Lab 03: *Trực quan hóa không gian địa lý*

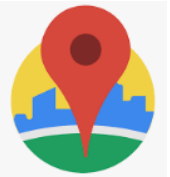
Nội dung

1.Giới thiệu

2.2 layer maps - Geopandas

3.Folium

4.Choropleth Map



Giới thiệu

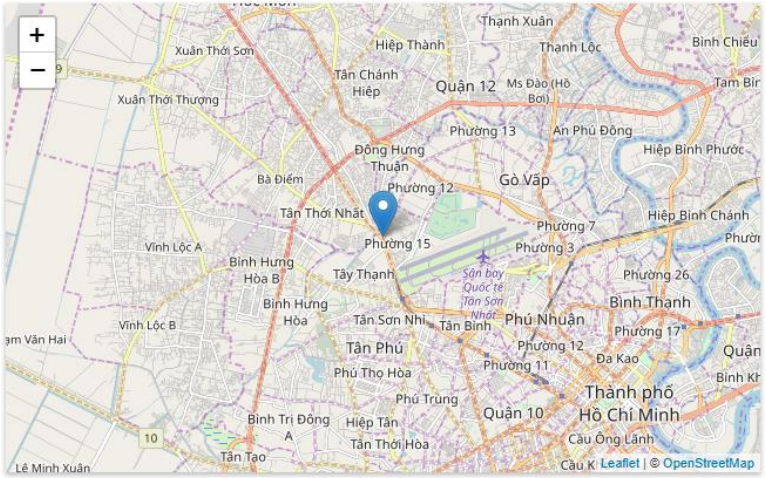
- Một trong những nhiệm vụ quan trọng nhất data scientist là tìm hiểu mối quan hệ giữa vị trí vật lý của dữ liệu và bối cảnh địa lý của chúng với các công việc, ví dụ như sử dụng dữ liệu không gian, liên kết dữ liệu với ngữ cảnh, phủ dữ liệu không gian địa lý vào bản đồ kèm tín hiệu không gian...
- GeoPandas là một package giúp tạo ra các hình ảnh trực quan hấp dẫn của dữ liệu không gian địa lý.

Giới thiệu

- Longitude (kinh độ) và latitude (vĩ độ)
  - Kinh độ và Vĩ độ là các đơn vị đại diện cho tọa độ trên hệ tọa độ địa lý.

  - Giống như mọi ngôi nhà thực tế đều có địa chỉ (bao gồm số, tên đường, thành phố, v.v.), mọi điểm trên bề mặt trái đất có thể được chỉ định bởi tọa độ kinh độ và vĩ độ. Do đó, bằng cách sử dụng vĩ độ và kinh độ, chúng ta có thể chỉ định hầu như bất kỳ điểm nào trên trái đất.

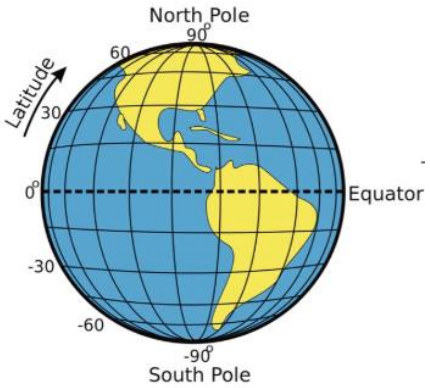
Tp. Hồ Chí Minh

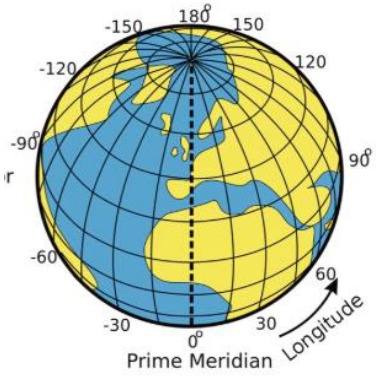


Lat	Long
(10.823099, 106.629662)	

Giới thiệu

- Longitude (kinh độ) và latitude (vĩ độ)
  - Latitude (ϕ), hiển thị góc giữa đường thẳng trong điểm nhất định và mặt phẳng xích đạo (equatorial plane). Vĩ độ được chỉ định theo độ, bắt đầu từ 0 ° và kết thúc bằng 90 ° cho cả hai phía của đường xích đạo, tạo ra vĩ độ Bắc và Nam. Đường xích đạo là đường có vĩ độ 0 °.
  - Longitude (λ) là một tọa độ góc khác xác định vị trí của một điểm trên bề mặt trái đất. Kinh độ được định nghĩa là một góc chỉ về hướng tây hoặc đông từ Kinh tuyến Greenwich, được lấy là Kinh tuyến gốc. Kinh độ có thể được xác định tối đa là 180 ° về về phía tây từ Kinh tuyến gốc Prime Meridian

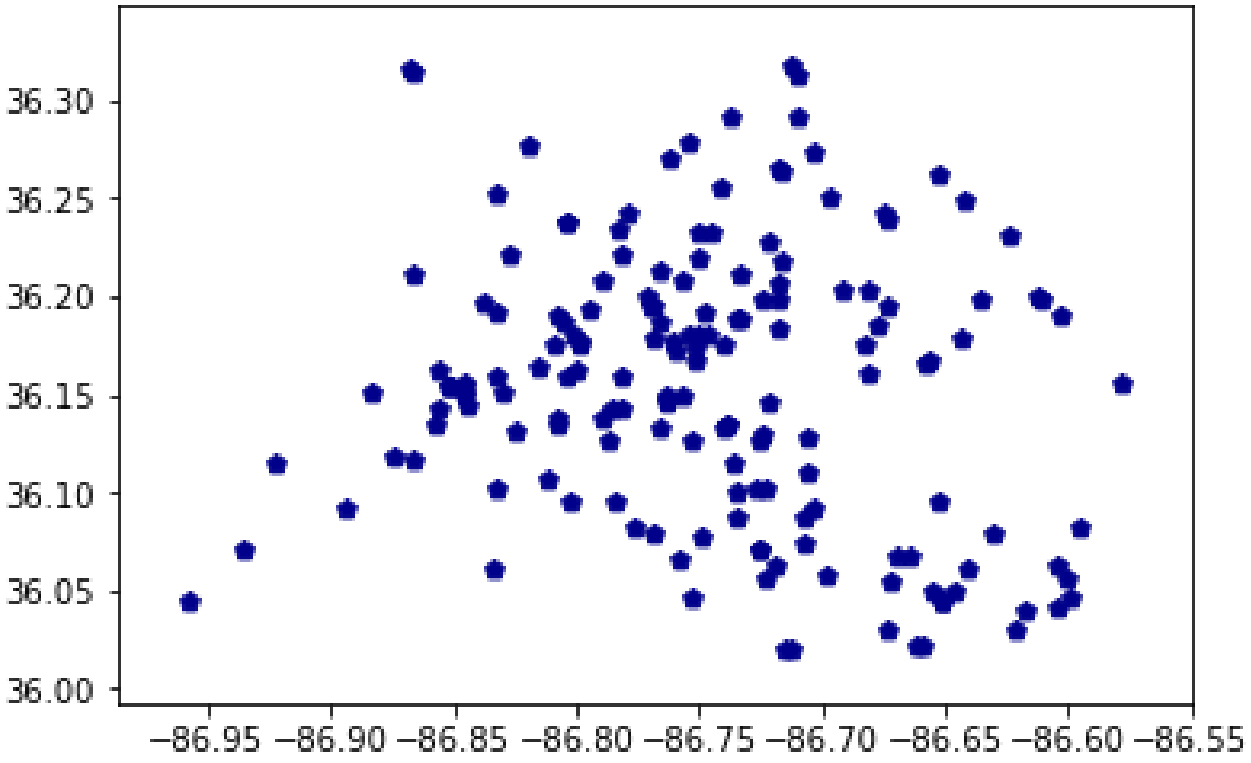




Nội dung

2-layer maps - Geopandas

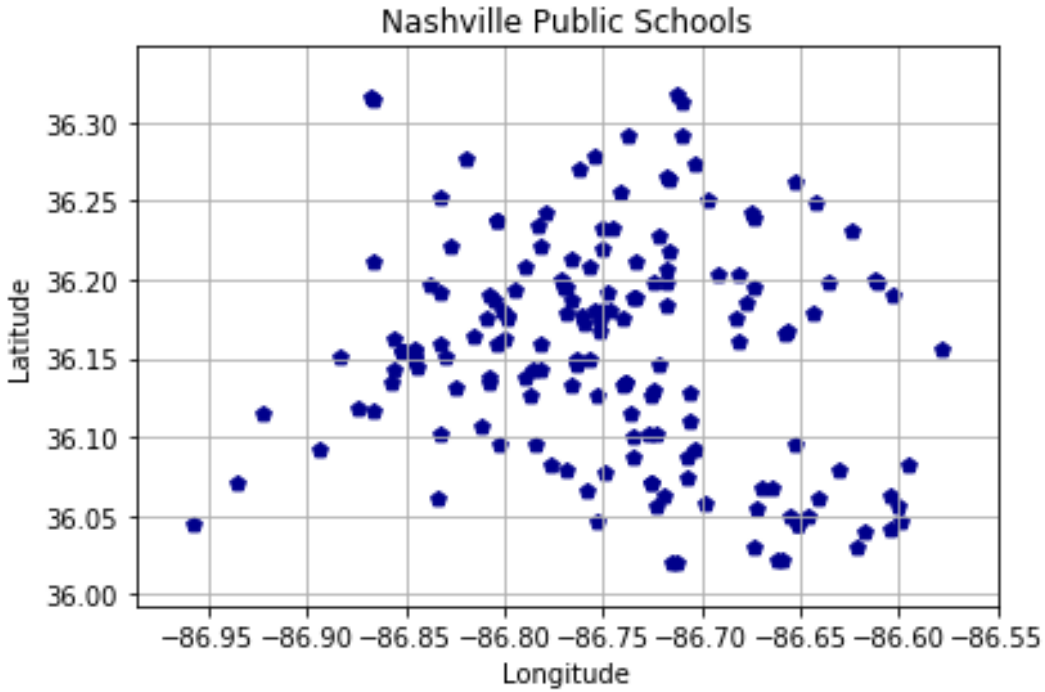
- ```
  plt.scatter(schools.Longitude,
  schools.Latitude,    c = 'darkblue',  marker = 'p')
  plt.show()
  ```


  ```

2-layer maps - Geopandas

- Scat

```
plt.scatter(schools.Longitude,
schools.Latitude,    c = 'darkblue',  marker = 'p')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Nashville Public Schools')
plt.grid()
plt.show()
```



2-layer maps - Geopandas

```
bus_stops.head(2)
```

lng, lat từ chuỗi

	Title	Last Name	First Name	Location	Medium	Type	Description	Latitude	Longitude	Mapped Location
0	[Cross Country Runners]	Frost	Miley	4001 Harding Rd., Nashville TN	Bronze	Sculpture	NaN	36.12856	-86.83660	(36.12856, -86.8366)
1	[Fourth and Commerce Sculpture]	Walker	Lin	333 Commerce Street, Nashville TN	NaN	Sculpture	NaN	36.16234	-86.77774	(36.16234, -86.77774)

```
location = [x[1:-1].split(", ") for x in bus_stops['Mapped Location']]
location
```

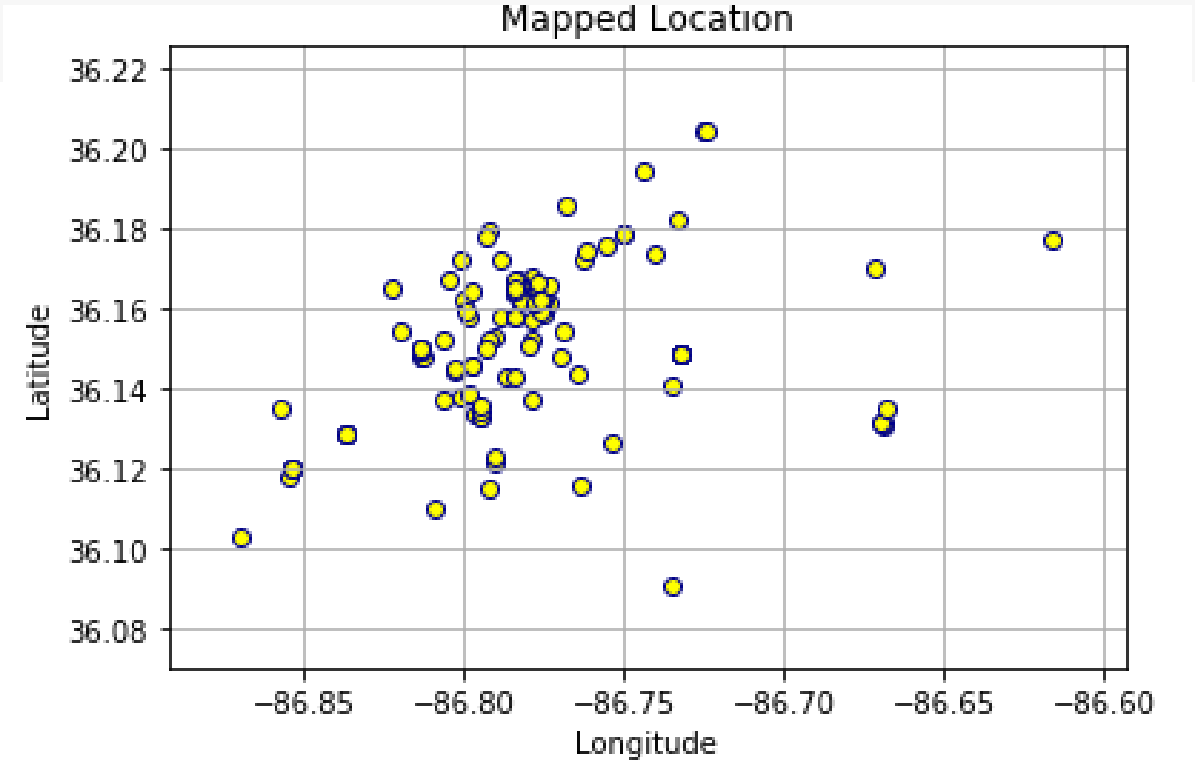
```
[['36.12856', '-86.8366'],
 ['36.16234', '-86.77774'],
 ['36.1579', '-86.78817'],
```

```
bus_stops['lat'] = [float(loc[0]) for loc in location]
bus_stops['lng'] = [float(loc[1]) for loc in location]
bus_stops.head(2)
```

	Title	Last Name	First Name	Location	Medium	Type	Description	Latitude	Longitude	Mapped Location	lat	lng
0	[Cross Country Runners]	Frost	Miley	4001 Harding Rd., Nashville TN	Bronze	Sculpture	NaN	36.12856	-86.83660	(36.12856, -86.8366)	36.12856	-86.83660
1	[Fourth and Commerce Sculpture]	Walker	Lin	333 Commerce Street, Nashville TN	NaN	Sculpture	NaN	36.16234	-86.77774	(36.16234, -86.77774)	36.16234	-86.77774

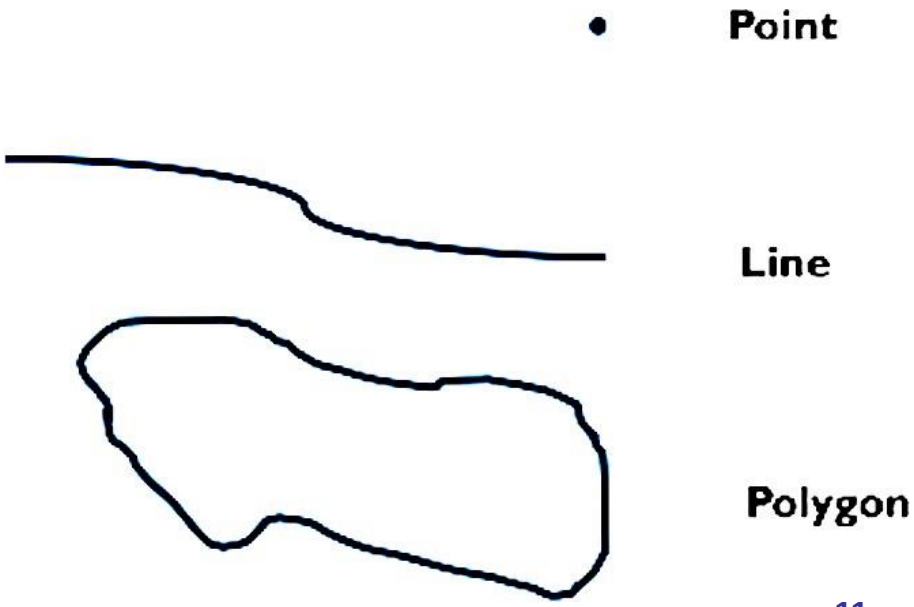
2-layer maps - Geopandas

```
# Scatterplot 2 - yellow markers with darkblue borders
plt.scatter(bus_stops.lng, bus_stops.lat, c = 'yellow', edgecolor = 'darkblue')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Mapped Location')
plt.grid()
plt.show()
```



2-layer maps - Geopandas

- Geometry và shapefile
  - Shapefile, còn gọi là Esri shapefile, là một loại tập tin được định dạng để lưu trữ dữ liệu vector địa lý, geometry



2-layer maps - Geopandas

- Geometry và shapefile
  - Các thành phần trong thư mục chứa shapefile:
    - xxx.shp (chứa geometry)
    - xxx.dbf (chứa các attribute của từng geometry)
    - xxx.shx (liên kết các attribute tới geometry)

2-layer maps - Geopandas

- Geopandas
  - Đọc shapefile vào GeoDataFrame

```
import geopandas as gpd
```

```
geo_df = gpd.read_file('My_Map_Files/stations.shp')
geo_df.head()
```

	name	marker-col	marker-sym	line	geometry
0	Van Dorn Street	#0000ff	rail-metro	blue	POINT (-77.12911152370515 38.79930767201779)
1	Franconia-Springfield	#0000ff	rail-metro	blue	POINT (-77.16797018042666 38.76652189268992)
2	Federal Center SW	#0000ff	rail-metro	blue	POINT (-77.01586821694521 38.88507235514479)
3	Judiciary Sq	#ff0000	rail-metro	red	POINT (-77.01663895662587 38.89609031766317)
4	Capitol South	#0000ff	rail-metro	blue	POINT (-77.00513941992737 38.885062500925)

2-layer maps - Geopandas

```
# Read in the services district shapefile and look at the first few rows.
service_district = gpd.read_file('data/school_districts.geojson.txt')
service_district.head()
```

```
# Print the contents of the service districts geometry in the first row
print(service_district.loc[0, 'geometry'])
```

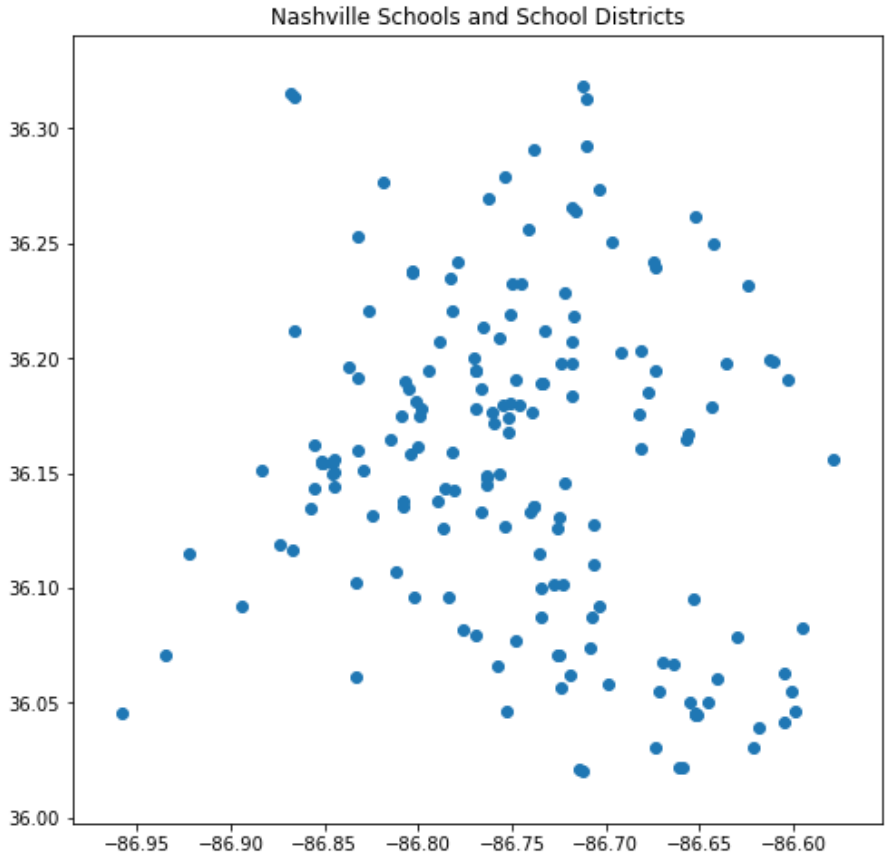
```
service_district.loc[0, 'geometry']
```

2-layer maps - Geopandas

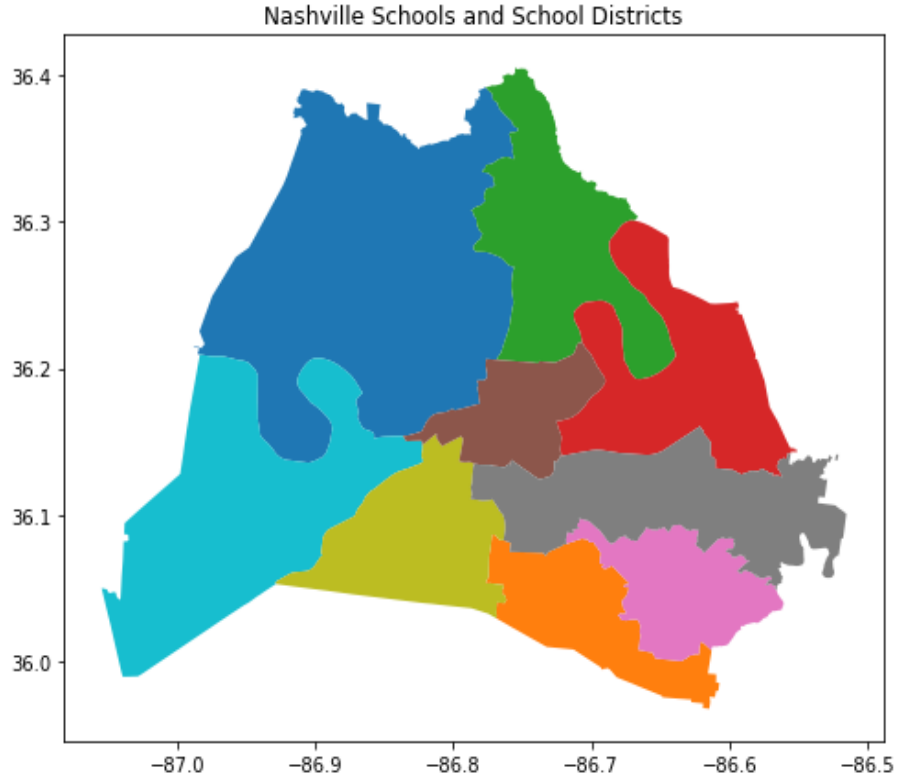
- Geometry và shapefile
  - Geometry field chứa series các cặp latitude/longitude giúp xác định biên của một polygon. In một geometry field sẽ hiển thị tất cả các cặp latitude/longitude pair tạo nên polygon boundary.

2-layer maps - Geopandas

```
plt.figure(figsize=(8,8))
plt.scatter(schools.Longitude, schools.Latitude)
plt.title('Nashville Schools and School Districts')
plt.show()
```

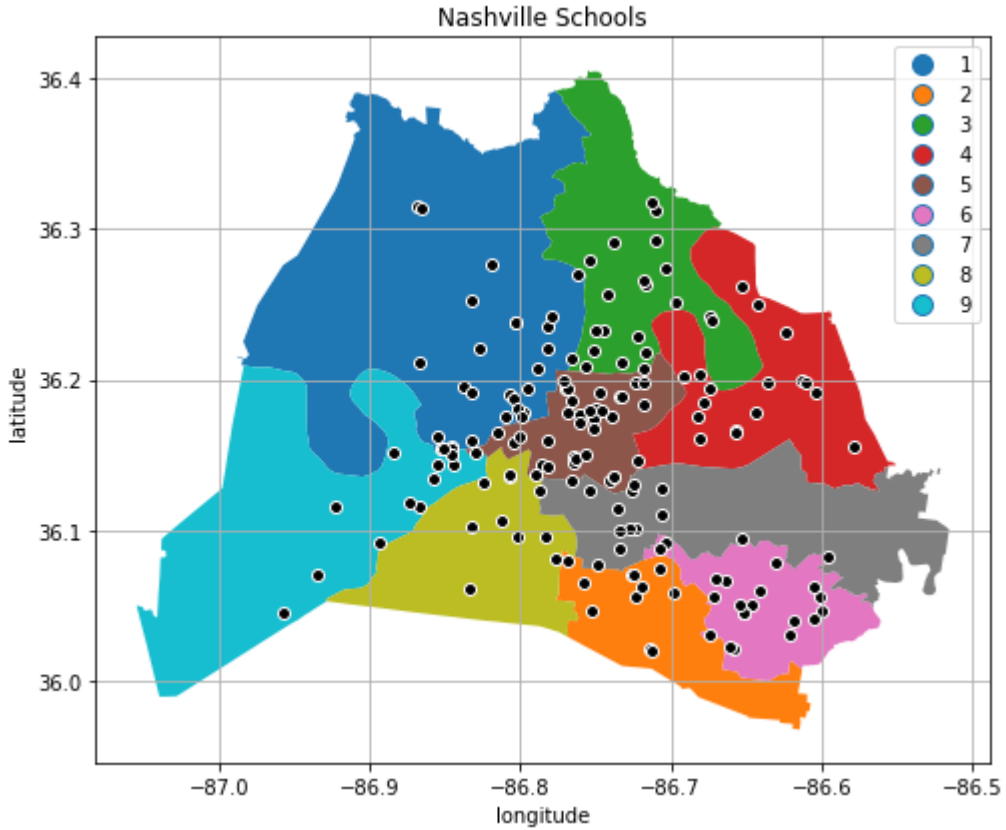



```
service_district.plot(column = 'district', figsize=(8,8))
plt.title('Nashville Schools and School Districts')
plt.show()
```



2-layer maps - Geopandas

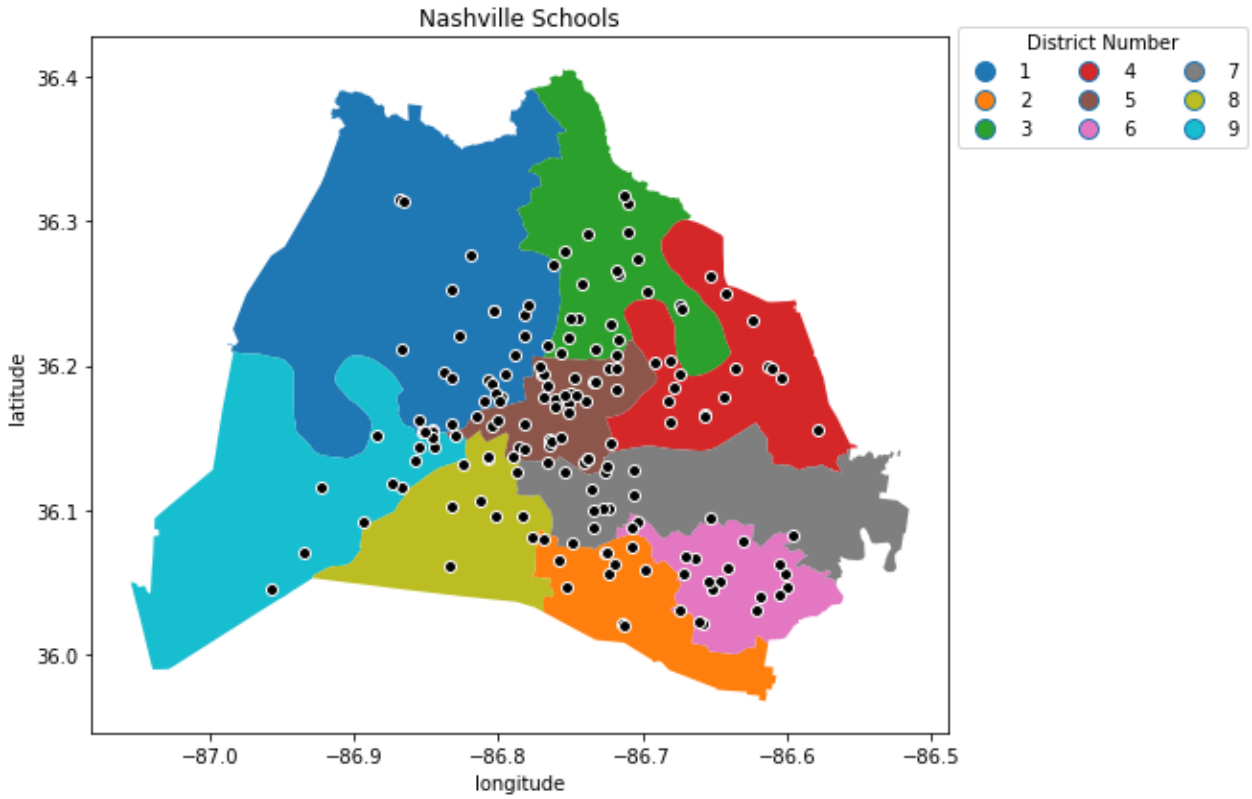
```
# Plot the service district shapefile
service_district.plot(column='district', legend=True, figsize=(8,8))
# Add the school locations
plt.scatter(schools.Longitude, schools.Latitude, c='black', edgecolor = 'white')
# Add labels and title
plt.title('Nashville Schools')
plt.xlabel('longitude')
plt.ylabel('latitude')
# Add grid lines and show the plot
plt.grid()
plt.show()
```



2-layer maps - Geopandas

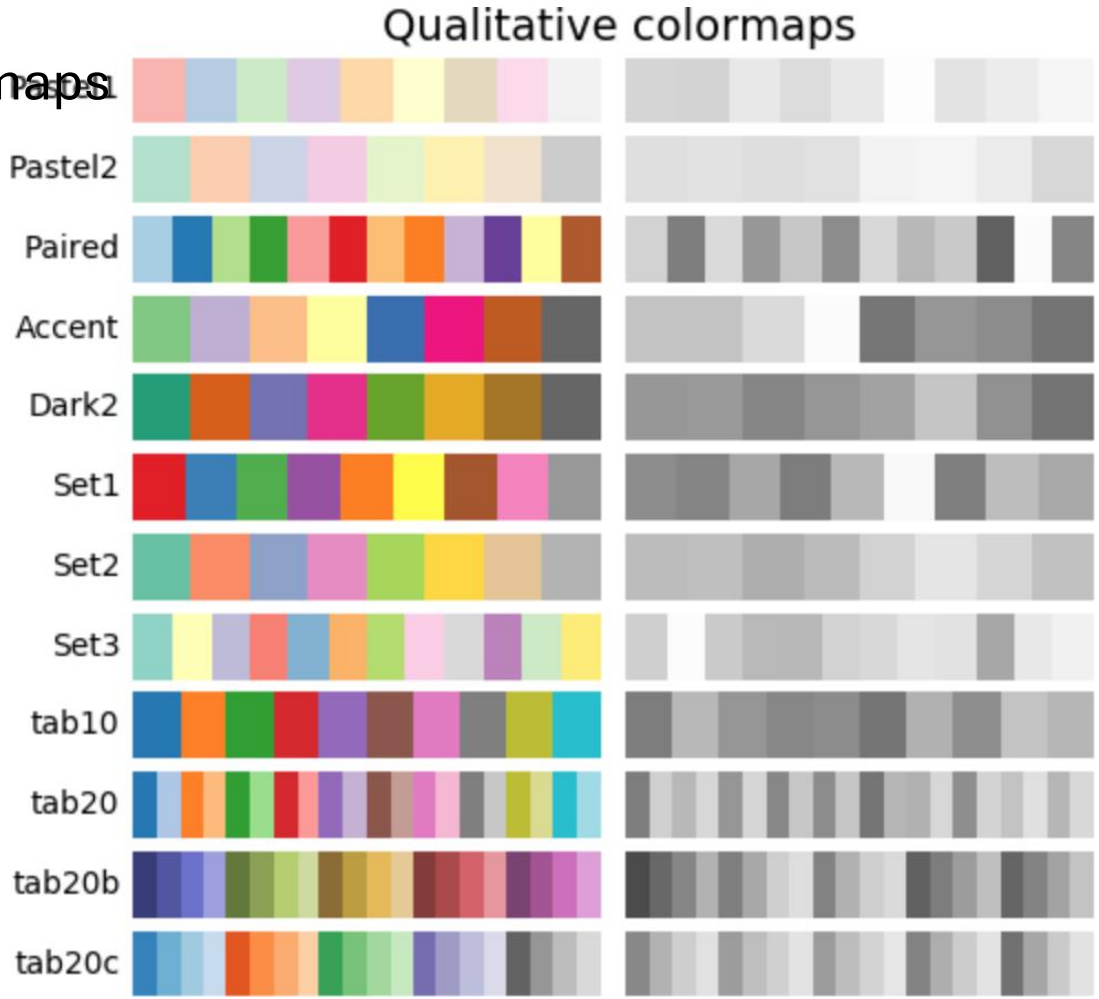
- Scatterplot và polygon

```
leg_kwds={'title':'District Number',  'loc': 'upper left', 'bbox_to_anchor':(1, 1.03),  'ncol':3}
service_district.plot(column='district', legend=True, figsize=(8,8), legend_kwds=leg_kwds)
plt.scatter(schools.Longitude, schools.Latitude, c='black', edgecolor = 'white')
plt.title('Nashville Schools')
plt.xlabel('longitude')
plt.ylabel('latitude')
plt.show()
```



2-layer maps - Geopandas

- Colormaps

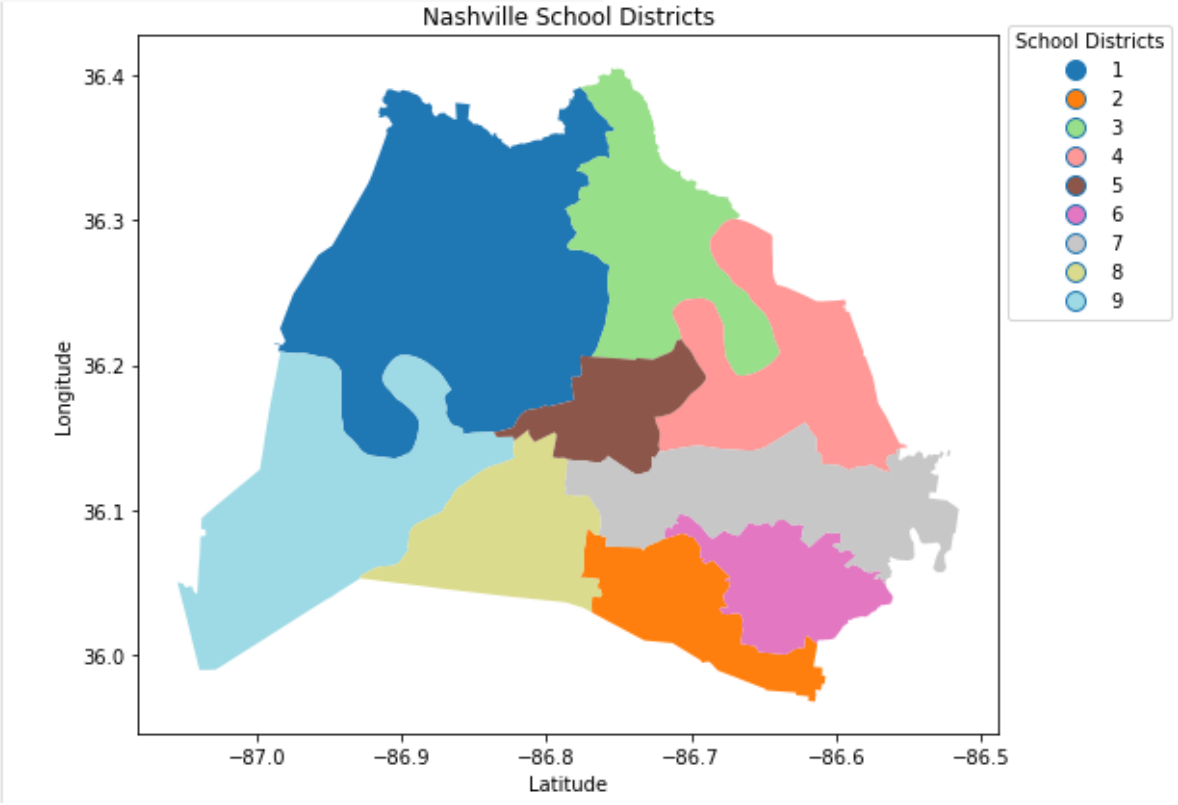


https://matplotlib.org/users/colormaps.html

2-layer maps - Geopandas

```
# Set legend style
lgnd_kwds = {'title': 'School Districts',
             'loc': 'upper left', 'bbox_to_anchor': (1, 1.03), 'ncol': 1}

# Plot the school districts using the tab20 colormap (qualitative)
service_district.plot(column = 'district', cmap = 'tab20', legend = True,
                      legend_kwds = lgnd_kwds, figsize=(8,8))
plt.xlabel('Latitude')
plt.ylabel('Longitude')
plt.title('Nashville School Districts')
plt.show()
```



2-layer maps - Geopandas

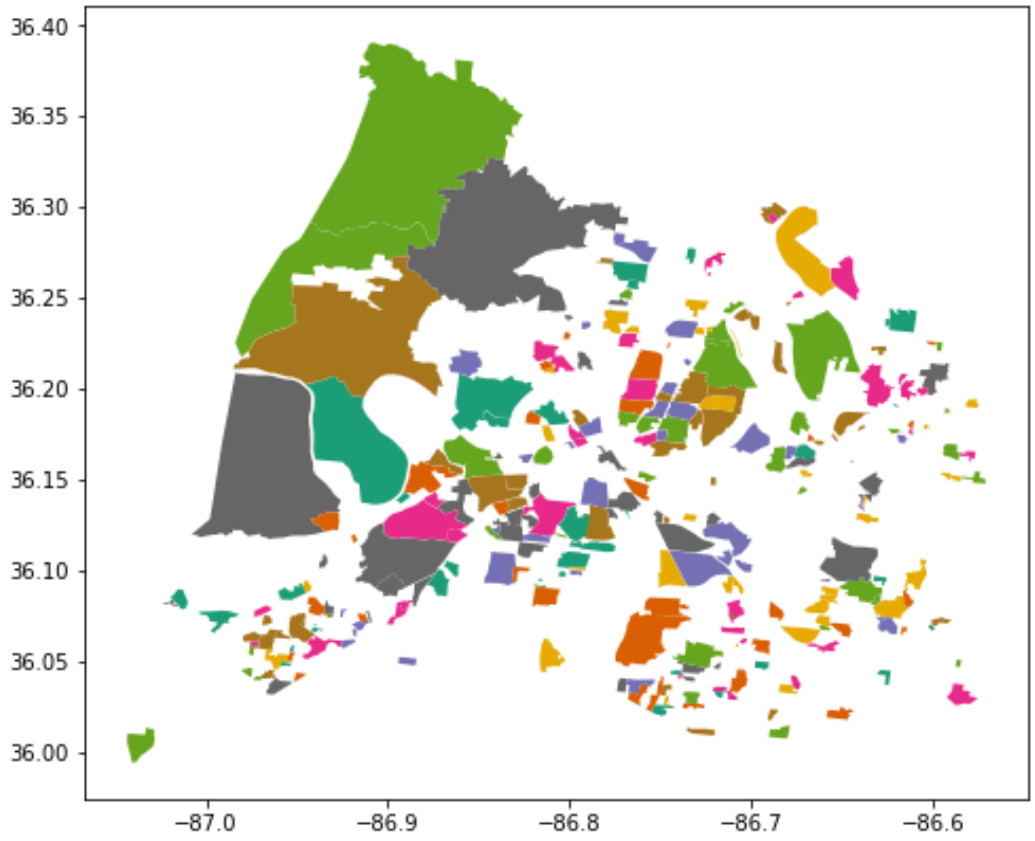
```
# Read in the neighborhoods geojson file
neighborhoods = gpd.read_file("data/neighborhoods.geojson.txt")

# Print the first few rows of neighborhoods
print(neighborhoods.head())

# Plot the neighborhoods, color according to name and use the Dark2 colormap
neighborhoods.plot(column = 'name', cmap = 'Dark2', figsize = (8,8))

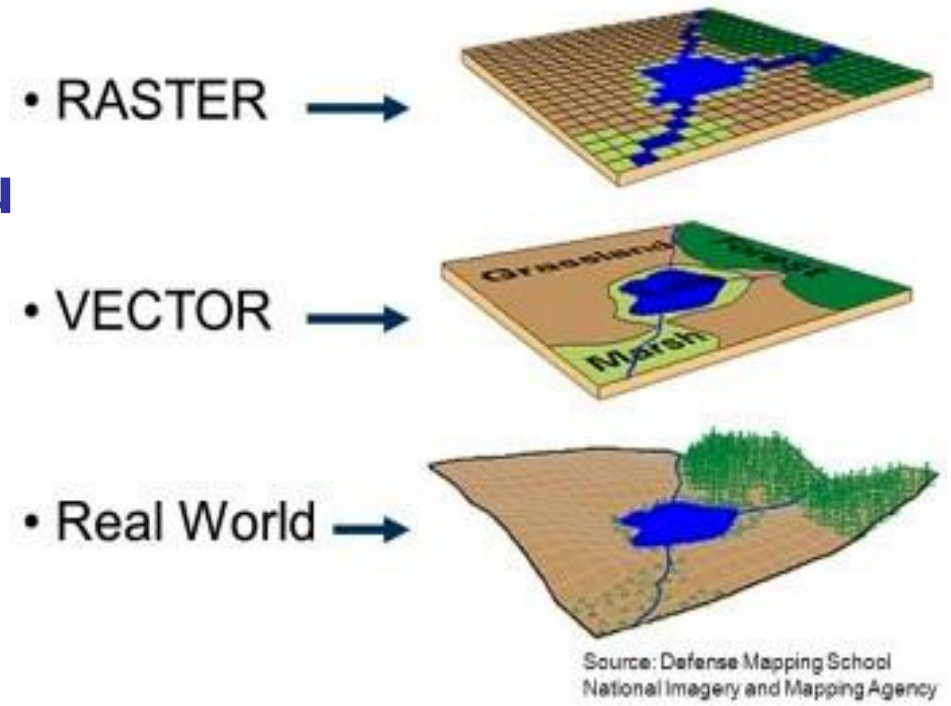
# Show the plot.
plt.show()
```

```
                    name
0  Historic Buena Vista  (POLYGON ((-86.79511056
1         Charlotte Park  (POLYGON ((-86.87459668
2               Hillwood  (POLYGON ((-86.87613708
3             West Meade  (POLYGON ((-86.90383803
4           White Bridge  (POLYGON ((-86.86321427
```



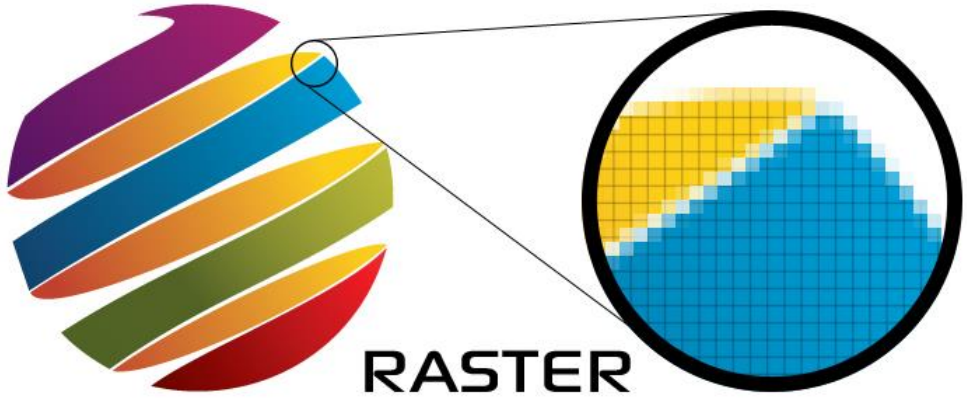
2-layer maps - Geopandas

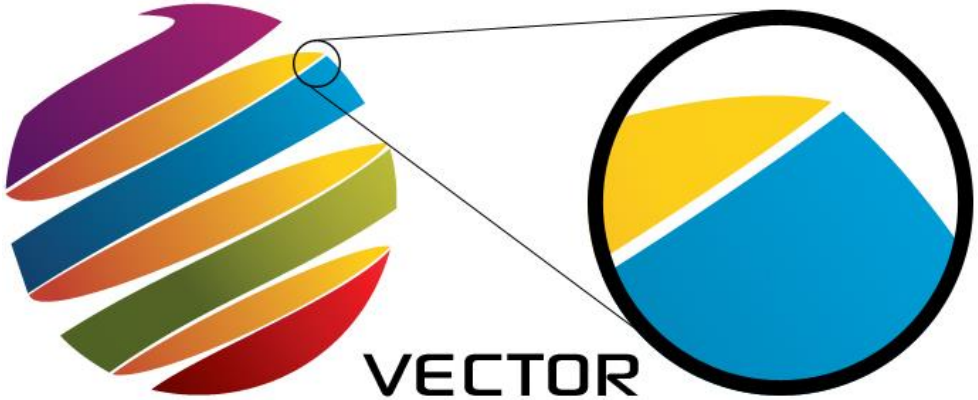
- Các thư viện hỗ trợ Geopandas
  - **Fiona: cung cấp python API cho OGR**
  - **GDAL/OGR:**
    - **GDAL dịch dữ liệu raster data**
    - **OGR dịch dữ liệu vector data**

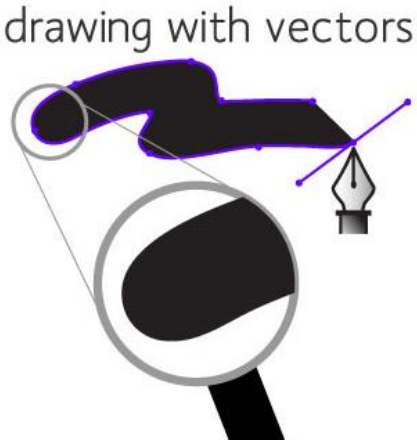


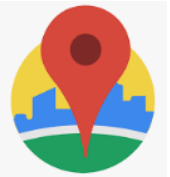
2-layer maps - Geopandas

- raster vs vector graphics







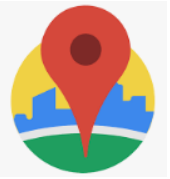


2-layer maps - Geopandas

- Đồ họa máy tính có thể được tạo ra dưới dạng hình ảnh raster hoặc vector. Raster graphic là bitmap. Một bitmap là một lưới các pixel riêng lẻ tổng hợp một hình ảnh. Raster graphic hiển thị hình ảnh là tập hợp vô số hình vuông nhỏ. Mỗi hình vuông, hoặc pixel, được mã hóa trong một màu sắc cụ thể. Những pixel đơn lẻ này là vô giá trị, nhưng khi kết hợp với nhau chúng đáng giá ngàn lời nói.
- Raster graphics sử dụng tốt nhất cho non-line art image; đặc biệt là hình ảng số hóa, bản scan của tác phẩm nghệ thuật. Non-line art image được thể hiện tốt nhất ở dạng raster bởi vì chúng thường chứa các màu sắc tinh tế, các đường và hình dạng không xác định và bố cụ phức tạp

2-layer maps - Geopandas

- Vector graphics dựa trên các công thức toán học xác định các hình học như đa giác, đường thẳng, đường cong, hình tròn và hình chữ nhật. Bởi vì vector graphic bao gồm các hình học thực sự, chúng được sử dụng tốt nhất để thể hiện các hình ảnh có cấu trúc, như line art graphic với màu sắc đồng nhất. Hầu hết các hình ảnh được tạo ra (không phải là hình ảnh tự nhiên) đáp ứng các thông số kỹ thuật này, bao gồm logo, tiêu đề thư và phông chữ.

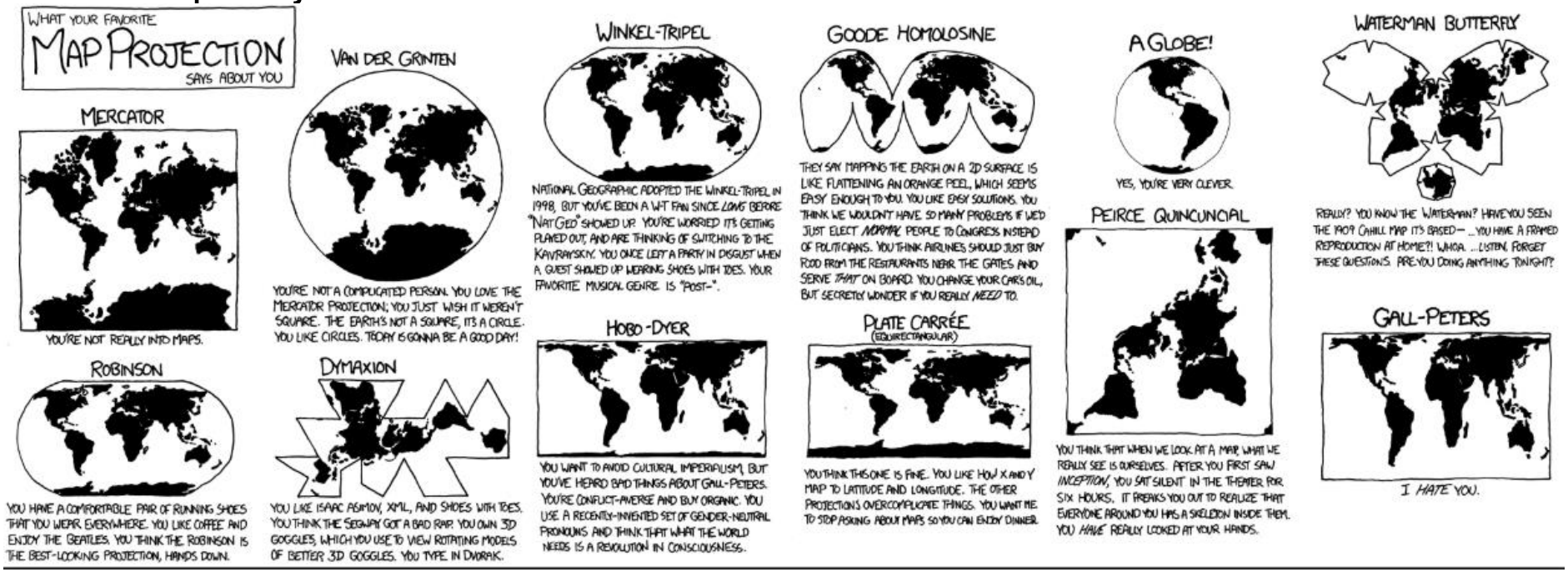


2-layer maps - Geopandas

- Về cơ bản, vector-based graphic dễ uốn hơn raster image - do đó, chúng linh hoạt và dễ sử dụng hơn nhiều.
- Ưu điểm rõ ràng nhất của vector image so với raster graphic là vector image có khả năng mở rộng nhanh chóng. Không có giới hạn trên hoặc dưới cho kích thước vector image. Giống như các quy tắc toán học áp dụng giống hệt nhau cho các tính toán liên quan đến số có hai chữ số hoặc số hai trăm chữ số, các công thức chi phối việc hiển thị vector image áp dụng giống hệt với graphic ở mọi kích thước.

2-layer maps - Geopandas

- Map Projections



2-layer maps - Geopandas

- Coordinate Reference Systems
  - EPSG:4326
    - Được sử dụng bởi Google Earth units là decimal degrees
  - EPSG:3857
    - Được sử dụng bởi Google Maps, Bing Maps, Open Street Maps units là meters

2-layer maps - Geopandas

- Thay đổi coordinate reference systems

```
# Print the first row of school districts GeoDataFrame and the crs
service_district.head(1)
```

	first_name	city	zip	email	state	last_name	address	position	term_expir	district	phone	geometry
0	Dr. Sharon	Nashville	37218	gentryfordistrict1@comcast.net	TN	Gentry	6108 Beals Lane	Member	2016	1	615-268-5269	(POLYGON ((-86.77136400034288 36.3835669997190...

```
print(service_district.crs)
# Convert the crs to epsg:3857
service_district.geometry = service_district.geometry.to_crs(epsg = 3857)

# Print the first row of school districts GeoDataFrame and the crs again
service_district.head(1)
```

```
{'init': 'epsg:4326'}
```

	first_name	city	zip	email	state	last_name	address	position	term_expir	district	phone	geometry
0	Dr. Sharon	Nashville	37218	gentryfordistrict1@comcast.net	TN	Gentry	6108 Beals Lane	Member	2016	1	615-268-5269	(POLYGON ((-9659344.055955959 4353528.76657080...

```
service_district.crs
```

```
{'init': 'epsg:3857', 'no_defs': True}
```

2-layer maps - Geopandas

- Cách tạo geometry column

```
# Create a geometry column from lng & lat
art['geometry'] = art.apply(lambda x: Point(float(x.Longitude), float(x.Latitude)), axis=1)
```

2-layer maps - Geopandas

```
from shapely.geometry import Point
import matplotlib.pyplot as plt
```

```
# Create a geometry column from lng & lat
art['geometry'] = art.apply(lambda x: Point(float(x.Longitude), float(x.Latitude)), axis=1)

art_crs = {'init':'epsg:4326'}
# Create a GeoDataFrame from art and verify the type
art_geo = gpd.GeoDataFrame(art, crs = art_crs, geometry = art.geometry)
type(art_geo)
```

```
geopandas.geodataframe.GeoDataFrame
```

```
art_geo.head(2)
```

	Title	Last Name	First Name	Location	Medium	Type	Description	Latitude	Longitude	Mapped Location	geometry
0	[Cross Country Runners]	Frost	Miley	4001 Harding Rd., Nashville TN	Bronze	Sculpture	NaN	36.12856	-86.83660	(36.12856, -86.8366)	POINT (-86.8366 36.12856)
1	[Fourth and Commerce Sculpture]	Walker	Lin	333 Commerce Street, Nashville TN	NaN	Sculpture	NaN	36.16234	-86.77774	(36.16234, -86.77774)	POINT (-86.77774000000001 36.16234)

Nội dung

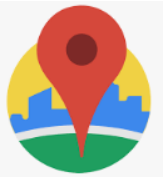
1.Giới thiệu

2.2 layer maps - Geopandas

3.Folium

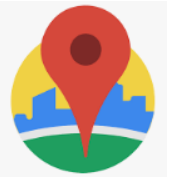
4.Choropleth Map

Folium



- Giới thiệu
  - Folium là một thư viện Python mạnh mẽ giúp tạo ra một số loại Leaflet map.
  - Folium được phát triển cho mục đích trực quan hóa dữ liệu không gian địa lý.

  - Cài đặt: pip install folium



Folium

- Folium xây dựng dựa trên các điểm mạnh về sắp xếp dữ liệu của Python ecosystem + các thế mạnh của thư viện Leaflet.js.
- Folium giúp dễ dàng trực quan dữ liệu đã được thao tác trong Python trên Leaflet map tương tác.
- Folium có thể liên kết dữ liệu với bản đồ để trực quan hóa Choropleth Map, làm marker trên bản đồ.

Folium

- Thư viện có một số built-in tileset dựng sẵn từ OpenStreetMap, Mapbox và Stamen và hỗ trợ tùy chỉnh tileset với các API key Mapbox hoặc Cloudcraft.
- Folium hỗ trợ cả overlay GeoJSON và TopoJSON, cũng như liên kết dữ liệu với các overlay đó để tạo bản đồ Choropleth Map với các bảng phối màu

Folium

- Tạo world map
  - Tạo world map rất dễ dàng trong Folium. Ta chỉ cần tạo một Folium Map object và sau đó hiển thị nó. Điều đáng chú ý về Folium map là chúng có thể tương tác, vì vậy ta có thể phóng to/thu nhỏ bất kỳ khu vực nào quan tâm.

Folium

```
# define the world map
world_map = folium.Map()

# display world map
world_map
```

Folium

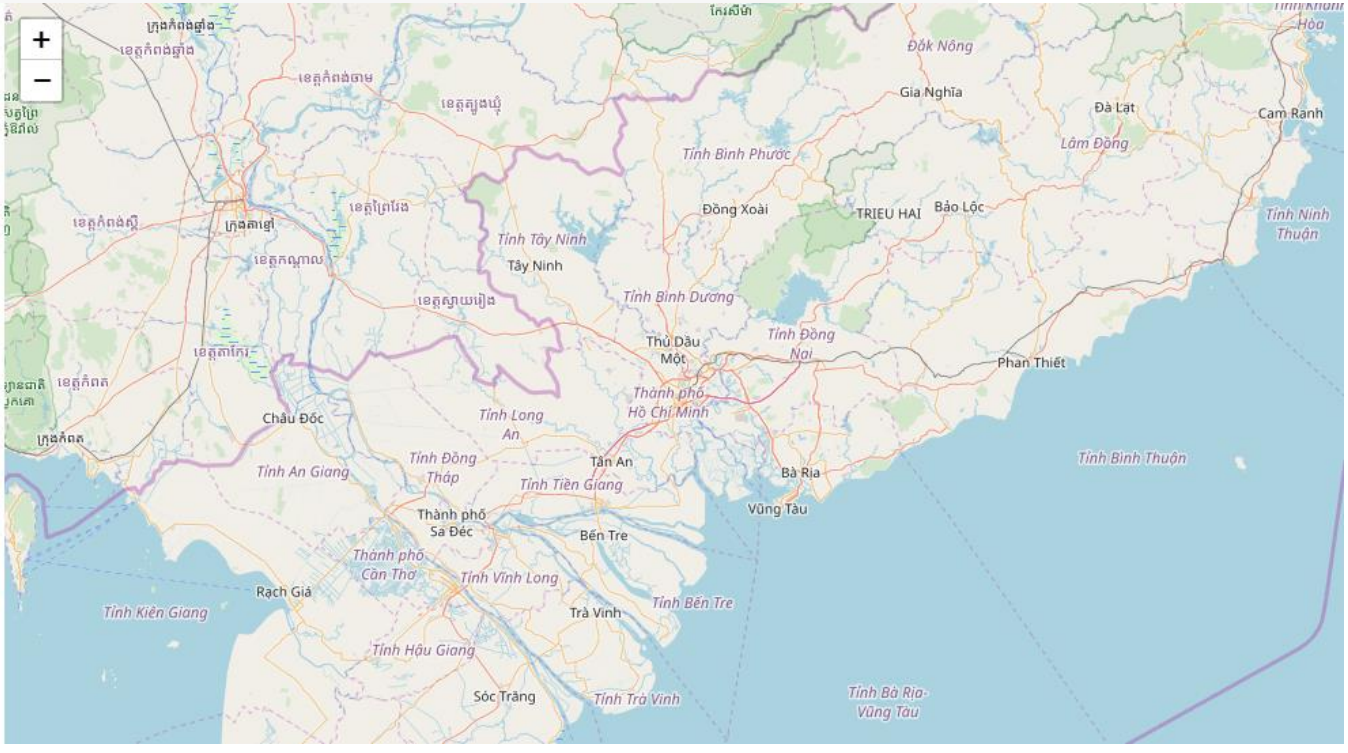
- Chỉ định trung tâm map và zoom level ban đầu
  - Mọi vị trí trên bản đồ đều được xác định bằng giá trị Latitude & Longitude. Do đó, ta có thể tạo map và đặt center là Latitude & Longitude = [0, 0].
  - Đối với center được xác định, ta cũng có thể thiết lập zoom level vào vị trí đó khi map được hiển thị. Giá trị zoom level càng lớn thì map càng được phóng to vào center.

Folium

- Ví dụ: tạo map và center ở quanh TP.HCM với zoom level = 8.

```
### HCMC
hcm_latitude = 10.762622
hcm_longitude = 106.660172

hcm_map = folium.Map(location=[hcm_latitude, hcm_longitude], zoom_start=8)
hcm_map
```



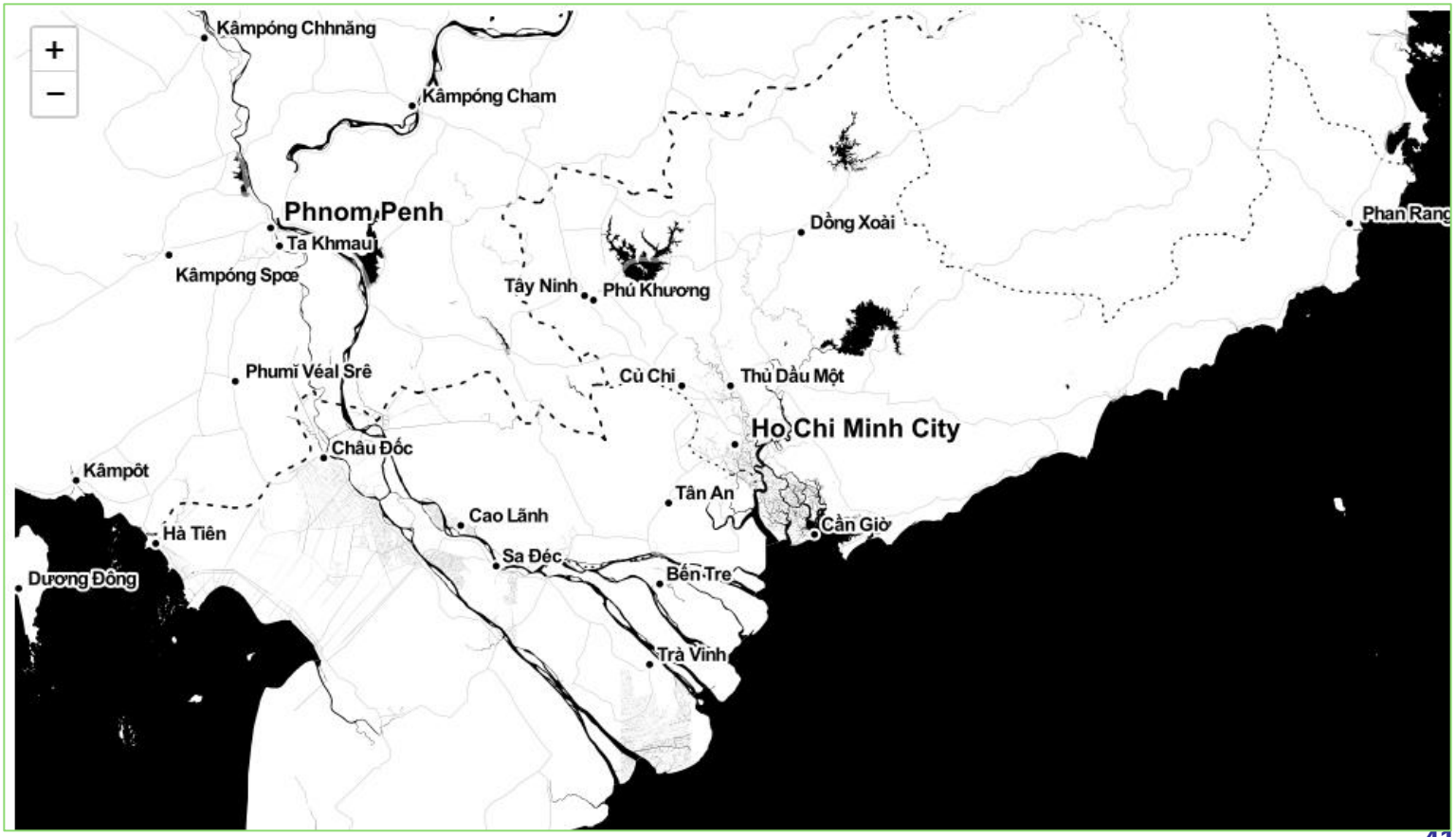
Pyth

Folium

- Folium map styles
  - Có nhiều style khác nhau như Stamen Toner, Stamen Watercolor, *Stamen Terrain, Mapbox Control Room, OpenStreetMap,* Map Quest Open, Mapbox Bright…
  - **Stamen Toner Map**
    - Là loại map high-contrast B+W (black & white) có độ tương phản cao.
    - Nó phù hợp cho cho việc khám phá dữ liệu khúc quanh các con sông và vùng ven biển

Folium

```
# create a Stamen Toner map of the world centered around HCMC
hcm_map = folium.Map(location=[hcm_latitude, hcm_longitude], zoom_start=8, tiles='Stamen Toner')
# display map
hcm_map
```

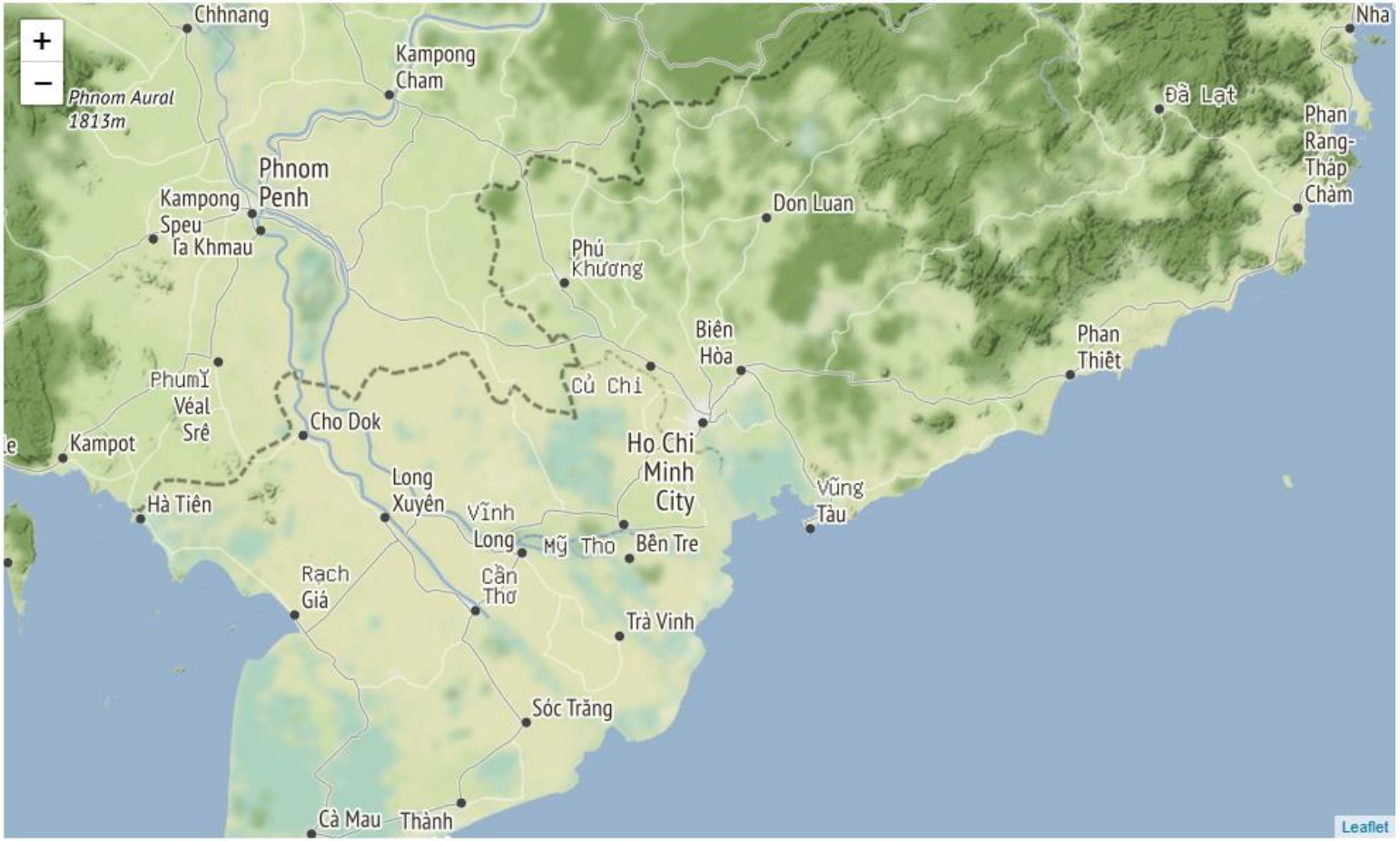


Folium

- **Stamen Terrain Map**
  - Là loại bản đồ có dạng đồi và thảm thực vật tự nhiên.
  - Được gán nhãn và khái quát hóa dạng đường của những đường hai chiều.

Folium

```
# create a Stamen Terrain map of the world centered around HCMC
hcm_map = folium.Map(location=[hcm_latitude, hcm_longitude], zoom_start=8, tiles='Stamen Terrain')
# display map
hcm_map
```



Folium

- Mapbox Bright Map
  - Là những bản đồ khá giống với kiểu mặc định, ngoại trừ các đường viền không hiển thị với mức thu phóng thấp.
  - Khác kiểu mặc định ở tên quốc gia được hiển thị bằng ngôn ngữ bản địa của mỗi quốc gia, Mapbox Bright style hiển thị tất cả tên quốc gia bằng tiếng Anh.

Folium

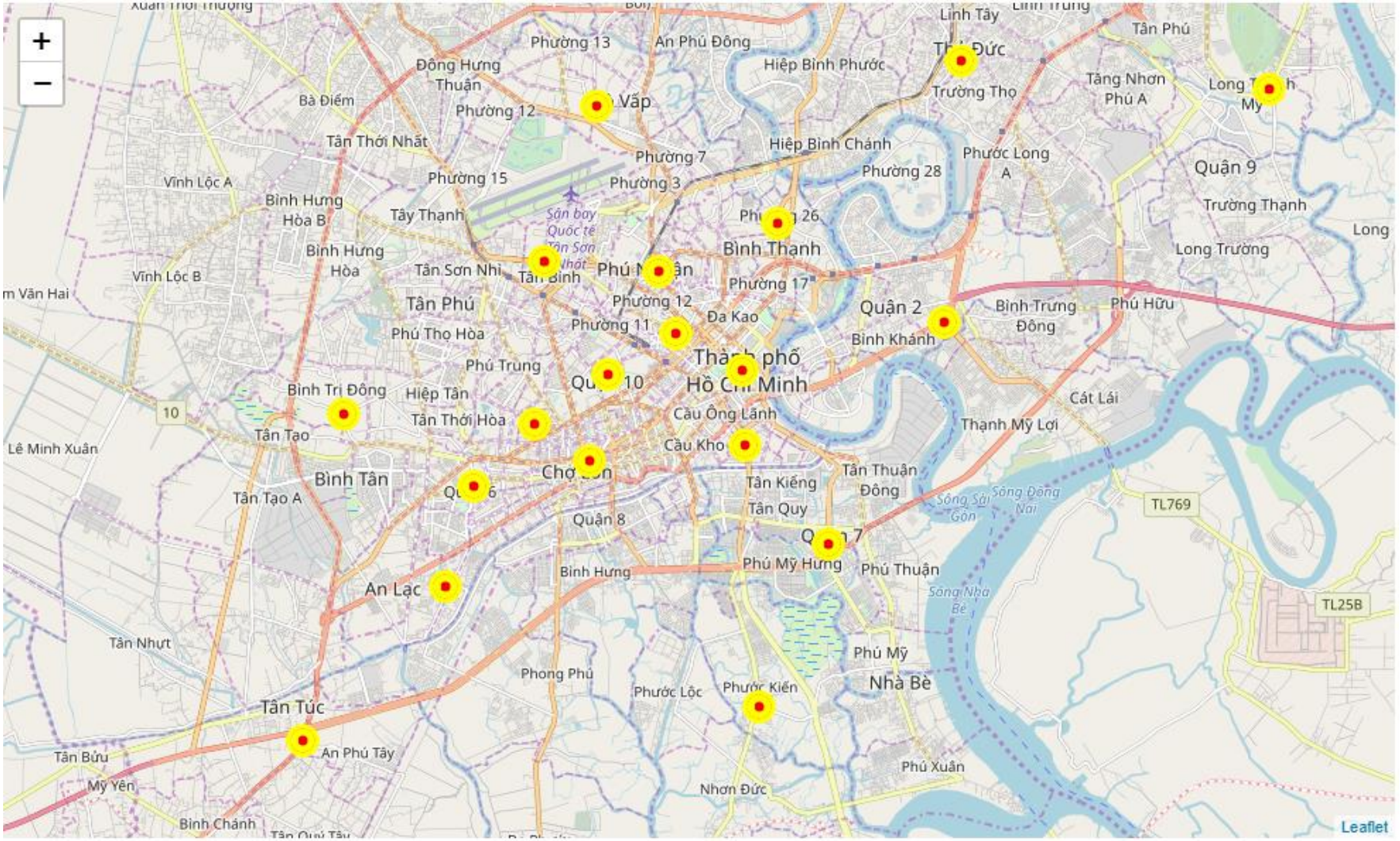
```
# create a Mapbox Bright map of the world centered around HCMC
hcm_map = folium.Map(location=[hcm_latitude, hcm_longitude], zoom_start=8, tiles='Mapbox Bright')
# display map
hcm_map
```

Folium

- Map với marker
  - Marker: là cách đánh dấu các vị trí trên map
    - Ví dụ: đánh dấu vị trí các quận của tp.HCM

```
# instantiate a feature group for the district in the dataframe
districts = folium.map.FeatureGroup()
# loop through the dictricts and add each to the dictrict feature group
for lat, lng, in zip(df_hcm.Latitude, df_hcm.Longitude):
    districts.add_child(
        folium.features.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='red',
            fill_opacity=0.6
        )
    )
# add dictricts to map
hcm_map.add_child(districts)
```

Folium


Linh Tây
Tân Phú
Phường 13
An Phú Đông
Hiệp Bình Phước
Đông Hưng Thuận
Trường Thọ
Tăng Nhơn Phú A
Bà Điểm
Phường 12
Tân Thới Nhất
Hiệp Bình Chánh
Phường 7
Phường 28
Phước Long A
Quận 9
Vĩnh Lộc A
Phường 15
Phường 3
Bình Hưng Hòa B
Tây Thạnh
Trường Thạnh
Long Trường
Bình Thạnh
Bình Hưng Hòa
Tân Sơn Nhì
Phường 17
Vĩnh Lộc B
Tân Phú
Phường 12
Quận 2
Bình Trưng Đông
Phú Hữu
Phường 11
Đa Kao
Bình Khánh
Phú Thọ Hòa
Phú Trung
Hồ Chí Minh
Bình Trị Đông
Hiệp Tân
Cát Lái
Tân Thới Hòa
Cầu Ông Lãnh
Thạnh Mỹ Lợi
Tân Tạo
Cầu Kho
Lê Minh Xuân
Bình Tân
Tân Kiểng
Tân Thuận Đông
Tân Tạo A
Tân Quy
Quận 8
TL769
Sông Sài Gòn
Sông Đồng Nai
Phú Mỹ Hưng
Phú Thuận
Bình Hưng
An Lạc
Sông Nhà Bè
TL25B
Tân Nhựt
Phú Mỹ
Phong Phú
Phước Lộc
Phước Kiển
Nhà Bè
Tân Túc
An Phú Tây
Tân Bửu
Phú Xuân
Mỹ Yên
Nhơn Đức
Bình Chánh
Leaflet

Folium

- Thêm pop-up text cho marker trên map

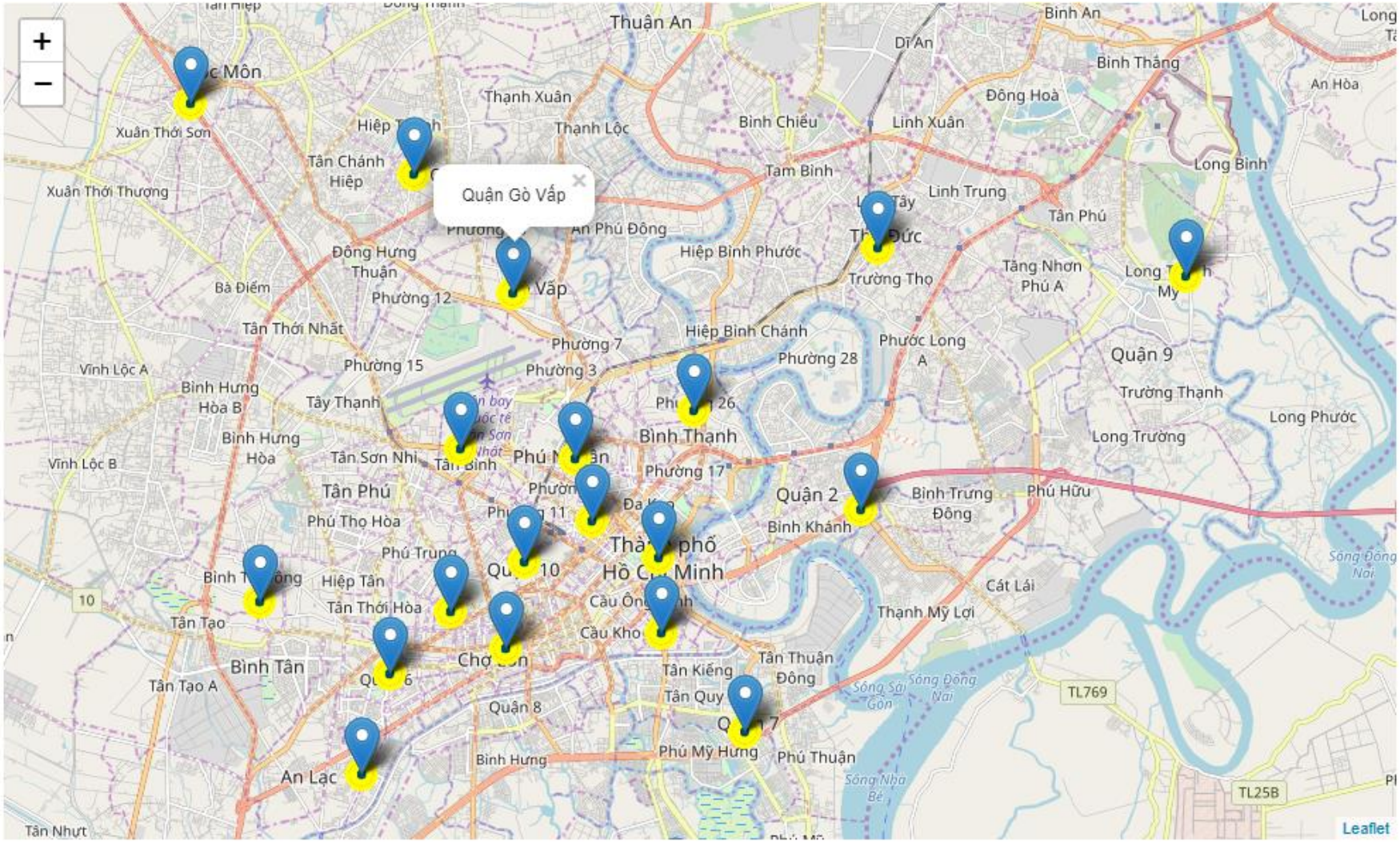
```
# instantiate a feature group for the dictricts  in the dataframe
dictricts = folium.map.FeatureGroup()

# loop through the district and add each to the dictricts feature group
for lat, lng, in zip(df_hcm.Latitude, df_hcm.Longitude):
    districts.add_child(
        folium.features.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='green',
            fill_opacity=0.6
        )
    )
# add pop-up text to each marker on the map
latitudes = list(df_hcm.Latitude)
longitudes = list(df_hcm.Longitude)
labels = list(df_hcm.Name)

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

# add dictricts to map
hcm_map.add_child(dictricts)
```

Folium

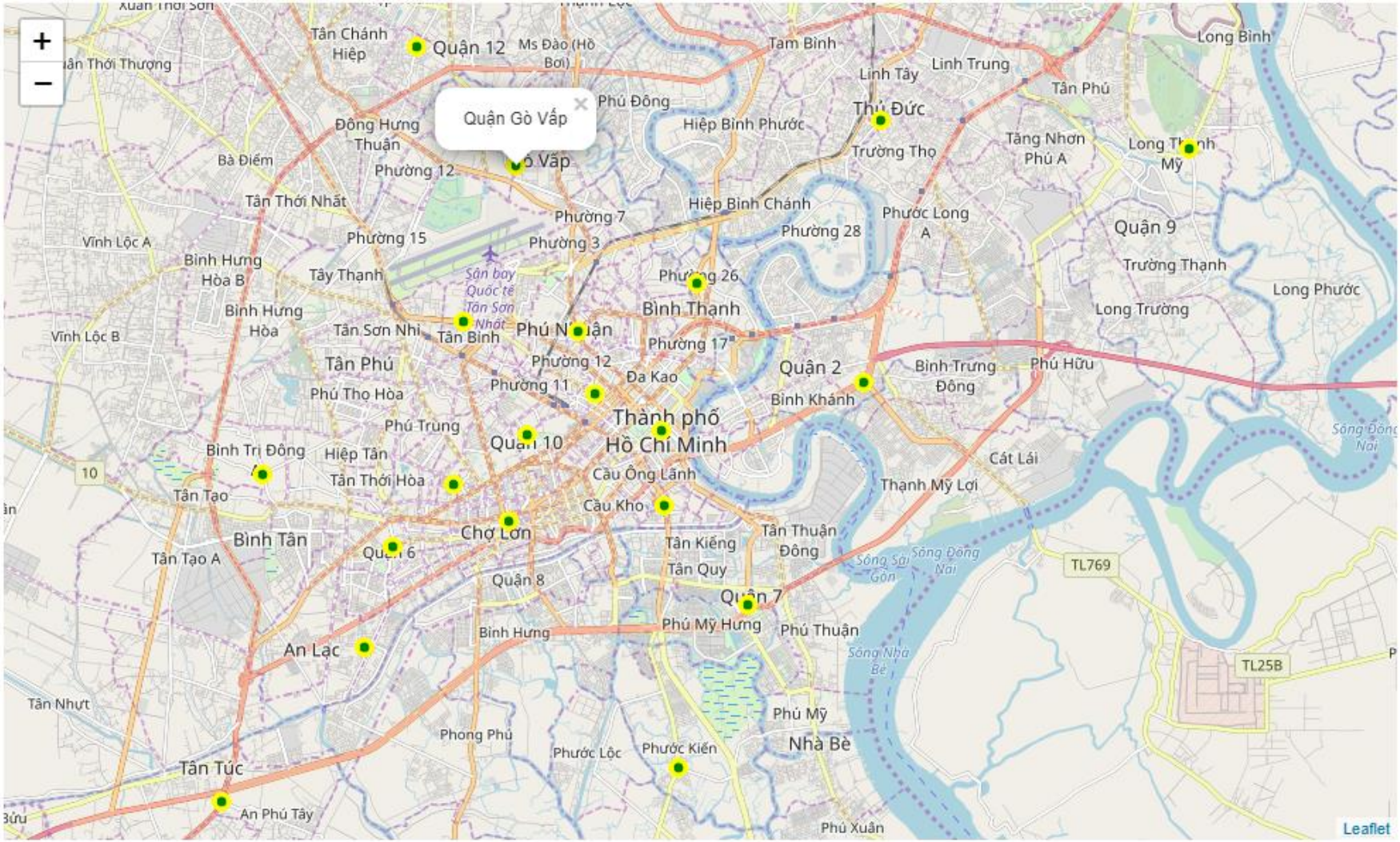

Quận Gò Vấp
Thuận An
Dĩ An
Bình An
Bình Thắng
Đông Hoà
An Hòa
Thạnh Xuân
Thạnh Lộc
Bình Chiểu
Linh Xuân
Xuân Thới Sơn
Tân Chánh Hiệp
Tam Bình
Long Bình
Xuân Thới Thượng
Linh Trung
Tân Phú
Đông Hưng Thuận
An Phú Đông
Hiệp Bình Phước
Trường Thọ
Tăng Nhơn Phú A
Bà Điểm
Phường 12
Tân Thới Nhất
Hiệp Bình Chánh
Phường 7
Phường 28
Phước Long A
Quận 9
Vĩnh Lộc A
Phường 15
Phường 3
Bình Hưng Hòa B
Tây Thạnh
Trường Thạnh
Long Phước
Bình Hưng Hòa
Bình Thạnh
Long Trường
Vĩnh Lộc B
Tân Sơn Nhì
Phường 17
Quận 2
Bình Trưng Đông
Phú Hữu
Tân Phú
Phú Thọ Hòa
Bình Khánh
Phú Trung
Hiệp Tân
Tân Thới Hòa
Cát Lái
Thạnh Mỹ Lợi
Tân Tạo
Cầu Kho
Bình Tân
Tân Tạo A
Tân Kiểng
Tân Thuận Đông
Tân Quy
Quận 8
TL769
Phú Mỹ Hưng
Phú Thuận
Bình Hưng
An Lạc
TL25B
Tân Nhựt
Leaflet

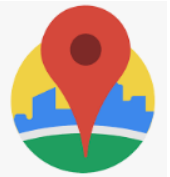
Folium

- Nếu bản đồ có quá nhiều marker gây khó quan sát thì có thể xóa các location marker và chỉ thêm text vào các điểm đánh dấu.

```
hcm_map = folium.Map(location=[hcm_latitude, hcm_longitude], zoom_start=12)
# instantiate a feature group for the dictricts in the dataframe
dictricts = folium.map.FeatureGroup()
# loop through the district and add each to the dictricts feature group
for lat, lng, label in zip(df_hcm.Latitude, df_hcm.Longitude, df_hcm.Name):
    districts.add_child(
        folium.features.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='green',
            fill_opacity=0.6,
            popup=label
        )
    ).add_to(hcm_map)
hcm_map
```

Folium


Quận Gò Vấp
Quận 12
Tân Chánh Hiệp
Ms Đào (Hồ Bơi)
Tam Bình
Long Bình
Linh Tây
Linh Trung
Tân Phú
Thủ Đức
Phú Đông
Hiệp Bình Phước
Đông Hưng Thuận
Bà Điểm
Phường 12
Trường Thọ
Tăng Nhơn Phú A
Long Thạnh Mỹ
Tân Thới Nhất
Phường 7
Hiệp Bình Chánh
Phước Long A
Phường 28
Quận 9
Vĩnh Lộc A
Phường 15
Phường 3
Bình Hưng Hòa B
Tây Thạnh
Trường Thạnh
Long Phước
Long Trường
Sân bay Quốc tế Tân Sơn Nhất
Bình Thạnh
Bình Hưng Hòa
Tân Sơn Nhì
Tân Bình
Phú Nhuận
Phường 17
Vĩnh Lộc B
Tân Phú
Phường 12
Đa Kao
Quận 2
Bình Trưng Đông
Phú Hữu
Phú Thọ Hòa
Phường 11
Bình Khánh
Thành phố Hồ Chí Minh
Phú Trung
Quận 10
Bình Trị Đông
Hiệp Tân
Cát Lái
10
Tân Thới Hòa
Cầu Ông Lãnh
Thạnh Mỹ Lợi
Tân Tạo
Cầu Kho
Chợ Lớn
Tân Thuận Đông
Bình Tân
Tân Kiểng
Tân Tạo A
Quận 6
Tân Quy
Quận 8
Quận 7
TL769
Phú Mỹ Hưng
Phú Thuận
Bình Hưng
An Lạc
TL25B
Tân Nhựt
Phú Mỹ
Phong Phú
Phước Lộc
Phước Kiển
Nhà Bè
Tân Túc
An Phú Tây
Phú Xuân
Leaflet



Nội dung

1.Giới thiệu

2.2 layer maps - Geopandas

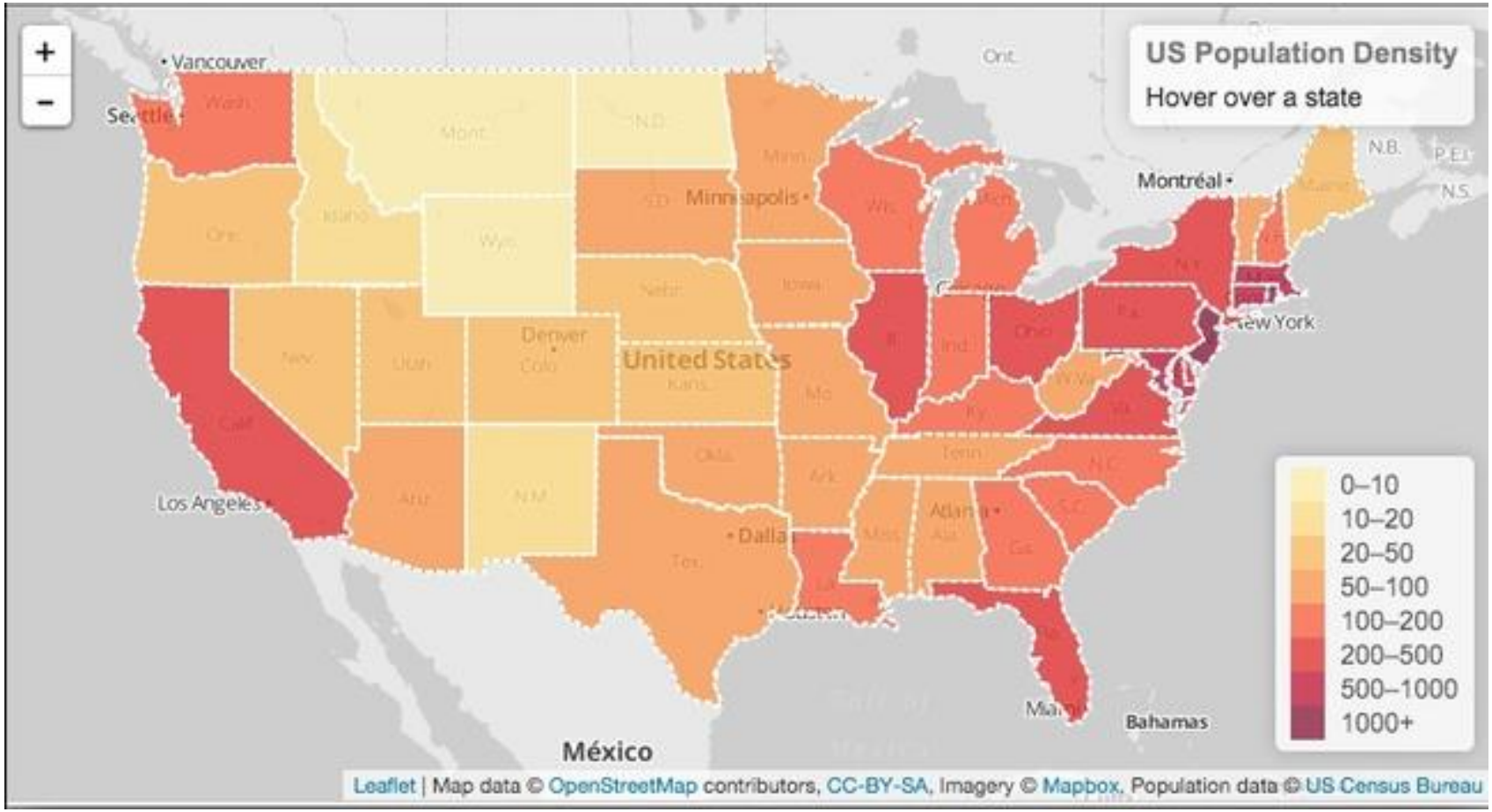
3.Folium

4.Choropleth Map

Choropleth Map

- Choropleth map là bản đồ chuyên đề, trong đó các khu vực được tô bóng hoặc tạo khuôn theo tỷ lệ với phép đo của biến thống kê được hiển thị trên bản đồ, như mật độ dân số hoặc thu nhập bình quân đầu người.
- Choropleth map trực quan hóa cách đo lường thay đổi trong một khu vực địa lý hoặc cho thấy mức độ biến đổi trong một khu vực.

Choropleth Map



Choropleth Map

- Để tạo bản đồ Choropleth, cần phải có tập tin GeoJSON xác định các khu vực / ranh giới của tiểu bang, hạt hoặc quốc gia sẽ vẽ.
- Khi cần tạo bản đồ thế giới, ta cần GeoJSON xác định ranh giới của tất cả các nước trên thế giới (được cung cấp sẵn trong tập tin world_countries.json)

Choropleth Map

- Tạo Choropleth Map
  - Sử dụng phương thức choropleth(), với các tham số chính như sau:
    - geo_data: GeoJSON file.
    - Data: dataframe chứa data.
    - Columns: là các cột trong dataframe sẽ được dùng để tạo Choropleth map.
    - key_on: là key/variable trong GeoJSON file chứa tên của variable quan tâm. Để xác định cần phải mở GeoJSON file và lưu ý name của key/ variable chứa tên của của variable quan tâm, tên có phân biệt chữ hoa/thường nên cần lấy chính xác tên trong GeoJSON file.

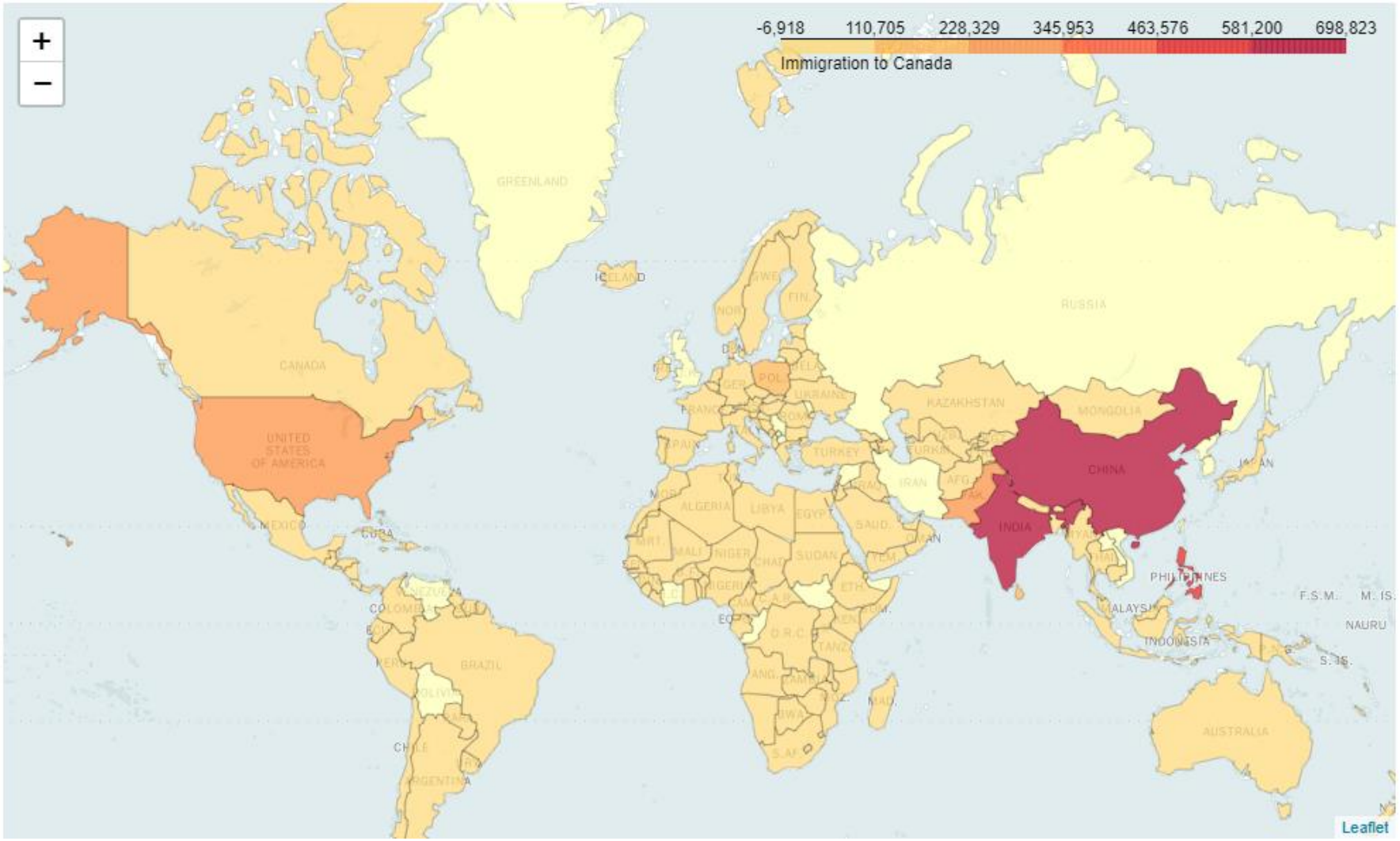
Choropleth Map

```
world_geo = r'data/world-countries.json' # geojson file
# create a plain world map
world_map = folium.Map(location=[0, 0], zoom_start=2, tiles='Mapbox Bright')
```

```
# generate choropleth map using the total immigration of each country to Canada from 1980 to 2013
world_map.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'
)

# display map
world_map
```

Choropleth Map



Choropleth Map

- Trong trường hợp legend hiển thị negative boundary threshold, có thể chỉnh lại bằng cách thiết lập thresholds = 0 (thay vì negative)

Choropleth Map

```
# create a numpy array of length 6 and has linear spacing
# from the minium total immigration to the maximum total immigration
threshold_scale = np.linspace(df_can['Total'].min(),
                              df_can['Total'].max(),
                              6, dtype=int)
threshold_scale = threshold_scale.tolist() # change the numpy array to a list
# make sure that the last value of the list is greater than the maximum immigration
threshold_scale[-1] = threshold_scale[-1] + 1

# let Folium determine the scale.
world_map = folium.Map(location=[0, 0], zoom_start=2, tiles='Mapbox Bright')
world_map.choropleth(
    geo_data=world_geo,
    data=df_can,
    columns=['Country', 'Total'],
    key_on='feature.properties.name',
    threshold_scale=threshold_scale,
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Immigration to Canada',
    reset=True
)
world_map
```

Choropleth Map