ExploringCompetitors

November 22, 2024

Exploring the Competitors

0.0.1 Part 1. Get location data using Foursquare

Foursquare Places API is very usefule online application used by many developers & other application like Uber etc. In this project you can used it to retrieve informtion about the places present in the neighborhoods of Toronto. The API returns a JSON file and we need to turn that into a data-frame. Here I've chosen similar businesses(pet grooming) for each neighborhood within a radius of 2.5km.

You will need to create an account with Foursquare to access the API. It is free to sign up, and you get $200 free credit.

[6]:
```
import requests
import pandas as pd
```

[7]:
```
sorted_neighborhood = pd.read_csv('raw_data/sorted_neighborhood.csv')
```

[8]:
```
display(sorted_neighborhood)
```

```
     Unnamed: 0.1 postal_code                                    data_df1  \
0              67         M5V  2021A0011M5V - Children - Counts - Total
1               3         M2N  2021A0011M2N - Children - Counts - Total
2              32         M6S  2021A0011M6S - Children - Counts - Total
3              35         M6P  2021A0011M6P - Children - Counts - Total
4              38         M8V  2021A0011M8V - Children - Counts - Total
..            …          …                                          …
98             98         M5L                                        NaN
99             99         M5K                                        NaN
100           100         M7R                                        NaN
101           101         M7Y                                        NaN
102           102         M7A                                        NaN

     Children in each neighbor  Children Points  \
0                       5705.0             29.0
1                      17785.0             93.0
2                      10095.0             64.0
3                       9180.0             61.0
4                       8915.0             58.0
```

```
..                          …              …
98                          NaN              NaN
99                          NaN              NaN
100                         NaN              NaN
101                         NaN              NaN
102                         NaN              NaN

                                                 data_df2  Single parents  \
0    2021A0011M5V - Total one-parent families - Cou…          1340.0
1    2021A0011M2N - Total one-parent families - Cou…          3355.0
2    2021A0011M6S - Total one-parent families - Cou…          1500.0
3    2021A0011M6P - Total one-parent families - Cou…          1660.0
4    2021A0011M8V - Total one-parent families - Cou…          2120.0
..                                                    …              …
98                                                   NaN              NaN
99                                                   NaN              NaN
100                                                  NaN              NaN
101                                                  NaN              NaN
102                                                  NaN              NaN

     Single parents Points  Unnamed: 0  \
0                     45.0        66.0
1                     90.0        22.0
2                     50.0        80.0
3                     56.0        78.0
4                     73.0        81.0
..                     …          …
98                    NaN          NaN
99                    NaN          NaN
100                   NaN          NaN
101                   NaN          NaN
102                   NaN          NaN

                                                    data  …  Unnamed: 0_df4  \
0    2021A0011M5V - Median after-tax income in 2020…  …              66.0
1    2021A0011M2N - Median after-tax income in 2020…  …              22.0
2    2021A0011M6S - Median after-tax income in 2020…  …              80.0
3    2021A0011M6P - Median after-tax income in 2020…  …              78.0
4    2021A0011M8V - Median after-tax income in 2020…  …              81.0
..                                                    …  …               …
98                                                   NaN  …               NaN
99                                                   NaN  …               NaN
100                                                  NaN  …               NaN
101                                                  NaN  …               NaN
102                                                  NaN  …               NaN

                                               data_df4  \
0    2021A0011M5V - Bachelor's degree or higher - C…
```

```
1    2021A0011M2N - Bachelor's degree or higher - C…
2    2021A0011M6S - Bachelor's degree or higher - C…
3    2021A0011M6P - Bachelor's degree or higher - C…
4    2021A0011M8V - Bachelor's degree or higher - C…
..                                                 …
98                                                 NaN
99                                                 NaN
100                                                NaN
101                                                NaN
102                                                NaN

     Number of bachlors degreess Education Score  Unnamed: 0_df5  \
0                        39245.0            95.0              68
1                        39350.0            96.0              22
2                        15230.0            85.0              84
3                        18145.0            91.0              82
4                        17070.0            89.0              88
..                           …               …               …
98                           NaN             NaN              61
99                           NaN             NaN              60
100                          NaN             NaN              86
101                          NaN             NaN              87
102                          NaN             NaN              85

                                      data_df5    value Score  \
0    2021A0011M5V - Non-movers - Counts - Total  37820.0    84
1    2021A0011M2N - Non-movers - Counts - Total  59225.0    95
2    2021A0011M6S - Non-movers - Counts - Total  30465.0    62
3    2021A0011M6P - Non-movers - Counts - Total  33125.0    74
4    2021A0011M8V - Non-movers - Counts - Total  36260.0    78
..                                          …      …   …
98   2021A0011M5J - Non-movers - Counts - Total  11065.0    14
99   2021A0011M5J - Non-movers - Counts - Total  11065.0    14
100  2021A0011M7A - Non-movers - Counts - Total      0.0     1
101  2021A0011M7A - Non-movers - Counts - Total      0.0     1
102  2021A0011M7A - Non-movers - Counts - Total      0.0     1

     mobility_score  final score
0                84        80.45
1                95        78.00
2                62        77.10
3                74        76.05
4                78        73.70
..              …          …
98               14          NaN
99               14          NaN
100               1          NaN
101               1          NaN
```

```
102              1          NaN

[103 rows x 22 columns]
```

[9]:
```
toronto_DF = pd.read_csv('raw_data/Toronto_Neighborhoods_with_Coordinates.csv')
```

[10]:
```
display(toronto_DF)
```

```
    Postalcode             Borough  \
0          M1B          Scarborough
1          M1C          Scarborough
2          M1E          Scarborough
3          M1G          Scarborough
4          M1H          Scarborough
..         …                    …
98         M9N                 York
99         M9P            Etobicoke
100        M9R            Etobicoke
101        M9V            Etobicoke
102        M9W  Etobicoke Northwest

                                      Neighborhood   Latitude  Longitude
0                                   Malvern / Rouge  43.806686 -79.194353
1            Rouge Hill / Port Union / Highland Creek  43.784535 -79.160497
2                 Guildwood / Morningside / West Hill  43.763573 -79.188711
3                                           Woburn  43.770992 -79.216917
4                                        Cedarbrae  43.773136 -79.239476
..                                             …        …         …
98                                          Weston  43.706876 -79.518188
99                                       Westmount  43.696319 -79.532242
100  Kingsview Village / St. Phillips / Martin Grov…  43.688905 -79.554724
101  South Steeles / Silverstone / Humbergate / Jam…  43.739416 -79.588437
102  Clairville / Humberwood / Woodbine Downs / Wes…  43.706748 -79.594054

[103 rows x 5 columns]
```

[11]:
```
# Extracting the top 5 neighborhoods based on a ranked csv of suitability
top_5 = sorted_neighborhood.head(5)
display(top_5)
```

```
   Unnamed: 0.1 postal_code                                  data_df1  \
0            67         M5V  2021A0011M5V - Children - Counts - Total
1             3         M2N  2021A0011M2N - Children - Counts - Total
2            32         M6S  2021A0011M6S - Children - Counts - Total
3            35         M6P  2021A0011M6P - Children - Counts - Total
4            38         M8V  2021A0011M8V - Children - Counts - Total

   Children in each neighbor  Children Points  \
```

```
0                    5705.0             29.0
1                   17785.0             93.0
2                   10095.0             64.0
3                    9180.0             61.0
4                    8915.0             58.0

                                          data_df2  Single parents  \
0  2021A0011M5V - Total one-parent families - Cou…          1340.0
1  2021A0011M2N - Total one-parent families - Cou…          3355.0
2  2021A0011M6S - Total one-parent families - Cou…          1500.0
3  2021A0011M6P - Total one-parent families - Cou…          1660.0
4  2021A0011M8V - Total one-parent families - Cou…          2120.0

   Single parents Points  Unnamed: 0  \
0                   45.0        66.0
1                   90.0        22.0
2                   50.0        80.0
3                   56.0        78.0
4                   73.0        81.0

                                             data  …  Unnamed: 0_df4  \
0  2021A0011M5V - Median after-tax income in 2020…  …            66.0
1  2021A0011M2N - Median after-tax income in 2020…  …            22.0
2  2021A0011M6S - Median after-tax income in 2020…  …            80.0
3  2021A0011M6P - Median after-tax income in 2020…  …            78.0
4  2021A0011M8V - Median after-tax income in 2020…  …            81.0

                                          data_df4  \
0  2021A0011M5V - Bachelor's degree or higher - C…
1  2021A0011M2N - Bachelor's degree or higher - C…
2  2021A0011M6S - Bachelor's degree or higher - C…
3  2021A0011M6P - Bachelor's degree or higher - C…
4  2021A0011M8V - Bachelor's degree or higher - C…

   Number of bachlors degreess Education Score  Unnamed: 0_df5  \
0                       39245.0            95.0              68
1                       39350.0            96.0              22
2                       15230.0            85.0              84
3                       18145.0            91.0              82
4                       17070.0            89.0              88

                                  data_df5    value Score  mobility_score  \
0  2021A0011M5V - Non-movers - Counts - Total  37820.0    84              84
1  2021A0011M2N - Non-movers - Counts - Total  59225.0    95              95
2  2021A0011M6S - Non-movers - Counts - Total  30465.0    62              62
3  2021A0011M6P - Non-movers - Counts - Total  33125.0    74              74
4  2021A0011M8V - Non-movers - Counts - Total  36260.0    78              78
```

```
   final score
0        80.45
1        78.00
2        77.10
3        76.05
4        73.70

[5 rows x 22 columns]
```

[12]:
```
#get their postal code
top_5_postalcodes = top_5['postal_code'].tolist()
```

[13]:
```
#Get the details of the top 5
top5WithDetails = toronto_DF[toronto_DF['Postalcode'].isin(top_5_postalcodes)]
display(top5WithDetails)
```

```
   Postalcode           Borough  \
22        M2N         North York
68        M5V  Downtown Toronto
82        M6P       West Toronto
84        M6S       West Toronto
88        M8V          Etobicoke

                                      Neighborhood   Latitude  Longitude
22                                Willowdale) South  43.770120 -79.408493
68  CN Tower / King and Spadina / Railway Lands / …  43.628947 -79.394420
82                    High Park / The Junction South  43.661608 -79.464763
84                                 Runnymede / Swansea  43.651571 -79.484450
88     New Toronto / Mimico South / Humber Bay Shores  43.605647 -79.501321
```

[14]:
```
# Set up your API key and other constants for Foursquare API
API_KEY = 'fsq31uf/SIS9hDzAI9L8axZ1ppvrJWfckk/wT4qZow2jD/s='  # Replace with␣
 ↪your actual Foursquare API key
LIMIT = 20  # Maximum number of venues to return
radius = 1500  # Search radius in meters

# a function to loop through each neighbourhood in the csv file, and search for␣
 ↪places
def getNearbyVenues(names, postalcode, latitudes, longitudes, radius):
    venues_list = []

    for name, postalcode, lat, lng in zip(names, postalcode, latitudes,␣
 ↪longitudes):
        print(f"Processing neighborhood: {name}")

        # Skip if latitude or longitude is missing
        if pd.isnull(lat) or pd.isnull(lng):
```

```
            print(f"Skipping {name} due to missing latitude or longitude.")
            continue

        # Create the API request URL and parameters
        # Change the params here to customize your search, https://docs.
↪foursquare.com/developer/reference/place-search
        # The following example looks for pet groomers in the neighbourhood
        # I am searching with categories ID 11134 - pet grooming services
        url = 'https://api.foursquare.com/v3/places/search'
        params = {
            'll': f'{lat},{lng}',
            'radius': radius,
            'limit': LIMIT,
            'categories': 11026
        }

        # Set up headers with the API key
        headers = {
            "Accept": "application/json",
            "Authorization": API_KEY
        }

        # Make the GET request
        response = requests.get(url, headers=headers, params=params)

        # Check if the request was successful
        if response.status_code != 200:
            print(f"Failed to get data for {name}. Status code: {response.
↪status_code}")
            print(f"Error message: {response.text}")
            continue

        results = response.json()

        #if this neighborhood doesn't have any daycare venue
        if not results.get('results', []):
            print(f"No venues found for {name}")
            continue

        # Parse the JSON response
        for venue in results.get('results', []):
            # Extract venue information
            venue_name = venue.get('name')
            venue_id = venue.get('fsq_id')
            venue_location = venue.get('geocodes', {}).get('main', {})
            venue_lat = venue_location.get('latitude')
            venue_lng = venue_location.get('longitude')
```

```
            venue_categories = venue.get('categories', [])
            venue_category = venue_categories[0]['name'] if venue_categories␣
 ↪else 'Unknown'

            # Append to the list
            venues_list.append([
                name,
                postalcode,
                lat,
                lng,
                venue_name,
                venue_id,
                venue_lat,
                venue_lng,
                venue_category
            ])

    # Create a DataFrame from the list
    nearby_venues = pd.DataFrame(venues_list, columns=[
        'Neighborhood',
        'Postalcode',
        'Neighborhood Latitude',
        'Neighborhood Longitude',
        'Venue',
        'fsq_id',
        'Venue Latitude',
        'Venue Longitude',
        'Venue Category'
    ])

    return nearby_venues

# Example usage with your DataFrame 'toronto_DF'
# Ensure 'toronto_DF' has the columns 'Neighborhood', 'Latitude', 'Longitude'
toronto_venues = getNearbyVenues(
    names=top5WithDetails['Neighborhood'],
    postalcode=top5WithDetails['Postalcode'],
    latitudes=top5WithDetails['Latitude'],
    longitudes=top5WithDetails['Longitude'],
    radius=radius
)

# Display the first few rows of the resulting DataFrame
toronto_venues.head()
```

```
Processing neighborhood: Willowdale) South
Processing neighborhood: CN Tower / King and Spadina / Railway Lands /
```

```
Harbourfront West / Bathurst Quay / South Niagara / Island airport
Processing neighborhood: High Park / The Junction South
Processing neighborhood: Runnymede / Swansea
Processing neighborhood: New Toronto / Mimico South / Humber Bay Shores
```

[14]:

	Neighborhood	Postalcode	Neighborhood Latitude	\
0	Willowdale) South	M2N	43.77012	
1	Willowdale) South	M2N	43.77012	
2	Willowdale) South	M2N	43.77012	
3	Willowdale) South	M2N	43.77012	
4	Willowdale) South	M2N	43.77012	

	Neighborhood Longitude	Venue	\
0	-79.408493	Yonge Hearts Child Care Centre	
1	-79.408493	Yorktown Montessori School	
2	-79.408493	Kids and Co	
3	-79.408493	St Cyril's B & a Club	
4	-79.408493	Markham Castlemore YMCA Before and After Schoo…	

	fsq_id	Venue Latitude	Venue Longitude	Venue Category
0	4d0a8b5dd823b1f7176ae45c	43.770222	-79.414411	Daycare
1	598c9974dd84420428480d92	43.775815	-79.410149	Daycare
2	9ea3d67fca194aecc4c60ddc	43.761842	-79.410632	Daycare
3	4c8783c5d92ea093df787672	43.776933	-79.416423	Daycare
4	5bc6b5a6e4c459002ce5258c	43.780317	-79.411620	Daycare

[15]:
```
# The total number of pet grooming services in Toronto
toronto_venues['Venue'].nunique()
```

[15]:
```
52
```

[17]:
```
# Oberserving the number of competitors in each neighbourhood
toronto_venues.groupby('Neighborhood')['Venue'].count()
```

[17]:
```
Neighborhood
CN Tower / King and Spadina / Railway Lands / Harbourfront West / Bathurst Quay
/ South Niagara / Island airport     1
High Park / The Junction South
20
New Toronto / Mimico South / Humber Bay Shores
10
Runnymede / Swansea
15
Willowdale) South
11
Name: Venue, dtype: int64
```

0.0.2 Part 2. Interactive leaflet map using coordinate data.

[20]:
```
import folium # map rendering library
```

[24]:
```
display(toronto_venues)
```

	Neighborhood	Postalcode
0	Willowdale) South	M2N
1	Willowdale) South	M2N
2	Willowdale) South	M2N
3	Willowdale) South	M2N
4	Willowdale) South	M2N
5	Willowdale) South	M2N
6	Willowdale) South	M2N
7	Willowdale) South	M2N
8	Willowdale) South	M2N
9	Willowdale) South	M2N
10	Willowdale) South	M2N
11	CN Tower / King and Spadina / Railway Lands / …	M5V
12	High Park / The Junction South	M6P
13	High Park / The Junction South	M6P
14	High Park / The Junction South	M6P
15	High Park / The Junction South	M6P
16	High Park / The Junction South	M6P
17	High Park / The Junction South	M6P
18	High Park / The Junction South	M6P
19	High Park / The Junction South	M6P
20	High Park / The Junction South	M6P
21	High Park / The Junction South	M6P
22	High Park / The Junction South	M6P
23	High Park / The Junction South	M6P
24	High Park / The Junction South	M6P
25	High Park / The Junction South	M6P
26	High Park / The Junction South	M6P
27	High Park / The Junction South	M6P
28	High Park / The Junction South	M6P
29	High Park / The Junction South	M6P
30	High Park / The Junction South	M6P
31	High Park / The Junction South	M6P
32	Runnymede / Swansea	M6S
33	Runnymede / Swansea	M6S
34	Runnymede / Swansea	M6S
35	Runnymede / Swansea	M6S
36	Runnymede / Swansea	M6S
37	Runnymede / Swansea	M6S
38	Runnymede / Swansea	M6S
39	Runnymede / Swansea	M6S
40	Runnymede / Swansea	M6S

41	Runnymede / Swansea	M6S
42	Runnymede / Swansea	M6S
43	Runnymede / Swansea	M6S
44	Runnymede / Swansea	M6S
45	Runnymede / Swansea	M6S
46	Runnymede / Swansea	M6S
47	New Toronto / Mimico South / Humber Bay Shores	M8V
48	New Toronto / Mimico South / Humber Bay Shores	M8V
49	New Toronto / Mimico South / Humber Bay Shores	M8V
50	New Toronto / Mimico South / Humber Bay Shores	M8V
51	New Toronto / Mimico South / Humber Bay Shores	M8V
52	New Toronto / Mimico South / Humber Bay Shores	M8V
53	New Toronto / Mimico South / Humber Bay Shores	M8V
54	New Toronto / Mimico South / Humber Bay Shores	M8V
55	New Toronto / Mimico South / Humber Bay Shores	M8V
56	New Toronto / Mimico South / Humber Bay Shores	M8V

	Neighborhood Latitude	Neighborhood Longitude	\
0	43.770120	-79.408493	
1	43.770120	-79.408493	
2	43.770120	-79.408493	
3	43.770120	-79.408493	
4	43.770120	-79.408493	
5	43.770120	-79.408493	
6	43.770120	-79.408493	
7	43.770120	-79.408493	
8	43.770120	-79.408493	
9	43.770120	-79.408493	
10	43.770120	-79.408493	
11	43.628947	-79.394420	
12	43.661608	-79.464763	
13	43.661608	-79.464763	
14	43.661608	-79.464763	
15	43.661608	-79.464763	
16	43.661608	-79.464763	
17	43.661608	-79.464763	
18	43.661608	-79.464763	
19	43.661608	-79.464763	
20	43.661608	-79.464763	
21	43.661608	-79.464763	
22	43.661608	-79.464763	
23	43.661608	-79.464763	
24	43.661608	-79.464763	
25	43.661608	-79.464763	
26	43.661608	-79.464763	
27	43.661608	-79.464763	
28	43.661608	-79.464763	
29	43.661608	-79.464763	

```
30              43.661608              -79.464763
31              43.661608              -79.464763
32              43.651571              -79.484450
33              43.651571              -79.484450
34              43.651571              -79.484450
35              43.651571              -79.484450
36              43.651571              -79.484450
37              43.651571              -79.484450
38              43.651571              -79.484450
39              43.651571              -79.484450
40              43.651571              -79.484450
41              43.651571              -79.484450
42              43.651571              -79.484450
43              43.651571              -79.484450
44              43.651571              -79.484450
45              43.651571              -79.484450
46              43.651571              -79.484450
47              43.605647              -79.501321
48              43.605647              -79.501321
49              43.605647              -79.501321
50              43.605647              -79.501321
51              43.605647              -79.501321
52              43.605647              -79.501321
53              43.605647              -79.501321
54              43.605647              -79.501321
55              43.605647              -79.501321
56              43.605647              -79.501321

                                                 Venue  \
0                     Yonge Hearts Child Care Centre
1                         Yorktown Montessori School
2                                        Kids and Co
3                               St Cyril's B & a Club
4   Markham Castlemore YMCA Before and After Schoo…
5                                         Kids & Co
6   Hollywood All-stars Child Care Program at Holl…
7       North York St. Gabriel YMCA Child Care Centre
8        Churchill Chums School Age Child Care Centre
9                                       McKee McKids
10                       Sunflower Garden Playschool
11                    Harbourfront Child Care Centre
12                       Sunnyside Day Care on Keele
13                             Novus Day Nursery Ltd
14                                Teddy Bear Academy
15                  Early Enrichment Day Care Centre
16                  Holland Bloorview Nursery School
17  Toronto Indian Road YMCA Before and After Scho…
18           Keelmount Day Care Centre of Toronto Inc
```

19	Child Development Institute
20	Parkdale High Park Oeyc
21	High Park Child Care
22	Humberside Day Care
23	Bloor West Village Daycare
24	Junction Day Care Centre
25	Atelier Kids
26	High Park Early Learning Centre
27	One Step at a Time Childcare Centre
28	Children's Creative Corner Co-Op Nursery School
29	Perth Early Learning Centre
30	Cpe les Amis du Monde
31	Time For Play Co-Operative Nursery School
32	Drake Medox Health Svc
33	Little Learners Childcare Inc
34	Precious People Place Child Care Centre Inc
35	Little Learners Academy
36	The Runnymede Adventure Club
37	Time For Play Co-Operative Nursery School
38	A Place to Play Nursery School of Bloor
39	Children's Creative Corner Co-Op Nursery School
40	Swansea School-Age Daycare
41	Toronto Humbercrest YMCA Before and After Scho…
42	Bloor West Village Daycare
43	Humbercrest Nursery School Inc
44	Junction Day Care Centre
45	One Step at a Time Childcare Centre
46	Red Robin Nature Preschool
47	Second Street YMCA Child Care
48	Toronto Lamp YMCA Child Care Centre
49	Lakeshore Community Childcare Centre
50	Etobicoke John English YMCA Before and After S…
51	Etobicoke St. Teresa YMCA Centre
52	Etobicoke St. Leo YMCA Before and After School…
53	Lamp Child Care
54	The Rec Room
55	Mimico Day Care Centre
56	Martin Luther Church Nursery

	fsq_id	Venue Latitude	Venue Longitude	Venue Category
0	4d0a8b5dd823b1f7176ae45c	43.770222	-79.414411	Daycare
1	598c9974dd84420428480d92	43.775815	-79.410149	Daycare
2	9ea3d67fca194aecc4c60ddc	43.761842	-79.410632	Daycare
3	4c8783c5d92ea093df787672	43.776933	-79.416423	Daycare
4	5bc6b5a6e4c459002ce5258c	43.780317	-79.411620	Daycare
5	4f73896fe4b06e8c1e429ad3	43.780230	-79.415919	Daycare
6	57f25cbab51643114d95b79b	43.770433	-79.391383	Daycare
7	5bc7b0de2aff31002cf7aa86	43.769283	-79.390965	Daycare

8	21c839150bb241f37f0d1479	43.770769	-79.426967	Daycare
9	86c73de6d9784dbc0c767c56	43.774179	-79.411013	Daycare
10	0af7363c4a5444ddac6ab18f	43.776681	-79.409069	Daycare
11	98e7ba928c3940f6e973ca0f	43.635664	-79.398808	Daycare
12	614d2aebce491d1a39837751	43.660580	-79.462777	Daycare
13	ba5ec6cfa7f247ea025951fc	43.660586	-79.462685	Daycare
14	4f27e30de4b06003784d821b	43.659340	-79.467441	Daycare
15	4d272776c406721e7e1965b6	43.660531	-79.468802	Daycare
16	a9d313b9896a437b9d10a9c6	43.662434	-79.460457	Daycare
17	5bc8212cb23dfa002c766d81	43.662624	-79.460160	Daycare
18	ac36ee2721e94953f48fc344	43.657982	-79.462219	Daycare
19	5086f668e4b030ac960e1ac5	43.665559	-79.466852	Daycare
20	55fa1faf498ebd9ea884ee3f	43.665559	-79.466852	Daycare
21	5bc6b78d270ee7002c772fad	43.664533	-79.459391	Daycare
22	524f385111d2cc0d47a63606	43.662808	-79.472208	Daycare
23	52935808498e140df060efa6	43.658102	-79.471309	Daycare
24	e1facac08b1a4ddbc6c4326f	43.661015	-79.473696	Daycare
25	5a2069c66a59501068949b80	43.654886	-79.462328	Daycare
26	4c60704a1e5cd13a82aba1ed	43.654121	-79.465104	Daycare
27	4c6dfc8ea437224bc3e42eb1	43.653712	-79.466862	Daycare
28	ba88c5d7669b4a626ccf359e	43.657068	-79.474696	Daycare
29	99f4478da79f4c461c64a8a3	43.662572	-79.452435	Daycare
30	c6c83b6f974c45feda7bfa24	43.667626	-79.455493	Daycare
31	2a59588b0f404458129e1374	43.655922	-79.478065	Daycare
32	740b3390992a4d926319bd4d	43.650490	-79.480181	Daycare
33	4c77bcc1923ba14301e467e6	43.654879	-79.486921	Daycare
34	2be6fdfaaa5144d2e15d4948	43.655508	-79.483703	Daycare
35	0deefc9e3f294dcf22819496	43.655694	-79.487287	Daycare
36	a6a715add14b44e247f18c81	43.655725	-79.478454	Daycare
37	2a59588b0f404458129e1374	43.655922	-79.478065	Daycare
38	83cbdf1026094d46758019fa	43.659241	-79.484866	Daycare
39	ba88c5d7669b4a626ccf359e	43.657068	-79.474696	Daycare
40	603efa427b174eff1cce984a	43.643413	-79.477170	Daycare
41	5bc8d69ae179100039a7b9c4	43.660419	-79.490996	Daycare
42	52935808498e140df060efa6	43.658102	-79.471309	Daycare
43	9d70b443e48c442cd85b51c4	43.663238	-79.487008	Daycare
44	e1facac08b1a4ddbc6c4326f	43.661015	-79.473696	Daycare
45	4c6dfc8ea437224bc3e42eb1	43.653712	-79.466862	Daycare
46	cc1dbcba052b4d0bbe62f3f2	43.651619	-79.475501	Daycare
47	5bc6f80e2be425002c24e965	43.604003	-79.500244	Daycare
48	5e5fc12139aba100084f83f6	43.603602	-79.503402	Daycare
49	49f597b243e541230317d894	43.599147	-79.504456	Daycare
50	5e5fc0542cdec700096c9429	43.611950	-79.494429	Daycare
51	5bfcb338dd8442002c32ab29	43.599376	-79.508976	Daycare
52	5dc011796d311c0008456dfa	43.614222	-79.496337	Daycare
53	c321cfbf8cb34f87f0b14a83	43.603581	-79.503679	Daycare
54	f289dc2d64e9465dab6e2336	43.612962	-79.496694	Daycare
55	14e04dff6e81402a0a2b9084	43.613518	-79.491412	Daycare

```
56  0c42877f075a4f772c4e9fd9       43.615042       -79.488248        Daycare
```

[33]:
```
# My example below shows day care in Lawrence Park
target = 'M5V'

search_area = toronto_venues[toronto_venues['Postalcode'] == target]
latitude = toronto_DF[toronto_DF['Postalcode'] == target]['Latitude']
longitude = toronto_DF[toronto_DF['Postalcode'] == target]['Longitude']
display(search_area)
```

```
                                        Neighborhood Postalcode  \
11  CN Tower / King and Spadina / Railway Lands / …        M5V

    Neighborhood Latitude  Neighborhood Longitude  \
11              43.628947               -79.39442

                              Venue                     fsq_id  Venue Latitude  \
11  Harbourfront Child Care Centre  98e7ba928c3940f6e973ca0f       43.635664

    Venue Longitude Venue Category
11       -79.398808        Daycare
```

[30]:
```
print(latitude, type(latitude))
print(longitude, type(longitude))
```

```
68    43.628947
Name: Latitude, dtype: float64 <class 'pandas.core.series.Series'>
68   -79.39442
Name: Longitude, dtype: float64 <class 'pandas.core.series.Series'>
```

[36]:
```
latitude = float(latitude)
longitude = float(longitude)

map_toronto = folium.Map(location=[latitude, longitude], zoom_start=14)

# add markers to map
for lat, lng, venue, neighborhood in zip(search_area['Venue Latitude'],␣
 ↪search_area['Venue Longitude'], search_area['Venue'],␣
 ↪search_area['Neighborhood']):
    label = '{},{}'.format(venue, neighborhood)
    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_toronto)

map_toronto
```

[36]: <folium.folium.Map at 0x7fe3d06eb510>

0.0.3 Part 3. Customer Reviews

It seems like there is one one pet groomer withing 2km of Scarborough Village, I am interested in seeing customer's comments on Funny Bunny. I can use Foursquare's Place Tips API.

[40]:
```
# a function to loop through the list of daycare in the neighborhood and␣
 ↪compile all the comments related to them
def getTips(venues, fsq_ids):
    venues_tips = []

    for venue, fsq_id in zip(venues, fsq_ids):
        print(f"Processing venue: {venue}")


        # Create the API request URL and parameters
        url = f"https://api.foursquare.com/v3/places/{fsq_id}/tips"

        headers = {
            "Accept": "application/json",
            "Authorization": API_KEY
        }


        # Make the GET request
        response = requests.get(url, headers=headers)

        # Check if the request was successful
        if response.status_code != 200:
            print(f"Failed to get data for {venue}. Status code: {response.
 ↪status_code}")
            print(f"Error message: {response.text}")
            continue

        results = response.json()

        # Parse the JSON response
        for tip in results:
            # Extract venue information
            tip_id = tip.get('id')
            tip_created = tip.get('created_at')
```

```
            tip_text = tip.get('text')


            # Append to the list
            venues_tips.append([
                venues,
                tip_id,
                tip_created,
                tip_text
            ])


    # Create a DataFrame from the list
    nearby_tips = pd.DataFrame(venues_tips, columns=[
        'venue_name',
        'id',
        'created_at',
        'text'
    ])

    return nearby_tips


nearby_tips = getTips(
    venues=search_area['Venue'],
    fsq_ids=search_area['fsq_id']
)

# Display the first few rows of the resulting DataFrame
nearby_tips.head()
```

```
Processing venue: Harbourfront Child Care Centre
Failed to get data for Harbourfront Child Care Centre. Status code: 404
Error message: invalid place specified: 98e7ba928c3940f6e973ca0f
```

[40]:
```
Empty DataFrame
Columns: [venue_name, id, created_at, text]
Index: []
```

0.0.4 Now it is your turn to discover interesting search results using Foursquare

You are expected to read through the API documentation and customize the searches.