

Exploring the Competitors

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.

```
In [1]: import requests
import pandas as pd
```

```
In [7]: toronto_DF = pd.read_csv('toronto_DF.csv', index_col = 0)
```

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

	Borough	Postalcode	Neighbourhood	Latitude	Longitude
0	Central Toronto	M4N	Lawrence Park	43.728020	-79.388790
1	Central Toronto	M4P	Davisville North	43.712751	-79.390197
2	Central Toronto	M4R	North Toronto West	43.715383	-79.405678
3	Central Toronto	M4S	Davisville	43.704324	-79.388790
4	Central Toronto	M4T	Moore Park, Summerhill East	43.689574	-79.383160
...
98	York	M6C	Humewood-Cedarvale	43.693781	-79.428191
99	York	M6E	Caledonia-Fairbanks	43.689026	-79.453512
100	York	M6M	Del Ray, Mount Dennis, Keelsdale and Silverthorn	43.691116	-79.476013
101	York	M6N	Runnymede, The Junction North	43.673185	-79.487262
102	York	M9N	Weston	43.706876	-79.518188

103 rows × 5 columns

```
In [11]: # Set up your API key and other constants for Foursquare API
API_KEY = 'fsq3t/Toy+772rvVo0H6URN+60+rUSA6RT+qtU2kgezM7Kk=' # Replace with you
```

```

LIMIT = 20 # Maximum number of venues to return
radius = 2500 # Search radius in meters

# a function to loop through each neighbourhood in the postalcode.csv file, and
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/
        # 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': 17135 # Retail -> Toy Store
        }

        # 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()

        # 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 None

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

```

```

        lng,
        venue_name,
        venue_id,
        venue_lat,
        venue_lng,
        venue_category
    ])

    # Create a DataFrame from the list
    nearby_venues = pd.DataFrame(venues_list, columns=[
        'Neighbourhood',
        'Postalcode',
        'Neighbourhood Latitude',
        'Neighbourhood 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=toronto_DF['Neighbourhood'],
    postalcode=toronto_DF['Postalcode'],
    latitudes=toronto_DF['Latitude'],
    longitudes=toronto_DF['Longitude'],
    radius=radius
)

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

```

Processing neighborhood: Lawrence Park
Processing neighborhood: Davisville North
Processing neighborhood: North Toronto West
Processing neighborhood: Davisville
Processing neighborhood: Moore Park, Summerhill East
Processing neighborhood: Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park
Processing neighborhood: Roselawn
Processing neighborhood: Forest Hill North & West
Processing neighborhood: The Annex, North Midtown, Yorkville
Processing neighborhood: Rosedale
Processing neighborhood: St. James Town, Cabbagetown
Processing neighborhood: Church and Wellesley
Processing neighborhood: Regent Park, Harbourfront
Processing neighborhood: Garden District, Ryerson
Processing neighborhood: St. James Town
Processing neighborhood: Berczy Park
Processing neighborhood: Central Bay Street
Processing neighborhood: Richmond, Adelaide, King
Processing neighborhood: Harbourfront East, Union Station, Toronto Islands
Processing neighborhood: Toronto Dominion Centre, Design Exchange
Processing neighborhood: Commerce Court, Victoria Hotel
Processing neighborhood: University of Toronto, Harbord
Processing neighborhood: Kensington Market, Chinatown, Grange Park
Processing neighborhood: CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay, South Niagara, Island airport
Processing neighborhood: First Canadian Place, Underground city
Processing neighborhood: Christie
Processing neighborhood: Enclave of M5E
Processing neighborhood: The Beaches
Processing neighborhood: The Danforth West, Riverdale
Processing neighborhood: India Bazaar, The Beaches West
Processing neighborhood: Studio District
Processing neighborhood: Enclave of M4L
Processing neighborhood: Parkview Hill, Woodbine Gardens
Processing neighborhood: Woodbine Heights
Processing neighborhood: Leaside
Processing neighborhood: Thorncliffe Park
Processing neighborhood: The Danforth East
Processing neighborhood: New Toronto, Mimico South, Humber Bay Shores
Processing neighborhood: Alderwood, Long Branch
Processing neighborhood: The Kingsway, Montgomery Road, Old Mill North
Processing neighborhood: Old Mill South, King's Mill Park, Sunnylea, Humber Bay, Mimico NE, The Queensway East, Royal York South East, Kingsway Park South East
Processing neighborhood: Mimico NW, The Queensway West, South of Bloor, Kingsway Park South West, Royal York South West
Processing neighborhood: Islington Avenue
Processing neighborhood: West Deane Park, Princess Gardens, Martin Grove, Islington, Cloverdale
Processing neighborhood: Eringate, Bloordale Gardens, Old Burnhamthorpe, Markland Wood
Processing neighborhood: Westmount
Processing neighborhood: Kingsview Village, St. Phillips, Martin Grove Gardens, Richmondview Gardens
Processing neighborhood: South Steeles, Silverstone, Humbergate, Jamestown, Mount Olive, Beaumont Heights, Thistletown, Albion Gardens
Processing neighborhood: Clairville, Humberwood, Woodbine Downs, West Humber, Kipling Heights, Rexdale, Elms, Tandridge, Old Rexdale
Processing neighborhood: Enclave of L4W
Processing neighborhood: Hillcrest Village

Processing neighborhood: Fairview, Henry Farm, Oriole
Processing neighborhood: Bayview Village
Processing neighborhood: York Mills, Silver Hills
Processing neighborhood: Willowdale, Newtonbrook
Processing neighborhood: Willowdale) Sout
Processing neighborhood: York Mills West
Processing neighborhood: Willowdale) Wes
Processing neighborhood: Parkwoods
Processing neighborhood: Don Mills) Nort
Processing neighborhood: Don Mills) South (Flemingdon Park
Processing neighborhood: Bathurst Manor, Wilson Heights, Downsview North
Processing neighborhood: Northwood Park, York University
Processing neighborhood: Downsview) East (CFB Toronto
Processing neighborhood: Downsview) Wes
Processing neighborhood: Downsview) Centra
Processing neighborhood: Downsview) Northwes
Processing neighborhood: Victoria Village
Processing neighborhood: Bedford Park, Lawrence Manor East
Processing neighborhood: Lawrence Manor, Lawrence Heights
Processing neighborhood: Glencairn
Processing neighborhood: North Park, Maple Leaf Park, Upwood Park
Processing neighborhood: Humber Summit
Processing neighborhood: Humberlea, Emery
Processing neighborhood: Ontario Provincial Government
Processing neighborhood: Malvern, Rouge
Processing neighborhood: Rouge Hill, Port Union, Highland Creek
Processing neighborhood: Guildwood, Morningside, West Hill
Processing neighborhood: Woburn
Processing neighborhood: Cedarbrae
Processing neighborhood: Scarborough Village
Processing neighborhood: Kennedy Park, Ionview, East Birchmount Park
Processing neighborhood: Golden Mile, Clairlea, Oakridge
Processing neighborhood: Cliffside, Cliffcrest, Scarborough Village West
Processing neighborhood: Birch Cliff, Cliffside West
Processing neighborhood: Dorset Park, Wexford Heights, Scarborough Town Centre
Processing neighborhood: Wexford, Maryvale
Processing neighborhood: Agincourt
Processing neighborhood: Clarks Corners, Tam O'Shanter, Sullivan
Processing neighborhood: Milliken, Agincourt North, Steeles East, L'Amoreaux East
Processing neighborhood: Steeles West, L'Amoreaux West
Processing neighborhood: Upper Rouge
Processing neighborhood: Dufferin, Dovercourt Village
Processing neighborhood: Little Portugal, Trinity
Processing neighborhood: Brockton, Parkdale Village, Exhibition Place
Processing neighborhood: High Park, The Junction South
Processing neighborhood: Parkdale, Roncesvalles
Processing neighborhood: Runnymede, Swansea
Processing neighborhood: Humewood-Cedarvale
Processing neighborhood: Caledonia-Fairbanks
Processing neighborhood: Del Ray, Mount Dennis, Keelsdale and Silverthorn
Processing neighborhood: Runnymede, The Junction North
Processing neighborhood: Weston

Out[11]:

	Neighbourhood	Postalcode	Neighbourhood Latitude	Neighbourhood Longitude	Venue	
0	Lawrence Park	M4N	43.72802	-79.38879	Mastermind Toys	4ac
1	Lawrence Park	M4N	43.72802	-79.38879	401 Games	5b
2	Lawrence Park	M4N	43.72802	-79.38879	Toys Toys Toys	4b7
3	Lawrence Park	M4N	43.72802	-79.38879	Game Trek Inc	4df
4	Lawrence Park	M4N	43.72802	-79.38879	Three Kingdoms Games	59

```
In [ ]: # Only keep toy stores (casino is also included, but it is not in our interest)
toronto_venues = toronto_venues[toronto_venues['Venue Category'] == 'Toy Store']
toronto_venues.to_csv('toronto_venues.csv')

# The total number of toy stores in Toronto
print(toronto_venues['Venue'].nunique())
```

118

```
In [30]: toronto_venues[toronto_venues['Postalcode'] == 'M4S']
```

Out[30]:

	Neighbourhood	Postalcode	Neighbourhood Latitude	Neighbourhood Longitude	Venue	
34	Davisville	M4S	43.704324	-79.38879	Mastermind Toys	4k
35	Davisville	M4S	43.704324	-79.38879	Little Dollhouse Co	4
36	Davisville	M4S	43.704324	-79.38879	Strategy Games	4
38	Davisville	M4S	43.704324	-79.38879	Crazy Frog Inc	.
39	Davisville	M4S	43.704324	-79.38879	Mind Games	56
40	Davisville	M4S	43.704324	-79.38879	Game Trek Inc	4c
41	Davisville	M4S	43.704324	-79.38879	Toys Toys Toys	4b
43	Davisville	M4S	43.704324	-79.38879	401 Games	5

```
In [27]: # Oberserving the number of competitors in each neighbourhood
df_competitors = toronto_venues.groupby('Neighbourhood')['Venue'].count()
df_competitors.sort_values()
```

```

Out[27]: Neighbourhood
York Mills, Silver Hills
1
Steeles West, L'Amoreaux West
1
West Deane Park, Princess Gardens, Martin Grove, Islington, Cloverdale
1
Scarborough Village
1
Clairville, Humberwood, Woodbine Downs, West Humber, Kipling Heights, Rexdale,
Elms, Tandridge, Old Rexdale      1

..
Central Bay Street
19
University of Toronto, Harbord
19
Garden District, Ryerson
19
St. James Town
19
Ontario Provincial Government
19
Name: Venue, Length: 93, dtype: int64

```

Part 2. Interactive leaflet map using coordinate data.

```

In [16]: import folium # map rendering library

```

```

In [31]: # My example below shows toy stores in Davisville
target = 'Davisville'

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

```

	Neighbourhood	Postalcode	Neighbourhood Latitude	Neighbourhood Longitude	Venue	
34	Davisville	M4S	43.704324	-79.38879	Mastermind Toys	4b8
35	Davisville	M4S	43.704324	-79.38879	Little Dollhouse Co	4dk
36	Davisville	M4S	43.704324	-79.38879	Strategy Games	4ce
38	Davisville	M4S	43.704324	-79.38879	Crazy Frog Inc	18
39	Davisville	M4S	43.704324	-79.38879	Mind Games	565
40	Davisville	M4S	43.704324	-79.38879	Game Trek Inc	4dfc
41	Davisville	M4S	43.704324	-79.38879	Toys Toys Toys	4b7e
43	Davisville	M4S	43.704324	-79.38879	401 Games	5b7

```
In [32]: 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
```

```
d:\Applications\Python\Python3107\lib\site-packages\folium\utilities.py:94: FutureWarning: Calling float on a single element Series is deprecated and will raise a
TypeError in the future. Use float(ser.iloc[0]) instead
float(coord)
```

```
d:\Applications\Python\Python3107\lib\site-packages\folium\utilities.py:100: FutureWarning: Calling float on a single element Series is deprecated and will raise
a TypeError in the future. Use float(ser.iloc[0]) instead
if math.isnan(float(coord)):
```

```
d:\Applications\Python\Python3107\lib\site-packages\folium\utilities.py:102: FutureWarning: Calling float on a single element Series is deprecated and will raise
a TypeError in the future. Use float(ser.iloc[0]) instead
return [float(x) for x in coords]
```


Out[32]:



 Leaflet | © OpenStreetMap contributors

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](#).

```
In [33]: # a function to loop through the list of pet groomers in the neighborhood and co
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"

        # Set up headers with the API key
        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.statu")
            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: Mastermind Toys
Processing venue: Little Dollhouse Co
Processing venue: Strategy Games
Processing venue: Crazy Frog Inc
Processing venue: Mind Games
Processing venue: Game Trek Inc
Processing venue: Toys Toys Toys
Processing venue: 401 Games

```

Out[33]:	venue_name		id	created_at	text
0	34 Mastermind Toys 35 Little Dollhou...	5ab6a229838e5939e696274e		2018-03-24T19:08:25.000Z	Love this place but bring your wallet because ...
1	34 Mastermind Toys 35 Little Dollhou...	4c9cc10a54c8a1cd6bfd7c4b		2010-09-24T15:17:30.000Z	Get on their email list - they mail out discou...
2	34 Mastermind Toys 35 Little Dollhou...	4d66d87859afa090e2a2983a		2011-02-24T22:15:20.000Z	We offer FREE Gift Wrapping on EVERYTHING in-s...
3	34 Mastermind Toys 35 Little Dollhou...	60318fa43fe3e5737d72dc4c		2021-02-20T22:39:32.000Z	are there fidget toys here?
4	34 Mastermind Toys 35 Little Dollhou...	55f18f9e498e8f61e64ffba4		2015-09-10T14:11:42.000Z	The most wholesome, kid-oriented place brights...

```
In [ ]: for review in nearby_tips['text']:
        print(review)
```

Love this place but bring your wallet because there is no tapping phones (or cards)

Get on their email list - they mail out discount coupons for up to 25% if you shop at particular times.

We offer FREE Gift Wrapping on EVERYTHING in-store!

are there fidget toys here?

The most wholesome, kid-oriented place brights your rainy days and give your mind a workout :) It's small, cheerful, colourful and for sure it's worth visiting!! :)

Surprisingly, this place is a chain. If they don't have what you're looking for, ask! They might be able to get it in from another location for you.

They provided a amazing kid's birthday party. Friendly staff as well !! Thank you!

Great store tons of toys for every age group. Fantastic prices.

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.