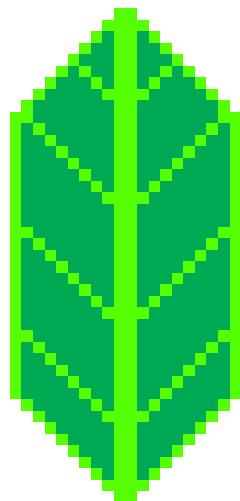


FOLIUM WORKBOOK



By: Anya Dougherty, Jenna Lobasso, and Lauren Hickman

Python data, leaflet.js maps

Folium combines Python's data manipulation with Leaflet.js's mapping to let you process your data in Python and visualize it on a Leaflet map.

Concepts

Folium simplifies the process of visualizing Python-manipulated data on interactive Leaflet maps. It facilitates binding data to maps for choropleth representations and allows the inclusion of diverse visualizations like vectors, rasters, and HTML markers. With built-in support for various tilesets like OpenStreetMap and Mapbox, as well as custom options, Folium offers versatility. It also supports overlays such as Image, Video, GeoJSON, and TopoJSON, with several pre-built vector layers available.

Installation

Folium can be installed with anaconda using:

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

Folium can be installed with vscode using:

```
$ pip install folium
```

Dependencies

Folium has the following dependencies, all of which are installed automatically with the installation commands above:

- branca

- Jinja2
- Numpy
- Requests

We will be using Requests for our tutorial.

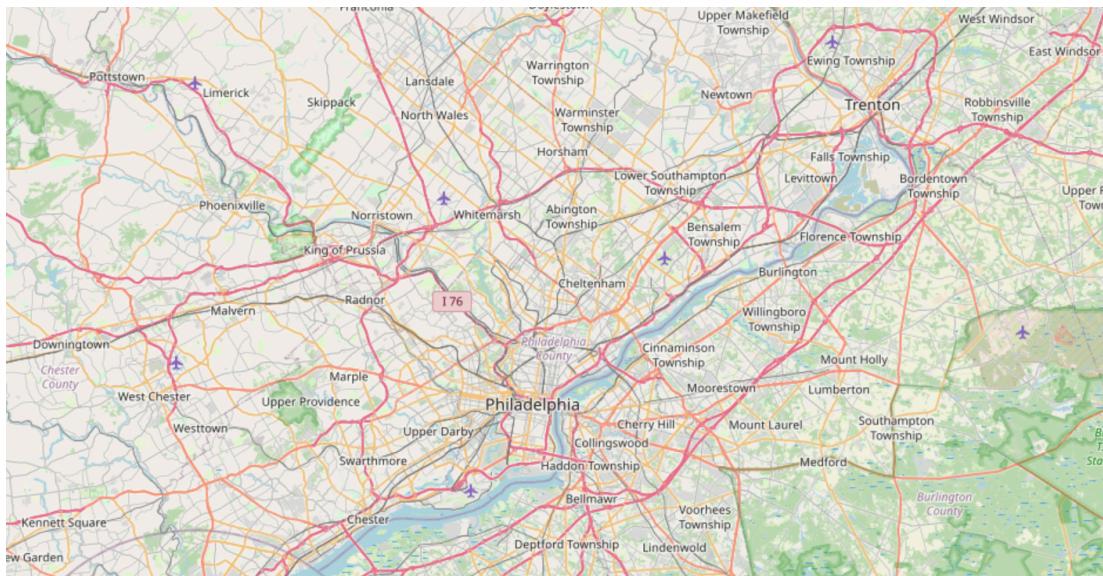
Basic Map

A basic map can be created by creating a variable and using the folium.Map function.

Inside there is a required `location=`clause, which can be fulfilled with coordinates. The basic code can be found below:

```
Import folium

m = folium.Map(location=(latitude, longitude))
```



Basic Map without customization of the Wissahickon Trail Coordinates

Further customization can be made to the following variables:

- Zoom
- Tileset
- Markers

Zoom

Zoom can be specified with the zoom start clause which will identify at what point the map should begin. After the location has been specified, the `zoom_start=` code will be added and with a specified number for where the zoom should begin. This number is best figured out through trial and error. The `zoom_start=` clause is necessary if the user does not want the map to start at the world view and would prefer a more local level appearance.

```
m = folium.Map(location=(latitude, longitude), zoom_start=##)
```



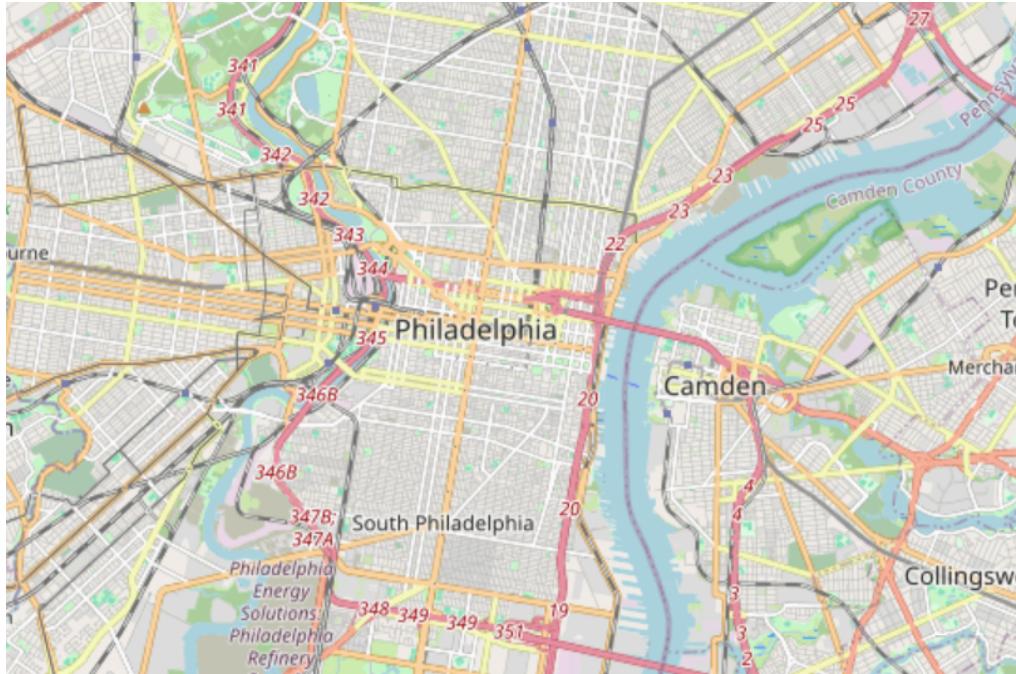
Previous map with zoom_start customization

Tileset

Tileset customization adjusts the base map that the data will appear on top of. The default tileset is the OpenStreetMap tiles(pictured below).

The folium package comes with some included tilesets such as:

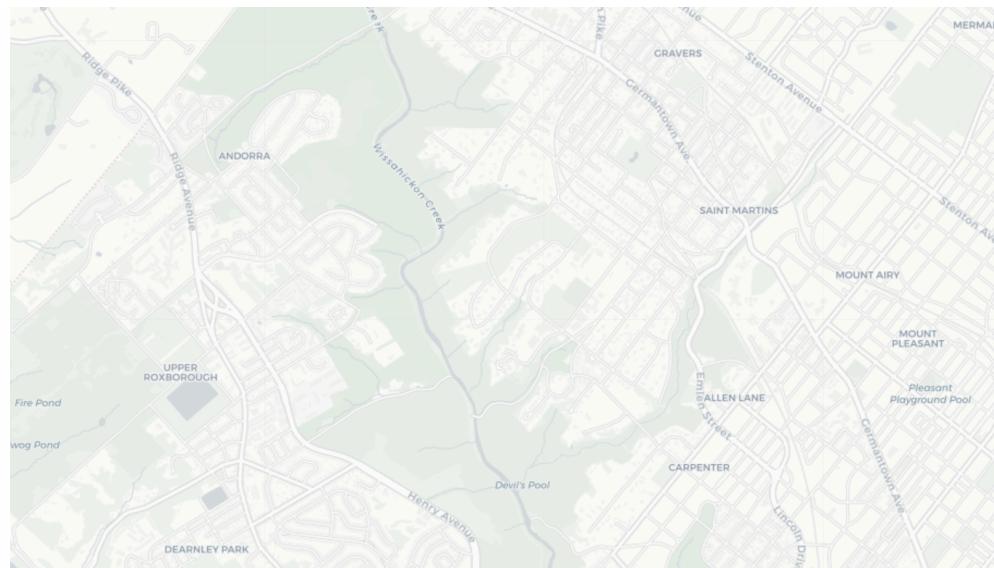
- Carto DB Positron
- Carto DB Dark Matter
- USGS US Topo
- The full list of tilesets that are included in the folium module can be found at
[https://leaflet-extras.github.io/leaflet-providers/preview/.](https://leaflet-extras.github.io/leaflet-providers/preview/)



A screenshot of the OpenStreetMap website zoomed in on Philadelphia

The tileset can be customized using the following code:

```
m = folium.Map(location=(latitude, longitude), tiles="Title of tileset")
```



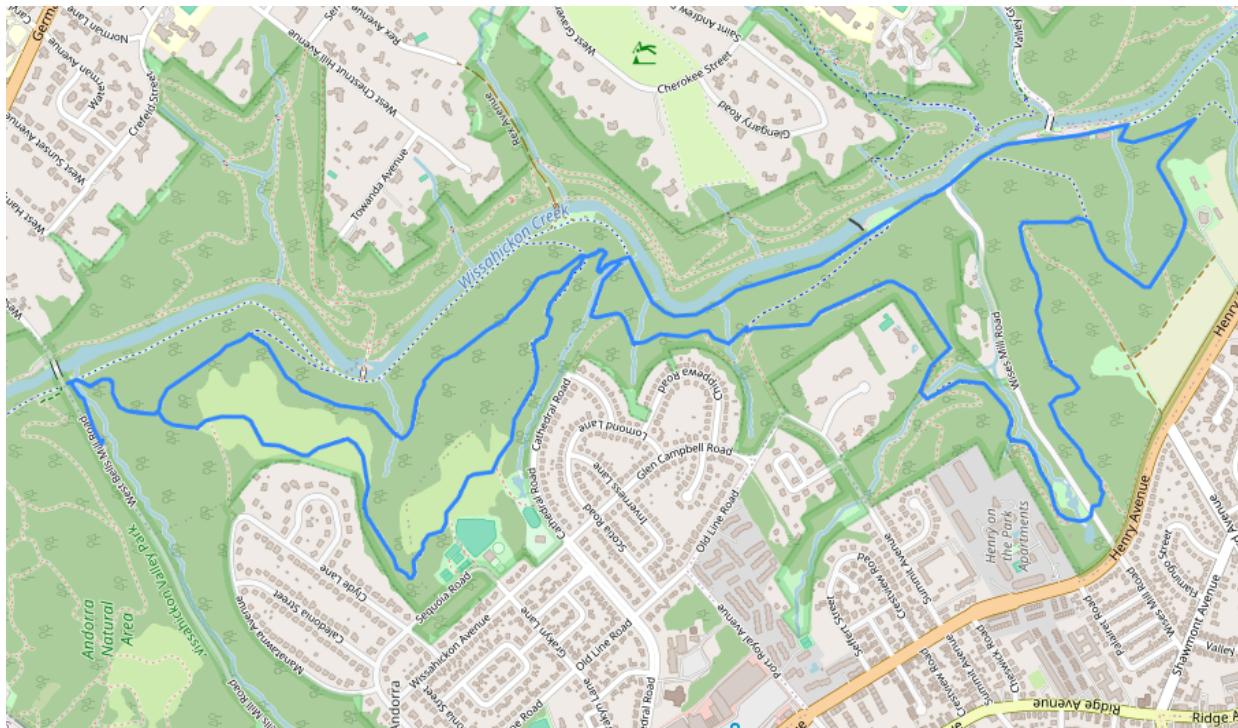
Previous map with the Carto DB Positron tiles

GeoJSON/TopoJSON overlays

Folium supports both GeoJSON and TopoJSON data in various formats, such as urls, file paths and dictionaries. A GeoJSON is a format for encoding geographic data and can include points, polygons, linestrings, multipoints, multipolygons, and multiline strings. A TopoJSON is an extension for GeoJSONs that contain topography information.

```
trails_file = ('your file path')  
f.GeoJson(trails_file).add_to(map)
```

The code above adds the Wissahickon Trail as a polyline to the map created previously.



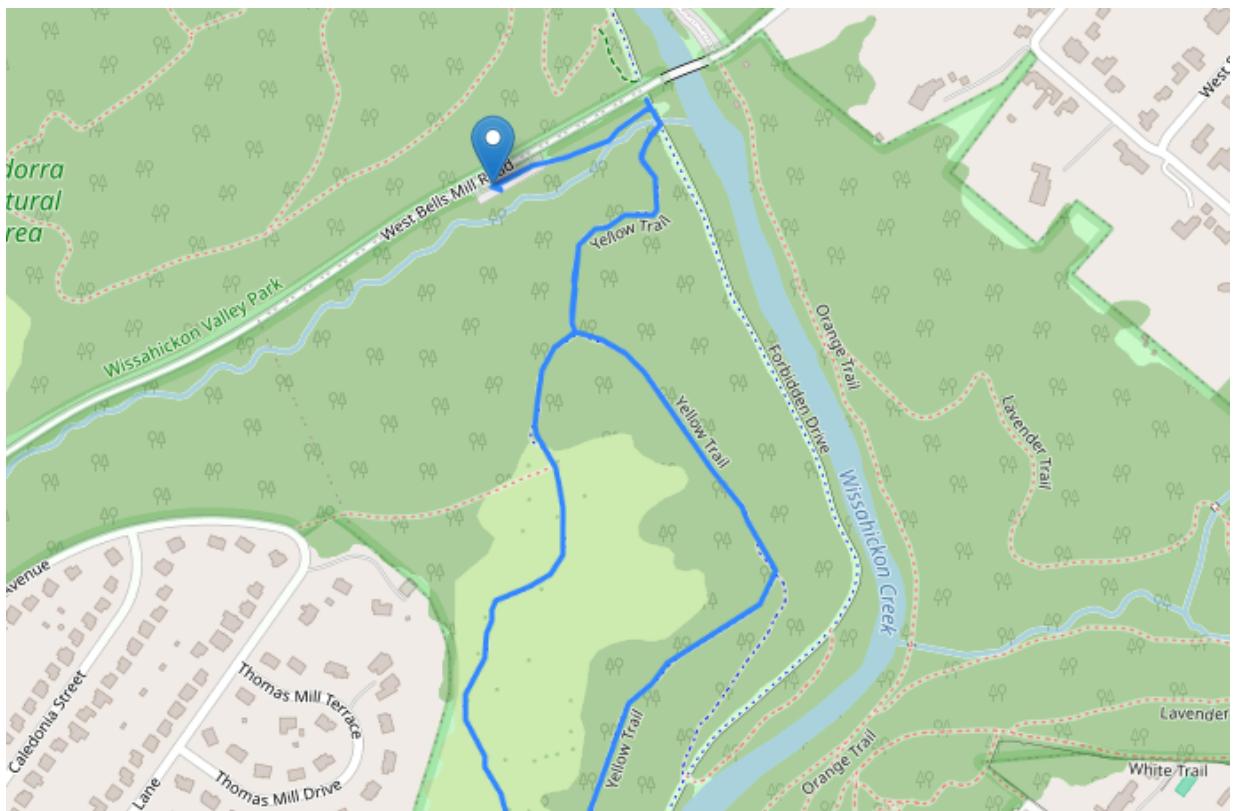
Wissahickon Trail

Adding markers and Popups

There are various marker types. You can add a popup and tooltip. You can also pick colors and icons .

Simple Marker

```
f.Marker([40.07817, -75.2278]).add_to(map)
```



Simple Marker on Wissahickon Trail

Custom Icon

```
icon_image = ('your file path')
```

```

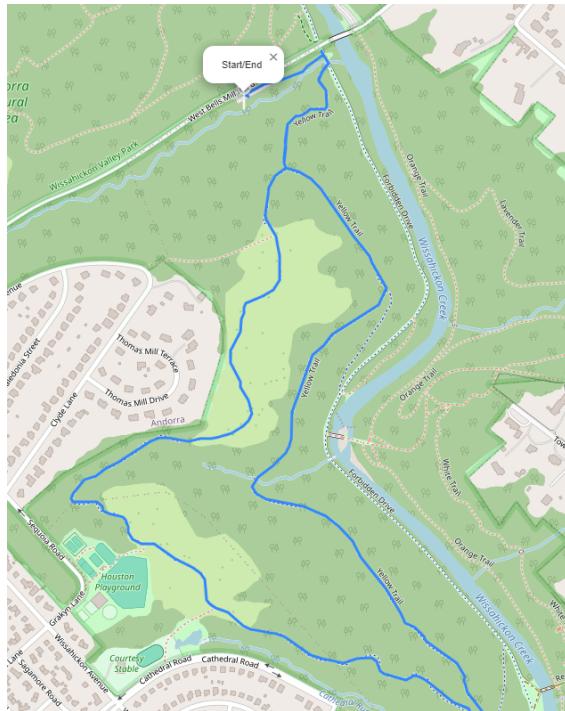
# create a variable for your icon

icon = f.CustomIcon(icon_image, icon_size=(38, 40))

# creating the custom icon marker

f.Marker(location=[40.07817, -75.2278], icon=icon,
popup="Start/End").add_to(map)

```



Start/End Marker

Choropleth maps

Choropleth can be created by binding the data between Pandas DataFrames/Series and Geo/TopoJSON geometries. The function `folium.Choropleth()` can be used to create these choropleth maps. The clause `geo_data=` is the most necessary element of

the choropleth map design and requires the geographic data (polygons are most effective) that will display the data.

```
import pandas

import folium

import requests

map2 = folium.Map([43, -100], zoom_start=4)

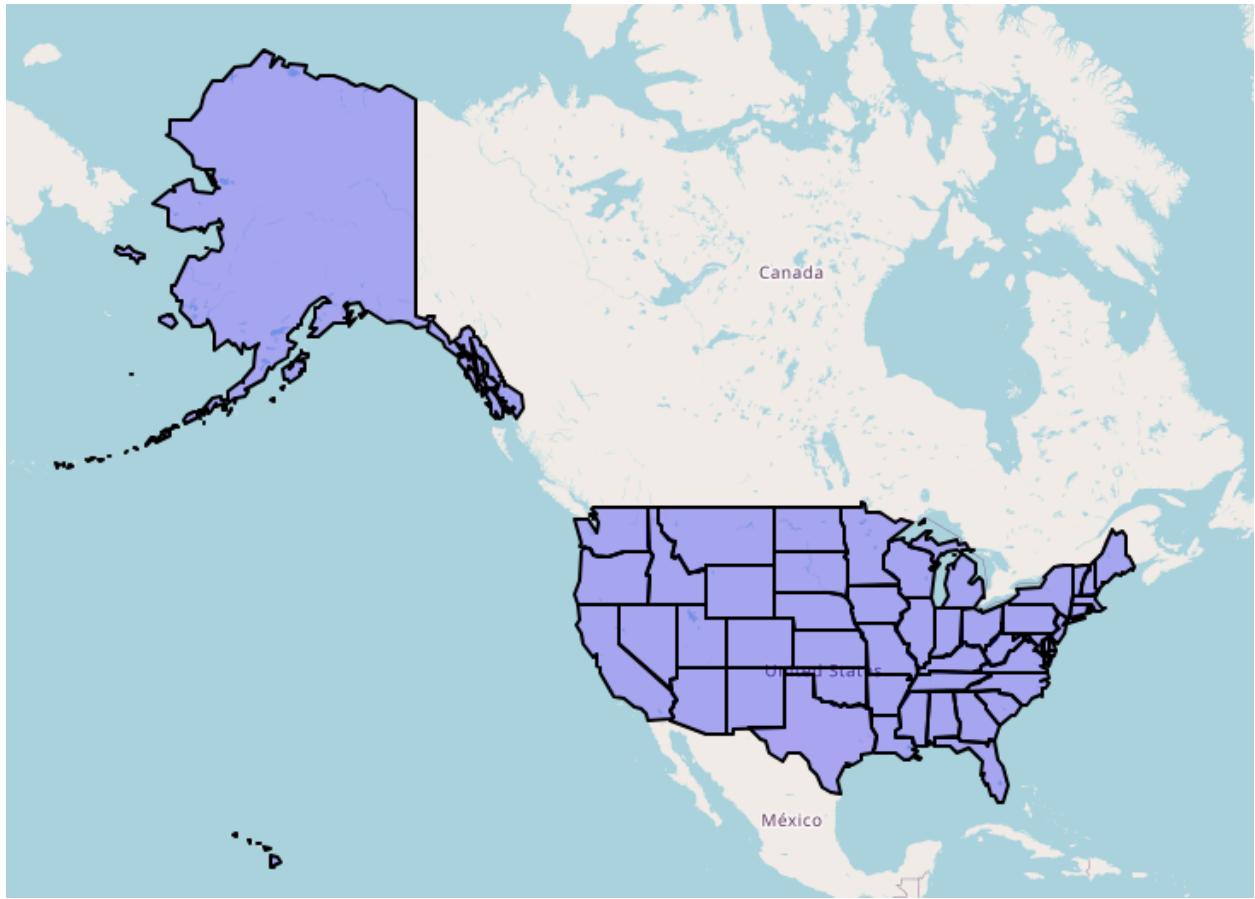
us_states = requests.get(

    "https://raw.githubusercontent.com/python-visualization/folium-exa
    mple-data/main/us_states.json"

).json()

folium.Choropleth(
    geo_data=us_states,
    fill_opacity=0.3,
    line_weight=2,
).add_to(map2)

map2.save('C:/folium/map2.html')
```

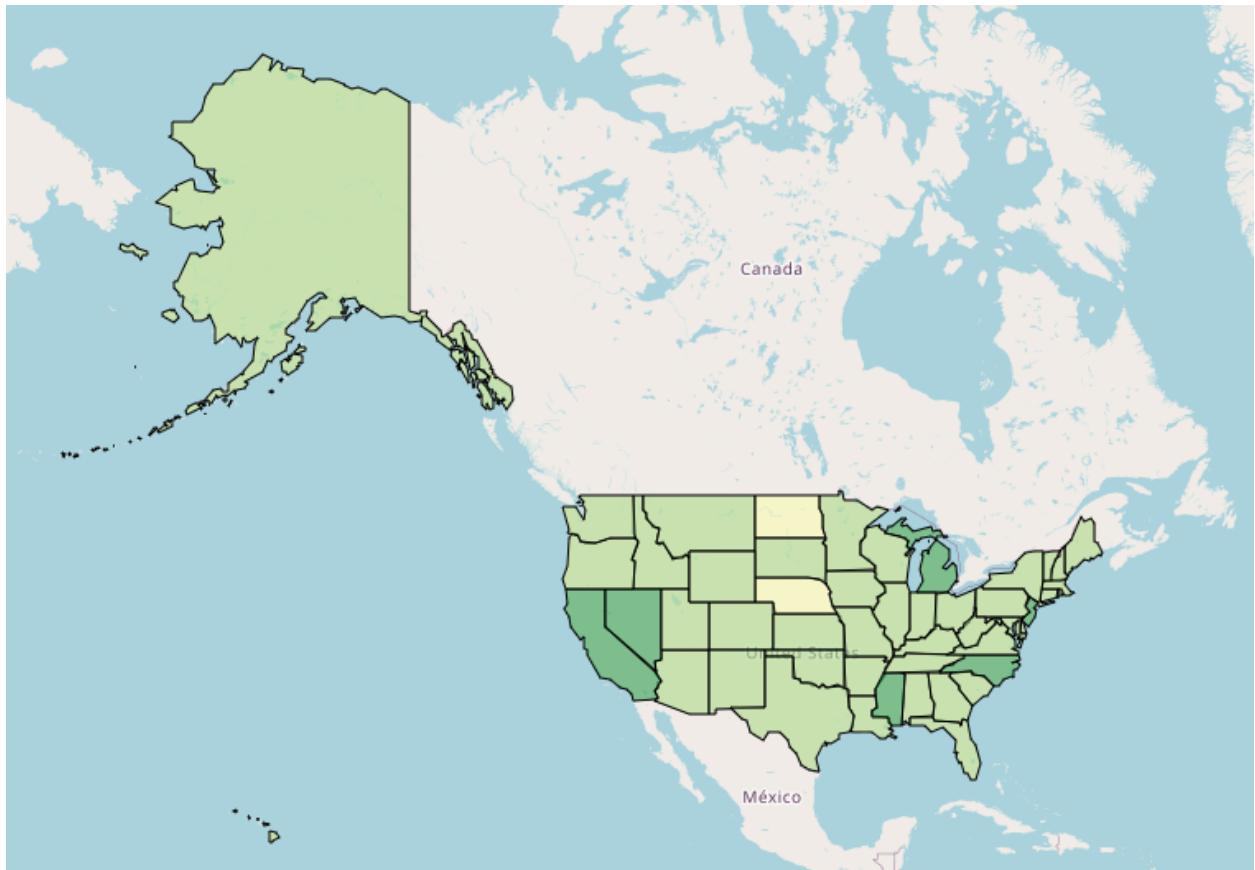


Choropleth map of the US

The next clauses that are necessary to make a choropleth map are the additions of the `data=` clause that will include the non-geographic data that will be the data displayed, in the example below that data is the unemployment rate per state. The `key_on=` clause is what connects the geographic and non-geographic data, this must be a column that is included in both the GeoJSON and the csv that are utilized in this example. Another helpful customization to make is the `bins=` clause which will decide how many natural breaks will be included in the choropleth map.

```
state_data = pandas.read_csv(  
    "https://raw.githubusercontent.com/python-visualization/folium-exa  
    mple-data/main/us_unemployment_oct_2012.csv")
```

```
)  
  
map3 = folium.Map([43, -100], zoom_start=4)  
  
folium.Choropleth(  
  
    geo_data=us_states,  
  
    data=state_data,  
  
    columns=["State", "Unemployment"],  
  
    key_on="feature.id",  
  
    nan_fill_color="grey", #makes N/A values different colors  
  
    nan_fill_opacity=0.4,  
  
    fill_color="YlGn", #changes fill colors  
  
    bins=[3, 4, 9, 11], #changes breaks  
) .add_to(map3)  
  
map3.save('C:/folium/map3.html')
```



Choropleth map of the US showing Unemployment