In [1]:

```
# Loading the dataset using pandas
import pandas as pd
df = pd.read_csv('customer_shopping_behavior.csv')
```

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

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
```

In [4]:

```
# Summary statistics using .describe()
df.describe()
```

Out[4]:

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

In [5]:

```
# Checking if missing data or null values are present in the dataset
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
```

In [6]:

```
# Imputing missing values in Review Rating column with the median rating of the pro

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
```

In [8]:

```
# Renaming columns according to snake casing for better readability and documentati

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

In [9]:

```
df.columns
```

Out[9]:

```
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')
```

In [10]:

```
# create a new column age_group
labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior']
df['age_group'] = pd.qcut(df['age'], q=4, labels = labels)
```

In [11]:

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

Out[11]:

	age	age_group
0	55	Middle-aged
1	19	Young Adult
2	50	Middle-aged
3	21	Young Adult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Young Adult
8	26	Young Adult
9	57	Middle-aged

In [12]:

```
df['frequency_of_purchases'].unique()
```

Out[12]:

```
array(['Fortnightly', 'Weekly', 'Annually', 'Quarterly', 'Bi-Weekly',
       'Monthly', 'Every 3 Months'], dtype=object)
```

In [13]:

```
# create new column purchase_frequency_days

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 [14]:

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

Out[14]:

	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

In [15]:

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

Out[15]:

	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 [16]:

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

Out[16]:

```
True
```

In [17]:

```
# Dropping promo code used column

df = df.drop('promo_code_used', axis=1)
```

In [18]:

```
df.columns
```

Out[18]:

```
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')
```

In [19]:

```
pip install pymysql sqlalchemy
```

```
Note: you may need to restart the kernel to use updated packages.
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pymysql in c:\users\hp\appdata\roaming\python\python3
12\site-packages (1.1.0)
Requirement already satisfied: sqlalchemy in c:\users\hp\appdata\roaming\python\pyth
on312\site-packages (2.0.38)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\hp\appdata\roaming\pytho
n\python312\site-packages (from sqlalchemy) (3.1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\hp\appdata\roami
ng\python\python312\site-packages (from sqlalchemy) (4.12.2)
```

```
[notice] A new release of pip is available: 25.0.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [21]:

```python
from sqlalchemy import create_engine

# MySQL connection
username = "root"
password = "eternallight#492000"
host = "localhost"
port = "3306"
database = "customer_trend_analysis"

engine = create_engine(f"mysql+pymysql://{username}:{password}@{host}:{port}/{datab

# Write DataFrame to MySQL
table_name = "customer"   # choose any table name
df.to_sql(table_name, engine, if_exists="replace", index=False)

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

Out[21]:

	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 [ ]:

In [ ]: