

Importing Python Libraries

```
In [90]: import glob
import random
import seaborn as sns
import calmap

import holoviews as hv
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
import plotly.express as px
import datashader as ds
import matplotlib.pyplot as plt
import numpy as np
from plotly.subplots import make_subplots
import pandas as pd
# get data file names

path =r'D:\Techstack\Covid Data\Project On Covid\csse_covid_19_daily_reports'

filenames = glob.glob(path + "/*.csv")

dfs = []

for filename in filenames:
    dfs.append(pd.read_csv(filename))

# Concatenate all data into one DataFrame

Df_covid = pd.concat(dfs, ignore_index=True)
```

```
In [91]: Df_covid.head()
```

```
Out[91]:
```

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	...	Combined_Key	Incident_F
0	NaN	NaN	NaN	Afghanistan	2021-01-02 05:22:33	33.93911	67.709953	51526.0	2191.0	41727.0	...	Afghanistan	0.000
1	NaN	NaN	NaN	Albania	2021-01-02 05:22:33	41.15330	20.168300	58316.0	1181.0	33634.0	...	Albania	2026.409
2	NaN	NaN	NaN	Algeria	2021-01-02 05:22:33	28.03390	1.659600	99897.0	2762.0	67395.0	...	Algeria	227.809
3	NaN	NaN	NaN	Andorra	2021-01-02 05:22:33	42.50630	1.521800	8117.0	84.0	7463.0	...	Andorra	10505.403
4	NaN	NaN	NaN	Angola	2021-01-02 05:22:33	-11.20270	17.873900	17568.0	405.0	11146.0	...	Angola	53.452

5 rows × 21 columns



Data Preprocessing

```
In [92]: Df_covid = Df_covid.rename(columns={"Last_Update":"Date"})
```

```
In [93]: Df_covid["Date"] = pd.to_datetime(Df_covid["Date"]).dt.date
```

```
In [94]: Df_covid["Date"] = pd.to_datetime(Df_covid["Date"], infer_datetime_format=True)
```

In [95]: Df_covid.head()

Out[95]:

	FIPS	Admin2	Province_State	Country_Region	Date	Lat	Long_	Confirmed	Deaths	Recovered	...	Combined_Key	Incident_Rate
0	NaN	NaN	NaN	Afghanistan	2021-01-02	33.93911	67.709953	51526.0	2191.0	41727.0	...	Afghanistan	0.000000
1	NaN	NaN	NaN	Albania	2021-01-02	41.15330	20.168300	58316.0	1181.0	33634.0	...	Albania	2026.409062
2	NaN	NaN	NaN	Algeria	2021-01-02	28.03390	1.659600	99897.0	2762.0	67395.0	...	Algeria	227.809861
3	NaN	NaN	NaN	Andorra	2021-01-02	42.50630	1.521800	8117.0	84.0	7463.0	...	Andorra	10505.403482
4	NaN	NaN	NaN	Angola	2021-01-02	-11.20270	17.873900	17568.0	405.0	11146.0	...	Angola	53.452981

5 rows × 21 columns

Selecting Important Features

In [114]: country=Df_covid[['Last_Update', 'Country_Region', 'Province_State', 'Lat', 'Long_', 'Confirmed', 'Deaths', 'Recovered', 'Active', 'Date']]

NOW PERFORMING EXPLORATORY DATA ANALYSIS (EDA)

In [115]: country.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1182267 entries, 0 to 1182266
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Last_Update      7617 non-null   object 
 1   Country_Region   1174650 non-null object 
 2   Province_State   1120994 non-null object 
 3   Lat               1151132 non-null float64
 4   Long_             1151132 non-null float64
 5   Confirmed         1182248 non-null float64
 6   Deaths            1181835 non-null float64
 7   Recovered         1181881 non-null float64
 8   Active             1174118 non-null float64
 9   Date              1174650 non-null datetime64[ns]
dtypes: datetime64[ns](1), float64(6), object(3)
memory usage: 90.2+ MB

```

In [116]: country.isna().sum()

```

Out[116]: Last_Update      1174650
Country_Region     7617
Province_State     61273
Lat                 31135
Long_               31135
Confirmed           19
Deaths              432
Recovered            386
Active               8149
Date                7617
dtype: int64

```

In [117]: full_latest = Df_covid[Df_covid['Date'] == max(Df_covid['Date'])].reset_index()
world = full_latest.groupby("Country_Region")['Confirmed', 'Active', 'Recovered', 'Deaths'].sum().reset_index()
world.head()

<ipython-input-117-303db51cad88>:2: FutureWarning:

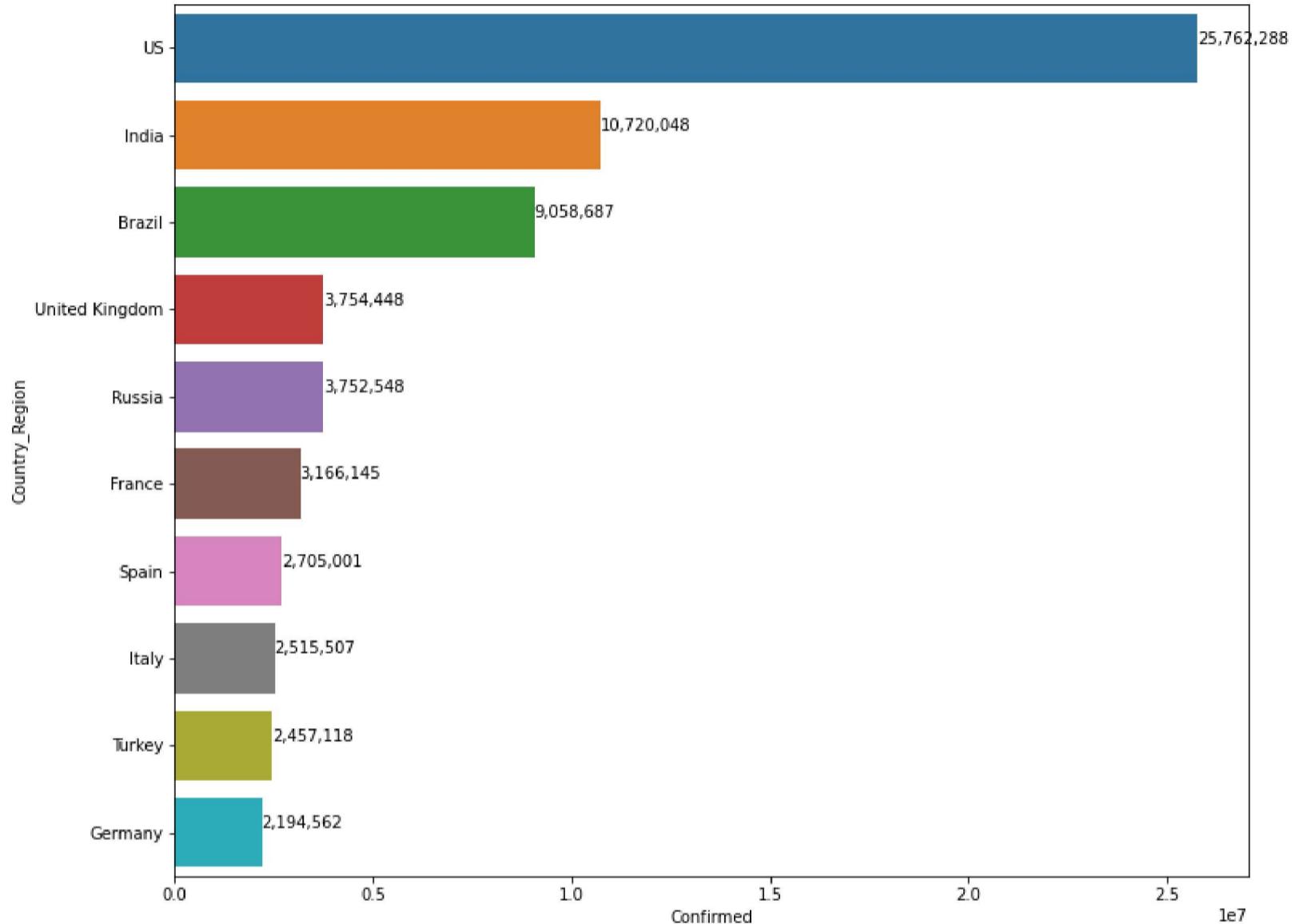
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

Out[117]:

	Country_Region	Confirmed	Active	Recovered	Deaths
0	Afghanistan	54891.0	4911.0	47583.0	2397.0
1	Albania	75454.0	28198.0	45906.0	1350.0
2	Algeria	106610.0	30984.0	72745.0	2881.0
3	Andorra	9779.0	737.0	8942.0	100.0
4	Angola	19672.0	1516.0	17692.0	464.0

```
In [118]: ### Find top 20 countries with maximum number of confirmed cases
top_10 = world.sort_values(by=['Confirmed'], ascending=False).head(10)
### Generate a Barplot
plt.figure(figsize=(12,10))
plot = sns.barplot(top_10['Confirmed'], top_10['Country_Region'])
for i,(value,name) in enumerate(zip(top_10['Confirmed'],top_10['Country_Region'])):
    plot.text(value,i-0.05,f'{value:,.0f}',size=10)
plt.show()
```

C:\Users\ANSH VOHRA\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

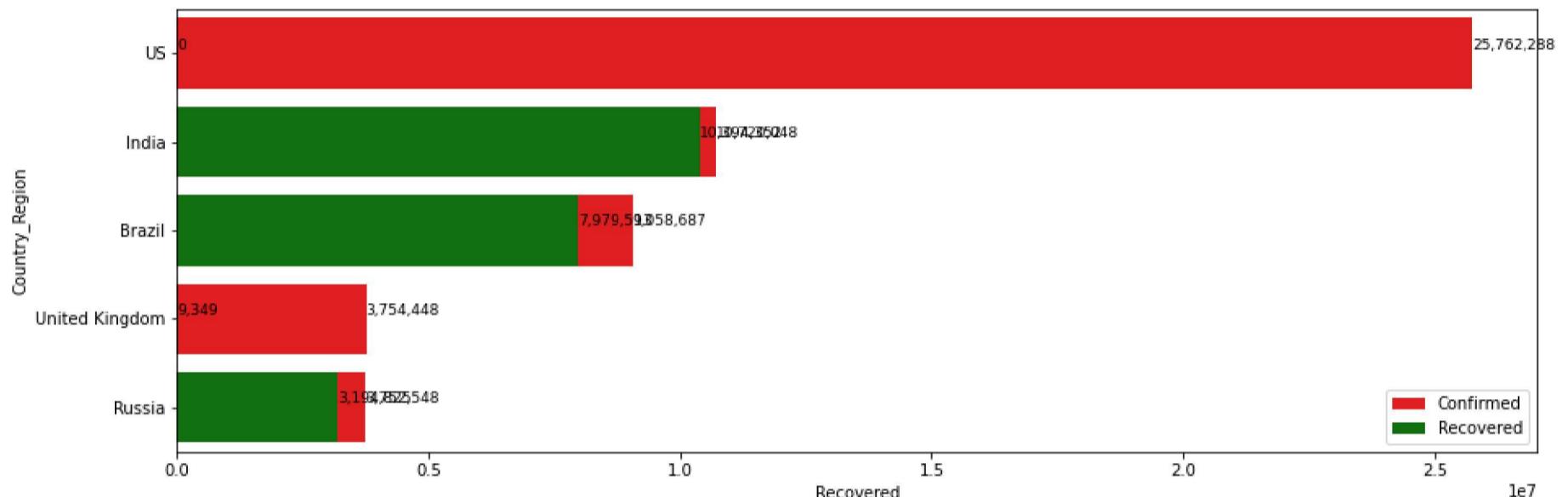


```
In [119]: top_5 = world.sort_values(by=['Confirmed'], ascending=False).head()

### Generate a Barplot
plt.figure(figsize=(15,5))
confirmed = sns.barplot(top_5['Confirmed'], top_5['Country_Region'], color = 'red', label='Confirmed')
recovered = sns.barplot(top_5['Recovered'], top_5['Country_Region'], color = 'green', label='Recovered')

### Adding Texts for barplots
for i,(value,name) in enumerate(zip(top_5['Confirmed'],top_5['Country_Region'])):
    confirmed.text(value,i-0.05,f'{value:,.0f}',size=9)
for i,(value,name) in enumerate(zip(top_5['Recovered'],top_5['Country_Region'])):
    recovered.text(value,i-0.05,f'{value:,.0f}',size=9)
plt.legend(loc=4)
plt.show()
```

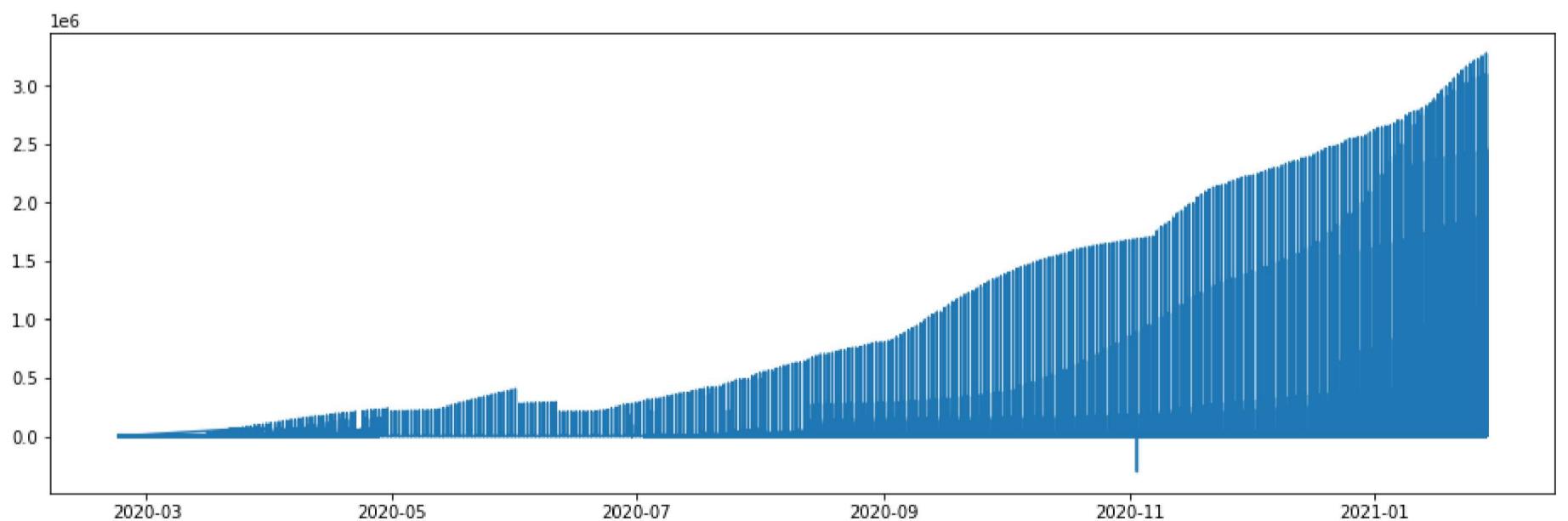
C:\Users\ANSH VOHRA\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
 Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`
 `, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
 C:\Users\ANSH VOHRA\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
 Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`
 `, and passing other arguments without an explicit keyword will result in an error or misinterpretation.



Covid Confirmed Cases

```
In [120]: #Plotting
import matplotlib.pyplot as plt
plt.figure(figsize=(16,5))
plt.plot(Df_covid.Date , Df_covid.Confirmed)
```

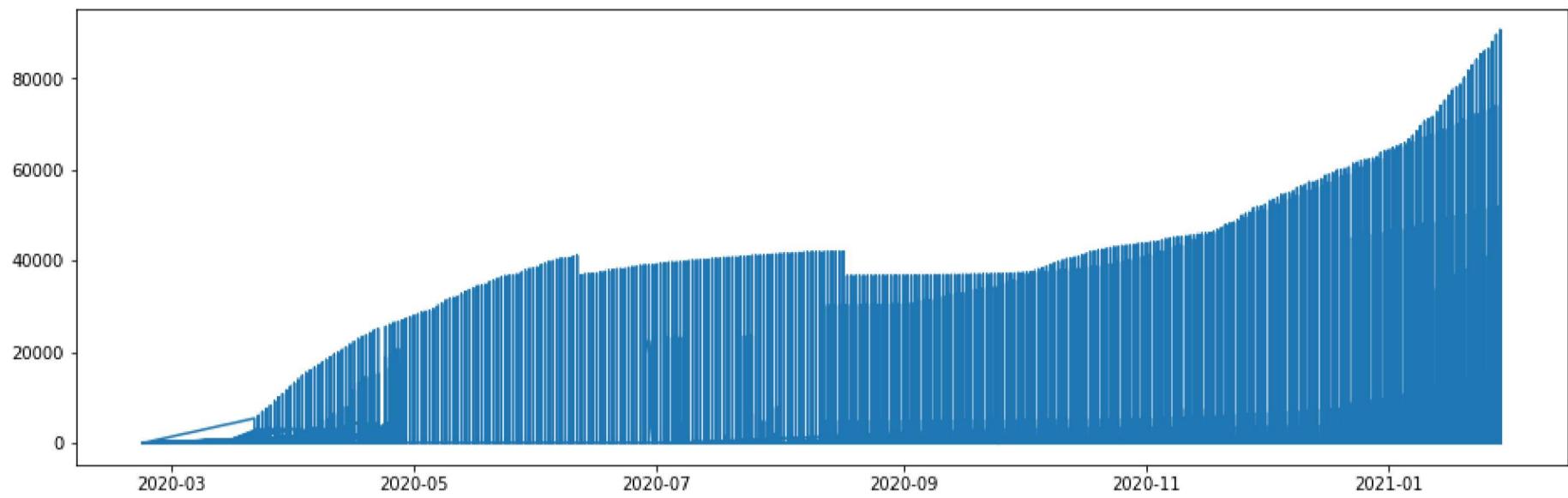
Out[120]: [`<matplotlib.lines.Line2D at 0x18d8ec1ee20>`]



Covid Deaths

```
In [121]: plt.figure(figsize=(16,5))
plt.plot(Df_covid.Date , Df_covid.Deaths)
```

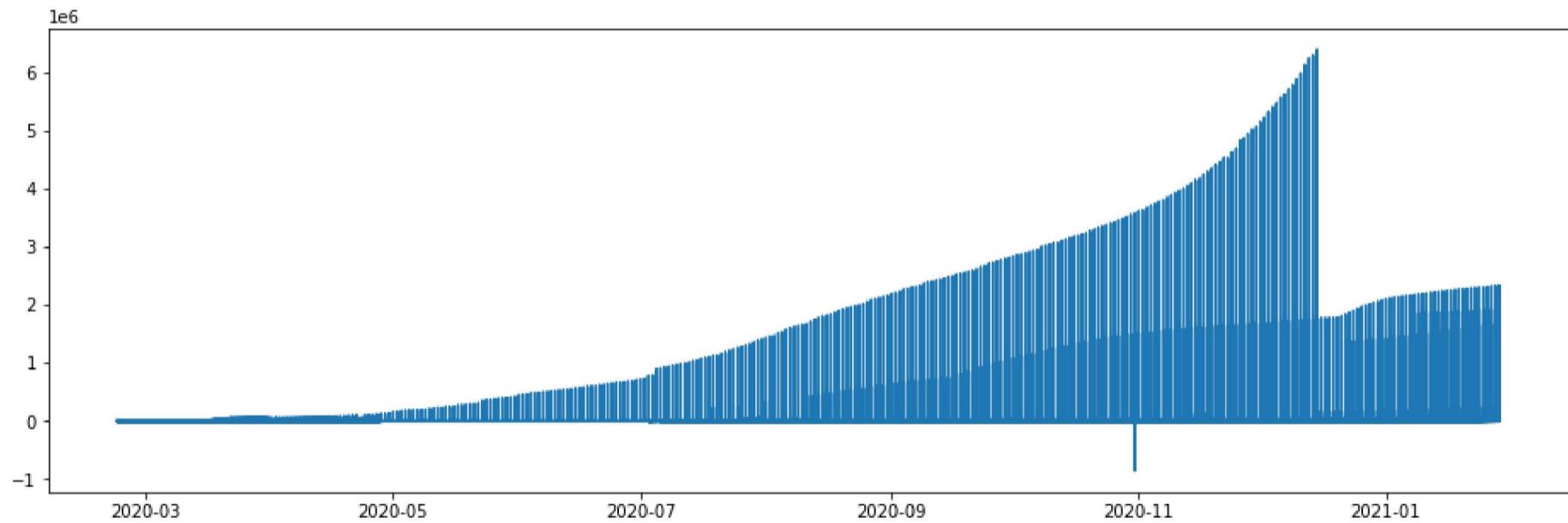
```
Out[121]: [<matplotlib.lines.Line2D at 0x18d927d3070>]
```



Covid Recovered Cases

```
In [122]: plt.figure(figsize=(16,5))
plt.plot(Df_covid.Date , Df_covid.Recovered)
```

```
Out[122]: [<matplotlib.lines.Line2D at 0x18dc96e37c0>]
```



```
In [123]: #Filtering Now Analysis of Covid cases only in India
Df_India = country[country.Country_Region=='India']
```

In [124]: Df_India

Out[124]:

	Last Update	Country_Region	Province_State	Lat	Long_	Confirmed	Deaths	Recovered	Active	Date
247	NaN	India	Andaman and Nicobar Islands	11.225999	92.968178	4945.0	62.0	4826.0	57.0	2021-01-02
248	NaN	India	Andhra Pradesh	15.912900	79.740000	882286.0	7108.0	871916.0	3262.0	2021-01-02
249	NaN	India	Arunachal Pradesh	27.768456	96.384277	16719.0	56.0	16564.0	99.0	2021-01-02
250	NaN	India	Assam	26.357149	92.830441	216211.0	1045.0	211910.0	3256.0	2021-01-02
251	NaN	India	Bihar	25.679658	85.604840	251743.0	1397.0	245476.0	4870.0	2021-01-02
...
1178570	NaN	India	Tripura	23.746783	91.743565	33264.0	385.0	32751.0	128.0	2021-01-01
1178571	NaN	India	Unknown	NaN	NaN	0.0	0.0	0.0	0.0	2021-01-01
1178572	NaN	India	Uttar Pradesh	26.925425	80.560982	584966.0	8352.0	562459.0	14155.0	2021-01-01
1178573	NaN	India	Uttarakhand	30.156447	79.197608	90616.0	1504.0	84149.0	4963.0	2021-01-01
1178574	NaN	India	West Bengal	23.814082	87.979803	550893.0	9683.0	528829.0	12381.0	2021-01-01

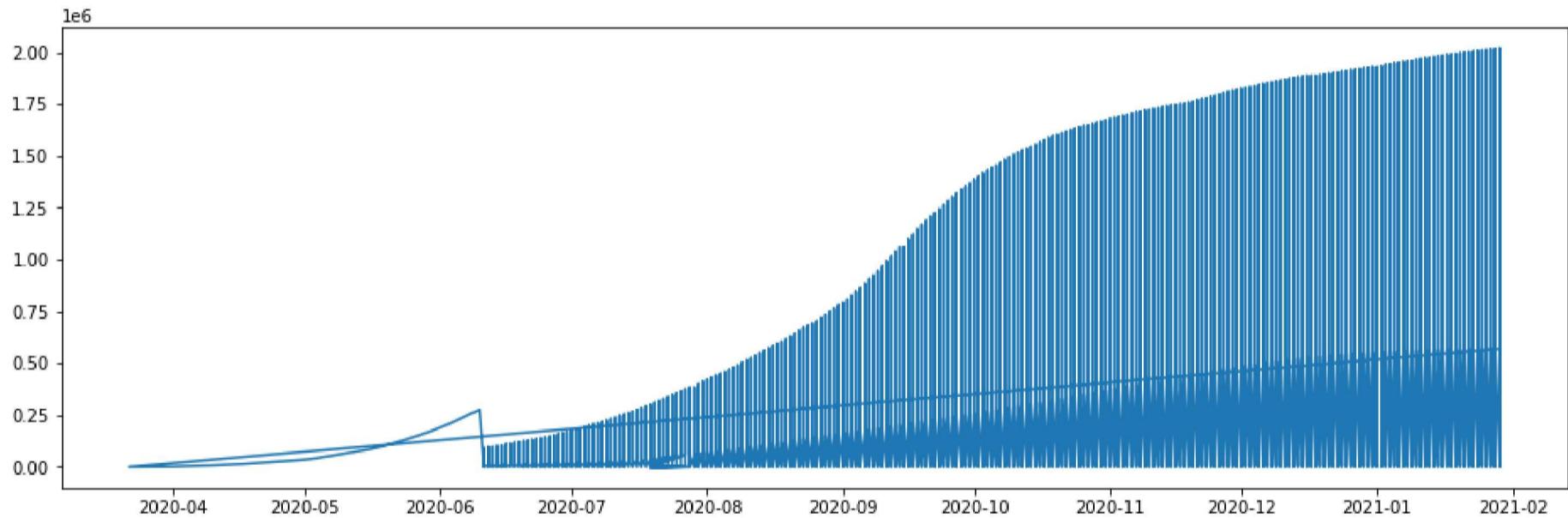
8653 rows × 10 columns

NOW VISUALISATION OF INDIA's COVID SITUATION

Confirmed Cases in India

In [125]: *#Plotting Of Confirmed Cases in India*

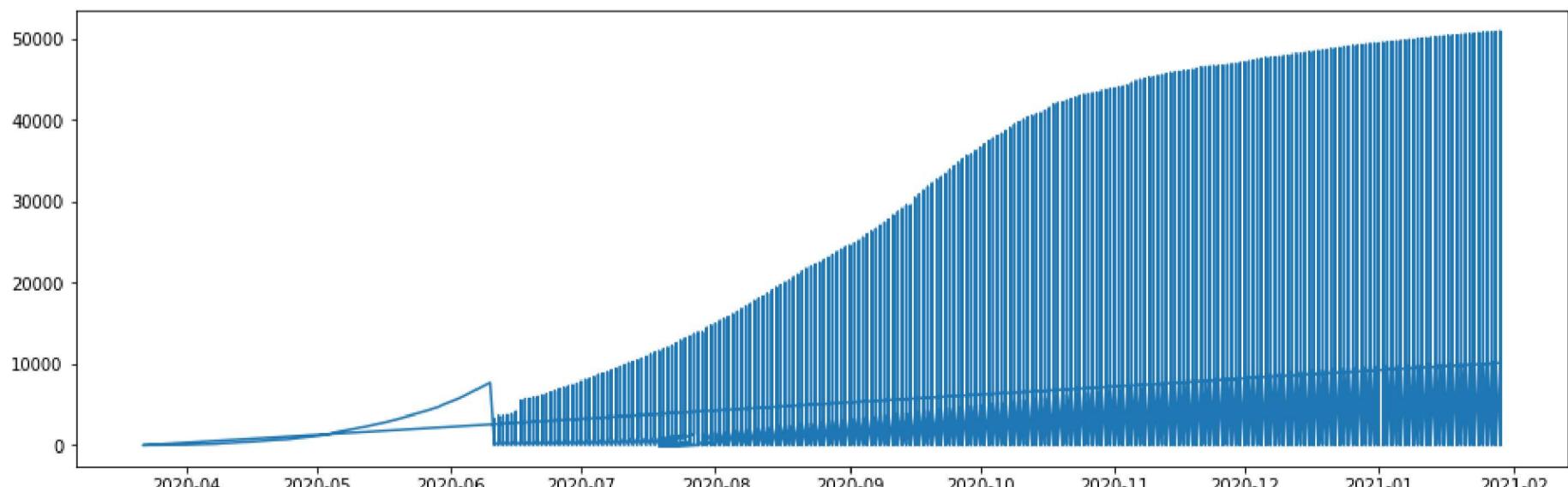
```
plt.figure(figsize=(16,5))
plt.plot(Df_India.Date , Df_India.Confirmed)
```

Out[125]: [`<matplotlib.lines.Line2D at 0x18e11f076a0>`]

Covid Deaths In India

In [126]: `plt.figure(figsize=(16,5))`

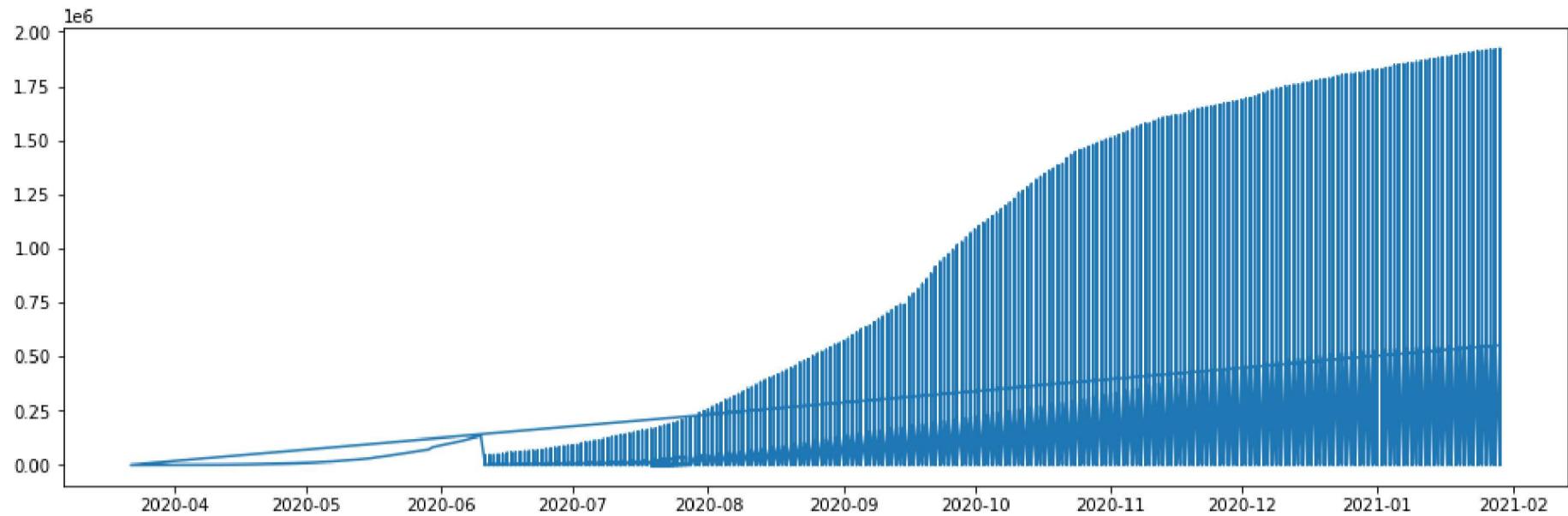
```
plt.plot(Df_India.Date , Df_India.Deaths)
```

Out[126]: [`<matplotlib.lines.Line2D at 0x18e4224dbb0>`]

Recovery From Covid In India

```
In [127]: plt.figure(figsize=(16,5))
plt.plot(Df_India.Date , Df_India.Recovered)
```

```
Out[127]: [matplotlib.lines.Line2D at 0x18d928d4100]
```



```
In [132]: state_wise=india.groupby('Province_State').sum().diff()
state_wise=india.groupby('Province_State').sum()
state_wise['state']=state_wise.index
state_wise=state_wise.iloc[1:,:]
state_wise.head()
```

```
Out[132]:
```

Province_State	Lat	Long_	Confirmed	Deaths	Recovered	Active	state
Andhra Pradesh	3707.705700	18579.420000	129645726.0	1075241.0	120405266.0	7825635.0	Andhra Pradesh
Arunachal Pradesh	6470.050248	22457.536541	2165506.0	6110.0	1925913.0	233483.0	Arunachal Pradesh
Assam	6141.215717	21629.492753	32690323.0	138816.0	29719812.0	2831695.0	Assam
Bihar	5983.360314	19945.927720	36630211.0	192979.0	34194901.0	2242331.0	Bihar
Chandigarh	7160.984487	17887.008774	2403209.0	36846.0	2165670.0	200693.0	Chandigarh

```
In [133]: state_wise.sort_values('Confirmed').tail(1)
```

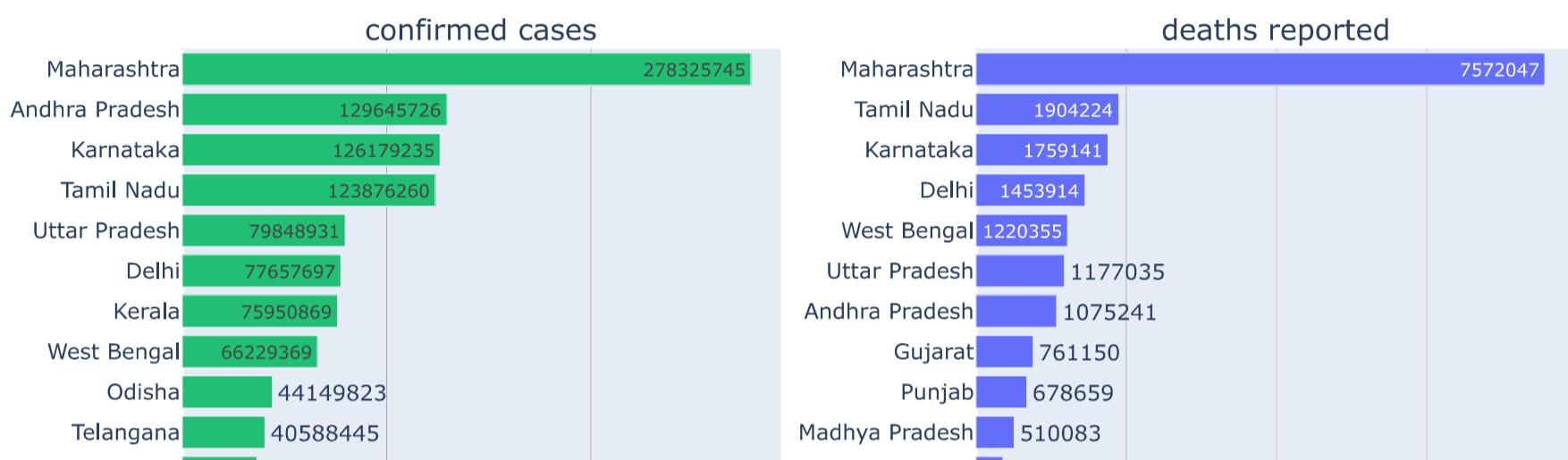
```
Out[133]:
```

Province_State	Lat	Long_	Confirmed	Deaths	Recovered	Active	state
Maharashtra	4531.793847	17733.215493	278325745.0	7572047.0	240801606.0	29952092.0	Maharashtra

In [139]:

```
#color palette
cnf="#393e46"
dth='ff2e63'
act="#21bf73"
rec="#fe9801'

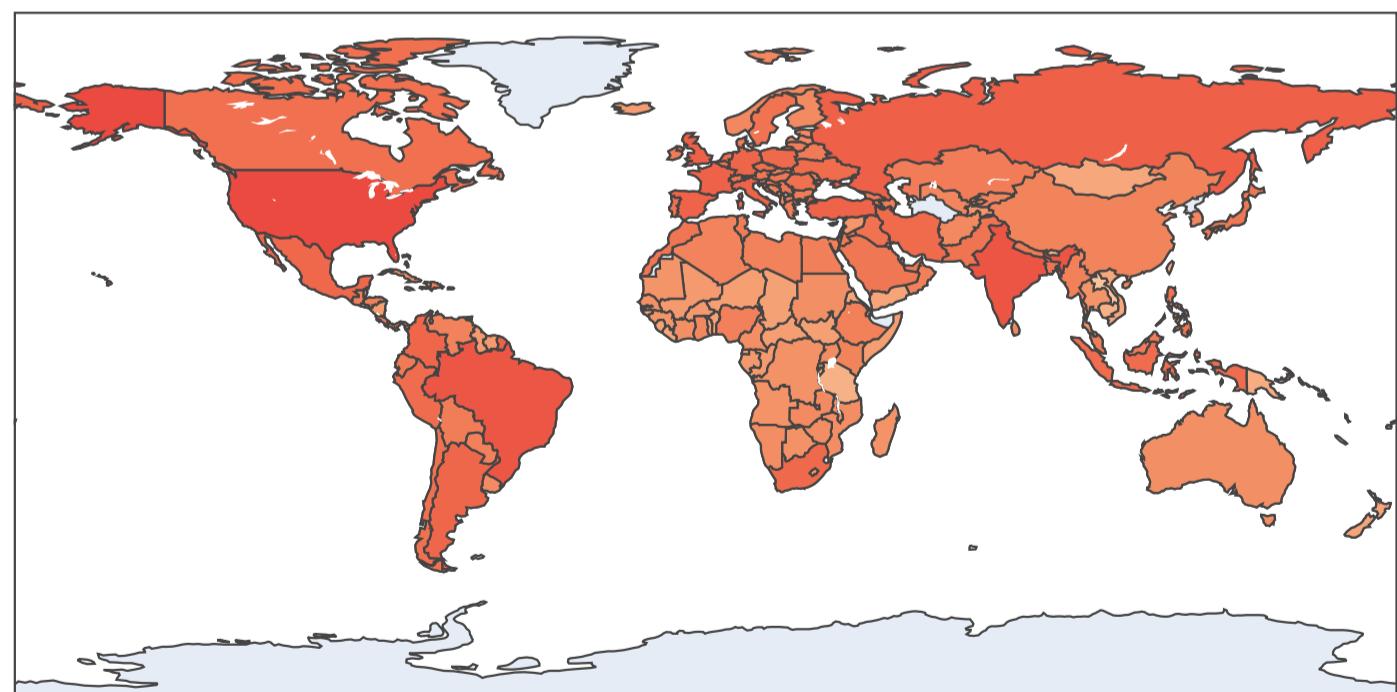
top=15
fig_c=px.bar(state_wise.sort_values('Confirmed').tail(top),x='Confirmed',y='state',text='Confirmed',
             orientation='h',color_discrete_sequence=[act])
fig_d=px.bar(state_wise.sort_values('Deaths').tail(top),x='Deaths',y='state',text='Deaths',
             orientation='h')
fig=make_subplots(rows=5,cols=2,shared_xaxes=False,horizontal_spacing=0.14,vertical_spacing=.1,
                  subplot_titles=('confirmed cases','deaths reported'))
fig.add_trace(fig_c['data'][0],row=1,col=1)
fig.add_trace(fig_d['data'][0],row=1,col=2)
fig.update_layout(height=3000)
fig.show()
```



In [140]:

```
import plotly.express as px
fig = px.choropleth(world, locations="Country_Region",
                     locationmode='country names', color=np.log(world["Confirmed"]),
                     hover_name="Country_Region", hover_data=['Confirmed'],
                     color_continuous_scale="peach",
                     title='Countries with Confirmed Cases')
fig.update(layout_coloraxis_showscale=False)
fig.show()
```

Countries with Confirmed Cases



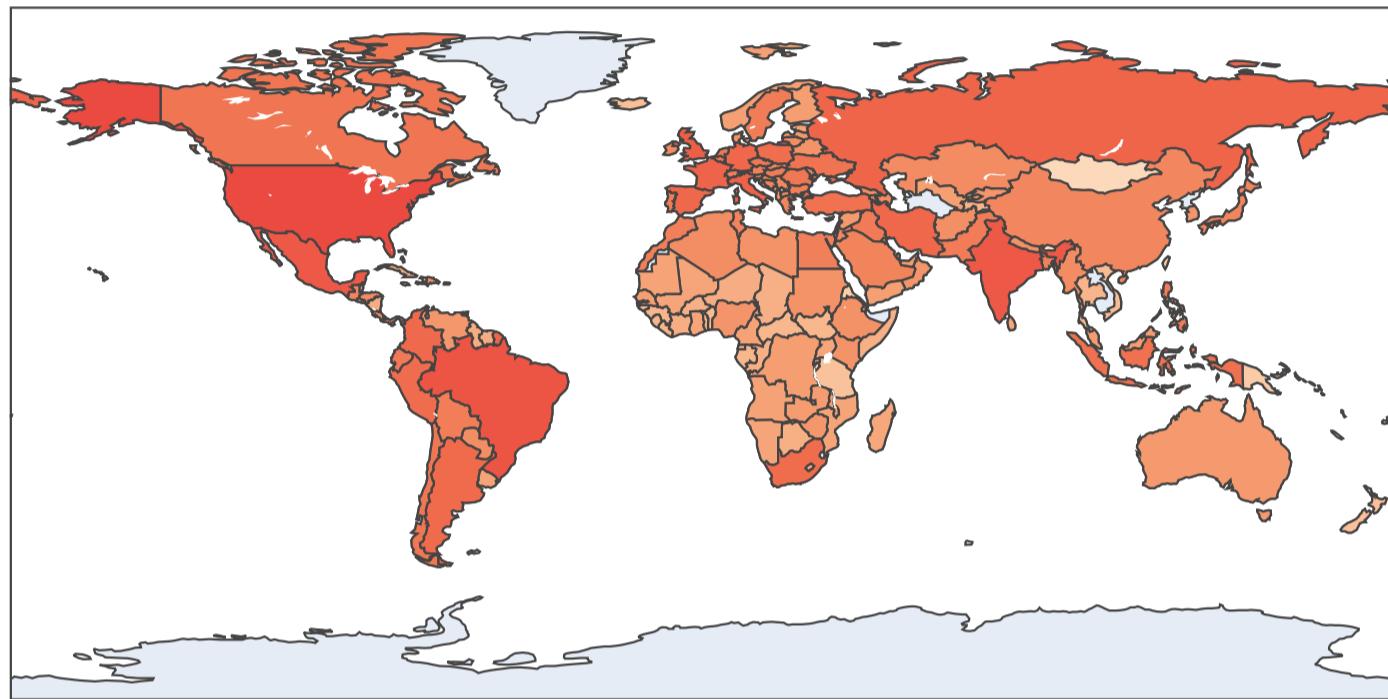
In [141]: # Deaths

```
fig = px.choropleth(world,
                     locations="Country_Region", locationmode='country names',
                     color=np.log(world["Deaths"]), hover_name="Country_Region",
                     color_continuous_scale="Peach", hover_data=['Deaths'],
                     title='Countries with Deaths Reported')
fig.update(layout_coloraxis_showscale=False)
fig.show()
```

C:\Users\ANSH VOHRA\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas\core\arraylike.py:358: RuntimeWarning:

divide by zero encountered in log

Countries with Deaths Reported



```
In [142]: temp = Df_covid.groupby('Date')[['Confirmed', 'Deaths', 'Recovered', 'Active']].sum().reset_index()
temp = temp[temp['Date'] == max(temp['Date'])].reset_index(drop=True)
temp['Global Mortality'] = temp['Deaths'] / temp['Confirmed']
temp['Deaths per 100 Confirmed Cases'] = temp['Global Mortality'] * 100
temp.style.background_gradient(cmap='Pastel1')
```

<ipython-input-142-23cda484d291>:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

Out[142]:

	Date	Confirmed	Deaths	Recovered	Active	Global Mortality	Deaths per 100 Confirmed Cases
0	2021-01-29 00:00:00	101452950.000000	2191069.000000	56043419.000000	43217953.000000	0.021597	2.159690

```
In [27]: full_latest = Df_covid[Df_covid['Date'] == max(Df_covid['Date'])].reset_index()
full_latest_grouped = full_latest.groupby('Country_Region')[['Confirmed', 'Deaths', 'Recovered', 'Active']].sum().reset_index()
temp_f = full_latest_grouped.sort_values(by='Confirmed', ascending=False)
temp_f = temp_f[['Country_Region', 'Confirmed', 'Active', 'Deaths', 'Recovered']]
temp_f = temp_f.reset_index(drop=True)

temp_f.style.background_gradient(cmap="Blues", subset=['Confirmed', 'Active'])\
    .background_gradient(cmap="Greens", subset=['Recovered'])\
    .background_gradient(cmap="Reds", subset=['Deaths'])
```

<ipython-input-27-e3805be97499>:2: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

Out[27]:

	Country_Region	Confirmed	Active	Deaths	Recovered
0	US	25762288.000000	25329226.000000	433062.000000	0.000000
1	India	10720048.000000	171686.000000	154010.000000	10394352.000000
2	Brazil	9058687.000000	857627.000000	221547.000000	7979513.000000
3	United Kingdom	3754448.000000	3641775.000000	103324.000000	9349.000000
4	Russia	3752548.000000	487190.000000	70533.000000	3194825.000000
5	France	3166145.000000	2864263.000000	74601.000000	226772.000000
6	Spain	2705001.000000	2496819.000000	57806.000000	150376.000000
7	Italy	2515507.000000	474617.000000	87381.000000	1953509.000000
8	Turkey	2457118.000000	91297.000000	25605.000000	2340216.000000
9	Germany	2194562.000000	223178.000000	55883.000000	1915501.000000
10	Colombia	2067575.000000	120278.000000	52913.000000	1894384.000000
11	Argentina	1905524.000000	164777.000000	47601.000000	1693146.000000
12	Mexico	1825519.000000	294301.000000	155145.000000	1376073.000000
13	Poland	1496665.000000	209330.000000	36443.000000	1250892.000000
14	South Africa	1437798.000000	122496.000000	43105.000000	1272197.000000
15	Iran	1398841.000000	151130.000000	57736.000000	1189975.000000
16	Ukraine	1247674.000000	197741.000000	23469.000000	1026464.000000
17	Peru	1113970.000000	45578.000000	40272.000000	1028120.000000
18	Indonesia	1037993.000000	166540.000000	29331.000000	842122.000000
19	Netherlands	979506.000000	953067.000000	13922.000000	12517.000000
20	Czechia	964660.000000	97316.000000	15944.000000	851400.000000
21	Canada	770414.000000	55191.000000	19658.000000	695565.000000
22	Romania	721513.000000	37407.000000	18105.000000	666001.000000
23	Chile	714143.000000	25634.000000	18174.000000	670336.000000
24	Belgium	702437.000000	681455.000000	20982.000000	0.000000
25	Portugal	685383.000000	180076.000000	11608.000000	493699.000000
26	Israel	628895.000000	72737.000000	4669.000000	551489.000000
27	Iraq	617202.000000	15918.000000	13024.000000	588260.000000
28	Sweden	564557.000000	553037.000000	11520.000000	0.000000
29	Pakistan	541031.000000	32726.000000	11560.000000	496745.000000
30	Bangladesh	533953.000000	47320.000000	8087.000000	478546.000000
31	Philippines	519575.000000	33427.000000	10552.000000	475596.000000
32	Switzerland	519404.000000	192496.000000	9308.000000	317600.000000
33	Morocco	469139.000000	13839.000000	8224.000000	447076.000000
34	Austria	410230.000000	14836.000000	7607.000000	387787.000000
35	Serbia	390637.000000	386672.000000	3965.000000	0.000000
36	Japan	380644.000000	60863.000000	5503.000000	314278.000000
37	Saudi Arabia	367276.000000	2157.000000	6366.000000	358753.000000
38	Hungary	363450.000000	102156.000000	12291.000000	249003.000000
39	Jordan	324169.000000	8012.000000	4269.000000	311888.000000
40	Panama	316808.000000	41975.000000	5196.000000	269637.000000
41	Lebanon	293157.000000	118039.000000	2621.000000	172497.000000
42	United Arab Emirates	293052.000000	25209.000000	819.000000	267024.000000
43	Nepal	270588.000000	3203.000000	2020.000000	265365.000000
44	Georgia	256287.000000	6701.000000	3127.000000	246459.000000

	Country_Region	Confirmed	Active	Deaths	Recovered
45	Ecuador	246000.000000	27163.000000	14766.000000	204071.000000
46	Slovakia	243427.000000	33769.000000	4411.000000	205247.000000
47	Belarus	242851.000000	11964.000000	1688.000000	229199.000000
48	Kazakhstan	231716.000000	27571.000000	3040.000000	201105.000000
49	Croatia	230978.000000	3173.000000	4943.000000	222862.000000
50	Azerbaijan	229793.000000	4526.000000	3113.000000	222154.000000
51	Bulgaria	217574.000000	25676.000000	8973.000000	182925.000000
52	Bolivia	210726.000000	44778.000000	10226.000000	155722.000000
53	Dominican Republic	208610.000000	54304.000000	2603.000000	151703.000000
54	Tunisia	204351.000000	45407.000000	6508.000000	152436.000000
55	Malaysia	198208.000000	43192.000000	717.000000	154299.000000
56	Denmark	197892.000000	10292.000000	2072.000000	185528.000000
57	Ireland	193645.000000	167114.000000	3167.000000	23364.000000
58	Costa Rica	192637.000000	38656.000000	2599.000000	151382.000000
59	Lithuania	180170.000000	52360.000000	2743.000000	125067.000000
60	Armenia	166669.000000	6957.000000	3067.000000	156645.000000
61	Egypt	164282.000000	26673.000000	9169.000000	128440.000000
62	Kuwait	163450.000000	6105.000000	958.000000	156387.000000
63	Slovenia	163235.000000	17653.000000	3448.000000	142134.000000
64	Moldova	158309.000000	6464.000000	3413.000000	148432.000000
65	Guatemala	157595.000000	9732.000000	5543.000000	142320.000000
66	West Bank and Gaza	157593.000000	8161.000000	1812.000000	147620.000000
67	Greece	154796.000000	55290.000000	5742.000000	93764.000000
68	Qatar	150280.000000	4618.000000	248.000000	145414.000000
69	Honduras	144992.000000	80708.000000	3549.000000	60735.000000
70	Burma	139152.000000	12493.000000	3103.000000	123556.000000
71	Ethiopia	135594.000000	11649.000000	2085.000000	121860.000000
72	Oman	133728.000000	5629.000000	1527.000000	126572.000000
73	Paraguay	130917.000000	22127.000000	2681.000000	106109.000000
74	Nigeria	127024.000000	24624.000000	1547.000000	100853.000000
75	Venezuela	125364.000000	6746.000000	1171.000000	117447.000000
76	Bosnia and Herzegovina	121194.000000	22463.000000	4659.000000	94072.000000
77	Libya	116779.000000	18654.000000	1832.000000	96293.000000
78	Algeria	106610.000000	30984.000000	2881.000000	72745.000000
79	Bahrain	101503.000000	3467.000000	372.000000	97664.000000
80	Kenya	100422.000000	14912.000000	1753.000000	83757.000000
81	China	99746.000000	2710.000000	4813.000000	92223.000000
82	North Macedonia	91891.000000	8667.000000	2831.000000	80393.000000
83	Kyrgyzstan	84303.000000	2414.000000	1408.000000	80481.000000
84	Uzbekistan	78556.000000	888.000000	621.000000	77047.000000
85	Korea, South	77395.000000	9493.000000	1399.000000	66503.000000
86	Albania	75454.000000	28198.000000	1350.000000	45906.000000
87	Latvia	63992.000000	10782.000000	1148.000000	52062.000000
88	Ghana	63883.000000	3940.000000	390.000000	59553.000000
89	Norway	62276.000000	43721.000000	557.000000	17998.000000
90	Sri Lanka	61586.000000	6854.000000	297.000000	54435.000000
91	Montenegro	60288.000000	7544.000000	790.000000	51954.000000
92	Singapore	59425.000000	248.000000	29.000000	59148.000000
93	Kosovo	58988.000000	5989.000000	1479.000000	51520.000000
94	Afghanistan	54891.000000	4911.000000	2397.000000	47583.000000
95	El Salvador	53989.000000	4747.000000	1599.000000	47643.000000
96	Zambia	50319.000000	6843.000000	705.000000	42771.000000
97	Luxembourg	50228.000000	2148.000000	574.000000	47506.000000
98	Finland	44039.000000	12375.000000	664.000000	31000.000000
99	Estonia	42656.000000	9852.000000	406.000000	32398.000000
100	Uruguay	39887.000000	7166.000000	415.000000	32306.000000
101	Uganda	39424.000000	24992.000000	318.000000	14114.000000

	Country_Region	Confirmed	Active	Deaths	Recovered
102	Mozambique	35833.000000	12736.000000	347.000000	22750.000000
103	Namibia	33459.000000	1718.000000	332.000000	31409.000000
104	Zimbabwe	32646.000000	7067.000000	1160.000000	24419.000000
105	Cyprus	30538.000000	28284.000000	197.000000	2057.000000
106	Cameroon	29617.000000	1110.000000	462.000000	28045.000000
107	Sudan	29066.000000	6113.000000	1798.000000	21155.000000
108	Australia	28799.000000	1856.000000	909.000000	26034.000000
109	Cote d'Ivoire	27694.000000	2001.000000	151.000000	25542.000000
110	Senegal	25711.000000	3702.000000	614.000000	21395.000000
111	Cuba	24105.000000	4847.000000	208.000000	19050.000000
112	Congo (Kinshasa)	22322.000000	6660.000000	665.000000	14997.000000
113	Malawi	22304.000000	14128.000000	628.000000	7548.000000
114	Botswana	21293.000000	3961.000000	134.000000	17198.000000
115	Angola	19672.000000	1516.000000	464.000000	17692.000000
116	Madagascar	18743.000000	534.000000	279.000000	17930.000000
117	Malta	17400.000000	2687.000000	261.000000	14452.000000
118	Mauritania	16543.000000	562.000000	421.000000	15560.000000
119	Thailand	16221.000000	4858.000000	76.000000	11287.000000
120	Maldives	15496.000000	1417.000000	51.000000	14028.000000
121	Jamaica	15435.000000	3130.000000	344.000000	11961.000000
122	Eswatini	15283.000000	4912.000000	538.000000	9833.000000
123	Rwanda	14529.000000	5109.000000	186.000000	9234.000000
124	Guinea	14435.000000	407.000000	82.000000	13946.000000
125	Syria	13885.000000	5650.000000	906.000000	7329.000000
126	Cabo Verde	13784.000000	672.000000	131.000000	12981.000000
127	Tajikistan	13308.000000	0.000000	90.000000	13218.000000
128	Belize	11845.000000	274.000000	298.000000	11273.000000
129	Haiti	11286.000000	2027.000000	243.000000	9016.000000
130	Gabon	10536.000000	346.000000	68.000000	10122.000000
131	Burkina Faso	10377.000000	1432.000000	120.000000	8825.000000
132	Andorra	9779.000000	737.000000	100.000000	8942.000000
133	Suriname	8293.000000	634.000000	154.000000	7505.000000
134	Lesotho	8278.000000	5606.000000	160.000000	2512.000000
135	Bahamas	8161.000000	1232.000000	175.000000	6754.000000
136	Mali	8056.000000	1845.000000	328.000000	5883.000000
137	Congo (Brazzaville)	7887.000000	1924.000000	117.000000	5846.000000
138	Trinidad and Tobago	7520.000000	287.000000	134.000000	7099.000000
139	Guyana	7470.000000	761.000000	175.000000	6534.000000
140	Nicaragua	6253.000000	1859.000000	169.000000	4225.000000
141	Iceland	6001.000000	47.000000	29.000000	5925.000000
142	Djibouti	5929.000000	22.000000	62.000000	5845.000000
143	Equatorial Guinea	5492.000000	138.000000	86.000000	5268.000000
144	Central African Republic	4981.000000	33.000000	63.000000	4885.000000
145	Togo	4922.000000	724.000000	76.000000	4122.000000
146	Somalia	4784.000000	988.000000	130.000000	3666.000000
147	Niger	4474.000000	695.000000	157.000000	3622.000000
148	Gambia	4019.000000	111.000000	128.000000	3780.000000
149	South Sudan	3929.000000	252.000000	64.000000	3613.000000
150	Benin	3786.000000	340.000000	48.000000	3398.000000
151	Chad	3296.000000	814.000000	118.000000	2364.000000
152	Sierra Leone	3282.000000	945.000000	77.000000	2260.000000
153	San Marino	2969.000000	220.000000	65.000000	2684.000000
154	Comoros	2598.000000	975.000000	85.000000	1538.000000
155	Guinea-Bissau	2532.000000	66.000000	45.000000	2421.000000
156	Liechtenstein	2475.000000	71.000000	52.000000	2352.000000
157	New Zealand	2305.000000	72.000000	25.000000	2208.000000
158	Yemen	2120.000000	80.000000	615.000000	1425.000000

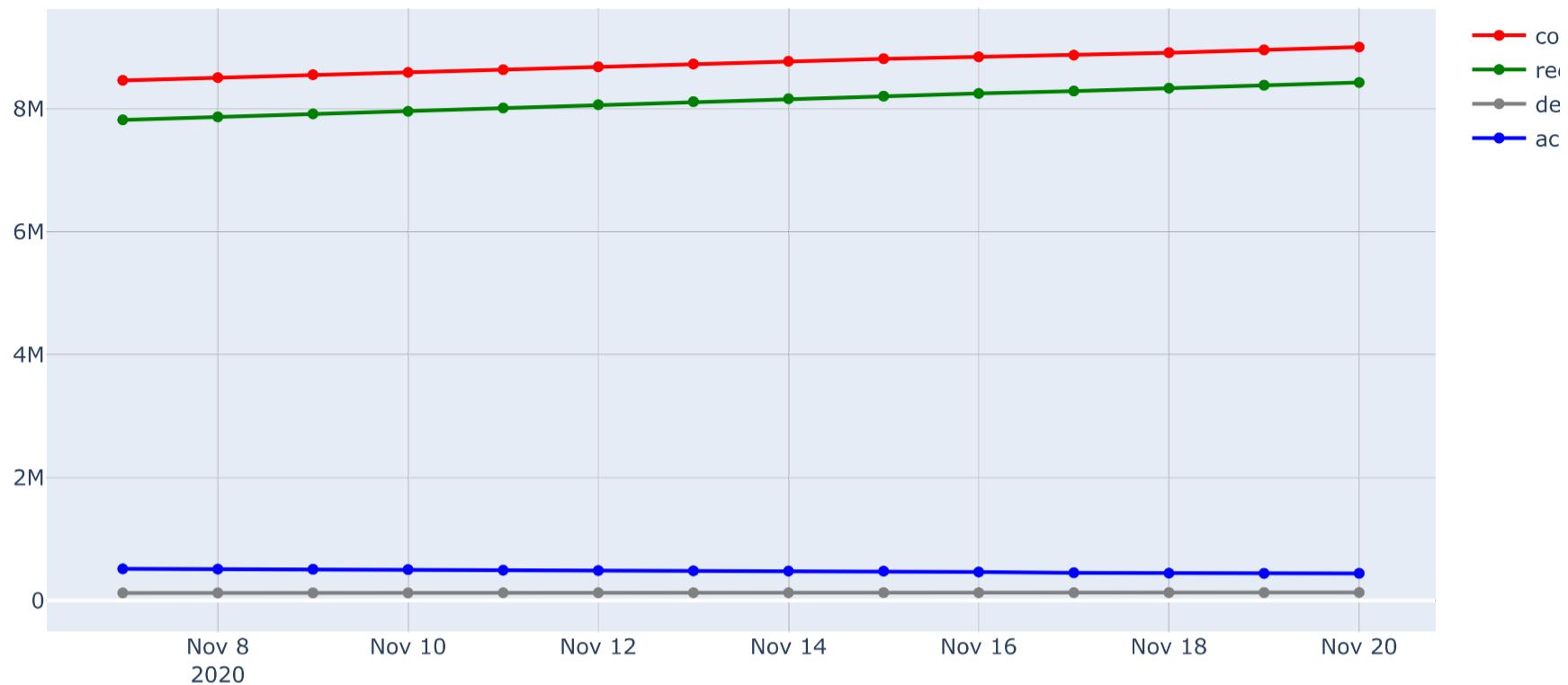
	Country_Region	Confirmed	Active	Deaths	Recovered
159	Eritrea	2085.000000	529.000000	7.000000	1549.000000
160	Liberia	1936.000000	95.000000	84.000000	1757.000000
161	Mongolia	1710.000000	405.000000	2.000000	1303.000000
162	Vietnam	1651.000000	186.000000	35.000000	1430.000000
163	Burundi	1584.000000	809.000000	2.000000	773.000000
164	Barbados	1477.000000	368.000000	11.000000	1098.000000
165	Monaco	1430.000000	221.000000	11.000000	1198.000000
166	Sao Tome and Principe	1232.000000	204.000000	17.000000	1011.000000
167	Seychelles	1149.000000	292.000000	3.000000	854.000000
168	Saint Lucia	1029.000000	493.000000	13.000000	523.000000
169	Taiwan*	895.000000	79.000000	7.000000	809.000000
170	Bhutan	857.000000	80.000000	1.000000	776.000000
171	Papua New Guinea	851.000000	30.000000	9.000000	812.000000
172	Saint Vincent and the Grenadines	827.000000	651.000000	2.000000	174.000000
173	Diamond Princess	712.000000	0.000000	13.000000	699.000000
174	Mauritius	569.000000	31.000000	10.000000	528.000000
175	Tanzania	509.000000	305.000000	21.000000	183.000000
176	Cambodia	463.000000	25.000000	0.000000	438.000000
177	Antigua and Barbuda	215.000000	35.000000	6.000000	174.000000
178	Brunei	180.000000	7.000000	3.000000	170.000000
179	Grenada	148.000000	11.000000	1.000000	136.000000
180	Dominica	117.000000	11.000000	0.000000	106.000000
181	Timor-Leste	68.000000	14.000000	0.000000	54.000000
182	Fiji	55.000000	0.000000	2.000000	53.000000
183	Laos	44.000000	3.000000	0.000000	41.000000
184	Saint Kitts and Nevis	37.000000	2.000000	0.000000	35.000000
185	Holy See	27.000000	12.000000	0.000000	15.000000
186	Solomon Islands	17.000000	7.000000	0.000000	10.000000
187	MS Zaandam	9.000000	0.000000	2.000000	7.000000
188	Marshall Islands	4.000000	0.000000	0.000000	4.000000
189	Samoa	2.000000	0.000000	0.000000	2.000000
190	Vanuatu	1.000000	0.000000	0.000000	1.000000
191	Micronesia	1.000000	0.000000	0.000000	0.000000

Covid19 During the week of Diwali

```
In [148]: Diwali_Week=Df_India[Df_India['Date']>'2020-11-06'].groupby('Date').sum().head(14).reset_index()
```

In [153]:

```
fig=go.Figure()
fig.add_trace(go.Scatter(x=diwali['Date'],y=diwali['Confirmed'],mode='lines+markers',name='confirmed',line=dict(color='red'))
fig.add_trace(go.Scatter(x=diwali['Date'],y=diwali['Recovered'],mode='lines+markers',name='recovered',line=dict(color='green'))
fig.add_trace(go.Scatter(x=diwali['Date'],y=diwali['Deaths'],mode='lines+markers',name='deaths',line=dict(color='grey'))
fig.add_trace(go.Scatter(x=diwali['Date'],y=diwali['Active'],mode='lines+markers',name='active',line=dict(color='blue'))
fig.show()
```



In []: