LOAD AND INSPECT DATA

In [1]:
```
import pandas as pd
df=pd.read_csv("E:/srushti/pandas/retail_sales.csv")
df
```

Out[1]:

	Order_ID	Customer_ID	Gender	Age	City	Product_Category	Quantity	Unit_Pr
0	O1000	C139	Female	35.0	Delhi	Groceries	4	14
1	O1001	C129	Female	20.0	Mumbai	Books	2	1
2	O1002	C115	Male	51.0	Kolkata	Books	5	2
3	O1003	C143	Female	33.0	Chennai	Home Decor	2	18
4	O1004	C108	Male	60.0	Mumbai	Electronics	4	9
...	...	...	...	...	...	...	...	
195	O1195	C142	Male	25.0	Bangalore	Electronics	4	1
196	O1196	C144	Male	40.0	Mumbai	Clothing	1	17
197	O1197	C124	Female	51.0	Kolkata	Clothing	2	5
198	O1198	C115	Male	51.0	Mumbai	Books	3	6
199	O1199	C132	Female	52.0	Mumbai	Clothing	4	4

200 rows × 10 columns

In [2]:
```
df.head(10)
```

Out[2]:

	Order_ID	Customer_ID	Gender	Age	City	Product_Category	Quantity	Unit_Price
0	O1000	C139	Female	35.0	Delhi	Groceries	4	1487
1	O1001	C129	Female	20.0	Mumbai	Books	2	1124
2	O1002	C115	Male	51.0	Kolkata	Books	5	254
3	O1003	C143	Female	33.0	Chennai	Home Decor	2	1866
4	O1004	C108	Male	60.0	Mumbai	Electronics	4	999
5	O1005	C121	Female	44.0	Kolkata	Electronics	1	977
6	O1006	C139	Male	46.0	Mumbai	Clothing	5	1681
7	O1007	C119	Female	45.0	Bangalore	Home Decor	1	361
8	O1008	C123	Female	22.0	Mumbai	Clothing	1	1603
9	O1009	C111	Female	51.0	Mumbai	Clothing	1	286

In [3]:
```
df.isna().sum()
```

Out[3]:
```
Order_ID          0
Customer_ID       0
Gender            0
Age               6
City              0
Product_Category  0
Quantity          0
Unit_Price        0
Order_Date        0
Payment_Mode      0
dtype: int64
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Order_ID          200 non-null    object 
 1   Customer_ID       200 non-null    object 
 2   Gender            200 non-null    object 
 3   Age               194 non-null    float64
 4   City              200 non-null    object 
 5   Product_Category  200 non-null    object 
 6   Quantity          200 non-null    int64  
 7   Unit_Price        200 non-null    int64  
 8   Order_Date        200 non-null    object 
 9   Payment_Mode      200 non-null    object 
dtypes: float64(1), int64(2), object(7)
memory usage: 15.8+ KB
```

DATA CLEANING

In [13]:
```
df["Age"].fillna(df["Age"].mean(),inplace=True)
```

```
C:\Users\srushti\AppData\Local\Temp\ipykernel_26588\2595122914.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained assig
nment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  df["Age"].fillna(df["Age"].mean(),inplace=True)
```

In [15]:
```
df.isna().sum()
```

Out[15]:
```
Order_ID            0
Customer_ID         0
Gender              0
Age                 0
City                0
Product_Category    0
Quantity            0
Unit_Price          0
Order_Date          0
Payment_Mode        0
dtype: int64
```

In [25]:
```
df.drop_duplicates(inplace=True)
print(df.duplicated().sum())
```

```
0
```

In [28]:
```
df["Order_Date"]=pd.to_datetime(df["Order_Date"])
df
```

Out[28]:

	Order_ID	Customer_ID	Gender	Age	City	Product_Category	Quantity	Unit_Pr
0	O1000	C139	Female	35.0	Delhi	Groceries	4	14
1	O1001	C129	Female	20.0	Mumbai	Books	2	1
2	O1002	C115	Male	51.0	Kolkata	Books	5	2
3	O1003	C143	Female	33.0	Chennai	Home Decor	2	18
4	O1004	C108	Male	60.0	Mumbai	Electronics	4	9
...	...	...	...	...	...	...	...	
195	O1195	C142	Male	25.0	Bangalore	Electronics	4	1
196	O1196	C144	Male	40.0	Mumbai	Clothing	1	17
197	O1197	C124	Female	51.0	Kolkata	Clothing	2	5
198	O1198	C115	Male	51.0	Mumbai	Books	3	6
199	O1199	C132	Female	52.0	Mumbai	Clothing	4	4

200 rows × 10 columns

FEATURE ENGINEERING

In [30]:
```
df["Total_amount"]=df["Quantity"]*df["Unit_Price"]
df
```

Out[30]:

	Order_ID	Customer_ID	Gender	Age	City	Product_Category	Quantity	Unit_Pr
0	O1000	C139	Female	35.0	Delhi	Groceries	4	1
1	O1001	C129	Female	20.0	Mumbai	Books	2	1
2	O1002	C115	Male	51.0	Kolkata	Books	5	
3	O1003	C143	Female	33.0	Chennai	Home Decor	2	1
4	O1004	C108	Male	60.0	Mumbai	Electronics	4	
...	...	...	...	...	...	...	...	...
195	O1195	C142	Male	25.0	Bangalore	Electronics	4	1
196	O1196	C144	Male	40.0	Mumbai	Clothing	1	1
197	O1197	C124	Female	51.0	Kolkata	Clothing	2	
198	O1198	C115	Male	51.0	Mumbai	Books	3	
199	O1199	C132	Female	52.0	Mumbai	Clothing	4	

200 rows × 11 columns

In [35]:
```
df["Month"]=df["Order_Date"].dt.month_name()
df
```

Out[35]:

	Order_ID	Customer_ID	Gender	Age	City	Product_Category	Quantity	Unit_Pr
0	O1000	C139	Female	35.0	Delhi	Groceries	4	1
1	O1001	C129	Female	20.0	Mumbai	Books	2	1
2	O1002	C115	Male	51.0	Kolkata	Books	5	
3	O1003	C143	Female	33.0	Chennai	Home Decor	2	1
4	O1004	C108	Male	60.0	Mumbai	Electronics	4	
...	...	...	...	...	...	...	...	...
195	O1195	C142	Male	25.0	Bangalore	Electronics	4	1
196	O1196	C144	Male	40.0	Mumbai	Clothing	1	1
197	O1197	C124	Female	51.0	Kolkata	Clothing	2	
198	O1198	C115	Male	51.0	Mumbai	Books	3	
199	O1199	C132	Female	52.0	Mumbai	Clothing	4	

200 rows × 12 columns

EXPLORATORY ANALYSIS

In [37]:
```
print("Total orders:",df["Order_ID"].count().sum())
```

```
Total orders: 200
```

In [39]:
```
print("Total_revenue:",df["Total_amount"].count().sum())
```

```
Total_revenue: 200
```

In [43]:
```
print("Average_order_value",df["Total_amount"].mean().sum())
```

```
Average_order_value 3257.0
```

In [63]:
```
city_highest_sales=df.groupby("City").agg({"Total_amount":"max"})
city_highest_sales
```

Out[63]:

City	Total_amount
Bangalore	8855
Chennai	9110
Delhi	9480
Kolkata	9925
Mumbai	8855

In [64]:
```
cat_revenue=df.groupby("Product_Category")["Total_amount"].sum()
cat_revenue
cat_revenue.sort_values(ascending=False)
```

Out[64]:
```
Product_Category
Clothing       146454
Groceries      140740
Books          132290
Electronics    131014
Home Decor     100902
Name: Total_amount, dtype: int64
```

In [67]:
```
avg_age_mean=df.loc[df["Payment_Mode"]=="UPI","Age"].mean()
print(avg_age_mean)
```

```
39.67131594906004
```

GROUPING AND AGGREGATIONS

In [68]:
```
df.groupby("City")["Total_amount"].sum()
```

Out[68]:
```
City
Bangalore    119156
Chennai      130684
Delhi        106591
Kolkata      170790
Mumbai       124179
Name: Total_amount, dtype: int64
```

In [79]:
```
df.groupby("Product_Category").agg({"Quantity":"mean"})
```

Out[79]:

Product_Category	Quantity
Books	3.305556
Clothing	2.869565
Electronics	3.179487
Groceries	2.744186
Home Decor	2.805556

In [80]:
```
avg_amt_bygender=df.groupby("Gender")["Total_amount"].mean()
avg_amt_bygender
```

Out[80]:
```
Gender
Female    3088.559633
Male      3458.758242
Name: Total_amount, dtype: float64
```

FILTERING

In [ ]:
```

```

In [90]:
```
print(df[df["Total_amount"] > 2000])
```

```
    Order_ID Customer_ID  Gender   Age       City Product_Category  Quantity  \
0      O1000        C139  Female  35.0      Delhi        Groceries         4
1      O1001        C129  Female  20.0     Mumbai            Books         2
3      O1003        C143  Female  33.0    Chennai       Home Decor         2
4      O1004        C108    Male  60.0     Mumbai      Electronics         4
6      O1006        C139    Male  46.0     Mumbai         Clothing         5
..       ...         ...     ...   ...        ...              ...       ...
192    O1192        C109  Female  52.0    Kolkata            Books         3
193    O1193        C137    Male  26.0      Delhi        Groceries         2
194    O1194        C133  Female  53.0      Delhi        Groceries         2
195    O1195        C142    Male  25.0  Bangalore      Electronics         4
198    O1198        C115    Male  51.0     Mumbai            Books         3

     Unit_Price Order_Date Payment_Mode  Total_amount     Month  Quantity_mean
0          1487 2024-12-25         Cash          5948  December           2.97
1          1124 2024-10-22         Cash          2248   October           2.97
3          1866 2024-02-21         Card          3732  February           2.97
4           999 2024-06-27         Card          3996      June           2.97
6          1681 2024-11-21         Card          8405  November           2.97
..          ...        ...          ...           ...       ...            ...
192        1962 2024-05-28       Wallet          5886       May           2.97
193        1467 2024-03-16       Wallet          2934     March           2.97
194        1714 2024-10-02       Wallet          3428   October           2.97
195        1157 2024-05-29          UPI          4628       May           2.97
198         681 2024-04-05         Card          2043     April           2.97

[120 rows x 13 columns]
```

In [95]:
```
print(df["City"]=="Mumbai")&(df["Product_Category"]=="Electronics").count().sum()
```

```
0      False
1       True
2      False
3      False
4       True
       ...
195    False
196     True
197    False
198     True
199     True
Name: City, Length: 200, dtype: bool
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[95], line 1
----> 1 print(df["City"]=="Mumbai")&(df["Product_Category"]=="Electronics").count().
sum()

TypeError: unsupported operand type(s) for &: 'NoneType' and 'int'
```

In [96]:
```
print(df[(df["City"] == "Mumbai") & (df["Product_Category"] == "Electronics")])
```

```
    Order_ID Customer_ID  Gender        Age    City Product_Category  \
4      O1004        C108    Male  60.000000  Mumbai      Electronics
20     O1020        C130    Male  49.000000  Mumbai      Electronics
59     O1059        C117    Male  31.000000  Mumbai      Electronics
93     O1093        C115    Male  44.000000  Mumbai      Electronics
100    O1100        C101  Female  40.237113  Mumbai      Electronics
159    O1159        C115  Female  59.000000  Mumbai      Electronics
169    O1169        C142  Female  45.000000  Mumbai      Electronics
185    O1185        C113  Female  23.000000  Mumbai      Electronics

     Quantity  Unit_Price Order_Date Payment_Mode  Total_amount     Month  \
4           4         999 2024-06-27         Card          3996      June
20          2        1990 2024-05-26       Wallet          3980       May
59          5        1771 2024-05-16          UPI          8855       May
93          4         258 2024-05-27          UPI          1032       May
100         2         725 2024-08-03          UPI          1450    August
159         3        1586 2024-05-30       Wallet          4758       May
169         2         210 2024-08-29         Card           420    August
185         5        1535 2024-12-20         Card          7675  December

     Quantity_mean
4             2.97
20            2.97
59            2.97
93            2.97
100           2.97
159           2.97
169           2.97
185           2.97
```

In [100…
```
print("ord_mnth_june:",df["Order_ID"]=="June")
```

```
ord_mnth_june: 0      False
1      False
2      False
3      False
4      False
       ...
195    False
196    False
197    False
198    False
199    False
Name: Order_ID, Length: 200, dtype: bool
```

SORTING

In [101…
```
df["Total_amount"].sort_values(ascending=True)
```

Out[101…
```
49      206
137     226
179     254
9       286
7       361
       ...
188    8855
134    9110
29     9135
52     9480
187    9925
Name: Total_amount, Length: 200, dtype: int64
```

In [103…
```
df.to_csv("retail_sales_cleaned.csv",index=False)
```

In [ ]:
```
```