

# CoronaVirus Summary Data-

## by Tobi Adedeji

### Analysis process and visualizations

This Project demonstrate how to use Python lib in Jupyter for Analysis and Visualizations

Link to data: [../input/covid19-global-dataset/worldometer\\_coronavirus\\_summary\\_data.csv](https://input/covid19-global-dataset/worldometer_coronavirus_summary_data.csv)

First thing i did was to import Python libraries to read the csv file using pandas:

```
[13]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from datetime import datetime
import plotly.express as px
import plotly.graph_objects as go
import geopandas as gpd
from plotly.subplots import make_subplots
import matplotlib.pyplot as plt
```

```
[14]: data_2 = pd.read_csv("../input/covid19-global-dataset/worldometer_coronavirus_summary_data.csv")
data_2
```

```
[14]:
```

	country	continent	total_confirmed	total_deaths	total_recovered	active_cases	serious_or_critical	total_cases_per_1m_population	total_deaths_per_1m_population	total_tests	total_tests_per_1m_population
0	Afghanistan	Asia	179267	7690.0	162202.0	9375.0	1124.0	4420	190.0	951337.0	23455.0
1	Albania	Europe	275574	3497.0	271826.0	251.0	2.0	95954	1218.0	1817530.0	632857.0
2	Algeria	Africa	265816	6875.0	178371.0	80570.0	6.0	5865	152.0	230861.0	5093.0
3	Andorra	Europe	42156	153.0	41021.0	982.0	14.0	543983	1974.0	249838.0	3223924.0
4	Angola	Africa	99194	1900.0	97149.0	145.0	NaN	2853	55.0	1499795.0	43136.0
...	...	...	...	...	...	...	...	...	...	...	...
221	Wallis And Futuna Islands	Australia/Oceania	454	7.0	438.0	9.0	NaN	41755	644.0	20508.0	1886140.0
222	Western Sahara	Africa	10	1.0	9.0	0.0	NaN	16	2.0	NaN	NaN
223	Yemen	Asia	11819	2149.0	9009.0	661.0	23.0	381	69.0	265253.0	8543.0
224	Zambia	Africa	320591	3983.0	315997.0	611.0	NaN	16575	206.0	3452554.0	178497.0
225	Zimbabwe	Africa	249206	5482.0	242417.0	1307.0	12.0	16324	359.0	2287793.0	149863.0

226 rows × 12 columns

Next, I replaced the empty values with Na values with 0:

```
1: data_2 = data_2.fillna(0)
data_2
```

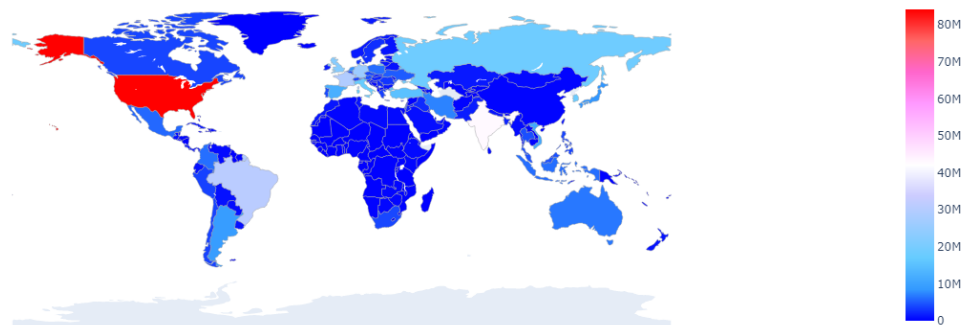
	country	continent	total_confirmed	total_deaths	total_recovered	active_cases	serious_or_critical	total_cases_per_1m_population	total_deaths_per_1m_population	total_tests	total_tests_per_1m_population
0	Afghanistan	Asia	179267	7690.0	162202.0	9375.0	1124.0	4420	190.0	951337.0	23455.0
1	Albania	Europe	275574	3497.0	271826.0	251.0	2.0	95954	1218.0	1817530.0	632857.0
2	Algeria	Africa	265816	6875.0	178371.0	80570.0	6.0	5865	152.0	230861.0	5093.0
3	Andorra	Europe	42156	153.0	41021.0	982.0	14.0	543983	1974.0	249838.0	3223924.0
4	Angola	Africa	99194	1900.0	97149.0	145.0	0.0	2853	55.0	1499795.0	43136.0
...	...	...	...	...	...	...	...	...	...	...	...
221	Wallis And Futuna Islands	Australia/Oceania	454	7.0	438.0	9.0	0.0	41755	644.0	20508.0	1886140.0
222	Western Sahara	Africa	10	1.0	9.0	0.0	0.0	16	2.0	0.0	0.0
223	Yemen	Asia	11819	2149.0	9009.0	661.0	23.0	381	69.0	265253.0	8543.0
224	Zambia	Africa	320591	3983.0	315997.0	611.0	0.0	16575	206.0	3452554.0	178497.0
225	Zimbabwe	Africa	249206	5482.0	242417.0	1307.0	12.0	16324	359.0	2287793.0	149863.0

226 rows × 12 columns

```
[22]: fig = go.Figure(data=go.Choropleth(locations = data_2['country'],
locationmode='country names',
z = data_2['total_confirmed'],
text = data_2['country'],
#color_continuous_scale='reds',
#Interactive color scale/palette,
colorscale = 'picnic',
autocolorscale=False,
#reversescale=True,
marker_line_color='darkgray',
marker_line_width=0.5))

fig.update_layout(
    title_text='Total confirmed cases around the world',
    geo=dict(
        showframe=False,
        showcoastlines=False,
        projection_type='equiangular'))
fig.show()
```

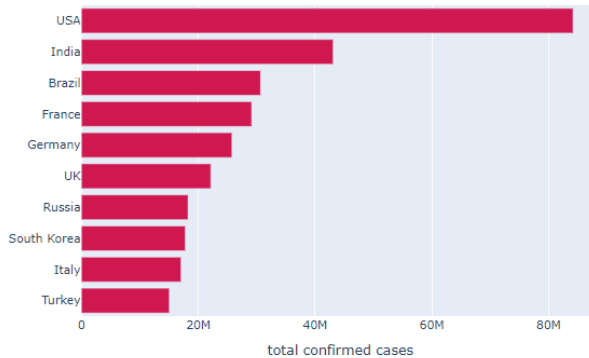
Total confirmed cases around the world



This Map represents the continent/Countries with the Highest Total confirmed covid cases around the world.

```
[23]: covid_top_10 = data_2[['country', 'total_confirmed']].sort_values('total_confirmed')[~10:]
fig = go.Figure(go.Bar(
    x=covid_top_10['total_confirmed'],
    y=covid_top_10['country'],
    marker_color = ['#CF1850']*10,
    orientation='h')
,layout=go.Layout(height=500, width=700))
fig.update_layout(title_text='Top 10 countries by total confirmed case',xaxis_title='total confirmed cases')
fig.show()
```

Top 10 countries by total confirmed case

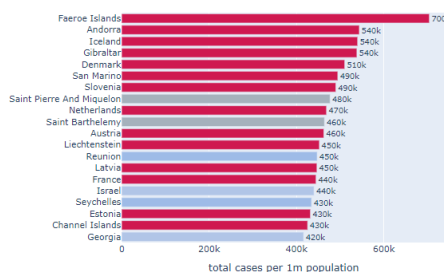


USA has the highest number of total confirmed cases which is 80m. The number is twice of any country on this list. Thus graph doesn't do justice to help me understand the actual stats for country hit by covid.

```
[24]: color_dict={'Europe': '#CF1850', 'North America': '#a5b1bc', 'Asia': '#b1c5e7', 'South America': '#d9dee7', 'Africa': '#9ebbe7', 'Australia/Oceania': '#979eea'}

covid_top_20_case_population = data_2[['country', 'total_cases_per_1m_population', 'continent']].sort_values('total_cases_per_1m_population')[~20:]
color_list = [color_dict[x] for x in covid_top_20_case_population['continent'].values]
fig = go.Figure(go.Bar(
    x=covid_top_20_case_population['total_cases_per_1m_population'],
    y=covid_top_20_case_population['country'],
    marker_color = color_list,
    text=covid_top_20_case_population['total_cases_per_1m_population'],
    orientation='h')
,layout=go.Layout(height=500, width=700))
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout(title_text='Top 20 countries by total cases per 1 million population',xaxis_title='total cases per 1m population')
fig.show()
```

Top 20 countries by total cases per 1 million population

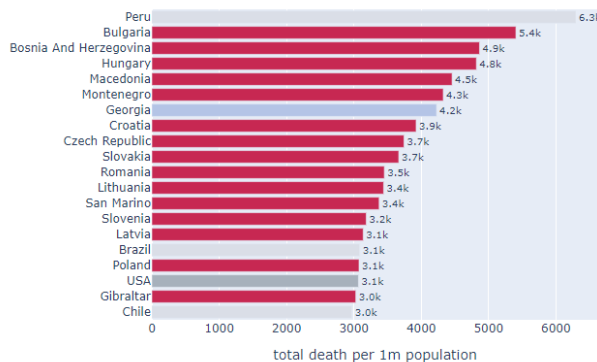


This Bar chart provides a better view of the covid data. This presents the total cases per 1 million population. The most countries with covid case per 1m are european countries.

```
[25]: color_dict={'Europe':'#CF1858','North America':'#a5b1bc','Asia':'#b1c5e7','South America':'#d9dee7','Africa':'#9ebbe7','Australia/Oceania':'#979eea'}

covid_top_20_death = data_2[['country','total_deaths_per_1m_population','continent']].sort_values('total_deaths_per_1m_population')[~20:]
color_list = [color_dict[x] for x in covid_top_20_death['continent'].values]
fig = go.Figure(go.Bar(
    x=covid_top_20_death['total_deaths_per_1m_population'],
    y=covid_top_20_death['country'],
    marker_color = color_list,
    text=covid_top_20_death['total_deaths_per_1m_population'],
    orientation='h'))
,layout=go.Layout(height=500, width=700))
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout(title_text='Top 20 countries by total deaths per 1 million population',xaxis_title='total death per 1m population')
fig.show()
```

Top 20 countries by total deaths per 1 million population

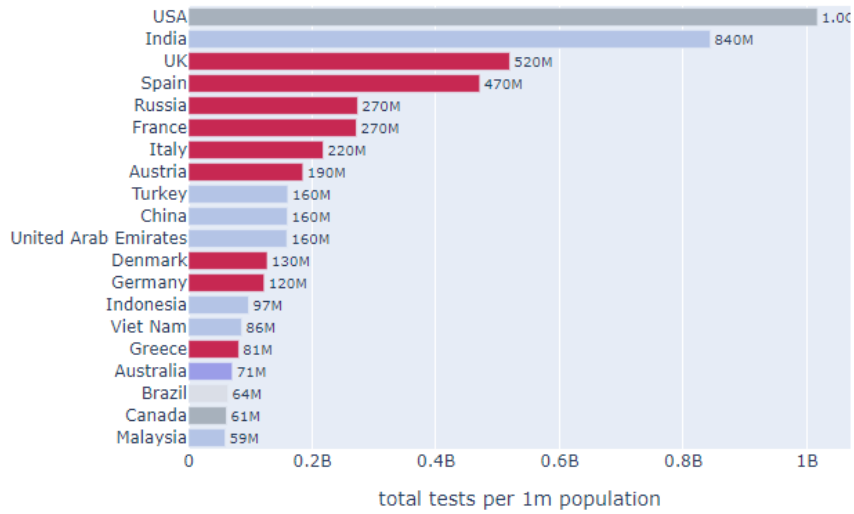


One would assume that the countries worst hit by covid would also record the most number of deaths, but this isn't the case. There are other factors like age-group, percentage of population with non - covid related illness, health care measures taken by the government like the rate of testing, public awareness and cooperation. We can see that this plot has more European countries than the one above it, and African countries do not even make this list.

```
[26]: covid_top_20_tests = data_2[['country','total_tests','continent']].dropna().sort_values('total_tests')[~20:]
color_list = [color_dict[x] for x in covid_top_20_tests['continent'].values]
fig = go.Figure(go.Bar(
    x=covid_top_20_tests['total_tests'],
    y=covid_top_20_tests['country'],
    text=covid_top_20_tests['total_tests'],
    marker_color= color_list,
    orientation='h'),layout=go.Layout(height=500, width=700))
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout(title_text='Top 20 countries by total tests per 1 million population',xaxis_title='total tests per 1m population')

fig.show()
```

Top 20 countries by total tests per 1 million population



According to the plot, USA has the most tests per 1 million. I'm surprised Peru did not make this list considering the fact that they have the most number of deaths per 1 million. Countries like Russia and France did not make the top 20 list for total deaths per 1 million, but they have about 270m test per 1 million population.