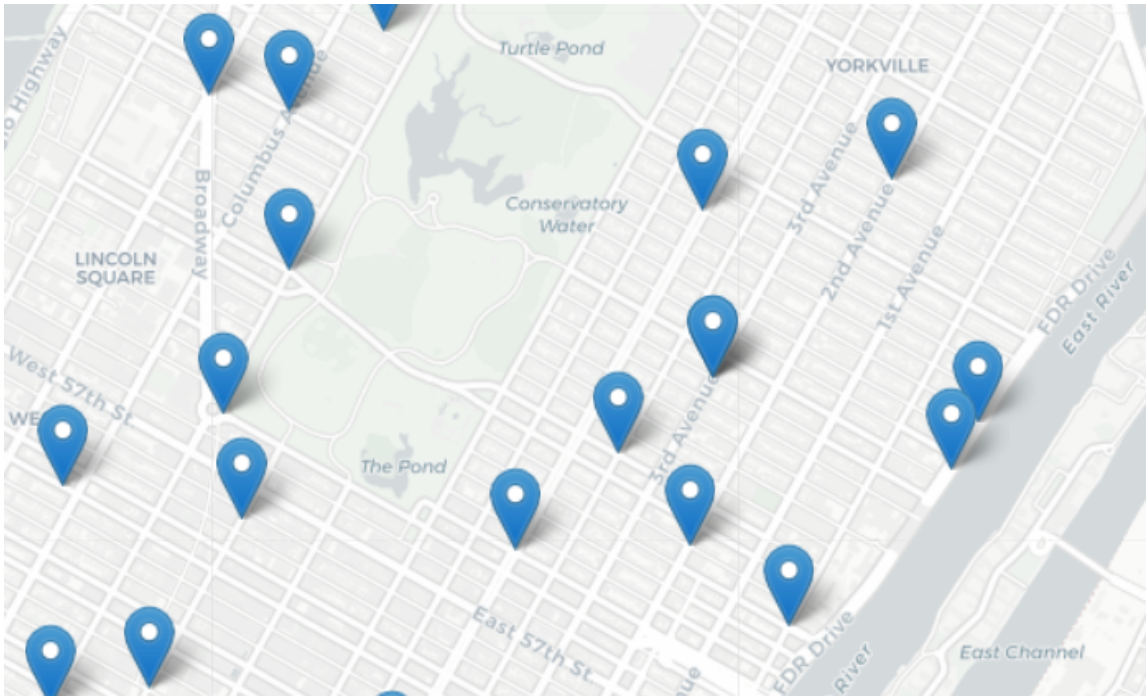
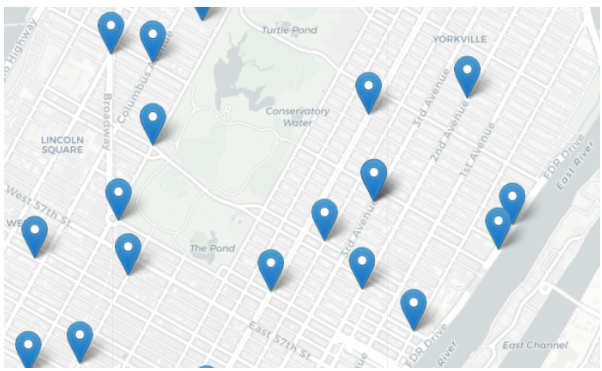
MfA: Python in the City



Katherine St. John
City University of New York
American Museum of Natural History

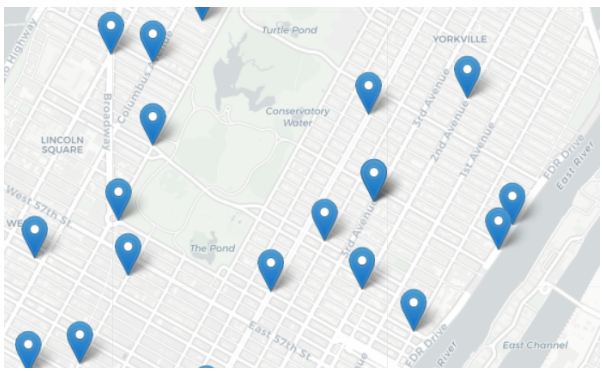
Goal: Sit at a table with someone who you did not sit with the first two sessions.

Outline



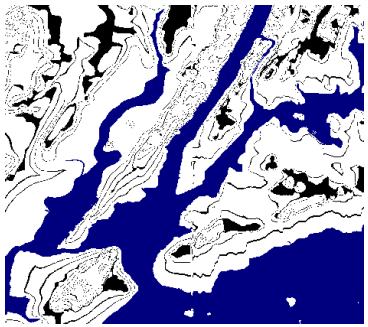
- Recap
- HTML-Scalable Maps: Folium
- Extracting Data
- geoJSON Format & Choropleth Maps
- Break
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- Wrap Up

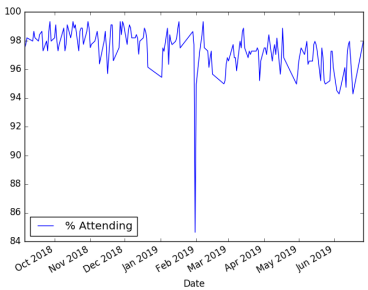
Outline

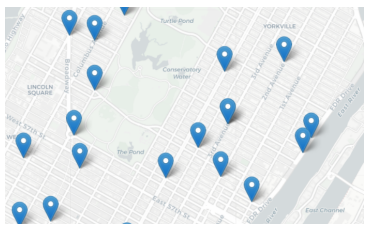


- **Recap**
- HTML-Scalable Maps: Folium
- Extracting Data
- geoJSON Format & Choropleth Maps
- Break
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- Wrap Up

Recap: Workshop Overview



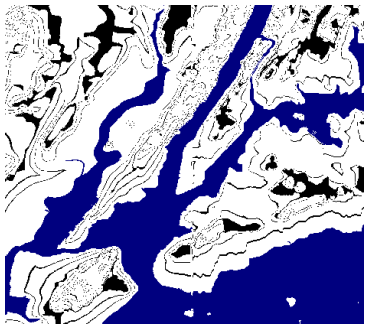


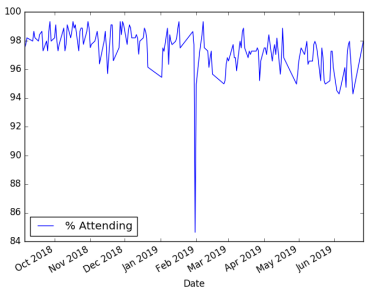


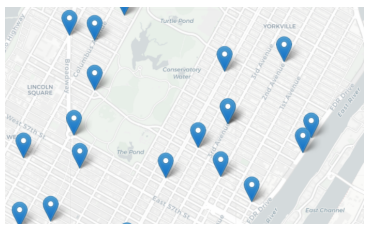
Three sessions:

1. Flood Maps (arrays & images)
2. School Attendence (structured data, file I/O)
3. Mapping Collisions (using objects, mapping coordinates)

Recap: Workshop Overview







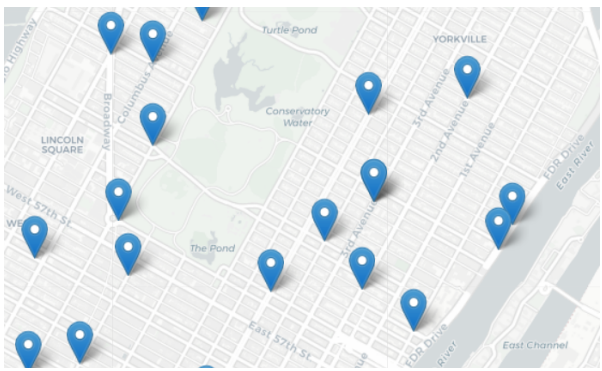
Three sessions:

1. Flood Maps (arrays & images)
2. School Attendence (structured data, file I/O)
3. Mapping Collisions (using objects, mapping coordinates)

Each session:

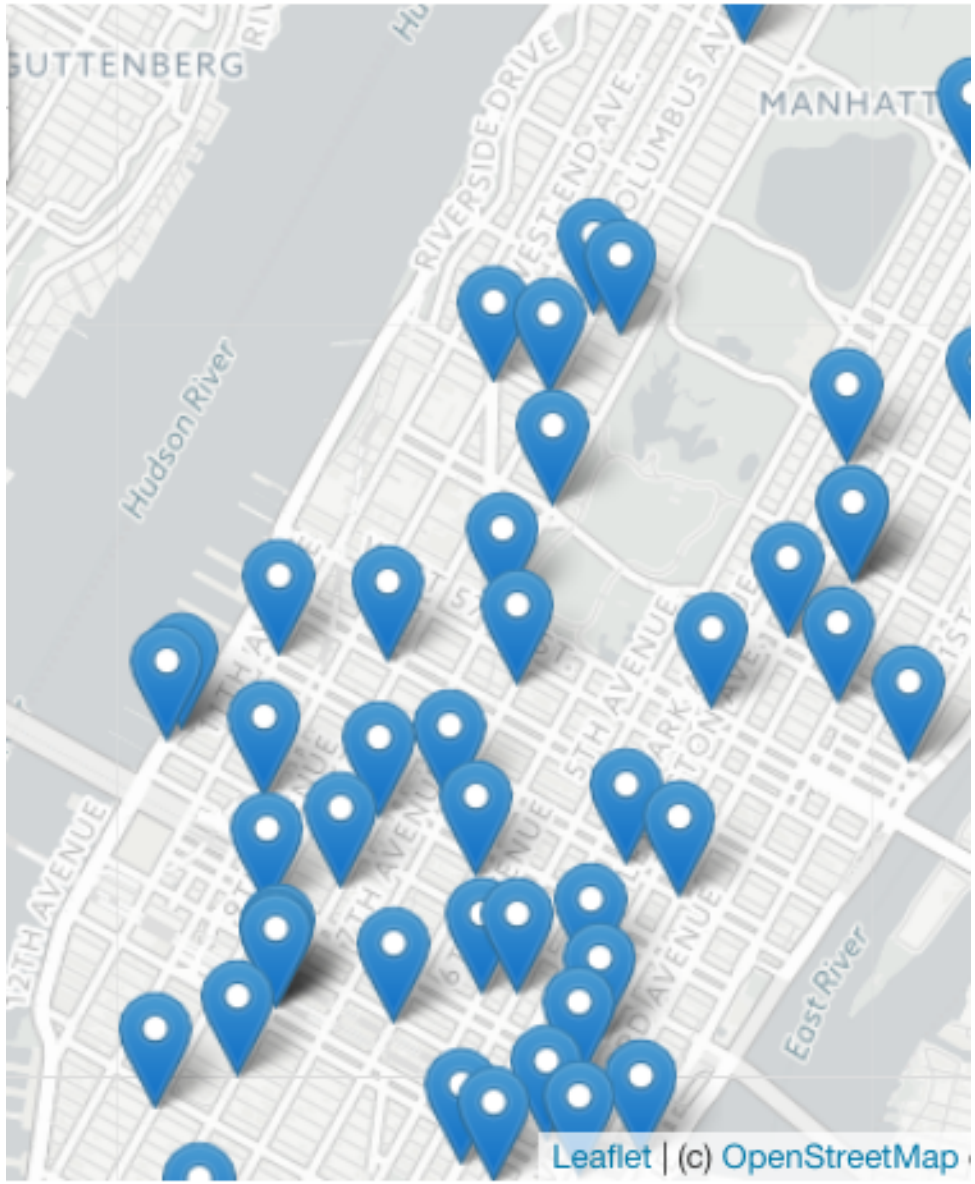
- Design Challenge
  - Analyze a publicly available dataset
  - Introduce computing concepts & packages
  - Write a program to solve the problem
- Variations on the theme
- Design a Challenge

Outline



- Recap
- **HTML-Scalable Maps: Folium**
- Extracting Data
- geoJSON Format & Choropleth Maps
- Break
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- Wrap Up

HTML-Scalable Maps: Folium



Folium

- A module for making HTML maps.

Folium

- A module for making HTML maps.
- It's a Python interface to the popular `leaflet.js`.

Folium

- A module for making HTML maps.
- It's a Python interface to the popular `leaflet.js`.
- Outputs `.html` files which you can open in a browser.

Folium

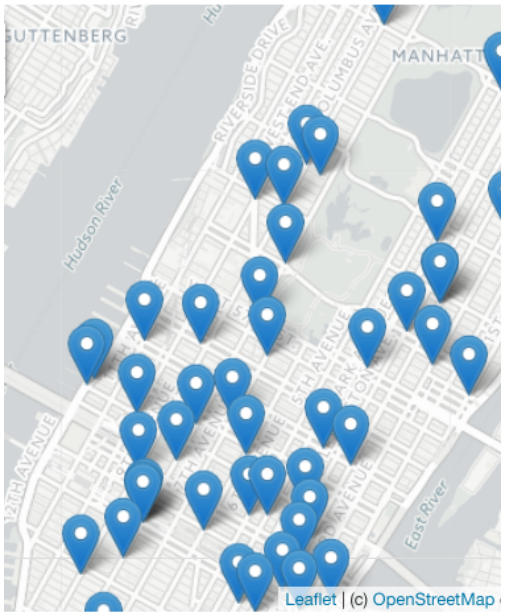
- A module for making HTML maps.
- It's a Python interface to the popular `leaflet.js`.
- Outputs `.html` files which you can open in a browser.
- An extra step:

Folium

- A module for making HTML maps.
- It's a Python interface to the popular `leaflet.js`.
- Outputs `.html` files which you can open in a browser.
- An extra step:

Write code. → *Run program.* → *Open .html in browser.*

Demo



(Map created by Folium.)

Your turn: Make a map using folium

- To use:

```
import folium
```

Your turn: Make a map using folium

- To use:

  ```
  import folium
  ```

- Create a map:

  ```
  myMap = folium.Map()
  ```

Your turn: Make a map using folium

- To use:

  ```
  import folium
  ```

- Create a map:

  ```
  myMap = folium.Map()
  ```

- Make markers:

  ```
  newMark = folium.Marker([lat,lon],popup=name)
  ```

Your turn: Make a map using folium

- To use:

  ```
  import folium
  ```

- Create a map:

  ```
  myMap = folium.Map()
  ```

- Make markers:

  ```
  newMark = folium.Marker([lat,lon],popup=name)
  ```

- Add to the map:

  ```
  newMark.add_to(myMap)
  ```

Your turn: Make a map using folium

- To use:

  `import folium`
- Create a map:

  `myMap = folium.Map()`
- Make markers:

  `newMark = folium.Marker([lat,lon],popup=name)`
- Add to the map:

  `newMark.add_to(myMap)`
- Can customize map with starting location, zoom level and background map ("tiles"):

  `myMap = folium.Map(location=[40.75, -74.125], zoom_start=10, tiles='Stamen Watercolor')`

  Many options to customize background map ("tiles"):

Your turn: Make a map using folium

- To use:

  `import folium`
- Create a map:

  `myMap = folium.Map()`
- Make markers:

  `newMark = folium.Marker([lat,lon],popup=name)`
- Add to the map:

  `newMark.add_to(myMap)`
- Can customize map with starting location, zoom level and background map ("tiles"):

  `myMap = folium.Map(location=[40.75, -74.125], zoom_start=10, tiles='Stamen Watercolor')`

  Many options to customize background map ("tiles"):

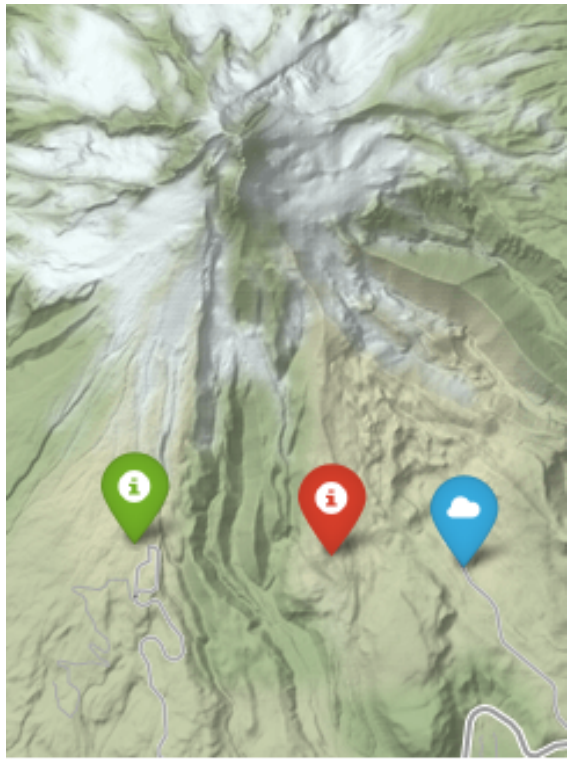
  (Some background map options: `Stamen Terrain'`, `'Stamen Watercolor'`, `'Mapbox Bright'`, `'Stamen Toner'`, `'Cartodb Positron'`)
- Save to a file:

  `myMap.save(outfile="myMap.html")`

In Pairs of Triples

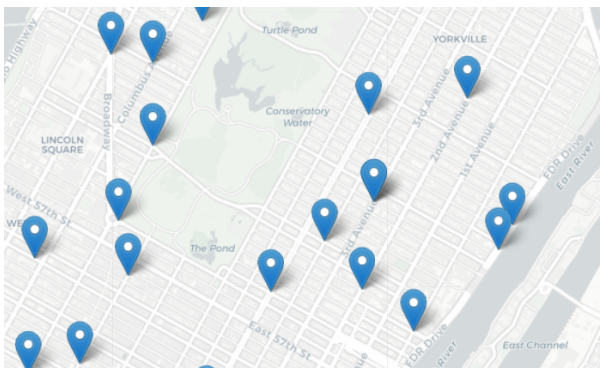
- Predict which each line of code does:

```
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)
```



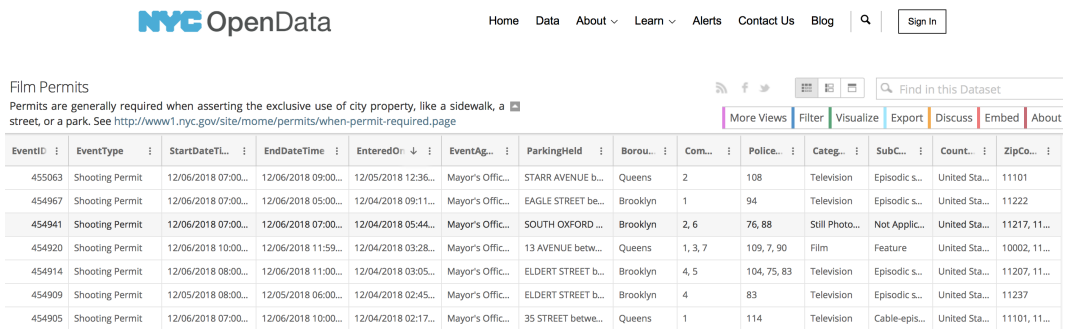
(example from Folium documentation)

Outline



- Recap
- HTML-Scalable Maps: Folium
- **Extracting Data**
- geoJSON Format & Choropleth Maps
- Break
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- Wrap Up

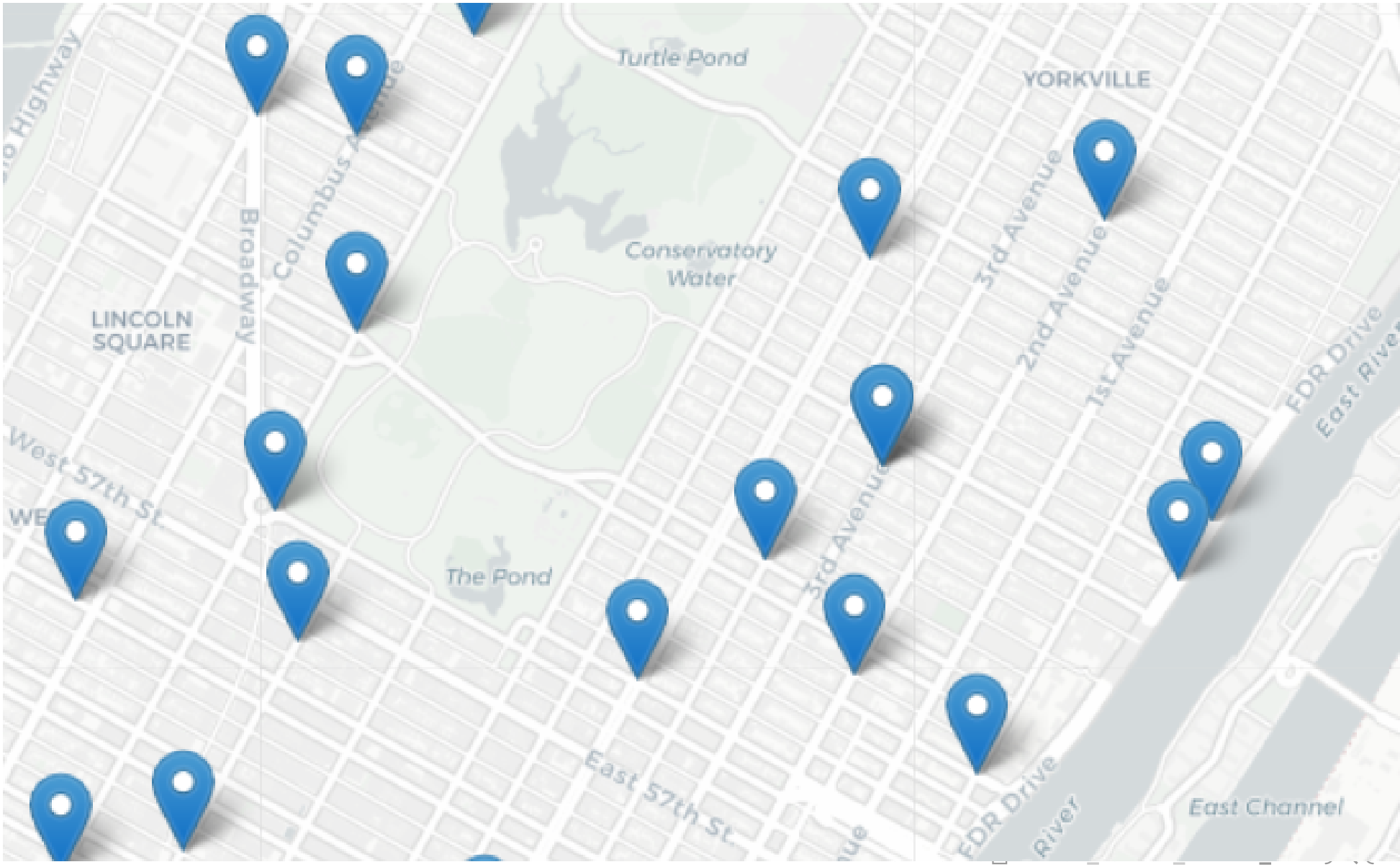
Recall: Film Permits Example



- Download the data as a CSV file and store on your computer.
- Python program:

```
#CSci 127 Teaching Staff
#March 2019
#OpenData Film Permits

#Import pandas for reading and analyzing CSV data:
import pandas as pd
csvFile = "filmPermits.csv"   #Name of the CSV file
tickets = pd.read_csv(csvFile)#Read in the file to a dataframe
print(tickets)                #Print out the dataframe
print(tickets["ParkingHeld"]) #Print out streets (multiple times)
print(tickets["ParkingHeld"].value_counts()) #Print out streets & number of times used
print(tickets["ParkingHeld"].value_counts()[:10])  #Print 10 most popular
```

Extracting Data for a Map

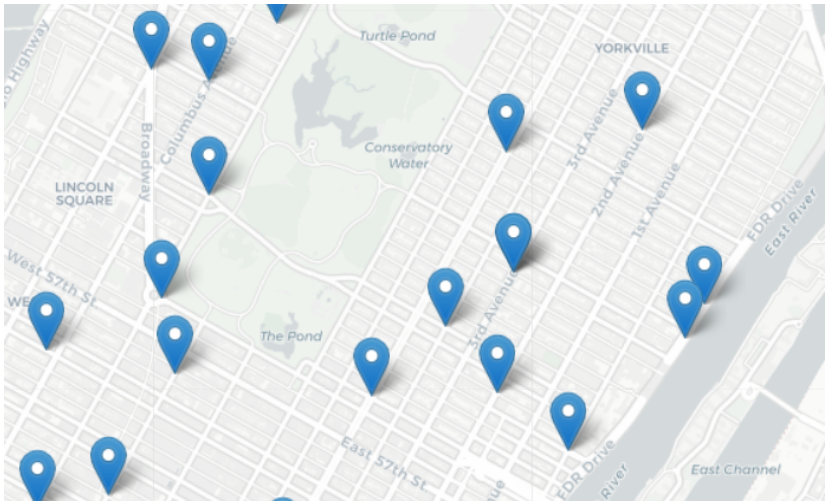


Extracting Data for a Map

A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 ST	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STR	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0

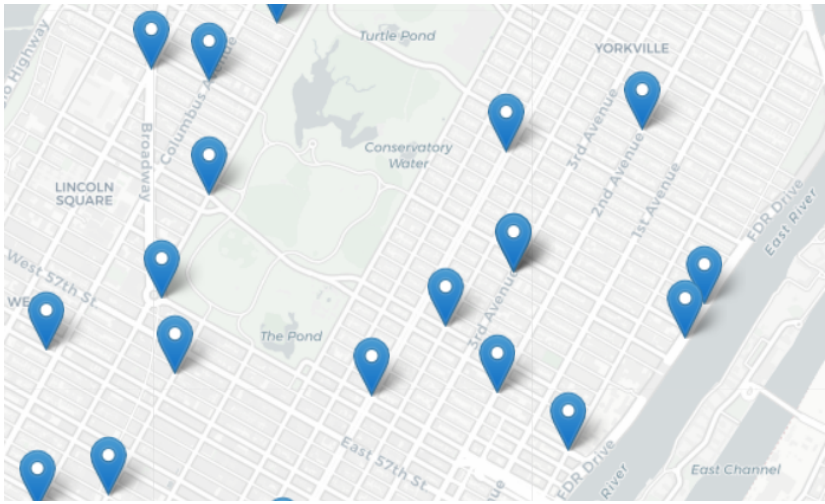
Extracting Data for a Map

A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



Extracting Data for a Map

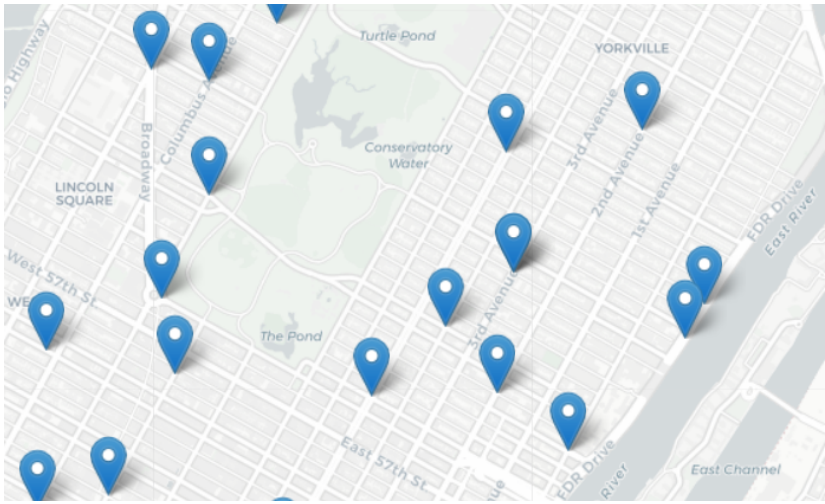
A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.

Extracting Data for a Map

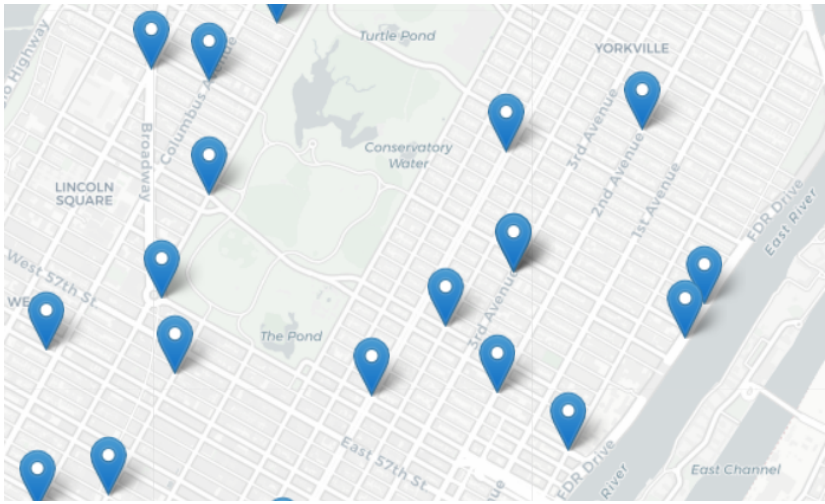
A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.
  - Filter for latitude and longitude (not blank).

Extracting Data for a Map

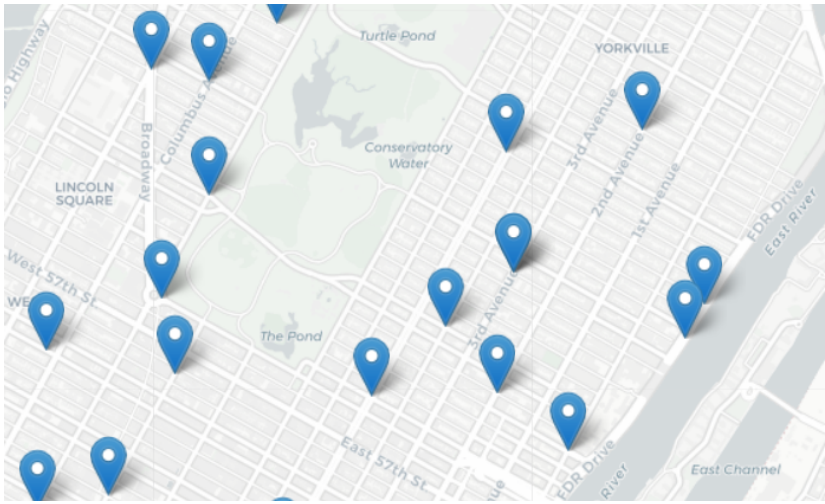
A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.
  - Filter for latitude and longitude (not blank).
  - Use pandas to read in data.

Extracting Data for a Map

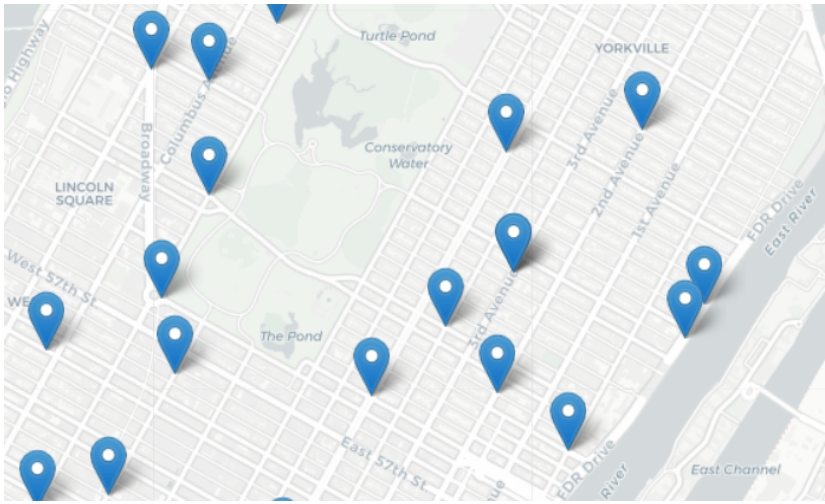
A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.
  - Filter for latitude and longitude (not blank).
  - Use pandas to read in data.
- Many ways to access data from pandas:

Extracting Data for a Map

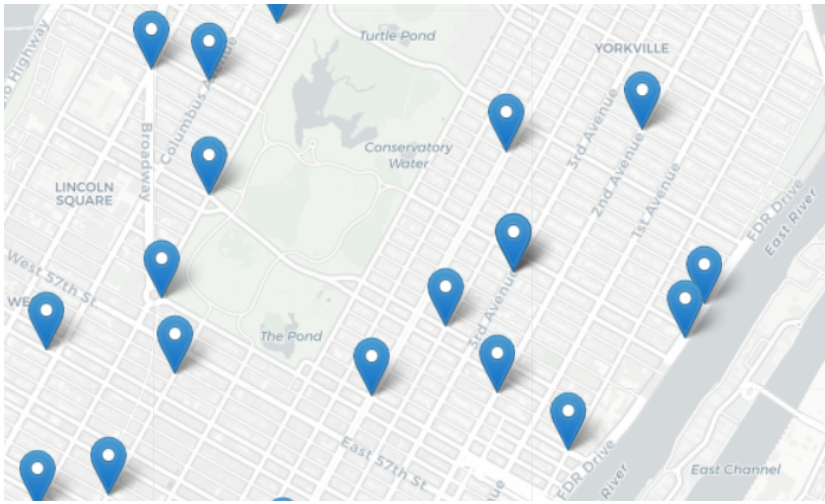
A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.
  - Filter for latitude and longitude (not blank).
  - Use pandas to read in data.
- Many ways to access data from pandas:
  - Since we are extracting row-by-row to create objects, will iterate through the dataFrame (`df.iterrows()`).

Extracting Data for a Map

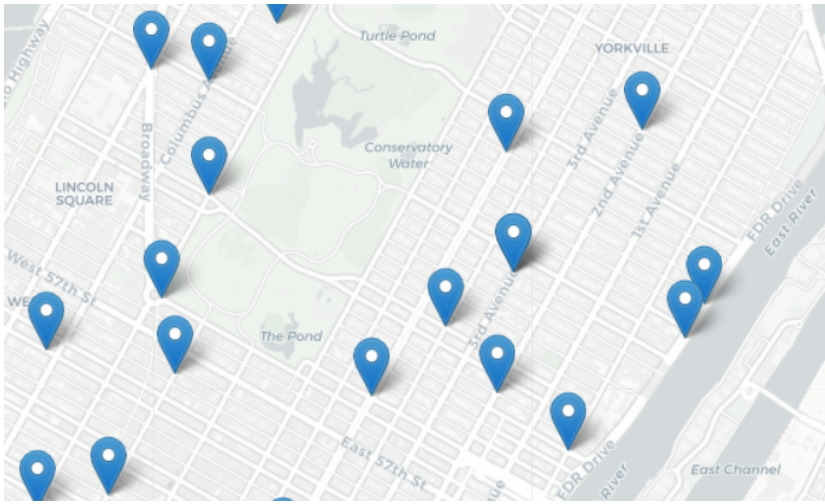
A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.
  - Filter for latitude and longitude (not blank).
  - Use pandas to read in data.
- Many ways to access data from pandas:
  - Since we are extracting row-by-row to create objects, will iterate through the dataFrame (`df.iterrows()`).
  - Can also, pull comments & iterate over zip:
    `for (lat,lon) in zip(df["LATITUTE"],df["LONGITUDE"])`

Extracting Data for a Map

A	B	C	D	E	F	G	H	I	J	K
DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
10/18/16	8:10	STATEN ISLA	10312	40.5405508	-74.193197	(40.5405508	RATHBUN AV	NIPPON AVENUE		1
10/18/16	8:10	BROOKLYN	11238	40.6864547	-73.968107	(40.6864547, -73.9681074)			107 GREE	1
10/18/16	8:10	BROOKLYN	11234	40.6261739	-73.922234	(40.6261739, -73.9222336)			5515 AVEI	1
10/18/16	8:10	BROOKLYN	11218	40.6312756	-73.976049	(40.6312756	18 AVENUE	EAST 2 STREET		1
10/18/16	8:10	BROOKLYN	11217	40.6827366	-73.972234	(40.6827366, -73.9722339)			718 ATLA	1
10/18/16	8:10			40.7741706	-73.82933	(40.7741706, -73.8293296)				0
10/18/16	8:10			40.7713758	-73.877131	(40.7713758, -73.8771311)				0
10/18/16	8:10			40.7415268	-73.78455	(40.7415268, -73.7845496)				0
10/18/16	8:10			40.7365373	-73.856099	(40.7365373, -73.8560987)				0
10/18/16	8:10			40.7361484	-73.926504	(40.7361484, -73.9265043)				0
10/18/16	8:10	QUEENS	11426	40.7351801	-73.718346	(40.7351801	HILLSIDE AVE	249 STREET		0
10/18/16	8:10	QUEENS	11417	40.6719314	-73.851635	(40.6719314	NORTH CONI	84 STREET		0
10/18/16	8:10	QUEENS	11355	40.7606728	-73.821007	(40.7606728, -73.8210074)			144-20 41 A	0
10/18/16	8:10	QUEENS	11103	40.7623893	-73.911797	(40.7623893	44 STREET	30 AVENUE		0
10/18/16	8:10	QUEENS	11101	40.7462958	-73.930409	(40.7462958	34 STREET	43 AVENUE		0
10/18/16	8:10	MANHATTAN	10035	40.8050573	-73.939034	(40.8050573	EAST 125 STI	PARK AVENUE		0
10/18/16	8:10	MANHATTAN	10018	40.7556432	-73.990962	(40.7556432, -73.9909619)			607 8 AVI	0
10/18/16	8:10	MANHATTAN	10016	40.7437832	-73.973508	(40.7437832	EAST 34 STRI	1 AVENUE		0
10/18/16	8:10	MANHATTAN	10011	40.7458255	-74.001812	(40.7458255	WEST 22 STR	9 AVENUE		0
10/18/16	8:10	MANHATTAN	10003	40.7317985	-73.982114	(40.7317985, -73.9821143)			251 1 AVI	0
10/18/16	8:10	BROOKLYN	11234	40.6166112	-73.926628	(40.6166112	UTICA AVENI	AVENUE O		0
10/18/16	8:10	BROOKLYN	11222	40.7317513	-73.945513	(40.7317513	RUSSELL STR	GREENPOINT AVENUE		0
10/18/16	8:10	BROOKLYN	11206	40.707655	-73.939838	(40.707655,	BUSHWICK A	MONTROSE AVENUE		0
10/18/16	8:10	BRONX	10468	40.8716697	-73.897797	(40.8716697	WEST 197 ST	RESERVOIR AVENUE		0



- Steps:
  - Download the data as a CSV file and store on your computer.
  - Filter for latitude and longitude (not blank).
  - Use pandas to read in data.
- Many ways to access data from pandas:
  - Since we are extracting row-by-row to create objects, will iterate through the dataFrame (`df.iterrows()`).
  - Can also, pull comments & iterate over zip:
    `for (lat,lon) in zip(df["LATITUTE"],df["LONGITUDE"])`
  - Using `apply()` is possible, but it returns a series or dataFrame, so, processing still required.

Extracting Data for a Map

- Python program:

Extracting Data for a Map

- Python program:

```
#Mapping Collisions

#Libraries
import folium
import pandas as pd
```

Extracting Data for a Map

- Python program:

```
#Mapping Collisions

#Libraries
import folium
import pandas as pd
```

```
#Getting file names:
inF = input('Enter CSV file name: ')
outF = input('Enter output file: ')
coll = pd.read_csv(inF)
```

Extracting Data for a Map

- Python program:

```
#Mapping Collisions

#Libraries
import folium
import pandas as pd
```

```
#Getting file names:
inF = input('Enter CSV file name: ')
outF = input('Enter output file: ')
coll = pd.read_csv(inF)
```

```
#Setting up the map:
mapCollisions = folium.Map(location=[40.768731, -73.964915],\
                           tiles="Cartodb Positron",zoom_start=11)
```

Extracting Data for a Map

- Python program:

```
#Mapping Collisions

#Libraries
import folium
import pandas as pd
```

```
#Getting file names:
inF = input('Enter CSV file name: ')
outF = input('Enter output file: ')
coll = pd.read_csv(inF)
```

```
#Setting up the map:
mapCollisions = folium.Map(location=[40.768731, -73.964915],\
                           tiles="Cartodb Positron",zoom_start=11)
```

```
#Looping through the file:
for index,row in coll.iterrows():
    lat = row["LATITUDE"]
    lon = row["LONGITUDE"]
    popname = row["CRASH TIME"]
    newMarker = folium.CircleMarker([lat, lon], popup=popname,\
                                    radius=5,color='blue')
    newMarker.add_to(mapCollisions)
```

Extracting Data for a Map

- Python program:

```
#Mapping Collisions

#Libraries
import folium
import pandas as pd
```

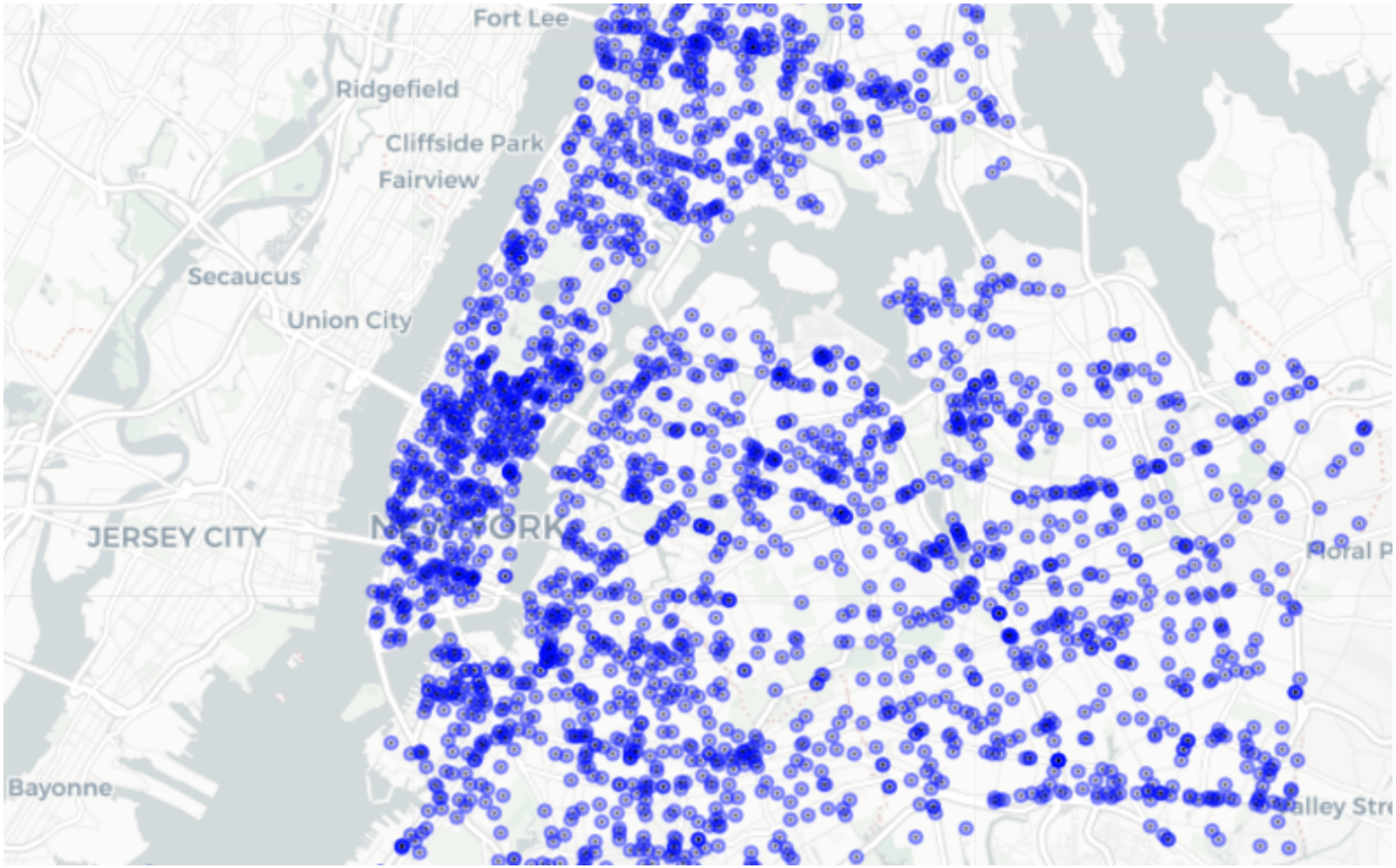
```
#Getting file names:
inF = input('Enter CSV file name: ')
outF = input('Enter output file: ')
coll = pd.read_csv(inF)
```

```
#Setting up the map:
mapCollisions = folium.Map(location=[40.768731, -73.964915],\
                           tiles="Cartodb Positron",zoom_start=11)
```

```
#Looping through the file:
for index,row in coll.iterrows():
    lat = row["LATITUDE"]
    lon = row["LONGITUDE"]
    popname = row["CRASH TIME"]
    newMarker = folium.CircleMarker([lat, lon], popup=popname,\
                                    radius=5,color='blue')
    newMarker.add_to(mapCollisions)
```

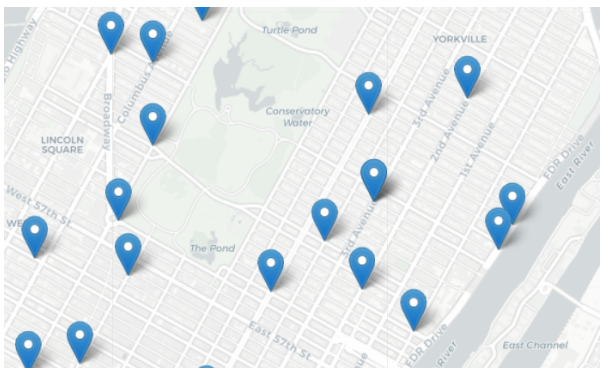
```
#Saving the HTML file:
mapCollisions.save(outfile=outF)
```

Extracting Data



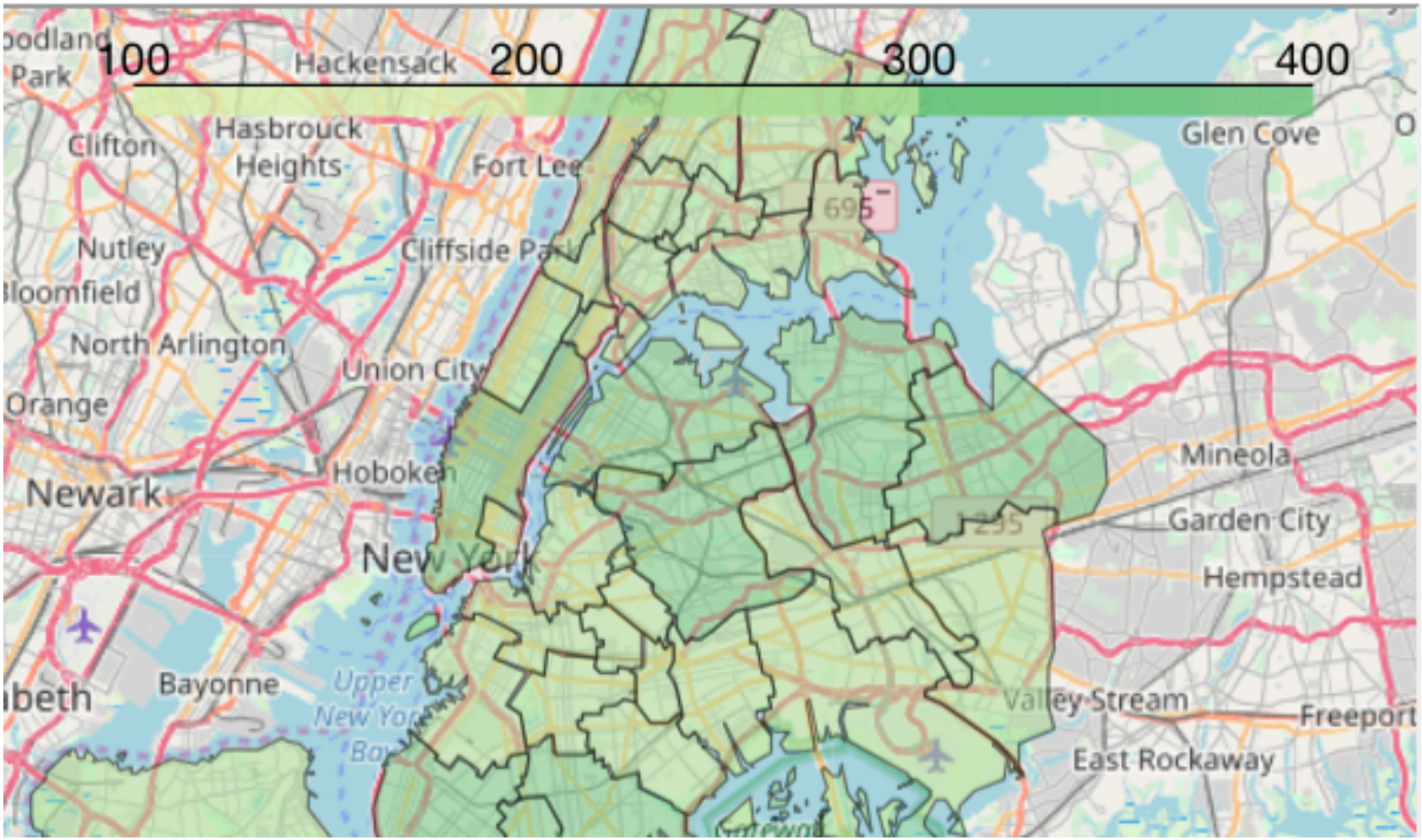
Make a map with only the accidents during evening rush hour.

Outline



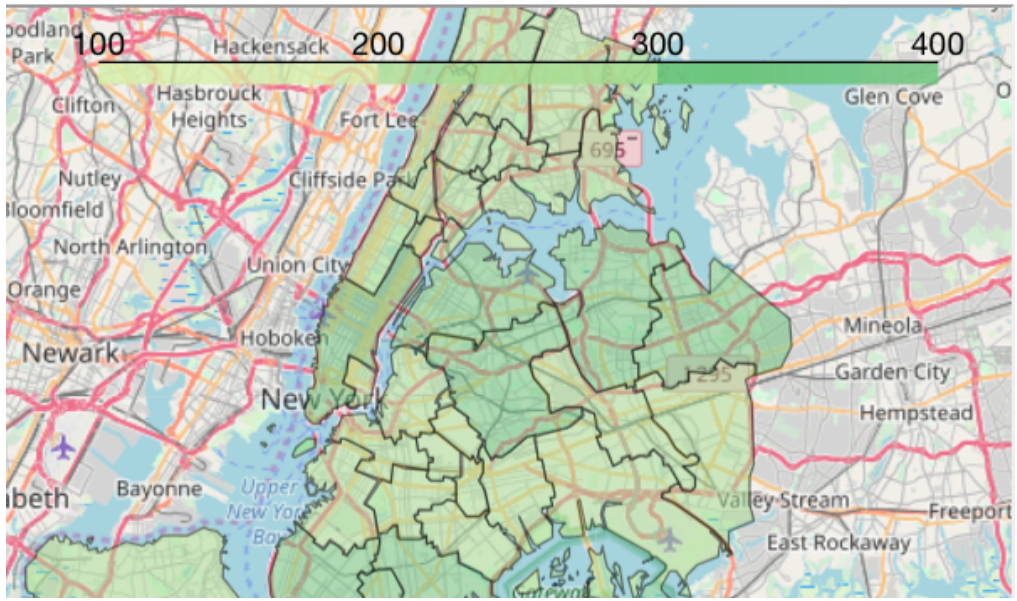
- Recap
- HTML-Scalable Maps: Folium
- Extracting Data
- **geoJSON Format & Choropleth Maps**
- Break
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- Wrap Up

geoJSON Format & Choropleth Maps



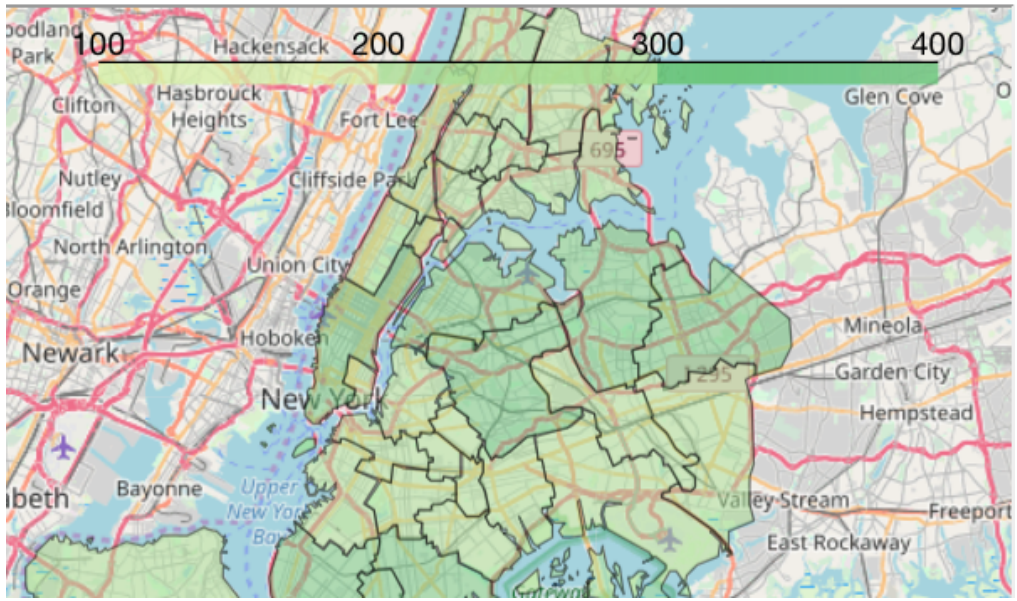
School districts shaded by math test scores.

geoJSON Format & Choropleth Maps



School districts shaded by math test scores.

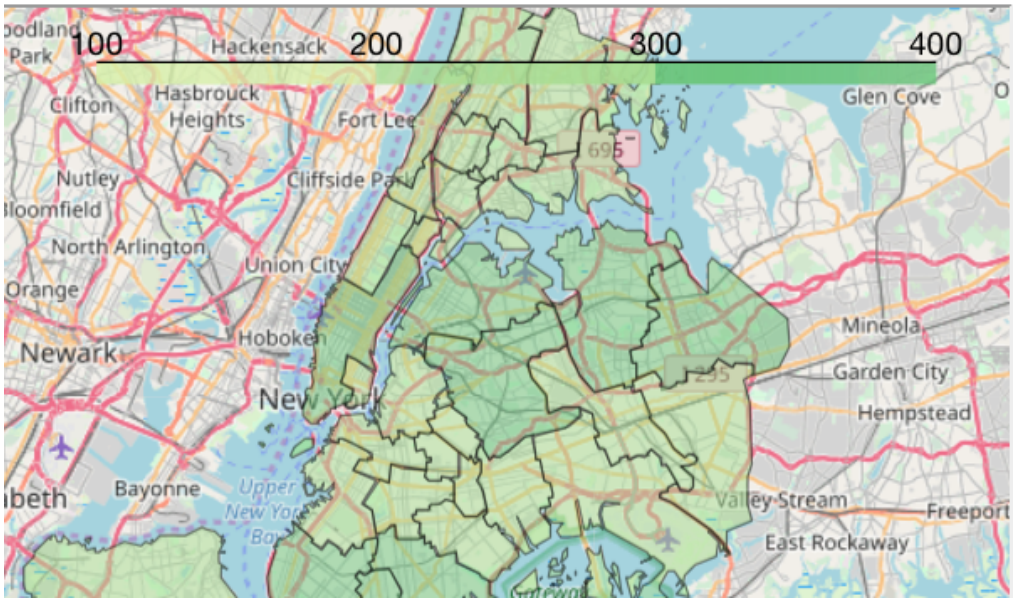
geoJSON Format & Choropleth Maps



School districts shaded by math test scores.

Two data files:

geoJSON Format & Choropleth Maps

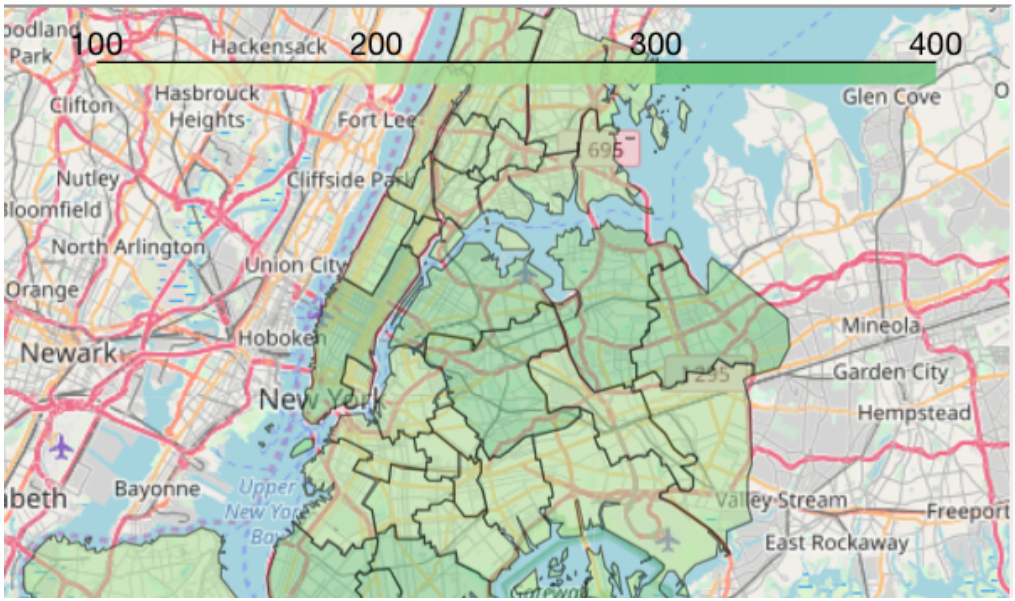


School districts shaded by math test scores.

Two data files:

- geoJSON file with polygonal regions (from OpenData NYC Planning)

geoJSON Format & Choropleth Maps

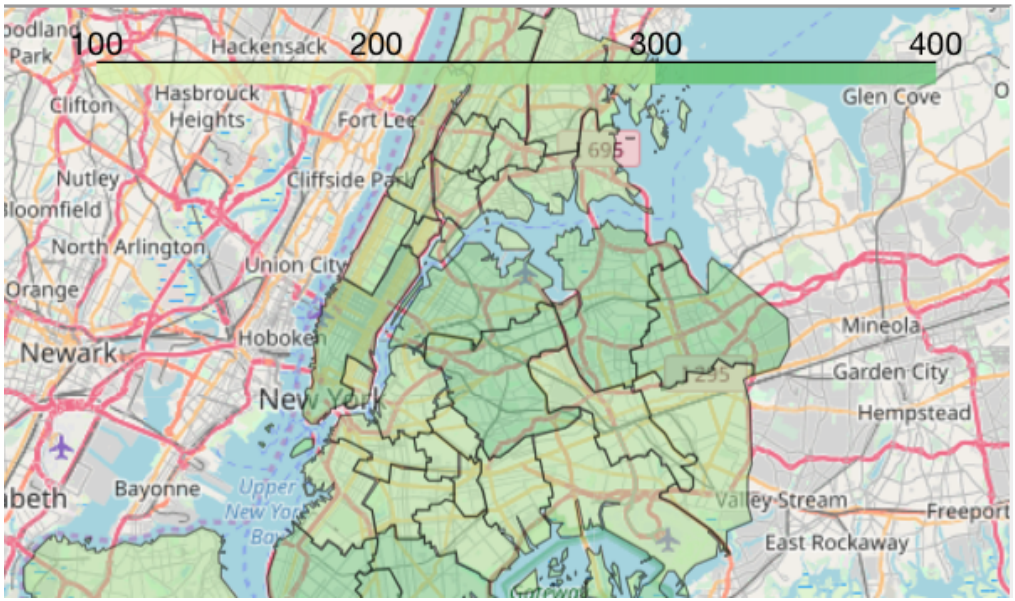


School districts shaded by math test scores.

Two data files:

- geoJSON file with polygonal regions (from OpenData NYC Planning)
- CSV file with test scores (NYC Department of Education)

geoJSON Format & Choropleth Maps



School districts shaded by math test scores.

Two data files:

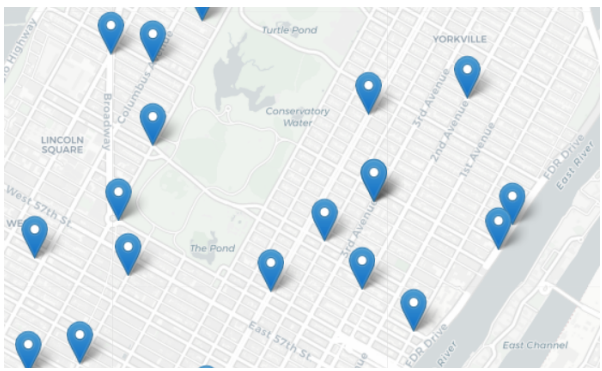
- geoJSON file with polygonal regions (from OpenData NYC Planning)
- CSV file with test scores (NYC Department of Education)

(Links on webpage.)

geoJSON Format & Choropleth Maps

```
#Import folium for maps and pandas for data wrangling
import folium
import pandas as pd

#Read in the test scores
fullData = pd.read_csv('math20132016.csv', skiprows = 6)
#Grab only 2016 data:
scores2016 = fullData[fullData.Year == 2016]
#Focus on 8th grade:
scores8th2016 = scores2016[fullData.Grade == "8"]
print(scores8th2016)

#Create a map:
schoolMap = folium.Map(location=[40.75, -74.125])

#Create a layer, shaded by test scores:
schoolMap.choropleth(geo_path="schoolDistricts.json",
                     fill_color='YlGn', fill_opacity=0.5, line_opacity=0.5,
                     threshold_scale = [100,200,300,400],
                     data = scores8th2016,
                     key_on='feature.properties.SchoolDist',
                     columns = ['district', 'Mean Scale Score']
                     )
#Output the map to an .html file:
schoolMap.save(outfile='testScores.html')
```

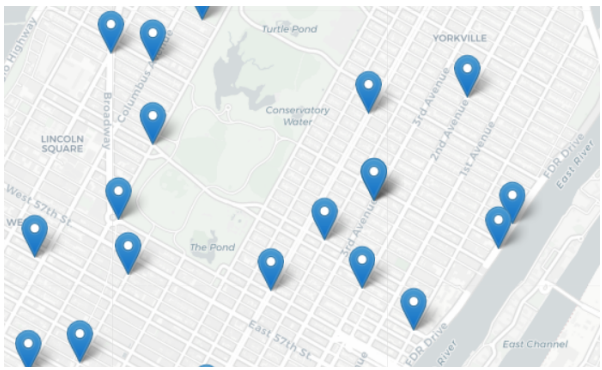
Outline



- Recap
- HTML-Scalable Maps: Folium
- Extracting Data
- geoJSON Format & Choropleth Maps
- **Break**
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- Wrap Up

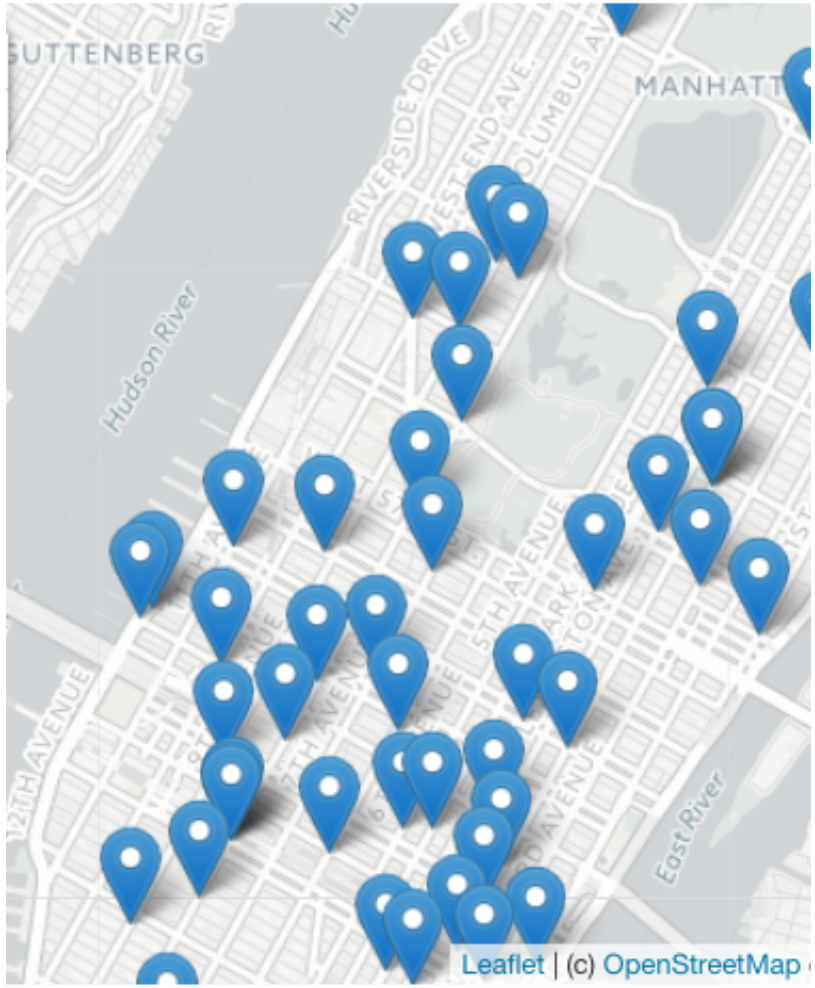
Break

Outline



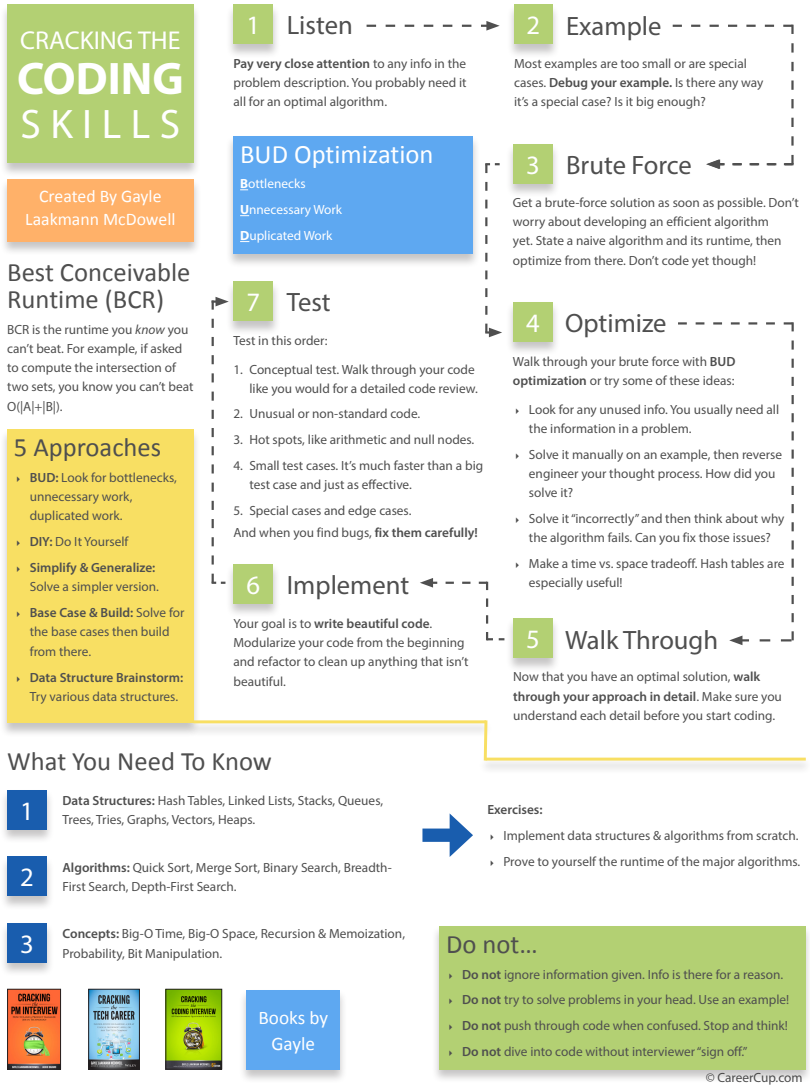
- Recap
- HTML-Scalable Maps: Folium
- Extracting Data
- geoJSON Format & Choropleth Maps
- Break
- **Design Challenge: Catchment Areas**
- Design Challenge: Clustering Data
- Wrap Up

Design Challenge: Catchment Areas



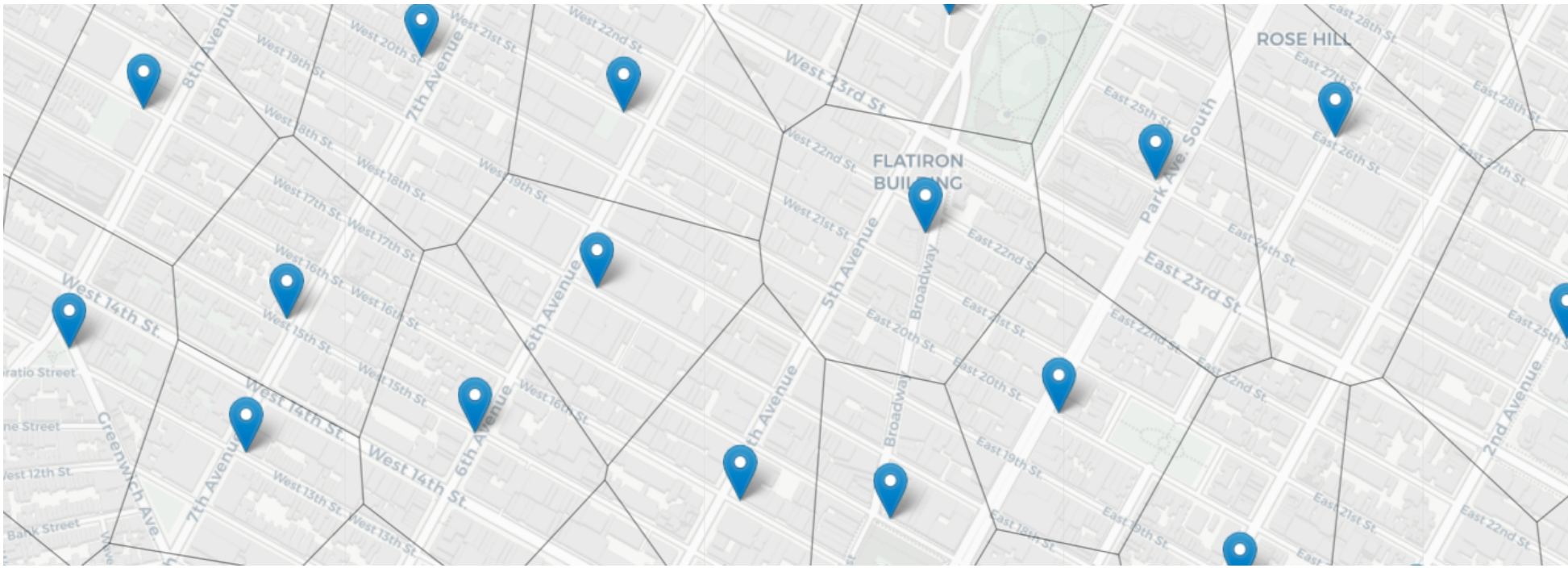
Worksheet: design an algorithm to find catchment areas for NYC libraries.

Design Challenge: Approaching Problems



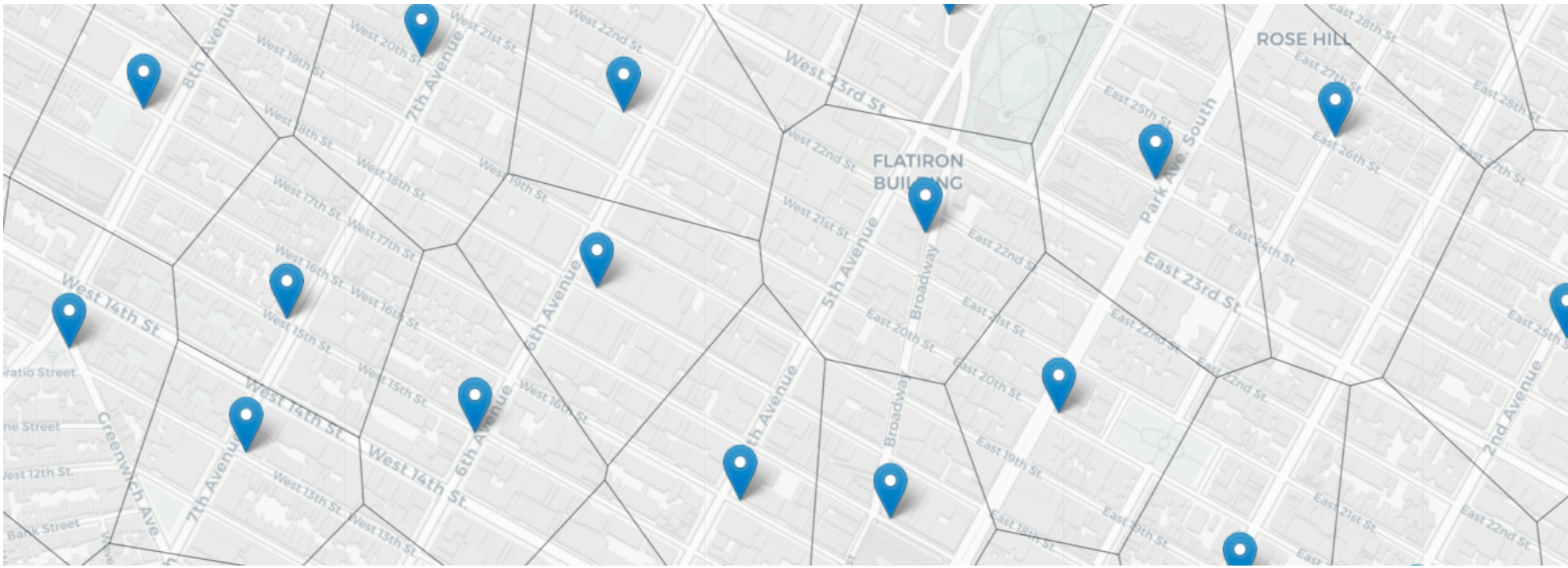
(From: *Cracking the Coding Interview*)

Design Challenge: Catchment Areas



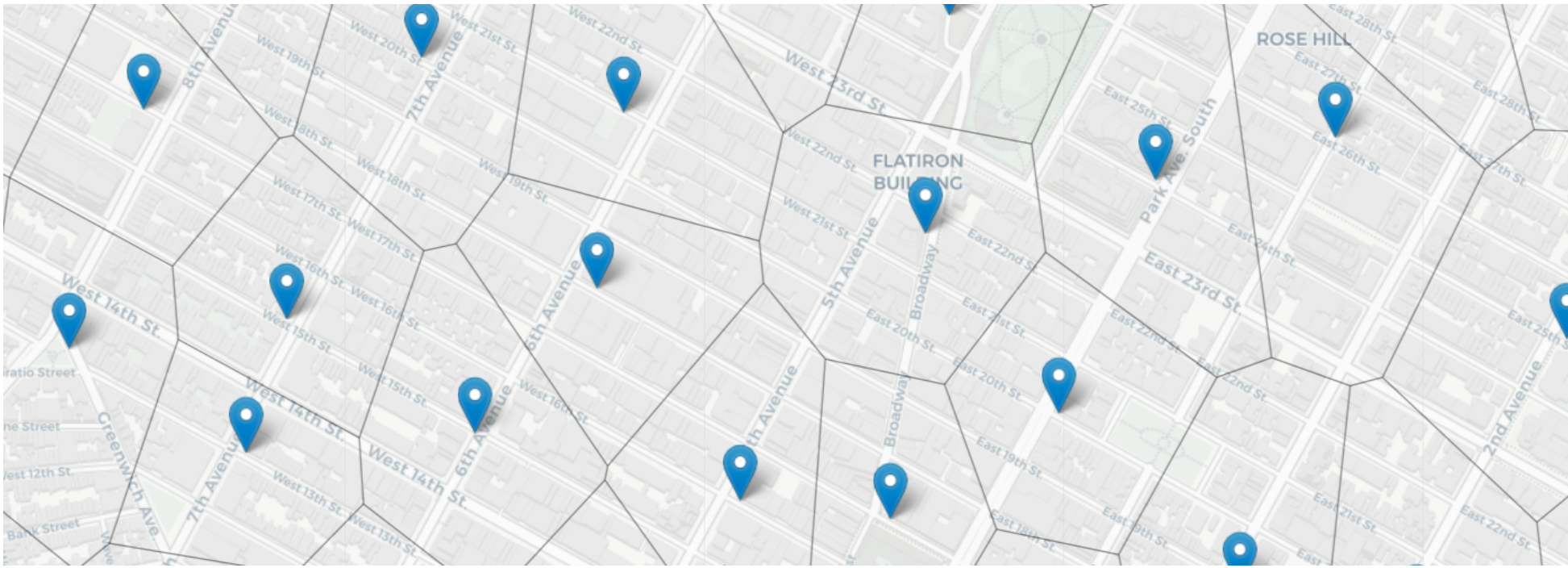
- Called **Voronoi Diagrams**.

Design Challenge: Catchment Areas



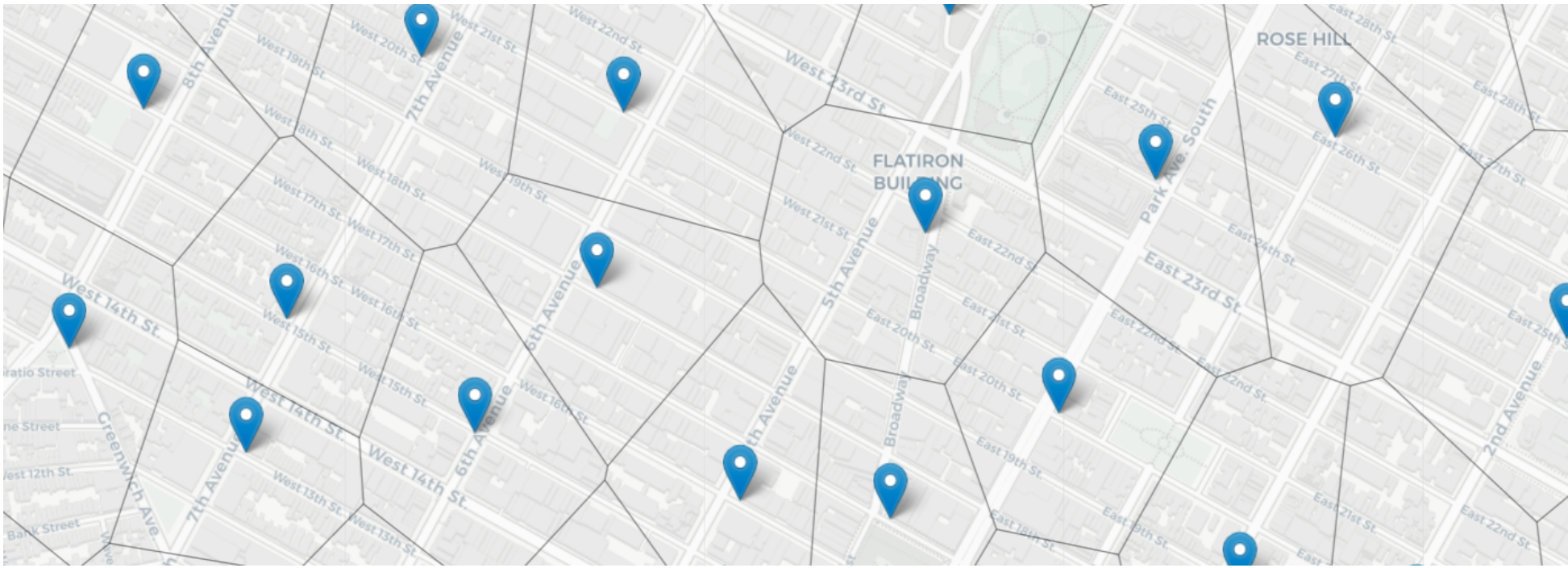
- Called **Voronoi Diagrams**.
- Can be computed in $O(n^2)$ time: many different approaches.

Design Challenge: Catchment Areas



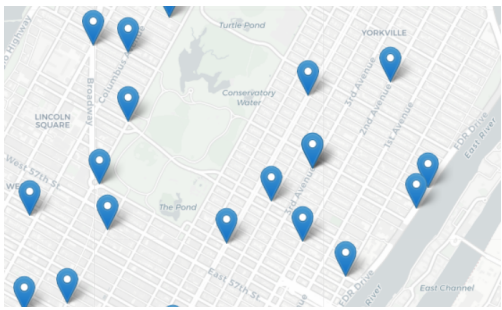
- Called **Voronoi Diagrams**.
- Can be computed in $O(n^2)$ time: many different approaches.
- Share your approaches.

Design Challenge: Catchment Areas



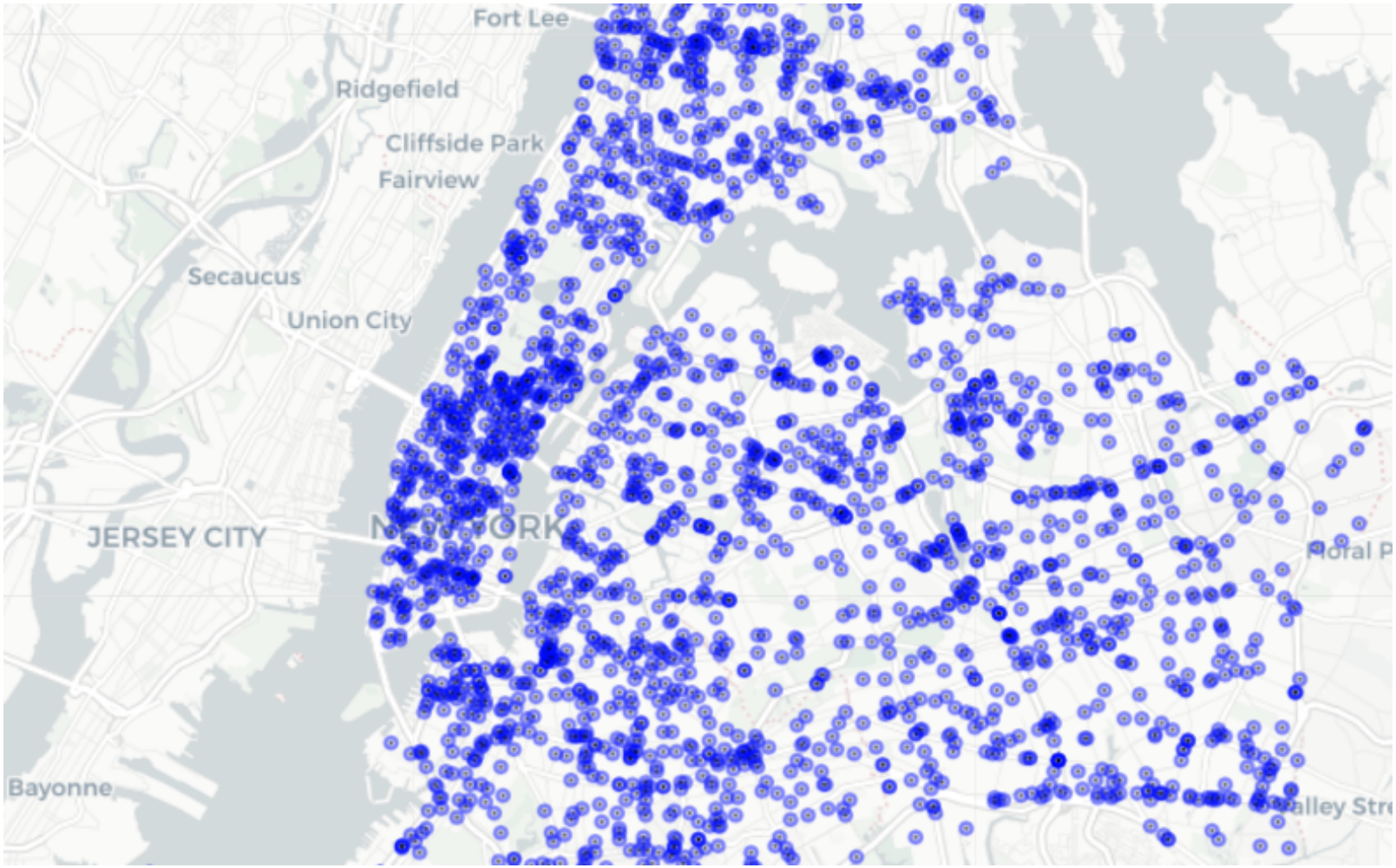
- Called **Voronoi Diagrams**.
- Can be computed in $O(n^2)$ time: many different approaches.
- Share your approaches. *(Links on webpage of some approaches.)*

Outline



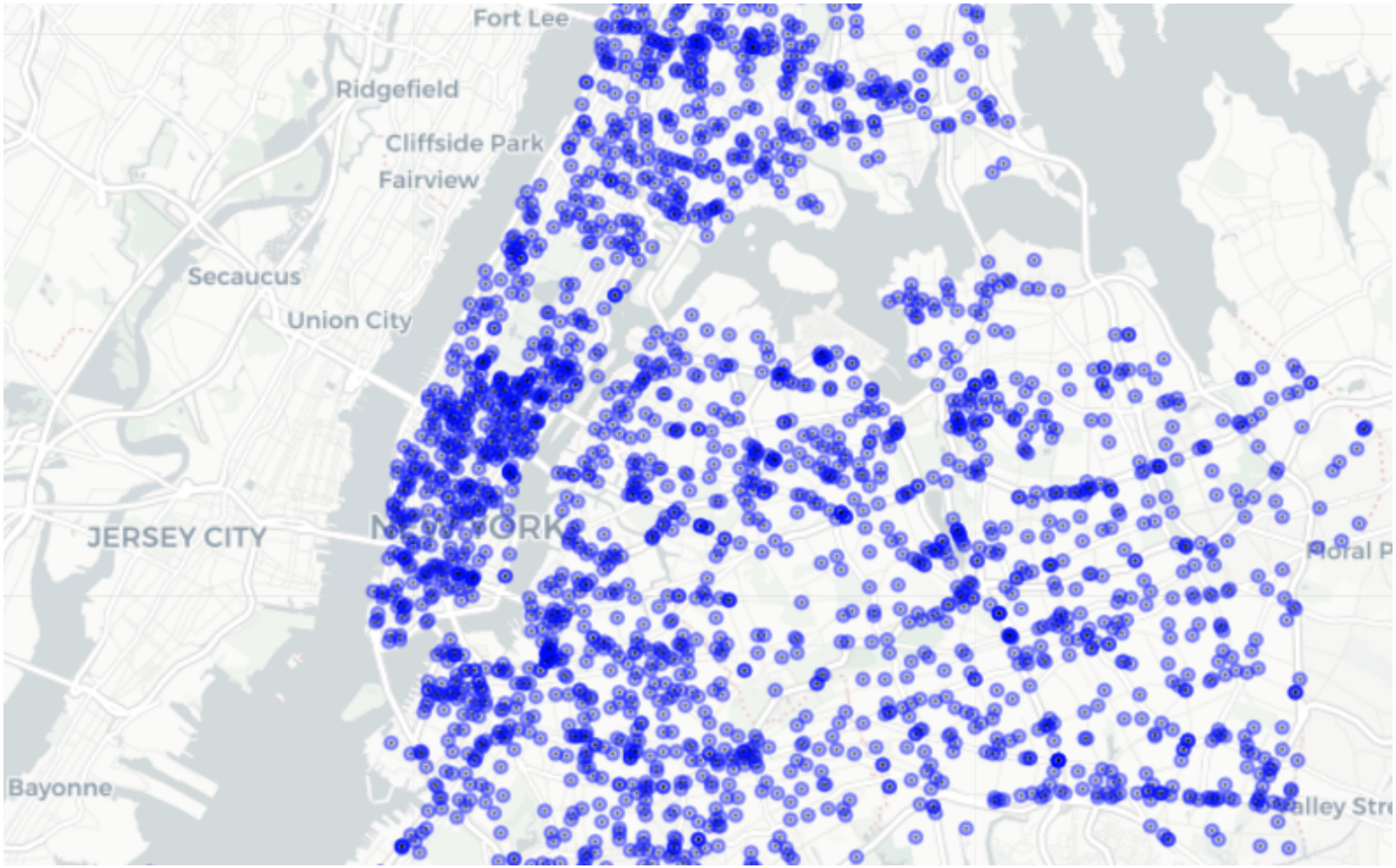
- Recap
- HTML-Scalable Maps: Folium
- Extracting Data
- geoJSON Format & Choropleth Maps
- Break
- Design Challenge: Catchment Areas
- **Design Challenge: Clustering Data**
- Wrap Up

Design Challenge: Clustering Data



You have 3 emergency service trucks.

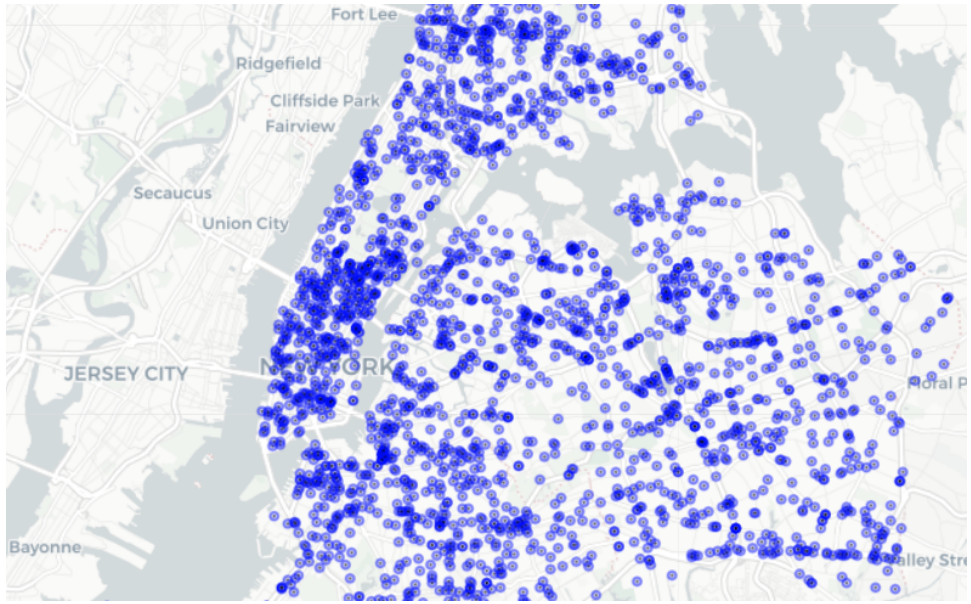
Design Challenge: Clustering Data



You have 3 emergency service trucks.

Where to put them to minimize distances to collisions?

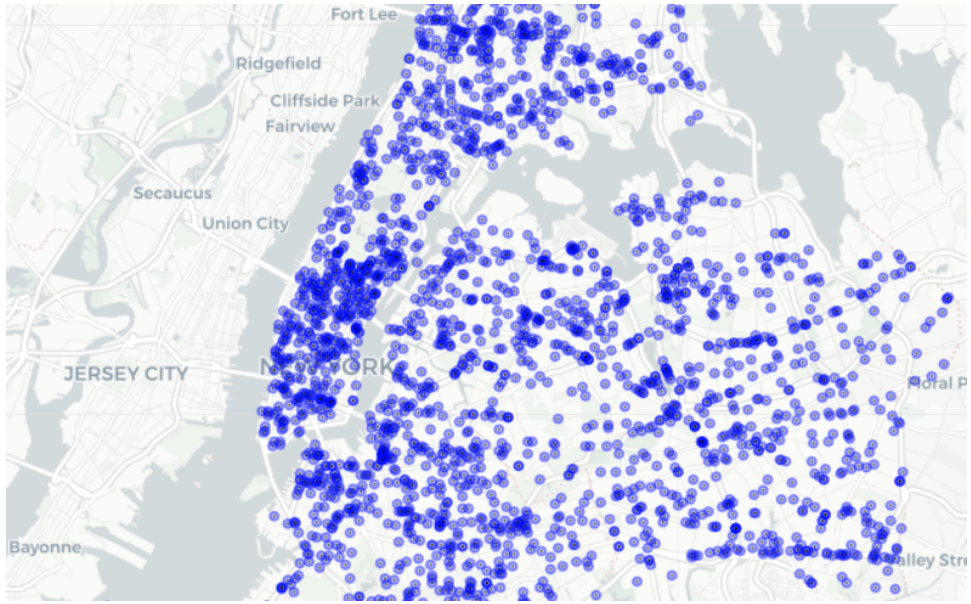
Design Challenge: Clustering Data



You have 3 emergency service trucks.

Where to put them to minimize distances to collisions?

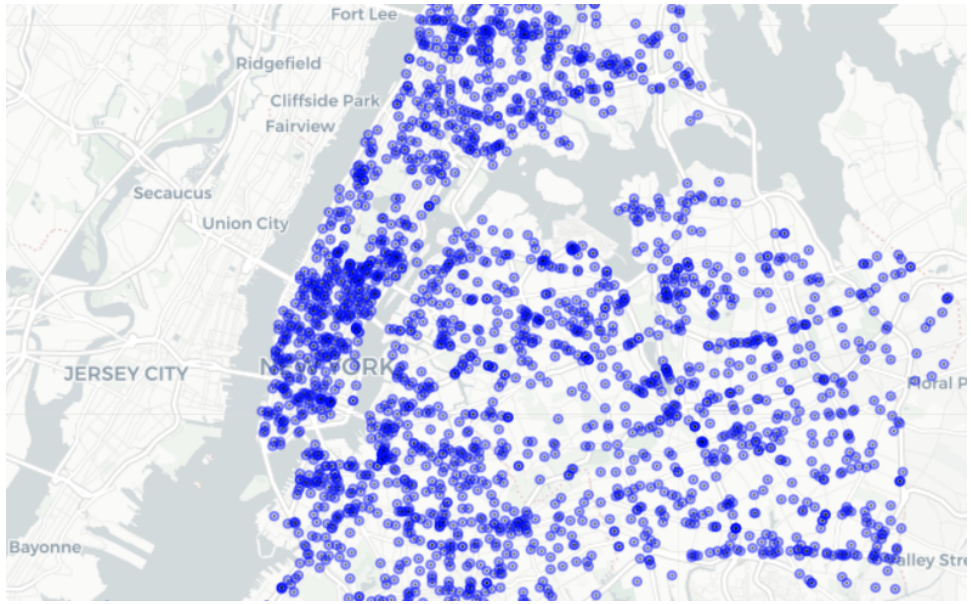
Design Challenge: Clustering Data



You have 3 emergency service trucks.
Where to put them to minimize distances to collisions?

- Called *k*-**means clustering**.

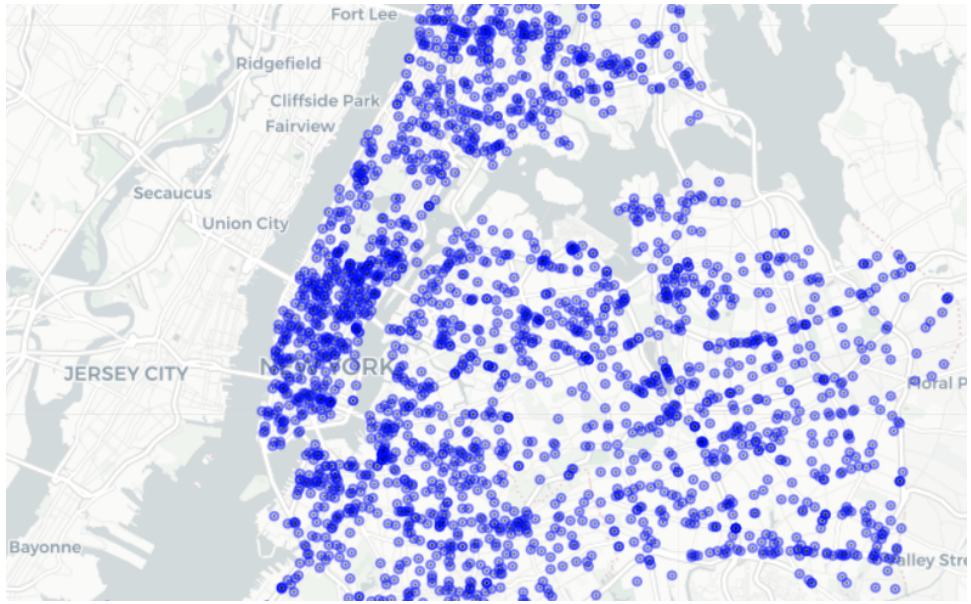
Design Challenge: Clustering Data



You have 3 emergency service trucks.
Where to put them to minimize distances to collisions?

- Called *k*-**means clustering**.
- Computationally hard to compute.

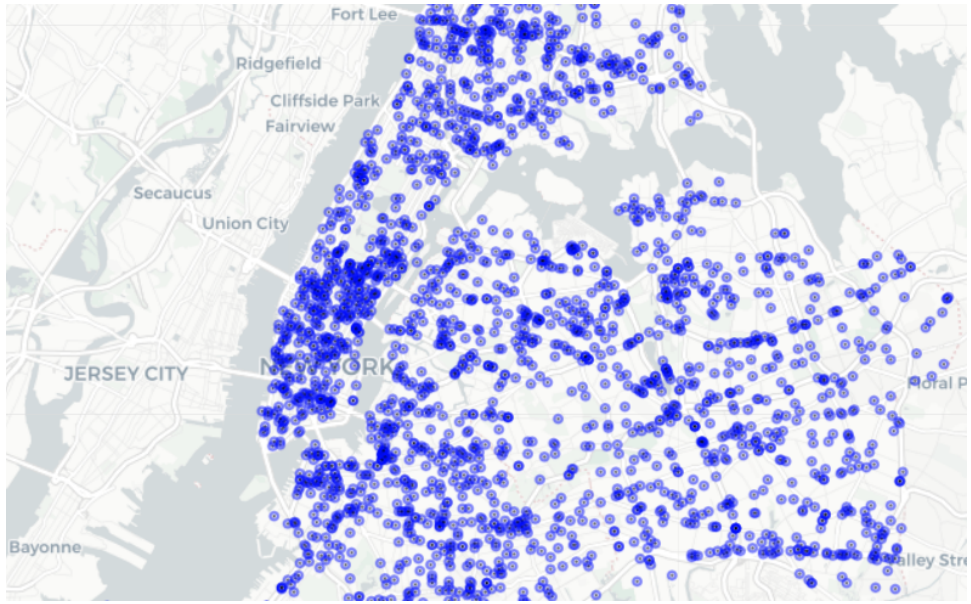
Design Challenge: Clustering Data



You have 3 emergency service trucks.
Where to put them to minimize distances to collisions?

- Called *k*-**means clustering**.
- Computationally hard to compute.
- Intuition for why: allowed to place the trucks anywhere (not restricted to inputted points) so many, many possible locations.

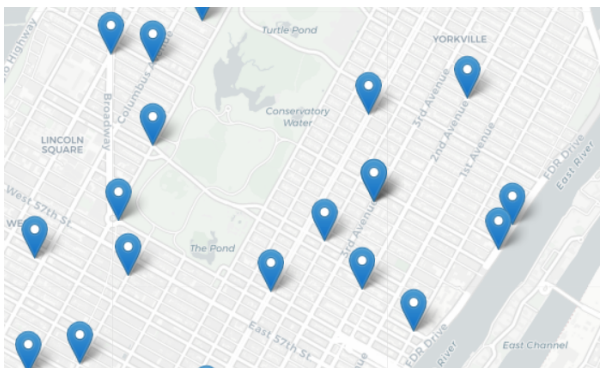
Design Challenge: Clustering Data



You have 3 emergency service trucks.
Where to put them to minimize distances to collisions?

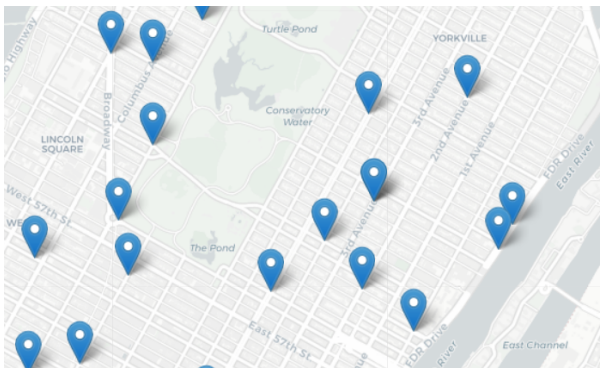
- Called *k*-**means clustering**.
- Computationally hard to compute.
- Intuition for why: allowed to place the trucks anywhere (not restricted to inputted points) so many, many possible locations.
- Approximations used instead.

Outline



- Recap
- HTML-Scalable Maps: Folium
- Extracting Data
- geoJSON Format & Choropleth Maps
- Break
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data
- **Wrap Up**

Wrap-Up



- Three sessions:
  1. Flood Maps (arrays & images)
  2. School Attendence (structured data, file I/O)
  3. Mapping Collisions (using objects, mapping coordinates)
- HTML-Scalable Maps: Folium
- Extracting Data: more on pandas
- geoJSON Format & Choropleth Maps
- Design Challenge: Catchment Areas
- Design Challenge: Clustering Data

Thank you!

(Image: NY Times)