The Battle of the Neighborhoods: Is Copenhagen like Paris?

Applied Data Science Capstone, Coursera/IBM

Author: Paw Hermansen (https://pawhermansen.dk (https://pawhermansen.dk))
Date: November 13, 2018

This notebook contains my Capstone Project for the Coursera/IBM course series *IBM Data Science Professional Certificate Specialization*.

Table of Contents

Import Pyton Code Libraries

In [1]:
```
from bs4 import BeautifulSoup
from geopy.geocoders import Nominatim
from sklearn.cluster import KMeans
from IPython.display import Markdown, display
import os
import numpy as np
import pandas as pd
import requests
import csv
import folium
import geopy.distance
```

1. The Problem

Travel guide publisher *Lonely Planet* recently put the danish capital Copenhagen on top of their list of the best cities to visit in 2019. Copenhagen has a lot to offer, says *Lonely Planet* and mentions the cyclists, the many green spaces, the old and new architecture, the great museums, Tivoli garden, the galleries, the restaurants, including fancy New Nordic restaurants and even marvelous street food markets and indie bars.

That made some Copenhageners claim in the local newspapers that Copenhagen is like Paris in the summer. If this is true it will be interesting not only for tourists trying to find new and exciting destinations but also for the Copenhagen tourist association, visitcopenhagen.dk, that could direct its marketing to compete directly against other cities like Paris.

It is not stated clearly in exactly what way the likeness between Copenhagen and Paris is thought to be. It is clearly not in the weather because the Copenhageners compare Copenhagen in the summer to Paris and they do not mention the winter. Also it is clearly not in the languages - even though that both Copenhagen and Paris are very alike in that they speak languages that are totally un-understandable to anyone else. In Copenhhagen, however, nearly everyone also speaks fluent English which is certainly not the case in Paris.

The likeness between Copenhagen and Paris is probably more a feeling that when you walk around in Copenhagen and Paris you will see the same kind and distribution of restaurants, bars, sights, bakeries and all other kinds of venues and this is this definition of likeness that I choose to investigate.

This notebook uses tools from Data Science and Machine Learning to investigate if Copenhagen is like Paris in the above mentioned sense.

2. Machine Learning Approach

My approach will be to part each of the two cities into neighborhoods that I will consider as homogeneous with respect to their venue types.

Then to see how alike the Copenhagen and the Paris neighborhoods are I will make a cluster analysis of all the neighborhoods together based on their frequency of their venue types. Clustering is a method of so-called unsupervised learning where the algorithm takes data-points that are not catagorized or grouped beforehand and groups them into a given number of clusters or groups based on their likeness. Here the list of venue type frequencies of each neighborhood is a data-point.

When I do this I actually cheat the cluster algorithm a little bit because the data-points **are** categorized beforehand with their city. I do not, however, reveal this to the cluster algorithm that will group the neighborhoods exclusively based on the venue type frequencies. My conclusion will be based on the result of the clustering algorithm compared to which city each neighborhood in each cluster is part of.

For example if I cluster all the neighborhoods into two or more clusters and all the Copenhagen neighborhoods end up in their own clusters and all the Paris neighborhoods end up in other clusters then Copenhagen and Paris are more alike to themself than to each other. But if the neighborhoods ends up being mixed in clusters across the two cities then you are very right in claiming that Copenhagen, or at least some neighborhoods of Copenhagen, are like Paris.

If the two cities in fact have neighborhoods that are alike then the groups will show which neighborhoods from the two cities are most like each other.

3. Data Requirements

Two sets of data for each city is needed for the cluster analysis - the neighborhoods and the venue types within each neighborhood.

The neighborhoods

To be used in the cluster algorithm the frequencies of the venue types must be brought into so-called One Hot Encoding showing their frequncy within each neighborhood.

The neighborhoods should be small enough to be close to homogeneous with respect to their venue types but they should also be large enough to contain enough venues and venue types that it makes sense to take the ten most frequently venue types. I guess that each neighborhood will need at least about 25 venues.

The nighborhoods can be found in several ways. One way will be to base them on postal codes but this might not define neighborhoods that align with the common tourist point of view. Instead the neighborhoods can be scraped from tourist websites and their locations can be found from wikipedia or from their addresses through geographical services. The Paris data might be available from OpenData Paris.

The Venue types within each neighborhood

Foursquare is a service that you can use to find the best places to eat, drink, shop, or visit in any city in the world. They also offer access through an open API with some limitations, registering necessary.

We can call the Foursquare API a list of venues and their types within a certain distance from any location within Copenhagen and Paris. This means that for our purpose the neighborhoods will be defined as a center location and a radius around this center.

4. Data Collection

The neighborhoods of Paris

Paris is parted into 20 so-called arrondissements that are administrative zones of Paris that fit most tourists view of neighborhoods in Paris. The name and the geographical coordinates of the location of each arrondissement can be downloaded in different formats from the Paris Data (https://opendata.paris.fr) website - the data is covered by the Open Database License (ODbL) (https://opendatacommons.org/licenses/odbl/).

In [2]:
```
arrondissementsUrl = "https://opendata.paris.fr/explore/dataset/arrondissements/download"
df_arrondissements = pd.read_csv(arrondissementsUrl, sep=';')

print("Number of rows = {}, number of columns = {}".format(df_arrondissements.shape[0], df_arrondissements.shape[1]))
df_arrondissements.head(3)
```

```
Number of rows = 20, number of columns = 12
```

Out[2]:

	n_sq_ar	c_ar	c_arinsee	l_ar	l_aroff	n_sq_co	surface	perimetre	geom_x_y	geom	objectid	longueur
0	750000002	2	75102	2ème Ardt	Bourse	750001537	9.911537e+05	4554.104360	48.8682792225, 2.34280254689	{"type": "Polygon", "coordinates": [[[2.351518...	2	4553.938764
1	750000003	3	75103	3ème Ardt	Temple	750001537	1.170883e+06	4519.263648	48.86287238, 2.3600009859	{"type": "Polygon", "coordinates": [[[2.363828...	3	4519.071982
2	750000012	12	75112	12ème Ardt	Reuilly	750001537	1.631478e+07	24089.666298	48.8349743815, 2.42132490078	{"type": "Polygon", "coordinates": [[[2.413879...	12	24088.038922

I select the name and location fields.

In [3]:
```
names = df_arrondissements['l_aroff'].str.strip()

coordinates = df_arrondissements['geom_x_y'].str.split(',', expand=True)
coordinates[0] = pd.to_numeric(coordinates[0])
coordinates[1] = pd.to_numeric(coordinates[1])

df_parisNeighborhoods = pd.concat([names, coordinates], axis=1)
df_parisNeighborhoods.columns = ['Neighborhood', 'Latitude', 'Longitude']

df_parisNeighborhoods.to_csv('data/paris_neighborhoods.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_parisNeighborhoods.shape[0], df_parisNeighborhoods.shape[1]))
df_parisNeighborhoods.head(3)
```

```
Number of rows = 20, number of columns = 3
```

Out[3]:

	Neighborhood	Latitude	Longitude
0	Bourse	48.868279	2.342803
1	Temple	48.862872	2.360001
2	Reuilly	48.834974	2.421325

Later I will use Foursquare to find venues in each neighborhood. To get the venues Foursquare requires the geographical latitude and longitude for each neighborhood center and a maximal distance away from each center to search. this gives me a circular search-area centered in each neighborhood. I want my search-areas to be large enough to reflect the venues in the real neighborhoods but I want to avoid too much overlap with other search-areas as this might dilute the result.

To meet this I computer the distance in meter to each neighborhoods closest neighbor and later I set the radius of each search area to half of this distance. Finally I save the Paris neighborhoods to a local csv file.

In [4]:
```
def nearestDistances(df):
    nearest_distance = []
    for index, row in df.iterrows():
        minimumDist = 9999999
        coords_1 = (df.loc[index, 'Latitude'], df.loc[index, 'Longitude'])
        for i in range(0, df.shape[0]):
            if (i != index):
                coords_2 = (df.loc[i, 'Latitude'], df.loc[i, 'Longitude'])
                dist = geopy.distance.vincenty(coords_1, coords_2).m
                if dist < minimumDist:
                    minimumDist = dist
        nearest_distance.append(minimumDist)
    return nearest_distance

df_parisNeighborhoods['Distance to Nearest'] = nearestDistances(df_parisNeighborhoods)

df_parisNeighborhoods.to_csv('data/paris_neighborhoods.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of neighborhoods is", df_parisNeighborhoods.shape[0])
df_parisNeighborhoods
```

```
Number of neighborhoods is 20
```

Out[4]:

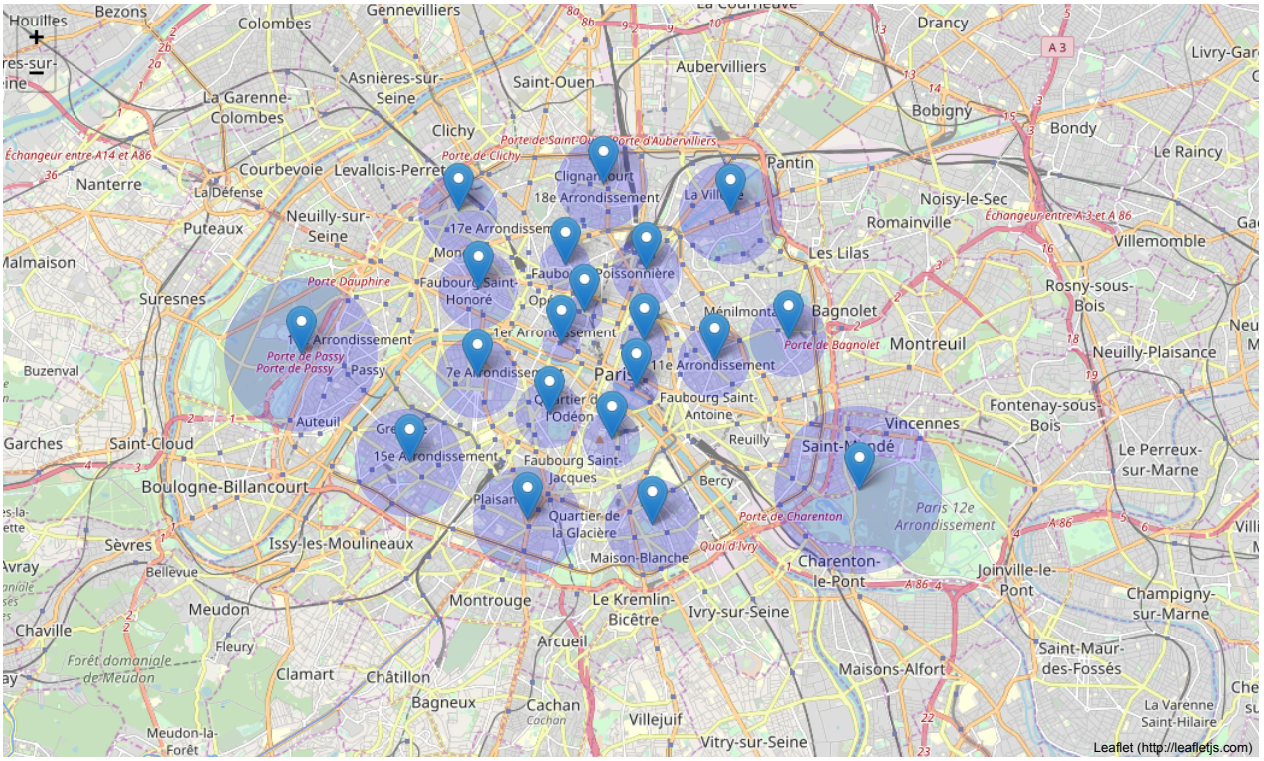
	Neighborhood	Latitude	Longitude	Distance to Nearest
0	Bourse	48.868279	2.342803	788.555891
1	Temple	48.862872	2.360001	964.527702
2	Reuilly	48.834974	2.421325	3495.637308
3	Louvre	48.862563	2.336443	788.555891
4	Hôtel-de-Ville	48.854341	2.357630	964.527702
5	Élysée	48.872721	2.312554	1678.621184
6	Observatoire	48.829245	2.326542	2260.122186
7	Buttes-Chaumont	48.887076	2.384821	2145.755629
8	Ménilmontant	48.863461	2.401188	1625.823751
9	Luxembourg	48.849130	2.332898	1407.726786
10	Opéra	48.877164	2.337458	1062.959118
11	Batignolles-Monceau	48.887327	2.306777	1678.621184
12	Vaugirard	48.840085	2.292826	2284.873885
13	Panthéon	48.844443	2.350715	1212.116820
14	Palais-Bourbon	48.856174	2.312188	1709.870559
15	Entrepôt	48.876130	2.360728	1475.312688
16	Popincourt	48.859059	2.380058	1531.584620
17	Gobelins	48.828388	2.362272	1976.786248
18	Passy	48.860392	2.261971	3198.053579
19	Buttes-Montmartre	48.892569	2.348161	1884.498861

My search-areas for the neighborhoods of Paris looks like this:

In [5]:
```
def createTownMap(df, zoom = 12):
    # Create map centered around the mean latitude and longitude values
    latitue = df['Latitude'].mean()
    longitude = df['Longitude'].mean()

    townmap = folium.Map(location=[latitue, longitude], zoom_start=zoom)

    # Add the search-areas to map.
    for lat, lng, neighborhood, radius in zip(df['Latitude'],
                                              df['Longitude'],
                                              df['Neighborhood'],
                                              df['Distance to Nearest'] / 2):
        label = folium.Popup(neighborhood, parse_html=True)

        folium.Marker(
            [lat, lng],
            popup = neighborhood).add_to(townmap)

        folium.Circle(
            radius=radius,
            location=[lat, lng],
            popup=label,
            color='blue',
            stroke= False,
            fill=True,
            fill_opacity=0.2).add_to(townmap)
    return townmap
```

In [6]:
```
createTownMap(df_parisNeighborhoods)
```

Out[6]:



The neighborhoods of Copenhagen

The Copenhagen neighborhoods are a little more difficult to get. From WikiPedia Bydele i Københavns kommune (https://da.wikipedia.org/wiki/Bydele_i_K%C3%B8benhavns_Kommune) ("neighbohoods in Copenhagen Commune") I collect the ten administrative areas of Copenhagen. They seem to fit most tourists view of neighborhoods in Copenhagen.

The neighborhood *Indre by* ("Inner City") can be subdivided into smaller functionel neigborhoods but it turns out that FourSquare, that I will use later, has too few venues registered for some of the smaller neighborhoods and so I stay with *Indre by* as one neighborhood.

Also I include *Frederiksberg* that is not administratively a part of the Copenhagen Commune but geographically lies inside the borders of Copenhagen (see https://www.quora.com/Why-is-Frederiksberg-not-a-part-of-Copenhagen (https://www.quora.com/Why-is-Frederiksberg-not-a-part-of-Copenhagen) for more information about this curiosity).

I get the data from the WikiPedia webpage and add *Frederiksberg* by hand.

In [7]:
```
baseUrl = 'https://da.wikipedia.org/'
url = baseUrl + 'wiki/Bydele_i_K%C3%B8benhavns_Kommune'

sauce = requests.get(url).content
soup = BeautifulSoup(sauce, 'lxml')

table = soup.find('table', {'class': 'navbox'})
td = table.find('td', {'class': 'navbox-list'})
links = td.find_all('a', href=True)

cphNames = []
for link in links:
    cphNames.append(link.text.strip())

cphNames.append('Frederiksberg')

print("Number of neighborhoods is " + str(len(cphNames)))
cphNames
```

```
Number of neighborhoods is 11
```

Out[7]:
```
['Amager Vest',
 'Amager Øst',
 'Bispebjerg',
 'Brønshøj-Husum',
 'Indre By',
 'Nørrebro',
 'Valby',
 'Vanløse',
 'Vesterbro/Kongens Enghave',
 'Østerbro',
 'Frederiksberg']
```

I am going to use the Nominatim geo locator to find the geographical coordinates near the center of each neighborhood. However, not all of the neighborhood names are found and some that are found returns coordinates far from their center.

I fix it by adding an address for each neighborhood and setting the address to a name that can be found by Nomination.

In [8]:
```
address = cphNames.copy()

address[address.index('Amager Vest')] = 'Vestamager'
address[address.index('Amager Øst')] = 'Øresundsvej'
address[address.index('Vesterbro/Kongens Enghave')] = 'Sønder Boulevard'
address[address.index('Indre By')] = 'Kongens Nytorv'

df_cphNeighborhoods = pd.DataFrame({'Neighborhood': cphNames, 'Address': address})
df_cphNeighborhoods['Latitude'] = np.nan
df_cphNeighborhoods['Longitude'] = np.nan

print("Number of rows = {}, number of columns = {}".format(df_cphNeighborhoods.shape[0], df_cphNeighborhoods.shape[1]))
df_cphNeighborhoods.head(3)
```

```
Number of rows = 11, number of columns = 4
```

Out[8]:

	Neighborhood	Address	Latitude	Longitude
0	Amager Vest	Vestamager	NaN	NaN
1	Amager Øst	Øresundsvej	NaN	NaN
2	Bispebjerg	Bispebjerg	NaN	NaN

I use Nominatim to find and add the geographical locations of the neighborhoods.

In [9]:
```
for name in df_cphNeighborhoods['Address']:
    address = name + ', København, Danmark'
    geolocator = Nominatim(user_agent='dk.pawhermansen')
    location = geolocator.geocode(address)
    if location != None:
        df_cphNeighborhoods.loc[df_cphNeighborhoods['Address'] == name, 'Latitude'] = location.latitude
        df_cphNeighborhoods.loc[df_cphNeighborhoods['Address'] == name, 'Longitude'] = location.longitude

print("Number of rows = {}, number of columns = {}".format(df_cphNeighborhoods.shape[0], df_cphNeighborhoods.shape[1]))
df_cphNeighborhoods.head(3)
```

```
Number of rows = 11, number of columns = 4
```

Out[9]:

	Neighborhood	Address	Latitude	Longitude
0	Amager Vest	Vestamager	55.619371	12.575584
1	Amager Øst	Øresundsvej	55.661537	12.626487
2	Bispebjerg	Bispebjerg	55.710950	12.534000

As explained for the Paris neighborhoods I want to find appropriate size of the search-areas and I computer the distance in meter to each neighborhoods closest neighbor. Finally I remove the address column again and save the Copenhagen neighborhoods to a local csv file.

In [10]:
```
df_cphNeighborhoods['Distance to Nearest'] = nearestDistances(df_cphNeighborhoods)
df_cphNeighborhoods = df_cphNeighborhoods.drop(columns=['Address'])

df_cphNeighborhoods.to_csv('data/cph_neighborhoods.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of neighborhoods is", df_cphNeighborhoods.shape[0])
df_cphNeighborhoods
```

```
Number of neighborhoods is 11
```

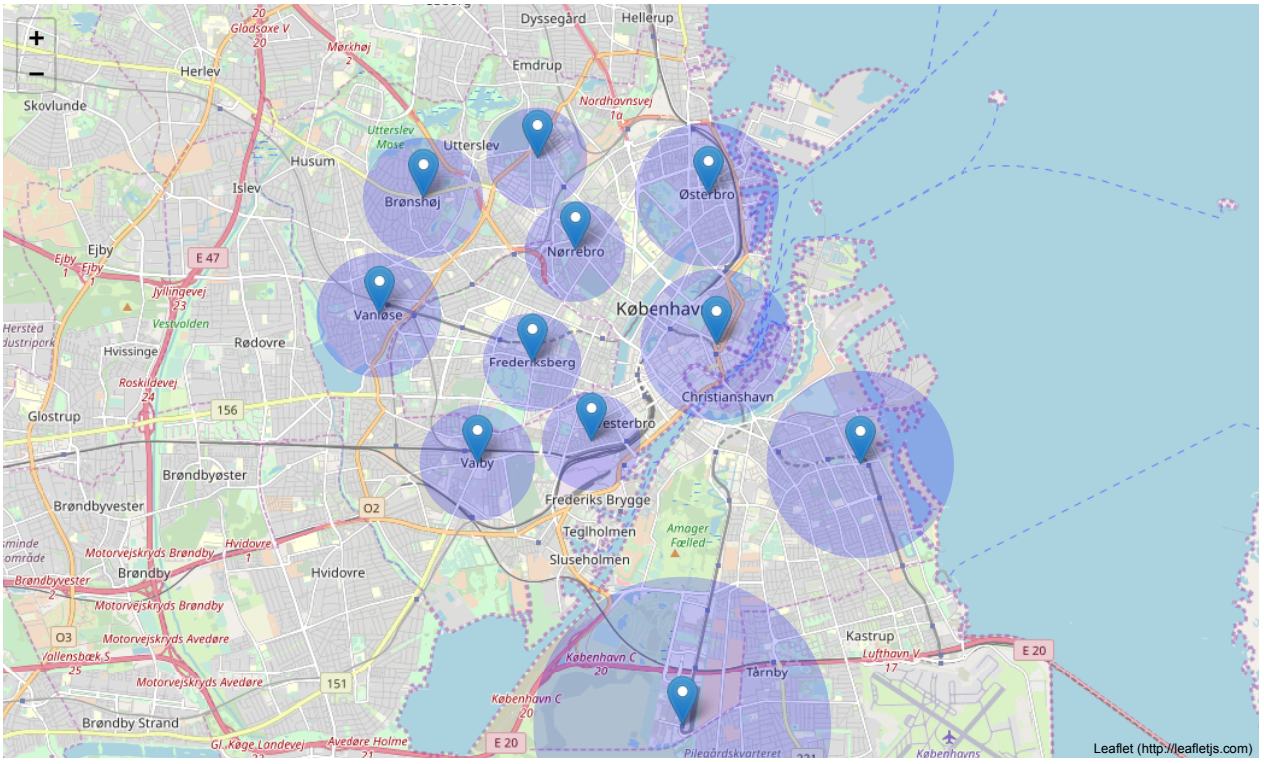
Out[10]:

	Neighborhood	Latitude	Longitude	Distance to Nearest
0	Amager Vest	55.619371	12.575584	5364.845473
1	Amager Øst	55.661537	12.626487	3372.547947
2	Bispebjerg	55.710950	12.534000	1812.305497
3	Brønshøj-Husum	55.704536	12.501445	2167.519938
4	Indre By	55.680889	12.585253	2698.854219
5	Nørrebro	55.695894	12.544956	1812.305497
6	Valby	55.661802	12.516952	2056.828338
7	Vanløse	55.685625	12.488809	2250.439743
8	Vesterbro/Kongens Enghave	55.665257	12.549573	1776.504126
9	Østerbro	55.705084	12.582614	2579.354408
10	Frederiksberg	55.678016	12.532619	1776.504126

My search-areas for the nighborhoods of Copenhagen looks like this:

In [11]:
```
createTownMap(df_cphNeighborhoods)
```

Out[11]:



Venues for the neighborhoods

Define Foursquare Credentials and Version

I define my personal credentials to Foursquare in environment variables on the machine where I execute the notebook and read them below from the environment variables.

In [12]:
```
CLIENT_ID = os.environ['FOURSQUARE_ID'] # your Foursquare ID
CLIENT_SECRET = os.environ['FOURSQUARE_SECRET'] # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

if CLIENT_ID and CLIENT_SECRET:
    print('Got your credentails!')
else:
    raise ValueError('Foursquare credentials are missing - should be set as environment variables')
```

```
Got your credentails!
```

Define function to call the Foursquare API and find venues nearby a given location

The function is copied from the *Applied Data Science Capstone* course material with my own addition to join categories if a venue has more than one category instead of just taking the first category.

The parameters are lists of:

- *names*: a name of the location or neighborhood.
- *cities*: the city name of the neighborhood.
- *latitudes*: the latitude of the location or neighborhood.
- *longitudes*: the longitude of the location or neighborhood.
- *radius*: the maximal distance in meters to from the location to search for venues. Defaults to 500m.

Returns a dataframes with a row for each found venue within *radius* meter from the given location.

In [13]:
```
def getNearbyVenues(names, cities, latitudes, longitudes, radii):

    venues_list=[]
    for name, city, lat, lng, radius in zip(names, cities, latitudes, longitudes, radii):
        print(name)

        # create the API request URL
        LIMIT = 100  # max according to foursquare documentation
        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,
            city,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            ', '.join([c['name'] for c in v['venue']['categories']])) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                             'City',
                             'Neighborhood Latitude',
                             'Neighborhood Longitude',
                             'Venue',
                             'Venue Latitude',
                             'Venue Longitude',
                             'Venue Category']

    return(nearby_venues)
```

Find the venues from Foursquare

First I create and save a combined table of all the neighborhoods.

In [14]:
```
df_parisNeighborhoods['City'] = 'Paris'
df_cphNeighborhoods['City'] = 'Copenhagen'

df_neighborhoods = df_cphNeighborhoods.append(df_parisNeighborhoods).reset_index(drop=True)
df_neighborhoods = df_neighborhoods[['Neighborhood', 'City', 'Latitude', 'Longitude', 'Distance to Nearest']]

df_neighborhoods.to_csv('data/neighborhoods.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_neighborhoods.shape[0], df_neighborhoods.shape[1]))
df_neighborhoods.head(3)
```

```
Number of rows = 31, number of columns = 5
```

Out[14]:

	Neighborhood	City	Latitude	Longitude	Distance to Nearest
0	Amager Vest	Copenhagen	55.619371	12.575584	5364.845473
1	Amager Øst	Copenhagen	55.661537	12.626487	3372.547947
2	Bispebjerg	Copenhagen	55.710950	12.534000	1812.305497

Then I find the venues for each neighborhood.

In [15]:
```
df_venues = getNearbyVenues(df_neighborhoods['Neighborhood'],
                            df_neighborhoods['City'],
                            df_neighborhoods['Latitude'],
                            df_neighborhoods['Longitude'],
                            df_neighborhoods['Distance to Nearest'] / 2)

print()
print('Total number of found venues for all neighborhoods: ', df_venues.shape[0])
df_venues.head(3)
```

```
Amager Vest
Amager Øst
Bispebjerg
Brønshøj-Husum
Indre By
Nørrebro
Valby
Vanløse
Vesterbro/Kongens Enghave
Østerbro
Frederiksberg
Bourse
Temple
Reuilly
Louvre
Hôtel-de-Ville
Élysée
Observatoire
Buttes-Chaumont
Ménilmontant
Luxembourg
Opéra
Batignolles-Monceau
Vaugirard
Panthéon
Palais-Bourbon
Entrepôt
Popincourt
Gobelins
Passy
Buttes-Montmartre

Total number of found venues for all neighborhoods:  2775
```

Out[15]:

	Neighborhood	City	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Amager Vest	Copenhagen	55.619371	12.575584	Naturcenter Amager	55.614620	12.574853	Park
1	Amager Vest	Copenhagen	55.619371	12.575584	8tallet	55.618353	12.571841	Building
2	Amager Vest	Copenhagen	55.619371	12.575584	Royal Arena	55.625469	12.573884	Event Space

No Multiple Categories were found

In the earlier defined function *getNearbyVenues* I separate multiple categories in the *Venue Category* by a comma. However, as observed from the count of zero in the next cell, it turns out that none of the results from Foursquare actually has multiple categories and so nothing more needs to be done.

In [16]:
```
df_venues['Venue Category'].str.contains(',').sum()
```

Out[16]: 0

'Neighborhood' is a venue category

In [17]:
```
df_venues[df_venues['Venue Category'] == 'Neighborhood']
```

Out[17]:

	Neighborhood	City	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
599	Vesterbro/Kongens Enghave	Copenhagen	55.665257	12.549573	Kødbyen	55.668336	12.558662	Neighborhood

Later in this notebook I will use onehot encoding of the venue categories. Onehot encoding will create a column with the name 'Neighborhood' for the venue category. But I already use the column name 'Neighborhood' for the name of the neighborhood and two different columns cannot have the same name. To solve this I decide to rename the venue category 'Neighborhood' to 'Locality'.

In [18]:
```
df_venues['Venue Category'] = df_venues['Venue Category'].str.replace('Neighborhood', 'Locality')

df_venues[df_venues['Venue Category'] == 'Locality']
```

Out[18]:

	Neighborhood	City	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
599	Vesterbro/Kongens Enghave	Copenhagen	55.665257	12.549573	Kødbyen	55.668336	12.558662	Locality

Save the Venues

Finally I can save the venues table to a local csv file.

In [19]:
```
df_venues.to_csv('data/venues.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_venues.shape[0], df_venues.shape[1]))
df_venues.head(3)
```

```
Number of rows = 2775, number of columns = 8
```

Out[19]:

	Neighborhood	City	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Amager Vest	Copenhagen	55.619371	12.575584	Naturcenter Amager	55.614620	12.574853	Park
1	Amager Vest	Copenhagen	55.619371	12.575584	8tallet	55.618353	12.571841	Building
2	Amager Vest	Copenhagen	55.619371	12.575584	Royal Arena	55.625469	12.573884	Event Space

5. Data Understanding

More Paris neighborhoods than Copenhagen neighborhoods

It is clear from the earlier shown lists of neighborhoods that the number of Paris neighborhoods are nearly the double of the number of Copenhagen neighborhoods. This must be of course be remembered later when comparing how many neighborhoods from each city are in the found clusters.

Number of Unique Venues Categories

In [20]:
```
df_categories = pd.DataFrame(df_venues['Venue Category'].unique(), columns = ['Venue Category'])

print('There are {} unique categories.'.format(len(df_categories)))
```

```
There are 291 unique categories.
```

That is quite a lot of different venue categories and there will certainly be no problems in expressing the differencies in the neighborhoods.

On the other hand, the venue categories might be too detailed, for example with restaurants that are categorized by their kitchens originating country. After seeing the first results of the clustering it might become relevant to consider if, for example, a Scandinavian Restaurant in Copenhagen should or should not be counted as being different from a French Restaurant in Paris.

In [21]:
```
df_restaurantCategories = df_categories[df_categories['Venue Category'].str.contains("Restaurant")]

print('Number of different Restaurant categories in the venues data is', len(df_restaurantCategories))
df_restaurantCategories.head(10)
```

```
Number of different Restaurant categories in the venues data is 65
```

Out[21]:

	Venue Category
9	Scandinavian Restaurant
11	Tapas Restaurant
20	Fast Food Restaurant
24	Indian Restaurant
25	Restaurant
45	Sushi Restaurant
55	Italian Restaurant
59	Thai Restaurant
62	Chinese Restaurant
66	American Restaurant

Counting the found venues in each neighborhood

The following table shows how many venues were found from FourSquare for each neighborhood.

It is seen that two of the neighborhoods have only around 30 venues and this is in the lower end of necessary venues in each neighborhood. However because it is only two neighborhoods and the rest have above fifty venues each I will use them as they are.

On the other hand we see that several neighborhoods have exactly 100 found venues and that no neighborhood has more than 100 venues. This is caused by the FourSquare API that has this as a maximum. I assume that the returned venues are representative for all the venues in the neighborhood and use them as they are.

In [22]:
```
df_venues.groupby('Neighborhood').size().reset_index(name='Venue count')
```

Out[22]:

	Neighborhood	Venue count
0	Amager Vest	58
1	Amager Øst	97
2	Batignolles-Monceau	100
3	Bispebjerg	38
4	Bourse	100
5	Brønshøj-Husum	24
6	Buttes-Chaumont	84
7	Buttes-Montmartre	100
8	Entrepôt	100
9	Frederiksberg	91
10	Gobelins	100
11	Hôtel-de-Ville	100
12	Indre By	100
13	Louvre	92
14	Luxembourg	100
15	Ménilmontant	88
16	Nørrebro	100
17	Observatoire	100
18	Opéra	100
19	Palais-Bourbon	100
20	Panthéon	100
21	Passy	100
22	Popincourt	100
23	Reuilly	100
24	Temple	100
25	Valby	65
26	Vanløse	54
27	Vaugirard	100
28	Vesterbro/Kongens Enghave	100
29	Élysée	100
30	Østerbro	84

Size of the Search Area and the Number of Venues from Foursquare

The table below shows that Foursquare returned three times as many venues per square kilometer for Paris when compared to Copenhagen.

This could indicate that Paris have more venues that are interesting enough to make it into Foursquare but I think it is much more likely that the Foursquare app is more popular in France than in Denmark and I consider this fact as having no influence on the results in this notebook.

In [23]:
```
df_parisSearchAreas = np.square(df_parisNeighborhoods['Distance to Nearest'] / 2) * 3.1416
df_cphSearchAreas = np.square(df_cphNeighborhoods['Distance to Nearest'] / 2) * 3.1416

df_venuesByCity = df_venues.groupby('City').size().reset_index(name='Venue count')
df_venuesByCity['Search Area in m2'] = [df_cphSearchAreas.sum(), df_parisSearchAreas.sum()]
df_venuesByCity['Venues per km2'] = 1e6 * df_venuesByCity['Venue count'] / df_venuesByCity['Search Area in m2']
df_venuesByCity['Venues per km2'] = df_venuesByCity['Venues per km2'].map('{:,.2f}'.format)

df_venuesByCity
```

Out[23]:

	City	Venue count	Search Area in m2	Venues per km2
0	Copenhagen	811	6.359111e+07	12.75
1	Paris	1964	5.360275e+07	36.64

6. Data Preparation

Frequency of Venue Categories for Each Neighborhood

To be used with the clustering algorithm I need a table of the frequencies of occurence of each venue category for each neighborhood.

First I arrange the venues in so-called onehot encoding.

In [24]:
```
# one hot encoding
df_onehot = pd.get_dummies(df_venues[['Venue Category']], sparse=False, prefix="", prefix_sep="")

# add neighborhood and city columns back to dataframe
df_onehot.insert(0, 'Neighborhood', df_venues['Neighborhood'])
df_onehot.insert(1, 'City', df_venues['City'])

print(df_onehot.shape)
df_onehot.head(3)
```

```
(2775, 293)
```

Out[24]:

	Neighborhood	City	Accessories Store	Advertising Agency	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Aquarium	Arepa Restaurant	...	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Vietnamese Restaurant	V
0	Amager Vest	Copenhagen	0	0	0	0	0	0	0	0	...	0	0	0	
1	Amager Vest	Copenhagen	0	0	0	0	0	0	0	0	...	0	0	0	
2	Amager Vest	Copenhagen	0	0	0	0	0	0	0	0	...	0	0	0	

3 rows × 293 columns

Onehot encoding works by, like the venues table, to have one row for each venue. The columns, however, are very different in that we have one column for each unique venue category (plus the Neighborhood column with the name of the neighborhood). Each row has the value 0 (zero) all over except in the column of its venue category where it has the value of 1 (one).

We saw earlier that the first three rows in the venues table are 'Amager Vest' having the venue categories 'Park', 'Building', and 'Event Space' respectively. The following example selection of the onehot table illustrates this.

In [25]:
```
df_onehot.loc[0:2, ['Neighborhood', 'Accessories Store', 'Building', 'Event Space', 'Park', 'Udon Restaurant']]
```

Out[25]:

	Neighborhood	Accessories Store	Building	Event Space	Park	Udon Restaurant
0	Amager Vest	0	0	0	1	0
1	Amager Vest	0	1	0	0	0
2	Amager Vest	0	0	1	0	0

Next I find the frequencies of the venue categories for each neighborhood by taking the mean of rows of each neighborhood.

Basically this is taking the sum of the rows of each neighborhood to find how many of each venue category each neighborhood has. However, this gives different weight to each neighborhood because they have a different number of found venues. With the frequencies the sum of each row is 1 (one) and in this way the weight of each neigborhood is the same.

In [26]:
```
df_freq = df_onehot.groupby(['Neighborhood', 'City']).mean().reset_index()

print(df_freq.shape)
df_freq.head()
```

```
(31, 293)
```

Out[26]:

	Neighborhood	City	Accessories Store	Advertising Agency	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Aquarium	Arepa Restaurant	...	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Vietnamese Restaurant	,
0	Amager Vest	Copenhagen	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.0	0.0	0.00	
1	Amager Øst	Copenhagen	0.0	0.0	0.0	0.0	0.010309	0.010309	0.0	0.0	...	0.0	0.0	0.00	
2	Batignolles-Monceau	Paris	0.0	0.0	0.0	0.0	0.010000	0.000000	0.0	0.0	...	0.0	0.0	0.01	
3	Bispebjerg	Copenhagen	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.0	0.0	0.00	
4	Bourse	Paris	0.0	0.0	0.0	0.0	0.000000	0.010000	0.0	0.0	...	0.0	0.0	0.00	

5 rows × 293 columns

Finally I save the frequency table to a local csv file.

In [27]:
```
df_freq.to_csv('data/frequencies.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_freq.shape[0], df_freq.shape[1]))
df_freq.head(3)
```

```
Number of rows = 31, number of columns = 293
```

Out[27]:

	Neighborhood	City	Accessories Store	Advertising Agency	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Aquarium	Arepa Restaurant	...	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Vietnamese Restaurant	,
0	Amager Vest	Copenhagen	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	...	0.0	0.0	0.00	
1	Amager Øst	Copenhagen	0.0	0.0	0.0	0.0	0.010309	0.010309	0.0	0.0	...	0.0	0.0	0.00	
2	Batignolles-Monceau	Paris	0.0	0.0	0.0	0.0	0.010000	0.000000	0.0	0.0	...	0.0	0.0	0.01	

3 rows × 293 columns

Frequency of Simplified Venue Categories for Each Neighborhood

I now repeat the previous steps but on venue data where I change all the different types of restaurants, like *French Restaurant*, *Scandinavian Restaurant*, *Japaneese Restaurant*, to just being of the venue type *Restaurant*.

First I change the venue type and then I arrange the simplified venues in onehot encoding. See the preceeding section to see how the onehot encoding works.

In [28]:
```
# Take a copy of the venues table and make all restaurants the same
df_venuesSimplified = df_venues.copy()
df_venuesSimplified.loc[df_venuesSimplified['Venue Category'].str.contains('Restaurant'), 'Venue Category'] = 'Restaurant'

# one hot encoding
df_onehot = pd.get_dummies(df_venuesSimplified[['Venue Category']], sparse=False, prefix="", prefix_sep="")

# add neighborhood and city columns back to dataframe
df_onehot.insert(0, 'Neighborhood', df_venuesSimplified['Neighborhood'])
df_onehot.insert(1, 'City', df_venuesSimplified['City'])

print(df_onehot.shape)
df_onehot.head(3)
```

```
(2775, 229)
```

Out[28]:

	Neighborhood	City	Accessories Store	Advertising Agency	Antique Shop	Aquarium	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	...	Trail	Train Station	Tram Station	Vineyard	Water Park	Wine Bar
0	Amager Vest	Copenhagen	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
1	Amager Vest	Copenhagen	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2	Amager Vest	Copenhagen	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

3 rows × 229 columns

Next I find the frequencies of the simplified venue categories for each neighborhood by taking the mean of rows of each neighborhood.

In [29]:
```
df_freqSimplified = df_onehot.groupby(['Neighborhood', 'City']).mean().reset_index()

print(df_freqSimplified.shape)
df_freqSimplified.head()
```

```
(31, 229)
```

Out[29]:

	Neighborhood	City	Accessories Store	Advertising Agency	Antique Shop	Aquarium	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	...	Trail	Train Station	Tram Station	Vineyard	Water Park	Wine Bar
0	Amager Vest	Copenhagen	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.00
1	Amager Øst	Copenhagen	0.0	0.0	0.010309	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.00
2	Batignolles-Monceau	Paris	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.01
3	Bispebjerg	Copenhagen	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.026316	...	0.0	0.0	0.0	0.0	0.0	0.00
4	Bourse	Paris	0.0	0.0	0.010000	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.04

5 rows × 229 columns

Finally I save the frequency table for simplified venue types to a local csv file.

In [30]:
```
df_freqSimplified.to_csv('data/frequenciesSimplified.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_freqSimplified.shape[0], df_freqSimplified.shape[1]))
df_freqSimplified.head(3)
```

```
Number of rows = 31, number of columns = 229
```

Out[30]:

	Neighborhood	City	Accessories Store	Advertising Agency	Antique Shop	Aquarium	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	...	Trail	Train Station	Tram Station	Vineyard	Water Park	Wine Bar
0	Amager Vest	Copenhagen	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.00
1	Amager Øst	Copenhagen	0.0	0.0	0.010309	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.00
2	Batignolles-Monceau	Paris	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.01

3 rows × 229 columns

Human Readable Table of Top Ten Venue Categories

To be able to compare the neighborhoods myself after clustering them, I create a table of the top ten venue categories for each neighborhood and save it as a local csv file.

In [31]:

```
# Create the columns
columns = ['Neighborhood', 'City']
indicators = {1: 'st', 2: 'nd', 3: 'rd'}
for i in range(1,11):
    columns.append('{}{} Most Common Venue'.format(i, indicators.get(i, 'th')))

# Create the dataframe with the coloumn names and add the neighborhood names and cities
df_topTenVenues = pd.DataFrame(columns=columns)
df_topTenVenues['Neighborhood'] = df_freq['Neighborhood']
df_topTenVenues['City'] = df_freq['City']

# Add all the neighborhoods as rows with the top ten venue categories
for i in range(df_freq.shape[0]):
    df_topTenVenues.iloc[i, 2:] = df_freq.iloc[i, 2:].sort_values(ascending=False).head(10).index.values

# Save the table
df_topTenVenues.to_csv('data/topTenVenues.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_topTenVenues.shape[0], df_topTenVenues.shape[1]))
df_topTenVenues
```

```
Number of rows = 31, number of columns = 12
```

Out[31]:

	Neighborhood	City	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	9th Most Common Venue	10th Most Common Venue
0	Amager Vest	Copenhagen	Hotel	Park	Supermarket	Pizza Place	Restaurant	Golf Course	Café	Gym / Fitness Center	Grocery Store	Indian Restaurant
1	Amager Øst	Copenhagen	Beach	Bakery	Coffee Shop	Café	Pizza Place	Gym / Fitness Center	Burger Joint	Sushi Restaurant	Grocery Store	Chinese Restaurant
2	Batignolles-Monceau	Paris	French Restaurant	Hotel	Italian Restaurant	Bakery	Restaurant	Bistro	Plaza	Pastry Shop	Japanese Restaurant	Bar
3	Bispebjerg	Copenhagen	Pizza Place	Grocery Store	Café	Convenience Store	Supermarket	Sushi Restaurant	Thai Restaurant	Gym / Fitness Center	Dance Studio	Plaza
4	Bourse	Paris	French Restaurant	Bistro	Wine Bar	Japanese Restaurant	Salad Place	Plaza	Cocktail Bar	Italian Restaurant	Bar	Hotel
5	Brønshøj-Husum	Copenhagen	Bakery	Park	Supermarket	Plaza	Grocery Store	Martial Arts Dojo	Café	Theater	Scandinavian Restaurant	Bus Stop
6	Buttes-Chaumont	Paris	French Restaurant	Bar	Pizza Place	Café	Multiplex	Beer Bar	Bed & Breakfast	Brasserie	Japanese Restaurant	Scenic Lookout
7	Buttes-Montmartre	Paris	French Restaurant	Bar	Bistro	Pizza Place	Italian Restaurant	Restaurant	Plaza	Café	Bookstore	Vegetarian / Vegan Restaurant
8	Entrepôt	Paris	Coffee Shop	French Restaurant	Bistro	Pizza Place	Cocktail Bar	Indian Restaurant	Italian Restaurant	Asian Restaurant	Restaurant	Seafood Restaurant
9	Frederiksberg	Copenhagen	Café	Scandinavian Restaurant	Park	Bakery	French Restaurant	Italian Restaurant	Bar	Sushi Restaurant	Theater	Coffee Shop
10	Gobelins	Paris	Vietnamese Restaurant	Thai Restaurant	Asian Restaurant	Chinese Restaurant	French Restaurant	Hotel	Bakery	Supermarket	Sushi Restaurant	Cantonese Restaurant
11	Hôtel-de-Ville	Paris	French Restaurant	Hotel	Ice Cream Shop	Plaza	Wine Bar	Pastry Shop	Bakery	Tea Room	Garden	Seafood Restaurant
12	Indre By	Copenhagen	Coffee Shop	Restaurant	Scandinavian Restaurant	Cocktail Bar	Café	Wine Bar	Bar	Furniture / Home Store	Plaza	Theater
13	Louvre	Paris	French Restaurant	Hotel	Plaza	Café	Exhibit	Coffee Shop	Japanese Restaurant	Historic Site	Bar	Art Museum
14	Luxembourg	Paris	French Restaurant	Italian Restaurant	Hotel	Wine Bar	Bistro	Tea Room	Bakery	Pastry Shop	Chocolate Shop	Plaza
15	Ménilmontant	Paris	French Restaurant	Bar	Plaza	Bakery	Supermarket	Bistro	Japanese Restaurant	Italian Restaurant	Café	Hotel
16	Nørrebro	Copenhagen	Coffee Shop	Pizza Place	Café	Bakery	Thai Restaurant	Gym / Fitness Center	Playground	Wine Bar	Park	Beer Bar
17	Observatoire	Paris	French Restaurant	Hotel	Italian Restaurant	Bar	Japanese Restaurant	Bakery	Sushi Restaurant	Indian Restaurant	Bistro	Pizza Place
18	Opéra	Paris	French Restaurant	Hotel	Cocktail Bar	Japanese Restaurant	Bistro	Italian Restaurant	Bar	Wine Bar	Bakery	Vegetarian / Vegan Restaurant
19	Palais-Bourbon	Paris	French Restaurant	Hotel	Plaza	Italian Restaurant	Café	History Museum	Historic Site	Garden	Cocktail Bar	Bakery
20	Panthéon	Paris	French Restaurant	Bar	Italian Restaurant	Café	Wine Bar	Bakery	Plaza	Pub	Hotel	Greek Restaurant
21	Passy	Paris	French Restaurant	Bakery	Italian Restaurant	Café	Supermarket	Garden	Park	Pizza Place	Bistro	Gym / Fitness Center
22	Popincourt	Paris	Bar	French Restaurant	Cocktail Bar	Bistro	Italian Restaurant	Restaurant	Vietnamese Restaurant	Beer Bar	Pizza Place	Wine Bar
23	Reuilly	Paris	French Restaurant	Japanese Restaurant	Hotel	Italian Restaurant	Plaza	Supermarket	Bakery	Pizza Place	Asian Restaurant	Restaurant
24	Temple	Paris	French Restaurant	Bistro	Hotel	Café	Art Gallery	Italian Restaurant	Coffee Shop	Burger Joint	Pizza Place	Boutique
25	Valby	Copenhagen	Coffee Shop	Café	Pizza Place	Gym / Fitness Center	Bakery	Grocery Store	Sushi Restaurant	Supermarket	Indian Restaurant	Thai Restaurant
26	Vanløse	Copenhagen	Grocery Store	Convenience Store	Pizza Place	Soccer Field	Supermarket	Train Station	Gym	Scandinavian Restaurant	Café	Bakery
27	Vaugirard	Paris	French Restaurant	Hotel	Italian Restaurant	Bakery	Lebanese Restaurant	Coffee Shop	Korean Restaurant	Park	Japanese Restaurant	Indian Restaurant
28	Vesterbro/Kongens Enghave	Copenhagen	Café	Cocktail Bar	Coffee Shop	Pizza Place	Scandinavian Restaurant	Wine Bar	Italian Restaurant	Music Venue	Park	Bakery
29	Élysée	Paris	Hotel	French Restaurant	Art Gallery	Italian Restaurant	Clothing Store	Bakery	Cosmetics Shop	Salad Place	Japanese Restaurant	Bar
30	Østerbro	Copenhagen	Harbor / Marina	Coffee Shop	Bakery	Playground	Pizza Place	Soccer Stadium	Gym	Ice Cream Shop	Plaza	Platform

In the same way I create a table of the top ten simplified venue categories for each neighborhood and save it as a local csv file.

In [32]:

```
# Create the columns
columns = ['Neighborhood', 'City']
indicators = {1: 'st', 2: 'nd', 3: 'rd'}
for i in range(1,11):
    columns.append('{}{} Most Common Venue'.format(i, indicators.get(i, 'th')))

# Create the dataframe with the coloumn names and add the neighborhood names and cities
df_topTenSimplifiedVenues = pd.DataFrame(columns=columns)
df_topTenSimplifiedVenues['Neighborhood'] = df_freqSimplified['Neighborhood']
df_topTenSimplifiedVenues['City'] = df_freqSimplified['City']

# Add all the neighborhoods as rows with the top ten venue categories
for i in range(df_freqSimplified.shape[0]):
    df_topTenSimplifiedVenues.iloc[i, 2:] = df_freqSimplified.iloc[i, 2:].sort_values(ascending=False).head(10).index.value
s

# Save the table
df_topTenSimplifiedVenues.to_csv('data/topTenSimplifiedVenues.csv', quoting=csv.QUOTE_ALL, index=False)

print("Number of rows = {}, number of columns = {}".format(df_topTenSimplifiedVenues.shape[0], df_topTenSimplifiedVenues.sh
ape[1]))
df_topTenSimplifiedVenues
```

```
Number of rows = 31, number of columns = 12
```

Out[32]:

	Neighborhood	City	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	9th Most Common Venue	10th Most Common Venue
0	Amager Vest	Copenhagen	Restaurant	Hotel	Supermarket	Park	Pizza Place	Coffee Shop	Café	Golf Course	Grocery Store	Gym / Fitness Center
1	Amager Øst	Copenhagen	Restaurant	Beach	Café	Coffee Shop	Bakery	Gym / Fitness Center	Burger Joint	Pizza Place	Grocery Store	Sports Club
2	Batignolles-Monceau	Paris	Restaurant	Hotel	Bakery	Bistro	Plaza	Pastry Shop	Cheese Shop	Park	Bar	Pizza Place
3	Bispebjerg	Copenhagen	Restaurant	Pizza Place	Café	Grocery Store	Dance Studio	Convenience Store	Gym / Fitness Center	Supermarket	Hostel	Farmers Market
4	Bourse	Paris	Restaurant	Bistro	Wine Bar	Plaza	Bar	Cocktail Bar	Hotel	Salad Place	Lounge	Pizza Place
5	Brønshøj-Husum	Copenhagen	Restaurant	Park	Supermarket	Bakery	Theater	Bus Stop	Snack Place	Bowling Alley	Café	Martial Arts Dojo
6	Buttes-Chaumont	Paris	Restaurant	Bar	Café	Pizza Place	Multiplex	Brasserie	Beer Bar	Pool	Canal	Hotel
7	Buttes-Montmartre	Paris	Restaurant	Bar	Bistro	Pizza Place	Plaza	Café	Deli / Bodega	Gastropub	Convenience Store	Sandwich Place
8	Entrepôt	Paris	Restaurant	Coffee Shop	Bistro	Pizza Place	Cocktail Bar	Wine Shop	Café	Breakfast Spot	Bakery	Burger Joint
9	Frederiksberg	Copenhagen	Restaurant	Café	Park	Bakery	Bar	Pizza Place	Exhibit	Burger Joint	Theater	Coffee Shop
10	Gobelins	Paris	Restaurant	Hotel	Bakery	Bistro	Park	Bar	Supermarket	Café	Farmers Market	Convenience Store
11	Hôtel-de-Ville	Paris	Restaurant	Hotel	Ice Cream Shop	Wine Bar	Pastry Shop	Bakery	Plaza	Tea Room	Garden	Art Gallery
12	Indre By	Copenhagen	Restaurant	Coffee Shop	Cocktail Bar	Café	Furniture / Home Store	Wine Bar	Bar	Plaza	Clothing Store	Steakhouse
13	Louvre	Paris	Restaurant	Hotel	Café	Plaza	Exhibit	Coffee Shop	Art Museum	Bar	Historic Site	Tea Room
14	Luxembourg	Paris	Restaurant	Hotel	Wine Bar	Tea Room	Bistro	Women's Store	Pastry Shop	Ice Cream Shop	Chocolate Shop	Bakery
15	Ménilmontant	Paris	Restaurant	Bar	Plaza	Bakery	Supermarket	Bistro	Café	Hotel	Pizza Place	Music Venue
16	Nørrebro	Copenhagen	Restaurant	Coffee Shop	Pizza Place	Café	Bakery	Playground	Gym / Fitness Center	Wine Bar	Park	Beer Bar
17	Observatoire	Paris	Restaurant	Hotel	Bar	Pizza Place	Bistro	Bakery	Wine Bar	Plaza	Coffee Shop	Art Museum
18	Opéra	Paris	Restaurant	Hotel	Cocktail Bar	Bistro	Bakery	Wine Bar	Bar	Pizza Place	Sandwich Place	Cupcake Shop
19	Palais-Bourbon	Paris	Restaurant	Hotel	Plaza	Café	History Museum	Cocktail Bar	Historic Site	Garden	Ice Cream Shop	Coffee Shop
20	Panthéon	Paris	Restaurant	Café	Bar	Wine Bar	Plaza	Bakery	Hotel	Pub	Coffee Shop	Garden
21	Passy	Paris	Restaurant	Bakery	Café	Garden	Supermarket	Pizza Place	Bistro	Gym / Fitness Center	Park	Plaza
22	Popincourt	Paris	Restaurant	Bar	Cocktail Bar	Bistro	Wine Bar	Beer Bar	Pizza Place	Pub	Bakery	Record Shop
23	Reuilly	Paris	Restaurant	Hotel	Plaza	Supermarket	Bakery	Park	Tram Station	Café	Sandwich Place	Pizza Place
24	Temple	Paris	Restaurant	Bistro	Coffee Shop	Hotel	Art Gallery	Café	Pizza Place	Sandwich Place	Boutique	Burger Joint
25	Valby	Copenhagen	Restaurant	Coffee Shop	Café	Pizza Place	Bakery	Gym / Fitness Center	Grocery Store	Supermarket	Burger Joint	Ice Cream Shop
26	Vanløse	Copenhagen	Restaurant	Grocery Store	Convenience Store	Pizza Place	Supermarket	Soccer Field	Train Station	Gym / Fitness Center	Café	Lake
27	Vaugirard	Paris	Restaurant	Hotel	Bakery	Park	Coffee Shop	Bistro	Gym / Fitness Center	Gym	Bar	Gastropub
28	Vesterbro/Kongens Enghave	Copenhagen	Restaurant	Café	Cocktail Bar	Coffee Shop	Pizza Place	Wine Bar	Music Venue	Bakery	Park	Bistro
29	Élysée	Paris	Restaurant	Hotel	Art Gallery	Clothing Store	Bakery	Theater	Garden	Cosmetics Shop	Bar	Salad Place
30	Østerbro	Copenhagen	Restaurant	Harbor / Marina	Coffee Shop	Bakery	Playground	Pizza Place	Soccer Stadium	Ice Cream Shop	Gym	Theater

7. Analyses

Method 1: Cluster into Two Clusters

Introduction and Clustering

First I want to see how well the Copenhagen and Paris neighborhoods mix across the cities if I cluster them into two groups.

In [33]:
```
kclusters = 2
df_freqClustering = df_freq.copy().drop(['Neighborhood', 'City'], 1)

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

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

Out[33]:
```
array([1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 1], dtype=int32)
```

Result and Discussion

To visualize the result I create a table for each clustered group that shows the neighborhoods of the group and the neighborhoods top 10 venue types.

First I create a function to do the work for this and the following experiments.

It is easy to see from the clustered groups that all but one, Entrepôt, of the Paris neighborhoods are clustered together in one group and all the Copenhagen neighborhoods and *Entrepôt* are clustered together in another group. *Entrepôt* has many coffee shops wich seems to be common in Copenhagen and less common in Paris.

This at least shows that a single neighborhood in Paris has many coffee shops like you find many places in Copenhagen.

In [34]:
```
def printClusters(df, labels):
    df.insert(2, 'Cluster', labels)
    for cluster in range(kclusters):
        countCopenhagen = df[(df['Cluster'] == cluster) & (df['City'] == 'Copenhagen')].shape[0]
        countParis = df[(df['Cluster'] == cluster) & (df['City'] == 'Paris')].shape[0]
        print()
        display(Markdown('**Group {} with {} Copenhagen and {} Paris neighborhoods:**'.format(cluster, countCopenhagen, cou
ntParis)))
        display(df[df['Cluster'] == cluster])
```

In [35]:
```
printClusters(df_topTenVenues.copy(), kmeans.labels_)
```

Group 0 with 0 Copenhagen and 19 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
2	Batignolles-Monceau	Paris	0	French Restaurant	Hotel	Italian Restaurant	Bakery	Restaurant	Bistro	Plaza	Pastry Shop	Japanese Restaurant	Bar
4	Bourse	Paris	0	French Restaurant	Bistro	Wine Bar	Japanese Restaurant	Salad Place	Plaza	Cocktail Bar	Italian Restaurant	Bar	Hotel
6	Buttes-Chaumont	Paris	0	French Restaurant	Bar	Pizza Place	Café	Multiplex	Beer Bar	Bed & Breakfast	Brasserie	Japanese Restaurant	Scenic Lookout
7	Buttes-Montmartre	Paris	0	French Restaurant	Bar	Bistro	Pizza Place	Italian Restaurant	Restaurant	Plaza	Café	Bookstore	Vegetarian / Vegan Restaurant
10	Gobelins	Paris	0	Vietnamese Restaurant	Thai Restaurant	Asian Restaurant	Chinese Restaurant	French Restaurant	Hotel	Bakery	Supermarket	Sushi Restaurant	Cantonese Restaurant
11	Hôtel-de-Ville	Paris	0	French Restaurant	Hotel	Ice Cream Shop	Plaza	Wine Bar	Pastry Shop	Bakery	Tea Room	Garden	Seafood Restaurant
13	Louvre	Paris	0	French Restaurant	Hotel	Plaza	Café	Exhibit	Coffee Shop	Japanese Restaurant	Historic Site	Bar	Art Museum
14	Luxembourg	Paris	0	French Restaurant	Italian Restaurant	Hotel	Wine Bar	Bistro	Tea Room	Bakery	Pastry Shop	Chocolate Shop	Plaza
15	Ménilmontant	Paris	0	French Restaurant	Bar	Plaza	Bakery	Supermarket	Bistro	Japanese Restaurant	Italian Restaurant	Café	Hotel
17	Observatoire	Paris	0	French Restaurant	Hotel	Italian Restaurant	Bar	Japanese Restaurant	Bakery	Sushi Restaurant	Indian Restaurant	Bistro	Pizza Place
18	Opéra	Paris	0	French Restaurant	Hotel	Cocktail Bar	Japanese Restaurant	Bistro	Italian Restaurant	Bar	Wine Bar	Bakery	Vegetarian / Vegan Restaurant
19	Palais-Bourbon	Paris	0	French Restaurant	Hotel	Plaza	Italian Restaurant	Café	History Museum	Historic Site	Garden	Cocktail Bar	Bakery
20	Panthéon	Paris	0	French Restaurant	Bar	Italian Restaurant	Café	Wine Bar	Bakery	Plaza	Pub	Hotel	Greek Restaurant
21	Passy	Paris	0	French Restaurant	Bakery	Italian Restaurant	Café	Supermarket	Garden	Park	Pizza Place	Bistro	Gym / Fitness Center
22	Popincourt	Paris	0	Bar	French Restaurant	Cocktail Bar	Bistro	Italian Restaurant	Restaurant	Vietnamese Restaurant	Beer Bar	Pizza Place	Wine Bar
23	Reuilly	Paris	0	French Restaurant	Japanese Restaurant	Hotel	Italian Restaurant	Plaza	Supermarket	Bakery	Pizza Place	Asian Restaurant	Restaurant
24	Temple	Paris	0	French Restaurant	Bistro	Hotel	Café	Art Gallery	Italian Restaurant	Coffee Shop	Burger Joint	Pizza Place	Boutique
27	Vaugirard	Paris	0	French Restaurant	Hotel	Italian Restaurant	Bakery	Lebanese Restaurant	Coffee Shop	Korean Restaurant	Park	Japanese Restaurant	Indian Restaurant
29	Élysée	Paris	0	Hotel	French Restaurant	Art Gallery	Italian Restaurant	Clothing Store	Bakery	Cosmetics Shop	Salad Place	Japanese Restaurant	Bar

Group 1 with 11 Copenhagen and 1 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
0	Amager Vest	Copenhagen	1	Hotel	Park	Supermarket	Pizza Place	Restaurant	Golf Course	Café	Gym / Fitness Center	Grocery Store	Indian Restaurant
1	Amager Øst	Copenhagen	1	Beach	Bakery	Coffee Shop	Café	Pizza Place	Gym / Fitness Center	Burger Joint	Sushi Restaurant	Grocery Store	Chinese Restaurant
3	Bispebjerg	Copenhagen	1	Pizza Place	Grocery Store	Café	Convenience Store	Supermarket	Sushi Restaurant	Thai Restaurant	Gym / Fitness Center	Dance Studio	Plaz
5	Brønshøj-Husum	Copenhagen	1	Bakery	Park	Supermarket	Plaza	Grocery Store	Martial Arts Dojo	Café	Theater	Scandinavian Restaurant	Bus Sto
8	Entrepôt	Paris	1	Coffee Shop	French Restaurant	Bistro	Pizza Place	Cocktail Bar	Indian Restaurant	Italian Restaurant	Asian Restaurant	Restaurant	Seafoo Restaurar
9	Frederiksberg	Copenhagen	1	Café	Scandinavian Restaurant	Park	Bakery	French Restaurant	Italian Restaurant	Bar	Sushi Restaurant	Theater	Coffe Sho
12	Indre By	Copenhagen	1	Coffee Shop	Restaurant	Scandinavian Restaurant	Cocktail Bar	Café	Wine Bar	Bar	Furniture / Home Store	Plaza	Theate
16	Nørrebro	Copenhagen	1	Coffee Shop	Pizza Place	Café	Bakery	Thai Restaurant	Gym / Fitness Center	Playground	Wine Bar	Park	Beer Ba
25	Valby	Copenhagen	1	Coffee Shop	Café	Pizza Place	Gym / Fitness Center	Bakery	Grocery Store	Sushi Restaurant	Supermarket	Indian Restaurant	Tha Restaurar
26	Vanløse	Copenhagen	1	Grocery Store	Convenience Store	Pizza Place	Soccer Field	Supermarket	Train Station	Gym	Scandinavian Restaurant	Café	Baker
28	Vesterbro/Kongens Enghave	Copenhagen	1	Café	Cocktail Bar	Coffee Shop	Pizza Place	Scandinavian Restaurant	Wine Bar	Italian Restaurant	Music Venue	Park	Baker
30	Østerbro	Copenhagen	1	Harbor / Marina	Coffee Shop	Bakery	Playground	Pizza Place	Soccer Stadium	Gym	Ice Cream Shop	Plaza	Platforr

Method 2: Cluster into 6 Clusters

Introduction and Clustering

The clustering into two groups gave absolute minimal mixing between Copenhagen and Paris neighborhoods. To see if it becomes different when clustering them into smaller groups I now try to cluster the neighborhoods into six groups giving groups that on the average have about five neighborhoods.

In [36]:
```
kclusters = 6
df_freqClustering = df_freq.copy().drop(['Neighborhood', 'City'], 1)

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

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

Out[36]:
```
array([2, 2, 4, 2, 4, 2, 0, 0, 5, 2, 3, 4, 5, 5, 5, 0, 2, 4, 4, 4, 4, 4,
       0, 4, 5, 2, 2, 4, 2, 4, 1], dtype=int32)
```

Result and Discussion

Again I create a table for each clustered group that shows the neighborhoods of the group and the neighborhoods top 10 venue types.

It turns out that most of the groups have no mixing between the two cities at all and only one group that has minimal mixing with *Indre By* (*Inner City*) of Copenhagen. Again it actually seems like the coffee shops plays a role in the clustering but also cafés, bars, wine bars and plaza might play a role.

This result is perhaps not a very strong proof of likeness between Copenhagen and Paris but at least it suggests that *Inner City* of Copenhagen is somewhat like the Paris neighborhoods *Entrepôt*, *Louvre*, *Luxembourg*, and *Temple*.

In [37]:
```
printClusters(df_topTenVenues.copy(), kmeans.labels_)
```

Group 0 with 0 Copenhagen and 4 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
6	Buttes-Chaumont	Paris	0	French Restaurant	Bar	Pizza Place	Café	Multiplex	Beer Bar	Bed & Breakfast	Brasserie	Japanese Restaurant	Scenic Lookout
7	Buttes-Montmartre	Paris	0	French Restaurant	Bar	Bistro	Pizza Place	Italian Restaurant	Restaurant	Plaza	Café	Bookstore	Vegetarian / Vegan Restaurant
15	Ménilmontant	Paris	0	French Restaurant	Bar	Plaza	Bakery	Supermarket	Bistro	Japanese Restaurant	Italian Restaurant	Café	Hotel
22	Popincourt	Paris	0	Bar	French Restaurant	Cocktail Bar	Bistro	Italian Restaurant	Restaurant	Vietnamese Restaurant	Beer Bar	Pizza Place	Wine Bar

Group 1 with 1 Copenhagen and 0 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
30	Østerbro	Copenhagen	1	Harbor / Marina	Coffee Shop	Bakery	Playground	Pizza Place	Soccer Stadium	Gym	Ice Cream Shop	Plaza	Platform

Group 2 with 9 Copenhagen and 0 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
0	Amager Vest	Copenhagen	2	Hotel	Park	Supermarket	Pizza Place	Restaurant	Golf Course	Café	Gym / Fitness Center	Grocery Store	Indian Restaurant
1	Amager Øst	Copenhagen	2	Beach	Bakery	Coffee Shop	Café	Pizza Place	Gym / Fitness Center	Burger Joint	Sushi Restaurant	Grocery Store	Chinese Restaurant
3	Bispebjerg	Copenhagen	2	Pizza Place	Grocery Store	Café	Convenience Store	Supermarket	Sushi Restaurant	Thai Restaurant	Gym / Fitness Center	Dance Studio	Plaza
5	Brønshøj-Husum	Copenhagen	2	Bakery	Park	Supermarket	Plaza	Grocery Store	Martial Arts Dojo	Café	Theater	Scandinavian Restaurant	Bus Stop
9	Frederiksberg	Copenhagen	2	Café	Scandinavian Restaurant	Park	Bakery	French Restaurant	Italian Restaurant	Bar	Sushi Restaurant	Theater	Coffee Shop
16	Nørrebro	Copenhagen	2	Coffee Shop	Pizza Place	Café	Bakery	Thai Restaurant	Gym / Fitness Center	Playground	Wine Bar	Park	Beer Bar
25	Valby	Copenhagen	2	Coffee Shop	Café	Pizza Place	Gym / Fitness Center	Bakery	Grocery Store	Sushi Restaurant	Supermarket	Indian Restaurant	Thai Restaurant
26	Vanløse	Copenhagen	2	Grocery Store	Convenience Store	Pizza Place	Soccer Field	Supermarket	Train Station	Gym	Scandinavian Restaurant	Café	Bakery
28	Vesterbro/Kongens Enghave	Copenhagen	2	Café	Cocktail Bar	Coffee Shop	Pizza Place	Scandinavian Restaurant	Wine Bar	Italian Restaurant	Music Venue	Park	Bakery

Group 3 with 0 Copenhagen and 1 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
10	Gobelins	Paris	3	Vietnamese Restaurant	Thai Restaurant	Asian Restaurant	Chinese Restaurant	French Restaurant	Hotel	Bakery	Supermarket	Sushi Restaurant	Cantonese Restaurant

Group 4 with 0 Copenhagen and 11 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
2	Batignolles-Monceau	Paris	4	French Restaurant	Hotel	Italian Restaurant	Bakery	Restaurant	Bistro	Plaza	Pastry Shop	Japanese Restaurant	Bar
4	Bourse	Paris	4	French Restaurant	Bistro	Wine Bar	Japanese Restaurant	Salad Place	Plaza	Cocktail Bar	Italian Restaurant	Bar	Hotel
11	Hôtel-de-Ville	Paris	4	French Restaurant	Hotel	Ice Cream Shop	Plaza	Wine Bar	Pastry Shop	Bakery	Tea Room	Garden	Seafood Restaurant
17	Observatoire	Paris	4	French Restaurant	Hotel	Italian Restaurant	Bar	Japanese Restaurant	Bakery	Sushi Restaurant	Indian Restaurant	Bistro	Pizza Place
18	Opéra	Paris	4	French Restaurant	Hotel	Cocktail Bar	Japanese Restaurant	Bistro	Italian Restaurant	Bar	Wine Bar	Bakery	Vegetarian / Vegan Restaurant
19	Palais-Bourbon	Paris	4	French Restaurant	Hotel	Plaza	Italian Restaurant	Café	History Museum	Historic Site	Garden	Cocktail Bar	Bakery
20	Panthéon	Paris	4	French Restaurant	Bar	Italian Restaurant	Café	Wine Bar	Bakery	Plaza	Pub	Hotel	Greek Restaurant
21	Passy	Paris	4	French Restaurant	Bakery	Italian Restaurant	Café	Supermarket	Garden	Park	Pizza Place	Bistro	Gym / Fitness Center
23	Reuilly	Paris	4	French Restaurant	Japanese Restaurant	Hotel	Italian Restaurant	Plaza	Supermarket	Bakery	Pizza Place	Asian Restaurant	Restaurant
27	Vaugirard	Paris	4	French Restaurant	Hotel	Italian Restaurant	Bakery	Lebanese Restaurant	Coffee Shop	Korean Restaurant	Park	Japanese Restaurant	Indian Restaurant
29	Élysée	Paris	4	Hotel	French Restaurant	Art Gallery	Italian Restaurant	Clothing Store	Bakery	Cosmetics Shop	Salad Place	Japanese Restaurant	Bar

Group 5 with 1 Copenhagen and 4 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
8	Entrepôt	Paris	5	Coffee Shop	French Restaurant	Bistro	Pizza Place	Cocktail Bar	Indian Restaurant	Italian Restaurant	Asian Restaurant	Restaurant	Seafood Restaurant
12	Indre By	Copenhagen	5	Coffee Shop	Restaurant	Scandinavian Restaurant	Cocktail Bar	Café	Wine Bar	Bar	Furniture / Home Store	Plaza	Theater
13	Louvre	Paris	5	French Restaurant	Hotel	Plaza	Café	Exhibit	Coffee Shop	Japanese Restaurant	Historic Site	Bar	Art Museum
14	Luxembourg	Paris	5	French Restaurant	Italian Restaurant	Hotel	Wine Bar	Bistro	Tea Room	Bakery	Pastry Shop	Chocolate Shop	Plaza
24	Temple	Paris	5	French Restaurant	Bistro	Hotel	Café	Art Gallery	Italian Restaurant	Coffee Shop	Burger Joint	Pizza Place	Boutique

Method 3: Cluster into Two Clusters with all Restaurant Venue Categories combined

Introduction and Clustering

When looking at the top ten venue type for the Paris neighborhoods it is quite clear that they all have French Restaurants as the first or second most frequent venue type. The Copenhagen neighborhoods, on the other hand, have more Scandinavian Restaurants on their top ten venue type list which totally makes sense.

Until now I how given the cluster algorithm data where for example French Restaurants are considered different from Scandinavian Restaurants. In this and the next experiment I will see what happens if we consider restaurants being the same whatever nationality their kitchen serve.

I already prepared a frequncy table of the venue types where I changed all the different types of restaurants to the same venue type *Restaurant* and I use this simplified venue data in the following.

In [38]:
```
kclusters = 2
df_freqClustering = df_freqSimplified.copy().drop(['Neighborhood', 'City'], 1)

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

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

Out[38]:
```
array([1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 0, 1, 1, 1], dtype=int32)
```

Result and Discussion

I create a table for each clustered group that shows the neighborhoods of the group and the neighborhoods top 10 venue types.

It turns out that most of the Paris neighborhoods are in a cluster for themself but that five of the Paris neighborhoods mix in with the eleven Copenhagen neighborhoods.

This result actually shows some evidence of the likeness between Copenhagen and Paris. You can say that some neighborhoods in Paris, like *Buttes-Chaumont*, *Louvre*, *Luxembourg*, *Temple*, and *Élysée*, have likeness to Copenhagen.

In [39]:
```
printClusters(df_topTenSimplifiedVenues.copy(), kmeans.labels_)
```

Group 0 with 0 Copenhagen and 15 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
2	Batignolles-Monceau	Paris	0	Restaurant	Hotel	Bakery	Bistro	Plaza	Pastry Shop	Cheese Shop	Park	Bar	Pizza Place
4	Bourse	Paris	0	Restaurant	Bistro	Wine Bar	Plaza	Bar	Cocktail Bar	Hotel	Salad Place	Lounge	Pizza Place
7	Buttes-Montmartre	Paris	0	Restaurant	Bar	Bistro	Pizza Place	Plaza	Café	Deli / Bodega	Gastropub	Convenience Store	Sandwich Place
8	Entrepôt	Paris	0	Restaurant	Coffee Shop	Bistro	Pizza Place	Cocktail Bar	Wine Shop	Café	Breakfast Spot	Bakery	Burger Joint
10	Gobelins	Paris	0	Restaurant	Hotel	Bakery	Bistro	Park	Bar	Supermarket	Café	Farmers Market	Convenience Store
11	Hôtel-de-Ville	Paris	0	Restaurant	Hotel	Ice Cream Shop	Wine Bar	Pastry Shop	Bakery	Plaza	Tea Room	Garden	Art Gallery
15	Ménilmontant	Paris	0	Restaurant	Bar	Plaza	Bakery	Supermarket	Bistro	Café	Hotel	Pizza Place	Music Venue
17	Observatoire	Paris	0	Restaurant	Hotel	Bar	Pizza Place	Bistro	Bakery	Wine Bar	Plaza	Coffee Shop	Art Museum
18	Opéra	Paris	0	Restaurant	Hotel	Cocktail Bar	Bistro	Bakery	Wine Bar	Bar	Pizza Place	Sandwich Place	Cupcake Shop
19	Palais-Bourbon	Paris	0	Restaurant	Hotel	Plaza	Café	History Museum	Cocktail Bar	Historic Site	Garden	Ice Cream Shop	Coffee Shop
20	Panthéon	Paris	0	Restaurant	Café	Bar	Wine Bar	Plaza	Bakery	Hotel	Pub	Coffee Shop	Garden
21	Passy	Paris	0	Restaurant	Bakery	Café	Garden	Supermarket	Pizza Place	Bistro	Gym / Fitness Center	Park	Plaza
22	Popincourt	Paris	0	Restaurant	Bar	Cocktail Bar	Bistro	Wine Bar	Beer Bar	Pizza Place	Pub	Bakery	Record Shop
23	Reuilly	Paris	0	Restaurant	Hotel	Plaza	Supermarket	Bakery	Park	Tram Station	Café	Sandwich Place	Pizza Place
27	Vaugirard	Paris	0	Restaurant	Hotel	Bakery	Park	Coffee Shop	Bistro	Gym / Fitness Center	Gym	Bar	Gastropub

Group 1 with 11 Copenhagen and 5 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
0	Amager Vest	Copenhagen	1	Restaurant	Hotel	Supermarket	Park	Pizza Place	Coffee Shop	Café	Golf Course	Grocery Store	Gym / Fitness Center
1	Amager Øst	Copenhagen	1	Restaurant	Beach	Café	Coffee Shop	Bakery	Gym / Fitness Center	Burger Joint	Pizza Place	Grocery Store	Sports Club
3	Bispebjerg	Copenhagen	1	Restaurant	Pizza Place	Café	Grocery Store	Dance Studio	Convenience Store	Gym / Fitness Center	Supermarket	Hostel	Farmers Market
5	Brønshøj-Husum	Copenhagen	1	Restaurant	Park	Supermarket	Bakery	Theater	Bus Stop	Snack Place	Bowling Alley	Café	Martial Arts Dojo
6	Buttes-Chaumont	Paris	1	Restaurant	Bar	Café	Pizza Place	Multiplex	Brasserie	Beer Bar	Pool	Canal	Hotel
9	Frederiksberg	Copenhagen	1	Restaurant	Café	Park	Bakery	Bar	Pizza Place	Exhibit	Burger Joint	Theater	Coffee Shop
12	Indre By	Copenhagen	1	Restaurant	Coffee Shop	Cocktail Bar	Café	Furniture / Home Store	Wine Bar	Bar	Plaza	Clothing Store	Steakhouse
13	Louvre	Paris	1	Restaurant	Hotel	Café	Plaza	Exhibit	Coffee Shop	Art Museum	Bar	Historic Site	Tea Room
14	Luxembourg	Paris	1	Restaurant	Hotel	Wine Bar	Tea Room	Bistro	Women's Store	Pastry Shop	Ice Cream Shop	Chocolate Shop	Bakery
16	Nørrebro	Copenhagen	1	Restaurant	Coffee Shop	Pizza Place	Café	Bakery	Playground	Gym / Fitness Center	Wine Bar	Park	Beer Bar
24	Temple	Paris	1	Restaurant	Bistro	Coffee Shop	Hotel	Art Gallery	Café	Pizza Place	Sandwich Place	Boutique	Burger Joint
25	Valby	Copenhagen	1	Restaurant	Coffee Shop	Café	Pizza Place	Bakery	Gym / Fitness Center	Grocery Store	Supermarket	Burger Joint	Ice Cream Shop
26	Vanløse	Copenhagen	1	Restaurant	Grocery Store	Convenience Store	Pizza Place	Supermarket	Soccer Field	Train Station	Gym / Fitness Center	Café	Lake
28	Vesterbro/Kongens Enghave	Copenhagen	1	Restaurant	Café	Cocktail Bar	Coffee Shop	Pizza Place	Wine Bar	Music Venue	Bakery	Park	Bistro
29	Élysée	Paris	1	Restaurant	Hotel	Art Gallery	Clothing Store	Bakery	Theater	Garden	Cosmetics Shop	Bar	Salad Place
30	Østerbro	Copenhagen	1	Restaurant	Harbor / Marina	Coffee Shop	Bakery	Playground	Pizza Place	Soccer Stadium	Ice Cream Shop	Gym	Theater

Method 4: Cluster into 6 Clusters with all Restaurant Venue Categories combined

Introduction and Clustering

In [40]:
```
kclusters = 6
df_freqClustering = df_freqSimplified.copy().drop(['Neighborhood', 'City'], 1)

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

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

Out[40]:
```
array([4, 4, 5, 3, 0, 4, 1, 0, 0, 1, 5, 0, 1, 2, 2, 0, 1, 5, 0, 0, 0, 0,
       0, 0, 2, 4, 3, 5, 1, 2, 4], dtype=int32)
```

Result and Discussion

I create a table for each clustered group that shows the neighborhoods of the group and the neighborhoods top 10 venue types.

The tables show that again most of the groups have no mixing between the two cities at all and only one group that has minimal mixing. This time it is *Buttes-Chaumont* that mixes in with some of the Copenhagen neighborhoods including *Indre By* (*Inner City*).

In [41]:
```
printClusters(df_topTenSimplifiedVenues.copy(), kmeans.labels_)
```

Group 0 with 0 Copenhagen and 11 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
4	Bourse	Paris	0	Restaurant	Bistro	Wine Bar	Plaza	Bar	Cocktail Bar	Hotel	Salad Place	Lounge	Pizza Place
7	Buttes-Montmartre	Paris	0	Restaurant	Bar	Bistro	Pizza Place	Plaza	Café	Deli / Bodega	Gastropub	Convenience Store	Sandwich Place
8	Entrepôt	Paris	0	Restaurant	Coffee Shop	Bistro	Pizza Place	Cocktail Bar	Wine Shop	Café	Breakfast Spot	Bakery	Burger Joint
11	Hôtel-de-Ville	Paris	0	Restaurant	Hotel	Ice Cream Shop	Wine Bar	Pastry Shop	Bakery	Plaza	Tea Room	Garden	Art Gallery
15	Ménilmontant	Paris	0	Restaurant	Bar	Plaza	Bakery	Supermarket	Bistro	Café	Hotel	Pizza Place	Music Venue
18	Opéra	Paris	0	Restaurant	Hotel	Cocktail Bar	Bistro	Bakery	Wine Bar	Bar	Pizza Place	Sandwich Place	Cupcake Shop
19	Palais-Bourbon	Paris	0	Restaurant	Hotel	Plaza	Café	History Museum	Cocktail Bar	Historic Site	Garden	Ice Cream Shop	Coffee Shop
20	Panthéon	Paris	0	Restaurant	Café	Bar	Wine Bar	Plaza	Bakery	Hotel	Pub	Coffee Shop	Garden
21	Passy	Paris	0	Restaurant	Bakery	Café	Garden	Supermarket	Pizza Place	Bistro	Gym / Fitness Center	Park	Plaza
22	Popincourt	Paris	0	Restaurant	Bar	Cocktail Bar	Bistro	Wine Bar	Beer Bar	Pizza Place	Pub	Bakery	Record Shop
23	Reuilly	Paris	0	Restaurant	Hotel	Plaza	Supermarket	Bakery	Park	Tram Station	Café	Sandwich Place	Pizza Place

Group 1 with 4 Copenhagen and 1 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
6	Buttes-Chaumont	Paris	1	Restaurant	Bar	Café	Pizza Place	Multiplex	Brasserie	Beer Bar	Pool	Canal	Hotel
9	Frederiksberg	Copenhagen	1	Restaurant	Café	Park	Bakery	Bar	Pizza Place	Exhibit	Burger Joint	Theater	Coffee Shop
12	Indre By	Copenhagen	1	Restaurant	Coffee Shop	Cocktail Bar	Café	Furniture / Home Store	Wine Bar	Bar	Plaza	Clothing Store	Steakhouse
16	Nørrebro	Copenhagen	1	Restaurant	Coffee Shop	Pizza Place	Café	Bakery	Playground	Gym / Fitness Center	Wine Bar	Park	Beer Bar
28	Vesterbro/Kongens Enghave	Copenhagen	1	Restaurant	Café	Cocktail Bar	Coffee Shop	Pizza Place	Wine Bar	Music Venue	Bakery	Park	Bistro

Group 2 with 0 Copenhagen and 4 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
13	Louvre	Paris	2	Restaurant	Hotel	Café	Plaza	Exhibit	Coffee Shop	Art Museum	Bar	Historic Site	Tea Room
14	Luxembourg	Paris	2	Restaurant	Hotel	Wine Bar	Tea Room	Bistro	Women's Store	Pastry Shop	Ice Cream Shop	Chocolate Shop	Bakery
24	Temple	Paris	2	Restaurant	Bistro	Coffee Shop	Hotel	Art Gallery	Café	Pizza Place	Sandwich Place	Boutique	Burger Joint
29	Élysée	Paris	2	Restaurant	Hotel	Art Gallery	Clothing Store	Bakery	Theater	Garden	Cosmetics Shop	Bar	Salad Place

Group 3 with 2 Copenhagen and 0 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
3	Bispebjerg	Copenhagen	3	Restaurant	Pizza Place	Café	Grocery Store	Dance Studio	Convenience Store	Gym / Fitness Center	Supermarket	Hostel	Farmers Market
26	Vanløse	Copenhagen	3	Restaurant	Grocery Store	Convenience Store	Pizza Place	Supermarket	Soccer Field	Train Station	Gym / Fitness Center	Café	Lake

Group 4 with 5 Copenhagen and 0 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
0	Amager Vest	Copenhagen	4	Restaurant	Hotel	Supermarket	Park	Pizza Place	Coffee Shop	Café	Golf Course	Grocery Store	Gym / Fitness Center
1	Amager Øst	Copenhagen	4	Restaurant	Beach	Café	Coffee Shop	Bakery	Gym / Fitness Center	Burger Joint	Pizza Place	Grocery Store	Sports Club
5	Brønshøj-Husum	Copenhagen	4	Restaurant	Park	Supermarket	Bakery	Theater	Bus Stop	Snack Place	Bowling Alley	Café	Martial Arts Dojo
25	Valby	Copenhagen	4	Restaurant	Coffee Shop	Café	Pizza Place	Bakery	Gym / Fitness Center	Grocery Store	Supermarket	Burger Joint	Ice Cream Shop
30	Østerbro	Copenhagen	4	Restaurant	Harbor / Marina	Coffee Shop	Bakery	Playground	Pizza Place	Soccer Stadium	Ice Cream Shop	Gym	Theater

Group 5 with 0 Copenhagen and 4 Paris neighborhoods:

	Neighborhood	City	Cluster	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	9th Most Common Venue	10th Most Common Venue
2	Batignolles-Monceau	Paris	5	Restaurant	Hotel	Bakery	Bistro	Plaza	Pastry Shop	Cheese Shop	Park	Bar	Pizza Place
10	Gobelins	Paris	5	Restaurant	Hotel	Bakery	Bistro	Park	Bar	Supermarket	Café	Farmers Market	Convenience Store
17	Observatoire	Paris	5	Restaurant	Hotel	Bar	Pizza Place	Bistro	Bakery	Wine Bar	Plaza	Coffee Shop	Art Museum
27	Vaugirard	Paris	5	Restaurant	Hotel	Bakery	Park	Coffee Shop	Bistro	Gym / Fitness Center	Gym	Bar	Gastropub

8. Conclusion

The claim that Copenhagen is like Paris is somewhat supported by this investigation using tools from Data Science and Machine Learning and using Foursquare venue type data to define likeness. The evidence is not overwhelming in that some of the experiments only showed minimal mixing between the neighborhoods of Copenhagen and Paris.

The mixing that does occur, however, seems to be consistent between the experiments and they show that several Copenhagen neighborhoods, and especially *Indre By* (*Inner City*), have many likenesses with the Paris neighborhoods *Entrepôt*, *Louvre*, *Luxembourg*, *Temple*, and *Buttes-Chaumont*.