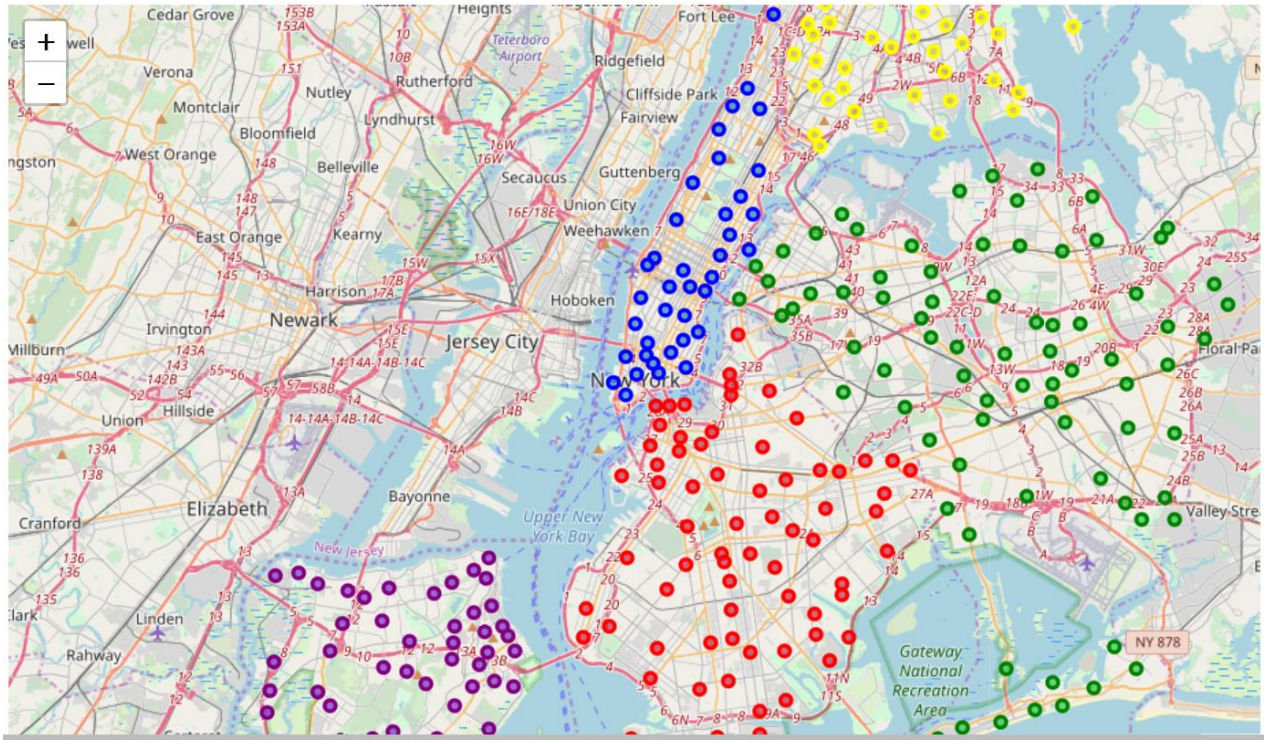
Similarities and Dissimilarities of the Venue Categories among Boroughs in New York City



Department of EECS, University of Tennessee

Xiao Kou, 01/01/2020

Table of Contents

A. Introduction

1. Background

New York City is the most populous metropolis in the United States. It is composed of 5 boroughs, including Manhattan, Brooklyn, Queens, Bronx and Staten Island. Each borough has its unique venues and characteristics.

2. Problem

This work intends to explore what are the top venue categories in each borough and what venue categories are exclusively-owned by each borough.

3. Stakeholders

This work can help tourists to get a better understanding of the similarities and dissimilarities among boroughs in New York City. It may also provide information for business-owners who want to choose new store locations in New York City.

B. Data

1. Data requirements

To analyze the venue categories of different boroughs in New York City, two datasets will be needed:

1) the geographical location of each neighborhood to visualize and segment the boroughs on the map.
2) the venues within each neighborhood (along with their corresponding categories) to explore the characteristics of each borough.

2. Data source

1) The geographical location of each neighbor is available from the website. The link for the neighborhood data is: https://geo.nyu.edu/catalog/nyu_2451_34572 [1]. People can also download the dataset from the following address: https://cocl.us/new_york_dataset.

2) The venue data within each neighborhood is obtained from Foursquare API [2]. Foursquare is a technology company that builds a massive dataset of accurate location data, and its API has been used widely used by the developers around the world [3]. With the non-commercial tier, developers can get 99,500 regular API calls and 500 premium API calls, which is sufficient for this work.

C.Methodology

1. Data acquisition

1) Get the geographical location of each neighborhood

First, the geographical data of each neighborhood is downloaded from https://cocl.us/new_york_dataset, as shown in Figure 1.

```
[3]: !wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
```

Figure 1. Downloading the geographical data.

After opening the json file, it is observed that the neighborhood dataset contains both the latitude and longitude data, as well as what borough it belongs to, as shown in Figure 2.

```
[4]: with open('newyork_data.json') as json_data:
         newyork_data = json.load(json_data)
```

Take *features* from the dataset.

```
[5]: neighborhoods_data = newyork_data['features']
     neighborhoods_data[0]
```

```
[5]: {'type': 'Feature',
      'id': 'nyu_2451_34572.1',
      'geometry': {'type': 'Point',
       'coordinates': [-73.84720052054902, 40.89470517661]},
      'geometry_name': 'geom',
      'properties': {'name': 'Wakefield',
       'stacked': 1,
       'annoline1': 'Wakefield',
       'annoline2': None,
       'annoline3': None,
       'annoangle': 0.0,
       'borough': 'Bronx',
       'bbox': [-73.84720052054902,
        40.89470517661,
        -73.84720052054902,
        40.89470517661]}}
```

Figure 2. Explore the New York neighborhood dataset.

Then, an empty Pandas dataframe is created. We loop through the data and fill the dataframe one row at a time, as shown in Figure 3.

```
[6]: # define the dataframe columns
     column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

     # instantiate the dataframe
     neighborhoods = pd.DataFrame(columns=column_names)
     neighborhoods
```

```
[6]:   Borough  Neighborhood  Latitude  Longitude
```

```
[7]: for data in neighborhoods_data:
         borough = data['properties']['borough']
         neighborhood_name = data['properties']['name']
         neighborhood_latlon = data['geometry']['coordinates']
         neighborhood_lat = neighborhood_latlon[1]
         neighborhood_lon = neighborhood_latlon[0]

         neighborhoods = neighborhoods.append({'Borough': borough,
                                               'Neighborhood': neighborhood_name,
                                               'Latitude': neighborhood_lat,
                                               'Longitude': neighborhood_lon}, ignore_index=True)
```

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

```
The dataframe has 5 boroughs and 306 neighborhoods.
```

Figure 3. Load the geographical data into the pandas dataframe.

Consequently, the resulted dataframe is given as follows:

```
neighborhoods.head()
```

[7]:

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

Figure 4. New York neighborhood dataframe.

2) Get the geographical location of each borough

The geographical location of each borough is obtained by executing the following code:

[9]:
```
address = 'New York City, NY'
geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
```

10]:
```
manhattan_data = neighborhoods[neighborhoods['Borough'] == 'Manhattan'].reset_index(drop=True)
brooklyn_data = neighborhoods[neighborhoods['Borough'] == 'Brooklyn'].reset_index(drop=True)
queens_data = neighborhoods[neighborhoods['Borough'] == 'Queens'].reset_index(drop=True)
bronx_data = neighborhoods[neighborhoods['Borough'] == 'Bronx'].reset_index(drop=True)
staten_island_data = neighborhoods[neighborhoods['Borough'] == 'Staten Island'].reset_index(drop=True)
```

Figure 5. Retrieve the geographical location of each borough.

3) Get the venue categories of each neighborhood.

First, the personal client_id and client_sceret data is loaded from a saved json file, as shown in Figure 6.

[19]:
```
secrets = json.load(open('credential.json'))
CLIENT_ID = secrets['CLIENT_ID']
CLIENT_SECRET = secrets['CLIENT_SECRET']
VERSION = '20200101'
```

Figure 6. Load client id and secret from the json file.

Then the maximum number of venues returned by Foursquare API is set as 100.

Limit of number of venues returned by Foursquare API

[20]:
```
LIMIT = 100
```

Figure 7. Limit the number of venues returned from API.

We define a get_category_type function to extract venue category, as shown in Figure 8.

Define function to extract the category of the venue.

[21]:
```
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']
```

Figure 8. Define a function to extract the venue category.

After that, a getNearbyVenues function is created to obtain the nearby venues of each neighborhood that are within 500-meter radius, as shown in Figure 9.

```
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 = 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])

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

    return(nearby_venues)
```

Figure 9. Define a function to get the nearby venues of each neighborhood.

We query the nearby venues' information of each borough and save them in the dataframe.

```
[23]: manhattan_venues = getNearbyVenues(names=manhattan_data['Neighborhood'],
                                         latitudes=manhattan_data['Latitude'],
                                         longitudes=manhattan_data['Longitude']
                                        )

[24]: brooklyn_venues = getNearbyVenues(names=brooklyn_data['Neighborhood'],
                                        latitudes=brooklyn_data['Latitude'],
                                        longitudes=brooklyn_data['Longitude']
                                       )

[25]: queens_venues = getNearbyVenues(names=queens_data['Neighborhood'],
                                      latitudes=queens_data['Latitude'],
                                      longitudes=queens_data['Longitude']
                                     )

[26]: bronx_venues = getNearbyVenues(names=bronx_data['Neighborhood'],
                                     latitudes=bronx_data['Latitude'],
                                     longitudes=bronx_data['Longitude']
                                    )

[27]: staten_island_venues = getNearbyVenues(names=staten_island_data['Neighborhood'],
                                             latitudes=staten_island_data['Latitude'],
                                             longitudes=staten_island_data['Longitude']
                                            )
```

Figure 10. Get the nearby venue data of each borough.

Finally, we check the size of the resulting dataframe and print how many unique categories in the returned venues, as shown in Figure 11.

```
[28]: print('The size of manhattan dataframe is {}.'.format(manhattan_venues.shape))
      print('The size of brooklyn dataframe is {}.'.format(brooklyn_venues.shape))
      print('The size of queens dataframe is {}.'.format(queens_venues.shape))
      print('The size of bronx dataframe is {}.'.format(bronx_venues.shape))
      print('The size of staten island dataframe is {}.'.format(staten_island_venues.shape))

      The size of manhattan dataframe is (3309, 7).
      The size of brooklyn dataframe is (2788, 7).
      The size of queens dataframe is (2119, 7).
      The size of bronx dataframe is (1215, 7).
      The size of staten island dataframe is (827, 7).

[29]: print('There are {} uniques categories in manhattan.'.format(len(manhattan_venues['Venue Category'].unique())))
      print('There are {} uniques categories in brooklyn.'.format(len(brooklyn_venues['Venue Category'].unique())))
      print('There are {} uniques categories in queens.'.format(len(queens_venues['Venue Category'].unique())))
      print('There are {} uniques categories in bronx.'.format(len(bronx_venues['Venue Category'].unique())))
      print('There are {} uniques categories in staten island.'.format(len(staten_island_venues['Venue Category'].unique())))

      There are 337 uniques categories in manhattan.
      There are 287 uniques categories in brooklyn.
      There are 271 uniques categories in queens.
      There are 168 uniques categories in bronx.
      There are 184 uniques categories in staten island.
```

Figure 11. Check the returned data.

2. Data preparation

1) List the top venue category in each borough.

In this section, first, we define a get_top_category function to group the venues in each borough and return the top 10 venue category in each borough.

```
[30]: def get_top_category(venue_name):
          df1=venue_name.groupby('Venue Category').size().reset_index(name="Count")
          df2 = df1.sort_values('Count', ascending=False)
          df2 = df2.set_index('Venue Category')
          df2[r'Percentage (%)']=df2['Count']/venue_name.shape[0]*100
          df3=df2.head(10)
          return(df3)
```

Figure 12. Define get_top_category function.

Next, this function is called to query the top venues in each borough.

```
[31]: man_top_venues=get_top_category(manhattan_venues)
      bln_top_venues=get_top_category(brooklyn_venues)
      que_top_venues=get_top_category(queens_venues)
      brx_top_venues=get_top_category(bronx_venues)
      sta_top_venues=get_top_category(staten_island_venues)
```

Figure 13. Call function to query the top venues in each borough.

2) List the unique venue category in each borough.

First, we define a get_category function.

```
[42]: def get_category(venue_name):
          df1=venue_name.groupby('Venue Category').size().reset_index(name="Count")
          df2 = df1.sort_values('Count', ascending=False)
          df2 = df2.set_index('Venue Category')
          return(df2)
```

Figure 14. Define get_category function.

Next, this function is called to get the venue categories in each borough.

```
[43]: man_venues=get_category(manhattan_venues)
      bln_venues=get_category(brooklyn_venues)
      que_venues=get_category(queens_venues)
      brx_venues=get_category(bronx_venues)
      sta_venues=get_category(staten_island_venues)
```

Figure 15. Call function to query the venue categories in each borough.

Then, the venue categories in the boroughs other than the selected one are extracted.

```
[44]: df1 = pd.concat([bln_venues, que_venues, brx_venues, sta_venues])
      df2 = pd.concat([man_venues, que_venues, brx_venues, sta_venues])
      df3 = pd.concat([man_venues, bln_venues, brx_venues, sta_venues])
      df4 = pd.concat([man_venues, bln_venues, que_venues, sta_venues])
      df5 = pd.concat([man_venues, bln_venues, que_venues, brx_venues])

[45]: temp1=list(df1.index)
      temp2=list(df2.index)
      temp3=list(df3.index)
      temp4=list(df4.index)
      temp5=list(df5.index)

[46]: man_venues.reset_index(inplace=True)
      bln_venues.reset_index(inplace=True)
      que_venues.reset_index(inplace=True)
      brx_venues.reset_index(inplace=True)
      sta_venues.reset_index(inplace=True)
```

Figure 16. Extract venue categories in boroughs that other than the selected one.

Finally, the unique venue categories in each borough are obtained.

```
[47]: man_venues.rename(columns = {'Venue Category':'Category'}, inplace = True)
      bln_venues.rename(columns = {'Venue Category':'Category'}, inplace = True)
      que_venues.rename(columns = {'Venue Category':'Category'}, inplace = True)
      brx_venues.rename(columns = {'Venue Category':'Category'}, inplace = True)
      sta_venues.rename(columns = {'Venue Category':'Category'}, inplace = True)

[48]: man_unique=man_venues[~man_venues.Category.isin(temp1)]
      bln_unique=bln_venues[~bln_venues.Category.isin(temp2)]
      que_unique=que_venues[~que_venues.Category.isin(temp3)]
      brx_unique=brx_venues[~brx_venues.Category.isin(temp4)]
      sta_unique=sta_venues[~sta_venues.Category.isin(temp5)]

[49]: man_tmp = man_unique.reset_index(drop=True)
      bln_tmp = bln_unique.reset_index(drop=True)
      que_tmp = que_unique.reset_index(drop=True)
      brx_tmp = brx_unique.reset_index(drop=True)
      sta_tmp = sta_unique.reset_index(drop=True)
```

Figure 17. Query the unique venue categories in each borough.

D.Results

1. Displaying boroughs on the map

In Figure 18, the blue circles represent the neighborhoods in Manhattan, the red circles represent the neighborhoods in Brooklyn, the green circles represent the neighborhoods in Queens, the yellow circles represent the neighborhoods in Bronx, and the purple circles represent the neighborhoods in Staten Island.

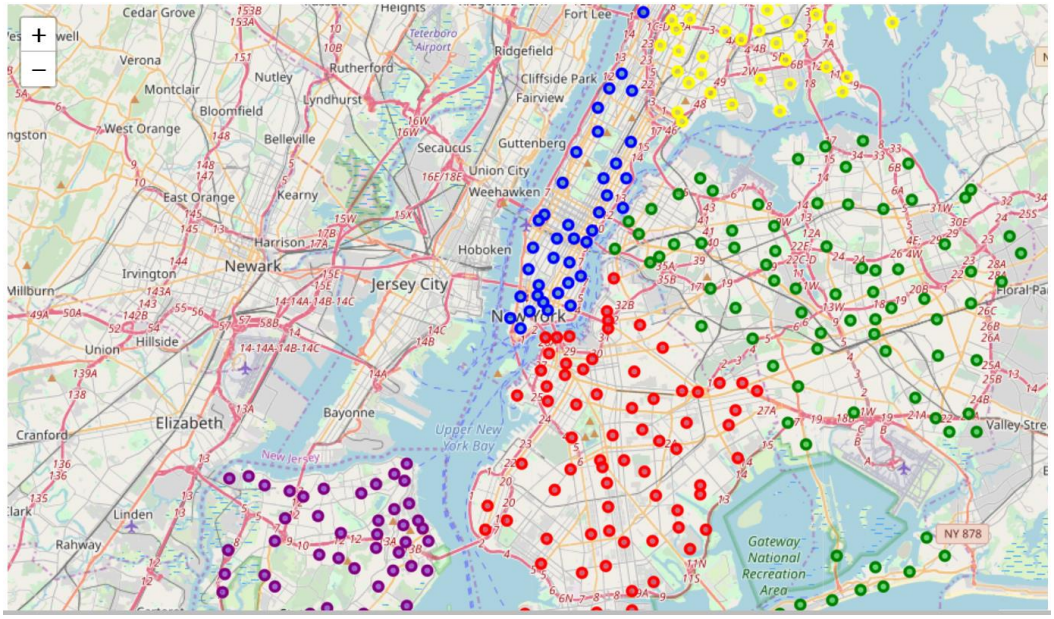


Figure 18. Neighborhoods in New York City.

2. Top venue categories in each borough.

The top 5 venue categories of each borough are given in TABLE I, and the top 10 venue categories in each borough are visualized in Figure 19.

TABLE I. Top venue categories in each borough.

Venue Category	Count	Percentage (%)
Coffee Shop	133	4.019341
Italian Restaurant	130	3.928679
American Restaurant	82	2.478090
Café	77	2.326987
Pizza Place	77	2.326987

(a) Manhattan

Venue Category	Count	Percentage (%)
Pizza Place	122	4.375897
Coffee Shop	90	3.228121
Bar	86	3.084648
Deli / Bodega	71	2.546628
Italian Restaurant	63	2.259684

(b) Brooklyn

Venue Category	Count	Percentage (%)
Pizza Place	88	4.152902
Deli / Bodega	78	3.680982
Chinese Restaurant	63	2.973101
Donut Shop	55	2.595564
Bakery	54	2.548372

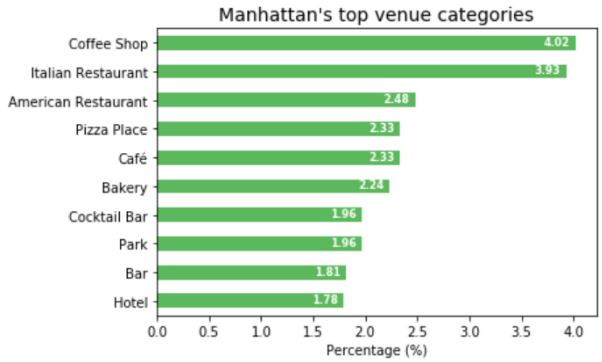
(c) Queens

Venue Category	Count	Percentage (%)
Pizza Place	98	8.065844
Deli / Bodega	53	4.362140
Donut Shop	46	3.786008
Italian Restaurant	43	3.539095

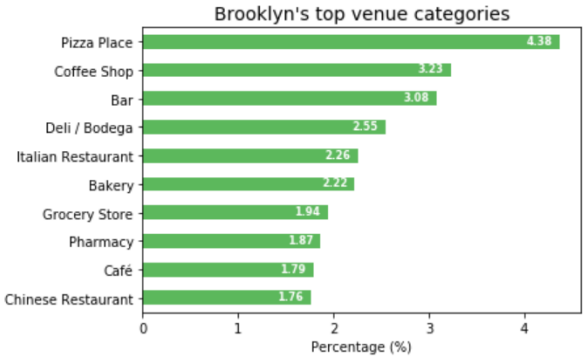
(d) Bronx

Venue Category	Count	Percentage (%)
Pizza Place	50	6.045949
Bus Stop	47	5.683192
Italian Restaurant	40	4.836759
Deli / Bodega	32	3.869407
Bagel Shop	23	2.781137

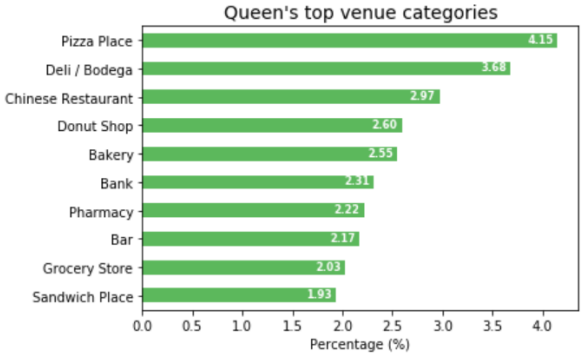
(e) Staten Island



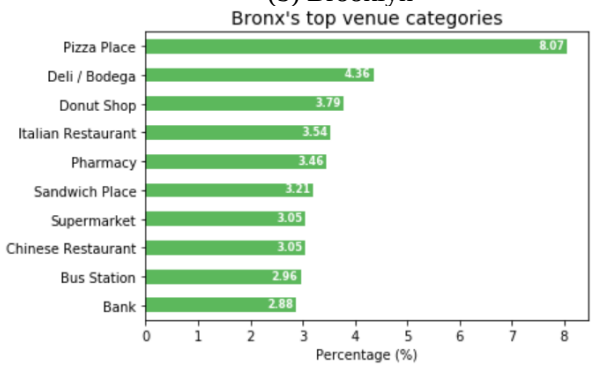
(a) Manhattan



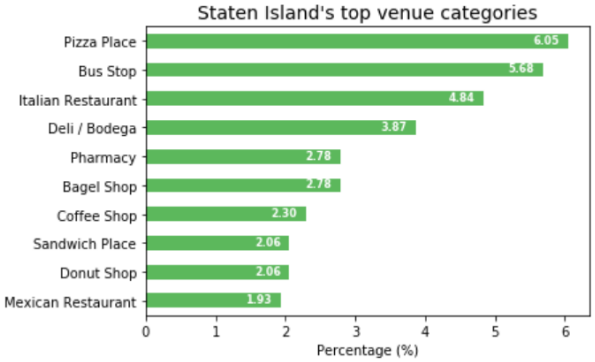
(b) Brooklyn



(c) Queens



(d) Bronx



(e) Staten Island

Figure 19. Top 10 venue categories in each borough.

3. Unique venue categories in each borough.

The top 5 unique venue categories of each borough are given in TABLE II, and the unique venue categories of each borough are visualized in Figure 20.

TABLE II. Top 5 unique venue categories in each borough.

	Category	Count
0	Exhibit	8
1	Australian Restaurant	5
2	Memorial Site	3
3	Pet Café	3
4	Music School	3

(a) Manhattan

	Category	Count
0	Piercing Parlor	2
1	Stadium	1
2	Auto Dealership	1
3	Event Service	1
4	Stationery Store	1

(b) Brooklyn

	Category	Count
0	Colombian Restaurant	3
1	Shop & Service	2
2	Bath House	1
3	State / Provincial Park	1
4	College Basketball Court	1

(c) Queens

	Category	Count
0	Track	2
1	Waste Facility	1
2	Shopping Plaza	1
3	River	1
4	Comic Shop	1

(d) Bronx

	Category	Count
0	Recording Studio	2
1	Toll Plaza	1
2	Sri Lankan Restaurant	1
3	Theme Park	1
4	Tex-Mex Restaurant	1

(e) Staten Island

(a) Manhattan

(b) Brooklyn

(c) Queens

(d) Bronx

(e) Staten Island

Figure 20. Unique venue categories in each borough.

E. Discussions

From TABLE I and Figure 19, the conclusions are as follows:

1) Pizza place is most popular venue category in New York City, except in Manhattan where coffee shop is the most popular venue category. The reason could be that many people work at Manhattan, and they need to drink a lot of coffee to perk themselves up.

2) Brooklyn has the most pizza places in New York City, with a total number of 122. While in Bronx, the number of pizza places is almost twice as many as the number of the second most popular venue category, which is deli/bodega.

3) Italian restaurant is also very popular in New York City, since it is within the top 5 popular venue categories most of the boroughs except Queens.

4) Queens has many Chinese restaurants and Korean restaurants, perhaps because Queens has the largest Asian population among the 5 boroughs.

5) Manhattan, Brooklyn, and Queens have a lot of bars.

6) Mexican restaurants are not very popular in New York City compared to Italian restaurants and Asian restaurants, but there are still some in Staten Island.

From TABLE II and Figure 20, the conclusions are as follows:

1) Manhattan and Queens have the most unique venue categories in New York City, which are followed by Brooklyn and Staten, and Bronx has the least number of unique venues categories in New York City.
2) Manhattan has as many as 8 exhibits, while other boroughs have none.
3) The memorial sites, pet cafés, music schools, and street arts also makes Manhattan unique to other boroughs.
4) There are 3 Columbian restaurants in Queens.
5) If people are interested in theme park or Sir Lankan food, they should visit Staten Island.

F. Conclusion

This work explores the similarities and dissimilarities among the 5 boroughs in New York City by comparing the venue categories within each borough. The data are from New York Department of City Planning and Foursquare. This work can help the people who are interested in New York City to get a better understanding of different boroughs. Moreover, it provides information for business-owners who want to choose new store locations in New York City. Finally, the methodology and code in this work also apply for analyzing other cities.

G. References

[1] New York Department of City Planning, "2014 New York City Neighborhood Names," [Online] available: https://geo.nyu.edu/catalog/nyu_2451_34572.

[2] Foursquare, [Online] available: https://developer.foursquare.com/.

[3] A. Aklson, and P. Lin, "Applied Data Science Capstone" [Online] available: https://www.coursera.org/.