Segmenting and Clustering Neighborhoods of Toronto

Importing necessary libraries

In [20]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!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 latitud
e and longitude values

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

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

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

#!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

#While scraping the wikipedia page an error occured where the absence of
lxml was indicated. This statement imports it. Upon installation, Kernel
restart was necessary. /


print('Libraries imported.')
```

```
Libraries imported.
```

In [1]:
```
#conda install -c anaconda lxml
```

Scraping the table in the Wikipedia URL

In [4]:
```
dfs = pd.read_html('https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M')
len(dfs)
```

Out[4]: 3

Turns out, there are multiple tables in the page. The first table has the information we value.

In [5]:
```
df_table1 = dfs[0]
```

Eliminating the Boroughs which have the value 'Not assigned'. From the data, it seems like if the borough is not assigned, the neighborhood is not assigned as well.

In [6]:
```
df_toronto = df_table1[df_table1['Borough']!='Not assigned'].reset_index(drop=True)
df_toronto.rename(columns={'Postal Code':'PostalCode'}, inplace=True)
```

The dataframe of interest is shown below. Truncated for presentability. For the full table check the other juypter notebook.

In [7]:
```
df_toronto.head()
```

Out[7]:

	PostalCode	Borough	Neighborhood
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

Shape method of the dataframe indicating the rows and columns

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

Out[8]: (103, 3)

An attempt was made to use the geocoder library. But since it took a long time to return coordinates for just one postal code, the csv file is being used

In [9]:
```
# %pip install geocoder
# import geocoder

# initialize your variable to None
# lat_lng_coords = None
#postal_code = "M5G"

# loop until you get the coordinates
#while(lat_lng_coords is None):
#  g = geocoder.google('{}, Toronto, Ontario'.format(postal_code))
#  lat_lng_coords = g.latlng

#latitude = lat_lng_coords[0]
#longitude = lat_lng_coords[1]
```

The lat, long info is being converted to a dataframe

In [10]:
```
df_lat_long = pd.read_csv("https://cocl.us/Geospatial_data")
df_lat_long.rename(columns={'Postal Code':'PostalCode'}, inplace=True)
df_lat_long.head()
```

Out[10]:

	PostalCode	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

New Columns are created in the Toronto neighborhood dataframe for Latitude and Longitude and are assigned 'None'

In [12]:
```
df_toronto['Latitude'] = None
df_toronto['Longitude'] = None
df_toronto.head()
```

Out[12]:

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M3A	North York	Parkwoods	None	None
1	M4A	North York	Victoria Village	None	None
2	M5A	Downtown Toronto	Regent Park, Harbourfront	None	None
3	M6A	North York	Lawrence Manor, Lawrence Heights	None	None
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	None	None

In the following cells, the population of latitudes and longitudes is achieved on the Toronto Neighborhood dataframe

This is achieved by querying the lat, long dataframe for postal code, obtaining latitude and longitudes as a list, converting the lists to a dataframe and appending the dataframes as columns to the Toronto Neighborhood dataframe

In [13]:
```
LatList = []
LongList = []
for index, row in df_toronto.iterrows():
    pc = (df_toronto.at[index, "PostalCode"])
    LatList.append(df_lat_long.query("PostalCode == '"+str(pc)+"'")["Lat
itude"].tolist()[0])
    LongList.append(df_lat_long.query("PostalCode == '"+str(pc)+"'")["Lo
ngitude"].tolist()[0])
```

In [16]:
```
df_toronto['Latitude']= pd.DataFrame(LatList,columns=['Latitude'])
df_toronto['Longitude']= pd.DataFrame(LongList,columns=['Latitude'])
```

Filtering Boroughs that contain the word 'Toronto'

In [33]:
```
toronto_data = df_toronto[df_toronto['Borough'].str.contains("Toronto",
case=False)].reset_index(drop=True)
toronto_data
```

Out[33]:

	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
1	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
2	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937
3	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418
4	M4E	East Toronto	The Beaches	43.676357	-79.293031
5	M5E	Downtown Toronto	Berczy Park	43.644771	-79.373306
6	M5G	Downtown Toronto	Central Bay Street	43.657952	-79.387383
7	M6G	Downtown Toronto	Christie	43.669542	-79.422564
8	M5H	Downtown Toronto	Richmond, Adelaide, King	43.650571	-79.384568
9	M6H	West Toronto	Dufferin, Dovercourt Village	43.669005	-79.442259
10	M5J	Downtown Toronto	Harbourfront East, Union Station, Toronto Islands	43.640816	-79.381752
11	M6J	West Toronto	Little Portugal, Trinity	43.647927	-79.419750
12	M4K	East Toronto	The Danforth West, Riverdale	43.679557	-79.352188
13	M5K	Downtown Toronto	Toronto Dominion Centre, Design Exchange	43.647177	-79.381576
14	M6K	West Toronto	Brockton, Parkdale Village, Exhibition Place	43.636847	-79.428191
15	M4L	East Toronto	India Bazaar, The Beaches West	43.668999	-79.315572
16	M5L	Downtown Toronto	Commerce Court, Victoria Hotel	43.648198	-79.379817
17	M4M	East Toronto	Studio District	43.659526	-79.340923
18	M4N	Central Toronto	Lawrence Park	43.728020	-79.388790
19	M5N	Central Toronto	Roselawn	43.711695	-79.416936
20	M4P	Central Toronto	Davisville North	43.712751	-79.390197
21	M5P	Central Toronto	Forest Hill North & West, Forest Hill Road Park	43.696948	-79.411307
22	M6P	West Toronto	High Park, The Junction South	43.661608	-79.464763
23	M4R	Central Toronto	North Toronto West, Lawrence Park	43.715383	-79.405678
24	M5R	Central Toronto	The Annex, North Midtown, Yorkville	43.672710	-79.405678
25	M6R	West Toronto	Parkdale, Roncesvalles	43.648960	-79.456325
26	M4S	Central Toronto	Davisville	43.704324	-79.388790
27	M5S	Downtown Toronto	University of Toronto, Harbord	43.662696	-79.400049

	PostalCode	Borough	Neighborhood	Latitude	Longitude
28	M6S	West Toronto	Runnymede, Swansea	43.651571	-79.484450
29	M4T	Central Toronto	Moore Park, Summerhill East	43.689574	-79.383160
30	M5T	Downtown Toronto	Kensington Market, Chinatown, Grange Park	43.653206	-79.400049
31	M4V	Central Toronto	Summerhill West, Rathnelly, South Hill, Forest...	43.686412	-79.400049
32	M5V	Downtown Toronto	CN Tower, King and Spadina, Railway Lands, Har...	43.628947	-79.394420
33	M4W	Downtown Toronto	Rosedale	43.679563	-79.377529
34	M5W	Downtown Toronto	Stn A PO Boxes	43.646435	-79.374846
35	M4X	Downtown Toronto	St. James Town, Cabbagetown	43.667967	-79.367675
36	M5X	Downtown Toronto	First Canadian Place, Underground city	43.648429	-79.382280
37	M4Y	Downtown Toronto	Church and Wellesley	43.665860	-79.383160
38	M7Y	East Toronto	Business reply mail Processing Centre, South C...	43.662744	-79.321558

Mapping the filtered Boroughs from the previous step

In [37]:
```
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
```

Getting the coordinates for Toronto to set the map zoom

In [38]:
```
address = 'Toronto, Canada'

geolocator = Nominatim(user_agent="ca_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.
```

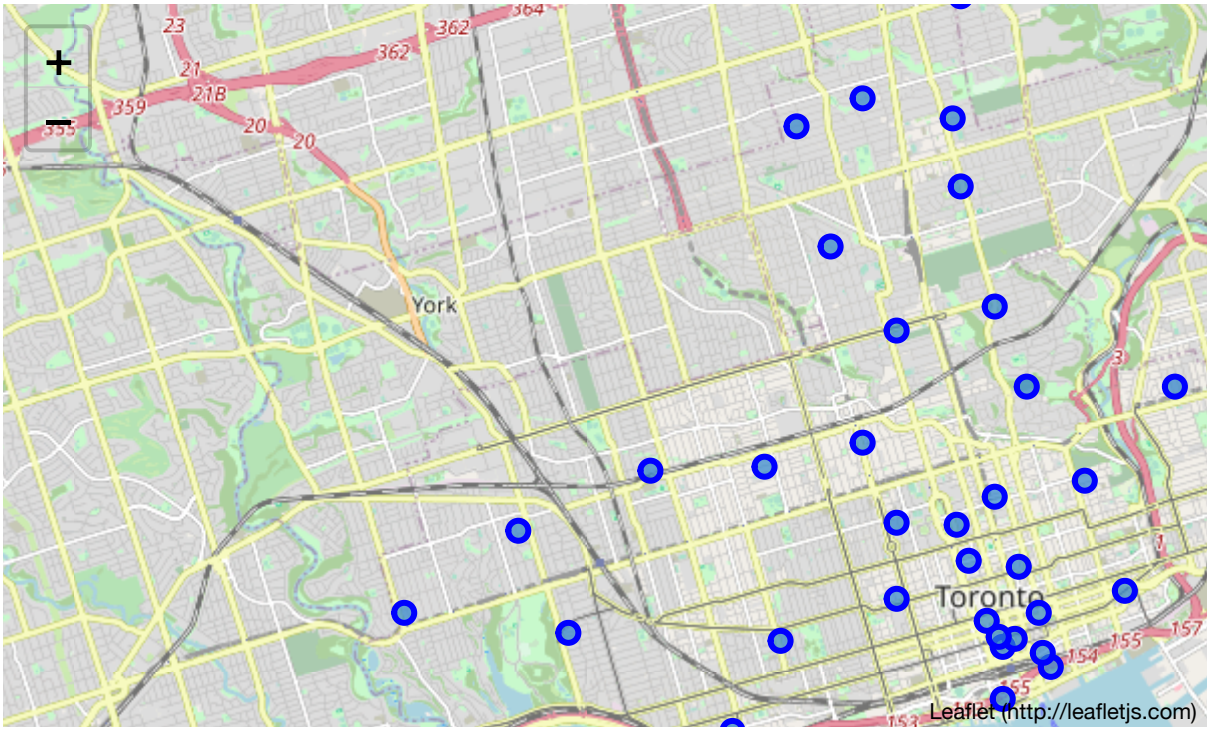
Mapping locations

In [39]:
```
# create map of Toronto using latitude and longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=12)

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

map_toronto
```

Out[39]:



Foursquare credentials

In [40]:
```
CLIENT_ID = '' # your Foursquare ID
CLIENT_SECRET = '' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

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

```
Your credentails:
CLIENT_ID: JBY0YANCCPVYDZJRGUC4PKUJXPXRSCB52IZYBIN3VV4BH3OQ
CLIENT_SECRET:MV4MQUCRVYAIOPQTHU2EGDWRV4SSLTQPHAZUR5LQOC5C1QWX
```

Methods to obtain top venues across all neighborhoods near Toronto

In [41]:
```
# 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']
```

In [43]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    LIMIT=100
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        #print(name)

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

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

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

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

Steps to know the number of interesting venues at every neighborhood near Toronto

In [69]:
```
toronto_nearby_venues = getNearbyVenues(names=toronto_data['Neighborhood'],
                                        latitudes=toronto_data['Latitude'],
                                        longitudes=toronto_data['Longitude']
                                       )

toronto_venue_aggregate = toronto_nearby_venues.groupby('Neighborhood').count()
```

In [70]:
```
toronto_nearby_venues.head()
```

Out[70]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Regent Park, Harbourfront	43.65426	-79.360636	Roselle Desserts	43.653447	-79.362017	Bakery
1	Regent Park, Harbourfront	43.65426	-79.360636	Tandem Coffee	43.653559	-79.361809	Coffee Shop
2	Regent Park, Harbourfront	43.65426	-79.360636	Morning Glory Cafe	43.653947	-79.361149	Breakfast Spot
3	Regent Park, Harbourfront	43.65426	-79.360636	Cooper Koo Family YMCA	43.653249	-79.358008	Distribution Center
4	Regent Park, Harbourfront	43.65426	-79.360636	Body Blitz Spa East	43.654735	-79.359874	Spa

In [71]:
```
toronto_venue_count = toronto_venue_aggregate.filter(['Neighborhood', 'Venue'])
toronto_venue_count.rename(columns={'Venue':'# of interesting venues'}, inplace=True)
```

In [72]:
```
#print(manhattan_venues)
toronto_venue_count
```

Out[72]:

Neighborhood	# of interesting venues
Berczy Park	56
Brockton, Parkdale Village, Exhibition Place	23
Business reply mail Processing Centre, South Central Letter Processing Plant Toronto	16
CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport	18
Central Bay Street	65
Christie	16
Church and Wellesley	78
Commerce Court, Victoria Hotel	100
Davisville	31
Davisville North	9
Dufferin, Dovercourt Village	15
First Canadian Place, Underground city	100
Forest Hill North & West, Forest Hill Road Park	4
Garden District, Ryerson	100
Harbourfront East, Union Station, Toronto Islands	100
High Park, The Junction South	23
India Bazaar, The Beaches West	21
Kensington Market, Chinatown, Grange Park	60
Lawrence Park	3
Little Portugal, Trinity	45
Moore Park, Summerhill East	2
North Toronto West, Lawrence Park	19
Parkdale, Roncesvalles	14
Queen's Park, Ontario Provincial Government	32
Regent Park, Harbourfront	46
Richmond, Adelaide, King	95
Rosedale	4
Roselawn	3
Runnymede, Swansea	38
St. James Town	78
St. James Town, Cabbagetown	48

Neighborhood	# of interesting venues
Stn A PO Boxes	94
Studio District	41
Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park	17
The Annex, North Midtown, Yorkville	22
The Beaches	4
The Danforth West, Riverdale	43
Toronto Dominion Centre, Design Exchange	100
University of Toronto, Harbord	35

Neighborhoods to spend time at

In [73]:
```
toronto_venue_count.loc[(toronto_venue_count['# of interesting venues']
>= 100)]
```

Out[73]:

Neighborhood	# of interesting venues
Commerce Court, Victoria Hotel	100
First Canadian Place, Underground city	100
Garden District, Ryerson	100
Harbourfront East, Union Station, Toronto Islands	100
Toronto Dominion Centre, Design Exchange	100

Neighborhoods that can be avoided

In [74]:
```
toronto_venue_count.loc[(toronto_venue_count['# of interesting venues']
<= 10)]
```

Out[74]:

Neighborhood	# of interesting venues
Davisville North	9
Forest Hill North & West, Forest Hill Road Park	4
Lawrence Park	3
Moore Park, Summerhill East	2
Rosedale	4
Roselawn	3
The Beaches	4

In [ ]: