World Map Geovisualization using Folium and Covid Data

Instructor: Vinita Silaparasetty

Import Libraries

In [2]:
```
pip install folium
```

```
Collecting folium
  Downloading folium-0.11.0-py2.py3-none-any.whl (93 kB)
Requirement already satisfied: jinja2>=2.9 in c:\users\ammuuu\appdata\local\p
rograms\python\python37\lib\site-packages (from folium) (2.11.2)
Requirement already satisfied: numpy in c:\users\ammuuu\appdata\local\program
s\python\python37\lib\site-packages (from folium) (1.19.1)
Requirement already satisfied: requests in c:\users\ammuuu\appdata\local\prog
rams\python\python37\lib\site-packages (from folium) (2.24.0)
Collecting branca>=0.3.0
  Downloading branca-0.4.1-py3-none-any.whl (24 kB)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\ammuuu\appdata\lo
cal\programs\python\python37\lib\site-packages (from jinja2>=2.9->folium) (1.
1.1)
Requirement already satisfied: idna<3,>=2.5 in c:\users\ammuuu\appdata\local
\programs\python\python37\lib\site-packages (from requests->folium) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\ammuuu\appdata
\local\programs\python\python37\lib\site-packages (from requests->folium) (20
20.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\ammuuu\appdata\l
ocal\programs\python\python37\lib\site-packages (from requests->folium) (3.0.
4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\ammuuu\appdata\local\programs\python\python37\lib\site-packages (fro
m requests->folium) (1.25.9)Note: you may need to restart the kernel to use u
pdated packages.

Installing collected packages: branca, folium
Successfully installed branca-0.4.1 folium-0.11.0
```

In [3]:
```
pip install requests mimetypes http
```

```
Requirement already satisfied: requests in c:\users\ammuuu\appdata\local\prog
rams\python\python37\lib\site-packages (2.24.0)
Note: you may need to restart the kernel to use updated packages.

ERROR: Could not find a version that satisfies the requirement mimetypes (fro
m versions: none)
ERROR: No matching distribution found for mimetypes
```

In [4]:
```
import json
import folium
import requests
import mimetypes
import http.client
import pandas as pd
from folium.plugins import HeatMap
from pandas.io.json import json_normalize
```

Access Data via API

In [7]:
```
conn = http.client.HTTPSConnection('api.covid19api.com')
#where data is stored
payload=''
#header stores the algorithm
headers={}
conn.request('GET', '/summary', payload, headers)
res=conn.getresponse()
data=res.read().decode('UTF-8')
```

Convert Data to JSON

In [13]:
```
covid1=json.loads(data)
```

Normalize Data

In [15]:
```
pd.json_normalize(covid1['Countries'], sep=',')
```

Out[15]:

	Country	CountryCode	Slug	NewConfirmed	TotalConfirmed	NewDeaths	TotalDeat
0	Afghanistan	AF	afghanistan	47	38243	3	14
1	Albania	AL	albania	122	9728	6	2
2	Algeria	DZ	algeria	325	45158	5	15
3	Andorra	AD	andorra	15	1199	0	
4	Angola	AO	angola	48	2777	3	1
...	...	...	...	...	...	...	
181	Viet Nam	VN	vietnam	2	1046	0	
182	Western Sahara	EH	western-sahara	0	10	0	
183	Yemen	YE	yemen	14	1976	1	5
184	Zambia	ZM	zambia	34	12415	2	2
185	Zimbabwe	ZW	zimbabwe	79	6638	3	2

186 rows × 10 columns

Convert to Pandas DataFrame

In [16]:
```
df=pd.DataFrame(covid1['Countries'])
df
```

Out[16]:

	Country	CountryCode	Slug	NewConfirmed	TotalConfirmed	NewDeaths	TotalDeat
0	Afghanistan	AF	afghanistan	47	38243	3	14
1	Albania	AL	albania	122	9728	6	2
2	Algeria	DZ	algeria	325	45158	5	15
3	Andorra	AD	andorra	15	1199	0	
4	Angola	AO	angola	48	2777	3	1
...	...	...	...	...	...	...	
181	Viet Nam	VN	vietnam	2	1046	0	
182	Western Sahara	EH	western-sahara	0	10	0	
183	Yemen	YE	yemen	14	1976	1	5
184	Zambia	ZM	zambia	34	12415	2	2
185	Zimbabwe	ZW	zimbabwe	79	6638	3	2

186 rows × 11 columns

Drop Unnecessary Columns

In [17]:
```
covid2=df.drop(columns=['CountryCode', 'Slug', 'Date', 'Premium'], axis=1)
covid2
```

Out[17]:

	Country	NewConfirmed	TotalConfirmed	NewDeaths	TotalDeaths	NewRecovered	TotalR
0	Afghanistan	47	38243	3	1409	84	
1	Albania	122	9728	6	296	141	
2	Algeria	325	45158	5	1523	253	
3	Andorra	15	1199	0	53	1	
4	Angola	48	2777	3	112	31	
...	...	...	...	...	...	...	
181	Viet Nam	2	1046	0	34	11	
182	Western Sahara	0	10	0	1	0	
183	Yemen	14	1976	1	571	15	
184	Zambia	34	12415	2	292	15	
185	Zimbabwe	79	6638	3	206	9	

186 rows × 7 columns

Generate Base Map

In [18]:
```
m=folium.Map(tiles='Stamen Terrain', min_zoom=1.5)
m
```

Out[18]: Make this Notebook Trusted to load map: File -> Trust Notebook

Obtain Geodata

geodata url: https://raw.githubusercontent.com/python-visualization/folium/master/examples/data (https://raw.githubusercontent.com/python-visualization/folium/master/examples/data)

geo data file: world-countries.json

In [23]:
```
url='https://raw.githubusercontent.com/python-visualization/folium/master/examples/data'
country_shapes=f'{url}/world-countries.json'
```

Generate Choropleth Map Layer

In [25]:
```
folium.Choropleth(
    geo_data=country_shapes,
    min_zoom=2,
    name='Covid-19',
    data=covid2,
    columns=['Country', 'TotalConfirmed'],
    key_on='feature.properties.name',
    fill_color='OrRd',
    nan_fill_color='black',
    legend_name='Total Confirmed Covid Cases'
).add_to(m)
m
```

Out[25]: Make this Notebook Trusted to load map: File -> Trust Notebook

Generate Circular Markers

In [26]:
```
covid2.update(covid2['TotalConfirmed'].map('Total Confirmed: {}'.format))
covid2.update(covid2['TotalRecovered'].map('Total Recovered: {}'.format))
```

coordinates of countries: https://raw.githubusercontent.com/VinitaSilaparasetty/covid-map/master/country-coordinates-world.csv (https://raw.githubusercontent.com/VinitaSilaparasetty/covid-map/master/country-coordinates-world.csv)

In [28]:
```
coordinates = pd.read_csv('https://raw.githubusercontent.com/VinitaSilaparasetty/covid-map/master/country-coordinates-world.csv')
```

In [29]:
```
coordinates
```

Out[29]:

	latitude	longitude	Country
0	33.939110	67.709953	Afghanistan
1	41.153332	20.168331	Albania
2	28.033886	1.659626	Algeria
3	-14.270972	-170.132217	American Samoa
4	42.546245	1.601554	Andorra
...	...	...	...
239	-13.768752	-177.156097	Wallis and Futuna
240	24.215527	-12.885834	Western Sahara
241	15.552727	48.516388	Yemen
242	-13.133897	27.849332	Zambia
243	-19.015438	29.154857	Zimbabwe

244 rows × 3 columns

In [31]:
```
#ensure only countries that have data are represented, do this by
#an inner merge, so only what is common will be extracted (if data is present, it is common)
covid_final = pd.merge(covid2, coordinates, on="Country")
```

In [32]:
```
def plotDot(point):
    folium.CircleMarker(location=[point.latitude, point.longitude],
                        radius=5,
                        weight=2,
                        popup=[point.Country, point.TotalConfirmed, point.TotalRecovered],
                        fill_color='#000000').add_to(m)
```

In [33]:
```
covid_final.apply(plotDot, axis=1)
m.fit_bounds(m.get_bounds())
m
```

Out[33]: Make this Notebook Trusted to load map: File -> Trust Notebook

Generate Base Map

In [34]:
```
m1=folium.Map(tiles='StamenToner', min_zoom=2)
m1
```

Out[34]: Make this Notebook Trusted to load map: File -> Trust Notebook

Generate Heat Map Layer

In [42]:
```
#data need to be of type float
deaths = covid_final['TotalDeaths'].astype(float)
```

In [43]:
```
lat=covid_final['latitude'].astype(float)
```

In [44]:
```
lon=covid_final['longitude'].astype(float)
```

In [46]:
```
m1.add_child(HeatMap(zip(lat, lon, deaths), radius=0))
```

Out[46]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [48]:
```
def plotDot(point):
    folium.CircleMarker(location=[point.latitude, point.longitude],
                        radius=5,
                        weight=2,
                        popup=[point.Country, point.TotalDeaths],
                        fill_color='#000000').add_to(m1)

covid_final.apply(plotDot, axis=1)
m1.fit_bounds(m1.get_bounds())
m1
```

Out[48]: Make this Notebook Trusted to load map: File -> Trust Notebook