In [1]:
```
import pandas as pd
df = pd.read_csv("C:\\Users\\VIHAL\\OneDrive\\Desktop\\customer_shopping_behavior.csv")
```

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

Out[2]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo	Fortnightly
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	Cash	Fortnightly
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	Credit Card	Weekly
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	PayPal	Weekly
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	PayPal	Annually

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]:
```
df.describe(include='all')
```

Out[4]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	3900	3900	3900.000000	3900	3900
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	2	2	NaN	6	7
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	No	No	NaN	PayPal	Every 3 Months
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	2223	2223	NaN	677	584
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	NaN	NaN	25.351538	NaN	NaN
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	NaN	NaN	14.447125	NaN	NaN
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	NaN	NaN	1.000000	NaN	NaN
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	NaN	NaN	13.000000	NaN	NaN
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	NaN	NaN	25.000000	NaN	NaN
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	NaN	NaN	38.000000	NaN	NaN
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	NaN	NaN	50.000000	NaN	NaN

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
```

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

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]:
```
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 column age_groups
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]:
```
# create 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 [13]:
```
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

Out[13]:

	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 [14]:
```
df[['discount_applied','promo_code_used']].head(10)
```

Out[14]:

	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 [15]:
```
(df['discount_applied'] == df['promo_code_used']).all()
```

Out[15]:
```
True
```

In [16]:
```
# Hence Both columns carry exactly ssame informations.
# So, we'll remove the column 'promo_code_used'

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

In [17]:
```
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', 'purchase_frequency_days'],
      dtype='object')
```

Connecting Python script to PostgreSQL

In [19]:
```
pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\vihal\anaconda3\lib\site-packages (2.9.11)Note: you may need to restart the kernel to use updated packages.


Requirement already satisfied: sqlalchemy in c:\users\vihal\anaconda3\lib\site-packages (2.0.30)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\vihal\anaconda3\lib\site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\vihal\anaconda3\lib\site-packages (from sqlalchemy) (3.0.1)
```

In [20]:
```
from sqlalchemy import create_engine

# Step 1: Connect to PostgreSQL
# Replace placeholders with your actual details
username = "postgres"     # default user
password = "1803" # the password you set during installation
host = "localhost"        # if running locally
port = "5432"             # default PostgreSQL port
database = "customer_behavior"    # the database you created in pgAdmin

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

# Step 2: Load DataFrame into PostgreSQL
table_name = "customer"   # choose any table name
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_behavior'.
```

Code my MySQL

In [22]:
```
!pip install pymysql sqlalchemy
```

```
Collecting pymysql
  Downloading pymysql-1.1.2-py3-none-any.whl.metadata (4.3 kB)
Requirement already satisfied: sqlalchemy in c:\users\vihal\anaconda3\lib\site-packages (2.0.30)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\vihal\anaconda3\lib\site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\vihal\anaconda3\lib\site-packages (from sqlalchemy) (3.0.1)
Downloading pymysql-1.1.2-py3-none-any.whl (45 kB)
   ---------------------------------------- 0.0/45.3 kB ? eta -:--:--
   ---------- ----------------------------- 12.2/45.3 kB ? eta -:--:--
   ---------------------------- ----------- 30.7/45.3 kB 262.6 kB/s eta 0:00:01
   ---------------------------------------- 45.3/45.3 kB 280.5 kB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.1.2
```

In [ ]:
```
from sqlalchemy import create_engine

# MySQL connection
username = "root"
password = "your_password"
host = "localhost"
port = "3306"
database = "customer_behavior"

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

# 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)
```

Code for MS SQL Server

In [ ]:
```
!pip install pyodbc sqlalchemy
```

In [ ]:
```
# Install required libraries

from sqlalchemy import create_engine
from urllib.parse import quote_plus

# SQL Server connection
username = "sa"
password = "your_password"
host = "localhost"
port = "1433"
database = "customer_behavior"

# Note: requires Microsoft ODBC Driver installed separately on your machine
driver = quote_plus("ODBC Driver 17 for SQL Server")
engine = create_engine(f"mssql+pyodbc://{username}:{password}@{host},{port}/{database}?driver={driver}")

# Write DataFrame to SQL Server
df.to_sql("customer", engine, if_exists="replace", index=False)

# Read back sample (SQL Server uses TOP instead of LIMIT)
pd.read_sql("SELECT TOP 5 * FROM customer;", engine)
```