In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler

from scipy import stats
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:
```
df=pd.read_csv("C:/Users/arvin/OneDrive/Desktop/COVID19/country_wise_latest.csv")
df.head()
```

Out[2]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	R
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	
1	Albania	4880	144	2745	1991	117	6	63	2.95	
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	
3	Andorra	907	52	803	52	10	0	0	5.73	
4	Angola	950	41	242	667	18	1	0	4.32	

In [3]:
```
df2=pd.read_csv("C:/Users/arvin/OneDrive/Desktop/COVID19/full_grouped.csv",par
se_dates=["Date"])
df2.head()
df2.sort_values(by=["Confirmed"],ascending=False)
```

Out[3]:

	Date	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered
35142	2020-07-27	US	4290259	148011	1325804	2816444	56336	1076	2794
34955	2020-07-26	US	4233923	146935	1297863	2789125	54953	470	1844
34768	2020-07-25	US	4178970	146465	1279414	2753091	66439	905	1779
34581	2020-07-24	US	4112531	145560	1261624	2705347	73715	1130	2835
34394	2020-07-23	US	4038816	144430	1233269	2661117	68695	1114	2242
...	...	...	...	...	...	...	...	...	...
5009	2020-02-17	Serbia	0	0	0	0	0	0	0
5008	2020-02-17	Senegal	0	0	0	0	0	0	0
5007	2020-02-17	Saudi Arabia	0	0	0	0	0	0	0
5006	2020-02-17	Sao Tome and Principe	0	0	0	0	0	0	0
0	2020-01-22	Afghanistan	0	0	0	0	0	0	0

35156 rows × 10 columns

In [4]:
```
df1=pd.read_csv("C:/Users/arvin/OneDrive/Desktop/COVID19/covid_19_clean_complete.csv", parse_dates=['Date'])
df1.head()
```

Out[4]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	A
0	NaN	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0	
1	NaN	Albania	41.15330	20.168300	2020-01-22	0	0	0	
2	NaN	Algeria	28.03390	1.659600	2020-01-22	0	0	0	
3	NaN	Andorra	42.50630	1.521800	2020-01-22	0	0	0	
4	NaN	Angola	-11.20270	17.873900	2020-01-22	0	0	0	

In [5]:
```
df.head()
```

Out[5]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	R
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	
1	Albania	4880	144	2745	1991	117	6	63	2.95	
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	
3	Andorra	907	52	803	52	10	0	0	5.73	
4	Angola	950	41	242	667	18	1	0	4.32	

In [6]:
```
df.shape
```

Out[6]:
```
(187, 15)
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Country/Region         187 non-null    object 
 1   Confirmed              187 non-null    int64  
 2   Deaths                 187 non-null    int64  
 3   Recovered              187 non-null    int64  
 4   Active                 187 non-null    int64  
 5   New cases              187 non-null    int64  
 6   New deaths             187 non-null    int64  
 7   New recovered          187 non-null    int64  
 8   Deaths / 100 Cases     187 non-null    float64
 9   Recovered / 100 Cases  187 non-null    float64
 10  Deaths / 100 Recovered  187 non-null    float64
 11  Confirmed last week    187 non-null    int64  
 12  1 week change          187 non-null    int64  
 13  1 week % increase      187 non-null    float64
 14  WHO Region             187 non-null    object 
dtypes: float64(4), int64(9), object(2)
memory usage: 22.0+ KB
```

In [8]:
```
df.describe(include="all")
```

Out[8]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases
count	187	1.870000e+02	187.000000	1.870000e+02	1.870000e+02	187.000000
unique	187	NaN	NaN	NaN	NaN	NaN
top	Grenada	NaN	NaN	NaN	NaN	NaN
freq	1	NaN	NaN	NaN	NaN	NaN
mean	NaN	8.813094e+04	3497.518717	5.063148e+04	3.400194e+04	1222.957219
std	NaN	3.833187e+05	14100.002482	1.901882e+05	2.133262e+05	5710.374790
min	NaN	1.000000e+01	0.000000	0.000000e+00	0.000000e+00	0.000000
25%	NaN	1.114000e+03	18.500000	6.265000e+02	1.415000e+02	4.000000
50%	NaN	5.059000e+03	108.000000	2.815000e+03	1.600000e+03	49.000000
75%	NaN	4.046050e+04	734.000000	2.260600e+04	9.149000e+03	419.500000
max	NaN	4.290259e+06	148011.000000	1.846641e+06	2.816444e+06	56336.000000

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Country/Region           0
Confirmed                0
Deaths                   0
Recovered                0
Active                   0
New cases                0
New deaths               0
New recovered            0
Deaths / 100 Cases       0
Recovered / 100 Cases    0
Deaths / 100 Recovered   0
Confirmed last week      0
1 week change            0
1 week % increase        0
WHO Region               0
dtype: int64
```

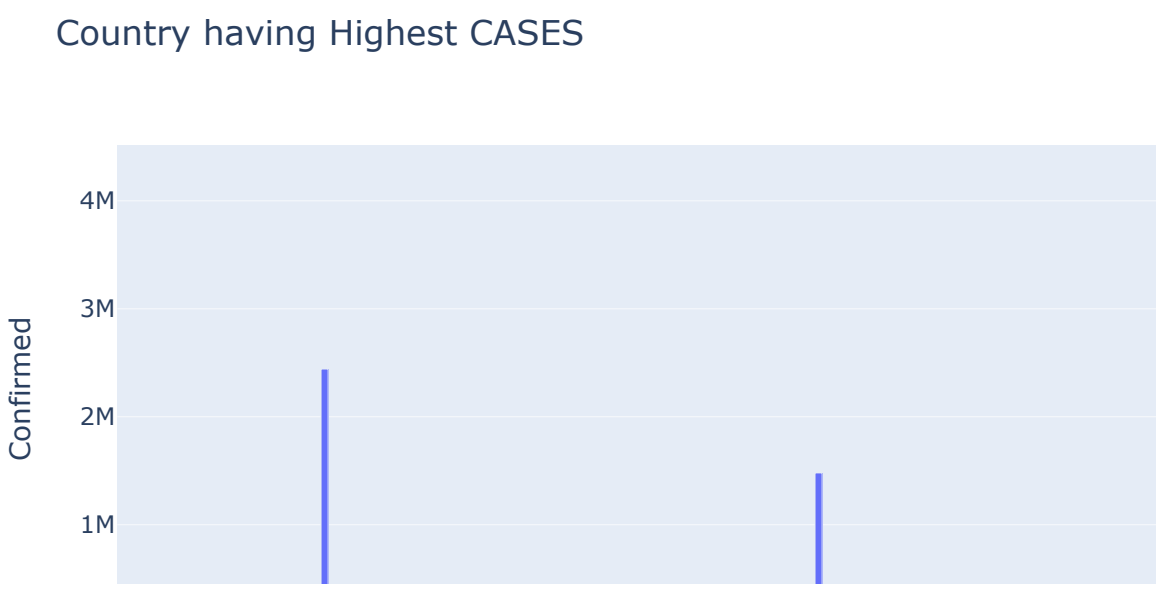
In [10]:
```
grouped=df[["Confirmed","Deaths","Recovered","Country/Region"]]
grouped.head()
```

Out[10]:

	Confirmed	Deaths	Recovered	Country/Region
0	36263	1269	25198	Afghanistan
1	4880	144	2745	Albania
2	27973	1163	18837	Algeria
3	907	52	803	Andorra
4	950	41	242	Angola

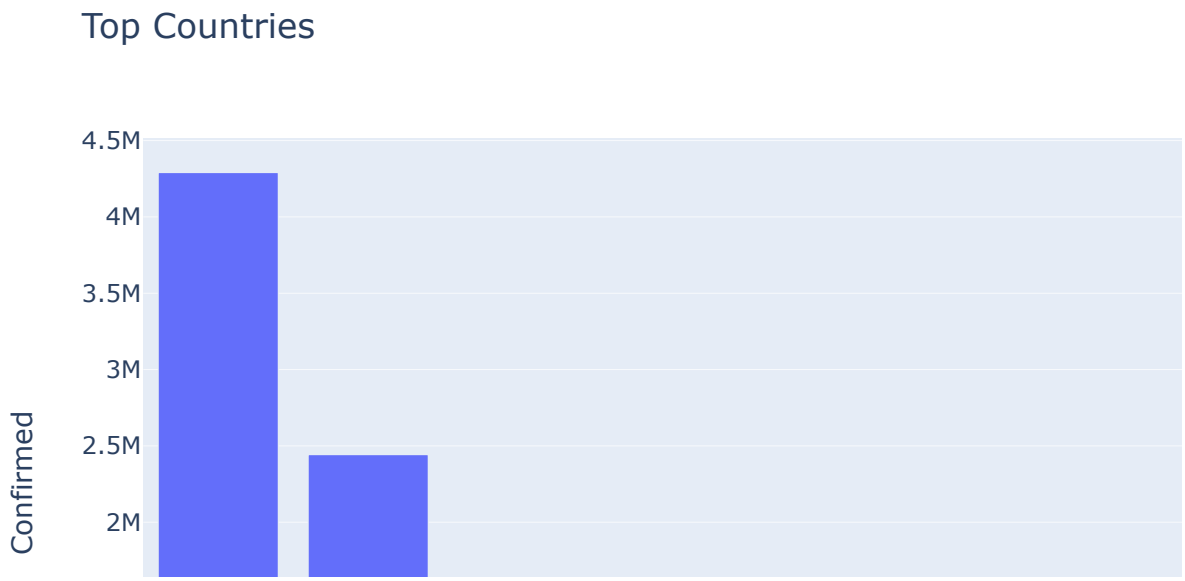
In [11]:
```
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
import plotly.express as px
```

In [12]:
```
fig=px.bar(grouped,y="Confirmed",x="Country/Region",title="Country having Highest CASES")

fig.show()
```

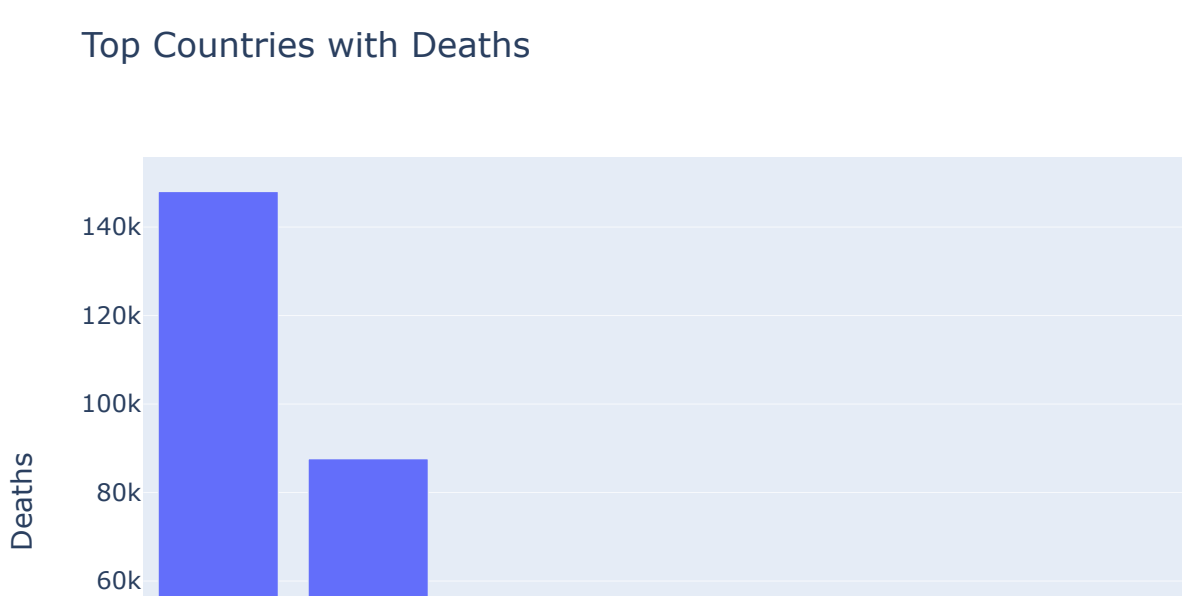


In [13]:
```
fig=px.bar(grouped,x="Country/Region",y="Deaths",title="Countries Having Highe
st Deaths", color_continuous_scale="Brand")
fig.show()
```

In [14]:
```
x=grouped.sort_values(by="Confirmed",ascending=False)

px.bar(x[0:11],x="Country/Region",y="Confirmed",title="Top Countries")
```



In [15]:
```
px.bar(x[0:11],x="Country/Region",y="Deaths",title="Top Countries with Deaths"
)
```



In [16]:
```
grouped["Date"]=df2.Date
grouped.head()
```

Out[16]:

	Confirmed	Deaths	Recovered	Country/Region	Date
0	36263	1269	25198	Afghanistan	2020-01-22
1	4880	144	2745	Albania	2020-01-22
2	27973	1163	18837	Algeria	2020-01-22
3	907	52	803	Andorra	2020-01-22
4	950	41	242	Angola	2020-01-22

In [17]:
```
grouped.set_index("Date")
grouped.head()
```

Out[17]:

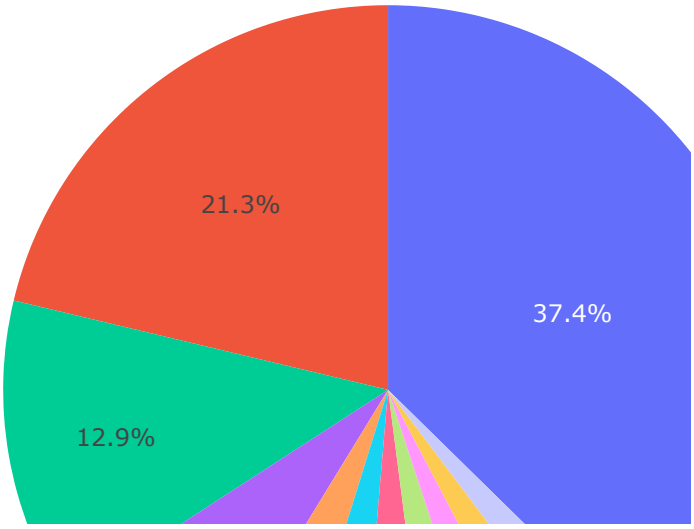
	Confirmed	Deaths	Recovered	Country/Region	Date
0	36263	1269	25198	Afghanistan	2020-01-22
1	4880	144	2745	Albania	2020-01-22
2	27973	1163	18837	Algeria	2020-01-22
3	907	52	803	Andorra	2020-01-22
4	950	41	242	Angola	2020-01-22

In [18]:
```
grouped["TOTAL"]=grouped["Confirmed"]+grouped['Deaths']+grouped["Recovered"]
grouped.head()
```

Out[18]:

	Confirmed	Deaths	Recovered	Country/Region	Date	TOTAL
0	36263	1269	25198	Afghanistan	2020-01-22	62730
1	4880	144	2745	Albania	2020-01-22	7769
2	27973	1163	18837	Algeria	2020-01-22	47973
3	907	52	803	Andorra	2020-01-22	1762
4	950	41	242	Angola	2020-01-22	1233

In [19]:
```
px.pie(x[0:11],values="Confirmed",names="Country/Region")
```

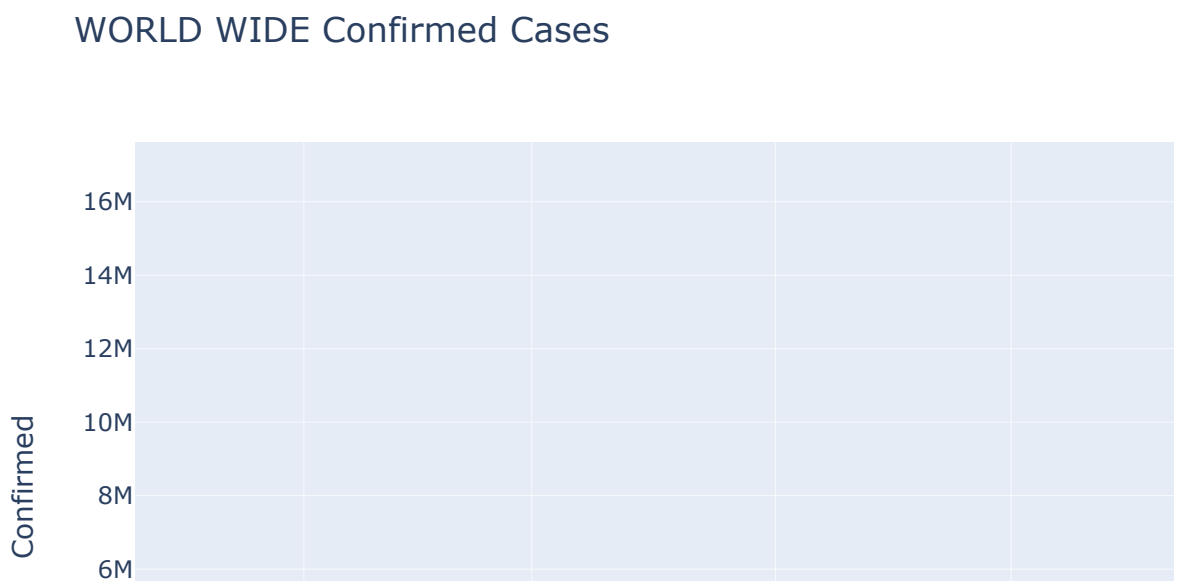


In [20]:
```
date_c = df1.groupby('Date')['Date', 'Confirmed', 'Deaths',"Lat","Long","Country/Region"].sum().reset_index()
date_c.head()
```

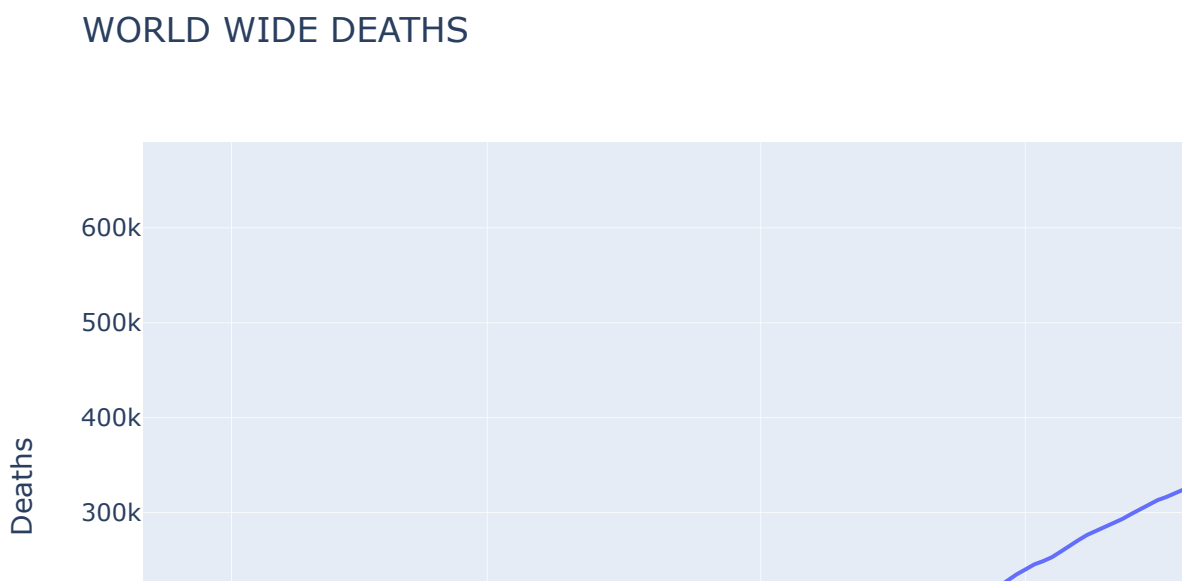
Out[20]:

	Date	Confirmed	Deaths	Lat	Long
0	2020-01-22	555	17	5594.20365	6140.869714
1	2020-01-23	654	18	5594.20365	6140.869714
2	2020-01-24	941	26	5594.20365	6140.869714
3	2020-01-25	1434	42	5594.20365	6140.869714
4	2020-01-26	2118	56	5594.20365	6140.869714

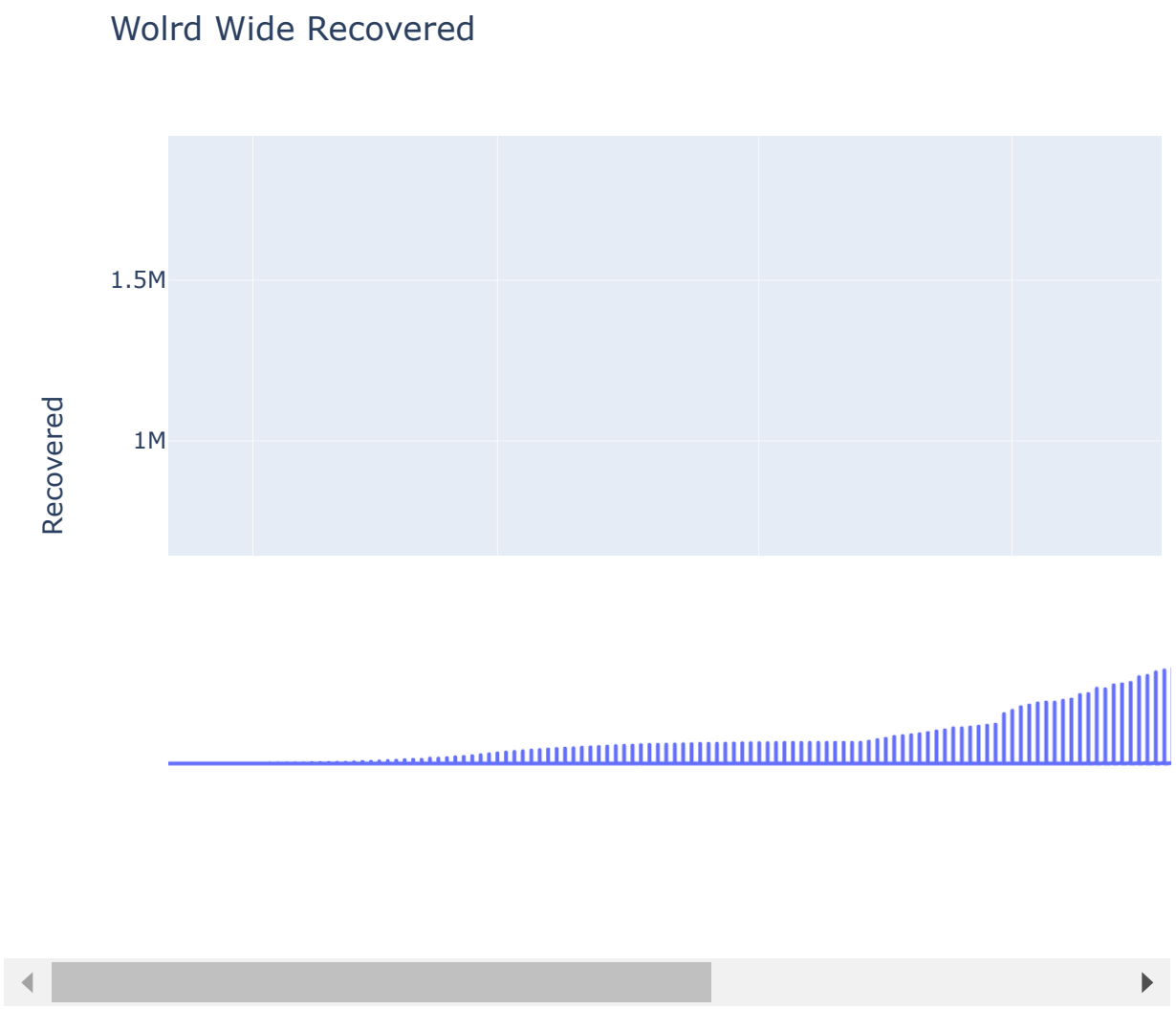
In [21]:
```
px.scatter(date_c,x="Date",y="Confirmed",title="WORLD WIDE Confirmed Cases ")
```



In [22]:
```
px.line(date_c,x="Date",y="Deaths",title="WORLD WIDE DEATHS")
```



In [23]:
```
px.line(df1,x="Date",y="Recovered",title="Wolrd Wide Recovered")
```



In [ ]:

In [ ]: