GeoSpatial Data Analysis Using Python

Fereshteh ASGARI

15 November 2018

PyParis

SystemX is one of eight Institutes for Research and Technology (IRT), an interdisciplinary thematic institute that conducts research and development projects at the international level and contributes to the engineering of initial and continuous training.

Overview

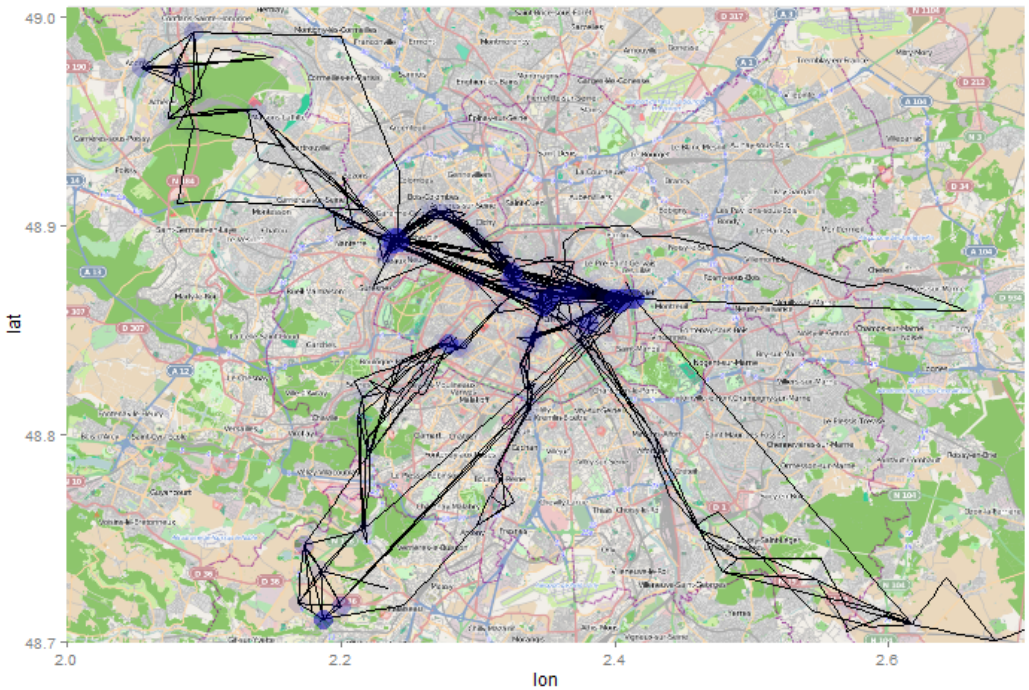
The presentation will cover these topics:

- Introduction
- Geospatial data (formats, import, etc)
- Hubway/BlueBike dataset
- Basic analysis with (pandas & numpy)
- Visualization (interactive maps with folium)

Requirements: jupyter notebook

Introduction

Difficulties of static maps:



I needed an interactive map, something I could open in my browser and zoom in to see more details.

Geospatial data

Geospatial data or spatial data is data that has a geographic aspect to it. Each record in this type of information has location information tied to them such as geographic data in the form of coordinates, address, city, or ZIP code

Types:

- Raster Data
- Vector Data

Geospatial data

Raster data: any type of digital image represented by reducible and enlargeable grids.
Raster graphics pixel as the smallest individual grid unit building block of an image are familiar for us.
Raster data type consists of rows and columns of cells, with each cell storing a single value. Examples of raster data: JPEG

Geospatial Data

Vector data: Geographical features are often expressed as vectors, by considering those features as geometrical shapes. Different geographical features are expressed by different types of geometry:

- Points
- Line
- Polygons

Example formats: Shapefile, GeoJson, etc.

Hubway Data

Bike sharing company in Boston metro area launched in July 2011.
By the end of the November 2012, the system had 105 stations and 1050 bikes.
November 2013 → 130 stations, 1200 bikes and 900k rides/year.
November 2015 → 155 stations, over 1500 bikes.
May 2017 → 180 stations ,1600 bikes and 5.3 rides/year.

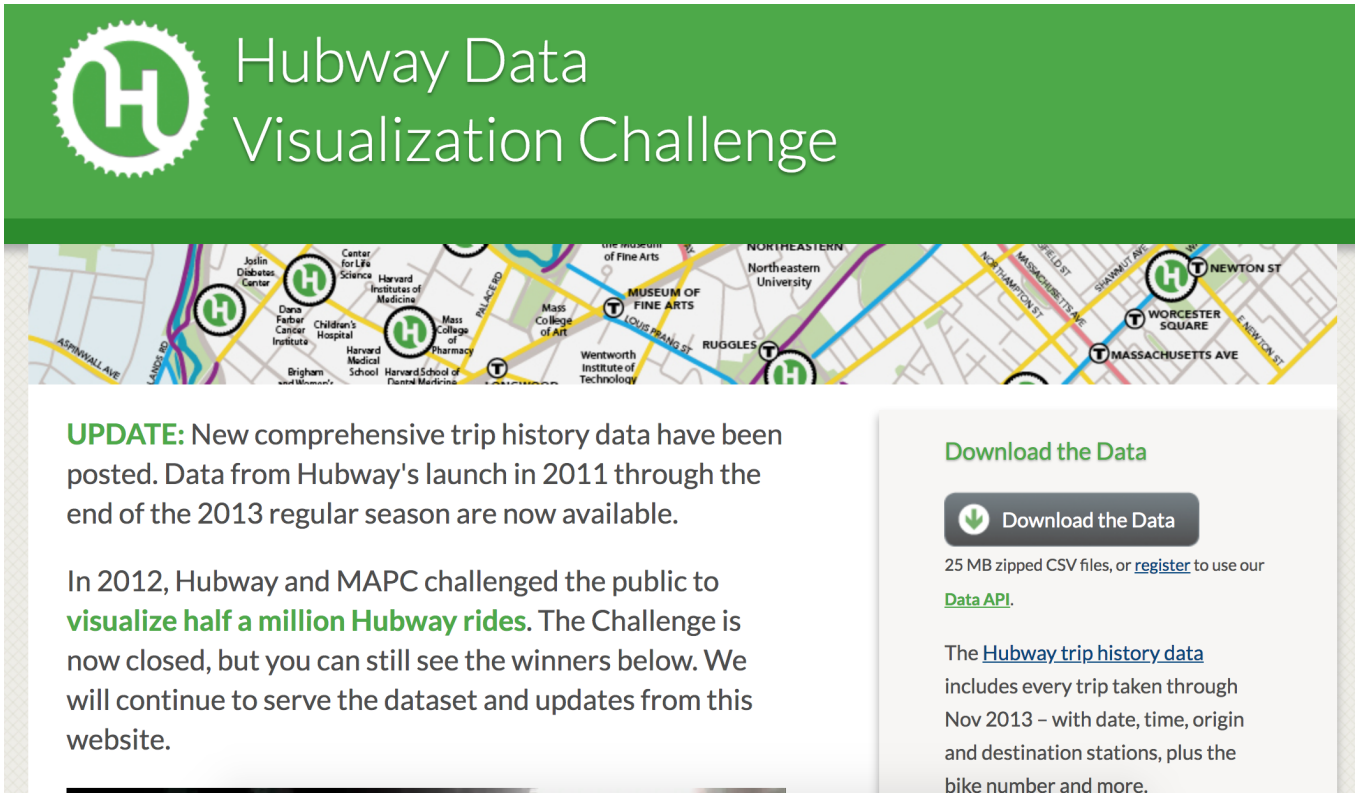
From 2013 to 2017, each year they put their data available for *Habway data challenge.*

Hubway Data

On March 7, 2018, Hubway announced a six-year partnership with Blue Cross Blue Shield of Massachusetts, which included a system-wide re-branding as Blue Bikes, along with expanding the system to 3,000 bikes and adding over 100 new stations by the end of 2019.
The name change took effect on May 9, 2018, with the release of new and re-branded bicycles.

Download the data

`http://hubwaydatachallenge.org/`



directory named: hubway-2011-07-through-2013-11

Hubway Data

What is the Hubway Data?

1. Hubway station
   Containing information of stations (location [lon,lat], station ID, station name, station address, station zipcode)

2. Hubway trips
   Each record is for one trip: trip duration, start station, end station, start data, end data, bike number, member type (member/casual) , gender (if available), etc

Hubway Data

Some important libraries you need for data analysis in python:

```
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
import matplotlib as mpl
import datetime as dt

%matplotlib inline
```

Hubway Data

Some important libraries you need for data analysis in python:

```
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
import matplotlib as mpl
import datetime as dt

%matplotlib inline
```

and read the csv file:

```
s1=pd.read_csv('Data/hubway_2011_07_through_2013_11/hubway_trips.csv')
```

Data Analysis

General information about your dataset:

```
s1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1579025 entries, 0 to 1579024
Data columns (total 13 columns):
seq_id         1579025 non-null int64
hubway_id      1579025 non-null int64
status         1579025 non-null object
duration       1579025 non-null int64
start_date     1579025 non-null object
strt_statn     1579011 non-null float64
end_date       1579025 non-null object
end_statn      1578980 non-null float64
bike_nr        1578559 non-null object
subsc_type     1579025 non-null object
zip_code       1106259 non-null object
birth_date     350644 non-null float64
gender         1106414 non-null object
dtypes: float64(3), int64(3), object(7)
memory usage: 156.6+ MB
```

Data Analysis

How many trips per day?

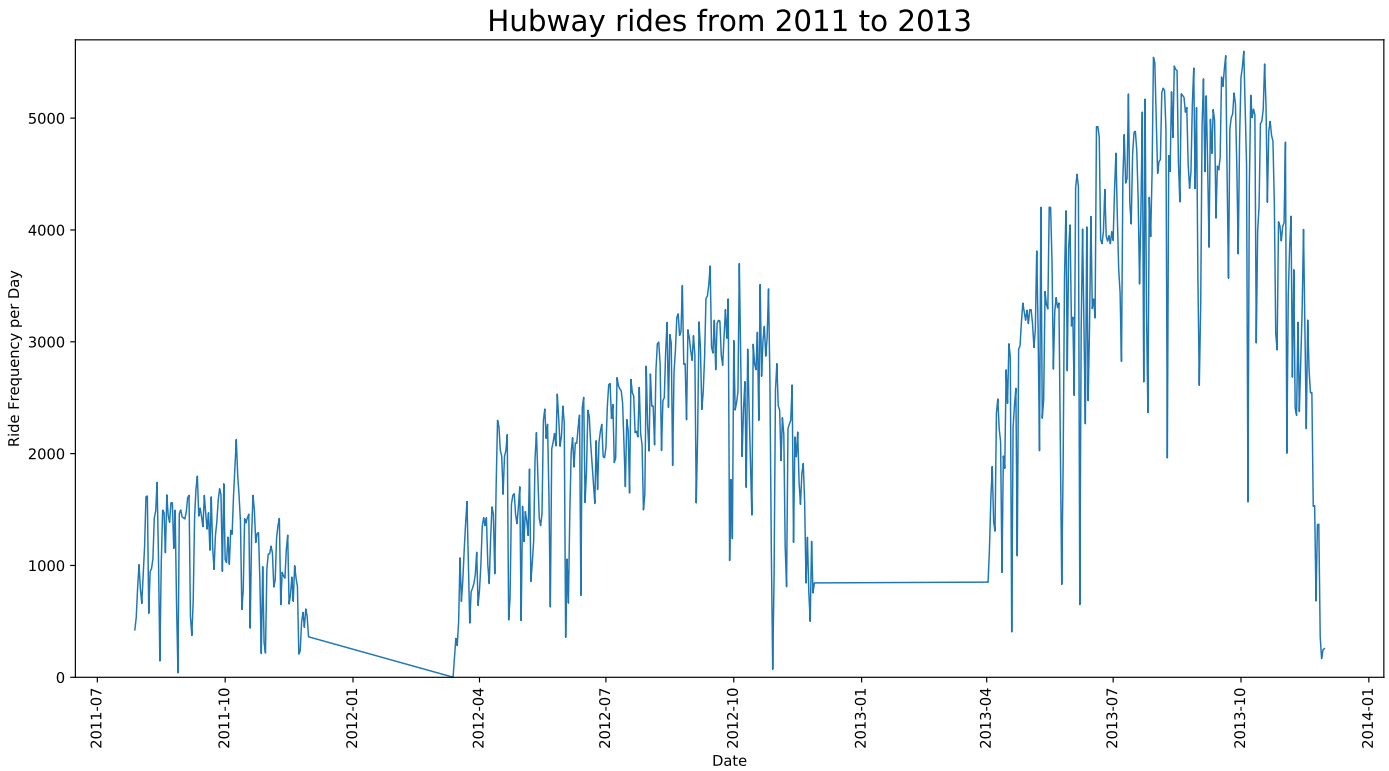
```
s1['s_date'] = pd.to_datetime(s1['start_date'])

s1['SDate'] = s1['s_date'].dt.date
s1['STime'] = s1['s_date'].dt.time
```

```
b=s1.groupby(s1['SDate']).count()
```

How to plot ?

```
plt.ylim(0,5700)
plt.xticks(rotation='vertical')
plt.plot(b['SDate'].index,b['SDate'],linewidth=1)
plt.show()
```

Data Analysis



Data Analysis

How the rides are separated between genders?

```
s2=s1.loc[s1['gender'] == 'Male']
s3=s1.loc[s1['gender'] == 'Female']
s4=s1.loc[s1['subsc_type'] == 'Casual']

print('male',len(s2)/len(s1))
print('female',len(s3)/len(s1))
print('casual',len(s4)/len(s1))

b2=s2.groupby(s2['SDate']).count()
b3=s3.groupby(s3['SDate']).count()
b4=s4.groupby(s4['SDate']).count()
```

```
male 0.5286262092113805
female 0.17206820664650654
casual 0.299305584142113
```

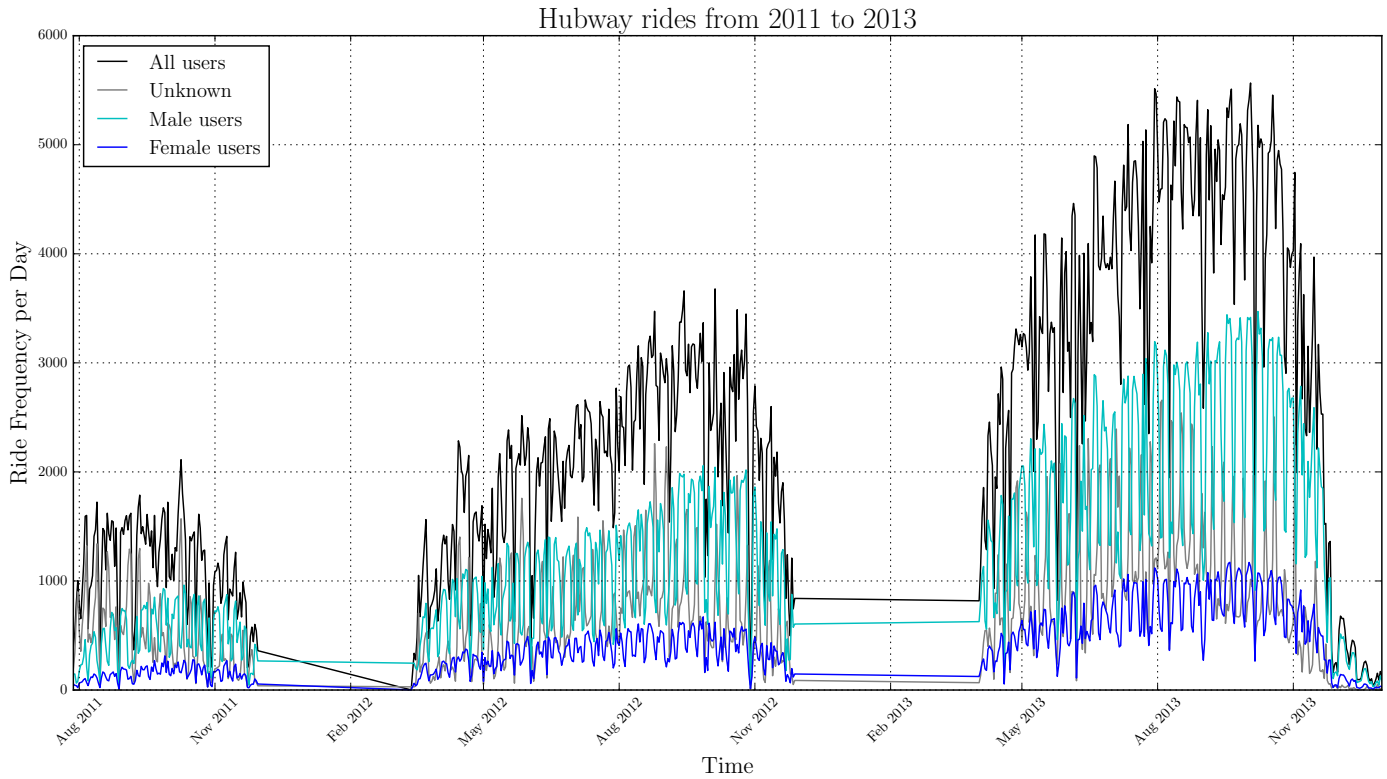
How users are growing from 2011 to 2013?

Data Analysis

```
plt.rc('text', usetex=True)
plt.rc('font', family='serif')

plt.figure(figsize=(16,8))

plt.plot(b.index,b['SDate'],color='k',linewidth=1, label='All users')
plt.plot(b.index,b3['status'], color='b',linewidth=1, label='Female users')

plt.plot(b4.index,b4['status'], color='y',linewidth=1 ,label='Unknown')
plt.plot(b2.index,b2['duration'], color='c',linewidth=1 ,label='Male users')

plt.ylim(0,6000)
plt.grid(True)
plt.ylabel('Ride Frequency per Day',fontsize=16)
plt.xlabel('Time',fontsize=16)
plt.xticks(rotation='45')
plt.title('Hubway rides from 2011 to 2013',fontsize=20)
plt.legend()
plt.grid(which = 'minor')
plt.legend(loc=2, fontsize=14)
plt.savefig('summary.pdf', format='pdf')
plt.show()
```

Data Analysis

Result:

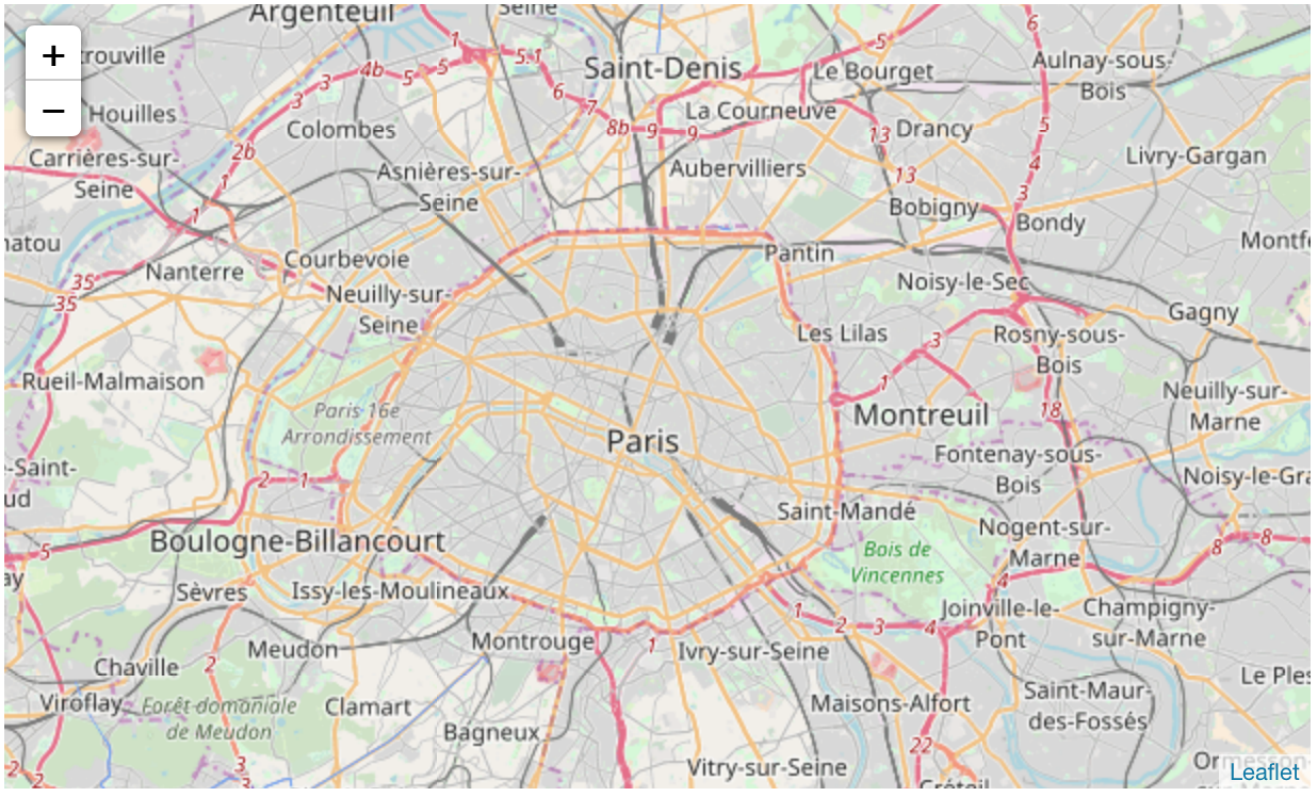


GeoSpatial Data Visualization

- Folium
- geopandas
- ipyleaflet
- osmnx

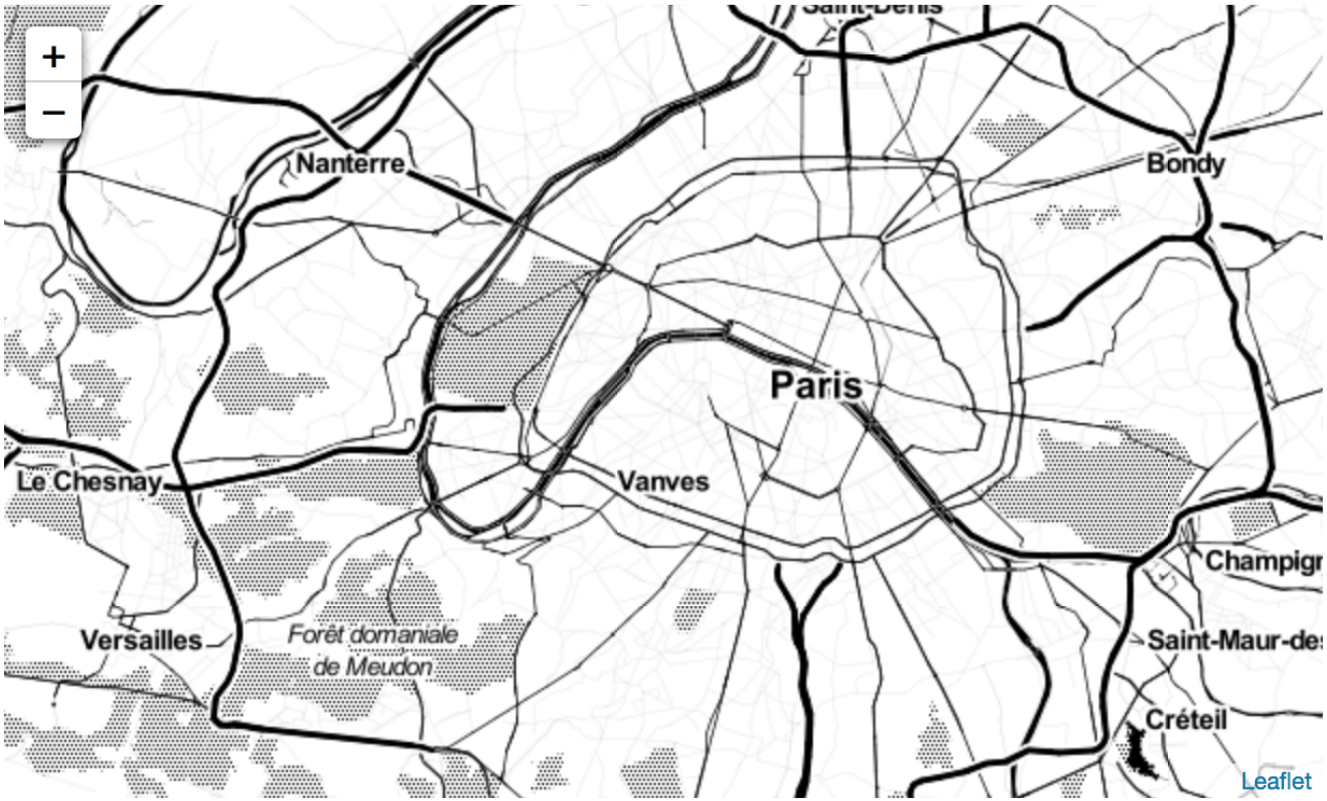
Interactive maps with folium

```
import folium
my_map = folium.Map(location=[48.8517,2.36],
                    zoom_start=11)

my_map
```



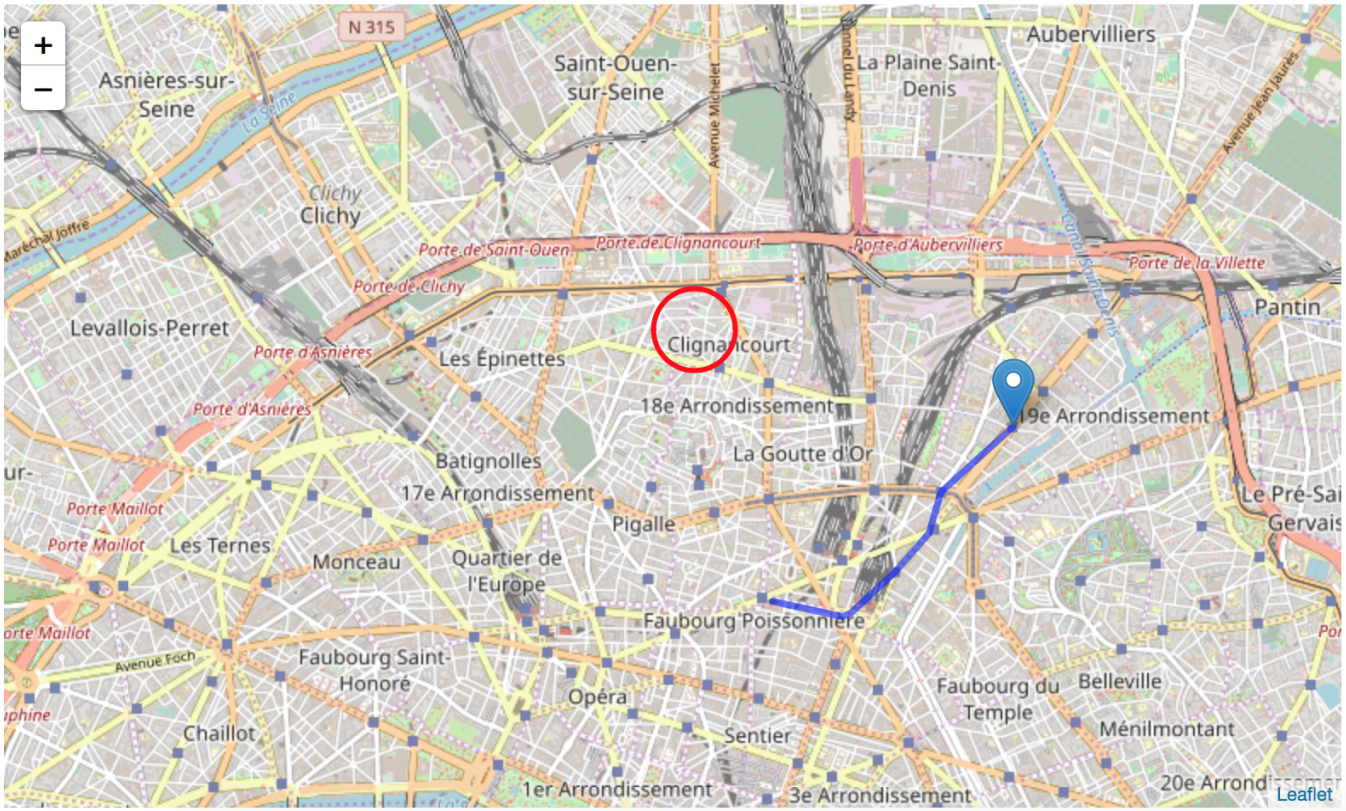
Folium, different tiles

```
import folium
my_map = folium.Map(location=[48.85,2.3] , tiles='Stamen Toner', zoom_start=11)

my_map
```



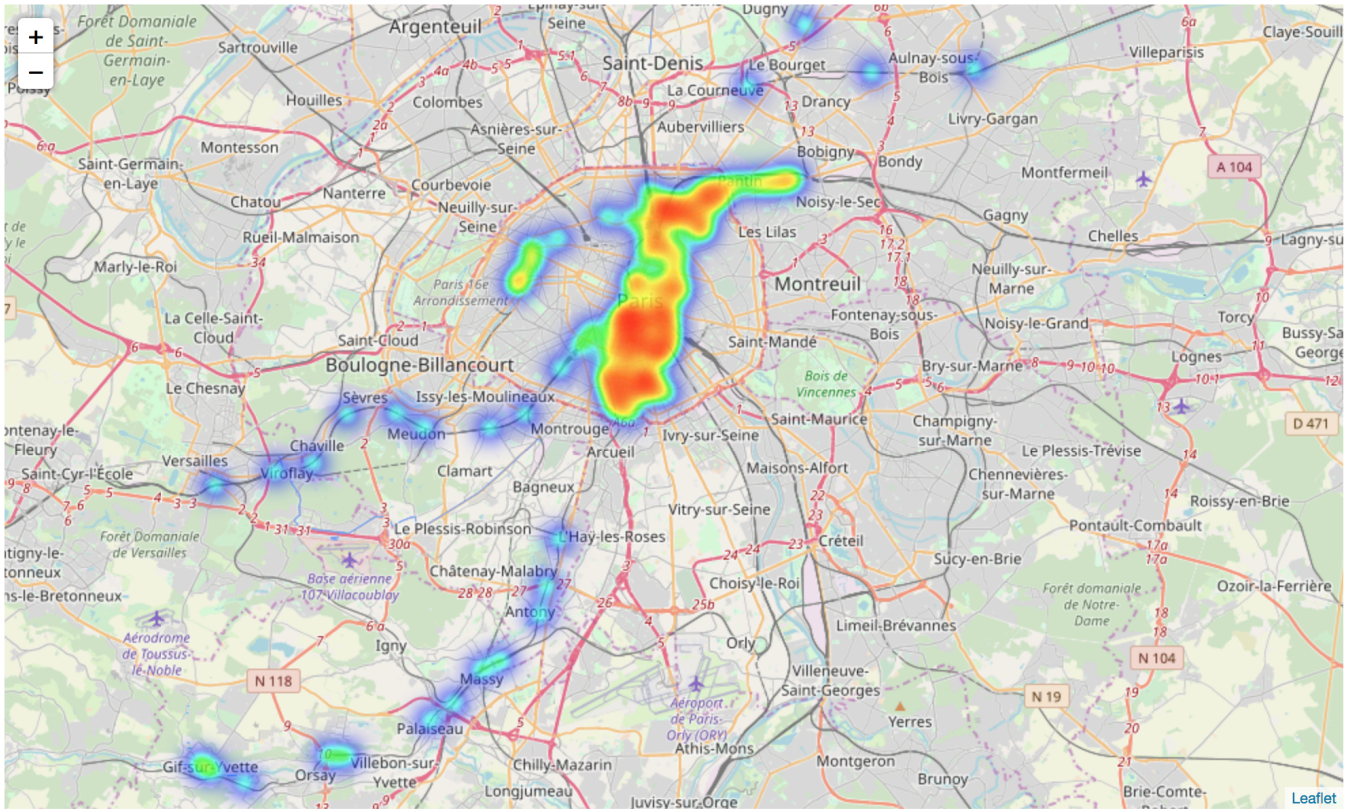
lines, markers, circles

```
m = folium.Map([48.89, 2.34], zoom_start=13)

line=[[48.877, 2.35],[48.876, 2.357],[48.8762027, 2.3579127],[48.8788951, 2.3624612],
      [48.8815852, 2.3659215],[48.8842416, 2.3670741],[48.8884812, 2.3742226]]

folium.Circle(location=[48.895, 2.3422] , color='red',radius=300).add_to(m)
folium.Marker(location=[48.8884812, 2.3742226], popup='Marker').add_to(m)
folium.PolyLine(locations=line,color= 'blue', weight=4,opacity=0.6).add_to(m)

m
```



Heatmap with folium

```
from folium.plugins import HeatMap

m = folium.Map([48.89, 2.34], zoom_start=11)
HeatMap(L , min_opacity=0.2,radius=10, blur=15,
                    max_zoom=1, ).add_to(m)
m
```



Continue with Hubway Dataset

Load the hubway-station file:

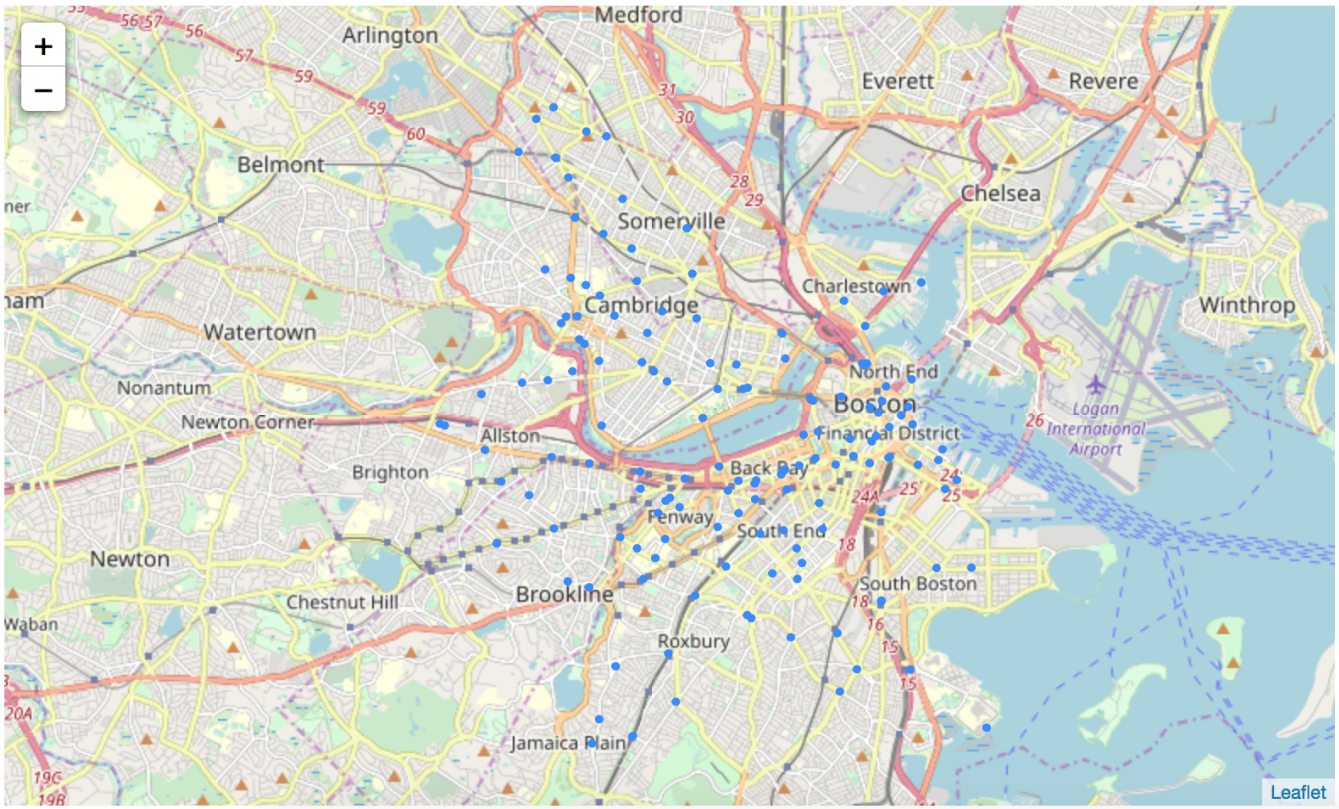
```
s_data=pd.read_csv('Data/hubway_2011_07_through_2013_11/hubway_stations.csv')
```

```
s_data.head()
```

	id	terminal	station	municipal	lat	lng	status
0	3	B32006	Colleges of the Fenway	Boston	42.340021	-71.100812	Existing
1	4	C32000	Tremont St. at Berkeley St.	Boston	42.345392	-71.069616	Existing
2	5	B32012	Northeastern U / North Parking Lot	Boston	42.341814	-71.090179	Existing
3	6	D32000	Cambridge St. at Joy St.	Boston	42.361285	-71.065140	Existing
4	7	A32000	Fan Pier	Boston	42.353412	-71.044624	Existing

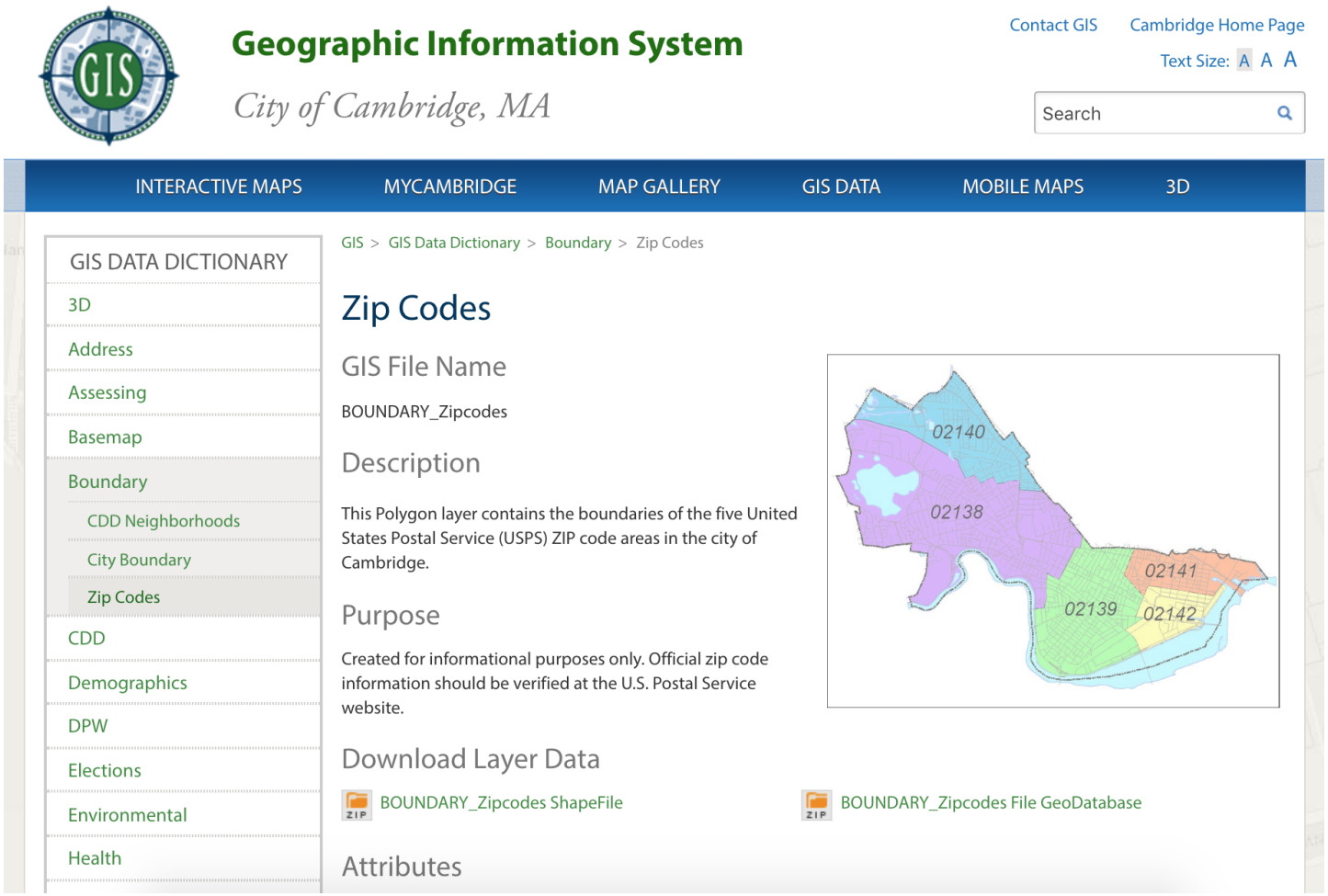
Hubway stations visualization

```
my_m = folium.Map(location=[42.36,-71.1] , zoom_start=12)

for i in s_data.index:
    lon=s_data.iloc[i]['lng']
    lat=s_data.iloc[i]['lat']
    folium.Circle(location=[lat,lon] ).add_to(my_m)

my_m
```



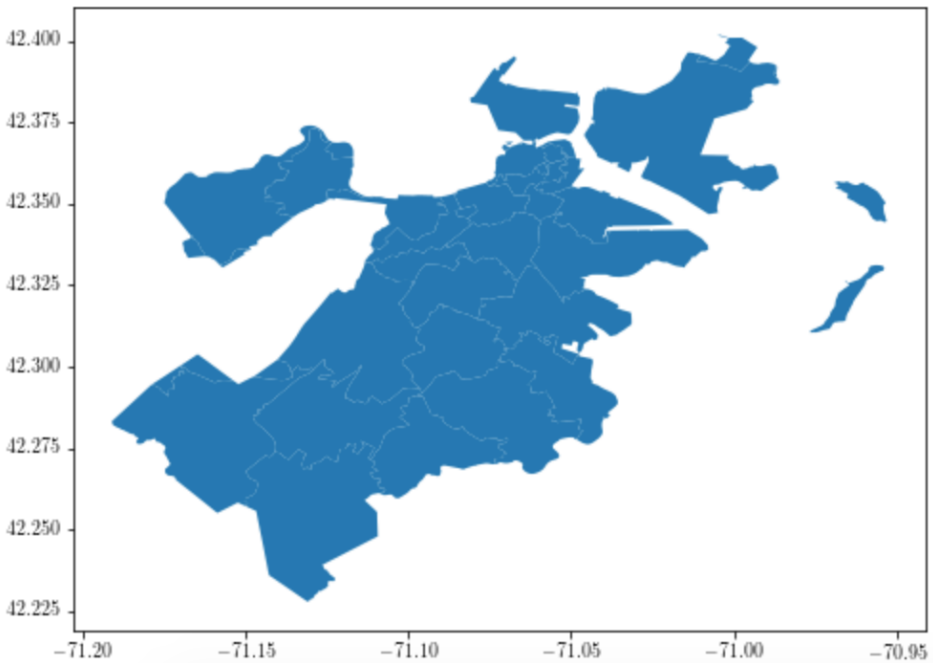
GeoSpatial data with GeoPandas

https://www.cambridgema.gov/GIS/gisdatadictionary/Boundary/BOUNDARY_Zipcodes



Geopandas

```
import geopandas

boston = geopandas.read_file('ZIP_codes.geojson')

boston.plot(figsize=(18,6))
plt.show()
```



Choropleth maps with geopandas

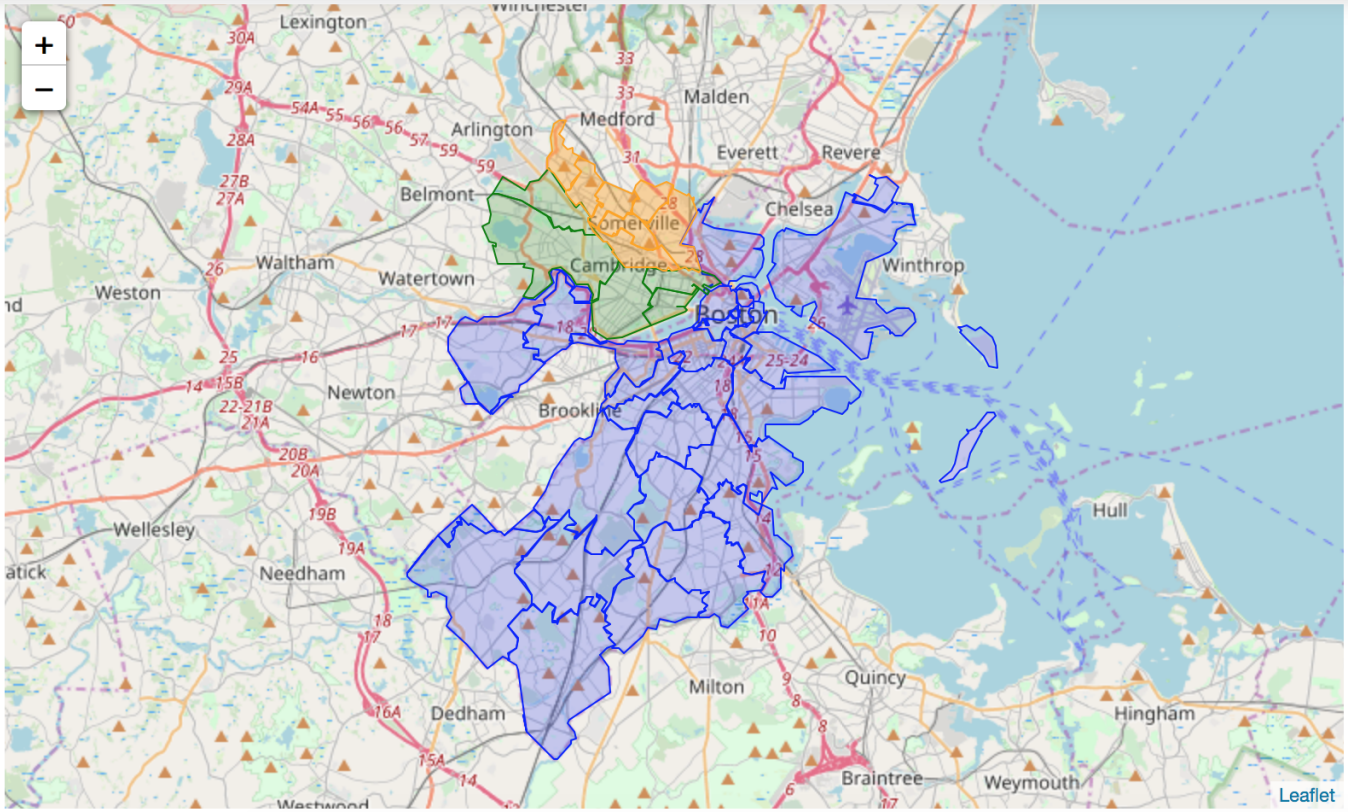
```
import geopandas

cambridge=geopandas.read_file('BOUNDARY_Zipcodes.shp/BOUNDARY_Zipcodes.shp')
sumervil=geopandas.read_file('Wards/Wards.shp')
boston = geopandas.read_file('ZIP_codes.geojson')
```

```
boston.head()
```

	OBJECTID	ZIP5	ShapeSTArea	ShapeSTLength	geometry
0	1	02134	3.721936e+07	40794.182396	POLYGON ((-71.12340461235522 42.36420867214283...
1	2	02125	6.476052e+07	62224.521440	POLYGON ((-71.04541458491363 42.32380666715233...
2	3	02110	6.637284e+06	18358.213496	POLYGON ((-71.05109058896998 42.36418367507441...
3	4	02118	3.116158e+07	32353.407618	POLYGON ((-71.06315159137533 42.34688867055895...
4	5	02126	6.078585e+07	45488.394711	POLYGON ((-71.09669659978795 42.29095065982932...

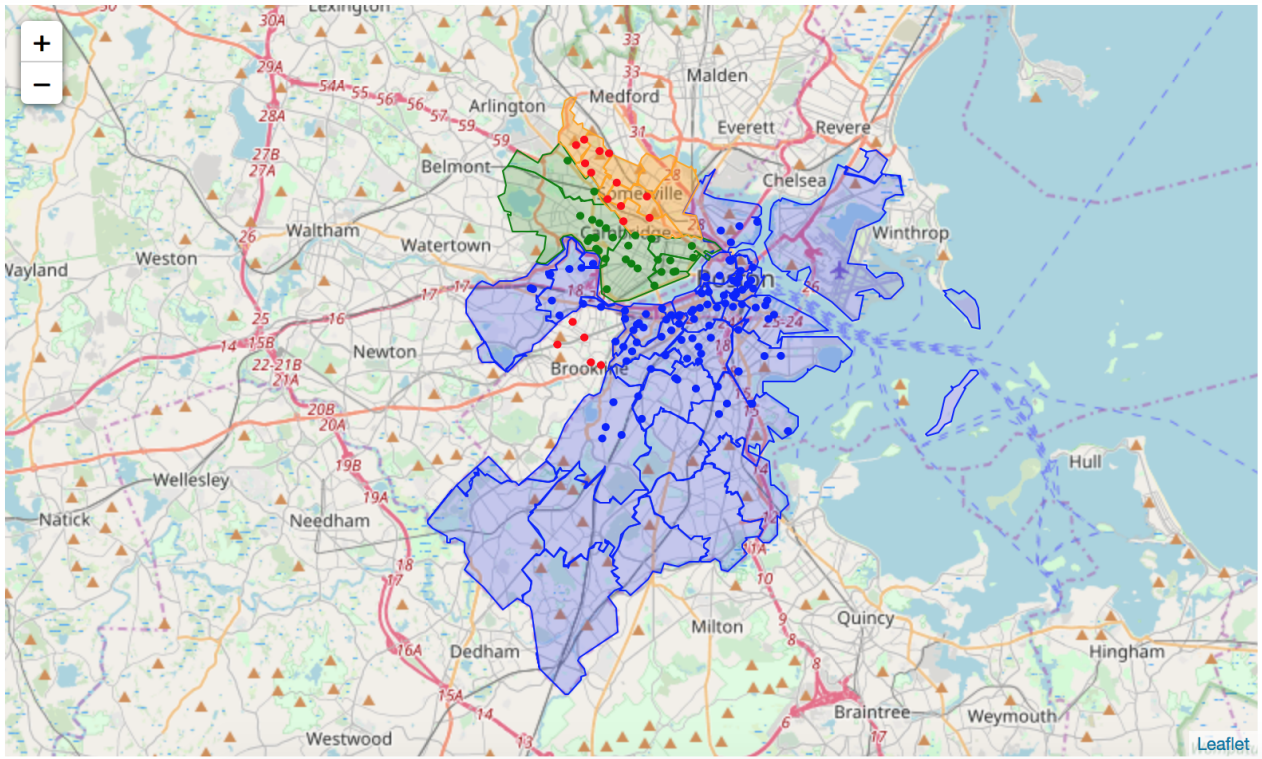
Choropleth maps with geopandas

```
my_m = folium.Map(location=[42.36,-71.1] , zoom_start=12)


my_m.choropleth(geo_data=boston, line_color='blue',fill_color='blue',fill_opacity=0.2, line_weight=1)
my_m.choropleth(geo_data=cambridge, line_color='green',fill_color='green',fill_opacity=0.2,line_weight=1)
my_m.choropleth(geo_data=sumervil, line_color='orange',fill_color='orange',fill_opacity=0.4,line_weight=1)

my_m
```



Geopandas

```
my_m = folium.Map(location=[42.36,-71.1] , zoom_start=12)


my_m.choropleth(geo_data=boston, line_color='blue',fill_color='blue',fill_opacity=0.2, line_weight=1)
my_m.choropleth(geo_data=cambridge, line_color='green',fill_color='green',fill_opacity=0.2,line_weight=1)
my_m.choropleth(geo_data=sumervil, line_color='orange',fill_color='orange',fill_opacity=0.4,line_weight=1)

for i in s_data.index:
    lon=s_data.iloc[i]['lng']
    lat=s_data.iloc[i]['lat']
    if s_data.iloc[i]['municipal']=='Boston' :
        folium.Circle(location=[lat,lon] , color='blue').add_to(my_m)
    elif s_data.iloc[i]['municipal']=='Cambridge':
        folium.Circle(location=[lat,lon] , color='green').add_to(my_m)
    else:
        folium.Circle(location=[lat,lon] , color='red').add_to(my_m)

my_m
```



Conclusion

What we reviewed in this session?

- What is GeoSpatial Data and its formats
- How to import & read & do basic analysis with geospatial data
- How to create interactive maps with folium library
- How to use geospatial data to improve visualization results (ipython leaflet, geopandas, osmnx)

Q & A

Thank You

fereshteh.asgari@gmail.com