Guillermo García Hernández
2ºDAW 25/26

UD02A03-Análisis con NumPy, Pandas y Matplotlib (Dataset Mercadona)

¿Cuantas filas y columnas tiene?

```
[94]  print("Filas y columnas:", df.shape)
      print("Guillermo García Hernández   2º DAW")
```

✓ 0.0s Python

```
Filas y columnas: (4723, 9)
Guillermo García Hernández   2º DAW
```

¿Cuáles son sus columnas?

```
[95]  print("Columnas del dataset:")
      print(df.columns.tolist())
      print("Guillermo García Hernández 2º DAW")
```

✓ 0.0s Python

```
Columnas del dataset:
['id', 'Category', 'name', 'subtitle', 'price', 'discount_price', 'main_image_url'
Guillermo García Hernández 2º DAW
```

¿De qué tipo son?

```
[96]  print("Tipos de datos de cada columna:")
      print(df.dtypes)
      print("Guillermo García Hernández 2º DAW")
```

✓ 0.0s Python

```
Tipos de datos de cada columna:
id                    int64
Category             object
name                 object
subtitle             object
price               float64
discount_price      float64
main_image_url       object
secondary_image_url  object
nutritional_info     object
dtype: object
Guillermo García Hernández 2º DAW
```

¿Cómo se indexa este dataset?

```
[97]  print("Indexado:", df.index)
      print("Guillermo García Hernández  2º DAW")
```

✓ 0.0s Python

```
Indexado: RangeIndex(start=0, stop=4723, step=1)
Guillermo García Hernández  2º DAW
```

Usa sample() para sacar 8 al azar.

```
muestra8 = df.sample(8)
print(muestra8)
print("Guillermo García Hernández  2º DAW")
```

[98] ✓ 0.0s Python

```
...           id                            Category  \
1213  1214        Café soluble y otras bebidas
3951  3952                                  Ojos
4601  4602  Menaje y conservación de alimentos
1523  1524                 Chicles y caramelos
1976  1977           Leche y bebidas vegetales
3635  3636                    Fijación cabello
3979  3980                              Labios
3105  3106                            Cerveza

                                                  name  \
1213               Café soluble descafeinado Hacendado
3951  Sombra de ojos Long Lasting Deliplus Multi-Sti...
4601                 Encendedor cocina largo Polyflame
1523                Caramelos blandos Fruits Hacendado
1976                     Leche semidesnatada Asturiana
3635    Espuma cabello Ondas Curly Deliplus fijación 3
3979  Perfilador de labios Long Lasting Deliplus 12 ...
3105                            Cerveza tostada Turia

                     subtitle  price  discount_price  \
1213             Bote 200 g   4.95             NaN
3951                 1 ud.    4.25             NaN
4601          Paquete 1 ud.   2.20             NaN
1523          Paquete 300 g   1.60             NaN
...
3635  Espuma cabello Ondas Curly Deliplus fijación 3...
3979  Perfilador de labios Long Lasting Deliplus 12 ...
3105  Cerveza tostada Turia, 12 botellines x 250 Mil...
Guillermo García Hernández  2º DAW
```

Sacame los 10 últimos

```
ultimos = df.tail(10)
print(ultimos)
print("Guillermo García Hernández  2º DAW")
```

[99] ✓ 0.0s Python

```
...           id   Category                                              name  \
4713  4714  Rebozados  Anillas de pota a la romana Hacendado ultracon...
4714  4715  Rebozados       Muslitos de surimi Hacendado ultracongelados
4715  4716  Rebozados  Palitos de merluza a la romana Hacendado ultra...
4716  4717  Rebozados  Figuritas de merluza empanadas Hacendado ultra...
4717  4718  Rebozados    Chipirones enharinados Hacendado ultracongelado
4718  4719  Rebozados  Langostino caballitos rebozados Hacendado ultr...
4719  4720  Rebozados  Filetes de boquerón en tempura Hacendado ultra...
4720  4721  Rebozados       Rollitos primavera Hacendado ultracongelados
4721  4722  Rebozados         Fingers de queso Hacendado ultracongelados
4722  4723  Rebozados        Tequeños de queso Hacendado ultracongelados

                  subtitle  price  discount_price  \
4713        Paquete 500 g   4.70             NaN
4714        Paquete 450 g   2.60             NaN
4715        Paquete 500 g   2.75             NaN
4716        Paquete 500 g   3.55             NaN
4717        Paquete 350 g   5.05             NaN
4718        Paquete 300 g   3.25             NaN
4719        Paquete 400 g   3.75             NaN
4720   Caja 6 ud. (300 g)   1.55             NaN
4721        Paquete 300 g   2.40             NaN
4722  Caja 12 ud. (480 g)   4.80             NaN

                                        main_image_url  \
...
4720  Rollitos primavera Hacendado ultracongelados, ...
4721  Fingers de queso Hacendado ultracongelados, Pa...
4722  Tequeños de queso Hacendado ultracongelados, C...
Guillermo García Hernández  2º DAW
```

Usame la función rename para cambiar la columna "*Category*" por "*category*"

```
print("Columnas:", df.columns.tolist())
df_minuscula = df.rename(columns={"Category": "category"})
print("Columnas:", df_minuscula.columns.tolist())
print("Guillermo García Hernández  2º DAW")
```

[100] ✓ 0.0s Python

```
Columnas: ['id', 'Category', 'name', 'subtitle', 'price', 'discount_price', 'main_
Columnas: ['id', 'category', 'name', 'subtitle', 'price', 'discount_price', 'main_
Guillermo García Hernández  2º DAW
```

Lanza un describe. ¿Qué conclusiones? ¿Qué significa cada uno de los valores?

```
print(df.describe())
print("Guillermo García Hernández  2º DAW")
# Descripción estadística de las columnas numéricas
#count: nº de valores no nulos
#mean: media
#std: desviación estándar
#min: valor mínimo
#25%: percentil 25
#50%: percentil 50 (mediana)
#75%: percentil 75
#max: valor máximo
```

[101] ✓ 0.0s Python

```
                id        price  discount_price
count  4723.000000  4723.000000      131.000000
mean   2362.000000     3.635281        5.421985
std    1363.556991    10.452852        4.730901
min       1.000000     0.180000        0.650000
25%    1181.500000     1.500000        2.350000
50%    2362.000000     2.350000        3.850000
75%    3542.500000     3.950000        7.000000
max    4723.000000   504.000000       29.950000
Guillermo García Hernández  2º DAW
```

Devuelve aquellos registros que sean de la categoría "Verdura"

```python
# Filas donde la categoría es 'Verdura'
verdura = df[df["Category"] == "Verdura"]
# Número de filas
print("Registos que sean 'Verdura':", len(verdura))
# Primeras filas de la categoría
print("Primeras filas de 'Verdura'")
print(verdura.head())
print("Guillermo García Hernández  2º DAW")
```

[102] ✓ 0.0s Python

```
Registos que sean 'Verdura': 142
Primeras filas de 'Verdura'
      id Category               name            subtitle  price  \
584  585  Verdura             Patata  Pieza 220 g aprox.   0.44
585  586  Verdura            Patatas          Malla 3 kg   5.10
586  587  Verdura      Patatas rojas          Malla 2 kg   3.80
587  588  Verdura            Patatas          Malla 5 kg   6.55
588  589  Verdura  Patatas guarnición          Malla 1 kg   2.55

     discount_price                                   main_image_url  \
584             NaN  https://prod-mercadona.imgix.net/images/a8c90b...
585            5.10  https://prod-mercadona.imgix.net/images/9a13af...
586             NaN  https://prod-mercadona.imgix.net/images/12962d...
587            6.55  https://prod-mercadona.imgix.net/images/709deb...
588             NaN  https://prod-mercadona.imgix.net/images/df2dd2...

                                secondary_image_url  \
584                                             NaN
585  https://prod-mercadona.imgix.net/images/c85764...
586  https://prod-mercadona.imgix.net/images/d99a17...
587  https://prod-mercadona.imgix.net/images/5f8faf...
588  https://prod-mercadona.imgix.net/images/f1aaaf...

                                  nutritional_info
584  Patata, Pieza, 220 Gramos aprox., 0,44€ por Un...
...
586    Patatas rojas, Malla, 2 Kilos, 3,80€ por Unidad
587  Patatas, Malla, 5 Kilos, Precio anterior: 6,55...
588  Patatas guarnición, Malla, 1 Kilo, 2,55€ por U...
Guillermo García Hernández  2º DAW
```

Aquellos productos que en su *subtitle* tenga la medida en kg.

```
# Filas donde el subtitle contiene 'kg'
con_kg = df[df["subtitle"].str.contains(r"\bkg\b", case=False, na=False)]
# Número de filas
print("Productos con 'kg' en el subtitle:", len(con_kg))
# Primeras filas con 'kg' en el subtitle
print("Primeras filas con 'kg' en el subtitle")
print(con_kg[["id", "Category", "name", "subtitle"]].head())
print("Guillermo García Hernández  2º DAW")
```

[103] ✓ 0.0s Python

```
Productos con 'kg' en el subtitle: 232
Primeras filas con 'kg' en el subtitle
      id          Category                                    name  \
31    32  Conejo y cordero                          Conejo entero
118  119            Cerdo              Pieza cabeza lomo de cerdo
123  124          Marisco                 Gambón grande congelado
128  129          Marisco  Gamba blanca pequeña Hacendado congelada
144  145          Marisco                                Mejillón

                subtitle
31   Pieza 1,21 kg aprox.
118  Pieza 1,15 kg aprox.
123            Caja 2 kg
128     Caja 1 kg aprox.
144           Malla 1 kg
Guillermo García Hernández  2º DAW
```

Aquellos productos que valgan entre 1 euro y 3.5 euros.

```
precio_rango = df[(df["price"] >= 1) & (df["price"] <= 3.5)]
print("Guillermo García Hernández  2º DAW")
print("Productos con precio entre 1 € y 3.5 €:")
print(precio_rango)
```

[109] ✓ 0.0s Python

```
Guillermo García Hernández  2º DAW
Productos con precio entre 1 € y 3.5 €:
        id             Category  \
11      12  Salazones y ahumados
12      13  Salazones y ahumados
14      15  Salazones y ahumados
15      16  Salazones y ahumados
17      18  Salazones y ahumados
...    ...                  ...
4714  4715            Rebozados
4715  4716            Rebozados
4718  4719            Rebozados
4720  4721            Rebozados
4721  4722            Rebozados

                                                   name  \
11     Filetes de anchoa en aceite de girasol Hacendado
12    Boquerones en vinagre Hacendado en aceite de g...
14       Filetes de anchoa en aceite de oliva Hacendado
15       Filetes de anchoa en aceite de oliva Hacendado
17          Migas de bacalao al estilo inglés Hacendado
...                                                 ...
4714      Muslitos de surimi Hacendado ultracongelados
4715  Palitos de merluza a la romana Hacendado ultra...
4718  Langostino caballitos rebozados Hacendado ultr...
...
4720  Rollitos primavera Hacendado ultracongelados, ...
4721  Fingers de queso Hacendado ultracongelados, Pa...
```

¿Cuántos hay? Usa el método shape.

```
cantidad_shape = precio_rango.shape[0]
print("Número de productos entre 1 € y 3.5 € (shape):", cantidad_shape)
print("Guillermo García Hernández  2º DAW")
```

[110] ✓ 0.0s Python

```
Número de productos entre 1 € y 3.5 € (shape): 2872
Guillermo García Hernández  2º DAW
```

Obtén el mismo resultado de antes pero usando count()

```
cantidad_count = precio_rango["price"].count()
print("Número de productos entre 1 € y 3.5 € (count):", cantidad_count)
print("Guillermo García Hernández  2º DAW")
```

[111] ✓ 0.0s Python

```
Número de productos entre 1 € y 3.5 € (count): 2872
Guillermo García Hernández  2º DAW
```

Devuelve los que sean de la marca "Hacendado".

```python
hacendado = df[df["name"].str.contains("Hacendado", case=False, na=False)]
print("Productos de la marca Hacendado")
print("Total:", hacendado.shape[0])
print("Guillermo García Hernández  2º DAW")
print(hacendado)
```

[112] ✓ 0.0s Python

```
Productos de la marca Hacendado
Total: 1921
Guillermo García Hernández  2º DAW
        id              Category  \
11      12  Salazones y ahumados
12      13  Salazones y ahumados
13      14  Salazones y ahumados
14      15  Salazones y ahumados
15      16  Salazones y ahumados
...    ...                   ...
4718  4719             Rebozados
4719  4720             Rebozados
4720  4721             Rebozados
4721  4722             Rebozados
4722  4723             Rebozados

                                                  name  \
11     Filetes de anchoa en aceite de girasol Hacendado
12    Boquerones en vinagre Hacendado en aceite de g...
13    Boquerones al vinagre Hacendado en aceite de g...
14       Filetes de anchoa en aceite de oliva Hacendado
15       Filetes de anchoa en aceite de oliva Hacendado
...                                                 ...
4718  Langostino caballitos rebozados Hacendado ultr...
4719  Filetes de boquerón en tempura Hacendado ultra...
...
4721  Fingers de queso Hacendado ultracongelados, Pa...
4722  Tequeños de queso Hacendado ultracongelados, C...
```

¿En qué índice se encuentra el producto con mayor precio?

```
indice_max_precio = df["price"].idxmax()
print("Índice del producto con mayor precio:", indice_max_precio)
print("Guillermo García Hernández  2º DAW")
```

[113] ✓ 0.0s Python

```
Índice del producto con mayor precio: 520
Guillermo García Hernández  2º DAW
```

Obtén la información de ese producto. No sirve poner el valor obtenido antes como una constante.

```
producto_max_precio = df.loc[indice_max_precio]
print("Producto con mayor precio (fila completa):")
# el id es diferente al índice, id = indice + 1
print(producto_max_precio)
print("Guillermo García Hernández  2º DAW")
```

[114] ✓ 0.0s Python

```
Producto con mayor precio (fila completa):
id                                                        521
Category                                        Jamón serrano
name                          Jamón bellota ibérico 100% Covap
subtitle                                   Pieza 9 kg aprox.
price                                                   504.0
discount_price                                            NaN
main_image_url          https://prod-mercadona.imgix.net/images/186414...
secondary_image_url     https://prod-mercadona.imgix.net/images/1b018f...
nutritional_info        Jamón bellota ibérico 100% Covap, Pieza, 9 Kil...
Name: 520, dtype: object
Guillermo García Hernández  2º DAW
```

Devuelve todos aquellos que tengan descuento. Puedes usar el método isnull() para realizar el filtrado.

```
    donde discount_price no es NaN
cuento = df[df["discount_price"].notna()]
Productos con descuento:", con_descuento.shape[0])
on_descuento[["id", "Category", "name", "price", "discount_price"]].head())
Guillermo García Hernández  2º DAW")
```

[115] ✓ 0.0s Python

```
Productos con descuento: 131
      id           Category  \
147  148            Marisco
181  182  Pescado congelado
187  188  Pescado congelado
585  586            Verdura
587  588            Verdura

                                                 name  price  discount_price
147                                   Almeja japonesa  10.80           10.80
181  Ventrescas de merluza del Cabo Vento ultracong...   3.20            3.20
187     Empanadillas de atún Hacendado ultracongeladas   2.00            2.00
585                                            Patatas   5.10            5.10
587                                            Patatas   6.55            6.55
Guillermo García Hernández  2º DAW
```

¿Si lanzas las funciones size(), count() y value_counts() sobre esa la columna de discount_price, cuál es la diferencia de la salida?

Respuesta:

```
[118]
print("Guillermo García Hernández  2º DAW")
print("size():  número total de elementos (incluye NaN)  ->", df["discount_pric
print("count(): número de NO nulos (excluye NaN)        ->", df["discount_price
print("value_counts() (primeros valores más frecuentes, NaN excluido por defect
print(df["discount_price"].value_counts().head())
```

✓ 0.0s Python

```
Guillermo García Hernández  2º DAW
size():  número total de elementos (incluye NaN)  -> 4723
count(): número de NO nulos (excluye NaN)        -> 131
value_counts() (primeros valores más frecuentes, NaN excluido por defecto):
discount_price
2.95    7
2.60    5
3.90    4
0.80    4
3.85    4
Name: count, dtype: int64
```

¿Cuántos productos hay por categoría? Ordena de menos a más.

```
oductos_por_cat = df["Category"].value_counts().sort_values(ascending=True)
int("Productos por categoría (de menos a más):")
int(productos_por_cat)
int("Guillermo García Hernández  2º DAW")
```

[172] ✓ 0.0s Python

```
Productos por categoría (de menos a más):
Category
Hielo                          3
Pinceles y brochas             5
Vino rosado                    5
Vino lambrusco y espumoso      5
Limpiacristales                7
                             ...
Insecticida y ambientador     73
Perfume y colonia             95
Coloración cabello           100
Leche y bebidas vegetales    121
Verdura                      142
Name: count, Length: 148, dtype: int64
Guillermo García Hernández  2º DAW
```

¿Cuántas categorías distintas hay? Usa el método nunique()

```
num_categorias = df["Category"].nunique()
print("Categorías distintas:", num_categorias)
print("Guillermo García Hernández  2º DAW")
```

[173] ✓ 0.0s Python

```
Categorías distintas: 148
Guillermo García Hernández  2º DAW
```

¿Cuál es el número de descuentos que hay por categoría?

```python
descuentos_por_cat = (
    df.assign(tiene_descuento=df["discount_price"].notna())
      .groupby("Category")["tiene_descuento"]
      .sum()
      .sort_values(ascending=False)
)
print("Número de productos con descuento por categoría:")
print(descuentos_por_cat)
print("Guillermo García Hernández  2º DAW")
```

[174] ✓ 0.0s Python

```
Número de productos con descuento por categoría:
Category
Cerveza                            14
Refresco de naranja y de limón     10
Licores                             8
Leche y bebidas vegetales           7
Desodorante                         7
                                   ..
Vino lambrusco y espumoso           0
Yogures desnatados                  0
Yogures griegos                     0
Yogures naturales y sabores         0
Yogures y postres infantiles        0
Name: tiene_descuento, Length: 148, dtype: int64
Guillermo García Hernández  2º DAW
```

¿Cuáles son las 5 categorías con mayor precio medio?

```python
top5_media_precio = (
    df.groupby("Category")["price"]
      .mean()
      .sort_values(ascending=False)
      .head(5)
)
print("Top 5 categorías por precio medio:")
print(top5_media_precicio := top5_media_precio)
print("Guillermo García Hernández  2º DAW")
```

[175] ✓ 0.0s Python

```
Top 5 categorías por precio medio:
Category
Jamón serrano          49.800357
Tartas y pasteles      10.453333
Licores                 9.439091
Pescado fresco          9.413810
Toallitas y pañales     9.296296
Name: price, dtype: float64
Guillermo García Hernández  2º DAW
```

Devuelve el producto con menor precio por cada categoría

```
egory")["price"].idxmin()
x_min_por_cat, ["Category", "name", "subtitle", "price", "discount_price"]]
tegoría:")
values("Category"))
  2º DAW")
```

[176] ✓ 0.0s Python

```
Producto más barato por categoría:
                            Category  \
1771           Aceite, vinagre y sal
2322          Aceitunas y encurtidos
3653     Acondicionador y mascarilla
3226  Afeitado y cuidado para hombre
2829                            Agua
...                              ...
1471              Yogures desnatados
1574                 Yogures griegos
1684               Yogures líquidos
1554     Yogures naturales y sabores
1571    Yogures y postres infantiles

                                                 name  \
1771                                Sal fina Hacendado
2322     Aceitunas verdes rellenas de anchoa Hacendado
3653  Acondicionador Repair & Nutrition Deliplus cab...
3226  Maquinillas de afeitar desechables Deliplus Fi...
2829                       Agua mineral pequeña Cortes
...                                                ...
1471  Yogur sabor fresa Hacendado 0% m.g 0% sin azúc...
1574                    Yogur griego natural Hacendado
1684              Bebida Kéfir natural Hacendado 0% m.g
1554                       Yogur sabor fresa Hacendado
...
1571                1 ud. (100 g)   0.95            NaN

[148 rows x 5 columns]
Guillermo García Hernández  2º DAW
```

Agrupa por categoría y por el descuento (sólo los que tengan), y casa la media de precio.

```python
con_descuento = df[df["discount_price"].notna()]
media_precio_con_descuento = (
    con_descuento
      .groupby("Category")["price"]
      .mean()
      .sort_values(ascending=False)
)
print("Media de precio (solo productos con descuento) por categoría:")
print(media_precio_con_descuento)
print("Guillermo García Hernández  2º DAW")
```

177] ✓ 0.0s Python

```
Media de precio (solo productos con descuento) por categoría:
Category
Toallitas y pañales                20.950000
Licores                            14.806250
Aceite, vinagre y sal              14.000000
Marisco                            10.800000
Perfume y colonia                  10.750000
Labios                             10.000000
Sidra y cava                        8.200000
Afeitado y cuidado para hombre      8.000000
Bases de maquillaje y corrector     7.000000
Colorete y polvos                   7.000000
Protector solar y aftersun          6.425000
Carne                               5.950000
Yogures líquidos                    5.790000
Cerveza                             5.758571
Leche y bebidas vegetales           5.705714
Tónica y bitter                     4.938000
Cuidado corporal                    4.583333
Verdura                             4.190000
Arroz y pasta                       4.150000
Higiene bucal                       4.033333
Refresco de cola                    3.900000
Limpieza vajilla                    3.750000
Refresco de naranja y de limón      3.491000
...
Vino blanco                         1.600000
Harina y preparado repostería       1.200000
Name: price, dtype: float64
```

Agrupa por subtitulo y cuenta cuántos productos tienen el mismo subtítulo.

```python
conteo_subtitle = df["subtitle"].value_counts(dropna=False)
print("Productos por subtitle (incluye NaN):")
print(conteo_subtitle.head())
print("Total de subtítulos distintos (incluyendo NaN):", conteo_subtitle.shape[
print("Guillermo García Hernández  2º DAW")
```

[178] ✓ 0.0s Python

```
Productos por subtitle (incluye NaN):
subtitle
Caja 1 ud.        132
1 ud.             124
Botella 750 ml     99
Brick 1 L          83
Paquete 500 g      77
Name: count, dtype: int64
Total de subtítulos distintos (incluyendo NaN): 1367
Guillermo García Hernández  2º DAW
```

¿Cuántos productos, por categoría tienen descuentos y cuantos no? Crea una nueva columna que se llame "tiene_descuento", de tipo booleano.

Tras ello, realiza el agrupamiento.

```python
df_td = df.assign(tiene_descuento=df["discount_price"].notna())
descuentos_por_cat = (
    df_td.groupby(["Category", "tiene_descuento"])
         .size()
         .unstack(fill_value=0)
         .rename(columns={False: "sin_descuento", True: "con_descuento"})
         .sort_values(by="con_descuento", ascending=False)
)
print("Conteo por categoría (con/sin descuento):")
print(descuentos_por_cat)
print("Guillermo García Hernández  2º DAW")
```

[179] ✓ 0.0s Python

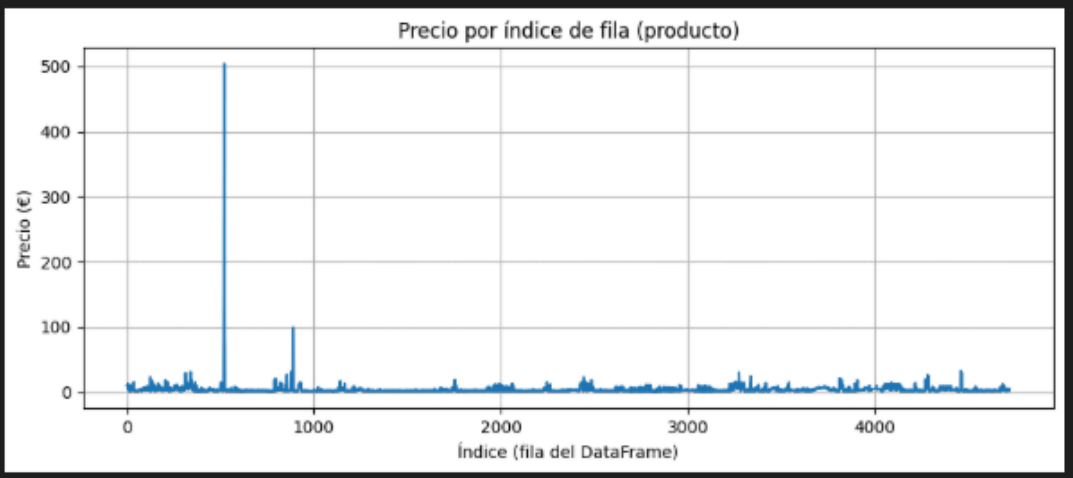
```
Conteo por categoría (con/sin descuento):
tiene_descuento                sin_descuento  con_descuento
Category
Cerveza                                   56             14
Refresco de naranja y de limón            23             10
Licores                                   47              8
Leche y bebidas vegetales                114              7
Desodorante                               34              7
...                                      ...            ...
Vino lambrusco y espumoso                  5              0
Yogures desnatados                        14              0
Yogures griegos                           10              0
Yogures naturales y sabores               19              0
Yogures y postres infantiles               8              0

[148 rows x 2 columns]
Guillermo García Hernández  2º DAW
```

Saca una diagrama de líneas del precio. ¿Qué información se presenta en cada eje?

```python
import matplotlib.pyplot as plt
plt.figure(figsize=(9,4))
df["price"].plot(kind="line")
plt.title("Precio por índice de fila (producto)")
plt.xlabel("Índice (fila del DataFrame)")
plt.ylabel("Precio (€)")
plt.grid(True)
plt.tight_layout()
plt.show()
print("Guillermo García Hernández  2º DAW")
```

[180] ✓ 0.1s Python

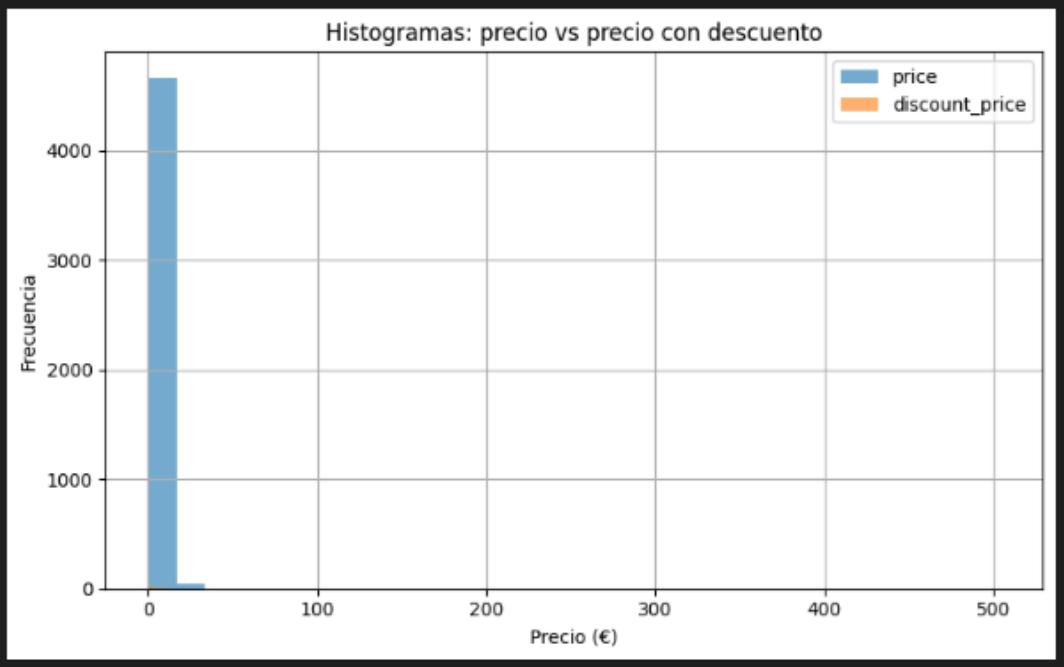


Guillermo García Hernández 2º DAW

Saca un histograma del precio, y del descuento, en el mismo gráfico.

```
plt.figure(figsize=(8,5))
plt.hist(df["price"].dropna(), bins=30, alpha=0.6, label="price")
plt.hist(df["discount_price"].dropna(), bins=30, alpha=0.6, label="discount_pri
plt.title("Histogramas: precio vs precio con descuento")
plt.xlabel("Precio (€)")
plt.ylabel("Frecuencia")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
print("Guillermo García Hernández  2º DAW")
```

[181] ✓ 0.1s Python

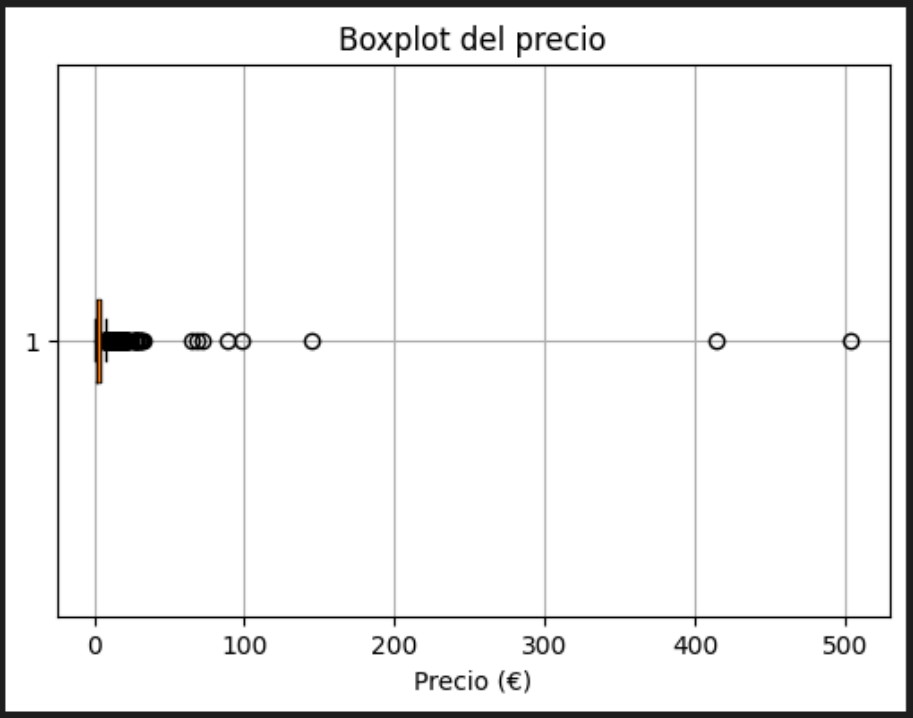


Guillermo García Hernández 2º DAW

Saca los box plot de estas dos variables, en dos celdas diferentes. ¿Tienen valores atípicos?

```
plt.figure(figsize=(6,4))
plt.boxplot(df["price"].dropna(), vert=False)
plt.title("Boxplot del precio")
plt.xlabel("Precio (€)")
plt.grid(True)
plt.show()
print("Guillermo García Hernández  2º DAW")
```

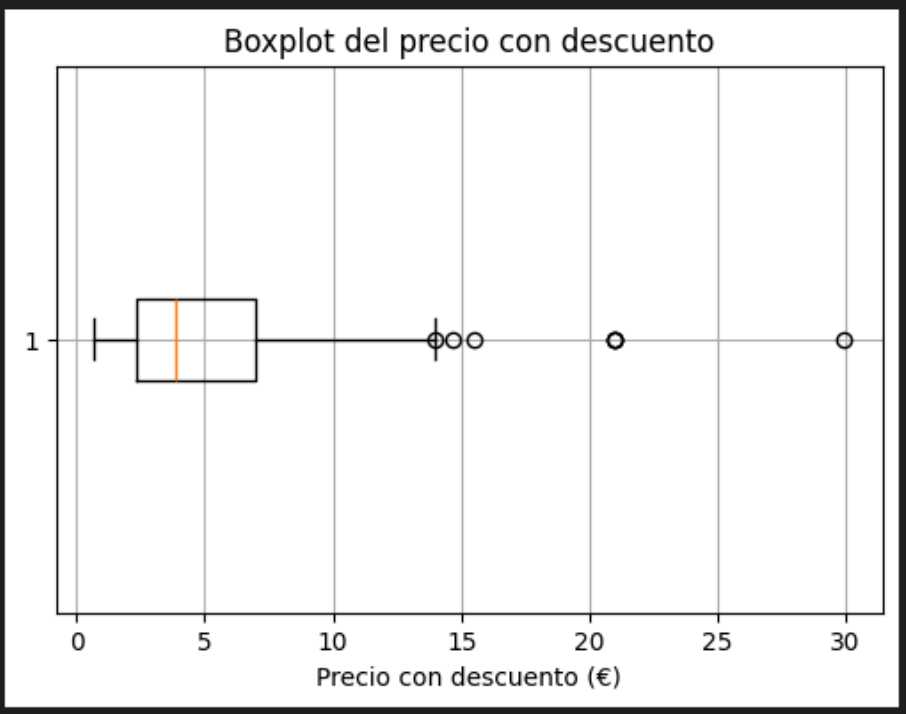
[182] ✓ 0.0s Python



Guillermo García Hernández 2º DAW

```python
plt.figure(figsize=(6,4))
plt.boxplot(df["discount_price"].dropna(), vert=False)
plt.title("Boxplot del precio con descuento")
plt.xlabel("Precio con descuento (€)")
plt.grid(True)
plt.show()
print("Guillermo García Hernández  2º DAW")
```

[183] ✓ 0.0s Python



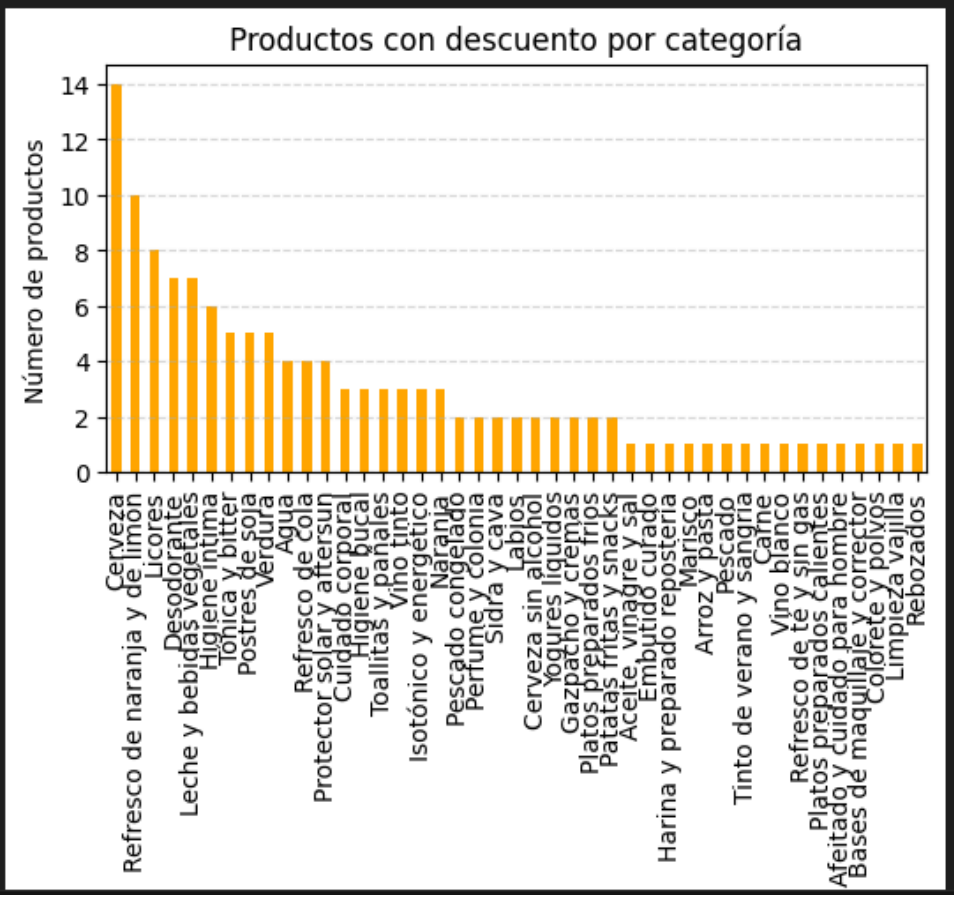
Guillermo García Hernández 2º DAW

```python
import matplotlib.pyplot as plt
# Filtrar los productos con descuento (no vacíos)
df_discount = df[df["discount_price"].notna()]

# Contar productos por categoría
count_by_category = df_discount["Category"].value_counts()

# Diagrama de barras
count_by_category.plot(kind="bar", color="orange", figsize=(6,3))
plt.title("Productos con descuento por categoría")
plt.xlabel("Categoría")
plt.ylabel("Número de productos")
plt.grid(axis="y", linestyle="--", alpha=0.5)
plt.show()
print("Guillermo García Hernández  2º DAW")
```

[184] ✓ 0.2s Python



Extra: Obtén, de forma programática, de la categoría con más descuentos, todos los productos SIN descuento.

```
print("Guillermo García Hernández  2º DAW")
# Categoría con más descuentos
top_category = df_discount["Category"].value_counts().idxmax()
print("Categoría con más descuentos:", top_category)
# Filtrar productos de esa categoría SIN descuento
sin_descuento = df[(df["Category"] == top_category) & (df["discount_price"].isr
sin_descuento[["Category", "name", "price", "discount_price"]].head()
```

[185] ✓ 0.0s Python

```
Guillermo García Hernández  2º DAW
Categoría con más descuentos: Cerveza
```

	Category	name	price	discount_price
3063	Cerveza	Cerveza Clásica Steinburg	3.72	NaN
3064	Cerveza	Cerveza Clásica Steinburg	0.31	NaN
3065	Cerveza	Cerveza Heineken	5.52	NaN
3066	Cerveza	Cerveza Heineken	0.86	NaN
3067	Cerveza	Cerveza Amstel	6.00	NaN

Usa el índice de correlación de Pearson de las variables del dataset

```python
# Calcular la correlación entre precio y descuento
corr = df[["price", "discount_price"]].corr(method="pearson")

print("Índice de correlación de Pearson:")
print(corr)

print("Guillermo García Hernández  2º DAW")
```

[186] ✓ 0.0s Python

```
Índice de correlación de Pearson:
                price  discount_price
price             1.0             1.0
discount_price    1.0             1.0
Guillermo García Hernández  2º DAW
```

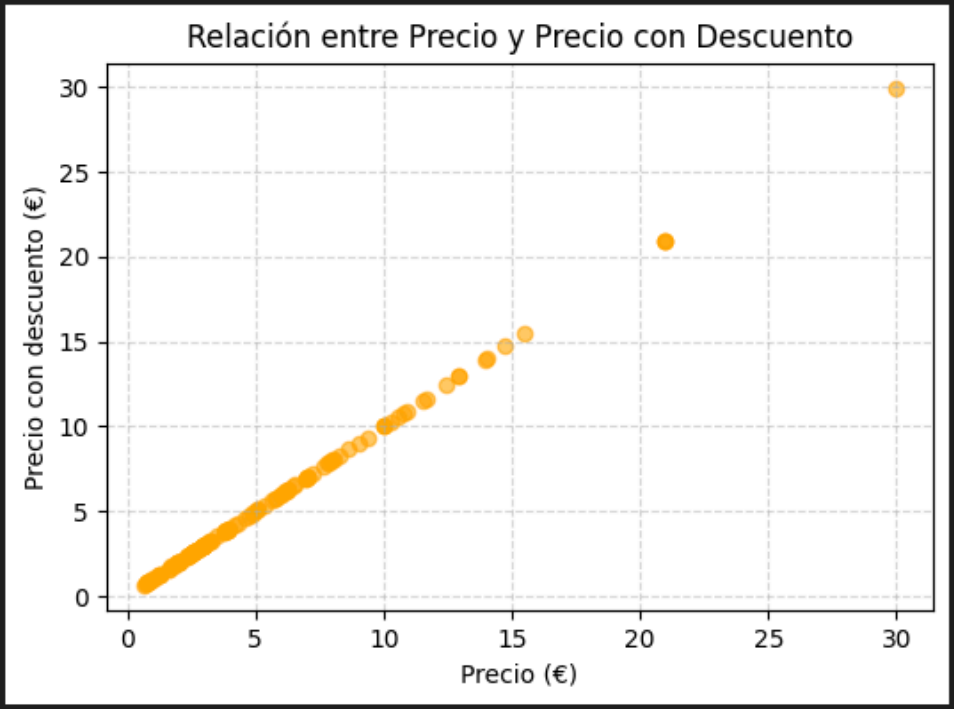
Saca un scatter plot. ¿Hay correlación entre las variables numéricas? ¿A qué se debe?

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(6,4))
plt.scatter(df["price"], df["discount_price"], alpha=0.6, color="orange")
plt.title("Relación entre Precio y Precio con Descuento")
plt.xlabel("Precio (€)")
plt.ylabel("Precio con descuento (€)")
plt.grid(True, linestyle="--", alpha=0.5)
plt.show()

print("Guillermo García Hernández  2º DAW")
```

[187] ✓ 0.0s Python



Guillermo García Hernández 2º DAW