Pyproj Practice - Mapping NYC's Water Distribution Sample Sites to Each Building

New York City has its water supplied from three different locations: *Delaware Water Supply System,* Croton Water Supply System, * Queens Groundwater Supply System. These water sources can be as far as 125mi from the point of use in the city. The water supplies are treated before traveling through the NYC distribution system to prevent waterborne pathogens which can cause sickness and death. NYC uses Sodium Hypochlorite (disinfectant) to prevent bacteria growth. The goal is to feed the disinfectant into the distribution system and maintain at least 0.5 mg/L until it reaches the water main of a building. To validate the effectiveness of this system, New York City has around 400 water distribution sample stations scattered throughout the city.

These sample stations are monitored to ensure proper disinfectant levels and also proper contaminant and bacteria levels. Through the Mayor's Office of Data Analyitics, NYC provides a large amount of information used by the city government to a public domain (OpenDataNYC). This includes sample station locations, water test results and individual building locations/parameters across the city.

The goal is to map each building to its closest sample station to get an idea of what the incoming water to that building looks like.

In [7]:
```
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import pyproj
from pyproj import Proj, transform
from scipy.spatial.distance import euclidean
from folium.plugins import FastMarkerCluster
import folium
```

Importing in the Data

Below we import in the sample station location data, the sample station water quality data and building location data.

In [8]:
```
df = pd.read_csv('data/drinking-water-quality-distribution-monitoring-data.csv',
df_ss = pd.read_excel('data/sample_sites.xlsx').drop('Sample Station (SS) - Loca
df_building = pd.read_csv('data/building_fin.csv').drop(['Unnamed: 0','the_geom'
```

Locating the Sample Stations

In [9]:
```
df_ss.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 401 entries, 0 to 400
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Sample Site     401 non-null    object
 1   X - Coordinate  401 non-null    int64 
 2   Y - Coordinate  401 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 9.5+ KB
```

So the sample station dataframe contains location coordinates but they are not in lat/lon. A little bit of research shows that this coordinate system is a projected coordinate system for New York City.

"*A projected coordinate system is a flat, two-dimensional representation of the Earth. It is based on a sphere or spheroid geographic coordinate system, but it uses linear units of measure for coordinates, so that calculations of distance and area are easily done in terms of those same units.*" IBM

The building locations are in lat/lon so the sample station locations needs to be transformed to lat/lon. This Article goes over how to convert projected coordinates to latitude and longitude using R. To do the same transformation in Python there is a library called pyproj. Pyproj is a cartographic projections and coordinate transformations tool. To convert the coordinates of the sample stations to lat/lon the CRS module is used. This is a pythonic Coordinate Reference System manager where the input coordinate system can be given and then a specific coordinate transformation requested.

The below code transforms the coordinates of the sample stations into lat/lon and attaches the lat/lon location to the water quality dataset.

In [10]:
```
def to_lat_long(x,y):
```

```
    '''
    Transforms EPSG 2263 x,y coordinates to latitude and longitude
    '''

    proj = pyproj.Transformer.from_crs(2263, 4326, always_xy=True)
    x1, y1 = (x, y)
    x2, y2 = proj.transform(x1, y1)
    return x2, y2

# Runs function on each sample site to generate lat/lon and creates a dictionary
location_dict = {}
for s,x,y in zip(df_ss['Sample Site'],df_ss['X - Coordinate'],df_ss['Y - Coordin
    x2,y2= to_lat_long(x,y)
    location_dict[s] = (y2,x2)

df['Location'] = df['Sample Site'].map(lambda x: location_dict[x])
```

The below code finds the assigns the nearest sample station to each building using euclidean distance. There are about 108,000 buildings in the dataframe and the code takes around 87min to run. The updated dataframe is then stored in a new .csv file

```
def find_closest_sample_site(building_location):
    '''
    compares a latitude and longitude to the sample station
dictionary and returns the closest sample station
    '''
    min_dist = 10000
    min_sample_site = ''
    for key in location_dict:
        if euclidean(location,location_dict[key]) < min_dist:
            min_dist = euclidean(location,location_dict[key])
            min_sample_site = key
    return min_sample_site

# runs the above function on the building dataframe to assign the
closest sample site to each building
df_building['Location'] = df_building['the_geom'].map(lambda x: (
float(x.split()[2][:-1]) , float(x.split()[1][3:]) ))
df_building['Sample Site'] =
df_building['Location'].map(find_closest_sample_site)

df_building.to_csv('data/building_fin.csv')
```

Once each building has an assigned sample station the most recently available water quality data can be merged with building data

In [12]:
```
df_building_FRO = df_building.merge(df,how='left',on='Sample Site')
```

Shown below is each building's lat/lon with it's closest sample station's most recent test results. The dataset now has generalized incoming water quality for each building in New York City.

In [14]:
```
df_building_FRO.head()
```

Out[14]:

	CNSTRCT_YR	HEIGHTROOF	FEAT_CODE	GROUNDELEV	SHAPE_AREA	SHAPE_LEN	BASE_E
0	1925.0	29.749853	2100.0	40.0	0	0	3.065220e+
1	1925.0	29.749853	2100.0	40.0	0	0	3.065220e+
2	1925.0	29.749853	2100.0	40.0	0	0	3.065220e+
3	1925.0	29.749853	2100.0	40.0	0	0	3.065220e+
4	1925.0	29.749853	2100.0	40.0	0	0	3.065220e+

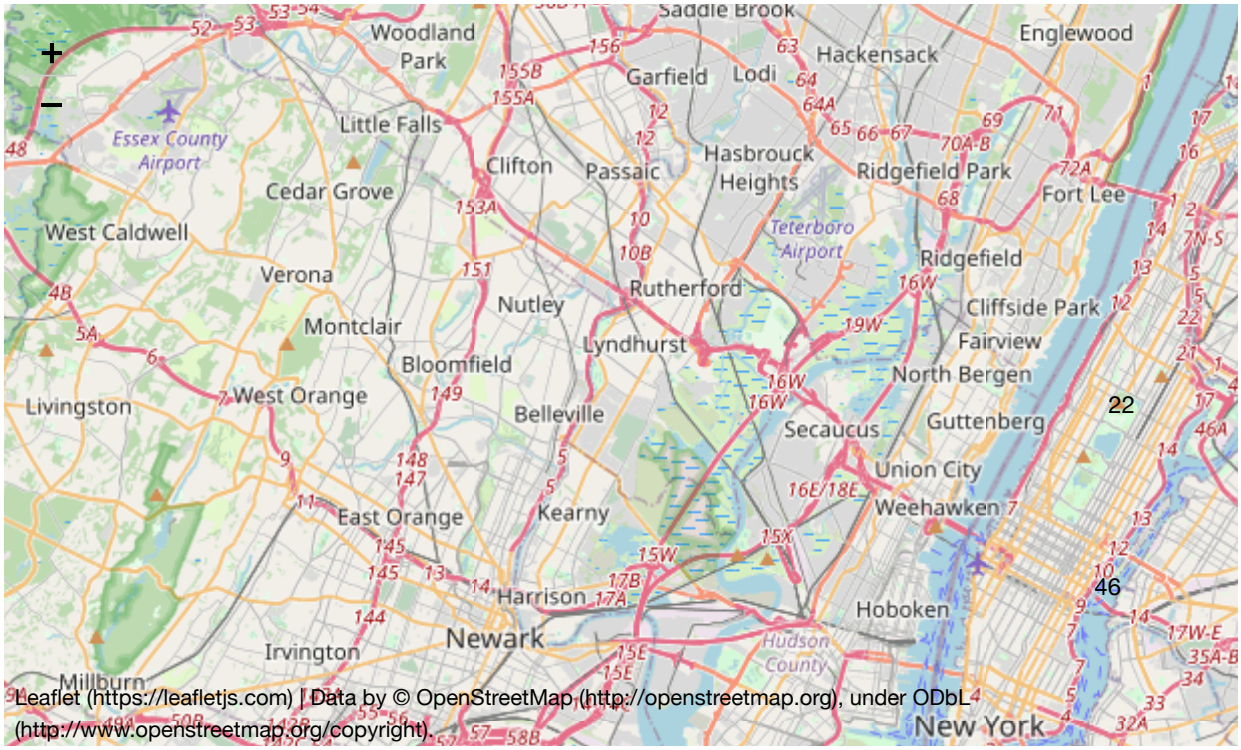
5 rows × 21 columns

Map of Sample Stations

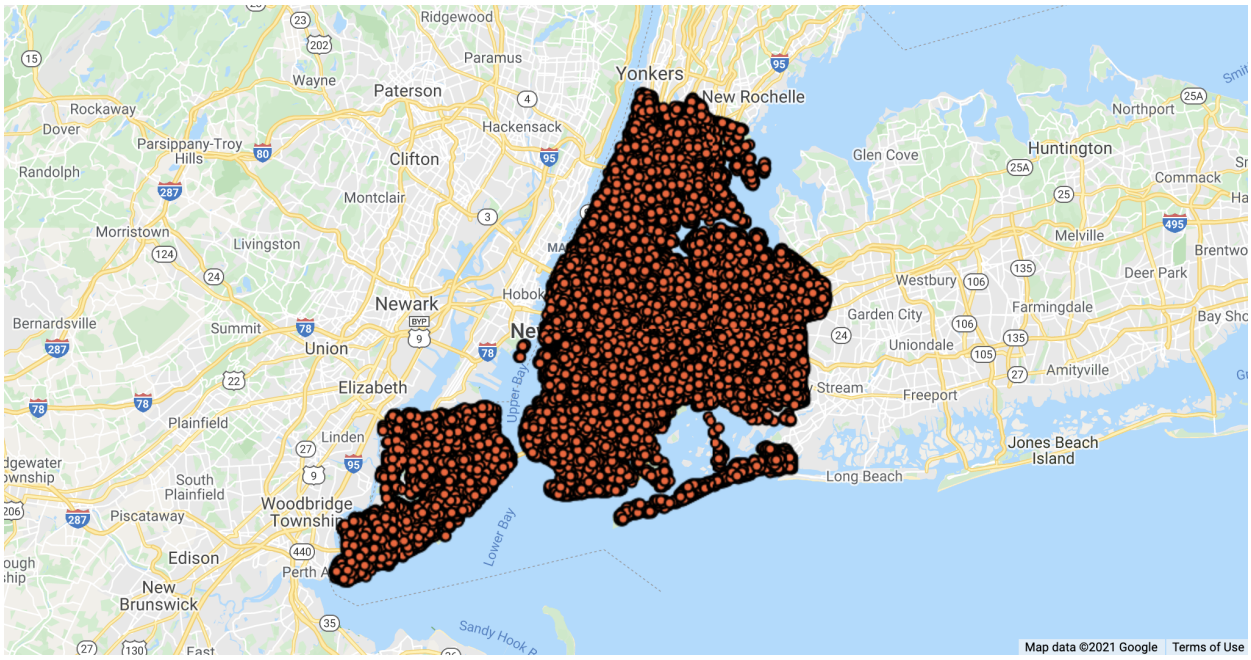
In [ ]:
```
lat = 40.7
long = -73.9
# Initialize a folium map to plot points
my_map = folium.Map([lat, long], zoom_start=11)
my_map.add_child(FastMarkerCluster(df.drop_duplicates(subset='Sample Site')[['Lo

my_map
```

Out[ ]:



Map of New York City Buildings



Summary

Building Features along with their incoming water quality would be a start towards developing a risk profile of each building with respect to water safety. There are still many more input variables that would be necessary to properly identify if a building is at risk for growing bacteria in its water system.

Citations

https://health.data.ny.gov/Health/Registered-Cooling-Tower-Map/unmf-baqa

https://data.cityofnewyork.us/Housing-Development/Building-Footprints/nqwf-w8eh

https://data.cityofnewyork.us/Health/Rooftop-Drinking-Water-Tank-Inspection-Results/gjm4-k24g/data

https://data.cityofnewyork.us/Environment/Drinking-Water-Quality-Distribution-Monitoring-Dat/bkwf-xfky/data

https://aem.run/posts/2021-07-02-learning-the-basics-of-gis-mapping-with-leaflet/