Untitled1

January 18, 2026

0.0.1 Resumen número para el dataset limpio de Ever Peak

Tras haber validado la limpieza de los datos ahora necesitamos entender como se distribuyen los datos para tomar decisiones informadas. El problema que solucionaremos en este paso dos es mitigar los sesgos y outliers que puedan comprometer la calidad de los reportes y la toma de decisiones correcta.

El equipo de Inteligencia Comercial necesita una visión rápida del desempeño de las categorías Fashion y Sports para entender su situación actual y detectar posibles anomalías en los datos.

Se mantiene este código para visualizar el dataset

[1]:
```
import pandas as pd

df = pd.read_csv('/datasets/everpeak_clean.csv')
print("Columnas disponibles:")
print(df.columns.tolist())
print("\nPrimeras filas del dataset:")
print(df.head())
```

```
Columnas disponibles:
['order_id', 'order_date', 'customer_id', 'product_category', 'price',
'quantity', 'order_value', 'payment_method', 'city', 'state', 'customer_age',
'quantity_invalid_flag', 'age_invalid_flag', 'state_missing_flag']

Primeras filas del dataset:
   order_id  order_date  customer_id product_category  price  quantity  \
0         1  2024-02-02         2616           Sports    269        50
1         2  2024-10-10         1736          Grocery     66        10
2         3  2024-08-27         2543           Sports    267        19
3         4  2024-06-09         2252             Toys    114       125
4         5  2024-06-07         1583          Fashion    729        16

   order_value payment_method         city state  customer_age  \
0        13385    credit_card     New York    NY            66
1          660     debit_card  Los Angeles    CA            24
2         5073    credit_card      Chicago    IL            23
3        14290    credit_card     New York    NY            70
4        11754    credit_card      Houston    TX            75
```

	quantity_invalid_flag	age_invalid_flag	state_missing_flag
0	0	0	0
1	1	0	0
2	1	0	0
3	0	0	0
4	0	0	0

[ ]:
```
Se realiza un resumen estadistico de Sports y Fashion
```

[2]:
```
import pandas as pd

df = pd.read_csv('/datasets/everpeak_clean.csv')

# Crear dataframes para cada categoría
df_fashion = df[df['product_category'] == 'Fashion']
df_sports = df[df['product_category'] == 'Sports']

# --- Resumen de columnas numéricas con describe()
columnas_numericas = ['order_value', 'customer_age', 'price', 'quantity']

print('Resumen estadístico de la categoría Fashion')
print(df_fashion[columnas_numericas].describe())

print()
print('Resumen estadístico de la categoría Sports')
print(df_sports[columnas_numericas].describe())
```

```
Resumen estadístico de la categoría Fashion
         order_value  customer_age         price     quantity
count     739.000000    739.000000    739.000000   739.000000
mean     8745.139378     48.066306    585.863329    24.124493
std     10318.954880     18.083657    655.641761    72.380942
min       122.000000     18.000000     25.000000     1.000000
25%      2913.000000     32.000000    274.500000     8.000000
50%      8115.000000     47.000000    457.000000    14.000000
75%     12527.000000     64.000000    680.000000    21.000000
max    224884.000000     80.000000  11836.000000  1835.000000

Resumen estadístico de la categoría Sports
        order_value  customer_age         price    quantity
count    702.000000    702.000000   702.000000  702.000000
mean   10365.264957     49.742165   839.692308   21.099715
std     8820.536231     17.437736   848.292770   47.126458
min      195.000000     18.000000    47.000000    1.000000
25%     4308.750000     36.000000   347.000000    7.000000
50%    10569.500000     49.000000   553.500000   13.000000
75%    13461.750000     64.750000  1016.250000   20.000000
max    78964.000000     80.000000  9541.000000  833.000000
```

Se observa que las metricas estadisticas presentan fluctuaciones y valores atípicos que no permiten reflejar el impacto real del comportamiento histórico. En el siguiente código validaremos que la mediana es la mejor metrica para el analisis de este dataset

[4]:
```
import pandas as pd
df = pd.read_csv('/datasets/everpeak_clean.csv')

# Filtrar la categoría Grocery
df_grocery = df[df['product_category'] == 'Grocery']

# Calcular media y mediana del gasto
promedio = df_grocery['order_value'].mean()
mediana = df_grocery['order_value'].median()

# Mostrar resultados
print("Promedio del gasto en Grocery:", promedio)
print("Mediana del gasto en Grocery:", mediana)

# Interpretación según comparación de media y mediana
print("El promedio está afectado por outliers o valores atípicos en Grocery")
```

```
Promedio del gasto en Grocery: 6943.137426900585
Mediana del gasto en Grocery: 4131.0
El promedio está afectado por outliers o valores atípicos en Grocery
```

[5]:
```
import pandas as pd
df = pd.read_csv('/datasets/everpeak_clean.csv')


# Calcular media y mediana del gasto
promedio = df['quantity'].mean()
mediana = df['quantity'].median()
# Promedio y mediana de quantity
print("Promedio de quantity: ", promedio)
print("Mediana de quantity: ", mediana)
print("El promedio está afectado por los outliers o valores atípicos.")
```

```
Promedio de quantity:  32.3598
Mediana de quantity:  14.0
El promedio está afectado por los outliers o valores atípicos.
```

[6]:
```
import pandas as pd

df = pd.read_csv('/datasets/everpeak_clean.csv')

# Crear dataframes para cada categoría
df_ny = df[df['city'] == 'New York']
df_la = df[df['city'] == 'Los Angeles']
```

```
# --- Resumen de columnas numéricas con describe()
columnas_numericas = ['order_value', 'customer_age', 'price', 'quantity']

print('Resumen estadístico de la ciudad New York')
print(df_ny[columnas_numericas].describe())

print()
print('Resumen estadístico de la ciudad Los Angeles')
print(df_la[columnas_numericas].describe())
```

```
Resumen estadístico de la ciudad New York
         order_value  customer_age         price     quantity
count     497.000000    497.000000    497.000000   497.000000
mean    10545.410463     49.847082    741.454728    40.800805
std     13914.329615     17.507624   1105.570500   147.484959
min        58.000000     18.000000     12.000000     1.000000
25%      2862.000000     35.000000    203.000000     8.000000
50%     10682.000000     49.000000    457.000000    14.000000
75%     13327.000000     64.000000    787.000000    24.000000
max    224884.000000     80.000000  11836.000000  2083.000000

Resumen estadístico de la ciudad Los Angeles
         order_value  customer_age         price     quantity
count     510.000000    510.000000    510.000000   510.000000
mean    10449.945098     49.741176    751.674510    39.511765
std     15979.195581     17.670518   1175.722235   128.503594
min        44.000000     18.000000     13.000000     1.000000
25%      3160.500000     35.000000    202.500000     7.000000
50%     10442.500000     49.000000    457.000000    14.000000
75%     13322.000000     65.000000    811.250000    23.750000
max    303824.000000     80.000000  17872.000000  1987.000000
```

Con este código concluimos la exploracion de los datos númericos, concluimos que la mediana es la mejor metrica estadística para hacer el analisis del dataset

Ahora procedo a hacer un resumen categorico y la distribucion de los datos categoricos para revisar outliers que afecten a los datos

[7]:
```
import pandas as pd

df = pd.read_csv('/datasets/everpeak_clean.csv')

#  Columnas categóricas
columnas_categoricas = ['payment_method', 'product_category']

# Filtra por ciudad
df_ny = df[df['city'] == 'New York']
```

```
df_chicago = df [df['city'] == 'Chicago']

# Resumen categórico New York
print("Resumen categórico - New York")
print(df_ny[columnas_categoricas].describe())

print()

# Resumen categórico Chicago
print("Resumen categórico - Chicago")
print(df_chicago[columnas_categoricas].describe())
```

```
Resumen categórico - New York
       payment_method product_category
count             497              497
unique              4                8
top       credit_card             Toys
freq              271               76

Resumen categórico - Chicago
       payment_method product_category
count             482              482
unique              4                8
top       credit_card             Toys
freq              284               74
```

[8]:
```
import pandas as pd
df = pd.read_csv('/datasets/everpeak_clean.csv')

columnas_categoricas = ['product_category', 'payment_method', 'city', 'state']

# --- Distribución completa de columnas categóricas usando for
for col in columnas_categoricas:
    print(col)
    print("Frecuencia absoluta")
    print(df[col].value_counts())
    print("Frecuencia relativa")
    print(df[col].value_counts(normalize=True))
    print()
```

```
product_category
Frecuencia absoluta
Fashion        739
Electronics    735
Beauty         721
Toys           715
Sports         702
Grocery        684
```

```
Home           679
?               25
Name: product_category, dtype: int64
Frecuencia relativa
Fashion        0.1478
Electronics    0.1470
Beauty         0.1442
Toys           0.1430
Sports         0.1404
Grocery        0.1368
Home           0.1358
?              0.0050
Name: product_category, dtype: float64

payment_method
Frecuencia absoluta
credit_card    2737
paypal         1175
debit_card      889
cash            199
Name: payment_method, dtype: int64
Frecuencia relativa
credit_card    0.5474
paypal         0.2350
debit_card     0.1778
cash           0.0398
Name: payment_method, dtype: float64

city
Frecuencia absoluta
Seattle          513
Houston          513
Los Angeles      510
New York         497
Miami            493
Phoenix          491
Chicago          482
Boston           474
San Francisco    467
Denver           460
unknown          100
Name: city, dtype: int64
Frecuencia relativa
Seattle          0.1026
Houston          0.1026
Los Angeles      0.1020
New York         0.0994
Miami            0.0986
```

```
Phoenix          0.0982
Chicago          0.0964
Boston           0.0948
San Francisco    0.0934
Denver           0.0920
unknown          0.0200
Name: city, dtype: float64

state
Frecuencia absoluta
CA          977
WA          513
TX          513
NY          497
FL          493
AZ          491
IL          482
MA          474
CO          460
unknown     100
Name: state, dtype: int64
Frecuencia relativa
CA          0.1954
WA          0.1026
TX          0.1026
NY          0.0994
FL          0.0986
AZ          0.0982
IL          0.0964
MA          0.0948
CO          0.0920
unknown     0.0200
Name: state, dtype: float64
```

Hacemos códigos para visualizar la distribucion de los métodos de pago y la ciudades para determinar patrones dominantes

[9]:
```python
import pandas as pd

df = pd.read_csv('/datasets/everpeak_clean.csv')

# Columnas categóricas
columnas_categoricas = ['payment_method', 'city']

# Filtra por categoría
df_toys = df[df['product_category'] == 'Toys']
```

```
# Resumen categórico Toys
print("Resumen categórico - Toys")
print(df['product_category'].describe())
```

```
Resumen categórico - Toys
count        5000
unique          8
top       Fashion
freq          739
Name: product_category, dtype: object
```

Distribucion completa de los value_counts de todas las ciudades

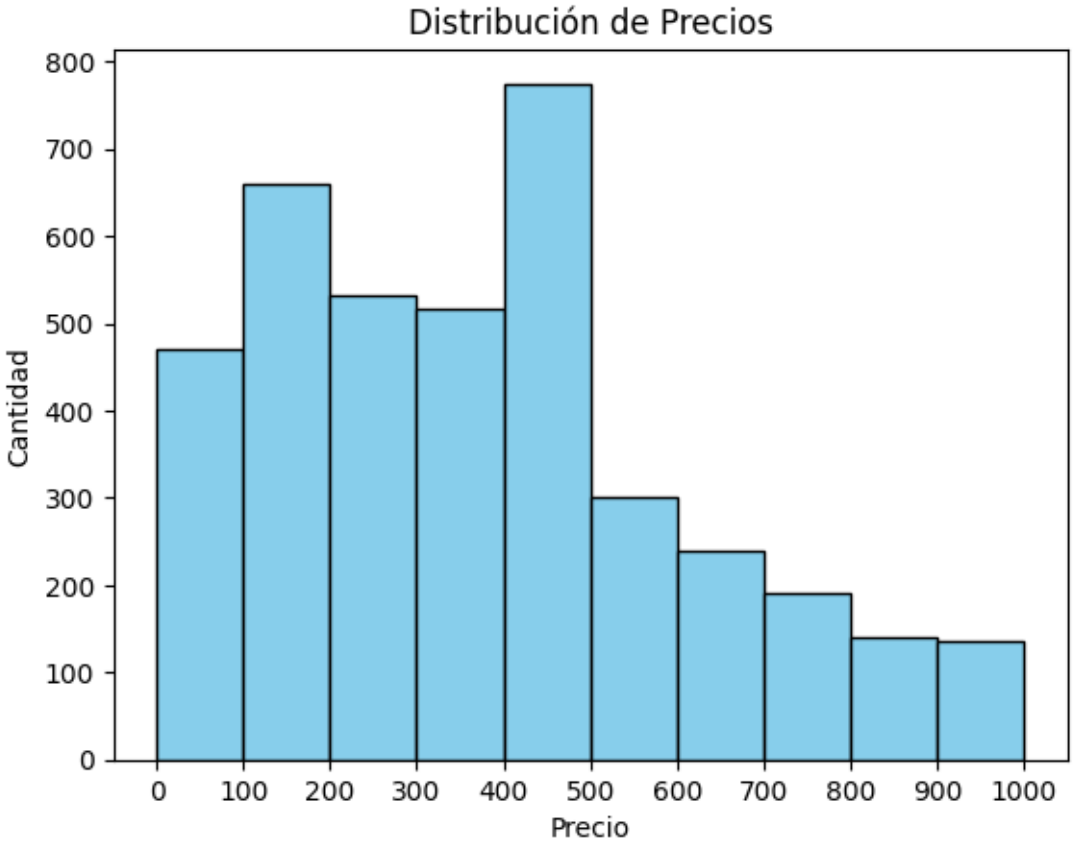
[10]:
```
import pandas as pd
df = pd.read_csv('/datasets/everpeak_clean.csv')

# Filtra por categoría
df_sports = df[df['product_category'] == 'Sports']

# Distribución de city
print("Frecuencia absoluta")
print(df['product_category'].describe())
print("\nFrecuencia relativa")
print(df['product_category'].describe())
```

```
Frecuencia absoluta
count        5000
unique          8
top       Fashion
freq          739
Name: product_category, dtype: object

Frecuencia relativa
count        5000
unique          8
top       Fashion
freq          739
Name: product_category, dtype: object
```

Ahora hacemos los histogramas de los datos

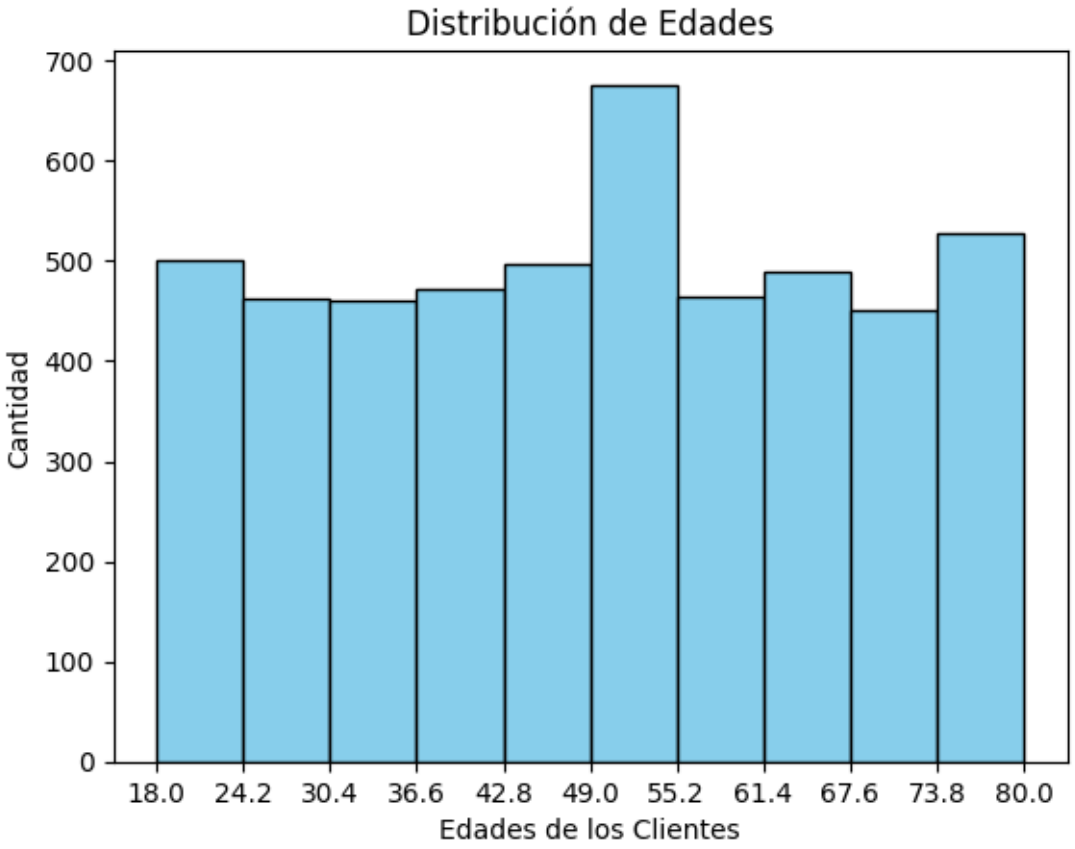
[1]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('/datasets/everpeak_clean.csv')

# Graficar histograma
counts, bin_edges, _ = plt.hist(df['price'], bins=10, range=(0, 1000),␣
 ↪color='skyblue', edgecolor='black')
```

```
# Mostrar las marcas de los bins en el eje X
plt.xticks(bin_edges)

# Etiquetas y título del gráfico
plt.xlabel('Precio')
plt.ylabel('Cantidad')
plt.title('Distribución de Precios')
plt.show()
```



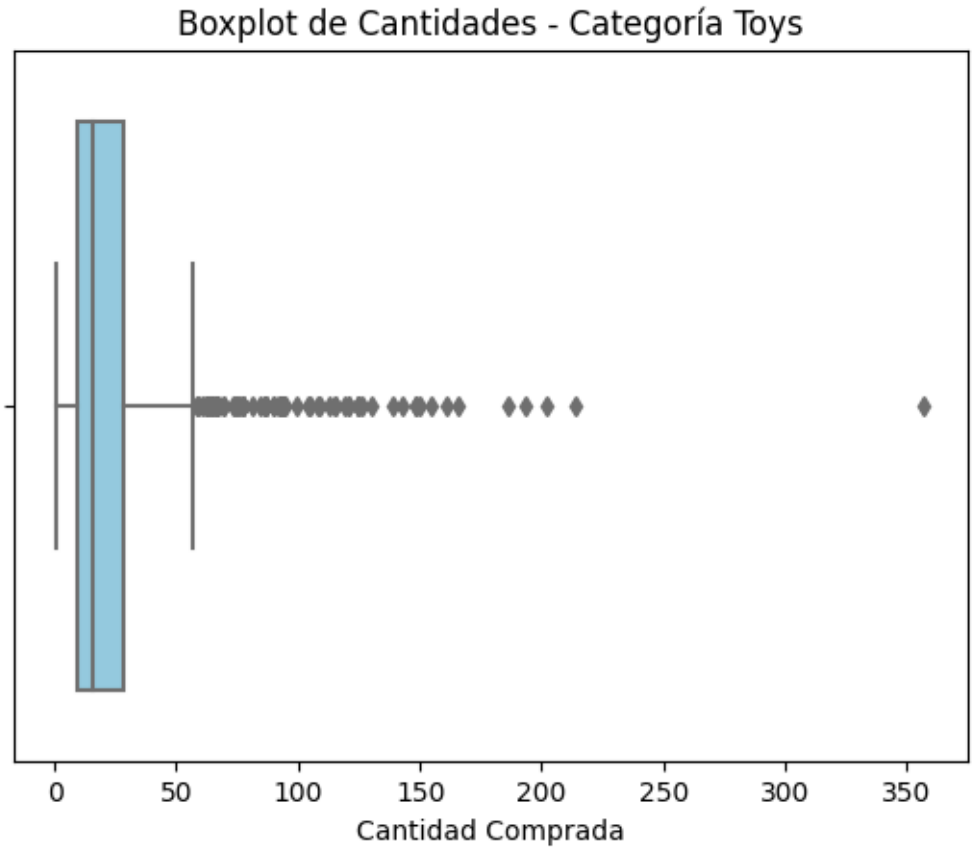
[2]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('/datasets/everpeak_clean.csv')

# Graficar histograma
counts, bin_edges, _ = plt.hist(df['customer_age'], bins=10, color='skyblue',␣
↪edgecolor='black')

# Mostrar las marcas de los bins en el eje X
```

```
plt.xticks(bin_edges)

# Etiquetas y título del gráfico
plt.xlabel('Edades de los Clientes')
plt.ylabel('Cantidad')
plt.title('Distribución de Edades')
plt.show()
```



[3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/datasets/everpeak_clean.csv')
df_toys = df[df['product_category'] == 'Toys']

# Graficar BoxPlot
sns.boxplot(x=df_toys['quantity'], color='skyblue')
# Etiquetas y título del gráfico
plt.xlabel('Cantidad Comprada')
plt.title('Boxplot de Cantidades - Categoría Toys')
```

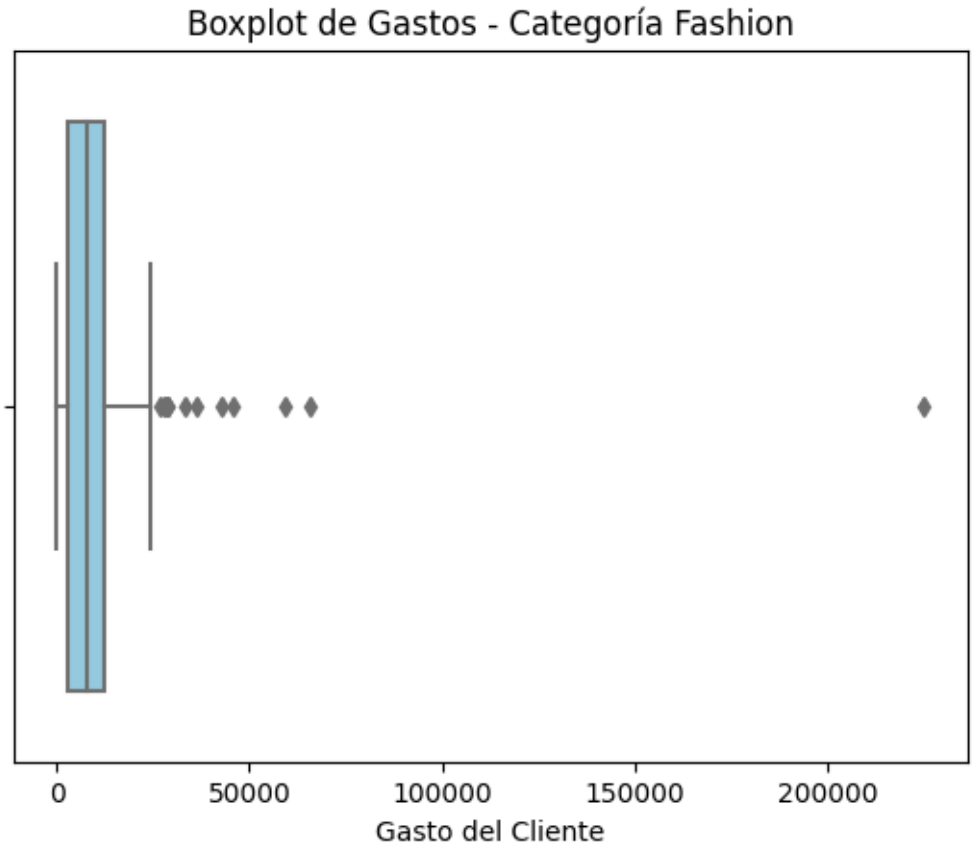
```
plt.show()
```

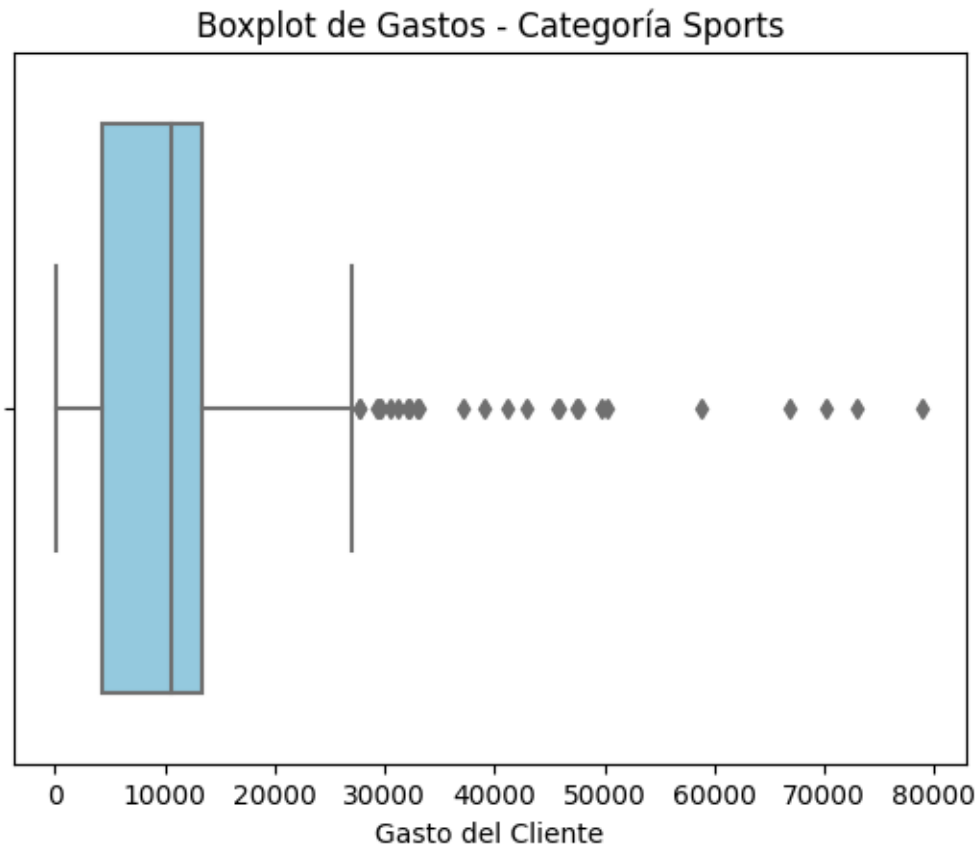


En el boxplot se pueden ver claramente los outliers representados como puntos individuales fuera de los “bigotes” (whiskers) de la caja.

[4]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/datasets/everpeak_clean.csv')

# 1. Filtrar datos

df_fashion = df[df['product_category'] == 'Fashion']
df_sports = df[df['product_category'] == 'Sports']

# 2. BoxPlot Categoría Fashion
sns.boxplot(x=df_fashion['order_value'], color='skyblue')
plt.xlabel('Gasto del Cliente')
```

```
plt.title('Boxplot de Gastos - Categoría Fashion')
plt.show()

# 3. BoxPlot Categoría Sports
sns.boxplot(x=df_sports['order_value'], color='skyblue')
plt.xlabel('Gasto del Cliente')
plt.title('Boxplot de Gastos - Categoría Sports')
plt.show()
```





Igualmente se observan outliers fuera del bloxpot

Como conclusión, los histogramas y las tablas nos muestran que los datos del dataset presentan outliers o valores atípicos y la mediana es el método más indicado para hacer inferencias estadísticas del negocio.