London-Paris-compare

Introduction and Business Problem

London and Paris are two of the most ancient city in the world. This project is aim to try to analysis the two cities, more specifically I aimed to analysis how the cities can be separated to several section(cluster) with similar shops and facilities. By doing so, I believe we can have betting understanding of the two cities.

The Reason for choosing London and Paris is because both cities have a lot of data available to the public and make it very easy to collect data for both of the cities.

Data Description

London

I get the data from https://en.wikipedia.org/wiki/List_of_London_boroughs

The dataframe contain data of London consist:

1. Borough : Name of Borough
2. Latitude : Latitude
3. Longitude : Longitude

Paris

I get the data from https://en.wikipedia.org/wiki/Arrondissements_of_Paris and from https://www.data.gouv.fr/fr/datasets/r/e88c6fda-1d09-42a0-a069-606d3259114e.

I preprocess the data into a cvs file already.

The dataframe contain data of Paris consist:

1. District : Name of District
2. Latitude : Latitude
3. Longitude : Longitude

Foursquare API Data

I will get data from foursquare API to analysis each distrinct in London and in Paris

In [1]:
```
import requests
import pandas as pd
import numpy as np
import urllib.request
from geopy.geocoders import Nominatim
import folium
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
```

In [2]:
```
CLIENT_ID = '32MVT5TKDCXDFDDR0DUTZTUKTE1J02ZOCSTRSFZ4V3RTV5VN' # your Foursquare ID
CLIENT_SECRET = 'POM4TH0RJ3U0NAEHV4VLSNGS2RGS2Q0BMWYXPF10AVXANF1E' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value
```

In [3]:
```
## URL to extract City neighbourhood information
urlL = "https://en.wikipedia.org/wiki/List_of_London_boroughs"
pageLondon = urllib.request.urlopen(urlL)
wikitablesL = pd.read_html(urlL,  attrs={"class":"wikitable"})
dfL0 = wikitablesL[0]
dfL1 = wikitablesL[1]
```

In [4]:
```
dfL0=dfL0.rename(columns={'Population (2019 est)[1]':'Population(2019 est)',}).drop(columns=['Nr. in map'])
dfL1=dfL1.drop(columns=['Nr. inmap'])
dfl=pd.concat([dfL0,dfL1]).reset_index().drop(columns=['index'])
dfl=dfl.drop(columns=['Inner','Status','Local authority','Political control','Headquarters','Area (sq mi)','
dfl['Co-ordinates'][0]='51.5607°N 0.1557°E'
lat_list=[51.5607]
log_list=[0.1557]
for i in range(1,dfl.shape[0]):
    index=dfl['Co-ordinates'][i].find('/')
    dfl['Co-ordinates'][i]=dfl['Co-ordinates'][i][index+1:]
    coord=dfl['Co-ordinates'][i].split(' ')
    lat=float(coord[1][:8].encode('utf-8').decode('utf-8-sig'))
    if coord[2][-1]=='W':
        flag=-1
    else:
        flag=1
    log=flag*float(coord[2][:6])
    lat_list.append(lat)
    log_list.append(log)
dfl['Latitude']=lat_list
dfl['Longitude']=log_list
dfl.drop(columns=['Co-ordinates'],inplace=True)
dfl.loc[0,'Borough']='Barking and Dagenham'
dfl.loc[9,'Borough']='Greenwich'
dfl.loc[11,'Borough']='Hammersmith and Fulham'
dfl.head()
```

Out[4]:

	Borough	Latitude	Longitude
0	Barking and Dagenham	51.5607	0.1557
1	Barnet	51.6252	-0.1517
2	Bexley	51.4549	0.1505
3	Brent	51.5588	-0.2817
4	Bromley	51.4039	0.0198

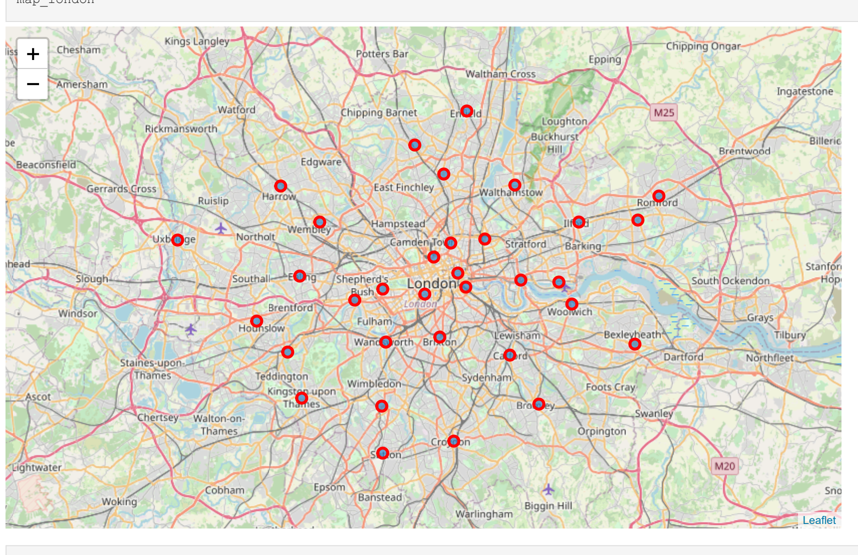
In [5]:
```
dfp=pd.read_csv(R"C:\Users\lautszchun\Desktop\projects_and_resources\coursera-data-science\London-Paris-camp
dfp.head()
```

Out[5]:

	postal_code	district	nom_dept	Latitude	Longitude
0	75001	Louvre	PARIS	48.862630	2.336293
1	75002	Bourse	PARIS	48.867903	2.344107
2	75003	Temple	PARIS	48.863054	2.359361
3	75004	Hotel-de-Ville	PARIS	48.854228	2.357362
4	75005	Pantheon	PARIS	48.844509	2.349859

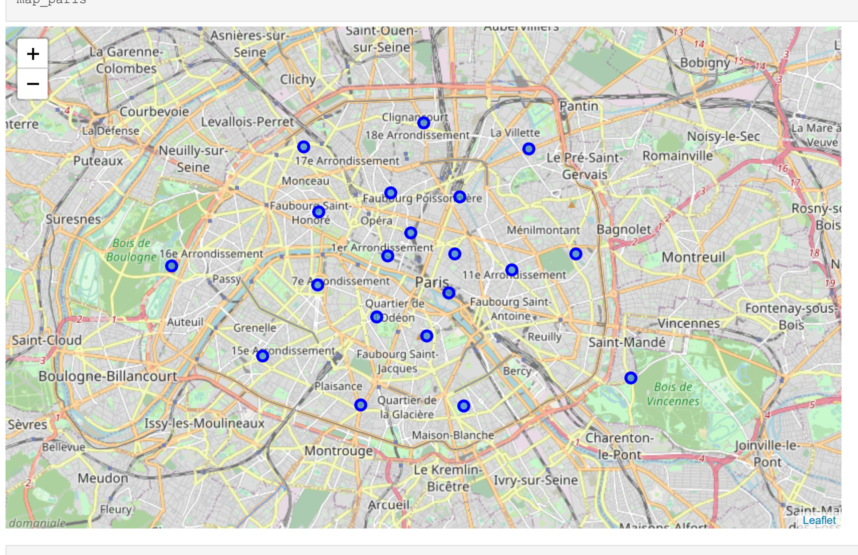
In [6]:
```
address = 'London'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
london_latitude = location.latitude
london_longitude = location.longitude

map_london = folium.Map(location=[london_latitude, london_longitude], zoom_start=10)

for lat, lng, borough in zip(dfl['Latitude'],dfl['Longitude'],dfl['Borough']):
    label = '{}'.format(borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_london)

map_london
```

Out[6]:



In [7]:
```
address = 'Paris'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
paris_latitude = location.latitude
paris_longitude = location.longitude

map_paris = folium.Map(location=[paris_latitude, paris_longitude], zoom_start=12)

for lat, lng, borough in zip(dfp['Latitude'],dfp['Longitude'],dfp['district']):
    label = '{}'.format(borough)
    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,
        parse_html=False).add_to(map_paris)

map_paris
```

Out[7]:



In [8]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&rad
            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)
```

In [9]:
```
London_venues=getNearbyVenues(dfl['Borough'],dfl['Latitude'],dfl['Longitude'])
```

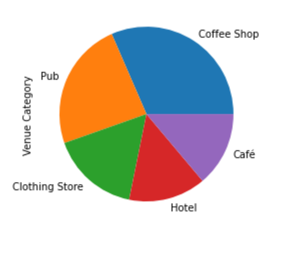
In [11]:
```
Paris_venues = getNearbyVenues(dfp['district'],dfp['Latitude'],dfp['Longitude'])
```

In [12]:
```
# one hot encoding
london_onehot = pd.get_dummies(London_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
london_onehot['Neighborhood'] = London_venues['Neighborhood']

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

# one hot encoding
paris_onehot = pd.get_dummies(Paris_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
paris_onehot['Neighborhood'] = Paris_venues['Neighborhood']

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

Showing the top 5 venue in London

In [13]:
```
ld_venue_count=London_venues['Venue Category'].value_counts()
ld_venue_count[:5].plot(kind = 'pie')
```

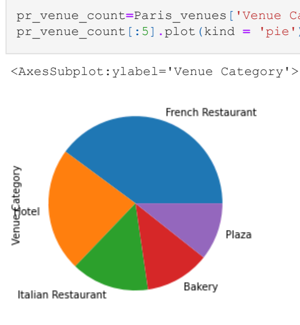
Out[13]: <AxesSubplot:ylabel='Venue Category'>



Showing the top 5 venue in Paris

In [14]:
```
pr_venue_count=Paris_venues['Venue Category'].value_counts()
pr_venue_count[:5].plot(kind = 'pie')
```

Out[14]: <AxesSubplot:ylabel='Venue Category'>



In [15]:
```
london_grouped = london_onehot.groupby('Neighborhood').mean().reset_index()
paris_grouped = paris_onehot.groupby('Neighborhood').mean().reset_index()
```

In [16]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

In [17]:
```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Borough']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
london_venues_sorted = pd.DataFrame(columns=columns)
london_venues_sorted['Borough'] = london_grouped['Neighborhood']

for ind in np.arange(london_grouped.shape[0]):
    london_venues_sorted.iloc[ind, 1:] = return_most_common_venues(london_grouped.iloc[ind, :], num_top_venu

columns = ['district']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

paris_venues_sorted = pd.DataFrame(columns=columns)
paris_venues_sorted['district'] = paris_grouped['Neighborhood']

for ind in np.arange(paris_grouped.shape[0]):
    paris_venues_sorted.iloc[ind, 1:] = return_most_common_venues(paris_grouped.iloc[ind, :], num_top_venues
```

In [18]:
```
k=3
london_grouped_clustering = london_grouped.drop('Neighborhood', 1)
london_kmeans = KMeans(n_clusters=k, random_state=0).fit(london_grouped_clustering)

paris_grouped_clustering = paris_grouped.drop('Neighborhood', 1)
paris_kmeans = KMeans(n_clusters=k, random_state=0).fit(paris_grouped_clustering)
```

In [19]:
```
london_venues_sorted.insert(0, 'Cluster Labels', london_kmeans.labels_)

london_merged = dfl

london_merged = london_merged.join(london_venues_sorted.set_index('Borough'), on='Borough')

london_merged.head()
```

Out[19]:

	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
0	Barking and Dagenham	51.5607	0.1557	1	Gym / Fitness Center	Park	Supermarket	Golf Course	Martial Arts School	Bus Station	Pool	Playground
1	Barnet	51.6252	-0.1517	2	Café	Bus Stop	Home Service	African Restaurant	Pedestrian Plaza	Nightclub	Okonomiyaki Restaurant	Optical Shop
2	Bexley	51.4549	0.1505	1	Pub	Clothing Store	Supermarket	Coffee Shop	Fast Food Restaurant	Pharmacy	Portuguese Restaurant	Bakery
3	Brent	51.5588	-0.2817	1	Coffee Shop	Hotel	Sporting Goods Shop	Clothing Store	Bar	Indian Restaurant	Sandwich Place	Grocery Store
4	Bromley	51.4039	0.0198	1	Clothing Store	Coffee Shop	Pizza Place	Gym / Fitness Center	Burger Joint	Bar	English Restaurant	Café

In [20]:
```
paris_venues_sorted.insert(0, 'Cluster Labels', paris_kmeans.labels_)

paris_merged = dfp

paris_merged = paris_merged.join(paris_venues_sorted.set_index('district'), on='district')

paris_merged.head()
```

Out[20]:

	postal_code	district	nom_dept	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
0	75001	Louvre	PARIS	48.862630	2.336293	0	French Restaurant	Hotel	Plaza	Japanese Restaurant	Art Museum	Coffee Shop
1	75002	Bourse	PARIS	48.867903	2.344107	0	French Restaurant	Cocktail Bar	Bakery	Coffee Shop	Creperie	Italian Restaurant
2	75003	Temple	PARIS	48.863054	2.359361	0	French Restaurant	Coffee Shop	Gourmet Shop	Japanese Restaurant	Italian Restaurant	Bakery
3	75004	Hotel-de-Ville	PARIS	48.854228	2.357362	0	French Restaurant	Ice Cream Shop	Clothing Store	Hotel	Park	Wine Bar
4	75005	Pantheon	PARIS	48.844509	2.349859	0	French Restaurant	Hotel	Bakery	Italian Restaurant	Plaza	Café

Showing the Clusters of London

In [21]:
```
# create map
london_map_clusters = folium.Map(location=[london_latitude, london_longitude], zoom_start=10)

# set color scheme for the clusters
x = np.arange(k)
ys = [i + x + (i*x)**2 for i in range(k)]
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(london_merged['Latitude'], london_merged['Longitude'], london_merged['Boro
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(london_map_clusters)

london_map_clusters
```

Out[21]:

Showing the Clusters of Paris

In [22]:
```
# create map
paris_map_clusters = folium.Map(location=[paris_latitude, paris_longitude], zoom_start=12)

# set color scheme for the clusters
x = np.arange(k)
ys = [i + x + (i*x)**2 for i in range(k)]
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(paris_merged['Latitude'], paris_merged['Longitude'], paris_merged['distric
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(paris_map_clusters)

paris_map_clusters
```

Out[22]:

In [29]:
```
(london_merged.loc[london_merged['Cluster Labels'] == 0])
```

Out[29]:

	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
16	Hounslow	51.4746	-0.368	0	Bed & Breakfast	Café	Park	Chinese Restaurant	African Restaurant	Nightclub	Okonomiyaki Restaurant	Optical Shop

In [30]:
```
(london_merged.loc[london_merged['Cluster Labels'] == 1])
```

Out[30]:

	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue
0	Barking and Dagenham	51.5607	0.1557	1	Gym / Fitness Center	Park	Supermarket	Golf Course	Martial Arts School	Bus Station	Pool
2	Bexley	51.4549	0.1505	1	Pub	Clothing Store	Supermarket	Coffee Shop	Fast Food Restaurant	Pharmacy	Portuguese Restaurant
3	Brent	51.5588	-0.2817	1	Coffee Shop	Hotel	Sporting Goods Shop	Clothing Store	Bar	Indian Restaurant	Sandwich Place
4	Bromley	51.4039	0.0198	1	Clothing Store	Coffee Shop	Pizza Place	Gym / Fitness Center	Burger Joint	Bar	English Restaurant
5	Camden	51.5290	-0.1255	1	Coffee Shop	Café	Hotel	Pub	Deli / Bodega	Burger Joint	Italian Restaurant
6	Croydon	51.3714	-0.0977	1	Pub	Coffee Shop	Portuguese Restaurant	Caribbean Restaurant	Clothing Store	Nightclub	Gaming Cafe
7	Ealing	51.5130	-0.3089	1	Coffee Shop	Clothing Store	Park	Bakery	Pub	Italian Restaurant	Gym / Fitness Center
8	Enfield	51.6538	-0.0799	1	Clothing Store	Coffee Shop	Pub	Supermarket	Department Store	Optical Shop	Stationery Store
9	Greenwich	51.4892	0.0648	1	Pub	Supermarket	Fast Food Restaurant	Clothing Store	Coffee Shop	Hotel	Grocery Store
10	Hackney	51.5450	-0.0553	1	Pub	Coffee Shop	Brewery	Bakery	Cocktail Bar	Organic Grocery	Café
11	Hammersmith and Fulham	51.4927	-0.2339	1	Pub	Café	Italian Restaurant	Indian Restaurant	Coffee Shop	Hotel	Gastropub
12	Haringey	51.6000	-0.1119	1	Park	Café	Italian Restaurant	Bakery	Portuguese Restaurant	Convenience Store	Fast Food Restaurant
13	Harrow	51.5898	-0.3346	1	Platform	Indian Restaurant	Grocery Store	Indie Movie Theater	Supermarket	Coffee Shop	Thai Restaurant
14	Havering	51.5812	0.1837	1	Coffee Shop	Hotel	Shopping Mall	Clothing Store	Department Store	Pub	Bookstore
15	Hillingdon	51.5441	-0.4760	1	Coffee Shop	Clothing Store	Fast Food Restaurant	Pub	Pharmacy	Department Store	Burger Joint
17	Islington	51.5416	-0.1022	1	Pub	Park	Theater	Music Venue	Bakery	Burger Joint	Boutique
18	Kensington and Chelsea	51.5020	-0.1947	1	Bakery	Café	Juice Bar	Restaurant	Hotel	Clothing Store	Art Gallery
19	Kingston upon Thames	51.4085	-0.3064	1	Coffee Shop	Café	Clothing Store	Pub	Italian Restaurant	Bakery	Department Store
20	Lambeth	51.4607	-0.1163	1	Caribbean Restaurant	Coffee Shop	Pub	Market	Beer Bar	Gym / Fitness Center	Pizza Place
21	Lewisham	51.4452	-0.0209	1	Supermarket	Grocery Store	Coffee Shop	Italian Restaurant	Platform	Train Station	Shopping Mall
22	Merton	51.4014	-0.1958	1	Diner	Café	Supermarket	Bakery	Pizza Place	Sandwich Place	Coffee Shop
23	Newham	51.5077	0.0469	1	Hotel	Airport Service	Sandwich Place	Pharmacy	Airport Lounge	Airport	Rafting
24	Redbridge	51.5590	0.0741	1	Fast Food Restaurant	Supermarket	Clothing Store	Grocery Store	Department Store	Bakery	Sandwich Place
25	Richmond upon Thames	51.4479	-0.3260	1	Pub	Coffee Shop	Italian Restaurant	Grocery Store	Pharmacy	Indian Restaurant	Platform
26	Southwark	51.5035	-0.0804	1	Coffee Shop	Pub	Bar	Scenic Lookout	Cocktail Bar	Restaurant	French Restaurant
27	Sutton	51.3618	-0.1945	1	Clothing Store	Pub	Coffee Shop	Café	Italian Restaurant	Sandwich Place	Department Store
28	Tower Hamlets	51.5099	-0.0059	1	Coffee Shop	Gym / Fitness Center	Italian Restaurant	Hotel	Sandwich Place	Café	Wine Shop
29	Waltham Forest	51.5908	-0.0134	1	Pub	Vegetarian / Vegan Restaurant	Concert Hall	Gym / Fitness Center	Beer Store	Tea Room	Pizza Place
30	Wandsworth	51.4567	-0.1910	1	Pub	Coffee Shop	Clothing Store	Indian Restaurant	Breakfast Spot	Supermarket	Asian Restaurant
31	Westminster	51.4973	-0.1372	1	Hotel	Coffee Shop	Sandwich Place	Theater	Italian Restaurant	Sushi Restaurant	Pub
32	City of London	51.5155	-0.0922	1	Coffee Shop	French Restaurant	Hotel	Steakhouse	Clothing Store	Wine Bar	Seafood Restaurant

In [31]:
```
(london_merged.loc[london_merged['Cluster Labels'] == 2])
```

Out[31]:

	Borough	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
1	Barnet	51.6252	-0.1517	2	Café	Bus Stop	Home Service	African Restaurant	Pedestrian Plaza	Nightclub	Okonomiyaki Restaurant	Optical Shop

In [33]:
```
paris_merged.loc[paris_merged['Cluster Labels'] == 0]
```

Out[33]:

	postal_code	district	nom_dept	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	75001	Louvre	PARIS	48.862630	2.336293	0	French Restaurant	Hotel	Plaza	Japanese Restaurant	Art Museum
1	75002	Bourse	PARIS	48.867903	2.344107	0	French Restaurant	Cocktail Bar	Bakery	Coffee Shop	Creperie
2	75003	Temple	PARIS	48.863054	2.359361	0	French Restaurant	Coffee Shop	Gourmet Shop	Japanese Restaurant	Italian Restaurant
3	75004	Hotel-de-Ville	PARIS	48.854228	2.357362	0	French Restaurant	Ice Cream Shop	Clothing Store	Hotel	Park
4	75005	Pantheon	PARIS	48.844509	2.349859	0	French Restaurant	Hotel	Bakery	Italian Restaurant	Plaza
5	75006	Luxembourg	PARIS	48.848968	2.332671	0	Bakery	Chocolate Shop	Pastry Shop	French Restaurant	Fountain
6	75007	Palais-Bourbon	PARIS	48.856083	2.312439	0	Hotel	French Restaurant	Italian Restaurant	Café	Plaza
7	75008	Elysse	PARIS	48.872527	2.312583	0	French Restaurant	Hotel	Spa	Bakery	Toy / Game Store
8	75009	Opera	PARIS	48.876896	2.337460	0	French Restaurant	Hotel	Bistro	Japanese Restaurant	Wine Bar
9	75010	Entrepot	PARIS	48.876029	2.361113	0	French Restaurant	Bistro	Hotel	Café	Coffee Shop
10	75011	Popincourt	PARIS	48.859415	2.378741	0	Restaurant	Italian Restaurant	Café	Bakery	French Restaurant
12	75013	Gobelins	PARIS	48.828718	2.362468	0	Vietnamese Restaurant	Asian Restaurant	Chinese Restaurant	Thai Restaurant	French Restaurant
13	75014	Observatoire	PARIS	48.828993	2.327101	0	French Restaurant	Bakery	Japanese Restaurant	Bistro	Hotel
14	75015	Vaugirard	PARIS	48.840155	2.293559	0	Italian Restaurant	Hotel	French Restaurant	Plaza	Park
16	75017	Batignolles-Monceau	PARIS	48.887337	2.307486	0	Hotel	French Restaurant	Italian Restaurant	Japanese Restaurant	Café
17	75018	Butte-Montmartre	PARIS	48.892735	2.348712	0	French Restaurant	Bar	Italian Restaurant	Plaza	Pizza Place
18	75019	Buttes-Chaumont	PARIS	48.886869	2.384694	0	French Restaurant	Bar	Bistro	Playground	Supermarket
19	75020	Menilmontant	PARIS	48.863187	2.400820	0	Bakery	Bistro	Japanese Restaurant	Plaza	French Restaurant

In [34]:
```
paris_merged.loc[paris_merged['Cluster Labels'] == 1]
```

Out[34]:

	postal_code	district	nom_dept	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
15	75016	Passy	PARIS	48.860399	2.2621	1	Lake	Plaza	Bus Station	Art Museum	Boat or Ferry	Bus Stop

In [35]:
```
paris_merged.loc[paris_merged['Cluster Labels'] == 2]
```

Out[35]:

	postal_code	district	nom_dept	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
11	75012	Reuilly	PARIS	48.835156	2.419807	2	Zoo Exhibit	Zoo	Supermarket	Monument / Landmark	Bistro	Pedestrian Plaza

Final Conculsion

As shown in the two cluster map of London and Paris, In fact most of the boroughs or districts in each city are quite similar with only some outliners that differ from the rest.

In London, Hounslow and Barnet are the outliner. In Paris, Passy and Reuilly are the outliner.