Visualization of COVID-19 Data using Jupyter Notebook

Zubair Iqbal[1], Prachi Gupta[2] and Kamal Kumar Gola[3]

[1,2]Assistant Professor, Department of Computer Science and Engineering, Moradabad Institute of Technology Moradabad, 244001, India.
[3]Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering, TMU, Moradabad, 244001, India.

[1]zubairiqbal17@gmail.com, [2]prachi.g19@gmail.com, [3]kkgolaa1503@gmail.com

ABSTRACT
This work is an effort to visualize the COVID-19 data using jupyter notebook. As we all know the world is going through one of the most disparaging pandemics started in Dec 2019 from a city name wuhan in china. Until now there is no sign of developing a vaccine for the virus, the things that we can do to stop its spread is social distancing and another thing is analysis of data for better decision to keep the destruction minimum as possible. In view of that we had write some code to visualize the data using one of the most popular tools now a days called jupyter for data analysis.

Keywords
COVID-19, Python, Jupyter, pandemic.

1. INTRODUCTION

A pneumonia of unknown cause detected in Wuhan, China was first reported to the WHO Country Office in China on 31 December 2019. The outbreak was declared a Public Health Emergency of International Concern on 30 January 2020. On 11 February 2020, WHO announced a name for the new coronavirus disease: COVID-19. In a very short span of one month this virus spread in the whole world. The current situation is that there is not a single country which is not affected by the pandemic. Still we don't have a vaccine for this kind of pandemic the ways to stop it from community spread is social distancing and Lockdown. As a preventive measure against the COVID-19 pandemic in India Lock down was scheduled as:

Phase 1: 25 March 2020 – 14 April 2020 (21 days)
Phase 2: 15 April 2020 – 3 May 2020 (19 days)
Phase 3: 4 May 2020 – 17 May 2020 (14 days)
Phase 4: 18 May 2020 – 31 May 2020 (14 days)
Phase 5: (only for containment zones): 1 June 2020 – ongoing.

The purpose of this paper is to visualize the data of confirmed, recovered and deaths due to the coronavirus. This data is very dynamic and changing on daily basis, we have used a dynamically updating data source for this visualization. The purpose of this visualization over jupyter is to understand the jupyter notebook basics for data visualization.

The rest of this paper is organized as follows. Section 2 briefly described the jupyter notebook and

its installation procedure. Section 3 presents the data set used for the purpose and visualization of that data. Section 4 presents the visualization results and observations. Finally, section 5 concludes the paper.

2. JUPYTER NOTEBOOK

The **Jupyter Notebook** is an open-source web application that allows us to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. We are going to use this tool for our data visualization.

2.1 INSTALLING JUPYTER NOTEBOOK

For the installation of Jupyter notebook we are going to use anaconda which is an Open Source Edition and the world's most popular Python distribution platform with over 20 million users worldwide. This can be downloaded from https://www.anaconda.com. In the download section you can choose the suitable platform then download and install individual edition of anaconda distribution.

Now on anaconda prompt type jupyter notebook and press enter or you can also use anaconda navigator to run jupyter notebook. This will run in your preferred web browser and looks like in Fig 1.

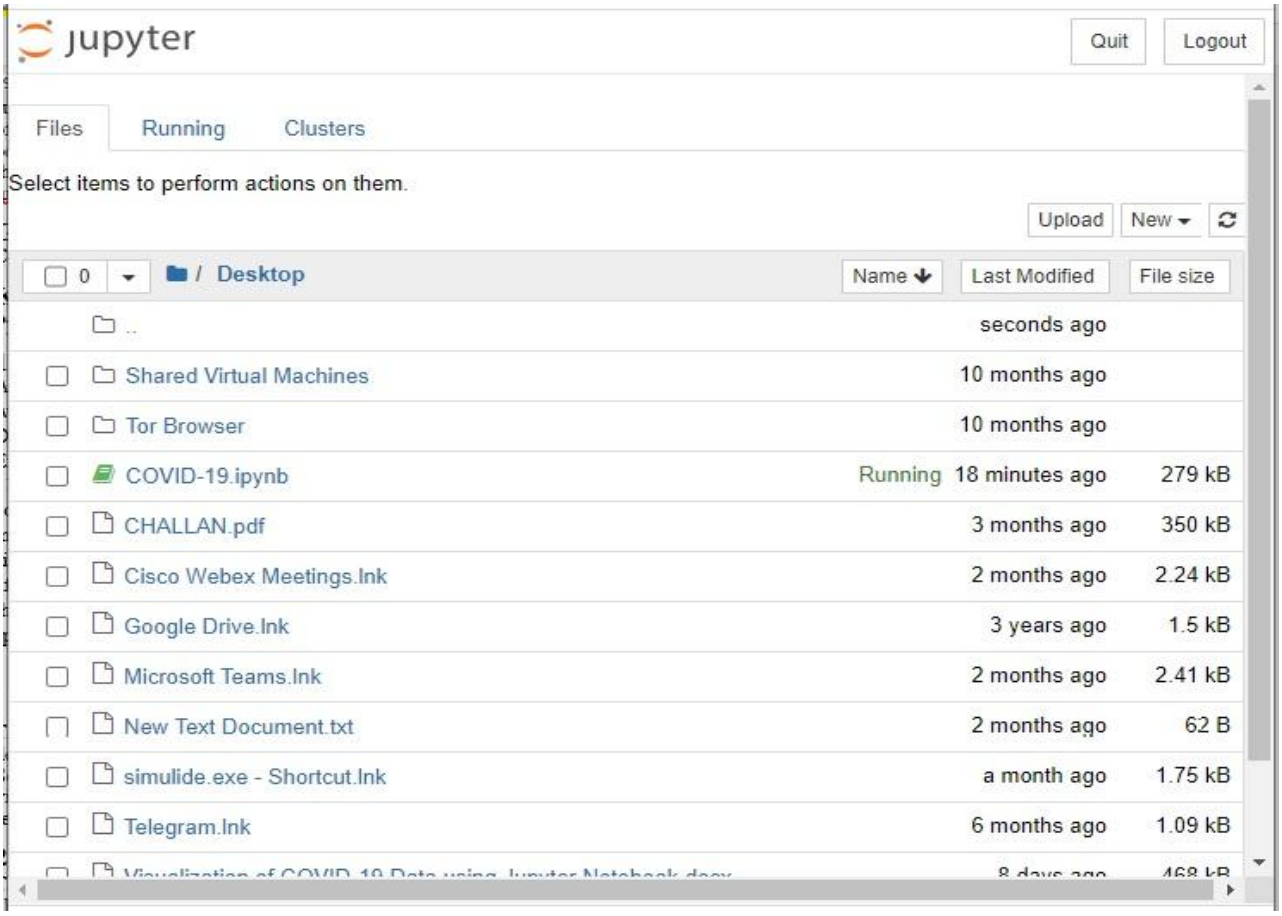

Fig 1: Jupyter Notebook

To start a new project, select the preferred location and choose new option on top right corner of the page. Then a new tab will open from there you can rename your project by just clicking on the name of the project initially it is an unnamed one.

2.1 Pandas

Some of those tools are included in a toolbox that comes with the language, known as the standard library. Others have been built by members of Python's developer community and need to be downloaded and installed from the web.

One that's important for this class is called pandas. It is a tool invented at a financial investment firm that has become a leading open-source library for accessing and analyzing data in many different fields.

```
import pandas as pd
```

2.2 NUMPY

Import numpy by writing the command
Import numpy as np
Numpy is the fundamental package for scientific computing with Python.

3. COVID-19 DATA SET

The first thing to do is Gather data from reliable and updating sources. In this work we are going to use the data set of COVID-19 Data repository by John Hopkins CSSE. This is the data repository for the 2019 Novel Coronavirus Visual Dashboard operated by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). Also, Supported by ESRI Living Atlas Team and the Johns Hopkins University Applied Physics Lab (JHU APL). This data gets updated every 24 hours from the official sources. we accessed the raw files from the csse_covid_19_time_series directory.

The following command is used to read the data from data sourse using panda's read.csv method

```
death_df = pd.read_csv("http://raw.githubusercontent.com/ CSSEGISandData/COVID-19/master/
csse_covid_19_time_series  /time_series_covid19_deaths_global.csv")
```

death_df.head() command is used to visualize the data as Fig2

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	6/20/20	6/21/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	569	581
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	43	44
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	837	845
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	52	52
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	9	9

Fig 2: Death Cases

Similarly, we can gather the data of confirmed cases Fig3 and recovered cases Fig4 from the same data source stored in the file names: time_series_covid19_confirmed_global.csv and time_series_covid19_recovered_global.csv

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	6/20/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	28424
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	1891
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	11631
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	855
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	176

Fig 3: Confirm Cases

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	6/20/20	6/21/20	6/22/20	6/23/20	6/24/20	6
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	8292	8764	8841	9260	9869	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	1126	1134	1159	1195	1217	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	8324	8422	8559	8674	8792	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	792	792	796	797	797	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	66	77	77	77	77	

Fig 4: Recovered Cases

4. PLOTLY

The plotly Python's library is an interactive, opensource plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

For the purpose of this visualization we have used plotly. First we need to import plotly if it is not installed you can run the command pip install plotly on the terminal to get it install.

This command is used to plot a scatter graph using plotly. This command plot 10 worst hit countries

fig=px.scatter(sorted_country_df.head(10),x='country_region',y='confirmed',size='confirmed',color='country_region',hover_name='country_region',size_max=60)

fig.show() is used to plot the graph shown in Fig5.

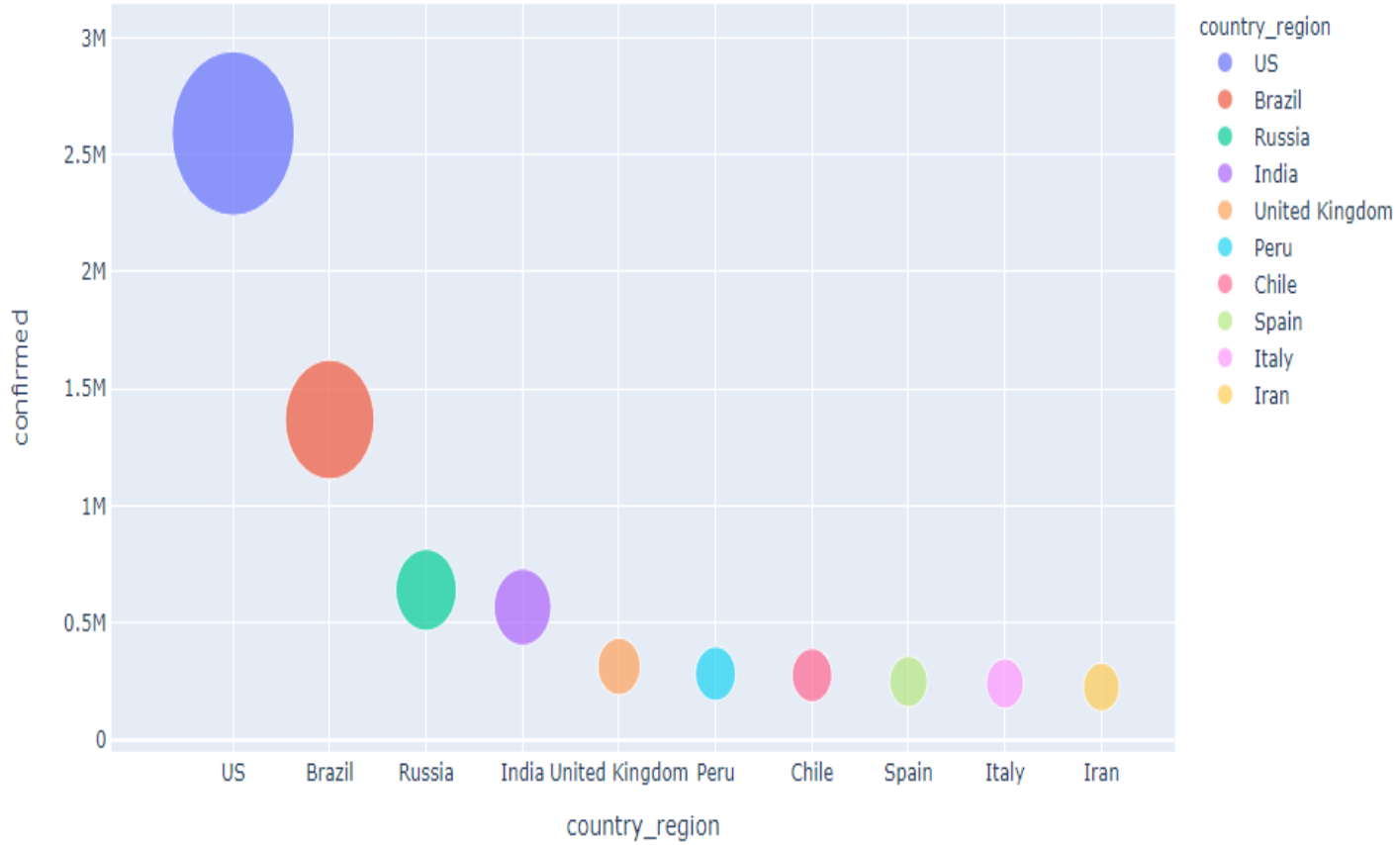

Fig 5: 10 worst hit countries

5. FOLIUM

folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. It enables both the binding of data to a map for choropleth visualizations as well as passing rich vector/raster/HTML visualizations as markers on the map.

The library has a number of built-in tilesets from OpenStreetMap, Mapbox, and Stamen, and supports custom tilesets with Mapbox or Cloudmade API keys. folium supports both Image, Video, GeoJSON and TopoJSON overlays.

We have used folium to plot confirmed cases of COVID-19 on world map. First, we need to install folium by running command pip install folium on terminal. Then import it by writing import folium. We have then created a function by the name world_map to define the properties of the graph and plot the confirmed cases

```
world_map=folium.Map(location=[11,0],title="cartodbpositron",zoom_start=2,max_zoom=6,min_zoom=2)

for i in range(len(confirmed_df)):

    folium.Circle(location=[confirmed_df.iloc[i]['lat'],confirmed_df.iloc[i]['long']],

    fill=True,radius=(int((np.log(confirmed_df.iloc[i,-1]+1.00001)))+0.2)*50000,fill_color='blue',
```

color='red').add_to(world_map)

then show the world_map object we have created by : world_map

Fig 6: Confirmed cases over world map

6. CONCLUSION

The results of this visualization are shown in figures these results are interactive and can be used for the purpose of presenting data, not only the covid data but it can be used for other data set also.

We have shown the code for basic functionalities of notebook these functionalities can be extended to professional level and can be combined with data analysis techniques to draw some conclusion and that may help our country to fight with this kind of pandemic in near future.

7. FUTURE SCOPE

This work can be extended by including some data prediction algorithms to predict the future of this outbreak. If some good prediction models are used, we can predict which turn will be taken by this outbreak of COVID-19 in near future. However, there are many other tools of machine learning that can help for better visualization and prediction of this outbreak.

8. REFERENCES

[1] World Health Organization (WHO):https://www.who.int/

[2] Gettleman, Jeffrey; Schultz, Kai (24 March 2020). "Modi Orders 3-Week Total Lockdown for All 1.3 Billion Indians". The New York Times. ISSN.

[3] Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE):https://systems.jhu.edu/.

[4] Lauren G. Coronavirus COVID-19 Global Cases by Johns Hopkins CSSE January 23, 2020. https://gisanddata .maps.arcgis.com/apps/opsdashboard/index.html.

[5] Plotly Documentation https://plotly.com/python-api-reference/.

[6] Folium Documentation https://python-visualization.githu b.io/folium/#.

