Capstone Project - Barcelona's Battle of the Neighborhoods

Applied Data Science Capstone by IBM/Coursera

Table of contents

Introduction: Business Problem

In this project we will try to find an optimal location for a restaurant. Specifically, this report will be targeted to stakeholders interested in opening an **Italian restaurant** in **Barcelona**, Spain.

Since there are lots of restaurants in Barcelona we will try to detect **locations that are not already crowded with restaurants**. We are also particularly interested in **areas with no Polish restaurants in vicinity**. We would also prefer locations **as close to city center as possible**, assuming that first two conditions are met.

We will use our data science powers to generate a few most promising neighborhoods based on this criteria. Advantages of each area will then be clearly expressed so that best possible final location can be chosen by stakeholders.

Data

Based on definition of our problem, factors that will influence our decission are:

- number of existing restaurants in the neighborhood (any type of restaurant)
- number of and distance to other popular placer in the neighborhood, if any
- distance of neighborhood from city center

We decided to use regularly spaced grid of locations, centered around city center, to define our neighborhoods.

Following data sources will be needed to extract/generate the required information:

- District centres of candidate areas will be obtained using **Barcelona Open Data source** from Local authorities.
- number of restaurants and their type and location in every neighborhood will be obtained using **Foursquare API**
- coordinate of Barcelona center will be obtained using **Barcelona Open Data source** of city centre location (Plaza Catalunya) and the main Districts.

Neighborhood Candidates

Let's create latitude & longitude coordinates for centroids of our candidate neighborhoods using the District coordinates provided by the Barcelona Open Data source.

Let's first find the latitude & longitude of Barcelona city center, using specific, well known address and Foursquare API.

Import necessary Libraries

In [1]:

```
conda install -c anaconda wget
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/rafadiazrios/opt/anaconda3

  added / updated specs:
    - wget


The following packages will be SUPERSEDED by a higher-priority chann
el:

  ca-certificates    conda-forge::ca-certificates-2020.6.2~ --> anac
onda::ca-certificates-2020.1.1-0
  certifi            conda-forge::certifi-2020.6.20-py37hc~ --> anac
onda::certifi-2020.6.20-py37_0
  conda              conda-forge::conda-4.8.3-py37hc8dfbb8~ --> anac
onda::conda-4.8.3-py37_0
  openssl            conda-forge::openssl-1.1.1g-h0b31af3_0 --> anac
onda::openssl-1.1.1g-h1de35cc_0


Preparing transaction: done
Verifying transaction: done
Executing transaction: done

Note: you may need to restart the kernel to use updated packages.
```

In [2]:

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

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

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't c
ompleted the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and lon
gitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dat
aframe

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

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

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you ha
ven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/rafadiazrios/opt/anaconda3

  added / updated specs:
    - geopy


The following packages will be UPDATED:

  ca-certificates      anaconda::ca-certificates-2020.1.1-0 --> cond
a-forge::ca-certificates-2020.6.20-hecda079_0
  conda                        anaconda::conda-4.8.3-py37_0 --> cond
a-forge::conda-4.8.3-py37hc8dfbb8_1

The following packages will be SUPERSEDED by a higher-priority chann
el:

  certifi                anaconda::certifi-2020.6.20-py37_0 --> cond
a-forge::certifi-2020.6.20-py37hc8dfbb8_0
  openssl               anaconda::openssl-1.1.1g-h1de35cc_0 --> cond
a-forge::openssl-1.1.1g-h0b31af3_0


Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Libraries imported.
```

In [3]:

```
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation

!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitu
de and longitude values

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library

print('Folium installed')
print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Folium installed
Libraries imported.
```

In [7]:

```
import matplotlib.pyplot as plt # plotting library
# backend for rendering plots within the browser
%matplotlib inline

from sklearn.cluster import KMeans
from sklearn.datasets.samples_generator import make_blobs

print('Libraries imported.')
```

```
Libraries imported.
```

Import JSON file for Barcelona

Import from Area Metropolitana de Barcelona website:

https://geoportalcartografia.amb.cat/AppGeoportalCartografia2/DadesAplicacio/Geoserveis/ca/default.html (https://geoportalcartografia.amb.cat/AppGeoportalCartografia2/DadesAplicacio/Geoserveis/ca/default.html)

In [8]:

```
#bcn_data = "/Users/rafadiazrios/Coursera Jupyter DS & PY learning/Course 9 - Ca
pstone project/districtes_i_barris_170705.json"
#print('Data downloaded!')

bcn_geodata = "/Users/rafadiazrios/Coursera Jupyter DS & PY learning/Course 9 -
 Capstone project/bcn_UNITATS_ADM_PUNTS.json"
print('Data2 downloaded!')
```

```
Data2 downloaded!
```

In [9]:

```
with open('bcn_UNITATS_ADM_PUNTS.json') as json_data:
    barna_data = json.load(json_data)
```

```
    'WEB2': ' ',
    'WEB3': ' ',
    'LITERAL': '068',
    'ANGLE_TXT': 0}},
  {'type': 'Feature',
   'id': 999,
   'geometry': {'type': 'Point',
    'coordinates': [431169.87040000036, 4588730.9081999995]},
   'properties': {'FID': 999,
    'ID_ANNEX': '01',
    'ANNEXDESCR': 'Grup - I',
    'ID_TEMA': '0104',
    'TEMA_DESCR': 'Unitats Administratives',
    'ID_CONJ': '010415',
    'CONJ_DESCR': 'Secció censal',
    'ID_SUBCONJ': '01041501',
    'SCONJDESCR': 'Secció censal',
    'ID_ELEMENT': '0104150102',
    'ELEM_DESCR': 'Codi secció censal',
    'NIVELL': 'ADM_05_RT',
    'NDESCR_CA': 'Codi secció censal (rètol)',
    'NDESCR_ES': 'Código sección censal (rótulo)',
    'NDESCR_EN': 'Census area code (label)',
    'DISTRICTE': '08',
    'BARRI': '50',
    'TERME': '080193',
    'AEB': '170',
    'SEC_CENS': '068',
    'GRANBARRI': '31',
    'ZUA': '53',
    'CODI_UA': ' ',
    'TIPUS_UA': 'SEC_CENS',
    'NOM': ' ',
    'WEB1': ' ',
    'WEB2': ' ',
    'WEB3': ' ',
    'LITERAL': '068',
    'ANGLE_TXT': 0}},
  ...]}
```


In [11]:

```
neighborhoods_data = barna_data['features']
```

Let's take a look at the first item in this list.

In [12]:

```
neighborhoods_data[0]
```

Out[12]:

```
{'type': 'Feature',
 'id': 0,
 'geometry': {'type': 'Point',
  'coordinates': [424861.74899999984, 4581559.359999999]},
 'properties': {'FID': 0,
  'ID_ANNEX': '01',
  'ANNEXDESCR': 'Grup - I',
  'ID_TEMA': '0104',
  'TEMA_DESCR': 'Unitats Administratives',
  'ID_CONJ': '010411',
  'CONJ_DESCR': 'Terme Municipal',
  'ID_SUBCONJ': '01041101',
  'SCONJDESCR': 'Terme Municipal',
  'ID_ELEMENT': '0104110104',
  'ELEM_DESCR': 'Noms municipis veïns',
  'NIVELL': 'ADM_01_aux_RT',
  'NDESCR_CA': 'Noms municipis veïns',
  'NDESCR_ES': 'Nombres municipios vecinos',
  'NDESCR_EN': 'Names neighboring municipalities',
  'DISTRICTE': '-',
  'BARRI': '-',
  'TERME': '080771',
  'AEB': '-',
  'SEC_CENS': '-',
  'GRANBARRI': '-',
  'ZUA': '-',
  'CODI_UA': '080771',
  'TIPUS_UA': 'TERME',
  'NOM': 'Esplugues de Llobregat',
  'WEB1': ' ',
  'WEB2': ' ',
  'WEB3': ' ',
  'LITERAL': '080771',
  'ANGLE_TXT': 285}}
```

In [13]:

```
# define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
```

Take a look at the empty dataframe to confirm that the columns are as intended.

In [14]:

```
neighborhoods
```

Out[14]:

Borough	Neighborhood	Latitude	Longitude

Then let's loop through the data and fill the dataframe one row at a time.

In [15]:

```
for data in neighborhoods_data:
    borough = neighborhood_name = data['properties']['TIPUS_UA']
    neighborhood_name = data['properties']['NOM']

    neighborhood_latlon = data['geometry']['coordinates']
    neighborhood_lat = neighborhood_latlon[1]
    neighborhood_lon = neighborhood_latlon[0]

    neighborhoods = neighborhoods.append({'Borough': borough,
                                          'Neighborhood': neighborhood_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore
_index=True)
```

In [234]:

```
# @hidden_cell
neighborhoods
```

	Borough	Neighborhood	Latitude	Longitude
1519	ZUA		4.588638e+06	429973.0989
1520	ZUA		4.588771e+06	431134.3408
1521	ZUA		4.588290e+06	431246.7223
1522	ZUA		4.588177e+06	431651.3157
1523	ZUA		4.589093e+06	431912.5638
1524	ZUA		4.589611e+06	430842.8058
1525	ZUA		4.589061e+06	432777.9417
1526	ZUA		4.587574e+06	433317.1531
1527	ZUA		4.587447e+06	432388.8514
1528	ZUA		4.587404e+06	432010.2050
1529	ZUA		4.586093e+06	432295.0817
1530	ZUA		4.586327e+06	431531.6775
1531	ZUA		4.585653e+06	432004.9342
1532	ZUA		4.584788e+06	431679.6241
1533	ZUA		4.584499e+06	432251.9554
1534	ZUA		4.583064e+06	432346.1147
1535	ZUA		4.582516e+06	433040.7758
1536	ZUA		4.583407e+06	433327.0480
1537	ZUA		4.584046e+06	434222.7536
1538	ZUA		4.584701e+06	433318.3684
1539	ZUA		4.585289e+06	432981.5495
1540	ZUA		4.586127e+06	433407.2298
1541	ZUA		4.584877e+06	434709.5523

Let's slice the dataframe to select only the districts of Barcelona city

In [148]:

```
bcncity_data = neighborhoods[neighborhoods['Borough'] == 'BARRI'].reset_index(drop=True)
bcncity_data.head(10)
```

Out[148]:

	Borough	Neighborhood	Latitude	Longitude
0	BARRI	el Raval	4.581121e+06	430624.9313
1	BARRI	el Barri Gòtic	4.581289e+06	431291.4440
2	BARRI	la Barceloneta	4.581448e+06	432355.6530
3	BARRI	Sant Pere, Santa Caterina i la Ribera	4.581984e+06	431707.9381
4	BARRI	el Fort Pienc	4.583261e+06	431580.2748
5	BARRI	la Sagrada Família	4.584175e+06	431275.9502
6	BARRI	la Dreta de l'Eixample	4.582931e+06	430582.0822
7	BARRI	l'Antiga Esquerra de l'Eixample	4.582208e+06	429278.2125
8	BARRI	la Nova Esquerra de l'Eixample	4.581700e+06	428919.9315
9	BARRI	Sant Antoni	4.581114e+06	429725.6300

In [149]:

```
bcncity_data.shape
```

Out[149]:

```
(75, 4)
```

Define Foursquare Credentials and Version

Make sure that you have created a Foursquare developer account and have your credentials handy

In [237]:

```
# @hidden_cell
client_id = [redacted]# your Foursquare ID
client_secret = [redacted]# your Foursquare Secret
version = '20190604'
limit = 2000
radius = 5000
print('Your credentials:')
print('CLIENT_ID: ' + client_id)
print('CLIENT_SECRET:' + client_secret)
```

```
Your credentials:
CLIENT_ID: [redacted]
CLIENT_SECRET:[redacted]
```

In [21]:

```
# @hidden_cell
## Google Api Key
[redacted]
```

Neighborhood Candidates

Let's create latitude & longitude coordinates for centroids of our candidate neighborhoods.

Let's first find the latitude & longitude of Berlin city center, using specific, well known address and Foursquare API

Let's get the coordinates of the place we would like to explore

In [23]:

```
#address = 'Plaza Catalunya, Barcelona, Spain'
address = 'Plaza de Catalunya, Barcelona, Spain'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(location)
print(latitude, longitude)
```

```
Plaça de Catalunya, la Dreta de l'Eixample, Eixample, Ciutat Vella,
Barcelona, Barcelonès, Barcelona, Catalunya, 08001, España
41.3861586 2.169774
```

In [ ]:

```

```

In [24]:

```
import requests

def get_coordinates(api_key, address, verbose=False):
    try:
        url = 'https://maps.googleapis.com/maps/api/geocode/json?key={}&address=
{}'.format(api_key, address)
        response = requests.get(url).json()
        if verbose:
            print('Google Maps API JSON result =>', response)
        results = response['results']
        geographical_data = results[0]['geometry']['location'] # get geographica
l coordinates
        lat = geographical_data['lat']
        lon = geographical_data['lng']
        return [lat, lon]
    except:
        return [None, None]

barcelona_center = get_coordinates(api_key, address)
print('Coordinate of {}: {}'.format(address, barcelona_center))
```

```
Coordinate of Plaza de Catalunya, Barcelona, Spain: [41.3870154, 2.1
700471]
```

Now let's create a grid of area candidates, equaly spaced, centered around city center and within ~6km from Plaza de Catalunya. Our neighborhoods will be defined as circular areas with a radius of 300 meters, so our neighborhood centers will be 600 meters apart.

To accurately calculate distances we need to create our grid of locations in Cartesian 2D coordinate system which allows us to calculate distances in meters (not in latitude/longitude degrees). Then we'll project those coordinates back to latitude/longitude degrees to be shown on Folium map. So let's create functions to convert between WGS84 spherical coordinate system (latitude/longitude degrees) and UTM Cartesian coordinate system (X/Y coordinates in meters).

In [25]:

```
!pip install shapely
```

```
Requirement already satisfied: shapely in /Users/rafadiazrios/opt/an
aconda3/lib/python3.7/site-packages (1.7.0)
```

In [26]:

```
!pip install pyproj
```

```
Requirement already satisfied: pyproj in /Users/rafadiazrios/opt/ana
conda3/lib/python3.7/site-packages (2.6.1.post1)
```

In [27]:

```
#!pip install shapely
import shapely.geometry

#!pip install pyproj
import pyproj

import math

def lonlat_to_xy(lon, lat):
    proj_latlon = pyproj.Proj(proj='latlong',datum='WGS84')
    proj_xy = pyproj.Proj(proj="utm", zone=31, datum='WGS84')
    xy = pyproj.transform(proj_latlon, proj_xy, lon, lat)
    return xy[0], xy[1]

def xy_to_lonlat(x, y):
    proj_latlon = pyproj.Proj(proj='latlong',datum='WGS84')
    proj_xy = pyproj.Proj(proj="utm", zone=31, datum='WGS84')
    lonlat = pyproj.transform(proj_xy, proj_latlon, x, y)
    return lonlat[0], lonlat[1]

def calc_xy_distance(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    return math.sqrt(dx*dx + dy*dy)

print('Coordinate transformation check')
print('-------------------------------')
print('Barcelona center longitude={}, latitude={}'.format(longitude, latitude))
x, y = lonlat_to_xy(longitude, latitude)
print('Barcelona center UTM X={}, Y={}'.format(x, y))
lo, la = xy_to_lonlat(x, y)
print('Barcelona center longitude={}, latitude={}'.format(lo, la))
```

```
Coordinate transformation check
-------------------------------
Barcelona center longitude=2.169774, latitude=41.3861586
Barcelona center UTM X=430585.515902288, Y=4581958.275650984
Barcelona center longitude=2.1697740000000003, latitude=41.3861586

/Users/rafadiazrios/opt/anaconda3/lib/python3.7/site-packages/ipyker
nel_launcher.py:12: DeprecationWarning: This function is deprecated.
See: https://pyproj4.github.io/pyproj/stable/gotchas.html#upgrading-
to-pyproj-2-from-pyproj-1
  if sys.path[0] == '':
/Users/rafadiazrios/opt/anaconda3/lib/python3.7/site-packages/ipyker
nel_launcher.py:18: DeprecationWarning: This function is deprecated.
See: https://pyproj4.github.io/pyproj/stable/gotchas.html#upgrading-
to-pyproj-2-from-pyproj-1
```

In [28]:

```
#address = 'Plaza Catalunya, Barcelona, Spain'
barcelona_center = (latitude,longitude)
print('Coordinate of {}: {}'.format(address, barcelona_center))
```

```
Coordinate of Plaza de Catalunya, Barcelona, Spain: (41.3861586, 2.1
69774)
```

Let's create a **hexagonal grid of cells**: we offset every other row, and adjust vertical row spacing so that **every cell center is equally distant from all it's neighbors.**

In [29]:

```
barcelona_center_x, barcelona_center_y = lonlat_to_xy(barcelona_center[1], barce
lona_center[0]) # City center in Cartesian coordinates

k = math.sqrt(3) / 2 # Vertical offset for hexagonal grid cells
x_min = barcelona_center_x - 6000
x_step = 600
y_min = barcelona_center_y - 6000 - (int(21/k)*k*600 - 12000)/2
y_step = 600 * k

latitudes = []
longitudes = []
distances_from_center = []
xs = []
ys = []
for i in range(0, int(21/k)):
    y = y_min + i * y_step
    x_offset = 300 if i%2==0 else 0
    for j in range(0, 21):
        x = x_min + j * x_step + x_offset
        distance_from_center = calc_xy_distance(barcelona_center_x, barcelona_ce
nter_y, x, y)
        if (distance_from_center <= 6001):
            lon, lat = xy_to_lonlat(x, y)
            latitudes.append(lat)
            longitudes.append(lon)
            distances_from_center.append(distance_from_center)
            xs.append(x)
            ys.append(y)

print(len(latitudes), 'candidate neighborhood centers generated.')
```

```
/Users/rafadiazrios/opt/anaconda3/lib/python3.7/site-packages/ipyker
nel_launcher.py:18: DeprecationWarning: This function is deprecated.
See: https://pyproj4.github.io/pyproj/stable/gotchas.html#upgrading-
to-pyproj-2-from-pyproj-1
/Users/rafadiazrios/opt/anaconda3/lib/python3.7/site-packages/ipyker
nel_launcher.py:18: DeprecationWarning: This function is deprecated.
See: https://pyproj4.github.io/pyproj/stable/gotchas.html#upgrading-
to-pyproj-2-from-pyproj-1
```

In [30]:

```
print(len(latitudes), 'candidate neighborhood centers generated.')
```

```
364 candidate neighborhood centers generated.
```

Let's visualize the data we have so far: city center location and candidate neighborhood centers:

In [33]:

```
map_barcelona = folium.Map(location=barcelona_center, zoom_start=13)
folium.Marker(barcelona_center, popup='Plaza Catalunya').add_to(map_barcelona)
for lat, lon in zip(latitudes, longitudes):
    #folium.CircleMarker([lat, lon], radius=2, color='blue', fill=True, fill_col
or='blue', fill_opacity=1).add_to(map_berlin)
    folium.Circle([lat, lon], radius=300, color='blue', fill=False).add_to(map_b
arcelona)
    #folium.Marker([lat, lon]).add_to(map_berlin)
map_barcelona
```

Out[33]:

Let's install folium

Let's visualize our well known address on the map

In [35]:

```
print(latitude, longitude)
venues_map = folium.Map(location=[latitude, longitude], zoom_start=16) # generat
e map centred Bcn Center


# add Center as a red circle mark
folium.features.CircleMarker(
    [latitude, longitude],
    radius=6,
    popup='Center',
    fill=True,
    color='red',
    fill_color='red',
    fill_opacity=0.6
    ).add_to(venues_map)

venues_map
```

```
41.3861586 2.169774
```

Out[35]:

OK, we now have the coordinates of centers of districts/areas to be evaluated from Plaza Catalunya as per our locations dataframe.

In [36]:

```
bcncity_data
#bcncity_data.shape
```

Out[36]:

	Borough	Neighborhood	Latitude	Longitude
0	DISTRICTE	Ciutat Vella	4.581700e+06	431143.2220
1	DISTRICTE	Eixample	4.582731e+06	430176.9065
2	DISTRICTE	Sants-Montjuïc	4.578407e+06	431122.7401
3	DISTRICTE	Sants-Montjuïc	4.579892e+06	429266.5550
4	DISTRICTE	Les Corts	4.582136e+06	425829.0568
5	DISTRICTE	Sarrià-Sant Gervasi	4.585476e+06	425951.2075
6	DISTRICTE	Sarrià-Sant Gervasi	4.586572e+06	421571.1313
7	DISTRICTE	Gràcia	4.584637e+06	429158.9280
8	DISTRICTE	Horta-Guinardó	4.587254e+06	429023.3310
9	DISTRICTE	Nou Barris	4.588654e+06	431128.7486
10	DISTRICTE	Sant Andreu	4.587273e+06	432622.8260
11	DISTRICTE	Sant Martí	4.584787e+06	433314.5396

In [37]:

```
def get_address(api_key, latitude, longitude, verbose=False):
    try:
        url = 'https://maps.googleapis.com/maps/api/geocode/json?key={}&latlng=
{},{}'.format(api_key, latitude, longitude)
        response = requests.get(url).json()
        if verbose:
            print('Google Maps API JSON result =>', response)
        results = response['results']
        address = results[0]['formatted_address']
        return address
    except:
        return None

addr = get_address(api_key, barcelona_center[0], barcelona_center[1])
print('Reverse geocoding check')
print('-----------------------')
print('Address of [{}, {}] is: {}'.format(barcelona_center[0], barcelona_center[
1], addr))
```

```
Reverse geocoding check
-----------------------
Address of [41.3861586, 2.169774] is: Pl. de Catalunya - Bergara, 08
002 Barcelona, Spain
```

In [47]:

```
print('Obtaining location addresses: ', end='')
addresses = []
for lat, lon in zip(latitudes, longitudes):
    address = get_address(api_key, lat, lon)
    if address is None:
        address = 'NO ADDRESS'
    address = address.replace(', Spain', '') # We don't need country part of add
ress
    addresses.append(address)
    print(' .', end='')
print(' done.')
```

```
Obtaining location addresses:  . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . done.
```

In [48]:

```
addresses[150:170]
```

Out[48]:

```
['Spain',
 'Spain',
 'Spain',
 'Avinguda Diagonal, 695, 08028 Barcelona',
 'Av. Dr. Marañón, 19, 08028 Barcelona',
 'Av. de Joan XXIII, 1, 08028 Barcelona',
 'Travessera de les Corts, 142, 08028 Barcelona',
 'Av. de Madrid, 208, 08014 Barcelona',
 'Carrer de Numància, 7, 08029 Barcelona',
 'Carrer de València, 62, 08015 Barcelona',
 'Carrer de la Diputació, 108, 08015 Barcelona',
 'Carrer de Floridablanca, 144, 08011 Barcelona',
 'Carrer Doctor Fleming, 9999, 08001 Barcelona',
 'Baixada de Sant Miquel, 8, 08002 Barcelona',
 'Carrer Pas Sota Murllas, 2, 08003 Barcelona',
 'Passeig de Salvat Papasseit, 18, 08003 Barcelona',
 'Escullera de Poblenou, 1, 08005 Barcelona',
 'Escullera de Poblenou, 167, 08005 Barcelona',
 'Escullera de Poblenou, 6, 08005 Barcelona',
 'Escullera de Poblenou, 2, 08005 Barcelona']
```

In [50]:

```
#addresses
```

Looking good. Let's now place all this into a Pandas dataframe.

In [51]:

```
import pandas as pd

df_locations = pd.DataFrame({'Address': addresses,
                             'Latitude': latitudes,
                             'Longitude': longitudes,
                             'X': xs,
                             'Y': ys,
                             'Distance from center': distances_from_center})

df_locations.head(10)
```

Out[51]:

	Address	Latitude	Longitude	X	Y	Distance from center
0	Dàrsena Sud, 08040 Barcelona	41.334521	2.148919	428785.515902	4.576243e+06	5992.495307
1	Ml Inflamables, 4, 08040 Barcelona	41.334574	2.156088	429385.515902	4.576243e+06	5840.376700
2	Carrer del Port de Ningbó, Barcelona	41.334626	2.163258	429985.515902	4.576243e+06	5747.173218
3	Spain	41.334678	2.170428	430585.515902	4.576243e+06	5715.767665
4	Spain	41.334730	2.177598	431185.515902	4.576243e+06	5747.173218
5	Spain	41.334781	2.184768	431785.515902	4.576243e+06	5840.376700
6	Spain	41.334831	2.191938	432385.515902	4.576243e+06	5992.495307
7	Unnamed Road, 08040 Barcelona	41.339121	2.138102	427885.515902	4.576762e+06	5855.766389
8	Via Circulació del Nord, 8, 08040 Barcelona	41.339175	2.145272	428485.515902	4.576762e+06	5604.462508
9	Ml Inflamables, 1, 08040 Barcelona	41.339228	2.152443	429085.515902	4.576762e+06	5408.326913

In [52]:

```
#...and let's now save/persist this data into local file.
df_locations.to_pickle('./locations.pkl')
```

Foursquare

Now that we have our location candidates, let's use Foursquare API to get info on restaurants in each neighborhood.

We're interested in venues in 'food' category, but only those that are proper restaurants - coffe shops, pizza places, bakeries etc. are not direct competitors so we don't care about those. So we will include in out list only venues that have 'restaurant' in category name, and we'll make sure to detect and include all the subcategories of specific 'Italian restaurant' category, as we need info on Italian restaurants in the neighborhood.

In [53]:

```
# Category IDs corresponding to Italian restaurants were taken from Foursquare w
eb site (https://developer.foursquare.com/docs/resources/categories):

food_category = '4d4b7105d754a06374d81259' # 'Root' category for all food-relate
d venues
# food_category = '52e81612bcbc57f1066b7a04' # Root category for Polish restaura
nt

italian_restaurant_categories = ['4bf58dd8d48988d110941735','55a5a1ebe4b01390908
7cbb6','55a5a1ebe4b013909087cb7c',
                                 '55a5a1ebe4b013909087cba7','55a5a1ebe4b01390908
7cba1','55a5a1ebe4b013909087cba4',
                                 '55a5a1ebe4b013909087cb95','55a5a1ebe4b01390908
7cb89','55a5a1ebe4b013909087cb9b',
                                 '55a5a1ebe4b013909087cb98','55a5a1ebe4b01390908
7cbbf','55a5a1ebe4b013909087cb79',
                                 '55a5a1ebe4b013909087cbb0','55a5a1ebe4b01390908
7cbb3','55a5a1ebe4b013909087cb74',
                                 '55a5a1ebe4b013909087cbaa','55a5a1ebe4b01390908
7cb83','55a5a1ebe4b013909087cb8c',
                                 '55a5a1ebe4b013909087cb92','55a5a1ebe4b01390908
7cb8f','55a5a1ebe4b013909087cb86',
                                 '55a5a1ebe4b013909087cbb9','55a5a1ebe4b01390908
7cb7f','55a5a1ebe4b013909087cbbc',
                                 '55a5a1ebe4b013909087cb9e','55a5a1ebe4b01390908
7cbc2','55a5a1ebe4b013909087cbad']

def is_restaurant(categories, specific_filter=None):
    restaurant_words = ['restaurant', 'diner', 'bar', 'pub']
    restaurant = False
    specific = False
    for c in categories:
        category_name = c[0].lower()
        category_id = c[1]
        for r in restaurant_words:
            if r in category_name:
                restaurant = True
        if 'fast food' in category_name:
            restaurant = False
        if not(specific_filter is None) and (category_id in specific_filter):
            specific = True
            restaurant = True
    return restaurant, specific

def get_categories(categories):
    return [(cat['name'], cat['id']) for cat in categories]

def format_address(location):
    address = ', '.join(location['formattedAddress'])
    address = address.replace(', Barcelona', '')
    address = address.replace(', Germany', '')
    return address

def get_venues_near_location(lat, lon, category, client_id, client_secret, radiu
s=500, limit=100):
    version = '20180724'
    url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secr
et={}&v={}&ll={},{}&categoryId={}&radius={}&limit={}'.format(
        client_id, client_secret, version, lat, lon, category, radius, limit)
```

```
    try:
        results = requests.get(url).json()['response']['groups'][0]['items']
        venues = [(item['venue']['id'],
                   item['venue']['name'],
                   get_categories(item['venue']['categories']),
                   (item['venue']['location']['lat'], item['venue']['location'][
'lng']),
                   format_address(item['venue']['location']),
                   item['venue']['location']['distance']) for item in results]
    except:
        venues = []
    return venues
```

In [54]:

```
# Let's now go over our neighborhood locations and get nearby restaurants; we'll
also maintain a dictionary of all found restaurants and all found italian restau
rants

import pickle

def get_restaurants(lats, lons):
    restaurants = {}
    italian_restaurants = {}
    location_restaurants = []

    print('Obtaining venues around candidate locations:', end='')
    for lat, lon in zip(lats, lons):
        # Using radius=350 to meke sure we have overlaps/full coverage so we do
n't miss any restaurant (we're using dictionaries to remove any duplicates resul
ting from area overlaps)
        venues = get_venues_near_location(lat, lon, food_category, client_id, cl
ient_secret, radius=350, limit=100)
        area_restaurants = []
        for venue in venues:
            venue_id = venue[0]
            venue_name = venue[1]
            venue_categories = venue[2]
            venue_latlon = venue[3]
            venue_address = venue[4]
            venue_distance = venue[5]
            is_res, is_italian = is_restaurant(venue_categories, specific_filter
=italian_restaurant_categories)
            if is_res:
                x, y = lonlat_to_xy(venue_latlon[1], venue_latlon[0])
                restaurant = (venue_id, venue_name, venue_latlon[0], venue_latlo
n[1], venue_address, venue_distance, is_italian, x, y)
                if venue_distance<=300:
                    area_restaurants.append(restaurant)
                restaurants[venue_id] = restaurant
                if is_italian:
                    italian_restaurants[venue_id] = restaurant
        location_restaurants.append(area_restaurants)
        print(' .', end='')
    print(' done.')
    return restaurants, italian_restaurants, location_restaurants

# Try to load from local file system in case we did this before
restaurants = {}
italian_restaurants = {}
location_restaurants = []
loaded = False
try:
    with open('restaurants_350.pkl', 'rb') as f:
        restaurants = pickle.load(f)
    with open('italian_restaurants_350.pkl', 'rb') as f:
        italian_restaurants = pickle.load(f)
    with open('location_restaurants_350.pkl', 'rb') as f:
        location_restaurants = pickle.load(f)
    print('Restaurant data loaded.')
    loaded = True
except:
    pass
```

```python
# If load failed use the Foursquare API to get the data
if not loaded:
    restaurants, italian_restaurants, location_restaurants = get_restaurants(lat
itudes, longitudes)

    # Let's persists this in local file system
    with open('restaurants_350.pkl', 'wb') as f:
        pickle.dump(restaurants, f)
    with open('italian_restaurants_350.pkl', 'wb') as f:
        pickle.dump(italian_restaurants, f)
    with open('location_restaurants_350.pkl', 'wb') as f:
        pickle.dump(location_restaurants, f)
```

```
Restaurant data loaded.
```

In [55]:

```python
import numpy as np

print('Total number of restaurants:', len(restaurants))
print('Total number of Italian restaurants:', len(italian_restaurants))
print('Percentage of Italian restaurants: {:.2f}%'.format(len(italian_restaurant
s) / len(restaurants) * 100))
print('Average number of restaurants in neighborhood:', np.array([len(r) for r i
n location_restaurants]).mean())
```

```
Total number of restaurants: 2573
Total number of Italian restaurants: 163
Percentage of Italian restaurants: 6.34%
Average number of restaurants in neighborhood: 13.847222222222221
```

In [56]:

```
print('List of all restaurants')
print('-----------------------')
for r in list(restaurants.values())[:10]:
    print(r)
print('...')
print('Total:', len(restaurants))
```

```
List of all restaurants
-----------------------
('5a4e98dcd03360688d95dd73', 'Kobuta Ramen', 41.36986017860928, 2.13
31966064726955, 'Súria, 8, 08014 Barcelona Cataluña, España', 330, F
alse, 427509.1859078528, 4580178.795731767)
('4c0b8cd5009a0f47273cebbf', 'La Bodegueta de Cal Pep', 41.373775964
64733, 2.1323718328933725, 'Canalejas, 12, Barcelona Cataluña, Españ
a', 198, False, 427444.56070112495, 4580614.209745048)
('4d02751568e38eec7167dfc4', 'Mson', 41.37280423695585, 2.1287224542
23953, 'C. Bacardí, 31 (C. Sugranyes), 08028 Barcelona Cataluña, Esp
aña', 251, False, 427138.2938704959, 4580509.391376183)
('59419af506fb6007376c90ec', 'Amassame', 41.375023, 2.1327457, 'Carr
er de Santa Medir 8, 08028 Barcelona Cataluña, España', 254, True, 4
27477.2108730002, 4580752.340675035)
('4e75111dae60c32850f7bfc0', 'El Candil', 41.37168651011002, 2.12918
9266294637, 'Pavia, 76 (Carreras Candi), Barcelona Cataluña, Españ
a', 315, False, 427176.08540046006, 4580384.9109116)
('4dbd59f443a1d8504ba2ddbe', 'El Rincón del Espino', 41.373861946457
75, 2.133364677429199, 'Sant Medir, 17, 08028 Barcelona Cataluña, Es
paña', 135, False, 427527.6836168759, 4580622.924691)
('52233bcf11d2b1585cb4a7f3', 'La Bodegueta de Sants', 41.37096933379
5266, 2.13453983608806l, 'Andalusia (Sagunt), Barcelona Cataluña, Es
paña', 184, False, 427622.7507793935, 4580300.809801028)
('4c0cf640b1b676b04ecddf86', 'Fondevila', 41.37480518653695, 2.13284
60116606256, 'C. Sant Medir, 14, Barcelona Cataluña, España', 114, F
alse, 427485.35743848566, 4580728.075451704)
('4f0d7904e4b0254b4cc138a3', 'Can Coca', 41.373033623290524, 2.13490
64780930473, 'Calle de los Juegos Florales, 78, 08014 Barcelona, 080
14 Barcelona Cataluña, España', 344, False, 427655.70004873595, 4580
529.67765062)
('56013354498ef4753686654b', 'El Bar del Mercat', 41.374822, 2.13327
2, 'Sant Medir (Casteras), 08028 Barcelona Cataluña, España', 78, Fa
lse, 427520.9991821838, 4580729.585750371)
...
Total: 2573
```

In [57]:

```
print('List of Italian restaurants')
print('---------------------------')
for r in list(italian_restaurants.values())[:10]:
    print(r)
print('...')
print('Total:', len(italian_restaurants))
```

```
List of Italian restaurants
---------------------------
('59419af506fb6007376c90ec', 'Amassame', 41.375023, 2.1327457, 'Carr
er de Santa Medir 8, 08028 Barcelona Cataluña, España', 254, True, 4
27477.2108730002, 4580752.340675035)
('4b689a67f964a52058822be3', 'La Briciola', 41.373719, 2.136506, 'Ol
zinelles, 19, 08014 Barcelona Cataluña, España', 202, True, 427790.2
2095180367, 4580604.43325217)
('4ba37733f964a5206c3f38e3', 'Teta de Monja', 41.37609405654974, 2.1
388755859792195, 'Pl. Osca, 2, 08014 Barcelona Cataluña, España', 26
1, True, 427990.9993548109, 4580866.136735985)
('4b6363cdf964a52063762ae3', 'Il Golfo di Napoli', 41.372321, 2.1553
62, 'Carrer Lleida 38, 08004 Barcelona Cataluña, España', 299, True,
429365.5584492378, 4580433.692575301)
('5a359979535d6f0cd0798a9f', 'Macchina', 41.375325418929144, 2.16108
0653801404, 'Parlament 1, 08015 Barcelona Cataluña, España', 320, Tr
ue, 429847.02426497947, 4580762.593603634)
('4b58677df964a5203f5628e3', 'La Bella Napoli', 41.37434432358266,
2.1637781112076477, 'Margarit 12, 08004 Barcelona Cataluña, España',
285, True, 430071.5450651584, 4580651.494353807)
('5b8ace72a0215b002ca704ce', 'BENZiNA', 41.376195, 2.162716, 'Passat
ge de Pere Calders, 6, 08015 Barcelona Cataluña, España', 286, True,
429984.7106900246, 4580857.810666056)
('4b911501f964a520d6a233e3', 'Il Mercante di Venezia', 41.3774032573
6459, 2.1779308409350047, 'C. Josep Anselm Clavé, 11, 08002 Barcelon
a Cataluña, España', 330, True, 431258.28834649164, 4580979.77032536
5)
('583ae703966e55479de85fd1', "Cecconi's", 41.378315450395725, 2.1795
368061941542, 'Passeig de Colom, 20, 08002 Barcelona Cataluña, Espań
a', 195, True, 431393.539560358, 4581079.76791008)
('4eecedd3aa1f98ce65265f22', 'Tucco', 41.37939270093422, 2.178638979
824464, 'C. Còdols, 27, 08002 Barcelona Cataluña, España', 332, Tru
e, 431319.5967796244, 4581200.073498989)
...
Total: 163
```

In [58]:

```
print('Restaurants around location')
print('---------------------------')
for i in range(100, 110):
    rs = location_restaurants[i][:8]
    names = ', '.join([r[1] for r in rs])
    print('Restaurants around location {}: {}'.format(i+1, names))
```

```
Restaurants around location
---------------------------
Restaurants around location 101: bcnKITCHEN, La Paradeta, Llamber, L
a Ciudadela, Murivecchi, Taquería Canta Y No Llores, Bar Celta Pulpe
ría, Buon Appetito
Restaurants around location 102: Rincón de Galicia
Restaurants around location 103: Ninoska Restaurante, TaTeTí, Els Po
llos de Llull, Pizzeria Roma, Rincón de Galicia, Bar de Baix, Restau
rant Terrari, Sushi 10
Restaurants around location 104: Restaurant Ají, Ninoska Restaurant
e, TaTeTí, Ugarit, Curry Barcelona, Enoteca, Barnabier, Jerusalem Re
staurante - Shisha Bar
Restaurants around location 105: La Barca del Salamanca, La Fonda de
l Port Olímpic, El Tinglado, La Taberna Gallega, El Cangrejo Loco, E
l Rey  de La Gamba II, Ugarit, مطعم خليجيه - Khalijia (Khalijia)
Restaurants around location 106: La Grangeta, Can Fusté, Aire, Resta
urante La Traviata, Bar Bayo, Glub, La Granja de Santander, Cafetete
ria Tanatori Les Corts
Restaurants around location 107: Punta Anguila, Leku, Setze, L'Altre
Caliu de Finestrelles, Alive, 4 Trocua, Femmena, Can Fusté
Restaurants around location 108: Punta Anguila, Setze, Leku, Polka B
arcelona, Ramen-ya Ajisen, Fragments Cafè, popeye, El Rebost Ibèric
Restaurants around location 109: Bangkok Cafe, Ramen-ya Ajisen, Rest
aurante Piornedo, Lagunak, La Mama Acasă, Restaurante Chino Hoy, El
Jardí de Bambú, Café Roslin
Restaurants around location 110: La cuina de l'Uribou, Restaurante P
iornedo, Charcutería-Restaurante Sanabres, Koxkera, Fukuya, Liban, C
ocina Hermanos Torres, Macao
```

In [227]:

```
map_barcelona = folium.Map(location=barcelona_center, zoom_start=13)
folium.Marker(barcelona_center, popup='Plaza Catalunya').add_to(map_barcelona)
for res in restaurants.values():
    lat = res[2]; lon = res[3]
    is_italian = res[6]
    color = 'red' if is_italian else 'blue'
    folium.CircleMarker([lat, lon], radius=3, color=color, fill=True, fill_color
=color, fill_opacity=1).add_to(map_barcelona)
map_barcelona
```

Out[227]:

Looking good. So now we have all the restaurants in area within few kilometers from Barcelona centre, and we know which ones are Italian restaurants! We also know which restaurants exactly are in vicinity of every neighborhood candidate center.

This concludes the data gathering phase - we're now ready to use this data for analysis to produce the report on optimal locations for a new Italian restaurant!

Methodology

In this project we will direct our efforts on detecting areas of Barcelona that have low restaurant density, particularly those with low number of Italian restaurants. We will limit our analysis to area approximately 6km around the city center.

In first step we have collected the required **data: location and type (category) of every restaurant within 6km from Barcelona center** (Plaza Catalunya). We have also **identified Italian restaurants** (according to Foursquare categorization).

Second step in our analysis will be calculation and exploration of '**restaurant density**' across different areas of Barcelona - we will use **heatmaps** to identify a few promising areas close to center with low number of restaurants in general (*and* no Italian restaurants in vicinity) and focus our attention on those areas.

In third and final step we will focus on most promising areas and within those create **clusters of locations that meet some basic requirements** established in discussion with stakeholders: we will take into consideration locations with **no more than two restaurants in radius of 250 meters**, and we want locations **without Italian restaurants in radius of 400 meters**. We will present map of all such locations but also create clusters (using **k-means clustering**) of those locations to identify general zones / neighborhoods / addresses which should be a starting point for final 'street level' exploration and search for optimal venue location by stakeholders.

Analysis

Let's perform some basic explanatory data analysis and derive some additional info from our raw data. First let's count the **number of restaurants in every area candidate**:

In [102]:

```
#location_restaurants
#df_locations.append['Restaurants in area']
df_locations.head()
```

Out[102]:

	Address	Latitude	Longitude	X	Y	Distance from center
0	Dàrsena Sud, 08040 Barcelona	41.334521	2.148919	428785.515902	4.576243e+06	5992.495307
1	Ml Inflamables, 4, 08040 Barcelona	41.334574	2.156088	429385.515902	4.576243e+06	5840.376700
2	Carrer del Port de Ningbó, Barcelona	41.334626	2.163258	429985.515902	4.576243e+06	5747.173218
3	Spain	41.334678	2.170428	430585.515902	4.576243e+06	5715.767665
4	Spain	41.334730	2.177598	431185.515902	4.576243e+06	5747.173218

In [103]:

```
location_restaurants_count
```

```
 0,
 2,
 2,
 0,
 1,
 1,
 1,
 1,
 2,
 11,
 15,
 7,
 5,
 6,
 11,
 13,
 14,
 11]
```

In [108]:

```python
location_restaurants_count = [len(res) for res in location_restaurants]

#df_locations['Restaurants in area'] = location_restaurants_count

print('Average number of restaurants in every area with radius=300m:', np.array(
location_restaurants_count).mean())

df_locations.head(10)
```

```
Average number of restaurants in every area with radius=300m: 13.847
222222222221
```

Out[108]:

	Address	Latitude	Longitude	X	Y	Distance from center
0	Dàrsena Sud, 08040 Barcelona	41.334521	2.148919	428785.515902	4.576243e+06	5992.495307
1	Ml Inflamables, 4, 08040 Barcelona	41.334574	2.156088	429385.515902	4.576243e+06	5840.376700
2	Carrer del Port de Ningbó, Barcelona	41.334626	2.163258	429985.515902	4.576243e+06	5747.173218
3	Spain	41.334678	2.170428	430585.515902	4.576243e+06	5715.767665
4	Spain	41.334730	2.177598	431185.515902	4.576243e+06	5747.173218
5	Spain	41.334781	2.184768	431785.515902	4.576243e+06	5840.376700
6	Spain	41.334831	2.191938	432385.515902	4.576243e+06	5992.495307
7	Unnamed Road, 08040 Barcelona	41.339121	2.138102	427885.515902	4.576762e+06	5855.766389
8	Via Circulació del Nord, 8, 08040 Barcelona	41.339175	2.145272	428485.515902	4.576762e+06	5604.462508
9	Ml Inflamables, 1, 08040 Barcelona	41.339228	2.152443	429085.515902	4.576762e+06	5408.326913

OK, now let's calculate the **distance to nearest Italian restaurant from every area candidate center** (not only those within 300m - we want distance to closest one, regardless of how distant it is).

In [109]:

```
distances_to_italian_restaurant = []

for area_x, area_y in zip(xs, ys):
    min_distance = 10000
    for res in italian_restaurants.values():
        res_x = res[7]
        res_y = res[8]
        d = calc_xy_distance(area_x, area_y, res_x, res_y)
        if d<min_distance:
            min_distance = d
    distances_to_italian_restaurant.append(min_distance)

df_locations['Distance to Italian restaurant'] = distances_to_italian_restaurant
```

In [110]:

```
df_locations.head(10)
```

Out[110]:

	Address	Latitude	Longitude	X	Y	Distance from center	Distance to Italian restaurant
0	Dàrsena Sud, 08040 Barcelona	41.334521	2.148919	428785.515902	4.576243e+06	5992.495307	4231.131955
1	MI Inflamables, 4, 08040 Barcelona	41.334574	2.156088	429385.515902	4.576243e+06	5840.376700	4191.232105
2	Carrer del Port de Ningbó, Barcelona	41.334626	2.163258	429985.515902	4.576243e+06	5747.173218	4236.788348
3	Spain	41.334678	2.170428	430585.515902	4.576243e+06	5715.767665	4365.125937
4	Spain	41.334730	2.177598	431185.515902	4.576243e+06	5747.173218	4544.690153
5	Spain	41.334781	2.184768	431785.515902	4.576243e+06	5840.376700	4510.751850
6	Spain	41.334831	2.191938	432385.515902	4.576243e+06	5992.495307	4556.265566
7	Unnamed Road, 08040 Barcelona	41.339121	2.138102	427885.515902	4.576762e+06	5855.766389	3843.491570
8	Via Circulació del Nord, 8, 08040 Barcelona	41.339175	2.145272	428485.515902	4.576762e+06	5604.462508	3775.565700
9	MI Inflamables, 1, 08040 Barcelona	41.339228	2.152443	429085.515902	4.576762e+06	5408.326913	3682.233738

In [111]:

```
print('Average distance to closest Italian restaurant from each area center:', d
f_locations['Distance to Italian restaurant'].mean())
```

```
Average distance to closest Italian restaurant from each area cente
r: 1465.343661241832
```

OK, so **on average Italian restaurant can be found within ~1,500m** from every area center candidate. That's not fairly close, so we need to filter our areas further!

In [150]:

```
# barna_boroughs = bcncity_data
bcncity_data
```

Out[150]:

	Borough	Neighborhood	Latitude	Longitude
0	BARRI	el Raval	4.581121e+06	430624.9313
1	BARRI	el Barri Gòtic	4.581289e+06	431291.4440
2	BARRI	la Barceloneta	4.581448e+06	432355.6530
3	BARRI	Sant Pere, Santa Caterina i la Ribera	4.581984e+06	431707.9381
4	BARRI	el Fort Pienc	4.583261e+06	431580.2748
5	BARRI	la Sagrada Família	4.584175e+06	431275.9502
6	BARRI	la Dreta de l'Eixample	4.582931e+06	430582.0822
7	BARRI	l'Antiga Esquerra de l'Eixample	4.582208e+06	429278.2125
8	BARRI	la Nova Esquerra de l'Eixample	4.581700e+06	428919.9315
9	BARRI	Sant Antoni	4.581114e+06	429725.6300
10	BARRI	el Poble-sec	4.580256e+06	429923.4210
11	BARRI	la Marina del Prat Vermell	4.578253e+06	428248.5990
12	BARRI	la Marina del Prat Vermell	4.577804e+06	430884.3833
13	BARRI	la Marina de Port	4.579000e+06	427948.9540
14	BARRI	la Font de la Guatlla	4.580223e+06	428650.9413
15	BARRI	Hostafrancs	4.580703e+06	428474.2800
16	BARRI	la Bordeta	4.580015e+06	427799.1118
17	BARRI	Sants - Badal	4.580735e+06	426997.4560
18	BARRI	Sants	4.581091e+06	427720.7800
19	BARRI	les Corts	4.582129e+06	427607.7907
20	BARRI	la Maternitat i Sant Ramon	4.581789e+06	426483.0492
21	BARRI	Pedralbes	4.582976e+06	425245.4386
22	BARRI	Vallvidrera, el Tibidabo i les Planes	4.585915e+06	423872.3530
23	BARRI	Vallvidrera, el Tibidabo i les Planes	4.586064e+06	421484.4561
24	BARRI	Sarrià	4.583963e+06	425963.4880
25	BARRI	les Tres Torres	4.583348e+06	427365.1885
26	BARRI	Sant Gervasi - la Bonanova	4.584675e+06	427300.3170
27	BARRI	Sant Gervasi - Galvany	4.583170e+06	428386.6714
28	BARRI	el Putxet i el Farró	4.584292e+06	428391.1766
29	BARRI	Vallcarca i els Penitents	4.585320e+06	428209.9230
30	BARRI	el Coll	4.585593e+06	428725.3020
31	BARRI	la Salut	4.584899e+06	429353.2173
32	BARRI	la Vila de Gràcia	4.583965e+06	429618.1455
33	BARRI	el Camp d'en Grassot i Gràcia Nova	4.584181e+06	430177.7815
34	BARRI	el Baix Guinardó	4.584784e+06	430285.8203
35	BARRI	Can Baró	4.585392e+06	430035.1590
36	BARRI	el Guinardó	4.585269e+06	430787.8776

	Borough	Neighborhood	Latitude	Longitude
37	BARRI	la Font d'en Fargues	4.586306e+06	430148.2884
38	BARRI	el Carmel	4.585900e+06	429464.0139
39	BARRI	la Teixonera	4.586016e+06	428555.7485
40	BARRI	Sant Genís dels Agudells	4.586631e+06	427294.9743
41	BARRI	Montbau	4.587588e+06	427942.1540
42	BARRI	la Vall d'Hebron	4.587009e+06	429036.2642
43	BARRI	la Clota	4.586856e+06	429288.4997
44	BARRI	Horta	4.588360e+06	429020.3780
45	BARRI	Vilapicina i la Torre Llobeta	4.586789e+06	430969.3133
46	BARRI	Porta	4.587280e+06	431362.4560
47	BARRI	el Turó de la Peira	4.587033e+06	430560.3190
48	BARRI	Can Peguera	4.587376e+06	430409.1821
49	BARRI	la Guineueta	4.587776e+06	430558.1946
50	BARRI	Canyelles	4.588756e+06	429963.1507
51	BARRI	les Roquetes	4.588829e+06	431087.5092
52	BARRI	Verdun	4.588238e+06	431040.8280
53	BARRI	la Prosperitat	4.588073e+06	431476.2362
54	BARRI	la Trinitat Nova	4.588993e+06	431914.5304
55	BARRI	Torre Baró	4.589708e+06	430839.9371
56	BARRI	Ciutat Meridiana	4.590178e+06	431013.9470
57	BARRI	Vallbona	4.590767e+06	431823.5740
58	BARRI	la Trinitat Vella	4.588932e+06	432759.0070
59	BARRI	Baró de Viver	4.588625e+06	433138.3370
60	BARRI	el Bon Pastor	4.587942e+06	433507.6532
61	BARRI	Sant Andreu	4.587615e+06	432238.9760
62	BARRI	la Sagrera	4.586152e+06	432278.1479
63	BARRI	el Congrés i els Indians	4.586382e+06	431523.2113
64	BARRI	Navas	4.585491e+06	431997.1546
65	BARRI	el Camp de l'Arpa del Clot	4.584759e+06	431683.4094
66	BARRI	el Clot	4.584571e+06	432272.9877
67	BARRI	el Parc i la Llacuna del Poblenou	4.583165e+06	432334.2085
68	BARRI	la Vila Olímpica del Poblenou	4.582419e+06	432742.4580
69	BARRI	el Poblenou	4.583459e+06	433274.1312
70	BARRI	Diagonal Mar i el Front Marítim del Poblenou	4.584126e+06	434167.1910
71	BARRI	el Besòs i el Maresme	4.584958e+06	434480.1580
72	BARRI	Provençals del Poblenou	4.584606e+06	433324.9830
73	BARRI	Sant Martí de Provençals	4.585222e+06	432910.1121
74	BARRI	la Verneda i la Pau	4.586091e+06	433447.4467

In [123]:

```
def boroughs_style(feature):
    return { 'color': 'blue', 'fill': False }
```

In [124]:

```
restaurant_latlons = [[res[2], res[3]] for res in restaurants.values()]

italian_latlons = [[res[2], res[3]] for res in italian_restaurants.values()]
```

In [141]:

```
from folium import plugins
from folium.plugins import HeatMap

map_barcelona = folium.Map(location=barcelona_center, zoom_start=13)
folium.TileLayer('cartodbpositron').add_to(map_barcelona) #cartodbpositron carto
dbdark_matter
HeatMap(restaurant_latlons).add_to(map_barcelona)
folium.Marker(barcelona_center).add_to(map_barcelona)
folium.Circle(barcelona_center, radius=1000, fill=False, color='white').add_to(m
ap_barcelona)
folium.Circle(barcelona_center, radius=2000, fill=False, color='white').add_to(m
ap_barcelona)
folium.Circle(barcelona_center, radius=3000, fill=False, color='white').add_to(m
ap_barcelona)
#folium.GeoJson(bcncity_data, style_function=boroughs_style, name='geojson').add
_to(map_barcelona)

# add markers to map
for lat, lng, name, category in zip(bcncity_data['Latitude'], bcncity_data['Long
itude'], bcncity_data['Neighborhood'], bcncity_data['Borough']):
    label = '{}, {}'.format(name, category)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='white',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_barcelona)


map_barcelona
```

Out[141]:

Looks like a few pockets of low restaurant density closest to city center can be found **north-east and south-west from Plaza Catalunya**.

Let's create another heatmap map showing **heatmap/density of Italian restaurants** only.

In [147]:

```
map_barcelona = folium.Map(location=barcelona_center, zoom_start=13)
folium.TileLayer('cartodbpositron').add_to(map_barcelona) #cartodbpositron cartodbdark_matter
HeatMap(italian_latlons).add_to(map_barcelona)
folium.Marker(barcelona_center).add_to(map_barcelona)
folium.Circle(barcelona_center, radius=1000, fill=False, color='white').add_to(map_barcelona)
folium.Circle(barcelona_center, radius=2000, fill=False, color='white').add_to(map_barcelona)
folium.Circle(barcelona_center, radius=3000, fill=False, color='white').add_to(map_barcelona)
#folium.GeoJson(bcncity_data, style_function=boroughs_style, name='geojson').add_to(map_barcelona)

map_barcelona
```

Out[147]:

This map is not so 'hot' (Italian restaurants represent a subset of ~13.5% of all restaurants in Barcelona) but it also indicates higher density of existing Italian restaurants directly north-west from Plaza Catalunya, with closest pockets of **low Italian restaurant density positioned north-east and south-west from city center**.

Based on this we will now focus our analysis on areas *north-east and south-west from the city center* - we will move the center of our area of interest and reduce it's size to have a radius of **3km**. This places our location candidates mostly in boroughs **Hostafrancs and Poblenou** which we will check whether they have popular places where locals and tourist come by often.

Hostafrancs and Poblenou

Analysis of popular travel guides and web sites often mention Kreuzberg and Friedrichshain as beautifull, interesting, rich with culture, 'hip' and 'cool' Berlin neighborhoods popular with tourists and loved by Berliners.

"*Bold and brazen, Kreuzberg's creative people, places, and spaces might challenge your paradigm.*" Tags: Nightlife, Artsy, Dining, Trendy, Loved by Berliners, Great Transit (airbnb.com)

"*Kreuzberg has long been revered for its diverse cultural life and as a part of Berlin where alternative lifestyles have flourished. Envisioning the glamorous yet gritty nature of Berlin often conjures up scenes from this neighbourhood, where cultures, movements and artistic flare adorn the walls of building and fills the air. Brimming with nightclubs, street food, and art galleries, Kreuzberg is the place to be for Berlin's young and trendy.*" (theculturetrip.com)

"*Imagine an art gallery turned inside out and you'll begin to envision Friedrichshain. Single walls aren't canvases for creative works, entire buildings are canvases. This zealously expressive east Berlin neighborhood forgoes social norms*" Tags: Artsy, Nightlife, Trendy, Dining, Touristy, Shopping, Great Transit, Loved by Berliners (airbnb.com)

"*As anyone from Kreuzberg will tell you, this district is not just the coolest in Berlin, but the hippest location in the entire universe. Kreuzberg has long been famed for its diverse cultural life, its experimental alternative lifestyles and the powerful spell it exercises on young people from across Germany. In 2001, Kreuzberg and Friedrichshain were merged to form one administrative borough. When it comes to club culture, Friedrichshain is now out in front – with southern Friedrichshain particularly ranked as home to the highest density of clubs in the city.*" (visitberlin.de)

Popular with tourists, alternative and bohemian but booming and trendy, relatively close to city center and well connected, those boroughs appear to justify further analysis.

Let's define new, more narrow region of interest, which will include low-restaurant-count parts of Kreuzberg and Friedrichshain closest to Alexanderplatz.

Explore the locations in terms of popular places on Foursquare

In [199]:

```
import requests
radius = 3000
```

In [200]:

```
bcncity_data.head(30)
```

Out[200]:

	Borough	Neighborhood	Latitude	Longitude
0	BARRI	el Raval	4.581121e+06	430624.9313
1	BARRI	el Barri Gòtic	4.581289e+06	431291.4440
2	BARRI	la Barceloneta	4.581448e+06	432355.6530
3	BARRI	Sant Pere, Santa Caterina i la Ribera	4.581984e+06	431707.9381
4	BARRI	el Fort Pienc	4.583261e+06	431580.2748
5	BARRI	la Sagrada Família	4.584175e+06	431275.9502
6	BARRI	la Dreta de l'Eixample	4.582931e+06	430582.0822
7	BARRI	l'Antiga Esquerra de l'Eixample	4.582208e+06	429278.2125
8	BARRI	la Nova Esquerra de l'Eixample	4.581700e+06	428919.9315
9	BARRI	Sant Antoni	4.581114e+06	429725.6300
10	BARRI	el Poble-sec	4.580256e+06	429923.4210
11	BARRI	la Marina del Prat Vermell	4.578253e+06	428248.5990
12	BARRI	la Marina del Prat Vermell	4.577804e+06	430884.3833
13	BARRI	la Marina de Port	4.579000e+06	427948.9540
14	BARRI	la Font de la Guatlla	4.580223e+06	428650.9413
15	BARRI	Hostafrancs	4.580703e+06	428474.2800
16	BARRI	la Bordeta	4.580015e+06	427799.1118
17	BARRI	Sants - Badal	4.580735e+06	426997.4560
18	BARRI	Sants	4.581091e+06	427720.7800
19	BARRI	les Corts	4.582129e+06	427607.7907
20	BARRI	la Maternitat i Sant Ramon	4.581789e+06	426483.0492
21	BARRI	Pedralbes	4.582976e+06	425245.4386
22	BARRI	Vallvidrera, el Tibidabo i les Planes	4.585915e+06	423872.3530
23	BARRI	Vallvidrera, el Tibidabo i les Planes	4.586064e+06	421484.4561
24	BARRI	Sarrià	4.583963e+06	425963.4880
25	BARRI	les Tres Torres	4.583348e+06	427365.1885
26	BARRI	Sant Gervasi - la Bonanova	4.584675e+06	427300.3170
27	BARRI	Sant Gervasi - Galvany	4.583170e+06	428386.6714
28	BARRI	el Putxet i el Farró	4.584292e+06	428391.1766
29	BARRI	Vallcarca i els Penitents	4.585320e+06	428209.9230

In [201]:

```
address = 'Hostafrancs, Barcelona, Spain'

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
newlat = location.latitude
newlong = location.longitude
print(newlat, newlong)
```

```
41.3750877 2.1429334
```

In [202]:

```
urlHost = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(client_id, client_secret, newlat, newlong, version, radius, limit)
urlHost
```

Out[202]:

```
'https://api.foursquare.com/v2/venues/explore?client_id=0SNKFIVOMYF1JOTTLWIRLO55QF11PAMDVGXXQ3OGZFMBYE53&client_secret=3TJE0KSUGPF0ZXYZQUS4HRRLMFSUQ403Q4WMSPZD2SDEOG4F&ll=41.3750877,2.1429334&v=20190604&radius=3000&limit=2000'
```

In [203]:

```
resultsHost = requests.get(urlHost).json()
resultsHost
```

```
          'shortName': 'Pizza',
          'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/
food/pizza_',
           'suffix': '.png'},
          'primary': True}],
        'photos': {'count': 0, 'groups': []}},
       'referralId': 'e-0-55e6ccca498e525119f14934-99'}]}]}}
```


In [204]:

```
address = 'Poblenou, Barcelona, Spain'

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
newlat2 = location.latitude
newlong2 = location.longitude
print(newlat2, newlong2)
```

```
41.400527 2.2017292
```

In [205]:

```
urlHost2 = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(client_id, client_secret, newlat2, newlong2, version, radius, limit)
urlHost2
```

Out[205]:

```
'https://api.foursquare.com/v2/venues/explore?client_id=0SNKFIVOMYF1
JOTTLWIRLO55QF11PAMDVGXXQ3OGZFMBYE53&client_secret=3TJE0KSUGPF0ZXYZQ
US4HRRLMFSUQ403Q4WMSPZD2SDEOG4F&ll=41.400527,2.2017292&v=20190604&ra
dius=3000&limit=2000'
```

In [206]:

```
resultsHost2 = requests.get(urlHost).json()
resultsHost2
```

```
          'shortName': 'Pizza',
          'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/
food/pizza_',
           'suffix': '.png'},
          'primary': True}],
        'photos': {'count': 0, 'groups': []}},
       'referralId': 'e-0-55e6ccca498e525119f14934-99'}]}]}}
```


Examine results

In [198]:

```
resultsHost = requests.get(urlHost).json()
'There are {} around Hostafrancs.'.format(len(resultsHost['response']['groups'][
0]['items']))
```

Out[198]:

```
'There are 100 around Hostafrancs.'
```

In [210]:

```
items1 = resultsHost['response']['groups'][0]['items']
items1[0]
```

Out[210]:


```
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
    'reasonName': 'globalInteractionReason'}]},
 'venue': {'id': '5831a67ccc05d15be8a8136f',
  'name': 'La Vicoca',
  'location': {'address': 'Hostafrancs de Sió, 18',
   'crossStreet': 'Entre Vilardell y Leiva',
   'lat': 41.374161,
   'lng': 2.144223,
   'labeledLatLngs': [{'label': 'display', 'lat': 41.374161, 'lng':
2.144223}],
   'distance': 149,
   'postalCode': '08014',
   'cc': 'ES',
   'city': 'Barcelona',
   'state': 'Cataluña',
   'country': 'España',
   'formattedAddress': ['Hostafrancs de Sió, 18 (Entre Vilardell y L
eiva)',
    '08014 Barcelona Cataluña',
    'España']},
  'categories': [{'id': '4bf58dd8d48988d123941735',
    'name': 'Wine Bar',
    'pluralName': 'Wine Bars',
    'shortName': 'Wine Bar',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/
winery_',
     'suffix': '.png'},
    'primary': True}],
  'photos': {'count': 0, 'groups': []},
  'venuePage': {'id': '372083227'}},
 'referralId': 'e-0-5831a67ccc05d15be8a8136f-0'}
```


In [207]:

```
resultsHost2 = requests.get(urlHost2).json()
'There are {} around Poblenou.'.format(len(resultsHost2['response']['groups'][0]
['items']))
```

Out[207]:

```
'There are 100 around Poblenou.'
```

In [213]:

```
items2 = resultsHost2['response']['groups'][0]['items']
items2[0]
```

Out[213]:


```
{'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
    'reasonName': 'globalInteractionReason'}]},
 'venue': {'id': '5034b740e4b0ab0c7394518a',
  'name': 'Cruixent BCN',
  'location': {'address': 'Pujades, 173',
   'crossStreet': 'Rbla. Poblenou',
   'lat': 41.40188991912975,
   'lng': 2.20072980500845,
   'labeledLatLngs': [{'label': 'display',
     'lat': 41.40188991912975,
     'lng': 2.20072980500845}],
   'distance': 173,
   'postalCode': '08005',
   'cc': 'ES',
   'city': 'Barcelona',
   'state': 'Cataluña',
   'country': 'España',
   'formattedAddress': ['Pujades, 173 (Rbla. Poblenou)',
    '08005 Barcelona Cataluña',
    'España']},
  'categories': [{'id': '4bf58dd8d48988d16a941735',
    'name': 'Bakery',
    'pluralName': 'Bakeries',
    'shortName': 'Bakery',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/
bakery_',
     'suffix': '.png'},
    'primary': True}],
  'photos': {'count': 0, 'groups': []}},
 'referralId': 'e-0-5034b740e4b0ab0c7394518a-0'}
```

In [219]:

```
dataframe1 = json_normalize(items1) # flatten JSON

# filter columns
filtered_columns1 = ['venue.name', 'venue.categories'] + [col for col in datafra
me1.columns if col.startswith('venue.location.')] + ['venue.id']
dataframe_filtered1 = dataframe1.loc[:, filtered_columns1]

# filter the category for each row
dataframe_filtered1['venue.categories'] = dataframe_filtered1.apply(get_category
_type, axis=1)

# clean columns
dataframe_filtered1.columns = [col.split('.')[-1] for col in dataframe_filtered1
.columns]

dataframe_filtered1.head(10)
```

Out[219]:

	name	categories	address	crossStreet	lat	lng	labeledLatLng
0	La Vicoca	Wine Bar	Hostafrancs de Sió, 18	Entre Vilardell y Leiva	41.374161	2.144223	[{'label': 'display', 'lat 41.374161, 'lng':.
1	La Caleta de Sants	Restaurant	Ctra. de la Bordeta, 54	NaN	41.373525	2.145157	[{'label': 'display', 'lat 41.37352504116426.
2	Zumzeig Cinema	Indie Movie Theater	C. Béjar, 53	NaN	41.377350	2.145076	[{'label': 'display', 'lat 41.37734966171251.
3	Petit Pau	Mediterranean Restaurant	C. Espanya Industrial, 22	NaN	41.376266	2.140496	[{'label': 'display', 'lat 41.37626588093834.
4	Plaça d'Espanya (Plaza de España)	Plaza	Pl. d'Espanya	NaN	41.375021	2.149115	[{'label': 'display', 'lat 41.37502128798987.
5	Casa Vives	Dessert Shop	C. Sants, 74	NaN	41.375408	2.137171	[{'label': 'display', 'lat 41.37540756780608.
6	La Mestressa	Tapas Restaurant	Plaça d'Osca, 7	NaN	41.376050	2.138702	[{'label': 'display', 'lat 41.37604998013339.
7	La terrassa Miró	Bar	Tarragona 129	NaN	41.377356	2.146089	[{'label': 'display', 'lat 41.37735601031151.
8	Tartela	Café	C. Llançà, 32	Diputació	41.377159	2.149722	[{'label': 'display', 'lat 41.377159, 'lng':.
9	Morrow Coffee	Coffee Shop	Av. Gran Via de les Corts Catalanes, 403	Vilamarí	41.377105	2.151378	[{'label': 'display', 'lat 41.37710480525754.

In [220]:

```
dataframe1.shape
```

Out[220]:

```
(100, 24)
```

In [221]:

```
dataframe2 = json_normalize(items2) # flatten JSON

# filter columns
filtered_columns2 = ['venue.name', 'venue.categories'] + [col for col in datafra
me2.columns if col.startswith('venue.location.')] + ['venue.id']
dataframe_filtered2 = dataframe2.loc[:, filtered_columns2]

# filter the category for each row
dataframe_filtered2['venue.categories'] = dataframe_filtered2.apply(get_category
_type, axis=1)

# clean columns
dataframe_filtered2.columns = [col.split('.')[-1] for col in dataframe_filtered2
.columns]

dataframe_filtered2.head(10)
```

Out[221]:

	name	categories	address	crossStreet	lat	lng	labeledLatLngs
0	Cruixent BCN	Bakery	Pujades, 173	Rbla. Poblenou	41.401890	2.200730	[{'label': 'display', 'lat': 41.40188991912975...
1	La Cervecita Nuestra de Cada Día	Beer Store	C. Llull, 184	Rambla del Poblenou	41.400454	2.201477	[{'label': 'display', 'lat': 41.40045398203872...
2	Melocomo	Italian Restaurant	Carrer de Pujades, 188	NaN	41.401788	2.200996	[{'label': 'display', 'lat': 41.401788, 'lng':...
3	La Tavernícola	Argentinian Restaurant	Roc Boronat, 70	Calle De Pujades	41.400349	2.197909	[{'label': 'display', 'lat': 41.40034854684869...
4	Dino's Ice Cream	Ice Cream Shop	Rambla Poble Nou 59	Llul	41.400910	2.201478	[{'label': 'display', 'lat': 41.40091005092279...
5	Rambla del Poblenou	Road	Rambla del Poblenou	NaN	41.401471	2.200484	[{'label': 'display', 'lat': 41.40147120183829...
6	Le Cinquante Huit	Gastropub	Rambla del Poblenou, 58	NaN	41.400653	2.202557	[{'label': 'display', 'lat': 41.40065308537522...
7	Can Dendê	Breakfast Spot	C. Ciutat de Granada, 44	C. de Llull	41.398296	2.198360	[{'label': 'display', 'lat': 41.39829633168689...
8	Little Fern	Breakfast Spot	C. Pere IV, 168	C. Llacuna	41.402232	2.197312	[{'label': 'display', 'lat': 41.40223206041831...
9	El Tío Che	Ice Cream Shop	Rambla del Poblenou, 44, TDA	NaN	41.400106	2.202770	[{'label': 'display', 'lat': 41.400105667, 'ln...

In [222]:

```
dataframe2.shape
```

Out[222]:

```
(100, 22)
```

clean columns

dataframe_filtered.columns = [col.split('.')[-1] for col in dataframe_filtered.columns]

dataframe_filtered.head(10)

In [224]:

```
# clean column names by keeping only last term
dataframe_filtered1.columns = [column.split('.')[-1] for column in dataframe_fil
tered1.columns]

dataframe_filtered1.head()
```

Out[224]:

	name	categories	address	crossStreet	lat	lng	labeledLatLng
0	La Vicoca	Wine Bar	Hostafrancs de Sió, 18	Entre Vilardell y Leiva	41.374161	2.144223	[{'label': 'display', 'lat 41.374161, 'lng':.
1	La Caleta de Sants	Restaurant	Ctra. de la Bordeta, 54	NaN	41.373525	2.145157	[{'label': 'display', 'lat 41.37352504116426.
2	Zumzeig Cinema	Indie Movie Theater	C. Béjar, 53	NaN	41.377350	2.145076	[{'label': 'display', 'lat 41.37734966171251.
3	Petit Pau	Mediterranean Restaurant	C. Espanya Industrial, 22	NaN	41.376266	2.140496	[{'label': 'display', 'lat 41.37626588093834.
4	Plaça d'Espanya (Plaza de España)	Plaza	Pl. d'Espanya	NaN	41.375021	2.149115	[{'label': 'display', 'lat 41.37502128798987.

In [225]:

```
# clean column names by keeping only last term
dataframe_filtered2.columns = [column.split('.')[-1] for column in dataframe_fil
tered2.columns]

dataframe_filtered2.head()
```

Out[225]:

	name	categories	address	crossStreet	lat	lng	labeledLatLngs	di
0	Cruixent BCN	Bakery	Pujades, 173	Rbla. Poblenou	41.401890	2.200730	[{'label': 'display', 'lat': 41.40188991912975...	
1	La Cervecita Nuestra de Cada Día	Beer Store	C. Llull, 184	Rambla del Poblenou	41.400454	2.201477	[{'label': 'display', 'lat': 41.40045398203872...	
2	Melocomo	Italian Restaurant	Carrer de Pujades, 188	NaN	41.401788	2.200996	[{'label': 'display', 'lat': 41.401788, 'lng':...	
3	La Tavernícola	Argentinian Restaurant	Roc Boronat, 70	Calle De Pujades	41.400349	2.197909	[{'label': 'display', 'lat': 41.40034854684869...	
4	Dino's Ice Cream	Ice Cream Shop	Rambla Poble Nou 59	Llul	41.400910	2.201478	[{'label': 'display', 'lat': 41.40091005092279...	

Let's visualize these items on the map around our location

In [232]:

```
print(latitude, longitude)
map_barcelona = folium.Map(location=[latitude, longitude], zoom_start=13) # gene
rate map centred Bcn Center

folium.Marker(barcelona_center, popup='Plaza Catalunya').add_to(map_barcelona)
for res in restaurants.values():
    lat = res[2]; lon = res[3]
    is_italian = res[6]
    color = 'red' if is_italian else 'blue'
    folium.CircleMarker([lat, lon], radius=3, color=color, fill=True, fill_color
=color, fill_opacity=1).add_to(map_barcelona)
map_barcelona

# add Center as a red circle mark
folium.features.CircleMarker(
    [latitude, longitude],
    radius=6,
    popup='Center',
    fill=True,
    color='red',
    fill_color='red',
    fill_opacity=0.6
    ).add_to(map_barcelona)

# add Hostafrancs markers to map
for lat, lng, name, category in zip(dataframe_filtered1['lat'], dataframe_filter
ed1['lng'], dataframe_filtered1['name'], dataframe_filtered1['categories']):
    label = '{}, {}'.format(name, category)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='orange',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_barcelona)

# add Poblenou markers to map
for lat, lng, name, category in zip(dataframe_filtered2['lat'], dataframe_filter
ed2['lng'], dataframe_filtered2['name'], dataframe_filtered2['categories']):
    label = '{}, {}'.format(name, category)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='yellow',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_barcelona)

# display map
map_barcelona
```

```
41.3861586 2.169774
```

Out[232]:

Let's define new, more narrow region of interest, which will include low-restaurant-count parts of Hostafrancs and Poblenou closest to Plaza Catalunya.

In [233]:

```
# @hidden_cell
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```