

1. EJERCICIO 1

```
In [1]: #Importamos las librerías
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

- Para cada archivo csv haga un dataframe en pandas

```
In [2]: #Se cargan los conjuntos de datos del archivo Campañas.csv
Campañas = pd.read_csv('Anexo_1_Campañas.csv', sep=';', encoding='latin-1') #Importamos los datos del archivo Campañas.csv
Métricas = pd.read_csv('Anexo_2_Métricas.csv', sep=';', encoding='latin-1') #Importamos los datos del archivo Métricas.csv
Audencias = pd.read_csv('Anexo_3_Audencias.csv', sep=';', encoding='latin-1') #Importamos los datos del archivo Audencias.csv
df_Campañas = pd.DataFrame(Campañas) #Asignamos los datos al DataFrame
df_Métricas = pd.DataFrame(Métricas) #Asignamos los datos al DataFrame
df_Audencias = pd.DataFrame(Audencias) #Asignamos los datos al DataFrame.
```

```
In [3]: #Convertimos las variables con datos tipo object en datos tipo float
df_Campañas['Presupuesto'] = df_Campañas['Presupuesto'].str.replace('$', '').str.replace(',', '').str.strip().astype(float)

#Convertimos las variables con datos tipo object en datos tipo float
df_Métricas['Tasa_Clics'] = df_Métricas['Tasa_Clics'].str.replace('%', '').str.replace(',', '').str.strip().astype(float)
df_Métricas['Tasa_Conversión'] = df_Métricas['Tasa_Conversión'].str.replace('%', '').str.replace(',', '').str.strip().astype(float)
df_Métricas['Ingreso_Generado'] = df_Métricas['Ingreso_Generado'].str.replace('$', '').str.replace(',', '').str.strip().astype(float)
df_Métricas['Tasa_Rebote'] = df_Métricas['Tasa_Rebote'].str.replace('%', '').str.replace(',', '').str.strip().astype(float)
```

```
In [4]: #Mostrar el DataFrame de Campañas
df_Campañas.head()
```

	ID_Campaña	Nombre	Fecha_Inicio	Fecha_Fin	Plataforma	Presupuesto	Objetivo	Tipo_Campaña	Estado
0	1	Campaña Verano 2023	1/06/2023	30/06/2023	Google Ads	1.2	Aumentar ventas	Búsqueda	Finalizada
1	2	Campaña Redes Sociales	10/07/2023	25/07/2023	Facebook Ads	800.0	Generar engagement	Redes Sociales	Finalizada
2	3	Campaña Lanzamiento Producto	15/08/2023	30/08/2023	Instagram Ads	1.5	Promocionar nuevo producto	Redes Sociales	En Curso
3	4	Campaña Regreso a Clases	1/09/2023	15/09/2023	Google Ads	1.0	Aumentar tráfico	Búsqueda	Planificada
4	5	Campaña Otoño 2023	1/01/2023	31/10/2023	Correo Electrónico	600.0	Fomentar compras	Email Marketing	En Curso

```
In [5]: #Mostrar el DataFrame de Métricas
df_Métricas.head()
```

	ID_Métrica	ID_Campaña	Fecha_Medición	Clics	Impresiones	Tasa_Clics	Conversiones	Tasa_Conversión	Ingreso_Generado	Rebotes	Tasa_rebote
0	1	1	10/06/2023	500	10000	5.00	10	2.00	500.0	245	49.00
1	2	1	15/06/2023	600	11000	5.45	12	2.00	600.0	302	50.33
2	7	1	5/06/2023	400	8000	5.00	8	2.00	400.0	134	33.50
3	8	1	16/06/2023	450	9000	5.00	10	2.22	450.0	308	68.44
4	3	2	10/07/2023	800	15000	5.33	20	2.50	800.0	540	67.50

128 rows x 7 columns

- Use la función "merge" de pandas para agrupar los dataframes, teniendo como eje de agrupación el id de campaña

```
In [7]: #Hacer merge de df_Campañas y df_Métricas usando 'ID_Campaña' como clave
df_merged_1 = pd.merge(df_Campañas, df_Métricas, on='ID_Campaña', how='outer')

#Hacer merge del resultado anterior con df_Audencias usando 'ID_Campaña' como clave
df_merged = pd.merge(df_merged_1, df_Audencias, on='ID_Campaña', how='outer')

#Mostrar el resultado
df_merged.head()
```

	ID_Campaña	Nombre	Fecha_Inicio	Fecha_Fin	Plataforma	Presupuesto	Objetivo	Tipo_Campaña	Estado	ID_Métrica	...	Tasa_Conversión	Ingreso_Generado	Rebotes	Tasa_rebote	ID_Audiencia	Edad	Género	Intereses	Comportamiento
0	1	Campaña Verano 2023	1/06/2023	30/06/2023	Google Ads	1.2	Aumentar ventas	Búsqueda	Finalizada	1.0	...	2.0	500.0	245.0	49.00	1.0	25-34	Femenino	Tecnología, Viajes	Búsqueda de Productos
1	1	Campaña Verano 2023	1/06/2023	30/06/2023	Google Ads	1.2	Aumentar ventas	Búsqueda	Finalizada	1.0	...	2.0	500.0	245.0	49.00	2.0	18-24	Masculino	Moda, Entretenimiento	Interacción en Redes
2	1	Campaña Verano 2023	1/06/2023	30/06/2023	Google Ads	1.2	Aumentar ventas	Búsqueda	Finalizada	2.0	...	2.0	600.0	302.0	50.33	1.0	25-34	Femenino	Tecnología, Viajes	Búsqueda de Productos
3	1	Campaña Verano 2023	1/06/2023	30/06/2023	Google Ads	1.2	Aumentar ventas	Búsqueda	Finalizada	2.0	...	2.0	600.0	302.0	50.33	2.0	18-24	Masculino	Moda, Entretenimiento	Interacción en Redes
4	1	Campaña Verano 2023	1/06/2023	30/06/2023	Google Ads	1.2	Aumentar ventas	Búsqueda	Finalizada	7.0	...	2.0	400.0	134.0	33.50	1.0	25-34	Femenino	Tecnología, Viajes	Búsqueda de Productos

5 rows x 25 columns

```
In [8]: #Reemplazar los valores nulos por 0
df_merged.fillna(0, inplace=True)

#Contar los valores nulos en cada columna del DataFrame df_merged
null_counts = df_merged.isnull().sum()

#Mostrar el resultado
null_counts
```

```
Out[8]: ID_Campaña      0
Nombre              0
Fecha_Inicio        0
Fecha_Fin           0
Plataforma          0
Presupuesto         0
Objetivo            0
Tipo_Campaña        0
Estado              0
ID_Métrica          0
Fecha_Medición      0
Clics               0
Impresiones         0
Tasa_Clics          0
Conversiones        0
Tasa_Conversión     0
Ingreso_Generado    0
Rebotes             0
Tasa_rebote         0
ID_Audiencia        0
Edad                0
Género              0
Intereses           0
Comportamiento      0
Ubicación           0
dtype: int64
```

```
In [9]: #Exportar dataframe a archivo excel o csv
df_merged_final.to_excel('df_merged_final_2.xlsx', index=False)
```

2. EJERCICIO 2. MÉTRICAS

Use pandas para hacer las siguientes consultas:

- Métricas de una campaña específica.

```
In [10]: #Se crea dataframe con las métricas de la campaña No 1
campana_métricas = df_merged[df_merged['ID_Campaña'] == 1]

#Muestra las métricas descriptivas de la campaña No 1
campana_métricas.describe().round(2).transpose()
```

	count	mean	std	min	25%	50%	75%	max
ID_Campaña	8.0	1.00	0.00	1.0	1.00	1.00	1.00	1.00
Presupuesto	8.0	1.20	0.00	1.2	1.20	1.20	1.20	1.20
ID_Métrica	8.0	4.50	3.25	1.0	1.75	4.50	7.25	8.00
Clics	8.0	487.50	79.06	400.0	437.50	475.00	525.00	600.00
Impresiones	8.0	9500.00	1195.23	8000.0	8750.00	9500.00	10250.00	11000.00
Tasa_Clics	8.0	5.11	0.21	5.0	5.00	5.00	5.11	5.45
Conversiones	8.0	10.00	1.51	8.0	9.50	10.00	10.50	12.00
Tasa_Conversión	8.0	2.06	0.10	2.0	2.00	2.00	2.06	2.22
Ingreso_Generado	8.0	487.50	79.06	400.0	437.50	475.00	525.00	600.00
Rebotes	8.0	247.25	74.68	134.0	217.25	273.50	303.50	308.00
Tasa_rebote	8.0	50.32	13.23	33.5	45.12	49.66	54.86	68.44
ID_Audiencia	8.0	1.50	0.53	1.0	1.00	1.50	2.00	2.00

- Métricas de clics e impresiones para todas las campañas.

```
In [11]: #Se crea dataframe con las métricas de clics e impresiones de todas las campañas
clics_impresiones_métricas = df_merged[['Nombre', 'Clics', 'Impresiones']]

#Muestra las métricas descriptivas de todas las campañas
clics_impresiones_métricas.describe().round(2).transpose()
```

	count	mean	std	min	25%	50%	75%	max
Clics	928.0	510.29	236.11	0.0	300.0	500.0	700.0	900.0
Impresiones	928.0	11688.58	4871.93	0.0	8000.0	12000.0	16000.0	19000.0

- Audiencias de una campaña específica.

```
In [12]: #Se crea dataframe con la audiencia de todas las campañas
Audiencia_campana = df_merged[['ID_Audiencia', 'ID_Campaña', 'Edad', 'Género', 'Intereses', 'Comportamiento', 'Ubicación']]

#Se crea dataframe con la audiencia de la campaña No 1
Audiencia_campana_1 = Audiencia_campana[Audiencia_campana['ID_Campaña'] == 2]

#Mostrar las audiencias de la campaña No 1
Audiencia_campana_1
```

	ID_Audiencia	ID_Campaña	Edad	Género	Intereses	Comportamiento	Ubicación
8	3.0	2	35-44	Femenino	Salud, Bienestar	Compras en línea	D
9	4.0	2	18-24	Femenino	Música, Deportes	Navegación de Noticias	A
10	7.0	2	18-24	Femenino	Tecnología, Gaming	Interacción en Redes	A
11	3.0	2	35-44	Femenino	Salud, Bienestar	Compras en línea	D
12	4.0	2	18-24	Femenino	Música, Deportes	Navegación de Noticias	A
13	7.0	2	18-24	Femenino	Tecnología, Gaming	Interacción en Redes	A

```
In [13]: #Obtener las métricas estadísticas de las variables categóricas
audiencia_campana_1_stats = Audiencia_campana_1.describe(include=['object']).transpose()
```

```
#Mostrar el resultado
audiencia_campana_1_stats
```

	count	unique	top	freq
Edad	6	2	18-24	4
Género	6	1	Femenino	6
Intereses	6	3	Salud, Bienestar	2
Comportamiento	6	3	Compras en línea	2
Ubicación	6	2	A	4

- Campañas activas.

```
In [14]: #Se crea dataframe con las métricas de la campaña No 2
Campana_activa = df_merged[df_merged['Estado'] == 'En Curso']

#Mostrar el resultado
Campana_activa.head()
```

	ID_Campaña	Nombre	Fecha_Inicio	Fecha_Fin	Plataforma	Presupuesto	Objetivo	Tipo_Campaña	Estado	ID_Métrica	...	Tasa_Conversión	Ingreso_Generado	Rebotes	Tasa_rebote	ID_Audiencia	Edad	Género	Intereses	Comportamiento
14	3	Campaña Lanzamiento Producto	15/08/2023	30/08/2023	Instagram Ads	1.5	Promocionar nuevo producto	Redes Sociales	En Curso	5.0	...	2.14	700.0	498.0	71.14	5.0	45-54	Masculino	Finanzas, Negocios	Búsqueda de Información
15	3	Campaña Lanzamiento Producto	15/08/2023	30/08/2023	Instagram Ads	1.5	Promocionar nuevo producto	Redes Sociales	En Curso	5.0	...	2.14	700.0	498.0	71.14	6.0	25-34	Masculino	Viajes, Gastronomía	Búsqueda de Oferta
16	3	Campaña Lanzamiento Producto	15/08/2023	30/08/2023	Instagram Ads	1.5	Promocionar nuevo producto	Redes Sociales	En Curso	5.0	...	2.14	700.0	498.0	71.14	8.0	25-34	Masculino	Deportes, Fitness	Compras en línea
17	3	Campaña Lanzamiento Producto	15/08/2023	30/08/2023	Instagram Ads	1.5	Promocionar nuevo producto	Redes Sociales	En Curso	6.0	...	2.13	750.0	196.0	26.13	5.0	45-54	Masculino	Finanzas, Negocios	Búsqueda de Información
18	3	Campaña Lanzamiento Producto	15/08/2023	30/08/2023	Instagram Ads	1.5	Promocionar nuevo producto	Redes Sociales	En Curso	6.0	...	2.13	750.0	196.0	26.13	6.0	25-34	Masculino	Viajes, Gastronomía	Búsqueda de Oferta

5 rows x 25 columns

```
In [ ]: #Agrupar el DataFrame por las columnas 'género' y 'ubicación'
Grupo_Campana_activa = Campana_activa.groupby(['Nombre', 'Estado'])
df_campana_activa = Grupo_Campana_activa.size().unstack()

#Mostrar el resultado
df_campana_activa
```

	Estado	En Curso
	Nombre	
	Campaña Lanzamiento Producto	6
	Campaña Otoño 2023	4

- El rendimiento promedio de conversión por tipo de campaña

```
In [ ]: #Se crea dataframe con el rendimiento promedio de conversión por tipo de campaña
tipo_campana_conver = df_merged[['Tipo_Campaña', 'Tasa_Conversión']].groupby('Tipo_Campaña').mean().round(2)
tipo_campana_conver.sort_values(by='Tasa_Conversión', inplace=True, ascending=False)

df_tipo_campanas_conver = pd.DataFrame(tipo_campana_conver)

#Mostrar el resultado
df_tipo_campanas_conver
```

	Tasa_Conversión
Tipo_Campaña	
Email Marketing	3.31
Redes Sociales	2.71
Búsqueda	2.46

```
In [ ]: #Graficar el rendimiento promedio de conversión por tipo de campaña
plt.figure(figsize=(8, 5))
df_tipo_campanas_conver.plot(kind='bar', color='SteelBlue', legend=False)
plt.title('Rendimiento promedio de conversión por tipo de campaña')
plt.xlabel('Tipo de Campaña')
plt.ylabel('Tasa de conversión')
plt.xticks(rotation=90)
plt.show()
```

<Figure size 800x500 with 0 Axes>

