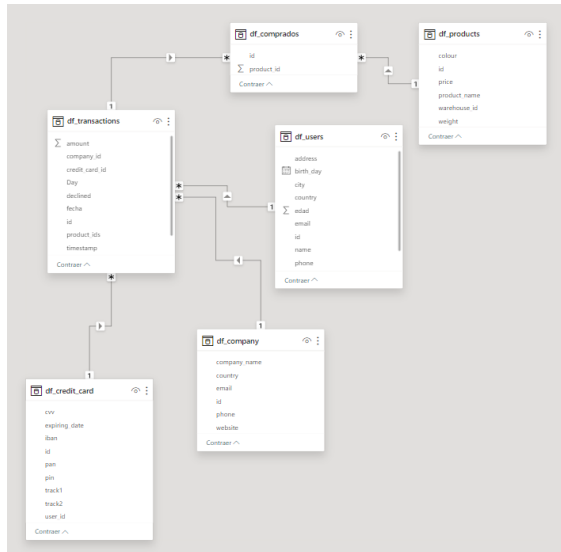


Aquesta tasca consisteix en l'elaboració d'un informe de Power BI, aprofitant les capacitats analítiques de Python. S'utilitzaran els scripts de Python creats prèviament en la Tasca 1 per a generar visualitzacions personalitzades amb les biblioteques Seaborn i Matplotlib. Aquestes visualitzacions seran 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.

Nivell 1

Diagrama:



En los 3 primeros ejercicios el código apenas cambio. Simplemente se tomaron las columnas correctas de Power BI para crear un dataset adecuado para cada una de las visualizaciones. En el ejercicio 4,6 y 7 elimine además los merge del código.

Ejemplo de uso:

*Ejercicio 4: Treemap

1. Columnas visualización



2.Código (en este caso puse comentario en el figsize porque no funciona correctamente con Power BI y hace que la imagen quede pequeña. Destacar que hace falta instalar la librería squarify para que funcione.)

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

```

transacciones_por_país = dataset[dataset['declined'] == 0]
transacciones_por_país = transacciones_por_país.groupby('country')['amount'].sum().reset_index()

# Ordenar por 'amount' en orden descendente
sorted_data = transacciones_por_país.sort_values(by='amount', ascending=False)

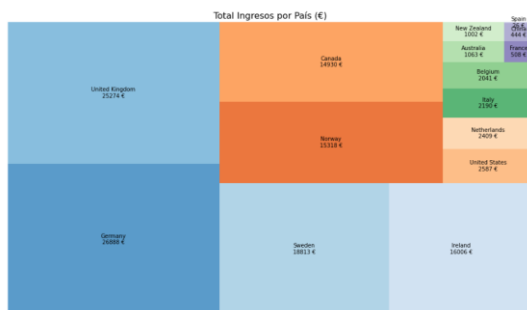
# Renombrar las columnas para abreviar el código
amount_sorted = sorted_data['amount'].astype(int)
country_sorted = sorted_data['country']

# Crear etiquetas combinando 'country' y 'amount' con el símbolo de euros
labels = country_sorted + '\n' + amount_sorted.astype(str) + ' €'

# plt.figure(figsize=(25, 10))
squarify.plot(sizes=amount_sorted, label=labels, alpha=0.8, color=plt.cm.tab20c.colors)
plt.title('Total Ingresos por País (€)', fontsize=16)
plt.axis('off') # Quitar los ejes
plt.show()

```

4. Treemap



*Ejercicio 5: Barras apiladas

1. Columnas visualización

Valores

country

×

|

>

declined

×

|

>

id

×

|

>

Agregar datos

2. Código

```

import pandas as pd
import matplotlib.pyplot as plt

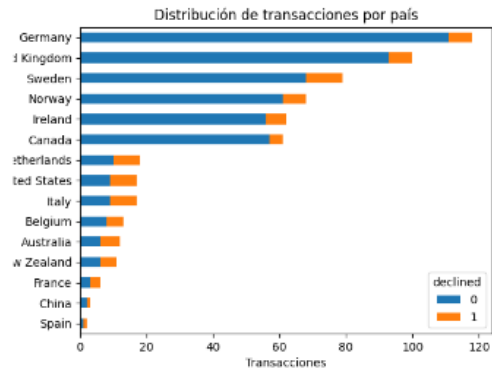
# Reemplazar 'transactions_and_companies' con 'dataset'
oper_por_país = dataset.groupby(['country', 'declined']).size().unstack()
oper_por_país['total_transactions'] = oper_por_país[0] + oper_por_país[1]
oper_por_país = oper_por_país.sort_values(by='total_transactions', ascending=True)
oper_por_país.drop('total_transactions', axis=1).plot(kind='barh', stacked=True)
plt.title('Distribución de transacciones por país')
plt.xlabel('Transacciones')
plt.ylabel('País')

```

```
plt.show()
```

3. Resultado (En este caso por algún motivo los nombres de los países aparecen cortados)

5. Barras apiladas



*Ejercicio 6: Diagrama de Dispersión

1. Columnas visualización

Valores

id	×		>
amount	×		>
id	×		>
country	×		>

Corrección: Se aplicó un filtro directo en Power BI de declined = 0 para que la visualización encajara con la realizada en el ejercicio 8.1. Dado que no lo había agregado en el código adaptado.

2. Código

```
import matplotlib.pyplot as plt
import seaborn as sns
bubble_chart_grouped = dataset.groupby('id').agg(
    total_amount=('amount', 'sum'),
    total_trans= ('id.1', 'count'),
    country = ('country','first')
)

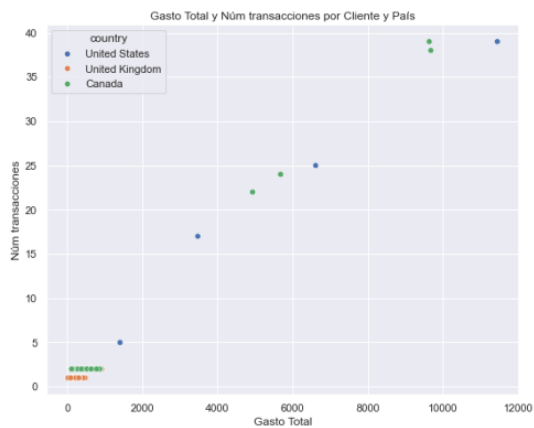
sns.set_theme(style="darkgrid")

sns.scatterplot(
    data= bubble_chart_grouped,
    x="total_amount",
    y="total_trans",
    hue="country",
    style="country"
)
```

)

3. Resultado

6. Diagrama de Dispersión



*Ejercicio 7: Pairplot

1. Tomo las columnas que utilizo en la visualización

Valores

amount

×

|

>

id

×

|

>

edad

×

|

>

country

×

|

>

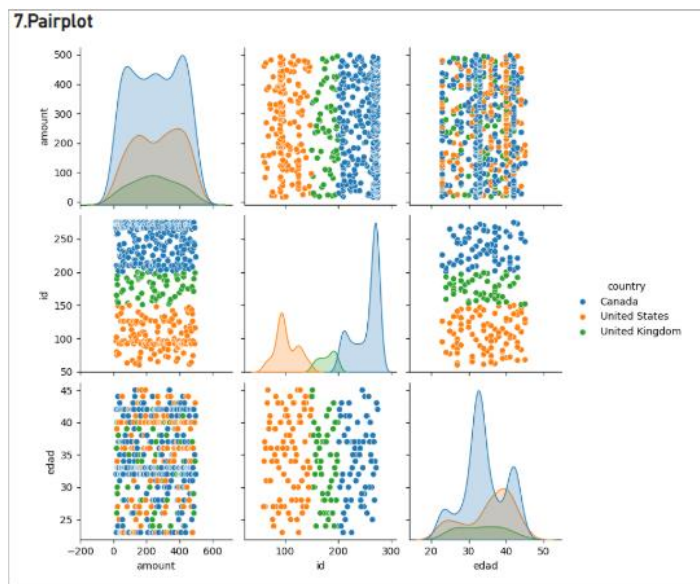
Corrección: tenía el amount con la agregación ‘Suma de amount’, la solución fue poner la agregación ‘No resumir’.

2. Inserto el código adaptado (en este caso, se simplifica porque no es necesario usar merge con varias tablas como en el código original)

```
# dataset = pandas.DataFrame(amount, id, edad, country)

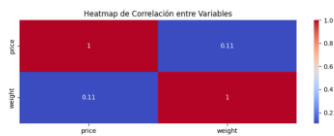
# Pegue o escriba aquí el código de script:
import matplotlib.pyplot as plt
import seaborn as sns
sns.pairplot(data=dataset[['amount', 'id', 'edad', 'country']], hue= 'country')
plt.show()
```

3.Resultado

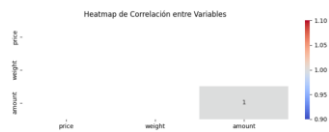


Nivell 2

*Ejercicio 1: Heatmap



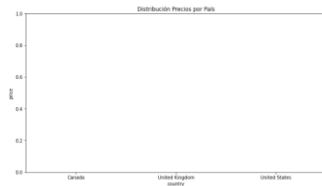
Respecto al Heatmap del 8.1 he decidido eliminar las columnas no cuantitativas como son users_id y declined.). En Power BI no pude agregar la columna amount de la tabla 'transactions' dado que al aplicarlo me desaparecerían el resto de resultados. Como pongo a continuación.



Se ha comprobado que el error ocurre solo cuando inserto datos de la tabla transaction. Esto apuntaría algún error de conexión entre tablas o de formato, pero no he encontrado nada.

Nivell 3

Ejercicio 1: Violinplot combinado



```
# dataset = pandas.DataFrame(country, price)
# dataset = dataset.drop_duplicates()

# Pegue o escriba aquí el código de script:

import matplotlib.pyplot as plt
import seaborn as sns
sns.violinplot(data= dataset, y='price', x='country', hue='country')
sns.swarmplot(data = dataset, y = 'price', x = 'country', color="black", alpha=0.7)
plt.title('Distribución Precios por País')
plt.show()
```

Me vuelve a dar error una visualización que contiene 'price' de la tabla products, pero sigo sin detectar el error.

Ejercicio 2: Facetgrid

1. Columnas visualización

Valores

country

×

|

>

Year

×

|

>

amount

×

|

>

id

×

|

>

2. Código

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('white')
g = sns.FacetGrid(dataset, col= 'country', row= 'Year')
g.map(sns.histplot,'amount', palette = 'inferno', hue = False)
g.set_titles(col_template = '{col_name}', row_template= '{row_name}')

plt.show()
```

3. Resultado

