The Battle of Neighborhoods of Singapore

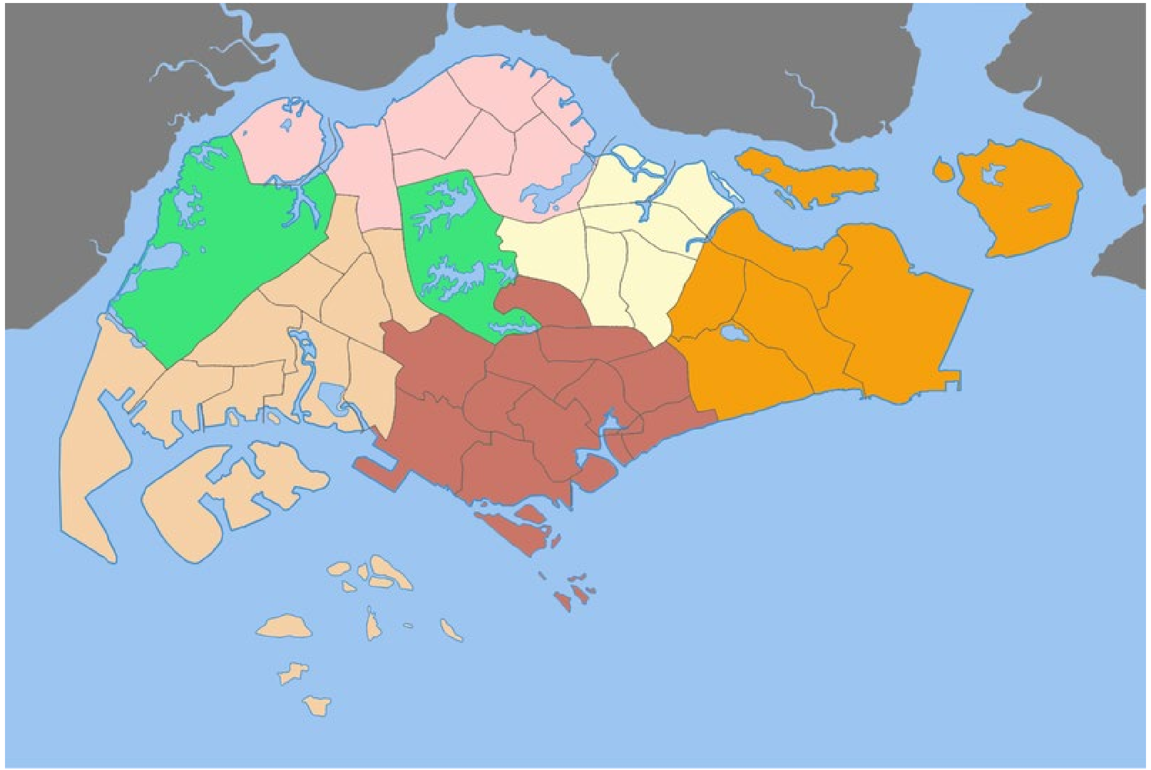
Business Problem

To Recommend the best place to open a Cafe in Singapore.

People will visit Cafe for numerous reasons and on different occasions. On weekdays, students might visit after school to ***do assignments*** together with their school mates. For cafes near to offices, working adults may use it as a place for ***informal meetings*** and ***discussions*** or just to ***grab a cup of coffee*** in the morning before heading to the office. On weekends, adults will usually visit a Cafe to meet and catch up with friends. For cafes ***near to public transportations*** and ***attractions*** or ***hostels and hotels***, tourists will tend to visit due to convenience. Therefore, we can infer first hand that the visitors to the cafe is ***heavily dependent on its location*** on weekdays but less likely on weekends.

In order to ***maximise revenues***, cafes need to be located ***near offices*** to capture the weekday crowd but also not too far away from ***places of attractions*** and ***hotels*** where tourists will pay a visit. As working adults has larger spending powers than students, the analysis will focus more on the working adults.

Introduction – Country of Interest



Source: https://en.wikipedia.org/wiki/Planning_Areas_of_Singapore

Singapore

There are a total of **55** planning areas organised into these **5** regions. The information of these **55** planning areas can be extracted from *wikipedia*.

The **5** Regions of Singapore are:

1. Central Region
2. East Region
3. North-East Region
4. North Region
5. West Region

Data Sources – Wikipedia, Google Map API & Foursquare API

Data Sources

To extract and transform into pandas dataframe:

1. **Webscrap Singapore's Neighborhood from Wikipedia**
   (Data Source: Wikipedia: https://en.wikipedia.org/wiki/Planning_Areas_of_Singapore)

2. **Extract Geographical Coordinates of Singapore's Neighborhood from Google Maps API**

3. **Extract Nearby Venues of Selected Neighborhoods using Foursquare API**

Data Extraction using *Beautiful Soup*

```
from bs4 import BeautifulSoup

source = requests.get('https://en.wikipedia.org/wiki/Planning_Areas_of_Singapore').text

soup = BeautifulSoup(source, 'html5lib')
```

```
table_post = soup.find_all('table')[2] #Grab the third table
fields = table_post.find_all('td')

EName = []
MName =[]
CName=[]
Pinyin =[]
TName =[]
Region = []
Area =[]
Population = []
Density =[]

for i in range(0, len(fields), 9):
    EName.append(fields[i].text.strip())
    MName.append(fields[i+1].text.strip())
    CName.append(fields[i+2].text.strip())
    Pinyin.append(fields[i+3].text.strip())
    TName.append(fields[i+4].text.strip())
    Region.append(fields[i+5].text.strip())
    Area.append(fields[i+6].text.strip())
    Population.append(fields[i+7].text.strip())
    Density.append(fields[i+8].text.strip())

df_sg = pd.DataFrame(data=[EName, MName, CName, Pinyin, TName, Region, Area, Population, Density]).transpose()
df_sg.columns = ['EName', 'MName', 'CName','Pinyin','TName','Region','Area','Population','Density']
df_sg.head()
```

Data Extraction using *Beautiful Soup*

df_sg.head()

Transform the data into a *pandas* dataframe that consists of the following columns: EName, MName, CName, Pinyin, TName, Region, Area, Population and Density

	EName	MName	CName	Pinyin	TName	Region	Area	Population	Density
0	Ang Mo Kio		宏茂桥	Hóng mào qiáo	ஆங் மோ கியோ	North-East	13.94	165,710	12,000
1	Bedok	*	勿洛	Wù luò	பிடோ	East	21.69	281,300	13,000
2	Bishan		碧山	Bì shān	பீஷான்	Central	7.62	88,490	12,000
3	Boon Lay		文礼	Wén lǐ	பூன் லே	West	8.23	30	3.6
4	Bukit Batok	*	武吉巴督	Wǔjí bā dū	புக்கிட் பாத்தோக்	West	11.13	144,410	13,000

Data Extraction using *Google Maps API*

googlemaps.Client (key = '*your keys*')

Use Googlemaps to get Geographical Coordinates for all 55 Singapore Neighborhoods.

```
# install the google map api client library
!pip install -U googlemaps
```

...

```
# @hidden_cell
import googlemaps
gmaps = googlemaps.Client(key='AIzaSyA9yEjViJb93QUEgGkZjIfwDzldcfqPoxQ')
```

```
lat = []
long = []

for i in range(0,len(df_sg2),1):
        geocode_result = gmaps.geocode('{}, Singapore'.format (df_sg2["EName2"][i]))
        lat.append(geocode_result[0]['geometry']['location']['lat'])
        long.append(geocode_result[0]['geometry']['location']['lng'])
print (lat, long)
```

```
[1.3691149, 1.3236038, 1.3525845, 1.3142556, 1.3590288, 1.2819046, 1.3774142, 1.3294113, 1.3551526, 1.3450101, 1.3219708, 1.3839803, 1.3161811, 1.2866961, 1.320
0544, 1.3612182, 1.3328572, 1.3403898, 1.3100334, 1.4304941, 1.4260074, 1.2912788, 1.279294, 1.3019687, 1.2966147, 1.3075517, 1.4063775, 1.3208572, 1.301674, 1.
2848825, 1.3720937, 1.3516087, 1.2857497, 1.3984457, 1.2941664, 1.2959376, 1.3051345, 1.405163, 1.4491107, 1.3868121, 1.3553567, 1.4443218, 1.2895263, 1.253656
8, 1.2785501, 1.4073821, 1.3495907, 1.306932, 1.3555189, 1.3343035, 1.2949472, 1.2478844, 1.3471977, 1.4381922, 1.430368] [103.8454342, 103.9273405, 103.835211
6, 103.7093099, 103.7636796, 103.8239182, 103.7719498, 103.8020777, 103.7972022, 103.9832089, 104.0290022, 103.7469611, 103.7649377, 103.8535097, 103.8917746, 1
03.8862529, 103.7435522, 103.7089875, 103.8651056, 103.7173325, 103.8241046, 103.8709039, 103.8701686, 103.8970821, 103.8485095, 103.8403765, 104.0323021, 103.8
424319, 103.8380766, 103.8438992, 103.9473728, 103.899516, 103.8079907, 103.9072046, 103.7861271, 103.8360614, 103.8509134, 103.8663418, 103.8184954, 103.891443
3, 103.8678708, 103.8427696, 103.8397852, 103.8257048, 103.8527531, 103.7561919, 103.9567879, 103.8188845, 103.7308145, 103.8563265, 103.6304833, 103.6767958, 1
03.682541, 103.7889597, 103.8353628]
```

Data Extraction using *Google Maps API*

df_sgNeigh.head (55)

Transform the geolocation data into a new *pandas* dataframe

```
# define the dataframe columns
neighborhood = ['Region','Neighborhood','Latitude','Longitude']

# instantiate the dataframe
df_sgNeigh = pd.DataFrame(columns=neighborhood)
df_sgNeigh
```

Region	Neighborhood	Latitude	Longitude

```
for i in range(0, df_sg.shape[0],1):

    df_sgNeigh = df_sgNeigh.append({'Region': df_sg['Region'][i],
                                    'Neighborhood': df_sg['EName'][i],
                                    'Latitude': lat[i],
                                    'Longitude': long[i],
                                    'Population Density': df_sg['Density'][i] }, ignore_index=True)

df_sgNeigh.head(55)
```

Data Extraction using *Google Maps API*

	Region	Neighborhood	Latitude	Longitude	Population Density
0	North-East	Ang Mo Kio	1.369115	103.845434	12,000
1	East	Bedok	1.323604	103.927340	13,000
2	Central	Bishan	1.352585	103.835212	12,000
3	West	Boon Lay	1.314256	103.709310	3.6
4	West	Bukit Batok	1.359029	103.763680	13,000
5	Central	Bukit Merah	1.281905	103.823918	11,000
6	West	Bukit Panjang	1.377414	103.771950	16,000
7	Central	Bukit Timah	1.329411	103.802078	4,400
8	North	Central Water Catchment	1.355153	103.797202	*
9	East	Changi	1.345010	103.983209	62.3
10	East	Changi Bay	1.321971	104.029002	*
11	West	Choa Chu Kang	1.383980	103.746961	31,000
12	West	Clementi	1.316181	103.764938	9,800
13	Central	Downtown Core	1.286696	103.853510	580
14	Central	Geylang	1.320054	103.891775	12,129
15	North-East	Hougang	1.361218	103.886253	16,000
16	West	Jurong East	1.332857	103.743552	4,600
17	West	Jurong West	1.340390	103.708988	27,000
18	Central	Kallang	1.310033	103.865106	11,000
19	North	Lim Chu Kang	1.430494	103.717332	5.2
20	North	Mandai	1.426007	103.824105	180.2
33	North-East	Punggol	1.398446	103.907205	17,000
34	Central	Queenstown	1.294166	103.786127	4,800.5
35	Central	River Valley	1.295938	103.836061	6,230.5
36	Central	Rochor	1.305135	103.850913	9,034.1
37	North-East	Seletar	1.405163	103.866342	26.3
38	North	Sembawang	1.449111	103.818495	6,203.3
39	North-East	Sengkang	1.386812	103.891443	19,511
40	North-East	Serangoon	1.355357	103.867871	11,945.2
41	North	Simpang	1.444322	103.842770	*
42	Central	Singapore River	1.289526	103.839785	2,842.2
43	Central	Southern Islands	1.253657	103.825705	244
44	Central	Straits View	1.278550	103.852753	*
45	North	Sungei Kadut	1.407382	103.756192	53.2
46	East	Tampines	1.349591	103.956788	12,506.2
47	Central	Tanglin	1.306932	103.818884	2,491.8
48	West	Tengah	1.355519	103.730814	1.4
49	Central	Toa Payoh	1.334304	103.856326	15,298.2
50	West	Tuas	1.294947	103.630483	2.3
51	West	Western Islands	1.247884	103.676796	
52	West	Western Water Catchment	1.347198	103.682541	13
53	North	Woodlands	1.438192	103.788960	18,424
54	North	Yishun	1.430368	103.835363	9,507.2

Data Visualisation using *Folium*

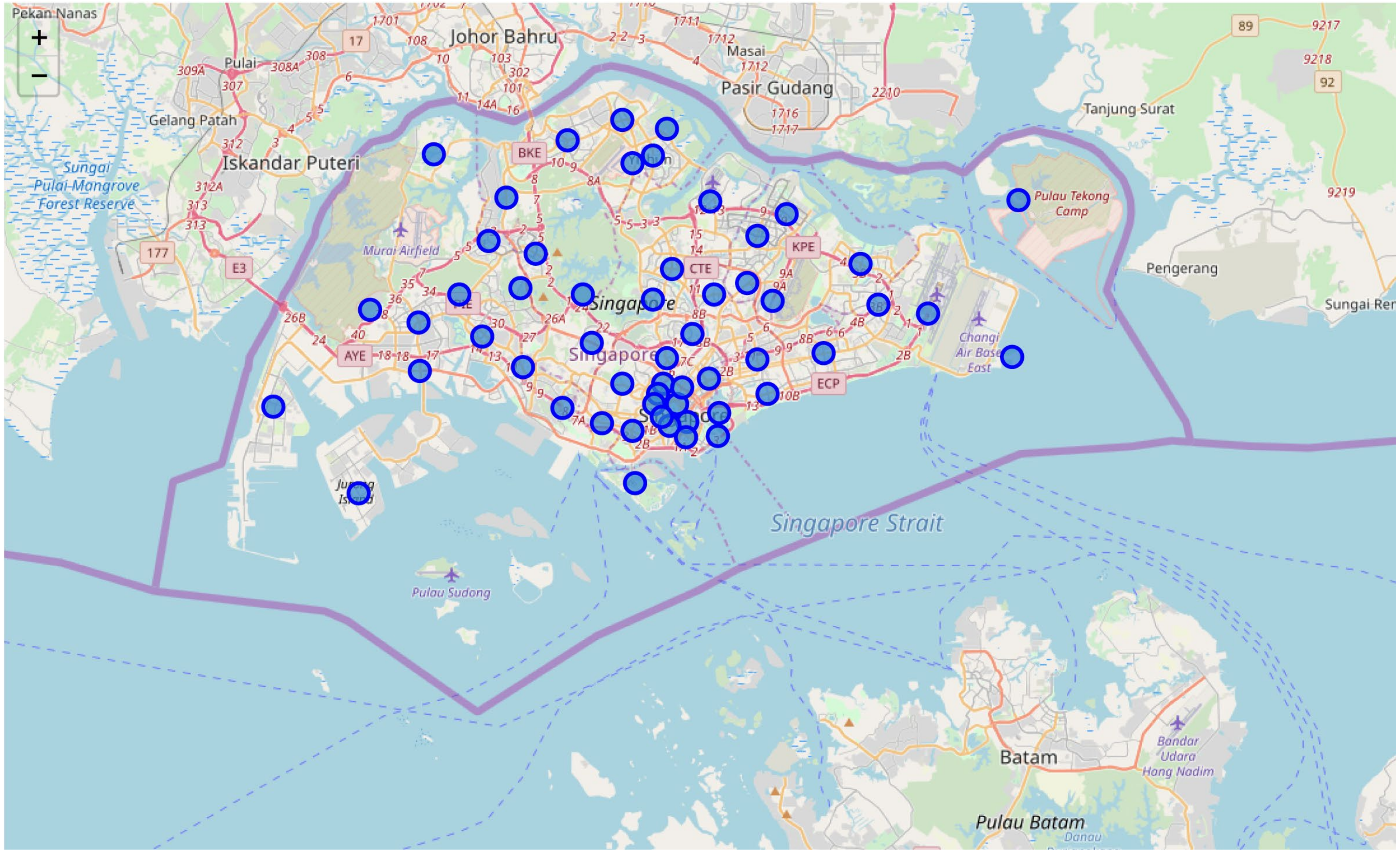
map_sg = folium.Map(location=[SGlat, SGlong], zoom_start=10)

Create a map of Singapore with the 55 neighborhoods superimposed on top.

```
# create map of Singapore using latitude and longitude values
map_sg = folium.Map(location=[SGlatitude, SGlongitude], zoom_start=10)

# add markers to map
for lat, lng, region, neighborhood in zip(df_sgNeigh['Latitude'], df_sgNeigh['Longitude'], df_sgNeigh['Region'], df_sgNeigh['Neighborhood']):
    label = '{}, {}'.format(neighborhood, region)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=8,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_sg)

map_sg
```

Data Visualisation using *Folium*

map_sg : A map of Singapore with the **55** neighborhoods



Data Extraction using *Foursquare API*

To explore the top 10 venues for each of the 55 neighborhood

We start by exploring the first neighborhood – *'Ang Mo Kio'*

```
# type your answer here
LIMIT = 100
radius = 500

url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    df_sgNeigh['Latitude'][0],
    df_sgNeigh['Longitude'][0],
    radius,
    LIMIT)
```

Data Extraction using *Foursquare API*

Create a *Function* using the get_category_type function

We start by exploring the first neighborhood – *'Ang Mo Kio'*

```python
# function that extracts the category of the venue
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']
```

Data Extraction using *Foursquare API*

Create a *getNearbyVenues* function:

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

    venues_list=[]
    for name, lat, lng in zip(neigborhood, 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)
```

Data Extraction using *Foursquare API*

nearby_venues.head()

Returned 63 venues of neighbourhood '*Ang Mo Kio*' from Foursquare

	name	categories	lat	lng
0	Kam Jia Zhuang Restaurant	Asian Restaurant	1.368167	103.844118
1	Old Chang Kee	Snack Place	1.369094	103.848389
2	Subway	Sandwich Place	1.369136	103.847612
3	MOS Burger	Burger Joint	1.369170	103.847831
4	Bun Master	Bakery	1.369242	103.849031

```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
63 venues were returned by Foursquare.
```

Data Extraction using *Foursquare API*

getNearbyVenues(names, latitudes, longitudes, radius=500)

Create a new dataframe to store venues of all 55 neighborhoods

```
sg_venues = getNearbyVenues(names=df_sgNeigh['Neighborhood'],
                            latitudes=df_sgNeigh['Latitude'],
                            longitudes=df_sgNeigh['Longitude']
                           )
```

```
Ang Mo Kio
Bedok
Bishan
Boon Lay
Bukit Batok
Bukit Merah
Bukit Panjang
Bukit Timah
Central Water Catchment
Changi
Changi Bay
Choa Chu Kang
Clementi
Downtown Core
Geylang
Hougang
Jurong East
Jurong West
Kallang
Lim Chu Kang
Mandai
Marina East
Marina South
Marine Parade
Museum
Newton
North-Eastern Islands
Novena
Orchard
Outram
Pasir Ris
Paya Lebar
Pioneer
Punggol
Queenstown
River Valley
Rochor
Seletar
Sembawang
Sengkang
Serangoon
Simpang
Singapore River
Southern Islands
Straits View
Sungei Kadut
Tampines
Tanglin
Tengah
Toa Payoh
Tuas
Western Islands
Western Water Catchment
Woodlands
Yishun
```

Data Extraction using *Foursquare API*

sg_venues.groupby('Neighborhood').count()

Show number of venues returned from each of the 55 neighborhood

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Ang Mo Kio	63	63	63	63	63	63
Bedok	61	61	61	61	61	61
Bishan	49	49	49	49	49	49
Boon Lay	1	1	1	1	1	1
Bukit Batok	34	34	34	34	34	34
Bukit Merah	12	12	12	12	12	12
Bukit Panjang	11	11	11	11	11	11
Bukit Timah	3	3	3	3	3	3
Central Water Catchment	2	2	2	2	2	2
Changi	12	12	12	12	12	12
Changi Bay	5	5	5	5	5	5
Choa Chu Kang	21	21	21	21	21	21
Clementi	59	59	59	59	59	59
Downtown Core	74	74	74	74	74	74

Data Analysis using *One Hot Encoding*

sg_onehot.head()

Show number of venues returned from each of the 55 neighborhood

```
# one hot encoding
sg_onehot = pd.get_dummies(sg_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
sg_onehot['Neighborhood'] = sg_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [sg_onehot.columns[-1]] + list(sg_onehot.columns[:-1])
sg_onehot = sg_onehot[fixed_columns]

sg_onehot.head()
```

	Neighborhood	Accessories Store	Airport	Airport Service	Airport Terminal	American Restaurant	Arcade	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	BBQ Joint	Bagel Shop	Bakery	Bank	Bar	Basketball Court	Bay	Bed & Breakfast	Beer Bar	Beer Garden
0	Ang Mo Kio	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	Ang Mo Kio	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Ang Mo Kio	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Ang Mo Kio	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Ang Mo Kio	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Data Analysis using *One Hot Encoding*

sg_grouped = sg_onehot.groupby('Neighborhood').mean().reset_index()

Group rows by neighbourhood and by taking the mean of the frequency of occurrence of each category

	Neighborhood	Accessories Store	Airport	Airport Service	Airport Terminal	American Restaurant	Arcade	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	BBQ Joint	Bagel Shop	Bakery	Bank	Bar	Basketball Court	Bay	Bed & Breakfast	Beer Bar	Beer Garden	Bistro
0	Ang Mo Kio	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.015873	0.000000	0.000000	0.000000	0.015873	0.015873	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
1	Bedok	0.000000	0.000000	0.000000	0.000000	0.016393	0.000000	0.000000	0.000000	0.000000	0.016393	0.000000	0.000000	0.000000	0.032787	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
2	Bishan	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.020408	0.000000	0.000000	0.000000	0.040816	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
3	Boon Lay	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
4	Bukit Batok	0.000000	0.000000	0.000000	0.000000	0.029412	0.000000	0.000000	0.000000	0.029412	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.029412	0.000000	0.000000	0.00	0.000000	0.029412	0.029412
5	Bukit Merah	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
6	Bukit Panjang	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
7	Bukit Timah	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
8	Central Water Catchment	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
9	Changi	0.000000	0.083333	0.083333	0.166667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
10	Changi Bay	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
11	Choa Chu Kang	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.047619	0.000000	0.000000	0.000000	0.047619	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
12	Clementi	0.000000	0.000000	0.000000	0.000000	0.016949	0.000000	0.000000	0.000000	0.016949	0.050847	0.000000	0.016949	0.000000	0.033898	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000
13	Downtown Core	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.013514	0.000000	0.000000	0.027027	0.000000	0.013514	0.000000	0.000000	0.000000	0.040541	0.000000	0.013514	0.00	0.000000	0.000000	0.000000
14	Geylang	0.000000	0.000000	0.000000	0.000000	0.020408	0.000000	0.000000	0.000000	0.000000	0.061224	0.000000	0.000000	0.000000	0.061224	0.000000	0.000000	0.020408	0.000000	0.00	0.000000	0.000000	0.000000

Data Analysis using *One Hot Encoding*

num_top_venues = 5

Print each neighbourhood along with the top 5 most common venues

```
----Ang Mo Kio----
                  venue  freq
0           Coffee Shop  0.10
1            Food Court  0.08
2  Fast Food Restaurant  0.06
3           Supermarket  0.05
4          Dessert Shop  0.05
```

```
----Boon Lay----
               venue  freq
0   Botanical Garden   1.0
1  Accessories Store   0.0
2  Other Repair Shop   0.0
3              Motel   0.0
4      Movie Theater   0.0
```

```
----Museum----
                 venue  freq
0                Hotel  0.10
1                 Café  0.07
2  Japanese Restaurant  0.07
3    French Restaurant  0.03
4       Ice Cream Shop  0.03
```

```
----Bedok----
                venue  freq
0         Coffee Shop  0.07
1     Sushi Restaurant  0.05
2          Food Court  0.05
3  Chinese Restaurant  0.05
4         Supermarket  0.05
```

```
----Bukit Batok----
                venue  freq
0  Italian Restaurant  0.09
1        Shopping Mall  0.06
2                 Café  0.06
3          Supermarket  0.06
4                  Gym  0.06
```

```
----Newton----
                venue  freq
0  Chinese Restaurant  0.14
1             Wine Bar  0.14
2         Noodle House  0.14
3  Seafood Restaurant  0.14
4                 Café  0.14
```

```
----Bishan----
                venue  freq
0                 Café  0.12
1  Chinese Restaurant  0.10
2     Thai Restaurant  0.08
3   Indian Restaurant  0.06
4      Ice Cream Shop  0.06
```

```
----Bukit Merah----
                 venue  freq
0   Chinese Restaurant  0.25
1    Convenience Store  0.17
2          Coffee Shop  0.08
3   Seafood Restaurant  0.08
4  Japanese Restaurant  0.08
```

```
----North-Eastern Islands----
         venue  freq
0       School  0.14
1   Food Court  0.14
2        Plaza  0.14
3       Island  0.14
4  Coffee Shop  0.14
```

The Museum neighborhood is the only one with hotel as top venues

Data Analysis using *One Hot Encoding*

neighborhoods_venues_sorted.head()

Create a new dataframe that display the top 5 venues for each neighborhood

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Ang Mo Kio	Coffee Shop	Food Court	Fast Food Restaurant	Supermarket	Dessert Shop
1	Bedok	Coffee Shop	Supermarket	Sushi Restaurant	Food Court	Chinese Restaurant
2	Bishan	Café	Chinese Restaurant	Thai Restaurant	Ice Cream Shop	Indian Restaurant
3	Boon Lay	Botanical Garden	Yoga Studio	Fast Food Restaurant	Frozen Yogurt Shop	Fried Chicken Joint
4	Bukit Batok	Italian Restaurant	Gym	Café	Ice Cream Shop	Indian Restaurant
5	Bukit Merah	Chinese Restaurant	Convenience Store	Residential Building (Apartment / Condo)	Food Court	Seafood Restaurant
6	Bukit Panjang	Food Court	Park	Miscellaneous Shop	Grocery Store	Dance Studio
7	Bukit Timah	Pool	Food	Gym / Fitness Center	Farmers Market	Fried Chicken Joint
8	Central Water Catchment	Trail	Yoga Studio	Farmers Market	Frozen Yogurt Shop	Fried Chicken Joint
9	Changi	Airport Terminal	Bus Station	Movie Theater	Tunnel	Café
10	Changi Bay	Boat or Ferry	Military Base	Field	Fruit & Vegetable Store	Frozen Yogurt Shop
11	Choa Chu Kang	Food Court	Coffee Shop	Bus Station	Fast Food Restaurant	Bubble Tea Shop
12	Clementi	Coffee Shop	Noodle House	Chinese Restaurant	Asian Restaurant	Snack Place
13	Downtown Core	Cocktail Bar	Italian Restaurant	Performing Arts Venue	Salad Place	Japanese Restaurant
14	Geylang	Fast Food Restaurant	Asian Restaurant	Bakery	Supermarket	Shopping Mall
15	Hougang	Chinese Restaurant	Coffee Shop	Noodle House	Café	Asian Restaurant

K-Mean Clustering

Run k-means to cluster the ***55*** neighbourhood into ***10*** clusters.
10 clusters are used because ***3*** clusters do not produce significant differences as most neighbourhood has very similar top ***10*** venues such as coffee shop, food court, restaurants and café.

In [413]:
```
# set number of clusters
kclusters = 10

sg_grouped_clustering = sg_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(sg_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:54]
```

Out[413]:
```
array([1, 1, 1, 4, 1, 1, 1, 1, 3, 1, 6, 1, 1, 1, 1, 1, 1, 1, 1, 8, 1, 7, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 5, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 5, 1,
       1, 1, 9, 1, 1, 1], dtype=int32)
```

Data Analysis using *K-Means Clustering*

sg_merged.head()

Create new dataframe that includes the Cluster Labels and the top 5 venues for each of the 55 neighborhood.

```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

sg_merged = df_sgNeigh

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
sg_merged = sg_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

# Replace NaN with '0'
sg_merged['Cluster Labels'].fillna(value=0, method=None, axis=None, inplace=True)
sg_merged.head() # check the last columns!
```

	Region	Neighborhood	Latitude	Longitude	Population Density	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	North-East	Ang Mo Kio	1.369115	103.845434	12,000	1.0	Coffee Shop	Food Court	Fast Food Restaurant	Supermarket	Dessert Shop
1	East	Bedok	1.323604	103.927340	13,000	1.0	Coffee Shop	Supermarket	Sushi Restaurant	Food Court	Chinese Restaurant
2	Central	Bishan	1.352585	103.835212	12,000	1.0	Café	Chinese Restaurant	Thai Restaurant	Ice Cream Shop	Indian Restaurant
3	West	Boon Lay	1.314256	103.709310	3.6	4.0	Botanical Garden	Yoga Studio	Fast Food Restaurant	Frozen Yogurt Shop	Fried Chicken Joint
4	West	Bukit Batok	1.359029	103.763680	13,000	1.0	Italian Restaurant	Gym	Café	Ice Cream Shop	Indian Restaurant

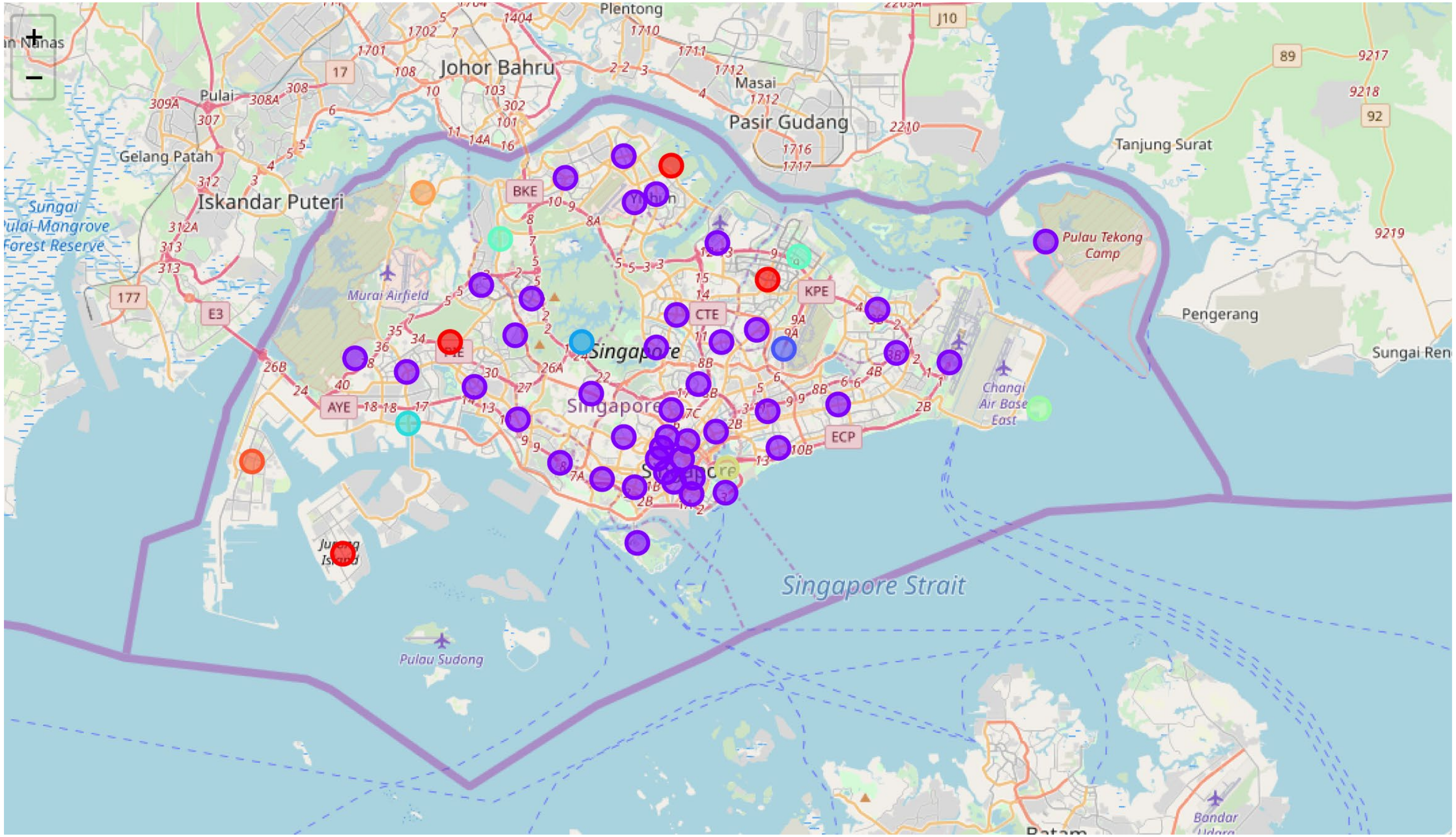
Data Visualisation using *Folium*

map_sg : A map of Singapore with the **10** Clusters

```
# create map
map_clusters = folium.Map(location=[SGlatitude, SGlongitude], zoom_start=11)

# 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(sg_merged['Latitude'], sg_merged['Longitude'], sg_merged['Neighborhood'], sg_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=8,
        popup=label,
        color=rainbow[int(cluster)-1],
        fill=True,
        fill_color=rainbow[int(cluster)-1],
        fill_opacity=0.6).add_to(map_clusters)

map_clusters
```

Data Visualisation using *Folium*

map_sg : A map of Singapore with the **10** Clusters



Data Analysis using *K-Means Clustering*

Analyse the resulting 10 clusters:

Create new dataframe that includes the Cluster Labels and the top 5 venues for each of the 55 neighborhood.

Cluster Labels == 0

```
In [416]: sg_merged.loc[sg_merged['Cluster Labels'] == 0, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```

Out[416]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
39	Sengkang	0.0	Bus Station	Grocery Store	Food Stand	General College & University	Metro Station
41	Simpang	0.0	NaN	NaN	NaN	NaN	NaN
48	Tengah	0.0	NaN	NaN	NaN	NaN	NaN
51	Western Islands	0.0	NaN	NaN	NaN	NaN	NaN

The resultant 'NaN' is due to several reasons:

1. Simpang is not yet developed and currently still a forested area.
2. Tengah is a new town that is still under construction.
3. Western island (Jurong island to be specific) is an full fledged industrial area.

Data Analysis using *K-Means Clustering*

Analyse the resulting 10 clusters:

Create new dataframe that includes the Cluster Labels and the top 5 venues for each of the 55 neighborhood.

Cluster Labels == 1

In [417]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 1, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```

Out[417]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Ang Mo Kio	1.0	Coffee Shop	Food Court	Fast Food Restaurant	Supermarket	Dessert Shop
1	Bedok	1.0	Coffee Shop	Supermarket	Sushi Restaurant	Food Court	Chinese Restaurant
2	Bishan	1.0	Café	Chinese Restaurant	Thai Restaurant	Ice Cream Shop	Indian Restaurant
4	Bukit Batok	1.0	Italian Restaurant	Gym	Café	Ice Cream Shop	Indian Restaurant
5	Bukit Merah	1.0	Chinese Restaurant	Convenience Store	Residential Building (Apartment / Condo)	Food Court	Seafood Restaurant
6	Bukit Panjang	1.0	Food Court	Park	Miscellaneous Shop	Grocery Store	Dance Studio
7	Bukit Timah	1.0	Pool	Food	Gym / Fitness Center	Farmers Market	Fried Chicken Joint
9	Changi	1.0	Airport Terminal	Bus Station	Movie Theater	Tunnel	Café

This is the cluster with most neighborhoods - 42 out of 55

Data Analysis using *K-Means Clustering*

Cluster Labels == 2, 3, 4 and 5

In [418]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 2, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```
Out[418]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
31	Paya Lebar	2.0	Military Base	Yoga Studio	Fast Food Restaurant	Frozen Yogurt Shop	Fried Chicken Joint

In [420]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 3, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```
Out[420]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
8	Central Water Catchment	3.0	Trail	Yoga Studio	Farmers Market	Frozen Yogurt Shop	Fried Chicken Joint

In [421]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 4, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```
Out[421]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
3	Boon Lay	4.0	Botanical Garden	Yoga Studio	Fast Food Restaurant	Frozen Yogurt Shop	Fried Chicken Joint

In [422]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 5, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```
Out[422]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
33	Punggol	5.0	Bus Station	High School	Chinese Restaurant	Yoga Studio	Fast Food Restaurant
45	Sungei Kadut	5.0	Bus Station	Café	Furniture / Home Store	Chinese Restaurant	Fast Food Restaurant

Data Analysis using *K-Means Clustering*

Cluster Labels == 6, 7, 8 and 9

In [423]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 6, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```

Out[423]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
10	Changi Bay	6.0	Boat or Ferry	Military Base	Field	Fruit & Vegetable Store	Frozen Yogurt Shop

In [424]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 7, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```

Out[424]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
21	Marina East	7.0	Golf Course	Park	Yoga Studio	Farmers Market	Fried Chicken Joint

In [425]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 8, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```

Out[425]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
19	Lim Chu Kang	8.0	Farm	Farmers Market	Theme Park Ride / Attraction	Frozen Yogurt Shop	Fried Chicken Joint

In [426]:
```
sg_merged.loc[sg_merged['Cluster Labels'] == 9, sg_merged.columns[[1] + list(range(5, sg_merged.shape[1]))]]
```

Out[426]:

	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
50	Tuas	9.0	Food Court	Asian Restaurant	Yoga Studio	Farmers Market	Frozen Yogurt Shop

Data Analysis using *K-Means Clustering*

Evaluating the resulting 10 clusters:

10 Clusters are chosen to better distinct the neighborhoods as it is initially found that Singapore's Neigborhoods are all too similar. Most Neighborhoods fall into Cluster '0' which is a cluster that is categorised by *eateries*. ***<u>The 1st most common venues are 'Coffee shop', 'Restaurant', 'Food court', 'Bar' etc.</u>*** Areas with coffee shops are *residential neighborhoods* as it is common to have coffee shops within HDB estates. Since *80%* of Singapore residential typologies are HDBs, Cluster '0' consists of most neighborhoods out of the 55 neighborhoods.

Cluster '0' will be the chosen cluster to locate the cafe as the other clusters are less populated areas that are do not have enough human traffic to generate consistent revenues.

The ***Museum*** neigborhood is ***<u>the only neighborhood in Cluster '0' that has hotels as the first most common venue.</u>*** Therefore the second part of the analysis will use the Museum's *geographically coordinates - (Lat 1.296615, Long 103.848510)* ***as the epicentre to query*** for potential location for our cafe.

Query request using *Foursquare API*

Search_query = ‘Hotel’

Search for the number of hotels and see its distribution in Museum neighborhood.

```
search_query = 'Hotel'
radius = 500
LIMIT = 100
print(search_query + ' .... OK!')
```

```
Hotel .... OK!
```

Send the GET request to *Foursquare*

```
In [448]: results = requests.get(url).json()
          results
```


```
Out[448]: {'meta': {'code': 200, 'requestId': '5cf93ecddb04f52f5edd7a52'},
  'response': {'venues': [{'categories': [{'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/travel/hotel_',
        'suffix': '.png'},
       'id': '4bf58dd8d48988d1fa931735',
       'name': 'Hotel',
       'pluralName': 'Hotels',
       'primary': True,
       'shortName': 'Hotel'}],
     'hasPerk': False,
     'id': '4b05880af964a520aead22e3',
     'location': {'address': '9 Bras Basah Road',
      'cc': 'SG',
      'city': 'Singapore',
      'country': 'Singapore',
      'distance': 227,
      'formattedAddress': ['9 Bras Basah Road', '189559', 'Singapore'],
      'labeledLatLngs': [{'label': 'display',
        'lat': 1.2985851758211133,
        'lng': 103.84906056770934}],
```

Query request using *Foursquare API*

dataframe_filtered

Filter the JSON an transform the data into a *pandas* dataframe

	name	categories	address	cc	city	country	crossStreet	distance	formattedAddress	labeledLatLngs	lat	lng
0	Rendezvous Hotel Singapore	Hotel	9 Bras Basah Road	SG	Singapore	Singapore	NaN	227	[9 Bras Basah Road, 189559, Singapore]	[{'lng': 103.84906056770934, 'lat': 1.29858517...	1.298585	103.849061
1	Carlton Hotel	Hotel	76 Bras Basah Rd	SG	Singapore	Singapore	NaN	471	[76 Bras Basah Rd, 189558, Singapore]	[{'lng': 103.85265190696873, 'lat': 1.29571376...	1.295714	103.852652
2	JW Marriott Hotel Singapore South Beach	Hotel	30 Beach Road, Access Via Nicoll Highway	SG	Singapore	Singapore	NaN	844	[30 Beach Road, Access Via Nicoll Highway, 189...	[{'lng': 103.85579288005829, 'lat': 1.29446877...	1.294469	103.855793
3	Hotel Bencoolen	Hotel	47 Bencoolen St.	SG	Singapore	Singapore	NaN	333	[47 Bencoolen St., 189626, Singapore]	[{'lng': 103.850186415232, 'lat': 1.2991005826...	1.299101	103.850186
4	Strand Hotel	Hotel	25 Bencoolen St	SG	Singapore	Singapore	NaN	259	[25 Bencoolen St, 189619, Singapore]	[{'lng': 103.84995898321019, 'lat': 1.29844189...	1.298442	103.849959
5	Marina Bay Sands Hotel	Hotel	10 Bayfront Ave.	SG	Singapore	Singapore	NaN	1996	[10 Bayfront Ave., 018956, Singapore]	NaN	1.283096	103.860296
6	Hotel G Singapore	Hotel	200 Middle Road	SG	Singapore	Singapore	NaN	558	[200 Middle Road, 188980, Singapore]	[{'lng': 103.85147545152623, 'lat': 1.30066073...	1.300661	103.851475
7	Mercure Hotel	Hotel	122 Middle Road	SG	Singapore	Singapore	NaN	610	[122 Middle Road, 188973, Singapore]	[{'lng': 103.85317286291435, 'lat': 1.29950306...	1.299503	103.853173
8	V Hotel Bencoolen	Hotel	48 Bencoolen Street #01-01	SG	Bugis	Singapore	NaN	362	[48 Bencoolen Street #01-01, 189627, Singapore]	[{'lng': 103.85058095759739, 'lat': 1.29912982...	1.299130	103.850581
9	Grand Park City Hall Hotel	Hotel	10 Coleman St.	SG	Singapore	Singapore	NaN	517	[10 Coleman St., 179809, Singapore]	[{'lng': 103.85031946942136, 'lat': 1.29232969...	1.292330	103.850319
10	Peninsula Excelsior Hotel	Hotel	5 Coleman St.	SG	Singapore	Singapore	Opp Grand Park Hotel	551	[5 Coleman St. (Opp Grand Park Hotel), 179805,...	[{'lng': 103.85016015585542, 'lat': 1.29194450...	1.291945	103.850160

Query request using *Foursquare API*

dataframe_filtered.name

See the list of all ***50 hotels*** nearby ***Musuem Neighborhood*** returned from *Foursquare*

```
0                Rendezvous Hotel Singapore
1                             Carlton Hotel
2     JW Marriott Hotel Singapore South Beach
3                           Hotel Bencoolen
4                              Strand Hotel
5                        Marina Bay Sands Hotel
6                         Hotel G Singapore
7                             Mercure Hotel
8                         V Hotel Bencoolen
9                 Grand Park City Hall Hotel
10                 Peninsula Excelsior Hotel
11                           Studio M Hotel
12                        Hotel Fort Canning
13                        The Fullerton Hotel
14        Hotel Jen Orchardgateway Singapore
15                           Concorde Hotel
16                           Victoria Hotel
17                            Raffles Hotel
18                      Hotel Grand Central
19              Tower 3 Marina Bay Sands Hotel
20                       Village Hotel Bugis
21          Fragrance Hotel Selegie Singapore
22                      Hotel Grand Pacific
23                                Hotel Kai
24                        Hotel 81 Dickson
25                  Fragrance Hotel (Bugis)
26                 Hotel Chancellor@Orchard
27                     Hotel Royal @ Queens
28                          Fragrance Hotel
29                         Summerview Hotel
30                       Amaris Hotel Bugis
31                            Hotel Innotel
32                         Park Regis Hotel
33      Bus Stop 01012 (Hotel Grand Pacific)
34             Hotel Intercontinental Gym
35          Grand Copthorne Waterfront Hotel
36             Tower 1 Marina Bay Sands Hotel
37                      Hotel 81 - Selegie
38                      Raffles Hotel Arcade
39                   The Fullerton Bay Hotel
40         Singapore Marriott Tang Plaza Hotel
41      Health Club @ Carlton Hotel Singapore
42            Hotel Fort Canning Parking Garage
43                    Hotel Supreme (原首酒店)
44                          Porcelain Hotel
45    Bus Stop 08138 (Concorde Hotel S'pore)
46                               Hotel Nuve
47             Vault @ Intercontinental Hotel
48              Hotel Clover | 33 Jalan Sultan
49                           Hotel 81 Bugis
```

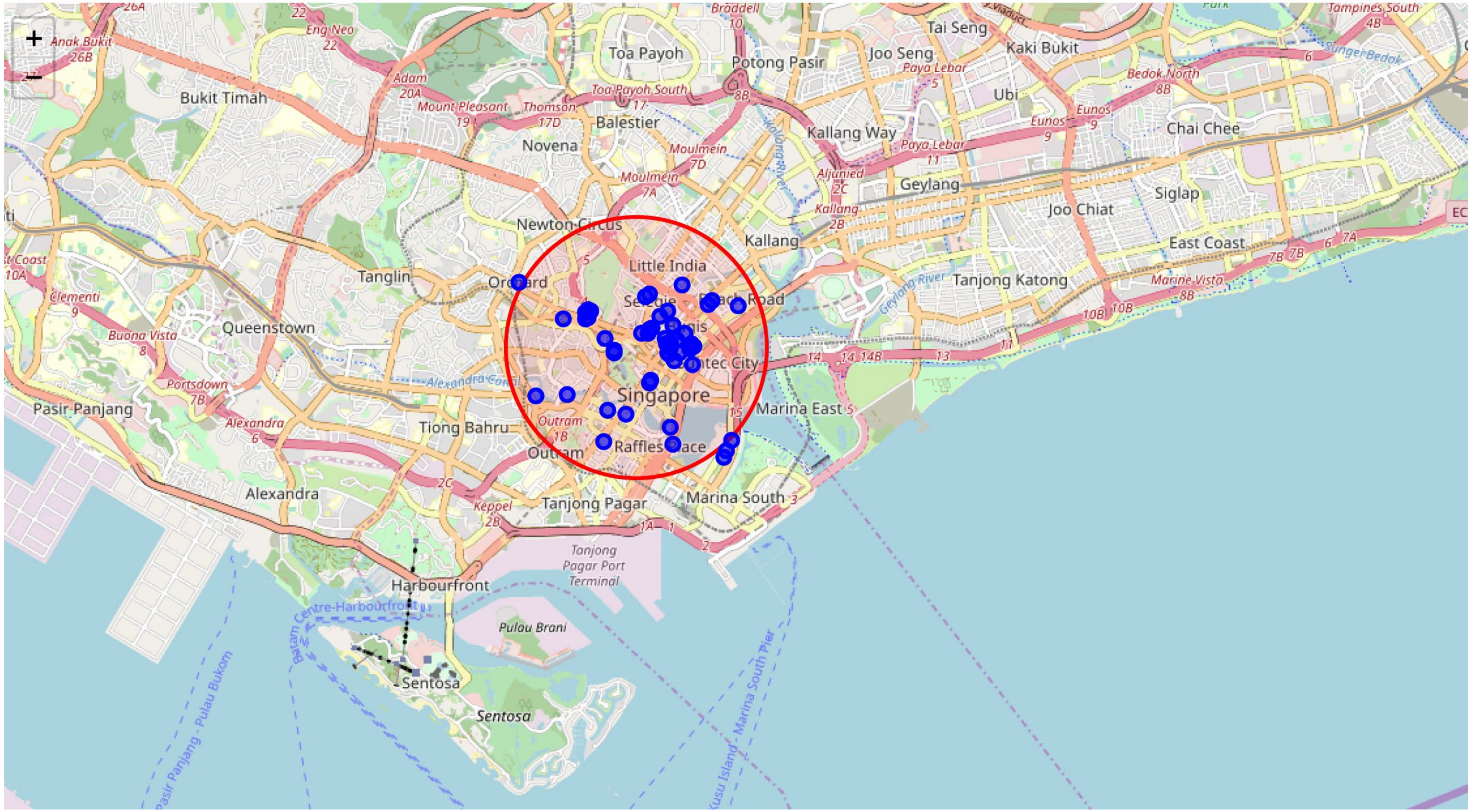
Data Visualisation using *Folium*

venues_map: A map of Singapore with the **50** Hotel locations within 500 metres radius of Museum Neighborhood

```
venues_map = folium.Map(location=[SGlatitude, SGlongitude], zoom_start=13) # generate map centred around the Conrad Hotel

# add a red circle radius with museum neighborhood as epicentre
folium.features.CircleMarker(
    [1.296615, 103.848510],
    radius=100,
    color='red',
    popup='Museum Neighborhood Radius',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.1
).add_to(venues_map)

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe_filtered.categories):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)

# display map
venues_map
```

Data Visualisation using *Folium*

venues_map: A map of Singapore with the **50** Hotel locations within 500 metres radius of Museum Neighborhood



Conclusion

Recommendation

To Recommend the best place to open a Cafe in Singapore.

The decision is to locate the cafe within the ***500 metres radius*** of the ***Museum Neighborhood*** due to the ***large number of hotels, restaurants, bars and other eating places***. It is also located within walking distance to Singapore's attractions such as Chinatown, Marina Bay Sands, Singapore Flyer, National Gallery and many more.