Segmenting and Clustering Neighborhoods in Toronto

In [1]:

```
import pandas as pd
```

In [2]:

```
from bs4 import BeautifulSoup
```

In [3]:

```
import requests

url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'

r = requests.get(url)

soup = BeautifulSoup(r.content)
```

In [4]:

```

table = soup.find('table')
df = pd.read_html(str(table))[0]
```

In [5]:

```
df.head()
```

Out[5]:

	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

In [6]:

```
# remove rows whos Borough value is Not assigned
df = df.drop(df[df.Borough == "Not assigned"].index)
```

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103 entries, 2 to 178
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Postal Code    103 non-null    object
 1   Borough        103 non-null    object
 2   Neighbourhood  103 non-null    object
dtypes: object(3)
memory usage: 3.2+ KB
```

In [8]:

```
df.head(10)
```

Out[8]:

	Postal Code	Borough	Neighbourhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
5	M6A	North York	Lawrence Manor, Lawrence Heights
6	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government
8	M9A	Etobicoke	Islington Avenue, Humber Valley Village
9	M1B	Scarborough	Malvern, Rouge
11	M3B	North York	Don Mills
12	M4B	East York	Parkview Hill, Woodbine Gardens
13	M5B	Downtown Toronto	Garden District, Ryerson

In [9]:

```
# check any neighbourhood has value Not assigned.
df[df.Neighbourhood == "Not assigned"].count()
```

Out[9]:

```
Postal Code      0
Borough          0
Neighbourhood    0
dtype: int64
```

In [10]:

```
# Show data frame Shape
df.shape
```

Out[10]:

```
(103, 3)
```

In [11]:

```
#import geocoder # 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(postal_code))
#  lat_lng_coords = g.latlng

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

In [12]:
```
#Load geographical coordinates
geo_df = pd.read_csv('Geospatial_Coordinates.csv')
geo_df.head()
```

Out[12]:

	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 [13]:
```
# Merge Neighbourhood dataframe with geographical coordinates
dfall = pd.merge(df, geo_df, on=['Postal Code','Postal Code'])
```

In [14]:
```
dfall.head()
```

Out[14]:

	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

In [15]:
```
#get only Toronto borough
dfToronto = dfall[dfall['Borough'].str.contains("Toronto")]

dfToronto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 39 entries, 2 to 100
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Postal Code   39 non-null     object 
 1   Borough       39 non-null     object 
 2   Neighbourhood 39 non-null     object 
 3   Latitude      39 non-null     float64
 4   Longitude     39 non-null     float64
dtypes: float64(2), object(3)
memory usage: 1.8+ KB
```

In [16]:
```
dfToronto.head()
```

Out[16]:

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
9	M5B	Downtown Toronto	Garden District, Ryerson	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

In [17]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(dfToronto['Borough'].unique()),
        dfToronto.shape[0]
    )
)
```

```
The dataframe has 4 boroughs and 39 neighborhoods.
```

In [20]:
```
from geopy.geocoders import Nominatim
address = 'Toronto'

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

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

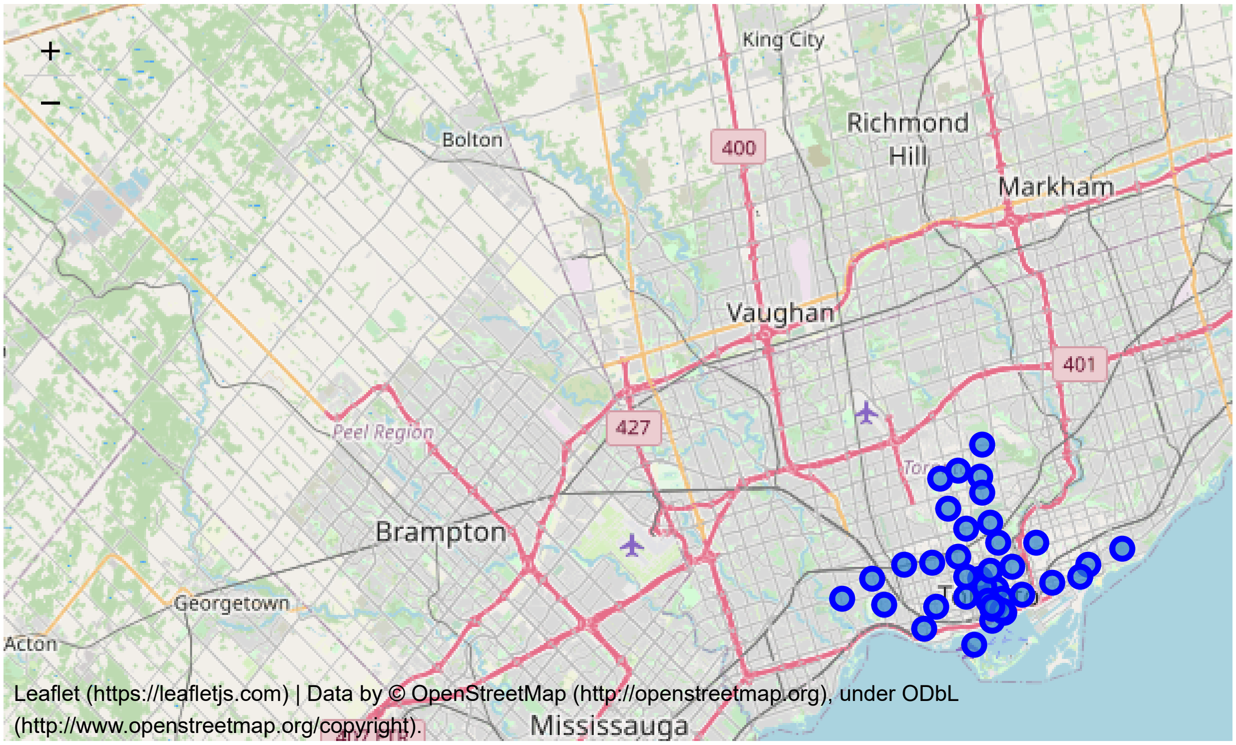
In [25]:

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

# add markers to map
for lat, lng, borough, neighborhood in zip(dfToronto['Latitude'], dfToronto['Long
    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[25]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [ ]: