Open in Colab

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
!conda install -c anaconda pandas-profiling
#! pip install pandas-profiling
import pandas_profiling
import seaborn as sns
%matplotlib inline
```

```
/bin/bash: conda: command not found
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util
  import pandas.util.testing as tm
```

```
demographics = pd.read_csv('/content/drive/My Drive/Colab Notebooks/demographics.csv', sep=';',header
demographics.head()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-2-591660702917> in <module>()
----> 1 demographics = pd.read_csv('/content/drive/My Drive/Colab Notebooks/demographics.csv', sep=
      2 demographics.head()

                              4 frames
/usr/local/lib/python3.6/dist-packages/pandas/io/parsers.py in __init__(self, src, **kwds)
   1889         kwds["usecols"] = self.usecols
   1890
-> 1891         self._reader = parsers.TextReader(src, **kwds)
   1892         self.unnamed_cols = self._reader.unnamed_cols
   1893

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()

FileNotFoundError: [Errno 2] File /content/drive/My Drive/Colab Notebooks/demographics.csv does not
```

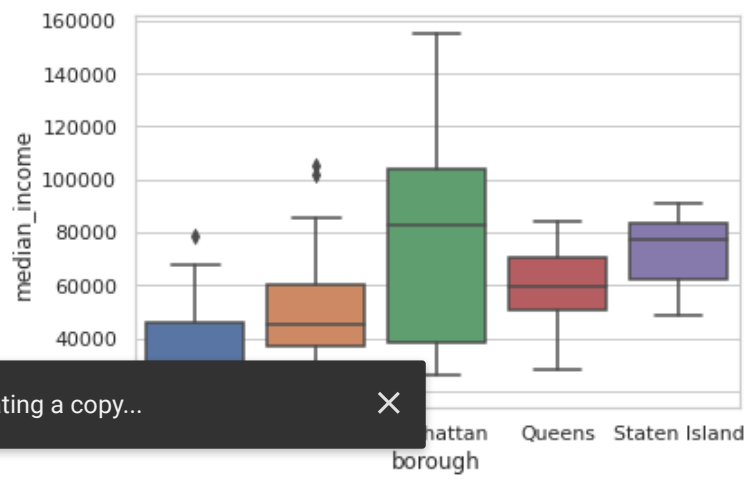
SEARCH STACK OVERFLOW

	population	under_5_years	5-9_years	10-14_years	15-19_years	20-24_years	25-29_ye
count	188.000000	188.00000	188.000000	188.000000	188.000000	188.000000	188.000
mean	43397.175532	2752.12766	2515.478723	2489.159574	2842.632979	3404.617021	3872.106
std	21288.062949	1695.80947	1496.808314	1459.440306	1574.371410	1886.482082	2462.683
min	13354.000000	506.00000	408.000000	326.000000	449.000000	798.000000	736.000
25%	27237.000000	1490.50000	1443.500000	1437.750000	1619.000000	1929.500000	2089.250
50%	37897.000000	2386.00000	2228.500000	2197.500000	2568.000000	2956.000000	3081.500
75%	54244.750000	3579.25000	3242.750000	3263.750000	3717.250000	4527.000000	5252.750
max	132378.000000	14703.00000	11971.000000	10024.000000	9094.000000	10046.000000	11971.000

```
from google.colab import drive
drive.mount('/content/drive')
```

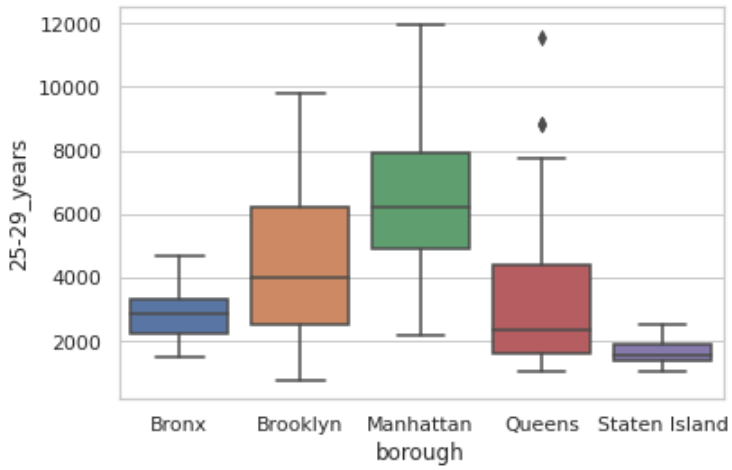
```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6c

Enter your authorization code:
··********
Mounted at /content/drive
```

```
import seaborn as sns
sns.set(style="whitegrid")
ax = sns.boxplot(x="borough", y="median_income", data=demographics)
```

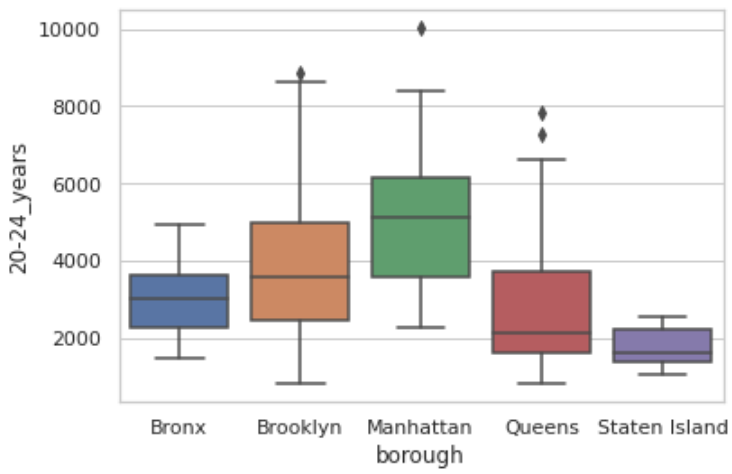


Creating a copy...

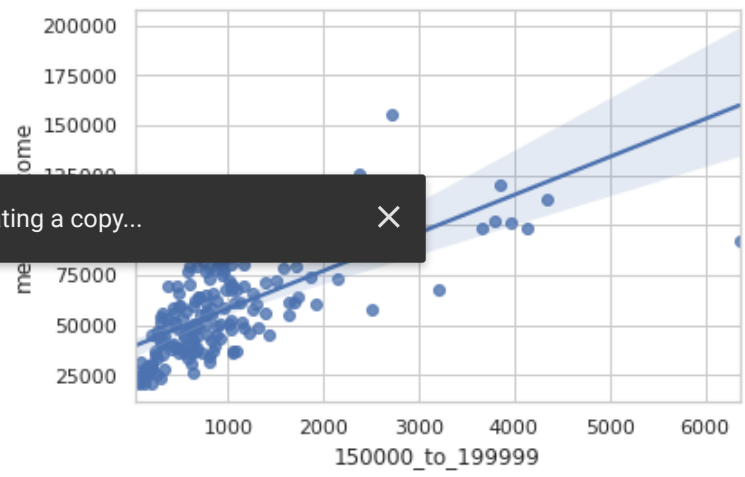
```
sns.set(style="whitegrid")
ax = sns.boxplot(x="borough", y="25-29_years", data=demographics)
```




```
sns.set(style="whitegrid")
ax = sns.boxplot(x="borough", y="20-24_years", data=demographics)
```




```
ax = sns.regplot(x="150000_to_199999", y="median_income", data=demographics)
```



Creating a copy...

```
import json # library to handle JSON files
with open('/content/drive/My Drive/Colab Notebooks/nyu_2451_34572-geojson.json') as json_data:
    newyork_data = json.load(json_data)

```

```
# Tranform the data into a pandas dataframe
# define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

neighborhoods_data = newyork_data['features']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
# neighborhoods

for data in neighborhoods_data:
    borough = neighborhood_name = data['properties']['borough']
    neighborhood_name = data['properties']['name']

    neighborhood_latlon = data['geometry']['coordinates']
    neighborhood_lat = neighborhood_latlon[1]
    neighborhood_lon = neighborhood_latlon[0]

    neighborhoods = neighborhoods.append({'Borough': borough,
                                          'Neighborhood': neighborhood_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore_index=True)
neighborhoods.head()
```

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

```
!wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
print('Data downloaded!')

with open('newyork_data.json') as json_data:
    newyork_data = json.load(json_data)

#pd.read_csv('https://geo.nyu.edu/catalog/nyu_2451_34572/')
```

Creating a copy...

```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(neighborhoods['Borough'].unique()),
        neighborhoods.shape[0]
    )
)
```

```
The dataframe has 5 boroughs and 306 neighborhoods.
```

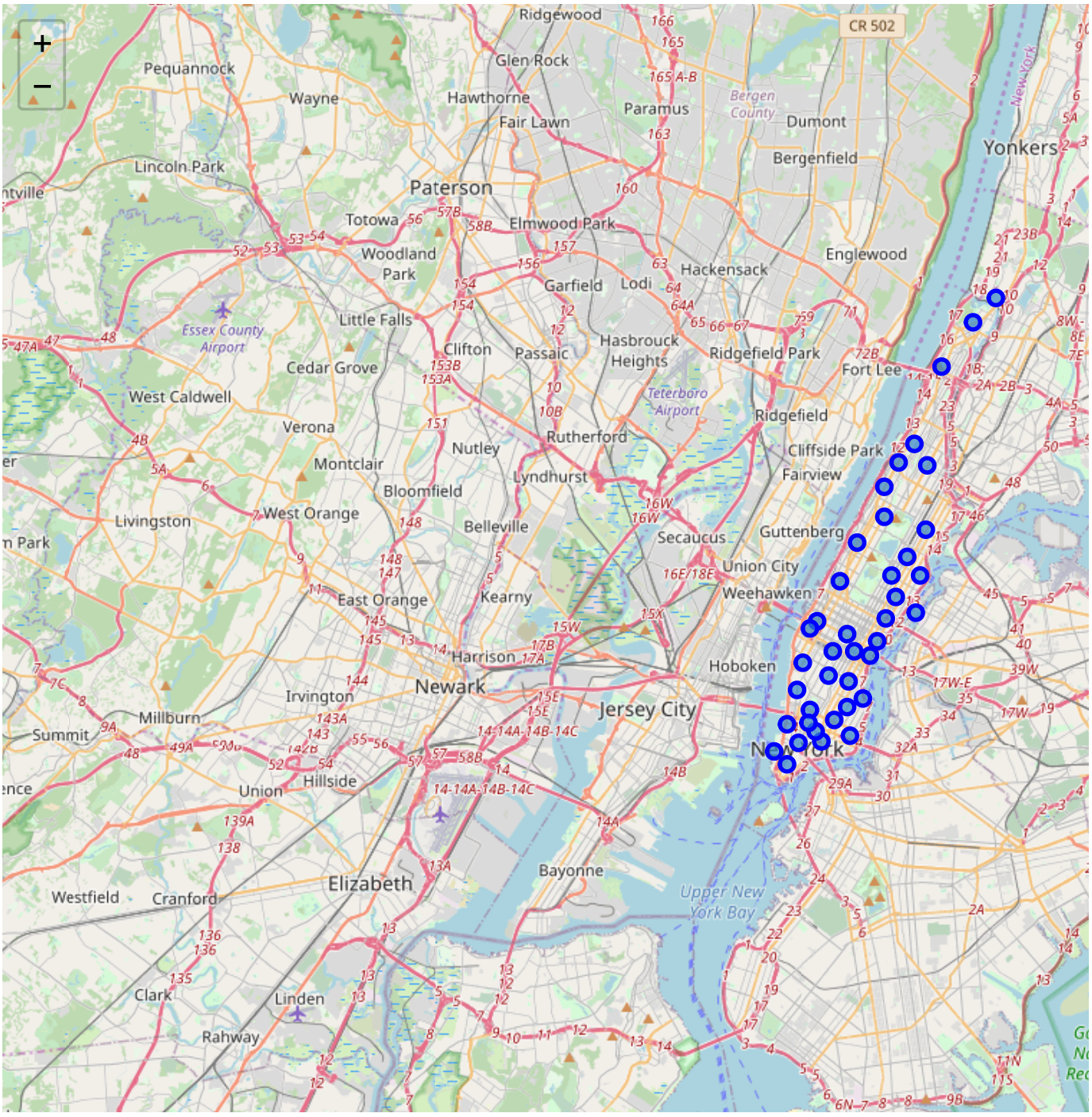
```
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
```

```
# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

import folium # map rendering library
```

```
manhattan_data = neighborhoods[neighborhoods['Borough'] == 'Manhattan'].reset_index(drop=True)
# manhattan_data.head()

address = 'Manhattan, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Manhattan are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Manhattan are 40.7896239, -73.9598939.
```

```
# create map of Manhattan using latitude and longitude values
map_manhattan = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(manhattan_data['Latitude'], manhattan_data['Longitude'], manhattan_data['N
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan)

map_manhattan
```



Creating a copy...

```
# Define Foursquare Credentials and Version
CLIENT_ID = 'JCE5JUSR5SWQ3W4SPFMYIFU2TJ5O2KG4UQF0HGBWK1ATHYG3' # your Foursquare ID
CLIENT_SECRET = 'TKMJYHPMH2LTSPGXOWIKBZM0WXYDGTB3KDIDNUOBNA4IUG0H' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
```

```
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: JCE5JUSR5SWQ3W4SPFMYIFU2TJ5O2KG4UQF0HGBWK1ATHYG3
CLIENT_SECRET:TKMJYHPMH2LTSPGXOWIKBZM0WXYDGTB3KDIDNUOBNA4IUG0H
```

```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

```
LIMIT = 100 # limit of number of venues returned by Foursquare API

def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={}
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
```

Creating a copy...

```
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                    'Neighborhood Latitude',
                    'Neighborhood Longitude',
                    'Venue',
```

```
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

```
# type your answer here

manhattan_venues = getNearbyVenues(names=manhattan_data['Neighborhood'],
                                   latitudes=manhattan_data['Latitude'],
                                   longitudes=manhattan_data['Longitude']
                                  )
```

```
Marble Hill
Chinatown
Washington Heights
Inwood
Hamilton Heights
Manhattanville
Central Harlem
East Harlem
Upper East Side
Yorkville
Lenox Hill
Roosevelt Island
Upper West Side
Lincoln Square
Clinton
Midtown
Murray Hill
Chelsea
Greenwich Village
East Village
Lower East Side
Tribeca
Little Italy
Soho
West Village
Manhattan Valley
Morningside Heights
Gramercy
Battery Park City
Financial District
Carnegie Hill
Noho
Civic Center
```

Creating a copy...

```
Tudor City
Stuyvesant Town
Flatiron
```

```
print(manhattan_venues.shape)
manhattan_venues.head()
```

```
(3176, 7)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue
0	Marble Hill	40.876551	-73.91066	Arturo's	40.874412	·
1	Marble Hill	40.876551	-73.91066	Bikram Yoga	40.876844	·
2	Marble Hill	40.876551	-73.91066	Tibbett Diner	40.880404	·
3	Marble Hill	40.876551	-73.91066	Dunkin'	40.877136	·
4	Marble Hill	40.876551	-73.91066	Starbucks	40.877531	·

```
manhattan_venues.groupby('Neighborhood').count()
```

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Lo
Battery Park City	70	70	70	70	
Carnegie Hill	90	90	90	90	
Central Harlem	41	41	41	41	
Chelsea	100	100	100	100	
Chinatown	100	100	100	100	
Civic Center	100	100	100	100	
Clinton	100	100	100	100	
East Harlem	41	41	41	41	
East Village	100	100	100	100	
Financial District	100	100	100	100	
Flatiron	100	100	100	100	
Gramercy	84	84	84	84	
Greenwich Village	100	100	100	100	
Hamilton Heights	59	59	59	59	
Hudson Yards	70	70	70	70	
Inwood	56	56	56	56	
Lenox Hill	100	100	100	100	
Lincoln Square	96	96	96	96	
Little Italy	100	100	100	100	
Lower East Side	55	55	55	55	
Manhattan Valley	52	52	52	52	
Manhattanville	41	41	41	41	
Marble Hill	25	25	25	25	
Midtown	100	100	100	100	
[illegible]	100	100	100	100	
Morningside Heights	38	38	38	38	
Murray Hill	100	100	100	100	
Noho	100	100	100	100	
Roosevelt Island	25	25	25	25	
Soho	100	100	100	100	
Stuyvesant Town	18	18	18	18	
Sutton Place	100	100	100	100	
Tribeca	82	82	82	82	

Tudor City	78	78	78	78
Turtle Bay	100	100	100	100
Upper East Side	89	89	89	89
Upper West Side	77	77	77	77

```
# one hot encoding
manhattan_onehot = pd.get_dummies(manhattan_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
manhattan_onehot['Neighborhood'] = manhattan_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [manhattan_onehot.columns[-1]] + list(manhattan_onehot.columns[:-1])
manhattan_onehot = manhattan_onehot[fixed_columns]

manhattan_onehot.head()
```

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Arcade	A Restau
0	Marble Hill	0	0	0	0	0	0	0	
1	Marble Hill	0	0	0	0	0	0	0	
2	Marble Hill	0	0	0	0	0	0	0	
3	Marble Hill	0	0	0	0	0	0	0	
4	Marble Hill	0	0	0	0	0	0	0	

```
manhattan_grouped = manhattan_onehot.groupby('Neighborhood').mean().reset_index()
manhattan_grouped
```

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Arcade	Rest
0	Battery Park City	0.000000	0.00	0.00	0.00000	0.014286	0.00	0.000000	0
1	Carnegie Hill	0.000000	0.00	0.00	0.00000	0.011111	0.00	0.000000	0
2	Central Harlem	0.000000	0.00	0.00	0.04878	0.048780	0.00	0.000000	0
3	Chelsea	0.000000	0.00	0.00	0.00000	0.040000	0.00	0.000000	0
4	Chinatown	0.000000	0.00	0.00	0.00000	0.040000	0.00	0.000000	0
5	Civic Center	0.000000	0.00	0.00	0.00000	0.030000	0.01	0.000000	0
6	Clinton	0.000000	0.00	0.00	0.00000	0.030000	0.00	0.000000	0
7	East Harlem	0.000000	0.00	0.00	0.00000	0.000000	0.00	0.000000	0
8	East Village	0.000000	0.00	0.00	0.00000	0.020000	0.00	0.000000	0
9	Financial District	0.000000	0.00	0.00	0.00000	0.030000	0.00	0.000000	0
10	Flatiron	0.000000	0.00	0.00	0.00000	0.030000	0.00	0.000000	0
11	Gramercy	0.000000	0.00	0.00	0.00000	0.047619	0.00	0.011905	0
12	Greenwich Village	0.010000	0.00	0.00	0.00000	0.020000	0.00	0.000000	0
13	Hamilton Heights	0.000000	0.00	0.00	0.00000	0.000000	0.00	0.000000	0
14	Hudson Yards	0.000000	0.00	0.00	0.00000	0.071429	0.00	0.000000	0
15	Inwood	0.000000	0.00	0.00	0.00000	0.035714	0.00	0.000000	0
16	Lenox Hill	0.000000	0.00	0.01	0.00000	0.000000	0.00	0.000000	0
17	Lincoln Square	0.000000	0.00	0.00	0.00000	0.031250	0.00	0.000000	0
18	Little Italy	0.000000	0.00	0.00	0.00000	0.000000	0.00	0.000000	0
19	Lower East	0.000000	0.00	0.00	0.00000	0.018182	0.00	0.000000	0
20	Valley	[illegible]	0.00	0.00	0.00000	0.019231	0.00	0.000000	0
21	Manhattanville	0.000000	0.00	0.00	0.00000	0.024390	0.00	0.000000	0
22	Marble Hill	0.000000	0.00	0.00	0.00000	0.040000	0.00	0.000000	0
23	Midtown	0.000000	0.00	0.00	0.00000	0.020000	0.00	0.000000	0
24	Midtown South	0.000000	0.00	0.00	0.00000	0.030000	0.00	0.000000	0
25	Morningside Heights	0.000000	0.00	0.00	0.00000	0.078947	0.00	0.000000	0
26	Murray Hill	0.000000	0.00	0.00	0.00000	0.020000	0.00	0.000000	0

27	Noho	0.000000	0.00	0.00	0.00000	0.030000	0.00	0.000000	0
28	Roosevelt Island	0.000000	0.00	0.00	0.00000	0.000000	0.00	0.000000	0
29	Soho	0.000000	0.00	0.00	0.00000	0.020000	0.00	0.000000	0
30	Stuyvesant Town	0.000000	0.00	0.00	0.00000	0.000000	0.00	0.000000	0
31	Sutton Place	0.000000	0.01	0.00	0.00000	0.010000	0.00	0.000000	0
32	Tribeca	0.000000	0.00	0.00	0.00000	0.036585	0.00	0.000000	0
33	Tudor City	0.000000	0.00	0.00	0.00000	0.012821	0.00	0.000000	0
34	Turtle Bay	0.000000	0.00	0.00	0.00000	0.010000	0.00	0.000000	0
35	Upper East Side	0.000000	0.00	0.00	0.00000	0.011236	0.00	0.000000	0
36	Upper West Side	0.000000	0.00	0.00	0.00000	0.025974	0.00	0.000000	0
	Washington								

```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = manhattan_grouped['Neighborhood']

                                            ouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(manhattan_grouped.iloc[ind,

neighborhoods_venues_sorted.head()
```

Creating a copy...

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
0	Battery Park City	Park	Hotel	Gym	Boat or Ferry	C
1	Carnegie Hill	Coffee Shop	Café	Yoga Studio	Gym	F

```
# set number of clusters
kclusters = 5

manhattan_grouped_clustering = manhattan_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(manhattan_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

```
array([1, 0, 0, 0, 1, 1, 1, 3, 0, 1], dtype=int32)
```

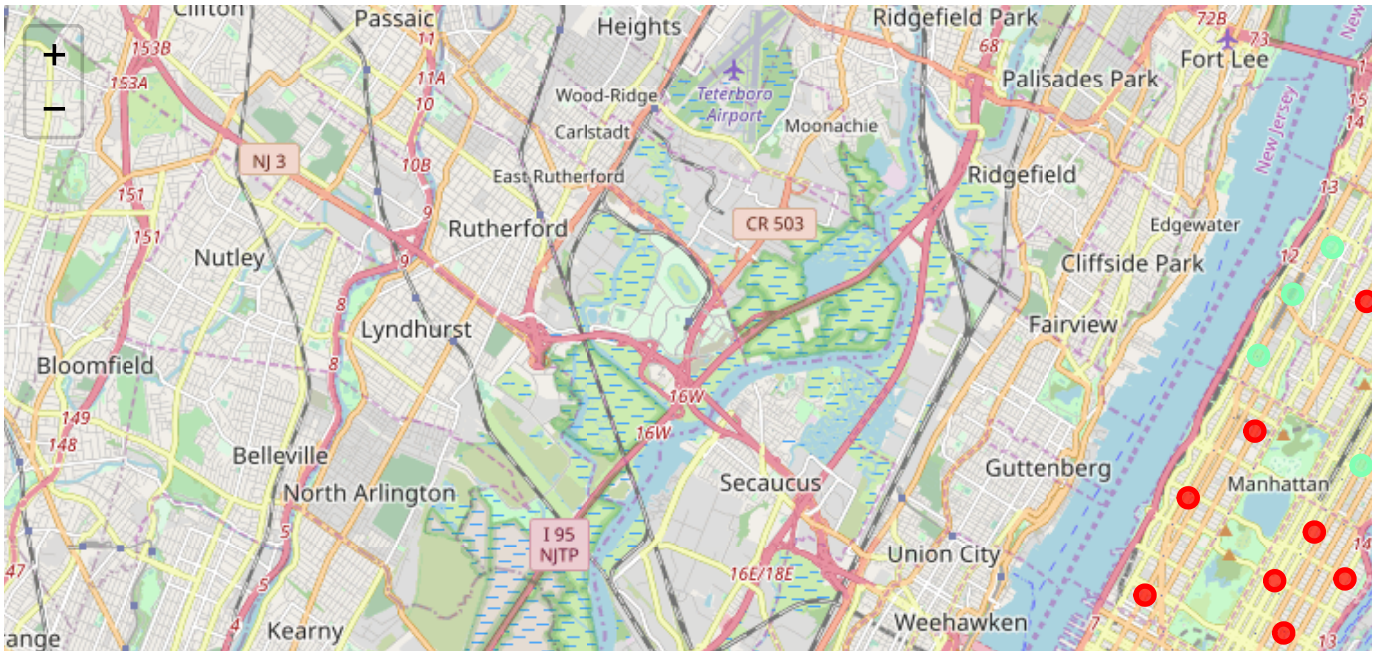
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

manhattan_merged = manhattan_data

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
manhattan_merged = manhattan_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='N

manhattan_merged.head() # check the last columns!
```

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Mos Common Venu
0	Manhattan	Marble Hill	40.876551	-73.910660	4	Sandwich Place	Gym	Coffee Sho
1	Manhattan	Chinatown	40.715618	-73.994279	1	Chinese Restaurant	Cocktail Bar	Bake
2	Manhattan	Washington Heights	40.851903	-73.936900	3	Café	Bakery	Grocery Sto
3	Manhattan	Inwood	40.867684	-73.921210	3	Mexican Restaurant	Café	Loung
		Heights	823604	-73.949688	3	Pizza Place	Café	Coffee Sho

Creating a copy...

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
```

```
for lat, lon, poi, cluster in zip(manhattan_merged['Latitude'], manhattan_merged['Longitude'], manhat
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```



Examing Clusters

```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 0, manhattan_merged.columns[[1] + list(ran
```

Creating a copy...

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mc
6	Central Harlem	Seafood Restaurant	Bar	African Restaurant	American Restaurant	Chinese
8	Upper East Side	Italian Restaurant	Coffee Shop	Bakery	Gym / Fitness Center	
9	Yorkville	Italian Restaurant	Coffee Shop	Gym	Bar	D
10	Lenox Hill	Italian Restaurant	Coffee Shop	Sushi Restaurant	Pizza Place	(
12	Upper West Side	Italian Restaurant	Bakery	Bar	Coffee Shop	Thai

```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 1, manhattan_merged.columns[[1] + list(rar
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mos
1	Chinatown	Chinese Restaurant	Cocktail Bar	Bakery	American Restaurant	
14	Clinton	Italian Restaurant	Gym / Fitness Center	Theater	Coffee Shop	
15	Midtown	Coffee Shop	Hotel	Theater	Sandwich Place	Sporti
16	Murray Hill	Sandwich Place	Hotel	Bar	Coffee Shop	Sushi F
22	Little Italy	Café	Bakery	Hotel	Bubble Tea Shop	Sandv
28	Battery Park City	Park	Hotel	Gym	Boat or Ferry	Co
29	Financial District	Coffee Shop	Bar	Pizza Place	Hotel	Co
32	Civic Center	Coffee Shop	French Restaurant	Hotel	Cocktail Bar	Yo
33	Midtown South	Korean Restaurant	Hotel	Japanese Restaurant	Dessert Shop	Bu
		American Restaurant	Gym / Fitness Center	Japanese Restaurant	Italian Restaurant	F
39	Hudson Yards	American	Hotel	Gym / Fitness	Italian Restaurant	

Creating a copy...

```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 2, manhattan_merged.columns[[1] + list(rar
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
37	Stuyvesant Town	Boat or Ferry	Park	Bar	Gas Station	Ba

```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 3, manhattan_merged.columns[[1] + list(ran
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
2	Washington Heights	Café	Bakery	Grocery Store	Deli / Bodega	Chinese
3	Inwood	Mexican Restaurant	Café	Lounge	Pizza Place	
4	Hamilton Heights	Pizza Place	Café	Coffee Shop	Deli / Bodega	Mexican
5	Manhattanville	Coffee Shop	Seafood Restaurant	Italian Restaurant	Park	Mexican
7	East Harlem	Mexican Restaurant	Bakery	Deli / Bodega	Thai Restaurant	Lat
11	Roosevelt Island	Pizza Place	Park	Gym	Coffee Shop	Kosher
26	Morningside Heights	Park	Bookstore	Coffee Shop	American Restaurant	D
36	Tudor City	Park	Café	Greek Restaurant	Mexican Restaurant	D

Creating a copy...

```
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == 4, manhattan_merged.columns[[1] + list(ran
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
0	Marble Hill	Sandwich Place	Gym	Coffee Shop	Yoga Studio	De

After we have identified the Clusters of interest and

```
brooklyn_data = neighborhoods[neighborhoods['Borough'] == 'Brooklyn'].reset_index(drop=True)
# brooklyn_data.head()

address = 'Brooklyn, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Brooklyn are {}, {}.'.format(latitude, longitude))

# Define Foursquare Credentials and Version
CLIENT_ID = 'KB1MGMXSFCU5NXWQWFVLPFNXZE3VRN3G5WC05TC0YNPUYDUO' # your Foursquare ID
CLIENT_SECRET = 'H1X3QY13IMSN3H1BTZQMKOOZAG2YAQXQWW1CNOND4RHTJVY3' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)

LIMIT = 100 # limit of number of venues returned by Foursquare API

def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={}
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # [illegible] information for each nearby venue

            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                    'Neighborhood Latitude',
                    'Neighborhood Longitude',
                    'Venue',
```

Creating a copy...

```
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)

brooklyn_venues = getNearbyVenues(names=brooklyn_data['Neighborhood'],
                                  latitudes=brooklyn_data['Latitude'],
                                  longitudes=brooklyn_data['Longitude']
                                 )

print(brooklyn_venues.shape)
brooklyn_venues.head()
brooklyn_venues.groupby('Neighborhood').count()

# one hot encoding
brooklyn_onehot = pd.get_dummies(brooklyn_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
brooklyn_onehot['Neighborhood'] = brooklyn_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [brooklyn_onehot.columns[-1]] + list(brooklyn_onehot.columns[:-1])
brooklyn_onehot = brooklyn_onehot[fixed_columns]

brooklyn_onehot.head()

brooklyn_grouped = brooklyn_onehot.groupby('Neighborhood').mean().reset_index()
brooklyn_grouped

num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
                                  DataFrame(columns=columns)
                                  hborhood'] = brooklyn_grouped['Neighborhood']

for ind in np.arange(brooklyn_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(brooklyn_grouped.iloc[ind,

neighborhoods_venues_sorted.head()
```

Creating a copy...

```
The geograpical coordinate of Brooklyn are 40.6501038, -73.9495823.
Your credentails:
CLIENT_ID: KB1MGMXSFCU5NXWQWFVLPFNXZE3VRN3G5WC05TC0YNPUYDUO
CLIENT_SECRET:H1X3QY13IMSN3H1BTZQMKOOZAG2YAQXQWW1CNOND4RHTJVY3
Bay Ridge
Bensonhurst
Sunset Park
Greenpoint
Gravesend
Brighton Beach
Sheepshead Bay
Manhattan Terrace
Flatbush
Crown Heights
East Flatbush
Kensington
Windsor Terrace
Prospect Heights
Brownsville
Williamsburg
Bushwick
Bedford Stuyvesant
Brooklyn Heights
Cobble Hill
Carroll Gardens
Red Hook
Gowanus
Fort Greene
Park Slope
Cypress Hills
East New York
Starrett City
Canarsie
Flatlands
Mill Island
Manhattan Beach
Coney Island
Bath Beach
Borough Park
Dyker Heights
Gerritsen Beach
Marine Park
Clinton Hill
Sea Gate
Downtown
Boerum Hill
Prospect Lefferts Gardens
```



```
Midwood
Prospect Park South
Georgetown
East Williamsburg
North Side
South Side
Ocean Parkway
Fort Hamilton
Ditmas Park
Wingate
Rugby
Remsen Village
New Lots
Paerdegat Basin
```

```
Paerdegat Basin
Mill Basin
Fulton Ferry
Vinegar Hill
Weeksville
Broadway Junction
Dumbo
Homecrest
Highland Park
Madison
Erasmus
(2734, 7)
```

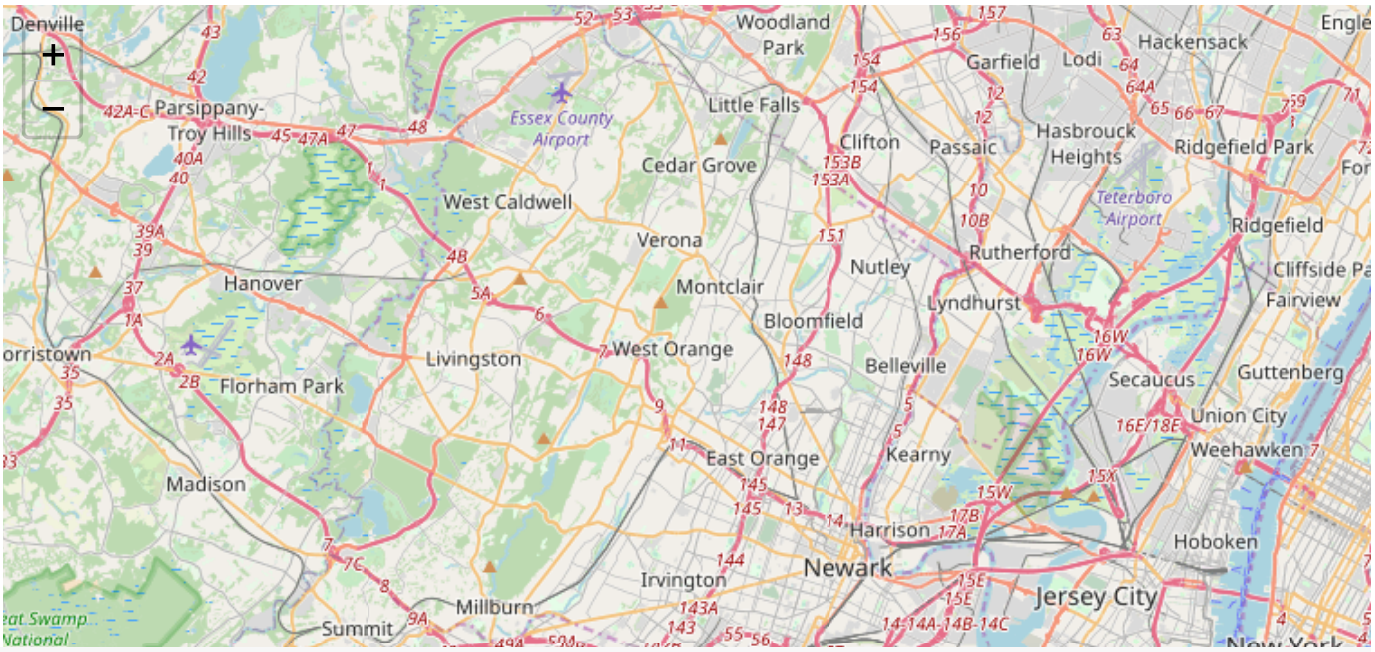
	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mos
0	Bath Beach	Pharmacy	Chinese Restaurant	Pizza Place	Sushi Restaurant	Italian R
1	Bay Ridge	Italian Restaurant	Pizza Place	Spa	Bar	Greek R
2	Bedford Stuyvesant	Coffee Shop	Café	Pizza Place	Bar	C

```
# set number of clusters
kclusters = 5

brooklyn_grouped_clustering = brooklyn_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(brooklyn_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]

# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

brooklyn_merged = brooklyn_data

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
brooklyn_merged = brooklyn_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Nei

brooklyn_merged.head() # check the last columns!

n=[latitude, longitude], zoom_start=11)
```

Creating a copy...

```
# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(brooklyn_merged['Latitude'], brooklyn_merged['Longitude'], brooklyn
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon]
```

```
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```




```
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 0, brooklyn_merged.columns[[1] + list(range(
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
45	Bergen Beach	Harbor / Marina	Donut Shop	Baseball Field	Playground	Athleti

```
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 1, brooklyn_merged.columns[[1] + list(range(
```

Creating a copy...

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
0	Bay Ridge	Italian Restaurant	Pizza Place	Spa	Bar	Greek
3	Greenpoint	Bar	Pizza Place	Cocktail Bar	Coffee Shop	Y
12	Windsor Terrace	Deli / Bodega	Café	Park	Grocery Store	
13	Prospect Heights	Bar	Mexican Restaurant	Wine Shop	Gourmet Shop	
14	Brownsville	Restaurant	Park	Chinese Restaurant	Moving Target	
15	Williamsburg	Bar	Coffee Shop	Grocery Store	Bagel Shop	Y
16	Bushwick	Bar	Coffee Shop	Mexican Restaurant	Deli / Bodega	Thr
17	Bedford Stuyvesant	Coffee Shop	Café	Pizza Place	Bar	C
18	Brooklyn Heights	Yoga Studio	Deli / Bodega	Park	Pizza Place	
19	Cobble Hill	Pizza Place	Bakery	Playground	Coffee Shop	
20	Carroll Gardens	Italian Restaurant	Coffee Shop	Pizza Place	Bakery	C
21	Red Hook	Seafood Restaurant	Art Gallery	Bar	American Restaurant	
22	Gowanus	Furniture / Home Store	Italian Restaurant	Bar	Chinese Restaurant	
23	Fort Greene	Italian Restaurant	Coffee Shop	Cocktail Bar	Flower Shop	
24	Park Slope	Coffee Shop	Burger Joint	Pub	Bakery	E
31	Manhattan Beach	Bus Stop	Ice Cream Shop	Food	Beach	Harb
36	Gerritsen Beach	Pizza Place	Bar	Convenience Store	Gas Station	S
		estaurant	Pizza Place	Thai Restaurant	Wine Shop	Indian
39	Sea Gate	Beach	Spa	Bus Line	Bus Station	
40	Downtown	Burger Joint	Coffee Shop	Pizza Place	Sandwich Place	
41	Boerum Hill	Coffee Shop	Bar	Dance Studio	Sandwich Place	Arts & C
42	Prospect Lefferts Gardens	Bakery	Café	Deli / Bodega	Pizza Place	
49	East Williamsburg	Bar	Deli / Bodega	Coffee Shop	Cocktail Bar	

50	North Side	Coffee Shop	Pizza Place	Yoga Studio	Wine Bar	
51	South Side	Bar	Pizza Place	Coffee Shop	American Restaurant	
59	Paerdegat Basin	Moving Target	Asian Restaurant	Harbor / Marina	Food	A
61	Fulton Ferry	Park	American Restaurant	Ice Cream Shop	Scenic Lookout	Bo
62	Vinegar Hill	Food Truck	Coffee Shop	Art Gallery	Café	

```
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 2, brooklyn_merged.columns[[1] + list(range(
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5t
1	Bensonhurst	Sushi Restaurant	Pizza Place	Ice Cream Shop	Grocery Store	It
2	Sunset Park	Bakery	Pizza Place	Mexican Restaurant	Latin American Restaurant	
4	Gravesend	Pizza Place	Bakery	Bus Station	Chinese Restaurant	M
5	Brighton Beach	Eastern European Restaurant	Russian Restaurant	Beach	Restaurant	
6	Sheepshead Bay	Turkish Restaurant	Dessert Shop	Sandwich Place	Yoga Studio	
7	Manhattan Terrace	Pizza Place	Donut Shop	Grocery Store	Ice Cream Shop	
9	Crown Heights	Pizza Place	Museum	Café	Cosmetics Shop	Fri
11	Kensington	Grocery Store	Thai Restaurant	Pizza Place	Ice Cream Shop	
25	Cypress Hills	Fried Chicken Joint	Ice Cream Shop	Pizza Place	Donut Shop	
33	Bath Beach	Pharmacy	Chinese Restaurant	Pizza Place	Sushi Restaurant	It
34	Borough Park	Bank	Pizza Place	Hotel	Café	
35	Dyker Heights	Burger Joint	Dance Studio	Golf Course	Food	
37	Marine Park	Baseball Field	Doctor's Office	Basketball Court	Gym	A
43	Ocean Hill	Deli / Bodega	Grocery Store	Southern / Soul Food Restaurant	Playground	
44	City Line	Donut Shop	Bakery	Grocery Store	Shoe Store	
46	Midwood	Pizza Place	Ice Cream Shop	Bakery	Pharmacy	Vi
48	Georgetown	Bank	Pharmacy	Breakfast Spot	Donut Shop	
		taurant	Gym / Fitness Center	Sake Bar	General Entertainment	
53	Fort Hamilton	Pizza Place	Deli / Bodega	Pharmacy	Chinese Restaurant	
55	Wingate	Fried Chicken Joint	Fast Food Restaurant	Juice Bar	Bus Station	
58	New Lots	Metro Station	Pizza Place	Fried Chicken Joint	Pharmacy	
60	Mill Basin	Pizza Place	Chinese Restaurant	Japanese Restaurant	Bank	
64	Broadway Junction	Donut Shop	Diner	Fried Chicken Joint	Dessert Shop	

```
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 3, brooklyn_merged.columns[[1] + list(range(
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
8	Flatbush	Caribbean Restaurant	Coffee Shop	Pharmacy	Mexican Restaurant	
10	East Flatbush	Food & Drink Shop	Pharmacy	Fast Food Restaurant	Park	Harc
26	East New York	Deli / Bodega	Fast Food Restaurant	Plaza	Event Service	F
27	Starrett City	Chinese Restaurant	Pharmacy	Caribbean Restaurant	Shopping Mall	
28	Canarsie	Grocery Store	Chinese Restaurant	Caribbean Restaurant	Asian Restaurant	
29	Flatlands	Pharmacy	Fried Chicken Joint	Caribbean Restaurant	Fast Food Restaurant	
32	Coney Island	Park	Caribbean Restaurant	Beach	Baseball Stadium	Theme
47	Prospect Park South	Caribbean Restaurant	Pizza Place	Fast Food Restaurant	Mobile Phone Shop	Gr
54	Ditmas Park	Caribbean Restaurant	Pizza Place	Deli / Bodega	Women's Store	Chinese
56	Rugby	Caribbean Restaurant	Bank	Grocery Store	Salon / Barbershop	F
57	Remsen Village	Caribbean Restaurant	Fast Food Restaurant	Deli / Bodega	Fish Market	S
		Caribbean				

```
brooklyn_merged.loc[brooklyn_merged['Cluster Labels'] == 4, brooklyn_merged.columns[[1] + list(range(
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
30	Mill Island	Pool	Women's Store	Filipino Restaurant	Factory	Falafel

Creating a copy...

The same with Queens

```
queens_data = neighborhoods[neighborhoods['Borough'] == 'Queens'].reset_index(drop=True)
# queens_data.head()

address = 'Queens, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Queens are {}, {}.'.format(latitude, longitude))
```

```
# Define Foursquare Credentials and Version
CLIENT_ID = 'KB1MGMXSFCU5NXWQWFVLPFNXZE3VRN3G5WC05TC0YNPUYDUO' # your Foursquare ID
CLIENT_SECRET = 'H1X3QY13IMSN3H1BTZQMKOOZAG2YAQXQWW1CNOND4RHTJVY3' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)

LIMIT = 100 # limit of number of venues returned by Foursquare API

def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={}
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
                                                                    ghborhood',
                                                                    atitude',
                                                                    ongitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)

queens_venues = getNearbyVenues(names=queens_data['Neighborhood'],
                                latitudes=queens_data['Latitude'],
                                longitudes=queens_data['Longitude']
                               )
```

Creating a copy...

```
print(queens_venues.shape)
queens_venues.head()
queens_venues.groupby('Neighborhood').count()

# one hot encoding
queens_onehot = pd.get_dummies(queens_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
queens_onehot['Neighborhood'] = queens_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [queens_onehot.columns[-1]] + list(queens_onehot.columns[:-1])
queens_onehot = queens_onehot[fixed_columns]

queens_onehot.head()

queens_grouped = queens_onehot.groupby('Neighborhood').mean().reset_index()
queens_grouped

num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = queens_grouped['Neighborhood']

for ind in np.arange(queens_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(queens_grouped.iloc[ind, :]

neighborhoods_venues_sorted.head()
```

Sunnyside
East Elmhurst
Maspeth
Ridgewood
Glendale
Rego Park
Woodhaven
Ozone Park
South Ozone Park
College Point
Whitestone
Bayside
Auburndale
Little Neck
Douglaston
Glen Oaks
Bellerose
Kew Gardens Hills
Fresh Meadows
Briarwood
Jamaica Center
Oakland Gardens
Queens Village
Hollis
South Jamaica
St. Albans
Rochdale
Springfield Gardens
Cambria Heights
Rosedale
Far Rockaway
Broad Channel
Breezy Point
Steinway
Beechhurst
Bay Terrace
Edgemere
Arverne
Rockaway Beach
Neponsit
Murray Hill
Floral Park
Holliswood
Jamaica Estates
Queensboro Hill
Hillcrest
Ravenswood
Lindenwood

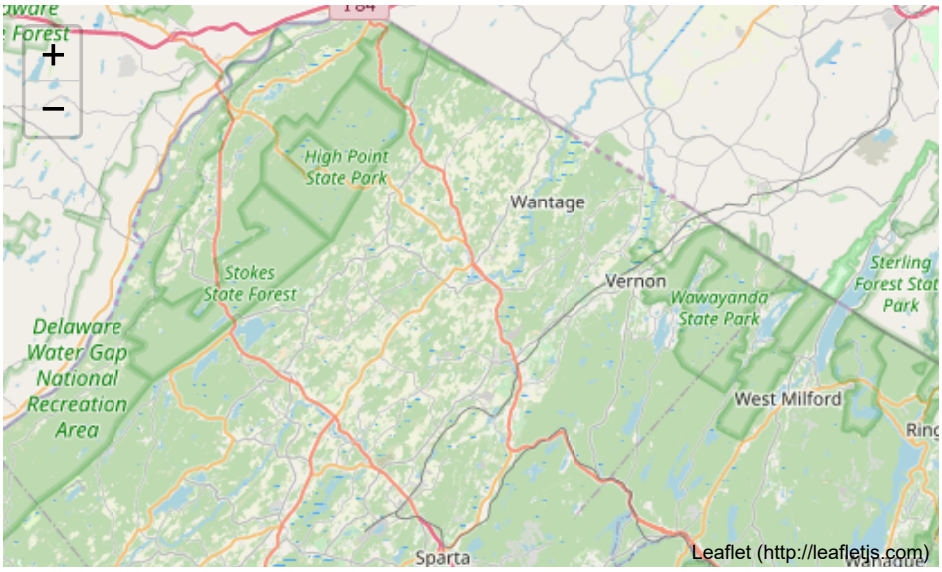
Rockaway Park
Somerville
Brookville
Bellaire
North Corona
Forest Hills Gardens
Jamaica Hills
Utopia
Pomonok
Astoria Heights
Hunters Point
Sunnyside Gardens
Blissville
Roxbury

```
Roxbury
Middle Village
Malba
Hammels
Bayswater
Queensbridge
(2119, 7)
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Mo
0	Arverne	Surf Spot	Sandwich Place	Metro Station	Playground	
1	Astoria	Bar	Middle Eastern Restaurant	Greek Restaurant	Hookah Bar	Seafood
2	Astoria Heights	Plaza	Playground	Burger Joint	Hostel	Italian
3	Auburndale	Italian Restaurant	Bar	Noodle House	Fast Food Restaurant	Korean
4	Bay Terrace	Clothing Store	Women's Store	American Restaurant	Lingerie Store	Mobile P

```
# set number of clusters
kclusters = 5

queens_grouped_clustering = queens_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(queens_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]

Creating a copy...
t(0, 'Cluster Labels', kmeans.labels_)

queens_merged = queens_data

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
queens_merged = queens_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighbo

queens_merged.head() # check the last columns!

# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
```

```
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(queens_merged['Latitude'], queens_merged['Longitude'], queens_merge
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```




```
queens_merged.loc[queens_merged['Cluster Labels'] == 0, queens_merged.columns[[1] + list(range(5, que
```

Creating a copy...

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th M
0	Astoria	Bar	Middle Eastern Restaurant	Greek Restaurant	Hookah Bar	
1	Woodside	Grocery Store	Latin American Restaurant	Bakery	Thai Restaurant	Filipinc
2	Jackson Heights	Latin American Restaurant	Peruvian Restaurant	South American Restaurant	Bakery	
3	Elmhurst	Thai Restaurant	Mexican Restaurant	South American Restaurant	Vietnamese Restaurant	
4	Howard Beach	Italian Restaurant	Pharmacy	Bagel Shop	Deli / Bodega	San
5	Corona	Mexican Restaurant	Convenience Store	Bakery	Deli / Bodega	
6	Forest Hills	Gym	Gym / Fitness Center	Yoga Studio	Pizza Place	
7	Kew Gardens	Chinese Restaurant	Bar	Indian Restaurant	Bank	
8	Richmond Hill	Deli / Bodega	Latin American Restaurant	Pizza Place	Lounge	
9	Flushing	Bubble Tea Shop	Hotpot Restaurant	Chinese Restaurant	Bakery	Korear
10	Long Island City	Coffee Shop	Hotel	Gym / Fitness Center	Pizza Place	
11	Sunnyside	Pizza Place	Italian Restaurant	Chinese Restaurant	Discount Store	
12	East Elmhurst	Donut Shop	Ice Cream Shop	Indie Movie Theater	Hotel Bar	
13	Maspeth	Pizza Place	Diner	Mobile Phone Shop	Chinese Restaurant	G
14	Ridgewood	Café	Pizza Place	Deli / Bodega	Bank	
16	Rego Park	Bakery	Chinese Restaurant	Pharmacy	Peruvian Restaurant	
		Bodega	Bank	Pharmacy	Park	S
18	Ozone Park	Pizza Place	Pharmacy	Gym	Diner	
19	South Ozone Park	Deli / Bodega	Park	Fast Food Restaurant	Bar	
20	College Point	Deli / Bodega	Pizza Place	Asian Restaurant	Latin American Restaurant	
21	Whitestone	Dance Studio	Deli / Bodega	Bubble Tea Shop	Candy Store	Filipinc
22	Bayside	Bar	Indian Restaurant	Sushi Restaurant	American Restaurant	