

SPRINT 8.2

Edoardo Brega

★ NIVELL 1

Aquesta tasca consisteix en l'elaboració d'un informe de Power BI, aprofitant les capacitats analítiques de Python. S'utilitzaran els scripts de Python creats prèviament en la Tasca 1 per a generar visualitzacions personalitzades amb les biblioteques Seaborn i Matplotlib. Aquestes visualitzacions seran integrades en l'informe de Power BI per a oferir una comprensió més profunda de la capacitat del llenguatge de programació en l'eina Power BI.

- Els 7 exercicis del nivell 1 de la tasca 01
- Código para cargar las tablas en Power BI.

```
from sqlalchemy import create_engine
import pandas as pd
```

```
host = 'localhost'
database = 'sprint_4_v2'
user = 'root'
password = '3721'
```

```
cadena_conexion = f'mysql+pymysql://{user}:{password}@{host}/{database}'
motor = create_engine(cadena_conexion)
```

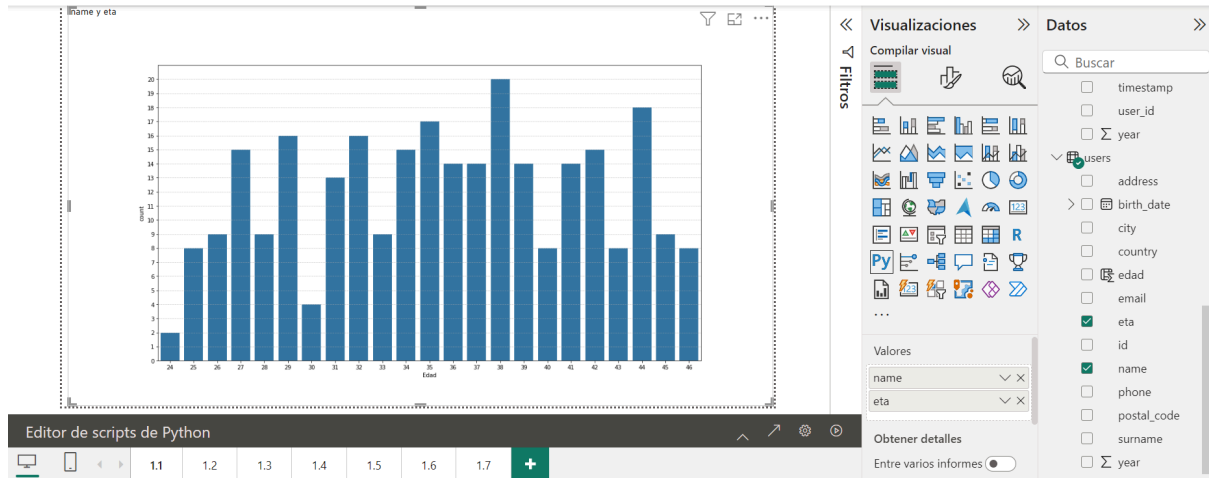
```
nombres_tablas = ['transactions', 'users']
```

```
for nombre in nombres_tablas:
    query = f'SELECT * FROM {nombre}'
    globals()[nombre] = pd.read_sql(query, con=motor)
```

```
transactions = transactions
users = users
```

→ exercici 1

- Una variable numèrica.



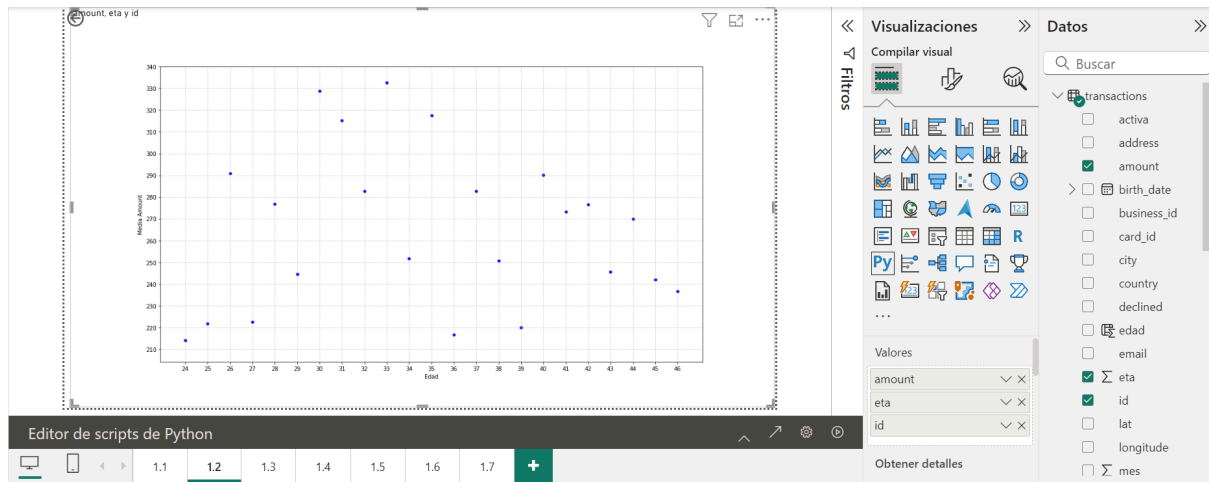
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```
sns.countplot(dataset, x='eta')
plt.xlabel('Edad')
plt.xticks(range(0, int(max(dataset['eta'].value_counts()) + 1, 1))
plt.grid(axis='y', linestyle='--', alpha=0.6)
```

```
plt.show()
```

→ exercici 2

- Dues variables numèriques.



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

mean_amount_by_age = dataset.groupby('eta')['amount'].mean().reset_index()

sns.scatterplot(data=mean_amount_by_age, x='eta', y='amount', color='blue')

plt.xlabel('Edad') # Etiqueta del eje X
plt.ylabel('Media Amount') # Etiqueta del eje Y

plt.grid(True, linestyle='--', alpha=0.6)

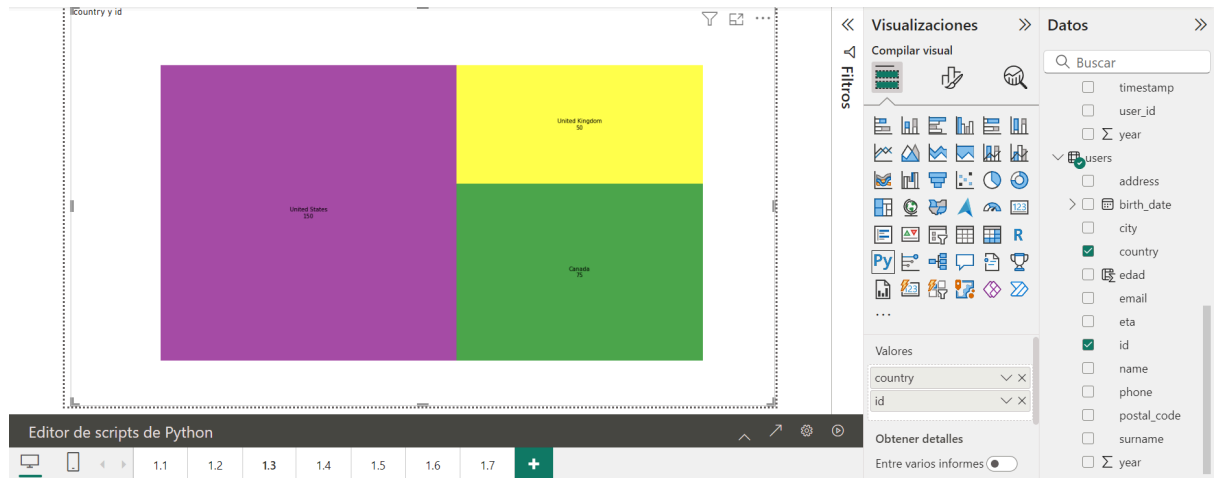
plt.xticks(range(int(mean_amount_by_age['eta'].min()), int(mean_amount_by_age['eta'].max())
+ 1, 1)) # Eje X de 1 en 1
plt.yticks(range(0, int(mean_amount_by_age['amount'].max()) + 10, 10)) # Eje Y de 10 en 10

plt.ylim(bottom=mean_amount_by_age['amount'].min() - 10) # Inicia desde la media más baja
menos 10

plt.show()
```

→ exercici 3

- Una variable categòrica.



```
import matplotlib.pyplot as plt
import pandas as pd
import squarify
```

```
country_counts = dataset['country'].value_counts()
```

```
country_colors = {
    'Canada': 'green',
    'United States': 'purple',
    'United Kingdom': 'yellow'
}
```

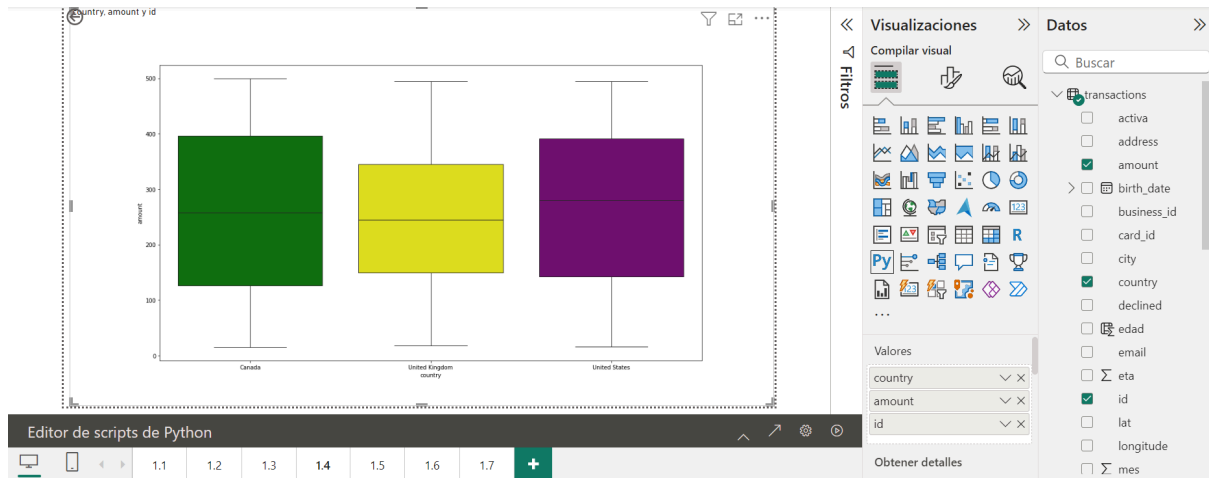
```
squarify.plot(
    sizes=country_counts.values,
    label=[f'{country}\n{count}' for country, count in zip(country_counts.index,
country_counts.values)],
    color=[country_colors.get(country, 'lightgrey') for country in country_counts.index],
    alpha=0.7
)
```

```
plt.axis('off')
```

```
plt.show()
```

→ exercici 4

- Una variable categòrica i una numèrica.



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

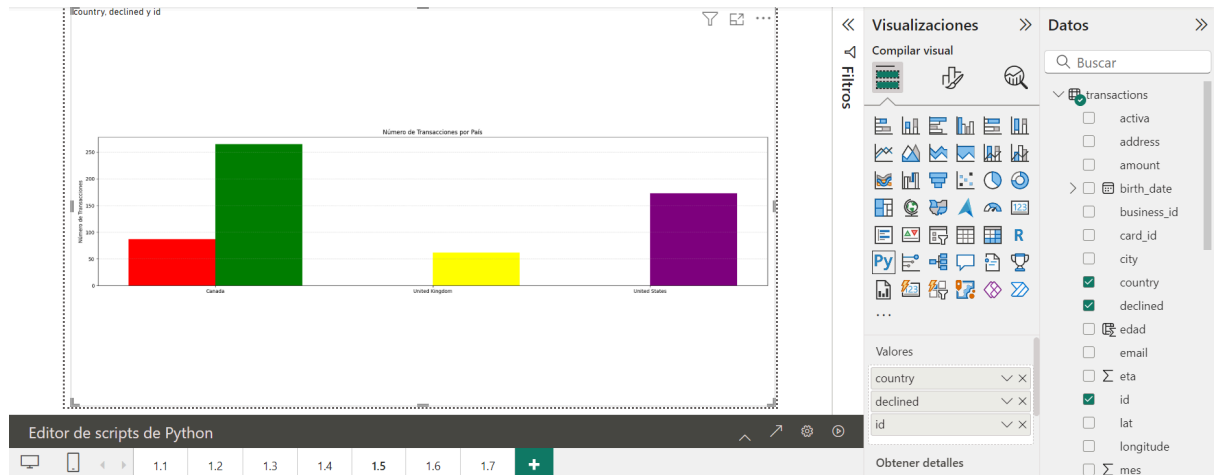
```
country_colors = {
    'Canada': 'green',
    'United States': 'purple',
    'United Kingdom': 'yellow'
}
```

```
sns.boxplot(data=dataset, x='country', y='amount', palette=country_colors)
```

```
plt.show()
```

→ exercici 5

- Dues variables categòriques.



```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

transaction_counts =
dataset.groupby('country')['declined'].value_counts().unstack(fill_value=0)

transaction_counts.columns = ['aceptadas', 'rechazadas']

fig, ax = plt.subplots(figsize=(20, 5)) # Tamaño de la figura

x = np.arange(len(transaction_counts.index))
bar_width = 0.4 # Ancho de las barras

colores_paises = {
    'Canada': 'green',
    'United States': 'purple',
    'United Kingdom': 'yellow'
}

ax.bar(
    x - bar_width / 2, # Desplazo las barras hacia la izquierda
    transaction_counts['rechazadas'], # Altura de las barras basada en el número de
    width=bar_width, # Ancho de las barras
    color='red' # Color rojo para las transacciones rechazadas
)

for i, country in enumerate(transaction_counts.index):
    color = colores_paises.get(country, 'blue') # Asigno un color basado en el país
    ax.bar(
        x[i] + bar_width / 2, # Desplazo las barras hacia la derecha
```

```
        transaction_counts['aceptadas'].iloc[i], # Altura de las barras basada en el
número de transacciones aceptadas
        width=bar_width, # Ancho de las barras
        color=color # Color correspondiente al país
    )

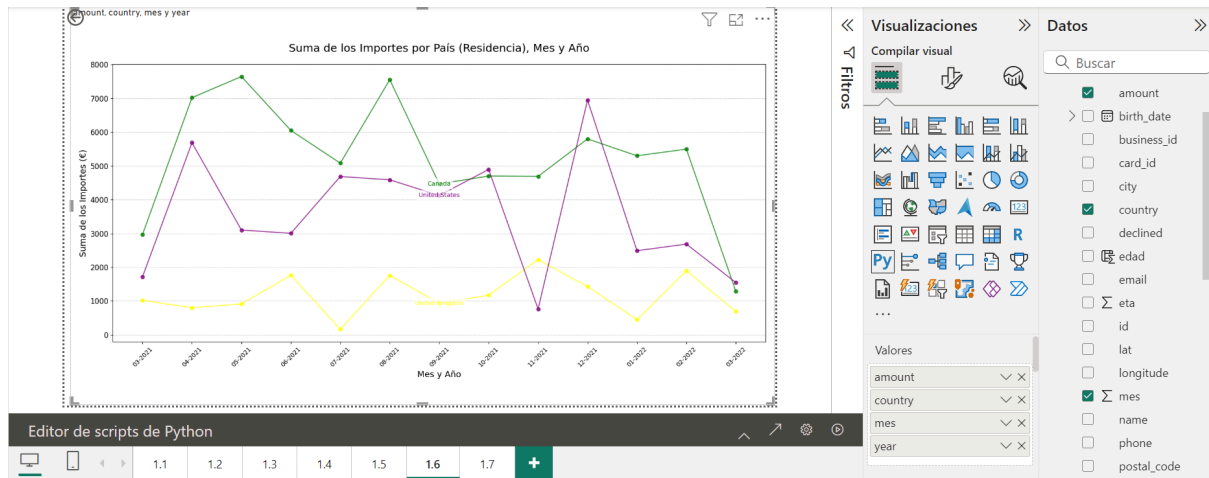
ax.set_xticks(x) # Posición de las etiquetas en el eje X
ax.set_xticklabels(transaction_counts.index) # Etiquetas con los nombres de los
países
ax.set_ylabel('Número de Transacciones') # Etiqueta del eje Y
ax.set_title('Número de Transacciones por País') # Título del gráfico

plt.grid(axis='y', linestyle='--', alpha=0.5)

plt.tight_layout() # Ajusta el espaciado
plt.show()
```


→ exercici 6

- Tres variables.



```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
df_grouped = dataset.groupby(['country', 'year', 'mes'])['amount'].sum().reset_index()
```

```
df_grouped['year_month'] = df_grouped['mes'].astype(str).str.zfill(2) + '-' +  
df_grouped['year'].astype(str)
```

```
country_colors = {  
    'Canada': 'green',  
    'United States': 'purple',  
    'United Kingdom': 'yellow'  
}
```

```
plt.figure(figsize=(16, 8))
```

```
for country in df_grouped['country'].unique():  
    country_data = df_grouped[df_grouped['country'] == country]
```

```
color = country_colors.get(country, 'blue')
```

```
line, = plt.plot(  
    country_data['year_month'],
```

```
    country_data['amount'],
```

```
    marker='o',
```

```
    label=country,
```

```
    color=color,
```

```
    alpha=0.8
```

```

)

num_points = len(country_data)
random_idx = num_points // 2 # Posición central

x_pos = country_data['year_month'].iloc[random_idx]
y_pos = country_data['amount'].iloc[random_idx]

plt.text(
    x_pos,

    y_pos,

    country,

    fontsize=10,

    color=line.get_color(),

    ha='right' if random_idx == 0 else ('left' if random_idx == -1 else 'center'),

    va='center',

    bbox=dict(facecolor='white', alpha=0.7, edgecolor='none')
)

plt.title('Suma de los Importes por País (Residencia), Mes y Año', fontsize=18, pad=20)
plt.xlabel('Mes y Año', fontsize=14)
plt.ylabel('Suma de los Importes (€)', fontsize=14)

plt.xticks(
    df_grouped['year_month'].unique(),
    fontsize=10,
    rotation=45
)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.6)

plt.tight_layout()
plt.show()

```

→ exercici 7

- Graficar un Pairplot.



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df_pairplot = dataset[['amount', 'year', 'eta', 'country']]
```

```
country_colors = {
    'Canada': 'green',
    'United States': 'purple',
    'United Kingdom': 'yellow'
}
```

```
g = sns.pairplot(df_pairplot, hue='country', markers='o', palette=country_colors)
```

```
g._legend.remove()
```

```
g.add_legend(title='Country')
g._legend.set_bbox_to_anchor((0.5, 1.1)) # Posiciono la leyenda arriba
g._legend.set_frame_on(False)
```

```
for ax in g.axes.flatten():
    ax.xaxis.set_major_locator(plt.MaxNLocator(integer=True))
    ax.xaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'{int(x)}'))

    ax.yaxis.set_major_locator(plt.MaxNLocator(integer=True)) # Solo números enteros en el eje Y
    ax.yaxis.set_major_formatter(plt.FuncFormatter(lambda y, _: f'{int(y)}'))
```

```
plt.tight_layout()
```

```
plt.show()
```