Geographical Dataset, Geopandas, and Plotly

August 26, 2020

1 Geogrphical Dataset, Geopandas and Plotly

This notebook presents how shapefiles and datasets with longitude and latitude data can be manipulated and visualized.

Components * Geographical Dataset: Dataset for Exploratory Data Analytics * Geopandas: used for Data Processing * Plotly: used for Visualiztaion

Considerations and Assumptions * For testing, region shapefile is used to minimize processing time * For testing, masked sample data is used with over 7k data points * User has a mapbox token. Mapbox tokens are offered for free. See https://docs.mapbox.com/help/tutorials/get-started-tokens-api/ on how to create mapbox tokens.

Definition of Terms * Dataframe / Dataset - Sample Scraped Data * GeoDataFrame / GeoData - Dataframe converted to geometrical data. Long, lat converted geometrically * Shapefile - Geospatial data (usually referred to bounded map) * Polygon / Overlay - How shapefile is bounded * GeoJson - geometrical json file

Challenges * Given a polygon from shapefiles and dataset with longitude and latitude columns, how do I 'merge' data from shapefiles to the datapoints? * Given a shapefile and a dataset, how do I create a choropleth map using plotly?

Objectives

1. Transform shapefiles and dataset for analysis
2. Spatially join information from region shapefile to each data point
3. Render choropleth map only basing on shapefile

Data Sources * http://philgis.org/country-vector-and-raster-datasets : For shapefiles * Sample data : Masked, scraped data of real estate properties

2 Dependencies

[76]:

```
import pandas as pd
import json
import geopandas as gpd
import plotly.graph_objects as go
```

3 OBJ 1: Transform shapefiles and dataset for analysis

3.1 Steps:

1. Setup
2. Dataset: DataFrame -> GeoDataFrame
3. Polygons: Shapefile -> GeoDataFrame
4. Shapefile GeoJSON: Shapefile GeoDataFrame -> Shapefile GeoJSON

3.1.1 Step 1: Setup

Import dataset

[33]:

```
dataset_df = pd.read_json('data/sample_set.json')
dataset_df
```

[33]:

```
        attributes.location_longitude  attributes.location_latitude  \
0                          120.498827                     14.796130   
1                          121.386902                     14.236900   
2                          121.044663                     14.484157   
3                          120.914866                     14.264605   
4                          121.386902                     14.236900   
…                                …                           …   
120391                     121.053876                     14.580422   
120392                     121.053897                     14.580562   
120393                     121.053604                     14.581224   
120394                     120.902037                     14.410720   
120395                     123.884242                     10.286925   

                location.area  location.city location.region    values  \
0                     Tugatog          Orani          Bataan  14000000   
1                  Bulakin II        Dolores          Quezon    166750   
2       Marcelo Green Village      Parañaque     Metro Manila   3500000   
3                    Javalera  General Trias          Cavite   1695560   
4                  Bulakin II        Dolores          Quezon    186875   
…                        …            …             …       …   
120391          Highway Hills    Mandaluyong     Metro Manila   7791860   
120392          Highway Hills    Mandaluyong     Metro Manila   6348699   
120393         Shaw Boulevard    Mandaluyong     Metro Manila   4200000   
120394            Alapan II-A           Imus          Cavite   2091000   
120395              Mambaling           Cebu            Cebu     15000   

                         sku  
0       NO094HO45DHIINTRESPH  
1          LA5AB3495E74DBAPH  
2          CD5BFBAFFAEB1A8PH  
3          LA5AAF05C0536FFPH  
4          LA5AB3495F1E542PH  
…                       …  
```

```
120391     CD5C9DDFF5E59B3PH
120392     CD5BA1F92A81488PH
120393     CD5DA433E093AACPH
120394  NO230HO60DUHINTRESPH
120395     CD5D86EC221842CPH

[120396 rows x 7 columns]
```

(Exploratory Data Analysis)

From the table given, we can see that it has geographical coordinates: *attributes.location_longitude*, *attributes_location_latitude*

3.1.2 Step 2: Dataset: DataFrame -> GeoDataFrame

Dataset needs to be converted to GeoDataFrame to convert coordinate columns to geometrically readable coordinates

Reference * https://geopandas.org/gallery/create_geopandas_from_pandas.html

Remark * CRS needs to match. CRS is related to map projection standard. Usually EPSG:4326 is standard. Double check shapefile's CRS data

[34]:

```
dataset_gdf = gpd.GeoDataFrame(dataset_df, geometry=gpd.
 ↪points_from_xy(dataset_df['attributes.location_longitude'],␣
 ↪dataset_df['attributes.location_latitude']), crs='EPSG:4326')
dataset_gdf
```

[34]:

```
        attributes.location_longitude  attributes.location_latitude  \
0                          120.498827                     14.796130
1                          121.386902                     14.236900
2                          121.044663                     14.484157
3                          120.914866                     14.264605
4                          121.386902                     14.236900
…                               …                           …
120391                     121.053876                     14.580422
120392                     121.053897                     14.580562
120393                     121.053604                     14.581224
120394                     120.902037                     14.410720
120395                     123.884242                     10.286925

                location.area  location.city location.region    values  \
0                     Tugatog          Orani          Bataan  14000000
1                  Bulakin II        Dolores          Quezon    166750
2       Marcelo Green Village      Parañaque    Metro Manila   3500000
3                    Javalera  General Trias          Cavite   1695560
4                  Bulakin II        Dolores          Quezon    186875
…                       …            …             …       …
120391          Highway Hills    Mandaluyong    Metro Manila   7791860
```

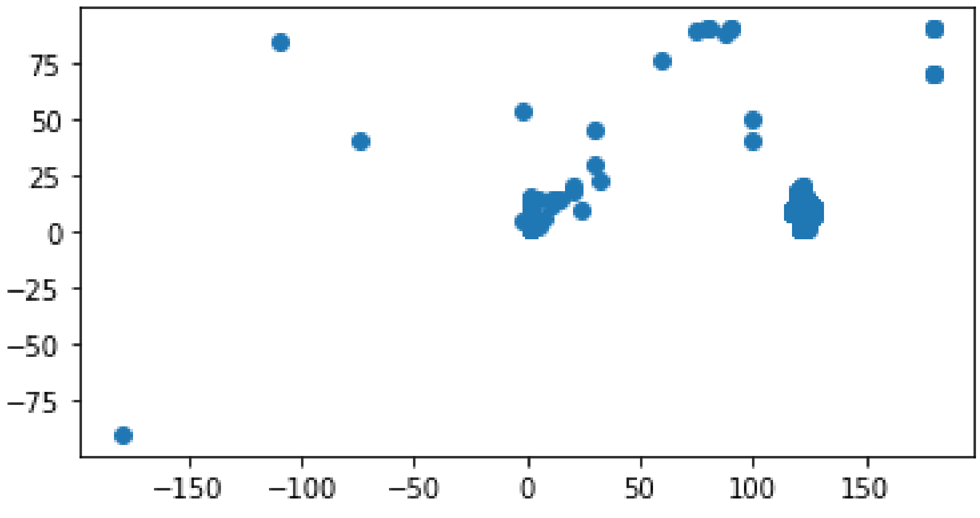
```
120392          Highway Hills    Mandaluyong    Metro Manila   6348699
120393         Shaw Boulevard    Mandaluyong    Metro Manila   4200000
120394             Alapan II-A           Imus          Cavite   2091000
120395               Mambaling           Cebu            Cebu     15000

                        sku                       geometry
0       N0094H045DHIINTRESPH  POINT (120.49883 14.79613)
1          LA5AB3495E74DBAPH  POINT (121.38690 14.23690)
2          CD5BFBAFFAEB1A8PH  POINT (121.04466 14.48416)
3          LA5AAF05C0536FFPH  POINT (120.91487 14.26460)
4          LA5AB3495F1E542PH  POINT (121.38690 14.23690)
…                       …                          …
120391     CD5C9DDFF5E59B3PH  POINT (121.05388 14.58042)
120392     CD5BA1F92A81488PH  POINT (121.05390 14.58056)
120393     CD5DA433E093AACPH  POINT (121.05360 14.58122)
120394  N0230H060DUHINTRESPH  POINT (120.90204 14.41072)
120395     CD5D86EC221842CPH  POINT (123.88424 10.28693)

[120396 rows x 8 columns]
```

[70]:

```python
dataset_gdf.plot()
```

[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd14f99450>
```



(Exploratory Data Analysis)

From the table given, added column for geometry

3.1.3 Step 3: Polygons: Shapefile -> GeoDataFrame

Shapefile needs to be converted to GeoDataFrame for further data processing

[37]:

```python
regions_gdf = gpd.read_file('data/ph_regions.shp')
regions_gdf
```

[37]:

```
                                         REGION  \
0   Autonomous Region of Muslim Mindanao (ARMM)
1                       Bicol Region (Region V)
2                      CALABARZON (Region IV-A)
3                    Cagayan Valley (Region II)
4                          Caraga (Region XIII)
5                    Central Luzon (Region III)
6                  Central Visayas (Region VII)
7        Cordillera Administrative Region (CAR)
8                      Davao Region (Region XI)
9                 Eastern Visayas (Region VIII)
10                     Ilocos Region (Region I)
11                       MIMAROPA (Region IV-B)
12                          Metropolitan Manila
13                 Northern Mindanao (Region X)
14                    SOCCSKSARGEN (Region XII)
15                  Western Visayas (Region VI)
16              Zamboanga Peninsula (Region IX)

                                             geometry
0   MULTIPOLYGON (((119.46694 4.58694, 119.46639 4…
1   MULTIPOLYGON (((122.98417 11.71056, 122.98333 …
2   MULTIPOLYGON (((125.22166 10.43444, 125.22195 …
3   MULTIPOLYGON (((122.47040 16.91995, 122.47040 …
4   MULTIPOLYGON (((126.41750 7.96417, 126.41778 7…
5   MULTIPOLYGON (((120.62363 14.36788, 120.62368 …
6   MULTIPOLYGON (((123.27111 9.08476, 123.27173 9…
7   POLYGON ((121.37679 17.95473, 121.36825 17.939…
8   MULTIPOLYGON (((125.39778 5.43583, 125.39778 5…
9   MULTIPOLYGON (((125.07361 9.89472, 125.07333 9…
10  MULTIPOLYGON (((120.39095 17.50012, 120.39131 …
11  MULTIPOLYGON (((117.31389 7.51417, 117.31416 7…
12  MULTIPOLYGON (((120.97972 14.49306, 120.98000 …
13  MULTIPOLYGON (((123.62193 7.82859, 123.62172 7…
14  POLYGON ((124.53799 7.68187, 124.54649 7.68032…
15  MULTIPOLYGON (((122.43522 9.64382, 122.43490 9…
16  MULTIPOLYGON (((122.06223 6.87278, 122.06250 6…
```

[38]:

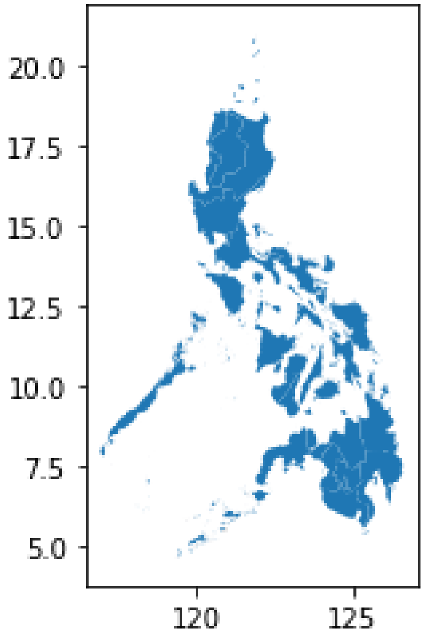
```python
regions_gdf.crs
```

[38]:

```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

[69]:
```
regions_gdf.plot()
```

[69]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efd178a18d0>
```



(Exploratory Data Analysis)

From the EDA above, CRS is **EPSG:4326** which validates CRS initialized in **Step 1**

3.1.4 Step 4: Shapefile GeoJSON: Shapefile GeoDataFrame -> Shapefile GeoJSON

Shapefile needs to be converted to json for visualization

[78]:
```
regions_json = json.loads(regions_gdf.to_json())
```

regions_json results

```
{'type': 'FeatureCollection',
 'features': [{'id': '0',
   'type': 'Feature',
   'properties': {'REGION': 'Autonomous Region of Muslim Mindanao (ARMM)'},
   'geometry': {'type': 'MultiPolygon',
    'coordinates': [[[[119.46694183349618, 4.586939811706523],
```

(Exploratory Data Analysis)

From json above, it can be seen that under features key, following key-pair value exists: *_id, properties, geometry*

Remark This is important note when setting up the choropleth map

4 OBJ 2: Spatially join information from region shapefile to each data point

This maps region polygon / shapefile information to datapoints inside respective polygon

[44]:

```python
dataset_x_region = gpd.sjoin(dataset_gdf, regions_gdf, op='within')
dataset_x_region
```

[44]:

```
        attributes.location_longitude  attributes.location_latitude  \
0                          120.498827                     14.796130
11                         120.860950                     14.834990
35                         120.893912                     14.836228
82                         121.550996                     15.771360
83                         120.624207                     15.128455
…                               …                            …
68693                      124.235969                      7.221250
68809                      124.250062                      7.219400
97640                      122.060608                      6.691331
102936                     124.286609                      7.062732
103802                     124.411953                      7.354087

        location.area      location.city location.region    values  \
0             Tugatog              Orani          Bataan  14000000
11              Tikay            Malolos         Bulacan   3644200
35                             Guiguinto         Bulacan         0
82           Buhangin              Baler          Aurora  17000000
83       Santo Cristo            Angeles        Pampanga  20000000
…               …                …             …       …
68693     Poblacion I           Cotabato     Maguindanao  22646000
68809       Tamontaka           Cotabato     Maguindanao    625000
97640                            Isabela         Basilan   1500000
102936  Dinaig Proper  Datu Odin Sinsuat     Maguindanao         0
103802       Dinganen             Buldon     Maguindanao  12200000

                         sku                    geometry  index_right  \
0       N0094H045DHIINTRESPH  POINT (120.49883 14.79613)            5
11         H05C52A1241D843PH  POINT (120.86095 14.83499)            5
35         C05C9AD711EED21PH  POINT (120.89391 14.83623)            5
```

```
82         LA5BD81410B273DPH  POINT (121.55100 15.77136)            5
83         LA5CCF885618009PH  POINT (120.62421 15.12846)            5
…                       …                          …          …
68693      HO5BCD06758EDCFPH   POINT (124.23597 7.22125)            0
68809      HO5BCD0678AEED7PH   POINT (124.25006 7.21940)            0
97640      CD5D9BF5B7D0660PH   POINT (122.06061 6.69133)            0
102936  NO961LA48KODINTRESPH   POINT (124.28661 7.06273)            0
103802     LA5B0288F657D2FPH   POINT (124.41195 7.35409)            0

                                                REGION
0                            Central Luzon (Region III)
11                           Central Luzon (Region III)
35                           Central Luzon (Region III)
82                           Central Luzon (Region III)
83                           Central Luzon (Region III)
…                                                  …
68693   Autonomous Region of Muslim Mindanao (ARMM)
68809   Autonomous Region of Muslim Mindanao (ARMM)
97640   Autonomous Region of Muslim Mindanao (ARMM)
102936  Autonomous Region of Muslim Mindanao (ARMM)
103802  Autonomous Region of Muslim Mindanao (ARMM)

[118853 rows x 10 columns]
```

5 OBJ 3: Render choropleth map only basing on shapefile

5.1 Steps:

1. Aggregate Dataset for Choropleth Map
2. Visualization

References * https://chart-studio.plotly.com/~empet/15238/tips-to-extract-data-from-a-geojson-di/#/

5.1.1 Step 1: Aggregate Dataset for Choropleth Map

Geopandas have functionalities of pandas DataFrame. For choropleth mapping, aggregation is needed to generated the heatmap

[54]:

```
aggregated_data = dataset_x_region[['REGION', 'values']].groupby('REGION').
 ↪mean().reset_index()
aggregated_data
```

[54]:

```
                                        REGION        values
0   Autonomous Region of Muslim Mindanao (ARMM)  5.834814e+06
1                         Bicol Region (Region V)  2.642311e+07
2                        CALABARZON (Region IV-A)  2.266276e+07
3                      Cagayan Valley (Region II)  2.804070e+07
```

```
4                            Caraga (Region XIII)  3.914598e+07
5                      Central Luzon (Region III)  1.629502e+07
6                     Central Visayas (Region VII)  1.838377e+07
7        Cordillera Administrative Region (CAR)  1.169404e+07
8                         Davao Region (Region XI)  2.118255e+07
9                    Eastern Visayas (Region VIII)  3.311153e+07
10                        Ilocos Region (Region I)  1.124419e+08
11                          MIMAROPA (Region IV-B)  8.967689e+07
12                             Metropolitan Manila  2.895184e+07
13                    Northern Mindanao (Region X)  1.769537e+07
14                       SOCCSKSARGEN (Region XII)  3.986138e+07
15                      Western Visayas (Region VI)  1.320619e+07
16                 Zamboanga Peninsula (Region IX)  1.804486e+07
```

5.1.2 Step 2: Visualization

From *Obj 1, Step 4, EDA remark*, shown in the geojson data that *'REGION'* data is nested under properties.

Thus in featureidkey, string should be in format "properties.*idkey*"

There is common mistake to ignore the prefix "properties" because this is not seen when visualizing the shapefile **GeoDataFrame** table

Always remember that shapefile needs to be converted to geojson for plotly and mapbox to read the polygons. Which is why prefix is needed since featureidkey is read from **geojson** and not from **GeoDataFrame**

GeoDataFrame is just preparatory step to convert **shapefile** to **geojson**

[75]:

```
token = open(".mapbox_token").read().strip()
fig = go.Figure(
    go.Choroplethmapbox(
        geojson=regions_json,
        featureidkey='properties.REGION',
        locations=aggregated_data['REGION'],
        z=aggregated_data['values'],
        colorscale="Viridis"
    )
)
fig.update_layout(mapbox_style="light", mapbox=dict(accesstoken=token))
fig.show()
```

[68]:

```
import plotly.express as px
fig = px.choropleth_mapbox(aggregated_data, geojson=regions_json,
                          featureidkey='properties.REGION', locations='REGION',␣
 ↪color='values')
fig.update_layout(mapbox_accesstoken=token)
fig.show()
```