

```
In [2]: #Data analysis and manipulation
import plotly.graph_objs as go
import plotly.io as pio
import plotly.express as px
import pandas as pd

#import visualization
import matplotlib.pyplot as plt

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

#initializing plotly
pio.renderers.default='notebook'
```

```
In [3]: # Importing Dataset1
dataset1 = pd.read_csv("covid.csv")
dataset1.head() # returns first 5 rows
```

Out[3]:

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	NewRecovered	ActiveCases
0	USA	North America	3.311981e+08	5032179	NaN	162804.0	NaN	2576668.0	NaN	49793127.0
1	Brazil	South America	2.127107e+08	2917562	NaN	98644.0	NaN	2047660.0	NaN	28188908.0
2	India	Asia	1.381345e+09	2025409	NaN	41638.0	NaN	1377384.0	NaN	19837627.0
3	Russia	Europe	1.459409e+08	871894	NaN	14606.0	NaN	676357.0	NaN	8542637.0
4	South Africa	Africa	5.938157e+07	538184	NaN	9604.0	NaN	387316.0	NaN	4404523.0

```
In [4]: # Returns tuple of shape (Rows, columns)
print(dataset1.shape)
```

```
# Returns size of dataframe  
print(dataset1.size)
```

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

```
In [5]: # 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
```

```
In [6]: # Importing Dataset2  
dataset2 = pd.read_csv("covid_grouped.csv")  
dataset2.head() # return first 5 rows of dataset2
```

Out[6]:

	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	2020-01-22	Algeria	0	0	0	0	0	0	0	Africa	DZA
3	2020-01-22	Andorra	0	0	0	0	0	0	0	Europe	AND
4	2020-01-22	Angola	0	0	0	0	0	0	0	Africa	AGO

In [7]:

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

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

(35156, 11)

386716

In [8]:

```
# 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
```

```
In [9]: # Columns Labels of a Dataset1
dataset1.columns
```

```
Out[9]: 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')
```

```
In [10]: # 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)
```

Out[10]:

	Country/Region	Continent	Population	TotalCases	TotalDeaths	TotalRecovered	ActiveCases	Serious,Critical	Tot Cases/1M pop
161	Tanzania	Africa	59886383.0	509	21.0	183.0	305.0	7.0	8.0
155	Lesotho	Africa	2143943.0	742	23.0	175.0	544.0	NaN	346.0
62	Venezuela	South America	28427499.0	22299	195.0	12146.0	9958.0	42.0	784.0
20	Iraq	Asia	40306025.0	140603	5161.0	101025.0	34417.0	517.0	3488.0
76	Bulgaria	Europe	6942854.0	13014	435.0	7374.0	5205.0	47.0	1874.0

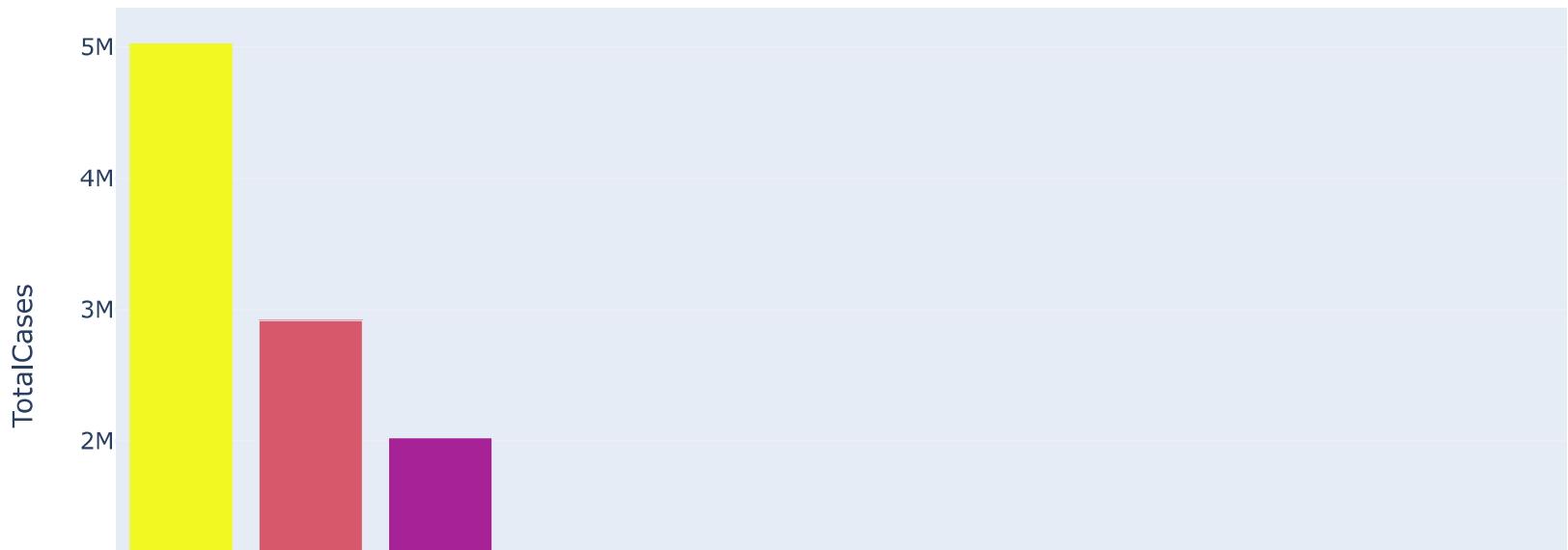
◀ ▶

In [24]:

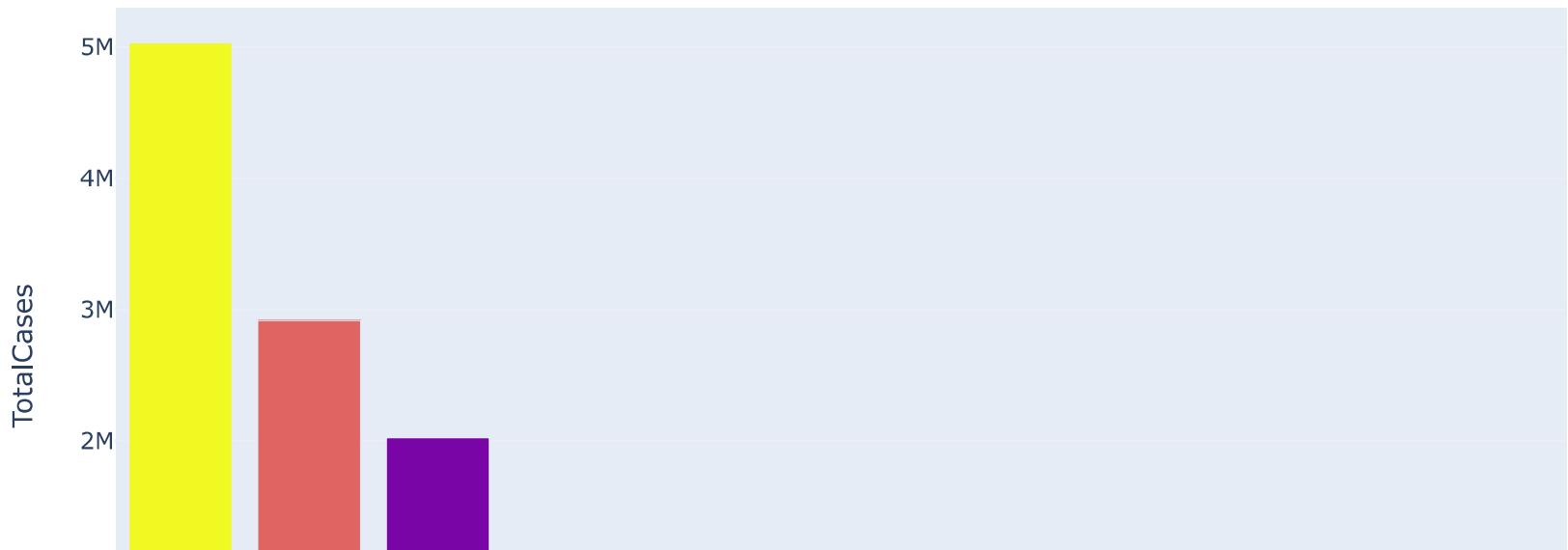
```
# Import create_table Figure Factory
import pandas as pd
import plotly.offline as py
py.init_notebook_mode(connected=True)
from plotly.figure_factory import create_table
colorscale = [[0, '#4d004c'], [.5, '#f2e5ff'], [1, '#ffffff']]
dataset1 = pd.read_csv("covid.csv")
table = create_table(dataset1.head(15), colorscale=colorscale)
py.iplot(table)
```

Country/Region	Continent	Population	TotalCases	NewCases	TotalDeath	NewDeaths	TotalRecovered	NewRecovered	ActiveCases	Serious, Critical Cases	TotalCases
USA	North America	3281198130.5	5032179	nan	162804.0	nan	2576668.0	nan	2292707.0	18296.0	15194.
Brazil	South America	212710692.0	1917562	nan	98644.0	nan	2047660.0	nan	771258.0	8318.0	13716.
India	Asia	1381344992.0	25409	nan	41638.0	nan	1377384.0	nan	606387.0	8944.0	1466.0
Russia	Europe	145940924.0	71894	nan	14606.0	nan	676357.0	nan	180931.0	2300.0	5974.0
South Africa	Africa	59381566.0	538184	nan	9604.0	nan	387316.0	nan	141264.0	539.0	9063.0
Mexico	North America	129066160.0	62690	6590.0	50517.0	819.0	308848.0	4140.0	103325.0	3987.0	3585.0
Peru	South America	32016319.0	455409	nan	20424.0	nan	310337.0	nan	124648.0	1426.0	13793.
Chile	South America	19132514.0	366671	nan	9889.0	nan	340168.0	nan	16614.0	1358.0	19165.
Colombia	South America	50936262.0	357710	nan	11939.0	nan	192355.0	nan	153416.0	1493.0	7023.0
Spain	Europe	46756648.0	354530	nan	28500.0	nan	nan	nan	nan	617.0	7582.0

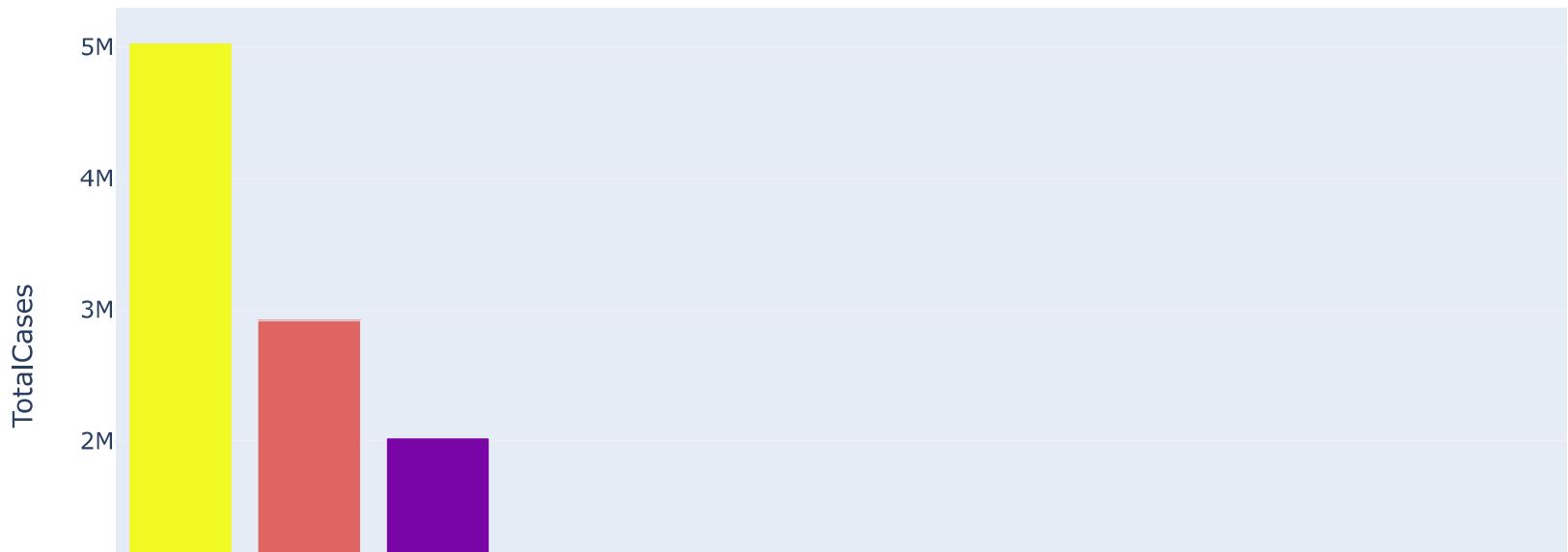
```
In [12]: px.bar(dataset1.head(15), x = 'Country/Region',
y = 'TotalCases', color = 'TotalCases',
height = 500, hover_data = ['Country/Region', 'Continent'])
```



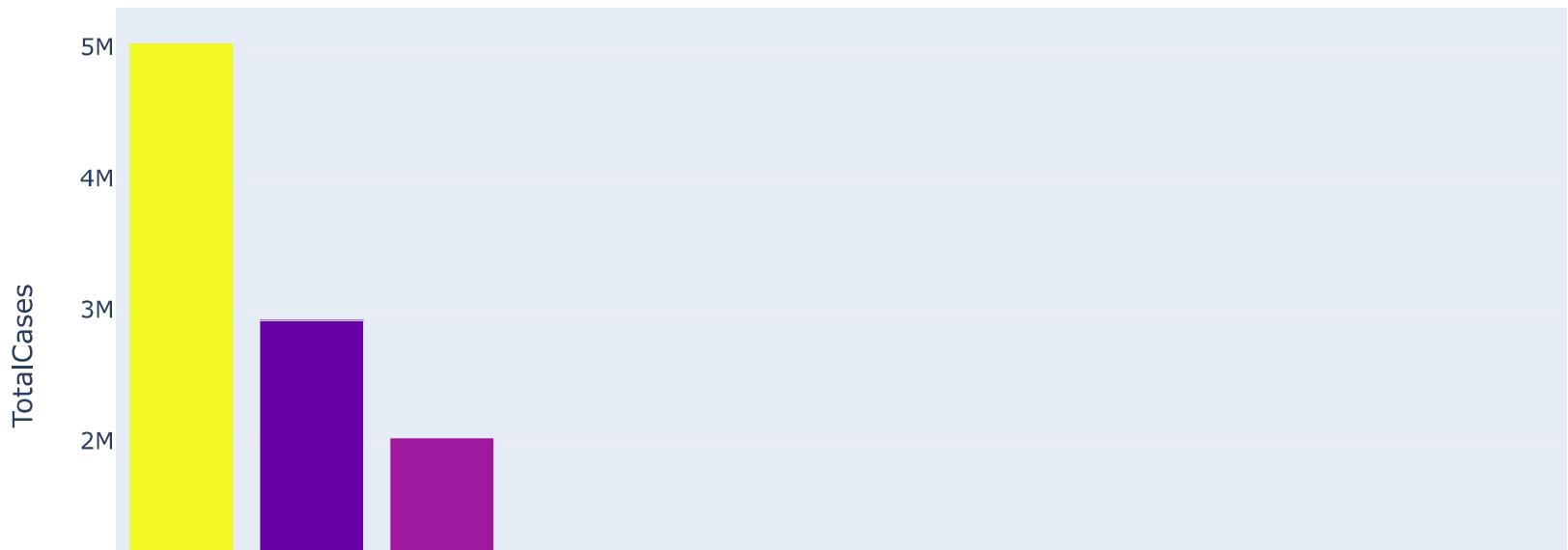
```
In [10]: px.bar(dataset1.head(15), x = 'Country/Region', y = 'TotalCases',
           color = 'TotalDeaths', height = 500,
           hover_data = ['Country/Region', 'Continent'])
```



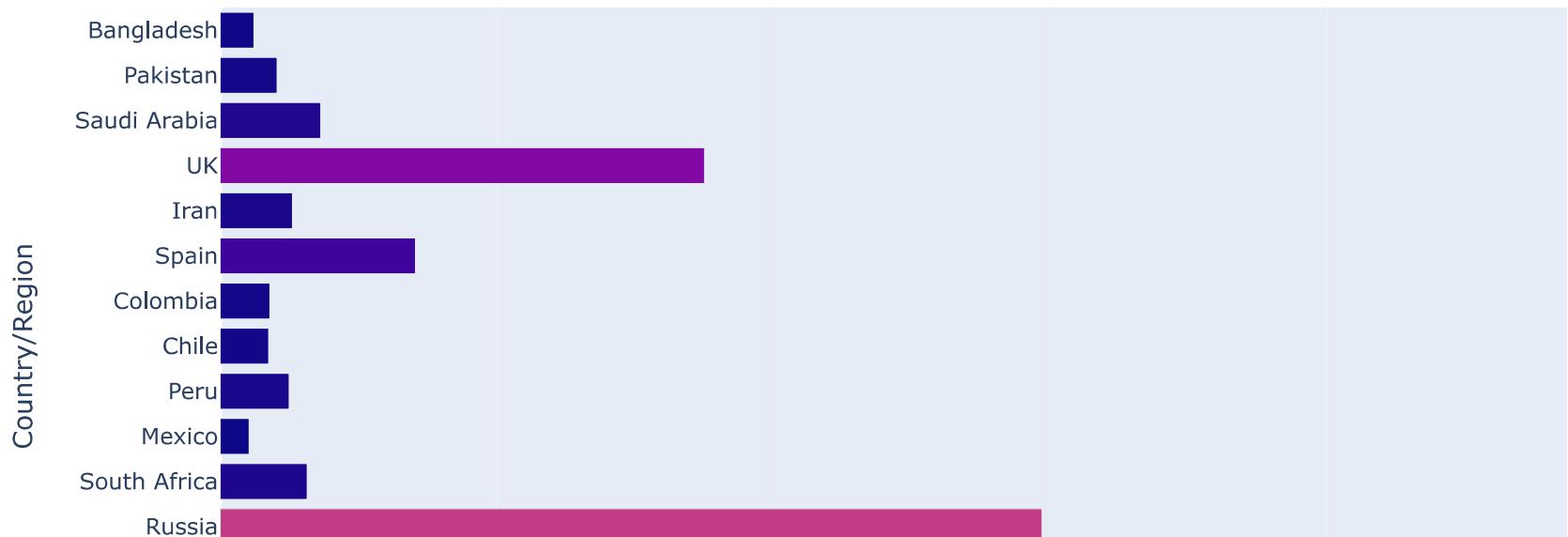
```
In [13]: px.bar(dataset1.head(15), x = 'Country/Region', y = 'TotalCases',
            color = 'TotalDeaths', height = 500,
            hover_data = ['Country/Region', 'Continent'])
```



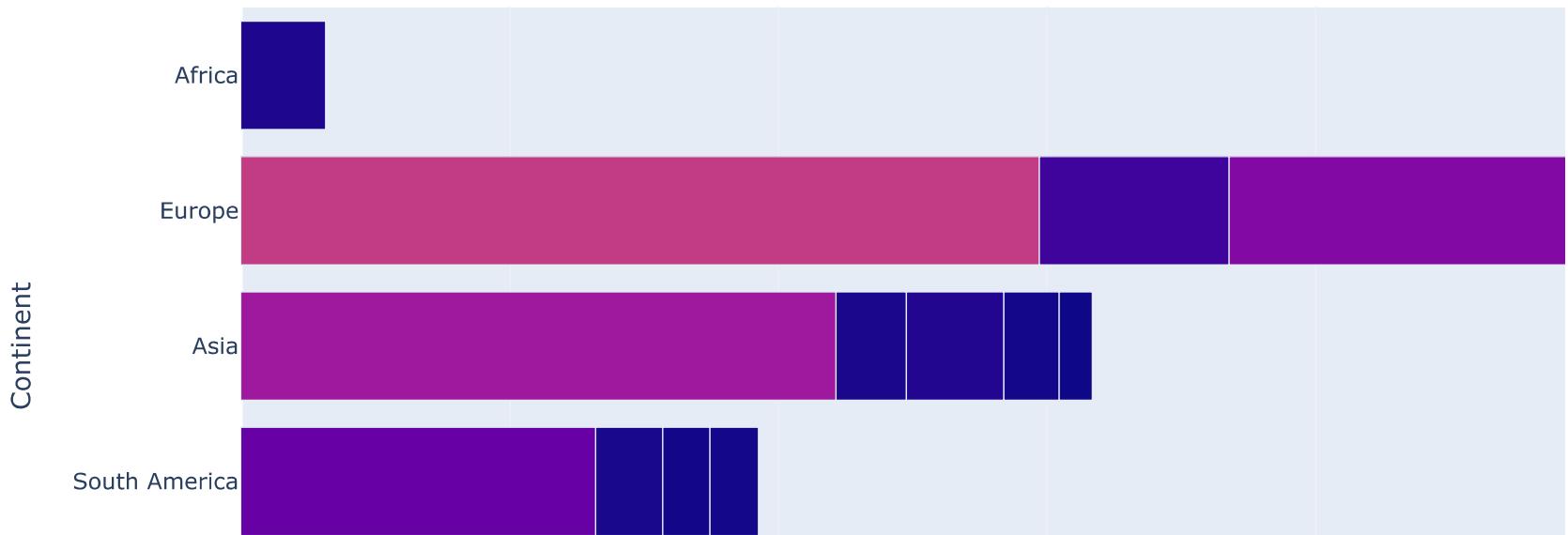
```
In [14]: px.bar(dataset1.head(15), x = 'Country/Region', y = 'TotalCases',
            color = 'TotalTests', height = 500, hover_data = ['Country/Region', 'Continent'])
```



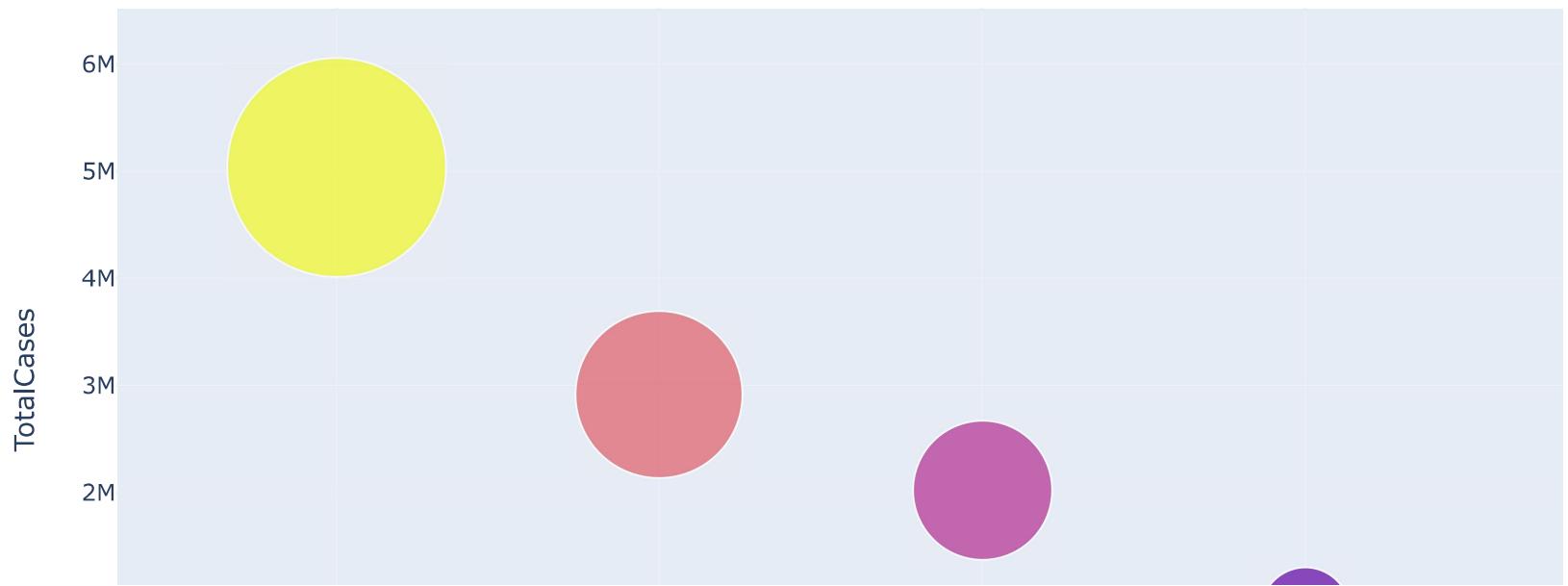
```
In [15]: px.bar(dataset1.head(15), x = 'TotalTests', y = 'Country/Region',
            color = 'TotalTests', orientation ='h', height = 500,
            hover_data = ['Country/Region', 'Continent'])
```



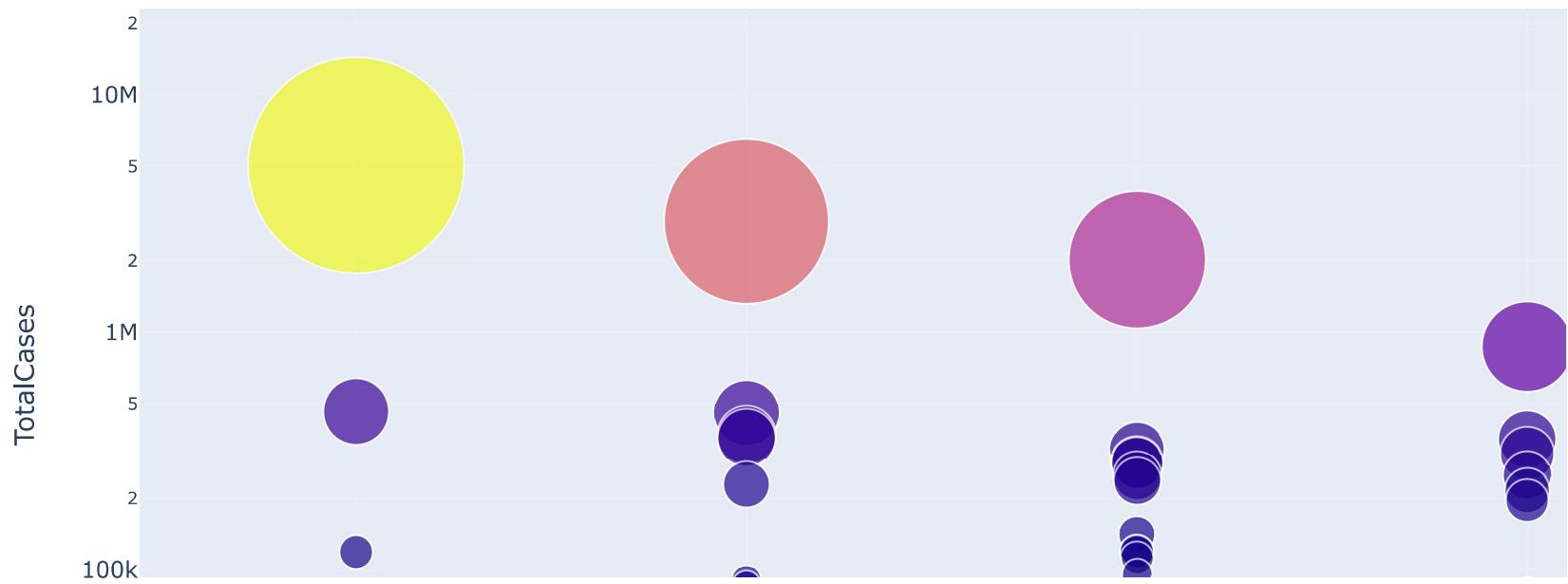
```
In [16]: px.bar(dataset1.head(15), x = 'TotalTests', y = 'Continent',
            color = 'TotalTests', orientation ='h', height = 500,
            hover_data = ['Country/Region', 'Continent'])
```



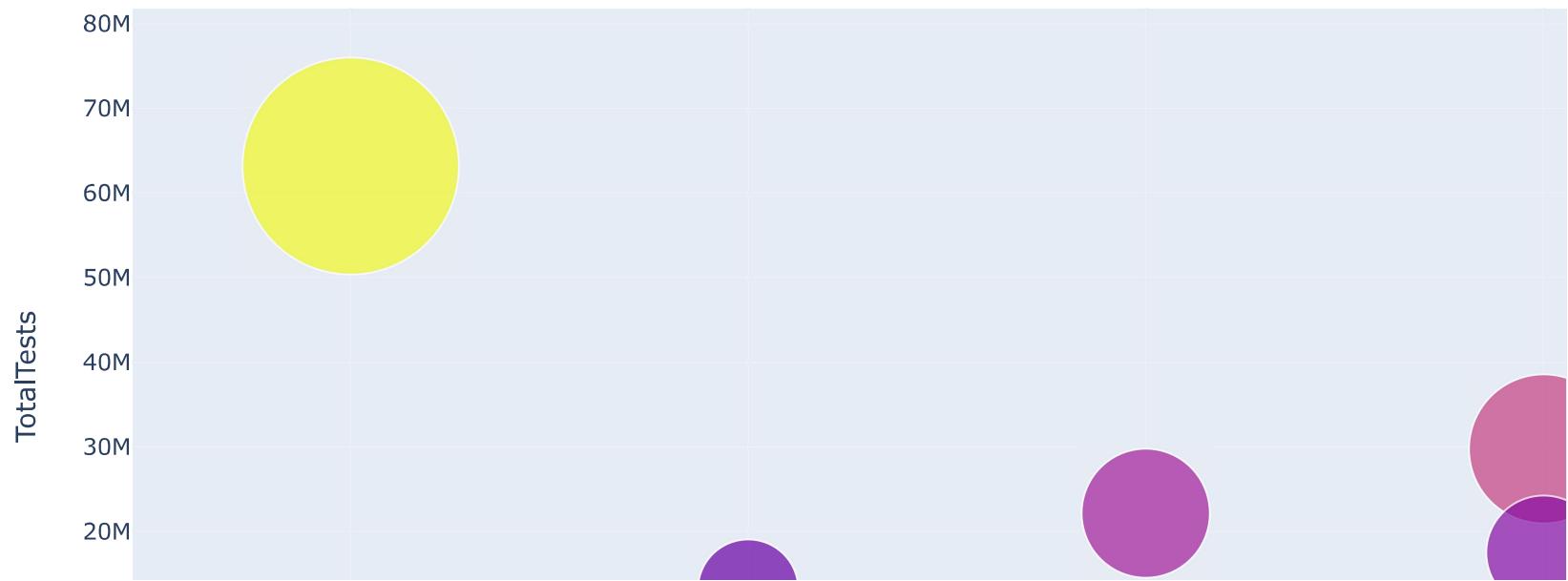
```
In [17]: px.scatter(dataset1, x='Continent',y='TotalCases',
                 hover_data=['Country/Region', 'Continent'],
                 color='TotalCases', size='TotalCases', size_max=80)
```



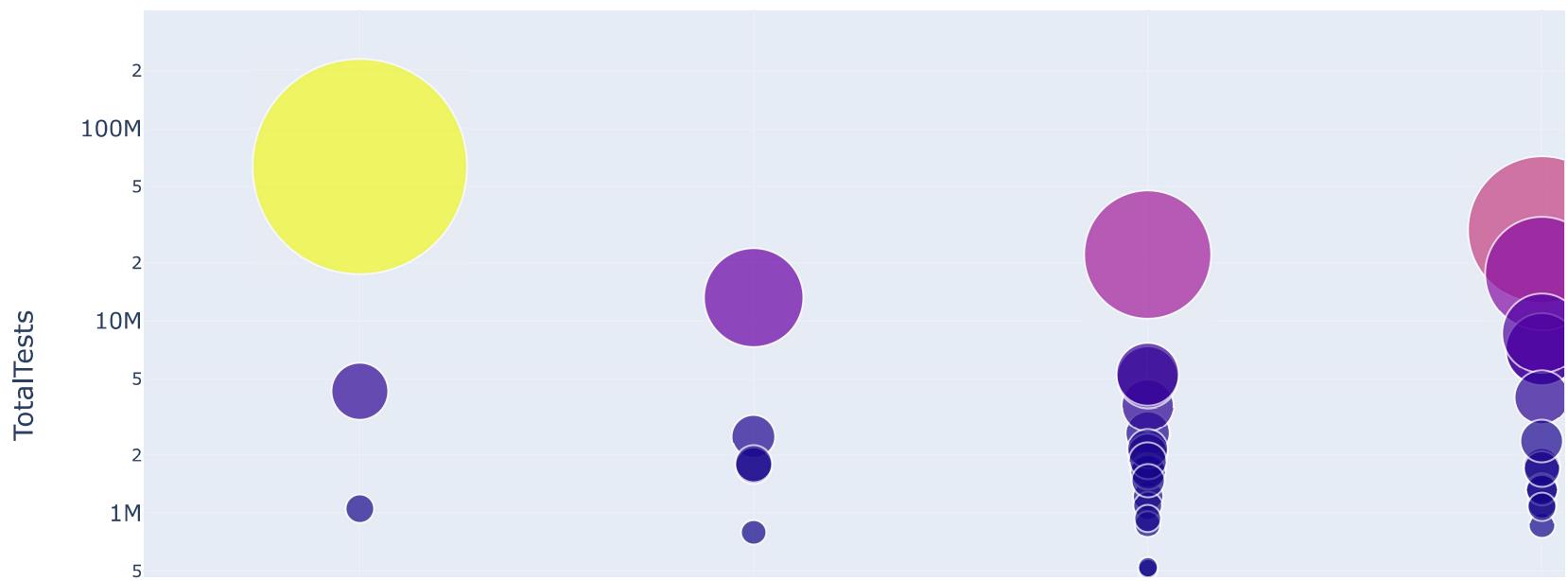
```
In [18]: px.scatter(dataset1.head(57), x='Continent',y='TotalCases',
                 hover_data=['Country/Region', 'Continent'],
                 color='TotalCases', size='TotalCases', size_max=80, log_y=True)
```



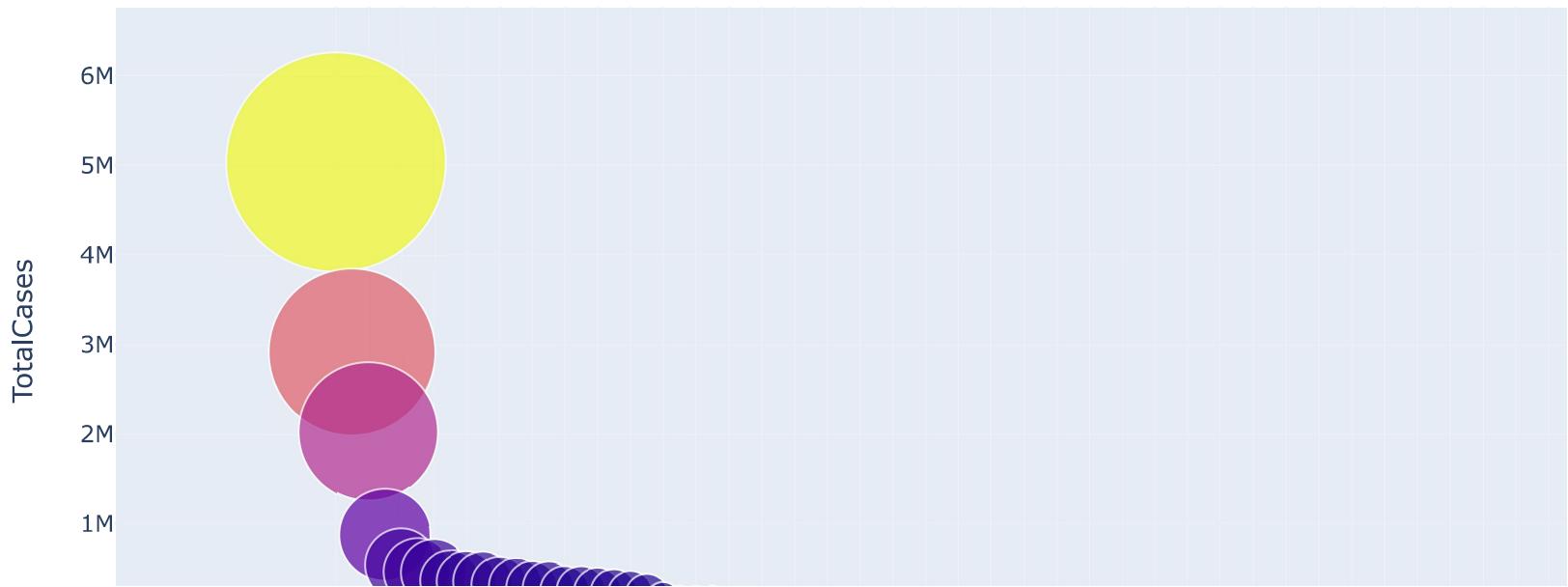
```
In [19]: px.scatter(dataset1.head(54), x='Continent',y='TotalTests',
                 hover_data=['Country/Region', 'Continent'],
                 color='TotalTests', size='TotalTests', size_max=80)
```



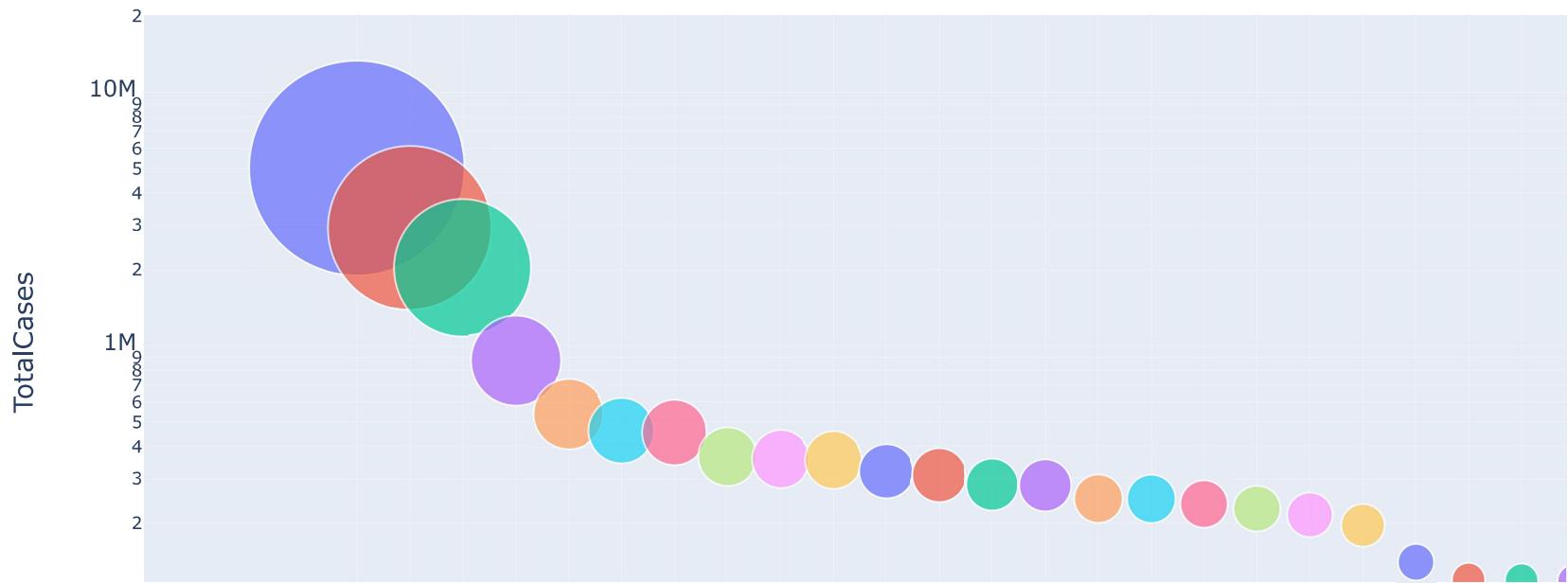
```
In [22]: px.scatter(dataset1.head(50), x='Continent', y='TotalTests',
                 hover_data=['Country/Region', 'Continent'],
                 color='TotalTests', size='TotalTests', size_max=80, log_y=True)
```



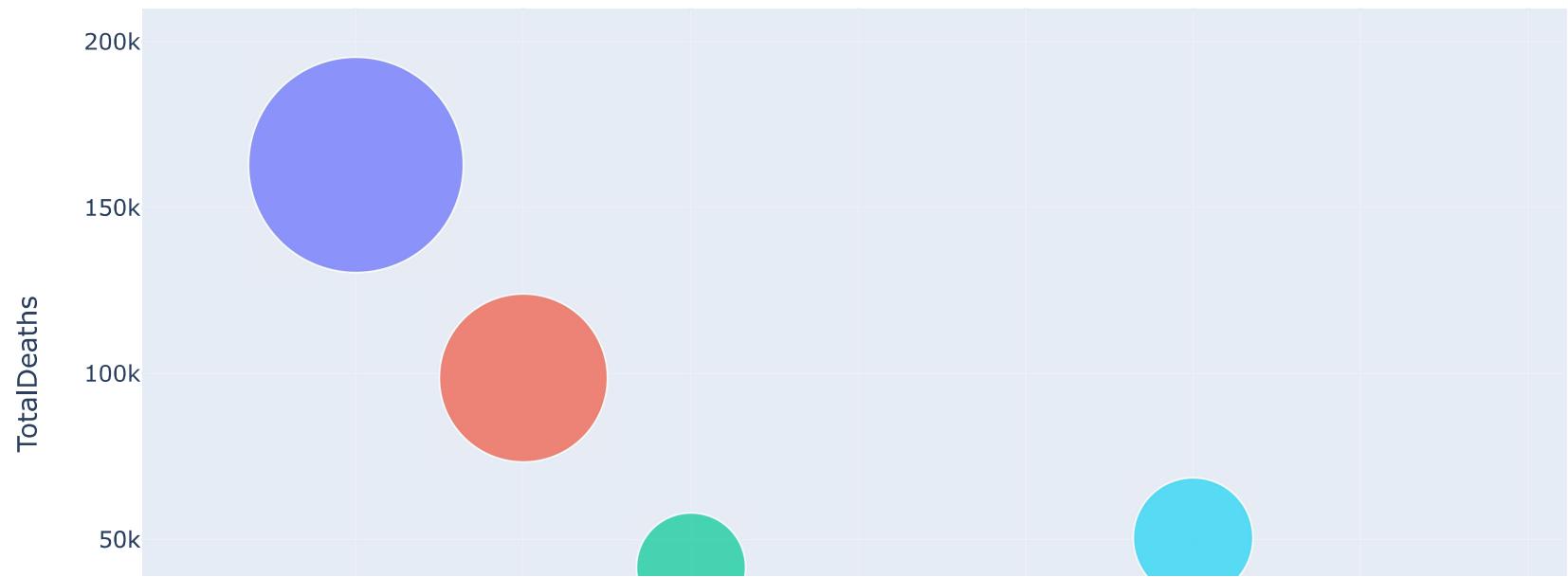
```
In [5]: import pandas as pd
import plotly.express as px
dataset1 = pd.read_csv("covid.csv")
px.scatter(dataset1.head(100), x='Country/Region', y='TotalCases',
           hover_data=['Country/Region', 'Continent'],
           color='TotalCases', size='TotalTests', size_max=80)
```



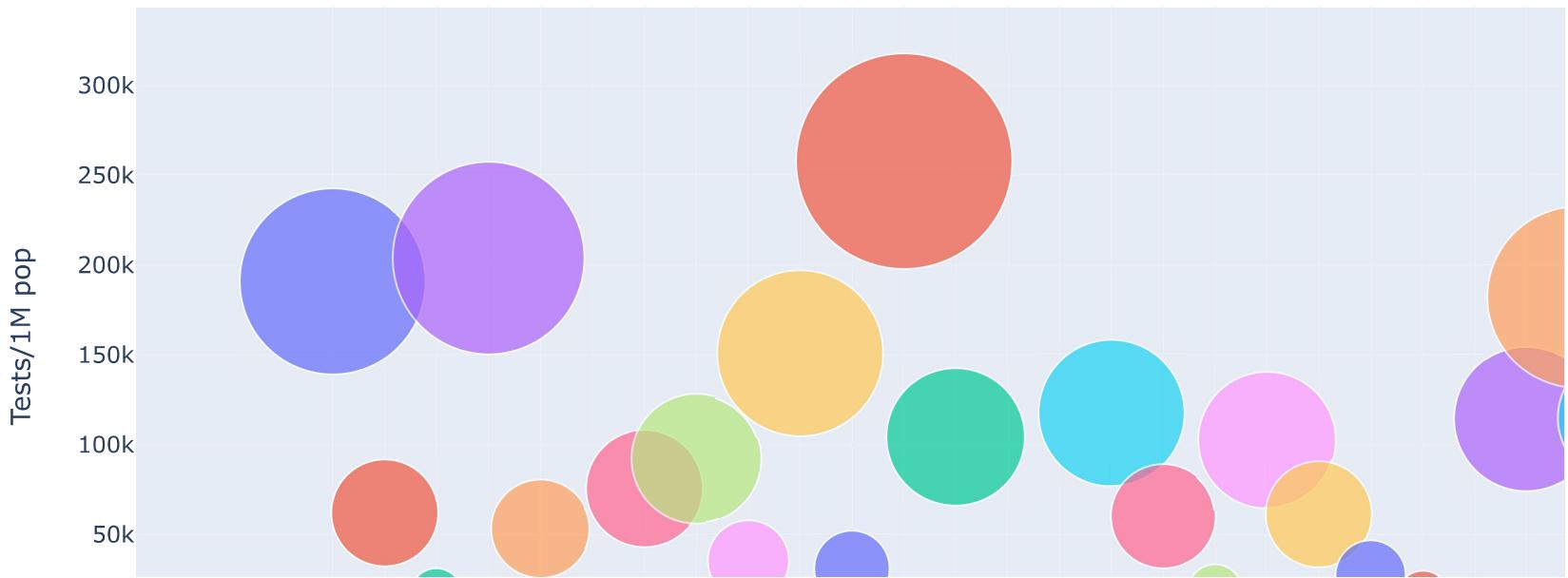
```
In [6]: px.scatter(dataset1.head(30), x='Country/Region', y='TotalCases',
               hover_data=['Country/Region', 'Continent'],
               color='Country/Region', size='TotalCases', size_max=80, log_y=True)
```



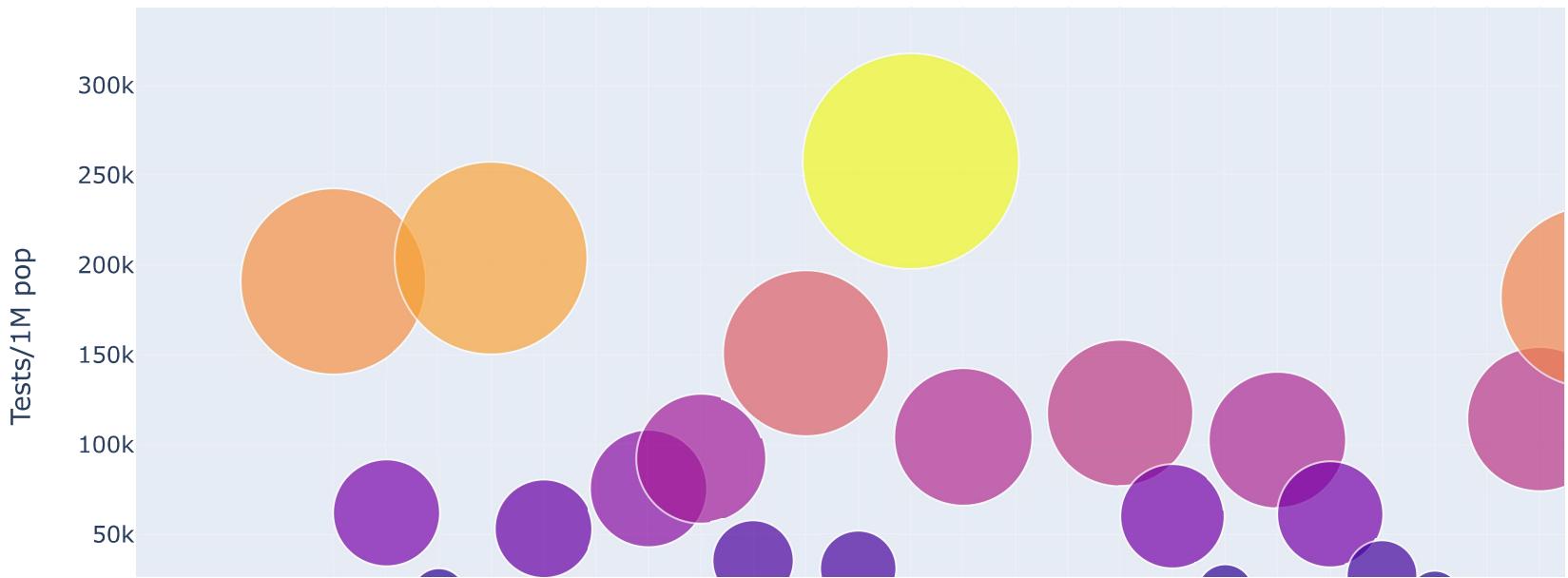
```
In [7]: px.scatter(dataset1.head(10), x='Country/Region', y= 'TotalDeaths',
               hover_data=['Country/Region', 'Continent'],
               color='Country/Region', size= 'TotalDeaths', size_max=80)
```



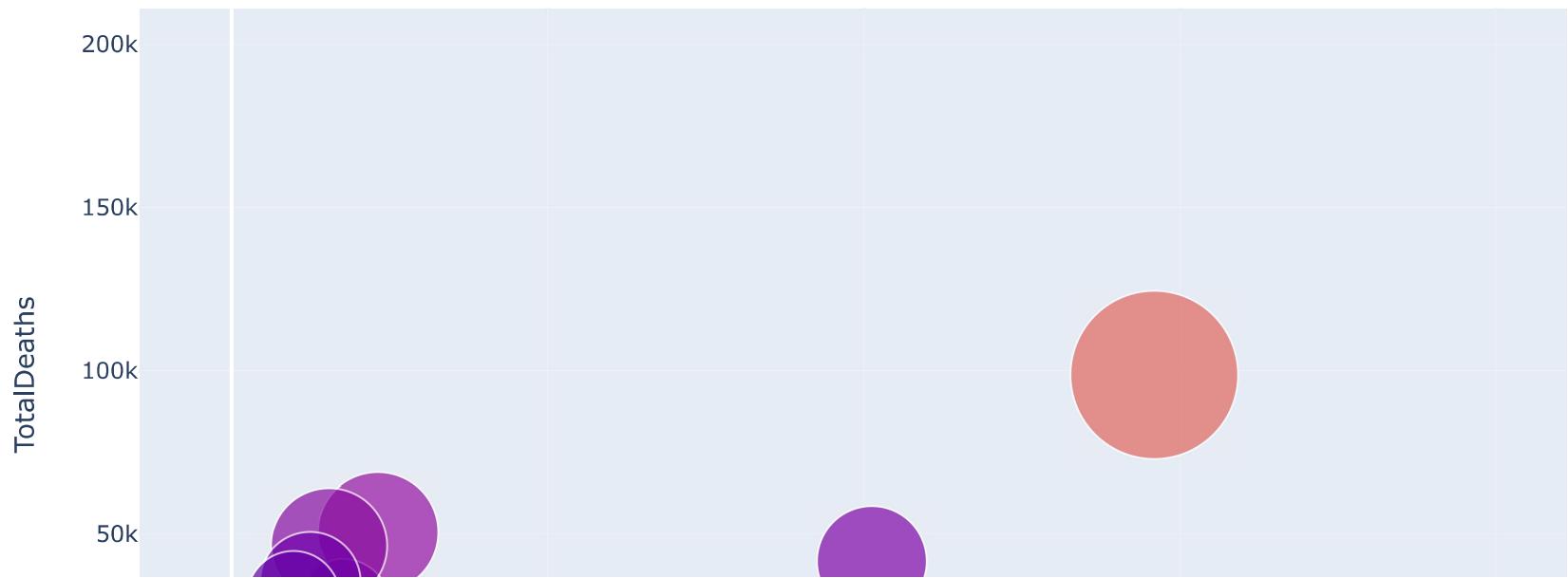
```
In [8]: px.scatter(dataset1.head(30), x='Country/Region', y= 'Tests/1M pop',
               hover_data=['Country/Region', 'Continent'],
               color='Country/Region', size= 'Tests/1M pop', size_max=80)
```



```
In [9]: 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)
```



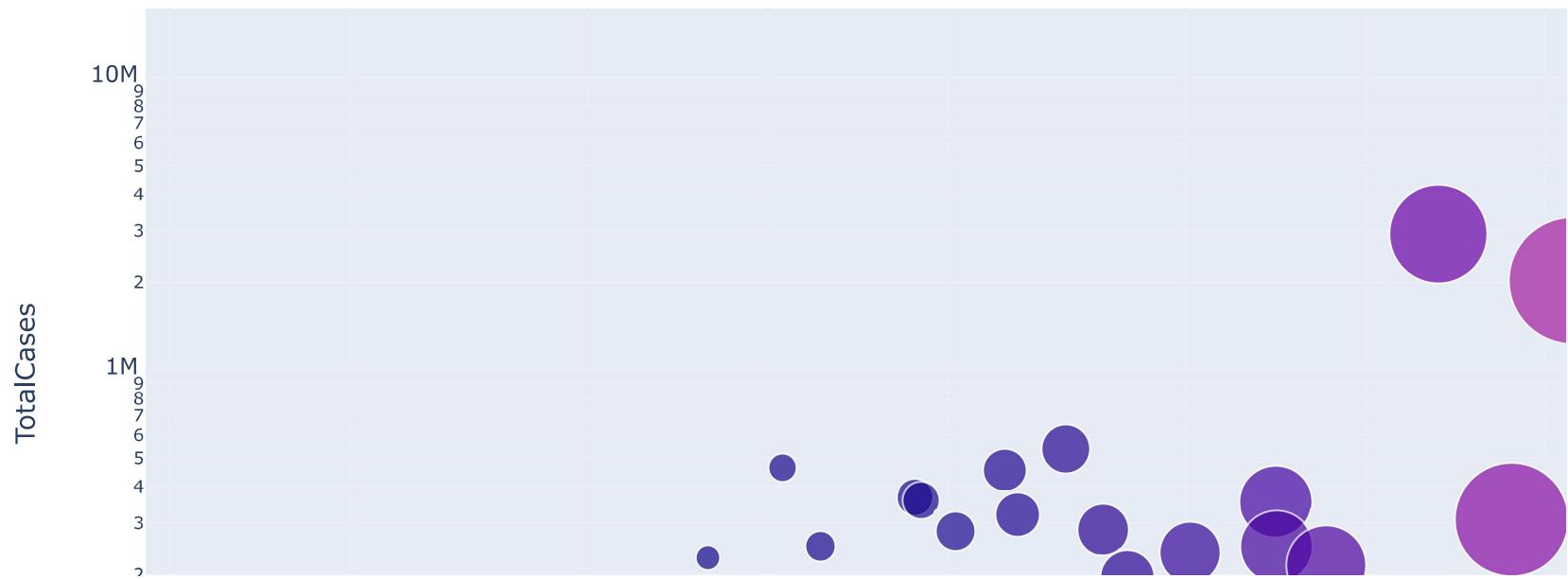
```
In [10]: px.scatter(dataset1.head(30), x='TotalCases', y= 'TotalDeaths',
                 hover_data=['Country/Region', 'Continent'],
                 color='TotalDeaths', size= 'TotalDeaths', size_max=80)
```



```
In [11]: 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)
```



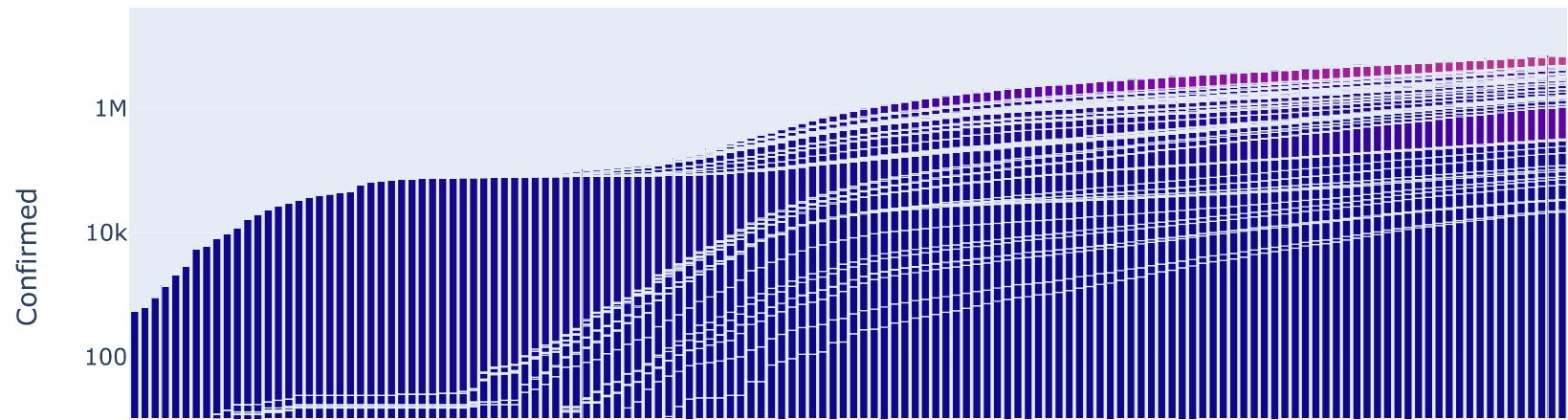
```
In [12]: px.scatter(dataset1.head(30), x='TotalTests', y= 'TotalCases',
                 hover_data=['Country/Region', 'Continent'],
                 color='TotalTests', size= 'TotalTests', size_max=80,
                 log_x=True, log_y=True)
```



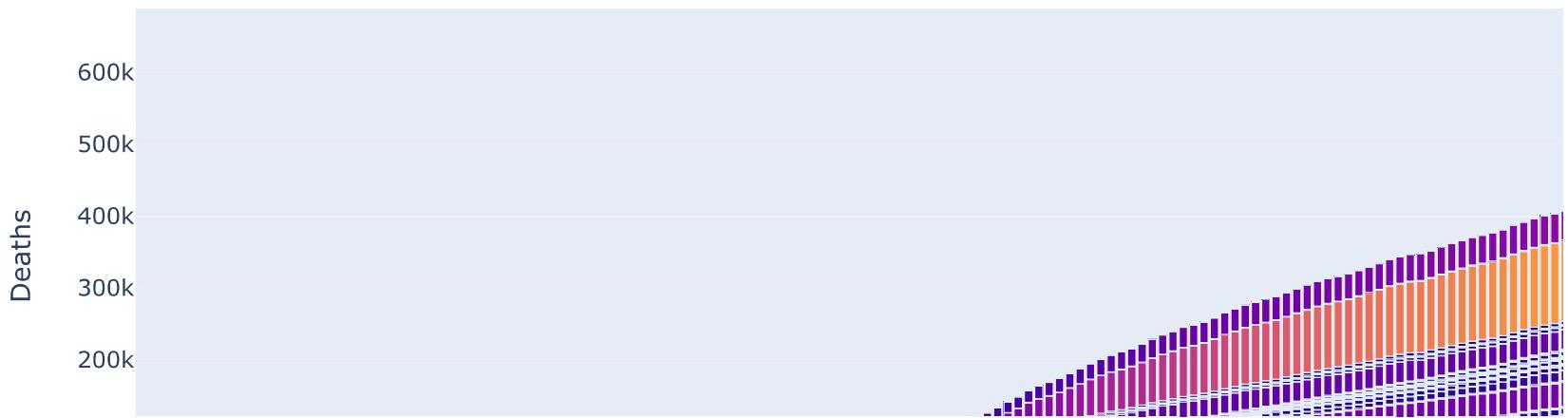
```
In [14]: dataset2 = pd.read_csv("covid_grouped.csv")
px.bar(dataset2, x="Date", y="Confirmed", color="Confirmed",
       hover_data=["Confirmed", "Date", "Country/Region"], height=400)
```



```
In [15]: px.bar(dataset2, x="Date", y="Confirmed", color="Confirmed",
            hover_data=["Confirmed", "Date", "Country/Region"], log_y=True, height=400)
```

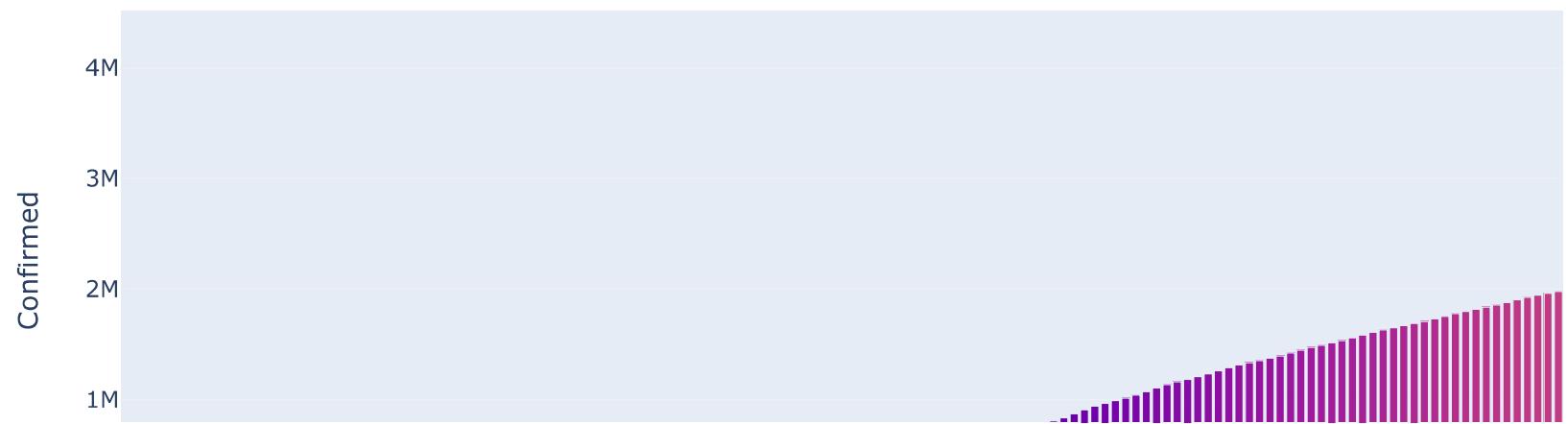


```
In [16]: px.bar(dataset2, x="Date", y="Deaths", color="Deaths",
            hover_data=["Confirmed", "Date", "Country/Region"],
            log_y=False, height=400)
```

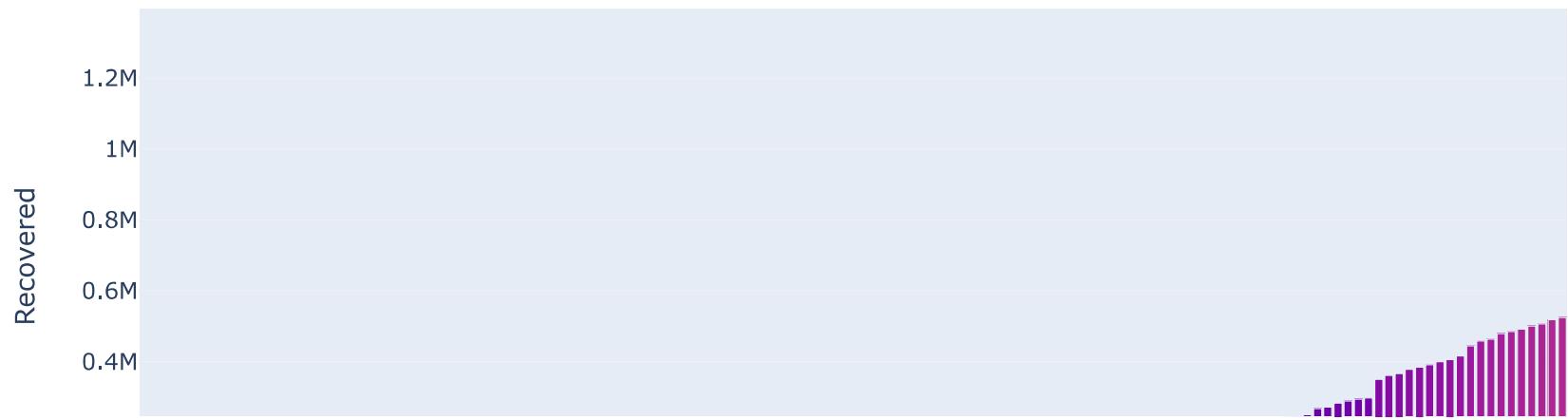


```
In [17]: df_US= dataset2.loc[dataset2["Country/Region"]=="US"]
```

```
In [18]: px.bar(df_US, x="Date", y="Confirmed", color="Confirmed", height=400)
```



```
In [19]: px.bar(df_US,x="Date", y="Recovered", color="Recovered", height=400)
```



```
In [20]: px.line(df_US,x="Date", y="Recovered", height=400)
```

```
In [21]: px.line(df_US,x="Date", y="Deaths", height=400)
```

```
In [22]: px.line(df_US,x="Date", y="Confirmed", height=400)
```

```
In [23]: px.line(df_US,x="Date", y="New cases", height=400)
```

```
In [24]: px.bar(df_US,x="Date", y="New cases", height=400)
```

```
In [25]: px.scatter(df_US, x="Confirmed", y="Deaths", height=400)
```

```
In [26]: px.choropleth(dataset2,
                      locations="iso_alpha",
                      color="Confirmed",
                      hover_name="Country/Region",
                      color_continuous_scale="Blues",
                      animation_frame="Date")
```

```
In [27]: px.choropleth(dataset2,
                     locations='iso_alpha',
                     color="Deaths",
                     hover_name="Country/Region",
                     color_continuous_scale="Viridis",
                     animation_frame="Date" )
```

