In this assignment, you will be required to explore, segment, and cluster the neighborhoods in the city of Toronto. However, unlike New York, the neighborhood data is not readily available on the internet. What is interesting about the field of data science is that each project can be challenging in its unique way, so you need to learn to be agile and refine the skill to learn new libraries and tools quickly depending on the project.

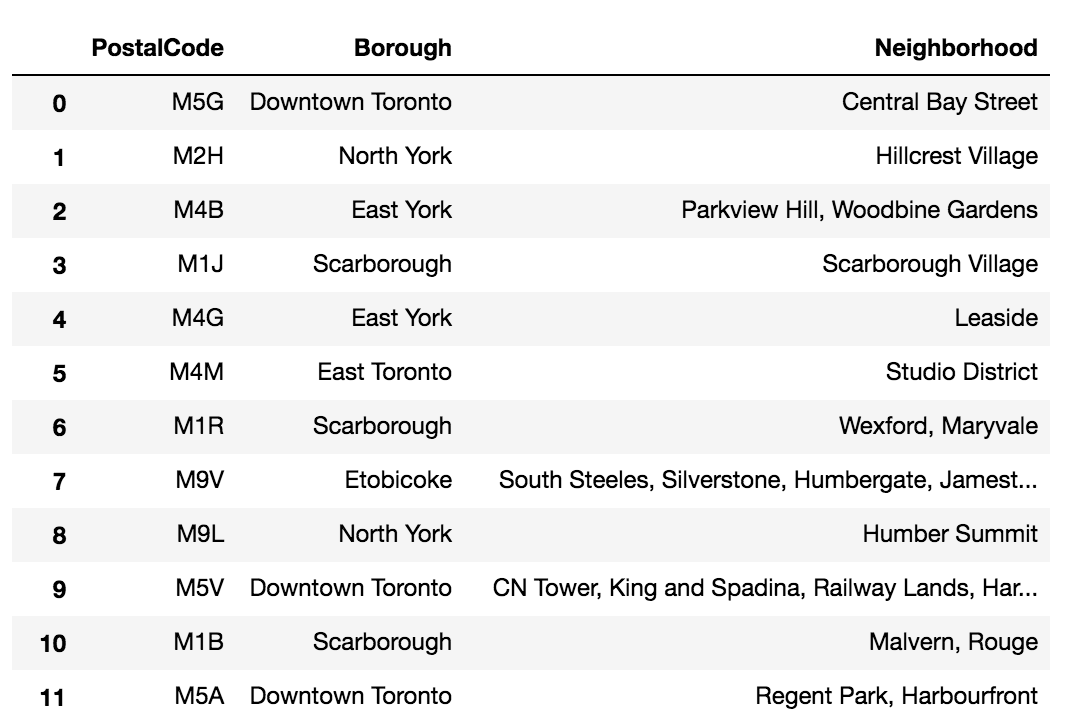
For the Toronto neighborhood data, a Wikipedia page exists that has all the information we need to explore and cluster the neighborhoods in Toronto. You will be required to scrape the Wikipedia page and wrangle the data, clean it, and then read it into a pandas dataframe so that it is in a structured format like the New York dataset.

Once the data is in a structured format, you can replicate the analysis that we did to the New York City dataset to explore and cluster the neighborhoods in the city of Toronto.

Your submission will be a link to your Jupyter Notebook on your Github repository.

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

1. Start by creating a new Notebook for this assignment.
2. Use the Notebook to build the code to scrape the following Wikipedia page, [https://en.wikipedia.org/wiki/List\_of\_postal\_codes\_of\_Canada:\_M,](https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M) in order to obtain the data that is in the table of postal codes and to transform the data into a *pandas*dataframe like the one shown below:



3. To create the above dataframe:

* The dataframe will consist of three columns: PostalCode, Borough, and Neighborhood
* Only process the cells that have an assigned borough. Ignore cells with a borough that is **Not assigned.**
* More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that **M5A** is listed twice and has two neighborhoods: **Harbourfront**and **Regent Park**. These two rows will be combined into one row with the neighborhoods separated with a comma as shown in **row 11**in the above table.
* If a cell has a borough but a **Not assigned**neighborhood, then the neighborhood will be the same as the borough.
* Clean your Notebook and add Markdown cells to explain your work and any assumptions you are making.
* In the last cell of your notebook, use the **.shape** method to print the number of rows of your dataframe.

4. Submit a link to your Notebook on your Github repository. (**10 marks**)

**Note**: There are different website scraping libraries and packages in Python. For scraping the above table, you can simply use *pandas*to read the table into a *pandas*dataframe.

Another way, which would help to learn for more complicated cases of web scraping is using the BeautifulSoup package. Here is the package's main documentation page: <http://beautiful-soup-4.readthedocs.io/en/latest/>

Use *pandas,*or the BeautifulSoup package, or any other way you are comfortable with to transform the data in the table on the Wikipedia page into the above *pandas*dataframe.

​

Now that you have built a dataframe of the postal code of each neighborhood along with the borough name and neighborhood name, in order to utilize the Foursquare location data, we need to get the latitude and the longitude coordinates of each neighborhood.

In an older version of this course, we were leveraging the Google Maps Geocoding API to get the latitude and the longitude coordinates of each neighborhood. However, recently Google started charging for their API: <http://geoawesomeness.com/developers-up-in-arms-over-google-maps-api-insane-price-hike/,> so we will use the Geocoder Python package instead: https://geocoder.readthedocs.io/index.html.

The problem with this Package is you have to be persistent sometimes in order to get the geographical coordinates of a given postal code. So you can make a call to get the latitude and longitude coordinates of a given postal code and the result would be None, and then make the call again and you would get the coordinates. So, in order to make sure that you get the coordinates for all of our neighborhoods, you can run a while loop for each postal code. Taking postal code **M5G**as an example, your code would look something like this:



1

2

3

4

5

6

7

8

9

10

11

12

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))

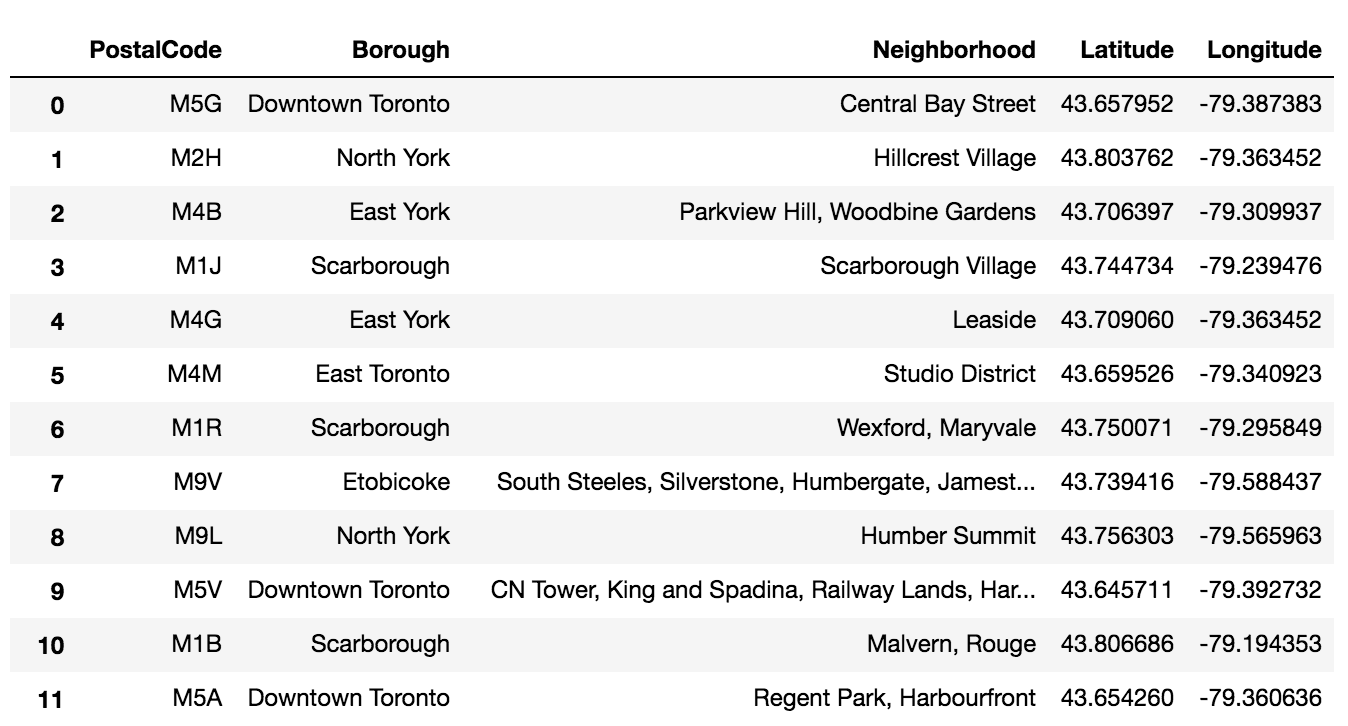
lat\_lng\_coords = g.latlng

latitude = lat\_lng\_coords[0]

longitude = lat\_lng\_coords[1]

Given that this package can be very unreliable, in case you are not able to get the geographical coordinates of the neighborhoods using the Geocoder package, here is a link to a csv file that has the geographical coordinates of each postal code: [http://cocl.us/Geospatial\_data](https://cocl.us/Geospatial_data)

Use the Geocoder package or the csv file to create the following dataframe:



**Important Note:**There is a limit on how many times you can call geocoder.google function. It is 2500 times per day. This should be way more than enough for you to get acquainted with the package and to use it to get the geographical coordinates of the neighborhoods in the Toronto.

Once you are able to create the above dataframe, submit a link to the new Notebook on your Github repository. (**2 marks**)



Explore and cluster the neighborhoods in Toronto. You can decide to work with only boroughs that contain the word Toronto and then replicate the same analysis we did to the New York City data. It is up to you.

Just make sure:

1. to add enough Markdown cells to explain what you decided to do and to report any observations you make.
2. to generate maps to visualize your neighborhoods and how they cluster together.

Once you are happy with your analysis, submit a link to the new Notebook on your Github repository. (**3 marks)**