

Covid-19 Data Analysis

February 23, 2025

- Sauda Haywood
- DSC DSC680
- Week 9-12: Milestone 2

1 Covid-19 Data Analysis

1.1 Introduction:

For this project, my goal is to analyze global COVID-19 data and develop a function to automate the process.

1.2 Import Data

```
[2]: # Import required libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px

import warnings
warnings.filterwarnings('ignore')
```

```
[54]: # Load the dataset
data4 = pd.read_csv('full_grouped.csv')
# Display the first 5 rows
data4.head()
```

```
[54]:
```

	Date	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	\
0	2020-01-22	Afghanistan	0	0	0	0	0	
1	2020-01-22	Albania	0	0	0	0	0	
2	2020-01-22	Algeria	0	0	0	0	0	
3	2020-01-22	Andorra	0	0	0	0	0	
4	2020-01-22	Angola	0	0	0	0	0	

	New deaths	New recovered	WHO Region
0	0	0	Eastern Mediterranean

1	0	0	Europe
2	0	0	Africa
3	0	0	Europe
4	0	0	Africa

```
[55]: data4.info
```

```
[55]: <bound method DataFrame.info of
Deaths Recovered Active \
0 2020-01-22 Afghanistan 0 0 0 0
1 2020-01-22 Albania 0 0 0 0
2 2020-01-22 Algeria 0 0 0 0
3 2020-01-22 Andorra 0 0 0 0
4 2020-01-22 Angola 0 0 0 0
...
35151 2020-07-27 West Bank and Gaza 10621 78 3752 6791
35152 2020-07-27 Western Sahara 10 1 8 1
35153 2020-07-27 Yemen 1691 483 833 375
35154 2020-07-27 Zambia 4552 140 2815 1597
35155 2020-07-27 Zimbabwe 2704 36 542 2126
```

	New cases	New deaths	New recovered	WHO Region
0	0	0	0	Eastern Mediterranean
1	0	0	0	Europe
2	0	0	0	Africa
3	0	0	0	Europe
4	0	0	0	Africa
...
35151	152	2	0	Eastern Mediterranean
35152	0	0	0	Africa
35153	10	4	36	Eastern Mediterranean
35154	71	1	465	Africa
35155	192	2	24	Africa

```
[35156 rows x 10 columns]>
```

```
[11]: # Load the dataset
data6 = pd.read_csv('worldometer_data.csv')
# Display the first 5 rows
data6.head()
```

```
[11]: Country/Region    Continent    Population    TotalCases    NewCases \
0 USA North America 3.311981e+08 5032179 NaN
1 Brazil South America 2.127107e+08 2917562 NaN
2 India Asia 1.381345e+09 2025409 NaN
3 Russia Europe 1.459409e+08 871894 NaN
4 South Africa Africa 5.938157e+07 538184 NaN
```

	TotalDeaths	NewDeaths	TotalRecovered	NewRecovered	ActiveCases	\
0	162804.0	NaN	2576668.0	NaN	2292707.0	
1	98644.0	NaN	2047660.0	NaN	771258.0	
2	41638.0	NaN	1377384.0	NaN	606387.0	
3	14606.0	NaN	676357.0	NaN	180931.0	
4	9604.0	NaN	387316.0	NaN	141264.0	

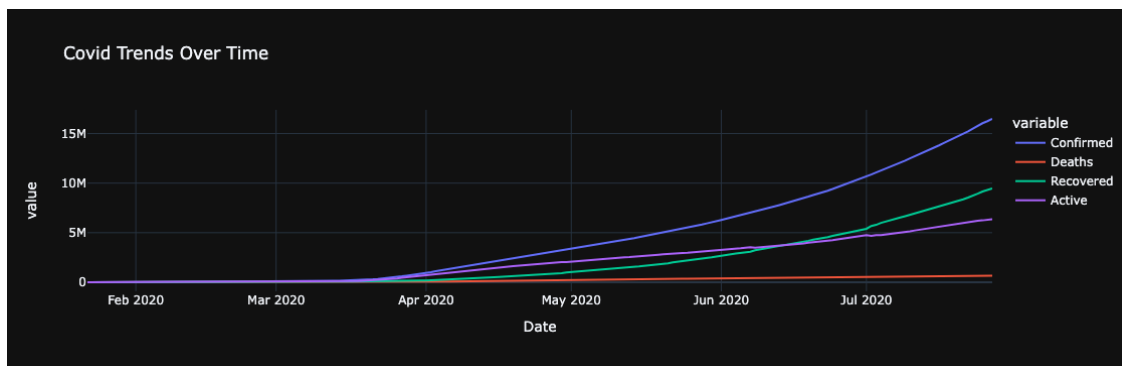
	Serious,Critical	Tot Cases/1M pop	Deaths/1M pop	TotalTests	\
0	18296.0	15194.0	492.0	63139605.0	
1	8318.0	13716.0	464.0	13206188.0	
2	8944.0	1466.0	30.0	22149351.0	
3	2300.0	5974.0	100.0	29716907.0	
4	539.0	9063.0	162.0	3149807.0	

	Tests/1M pop	WHO Region
0	190640.0	Americas
1	62085.0	Americas
2	16035.0	South-EastAsia
3	203623.0	Europe
4	53044.0	Africa

1.3 Data analysis and exploration

1.3.1 What is the global trend of Confirmed, Deaths, Recovered, and Active cases?

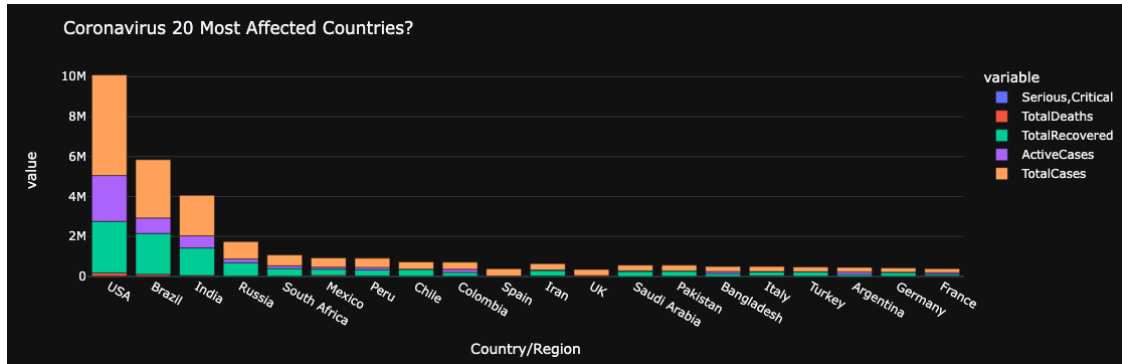
```
[58]: fig=px.  
      ↪line(data3,x="Date",y=["Confirmed","Deaths","Recovered","Active"],title="Covid_  
      ↪Trends Over Time",template="plotly_dark")  
      fig.show()
```



1.3.2 What are the 20 most effected countries?

```
[62]: fig=px.bar(data6.iloc[0:20],x='Country/
      ↪Region',y=['Serious,Critical','TotalDeaths','TotalRecovered','ActiveCases','TotalCases'],te

[63]: fig.update_layout({'title':"Coronavirus 20 Most Affected Countries?"})
      fig.show()
```

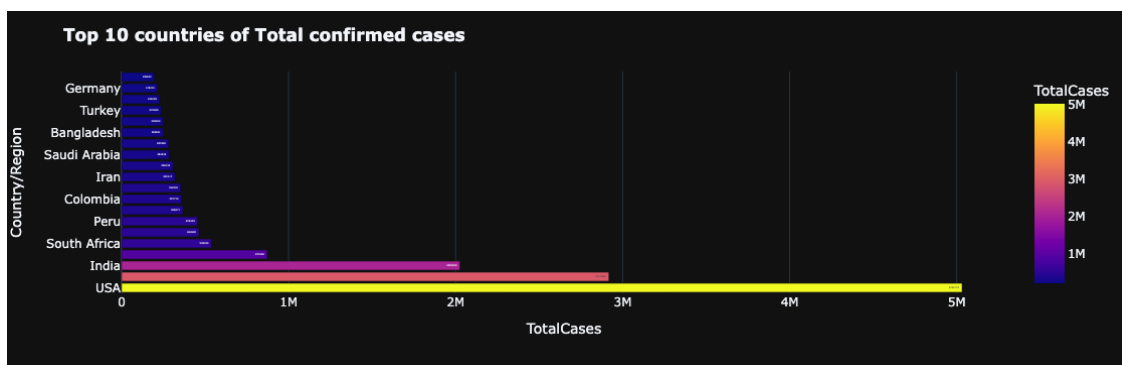


```
[27]: data6['Country/Region'].nunique()
```

[27]: 209

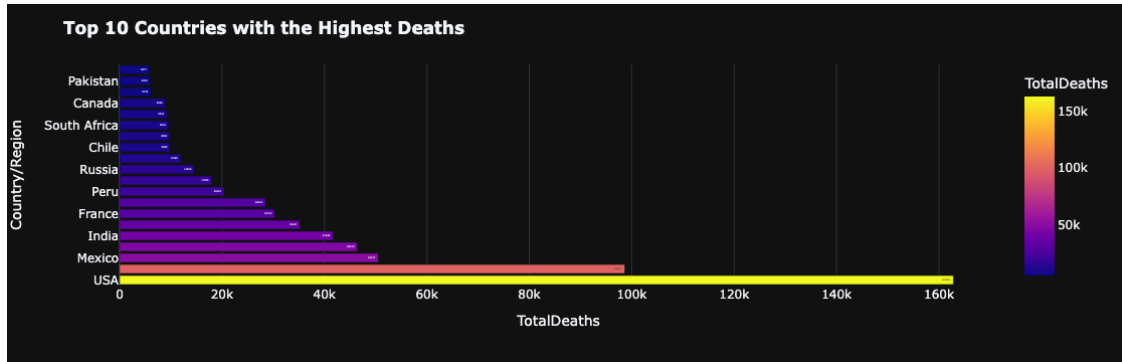
1.3.3 What are the Top 10 countries of Total Confirmed cases

```
[64]: fig=px.bar(data6.iloc[0:20],y='Country/
      ↪Region',x='TotalCases',color='TotalCases',text="TotalCases")
      fig.update_layout(template="plotly_dark",title_text="<b>Top 10 countries of_
      ↪Total confirmed cases</b>")
      fig.show()
```



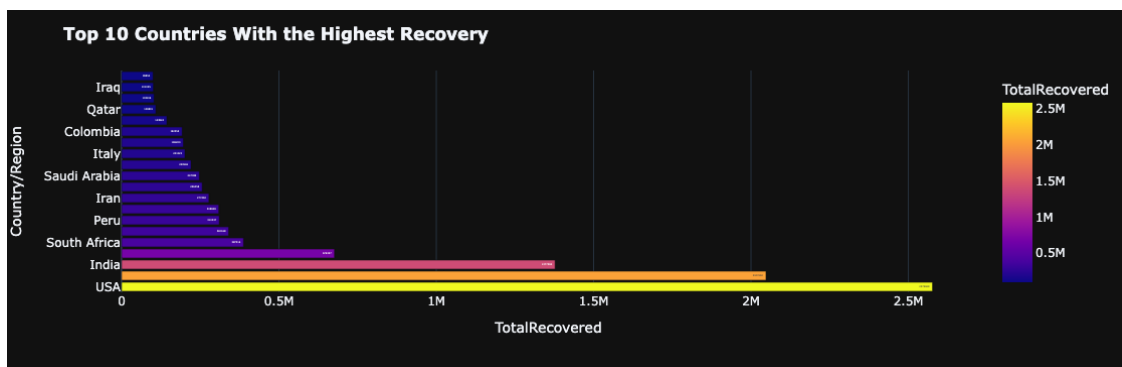
1.3.4 What are the top 10 countries with the highest deaths?

```
[67]: fig=px.bar(data6.sort_values(by='TotalDeaths',ascending=False)[0:20],y='Country/Region',x='TotalDeaths',color='TotalDeaths',text="TotalDeaths")
fig.update_layout(template="plotly_dark",title_text="<b>Top 10 Countries with the Highest Deaths</b>")
fig.show()
```



1.3.5 What are the Top 10 countries with the highest Recoveries?

```
[66]: fig=px.bar(data6.sort_values(by='TotalRecovered',ascending=False)[:20],y='Country/Region',x='TotalRecovered',color='TotalRecovered',text='TotalRecovered')
fig.update_layout(template="plotly_dark",title_text="<b>Top 10 Countries With the Highest Recovery</b>")
fig.show()
```



```
[32]: data6.columns
```

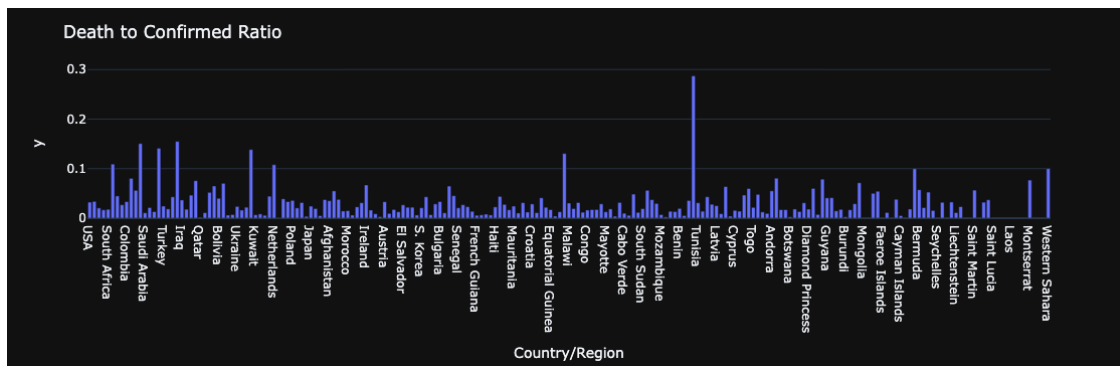
```
[32]: Index(['Country/Region', 'Continent', 'Population', 'TotalCases', 'NewCases',
          'TotalDeaths', 'NewDeaths', 'TotalRecovered', 'NewRecovered',
          'ActiveCases', 'Serious,Critical', 'Tot Cases/1M pop', 'Deaths/1M pop',
          'TotalTests', 'Tests/1M pop', 'WHO Region'],
          dtype='object')
```

```
[33]: data6[0:15]['Country/Region'].values
```

```
[33]: array(['USA', 'Brazil', 'India', 'Russia', 'South Africa', 'Mexico',
          'Peru', 'Chile', 'Colombia', 'Spain', 'Iran', 'UK', 'Saudi Arabia',
          'Pakistan', 'Bangladesh'], dtype=object)
```

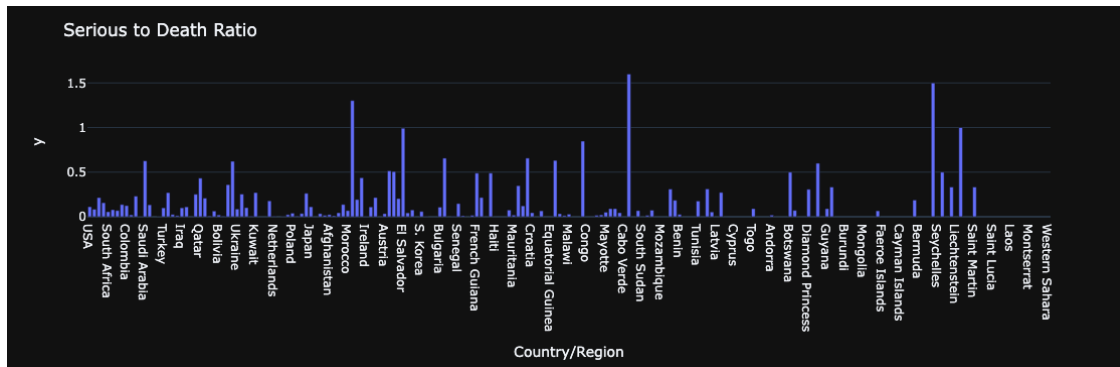
1.3.6 Deaths to Confirmed Ratio

```
[68]: deaths_to_confirmed=((data6['TotalDeaths']/data6['TotalCases']))
fig = px.bar(data6,x='Country/Region',y=deaths_to_confirmed)
fig.update_layout(title={'text':"Death to Confirmed Ratio",'xanchor':
    ↳'left'},template="plotly_dark")
fig.show()
```



1.3.7 Serious to Deaths Ratio

```
[72]: serious_to_death=((data6['Serious,Critical']/data6['TotalDeaths']))
fig = px.bar(data6,x='Country/Region',y=serious_to_death)
fig.update_layout(title={'text':"Serious to Death Ratio",'xanchor':
    ↳'left'},template="plotly_dark")
fig.show()
```



[]:

1.4 Automate Visualization of Confirmed, Active, Recovered , Deaths Cases of any Country

[39]: `data4.head()`

```
[39]:
```

	Date	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	\
0	2020-01-22	Afghanistan	0	0	0	0	0	
1	2020-01-22	Albania	0	0	0	0	0	
2	2020-01-22	Algeria	0	0	0	0	0	
3	2020-01-22	Andorra	0	0	0	0	0	
4	2020-01-22	Angola	0	0	0	0	0	

	New deaths	New recovered	WHO Region
0	0	0	Eastern Mediterranean
1	0	0	Europe
2	0	0	Africa
3	0	0	Europe
4	0	0	Africa

[73]: `from plotly.subplots import make_subplots`
`import plotly.graph_objects as go`

```
[46]: def country_visualization(data4,country):

    data=data4[data4['Country/Region']==country]
    df=data.loc[:,['Date','Confirmed','Deaths','Recovered','Active']]
    fig = make_subplots(rows=1, cols=4,subplot_titles=("Confirmed", "Active",
↵↵↵↵"Recovered", 'Deaths'))
    fig.add_trace(
        go.Scatter(name="Confirmed",x=df['Date'],y=df['Confirmed']),
        row=1, col=1
    )
```

```

fig.add_trace(
    go.Scatter(name="Active",x=df['Date'],y=df['Active']),
    row=1, col=2
)
fig.add_trace(
    go.Scatter(name="Recovered",x=df['Date'],y=df['Recovered']),
    row=1, col=3
)

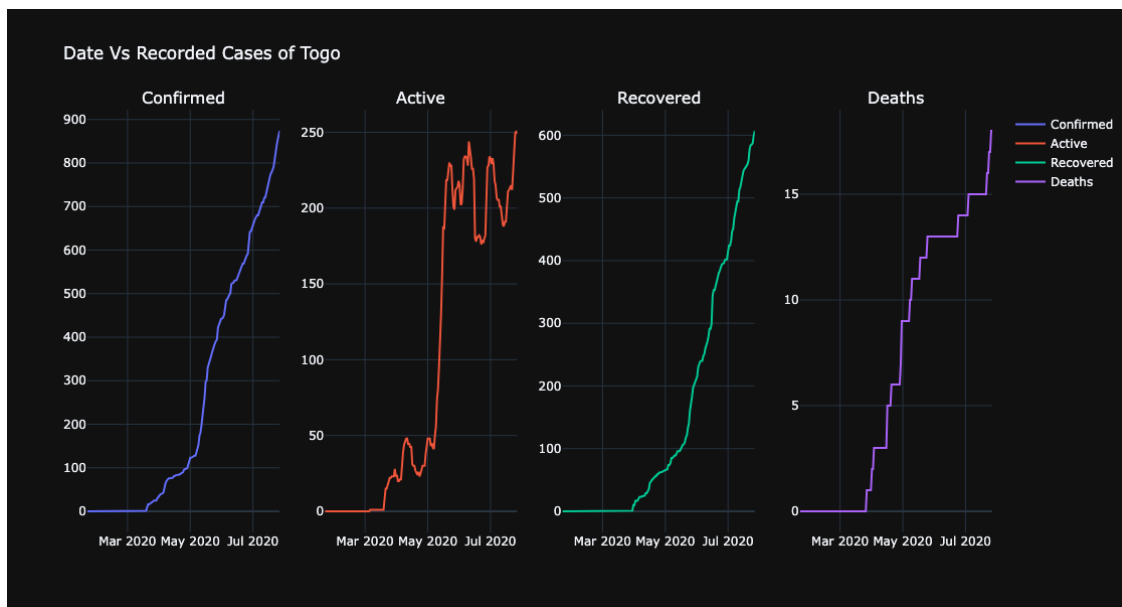
fig.add_trace(
    go.Scatter(name="Deaths",x=df['Date'],y=df['Deaths']),
    row=1, col=4
)

fig.update_layout(height=600, width=1000, title_text="Date Vs Recorded_
↪Cases of {}".format(country),template="plotly_dark")
fig.show()

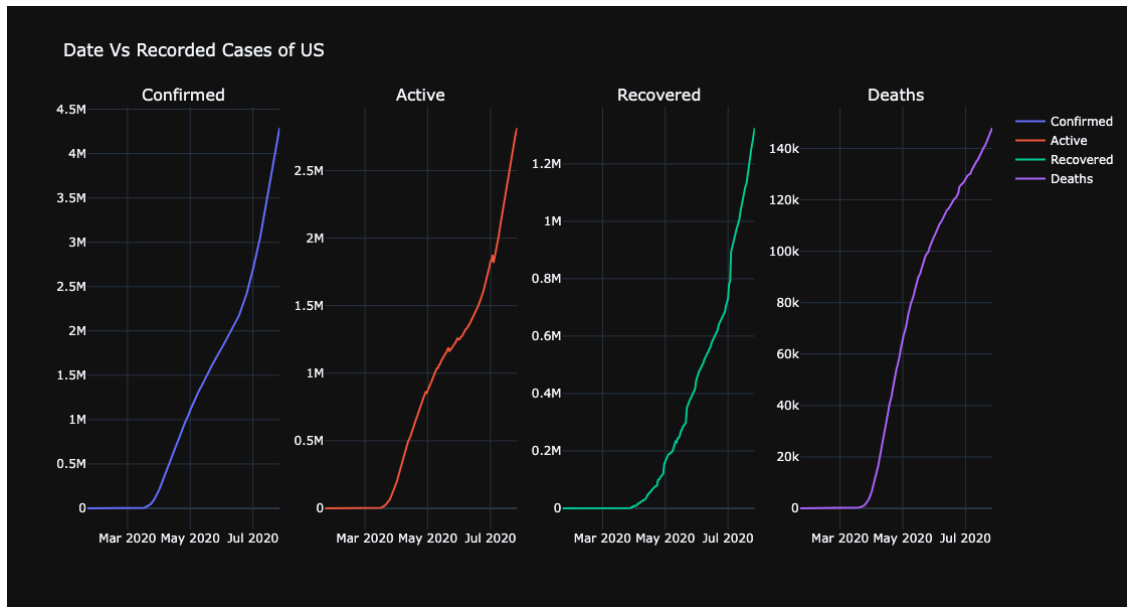
```

```
[ ]:
```

```
[74]: country_visualization(data4,'Togo')
```



```
[48]: country_visualization(data4,'US')
```

[]: