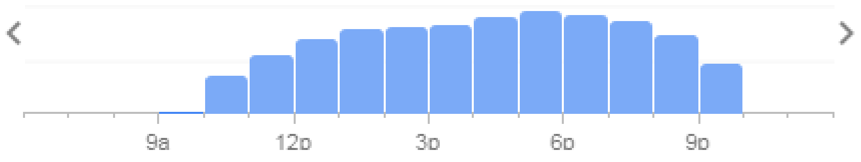
Mapping the most popular places in the city

Using Python, Pandas and Folium

Maradona Morais
May 22, 2019 · 6 min read

"Mass" by Gaetano Virgallito. Licensed under CC BY-ND 2.0

Some studies related to analyze human behavior, big cities, consumption of services and others are of my interest when the term "data science" comes up. Last time working with it I made a map for Uber services in Natal, Brazil, interested in see how central neighborhoods are better attended. This time, the interest came from a Google Search feature for places: the "Popular times."



Google, smart as it is, can show us (sometimes even “live”) how busy is the place you searched. So…

What if I compare several places based on popular times? and discover which times certain types of places are more popular than others? many questions can be asked working with that kind of data.

So, why not get different types of places in a 15 miles radius from the center of Natal (city I live in), then catch the popular times data for each place and plot a interactive map with that data? That’s what we’re doing here.

I’ll try to explain, with enough detail to don’t make this article annoying, how I did and how this experiment can be replicated. The technologies used here are Python, pandas and folium libraries. If you want to learn more about, the code is available on GitHub, such as interactives Colab notebooks in which you can upload your own database and generate the maps.

Google, Google…

Before plotting maps, we have to extract the data we’ll need. It will be necessary two types of data: a set of places in Natal and for each of these places, the data about popular moments.

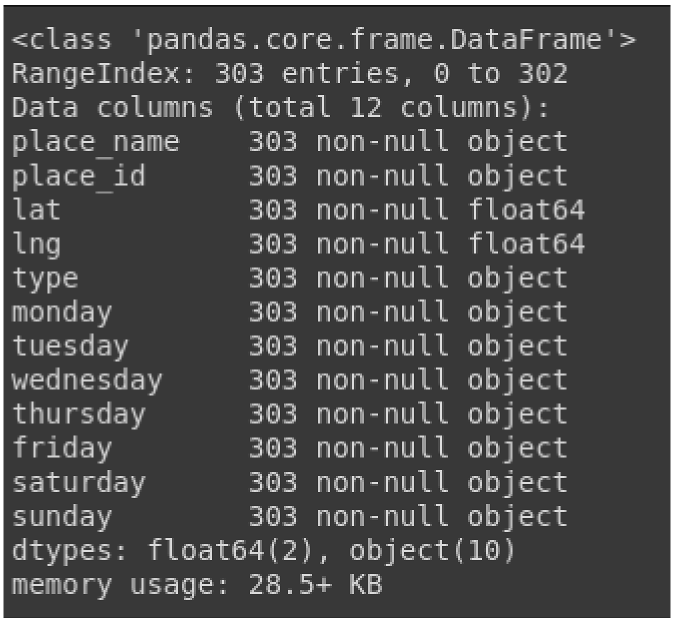
We are going to use Google Places API in order to get the places. There are several services offered by this API, for all of them you’ll need a access token to consume data.

I used the “Nearby Search” to look for places near to a point in the center of Natal and with 15 miles of radius. The `read_places.py` script search these places selecting by types (gym, supermarket, bank…) — for each type I defined the number of places I want; this way I’ll have the most relevant results. Then, the script saves the .csv file for the places dataset as `places.csv` .

topic in the Places API documentation you will note that there is no metion to any "popular times" (at least not until today, May 2019.) But with a quick Google search I found this repo that give us access to this service. It is said in the repo that Google does allow that kind of requesition but it is not for free, the API call is SKU'd as "Find Current Place". You can make 5000 calls to this API with the alloted monthly budget.

Using that API the `get_places_popular_moments.py` reads the places dataset and includes the popular moments on it, generating a new `places_with_moments.csv` dataset. Now, we supposed to have 800 places, but instead we have only 300 because there are no "popular times" for every place.

By now our dataset looks like this:



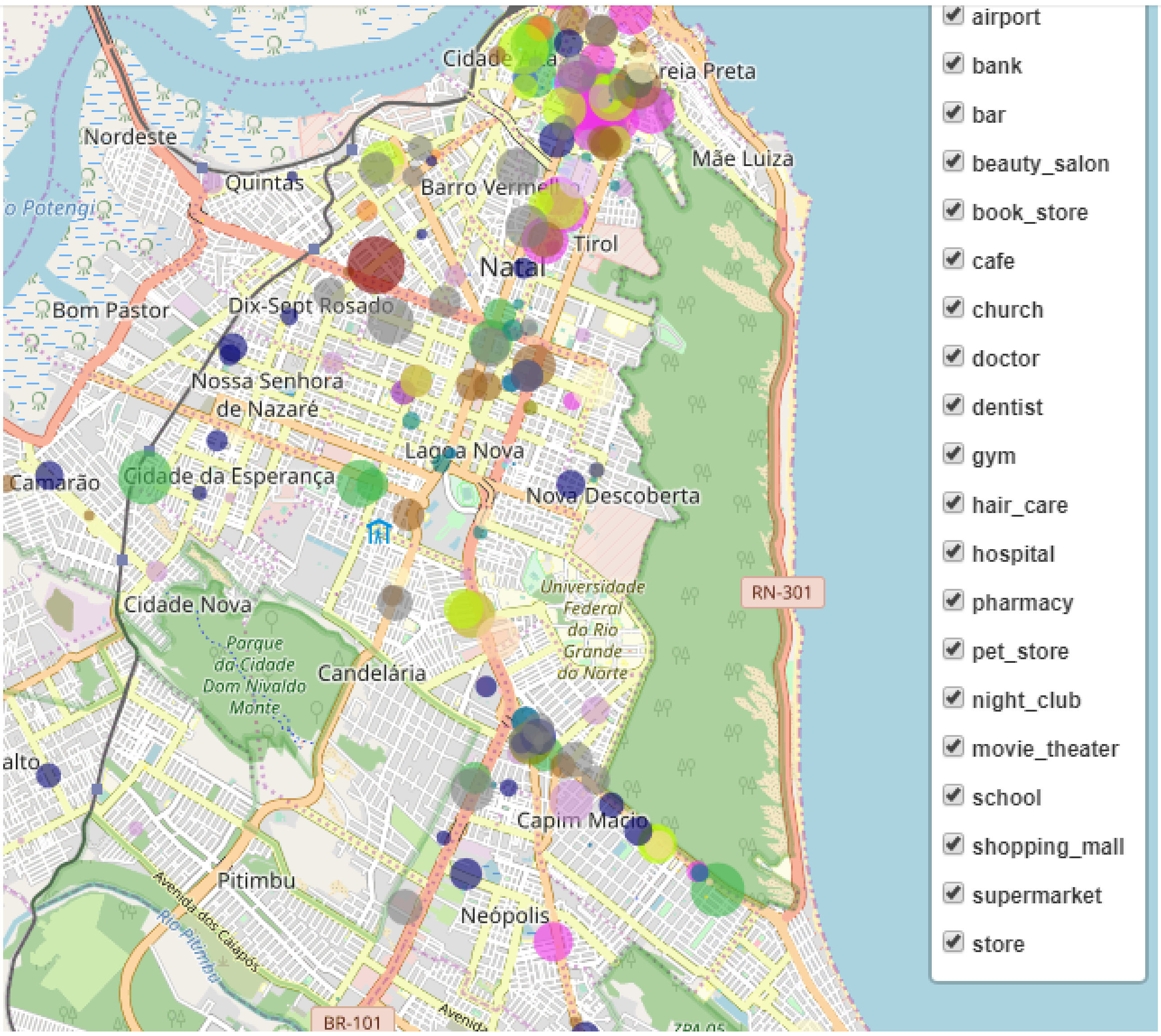
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 12 columns):
place_name    303 non-null object
place_id      303 non-null object
lat           303 non-null float64
lng           303 non-null float64
type          303 non-null object
monday        303 non-null object
tuesday       303 non-null object
wednesday     303 non-null object
thursday      303 non-null object
friday        303 non-null object
saturday      303 non-null object
sunday        303 non-null object
dtypes: float64(2), object(10)
memory usage: 28.5+ KB
```

For each weekday column there is a 24 positions with moment values for each hour of a day.

Generating maps with Folium and Kepler

Now we would like to visualize these data in a pretty and interactive map. We'll use Folium — a Python library based on leaflet.js to visualize maps.

Here's the map of popular moments for places in Natal on mondays 10 a.m. The circle's color means the place's type and the size, the moment value.



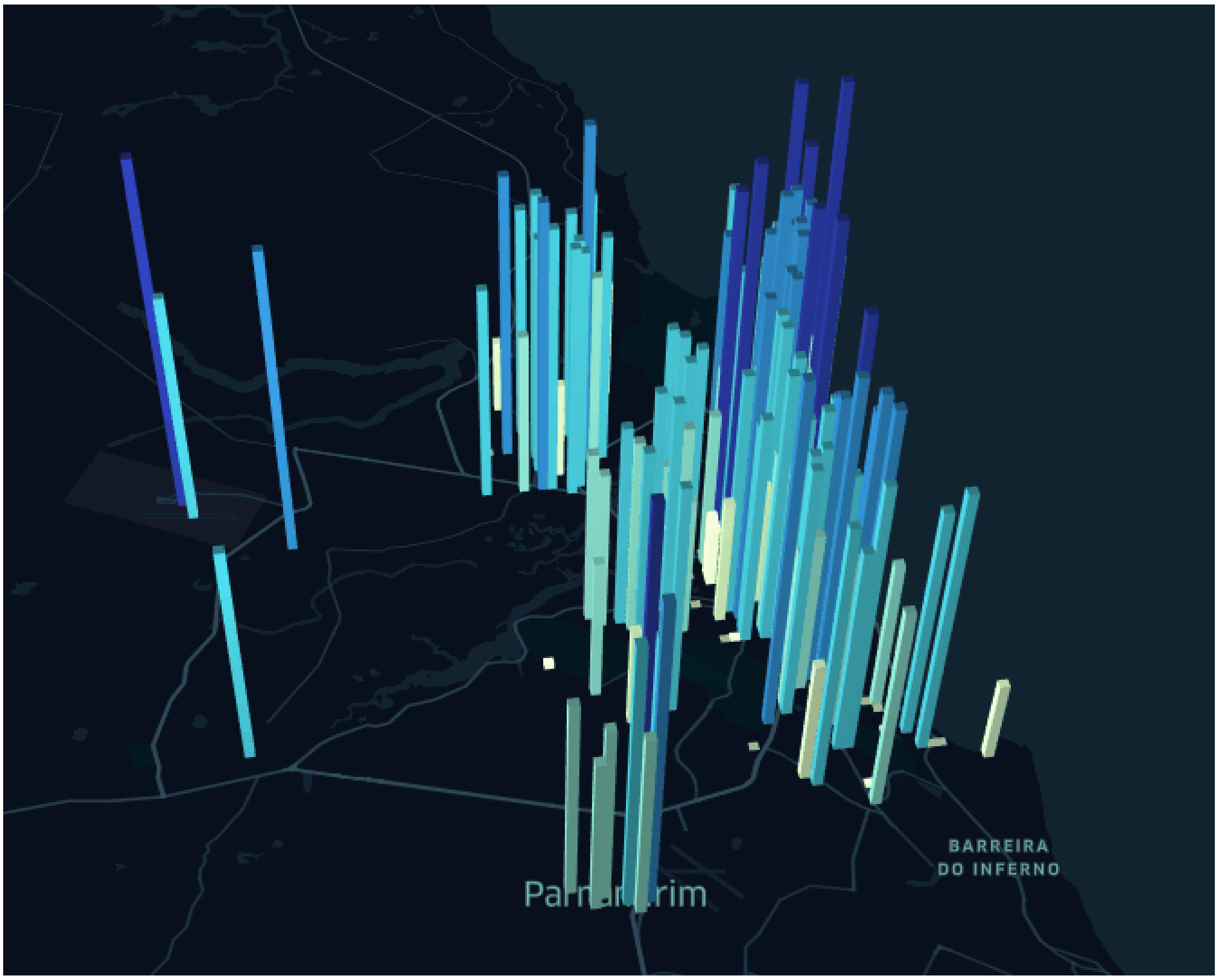
The piece of code that generate this map is this one:

```
def generate_map(weekday, hour, types=places_types):
  natal = folium.Map(location=[-5.831308, -35.20470], zoom_start=13)
  ptypes_groups = {}

  for ptype in types:
    ptypes_groups[ptype] = FeatureGroup(name=ptype)

  for index, place in natal_places.iterrows():
    moments = json.loads(place[weekday])
    if (place.type in types):
      folium.Circle(location=[place.lat, place.lng],
                    radius=int(moments[hour])*3,
                    fill_color=colors[places_types.index(place.type)],
                    fill_opacity=0.6).add_to(ptypes_groups[place.type])
  for ptype, group in ptypes_groups.items():
    group.add_to(natal)
```

It takes as parameter a week day, an hour and, optionally, a subset of places types. The first thing it does is create a new map `natal` and for each desired type it creates a `FeatureGroup` . This resource allows us to create a menu on the map with the option to toggle types.

Looping through all items on dataset we add each item to the corresponding FeatureGroup.

Going deeper in the colab notebook you'll see that there is a modification on the dataset that adapts it to using with Kepler.gl — a powerfull web tool to geospatial analysis. Kepler gives us more resources than Folium and with a extremely good performance. So we can do this:



Anitation made with Kepler

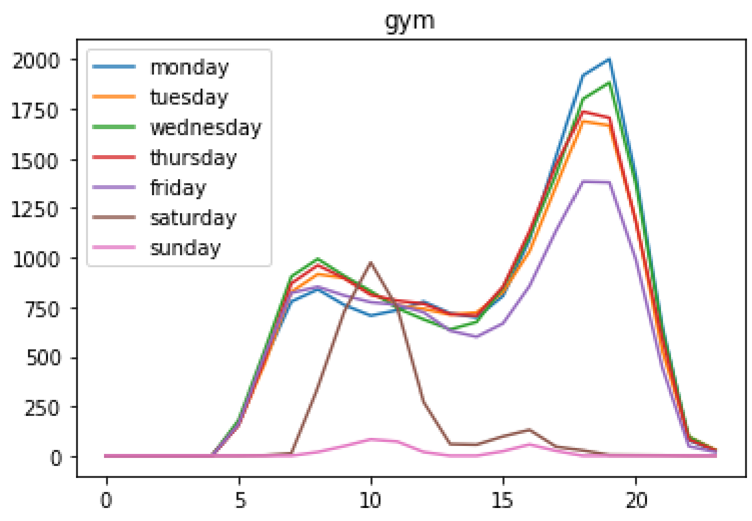
environment to make it acessible online, in this address you can access that map above.

What else can we do?

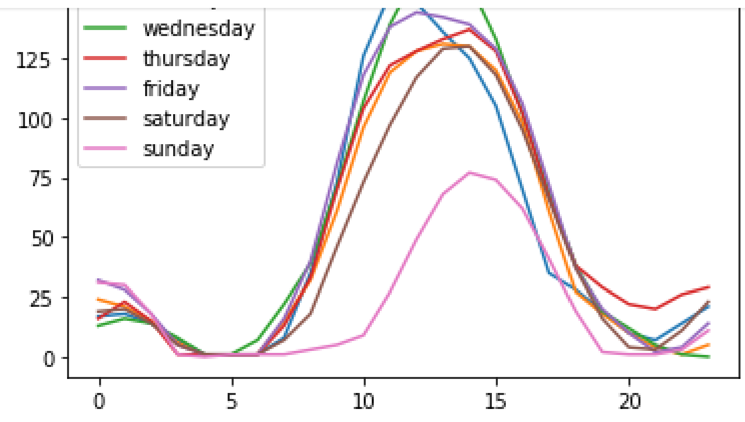
The information about popular times at first glance doesn't means a lot. What Google means with "the place is 100 crowded"? It's a density scale of people in the place? What means, in the end, these popular times data?

Analysing better some values, I realized that a "momet" scale goes from 0 to 100, which leads to classifications like "Usually not too busy" or "Usually a little busy", for instance. But my real question is "how much that information means something?"

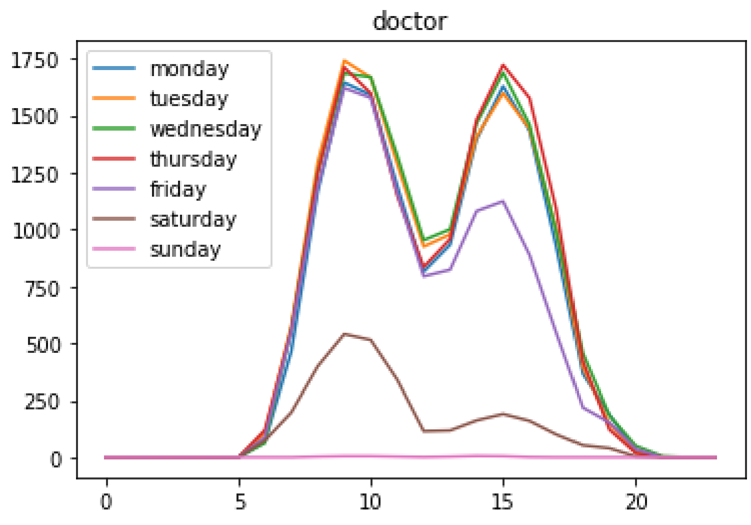
To undestand it better I went for another approach, interested in find some useful informations in that dataset. So here's what I found:



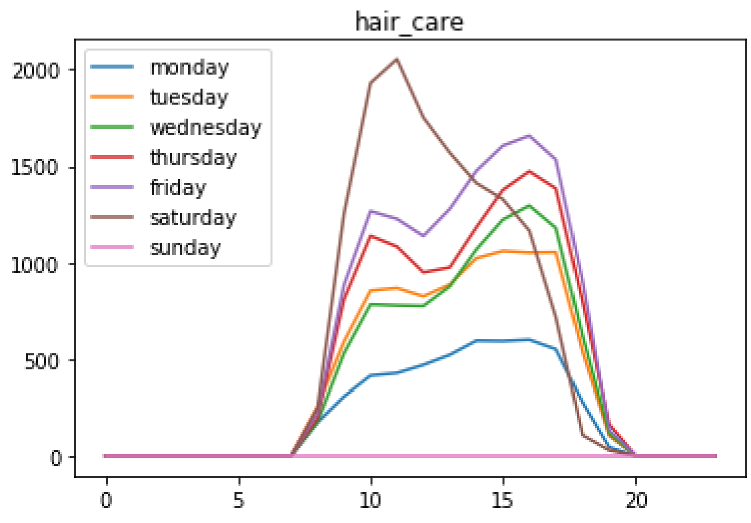
This is a good one. The gym flow shown in the graph is a common sense confirmation: gyms are busier on Mondays and less busy on Fridays. The rush hour is around 20 p.m.



Most of the charts do not have peak hours at lunch time, this is different with airports, which is busier at 12 pm



On Fridays afternoon the medical clinics have a decline.



These are some of the interesting information that can be inferred from the date we have obtained. Overall, I was pleased that the data was compatible with reality, which led me to believe that the is, indeed, a meaning in “popular times” data. These results may lead us to interesting analyzes that may be meaningful for understanding behavior in Natal and, of course, other cities.

Data Science