WHICH IS THE BEST

NEIGHBORHOOD

TO OPEN A

CROSSFIT

GYM IN

NEW YORK CITY ?

EDUARDO VIEIRA - IBM- COURSERA - APPLIED DATA SCIENCE CAPSTONE - THE BATTLE OF NEIGHBORHOODS (WEEK 2)

WHY **NEW YORK**?

The City of New York is the most populous city in the United States and also the most densely populated major city in the United States. The New York metropolitan area is estimated to produce a gross metropolitan product (GMP) of US$1.9 trillion. If greater New York City were a sovereign state, it would have the 12th highest GDP in the world. New York is home to the highest number of billionaires of any city in the world.

WHY **CROSSFIT**?

CrossFit is a lifestyle characterized by safe, effective exercise and sound nutrition. CrossFit can be used to accomplish any goal, from improved health to weight loss to better performance. The program works for everyone—people who are just starting out and people who have trained for years.

The magic is in the movements. Workouts are different every day and modified to help each athlete achieve his or her goals. CrossFit workouts can be adapted for people at any age and level of fitness

Off the carbs, off the couch. The CrossFit lifestyle—a combination of diet and exercise—is the key to fitness and long-term health.

PROBLEM

First of all, for choosing a place to open a new business, it is necessary to study the best location, according to many criterions, as population, density, per capita... and also if there are similar businesses in the region.

CrossFit is the fastest-growing gym in the world. It is a lifestyle, it is socialization, it is living well and it is also politically and ecologically correct.

Beeing NYC the capital of the World, why don’t open a CrossFit gym in New York City?

Choosing the best Neighborhood is the objective of this project.

DATA ACQUISITION

NYC OPEN DATA – Boroughs, neighborhoods and geographical coordinates (https://data.cityofnewyork.us/).

GEOPY – Geographical coordinates o any place.

FOURSQUARE API – To search venues by name, category, location, and many others queries.

WIKIPEDIA – Where I can find tons of good and free information about New York City, its Boroughs and Neighborhoods.

ANY DATA REPOSITORY – Where I can find interesting data for the project, to corroborate my analysis.

METHODOLOGY

1) Getting Officials Data about Boroughs and Neighborhoods of NYC.

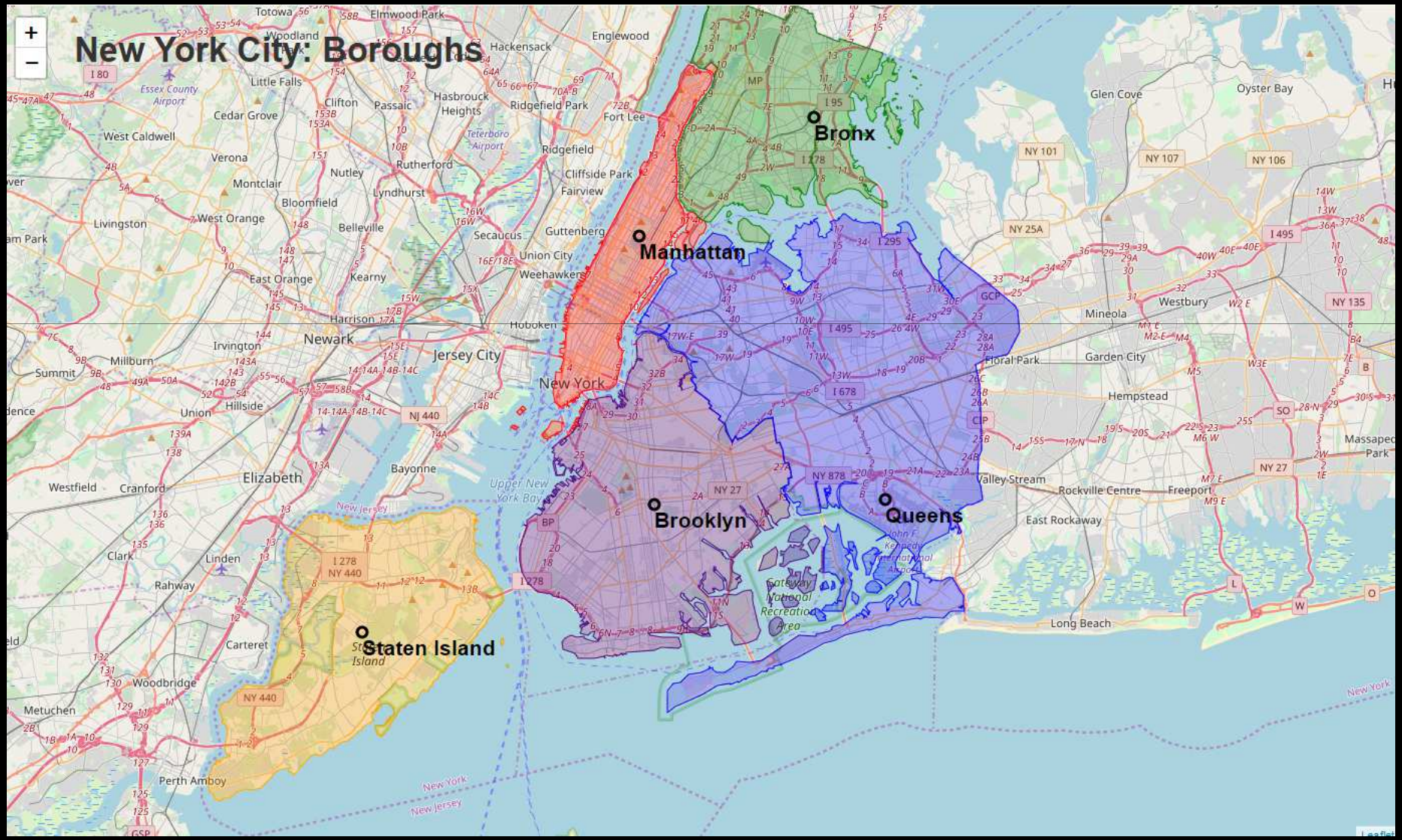
2) Using the Foursquare API to search about “CrossFit” in NYC.
   a) Only venues with “CrossFit” in the name (Officials CrossFit Gyms have this authorization!)
   b) Deleting duplicates and different Cities, according as criterions stablished.
   c) Obtaining two files: Neighboorhods/Boroughs with and without CrossFits gyms.

3) Getting Statistical Data about NYC, like Population, Density, Area, Per Capita and etc.
   a) Creating a file with all Statistics Data about Boroughs and quantities of CrossFit gyms.
   b) Normalizing these data to compare and to obtain the score of each Borough.

4) After choosing the Borough, analyze the Neighborhoods, using the same techniques availables.

5) Creating a lot of MAPs and Plotting Statisticals Graphics to illustrate the analysis.

METHODOLOGY



METHODOLOGY - WORKING ON THE PROJECT

1) Getting the all boroughs and Neighborhoods of New York City and cleaning the file.

GETTING ALL THE NEIGHBORHOODS / ZONE OF THE NEW YORK CITY OFFICIAL WEBSITE

[8]:
```
# OFFICIAL SITE OF NEW YORK CITY
url = "https://data.cityofnewyork.us/api/views/xyye-rtrs/rows.json?accessType=DOWNLOAD"
newyork_data2 = requests.get(url).json()
#newyork_data2
```

[10]:
```
nyb = pd.DataFrame(newyork_data2['data'])
```

[11]:
```
#nyb.shape
```

[44]:
```
nyb.head(2)
```

[44]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	row-7q9n-adg4_xvee	00000000-0000-0000-BC1E-0359B8CD43D9	0	1450726363	None	1450726363	None	{ }	POINT (-73.8472005205491 40.89470517661004)	1	Wakefield	1	Wakefield			0.0	Bronx
1	row-s456-uyfi_n9s7	00000000-0000-0000-20BE-C45ADF984BA3	0	1450726363	None	1450726363	None	{ }	POINT (-73.82993910812405 40.87429419303015)	2	Co-op City	2	Co-op	City		0.0	Bronx

[45]:
```
#nyb[16].unique()
```

[46]:
```
nyb2 = nyb[[8,9,10,16]]
```

[47]:
```
nyb2.columns = ['coordenadas','numero','neighborhood','borough']
```

[48]:
```
nyb2.head(2)
```

[48]:

	coordenadas	numero	neighborhood	borough
0	POINT (-73.8472005205491 40.89470517661004)	1	Wakefield	Bronx

METHODOLOGY - WORKING ON THE PROJECT

2) Dataframe neighborhoods_all, contains all boroughs, neighborhoods and coordinates.

```
[56]: neighborhoods_all.head()
```

[56]:

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.89470517661004	-73.8472005205491
1	Bronx	Co-op City	40.87429419303015	-73.82993910812405
2	Bronx	Eastchester	40.88755567735082	-73.82780644716419
3	Bronx	Fieldston	40.895437426903875	-73.90564259591689
4	Bronx	Riverdale	40.89083449389134	-73.91258546108577

3) Creating a Dataframe only with the boroughs.

CREATING BOROUGHS FILE AND OBTAINING GEOGRAPHICAL COORDINATES

```
[193]: # pegando as coordenadas dos Boroughs para futuro uso
       Boroughs = neighborhoods_all.Borough.unique()
```

```
[194]: Boroughs
```

```
[194]: array(['Bronx', 'Manhattan', 'Brooklyn', 'Queens', 'Staten Island'],
             dtype=object)
```

METHODOLOGY - WORKING ON THE PROJECT

4) Searching the coordinates of each Borough.

[66]:
```
Boroughs_latlong = pd.DataFrame(columns=['Borough','Latitude','Longitude'])

for borou in Boroughs:

    address = borou+',New York City, NY'

    geolocator = Nominatim(user_agent="ny_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    Boroughs_latlong = Boroughs_latlong.append({'Borough': borou,
                                        'Latitude': latitude,
                                        'Longitude': longitude}, ignore_index=True)
    print('The geograpical coordinate of {} in New York City are {}, {}.'.format(borou, latitude, longitude))
```

```
The geograpical coordinate of Bronx in New York City are 40.85048545, -73.8404035580209.
The geograpical coordinate of Manhattan in New York City are 40.7896239, -73.9598939.
The geograpical coordinate of Brooklyn in New York City are 40.6501038, -73.9495823.
The geograpical coordinate of Queens in New York City are 40.6524927, -73.7914214158161.
The geograpical coordinate of Staten Island in New York City are 40.5834557, -74.1496048.
```

[67]:
```
Boroughs_latlong
```

[67]:

	Borough	Latitude	Longitude
0	Bronx	40.850485	-73.840404
1	Manhattan	40.789624	-73.959894
2	Brooklyn	40.650104	-73.949582
3	Queens	40.652493	-73.791421
4	Staten Island	40.583456	-74.149605

METHODOLOGY - WORKING ON THE PROJECT

5) Obtaining NYC coordinates.

OBTAINING NEW YORK CITY GEOGRAPHICAL COORDINATES

[32]:
```
address = 'New York City, NY'

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

```
The geograpical coordinate of New York City are 40.7127281, -74.0060152.
```

[37]:
```
cores = {'Bronx':'green', 'Manhattan':'red', 'Brooklyn':'indigo','Queens':'blue', 'Staten Island': 'orange'}
```

[88]:
```
# CREATING A DATAFRAME
coordenadas = {'Lat': [latitude_nyc], 'Lng': [longitude_nyc]}
nyc_latlong = pd.DataFrame(coordenadas)
nyc_latlong
```

[88]:

	Lat	Lng
0	40.712728	-74.006015

SAVING THE FILE FOR FUTURE USE (DO NOT USE UNNECESSARILY THE GEOLOCATOR)

[126]:
```
#SAVIND THE FILE FOR FUTURE USE
#nyc_latlong.to_csv('nyc_latlong.csv')
```

METHODOLOGY - WORKING ON THE PROJECT

6) Creating a Function to search the venues on Foursquare (API), based on query “CrossFit”. This function was adapted to treat possible errors of connection, errors on "JSON" file and errors “List/append" and “Pandas/DataFrame” commands, and save these errors in files to use afterward.

FUNCTION TO SEARCH ALL NYC NEIGHBORHOODS AT ONE TIME

[49]:

```
# PESQUISA TODOS OS BAIRROS DE UMA VEZ - ARQUIVO ESTÁ INTERNO

def getNearbyVenues_all(query, radius, limit):

    search_query = query
    search_radius = radius
    search_limit = limit

    nearby_venues = []
    nearby_erros = []
    venues_list = []
    erros_get_list = []

    count = 0
    count_erro_get = 0
    count_erro_append = 0
    count_erro_dataframe=0
    count_venues = 0

    #print('------- INICIO DA LEITURA DO ARQUIVO -------')
    #print(' ')

    # PARÂMETRO PARA A PESQUISA
    Neighborhoods_to_search = neighborhoods_all[:]

    for index, row in Neighborhoods_to_search[:].iterrows():
        count = count+1
```

METHODOLOGY - WORKING ON THE PROJECT

7) Calling the function and passing the parameters.

CALL FUNCTION TO SEARCH ALL SAME TIME WITH ERROR TREATMENT AND FILE GENERATION

[50]:
```
# CHAMADA DA FUNÇÃO DA FUNÇÃO, UMA VEZ PARA CADA BOROUGH, POR CONTA DOS ERROS NO FOURSQUARE - RADIUS 3000m

search_venues_results, search_erros = getNearbyVenues_all(query='crossfit', radius=3000, limit=100)
```

```
Bronx ,  Wakefield ,  lat: 40.89470517661004  - Lng: -73.8472005205491  -->  1
Bronx ,  Co-op City ,  lat: 40.87429419303015  - Lng: -73.82993910812405  -->  2
Bronx ,  Eastchester ,  lat: 40.88755567735082  - Lng: -73.82780644716419  -->  3
Bronx ,  Fieldston ,  lat: 40.895437426903875  - Lng: -73.90564259591689  -->  4
Bronx ,  Riverdale ,  lat: 40.89083449389134  - Lng: -73.91258546108577  -->  5
Bronx ,  Kingsbridge ,  lat: 40.88168737120525  - Lng: -73.90281798724611  -->  6
Manhattan ,  Marble Hill ,  lat: 40.87655077879968  - Lng: -73.91065965862988  -->  7
Bronx ,  Woodlawn ,  lat: 40.898272612138086  - Lng: -73.86731496814183  -->  8
Bronx ,  Norwood ,  lat: 40.877224155994504  - Lng: -73.87939073956817  -->  9
Bronx ,  Williamsbridge ,  lat: 40.88103887819214  - Lng: -73.85744642974214  -->  10
Bronx ,  Baychester ,  lat: 40.86685810725274  - Lng: -73.83579759808124  -->  11
Bronx ,  Pelham Parkway ,  lat: 40.85741349808869  - Lng: -73.85475564018006  -->  12
Bronx ,  City Island ,  lat: 40.84724670491817  - Lng: -73.7864884526742  -->  13
Bronx ,  Bedford Park ,  lat: 40.87018516497537  - Lng: -73.88551218419137  -->  14
Bronx ,  University Heights ,  lat: 40.85572707719668  - Lng: -73.91041596191317  -->  15
Bronx ,  Morris Heights ,  lat: 40.84789792606274  - Lng: -73.91967159119572  -->  16
Bronx ,  Fordham ,  lat: 40.86099679638657  - Lng: -73.8964265598163  -->  17
Bronx ,  East Tremont ,  lat: 40.84269615786057  - Lng: -73.88735617532345  -->  18
Bronx ,  West Farms ,  lat: 40.83947505672657  - Lng: -73.87774474910552  -->  19
Bronx ,  High  Bridge ,  lat: 40.8366230107061  - Lng: -73.92610209358138  -->  20
Bronx ,  Melrose ,  lat: 40.81975437059498  - Lng: -73.90942160757443  -->  21
Bronx ,  Mott Haven ,  lat: 40.80623874935181  - Lng: -73.91609987487583  -->  22
Bronx ,  Port Morris ,  lat: 40.80166362775625  - Lng: -73.91322139386142  -->  23
```

METHODOLOGY - WORKING ON THE PROJECT

8) Saving and reading the file. Don’t use Foursquare API unnecessarilly!

SAVING THE ORIGINAL RESULTS FILE TO AVOID NEW RESEARCH

[55]:
```
# SAVING THE FILE
#neighborhoods_all_crossfit_results = search_venues_results[:]
#neighborhoods_all_crossfit_results.shape
```

[55]:
```
(1074, 13)
```

[184]:
```
# GRAVANDO O RESULTADO PARA USO POSTERIOR
# neighborhoods_all_crossfit_results.to_csv('neighborhoods_all_crossfit_results.csv')
```

READING THE FILE WITH THE RESULTS AND CONTINUE ANALYSIS

[196]:
```
neighborhoods_all_crossfit_results = pd.read_csv('neighborhoods_all_crossfit_results.csv')
#neighborhoods_all_crossfit_results = neighborhoods_all_crossfit_results.drop(columns=['Unnamed: 0'])
```

[197]:
```
neighborhoods_all_crossfit_results.head(2)
```

[197]:

	Unnamed: 0	Borough	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Address	Distance	Postalcode	City	State	Venue Category
0	0	Bronx	Fieldston	40.895437	-73.905643	Spuyten Duyvil Crossfit	40.887847	-73.907061	3603 Fieldston Rd	853	10463.0	Bronx	NY	Gym / Fitness Center
1	1	Bronx	Riverdale	40.890834	-73.912585	Spuyten Duyvil Crossfit	40.887847	-73.907061	3603 Fieldston Rd	571	10463.0	Bronx	NY	Gym / Fitness Center

METHODOLOGY - WORKING ON THE PROJECT

9) Cleaning the file

LET'S DELETE ALL RECORDS THAT HAVE NOT 'CROSSFIT' IN THE NAME OF THE VENUE (ONLY OFFICIAL CROSSFIT GYMS CAN PUT CROSSFIT ON NAME)

[193]:
```
df_results = df_results[df_results['Venue'].str.contains('crossfit', case=False, regex=True)].reset_index(drop=True)
```

[200]:

	Unnamed: 0	Borough	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Address	Distance	Postalcode	City	State	Venue Category
0	0	Bronx	Fieldston	40.895437	-73.905643	Spuyten Duyvil Crossfit	40.887847	-73.907061	3603 Fieldston Rd	853	10463.0	Bronx	NY	Gym / Fitness Center
1	1	Bronx	Riverdale	40.890834	-73.912585	Spuyten Duyvil Crossfit	40.887847	-73.907061	3603 Fieldston Rd	571	10463.0	Bronx	NY	Gym / Fitness Center

DELETING DUPLICATE RECORDS, MAINTAINING THE LOWER 'DISTANCE' BECAUSE IT IS THE BEST WAY

[617]:
```
# o sort ordena e altera a porra do indice... já o sort_index(inplace=True) retorna ao que era antes...ufa!

df_results = df_results.sort_values('Distance').drop_duplicates(subset=['Venue','Venue Latitude','Venue Longitude'], keep="first")
df_results.sort_index(inplace=True)
```

SOME TRICKS FROM A OLD BYTES BRUSHER... PAY ATTENTION!

NOW LET'S ASSIGN 'BOROUGH' TO 'CITY' IF 'CITY' EQUAL 'NEIGHBORHOOD'

[622]:
```
# VERIFICAR SE EXISTE CITY = NEIGHBORHOOD
df_results.loc[df_results['City'] == df_results['Neighborhood']].shape
```

[622]:
```
(7, 14)
```

[623]:
```
# CITY = BOROUGH
df_results.loc[df_results['City'] == df_results['Neighborhood'],'City']= df_results['Borough']
```

METHODOLOGY - WORKING ON THE PROJECT

10) Continuing the cleaning...

NOW LET'S REPLACE THE VALUES OF THE FIELD CITY:

WHEN CITY=NEW YORK, REPLACE CITY=MANHATTAN IF BOROUGH=MANHATTAN.

PAY ATTENTION!!! MANHATTAN IS NOT APPEARING IN THE FIELD 'CITY' AND I AM USING THIS FIELD AS REFERENCE TO FIX ERRORS

[627]:
```
# ATRIBUIR A CITY O VALOR DE BOROUGH
count = 0
for row, linha in df_results.iterrows():

    if (df_results.loc[row,'City']=='New York') & (df_results.loc[row,'Borough']=='Manhattan'):

        df_results.loc[row,'City'] = df_results.loc[row,'Borough']
        count +=1

print('Total: ', count)
```

NOW LET'S READ THE NYC OFFICIAL FILE TO OBTAIN THE OFFICIALS NEIGHBORHOODS AND BOROUGHS.

AFTER, WE WILL CHECK IF THEY ARE EQUAL TO THE NEIGHBORHOODS OF THE RESULT FILE.

IF THEY ARE EQUAL, CITY FIELD WILL RECEIVE OFFICIAL BOROUGH

[630]:
```
df_all = pd.read_csv('Neighborhoods_all.csv')
```

[631]:
```
count = 0
for n in (df_results.loc[df_results.City.isin(df_all.Neighborhood)].index):

    for row, linha in df_all.iterrows():

        if df_results.loc[n,'City'] == df_all.loc[row]['Neighborhood']:
            print(df_all.loc[row]['Borough'],df_all.loc[row]['Neighborhood'],df_results.loc[n,'City'], n)
            df_results.loc[n,'City'] = df_all.loc[row]['Borough']
            count+=1


print('Total: ',count)
```

METHODOLOGY - WORKING ON THE PROJECT

11) Continuing the cleaning...

NOW WE WILL CHECK THAT SOME 'CITY', WHICH WE USE AS A REFERENCE, IS DIFFERENT FROM 'BOROUGH' '

[642]:
```
df_results[df_results.Borough != df_results.City]
```

[642]:

	Unnamed: 0	Borough	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Address	Distance	Postalcode	City	State	Ver Categ
19	19	Bronx	Mott Haven	40.806239	-73.916100	Crossfit Concrete Jungle	40.808122	-73.930234	7 Bruckner Blvd, bronx Ny 10454	1209	10454.0	New York	NY	Gy Fitn Cer
127	129	Brooklyn	Bedford Stuyvesant	40.687232	-73.941785	Crossfit Outbreak	40.686850	-73.941751	492 Throop Ave	42	11221.0	New York	NY	Gy Fitn Cer
385	421	Manhattan	Upper West Side	40.787658	-73.977059	CrossFit Elite Core Fitness (ECF)	40.791000	-74.008470	6500 Dewey Ave	2673	7093.0	West New York	NJ	Gy Fitn Cer

LET'S DELETE THE CITY THAT IS DIFFERENT FROM THE TRADITIONAL BOROUGH

[645]:
```
list_drop = df_results.loc[~df_results.City.isin(['Bronx', 'New York', 'Brooklyn', 'Manhattan', 'Queens','Staten Island'])].index
```

[646]:
```
df_results = df_results.drop(list_drop)
```

LET'S SEE IF SOME 'CITY' IS DIFFERENT FROM 'BOROUGH'

[648]:
```
df_results[df_results.Borough != df_results.City]
```

[648]:

	Unnamed: 0	Borough	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Address	Distance	Postalcode	City	State	Venue Category
19	19	Bronx	Mott Haven	40.806239	-73.916100	Crossfit Concrete Jungle	40.808122	-73.930234	7 Bruckner Blvd, bronx Ny 10454	1209	10454.0	New York	NY	Gym / Fitness Center

METHODOLOGY - WORKING ON THE PROJECT

12) Finally the file is cleaned.

THE FINAL FILE WITHOUT DUPLICATES, CLEAN AND FULLY ANALYZED IS ...

[651]:
```
df_results.shape
```

[651]:
```
(81, 14)
```

[652]:
```
df_results
```

[652]:

	Unnamed: 0	Borough	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Address	Distance	Postalcode	City	State	Venu Catego
1	1	Bronx	Riverdale	40.890834	-73.912585	Spuyten Duyvil Crossfit	40.887847	-73.907061	3603 Fieldston Rd	571	10463.0	Bronx	NY	Gym Fitne Cent
7	7	Bronx	Morris Heights	40.847898	-73.919672	Crossfit Uws	40.846545	-73.925147	NaN	484	NaN	Bronx	NY	Athleti & Spor
18	18	Bronx	Mott Haven	40.806239	-73.916100	CrossFit SoBro (South Bronx)	40.807106	-73.911477	229 Bruckner Blvd	401	10454.0	Bronx	NY	Gy
19	19	Bronx	Mott Haven	40.806239	-73.916100	Crossfit Concrete Jungle	40.808122	-73.930234	7 Bruckner Blvd, bronx Ny 10454	1209	10454.0	Bronx	NY	Gym Fitne Cent
45	45	Brooklyn	Bay Ridge	40.625801	-74.030621	Bay Ridge Crossfit	40.624143	-74.030823	3rd Avenue	185	11209.0	Brooklyn	NY	Gym Fitne Cent
46	46	Brooklyn	Bay Ridge	40.625801	-74.030621	bayridge crossfit	40.624227	-74.030727	NaN	175	NaN	Brooklyn	NY	Spor Clu
51	52	Brooklyn	Greenpoint	40.730201	-73.954241	Crossfit Greenpoint	40.736113	-73.951996	188 DuPont St	684	11222.0	Brooklyn	NY	Gym Fitne Cent
59	60	Brooklyn	Greenpoint	40.730201	-73.954241	willyb crossfit	40.729465	-73.950691	220 Newel Street	310	NaN	Brooklyn	NY	Gy

METHODOLOGY - WORKING ON THE PROJECT

13) Creating a Dataframe with total CrossFit gyms by Borough and Neighborhood.

CREATING A DATAFRAME WITH THE TOTAL CROSSFIT BY BOROUGHS, NEIGHBORHOODS

[466]:
```
analise_list = []

for borough in list_borough:
    borough_total = df_results[df_results.Borough==borough].shape[0]
    print(borough,'has: ', borough_total)

    lista_bairro = list(df_results[df_results.Borough==borough]['Neighborhood'].unique())

    for neighborhood in lista_bairro:
        neighborhood_total =  df_results[df_results.Neighborhood == neighborhood].shape[0]
        print('--->', neighborhood,'has:', neighborhood_total)

        analise_list.append([borough,borough_total, neighborhood, neighborhood_total])

    print('')

analise_with_cross = pd.DataFrame(analise_list)
analise_with_cross.columns = ['Borough','B_total', 'Neighborhood', 'N_total']
print('')
print('Dataframe "analise_with_cross" created!')
```

```
Bronx has:  5
---> Riverdale has: 1
---> Morris Heights has: 1
---> Mott Haven has: 2
---> Concourse Village has: 1

Brooklyn has:  31
---> Bay Ridge has: 2
---> Greenpoint has: 3
---> Gravesend has: 1
---> Windsor Terrace has: 1
---> Prospect Heights has: 2
```

METHODOLOGY - WORKING ON THE PROJECT

14) Creating a file with Statistics about NYC and adding the totals of CrossFit gyms.

NOW I WILL GET ON INTERNET SOME INFORMATION ABOUT NYC

GETTING STATISTICAL DATA ON NEW YORK CITY FROM WIKIPEDIA

[484]:
```
# GET THE PAGE
page = urllib.request.urlopen("https://en.wikipedia.org/wiki/Demographics_of_New_York_City")

# USE BEAUTIFUL SOUP
soup = BeautifulSoup(page, 'html.parser')

# THAT RETURNS A LIST.. YOU NEED TO SELECT THE FIRST TABLE [0]... THERE ARE FIVE!
table = soup.find_all('table')[0]

# READ_HTML... THAT RETURNS A LIST... YOU NEED TO SELECT THE FIRST [0] OR YOU HAVE A PROBLEM
nyc_statistics = pd.read_html(str(table))[0]
```

COPYING COORDENATES FROM FILE BOROUGHS_LATLONG, PREVIOUSLY SAVED WITH COORDENATES OF THE BOROUGHS

NOW I WILL ADD THE RESULTS OF DF_ANALISE TO THIS DF

[501]:

	Borough	Population	Billions	Per_capita	Square_mi	Square_km	Person_mi	Person_km	Latitude	Longitude	Crossfits
0	Bronx	1471160	28.787	19570.0	42.10	109.04	34653.0	13231.0	40.850485	-73.840404	5
1	Brooklyn	2648771	63.303	23900.0	70.82	183.42	37137.0	14649.0	40.650104	-73.949582	31
2	Manhattan	1664727	629.682	378250.0	22.83	59.13	72033.0	27826.0	40.789624	-73.959894	23
3	Queens	2358582	73.842	31310.0	108.53	281.09	21460.0	8354.0	40.652493	-73.791421	18
4	Staten Island	479458	11.249	23460.0	58.37	151.18	8112.0	3132.0	40.583456	-74.149605	4

METHODOLOGY - WORKING ON THE PROJECT

15) Normalizing the Dataframe because there are different magnitudes involved, by myself method.

I WILL NORMALIZE THE DATA BECAUSE THERE ARE DIFFERENT MAGNITUDES INVOLVED

[504]:
```
df_analise = df[['Population', 'Billions', 'Per_capita', 'Square_mi', 'Person_mi', 'Crossfits']]
```

[505]:
```
df_analise
```

[505]:

	Population	Billions	Per_capita	Square_mi	Person_mi	Crossfits
0	1471160	28.787	19570.0	42.10	34653.0	5
1	2648771	63.303	23900.0	70.82	37137.0	31
2	1664727	629.682	378250.0	22.83	72033.0	23
3	2358582	73.842	31310.0	108.53	21460.0	18
4	479458	11.249	23460.0	58.37	8112.0	4

NORMALIZE BY MYSELF

[506]:
```
df1 = df_analise/df_analise.max().astype(np.float64)
```

[507]:
```
df1
```

[507]:

	Population	Billions	Per_capita	Square_mi	Person_mi	Crossfits
0	0.555412	0.045717	0.051738	0.387911	0.481071	0.161290
1	1.000000	0.100532	0.063186	0.652538	0.515555	1.000000
2	0.628490	1.000000	1.000000	0.210357	1.000000	0.741935
3	0.890444	0.117269	0.082776	1.000000	0.297919	0.580645
4	0.181011	0.017865	0.062022	0.537824	0.112615	0.129032

METHODOLOGY - WORKING ON THE PROJECT

16) Normalizing the Dataframe because there are different magnitudes involved, by Scikit-learn (SkLearn), and putting the results on new Dataframe.

NORMALIZE BY SKLEARN

[508]:
```
from sklearn import preprocessing
x = df_analise.values
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
df2 = pd.DataFrame(x_scaled)
```

[509]:
```
df2
```

[509]:

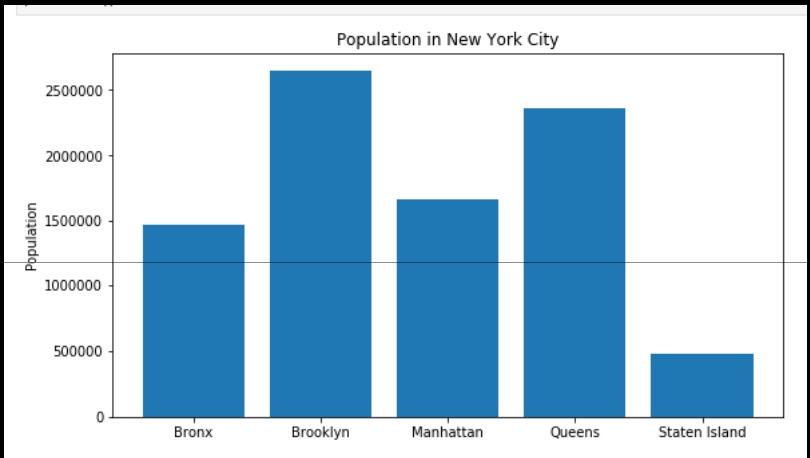
	0	1	2	3	4	5
0	0.45715	0.028359	0.000000	0.224854	0.415216	0.037037
1	1.00000	0.084171	0.012072	0.559977	0.454076	1.000000
2	0.54638	1.000000	1.000000	0.000000	1.000000	0.703704
3	0.86623	0.101212	0.032731	1.000000	0.208820	0.518519
4	0.00000	0.000000	0.010845	0.414702	0.000000	0.000000

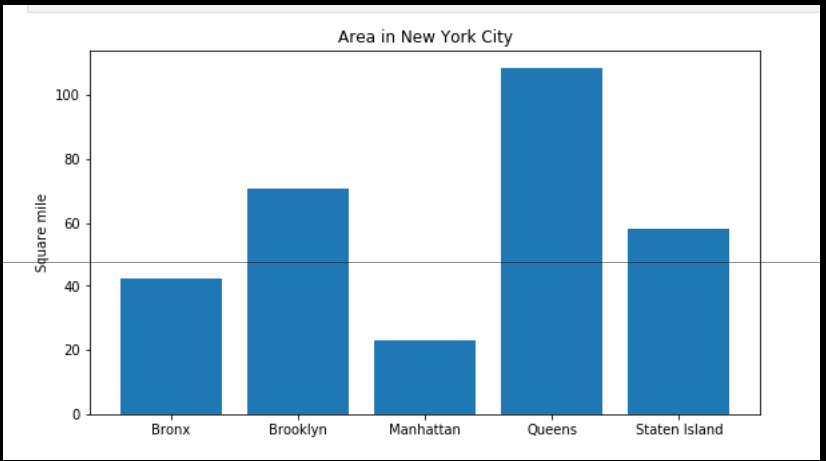
[522]:

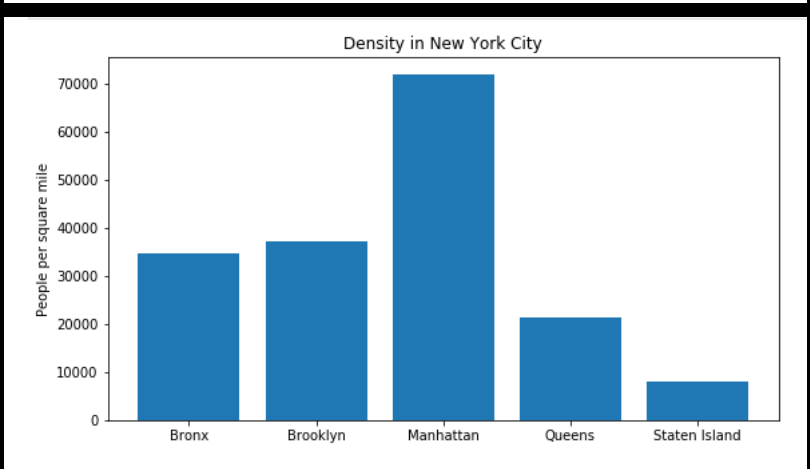
	Borough	Population	Billions	Per_capita	Square_mi	Square_km	Person_mi	Person_km	Latitude	Longitude	Crossfits	Score_my	Score_sk
0	Bronx	1471160	28.787	19570.0	42.10	109.04	34653.0	13231.0	40.850485	-73.840404	5	1.683140	1.162616
1	Brooklyn	2648771	63.303	23900.0	70.82	183.42	37137.0	14649.0	40.650104	-73.949582	31	3.331811	3.110296
2	Manhattan	1664727	629.682	378250.0	22.83	59.13	72033.0	27826.0	40.789624	-73.959894	23	4.580782	4.250084
3	Queens	2358582	73.842	31310.0	108.53	281.09	21460.0	8354.0	40.652493	-73.791421	18	2.969053	2.727512
4	Staten Island	479458	11.249	23460.0	58.37	151.18	8112.0	3132.0	40.583456	-74.149605	4	1.040369	0.425548

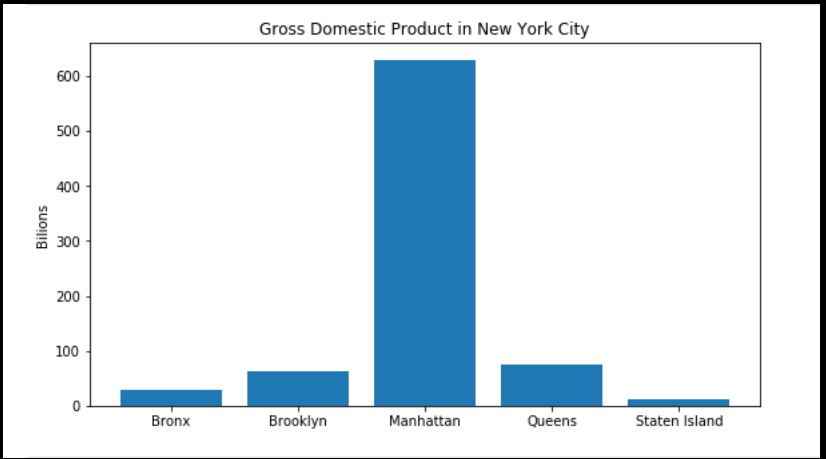
METHODOLOGY - WORKING ON THE PROJECT

17) This part of the Project is completed. The Best Borough to open a CrossFit gym is Manhattan. Let’s see some graphics about the Data.



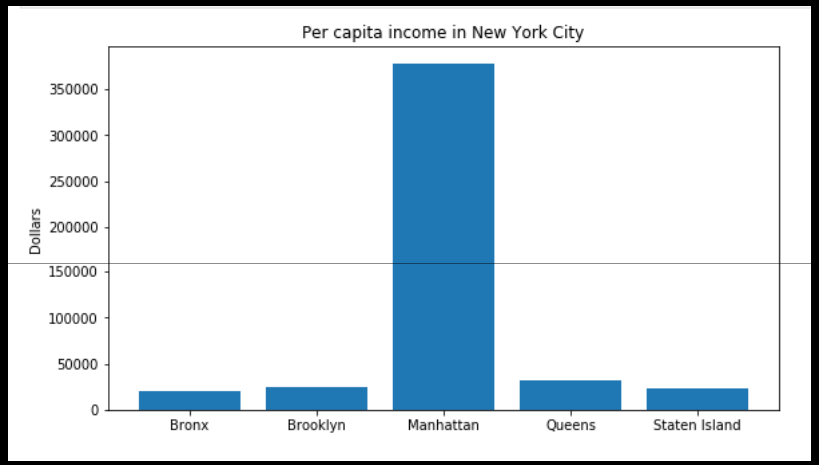


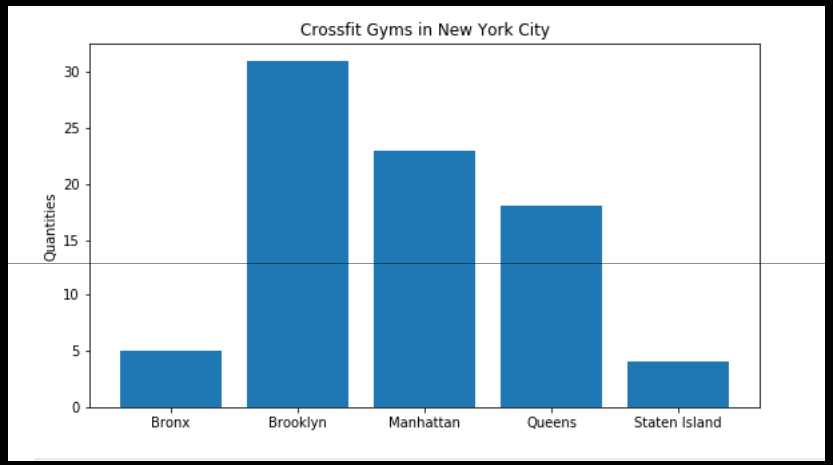


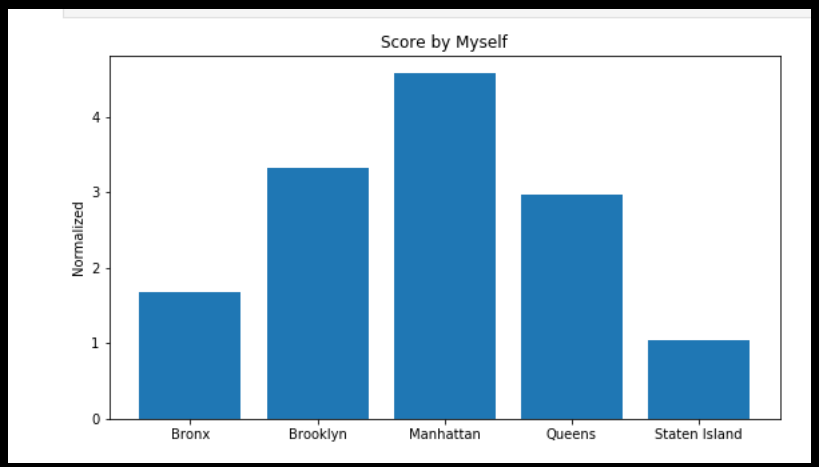


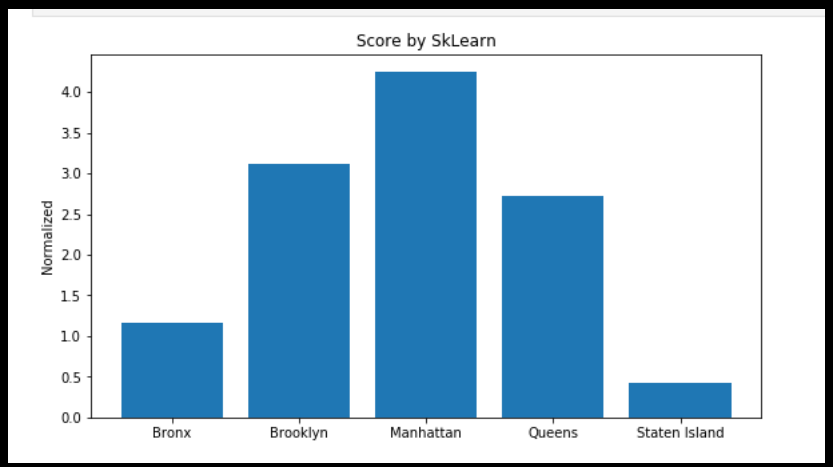
METHODOLOGY - WORKING ON THE PROJECT

18) This part of the Project is completed. The Best Borough to open a CrossFit gym is MANHATTAN. Let's see some graphics about the Data.



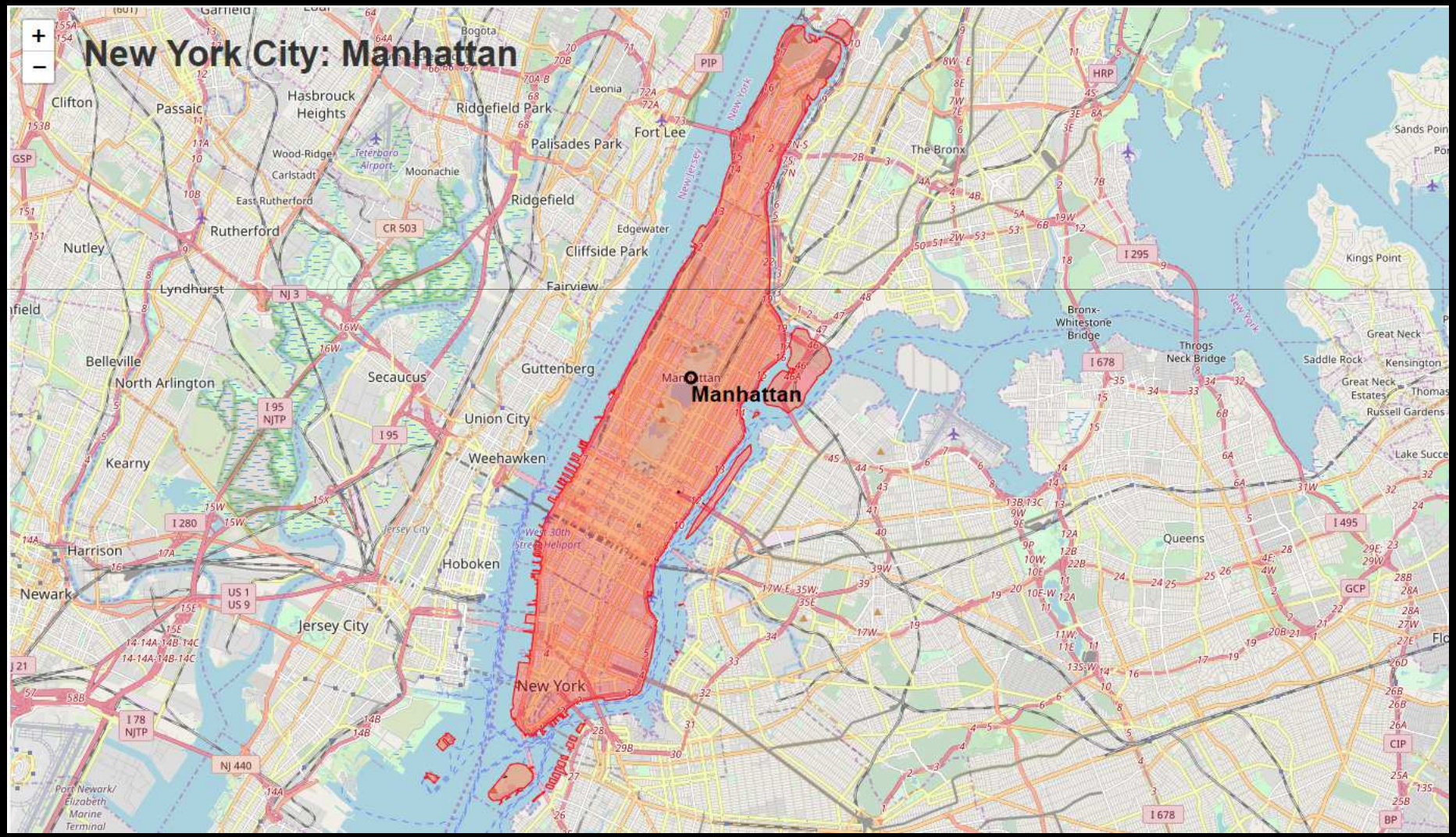






METHODOLOGY - WORKING ON THE PROJECT

19) MANHATTAN has the best score after normalized the different magnitudes involved. It is not a surprise because there are the biggest density and “per capita” numbers.



METHODOLOGY - WORKING ON THE PROJECT

19) Analyzing the files and discovering which Neighborhoods have and which don't have any CrossFit gyms.

SHOWING THE NUMBERS ...

```
# SHOWING THE NUMBER...
total_neigh = len(list_mc)
total = df_manhattan_neighborhoods.shape[0]
total_crossfit = df_manhattan_crossfits.shape[0]

# DISCOVERING WHICH PLACES THERE IS NOT CROSSFIT
df_manhattan_without = df_manhattan_neighborhoods.loc[~df_manhattan_neighborhoods.Neighborhood.isin(list_mc)]
total_w = df_manhattan_without.shape[0]
list_no_crossfit = list(df_manhattan_without.Neighborhood)
total_no_crossfit = len(list_no_crossfit)

print('\n\nPAY ATTENTION!')
print('There are {} neighborhoods in Manhattan, according the official NYC database'.format(total))
print('There are {} neighborhoods in Manhattan with Crossfit, according Foursquare'.format(total_neigh))
print('There are {} Crossfit Gyms in Neighboorhoods of Manhattan, according Foursquare.'.format(total_crossfit))
print('There are {} neighborhoods in Manhattan without Crossfit Gym, according Foursquare\n\n'.format(total_w))
```

```
PAY ATTENTION!
There are 39 neighborhoods in Manhattan, according the official NYC database
There are 13 neighborhoods in Manhattan with Crossfit, according Foursquare
There are 23 Crossfit Gyms in Neighboorhoods of Manhattan, according Foursquare.
There are 26 neighborhoods in Manhattan without Crossfit Gym, according Foursquare
```

THESE NEIGHBORHOOD HAVE CROSSFIT GYMS (ACCORDING TO FOURSQUARE):

'Lincoln Square', 'Financial District', 'Tribeca', 'Central Harlem', 'Civic Center', 'Chinatown', 'East Harlem', 'Clinton', 'Murray Hill', 'East Village', 'Carnegie Hill', 'Midtown South', 'Flatiron'

AND THESE NEIGHBORHOOD DON'T HAVE CROSSFIT GYMS (ACCORDING TO FOURSQUARE):

'Marble Hill','Washington Heights', 'Inwood', 'Hamilton Heights', 'Manhattanville', 'Upper East Side', 'Yorkville', 'Lenox Hill', 'Roosevelt Island', 'Upper West Side', 'Midtown', 'Chelsea', 'Greenwich Village', 'Lower East Side', 'Little Italy', 'Soho', 'West Village', 'Manhattan Valley', 'Morningside Heights', 'Gramercy', 'Battery Park City', 'Noho', 'Sutton Place', 'Turtle Bay', 'Tudor City', 'Stuyvesant Town'

METHODOLOGY - WORKING ON THE PROJECT

20) Getting more information on Internet about the Neighborhoods of Manhattan.

LETS GO AND GET SOME INFORMATION ON INTERNET

```
### GETTING STATISTICAL DATA ON NEW YORK CITY FROM WORLDATLAS.COM
```

```
# GET THE PAGE
page = urllib.request.urlopen("https://www.worldatlas.com/articles/manhattan-neighborhoods-by-population.html")

# USE BEAUTIFUL SOUP
soup = BeautifulSoup(page, 'html.parser')

# THAT RETURNS A LIST.. YOU NEED TO SELECT THE FIRST TABLE [0]... THERE ARE FIVE!
table = soup.find_all('table')[0]

# READ_HTML... THAT RETURNS A LIST... YOU NEED TO SELECT THE FIRST [0] OR YOU HAVE A PROBLEM
neighborhoods_statistics = pd.read_html(str(table))[0]
```

```
# FIXING PROBLEM WITH HIDDEN CHARACTERES
neighborhoods_statistics.columns = ['Rank', 'Neighborhood', 'Population']

# SHOWING THE FILE
neighborhoods_statistics.head(10)
```

	Rank	Neighborhood	Population
0	1	Midtown	391371
1	2	Lower Manhattan	382654
2	3	Harlem	335109
3	4	Upper East Side	229688
4	5	Upper West Side	209084
5	6	Washington Heights	158318
6	7	East Harlem	115921
7	8	Chinatown	100000
8	9	Lower East Village	72957
9	10	Alphabet City	63347

COMPARING THE WORLDATLAS DATA WITH OUR LIST OF NEIGBORHOOD WITHOUT CROSSFIT GYMS

METHODOLOGY - WORKING ON THE PROJECT

21) Comparing the Data.

COMPARING THE WORLDATLAS DATA WITH OUR LIST OF NEIGBORHOOD WITHOUT CROSSFIT GYMS

[143]:
```
df = neighborhoods_statistics.loc[neighborhoods_statistics.Neighborhood.isin(list_no_crossfit)]
```

[144]:
```
df.head()
```

[144]:

	Rank	Neighborhood	Population
0	1	Midtown	391371
3	4	Upper East Side	229688
4	5	Upper West Side	209084
5	6	Washington Heights	158318
13	14	Morningside Heights	55929

THIS WEBSITE SAYS THAT MIDTOWN IS THE MOST POPULOUS NEIGHBORHOOD OF MANHATTAN

BUT IS IT TRUE?

[ ]:

GETTING SOME INFORAMTION ABOUT NYC ON HEALTH.NY.GOV ¶

[153]:
```
# GET THE PAGE
page = urllib.request.urlopen("https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoodpop.htm")

# USE BEAUTIFUL SOUP
soup = BeautifulSoup(page, 'html.parser')

# THAT RETURNS A LIST.. YOU NEED TO SELECT THE FIRST TABLE [0]... THERE ARE FIVE!
table = soup.find_all('table')[0]

health = pd.read_html(str(table))[0]
```

[180]:
```
wiki_health = health.loc[28:37]
```

9	10	Alphabet City	63347

COMPARING THE WORLDATLAS DATA WITH OUR LIST OF NEIGBORHOOD WITHOUT CROSSFIT GYMS

METHODOLOGY - WORKING ON THE PROJECT

22) Showing the results after the comparison. According the analysis, the best Neighborhood is MIDTOWN. But, let´s check on the map (Folium).

[181]:

	Borough	region	Males	Females	Total Population
28	New York (Manhattan)	Washington Heights, Inwood & Marble Hill	97142	106275	203417
29	NaN	Hamilton Heights, Manhattanville & West Harlem	61481	68085	129566
30	NaN	Central Harlem	56270	65431	121701
31	NaN	East Harlem	56312	64124	120435
32	NaN	Upper East Side	102121	127056	229177
33	NaN	Upper West Side & West Side	93032	108808	201840
34	NaN	Chelsea, Clinton & Midtown Business District	77568	71985	149553
35	NaN	Murray Hill, Gramercy & Stuyvesant Town	71357	84491	155848
36	NaN	Chinatown & Lower East Side	81995	87276	169271
37	NaN	Battery Park City, Greenwich Village & Soho	75851	78330	154181

[182]:

```
# PRINT SOME INFORMATION

print('These neighborhoods are in Midtown region:')
print(wiki_health.loc[34]['region'],',',wiki_health.loc[35]['region'] )
print('and their population is:', int(wiki_health.loc[34]['Total Population']+wiki_health.loc[35]['Total Population']))
```

```
These neighborhoods are in Midtown region:
Chelsea, Clinton & Midtown Business District , Murray Hill, Gramercy & Stuyvesant Town
and their population is: 305401
```

[ ]:

[ ]:

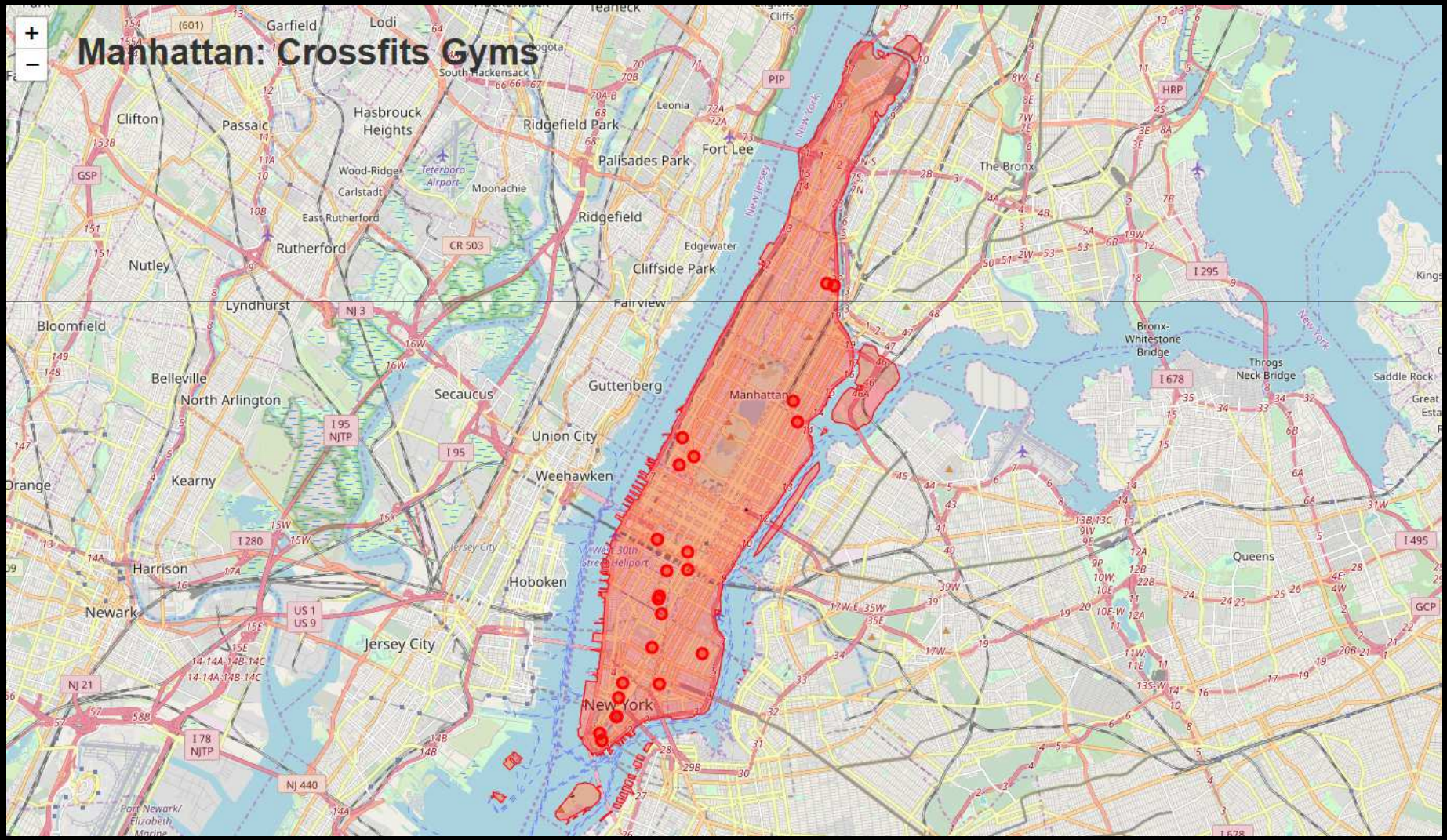
WELL, WE HAVE ANOTHER WINNER: MIDTOWN REGION! ¶

Midtown is the largest commercial, media, and entertainment center in the US. The neighborhood is the most populated in Manhattan, with a population of approximately 391,371 residents

[86]:

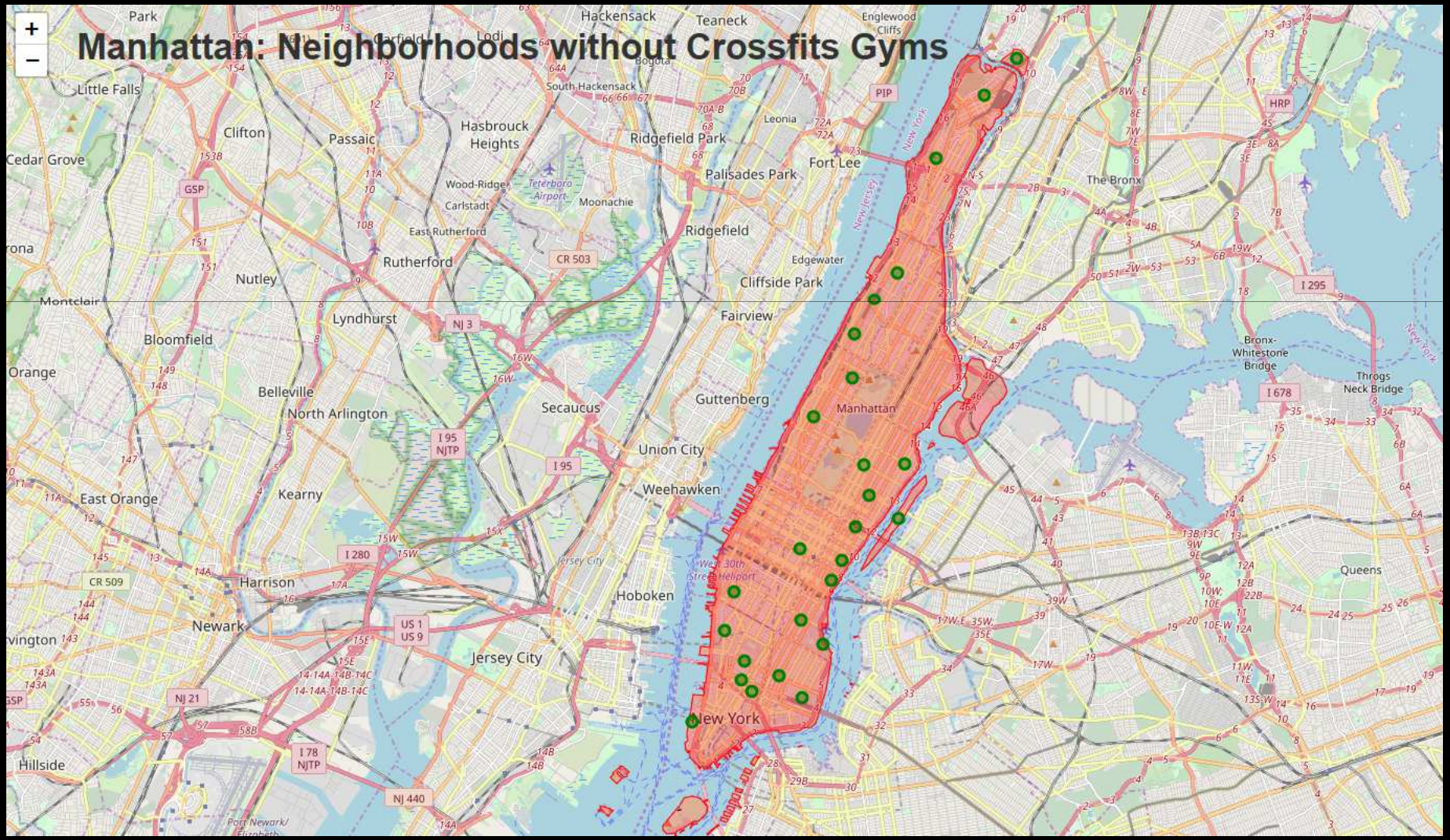
METHODOLOGY - WORKING ON THE PROJECT

23) There are 23 CrossFit gyms in Manhattan, in 13 neighborhoods, according to Foursquare.



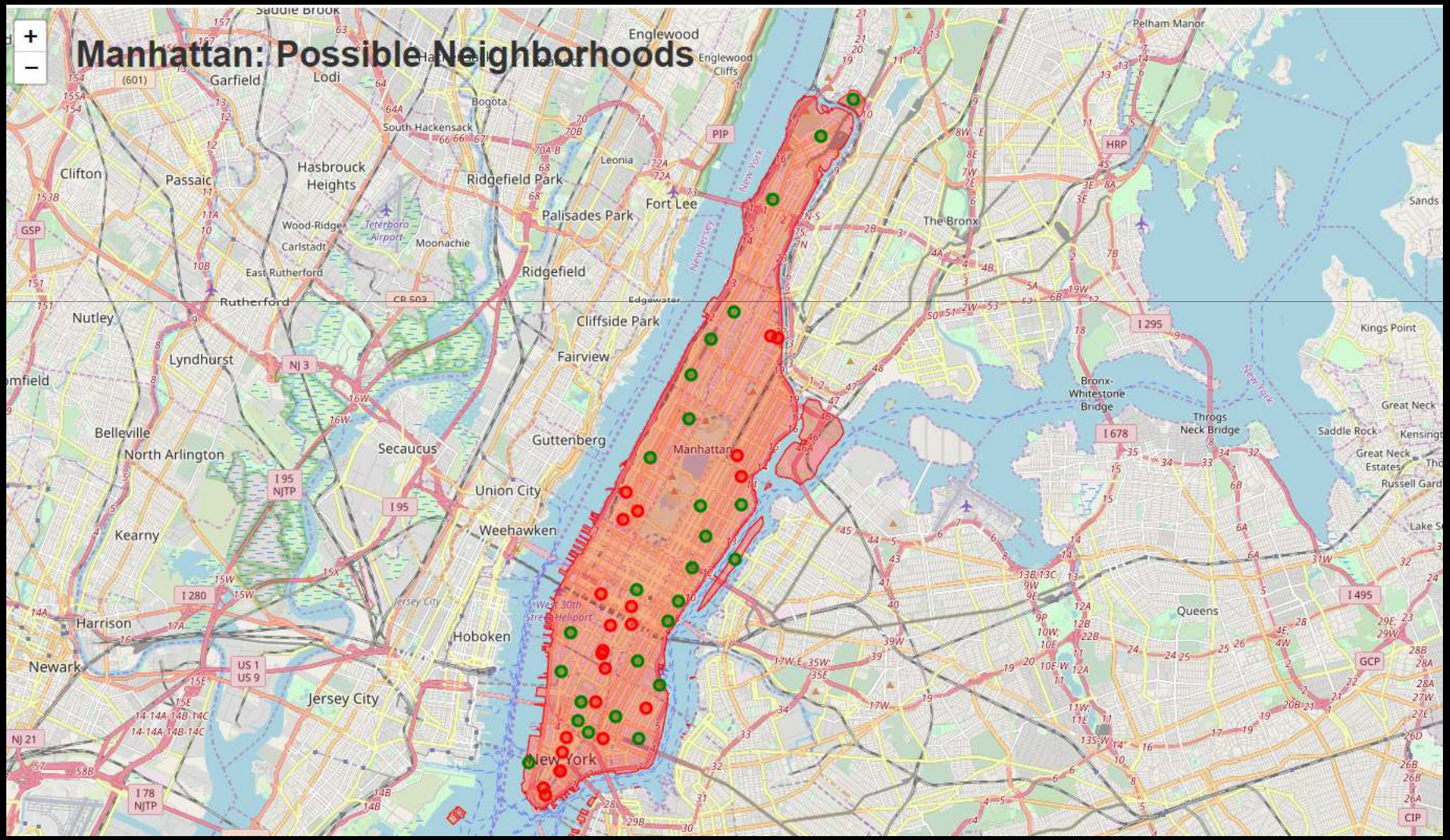
METHODOLOGY - WORKING ON THE PROJECT

24) There are no CrossFit gyms in 26 neighyborhoods of Manhattan.



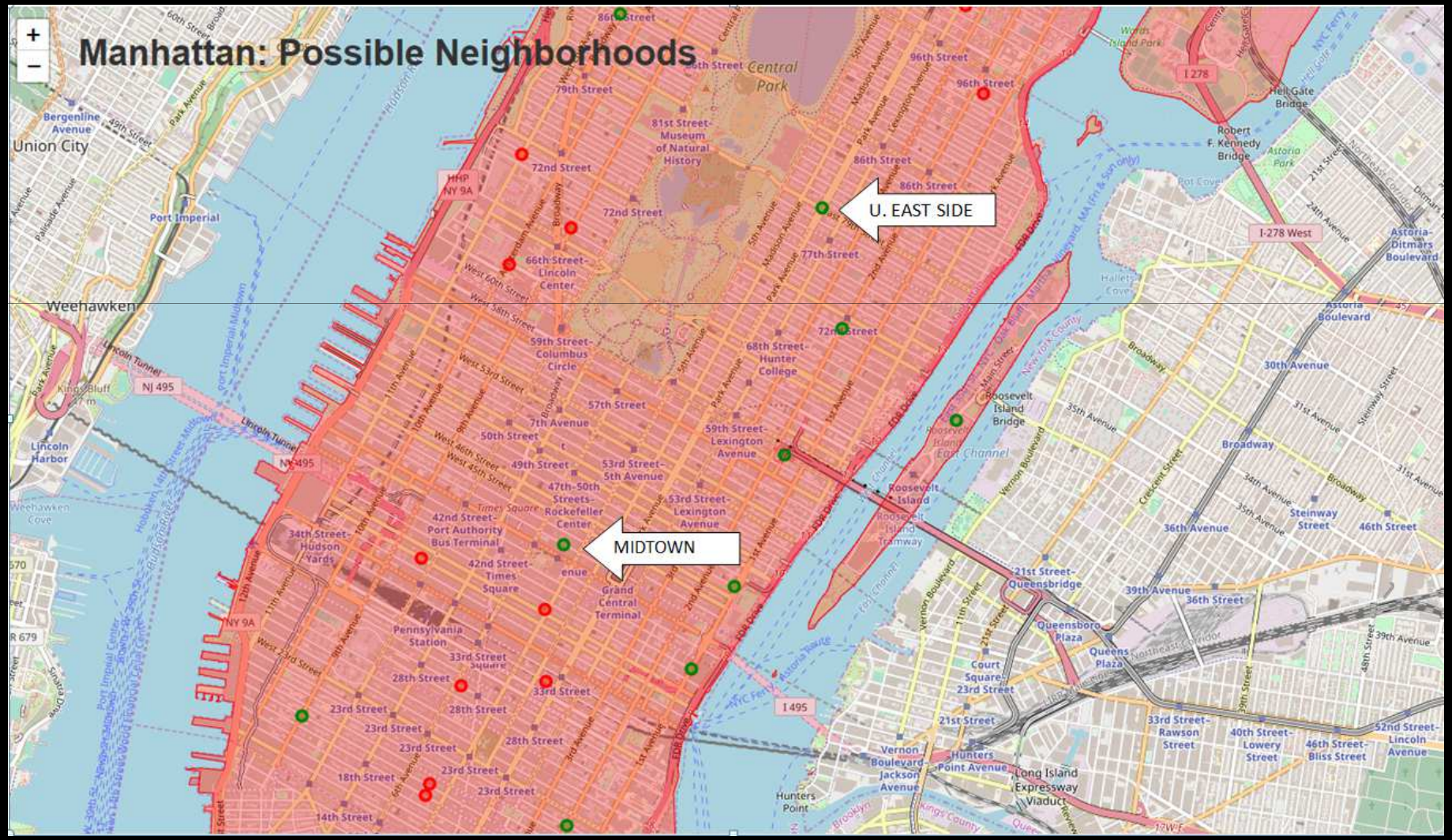
METHODOLOGY - WORKING ON THE PROJECT

25) Let’s see the both situations (with and without CrossFit gyms).



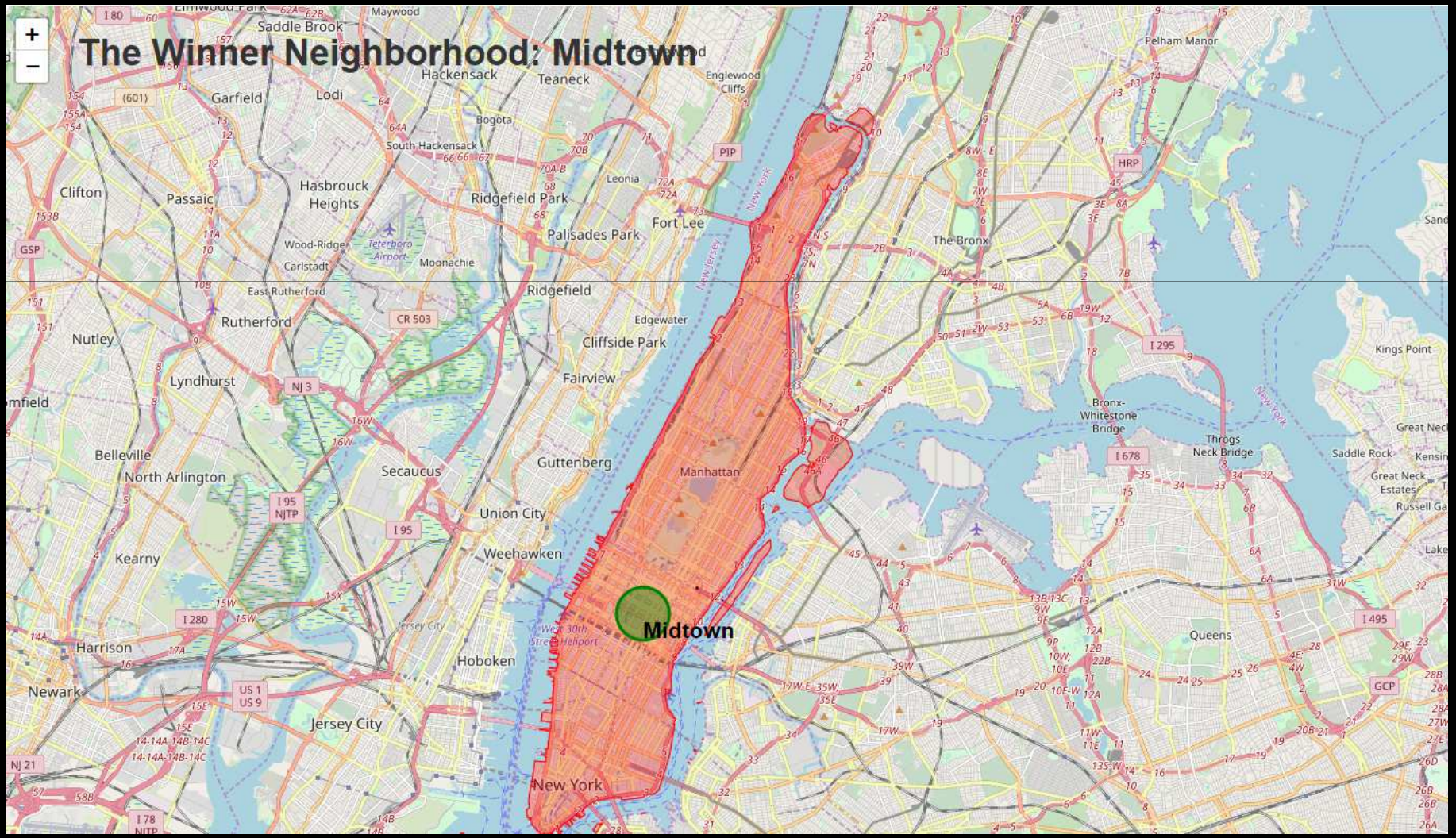
METHODOLOGY - WORKING ON THE PROJECT

26) These regions, together, have more than 500,000 people.



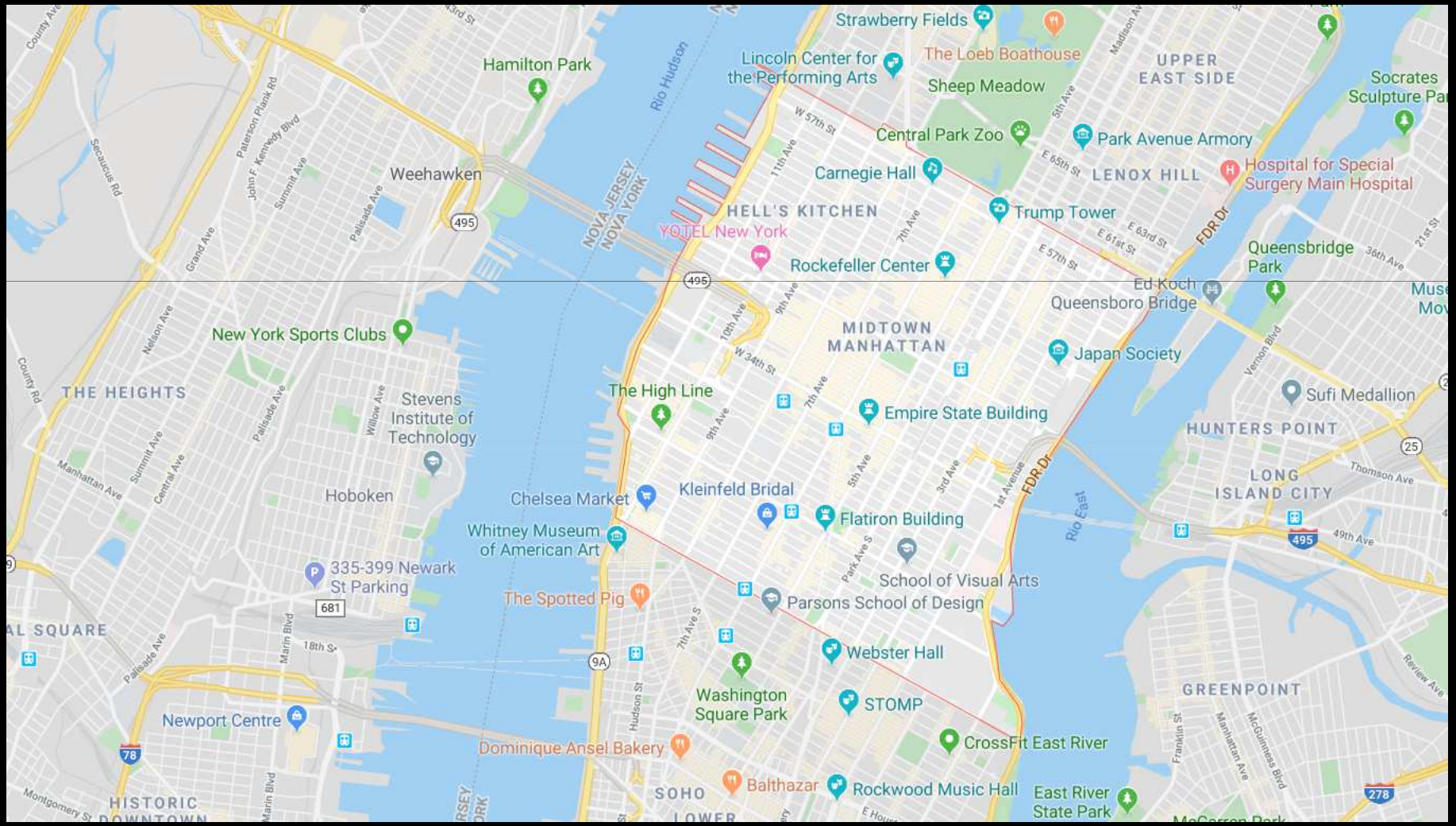
METHODOLOGY - WORKING ON THE PROJECT

27) Although it seems that Upper East Side is a good option as it is one farther from the gyms, the best option remains MIDTOWN as it is at the heart of one of MANHATTAN's richest and most populous regions, with the largest companies of the World.

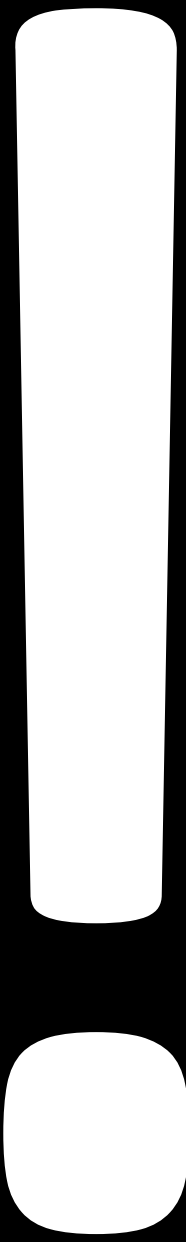


METHODOLOGY - WORKING ON THE PROJECT

28) MIDTOWN region.



THE BEST NEIGHBORHOOD OF NEW YORK CITY TO OPEN A CROSSFIT GYM IS MIDTOWN, IN MANHATTAN



EDUARDO VIEIRA - IBM- COURSERA - APPLIED DATA SCIENCE CAPSTONE - THE BATTLE OF NEIGHBORHOODS (WEEK 2)