

In [5]:

```
import pandas as pd
df = pd.read_csv("customer_shopping_behavior.csv")
df.head()
```

Out[5]:

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

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 [8]:

```
df.describe(include='all') #include all will include all character columns
```

Out[8]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900
unique	Nan	Nan	2	25	4	Nan	50	4	25
top	Nan	Nan	Male	Blouse	Clothing	Nan	Montana	M	Olive
freq	Nan	Nan	2652	171	1737	Nan	96	1755	177
mean	1950.500000	44.068462	Nan	Nan	Nan	59.764359	Nan	Nan	Nan
std	1125.977353	15.207589	Nan	Nan	Nan	23.685392	Nan	Nan	Nan
min	1.000000	18.000000	Nan	Nan	Nan	20.000000	Nan	Nan	Nan
25%	975.750000	31.000000	Nan	Nan	Nan	39.000000	Nan	Nan	Nan
50%	1950.500000	44.000000	Nan	Nan	Nan	60.000000	Nan	Nan	Nan
75%	2925.250000	57.000000	Nan	Nan	Nan	81.000000	Nan	Nan	Nan
max	3900.000000	70.000000	Nan	Nan	Nan	100.000000	Nan	Nan	Nan

In [9]:

```
df.isnull().sum() #hello
```

Out[9]:

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 [ ]:

In [13]:

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

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
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 [3]:

```
import pandas as pd
df = pd.read_csv("customer_shopping_behavior.csv")
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
df = df.rename(columns={'purchase_amount_(usd)':'purchase_amount'})
```

In [4]:

```
df.columns
```

Out[4]:

```
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 [6]:

```
# create a column age_group
labels = [ 'Youngadult','Adult','Middle-aged','Senior']
df['age_group'] = pd.qcut(df['age'], q=4 , labels = labels)
df[['age','age_group']].head(10)
```

Out[6]:

	age	age_group
0	55	Middle-aged
1	19	Youngadult
2	50	Middle-aged
3	21	Youngadult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Youngadult

age	age_group
-----	-----------

8	26	Youngadult
---	----	------------

9	57	Middle-aged
---	----	-------------

In [11]:

```
# create column purchases_frequency_days
# map function is used to replace text
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)
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

Out[11]:

	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 [12]:

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

Out[12]:

	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

	discount_applied	promo_code_used
7	Yes	Yes
8	Yes	Yes
9	Yes	Yes

In [16]:

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

Out[16]:

True

In [17]:

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

Out[17]:

```
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', 'pirchases_frequency_days',
       'purchases_frequency_days', 'purchase_frequency_days'],
      dtype='object')
```

In [18]:

```
pip install psycopg2-binary sqlalchemy
```

Defaulting to user installation because normal site-packages is not writeable  
Note: you may need to restart the kernel to use updated packages.

Collecting psycopg2-binary

```
  Downloading psycopg2_binary-2.9.11-cp312-cp312-win_amd64.whl.metadata (5.1 kB)
Requirement already satisfied: sqlalchemy in c:\programdata\anaconda3\lib\site-packages (2.0.34)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\programdata\anaconda3\lib\site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\programdata\anaconda3\lib\site-packages (from sqlalchemy) (3.0.1)
  Downloading psycopg2_binary-2.9.11-cp312-cp312-win_amd64.whl (2.7 MB)
----- 0.0/2.7 MB ? eta ------
----- 1.0/2.7 MB 6.3 MB/s eta 0:00:01
----- 2.4/2.7 MB 6.4 MB/s eta 0:00:01
----- 2.7/2.7 MB 6.0 MB/s eta 0:00:00
```

Installing collected packages: psycopg2-binary

Successfully installed psycopg2-binary-2.9.11

In [20]:

```
!pip install psycopg2-binary
```

Defaulting to user installation because normal site-packages is not writeable  
Requirement already satisfied: psycopg2-binary in c:\users\surjeet verma\appdata\roaming\python\python312\site-packages (2.9.11)

In [21]:

```
import sys
!{sys.executable} -m pip install psycopg2-binary
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: psycopg2-binary in c:\users\surjeet verma\appdata\roaming
\python\python312\site-packages (2.9.11)
```

In [22]:

```
import sys
print(sys.executable)
```

C:\ProgramData\anaconda3\python.exe

In [23]:

```
!C:\ProgramData\anaconda3\python.exe -m pip install psycopg2-binary
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: psycopg2-binary in c:\users\surjeet verma\appdata\roaming
\python\python312\site-packages (2.9.11)
```

In [ ]:

```
!conda install -c conda-forge psycopg2-binary -y
```

In [1]:

```
import psycopg2
print("psycopg2 installed successfully!")
```

psycopg2 installed successfully!

In [3]:

```
import pandas as pd
from sqlalchemy import create_engine

# Load CSV
df = pd.read_csv("customer_shopping_behavior.csv")

# Connection details
username = "postgres"
password = "MyNewPass123!"      # URL-encoded password
host = "localhost"
port = "5432"
database = "customer_beaviour"   # match pgAdmin spelling

# Create engine
engine = create_engine(
    f"postgresql+psycopg2://{{username}}:{{password}}@{{host}}:{{port}}/{{database}}"
)

# Table name
table_name = "customer"

# Load dataframe into PostgreSQL
df.to_sql(table_name, engine, if_exists="replace", index=False)

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

Data successfully loaded into table 'customer' in database 'customer\_beaviour'.

In [ ]: