Part 1

Build a dataframe of the postal code of each neighborhood along with the borough name and neighborhood name in Toronto.

In [6]:

```
pip install geopy
```

```
Requirement already satisfied: geopy in /home/jupyterlab/conda/envs/python/li
b/python3.6/site-packages (1.20.0)
Requirement already satisfied: geographiclib<2,>=1.49 in /home/jupyterlab/con
da/envs/python/lib/python3.6/site-packages (from geopy) (1.50)
Note: you may need to restart the kernel to use updated packages.
```

In [7]:

```
pip install bs4
```

```
Collecting bs4
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f45617
4139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Collecting beautifulsoup4 (from bs4)
  Downloading https://files.pythonhosted.org/packages/cb/a1/c698cf319e9cfed6b
17376281bd0efc6bfc8465698f54170ef60a485ab5d/beautifulsoup4-4.8.2-py3-none-an
y.whl (106kB)
     |████████████████████████████████| 112kB 6.6MB/s eta 0:00:01
Collecting soupsieve>=1.2 (from beautifulsoup4->bs4)
  Downloading https://files.pythonhosted.org/packages/81/94/03c0f04471fc245d0
8d0a99f7946ac228ca98da4fa75796c507f61e688c2/soupsieve-1.9.5-py2.py3-none-any.
whl
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) ... done
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/a0/b0/b2/4f80b9456b
87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: soupsieve, beautifulsoup4, bs4
Successfully installed beautifulsoup4-4.8.2 bs4-0.0.1 soupsieve-1.9.5
Note: you may need to restart the kernel to use updated packages.
```

In [8]:

```
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

from geopy.geocoders import Nominatim # convert an address into latitude and longitude val
ues

import requests # library to handle requests
from bs4 import BeautifulSoup # library to parse HTML and XML documents

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

import folium # map rendering library

print("Libraries imported.")
```

```
Libraries imported.
```

2. Scrap data from Wikipedia page into a DataFrame¶

In [9]:

```
# send the GET request
data = requests.get('https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M').tex
t
```

In [10]:

```
# parse data from the html into a beautifulsoup object
soup = BeautifulSoup(data, 'html.parser')
```

In [11]:

```
# create three lists to store table data
postalCodeList = []
boroughList = []
neighborhoodList = []
```

Using BeautifulSoup

In [12]:

```
# find the table
soup.find('table').find_all('tr')

# find all the rows of the table
soup.find('table').find_all('tr')

# for each row of the table, find all the table data
for row in soup.find('table').find_all('tr'):
    cells = row.find_all('td')
```

In [13]:

```
# append the data into the respective lists
for row in soup.find('table').find_all('tr'):
    cells = row.find_all('td')
    if(len(cells) > 0):
        postalCodeList.append(cells[0].text)
        boroughList.append(cells[1].text)
        neighborhoodList.append(cells[2].text.rstrip('\n')) # avoid new lines in neighborh
ood cell
```

In [14]:

```
# create a new DataFrame from the three lists
toronto_df = pd.DataFrame({"PostalCode": postalCodeList,
                           "Borough": boroughList,
                           "Neighborhood": neighborhoodList})

toronto_df.head()
```

Out[14]:

	PostalCode	Borough	Neighborhood
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	Harbourfront

3. Drop cells with a borough that is "Not assigned"¶

In [15]:

```
# drop cells with a borough that is Not assigned
toronto_df_dropna = toronto_df[toronto_df.Borough != "Not assigned"].reset_index(drop=True
)
toronto_df_dropna.head()
```

Out[15]:

	PostalCode	Borough	Neighborhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Harbourfront
3	M6A	North York	Lawrence Heights
4	M6A	North York	Lawrence Manor

4. Group neighborhoods in the same borough¶

In [17]:

```
# group neighborhoods in the same borough
toronto_df_grouped = toronto_df_dropna.groupby(["PostalCode", "Borough"], as_index=False).
agg(lambda x: ", ".join(x))
toronto_df_grouped.head()
```

Out[17]:

	PostalCode	Borough	Neighborhood
0	M1B	Scarborough	Rouge, Malvern
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

5. For Neighborhood="Not assigned", make the value the same as Borough¶

In [18]:

```
# for Neighborhood="Not assigned", make the value the same as Borough
for index, row in toronto_df_grouped.iterrows():
    if row["Neighborhood"] == "Not assigned":
        row["Neighborhood"] = row["Borough"]

toronto_df_grouped.head()
```

Out[18]:

	PostalCode	Borough	Neighborhood
0	M1B	Scarborough	Rouge, Malvern
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

6. Check whether it is the same as required by the question¶

In [19]:

```
# create a new test dataframe
column_names = ["PostalCode", "Borough", "Neighborhood"]
test_df = pd.DataFrame(columns=column_names)

test_list = ["M5G", "M2H", "M4B", "M1J", "M4G", "M4M", "M1R", "M9V", "M9L", "M5V", "M1B",
"M5A"]

for postcode in test_list:
    test_df = test_df.append(toronto_df_grouped[toronto_df_grouped["PostalCode"]==postcode
], ignore_index=True)

test_df
```

Out[19]:

	PostalCode	Borough	Neighborhood
0	M5G	Downtown Toronto	Central Bay Street
1	M2H	North York	Hillcrest Village
2	M4B	East York	Woodbine Gardens, Parkview Hill
3	M1J	Scarborough	Scarborough Village
4	M4G	East York	Leaside
5	M4M	East Toronto	Studio District
6	M1R	Scarborough	Maryvale, Wexford
7	M9V	Etobicoke	Albion Gardens, Beaumond Heights, Humbergate, ...
8	M9L	North York	Humber Summit
9	M5V	Downtown Toronto	CN Tower, Bathurst Quay, Island airport, Harbo...
10	M1B	Scarborough	Rouge, Malvern
11	M5A	Downtown Toronto	Harbourfront

7. Finally, print the number of rows of the cleaned dataframe¶

In [20]:

```
# print the number of rows of the cleaned dataframe
toronto_df_grouped.shape
```

Out[20]:

```
(103, 3)
```

In [ ]:

```

```