ExploringCompetitors

November 26, 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 we've chosen bagel shops 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.

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

[2]:
```
toronto_DF = pd.read_csv('exploring toronto neighbourhoods.csv')
```

[3]:
```
display(toronto_DF)
```

```
             Borough Postalcode  \
0    Central Toronto        M4N
1    Central Toronto        M4P
2    Central Toronto        M4R
3    Central Toronto        M4S
4    Central Toronto        M4T
..               …         …
98              York        M6C
99              York        M6E
100             York        M6M
101             York        M6N
102             York        M9N

                                         Neighbourhood   Latitude  Longitude
0                                        Lawrence Park  43.728020 -79.388790
1                                     Davisville North  43.712751 -79.390197
2                                   North Toronto West  43.715383 -79.405678
3                                           Davisville  43.704324 -79.388790
4                         Moore Park / Summerhill East  43.689574 -79.383160
..                                                  …         …         …
```

```
98                                  Humewood-Cedarvale  43.693781 -79.428191
99                                 Caledonia-Fairbanks  43.689026 -79.453512
100  Del Ray / Mount Dennis / Keelsdale and Silvert…  43.691116 -79.476013
101                       Runnymede / The Junction North  43.673185 -79.487262
102                                             Weston  43.706876 -79.518188

[103 rows x 5 columns]
```

[4]:

```
# Set up API key and other constants for Foursquare API
API_KEY = 'fsq3t+OW6hBVze7MGHEzJyy7agmYljQRdp9e7SzJC9vpmAk='
LIMIT = 20  # Maximum number of venues to return
radius = 2500  # Search radius in meters

# a function to loop through each neighbourhood in the exploring toronto␣
↪neighbourhoods.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
        # I am searching with categories ID 13001 - bagel shops
        url = 'https://api.foursquare.com/v3/places/search'
        params = {
            'll': f'{lat},{lng}',
            'radius': radius,
            'limit': LIMIT,
            'categories': 13001
        }

        # Set up headers with the API key
        headers = {
            "Accept": "application/json",
            "Authorization": 'fsq3t+OW6hBVze7MGHEzJyy7agmYljQRdp9e7SzJC9vpmAk='
        }

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

```python
        # 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 '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=[
        'Neighbourhood',
        'Postalcode',
        'Neighborhood Latitude',
        'Neighborhood Longitude',
        'Venue',
        'fsq_id',
        'Venue Latitude',
        'Venue Longitude',
        'Venue Category'
    ])
    return nearby_venues

# setting up
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: Flemingdon Par
Processing neighborhood: CFB Toront
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
/ Richview Gardens
Processing neighborhood: South Steeles / Silverstone / Humbergate / Jamestown /
Mount Olive / Beaumond 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 South
Processing neighborhood: York Mills West
Processing neighborhood: Willowdale West
Processing neighborhood: Parkwoods
Processing neighborhood: Don Mills North
Processing neighborhood: Bathurst Manor / Wilson Heights / Downsview North
Processing neighborhood: Northwood Park / York University
Processing neighborhood: Downsview West
Processing neighborhood: Downsview Central
Processing neighborhood: Downsview Northwest
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 OShanter / Sullivan
Processing neighborhood: Milliken / Agincourt North / Steeles East / LAmoreaux
East
Processing neighborhood: Steeles West / LAmoreaux 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
```

```
[4]:   Neighbourhood Postalcode  Neighborhood Latitude  Neighborhood Longitude  \
     0  Lawrence Park        M4N               43.72802               -79.38879
     1  Lawrence Park        M4N               43.72802               -79.38879
     2  Lawrence Park        M4N               43.72802               -79.38879
     3  Lawrence Park        M4N               43.72802               -79.38879
     4  Lawrence Park        M4N               43.72802               -79.38879

                  Venue                    fsq_id  Venue Latitude  \
     0      The Bagel House  4b783705f964a520febc2ee3       43.728357
     1          Bagel World  4eda86689a5286d91f850c6a       43.731513
     2               Kiva's  51ab8e4c498ea88f386e894f       43.713830
     3      The Bagel House  5812277938fa39a99b19c99d       43.714005
```

```
4  Kaivas Bakery Market  53ca9ca7498e9c40283136fd       43.712072

   Venue Longitude Venue Category
0       -79.418297     Bagel Shop
1       -79.406579     Bagel Shop
2       -79.399447     Bagel Shop
3       -79.400130     Bagel Shop
4       -79.399681     Bagel Shop
```

[5]:

```
# The total number of Bagel Shops in Toronto
toronto_venues['Venue'].nunique()
```

[5]:

```
67
```

[6]:

```
# Oberserving the number of competitors in each neighbourhood
count_competitors = toronto_venues.groupby('Neighbourhood')['Venue'].count()
print (count_competitors)

#Get count per neighbourhood and set up table to be exported
count_competitors_df = count_competitors.reset_index()
count_competitors_df.columns = ['Neighbourhood', 'Count']  # Rename columns for␣
 ↪clarity

# Export to csv for further analysis in Tableau
count_competitors_df.to_csv('competitors_count.csv', index=False)
```

```
Neighbourhood
Agincourt                                             1
Alderwood / Long Branch                               1
Bathurst Manor / Wilson Heights / Downsview North     3
Bedford Park / Lawrence Manor East                    8
Berczy Park                                          20
                                                     ..
Wexford / Maryvale                                    1
Willowdale South                                      2
Willowdale West                                       5
York Mills / Silver Hills                             1
York Mills West                                       3
Name: Venue, Length: 80, dtype: int64
```

0.0.2 Part 2. Interactive leaflet map using coordinate data.

[7]:

```
import folium # map rendering library
```

[8]:

```
# We first considered Downtown Toronto
target = 'University of Toronto / Harbord'
```

```
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  Neighborhood Latitude  \
355  University of Toronto / Harbord        M5S              43.662696
356  University of Toronto / Harbord        M5S              43.662696
357  University of Toronto / Harbord        M5S              43.662696
358  University of Toronto / Harbord        M5S              43.662696
359  University of Toronto / Harbord        M5S              43.662696
360  University of Toronto / Harbord        M5S              43.662696
361  University of Toronto / Harbord        M5S              43.662696
362  University of Toronto / Harbord        M5S              43.662696
363  University of Toronto / Harbord        M5S              43.662696
364  University of Toronto / Harbord        M5S              43.662696
365  University of Toronto / Harbord        M5S              43.662696
366  University of Toronto / Harbord        M5S              43.662696
367  University of Toronto / Harbord        M5S              43.662696
368  University of Toronto / Harbord        M5S              43.662696
369  University of Toronto / Harbord        M5S              43.662696
370  University of Toronto / Harbord        M5S              43.662696
371  University of Toronto / Harbord        M5S              43.662696
372  University of Toronto / Harbord        M5S              43.662696
373  University of Toronto / Harbord        M5S              43.662696
374  University of Toronto / Harbord        M5S              43.662696

     Neighborhood Longitude                       Venue  \
355              -79.400049                    Nu Bugel
356              -79.400049             Simit & Chai Co
357              -79.400049                What a Bagel
358              -79.400049            Kettlemans Bagel
359              -79.400049                Krispy Kreme
360              -79.400049                  Bagel Stop
361              -79.400049             Manulife Centre
362              -79.400049        Great Canadian Bagel
363              -79.400049  Tasty's Chicken and Donuts
364              -79.400049   Bagel Time Montreal Style
365              -79.400049              The Bagel Stop
366              -79.400049              The Bagel Stop
367              -79.400049     Bagel Stop College Park
368              -79.400049                  Bagel Stop
369              -79.400049        Great Canadian Bagel
370              -79.400049                  Bagel Time
371              -79.400049              Bagels on Fire
372              -79.400049              The Bagel Stop
373              -79.400049            Kiva's Bagel Bar
```

```
374             -79.400049                Bagels & Co

                        fsq_id  Venue Latitude  Venue Longitude Venue Category
355  510d3c8ae4b0a3ee4a9b231a       43.655370       -79.402606     Bagel Shop
356  5620d090498e997b0002172e       43.642941       -79.406793     Bagel Shop
357  4f7dd9a6e4b09d309a47f7fe       43.647638       -79.396230     Bagel Shop
358  62220a8077ee2a41ccbf70b5       43.641854       -79.401665     Bagel Shop
359  5760a4c4498e158c411e92e6       43.655931       -79.399517     Bagel Shop
360  4b7ea6e0f964a52056f72fe3       43.671249       -79.384937     Bagel Shop
361  4ad4c063f964a52024f820e3       43.669812       -79.388496     Bagel Shop
362  4b114167f964a5205e7923e3       43.648481       -79.382294     Bagel Shop
363  5dde0f08ed2ead0008e19e5c       43.655548       -79.398614     Bagel Shop
364  600067e8d9cc927aceb2c209       43.665144       -79.410246     Bagel Shop
365  4f560281e4b05b4bc3d4a8e0       43.653836       -79.405167     Bagel Shop
366  4e4a6f7cd22d876aed1947d0       43.659108       -79.388240     Bagel Shop
367  4cc817b7fe64ef3bf27da7f0       43.660217       -79.385331     Bagel Shop
368  5c5c5de1916bc1002c1abf8f       43.661128       -79.383606     Bagel Shop
369  506d800ee4b01fa633e4c36a       43.654895       -79.385185     Bagel Shop
370  4b7c0070f964a520e3772fe3       43.659720       -79.382119     Bagel Shop
371  5dfeb9e8e58b0b0007e974af       43.648509       -79.398146     Bagel Shop
372  4cc5b6a1d43ba14331606ff8       43.656575       -79.380590     Bagel Shop
373  58fe349ab3d8e242db5c3d41       43.650102       -79.384702     Bagel Shop
374  4c6e88f6f338236ae9a30a1b       43.649974       -79.383373     Bagel Shop
```

[9]:

```
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['Neighbourhood']):
    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
```

```
/opt/conda/lib/python3.11/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)
```

```
/opt/conda/lib/python3.11/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)):
/opt/conda/lib/python3.11/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]
```

[9]:

```
<folium.folium.Map at 0x7f410907b9d0>
```

[10]:

```python
# We now compare it with Midtown, Lawrence Park to get an idea of difference
target = 'Lawrence Park'

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  Neighborhood Latitude  Neighborhood Longitude  \
0  Lawrence Park        M4N               43.72802               -79.38879
1  Lawrence Park        M4N               43.72802               -79.38879
2  Lawrence Park        M4N               43.72802               -79.38879
3  Lawrence Park        M4N               43.72802               -79.38879
4  Lawrence Park        M4N               43.72802               -79.38879
5  Lawrence Park        M4N               43.72802               -79.38879
6  Lawrence Park        M4N               43.72802               -79.38879

                 Venue                    fsq_id  Venue Latitude  \
0      The Bagel House  4b783705f964a520febc2ee3       43.728357
1          Bagel World  4eda86689a5286d91f850c6a       43.731513
2               Kiva's  51ab8e4c498ea88f386e894f       43.713830
3      The Bagel House  5812277938fa39a99b19c99d       43.714005
4  Kaivas Bakery Market  53ca9ca7498e9c40283136fd       43.712072
5     Kettlemans Bagel  6570ddb885158224c961102c       43.707170
6           Bagel Stop  4b8d4f68f964a520a2f332e3       43.707140

   Venue Longitude Venue Category
0       -79.418297     Bagel Shop
1       -79.406579     Bagel Shop
2       -79.399447     Bagel Shop
3       -79.400130     Bagel Shop
4       -79.399681     Bagel Shop
5       -79.397200     Bagel Shop
6       -79.398753     Bagel Shop
```

[11]:

```
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['Neighbourhood']):
    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
```

```
/opt/conda/lib/python3.11/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)
/opt/conda/lib/python3.11/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)):
/opt/conda/lib/python3.11/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]
```

[11]:

```
<folium.folium.Map at 0x7f410907a010>
```

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.

[12]:

```
# a function to loop through the list of Bagel Shops 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": 'fsq3t+OW6hBVze7MGHEzJyy7agmYljQRdp9e7SzJC9vpmAk='
        }

        # 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: The Bagel House
Processing venue: Bagel World
Processing venue: Kiva's
Processing venue: The Bagel House
Processing venue: Kaivas Bakery Market
Processing venue: Kettlemans Bagel
Processing venue: Bagel Stop
```

[12]:

```
                                          venue_name  \
0  0         The Bagel House
1             Bagel …
1  0         The Bagel House
1             Bagel …
2  0         The Bagel House
1             Bagel …
3  0         The Bagel House
1             Bagel …
4  0         The Bagel House
1             Bagel …

                         id                created_at  \
0  58153fa438fa7e98965eff1d  2016-10-30T00:32:36.000Z
1  4ee4c6a46c25be9630b53548  2011-12-11T15:05:08.000Z
2  5d55d88394b4af000704bf4d  2019-08-15T22:11:15.000Z
3  4eb56643e5fa17fc86bfba92  2011-11-05T16:37:23.000Z
4  54be213b498e236c01226767  2015-01-20T09:34:51.000Z

                                                text
0  Our go to place for Breakfast bagels and Lox &…
1  These guys stole the bagel recipe from Montrea…
2   Fresh bagel with smoked beef was very delicious!
3  Don't expect friendly service here.  Staff are…
4                    Amazing spot, lots of variety.
```

[13]:

```
nearby_tips.to_csv('bagel_house_tips.csv', index = False)
```

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.