IBM CAPSTONE FINAL

VINICIUS RODRIGUES REGGIO

1. Introduction

1.1 Description of the problem.

Brazil has a huge coast and many coastal cities. There are famous and remarkable metropolis, like Rio de Janeiro, known as gorgeous tourist place. In order to achieve a business (restaurant), how could we provide answers to some questions like: What are the most commons types of restaurants? Which kind are they? How many are they in the city, and in which part of the city are they more common? In comparison to another city, is there a correlation between the types and the numbers?

1.1.1 Target Audience

The target audience is:

- food business entrepreneurs
- real state agents
- tourists that wants to see more deep the characteristics of the two cities

1.2 Discussion of the background

Rio de Janeiro is a Brazilian´s big city, with nearly 7 million inhabitants that receive more than 2 million tourists a year. Entertainment places well known are the “Cristo Redentor” and “Pão de Açúcar”, but there are many museums, beaches, parks, aquariums available to the visitors. For them, hotels and restaurants from different flags are available options. An ordinary traveler must spend a good money daily in restaurants, so this kind of business must be thriving in touristic places. The choice to open a restaurant from different categories must be supported by analysis whether there are same venues around and so on. And just for comparison, we´ll try to see, using the Fourquare API with the Folium library, if there is any kind of correlation between the restaurant types of Rio de Janeiro and other state capital and touristic city, Recife.

2. Description of the data

For the subsequent task, will be used two geojson files with the neighborhoods of Rio de Janeiro and Recife cities. They are available, respectively:

a) “https://opendata.arcgis.com/datasets/dc94b29fc3594a5bb4d297bee0c9a3f2_15.geojson";

b) “http://dados.recife.pe.gov.br/dataset/c1f100f0-f56f-4dd4-9dcc-1aa4da28798a/resource/e43bee60-9448-4d3d-92ff-2378bc3b5b00/download/bairros.geojson".

Besides, will be used the Geopy library to obtain latitudes and longitudes. Also, FourSquare API library, for the purpose of collect the data about the restaurants in each neighborhood of each city.

In the end, the last data explored is about the travelers from other countries which visit the cities and will be used to discuss the conclusion. It can be downloaded here:

http://www.dadosefatos.turismo.gov.br/2016-02-04-11-54-03/demanda-tur%C3%ADstica-internacional/item/download/980_7bcddf9f8e6f247f68c5f5754ce64df7.html

3. Methodology Section

In order to save time, I execute the commands of both cities concomitantly First, it´s necessary to import the libraries that will be used.

```
import pandas as pd
import numpy as np
import requests, folium, json
from unidecode import unidecode
from geopy.geocoders import Nominatim
from folium import plugins
from folium.plugins import MarkerCluster
import matplotlib as plt
import seaborn as sns
```

The first city to collect data is “Recife”. Below is the data to import the geojson. The content is the information about the neighborhoods, the latitudes as longitudes of their borders and their names.

```
s = requests.Session()
recife_geodata = s.get("http://dados.recife.pe.gov.br/dataset/c1f100f0-f56f-4dd4-9dcc-1aa4da28798a/
                        resource/e43bee60-9448-4d3d-92ff-2378bc3b5b00/download/bairros.geojson").json()
recife_geodata
```


```
{'type': 'FeatureCollection',
 'features': [{'type': 'Feature',
   'id': 0,
   'properties': {'bairro_codigo': 728,
    'bairro_nome_ca': 'CIDADE UNIVERSITARIA',
    'rpa': 4,
    'microrregiao': 3,
    'bairro_nome': 'Cidade Universitária'},
   'geometry': {'type': 'Polygon',
    'coordinates': [[[-34.944159036934906, -8.04984630501871],
      [-34.94419151990481, -8.050204895163283],
      [-34.94419743895523, -8.05027023442114],
      [-34.94431120757218, -8.051585950694623],
      [-34.94431131783154, -8.051594802624184],
      [-34.944311753991926, -8.051629758340791],
      [-34.942945018203595, -8.052009873555523],
      [-34.94315216357978, -8.053576771888464],
      [-34.94315812584044, -8.053621878336617],
      [-34.94601997813883, -8.052832348362527],
```


And here are the neighborhoods data from “Rio de Janeiro”.

```python
rio_geodata = s.get("https://opendata.arcgis.com/datasets/dc94b29fc3594a5bb4d297bee0c9a3f2_15.geojson").json()
rio_geodata
```


```
{'type': 'FeatureCollection',
 'name': 'Limite_de_Bairros',
 'crs': {'type': 'name',
  'properties': {'name': 'urn:ogc:def:crs:OGC:1.3:CRS84'}},
 'features': [{'type': 'Feature',
   'properties': {'OBJECTID': 325,
    'Área': 1705684.50390625,
    'NOME': 'Paquetá                  ',
    'REGIAO_ADM': 'PAQUETA                 ',
    'AREA_PLANE': '1',
    'CODBAIRRO': '013',
    'CODRA': 21,
    'CODBNUM': 13,
    'LINK': 'Paqueta&area=013                                  ',
    'RP': 'Centro',
    'Cod_RP': '1.1',
    'CODBAIRRO_LONG': 13,
    'SHAPESTArea': 1705684.5081324228,
    'SHAPESTLength': 24841.426668559936},
```


Now, it´s important to extract only the neighborhood´s names. This is how to do that.

```python
recife_neighborhood_list = []
for neighborhood in range(0,len(recife_geodata["features"])):
    recife_neighborhood_list.append((recife_geodata["features"][neighborhood]["properties"]['bairro_nome']))
recife_neighborhood_list
```

```
['Cidade Universitária',
 'Soledade',
 'Engenho do Meio',
 'Caçote',
 'Cohab',
 'Várzea',
 'Torrões',
 'Iputinga',
 'Curado',
 'San Martin',
 'Ipsep',
 'Passarinho',
 'Dois Irmãos',
 'Jaqueira',
 'Jardim São Paulo',
 'Areias',
 'Sancho',
 'Barro',
 'Estância',
 'Santana',
 'Tejipió',
 'Zumbi',
 'Cordeiro',
```

```
rio_neighborhood_list = []
for neighborhood in range(0,len(rio_geodata["features"])):
    rio_neighborhood_list.append((rio_geodata["features"][neighborhood]["properties"]['NOME']))
rio_neighborhood_list
```

```
['Paquetá                      ',
 'Freguesia (Ilha)             ',
 'Bancários                    ',
 'Galeão                       ',
 'Tauá                         ',
 'Portuguesa                   ',
 'Moneró                       ',
 'Vigário Geral                ',
 'Cocotá                       ',
 'Jardim América               ',
 'Jardim Carioca               ',
 'Pavuna                       ',
 'Cordovil                     ',
 'Jardim Guanabara             ',
 'Parada de Lucas              ',
 'Parque Colúmbia              ',
 'Praia da Bandeira            ',
```

Let´s create a Dataframe with the Neighborhoods and the columns of Latitude and Longitude, for each city. Note that in the city of Recife, I changed a Neighborhood name, because in the geopy the name is different. In the city of Rio de Janeiro, I remove the spaces of right side from the Neighborhood column. In both cases, i applied the unidecode to remove Portuguese accents and other specific characters. I add two new columns, to insert the latitude and longitude of each neighborhood.

```python
recife_df = pd.DataFrame(columns=["Neighborhoods"])
recife_df["Neighborhoods"] = recife_neighborhood_list
recife_df["Neighborhoods"] = recife_df["Neighborhoods"].apply(unidecode)
recife_df["Latitude"] = ""
recife_df["Longitude"] = ""
recife_df["Neighborhoods"] = recife_df["Neighborhoods"].str.replace('Terezinha','Teresinha')
recife_df
```

	Neighborhoods	Latitude	Longitude
0	Cidade Universitaria		
1	Soledade		
2	Engenho do Meio		
3	Cacote		
4	Cohab		
...	...	...	...
89	Jiquia		
90	Afogados		
91	Apipucos		
92	Guabiraba		
93	Corrego do Jenipapo		

94 rows × 3 columns

```
rio_df = pd.DataFrame(columns=["Neighborhoods"])
rio_df["Neighborhoods"] = rio_neighborhood_list
rio_df["Neighborhoods"] = rio_df["Neighborhoods"].apply(unidecode)
rio_df["Latitude"] = ""
rio_df["Longitude"] = ""
rio_df["Neighborhoods"] = rio_df["Neighborhoods"].str.strip()
rio_df
```

	Neighborhoods	Latitude	Longitude
0	Paqueta		
1	Freguesia (Ilha)		
2	Bancarios		
3	Galeao		
4	Taua		
...	...	...	...
158	Campo Grande		
159	Bangu		
160	Gericino		
161	Jabour		
162	Vila Kennedy		

163 rows × 3 columns

So we can see the Rio has 163 neighborhoods and Recife has 94. Now we have to collect the Latitude and the Longitude of each neighborhood of each city. Notice that this is necessary because the geojson only have the coordinates of the neighborhood borders and we need the central coordinate. To do that I use the Geopy library.

```python
iteration = 0
for recife_neighborhoods in recife_df["Neighborhoods"]:
    address = str(recife_neighborhoods) + " Recife"
    geolocator = Nominatim(user_agent="recife_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    recife_df["Latitude"][iteration] = latitude
    recife_df["Longitude"][iteration] = longitude
    iteration +=1
    print('The geographical coordinate of {} are {}, {}.'.format(recife_neighborhoods, latitude, longitude))
    print (iteration)
recife_df
```

0	Cidade Universitaria	-8.05441	-34.9516
1	Soledade	-8.05598	-34.8906
2	Engenho do Meio	-8.05659	-34.9424
3	Cacote	-8.10106	-34.9327
4	Cohab	-8.12411	-34.9488
...	...	...	...
89	Jiquia	-8.08752	-34.9245
90	Afogados	-8.07557	-34.9078
91	Apipucos	-8.01902	-34.9381
92	Guabiraba	-7.95329	-34.9537
93	Corrego do Jenipapo	-8.00131	-34.9372

94 rows × 3 columns

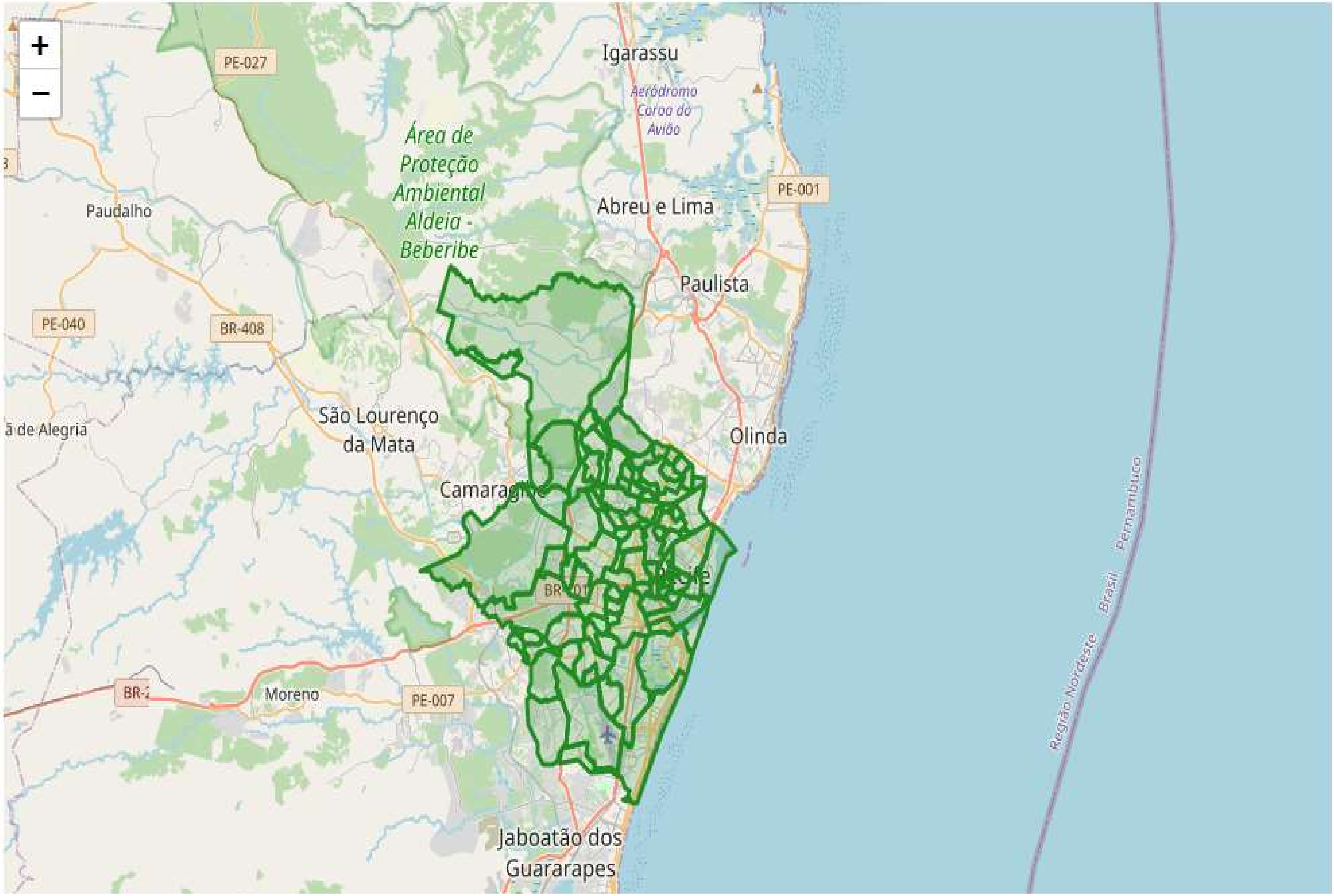
```python
iteration = 0
for rio_neighborhoods in rio_df["Neighborhoods"]:
    address = str(rio_neighborhoods) + " RJ"
    geolocator = Nominatim(user_agent="rio_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    rio_df["Latitude"][iteration] = latitude
    rio_df["Longitude"][iteration] = longitude
    iteration +=1
    print('The geographical coordinate of {} are {}, {}.'.format(rio_neighborhoods, latitude, longitude))
    print (iteration)
rio_df
```

0	Paqueta	-22.7589	-43.1092
1	Freguesia (Ilha)	-22.7851	-43.1695
2	Bancarios	-22.7918	-43.181
3	Galeao	-22.8075	-43.2355
4	Taua	-22.7977	-43.1867
...	...	...	...
158	Campo Grande	-22.903	-43.5591
159	Bangu	-22.8753	-43.4649
160	Gericino	-22.8418	-43.4774
161	Jabour	-22.8808	-43.4932
162	Vila Kennedy	-22.8557	-43.49

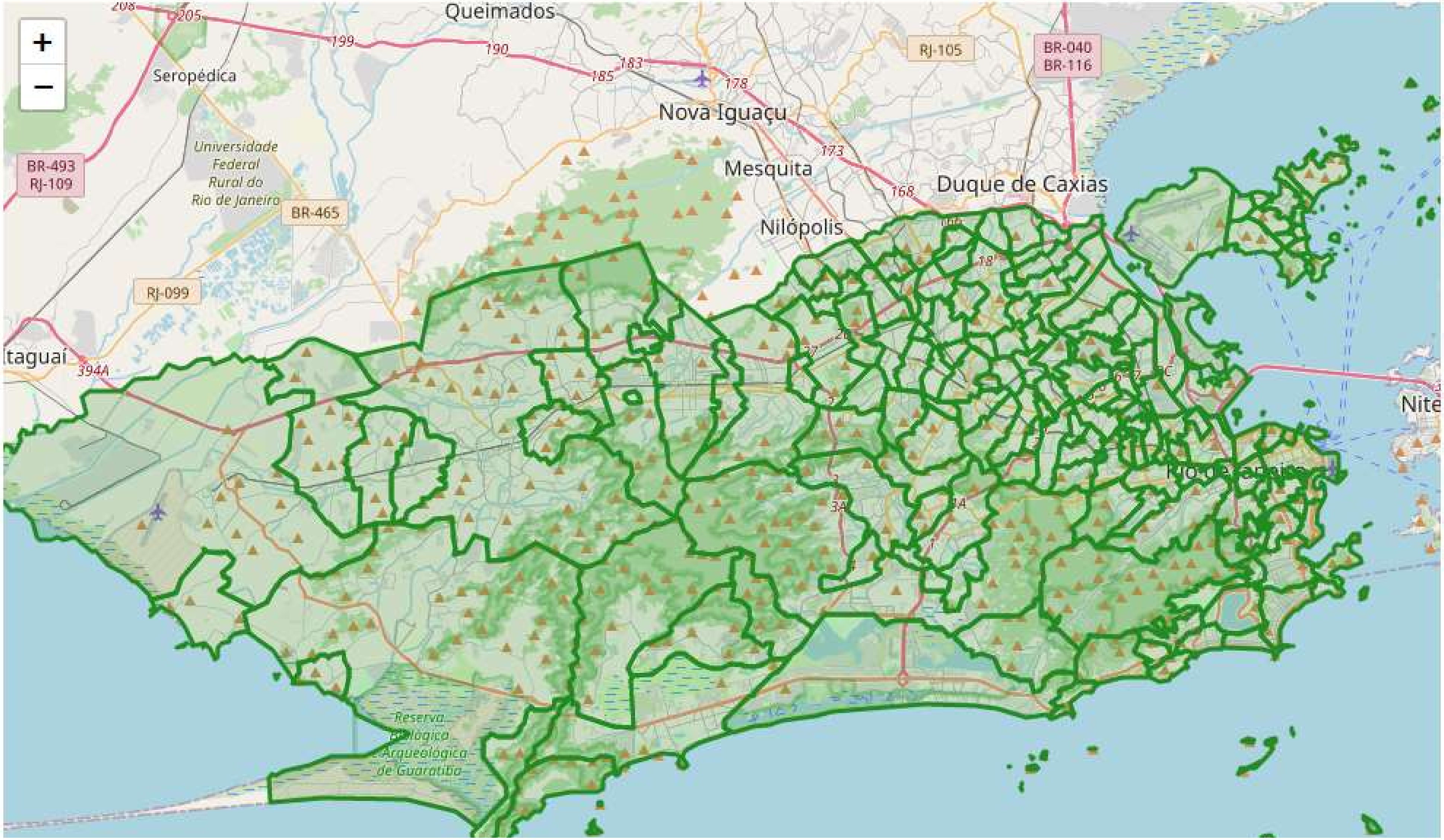
163 rows × 3 columns

Now let´s create a map that shows us the neighborhoods from each city, using the Folium library.

```
recife_map = folium.Map(location=[-8.05428,-34.8813], zoom_start=11, tiles='OpenStreetMap')
color = {'fillColor': '#228B22', 'color': '#228B22'}
folium.GeoJson(recife_geodata,name='geojson',style_function=lambda x:color).add_to(recife_map)
recife_map
```



```python
rio_map = folium.Map(location=[-22.9000, -43.451354], zoom_start=11, tiles='OpenStreetMap')
color = {'fillColor': '#228B22', 'color': '#228B22'}
folium.GeoJson(rio_geodata,name='geojson',style_function=lambda x:color).add_to(rio_map)
rio_map
```



In order to obtain the data about the venues of the cities I must enter the credentials of the Foursquare API.

```
LIMIT = 500 # limit of number of venues returned by Foursquare API
radius = 500 # define radius

CLIENT_ID = 'put your id here' # your Foursquare ID
CLIENT_SECRET = 'put your secret here' # your Foursquare Secret
VERSION = '20180604'
print('My credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

Below are the functions to return data from the Foursquare API

```python
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']

def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&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,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

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

And now it´s time to get the data!

```
recife_df_venues = getNearbyVenues(names=recife_df['Neighborhoods'],
                                   latitudes=recife_df['Latitude'],
                                   longitudes=recife_df['Longitude']
                                  )
```

```
Cidade Universitaria
Soledade
Engenho do Meio
Cacote
Cohab
Varzea
Torroes
Iputinga
Curado
San Martin
Ipsep
Passarinho
Dois Irmaos
```

```
print(recife_df_venues.shape)
recife_df_venues.head()
```

```
(2286, 7)
```

	Neighborhoods	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Cidade Universitaria	-8.05441	-34.951618	Museu de Oceanografia	-8.054785	-34.953296	Museum
1	Cidade Universitaria	-8.05441	-34.951618	Concha Acústica UFPE	-8.054166	-34.953298	Music Venue
2	Cidade Universitaria	-8.05441	-34.951618	Teatro da UFPE	-8.052367	-34.950836	Theater
3	Cidade Universitaria	-8.05441	-34.951618	Bigode	-8.053711	-34.948066	Food
4	Cidade Universitaria	-8.05441	-34.951618	Natação	-8.053907	-34.948501	Water Park

```
rio_df_venues = getNearbyVenues(names=rio_df['Neighborhoods'],
                                   latitudes=rio_df['Latitude'],
                                   longitudes=rio_df['Longitude']
                                  )
```

```
Barra da Tijuca
Leblon
Ipanema
Sao Conrado
Rocinha
Pedra de Guaratiba
Recreio dos Bandeirantes
Vidigal
Joa
Barra de Guaratiba
Grumari
Caju
Deodoro
Lapa
Campo Grande
Bangu
Gericino
```

```python
rio_df_venues.groupby('Neighborhoods').count()
```

	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhoods						
Abolicao	31	31	31	31	31	31
Acari	5	5	5	5	5	5
Agua Santa	3	3	3	3	3	3
Alto da Boa Vista	2	2	2	2	2	2
Anchieta	14	14	14	14	14	14
...	...	...	...	...	...	...
Vila Militar	10	10	10	10	10	10
Vila Valqueire	14	14	14	14	14	14
Vila da Penha	49	49	49	49	49	49
Vista Alegre	25	25	25	25	25	25
Zumbi	15	15	15	15	15	15

160 rows × 6 columns

```
recife_df_venues.groupby('Neighborhoods').count()
```

Neighborhoods	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Aflitos	100	100	100	100	100	100
Afogados	27	27	27	27	27	27
Agua Fria	24	24	24	24	24	24
Alto Jose Bonifacio	9	9	9	9	9	9
Alto Jose do Pinho	19	19	19	19	19	19
...	...	...	...	...	...	...
Torreao	15	15	15	15	15	15
Torroes	16	16	16	16	16	16
Toto	8	8	8	8	8	8
Vasco da Gama	14	14	14	14	14	14
Zumbi	29	29	29	29	29	29

91 rows × 6 columns

```
print('There are {} uniques categories in Recife and there are {} uniques categories in Rio de Janeiro.'
      .format(len(recife_df_venues['Venue Category'].unique()),len(rio_df_venues['Venue Category'].unique())))
```

```
There are 248 uniques categories in Recife and there are 294 uniques categories in Rio de Janeiro.
```

```python
print("The Recife´s Dataframe is",recife_df_venues.shape)
print("The Rio de Janeiro´s Dataframe is",rio_df_venues.shape)
```

```
The Recife´s Dataframe is (2286, 7)
The Rio de Janeiro´s Dataframe is (3124, 7)
```

```python
recife_restaurants = recife_df_venues[recife_df_venues['Venue Category'].str.contains('Restaurant')]
recife_restaurants
```

	Neighborhoods	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
8	Cidade Universitaria	-8.054410	-34.951618	ASSIF-PE - Restaurante Mania	-8.058218	-34.951586	Restaurant
23	Soledade	-8.055980	-34.890561	Janete Self-Service	-8.058208	-34.891276	Brazilian Restaurant
24	Soledade	-8.055980	-34.890561	Odisan Temakeria - UNICAP	-8.054864	-34.886963	Japanese Restaurant
25	Soledade	-8.055980	-34.890561	Coni Móvel	-8.056739	-34.892942	Japanese Restaurant
26	Soledade	-8.055980	-34.890561	Restaurante O Vegetariano	-8.052293	-34.889007	Vegetarian / Vegan Restaurant
...	...	...	...	...	...	...	...
2251	Afogados	-8.075565	-34.907807	Bar do Bernardo	-8.075932	-34.908116	Brazilian Restaurant
2263	Afogados	-8.075565	-34.907807	Cantinho da Amara	-8.075431	-34.905428	Restaurant
2265	Afogados	-8.075565	-34.907807	Kung Fu Chinês	-8.079341	-34.905956	Chinese Restaurant
2277	Corrego do Jenipapo	-8.001306	-34.937217	Recanto do Lau	-8.001523	-34.935991	Brazilian Restaurant
2281	Corrego do Jenipapo	-8.001306	-34.937217	Dim Sum Sam	-8.003390	-34.935932	Dim Sum Restaurant

402 rows × 7 columns

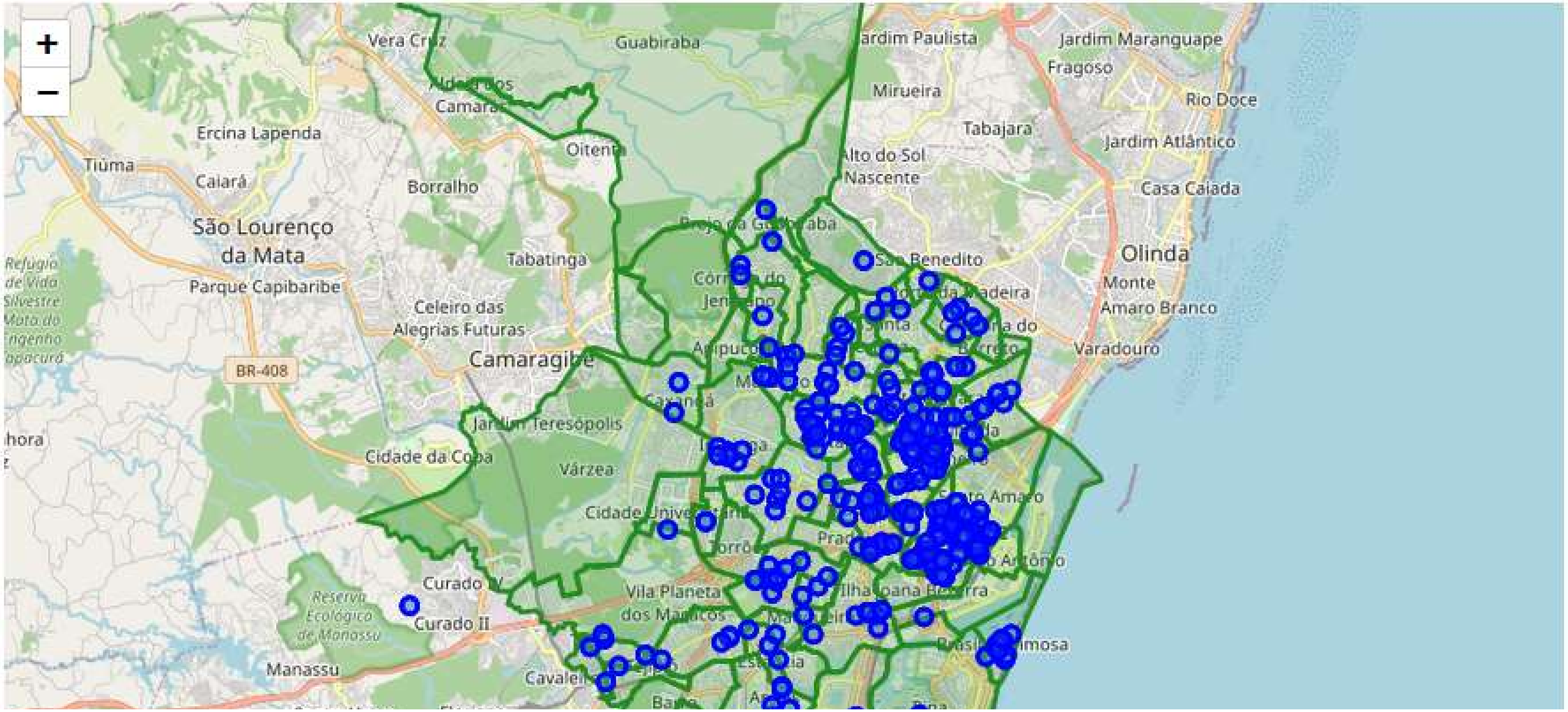
```
rio_restaurants = rio_df_venues[rio_df_venues['Venue Category'].str.contains('Restaurant')]
rio_restaurants
```

	Neighborhoods	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
4	Paqueta	-22.758926	-43.109199	Tia Leleta Bar	-22.760765	-43.108334	Brazilian Restaurant
7	Paqueta	-22.758926	-43.109199	Zeca's Restaurante	-22.761253	-43.107708	Brazilian Restaurant
24	Bancarios	-22.791759	-43.180966	Rei do Bolinho de Bacalhau	-22.795611	-43.183378	Seafood Restaurant
30	Bancarios	-22.791759	-43.180966	Valão Grill	-22.789675	-43.182052	Comfort Food Restaurant
44	Galeao	-22.807506	-43.235521	Fruit	-22.811472	-43.237746	American Restaurant
...	...	...	...	...	...	...	...
3090	Bangu	-22.875305	-43.464880	Koni Store	-22.878502	-43.468219	Japanese Restaurant
3093	Bangu	-22.875305	-43.464880	McDonald's	-22.878487	-43.468203	Fast Food Restaurant
3095	Bangu	-22.875305	-43.464880	Bob's	-22.878488	-43.468179	Fast Food Restaurant
3100	Bangu	-22.875305	-43.464880	Vivenda do Camarão	-22.878446	-43.468133	Seafood Restaurant
3115	Vila Kennedy	-22.855678	-43.490030	Habib's	-22.854458	-43.486758	Fast Food Restaurant

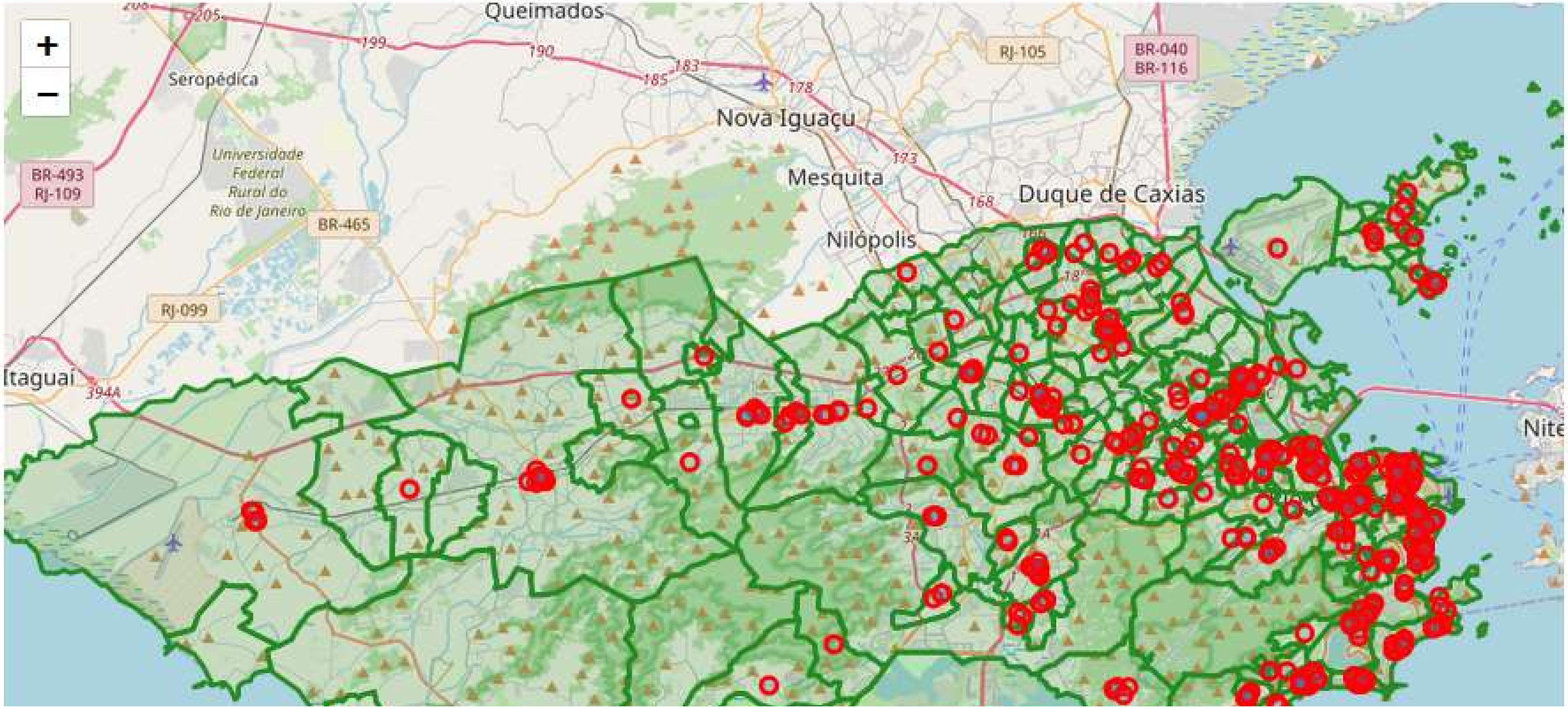
542 rows × 7 columns

In the map bellow, all the restaurants in the city is shown in their correct places

```python
for lat, lng, restaurants in zip(recife_restaurants['Venue Latitude'], recife_restaurants['Venue Longitude'],
                                 recife_restaurants['Venue'].apply(unidecode)):
    label = '{}'.format(restaurants)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.2,
        parse_html=False).add_to(recife_map)

recife_map
```

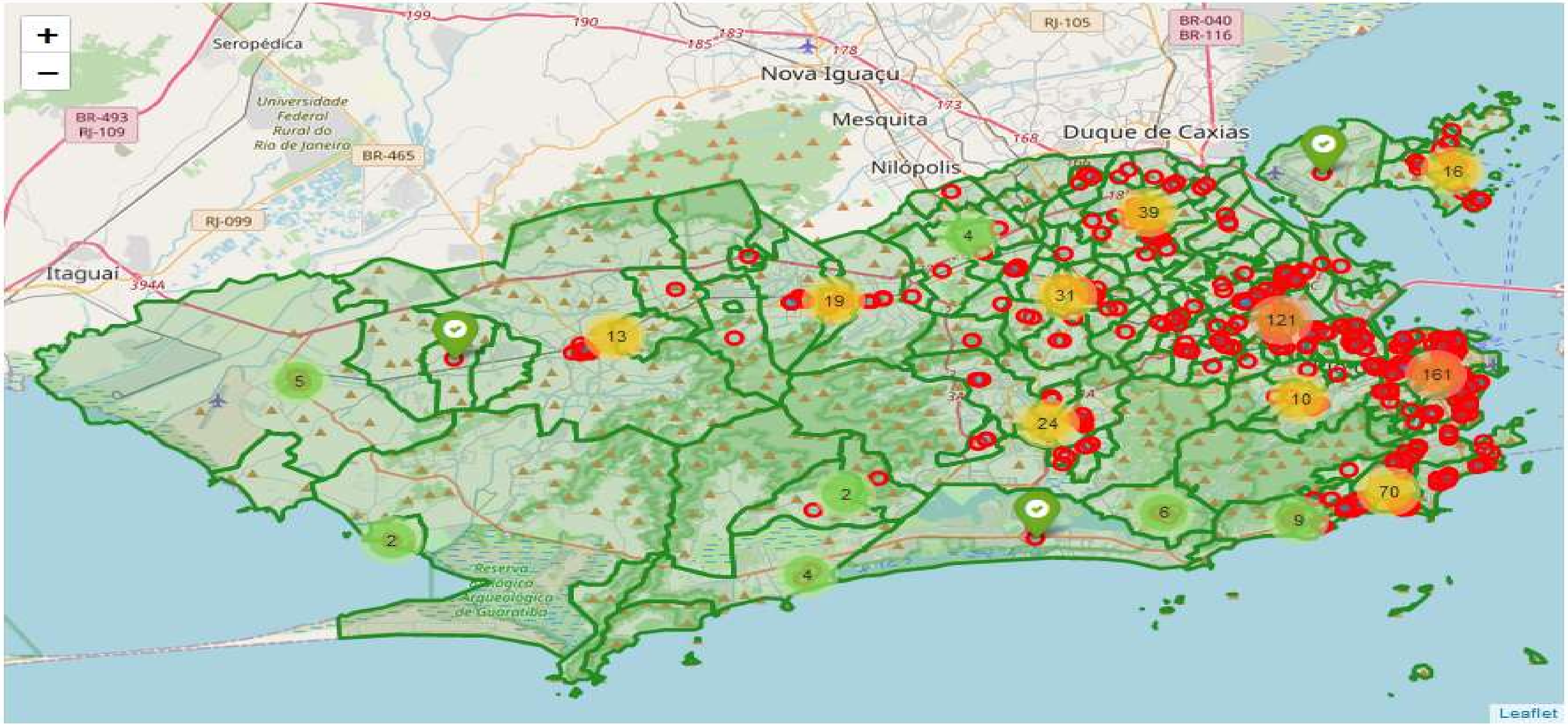


```python
for lat, lng, restaurants in zip(rio_restaurants['Venue Latitude'], rio_restaurants['Venue Longitude'], rio_res
    label = '{}'.format(restaurants)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.2,
        parse_html=False).add_to(rio_map)

rio_map
```

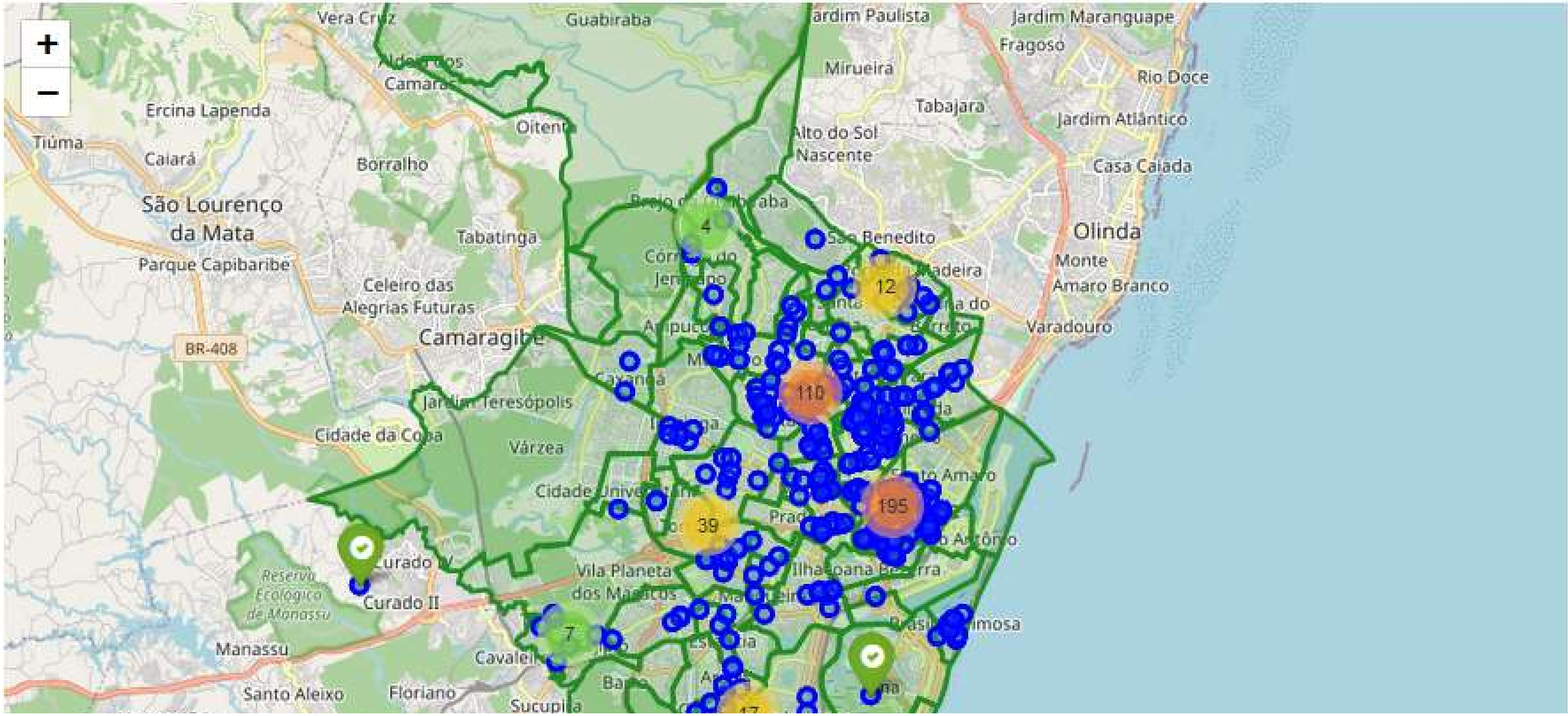


In the following maps, the cluster technique is applied to show which city areas has most restaurants.

```
restaurants = plugins.MarkerCluster().add_to(rio_map)
for lat, lng, rests in zip(rio_restaurants['Venue Latitude'], rio_restaurants['Venue Longitude'],rio_restaurants
    label = '{}'.format(rests)
    label = folium.Popup(label, parse_html=True)
    folium.Marker(
        location=[lat,lng],
        popup=label,
        icon=folium.Icon(color='green', icon='ok-sign'),
    ).add_to(restaurants)

rio_map
```



```
restaurants = plugins.MarkerCluster().add_to(recife_map)

for lat, lng, rests, in zip(recife_restaurants['Venue Latitude'], recife_restaurants['Venue Longitude'],
                            recife_restaurants['Venue'].apply(unidecode)):
    label = '{}'.format(rests)
    label = folium.Popup(label, parse_html=True)
    folium.Marker(
        location=[lat,lng],
        popup=label,
        icon=folium.Icon(color='green', icon='ok-sign'),
    ).add_to(restaurants)



recife_map
```



Now let´s count how many restaurant categories exists in each city.

```
recife_restaurants_categories = recife_restaurants.groupby('Venue Category').count().sort_values(
    by='Neighborhoods',ascending=False)
recife_restaurants_categories.rename(columns = {'Neighborhoods':'Qty'}, inplace = True)
recife_restaurants_categories.drop(recife_restaurants_categories.columns[[1,2,3,4,5]], axis=1, inplace=True)
recife_restaurants_categories = recife_restaurants_categories.reset_index()
recife_restaurants_categories["%"] = (recife_restaurants_categories['Qty'] /
                                       recife_restaurants_categories['Qty'].sum())
recife_restaurants_categories
```

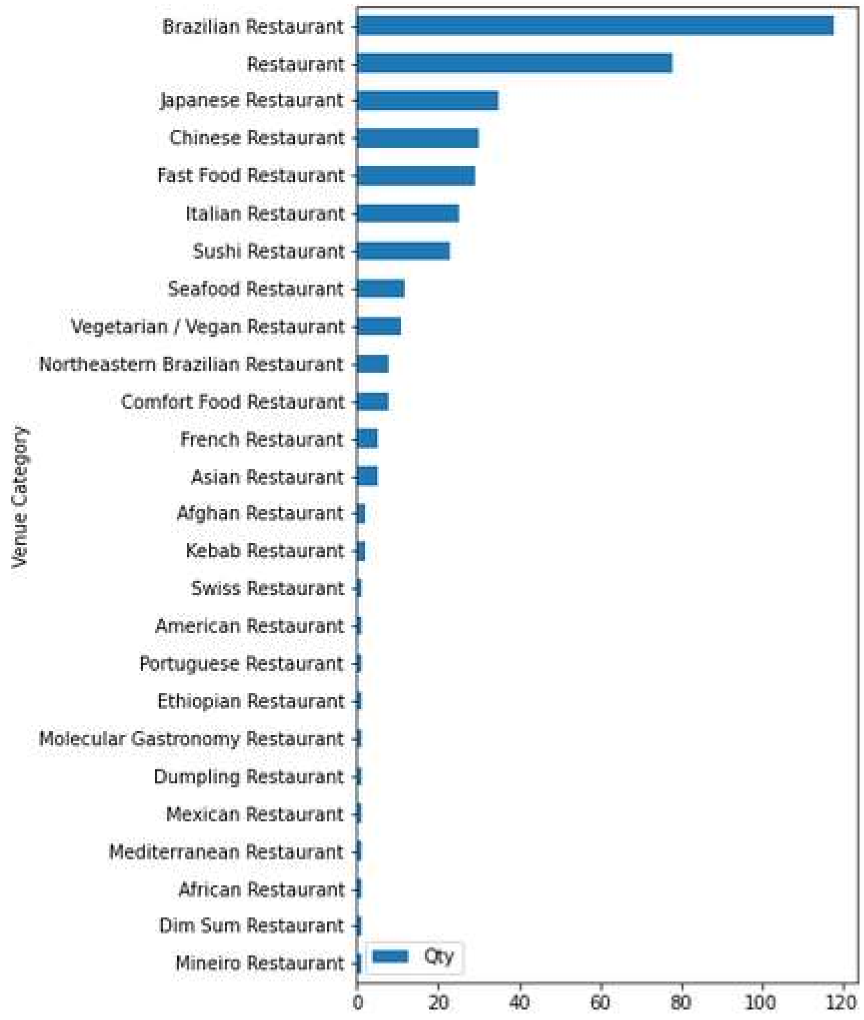
	Venue Category	Qty	%
0	Brazilian Restaurant	118	0.293532
1	Restaurant	78	0.194030
2	Japanese Restaurant	35	0.087065
3	Chinese Restaurant	30	0.074627
4	Fast Food Restaurant	29	0.072139
5	Italian Restaurant	25	0.062189
6	Sushi Restaurant	23	0.057214
7	Seafood Restaurant	12	0.029851
8	Vegetarian / Vegan Restaurant	11	0.027363
9	Northeastern Brazilian Restaurant	8	0.019900
10	Comfort Food Restaurant	8	0.019900
11	French Restaurant	5	0.012438
12	Asian Restaurant	5	0.012438
13	Afghan Restaurant	2	0.004975
14	Kebab Restaurant	2	0.004975
15	Swiss Restaurant	1	0.002488
16	American Restaurant	1	0.002488
17	Portuguese Restaurant	1	0.002488
18	Ethiopian Restaurant	1	0.002488
19	Molecular Gastronomy Restaurant	1	0.002488
20	Dumpling Restaurant	1	0.002488
21	Mexican Restaurant	1	0.002488
22	Mediterranean Restaurant	1	0.002488
23	African Restaurant	1	0.002488
24	Dim Sum Restaurant	1	0.002488

```
rio_restaurants_categories = rio_restaurants.groupby('Venue Category').count().sort_values(
    by='Neighborhoods',ascending=False)
rio_restaurants_categories.rename(columns = {'Neighborhoods':'Qty'}, inplace = True)
rio_restaurants_categories.drop(rio_restaurants_categories.columns[[1,2,3,4,5]], axis=1, inplace=True)
rio_restaurants_categories = rio_restaurants_categories.reset_index()
rio_restaurants_categories["%"] = rio_restaurants_categories['Qty'] / rio_restaurants_categories['Qty'].sum()
rio_restaurants_categories
```

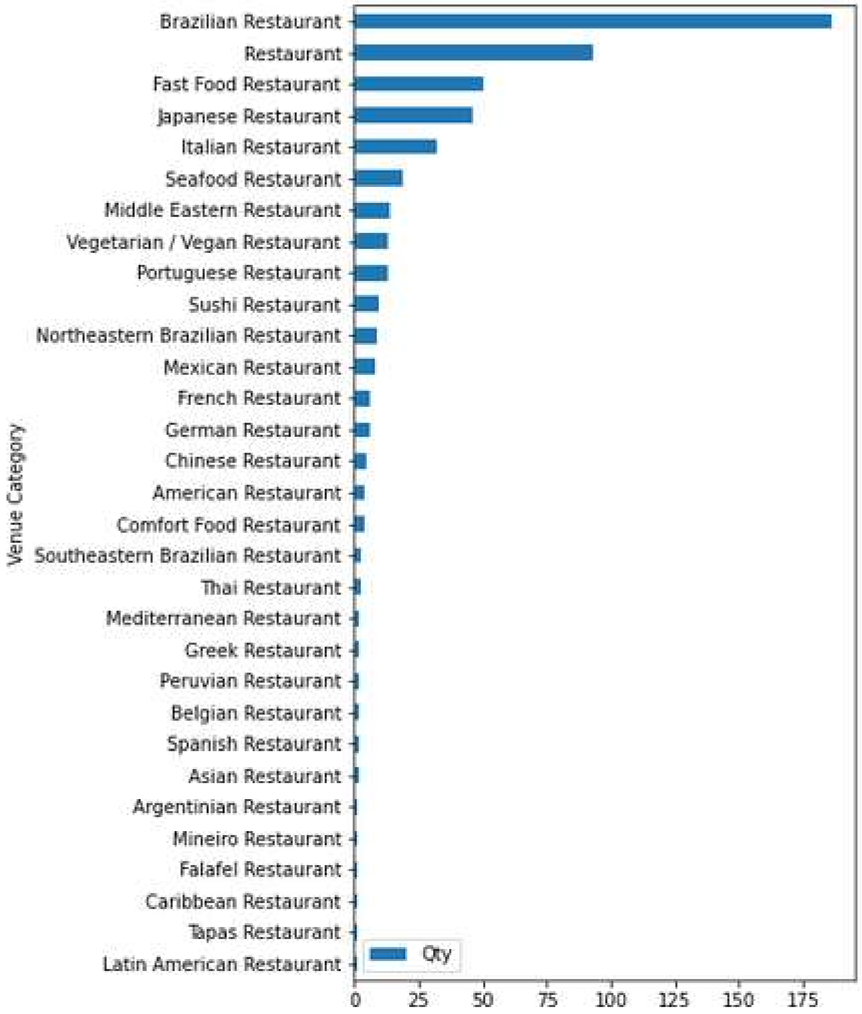
	Venue Category	Qty	%
0	Brazilian Restaurant	186	0.343173
1	Restaurant	93	0.171587
2	Fast Food Restaurant	50	0.092251
3	Japanese Restaurant	46	0.084871
4	Italian Restaurant	32	0.059041
5	Seafood Restaurant	19	0.035055
6	Middle Eastern Restaurant	14	0.025830
7	Vegetarian / Vegan Restaurant	13	0.023985
8	Portuguese Restaurant	13	0.023985
9	Sushi Restaurant	10	0.018450
10	Northeastern Brazilian Restaurant	9	0.016605
11	Mexican Restaurant	8	0.014760
12	French Restaurant	6	0.011070
13	German Restaurant	6	0.011070
14	Chinese Restaurant	5	0.009225
15	American Restaurant	4	0.007380
16	Comfort Food Restaurant	4	0.007380
17	Southeastern Brazilian Restaurant	3	0.005535
18	Thai Restaurant	3	0.005535
19	Mediterranean Restaurant	2	0.003690
20	Greek Restaurant	2	0.003690
21	Peruvian Restaurant	2	0.003690
22	Belgian Restaurant	2	0.003690
23	Spanish Restaurant	2	0.003690
24	Asian Restaurant	2	0.003690

In the following barplots, is shown the frequency of each type of restaurant.

```
recife_plot = recife_restaurants_categories.plot(x='Venue Category', y='Qty',kind='barh', figsize=(5,10))
recife_plot.invert_yaxis()
```



```python
rio_plot = rio_restaurants_categories.plot(x='Venue Category', y='Qty',kind='barh', figsize=(5,10))
rio_plot.invert_yaxis()
```



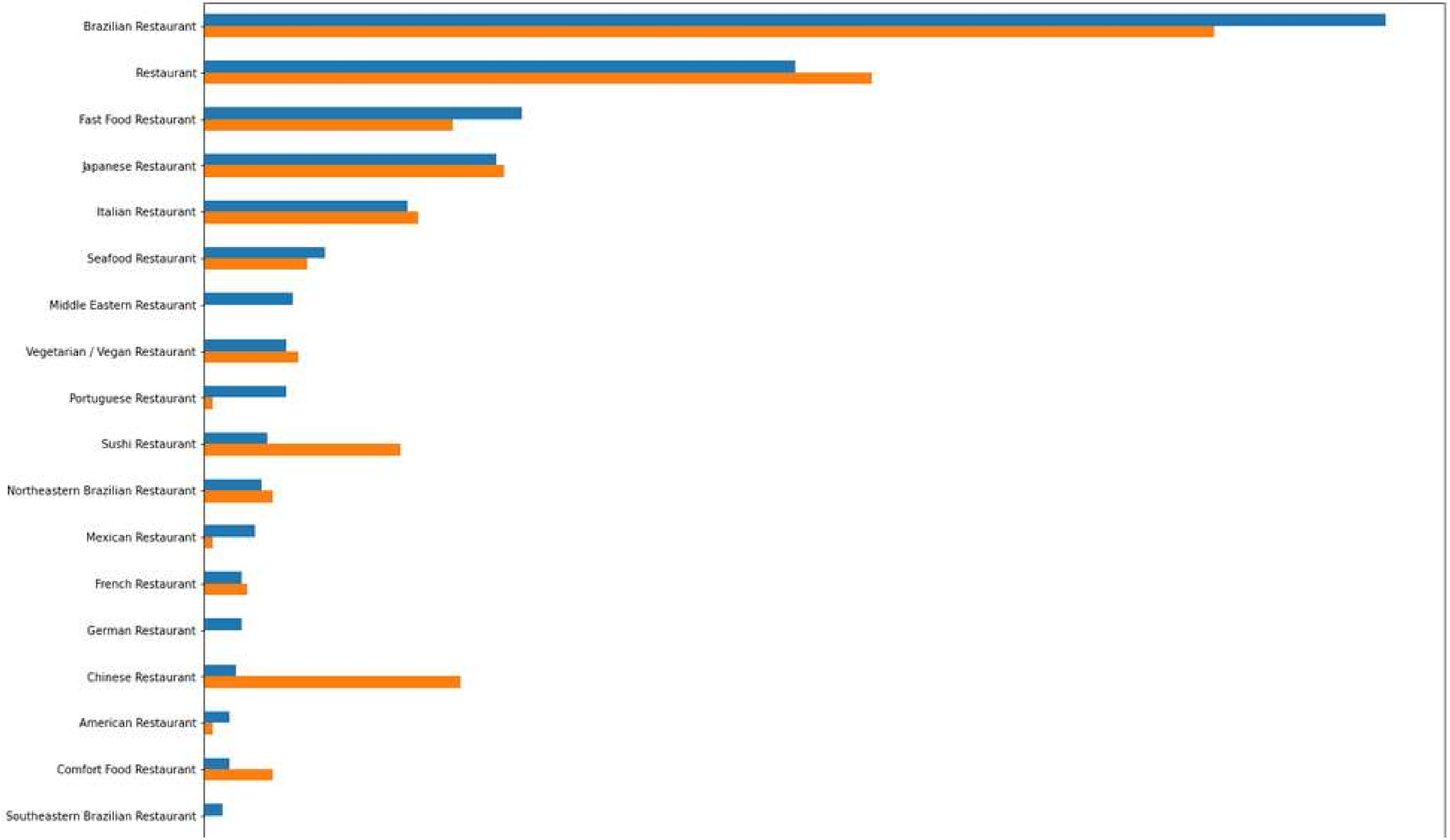
In order to compare the frequencies, we have to gather two data in a single dataframe.

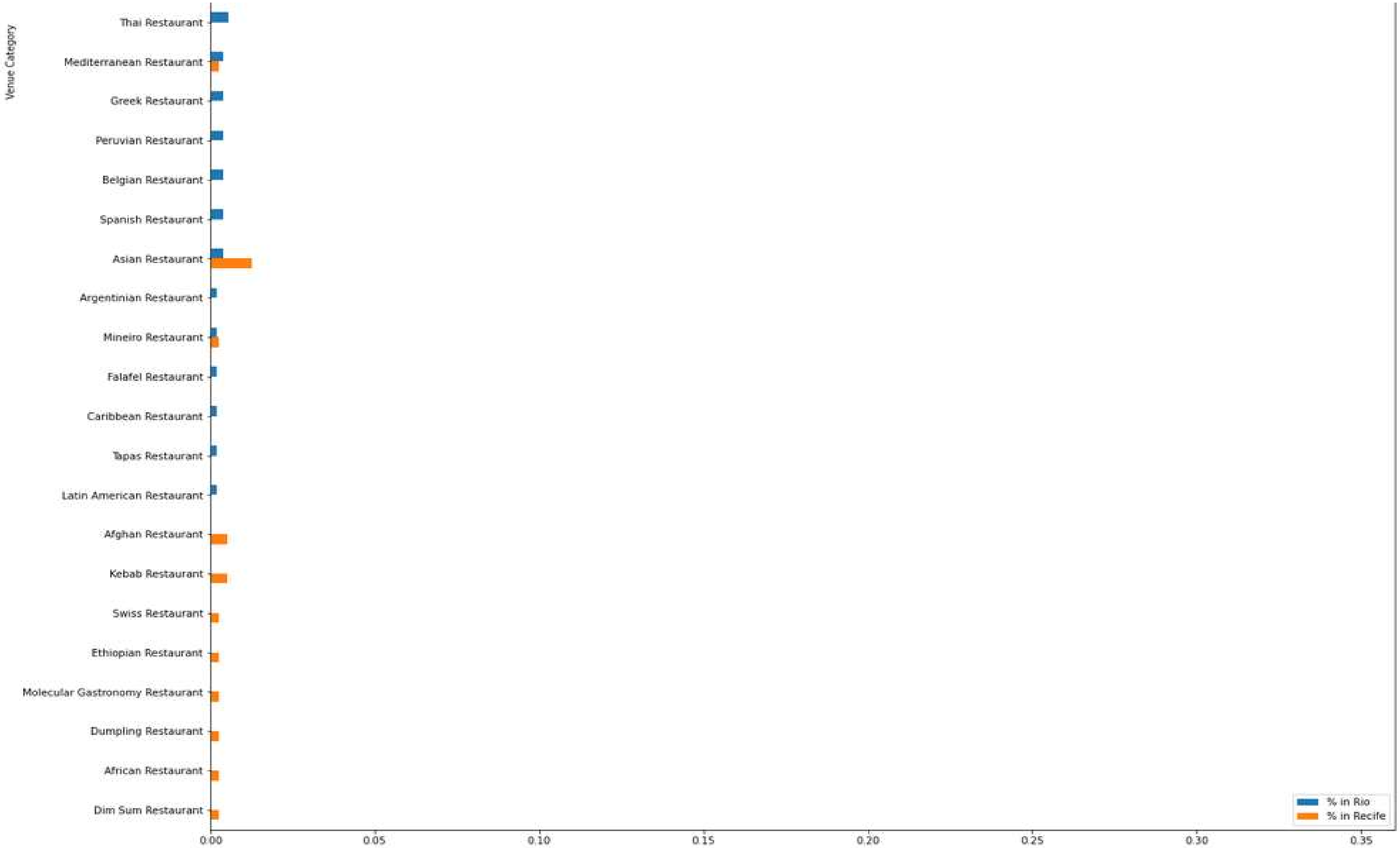
```
qty_categories_in_both_cities = pd.merge(rio_restaurants_categories,
                                         recife_restaurants_categories, how='outer', on=['Venue Category'])
qty_categories_in_both_cities.rename(columns = {'%_x':'% in Rio', '%_y':'% in Recife'}, inplace = True)
qty_categories_in_both_cities
```

	Venue Category	Qty_x	% in Rio	Qty_y	% in Recife
0	Brazilian Restaurant	186.0	0.343173	118.0	0.293532
1	Restaurant	93.0	0.171587	78.0	0.194030
2	Fast Food Restaurant	50.0	0.092251	29.0	0.072139
3	Japanese Restaurant	46.0	0.084871	35.0	0.087065
4	Italian Restaurant	32.0	0.059041	25.0	0.062189
5	Seafood Restaurant	19.0	0.035055	12.0	0.029851
6	Middle Eastern Restaurant	14.0	0.025830	NaN	NaN
7	Vegetarian / Vegan Restaurant	13.0	0.023985	11.0	0.027363
8	Portuguese Restaurant	13.0	0.023985	1.0	0.002488
9	Sushi Restaurant	10.0	0.018450	23.0	0.057214
10	Northeastern Brazilian Restaurant	9.0	0.016605	8.0	0.019900
11	Mexican Restaurant	8.0	0.014760	1.0	0.002488
12	French Restaurant	6.0	0.011070	5.0	0.012438
13	German Restaurant	6.0	0.011070	NaN	NaN
14	Chinese Restaurant	5.0	0.009225	30.0	0.074627
15	American Restaurant	4.0	0.007380	1.0	0.002488
16	Comfort Food Restaurant	4.0	0.007380	8.0	0.019900
17	Southeastern Brazilian Restaurant	3.0	0.005535	NaN	NaN
18	Thai Restaurant	3.0	0.005535	NaN	NaN
19	Mediterranean Restaurant	2.0	0.003690	1.0	0.002488
20	Greek Restaurant	2.0	0.003690	NaN	NaN
21	Peruvian Restaurant	2.0	0.003690	NaN	NaN
22	Belgian Restaurant	2.0	0.003690	NaN	NaN
23	Spanish Restaurant	2.0	0.003690	NaN	NaN
24	Asian Restaurant	2.0	0.003690	5.0	0.012438
25	Argentinian Restaurant	1.0	0.001845	NaN	NaN
26	Mineiro Restaurant	1.0	0.001845	1.0	0.002488
27	Falafel Restaurant	1.0	0.001845	NaN	NaN

And show a barplot about both cities together.

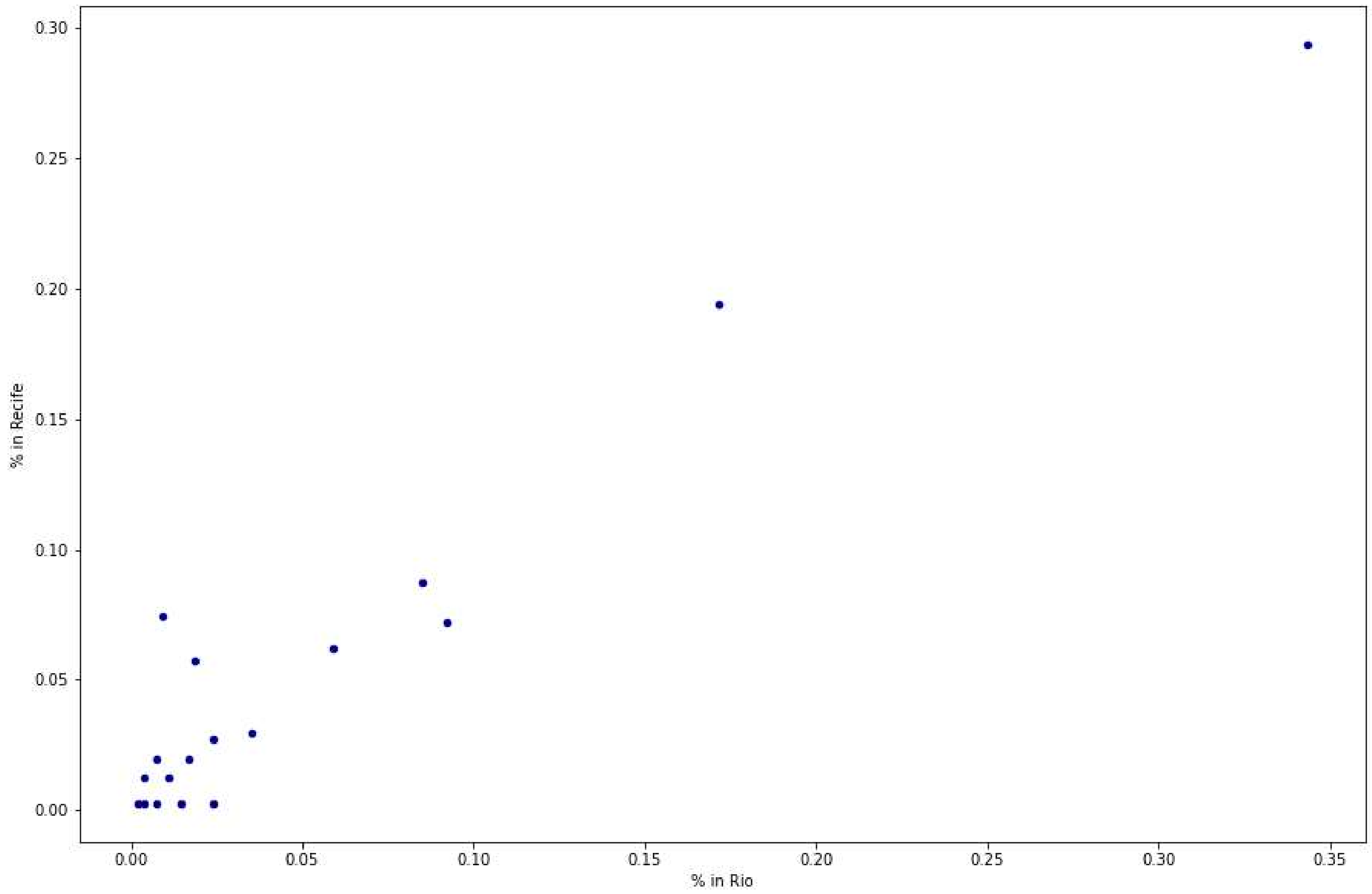
```
qty_categories_in_both_cities.drop(qty_categories_in_both_cities.columns[[1,3]], axis=1, inplace=True)
qty_categories_in_both_cities_plot = qty_categories_in_both_cities.plot.barh(x='Venue Category',figsize=(20,30))
qty_categories_in_both_cities_plot.invert_yaxis()
```




Venue Category
Thai Restaurant
Mediterranean Restaurant
Greek Restaurant
Peruvian Restaurant
Belgian Restaurant
Spanish Restaurant
Asian Restaurant
Argentinian Restaurant
Mineiro Restaurant
Falafel Restaurant
Caribbean Restaurant
Tapas Restaurant
Latin American Restaurant
Afghan Restaurant
Kebab Restaurant
Swiss Restaurant
Ethiopian Restaurant
Molecular Gastronomy Restaurant
Dumpling Restaurant
African Restaurant
Dim Sum Restaurant
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
% in Rio
% in Recife

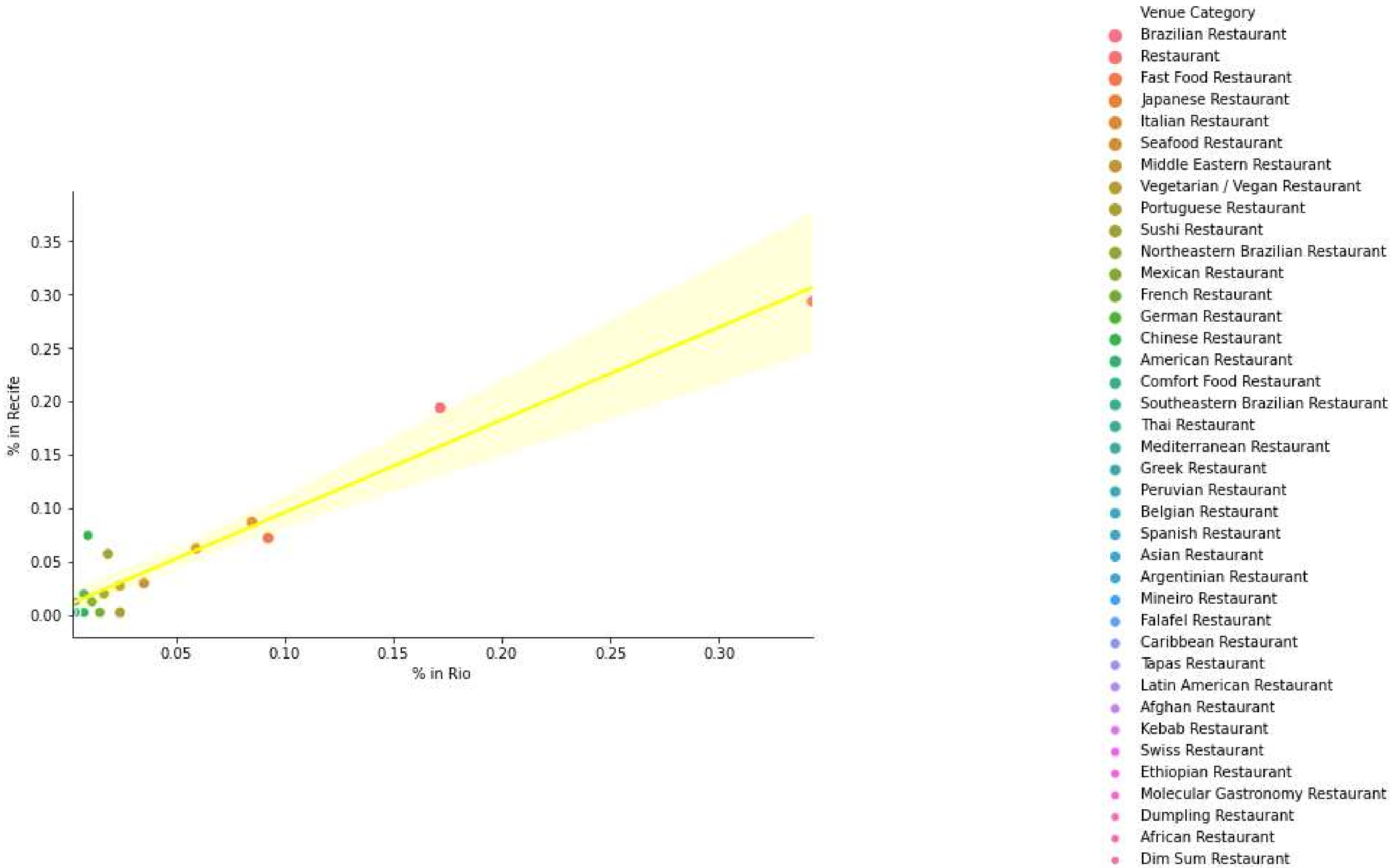
To help the visualization, let´s make a scatterplot

```
ax1 = qty_categories_in_both_cities.plot.scatter(x='% in Rio', y='% in Recife',c='DarkBlue', figsize=(15,10))
```



Now we´ll verify if there is a correlation between the restaurant categories of each city.

```
ax2 = sns.relplot(data=qty_categories_in_both_cities,x='% in Rio', y='% in Recife',
                  hue="Venue Category", size = "Venue Category")
ax2 = sns.regplot(y='% in Recife', x='% in Rio', data= qty_categories_in_both_cities,
                  color='k', scatter_kws={"alpha" : 0.0}, line_kws={"color": "yellow"})
ax2.figure.set_size_inches(15,5)
```



Clearly, there is a strong correlation. Let´s verify the correlation results.

```
corr = qty_categories_in_both_cities.corr(method ='pearson')
corr.style.background_gradient(cmap='coolwarm')
```

	% in Rio	% in Recife
% in Rio	1.000000	0.960114
% in Recife	0.960114	1.000000

4. Results

The results obtained are:

a) Recife has 94 neighborhoods, 2286 venues and 248 unique categories. Related to restaurants, were found a total of 402 of 26 categories of different types in the city.

b) Rio de Janeiro has 163 neighborhoods, 3124 venues and 294 unique categories. Related to restaurants, were found a total of 542 of 31 different types in the city.

c) The correlation between the restaurant categories of the two cities is very strong (0.960114).

5. Discussion

Recife and Rio de Janeiro have some similarities. Both of them are state capitals, are coastal and touristic places. Recife is in the Brazil´s northeast and has 1.6 million of population. In contrast, Rio is more populated with 6.7 inhabitants and located in the southeast of the country. The number of places as well as the number of restaurants follows the size of the population. When we analyze the origin of tourists, we have the two tables bellow:

RECIFE	2014	2015	2016	2017	2018
Country of residence			(%)		
Argentina	8,8	12,8	23,1	17,6	19,8
Germany	7,3	9,9	8,8	5,7	13,6
U.S.A	22,6	12,8	13	13,5	12,2
Portugal	4,6	7,5	7,2	5,8	6,4
Italy	6,4	7	5,6	8,4	5,3
France	3,6	5	2,9	2,7	5,1
Uruguay	0,7	2,6	0,9	4,6	4,4
Chile	1,8	2,9	3,5	4,9	4
Canada	2,2	1,4	0,9	1,4	3,1
Spain	3,3	4,9	3,1	2,9	2,4

RIO DE JANEIRO	2014	2015	2016	2017	2018
Country of residence			(%)		
Argentina	20,2	17,4	17,7	24,6	27,4
Chile	6	6,9	7,8	10,7	12,4
U.S.A	12,5	13,1	14,2	9,1	9,2
France	6,7	7,8	6,7	6,5	6,3
United Kingdom	6,1	6,5	6,2	6,3	3,9
Germany	4,4	4,3	4	3,8	3,4
Colombia	3,3	2,5	2,6	3	3,2
Paraguay	1,2	1,1	1,5	2,2	3
Italy	3,1	3,5	3,2	2,2	2,3
Uruguay	1,6	2,1	2	2,3	2,3

Argentinians are the most popular tourist in both cities, but in the other positions the result is varied. In spite of that, there is only a single Argentinian restaurant in the data. On the other hand Japanese tourists are not in the list, but the Japanese is the one most present in both cities. We can infer that maybe tourists, when they come abroad, want to taste different spices from those they have in their respective countries. As we can see in the cluster maps, the largest number of restaurants is not near the coastal strip, but in the center of the cities. This is probably because in Brazil, the city´s center is where more people go to work, and thus there is more need to have restaurants nearby. The correlation between the types of restaurants was, in a way, surprising. I can’t glimpse a direct cause, I just imagine that maybe it’s because the population has similar tastes, but I’m not sure about that. In the future I will take a larger sample of cities and analyze if this correlation is maintained.

6. Conclusion

By the end of this project, we saw that distant cities (more than 2.300 km), in the same country, have more similarities than imagined. The places where the restaurants are and the kind of restaurants are more likely to be comparable. Using the great Folium library we can see the data in a way that everybody can understand it. This project took a valuable time, totally necessary to better understand how to use the tools available to data scientists.