In [2]:

```
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

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

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

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.5.11
  latest version: 4.7.12

Please update conda by running

    $ conda update -n base -c defaults conda



# All requested packages already installed.

Libraries imported.
```

In [4]:

```
pip install lxml
```

```
Collecting lxml
  Downloading https://files.pythonhosted.org/packages/ec/be/5ab8abdd8663c0386
ec2dd595a5bc0e23330a0549b8a91e32f38c20845b6/lxml-4.4.1-cp36-cp36m-manylinux1_
x86_64.whl (5.8MB)
     |████████████████████████████████| 5.8MB 25.6MB/s eta 0:00:01
Installing collected packages: lxml
Successfully installed lxml-4.4.1
Note: you may need to restart the kernel to use updated packages.
```

In [19]:

```
#Fetching the Data from the webpage
url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
toronto_data = pd.read_html(url, header=0)
```

In [20]:

```
#Converting the data into a pandas dataframe
toronto_data = toronto_data[0]
toronto_data.head()
```

Out[20]:

	Postcode	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	Harbourfront

In [25]:

```
#Dropping the Boroughs which are 'Not Assigned'
neighborhood_data = toronto_data[toronto_data.Borough != 'Not assigned']
neighborhood_data.reset_index(drop = True, inplace = True)
neighborhood_data.head()
```

Out[25]:

	Postcode	Borough	Neighbourhood
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

In place of the above code, we can also use the following neighborhood_data = toronto_data.set_index("Borough") neighborhood_data.drop("Not assigned") neighborhood_data.reset_index(drop = True, inplace = True) neighborhood_data.head() #to view the first five observations

In [33]:

```
#Combining the neighbourhoods of different coastal areas in one row seperated by comma
neighborhood_data = neighborhood_data.groupby(['Postcode','Borough'])['Neighbourhood'].app
ly(','.join)
neighborhood_data = neighborhood_data.reset_index(level = ['Postcode','Borough'])
neighborhood_data.head()
```

Out[33]:

	Postcode	Borough	Neighbourhood
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

In [39]:

```
#Here, we want to check the Neighbourhoods with the value 'Not assigned'
neighborhood_data[neighborhood_data.Neighbourhood == 'Not assigned']
```

Out[39]:

	Postcode	Borough	Neighbourhood
85	M7A	Queen's Park	Not assigned

In [41]:

```
#For the Neighbourhood with the value 'Not Assigned', we name the Neighbourhood with it's
 'Borough' name
#and then check is there is still any Neighbourhood with a 'Not assigned' observation.
#None exist anymore
neighborhood_data.loc[(neighborhood_data.Neighbourhood =='Not assigned'), 'Neighbourhood']
= neighborhood_data.Borough
neighborhood_data[neighborhood_data.Neighbourhood == 'Not assigned']
```

Out[41]:

Postcode	Borough	Neighbourhood

In [42]:

```
#This code the verifies if the Neighbourhood observation of 'Not assigned' on Postcode 'M7
A' has been
#Replaced by the Borough name 'Queen's Park
neighborhood_data[neighborhood_data.Postcode == 'M7A']
```

Out[42]:

	Postcode	Borough	Neighbourhood
85	M7A	Queen's Park	Queen's Park

In [44]:

```
#The shape of the data is 103 observations and 3 Variables
neighborhood_data.shape
```

Out[44]:

```
(103, 3)
```

In [ ]:

```

```