In [4]:
```
import pandas as pd
df = pd.read_csv(r"C:\Users\user\OneDrive\Desktop\customer_shopping_behavior.csv
```

In [5]:
```
df.head()
```

Out[5]:

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

In [6]:
```
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 [9]:
```
df.describe(include='all')
```

Out[9]:

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

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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 [11]:
```
df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
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 [25]:
```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ','_')
df = df.rename(columns={'purchase_amount_(usd)':'purchase_amount'})
```

In [29]:
```
df.columns
```

Out[29]:
```
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', 'age_group'],
      dtype='object')
```

In [30]:
```
# create a column for age-group
labels = ['Young Adult', 'Adult', 'Middle-aged', 'Senior']
df['age_group']= pd.qcut(df['age'], q=4, labels=labels)
```

In [31]:
```
df[['age','age_group']].head()
```

Out[31]:

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

In [32]:
```
#create column for 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_mappin
```

In [38]:
```
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

Out[38]:

	purchase_frequency_days	frequency_of_purchases
0	14.0	Fortnightly
1	14.0	Fortnightly
2	7.0	Weekly
3	7.0	Weekly
4	365.0	Annually
5	7.0	Weekly
6	90.0	Quarterly
7	7.0	Weekly
8	365.0	Annually
9	90.0	Quarterly

In [41]:
```
df[['discount_applied', 'promo_code_used']].head(10)
```

Out[41]:

	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 [51]:
```
df['discount_applied'].equals(df['promo_code_used'])
```

Out[51]:
```
True
```

In [52]:
```
(df['discount_applied'] == df['promo_code_used']).all()
```

Out[52]:
```
np.True_
```

In [54]:
```
df = df.drop('promo_code_used',axis=1)
```

In [58]:
```
df.columns
```

Out[58]:
```
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 [60]:
```
pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\user\sql_notebook\env
\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\user\sql_notebook\env\lib\s
ite-packages (2.0.43)
Requirement already satisfied: greenlet>=1 in c:\users\user\sql_notebook\env\lib
\site-packages (from sqlalchemy) (3.2.4)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\user\sql_note
book\env\lib\site-packages (from sqlalchemy) (4.15.0)
Note: you may need to restart the kernel to use updated packages.
[notice] A new release of pip is available: 24.3.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [65]:
```
from sqlalchemy import create_engine
from urllib.parse import quote_plus
username="postgres"
password= quote_plus("Shasti@1369")
host="localhost"
port="5432"
database="customer_behavior"

engine = create_engine(f"postgresql+psycopg2://{username}:{password}@{host}:{por

table_name = "customer"
df.to_sql(table_name, engine, if_exists="replace", index=False)

print(f"Data successfully loaded into table '{table_name}' in database '{databas
```

```
Data successfully loaded into table 'customer' in database 'customer_behavior'.
```

In [ ]: