



PROYECTO 1

MARIA FERNANDA TELLO VERGARA
ANA CRISTINA QUINTERO CARPINTERO
JOHAN EDELBERTO HURTADO ENRIQUEZ

DOCENTE: JAVIER ALEJANDRO VERGARA

ELT

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
SANTIAGO DE CALI
28 AGOSTO 2024



CONTEXT

This project focuses on the analysis of a health dataset that includes detailed information on 1,879 patients. Each patient is uniquely identified with an ID between 6000 and 7878. The dataset covers demographic details, lifestyle factors, medical history, clinical measurements, medication use, symptoms, quality of life scores, environmental exposures, and health behaviors.

The analysis is performed with the aim of extracting valuable information about health patterns, effectiveness of treatments, and factors affecting patients' quality of life. Confidentiality and privacy of information is ensured by associating each patient with a physician in charge, whose name is kept confidential.

PURPOSE

The purpose of this project is to apply advanced data analysis techniques to improve understanding of factors influencing patients' health, optimize medical treatments, and promote better quality of life. In addition, the project seeks to ensure proper and secure handling of sensitive patient information by using appropriate data encryption and storage technologies.

DESCRIPTION

This project involves the comprehensive analysis of a health dataset comprising detailed information on 1,879 patients, each uniquely identified with an ID in the range of 6000 to 7878. The dataset includes a wide variety of information, such as demographic details (age, sex, ethnicity), lifestyle factors (eating habits, physical activity), medical history (previous illnesses, family history), clinical measurements (blood pressure, glucose levels), medication use (current and past prescriptions), reported symptoms, quality of life scores (measures of physical and mental well-being), environmental exposures (pollution levels, exposure to toxicants), and health behaviors (smoking, alcohol consumption).

TOOLS

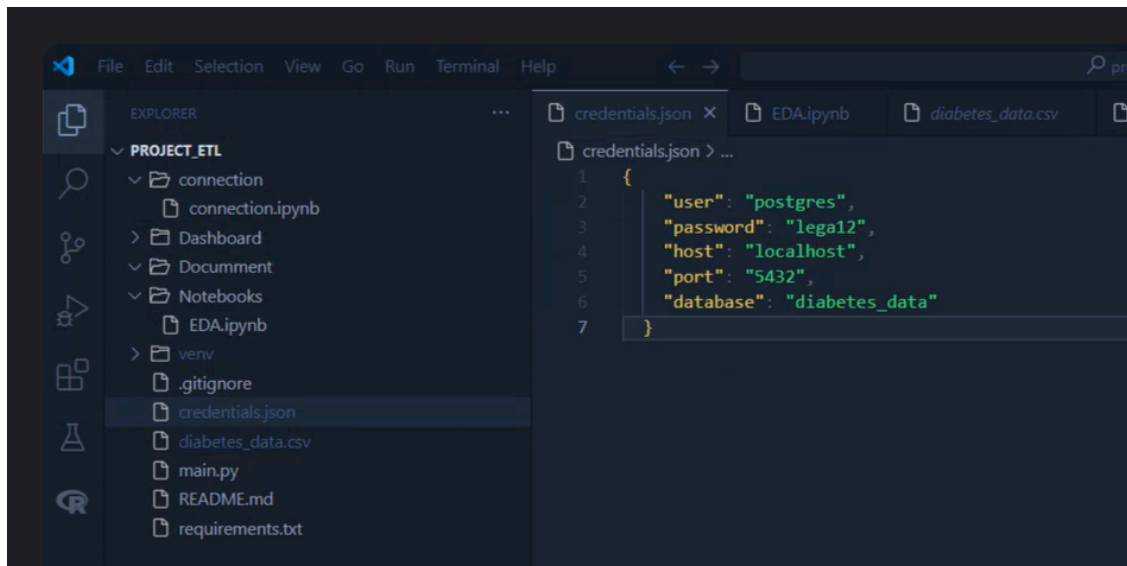
- Python (Pandas, Matplotlib, SQLAlchemy, dotenv).
- PostgreSQL.
- Jupyter Notebook.
- Dataset (Candidates).
- Encryption of credentials using a .env file (Environment variables).



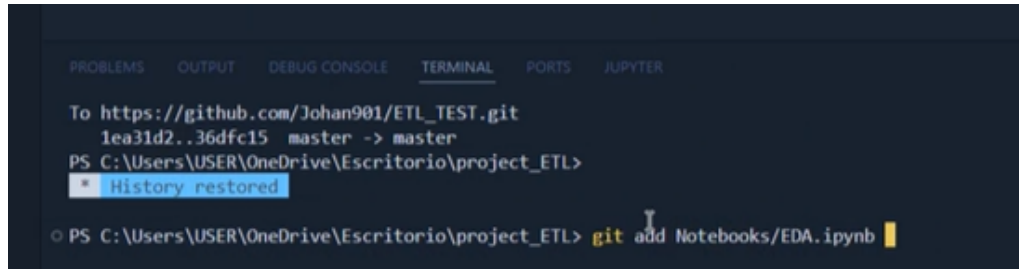
STEP BY STEP

The connection is established with the credentials in the database.

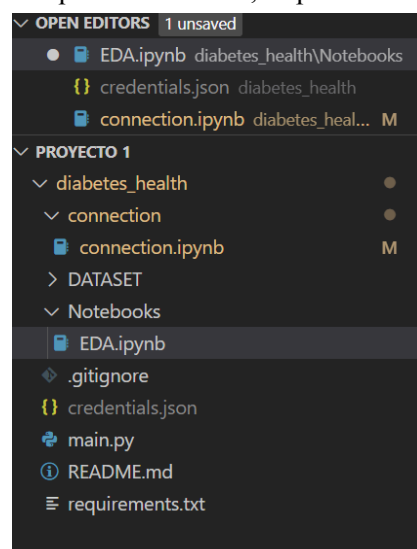
The credentials.json file contains the credentials for connecting to the PostgreSQL database. We use it to store the credentials in an organized way.



Here we can see how we upload the updated changes in Visual Studio Code.



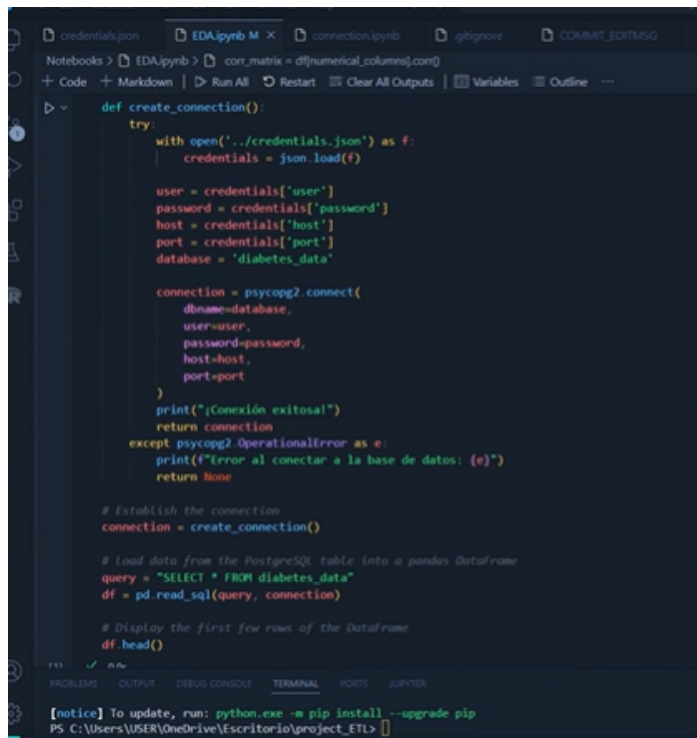
The folders are organized with the requested structure, as presented below:



The database is created, it is important to clarify that we are using postgres.

```
postgres=# CREATE DATABASE diabetes_data;  
CREATE DATABASE
```

At this point we are going to clean the data.



```
def create_connection():  
    try:  
        with open('..credentials.json') as f:  
            credentials = json.load(f)  
  
        user = credentials['user']  
        password = credentials['password']  
        host = credentials['host']  
        port = credentials['port']  
        database = 'diabetes_data'  
  
        connection = psycopg2.connect(  
            dbname=database,  
            user=user,  
            password=password,  
            host=host,  
            port=port  
        )  
        print("¡Conexión exitosa!")  
        return connection  
    except psycopg2.OperationalError as e:  
        print(f"Error al conectar a la base de datos: {e}")  
        return None  
  
# Establish the connection  
connection = create_connection()  
  
# Load data from the PostgreSQL table into a pandas DataFrame  
query = "SELECT * FROM diabetes_data"  
df = pd.read_sql(query, connection)  
  
# Display the first few rows of the DataFrame  
df.head()
```



```
(Conexión exitosa!  
C:\Users\USER\AppData\Local\Temp\inkernel_42228\3734632416.py:30: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connecti  
df = pd.read_sql(query, connection)
```

	patientid	age	gender	ethnicity	socioeconomicstatus	educationlevel	bmi	smoking	alcoholconsumption	physicalactivity	...	tinglinghandsfeet	qualityoflifescore	heavymetalsexposure	occups
0	6000	44	0	1		2	1 32.985284	1	4.499364662559289	2.443385277880059	...	1	73.765109		0
1	6001	51	1	0		1	2 39.916764	0	1.578919022031171	8.301264419669659	...	0	91.445753		0
2	6002	89	1	0		1	3 19.782251	0	1.1773011585548998	6.103395048386896	...	0	54.485744		0
3	6003	21	1	1		1	2 32.376881	1	1.714621007745527	8.64546518551969	...	0	77.866758		0
4	6004	27	1	0		1	3 16.808600	0	15.4625488312587	4.62938308903732	...	0	37.731808		0

In this cleaning we identify columns with missing values, categorical variables such as Gender, Ethnicity, Smoking, and Diagnosis are converted to categorical type (or int if they are kept in numeric format). We check that the values of the continuous variables are within the expected ranges (for example, BMI between 15 and 40, BloodGlucose between 70 and 200 mg/dL).

```

.. Valores faltantes por columna:
   patientid      0
   age            0
   gender         0
   ethnicity      0
   socioeconomicstatus 0
   educationlevel 0
   bmi           0
   smoking       0
   alcoholconsumption 0
   physicalactivity 0
   dietquality   0
   sleepquality  0
   familyhistorydiabetes 0
   gestationaldiabetes 0
   polycysticovarysyndrome 0
   previousprediabetes 0
   hypertension  0
   systolicbp    0
   diastolicbp   0
   fastingbloodsugar 0
   hba1c         0
   serumcreatinine I 0
   bunlevels     0
   cholesteroltotal 0
   ...
   healthliteracy      float64
   diagnosis           category
   doctorincharge      object
   dtype: object

```

In this image we can see that there are no missing values.

Export clean data to a new table called diabetes data clean, and proceed to connect to the database to perform the EDA.

```

# exportamos datos limpios a la base de datos
with open('../credentials.json') as f:
    credentials = json.load(f)

user = credentials['user']
password = credentials['password']
host = credentials['host']
port = credentials['port']
database = 'diabetes_data'

connection_string = f'postgresql+psycopg2://{user}:{password}@{host}:{port}/{database}'
engine = create_engine(connection_string)

df.to_sql('diabetes_data_clean', engine, if_exists='replace', index=False)
print("Datos exportados exitosamente a la tabla 'diabetes_data_clean' en PostgreSQL.")

# comprobamos
print("DataFrame procesado:")
print(df.head())
print(df.dtypes)

```

```

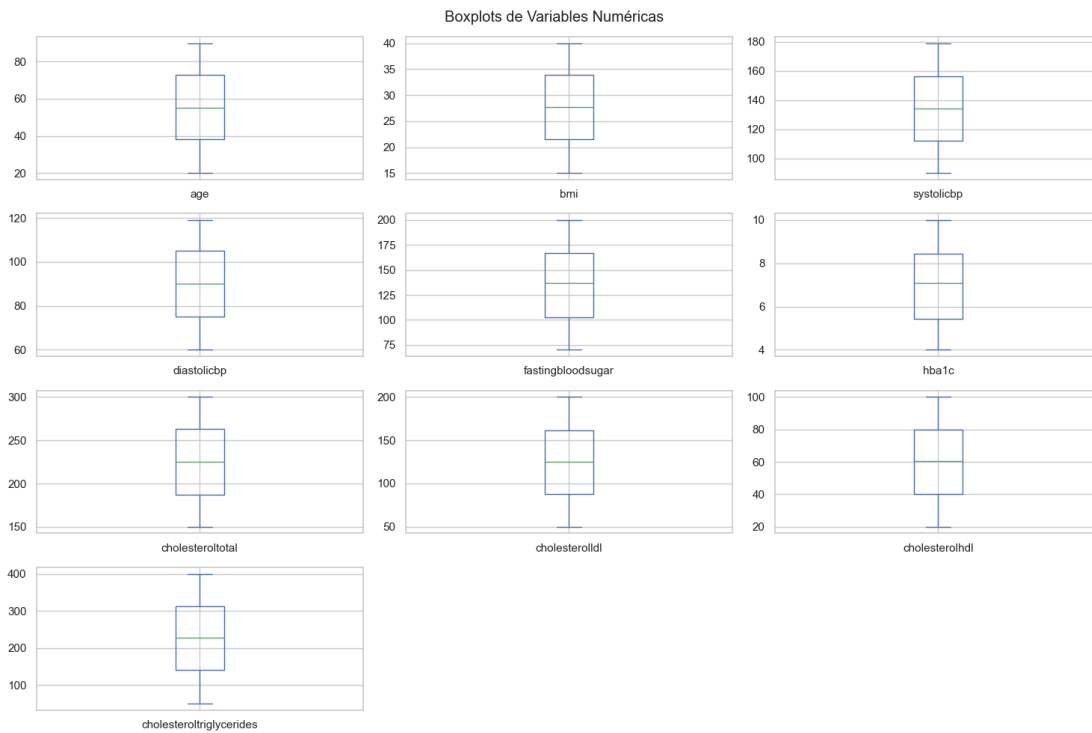
✓ 0.4s
diastolicbp      float64
fastingbloodsugar float64
hba1c            float64
serumcreatinine  float64
bunlevels        float64
cholesteroltotal float64
cholesterolhdl   float64

```

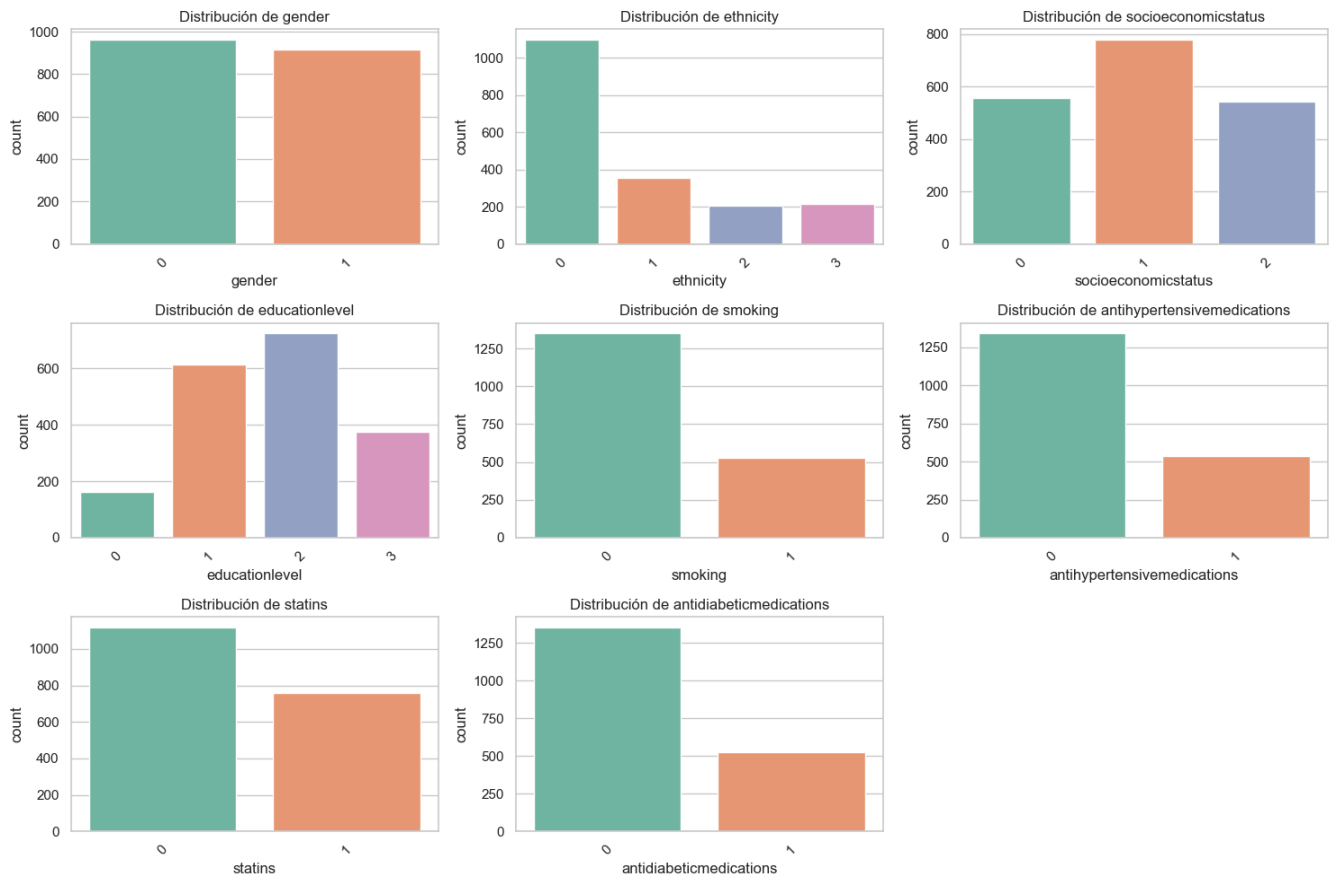
Here we can see some graphs that help us understand the exploratory analysis of the data.



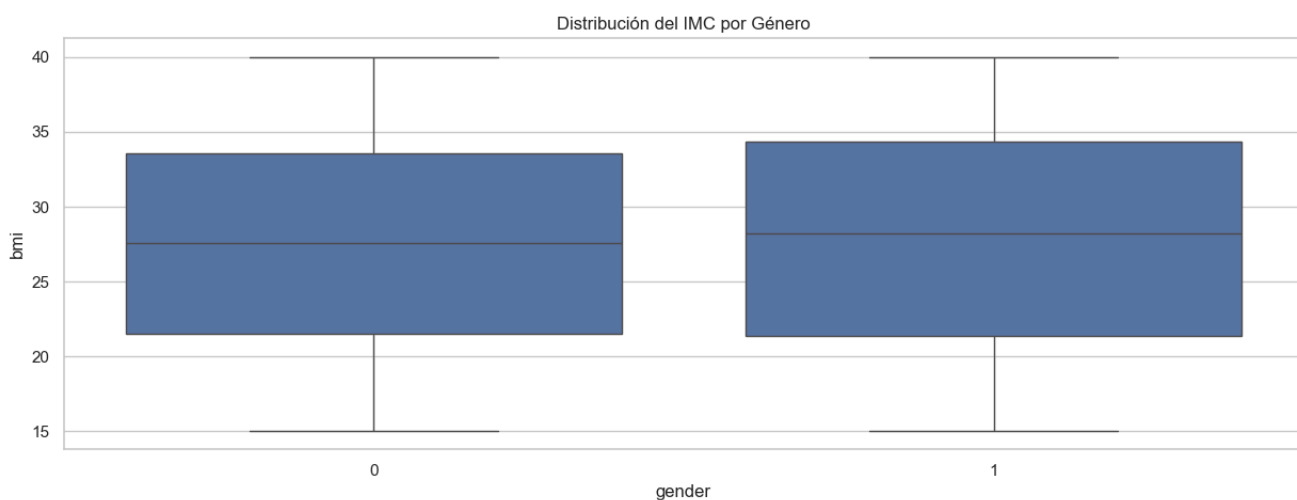
The results of the graph show the distribution of several health-related numerical variables, such as age, blood pressure, blood sugar levels, and different types of cholesterol. Each histogram reveals how the data is distributed for each variable, allowing you to identify patterns, trends, and potential anomalies. For example, we can look at the frequency of different age ranges in the sample, systolic and diastolic blood pressure levels, and HDL, LDL, and total cholesterol levels.



The results of the graph show the distribution of several numerical variables related to health, such as age, body mass index (BMI), systolic and diastolic blood pressure, testosterone levels and different types of cholesterol. Each boxplot reveals the dispersion and central trend of the data, highlighting the median, interquartile range, and possible outliers.

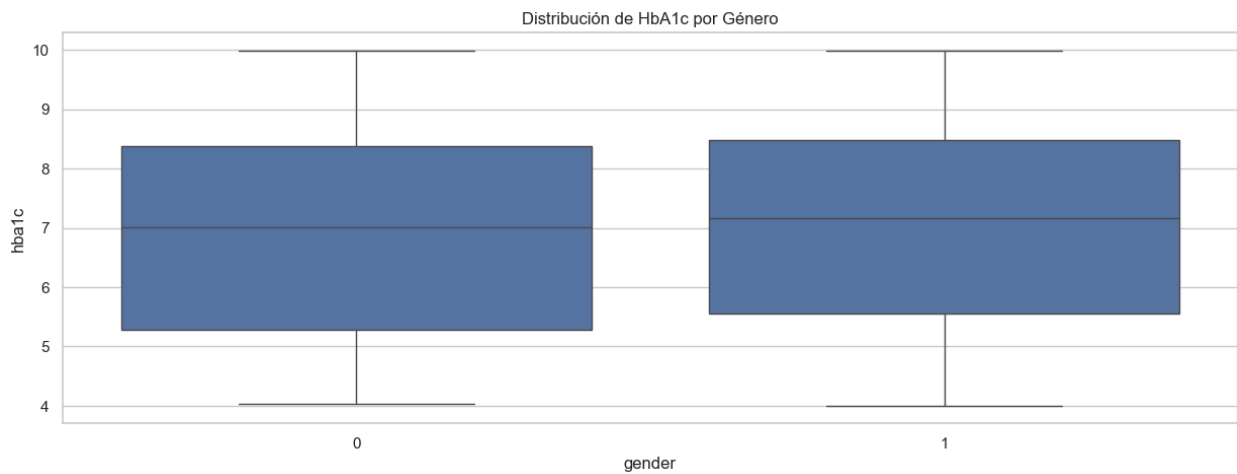


The results in the graph show the distribution of various demographic and health variables. Each bar graph represents a different variable, such as gender, ethnicity, socioeconomic status, education level, smoking status, use of antihypertensive drugs, use of statins, and use of antidiabetic drugs. The graphs allow you to visualize the frequency or count of each category within these variables.

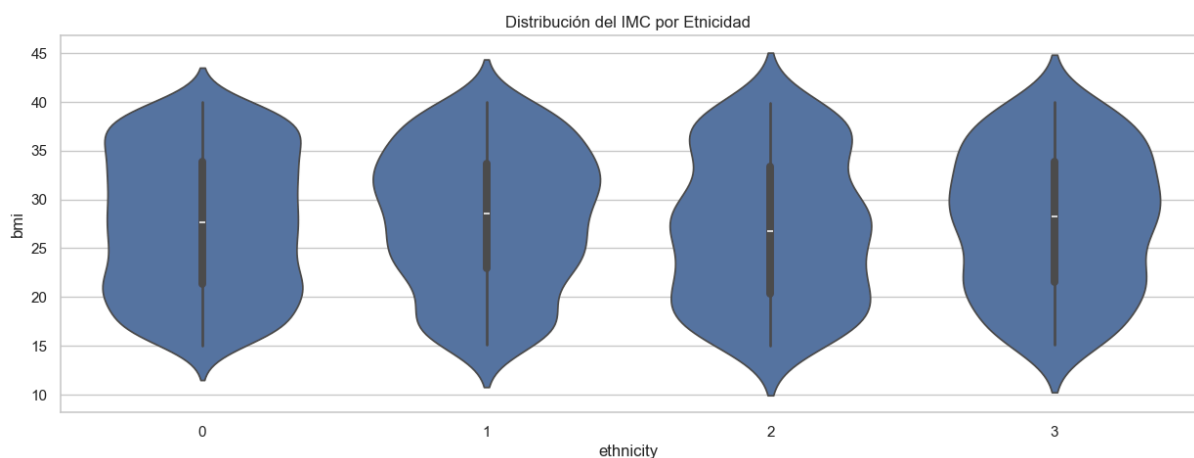


The graph shows the distribution of Body Mass Index (BMI) by gender. There are two boxplots, one for each gender, which represent the dispersion of BMI values. The box in each boxplot indicates the interquartile range (IQR), which contains the middle 50% of the data, while the line inside the box

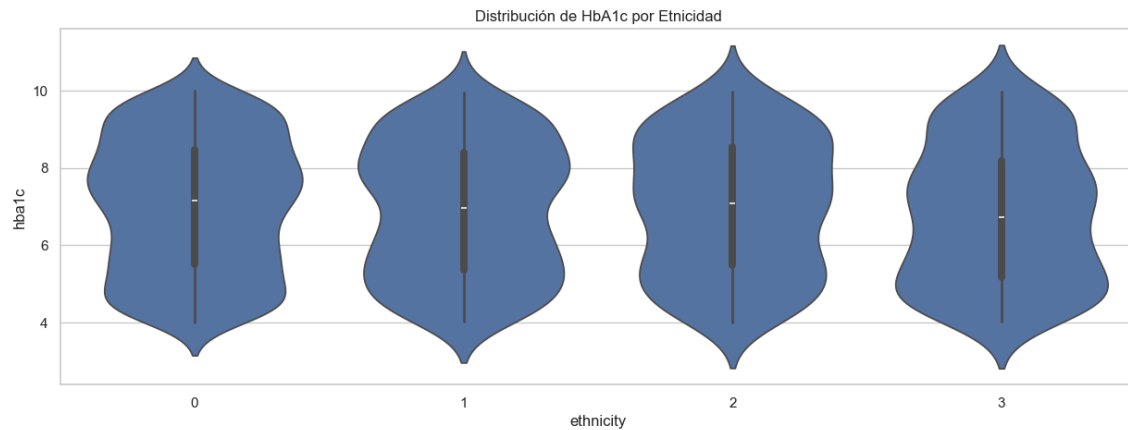
represents the median BMI. The "whiskers" extend to the furthest values that are not considered outliers, and the points outside the whiskers are outliers.



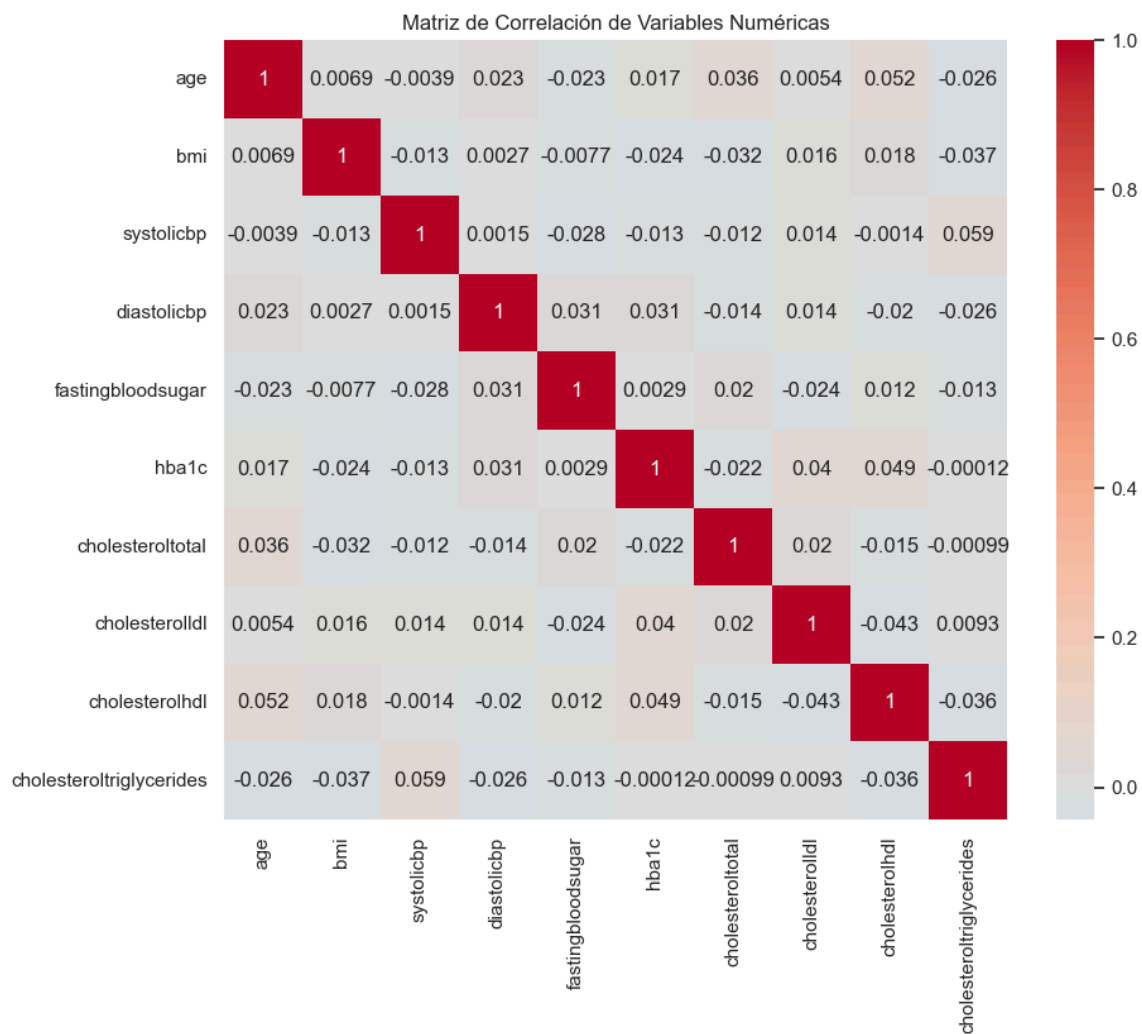
The graph shows the distribution of HbA1c levels by gender. HbA1c is a measure that reflects the average blood glucose levels over the past three months. There are two boxplots, one for each gender, which represent the dispersion of HbA1c values. The box in each boxplot indicates the interquartile range (IQR), which contains the central 50% of the data, while the line inside the box represents the median HbA1c. The "whiskers" extend to the furthest values that are not considered outliers, and the points outside the whiskers are outliers.



The graph shows the distribution of Body Mass Index (BMI) by ethnicity using a violin diagram. Each violin represents a category of ethnicity and shows the density of BMI values within each group. The width of each violin at different BMI levels indicates the frequency of the data at those levels; wider sections mean more data in that range. In addition, the median (white point) and the interquartile range (thick black bar) are marked within each violin.



The graph shows the distribution of HbA1c levels by ethnicity using a violin diagram. Each violin represents a category of ethnicity and shows the density of HbA1c values within each group. The width of each violin at different HbA1c levels indicates the frequency of the data at those levels; wider sections mean more data in that range. In addition, the median (white point) and the interquartile range (thick black bar) are marked within each violin.



Here we can see the correlation matrix, which is a table that shows the evaluation coefficients between different variables. Each cell in the table shows the valuation between two variables. The values range from -1 to 1:

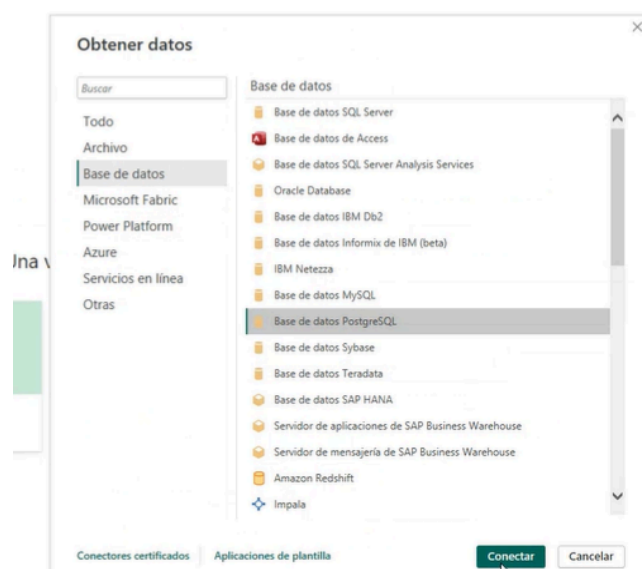
- 1 indicates a perfect positive improvement (when one variable increases, the other does too).
- -1 indicates perfect negative compensation (when one variable increases, the other decreases).
- 0 indicates that there is no offset between the variables.

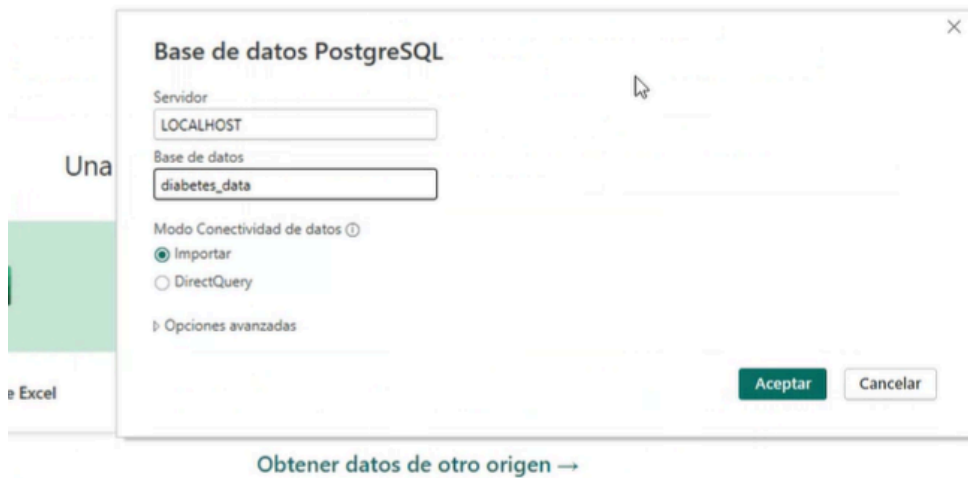
Various health-related metrics are included in this matrix, such as body mass index (BMI), age, systolic blood pressure (systolicbp), diastolic blood pressure (diastolicbp), fasting blood sugar (fasting sugar), hemoglobin A1c levels (hba1c), total cholesterol levels (totalcholesterol) and HDL and LDL cholesterol ratios (hdlcholesterol and ldlcholesterol).

The diagonal from the upper left corner to the lower right corner shows perfect correlations of 1 because each variable is correlated with itself. What's interesting about this image is that it visually represents how different health metrics are related to each other, which can be useful in medical research and diagnoses.

For example, there is a notable positive elevation of 0.031 between age and fasting blood sugar levels, suggesting that as age increases, fasting blood sugar levels tend to increase slightly as well.

Below we can see the connection of the database with Power BI, we create the dashboard.





These are some questions we can answer in the dashboard

- What is the distribution of patients by diabetes diagnosis?
- How are patients distributed by gender?
- What is the relationship between alcohol consumption and medication adherence among patients?
- What is the distribution of patients by socioeconomic status?
- What is the relationship between HbA1c levels and fasting glucose in patients, and how does this relationship vary between patients diagnosed with diabetes and those not?
- How is the diagnosis of diabetes distributed among patients who have been exposed to chemicals?
- What percentage of patients have a history of gestational diabetes?

Finally, we uploaded all the folders to the github repository.