Requirements for assignment at Week 3:

1. Scrape the Wikipedia page
2. Wrangle, clean and read data into apandas dataframe (make it as a structured format like the New York dataset)
3. Replicate the analysis that we did to the New York City to explore and cluster the neighborhoods in the city of Toronto.

Please note that, since directly sharing notebook might lead to the missing of visualized maps, I saved the whole notebook as a pdf file and upload it to the Github, which will make you easier to check the results of this assignment.

Part one: Data Scrapping and Pre-processing

In [1]:

```
# importing all necessary packages for scraping
from bs4 import BeautifulSoup
from urllib.request import urlopen
import pandas as pd
import urllib
import requests

def get_html(url):
    headers = {'User-Agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHT
    req = urllib.request.Request(url, headers=headers)
    global html
    html = urlopen(req).read().decode('ISO-8859-1')
    global soup
    soup = BeautifulSoup(html,'html.parser')

get_html("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M")
```

In [2]:

```
# find the table first, go to the tbody section and find all labels called "tr"
content_extracted = soup.find("table")
content = content_extracted.tbody.find_all("tr")

res = []
for tr in content:

    td = tr.find_all("td")
    data = [tr.text for tr in td]

    # according to the requirement:
    # only process the cells that have an assigned borough.
    # Ignore cells with a borough that is Not assigned.
    if (data != []) and (data[1].strip() != "Not assigned"):

        # according to another requirement:
        # if a cell has a borough but a Not assigned neighborhood
        # then the neighborhood will be the same as the borough
        if data[2].strip() == "Not assigned":
            data[2] = data[1]

        res.append(data)

# Create the dataframe
df = pd.DataFrame(res, columns = ["PostalCode", "Borough", "Neighborhood"])
df.head()
```

Out[2]:

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

In [3]:

```
# there are some "\n", which needs to be replaced
df["Neighborhood"] = df["Neighborhood"].str.replace("\n","")
df["Borough"] = df["Borough"].str.replace("\n","")
df["PostalCode"] = df["PostalCode"].str.replace("\n","")
print(df.head())
print("Shape: ", df.shape)
# we don't need to group the postcodes since it has been done by wiki website itself!!!
```

```
  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:  (103, 3)
```

Part two: transfer addresses to lat/lon

In [4]:

```
# below it's the try to retrieve the data from geocoder package,
# however, there is no response for a long time. so choose to use the csv file provided by this cour

# import geocoder

# # initialize your variable to None
# lat_lng_coords = None

# # loop until you get the coordinates
# while(lat_lng_coords is None):
#   g = geocoder.google('{}, Toronto, Ontario'.format("M5G"))
#   lat_lng_coords = g.latlng

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

In [5]:

```
# import the csv file from online source
lat_lon = pd.read_csv('https://cocl.us/Geospatial_data')
lat_lon.head()
```

Out[5]:

	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

In [6]:

```
# merge two tables according to the shared postcodes

df_toronto = pd.merge(df, lat_lon, how = "left", left_on = 'PostalCode', \
                      right_on = 'Postal Code')
df_toronto.drop("Postal Code", axis=1, inplace=True)
df_toronto.head()
```

Out[6]:

	PostalCode	Borough	Neighborhood	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

Part three: Explore and cluster the neighborhoods

In [7]:

```
# first of all, cluster the neighborhoods according to their lat/lon.

# get a general idea of how many boroughs and neighborhoods we have
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(df_toronto['Borough'].unique()),
        len(df_toronto['Neighborhood'].unique())
    )
)
```

```
The dataframe has 10 boroughs and 99 neighborhoods.
```

In [8]:

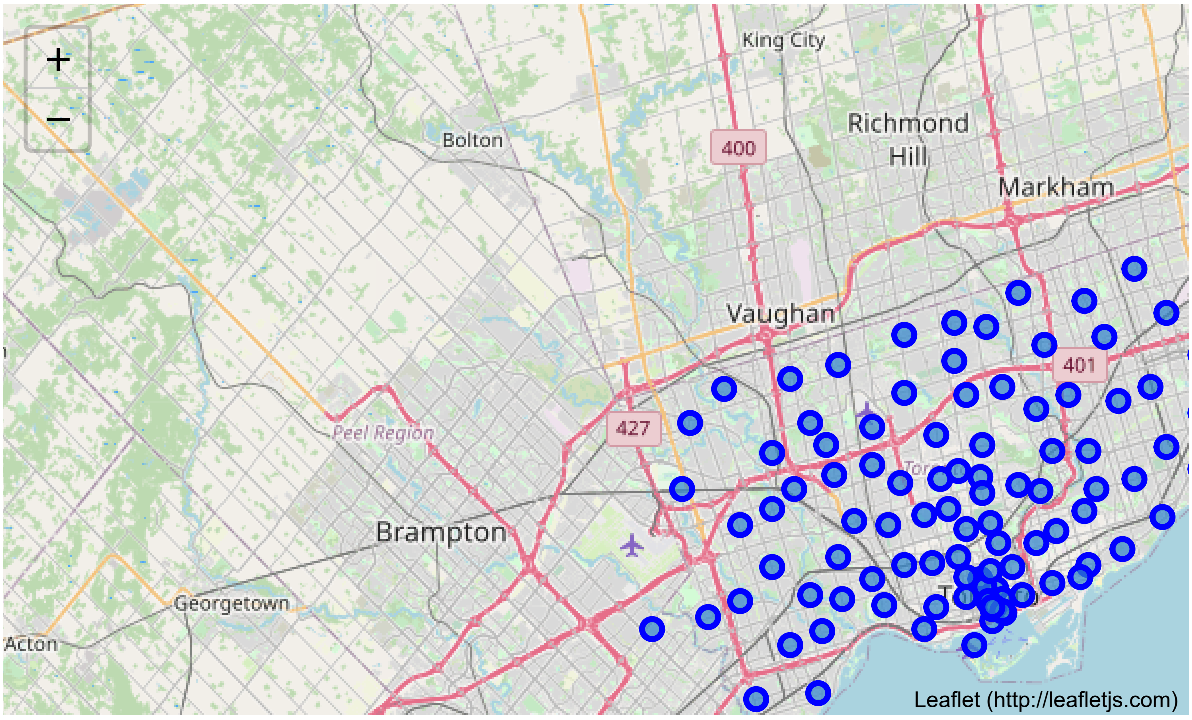
```
# create a map of Toronto with neighborhoods

from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
import folium # map rendering library

address = "Toronto, ON"

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

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

In [9]:

```
# create map of Toronto using latitude and longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=10)
map_toronto

# add more points on the map
for lat, lng, borough, neighborhood in zip(
        df_toronto[’Latitude’],
        df_toronto[’Longitude’],
        df_toronto[’Borough’],
        df_toronto[’Neighborhood’]):
    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(map_toronto)

map_toronto
```

Out[9]:



In [10]:

```
# next, I want to know the cluster distributions of the neighborhoods around Toronto.
# So I used K-means clustering for all the neigbourhoods
from sklearn.cluster import KMeans

k = 5 # let's assume the number of clusters is 5
toronto_clustering = df_toronto.drop(['PostalCode','Borough','Neighborhood'],1)
kmeans = KMeans(n_clusters = k,random_state=0).fit(toronto_clustering)
kmeans.labels_[0:10]

# create a new dataframe that includes the clustering information
df_toronto.insert(0, 'Cluster Labels', kmeans.labels_)
df_toronto
```

Out[10]:

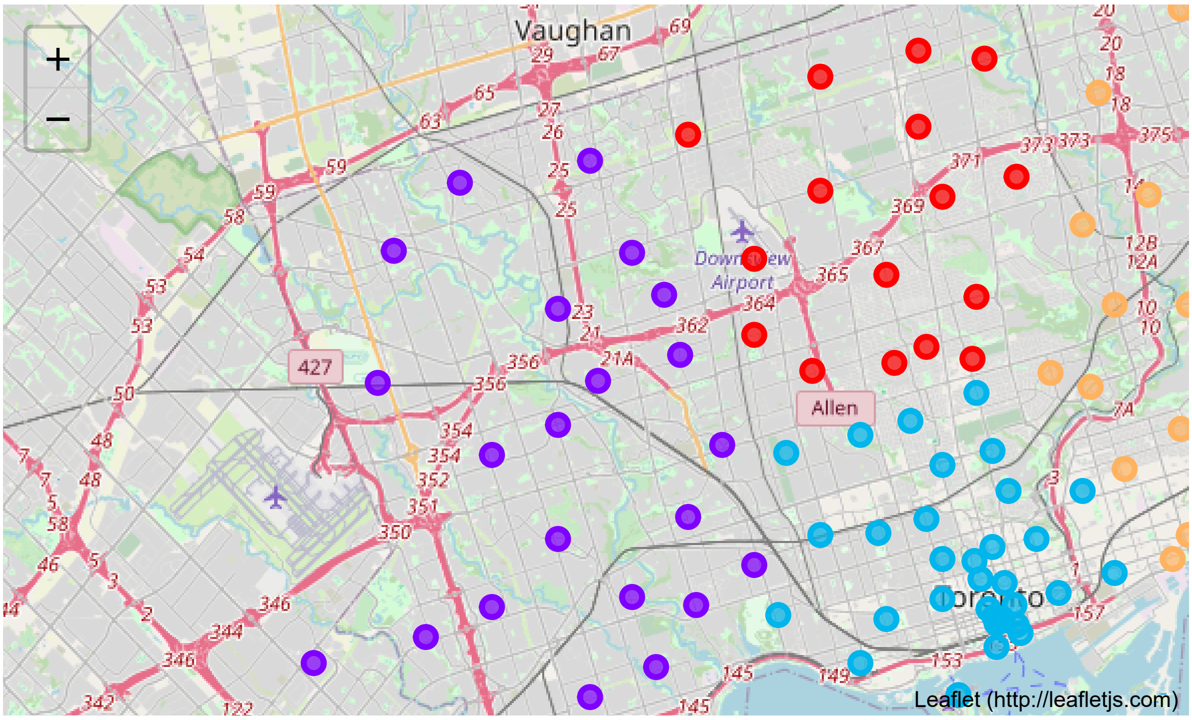
	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	4	M3A	North York	Parkwoods	43.753259	-79.329656
1	4	M4A	North York	Victoria Village	43.725882	-79.315572
2	2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	0	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	2	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
...	...	...	...	...	...	...
98	1	M8X	Etobicoke	The Kingsway, Montgomery Road, Old Mill North	43.653654	-79.506944
99	2	M4Y	Downtown Toronto	Church and Wellesley	43.665860	-79.383160
100	4	M7Y	East Toronto	Business reply mail Processing Centre, South C...	43.662744	-79.321558
101	1	M8Y	Etobicoke	Old Mill South, King's Mill Park, Sunnylea, Hu...	43.636258	-79.498509
102	1	M8Z	Etobicoke	Mimico NW, The Queensway West, South of Bloor,...	43.628841	-79.520999

103 rows × 6 columns

In [11]:

```
# let’s visualize the resulting clusters

import matplotlib.cm as cm
import matplotlib.colors as colors
import numpy as np

map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(k)
ys = [i + x + (i*x)**2 for i in range(k)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(df_toronto[’Latitude’], \
                                  df_toronto[’Longitude’], \
                                  df_toronto[’Neighborhood’], \
                                  df_toronto[’Cluster Labels’]):
    label = folium.Popup(str(poi) + ’ Cluster ’ + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[11]:



In [12]:

```
# Examine Clusters before explore any specific cluster
# cluster 1
df_toronto.loc[df_toronto['Cluster Labels'] == 0]
```

Out[12]:

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
3	0	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
10	0	M6B	North York	Glencairn	43.709577	-79.445073
27	0	M2H	North York	Hillcrest Village	43.803762	-79.363452
28	0	M3H	North York	Bathurst Manor, Wilson Heights, Downsview North	43.754328	-79.442259
34	0	M3J	North York	Northwood Park, York University	43.767980	-79.487262
39	0	M2K	North York	Bayview Village	43.786947	-79.385975
40	0	M3K	North York	Downsview	43.737473	-79.464763
45	0	M2L	North York	York Mills, Silver Hills	43.757490	-79.374714
52	0	M2M	North York	Willowdale, Newtonbrook	43.789053	-79.408493
55	0	M5M	North York	Bedford Park, Lawrence Manor East	43.733283	-79.419750
59	0	M2N	North York	Willowdale, Willowdale East	43.770120	-79.408493
61	0	M4N	Central Toronto	Lawrence Park	43.728020	-79.388790
62	0	M5N	Central Toronto	Roselawn	43.711695	-79.416936
66	0	M2P	North York	York Mills West	43.752758	-79.400049
67	0	M4P	Central Toronto	Davisville North	43.712751	-79.390197
72	0	M2R	North York	Willowdale, Willowdale West	43.782736	-79.442259
73	0	M4R	Central Toronto	North Toronto West, Lawrence Park	43.715383	-79.405678

In [13]:

```
# cluster 2
df_toronto.loc[df_toronto['Cluster Labels'] == 1]
```

Out[13]:

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
5	1	M9A	Etobicoke	Islington Avenue, Humber Valley Village	43.667856	-79.532242
11	1	M9B	Etobicoke	West Deane Park, Princess Gardens, Martin Grov...	43.650943	-79.554724
17	1	M9C	Etobicoke	Eringate, Bloordale Gardens, Old Burnhamthorpe...	43.643515	-79.577201
46	1	M3L	North York	Downsview	43.739015	-79.506944
49	1	M6L	North York	North Park, Maple Leaf Park, Upwood Park	43.713756	-79.490074
50	1	M9L	North York	Humber Summit	43.756303	-79.565963
53	1	M3M	North York	Downsview	43.728496	-79.495697
56	1	M6M	York	Del Ray, Mount Dennis, Keelsdale and Silverthorn	43.691116	-79.476013
57	1	M9M	North York	Humberlea, Emery	43.724766	-79.532242
60	1	M3N	North York	Downsview	43.761631	-79.520999
63	1	M6N	York	Runnymede, The Junction North	43.673185	-79.487262
64	1	M9N	York	Weston	43.706876	-79.518188
69	1	M6P	West Toronto	High Park, The Junction South	43.661608	-79.464763
70	1	M9P	Etobicoke	Westmount	43.696319	-79.532242
76	1	M7R	Mississauga	Canada Post Gateway Processing Centre	43.636966	-79.615819
77	1	M9R	Etobicoke	Kingsview Village, St. Phillips, Martin Grove ...	43.688905	-79.554724
81	1	M6S	West Toronto	Runnymede, Swansea	43.651571	-79.484450
88	1	M8V	Etobicoke	New Toronto, Mimico South, Humber Bay Shores	43.605647	-79.501321
89	1	M9V	Etobicoke	South Steeles, Silverstone, Humbergate, Jamest...	43.739416	-79.588437
93	1	M8W	Etobicoke	Alderwood, Long Branch	43.602414	-79.543484
94	1	M9W	Etobicoke	Northwest, West Humber - Clairville	43.706748	-79.594054
98	1	M8X	Etobicoke	The Kingsway, Montgomery Road, Old Mill North	43.653654	-79.506944
101	1	M8Y	Etobicoke	Old Mill South, King's Mill Park, Sunnylea, Hu...	43.636258	-79.498509
102	1	M8Z	Etobicoke	Mimico NW, The Queensway West, South of Bloor,...	43.628841	-79.520999

In [14]:

```
# cluster 3
df_toronto.loc[df_toronto['Cluster Labels'] == 2]
```

Out[14]:

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
2	2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
4	2	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
9	2	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937
15	2	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418
16	2	M6C	York	Humewood-Cedarvale	43.693781	-79.428191
20	2	M5E	Downtown Toronto	Berczy Park	43.644771	-79.373306
21	2	M6E	York	Caledonia-Fairbanks	43.689026	-79.453512
24	2	M5G	Downtown Toronto	Central Bay Street	43.657952	-79.387383
25	2	M6G	Downtown Toronto	Christie	43.669542	-79.422564
30	2	M5H	Downtown Toronto	Richmond, Adelaide, King	43.650571	-79.384568
31	2	M6H	West Toronto	Dufferin, Dovercourt Village	43.669005	-79.442259
36	2	M5J	Downtown Toronto	Harbourfront East, Union Station, Toronto Islands	43.640816	-79.381752
37	2	M6J	West Toronto	Little Portugal, Trinity	43.647927	-79.419750
41	2	M4K	East Toronto	The Danforth West, Riverdale	43.679557	-79.352188
42	2	M5K	Downtown Toronto	Toronto Dominion Centre, Design Exchange	43.647177	-79.381576
43	2	M6K	West Toronto	Brockton, Parkdale Village, Exhibition Place	43.636847	-79.428191
48	2	M5L	Downtown Toronto	Commerce Court, Victoria Hotel	43.648198	-79.379817
54	2	M4M	East Toronto	Studio District	43.659526	-79.340923
68	2	M5P	Central Toronto	Forest Hill North & West, Forest Hill Road Park	43.696948	-79.411307
74	2	M5R	Central Toronto	The Annex, North Midtown, Yorkville	43.672710	-79.405678
75	2	M6R	West Toronto	Parkdale, Roncesvalles	43.648960	-79.456325
79	2	M4S	Central Toronto	Davisville	43.704324	-79.388790
80	2	M5S	Downtown Toronto	University of Toronto, Harbord	43.662696	-79.400049

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
83	2	M4T	Central Toronto	Moore Park, Summerhill East	43.689574	-79.383160
84	2	M5T	Downtown Toronto	Kensington Market, Chinatown, Grange Park	43.653206	-79.400049
86	2	M4V	Central Toronto	Summerhill West, Rathnelly, South Hill, Forest...	43.686412	-79.400049
87	2	M5V	Downtown Toronto	CN Tower, King and Spadina, Railway Lands, Har...	43.628947	-79.394420
91	2	M4W	Downtown Toronto	Rosedale	43.679563	-79.377529
92	2	M5W	Downtown Toronto	Stn A PO Boxes	43.646435	-79.374846
96	2	M4X	Downtown Toronto	St. James Town, Cabbagetown	43.667967	-79.367675
97	2	M5X	Downtown Toronto	First Canadian Place, Underground city	43.648429	-79.382280
99	2	M4Y	Downtown Toronto	Church and Wellesley	43.665860	-79.383160

In [15]:

```
# cluster 4
df_toronto.loc[df_toronto['Cluster Labels'] == 3]
```

Out[15]:

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
6	3	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353
12	3	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497
18	3	M1E	Scarborough	Guildwood, Morningside, West Hill	43.763573	-79.188711
22	3	M1G	Scarborough	Woburn	43.770992	-79.216917
26	3	M1H	Scarborough	Cedarbrae	43.773136	-79.239476
32	3	M1J	Scarborough	Scarborough Village	43.744734	-79.239476
51	3	M1M	Scarborough	Cliffside, Cliffcrest, Scarborough Village West	43.716316	-79.239476
78	3	M1S	Scarborough	Agincourt	43.794200	-79.262029
85	3	M1V	Scarborough	Milliken, Agincourt North, Steeles East, L'Amo...	43.815252	-79.284577
95	3	M1X	Scarborough	Upper Rouge	43.836125	-79.205636

In [16]:

```
# cluster 5
df_toronto.loc[df_toronto['Cluster Labels'] == 4]
```

Out[16]:

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
0	4	M3A	North York	Parkwoods	43.753259	-79.329656
1	4	M4A	North York	Victoria Village	43.725882	-79.315572
7	4	M3B	North York	Don Mills	43.745906	-79.352188
8	4	M4B	East York	Parkview Hill, Woodbine Gardens	43.706397	-79.309937
13	4	M3C	North York	Don Mills	43.725900	-79.340923
14	4	M4C	East York	Woodbine Heights	43.695344	-79.318389
19	4	M4E	East Toronto	The Beaches	43.676357	-79.293031
23	4	M4G	East York	Leaside	43.709060	-79.363452
29	4	M4H	East York	Thorncliffe Park	43.705369	-79.349372
33	4	M2J	North York	Fairview, Henry Farm, Oriole	43.778517	-79.346556
35	4	M4J	East York	East Toronto, Broadview North (Old East York)	43.685347	-79.338106
38	4	M1K	Scarborough	Kennedy Park, Ionview, East Birchmount Park	43.727929	-79.262029
44	4	M1L	Scarborough	Golden Mile, Clairlea, Oakridge	43.711112	-79.284577
47	4	M4L	East Toronto	India Bazaar, The Beaches West	43.668999	-79.315572
58	4	M1N	Scarborough	Birch Cliff, Cliffside West	43.692657	-79.264848
65	4	M1P	Scarborough	Dorset Park, Wexford Heights, Scarborough Town...	43.757410	-79.273304
71	4	M1R	Scarborough	Wexford, Maryvale	43.750072	-79.295849
82	4	M1T	Scarborough	Clarks Corners, Tam O'Shanter, Sullivan	43.781638	-79.304302
90	4	M1W	Scarborough	Steeles West, L'Amoreaux West	43.799525	-79.318389
100	4	M7Y	East Toronto	Business reply mail Processing Centre, South C...	43.662744	-79.321558

In [17]:

```
# Then, we have five clusters. I pick up one of them to explore a little bit more.
# Here, I choose the cluster 3, which is mostly locating within the downtown of Toronto (central Tor
# first of all, we need to extract all of the neighborhoods within cluster 3 into a new dataframe
df_toronto_denc = df_toronto.loc[df_toronto['Cluster Labels'] == 2]
df_toronto_denc.head()
```

Out[17]:

	Cluster Labels	PostalCode	Borough	Neighborhood	Latitude	Longitude
2	2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
4	2	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
9	2	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937
15	2	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418
16	2	M6C	York	Humewood-Cedarvale	43.693781	-79.428191

In [22]:

```
# before we go further using the FourSquare API, we need to set up the keys, limits and other premet

CLIENT_ID = 'VESQMT240RUAHFDKAC5LSEURA44BOFYJD4VUG3SL351BRIKG'
CLIENT_SECRET = 'XQF5ZD4KTW4AIDHCPDWFWHRHRLZ2OAPWFCOTQZYLS5T5Q14P'
VERSION = '20180604'

LIMIT = 100
radius = 500
```

In [19]:

```
# I copied and pasted the function from the teaching materials of this course
# to repeat the process of extracting venues' information

def getNearbyVenues(names, latitudes, longitudes, radius=500):
    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={
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

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

        # 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)
```

In [23]:

```
# run the above function and create a dataframe to store the results
toronto_denc_venues = getNearbyVenues(names=df_toronto_denc['Neighborhood'],
                                   latitudes=df_toronto_denc['Latitude'],
                                   longitudes=df_toronto_denc['Longitude']
                                  )
```

In [26]:

```
# have a quick look at the results
toronto_denc_venues.head()
```

Out[26]:

	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	Cooper Koo Family YMCA	43.653249	-79.358008	Distribution Center
3	Regent Park, Harbourfront	43.65426	-79.360636	Impact Kitchen	43.656369	-79.356980	Restaurant
4	Regent Park, Harbourfront	43.65426	-79.360636	Body Blitz Spa East	43.654735	-79.359874	Spa

In [27]:

```
print('There are {} uniques categories.'.format(len(toronto_denc_venues['Venue Category'].unique())))
```

```
There are 231 uniques categories.
```

Analyze Each Neighborhood

In [28]:

```
# one hot encoding
toronto_denc_onehot = pd.get_dummies(toronto_denc_venues[['Venue Category']], prefix="", prefix_sep=

# add neighborhood column back to dataframe
toronto_denc_onehot['Neighborhood'] = toronto_denc_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [toronto_denc_onehot.columns[-1]] + list(toronto_denc_onehot.columns[:-1])
toronto_denc_onehot = toronto_denc_onehot[fixed_columns]

toronto_denc_onehot.head()
```

Out[28]:

	Yoga Studio	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0

5 rows × 231 columns

In [31]:

```
# examine the new dataframe size
toronto_denc_onehot.shape
```

Out[31]:

```
(1515, 231)
```

In [32]:

```
# group rows by neighborhood and by taking the mean of the frequency of occurrence of each category
toronto_denc_grouped = toronto_denc_onehot.groupby(’Neighborhood’).mean().reset_index()
toronto_denc_grouped.head()
```

Out[32]:

	Neighborhood	Yoga Studio	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Ai Tern
0	Berczy Park	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
1	Brockton, Parkdale Village, Exhibition Place	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
2	CN Tower, King and Spadina, Railway Lands, Har...	0.000000	0.0	0.058824	0.058824	0.058824	0.117647	0.117647	0.05
3	Caledonia-Fairbanks	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
4	Central Bay Street	0.014706	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.00

5 rows × 231 columns

In [33]:

```
# examine the new dataframe size
toronto_denc_grouped.shape
```

Out[33]:

```
(32, 231)
```

In [34]:

```
# Let's print each neighborhood along with the top 5 most common venues

num_top_venues = 5

for hood in toronto_denc_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = toronto_denc_grouped[toronto_denc_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Berczy Park----
                venue  freq
0         Coffee Shop  0.09
1        Cocktail Bar  0.04
2          Restaurant  0.04
3  Seafood Restaurant  0.04
4         Cheese Shop  0.04


----Brockton, Parkdale Village, Exhibition Place----
                   venue  freq
0                   Café  0.12
1         Breakfast Spot  0.08
2            Coffee Shop  0.08
3              Nightclub  0.08
4  Performing Arts Venue  0.04


----CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, S
```

In [35]:

```
# create the new dataframe and display the top 10 venues for each neighborhood

def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)
    return row_categories_sorted.index.values[0:num_top_venues]

num_top_venues = 10

indicators = [’st’, ’nd’, ’rd’]

# create columns according to number of top venues
columns = [’Neighborhood’]
for ind in np.arange(num_top_venues):
    try:
        columns.append(’{}{} Most Common Venue’.format(ind+1, indicators[ind]))
    except:
        columns.append(’{}th Most Common Venue’.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted[’Neighborhood’] = toronto_denc_grouped[’Neighborhood’]

for ind in np.arange(toronto_denc_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(toronto_denc_grouped.iloc[

neighborhoods_venues_sorted.head()
```

Out[35]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue
0	Berczy Park	Coffee Shop	Farmers Market	Bakery	Cocktail Bar	Beer Bar	Seafood Restaurant	Cheese Shop
1	Brockton, Parkdale Village, Exhibition Place	Café	Nightclub	Coffee Shop	Breakfast Spot	Bakery	Convenience Store	Performing Arts Venue
2	CN Tower, King and Spadina, Railway Lands, Har...	Airport Lounge	Airport Service	Plane	Bar	Rental Car Location	Boat or Ferry	Boutique
3	Caledonia-Fairbanks	Park	Women's Store	Pool	Dessert Shop	Electronics Store	Eastern European Restaurant	Dumpling Restaurant
4	Central Bay Street	Coffee Shop	Café	Sandwich Place	Italian Restaurant	Salad Place	Thai Restaurant	Department Store

In [ ]:

In [ ]: