Covid-19 Analysis and Visualization using Plotly Express

```
# Data analysis and Manipulation
import plotly.graph_objs as go
import plotly.io as pio
import plotly.express as px
import pandas as pd

# Data Visualization
import matplotlib.pyplot as plt

# Importing Plotly
import plotly.offline as py
py.init_notebook_mode(connected=True)

# Initializing Plotly
pio.renderers.default = 'colab'
```

Importing Dataset

```
# Importing Dataset1
dataset1 = pd.read_csv("covid.csv")
dataset1.head()  # returns first 5 rows

```

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	NewRecovered	ActiveCases	Seri
0	USA	North America	3.311981e+08	5032179	NaN	162804.0	NaN	2576668.0	NaN	2292707.0	
1	Brazil	South America	2.127107e+08	2917562	NaN	98644.0	NaN	2047660.0	NaN	771258.0	
2	India	Asia	1.381345e+09	2025409	NaN	41638.0	NaN	1377384.0	NaN	606387.0	
3	Russia	Europe	1.459409e+08	871894	NaN	14606.0	NaN	676357.0	NaN	180931.0	
4	South Africa	Africa	5.938157e+07	538184	NaN	9604.0	NaN	387316.0	NaN	141264.0	

Next steps: Generate code with dataset1 View recommended plots New interactive sheet

getting dataset information

```
# Returns tuple of shape (Rows, columns)
print(dataset1.shape)

# Returns size of dataframe
print(dataset1.size)

```

```
(209, 17)
3553
```

dataset information

```
# Information about Dataset1
# return concise summary of dataframe
dataset1.info()

```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209 entries, 0 to 208
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
```

```
 0   Country/Region    209 non-null    object
 1   Continent         208 non-null    object
 2   Population        208 non-null    float64
 3   TotalCases        209 non-null    int64
 4   NewCases          4 non-null      float64
 5   TotalDeaths       188 non-null    float64
 6   NewDeaths         3 non-null      float64
 7   TotalRecovered    205 non-null    float64
 8   NewRecovered      3 non-null      float64
 9   ActiveCases       205 non-null    float64
 10  Serious,Critical  122 non-null    float64
 11  Tot Cases/1M pop  208 non-null    float64
 12  Deaths/1M pop     187 non-null    float64
 13  TotalTests        191 non-null    float64
 14  Tests/1M pop      191 non-null    float64
 15  WHO Region        184 non-null    object
 16  iso_alpha         209 non-null    object
dtypes: float64(12), int64(1), object(4)
memory usage: 27.9+ KB
```

Importing dataset

```
# Importing Dataset2
dataset2 = pd.read_csv("covid_grouped.csv")
dataset2.head()  # return first 5 rows of dataset2

```

	Date	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	WHO Region	iso_alpha
0	2020-01-22	Afghanistan	0	0	0	0	0	0	0	Eastern Mediterranean	AFG
1	2020-01-22	Albania	0	0	0	0	0	0	0	Europe	ALB
2	2020-01-	Algeria	0	0	0	0	0	0	0	Africa	DZA

Next steps: Generate code with dataset2 View recommended plots New interactive sheet

Getting dataset information

```
# Returns tuple of shape (Rows, columns)
print(dataset2.shape)

# Returns size of dataframe
print(dataset2.size)
```

```
(35156, 11)
386716
```

Dataset information

```
# Information about Dataset2
dataset2.info()  # return concise summary of dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35156 entries, 0 to 35155
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            35156 non-null  object
 1   Country/Region  35156 non-null  object
 2   Confirmed       35156 non-null  int64
 3   Deaths          35156 non-null  int64
 4   Recovered       35156 non-null  int64
 5   Active          35156 non-null  int64
 6   New cases       35156 non-null  int64
 7   New deaths      35156 non-null  int64
 8   New recovered   35156 non-null  int64
 9   WHO Region      35156 non-null  object
 10  iso_alpha       35156 non-null  object
dtypes: int64(7), object(4)
memory usage: 3.0+ MB
```

Dataset cleaning

```
# Columns labels of a Dataset1
dataset1.columns
```

```
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', 'iso_alpha'],
      dtype='object')
```

```
# Drop NewCases, NewDeaths, NewRecovered rows from dataset1

dataset1.drop(['NewCases', 'NewDeaths', 'NewRecovered'],
              axis=1, inplace=True)

# Select random set of values from dataset1
dataset1.sample(5)
```

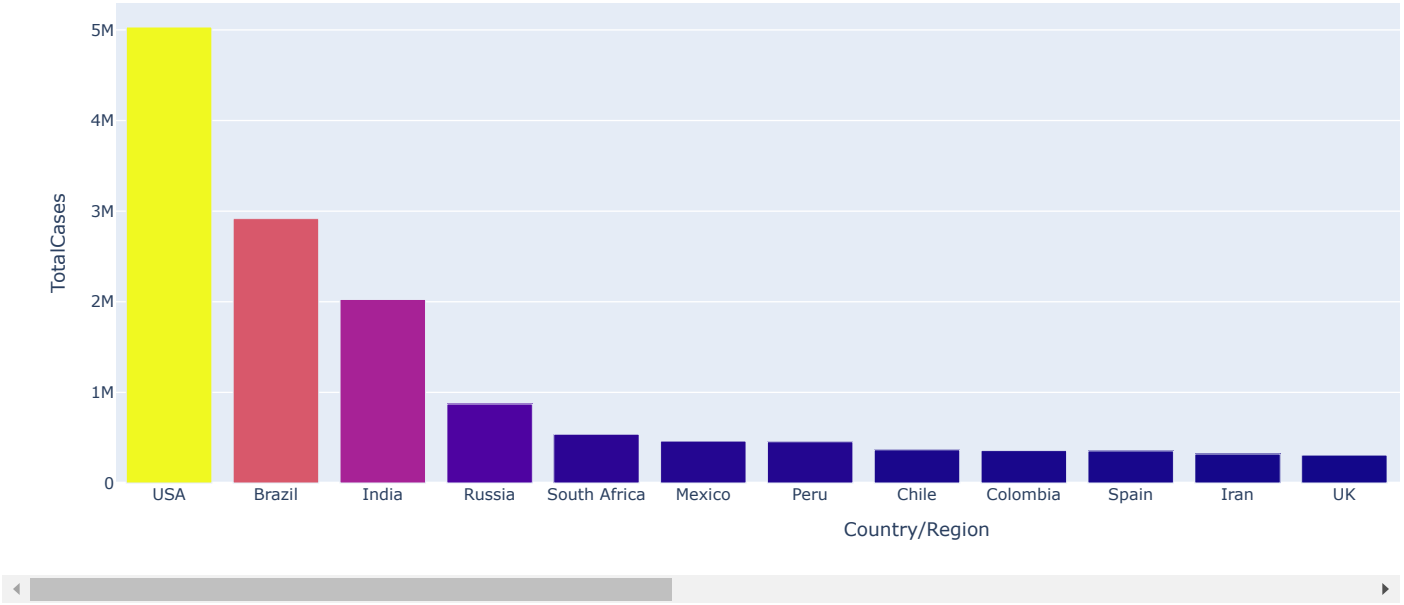
	Country/Region	Continent	Population	TotalCases	TotalDeaths	TotalRecovered	ActiveCases	Serious,Critical	Tot Cases/1M pop	Death
73	Denmark	Europe	5794279.0	14306	617.0	12787.0	902.0	2.0	2469.0	
89	Finland	Europe	5541604.0	7532	331.0	6980.0	221.0	NaN	1359.0	
114	Sri Lanka	Asia	21422362.0	2839	11.0	2541.0	287.0	1.0	133.0	
190	French Polynesia	Australia/Oceania	281072.0	64	NaN	62.0	2.0	NaN	228.0	
13	Pakistan	Asia	221295851.0	281863	6035.0	256058.0	19770.0	809.0	1274.0	

```
# Import create_table Figure Factory

from plotly.figure_factory import create_table

colorscale = [[0, '#4d004c'], [.5, '#f2e5ff'], [1, '#ffffff']]
table = create_table(dataset1.head(15), colorscale=colorscale)
py.iplot(table)
```

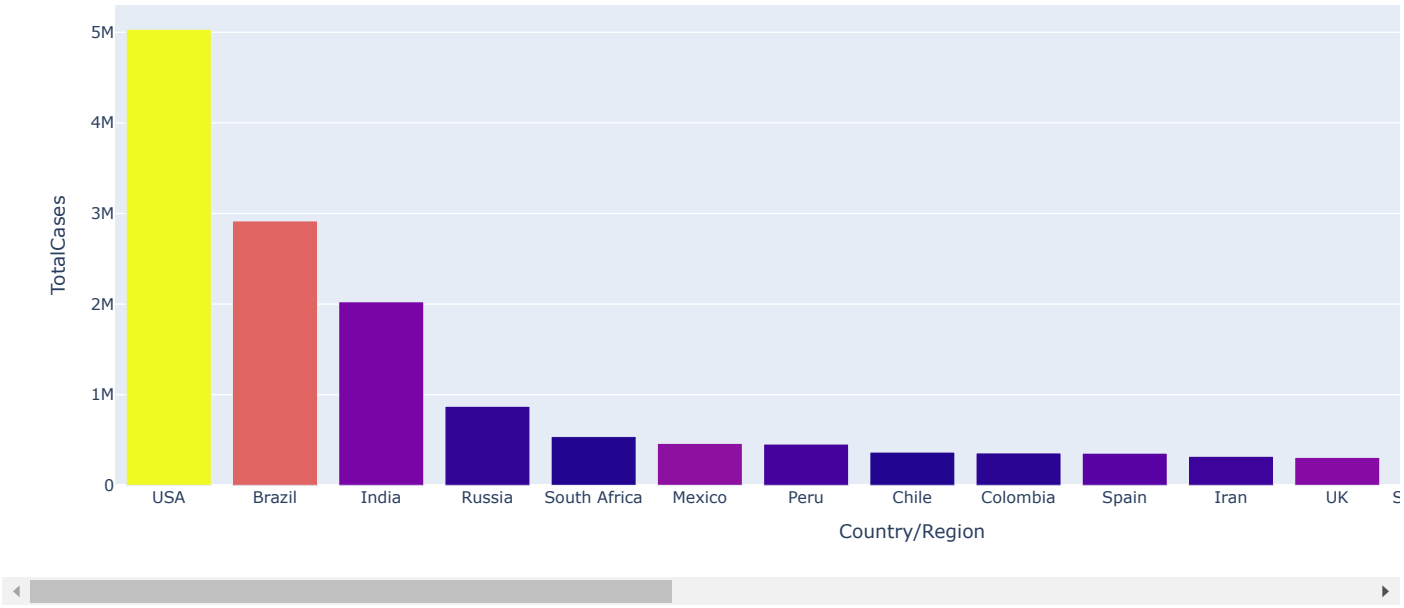
Country/Region	Continent	Population	TotalCases	TotalDeaths	TotalRecovered	ActiveCases	Serious,Critical	Tot Cases/1M pop	Deaths/1M pop	Tota
USA	North America	331198130.0	5032179	162804.0	2576668.0	2292707.0	18296.0	15194.0	492.0	6313
Brazil	South America	212710692.0	2917562	98644.0	2047660.0	771258.0	8318.0	13716.0	464.0	1320
India	Asia	1381344997.0	2025409	41638.0	1377384.0	606387.0	8944.0	1466.0	30.0	2214
Russia	Europe	145940924.0	871894	14606.0	676357.0	180931.0	2300.0	5974.0	100.0	2971
South Africa	Africa	59381566.0	538184	9604.0	387316.0	141264.0	539.0	9063.0	162.0	3149
Mexico	North America	129066160.0	462690	50517.0	308848.0	103325.0	3987.0	3585.0	391.0	1056
Peru	South America	33016319.0	455409	20424.0	310337.0	124648.0	1426.0	13793.0	619.0	2493
Chile	South America	19132514.0	366671	9889.0	340168.0	16614.0	1358.0	19165.0	517.0	1760
Colombia	South America	50936262.0	357710	11939.0	192355.0	153416.0	1493.0	7023.0	234.0	1801
Spain	Europe	46756648.0	354530	28500.0	nan	nan	617.0	7582.0	610.0	7064
Iran	Asia	84097623.0	320117	17976.0	277463.0	24678.0	4156.0	3806.0	214.0	2612
UK	Europe	67922029.0	308134	46413.0	nan	nan	73.0	4537.0	683.0	1751
Saudi Arabia	Asia	34865919.0	284226	3055.0	247089.0	34082.0	1915.0	8152.0	88.0	3635
Pakistan	Asia	221295851.0	281863	6035.0	256058.0	19770.0	809.0	1274.0	27.0	2058
Bangladesh	Asia	164851401.0	249651	3306.0	143824.0	102521.0	nan	1514.0	20.0	1225

Bar graphs- Comparisons between COVID infected countries in terms of total cases, total deaths, total recovered & total tests

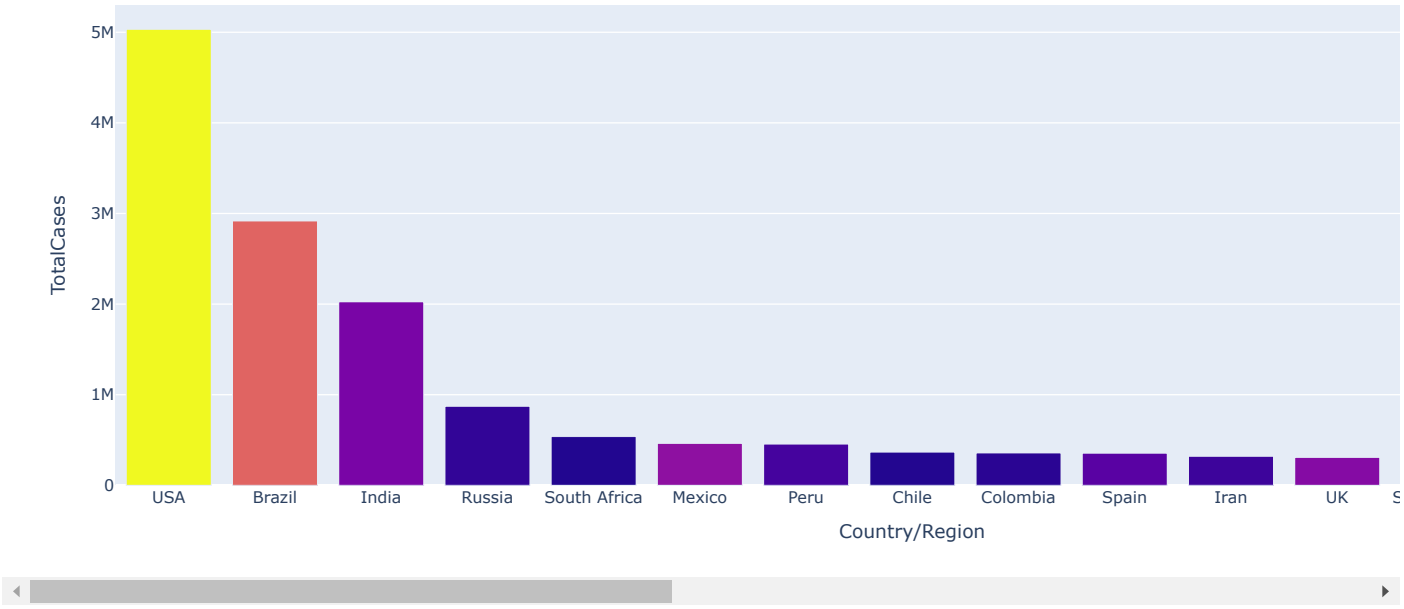
```
px.bar(dataset1.head(15), x = 'Country/Region',
       y = 'TotalCases',color = 'TotalCases',
       height = 500,hover_data = ['Country/Region', 'Continent'])
```



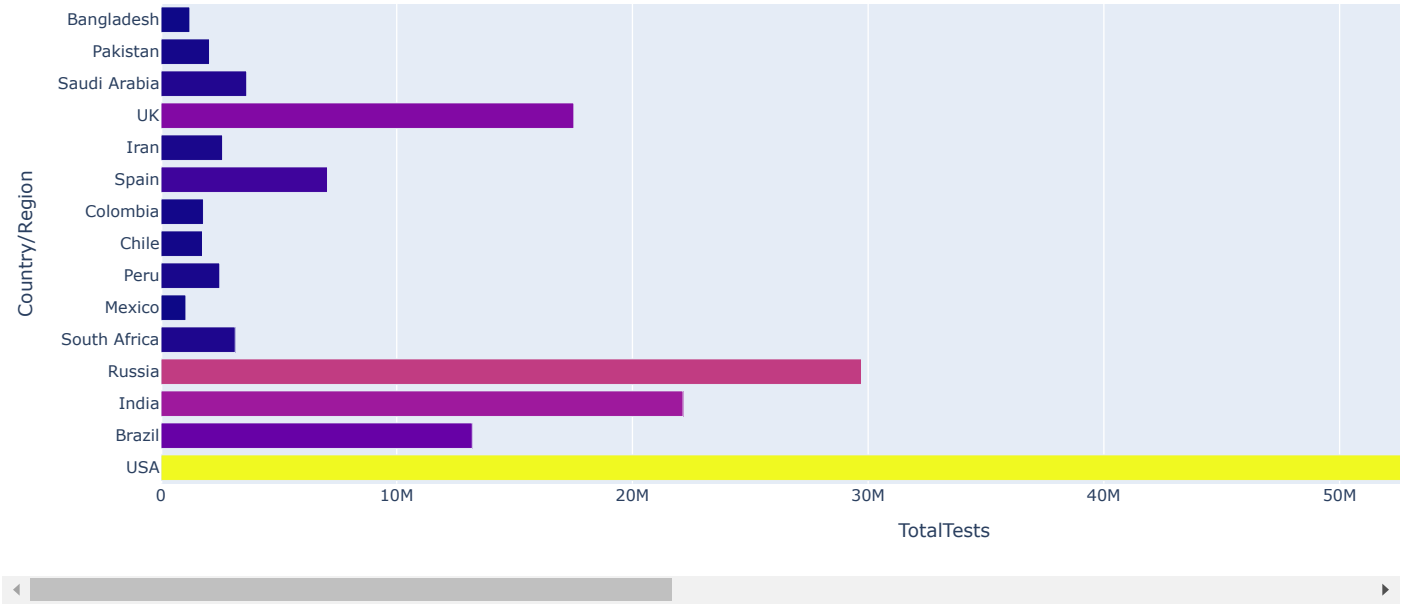

```
px.bar(dataset1.head(15), x = 'Country/Region', y = 'TotalCases',
       color = 'TotalDeaths', height = 500,
       hover_data = ['Country/Region', 'Continent'])
```




```
px.bar(dataset1.head(15), x = 'Country/Region', y = 'TotalCases',
       color = 'TotalDeaths', height = 500,
       hover_data = ['Country/Region', 'Continent'])
```




```
px.bar(dataset1.head(15), x = 'TotalTests', y = 'Country/Region',
       color = 'TotalTests',orientation ='h',  height = 500,
       hover_data = ['Country/Region', 'Continent'])
```



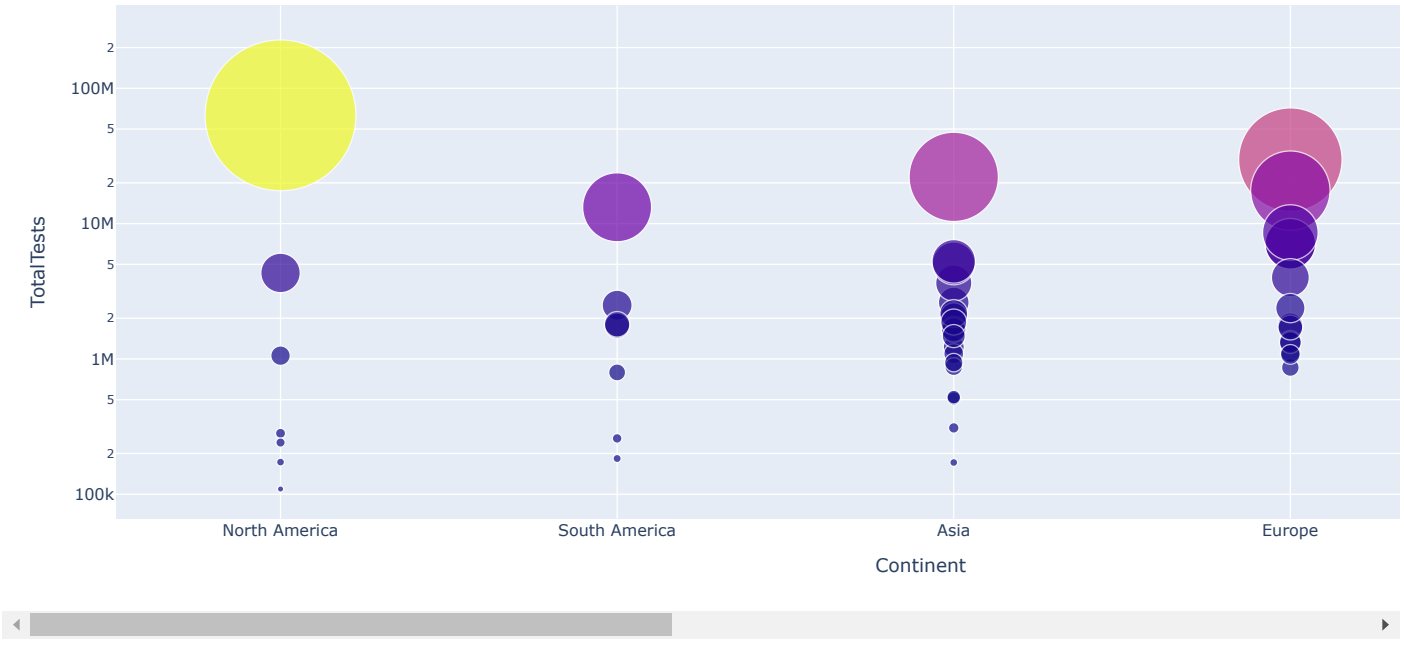
Data Visualization through Bubble Charts-Continent Wise

```
px.scatter(dataset1, x='Continent',y='TotalCases',
           hover_data=['Country/Region', 'Continent'],
           color='TotalCases', size='TotalCases', size_max=80)

```


```
px.scatter(dataset1, x='Continent',y='TotalCases',
           hover_data=['Country/Region', 'Continent'],
           color='TotalCases', size='TotalCases', size_max=80,log_y=True)

```

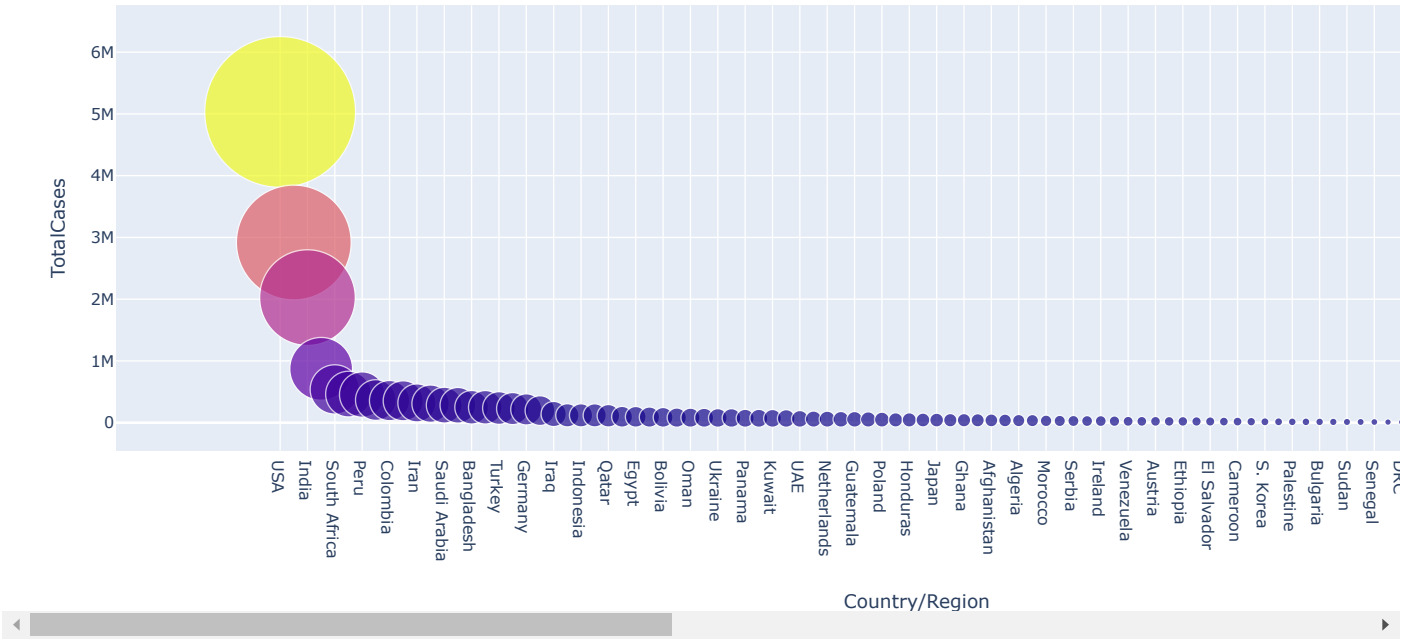

```
px.scatter(dataset1.head(54), x='Continent',y='TotalTests',
           hover_data=['Country/Region', 'Continent'],
           color='TotalTests', size='TotalTests', size_max=80)

```


```
px.scatter(dataset1.head(50), x='Continent',y='TotalTests',
           hover_data=['Country/Region', 'Continent'],
           color='TotalTests', size='TotalTests', size_max=80, log_y=True)

```

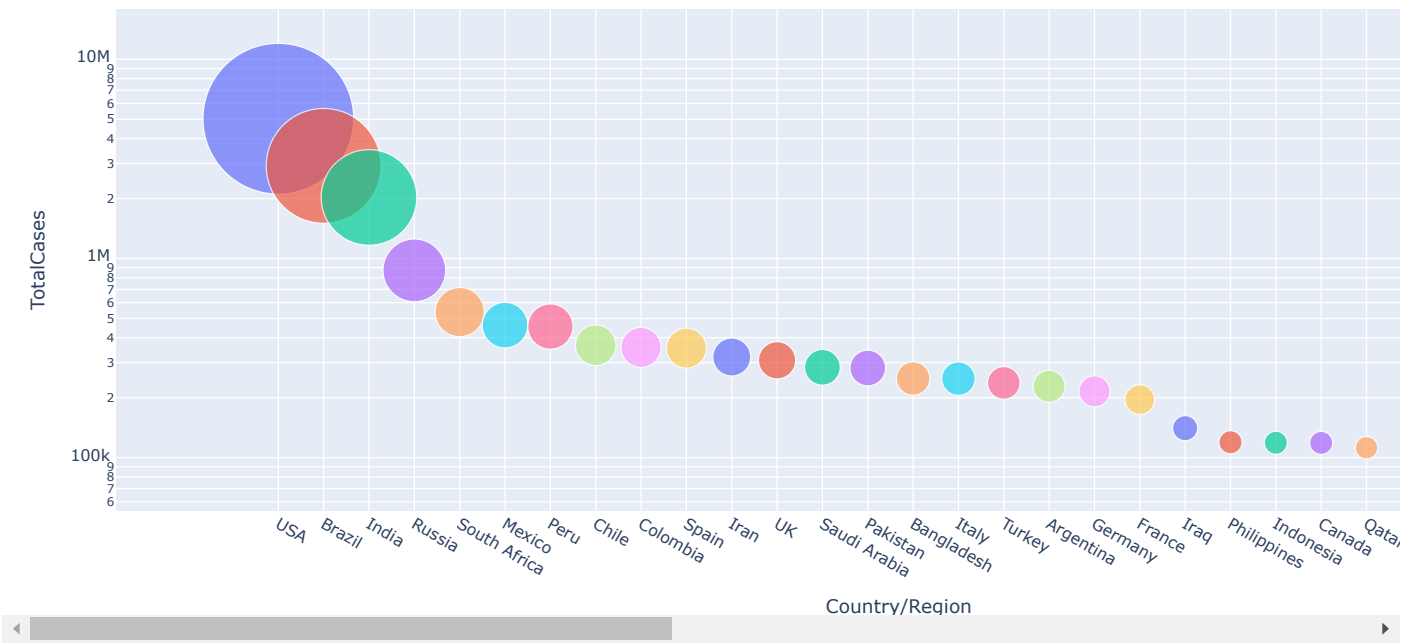


Data Visualization through Bubble Charts-Country Wise

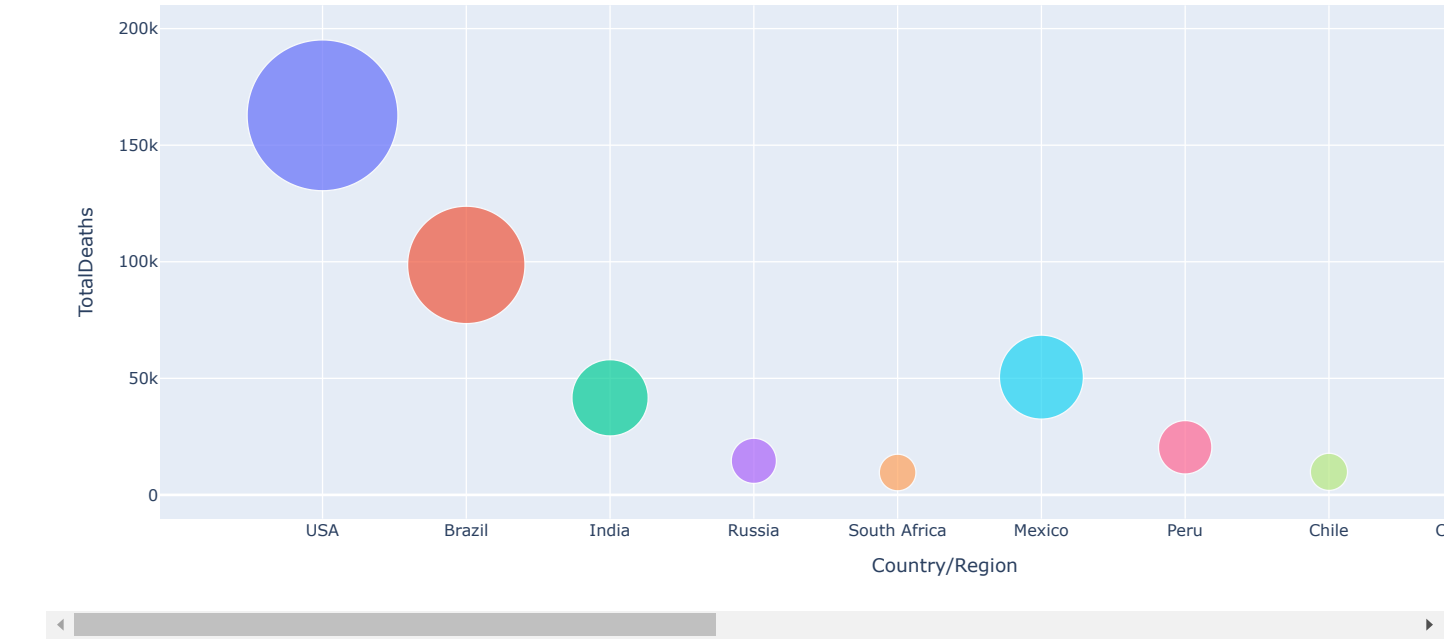
```
px.scatter(dataset1.head(100), x='Country/Region', y='TotalCases',
           hover_data=['Country/Region', 'Continent'],
           color='TotalCases', size='TotalCases', size_max=80)

```



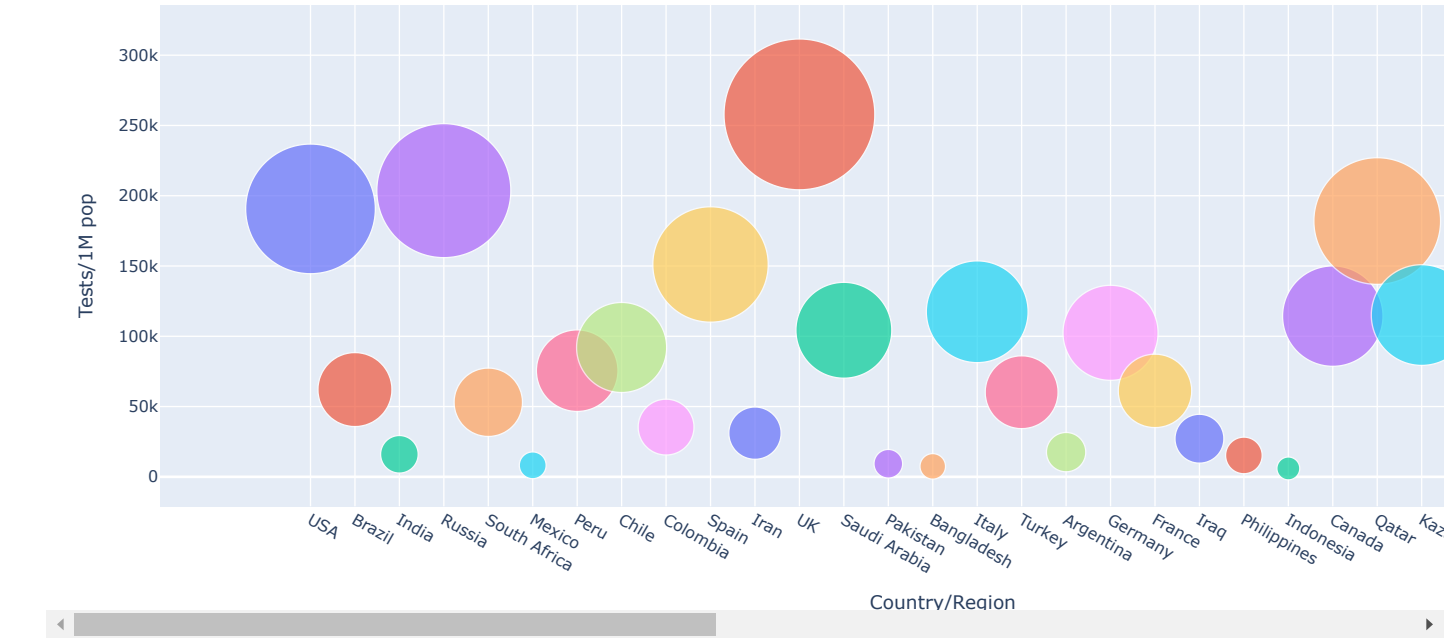

```
px.scatter(dataset1.head(30), x='Country/Region', y='TotalCases',
           hover_data=['Country/Region', 'Continent'],
           color='Country/Region', size='TotalCases', size_max=80, log_y=True)

```



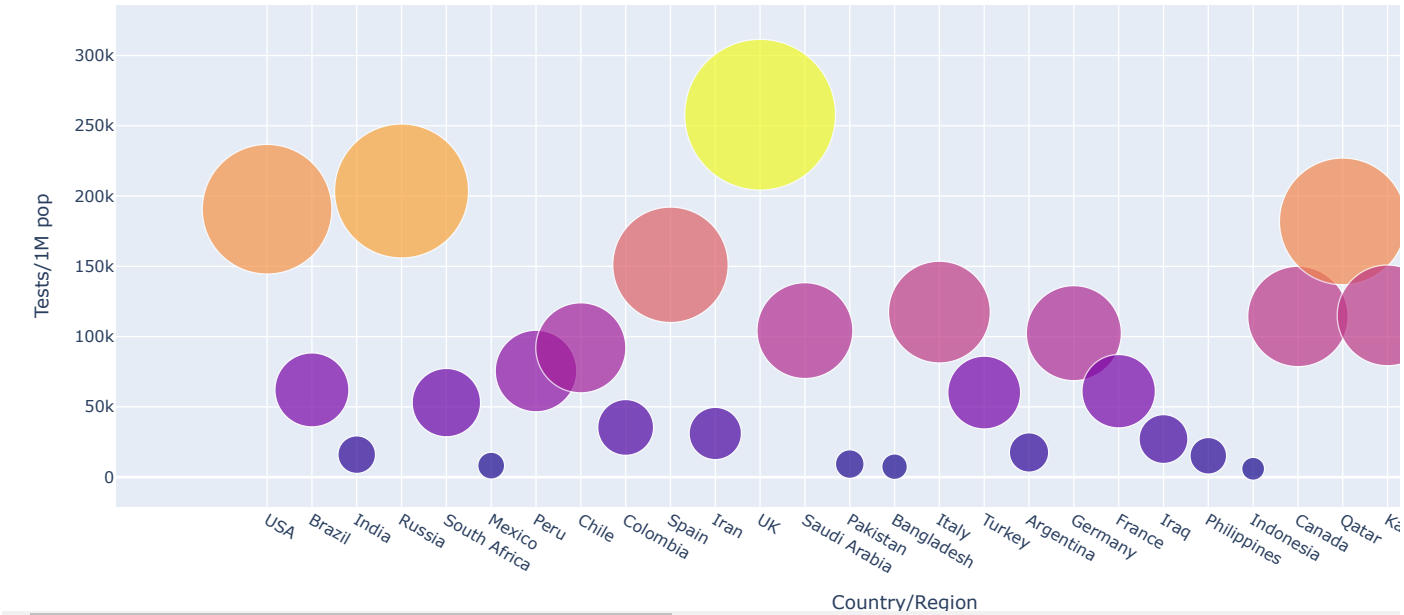

```
px.scatter(dataset1.head(10), x='Country/Region', y= 'TotalDeaths',
           hover_data=['Country/Region', 'Continent'],
           color='Country/Region', size= 'TotalDeaths', size_max=80)

```



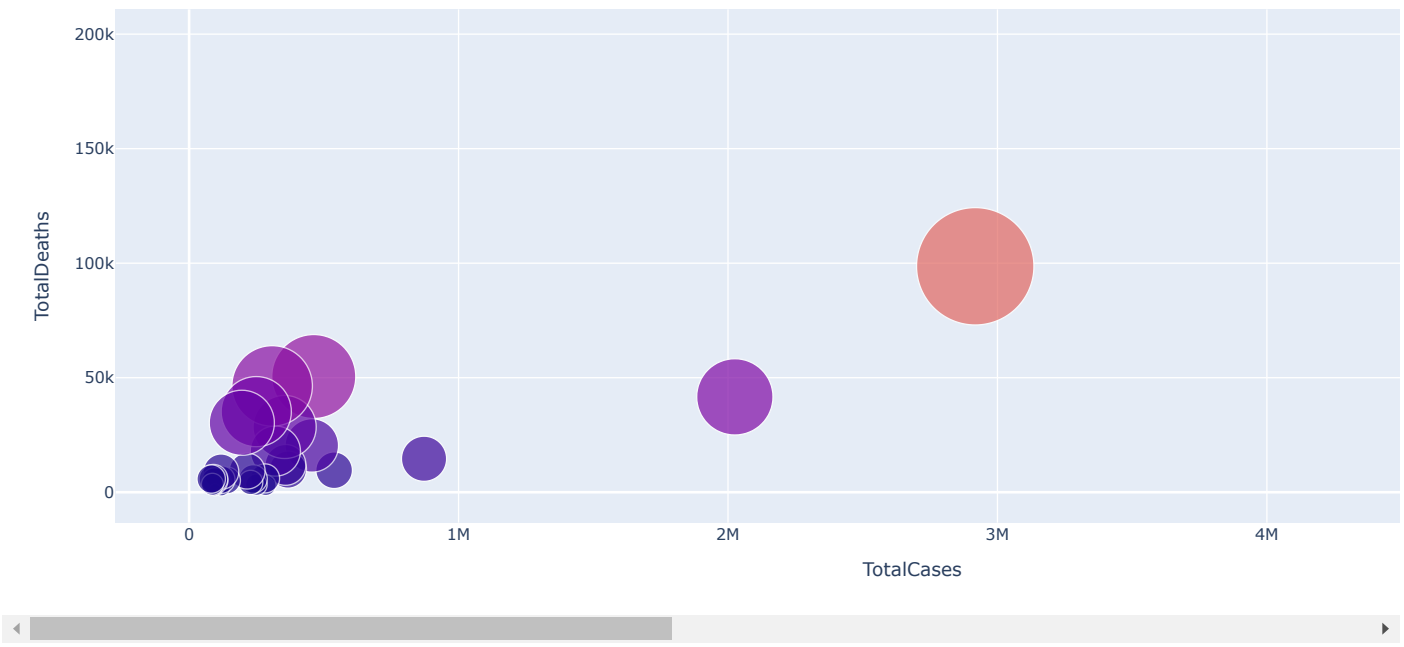

```
px.scatter(dataset1.head(30), x='Country/Region', y= 'Tests/1M pop',
           hover_data=['Country/Region', 'Continent'],
           color='Country/Region', size= 'Tests/1M pop', size_max=80)
```



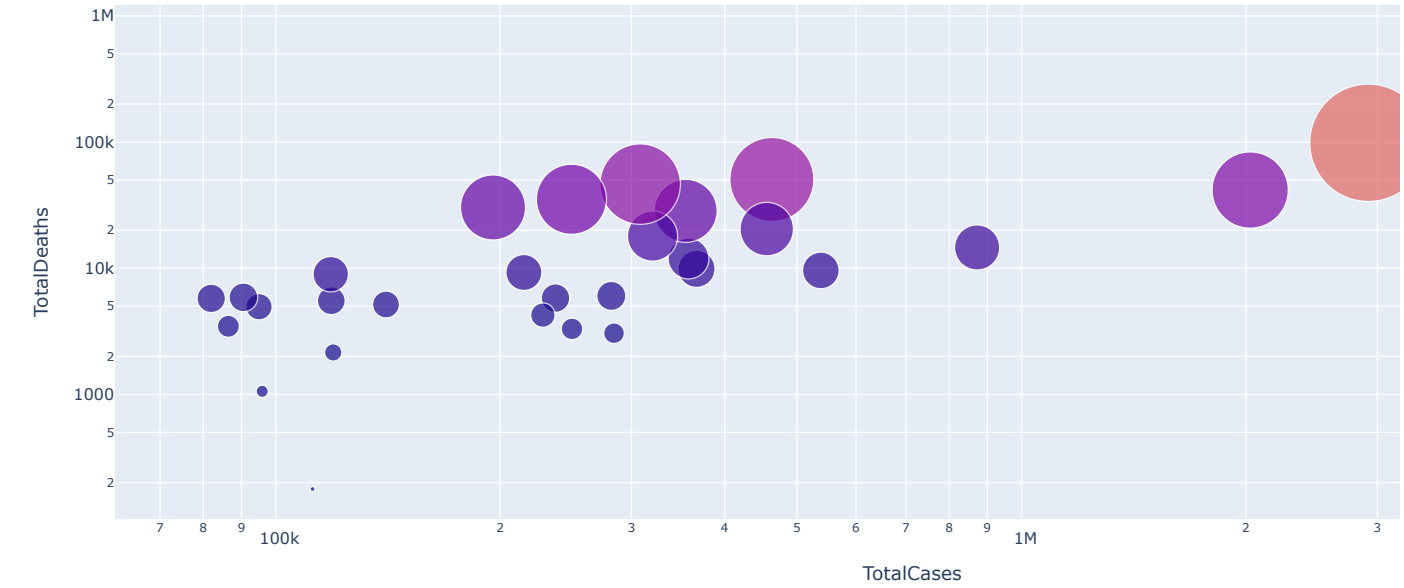

```
px.scatter(dataset1.head(30), x='Country/Region', y= 'Tests/1M pop',
           hover_data=['Country/Region', 'Continent'],
           color='Tests/1M pop', size= 'Tests/1M pop', size_max=80)

```



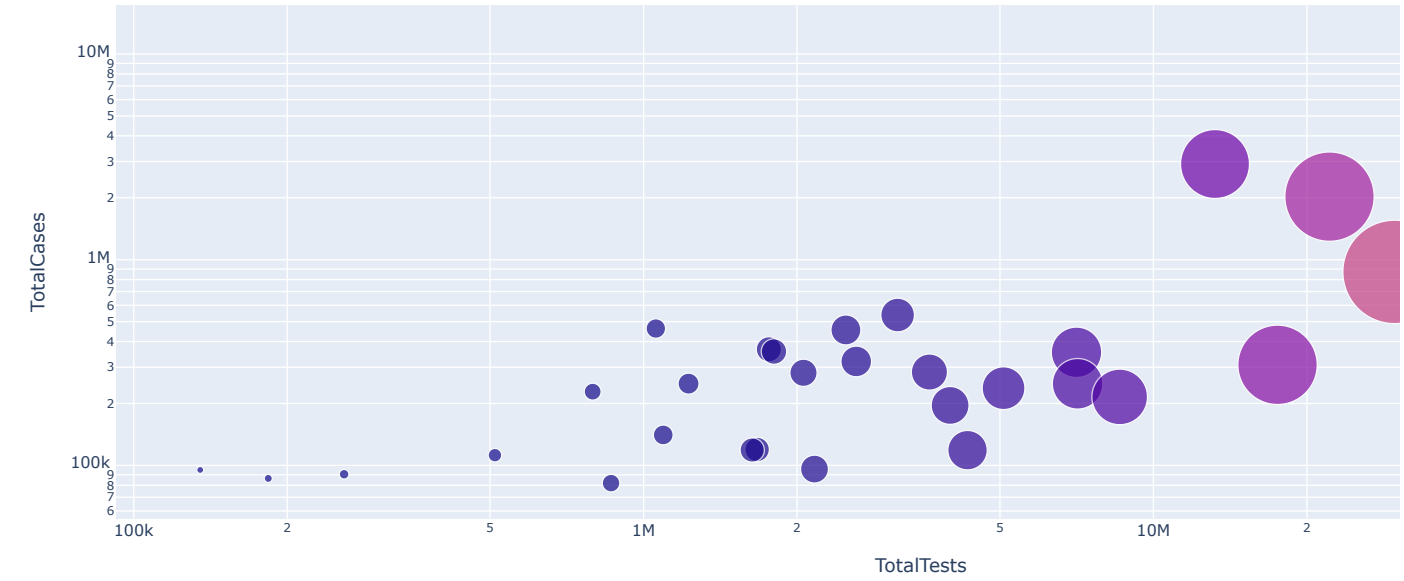

```
px.scatter(dataset1.head(30), x='TotalCases', y= 'TotalDeaths',
           hover_data=['Country/Region', 'Continent'],
           color='TotalDeaths', size= 'TotalDeaths', size_max=80)

```




```
px.scatter(dataset1.head(30), x='TotalCases', y= 'TotalDeaths',
           hover_data=['Country/Region', 'Continent'],
           color='TotalDeaths', size= 'TotalDeaths', size_max=80,
           log_x=True, log_y=True)
```

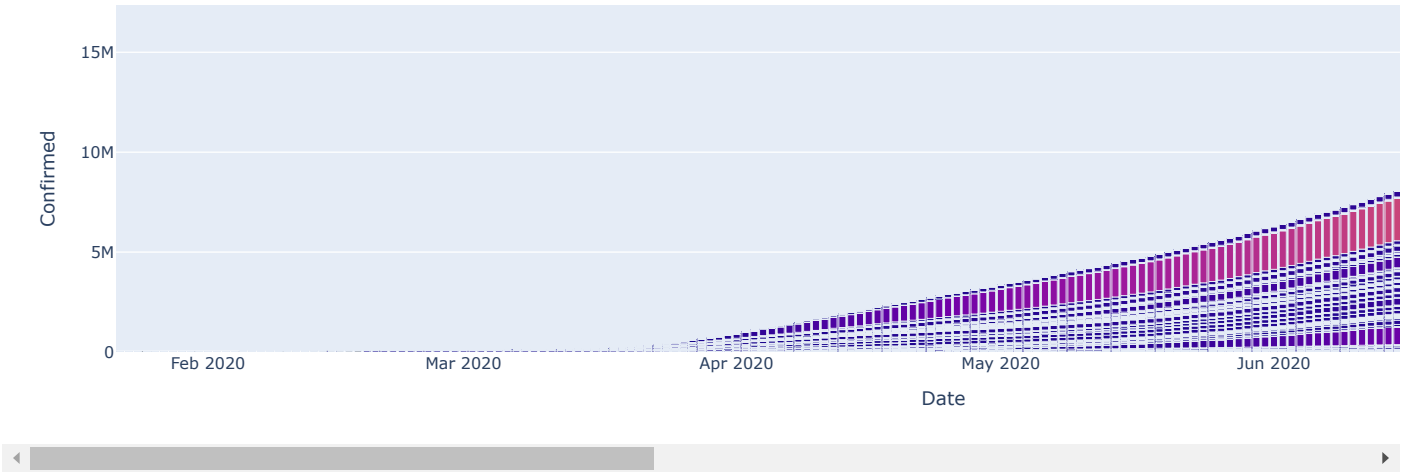



```
px.scatter(dataset1.head(30), x='TotalTests', y= 'TotalCases',
           hover_data=['Country/Region', 'Continent'],
           color='TotalTests', size= 'TotalTests', size_max=80,
           log_x=True, log_y=True)
```

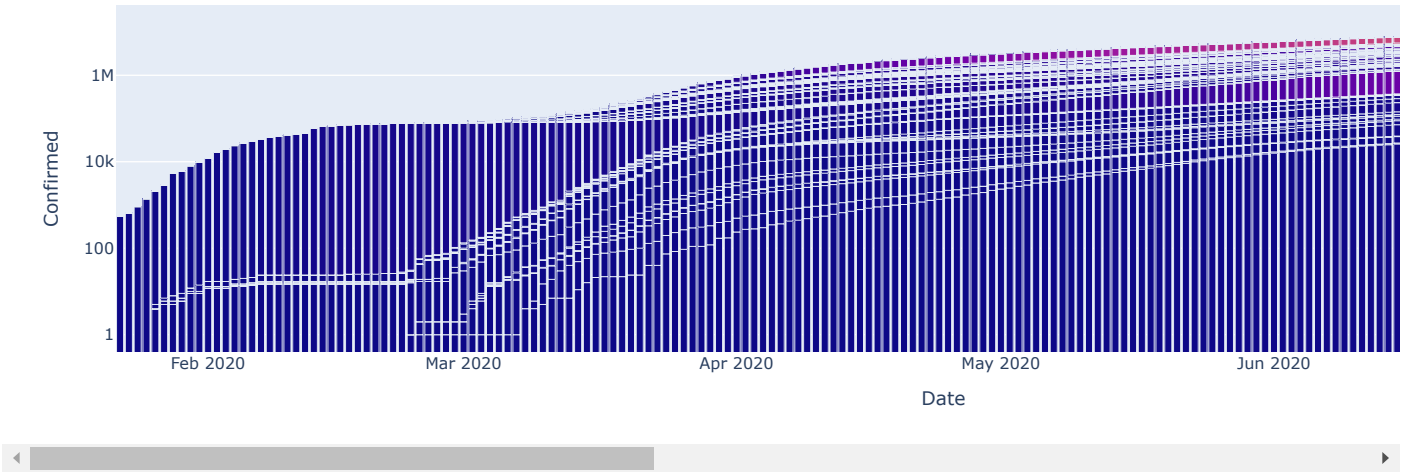


Advanced Data Visualization- Bar graphs for All top infected Countries

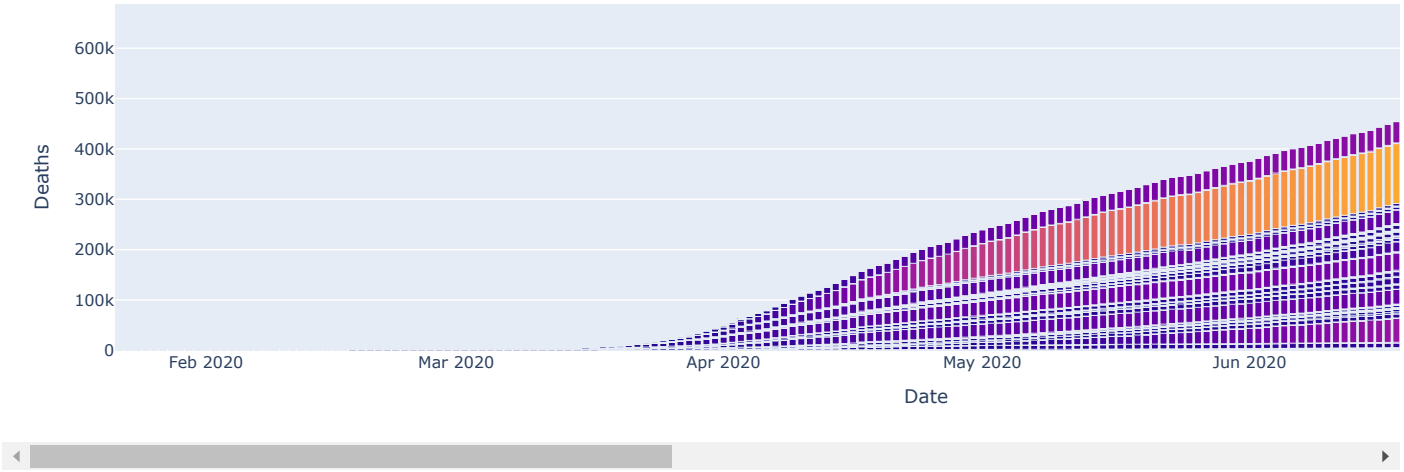
```
px.bar(dataset2, x="Date", y="Confirmed", color="Confirmed",
       hover_data=["Confirmed", "Date", "Country/Region"], height=400)

```



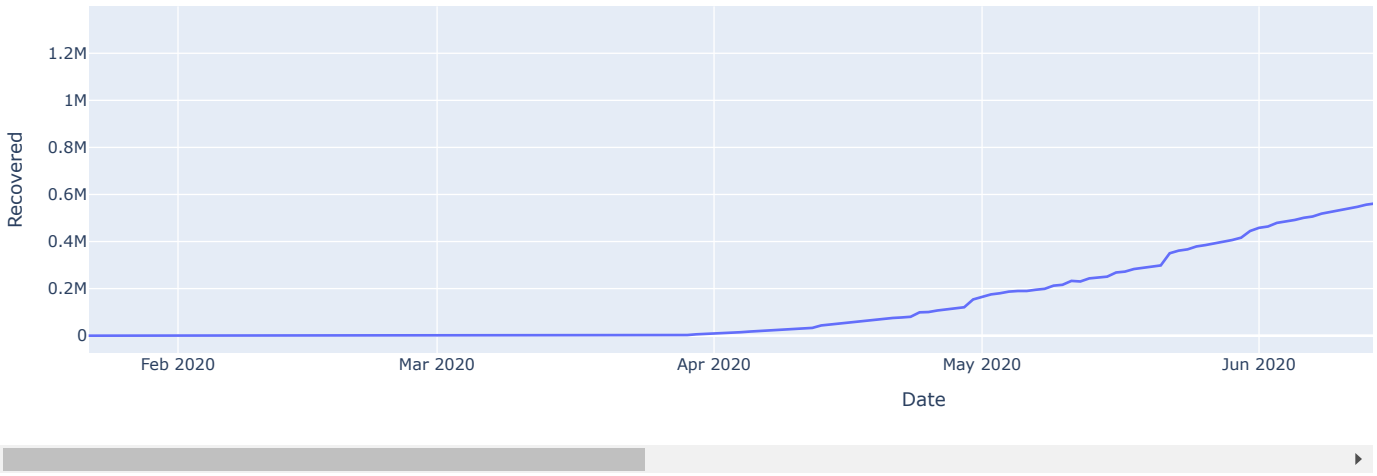

```
px.bar(dataset2, x="Date", y="Confirmed", color="Confirmed",
       hover_data=["Confirmed", "Date", "Country/Region"],log_y=True, height=400)

```



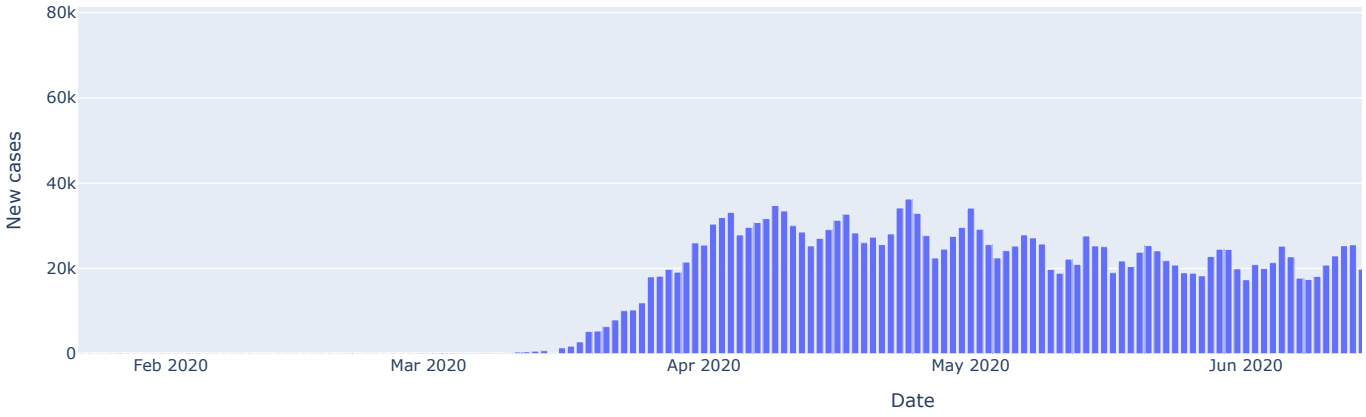

```
px.bar(dataset2, x="Date", y="Deaths", color="Deaths",
       hover_data=["Confirmed", "Date", "Country/Region"],
       log_y=False, height=400)

```



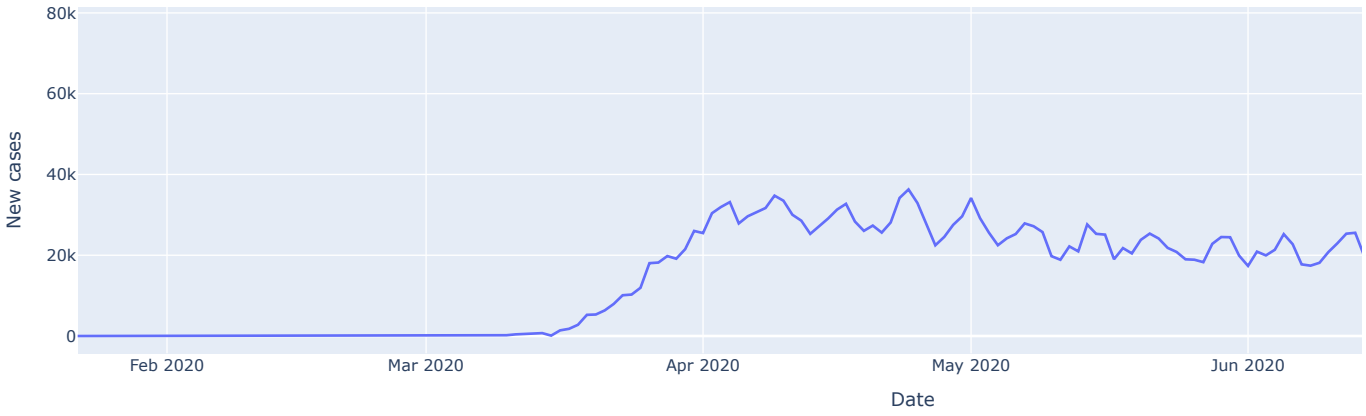
```
df_US= dataset2.loc[dataset2["Country/Region"]=="US"]
px.line(df_US,x="Date", y="Recovered", height=400)

```



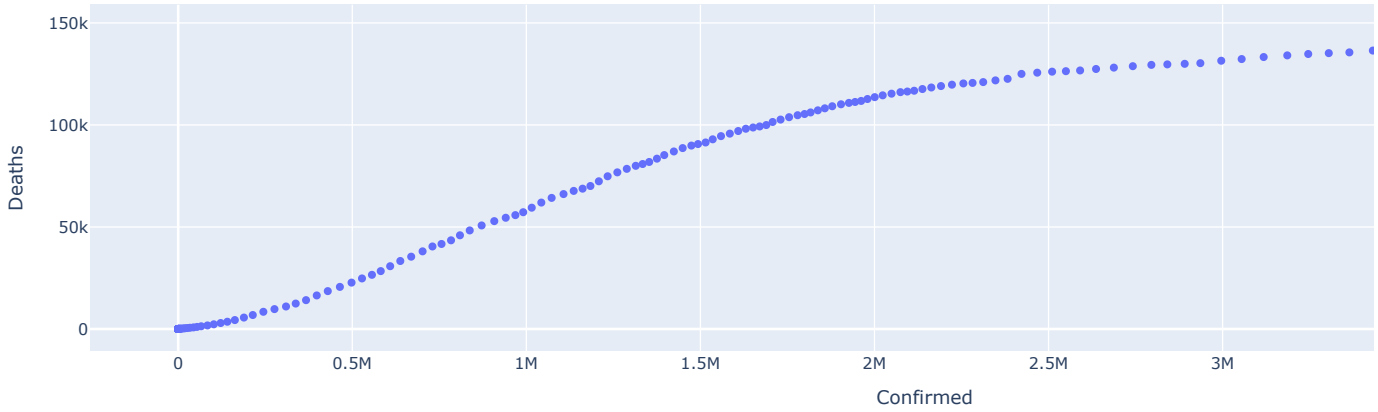

```
px.bar(df_US,x="Date", y="New cases", height=400)

```




```
px.line(df_US,x="Date", y="New cases", height=400)

```

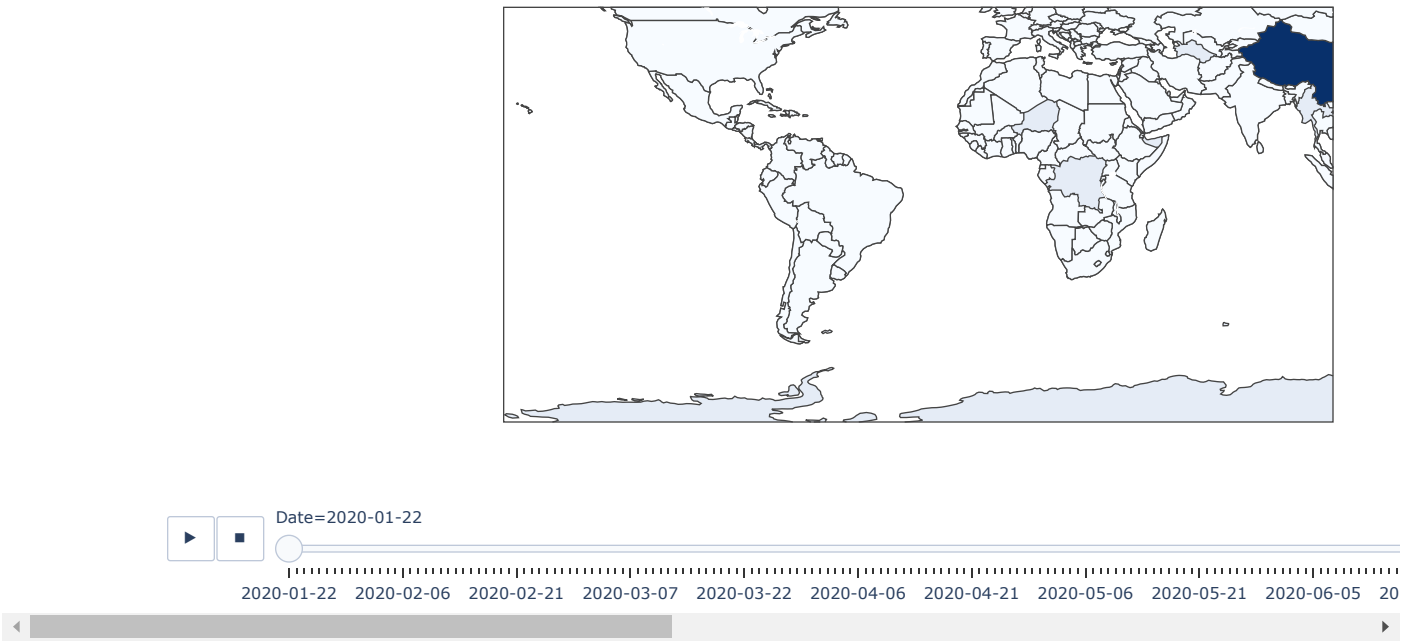


```
1 px.scatter(df_US, x="Confirmed", y="Deaths", height=400)
```

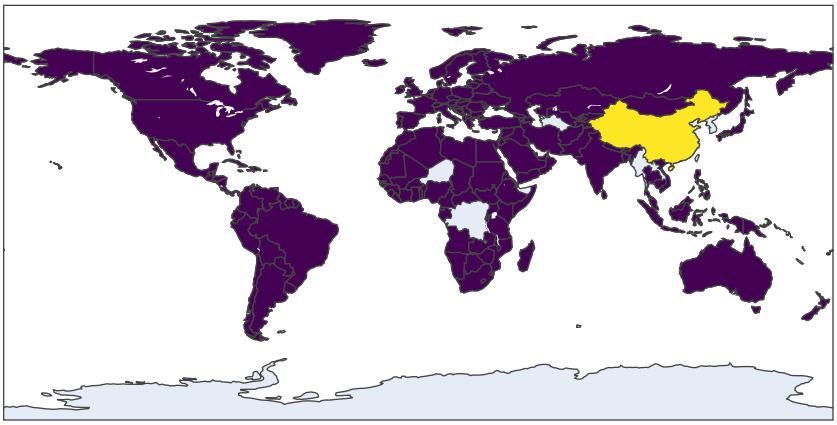


Visualization of Data in terms of Maps

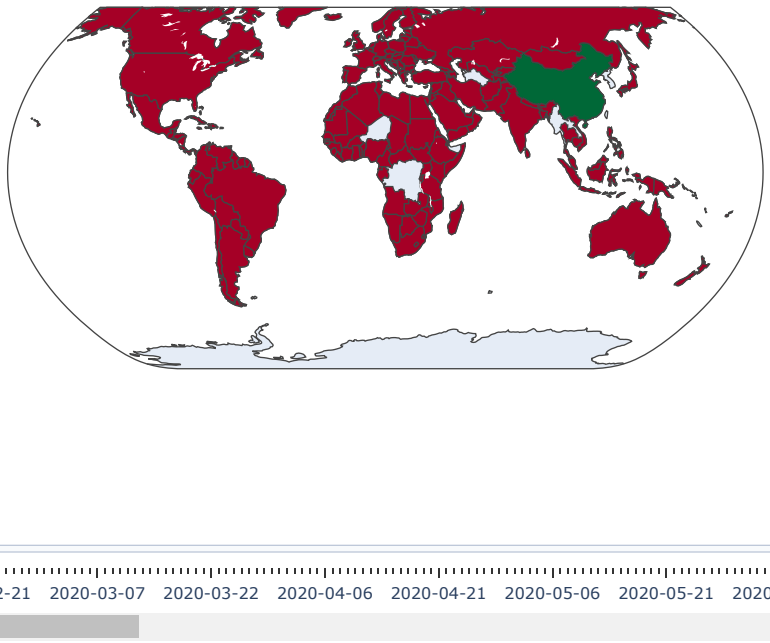
```
1 px.choropleth(dataset2,
2                locations="iso_alpha",
3                color="Confirmed",
4                hover_name="Country/Region",
5                color_continuous_scale="Blues",
6                animation_frame="Date")
```



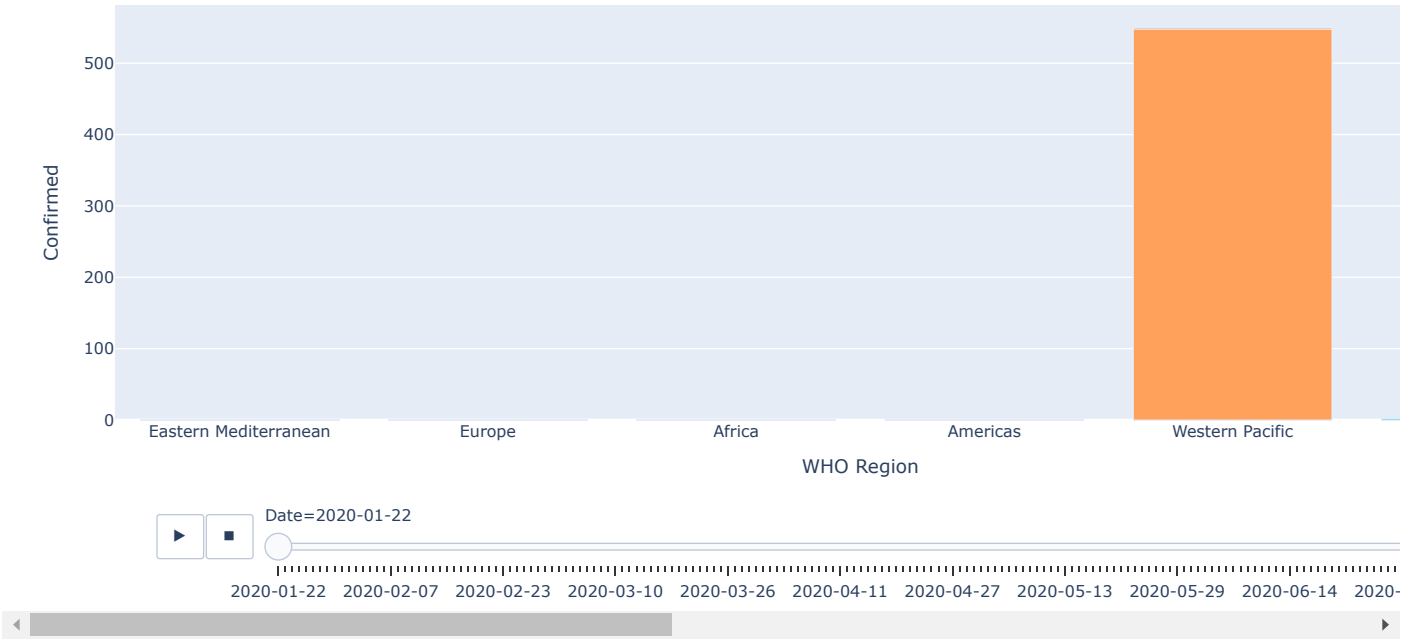

```
1 px.choropleth(dataset2,
2                locations='iso_alpha',
3                color="Deaths",
4                hover_name="Country/Region",
5                color_continuous_scale="Viridis",
6                animation_frame="Date" )
```




```
px.choropleth(dataset2,
              locations='iso_alpha',
              color="Recovered",
              hover_name="Country/Region",
              color_continuous_scale="RdYlGn",
              projection="natural earth",
              animation_frame="Date" )
```




```
px.bar(dataset2, x="WHO Region", y="Confirmed", color="WHO Region",
       animation_frame="Date", hover_name="Country/Region")
```



DATASET 3 Analysis: Visualize text using Word Cloud

Visualize text using Word Cloud

```
1 dataset3= pd.read_csv("coviddeath.csv")
2 dataset3.head()
```

	Data as of	Start Week	End Week	State	Condition Group	Condition	ICD10_codes	Age Group	Number of COVID-19 Deaths	Flag
0	08/30/2020	02/01/2020	08/29/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	0-24	122.0	NaN
1	08/30/2020	02/01/2020	08/29/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	25-34	596.0	NaN
2	08/30/2020	02/01/2020	08/29/2020	US	Respiratory	Influenza and	J09-J18	35-44	1521.0	NaN

Next steps: Generate code with dataset3 View recommended plots New interactive sheet

```
1 dataset3.tail()
```

	Data as of	Start Week	End Week	State	Condition Group	Condition	ICD10_codes	Age Group	Number of COVID-19 Deaths	Flag
12255	08/30/2020	02/01/2020	08/29/2020	YC	Coronavirus Disease 2019	COVID-19	U071	65-74	5024.0	NaN
12256	08/30/2020	02/01/2020	08/29/2020	YC	Coronavirus Disease 2019	COVID-19	U071	75-84	5381.0	NaN

```
1 dataset3.groupby(["Condition"]).count()
```

Condition	Data as of	Start Week	End Week	State	Condition Group	ICD10_codes	Age Group	Number of COVID-19 Deaths	Flag
Adult respiratory distress syndrome	540	540	540	540	540	540	540	272	268
All other conditions and causes (residual)	540	540	540	540	540	540	540	363	177
Alzheimer disease	530	530	530	530	530	530	530	144	386
COVID-19	540	540	540	540	540	540	540	377	163
Cardiac arrest	520	520	520	520	520	520	520	219	301
Cardiac arrhythmia	540	540	540	540	540	540	540	192	348
Cerebrovascular diseases	530	530	530	530	530	530	530	187	343
Chronic lower respiratory diseases	540	540	540	540	540	540	540	229	311
Diabetes	540	540	540	540	540	540	540	276	264
Heart failure	540	540	540	540	540	540	540	204	336
Hypertensive diseases	540	540	540	540	540	540	540	264	276
Influenza and pneumonia	540	540	540	540	540	540	540	331	209
Intentional and unintentional injury, poisoning, and other adverse events	520	520	520	520	520	520	520	188	332
Ischemic heart disease	540	540	540	540	540	540	540	224	316
Malignant neoplasms	540	540	540	540	540	540	540	198	342
Obesity	530	530	530	530	530	530	530	182	348
Other diseases of the circulatory system	530	530	530	530	530	530	530	213	317
Other diseases of the respiratory system	540	540	540	540	540	540	540	188	352
Renal failure	540	540	540	540	540	540	540	238	302
Respiratory arrest	480	480	480	480	480	480	480	111	369
Respiratory failure	540	540	540	540	540	540	540	320	220
Sepsis	530	530	530	530	530	530	530	243	287
Vascular and unspecified dementia	530	530	530	530	530	530	530	191	339

```
# import word cloud
from wordcloud import WordCloud

sentences = dataset3["Condition"].tolist()
sentences_as_a_string = ' '.join(sentences)


# Convert the string into WordCloud
plt.figure(figsize=(20, 20))
plt.imshow(WordCloud().generate(sentences_as_a_string))
```

```
<matplotlib.image.AxesImage at 0x79c52fd73cd0>
```