

In [5]:

```
import numpy as np # Library to handle data in a vectorized manner

import pandas as pd # Library for data analysis
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 th
from geopy.geocoders import Nominatim # convert an address into latitude and longitude valu

import requests # Library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas 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

# for webscraping import Beautiful Soup
from bs4 import BeautifulSoup

import xml

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't compl
import folium # map rendering library

print('Libraries imported.')
```

```
usage: conda-script.py [-h] [-V] command ...
conda-script.py: error: unrecognized arguments: # uncomment this line if you
haven't completed the Foursquare API lab
```

Libraries imported.

```
usage: conda-script.py [-h] [-V] command ...
conda-script.py: error: unrecognized arguments: # uncomment this line if you
haven't completed the Foursquare API lab
```

In [6]:

```
url = requests.get('https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M').text
soup = BeautifulSoup(url, 'lxml')
```

In [7]:

```
table_post = soup.find('table')
fields = table_post.find_all('td')

postcode = []
borough = []
neighbourhood = []

for i in range(0, len(fields), 3):
    postcode.append(fields[i].text.strip())
    borough.append(fields[i+1].text.strip())
    neighbourhood.append(fields[i+2].text.strip())

df_pc = pd.DataFrame(data=[postcode, borough, neighbourhood]).transpose()
df_pc.columns = ['Postcode', 'Borough', 'Neighbourhood']
df_pc.head()
```

Out[7]:

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

In [8]:

```
df_pc.head()
```

Out[8]:

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

In [9]:

```
df_pc['Borough'].replace('Not assigned', np.nan, inplace=True)
df_pc.dropna(subset=['Borough'], inplace=True)

df_pc.head()
```

Out[9]:

	Postcode	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

In [10]:

```
df_pc.head()
```

Out[10]:

	Postcode	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

In [11]:

```
df_pcn = df_pc.groupby(['Postcode', 'Borough'])['Neighbourhood'].apply(', '.join).reset_index
df_pcn.columns = ['Postcode', 'Borough', 'Neighbourhood']
df_pcn
```

86	M7R	Mississauga	Canada Post Gateway Processing Centre
87	M7Y	East Toronto	Business reply mail Processing Centre
88	M8V	Etobicoke	New Toronto / Mimico South / Humber Bay Shores
89	M8W	Etobicoke	Alderwood / Long Branch
90	M8X	Etobicoke	The Kingsway / Montgomery Road / Old Mill North
91	M8Y	Etobicoke	Old Mill South / King's Mill Park / Sunnylea / ...
92	M8Z	Etobicoke	Mimico NW / The Queensway West / South of Bloo...
93	M9A	Etobicoke	Islington Avenue
94	M9B	Etobicoke	West Deane Park / Princess Gardens / Martin Gr...
95	M9C	Etobicoke	Eringate / Bloordale Gardens / Old Burnhamthor...
96	M9L	North York	Humber Summit
97	M9M	North York	Humberlea / Emery
98	M9N	York	Weston

In [12]:

```
df_pcn['Neighbourhood'].replace('Not assigned', "Queen's Park", inplace=True)
df_pcn
```

Out[12]:

	Postcode	Borough	Neighbourhood
0	M1B	Scarborough	Malvern / Rouge
1	M1C	Scarborough	Rouge Hill / Port Union / Highland Creek
2	M1E	Scarborough	Guildwood / Morningside / West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae
5	M1J	Scarborough	Scarborough Village
6	M1K	Scarborough	Kennedy Park / Ionview / East Birchmount Park
7	M1L	Scarborough	Golden Mile / Clairlea / Oakridge
8	M1M	Scarborough	Cliffside / Cliffcrest / Scarborough Village West
9	M1N	Scarborough	Birch Cliff / Cliffside West

In [13]:

```
df_pcn.shape
```

Out[13]:

(103, 3)

In [14]:

```
df_geo = pd.read_csv('http://coc1.us/Geospatial_data')
df_geo.columns = ['Postcode', 'Latitude', 'Longitude']
```

In [15]:

```
df_pos = pd.merge(df_pcn, df_geo, on=['Postcode'], how='inner')
df_tor = df_pos[['Borough', 'Neighbourhood', 'Postcode', 'Latitude', 'Longitude']].copy()
df_tor.head()
```

Out[15]:

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

In [16]:

```
address = 'Toronto, Canada'

geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of the City of Toronto are {}, {}'.format(latitude, long
```

C:\Users\GANTAYASHWANTHINI\anaconda3\lib\site-packages\ipykernel\_launcher.p  
y:3: DeprecationWarning: Using Nominatim with the default "geopy/1.21.0" `us  
er\_agent` is strongly discouraged, as it violates Nominatim's ToS [https://op  
erations.osmfoundation.org/policies/nominatim/](https://operations.osmfoundation.org/policies/nominatim/) ([https://operations.osmfounda  
tion.org/policies/nominatim/](https://operations.osmfoundation.org/policies/nominatim/)) and may possibly cause 403 and 429 HTTP error  
s. Please specify a custom `user\_agent` with `Nominatim(user\_agent="my-appli  
cation")` or by overriding the default `user\_agent`: `geopy.geocoders.option  
s.default\_user\_agent = "my-application"`. In geopy 2.0 this will become an e  
xception.

This is separate from the ipykernel package so we can avoid doing imports  
until

The geograpical coordinate of the City of Toronto are 43.6534817, -79.383934  
7.

In [17]:

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

# add markers to map
for lat, lng, borough, neighborhood in zip(df_tor['Latitude'], df_tor['Longitude'], df_tor[
    label = '{} , {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=3,
        popup=label,
        color='green',
        fill=True,
        fill_color='#3199cc',
        fill_opacity=0.3,
        parse_html=False).add_to(map_toronto)

map_toronto
```

Out[17]:



In [ ]: