Battle of Neighborhoods *(Week 1)*

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

Prospects of a Lunch Restaurant, Close to Office Areas in Tokyo, Japan.

Tokyo, where I am currently staying, is the most populous metropolitan area in the world. Currently ranked 3rd in the global economic power index, Tokyo is definitely one of the best places to start up a new business.

During the daytime, specially in the morning and lunch hours, office areas provide huge opportunities for restaurants. Reasonably priced (one lunch meal $ 8\text{\$} $) shops are usually always fill during the lunch hours (11 am -- 2 pm) and given this scenario, we will go through the benefits and pitfalls of opening a breakfast cum lunch restaurants in highly densed office places. Usually the profit margin for a decent restaurant lie within $15 - 20\% $ range but, it can even go high enough to 35%, as discussed here.

We will go through each step of this project and address them separately. For this week I just describe the initial data preparation and future steps to start the battle of neighborhoods in Tokyo.

1. Obtain the Data

   1.a. Name of the 23 Wards, area and population from web scrapping

   1.b. Obtain information about best business districts.

   1.c. Use Foresquare Data to obtain info about restaurants.

1. Data Visualization and Some Simple Statistical Analysis.
2. Analysis Using Clustering, Specially K-Means Clustering.

   3.a. Maximize the number of clusters.

   3.b. Visualization using Chloropleth Map

1. Compare the Neighborhoods to Find the Best Place for Starting up a Restaurant.

1. Inference From these Results and related Conclusions.

__Target Audience__

1. Business personnel who wants to invest or open a restaurant. This analysis will be a comprehensive guide to start or expand restaurants targeting the large pool of office workers in Tokyo during lunch hours.
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. New graduates, to find reasonable lunch/breakfast place close to office.
4. Budding Data Scientists, who wants to implement some of the most used Exploratory Data Analysis techniques to obtain
necessary data, analyze it and, finally be able to tell a story out of it.

2. Preparation for Data (Data Section)

Use Wikipedia and scrap the names of 23 Wards and corresponding Major District Name within each ward.

Process the information properly to retrieve the necessary details and create a dataframe.

Use Geopy to get the coordinates of all the major districts. Refine and check and compare with google searches for extra cautions.

3. Programming Section (Initial Data Processing From Wikipage and Get the Coordinates)

3.1. Import Libraries

In [5]:

```
import requests
import json

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors

from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
```

3.2. Use Requests and BeautifulSoup to Get the Info from Table in Wikipedia Page¶

In [6]:

```
response_obj =
requests.get('https://en.wikipedia.org/wiki/Special_wards_of_Tokyo').text
soup = BeautifulSoup(response_obj,'lxml')
Wards_Tokyo_Table = soup.find('table', {'class':'wikitable sortable'})
```

3.3. Process the Info and Create the DataFrame

In [9]:

```
Name=[]
Kanji = []
Pop = []
Density = []
num = []
flag = []
Area = []
Major_District = []

for row in Wards_Tokyo_Table.findAll("tr"):
    Ward = row.findAll('td')
    if len(Ward)==8:
        num.append(Ward[0].find(text=True))
        flag.append(Ward[1].findAll('a'))
        Name.append(Ward[2])
        Kanji.append(Ward[3].find(text=True))
        Pop.append(Ward[4])
        Area.append(Ward[5].find(text=True))
        Major_District.append(Ward[7].find(text=True))



#+++++++++++++++++++++++++++++++++++++++++++++++
#+ Area
#+++++++++++++++++++++++++++++++++++++++++++++++
Area = ['5100' if x=='0' else x for x in Area]
New_Area = []

# change the type of Area list
for l in range(len(Area)):
    x=Area[l].replace(",","")
    New_Area.append(x)

New_Area=[int(s) for s in New_Area]

#++++++++++++++++++++++++++++++++++++++++++++++++++++++
#+ Name of the Wards
#++++++++++++++++++++++++++++++++++++++++++++++++++++++
new_names = []
for n in range(len(Name)):
    names = Name[n].findAll('a')
    new_names.append(names)

flat_new_names_list = [item for sublist in new_names for item in sublist]

Wards_names= []
```

```
#now
for name_wards in flat_new_names_list:
        Wards_names.append(name_wards.get('title'))

# replace the elements in the list that contains 'Tokyo' with only the ward names
replace_names={'Chiyoda, Tokyo':'Chiyoda', 'Chūō, Tokyo':'Chuo', 'Minato,
Tokyo':'Minato',
               'Sumida, Tokyo':'Sumida', 'Koto, Tokyo':'Koto', 'Ōta, Tokyo':'Ota',
'Nakano, Tokyo':'Nakano',
               'Kita, Tokyo':'Kita', 'Arakawa, Tokyo':'Arakawa', 'Adachi,
Tokyo':'Adachi', 'Edogawa, Tokyo':'Edogawa'}


Wards_names1 = [replace_names.get(n1,n1) for n1 in Wards_names]

#+++++++++++++++++++++++++++++++++++++++++++++++++++++
#+ Population
#+++++++++++++++++++++++++++++++++++++++++++++++++++++
population = []
for p in range(len(Pop)):
    pops = Pop[p].text[1:9]
    population.append(pops)

New_population = []
for po in range(len(population)):
    xy=population[po].replace(",","")
    New_population.append(xy)

New_population=[int(s1) for s1 in New_population]
#++++++++++++++++++++++++++++++++++++++++++++++++
#+ Major Districts
#++++++++++++++++++++++++++++++++++++++++++++++++
replace_districts = {'Nagatachō':'Nagatacho', 'Hongō':'Hongo',
'Kinshichō':'Kinshicho', 'Ōmori': 'Omori',
                     'Kōenji':'Koenji', 'Arakawa, Machiya, ':'Arakawa', 'Ayase,
':'Ayase', 'Kasai, Koiwa\n':'Kasai'}


Major_District_names1 = [replace_districts.get(n2,n2) for n2 in Major_District]


#++++++++++++++++++++++++++++
#+ Create DataFrame
#++++++++++++++++++++++++++++
df=pd.DataFrame(Wards_names1,columns=['Ward'])
df['Area_SqKm'] = New_Area
df['Population'] = New_population
```

```
df['Major_District'] = Major_District_names1
df.index = np.arange(1, len(df) + 1)
df
```

Out[9]:

	Ward	Area_SqKm	Population	Major_District
1	Chiyoda	5100	59441	Nagatacho
2	Chuo	14460	147620	Nihonbashi
3	Minato	12180	248071	Odaiba
4	Shinjuku	18620	339211	Shinjuku
5	Bunkyō	19790	223389	Hongo
6	Taitō	19830	200486	Ueno
7	Sumida	18910	260358	Kinshicho
8	Koto	12510	502579	Kiba
9	Shinagawa	17180	392492	Shinagawa
10	Meguro	19110	280283	Meguro
11	Ota	11910	722608	Omori
12	Setagaya	15690	910868	Setagaya
13	Shibuya	15080	227850	Shibuya
14	Nakano	21350	332902	Nakano
15	Suginami	16750	570483	Koenji
16	Toshima	22650	294673	Ikebukuro
17	Kita	16740	345063	Akabane
18	Arakawa	21030	213648	Arakawa
19	Itabashi	17670	569225	Itabashi
20	Nerima	15120	726748	Nerima
21	Adachi	12660	674067	Ayase
22	Katsushika	12850	447140	Tateishi
23	Edogawa	13750	685899	Kasai

3.4. Get the Coordinates of the Major Districts, Process and Refine

In [10]:

```
from geopy.geocoders import Nominatim
geolocator = Nominatim()
df['Major_Dist_Coord']= df['Major_District'].apply(geolocator.geocode).apply(lambda
x: (x.latitude, x.longitude))
df[['Latitude', 'Longitude']] = df['Major_Dist_Coord'].apply(pd.Series)



df[['Latitude', 'Longitude']] = df['Major_Dist_Coord'].apply(pd.Series)
df.drop(['Major_Dist_Coord'], axis=1, inplace=True)
```

```
df
```

```
/home/suvo/Videos/Jupyter/lib/python3.6/site-packages/ipykernel_launcher.py:2: DeprecationWarning: Using Nominatim with the default "geopy/1.19.0" `user_agent` is strongly discouraged, as it violates Nominatim's ToS https://operations.osmfoundation.org/policies/nominatim/ and may possibly cause 403 and 429 HTTP errors. Please specify a custom `user_agent` with `Nominatim(user_agent="my-application")` or by overriding the default `user_agent`: `geopy.geocoders.options.default_user_agent = "my-application"`. In geopy 2.0 this will become an exception.
```

Out[10]:

	Ward	Area_SqKm	Population	Major_District	Latitude	Longitude
1	Chiyoda	5100	59441	Nagatacho	35.675618	139.743469
2	Chuo	14460	147620	Nihonbashi	35.684058	139.774501
3	Minato	12180	248071	Odaiba	35.619128	139.779403
4	Shinjuku	18620	339211	Shinjuku	35.693763	139.703632
5	Bunkyō	19790	223389	Hongo	32.509380	-116.297001
6	Taitō	19830	200486	Ueno	35.711788	139.776096
7	Sumida	18910	260358	Kinshicho	35.696752	139.814151
8	Koto	12510	502579	Kiba	23.013134	-80.832875
9	Shinagawa	17180	392492	Shinagawa	35.599252	139.738910
10	Meguro	19110	280283	Meguro	35.621250	139.688014
11	Ota	11910	722608	Omori	-38.904706	175.755211
12	Setagaya	15690	910868	Setagaya	35.646096	139.656270
13	Shibuya	15080	227850	Shibuya	35.664596	139.698711
14	Nakano	21350	332902	Nakano	35.718123	139.664468
15	Suginami	16750	570483	Koenji	35.704942	139.649909
16	Toshima	22650	294673	Ikebukuro	35.730103	139.711884
17	Kita	16740	345063	Akabane	35.778139	139.720800
18	Arakawa	21030	213648	Arakawa	35.737529	139.781310
19	Itabashi	17670	569225	Itabashi	35.774143	139.681209
20	Nerima	15120	726748	Nerima	35.748360	139.638735
21	Adachi	12660	674067	Ayase	35.446369	139.430925
22	Katsushika	12850	447140	Tateishi	34.176335	132.226020
23	Edogawa	13750	685899	Kasai	-5.349800	21.424098

3.5. Replace the 4 Wrong Coodinates¶

Google Search Helps

Hongo -- 35.7088° N, 139.7601° E

Kiba -- 35.6722° N, 139.8061° E

Omori -- 35.5884° N, 139.7279° E

Kasai -- 35.6634° N, 139.8731° E

In [19]:

```
Lat_list = df['Latitude'].tolist()
Long_list = df['Longitude'].tolist()

replace_latitudes = {32.5093796:35.7088, 23.0131338:35.6722, -38.9047057:35.5884, -
5.3498001:35.6634}
replace_longitudes = {-116.2970014:139.7601, -80.8328748:139.8061,
175.7552111:139.7279, 21.424098:139.8731}

latitudes_new = [replace_latitudes.get(n3,n3) for n3 in Lat_list]
longtitudes_new = [replace_longitudes.get(n4,n4) for n4 in Long_list]
Tokyo_df = df.drop(['Latitude', 'Longitude'], axis=1)
# #df.drop(['Longitude'], axis=1, inplace=True)
# Tokyo_df
```

In [20]:

```
Tokyo_df['Dist_Latitude'] = latitudes_new
Tokyo_df['Dist_Longitude'] = longtitudes_new



Tokyo_df
```

Out[20]:

	Ward	Area_SqKm	Population	Major_District	Dist_Latitude	Dist_Longitude
1	Chiyoda	5100	59441	Nagatacho	35.675618	139.743469
2	Chuo	14460	147620	Nihonbashi	35.684058	139.774501
3	Minato	12180	248071	Odaiba	35.619128	139.779403
4	Shinjuku	18620	339211	Shinjuku	35.693763	139.703632
5	Bunkyō	19790	223389	Hongo	35.708800	139.760100
6	Taitō	19830	200486	Ueno	35.711788	139.776096
7	Sumida	18910	260358	Kinshicho	35.696752	139.814151
8	Koto	12510	502579	Kiba	35.672200	139.806100
9	Shinagawa	17180	392492	Shinagawa	35.599252	139.738910
10	Meguro	19110	280283	Meguro	35.621250	139.688014
11	Ota	11910	722608	Omori	35.588400	139.727900
12	Setagaya	15690	910868	Setagaya	35.646096	139.656270
13	Shibuya	15080	227850	Shibuya	35.664596	139.698711
14	Nakano	21350	332902	Nakano	35.718123	139.664468
15	Suginami	16750	570483	Koenji	35.704942	139.649909

	Ward	Area_SqKm	Population	Major_District	Dist_Latitude	Dist_Longitude
16	Toshima	22650	294673	Ikebukuro	35.730103	139.711884
17	Kita	16740	345063	Akabane	35.778139	139.720800
18	Arakawa	21030	213648	Arakawa	35.737529	139.781310
19	Itabashi	17670	569225	Itabashi	35.774143	139.681209
20	Nerima	15120	726748	Nerima	35.748360	139.638735
21	Adachi	12660	674067	Ayase	35.446369	139.430925
22	Katsushika	12850	447140	Tateishi	34.176335	132.226020
23	Edogawa	13750	685899	Kasai	35.663400	139.873100

Conclusion

1st Week: Description of Problem and Data Preparation

We get the Initial Data-Frame with Names of Major Wards, and corresponding districts in those Major Wards
and the coordinates of those major districts. Before comparing all the wards, since we want to concentrate only on lunch restaurants targeting the office workers, we need to get the idea about the best business areas in Tokyo. Here we want to concentrate on the best five wards

1. Chiyoda. Major District: *Nagatacho*
2. Shinjuku. Major District: *Shinjuku*
3. Shibuya. Major District: *Shibuya*
4. Chuo. Major District: *Nihombashi*
5. Shinagawa. Major District: *Shinagawa*

So as the next step we will use Foursquare data and obtain information on restaurants. With these, we can start with our battle of neighborhoods for opening a restaurant in Tokyo.

In [ ]: