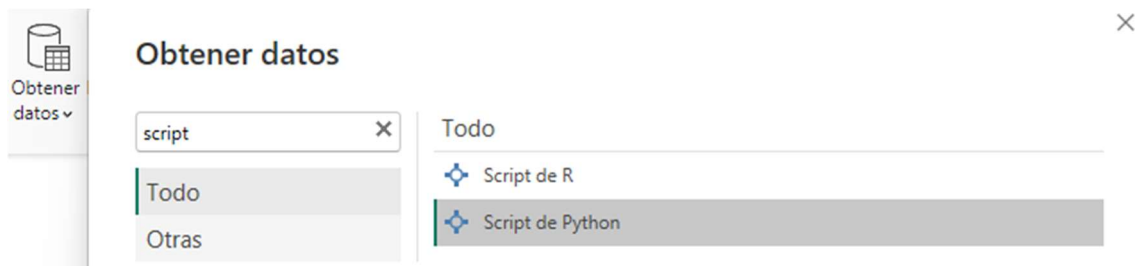


Sprint 08.2

1er Paso : Conectar Python a PowerBi

Primeramente, debemos seleccionar “Obtener Datos” y luego buscar entre las opciones “Script de Python”



Una vez hecho nos saldrá un cuadro de diálogo donde debemos introducir nuestro código que obtiene la conexión a la base de datos además de generar los dataframes correspondientes:

```
import pandas as pd
from sqlalchemy import create_engine

connection = create_engine('mysql+pymysql://root:itjuanr@localhost/sprint0_4')

tablas_db = ['transactions', 'companies', 'productos_transacciones_enlace', 'user', 'products']
df = {}

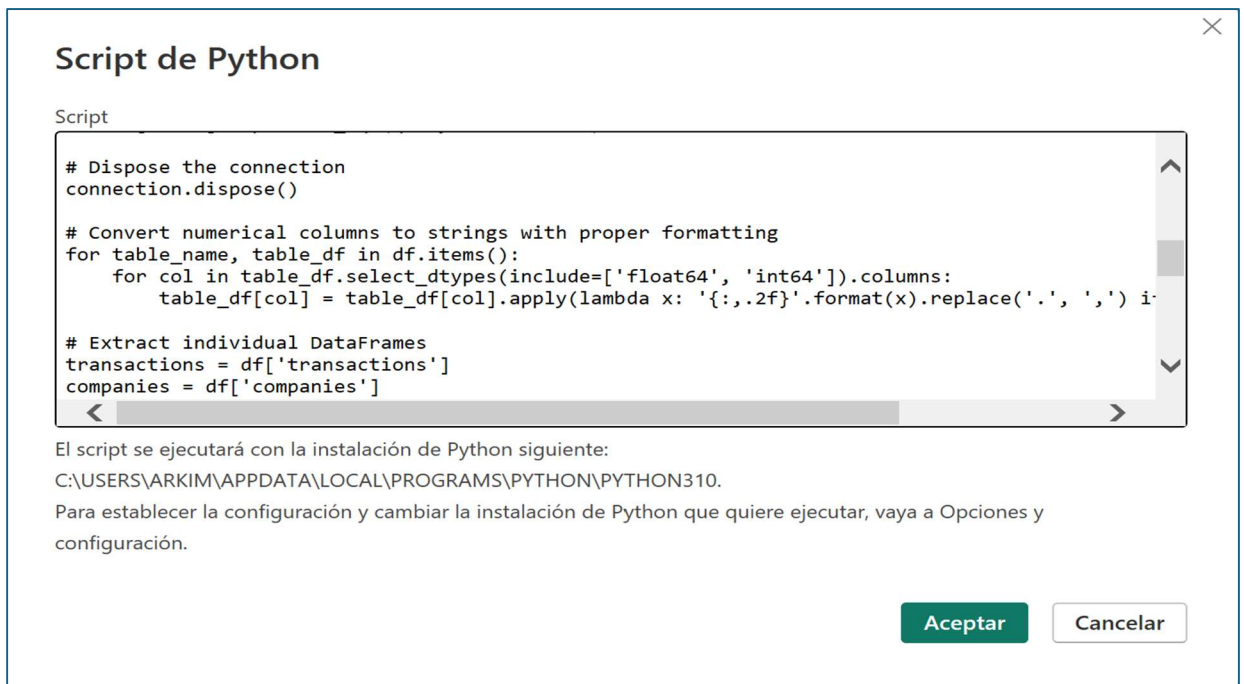
for tabla in tablas_db:
    query = f'SELECT * FROM {tabla}'
    df[tabla] = pd.read_sql(query, connection)

connection.dispose()

for table_name, table_df in df.items():
    for col in table_df.select_dtypes(include=['float64', 'int64']).columns:
        table_df[col] = table_df[col].apply(lambda x: '{:,.2f}'.format(x).replace('.', ',')) if pd.notnull(x) else x

transactions = df['transactions']
companies = df['companies']
productos_transacciones_enlace = df['productos_transacciones_enlace']
user = df['user']
products = df['products']
```

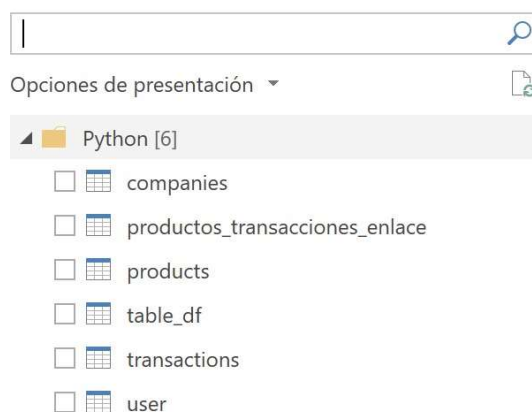
(imagen del código utilizado incluido en jupyter notebook)



Cuadro Dialogo para introducir el script en PowerBi

Ahora como habitualmente seleccionamos nuestras “Tablas” para trabajar.

Navegador



*Remarco que para realizar este ejercicio tuve que cambiar la conexión con mysql.connector a usar mysql alchemy ya que provocaba un error al usar el anterior.

2do Paso: Creación del Gráfico

Como es habitual, en el menú de las visualizaciones, hacemos clic sobre la imagen de Python para iniciar la creación:



Una vez hemos hecho clic, deberemos seleccionar las columnas correspondientes con las que trabajaremos.

Cuando hayamos seleccionado las columnas se nos abrirá un cuadro de texto donde deberemos introducir el código para crear la visualización

```
Editor de scripts de Python

⚠ Las filas duplicadas se quitarán de los datos.

1 # El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta :
2
3 # dataset = pandas.DataFrame(amount)
4 # dataset = dataset.drop_duplicates()
5
6 # Pegue o escriba aquí el código de script:
7 import matplotlib.pyplot as plt
8 import seaborn as sns
9 import numpy as np
10 amount_df = dataset["amount"]
11
12 plt.figure(figsize=(15, 9))
13 sns.histplot(amount_df, kde=True, bins=10)
14 plt.title('Histograma del Amount')
15 plt.xlabel('Valores')
16 plt.xticks(np.arange(min(amount_df), max(amount_df)+1,30))
17 plt.ylabel('Frecuencia')
18 plt.tight_layout()
19 plt.show()
20
```

Una vez hecho esto ejecutamos el código y se nos generará el gráfico.

Este proceso se debe repetir para cada gráfico.

Notas:

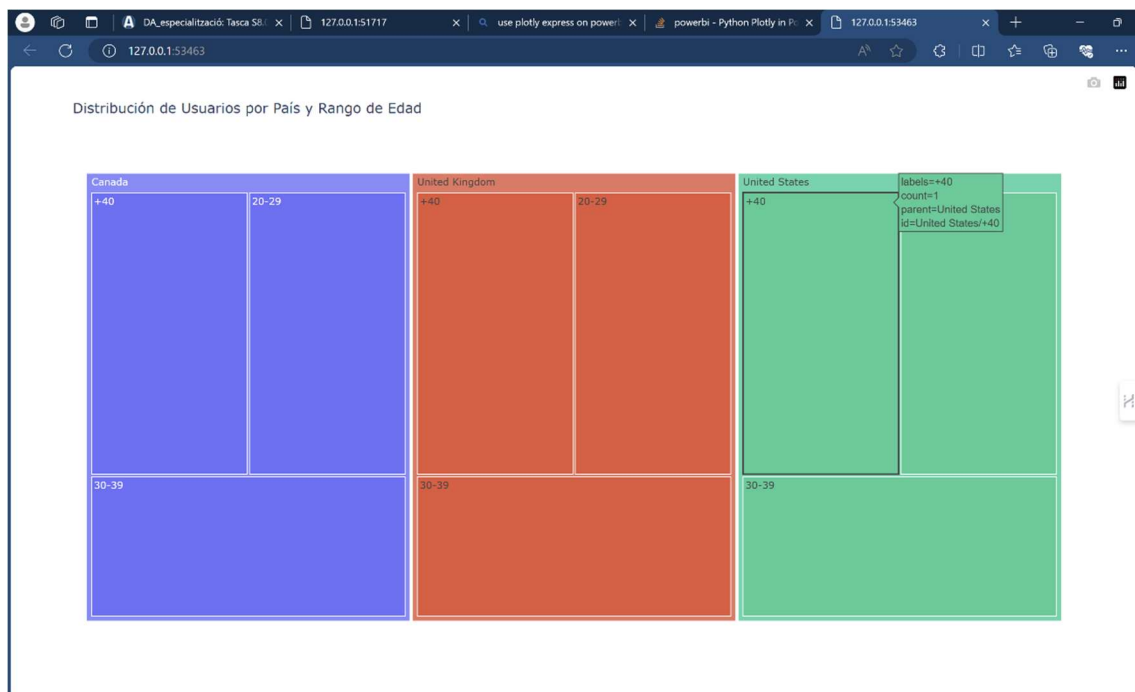
-Nuevo nombre del dataframe

Hay que tener en cuenta que, al empezar la creación de la visualización, al seleccionar las columnas, estas estarán dentro de la variable “dataset”. Por lo que debemos adaptar el código del Sprint anterior para ajustarlo a esta nueva variable.

-Nivel 1 Ejercicio 5

En este ejercicio, en el sprint 8.1 lo creamos con plotly. Esta librería no está adecuadamente soportada en PowerBi por lo que surge un “error” al ejecutarla.

El gráfico se abrirá en una página web (local) pero no será posible verla dentro del informe.



El ejercicio no fue corregido por dos motivos:

1. Quería mostrar las limitaciones de la conexión entre ambos programas.
2. El ejercicio en cuestión pedía que replicáramos los mismos gráficos con el mismo código

Limitaciones en el tamaño

Las visualizaciones hechas con Python en PowerBi se crean con un margen añadido dentro de un recuadro haciendo los gráficos menos intuitivos a tamaños normales.

Por ello he creado un gráfico por página para su correcta visualización