

Reto: Preguntas Iniciales

Por Jacobo Hirsch Rodríguez - A00829679, Eryk Elizondo González - A01284899, Cleber Gerardo Pérez Galicia - A01236390, Juan Pablo Bernal Lafarga - A01742342.

Carga de Datos y Limpieza

```
# Cargar librerías
import pandas as pd
import numpy as np
import statistics as st

# Cargar datos
Sales_df = pd.read_csv('ventas.csv')
Customers_df = pd.read_csv('customers_sampled.csv')
Products_df = pd.read_csv('20230223_productos.csv')

# Limpieza
Sales_df.columns = Sales_df.columns.str.lower()
Customers_df.columns = Customers_df.columns.str.lower()
Products_df.columns = Products_df.columns.str.lower()

Sales_df['calmonth_convert'] = pd.to_datetime(Sales_df['calmonth'], format='%Y%m')
```

```
# Ventas
Sales_df.head()
```

	customerid int64	material int64	calmonth int64	uni_box float64	calmonth_convert
0	499920078	9151	201909	0.4364	2019-09-01 00:00:00
1	499920078	2287	201909	3.1701	2019-09-01 00:00:00
2	499920078	4526	201909	0.2818	2019-09-01 00:00:00
3	499920078	14050	201909	0.2642	2019-09-01 00:00:00
4	499920078	1333	201909	2.1134	2019-09-01 00:00:00

5 rows, 5 cols 10 / page

<< < Page 1 of 1 > >>



```
# Clientes
Customers_df.head()
```

	customerid int64	pc_agr_300m flo...	pc_comercial_30...	pc_generales_30...	pc_habitacional_...	pc_habitacional_...	pc_industrial_30...	pc_minero...
0	499920078	0	0	6.11	48.4	37.71	6.28	
1	499920499	0	0	0	89.38	6.39	4.23	
2	499921473	0	1.17	15.51	66.28	16.65	0.39	
3	499921557	0	0	81.14	16.57	1.99	0	
4	499921908	0	0	0	100	0	0	

5 rows, 207 cols 10 / page

<< < Page 1 of 1 > >>



```
# Productos
Products_df.head()
```

	material int64	material_desc ob...	productos_por_e...	brandpresret obj...	prodkey object	brand object	presentation obj...	mlsize int6
0	9465	CIEL EXPRIM LIM...	6	CIEL EXPRIM 600...	AGUA SABORIZA...	CIEL EXPRIM	600 ML NR	
1	14450	PWD FIT ARNDN...	6	POWERADE FIT 5...	ISOTONICOS	POWERADE FIT	500 ML NR PET	
2	148	FANTA UVA 500 ...	24	FANTA 500 ML R...	SABORES INDIVI...	FANTA	500 ML RET	
3	1088	SIERRAZUL AP 3...	4	SIERRAZUL AGU...	AGUA PURIFICAD...	SIERRAZUL AGU...	GALÓN 3.8 LTS.	
4	2142	TOPO CHICO AM ...	6	TOPO CHICO A.M...	AGUA MINERAL I...	TOPO CHICO A.M.	12 OZ. NR VIDRIO	

5 rows, 22 cols 10 / page Page 1 of 1

```
# Ventas 2019
Sales_2019 = Sales_df[Sales_df['calmonth'] < 202001]

# Ventas 2020
Sales_2020 = Sales_df[Sales_df['calmonth'].astype(str).str[:4] == '2020']

# Ventas 2021
Sales_2021 = Sales_df[Sales_df['calmonth'].astype(str).str[:4] == '2021']

# Ventas 2022
Sales_2022 = Sales_df[Sales_df['calmonth'].astype(str).str[:4] == '2022']
```

```
print(f'Hubo {len(Sales_2019)} ventas en 2019')
print(f'Hubo {len(Sales_2020)} ventas en 2020')
print(f'Hubo {len(Sales_2021)} ventas en 2021')
print(f'Hubo {len(Sales_2022)} ventas en 2022')
```

```
Hubo 207318 ventas en 2019
Hubo 633643 ventas en 2020
Hubo 714856 ventas en 2021
Hubo 791293 ventas en 2022
```

1. ¿Qué meses tienen más venta?

```
print('El mes con mayor ventas en 2019 fue:', st.mode(Sales_2019['calmonth']), 'con',
len(Sales_2019[Sales_2019['calmonth']==st.mode(Sales_2019['calmonth'])]), 'ventas')
print('El mes con mayor ventas en 2020 fue:', st.mode(Sales_2020['calmonth']), 'con',
len(Sales_2020[Sales_2020['calmonth']==st.mode(Sales_2020['calmonth'])]), 'ventas')
print('El mes con mayor ventas en 2021 fue:', st.mode(Sales_2021['calmonth']), 'con',
len(Sales_2021[Sales_2021['calmonth']==st.mode(Sales_2021['calmonth'])]), 'ventas')
print('El mes con mayor ventas en 2022 fue:', st.mode(Sales_2022['calmonth']), 'con',
len(Sales_2022[Sales_2022['calmonth']==st.mode(Sales_2022['calmonth'])]), 'ventas')
```

```
El mes con mayor ventas en 2019 fue: 201910 con 53016 ventas
El mes con mayor ventas en 2020 fue: 202010 con 56308 ventas
El mes con mayor ventas en 2021 fue: 202112 con 62598 ventas
El mes con mayor ventas en 2022 fue: 202208 con 68708 ventas
```

En el siguiente conteo del número de ventas por mes se considerará el periodo 2020-2022, pues los registros del año 2019 cubren únicamente las ventas desde agosto hasta diciembre, quedando con 7 meses de registros faltantes.

```

Sales_Dict = {2020: Sales_2020, 2021: Sales_2021, 2022: Sales_2022}
Months = [[202001,202101,202201],[202002,202102,202202],[202003,202103,202203],[202004,202104,202204],
[202005,202105,202205],[202006,202106,202206],[202007,202107,202207],[202008,202108,202208],[202009,202109,202209],
[202010, 202110, 202210],[202011,202111,202211],[202012, 202112, 202212]]
Months_Dict = ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre',
'Noviembre', 'Diciembre']
total_len = 0

print('-----PERIODO DE VENTAS 2020 A 2022-----')
for i in range(12):
    total_len = 0
    for year, month in zip(Sales_Dict.keys(), Months[i]):
        total_len += len(Sales_Dict[year][Sales_Dict[year]['calmonth'] == month])

    print(f"Total de ventas en {Months_Dict[i]}: {total_len}")

```

```

(-----PERIODO DE VENTAS 2020 A 2022-----)
Total de ventas en Enero: 161854
Total de ventas en Febrero: 161694
Total de ventas en Marzo: 176898
Total de ventas en Abril: 174084
Total de ventas en Mayo: 180480
Total de ventas en Junio: 182138
Total de ventas en Julio: 181511
Total de ventas en Agosto: 184465
Total de ventas en Septiembre: 184046
Total de ventas en Octubre: 185164
Total de ventas en Noviembre: 182705
Total de ventas en Diciembre: 184753

```

Los meses con mayor número de ventas en el periodo 2020-2022 han sido: Octubre, seguido de Diciembre y Agosto.

2. ¿En qué meses se vende menos?

```

print('El mes con menor ventas en 2019 fue:', Sales_2019['calmonth'].value_counts().idxmin(),
'con', len(Sales_2019[Sales_2019['calmonth']==Sales_2019['calmonth'].value_counts().idxmin()]), 'ventas')
print('El mes con menor ventas en 2020 fue:', Sales_2020['calmonth'].value_counts().idxmin(),
'con', len(Sales_2020[Sales_2020['calmonth']==Sales_2020['calmonth'].value_counts().idxmin()]), 'ventas')
print('El mes con menor ventas en 2021 fue:', Sales_2021['calmonth'].value_counts().idxmin(),
'con', len(Sales_2021[Sales_2021['calmonth']==Sales_2021['calmonth'].value_counts().idxmin()]), 'ventas')
print('El mes con menor ventas en 2022 fue:', Sales_2022['calmonth'].value_counts().idxmin(),
'con', len(Sales_2022[Sales_2022['calmonth']==Sales_2022['calmonth'].value_counts().idxmin()]), 'ventas')

```

```

El mes con menor ventas en 2019 fue: 201911 con 50742 ventas
El mes con menor ventas en 2020 fue: 202004 con 48097 ventas
El mes con menor ventas en 2021 fue: 202101 con 52246 ventas
El mes con menor ventas en 2022 fue: 202201 con 59312 ventas

```

Además, gracias a la tabla anterior podemos ver que los meses con peores ventas en el periodo 2020-2022 fueron: 1ro Febrero, 2do Enero y 3ro Abril.

3. ¿Cuántas veces un cliente compra un producto que no compraba antes?

Se obtienen cuantos clientes únicos hay.

```

num_customers = Sales_df['customerid'].nunique()

# Mostramos el resultado
print(num_customers)





```

```
2025
```

Se organiza el dataframe por id de cliente y se ordena con fecha de menor a mayor.

```
# Ordenar por customerid y calmonth_convert
Customers_By_Month = Sales_df.sort_values(by=['customerid', 'calmonth'])

Customers_By_Month.head(10)
```

	<div>customerid int64</div> <div>499920078 - 4999...</div> <div></div>	<div>material int64</div> <div>117 - 14050</div> <div></div>	<div>calmonth int64</div> <div>201909 - 201909</div> <div></div>	<div>uni_box float64</div> <div>0.2642 - 15.2166</div> <div></div>	<div>calmonth_convert</div> <div>2019-09-01 00:00:...</div>	
0	499920078	9151	201909	0.4364	2019-09-01 00:00:...	
1	499920078	2287	201909	3.1701	2019-09-01 00:00:...	
2	499920078	4526	201909	0.2818	2019-09-01 00:00:...	
3	499920078	14050	201909	0.2642	2019-09-01 00:00:...	
4	499920078	1333	201909	2.1134	2019-09-01 00:00:...	
5	499920078	117	201909	2.1134	2019-09-01 00:00:...	
6	499920078	520	201909	2.1134	2019-09-01 00:00:...	
7	499920078	2589	201909	1.2681	2019-09-01 00:00:...	
8	499920078	759	201909	0.5284	2019-09-01 00:00:...	
9	499920078	402	201909	15.2166	2019-09-01 00:00:...	

10 rows, 5 cols10 / page<< < Page 1 of 1 > >>⬇

```
# Agrupar por CustomerId y calmonth para juntar las compras del mismo cliente en el mismo mes
grouped = Customers_By_Month.groupby(['customerid', 'calmonth'])['material'].apply(set).reset_index()

# Diccionario para contar cuántos productos nuevos ha comprado cada cliente
new_purchases_count = {}

# Diccionario para almacenar el historial de productos comprados por cliente hasta el mes anterior
purchase_history = {}

# Recorrer cada grupo (cada cliente y mes)
for index, row in grouped.iterrows():
    customer_id = row['customerid']
    products_in_month = row['material'] # Set de productos comprados en el mes actual
    month = row['calmonth']

    # Inicializamos el historial y el contador de nuevos productos si es la primera vez que vemos al cliente
    if customer_id not in purchase_history:
        purchase_history[customer_id] = set() # Historial de productos
        new_purchases_count[customer_id] = 0 # Contador de productos nuevos
        continue # No consideramos el primer mes, pasamos a la siguiente fila

    # Productos que son nuevos (no estaban en el historial hasta el mes anterior)
    new_products = products_in_month - purchase_history[customer_id]

    # Si hay productos nuevos, los añadimos al historial y actualizamos el contador
    if new_products:
        purchase_history[customer_id].update(new_products) # Actualizamos el historial
        new_purchases_count[customer_id] += len(new_products) # Sumamos al contador

# Mostramos el resultado: cuántos productos nuevos ha comprado cada cliente
new_purchases_count = pd.DataFrame(new_purchases_count.items(), columns=['customerid', 'new_purchases_count'])

new_purchases_count
```

	customerid int64 499920078 - 5108...	new_purchases_... 0 - 297	
0	499920078	155	
1	499920499	142	
2	499921473	189	
3	499921557	207	
4	499921908	210	
5	499922150	76	
6	499922756	115	
7	499923595	104	
8	499924161	95	
9	499924210	133	

2025 rows, 2 cols 10 / page << < Page 1 of 203 > >> [↓](#)

La función de arriba regresa un diccionario con la cantidad de veces que los diferentes clientes (las llaves del diccionario) compraron un producto nuevo, la función considera que se compra un producto nuevo y lo analiza mes a mes.

4. ¿Qué productos de lanzamiento tiene la base de datos? [definir por equipo el término producto de lanzamiento]

Definimos un producto de lanzamiento aquel culla primera venta sea dentro de los últimos 6 meses de la base de datos.

```
# Primera venta para cada material
Sales_first = (
    pd.merge(Sales_df, Products_df, on='material', how='inner') # Merge con los datos de los productos
    .groupby('material', as_index=False) # Agrupar por material
    .agg({'material_desc': 'first', 'calmonth_convert': 'min'}) # Obtener la primera fecha de venta
    .sort_values(by='calmonth_convert', ascending=False) # Ordenar por fecha más reciente
)

# Obtener la fecha más reciente y restarle 6 meses
fecha_limite = Sales_df.agg({'calmonth_convert': 'max'})['calmonth_convert'] - pd.DateOffset(months=6)

# Filtrar las ventas que ocurrieron después de la fecha límite
Sales_launch = Sales_first[Sales_first['calmonth_convert'] > fecha_limite]

# Mostrar el resultado
Sales_launch
```

	material int64 26 - 14595	material_desc ob... BT CIEL AP ... 5.3% PULPY DUR... 5.3% 17 others 89.5%	calmonth_convert < 2022-07-01 00:00:...	
215	2184	BT CIEL AP BIDO...	2022-12-01 00:00:...	
479	9380	PULPY DURAZN...	2022-11-01 00:00:...	
663	14434	T CH AGUA MINE...	2022-11-01 00:00:...	
712	14534	COSTA DESCA T...	2022-09-01 00:00:...	
730	14589	JOYA PONCHE 2....	2022-09-01 00:00:...	
477	9366	FZT NG DZO 453...	2022-09-01 00:00:...	
728	14581	TCH HARD S FRE...	2022-08-01 00:00:...	
711	14532	COSTA DESCA T...	2022-08-01 00:00:...	
723	14569	MIX 473 ML LAT ...	2022-07-01 00:00:...	
15	26	FRESCA 250 ML ...	2022-07-01 00:00:...	

19 rows, 3 cols 10 / page << < Page 1 of 2 > >> [↓](#)

--

```
# Obtener el 20% superior de los productos más recientes
Sales_top_20 = Sales_first.head(int(len(Sales_first) * 0.2))

# Mostrar el resultado
Sales_top_20
```

	material int64 26 - 17369	material_desc ob... BT CIEL AP ... 0.7% PULPY DUR... 0.7% 144 others 98.6%	calmonth_convert < 2021-03-01 00:00:...	
215	2184	BT CIEL AP BIDO...	2022-12-01 00:00:...	
479	9380	PULPY DURAZN...	2022-11-01 00:00:...	
663	14434	T CH AGUA MINE...	2022-11-01 00:00:...	
712	14534	COSTA DESCA T...	2022-09-01 00:00:...	
730	14589	JOYA PONCHE 2....	2022-09-01 00:00:...	
477	9366	FZT NG DZO 453...	2022-09-01 00:00:...	
728	14581	TCH HARD S FRE...	2022-08-01 00:00:...	
711	14532	COSTA DESCA T...	2022-08-01 00:00:...	
723	14569	MIX 473 ML LAT ...	2022-07-01 00:00:...	
15	26	FRESCA 250 ML ...	2022-07-01 00:00:...	

146 rows, 3 cols 10 / page << < Page 1 of 15 > >> [↓](#)

```
# Obtener la lista de tipos de productos únicos
Products_list = pd.DataFrame(Products_df['producttype'].unique(), columns=['producttype'])

# Mostrar el número de tipos de productos
print(f"Hay {len(Products_list)} tipos de productos.")

# Mostrar los tipos de productos
print("Los tipos de productos son:")
print(Products_list)
```

Hay 25 tipos de productos.

Los tipos de productos son:

	producttype
0	AGUA SABORIZADA
1	ISOTONICOS
2	SABORES REGULAR
3	AGUA PURIFICADA
4	AGUA MINERAL
5	COLAS REGULAR
6	LECHE UHT SABORIZADA
7	BEBIDAS REFRESCANTES
8	BEBIDAS ENERGETICAS
9	JUGOS Y NECTARES
10	BEBIDAS DE SOYA
11	NARANJADAS
12	SABORES LIGHT
13	BEBIDA CON ELECTROLITOS
14	COLAS LIGHT
15	CAFE MOLIDO
16	MIXTOS
17	TE
18	CAFE GRANO
19	BEBIDAS INFANTILES
20	AGUA FUNCIONAL
21	LECHE UHT ESPECIALIZADA
22	BEBIDA ALCOHOLICA
23	LECHE UHT REGULAR
24	YOGURT BATIDO

```

# Obtener y mostrar el número de clientes únicos
print(f'Existen {Customers_df['customerid'].nunique()} clientes.')

# Separar los valores en 'sub_canal_comercial', explotar los valores y obtener los comercios únicos
Commerce_list = pd.DataFrame(Customers_df['sub_canal_comercial'].str.split(' / ').explode().unique(),
columns=['sub_canal_comercial'])

# Mostrar la cantidad y los tipos de comercios
print(f'Hay {len(Commerce_list)} tipos de comercios.')
print("Los comercios son:")
print(Commerce_list)

```

```

Existen 2041 clientes.
Hay 21 tipos de comercios.
Los comercios son:

```

	sub_canal_comercial
0	Estanquillos
1	kioscos
2	Abarrotes
3	Almacenes
4	Bodegas
5	Viveres
6	Frutas y Verduras
7	Hogar con Venta
8	Carnicería
9	Pollería
10	Pescadería
11	Cerveza y Licores
12	Tortillería
13	Farmacia Independiente
14	Mayorista Abarrotero
15	Panadería
16	Pastelería
17	Minisuper
18	Minimarket
19	Tiendas de Alimentos Especializados Orgánicos
20	TDC/Proximidad Independiente