# Final Capstone Project Luis Mantilla

A full report consisting of all the following components (**15 marks**):

1. Introduction where you discuss the business problem and who would be interested in this project.

As a parent I always wanted to send my kids to study abroad, it is a personal goal, coming from a small country I am not aware of the sheer distances that people faced in big countries, sometimes they were compel to move from one place to another to have a new job, to give you an idea of the size of my country, it is divided politically into seven provinces, in an area called "The great Metropolitan Area", located in the central valley, of roughly the 4% of the country's total area you will see the following:

- Four of our biggest cities are clustered (~four provinces)
- 60% of the population lives there (~3 million people)
- 27 Private and Public Universities and Technological Institutions were located there

You can walk our downtown in less than 30 minutes. That is why is so challenging for us to think in terms of cities bigger than our own country. That is why I would like to make a guide on how I identify not only the universities on a particular country and city, but the nearest and safest neighborhoods for them (my kids) to study abroad.

2. Data where you describe the data that will be used to solve the problem and the source of the data.

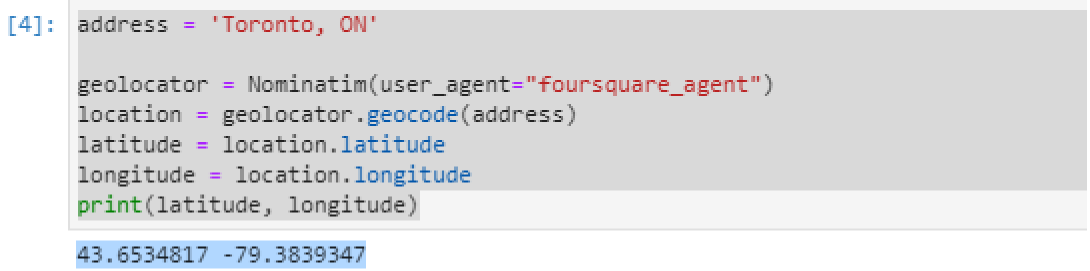
Once the problem was described I would like to explain the data that I am going to use to answer the following question: "Once I selected the Country, City and University, which neighborhoods are the safest and nearest from the selected learning institution?

To answer the above question, I propose the following:

- Get the selected Country and Citi, in this case – Canada and Toronto, to use Foursquare to identify the Best Universities in the area selected.
- Once the Country and City are selected, find a list of Universities (Using Foursquare and the location of the City)
- Get a list of Neighborhoods nearby to the selected institution (Wikipedia Postal Codes of Toronto) and start the analysis

3. Methodology section which represents the main component of the report where you discuss and describe any exploratory data analysis that you did, any inferential statistical testing that you performed, if any, and what machine learnings were used and why.

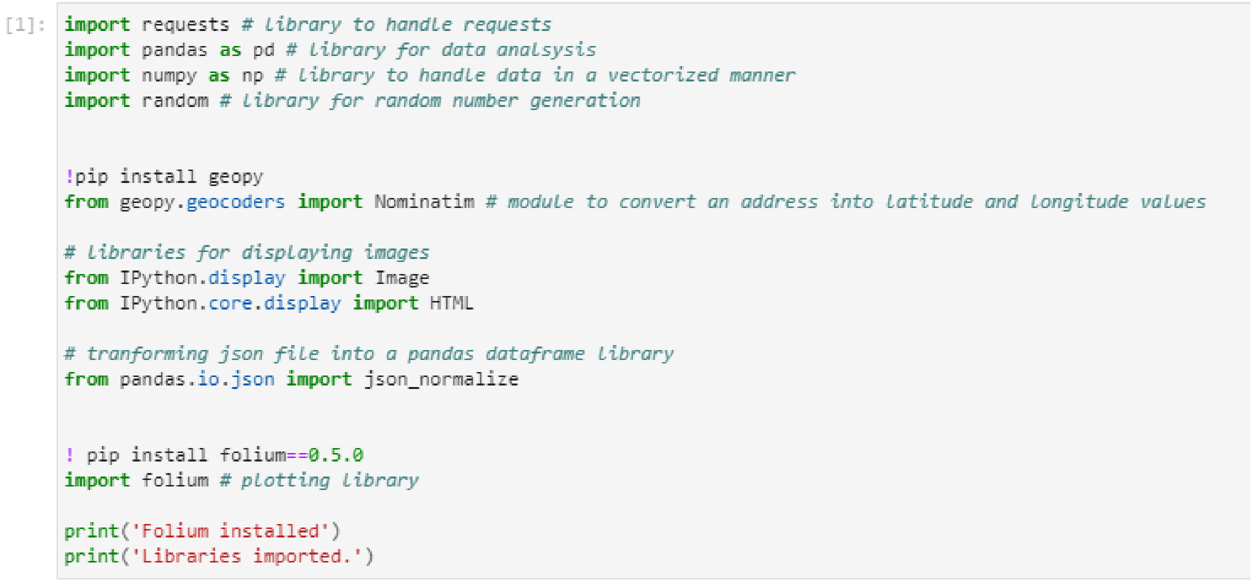
First, we need to know the Latitude and Longitude coordinates, 43.6534817 -79.3839347



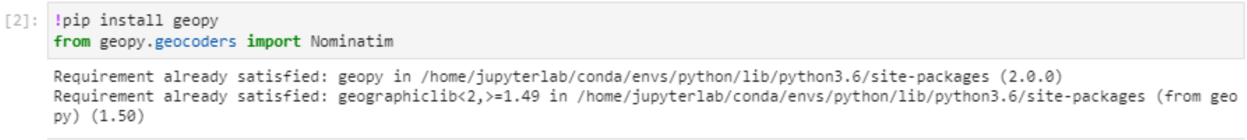
```
[4]: address = 'Toronto, ON'

     geolocator = Nominatim(user_agent="foursquare_agent")
     location = geolocator.geocode(address)
     latitude = location.latitude
     longitude = location.longitude
     print(latitude, longitude)

     43.6534817 -79.3839347
```

.

Once the location is found (See the Notebook), is just a matter to search with the help of Foursquare



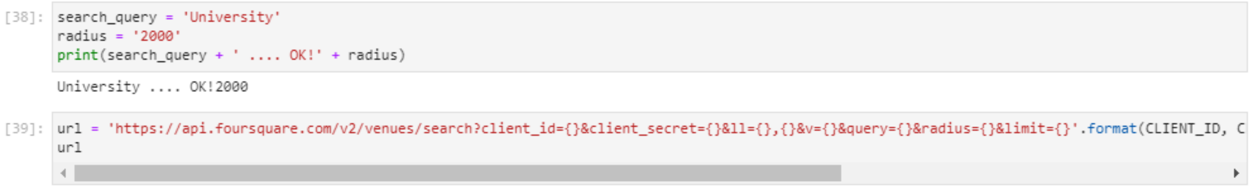
```
[1]: import requests # library to handle requests
     import pandas as pd # library for data analsysis
     import numpy as np # library to handle data in a vectorized manner
     import random # library for random number generation


     !pip install geopy
     from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values

     # libraries for displaying images
     from IPython.display import Image
     from IPython.core.display import HTML

     # tranforming json file into a pandas dataframe library
     from pandas.io.json import json_normalize


     ! pip install folium==0.5.0
     import folium # plotting library

     print('Folium installed')
     print('Libraries imported.')
```



```
[2]: !pip install geopy
     from geopy.geocoders import Nominatim

     Requirement already satisfied: geopy in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (2.0.0)
     Requirement already satisfied: geographiclib<2,>=1.49 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from geo
     py) (1.50)
```

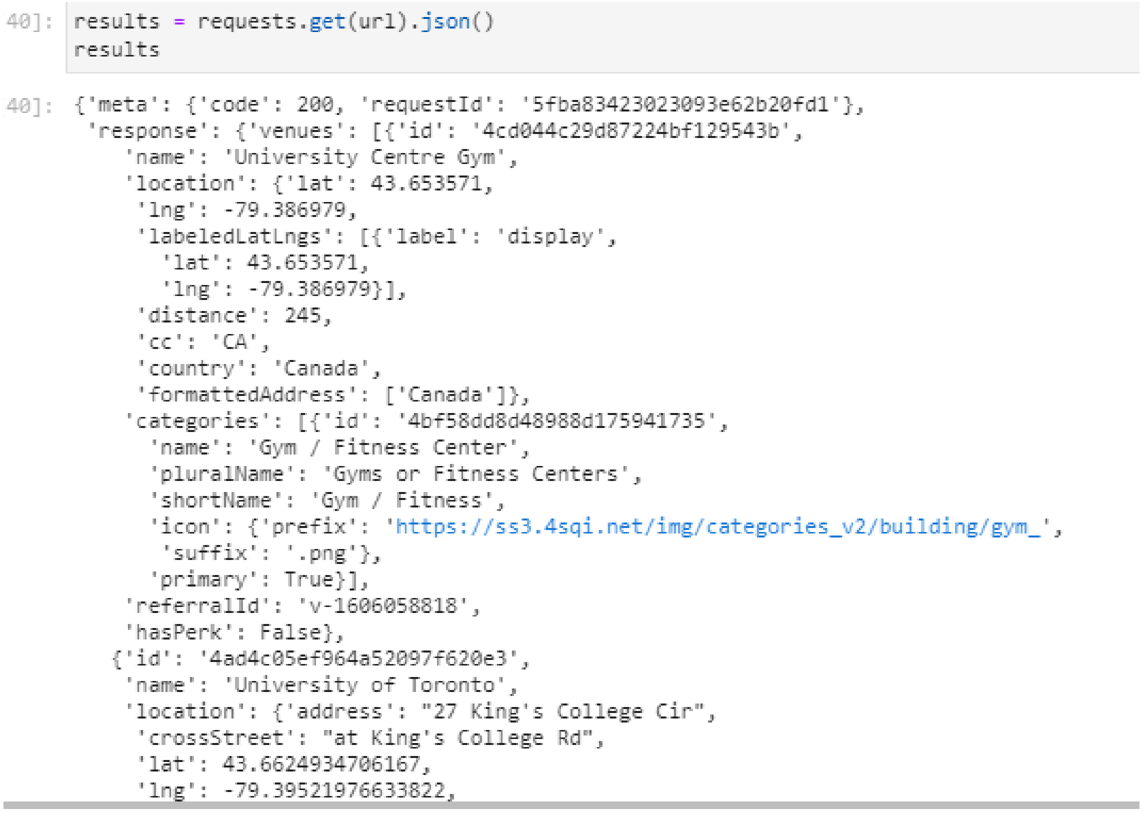
The search query for the Universities is by looking at the venues categories in the following site:

https://developer.foursquare.com/docs/build-with-foursquare/categories/ and investigate in the documentation for the venue code, for College & University. To look for a radius of 2000 meters the venue that I am looking for:



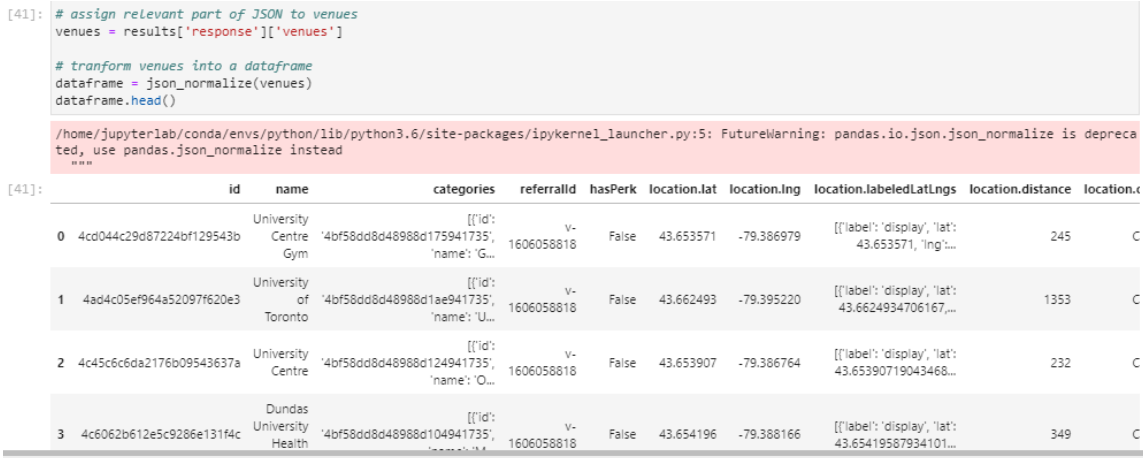
```
[38]: search_query = 'University'
      radius = '2000'
      print(search_query + ' .... OK!' + radius)

      University .... OK!2000

[39]: url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, C
      url
```

To clean the list this is the information contained, filtered by name



```
40]: results = requests.get(url).json()
     results
```

```
40]: {'meta': {'code': 200, 'requestId': '5fba83423023093e62b20fd1'},
      'response': {'venues': [{'id': '4cd044c29d87224bf129543b',
         'name': 'University Centre Gym',
         'location': {'lat': 43.653571,
          'lng': -79.386979,
          'labeledLatLngs': [{'label': 'display',
            'lat': 43.653571,
            'lng': -79.386979}],
          'distance': 245,
          'cc': 'CA',
          'country': 'Canada',
          'formattedAddress': ['Canada']},
         'categories': [{'id': '4bf58dd8d48988d175941735',
           'name': 'Gym / Fitness Center',
           'pluralName': 'Gyms or Fitness Centers',
           'shortName': 'Gym / Fitness',
           'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/building/gym_',
            'suffix': '.png'},
           'primary': True}],
         'referralId': 'v-1606058818',
         'hasPerk': False},
        {'id': '4ad4c05ef964a52097f620e3',
         'name': 'University of Toronto',
         'location': {'address': "27 King's College Cir",
          'crossStreet': "at King's College Rd",
          'lat': 43.6624934706167,
          'lng': -79.39521976633822,
```

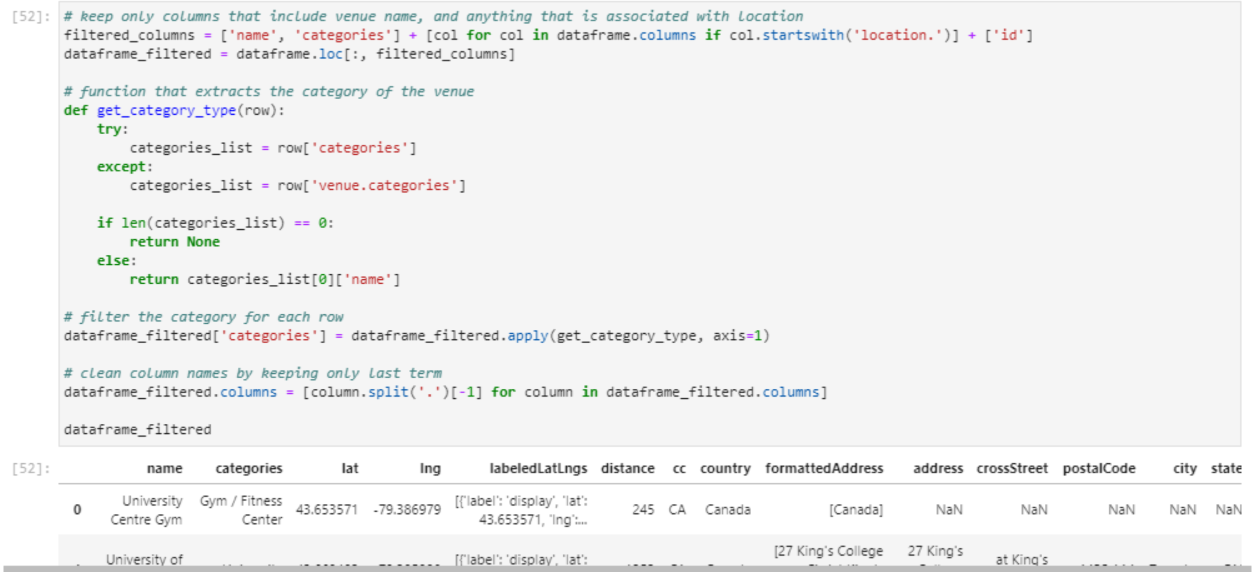


```
[41]: # assign relevant part of JSON to venues
      venues = results['response']['venues']

      # tranform venues into a dataframe
      dataframe = json_normalize(venues)
      dataframe.head()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages/ipykernel_launcher.py:5: FutureWarning: pandas.io.json.json_normalize is depreca
ted, use pandas.json_normalize instead
  """
```

[41]:

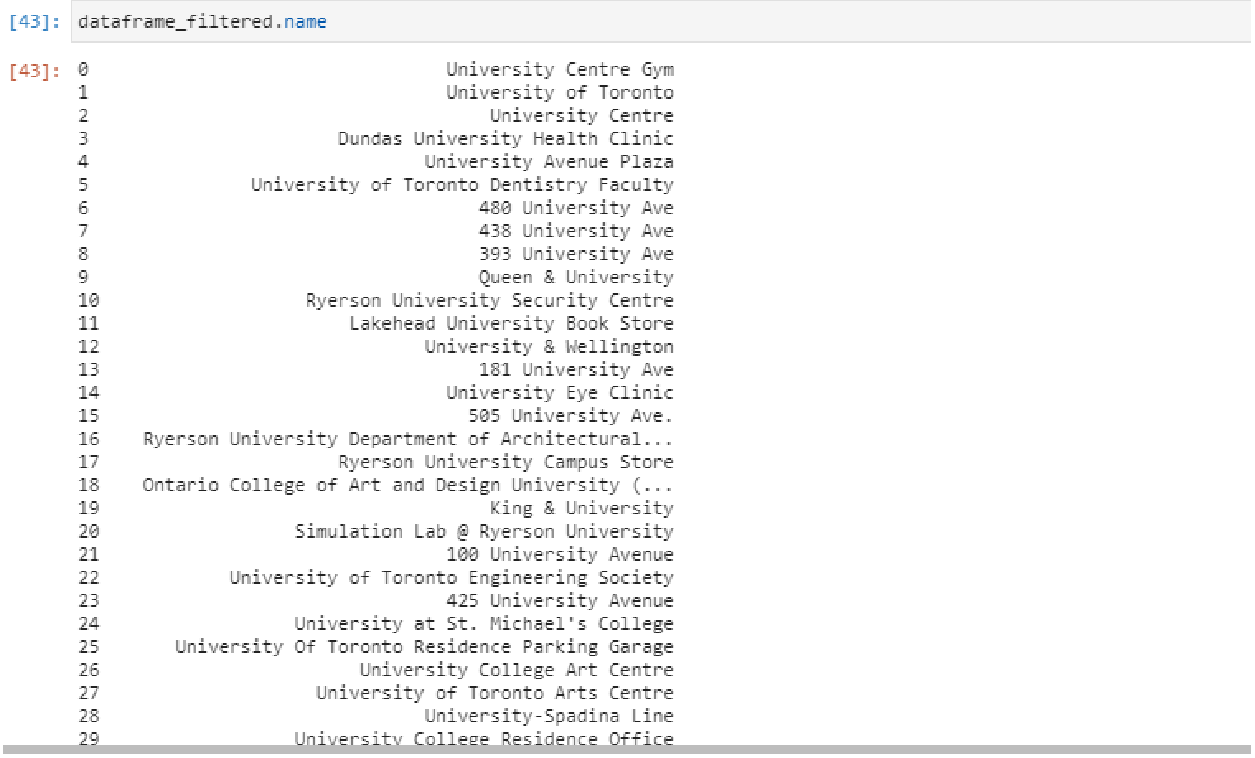
| | id | name | categories | referralId | hasPerk | location.lat | location.lng | location.labeledLatLngs | location.distance | location.c |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4cd044c29d87224bf129543b | University Centre Gym | [{'id': '4bf58dd8d48988d175941735', 'name': 'G... | v-1606058818 | False | 43.653571 | -79.386979 | [{'label': 'display', 'lat': 43.653571, 'lng':... | 245 | C |
| 1 | 4ad4c05ef964a52097f620e3 | University of Toronto | [{'id': '4bf58dd8d48988d1ae941735', 'name': 'U... | v-1606058818 | False | 43.662493 | -79.395220 | [{'label': 'display', 'lat': 43.6624934706167,... | 1353 | C |
| 2 | 4c45c6c6da2176b09543637a | University Centre | [{'id': '4bf58dd8d48988d124941735', 'name': 'O... | v-1606058818 | False | 43.653907 | -79.386764 | [{'label': 'display', 'lat': 43.65390719043468... | 232 | C |
| 3 | 4c6062b612e5c9286e131f4c | Dundas University Health | [{'id': '4bf58dd8d48988d104941735', | v-1606058818 | False | 43.654196 | -79.388166 | [{'label': 'display', 'lat': 43.65419587934101... | 349 | C |



```
[52]: # keep only columns that include venue name, and anything that is associated with location
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']

# filter the category for each row
dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

# clean column names by keeping only last term
dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.columns]

dataframe_filtered
```

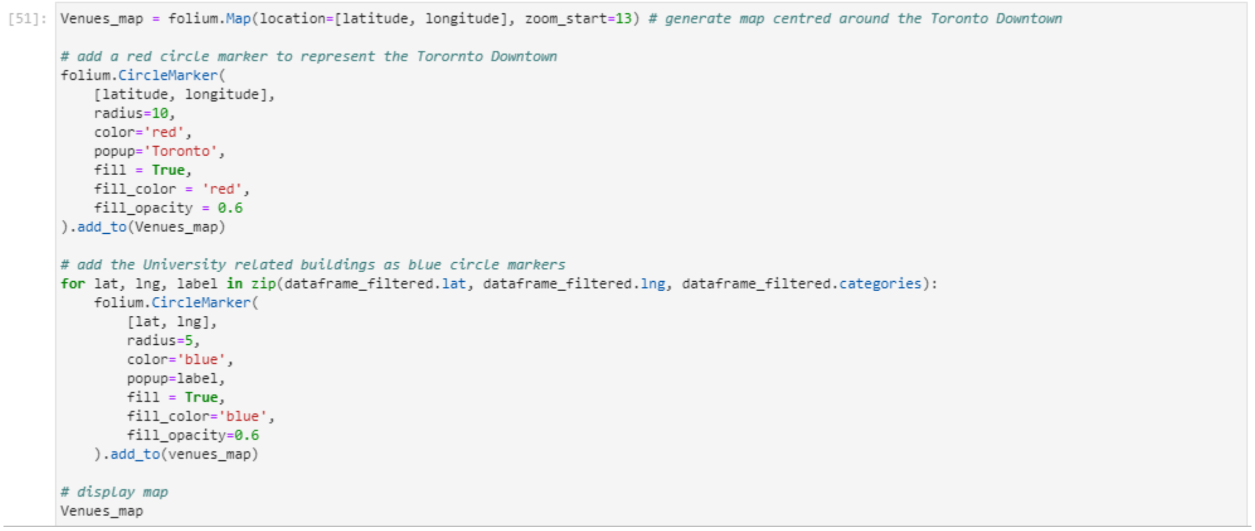
[52]:

| | name | categories | lat | lng | labeledLatLngs | distance | cc | country | formattedAddress | address | crossStreet | postalCode | city | state |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | University Centre Gym | Gym / Fitness Center | 43.653571 | -79.386979 | [{'label': 'display', 'lat': 43.653571, 'lng':... | 245 | CA | Canada | [Canada] | NaN | NaN | NaN | NaN | NaN |
| | University of | | | | [{'label': 'display', 'lat': | | | | [27 King's College | 27 King's | at King's | | | |



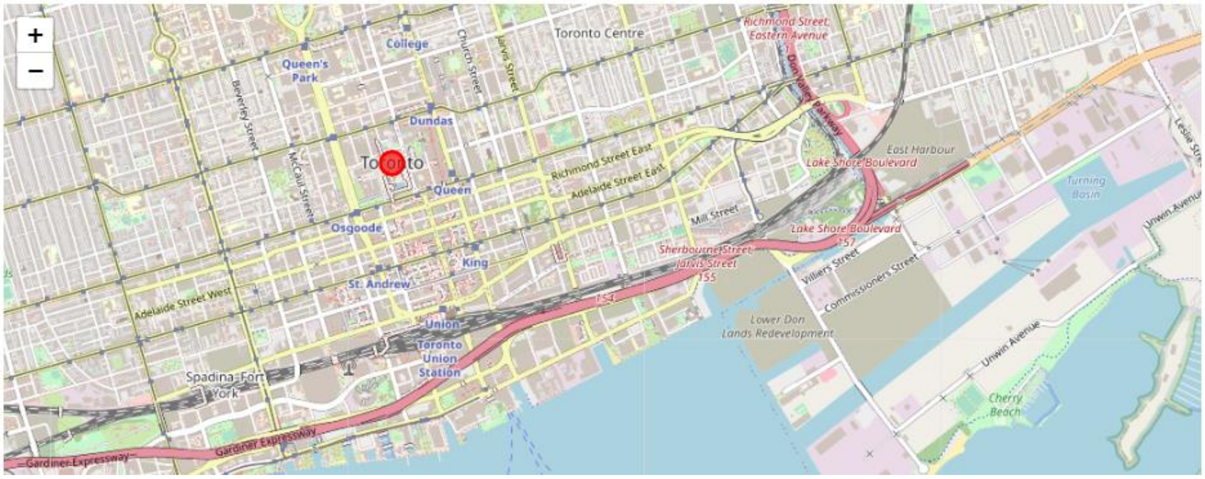
```
[43]: dataframe_filtered.name

[43]: 0                                  University Centre Gym
      1                                  University of Toronto
      2                                      University Centre
      3                         Dundas University Health Clinic
      4                                University Avenue Plaza
      5                 University of Toronto Dentistry Faculty
      6                                    480 University Ave
      7                                    438 University Ave
      8                                    393 University Ave
      9                                    Queen & University
      10                     Ryerson University Security Centre
      11                          Lakehead University Book Store
      12                               University & Wellington
      13                                   181 University Ave
      14                                 University Eye Clinic
      15                                  505 University Ave.
      16     Ryerson University Department of Architectural...
      17                        Ryerson University Campus Store
      18     Ontario College of Art and Design University (...
      19                                     King & University
      20                       Simulation Lab @ Ryerson University
      21                                 100 University Avenue
      22                University of Toronto Engineering Society
      23                                 425 University Avenue
      24                       University at St. Michael's College
      25          University Of Toronto Residence Parking Garage
      26                            University College Art Centre
      27                         University of Toronto Arts Centre
      28                                  University-Spadina Line
      29                       University College Residence Office
```

To locate the venues in the map

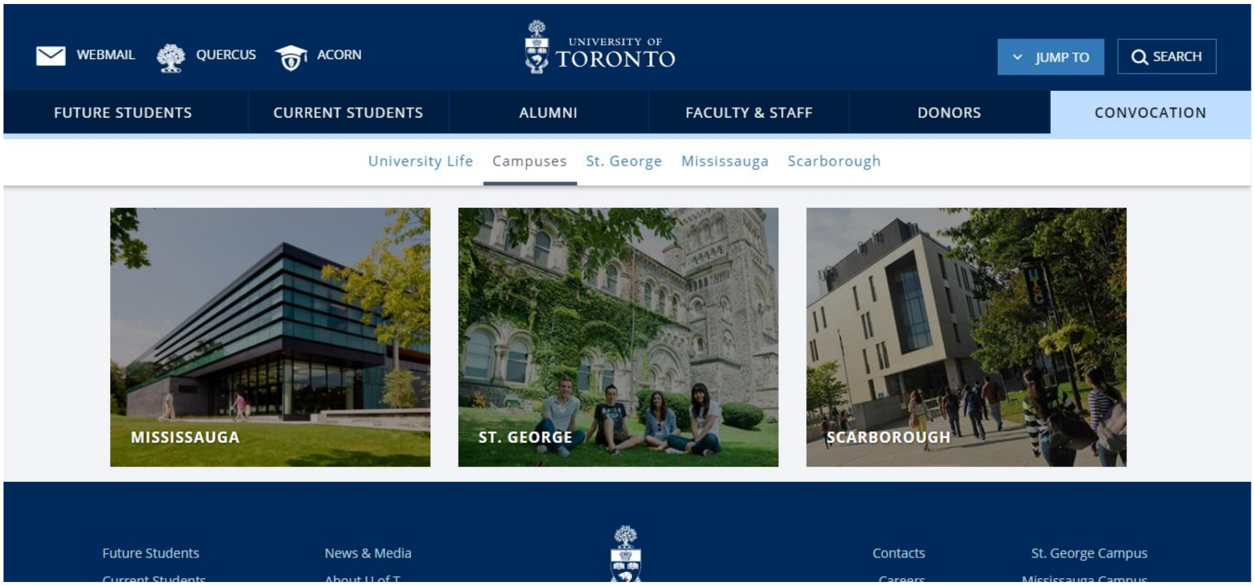


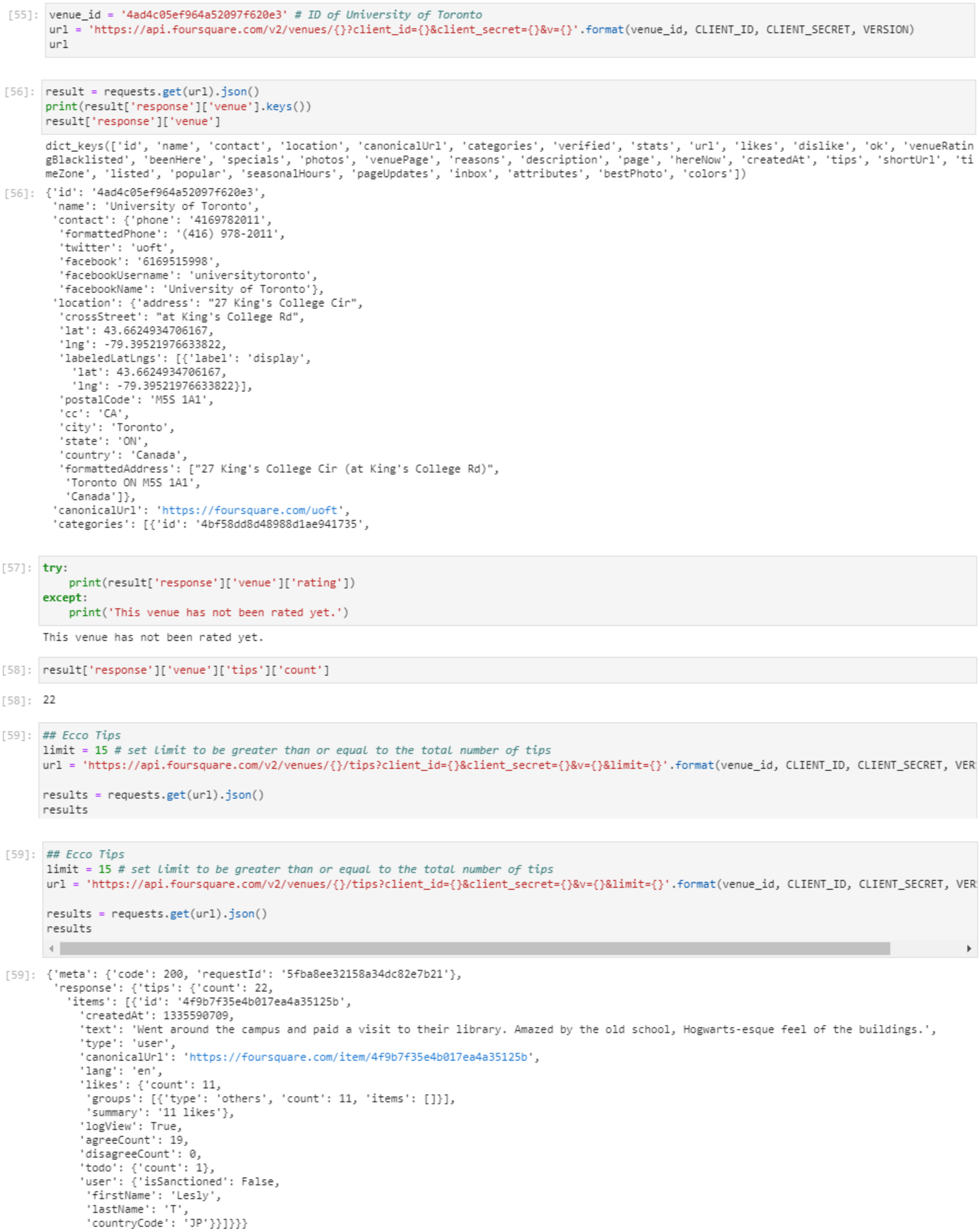
```
[51]: Venues_map = folium.Map(location=[latitude, longitude], zoom_start=13) # generate map centred around the Toronto Downtown

      # add a red circle marker to represent the Torornto Downtown
      folium.CircleMarker(
          [latitude, longitude],
          radius=10,
          color='red',
          popup='Toronto',
          fill = True,
          fill_color = 'red',
          fill_opacity = 0.6
      ).add_to(Venues_map)

      # add the University related buildings as blue circle markers
      for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe_filtered.categories):
          folium.CircleMarker(
              [lat, lng],
              radius=5,
              color='blue',
              popup=label,
              fill = True,
              fill_color='blue',
              fill_opacity=0.6
          ).add_to(venues_map)

      # display map
      Venues_map
```

[51]:



Of interest is the line 1 of the dataframe: https://www.utoronto.ca/university-life/campuses





```
[55]: venue_id = '4ad4c05ef964a52097f620e3' # ID of University of Toronto
      url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VERSION)
      url
```

```
[56]: result = requests.get(url).json()
      print(result['response']['venue'].keys())
      result['response']['venue']
```

```
      dict_keys(['id', 'name', 'contact', 'location', 'canonicalUrl', 'categories', 'verified', 'stats', 'url', 'likes', 'dislike', 'ok', 'venueRatin
      gBlacklisted', 'beenHere', 'specials', 'photos', 'venuePage', 'reasons', 'description', 'page', 'hereNow', 'createdAt', 'tips', 'shortUrl', 'ti
      meZone', 'listed', 'popular', 'seasonalHours', 'pageUpdates', 'inbox', 'attributes', 'bestPhoto', 'colors'])
[56]: {'id': '4ad4c05ef964a52097f620e3',
       'name': 'University of Toronto',
       'contact': {'phone': '4169782011',
        'formattedPhone': '(416) 978-2011',
        'twitter': 'uoft',
        'facebook': '6169515998',
        'facebookUsername': 'universitytoronto',
        'facebookName': 'University of Toronto'},
       'location': {'address': "27 King's College Cir",
        'crossStreet': "at King's College Rd",
        'lat': 43.6624934706167,
        'lng': -79.39521976633822,
        'labeledLatLngs': [{'label': 'display',
          'lat': 43.6624934706167,
          'lng': -79.39521976633822}],
        'postalCode': 'M5S 1A1',
        'cc': 'CA',
        'city': 'Toronto',
        'state': 'ON',
        'country': 'Canada',
        'formattedAddress': ["27 King's College Cir (at King's College Rd)",
         'Toronto ON M5S 1A1',
         'Canada']},
       'canonicalUrl': 'https://foursquare.com/uoft',
       'categories': [{'id': '4bf58dd8d48988d1ae941735',
```

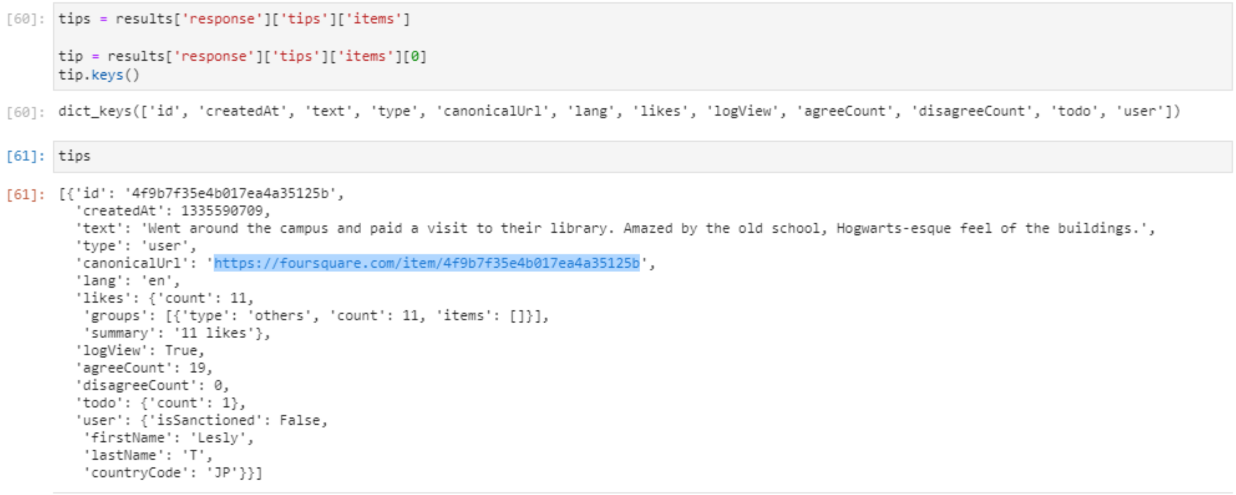
```
[57]: try:
          print(result['response']['venue']['rating'])
      except:
          print('This venue has not been rated yet.')

      This venue has not been rated yet.
```

```
[58]: result['response']['venue']['tips']['count']

[58]: 22
```

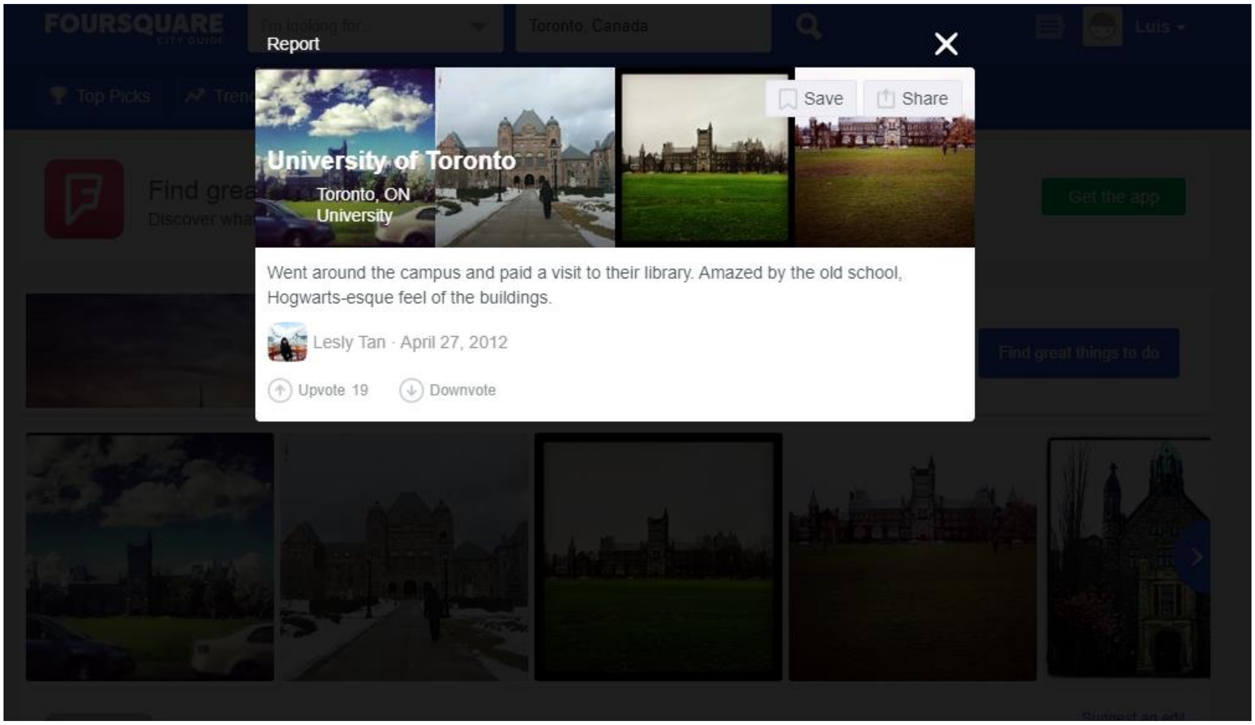
```
[59]: ## Ecco Tips
      limit = 15 # set limit to be greater than or equal to the total number of tips
      url = 'https://api.foursquare.com/v2/venues/{}/tips?client_id={}&client_secret={}&v={}&limit={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VER

      results = requests.get(url).json()
      results
```

```
[59]: ## Ecco Tips
      limit = 15 # set limit to be greater than or equal to the total number of tips
      url = 'https://api.foursquare.com/v2/venues/{}/tips?client_id={}&client_secret={}&v={}&limit={}'.format(venue_id, CLIENT_ID, CLIENT_SECRET, VER

      results = requests.get(url).json()
      results
```

```
[59]: {'meta': {'code': 200, 'requestId': '5fba8ee32158a34dc82e7b21'},
       'response': {'tips': {'count': 22,
         'items': [{'id': '4f9b7f35e4b017ea4a35125b',
           'createdAt': 1335590709,
           'text': 'Went around the campus and paid a visit to their library. Amazed by the old school, Hogwarts-esque feel of the buildings.',
           'type': 'user',
           'canonicalUrl': 'https://foursquare.com/item/4f9b7f35e4b017ea4a35125b',
           'lang': 'en',
           'likes': {'count': 11,
            'groups': [{'type': 'others', 'count': 11, 'items': []}],
            'summary': '11 likes'},
           'logView': True,
           'agreeCount': 19,
           'disagreeCount': 0,
           'todo': {'count': 1},
           'user': {'isSanctioned': False,
            'firstName': 'Lesly',
            'lastName': 'T',
            'countryCode': 'JP'}}]}}}
```



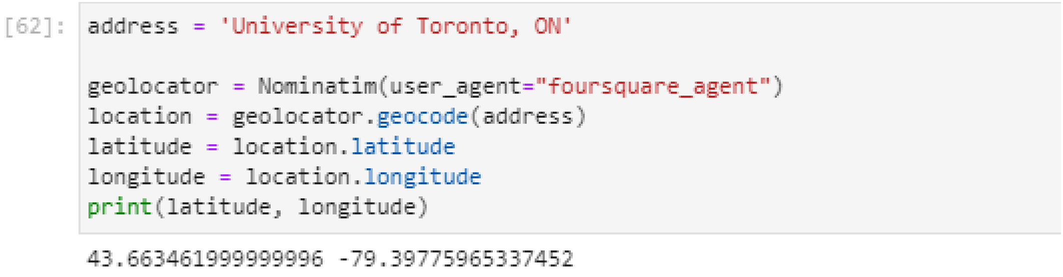
```
[60]: tips = results['response']['tips']['items']

      tip = results['response']['tips']['items'][0]
      tip.keys()

[60]: dict_keys(['id', 'createdAt', 'text', 'type', 'canonicalUrl', 'lang', 'likes', 'logView', 'agreeCount', 'disagreeCount', 'todo', 'user'])

[61]: tips

[61]: [{'id': '4f9b7f35e4b017ea4a35125b',
        'createdAt': 1335590709,
        'text': 'Went around the campus and paid a visit to their library. Amazed by the old school, Hogwarts-esque feel of the buildings.',
        'type': 'user',
        'canonicalUrl': 'https://foursquare.com/item/4f9b7f35e4b017ea4a35125b',
        'lang': 'en',
        'likes': {'count': 11,
         'groups': [{'type': 'others', 'count': 11, 'items': []}],
         'summary': '11 likes'},
        'logView': True,
        'agreeCount': 19,
        'disagreeCount': 0,
        'todo': {'count': 1},
        'user': {'isSanctioned': False,
         'firstName': 'Lesly',
         'lastName': 'T',
         'countryCode': 'JP'}}]
```



Went around the campus and paid a visit to their library. Amazed by the old school, Hogwarts-esque feel of the buildings.

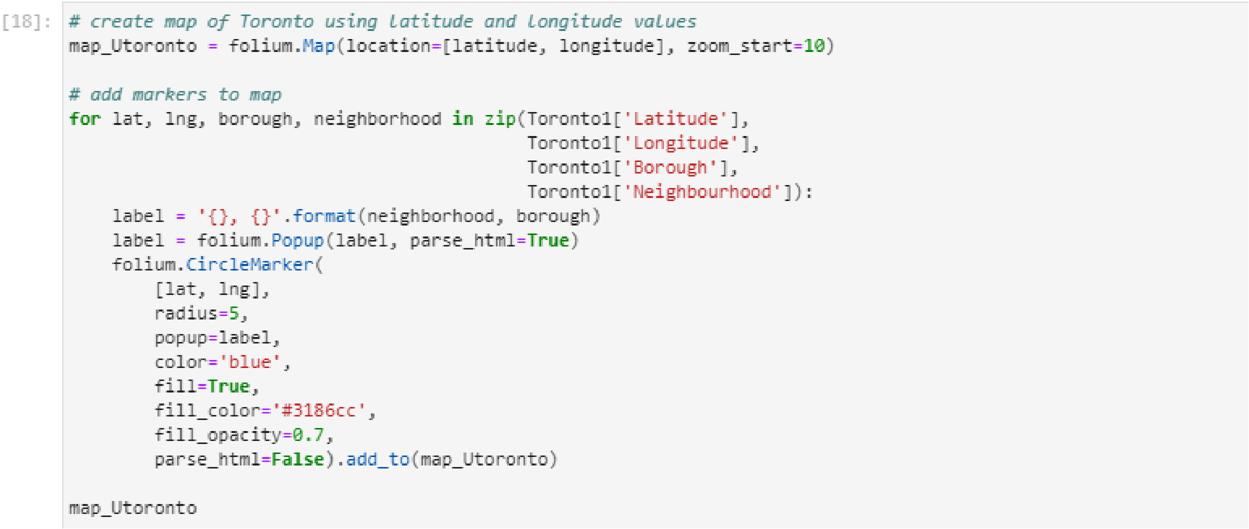
Lesly Tan · April 27, 2012

The University Selected for the Analysis is University of Toronto



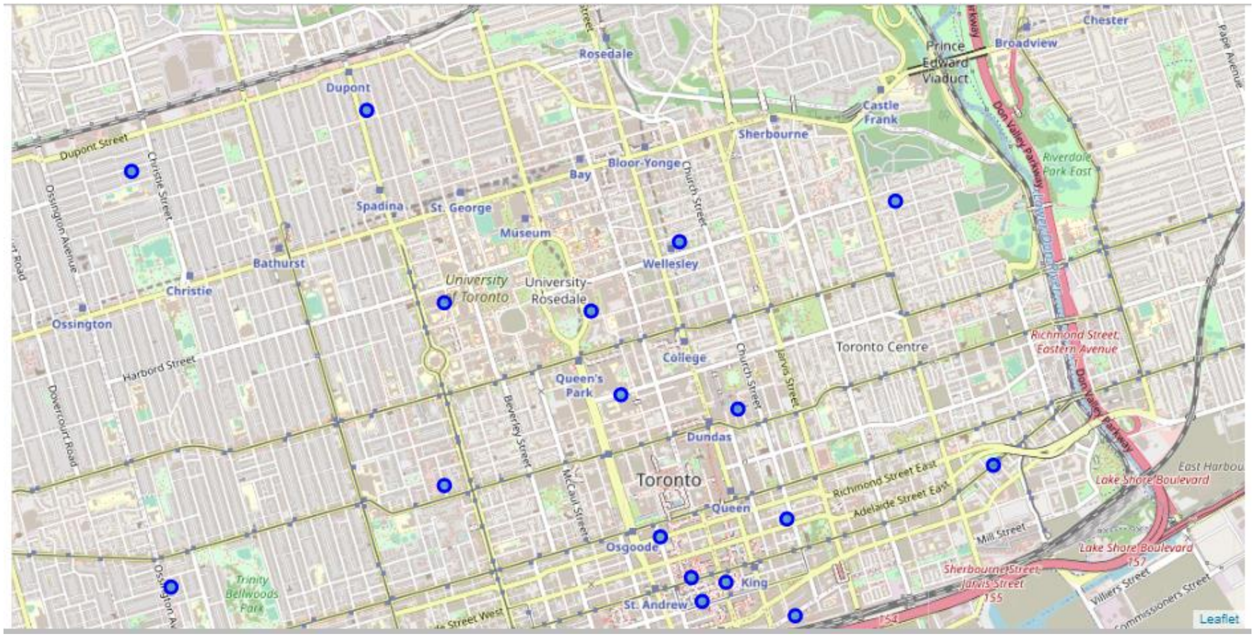
```
[62]: address = 'University of Toronto, ON'

      geolocator = Nominatim(user_agent="foursquare_agent")
      location = geolocator.geocode(address)
      latitude = location.latitude
      longitude = location.longitude
      print(latitude, longitude)

      43.663461999999996 -79.39775965337452
```

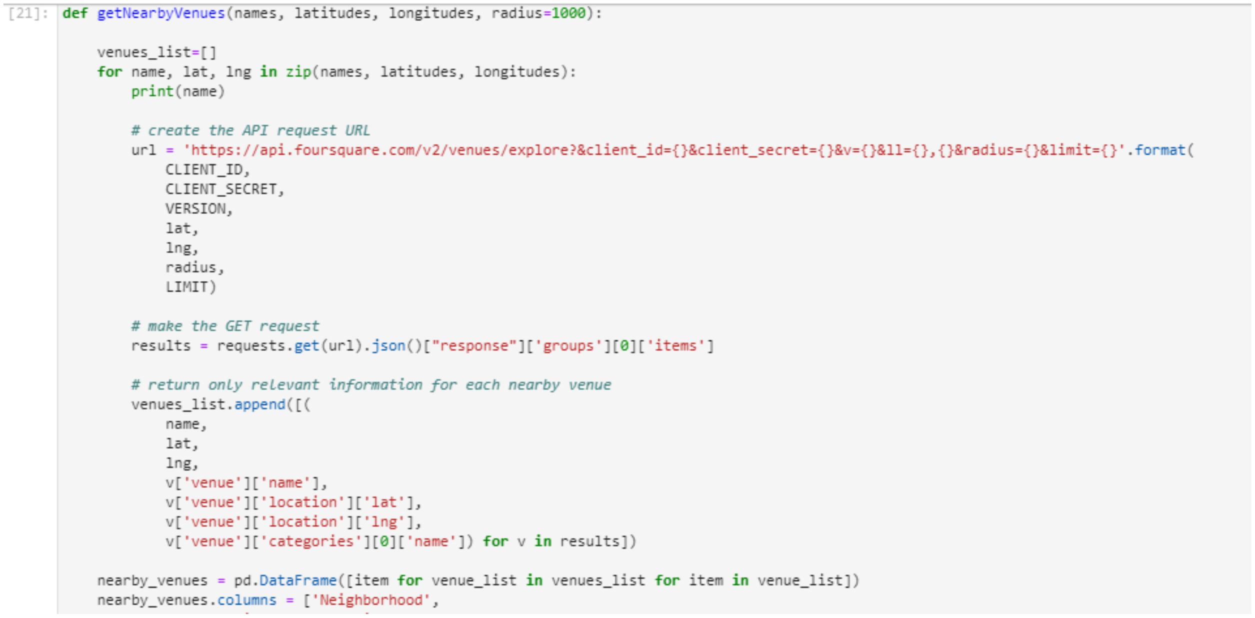
Now follows the identification of the Neighborhoods near the University of Toronto, Postal Code M5S



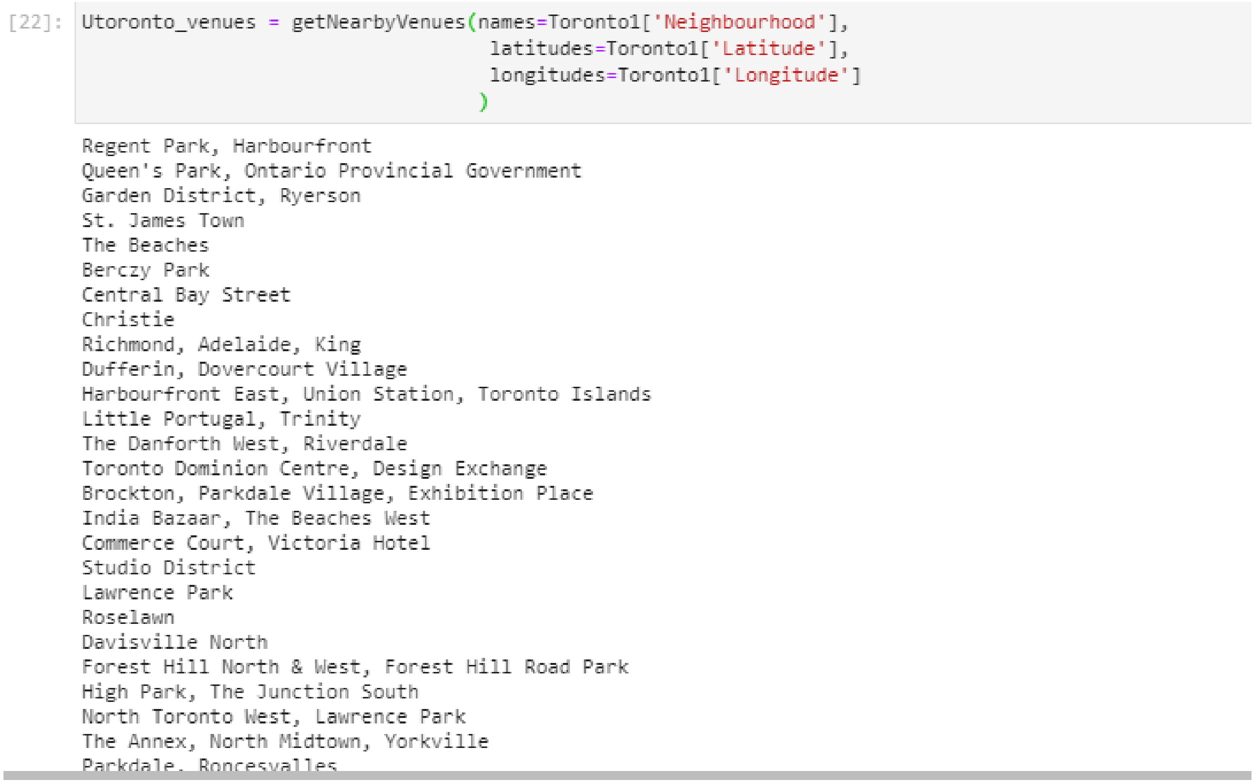
[18]:

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

# add markers to map
for lat, lng, borough, neighborhood in zip(Toronto1['Latitude'],
                                           Toronto1['Longitude'],
                                           Toronto1['Borough'],
                                           Toronto1['Neighbourhood']):
    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_Utoronto)

map_Utoronto
```



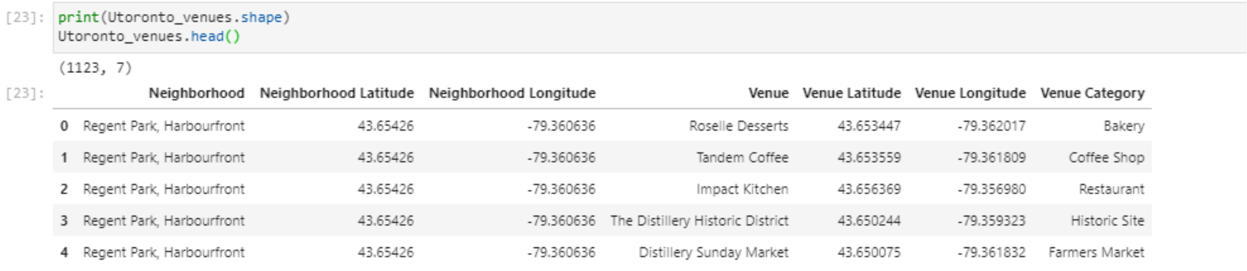


```
[21]: def getNearbyVenues(names, latitudes, longitudes, radius=1000):

          venues_list=[]
          for name, lat, lng in zip(names, latitudes, longitudes):
              print(name)

              # create the API request URL
              url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
                  CLIENT_ID,
                  CLIENT_SECRET,
                  VERSION,
                  lat,
                  lng,
                  radius,
                  LIMIT)

              # make the GET request
              results = requests.get(url).json()["response"]['groups'][0]['items']

              # return only relevant information for each nearby venue
              venues_list.append([(
                  name,
                  lat,
                  lng,
                  v['venue']['name'],
                  v['venue']['location']['lat'],
                  v['venue']['location']['lng'],
                  v['venue']['categories'][0]['name']) for v in results])

          nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
          nearby_venues.columns = ['Neighborhood',
```



```
[22]: Utoronto_venues = getNearbyVenues(names=Toronto1['Neighbourhood'],
                                         latitudes=Toronto1['Latitude'],
                                         longitudes=Toronto1['Longitude']
                                        )
```

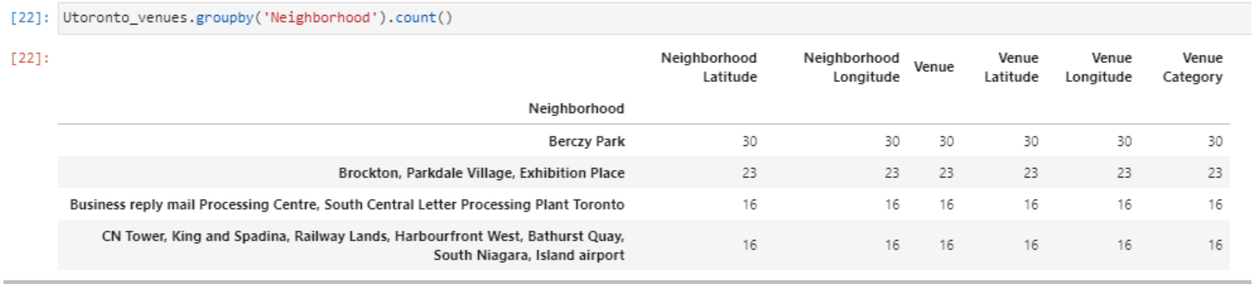
```
Regent Park, Harbourfront
Queen's Park, Ontario Provincial Government
Garden District, Ryerson
St. James Town
The Beaches
Berczy Park
Central Bay Street
Christie
Richmond, Adelaide, King
Dufferin, Dovercourt Village
Harbourfront East, Union Station, Toronto Islands
Little Portugal, Trinity
The Danforth West, Riverdale
Toronto Dominion Centre, Design Exchange
Brockton, Parkdale Village, Exhibition Place
India Bazaar, The Beaches West
Commerce Court, Victoria Hotel
Studio District
Lawrence Park
Roselawn
Davisville North
Forest Hill North & West, Forest Hill Road Park
High Park, The Junction South
North Toronto West, Lawrence Park
The Annex, North Midtown, Yorkville
Parkdale, Roncesvalles
```



```
[23]: print(Utoronto_venues.shape)
      Utoronto_venues.head()

      (1123, 7)
```

[23]:

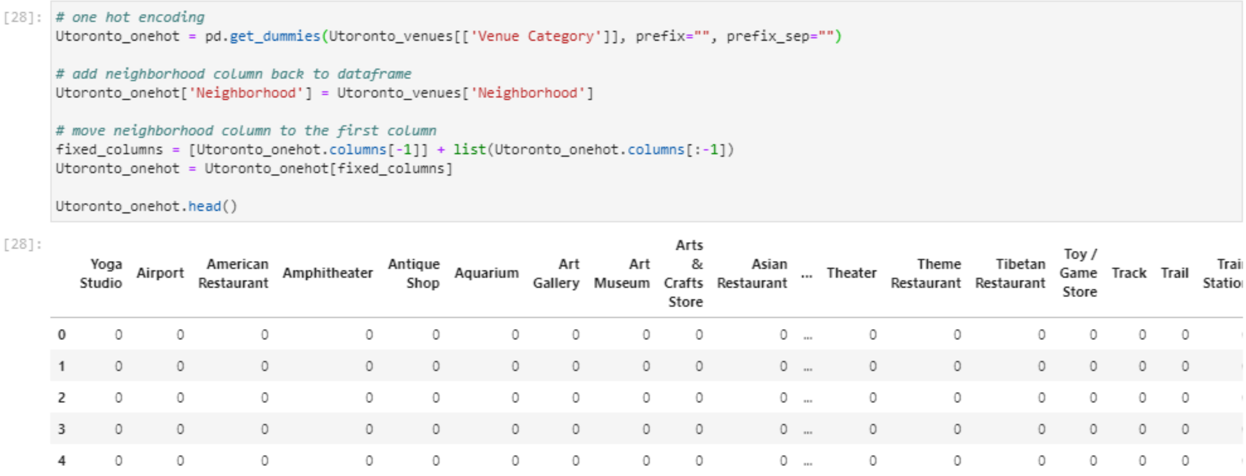
| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Regent Park, Harbourfront | 43.65426 | -79.360636 | Roselle Desserts | 43.653447 | -79.362017 | Bakery |
| 1 | Regent Park, Harbourfront | 43.65426 | -79.360636 | Tandem Coffee | 43.653559 | -79.361809 | Coffee Shop |
| 2 | Regent Park, Harbourfront | 43.65426 | -79.360636 | Impact Kitchen | 43.656369 | -79.356980 | Restaurant |
| 3 | Regent Park, Harbourfront | 43.65426 | -79.360636 | The Distillery Historic District | 43.650244 | -79.359323 | Historic Site |
| 4 | Regent Park, Harbourfront | 43.65426 | -79.360636 | Distillery Sunday Market | 43.650075 | -79.361832 | Farmers Market |



```
[22]: Utoronto_venues.groupby('Neighborhood').count()
```

[22]:

| | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|
| Neighborhood | | | | | | |
| Berczy Park | 30 | 30 | 30 | 30 | 30 | 30 |
| Brockton, Parkdale Village, Exhibition Place | 23 | 23 | 23 | 23 | 23 | 23 |
| Business reply mail Processing Centre, South Central Letter Processing Plant Toronto | 16 | 16 | 16 | 16 | 16 | 16 |
| CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport | 16 | 16 | 16 | 16 | 16 | 16 |

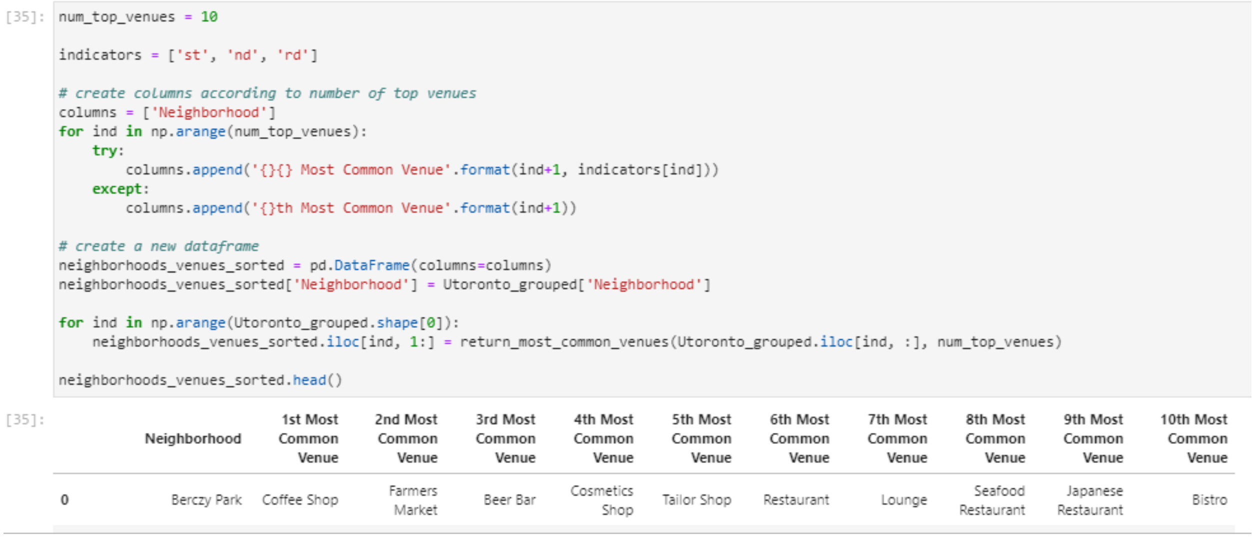


```
[28]: # one hot encoding
      Utoronto_onehot = pd.get_dummies(Utoronto_venues[['Venue Category']], prefix="", prefix_sep="")

      # add neighborhood column back to dataframe
      Utoronto_onehot['Neighborhood'] = Utoronto_venues['Neighborhood']

      # move neighborhood column to the first column
      fixed_columns = [Utoronto_onehot.columns[-1]] + list(Utoronto_onehot.columns[:-1])
      Utoronto_onehot = Utoronto_onehot[fixed_columns]

      Utoronto_onehot.head()
```

[28]:

| | Yoga Studio | Airport | American Restaurant | Amphitheater | Antique Shop | Aquarium | Art Gallery | Art Museum | Arts & Crafts Store | Asian Restaurant | ... | Theater | Theme Restaurant | Tibetan Restaurant | Toy / Game Store | Track | Trail | Trai<br>Statio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | |

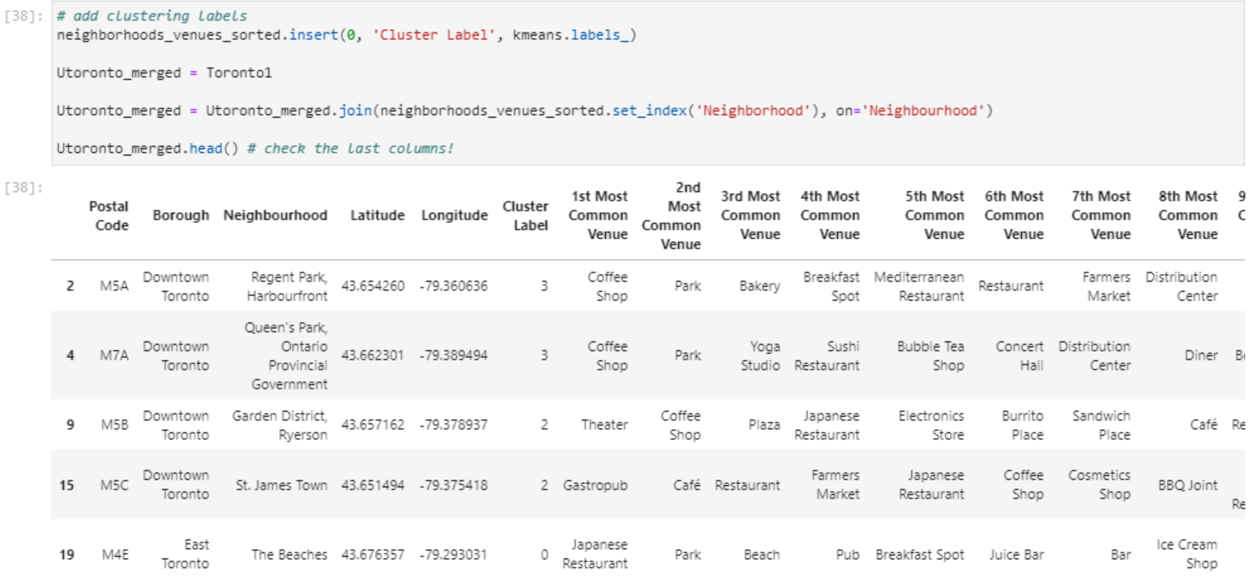
The list of Neighborhoods comes as follows (Criteria is the number of venues):



```
[35]: num_top_venues = 10

      indicators = ['st', 'nd', 'rd']

      # create columns according to number of top venues
      columns = ['Neighborhood']
      for ind in np.arange(num_top_venues):
          try:
              columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
          except:
              columns.append('{}th Most Common Venue'.format(ind+1))

      # create a new dataframe
      neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
      neighborhoods_venues_sorted['Neighborhood'] = Utoronto_grouped['Neighborhood']

      for ind in np.arange(Utoronto_grouped.shape[0]):
          neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(Utoronto_grouped.iloc[ind, :], num_top_venues)

      neighborhoods_venues_sorted.head()
```

[35]:

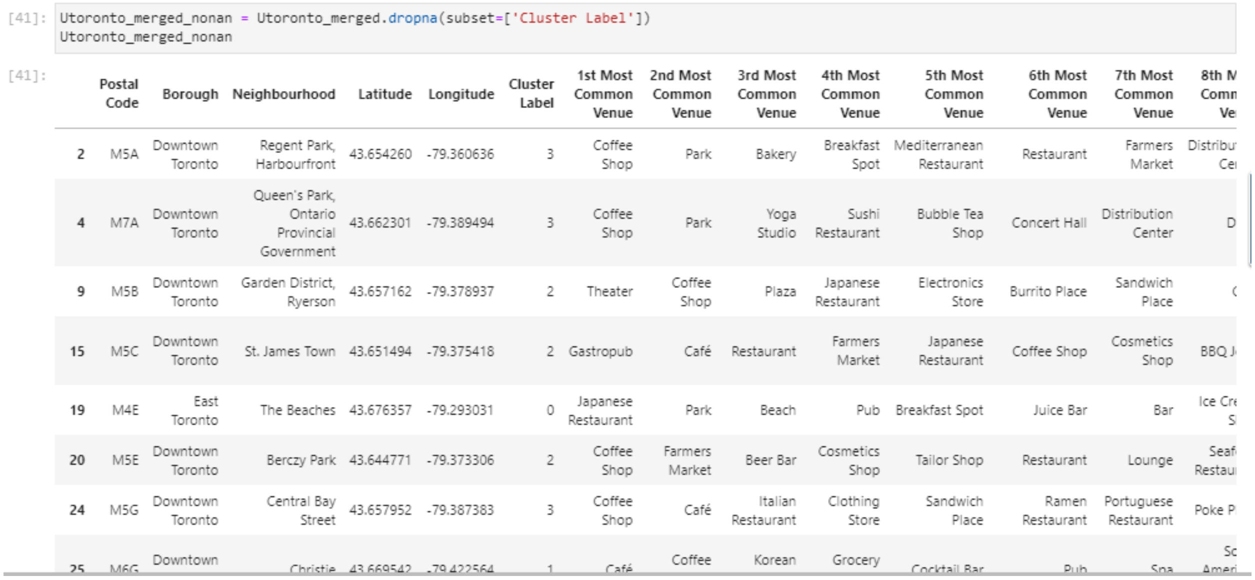
| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9th Most Common Venue | 10th Most Common Venue |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Berczy Park | Coffee Shop | Farmers Market | Beer Bar | Cosmetics Shop | Tailor Shop | Restaurant | Lounge | Seafood Restaurant | Japanese Restaurant | Bistro |



```
[38]: # add clustering labels
      neighborhoods_venues_sorted.insert(0, 'Cluster Label', kmeans.labels_)

      Utoronto_merged = Toronto1

      Utoronto_merged = Utoronto_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighbourhood')

      Utoronto_merged.head() # check the last columns!
```

[38]:

| | Postal Code | Borough | Neighbourhood | Latitude | Longitude | Cluster Label | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th Most Common Venue | 9<br>C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | M5A | Downtown Toronto | Regent Park, Harbourfront | 43.654260 | -79.360636 | 3 | Coffee Shop | Park | Bakery | Breakfast Spot | Mediterranean Restaurant | Restaurant | Farmers Market | Distribution Center | |
| 4 | M7A | Downtown Toronto | Queen's Park, Ontario Provincial Government | 43.662301 | -79.389494 | 3 | Coffee Shop | Park | Yoga Studio | Sushi Restaurant | Bubble Tea Shop | Concert Hall | Distribution Center | Diner | B |
| 9 | M5B | Downtown Toronto | Garden District, Ryerson | 43.657162 | -79.378937 | 2 | Theater | Coffee Shop | Plaza | Japanese Restaurant | Electronics Store | Burrito Place | Sandwich Place | Café | Re |
| 15 | M5C | Downtown Toronto | St. James Town | 43.651494 | -79.375418 | 2 | Gastropub | Café | Restaurant | Farmers Market | Japanese Restaurant | Coffee Shop | Cosmetics Shop | BBQ Joint | Re |
| 19 | M4E | East Toronto | The Beaches | 43.676357 | -79.293031 | 0 | Japanese Restaurant | Park | Beach | Pub | Breakfast Spot | Juice Bar | Bar | Ice Cream Shop | |



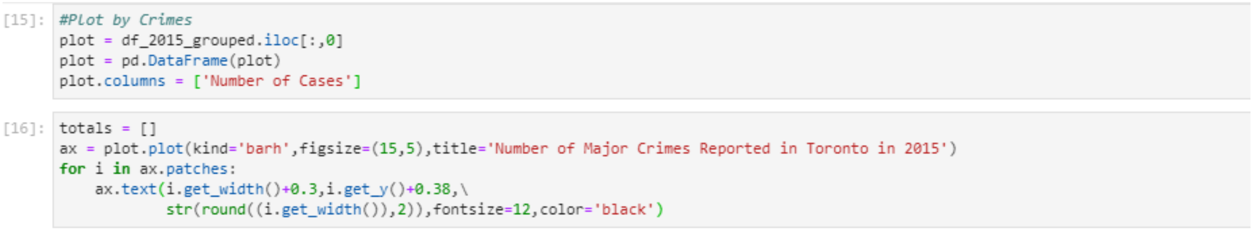
```
[41]: Utoronto_merged_nonan = Utoronto_merged.dropna(subset=['Cluster Label'])
      Utoronto_merged_nonan
```

[41]:

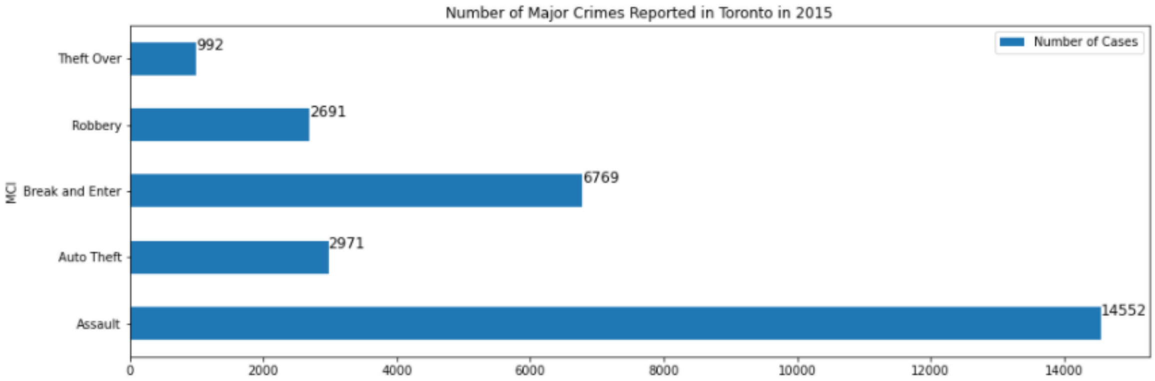
| | Postal Code | Borough | Neighbourhood | Latitude | Longitude | Cluster Label | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7th Most Common Venue | 8th M Comn Ve |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | M5A | Downtown Toronto | Regent Park, Harbourfront | 43.654260 | -79.360636 | 3 | Coffee Shop | Park | Bakery | Breakfast Spot | Mediterranean Restaurant | Restaurant | Farmers Market | Distribu Cei |
| 4 | M7A | Downtown Toronto | Queen's Park, Ontario Provincial Government | 43.662301 | -79.389494 | 3 | Coffee Shop | Park | Yoga Studio | Sushi Restaurant | Bubble Tea Shop | Concert Hall | Distribution Center | D |
| 9 | M5B | Downtown Toronto | Garden District, Ryerson | 43.657162 | -79.378937 | 2 | Theater | Coffee Shop | Plaza | Japanese Restaurant | Electronics Store | Burrito Place | Sandwich Place | ( |
| 15 | M5C | Downtown Toronto | St. James Town | 43.651494 | -79.375418 | 2 | Gastropub | Café | Restaurant | Farmers Market | Japanese Restaurant | Coffee Shop | Cosmetics Shop | BBQ J |
| 19 | M4E | East Toronto | The Beaches | 43.676357 | -79.293031 | 0 | Japanese Restaurant | Park | Beach | Pub | Breakfast Spot | Juice Bar | Bar | Ice Cre S |
| 20 | M5E | Downtown Toronto | Berczy Park | 43.644771 | -79.373306 | 2 | Coffee Shop | Farmers Market | Beer Bar | Cosmetics Shop | Tailor Shop | Restaurant | Lounge | Seaf Restau |
| 24 | M5G | Downtown Toronto | Central Bay Street | 43.657952 | -79.387383 | 3 | Coffee Shop | Café | Italian Restaurant | Clothing Store | Sandwich Place | Ramen Restaurant | Portuguese Restaurant | Poke P |
| 25 | M6G | Downtown | Christie | 43.669542 | -79.422564 | 1 | Café | Coffee | Korean | Grocery | Cocktail Bar | Pub | Spa | Sc Ameri |

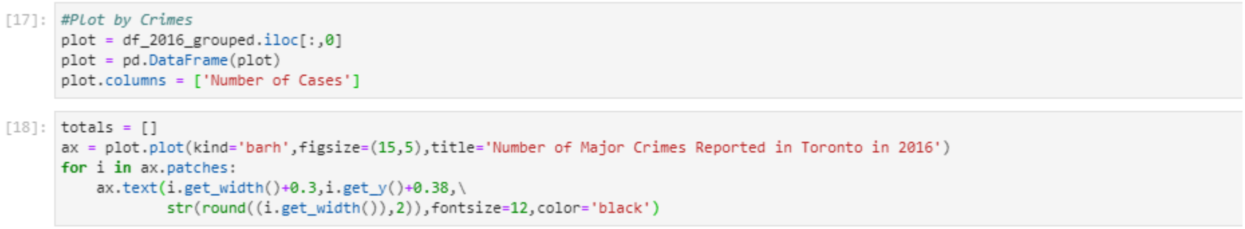
Based on the number of venues the Neighborhood selected was East Toronto, specifically in Upper Beaches, The Upper Beaches area is a Toronto neighborhood that is in the east end of the city. As its name suggests, it is located north of the Beaches/Beach area of the city. The neighborhood is also known as “East End Danforth” or simply “Upper Beach. “The area’s boundaries are Coxwell Avenue in the west, Victoria Park Avenue in the east, Kingston Road in the south and while the northern boundary is generally considered to be the CN tracks between Gerrard Street and Danforth Avenue. In the area north of Kingston Road, between Victoria Park Avenue and Main Street, there is an average of 7.33 crimes against people per 1000 living and working in the area. There is an average of 32.73 crimes against property per 1000.Between Woodbine Avenue and Main Street, still north of Kingston Road, there is an average of 10.49 crimes against people and 21.89 crimes against property. Between Woodbine and Coxwell Avenue, but south of Gerrard Street, there is an average of 12.11 crimes against people and 24.87 crimes against property per 1000. In that same area but north of Gerrard, there is an average of 8.46 crimes against people and 33.13 crimes against property.

The following information have only information from 2015 to 2017 of crimes in the city of Toronto

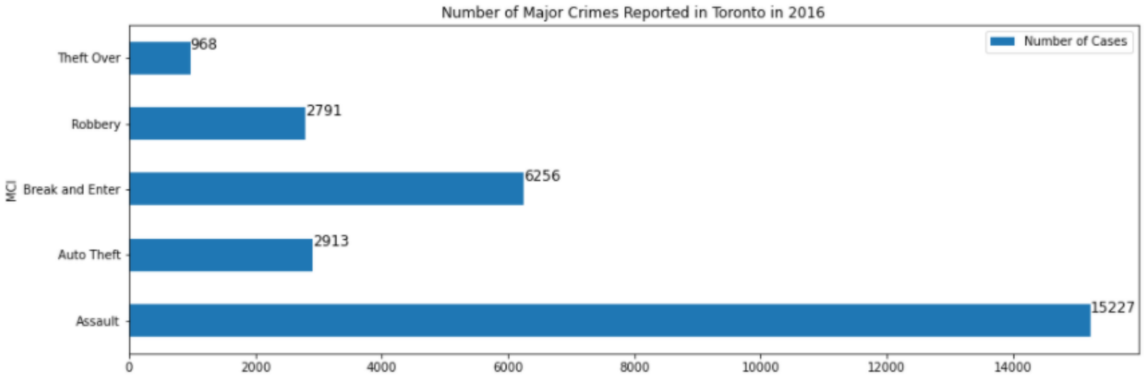


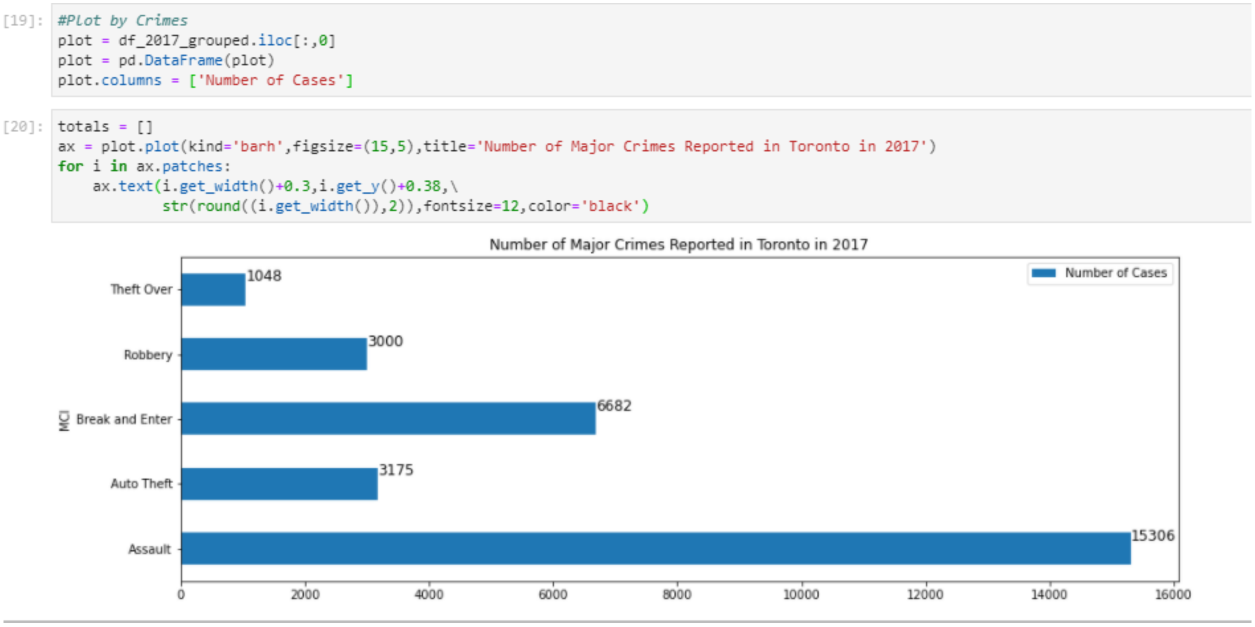
```
[15]: #Plot by Crimes
      plot = df_2015_grouped.iloc[:,0]
      plot = pd.DataFrame(plot)
      plot.columns = ['Number of Cases']

[16]: totals = []
      ax = plot.plot(kind='barh',figsize=(15,5),title='Number of Major Crimes Reported in Toronto in 2015')
      for i in ax.patches:
          ax.text(i.get_width()+0.3,i.get_y()+0.38,\
                  str(round((i.get_width()),2)),fontsize=12,color='black')
```





```
[17]: #Plot by Crimes
      plot = df_2016_grouped.iloc[:,0]
      plot = pd.DataFrame(plot)
      plot.columns = ['Number of Cases']

[18]: totals = []
      ax = plot.plot(kind='barh',figsize=(15,5),title='Number of Major Crimes Reported in Toronto in 2016')
      for i in ax.patches:
          ax.text(i.get_width()+0.3,i.get_y()+0.38,\
                  str(round((i.get_width()),2)),fontsize=12,color='black')
```

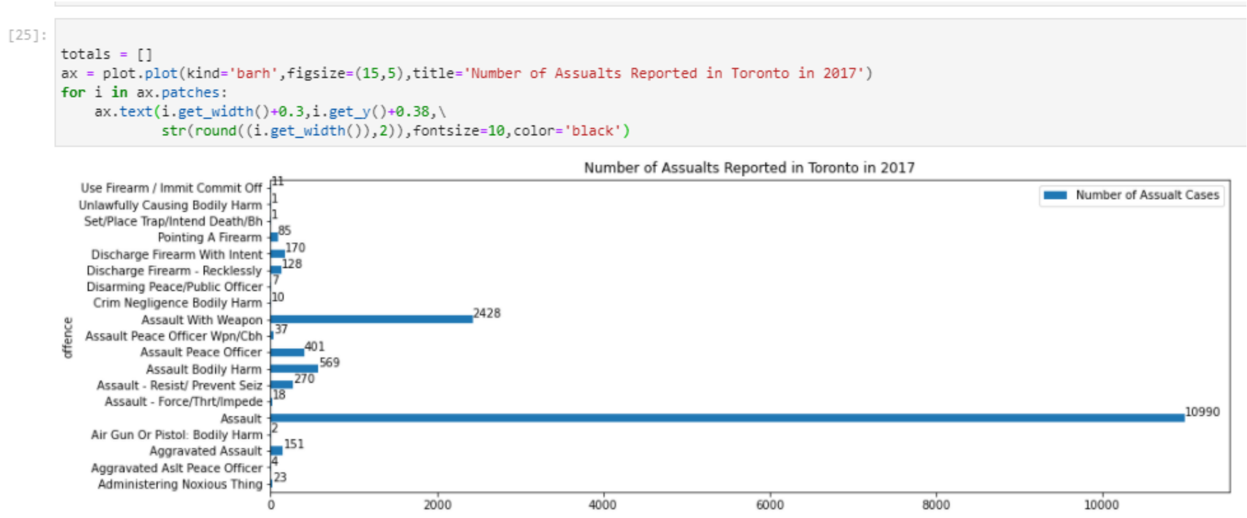




```
[19]: #PLot by Crimes
      plot = df_2017_grouped.iloc[:,0]
      plot = pd.DataFrame(plot)
      plot.columns = ['Number of Cases']
```

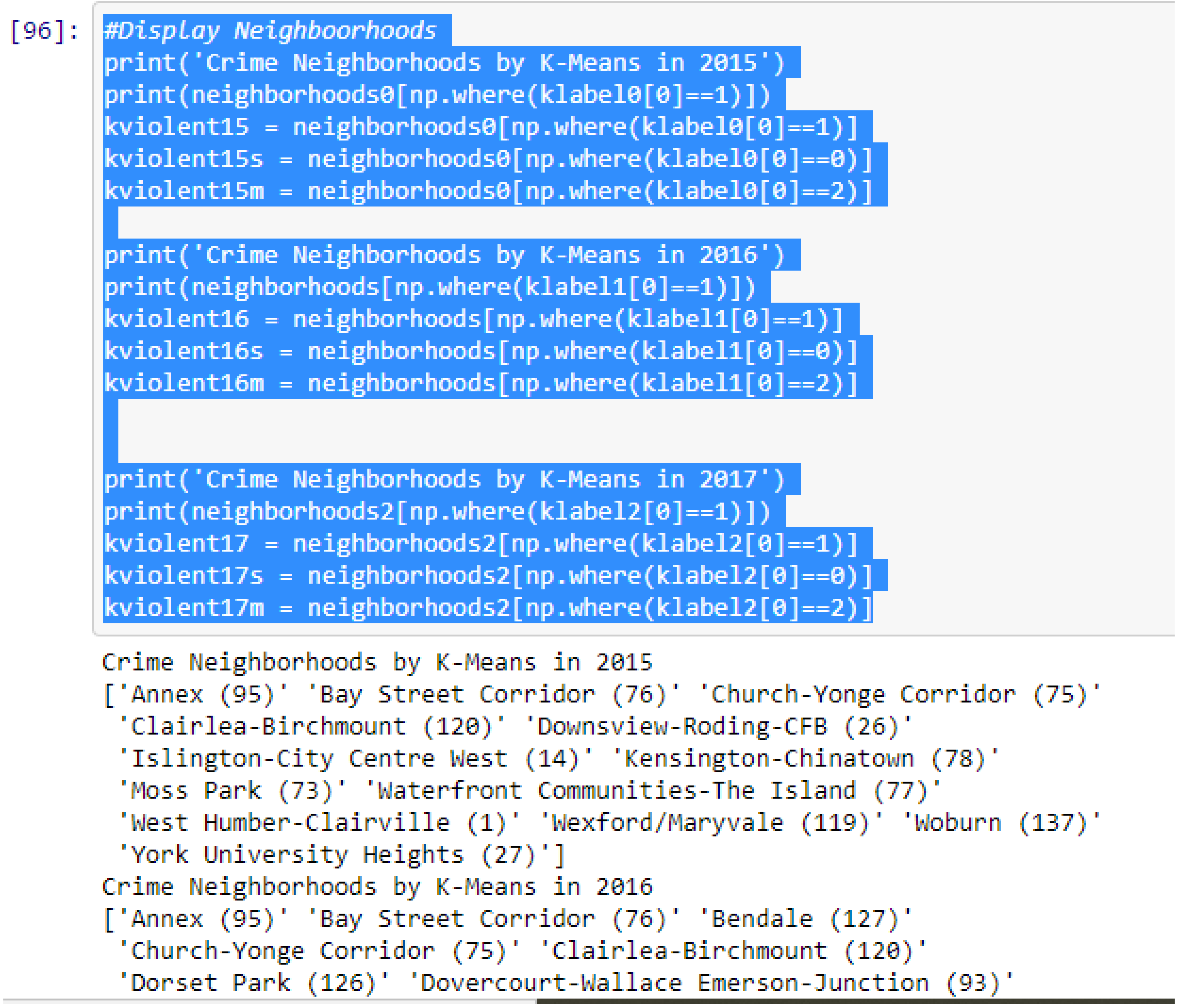
```
[20]: totals = []
      ax = plot.plot(kind='barh',figsize=(15,5),title='Number of Major Crimes Reported in Toronto in 2017')
      for i in ax.patches:
          ax.text(i.get_width()+0.3,i.get_y()+0.38,\
                  str(round((i.get_width()),2)),fontsize=12,color='black')
```

Assault is the major crime by far. The next iteration is the details on the assaults, for that I only display 2017 data:



```
[25]: totals = []
      ax = plot.plot(kind='barh',figsize=(15,5),title='Number of Assaults Reported in Toronto in 2017')
      for i in ax.patches:
          ax.text(i.get_width()+0.3,i.get_y()+0.38,\
                  str(round((i.get_width()),2)),fontsize=10,color='black')
```

Number of Assualts Reported in Toronto in 2017
Number of Assualt Cases
offence
Use Firearm / Immit Commit Off
Unlawfully Causing Bodily Harm
Set/Place Trap/Intend Death/Bh
Pointing A Firearm
Discharge Firearm With Intent
Discharge Firearm - Recklessly
Disarming Peace/Public Officer
Crim Negligence Bodily Harm
Assault With Weapon
Assault Peace Officer Wpn/Cbh
Assault Peace Officer
Assault Bodily Harm
Assault - Resist/ Prevent Seiz
Assault - Force/Thrt/Impede
Assault
Air Gun Or Pistol: Bodily Harm
Aggravated Assault
Aggravated Aslt Peace Officer
Administering Noxious Thing
11
1
1
85
170
128
7
10
2428
37
401
569
270
18
10990
2
151
4
23
0
2000
4000
6000
8000
10000



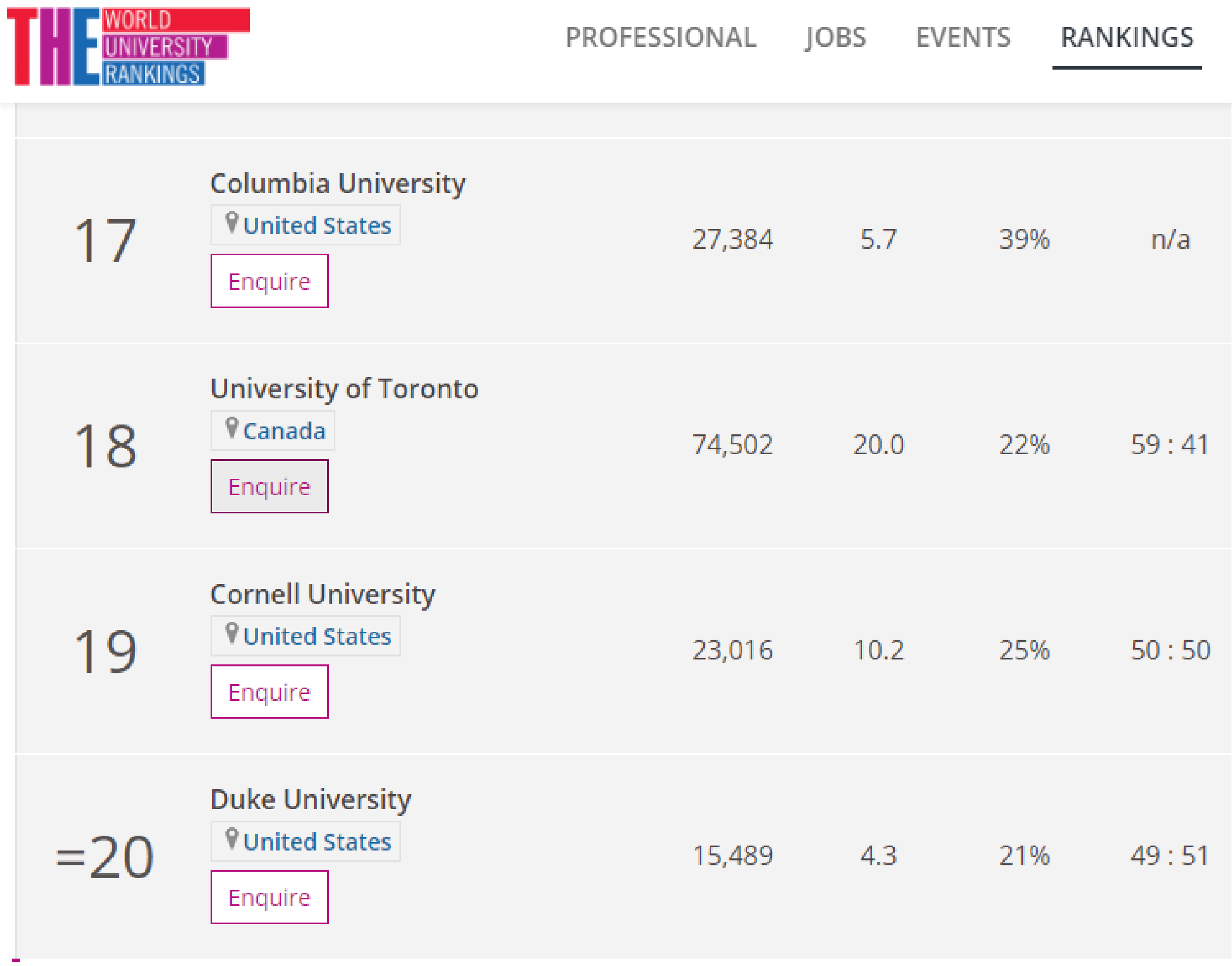
```
[96]: #Display Neighborhoods
print('Crime Neighborhoods by K-Means in 2015')
print(neighborhoods0[np.where(klabel0[0]==1)])
kviolent15 = neighborhoods0[np.where(klabel0[0]==1)]
kviolent15s = neighborhoods0[np.where(klabel0[0]==0)]
kviolent15m = neighborhoods0[np.where(klabel0[0]==2)]

print('Crime Neighborhoods by K-Means in 2016')
print(neighborhoods[np.where(klabel1[0]==1)])
kviolent16 = neighborhoods[np.where(klabel1[0]==1)]
kviolent16s = neighborhoods[np.where(klabel1[0]==0)]
kviolent16m = neighborhoods[np.where(klabel1[0]==2)]


print('Crime Neighborhoods by K-Means in 2017')
print(neighborhoods2[np.where(klabel2[0]==1)])
kviolent17 = neighborhoods2[np.where(klabel2[0]==1)]
kviolent17s = neighborhoods2[np.where(klabel2[0]==0)]
kviolent17m = neighborhoods2[np.where(klabel2[0]==2)]
```

```
Crime Neighborhoods by K-Means in 2015
['Annex (95)' 'Bay Street Corridor (76)' 'Church-Yonge Corridor (75)'
 'Clairlea-Birchmount (120)' 'Downsview-Roding-CFB (26)'
 'Islington-City Centre West (14)' 'Kensington-Chinatown (78)'
 'Moss Park (73)' 'Waterfront Communities-The Island (77)'
 'West Humber-Clairville (1)' 'Wexford/Maryvale (119)' 'Woburn (137)'
 'York University Heights (27)']
Crime Neighborhoods by K-Means in 2016
['Annex (95)' 'Bay Street Corridor (76)' 'Bendale (127)'
 'Church-Yonge Corridor (75)' 'Clairlea-Birchmount (120)'
 'Dorset Park (126)' 'Dovercourt-Wallace Emerson-Junction (93)'
```

The area selected East Toronto, The Upper Beaches using data from 2015 thru 2017, is not as violent as the neighborhoods displayed above.

4. Discussion section where you discuss any observations you noted and any recommendations you can make based on the results.

Based on the results from the data manipulation, and after analyzing the Universities in Toronto, the University selected based on the number of votes was University of Toronto, this is an old and prestigious University, ranking globally as number 18. And in terms of the yearly invest, the amount of money is less than $10,000 less than the US counterparts. (See the below table with the rank information).

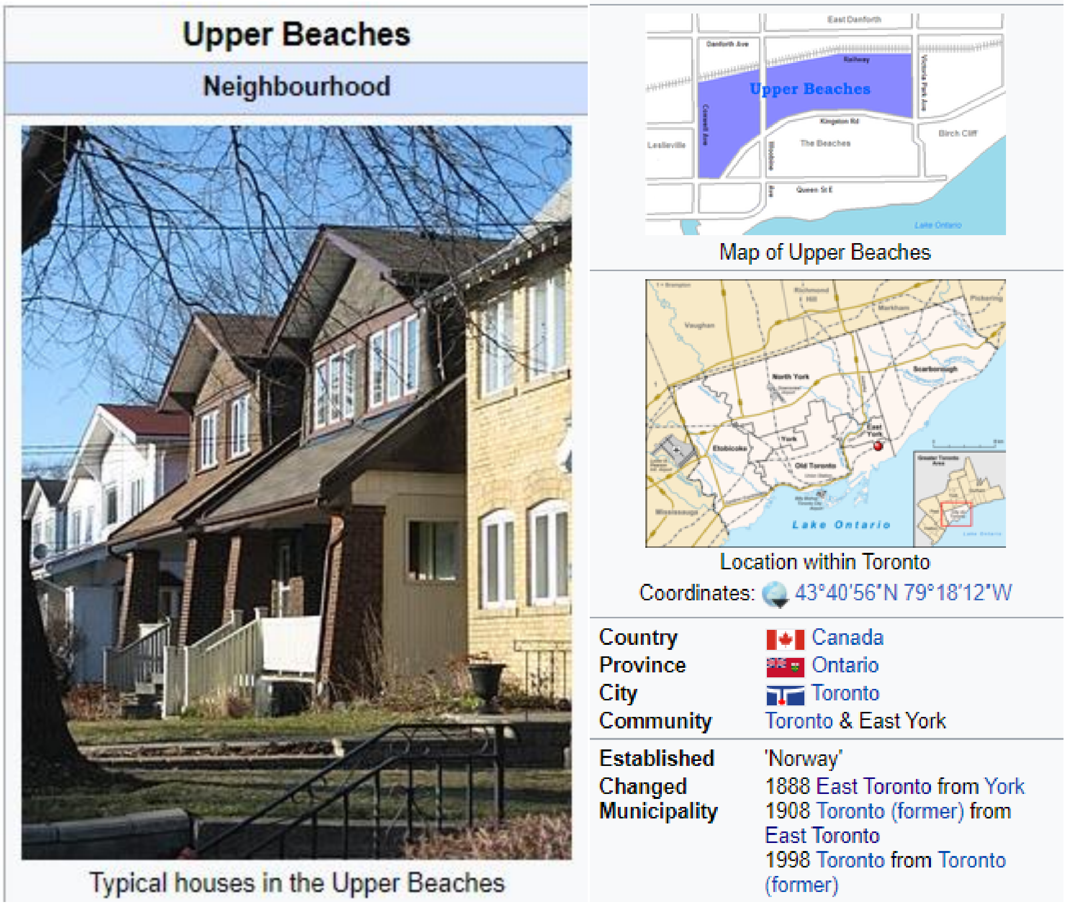


| | | | | | |
|---|---|---|---|---|---|
| 17 | Columbia University<br>United States<br>Enquire | 27,384 | 5.7 | 39% | n/a |
| 18 | University of Toronto<br>Canada<br>Enquire | 74,502 | 20.0 | 22% | 59 : 41 |
| 19 | Cornell University<br>United States<br>Enquire | 23,016 | 10.2 | 25% | 50 : 50 |
| =20 | Duke University<br>United States<br>Enquire | 15,489 | 4.3 | 21% | 49 : 51 |

Imparting Computer Science, that is a major that my older kid prefers. Once the University was selected, the next step was to consider all the neighborhoods near the University with the following criteria:

1. Venues nearby (Groceries, Supermarkets, etc.)
2. Crime statistics (Assaults specifically as my kids are planning on use public transportation for movement in and out of the campus)

Once finishing the analysis the area selected was East Toronto, specifically the "Upper Beaches" area, having FourSquare for supplementing the analysis was very important, the analytical approach followed ensure me of a successfully selection of the area, and to have the feedback of people referring to the University and the other venues.



**Upper Beaches**

**Neighbourhood**

Typical houses in the Upper Beaches

Map of Upper Beaches

Location within Toronto

Coordinates: 43°40′56″N 79°18′12″W

| | |
|---|---|
| **Country** | Canada |
| **Province** | Ontario |
| **City** | Toronto |
| **Community** | Toronto & East York |
| **Established** | 'Norway' |
| **Changed Municipality** | 1888 East Toronto from York<br>1908 Toronto (former) from East Toronto<br>1998 Toronto from Toronto (former) |

Is a nice and old neighborhood. The last analysis of the report was focused on the crime rates on the neighborhoods around the University campus, and using data from 2015 to 2017, the Upper Beaches is one with the least criminal rates, as a matter of fact the top 15 list not include the Upper Beaches.

5. Conclusion section where you conclude the report.

The work of a Data Scientist is to get the correct and specific question every time that he/she started an analysis. In this case the questions were clear:

- Which University in Toronto is one of the best?
- Which Neighborhood is the safest to live and commute to/from the University?
- Can we use statistics to determine the safest place?

The use of tools was exceptionally good:

- Jupyter Notebook
- Python
- FourSquare
- GoogleMaps API
- Matplotlib
- Datasets
- IBM Cloud
- Github

- Statistical Analysis

As a non-System Engineer, I was amazed that the availability of open source information and the free access tools, like Watson Studio, GitHub, and Jupyter Notebook. The free code available, like the plotting capabilities od Matplotlib. Anyway, the approach that I follow for selecting the University and Neighborhood, using for the later statistical analysis, base on the criminal key indicators, allow me and my fellow colleagues on this certification a level of granularity that ensure a proper backup for the decision make process. The selection a specific University based on the ratings made by the users, allow me to do an educated analysis, and to perform a problem solving, using a quantitative analysis. The possibility to locate with the coordinates any venue, and perform a detail analysis with flexible data selection, to reach for a conclusion it’s paramount.