

Covid-19 Data Exploration Using Python

Here I installed the Covid module along with cufflinks. Covid here is python package for getting data and information regarding corona virus provided by John Hopkins University and Worldometers.info. Cufflinks connects Plotly with Pandas to create graphs of Dataframes directly.

Installing Covid Module

```
In [1]: !pip install covid
!pip install cufflinks
```

```
Requirement already satisfied: covid in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: requests in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: pydantic in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: beautifulsoup4 in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: typer in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: soupsieve>1.2 in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sambhav\anaconda3\lib\site-packages
Requirement already satisfied: idna<3,>=2.5 in c:\users\sambhav\anaconda3\lib\site-packages
```

Here I have displayed the Dataframe in Heatmap style for better visualisation. Now it is clearer by simply looking at the intensity of the colour rather than to scroll down and look at each numerical value.

CMap for the Dataset for better visualisation

```
In [4]: df.style.background_gradient(cmap='Reds')
```

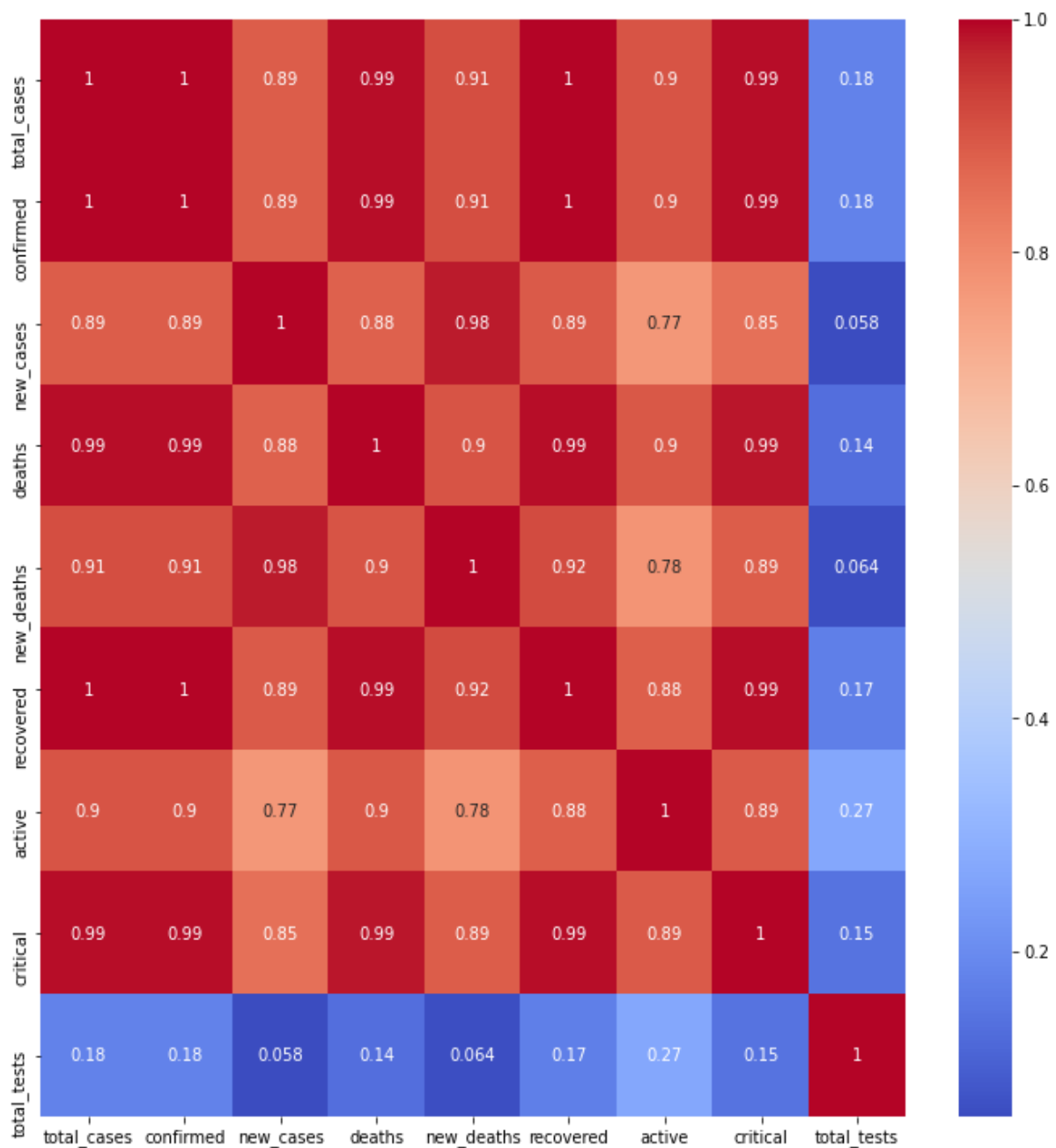
Out[4]:

	country	total_cases	confirmed	new_cases	deaths	new_deaths	recovered	active	active_cases	critical	total_tests	total_tests_p
0	North America	56188744	56188744	10198	1147015	373	44806233	10235496	10235496	19576	0	
1	Asia	79244152	79244152	58728	1169703	1478	76292797	1781652	1781652	26939	0	
2	South America	38389806	38389806	2142	1169971	14	36549924	669911	669911	12335	0	
3	Europe	64310624	64310624	209066	1299759	2771	57923467	5087398	5087398	13203	0	
4	Africa	8567137	8567137	282	218522	8	7902779	445836	445836	2264	0	
5	Oceania	308526	308526	1789	3699	14	251921	52906	52906	324	0	
6	0	721	721	0	15	0	706	0	0	0	0	
7	World	247009710	247009710	282205	5008684	4658	223727827	18273199	18273199	74641	0	
8	USA	46776345	46776345	4366	765757	35	36646900	9363688	9363688	12716	697847798	
9	India	34272260	34272260	12541	458213	440	33647055	166992	166992	8944	607062619	

Correlation Matrix

```
In [8]: plt.figure(figsize=(12,12))  
sns.heatmap(df_1.corr(), annot=True, cmap='coolwarm')
```

```
Out[8]: <AxesSubplot:>
```

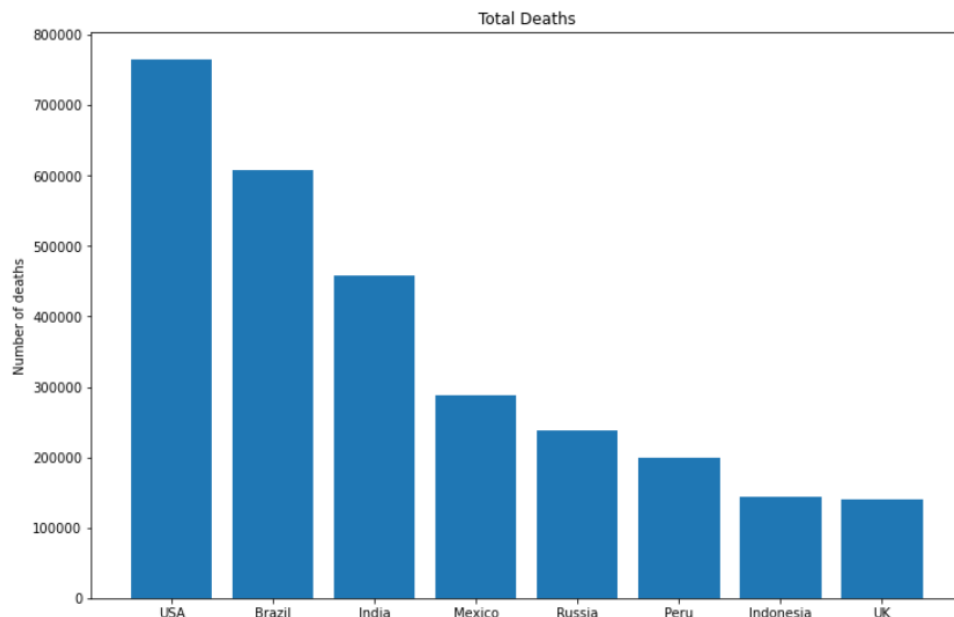


Here I have plotted the bar graph of total deaths in different countries. As we can see it is a very simple graph and we can't really get much information by looking to it.

Visualisation using Matplotlib

```
In [11]: df_1 = df[8:].sort_values(by='deaths',ascending=False)[0:8].reset_index()  
plt.bar(df_1['country'], df_1['deaths'], )  
plt.title("Total Deaths")  
plt.xlabel("Countries")  
plt.ylabel("Number of deaths")
```

```
Out[11]: Text(0, 0.5, 'Number of deaths')
```

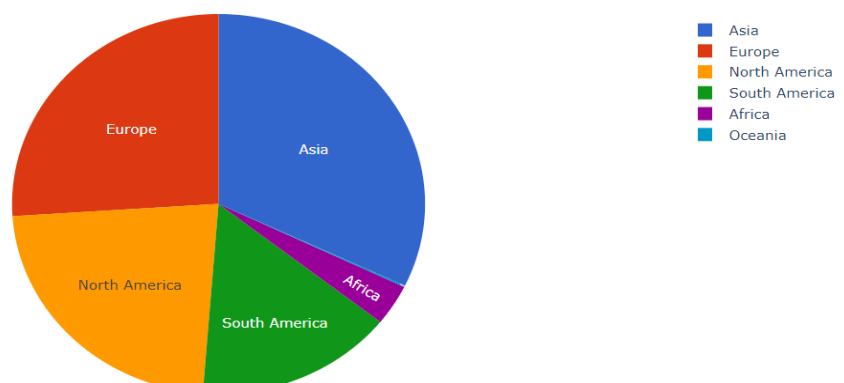


So, I used Plotly instead of Matplotlib for a much more sophisticated data visualisation. Plotly is great for creating plots more efficiently and elaborately. Here is a pie chart of comparing confirmed cases in different continents.

Pie Charts using Plotly

```
[12]: fig = px.pie(df[0:6], values='confirmed', names='country', title='Confirmed Cases', color_discrete_sequence =px.colors.qualitative  
fig.update_traces(textposition='inside', textinfo='label')  
fig.show()
```

Confirmed Cases

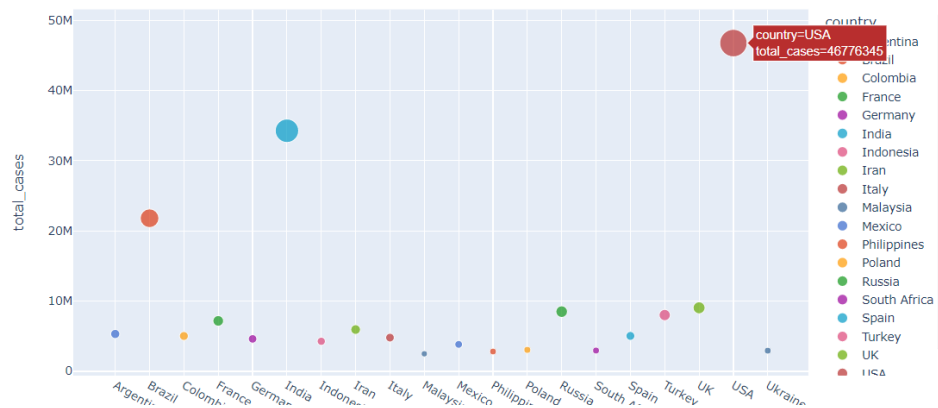


Here it is a scatter plot of total cases in different countries. As we can see that here the size of the circle in scatter plot gives an idea about number of total cases. On hovering the mouse pointer over the circle, it gives the number of total cases along with that country associated with the cases.

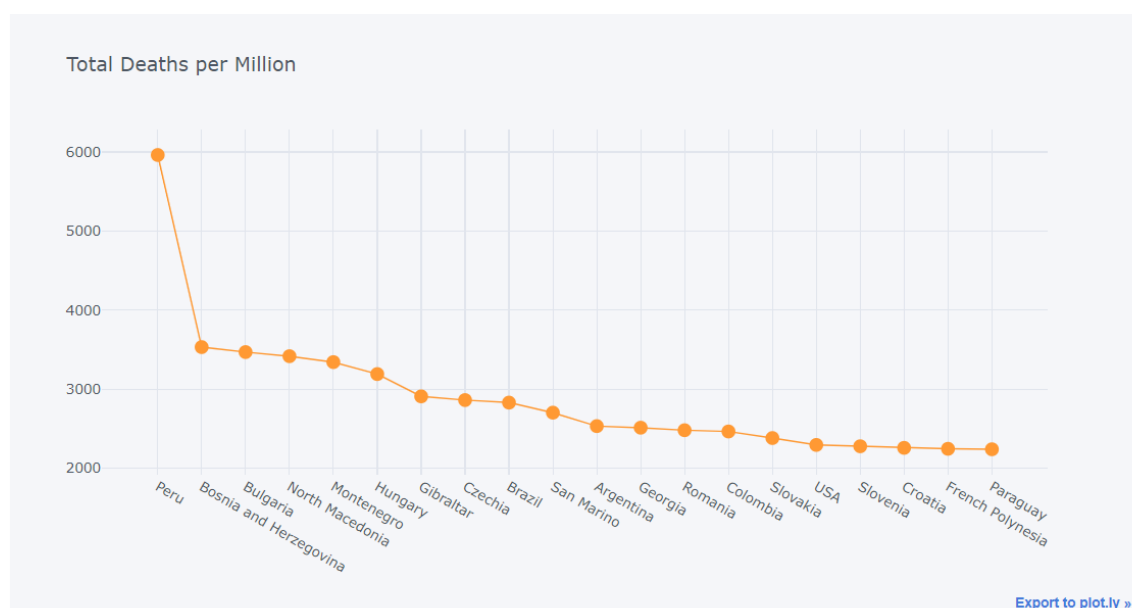
Scatter Plot

```
In [14]: # scatter-plot
df_1 = df[8:].sort_values(by='total_cases',ascending=False)[0:20].reset_index()
df_1 = pd.DataFrame(df_1.groupby('country')['total_cases'].sum())

fig = px.scatter(df_1, x = df_1.index, y = 'total_cases', size = 'total_cases', color_discrete_sequence =px.colors.qualitative.G
fig.show()
```



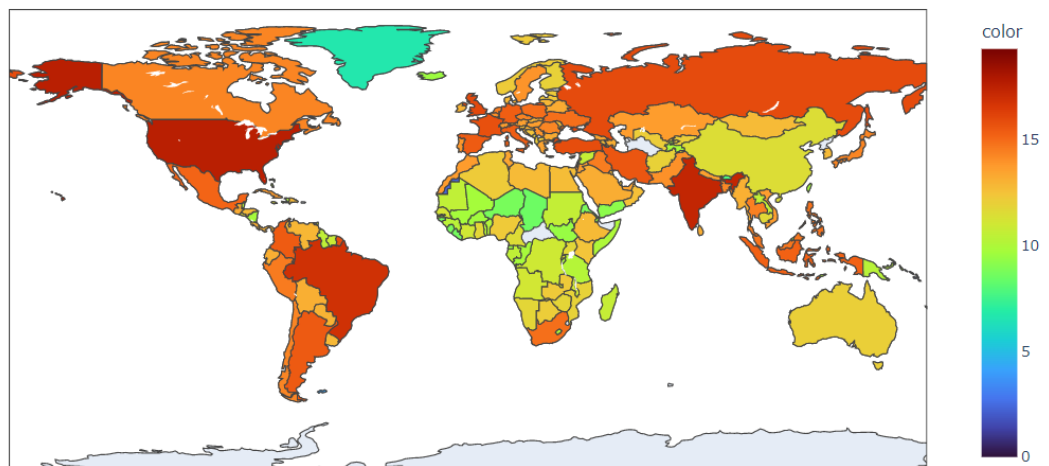
This is a scatter plot with marker and lines. It is the graph of total deaths per million in different countries. The country with relatively less population has more death per million as per the graph.



A Choropleth Map is a map composed of different coloured polygons and used for visual variation with help of colour to show the difference in the intensity of particular thing. Here the colour scale is in log scale for better visualisation. Red shows areas with a greater number of total cases and blue shows least.

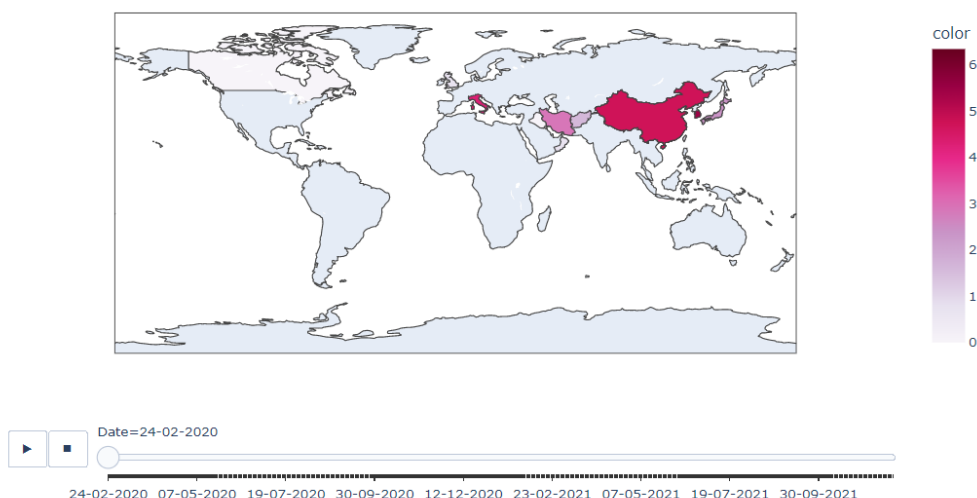
Visualising Total Cases with Choropleth Map (Color Value is in log scale for better visualisation)

```
5]: fig = px.choropleth(df, locations='country', color=np.log(df['total_cases']), locationmode='country names', color_continuous_sca
fig.show()
```



It is a similar kind of choropleth map but it also has a slider to change dates and see the Daily confirmed cases in different countries. Red shows a greater number of daily cases while white shows least.

```
In [26]: fig = px.choropleth(daywise, locations='Country', color = np.log(daywise["Daily new confirmed cases of COVID-19"]), animation_fr
fig.show()
```



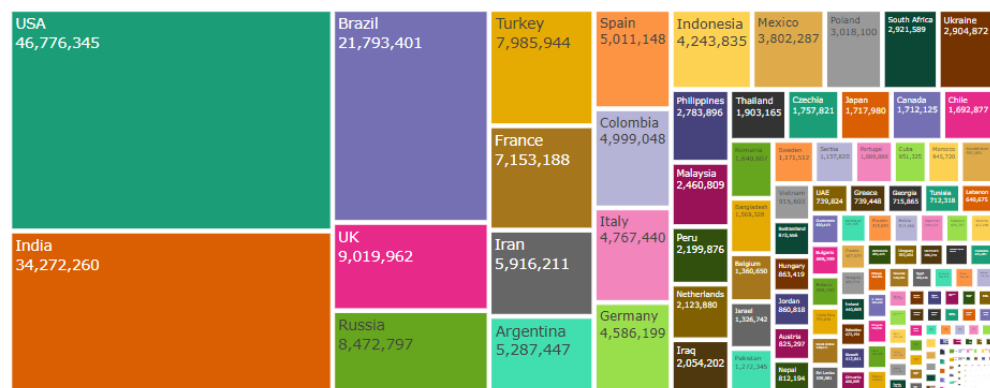
Treemap is used for providing a stratified view of data and produces easy to see figures. The branches of tree are characterized by rectangles and each sub branch is shown by a smaller rectangle.

Tree Map Analysis

```
In [27]: fig = px.treemap(df[8:].sort_values(by = 'total_cases', ascending = False).reset_index(drop = True),
                        path = ['country'], values = 'total_cases',
                        title = 'Total Number of Cases',
                        color_discrete_sequence = px.colors.qualitative.Dark2)

fig.data[0].textinfo = 'label+text+value'
fig.show()
```

Total Number of Cases



Covid-19 Confirmed Cases and Deaths uptill now



This is a dashboard kind of thing which gives the stats of Covid-19 Worldwide from the beginning of this epidemic.

Total Confirmed Cases

247M

Total Recovered

223.7M

Total Deaths

5.01M

Total Active Cases

18.27M

Sambhav Agrawal

19264

Data Science Engineering

