Segmenting and Clustering Neighborhoods in Toronto - Part 3

For this assignment, you will be required to explore and cluster the neighborhoods in Toronto.

Step 1: Load the data to a dataframe

In [5]:
```
#Load necessary libraries
import pandas as pd
import requests
```

In [6]:
```
from bs4 import BeautifulSoup
url="https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
soup = BeautifulSoup(requests.get(url).text,'lxml')
My_table = soup.find('table',{'class':'wikitable sortable'})

My_table
table_rows = My_table.find_all('tr')

t=[]
for tr in table_rows:
    td = tr.find_all('td')
    row = [tr.text.rstrip('\n') for tr in td]
    t.append(row)
df=pd.DataFrame(t)

df.columns=['Postcode','Borough','Neighbourhood']

df.drop([0],axis=0,inplace=True)

df.reset_index()

df.head()
```

Out[6]:

	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

A different way to load the table

▶ Click here to see...

Step 2. Process the data

2.1. Ignore rows with a Borough that is "Not assigned"

In [7]:
```
df.drop(df[df['Borough'] == 'Not assigned'].index, inplace = True)

#re-index the dataframe
df = df.reset_index(drop=True)

df.head()
```

Out[7]:

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

2.2. Combine neighborhoods that has the same postcode

In [8]:
```
df_new = df.groupby("Postcode").agg(lambda x:', '.join(set(x)))
df_new = df_new.reset_index()
df_new.loc[51:55]
```

Out[8]:

	Postcode	Borough	Neighbourhood
51	M4X	Downtown Toronto	St. James Town, Cabbagetown
52	M4Y	Downtown Toronto	Church and Wellesley
53	M5A	Downtown Toronto	Harbourfront, Regent Park
54	M5B	Downtown Toronto	Ryerson, Garden District
55	M5C	Downtown Toronto	St. James Town

2.3. Name "Not assigned" neighbourhood after the Borough

In [9]:
```
for index, row in df_new.iterrows():
    if df_new.loc[index, 'Neighbourhood'] == "Not assigned":
        df_new.loc[index, 'Neighbourhood'] = df_new.loc[index, 'Borough']
        print(index)
```

```
85
```

In [10]:
```
df_new.shape
```

Out[10]:
```
(103, 3)
```

Step 3. Load and include the geospatial data

In [11]:
```
g_data = pd.read_csv("https://cocl.us/Geospatial_data")
g_data.head()
```

Out[11]:

	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 [12]:
```
df_g = df_new
df_g['Latitude'] = g_data['Latitude'].values
df_g['Longitude'] = g_data['Longitude'].values
df_g.head()
```

Out[12]:

	Postcode	Borough	Neighbourhood	Latitude	Longitude
0	M1B	Scarborough	Rouge, Malvern	43.806686	-79.194353
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497
2	M1E	Scarborough	Guildwood, West Hill, Morningside	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

In [62]:
```
df_g.shape
```

Out[62]:
```
(103, 5)
```

Result of Part 3: Visualize the data on map

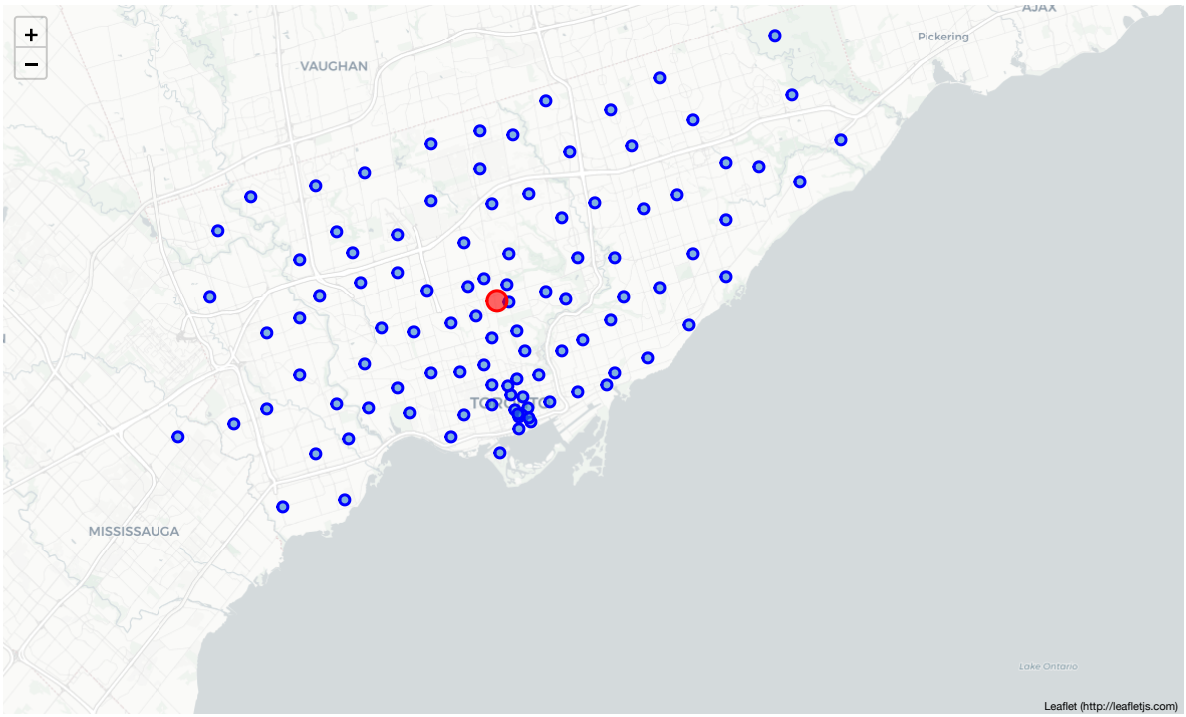
In [30]:
```
!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library

print('Folium installed')
```

```
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

Folium installed
```

Vis 1. Show all the Boroughs and Neighbourhoods

In [80]:
```
lat_center = df_g['Latitude'].mean()
lon_center = df_g['Longitude'].mean()
#print("The center of the map is: (x: %s, y: %s)" % (lat_center, lon_center))

#Draw the map
map_all = folium.Map(location=[lat_center, lon_center], tiles='cartodbpositron', zoom_start=11)

# add markers to map
for lat, lng, borough, neighborhood, postcode in zip(df_g['Latitude'], df_g['Longitude'], df_g['Borough'], df_g['Neighbourhood'], df_g['Postc
ode']):
    label = '{}, {}, {}'.format(neighborhood, borough, postcode)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.6).add_to(map_all)

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [lat_center, lon_center],
    radius=10,
    color='red',
    popup='Center of the map',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(map_all)

# display map
map_all
```

Out[80]:



Vis 2: Show all Boroughs that contains the word "Toronto"

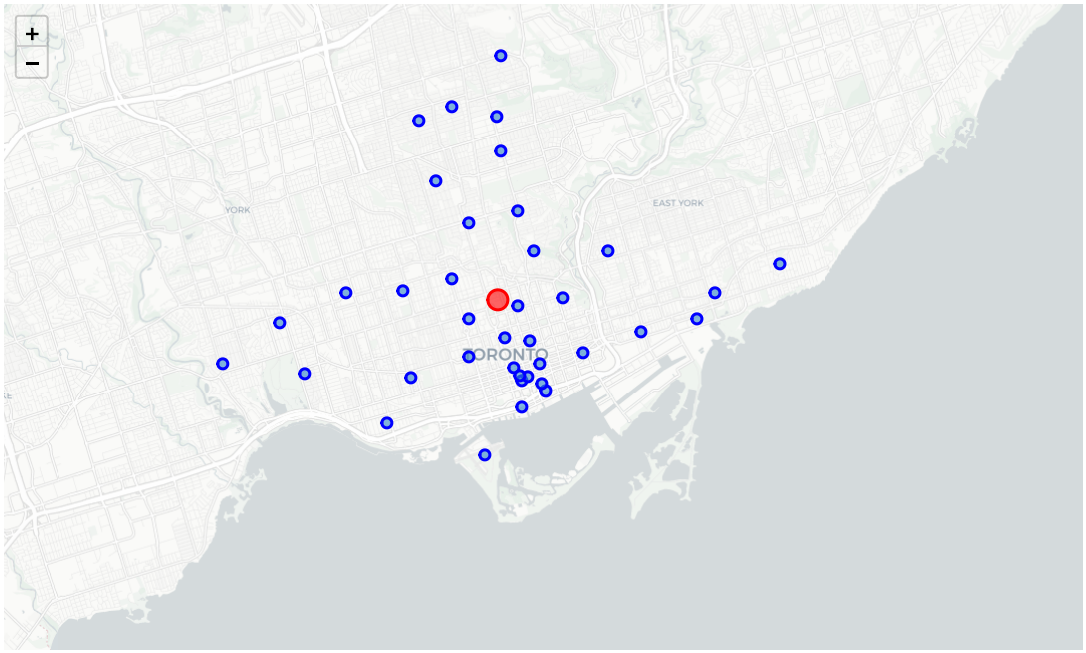
In [77]:
```
df_toronto = df_g.set_index('Borough').filter(like='Toronto', axis=0)
df_toronto = df_toronto.reset_index()
df_toronto.head()
```

Out[77]:

	Borough	Postcode	Neighbourhood	Latitude	Longitude
0	East Toronto	M4E	The Beaches	43.676357	-79.293031
1	East Toronto	M4K	Riverdale, The Danforth West	43.679557	-79.352188
2	East Toronto	M4L	India Bazaar, The Beaches West	43.668999	-79.315572
3	East Toronto	M4M	Studio District	43.659526	-79.340923
4	Central Toronto	M4N	Lawrence Park	43.728020	-79.388790

In [84]:
```
lat_center_t = df_toronto['Latitude'].mean()
lon_center_t = df_toronto['Longitude'].mean()

#Draw the map
map_toronto = folium.Map(location=[lat_center_t, lon_center_t], tiles='cartodbpositron', zoom_start=12)

# add markers to map
for lat, lng, borough, neighborhood, postcode in zip(df_toronto['Latitude'], df_toronto['Longitude'], df_toronto['Borough'], df_toronto['Neig
hbourhood'], df_toronto['Postcode']):
    label = '{}, {}, {}'.format(neighborhood, borough, postcode)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.6).add_to(map_toronto)

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [lat_center_t, lon_center_t],
    radius=10,
    color='red',
    popup='Center of the map',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(map_toronto)

# display map
map_toronto
```

Out[84]:



Leaflet (http://leafletjs.com)

Vis 3: Show all the Boroughs

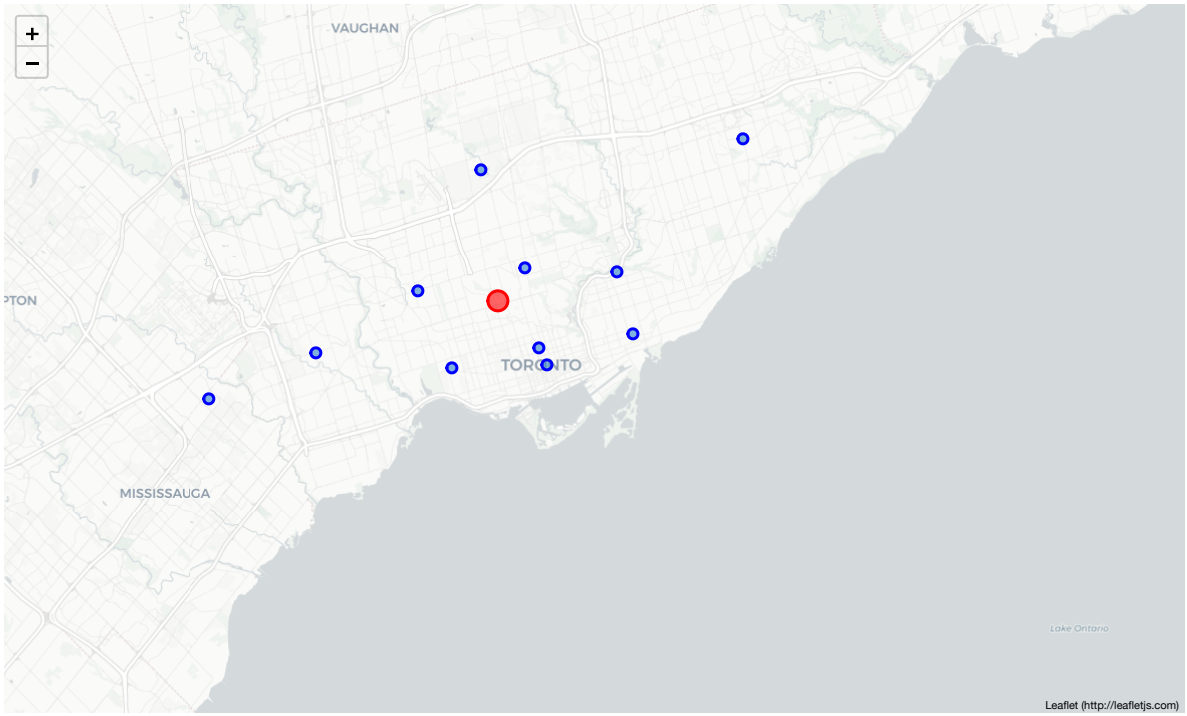
In [96]:
```
df_borough = df_g.groupby(['Borough'],as_index=False).mean()
df_borough = df_borough.reset_index()
df_borough.head()
```

Out[96]:

	index	Borough	Latitude	Longitude
0	0	Central Toronto	43.701980	-79.398954
1	1	Downtown Toronto	43.654169	-79.383665
2	2	East Toronto	43.669436	-79.324654
3	3	East York	43.700303	-79.335851
4	4	Etobicoke	43.660043	-79.542074

In [100]:
```
lat_center_b = df_borough['Latitude'].mean()
lon_center_b = df_borough['Longitude'].mean()

#Draw the map
map_borough = folium.Map(location=[lat_center_b, lon_center_b], tiles='cartodbpositron', zoom_start=11)

# add markers to map
for lat, lng, borough in zip(df_borough['Latitude'], df_borough['Longitude'], df_borough['Borough']):
    label = 'Center of {}'.format(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=0.6).add_to(map_borough)

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [lat_center_b, lon_center_b],
    radius=10,
    color='red',
    popup='Center of the map',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(map_borough)

# display map
map_borough
```

Out[100]:



In [ ]: