

In [6]:

Libraries imported.

In [1]:

```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analysis
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", None)

import json # library to handle JSON files

from geopy.geocoders import Nominatim # convert an address into latitude
and longitude values
import geocoder # to get coordinates

import requests # library to handle requests
from bs4 import BeautifulSoup # library to parse HTML and XML documents

from pandas.io.json import json_normalize # tranform JSON file into a p
andas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

import folium # map rendering library

print("Libraries imported.")
```

Libraries imported.

In [7]:

```
data = requests.get("https://en.wikipedia.org/wiki/Category:Suburbs_in_
```

```
Kuala_Lumpur").text
```

```
In [8]: soup = BeautifulSoup(data, 'html.parser')
```

```
In [9]: neighborhoodList = []
```

```
In [10]: for row in soup.find_all("div", class_="mw-category")[0].findAll("li"):
          neighborhoodList.append(row.text)
```

```
In [11]: kl_df = pd.DataFrame({"Neighborhood": neighborhoodList})
          kl_df.head()
```

Out[11]:

	Neighborhood
0	Alam Damai
1	Ampang, Kuala Lumpur
2	Bandar Menjalara
3	Bandar Sri Permaisuri
4	Bandar Tasik Selatan

```
In [12]: kl_df.shape
```

Out[12]: (71, 1)

```
In [13]: def get_latlng(neighborhood):
          lat_lng_coords = None

          while(lat_lng_coords is None):
              g = geocoder.arcgis('{}, Kuala Lumpur, Malaysia'.format(neighbo
rhood))
              lat_lng_coords = g.latlng
          return lat_lng_coords
```

```
In [29]: coords = [ get_latlng(neighborhood) for neighborhood in kl_df["Neighborhood"].tolist() ]
```

```
In [26]: coords
```

```
Out[26]: [[3.05769000000000364, 101.743880000000005],  
[3.148494115588384, 101.69672876508707],  
[3.19035000000000236, 101.625450000000006],  
[3.10391000000000417, 101.712260000000007],  
[3.0727500000000042, 101.714610000000005],  
[3.082800000000002, 101.722810000000004],  
[3.12920000000000257, 101.678440000000008],  
[3.12920000000000257, 101.678440000000008],  
[3.1110200000000053, 101.662830000000004],  
[3.0618700000000056, 101.746750000000008],  
[3.1478900000000075, 101.694050000000006],  
[3.129160000000007, 101.684060000000004],  
[3.1477700000000037, 101.708550000000006],  
[3.05781000000000745, 101.689650000000009],  
[3.14348000000000678, 101.644330000000008],  
[3.1511418740671178, 101.69937468268016],  
[3.1292900000000026, 101.698960000000006],  
[3.173810000000006, 101.682760000000009],  
[3.0618700000000056, 101.746750000000008],  
[3.1635900000000056, 101.698110000000004],  
[3.14797000000000435, 101.667950000000008],  
[3.136442497863743, 101.69029641203808],  
[3.1419057131529953, 101.67967833034047],  
[3.1566851613075664, 101.69807685476978],  
[3.0833000000000065, 101.704380000000007],  
[3.13637000000000563, 101.685640000000003],  
[3.2016300000000023, 101.721070000000005],  
[3.1329027727420717, 101.72467923770417],  
[3.1800248500092656, 101.67783444508486],  
[3.2095000000000048, 101.658740000000008],  
[3.1654600000000053, 101.710280000000007],  
[3.1664000000000067, 101.730460000000005],  
[3.20943000000000544, 101.693180000000004],
```

[3.2175000000000296, 101.637630000000006],
[3.207768158327776, 101.64516652936956],
[3.1171300000000315, 101.673840000000004],
[3.0907400000000393, 101.677330000000004],
[3.121202171543197, 101.66389917991445],
[3.1478900000000075, 101.694050000000006],
[3.15926000000000743, 101.698340000000009],
[3.1478900000000075, 101.694050000000006],
[3.1653200000000065, 101.652430000000004],
[3.0947600000000065, 101.667470000000004],
[3.1335400000000039, 101.713070000000007],
[3.1258432855470684, 101.71850881971983],
[3.08102000000000236, 101.697240000000008],
[3.1863900000000074, 101.668100000000004],
[3.1799164410297633, 101.72143714871991],
[3.17508000000000368, 101.693050000000008],
[3.1881600000000039, 101.704150000000008],
[3.191802843003158, 101.74007037312064],
[3.12457000000000625, 101.735970000000007],
[3.1622000000000041, 101.650360000000003],
[3.07260000000000226, 101.682520000000007],
[3.04997000000000303, 101.706030000000006],
[3.20066000000000276, 101.633370000000007],
[3.08269000000000705, 101.736890000000007],
[3.10297000000000275, 101.684710000000005],
[3.2121600000000004, 101.715400000000005],
[3.06908000000000042, 101.742870000000004],
[3.22357000000000052, 101.723990000000007],
[3.09359000000000063, 101.728370000000004],
[3.2100497379521316, 101.63450794879562],
[3.193940000000000546, 101.705730000000007],
[3.190070000000000484, 101.652930000000008],
[3.087070000000000397, 101.736810000000005],
[3.152830000000000513, 101.622710000000004],
[3.15770000000000034, 101.724520000000004],
[3.2224000000000005, 101.671730000000008],
[3.180730000000000396, 101.703210000000007],
[3.203910000000000644, 101.737190000000006]]

```
In [36]: df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
```

```
In [37]: kl_df['Latitude'] = df_coords['Latitude']  
kl_df['Longitude'] = df_coords['Longitude']
```

```
In [35]: print(kl_df.shape)  
kl_df
```

(71, 3)

Out[35]:

	Neighborhood	Latitude	Longitude
0	Alam Damai	3.057690	101.743880
1	Ampang, Kuala Lumpur	3.148494	101.696729
2	Bandar Menjalara	3.190350	101.625450
3	Bandar Sri Permaisuri	3.103910	101.712260
4	Bandar Tasik Selatan	3.072750	101.714610
5	Bandar Tun Razak	3.082800	101.722810
6	Bangsar	3.129200	101.678440
7	Bangsar Park	3.129200	101.678440
8	Bangsar South	3.111020	101.662830
9	Batu 11 Cheras	3.061870	101.746750
10	Batu, Kuala Lumpur	3.147890	101.694050
11	Brickfields	3.129160	101.684060
12	Bukit Bintang	3.147770	101.708550
13	Bukit Jalil	3.057810	101.689650
14	Bukit Kiara	3.143480	101.644330
15	Bukit Nanas	3.151142	101.699375
16	Bukit Petaling	3.129290	101.698960

	Neighborhood	Latitude	Longitude
17	Bukit Tunku	3.173810	101.682760
18	Cheras, Kuala Lumpur	3.061870	101.746750
19	Chow Kit	3.163590	101.698110
20	Damansara Heights	3.147970	101.667950
21	Damansara Town Centre	3.136442	101.690296
22	Damansara, Kuala Lumpur	3.141906	101.679678
23	Dang Wangi	3.156685	101.698077
24	Desa Petaling	3.083300	101.704380
25	Federal Hill, Kuala Lumpur	3.136370	101.685640
26	Happy Garden	3.201630	101.721070
27	Jalan Cochrane, Kuala Lumpur	3.132903	101.724679
28	Jalan Duta	3.180025	101.677834
29	Jinjang	3.209500	101.658740
30	Kampung Baru, Kuala Lumpur	3.165460	101.710280
31	Kampung Datuk Keramat	3.166400	101.730460
32	Kampung Padang Balang	3.209430	101.693180
33	Kepong	3.217500	101.637630
34	Kepong Baru	3.207768	101.645167
35	KL Eco City	3.117130	101.673840
36	Kuchai Lama	3.090740	101.677330
37	Lembah Pantai	3.121202	101.663899
38	Maluri	3.147890	101.694050
39	Medan Tuanku	3.159260	101.698340
40	Miharja	3.147890	101.694050
41	Mont Kiara	3.165320	101.652430

	Neighborhood	Latitude	Longitude
42	Pantai Dalam	3.094760	101.667470
43	Pudu, Kuala Lumpur	3.133540	101.713070
44	Putrajaya	3.125843	101.718509
45	Salak South	3.081020	101.697240
46	Segambut	3.186390	101.668100
47	Semarak	3.179916	101.721437
48	Sentul, Kuala Lumpur	3.175080	101.693050
49	Setapak	3.188160	101.704150
50	Setiawangsa	3.191803	101.740070
51	Shamelin	3.124570	101.735970
52	Sri Hartamas	3.162200	101.650360
53	Sri Petaling	3.072600	101.682520
54	Sungai Besi	3.049970	101.706030
55	Taman Bukit Maluri	3.200660	101.633370
56	Taman Connaught	3.082690	101.736890
57	Taman Desa	3.102970	101.684710
58	Taman Ibukota	3.212160	101.715400
59	Taman Len Seng	3.069080	101.742870
60	Taman Melati	3.223570	101.723990
61	Taman Midah	3.093590	101.728370
62	Taman OUG	3.210050	101.634508
63	Taman P. Ramlee	3.193940	101.705730
64	Taman Sri Sinar	3.190070	101.652930
65	Taman Taynton View	3.087070	101.736810
66	Taman Tun Dr Ismail	3.152830	101.622710

	Neighborhood	Latitude	Longitude
67	Taman U-Thant	3.157700	101.724520
68	Taman Wahyu	3.222400	101.671730
69	Titivangsa	3.180730	101.703210
70	Wangsa Maju	3.203910	101.737190

```
In [38]: kl_df.to_csv("kl_df.csv", index=False)
```

```
In [39]: address = 'Kuala Lumpur, Malaysia'

geolocator = Nominatim(user_agent="my-application")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Kuala Lumpur, Malaysiae {}, {}'.format(latitude, longitude))
```

The geograpical coordinate of Kuala Lumpur, Malaysiae 3.1516964, 101.6942371.

```
In [40]: map_kl = folium.Map(location=[latitude, longitude], zoom_start=11)

for lat, lng, neighborhood in zip(kl_df['Latitude'], kl_df['Longitude'],
kl_df['Neighborhood']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_kl)
```


map_kl

Out[40]:



In [41]: `map_kl.save('map_kl.html')`

In [91]: `CLIENT_ID = 'your Foursquare ID'`
`CLIENT_SECRET = 'your Foursquare Secret'`
`VERSION = '20180605'`

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

Your credentails:
CLIENT_ID: your Foursquare ID
CLIENT_SECRET:your Foursquare Secret

```
In [108]: kl_df.loc[0, 'Neighborhood']
```

```
Out[108]: 'Alam Damai'
```

```
In [115]: neighborhood_latitude = kl_df.loc[0, 'Latitude'] # neighborhood latitude value
neighborhood_longitude = kl_df.loc[0, 'Longitude'] # neighborhood longitude value

neighborhood_name = kl_df.loc[0, 'Neighborhood'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.format(neighborhood_name,
                                                                neighborhood_latitude,
                                                                neighborhood_longitude))
```

```
Latitude and longitude values of Alam Damai are 3.0576900000000364, 10
1.743880000000005.
```

```
In [110]: results = requests.get(url).json()
results
```

```
Out[110]: {'meta': {'code': 400,
                    'errorType': 'invalid_auth',
                    'errorDetail': 'Missing access credentials. See https://developer.foursquare.com/docs/api/configuration/authentication for details.',
                    'requestId': '5ee13470e6573d7f9e6c9438'},
           'response': {}}
```

```
In [117]: print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-117-8d3c8155afff> in <module>
```

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

NameError: name 'nearby_venues' is not defined

```
In [25]: def getNearbyVenues(names, latitudes, longitudes, radius=2000):

    venues=[]
    for name, lat, lng in zip(names,l_df['Latitude'], kl_df['Longitude'], kl_df['Neighborhood']):
        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.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)
```