

NYC Trees

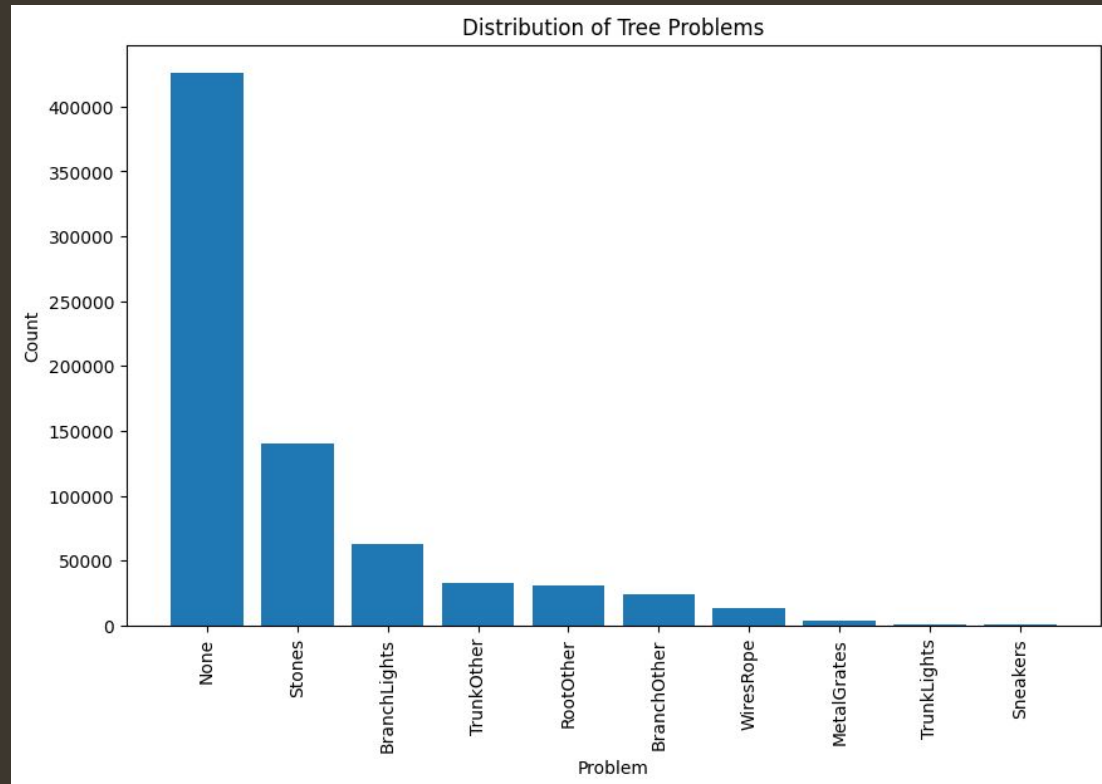
Project SQL



Research Questions

- What are some common tree problems in NYC?
- Is there a measurable difference in DBH measured over a 10 year span?
- Can we visualize DBH in variation in different parts of the city?
- Where is data being collected in the city and where is it not? See heat map
- How many trees per borough
- Can we estimate a trees age using DBH? How would this be distributed across a target genus with multiple representative species?
- Which trees have a nice red fall color?
- Can this data be used to understand areas with a similar latitude? (research lead)

Typical Tree Problems in 2015 Survey

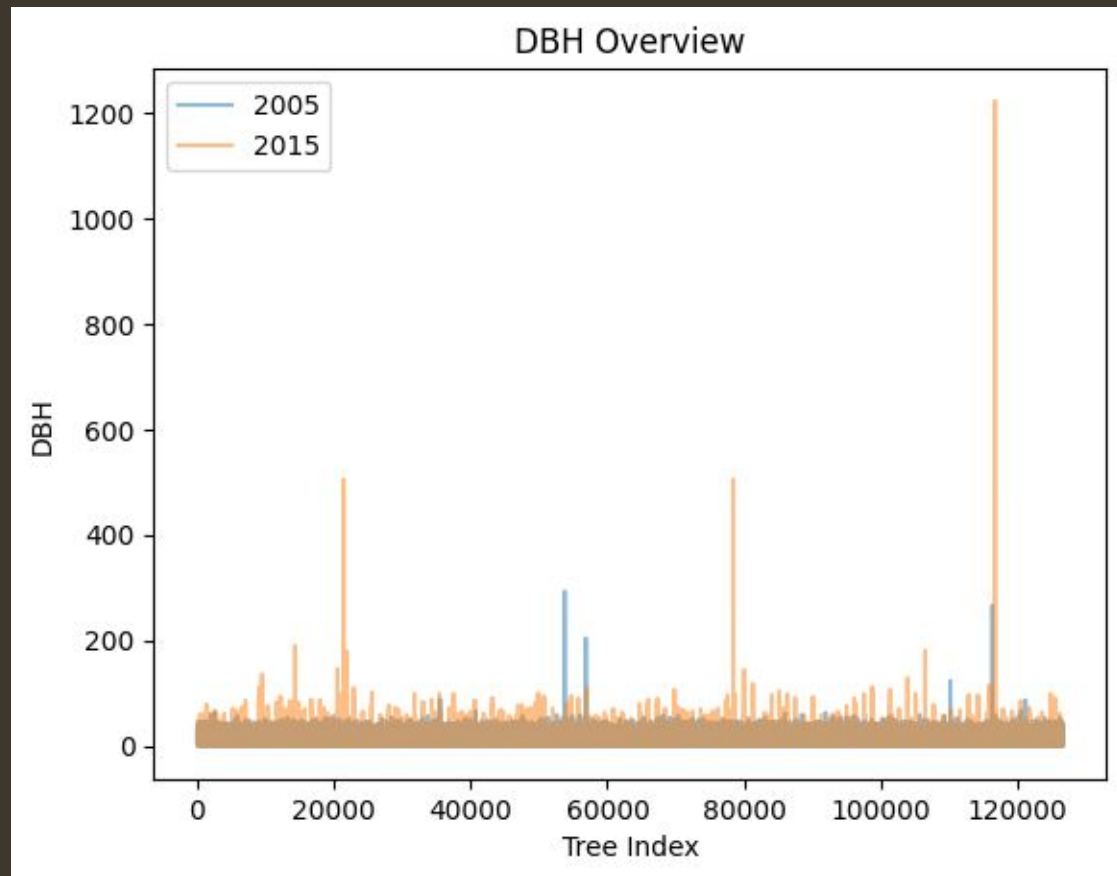


```
environmental_tolerance_query_1 = ""  
    SELECT DISTINCT environmental_tolerances  
    FROM `bigquery-public-data.new_york_trees.tree_species`;  
    ""
```

```
environmental_tolerances  
Drought and Pollution Tolerant  
    none  
    Shade Tolerant  
    High pH Tolerant  
Wet Site and Salt Tolerant
```


DBH between 2005 and 2015

```
id_join_query = """
SELECT *
FROM bigquery-public-data.new_york_trees.tree_census_2005 AS t_2005
JOIN bigquery-public-data.new_york_trees.tree_census_2015 AS t_2015 ON t_2005.objectid = t_2015.tree_id
"""
```



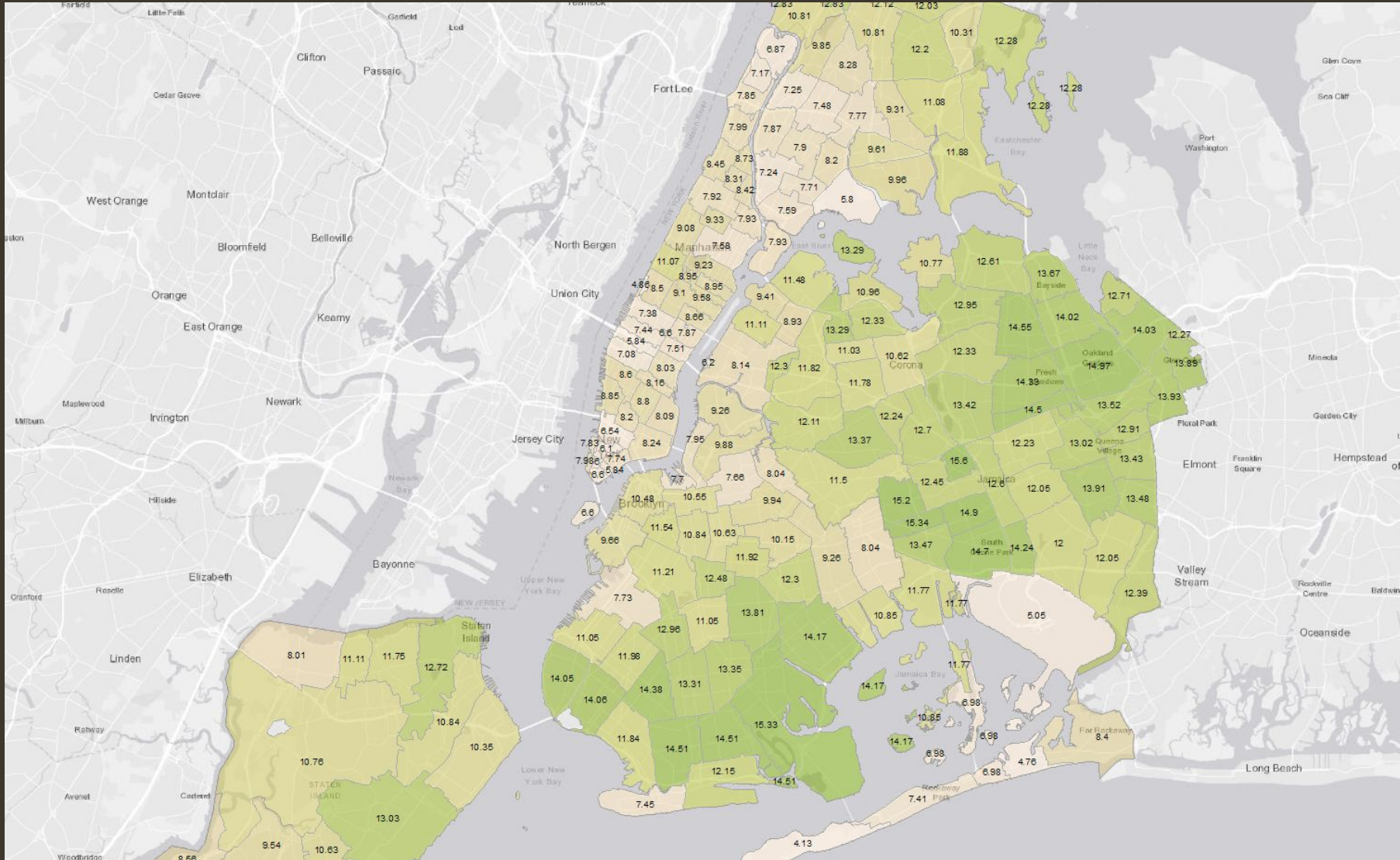
```
1 dbh_comparison = id_join_results[['objectid', 'tree_dbh_1', 'tree_dbh']]
2
3 plt.plot(dbh_comparison['tree_dbh_1'], label='2005', alpha= 0.5)
4 plt.plot(dbh_comparison['tree_dbh'], label='2015', alpha= 0.5)
5
6 plt.xlabel('Tree Index')
7 plt.ylabel('DBH')
8 plt.title('DBH Overview')
9
10 plt.legend()
11
12 plt.show()
```



Spatial Analysis

```
zip_dia_query = ""
```

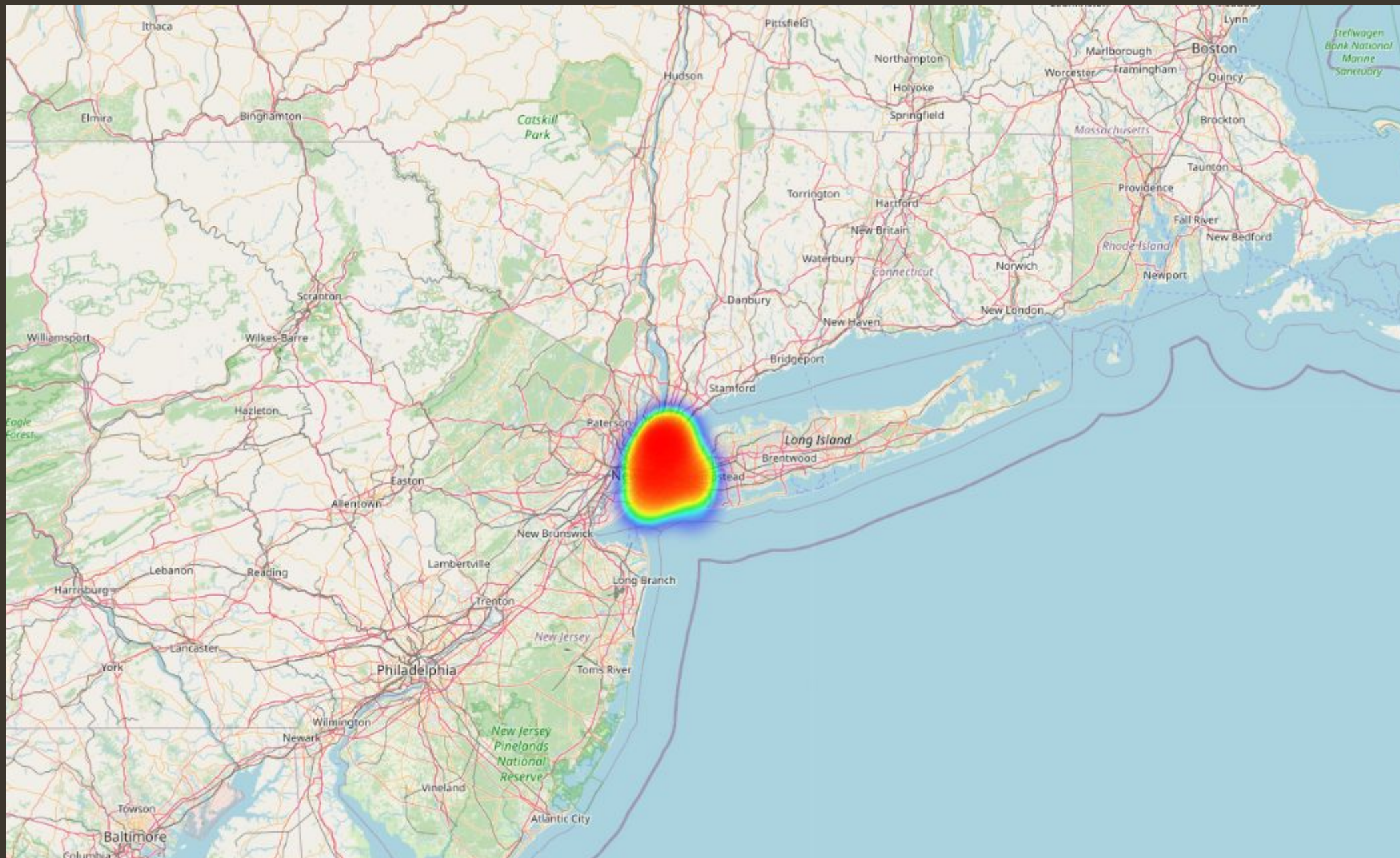
```
SELECT zipcode, AVG(tree_dbh) AS average_diameter  
FROM `bigquery-public-data.new_york_trees.tree_census_2015`  
GROUP BY zipcode;  
""
```

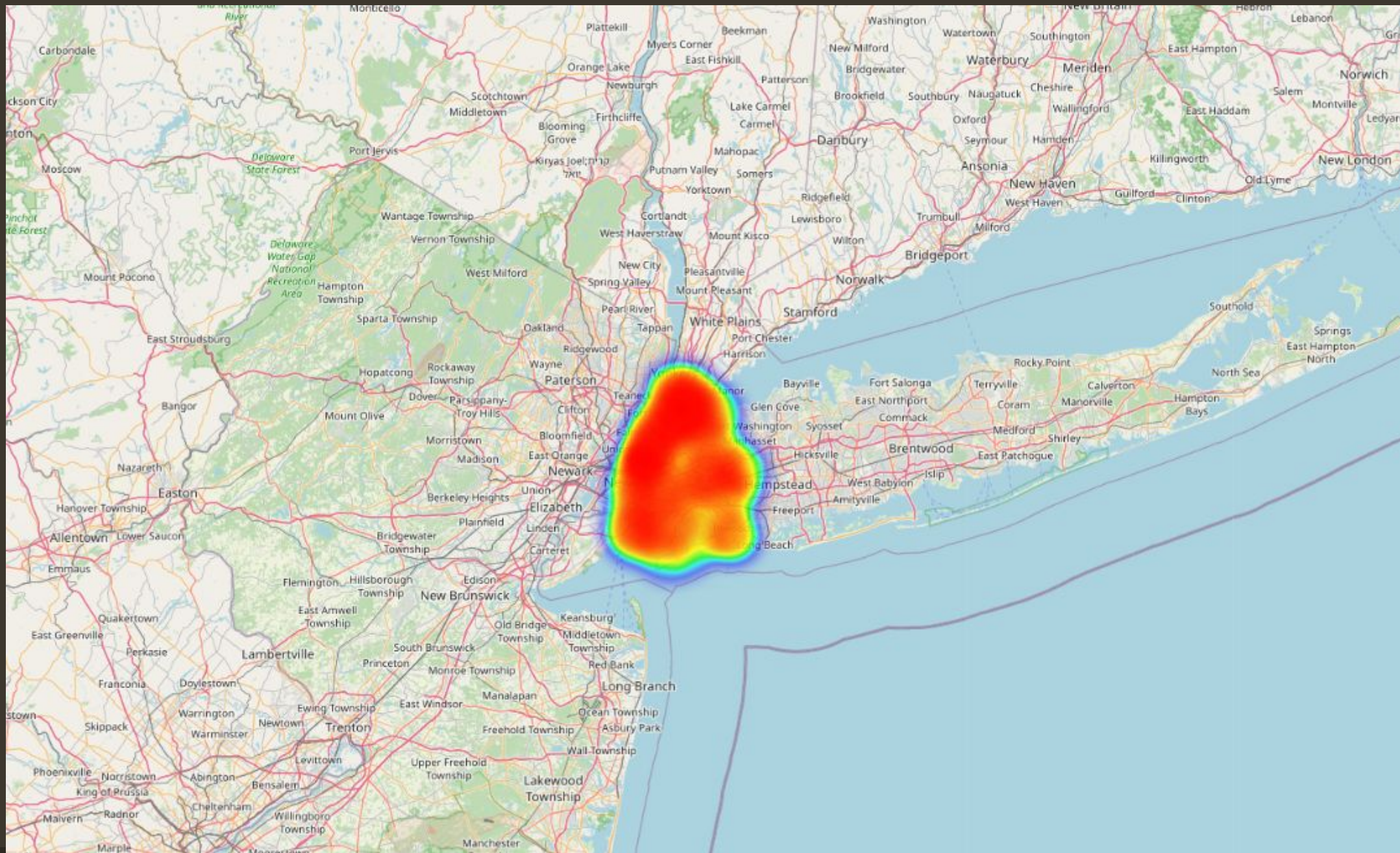


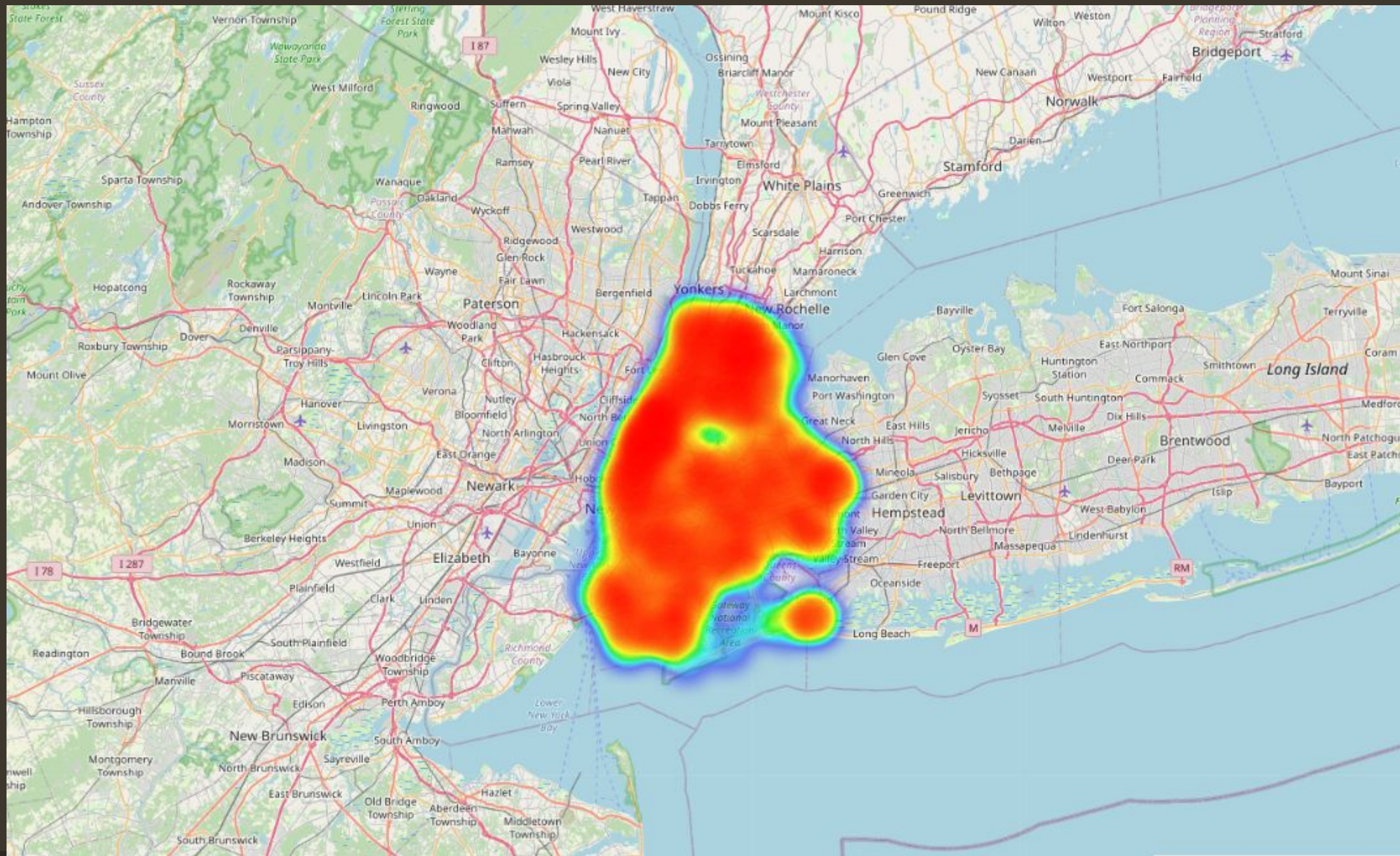


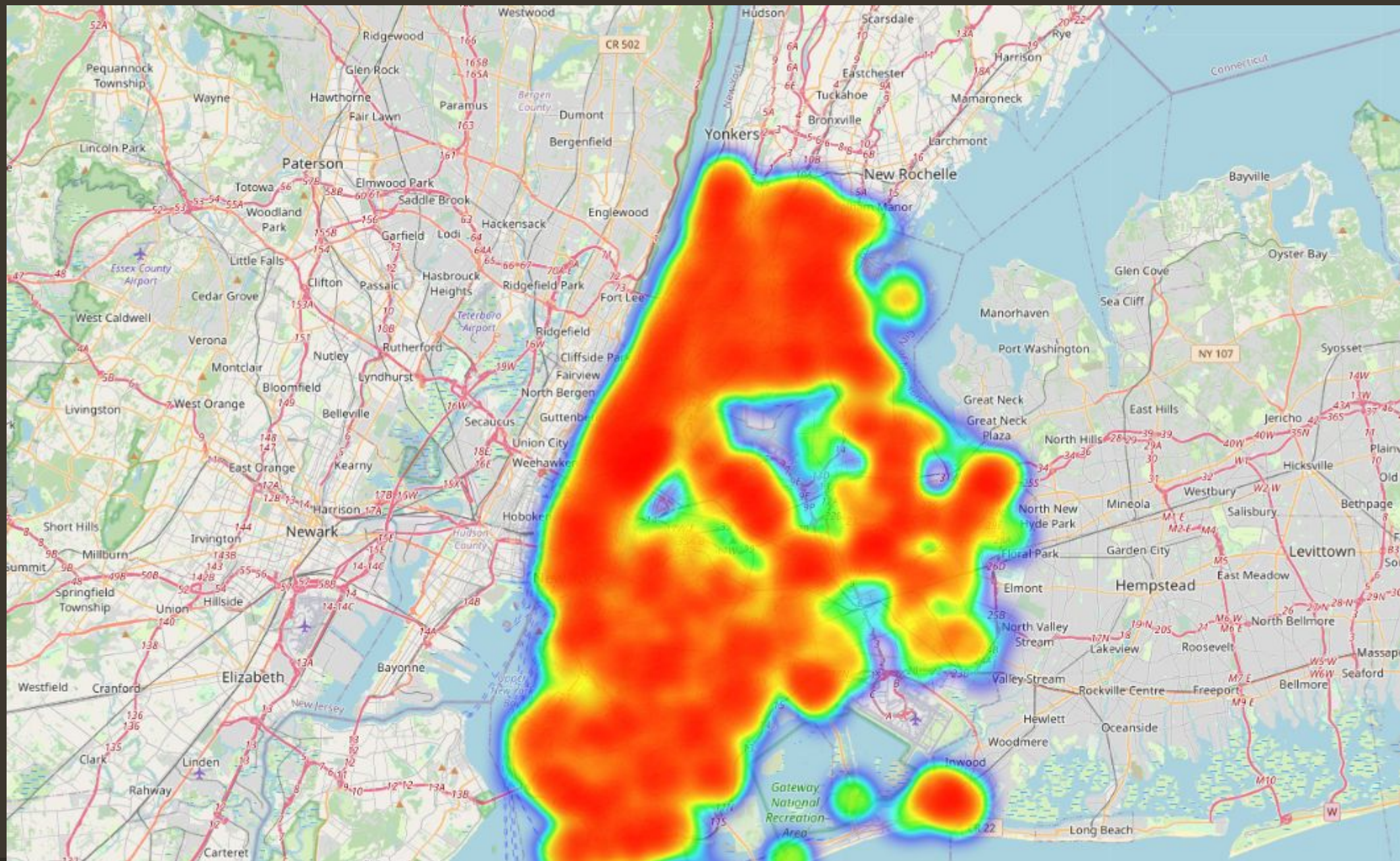
Heat Map

```
1 # Create a map centered around the mean latitude and longitude
2 map_center = [id_join_results['latitude'].mean(), id_join_results['longitude'].mean()]
3 m = folium.Map(location=map_center, zoom_start=12)
4
5 # Create a list of coordinate points from the latitude and longitude columns
6 locations = id_join_results[['latitude', 'longitude']].values
7
8 # Create a heatmap layer using the coordinate points
9 heatmap_layer = HeatMap(locations)
10
11 # Add the heatmap layer to the map
12 heatmap_layer.add_to(m)
13
14 # Display the map
15 m
```

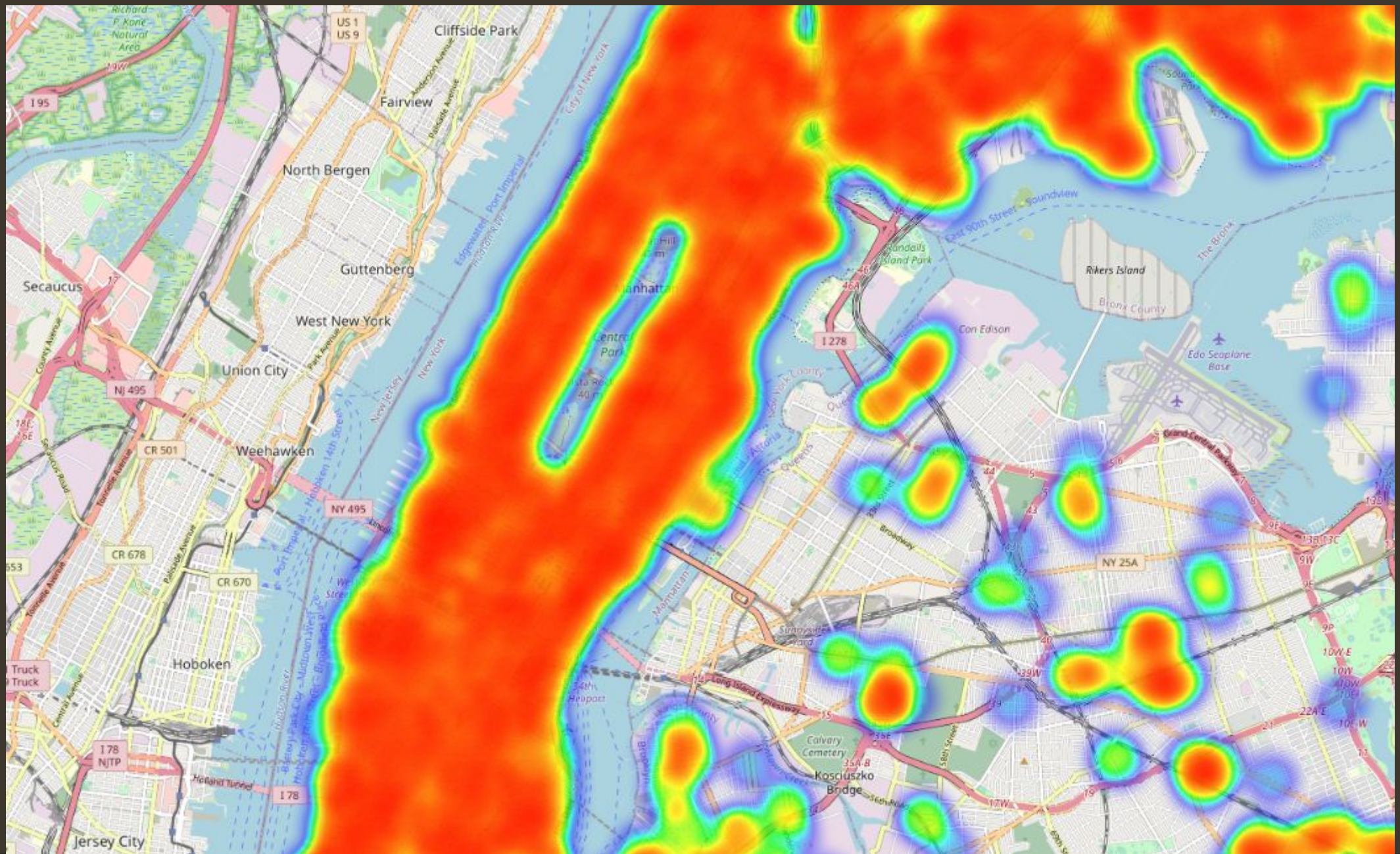







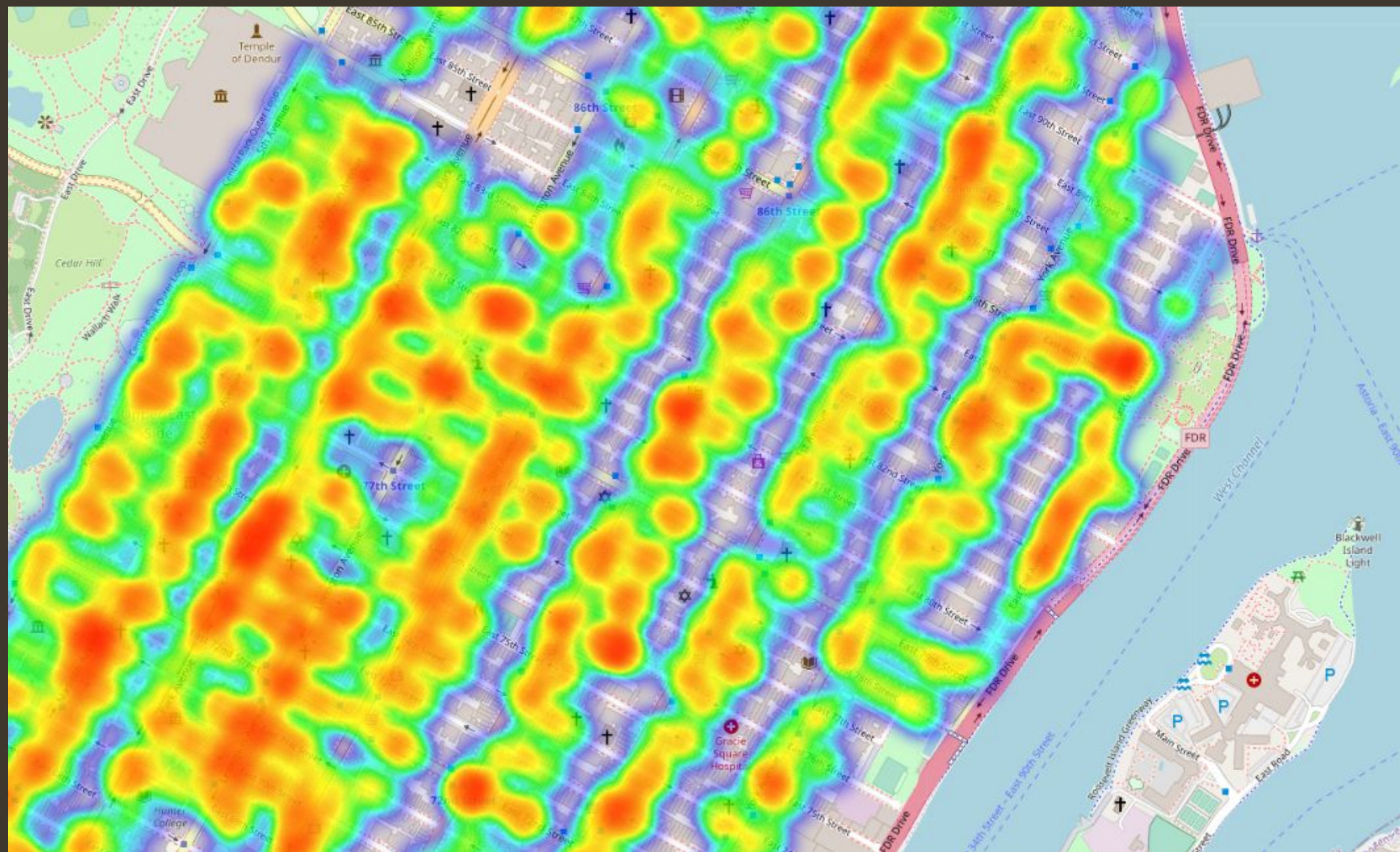


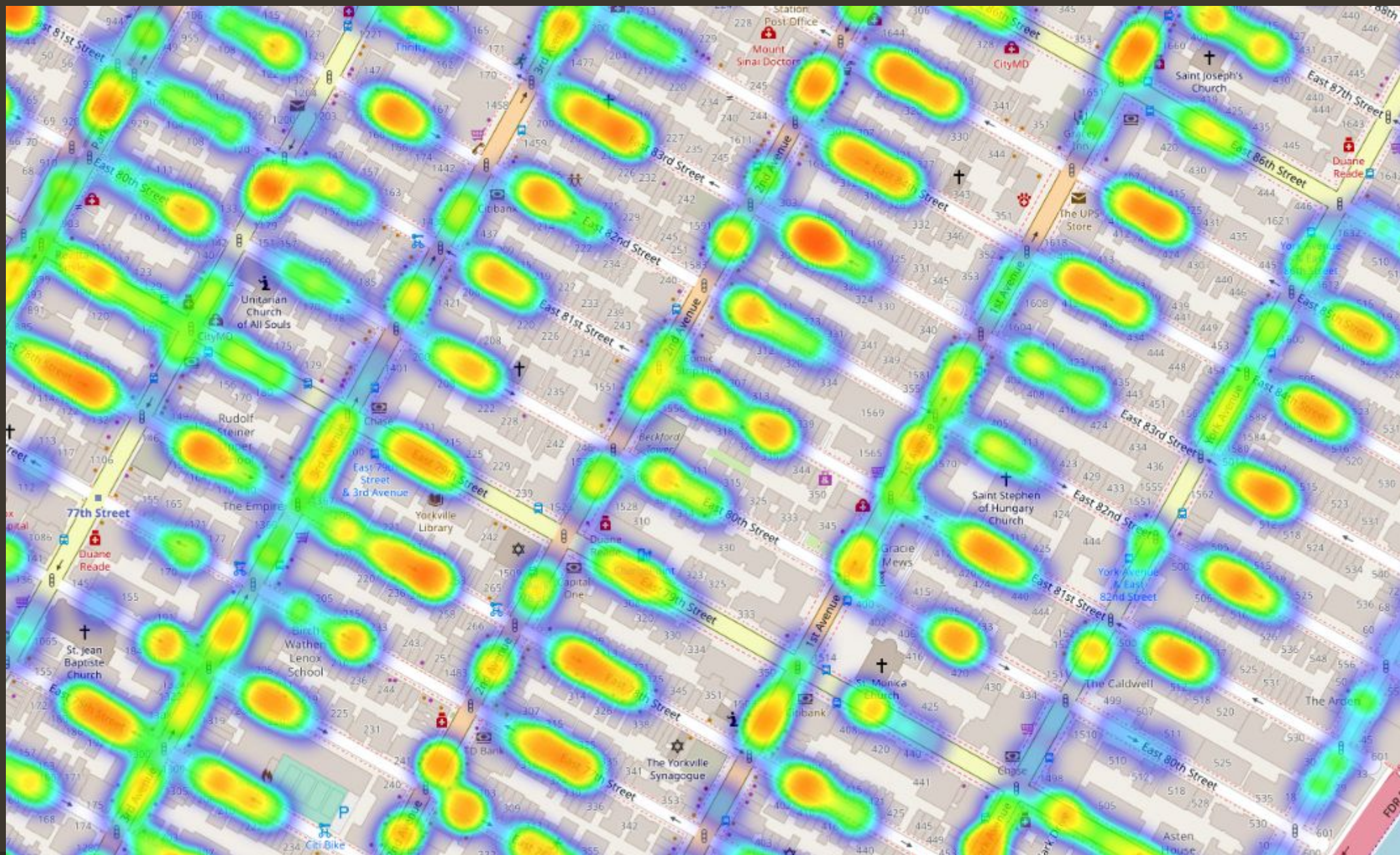






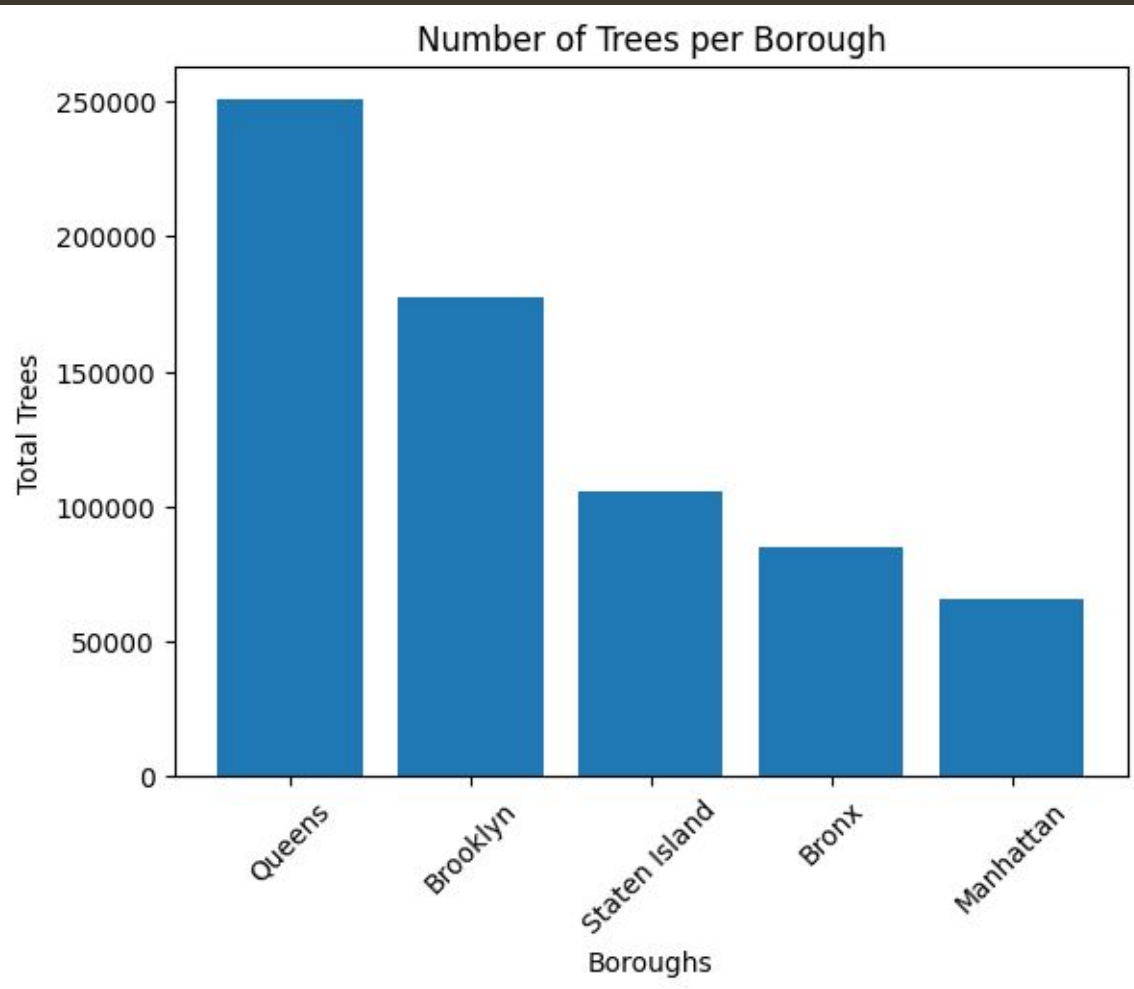








Trees per Borough



```
tree_boro_query = """
SELECT boroname, COUNT(*) AS total_trees
FROM `bigquery-public-data.new_york_trees.tree_census_2015`
GROUP BY boroname;
"""
```

boroname	total_trees
Bronx	85203
Queens	250551
Brooklyn	177293
Manhattan	65423
Staten Island	105318

Oak Age

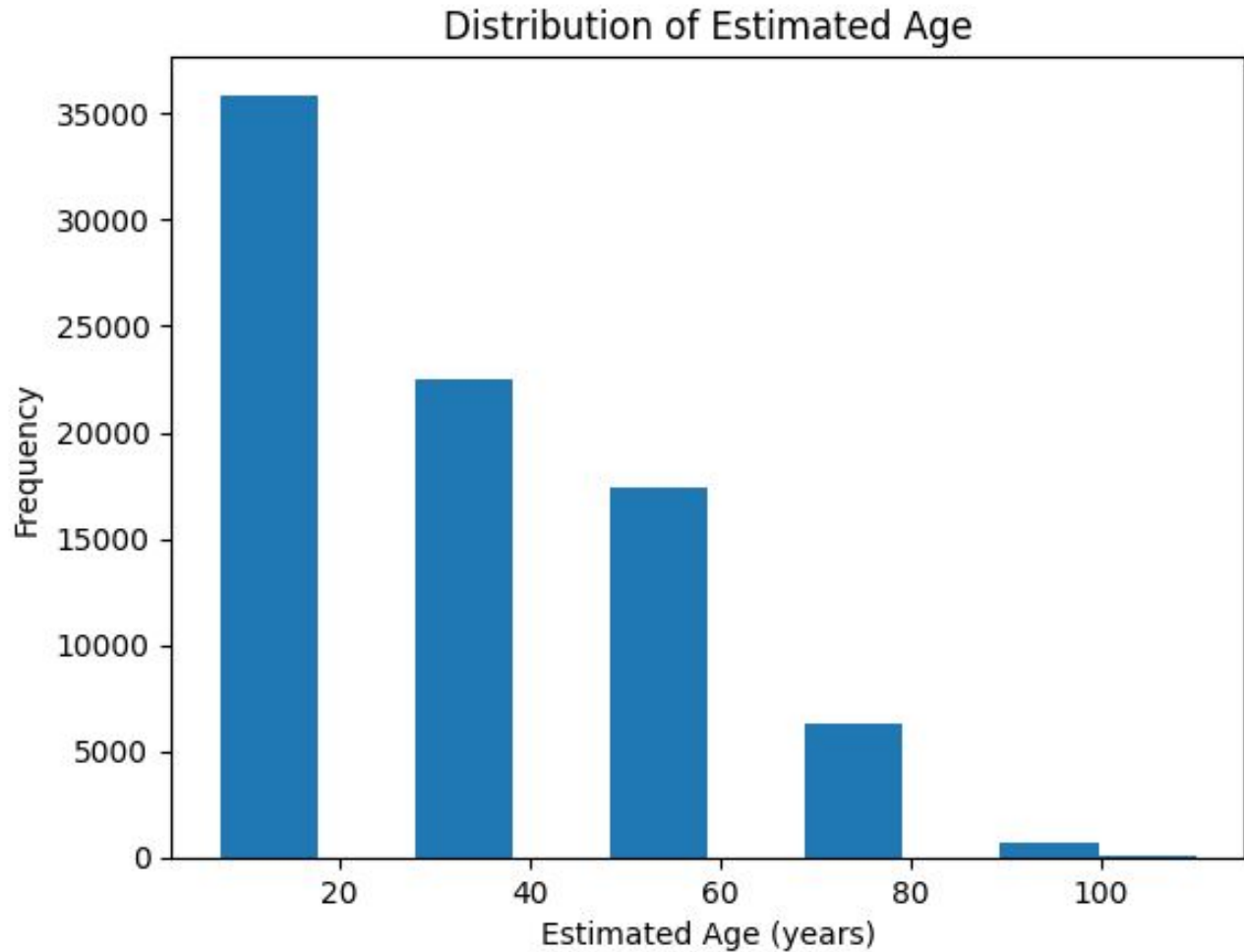
```
oak_query = """
SELECT tree_dbh, spc_latin
FROM `bigquery-public-data.new_york_trees.tree_census_2015`
WHERE spc_latin LIKE 'Quer%';
"""
```

	tree_dbh	spc_latin
0	39	Quercus phellos
1	1	Quercus palustris
2	38	Quercus phellos
3	1	Quercus acutissima
4	37	Quercus velutina
5	37	Quercus rubra
6	36	Quercus palustris
7	43	Quercus velutina
8	38	Quercus palustris
9	39	Quercus palustris
10	37	Quercus shumardii
11	36	Quercus palustris
12	38	Quercus palustris
13	39	Quercus palustris
14	36	Quercus palustris
15	39	Quercus palustris
16	40	Quercus rubra
17	1	Quercus coccinea
18	38	Quercus palustris
19	45	Quercus rubra
20	37	Quercus rubra
21	39	Quercus palustris

```
1 def estimate_oak_tree_age(dbh):
2     """
3     Estimate the age of an oak tree based on its Diameter at Breast Height (DBH).
4
5     Args:
6     |   dbh (float): Diameter at Breast Height (in inches).
7
8     Returns:
9     |   float: Estimated age of the oak tree (in years).
10    """
11
12    if dbh < 0:
13        raise ValueError("DBH must be a positive value.")
14
15    # Convert DBH from inches to centimeters
16    # dbh_cm = dbh * 2.54
17
18    # Determine the age range based on DBH
19    if dbh < 5:
20        age_range = (5, 10)
21    elif dbh < 10:
22        age_range = (10, 20)
23    elif dbh < 20:
24        age_range = (20, 40)
25    elif dbh < 30:
26        age_range = (40, 60)
27    elif dbh < 40:
28        age_range = (60, 80)
29    elif dbh < 50:
30        age_range = (80, 100)
31    else:
32        age_range = (100, 120)
33
34    # Calculate the estimated age within the age range
35    age_estimate = sum(age_range) / 2
36
37    return age_estimate
```

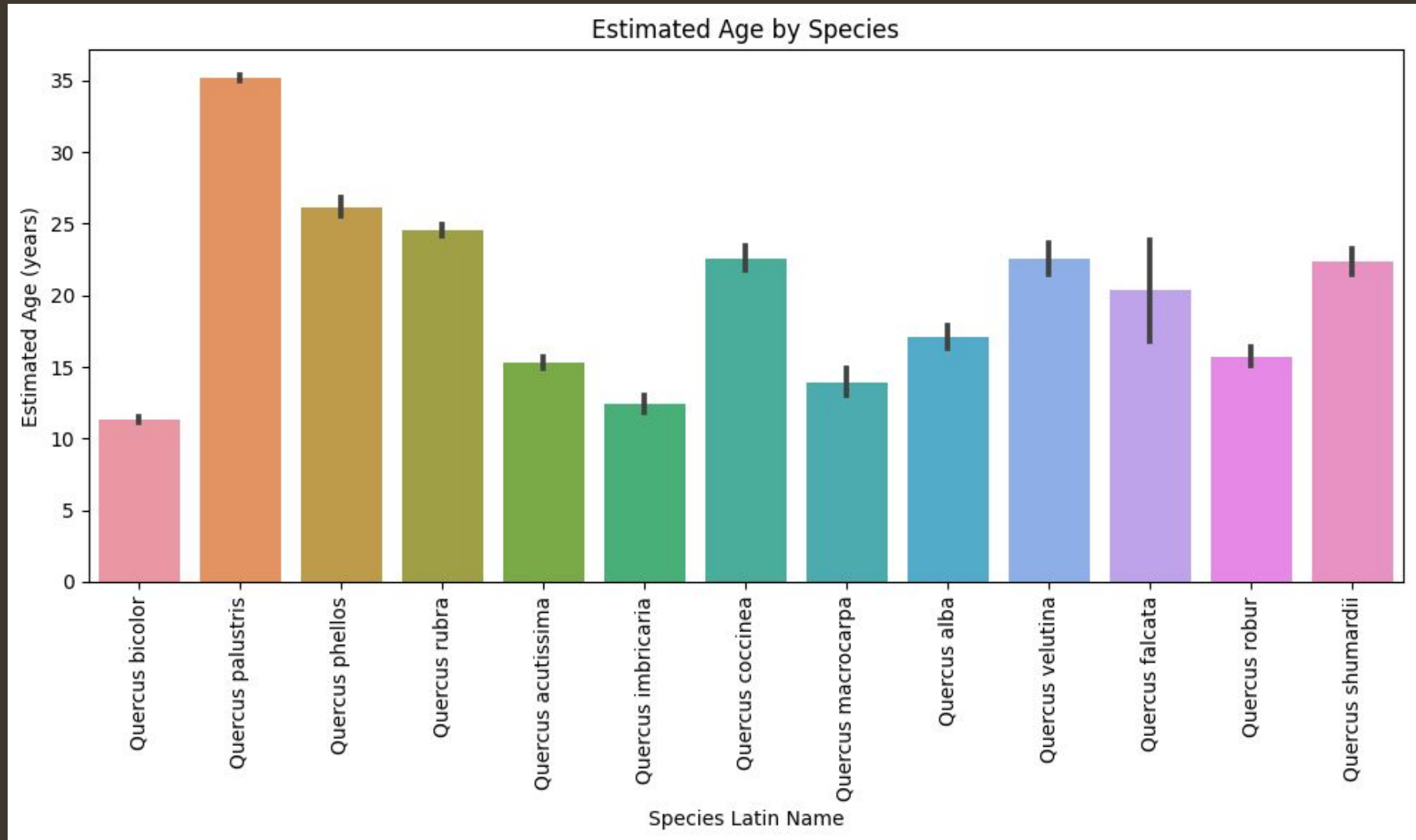

Oak Age

```
# Calculate estimated age for each oak tree
oak_results['estimated_age'] = oak_results['tree_dbh'].apply(estimate_oak_tree_age)
oak_results.head(30)
```



	tree_dbh	spc_latin	estimated_age
0	39	Quercus phellos	70.0
1	1	Quercus palustris	7.5
2	38	Quercus phellos	70.0
3	1	Quercus acutissima	7.5
4	37	Quercus velutina	70.0
5	37	Quercus rubra	70.0
6	36	Quercus palustris	70.0
7	43	Quercus velutina	90.0
8	38	Quercus palustris	70.0
9	39	Quercus palustris	70.0
10	37	Quercus shumardii	70.0
11	36	Quercus palustris	70.0
12	38	Quercus palustris	70.0
13	39	Quercus palustris	70.0
14	36	Quercus palustris	70.0
15	39	Quercus palustris	70.0
16	40	Quercus rubra	90.0
17	1	Quercus coccinea	7.5

Oak Age



Fall Color

```
1 fall_color1_query = ""
2 | SELECT species_common_name
3 | FROM `bigquery-public-data.new_york_trees.tree_species`
4 | WHERE fall_color = 'Maroon' OR fall_color = 'Red';
5 ""
```

	species_common_name
0	Amur Maple
1	Northern Red Oak
2	Pin Oak
3	Purpleleaf Plum
4	Hawthorn
5	Red Maple
6	Callery Pear
7	Fastigiata Oak
8	Schubert Cherry
9	Black Gum



```
1 fall_color_query = ""
2 | SELECT DISTINCT fall_color
3 | FROM `bigquery-public-data.new_york_trees.tree_species`;
4 ""
```

	fall_color
0	Yellow/Orange
1	Red/Bronze
2	Yellow/Bronze
3	Red/Orange
4	Red
5	Yellow
6	Red/ Yellow
7	Red/ yellow
8	Yellow/ Bronze
9	Maroon
10	Cream
11	Purple/Maroon
12	Orange/Brown

Latitude Comparison

```
latitude1 = 40.912918
latitude2 = 40.498466

# Create a map centered around the latitudes
m = folium.Map(location=[latitude1, 0], zoom_start=10)

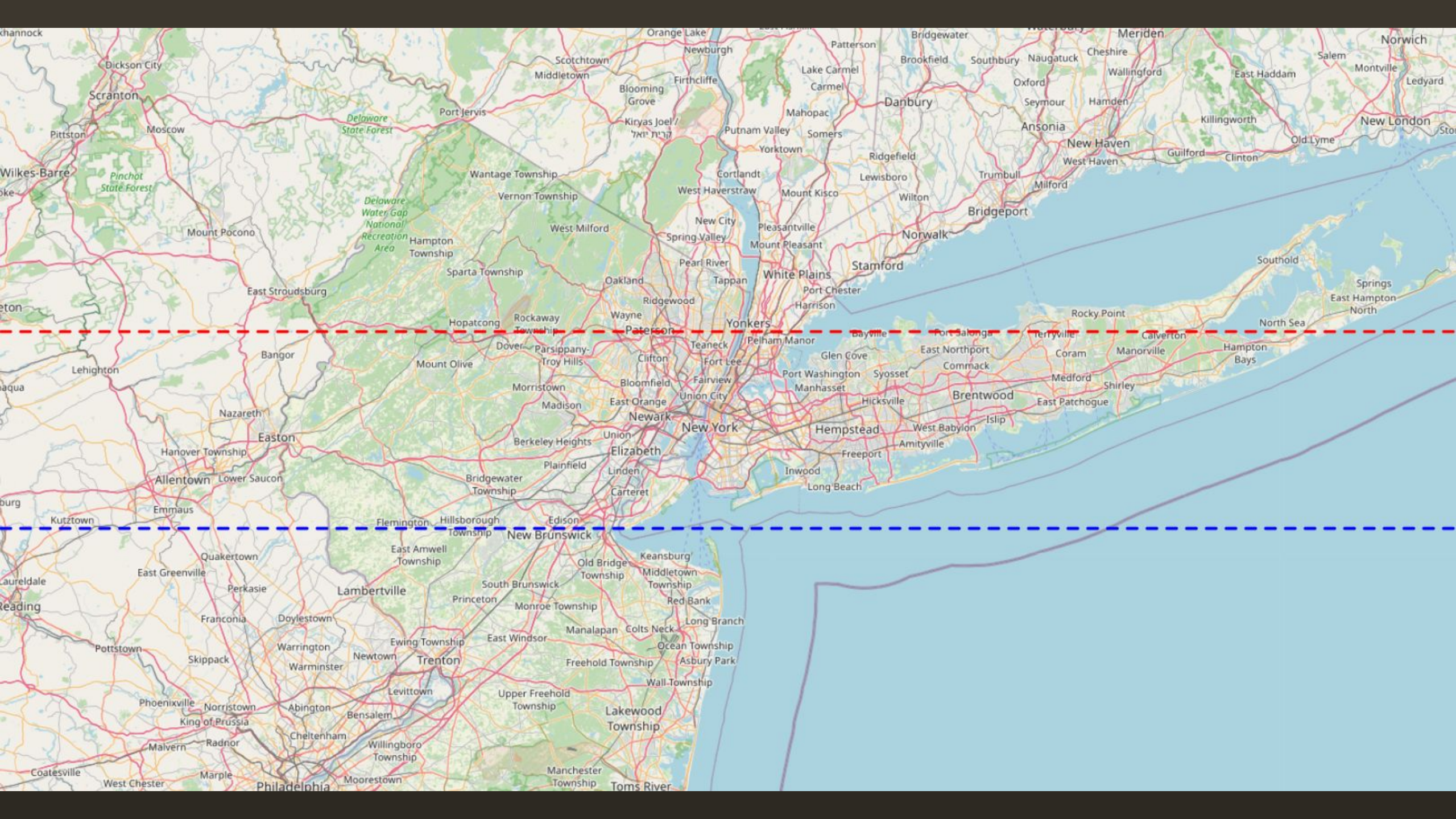
# Create a dashed red line
line1 = folium.PolyLine(
    locations=[[latitude1, -180], [latitude1, 180]],
    dash_array='10,10',
    color='red'
).add_to(m)

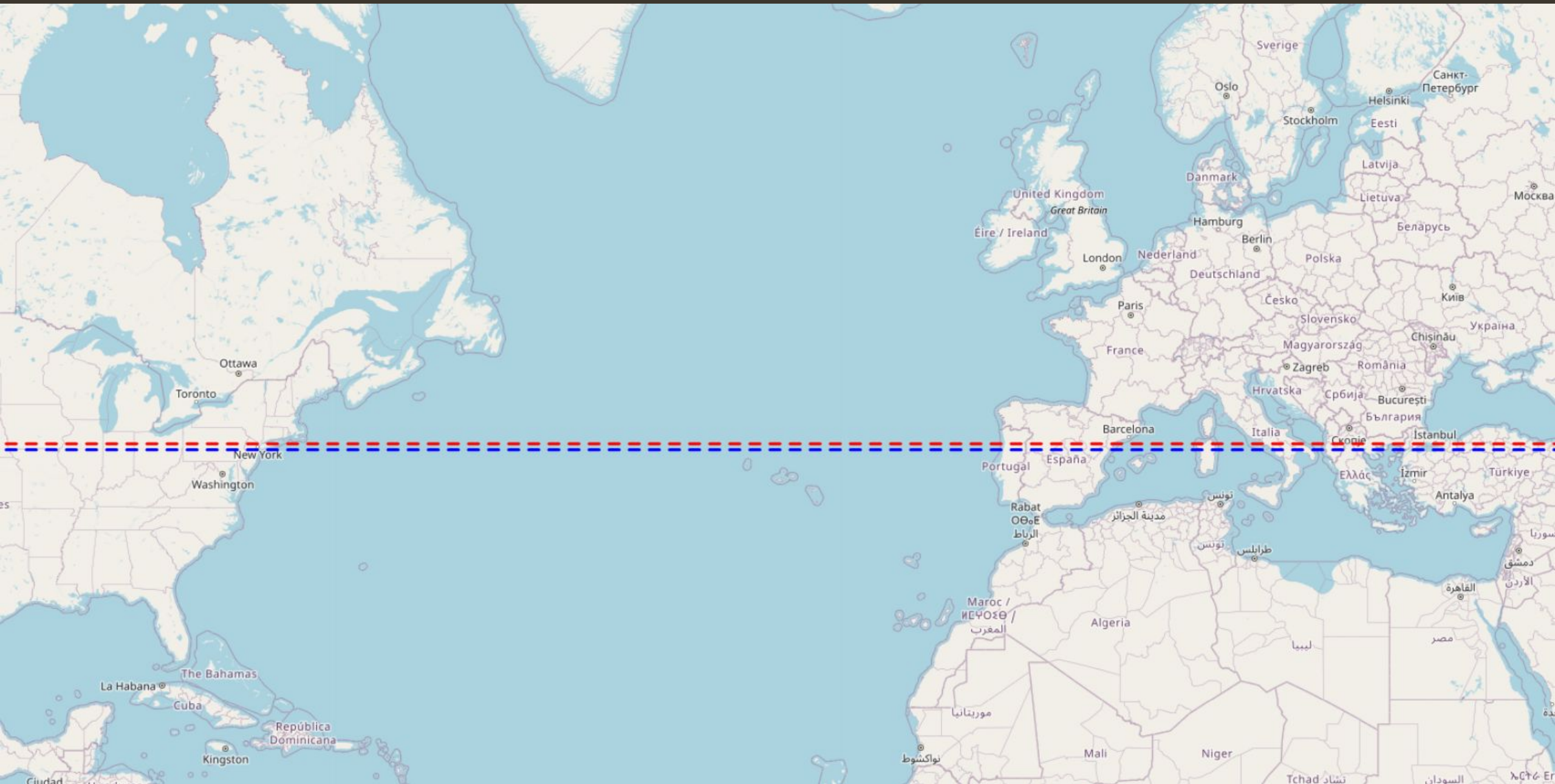
# Create a dashed blue line
line2 = folium.PolyLine(
    locations=[[latitude2, -180], [latitude2, 180]],
    dash_array='10,10',
    color='blue'
).add_to(m)

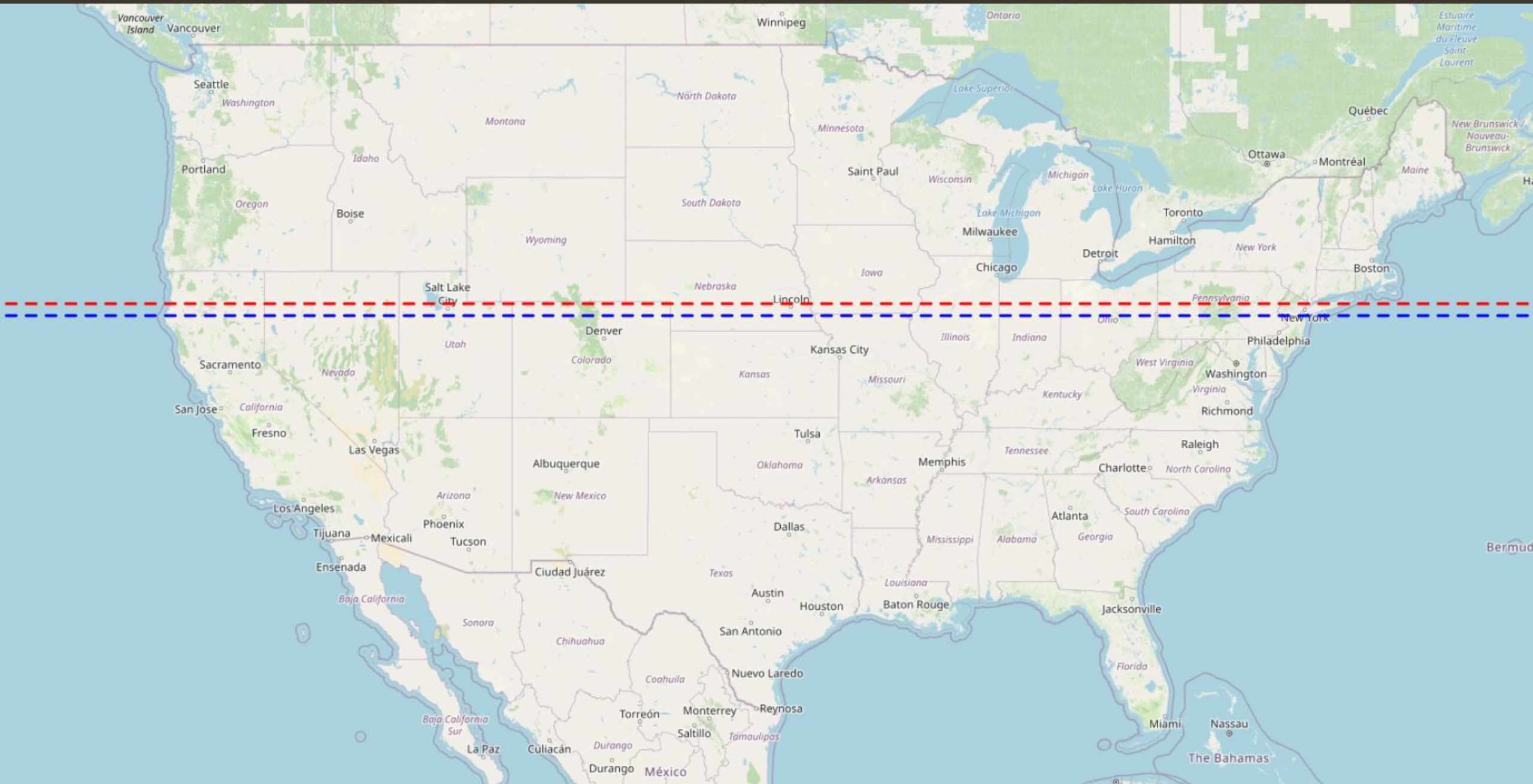
# Display the map
m
```

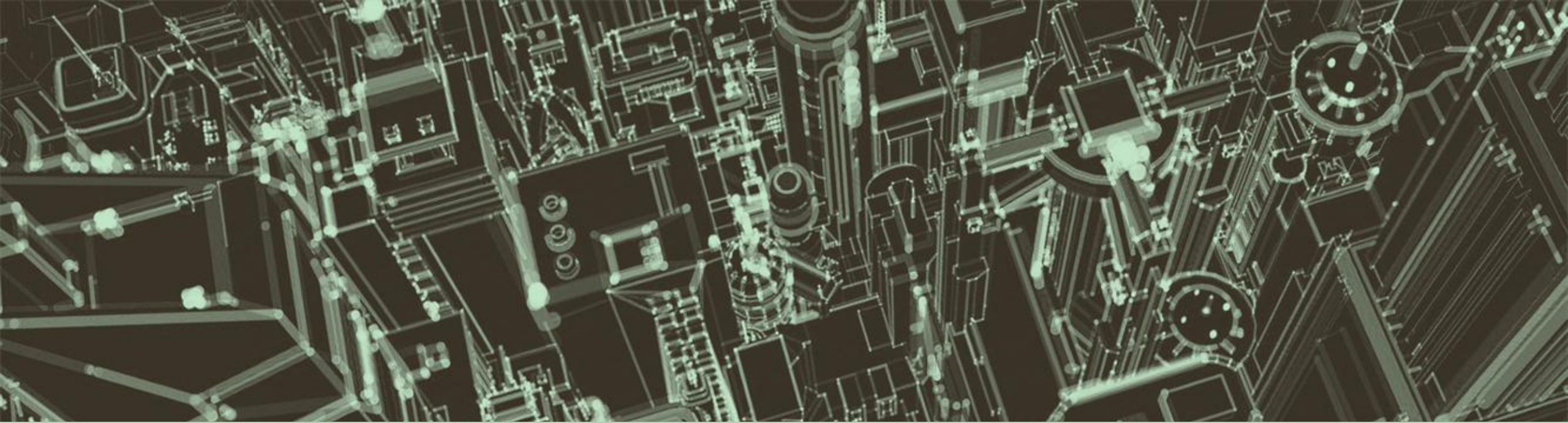
```
max_lat_query = """
SELECT MAX(latitude)
FROM `bigquery-public-data.new_york_trees.tree_census_2015`;
"""
```

```
min_lat_query = """
SELECT MIN(latitude)
FROM `bigquery-public-data.new_york_trees.tree_census_2015`;
"""
```







Thanks!

