This notebook contains the assignment for capstone project Neighbouhoods in Toronto.

SCRAPING THE WIKIPEDIA PAGE

In [1]:
```
# install geopy & folium
!conda install -c conda-forge geopy --yes
!conda install -c conda-forge folium=0.5.0 --yes
# import Libraries that we need
import requests
from bs4 import BeautifulSoup
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import numpy as np
from geopy.geocoders import Nominatim
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
import folium
```

```
folium-0.5.0          | 45 KB     |            |   0%
folium-0.5.0          | 45 KB     | ###5       |  35%
folium-0.5.0          | 45 KB     | ########## | 100%

altair-3.2.0          | 749 KB    |            |   0%
altair-3.2.0          | 749 KB    | 2          |   2%
altair-3.2.0          | 749 KB    | 8          |   9%
altair-3.2.0          | 749 KB    | #7         |  17%
altair-3.2.0          | 749 KB    | ##1        |  21%
altair-3.2.0          | 749 KB    | ###2       |  32%
altair-3.2.0          | 749 KB    | ###8       |  38%
altair-3.2.0          | 749 KB    | ####7      |  47%
altair-3.2.0          | 749 KB    | #####3     |  53%
altair-3.2.0          | 749 KB    | #####9     |  60%
altair-3.2.0          | 749 KB    | ######6    |  66%
altair-3.2.0          | 749 KB    | #######2   |  73%
altair-3.2.0          | 749 KB    | #######9   |  79%
altair-3.2.0          | 749 KB    | ########9  |  90%
altair-3.2.0          | 749 KB    | ########## |  96%
```

In [15]:
```
page = requests.get("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Ca
page_soup=BeautifulSoup(page.content,'html.parser')
```

In [16]:
```
table = page_soup.find('tbody')
rows = table.select('tr')
row = [r.get_text() for r in rows]
```

Data Wrangling

In [17]:
```
df = pd.DataFrame(row)
df = df[0].str.split('\n', expand=True)
df = df.rename(columns=df.iloc[0])
df = df.drop(df.index[0])
df.head()
```

Out[17]:

	Postcode	Borough	Neighbourhood
1	M1A	Not assigned	Not assigned
2	M2A	Not assigned	Not assigned
3	M3A	North York	Parkwoods
4	M4A	North York	Victoria Village
5	M5A	Downtown Toronto	Harbourfront

Ignore cells with a borough that is Not assigned

In [18]:
```
df=df[df.Borough !="Not assigned"]
df.head()
```

Out[18]:

	Postcode	Borough	Neighbourhood
3	M3A	North York	Parkwoods
4	M4A	North York	Victoria Village
5	M5A	Downtown Toronto	Harbourfront
6	M5A	Downtown Toronto	Regent Park
7	M6A	North York	Lawrence Heights

Combine neighborhoods which have the same postcode

In [6]:
```
df = df.groupby(['Postcode', 'Borough'], sort = False).agg(','.join)
df.reset_index(inplace = True)
df.head()
```

Out[6]:

	Postcode	Borough	Neighbourhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Harbourfront,Regent Park
3	M6A	North York	Lawrence Heights,Lawrence Manor
4	M7A	Queen's Park	Not assigned

If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough.

In [7]:
```
df = df.replace("Not assigned", "Queen's Park")
df.head()
```

Out[7]:

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

.shape method to print the number of rows of your dataframe

In [8]:
```
df.shape
```

Out[8]: (103, 3)

Read Geo CSV file

In [9]:
```
url = "http://cocl.us/Geospatial_data"
df1 = pd.read_csv(url)
df1.head()
```

Out[9]:

	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

Rename first column name

In [10]:
```
df1.rename(columns={'Postal Code': 'Postcode'}, inplace=True)
df1.head()
```

Out[10]:

	Postcode	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 frames on Postcode

In [11]:
```
df = pd.merge(df, df1, on='Postcode')
df.head()
```

Out[11]:

	Postcode	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	Harbourfront,Regent Park	43.654260	-79.360636
3	M6A	North York	Lawrence Heights,Lawrence Manor	43.718518	-79.464763
4	M7A	Queen's Park	Queen's Park	43.662301	-79.389494

Show how many Borough & Neighbourhood in the dataframe

In [12]:
```
print('The dataframe has {} Borough and {} Neighbourhood.'.format(
        len(df['Borough'].unique()),
        df.shape[0]
    )
)
```

```
The dataframe has 11 Borough and 103 Neighbourhood.
```

Create a new dataframe for only boroughs that contain the word Toronto

In [13]:
```
Toronto=df[df['Borough'].str.contains('Toronto')]
Toronto
```

Out[13]:

	Postcode	Borough	Neighbourhood	Latitude	Longitude
2	M5A	Downtown Toronto	Harbourfront,Regent Park	43.654260	-79.360636
9	M5B	Downtown Toronto	Ryerson,Garden District	43.657162	-79.378937
15	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418
19	M4E	East Toronto	The Beaches	43.676357	-79.293031
20	M5E	Downtown Toronto	Berczy Park	43.644771	-79.373306
24	M5G	Downtown Toronto	Central Bay Street	43.657952	-79.387383
25	M6G	Downtown Toronto	Christie	43.669542	-79.422564
30	M5H	Downtown Toronto	Adelaide,King,Richmond	43.650571	-79.384568
31	M6H	West Toronto	Dovercourt Village,Dufferin	43.669005	-79.442259
36	M5J	Downtown Toronto	Harbourfront East,Toronto Islands,Union Station	43.640816	-79.381752
37	M6J	West Toronto	Little Portugal,Trinity	43.647927	-79.419750
41	M4K	East Toronto	The Danforth West,Riverdale	43.679557	-79.352188
42	M5K	Downtown Toronto	Design Exchange,Toronto Dominion Centre	43.647177	-79.381576
43	M6K	West Toronto	Brockton,Exhibition Place,Parkdale Village	43.636847	-79.428191
47	M4L	East Toronto	The Beaches West,India Bazaar	43.668999	-79.315572
48	M5L	Downtown Toronto	Commerce Court,Victoria Hotel	43.648198	-79.379817
54	M4M	East Toronto	Studio District	43.659526	-79.340923
61	M4N	Central Toronto	Lawrence Park	43.728020	-79.388790
62	M5N	Central Toronto	Roselawn	43.711695	-79.416936
67	M4P	Central Toronto	Davisville North	43.712751	-79.390197
68	M5P	Central Toronto	Forest Hill North,Forest Hill West	43.696948	-79.411307
69	M6P	West Toronto	High Park,The Junction South	43.661608	-79.464763
73	M4R	Central Toronto	North Toronto West	43.715383	-79.405678
74	M5R	Central Toronto	The Annex,North Midtown,Yorkville	43.672710	-79.405678
75	M6R	West Toronto	Parkdale,Roncesvalles	43.648960	-79.456325
79	M4S	Central Toronto	Davisville	43.704324	-79.388790
80	M5S	Downtown Toronto	Harbord,University of Toronto	43.662696	-79.400049

	Postcode	Borough	Neighbourhood	Latitude	Longitude
81	M6S	West Toronto	Runnymede,Swansea	43.651571	-79.484450
83	M4T	Central Toronto	Moore Park,Summerhill East	43.689574	-79.383160
84	M5T	Downtown Toronto	Chinatown,Grange Park,Kensington Market	43.653206	-79.400049
86	M4V	Central Toronto	Deer Park,Forest Hill SE,Rathnelly,South Hill,...	43.686412	-79.400049
87	M5V	Downtown Toronto	CN Tower,Bathurst Quay,Island airport,Harbourf...	43.628947	-79.394420
91	M4W	Downtown Toronto	Rosedale	43.679563	-79.377529
92	M5W	Downtown Toronto	Stn A PO Boxes 25 The Esplanade	43.646435	-79.374846
96	M4X	Downtown Toronto	Cabbagetown,St. James Town	43.667967	-79.367675
97	M5X	Downtown Toronto	First Canadian Place,Underground city	43.648429	-79.382280
99	M4Y	Downtown Toronto	Church and Wellesley	43.665860	-79.383160
100	M7Y	East Toronto	Business Reply Mail Processing Centre 969 Eastern	43.662744	-79.321558

Generate map to visualize neighborhoods and how they cluster together

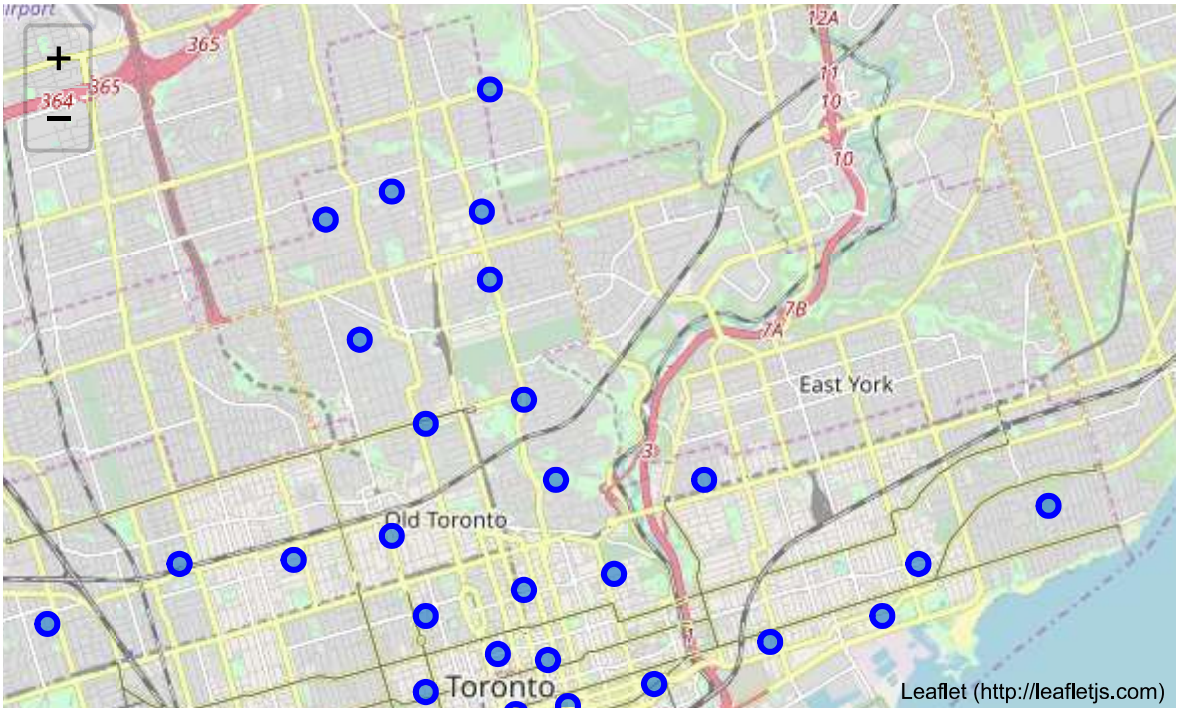
In [19]:
```
address = 'Toronto'
geolocator = Nominatim(user_agent="Toronto_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude

Toronto_map = folium.Map(location=[latitude, longitude], zoom_start=10)

for lat, lng, borough, neighborhood in zip(Toronto['Latitude'], Toronto['Long
                                           Toronto['Borough'], Toronto['Neigh
    label = '{}, {}'.format(neighborhood, borough)
    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(Toronto_map)

Toronto_map
```

Out[19]:



In [ ]: