Lab2-Python

Operating Systems

Claudio Maiorana - Carlos Castillo

Githubs: CMaio, carloscastillor

2020-2021

12 March 2021

Part 2: Selected Data Set

1. What it the purpose of the data set

The purpose of the data set is to have in one document all the information about catalunya and all the comarcas and provincies of it and all the centers with the information of how many persons can enter to the center and all the levels of every center.

2. Which is the information you have on each row of the data set

The information we have in the data set is:

- Curs: Curs de referencia
- Codi centre: Codi del centre
- Denominació completa: Nom complet del centre
- Codi naturalesa: Codi públic/privat
- Nom naturalesa: Públic/Privat
- Codi titularitat: Codi de la propietat del centre
- Nom titularitat: Nom de la propietat del centre
- Codi delegació: Codi territorial del centre
- Nom delegació: Nom territorial del centre
- Codi comarca: Codi de la comarca
- Nom comarca: Nom de la comarca
- Codi municipi_5: Codi del municipi a 4 digits
- Codi municipi_6: Codi del municipi a 5 digits
- Nom municipi: Nom del municipi
- Codi districte municipal: Codi districte municipal del centre educatiu
- Nom DM: Nom del districte del centre educatiu
- Coordenades UTM X: Coordenada X expressada en metres
- Coordenades UTM Y: Coordenada Y expressada en metres
- Longitud: Longitud expressada en graus decimals.
- Latitud: Latitud expressada en graus decimals.
- Nom ensenyament: Nom de l'ensenyament.
- Nivell: Nivell de l'ensenyament
- Nombre grups: Quantitat de grups
- Mixt: Si el grup es mixt o no
- Nombre places: Nombre de places
- Places ofertades a la preinscripció: Places ofertades per a fer la preinscripció
- Assignacions: Nombre d'assignacions
- Assignacions 1ª petició: Nombre d'assignacions a la primera petició
- Assignacions Altres peticions: Nombre assignacions a la resta de peticions
- Columna amb la georeferencia: Georeferencia del centre

3. How can the dataset be exported. List of the formats and explained all the formats the dataset can be exported to. How the information is represented in the exported file format.
    - CSV: In this file, the data is represented in lines, for each line all the information of a center with a level is represented, divided by “,”.
    - CSV for Excel: The representation is the same as CSV.
    - RDF: Here the data is represented in parts. There is a line with the id of this part and then a subpart with a line for each column named in the question before.
    - TSV for Excel: This file is represented in the same way of the CSV but in this case the file is divided by spaces
    - XML: The XML file is similar to the RDF, for every center has an id so if you have this id you can directly access information about this center using this id. Also, the XML file has an id position, so if you know the correct position of the center you want to see it’s easier to find it.
    - RSS: For every center we have, first of all, the name for every column named before as an index of the parts the center has and then in the same order we have the lines with the continent of the center.

Part 3: Process selected Data Set

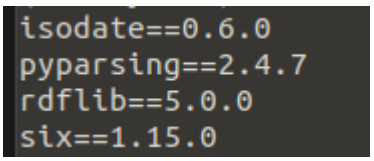
1. How the environment has been saved

In order to save the environment we typed in the terminal the following command “pip3 freeze > requeriments.txt”. This command generates a .txt file with all the packages that we used in the lab

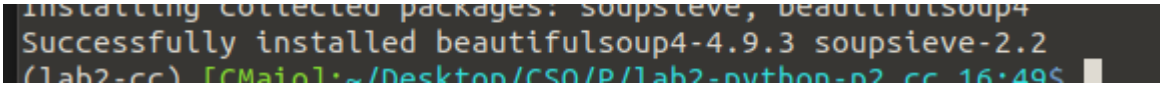
2. How the environment must be imported

In order to have all the libraries correctly working in your environment, you have to install the requirements.txt in your environment using the command pip3 install -r requeriments.txt being in the directory of the project.

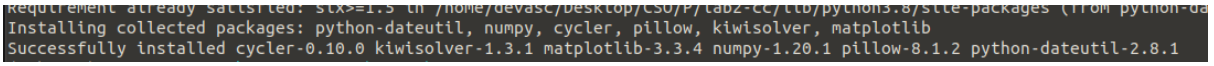
3. Packages



rdflib: This library is used to correctly read the data set about the information of the centers in Catalunya.



beautifulsoup4: Used to read correctly a xls file used to know the correspondent Provincia for each Comarca.



matplotlib:Used to show the images of the places or the histogram of the centers when selecting some options.

geopandas:Used to read a geojson file and later create a map with the help of matplotlib.
pandas: Used to create the data frames

4. Structure program

```
doc = lightrdf.RDFDocument("rows.rdf")
with open("ComarquesProvincies.xls") as fp:
    soup = BeautifulSoup(fp, "html.parser")
```

```
def dos(self):
    listGeneral = []
    for s, p, o in doc.search_triples(None, "https://analisi.transparenciacatalunya.cat/resource/_99md-r3rq/nom_comarca", None):
        if fileName in o:
            b = doc.search_triples(s, "https://analisi.transparenciacatalunya.cat/resource/_99md-r3rq/denominaci_completa", None)
            for item1, item2, item3 in b:
                result1=item3.replace('"',"")

            d = doc.search_triples(s, "https://analisi.transparenciacatalunya.cat/resource/_99md-r3rq/oferta_inicial_places", None)
            for item1, item2, item3 in d:
               result2= item3.replace('"',"")
```

Here we access the doc variable that has the dataset with all the information, and we use a for to look at the concrete comarca we want. When we find it, we know the correct id, so now we can add this to the first position of the search_triples and only search the things we need in this comarca.

```
natalidad = "MapaCatProvincies.geojson"
map_data = gpd.read_file(natalidad)
```

Here we use gdp(geopandas) to read the geojson with the coordinates of Catalunya to draw the image.

```
# Mostrar el mapa finalizado
map_data.plot(cmap='Greens', ax=ax,legend=True, zorder=5)
plt.scatter(geox,geoy, s=100,c='red',zorder=6)
plt.axis([geox - 0.3, geox+0.3, geoy-0.3, geoy+0.3])

plt.show()
```

In this screenshot we can see that we use .plot to create the map of catalunya using the geojson file with a color map of greens, then use the plt.scatter to mark points using the geox and geoy to set the position of the points and the other parameters for the size, color and order with the map.
The last line is for setting the middle of the map to make a zoom to where the point is.

```
for item1, item2, item3 in h:
    geox.append(float(item3.replace('"',"")))
```

```
for item1, item2, item3 in c:
    geoy.append(float(item3.replace('"',"")))
```

```
plt.scatter(geox,geoy, s=10,c='red',zorder=6)
plt.axis([geox[0] - 0.2, geox[0]+0.2, geoy[0]-0.2, geoy[0]+0.2])
```

Here we can see that we use and append to add values to geox and geoy, in this part, one that is different to the code just above, we added a lot of points and then the scatter understands that there are multiple points and by itself added to the map. later we took the coordinates in the first point to set it as the middle of the map we want to show.

```
plt.bar(x, y)
plt.xticks(rotation=90)
plt.title('Ofertas per comarca')
plt.xlabel('Comarca')
plt.ylabel('Oferta')

plt.show()
```

This part is used to create an histogram, bar() to say the data we are gonna use, xticks() to rotate the names and be able to read it a little bit better and the rest to set titles.
The last line is used to show to the user the histogram perfectly done.

```
paintTable=[]
def addHistory(comar,ofer,assis):
    esta=False
    for n in range(0,len(paintTable)):
        if esta==False:
            if paintTable[n][0] == comar:
                paintTable[n][1]+=ofer
                paintTable[n][2]+=assis
                paintTable[n][3]=0
                esta=True

    if esta == False:
        paintTable.append([comar,ofer,assis,0])
```

To make the colormap for the option5, we recollect and add all the information we need, from all the comarques, to a table using the method addHistory.

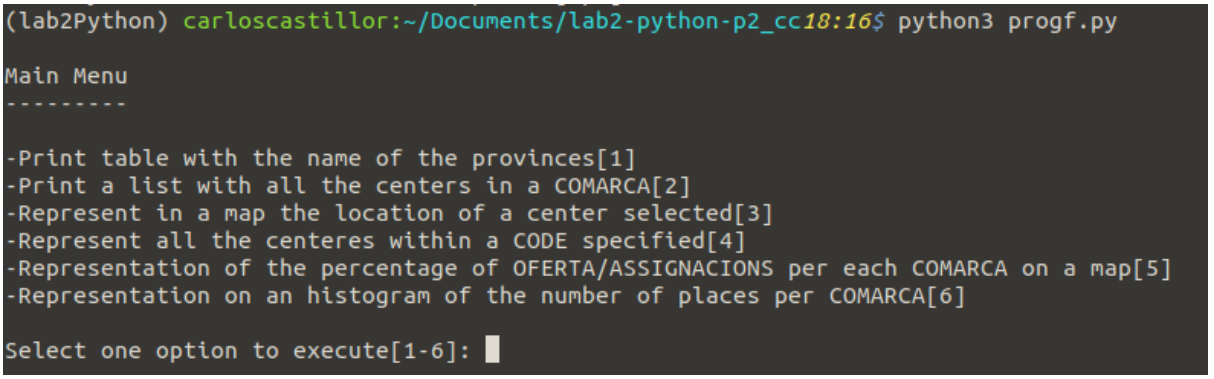
```
def cinco(self):

    percentageComarca = []
    for s, p, o in doc.search_triples(None, "https://analisi.transparenciacatalunya.cat/resource/_99md-r3rq/nom_comarca", None):
        result = o.replace('"',"")
        h = doc.search_triples(s, "https://analisi.transparenciacatalunya.cat/resource/_99md-r3rq/places_ofertades_a_la", None)
        for item1, item2, item3 in h:
            ofer =int(item3.replace('"',""))

        c = doc.search_triples(s, "https://analisi.transparenciacatalunya.cat/resource/_99md-r3rq/assignacions", None)
        for item1, item2, item3 in c:
            assi =int(item3.replace('"',""))
        addHistory(result,ofer,assi)

    for i in range(0,len(paintTable)):
        if paintTable[i][2] == 0:
            percentageComarca.append({"nom_comar": paintTable[i][0], "percentage": 0})
        else:
            percentageComarca.append({"nom_comar": paintTable[i][0], "percentage": paintTable[i][1]/paintTable[i][2]})

    data_df = pd.DataFrame(percentageComarca)

    natalidad = "MapaCatProvincies.geojson"
    map_data = gpd.read_file(natalidad)

    dataMapMerge = pd.merge(map_data, data_df, how = 'left', on = 'nom_comar')
    fig, ax = plt.subplots(figsize=(10, 10))

    ax.set_title('percentatge OFERTA/ASSIGNACIONS',
                 pad = 20,
                 fontdict={'fontsize':20, 'color': '#4873ab'})
    ax.set_xlabel('Longitud')
    ax.set_ylabel('Latitud')

    dataMapMerge.plot(dataMapMerge["percentage"], ax=ax, legend=True, zorder=5)

    plt.show()
```

Then we access this table to create a dictionary (percentageComarca) that relates each name of the cormarca with its percentage of (Offers/Assignations) in order to create a dataframe (data_df) from that dictionary.
Once we have that, we read the coordinates as it's explained before and merge the information of the coordinates (map_data) and the data frame with the percentage information (data_df) to generate a new data frame (dataMapMerge) combining those two by its row "nom_comar" which are all the comarques names.
To finally generate the map, we set some titles and labels, and then make a plot using the dataMapMerge. The first parameter(dataMapMerge["percentage"]) is just the information about the percentage of every comarca, so matplotlib understands that the color of the comarca has to be related with the values of percentage to choose which color to use.

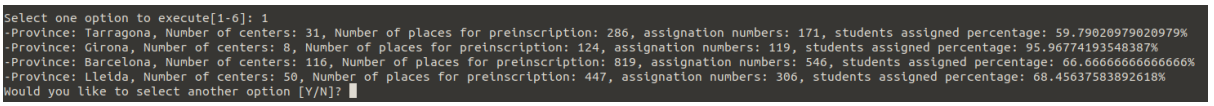
5. Screenshots execution program

You must know that if an option generates an image, such as a map or histogram, it will take some time to generate it, especially in option 5, that option will take a couple minutes. Also when the image is generated, in order to keep using the program, you must close the image or you won't be able to select a new option.

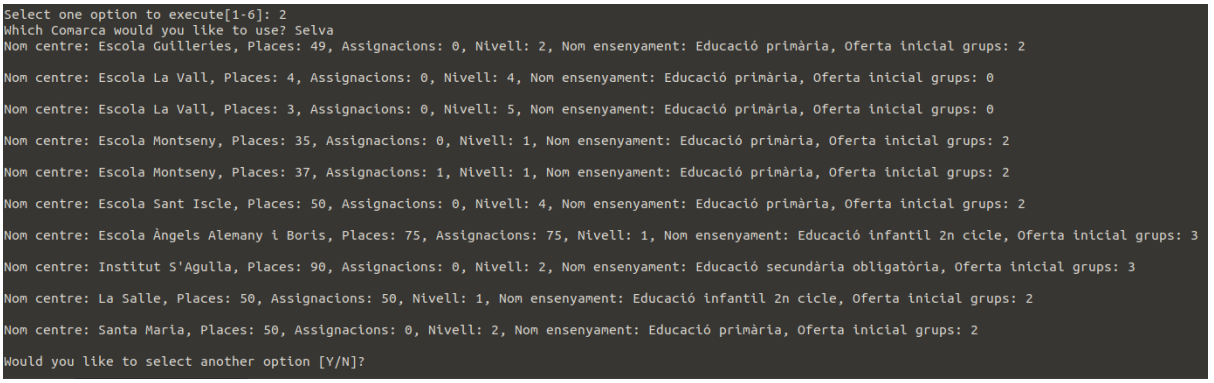
Main menu to select option:



Option 1:

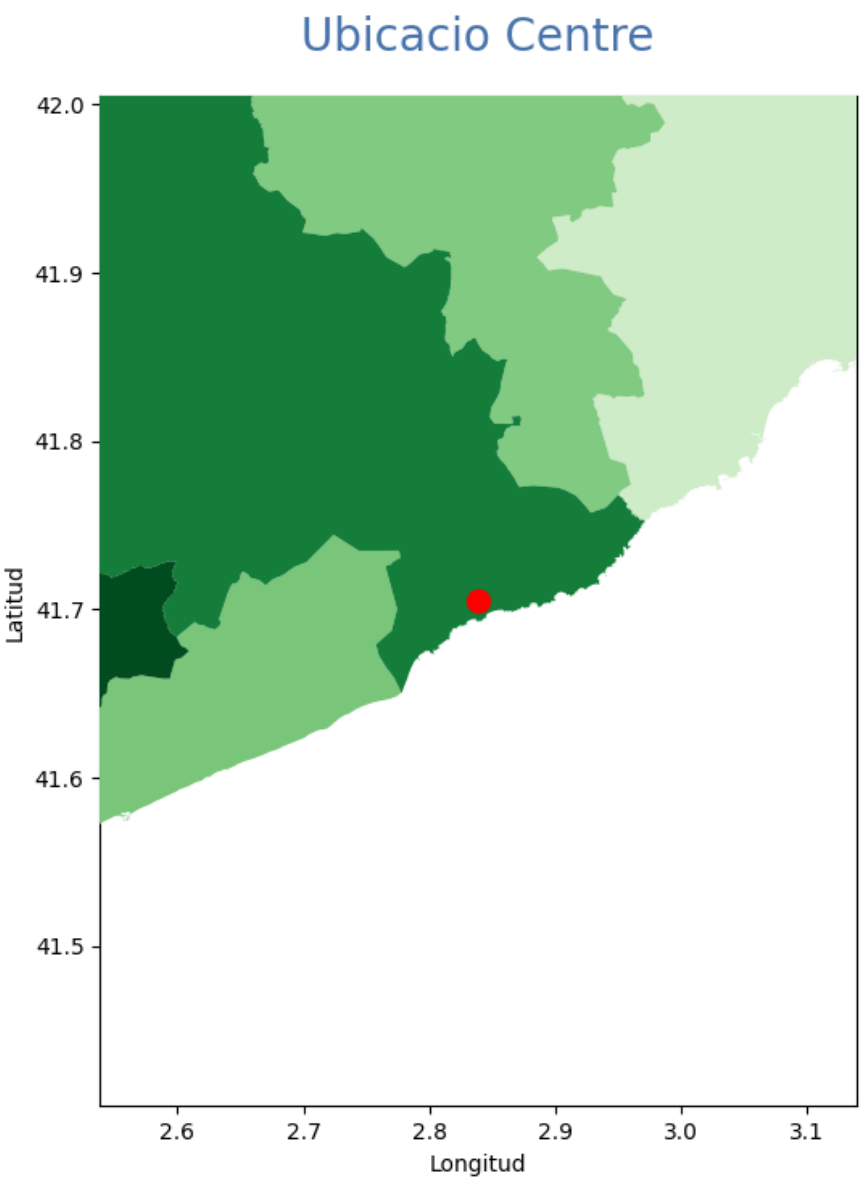


Option 2:



Option 3:

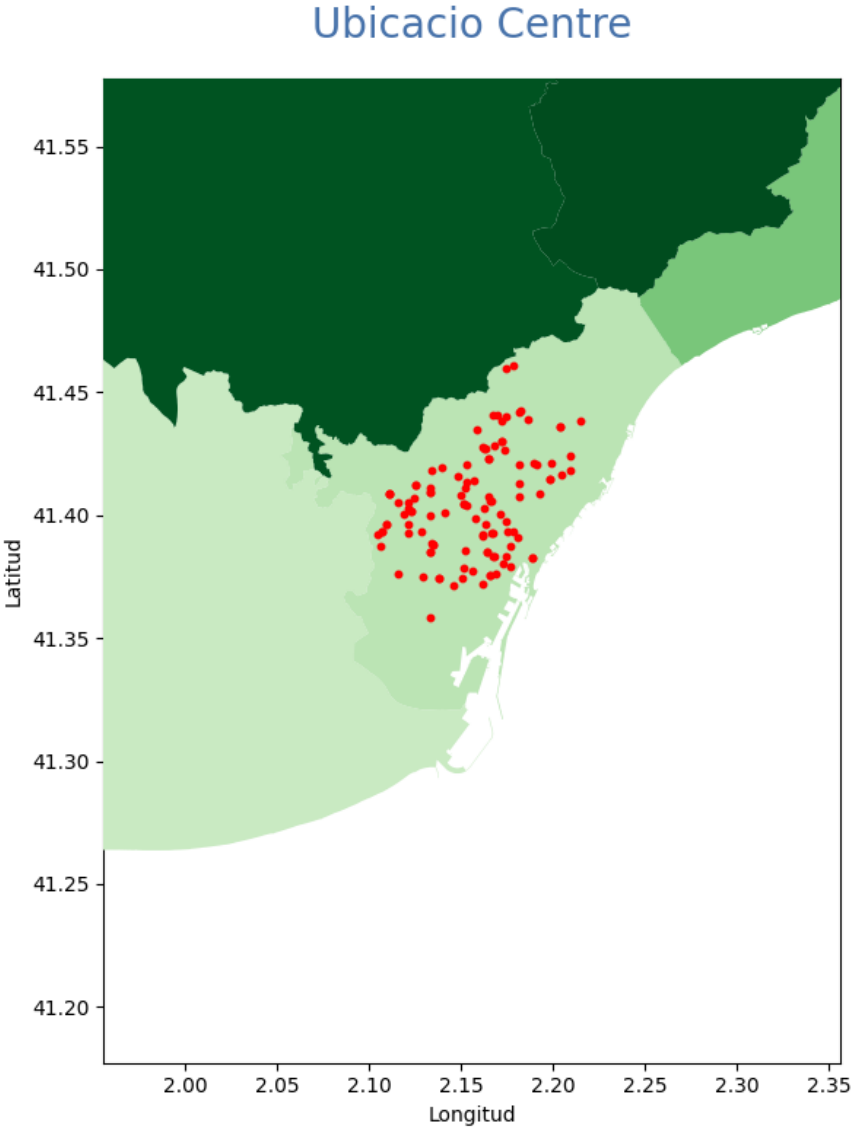
```
Select one option to execute[1-6]: 3
Which Comarca would you like to use? Selva
\Municipis:
['Breda', 'Osor', 'Breda', 'Blanes', 'Vidreres', 'Osor', 'Lloret de Mar', 'Blanes', 'Santa Coloma de Farners', 'Sant Hilari Sacalm']

Select a Municipi to see all its centers: Lloret de Mar

Centers:
['Escola Àngels Alemany i Boris']

Select one center: Escola Àngels Alemany i Boris
```



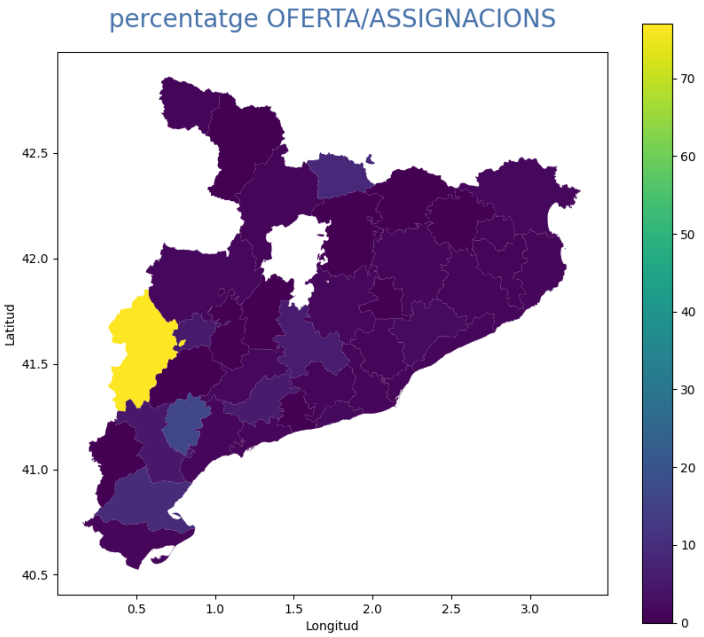
Option 4:

In this option you must use a valid municipi code with 4 digits. The row that is used, to have a reference for this code, is “codi_municipi_5”.

```
Select one option to execute[1-6]: 4
Which Code would you like to use? 9000

This code is not a valid one.
Would you like to select another option [Y/N]? Y
Select one option to execute[1-6]: 4
Which Code would you like to use? 8019

Centers:
['Salesià de Sant Josep', 'Claret', 'Escola Tibidabo', 'Sant Marc de Sarrià', 'Montserrat', 'San Rafael', 'Escola Joan Miró', 'Escola Castella', 'Abat Oliba-Loreto', 'Escola Montseny', 'Sagrat Cor', 'FEDAC-Amilc
ar', 'Escola Ferran Sunyer', 'Escola Vila Olímpica', 'Sagrada Família', 'Sant Felip Neri', 'Escola Ellsenda de Montcada', 'Decroly', 'Escolàpies Sant Martí - Barcelona', 'Escola Pere Vila', 'Institut Jaume Balme
s', 'Canigó', 'Escola Duran i Bas', 'Mare del Diví Pastor', 'Mare de Déu dels Àngels', 'Escola Font d'en Fargas', 'Vedruna Immaculada', 'Santa Teresa de Lisieux', 'Labouré', 'Santa Dorotea', 'Maristes Anna Ravel
l', 'Inmaculada Concepció Barcelona', 'Lestonnac', 'Escola Anglesola', 'Betània-Patmos', 'Escola Casas', 'Mirela', 'Escola Josep Maria de Sagarra', 'Sant Joan Baptista', 'Institut Manuel Carrasco i Formiguera',
'Escola Francesc Macià', 'Ramon Llull', 'Sant Pau', 'Shalom', 'Peter Pan', 'Mary Ward', 'Jesuïtes Gràcia - Kostka', 'Reina Elisenda', 'Escola Auró', 'Sant Gregori', 'Santa Teresa - Ganduxer', 'Escola Collaso i G
il', 'San Nicolás', 'Escola Cervantes', 'Sagrado Corazón', 'Sant Ramon Nonat', 'Escola Reina Violant-Comas i Solà', 'Institut Secretari Colonia', 'Escola Víctor Català', 'Regina Carmeli', 'Sant Ignasi', 'Escola C
arlit', 'Escola Dolors Monserdà-Santapau', 'Escola Turó del Cargol', 'Magòria', 'Escola Torrent de Can Carabassa', 'Vedruna Àngels', 'Jesús Maria', 'Mare Alfonsa Cavin', 'Cor de Maria-Sabastida', 'Institut Milà
i Fontanals', 'Escola Rius i Taulet', 'Bon Pastor', 'Escola Coves d'en Cimany', 'Escola Pia de Sarrià-Calassanç', 'Escola Patronat Domènech', 'Thau', 'Escola Marinada', 'Vedruna-Gràcia', 'Josep Tous', 'Escola He
ura', 'Escola Els Porxos', 'Sadako', 'Immaculada Concepció', 'Institut Lluís Domènech i Montaner', 'Sant Josep-Teresianes', "Dominiques de l'Ensenyament de Barcelona", "Princeps 23 d'Abril", 'Santa Maria dels Ap
òstols', 'Escola Ítaca', 'Sagrat Cor-Sarrià', 'Escola Brasil', 'Escola La Pau', 'Escola Drassanes', "Escola l'Univers", 'Escola El Sagrer', 'Grey']
```

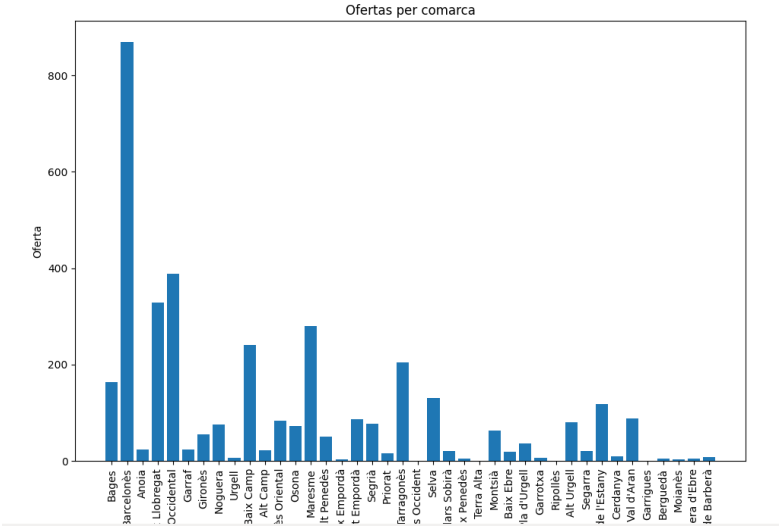


Option 5:



In that option, the comarques that are not colored, means that there is no data about their offers and assignations.

Option 6:



Proof that you can select a new option or exit after executing one option:

```
Would you like to select another option [Y/N]? Y
Select one option to execute[1-6]: 4
```

```
Would you like to select another option [Y/N]? N
(lab2Python) carloscastillor:~/Documents/lab2-python-p2_cc14:36$ pip3 freeze
attrs==20.3.0
```