In [56]:
```
import pandas as pd
```

In [57]:
```
df = pd.read_csv("S:/pro/face-attendance/customer_shopping_behavior.csv")
df.head(10)
```

Out[57]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	
1	2	19	Male	Sweater	Clothing	64	Maine	L	M
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	M
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	M
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turc
5	6	46	Male	Sneakers	Footwear	20	Wyoming	M	
6	7	63	Male	Shirt	Clothing	85	Montana	M	
7	8	27	Male	Shorts	Clothing	34	Louisiana	L	Cha
8	9	26	Male	Coat	Outerwear	97	West Virginia	L	
9	10	57	Male	Handbag	Accessories	31	Missouri	M	

In [58]:
```
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 [59]:
```
df.describe()
```

Out[59]:

	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 [60]:
```
df.isnull().sum()
```

Out[60]:
```
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 [61]:
```
df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x:›
```

In [62]:
```
df.isnull().sum()
```

Out[62]:
```
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 [63]:
```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ','_')
df = df.rename(columns = {'purchase_amount_(usd)' : 'purchase_amount'})
```

In [64]:
```
df.columns
```

Out[64]:
```
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 [65]:
```
# 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 [66]:
```
df[['age','age_group']].head(10)
```

Out[66]:

	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 [67]:
```
# 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 [68]:
```
df[['Purchase_frequency_days','frequency_of_purchases']].head(10)
```

Out[68]:

	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 [69]:
```
df[['discount_applied','promo_code_used']].head(10)
```

Out[69]:

	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 [70]:
```
df[['discount_applied','promo_code_used']].all()
```

Out[70]:
```
discount_applied    True
promo_code_used     True
dtype: bool
```

In [71]:
```
df =df.drop('promo_code_used',axis = 1)
```

In [72]:
```
df.columns
```

Out[72]:
```
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 [73]:
```
!pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\gunam\appdata\local\prog
rams\python\python313\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\gunam\appdata\local\programs
\python\python313\lib\site-packages (2.0.44)
Requirement already satisfied: greenlet>=1 in c:\users\gunam\appdata\local\programs
\python\python313\lib\site-packages (from sqlalchemy) (3.2.4)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\gunam\appdata\l
ocal\programs\python\python313\lib\site-packages (from sqlalchemy) (4.15.0)
[notice] A new release of pip is available: 25.3 -> 26.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [74]:
```
from sqlalchemy import create_engine
from urllib.parse import quote_plus
# step 1: Connect to postgreSQL
# Replace placeholder with your actual details

username = "DESKTOP2024\gunam" # default user
```

```
password = "sachin123" # the password you set during installation
host = "localhost"    # if running locally
port = "1433"         # default postgres port
database = "customer_behavior"  # the database you created in pgAdmin

# Note:
driver = quote_plus("ODBC Driver 17 for SQL Server")
engine = create_engine(f"mssql+pyodbc://{username}:{password}@{host}:{port}/{datab

# step 2: Load database into postgreSQL
table_name = "customer" # choose any table name
df.to_sql(table_name, engine, if_exist="replace", index = False)

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

```
<>:6: SyntaxWarning: invalid escape sequence '\g'
<>:6: SyntaxWarning: invalid escape sequence '\g'
C:\Users\gunam\AppData\Local\Temp\ipykernel_20832\2652939740.py:6: SyntaxWarning: i
nvalid escape sequence '\g'
  username = "DESKTOP2024\gunam" # default user
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[74], line 14
     12 # Note:
     13 driver = quote_plus("ODBC Driver 17 for SQL Server")
---> 14 engine = create_engine(                {username} {password} {host} {por
t} {database}        {driver} )
     16 # step 2: Load database into postgreSQL
     17 table_name = "customer" # choose any table name

File <string>:2, in create_engine(url, **kwargs)

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\sqlalchemy\util\de
precations.py:281, in deprecated_params.<locals>.decorate.<locals>.warned(fn, *arg
s, **kwargs)
    274     if m in kwargs:
    275         _warn_with_version(
    276             messages[m],
    277             versions[m],
    278             version_warnings[m],
    279             stacklevel=3,
    280         )
--> 281 return fn(*args, **kwargs)

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\sqlalchemy\engine
\create.py:617, in create_engine(url, **kwargs)
    615         if k in kwargs:
    616             dbapi_args[k] = pop_kwarg(k)
--> 617     dbapi = dbapi_meth(**dbapi_args)
    619 dialect_args["dbapi"] = dbapi
    621 dialect_args.setdefault("compiler_linting", compiler.NO_LINTING)

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\sqlalchemy\connect
ors\pyodbc.py:58, in PyODBCConnector.import_dbapi(cls)
     56 @classmethod
     57 def import_dbapi(cls) -> DBAPIModule:
---> 58     return __import__(        )

ModuleNotFoundError: No module named 'pyodbc'
```

In [ ]:

In [ ]: