

Analyzing Airbnb listings in Barcelona

Visualizing Geographically Distributed Data.

Overview

Here, we will use Python libraries **GeoPandas** and **contextily** to read and analyze datasets related to Airbnb listings in Barcelona. We will concentrate on visualizing data on/together with some maps. The datasets can be downloaded [from this link](#). The size of the file is only around 4.5 Mb. Unzip them in your working folder. In other words, your data should be sitting in the `Data\` subfolder of your working folder.

There are two types of datasets. One is a .geojson file that contains “geometry” of Barcelona neighborhoods (73 of them). Others are listings of apartments/rooms available for rent on Airbnb at various times. These were downloaded from Airbnb archives and contain information about locations (including which neighborhood), prices, room types, and more.

As always with Python libraries you will have to work with documentation, especially for visualization tasks. I went over some details in a class lecture, but you still might need to discover certain things by yourself.

This project requires creating a new environment and installing new packages as well as potentially reinstalling some of the old ones. This may require multiple attempts, perhaps updating Anaconda, and/or other packages. Start early!

Task 1.

Create a new environment in Anaconda Navigator - go to ‘Environments’, click ‘Create’, call it ‘geoEnv’ (or something like that) and **use Python 3.8 or 3.9**. This will create a new environment (completely independent of whatever is installed in your base environment) where you will have to install **geopandas**, **descartes** and **geoplot** libraries later. But first check if you have Jupyter Notebook, **numpy**, **matplotlib**, and **pandas** installed in this new environment. Chances are that some are not, so install them first. We walked through the process of creating new environments in a class lecture.

Task 2.

Install **geopandas** in your new **geoEnv** environment.

You also need a **descartes** package installed that will allow drawing polygons. Either install it through Anaconda, or open a Jupyter notebook and type: **conda install descartes** to install it. (Note that you don't need to install descartes if you are working with a CoLabs notebook).

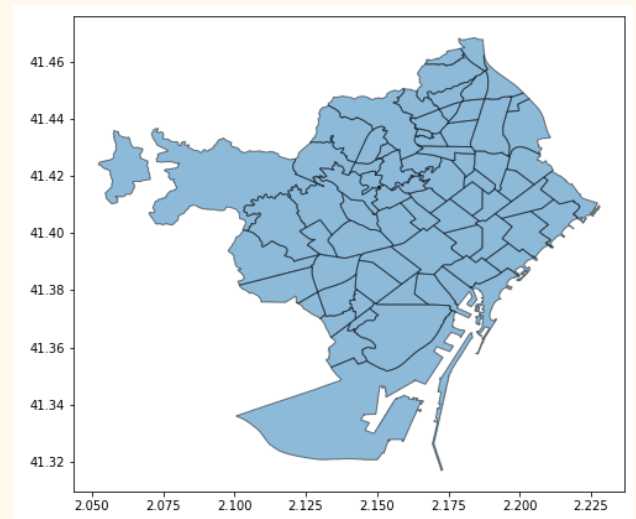
Instead of installing the **contextily** package it may be easier to install a **geoplot** library that brings in several other packages, including **contextily**. You can do it by typing **conda install geoplot -c conda-forge** in a Notebook cell, or in a terminal window.

Check if everything is installed by running the imports in your Notebook.

```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import contextily as ctx
```

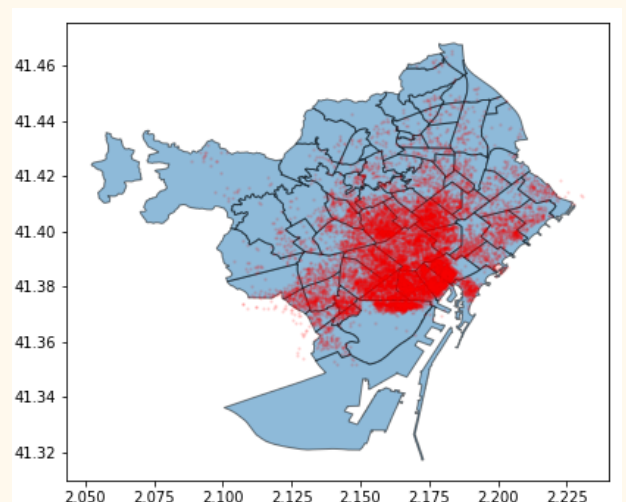
Task 3.

Read '**barris.geojson**' into a GeoPandas dataframe. This will load geographical information about Barcelona neighborhoods (barris in Catalan). As you look carefully through this dataframe, notice that the names column is 'NOM'. There is also an 'AREA' column, and most importantly a 'geometry' column, which contains a Polygon object. You can now easily draw all the polygons using geopandas plot command to get a picture like the one on the right.



Task 4.

Read '**listings_24.csv**' file into a Pandas dataframe. You can then use GeoPandas **points_from_xy()** method to read the longitude and latitude of listings into a geometry column of a new GeoPandas dataframe (see [here](#) for a simple example). Once you have it, it's easy to create a plot of all listings' locations on top of the neighborhoods map, using geopandas plot command. Make it a (10,10) figure size and use semi-transparent red markers.

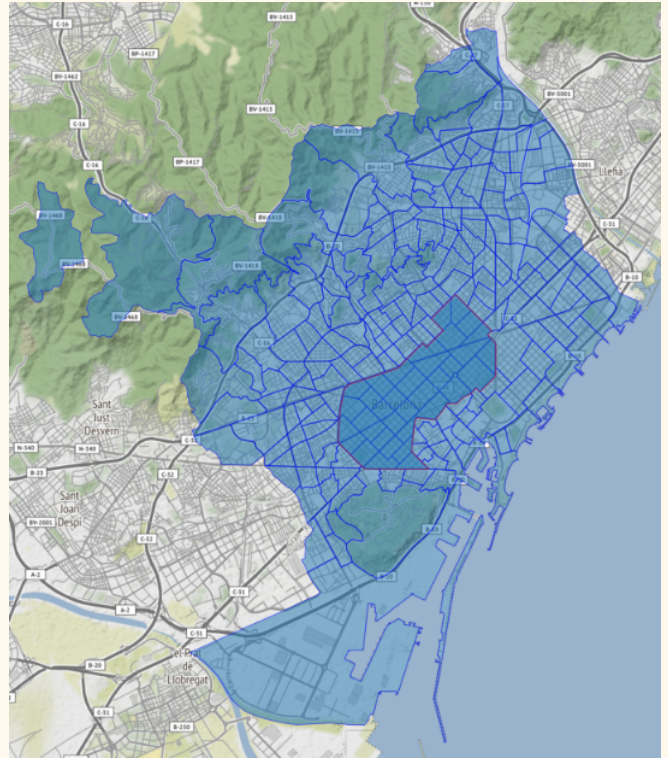


Task 5.

Use **contextily** to add a basemap underneath the polygons picture. Keep the polygons semi-transparent. Read in the districts geometry from `'districts.geojson'` and make the edge of the district named 'Eixample' red (district names would be in column 'NOM').

Don't forget to change the Coordinate Reference System (CRS) for your polygons dataframes so that they end up on top of the basemap.

Small warning: the picture that you see will be different from yours. Mine has smaller polygons (used different file).



Task 6.

Read in `'listings_24.csv'`, `'listings_30.csv'`, `'listings_33.csv'`, and `'listings_45.csv'` as well. The four csv files are Barcelona listings made at different times. For each of those find

- The total number of listings for each neighborhood.
- The “density” of listings (total number divided by the area) for each neighborhood.
- The median price of listings for each neighborhood.

Note: You may need to "clean" the data so you don't end up with NaNs in your values (e.g. price). If you find situations that produce NaN, deal with it the way you see fit and give a short explanation in a markup cell.

Task 7.

Find the same information as in 6 but this time only use listings for 'Entire home/apt' room type.

- The total number of listings for each neighborhood.
- The “density” of listings (total number divided by the area) for each neighborhood.
- The median price of listings for each neighborhood.

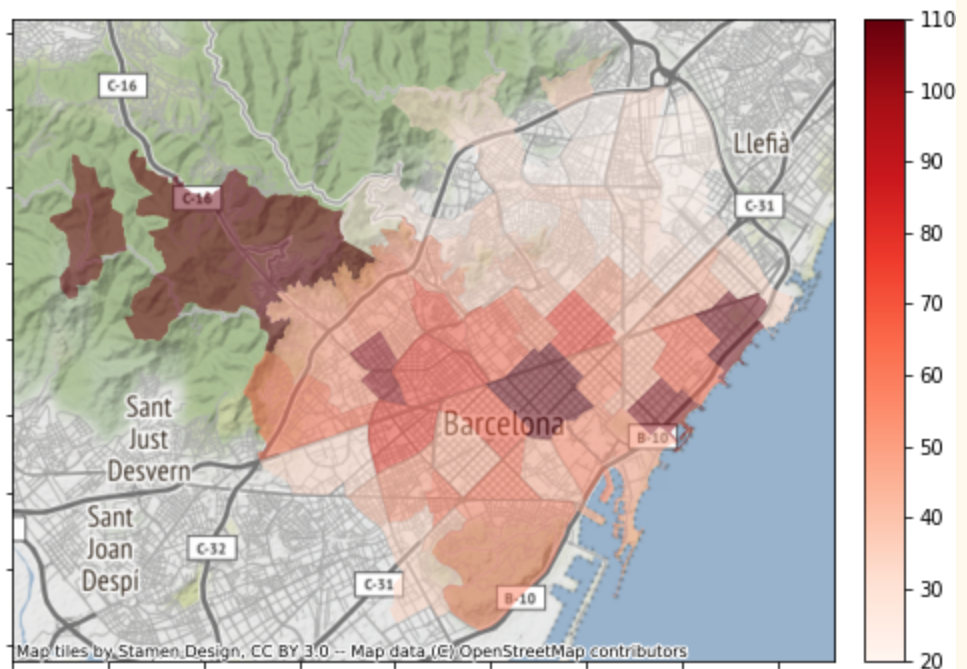
Task 8.

Merge the results in 7 about the “density” and “price” with the dataframe in task 3. In other words, you should have a dataframe with neighborhood information, plus 8 more columns: density_24, density_30, density_33, density_45, price_24, price_30, price_33, price_45. (Suggestion: use `pd.merge(one_dataframe, another_dataframe, on= 'SOME_COLUMN_NAME')`; or something similar, for example, with `'left_on=` and `'right_on=` instead of just `'on=`... `.join` will also work. See some examples [here](#).)

Task 9.

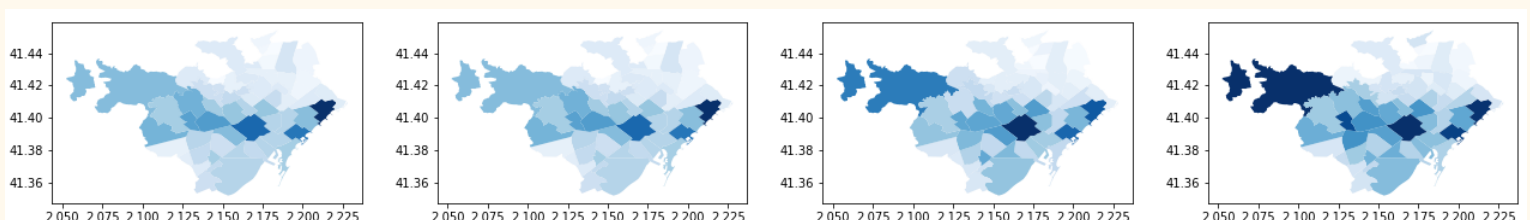
Use options of GeoPandas plot to construct a **choropleth map** in which the neighborhood polygons are shaded in proportion to their price_33 values from your dataframe from Task 8. You can pick any colormap you want from the available options.

Your plot should look something like that (*but not necessarily exactly like that!*):



Task 10.

Use subplots to show choropleth maps for all four price columns (**all plots in one row**). Place titles on each plot. Pick your own colormap. Your plot should look something like that (*but not necessarily exactly like that!*):



Email me your project (as an attached Jupyter Notebook ipynb file which should be named **YourName_Project2.ipynb**). Make the subject of the e-mail exactly **CS210 Project2**

ADDITIONAL INFORMATION:

Some additional information may be added here,
if needed. Check frequently.

1. ...