WORKSHOP 1

MARÍA FERNANDA TELLO VERGARA

2225338

JAVIER ALEJANDRO VERGARA

ETL

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE

AGOSTO 28 2024

Contexto

We are given a csv file called Candidates, in which we find a total of 50000 records which include the following information:

- First Name: Name of candidate
- Last Name: Last name
- Email: Candidate's email
- Application Date: Application date
- Country: Country of candidate
- YOE: Candidate's years of experience
- Seniority: Seniority level of the candidate
- Technology: Technology related to the position for which the candidate applied
- Code Challenge Score: Code challenge score
- Technical Interview Score: Technical interview score

You have to read the csv file with python and migrate this table to our database, since here we connect from python to the database to read the cvs and make the modification that is asked, which is to clone the original table and add it a new field called "Hired" which will have as unique values YES or NO depending if the record meets the following restrictions (All this process has to be done with python connected to our database).

The value in the Code Challenge Score field must be greater than or equal to 7, and the same restriction for the Technical Interview Score field, if the record only meets one of these restrictions, its value in the record will be "NO" and if the record complies with the two restrictions the value of the record will be "YES".

Hired: Indicates whether the candidate was hired or not.

Description

We are going to do the Exploratory Data Analysis (EDA) process to identify the columns, find and clean outliers, understand the context and data to gain insights and know how to handle the data, get value from the data and finally make the following visualizations.
• Hires by technology (pie chart)
• Hires by year (horizontal bar chart)
• Hires by seniority (bar chart)
• Hires by country over the years (US, Brazil, Colombia and Ecuador only) (multi-line chart).

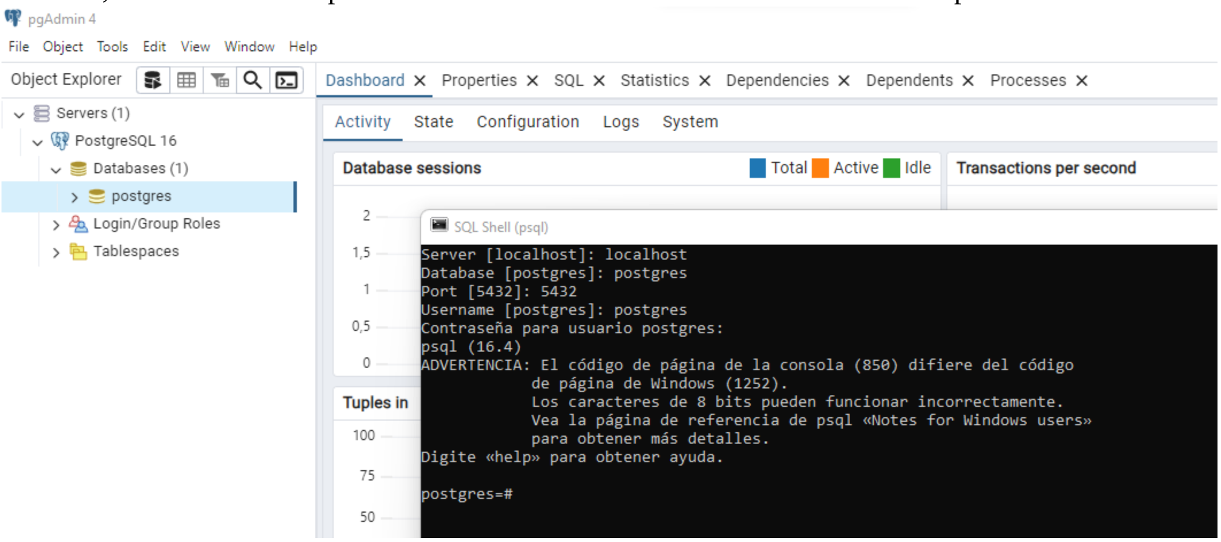
Tools

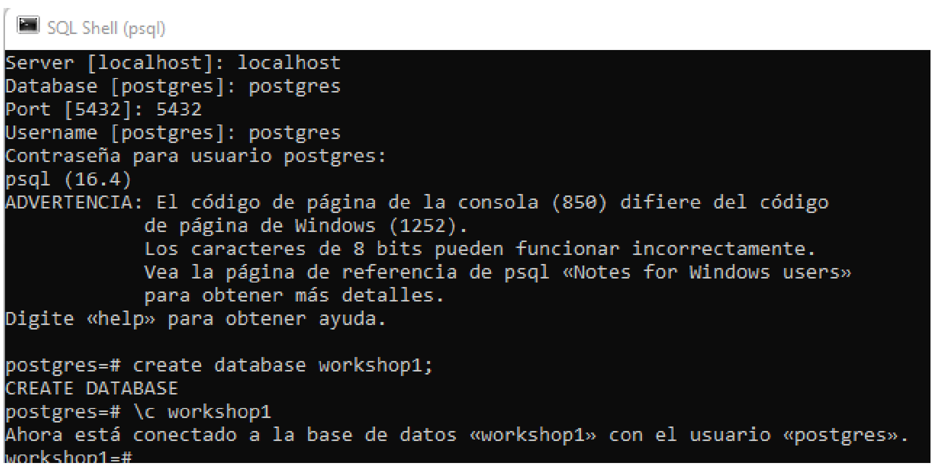
• Python (Pandas, Matplotlib, Sqlalchemy dotenv).
• Postgress.
• Jupyter Notebook.
• Dataset (Candidates).
• Encryption of credentials using a .env file (Environment variables).

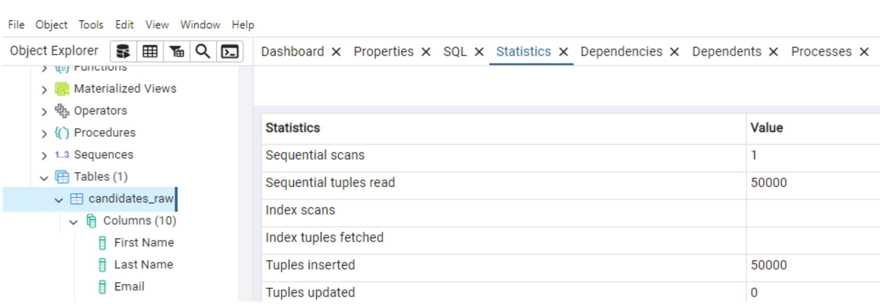
Step by step

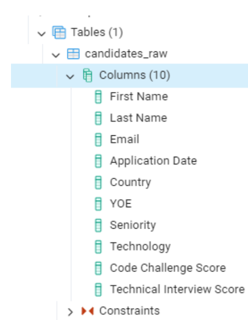
Installing Postgres and Creating the Database:

We install PostgreSQL, which we will use to manage the data in this workshop. Then we create the database, which will be the place where the exercise data will be stored and manipulated.



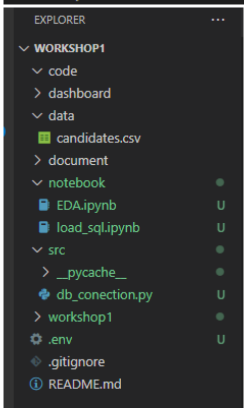






```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> python
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun  6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
KeyboardInterrupt
>>> exit()
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> pip install ipykernel
Collecting ipykernel
  Downloading ipykernel-6.29.5-py3-none-any.whl.metadata (6.3 kB)
Collecting comm>=0.1.1 (from ipykernel)
  Downloading comm-0.2.2-py3-none-any.whl.metadata (3.7 kB)
Collecting debugpy>=1.6.5 (from ipykernel)
  Downloading debugpy-1.8.5-cp312-cp312-win_amd64.whl.metadata (1.1 kB)
Collecting ipython>=7.23.1 (from ipykernel)
  Downloading ipython-8.26.0-py3-none-any.whl.metadata (5.0 kB)
Collecting jupyter-client>=6.1.12 (from ipykernel)
  Downloading jupyter_client-8.6.2-py3-none-any.whl.metadata (8.3 kB)
```

```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> python -m venv
 workshop1
```



Viewing file and folder management.

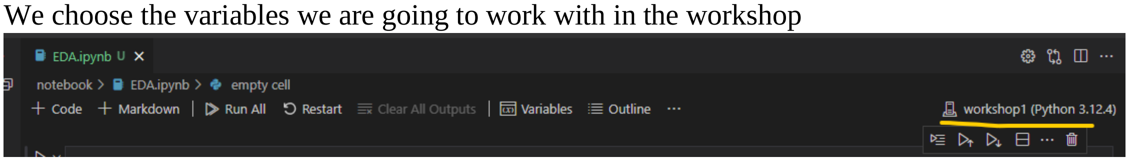
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   JUPYTER                                  powershell
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> .\workshop1\Scripts\activate

(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1>
```

In the visual we can see this image where the command to create a Python kernel in Jupyter Notebook called 'workshop1' is shown. This command configures a specific virtual environment for the workshop.

Command: Python -m ipykernel install --user --name 'workshop1' --display -name 'workshop1'

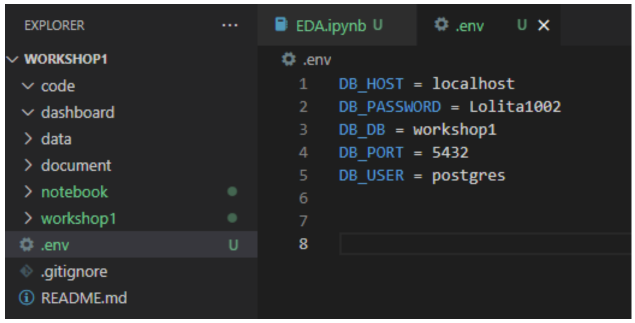
```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> .\workshop1\Scripts\activate
(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> pip install ipyk
ernel
Collecting ipykernel
  Using cached ipykernel-6.29.5-py3-none-any.whl.metadata (6.3 kB)
```

We choose the variables we are going to work with in the workshop

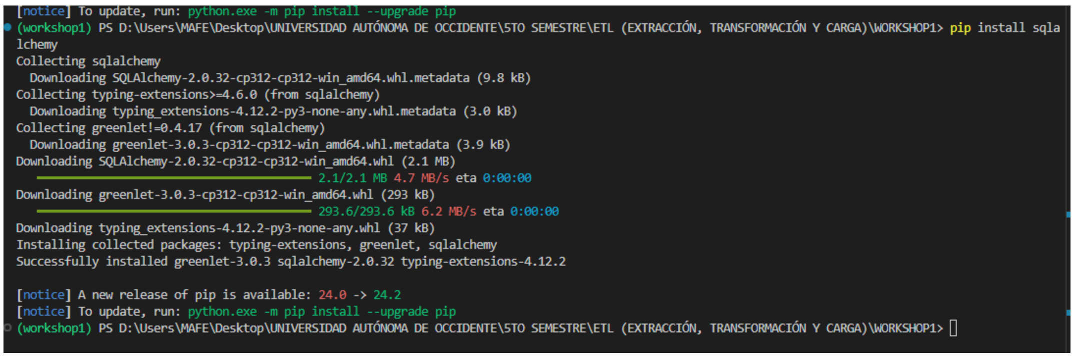


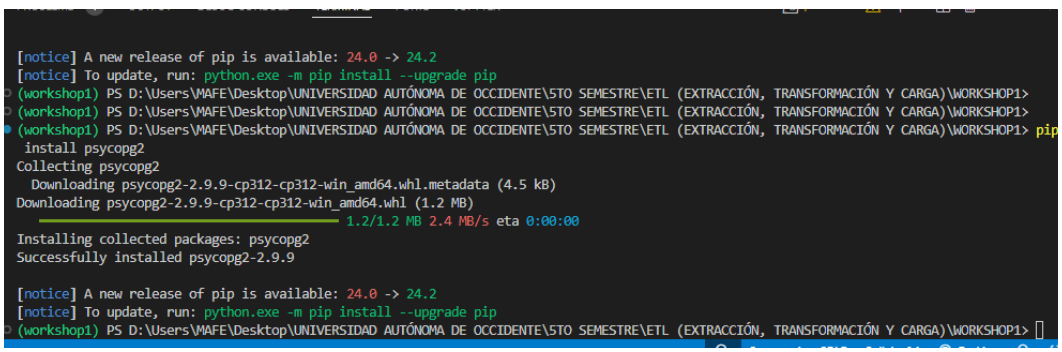
Here we can see the selection of variables that will be worked with during the workshop, as well as the installation of necessary libraries, such as pandas, which are crucial for data manipulation.

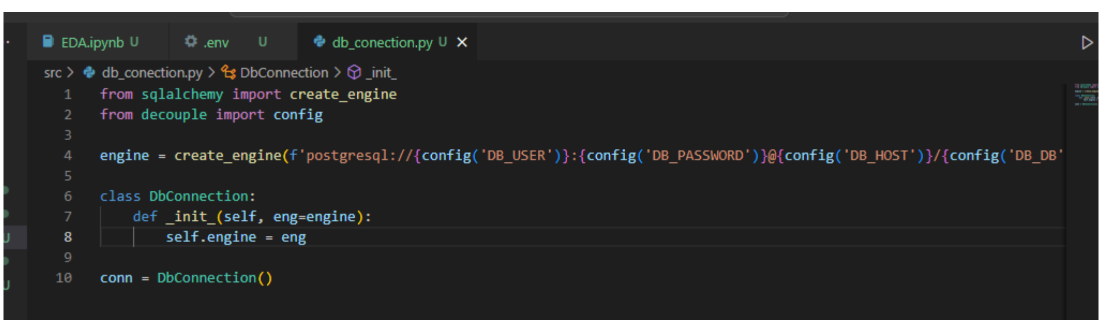
Here is the code and process to establish a connection to the PostgreSQL database from Python, using libraries such as psycopg2.



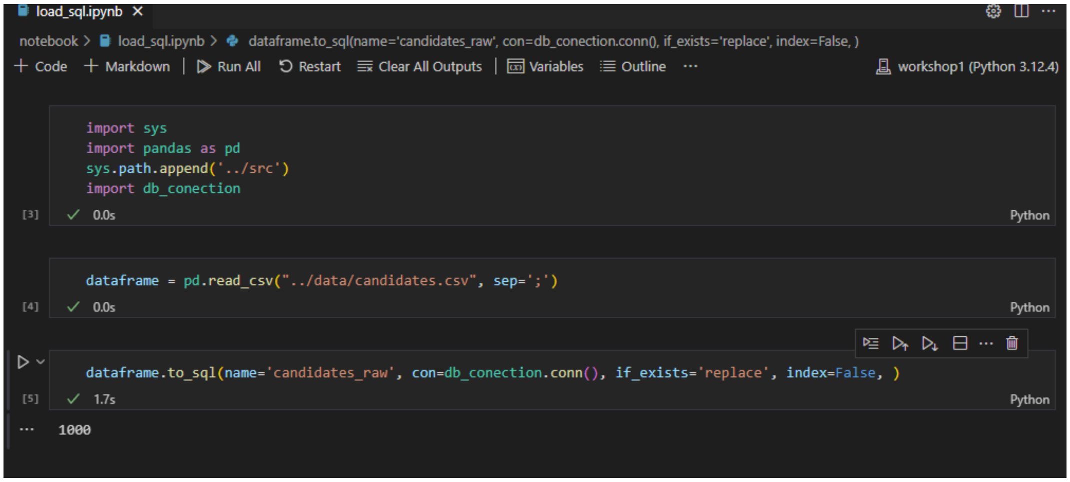
Create the database connection.

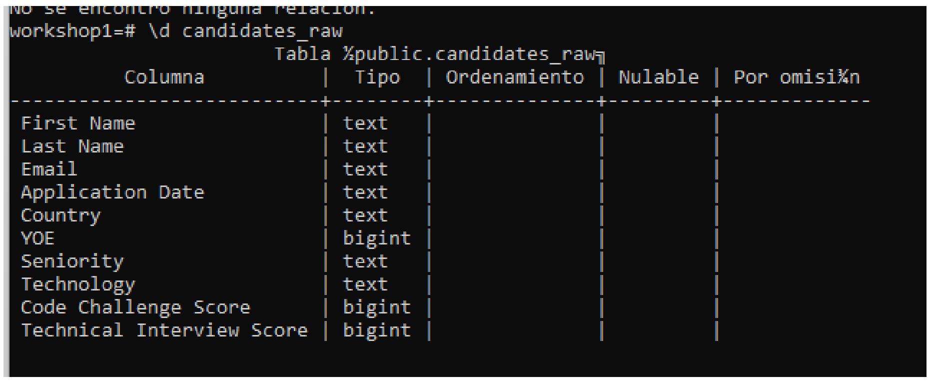






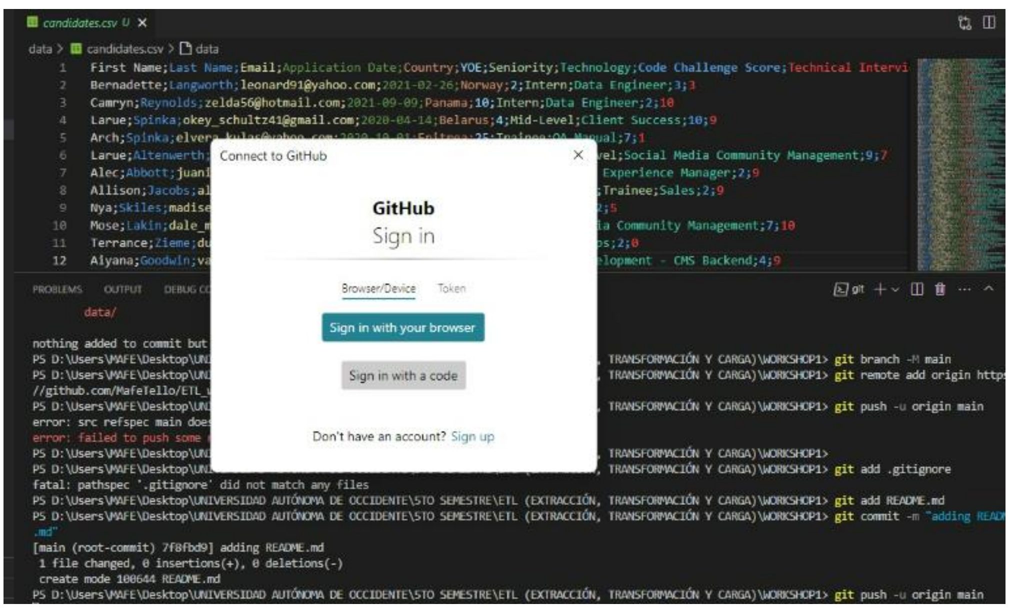
The image shows how to load and read data from a (dirty) CSV file into a pandas DataFrame.





We can see that we started the Exploratory Data Analysis (EDA) using pandas to inspect and clean the data.

Git Hub Connection



We then run git status to show the current status of the work area and staging area in the repository.

```
[notice] To update, run: python.exe -m pip install --upgrade pip
(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1>
(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1>
(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1>
(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .env
        notebook/
        src/
        workshop1/

nothing added to commit but untracked files present (use "git add" to track)
(workshop1) PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\WORKSHOP1>
```

EDA:

Reading the first data from the dataframe.

```
dataframe_raw.head()
```

0.0s

	First Name	Last Name	Email	Application Date	Country	YOE	Seniority	Technology	Code Challenge Score	Technical Interview Score
0	Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway	2	Intern	Data Engineer	3	3
1	Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama	10	Intern	Data Engineer	2	10
2	Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus	4	Mid-Level	Client Success	10	9
3	Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea	25	Trainee	QA Manual	7	1
4	Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar	13	Mid-Level	Social Media Community Management	9	7

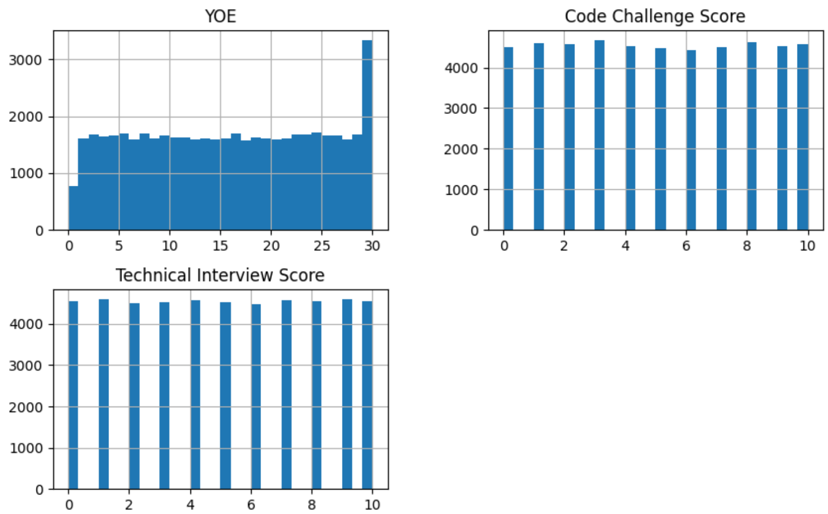
Basic description and descriptive statistics of the dataframe.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   First Name                 50000 non-null  object
 1   Last Name                  50000 non-null  object
 2   Email                      50000 non-null  object
 3   Application Date           50000 non-null  object
 4   Country                    50000 non-null  object
 5   YOE                        50000 non-null  int64
 6   Seniority                  50000 non-null  object
 7   Technology                 50000 non-null  object
 8   Code Challenge Score       50000 non-null  int64
 9   Technical Interview Score  50000 non-null  int64
dtypes: int64(3), object(7)
memory usage: 3.8+ MB
First Name: 3007 unique values
Last Name: 474 unique values
Email: 49833 unique values
Application Date: 1646 unique values
Country: 244 unique values
YOE: 31 unique values
Seniority: 7 unique values
Technology: 24 unique values
Code Challenge Score: 11 unique values
Technical Interview Score: 11 unique values
```

Checking for null data

```
0.0s

First Name                  0
Last Name                   0
Email                       0
Application Date            0
Country                     0
YOE                         0
Seniority                   0
Technology                  0
Code Challenge Score        0
Technical Interview Score   0
dtype: int64
Duplicated rows: 0
```

Visualization of the distribution of the variables.



Creation of the new "hired" column.

```
0.0s

   Code Challenge Score  Technical Interview Score  hired
0                     3                          3      0
1                     2                         10      0
2                    10                          9      1
3                     7                          1      0
4                     9                          7      1
```

After performing the cleaning and EDA, we can see that we now have a new dataset "candidates_hired" with the cleaned and loaded data. A new dataframe is created which is a copy of the original table with a new field hired.

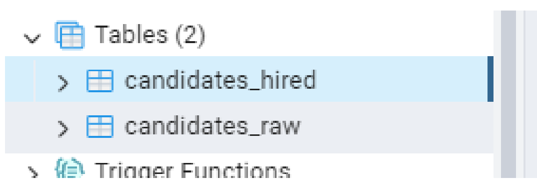
```
#cargar los datos en postgres y leer de nuevo
dataframe_raw.to_sql("candidates_hired" ,con = db_conection.conn(), if_exists="replace", index=False) #load into postgres
```
✓ 1.9s

```
1000
```

```
get_all_data_hired = f"SELECT * FROM candidates_hired" #The query to get the data for the pd dataframe
candidates_hired = pd.read_sql(get_all_data_hired,con = db_conection.conn())
```
✓ 0.3s

```
candidates_hired
```
✓ 0.0s

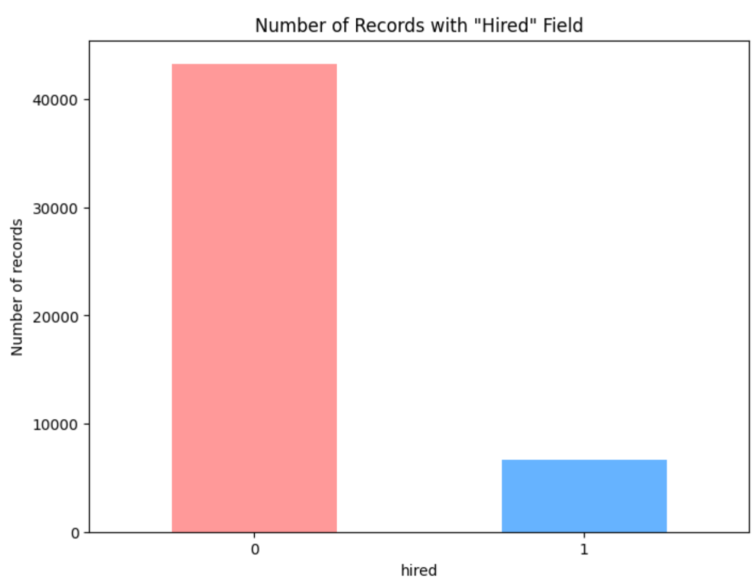
	First Name	Last Name	Email	Application Date	Country	YOE	Seniority	Technology	Code C
0	Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway	2	Intern	Data Engineer	
1	Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama	10	Intern	Data Engineer	
2	Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus	4	Mid-Level	Client Success	
3	Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea	25	Trainee	QA Manual	
4	Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar	13	Mid-Level	Social Media Community Management	
...	...	...	...	...	...	...	...	...	
49995	Bethany	Shields	rocky_mitchell@hotmail.com	2022-01-09	Dominican Republic	27	Trainee	Security	
49996	Era	Swaniawski	dolores.roob@hotmail.com	2020-06-02	Morocco	21	Lead	Game Development	
49997	Martin	Lakin	savanah.stracke@gmail.com	2018-12-15	Uganda	20	Trainee	System Administration	
49998	Aliya	Abernathy	vivienne.fritsch@yahoo.com	2020-05-30	Czech Republic	20	Senior	Database Administration	
49999	Coleman	Wisozk	abigayle.crooks@yahoo.com	2022-06-13	Palau	15	Intern	Mulesoft	



We proceed to analyze the visualizations.

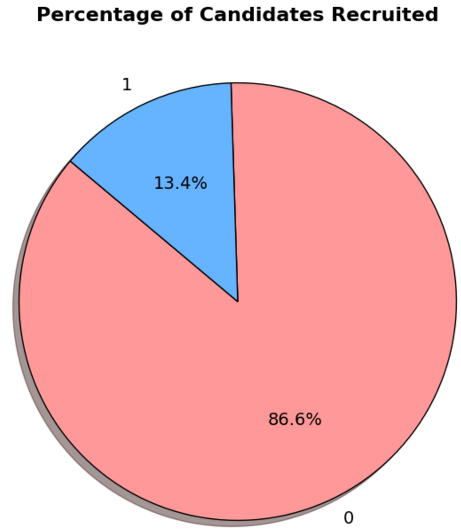
With this graphic we can give an idea of the difference between the candidates who were approved and those who were not in our dataset.

hired vs no hired



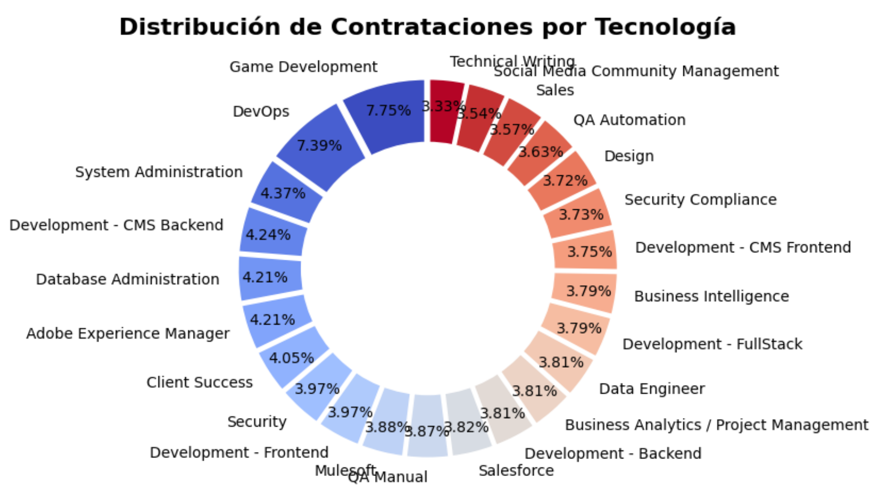
Analysis of the percentage of hires who were hired according to the new dataframe.

It is interesting to see how many total records only 13.40% of all those records met the requirements to be hired.



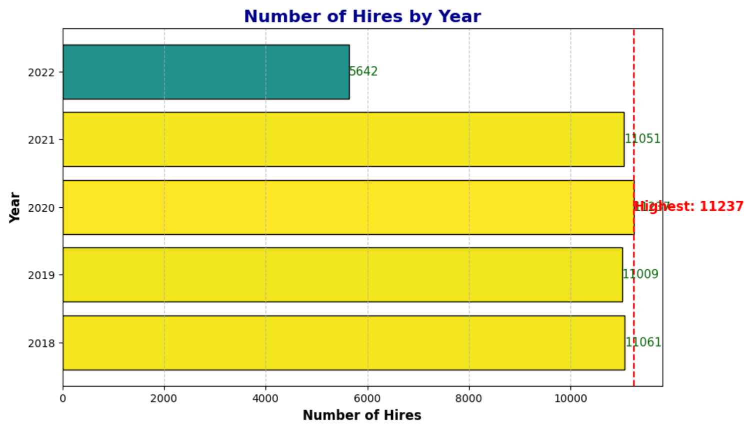
Hires by technology (pie chart)

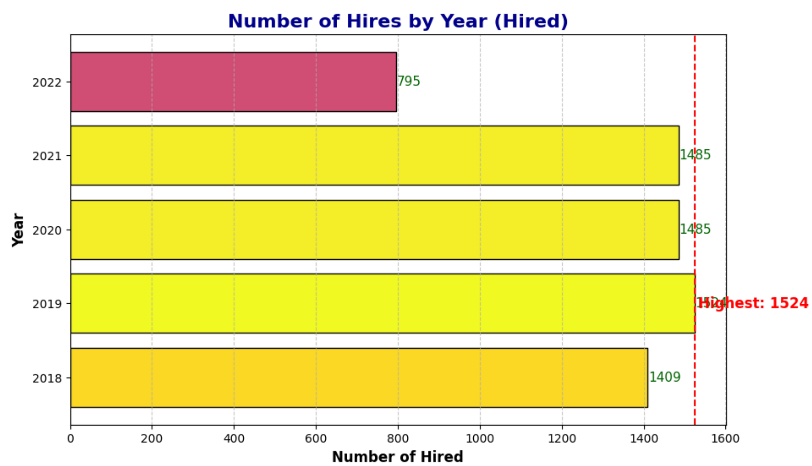
We can see the distribution of hires by technology, highlighting which areas are in higher or lower demand. The most represented technologies indicate current priorities, while the least represented ones could indicate opportunities to improve recruiting or training.



Hires by year (horizontal bar chart)

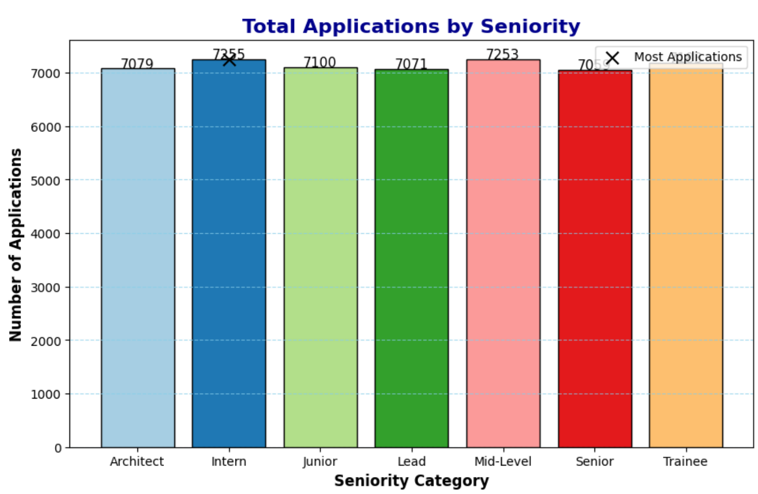
We can see the number of hires made in each year over time. The horizontal bars make it easy to compare the volume of hires year by year.

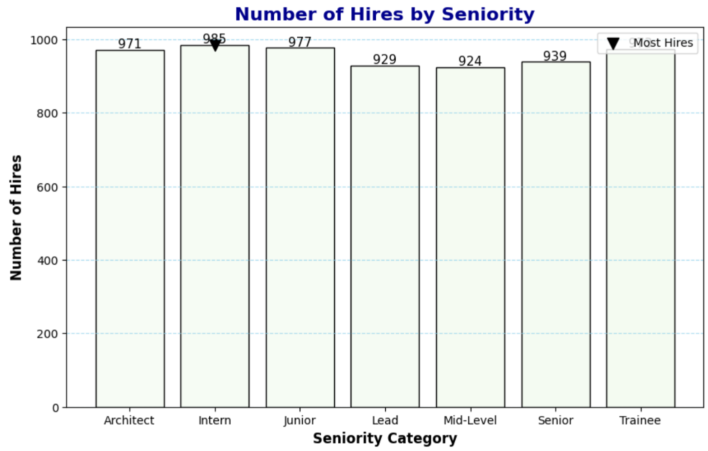




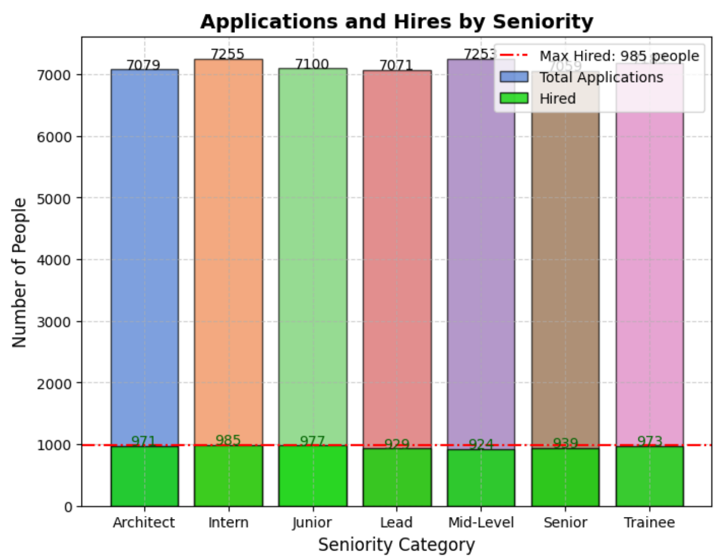
Hires by seniority (bar chart)

shows the distribution of hires according to the level of seniority or seniority of the candidates, with each bar representing a different level of seniority.

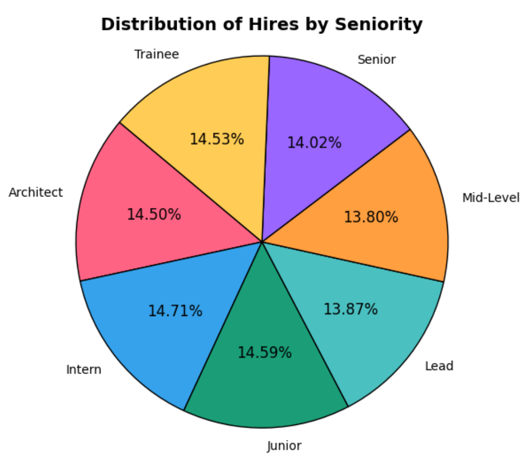




This chart shows a comparison between the number of applications and the number of hires according to the seniority level of the candidates.

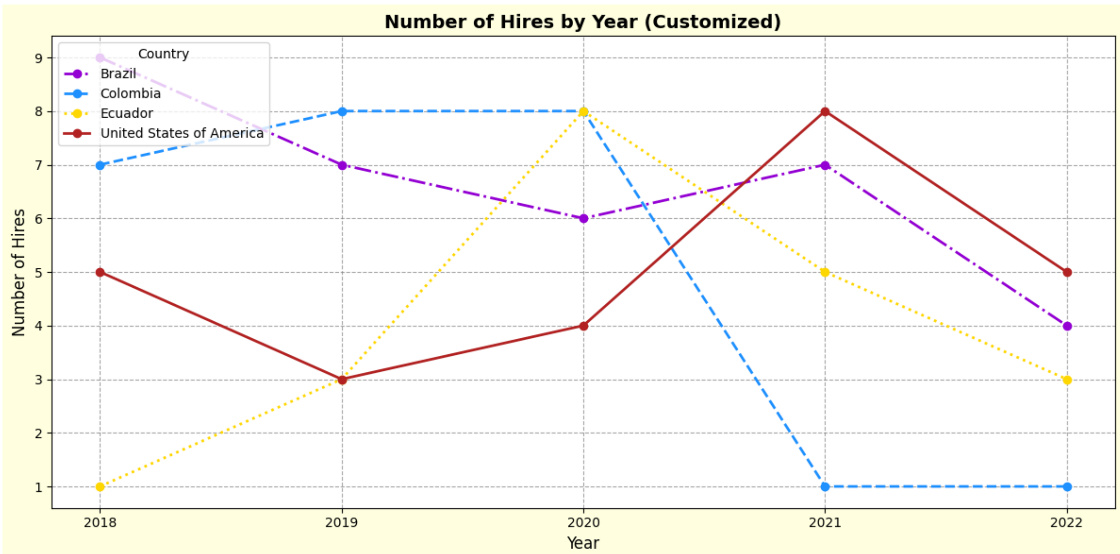


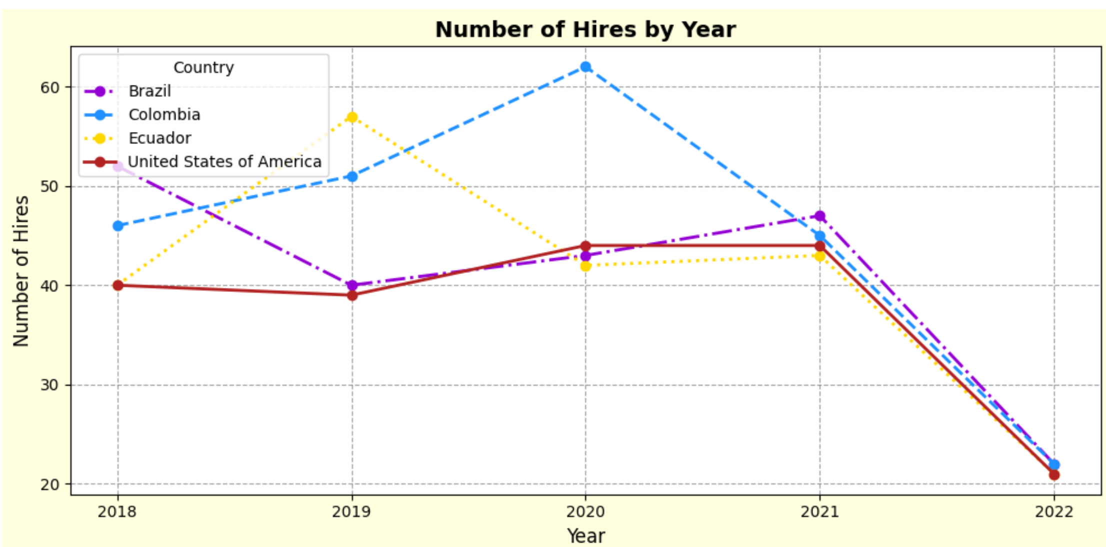
This chart shows how hires are distributed across different seniority levels.



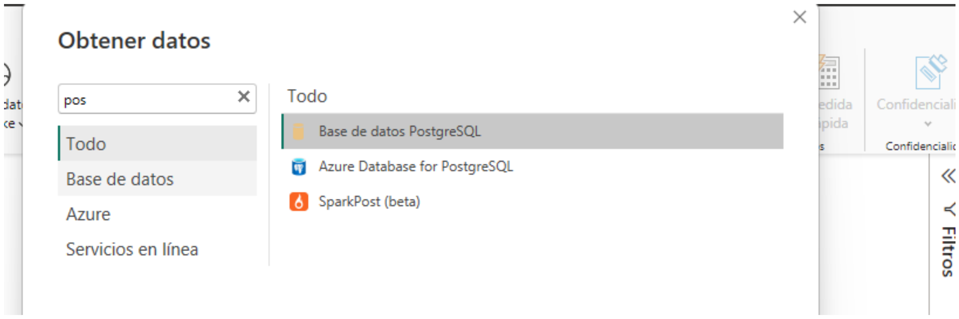
Hires by country over years (USA, Brazil, Colombia and Ecuador only) (multiline chart)

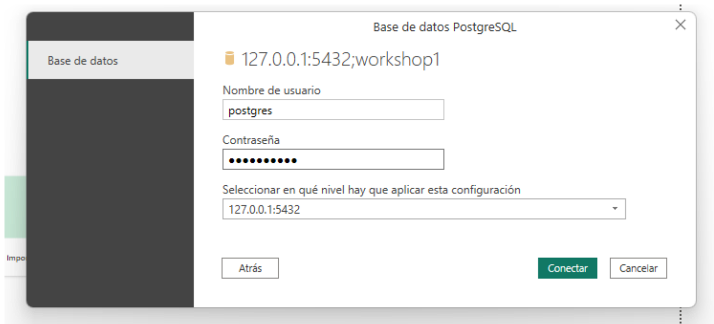
These multi-line graphs show the evolution of hiring over the years in four specific countries: USA, Brazil, Colombia and Ecuador. Each line represents a country, allowing for comparison of hiring trends in each one.





We make the connection of Power BI with the database.





Here are some of the questions we'll be answering in the Power BI dashboard:

• What is the most common seniority level among hired candidates?
Most hires are at the highest level, which occupies the largest segment of the chart.

• How do hires vary by year based on seniority?
This can be seen in the "Seniority Count by Year" chart. By looking at this chart, you can see how hires at different seniority levels have evolved over the years.

• What is the trend in hires by technology and year?
The "Sum of Hires by Year and Technology" bar chart shows how hires in different technologies have changed over the years.

• How do hires vary in different countries over time?
The "Hires by Year and Country" line chart shows the hiring trend in specific countries such as the US, Colombia, Brazil, and Ecuador over time.

• Which countries have the most hires in a specific technology?
The “Technology Count by Country and Hired” bar chart along with the technology filter, you can see which countries hire the most in a specific technology. For example, filter by “Game Development” to see which country has the most hires in that technology.

• What is the most commonly hired technology in the countries with the fewest hires?
You can filter by the countries with the fewest hires in the “Technology Count by Country and Hired” bar chart and look under “Hires by Technology” to see which technology is the most hired in those countries.

• How do hires vary from year to year at different seniority levels?
"Seniority Count by Year" - In this chart we can see if there is a particular year where more people have been hired at a specific level.

• How are hires for a specific technology distributed by country?
We used the technology filter in conjunction with the "Technology Count by Country and Hired" chart to see how hires for a specific technology are distributed across different countries.

Finally, we uploaded all the folders to the git hub repository.