Battle of Neighborhoods:

Business Problem:

Toronto is one of the largest cities and the financial hub in Canada and it is often compared with New York. Toronto is highly populated and open to new business always which interested my client in starting new physical training center. People are always about their health and fitness and in recent time, more people use physical training center. Even though there are more training centers than historical days, still the most of them are crowded. In this project we will explore the neighborhoods of Toronto, perform analysis and recommend a best neighborhood to start a new physical training center.

Data:

Geographical Data:

Using python goopy library to get the coordinated of the location.

Toronto Data:

Data Source: I will use the "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M" to identify the neighborhoods of Toronto for further analysis in this project.

Existing gym data:

Data Source: Foursquare API, which will provide us with all the existing venues of each neighborhood.

Analysis Methodology:

Scrape the data from the Wikipedia page.

Create a data frame with the scraped data defining the neighborhoods.

Using geopy library get the coordinates of each neighborhood

From foursquare API get the venues details of each neighborhood including category.

Slice the Physical/Training center data into a data frame

Business Questions:

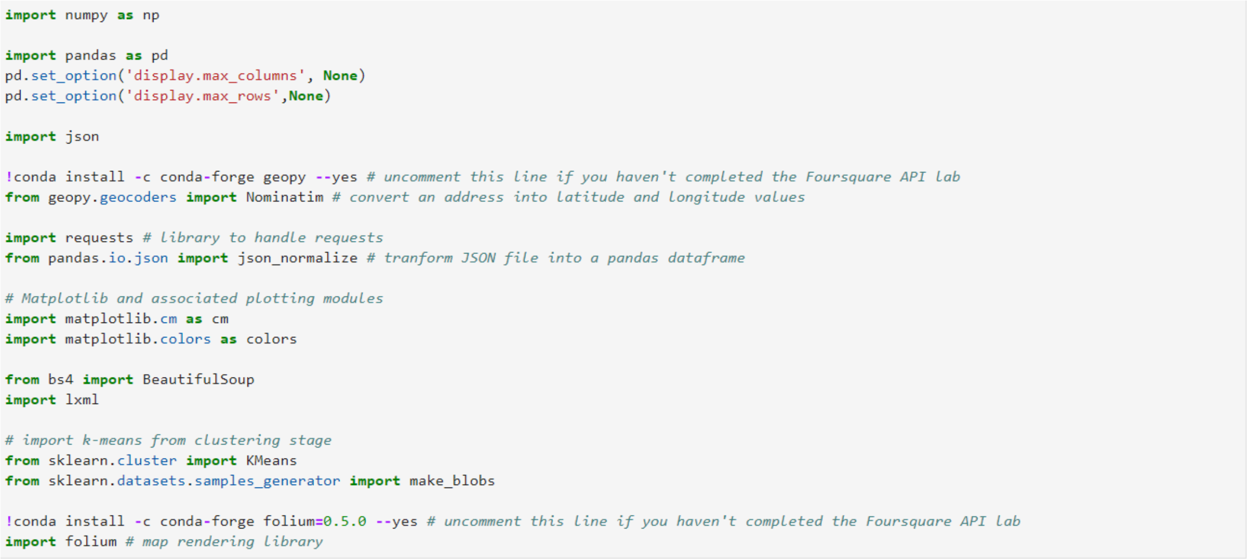
Which Borough has maximum number of neighborhood ?

What are the different types of venues available in the neighborhood ?

Which neighborhood has the highest number of Physical/ Training center ?

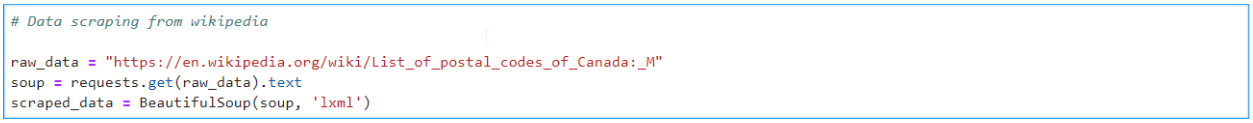
Analysis:

Import all the required python libraries.



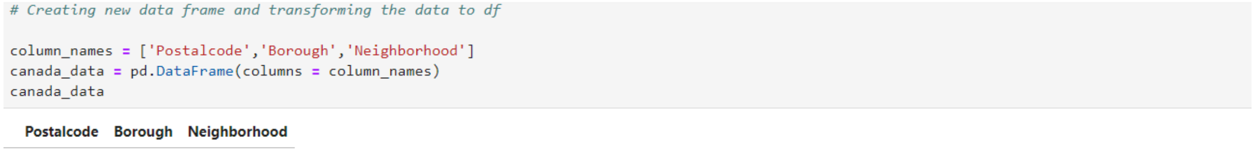
```
import numpy as np

import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows',None)

import json

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
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

from bs4 import BeautifulSoup
import lxml

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

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library
```

Scraping the data from wikipedia



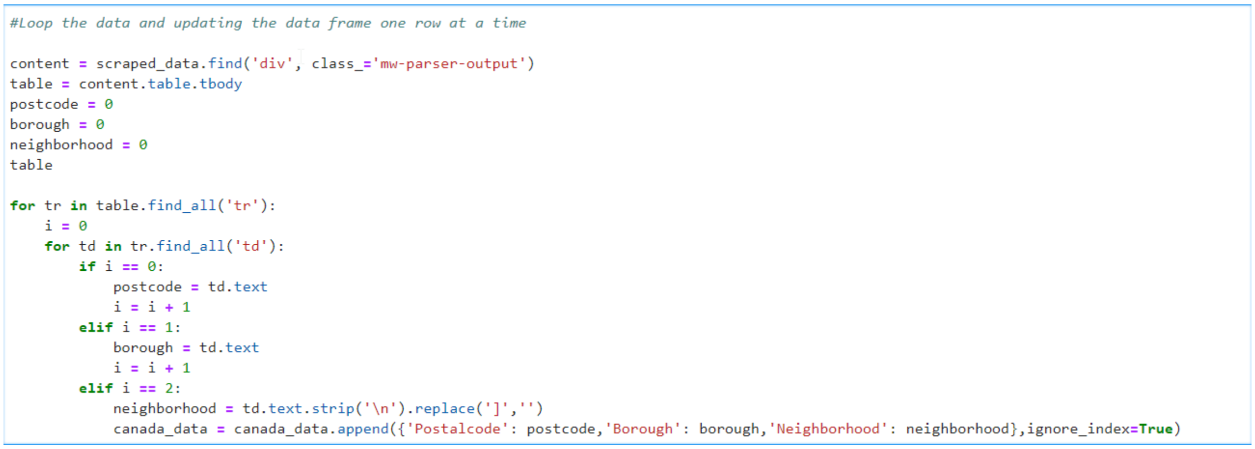
```
# Data scraping from wikipedia

raw_data = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
soup = requests.get(raw_data).text
scraped_data = BeautifulSoup(soup, 'lxml')
```

Creating a data frame and updating each row.



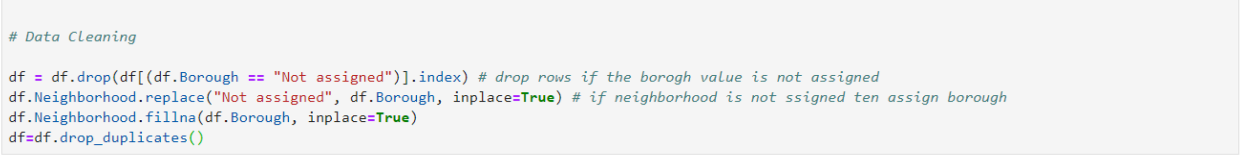
```
# Creating new data frame and transforming the data to df

column_names = ['Postalcode','Borough','Neighborhood']
canada_data = pd.DataFrame(columns = column_names)
canada_data
```

Postalcode	Borough	Neighborhood

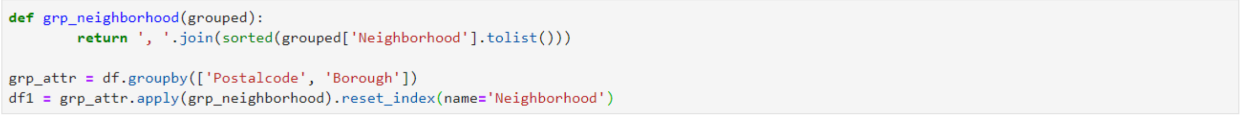


```
#Loop the data and updating the data frame one row at a time

content = scraped_data.find('div', class_='mw-parser-output')
table = content.table.tbody
postcode = 0
borough = 0
neighborhood = 0
table

for tr in table.find_all('tr'):
    i = 0
    for td in tr.find_all('td'):
        if i == 0:
            postcode = td.text
            i = i + 1
        elif i == 1:
            borough = td.text
            i = i + 1
        elif i == 2:
            neighborhood = td.text.strip('\n').replace(']','')
            canada_data = canada_data.append({'Postalcode': postcode,'Borough': borough,'Neighborhood': neighborhood},ignore_index=True)
```

Data Cleaning:

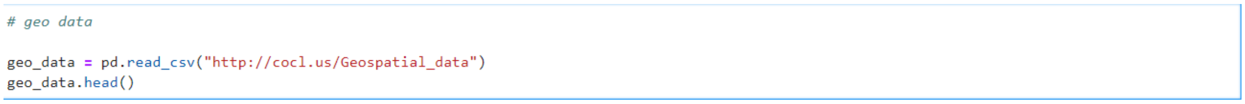


```
# Data Cleaning

df = df.drop(df[(df.Borough == "Not assigned")].index) # drop rows if the borogh value is not assigned
df.Neighborhood.replace("Not assigned", df.Borough, inplace=True) # if neighborhood is not ssigned ten assign borough
df.Neighborhood.fillna(df.Borough, inplace=True)
df=df.drop_duplicates()
```

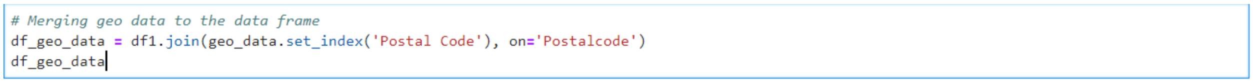


```
def grp_neighborhood(grouped):
        return ', '.join(sorted(grouped['Neighborhood'].tolist()))

grp_attr = df.groupby(['Postalcode', 'Borough'])
df1 = grp_attr.apply(grp_neighborhood).reset_index(name='Neighborhood')
```

Loading the geospatial data for each neighborhood

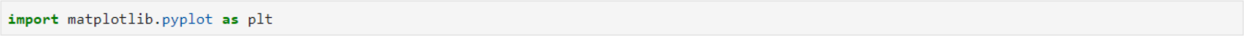


```
# geo data

geo_data = pd.read_csv("http://cocl.us/Geospatial_data")
geo_data.head()
```

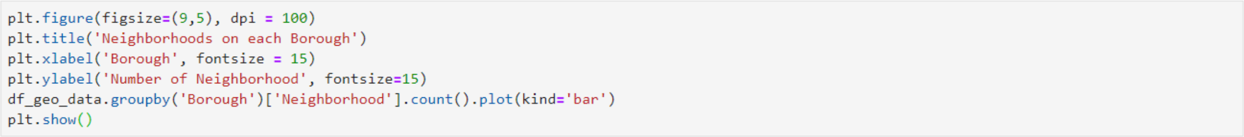


```
# Merging geo data to the data frame
df_geo_data = df1.join(geo_data.set_index('Postal Code'), on='Postalcode')
df_geo_data
```

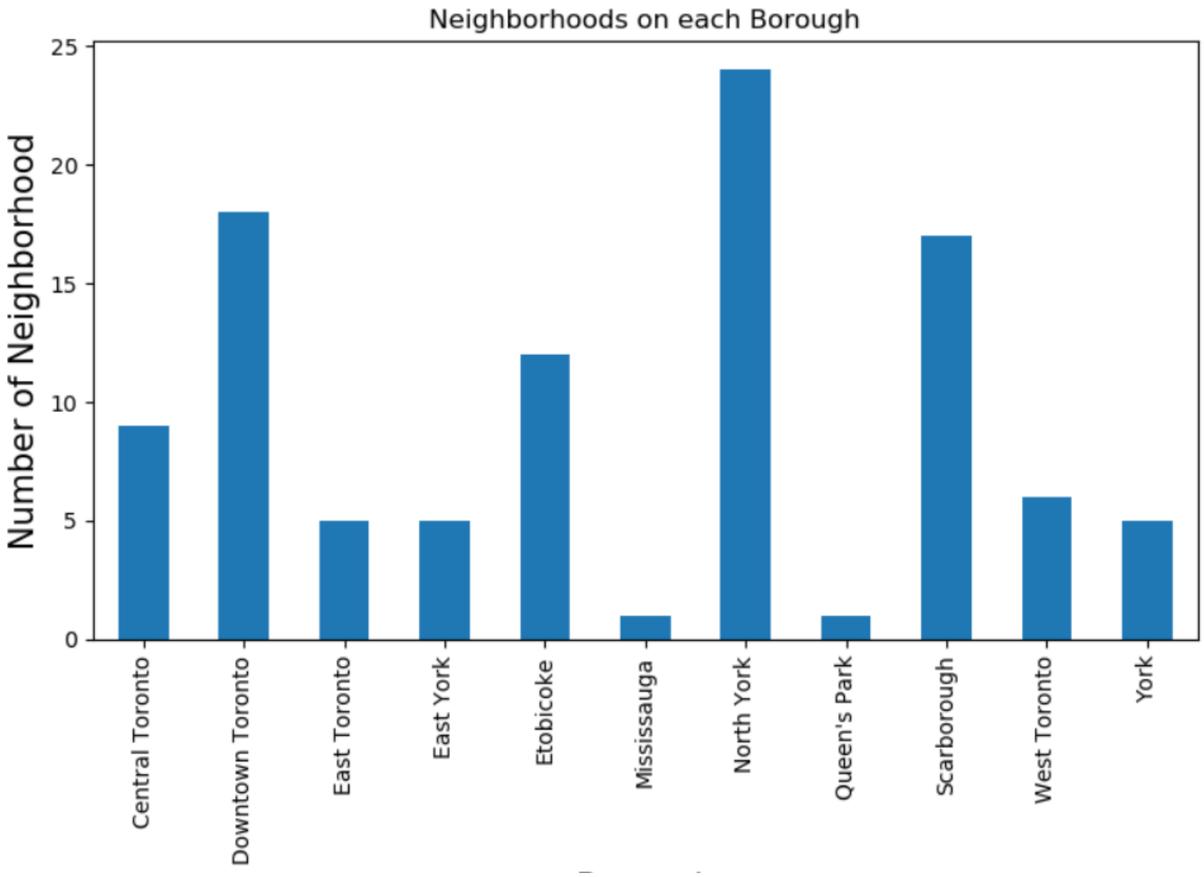
Visualizing the Boroughs



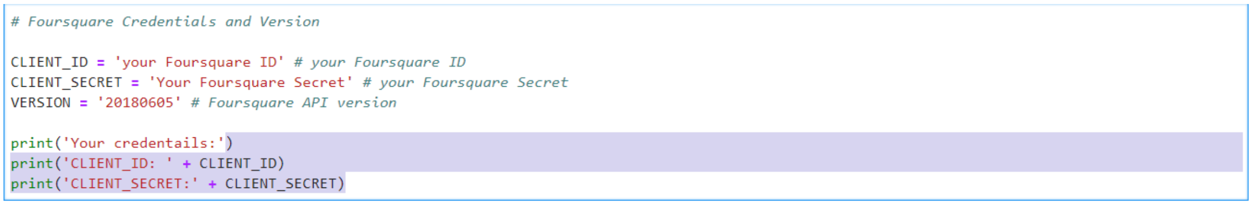
```
import matplotlib.pyplot as plt
```



```
plt.figure(figsize=(9,5), dpi = 100)
plt.title('Neighborhoods on each Borough')
plt.xlabel('Borough', fontsize = 15)
plt.ylabel('Number of Neighborhood', fontsize=15)
df_geo_data.groupby('Borough')['Neighborhood'].count().plot(kind='bar')
plt.show()
```

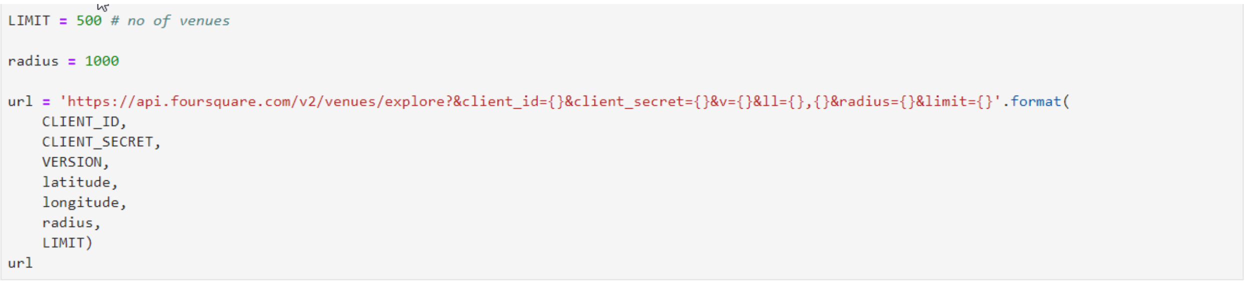


Getting data from fourquare



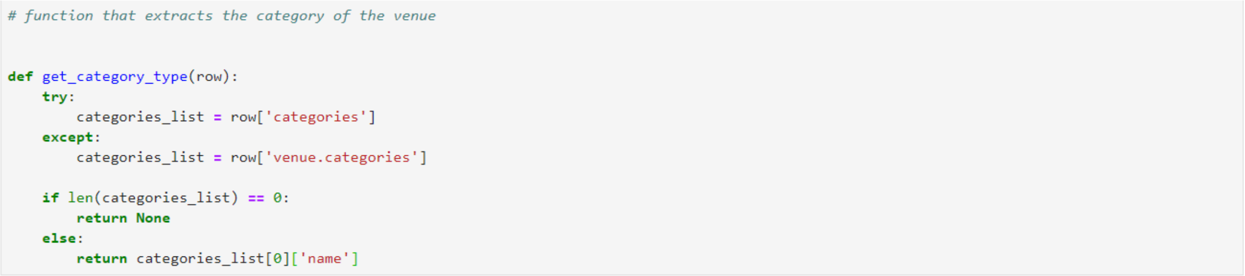
```
# Foursquare Credentials and Version

CLIENT_ID = 'your Foursquare ID' # your Foursquare ID
CLIENT_SECRET = 'Your Foursquare Secret' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

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

Extracting the venues details from foursquare

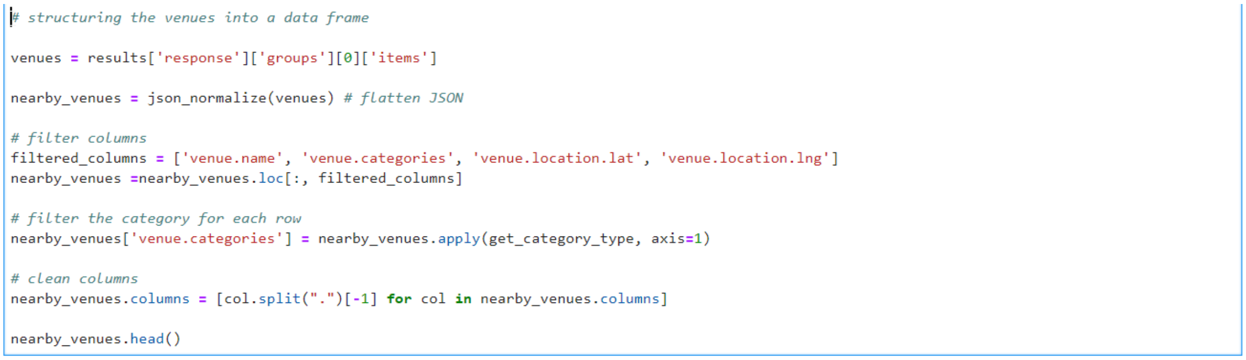


```
LIMIT = 500 # no of venues

radius = 1000

url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    latitude,
    longitude,
    radius,
    LIMIT)
url
```

Identifying the venue category and loading into a data frame

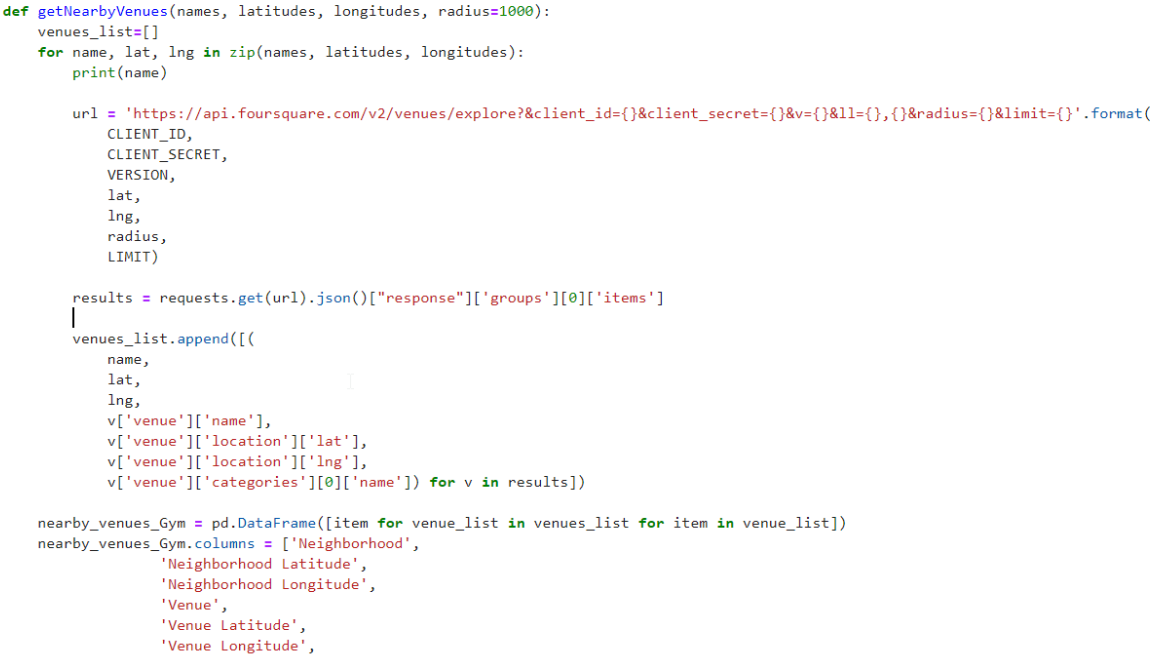


```
# function that extracts the category of the venue


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']
```

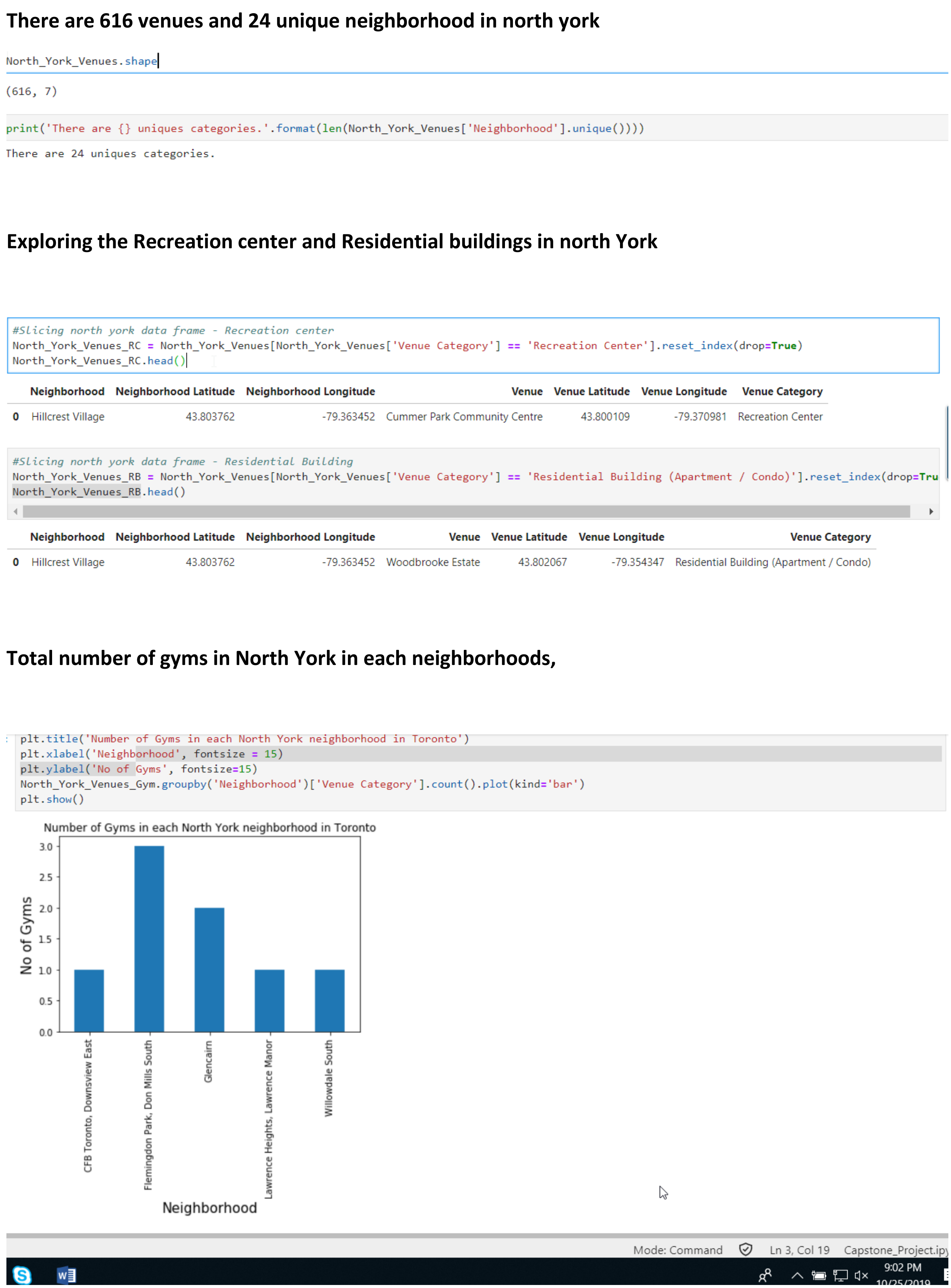


```
# structuring the venues into a data frame

venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

Extracting all the venues in the Northyork Borough



```
def getNearbyVenues(names, latitudes, longitudes, radius=1000):
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        results = requests.get(url).json()["response"]['groups'][0]['items']

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



There are 616 venues and 24 unique neighborhood in north york

```
North_York_Venues.shape
```

```
(616, 7)
```

```
print('There are {} uniques categories.'.format(len(North_York_Venues['Neighborhood'].unique())))
```

```
There are 24 uniques categories.
```

Exploring the Recreation center and Residential buildings in north York

```
#Slicing north york data frame - Recreation center
North_York_Venues_RC = North_York_Venues[North_York_Venues['Venue Category'] == 'Recreation Center'].reset_index(drop=True)
North_York_Venues_RC.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Hillcrest Village	43.803762	-79.363452	Cummer Park Community Centre	43.800109	-79.370981	Recreation Center

```
#Slicing north york data frame - Residential Building
North_York_Venues_RB = North_York_Venues[North_York_Venues['Venue Category'] == 'Residential Building (Apartment / Condo)'].reset_index(drop=Tru
North_York_Venues_RB.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Hillcrest Village	43.803762	-79.363452	Woodbrooke Estate	43.802067	-79.354347	Residential Building (Apartment / Condo)

Total number of gyms in North York in each neighborhoods,

```
plt.title('Number of Gyms in each North York neighborhood in Toronto')
plt.xlabel('Neighborhood', fontsize = 15)
plt.ylabel('No of Gyms', fontsize=15)
North_York_Venues_Gym.groupby('Neighborhood')['Venue Category'].count().plot(kind='bar')
plt.show()
```

Number of Gyms in each North York neighborhood in Toronto
No of Gyms
0.0
0.5
1.0
1.5
2.0
2.5
3.0
CFB Toronto, Downsview East
Flemingdon Park, Don Mills South
Glencairn
Lawrence Heights, Lawrence Manor
Willowdale South
Neighborhood
Mode: Command
Ln 3, Col 19
Capstone_Project.ipy
9:02 PM
10/25/2019

Conclusion:

- There are total 24 neighborhoods in North York , but only 8 gym’s are available in across 8 neighborhood.
- Hill crest village does not have a gym, 1 recreation center is available and also this neighborhood includes a residential building. Most of the residential building is equipped with physical center.
- There is large scope for physical center in North York borough.

Assumptions, limitation and future development:

- These results are based only the data available foursquare
- Accuracy of the data can be improved by combing other data sets related to population, and also analyzing the other categories like “indoor center”, “park”.