

Google Map Data Acquisition

May 17, 2020

1 Clé API

AIzaSyB7o00WEuHRv8G_46zZYL7DX2ghzIGcZsw

2 Importation des modules de python

```
[2]: import os
import pandas as pd
import geopandas as gpd
from shapely.geometry import point
import gmmaps
import gmmaps.datasets
import gmmaps.geojson_geometries
import json
import pandas as pd
```

2.1 Configuration de L' API

```
[3]: gmmaps.configure(api_key='AIzaSyB7o00WEuHRv8G_46zZYL7DX2ghzIGcZsw') # Fill in
    ↳with your API key
```

2.2 Téléchargement des données des séismes

on peut télécharger des différent données selon nos besoin

```
[108]: earthquake_data = gmmaps.datasets.load_dataset_as_df('earthquakes')
```

2.2.1 Affichage des 5 premières lignes

On peut extraire ces coordonnées et les transfromé en fichier shp pour les visualiser dans arcgis

```
[109]: earthquake_data.head()
```

```
[109]:    latitude  longitude  magnitude
0    65.193300 -149.072500         1.70
1    38.791832 -122.780830         2.10
```

```

2  38.818001 -122.792168      0.48
3  33.601667 -116.727667      0.78
4  37.378334 -118.520836      3.64

```

3 Acquisition de Google Map

4 Chargement de la liste des pays pour notre projet

```
[110]: countries_geojson = gmaps.geojson_geometries.load_geometry('countries')
```

4.1 Affichage des données

```
[ ]: print(countries_geojson)
```

On voit que ces données la sont sous forme geojson format .

4.2 Sauvgard du fichier

```
[43]: with open("SigDataAnalyticsforGoogleMap/CountryData.gejson", "a") as outfile:
      json.dump(countries_geojson, outfile)
```

5 Chargement du fichier en utilisant geopandas

```
[44]: country_load = gpd.read_file("SigDataAnalyticsforGoogleMap/CountryData.geojson")
```

5.1 Affichage des 5 premières lignes

```
[45]: country_load.head()
```

```
[45]:   ISO_A3      name      geometry
0    AFG  Afghanistan  POLYGON ((71.04980 38.40866, 71.33438 38.28066...
1    AGO      Angola  MULTIPOLYGON (((23.96746 -10.87231, 24.00373 -...
2    ALB    Albania  POLYGON ((20.06496 42.54676, 20.13555 42.50963...
3    ALA      Aland  POLYGON ((19.91847 60.37018, 20.09303 60.35513...
4    AND    Andorra  POLYGON ((1.70701 42.50278, 1.63952 42.46643, ...
```

On veut avoir seulement les pays d’afrique .

```
[46]: Africa_Country=['Algeria',
  'Angola',
  'Benin',
  'Botswana',
  'Burkina Faso',
```

'Burundi',
'Cameroon',
'Cape Verde',
'Central African Republic',
'Chad',
'Comoros',
'Democratic Republic of the Congo',
'Republic of the Congo',
'Djibouti',
'Egypt',
'Equatorial Guinea',
'Eritrea',
'Ethiopia',
'Gabon',
'Gambia',
'Ghana',
'Guinea',
'Guinea-Bissau',
'Ivory Coast',
'Kenya',
'Lesotho',
'Liberia',
'Libya',
'Madagascar',
'Malawi',
'Mali',
'Mauritania',
'Mauritius',
'Morocco',
'Mozambique',
'Namibia',
'Niger',
'Nigeria',
'Rwanda',
'Sao Tome and Principe',
'Senegal',
'Seychelles',
'Sierra Leone',
'Somalia',
'South Africa',
'South Sudan',
'Sudan',
'Swaziland',
'Tanzania',
'Togo',
'Tunisia',
'Uganda',

```
'Zambia',  
'Zimbabwe']
```

5.2 Selection des pays D'Afrique

```
[83]: africa = country_load.loc[country_load['name'].isin(Africa_Country),['name',  
    → 'geometry']]  
africa.head()
```

```
[83]:
```

	name	geometry
1	Angola	MULTIPOLYGON (((23.96746 -10.87231, 24.00373 -...
15	Burundi	POLYGON ((30.55460 -2.40063, 30.41600 -2.64557...
17	Benin	POLYGON ((3.59640 11.69577, 3.50493 11.55656, ...
18	Burkina Faso	POLYGON ((0.21847 14.91098, 0.18410 14.81956, ...
31	Botswana	POLYGON ((25.25978 -17.79411, 25.22609 -17.931...

6 Sauvgarde du fichier sous forme shp

Avant DE sauvgarder le fichier il faut définir le crs et la geometry

```
[52]: crs = {'init': 'epsg:27700'}  
gdf = gpd.GeoDataFrame(africa, crs=crs, geometry = africa.geometry)  
gdf.to_file("SigDataAnalyticsforGoogleMap/africa.shp")
```

6.0.1 Pour cela on a terminé l'étape de l'acquisition de données à partir de google map . Maintenant On veut avoir les données de covid 19

7 Chargement de mes données de covid 19

Ces données sont analysées et traitées, vous trouvez tout le processus dans ma page de github [GitHub-COVID-19](#)

```
[95]: data_africa_covid=pd.read_csv("SigDataAnalyticsforGoogleMap/CovidAfrica.csv")
```

```
[96]: data_africa_covid.tail()
```

```
[96]:
```

	Unnamed: 0	Country/Region	DRR	Lat	Long
41	3443	Togo	9.375000	8.6195	0.8248
42	3525	Tunisia	41.891892	34.0000	9.0000
43	3607	Uganda	0.000000	1.0000	32.0000
44	3689	Zambia	6.250000	-15.4167	28.2833
45	3771	Zimbabwe	100.000000	-20.0000	30.0000

8 Conversion des données (lat et long) au geometry

```
[99]: geometry = [point.Point(xy) for xy in zip(data_africa_covid.Long,
→data_africa_covid.Lat)]
```

```
[100]: data_africa_covid['geometry']=geometry
```

8.1 Affichage des 5 premières lignes

```
[101]: data_africa_covid.tail()
```

```
[101]:
```

	Unnamed: 0	Country/Region	DRR	Lat	Long	\
41	3443	Togo	9.375000	8.6195	0.8248	
42	3525	Tunisia	41.891892	34.0000	9.0000	
43	3607	Uganda	0.000000	1.0000	32.0000	
44	3689	Zambia	6.250000	-15.4167	28.2833	
45	3771	Zimbabwe	100.000000	-20.0000	30.0000	


```
geometry
```

41	POINT (0.8248 8.6195)
42	POINT (9 34)
43	POINT (32 1)
44	POINT (28.2833 -15.4167)
45	POINT (30 -20)

9 Sauvgard les données sous forme shp

```
[102]: crs = {'init': 'epsg:4326'}
gdf_africa = gpd.GeoDataFrame(data_africa_covid, crs=crs,
→geometry=data_africa_covid.geometry)
gdf_africa.to_file("SigDataAnalyticsforGoogleMap/africa_covid.shp")
```

```
[ ]:
```

```
[ ]:
```