Chapter 8 - Exercise 1: Geojson_hcmc

Entrée [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import geopandas as gpd
```

Entrée [2]:

```
# Câu 1: Đọc file dữ liệu ranh giới các quận - khu vực của TP.HCM
district = gpd.read_file('data\district-boundary-hcm-city.geojson')
```

Entrée [3]:

```
# Hiển thị 5 dòng đầu của dữ liệu
district.head()
```

Out[3]:

	id	name	localname	timestamp	SRID	admin_level	tags	geometry
0	3850184	Saigon South	Khu đô thị Nam Sài Gòn	2016-03-18T23:05:02	4326	5	{'name': 'Khu đô thị Nam Sài Gòn', 'name:en': ...	MULTIPOLYGON (((106.69344 10.72213, 106.69475 ...
1	3797166	Binh Thanh District	Quận Bình Thạnh	2016-03-18T23:05:02	4326	6	{'name': 'Quận Bình Thạnh', 'name:en': 'Binh T...	MULTIPOLYGON (((106.68386 10.80711, 106.68388 ...
2	2587287	District 1	Quận 1	2016-03-18T23:05:02	4326	6	{'name': 'Quận 1', 'name:en': 'District 1', 'n...	MULTIPOLYGON (((106.68165 10.76543, 106.68187 ...
3	3819816	District 3	Quận 3	2016-03-18T23:05:02	4326	6	{'name': 'Quận 3', 'name:en': 'District 3', 'n...	MULTIPOLYGON (((106.66422 10.78714, 106.66457 ...
4	2778323	District 4	Quận 4	2016-03-18T23:05:02	4326	6	{'name': 'Quận 4', 'name:en': 'District 4', 'b...	MULTIPOLYGON (((106.68639 10.75184, 106.68641 ...

Entrée [4]:

```
# Hiển thị 5 dòng cuối của dữ liệu
district.tail()
```

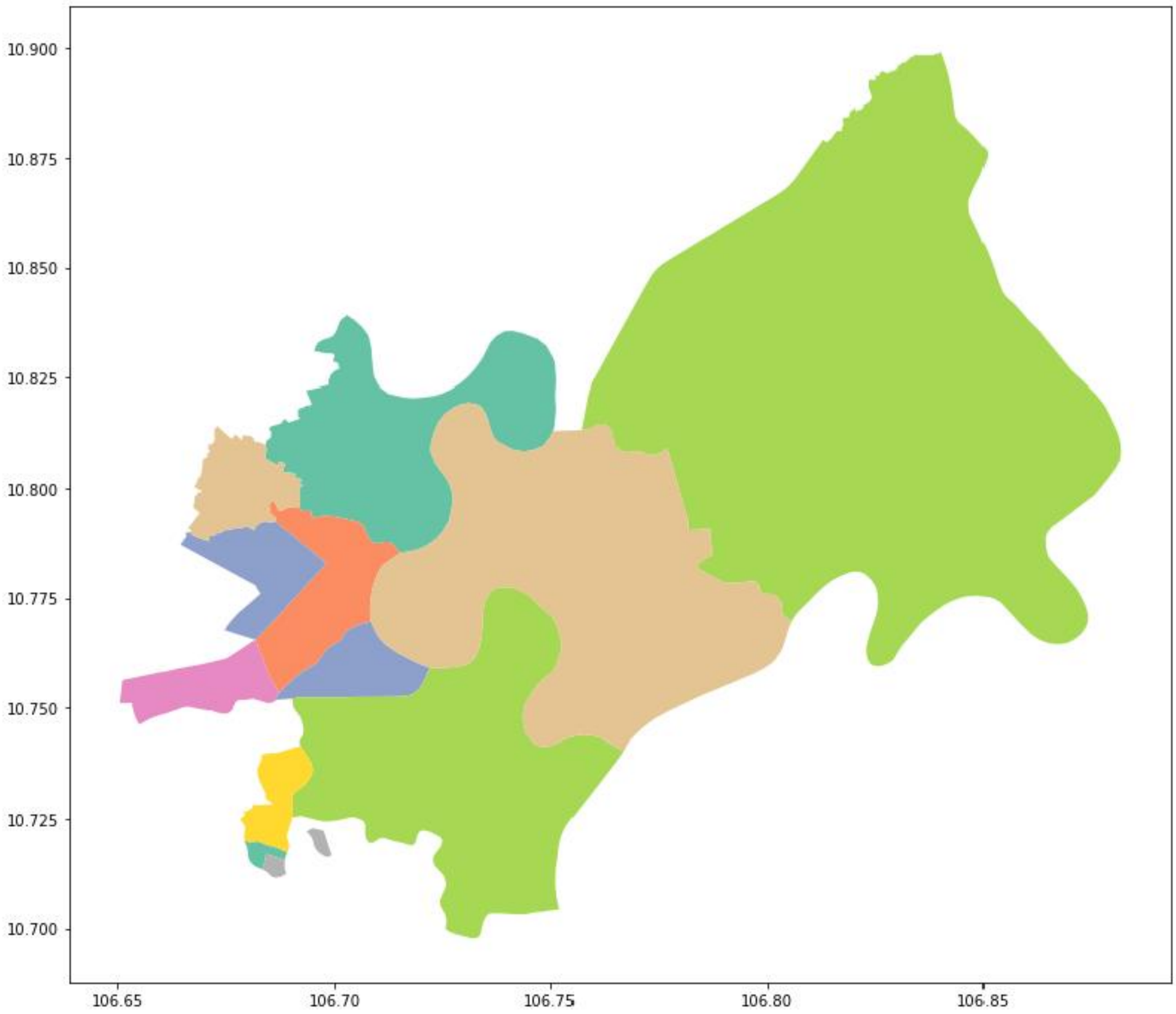
Out[4]:

	id	name	localname	timestamp	SRID	admin_level	tags	geo
8	3799817	Quận 2	Quận 2	2016-03-18T23:05:02	4326	6	{'name': 'Quận 2', 'boundary': 'administrative...	MULTIPOL (((106 10. 106.7(
9	3851694	Quận Phú Nhuận	Quận Phú Nhuận	2016-03-18T23:05:02	4326	6	{'name': 'Quận Phú Nhuận', 'boundary': 'admini...	MULTIPOL (((106 10. 106.6(
10	3853748	Khu 6	Khu 6	2016-03-18T23:05:02	4326	9	{'name': 'Khu 6', 'boundary': 'administrative'...	MULTIPOL (((106 10. 106.67
11	3854476	Dai Phuc Residences	Khu Nhà ở Rạch Bà Tánh	2016-03-18T23:05:02	4326	10	{'name': 'Khu Nhà ở Rạch Bà Tánh', 'name:en': ...	MULTIPOL (((106 10. 106.6(
12	3854477	T30 Residential Area	Khu Dân cư T30	2016-03-18T23:05:02	4326	10	{'name': 'Khu Dân cư T30', 'name:en': 'T30 Res...	MULTIPOL (((106 10. 106.6(

Entrée [5]:

```
# Hiển thị bản đồ ranh giới từ dữ liệu district, cmap = 'Set2'
plt.figure(figsize=(8,8))
district.plot(column = 'name', cmap = 'Set2', figsize=(16,12))
plt.show()
```

```
<Figure size 576x576 with 0 Axes>
```



Entrée [6]:

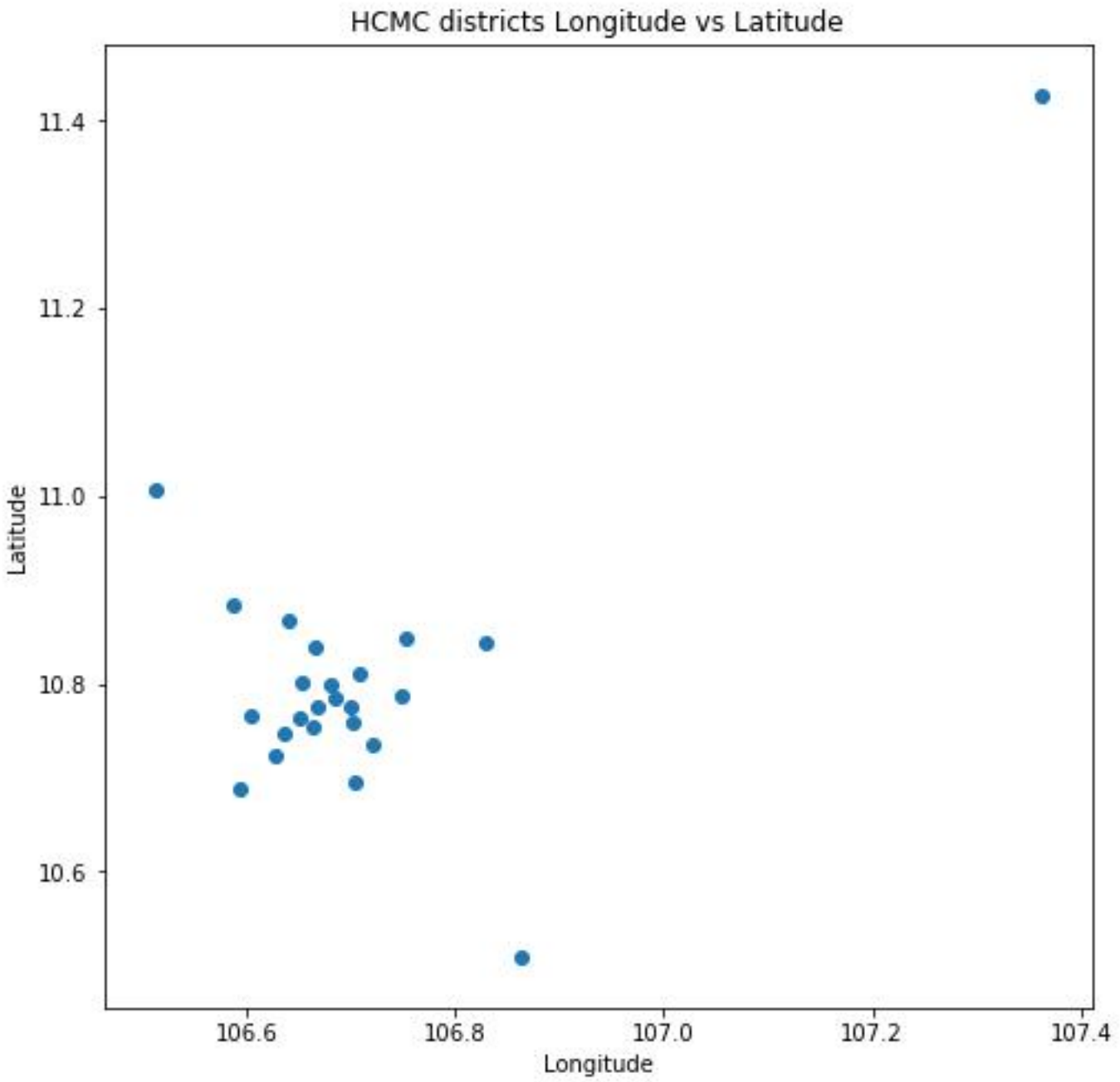
```
# Câu 2: Đọc dữ liệu trong sheet 'location' của file HCMC_location.xlsx
df_hcm = pd.read_excel('data\HCMC_location.xlsx', sheet_name='Location')
# Hiển thị 5 dòng đầu của dữ liệu
df_hcm.head()
```

Out[6]:

	STT	ID	Name	Bourough	Postal cost	Latitude	Longitude	Population	Population_all	Avg_
0	1	760	Quận 1	Vietnam, Quan 1	NaN	10.775659	106.700424	193.632	193632	
1	2	761	Quận 12	Vietnam, Quan 12	NaN	10.867153	106.641332	510.326	510326	
2	3	762	Quận Thủ Đức	Vietnam, Thu Duc	NaN	10.849409	106.753705	528.413	528413	
3	4	763	Quận 9	Vietnam, Quan 9	NaN	10.842840	106.828685	290.620	290620	
4	5	764	Quận Gò Vấp	Vietnam, Go Vap	NaN	10.838678	106.665290	634.146	634146	

Entrée [7]:

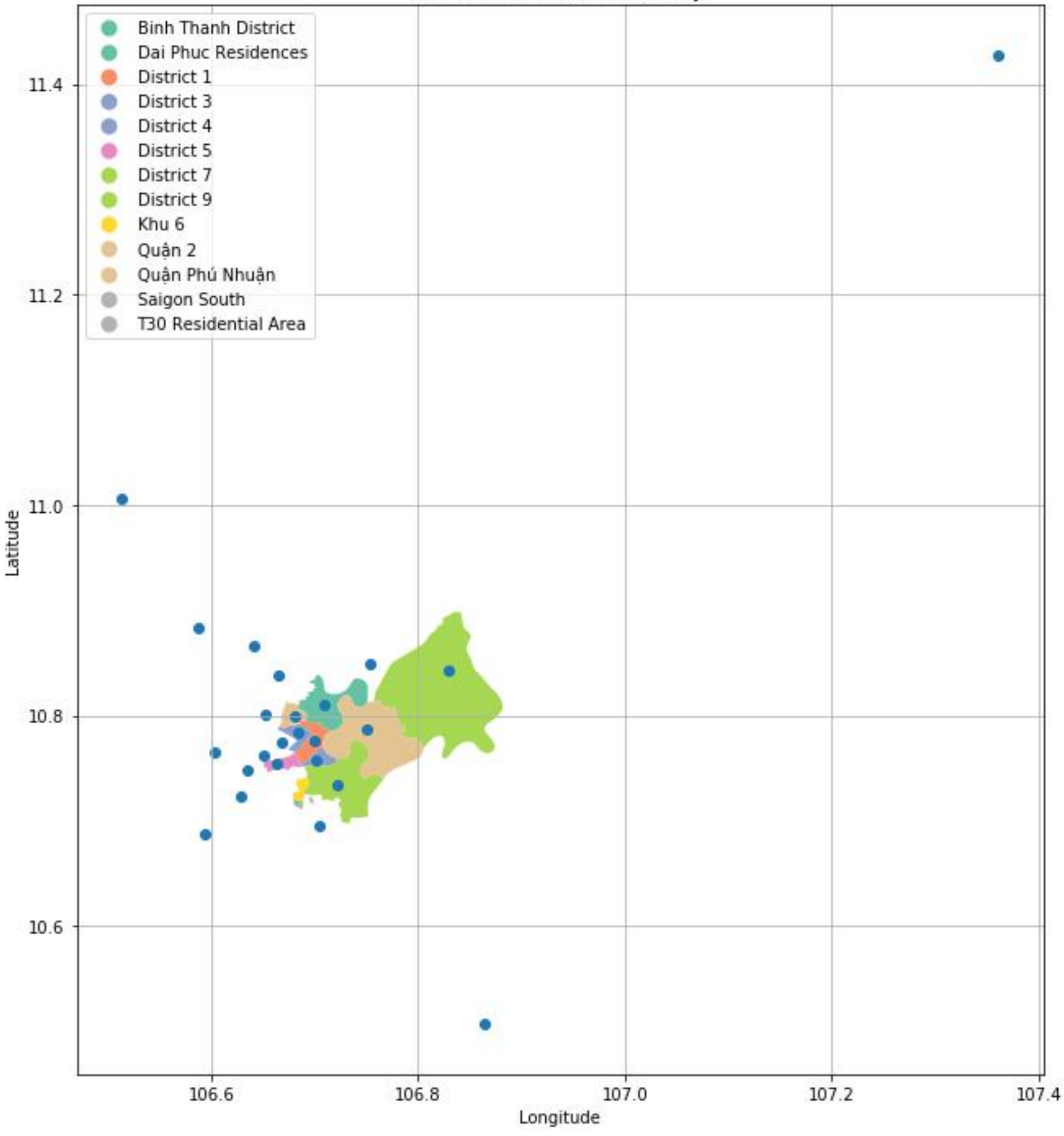
```
# Câu 3: Vẽ scatterplot với dữ liệu Longitude, Latitude của dữ liệu của câu 2
plt.figure(figsize=(8,8))
plt.scatter(df_hcm.Longitude, df_hcm.Latitude)
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.title('HCMC districts Longitude vs Latitude')
plt.show()
```



Entrée [8]:

```
# Câu 4: Vẽ bản đồ thể hiện vị trí các quận-khu vực tại TP.HCM
plt.figure(figsize=(16,12))

district.plot(column = 'name', cmap = 'Set2', legend=True, figsize=(16,12))
plt.scatter(df_hcm.Longitude, df_hcm.Latitude)
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.title('Districts in HoChiMinh in City')

plt.grid()
plt.show()
```

```
<Figure size 1152x864 with 0 Axes>
```



Chapter 8 - Exercise 2: Canada

Dữ liệu Canada.xlsx chứa thông tin nhập cư vào Canada từ năm 1980 đến năm 2013. Bộ dữ liệu chứa dữ liệu hàng năm về dòng người di cư đến Canada được ghi nhận, trình bày thông tin inflows and outflows theo nơi sinh, quốc tịch hoặc nơi cư trú trước đó / tiếp theo cho cả người nước ngoài và quốc tịch. Chúng tôi sẽ tập trung vào dữ liệu nhập cư Canada.

Entrée [1]:

```
import folium
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Map

Entrée [2]:

```
# Câu 1: Hiển thị bản đồ thế giới
world_map = folium.Map()
world_map
```

Out[2]:

Make this Notebook Trusted to load map: File -> Trust Notebook

Entrée [3]:

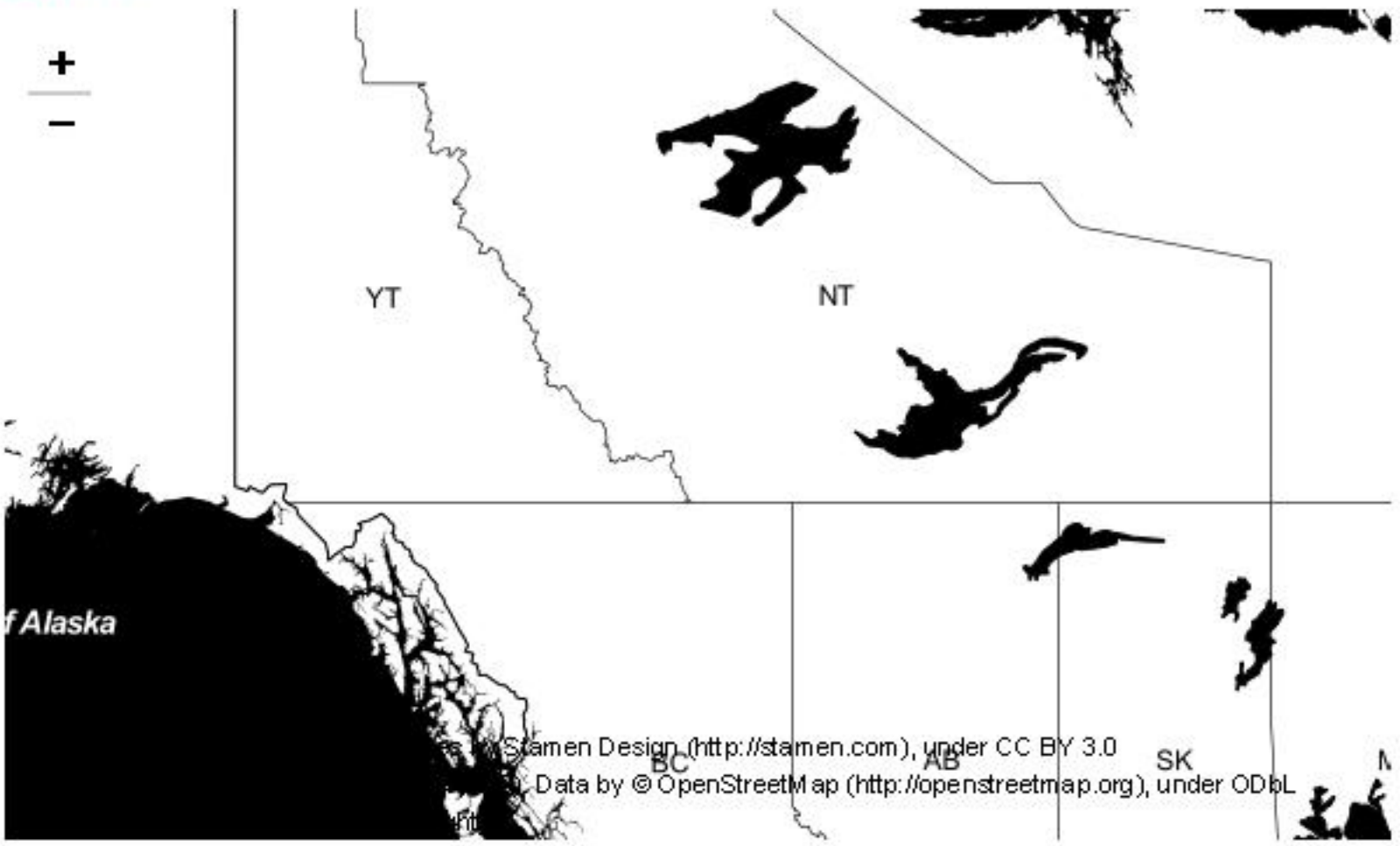
```
# Câu 2: Tạo bản đồ với center là Canada (location=[56.130, -106.35])
#      và zoom level (zoom_start=4)
canada_map = folium.Map(location=[56.130, -106.35], zoom_start=4)
canada_map
```

Out[3]:

Entrée [4]:

```
# Câu 3: Tạo Stamen Toner Map với center là Canada, và zoom level là 4
canada_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Toner')
canada_map
```

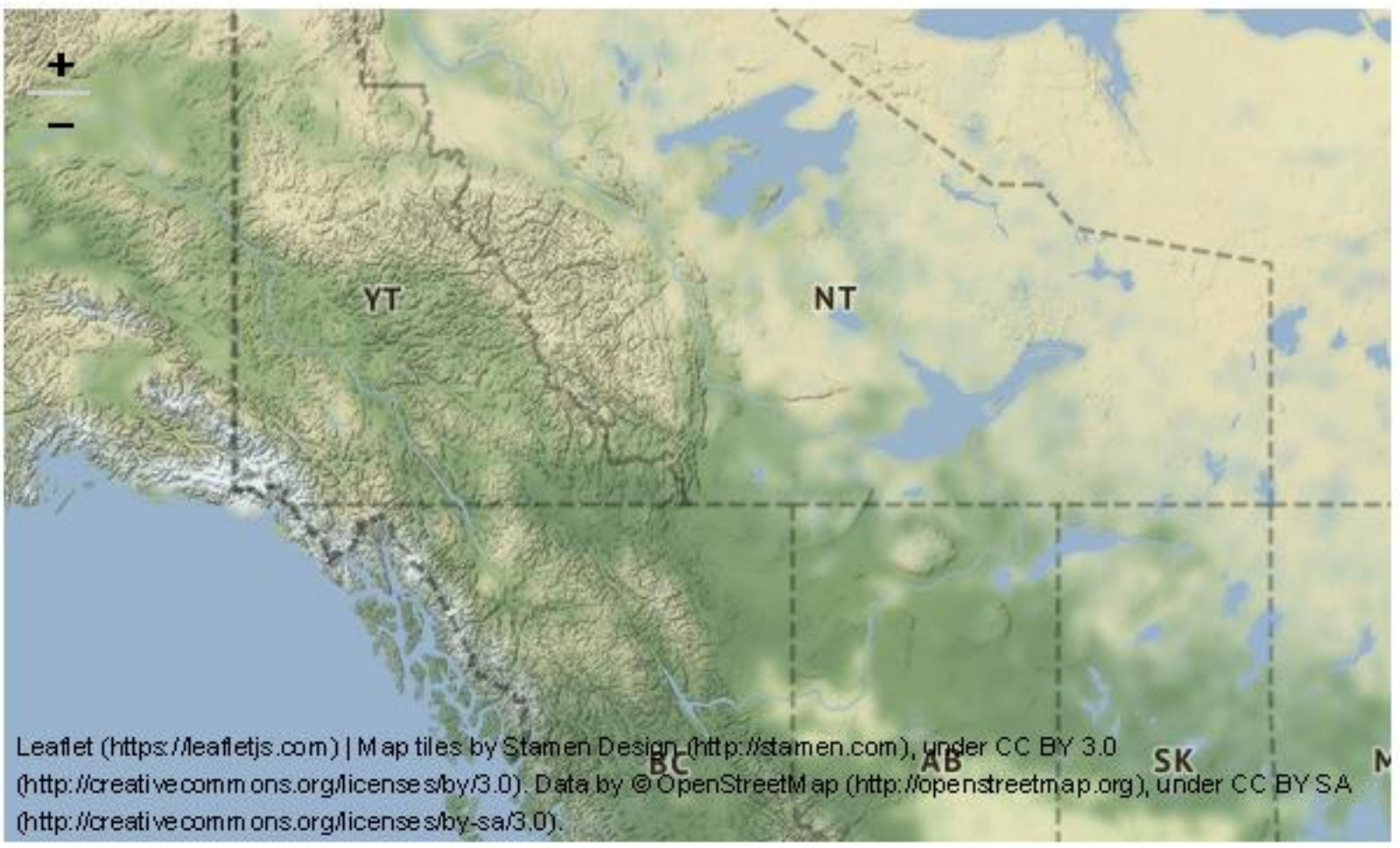
Out[4]:



Entrée [5]:

```
# Câu 4: Tạo Stamen Terrain Map với center là Canada, và zoom level là 4
canada_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Terrain')
canada_map
```

Out[5]:



Entrée [6]:

```
# Câu 5: Tạo Stamen Watercolor Map với center là Canada, và zoom level 4
world_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Watercolor')
world_map
```

Out[6]:

Choropleth Map

Entrée [7]:

```
# Câu 1: Đọc dữ liệu Canada.xlsx và lưu vào df_can,
#    tìm hiểu về dữ liệu với: describe, head, shape, columns
df_can = pd.read_excel('data\Canada.xlsx', sheet_name='Canada by Citizenship',
                       skiprows=range(20), skipfooter=2)
```

Entrée [8]:

```
# Cho biết thông tin thống kê chung của df_can
df_can.describe()
```

Out[8]:

	AREA	REG	DEV	1980	1981	1982	
count	195.000000	195.000000	195.000000	195.000000	195.000000	195.000000	195.0
mean	912.764103	1249.015385	901.753846	508.394872	566.989744	534.723077	387.4
std	13.082835	1185.526885	0.431878	1949.588546	2152.643752	1866.997511	1204.3
min	903.000000	905.000000	901.000000	0.000000	0.000000	0.000000	0.0
25%	903.000000	914.000000	902.000000	0.000000	0.000000	0.000000	0.0
50%	908.000000	922.000000	902.000000	13.000000	10.000000	11.000000	12.0
75%	922.000000	925.500000	902.000000	251.500000	295.500000	275.000000	173.0
max	935.000000	5501.000000	902.000000	22045.000000	24796.000000	20620.000000	10015.0

8 rows × 37 columns

Entrée [9]:

```
# In vài dòng dữ liệu đầu của df_can
df_can.head()
```

Out[9]:

	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	198
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	1
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	8
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	

5 rows × 43 columns

Entrée [10]:

```
# Cho biết kích thước của df_can
df_can.shape
```

Out[10]:

```
(195, 43)
```

Entrée [11]:

```
# Xem danh sách các cột của df_can
df_can.columns
```

Out[11]:

```
Index([    'Type', 'Coverage',   'OdName',     'AREA', 'AreaName',       'RE
G',
       'RegName',      'DEV',  'DevName',       1980,       1981,       198
2,
            1983,       1984,       1985,       1986,       1987,       198
8,
            1989,       1990,       1991,       1992,       1993,       199
4,
            1995,       1996,       1997,       1998,       1999,       200
0,
            2001,       2002,       2003,       2004,       2005,       200
6,
            2007,       2008,       2009,       2010,       2011,       201
2,
            2013],
      dtype='object')
```

Entrée [12]:

```
# Câu 2: Làm sạch dữ liệu:
# Bỏ đi những cột không cần thiết như 'AREA','REG','DEV','Type','Coverage'
df_can.drop(['AREA','REG','DEV','Type','Coverage'], axis=1, inplace=True)

# Đổi tên một số cột như sau:
#   'OdName' => 'Country', 'AreaName' => 'Continent', 'RegName' => 'Region'
df_can.rename(columns={'OdName':'Country', 'AreaName':'Continent','RegName':'Region'},
              inplace=True)

# Đổi tất cả các cột sang kiểu string
df_can.columns = list(map(str, df_can.columns))

# Thêm cột Total chứa tổng lượng nhập cư qua các năm
df_can['Total'] = df_can.sum(axis=1)
```

Entrée [13]:

```
# Câu 3: Xem thông tin dữ liệu lúc này:
# Hiển thị 5 dòng dữ liệu đầu của df_can sau khi làm sạch dữ liệu
df_can.head()
```

Out[13]:

	Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	...	2005
0	Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	3436
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	1223
2	Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	...	3626
3	American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	...	0
4	Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	...	0

5 rows × 39 columns

Entrée [14]:

```
# Cho biết kích thước của df_can sau khi làm sạch dữ liệu
df_can.shape
```

Out[14]:

```
(195, 39)
```

Entrée [15]:

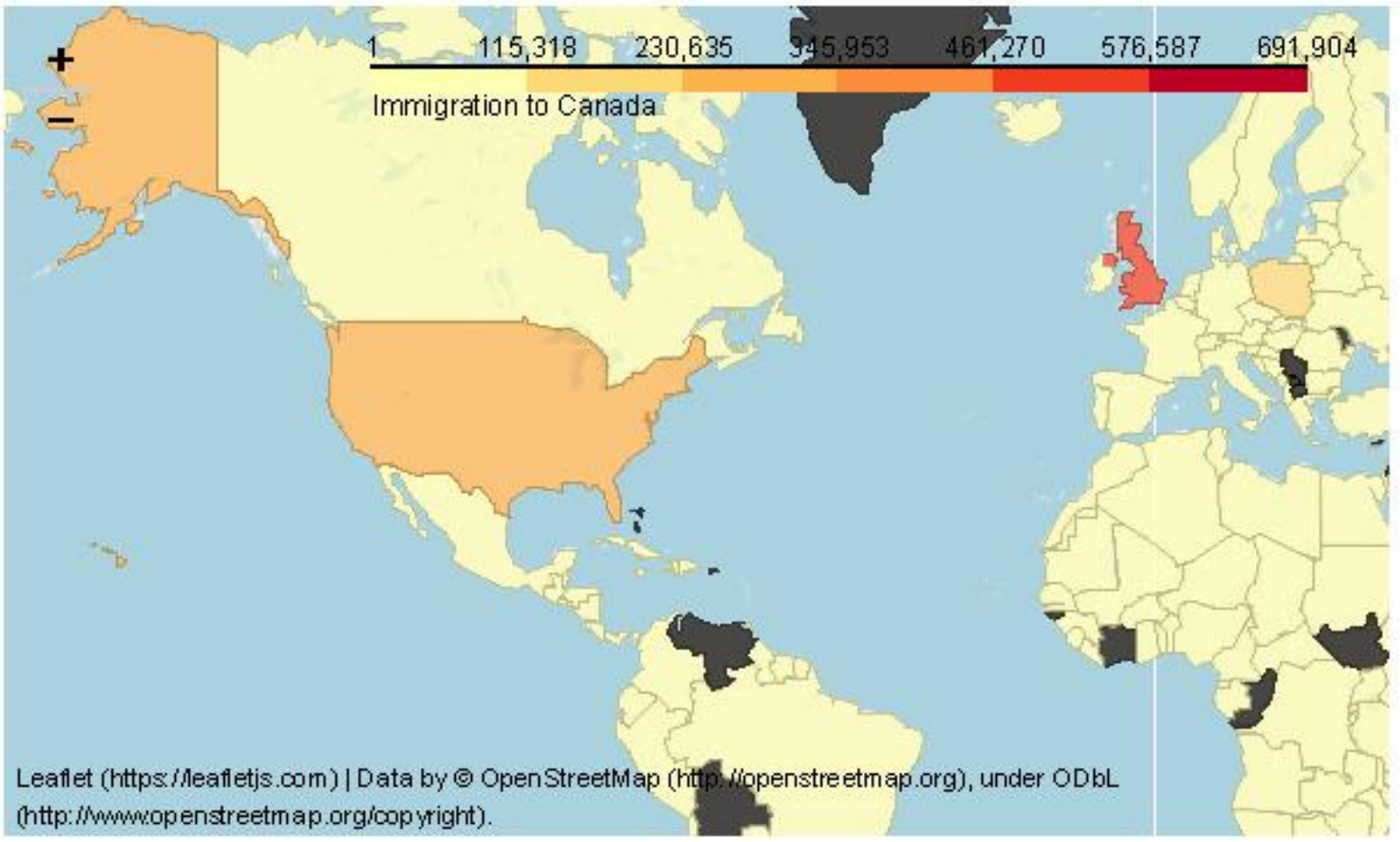
```
# Câu 4: Tạo world map, với center [0, 0] là latitude và longitude, zoom level là 2,
#        sử dụng tiles là OpenStreetMap
world_map = folium.Map(location=[0, 0], zoom_start=2, tiles='OpenStreetMap')
world_map
```

Out[15]:

Entrée [16]:

```
# Câu 5: Tạo choropleth map sử dụng total nhập cư của từng quốc gia vào Canada
#       từ năm 1980 đến năm 2013

# Lấy file GeoJSON có tên là world-countries.json
world_geo = r'data\world-countries.json'

folium.Choropleth(
    geo_data=world_geo,
    data=df_can,
    columns=['Country', 'Total'],
    key_on='feature.properties.name',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Immigration to Canada'
).add_to(world_map)

world_map
```

Out[16]:



Chapter 8 - Exercise 3: Mexico

Entrée [1]:

```
import folium
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Map: Mexico

Entrée [2]:

```
# Câu 1: Tạo biểu đồ có center là Mexico (location= [23.6345, -102.5528])
#        với zoom level là 5
mexico_map = folium.Map(location=[23.6345, -102.5528], zoom_start=5)
mexico_map
```

Out[2]:

Entrée [3]:

```
# Câu 2: Tạo OpenStreetMap với center là Mexico  with zoom level 6.3
mexico_map = folium.Map(location=[23.6345, -102.5528], zoom_start=6.3,
                        tiles='OpenStreetMap')
mexico_map
```

Out[3]:

Map với Marker

San Francisco Police Department Incidents năm 2016 - được cung cấp từ cổng thông tin dữ liệu công cộng San Francisco. Các sự cố bắt nguồn từ hệ thống báo cáo sự cố tội phạm của Sở cảnh sát San Francisco (SFPD). Được cập nhật hàng ngày, hiển thị dữ liệu cho cả năm 2016. Địa chỉ và vị trí đã được ẩn danh bằng cách di chuyển đến giữa khối (mid-block) hoặc đến một giao lộ (intersection)

Entrée [4]:

```
# Câu 1: Đọc dữ liệu Police_Department_Incidents_-Previous_Year__2016.csv
#       và lưu vào df_incidents
df_incidents = pd.read_csv('data\Police_Department_Incidents_-_Previous_Year__2016_.csv')
```

So each row consists of 13 features:

1. **IncidntNum**: Incident Number
2. **Category**: Category of crime or incident
3. **Descript**: Description of the crime or incident
4. **DayOfWeek**: The day of week on which the incident occurred
5. **Date**: The Date on which the incident occurred
6. **Time**: The time of day on which the incident occurred
7. **PdDistrict**: The police department district
8. **Resolution**: The resolution of the crime in terms whether the perpetrator was arrested or not
9. **Address**: The closest address to where the incident took place
10. **X**: The longitude value of the crime location
11. **Y**: The latitude value of the crime location
12. **Location**: A tuple of the latitude and the longitude values
13. **PdId**: The police department ID

Entrée [5]:

```
# Cho biết thông tin của df_incidents
df_incidents.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150500 entries, 0 to 150499
Data columns (total 13 columns):
IncidntNum    150500 non-null int64
Category      150500 non-null object
Descript      150500 non-null object
DayOfWeek     150500 non-null object
Date          150500 non-null object
Time          150500 non-null object
PdDistrict    150499 non-null object
Resolution    150500 non-null object
Address       150500 non-null object
X             150500 non-null float64
Y             150500 non-null float64
Location      150500 non-null object
PdId          150500 non-null int64
dtypes: float64(2), int64(2), object(9)
memory usage: 14.9+ MB
```

Entrée [6]:

```
# Hiển thị 5 dòng dữ liệu đầu của df_incidents
df_incidents.head()
```

Out[6]:

	IncidntNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolu
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	ARR BOO
1	120058272	WEAPON LAWS	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	ARR BOO
2	141059263	WARRANTS	WARRANT ARREST	Monday	04/25/2016 12:00:00 AM	14:59	BAYVIEW	ARR BOO
3	160013662	NON-CRIMINAL	LOST PROPERTY	Tuesday	01/05/2016 12:00:00 AM	23:50	TENDERLOIN	N(
4	160002740	NON-CRIMINAL	LOST PROPERTY	Friday	01/01/2016 12:00:00 AM	00:30	MISSION	N(

Entrée [7]:

```
# Cho biết kích thước của df_incidents
df_incidents.shape
```

Out[7]:

```
(150500, 13)
```

Entrée [8]:

```
# Câu 2: Rút trích dữ liệu: có đến 150.500 tội phạm, diễn ra vào năm 2016.
# Tạo bộ dữ liệu mới chỉ lấy 100 tội phạm đầu tiên trong bộ dữ liệu df_incidents
limit = 100
df_incidents = df_incidents.iloc[0:limit, :]
# Cho biết kích thước của df_incidents
df_incidents.shape
```

Out[8]:

```
(100, 13)
```

Entrée [9]:

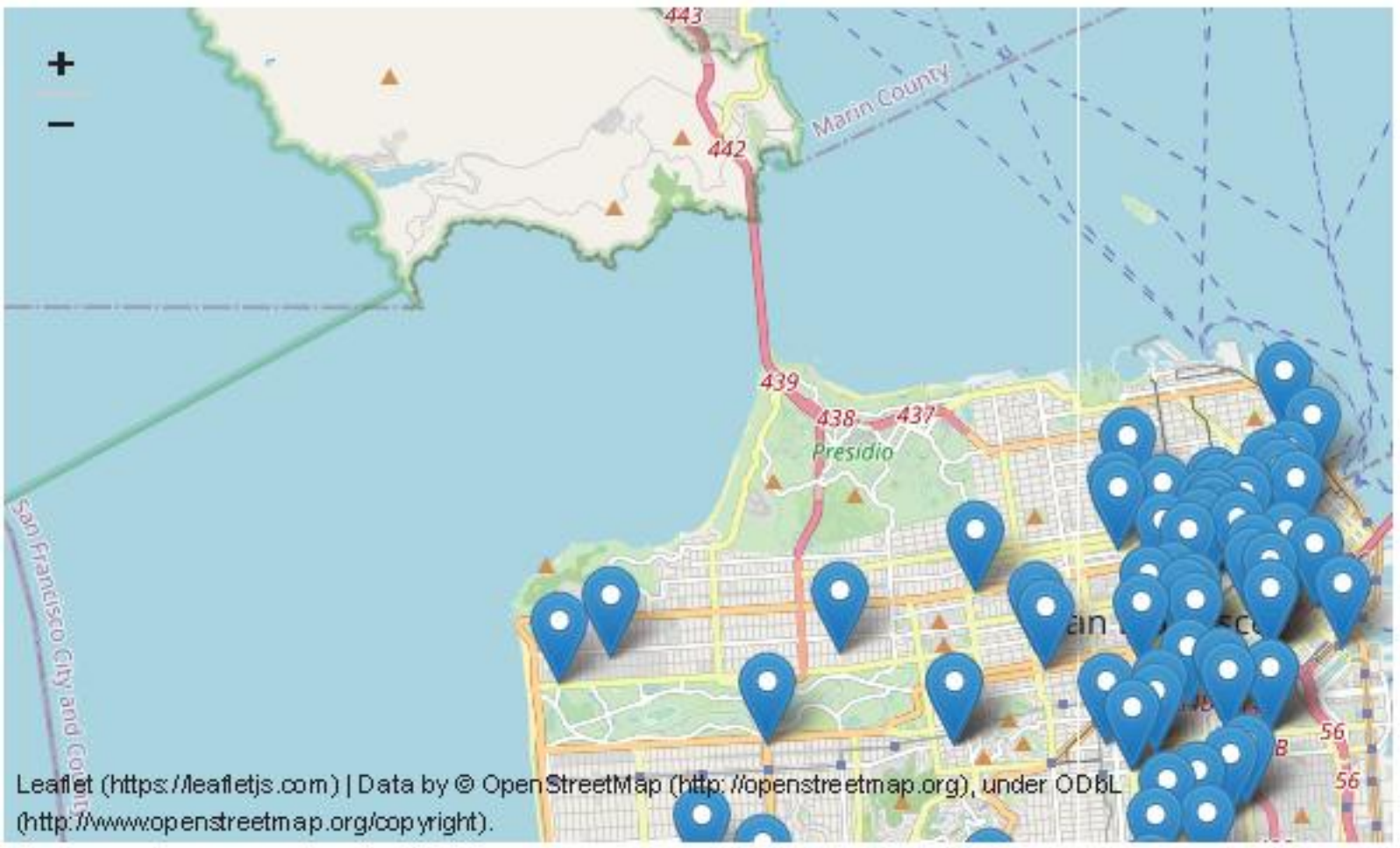
```
# Câu 3: Tạo biểu đồ có center là San Francisco (location= [37.77, -122.42])
#       với zoom level là 12
sanfran_map = folium.Map(location=[37.77, -122.42], zoom_start=12)
sanfran_map
```

Out[9]:

Entrée [10]:

```
# Câu 4: Đưa 100 điểm tội phạm lên bản đồ (với marker)
incidents = folium.map.FeatureGroup()

for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.features.Marker(
            [lat, lng],
            radius=5,
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

sanfran_map.add_child(incidents)
```

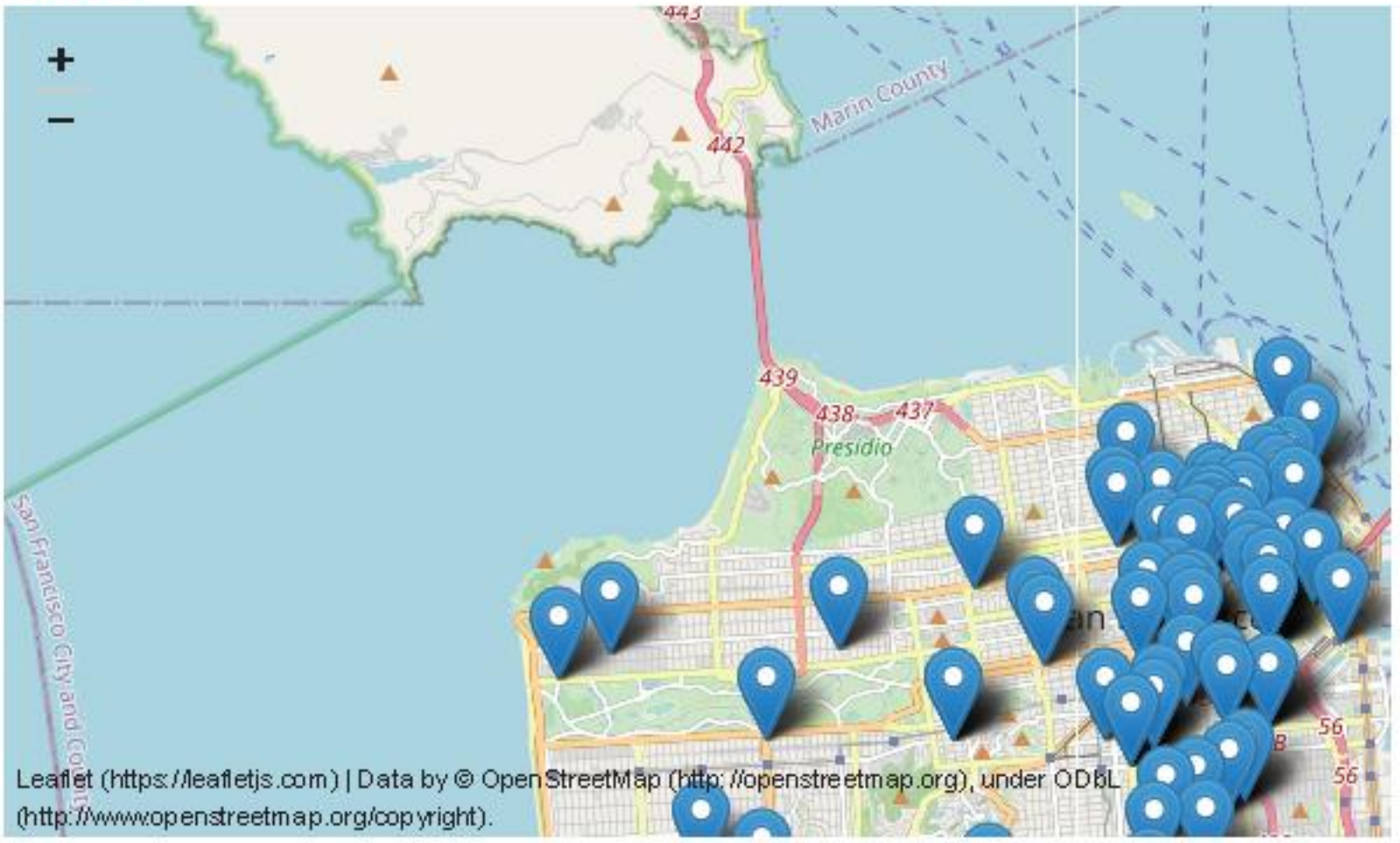
Out[10]:



Entrée [11]:

```
# Câu 5: Thêm pop-up text sẽ được hiển thị thông tin Category
#      khi người dùng di chuyển chuột qua chuột qua marker
incidents = folium.map.FeatureGroup()

for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.features.Marker(
            [lat, lng],
            radius=5,
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

latitudes = list(df_incidents.Y)
longitudes = list(df_incidents.X)
labels = list(df_incidents.Category)

for lat, lng, label in zip(latitudes, longitudes, labels):
    folium.Marker([lat, lng], popup=label).add_to(sanfran_map)

sanfran_map.add_child(incidents)
```

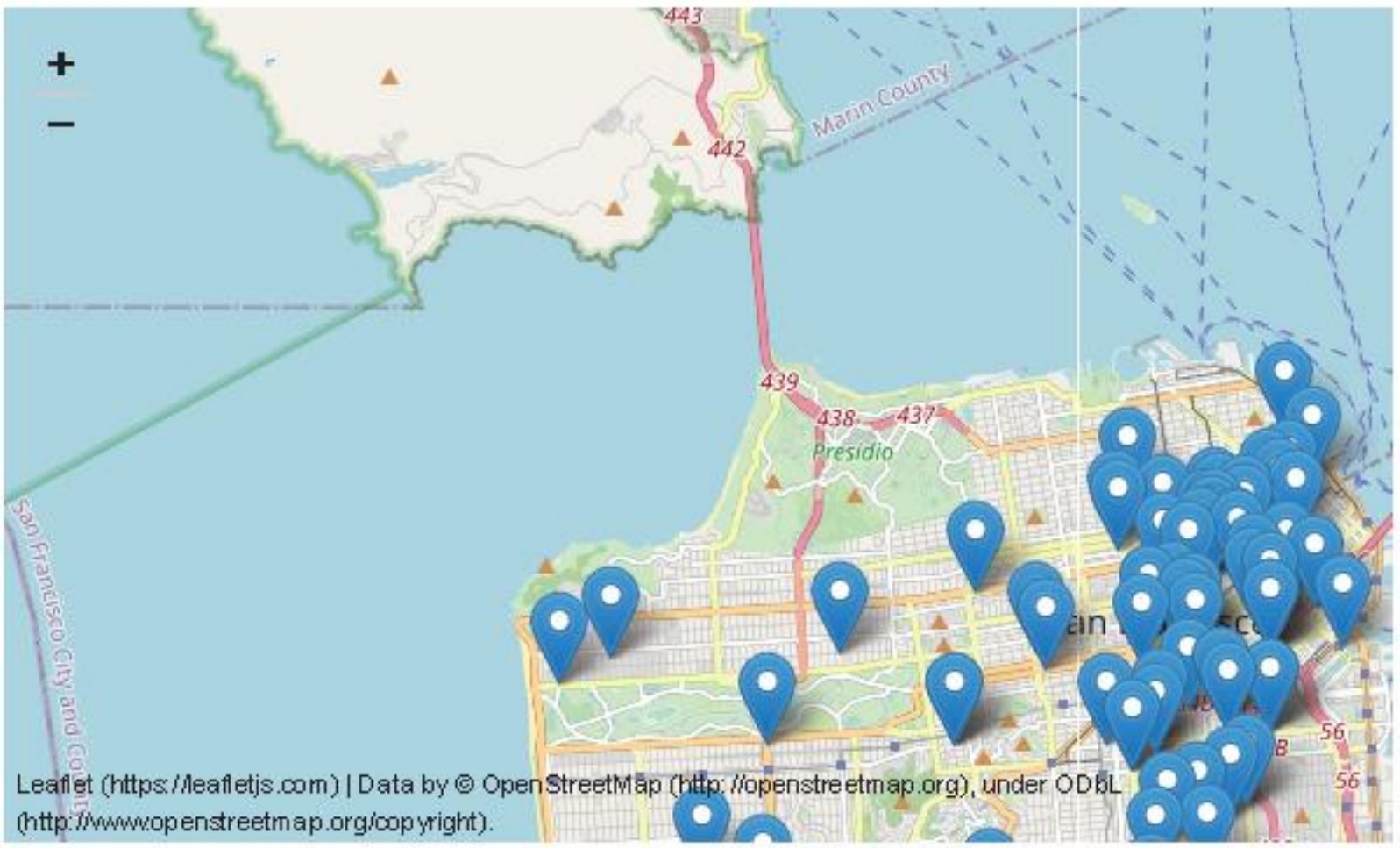
Out[11]:



Entrée [12]:

```
# Câu 6: Để cho bản đồ khỏi rối, bỏ đi các location marker,
#       và chỉ thêm text vào từng circle marker
sanfran_map = folium.Map(location=[37.77, -122.42], zoom_start=12)

for lat, lng, label in zip(df_incidents.Y, df_incidents.X, df_incidents.Category):
    folium.features.Marker(
        [lat, lng],
        radius=5,
        color='yellow',
        fill=True,
        popup=label,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(sanfran_map)

sanfran_map
```

Out[12]:



Entrée [13]:

```
# Câu 7: Nhóm các markers vào các cluster.
#      Mỗi cluster sẽ hiển thị số lượng các tội phạm trong mỗi neighborhood.
# Gợi ý: Sử dụng MarkerCluster object và thêm tất cả các data point trong dataframe
#        vào object này

from folium import plugins

sanfran_map = folium.Map(location = [37.77, -122.42], zoom_start = 12)

incidents = plugins.MarkerCluster().add_to(sanfran_map)

for lat, lng, label, in zip(df_incidents.Y, df_incidents.X, df_incidents.Category):
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(incidents)

sanfran_map
```

Out[13]: