# Capstone Project

## The Battle of Neighborhoods

## -Toyko City

# Table of contents

## 1. Introduction: Business Problem

1.1 Description of the problem

The business problem we are currently posing is: we are going to run some business in Toyko city. It is important to list and visualize Tokyo districts so that we could make a decision what business is best suitable in Tokyo.
Which business is suitable in Tokyo?
Which district is better?

1.2 Discussion of the background

Tokyo, as Japanese capital has a population of 13.92 million habitats and 44 million inhabitants. There are 23 districts in Tokyo. In 2021, more than 600000 overseas are expected to come to this city and surrounding regions to attend this upcoming Olympic Games event.

We believe it is good time to open a restaurant and provide more choices for visitors. Since they are from all over the world. On the contrary, I believe it's difficult for a traveler, especially restaurant-goers, to make a choice from among many options since there is also too much information on the web because everybody's got their own take of where to go and it's all so fragmented that you have to assemble it yourself especially if you're interested in non-touristy recommendations.

Therefore, we use Foursquare location data and machine learning to help us make decision and find appropriate neighborhoods. This is the problem we would like to address in this capstone project taking Tokyo as an example. In this project, I am going to use Foursquare location data and clustering methods to group the districts to different group by their venues information.

## 2. Data

We use two data sources as below:
**One from wiki website:**
https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#List_of_special_wards
Tokyo data that contains list districts (Wards) along with their latitude and longitude.

Description: We will Scrap Tokyo districts (Wards) Table from Wikipedia and get the coordinates of these 23 major districts using geocoder class of Geopy client.

## <u>Second form Foursquare APIs</u>

Description : By using this API we will get all the venues in each neighborhood. We can filter these venues to get different categories.

## 3. Methodology

### 3.1.1Tokyo Wards Table

In [3]:

```
data = pd.read_html("https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#List_of_special_wards")
df = data[3]
df
```

Out[3]:

| | No. | Flag | Name | Kanji | Population(as of October 2016 | Density(/km2) | Area(km2) | Major districts |
|---|---|---|---|---|---|---|---|---|
| **0** | 01 | NaN | Chiyoda | 千代田区 | 59441 | 5100 | 11.66 | Nagatachō, Kasumigaseki, Ōtemachi, Marunouchi,... |
| **1** | 02 | NaN | Chūō | 中央区 | 147620 | 14460 | 10.21 | Nihonbashi, Kayabachō, Ginza, Tsukiji, Hatchōb... |
| **2** | 03 | NaN | Minato | 港区 | 248071 | 12180 | 20.37 | Odaiba, Shinbashi, Hamamatsuchō, Mita, Roppong... |
| **3** | 04 | NaN | Shinjuku | 新宿区 | 339211 | 18620 | 18.22 | Shinjuku, Takadanobaba, Ōkubo, Kagurazaka, Ich... |
| **4** | 05 | NaN | Bunkyō | 文京区 | 223389 | 19790 | 11.29 | Hongō, Yayoi, Hakusan |
| **5** | 06 | NaN | Taitō | 台東区 | 200486 | 19830 | 10.11 | Ueno, Asakusa |
| **6** | 07 | NaN | Sumida | 墨田区 | 260358 | 18910 | 13.77 | Kinshichō, Morishita, Ryōgoku |
| **7** | 08 | NaN | Kōtō | 江東区 | 502579 | 12510 | 40.16 | Kiba, Ariake, Kameido, Tōyōchō, Monzennakachō,... |
| **8** | 09 | NaN | Shinagawa | 品川区 | 392492 | 17180 | 22.84 | Shinagawa, Gotanda, Ōsaki, Hatanodai, Ōimachi,... |
| **9** | 10 | NaN | Meguro | 目黒区 | 280283 | 19110 | 14.67 | Meguro, Nakameguro, Jiyugaoka, Komaba, Aobadai |
| **10** | 11 | NaN | Ōta | 大田区 | 722608 | 11910 | 60.66 | Ōmori, Kamata, Haneda, Den-en-chōfu |
| **11** | 12 | NaN | Setagaya | 世田谷区 | 910868 | 15690 | 58.05 | Setagaya, Shimokitazawa, Kinuta, Karasuyama, T... |
| **12** | 13 | NaN | Shibuya | 渋谷区 | 227850 | 15080 | 15.11 | Shibuya, Ebisu, Harajuku, Daikanyama, Hiroo, S... |
| **13** | 14 | NaN | Nakano | 中野区 | 332902 | 21350 | 15.59 | Nakano |
| **14** | 15 | NaN | Suginami | 杉並区 | 570483 | 16750 | 34.06 | Kōenji, Asagaya, Ogikubo |
| **15** | 16 | NaN | Toshima | 豊島区 | 294673 | 22650 | 13.01 | Ikebukuro, Komagome, Senkawa, Sugamo |
| **16** | 17 | NaN | Kita | 北区 | 345063 | 16740 | 20.61 | Akabane, Ōji, Tabata |
| **17** | 18 | NaN | Arakawa | 荒川区 | 213648 | 21030 | 10.16 | Arakawa, Machiya, Nippori, Minamisenju |
| **18** | 19 | NaN | Itabashi | 板橋区 | 569225 | 17670 | 32.22 | Itabashi, Takashimadaira |
| **19** | 20 | NaN | Nerima | 練馬区 | 726748 | 15120 | 48.08 | Nerima, Ōizumi, Hikarigaoka |
| **20** | 21 | NaN | Adachi | 足立区 | 674067 | 12660 | 53.25 | Ayase, Kitasenju, Takenotsuka |
| **21** | 22 | NaN | Katsushika | 葛飾区 | 447140 | 12850 | 34.80 | Tateishi, Aoto, Kameari, Shibamata |
| **22** | 23 | NaN | Edogawa | 江戸川区 | 685899 | 13750 | 49.90 | Kasai, Koiwa |
| **23** | Overall | Overall | Overall | Overall | 9375104 | 15146 | 619.00 | NaN |

In [6]:
```
df3=df2.drop('Major districts', axis=1)
df3.rename(columns={ "Population(as of October 2016)":"Population","Density(/km2)":"Density","Area(km2)":"Area" }, inplace=True
df3
```

Out[6]:

| | No. | Name | Kanji | Population(as of October 2016 | Density | Area |
|---|---|---|---|---|---|---|
| 0 | 01 | Chiyoda | 千代田区 | 59441 | 5100 | 11.66 |
| 1 | 02 | Chūō | 中央区 | 147620 | 14460 | 10.21 |
| 2 | 03 | Minato | 港区 | 248071 | 12180 | 20.37 |
| 3 | 04 | Shinjuku | 新宿区 | 339211 | 18620 | 18.22 |
| 4 | 05 | Bunkyō | 文京区 | 223389 | 19790 | 11.29 |
| 5 | 06 | Taitō | 台東区 | 200486 | 19830 | 10.11 |
| 6 | 07 | Sumida | 墨田区 | 260358 | 18910 | 13.77 |
| 7 | 08 | Kōtō | 江東区 | 502579 | 12510 | 40.16 |
| 8 | 09 | Shinagawa | 品川区 | 392492 | 17180 | 22.84 |
| 9 | 10 | Meguro | 目黒区 | 280283 | 19110 | 14.67 |
| 10 | 11 | Ōta | 大田区 | 722608 | 11910 | 60.66 |
| 11 | 12 | Setagaya | 世田谷区 | 910868 | 15690 | 58.05 |
| 12 | 13 | Shibuya | 渋谷区 | 227850 | 15080 | 15.11 |
| 13 | 14 | Nakano | 中野区 | 332902 | 21350 | 15.59 |
| 14 | 15 | Suginami | 杉並区 | 570483 | 16750 | 34.06 |
| 15 | 16 | Toshima | 豊島区 | 294673 | 22650 | 13.01 |
| 16 | 17 | Kita | 北区 | 345063 | 16740 | 20.61 |
| 17 | 18 | Arakawa | 荒川区 | 213648 | 21030 | 10.16 |
| 18 | 19 | Itabashi | 板橋区 | 569225 | 17670 | 32.22 |
| 19 | 20 | Nerima | 練馬区 | 726748 | 15120 | 48.08 |
| 20 | 21 | Adachi | 足立区 | 674067 | 12660 | 53.25 |
| 21 | 22 | Katsushika | 葛飾区 | 447140 | 12850 | 34.80 |
| 22 | 23 | Edogawa | 江戸川区 | 685899 | 13750 | 49.90 |

## 3.1.2 Getting Coordinates

In [7]:
```
from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="Tokyo_explorer")

df3['Major_Dist_Coord']= df3['Kanji'].apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude))
df3[['Latitude','Longitude']] = df3['Major_Dist_Coord'].apply(pd.Series)

df3.drop(['Major_Dist_Coord'], axis=1, inplace=True)
tokyo_data=df3
tokyo_data
```

Out[7]:

| | No. | Name | Kanji | Population(as of October 2016 | Density | Area | Latitude | Longitude |
|---|---|---|---|---|---|---|---|---|
| **0** | 01 | Chiyoda | 千代田区 | 59441 | 5100 | 11.66 | 35.693810 | 139.753216 |
| **1** | 02 | Chūō | 中央区 | 147620 | 14460 | 10.21 | 35.666255 | 139.775565 |
| **2** | 03 | Minato | 港区 | 248071 | 12180 | 20.37 | 35.643227 | 139.740055 |
| **3** | 04 | Shinjuku | 新宿区 | 339211 | 18620 | 18.22 | 35.693763 | 139.703632 |
| **4** | 05 | Bunkyō | 文京区 | 223389 | 19790 | 11.29 | 35.718810 | 139.744732 |
| **5** | 06 | Taitō | 台東区 | 200486 | 19830 | 10.11 | 35.717450 | 139.790859 |
| **6** | 07 | Sumida | 墨田区 | 260358 | 18910 | 13.77 | 35.700429 | 139.805017 |
| **7** | 08 | Kōtō | 江東区 | 502579 | 12510 | 40.16 | 35.649154 | 139.812790 |
| **8** | 09 | Shinagawa | 品川区 | 392492 | 17180 | 22.84 | 35.599252 | 139.738910 |
| **9** | 10 | Meguro | 目黒区 | 280283 | 19110 | 14.67 | 35.621250 | 139.688014 |
| **10** | 11 | Ōta | 大田区 | 722608 | 11910 | 60.66 | 35.561206 | 139.715843 |
| **11** | 12 | Setagaya | 世田谷区 | 910868 | 15690 | 58.05 | 35.646096 | 139.656270 |
| **12** | 13 | Shibuya | 渋谷区 | 227850 | 15080 | 15.11 | 35.664596 | 139.698711 |
| **13** | 14 | Nakano | 中野区 | 332902 | 21350 | 15.59 | 35.718123 | 139.664468 |
| **14** | 15 | Suginami | 杉並区 | 570483 | 16750 | 34.06 | 35.699493 | 139.636288 |
| **15** | 16 | Toshima | 豊島区 | 294673 | 22650 | 13.01 | 35.736156 | 139.714222 |
| **16** | 17 | Kita | 北区 | 345063 | 16740 | 20.61 | 35.755838 | 139.736687 |
| **17** | 18 | Arakawa | 荒川区 | 213648 | 21030 | 10.16 | 35.737529 | 139.781310 |
| **18** | 19 | Itabashi | 板橋区 | 569225 | 17670 | 32.22 | 35.774143 | 139.681209 |
| **19** | 20 | Nerima | 練馬区 | 726748 | 15120 | 48.08 | 35.748360 | 139.638735 |
| **20** | 21 | Adachi | 足立区 | 674067 | 12660 | 53.25 | 35.783703 | 139.795319 |

## 3.2. Exploratory Data Analysis:

### 3.2.1 Using Foursquare API


### Define Foursquare Credentials and Version

In [10]:
```
CLIENT_ID = 'CZLFHE305MTINUUGLMXCBOEXKMKYNX0J1VV3MLUS5S0TARSS' # your Foursquare ID
CLIENT_SECRET = 'JGF0UE02SMJWDNUHNWCWQNKPXKV5S1MQ2THL2CLZQP3QI0Q3' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: CZLFHE305MTINUUGLMXCBOEXKMKYNX0J1VV3MLUS5S0TARSS
CLIENT_SECRET:JGF0UE02SMJWDNUHNWCWQNKPXKV5S1MQ2THL2CLZQP3QI0Q3
```

In [12]:
```
tokyo_data.loc[0, 'Kanji']
```

Out[12]:
```
'千代田区'
```

### Get the neighborhood's latitude and longitude values.

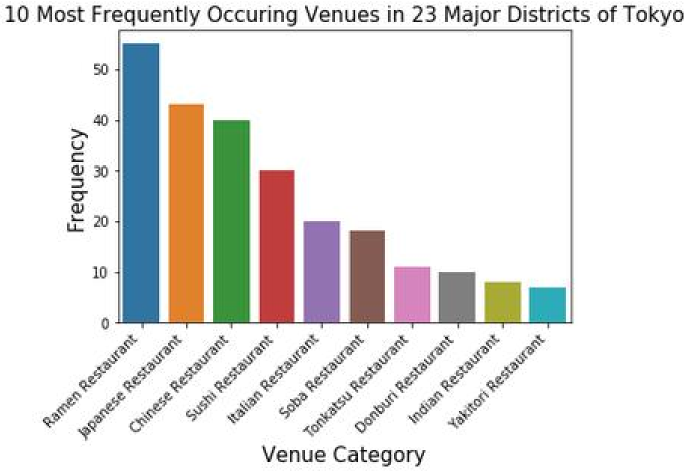
In [13]:
```
kanji_latitude = tokyo_data.loc[0, 'Latitude'] # neighborhood latitude value
kanji_longitude = tokyo_data.loc[0, 'Longitude'] # neighborhood longitude value

kanji_name = tokyo_data.loc[0, 'Kanji'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.format(kanji_name,
                                                               kanji_latitude,
                                                               kanji_longitude))
```

```
Latitude and longitude values of 千代田区 are 35.6938097, 139.7532163.
```

# 4. Results and Discussion

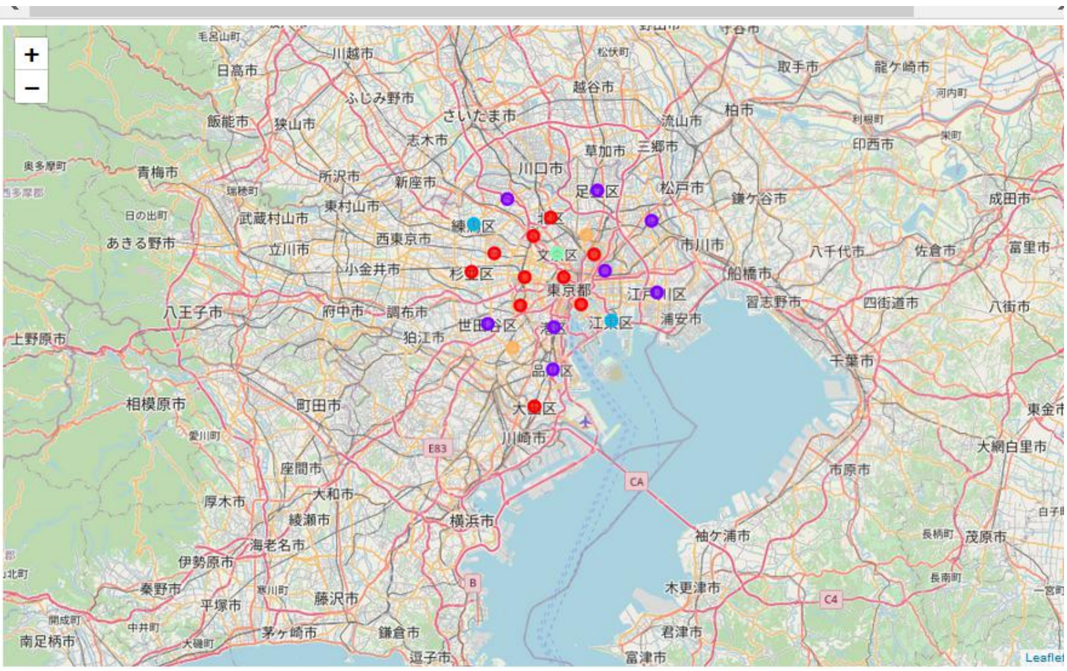
In [55]:

```
import seaborn as sns
from matplotlib import pyplot as plt

s=sns.barplot(x="Venue_Category", y="Frequency", data=Tokyo_5_Dist_Venues_Top10)
s.set_xticklabels(s.get_xticklabels(), rotation=45, horizontalalignment='right')

plt.title('10 Most Frequently Occuring Venues in 23 Major Districts of Tokyo', fontsize=15)
plt.xlabel("Venue Category", fontsize=15)
plt.ylabel ("Frequency", fontsize=15)
plt.savefig("Most_Freq_Venues1.png", dpi=300)
fig = plt.figure(figsize=(18,7))
plt.show()
```




```
<Figure size 1296x504 with 0 Axes>
```

4]:



## 5. Conclusion