Exploring Toronto Neighborhoods to open an Chinese Restaurant.

As a part of the final IBM Capstone Project, we work on the real dataset to get an exprencies of what a data scientists go through in real life. Objectives of the final assignments were to define a business problem, look for data in the web and, use Foursquare location data to compare different neighborhoods of Toronto to figure out which neighborhood is suitable for starting a new restaurant business. In this project, I go through all the process in a step by step manner from problem designing, data preparation to final analysis and finally will provide a conclusion that can be leveraged by the business stakeholders to make their decisions.

1. Description of the Problem and Discussion of the Background (Introduction Section)

Prospects of a opening an Chinese Restaurant in Toronto, Canada.

Toronto, the capital of the province of Ontario, is the most populous Canadian city. Its diversity is reflected in Toronto's ethnic neighborhoods such as Chinatown, Corso Italia, Greektown, Kensington Market, Koreatown, Little India, Little Italy, Little Jamaica, Little Portugal & Roncesvalles. One of the most immigrant-friendly cities in North America with more than half of the entire Chinese Canadian population residing in Toronto it is one of the best places to start an Chinese restaurant.

In this project we will go through step by step process to make a decision whether it is a good idea to open an Chinese restaurant. We analyze the neighborhoods in Toronto to identify the most profitable area since the success of the restaurant depends on the people and ambience. Since we already know that Toronto shelter a greater number of Chinese than any other city in Canada, it is a good idea to start the restaurant here, but we just need to make sure whether it is a profitable idea or not. If so, where we can place it, so it yields more profit to the owner.

Target Audience

Who will be more interested in this project? What type of clients or a group of people would be benefitted?

1. **Business personnel who wants to invest or open an Chinese restaurant in Toronto. This analysis will be a comprehensive guide to start or expand restaurants targeting the Chinese crowd.**
2. **Freelancer who loves to have their own restaurant as a side business. This analysis will give an idea, how beneficial it is to open a restaurant and what are the pros and cons of this business.**
3. **Chinese crowd who wants to find neighborhoods with lots of option for Chinese restaurants.**
4. **Business Analyst or Data Scientists, who wish to analyze the neighborhoods of Toronto using Exploratory Data Analysis and other statistical & machine learning techniques to obtain all the necessary data, perform some operations on it and, finally be able to tell a story out of it.**

2. Data acquisition and cleaning

2.1 Data Sources

a) I'm using "List of Postal code of Canada: M" (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M) wiki page to get all the information about the neighborhoods present in Toronto. This page has the postal code, borough & the name of all the neighborhoods present in Toronto.

b) Then I'm using "https://cocl.us/Geospatial_data" csv file to get all the geographical coordinates of the neighborhoods.

c) To get information about the distribution of population by their ethnicity I'm using "Demographics of Toronto" (https://en.m.wikipedia.org/wiki/Demographics_of_Toronto#Ethnic_diversity) wiki page. Using this page I'm going to identify the neighborhoods which are densely populated with Chinese as it might be helpful in identifying the suitable neighborhood to open a new Chinese restaurant.

d) To get location and other information about various venues in Toronto I'm using Foursquare's explore API. Using the Foursquare's explore API (which gives venues recommendations), I'm fetching details about the venues up present in Toronto and collected their names, categories and locations (latitude and longitude). From Foursquare API (https://developer.foursquare.com/docs), I retrieved the following for each venue:

- Name: The name of the venue.
- Category: The category type as defined by the API.
- Latitude: The latitude value of the venue.
- Longitude: The longitude value of the venue.

2.2 Data Cleaning

a) Scraping Toronto Neighborhoods Table from Wikipedia

Scraped the following Wikipedia page, "List of Postal code of Canada: M" in order to obtain the data about the Toronto & the Neighborhoods in it.

Assumptions made to attain the below DataFrame:

- Dataframe will consist of three columns: PostalCode, Borough, and Neighborhood
- Only the cells that have an assigned borough will be processed. Borough that is not assigned are ignored.
- More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that M5A is listed twice and has two neighborhoods: Harbourfront and Regent Park. These two rows will be combined into one row with the neighborhoods separated with a comma as shown in row 11 in the above table.
- If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough.

wikipedia - package is used to scrape the data from wiki.

In [12]:

```
!conda install -c conda-forge wikipedia --yes

import pandas as pd
import numpy as np
import wikipedia as wp
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

In [13]:

```
html = wp.page("List of postal codes of Canada: M").html().encode("UTF-8")
df = pd.read_html(html, header = 0)[0]
df.head()
```

Out[13]:

	Postcode	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods

	Postcode	Borough	Neighbourhood
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront

In [14]:

```
#Only process the cells that have an assigned borough. Ignore cells with a borough that i
s Not assigned.
df = df[df.Borough != 'Not assigned']
df = df.rename(columns={'Postcode': 'Postalcode'})

#If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be t
he same as the borough.
#So for the 9th cell in the table on the Wikipedia page, the value of the Borough and the
Neighborhood columns will be Queen's Park.
for index, row in df.iterrows():
    if row['Neighbourhood'] == 'Not assigned':
        row['Neighbourhood'] = row['Borough']


df.head()
```

Out[14]:

	Postalcode	Borough	Neighbourhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront
5	M6A	North York	Lawrence Heights
6	M6A	North York	Lawrence Manor

In [15]:

```
df = df.groupby(['Borough', 'Postalcode'])['Neighbourhood'].apply(list).apply(lambda x:'
, '.join(x)).to_frame().reset_index()
df.head()
```

Out[15]:

	Borough	Postalcode	Neighbourhood
0	Central Toronto	M4N	Lawrence Park
1	Central Toronto	M4P	Davisville North
2	Central Toronto	M4R	North Toronto West
3	Central Toronto	M4S	Davisville
4	Central Toronto	M4T	Moore Park, Summerhill East

b) Adding geographical coordinates to the neighborhoods

Next important step is adding the geographical coordinates to these neighborhoods. To do so I'm extracting the data present in the Geospatial Data csv file and I'm combining it with the existing neighborhood dataframe by merging them both based on the postal code.

In [16]:

```
#Reading the latitude & longitude data from CSV file

import io
import requests

url = "https://cocl.us/Geospatial_data"
lat_long = requests.get(url).text
```

```
lat_long_df=pd.read_csv(io.StringIO(lat_long))
lat_long_df.head()
```

Out[16]:

	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476

I'm renaming the columns to match the existing dataframe

In [17]:

```
lat_long_df = lat_long_df.rename(columns={'Postal Code': 'Postalcode'})
lat_long_df.head()
```

Out[17]:

	Postalcode	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476

After that I'm merging both the dataframe into one by matching on the postal code.

In [18]:

```
toronto_DF = pd.merge(df,lat_long_df, on='Postalcode')
toronto_DF = toronto_DF.rename(columns={'Neighbourhood':'Neighborhood'})
toronto_DF.head()
```

Out[18]:

	Borough	Postalcode	Neighborhood	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

In [19]:

```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(toronto_DF['Borough'].unique()),
        toronto_DF.shape[0]
    )
)
```

```
The dataframe has 11 boroughs and 103 neighborhoods.
```

c) Scrap the distribution of population from Wikipedia

Another factor that can help us in deciding which neighborhood would be best option to open a restaurant is

Another factor that can help us in deciding which neighborhood would be best option to open a restaurant is, the distribution of population based on the ethnic diversity for each neighborhood. As this helps us in identifying the neighborhoods which are densely populated with Chinese crowd since that neighborhood would be an ideal place to open an Chinese restaurant.

Scraped the following Wikipedia page, “Demographics of Toronto” in order to obtain the data about the Toronto & the Neighborhoods in it. Compared to all the neighborhoods in Toronto below given neighborhoods only had considerable amount of Chinese crowd. We are examing those neighborhood's population to identify the densely populated neighborhoods with Chinese population.

In [20]:

```
#overall population distribution
html = wp.page("Demographics of Toronto").html().encode("UTF-8")
```

In [21]:

```
#TORONTO & EAST YORK population distribution by ethnicity
TEY_population_df = pd.read_html(html, header = 0)[13]
TEY_population_df = TEY_population_df.rename(columns={'%':'Ethnic Origin 1 in %',
                                                       '%.1':'Ethnic Origin 2 in %',
                                                      '%.2':'Ethnic Origin 3 in %',
                                                      '%.3':'Ethnic Origin 4 in %',
                                                      '%.4':'Ethnic Origin 5 in %',
                                                      '%.5':'Ethnic Origin 6 in %',
                                                      '%.6':'Ethnic Origin 7 in %',
                                                      '%.7':'Ethnic Origin 8 in %',
                                                      '%.8':'Ethnic Origin 9 in %'})
TEY_population_df
```

Out[21]:

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %	Ethr Orig
0	Spadina-Fort York	114315	English	16.4	Chinese	16.0	Irish	14.6	Canadian	14.0	Scottish	13.2	Fren
1	Beaches-East York	108435	English	24.2	Irish	19.9	Canadian	19.7	Scottish	18.9	French	8.7	Germ
2	Davenport	107395	Portuguese	22.7	English	13.6	Canadian	12.8	Irish	11.5	Italian	11.1	Scotti
3	Parkdale-High Park	106445	English	22.3	Irish	20.0	Scottish	18.7	Canadian	16.1	German	9.8	Fren
4	Toronto-Danforth	105395	English	22.9	Irish	19.5	Scottish	18.7	Canadian	18.4	Chinese	13.8	Fren
5	Toronto-St. Paul's	104940	English	18.5	Canadian	16.1	Irish	15.2	Scottish	14.8	Polish	10.3	Germ
6	University-Rosedale	100520	English	20.6	Irish	16.6	Scottish	16.3	Canadian	15.2	Chinese	14.7	Germ
7	Toronto Centre	99590	English	15.7	Canadian	13.7	Irish	13.4	Scottish	12.6	Chinese	12.5	Fren

In [22]:

```
#NORTH YORK population distribution by ethnicity
North_population_df = pd.read_html(html, header = 0)[14]
North_population_df = North_population_df.rename(columns={'%':'Ethnic Origin 1 in %',
                                                       '%.1':'Ethnic Origin 2 in %',
                                                      '%.2':'Ethnic Origin 3 in %',
                                                      '%.3':'Ethnic Origin 4 in %',
                                                      '%.4':'Ethnic Origin 5 in %',
                                                      '%.5':'Ethnic Origin 6 in %',
                                                      '%.6':'Ethnic Origin 7 in %',
                                                      '%.7':'Ethnic Origin 8 in %'})
North_population_df
```

Out[22]:

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %	Et Or
0	Willowdale	117405	Chinese	25.9	Iranian	12.1	Korean	10.6	NaN	NaN	NaN	NaN	I
1	Eglinton-Lawrence	112925	Canadian	14.7	English	12.6	Polish	12.0	Filipino	11.0	Scottish	9.7	Ita
2	Don Valley North	109060	Chinese	32.4	East Indian	7.3	Iranian	7.3	NaN	NaN	NaN	NaN	I
3	Humber River-Black Creek	107725	Italian	12.8	East Indian	9.2	Jamaican	8.5	Vietnamese	8.0	Canadian	7.4	I
4	York Centre	103760	Filipino	17.0	Italian	13.4	Russian	9.5	Canadian	8.6	NaN	NaN	I
5	Don Valley West	101790	English	19.2	Canadian	15.1	Scottish	14.9	Irish	14.2	Chinese	11.2	I
6	Don Valley East	93170	East Indian	10.6	Canadian	10.4	English	10.1	Chinese	8.9	Irish	8.1	Scot

In [23]:

```
#SCARBOROUGH population distribution by ethnicity
Scar_population_df = pd.read_html(html, header = 0)[15]
Scar_population_df = Scar_population_df.rename(columns={'%':'Ethnic Origin 1 in %',
                                                         '%.1':'Ethnic Origin 2 in %',
                                                        '%.2':'Ethnic Origin 3 in %',
                                                        '%.3':'Ethnic Origin 4 in %',
                                                        '%.4':'Ethnic Origin 5 in %',
                                                        '%.5':'Ethnic Origin 6 in %',
                                                        '%.6':'Ethnic Origin 7 in %',
                                                        '%.7':'Ethnic Origin 8 in %'})
Scar_population_df
```

Out[23]:

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %	Ethi Origin
0	Scarborough Centre	110450	Filipino	13.1	East Indian	12.2	Canadian	11.2	Chinese	10.7	English	7.8	Lank
1	Scarborough Southwest	108295	Canadian	16.2	English	14.3	Irish	11.5	Scottish	10.9	Filipino	9.5	Ea Indi
2	Scarborough-Agincourt	104225	Chinese	47.0	East Indian	7.4	NaN	NaN	NaN	NaN	NaN	NaN	N
3	Scarborough-Rouge Park	101445	East Indian	16.7	Canadian	11.8	Sri Lankan	11.1	English	9.8	Filipino	9.3	Jamaic
4	Scarborough-Guildwood	101115	East Indian	18.0	Canadian	11.6	English	9.7	Filipino	8.5	Sri Lankan	7.8	Chine
5	Scarborough North	97610	Chinese	46.6	East Indian	11.8	Sri Lankan	9.4	NaN	NaN	NaN	NaN	N

In [24]:

```
#ETOBICOKE & YORK population distribution by ethnicity
ETY_population_df = pd.read_html(html, header = 0)[16]
ETY_population_df = ETY_population_df.rename(columns={'%':'Ethnic Origin 1 in %',
                                                       '%.1':'Ethnic Origin 2 in %',
                                                      '%.2':'Ethnic Origin 3 in %',
                                                      '%.3':'Ethnic Origin 4 in %',
```

```
                                          '%.4':'Ethnic Origin 5 in %',
                                          '%.5':'Ethnic Origin 6 in %',
                                          '%.6':'Ethnic Origin 7 in %',
                                          '%.7':'Ethnic Origin 8 in %'})
ETY_population_df
```

Out[24]:

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %	Etl Origi
0	Etobicoke-Lakeshore	127520	English	17.1	Canadian	15.9	Irish	14.4	Scottish	13.5	Polish	9.2	Ita
1	Etobicoke North	116960	East Indian	22.2	Canadian	7.9	NaN	NaN	NaN	NaN	NaN	NaN	I
2	Etobicoke Centre	116055	Italian	15.1	English	14.3	Canadian	12.1	Irish	10.8	Scottish	10.4	Ukrai
3	York South-Weston	115130	Portuguese	14.5	Italian	12.8	Canadian	8.7	Jamaican	8.4	NaN	NaN	I

d) Get location data using Foursquare

Foursquare API is very usefule online application used my many developers & other application like Uber etc. In this project I have 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 100 popular spots for each neighborhood within a radius of 1km.

In [25]:

```
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # convert an address into latitude and longitude va
lues
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

In [26]:

```
#Use geopy library to get the latitude and longitude values of Toronto City.

address = 'Toronto'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Toronto are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Toronto are 43.653963, -79.387207.
```

In [27]:

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

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

In [31]:

In [31]:

```
CLIENT_ID = 'HT24SLU3L4XRECAPPE0HK5SGMTQ54DLOIMRMOXUAHUB23P22'
CLIENT_SECRET = '3PH5YFS4I3XAH4JMEZLE5O1GI2EQNLMXQKGMJ5EP5B5TTZPH'
VERSION = '20180605'

radius=1000
url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{
}&v={}&radius={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, radius)
results = requests.get(url).json()
```

In [32]:

```
#Function to get the category

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']
```

Using the get_category_type function, we clean up the json and turn it into a pandas dataframe. Before we start that we need to import certain libraries.

In [33]:

```
import json
from pandas.io.json import json_normalize

venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.locati
on.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

Out[33]:

	name	categories	lat	lng
0	Downtown Toronto	Neighborhood	43.653232	-79.385296
1	Japango	Sushi Restaurant	43.655268	-79.385165
2	Rolltation	Japanese Restaurant	43.654918	-79.387424
3	Nathan Phillips Square	Plaza	43.652270	-79.383516
4	Sansotei Ramen 三草亭	Ramen Restaurant	43.655157	-79.386501

In [38]:

```
results['response']['groups'][0]['items']
```

Out[38]:


```
[{'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '5227bb01498e17bf485e6202',
```

```
    'name': 'Downtown Toronto',
    'location': {'lat': 43.65323167517444,
     'lng': -79.38529600606677,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.65323167517444,
       'lng': -79.38529600606677}],
     'distance': 174,
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
     'formattedAddress': ['Toronto ON', 'Canada']},
    'categories': [{'id': '4f2a25ac4b909258e854f55f',
      'name': 'Neighborhood',
      'pluralName': 'Neighborhoods',
      'shortName': 'Neighborhood',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/parks_outdoors/neighborho
od_',
       'suffix': '.png'},
      'primary': True}],
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-5227bb01498e17bf485e6202-0'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '4ae7b27df964a52068ad21e3',
    'name': 'Japango',
    'location': {'address': '122 Elizabeth St.',
     'crossStreet': 'at Dundas St. W',
     'lat': 43.65526771691681,
     'lng': -79.38516506734886,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.65526771691681,
       'lng': -79.38516506734886}],
     'distance': 219,
     'postalCode': 'M5G 1P5',
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
     'formattedAddress': ['122 Elizabeth St. (at Dundas St. W)',
      'Toronto ON M5G 1P5',
      'Canada']},
    'categories': [{'id': '4bf58dd8d48988d1d2941735',
      'name': 'Sushi Restaurant',
      'pluralName': 'Sushi Restaurants',
      'shortName': 'Sushi',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/sushi_',
       'suffix': '.png'},
      'primary': True}],
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-4ae7b27df964a52068ad21e3-1'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '5773f01f498e98371390bdfd',
    'name': 'Rolltation',
    'location': {'address': '207 Dundas St W',
     'crossStreet': 'at University Ave',
     'lat': 43.65491791857301,
     'lng': -79.3874242454196,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.65491791857301,
       'lng': -79.3874242454196}],
     'distance': 107,
     'postalCode': 'M5G 1C8',
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
```

```
    'formattedAddress': ['207 Dundas St W (at University Ave)',
     'Toronto ON M5G 1C8',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d111941735',
     'name': 'Japanese Restaurant',
     'pluralName': 'Japanese Restaurants',
     'shortName': 'Japanese',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/japanese_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-5773f01f498e98371390bdfd-2'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4ad4c05ef964a520a6f620e3',
   'name': 'Nathan Phillips Square',
   'location': {'address': '100 Queen St W',
    'crossStreet': 'at Bay St',
    'lat': 43.65227047322295,
    'lng': -79.38351631164551,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.65227047322295,
      'lng': -79.38351631164551}],
    'distance': 351,
    'postalCode': 'M5H 2N1',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['100 Queen St W (at Bay St)',
     'Toronto ON M5H 2N1',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d164941735',
     'name': 'Plaza',
     'pluralName': 'Plazas',
     'shortName': 'Plaza',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/parks_outdoors/plaza_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4ad4c05ef964a520a6f620e3-3'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '504bbf2ce4b0168121235cbe',
   'name': 'Sansotei Ramen 三草亭',
   'location': {'address': '179 Dundas St. W',
    'crossStreet': 'btwn Centre Ave. & Chestnut St.',
    'lat': 43.655157467561246,
    'lng': -79.38650067479335,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.655157467561246,
      'lng': -79.38650067479335}],
    'distance': 144,
    'postalCode': 'M5G 1Z8',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['179 Dundas St. W (btwn Centre Ave. & Chestnut St.)',
     'Toronto ON M5G 1Z8',
     'Canada']},
   'categories': [{'id': '55a59bace4b013909087cb24',
     'name': 'Ramen Restaurant',
     'pluralName': 'Ramen Restaurants',
     'shortName': 'Ramen',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/ramen_',
      'suffix': '.png'},
     'primary': True}],
```

```
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-504bbf2ce4b0168121235cbe-4'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '5a6b737b35f98359eed11974',
    'name': 'The Library Specialty Coffee',
    'location': {'address': '281 Dundas St West',
     'crossStreet': 'St Patrick And Dundas St W',
     'lat': 43.65441282740799,
     'lng': -79.39090161351724,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.65441282740799,
       'lng': -79.39090161351724}],
     'distance': 301,
     'postalCode': 'M5T 2W5',
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
     'formattedAddress': ['281 Dundas St West (St Patrick And Dundas St W)',
      'Toronto ON M5T 2W5',
      'Canada']},
    'categories': [{'id': '4bf58dd8d48988d1e0931735',
      'name': 'Coffee Shop',
      'pluralName': 'Coffee Shops',
      'shortName': 'Coffee Shop',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/coffeeshop_',
       'suffix': '.png'},
      'primary': True}],
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-5a6b737b35f98359eed11974-5'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '4c90c810ae96a093599f9d46',
    'name': "Karine's",
    'location': {'address': '109 McCaul St.',
     'crossStreet': 'at Dundas St. W',
     'lat': 43.653698928318526,
     'lng': -79.39074313568692,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.653698928318526,
       'lng': -79.39074313568692}],
     'distance': 286,
     'postalCode': 'M5T 3K5',
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
     'formattedAddress': ['109 McCaul St. (at Dundas St. W)',
      'Toronto ON M5T 3K5',
      'Canada']},
    'categories': [{'id': '4bf58dd8d48988d143941735',
      'name': 'Breakfast Spot',
      'pluralName': 'Breakfast Spots',
      'shortName': 'Breakfast',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/breakfast_',
       'suffix': '.png'},
      'primary': True}],
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-4c90c810ae96a093599f9d46-6'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '56ccd5cfcd1069ca160a797e',
    'name': 'Tsujiri',
    'location': {'address': '147 Dundas St W',
     'crossStreet': 'at Elizabeth St',
```

'lat': 43.65537430780922,
'lng': -79.38535434742991,
'labeledLatLngs': [{'label': 'display',
'lat': 43.65537430780922,
'lng': -79.38535434742991}],
'distance': 216,
'postalCode': 'M5G 1P5',
'cc': 'CA',
'city': 'Toronto',
'state': 'ON',
'country': 'Canada',
'formattedAddress': ['147 Dundas St W (at Elizabeth St)',
'Toronto ON M5G 1P5',
'Canada']},
'categories': [{'id': '4bf58dd8d48988d1dc931735',
'name': 'Tea Room',
'pluralName': 'Tea Rooms',
'shortName': 'Tea Room',
'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/tearoom_',
'suffix': '.png'},
'primary': True}],
'photos': {'count': 0, 'groups': []}},
'referralId': 'e-0-56ccd5cfcd1069ca160a797e-7'},
{'reasons': {'count': 0,
'items': [{'summary': 'This spot is popular',
'type': 'general',
'reasonName': 'globalInteractionReason'}]},
'venue': {'id': '4ad9f607f964a520691c21e3',
'name': 'Manpuku まんぷく',
'location': {'address': '105 McCaul St. Unit 29-31',
'crossStreet': 'at Dundas St. W.',
'lat': 43.653612411792935,
'lng': -79.39061276446213,
'labeledLatLngs': [{'label': 'display',
'lat': 43.653612411792935,
'lng': -79.39061276446213}],
'distance': 277,
'postalCode': 'M5T 2X4',
'cc': 'CA',
'city': 'Toronto',
'state': 'ON',
'country': 'Canada',
'formattedAddress': ['105 McCaul St. Unit 29-31 (at Dundas St. W.)',
'Toronto ON M5T 2X4',
'Canada']},
'categories': [{'id': '4bf58dd8d48988d111941735',
'name': 'Japanese Restaurant',
'pluralName': 'Japanese Restaurants',
'shortName': 'Japanese',
'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/japanese_',
'suffix': '.png'},
'primary': True}],
'photos': {'count': 0, 'groups': []}},
'referralId': 'e-0-4ad9f607f964a520691c21e3-8'},
{'reasons': {'count': 0,
'items': [{'summary': 'This spot is popular',
'type': 'general',
'reasonName': 'globalInteractionReason'}]},
'venue': {'id': '4adf3c01f964a5208f7821e3',
'name': 'Aboveground Art Supplies',
'location': {'address': '74 McCaul St',
'crossStreet': 'Dundas St.',
'lat': 43.652645648070006,
'lng': -79.3909251764317,
'labeledLatLngs': [{'label': 'display',
'lat': 43.652645648070006,
'lng': -79.3909251764317}],
'distance': 333,
'postalCode': 'M5T 3K2',
'cc': 'CA',
'city': 'Toronto',
'state': 'ON',

```
    'country': 'Canada',
    'formattedAddress': ['74 McCaul St (Dundas St.)',
     'Toronto ON M5T 3K2',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d127951735',
     'name': 'Arts & Crafts Store',
     'pluralName': 'Arts & Crafts Stores',
     'shortName': 'Arts & Crafts',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/artstore_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4adf3c01f964a5208f7821e3-9'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4ad4c062f964a520e5f720e3',
   'name': 'Four Seasons Centre for the Performing Arts',
   'location': {'address': '145 Queen St. W',
    'crossStreet': 'at University Ave.',
    'lat': 43.650592,
    'lng': -79.385806,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.650592,
      'lng': -79.385806}],
    'distance': 391,
    'postalCode': 'M5H 4G1',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['145 Queen St. W (at University Ave.)',
     'Toronto ON M5H 4G1',
     'Canada']},
   'categories': [{'id': '5032792091d4c4b30a586d5c',
     'name': 'Concert Hall',
     'pluralName': 'Concert Halls',
     'shortName': 'Concert Hall',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/musicv
enue_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4ad4c062f964a520e5f720e3-10'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4e2284b11fc7c0ef9857d143',
   'name': 'Chatime 日出茶太',
   'location': {'address': '132 Dundas St W',
    'crossStreet': 'btwn Bay & University',
    'lat': 43.65554164147378,
    'lng': -79.38468427043244,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.65554164147378,
      'lng': -79.38468427043244}],
    'distance': 268,
    'postalCode': 'M5G 1C3',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['132 Dundas St W (btwn Bay & University)',
     'Toronto ON M5G 1C3',
     'Canada']},
   'categories': [{'id': '52e81612bcbc57f1066b7a0c',
     'name': 'Bubble Tea Shop',
     'pluralName': 'Bubble Tea Shops',
     'shortName': 'Bubble Tea',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/bubble_',
```

```
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4e2284b11fc7c0ef9857d143-11'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '57bcd3b7498e652a678d0378',
   'name': 'Poke Guys',
   'location': {'address': '112 Elizabeth St',
    'crossStreet': 'at Dundas St W',
    'lat': 43.65489527525682,
    'lng': -79.38505238381624,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.65489527525682,
      'lng': -79.38505238381624}],
    'distance': 202,
    'postalCode': 'M5G 1P5',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['112 Elizabeth St (at Dundas St W)',
     'Toronto ON M5G 1P5',
     'Canada']},
   'categories': [{'id': '5bae9231bedf3950379f89d4',
     'name': 'Poke Place',
     'pluralName': 'Poke Places',
     'shortName': 'Poke Place',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/default_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-57bcd3b7498e652a678d0378-12'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4ad4c05ef964a520daf620e3',
   'name': 'Art Gallery of Ontario',
   'location': {'address': '317 Dundas St W',
    'crossStreet': 'at Beverley St',
    'lat': 43.654002860337386,
    'lng': -79.39292172707437,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.654002860337386,
      'lng': -79.39292172707437}],
    'distance': 460,
    'postalCode': 'M5T 1G4',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['317 Dundas St W (at Beverley St)',
     'Toronto ON M5T 1G4',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1e2931735',
     'name': 'Art Gallery',
     'pluralName': 'Art Galleries',
     'shortName': 'Art Gallery',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/artgal
lery_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []},
   'venuePage': {'id': '33853777'}},
  'referralId': 'e-0-4ad4c05ef964a520daf620e3-13'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
```

```
   'venue': {'id': '4ad4c064f964a52065f820e3',
    'name': 'Ontario College of Art and Design University (OCADU)',
    'location': {'address': '100 McCaul St',
     'crossStreet': 'at Dundas St W',
     'lat': 43.65280251171013,
     'lng': -79.3910743699992,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.65280251171013,
       'lng': -79.3910743699992}],
     'distance': 337,
     'postalCode': 'M5T 1W1',
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
     'formattedAddress': ['100 McCaul St (at Dundas St W)',
      'Toronto ON M5T 1W1',
      'Canada']},
    'categories': [{'id': '4bf58dd8d48988d1ae941735',
      'name': 'University',
      'pluralName': 'Universities',
      'shortName': 'University',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/education/default_',
       'suffix': '.png'},
      'primary': True}],
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-4ad4c064f964a52065f820e3-14'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '566f1451498ed44bec09c4f2',
    'name': 'Provo Food Bar',
    'location': {'address': '308 Dundas Street West',
     'crossStreet': 'McCaul',
     'lat': 43.654293123974384,
     'lng': -79.39178769369552,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.654293123974384,
       'lng': -79.39178769369552}],
     'distance': 370,
     'postalCode': 'M5T 1G4',
     'cc': 'CA',
     'city': 'Toronto',
     'state': 'ON',
     'country': 'Canada',
     'formattedAddress': ['308 Dundas Street West (McCaul)',
      'Toronto ON M5T 1G4',
      'Canada']},
    'categories': [{'id': '4bf58dd8d48988d1db931735',
      'name': 'Tapas Restaurant',
      'pluralName': 'Tapas Restaurants',
      'shortName': 'Tapas',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/tapas_',
       'suffix': '.png'},
      'primary': True}],
    'photos': {'count': 0, 'groups': []}},
   'referralId': 'e-0-566f1451498ed44bec09c4f2-15'},
  {'reasons': {'count': 0,
    'items': [{'summary': 'This spot is popular',
      'type': 'general',
      'reasonName': 'globalInteractionReason'}]},
   'venue': {'id': '57745cb8498ef903ab623cca',
    'name': 'Fugo Desserts',
    'location': {'address': '205 Dundas Street W',
     'crossStreet': 'at University Ave',
     'lat': 43.654922556722134,
     'lng': -79.38738186530605,
     'labeledLatLngs': [{'label': 'display',
       'lat': 43.654922556722134,
       'lng': -79.38738186530605}],
     'distance': 107,
```

```
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['205 Dundas Street W (at University Ave)',
     'Toronto ON',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1c9941735',
     'name': 'Ice Cream Shop',
     'pluralName': 'Ice Cream Shops',
     'shortName': 'Ice Cream',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/icecream_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-57745cb8498ef903ab623cca-16'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4b7ed424f964a5208a0230e3',
   'name': 'Friendly Stranger - Cannabis Culture Shop',
   'location': {'address': '241 Queen St. W',
    'crossStreet': 'at St. Patrick St.',
    'lat': 43.650386666114635,
    'lng': -79.38852348995324,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.650386666114635,
      'lng': -79.38852348995324}],
    'distance': 411,
    'postalCode': 'M5V 1Z4',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['241 Queen St. W (at St. Patrick St.)',
     'Toronto ON M5V 1Z4',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d123951735',
     'name': 'Smoke Shop',
     'pluralName': 'Smoke Shops',
     'shortName': 'Smoke Shop',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/tobacco_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []},
   'venuePage': {'id': '42625936'}},
  'referralId': 'e-0-4b7ed424f964a5208a0230e3-17'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '5aff06ca6e4650002cc6286b',
   'name': 'Rosalinda',
   'location': {'address': '133 Richmond St W',
    'lat': 43.650252483069295,
    'lng': -79.38515575620075,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.650252483069295,
      'lng': -79.38515575620075}],
    'distance': 444,
    'postalCode': 'M5H 2L2',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['133 Richmond St W', 'Toronto ON M5H 2L2', 'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1d3941735',
     'name': 'Vegetarian / Vegan Restaurant',
     'pluralName': 'Vegetarian / Vegan Restaurants',
     'shortName': 'Vegetarian / Vegan',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/vegetarian_',
```

```
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-5aff06ca6e4650002cc6286b-18'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4ad4c062f964a520baf720e3',
   'name': 'Canadian Opera Company',
   'location': {'address': '145 Queen Street West',
    'lat': 43.6506601893789,
    'lng': -79.38624169559013,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.6506601893789,
      'lng': -79.38624169559013}],
    'distance': 375,
    'postalCode': 'M5H 2N7',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['145 Queen Street West',
     'Toronto ON M5H 2N7',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d136941735',
     'name': 'Opera House',
     'pluralName': 'Opera Houses',
     'shortName': 'Opera House',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/arts_entertainment/perfor
mingarts_operahouse_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4ad4c062f964a520baf720e3-19'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '52a7ae41498eed3af4d0a3fa',
   'name': 'Yueh Tung Chinese Restaurant',
   'location': {'address': '126 Elizabeth St.',
    'crossStreet': 'Dundas St.',
    'lat': 43.655281263429195,
    'lng': -79.3853365267765,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.655281263429195,
      'lng': -79.3853365267765}],
    'distance': 210,
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['126 Elizabeth St. (Dundas St.)',
     'Toronto ON',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d145941735',
     'name': 'Chinese Restaurant',
     'pluralName': 'Chinese Restaurants',
     'shortName': 'Chinese',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/asian_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-52a7ae41498eed3af4d0a3fa-20'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4ae3398ff964a520ed9121e3',
   'name': 'Red Lobster',
   'location': {'address': '20 Dundas St W',
```

```
      'crossStreet': 'at Bay St',
      'lat': 43.656328,
      'lng': -79.383621,
      'labeledLatLngs': [{'label': 'display',
        'lat': 43.656328,
        'lng': -79.383621}],
      'distance': 390,
      'postalCode': 'M5G 2C2',
      'cc': 'CA',
      'city': 'Toronto',
      'state': 'ON',
      'country': 'Canada',
      'formattedAddress': ['20 Dundas St W (at Bay St)',
       'Toronto ON M5G 2C2',
       'Canada']},
     'categories': [{'id': '4bf58dd8d48988d1ce941735',
       'name': 'Seafood Restaurant',
       'pluralName': 'Seafood Restaurants',
       'shortName': 'Seafood',
       'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/seafood_',
        'suffix': '.png'},
       'primary': True}],
     'photos': {'count': 0, 'groups': []}},
    'referralId': 'e-0-4ae3398ff964a520ed9121e3-21'},
   {'reasons': {'count': 0,
     'items': [{'summary': 'This spot is popular',
       'type': 'general',
       'reasonName': 'globalInteractionReason'}]},
    'venue': {'id': '537773d1498e74a75bb75c1e',
     'name': 'Eggspectation Bell Trinity Square',
     'location': {'address': '483 Bay Street',
      'crossStreet': 'Albert Street',
      'lat': 43.65314383888587,
      'lng': -79.38198016678167,
      'labeledLatLngs': [{'label': 'display',
        'lat': 43.65314383888587,
        'lng': -79.38198016678167}],
      'distance': 430,
      'postalCode': 'M5G 2C9',
      'cc': 'CA',
      'city': 'Toronto',
      'state': 'ON',
      'country': 'Canada',
      'formattedAddress': ['483 Bay Street (Albert Street)',
       'Toronto ON M5G 2C9',
       'Canada']},
     'categories': [{'id': '4bf58dd8d48988d143941735',
       'name': 'Breakfast Spot',
       'pluralName': 'Breakfast Spots',
       'shortName': 'Breakfast',
       'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/breakfast_',
        'suffix': '.png'},
       'primary': True}],
     'photos': {'count': 0, 'groups': []},
     'venuePage': {'id': '97507838'}},
    'referralId': 'e-0-537773d1498e74a75bb75c1e-22'},
   {'reasons': {'count': 0,
     'items': [{'summary': 'This spot is popular',
       'type': 'general',
       'reasonName': 'globalInteractionReason'}]},
    'venue': {'id': '509bb871e4b09c7ac93f6642',
     'name': 'JaBistro',
     'location': {'address': '222 Richmond St W',
      'lat': 43.64968685893743,
      'lng': -79.38809001547467,
      'labeledLatLngs': [{'label': 'display',
        'lat': 43.64968685893743,
        'lng': -79.38809001547467}],
      'distance': 481,
      'postalCode': 'M5V 1W4',
      'cc': 'CA',
      'city': 'Toronto',
```

```
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['222 Richmond St W', 'Toronto ON M5V 1W4', 'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1d2941735',
     'name': 'Sushi Restaurant',
     'pluralName': 'Sushi Restaurants',
     'shortName': 'Sushi',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/sushi_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []},
   'venuePage': {'id': '148995303'}},
  'referralId': 'e-0-509bb871e4b09c7ac93f6642-23'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '57fe5f64498e08c9fc55cb87',
   'name': "Jimmy's Coffee",
   'location': {'address': '166 McCaul Street',
    'lat': 43.65582710322224,
    'lng': -79.39204216888555,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.65582710322224,
      'lng': -79.39204216888555}],
    'distance': 441,
    'postalCode': 'M5T 1J7',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['166 McCaul Street', 'Toronto ON M5T 1J7', 'Canada']},
   'categories': [{'id': '4bf58dd8d48988d16d941735',
     'name': 'Café',
     'pluralName': 'Cafés',
     'shortName': 'Café',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/cafe_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-57fe5f64498e08c9fc55cb87-24'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4ad69511f964a520e40721e3',
   'name': 'The Keg Steakhouse + Bar',
   'location': {'address': '165 York St',
    'crossStreet': 'btwn Richmond St. & Adelaide St.',
    'lat': 43.649937252985254,
    'lng': -79.38419604942506,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.649937252985254,
      'lng': -79.38419604942506}],
    'distance': 509,
    'postalCode': 'M5H 3R8',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['165 York St (btwn Richmond St. & Adelaide St.)',
     'Toronto ON M5H 3R8',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1cc941735',
     'name': 'Steakhouse',
     'pluralName': 'Steakhouses',
     'shortName': 'Steakhouse',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/steakhouse_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4ad69511f964a520e40721e3-25'},
```

```
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
    'reasonName': 'globalInteractionReason'}]},
 'venue': {'id': '5479da4f498e8569fb44985c',
  'name': 'MUJI',
  'location': {'address': '595 Bay St E',
   'crossStreet': 'at Dundas St W',
   'lat': 43.656024,
   'lng': -79.383284,
   'labeledLatLngs': [{'label': 'display',
     'lat': 43.656024,
     'lng': -79.383284}],
   'distance': 390,
   'cc': 'CA',
   'city': 'Toronto',
   'state': 'ON',
   'country': 'Canada',
   'formattedAddress': ['595 Bay St E (at Dundas St W)',
    'Toronto ON',
    'Canada']},
  'categories': [{'id': '4bf58dd8d48988d1ff941735',
    'name': 'Miscellaneous Shop',
    'pluralName': 'Miscellaneous Shops',
    'shortName': 'Shop',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/default_',
     'suffix': '.png'},
    'primary': True}],
  'photos': {'count': 0, 'groups': []}},
 'referralId': 'e-0-5479da4f498e8569fb44985c-26'},
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
    'reasonName': 'globalInteractionReason'}]},
 'venue': {'id': '4b8d5856f964a520f4f532e3',
  'name': 'Noodle King',
  'location': {'address': '123 Queen St. W.',
   'crossStreet': "in Toronto's PATH Walkway",
   'lat': 43.65170632794881,
   'lng': -79.38304628912547,
   'labeledLatLngs': [{'label': 'display',
     'lat': 43.65170632794881,
     'lng': -79.38304628912547}],
   'distance': 418,
   'postalCode': 'M5H 3M9',
   'cc': 'CA',
   'city': 'Toronto',
   'state': 'ON',
   'country': 'Canada',
   'formattedAddress': ["123 Queen St. W. (in Toronto's PATH Walkway)",
    'Toronto ON M5H 3M9',
    'Canada']},
  'categories': [{'id': '4bf58dd8d48988d142941735',
    'name': 'Asian Restaurant',
    'pluralName': 'Asian Restaurants',
    'shortName': 'Asian',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/asian_',
     'suffix': '.png'},
    'primary': True}],
  'photos': {'count': 0, 'groups': []}},
 'referralId': 'e-0-4b8d5856f964a520f4f532e3-27'},
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
    'reasonName': 'globalInteractionReason'}]},
 'venue': {'id': '537d4d6d498ec171ba22e7fe',
  'name': "Jimmy's Coffee",
  'location': {'address': '82 Gerrard Street W',
   'crossStreet': 'Gerrard & LaPlante',
   'lat': 43.65842123574496,
   'lng': -79.38561319551111,
   'labeledLatLngs': [{'label': 'display',
```

```
      'lat': 43.65842123574496,
      'lng': -79.38561319551111}],
    'distance': 512,
    'postalCode': 'M5G 1Z4',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['82 Gerrard Street W (Gerrard & LaPlante)',
     'Toronto ON M5G 1Z4',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1e0931735',
     'name': 'Coffee Shop',
     'pluralName': 'Coffee Shops',
     'shortName': 'Coffee Shop',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/coffeeshop_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-537d4d6d498ec171ba22e7fe-28'},
 {'reasons': {'count': 0,
   'items': [{'summary': 'This spot is popular',
     'type': 'general',
     'reasonName': 'globalInteractionReason'}]},
  'venue': {'id': '4e31b74252b131dcebb08743',
   'name': 'Shangri-La Toronto',
   'location': {'address': '188 University Ave.',
    'crossStreet': 'at Adelaide St. W',
    'lat': 43.64912919417502,
    'lng': -79.3865566853963,
    'labeledLatLngs': [{'label': 'display',
      'lat': 43.64912919417502,
      'lng': -79.3865566853963}],
    'distance': 540,
    'postalCode': 'M5H 0A3',
    'cc': 'CA',
    'city': 'Toronto',
    'state': 'ON',
    'country': 'Canada',
    'formattedAddress': ['188 University Ave. (at Adelaide St. W)',
     'Toronto ON M5H 0A3',
     'Canada']},
   'categories': [{'id': '4bf58dd8d48988d1fa931735',
     'name': 'Hotel',
     'pluralName': 'Hotels',
     'shortName': 'Hotel',
     'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/travel/hotel_',
      'suffix': '.png'},
     'primary': True}],
   'photos': {'count': 0, 'groups': []}},
  'referralId': 'e-0-4e31b74252b131dcebb08743-29'}]
```


Now we can explore the nearby venues!

In [39]:

```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

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

```
            LIMIT)

        # make the GET request
        results_1 = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results_1])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_l
ist])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [40]:

```
LIMIT = 100
toronto_venues = getNearbyVenues(names=toronto_DF['Neighborhood'],
                                   latitudes=toronto_DF['Latitude'],
                                   longitudes=toronto_DF['Longitude']
                                  )
```

```
Lawrence Park
Davisville North
North Toronto West
Davisville
Moore Park, Summerhill East
Deer Park, Forest Hill SE, Rathnelly, South Hill, Summerhill West
Roselawn
Forest Hill North, Forest Hill West
The Annex, North Midtown, Yorkville
Rosedale
Cabbagetown, St. James Town
Church and Wellesley
Harbourfront
Ryerson, Garden District
St. James Town
Berczy Park
Central Bay Street
Adelaide, King, Richmond
Harbourfront East, Toronto Islands, Union Station
Design Exchange, Toronto Dominion Centre
Commerce Court, Victoria Hotel
Harbord, University of Toronto
Chinatown, Grange Park, Kensington Market
CN Tower, Bathurst Quay, Island airport, Harbourfront West, King and Spadina, Railway Lan
ds, South Niagara
Stn A PO Boxes 25 The Esplanade
First Canadian Place, Underground city
Christie
Queen's Park
The Beaches
The Danforth West, Riverdale
The Beaches West, India Bazaar
Studio District
Business Reply Mail Processing Centre 969 Eastern
Woodbine Gardens, Parkview Hill
Woodbine Heights
Leaside
```

```
Thorncliffe Park
East Toronto
Humber Bay Shores, Mimico South, New Toronto
Alderwood, Long Branch
The Kingsway, Montgomery Road, Old Mill North
Humber Bay, King's Mill Park, Kingsway Park South East, Mimico NE, Old Mill South, The Qu
eensway East, Royal York South East, Sunnylea
Kingsway Park South West, Mimico NW, The Queensway West, Royal York South West, South of
Bloor
Cloverdale, Islington, Martin Grove, Princess Gardens, West Deane Park
Bloordale Gardens, Eringate, Markland Wood, Old Burnhamthorpe
Westmount
Kingsview Village, Martin Grove Gardens, Richview Gardens, St. Phillips
Albion Gardens, Beaumond Heights, Humbergate, Jamestown, Mount Olive, Silverstone, South
Steeles, Thistletown
Northwest
Canada Post Gateway Processing Centre
Hillcrest Village
Fairview, Henry Farm, Oriole
Bayview Village
Silver Hills, York Mills
Newtonbrook, Willowdale
Willowdale South
York Mills West
Willowdale West
Parkwoods
Don Mills North
Flemingdon Park, Don Mills South
Bathurst Manor, Downsview North, Wilson Heights
Northwood Park, York University
CFB Toronto, Downsview East
Downsview West
Downsview Central
Downsview Northwest
Victoria Village
Bedford Park, Lawrence Manor East
Lawrence Heights, Lawrence Manor
Glencairn
Downsview, North Park, Upwood Park
Humber Summit
Emery, Humberlea
Queen's Park
Rouge, Malvern
Highland Creek, Rouge Hill, Port Union
Guildwood, Morningside, West Hill
Woburn
Cedarbrae
Scarborough Village
East Birchmount Park, Ionview, Kennedy Park
Clairlea, Golden Mile, Oakridge
Cliffcrest, Cliffside, Scarborough Village West
Birch Cliff, Cliffside West
Dorset Park, Scarborough Town Centre, Wexford Heights
Maryvale, Wexford
Agincourt
Clarks Corners, Sullivan, Tam O'Shanter
Agincourt North, L'Amoreaux East, Milliken, Steeles East
L'Amoreaux West
Upper Rouge
Dovercourt Village, Dufferin
Little Portugal, Trinity
Brockton, Exhibition Place, Parkdale Village
High Park, The Junction South
Parkdale, Roncesvalles
Runnymede, Swansea
Humewood-Cedarvale
Caledonia-Fairbanks
Del Ray, Keelesdale, Mount Dennis, Silverthorn
The Junction North, Runnymede
Weston
```

In [41]:

In [41]:

```
toronto_venues.head(10)
```

Out[41]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Lawrence Park	43.728020	-79.388790	Lawrence Park Ravine	43.726963	-79.394382	Park
1	Lawrence Park	43.728020	-79.388790	Dim Sum Deluxe	43.726953	-79.394260	Dim Sum Restaurant
2	Lawrence Park	43.728020	-79.388790	Zodiac Swim School	43.728532	-79.382860	Swim School
3	Lawrence Park	43.728020	-79.388790	TTC Bus #162 - Lawrence-Donway	43.728026	-79.382805	Bus Line
4	Davisville North	43.712751	-79.390197	Summerhill Market North	43.715499	-79.392881	Food & Drink Shop
5	Davisville North	43.712751	-79.390197	Sherwood Park	43.716551	-79.387776	Park
6	Davisville North	43.712751	-79.390197	Homeway Restaurant & Brunch	43.712641	-79.391557	Breakfast Spot
7	Davisville North	43.712751	-79.390197	Winners	43.713236	-79.393873	Department Store
8	Davisville North	43.712751	-79.390197	Best Western Roehampton Hotel & Suites	43.708878	-79.390880	Hotel
9	Davisville North	43.712751	-79.390197	Subway	43.708474	-79.390674	Sandwich Place

In [42]:

```
toronto_venues.groupby('Neighborhood').count()
```

Out[42]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Adelaide, King, Richmond	100	100	100	100	100	100
Agincourt	4	4	4	4	4	4
Agincourt North, L'Amoreaux East, Milliken, Steeles East	2	2	2	2	2	2
Albion Gardens, Beaumond Heights, Humbergate, Jamestown, Mount Olive, Silverstone, South Steeles, Thistletown	11	11	11	11	11	11
Alderwood, Long Branch	9	9	9	9	9	9
...	...	...	...	...	...	...
Willowdale West	6	6	6	6	6	6
Woburn	3	3	3	3	3	3
Woodbine Gardens, Parkview Hill	13	13	13	13	13	13
Woodbine Heights	9	9	9	9	9	9
York Mills West	3	3	3	3	3	3

100 rows × 6 columns

In [43]:

```
print('There are {} uniques categories.'.format(len(toronto_venues['Venue Category'].unique())))
```

```
There are 271 uniques categories.
```

There are 274 unique categories in which Chinese Restaurant is one of them. We will do one hot encoding for getting dummies of venue category. So that we will calculate mean of all venue groupby there neighborhoods.

In [44]:

```
toronto_onehot = pd.get_dummies(toronto_venues[['Venue Category']], prefix="", prefix_se
p="")

toronto_onehot['Neighborhood'] = toronto_venues['Neighborhood']

fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]
toronto_grouped = toronto_onehot.groupby('Neighborhood').mean().reset_index()
toronto_grouped
```

Out[44]:

	Neighborhood	Yoga Studio	Accessories Store	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	...	Train Station	Vegetaria / Vega Restaura
0	Adelaide, King, Richmond	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
1	Agincourt	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
2	Agincourt North, L'Amoreaux East, Milliken, St...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
3	Albion Gardens, Beaumond Heights, Humbergate, ...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
4	Alderwood, Long Branch	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	
95	Willowdale West	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
96	Woburn	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
97	Woodbine Gardens, Parkview Hill	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
98	Woodbine Heights	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.
99	York Mills West	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.

100 rows × 271 columns

In [45]:

```
print (toronto_venues['Venue Category'].value_counts())
```

```
Coffee Shop            193
Café                    96
Restaurant              59
Pizza Place             51
Park                    50
                      ...
Nail Salon               1
```

```
Flea Market              1
Optical Shop             1
Empanada Restaurant      1
Warehouse Store          1
Name: Venue Category, Length: 271, dtype: int64
```

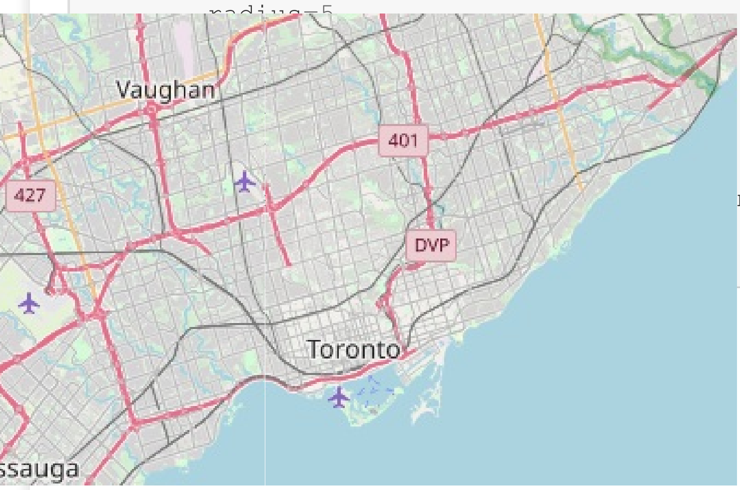
3. Exploratory Data Analysis

3.1 Folium Library and Leaflet Map

Folium is a python library, I'm using it to draw an interactive leaflet map using coordinate data.

In [46]:

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

# add markers to map
for lat, lng, borough, neighborhood in zip(toronto_DF['Latitude'], toronto_DF['Longitude
'], toronto_DF['Borough'], toronto_DF['Neighborhood']):
    label = '{},{}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
```




```
nto)
```

3.2 Relationship between neighborhood and Chinese Restaurant

First we will extract the Neighborhood and Chinese Restaurant column from the above toronto dataframe for further analysis:

In [50]:

```
toronto_part = toronto_grouped[['Neighborhood', 'Chinese Restaurant']]
toronto_part
```

Out[50]:

	Neighborhood	Chinese Restaurant
0	Adelaide, King, Richmond	0.0
1	Agincourt	0.0
2	Agincourt North, L'Amoreaux East, Milliken, St...	0.0
3	Albion Gardens, Beaumond Heights, Humbergate, ...	0.0
4	Alderwood, Long Branch	0.0
...	...	...
95	Willowdale West	0.0
96	Woburn	0.0
97	Woodbine Gardens, Parkview Hill	0.0
98	Woodbine Heights	0.0
99	York Mills West	0.0

100 rows × 2 columns

In [51]:

```
toronto_merged = pd.merge(toronto_DF, toronto_part, on='Neighborhood')
toronto_merged
```

Out[51]:

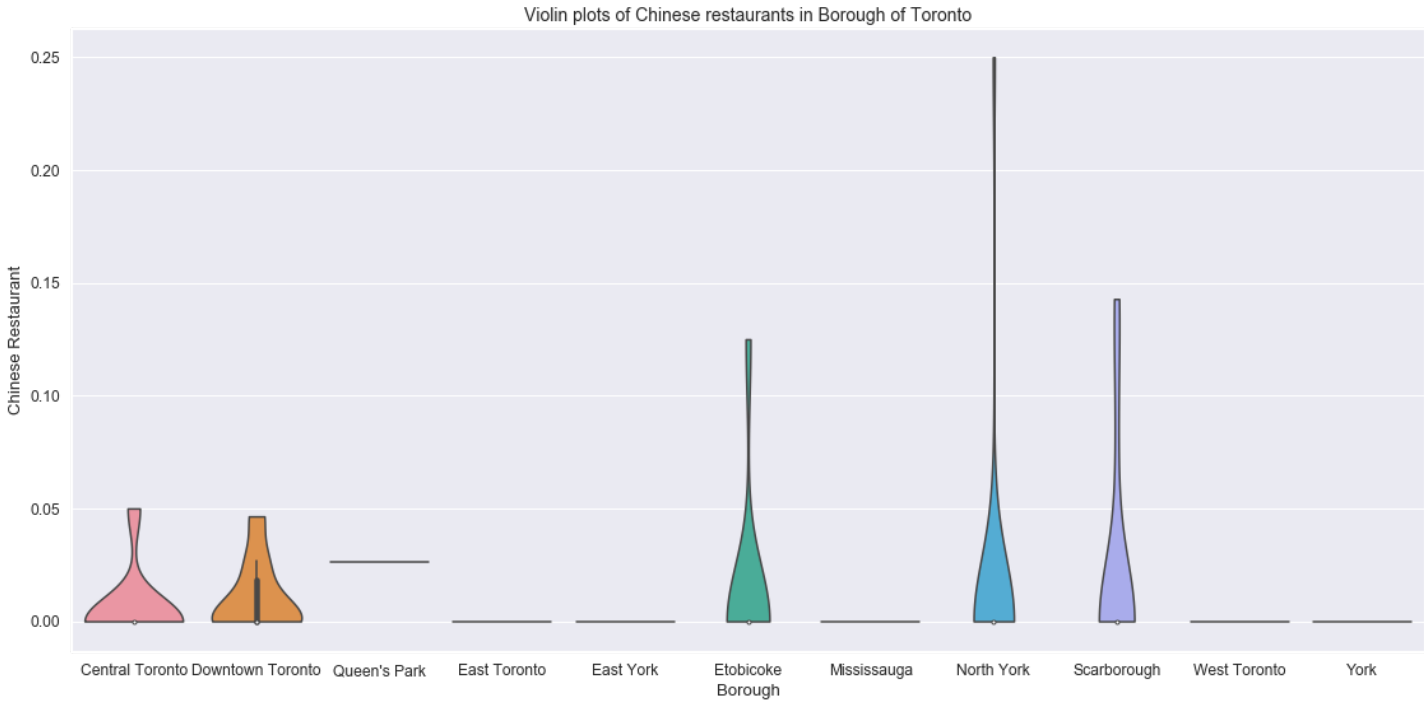
	Borough	Postalcode	Neighborhood	Latitude	Longitude	Chinese Restaurant
0	Central Toronto	M4N	Lawrence Park	43.728020	-79.388790	0.00
1	Central Toronto	M4P	Davisville North	43.712751	-79.390197	0.00
2	Central Toronto	M4R	North Toronto West	43.715383	-79.405678	0.05
3	Central Toronto	M4S	Davisville	43.704324	-79.388790	0.00
4	Central Toronto	M4T	Moore Park, Summerhill East	43.689574	-79.383160	0.00
...	...	...	...	...	...	...
96	York	M6C	Humewood-Cedarvale	43.693781	-79.428191	0.00
97	York	M6E	Caledonia-Fairbanks	43.689026	-79.453512	0.00
98	York	M6M	Del Ray, Keelesdale, Mount Dennis, Silverthorn	43.691116	-79.476013	0.00
99	York	M6N	The Junction North, Runnymede	43.673185	-79.487262	0.00
100	York	M9N	Weston	43.706876	-79.518188	0.00

101 rows × 6 columns

In [52]:

```
# Let's try Categorical plot

%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

```
fig = plt.figure(figsize=(19,9))

sns.set(font_scale=1.1)
sns.violinplot(y="Chinese Restaurant", x="Borough", data=toronto_merged, cut=0);

plt.title('Violin plots of Chinese restaurants in Borough of Toronto', fontsize=14)
plt.show()
```

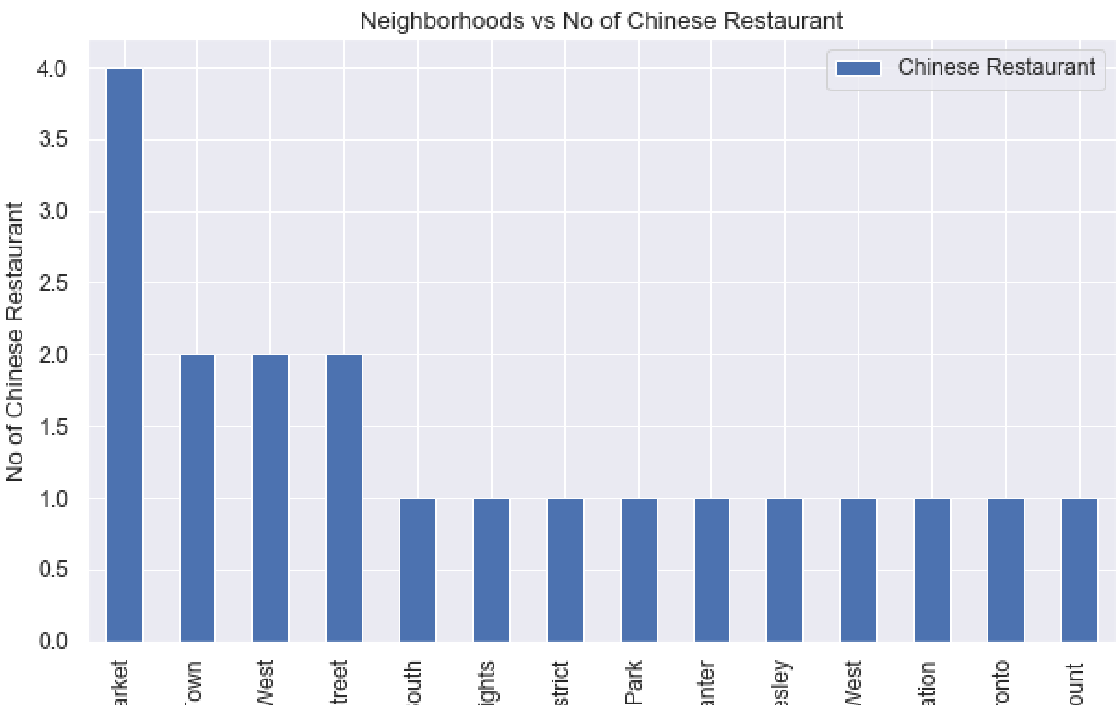


This plot helps in identifying the boroughs with densely populated Chinese restaurants.

Lets visualize the neighborhood with Chinese Restaurants

In [53]:

```
graph = pd.DataFrame(toronto_onehot.groupby('Neighborhood')['Chinese Restaurant'].sum())
graph = graph.sort_values(by ='Chinese Restaurant', ascending=False)
graph.iloc[:14].plot(kind='bar', figsize=(10,6))
plt.xlabel("Neighborhoods")
plt.ylabel("No of Chinese Restaurant")
plt.title("Neighborhoods vs No of Chinese Restaurant")
plt.show()
```



3.3 Relationship between neighborhood and Chinese popultion

In [54]:

```
#Merge all the population table with the ethnic percentage by neighborhood
ET = ETY_population_df.append(TEY_population_df,sort=True).reset_index()
ET.drop('index',axis=1,inplace=True)
SN = North_population_df.append(Scar_population_df,sort=True).reset_index()
SN.drop('index',axis=1,inplace=True)
pop_ethnic_df = SN.append(ET,sort=True).reset_index()
pop_ethnic_df.drop('index',axis=1,inplace=True)
pop_ethnic_df = pop_ethnic_df[['Riding', 'Population','Ethnic Origin #1', 'Ethnic Origin
1 in %','Ethnic Origin #2', 'Ethnic Origin 2 in %',
                              'Ethnic Origin #3','Ethnic Origin 3 in %','Ethnic Origin
#4', 'Ethnic Origin 4 in %','Ethnic Origin #5','Ethnic Origin 5 in %',
                              'Ethnic Origin #6','Ethnic Origin 6 in %','Ethnic Origin
#7', 'Ethnic Origin 7 in %','Ethnic Origin #8', 'Ethnic Origin 8 in %',
                              'Ethnic Origin #9','Ethnic Origin 9 in %',
                             ]]
pop_ethnic_df
```

Out[54]:

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %
0	Willowdale	117405	Chinese	25.9	Iranian	12.1	Korean	10.6	NaN	NaN	NaN	NaN
1	Eglinton-Lawrence	112925	Canadian	14.7	English	12.6	Polish	12.0	Filipino	11.0	Scottish	9.7
2	Don Valley North	109060	Chinese	32.4	East Indian	7.3	Iranian	7.3	NaN	NaN	NaN	NaN
3	Humber River-Black Creek	107725	Italian	12.8	East Indian	9.2	Jamaican	8.5	Vietnamese	8.0	Canadian	7.4
4	York Centre	103760	Filipino	17.0	Italian	13.4	Russian	9.5	Canadian	8.6	NaN	NaN
5	Don Valley West	101790	English	19.2	Canadian	15.1	Scottish	14.9	Irish	14.2	Chinese	11.2
6	Don Valley East	93170	East Indian	10.6	Canadian	10.4	English	10.1	Chinese	8.9	Irish	8.1
7	Scarborough Centre	110450	Filipino	13.1	East Indian	12.2	Canadian	11.2	Chinese	10.7	English	7.8
8	Scarborough Southwest	108295	Canadian	16.2	English	14.3	Irish	11.5	Scottish	10.9	Filipino	9.5
9	Scarborough-Agincourt	104225	Chinese	47.0	East Indian	7.4	NaN	NaN	NaN	NaN	NaN	NaN

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %
10	Scarborough-Rouge Park	101445	East Indian	16.7	Canadian	11.8	Sri Lankan	11.1	English	9.8	Filipino	9.3
11	Scarborough-Guildwood	101115	East Indian	18.0	Canadian	11.6	English	9.7	Filipino	8.5	Sri Lankan	7.8
12	Scarborough North	97610	Chinese	46.6	East Indian	11.8	Sri Lankan	9.4	NaN	NaN	NaN	NaN
13	Etobicoke-Lakeshore	127520	English	17.1	Canadian	15.9	Irish	14.4	Scottish	13.5	Polish	9.2
14	Etobicoke North	116960	East Indian	22.2	Canadian	7.9	NaN	NaN	NaN	NaN	NaN	NaN
15	Etobicoke Centre	116055	Italian	15.1	English	14.3	Canadian	12.1	Irish	10.8	Scottish	10.4
16	York South-Weston	115130	Portuguese	14.5	Italian	12.8	Canadian	8.7	Jamaican	8.4	NaN	NaN
17	Spadina-Fort York	114315	English	16.4	Chinese	16.0	Irish	14.6	Canadian	14.0	Scottish	13.2
18	Beaches-East York	108435	English	24.2	Irish	19.9	Canadian	19.7	Scottish	18.9	French	8.7
19	Davenport	107395	Portuguese	22.7	English	13.6	Canadian	12.8	Irish	11.5	Italian	11.1
20	Parkdale-High Park	106445	English	22.3	Irish	20.0	Scottish	18.7	Canadian	16.1	German	9.8
21	Toronto-Danforth	105395	English	22.9	Irish	19.5	Scottish	18.7	Canadian	18.4	Chinese	13.8
22	Toronto-St. Paul's	104940	English	18.5	Canadian	16.1	Irish	15.2	Scottish	14.8	Polish	10.3
23	University-Rosedale	100520	English	20.6	Irish	16.6	Scottish	16.3	Canadian	15.2	Chinese	14.7
24	Toronto Centre	99590	English	15.7	Canadian	13.7	Irish	13.4	Scottish	12.6	Chinese	12.5

From the above dataframe we can pickout the neighborhoods with highest Chinese population percentage by using the below given method.

In [55]:

```
#Filtering the riding with Chinese ethnic crowd
temp = pop_ethnic_df.loc[(pop_ethnic_df['Ethnic Origin #1'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #2'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #3'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #4'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #5'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #6'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #7'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #8'] == 'Chinese')|
                          (pop_ethnic_df['Ethnic Origin #9'] == 'Chinese')]
pop_chinese_df = pd.DataFrame(temp).reset_index()
pop_chinese_df.drop('index',axis=1,inplace=True)

pop_chinese_df
```

Out[55]:

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %	Et O
0	Willowdale	117405	Chinese	25.9	Iranian	12.1	Korean	10.6	NaN	NaN	NaN	NaN	
1	Don Valley North	109060	Chinese	32.4	East Indian	7.3	Iranian	7.3	NaN	NaN	NaN	NaN	
2	Don Valley West	101790	English	19.2	Canadian	15.1	Scottish	14.9	Irish	14.2	Chinese	11.2	

	Riding	Population	Ethnic Origin #1	Ethnic Origin 1 in %	Ethnic Origin #2	Ethnic Origin 2 in %	Ethnic Origin #3	Ethnic Origin 3 in %	Ethnic Origin #4	Ethnic Origin 4 in %	Ethnic Origin #5	Ethnic Origin 5 in %	Et
3	Don Valley East	93170	East Indian	10.6	Canadian	10.4	English	10.1	Chinese	8.9	Irish	8.1	Sco
4	Scarborough Centre	110450	Filipino	13.1	East Indian	12.2	Canadian	11.2	Chinese	10.7	English	7.8	La
5	Scarborough Southwest	108295	Canadian	16.2	English	14.3	Irish	11.5	Scottish	10.9	Filipino	9.5	In
6	Scarborough-Agincourt	104225	Chinese	47.0	East Indian	7.4	NaN	NaN	NaN	NaN	NaN	NaN	
7	Scarborough-Guildwood	101115	East Indian	18.0	Canadian	11.6	English	9.7	Filipino	8.5	Sri Lankan	7.8	Chi
8	Scarborough North	97610	Chinese	46.6	East Indian	11.8	Sri Lankan	9.4	NaN	NaN	NaN	NaN	
9	Spadina-Fort York	114315	English	16.4	Chinese	16.0	Irish	14.6	Canadian	14.0	Scottish	13.2	Fr
10	Toronto-Danforth	105395	English	22.9	Irish	19.5	Scottish	18.7	Canadian	18.4	Chinese	13.8	Fr
11	University-Rosedale	100520	English	20.6	Irish	16.6	Scottish	16.3	Canadian	15.2	Chinese	14.7	Ger
12	Toronto Centre	99590	English	15.7	Canadian	13.7	Irish	13.4	Scottish	12.6	Chinese	12.5	Fr

In [68]:

```python
#retaining only Chinese ethnic percentage & the neighborhood name
columns_list = pop_chinese_df.columns.to_list()
pop_chinese_DF_with_percent = pd.DataFrame()
#removing Riding & Population from the column names list
del columns_list[0]
del columns_list[0]


for i in range(0,pop_chinese_df.shape[0]):
    for j in columns_list:
        print(j)
        if pop_chinese_df.at[i, j] == 'Chinese':
            k = columns_list.index(j) + 1
            percent_col = columns_list[k]
            pop_chinese_DF_with_percent = pop_chinese_DF_with_percent.append({'Riding':p
op_chinese_df.at[i, 'Riding'], 'Population':pop_chinese_df.at[i, 'Population']
                                                                                , 'Ethnicity
': pop_chinese_df.at[i, j], 'Percentage': pop_chinese_df.at[i, percent_col]},ignore_inde
x=True)

pop_chinese_DF_with_percent
```

```
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
```

Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2

```
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
```

```
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
Ethnic Origin #1
Ethnic Origin 1 in %
Ethnic Origin #2
Ethnic Origin 2 in %
Ethnic Origin #3
Ethnic Origin 3 in %
Ethnic Origin #4
Ethnic Origin 4 in %
Ethnic Origin #5
Ethnic Origin 5 in %
Ethnic Origin #6
Ethnic Origin 6 in %
Ethnic Origin #7
Ethnic Origin 7 in %
Ethnic Origin #8
Ethnic Origin 8 in %
Ethnic Origin #9
Ethnic Origin 9 in %
```

Out[68]:

	Ethnicity	Percentage	Population	Riding
0	Chinese	25.9	117405.0	Willowdale
1	Chinese	32.4	109060.0	Don Valley North
2	Chinese	11.2	101790.0	Don Valley West
3	Chinese	8.9	93170.0	Don Valley East
4	Chinese	10.7	110450.0	Scarborough Centre
5	Chinese	7.2	108295.0	Scarborough Southwest
6	Chinese	47.0	104225.0	Scarborough-Agincourt
7	Chinese	7.1	101115.0	Scarborough-Guildwood
8	Chinese	46.6	97610.0	Scarborough North
9	Chinese	16.0	114315.0	Spadina-Fort York
10	Chinese	13.8	105395.0	Toronto-Danforth
11	Chinese	14.7	100520.0	University-Rosedale
12	Chinese	12.5	99590.0	Toronto Centre

In [59]:

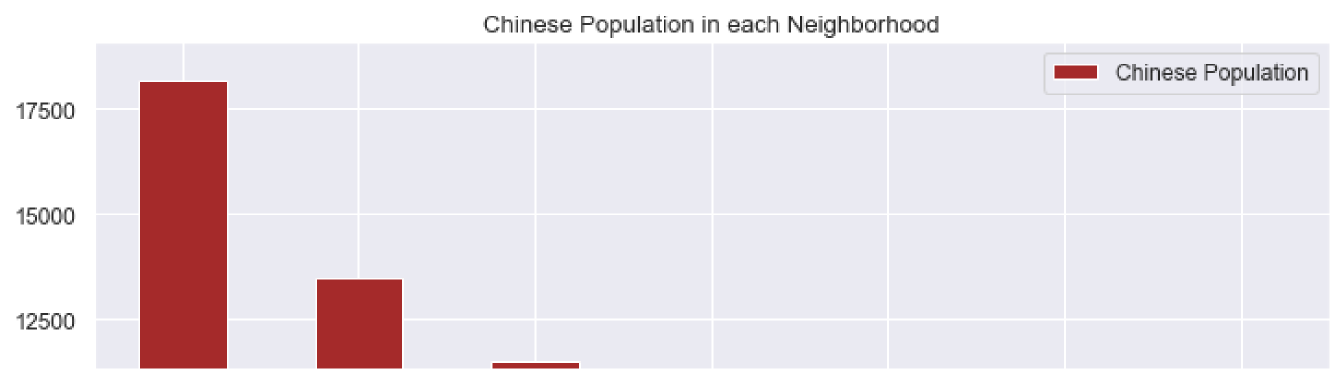
```
pop_chinese_DF_with_percent['Chinese Population'] = (pop_chinese_DF_with_percent['Percent
age'] * pop_chinese_DF_with_percent['Population'])/100
pop_chinese_DF_with_percent.drop(columns={'Percentage','Population','Ethnicity'},axis=1,
inplace =True)
pop_chinese_DF_with_percent.drop_duplicates(keep='first',inplace=True)
pop_chinese_DF_with_percent
```

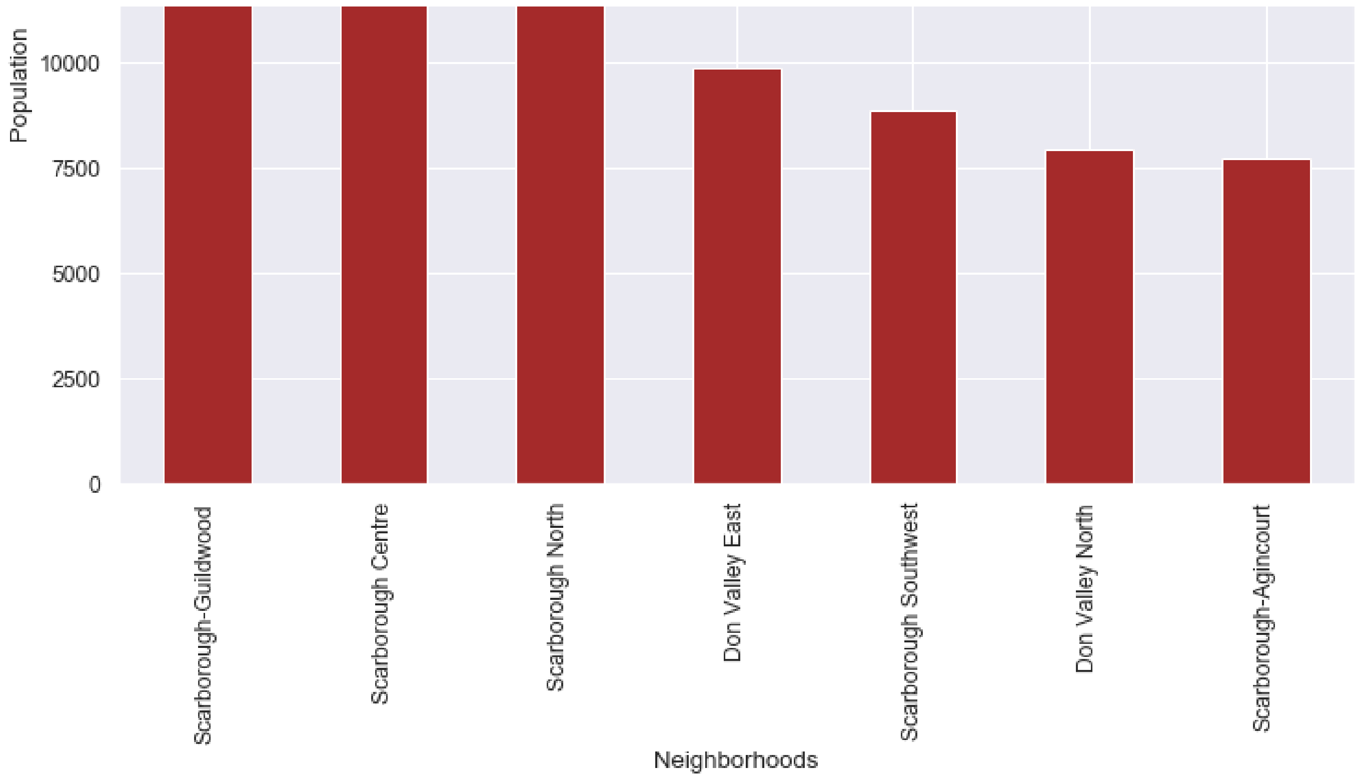
Out[59]:

	Riding	Chinese Population
0	Don Valley North	7961.38
1	Don Valley East	9876.02
2	Scarborough Centre	13474.90
3	Scarborough Southwest	8880.19
4	Scarborough-Agincourt	7712.65
5	Scarborough-Guildwood	18200.70
6	Scarborough North	11517.98

In [60]:

```
bar_graph = pop_chinese_DF_with_percent.sort_values(by='Chinese Population', ascending=F
alse)
bar_graph.plot(kind='bar',x='Riding', y='Chinese Population',figsize=(12,8), color='brow
n')
plt.title("Chinese Population in each Neighborhood")
plt.xlabel("Neighborhoods")
plt.ylabel("Population")
plt.show()
```





This analysis & visualization of the relationship between neighborhoods & chinese population present in those neighborhoods helps us in identifying the highly populated chinese neighborhoods. Once we identify those neighborhoods it helps us in deciding where to place the new chinese restaurant. chinese restaurant placed in an densely populated chinese neighborhood is more likely to get more chinese customers than a restaurant placed in a neighborhood with less or no chinese population. Thus this analysis helps in the determining the success of the new chinese restaurant.

3.4 Relationship between chinese poplation and chinese restaurant

First get the list of neighborhoods present in the riding using the wikipedia geography section for each riding. Altering the riding names to match the wikipedia page so we can retrieve the neighborhoods present in those ridings

In [61]:

```
#Altering the list to match the wikipedia page so we can retrieve the neighborhoods prese
nt in those Ridings
riding_list = pop_chinese_DF_with_percent['Riding'].to_list()
riding_list[riding_list.index('Scarborough Centre')] = 'Scarborough Centre (electoral di
strict)'
riding_list[riding_list.index('Scarborough North')] = 'Scarborough North (electoral dist
rict)'
riding_list
```

Out[61]:

```
['Don Valley North',
 'Don Valley East',
 'Scarborough Centre (electoral district)',
 'Scarborough Southwest',
 'Scarborough-Agincourt',
 'Scarborough-Guildwood',
 'Scarborough North (electoral district)']
```

In [62]:

```
#Scraping wiki page to get the neighborhoods of ech Ridings
import wikipedia

Riding_neighborhood_df = pd.DataFrame()
```

```
for item in riding_list:
    section = wikipedia.WikipediaPage(item).section('Geography')
    start = section.index('neighbourhoods of') + 17
    stop = section.index('.',start)
    Riding_neighborhood_df = Riding_neighborhood_df.append({'Riding':item, 'Neighborhood
s':section[start:stop]},ignore_index=True)


Riding_neighborhood_df = Riding_neighborhood_df[['Riding','Neighborhoods']]
Riding_neighborhood_df
```

Out[62]:

	Riding	Neighborhoods
0	Don Valley North	Henry Farm, Bayview Village, Bayview Woods-St...
1	Don Valley East	Flemingdon Park, Don Mills, Graydon Hall, Par...
2	Scarborough Centre (electoral district)	Scarborough City Centre (west of McCowan Road...
3	Scarborough Southwest	Birch Cliff, Oakridge, Cliffside, Kennedy Par...
4	Scarborough-Agincourt	Steeles, L'Amoreaux, Tam O'Shanter-Sullivan, ...
5	Scarborough-Guildwood	Guildwood, West Hill (west of Morningside Ave...
6	Scarborough North (electoral district)	Agincourt (east of Midland Avenue), Milliken ...

In [63]:

```
#Merging the pop_chinese_DF_with_percent dataframe containing population information with
the Riding_neighborhood_df dataframe.

Neigh_pop = pd.merge(pop_chinese_DF_with_percent, Riding_neighborhood_df, on='Riding')

Neigh_pop.drop(columns=['Riding'],inplace =True)
Neigh_pop
```

Out[63]:

	Chinese Population	Neighborhoods
0	7961.38	Henry Farm, Bayview Village, Bayview Woods-St...
1	9876.02	Flemingdon Park, Don Mills, Graydon Hall, Par...
2	8880.19	Birch Cliff, Oakridge, Cliffside, Kennedy Par...
3	7712.65	Steeles, L'Amoreaux, Tam O'Shanter-Sullivan, ...
4	18200.70	Guildwood, West Hill (west of Morningside Ave...

In [64]:

```
Neigh_pop['split_neighborhoods'] = Neigh_pop['Neighborhoods'].str.split(',')
Neigh_pop.drop(columns=['Neighborhoods'],inplace=True,axis=1)
Neigh_pop = Neigh_pop.split_neighborhoods.apply(pd.Series).merge(Neigh_pop, left_index =
True, right_index = True).drop(["split_neighborhoods"], axis = 1)\
                    .melt(id_vars = ['Chinese Population'], value_name = "Neighborhood")
.drop("variable", axis = 1).dropna()

Neigh_pop.reset_index()
Neigh_pop
```

Out[64]:

	Chinese Population	Neighborhood
0	7961.38	Henry Farm
1	9876.02	Flemingdon Park
2	8880.19	Birch Cliff
3	7712.65	Steeles

	Chinese Population	Neighborhood
4	18200.70	Guildwood
5	7961.38	Bayview Village
6	9876.02	Don Mills
7	8880.19	Oakridge
8	7712.65	L'Amoreaux
9	18200.70	West Hill (west of Morningside Avenue)
10	7961.38	Bayview Woods-Steeles
11	9876.02	Graydon Hall
12	8880.19	Cliffside
13	7712.65	Tam O'Shanter-Sullivan
14	18200.70	Morningside
15	7961.38	Hillcrest Village
16	9876.02	Parkwoods and Victoria Village
17	8880.19	Kennedy Park
18	7712.65	Agincourt (west of Midland Avenue) and Millik...
19	18200.70	Woburn
20	7961.38	Don Valley Village
22	8880.19	Clairlea
24	18200.70	and Scarborough Village (east of Markham Road)
25	7961.38	and Pleasant View
27	8880.19	Cliffcrest and parts of Scarborough Village a...

In [66]:

```
toronto_part['split_neighborhoods'] = toronto_part['Neighborhood'].str.split(',')
toronto_part.drop(columns=['Neighborhood'],inplace=True,axis=1)
toronto_part = toronto_part.split_neighborhoods.apply(pd.Series).merge(toronto_part, lef
t_index = True, right_index = True).drop(["split_neighborhoods"], axis = 1)\
                    .melt(id_vars = ['Chinese Restaurant'], value_name = "Neighborhood")
.drop("variable", axis = 1).dropna()

toronto_part.reset_index()
toronto_part
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
/opt/anaconda3/lib/python3.7/site-packages/pandas/core/frame.py:4102: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

Out[66]:

	Chinese Restaurant	Neighborhood
0	0.0	Adelaide
1	0.0	Agincourt
2	0.0	Agincourt North
3	0.0	Albion Gardens

	Chinese Restaurant	Neighborhood
3	0.0	Albion Gardens
4	0.0	Alderwood
...	...	...
603	0.0	South Steeles
614	0.0	South Niagara
655	0.0	Royal York South East
703	0.0	Thistletown
755	0.0	Sunnylea

203 rows × 2 columns

In [67]:

```
pop_merged_restaurant_percent = pd.merge(Neigh_pop, toronto_part, on='Neighborhood')
pop_merged_restaurant_percent.head()
```

Out[67]:

	Chinese Population	Neighborhood	Chinese Restaurant
0	7961.38	Henry Farm	0.0
1	8880.19	Oakridge	0.0
2	8880.19	Cliffside	0.0
3	18200.70	Morningside	0.0
4	8880.19	Kennedy Park	0.0

After performing the data cleaning & data analysis we can identify that their no big relationship established in terms of the Chinese population & the popular Chinese restaurants.

Thus this marks end of the data cleaning & analyses step in this project. Next we will look into the predictive modeling. In the predictive modelling we are going to use Clustering techniques since this is analysis of unlabelled data. K-Means clustering is used to perform the analysis of the data at hand.

4. Predictive Modeling

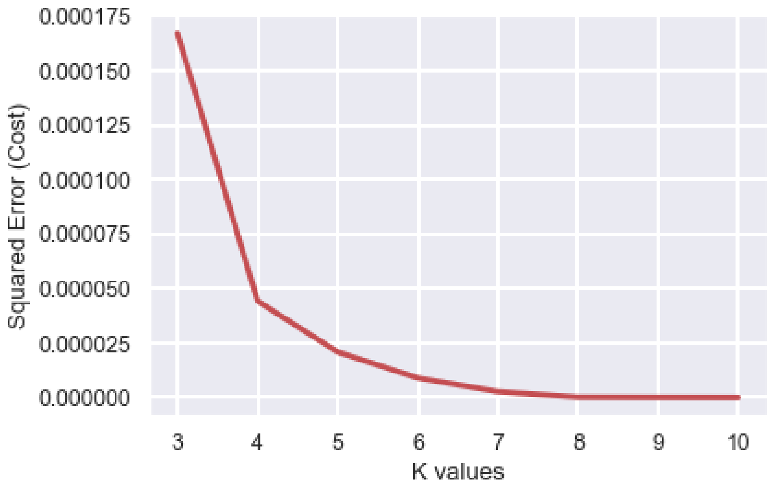
4.1 Clustering Neighborhoods of Toronto:

First step in K-means clustering is to identify best K value meaning the number of clusters in a given dataset. To do so we are going to use the elbow method on the Toronto dataset with Chinese restaurant percentage (i.e. toronto_merged dataframe).

In [69]:

```
from sklearn.cluster import KMeans

toronto_part_clustering = toronto_part.drop('Neighborhood', 1)


error_cost = []

for i in range(3,11):
    KM = KMeans(n_clusters = i, max_iter = 100)
    try:
        KM.fit(toronto_part_clustering)
    except ValueError:
        print("error on line",i)
```

```
    #calculate squared error for the clustered points
    error_cost.append(KM.inertia_/100)

#plot the K values aganist the squared error cost
plt.plot(range(3,11), error_cost, color='r', linewidth='3')
plt.xlabel('K values')
plt.ylabel('Squared Error (Cost)')
plt.grid(color='white', linestyle='-', linewidth=2)
plt.show()
```

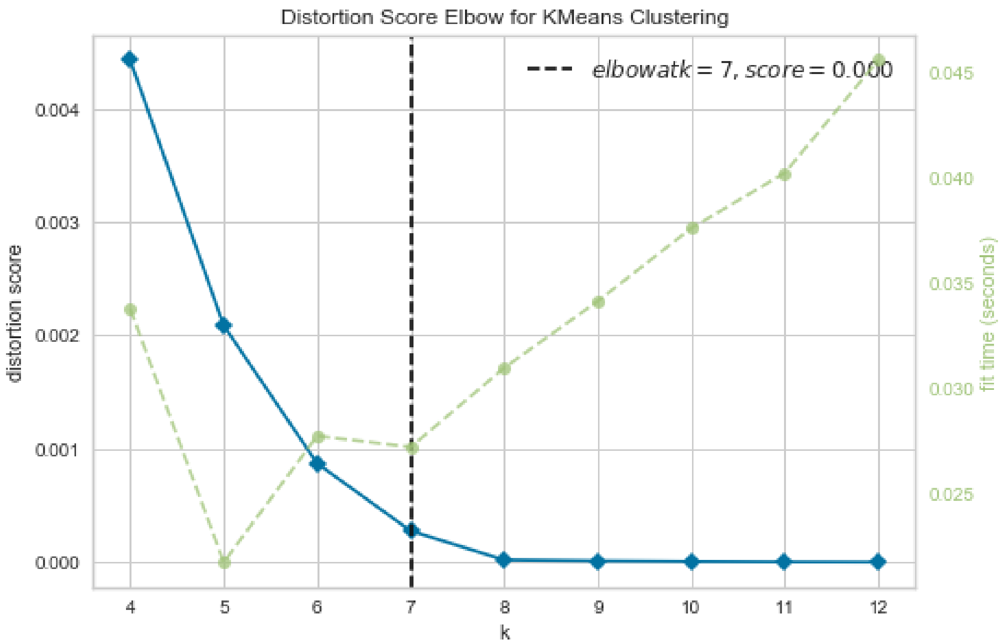


In [72]:

```
#!conda install -c districtdatalabs yellowbrick

from yellowbrick.cluster import KElbowVisualizer
```

In [73]:

```
# Instantiate the clustering model and visualizer
model = KMeans()
visualizer = KElbowVisualizer(model, k=(4,13))

visualizer.fit(toronto_part_clustering)        # Fit the data to the visualizer
visualizer.show()        # Finalize and render the figure
```



Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c22517650>
```

After analysing using elbow method using distortion score & Squared error for each K value, looks like K = 6 is the best value.

Clustering the Toronto Neighborhood Using K-Means with K = 6

In [74]:

```
kclusters = 6

toronto_part_clustering = toronto_part.drop('Neighborhood', 1)

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(toronto_part_clustering)

kmeans.labels_
```

Out[74]:

```
array([0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 4, 2, 0,
       0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       2, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 2, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 2, 2, 0, 5, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 4, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 5, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0], dtype=int32)
```

In [75]:

```
#sorted_neighborhoods_venues.drop(['Cluster Labels'],axis=1,inplace=True)
toronto_part.insert(0, 'Cluster Labels', kmeans.labels_)
toronto_merged = toronto_DF
# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
toronto_merged = toronto_merged.join(toronto_part.set_index('Neighborhood'), on='Neighbor
hood')
toronto_merged.dropna(subset=["Cluster Labels"], axis=0, inplace=True)
toronto_merged.reset_index(drop=True, inplace=True)
toronto_merged['Cluster Labels'].astype(int)
toronto_merged.head()
```

Out[75]:

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
0	Central Toronto	M4N	Lawrence Park	43.728020	-79.388790	0.0	0.00
1	Central Toronto	M4P	Davisville North	43.712751	-79.390197	0.0	0.00
2	Central Toronto	M4R	North Toronto West	43.715383	-79.405678	2.0	0.05
3	Central Toronto	M4S	Davisville	43.704324	-79.388790	0.0	0.00
4	Central Toronto	M5N	Roselawn	43.711695	-79.416936	0.0	0.00

Let us see the clusters visually on the map with the help of Folium.

In [76]:

```
import matplotlib.cm as cm
import matplotlib.colors as colors

map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11, width='90%', he
ight='70%')

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
```

```
markers_colors = []
for lat, lon, poi, cluster in zip(toronto_merged['Latitude'], toronto_merged['Longitude']
, toronto_merged['Neighborhood'], toronto_merged['Cluster Labels'].astype(int)):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
-1],
ap_clusters)
```

4.2 Examing the Clusters:

We have total of 6 clusters such as 0,1,2,3,4,5. Let us examine one after the other.

Cluster 0 contains all the neighborhoods which has least number of Chinese restaurants. It is shown in red color in the map

In [77]:

```
#Cluster 0
toronto_merged.loc[toronto_merged['Cluster Labels'] == 0]
```

Out[77]:

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
0	Central Toronto	M4N	Lawrence Park	43.728020	-79.388790	0.0	0.000000
1	Central Toronto	M4P	Davisville North	43.712751	-79.390197	0.0	0.000000
3	Central Toronto	M4S	Davisville	43.704324	-79.388790	0.0	0.000000

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
4	Central Toronto	M5N	Roselawn	43.711695	-79.416936	0.0	0.000000
5	Downtown Toronto	M4W	Rosedale	43.679563	-79.377529	0.0	0.000000
6	Downtown Toronto	M4Y	Church and Wellesley	43.665860	-79.383160	0.0	0.012195
7	Downtown Toronto	M5A	Harbourfront	43.654260	-79.360636	0.0	0.000000
8	Downtown Toronto	M5C	St. James Town	43.651494	-79.375418	0.0	0.000000
9	Downtown Toronto	M5E	Berczy Park	43.644771	-79.373306	0.0	0.000000
11	Downtown Toronto	M5W	Stn A PO Boxes 25 The Esplanade	43.646435	-79.374846	0.0	0.000000
12	Downtown Toronto	M6G	Christie	43.669542	-79.422564	0.0	0.000000
14	East Toronto	M4E	The Beaches	43.676357	-79.293031	0.0	0.000000
15	East Toronto	M4M	Studio District	43.659526	-79.340923	0.0	0.000000
16	East Toronto	M7Y	Business Reply Mail Processing Centre 969 Eastern	43.662744	-79.321558	0.0	0.000000
17	East York	M4C	Woodbine Heights	43.695344	-79.318389	0.0	0.000000
18	East York	M4G	Leaside	43.709060	-79.363452	0.0	0.000000
19	East York	M4H	Thorncliffe Park	43.705369	-79.349372	0.0	0.000000
20	East York	M4J	East Toronto	43.685347	-79.338106	0.0	0.000000
22	Etobicoke	M9W	Northwest	43.706748	-79.594054	0.0	0.000000
23	Mississauga	M7R	Canada Post Gateway Processing Centre	43.636966	-79.615819	0.0	0.000000
24	North York	M2H	Hillcrest Village	43.803762	-79.363452	0.0	0.000000
26	North York	M2N	Willowdale South	43.770120	-79.408493	0.0	0.000000
27	North York	M2P	York Mills West	43.752758	-79.400049	0.0	0.000000
28	North York	M2R	Willowdale West	43.782736	-79.442259	0.0	0.000000
29	North York	M3A	Parkwoods	43.753259	-79.329656	0.0	0.000000
30	North York	M3B	Don Mills North	43.745906	-79.352188	0.0	0.000000
31	North York	M3L	Downsview West	43.739015	-79.506944	0.0	0.000000
32	North York	M3M	Downsview Central	43.728496	-79.495697	0.0	0.000000
33	North York	M3N	Downsview Northwest	43.761631	-79.520999	0.0	0.000000
34	North York	M4A	Victoria Village	43.725882	-79.315572	0.0	0.000000
35	North York	M6B	Glencairn	43.709577	-79.445073	0.0	0.000000
36	North York	M9L	Humber Summit	43.756303	-[illegible]	0.0	0.000000

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
					79.565963		
38	Scarborough	M1G	Woburn	43.770992	-79.216917	0.0	0.000000
39	Scarborough	M1H	Cedarbrae	43.773136	-79.239476	0.0	0.000000
40	Scarborough	M1J	Scarborough Village	43.744734	-79.239476	0.0	0.000000
41	Scarborough	M1S	Agincourt	43.794200	-79.262029	0.0	0.000000
43	York	M6C	Humewood-Cedarvale	43.693781	-79.428191	0.0	0.000000
44	York	M6E	Caledonia-Fairbanks	43.689026	-79.453512	0.0	0.000000
45	York	M9N	Weston	43.706876	-79.518188	0.0	0.000000

Cluster 1 contains the neighborhoods which is sparsely populated with Chinese restaurants. It is shown in purple color in the map.

In [ ]:

```
#Cluster 1
toronto_merged.loc[toronto_merged['Cluster Labels'] == 1]
```

Cluster 2 has no rows meaning no data points or neighborhood was near to this centroid.

In [78]:

```
#Cluster 2
toronto_merged.loc[toronto_merged['Cluster Labels'] == 2]
```

Out[78]:

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
2	Central Toronto	M4R	North Toronto West	43.715383	-79.405678	2.0	0.05

Cluster 3 contains all the neighborhoods which is medium populated with Chinese restaurants. It is shown in blue color in the map.

In [79]:

```
#Cluster 3
toronto_merged.loc[toronto_merged['Cluster Labels'] == 3]
```

Out[79]:

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
25	North York	M2K	Bayview Village	43.786947	-79.385975	3.0	0.25

Cluster 4 has no rows meaning no data points or neighborhood was near to this centroid.

In [80]:

```
#Cluster 4
toronto_merged.loc[toronto_merged['Cluster Labels'] == 4]
```

Out[80]:

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
10	Downtown Toronto	M5G	Central Bay Street	43.657952	-79.387383	4.0	0.024096

	Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant
13	Downtown Toronto	M7A	Queen's Park	43.662301	-79.389494	4.0	0.026316
37	Queen's Park	M9A	Queen's Park	43.667856	-79.532242	4.0	0.026316

Cluster 5 contains all the neighborhoods which is densely populated with Chinese restaurants. It is shown in Orange color in the map

In [82]:

```
#Cluster 5
toronto_merged.loc[toronto_merged['Cluster Labels'] == 5]
```

Out[82]:

Borough	Postalcode	Neighborhood	Latitude	Longitude	Cluster Labels	Chinese Restaurant

5. Results and Discussion:

5.1 Results

We have reached the end of the analysis, in the result section we can document all the findinds from above clustering & visualization of the datas. In this project, as the business problem started with identifying a good neighborhood to open a new Chinese restaurant, we looked into all the neighborhoods in Toronto, analysed the Chinese population in each neighborhood & spread of Chinese restaurants in those neighborhoods to come to conclusion about which neighborhood would be a better spot for opening a new Chinese restaurant. I have used data from web resources like Wikipedia, geospatial coordinates of Toronto neighborhoods, and Foursquare API, to set up a very realistic data-analysis scenario. We have found out that —

- **In those 11 boroughs we identified that only Central Toronto, Downtown Tronto, East Toronto, East York, North York & Scarborough boroughs have high amount of Chinese restaurants with the help of Violin plots between Number of Chinese restaurants in Borough of Toronto.**

- **In all the ridings, Scarborough-Guildwood, Scarborough-Rouge Park, Scarborough Centre, Scarborough North, Humber River-Black Creek, Don Valley East, Scarborough Southwest, Don Valley North & Scarborough-Agincourt are the densely populated with Chinese crowd ridings.**

- **With the help of clusters examing & violin plots looks like Downtown Toronto, Central Toronto, East York are already densely populated with Chinese restaurants. So it is better idea to leave those boroughs out and consider only Scarborough, East Toronto & North York for the new restaurant's location.**

- **After careful consideration it is a good idea to open a new Chinese restaurant in Scarborough borough since it has high number of Chinese population which gives a higher number of customers possibility and lower competition since very less Chinese restaurants in the neighborhoods.**

5.2 Discussion

6. Conclusion:

Finally to conclude this project, We have got a chance to on a business problem like how a real like data scientists would do. We have used many python libraries to fetch the data , to manipulate the contents & to analyze and visualize those datasets. We have made use of Foursquare API to explore the venues in enighborhoods of Toronto, then get good amount of data from Wikipedia which we scraped with help of Wikipedia python library and visualized using various plots present in seaborn & matplotlib. We also applied machine learning technique to to predict the output given the data and used Folium to visualize it on a map. Also, some of the drawbacks or areas of improvements shows us that this analysis can further be improved with

Also, some of the drawbacks or areas of improvements shows us that this analysis can further be improved with help more data and different machine learning technique. Similarly we can use this project to analysis any scenario such opening a different cuisine or success of opening a new gym and etc. Hopefully, this project helps acts as initial guidance to take more complex real-life challenges using data-science.

In [ ]:

```
toronto_part.drop('Cluster Labels',axis=1, inplace=True)
```

In [ ]:

```

```