💾

Workhsop 1

Student: Santiago Valencia Rosero, 2221658

About this project.

This is the repository in which I'll be showing the ETL process for the ETL's signature at Universidad Autonoma de Occidente. Here we will have to create the database, its tables as well as loading the data that has been modified from a CSV file according to the objectives of this excersice. This has been done in Jupyter notebooks.

Quick Data Overview

We have been requested to start load the dataset which has 50,000 rows and 11 columns, which are the next ones:

- First name, Last Name, Email, country, Application Date, Years of Experience (YOE), Seniority Level, Technology, Code Challenge Score, Technical Interview.

Objectives

This workshop aimed to develop an ETL process that involves creating a relational database from a Python script. The script is responsible for creating the tables and loading them with the data that has been transformed from the CSV file to the Postgres database.

The workshop also aimed to produce visualizations of the following queries using PowerBI:

- Hires by technology (pie chart).
- Hires by year (horizontal bar chart).

- Hires by seniority (bar chart).
- Hires by country over years (USA, Brazil, Colombia, and Ecuador only).

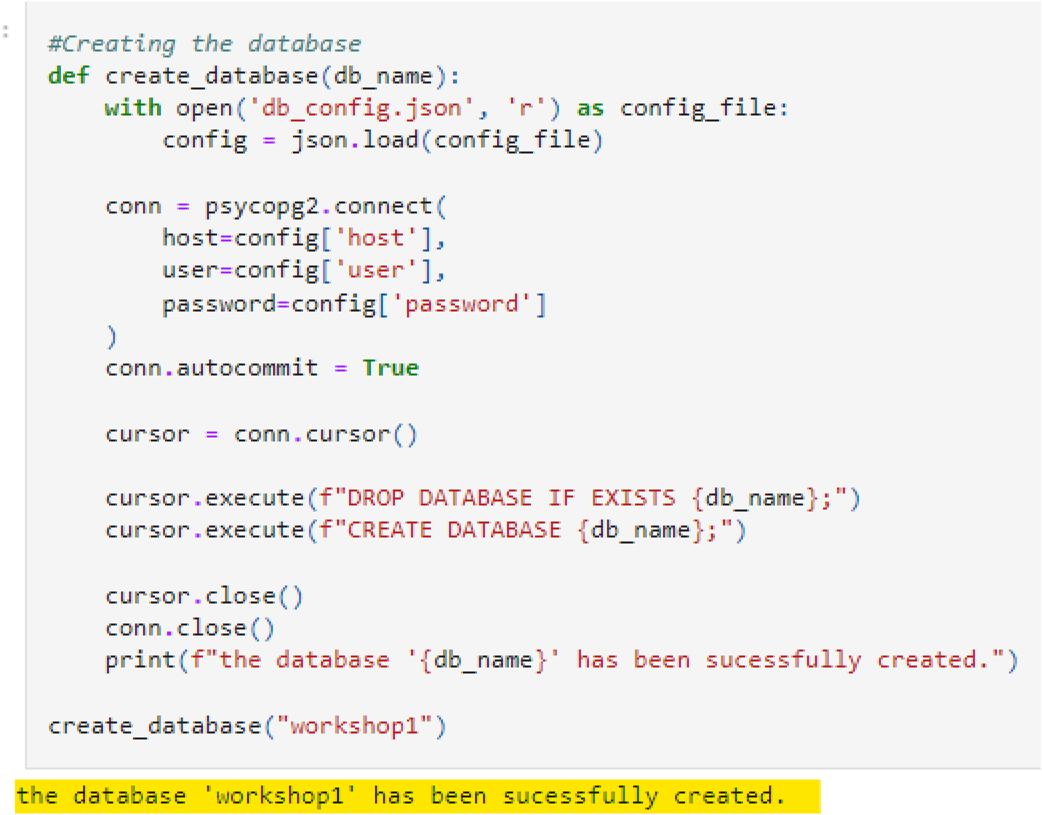
Process:

In this repository, I carried out the following tasks:

1. First, in the folder named "`loading_to_postgres`" there is a Jupyter notebook called "loading_data.ipynb":
    - In the "`loading_data.ipynb`" file, the first step is to create a connection to Postgres through the configuration file db_config.json, where we subsequently create a database to work with.
    - Another function is created to connect to the database, which will be used in the following functions.
    - A further function is created to create the tables within the database.
    - Finally, a function is created to read the CSV with the data and insert it into the previously created tables. When loading the data, the calculation is also made to determine which candidates were hired based on their scores in the "Code Challenge Score" & "Technical Interview Score" columns, where the score must be higher than 7 to consider that the candidate was hired. This way, it is added to the "hired_status" column.
2. Second, in the "notebooks" folder, we have a Jupyter notebook called "`EDA.ipynb`" where an exploratory analysis of the dataset is performed. This is done to get a first look at the data.

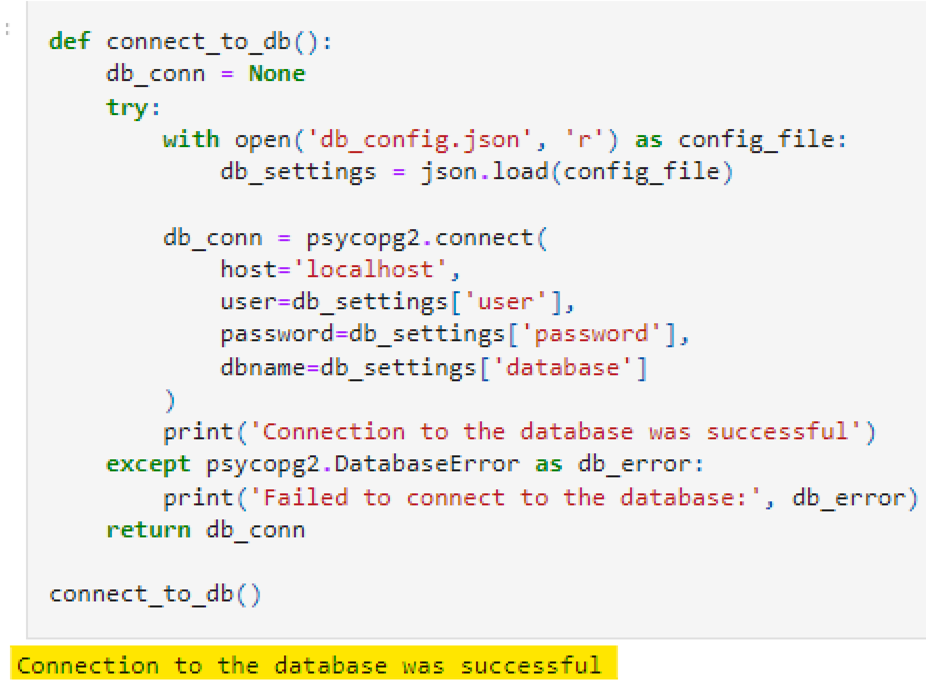
Loading the data to the database

After making sure we've executed the jupyter notebook `loading_data.ipynb` as shown in the next screenshots:



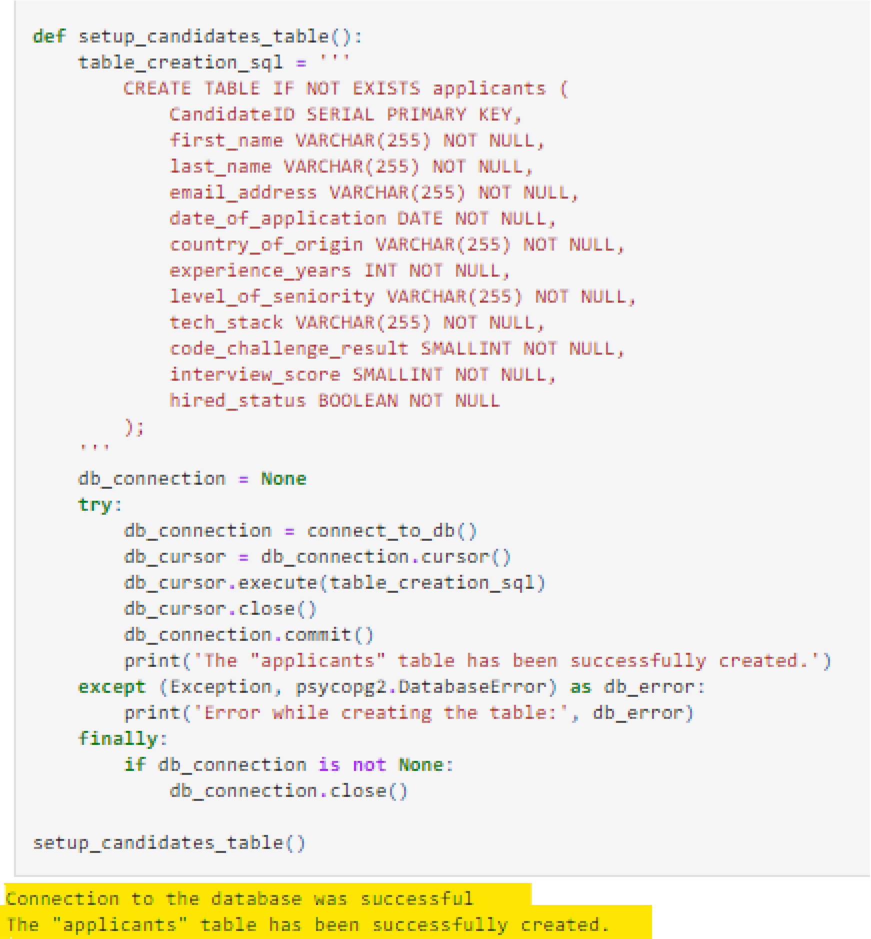
```python
#Creating the database
def create_database(db_name):
    with open('db_config.json', 'r') as config_file:
        config = json.load(config_file)

    conn = psycopg2.connect(
        host=config['host'],
        user=config['user'],
        password=config['password']
    )
    conn.autocommit = True

    cursor = conn.cursor()

    cursor.execute(f"DROP DATABASE IF EXISTS {db_name};")
    cursor.execute(f"CREATE DATABASE {db_name};")

    cursor.close()
    conn.close()
    print(f"the database '{db_name}' has been sucessfully created.")

create_database("workshop1")
```

```
the database 'workshop1' has been sucessfully created.
```



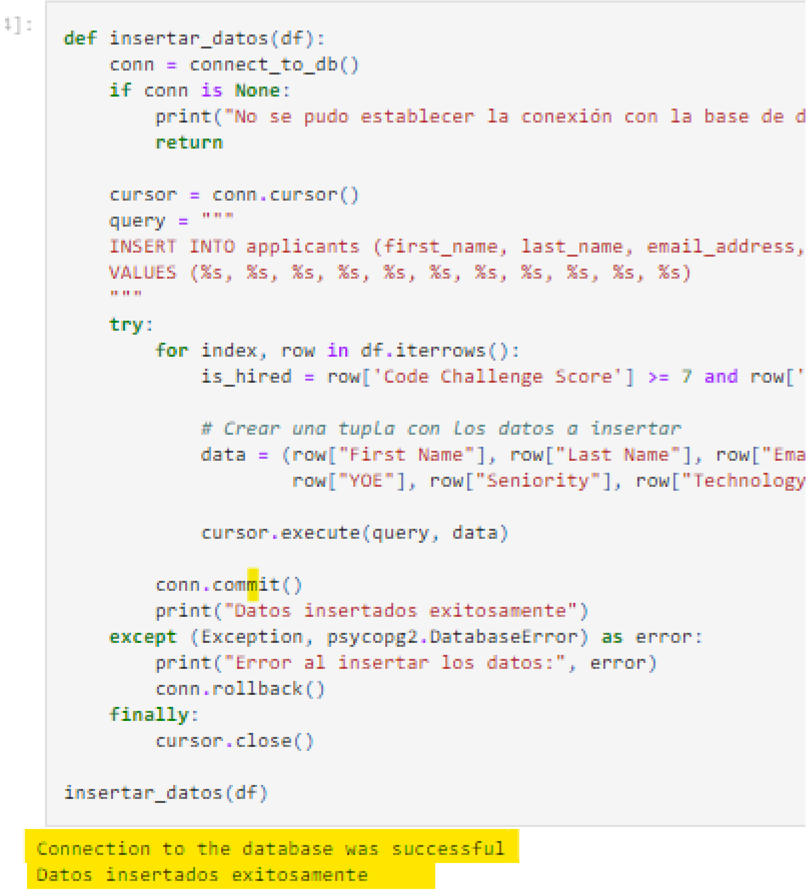
```python
def connect_to_db():
    db_conn = None
    try:
        with open('db_config.json', 'r') as config_file:
            db_settings = json.load(config_file)

        db_conn = psycopg2.connect(
            host='localhost',
            user=db_settings['user'],
            password=db_settings['password'],
            dbname=db_settings['database']
        )
        print('Connection to the database was successful')
    except psycopg2.DatabaseError as db_error:
        print('Failed to connect to the database:', db_error)
    return db_conn

connect_to_db()
```

```
Connection to the database was successful
```



```python
def setup_candidates_table():
    table_creation_sql = '''
        CREATE TABLE IF NOT EXISTS applicants (
            CandidateID SERIAL PRIMARY KEY,
            first_name VARCHAR(255) NOT NULL,
            last_name VARCHAR(255) NOT NULL,
            email_address VARCHAR(255) NOT NULL,
            date_of_application DATE NOT NULL,
            country_of_origin VARCHAR(255) NOT NULL,
            experience_years INT NOT NULL,
            level_of_seniority VARCHAR(255) NOT NULL,
            tech_stack VARCHAR(255) NOT NULL,
            code_challenge_result SMALLINT NOT NULL,
            interview_score SMALLINT NOT NULL,
            hired_status BOOLEAN NOT NULL
        );
    '''
    db_connection = None
    try:
        db_connection = connect_to_db()
        db_cursor = db_connection.cursor()
        db_cursor.execute(table_creation_sql)
        db_cursor.close()
        db_connection.commit()
        print('The "applicants" table has been successfully created.')
    except (Exception, psycopg2.DatabaseError) as db_error:
        print('Error while creating the table:', db_error)
    finally:
        if db_connection is not None:
            db_connection.close()

setup_candidates_table()
```

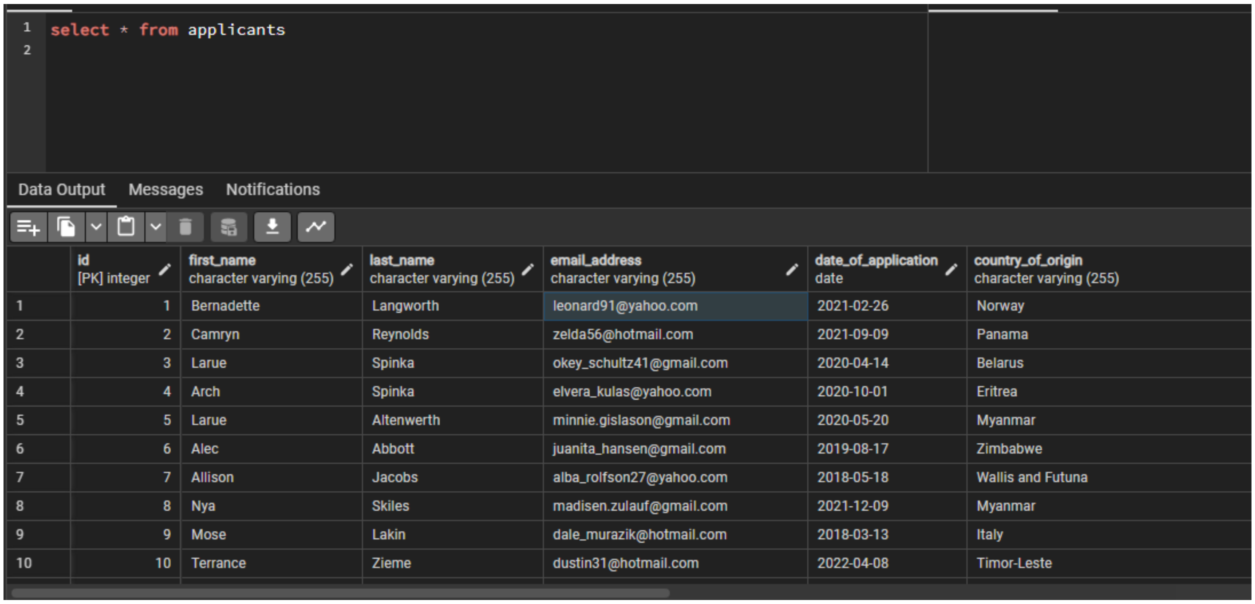
```
Connection to the database was successful
The "applicants" table has been successfully created.
```



```
4]: def insertar_datos(df):
        conn = connect_to_db()
        if conn is None:
            print("No se pudo establecer la conexión con la base de d
            return

        cursor = conn.cursor()
        query = """
        INSERT INTO applicants (first_name, last_name, email_address,
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        """
        try:
            for index, row in df.iterrows():
                is_hired = row['Code Challenge Score'] >= 7 and row['

                # Crear una tupla con los datos a insertar
                data = (row["First Name"], row["Last Name"], row["Ema
                        row["YOE"], row["Seniority"], row["Technology

                cursor.execute(query, data)

            conn.commit()
            print("Datos insertados exitosamente")
        except (Exception, psycopg2.DatabaseError) as error:
            print("Error al insertar los datos:", error)
            conn.rollback()
        finally:
            cursor.close()

    insertar_datos(df)
```

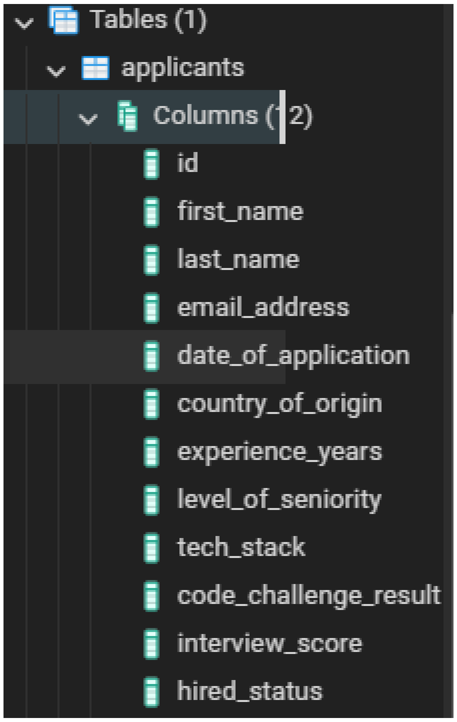
```
Connection to the database was successful
Datos insertados exitosamente
```

Let’s verify if the database has been sucessfully created and the data has been sucessfully uploaded:



```
select * from applicants
```

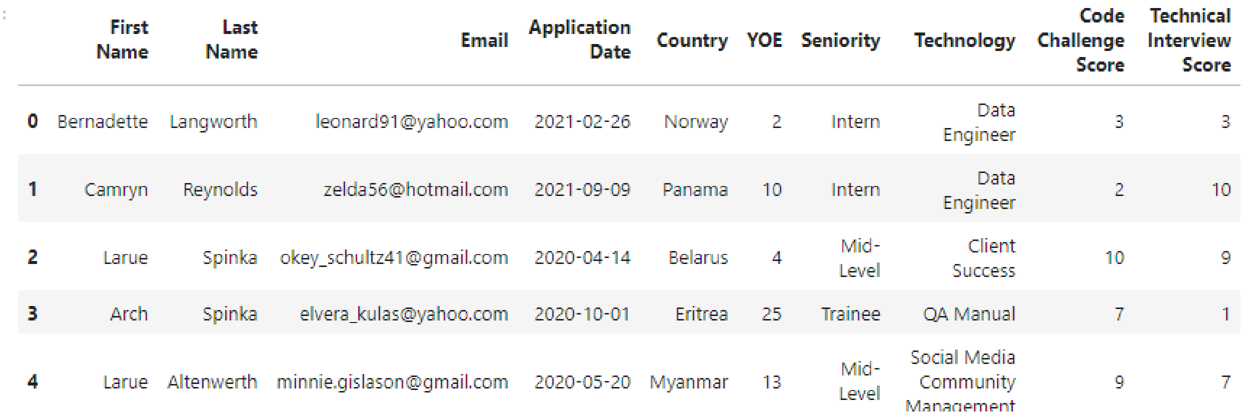
	id [PK] integer	first_name character varying (255)	last_name character varying (255)	email_address character varying (255)	date_of_application date	country_of_origin character varying (255)
1	1	Bernadette	Langworth	leonard91@yahoo.com	2021-02-26	Norway
2	2	Camryn	Reynolds	zelda56@hotmail.com	2021-09-09	Panama
3	3	Larue	Spinka	okey_schultz41@gmail.com	2020-04-14	Belarus
4	4	Arch	Spinka	elvera_kulas@yahoo.com	2020-10-01	Eritrea
5	5	Larue	Altenwerth	minnie.gislason@gmail.com	2020-05-20	Myanmar
6	6	Alec	Abbott	juanita_hansen@gmail.com	2019-08-17	Zimbabwe
7	7	Allison	Jacobs	alba_rolfson27@yahoo.com	2018-05-18	Wallis and Futuna
8	8	Nya	Skiles	madisen.zulauf@gmail.com	2021-12-09	Myanmar
9	9	Mose	Lakin	dale_murazik@hotmail.com	2018-03-13	Italy
10	10	Terrance	Zieme	dustin31@hotmail.com	2022-04-08	Timor-Leste



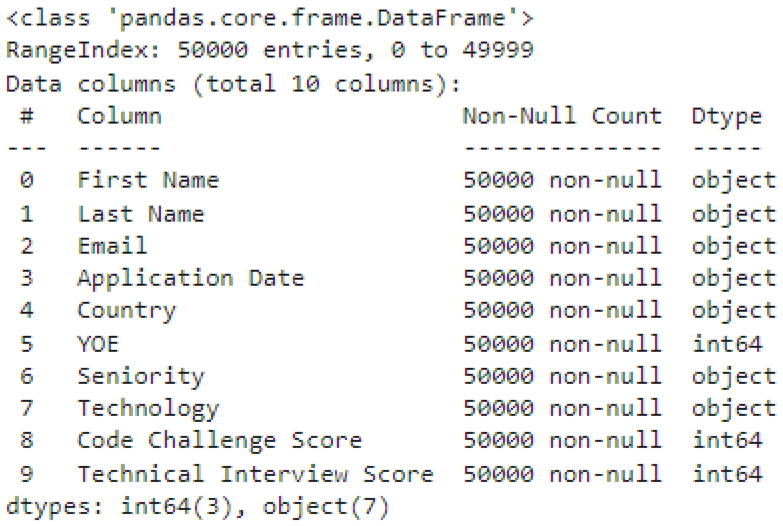
Tables (1)
applicants
Columns (12)
id
first_name
last_name
email_address
date_of_application
country_of_origin
experience_years
level_of_seniority
tech_stack
code_challenge_result
interview_score
hired_status

Exploratory Data Analysis (EDA)

In this section we'll have to execute the jupyter notebook `EDA.ipynb` , these are the results:



	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

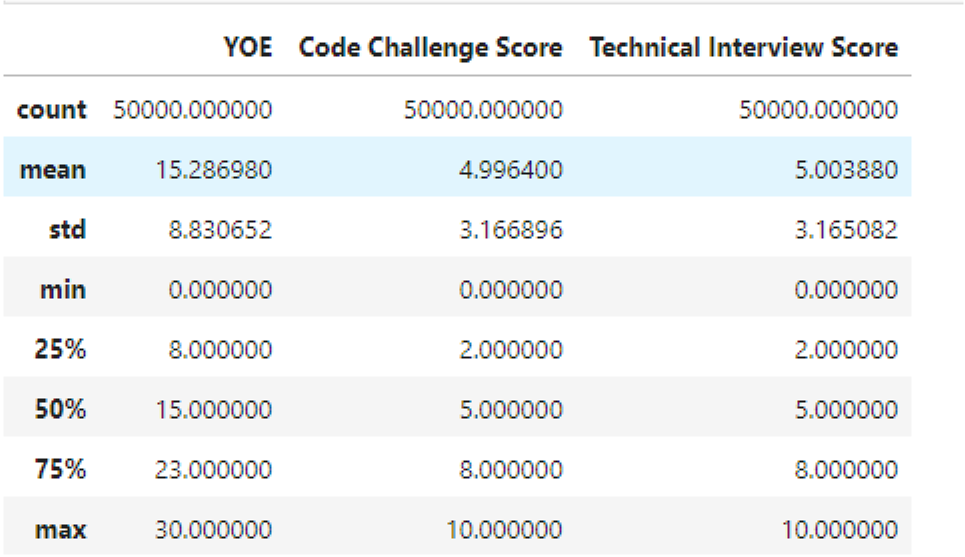


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   First Name                 50000 non-null  object
 1   Last Name                  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)
```

The csv has 10 columns with different data types. Most are text or varchar, except YOE, Code Challenge Score and Technical Interview Score which are integers.

Also, after reviewing the dataset, we can say there are not duplicated records on it.

Let's give it an statistical overview:



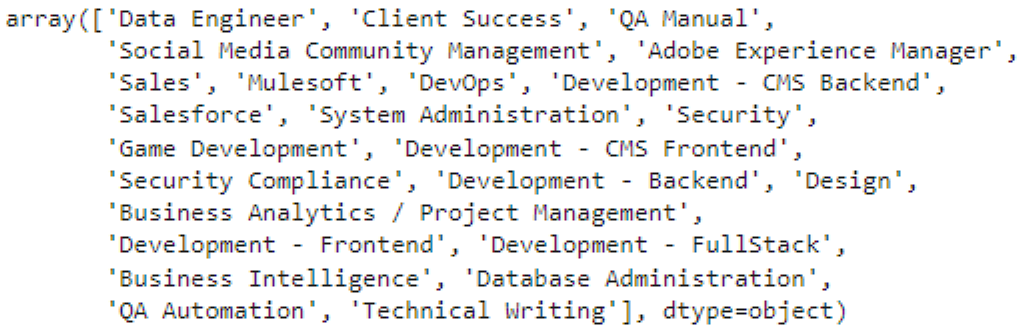
	YOE	Code Challenge Score	Technical Interview Score
count	50000.000000	50000.000000	50000.000000
mean	15.286980	4.996400	5.003880
std	8.830652	3.166896	3.165082
min	0.000000	0.000000	0.000000
25%	8.000000	2.000000	2.000000
50%	15.000000	5.000000	5.000000
75%	23.000000	8.000000	8.000000
max	30.000000	10.000000	10.000000

This table shows a statistical summary of a DataFrame for three columns: 'Years of Experience' (YOE), 'Code Challenge Score', and 'Technical Interview Score'. Here are some insights:

- Count: Each column contains data for 50,000 entries, suggesting no missing values in these fields.
- Mean: On average, candidates have around 15.29 years of experience. The average scores for both the Code Challenge and Technical Interview are close to 5.
- Std (Standard Deviation): The years of experience have a standard deviation of approximately 8.83, indicating a wide range of experience levels among candidates. The Code Challenge and Technical Interview scores have a standard deviation of around 3.17, showing moderate variability in scores.
- Min: The minimum value for all three columns is 0, indicating that there are entries with no experience or no score given for some candidates.
- 25% (1st Quartile): 25% of candidates have 8 or fewer years of experience and scored 2 or lower in both the Code Challenge and Technical Interview.
- 50% (Median): The median years of experience is 15, and the median score for both the Code Challenge and Technical Interview is 5, which is also the maximum possible score.

- 75% (3rd Quartile): 75% of candidates have up to 23 years of experience and scored 8 or lower in the assessments.
- Max: The maximum years of experience recorded is 30, and the highest score for both the Code Challenge and Technical Interview is 10.

Let’s give it a check to the tech categories:

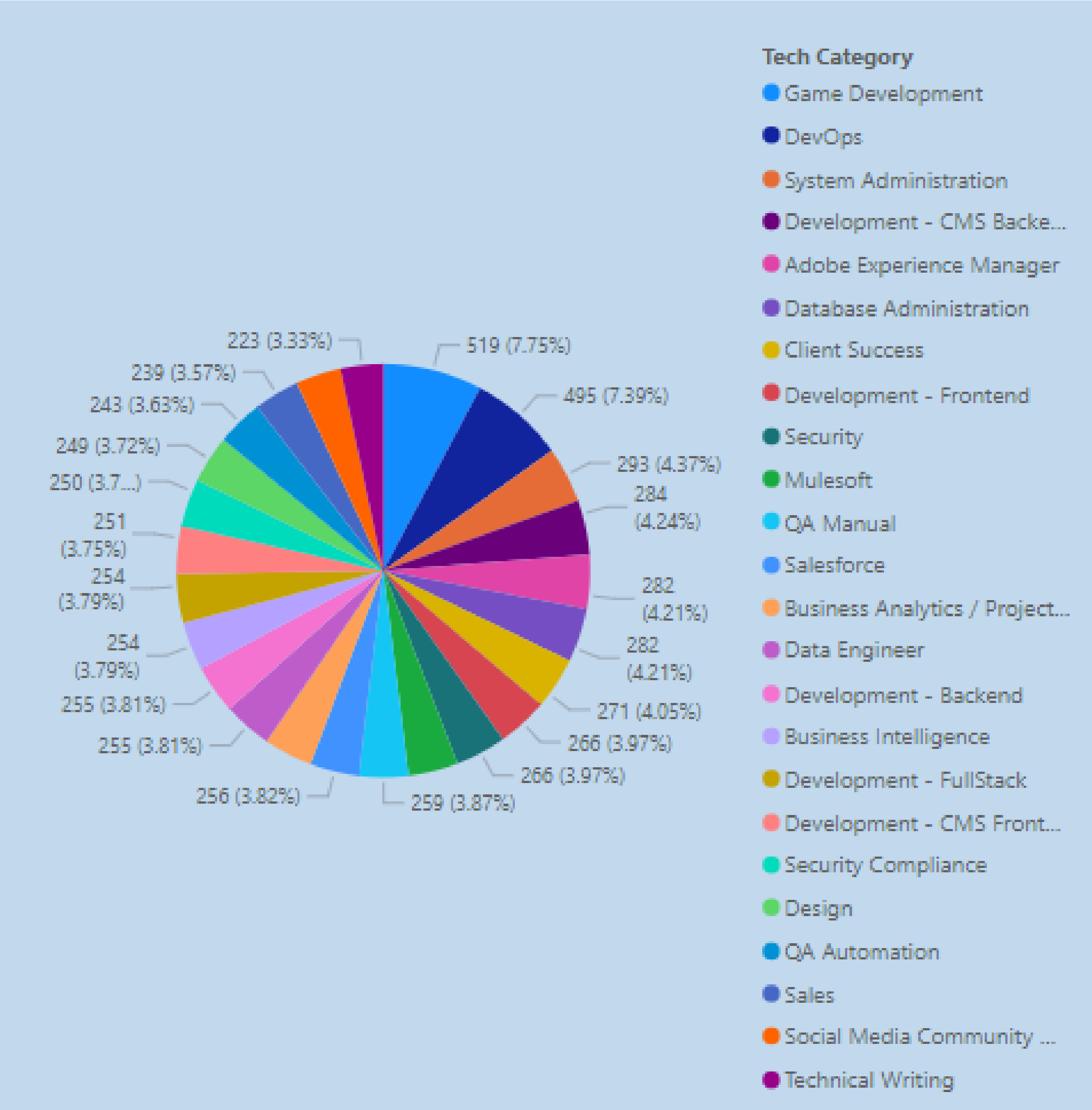


```
array(['Data Engineer', 'Client Success', 'QA Manual',
       'Social Media Community Management', 'Adobe Experience Manager',
       'Sales', 'Mulesoft', 'DevOps', 'Development - CMS Backend',
       'Salesforce', 'System Administration', 'Security',
       'Game Development', 'Development - CMS Frontend',
       'Security Compliance', 'Development - Backend', 'Design',
       'Business Analytics / Project Management',
       'Development - Frontend', 'Development - FullStack',
       'Business Intelligence', 'Database Administration',
       'QA Automation', 'Technical Writing'], dtype=object)
```

Based on the output, it seems there are 24 unique technology specializations or roles, including 'Data Engineer', 'Client Success', 'QA Manual', and others, suggesting a diverse set of technology-related fields among the candidates in the dataset

Dashboard and analysis.

Hires by Tech Category:



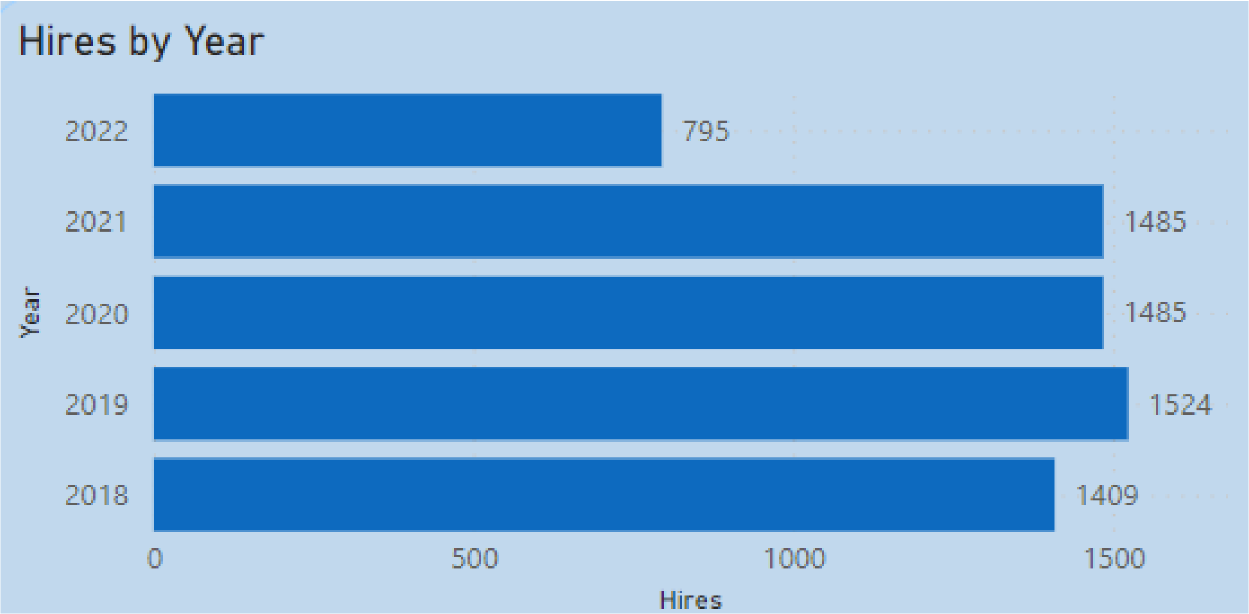
The pie chart presents a distribution of various technology categories within a certain dataset or environment. The most significant category is 'Game Development' with 519 individuals, accounting for 7.75% of the total. 'DevOps' and 'Client Success' are also notable categories, each representing 7.39% with 495 and 495 individuals, respectively.

'Development - Frontend', 'Mulesoft', and 'Business Analytics / Project...' have a comparatively smaller share but still significant, ranging from 4.21% to 4.37%. Other categories like 'Development - Backend', 'Data Engineer', and 'QA Manual' hold around 3.7% to 3.8% each, which indicates a moderate representation in the dataset.

Smaller categories like 'Adobe Experience Manager (AEM) CMS Frontend Development', 'Security Compliance', 'QA Automation', and 'Sales' have the least representation, with each comprising about 3.37% to 3.57%.

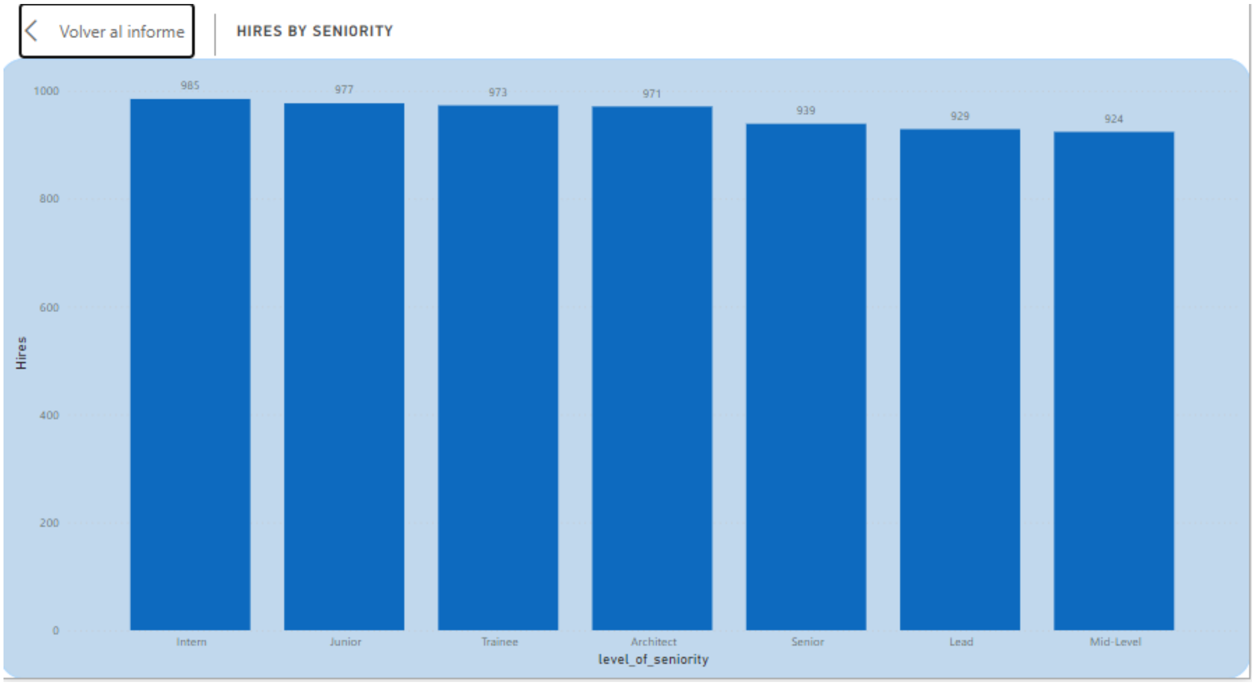
It is evident from the chart that there is a diverse range of technology categories represented, with 'Game Development' being the most prominent, while other roles like 'Technical Writing' and 'Social Media Community ...' are the least represented in this particular dataset. The distribution suggests a varied tech environment with a wide range of specializations.

Hires by year:



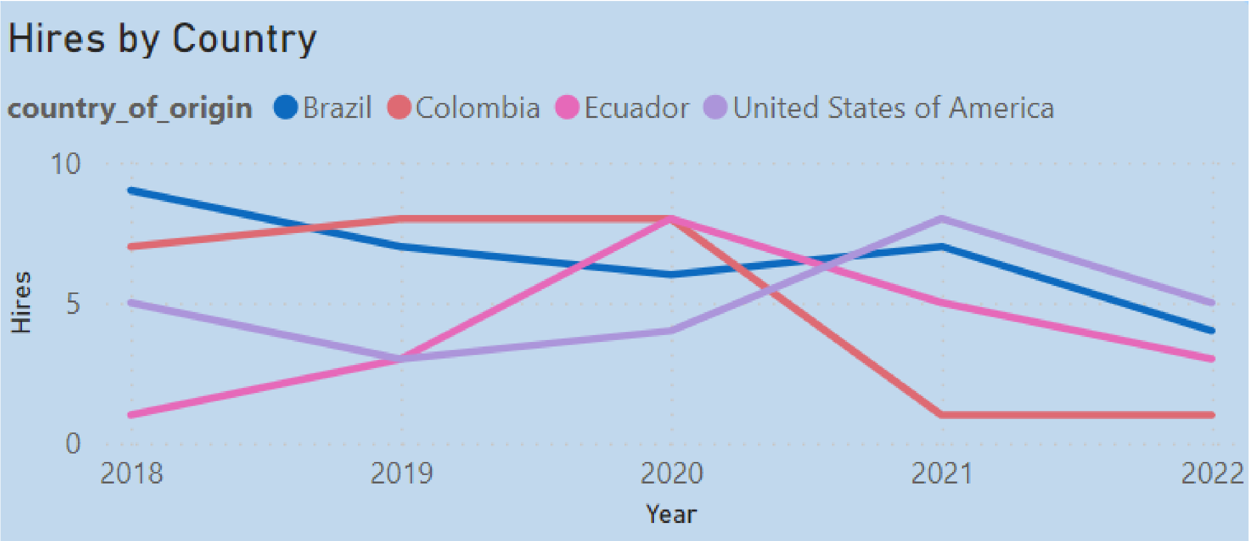
The bar chart, "Hires by Year," displays the number of hires from 2018 through July 2022. The data suggests a stable hiring rate for 2019 through 2021, with the peak at 1,524 hires in 2019 and both 2020 and 2021 showing an identical figure of 1,485 hires. The year 2018 is observed to have the lowest among these, with 1,409 hires. It's important to note that the data for 2022, which shows 795 hires, only accounts for the period up to July. Therefore, it cannot be directly compared to the full-year data of previous years. Without the data for the remaining months, any conclusions about a decrease or trend in 2022 would be speculative. The chart provides a snapshot of hiring patterns over the years but necessitates the complete 2022 data for a comprehensive annual comparison.

Hires by Seniority Level:



This shows how many hires were made at different experience levels within an organization. The distribution is fairly uniform, implying that the hiring covered various levels of expertise. The ‘Intern’ position had the most hires at 985, which could indicate a strong emphasis on nurturing new talent. The ‘Junior’ level was close behind with 977 hires, and the ‘Trainee’ and ‘Architect’ positions had almost the same numbers at 973 and 971 hires, respectively, implying a balanced hiring strategy for these roles. Senior positions were also well-filled with 939 hires, while the ‘Lead’ and ‘Mid-Level’ categories had a slightly smaller number, at 929 and 924 hires respectively. The chart implies a solid hiring plan that did not favor any specific level of seniority, but rather showed a varied approach to talent recruitment across the range.

Hires by Seniority



The multiline graph compares the number of hires from four countries—Brazil, Colombia, Ecuador, and the United States of America—over a five-year period from 2018 to 2022. Here's an interpretation of the trends shown:

- Hiring trends fluctuate over the years for each country, indicating variability in employment activities.
- In 2018, Colombia started as the leading country in hires, but saw a decline by 2019, followed by a sharp increase in 2020, and then a decline once more in 2021.
- Brazil shows an overall increasing trend in hires, starting from lower than Colombia in 2018 and surpassing it in 2021.
- Ecuador has the most stable line, indicating consistency in the number of hires over the years, with a slight uptick in 2020.
- The United States shows a peak in hires in 2019, surpassing all other countries, but there’s a notable drop in the following years, with hires declining to the lowest point of all four countries by 2022.
- For 2022, it appears that hiring numbers are generally lower across all countries, which could suggest a common downward trend, although, similar to the previous dataset, if the data for 2022 does not cover the entire year, it would be premature to conclude this as a definite trend.

This graph could serve as a tool to analyze the hiring practices of an organization across different geographical locations and could be indicative of broader economic or organizational shifts affecting employment.

Conclusions

The hiring data from the last five years reveals an interesting landscape with varying trends across time, seniority levels, and countries. A initial spike in hires reached its peak in 2019, followed by a stable trend until 2021, indicating a phase of stabilization in the organization's growth. The distribution among seniority levels has been remarkably fair, with a slight emphasis on junior talent, suggesting a balanced approach to building a strong workforce.

Geographical analysis shows different regional trends. Colombia and Brazil experienced significant growth, with Brazil showing a steady increase in hires. The United States and Ecuador showed more volatility, the former with a noticeable drop after 2019. However, the 2022 data, incomplete as of July, cautions against drawing firm conclusions for the current year, as it shows a general downtick in hiring numbers.

In summary, the organization has shown adaptability across diverse markets and a consistent investment in talent across all seniority levels. The comprehensive data from the complete 2022 hiring cycle will be crucial for understanding the full scope of current hiring dynamics.

Link to this workshop's repository: https://github.com/s4ntiagor/Workshop_1.git