SPRINT 8.2 – POWER BI + PYTHON

ALUMNO : RODRIGO PADILLA

MATERIA: : Visualizaciones con PBI y Python

GITHUB: https://github.com/ErrePad/Sprint8

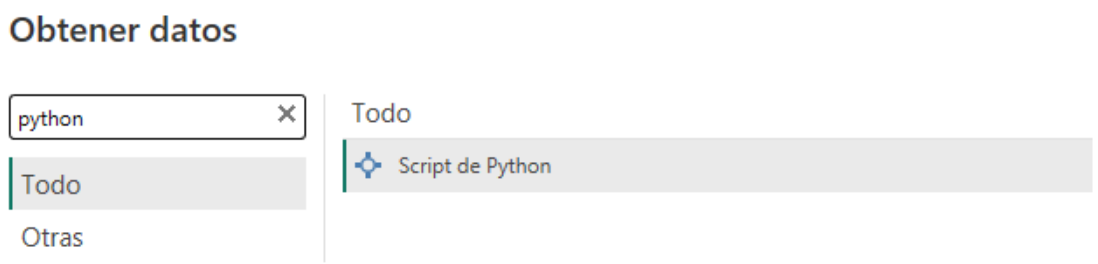
Descripció

Esta labor consiste en la elaboración de un informe de Power BI, aprovechando las capacidades analíticas de Python. Se utilizarán los scripts de Python creados previamente en la Tarea 1 para generar visualizaciones personalizadas con las bibliotecas Seaborn y Matplotlib. Estas visualizaciones estarán integradas en el informe de Power BI para ofrecer una comprensión más profunda de la capacidad del lenguaje de programación en la herramienta Power BI.

Preparación previa de la conexión de datos en PBI con Python.

En primer lugar debemos cargar los datos en PBI mediante un Script de Python. Esto se puede realizar en :

Obtener datos,



Acá pegaremos nuestro código de conexión de Python que previamente utilizamos en Colab, pero ahora agregaremos los 4 dataframe utilizados, según la definición que establecimos como estrategia de data.

Script de Python

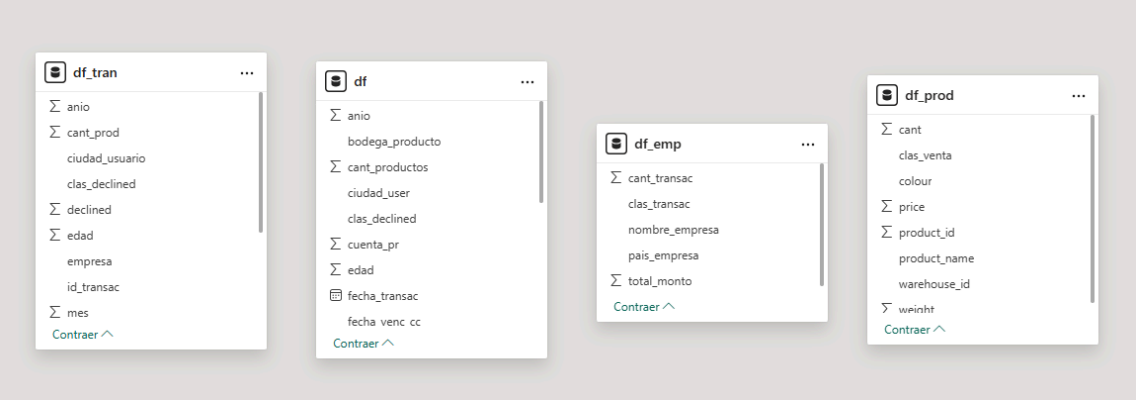
Script

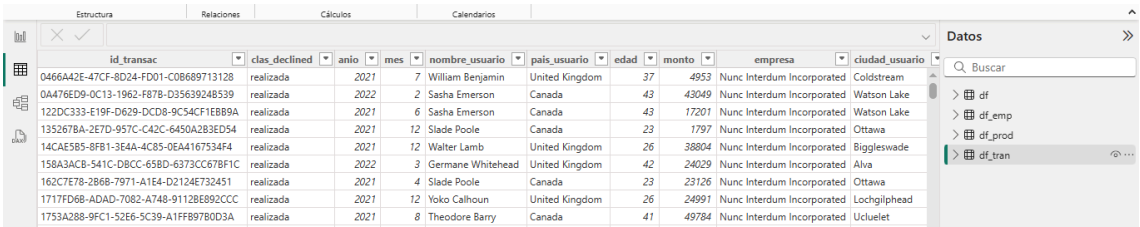
```
    host=host,
    user=user,
    password=password,
    database=database
)

df_products = pd.read_sql("select * from products", connection)

df_transactions = pd.read_sql("select * from transactions", connection)

df_emp = pd.read_sql("""
            select c.company_name nombre_empresa, c.country pais_empresa,
            case when count(transac_id) <= 5 then "Bajas Transacciones"
```

El script se ejecutará con la instalación de Python siguiente:
C:\Users\admin\AppData\Local\Programs\Python\Python313.
Para establecer la configuración y cambiar la instalación de Python que quiere ejecutar, vaya a Opciones y configuración.

Aceptar Cancelar

En este caso no relacionamos los dataframe uno con otro en el modelo PBI, por ya estar relacionado internamente por cada consulta SQL con el que se construyó el df.. Es decir cada uno es un dataset por sí mismo, que conectará con el dataset al crear el objeto visual de Python..





id_transac	clas_declined	anio	mes	nombre_usuario	pais_usuario	edad	monto	empresa	ciudad_usuario
0466A42E-47CF-8D24-FD01-C0B689713128	realizada	2021	7	William Benjamin	United Kingdom	37	4953	Nunc Interdum Incorporated	Coldstream
0A476ED9-0C13-1962-F87B-D3563924B539	realizada	2022	2	Sasha Emerson	Canada	43	43049	Nunc Interdum Incorporated	Watson Lake
122DC333-E19F-D629-DCD8-9C54CF1EBB9A	realizada	2021	6	Sasha Emerson	Canada	43	17201	Nunc Interdum Incorporated	Watson Lake
135267BA-2E7D-957C-C42C-6450A2B3ED54	realizada	2021	12	Slade Poole	Canada	23	1797	Nunc Interdum Incorporated	Ottawa
14CAE5B5-8FB1-3E4A-4C85-0EA4167534F4	realizada	2021	12	Walter Lamb	United Kingdom	26	38804	Nunc Interdum Incorporated	Biggleswade
158A3ACB-541C-DBCC-65BD-6373CC67BF1C	realizada	2022	3	Germane Whitehead	United Kingdom	42	24029	Nunc Interdum Incorporated	Alva
162C7E78-286B-7971-A1E4-D2124E732451	realizada	2021	4	Slade Poole	Canada	23	23126	Nunc Interdum Incorporated	Ottawa
1717FD6B-ADAD-7082-A748-9112BE892CCC	realizada	2021	12	Yoko Calhoun	United Kingdom	26	24991	Nunc Interdum Incorporated	Lochgilphead
1753A288-9FC1-52E6-5C39-A1FFB97B0D3A	realizada	2021	8	Theodore Barry	Canada	41	49784	Nunc Interdum Incorporated	Ucluelet

Consideraciones previas

Por cada Objeto Visual Python, PBI establece una variable dataset que es la que contiene el resultado de las variables vinculadas al informe (arrastramos) al objeto visual.

En este caso, asignamos en dataset a nuestro dataframe respectivo con el que funciona cada gráfico.

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

df_tran = dataset

sns.boxplot(data=df_tran, y="monto", hue="pais_usuario", palette="Dark2")
plt.legend(title="Pais usuario", loc="upper left", bbox_to_anchor=(1.05, 1))
plt.title('Histograma de Montos pagados por país usuario')
plt.ylabel('Frecuencia')
plt.show()
```

De esta forma se programó con Python en 2 áreas funcionales.

- Origen de Datos -> Python Script

  Esta opción permite traer nuestros dataframe establecidos según nuestra estrategia de funcionamiento para la data.

- Objeto Visual de Python ->

  Se importan las librerías y se programa el gráfico, conectando con el dataset del objeto visual de python

EJERCICIOS

.Nivell 1

Los 7 ejercicios del nivel 1 de la tarea 01

EJERCICIO 1

En este primer ejercicio se muestran las partes del código, después no se repetirá la importación de librerías para no extender tanto el texto.

IMPORTACIÓN LIBRERIAS

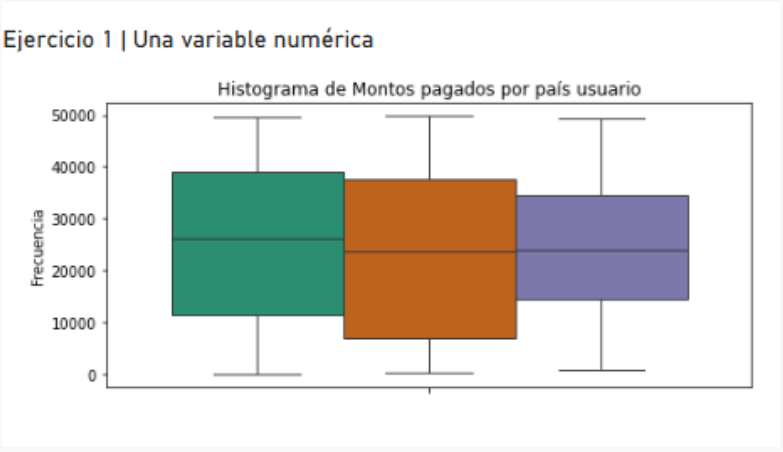
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

DATA

```
df_tran = dataset
```

GRÁFICO

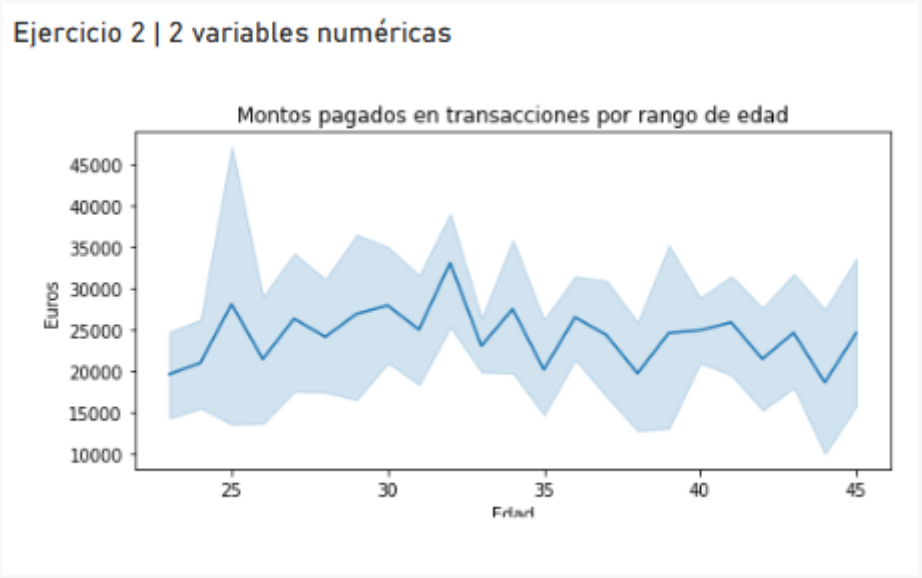
```
sns.boxplot(data=df_tran, y="monto", hue="pais_usuario",
palette="Dark2")
plt.legend(title="Pais usuario", loc="upper left",
bbox_to_anchor=(1.05, 1))
plt.title('Histograma de Montos pagados por país usuario')
plt.ylabel('Frecuencia')
plt.show()
```



Boxplot para identificar la distribución por país del usuario. El ons muestra 5 puntos, el principal, la línea central de la Mediana. También los cuartiles, la mínima y la máxima. No se observan outlier. Las medianas tocan todas las cajas, por tanto son normales las frecuencias de compra de los países de los usuarios.

EJERCICIO 2

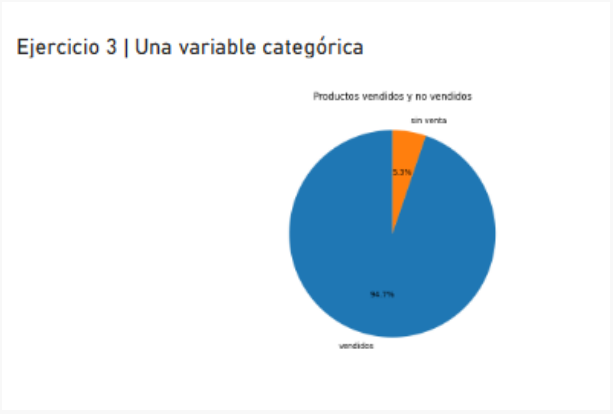
```
df_tran = dataset

sns.lineplot(data=df_tran,y="monto", x="edad")
plt.title('Montos pagados en transacciones por rango de edad')
plt.xlabel('Edad')
plt.ylabel('Euros')
plt.show()
```



Lineplot nos permite medir tendencias respecto a una variable continua en su eje X.

EJERCICIO 3

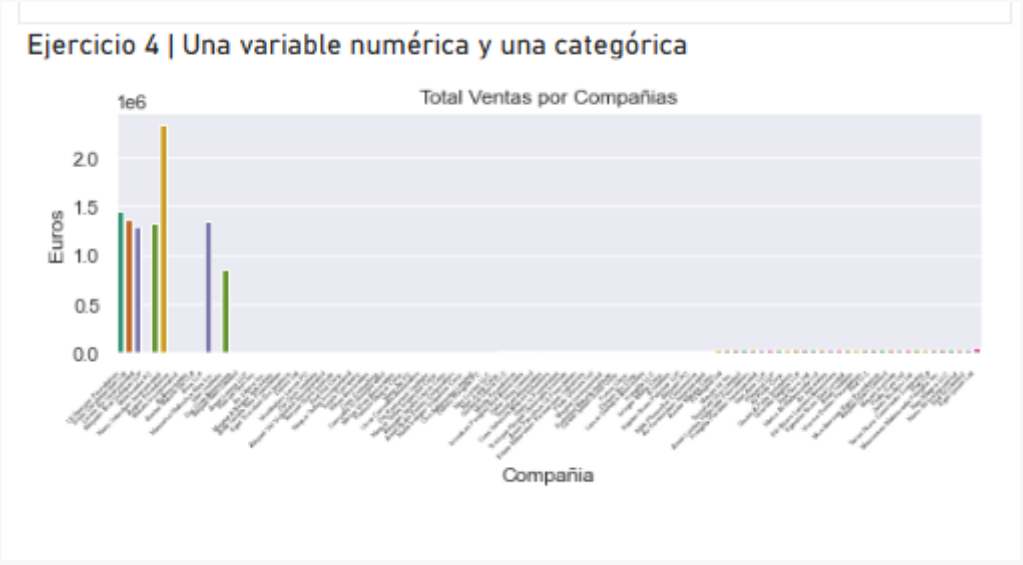
```
df_prod = dataset
counts = df_prod['clas_venta'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(counts, labels=counts.index, autopct='%1.1f%%',
startangle=90)
plt.title('Productos vendidos y no vendidos')
plt.show()
```



Entrando en definiciones más cercanas al negocio, este gráfico es importante porque muestra cómo están las ventas concentradas en tan solo 26 productos. La mejor forma de mostrarlo al ser solo dos opciones, es un gráfico de torta, Pie.

EJERCICIO 4

```
df_tran = dataset


sns.set_theme()
sns.barplot(x="empresa", y="monto", data=df_tran, palette= "Dark2", estimator= "sum", errorbar = None)
plt.title('Total Ventas por Compañias')
plt.ylabel('Euros')
plt.xlabel('Compañia')
plt.xticks(rotation=45, ha='right', fontsize=4)
plt.tight_layout()
plt.show()
```

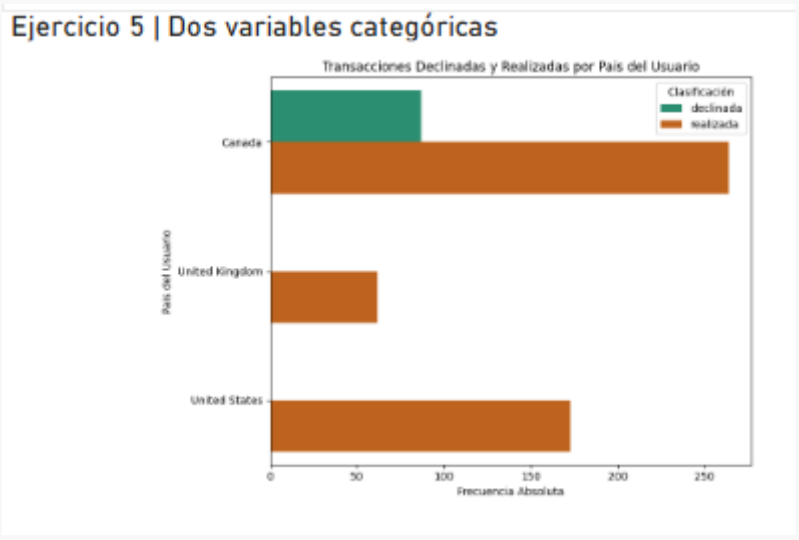


Este ejercicio muestra con claridad como tan solo unas pocas compañías realizan ventas. El gráfico de barras en estos casos, fue el escogido por la gran cantidad de datos con nombres de empresas con ventas ínfimas o nulas.

En este gráfico intentamos resolver el problema del orden con PBI pero no fue posible.

EJERCICIO 5

```
grouped_data = dataset.groupby(['pais_user', 'clas_declined']).size().reset_index(name='frecuencia')

plt.figure(figsize=(8, 6))

sns.barplot(data=grouped_data, y="pais_user", x="frecuencia", hue="clas_declined", palette="Dark2")


plt.title('Transacciones Declinadas y Realizadas por Pais del Usuario')

plt.ylabel('Pais del Usuario')

plt.xlabel('Frecuencia ')

plt.legend(title="Clasificación", loc="best")

plt.tight_layout()

plt.show()
```



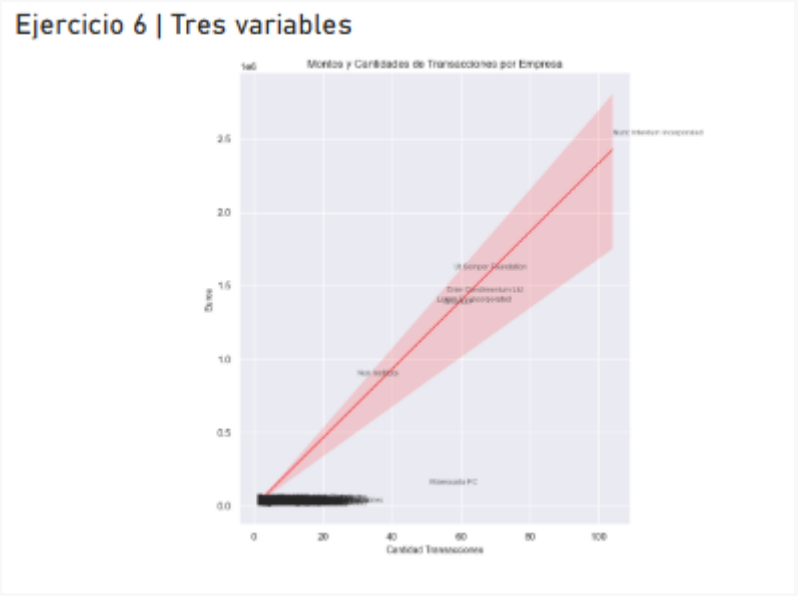
En este ejercicio, ponemos de evidencia como las operaciones declinadas se refieren solo a un país del usuario, este es Canada. La identificación de este insight da oportunidad de focalizar esfuerzos de solución a problemas que causan que estas operaciones no sean exitosas.

Respecto al gráfico, PBI no actuó de la misma forma como lo hace el gráfico en Colab. Esto también ocurrió en el ejercicio 3. Para resolverlo, debimos agregar una variable extra desde PBI. Esto ocurre por algunas definiciones por defecto que tiene PBI. Al agregar una variable extra, pero no visible, el problema se resuelve.

EJERCICIO 6

```
df_emp = dataset


sns.set_theme()

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

#sns.scatterplot(data=df_emp, x='cant_transac', y="total_monto", size="cant_transac", hue="clas_transac", alpha=0.7, #sizes=(100, 600))


sns.regplot(data=df_emp, x='cant_transac', y="total_monto", scatter=False, color='red',line_kws={"linewidth": 1})


for i in range(len(df_emp)):

    plt.text(

        x=df_emp['cant_transac'].iloc[i],

        y=df_emp['total_monto'].iloc[i],
```

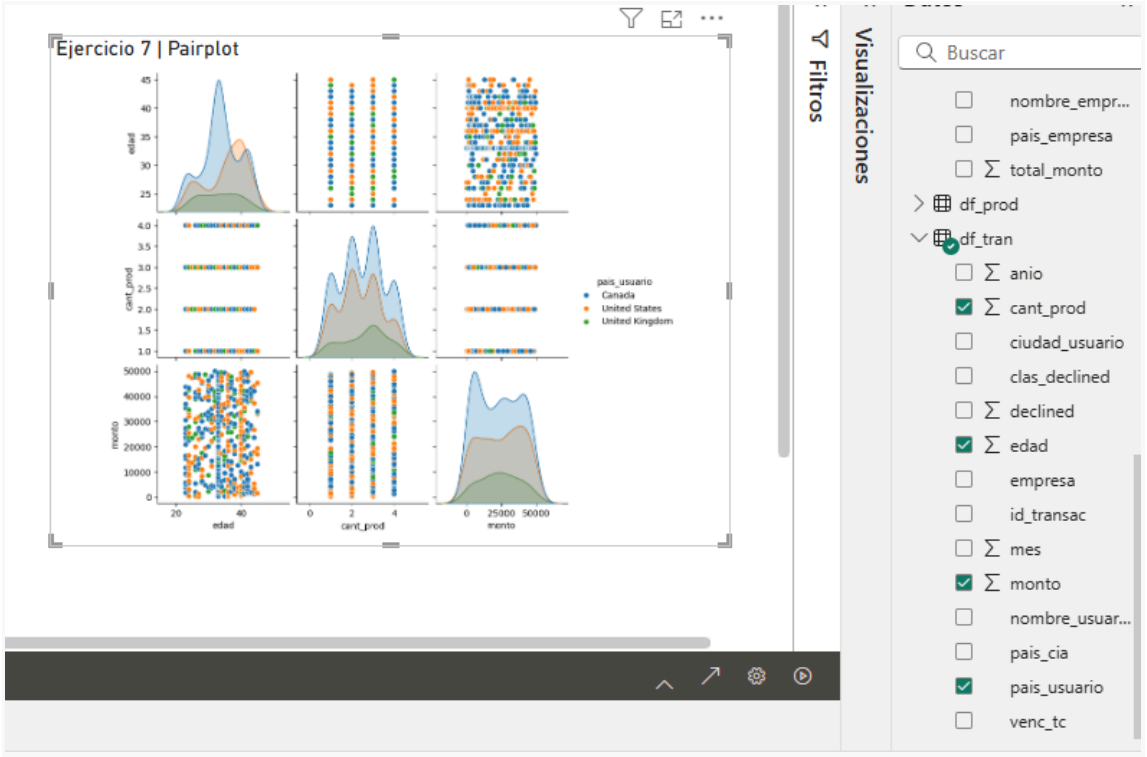
```
            s=df_emp['nombre_empresa'].iloc[i],
            fontsize=8,
            alpha=0.7
        )

plt.title('Montos y Cantidades de Transacciones por Empresa', fontsize=12)
plt.xlabel('Cantidad Transacciones', fontsize=10)
plt.ylabel('Euros', fontsize=10)
plt.tight_layout()
plt.show()
```



Encontrando más insight, logramos encontrar una relación positiva entre dos variables. Lo que confirma la concentración de ventas en algunas empresas. Estas empresas son las que tienen mayor cantidad de transacciones y por ende mayores montos de recaudación. Vemos una pendiente positiva, pero con alta variabilidad. Las tres variables escogidas son cantidad de transacciones, monto total y nombre de empresa.

EJERCICIO 7

```
df_tran = dataset

sns.pairplot(
    df_tran,
    x_vars=["edad", "cant_prod", "monto"],
    y_vars=["edad", "cant_prod", "monto"],
    hue="pais_usuario"
)

plt.show()
```



En este Pairplot, elegimos 3 variables numéricas como son la Edad, Monto, la Cantidad de producto y el País usuario, para poder identificar alguna otra relación. Además mostramos cómo estas columnas aparecen marcadas al ser parte del informe realizado con Python.

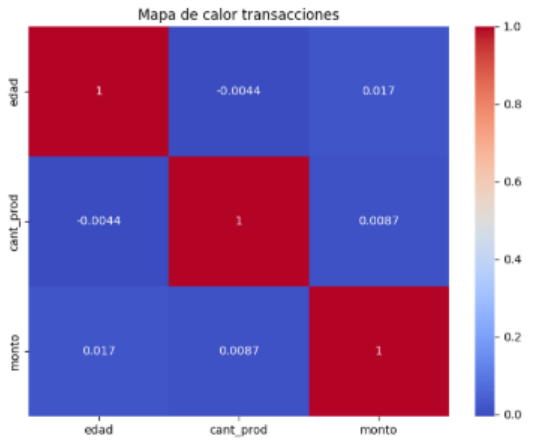
Nivell 2

Los 2 ejercicios del nivel 2 de la tarea 01

N2 EJERCICIO 1

```
df_tran = dataset

plt.figure(figsize=(8, 6))

df_subset = df_tran[['edad', 'cant_prod', 'monto']]

correlation_matrix = df_subset.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Mapa de calor transacciones')
plt.show()
```

N2 Ejercicio 1 | Mapa de Calor

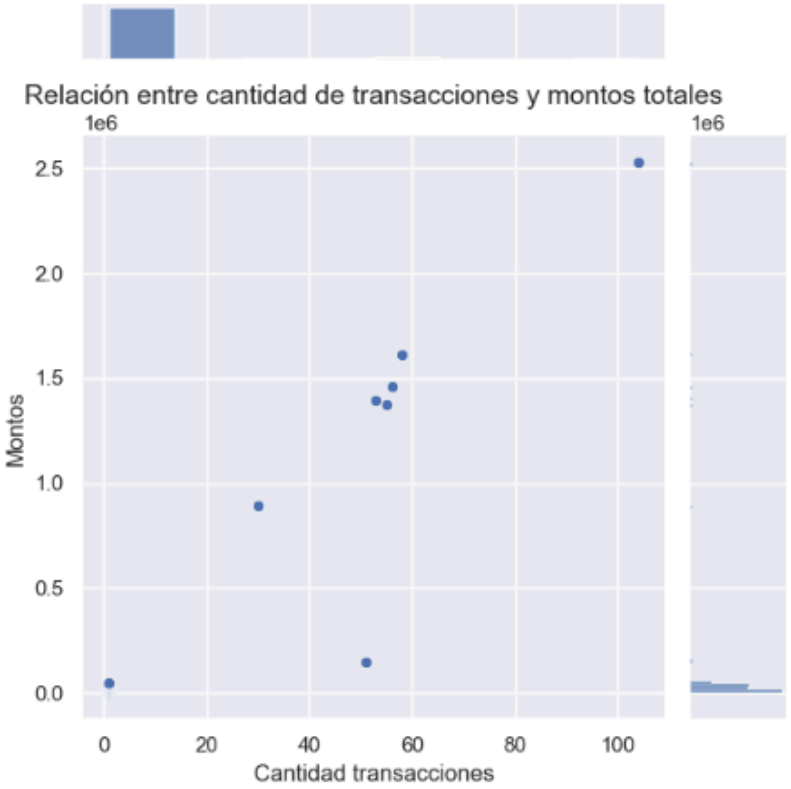


El mapa de calor nos permite medir la intensidad de las relaciones entre las variables numéricas. Acá elegimos la edad, la cantidad de productos por transacción y el monto. El gráfico nos muestra que no existen mediciones considerables como para establecer correlación de alta intensidad.

N2 EJERCICIO 2

```
df_emp = dataset

sns.set_theme()

sns.jointplot(data=df_emp, x="cant_transac", y="total_monto",
palette="Dark2")


plt.title('Relación entre cantidad de transacciones y montos
totales', fontsize=14)

plt.xlabel('Cantidad transacciones', fontsize=12)

plt.ylabel('Montos', fontsize=12)

plt.tight_layout()

plt.show()
```

N2 Ejercicio 2 | Join Plot



El Join Plot de la figura nos muestra cómo en esta relación positiva entre montos y cantidades de transacciones, podemos ahora comprobar su distribución agregada a cada variable. Se complementa, como existe una gran cantidad de frecuencia en montos pequeños. Esto en términos del negocio como se mencionó anteriormente, nos dice que focalizar, fidelizar, puede ser una gran

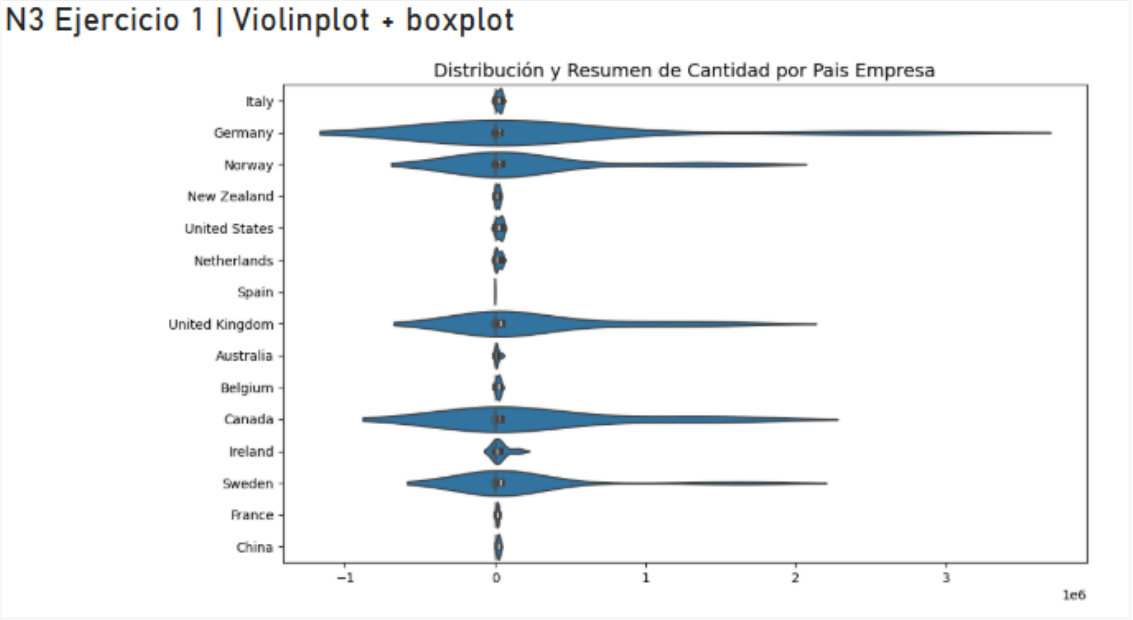
decisión estratégica que permite enfocarse en los high ticket de las empresas más eficientes a la hora de vender.

Nivell 3

Los 2 ejercicios del nivel 3 de la tarea 01

N3 EJERCICIO 1

```
df_emp = dataset


plt.figure(figsize=(10, 6))
sns.violinplot(data=df_emp, x="total_monto", y="pais_empresa", scale="width")


sns.boxplot(
    data=df_emp,
    x="cant_transac",
    y="pais_empresa"
)


plt.title('Distribución y Resumen de Cantidad por Pais Empresa', fontsize=14)
plt.xlabel('', fontsize=12)
plt.ylabel('', fontsize=12)
plt.tight_layout()
plt.show()
```



Bajo una mirada de negocio, Alemania es el país que cuenta con algunas de las empresas más efectivas a la hora de generar ganancias.

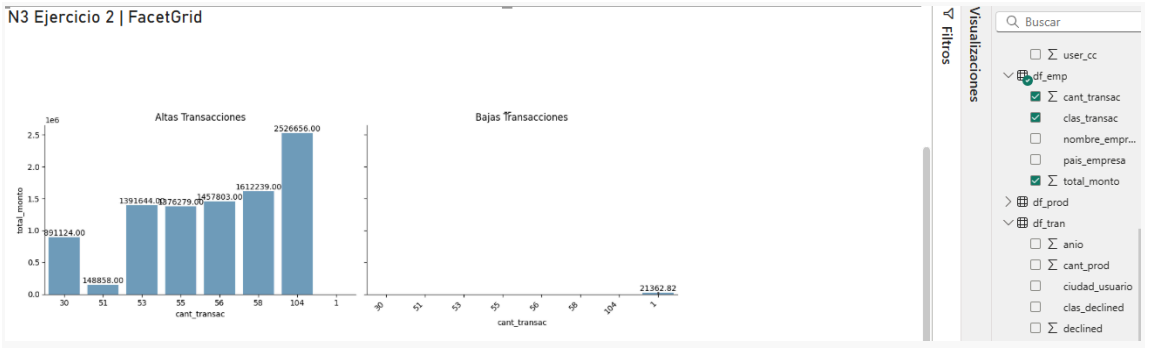
N3 EJERCICIO 2

```
df_emp = dataset
```

```
g = sns.FacetGrid(df_emp, col="clas_transac", col_wrap=3, height=4, aspect=1.5)
g.map_dataframe(sns.barplot, y="total_monto", x="cant_transac", alpha=0.7)
```

```
#agregar etiquetas
for ax in g.axes.flat:
    for container in ax.containers:
        ax.bar_label(container, fmt="%.2f", fontsize=10)
```

```
g.set_titles(col_template="{col_name}", size=12)
g.fig.subplots_adjust(top=0.9)
g.fig.suptitle("--", fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.show()
```



Para el último ejercicio Facet Grid, se reforzó la idea de las empresas más productivas y la concentración de estas, mostrando de manera muy gráfica. Se incluyen en la fotografía las columnas de PBI para el informe.