

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO HỌC THUẬT

**TRỰC QUAN HÓA DỮ LIỆU BẢN ĐỒ TƯƠNG TÁC
VỚI FOLIUM TRONG PYTHON**

PHẠM AN CƯỜNG

Hà Nội 6/2023

MỤC LỤC

I. KHÁI NIỆM VỀ FOLIUM.....	3
I.1 Folium là gì?.....	3
I.2 Vẽ bản đồ với Folium	3
I.3 Lớp và Ô trong Folium.....	4
I.4 Đánh dấu đồ thị trên bản đồ	5
I.5 Vẽ đường dẫn bằng Folium	8
II. TẠO BẢN ĐỒ VỚI FOLIUM VÀ PYTHON.....	12
II.1 Import folium và các gói hữu ích khác	12
II.2 Điểm đánh dấu:	13
II.3 Kiểu đánh dấu Vincent/Vega and Altair/VegaLite	16
II.4 Lớp phủ GeoJSON / TopoJSON	17
II.5 Bản đồ Choropleth.....	18
II.6 Chức năng tạo kiểu:	20
III. KẾT LUẬN.....	23

I. KHÁI NIỆM VỀ FOLIUM

I.1 Folium là gì?

Folium là một thư viện Python được sử dụng để hiển thị dữ liệu không gian địa lý. Nó rất dễ sử dụng và vẫn là một thư viện mạnh mẽ. Folium là một trình bao bọc Python cho Leaflet.js, là thư viện JavaScript mã nguồn mở hàng đầu để vẽ các bản đồ tương tác.

Nó có sức mạnh của Leaflet.js và sự đơn giản của Python, khiến nó trở thành một công cụ tuyệt vời để vẽ bản đồ. Folium được thiết kế chú trọng đến tính đơn giản, hiệu suất và khả năng sử dụng. Nó hoạt động hiệu quả, có thể mở rộng với rất nhiều plugin, có API đẹp và dễ sử dụng.

Cài đặt Folium

Folium được cài đặt trên các nhân Google Colab và Kaggle. Tuy nhiên, nếu bạn chưa cài đặt Folium theo mặc định trên hệ thống của mình, bạn có thể cài đặt Folium bằng lệnh sau:

```
pip install folium
```

```
conda install -c conda-forge folium
```

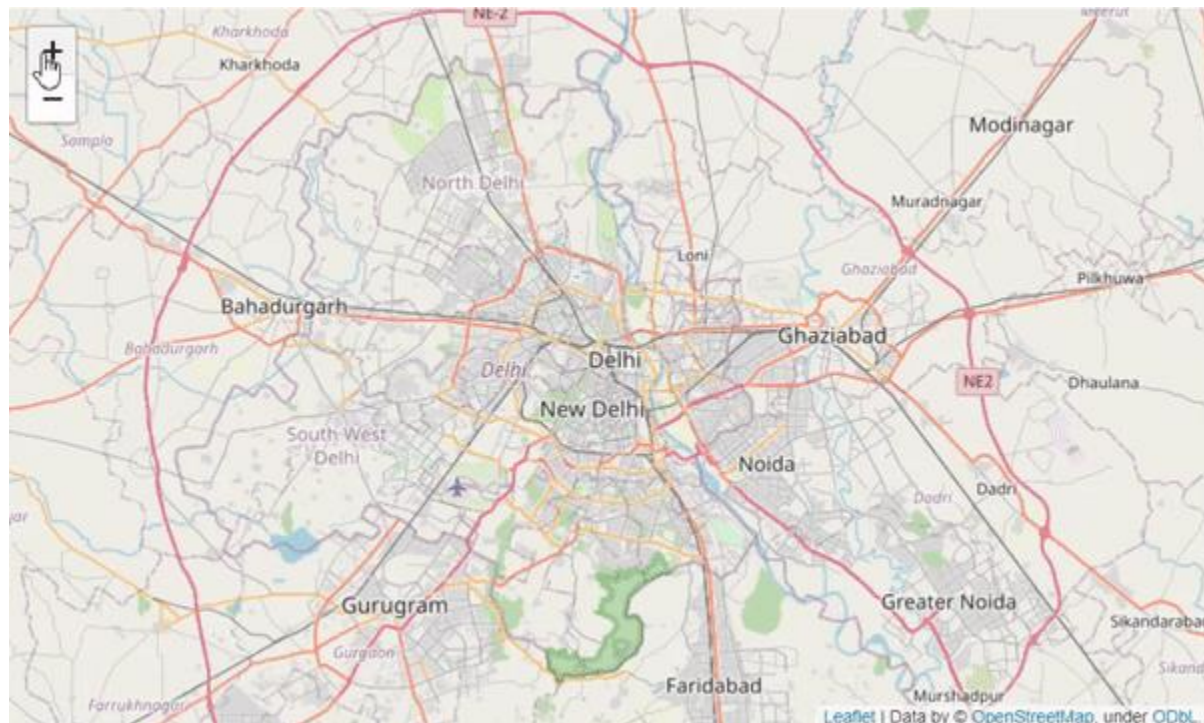
I.2 Vẽ bản đồ với Folium

Folium cung cấp lớp `folium.Map()` lấy tham số vị trí về vĩ độ và kinh độ và tạo bản đồ xung quanh nó. Vì vậy, hãy vẽ một bản đồ của Delhi với vĩ độ và kinh độ lần lượt là 28,644800 và 77,216721:

```
import folium
```

```
m=folium.Map(location=[28.644800, 77.216721])
```

```
m
```



Những bản đồ này có tính tương tác. Có thể phóng to và thu nhỏ bằng cách nhấp vào các nút + và - ở góc trên bên trái của bản đồ. Cũng có thể kéo bản đồ và xem các khu vực khác nhau.

Hãy thử tùy chỉnh bản đồ này ngay bây giờ. Trước tiên, giảm chiều cao và chiều rộng của bản đồ, sau đó thay đổi mức thu phóng.

Có thể thay đổi kích thước bản đồ của mình bằng cách sử dụng thư viện branca trong Python. Nó là một bản phụ từ Folium lưu trữ các tính năng cụ thể không phải bản đồ. Có thể sử dụng lớp hình của nó để thay đổi kích thước bản đồ và chuyển chiều rộng và chiều cao mong muốn theo pixel:

```
from branca.element import Figure
```

```
fig=Figure(width=550,height=350)
```

Theo mặc định, có thể phóng to và thu nhỏ bao nhiêu tùy thích. Mức thu phóng bắt đầu được cố định thành 10 (bản đồ được vẽ với mức thu phóng 10x). Folium cung cấp ba tham số - `zoom_start`, `min_zoom` và `max_zoom` để vẽ lại bản đồ với kích thước cố định và với mức thu phóng hạn chế:

```
m1=folium.Map(width=550,height=350,location=[28.644800,  
77.216721],zoom_start=11,min_zoom=8,max_zoom=14)  
fig.add_child(m1)  
m1
```



1.3 Lớp và Ô trong Folium

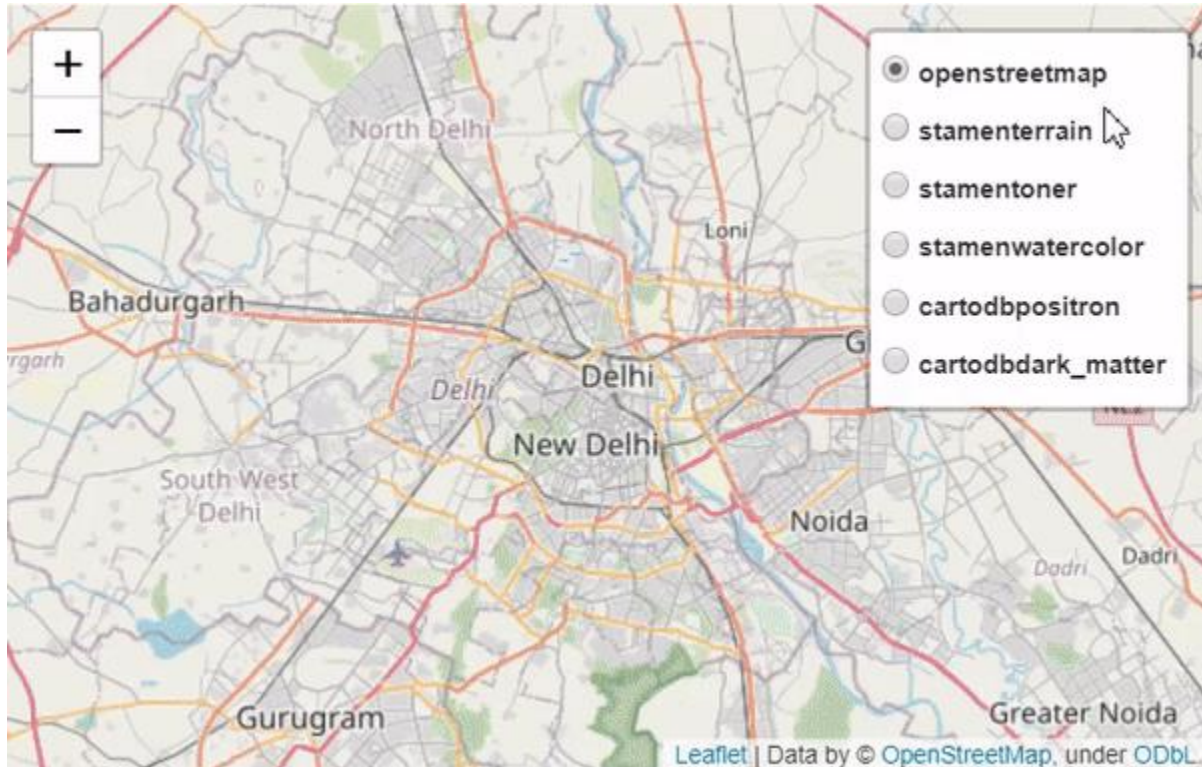
Tập hợp các ô là tập hợp dữ liệu raster và vectơ được chia thành một lưới ô vuông đồng nhất. Mỗi tập hợp có một cách khác nhau để biểu diễn dữ liệu trong bản đồ. Folium cho phép chúng tôi tạo bản đồ với các ô khác nhau như Stamen Terrain, Stamen Toner, Stamen Water Color, CartoDB Positron, v.v. Theo mặc định, các ô được đặt thành OpenStreetMap.

Mỗi bộ xếp hình hiển thị các tính năng khác nhau của bản đồ và phù hợp với các mục đích khác nhau. Ví dụ, Stamen Terrain có bóng râm trên đồi và màu sắc của thảm thực vật tự nhiên. Nó giới thiệu sự khái quát về nhãn hiệu tiên tiến và cách xếp hàng của đường hai chiều. Và, CartoDB Dark Matter hiển thị bản đồ CartoDB Positron ở chế độ tối.

Vì bây giờ chúng tôi biết rằng mỗi tập hợp cung cấp thông tin theo một cách khác nhau và phục vụ một mục đích khác nhau, chúng tôi có thể xếp chúng chồng lên nhau để có thêm thông tin bằng cách vẽ một bản đồ duy nhất. Chúng ta có thể thực hiện việc này bằng cách thêm các lớp lát khác nhau vào một bản đồ:

```
fig2=Figure(width=550,height=350)
```

```
m2=folium.Map(location=[28.644800, 77.216721])
fig2.add_child(m2)
folium.TileLayer('Stamen Terrain').add_to(m2)
folium.TileLayer('Stamen Toner').add_to(m2)
folium.TileLayer('Stamen Water Color').add_to(m2)
folium.TileLayer('cartodbpositron').add_to(m2)
folium.TileLayer('cartodbdark_matter').add_to(m2)
folium.LayerControl().add_to(m2)
m2
```



I.4 Đánh dấu đồ thị trên bản đồ

Điểm đánh dấu là các mục được sử dụng để đánh dấu một vị trí trên bản đồ. Ví dụ: khi bạn sử dụng Google Maps để điều hướng, vị trí của bạn được đánh dấu bằng một điểm đánh dấu và điểm đến của bạn được đánh dấu bằng một điểm đánh dấu khác. Điểm đánh dấu là một trong những thứ quan trọng và hữu ích nhất trên bản đồ.

Folium cung cấp một lớp `folium.Marker()` để vẽ các điểm đánh dấu trên bản đồ. Chỉ cần vượt qua vĩ độ và kinh độ của vị trí, đề cập đến cửa sổ bật lên và chú giải công cụ và thêm nó vào bản đồ.

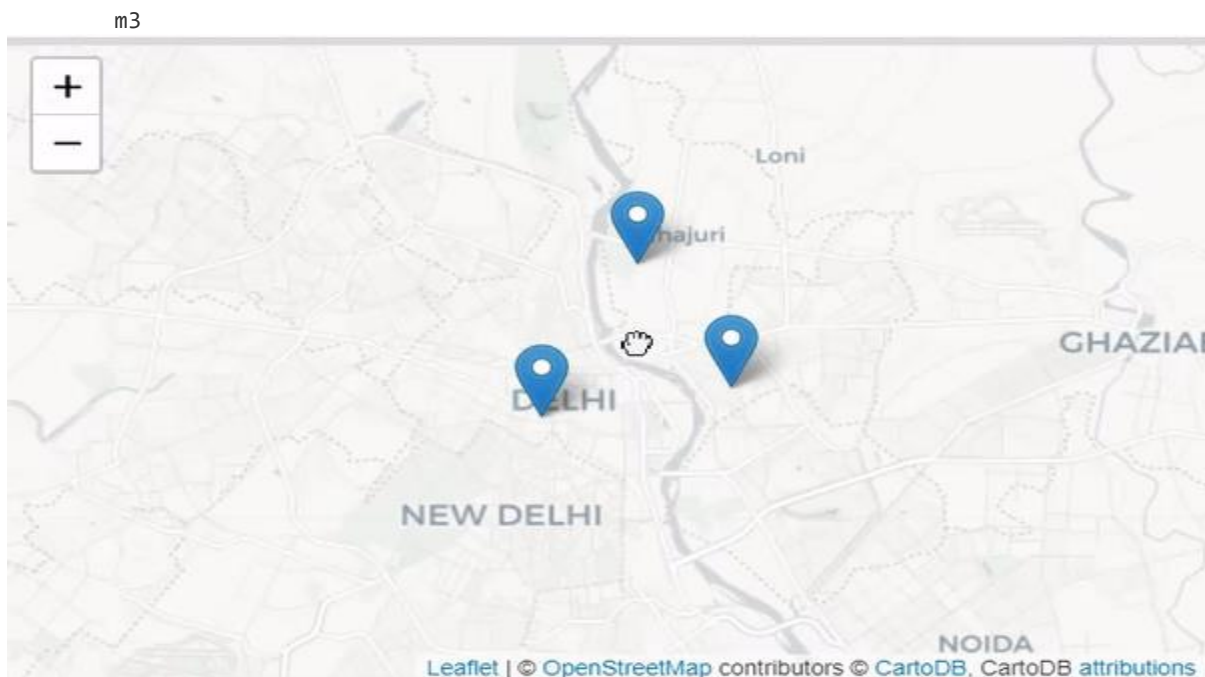
Vẽ các điểm đánh dấu là một quá trình gồm hai bước. Trước tiên, bạn cần tạo một bản đồ cơ sở mà trên đó các điểm đánh dấu của bạn sẽ được đặt, sau đó thêm các điểm đánh dấu của bạn vào đó:

```

#
Creating
Basemap
fig3=Figure(width=550,height=350)
m3=folium.Map(location=[28.644800, 77.216721],tiles='cartodbpositron',zoom_start=11)
fig3.add_child(m3)

#Adding markers to the map
folium.Marker(location=[28.695800, 77.244721],popup='Default popup
Marker1',tooltip='Click here to see Popup').add_to(m3)
folium.Marker(location=[28.645800,
77.214721],popup='<strong>Marker3</strong>',tooltip='<strong>Click here to see
Popup</strong>').add_to(m3)
folium.Marker(location=[28.655800, 77.274721],popup='<h3
style="color:green;">Marker2</h3>',tooltip='<strong>Click here to see
Popup</strong>').add_to(m3)

```



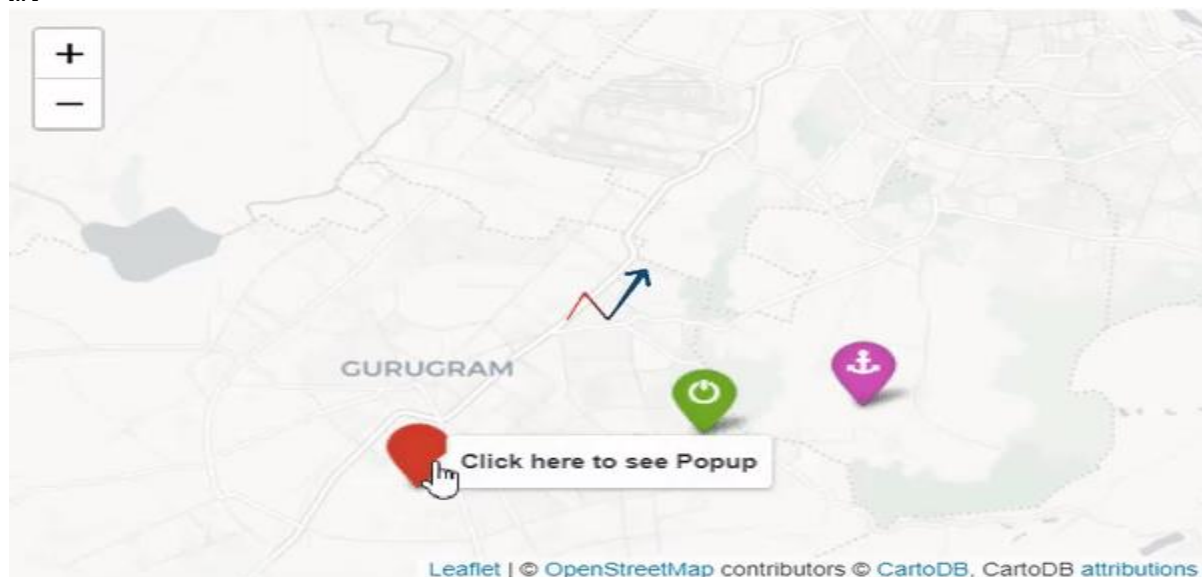
Ở đây, tôi đã tạo ra ba điểm đánh dấu trên bản đồ của Delhi. Văn bản khi bạn di chuột qua điểm đánh dấu được gọi là chú giải công cụ và nội dung khi bạn nhấp vào điểm đánh dấu được gọi là cửa sổ bật lên. Nếu bạn quan sát kỹ ở đây, tôi đã tùy chỉnh các chú giải công cụ và cửa sổ bật lên của điểm đánh dấu 2 và 3. Bạn cũng có thể tùy chỉnh chúng theo nhu cầu của mình bằng cách chuyển nội dung dưới dạng HTML vào lớp.

Có hai thứ mà có thể tùy chỉnh để thay đổi hình thức của điểm đánh dấu. Đầu tiên, bạn có thể thay đổi biểu tượng của điểm đánh dấu và thứ hai, bạn có thể thay đổi hình dạng của điểm đánh dấu. Folium cung cấp cho lớp `folium.Icon()` có thể được sử dụng để tạo các biểu tượng tùy chỉnh cho các điểm đánh dấu. Bạn có thể chọn các biểu tượng từ nhiều tùy chọn.

Biểu tượng `()` có ba đối số - màu sắc, tiền tố và biểu tượng. Màu được sử dụng để thay đổi màu của điểm đánh dấu, tiền tố được sử dụng để chọn nhà cung cấp biểu tượng (fa cho Fontawesome và glyphicon cho Glyphicon) và biểu tượng được sử dụng để chọn tên biểu tượng.

Bây giờ, hãy tìm hiểu cách bạn có thể thay đổi hình dạng của điểm đánh dấu. Folium có lớp `folium.features.CustomIcon()` có thể được sử dụng để tạo các điểm đánh dấu tùy chỉnh. Nó lấy đường dẫn của hình ảnh và kích thước biểu tượng tính bằng pixel làm đối số và tạo một đối tượng biểu tượng. Đối tượng biểu tượng này có thể được chuyển đến lớp `folium.Icon()` như một biểu tượng để tạo các điểm đánh dấu tùy chỉnh:

```
# Creating Basemap
fig4=Figure(height=350,width=550)
m4=folium.Map(location=[28.4911091,77.0867361],tiles='cartodbpositron',zoom_start=11)
fig4.add_child(m4)
# Adding Custom Markers
folium.Marker(location=[28.4211091,77.0267361],popup='Custom Marker 1',tooltip='<strong>Click here to see Popup</strong>',icon=folium.Icon(color='red',icon='none')).add_to(m4)
folium.Marker(location=[28.4411091,77.1167361],popup='Custom Marker 2',tooltip='<strong>Click here to see Popup</strong>',icon=folium.Icon(color='green',prefix='glyphicon',icon='off')).add_to(m4)
folium.Marker(location=[28.4511091,77.1667361],popup='Custom Marker 3',tooltip='<strong>Click here to see Popup</strong>',icon=folium.Icon(color='purple',prefix='fa',icon='anchor')).add_to(m4)
folium.Marker(location=[28.4911091,77.0867361],popup='Custom Marker 4- <b>Analytics Vidhya</b>',tooltip='<strong>Click here to see Popup</strong>',icon=folium.features.CustomIcon('av.png',icon_size=(50,50))).add_to(m4)
m4
```



L.5 Vẽ đường dẫn bằng Folium

Như tôi đã nói trước đó, chúng tôi sẽ vẽ đường đi của 3 phương tiện trên bản đồ. Chúng tôi có ba mảng cho ba phương tiện chứa vĩ độ và kinh độ của đường đi của chúng:

```
coords_1=[[28.65685,77.21899],[28.65699,77.21898],[28.65699,77.21898],  
[28.65702,77.21875],[28.65702,77.21875],[28.6547,77.21902],[28.65308,77.21921  
],  
[28.65286,77.21926],[28.65255,77.21938],[28.65227,77.21954],[28.65152,77.2200  
5],  
[28.64655,77.22346],[28.64643,77.22354],[28.64635,77.22357],[28.64623,77.2236  
],  
[28.64616,77.22363],[28.64612,77.22364],[28.64557,77.22364],[28.64525,77.2236  
5],  
[28.64525,77.22365],[28.64517,77.22366],[28.64515,77.22367],[28.64512,77.2236  
7],  
[28.64504,77.22369],[28.64493,77.22371],[28.64483,77.22373],[28.64462,77.2237  
7],  
[28.6445,77.22381],[28.64444,77.22383],[28.64439,77.22387],[28.64434,77.2239]  
,  
[28.64429,77.22395],[28.64421,77.22403],[28.64417,77.22412],[28.64407,77.2242  
9],  
[28.64391,77.22465],[28.64353,77.22539],[28.64282,77.22678],[28.64282,77.2267  
8],  
[28.64285,77.2268],[28.64287,77.22683],[28.64289,77.22686],[28.64291,77.2269]  
,  
[28.64292,77.22694],[28.64292,77.22697],[28.64292,77.22701],[28.64292,77.2270  
5],  
[28.64292,77.22705],[28.64291,77.22711],[28.64288,77.22716],[28.64285,77.2272  
],  
[28.64281,77.22724],[28.64276,77.22726],[28.64271,77.22728],[28.64266,77.2272  
8],  
[28.64261,77.22727],[28.64261,77.22727],[28.64257,77.22725],[28.64254,77.2272  
3],  
[28.64251,77.2272],[28.64249,77.22717],[28.64247,77.22713],[28.64245,77.22709  
],  
[28.64244,77.22704],[28.64244,77.22699],[28.64245,77.22693],[28.64246,77.2268  
9],  
[28.64214,77.22665],[28.64169,77.22632],[28.64163,77.22627],[28.64095,77.2258  
2],  
[28.64009,77.22516],[28.63913,77.22445],[28.63799,77.22352],[28.63794,77.2234  
9],  
[28.63761,77.22327],[28.6375,77.2232],[28.63737,77.22311],[28.63727,77.22304]  
,  
[28.63706,77.22288],[28.63631,77.22233],[28.63528,77.2216],[28.63521,77.22154  
],  
[28.63521,77.22154],[28.63518,77.22158],[28.63499,77.2218],[28.63471,77.22212  
],  
[28.63442,77.22238],[28.63411,77.2226],[28.6339,77.22272],[28.63368,77.22282]  
,  
[28.63345,77.2229],[28.63313,77.22297],[28.63281,77.22299],[28.63249,77.22298  
],  
[28.63217,77.22293],[28.63186,77.22283],[28.63141,77.22256],[28.63125,77.2224  
],  
[28.63101,77.22227],[28.63082,77.22207],[28.63062,77.22185],[28.63046,77.2216  
2],  
[28.63029,77.22141],[28.63029,77.22141],[28.63026,77.22138],[28.63022,77.2213  
6],
```


[28.63018,77.22135],[28.63014,77.22134],[28.6301,77.22134],[28.63006,77.22135],
[28.63004,77.22135],[28.63004,77.22135]]

coords_2=[[28.63382,77.22045],[28.63458,77.221],[28.63514,77.22142],
[28.63517,77.22146],[28.63519,77.22151],[28.63518,77.22158],[28.63518,77.22158],
[28.63499,77.2218],[28.63471,77.22212],[28.63442,77.22238],[28.63411,77.2226],
[28.6339,77.22272],[28.63368,77.22282],[28.63345,77.2229],[28.63313,77.22297],
[28.63281,77.22299],[28.63249,77.22298],[28.63217,77.22293],[28.63186,77.22283],
[28.63141,77.22256],[28.63141,77.22256],[28.63129,77.22258],[28.63121,77.22259],
[28.63113,77.22266],[28.63072,77.22335],[28.63063,77.22353],[28.63059,77.22359],
[28.63046,77.22381],[28.63043,77.22388],[28.62914,77.22615],[28.62914,77.22615],
[28.62922,77.22648],[28.62923,77.22651],[28.62924,77.22653],[28.62925,77.22655],
[28.62926,77.22657],[28.62935,77.22665],[28.62937,77.22667],[28.62938,77.22669],
[28.62939,77.22672],[28.62943,77.22682],[28.62943,77.22682],[28.62972,77.22701],
[28.63027,77.22744],[28.63436,77.23048],[28.63456,77.2306],[28.63482,77.23072],
[28.63511,77.23084],[28.63545,77.23091],[28.63581,77.23095],[28.63656,77.23101],
[28.63673,77.23102],[28.63824,77.23116],[28.63985,77.23134],[28.64039,77.23144],
[28.64039,77.23144],[28.64049,77.23147],[28.64038,77.23156],[28.64022,77.23153],
[28.64022,77.23153]]

coords_3=[[28.63216,77.21864],[28.63225,77.21858],[28.63237,77.21851],
[28.6325,77.21846],[28.63262,77.21843],[28.63279,77.21841],[28.63295,77.21843],
[28.63311,77.21846],[28.63326,77.21853],[28.63341,77.21862],[28.63341,77.21862],
[28.63394,77.21768],[28.6342,77.21722],[28.63425,77.21713],[28.6343,77.21704],
[28.6343,77.21704],[28.63432,77.21686],[28.63495,77.21581],[28.6357,77.21452],
[28.63745,77.21151],[28.63951,77.20794],[28.63956,77.20787],[28.63971,77.20763],
[28.64055,77.2061],[28.64096,77.20533],[28.64106,77.20519],[28.64121,77.20486],
[28.64132,77.20468],[28.64179,77.20418],[28.64222,77.20381],[28.64254,77.20346],
[28.64267,77.20333],[28.64277,77.2032],[28.64277,77.2032],[28.64281,77.20314],
[28.64285,77.20309],[28.64291,77.20306],[28.64297,77.20304],[28.64303,77.20303],
[28.64309,77.20304],[28.64315,77.20306],[28.6432,77.2031],[28.64324,77.20314],
,

```
[28.64327,77.2032],[28.64327,77.2032],[28.64341,77.20336],[28.64375,77.20351]
',
[28.64406,77.20367],[28.64495,77.20426],[28.6459,77.20495],[28.64612,77.2051]
',
[28.64612,77.2051],[28.64629,77.2051],[28.64639,77.20508],[28.64648,77.20506]
',
[28.64657,77.20497],[28.64659,77.20495],[28.64659,77.20495]]
```

Chúng tôi sẽ sử dụng các vĩ độ và kinh độ này để vẽ đường đi cho ba phương tiện. Đây là cách chúng tôi sẽ làm điều đó:

Tạo bản đồ cơ sở

Tạo một nhóm tính năng cho mỗi phương tiện

Sau đó, tạo đường dẫn và thêm chúng vào các nhóm tính năng

Cuối cùng, thêm các nhóm tính năng này và kiểm soát lớp vào bản đồ

```
fig5=Figure(height=550,width=750)
m5=folium.Map(location=[28.644800,
77.216721],tiles='cartodbpositron',zoom_start=14)
fig5.add_child(m5)
```

Nhóm tính năng là thứ mà chúng ta có thể đặt mọi thứ và coi chúng như một lớp duy nhất. Nó có thể được tạo bằng cách sử dụng lớp `folium.FeatureGroup()`. Chúng tôi sẽ đặt đường dẫn của mình trong các nhóm tính năng này.

Tiếp theo, tôi sẽ vẽ đường đi của các phương tiện và thêm chúng vào các nhóm đối tượng địa lý. Đối với điều này, tôi sẽ sử dụng lớp `PolyLine` từ `vector_layers` của `Folium`. `vector_layers` bao bọc các tính năng `Polyline`, `Polygon`, `Rectangle`, `Circle` và `CircleMarker` của `Leaflet.js` được sử dụng để tạo các hình dạng khác nhau trên bản đồ.

Lớp `Polyline` được sử dụng để tạo lớp phủ `polyline` trên bản đồ và chúng tôi sẽ làm như vậy với nó. Nó lấy mảng vĩ độ và kinh độ, cửa sổ bật lên, chú giải công cụ, màu sắc và trọng lượng làm đối số và vẽ một đường đa giác trên bản đồ.

Trọng lượng xác định độ dày hay mỏng của `polyline`.

```
# Creating feature groups
f1=folium.FeatureGroup("Vehicle 1")
f2=folium.FeatureGroup("Vehicle 2")
f3=folium.FeatureGroup("Vehicle 3")

# Adding lines to the different feature groups
line_1=folium.vector_layers.PolyLine(coords_1,popup='<b>Path of
Vehicle_1</b>',tooltip='Vehicle_1',color='blue',weight=10).add_to(f1)
line_2=folium.vector_layers.PolyLine(coords_2,popup='<b>Path of
Vehicle_2</b>',tooltip='Vehicle_2',color='red',weight=10).add_to(f2)
line_3=folium.vector_layers.PolyLine(coords_3,popup='<b>Path of
Vehicle_3</b>',tooltip='Vehicle_3',color='green',weight=10).add_to(f3)
```

Tiếp theo, chúng tôi sẽ thêm các nhóm đối tượng này vào bản đồ cơ sở của mình và chúng tôi cũng sẽ thêm kiểm soát lớp để kiểm soát các lớp:

```
f1.add_to(m5)
f2.add_to(m5)
f3.add_to(m5)
folium.LayerControl().add_to(m5)
m5
```



Chúng ta có thể thấy rõ đường đi của cả ba phương tiện. Lưu ý ở góc trên bên phải, chúng ta có tùy chọn điều khiển lớp mà qua đó chúng ta có thể chuyển đổi giữa các lớp khác nhau. Chúng ta có thể xem thông tin về từng đường dẫn bằng cách nhấp vào nó. Bạn cũng có thể thử các chức năng khác của `vector_layers` và tạo các lớp phủ khác nhau trên bản đồ.

II. TẠO BẢN ĐỒ VỚI FOLIUM VÀ PYTHON

II.1 Import folium và các gói hữu ích khác

```
import folium
```

```
from pyproj import crs
```

```
import geopandas as gpd
```

```
import matplotlib.pyplot as plt
```

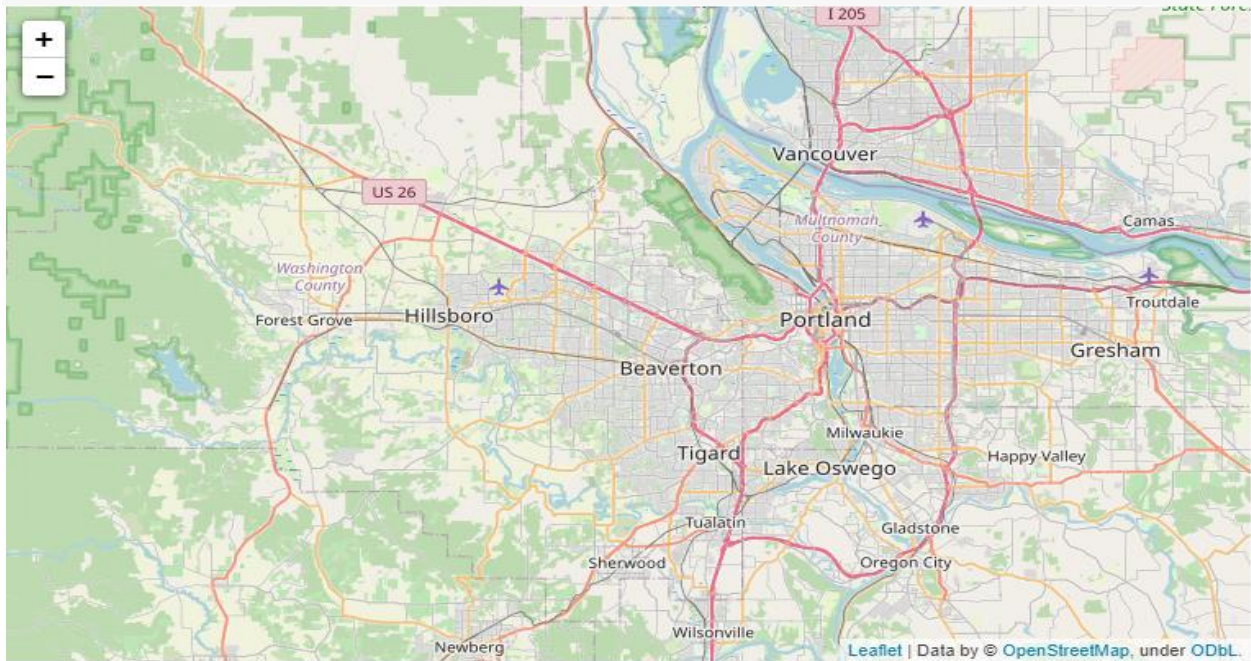
Chúng tôi sẽ bắt đầu bằng cách tạo một bản đồ web tương tác đơn giản mà không có bất kỳ dữ liệu nào trên đó. Chúng tôi chỉ hình dung OpenStreetMap trên một vị trí cụ thể trên thế giới.

Điều đầu tiên chúng ta cần làm là tạo một cá thể Bản đồ và xác định vị trí để phóng to dữ liệu:

```
# Create a Map instance
```

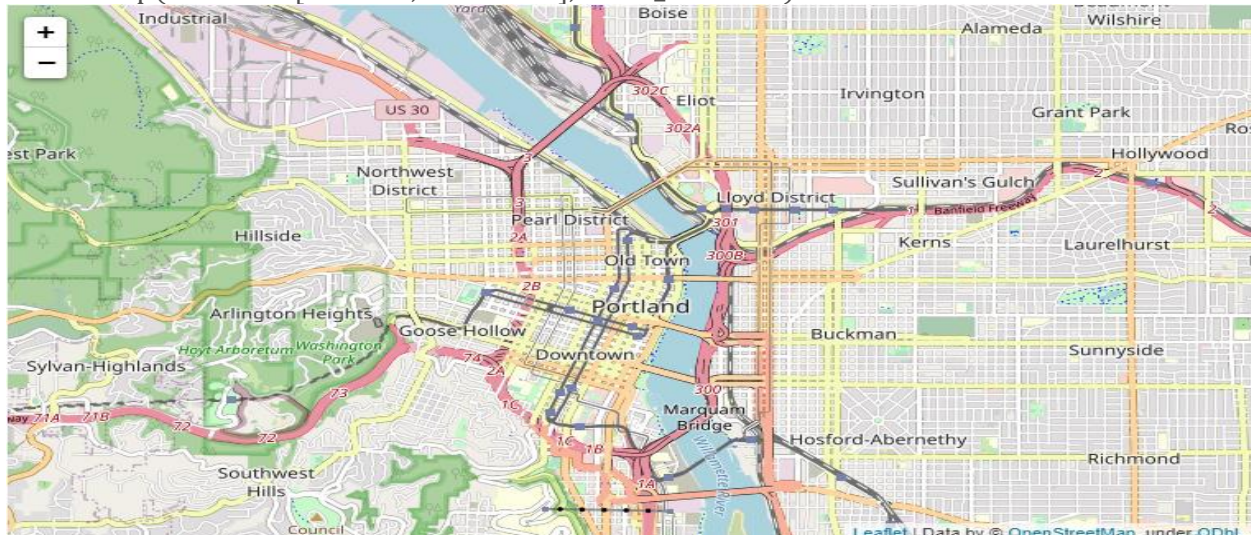
```
m = folium.Map(location=[60.25, 24.8], zoom_start=10, control_scale=True)
```

```
m
```



Các ô mặc định được đặt thành OpenStreetMap, nhưng Stamen Terrain, Stamen Toner, Mapbox Bright, Mapbox Control Room và nhiều ô khác được tích hợp sẵn.

```
folium.Map(location=[45.5236, -122.6750], zoom_start=13)
```



Có thể sử dụng các ô tùy chỉnh của Cloudmade hoặc Mapbox – chỉ cần chuyển khóa của bạn đến từ khóa API_key:

```
folium.Map(location=[45.5236, -122.6750],
            tiles='Mapbox',
            API_key='your.API.key')
```

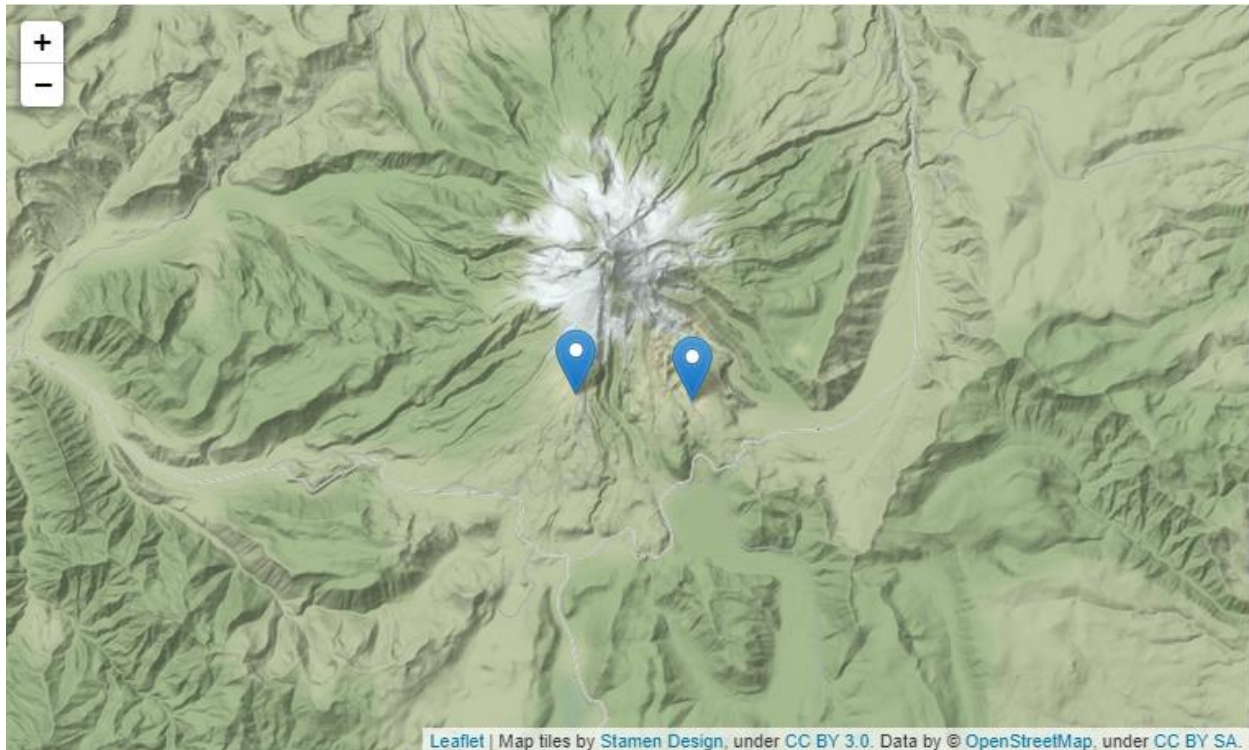
Folium hỗ trợ chuyển bất kỳ ô tùy chỉnh tương thích nào của leaflet.js:

```
folium.Map(location=[45.372, -121.6972],
            zoom_start=12,
            tiles='http://{s}.tiles.yourtiles.com/{z}/{x}/{y}.png',
            attr='My Data Attribution')
```

II.2 Điểm đánh dấu:

Có rất nhiều loại điểm đánh dấu, bắt đầu bằng điểm đánh dấu vị trí kiểu Leaflet đơn giản với HTML của số popup và công cụ chú giải.

```
m = folium.Map(location=[45.372, -121.6972], zoom_start=12, tiles="Stamen Terrain")
tooltip = "Click me!"
folium.Marker(
    [45.3288, -121.6625], popup="<i>Mt. Hood Meadows</i>", tooltip=tooltip
).add_to(m)
folium.Marker(
    [45.3311, -121.7113], popup="<b>Timberline Lodge</b>", tooltip=tooltip
).add_to(m)
m
```



Có sẵn hỗ trợ cho màu sắc và các loại biểu tượng điểm đánh dấu từ bootstrap.

```
m = folium.Map(location=[45.372, -121.6972], zoom_start=12, tiles="Stamen Terrain")

folium.Marker(
```



```

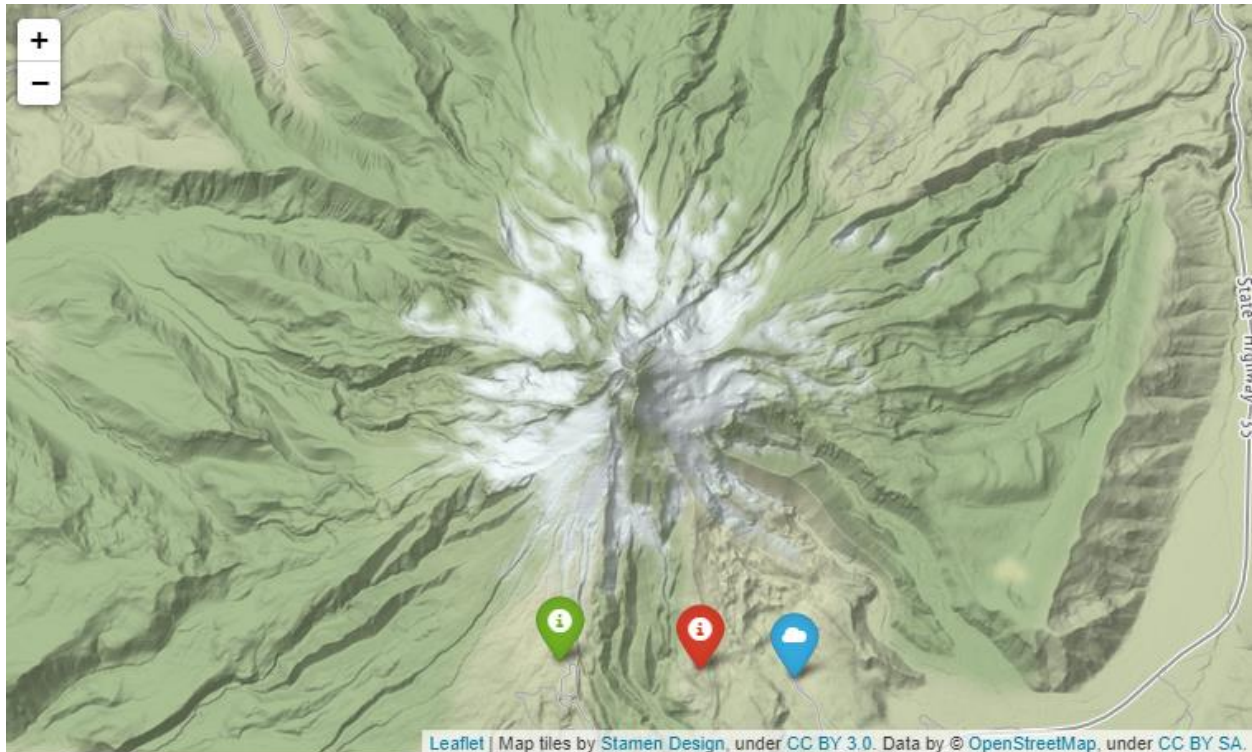
location=[45.3288, -121.6625],
popup="Mt. Hood Meadows",
icon=folium.Icon(icon="cloud"),
).add_to(m)

folium.Marker(
location=[45.3311, -121.7113],
popup="Timberline Lodge",
icon=folium.Icon(color="green"),
).add_to(m)

folium.Marker(
location=[45.3300, -121.6823],
popup="Some Other Location",
icon=folium.Icon(color="red", icon="info-sign"),
).add_to(m)

```

m



Leaflet's Circle và CircleMarker, được triển khai để phản ánh bán kính theo đơn vị mét và pixel tương ứng, có sẵn dưới dạng tính năng. Xem các features.py để có thêm các tùy chọn.

```
m = folium.Map(location=[45.5236, -122.6750], tiles="Stamen Toner", zoom_start=13)
```

```

folium.Circle(
radius=100,
location=[45.5244, -122.6699],
popup="The Waterfront",
color="crimson",
fill=False,

```

```
).add_to(m)
```

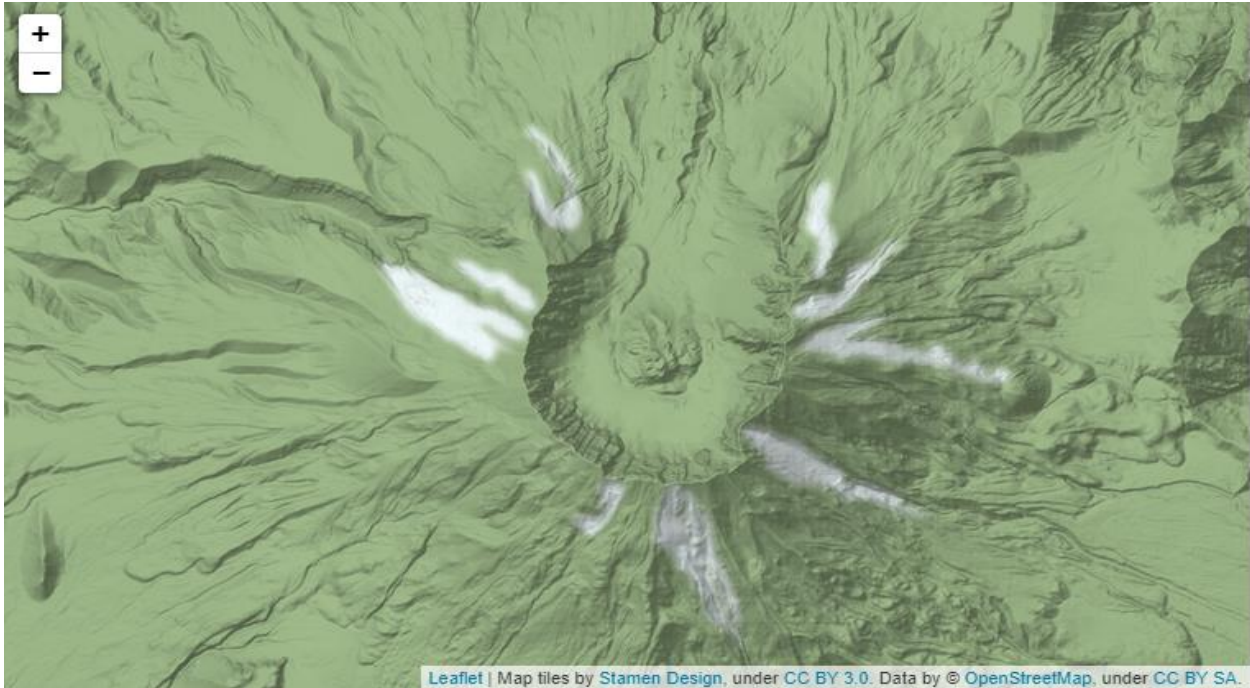
```
folium.CircleMarker(  
    location=[45.5215, -122.6261],  
    radius=50,  
    popup="Laurelhurst Park",  
    color="#3186cc",  
    fill=True,  
    fill_color="#3186cc",  
).add_to(m)
```



Một chức năng tiện lợi để kích hoạt cửa sổ vĩ độ / kinh độ. Điều này có thể giúp người dùng tìm một vị trí bằng cách duyệt qua bản đồ một cách tương tác.

```
m = folium.Map(location=[46.1991, -122.1889], tiles="Stamen Terrain", zoom_start=13)  
m.add_child(folium.LatLngPopup())
```

```
m
```

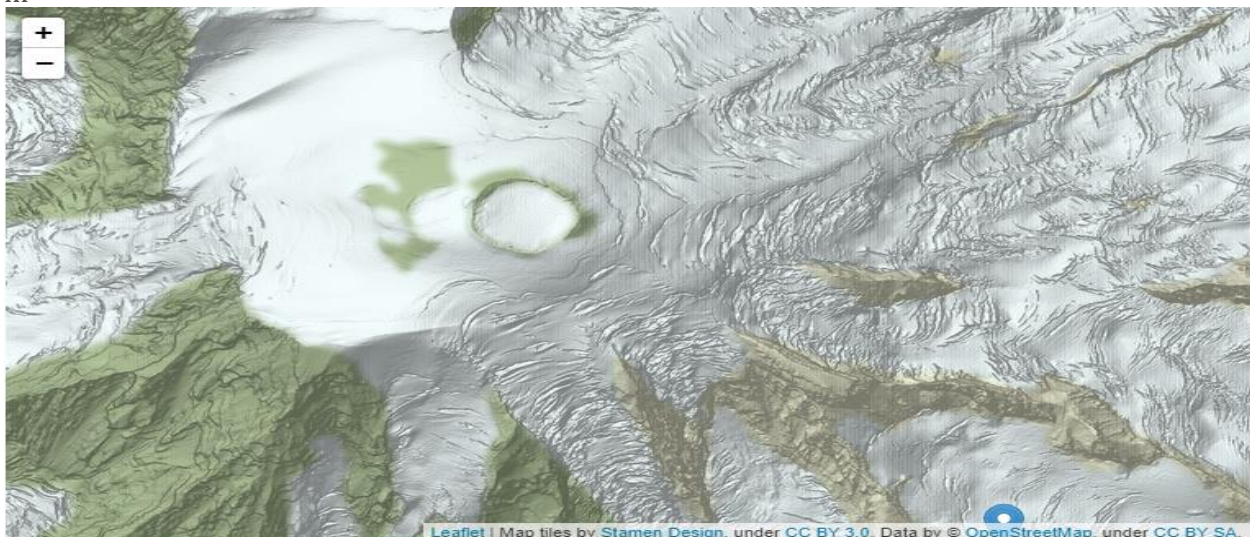
Chức năng nhấp để đánh dấu cho vị trí nhanh chóng của điểm đánh dấu:

```
m = folium.Map(location=[46.8527, -121.7649], tiles="Stamen Terrain", zoom_start=13)
```

```
folium.Marker([46.8354, -121.7325], popup="Camp Muir").add_to(m)
```

```
m.add_child(folium.ClickForMarker(popup="Waypoint"))
```

m



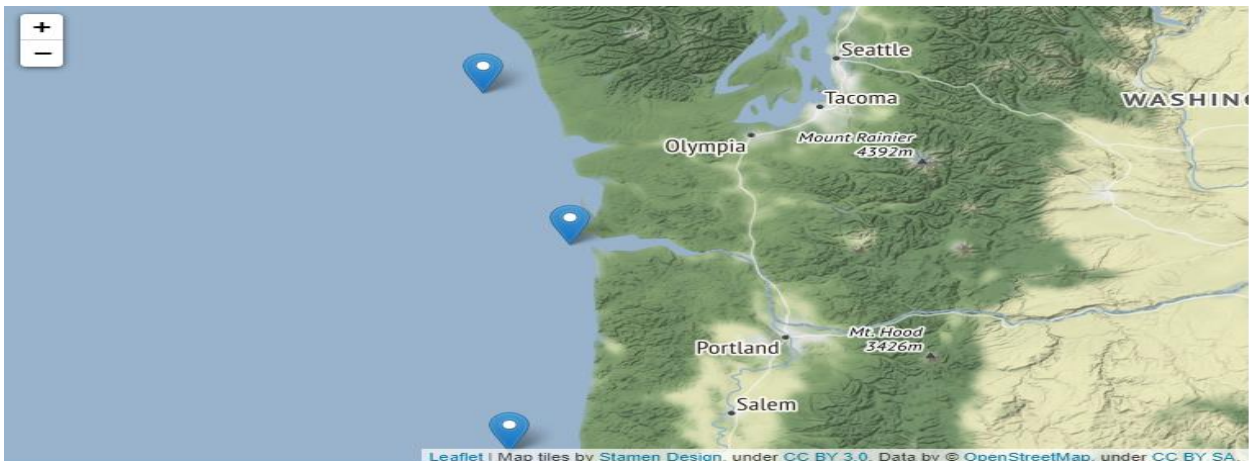
II.3 Kiểu đánh dấu Vincent/Vega and Altair/VegaLite

Folium cho phép chuyển bất kỳ đối tượng HTML nào dưới dạng cửa sổ popup, bao gồm `bokeh` <https://bokeh.pydata.org/en/latest/> `__` plot, nhưng có một hỗ trợ tích hợp cho hình ảnh hóa vincent và altair cho bất kỳ loại điểm đánh dấu nào, với hình ảnh hóa dưới dạng cửa sổ popup.

```
import json
```

import requests

```
url = (  
    "https://raw.githubusercontent.com/python-visualization/folium/master/examples/data"  
)  
vis1 = json.loads(requests.get(f"{url}/vis1.json").text)  
vis2 = json.loads(requests.get(f"{url}/vis2.json").text)  
vis3 = json.loads(requests.get(f"{url}/vis3.json").text)  
m = folium.Map(location=[46.3014, -123.7390], zoom_start=7, tiles="Stamen Terrain")  
folium.Marker(  
    location=[47.3489, -124.708],  
    popup=folium.Popup(max_width=450).add_child(  
        folium.Vega(vis1, width=450, height=250)  
    ),  
)  
.add_to(m)  
folium.Marker(  
    location=[44.639, -124.5339],  
    popup=folium.Popup(max_width=450).add_child(  
        folium.Vega(vis2, width=450, height=250)  
    ),  
)  
.add_to(m)  
folium.Marker(  
    location=[46.216, -124.1280],  
    popup=folium.Popup(max_width=450).add_child(  
        folium.Vega(vis3, width=450, height=250)  
    ),  
)  
.add_to(m)  
m
```



II.4 Lớp phủ GeoJSON / TopoJSON

Cả hai lớp GeoJSON và TopoJSON đều có thể được chuyển tới bản đồ dưới dạng lớp phủ và nhiều lớp có thể được hiển thị trên cùng một bản đồ:

```
url = (  
    "https://raw.githubusercontent.com/python-visualization/folium/master/examples/data"  
)  
antarctic_ice_edge = f"{url}/antarctic_ice_edge.json"  
antarctic_ice_shelf_topo = f"{url}/antarctic_ice_shelf_topo.json"  
m = folium.Map(  
    location=[60, -120],  
    zoom_start=2,  
    tiles="Stamen Terrain",  
    data=[  
        (antarctic_ice_edge, "Ice Edge"),  
        (antarctic_ice_shelf_topo, "Ice Shelf Topo")  
    ]  
)  
m
```

```

location=[-59.1759, -11.6016],
tiles="cartodbpositron",
zoom_start=2,
)
folium.GeoJson(antarctic_ice_edge, name="geojson").add_to(m)
folium.TopoJson(
    json.loads(requests.get(antarctic_ice_shelf_topo).text),
    "objects.antarctic_ice_shelf",
    name="topojson",
).add_to(m)

folium.LayerControl().add_to(m)

```

m



II.5 Bản đồ Choropleth

Có thể dễ dàng tạo Choropleth bằng cách liên kết dữ liệu giữa các hình học Pandas DataFrames / Series và Geo/TopoJSON. Các lược đồ màu tuần tự của Color Brewer được tích hợp sẵn trong thư viện và có thể được chuyển qua để nhanh chóng hình dung các kết hợp khác nhau.

```
import pandas as pd
```

```

url = (
    "https://raw.githubusercontent.com/python-visualization/folium/master/examples/data"
)
state_geo = f"{url}/us-states.json"
state_unemployment = f"{url}/US_Unemployment_Oct2012.csv"
state_data = pd.read_csv(state_unemployment)
m = folium.Map(location=[48, -102], zoom_start=3)

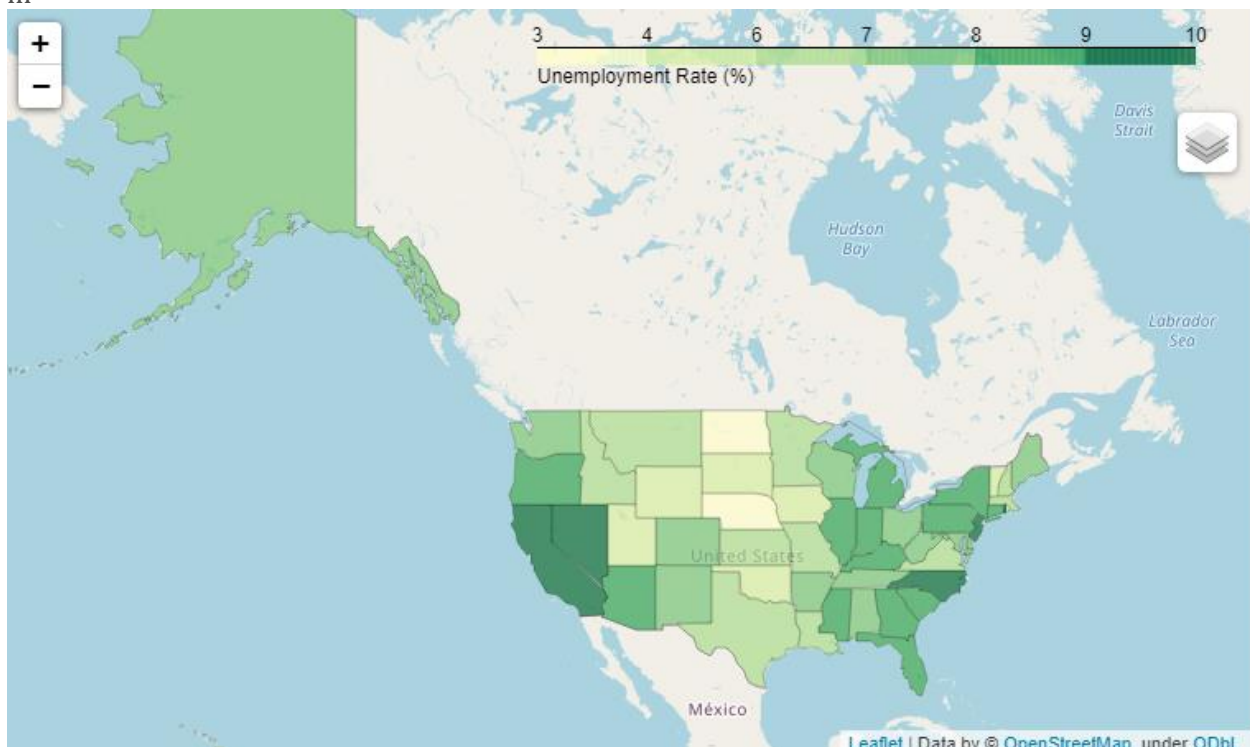
```



```

folium.Choropleth(
    geo_data=state_geo,
    name="choropleth",
    data=state_data,
    columns=["State", "Unemployment"],
    key_on="feature.id",
    fill_color="YlGn",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Unemployment Rate (%)",
).add_to(m)
folium.LayerControl().add_to(m)
m

```



Chú giải ở phía trên bên phải được tạo tự động cho các giá trị của bạn bằng cách sử dụng 6 bins có cùng kích thước:

```
bins = list(state_data["Unemployment"].quantile([0, 0.25, 0.5, 0.75, 1]))
```

```
m = folium.Map(location=[48, -102], zoom_start=3)
```

```

folium.Choropleth(
    geo_data=state_geo,
    data=state_data,
    columns=["State", "Unemployment"],
    key_on="feature.id",
    fill_color="BuPu",
    fill_opacity=0.7,
    line_opacity=0.5,
    legend_name="Unemployment Rate (%)",

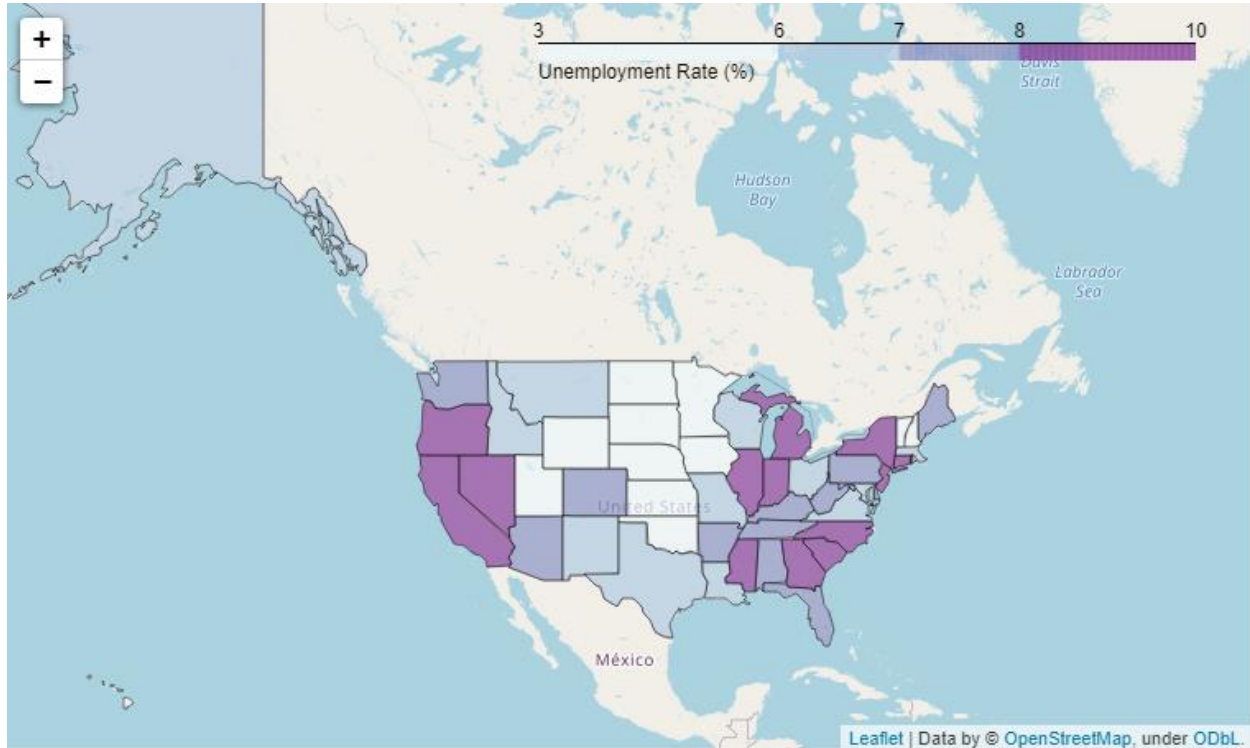
```

```

bins=bins,
reset=True,
).add_to(m)

```

m



II.6 Chức năng tạo kiểu:

Các tính năng của GeoJson và TopoJson chấp nhận kiểu_ Chức năng để cho phép tiếp tục tối ưu hóa bản đồ. Hãy xem các ví dụ sử dụng bên dưới.

```
import branca
```

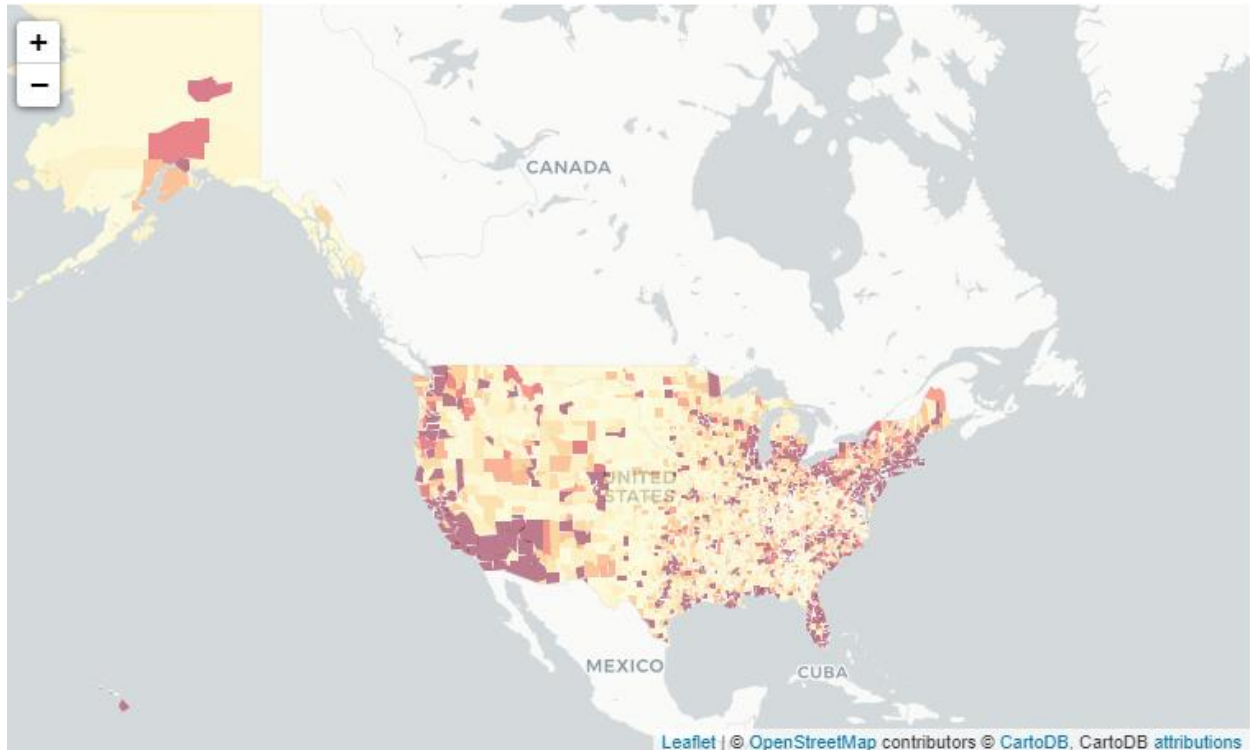
```

url = (
    "https://raw.githubusercontent.com/python-visualization/folium/master/examples/data"
)
county_data = f"{url}/us_county_data.csv"
county_geo = f"{url}/us_counties_20m_topo.json"
df = pd.read_csv(county_data, na_values=[""])
colorscale = branca.colormap.linear.YlOrRd_09.scale(0, 50e3)
employed_series = df.set_index("FIPS_Code")["Employed_2011"]
def style_function(feature):
    employed = employed_series.get(int(feature["id"][-5:]), None)
    return {
        "fillOpacity": 0.5,
        "weight": 0,
        "fillColor": "#black" if employed is None else colorscale(employed),
    }
m = folium.Map(location=[48, -102], tiles="cartodbpositron", zoom_start=3)
folium.TopoJson(
    json.loads(requests.get(county_geo).text),

```

```
"objects.us_counties_20m",
style_function=style_function,
).add_to(m)
```

m

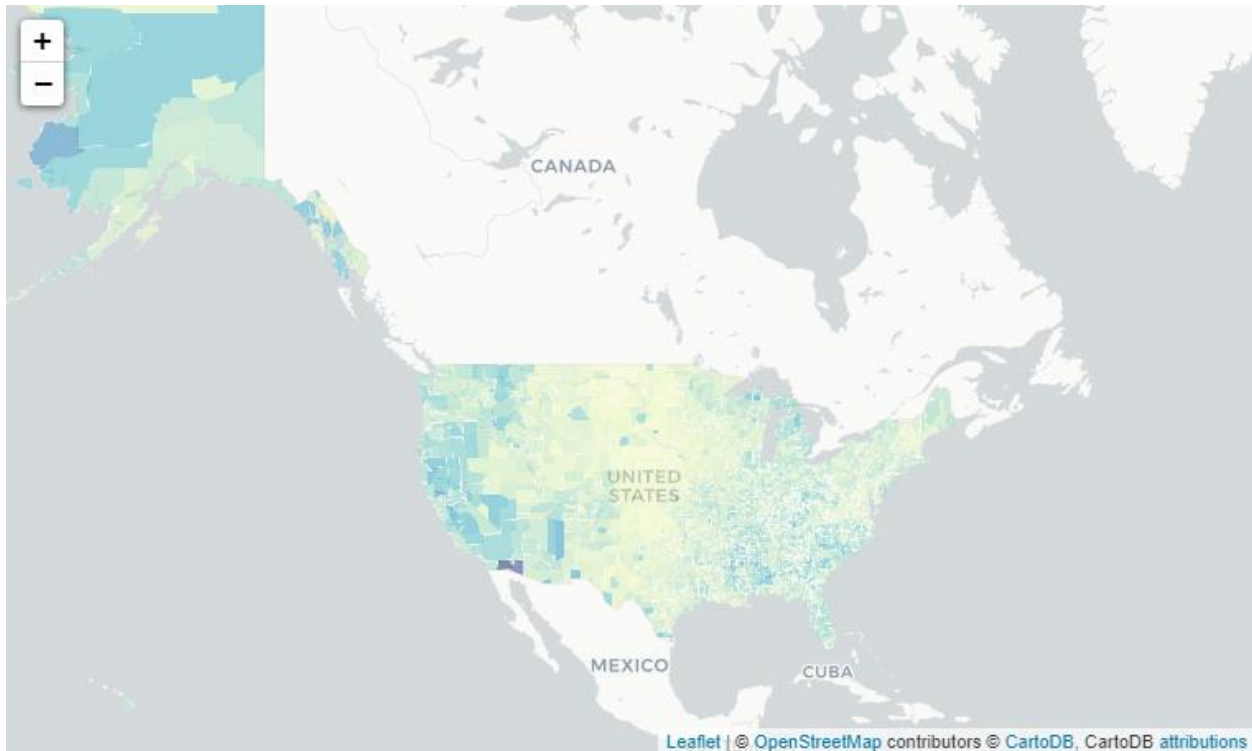


```
colorscale = branca.colormap.linear.YlGnBu_09.scale(0, 30)
employed_series = df.set_index("FIPS_Code")["Unemployment_rate_2011"]
def style_function(feature):
    employed = employed_series.get(int(feature["id"][-5:]), None)
    return {
        "fillOpacity": 0.5,
        "weight": 0,
        "fillColor": "#black" if employed is None else colorscale(employed),
    }
```

```
m = folium.Map(location=[48, -102], tiles="cartodbpositron", zoom_start=3)
```

```
folium.TopoJson(
    json.loads(requests.get(county_geo).text),
    "objects.us_counties_20m",
    style_function=style_function,
).add_to(m)
```

m



```
colorscale = branca.colormap.linear.PuRd_09.scale(0, 100000)
```

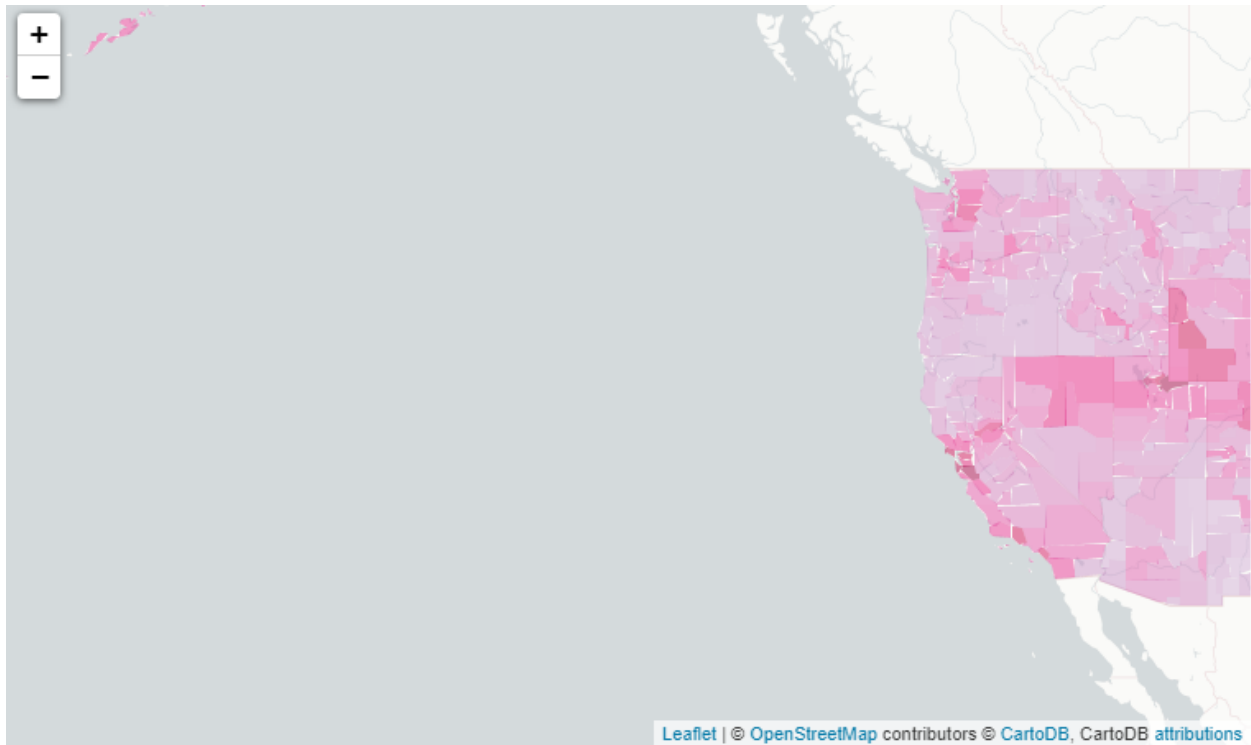
```
employed_series = df.set_index("FIPS_Code")["Median_Household_Income_2011"].dropna()
```

```
def style_function(feature):
    employed = employed_series.get(int(feature["id"][-5:]), None)
    return {
        "fillOpacity": 0.5,
        "weight": 0,
        "fillColor": "#black" if employed is None else colorscale(employed),
    }
```

```
m = folium.Map(location=[48, -102], tiles="cartodbpositron", zoom_start=3)
```

```
folium.TopoJson(
    json.loads(requests.get(county_geo).text),
    "objects.us_counties_20m",
    style_function=style_function,
).add_to(m)
```

```
m
```



III. KẾT LUẬN

Folium giúp dễ dàng hình dung dữ liệu được thao tác bằng Python trên bản đồ tương tác Leaflet. Nó cho phép cả hai liên kết dữ liệu với bản đồ để trực quan hóa choropleth cũng như chuyển các hình ảnh hóa Vincent / Vega làm điểm đánh dấu trên bản đồ.

Thư viện có một số tập hợp sẵn từ OpenStreetMap, Mapbox và Stamen, đồng thời hỗ trợ các tập xếp tùy chỉnh với các khóa API Mapbox hoặc Cloudmade. Folium hỗ trợ cả lớp phủ GeoJSON và TopoJSON, cũng như ràng buộc dữ liệu vào các lớp phủ đó để tạo bản đồ choropleth với các lược đồ màu color-brewer.

TÀI LIỆU THAM KHẢO

- [1] <https://python-visualization.github.io/folium/quickstart.html>
- [2] <https://www.analyticsvidhya.com/blog/2020/06/guide-geospatial-analysis-folium-python/>
- [3] https://autogis-site.readthedocs.io/en/latest/notebooks/L5/02_interactive-map-folium.html
- [4] <https://python-visualization.github.io/folium/modules.html>