COVID 19 GEOSPATIAL ANALYSIS

By Koome Derrick

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly
import plotly.graph_objs as go
from plotly import tools
from plotly.offline import init_notebook_mode, plot, iplot
```

In [2]:

```
df=pd.read_csv('https://raw.githubusercontent.com/datasets/covid-19/master/
df.head()
```

Out[2]:

	Date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0
2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0
4	2020-01-26	Afghanistan	0	0	0

In [3]:

```
df['Country'].nunique()
```

Out[3]:

```
198
```

The dataset contains data from a good 198 countries

In [4]:
```
df['Country'].unique()
```

Out[4]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Andorra', 'Angola',
       'Antarctica', 'Antigua and Barbuda', 'Argentina', 'Armenia',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana',
       'Brazil', 'Brunei', 'Bulgaria', 'Burkina Faso', 'Burma', 'Burund
i',
       'Cabo Verde', 'Cambodia', 'Cameroon', 'Canada',
       'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia',
       'Comoros', 'Congo (Brazzaville)', 'Congo (Kinshasa)', 'Costa Ric
a',
       "Cote d'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czechia', 'Denmar
k',
       'Diamond Princess', 'Djibouti', 'Dominica', 'Dominican Republic',
       'Ecuador', 'Egypt', 'El Salvador', 'Equatorial Guinea', 'Eritrea',
       'Estonia', 'Eswatini', 'Ethiopia', 'Fiji', 'Finland', 'France',
       'Gabon', 'Gambia', 'Georgia', 'Germany', 'Ghana', 'Greece',
       'Grenada', 'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana',
       'Haiti', 'Holy See', 'Honduras', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
       'Korea, South', 'Kosovo', 'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvi
a',
       'Lebanon', 'Lesotho', 'Liberia', 'Libya', 'Liechtenstein',
       'Lithuania', 'Luxembourg', 'MS Zaandam', 'Madagascar', 'Malawi',
       'Malaysia', 'Maldives', 'Mali', 'Malta', 'Marshall Islands',
       'Mauritania', 'Mauritius', 'Mexico', 'Micronesia', 'Moldova',
       'Monaco', 'Mongolia', 'Montenegro', 'Morocco', 'Mozambique',
       'Namibia', 'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua',
       'Niger', 'Nigeria', 'North Macedonia', 'Norway', 'Oman',
       'Pakistan', 'Palau', 'Panama', 'Papua New Guinea', 'Paraguay',
       'Peru', 'Philippines', 'Poland', 'Portugal', 'Qatar', 'Romania',
       'Russia', 'Rwanda', 'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Vincent and the Grenadines', 'Samoa', 'San Marino',
       'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
       'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
       'Solomon Islands', 'Somalia', 'South Africa', 'South Sudan',
       'Spain', 'Sri Lanka', 'Sudan', 'Summer Olympics 2020', 'Suriname',
       'Sweden', 'Switzerland', 'Syria', 'Taiwan*', 'Tajikistan',
       'Tanzania', 'Thailand', 'Timor-Leste', 'Togo', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Turkey', 'US', 'Uganda',
       'Ukraine', 'United Arab Emirates', 'United Kingdom', 'Uruguay',
       'Uzbekistan', 'Vanuatu', 'Venezuela', 'Vietnam',
       'West Bank and Gaza', 'Winter Olympics 2022', 'Yemen', 'Zambia',
       'Zimbabwe'], dtype=object)
```

In [5]:
```
df.shape
```

Out[5]:
```
(161568, 5)
```

```
In [6]:  df.tail(10)
```

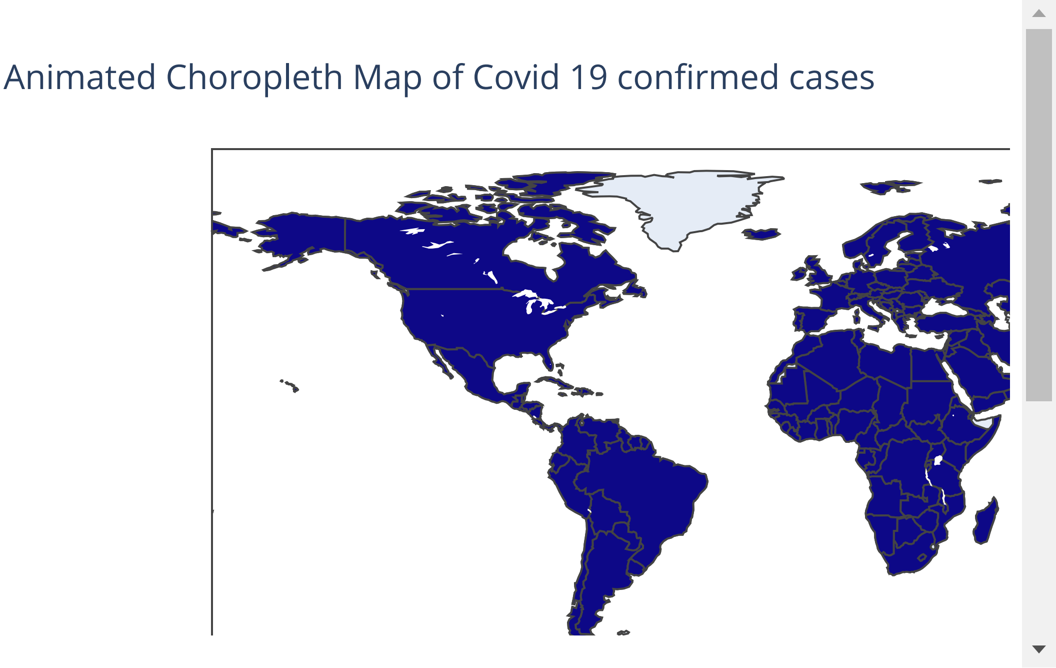
Out[6]:

	Date	Country	Confirmed	Recovered	Deaths
161558	2022-04-07	Zimbabwe	246870	0	5455
161559	2022-04-08	Zimbabwe	246925	0	5457
161560	2022-04-09	Zimbabwe	246925	0	5457
161561	2022-04-10	Zimbabwe	246958	0	5457
161562	2022-04-11	Zimbabwe	247010	0	5460
161563	2022-04-12	Zimbabwe	247094	0	5460
161564	2022-04-13	Zimbabwe	247160	0	5460
161565	2022-04-14	Zimbabwe	247208	0	5462
161566	2022-04-15	Zimbabwe	247237	0	5462
161567	2022-04-16	Zimbabwe	247237	0	5462

With this data a choropleth map can be derived

```
In [7]:  fig=px.choropleth(df,locations='Country',locationmode='country names',color
```

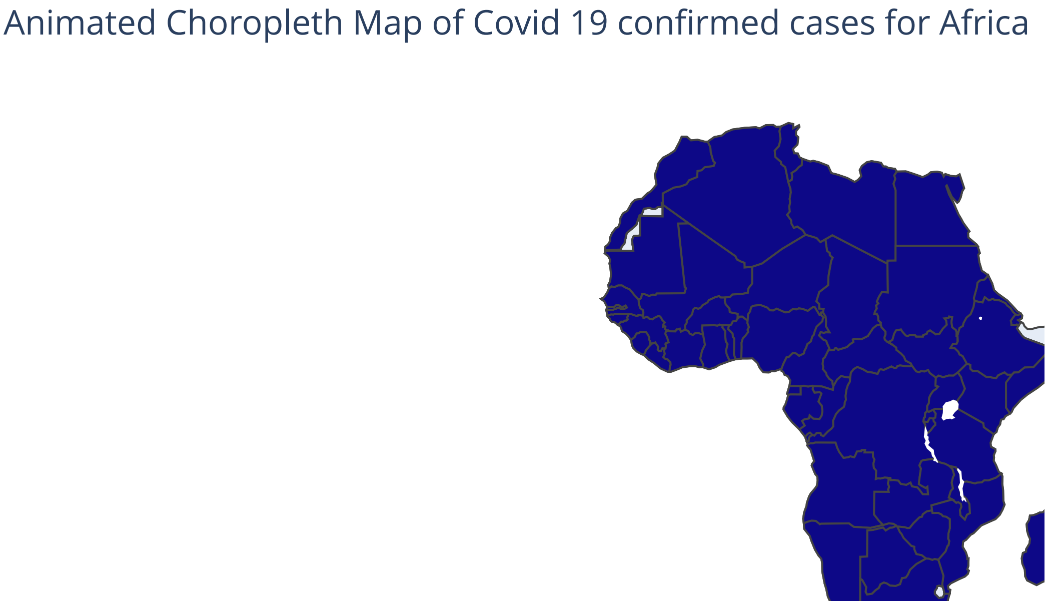
```
In [8]:  fig.update_layout(title='Animated Choropleth Map of Covid 19 confirmed case
         fig.show()
```



There is need to minimize the scope to say continent level for a better perspective

In [9]:

```
fig=px.choropleth(df,locations='Country',locationmode='country names',color
fig.update_layout(title='Animated Choropleth Map of Covid 19 confirmed case
fig.show()
```


Animated Choropleth Map of Covid 19 confirmed cases for Africa

Now to view the data in a Geographical Scatter Plot

In [10]:
```
fig2=px.scatter_geo(df,locations='Country',locationmode='country names',col
fig2.show()
```

Now interested in visualizing the recovered cases in both choropleth and scatter maps

In [11]:
```
df.columns
```

Out[11]:
```
Index(['Date', 'Country', 'Confirmed', 'Recovered', 'Deaths'], dtype='obj
ect')
```

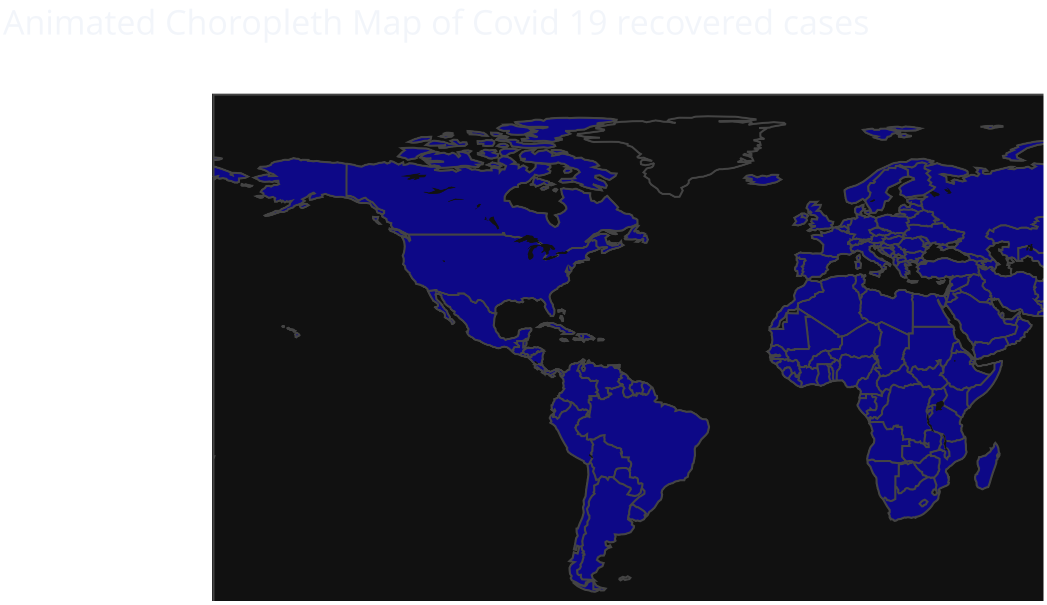
In [12]:

```
fig2=px.scatter_geo(df,locations='Country',locationmode='country names',col
fig2.show()
```

Scatter Plot showing Recovered Cases

In [13]:

```
fig=px.choropleth(df,locations='Country',locationmode='country names',color
fig.update_layout(title='Animated Choropleth Map of Covid 19 recovered case
fig.show()
```



Animated Choropleth Map of Covid 19 recovered cases

In [14]:

```
pip install geopy
```

```
Requirement already satisfied: geopy in c:\users\koome\anaconda3\lib\site
-packages (2.4.1)
Requirement already satisfied: geographiclib<3,>=1.52 in c:\users\koome\a
naconda3\lib\site-packages (from geopy) (2.0)
Note: you may need to restart the kernel to use updated packages.
```

In [15]:

```
import geopy
from geopy.geocoders import Nominatim #Nominatim is a tool used to search o
```

In [16]:

```
geolocator=Nominatim(user_agent='app')
```

In [17]:
```
location=geolocator.geocode('KICC')
location.latitude
```

Out[17]:
```
27.6812858
```

In [18]:
```
location.longitude
```

Out[18]:
```
85.3062175
```

KICC stands for Kenyatta International Conference Center, the iconic building in Nairobi.

In [19]:
```
df2=df.copy()
df2
```

Out[19]:

	Date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0
2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0
4	2020-01-26	Afghanistan	0	0	0
...	...	...	...	...	...
161563	2022-04-12	Zimbabwe	247094	0	5460
161564	2022-04-13	Zimbabwe	247160	0	5460
161565	2022-04-14	Zimbabwe	247208	0	5462
161566	2022-04-15	Zimbabwe	247237	0	5462
161567	2022-04-16	Zimbabwe	247237	0	5462

161568 rows × 5 columns

I am interested in knowing the total confirmed, revovered and deaths for each country

In [20]:

```
df2=df2.groupby(['Country'])[['Confirmed','Recovered','Deaths']].max().rese
df2
```

Out[20]:

	Country	Confirmed	Recovered	Deaths
0	Afghanistan	178387	82586	7676
1	Albania	274462	130314	3496
2	Algeria	265739	118409	6874
3	Andorra	40709	14380	155
4	Angola	99194	39582	1900
...	...	...	...	...
193	West Bank and Gaza	656617	312320	5656
194	Winter Olympics 2022	535	0	0
195	Yemen	11817	4251	2148
196	Zambia	318467	189658	3973
197	Zimbabwe	247237	82994	5462

198 rows × 4 columns

In [21]:

```
df2.head(20)
```

Out[21]:

	Country	Confirmed	Recovered	Deaths
0	Afghanistan	178387	82586	7676
1	Albania	274462	130314	3496
2	Algeria	265739	118409	6874
3	Andorra	40709	14380	155
4	Angola	99194	39582	1900
5	Antarctica	11	0	0
6	Antigua and Barbuda	7535	1239	135
7	Argentina	9060495	4615834	128344
8	Armenia	422747	220438	8621
9	Australia	5384615	24203	6779
10	Austria	4045809	644388	16407
11	Azerbaijan	792349	333694	9705
12	Bahamas	33391	12702	789
13	Bahrain	562759	267220	1473
14	Bangladesh	1952275	1141157	29124
15	Barbados	64348	4251	383
16	Belarus	974046	443417	6899
17	Belgium	3972963	31130	31165
18	Belize	57331	13543	672
19	Benin	26952	8136	163

In order to plot the above data using a heatmap or marker cluster, I need the latitude and longitudes for the various countries

In [22]:

```
lat=[]
long=[]
geolocator=Nominatim(user_agent='app')
for country in df2['Country']:
    location=geolocator.geocode(country)
    lat.append(location.latitude)
    long.append(location.longitude)
```

In [23]:

```
df2['latitude']=lat
df2['longitude']=long
df2
```

Out[23]:

	Country	Confirmed	Recovered	Deaths	latitude	longitude
0	Afghanistan	178387	82586	7676	33.768006	66.238514
1	Albania	274462	130314	3496	11.244803	-72.516097
2	Algeria	265739	118409	6874	28.000027	2.999983
3	Andorra	40709	14380	155	42.540717	1.573203
4	Angola	99194	39582	1900	-11.877577	17.569124
...	...	...	...	...	...	...
193	West Bank and Gaza	656617	312320	5656	31.904966	35.202341
194	Winter Olympics 2022	535	0	0	45.746936	126.696493
195	Yemen	11817	4251	2148	16.347124	47.891527
196	Zambia	318467	189658	3973	-14.518912	27.558988
197	Zimbabwe	247237	82994	5462	-18.455496	29.746841

198 rows × 6 columns

Now that the data is ready, I need a basemap to perform the analysis. For that I'll use Folium.

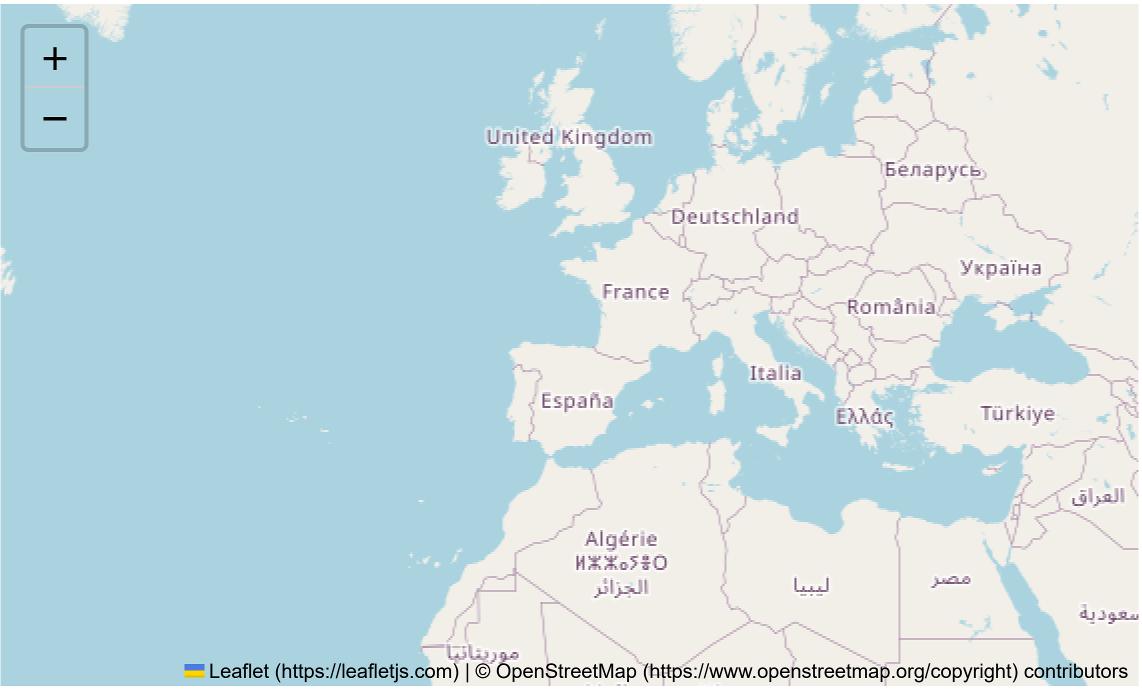
In [25]:

```
import folium
```

In [53]:

```
basemap=folium.Map(location=[27,30],zoom_start=3)
basemap
```

Out[53]:



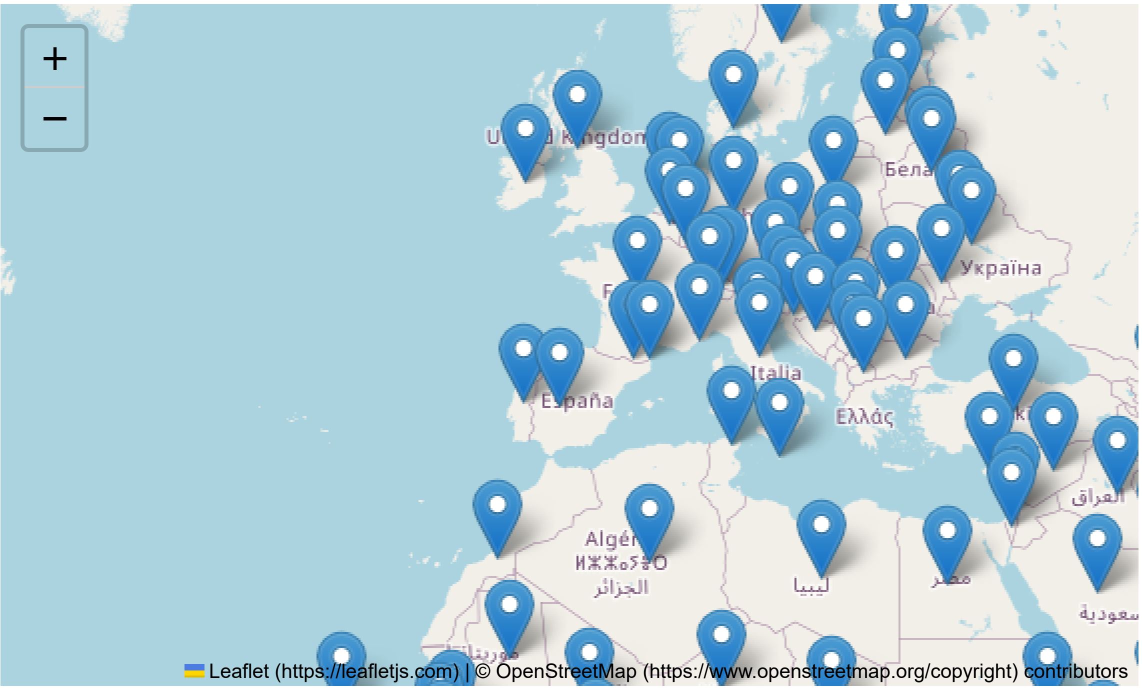
Now going to map the confirmed cases on this basemap using a marker.

In [54]:

```
for id,row in df2.iterrows():
    folium.Marker(location=[row['latitude'],row['longitude']],popup=row['Co

basemap
```

Out[54]:


+
−
Україна
Italia
España
Ελλάς
ليبيا
Leaflet (https://leafletjs.com) | © OpenStreetMap (https://www.openstreetmap.org/copyright) contributors

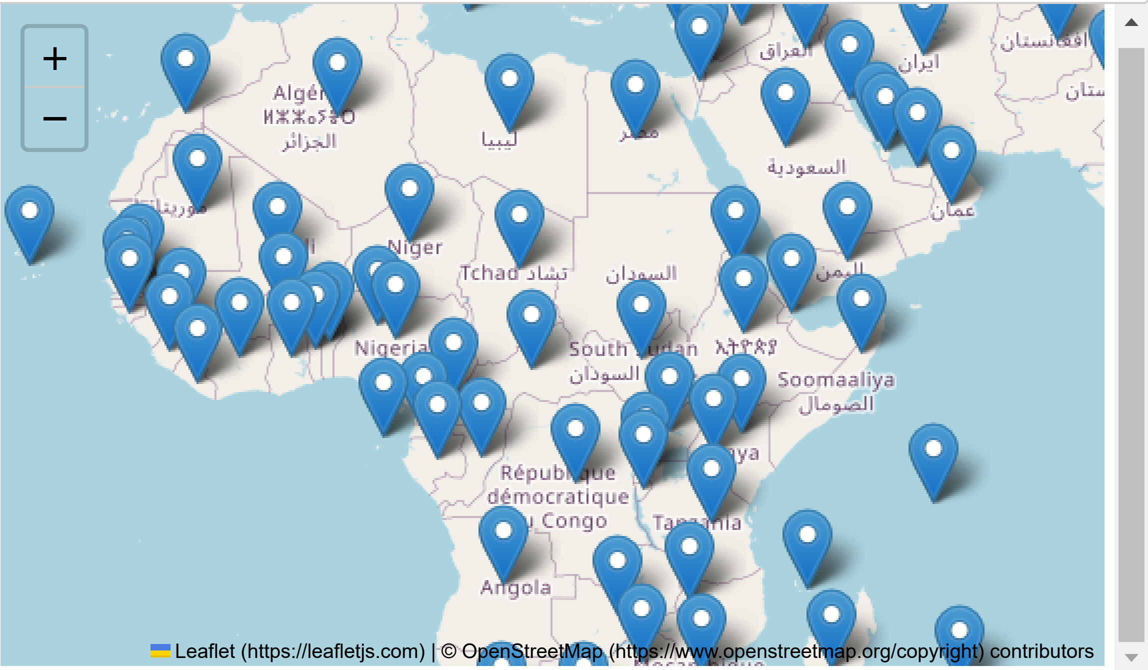
Now going to do the same for Recovered and Death cases

In [55]:

```
for id,row in df2.iterrows():
    folium.Marker(location=[row['latitude'],row['longitude']],popup=row['Re

basemap
```

Out[55]:

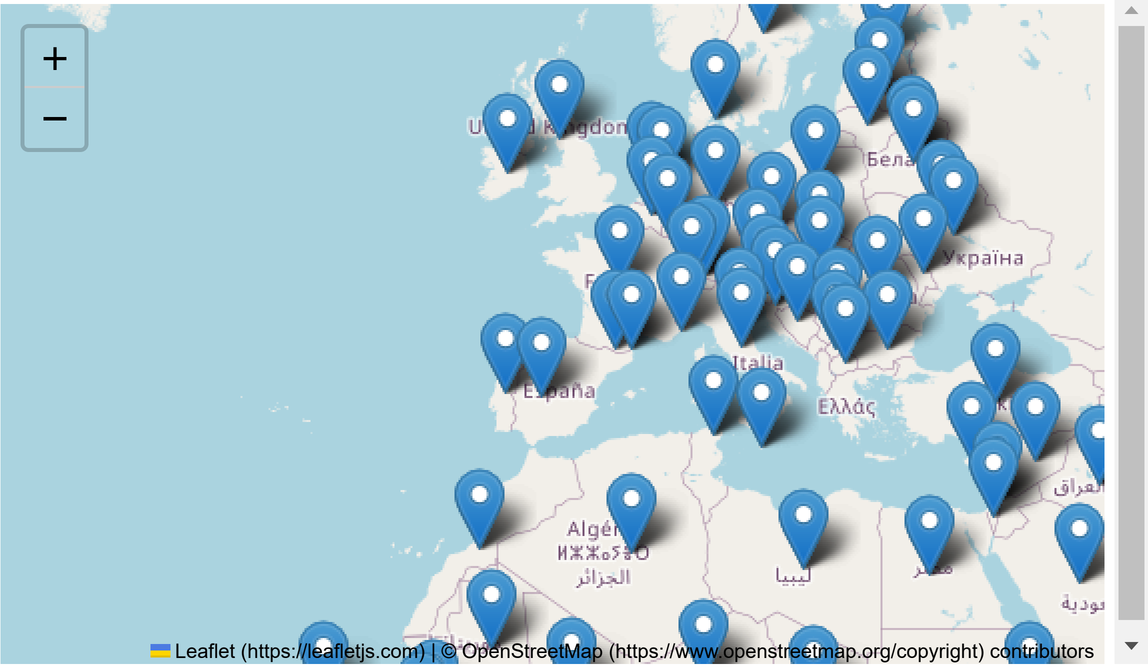

+
−
Algér
الجزائر
ليبيا
مصر
العراق
ايران
افغانستان
السعودية
عمان
Niger
Tchad تشاد
السودان
اليمن
South Sudan
ኢትዮጵያ
Soomaaliya
الصومال
Nigeria
République démocratique du Congo
Tanzania
Angola
Leaflet (https://leafletjs.com) | © OpenStreetMap (https://www.openstreetmap.org/copyright) contributors

In [56]:

```
for id,row in df2.iterrows():
    folium.Marker(location=[row['latitude'],row['longitude']],popup=row['De

basemap
```

Out[56]:


+
−
United Kingdom
Бела
Україна
Italia
España
Ελλάς
Algér
الجزائر
ليبيا
مصر
العراق
Leaflet (https://leafletjs.com) | © OpenStreetMap (https://www.openstreetmap.org/copyright) contributors

Now interested in doing a marker cluster for the total number of deaths for each country.

In [58]:

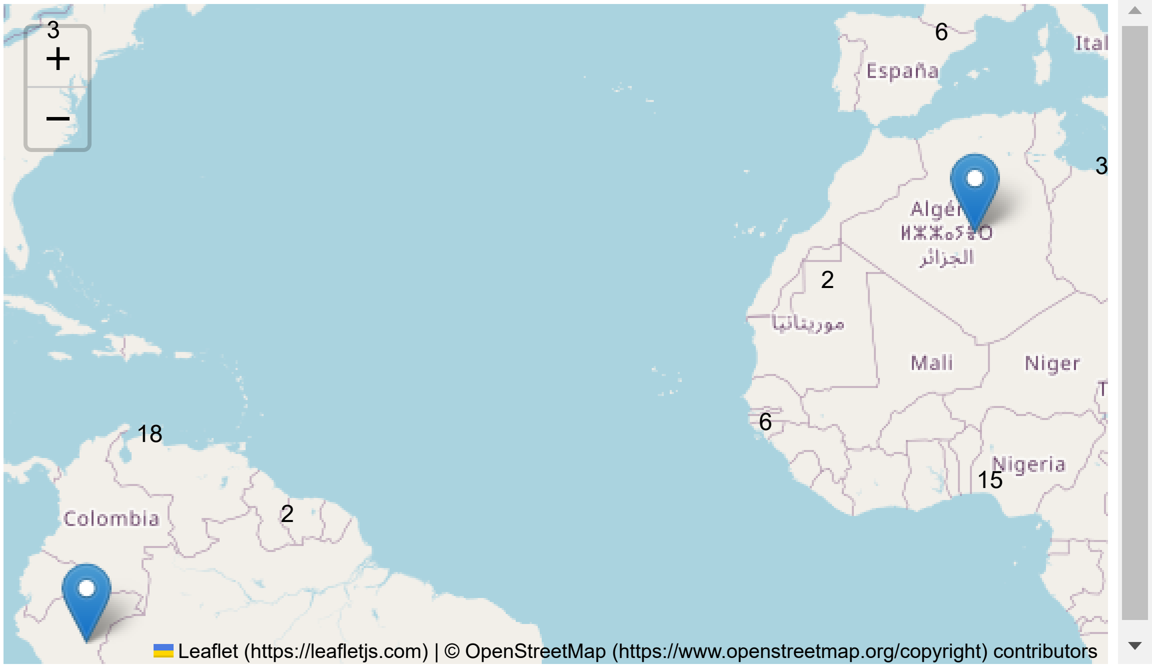
```
basemap2=folium.Map(zoom_start=3)

from folium.plugins import MarkerCluster
MC=MarkerCluster()

for id,row in df2.iterrows():
    MC.add_child(folium.Marker(location=[row['latitude'],row['longitude']],
basemap2.add_child(MC)


basemap2
```

Out[58]:



Leaflet (https://leafletjs.com) | © OpenStreetMap (https://www.openstreetmap.org/copyright) contributors

Now let's plot a geographical heat map

In [59]:

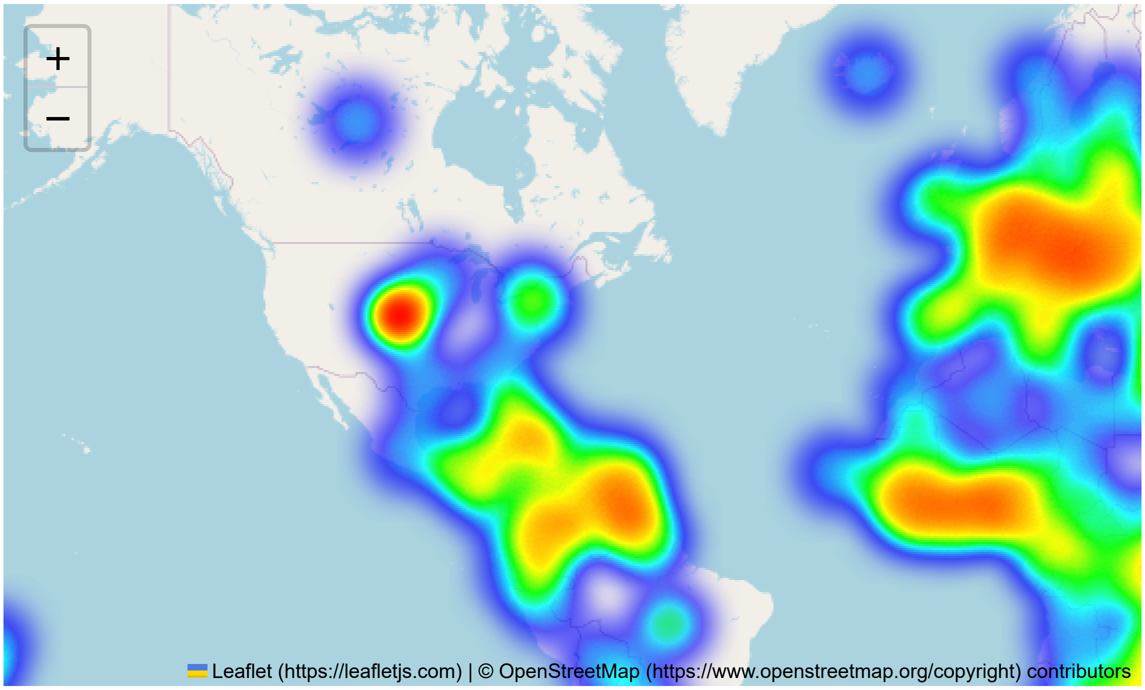
```
basemap2=folium.Map(zoom_start=3)

from folium.plugins import HeatMap
```

In [63]:

```
(HeatMap(data=df2[['latitude','longitude','Deaths']],radius=20)).add_to(bas
basemap2
```

Out[63]:


+
−
Leaflet (https://leafletjs.com) | © OpenStreetMap (https://www.openstreetmap.org/copyright) contributors