```
! pwd
```

```
/content/sample_data
```

```
cd sample_data/
```

```
[Errno 2] No such file or directory: 'sample_data/'
/content/sample_data
```

Importando as bibliotecas/Importing libraries

```
import pandas as pd
import numpy as np
from folium import plugins
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from sklearn.cluster import KMeans
warnings.filterwarnings('ignore')
```

Carregando o dataset/Loading the dataset

```
df=pd.read_csv('AB_NYC_2019.csv')
```

Visualizando as primeiras linhas/Viewing the first lines

```
df.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
		THE VILLAGE OF							Private	

Numero de linhas e colunas/Number of rows and columns

```
df.shape
```

```
(48895, 16)
```

Visualizando o tipo de dados/Viewing the data type

```
df.dtypes
```

```
id                                int64
name                             object
host_id                           int64
host_name                        object
neighbourhood_group              object
neighbourhood                    object
latitude                        float64
longitude                       float64
room_type                        object
price                             int64
minimum_nights                    int64
number_of_reviews                 int64
last_review                      object
reviews_per_month               float64
calculated_host_listings_count    int64
availability_365                  int64
dtype: object
```

Numero de valores nulos/Number of null values

```
df.isnull().sum()
```

```
id                                  0
name                               16
host_id                             0
host_name                          21
neighbourhood_group                 0
neighbourhood                       0
latitude                            0
longitude                           0
room_type                           0
price                               0
minimum_nights                      0
number_of_reviews                   0
last_review                     10052
reviews_per_month               10052
calculated_host_listings_count      0
availability_365                    0
dtype: int64
```

Remover colunas desnecessárias para análise /Remove unnecessary columns for analysis

```
df.drop(['name','host_name','last_review','reviews_per_month'], axis=1, inplace=True)
```

```
df.isnull().sum()
```

```
id                              0
host_id                         0
neighbourhood_group             0
neighbourhood                   0
```

Dados estatísticos / Statistic data

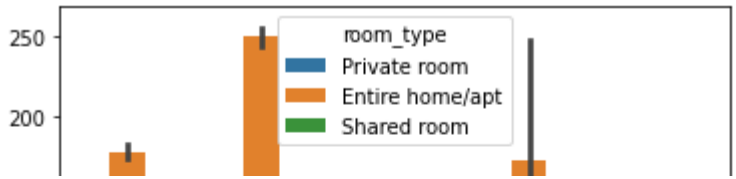
```
room_type                       0
```

```
df.describe()
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	calculate
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	

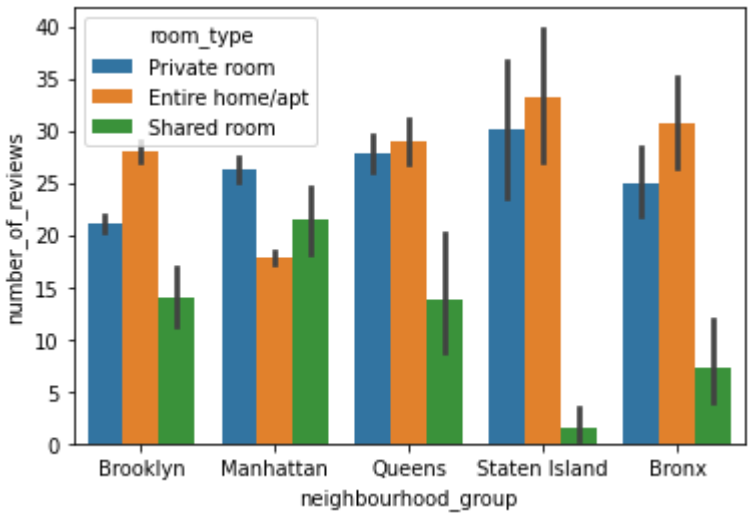
Relação do preço com a localização do imóvel / Price relation with property location

```
sns.barplot(x='neighbourhood_group',y='price', hue='room_type',data=df);
```



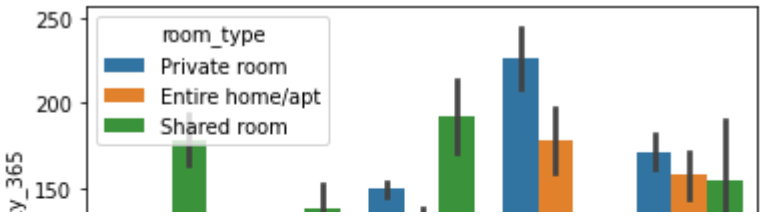
Número de avaliações por bairro / Number of reviews by neighborhood

```
sns.barplot(x='neighbourhood_group',y='number_of_reviews', hue='room_type',data=df);
```



Disponibilidade de dias ao ano para reserva / Availability of days a year for reservation

```
sns.barplot(x='neighbourhood_group',y='availability_365', hue='room_type',data=df);
```



Instalando o folium/Installing folium

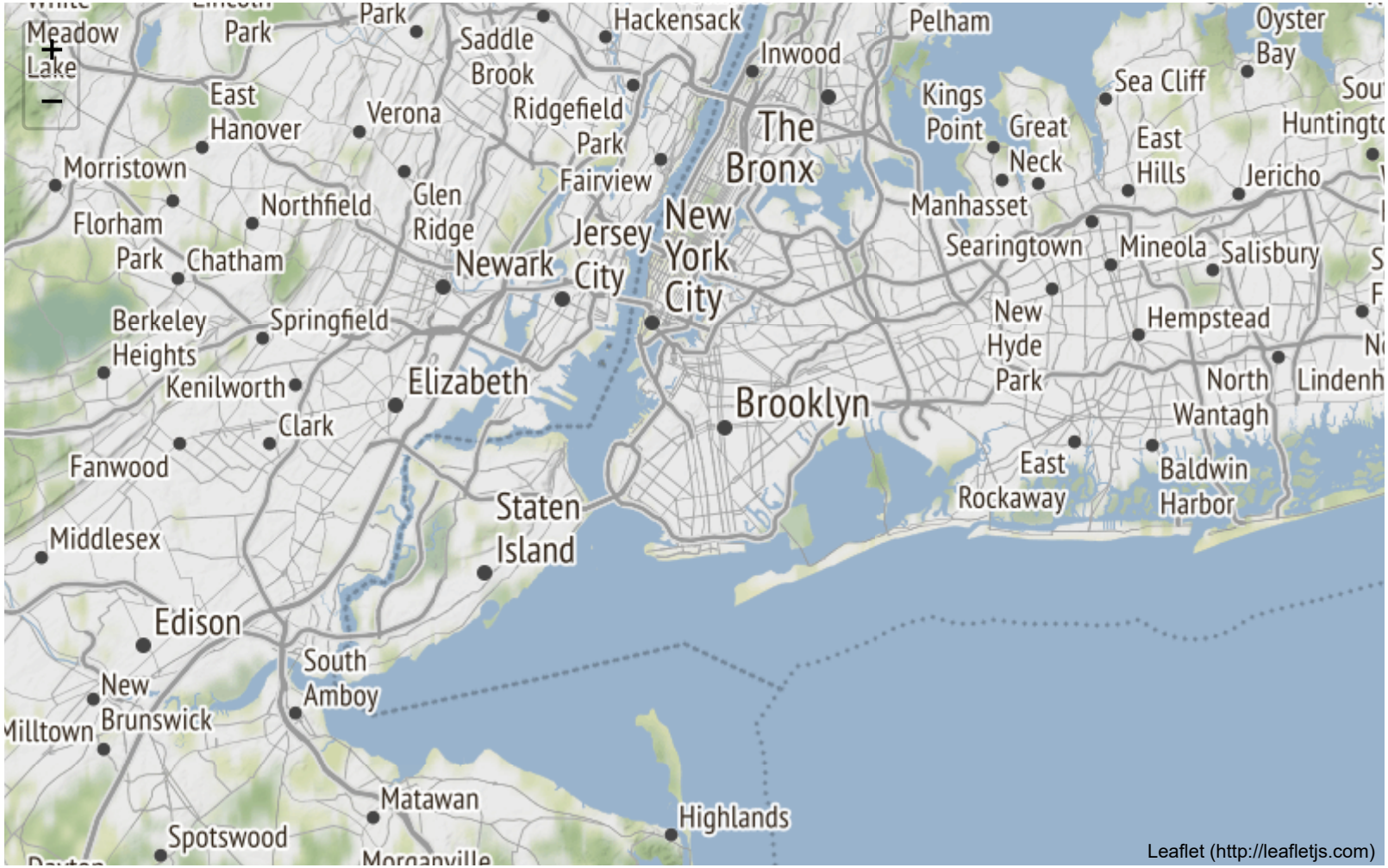
```
pip install folium
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.6/dist-packages (0.8.3)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.6/dist-packages (from folium) (0.4.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from folium) (1.12.0)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from folium) (2.23.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from folium) (1.18.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.6/dist-packages (from folium) (2.11.2)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests->folium) (20
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests->folium) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests->folium) (3.0
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages (from jinja2->folium) (1.1.1)
```

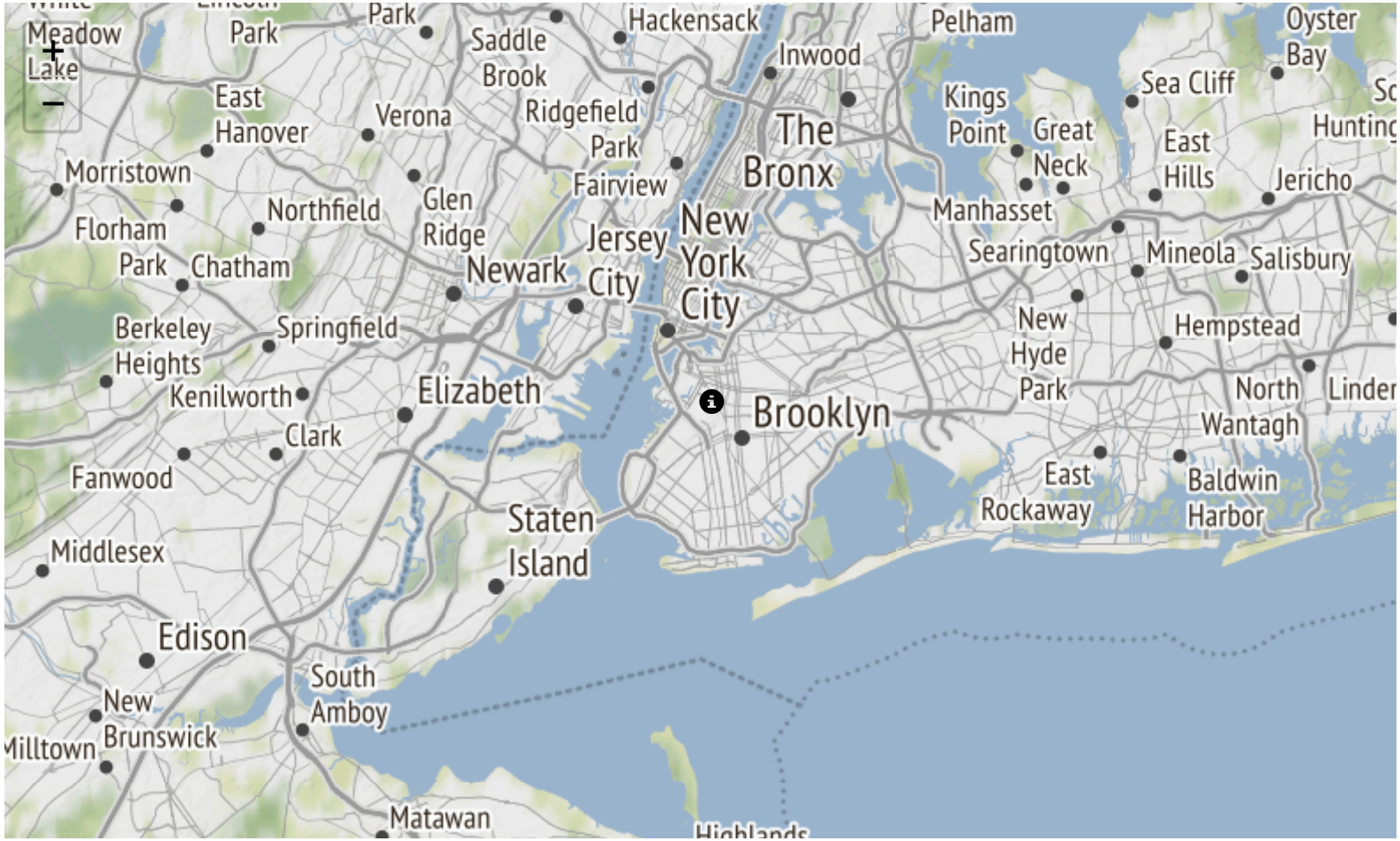
```
import folium
```

Localizando área no mapa através da latitude e longitude/Locating area on the map by latitude and longitude

```
mapa = folium.Map(location=[40.64749 ,  -73.97237], width=800,height=500,tiles='Stamen Terrain')
mapa
```


Meadow Lake
Park
Park
Saddle Brook
Hackensack
Pelham
Oyster Bay
Inwood
Sea Cliff
East Hanover
Verona
Ridgefield Park
Kings Point
Great Neck
East Hills
Huntingto
The Bronx
Morristown
Fairview
Jericho
Manhasset
Florham Park
Northfield
Glen Ridge
Jersey City
New York City
Searingtown
Mineola
Salisbury
Chatham
Newark
Berkeley Heights
Springfield
New Hyde Park
Hempstead
Kenilworth
Elizabeth
Brooklyn
North
Lindenh
Clark
Wantagh
Fanwood
East Rockaway
Baldwin Harbor
Staten Island
Middlesex
Edison
South Amboy
New Brunswick
Milltown
Matawan
Highlands
Spotswood
Leaflet (http://leafletjs.com)

```
folium.Marker([40.64749,  -73.97237], popup='<i>Brooklyn</i>', tooltip='click here!', icon=folium.Icon(color='red')).add_to(m
mapa.add_child(folium.LatLngPopup())
mapa
```

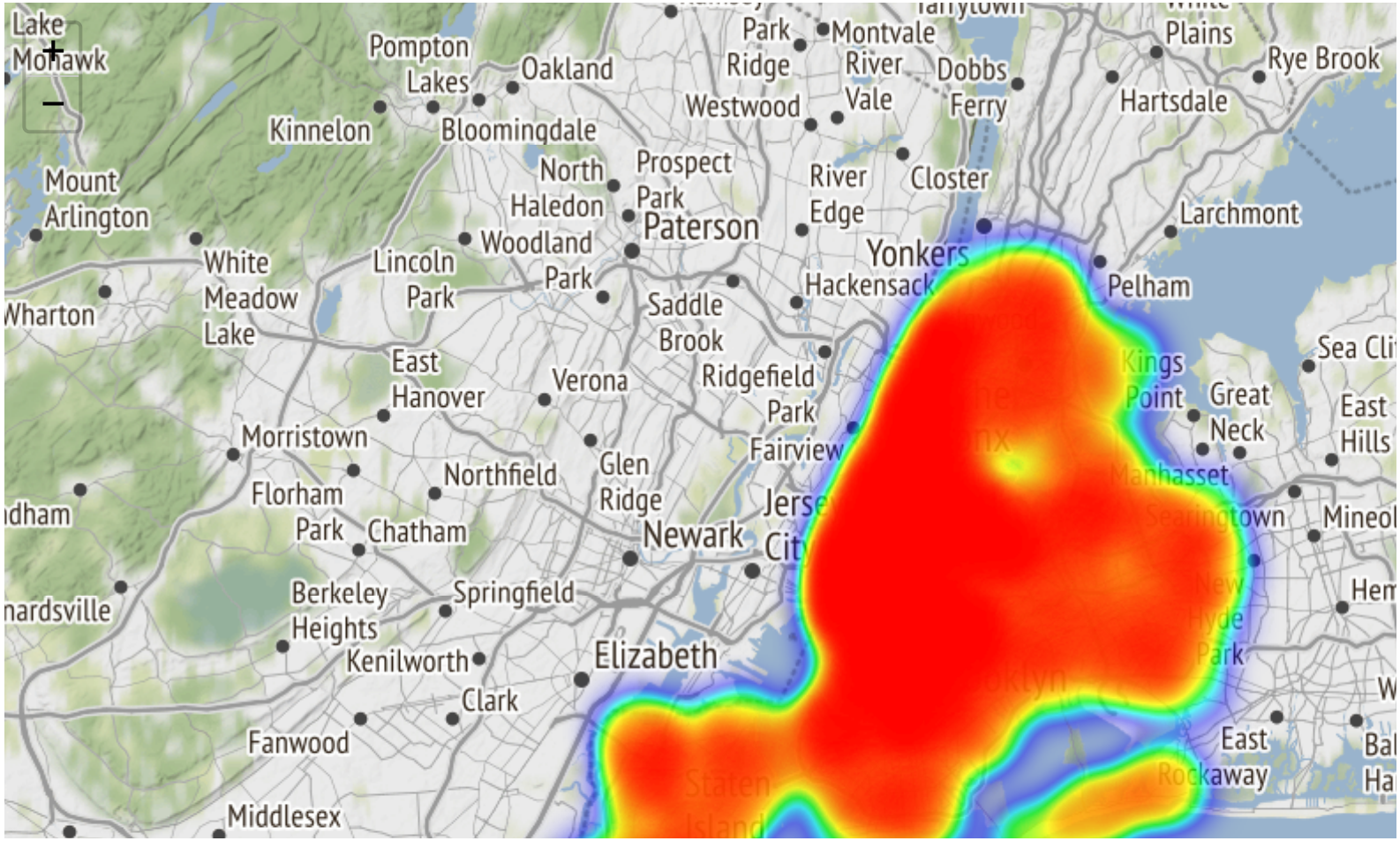

Meadow Lake
Park
Park
Saddle Brook
Hackensack
Pelham
Oyster Bay
Inwood
Sea Cliff
East Hanover
Verona
Ridgefield Park
Kings Point
Great Neck
East Hills
Hunting
The Bronx
Morristown
Fairview
Jericho
Northfield
Glen Ridge
Manhasset
Florham Park
Chatham
Jersey City
New York City
Searingtown
Mineola
Salisbury
Newark
Berkeley Heights
Springfield
New Hyde Park
Hempstead
Kenilworth
Elizabeth
Brooklyn
North
Linder
Clark
Wantagh
Fanwood
East Rockaway
Baldwin Harbor
Staten Island
Middlesex
Edison
South Amboy
New Brunswick
Milltown
Matawan
Highlands

Visualizando o mapa de calor/Viewing the heat map

```
coordenadas=[]
for lat, long in zip(df.latitude.values[:45000], df.longitude.values[:45000]):
  coordenadas.append([lat,long])
```

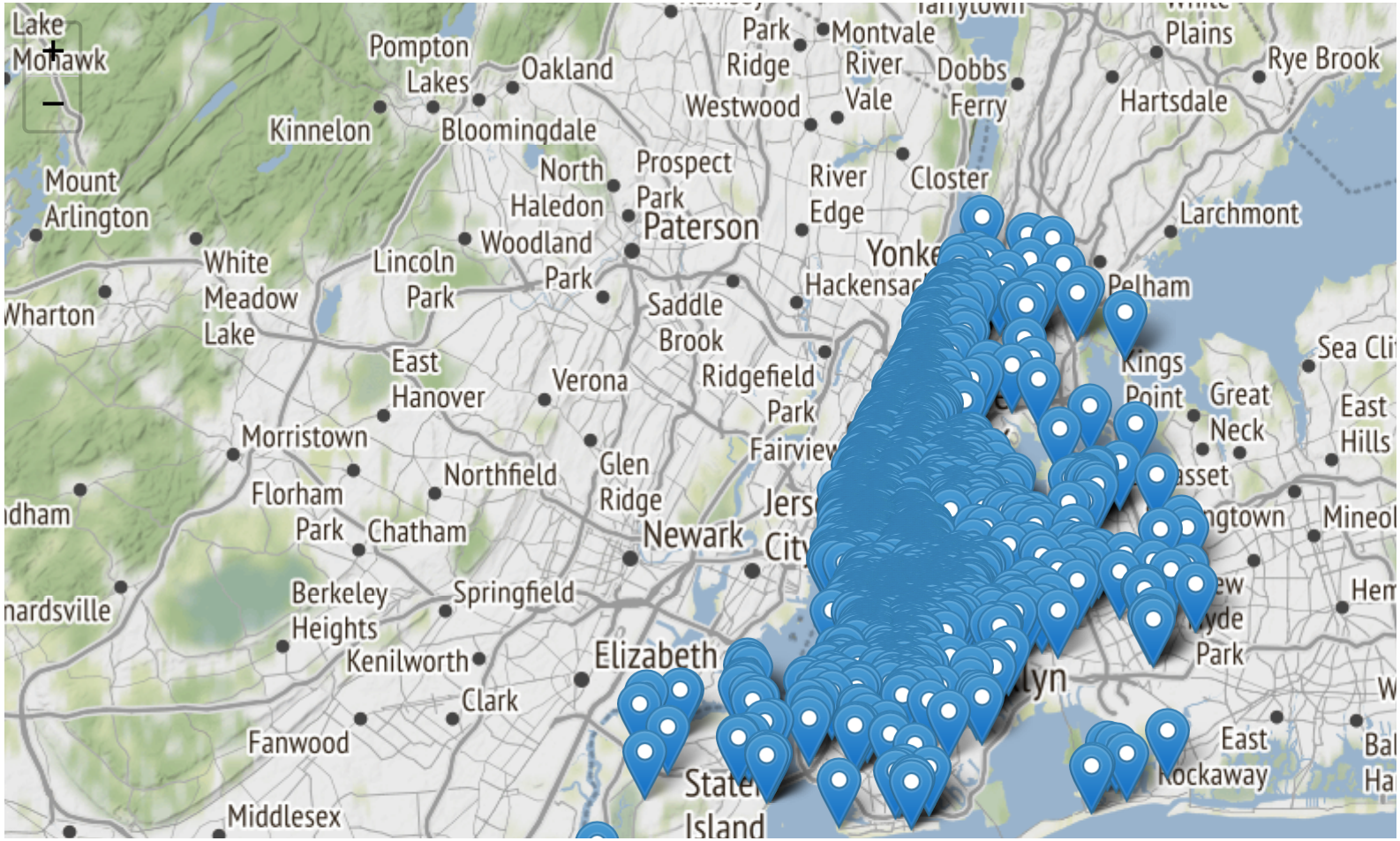
```
mapa = folium.Map(location=[40.64749, -73.97237], width=1000,height=800,tiles='Stamen Terrain')
```

```
mapa.add_child(plugins.HeatMap(coordenadas))
```


Lake
Mohawk
Pompton
Lakes
Oakland
Park
Ridge
Montvale
River
Dobbs
Ferry
Plains
Rye Brook
Hartsdale
Westwood
Vale
Kinnelon
Bloomingdale
North
Haledon
Prospect
Park
Paterson
River
Edge
Closter
Mount
Arlington
Woodland
Park
Yonkers
Larchmont
Hackensack
Pelham
White
Meadow
Lake
Lincoln
Park
Saddle
Brook
Wharton
East
Hanover
Verona
Ridgefield
Park
Kings
Point
Great
Neck
Sea Cli
East
Hills
Morristown
Fairview
Glen
Ridge
Northfield
Florham
Park
Chatham
Newark
Jerse
Cit
Manhasset
Searingtown
Mineol
Berkeley
Heights
Springfield
Elizabeth
Kenilworth
Clark
Fanwood
Middlesex
Hyde
Park
East
Rockaway
Staten
Island

Inserindo marcadores/Inserting bookmarks

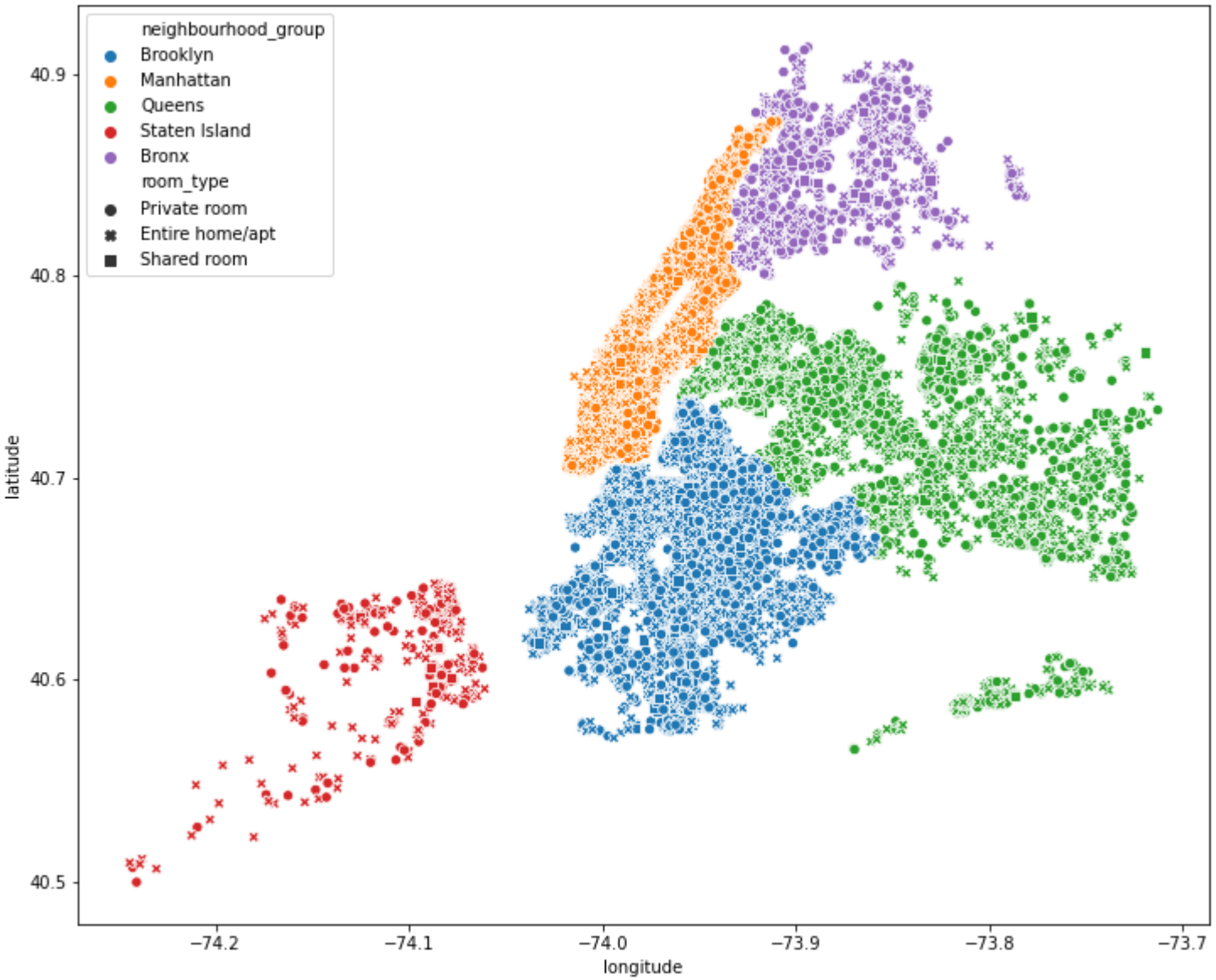
```
lat = df['latitude'][:4500].values
long = df['longitude'][:4500].values
mapa = folium.Map(location=[40.64749, -73.97237], width=1000,height=800,tiles='Stamen Terrain')

for la,lo in zip(lat,long):
    folium.Marker([la, lo]).add_to(mapa)
mapa
```




```
from IPython.display import Image
Image(filename='New York.jpg')
```

Plotando mapa de Nova Iorque definindo as areas de acordo com estilos de acomodação / Plotting New York map defining areas acc to accommodation styles

```
 plt.subplots(figsize = (12,10))
sns.scatterplot(x='longitude', y='latitude', hue='neighbourhood_group', s = 40, alpha = 'auto', style='room_type',data=df);
```


neighbourhood_group
Brooklyn
Manhattan
Queens
Staten Island
Bronx
room_type
Private room
Entire home/apt
Shared room
40.9
40.8
40.7
40.6
40.5
latitude
−74.2
−74.1
−74.0
−73.9
−73.8
−73.7
longitude