IBM Capstone Project

L V Pavan Kumar Maddula

Exploring Toronto Neighbourhoods - to open a Gym

Background

This project is a part of IBM Data Science professional program Capstone Project. The main objective of this project is to define a business problem and work through real time data to make informed decision which can help to solve the taken problem.

In this project the steps taken to gather, analyse and analysing the data explained and provided a conclusion which can help the business to take the decision.

1. Introduction

Prospect of opening a Gym in Toronto, Canada

Toronto, the capital of the province of Ontario, is the most populous Canadian city. We are using general assumption that with more populous area there is more chance of foot fall and subscription in gym. To verify this general perception, we will try to analyze if there is any correlation between dense areas Vs number of gyms in any area.

Finally, the aim of this project is to analyze each neighborhood in Toronto to identify the profitable area and will go through the process to plan where to open a gym.

1.1 Target Audience

Who will be interested in this project

1. Business personnel who wants to invest or open a gym
2. Business Analyst or Data Scientists, who wish to analyze the neighbourhoods of Toronto using Exploratory Data Analysis and other statistical & machine learning techniques to obtain all the necessary data, perform some operations on it.

2. Data acquisition and cleaning

2.1 Data sources

2.1.1) https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M - this wiki page contains information about the neighborhoods present in Toronto. This page has the postal code, borough & the name of all the neighborhoods present in Toronto.

2.1.2) “https://cocl.us/Geospatial_data” csv file to get all the geographical coordinates of the neighborhoods.

2.1.3) Location (latitude and longitude) and other information about various venues in Toronto (https://developer.foursquare.com/docs), Following information collected from this API,- Name, category, Latitude, Longitude

2.2 Data Cleaning

a) Scraping Toronto Neighborhoods Table from Wikipedia

Scraped the following Wikipedia page, "List of Postal code of Canada: M" in order to obtain the data about the Toronto & the Neighborhoods in it.

Data frame will consist of three columns: Postal Code, Borough, and Neighborhood

Only the cells that have an assigned borough will be processed. Borough that is not assigned are ignored.

More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that M5A is listed twice and has two neighborhoods: Harbour front and Regent Park.

These two rows will be combined into one row with the neighborhoods separated with a comma as shown in row 11 in the above table.

If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough. wikipedia - package is used to scrape the data from wiki.

Import data from wikipedia HTML

In [113]:
```
import pandas as pd
df = pd.read_html('https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M')[0]
```

In [114]:
```
df.head()
```

Out[114]:

	Postal Code	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront

Create a separate data frame with out not assigned in Borough column

In [115]:
```
df_1 = df [df ['Borough'] != 'Not assigned' ].reset_index(drop = True)
```

In [116]:
```
df_1.head()
```

Out[116]:

	Postal Code	Borough	Neighbourhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Regent Park, Harbourfront
3	M6A	North York	Lawrence Manor, Lawrence Heights
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

Assigning neighbourhood name as borough where non assigned in neighbourhood

In [117]:
```
#count =0
for index in range(len(df_1)):
     if (df_1.iloc[index]['Neighbourhood'])  == 'Not assigned' :
        df_1.iloc[index]['Neighbourhood'] = df_1.iloc[index]['Borough']
```

In [118]:
```
df_1.head()
```

Out[118]:

	Postal Code	Borough	Neighbourhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Regent Park, Harbourfront
3	M6A	North York	Lawrence Manor, Lawrence Heights
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

b) Adding geographical coordinates to the neighborhoods

Next important step is adding the geographical coordinates to these neighborhoods. To do so I'm extracting the data present in the Geospatial Data csv file and I'm combining it with the existing neighborhood dataframe by merging them both based on the postal code.

using the CSV file to get geo spatial data

In [120]:
```
df_CSV = pd.read_csv('http://cocl.us/Geospatial_data')
df_CSV.head()
```

Out[120]:

	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476

merging two databases based on postal code

In [121]:
```
df_1 = pd.merge(df_1, df_CSV, on='Postal Code')
```

In [122]:
```
df_1.head(10)
```

Out[122]:

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
5	M9A	Etobicoke	Islington Avenue, Humber Valley Village	43.667856	-79.532242
6	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353
7	M3B	North York	Don Mills	43.745906	-79.352188
8	M4B	East York	Parkview Hill, Woodbine Gardens	43.706397	-79.309937
9	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937

Explore the neighbourhoods in Toronto

In [123]:
```
Toronto_data = df_1 #[df_1 ['Borough'].str.contains('Toronto')].reset_index(drop = True)
Toronto_data.head(10)
#Toronto_data.shape
```

Out[123]:

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
5	M9A	Etobicoke	Islington Avenue, Humber Valley Village	43.667856	-79.532242
6	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353
7	M3B	North York	Don Mills	43.745906	-79.352188
8	M4B	East York	Parkview Hill, Woodbine Gardens	43.706397	-79.309937
9	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937

Number of neighbourhoods in each borough in Toronto

In [124]:
```
Toronto_data.groupby('Borough').count()
```

Out[124]:

Borough	Postal Code	Neighbourhood	Latitude	Longitude
Central Toronto	9	9	9	9
Downtown Toronto	19	19	19	19
East Toronto	5	5	5	5
East York	5	5	5	5
Etobicoke	12	12	12	12
Mississauga	1	1	1	1
North York	24	24	24	24
Scarborough	17	17	17	17
West Toronto	6	6	6	6
York	5	5	5	5

Toronto coordinates

In [126]:
```
from geopy.geocoders import Nominatim
address = 'Toronto, ON'

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

```
The geograpical coordinate of Toronto are 43.6534817, -79.3839347.
```

In [183]:
```
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=11)
map_toronto
```

Add markers for all neighbourhoods in Toronto

Get nearby venues and category of the venue in each neighbourhood

```
In [137]: print(Toronto_venues.shape)
          Toronto_venues.head(10)
```

(2146, 7)

Out[137]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Parkwoods	43.753259	-79.329656	Brookbanks Park	43.751976	-79.332140	Park
1	Parkwoods	43.753259	-79.329656	Variety Store	43.751974	-79.333114	Food & Drink Shop
2	Victoria Village	43.725882	-79.315572	Victoria Village Arena	43.723481	-79.315635	Hockey Arena
3	Victoria Village	43.725882	-79.315572	Portugril	43.725819	-79.312785	Portuguese Restaurant
4	Victoria Village	43.725882	-79.315572	Tim Hortons	43.725517	-79.313103	Coffee Shop
5	Victoria Village	43.725882	-79.315572	The Frig	43.727051	-79.317418	French Restaurant
6	Victoria Village	43.725882	-79.315572	Eglinton Ave E & Sloane Ave/Bermondsey Rd	43.726086	-79.313620	Intersection
7	Victoria Village	43.725882	-79.315572	Pizza Nova	43.725824	-79.312860	Pizza Place
8	Regent Park, Harbourfront	43.654260	-79.360636	Tandem Coffee	43.653559	-79.361809	Coffee Shop
9	Regent Park, Harbourfront	43.654260	-79.360636	Roselle Desserts	43.653447	-79.362017	Bakery

Do one hot coding to analyse each neighbourhood to understand percentage of different venue in each neighbourhood

Analyze Each Neighbourhood

```
In [140]: # one hot encoding
          Torento_onehot = pd.get_dummies(Toronto_venues[['Venue Category']], prefix="", prefix_sep="")

          # add neighborhood column back to dataframe
          Torento_onehot['Neighbourhood'] = Toronto_venues['Neighbourhood']
          #Torento_onehot['Neighborhood']
          # move neighborhood column to the first column
          fixed_columns = [Torento_onehot.columns[-1]] + list(Torento_onehot.columns[:-1])
          Torento_onehot = Torento_onehot[fixed_columns]

          Torento_onehot.head()
```

Out[140]:

	Neighbourhood	Accessories Store	Afghan Restaurant	Airport	Airport Food Court	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	...	Train Station	Vegetarian / Vegan Restaurant	Video Game Store	Vietnamese Restaurant	Warehouse Store	Wine Bar	Wine Shop	Wings Joint	Women's Store	Yoga Studio
0	Parkwoods	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	Parkwoods	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	Victoria Village	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	Victoria Village	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	Victoria Village	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 265 columns

Total number of different venues category. Total there are 35 gyms in toronoto

In [144]:
```
print(Toronto_venues['Venue Category'].value_counts())
```

```
Coffee Shop                          193
Café                                  99
Restaurant                            68
Pizza Place                           52
Park                                  52
Japanese Restaurant                   42
Sandwich Place                        42
Hotel                                 42
Italian Restaurant                    42
Bakery                                40
Clothing Store                        37
Gym                                   35
Bar                                   33
Grocery Store                         28
American Restaurant                   27
Sushi Restaurant                      26
Bank                                  25
Pub                                   24
Fast Food Restaurant                  24
Breakfast Spot                        24
Seafood Restaurant                    23
Thai Restaurant                       21
Pharmacy                              20
Ice Cream Shop                        20
Diner                                 19
Beer Bar                              18
Vegetarian / Vegan Restaurant         17
Gastropub                             17
Chinese Restaurant                    17
Bookstore                             16
                                     ...
Coworking Space                        1
```

3. Exploratory data analysis

3. Exploratory Data Analysis

3.1 Relationship between neighborhood and Gym

First we will extract the Neighborhood and Gym column from the above toronto dataframe for further analysis:

In [145]:
```
Toronto_part = Torento_grouped[['Neighbourhood', 'Gym']]
Toronto_part
```

Out[145]:

	Neighbourhood	Gym
0	Agincourt	0.000000
1	Alderwood, Long Branch	0.166667
2	Bathurst Manor, Wilson Heights, Downsview North	0.000000
3	Bayview Village	0.000000
4	Bedford Park, Lawrence Manor East	0.000000
5	Berczy Park	0.017241
6	Birch Cliff, Cliffside West	0.000000
7	Brockton, Parkdale Village, Exhibition Place	0.045455
8	Business reply mail Processing Centre, South C...	0.000000
9	CN Tower, King and Spadina, Railway Lands, Har...	0.000000
10	Caledonia-Fairbanks	0.000000
11	Canada Post Gateway Processing Centre	0.076923
12	Cedarbrae	0.000000
13	Central Bay Street	0.000000
14	Christie	0.000000
15	Church and Wellesley	0.000000
16	Clarks Corners, Tam O'Shanter, Sullivan	0.000000
17	Cliffside, Cliffcrest, Scarborough Village West	0.000000

Add latitude and longitude to neighbourhood

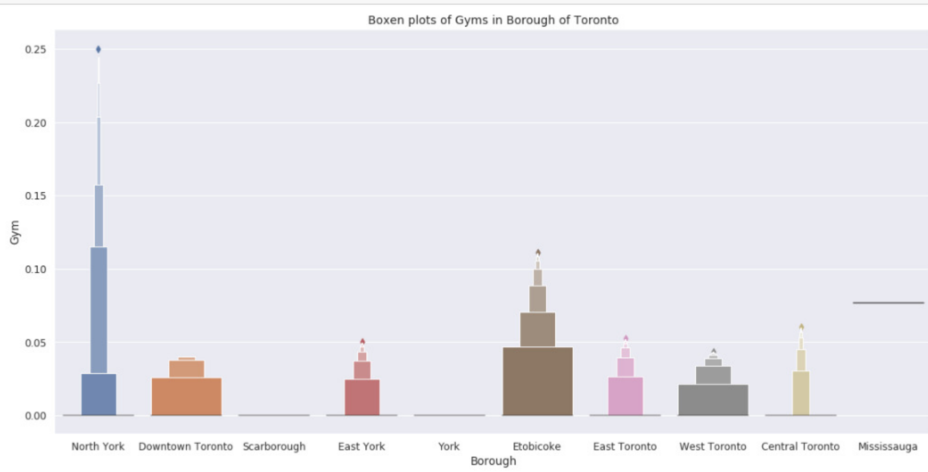
In [146]:
```
Toronto_merged = pd.merge(Toronto_data, Toronto_part, on='Neighbourhood')
Toronto_merged
```

Out[146]:

	Postal Code	Borough	Neighbourhood	Latitude	Longitude	Gym
0	M3A	North York	Parkwoods	43.753259	-79.329656	0.000000
1	M4A	North York	Victoria Village	43.725882	-79.315572	0.000000
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636	0.000000
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763	0.000000
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494	0.031250
5	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353	0.000000
6	M3B	North York	Don Mills	43.745906	-79.352188	0.120000
7	M3C	North York	Don Mills	43.725900	-79.340923	0.120000
8	M4B	East York	Parkview Hill, Woodbine Gardens	43.706397	-79.309937	0.000000
9	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937	0.010000
10	M6B	North York	Glencairn	43.709577	-79.445073	0.000000
11	M9B	Etobicoke	West Deane Park, Princess Gardens, Martin Grov...	43.650943	-79.554724	0.000000
12	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	0.000000
13	M4C	East York	Woodbine Heights	43.695344	-79.318389	0.000000

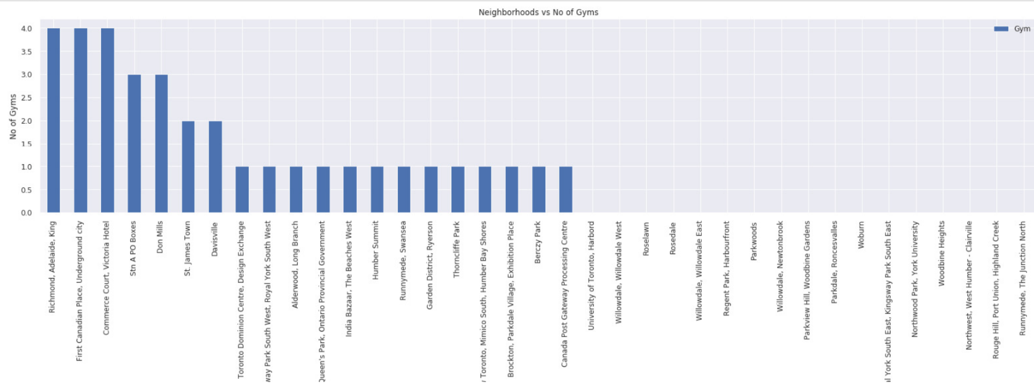
Plot to show number of gyms in each borough

In [36]:
```
# use categorical plot to identify most boroughs with densly populated with gyms

import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

fig = plt.figure(figsize=(19,9))

sns.set(font_scale=1.1)
sns.boxenplot(y="Gym", x="Borough", data=Toronto_merged);

plt.title('Boxen plots of Gyms in Borough of Toronto', fontsize=14)
plt.show()
```



Neighbourhood Vs gyms

In [148]:
```
graph = pd.DataFrame(Torento_onehot.groupby('Neighbourhood')['Gym'].sum())
graph = graph.sort_values(by ='Gym', ascending=False)
graph.iloc[:37].plot(kind='bar', figsize=(30,6))
plt.xlabel("Neighborhoods")
plt.ylabel("No of Gyms")
plt.title("Neighborhoods vs No of Gyms")
plt.show()
```



Getting population in each neighbourhood to understand if there is a relation between population and number of gyms

The Population information is available only according to Riding. So we will scrape the information riding Vs population. Each riding has many neighbourhoods. So our process will be to get data contains neighbourhood, its population and number of gyms in each neighbourhood. To get that we need to merge these three tables

1. Riding Vs population
2. Riding Vs neighbourhood
3. Neighbourhood Vs Gyms

So will in the end should get Neighbourhood , Population and number of gyms

In [149]:
```
#TORONTO & EAST YORK population distribution
TEY_population_df  = pd.read_html('https://en.m.wikipedia.org/wiki/Demographics_of_Toronto')[12]
TEY_population_df
```

Out[149]:

	Riding	Population	Ethnic Origin #1	%	Ethnic Origin #2	%.1	Ethnic Origin #3	%.2	Ethnic Origin #4	%.3	Ethnic Origin #5	%.4	Ethnic Origin #6	%.5	Ethnic Origin #7	%.6	Ethnic Origin #8	%.7	Ethnic Origin #9	%.8
0	Spadina-Fort York	114315	English	16.4	Chinese	16.0	Irish	14.6	Canadian	14.0	Scottish	13.2	French	7.70	German	7.6	NaN	NaN	NaN	NaN
1	Beaches-East York	108435	English	24.2	Irish	19.9	Canadian	19.7	Scottish	18.9	French	8.7	German	8.40	NaN	NaN	NaN	NaN	NaN	NaN
2	Davenport	107395	Portuguese	22.7	English	13.6	Canadian	12.8	Irish	11.5	Italian	11.1	Scottish	11.00	NaN	NaN	NaN	NaN	NaN	NaN
3	Parkdale-High Park	106445	English	22.3	Irish	20.0	Scottish	18.7	Canadian	16.1	German	9.8	French	8.88	Polish	8.5	NaN	NaN	NaN	NaN
4	Toronto-Danforth	105395	English	22.9	Irish	19.5	Scottish	18.7	Canadian	18.4	Chinese	13.8	French	8.86	German	8.8	Greek	7.3	NaN	NaN
5	Toronto-St. Paul's	104940	English	18.5	Canadian	16.1	Irish	15.2	Scottish	14.8	Polish	10.3	German	7.90	Russian	7.7	Italian	7.3	French	7.2
6	University-Rosedale	100520	English	20.6	Irish	16.6	Scottish	16.3	Canadian	15.2	Chinese	14.7	German	8.70	French	7.7	Italian	7.4	NaN	NaN
7	Toronto Centre	99590	English	15.7	Canadian	13.7	Irish	13.4	Scottish	12.6	Chinese	12.5	French	7.20	NaN	NaN	NaN	NaN	NaN	NaN

In [150]:
```
TEY_population_df_1  =   TEY_population_df[['Riding','Population']]
TEY_population_df_1
```

Out[150]:

	Riding	Population
0	Spadina-Fort York	114315
1	Beaches-East York	108435
2	Davenport	107395
3	Parkdale-High Park	106445
4	Toronto-Danforth	105395
5	Toronto-St. Paul's	104940
6	University-Rosedale	100520
7	Toronto Centre	99590

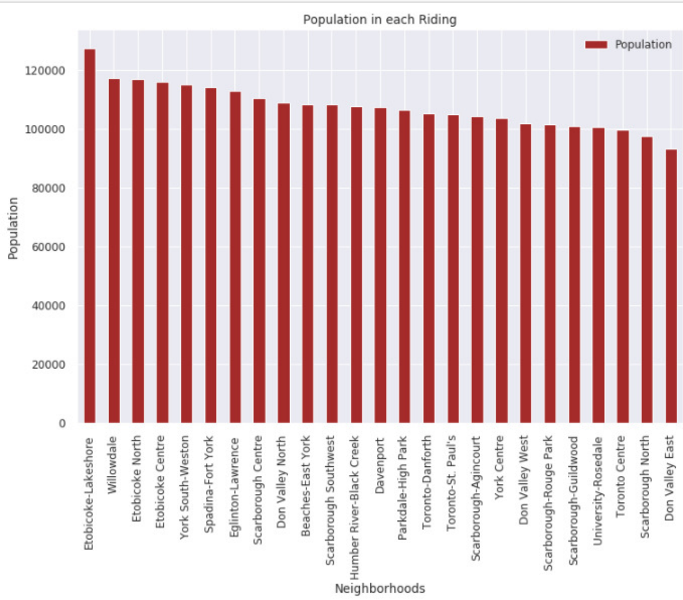
Merge all populations from different ridings

```
In [154]: #Merge all the population table
ET = ETY_population_df_1.append(TEY_population_df_1,sort=True).reset_index()
ET.drop('index',axis=1,inplace=True)
SN = NY_population_df_1.append(SC_population_df_1,sort=True).reset_index()
SN.drop('index',axis=1,inplace=True)
pop_df = SN.append(ET,sort=True).reset_index()
pop_df.drop('index',axis=1,inplace=True)
pop_df_1 = pop_df[['Riding', 'Population']]
pop_df_1
```

Out[154]:

	Riding	Population
0	Willowdale	117405
1	Eglinton-Lawrence	112925
2	Don Valley North	109060
3	Humber River-Black Creek	107725
4	York Centre	103760
5	Don Valley West	101790
6	Don Valley East	93170
7	Scarborough Centre	110450
8	Scarborough Southwest	108295
9	Scarborough-Agincourt	104225
10	Scarborough-Rouge Park	101445
11	Scarborough-Guildwood	101115
12	Scarborough North	97610
13	Etobicoke-Lakeshore	127520
14	Etobicoke North	116960
15	Etobicoke Centre	116055
16	York South-Weston	115130
17	Spadina-Fort York	114315
18	Beaches-East York	108435
19	Davenport	107395
20	Parkdale-High Park	106445
21	Toronto-Danforth	105395
22	Toronto-St. Paul's	104940
23	University-Rosedale	100520
24	Toronto Centre	99590

```
In [155]: bar_graph = pop_df_1.sort_values(by='Population', ascending=False)
bar_graph.plot(kind='bar',x='Riding', y='Population',figsize=(12,8), color='brown')
plt.title("Population in each Riding")
plt.xlabel("Ridings")
plt.ylabel("Population")
plt.show()
```



```
#### 3.4 Relationship between population and Gyms

#### First get the list of neighbourhoods present in the riding using the wikipedia geography section for each riding. Altering the riding names to match the wikipedia page so we can retrieve the neighborhoods present in those ridings
```

First get the list of neighborhoods present in the riding using the wikipedia geography section for each riding. Altering the riding names to match the wikipedia page so we can retrieve the neighborhoods present in those ridings

In [156]:
```python
#Altering the list to match the wikipedia page so we can retrieve the neighborhoods present in those Ridings
riding_list = pop_df_1['Riding'].to_list()
riding_list[riding_list.index('Scarborough Centre')] = 'Scarborough Centre (electoral district)'
riding_list[riding_list.index('Scarborough North')] = 'Scarborough North (electoral district)'
riding_list[riding_list.index('Willowdale')] = 'Willowdale, Toronto'
riding_list[riding_list.index('Etobicoke Centre')] = 'Etobicoke Centre (electoral district)'
riding_list[riding_list.index('Davenport')] = 'Davenport, Toronto'
riding_list
```

Out[156]:
```
['Willowdale, Toronto',
 'Eglinton-Lawrence',
 'Don Valley North',
 'Humber River-Black Creek',
 'York Centre',
 'Don Valley West',
 'Don Valley East',
 'Scarborough Centre (electoral district)',
 'Scarborough Southwest',
 'Scarborough-Agincourt',
 'Scarborough-Rouge Park',
 'Scarborough-Guildwood',
 'Scarborough North (electoral district)',
 'Etobicoke-Lakeshore',
 'Etobicoke North',
 'Etobicoke Centre (electoral district)',
 'York South-Weston',
 'Spadina-Fort York',
 'Beaches-East York',
 'Davenport, Toronto',
 'Parkdale-High Park',
 'Toronto-Danforth',
 "Toronto-St. Paul's",
 'University-Rosedale',
 'Toronto Centre']
```

Riding Vs neighbourhood

In [157]:
```python
import pandas as pd
Riding_neighborhood_df = pd.DataFrame()

for item in riding_list:
    section = wikipedia.WikipediaPage(item).section('Geography')
    if section!= None:
        start = section.find('neighbourhoods of') + 17
        stop = section.index('.',start)
        Riding_neighborhood_df = Riding_neighborhood_df.append({'Riding':item, 'Neighbourhoods':section[start:stop]},ignore_index=True)


Riding_neighborhood_df = Riding_neighborhood_df[['Riding','Neighbourhoods']]
Riding_neighborhood_df
```

Out[157]:

	Riding	Neighbourhoods
0	Don Valley North	Henry Farm, Bayview Village, Bayview Woods-St...
1	Humber River-Black Creek	Humber Summit, Humbermede, Humberlea, York Un...
2	York Centre	Westminster–Branson, Bathurst Manor, Wilson H...
3	Don Valley West	York Mills, Silver Hills, the western half of...
4	Don Valley East	Flemingdon Park, Don Mills, Graydon Hall, Par...
5	Scarborough Centre (electoral district)	Scarborough City Centre (west of McCowan Road...
6	Scarborough Southwest	Birch Cliff, Oakridge, Cliffside, Kennedy Par...
7	Scarborough-Agincourt	Steeles, L'Amoreaux, Tam O'Shanter-Sullivan, ...
8	Scarborough-Rouge Park	Morningside Heights, Rouge, Port Union, West ...
9	Scarborough-Guildwood	Guildwood, West Hill (west of Morningside Ave...
10	Scarborough North (electoral district)	Agincourt (east of Midland Avenue), Milliken ...
11	Etobicoke-Lakeshore	at part of the City of Toronto described as fo...
12	Etobicoke North	The Elms, Humberwood, Kingsview Village, This...
13	Etobicoke Centre (electoral district)	Eatonville (part), Islington-City Centre West...
14	Beaches-East York	the Beaches, Upper Beaches, East Danforth, O'...
15	Parkdale-High Park	High Park North and the south half of The Jun...
16	University-Rosedale	Rosedale, Little Italy, the Annex and Yorkvil...

Neighbourhood Vs population

In [158]:
```python
Neigh_pop = pd.merge(pop_df_1, Riding_neighborhood_df, on='Riding')

Neigh_pop.drop(columns=['Riding'],inplace =True)
Neigh_pop
```

Out[158]:

	Population	Neighbourhoods
0	109060	Henry Farm, Bayview Village, Bayview Woods-St...
1	107725	Humber Summit, Humbermede, Humberlea, York Un...
2	103760	Westminster–Branson, Bathurst Manor, Wilson H...
3	101790	York Mills, Silver Hills, the western half of...
4	93170	Flemingdon Park, Don Mills, Graydon Hall, Par...
5	108295	Birch Cliff, Oakridge, Cliffside, Kennedy Par...
6	104225	Steeles, L'Amoreaux, Tam O'Shanter-Sullivan, ...
7	101445	Morningside Heights, Rouge, Port Union, West ...
8	101115	Guildwood, West Hill (west of Morningside Ave...
9	127520	at part of the City of Toronto described as fo...
10	116960	The Elms, Humberwood, Kingsview Village, This...
11	108435	the Beaches, Upper Beaches, East Danforth, O'...
12	106445	High Park North and the south half of The Jun...
13	100520	Rosedale, Little Italy, the Annex and Yorkvil...

Spilt neighbourhood

In [159]:
```
Neigh_pop['split_neighborhoods'] = Neigh_pop['Neighbourhoods'].str.split(',')
Neigh_pop.drop(columns=['Neighbourhoods'],inplace=True,axis=1)
Neigh_pop = Neigh_pop.split_neighborhoods.apply(pd.Series).merge(Neigh_pop, left_index = True, right_index = True).drop(["split_neighborhoods"], axis = 1)\
                    .melt(id_vars = ['Population'], value_name = "Neighbourhood").drop("variable", axis = 1).dropna()

Neigh_pop.reset_index()
Neigh_pop
```

Out[159]:

	Population	Neighbourhood
0	109060	Henry Farm
1	107725	Humber Summit
2	103760	Westminster–Branson
3	101790	York Mills
4	93170	Flemingdon Park
5	108295	Birch Cliff
6	104225	Steeles
7	101445	Morningside Heights
8	101115	Guildwood
9	127520	at part of the City of Toronto described as fo...
10	116960	The Elms
11	108435	the Beaches
12	106445	High Park North and the south half of The Jun...
13	100520	Rosedale
14	109060	Bayview Village

Neighbourhood Vs Number of gyms

In [161]:
```
Toronto_part = Toronto_part.split_neighbourhoods.apply(pd.Series).merge(Toronto_part, left_index = True, right_index = True).drop(["split_neighbourhoods"], axis = 1)\
                    .melt(id_vars = ['Gym'], value_name = "Neighbourhood").drop("variable", axis = 1).dropna()

Toronto_part.reset_index()
Toronto_part
```

Out[161]:

	Gym	Neighbourhood
0	0.000000	Agincourt
1	0.166667	Alderwood
2	0.000000	Bathurst Manor
3	0.000000	Bayview Village
4	0.000000	Bedford Park
5	0.017241	Berczy Park
6	0.000000	Birch Cliff
7	0.045455	Brockton
8	0.000000	Business reply mail Processing Centre
9	0.000000	CN Tower
10	0.000000	Caledonia-Fairbanks
11	0.076923	Canada Post Gateway Processing Centre
12	0.000000	Cedarbrae
13	0.000000	Central Bay Street
14	0.000000	Christie
15	0.000000	Church and Wellesley
16	0.000000	Clarks Corners
17	0.000000	Cliffside
18	0.040000	Commerce Court
19	0.062500	Davisville
20	0.000000	Davisville North

Neighbourhood Vs Population Vs Number of gyms

In [162]:
```
pop_merged_Gym_perc = pd.merge(Neigh_pop, Toronto_part, on='Neighbourhood')
pop_merged_Gym_perc.head()
```

Out[162]:

	Population	Neighbourhood	Gym
0	109060	Henry Farm	0.0
1	108295	Oakridge	0.0
2	101445	Rouge	0.0
3	103760	Wilson Heights	0.0
4	101445	Port Union	0.0

From above table we can see that their no correlation between population & Number of gyms. Thus this marks end of the data cleaning & analyses step in this project.

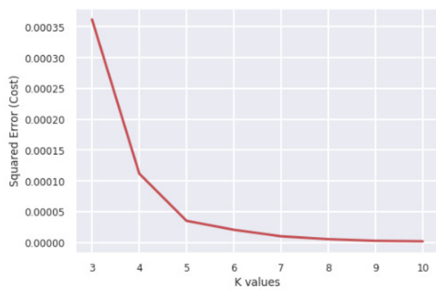
Next we will look into the predictive modelling. In the predictive modelling we are going to use Clustering techniques since this is analysis of unlabelled data. K-Means clustering is used to perform the analysis of the data at hand.

4. Predictive Modelling

4.1 Clustering Neighbourhoods of Toronto:

First step in K-means clustering is to identify best K value meaning the number of clusters in a given dataset. To do so we are going to use the elbow method on the Toronto dataset with no of Gyms percentage (i.e. toronto_merged dataframe).

In [163]:
```
from sklearn.cluster import KMeans

Toronto_part_clustering = Toronto_part.drop('Neighbourhood', 1)


error_cost = []

for i in range(3,11):
    KM = KMeans(n_clusters = i, max_iter = 100)
    try:
        KM.fit(Toronto_part_clustering)
    except ValueError:
        print("error on line",i)



    #calculate squared error for the clustered points
    error_cost.append(KM.inertia_/100)

#plot the K values aganist the squared error cost
plt.plot(range(3,11), error_cost, color='r', linewidth='3')
plt.xlabel('K values')
plt.ylabel('Squared Error (Cost)')
plt.grid(color='white', linestyle='-', linewidth=2)
plt.show()
```



Out[164]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f8b49648a90>`

In [165]:
```
kclusters = 6

Toronto_part_clustering = Toronto_part.drop('Neighbourhood', 1)

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(Toronto_part_clustering)

kmeans.labels_
```

Out[165]:
```
array([0, 4, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 3, 2, 0, 0,
       5, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 0,
       0, 0, 3, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 3, 0, 0, 0, 0,
       0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0,
       0, 0, 0, 0, 0, 0, 3, 0, 3, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2,
       0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int32)
```

Get neighbourhood with Gym, cluster labels longitude and latitude information

In [169]:
```
Toronto_merged_1 = Toronto_data
# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
Toronto_merged_1 = Toronto_merged_1.join(Toronto_part.set_index('Neighbourhood'), on='Neighbourhood')
Toronto_merged_1.dropna(subset=["Cluster Labels"], axis=0, inplace=True)
Toronto_merged_1.reset_index(drop=True, inplace=True)
Toronto_merged_1['Cluster Labels'].astype(int)
Toronto_merged_1
```

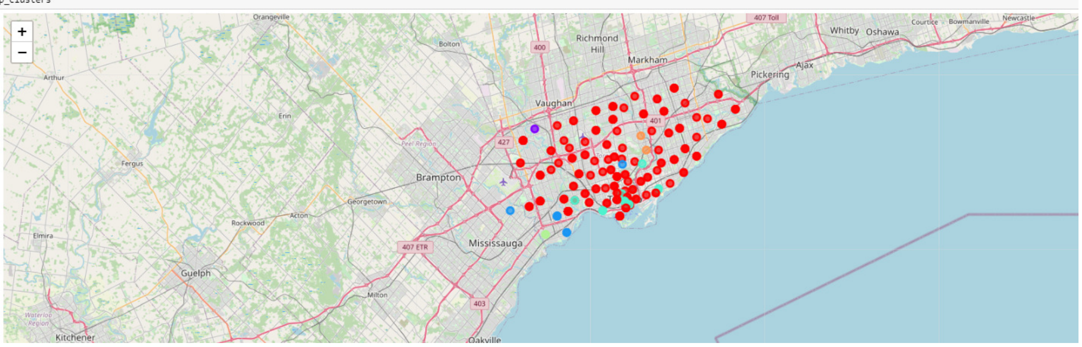
Out[169]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
0	M3A	North York	43.753259	-79.329656	Parkwoods	0.0	0.000000
1	M4A	North York	43.725882	-79.315572	Victoria Village	0.0	0.000000
2	M5A	Downtown Toronto	43.654260	-79.360636	Regent Park	0.0	0.000000
3	M6A	North York	43.718518	-79.464763	Lawrence Manor	0.0	0.000000
4	M7A	Downtown Toronto	43.662301	-79.389494	Queen's Park	3.0	0.031250
5	M1B	Scarborough	43.806686	-79.194353	Malvern	0.0	0.000000
6	M3B	North York	43.745906	-79.352188	Don Mills	5.0	0.120000
7	M4B	East York	43.706397	-79.309937	Parkview Hill	0.0	0.000000
8	M5B	Downtown Toronto	43.657162	-79.378937	Garden District	0.0	0.010000
9	M6B	North York	43.709577	-79.445073	Glencairn	0.0	0.000000
10	M9B	Etobicoke	43.650943	-79.554724	West Deane Park	0.0	0.000000
11	M1C	Scarborough	43.784535	-79.160497	Rouge Hill	0.0	0.000000
12	M3C	North York	43.725900	-79.340923	Don Mills	5.0	0.120000
13	M4C	East York	43.695344	-79.318389	Woodbine Heights	0.0	0.000000
14	M5C	Downtown Toronto	43.651494	-79.375418	St. James Town	3.0	0.022989
15	M5C	Downtown Toronto	43.651494	-79.375418	St. James Town	0.0	0.000000
16	M6C	York	43.693781	-79.428191	Humewood-Cedarvale	0.0	0.000000
17	M9C	Etobicoke	43.643515	-79.577201	Eringate	0.0	0.000000
18	M1E	Scarborough	43.763573	-79.188711	Guildwood	0.0	0.000000
19	M4E	East Toronto	43.676357	-79.293031	The Beaches	0.0	0.000000
20	M5E	Downtown Toronto	43.644771	-79.373306	Berczy Park	0.0	0.017241
21	M6E	York	43.689026	-79.453512	Caledonia-Fairbanks	0.0	0.000000
22	M1G	Scarborough	43.770992	-79.216917	Woburn	0.0	0.000000

Add markers to show clusters

```
        fill_opacity=0.7).add_to(map_clusters)
map_clusters
```

Out[175]:



4.2 Examine the Clusters:

In [176]:
```
#Cluster 0
Toronto_merged_1.loc[Toronto_merged_1['Cluster Labels'] == 0]
```

Out[176]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
0	M3A	North York	43.753259	-79.329656	Parkwoods	0.0	0.000000
1	M4A	North York	43.725882	-79.315572	Victoria Village	0.0	0.000000
2	M5A	Downtown Toronto	43.654260	-79.360636	Regent Park	0.0	0.000000
3	M6A	North York	43.718518	-79.464763	Lawrence Manor	0.0	0.000000
5	M1B	Scarborough	43.806686	-79.194353	Malvern	0.0	0.000000
7	M4B	East York	43.706397	-79.309937	Parkview Hill	0.0	0.000000
8	M5B	Downtown Toronto	43.657162	-79.378937	Garden District	0.0	0.010000
9	M6B	North York	43.709577	-79.445073	Glencairn	0.0	0.000000
10	M9B	Etobicoke	43.650943	-79.554724	West Deane Park	0.0	0.000000
11	M1C	Scarborough	43.784535	-79.160497	Rouge Hill	0.0	0.000000
13	M4C	East York	43.695344	-79.318389	Woodbine Heights	0.0	0.000000
15	M5C	Downtown Toronto	43.651494	-79.375418	St. James Town	0.0	0.000000
16	M6C	York	43.693781	-79.428191	Humewood-Cedarvale	0.0	0.000000
17	M9C	Etobicoke	43.643515	-79.577201	Eringate	0.0	0.000000

100 rows × 7 columns

```
In [177]: #Cluster 1
Toronto_merged_1.loc[Toronto_merged_1['Cluster Labels'] == 1]
```

Out[177]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
49	M9L	North York	43.756303	-79.565963	Humber Summit	1.0	0.5

```
In [178]: #Cluster 2
Toronto_merged_1.loc[Toronto_merged_1['Cluster Labels'] == 2]
```

Out[178]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
82	M7R	Mississauga	43.636966	-79.615819	Canada Post Gateway Processing Centre	2.0	0.076923
85	M4S	Central Toronto	43.704324	-79.388790	Davisville	2.0	0.062500
95	M8V	Etobicoke	43.605647	-79.501321	New Toronto	2.0	0.071429
109	M8Z	Etobicoke	43.628841	-79.520999	Mimico NW	2.0	0.076923
161	M8V	Etobicoke	43.605647	-79.501321	Mimico South	2.0	0.071429
171	M8Z	Etobicoke	43.628841	-79.520999	The Queensway West	2.0	0.076923
194	M8V	Etobicoke	43.605647	-79.501321	Humber Bay Shores	2.0	0.071429
198	M8Z	Etobicoke	43.628841	-79.520999	South of Bloor	2.0	0.076923
207	M8Z	Etobicoke	43.628841	-79.520999	Kingsway Park South West	2.0	0.076923
213	M8Z	Etobicoke	43.628841	-79.520999	Royal York South West	2.0	0.076923

```
In [179]: #Cluster 3
Toronto_merged_1.loc[Toronto_merged_1['Cluster Labels'] == 3]
```

Out[179]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
4	M7A	Downtown Toronto	43.662301	-79.389494	Queen's Park	3.0	0.031250
14	M5C	Downtown Toronto	43.651494	-79.375418	St. James Town	3.0	0.022989
29	M4H	East York	43.705369	-79.349372	Thorncliffe Park	3.0	0.045455
30	M5H	Downtown Toronto	43.650571	-79.384568	Richmond	3.0	0.040000
43	M6K	West Toronto	43.636847	-79.428191	Brockton	3.0	0.045455
46	M4L	East Toronto	43.668999	-79.315572	India Bazaar	3.0	0.052632
47	M5L	Downtown Toronto	43.648198	-79.379817	Commerce Court	3.0	0.040000
66	M6N	York	43.673185	-79.487262	Runnymede	3.0	0.028571

```
In [180]: #Cluster 4
Toronto_merged_1.loc[Toronto_merged_1['Cluster Labels'] == 4]
```

Out[180]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
100	M8W	Etobicoke	43.602414	-79.543484	Alderwood	4.0	0.166667
164	M8W	Etobicoke	43.602414	-79.543484	Long Branch	4.0	0.166667

```
In [181]: #Cluster 5
Toronto_merged_1.loc[Toronto_merged_1['Cluster Labels'] == 5]
```

Out[181]:

	Postal Code	Borough	Latitude	Longitude	Neighbourhood	Cluster Labels	Gym
6	M3B	North York	43.745906	-79.352188	Don Mills	5.0	0.12
12	M3C	North York	43.725900	-79.340923	Don Mills	5.0	0.12

5. Results and Discussion:

5.1 Results

In this project, as the business problem started with identifying a good neighborhood to open a new Gym, we looked into all the neighborhoods in Toronto, analyzed the population in each neighborhood & spread of gyms in those neighborhoods to come to conclusion about which neighborhood would be a better spot for opening a new Gym.

We identified that only North York, Etobicoke, Downtown Toronto, East York, & Scarborough boroughs have high amount of Gyms with the help of Violin plots between Number of Gyms in Borough of Toronto.

In all the ridings, Scarborough-Oakridge, Scarborough-Rouge, Scarborough- Port Union are the densely populated ridings.

With the help of clusters examining & boxen plots looks like North York, Etobicoke are already densely populated with Gyms. So, it is better idea to leave those boroughs out and consider only Scarborough, East Toronto for the new Gym's location.

After careful consideration it is a good idea to open a new Gym in Scarborough borough since it has high number of population which gives a higher number of customers possibility and lower competition since very less Gyms in the neighborhoods.

5.2 Discussion¶

According to this analysis, Scarborough borough will provide least competition for the new upcoming Gym as there are very less gyms in neighbourhoods. Also looking at the population distribution looks like it is densely populated which helps the new gyms by providing high customer visit possibility. So, this region could potentially be a perfect place for starting a gym.

Since population distribution of in each neighbourhood & number of gyms are the major feature in this analysis and it is not fully up-to date data, this analysis is definitely not far from being conclusory & it has lot of areas where it can be improved.

6. Conclusion:

We have used many python libraries to fetch the data , to manipulate the contents & to analyze and visualize those datasets. We have made use of Foursquare API to explore the venues in neighborhoods of Toronto, then get good amount of data from Wikipedia which we scraped with help of Wikipedia python library and visualized using various plots present in seaborn & matplotlib. We also applied machine learning technique to predict the output given the data and used Folium to visualize it on a map. Also, some of the drawbacks or areas of improvements shows us that this analysis can further be improved with help more data and different machine learning technique. Similarly we can use this project to analysis any scenario such opening a different cuisine etc. Hopefully, this project helps acts as initial guidance to take more complex real-life challenges using data-science.