

Battle of neighborhoods

part/week 1 and 2 (complete work)

by Miguel Rozsas

note to the fellow colleague examiner: This notebook has the entire assignment, not only part 1 or part 2 as many did. You are reading a HTML version of my notebook. The *live* version is available as a Jupyter Notebook ([jpynb](#)) in my [GitHub](#)

(https://github.com/marozsas/Coursera_Capstone/blob/main/Assignment%20the%20battle%20of%20neighborhoods)

Introduction

In this assignment, I decided to explore the city of São Paulo, Brazil, which is the city where I was born.

Not only because I was born there, but mainly, because I wanted to know What kind of data it is available in a city outside of major countries and international cities.

Another reason I decide to use São Paulo as a case of study is based in a recent discussion I had with a colleague of mine, about a very distinct business type that occurs in Brazil, named "Padaria" which loosely translates as 'Bakery' but not quite.

'*Padarias*' in Brazil are quite different for Bakeries in North America and Europe.

We can be amazed that the main product of a '*Padaria*' are not bread as you can expect.

Of course you find bread in a Brazilian '*Padaria*' but not the variety you see in other Bakeries.

Simply saying, bread are not their focus. For sure they have a few type of cakes and breads, the most basic ones, produced several times during the day, as the fresh baked breads are very appreciated in Brazil and bring people to the Padaria that buys several supplementary products like in a delicatessen in USA and Europe: Ham, salame (peperoni), sliced cheeses, juices (orange mostly), filtered coffee (there are espresso too but filtered coffee is the preferred one), yogurt, milk (in Brazil you buy your dairy products at the bakery, not at the supermarket, generally speaking, at daily basis)

Another distinctive aspect of *Padarias* are they are a place where you make your morning breakfast, not at home as is the norm in USA/Europe. Also, another surprising aspect is *Padarias* act like a low budget restaurant, serving a buffet with a complete meal at midday time.

And because *Padarias* are so important in Brazilian's life style, there are many of them, all over the city, and a question arises: Where is the best place to open a *Padaria* in Zona Leste of São Paulo city ?

Lets see what we can find out.

In [1]:

```
import numpy as np # library for vectorized computation
import pandas as pd # library to process data as dataframes
import requests # library to handle requests
from bs4 import BeautifulSoup #to web scrap
```

Data

The venues, categories and geographic coordinates data comes from ForSquare. Using ForuSquare API we will query FourSquare database using the *near* API entry point that allows us to get results around a place, around a borough. The boroughs names comes from a well reputed news site, globo.com.

Using the boroghs names from globo.com we get the venues details around each borough. This detailed data is used in our analysis and mapping, to answer the question asked: *"Where is the best place to open a Padaria in Zona Leste of São Paulo city ?"*

Methodology

Data about São Paulo Boroghs and Neighborhoods

I've tried to find a geograpfic coordinates database of Borogh and Neighborhood in Brazil to use this information as entry-point in Foursquare, as we did before.

But no, I could't find one publicly available database on the internet. The Brazilian governamental site that should have this data didn't put it for general use in their site. You should fill a request explaining why you want that information. I fill the form but until now I didn't received a answer.

What I was able to find, was a list of Boroghs (in the city of São Paulo there are 5 boroghs and they are named '*Zona*' and their names correspond to cardinal directions: '*Norte, Sul, Leste, Oeste, Centro*' - North, South, East, West, Center) and their respective neighborhoods.

I verified that **Foursquare** API has another entry-point named **near** which I can pass a borogh name and receive the data around that place. Nice ! Lets use **near** with boroghs names, instead using borogh's geographical coordintes.

The list of boroghs and neighborhoods came from a brazilian news site and I will use *beautifulsoup module* to scrap their page.

In [77]:

```
url= 'http://g1.globo.com/sao-paulo/noticia/2013/11/veja-distribuicao-oficial-do-s-bairros-nas-cinco-regioes-da-cidade.html'
response = requests.get(url)

if response.status_code!=200:
    print (f"\npage {url} returned error status {response.status_code}")
    print ("Aborting.")
    quit ()
else:
    print ("page loaded")
```

page loaded

In [78]:

```
html= BeautifulSoup(response.text, 'html.parser')
```

In [84]:

```
# create the dataframe to host the data
boroghs_sp = pd.DataFrame(columns=('Borough', 'Neighborhood'))
i= 0 # index used to populate the dataframe rows
kind= 0 # flag to indicate if a block is a borough or a neighborhood

# use BeautifulSoup to parse the HTML page
html= BeautifulSoup(response.text, 'html.parser')

# find the div id specified in HTML, and the inner div
# /*[@id="materia-letra"]
div= html.find ('div', id='materia-letra').find ('div')
blocks= div.find_all ('p')

# block[0] has a introductory text
# block[i] has the borough name
# block[i+1] has a neighborhood list

# iterate over all blocks starting in block[1]
borough= True # starting with a borough
for blk in blocks[1:]:
    if borough:
        # borough
        b= blk.text
        # set next element to be processed as a neighborhood
        borough= not borough
        continue
    else:
        # neighborhood list
        nbhd_list= blk.text.replace ('\t', '').split ('\n')

        # set next element to be processed as a borough
        borough= not borough

        # populate our dataframe, ignore empty strings
        for nbhd in list (filter (None, nbhd_list)):
            boroghs_sp.loc[i]= [b.strip (), nbhd.strip ()]
            i= i+1

boroghs_sp
```

Out[84]:

	Borough	Neighborhood
0	Centro	Sé Bela Vista
1	Centro	Bom Retiro
2	Centro	Cambuci
3	Centro	Consolação
4	Centro	Liberdade
...
93	Sul	Campo Grande
94	Sul	Santo Amaro
95	Sul	Moema
96	Sul	Saúde
97	Sul	Vila Mariana

98 rows × 2 columns

Above you can see a few neighborhoods in Central area of São Paulo city.

Let check the shape of our dataframe.

In [48]:

```
boroghs_sp.shape
```

Out[48]:

(98, 2)

Retring data from Foursquare of veneus around São Paulo boroghs.

Now we have the boroghs names I adjusted the Foursquare URL to use the **near** API entry point. I change the number of data to retrieve and extended the circle around the borogh as 500m looks like so small.

In [82]:

```
# initialize my Foursquare credentials
CLIENT_ID = 'FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH' # my Foursquare I
D
CLIENT_SECRET = 'L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q' # my Foursqua
re Secret
ACCESS_TOKEN = 'YJ55TLGLG5XTRKKQ0PZZ1RVVWVZ4EK4BWJLS2ZCDA100YDVJ' # my FourSqua
re Access Token
VERSION = '20180604'
LIMIT = 500
print ("Foursquare credentilas are set.")
```

Foursquare credentilas are set.

In [107]:

```
def getNearbyVenues(df):
    # radius around the lat,lng
    radius=3000

    # iterate over the input data frame getting the neighborhood name, escape any space on it.

    # define the base Foursquare API URL using the near keyword
    base_url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&near={}&radius={}'

    venues_list=[]
    for i, r in df.iterrows():
        nbhd= r[1]
        borogh= r[0]
        near= nbhd + ",São Paulo,SP"
        near= requests.utils.quote(near)
        #print ("DEBUG: borogh:{}, nbhd: {}, near:{}".format (r[0], nbhd, near))

        # create the request URL using the 'near' API endpoint
        url= base_url.format(CLIENT_ID, CLIENT_SECRET, VERSION, near, radius)
        # GET request
        result = requests.get(url).json()

        # check the result, skip if return code is not 200
        if result['meta']['code']!=200:
            #print (url)
            print ("{: Request return error code {}, error type {}".format (url
, result['meta']['code'], result['meta']['errorType']))
            #print (result['meta']['errorType'])
            #print (result['meta']['errorDetail'])
            #print ()
            continue

        # iterate over Foursquare data, get the venue name and category
        items= result["response"]['groups'][0]['items']
        for i in items:
            v= i['venue']
            lat= v['location']['lat']
            lng= v['location']['lng']
            try:
                name= v['name']
            except KeyError:
                name= 'NA'
            try:
                addr= v['location']['address']
            except KeyError:
                addr= 'NA'
            try:
                pc= v['location']['postalCode']
            except KeyError:
                pc= 'NA'
            try:
                nbhd= v['location']['neighborhood']
            except KeyError:
                # do nothing, use the nbhd set in input argument
                pass
```

```
try:
    categ= v['categories'][0]['name']
except KeyError:
    categ= 'NA'
venue= (name, addr, pc, borogh, nbhd, categ, lat, lng)
venues_list.append(venue)

# return venues_list as a dataframe
nearby= pd.DataFrame(data= venues_list, columns= ['Venue', 'Address', 'PostalCode', 'Borough', 'Neighborhood', 'Category', 'V_Lat', 'V_Long'])
return(nearby)
```

In [108]:

```
print ("This may take a while...be patient")
venues_SP= getNearbyVenues(boroughs_sp)
print ("...done.")
```

This may take a while...be patient

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=S%C3%A9%20Bela%20Vista%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Cambuci%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Santa%20Cec%C3%ADlia%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Carr%C3%A3o%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Formosa%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Ponte%20Rasa%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Curu%C3%A7%C3%A1%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Cidade%20L%C3%ADder%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Parque%20do%20Carro%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Mooca%20%C3%81gua%20Rasa%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Bel%C3%A9m%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Mo%C3%B3ca%2C%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Artur%20Alvim%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Canga%3ADba%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Matilde%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=S%3%A3o%20Rafael%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Jacu%3AD%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=S%3%A3o%20Lucas%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Brasil%3A2ndia%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Freguesia%20do%20%3A93%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Ja%3A7an%3A3%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Anhanguera%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=S%3%A3o%20Domingos%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Mandaqui%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3LOEUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSGEXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Mandaqui%2CS%3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, error type failed_geocode

EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Guilherme%
2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400,
error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Medeiros%2
CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400,
error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Morumbi%2CS%C3%A3
o%20Paulo%2CSP&radius=3000: Request return error code 400, error ty
pe failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Raposo%20Tavares%
2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400,
error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20S%C3%B4ni
a%2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 40
0, error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Barra%20Funda%2C
S%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, e
rror type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Jaguara%2CS%C3%A3
o%20Paulo%2CSP&radius=3000: Request return error code 400, error ty
pe failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Jaguar%C3%A9%2CS%
C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, err
or type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Perdizes%2CS%C3%A
3o%20Paulo%2CSP&radius=3000: Request return error code 400, error t
ype failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Leopoldin
a%2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 40
0, error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Alto%20de%20Pinhe
iros%2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code
400, error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Vila%20Andrade%2C
S%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, e
rror type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG

```

EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Capela%20do%20Soc
orro%20Cidade%20Dutra%2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request
return error code 400, error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Graja%C3%BA%2CS%C
3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, erro
r type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Sacom%C3%A3%2CS%C
3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, erro
r type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Jardim%20%C3%82ng
ela%2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return error code
400, error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Jardim%20S%C3%A3
o%20Lu%C3%ADs%2CS%C3%A3o%20Paulo%2CSP&radius=3000: Request return e
rror code 400, error type failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Marsilac%2CS%C3%A
3o%20Paulo%2CSP&radius=3000: Request return error code 400, error t
ype failed_geocode
https://api.foursquare.com/v2/venues/explore?&client_id=FZDPFCII3L0
EUDNI2GETXZM2T2AFWPKWZMDP4VX5X30K4DFH&client_secret=L314JHHIHHYXUSG
EXJU1UZ5JSXOWNHACRNWAAD5EWPFHJ05Q&v=20180604&near=Campo%20Belo%2CS%
C3%A3o%20Paulo%2CSP&radius=3000: Request return error code 400, err
or type failed_geocode
...done.

```

Looks like FourSquare didn't recognized some boroughs names, it's fine.

In [109]:

```
venues_SP.head()
```

Out[109]:

	Venue	Address	PostalCode	Borough	Neighborhood	Category	V_Lat	V_
0	Um Coffee Co.	R. Júlio Conceição, 553	01126-001	Centro	Bom Retiro	Coffee Shop	-23.527495	-46.64
1	Acrópolis	R. da Graça, 364	01125-000	Centro	Bom Retiro	Greek Restaurant	-23.528093	-46.63
2	Béni Café	R. Lubavitch, 79	01123-010	Centro	Bom Retiro	Café	-23.529453	-46.63
3	Fresh Cake Factory	R. Prates, 585	01121-000	Centro	Bom Retiro	Dessert Shop	-23.528858	-46.63
4	Malagueta Fashion	Rua José Paulino, 541	01120-001	Centro	Bom Retiro	Women's Store	-23.529574	-46.63

In [110]:

```
venues_SP.shape
```

Out[110]:

(1543, 8)

In [111]:

```
venues_SP['Neighborhood'].unique ()
```

Out[111]:

```
array(['Bom Retiro', 'Campos Elísios', 'Santa Cecília', 'Luz',
      'Santa Ifigênia', 'Consolação', 'Baixo Augusta', 'Bela Vista',
      'paraíso', 'Conso', 'Avenida Paulista', 'Liberdade', 'Centro',
      'Sé', 'Jardim Eroísi', 'Nova Leme', 'Aricanduva',
      'Cidade Tiradentes', 'Itaquera', 'José Bonifácio',
      'Ermelino Matarazzo', 'Vila Norma', 'Guaianases', 'Lajeado',
      'Vila Michelin', 'Itaim Paulista', 'Vila Anny', 'Brás',
      'Mooca, São Paulo, SP', 'Moóca', 'Pari', 'Tatuapé', 'Penha',
      'Vila Gui', 'São Mateus', 'Cidade Satélite Santa Bárbara',
      'São Miguel', 'Jardim Helena', 'Vila Belvedere', 'Sapopemba',
      'Vila Prudente', 'Vila Zelina', 'Jardim Avelino', 'Casa Verde',
      'Vila Paulicéia', 'Santana', 'Cachoeirinha', 'Rosa Mendes',
      'Limão', 'Jardim Primavera', 'Jardim Santana', 'Perus', 'Pirituba',
      'Jaraguá', 'Santa Fé', 'Tucuruvi', 'Vila Maria', 'Vila Guilherme',
      'Butantã', 'Universidade de São Paulo', 'Butantã, São Paulo, SP',
      'Vila Indiana', 'Rio Pequeno', 'Lapa', 'Vila Romana', 'Pinheiros',
      'Itaim Bibi', 'Jardim Paulista', 'Picanço', 'Campo Limpo',
      'Vila Prel', 'pq pinheiros', 'Capão Redondo', 'Socorro',
      'Vila Missionária', 'Vila Castelo', 'Cidade Ademar', 'Pedreira',
      'Ipiranga', 'Cursino', 'Vila Gumerindo', 'Klabín', 'Vila Mariana',
      'Jabaquara', "M'Boi Mirim", 'Jardim Ângela', 'jardim valo velho',
      'Jardim São Luís', 'Parelheiros', 'Santo Amaro', 'Campo Grande',
      'Vila Sofia', 'José Menino', 'Gonzaga', 'Morro da', 'Moema',
      'Vila Nova Conceição', 'Vl. Nova Conceição', 'Saúde', 'Santa Cruz',
      'Vila Clementino', 'Jardim da Glória', 'Aclimação'], dtype=object)
```

In [112]:

```
venues_SP['Category'].unique ().shape
```

Out[112]:

(219,)

In [113]:

```
venues_SP.groupby('Borough')['Venue'].count().sort_values(ascending=False)
```

Out[113]:

```
Borough
Leste      525
Sul        409
Norte      249
Oeste      210
Centro     150
Name: Venue, dtype: int64
```

In [114]:

```
venues_East= venues_SP[venues_SP['Borough']=='Leste']
venues_East.groupby('Category')['Neighborhood'].count().sort_values(ascending=False)
```

Out[114]:

```
Category
Bakery          45
Pizza Place     28
Brazilian Restaurant 28
Gym / Fitness Center 24
Ice Cream Shop  23
..
Home Service    1
Juice Bar       1
Lingerie Store  1
Liquor Store    1
Airport         1
Name: Neighborhood, Length: 115, dtype: int64
```

Even there are failed requests caused by unrecognized boroughs names, we are plenty of data to work with:

Our dataframe has:

- 1543 venues in the 5 zones/boroughs,
- 219 venues categories,
- 525 venues in Zona Leste, which is the borough of our interest,
- 45 bakeries in Zona Leste

Mapping the venues in São Paulo

Lets put on a map the venues returned by FourSquare, in each borough (*Zona*) of São Paulo to see what we got.

To show each venue in a Borough(*Zona*) lets add a borough id to our dataframe. Later, we use that borough id to pick a color for each borough.

In [118]:

```
borough_list= venues_SP.Borough.unique()
borough_ids= [i for i in range(0, len(borough_list))]
```

In [119]:

```
venues_SP['borogh_id']= venues_SP['Borogh'].replace(to_replace=borogh_list, value=borogh_ids, inplace=False)
venues_SP.head ()
```

Out[119]:

	Venue	Address	PostalCode	Borogh	Neighborhood	Category	V_Lat	V_
0	Um Coffee Co.	R. Júlio Conceição, 553	01126-001	Centro	Bom Retiro	Coffee Shop	-23.527495	-46.64
1	Acrópolis	R. da Graça, 364	01125-000	Centro	Bom Retiro	Greek Restaurant	-23.528093	-46.63
2	Béni Café	R. Lubavitch, 79	01123-010	Centro	Bom Retiro	Café	-23.529453	-46.63
3	Fresh Cake Factory	R. Prates, 585	01121-000	Centro	Bom Retiro	Dessert Shop	-23.528858	-46.63
4	Malagueta Fashion	Rua José Paulino, 541	01120-001	Centro	Bom Retiro	Women's Store	-23.529574	-46.63

now we are ready to plot the venues by borogh. We just need the coordinates of SP to center it on the map.

In [137]:

```
# Here is a good point to save our dataframes in a file, case we need go back without having to redo all this.
file= 'Battle-of_neighborhoods_w1_venues_SP.csv'
venues_SP.to_csv(file)
print('CSV file {} was saved!'.format (file))

#Save data so we can go back to this at any time.
file= 'Battle-of_neighborhoods_w1_boroghs_sp.csv'
boroghs_sp.to_csv(file)
print('CSV file {} was saved!'.format (file))
```

CSV file Battle-of_neighborhoods_w1_venues_SP.csv was saved!
 CSV file Battle-of_neighborhoods_w1_boroghs_sp.csv was saved!

In [15]:

```
# Use this to load our dataframes saved above
###
### YOU CAN SKIP THIS IF YOU ARE RUNNING THIS NOTEBOOK FROM THE BEGINING
###
file= 'Battle-of_neighborhoods_w1_venues_SP.csv'
venues_SP= pd.read_csv(file)
venues_SP.drop ('Unnamed: 0', axis=1, inplace=True)
venues_SP.head ()

## Use this code to import the DataFrame saved above
file= 'Battle-of_neighborhoods_w1_boroughs_sp.csv'
boroughs_sp= pd.read_csv(file)
boroughs_sp.drop ('Unnamed: 0', axis=1, inplace=True)
# we need to redo theses as it are used ahead
borough_list= venues_SP.Borough.unique()
borough_ids= [i for i in range (0, len (borough_list))]
venues_East= venues_SP[venues_SP['Borough']=='Leste']
venues_East.groupby('Category')['Neighborhood'].count ().sort_values(ascending=False)

#
boroughs_sp.head ()
```

Out[15]:

	Borough	Neighborhood
0	Centro	Sé Bela Vista
1	Centro	Bom Retiro
2	Centro	Cambuci
3	Centro	Consolação
4	Centro	Liberdade

Get the coordinates of São Paulo city.

In [9]:

```
from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values
interest= "São Paulo, SP, Brazil"
geolocator = Nominatim(user_agent="sp_explorer")
saopaulo = geolocator.geocode(interest)
print('The coordinates of São Paulo are {}, {}'.format(saopaulo.latitude, saopaulo.longitude))
```

The coordinates of São Paulo are -23.5506507, -46.6333824.

Let's define a function to draw a legend in the right-top corner of map.

In [10]:

```

def add_categorical_legend(folium_map, title, colors, labels):
    if len(colors) != len(labels):
        raise ValueError("colors and labels must have the same length.")

    color_by_label = dict(zip(labels, colors))

    legend_categories = ""
    for label, color in color_by_label.items():
        legend_categories += f"<li><span style='background:{color}'></span>{label}</li>"

    legend_html = f"""
<div id='maplegend' class='maplegend'>
  <div class='legend-title'>{title}</div>
  <div class='legend-scale'>
    <ul class='legend-labels'>
      {legend_categories}
    </ul>
  </div>
</div>
"""
    script = f"""
<script type="text/javascript">
  var oneTimeExecution = (function() {{
    var executed = false;
    return function() {{
      if (!executed) {{
        var checkExist = setInterval(function() {{
          if ((document.getElementsByClassName('leaflet-top leaflet-right').length) || (!executed)) {{
            document.getElementsByClassName('leaflet-top leaflet-right')[0].style.display = "flex"
            document.getElementsByClassName('leaflet-top leaflet-right')[0].style.flexDirection = "column"
            document.getElementsByClassName('leaflet-top leaflet-right')[0].innerHTML += `{legend_html}`;
            clearInterval(checkExist);
            executed = true;
          }}
        }}, 100);
      }}
    }});
  })();
  oneTimeExecution()
</script>
"""

    css = """
<style type='text/css'>
  .maplegend {
    z-index:9999;
    float:right;
    background-color: rgba(255, 255, 255, 1);
    border-radius: 5px;
    border: 2px solid #bbb;
    padding: 10px;
    font-size:12px;
  }

```

```
        positon: relative;
    }
    .maplegend .legend-title {
        text-align: left;
        margin-bottom: 5px;
        font-weight: bold;
        font-size: 90%;
    }
    .maplegend .legend-scale ul {
        margin: 0;
        margin-bottom: 5px;
        padding: 0;
        float: left;
        list-style: none;
    }
    .maplegend .legend-scale ul li {
        font-size: 80%;
        list-style: none;
        margin-left: 0;
        line-height: 18px;
        margin-bottom: 2px;
    }
    .maplegend ul.legend-labels li span {
        display: block;
        float: left;
        height: 16px;
        width: 30px;
        margin-right: 5px;
        margin-left: 0;
        border: 0px solid #ccc;
    }
    .maplegend .legend-source {
        font-size: 80%;
        color: #777;
        clear: both;
    }
    .maplegend a {
        color: #777;
    }
</style>
"""

folium_map.get_root().header.add_child(folium.Element(script + css))

return folium_map
```

In [11]:

```

import matplotlib.cm as cm
import folium # plotting library
import matplotlib.colors as colors

borough_names= venues_SP.Borough.unique()

# n= number of Boroughs
n= len (borough_ids)

# create map
map_SP = folium.Map(location=[saopaulo.latitude, saopaulo.longitude], zoom_start
=11)
title= "Venues by Boroughs in São Paulo"
subtitle= "marker: neighborhood: venue(category)"
title_html = '''
        <h3 align="center" style="font-size:16px"><b>{}</b></h3>
        <h2 align="center" style="font-size:14px"><b>{}</b></h2>
        '''.format(title, subtitle)
map_SP.get_root().html.add_child(folium.Element(title_html))

# set color scheme for the clusters
x = np.arange(n)
ys = [i + x + (i*x)**2 for i in range(n)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add venues to the map
# Venue Address PostalCode      Neighborhood      Category      V_Lat      V_Long
for name, category, b_id, nbhd, v_lat, v_lon, in zip(venues_SP['Venue'], venues_
SP['Category'], venues_SP['borough_id'], venues_SP['Neighborhood'], venues_SP['V_
Lat'], venues_SP['V_Long']):
    label = '{}: {}({})'.format(nbhd, name, category)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [v_lat, v_lon],
        radius=3,
        popup=label,
        color=rainbow[b_id],
        fill=True,
        fill_color=rainbow[b_id],
        fill_opacity=0.6).add_to(map_SP)

# add a legend on the right upper corner
borough_colors= []
borough_legends= []
for i in range (n):
    borough_colors.append (rainbow[i])
    nbhd=borough_list[i]
    borough_legends.append (nbhd)

map_SP= add_categorical_legend(map_SP, 'Boroughs',
                               colors = borough_colors,
                               labels = borough_legends)

print ("done")

```

done

In [12]:

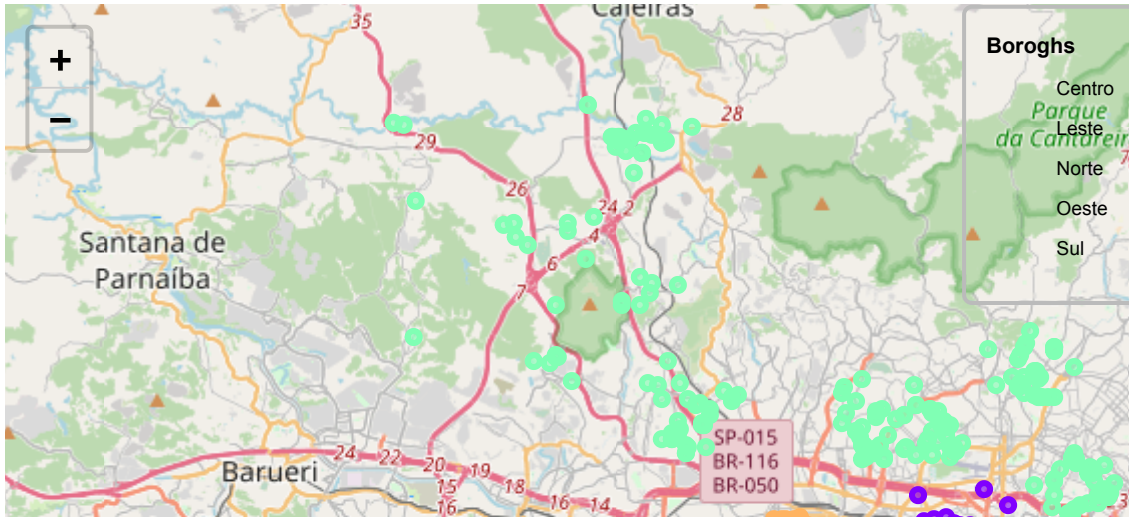
```
map_SP
```

Out[12]:

Make this Notebook Trusted to load map: File -> Trust Notebook

Venues by Boroghs in São Paulo

marker: neighborhood: venue(category)



Nice ! We clearly see the venues in the zones North, South, East, West and Center of São Paulo.

There is a minor glitch regarding the some venues placed in the North-East as West zone. This is because they are in fact, in another county (Guarulhos), and somehow it was accounted as it is in São Paulo. In fact, that cluster is in West of Guarulhos city. They don't affect our analysis, so let it aside.

Now we are sure our data is good, let move on to focus our analysis on the bakeries in *Zona Leste* (East) of São Paulo.

***Padarias* in Zona Leste of São Paulo city**

Lets filter our dataframe to focus on bakeries in Zona Lest of São Paulo.

I will create two dataframes to work on it. The first one has only with Bakeries on it and the other has everything else.

The reasoning of having a bakery dataset and a not-a-bakery dataset is I want to map the bakeries along the other business in the area.

As this kind of business needs a great flux of people near by, the higher the number of others business, better to a future place for a bakery.

So, our criteria to choose a place to out a new bakery is a place near other business and far from other bakeries in the area.

In [26]:

```
bakeries_East= venues_East[venues_East.Category=='Bakery']  
bakeries_East.shape
```

Out[26]:

(45, 9)

In [27]:

```
notBakery= bakeries_East.copy ()  
notBakery.shape
```

Out[27]:

(45, 9)

In [30]:

```
notBakery= venues_East[~venues_East.Category.str.contains ('Bakery')]
notBakery.Category.unique ()
```

Out[30]:

```
array(['Dessert Shop', 'Gym / Fitness Center', 'Clothing Store',
      'Pizza Place', 'Brewery', 'Pet Store', 'Pastelaria',
      'Frozen Yogurt Shop', 'Food & Drink Shop', 'Steakhouse',
      'Pharmacy', 'Food Truck', 'Soccer Field', 'Snack Place', 'Sc
hool',
      'Miscellaneous Shop', 'Taco Place', 'Italian Restaurant',
      'Convenience Store', 'Cosmetics Shop', 'Gym', 'Café', 'Par
k',
      'Brazilian Restaurant', 'Big Box Store', 'Ice Cream Shop',
      'Market', 'Gymnastics Gym', 'Japanese Restaurant', 'Flower S
hop',
      'Student Center', 'Grocery Store', 'Fast Food Restaurant',
      'Aquarium', 'Churrascaria', 'Farmers Market', 'Shop & Servic
e',
      'BBQ Joint', 'Comfort Food Restaurant', 'Paintball Field',
      'Northeastern Brazilian Restaurant', 'Plaza', 'Supermarket',
      'Chinese Restaurant', 'Historic Site', 'Sandwich Place',
      'Restaurant', 'Diner', 'Juice Bar', 'Burger Joint',
      'Asian Restaurant', 'Neighborhood', 'Chocolate Shop', 'Drugs
tore',
      'Bar', 'Deli / Bodega', 'Vegetarian / Vegan Restaurant', 'Ai
rport',
      'Candy Store', 'Creperie', 'Bistro', 'Wine Shop', 'Veterinar
ian',
      'Bagel Shop', 'Central Brazilian Restaurant', 'Persian Resta
urant',
      'Nightclub', 'Coffee Shop', 'Hotel', 'History Museum',
      'Cultural Center', 'Wine Bar', 'Health Food Store',
      'Middle Eastern Restaurant', 'Home Service', 'Lingerie Stor
e',
      'Beach Bar', 'Water Park', 'Seafood Restaurant',
      'Department Store', 'Theater', 'Peruvian Restaurant', 'Beer
Bar',
      'Hawaiian Restaurant', 'Soccer Stadium', 'Mexican Restaura
nt',
      'Athletics & Sports', 'Fruit & Vegetable Store', 'Gourmet Sh
op',
      'Dance Studio', 'Dive Bar', 'Martial Arts School', 'Cupcake
Shop',
      'Trail', 'Pub', 'General Entertainment', 'Food', 'Rock Clu
b',
      'Gay Bar', 'Gym Pool', 'Paper / Office Supplies Store',
      'Salad Place', 'Tea Room', 'Health & Beauty Service', 'Cafet
eria',
      'Gas Station', 'Breakfast Spot', 'Smoke Shop', 'Liquor Stor
e',
      'Warehouse Store', 'Church', 'Salon / Barbershop', "Men's St
ore",
      'Butcher'], dtype=object)
```

In [31]:

```
len(notBakery.Category.unique())
```

Out[31]:

114

Results section

Mapping the bakeries and not bakeries in São Paulo map.

This map will show us where the bakeries and the not-a-bakery are in Zona Leste of SP.

Then, we can, visually, find a place where there are other business around and no bakeries near by, which is, supposedly, a good place to start a *Padaria*, in Zona Leste of São Paulo.

In [32]:

```
import matplotlib.cm as cm
import folium # plotting library
import matplotlib.colors as colors

borogh_names= venues_SP.Borogh.unique()

# n= number of business: bakeries/not bakery
n= 2

# East of São Paulo coordinates
leste_latitude= -23.5471308
leste_longitude= -46.4970469

# create bakeries map
# bakeries_East

bakeries_map = folium.Map(location=[leste_latitude, leste_longitude], zoom_start
=13, control_scale= True)
title= "Bakeries and not bakeries in East of São Paulo"
subtitle= "marker: neighborhood: venue(category)"
title_html = '''
    <h3 align="center" style="font-size:16px"><b>{}</b></h3>
    <h2 align="center" style="font-size:14px"><b>{}</b></h2>
    '''.format(title, subtitle)
bakeries_map.get_root().html.add_child(folium.Element(title_html))

# set color scheme for the clusters
x = np.arange(n)
ys = [i + x + (i*x)**2 for i in range(n)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add bakeries to the map
# Venue Address PostalCode      Neighborhood      Category      V_Lat      V_Long
for name, category, nbhd, v_lat, v_lon, in zip(bakeries_East['Venue'], bakeries_
East['Category'], bakeries_East['Neighborhood'], bakeries_East['V_Lat'], bakerie
s_East['V_Long']):
    b_id= 0
    label = '{}: {}({})'.format(nbhd, name, category)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [v_lat, v_lon],
        radius=4,
        popup=label,
        color=rainbow[b_id],
        fill=False,
        fill_color=rainbow[b_id],
        fill_opacity=0.6).add_to(bakeries_map)

# add not-bakery to the map
# Venue Address PostalCode      Neighborhood      Category      V_Lat      V_Long
for name, category, nbhd, v_lat, v_lon, in zip(notBakery['Venue'], notBakery['Ca
teory'], notBakery['Neighborhood'], notBakery['V_Lat'], notBakery['V_Long']):
    b_id= 1
    label = '{}: {}({})'.format(nbhd, name, category)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [v_lat, v_lon],
```



```
radius=3,
popup=label,
color=rainbow[b_id],
fill=True,
fill_color=rainbow[b_id],
fill_opacity=0.6).add_to(bakeries_map)
```

```
# add a legend on the right upper corner
```

```
bakery_colors= []
```

```
bakery_colors.append (rainbow[0])
```

```
bakery_colors.append (rainbow[1])
```

```
bakery_legends= ['Bakery', 'not a Bakery']
```

```
bakeries_map= add_categorical_legend(bakeries_map, 'Business type',
                                     colors = bakery_colors,
                                     labels = bakery_legends)
```

```
print ("done")
```

done

In [33]:

```
bakeries_map
```

Out[33]:

Make this Notebook Trusted to load map: File -> Trust Notebook

Bakeries and not bakeries in East of São Paulo

marker: neighborhood: venue(category)



Discussion

This study started with the assumption that the stackholder wanted a new business in the East zone of São Paulo. However this code could easily adapted to do this analysis in any other borough of choice.

If there was not a preferential borough to start a *padaria* we could do a quick study about wich borough is most suitable to receive a *padaria* based on the relative proportion of bakary/not-bakary ratio in each borough and select the borough with the lowest ratio.

And not only bakaries !

We can adapt this study to deal with any other kind of business that a stackholder is willing to start.

Conclusion

As we can see in the above map, there are a few places where you can start a *Padaria* that is located near from other business and far from others *padarias*. There are even "voids" of other business where potentially can have a *padaria*. Although theses regions are empty of others business, for sure, they are populated areas that will welcome a new *padaria* in their neighborhood.

In []: