Toronto Health Seminar - Searching Venues of interest for Delegates and Families

Introduction

Toronto has been hosting its famous health congregation Seminar since 2010. Its the time of the year when health professionals converge in hordes to participate, learn and share the latest technologies and breakthroughs in their field. It's a Five-day event and the professionals are also accompanied by their families. This year is no exception. However, with the COVID-19 threat the seminar has been postponed to August, considering that the threat is reduced by that time. The event management company entrusted with the logistics of this event has employed me to provide them with data points and map that would assist them in arriving at probable logistics with regards to:

1. Hotels as Accommodation
2. Shopping malls and Event places
3. Parks,Restaurant,Cafetaria for family recreation

This Project Therefore would have the objective of looking at such data points,clean them,merge them into probable sets and visualize the same with maps.

Import Libraries ¶

In [2]:
```
import requests # to handle requests import
 pandas as pd # for data analsysis
 import numpy as np # to handle data in a vectorized manner

 !conda install -c conda-forge geopy --yes  from geopy.geocoders import
Nominatim # module to convert an address into latitu de 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

 !conda install -c conda-forge folium=0.5.0 --yes
 import folium # plotting library
```

```
Solving environment: done

## Package Plan ##

  environment location:
/opt/conda/envs/Python36    added / updated
specs:
    - geopy


The following packages will be downloaded:

    package                    |            build     ---------------------------
|-----------------
    geographiclib-1.50         |             py_0          34 KB  conda-forge
ca-certificates-2019.11.28 |        hecc5488_0          145 KB  conda-forge
geopy-1.21.0                |              py_0           58 KB  conda-forge
openssl-1.1.1d              |       h516909a_0          2.1 MB  conda-forge
python_abi-3.6              |           1_cp36m            4 KB  conda-forge
certifi-2019.11.28         |   py36h9f0ad1d_1         149 KB  conda-forge
    ------------------------------------------------------------
                                          Total:         2.5 MB

The following NEW packages will be INSTALLED:

    geographiclib:   1.50-py_0         conda-forge
geopy:           1.21.0-py_0       conda-forge
python_abi:      3.6-1_cp36m       conda-forge

The following packages will be UPDATED:

    ca-certificates: 2019.11.27-0                 --> 2019.11.28-hecc5488_0
conda-forge
    certifi:         2019.11.28-py36_0            --> 2019.11.28-py36h9f0ad1d_1
conda-forge

The following packages will be DOWNGRADED:

    openssl:         1.1.1d-h7b6447c_3            --> 1.1.1d-h516909a_0
conda-forge


Downloading and Extracting Packages
geographiclib-1.50   | 34 KB     | ##################################### | 100%
ca-certificates-2019 | 145 KB    | ##################################### | 100%
geopy-1.21.0         | 58 KB     | ##################################### | 100%
openssl-1.1.1d       | 2.1 MB    | ##################################### | 100%
python_abi-3.6       | 4 KB      | ##################################### | 100%
certifi-2019.11.28   | 149 KB    | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done
```

```
## Package Plan ##

  environment location:
/opt/conda/envs/Python36    added / updated
specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build     ---------------------------
|-----------------
    altair-4.0.1               |             py_0         575 KB  conda-forge
branca-0.4.0                 |              py_0          26 KB  conda-forge
vincent-0.4.4                |              py_1          28 KB  conda-forge
folium-0.5.0               |            py_0          45 KB  conda-forge
    ------------------------------------------------------------
                                          Total:         673 KB

The following NEW packages will be INSTALLED:

    altair:  4.0.1-py_0 conda-forge
branca:    0.4.0-py_0  conda-forge
folium:    0.5.0-py_0  conda-forge
vincent: 0.4.4-py_1 conda-forge


  Downloading and Extracting Packages
altair-4.0.1         | 575 KB    | ##################################### | 100%
branca-0.4.0         | 26 KB     | ##################################### | 100%
vincent-0.4.4        | 28 KB     | ##################################### | 100%
folium-0.5.0         | 45 KB     | ##################################### | 100%
  Preparing transaction: done
  Verifying transaction: done
  Executing transaction: done
```

Mapping out Credentials in Foursquare API

In [3]:
```
ClIENT_ID = 'NGC2LWZF34JVJEF042ZFUKN1TKN42ESDDJGADQMYE0QXCLVN' # your Foursquare
ID
ClIENT_SECRET = 'G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFED4WU' # your Foursq
uare Secret
VERSION = '20180604' LIMIT
=30
print('Your credentails:') print('Foursquare_ID:
' + ClIENT_ID) print('Foursquare_Secret:' +
ClIENT_SECRET)
```

```
Your credentails:
Foursquare_ID: NGC2LWZF34JVJEF042ZFUKN1TKN42ESDDJGADQMYE0QXCLVN
Foursquare_Secret:G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFED4WU
```

City Definition and Latitude and Longitude retrieval

In [4]:
```
# define the city and get its latitude & longitude
city = 'Toronto'
geolocator = Nominatim(user_agent="My-aaplication")
location = geolocator.geocode(city)
latitude = location.latitude
longitude = location.longitude
print("Geographical Cordinates of the City of {} is {},{}".format(city,latitude,
longitude))
```

```
Geographical Cordinates of the City of Toronto is 43.653963,-79.387207
```

Accomodation/Hotel Search

In [5]:
```
# search for hotels
search_query = 'Hotel'
radius = 500

# Define the corresponding URL
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}
&ll={},{}&v={}&query={}&radius={}&limit={}'\
.format(ClIENT_ID, ClIENT_SECRET, latitude, longitude, VERSION, search_query, ra
dius, LIMIT)
url
```

Out[5]:
```
'https://api.foursquare.com/v2/venues/search?client_id=NGC2LWZF34JVJEF042ZFUKN1TK
N42ESDDJGADQMYE0QXCLVN&client_secret=G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFE
D4WU&ll=43.653963,-79.387207&v=20180604&query=Hotel&radius=500&limit=30'
```

In [44]:
```
# GET Request and examine the results
results = requests.get(url).json()
#results
```

Out[7]:

In [7]:
```
# assign relevant part of JSON to venues
venues = results['response']['venues']

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

	categories	hasPerk	id	location.address	loca
0	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	False	4ab2d511f964a5209b6c20e3	123 Queen Street West	CA
1	[{'id': '4bf58dd8d48988d1e7931735', 'name': 'J...	False	4b68aed1f964a520de862be3	194 Queen St W	CA
2	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	False	4f343a31e4b0230a3b337a90	123 Test Drive	CA
3	[{'id': '56aa371be4b08b9a8d5734cf', 'name': 'B...	False	5545d07e498e2facac03f666	123 Queen Street West	CA
4	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	False	4f53fb2ee4b036244bea152f	NaN	CA

Cleaning the data frame

Trivia - Cleaning consumes 80% of a Data Scientists Workload Out[8]:

In [8]:
```
# keep only columns that include venue name, and anything that is associated wit
h location
clean_columns = ['name', 'categories'] + [col for col in dataframe.columns if co
l.startswith('location.')]+ ['id']
clean_dataframe = dataframe.loc[:,clean_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
clean_dataframe['categories'] = clean_dataframe.apply(get_category_type, axis=1)

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

clean_dataframe.head()
```

	name	categories	address	cc	city	country	crossStreet	distance	formattedAddr
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	CA	Toronto	Canada	at York St.	432	[123 Queen Str West (at York S Toronto ...
1	The Rex Hotel Jazz & Blues Bar	Jazz Club	194 Queen St W	CA	Toronto	Canada	Queen & St. Patrick	400	[194 Queen St (Queen & St. Patrick), Toront
2	VFM Test Hotel	Hotel	123 Test Drive	CA	Toronto	Canada	at somewhere St	500	[123 Test Drive somewhere St) Toronto ON ...
3	Sheraton Centre Toronto Hotel - Grand Ballroom	Ballroom	123 Queen Street West	CA	Toronto	Canada	at Bay Street	376	[123 Queen Str West (at Bay Street), Toront.
4	Stathcona Hotel	Hotel	NaN	CA	Toronto	Canada	NaN	129	[Toronto ON, Canada]

In [9]:
```
# delete unnecessary columns clean_dataframe2= clean_dataframe.drop(['cc',
 'city', 'country', 'crossStreet', 'distance', 'formattedAddress',\
                                          'labeledLatLngs','neighborhood', 'id'],
axis=1) clean_dataframe2
```

	name	categories	address	lat	lng	postalCode	state
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	43.650594	-79.384530	M5H 2M9	ON
1	The Rex Hotel Jazz & Blues Bar	Jazz Club	194 Queen St W	43.650505	-79.388577	M5V 1Z1	ON
2	VFM Test Hotel	Hotel	123 Test Drive	43.658434	-79.387894	M2M 2M2	ON
3	Sheraton Centre Toronto Hotel - Grand Ballroom	Ballroom	123 Queen Street West	43.651200	-79.384520	NaN	ON
4	Stathcona Hotel	Hotel	NaN	43.654947	-79.386359	NaN	ON
5	Sheraton Centre Hotel Club Lounge	Lounge	NaN	43.651063	-79.384527	M5H 2M9	ON
6	650 Hotel	Hotel	650 Bay Street	43.657046	-79.384411	NaN	ON
7	Be SixFifty Hotel	Hotel	650 Bay Street	43.657120	-79.384560	NaN	ON
8	DoubleTree by Hilton	Hotel	108 Chestnut Street	43.654608	-79.385942	M5G 1R3	ON
9	Shangri-La Toronto	Hotel	188 University Ave.	43.649129	-79.386557	M5H 0A3	ON
10	Hilton	Hotel	145 Richmond St W	43.649946	-79.385479	M5H 2L2	ON
11	Marriott Downtown at CF Toronto Eaton Centre	Hotel	525 Bay Street	43.654728	-79.382422	M5G 2L2	ON
12	89 Chestnut Residence	College Residence Hall	89 Chestnut St.	43.654160	-79.385291	M5G 1R1	ON

Out[9]:

13	Op mijn hotelkamer In Toronto	Hotel	NaN	43.654235	-79.386208	NaN	ON
14	Lobby Lounge at the Shangri-La Toronto	Lounge	188 University Ave.	43.649155	-79.386546	M5H 0A3	ON
15	Fitness Centre	Gym	Sheraton Centre	43.650985	-79.384002	M5H 2M9	ON
16	Fitness Centre	Gym	525 Bay St.	43.654690	-79.381739	M5G 2L2	ON
17	Pool	Pool	Intercontinental	43.650975	-79.384053	NaN	Ontario

	name	categories	address	lat	lng	postalCode	state
18	Kumar For Men	Men's Store	Sheraton Centre	43.651044	-79.383302	M5H 2M9	ON
19	Grand Ballroom	Event Space	123 Queen St. W	43.651217	-79.383771	M5H 2M9	ON
20	Bistro on Two	Breakfast Spot	123 Queen St. W	43.651011	-79.383658	M5H 2M9	ON

In [10]:

```
# delete rows with none values
clean_dataframe3 = clean_dataframe2.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
clean_dataframe3
```

Out[10]:

	name	categories	address	lat	lng	postalCode	state
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	43.650594	-79.384530	M5H 2M9	ON
1	The Rex Hotel Jazz & Blues Bar	Jazz Club	194 Queen St W	43.650505	-79.388577	M5V 1Z1	ON
2	VFM Test Hotel	Hotel	123 Test Drive	43.658434	-79.387894	M2M 2M2	ON
8	DoubleTree by Hilton	Hotel	108 Chestnut Street	43.654608	-79.385942	M5G 1R3	ON
9	Shangri-La Toronto	Hotel	188 University Ave.	43.649129	-79.386557	M5H 0A3	ON
10	Hilton	Hotel	145 Richmond St W	43.649946	-79.385479	M5H 2L2	ON
11	Marriott Downtown at CF Toronto Eaton Centre	Hotel	525 Bay Street	43.654728	-79.382422	M5G 2L2	ON
12	89 Chestnut Residence	College Residence Hall	89 Chestnut St.	43.654160	-79.385291	M5G 1R1	ON
14	Lobby Lounge at the Shangri-La Toronto	Lounge	188 University Ave.	43.649155	-79.386546	M5H 0A3	ON
15	Fitness Centre	Gym	Sheraton Centre	43.650985	-79.384002	M5H 2M9	ON
16	Fitness Centre	Gym	525 Bay St.	43.654690	-79.381739	M5G 2L2	ON
18	Kumar For Men	Men's Store	Sheraton Centre	43.651044	-79.383302	M5H 2M9	ON
19	Grand Ballroom	Event Space	123 Queen St. W	43.651217	-79.383771	M5H 2M9	ON
20	Bistro on Two	Breakfast Spot	123 Queen St. W	43.651011	-79.383658	M5H 2M9	ON

In [11]:

```
# delete rows which its category is not Hotel or Event Space
array= ['Hotel', 'Event Space']
hotel_dataframe= clean_dataframe3.loc[clean_dataframe3['categories'].isin(array
)]
hotel_dataframe
```

Out[11]:

	name	categories	address	lat	lng	postalCode	state
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	43.650594	-79.384530	M5H 2M9	ON
2	VFM Test Hotel	Hotel	123 Test Drive	43.658434	-79.387894	M2M 2M2	ON
8	DoubleTree by Hilton	Hotel	108 Chestnut Street	43.654608	-79.385942	M5G 1R3	ON
9	Shangri-La Toronto	Hotel	188 University Ave.	43.649129	-79.386557	M5H 0A3	ON
10	Hilton	Hotel	145 Richmond St W	43.649946	-79.385479	M5H 2L2	ON
11	Marriott Downtown at CF Toronto Eaton Centre	Hotel	525 Bay Street	43.654728	-79.382422	M5G 2L2	ON
19	Grand Ballroom	Event Space	123 Queen St. W	43.651217	-79.383771	M5H 2M9	ON

In [12]:

```
# delete rows which has duplicate hotel's name
df_hotels = hotel_dataframe.drop_duplicates(subset='name', keep="first")
df_hotels
```

	name	categories	address	lat	lng	postalCode	state
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	43.650594	-79.384530	M5H 2M9	ON
2	VFM Test Hotel	Hotel	123 Test Drive	43.658434	-79.387894	M2M 2M2	ON
8	DoubleTree by Hilton	Hotel	108 Chestnut Street	43.654608	-79.385942	M5G 1R3	ON
9	Shangri-La Toronto	Hotel	188 University Ave.	43.649129	-79.386557	M5H 0A3	ON
10	Hilton	Hotel	145 Richmond St W	43.649946	-79.385479	M5H 2L2	ON
11	Marriott Downtown at CF Toronto Eaton Centre	Hotel	525 Bay Street	43.654728	-79.382422	M5G 2L2	ON
19	Grand Ballroom	Event Space	123 Queen St. W	43.651217	-79.383771	M5H 2M9	ON

In [13]:

```
# choose the hotel which has the same postalCode with the event space
df_hotel = df_hotels[df_hotels.postalCode == 'M5H 2M9']
df_hotel
```

	name	categories	address	lat	lng	postalCode	state
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	43.650594	-79.384530	M5H 2M9	ON
19	Grand Ballroom	Event Space	123 Queen St. W	43.651217	-79.383771	M5H 2M9	ON

Out[12]:

In [13]:
Out[13]:

Time for Fresh air and family time - Parks Please

In [14]:
```
# search for Parks
search_query = 'Park'
radius = 10000

# Define the corresponding URL
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}
&ll={},{}&v={}&query={}&radius={}&limit={}'\
.format(ClIENT_ID, ClIENT_SECRET, latitude, longitude, VERSION, search_query, ra
dius, LIMIT)
url
```

Out[14]:
```
'https://api.foursquare.com/v2/venues/search?client_id=NGC2LWZF34JVJEF042ZFUKN1TK
N42ESDDJGADQMYE0QXCLVN&client_secret=G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFE
D4WU&ll=43.653963,-79.387207&v=20180604&query=Park&radius=10000&limit=30'
```

Out[16]:

In [43]:
```
# Send the GET Request and examine the results
presults = requests.get(url).json()
#presults
```

In [16]:
```
# assign relevant part of JSON to venues
venues = presults['response']['venues']

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

	categories	hasPerk	id	location.address	locat
0	[{'id': '4bf58dd8d48988d1fd931735', 'name': 'M...	False	4b170e47f964a5208cc123e3	691 University Ave	CA
1	[{'id': '4bf58dd8d48988d163941735', 'name': 'P...	False	4b54deadf964a520a3d027e3	NaN	CA
2	[{'id': '4bf58dd8d48988d163941735', 'name': 'P...	False	4b9d206bf964a520e69136e3	University Ave.	CA
3	[{'id': '4bf58dd8d48988d163941735', 'name': 'P...	False	4bb79860cf2fc9b616779e02	35 Wellington St. East	CA
4	[{'id': '4bf58dd8d48988d163941735', 'name': 'P...	False	4bfc5019da7120a132df49fd	450 Adelaide Street West	CA

Sweeping down the Park Data

Out[17]:

In [17]:
```
# keep only columns that include venue name, and anything that is associated wit
h location
park_clean_columns = ['name', 'categories'] + [col for col in park_dataframe.col
umns if col.startswith('location.')]+ ['id']
clean_park_dataframe = park_dataframe.loc[:,park_clean_columns]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list1 = row['categories']
    except:
        categories_list1 = row['venue.categories']

    if len(categories_list1) == 0:
        return None
    else:
        return categories_list1[0]['name']

# filter the category for each row
clean_park_dataframe['categories'] = clean_park_dataframe.apply(get_category_typ
e, axis=1)

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

clean_park_dataframe.head()
```

	name	categories	address	cc	city	country	crossStreet	distance	formattedA
0	Queen's Park Subway Station	Metro Station	691 University Ave	CA	Toronto	Canada	at College St	716	[691 Univers (at College S Toronto O...
1	Grange Park	Park	NaN	CA	Toronto	Canada	NaN	423	[btwn Colleg University, C & Spadina, T
2	Queen's Park	Park	University Ave.	CA	Toronto	Canada	at Wellesley Ave.	1181	[University A Wellesley Av Toronto ...
3	Berczy Park	Park	35 Wellington St. East	CA	Toronto	Canada	NaN	1171	[35 Wellingt East, Toront Canada]
4	St. Andrews Playground / Dog Park	Park	450 Adelaide Street West	CA	Toronto	Canada	Brant St & Adelaide St W	1183	[450 Adelaic Street West St & Adelaic

In [18]:
```
# delete unnecessary columns clean_park_dataframe2=
clean_park_dataframe.drop(['cc', 'city', 'country', 'cros sStreet', 'distance',
'formattedAddress',\
                                          'labeledLatLngs', 'id'], axis=1)
clean_park_dataframe2
```

	name	categories	address	lat	lng	postalCode	state
0	Queen's Park Subway Station	Metro Station	691 University Ave	43.660006	-79.390272	M5G 2P1	ON
1	Grange Park	Park	NaN	43.652488	-79.392053	NaN	ON
2	Queen's Park	Park	University Ave.	43.663946	-79.392180	M5R 2E8	ON
3	Berczy Park	Park	35 Wellington St. East	43.648048	-79.375172	NaN	ON
4	St. Andrews Playground / Dog Park	Park	450 Adelaide Street West	43.647388	-79.398752	NaN	ON
5	Residences at College Park North	Residential Building (Apartment / Condo)	763 Bay St.	43.659822	-79.385159	M5G 2R3	ON
6	One Park Lane	Residential Building (Apartment / Condo)	280 Simcoe St	43.655442	-79.390056	M5T 2Y5	ON
7	Crunch Fitness - College Park	Gym / Fitness Center	382 Yonge Street	43.659263	-79.382632	M5G 1S8	ON
8	TTC Streetcar #403 - Victoria park	Light Rail Station	NaN	43.650735	-79.386921	NaN	ON
9	Clarence Square Park	Park	Spadina Ave	43.644222	-79.394230	NaN	ON
10	Bellevue Square Park	Park	btwn Bellevue & Augusta Ave.	43.653610	-79.402199	M5T 2N4	ON
11	Physiomed College Park	Medical Center	382 Yonge St	43.659240	-79.382860	M5B 1S8	ON

Out[18]:

12	Toothworks College Park Dental	Dentist's Office	444 Yonge St, Unit M16	43.660009	-79.383905	NaN	ON
13	RBC WaterPark Place	Office	88 Queens Quay	43.641075	-79.379154	NaN	ON
14	College Park	Shopping Mall	444 Yonge St	43.661237	-79.383603	M5B 2H4	ON
15	Canoe Landing Park	Park	50 Fort York Blvd	43.638762	-79.397067	M5V 3Z1	ON

	name	categories	address	lat	lng	postalCode	state
16	St Lawrence on the Park	Building	65 Scadding Ave	43.648963	-79.364868	NaN	ON
17	Trinity Bellwoods Park	Park	1053 Dundas St. W.	43.647072	-79.413756	M5H 2N2	ON
18	HTO Park	Park	Queen's Quay	43.637949	-79.387820	NaN	ON
19	Village of Yorkville Park	Park	115 Cumberland St.	43.670186	-79.391180	NaN	ON
20	Regent Park	Neighborhood	NaN	43.660009	-79.364191	NaN	ON
21	Riverdale Park West	Park	500 Gerrard St.	43.666048	-79.360941	M5A 2H3	ON
22	Barbara Hall Park	Park	519 Church St	43.666879	-79.381068	M4Y 2K9	ON
23	Regent Park Community Centre	Community Center	465 Dundas St E	43.659512	-79.363048	M5A 2B2	ON
24	High Park	Park	1873 Bloor St. W	43.646479	-79.463425	M6R 2Z3	ON
25	Cawthra Square Dog Park	Dog Run	519 Church St.	43.666583	-79.380040	NaN	ON
26	Parliament Square Park	Park	44 Parliament Street	43.650264	-79.362195	NaN	ON
27	Moss Park Tennis Courts	Tennis Court	Moss Park	43.655337	-79.371923	NaN	ON
28	Stanley Park	Park	King Street West	43.642395	-79.409330	NaN	ON
29	Riverdale Park East	Park	550 Broadview Ave	43.669951	-79.355493	M4K 2P1	ON

In [19]:
```
# delete rows with none values clean_park_dataframe3 =
clean_park_dataframe2.dropna(axis=0, how='any', thresh=N one, subset=None,
inplace=False) clean_park_dataframe3
```

Out[19]:

	name	categories	address	lat	lng	postalCode	state
0	Queen's Park Subway Station	Metro Station	691 University Ave	43.660006	-79.390272	M5G 2P1	ON
2	Queen's Park	Park	University Ave.	43.663946	-79.392180	M5R 2E8	ON
5	Residences at College Park North	Residential Building (Apartment / Condo)	763 Bay St.	43.659822	-79.385159	M5G 2R3	ON
6	One Park Lane	Residential Building (Apartment / Condo)	280 Simcoe St	43.655442	-79.390056	M5T 2Y5	ON
7	Crunch Fitness - College Park	Gym / Fitness Center	382 Yonge Street	43.659263	-79.382632	M5G 1S8	ON
10	Bellevue Square Park	Park	btwn Bellevue & Augusta Ave.	43.653610	-79.402199	M5T 2N4	ON
11	Physiomed College Park	Medical Center	382 Yonge St	43.659240	-79.382860	M5B 1S8	ON
14	College Park	Shopping Mall	444 Yonge St	43.661237	-79.383603	M5B 2H4	ON
15	Canoe Landing Park	Park	50 Fort York Blvd	43.638762	-79.397067	M5V 3Z1	ON
17	Trinity Bellwoods Park	Park	1053 Dundas St. W.	43.647072	-79.413756	M5H 2N2	ON
21	Riverdale Park West	Park	500 Gerrard St.	43.666048	-79.360941	M5A 2H3	ON
22	Barbara Hall Park	Park	519 Church St	43.666879	-79.381068	M4Y 2K9	ON
23	Regent Park Community Centre	Community Center	465 Dundas St E	43.659512	-79.363048	M5A 2B2	ON
24	High Park	Park	1873 Bloor St. W	43.646479	-79.463425	M6R 2Z3	ON
29	Riverdale Park East	Park	550 Broadview Ave	43.669951	-79.355493	M4K 2P1	ON

Out[20]:

In [20]:
```
# delete rows which its category is not Park
df_park = clean_park_dataframe3[clean_park_dataframe3.categories == 'Park']
df_park
```

	name	categories	address	lat	lng	postalCode	state
2	Queen's Park	Park	University Ave.	43.663946	-79.392180	M5R 2E8	ON
10	Bellevue Square Park	Park	btwn Bellevue & Augusta Ave.	43.653610	-79.402199	M5T 2N4	ON
15	Canoe Landing Park	Park	50 Fort York Blvd	43.638762	-79.397067	M5V 3Z1	ON
17	Trinity Bellwoods Park	Park	1053 Dundas St. W.	43.647072	-79.413756	M5H 2N2	ON
21	Riverdale Park West	Park	500 Gerrard St.	43.666048	-79.360941	M5A 2H3	ON
22	Barbara Hall Park	Park	519 Church St	43.666879	-79.381068	M4Y 2K9	ON
24	High Park	Park	1873 Bloor St. W	43.646479	-79.463425	M6R 2Z3	ON
29	Riverdale Park East	Park	550 Broadview Ave	43.669951	-79.355493	M4K 2P1	ON

Need for FOOD - Restaurants

In [21]:
```
# search for Restaurants
search_query = 'Restaurant'
radius = 10000

# Define the corresponding URL
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}
&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, lat
itude, longitude, VERSION, search_query, radius, LIMIT) url
```

Out[21]:
```
'https://api.foursquare.com/v2/venues/search?client_id=NGC2LWZF34JVJEF042ZFUKN1TK
N42ESDDJGADQMYE0QXCLVN&client_secret=G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFE
D4WU&ll=43.653963,-79.387207&v=20180604&query=Restaurant&radius=10000&limit=30'
```

In [45]:
```
# Send the GET Request and examine the results
Rresults = requests.get(url).json()
#Rresults
```

Out[23]:

In [23]:
```
# assign relevant part of JSON to venues
venues = Rresults['response']['venues']

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

	categories	hasPerk	id	location.address	locat
0	[{'id': '4bf58dd8d48988d14e941735', 'name': 'A...	False	4ad4c05ff964a52048f720e3	110 Chestnut Street	CA
1	[{'id': '4bf58dd8d48988d1c4941735', 'name': 'R...	False	4b223f5af964a520ba4424e3	225 Front St W	CA
2	[{'id': '4bf58dd8d48988d123941735', 'name': 'W...	False	4ad4c05cf964a520dff520e3	301 Front St W	CA
3	[{'id': '4bf58dd8d48988d1d1941735', 'name': 'N...	False	4b266f05f964a520657b24e3	266 Spadina Ave	CA
4	[{'id': '4bf58dd8d48988d1f5931735', 'name': 'D...	False	4ad4c060f964a5207ff720e3	323 Spadina Ave.	CA

Restaurant Cleanup

Out[24]:

In [24]:
```
# keep only columns that include venue name, and anything that is associated wit
h location
Restaurant_clean_columns = ['name', 'categories'] + [col for col in Restaurant_d
ataframe.columns if col.startswith('location.')]+ ['id']
clean_Restaurant_dataframe = Restaurant_dataframe.loc[:,Restaurant_clean_columns
]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list3 = row['categories']
    except:
        categories_list3 = row['venue.categories']

    if len(categories_list3) == 0:
        return None
    else:
        return categories_list3[0]['name']

# filter the category for each row
clean_Restaurant_dataframe['categories'] = clean_Restaurant_dataframe.apply(get_
category_type, axis=1)

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

clean_Restaurant_dataframe.head()
```

	name	categories	address	cc	city	country	crossStreet	distance	formatte
0	Hemispheres Restaurant & Bistro	American Restaurant	110 Chestnut Street	CA	Toronto	Canada	NaN	145	[110 Che Street, T M5G 1R
1	Azure Restaurant & Bar	Restaurant	225 Front St W	CA	Toronto	Canada	in InterContinental Toronto Centre	1039	[225 Fro InterCor Toronto
2	360 Restaurant	Wine Bar	301 Front St W	CA	Toronto	Canada	301 Front St. W	1271	[301 Fro (301 Fro Toronto
3	Goldstone Noodle Restaurant 金石	Noodle House	266 Spadina Ave	CA	Toronto	Canada	at Willison Sq	892	[266 Spa (at Willis Toronto
4	Rol San Restaurant 龍笙棧	Dim Sum Restaurant	323 Spadina Ave.	CA	Toronto	Canada	at D'Arcy St.	922	[323 Spa (at D'Arc Toronto

In [25]:
```
# delete unnecessary columns clean_Restaurant_dataframe2=
clean_Restaurant_dataframe.drop(['cc', 'city', 'cou ntry', 'crossStreet',
'distance', 'formattedAddress',\
                                          'labeledLatLngs', 'neighborhood', 'id'],
axis=1)
clean_Restaurant_dataframe2
```

	name	categories	address	lat	lng	postalCode	state
0	Hemispheres Restaurant & Bistro	American Restaurant	110 Chestnut Street	43.654884	-79.385931	M5G 1R3	ON
1	Azure Restaurant & Bar	Restaurant	225 Front St W	43.644749	-79.385113	M5V 2X3	ON
2	360 Restaurant	Wine Bar	301 Front St W	43.642537	-79.387042	M5V 2T6	ON
3	Goldstone Noodle Restaurant 金石	Noodle House	266 Spadina Ave	43.652278	-79.398039	M5T 2E4	ON
4	Rol San Restaurant 龍笙棧	Dim Sum Restaurant	323 Spadina Ave.	43.654318	-79.398650	M5T 2E9	ON
5	New Sky Restaurant 小沙田食家	Chinese Restaurant	353 Spadina Ave.	43.655337	-79.398897	M5T 2G3	ON
6	Swatow Restaurant 汕頭小食家	Chinese Restaurant	309 Spadina Ave.	43.653866	-79.398334	M5T 2E6	ON
7	North-East Chinese Restaurant 華北美食	Chinese Restaurant	476 Dundas St.	43.653185	-79.396677	NaN	ON
8	Aroma Fine Indian Restaurant	Indian Restaurant	287 King St. W	43.646463	-79.389644	M5V 1J5	ON
9	Sky Dragon Chinese Restaurant 龍翔酒樓	Dim Sum Restaurant	280 Spadina Ave.	43.652783	-79.398174	NaN	ON
10	Victoria's Restaurant	Restaurant	37 King Street East	43.649298	-79.376431	M5C 1E9	ON
11	The Hot House Restaurant & Bar	American Restaurant	35 Church St	43.648824	-79.373702	M5E 1T3	ON
12	Ka Chi Korean Restaurant	Korean Restaurant	8 St Andrew St.	43.654307	-79.399277	M5T 1K6	ON

Out[25]:

13	Some Time BBQ Grill Restaurant 碳烤屋	Szechuan Restaurant	988 Baldwin Street	43.655874	-79.393826	NaN	ON
14	Victor Restaurant & Bar	Bar	30 Mercer Street	43.645634	-79.391125	M5V 1H3	ON
15	Sassafraz | Cafe | Restaurant | Private Events	Event Space	100 Cumberland Street	43.670342	-79.391041	M5R 1A6	ON
16	Green Tea Restaurant Downtown	Chinese Restaurant	261 Spadina Avenue. Upper level	43.652488	-79.397501	M5T 2E3	ON

	name	categories	address	lat	lng	postalCode	state
17	Hong Shing Chinese Restaurant	Chinese Restaurant	195 Dundas St W	43.654925	-79.387089	M5G 1C7	ON
18	Wah Too Seafood Restaurant	Chinese Restaurant	56 Centre Ave.	43.654833	-79.387206	M5G 1R5	ON
19	Tasty's Caribbean Restaurant & Catering	Caribbean Restaurant	405 Spadina Ave	43.656794	-79.399251	M5T 2G6	ON
20	Kensington Cornerstone Restaurant	Breakfast Spot	2A Kensington Ave.	43.652803	-79.399958	M5T 2J7	ON
21	Studio Restaurant	Breakfast Spot	389 Church St.	43.661500	-79.379319	M5B	ON
22	Kyoto House Japanese Restaurant	Sushi Restaurant	143 Dundas St. West	43.655381	-79.385270	NaN	ON
23	The Lakeview Restaurant	Diner	1132 Dundas St. W	43.649435	-79.420390	M6J 1X2	ON
24	Cottage Restaurant & Lounge	Thai Restaurant	338 Jarvis St.	43.662770	-79.376894	M4Y 2G6	ON
25	Richtree Natural Market Restaurants	Restaurant	14 Queen St W	43.652614	-79.380231	M5H 3X4	ON
26	Yueh Tung Chinese Restaurant	Chinese Restaurant	126 Elizabeth St.	43.655281	-79.385337	NaN	ON
27	ONE Restaurant/Lounge	New American Restaurant	116 Yorkville Ave	43.670809	-79.393272	NaN	ON
28	Sightlines Restaurant	American Restaurant	Rogers Centre	43.641635	-79.389365	NaN	ON
29	Micheal's Restaurant and Deli	Bar	566 Queen St W	43.647289	-79.403493	NaN	ON

In [26]:
```
# delete rows with none values df_Restaurant =
clean_Restaurant_dataframe2.dropna(axis=0, how='any', thresh=Non e,
subset=None, inplace=False) df_Restaurant
```

	name	categories	address	lat	lng	postalCode	state
0	Hemispheres Restaurant & Bistro	American Restaurant	110 Chestnut Street	43.654884	-79.385931	M5G 1R3	ON
1	Azure Restaurant & Bar	Restaurant	225 Front St W	43.644749	-79.385113	M5V 2X3	ON
2	360 Restaurant	Wine Bar	301 Front St W	43.642537	-79.387042	M5V 2T6	ON
3	Goldstone Noodle Restaurant 金石	Noodle House	266 Spadina Ave	43.652278	-79.398039	M5T 2E4	ON
4	Rol San Restaurant 龍笙棧	Dim Sum Restaurant	323 Spadina Ave.	43.654318	-79.398650	M5T 2E9	ON
5	New Sky Restaurant 小沙田食家	Chinese Restaurant	353 Spadina Ave.	43.655337	-79.398897	M5T 2G3	ON
6	Swatow Restaurant 汕頭小食家	Chinese Restaurant	309 Spadina Ave.	43.653866	-79.398334	M5T 2E6	ON
8	Aroma Fine Indian Restaurant	Indian Restaurant	287 King St. W	43.646463	-79.389644	M5V 1J5	ON
10	Victoria's Restaurant	Restaurant	37 King Street East	43.649298	-79.376431	M5C 1E9	ON
11	The Hot House Restaurant & Bar	American Restaurant	35 Church St	43.648824	-79.373702	M5E 1T3	ON
12	Ka Chi Korean Restaurant	Korean Restaurant	8 St Andrew St.	43.654307	-79.399277	M5T 1K6	ON
14	Victor Restaurant & Bar	Bar	30 Mercer Street	43.645634	-79.391125	M5V 1H3	ON
15	Sassafraz | Cafe | Restaurant | Private Events	Event Space	100 Cumberland Street	43.670342	-79.391041	M5R 1A6	ON

Out[26]:

16	Green Tea Restaurant Downtown	Chinese Restaurant	261 Spadina Avenue. Upper level	43.652488	-79.397501	M5T 2E3	ON
17	Hong Shing Chinese Restaurant	Chinese Restaurant	195 Dundas St W	43.654925	-79.387089	M5G 1C7	ON
18	Wah Too Seafood Restaurant	Chinese Restaurant	56 Centre Ave.	43.654833	-79.387206	M5G 1R5	ON
19	Tasty's Caribbean Restaurant & Catering	Caribbean Restaurant	405 Spadina Ave	43.656794	-79.399251	M5T 2G6	ON

	name	categories	address	lat	lng	postalCode	state
20	Kensington Cornerstone Restaurant	Breakfast Spot	2A Kensington Ave.	43.652803	-79.399958	M5T 2J7	ON
21	Studio Restaurant	Breakfast Spot	389 Church St.	43.661500	-79.379319	M5B	ON
23	The Lakeview Restaurant	Diner	1132 Dundas St. W	43.649435	-79.420390	M6J 1X2	ON
24	Cottage Restaurant & Lounge	Thai Restaurant	338 Jarvis St.	43.662770	-79.376894	M4Y 2G6	ON
25	Richtree Natural Market Restaurants	Restaurant	14 Queen St W	43.652614	-79.380231	M5H 3X4	ON

Cafeteria whereabouts

In [27]:
```
# search for Cafeteria
search_query = 'Cafeteria'
radius = 10000

# Define the corresponding URL
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}
&ll={},{}&v={}&query={}&radius={}&limit={}'.format(ClIENT_ID, ClIENT_SECRET, lat
itude, longitude, VERSION, search_query, radius, LIMIT) url
```

Out[27]:
```
'https://api.foursquare.com/v2/venues/search?client_id=NGC2LWZF34JVJEF042ZFUKN1TK
N42ESDDJGADQMYE0QXCLVN&client_secret=G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFE
D4WU&ll=43.653963,-79.387207&v=20180604&query=Cafeteria&radius=10000&limit=30'
```

In [46]:
```
# Send the GET Request and examine the results
cresults = requests.get(url).json()
#cresults
```

Out[29]:

In [29]:
```
# assign relevant part of JSON to venues
venues = cresults['response']['venues']

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

	categories	hasPerk	id	location.address	loca
0	[{'id': '4bf58dd8d48988d16e941735', 'name': 'F...	False	4f9165bde4b04fef0087e4e2	The Hospital for Sick Children (SickKids)	CA
1	[{'id': '4bf58dd8d48988d142941735', 'name': 'A...	False	528e99a211d262edc3708a6f	388 Spadina Ave	CA
2	[{'id': '4bf58dd8d48988d128941735', 'name': 'C...	False	53bd6dfe498e6c1d3bc9333e	NaN	CA
3	[{'id': '4bf58dd8d48988d1a1941735', 'name': 'C...	False	4ec2b68a8231a83de8c45191	100 McCaul St.	CA
4	[{'id': '4bf58dd8d48988d1d0941735', 'name': 'D...	False	556f826b498e07f469e6bf1f	NaN	CA

Cleanup on Cafetaria Data frame

Out[30]:

In [30]:
```
# keep only columns that include venue name, and anything that is associated wit
h location
Cafeteria_clean_columns = ['name', 'categories'] + [col for col in Cafeteria_dat
aframe.columns if col.startswith('location.')]+ ['id']
clean_Cafeteria_dataframe = Cafeteria_dataframe.loc[:,Cafeteria_clean_columns]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list4 = row['categories']
    except:
        categories_list4 = row['venue.categories']

    if len(categories_list4) == 0:
        return None
    else:
        return categories_list4[0]['name']

# filter the category for each row
clean_Cafeteria_dataframe['categories'] = clean_Cafeteria_dataframe.apply(get_ca
tegory_type, axis=1)

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

clean_Cafeteria_dataframe.head()
```

	name	categories	address	cc	city	country	crossStreet	distance	formattedAdd
0	Cafeteria	Fast Food Restaurant	The Hospital for Sick Children (SickKids)	CA	Toronto	Canada	NaN	372	[The Hospital f Sick Children (SickKids), To.
1	Stay Cafeteria 慢走	Asian Restaurant	388 Spadina Ave	CA	Toronto	Canada	at Nassau St	977	[388 Spadina A (at Nassau St) Toronto ON M5
2	Industrial Alliance Cafeteria	Cafeteria	NaN	CA	Toronto	Canada	NaN	260	[Toronto ON, Canada]
3	OCAD Cafeteria	College Cafeteria	100 McCaul St.	CA	Toronto	Canada	Second Floor	389	[100 McCaul S (Second Floor) Toronto ON, C
4	Dears Cafeteria	Dessert Shop	NaN	CA	NaN	Canada	NaN	431	[Canada]

In [31]:
```
# delete unnecessary columns clean_Cafeteria_dataframe2=
clean_Cafeteria_dataframe.drop(['cc', 'city', 'count ry', 'crossStreet',
'distance', 'formattedAddress',\
                                          'labeledLatLngs', 'id'], axis=1)
clean_Cafeteria_dataframe2
```

	name	categories	address	lat	lng	postalCode	state
0	Cafeteria	Fast Food Restaurant	The Hospital for Sick Children (SickKids)	43.657209	-79.386063	NaN	ON
1	Stay Cafeteria 慢走	Asian Restaurant	388 Spadina Ave	43.655454	-79.399163	M5T 2G5	ON
2	Industrial Alliance Cafeteria	Cafeteria	NaN	43.655657	-79.389443	NaN	ON
3	OCAD Cafeteria	College Cafeteria	100 McCaul St.	43.652780	-79.391762	NaN	ON
4	Dears Cafeteria	Dessert Shop	NaN	43.654179	-79.392555	NaN	NaN
5	Sears Cafeteria	Food Court	Yonge	43.656038	-79.380672	NaN	ON
6	Chestnut Tree Cafeteria	Cafeteria	89 Chestnut Street	43.648231	-79.384045	NaN	ON
7	Med Sci Cafeteria	College Cafeteria	NaN	43.660260	-79.392977	NaN	ON
8	De La Salle College Cafeteria	College Cafeteria	131 Farnham Avenue	43.683003	-79.397815	M4V 1H7	ON
9	Ryerson Hub Cafeteria	College Cafeteria	350 Victoria St.	43.658475	-79.377653	NaN	ON
10	SickKids Cafeteria	Food Court	555 University Avenue	43.657584	-79.375652	NaN	ON
11	Macdonald Block Cafeteria	Cafeteria	NaN	43.663212	-79.388062	NaN	NaN
12	Robarts Cafeteria	College Cafeteria	130 St. George St.	43.664864	-79.399766	NaN	ON
13	OMP Cafeteria	Cafeteria	1 Mount Pleasant	43.669801	-79.379790	NaN	ON

Out[31]:

14	Rogers Cafeteria	Food Court	NaN	43.669904	-79.379748	NaN	ON
15	Choices Cafeteria	Food Court	200 Bloor Street E	43.671411	-79.381849	NaN	ON
16	George Brown College St. James Campus	Community College	200 King St. East	43.651366	-79.370158	NaN	ON
17	Marketeria	Restaurant	30 Bond St.	43.653585	-79.378843	M5B 1W8	ON
18	The Hub	College Cafeteria	350 Victoria St.	43.658585	-79.380622	M5B 2K3	ON

	name	categories	address	lat	lng	postalCode	state
19	Cafeteria, Lawrence Park Collegiate	High School	NaN	43.721732	-79.409600	NaN	NaN
20	Cafeteria, Lawrence Park Collegiate	Event Space	NaN	43.723286	-79.412222	NaN	NaN
21	Kasala Cafeteria	African Restaurant	Bloor	43.660759	-79.424018	NaN	NaN
22	CC Cafeteria	College Cafeteria	951 Carlaw Ave	43.684631	-79.348862	NaN	ON
23	Innis College	Student Center	2 Sussex Ave	43.665556	-79.399298	M5S 1J5	ON
24	Estonian House Cafeteria	Café	958 Broadview avenue	43.681763	-79.358246	NaN	ON
25	Radio Cafe	Food Court	333 Bloor st east	43.671696	-79.380617	NaN	ON
26	Glendon Cafeteria	College Cafeteria	2275 Bayview Ave	43.727383	-79.378080	NaN	ON
27	Lawrence Park Collegiate Cafeteria	High School	NaN	43.722757	-79.413734	NaN	ON
28	6th Floor Cafeteria 12 Concorde Place	Café	12 Concorde Place	43.721861	-79.329339	m3c 3k7	ON
29	L5 Cafeteria	Cuban Restaurant	NaN	43.723019	-79.345012	NaN	NaN

Out[32]:

In [32]:
```
# delete rows with none values
df_Cafeteria = clean_Cafeteria_dataframe2.dropna(axis=0, how='any', thresh=None,
subset=None, inplace=False)
df_Cafeteria
```

	name	categories	address	lat	lng	postalCode	state
1	Stay Cafeteria 慢走	Asian Restaurant	388 Spadina Ave	43.655454	-79.399163	M5T 2G5	ON
8	De La Salle College Cafeteria	College Cafeteria	131 Farnham Avenue	43.683003	-79.397815	M4V 1H7	ON
17	Marketeria	Restaurant	30 Bond St.	43.653585	-79.378843	M5B 1W8	ON
18	The Hub	College Cafeteria	350 Victoria St.	43.658585	-79.380622	M5B 2K3	ON
23	Innis College	Student Center	2 Sussex Ave	43.665556	-79.399298	M5S 1J5	ON
28	6th Floor Cafeteria 12 Concorde Place	Café	12 Concorde Place	43.721861	-79.329339	m3c 3k7	ON

Shopping Store Search

In [33]:
```
# search for Shopping
search_query = 'Shopping'
radius = 1000

# Define the corresponding URL
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}
&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, lat
itude, longitude, VERSION, search_query, radius, LIMIT) url
```

Out[33]:
```
'https://api.foursquare.com/v2/venues/search?client_id=NGC2LWZF34JVJEF042ZFUKN1TK
N42ESDDJGADQMYE0QXCLVN&client_secret=G4OP1MONUTGD1UK1LRWDUKGMC2LU2F5HNRDWZRW1TMFE
D4WU&ll=43.653963,-79.387207&v=20180604&query=Shopping&radius=1000&limit=30'
```

In [47]:
```
# Send the GET Request and examine the results
sresults = requests.get(url).json()
#sresults
```

In [41]:

```
# assign relevant part of JSON to venues
venues = sresults['response']['venues']

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

Out[35]:

	categories	hasPerk	id	location.address	locat
0	[{'id': '4bf58dd8d48988d1fd941735', 'name': 'S...	False	4ad4c062f964a520fcf720e3	280 Spadina Ave	CA
1	[{'id': '4bf58dd8d48988d1fd941735', 'name': 'S...	False	4b9e86dff964a520e6eb36e3	66 Wellington St W	CA
2	[{'id': '4bf58dd8d48988d1fd941735', 'name': 'S...	False	4ad77a12f964a520260b21e3	220 Yonge St	CA
3	[{'id': '4bf58dd8d48988d1f6941735', 'name': 'D...	False	5840cafe06f1a34af80cc609	176 Yonge Street	CA
4	[{'id': '4bf58dd8d48988d124941735', 'name': 'O...	False	4b83d653f964a520d91231e3	461 King St. West	CA

In [42]:

Cleaning the Shopping Data

In [43]:

```
# keep only columns that include venue name, and anything that is associated wit
h location
Shopping_clean_columns = ['name', 'categories'] + [col for col in Shopping_dataf
rame.columns if col.startswith('location.')]+ ['id']
clean_Shopping_dataframe = Shopping_dataframe.loc[:,Shopping_clean_columns]

# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list5 = row['categories']
    except:
        categories_list5 = row['venue.categories']

    if len(categories_list5) == 0:
        return None
    else:
        return categories_list5[0]['name']

# filter the category for each row
clean_Shopping_dataframe['categories'] = clean_Shopping_dataframe.apply(get_cate
gory_type, axis=1)

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

clean_Shopping_dataframe.head()
```

Out[36]:

	name	categories	address	cc	city	country	crossStreet	distance	forr
0	Dragon City Shopping Mall 龍城	Shopping Mall	280 Spadina Ave	CA	Toronto	Canada	at Dundas St W	897	[280 (at I Toro
1	TD Centre Shopping Concourse	Shopping Mall	66 Wellington St W	CA	Toronto	Canada	btw York & Bay St	908	[66 W (St),
2	CF Toronto Eaton Centre	Shopping Mall	220 Yonge St	CA	Toronto	Canada	btwn Queen & Dundas	529	[220 (btw Dur O...
3	Saks Fifth Avenue Club - Personal Shopping	Department Store	176 Yonge Street	CA	Toronto	Canada	Queen Street West	688	[176 (Qu Wes
4	Yellow Pages - RedFlagDeals.com	Office	461 King St. West	CA	Toronto	Canada	NaN	974	[461 Toro 1K4

In [44]:

```
# delete unnecessary columns
clean_Shopping_dataframe2= clean_Shopping_dataframe.drop(['cc', 'city', 'countr
y', 'crossStreet', 'distance', 'formattedAddress',\
                                          'labeledLatLngs', 'neighborhood', 'id'],
axis=1)
clean_Shopping_dataframe2
```

	name	categories	address	lat	lng	postalCode	state
0	Dragon City Shopping Mall 龍城	Shopping Mall	280 Spadina Ave	43.652774	-79.398222	M5T 3A5	ON
1	TD Centre Shopping Concourse	Shopping Mall	66 Wellington St W	43.647184	-79.380932	M5K 1A1	ON
2	CF Toronto Eaton Centre	Shopping Mall	220 Yonge St	43.654540	-79.380677	M5B 2H1	ON
3	Saks Fifth Avenue Club - Personal Shopping	Department Store	176 Yonge Street	43.651810	-79.379192	M5C 2L7	ON
4	Yellow Pages - RedFlagDeals.com	Office	461 King St. West	43.647958	-79.396006	M5V 1K4	ON

In [38]:

```
# delete rows which its category is not Shopping Mall
df_Shopping = clean_Shopping_dataframe2[clean_Shopping_dataframe2.categories ==
'Shopping Mall']
df_Shopping
```

	name	categories	address	lat	lng	postalCode	state
0	Dragon City Shopping Mall 龍城	Shopping Mall	280 Spadina Ave	43.652774	-79.398222	M5T 3A5	ON
1	TD Centre Shopping Concourse	Shopping Mall	66 Wellington St W	43.647184	-79.380932	M5K 1A1	ON
2	CF Toronto Eaton Centre	Shopping Mall	220 Yonge St	43.654540	-79.380677	M5B 2H1	ON

Out[37]:

In [45]:
Out[38]:

Visualizing Hotels,Shopping store and Cafetaria clusters

Out[39]:

In [39]:
```
# create dataframe of hotels, shopping stores and Cafeteria
hotel_neighbourhood_df = pd.concat([df_hotel, df_Cafeteria, df_Shopping], ignore
_index=True)
hotel_neighbourhood_df.head()
```

	name	categories	address	lat	lng	postalCode	state
0	Sheraton Centre Toronto Hotel	Hotel	123 Queen Street West	43.650594	-79.384530	M5H 2M9	ON
1	Grand Ballroom	Event Space	123 Queen St. W	43.651217	-79.383771	M5H 2M9	ON
2	Stay Cafeteria 慢走	Asian Restaurant	388 Spadina Ave	43.655454	-79.399163	M5T 2G5	ON
3	De La Salle College Cafeteria	College Cafeteria	131 Farnham Avenue	43.683003	-79.397815	M4V 1H7	ON
4	Marketeria	Restaurant	30 Bond St.	43.653585	-79.378843	M5B 1W8	ON

In [40]:
```
# Generate map to visualize hotel neighbourhood including shopping stores and Ca
feteria
hotel_map = folium.Map(location=[latitude, longitude], zoom_start=14)

for lat, lng, name, categories, address in zip(hotel_neighbourhood_df['lat'], ho
tel_neighbourhood_df['lng'],
                                               hotel_neighbourhood_df['name'], hotel
_neighbourhood_df['categories'],\
                                                   hotel_neighbourhood_df['address'
]):
    label = '{}, {}'.format(name, address)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7,
        parse_html=False).add_to(hotel_map)

hotel_map
```

Out[40]:

Visualization of Park Restaurant and Cafetaria together

In [41]:
```
# create dataframe of Park, Restaurant and Cafeteria park_neighbourhood_df =
pd.concat([df_park, df_Restaurant, df_Cafeteria,], ignor e_index=True)
park_neighbourhood_df
```

	name	categories	address	lat	lng	postalCode	state
0	Queen's Park	Park	University Ave.	43.663946	-79.392180	M5R 2E8	ON
1	Bellevue Square Park	Park	btwn Bellevue & Augusta Ave.	43.653610	-79.402199	M5T 2N4	ON
2	Canoe Landing Park	Park	50 Fort York Blvd	43.638762	-79.397067	M5V 3Z1	ON
3	Trinity Bellwoods Park	Park	1053 Dundas St. W.	43.647072	-79.413756	M5H 2N2	ON
4	Riverdale Park West	Park	500 Gerrard St.	43.666048	-79.360941	M5A 2H3	ON
5	Barbara Hall Park	Park	519 Church St	43.666879	-79.381068	M4Y 2K9	ON
6	High Park	Park	1873 Bloor St. W	43.646479	-79.463425	M6R 2Z3	ON
7	Riverdale Park East	Park	550 Broadview Ave	43.669951	-79.355493	M4K 2P1	ON
8	Hemispheres Restaurant & Bistro	American Restaurant	110 Chestnut Street	43.654884	-79.385931	M5G 1R3	ON
9	Azure Restaurant & Bar	Restaurant	225 Front St W	43.644749	-79.385113	M5V 2X3	ON
10	360 Restaurant	Wine Bar	301 Front St W	43.642537	-79.387042	M5V 2T6	ON
11	Goldstone Noodle Restaurant 金石	Noodle House	266 Spadina Ave	43.652278	-79.398039	M5T 2E4	ON
12	Rol San Restaurant 龍笙棧	Dim Sum Restaurant	323 Spadina Ave.	43.654318	-79.398650	M5T 2E9	ON
13	New Sky Restaurant 小沙田食家	Chinese Restaurant	353 Spadina Ave.	43.655337	-79.398897	M5T 2G3	ON
14	Swatow Restaurant 汕頭小食家	Chinese Restaurant	309 Spadina Ave.	43.653866	-79.398334	M5T 2E6	ON

Out[41]:

15	Aroma Fine Indian Restaurant	Indian Restaurant	287 King St. W	43.646463	-79.389644	M5V 1J5	ON
16	Victoria's Restaurant	Restaurant	37 King Street East	43.649298	-79.376431	M5C 1E9	ON
17	The Hot House Restaurant & Bar	American Restaurant	35 Church St	43.648824	-79.373702	M5E 1T3	ON
18	Ka Chi Korean Restaurant	Korean Restaurant	8 St Andrew St.	43.654307	-79.399277	M5T 1K6	ON

	name	categories	address	lat	lng	postalCode	state
19	Victor Restaurant & Bar	Bar	30 Mercer Street	43.645634	-79.391125	M5V 1H3	ON
20	Sassafraz | Cafe | Restaurant | Private Events	Event Space	100 Cumberland Street	43.670342	-79.391041	M5R 1A6	ON
21	Green Tea Restaurant Downtown	Chinese Restaurant	261 Spadina Avenue. Upper level	43.652488	-79.397501	M5T 2E3	ON
22	Hong Shing Chinese Restaurant	Chinese Restaurant	195 Dundas St W	43.654925	-79.387089	M5G 1C7	ON
23	Wah Too Seafood Restaurant	Chinese Restaurant	56 Centre Ave.	43.654833	-79.387206	M5G 1R5	ON
24	Tasty's Caribbean Restaurant & Catering	Caribbean Restaurant	405 Spadina Ave	43.656794	-79.399251	M5T 2G6	ON
25	Kensington Cornerstone Restaurant	Breakfast Spot	2A Kensington Ave.	43.652803	-79.399958	M5T 2J7	ON
26	Studio Restaurant	Breakfast Spot	389 Church St.	43.661500	-79.379319	M5B	ON
27	The Lakeview Restaurant	Diner	1132 Dundas St. W	43.649435	-79.420390	M6J 1X2	ON
28	Cottage Restaurant & Lounge	Thai Restaurant	338 Jarvis St.	43.662770	-79.376894	M4Y 2G6	ON
29	Richtree Natural Market Restaurants	Restaurant	14 Queen St W	43.652614	-79.380231	M5H 3X4	ON
30	Stay Cafeteria 慢走	Asian Restaurant	388 Spadina Ave	43.655454	-79.399163	M5T 2G5	ON
31	De La Salle College Cafeteria	College Cafeteria	131 Farnham Avenue	43.683003	-79.397815	M4V 1H7	ON
32	Marketeria	Restaurant	30 Bond St.	43.653585	-79.378843	M5B 1W8	ON
33	The Hub	College Cafeteria	350 Victoria St.	43.658585	-79.380622	M5B 2K3	ON
34	Innis College	Student Center	2 Sussex Ave	43.665556	-79.399298	M5S 1J5	ON
35	6th Floor Cafeteria 12 Concorde	Café	12 Concorde	43.721861	-79.329339	m3c 3k7	ON

In [42]:
```
# Generate map to visualize park neighbourhood including Restaurant and Cafeteri
a
park_map = folium.Map(location=[latitude, longitude], zoom_start=14)

for lat, lng, name, categories, address in zip(park_neighbourhood_df['lat'], par
k_neighbourhood_df['lng'],
                                               park_neighbourhood_df['name'], park_n
eighbourhood_df['categories'],\
                                                   park_neighbourhood_df['address'
]):
    label = '{}, {}'.format(name, address)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7,
        parse_html=False).add_to(park_map)

park_map
```

Out[42]:

Final Observation

1. There is a definite opportunity for the delegates to be housed in and around Queens
2. Sheraton Hotel seems like a good choice considering there is a event place available to host the meeting
3. The second map provides for the families of the visiting delegates who would be interested on recreational pursuits
4. The area in and around Queens have ample parks,restaurant and cafetaria
5. Therefore the delegates need to be housed in the Queens locality and preferably at Sheraton Disclaimer:

The Above observation/conclusion is bereft of other factors like cost and ratings and could be a future study artifact.