CUSTOMER ANALYSIS PROJECT

In [11]:
```
import pandas as pd
```

In [12]:
```
df =pd.read_csv("customer_shopping_behavior.csv")
```

In [13]:
```
df.head(10)
```

Out[13]:

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

In [14]:
```
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 [15]:
```
df.describe()
```

Out[15]:

	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 [16]:
```
df.isnull().sum()
```

Out[16]:
```
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
```

Replace the null values with meadian review rating withiin each category

In [17]:
```
df['Review Rating'] =df.groupby('Category')['Review Rating'].transform(lambda x: x.
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_7984\3041723296.py in ?()
----> 1 df['Review Rating'] =df.groupby('Category')['Review Rating'].transform(lambd
a x: x.fillna(x.meadian()))

~\PyCharmMiscProject\.venv\Lib\site-packages\pandas\core\groupby\generic.py in ?(sel
f, func, engine, engine_kwargs, *args, **kwargs)
    514     @Substitution(klass="Series", example=__examples_series_doc)
    515     @Appender(_transform_template)
    516     def transform(self, func, *args, engine=None, engine_kwargs=None, **kwar
gs):
--> 517         return self._transform(
    518             func, *args, engine=engine, engine_kwargs=engine_kwargs, **kwarg
s
    519         )

~\PyCharmMiscProject\.venv\Lib\site-packages\pandas\core\groupby\groupby.py in ?(sel
f, func, engine, engine_kwargs, *args, **kwargs)
   2024         if orig_func != func:
   2025             warn_alias_replacement(self, orig_func, func)
   2026
   2027         if not isinstance(func, str):
-> 2028             return self._transform_general(func, engine, engine_kwargs, *arg
s, **kwargs)
   2029
   2030         elif func not in base.transform_kernel_allowlist:
   2031             msg = f"'{func}' is not a valid function name for transform(nam
e)"

~\PyCharmMiscProject\.venv\Lib\site-packages\pandas\core\groupby\generic.py in ?(sel
f, func, engine, engine_kwargs, *args, **kwargs)
    553             self._obj_with_exclusions, axis=self.axis
    554         ):
    555             # this setattr is needed for test_transform_lambda_with_datetime
tz
    556             object.__setattr__(group, "name", name)
--> 557             res = func(group, *args, **kwargs)
    558
    559             results.append(klass(res, index=group.index))
    560

~\AppData\Local\Temp\ipykernel_7984\3041723296.py in ?(x)
----> 1 df['Review Rating'] =df.groupby('Category')['Review Rating'].transform(lambd
a x: x.fillna(x.meadian()))

~\PyCharmMiscProject\.venv\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   6317             and name not in self._accessors
   6318             and self._info_axis._can_hold_identifiers_and_holds_name(name)
   6319         ):
   6320             return self[name]
-> 6321         return object.__getattribute__(self, name)

AttributeError: 'Series' object has no attribute 'meadian'
```

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
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 [19]:
```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ' , '_')
```

In [20]:
```
df.columns
```

Out[20]:
```
Index(['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'],
      dtype='object')
```

In [21]:
```
df= df.rename(columns={'purchase_amount_(usd)' :'purchase_amount'})
```

In [22]:
```
df.columns
```

Out[22]:
```
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 [23]:
```
labels = ['Child', 'Young Adult', 'Adult', 'Senior']

# Corrected function: pd.qcut and parameter: labels
df['age_group'] = pd.qcut(df['age'], q=4, labels=labels)
```

In [24]:
```
df.columns
```

Out[24]:
```
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 [25]:
```
df.describe()
```

Out[25]:

	customer_id	age	purchase_amount	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 [26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 19 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         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
 18  age_group               3900 non-null   category
dtypes: category(1), float64(1), int64(4), object(13)
memory usage: 552.6+ KB
```

In [27]:
```
df[['age','age_group']].head(2)
```

Out[27]:

	age	age_group
0	55	Adult
1	19	Child

In [28]:
```
# purchae 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 [29]:
```
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

Out[29]:

	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 [30]:
```
df[['discount_applied','promo_code_used']].head(80)
```

Out[30]:

	discount_applied	promo_code_used
0	Yes	Yes
1	Yes	Yes
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
...	...	...
75	Yes	Yes
76	Yes	Yes
77	Yes	Yes
78	Yes	Yes
79	Yes	Yes

80 rows × 2 columns

In [31]:
```
(df['discount_applied'] == df['promo_code_used']).all()
```

Out[31]:
```
np.True_
```

In [32]:
```
df.drop('promo_code_used',axis=1)
```

Out[32]:

	customer_id	age	gender	item_purchased	category	purchase_amount	locat
0	1	55	Male	Blouse	Clothing	53	Kentu
1	2	19	Male	Sweater	Clothing	64	Ma
2	3	50	Male	Jeans	Clothing	73	Massachus
3	4	21	Male	Sandals	Footwear	90	Rhode Isl
4	5	45	Male	Blouse	Clothing	49	Ore
...	...	...	...	...	...	...	
3895	3896	40	Female	Hoodie	Clothing	28	Virg
3896	3897	52	Female	Backpack	Accessories	49	I
3897	3898	46	Female	Belt	Accessories	33	New Je
3898	3899	44	Female	Shoes	Footwear	77	Minnes
3899	3900	52	Female	Handbag	Accessories	81	Califo

3900 rows × 19 columns

In [33]:
```
df.columns
```

Out[33]:
```
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',
       'purchase_frequency_days'],
      dtype='object')
```

In [34]:
```
pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\admin\pycharmmiscproject
\.venv\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\admin\pycharmmiscproject\.venv
\lib\site-packages (2.0.45)
Requirement already satisfied: greenlet>=1 in c:\users\admin\pycharmmiscproject\.ven
v\lib\site-packages (from sqlalchemy) (3.3.0)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\admin\pycharmmis
cproject\.venv\lib\site-packages (from sqlalchemy) (4.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [40]:
```
from sqlalchemy import create_engine

# step 1: Connect to PostgreSQL
# Replace placeholders with your actual details

username = "postgres"
```

```
password = "root"
host = "localhost"   # if running locally
port= "5432"
database = "Customer"

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

# step 2: Load DataFrame into PostgreSQL
table_name= "customers"    # 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 'customers' in database 'Customer'.
```

In [ ]: