Customer Shopping Behavior Data Analysis

In [1]:
```
import numpy as np
import pandas as pd
df= pd.read_csv("C:/Users/ankit/Downloads/customer_shopping_behavior.csv")
```

Headers

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

Out[2]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise

Information of Tables columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             3900 non-null   int64  
 1   Age                     3900 non-null   int64  
 2   Gender                  3900 non-null   object 
 3   Item Purchased          3900 non-null   object 
 4   Category                3900 non-null   object 
 5   Purchase Amount (USD)   3900 non-null   int64  
 6   Location                3900 non-null   object 
 7   Size                    3900 non-null   object 
 8   Color                   3900 non-null   object 
 9   Season                  3900 non-null   object 
 10  Review Rating           3863 non-null   float64
 11  Subscription Status     3900 non-null   object 
 12  Shipping Type           3900 non-null   object 
 13  Discount Applied        3900 non-null   object 
 14  Promo Code Used         3900 non-null   object 
 15  Previous Purchases      3900 non-null   int64  
 16  Payment Method          3900 non-null   object 
 17  Frequency of Purchases  3900 non-null   object 
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

Statistics Summary of Numerical columns

In [4]:
```
df.describe(include= "all")
```

Out[4]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	S
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	39
unique	NaN	NaN	2	25	4	NaN	50	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	
freq	NaN	NaN	2652	171	1737	NaN	96	17
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	N
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	N
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	N
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	N
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	N
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	N
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	N

How to get null values of table

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Customer ID                0
Age                        0
Gender                     0
Item Purchased             0
Category                   0
Purchase Amount (USD)      0
Location                   0
Size                       0
Color                      0
Season                     0
Review Rating             37
Subscription Status        0
Shipping Type              0
Discount Applied           0
Promo Code Used            0
Previous Purchases         0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```

We handle missing values in the review rating column by imputing them with the median rating calculated within

each category. This preserves category-level distribution and avoids bias caused by global aggregation.

In [6]:
```
df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x: x
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             0
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64
```

We standardize column names to snake_case using pandas string methods to improve readability, consistency, and ease of access.

In [8]:
```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
df = df.rename(columns={'purchase_amount_(usd)': 'purchase_amount'})
df.columns
```

Out[8]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'promo_code_used', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

Create a new column age_group

In [9]:
```
labels = ['Young age', 'Adult', 'Middle-aged', 'senior']
df['age_group'] = pd.qcut(df['age'], q=4, labels = labels)
```

In [10]:
```
df[['age', 'age_group']].head(5)
```

Out[10]:

	age	age_group
0	55	Middle-aged
1	19	Young age
2	50	Middle-aged
3	21	Young age
4	45	Middle-aged

Create a new column purchase_frequency_days

In [11]:
```
frequency_mapping = {
    'Fortnightly': 14,
    'Weekly' : 7,
    'Monthly' : 30,
    'Quarterly' : 90,
    'Bi-Weekly': 14,
    'Annually' : 365,
    'Every 3 Months' : 90
}
df['purchase_frequency_days'] = df['frequency_of_purchases'].map(frequency_mapping)
```

In [12]:
```
df[['purchase_frequency_days', 'frequency_of_purchases']].head(10)
```

Out[12]:

	purchase_frequency_days	frequency_of_purchases
0	14	Fortnightly
1	14	Fortnightly
2	7	Weekly
3	7	Weekly
4	365	Annually
5	7	Weekly
6	90	Quarterly
7	7	Weekly
8	365	Annually
9	90	Quarterly

How to remove the columns-discount_applied columns and promo_code_used columns have same values

In [13]:
```
df[['discount_applied','promo_code_used']].head(10)
```

Out[13]:

	discount_applied	promo_code_used
0	Yes	Yes
1	Yes	Yes
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
5	Yes	Yes
6	Yes	Yes
7	Yes	Yes
8	Yes	Yes
9	Yes	Yes

In [14]:
```
(df['discount_applied'] == df['promo_code_used']).all()
```

Out[14]:
```
True
```

In [15]:
```
df = df.drop('promo_code_used', axis = 1)
```

In [16]:
```
df.columns
```

Out[16]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'previous_purchases', 'payment_method',
       'frequency_of_purchases', 'age_group', 'purchase_frequency_days'],
      dtype='object')
```

How to connect with MySQL

In [17]:
```
!pip install pymysql
!pip install sqlalchemy
```

```
Requirement already satisfied: pymysql in c:\users\ankit\anaconda3\lib\site-packages
(1.1.2)
Requirement already satisfied: sqlalchemy in c:\users\ankit\anaconda3\lib\site-packa
ges (2.0.34)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\ankit\anaconda3
\lib\site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\ankit\anaconda3\lib\site
-packages (from sqlalchemy) (3.0.1)
```

In [18]:
```
from sqlalchemy import create_engine

engine = create_engine(
    "mysql+pymysql://root:root@localhost:3306/customer_behavior"
)
```

```
# write DataFrame to MySQL
table_name = "mytable"
df.to_sql(table_name, engine, if_exists='replace', index=False)

#Read back sample
pd.read_sql("SELECT * FROM mytable LIMIT 5;", engine)
```

Out[18]:

	customer_id	age	gender	item_purchased	category	purchase_amount	location
0	1	55	Male	Blouse	Clothing	53	Kentucky
1	2	19	Male	Sweater	Clothing	64	Maine
2	3	50	Male	Jeans	Clothing	73	Massachusetts
3	4	21	Male	Sandals	Footwear	90	Rhode Island
4	5	45	Male	Blouse	Clothing	49	Oregon

In [ ]: