Map Visualization: New York taxi pickups

Goal: biuld several visualization of New York taxi pickups map.

Step 1. Let's prepare the dataset of New York taxi moving over the year.

First of all we download the libraries.

In [28]:
```
import numpy as np
import pandas as pd
```

In [29]:
```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.style.use(['ggplot'])
```

Now we download the dataset.

In [30]:
```
data = pd.read_csv('data_taxi.csv')
data.head()
```

Out[30]:

	id	vendor_id	pickup_datetime	passenger_count	pickup_longitude	pickup_latitude	dropc
0	id3004672	1	2016-06-30 23:59:58	1	-73.988129	40.732029	
1	id3505355	1	2016-06-30 23:59:53	1	-73.964203	40.679993	
2	id1217141	1	2016-06-30 23:59:47	1	-73.997437	40.737583	
3	id2150126	2	2016-06-30 23:59:41	1	-73.956070	40.771900	
4	id1598245	1	2016-06-30 23:59:33	1	-73.970215	40.761475	

Let's check the data info.

In [31]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 625134 entries, 0 to 625133
Data columns (total 9 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   id                  625134 non-null  object 
 1   vendor_id           625134 non-null  int64  
 2   pickup_datetime     625134 non-null  object 
 3   passenger_count     625134 non-null  int64  
 4   pickup_longitude    625134 non-null  float64
 5   pickup_latitude     625134 non-null  float64
 6   dropoff_longitude   625134 non-null  float64
 7   dropoff_latitude    625134 non-null  float64
 8   store_and_fwd_flag  625134 non-null  object 
```

```
dtypes: float64(4), int64(2), object(3)
memory usage: 42.9+ MB
```

We need pickup info: longitudes and latitude.

Also we will require pickup_datetime info.

All these features are in the necessary dtype: **float64** and **object** (the last for datetime).

Datetime is for the markers info so we will not convert it in any other type.

In [32]:
```
data.shape
```

Out[32]:
```
(625134, 9)
```

As we see, we totally have **625134** observations.

Step 2. Generating basic toner and terrain maps.

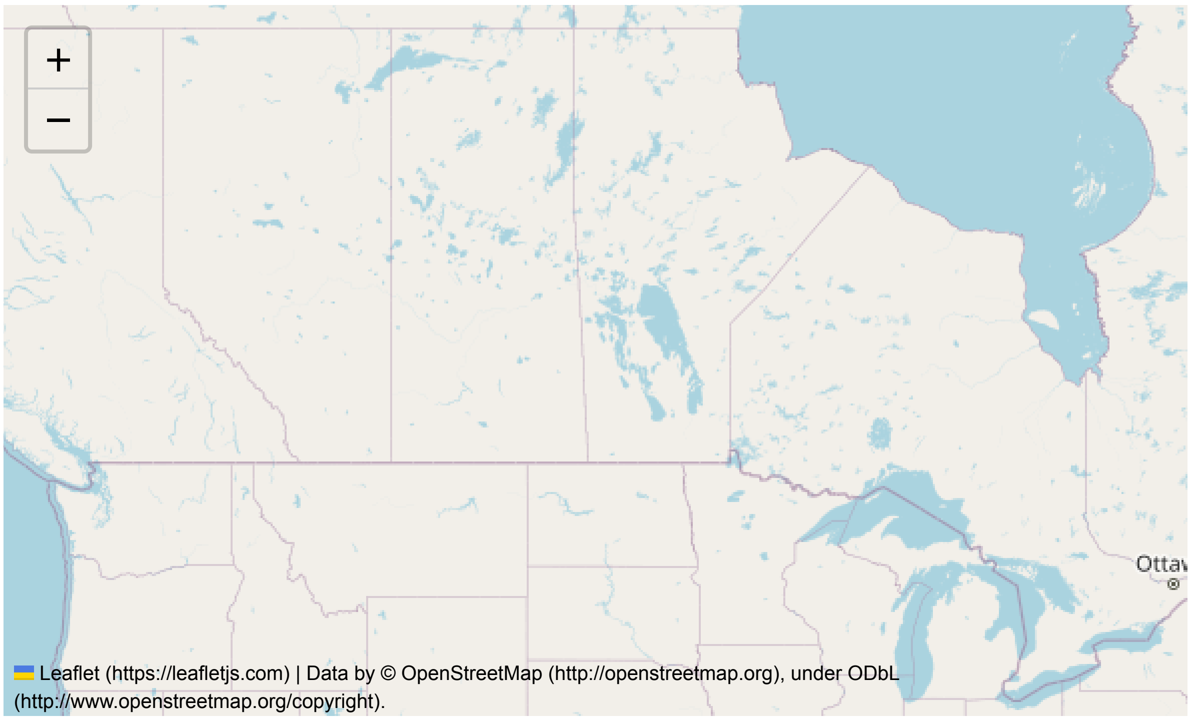
Let's import **folium** library.

In [33]:
```
import folium
```

Now we build a map of New York location based on the correspond latitude and longitude.

In [34]:
```
ny_map_base = folium.Map(location=[40.730610, -73.935242], zoom_start=4)
ny_map_base
```

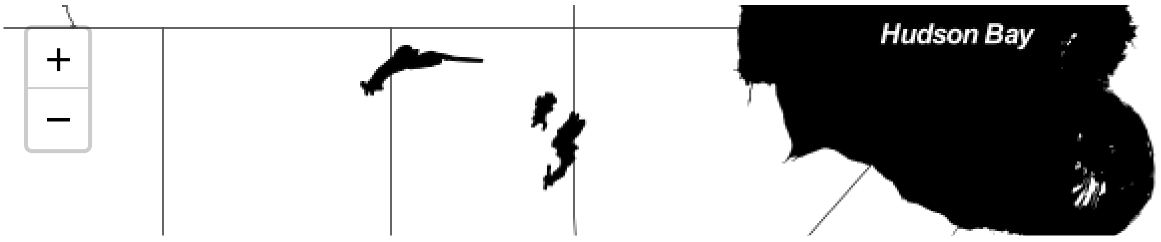
Out[34]:

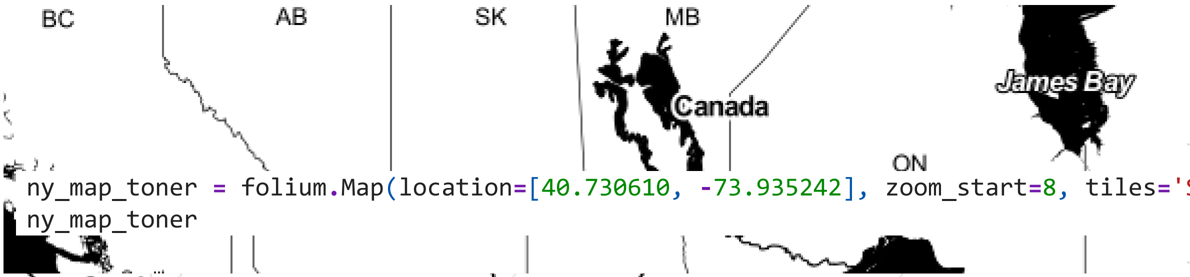

+
−
Ottaw
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

Let's check the toner varsion map, which is really useful for states, river and lake borders.

In [35]:
```
ny_map_toner = folium.Map(location=[40.730610, -73.935242], zoom_start=4, tiles='Sta
ny_map_toner
```

Out[35]:

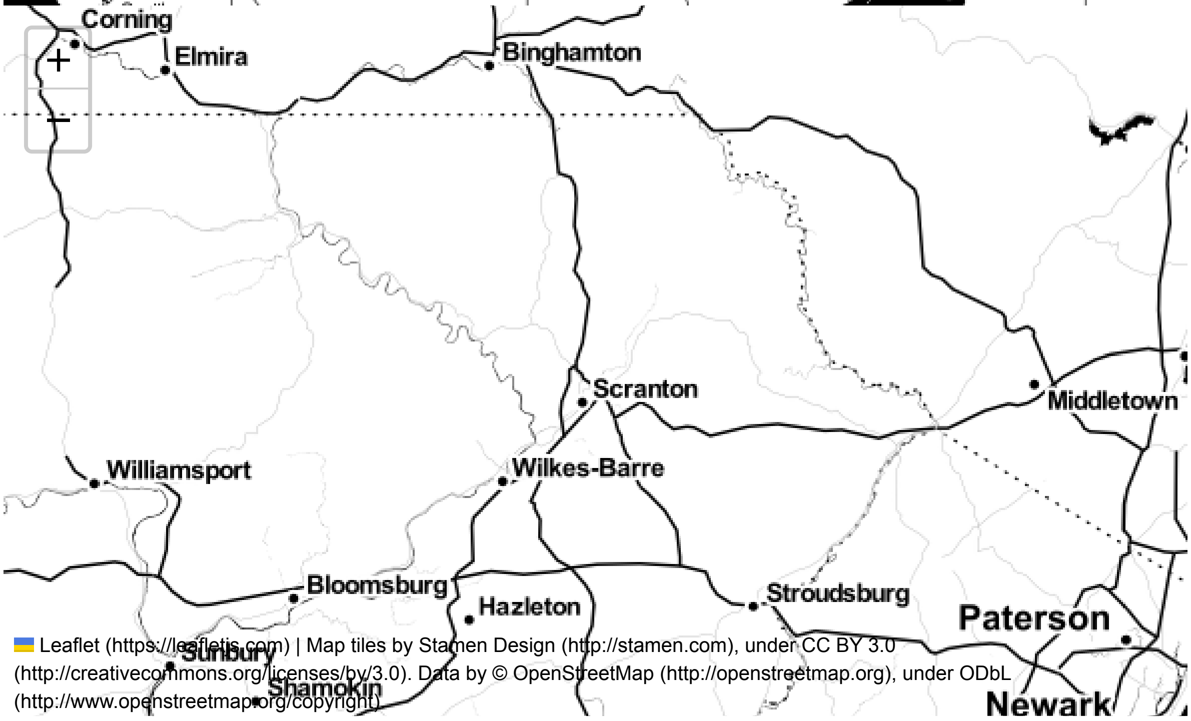




In [36]:

```
ny_map_toner = folium.Map(location=[40.730610, -73.935242], zoom_start=8, tiles='Sta
ny_map_toner
```

Out[36]:

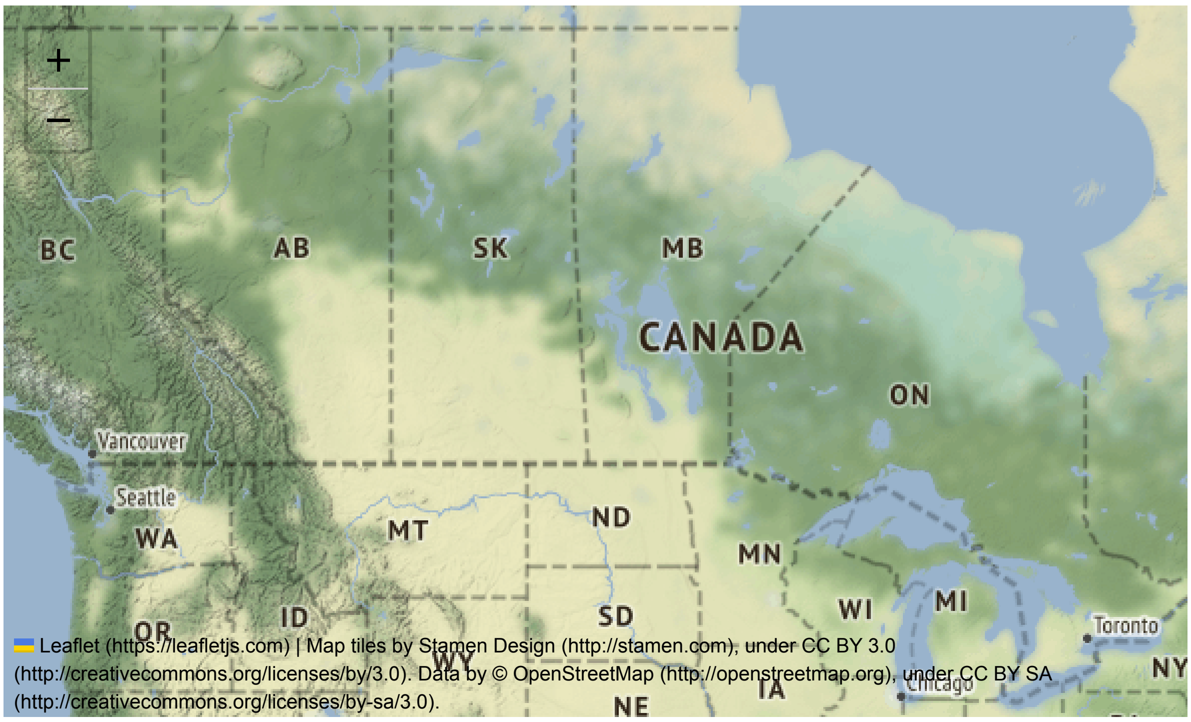

+
−
Corning
Elmira
Binghamton
Scranton
Middletown
Williamsport
Wilkes-Barre
Bloomsburg
Hazleton
Stroudsburg
Paterson
Sunbury
Shamokin
Newark
Leaflet (https://leafletjs.com) | Map tiles by Stamen Design (http://stamen.com), under CC BY 3.0 (http://creativecommons.org/licenses/by/3.0). Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright)

We also can check the terrain map, with different terrain features and locations.

In [37]:

```
ny_map_terrain = folium.Map(location=[40.730610, -73.935242], zoom_start=4, tiles='S
ny_map_terrain
```

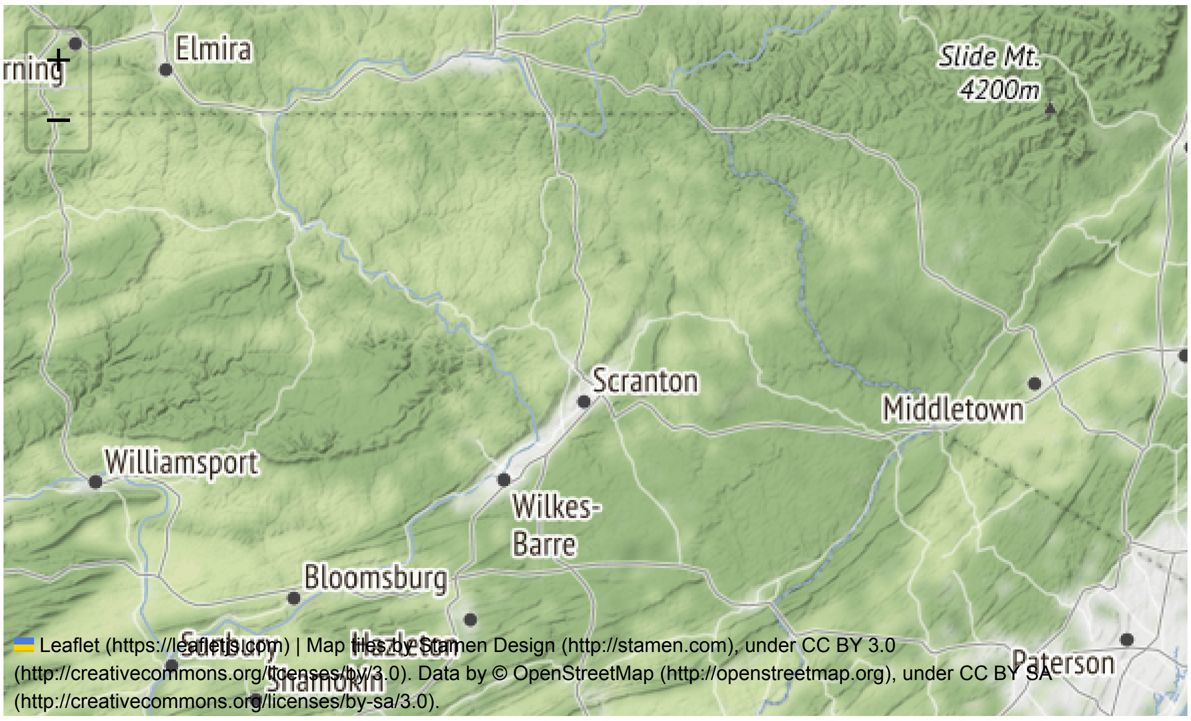
Out[37]:


+
−
BC
AB
SK
MB
CANADA
ON
Vancouver
Seattle
WA
MT
ND
MN
OR
ID
SD
WI
MI
Toronto
WY
IA
Chicago
NY
NE
Leaflet (https://leafletjs.com) | Map tiles by Stamen Design (http://stamen.com), under CC BY 3.0 (http://creativecommons.org/licenses/by/3.0). Data by © OpenStreetMap (http://openstreetmap.org), under CC BY SA (http://creativecommons.org/licenses/by-sa/3.0).

The zoomed version is also really useful.

In [38]:
```
ny_map_terrain = folium.Map(location=[40.730610, -73.935242], zoom_start=8, tiles='S
ny_map_terrain
```

Out[38]:



🟨 Leaflet (https://leafletjs.com) | Map tiles by Stamen Design (http://stamen.com), under CC BY 3.0 (http://creativecommons.org/licenses/by/3.0). Data by © OpenStreetMap (http://openstreetmap.org), under CC BY SA (http://creativecommons.org/licenses/by-sa/3.0).

Step 3. Building New York map with the first 100 pickup location from the dataset.

Let's get the work dataset. We get the first 100 items from the taxi records.

In [39]:
```
data_work = data.iloc[:100,:]
data_work.head()
```

Out[39]:

	id	vendor_id	pickup_datetime	passenger_count	pickup_longitude	pickup_latitude	dropc
0	id3004672	1	2016-06-30 23:59:58	1	-73.988129	40.732029	
1	id3505355	1	2016-06-30 23:59:53	1	-73.964203	40.679993	
2	id1217141	1	2016-06-30 23:59:47	1	-73.997437	40.737583	
3	id2150126	2	2016-06-30 23:59:41	1	-73.956070	40.771900	
4	id1598245	1	2016-06-30 23:59:33	1	-73.970215	40.761475	

We can check the result. Everything is correct: we got 100 rows with 9 features.

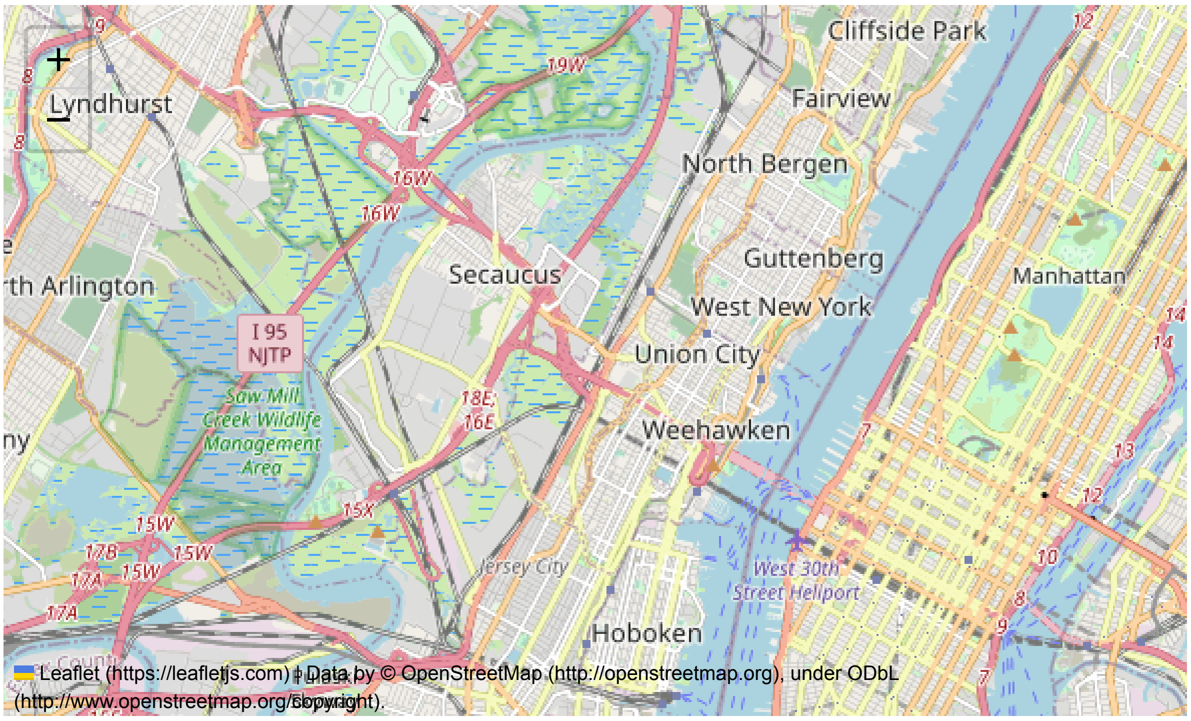
In [40]:
```
data_work.shape
```

Out[40]:
```
(100, 9)
```

Now we can use latitude and longitude coordinates of New York for the correspond map. As we will work with pickup data (street level details), we will use zoom value set for 12.

In [41]:
```
latitude = 40.730610
longitude = -73.935242
```

In [42]:
```
ny_map_work = folium.Map(location=[latitude, longitude], zoom_start=12)
ny_map_work
```

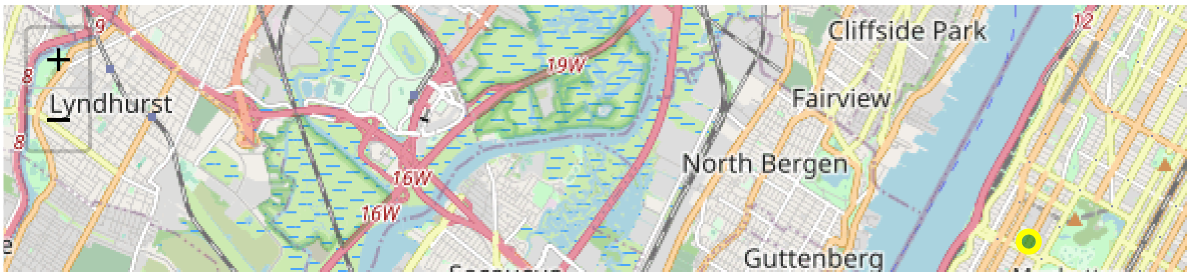
Out[42]:

Cliffside Park
Fairview
North Bergen
Lyndhurst
Guttenberg
Manhattan
Secaucus
West New York
Union City
Weehawken
Hoboken
Jersey City
I 95 NJTP
Saw Mill Creek Wildlife Management Area
West 30th Street Heliport
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

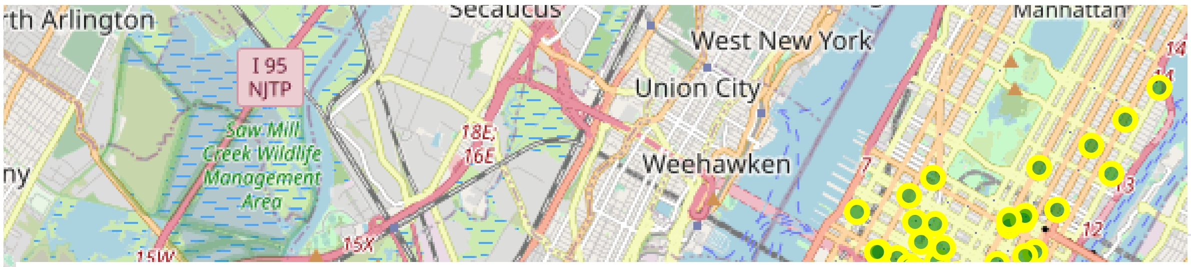
Now we will set all the necessary markers on the map.
We will use yellow border color and green fill color.

In [43]:
```
pickup_data = folium.map.FeatureGroup()
pickup_data
```

Out[43]:
```
<folium.map.FeatureGroup at 0xb3af050c10>
```

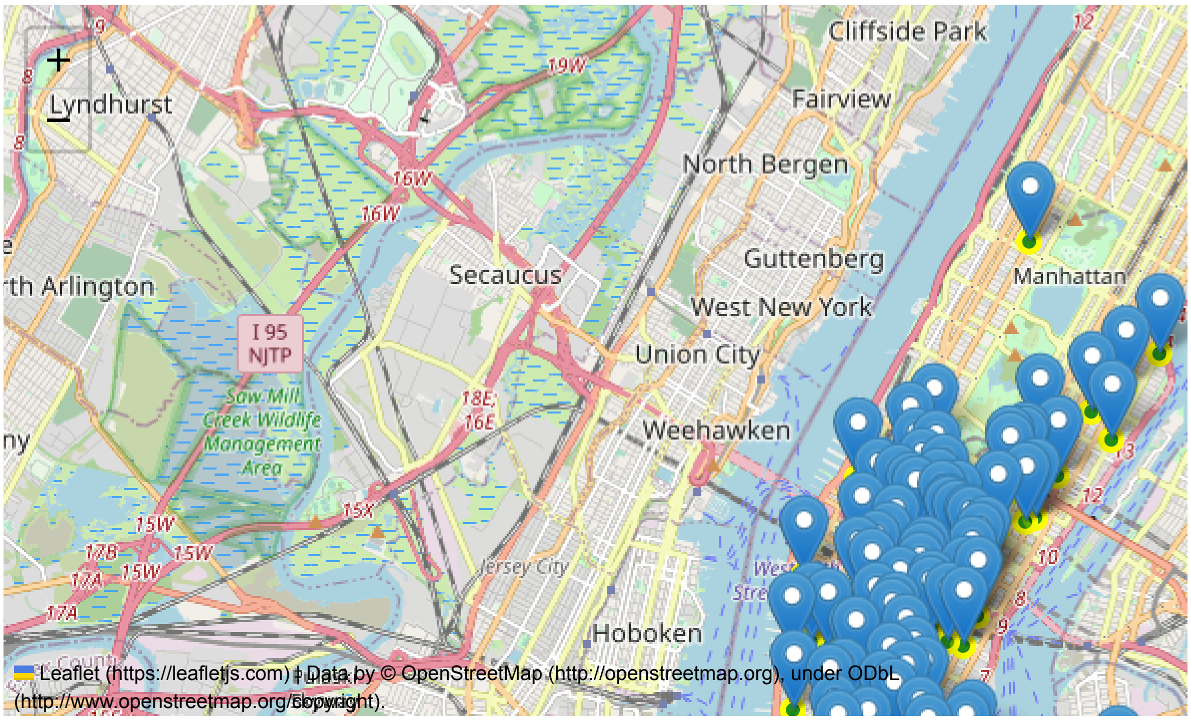
In [44]:
```
for lat, lon in zip(data_work.pickup_latitude, data_work.pickup_longitude):
    pickup_data.add_child(
        folium.features.CircleMarker(
            [lat, lon],
            radius=5,
            color='yellow',
            fill=True,
            fill_color='green',
            fill_opacity=0.6
        )
    )

ny_map_work.add_child(pickup_data)
```

Out[44]:




In [45]:
```python
pickup_data = folium.map.FeatureGroup()

for lat, lon in zip(data_work.pickup_latitude, data_work.pickup_longitude):
    pickup_data.add_child(
        folium.features.CircleMarker(
            [lat, lon],
            radius=5,
            color='yellow',
            fill=True,
            fill_color='green',
            fill_opacity=0.6
        )
    )

latitudes = list(data_work.pickup_latitude)
longitudes = list(data_work.pickup_longitude)
labels = list(data_work.pickup_datetime)

for lat, lng, label in zip(latitudes, longitudes, labels):
    folium.Marker([lat, lng], popup=label).add_to(ny_map_work)

ny_map_work.add_child(pickup_data)
```

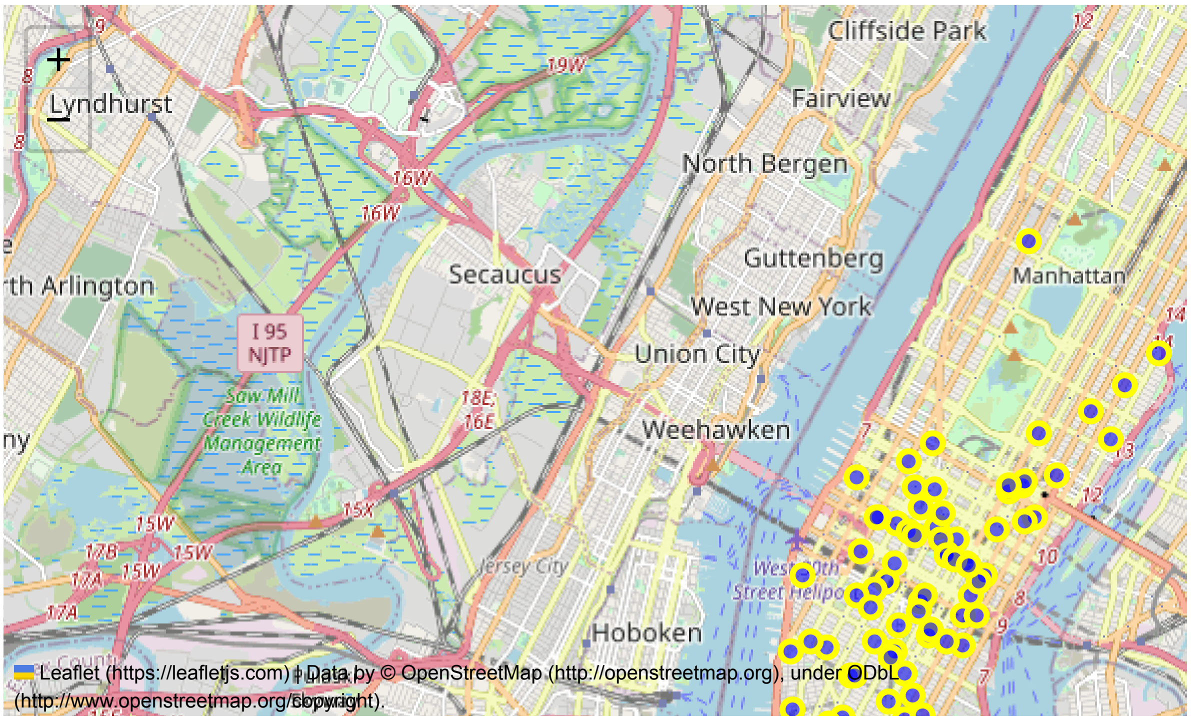
Out[45]:


Lyndhurst
Cliffside Park
Fairview
North Bergen
Guttenberg
Secaucus
Manhattan
West New York
Union City
Weehawken
Hoboken
Jersey City
rth Arlington
I 95 NJTP
Saw Mill Creek Wildlife Management Area
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

If markers are not necessary, we can use the same data dots that shows popup messages after clicking on them.
For popup info we used the same feature: pickup_datetime.

In [46]:
```python
ny_map_work = folium.Map(location=[latitude, longitude], zoom_start=12)

for lat, lng, label in zip(data_work.pickup_latitude, data_work.pickup_longitude, da
    folium.features.CircleMarker(
        [lat, lng],
        radius=5, # define how big you want the circle markers to be
        color='yellow',
        fill=True,
        popup=label,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(ny_map_work)

ny_map_work
```

Out[46]:

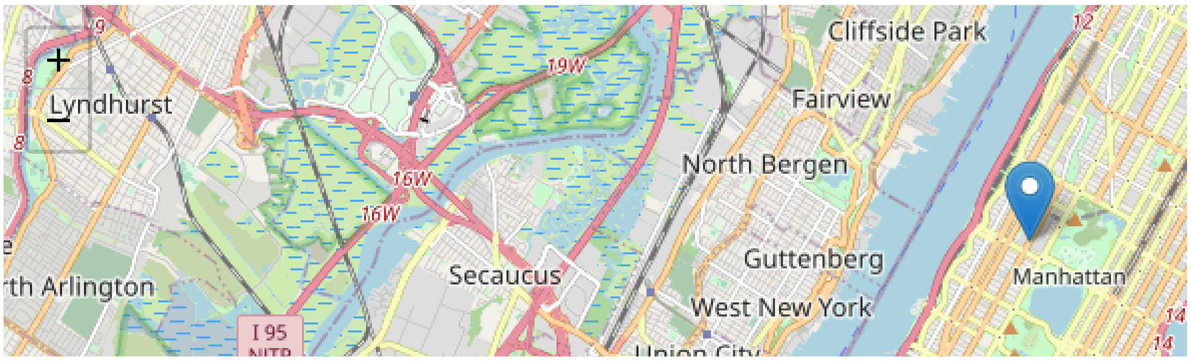

Lyndhurst
Cliffside Park
Fairview
North Bergen
Secaucus
Guttenberg
Manhattan
West New York
Union City
Weehawken
Hoboken
Jersey City
Saw Mill Creek Wildlife Management Area
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

We also can use the clusters. We have many different pickup dots, so it can be difficult to check them not on a big scales. The clusters will combine different data according to the scale in different areas.

In [47]:

```
from folium import plugins

ny_map_work = folium.Map(location = [latitude, longitude], zoom_start = 12)

pickup_data = plugins.MarkerCluster().add_to(ny_map_work)

for lat, lng, label, in zip(data_work.pickup_latitude, data_work.pickup_longitude, d
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(pickup_data)

ny_map_work
```

Out[47]:



Saw Mill Creek Wildlife Management Area
Weehawken
Jersey City
Hoboken
West 30th Street Heliport
6
13
66
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).