School of Computer Science and Engineering

VISUAL ANALYTICS ON COVID-19

CSE3020 – DATA VISUALISATION

PROJECT-BASED COMPONENT REPORT

By

Bella Babu – 20BCE0558

B. Shivani – 20BCE0563

Debasmita Paul – 20BCE0841

Dishi Agarwal – 20BCE0732

Eshana Mohan- 19BCE2216

DECLARATION

I hereby declare that the report entitled “Visual Analytics on Covid-19” submitted by me, for the CSE3020 DATA VISUALISATION (EPJ) to VIT is a record of bonafide work carried out by me under the supervision of Dr.S. VENGADESWARAN.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

Place: Vellore

Date: 28/04/2022

Shivani

Shivani

Signature of the Candidate

CONTENTS

ABSTRACT

In this project, we have created a visual dashboard to represent the data that was generated during the global outbreak, Covid-19. The data we collected consisted of the number of active cases, the total deaths, and the number of recovered cases for various countries. With this data, several graphs interactive and non-interactive graphs were designed. Through this dashboard, one could infer how the wave of Covid-19 had infected the entire world. Through our dashboard, we could also predict how Covid-19 may spread in the future.

INTRODUCTION

Objective: To create a visualization dashboard to show the effect of Covid-19 on the world.

Problem Statement: As the covid-19 disease and pandemic have led to a crisis across the globe, people across the globe are facing many deaths. This virus has hurt in every country across the globe and therefore we have implemented a visualization dashboard that has graphs based on the number of deaths, recovered cases and currently active cases. This would help countries prepare better for the covid battle case of the worst scenario third wave and also countries can demand resources from other countries if needed.

Functional Requirements: Datasets, Python

The global outbreak of COVID-19 has had a strong impact on economic and social life in various countries. Every day there are a huge number of positive cases, recoveries and deaths in every country. It is very difficult to analyze the data and difficult to understand the analysis, we thought that the visual representation of covid19 data is a better way to represent the data to be understood by everyone.

COVID-19 has become a pandemic that is affecting our daily routine very badly, we need to understand its effect more broadly. We are going to create visualizations on COVID-19. This project aims to visualize, deaths, and record series The visualizations help to interpret the way a country is handling the situation or how badly a country is affected by the COVID-19 pandemic.

Applications: The data obtained can further be trained over again for developing future preventive methods. Helps to effectively analyze the fast-moving disease as efficiently as possible Potential to handle appropriate information regarding the disease. By providing the captured data, this technology helps in the identification of the infected cases undertaking take a further analysis of the level of risks. Quickly helps to identify the infected patient at an early stage. Helps to analyze and identify persons who can be infected by this virus in the future.

A Study on Covid-19

Coronavirus disease 2019 (COVID-19) is a coronavirus 2 (SARS-CoV-2)-related viral disease that causes severe acute respiratory sickness. It was first found in December 2019 in Wuhan, China, and has since spread over the world, leading to a pandemic. The first incident occurred on November 17, 2019. More than 17 million cases had been documented across 188 nations and territories as of June 1, 2021, leading to over 3,72,000 deaths. Over 8.196 million people have regained their health.

Coughing, sneezing, and chatting are the most typical ways for the virus to spread through close contact. Droplets generally fall to the ground or onto objects rather than travelling great distances in the air. By contacting a contaminated surface and subsequently contacting one's face, the infection can be caught. The most run of the mill ways for the infection to spread between individuals in closeness incorporate hacking, sniffling, and talking.

India's first COVID-19 case was reported in Kerala on January 30, 2020. It was a student who had visited China previously. On March 25th the entire country was put on a lockdown. India had around 200 thousand confirmed cases as of the beginning of June. Infection rates started to drop in September, along with the number of new and active cases. Daily cases peaked in mid-September with over 90,000 cases reported per day, dropping to below 15,000 in January 2021. A second wave beginning in March 2021 was much more devastating than the first, with shortages of vaccines, hospital beds, oxygen cylinders and other medical supplies in parts of the country. By late April, India led the world in new and active cases. On 30 April 2021, it became the first country to report over 400,000 new cases in 24 hours. By March 2022, India had just 22,487 cases across the country. With 58.8% population fully vaccinated and 70% having received at least one dose opening up post-pandemic has been steady.

DATA ABSTRACTION

The attributes that we have taken are the number of active cases, the number of recovered cases, and the number of deaths, date-wise, as well as the country names. These attributes can be classified as:

The number of cases and the date is classified as Quantitative Data, as the data consists of discrete values.

The country names are classified as Qualitative Data, as this data is descriptive.

For Date, we have used the level of measurement as Interval, as in this every value is placed at an equal distance from each other, and there is no value that we can consider as zero.

For the number of cases, we have used the level of measurement as a Ratio, as in this we can determine a definitive ratio between two values and we have elements where the value is an absolute “zero”.

For Country names, we have used the level of measurement as Nominal, as in this the data can be labeled into mutually exclusive categories within a variable, but these categories cannot be ordered in a meaningful way.

The type of dataset that is used is Table, as the given data can be arranged in rows and columns, where each cell may have a unique value.

The dataset type is Table.

TASK ABSTRACTION

Analyze:

- Analyze the given data sets.
- Discover the way covid cases can rise or fall.

Produce:

- Record: A bar graph is made between
- Derive: We can derive the rate of inc/dec in date and No. Of Cases the number of covid cases.

Search:

- A Lookup can be performed on the dataset

Query:

- Identify: If we only search for one target
- Compare: Comparing multiple sets of values to understand the trend in covid cases.
- Summarize: We can find the aggregate values.

ANALYSIS OF METHODOLOGY USED:

Python is an open-source, interpreted, high-level language and provides a great approach for object-oriented programming. It is one of the best languages used by data scientists for various data science projects/applications. Python provides great functionality to deal with mathematics, statistics, and scientific function.

LIBRARIES REQUIRED:

Numpy

Numpy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

Pandas

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, statistics, analytics, etc.

Ggplot2

ggplot is a Python implementation of the grammar of graphics. Ggplot2 is a plotting package that provides helpful commands to create complex plots from data in a data frame. It provides a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties.

Matplotlib.pyplot

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

Seaborn

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with data frames and the Pandas library. The graphs created can also be customized easily. Seaborn allows the creation of statistical graphics. It supports multiplot grids. Allows comparison between multiple variables.

DASHBOARD IMPLEMENTATION:

Google collab:

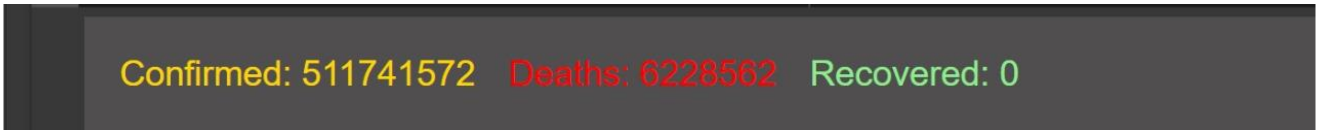
https://colab.research.google.com/drive/1A7wsqLy05QPyNUVF1fFUQcdJPwDT7JAE?usp=sharing

Dataset:

https://drive.google.com/drive/folders/1kmilrSvznHminEICmlYGD-TR8NF_ieup?usp=sharing

RESULT AND ANALYSIS:

This displays the total number of confirmed, dead, and recovered cases.

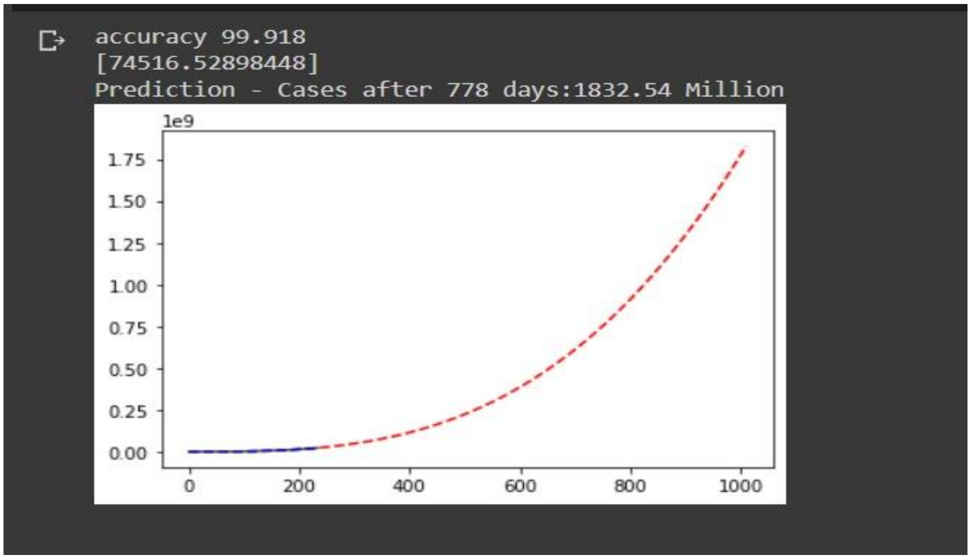


Confirmed: 511741572 Deaths: 6228562 Recovered: 0

The line graph below indicates the prediction of covid cases after 778 days which count to be 1832.54 Million with an accuracy of 99.918%.

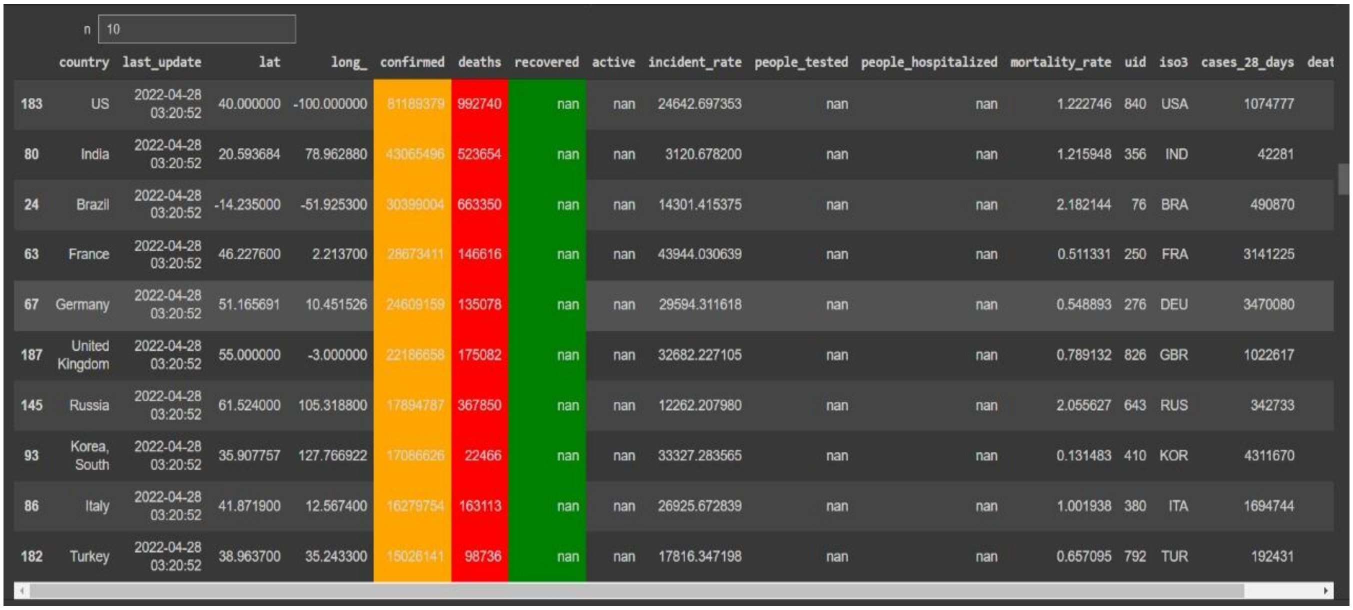
MARKS: Dimensional line mark is used.

CHANNELS: Color channel is used.



```
accuracy 99.918
[74516.52898448]
Prediction - Cases after 778 days:1832.54 Million
```

It is an interactive table that displays the n leading countries’ highest number of confirmed covid cases and highest number of deaths.



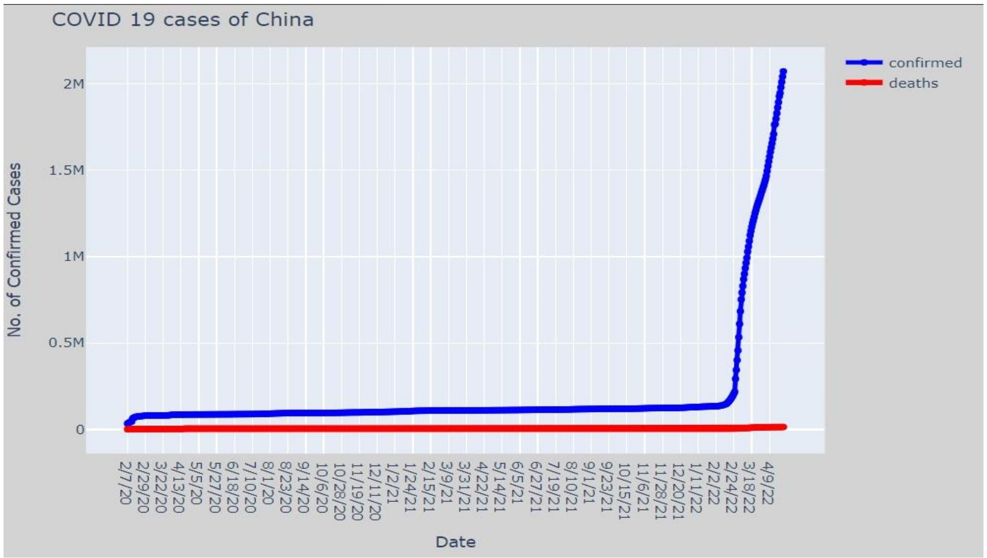
n 10

	country	last_update	lat	long_	confirmed	deaths	recovered	active	incident_rate	people_tested	people_hospitalized	mortality_rate	uid	iso3	cases_28_days	deat
183	US	2022-04-28 03:20:52	40.000000	-100.000000	81189379	992740	nan	nan	24642.697353	nan	nan	1.222746	840	USA	1074777	
80	India	2022-04-28 03:20:52	20.593684	78.962880	43065496	523654	nan	nan	3120.678200	nan	nan	1.215948	356	IND	42281	
24	Brazil	2022-04-28 03:20:52	-14.235000	-51.925300	30399004	663350	nan	nan	14301.415375	nan	nan	2.182144	76	BRA	490870	
63	France	2022-04-28 03:20:52	46.227600	2.213700	28673411	146616	nan	nan	43944.030639	nan	nan	0.511331	250	FRA	3141225	
67	Germany	2022-04-28 03:20:52	51.165691	10.451526	24609159	135078	nan	nan	29594.311618	nan	nan	0.548893	276	DEU	3470080	
187	United Kingdom	2022-04-28 03:20:52	55.000000	-3.000000	22186658	175082	nan	nan	32682.227105	nan	nan	0.789132	826	GBR	1022617	
145	Russia	2022-04-28 03:20:52	61.524000	105.318800	17894787	367850	nan	nan	12262.207980	nan	nan	2.055627	643	RUS	342733	
93	Korea, South	2022-04-28 03:20:52	35.907757	127.766922	17086626	22466	nan	nan	33327.283565	nan	nan	0.131483	410	KOR	4311670	
86	Italy	2022-04-28 03:20:52	41.871900	12.567400	16279754	163113	nan	nan	26925.672839	nan	nan	1.001938	380	ITA	1694744	
182	Turkey	2022-04-28 03:20:52	38.963700	35.243300	15026141	98736	nan	nan	17816.347198	nan	nan	0.657095	792	TUR	192431	

The line graph below shows the relation between the number of confirmed cases versus the Date. The blue line is confirmed cases whereas the red line indicates the number of deaths.

MARKS: Dimensional line mark is used.

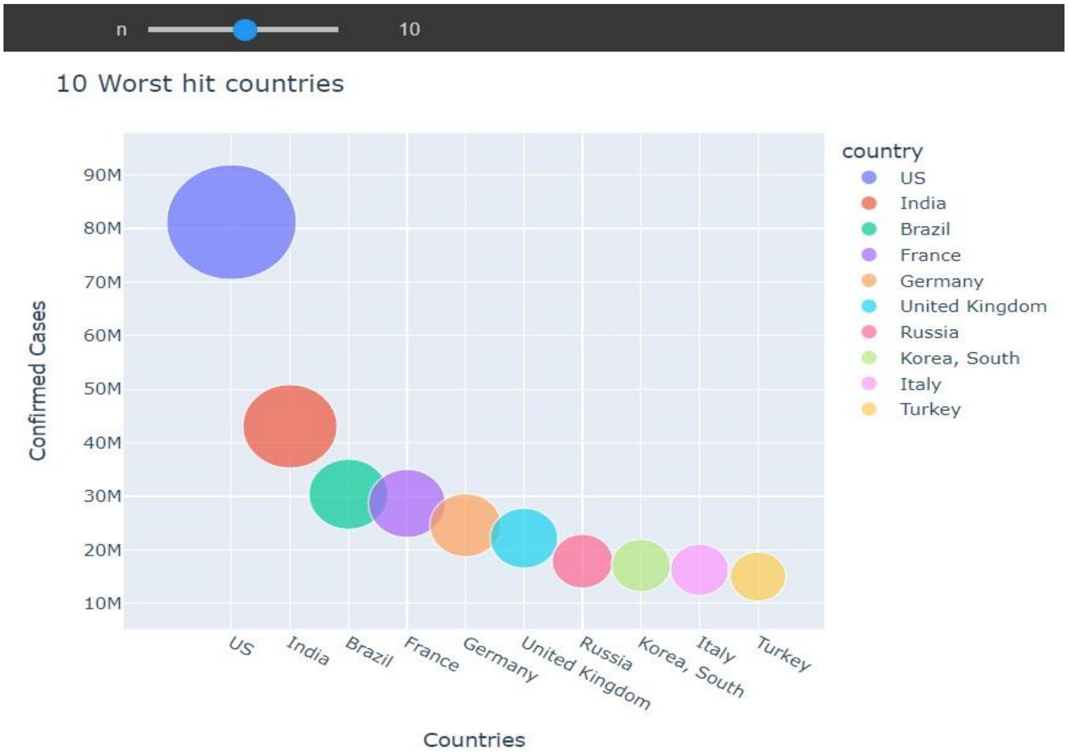
CHANNELS: Color channel is used to differentiate confirmed cases and deaths.



The Bubble graph below shows the relation between the number of confirmed cases versus Countries for 10 countries.

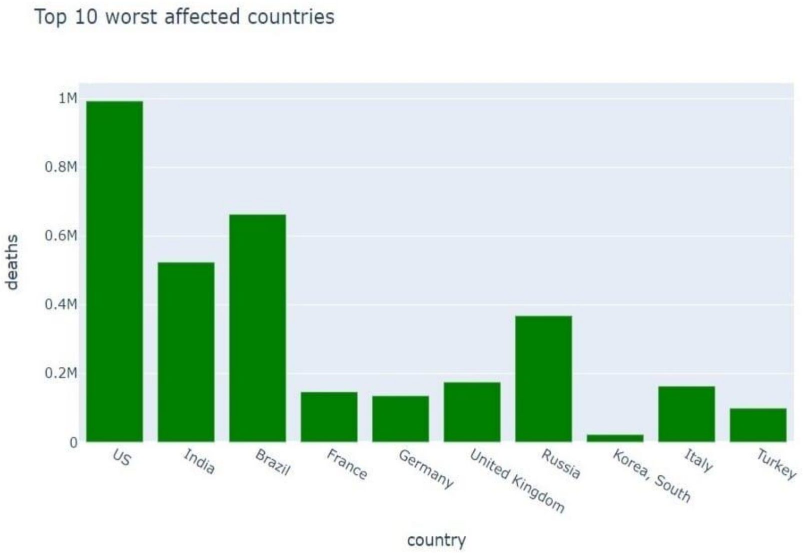
MARKS: Point marks are used.

CHANNELS: Color channel is used to differentiate cases of each country and we use the size channel to indicate the number of confirmed cases for each country.



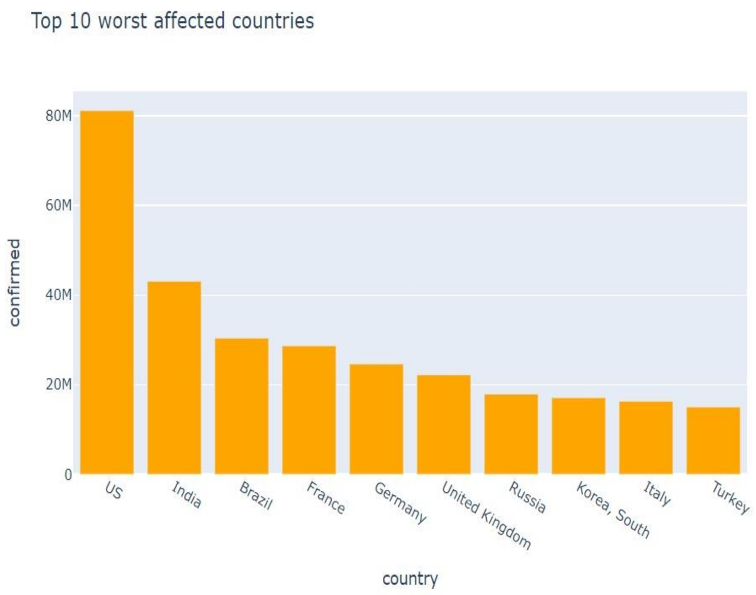
The Bar graph below shows the relation between the number of death cases versus Countries for 10 countries.

MARKS: Area marks are used.



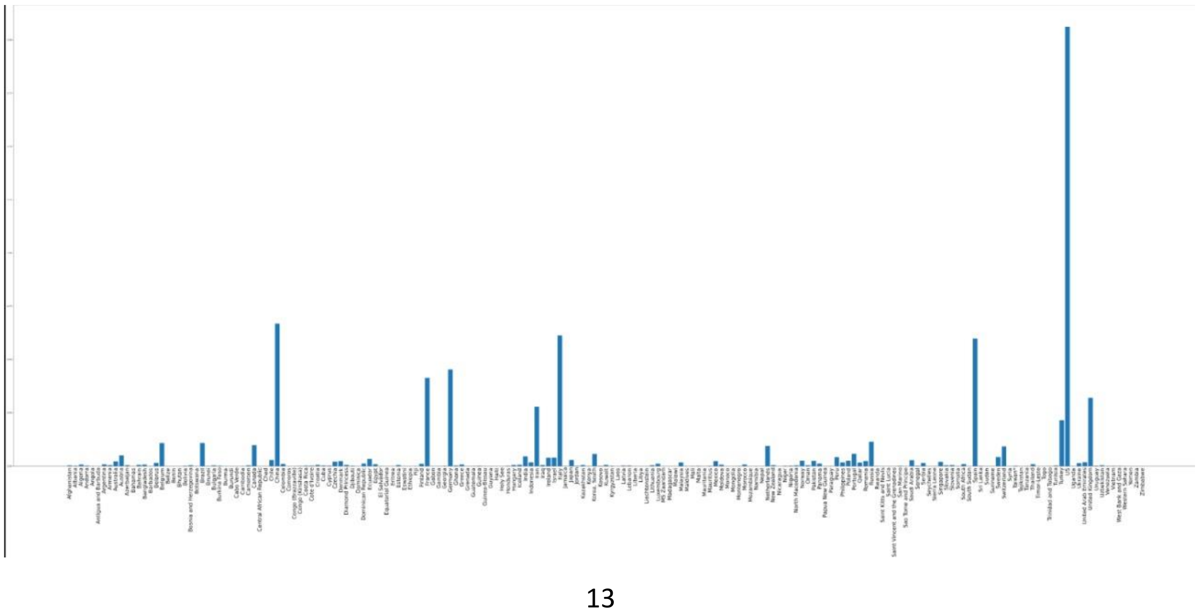
The Bar graph below shows the relation between the number of confirmed covid cases versus Countries for 10 countries.

MARKS: Area marks are used.



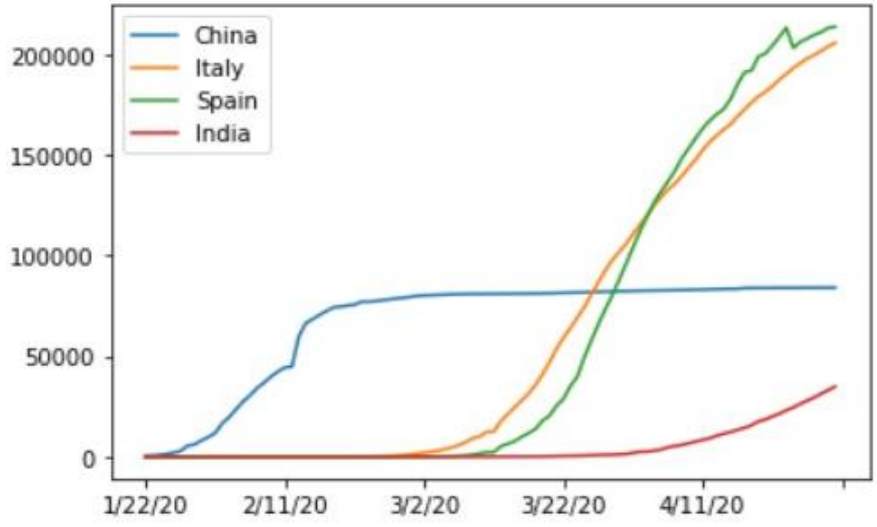
The Bar graph below shows the Mean infection rates for each country.

MARKS: Area marks are used.



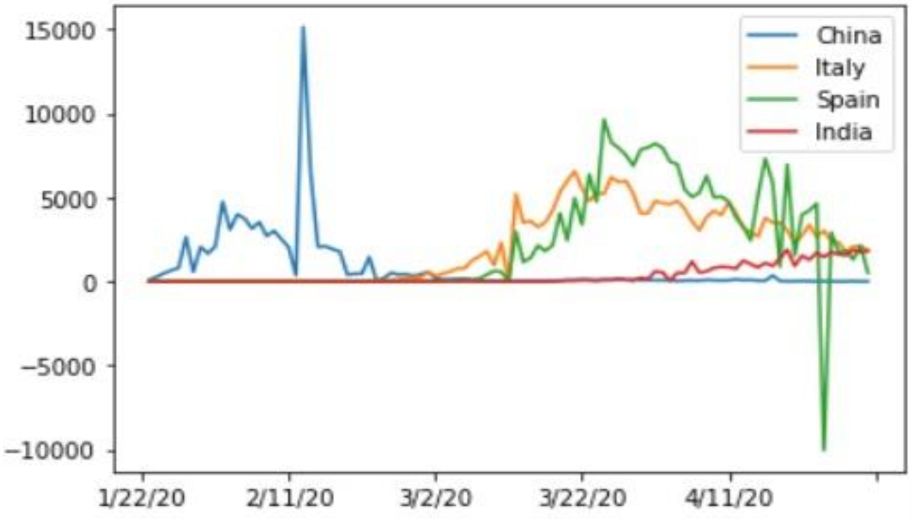
The Line graph below compares Mean infection rates for 4 countries.

MARKS: line marks are used.



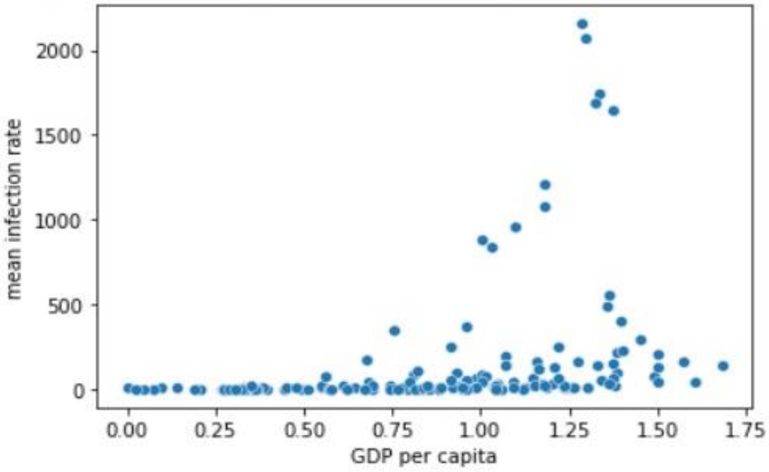
The Line graph below shows the difference between each day's Mean infection rates for 4 countries. This can help these countries to have an insight into the increase in several cases each day.

MARKS: line marks are used.



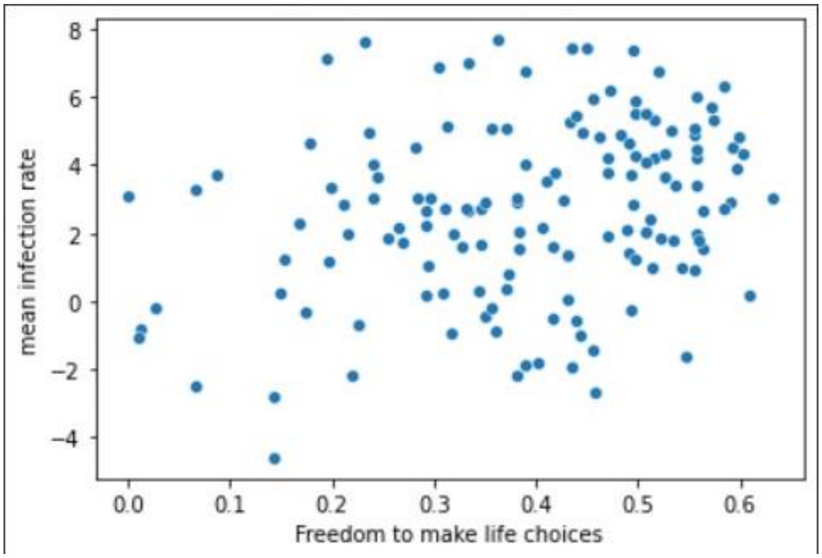
The scatterplot graph below shows how GDP per capita influences the Mean infection rates. This plot shows that as GDP per capita increases, the number of cases also increases due to more international exports and imports as well as increased tourism.

MARKS: Point marks are used.



The scatterplot graph below shows how the freedom to make life choices in each country influences the Mean infection rates. This plot shows that like the freedom to make life choices increases, the number of cases also increases due to a more expensive lifestyle.

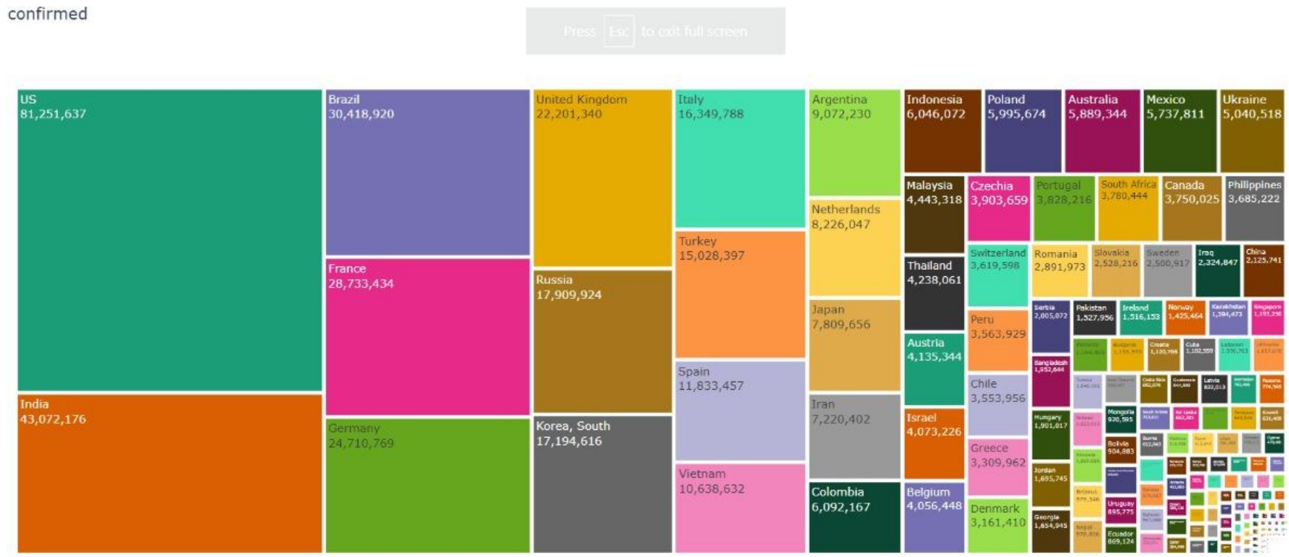
MARKS: Point marks are used.



TreeMap shows several confirmed Covid cases across world countries.

Marks: Area

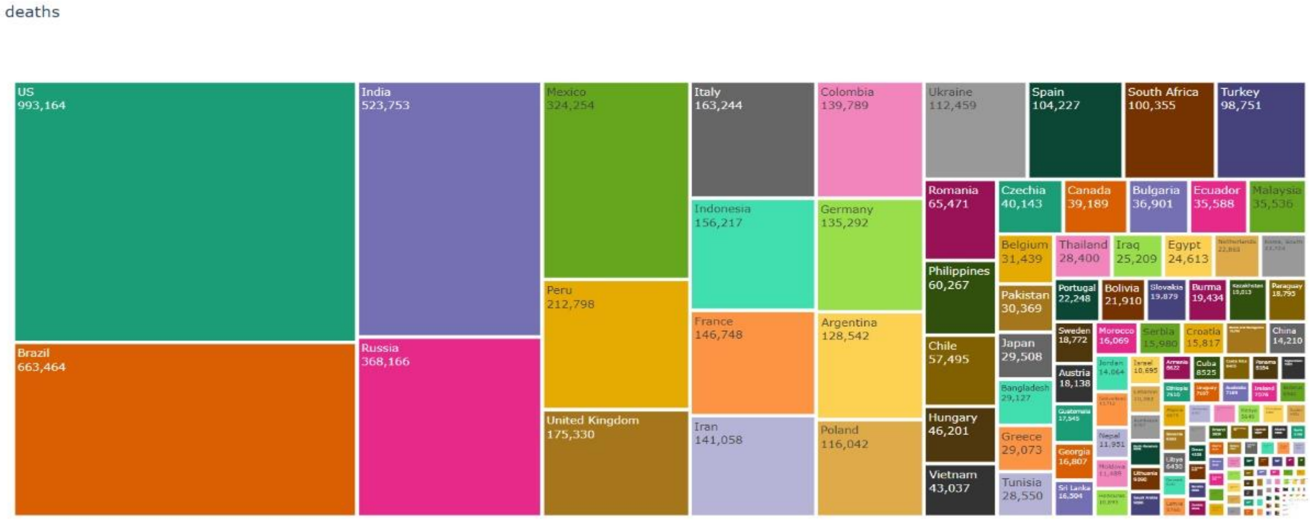
Channels: Color



TreeMap shows the number of deaths in Covid cases across world countries.

Marks: Area

Channels: Color



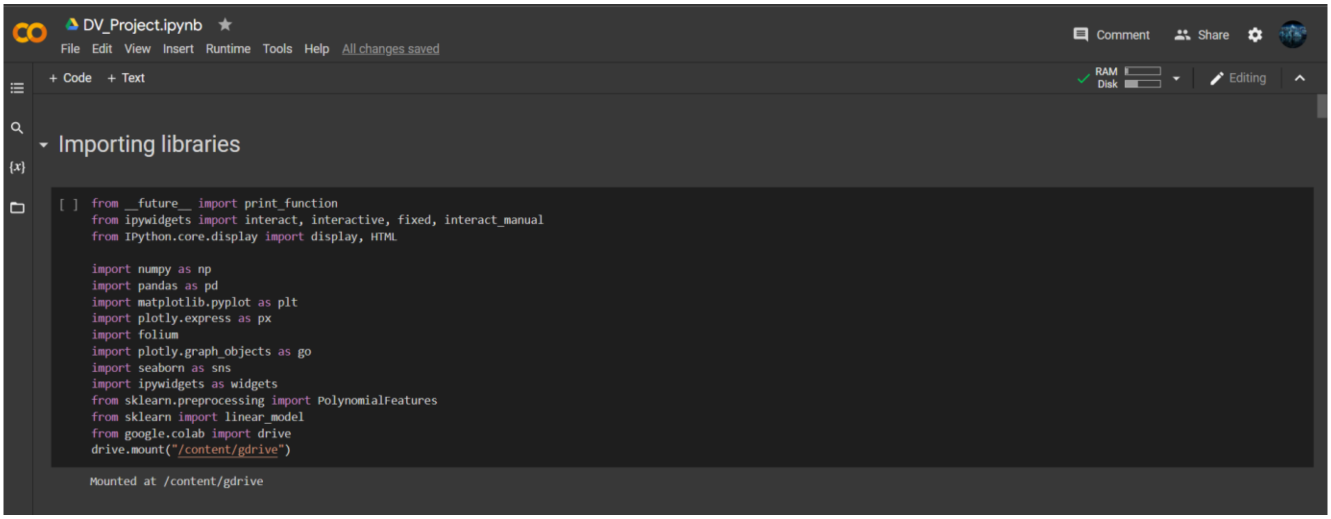
CONCLUSION:

Our prediction of covid cases after 778 days which count to be 1832.54 Million had an accuracy of 99.918% that is, it is very close to real numbers, which indicates that we can use this representation to predict future trends. We can also see the most confirmed cases are in the USA followed by India and Brazil. In the plot, we also see the most confirmed cases are in the USA, India, and Brazil, the 3 worst-hit countries. But, in a number of deaths, Brazil overtakes India. We can also see the most confirmed cases are in US, India, and Brazil.

The number of cases for China, Italy, Spain, and India has also been compared. In the 2020s, Italy and Spain rose exponentially, while the number of cases in India slowly increased. However, we also note that the number of Chinese cases remained constant after some time.

When the GDP per capita and freedom to make life-altering decisions increase, the mean infection rate increases, and cases increase more in developing countries. As the freedom to make life choices increases the number of cases also increases due to a more expensive lifestyle. From the dataset, we observed that the freedom to make life choices is more in a democratic country. We also observe that as GDP per capita increases, the number of cases also increases due to more international exports and imports as well as increased tourism.

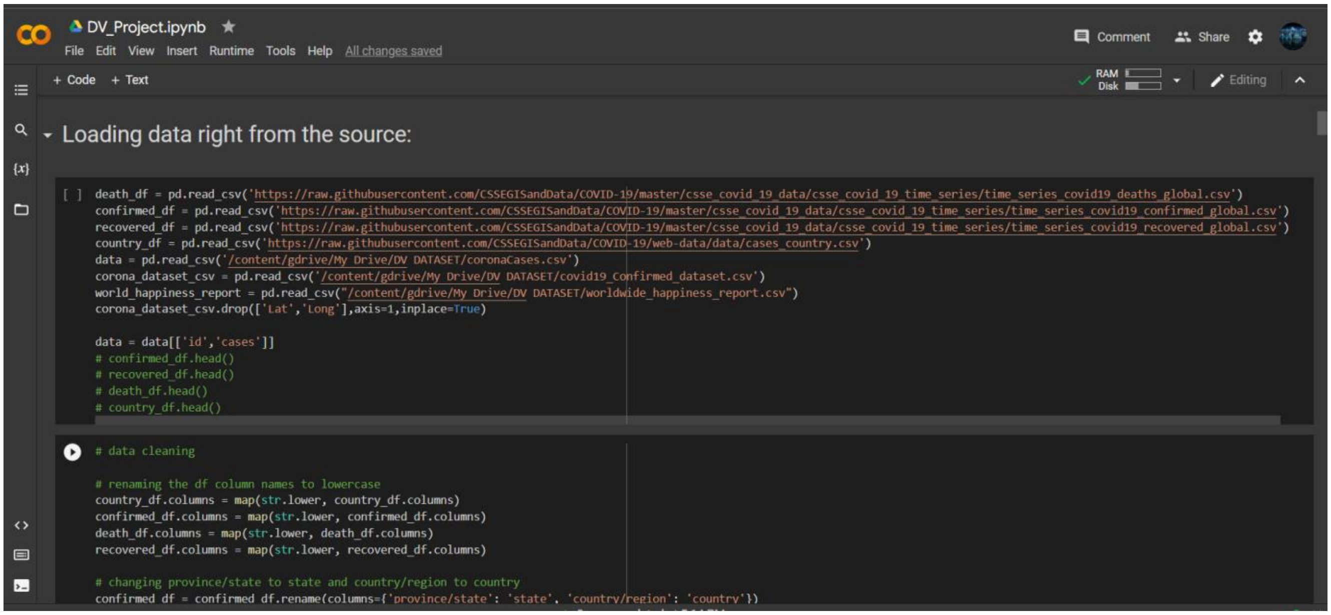
APPENDIX:



Importing libraries

```
[ ] from __future__ import print_function
    from ipywidgets import interact, interactive, fixed, interact_manual
    from IPython.core.display import display, HTML

    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import plotly.express as px
    import folium
    import plotly.graph_objects as go
    import seaborn as sns
    import ipywidgets as widgets
    from sklearn.preprocessing import PolynomialFeatures
    from sklearn import linear_model
    from google.colab import drive
    drive.mount("/content/gdrive")
```

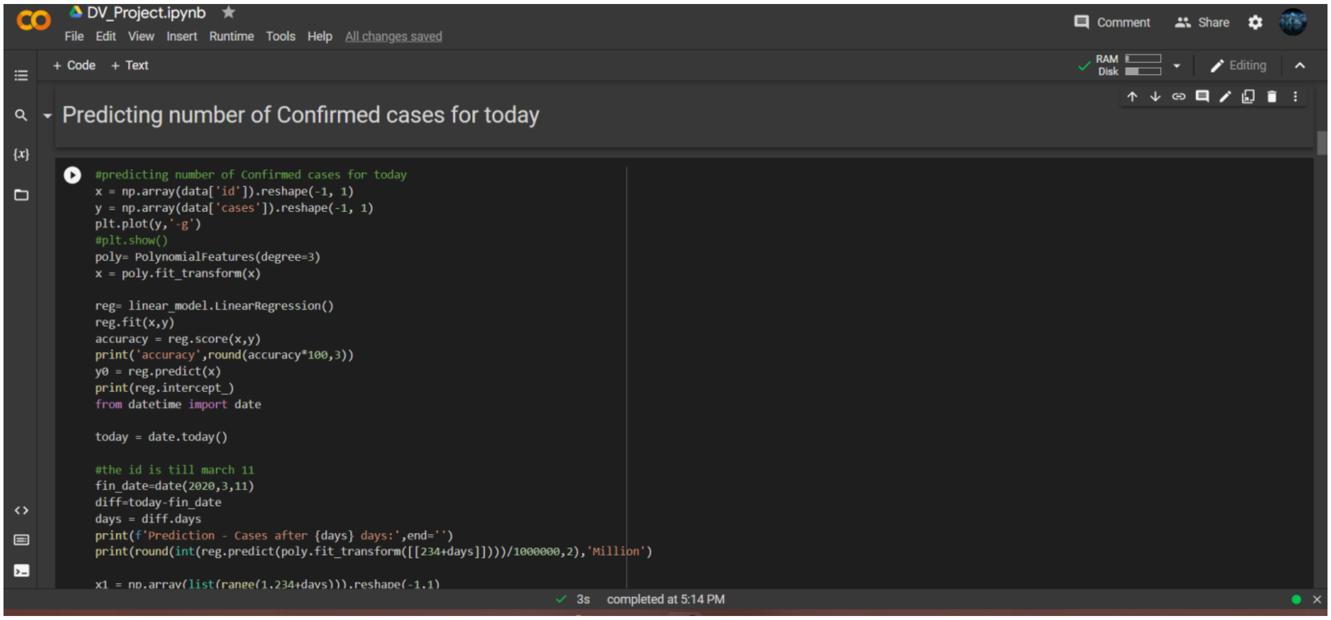
```
Mounted at /content/gdrive
```



Loading data right from the source:

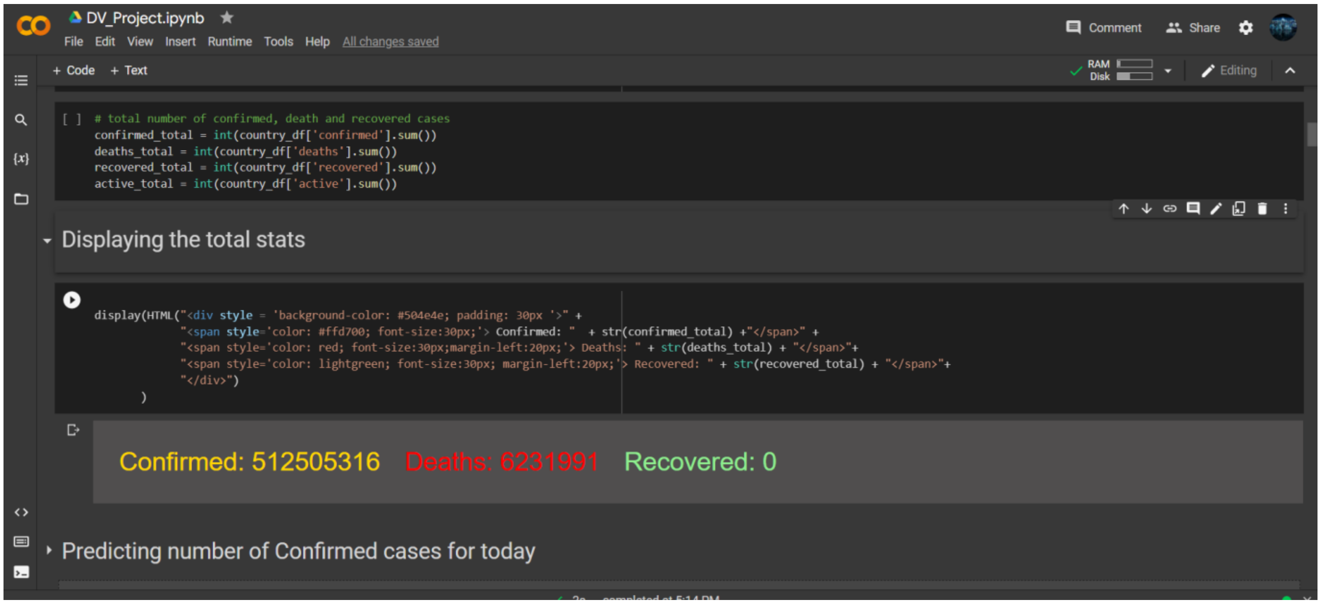
```
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_global.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')
data = pd.read_csv('/content/gdrive/My Drive/DV DATASET/coronaCases.csv')
corona_dataset_csv = pd.read_csv('/content/gdrive/My Drive/DV DATASET/covid19_Confirmed_dataset.csv')
world_happiness_report = pd.read_csv("/content/gdrive/My Drive/DV DATASET/worldwide_happiness_report.csv")
corona_dataset_csv.drop(['Lat','Long'],axis=1,inplace=True)

data = data[['id','cases']]
# confirmed_df.head()
# recovered_df.head()
# death_df.head()
# country_df.head()
```

```
# data cleaning

# renaming the df column names to lowercase
country_df.columns = map(str.lower, country_df.columns)
confirmed_df.columns = map(str.lower, confirmed_df.columns)
death_df.columns = map(str.lower, death_df.columns)
recovered_df.columns = map(str.lower, recovered_df.columns)

# changing province/state to state and country/region to country
confirmed_df = confirmed_df.rename(columns={'province/state': 'state', 'country/region': 'country'})
```



Predicting number of Confirmed cases for today

```
#predicting number of Confirmed cases for today
x = np.array(data['id']).reshape(-1, 1)
y = np.array(data['cases']).reshape(-1, 1)
plt.plot(y,'-g')
#plt.show()
poly= PolynomialFeatures(degree=3)
x = poly.fit_transform(x)

reg= linear_model.LinearRegression()
reg.fit(x,y)
accuracy = reg.score(x,y)
print('accuracy',round(accuracy*100,3))
y0 = reg.predict(x)
print(reg.intercept_)
from datetime import date

today = date.today()

#the id is till march 11
fin_date=date(2020,3,11)
diff=today-fin_date
days = diff.days
print(f'Prediction - Cases after {days} days:',end='')
print(round(int(reg.predict(poly.fit_transform([[234+days]])))/1000000,2),'Million')

x1 = np.array(list(range(1,234+days))).reshape(-1,1)
```



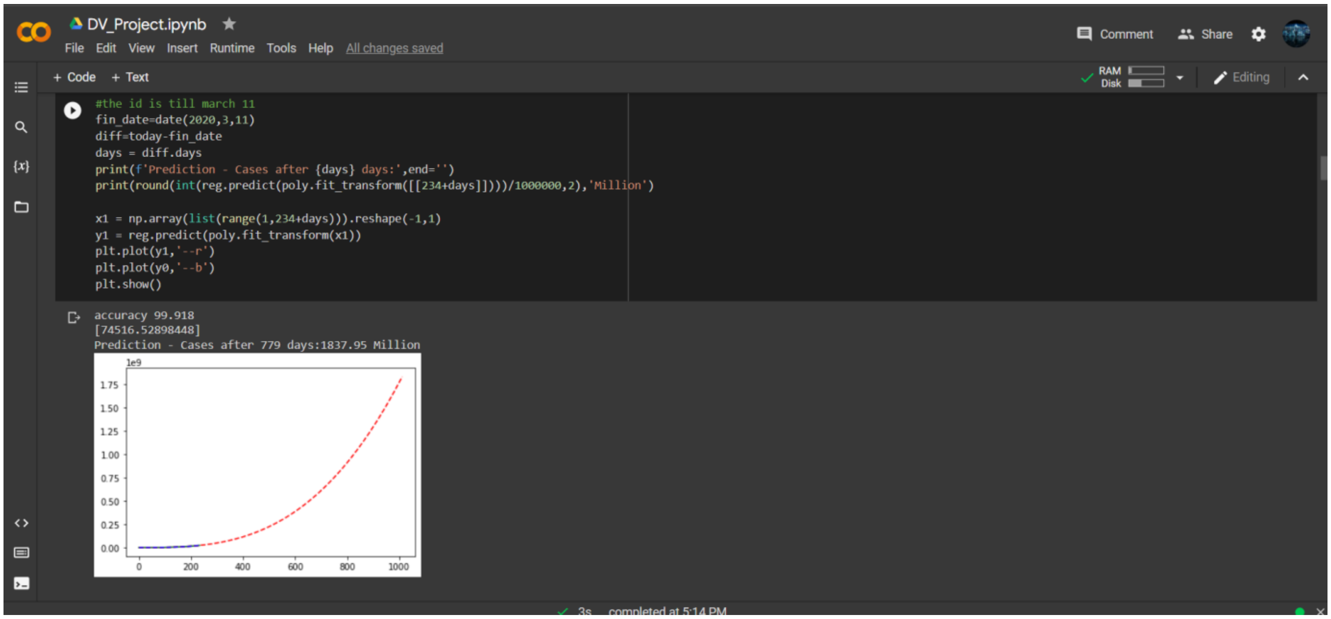
```
[ ] # total number of confirmed, death and recovered cases
    confirmed_total = int(country_df['confirmed'].sum())
    deaths_total = int(country_df['deaths'].sum())
    recovered_total = int(country_df['recovered'].sum())
    active_total = int(country_df['active'].sum())
```

Displaying the total stats

```
display(HTML("<div style = 'background-color: #504e4e; padding: 30px '>" +
             "<span style='color: #ffd700; font-size:30px;'> Confirmed: "  + str(confirmed_total) +"</span>" +
             "<span style='color: red; font-size:30px;margin-left:20px;'> Deaths: " + str(deaths_total) + "</span>"+
             "<span style='color: lightgreen; font-size:30px; margin-left:20px;'> Recovered: " + str(recovered_total) + "</span>"+
             "</div>")
       )
```

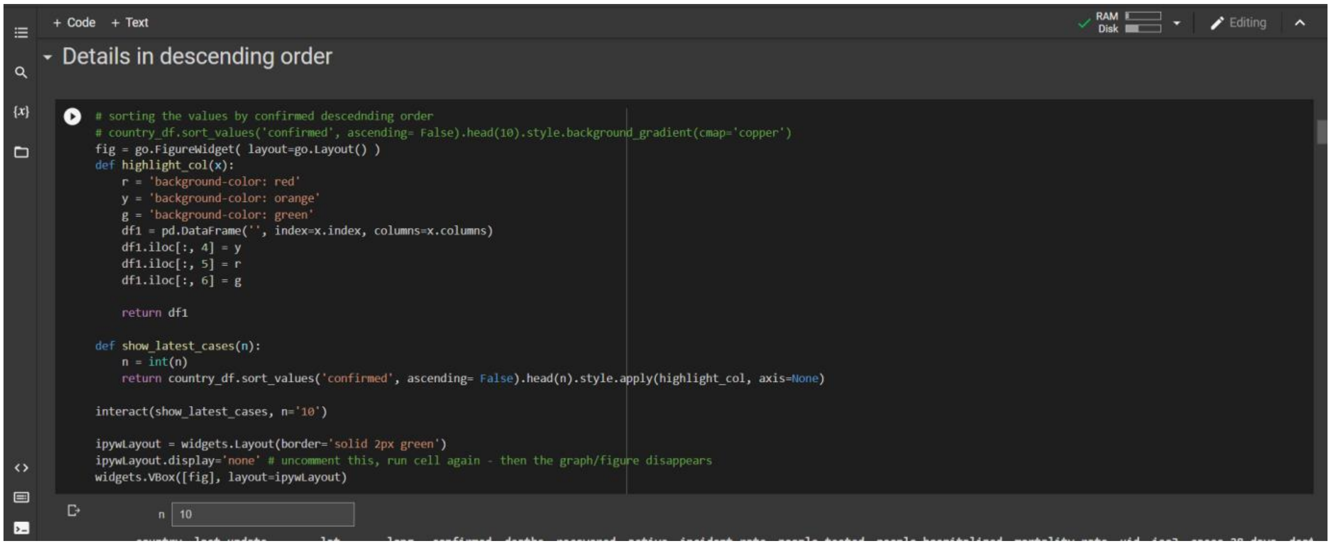
Confirmed: 512505316 Deaths: 6231991 Recovered: 0

Predicting number of Confirmed cases for today



```
#the id is till march 11
fin_date=date(2020,3,11)
diff=today-fin_date
days = diff.days
print(f'Prediction - Cases after {days} days:',end='')
print(round(int(reg.predict(poly.fit_transform([[234+days]])))/1000000,2),'Million')

x1 = np.array(list(range(1,234+days))).reshape(-1,1)
y1 = reg.predict(poly.fit_transform(x1))
plt.plot(y1,'--r')
plt.plot(y0,'--b')
plt.show()
```

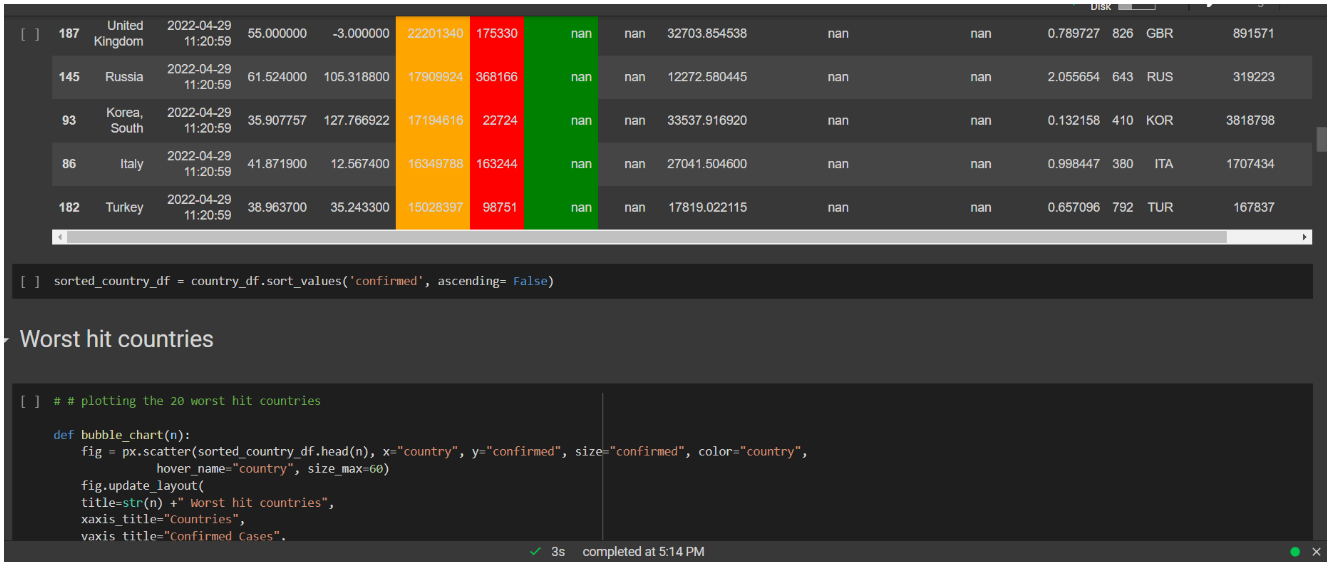
```
accuracy 99.918
[74516.52898448]
Prediction - Cases after 779 days:1837.95 Million
```



Details in descending order

```
# sorting the values by confirmed descednding order
# country_df.sort_values('confirmed', ascending= False).head(10).style.background_gradient(cmap='copper')
fig = go.FigureWidget( layout=go.Layout() )
def highlight_col(x):
    r = 'background-color: red'
    y = 'background-color: orange'
    g = 'background-color: green'
    df1 = pd.DataFrame('', index=x.index, columns=x.columns)
    df1.iloc[:, 4] = y
    df1.iloc[:, 5] = r
    df1.iloc[:, 6] = g

    return df1

def show_latest_cases(n):
    n = int(n)
    return country_df.sort_values('confirmed', ascending= False).head(n).style.apply(highlight_col, axis=None)

interact(show_latest_cases, n='10')

ipywLayout = widgets.Layout(border='solid 2px green')
ipywLayout.display='none' # uncomment this, run cell again - then the graph/figure disappears
widgets.VBox([fig], layout=ipywLayout)
```

n 10

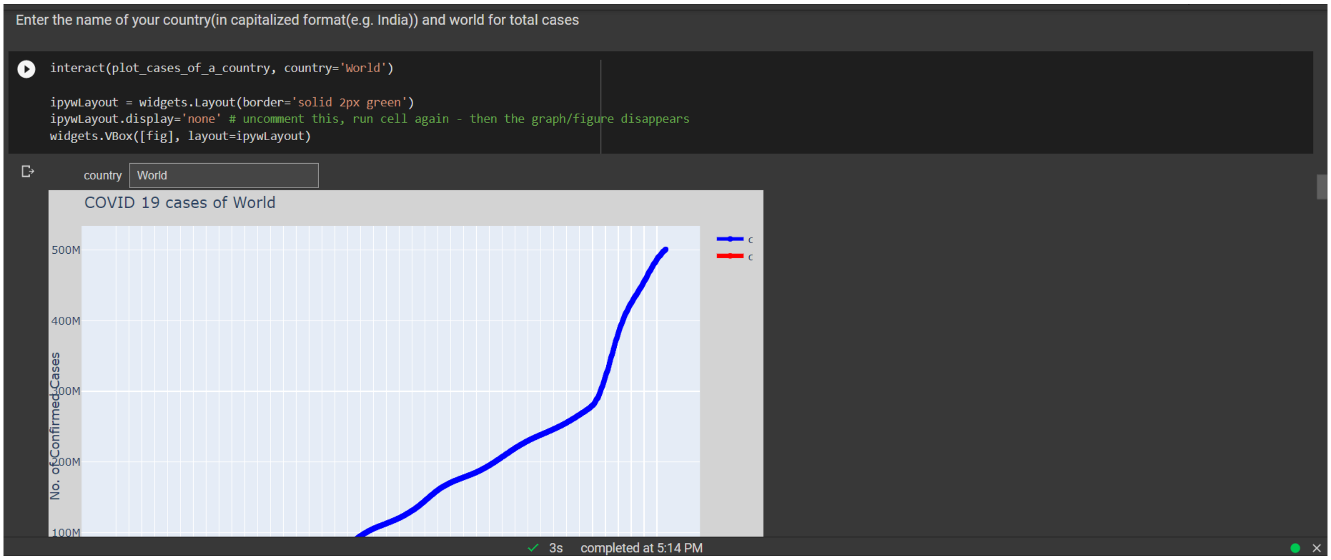


187	United Kingdom	2022-04-29 11:20:59	55.000000	-3.000000	22201340	175330	nan	nan	32703.854538	nan	nan	0.789727	826	GBR	891571
145	Russia	2022-04-29 11:20:59	61.524000	105.318800	17909924	368166	nan	nan	12272.580445	nan	nan	2.055654	643	RUS	319223
93	Korea, South	2022-04-29 11:20:59	35.907757	127.766922	17194616	22724	nan	nan	33537.916920	nan	nan	0.132158	410	KOR	3818798
86	Italy	2022-04-29 11:20:59	41.871900	12.567400	16349788	163244	nan	nan	27041.504600	nan	nan	0.998447	380	ITA	1707434
182	Turkey	2022-04-29 11:20:59	38.963700	35.243300	15028397	98751	nan	nan	17819.022115	nan	nan	0.657096	792	TUR	167837

```
sorted_country_df = country_df.sort_values('confirmed', ascending= False)
```

Worst hit countries

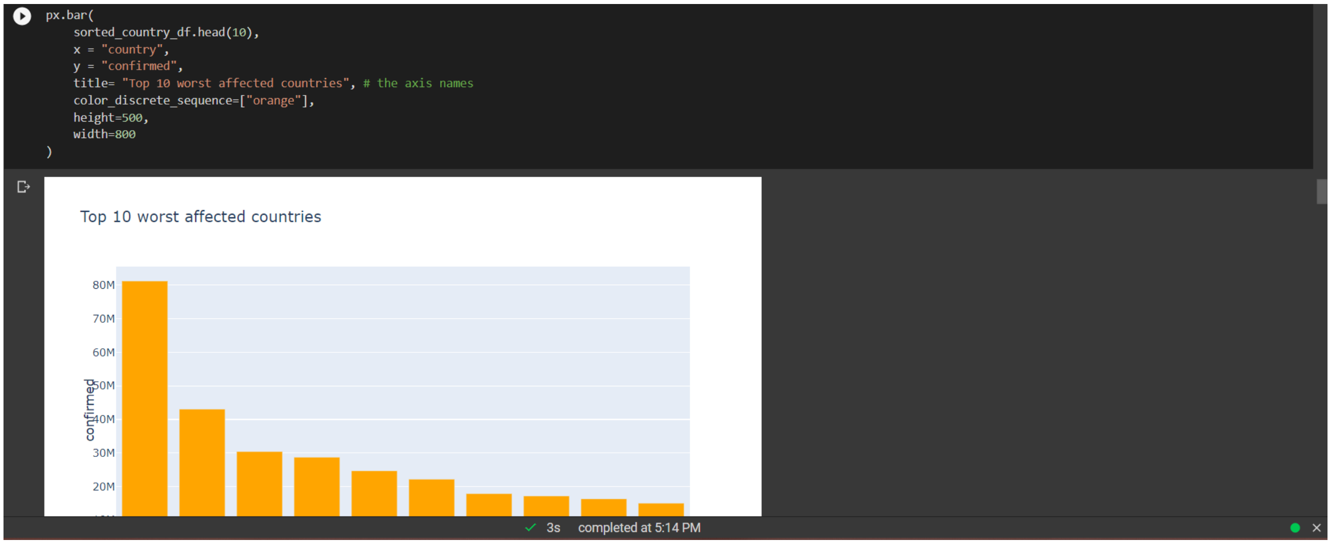
```
# # plotting the 20 worst hit countries

def bubble_chart(n):
    fig = px.scatter(sorted_country_df.head(n), x="country", y="confirmed", size="confirmed", color="country",
               hover_name="country", size_max=60)
    fig.update_layout(
    title=str(n) +" Worst hit countries",
    xaxis_title="Countries",
    yaxis_title="Confirmed Cases",
```



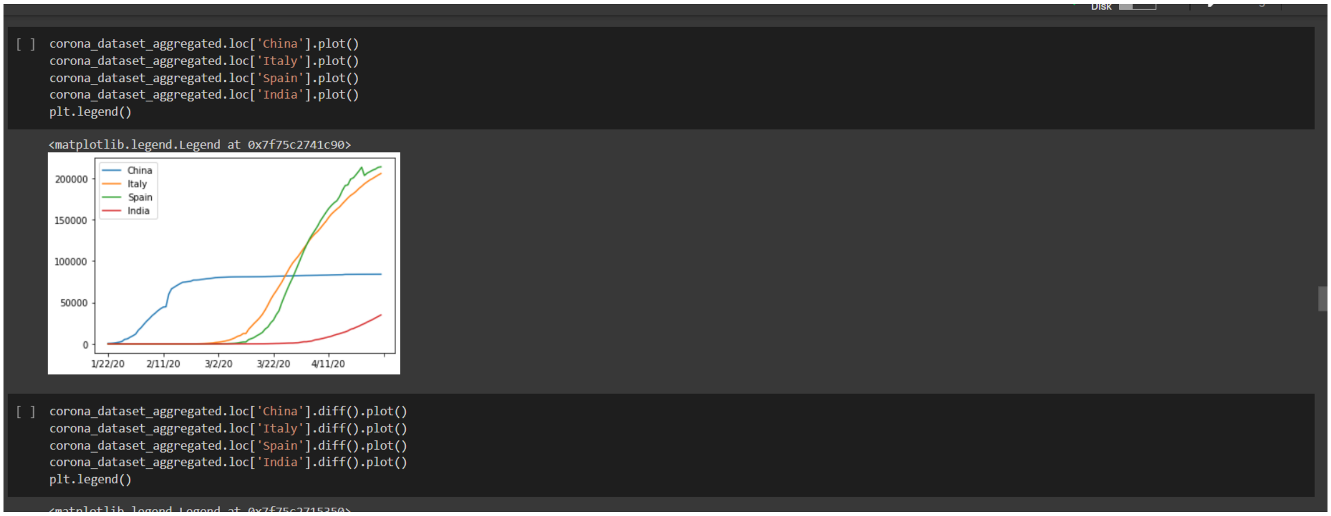
Enter the name of your country(in capitalized format(e.g. India)) and world for total cases

```
interact(plot_cases_of_a_country, country='World')

ipywLayout = widgets.Layout(border='solid 2px green')
ipywLayout.display='none' # uncomment this, run cell again - then the graph/figure disappears
widgets.VBox([fig], layout=ipywLayout)
```

country World



```
px.bar(
    sorted_country_df.head(10),
    x = "country",
    y = "confirmed",
    title= "Top 10 worst affected countries", # the axis names
    color_discrete_sequence=["orange"],
    height=500,
    width=800
)
```

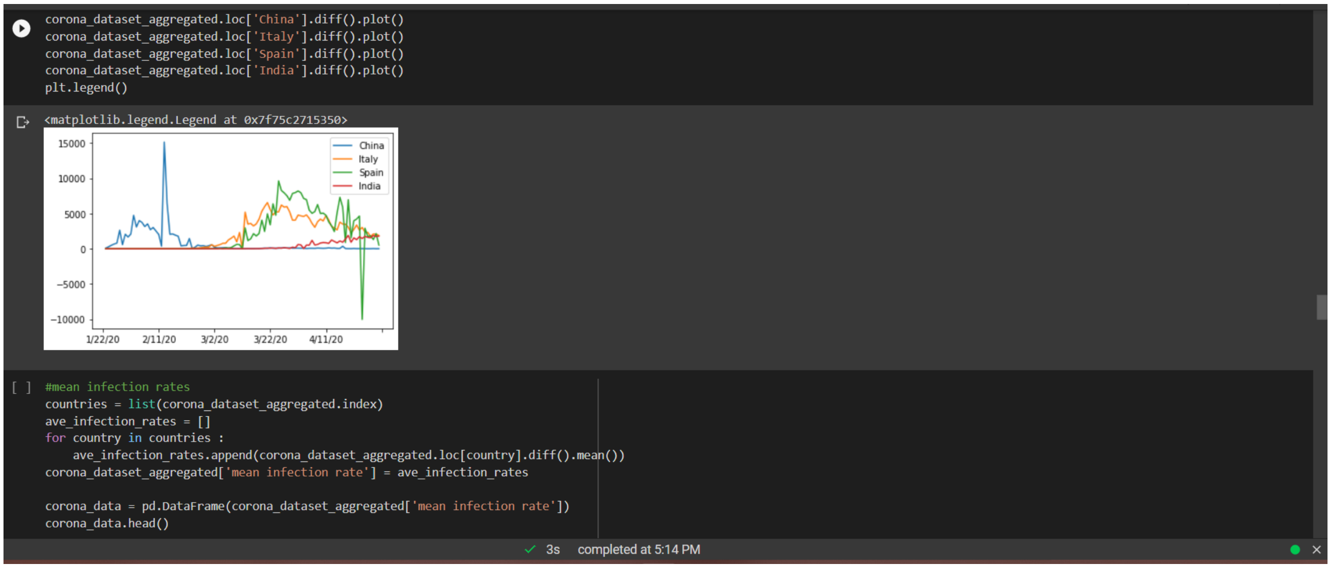


```
[ ]  corona_dataset_aggregated.loc['China'].plot()
     corona_dataset_aggregated.loc['Italy'].plot()
     corona_dataset_aggregated.loc['Spain'].plot()
     corona_dataset_aggregated.loc['India'].plot()
     plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f75c2741c90>
```

```
[ ]  corona_dataset_aggregated.loc['China'].diff().plot()
     corona_dataset_aggregated.loc['Italy'].diff().plot()
     corona_dataset_aggregated.loc['Spain'].diff().plot()
     corona_dataset_aggregated.loc['India'].diff().plot()
     plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f75c2715350>
```

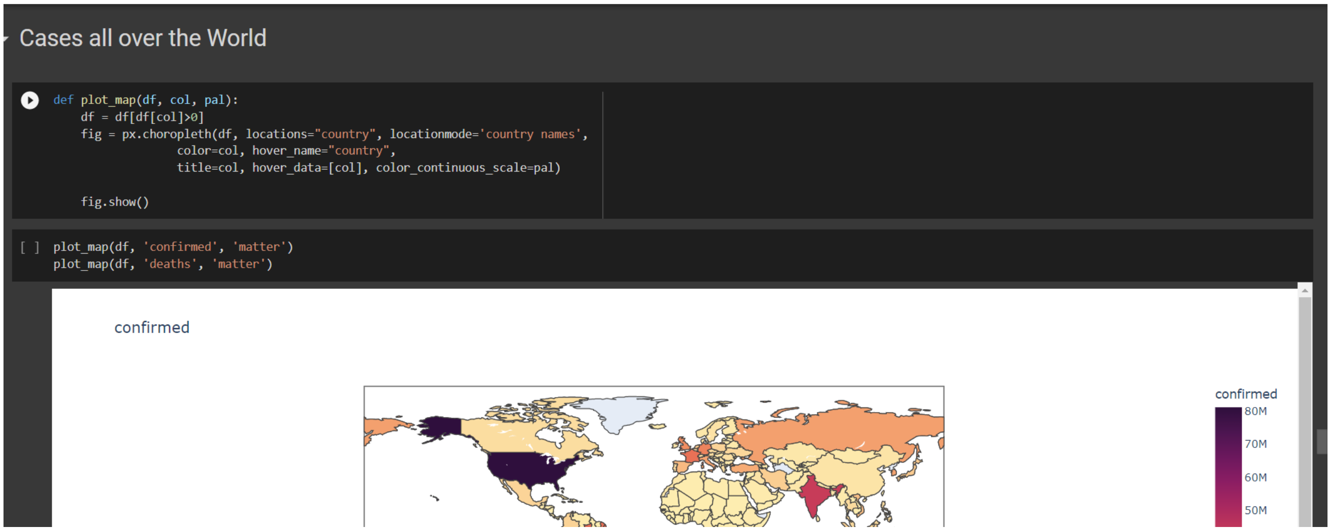


```
corona_dataset_aggregated.loc['China'].diff().plot()
corona_dataset_aggregated.loc['Italy'].diff().plot()
corona_dataset_aggregated.loc['Spain'].diff().plot()
corona_dataset_aggregated.loc['India'].diff().plot()
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f75c2715350>
```

```
[ ]  #mean infection rates
     countries = list(corona_dataset_aggregated.index)
     ave_infection_rates = []
     for country in countries :
         ave_infection_rates.append(corona_dataset_aggregated.loc[country].diff().mean())
     corona_dataset_aggregated['mean infection rate'] = ave_infection_rates

     corona_data = pd.DataFrame(corona_dataset_aggregated['mean infection rate'])
     corona_data.head()
```

✓ 3s completed at 5:14 PM

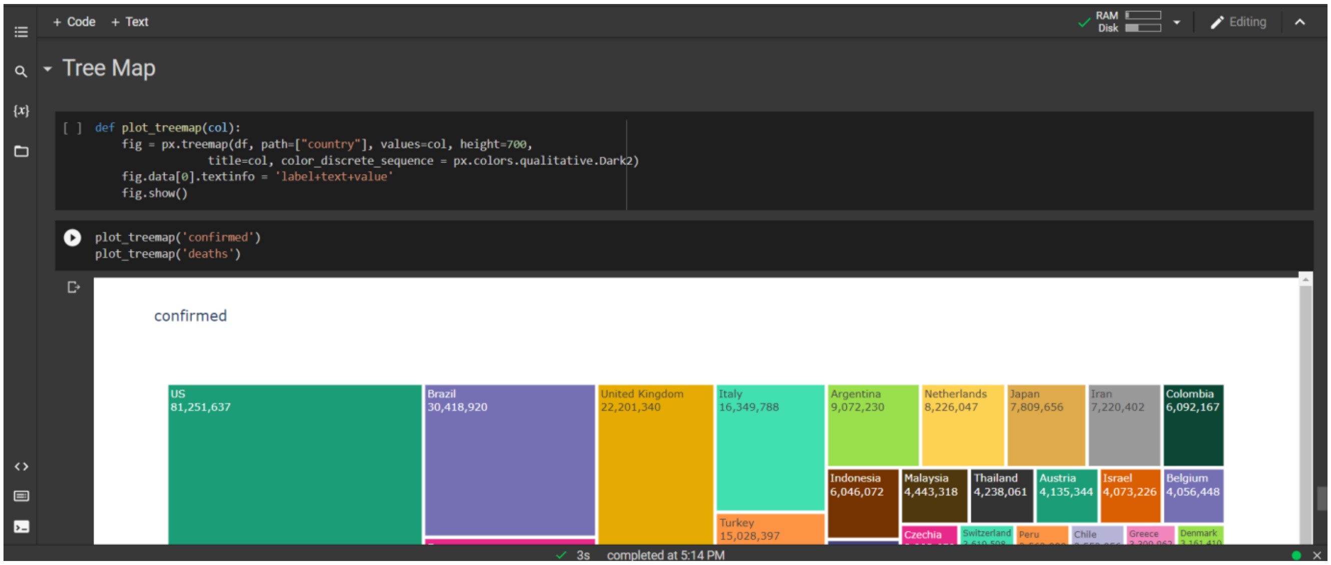


Cases all over the World

```
def plot_map(df, col, pal):
    df = df[df[col]>0]
    fig = px.choropleth(df, locations="country", locationmode='country names',
                        color=col, hover_name="country",
                        title=col, hover_data=[col], color_continuous_scale=pal)

    fig.show()
```

```
[ ] plot_map(df, 'confirmed', 'matter')
    plot_map(df, 'deaths', 'matter')
```

confirmed



Tree Map

```
[ ] def plot_treemap(col):
        fig = px.treemap(df, path=["country"], values=col, height=700,
                     title=col, color_discrete_sequence = px.colors.qualitative.Dark2)
        fig.data[0].textinfo = 'label+text+value'
        fig.show()
```

```
plot_treemap('confirmed')
plot_treemap('deaths')
```

confirmed

✓ 3s completed at 5:14 PM