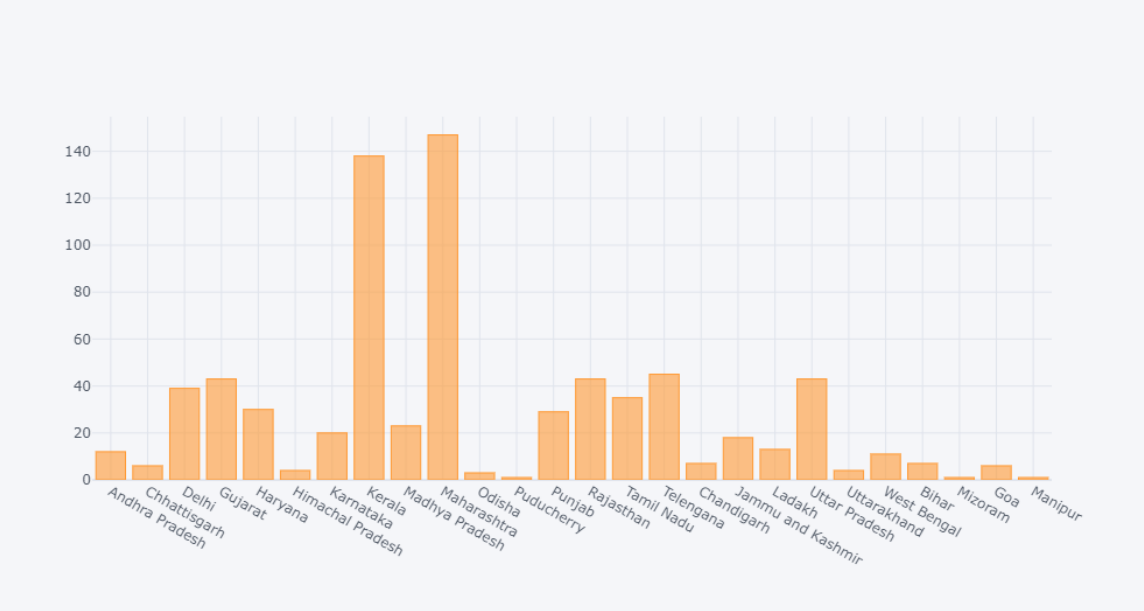

Fig 5.3 Snapshot (Plot graph)

vi.Scatter

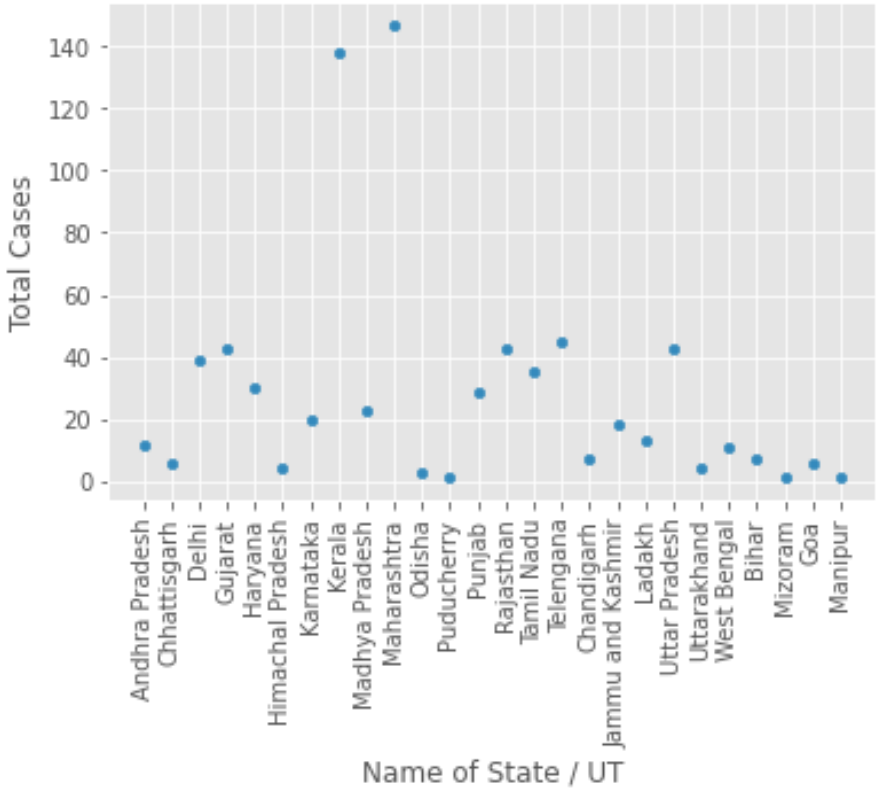

Fig 5.3 Snapshot (Scatter plot)

vii.Scatter iPlot

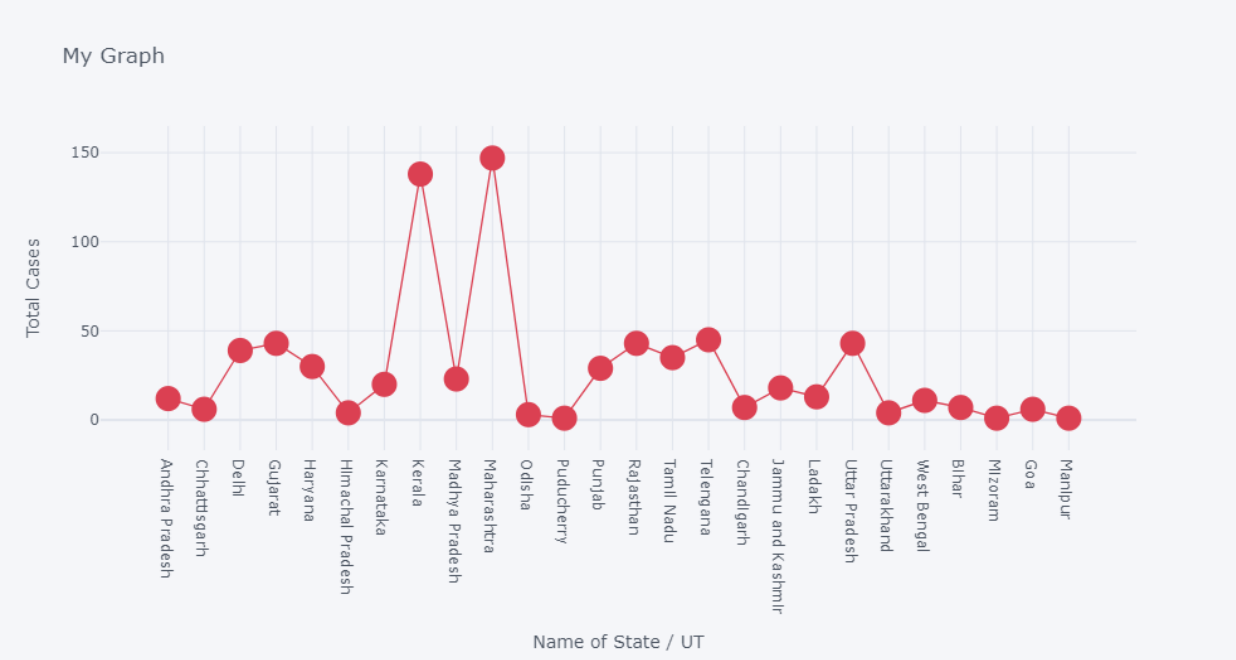

Fig 5.3 Snapshot (Scatter iPlot)

viii.Plotly express

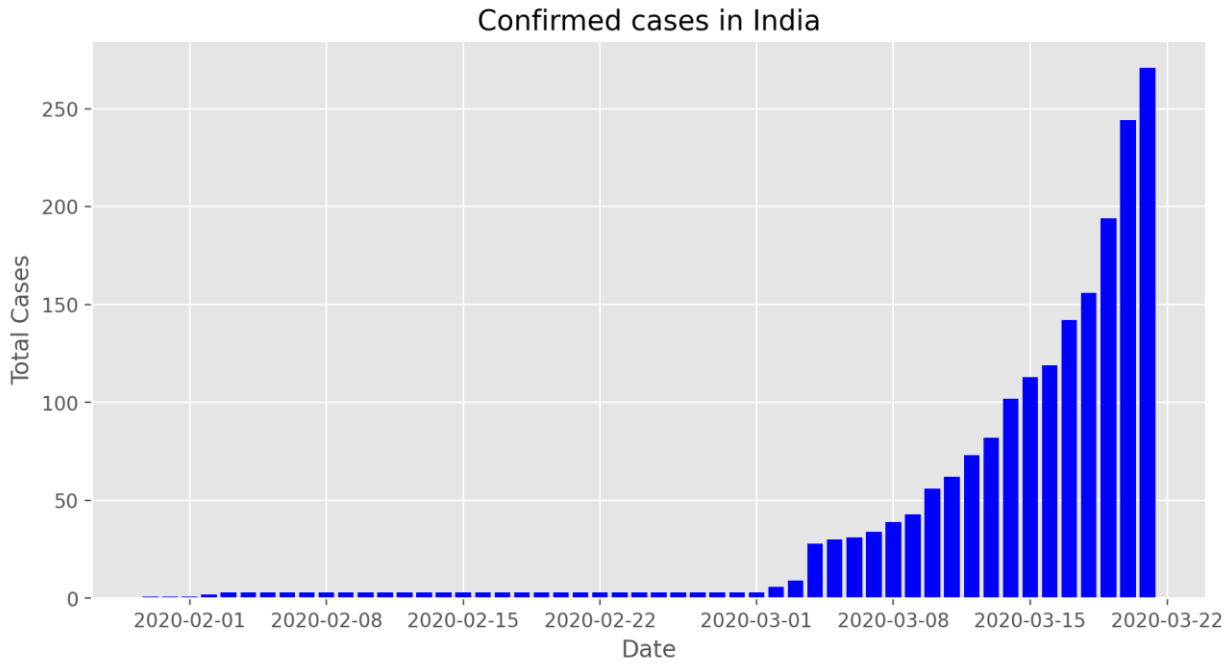

Fig 5.3 Snapshot (Plotly express)

ix.Sub plots

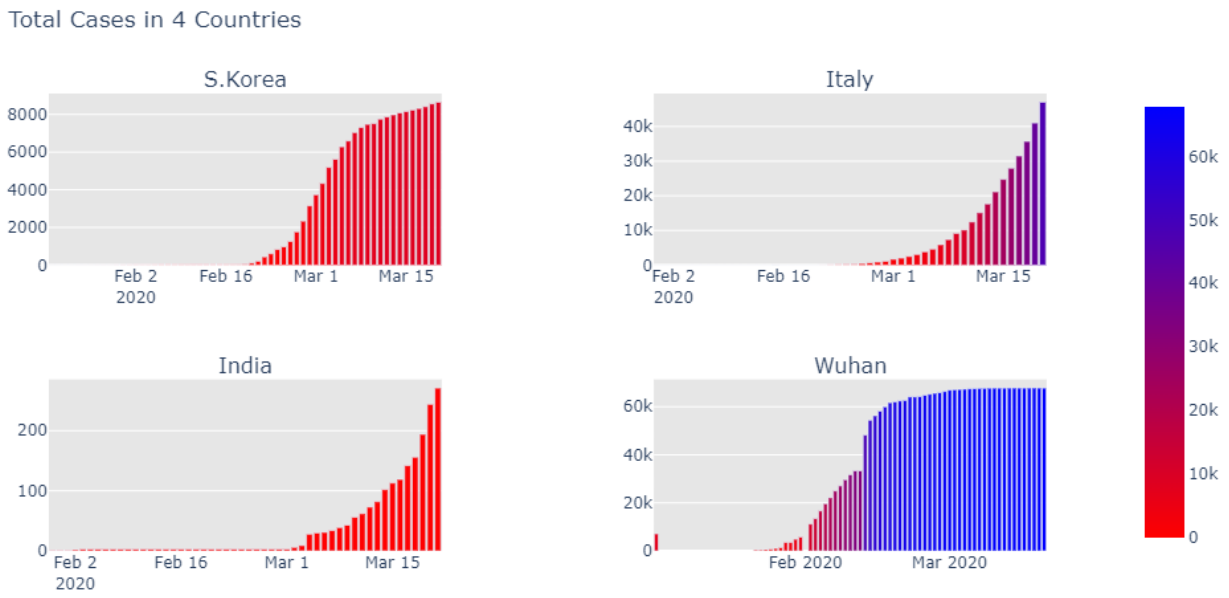

Fig 5.3 Snapshot (Sub plots)

x.Express Density

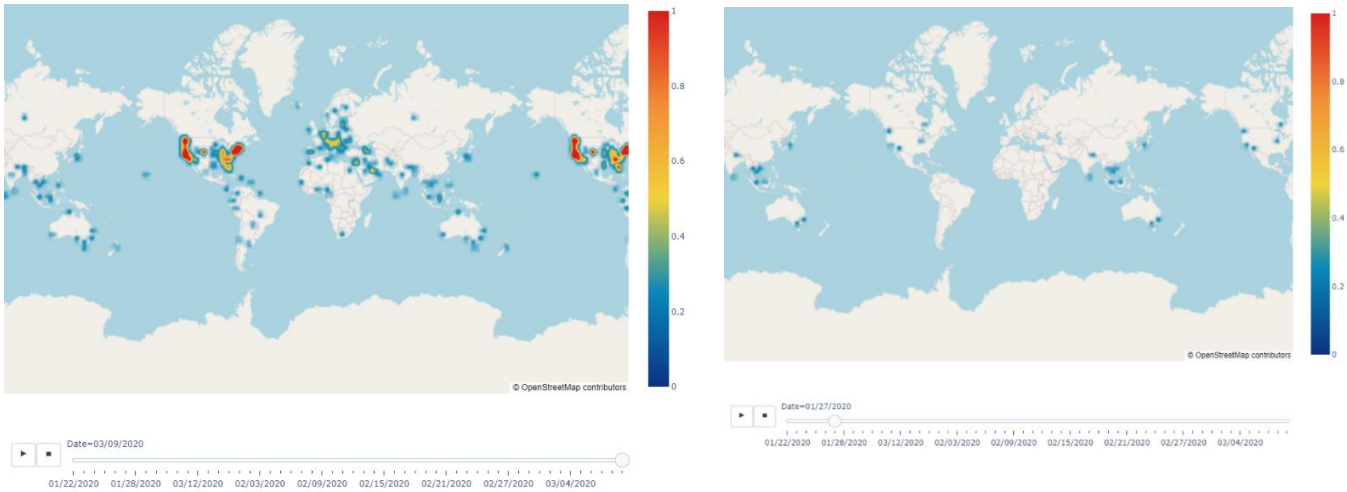

Fig 5.3 Snapshot (Express density)

CHAPTER 6

TESTING

Testing is one of the most critical processes of the Software Development Lifecycle (SDLC). It helps companies to perform a comprehensive assessment of software and ensure that their product fulfills the client's needs.

6.1 Software testing

Software testing is the process in which a developer ensures that the actual output of the software matches with the desired output by providing some test inputs to the software. Software testing is an important step because if performed properly, it can help the developer to find bugs in the software in very less amount of time.

Software testing can be divided into two classes, **Manual testing,** and **Automated testing**. Automated testing is the execution of your tests using a script instead of a human. In this article, we'll discuss some of the methods of automated software testing with Python.

Chapter 7
Result and analysis

The Results section of a scientific research paper represents the core findings of a study derived from the methods applied to gather and analyze information.

7.1 Analysis

India recorded its first COVID-19 case on 30th January 2020 in Kerala. The infected person was a student who had travelled to China for academic purpose. And since then, cases in India are rising exponentially. A complete Dashboard is released on http://siddfulzele.pythonanywhere.com/covid_India/

India had recorded over 500 cases till 24th March. So, government declared nation-wide lockdown from 25th march to 14th April also known as lockdown 1.0 and after this government has been extending nation-wide lockdown step by step. Situation in India till 1st October 2020.

Confirm Cases: 3691166

Recovered: 65081

Active: 785996

Death: 65288

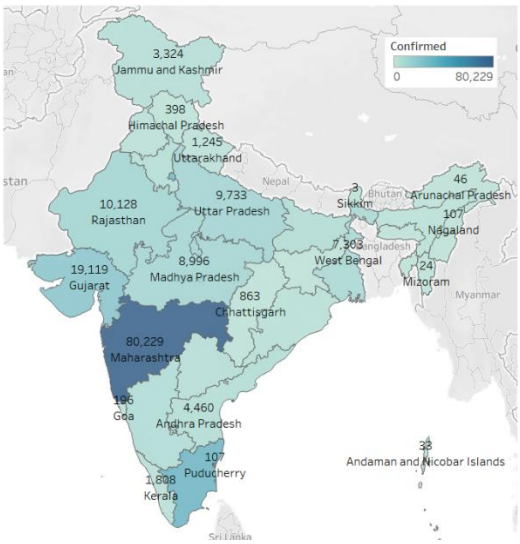

Fig 7.1 Confirmed cases till June-20

7.2 State-wise comparison

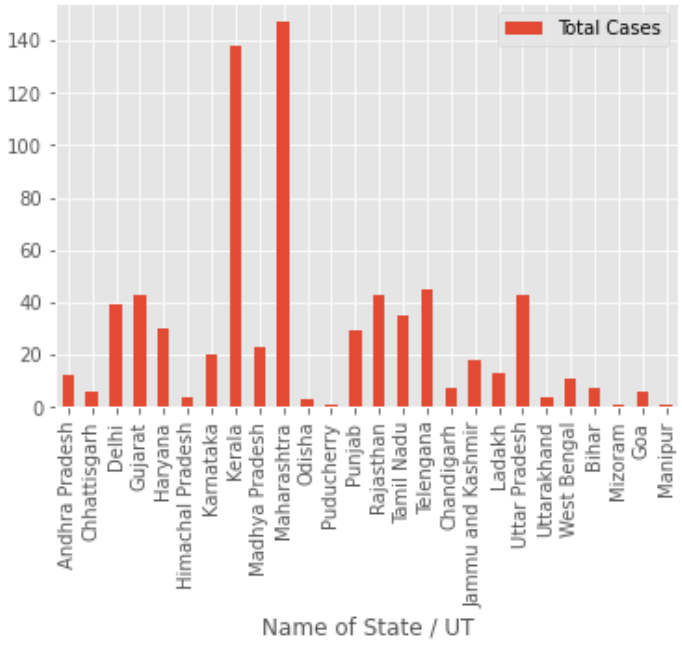

Fig 7.2 State-wise comparison

India consists of 28 states and 8 Union Territories with varying features such as demography, geography, location, lifestyle etc. Which can decide the spread of contagious virus. Comparing COVID-19 situation in states gives us insight of which state/area is to be focused.

Chapter 8
Conclusion and Future work

8.1 CONCLUSION

India is now 6th largest confirmed cases of COVID-19 in the world. India has not reached the peak yet so as of now we cannot predict the approximate number of confirmed cases in India.

Recovery rate of India is also showing exponential behaviour same as confirmed cases. But if cases increase beyond certain point, then thing can go out of control which will affect the recovery rate. Gender information of most of the patients is nor released by the government but whatever data is available shows number of infected males is more than that of females. This may be due to more exposure/contact of males with outdoor world.

Same as gender, age information is unavailable for most of the patients, but whatever data is available shows age and cases are normally distributed and 21-40 is the age bin which has been infected more.

This analysis showed that pandemic like this affects economy the most. Whichever the country's GDP source is it gets targeted the most. Slowing down the economy then unemployment, job losses and then this chain reaction continues.

India was in lockdown for more than 2 moths but still situation did not get any better. This may be due to weak administration or the violation of lockdown by citizens.

8.2 FUTURE ENHANCEMENT

8.2.1 Prediction Model

India has not reached the peak yet, once it reaches the peak the prediction model can be built to show that how much time it will take to get things back to the normal.

8.2.2 Sentiment Analysis

India has never experienced such pandemic in last 100 years so what do people think about this pandemic, lockdown, government approach/policies etc can be studied to have sentiment insight of this pandemic.

References

Reference Books

[1] Mastering Python For Data Science: Samir Madhavan

[2] Learning With Python

Websites

[3] https://api.covid19india.org

[4] http://kaggle.com

[5] http://github.com

[6] www.wikipedia.com

[7] https://www.w3schools.com/

[8] https://www.pythonanywhere.com/

DATASETS PREFERED

[9] COVID19-India API: https://data.covid19india.org/

[10] Coronavirus Cases in India: https://www.kaggle.com/parulpandey/coronavirus-cases-in-india