

# Week5

January 28, 2019

## 0.1 Introduction

An Italian start-up company has decided to expand its presence to France and wants to open its new offices in central Paris.

In order to ensure the satisfaction of its employees, the company wants to find a spot which has the same characteristics of its Italian location in central Milan (Brera district).

In order to find out which neighborhood satisfies these requirements, the central areas of the two cities will be compared using K-Means clustering technique.

## 0.2 Data

The data used for this analysis will be obtained mainly from these sources:

- Geopy, to obtain coordinates
- Wikipedia (as shown below) for neighborhoods names
- Foursquare, through its API, for venues in each neighborhood

```
In [2]: # importing required modules
```

```
import pandas as pd
import requests
```

```
In [3]: # importing Milano neighborhoods
```

```
url = 'https://it.wikipedia.org/wiki/Municipi_di_Milano#Schema_delle_zone_di_Milano'
results = requests.get(url).text
df = pd.read_html(results, header=0, attrs={"class":"wikitable sortable"})[0]
df = df[0:9]
df.drop(['Superficie(kmš)', 'Abitanti(31.12.2015)', 'Densità(ab/kmš)', 'Denominazione'], axis=1)
df.rename(columns={'#': 'Borough', 'Quartieri compresi[4]': 'Neighborhood'}, inplace=True)
df_raw = df.copy()
df_raw.head()
```

```
# refining Milano dataframe
```

```
df_rev = df_raw.set_index('Borough').Neighborhood.str.split(',', expand=True).stack().reset_index()
df_rev.rename(columns={0: 'Neighborhood'}, inplace=True)
df_rev.Neighborhood = df_rev.Neighborhood.astype(str) + ', Milano, IT'
mi = df_rev.copy()
mi.reset_index(drop=True, inplace=True)
```

```
# filtering only center area of Milano
```

```
mi = mi.loc[mi['Neighborhood'] == ' Brera, Milano, IT']
mi.reset_index(drop=True, inplace=True)
mi
```

```
Out[3]:
```

	Borough	Neighborhood
0	Municipio 1	Brera, Milano, IT

```
In [4]: # importing Paris neighborhoods
url = 'https://en.wikipedia.org/wiki/Arrondissements_of_Paris#Arrondissements'
results = requests.get(url).text
df = pd.read_html(results, header=0, attrs={"class":"wikitable sortable"})[0]
df.drop(['Area (in kmš)', 'Population(March 1999 census)', 'Population(July 2005 estimate)'],
        axis=1, inplace=True)
df.rename(columns={'Arrondissement (R for Right Bank, L for Left Bank)': 'Borough', 'Name':
                  'Neighborhood'})

df.Neighborhood = df.Neighborhood.astype(str) + ', Paris, FR'
pa = df.copy()

# filtering only center area of Paris
pa = pa[0:9].copy()
pa
```

```
Out[4]:
```

	Borough	Neighborhood
0	1st (Ie) R	Louvre, Paris, FR
1	2nd (IIe) R	Bourse, Paris, FR
2	3rd (IIIe) R	Temple, Paris, FR
3	4th (IVe) R	Hôtel-de-Ville, Paris, FR
4	5th (Ve) L	Panthéon, Paris, FR
5	6th (VIe) L	Luxembourg, Paris, FR
6	7th (VIIe) L	Palais-Bourbon, Paris, FR
7	8th (VIIIe) R	Élysée, Paris, FR
8	9th (IXe) R	Opéra, Paris, FR

```
In [5]: # merging dataframes for next steps
df2 = mi.append(pa)
df2.reset_index(drop=True, inplace=True)
df2
```

```
Out[5]:
```

	Borough	Neighborhood
0	Municipio 1	Brera, Milano, IT
1	1st (Ie) R	Louvre, Paris, FR
2	2nd (IIe) R	Bourse, Paris, FR
3	3rd (IIIe) R	Temple, Paris, FR
4	4th (IVe) R	Hôtel-de-Ville, Paris, FR
5	5th (Ve) L	Panthéon, Paris, FR
6	6th (VIe) L	Luxembourg, Paris, FR
7	7th (VIIe) L	Palais-Bourbon, Paris, FR
8	8th (VIIIe) R	Élysée, Paris, FR
9	9th (IXe) R	Opéra, Paris, FR

### 0.3 Analysis

```
In [6]: #importing modules
import numpy as np
from pandas.io.json import json_normalize
from geopy.geocoders import Nominatim
from sklearn.cluster import KMeans
!conda install -c conda-forge folium=0.5.0 -y
import folium
import matplotlib.cm as cm
import matplotlib.colors as colors
```

Solving environment: done

## Package Plan ##

environment location: /home/jupyterlab/conda

added / updated specs:

- folium=0.5.0

The following packages will be downloaded:

package	build	
conda-4.6.1	py36_0	878 KB conda-forge

The following packages will be UPDATED:

conda: 4.5.12-py36\_1000 conda-forge --> 4.6.1-py36\_0 conda-forge

Downloading and Extracting Packages

conda-4.6.1 | 878 KB | ##### | 100%

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

```
In [7]: # @hidden_cell
```

```
CLIENT_ID = 'ETP1NEPRDDV5NK2QFTDWZK1AI5Q1JRWJRH1CSE4JPHIZIHPA' # your Foursquare ID
CLIENT_SECRET = 'SCV01YQ4NC2M5NN5CBTQAOUW1Y5SYPVNUQNRCLGHBXBPR3Z' # your Foursquare Sec
VERSION = '20190112' # Foursquare API version

radius = 500
LIMIT = 100

print('Your credentails:')
```

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

Your credentials:

CLIENT\_ID: ETP1NEPRDDV5NK2QFTDWZK1AI5Q1JRWJRH1CSE4JPHIZIHPA

CLIENT\_SECRET:SCV01YQ4NC2M5NN5CBTQAOUW1Y5SYPVNUQNRCLGHBXBPR3Z

```
In [8]: def findlat(row):
        geolocator = Nominatim(user_agent="my-app")
        location = geolocator.geocode(row['Neighborhood'])
        lat = location.latitude
        return lat
```

```
In [9]: def findlng(row):
        geolocator = Nominatim(user_agent="my-app")
        location = geolocator.geocode(row['Neighborhood'])
        lng = location.longitude
        return lng
```

```
In [10]: # finding latitude and longitude per each neighborhood in dataset
df2['lat'] = df2.apply(findlat, axis=1)
df2['lng'] = df2.apply(findlng, axis=1)
print(df2.shape)
df2.head()
```

(10, 4)

```
Out[10]:
```

	Borough	Neighborhood	lat	lng
0	Municipio 1	Brera, Milano, IT	45.471519	9.187735
1	1st (Ie) R	Louvre, Paris, FR	48.861147	2.338028
2	2nd (IIe) R	Bourse, Paris, FR	48.867687	2.343122
3	3rd (IIIe) R	Temple, Paris, FR	48.862701	2.358683
4	4th (IVe) R	Hôtel-de-Ville, Paris, FR	48.856426	2.352528

```
In [11]: # function to find venues through Foursquare API
def getNearbyVenues(names, latitudes, longitudes, radius=800):

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&lat={}&lng={}&radius={}&version={}'
        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']['id'],
        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 ID',
                             'Venue Category']

    return(nearby_venues)

```

```

In [12]: # finding venues for each neighborhood
venues = getNearbyVenues(names=df2['Neighborhood'],latitudes=df2['lat'],longitudes=df2['lng'])
venues.head()

```

```

Out[12]:
   Neighborhood Neighborhood Latitude Neighborhood Longitude \
0  Brera, Milano, IT          45.471519          9.187735
1  Brera, Milano, IT          45.471519          9.187735
2  Brera, Milano, IT          45.471519          9.187735
3  Brera, Milano, IT          45.471519          9.187735
4  Brera, Milano, IT          45.471519          9.187735

   Venue Venue Latitude Venue Longitude \
0  Pinacoteca di Brera    45.471979    9.188128
1  Di Viole Di Liquirizia  45.471460    9.185336
2  Bulgari Lounge Bar     45.470014    9.188943
3  Bulgari Hotel Milano   45.470535    9.190173
4  Palazzo di Brera       45.472019    9.188043

```

```

Venue Category
0      Art Museum
1      Cupcake Shop
2      Cocktail Bar
3      Hotel
4  College Arts Building

```

```
In [13]: venues.shape
```

```
Out[13]: (1000, 7)
```

```
In [14]: # venue categories in the dataset venues
```

```
print('There are {} uniques categories.'.format(len(venues['Venue Category'].unique())))
```

```
There are 176 uniques categories.
```

```
In [15]: # venue categories in Brera district in Milan only
```

```
brera = venues.loc[venues['Neighborhood'] == ' Brera, Milano, IT'].copy()
```

```
print('There are {} uniques categories.'.format(len(brera['Venue Category'].unique())))
```

```
There are 49 uniques categories.
```

```
In [16]: brera.shape
```

```
Out[16]: (100, 7)
```

```
In [17]: brera['Venue Category'].unique()
```

```
Out[17]: array(['Art Museum', 'Cupcake Shop', 'Cocktail Bar', 'Hotel',
                'College Arts Building', 'Wine Bar', 'Garden', 'Lounge',
                'Japanese Restaurant', 'Chocolate Shop', 'Italian Restaurant',
                'Men's Store', 'Boutique', 'Ice Cream Shop', 'Trattoria/Osteria',
                'Plaza', 'Bakery', 'Accessories Store', 'Theater', 'Fabric Shop',
                'Sandwich Place', 'Tea Room', 'Karaoke Bar', 'Bookstore',
                'Tapas Restaurant', 'Burger Joint', 'Café', 'Sushi Restaurant',
                'Park', 'Restaurant', 'Pizza Place', 'Art Gallery', 'Opera House',
                'Beer Store', 'Diner', 'Nightclub', 'Seafood Restaurant', 'Bistro',
                'Castle', 'Road', 'Fountain', 'Arts & Crafts Store',
                'Furniture / Home Store', 'Creperie', 'Coffee Shop',
                'French Restaurant', 'Spa', 'Paper / Office Supplies Store',
                'Jewelry Store'], dtype=object)
```

```
In [18]: # storing Brera district coordinates into variables for data visualization
```

```
geolocator = Nominatim(user_agent="my-app")
```

```
location = geolocator.geocode('Brera, Milano, IT')
```

```
lat_mi = location.latitude
```

```
lng_mi = location.longitude
```

```
print(lat_mi, lng_mi)
```

45.4715187 9.1877351

```
In [19]: # visualizing location of venue in a radius of 500 meters from the coordinates of Brera
brera_map = folium.Map(location=[lat_mi, lng_mi], zoom_start=15)
folium.Circle([lat_mi, lng_mi],
              radius=500,
              color='red').add_to(brera_map)

for lat, lon, poi in zip(brera['Venue Latitude'], brera['Venue Longitude'], brera['Venue Name']):
    label = folium.Popup(str(poi), parse_html=True)
    folium.CircleMarker(
        [float(lat), float(lon)],
        # radius=res_count,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7).add_to(brera_map)

brera_map
```

Out[19]: <folium.folium.Map at 0x7f78943efba8>

## 0.4 One Hot encoding

```
In [20]: # one hot encoding
df_onehot = pd.get_dummies(venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
df_onehot['Neighborhood'] = venues['Neighborhood']

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

df_onehot.head()
```

Out[20]:

	Neighborhood	Accessories Store	Alsatian Restaurant	\
0	Brera, Milano, IT	0	0	
1	Brera, Milano, IT	0	0	
2	Brera, Milano, IT	0	0	
3	Brera, Milano, IT	0	0	
4	Brera, Milano, IT	0	0	

  

	American Restaurant	Antique Shop	Argentinian Restaurant	Art Gallery	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	

3	0	0	0	0
4	0	0	0	0

	Art Museum	Arts & Crafts Store	Asian Restaurant	...	\
0	1	0	0	...	
1	0	0	0	...	
2	0	0	0	...	
3	0	0	0	...	
4	0	0	0	...	

	Train Station	Trattoria/Osteria	Udon Restaurant	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	Wine Bar	Wine Shop	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Women's Store	Zoo	Zoo Exhibit
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

[5 rows x 177 columns]

```
In [21]: df_grouped = df_onehot.groupby('Neighborhood').mean().reset_index()
print(df_grouped.shape)
df_grouped.head()
```

(10, 177)

```
Out[21]:
```

	Neighborhood	Accessories Store	Alsatian Restaurant	\
0	Brera, Milano, IT	0.01	0.00	
1	Bourse, Paris, FR	0.00	0.00	
2	Hôtel-de-Ville, Paris, FR	0.00	0.01	
3	Louvre, Paris, FR	0.00	0.00	
4	Luxembourg, Paris, FR	0.00	0.00	

	American Restaurant	Antique Shop	Argentinian Restaurant	Art Gallery	\
--	---------------------	--------------	------------------------	-------------	---



0	0.00	0.0	0.0	0.02
1	0.01	0.0	0.0	0.00
2	0.00	0.0	0.0	0.03
3	0.00	0.0	0.0	0.01
4	0.01	0.0	0.0	0.01

	Art Museum	Arts & Crafts Store	Asian Restaurant	...	\
0	0.03	0.01	0.0	...	
1	0.00	0.00	0.0	...	
2	0.02	0.02	0.0	...	
3	0.03	0.00	0.0	...	
4	0.01	0.00	0.0	...	

	Train Station	Trattoria/Osteria	Udon Restaurant	\
0	0.0	0.01	0.00	
1	0.0	0.00	0.01	
2	0.0	0.00	0.00	
3	0.0	0.00	0.02	
4	0.0	0.00	0.00	

	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	Wine Bar	Wine Shop	\
0	0.0	0.0	0.04	0.00	
1	0.0	0.0	0.06	0.01	
2	0.0	0.0	0.03	0.00	
3	0.0	0.0	0.03	0.01	
4	0.0	0.0	0.04	0.00	

	Women's Store	Zoo	Zoo Exhibit
0	0.00	0.0	0.0
1	0.01	0.0	0.0
2	0.00	0.0	0.0
3	0.00	0.0	0.0
4	0.01	0.0	0.0

[5 rows x 177 columns]

```
In [22]: # function to return the most common venues
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 [23]: num_top_venues = 10
```

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

```
# create columns according to number of top venues
```

```

columns = ['Neighborhood']
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
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = df_grouped['Neighborhood']

for ind in np.arange(df_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(df_grouped.il

neighborhoods_venues_sorted.head()

```

```

Out[23]:

```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	\
0	Brera, Milano, IT	Italian Restaurant	Hotel	
1	Bourse, Paris, FR	French Restaurant	Wine Bar	
2	Hôtel-de-Ville, Paris, FR	French Restaurant	Ice Cream Shop	
3	Louvre, Paris, FR	French Restaurant	Hotel	
4	Luxembourg, Paris, FR	French Restaurant	Italian Restaurant	

  

	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	\
0	Boutique	Plaza	Ice Cream Shop	
1	Cocktail Bar	Italian Restaurant	Boutique	
2	Gourmet Shop	Art Gallery	Plaza	
3	Plaza	Café	Restaurant	
4	Hotel	Ice Cream Shop	Wine Bar	

  

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	\
0	Wine Bar	Japanese Restaurant	Art Museum	
1	Bistro	Ice Cream Shop	Hotel	
2	Cocktail Bar	Coffee Shop	Bookstore	
3	Historic Site	Wine Bar	Cosmetics Shop	
4	Tea Room	Bakery	Seafood Restaurant	

  

	9th Most Common Venue	10th Most Common Venue
0	Bakery	Men's Store
1	Indie Movie Theater	Japanese Restaurant
2	Wine Bar	Art Museum
3	Art Museum	Bakery
4	Plaza	Pastry Shop

```

In [24]: kclusters = 2

```

```

df_grouped_clustering = df_grouped.drop('Neighborhood', 1)

```

```

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_

Out[24]: array([0, 1, 1, 1, 1, 0, 1, 1, 1, 1], dtype=int32)

In [25]: df_merged = df2

# add clustering labels
df_merged['Cluster Labels'] = kmeans.labels_

# merge df_merged with neighborhoods_venues_sorted dataframe to add latitude/longitude
df_merged = df_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='N

df_merged

Out[25]:
      Borough      Neighborhood      lat      lng \
0  Municipio 1  Brera, Milano, IT  45.471519  9.187735
1    1st (Ie) R    Louvre, Paris, FR  48.861147  2.338028
2    2nd (IIe) R    Bourse, Paris, FR  48.867687  2.343122
3    3rd (IIIe) R    Temple, Paris, FR  48.862701  2.358683
4    4th (IVe) R  Hôtel-de-Ville, Paris, FR  48.856426  2.352528
5    5th (Ve) L    Panthéon, Paris, FR  48.846191  2.346079
6    6th (VIe) L    Luxembourg, Paris, FR  48.849392  2.332260
7    7th (VIIe) L  Palais-Bourbon, Paris, FR  48.861692  2.319031
8    8th (VIIIe) R    Élysée, Paris, FR  48.846644  2.369830
9    9th (IXe) R    Opéra, Paris, FR  48.870645  2.332330

      Cluster Labels 1st Most Common Venue 2nd Most Common Venue \
0                0    Italian Restaurant      Hotel
1                1    French Restaurant      Hotel
2                1    French Restaurant      Wine Bar
3                1    French Restaurant      Coffee Shop
4                1    French Restaurant      Ice Cream Shop
5                0    French Restaurant      Hotel
6                1    French Restaurant    Italian Restaurant
7                1    French Restaurant      Garden
8                1    French Restaurant      Bar
9                1      Hotel      Jewelry Store

      3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue \
0      Boutique      Plaza      Ice Cream Shop
1      Plaza      Café      Restaurant
2    Cocktail Bar    Italian Restaurant      Boutique
3      Café    Japanese Restaurant    Moroccan Restaurant
4    Gourmet Shop      Art Gallery      Plaza

```

5	Bar	Pub	Bakery
6	Hotel	Ice Cream Shop	Wine Bar
7	Hotel	Historic Site	Bookstore
8	Hotel	Italian Restaurant	Cocktail Bar
9	Japanese Restaurant	French Restaurant	Gourmet Shop

  

6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	\
0 Wine Bar	Japanese Restaurant	Art Museum	
1 Historic Site	Wine Bar	Cosmetics Shop	
2 Bistro	Ice Cream Shop	Hotel	
3 Burger Joint	Sandwich Place	Cocktail Bar	
4 Cocktail Bar	Coffee Shop	Bookstore	
5 Café	Wine Bar	Creperie	
6 Tea Room	Bakery	Seafood Restaurant	
7 Art Museum	Plaza	Fountain	
8 Nightclub	Plaza	Pedestrian Plaza	
9 Theater	Plaza	Chocolate Shop	

  

9th Most Common Venue	10th Most Common Venue
0 Bakery	Men's Store
1 Art Museum	Bakery
2 Indie Movie Theater	Japanese Restaurant
3 Chinese Restaurant	Bistro
4 Wine Bar	Art Museum
5 Ice Cream Shop	Plaza
6 Plaza	Pastry Shop
7 Bakery	Hotel Bar
8 Garden	Farmers Market
9 Cocktail Bar	Coffee Shop

## 0.5 Results

```
In [26]: df_merged.loc[df_merged['Cluster Labels'] == 0]
```

```
Out[26]:
```

	Borough	Neighborhood	lat	lng	Cluster Labels	\
0	Municipio 1	Brera, Milano, IT	45.471519	9.187735	0	
5	5th (Ve) L	Panthéon, Paris, FR	48.846191	2.346079	0	

  

1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	\
0 Italian Restaurant	Hotel	Boutique	
5 French Restaurant	Hotel	Bar	

  

4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	\
0 Plaza	Ice Cream Shop	Wine Bar	
5 Pub	Bakery	Café	

  

7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	\
0 Japanese Restaurant	Art Museum	Bakery	

	Wine Bar	Creperie	Ice Cream Shop
5			
	10th Most Common Venue		
0	Men's Store		
5	Plaza		

```
In [27]: # venue categories in Panthéon district in Paris only
pantheon = venues.loc[venues['Neighborhood'] == 'Panthéon, Paris, FR'].copy()
print('There are {} uniques categories.'.format(len(brera['Venue Category'].unique())))
```

There are 49 uniques categories.

```
In [28]: pantheon.shape
```

```
Out[28]: (100, 7)
```

```
In [29]: pantheon['Venue Category'].unique()
```

```
Out[29]: array(['Monument / Landmark', 'Plaza', 'Pub', 'Ethiopian Restaurant',
                'Cupcake Shop', 'Portuguese Restaurant', 'Italian Restaurant',
                'Wine Bar', 'French Restaurant', 'Bakery', 'Bistro',
                'Sushi Restaurant', 'Hotel', 'Science Museum', 'Coffee Shop',
                'Café', 'Bar', 'Ice Cream Shop', 'Creperie', 'Korean Restaurant',
                'Indie Movie Theater', 'Toy / Game Store', 'Cheese Shop',
                'Pastry Shop', 'Vietnamese Restaurant', 'Burger Joint', 'Diner',
                'Comic Shop', 'Asian Restaurant', 'Chinese Restaurant',
                'Wine Shop', 'History Museum', 'Mexican Restaurant',
                'Historic Site', 'Bookstore', 'Greek Restaurant', 'Garden', 'Gym',
                'Tapas Restaurant', 'Fountain', 'Hobby Shop', 'Miscellaneous Shop',
                'Japanese Restaurant', 'Lebanese Restaurant', 'Seafood Restaurant',
                'Poke Place', 'Gourmet Shop'], dtype=object)
```

It seems that Panthéon has the same variety of venue categories.

Let's visualize the **location** of each venue on a map of the Panthéon district.

```
In [30]: # storing Panthéon district coordinates into variables for data visualization
geolocator = Nominatim(user_agent="my-app")
location = geolocator.geocode('Panthéon, Paris, FR')
lat_pa = location.latitude
lng_pa = location.longitude
print(lat_pa, lng_pa)
```

```
48.84619085 2.34607852190515
```

```
In [31]: # visualizing location of venues in a radius of 500 meters from the coordinates of Pant
pantheon_map = folium.Map(location=[lat_pa, lng_pa], zoom_start=15)
folium.Circle([lat_pa, lng_pa],
```

```

        radius=500,
        color='red').add_to(pantheon_map)

for lat, lon, poi in zip(pantheon['Venue Latitude'], pantheon['Venue Longitude'], pantheon['Venue Name']):
    label = folium.Popup(str(poi), parse_html=True)
    folium.CircleMarker(
        [float(lat), float(lon)],
        # radius=res_count,
        popup=label,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.7).add_to(pantheon_map)

```

```
pantheon_map
```

```
Out[31]: <folium.folium.Map at 0x7f789235de80>
```

```
In [ ]:
```