```python
# Loading the dataset using pandas

import pandas as pd

df = pd.read_csv('customer_shopping_behavior.csv')

df.head()
```

```
   Customer ID  Age Gender Item Purchased  Category  Purchase Amount
(USD)  \
0            1   55   Male         Blouse  Clothing
53
1            2   19   Male        Sweater  Clothing
64
2            3   50   Male          Jeans  Clothing
73
3            4   21   Male        Sandals  Footwear
90
4            5   45   Male         Blouse  Clothing
49

        Location Size      Color  Season  Review Rating Subscription
Status  \
0       Kentucky    L       Gray  Winter            3.1
Yes
1          Maine    L     Maroon  Winter            3.1
Yes
2  Massachusetts    S     Maroon  Spring            3.1
Yes
3   Rhode Island    M     Maroon  Spring            3.5
Yes
4         Oregon    M  Turquoise  Spring            2.7
Yes

   Shipping Type Discount Applied Promo Code Used  Previous Purchases
\
0        Express              Yes              Yes                  14

1        Express              Yes              Yes                   2

2  Free Shipping              Yes              Yes                  23

3   Next Day Air              Yes              Yes                  49

4  Free Shipping              Yes              Yes                  31


  Payment Method Frequency of Purchases
0          Venmo            Fortnightly
1           Cash            Fortnightly
2    Credit Card                 Weekly
```

```
3         PayPal                 Weekly
4         PayPal               Annually
```

```
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
```

```
# Summary statistics using .describe()
df.describe(include='all')
```

```
        Customer ID          Age Gender Item Purchased  Category  \
count   3900.000000  3900.000000   3900           3900      3900   
unique          NaN          NaN      2             25         4   
top             NaN          NaN   Male         Blouse  Clothing   
freq            NaN          NaN   2652            171      1737   
mean    1950.500000    44.068462    NaN            NaN       NaN   
std     1125.977353    15.207589    NaN            NaN       NaN   
min        1.000000    18.000000    NaN            NaN       NaN   
25%      975.750000    31.000000    NaN            NaN       NaN   
50%     1950.500000    44.000000    NaN            NaN       NaN   
75%     2925.250000    57.000000    NaN            NaN       NaN   
max     3900.000000    70.000000    NaN            NaN       NaN   

        Purchase Amount (USD) Location  Size  Color  Season  Review 
Rating  \
count             3900.000000     3900  3900   3900    3900   
```

```
3863.000000
unique                    NaN       50     4     25       4
NaN
top                       NaN  Montana     M  Olive  Spring
NaN
freq                      NaN       96  1755    177     999
NaN
mean                59.764359      NaN   NaN    NaN     NaN
3.750065
std                 23.685392      NaN   NaN    NaN     NaN
0.716983
min                 20.000000      NaN   NaN    NaN     NaN
2.500000
25%                 39.000000      NaN   NaN    NaN     NaN
3.100000
50%                 60.000000      NaN   NaN    NaN     NaN
3.800000
75%                 81.000000      NaN   NaN    NaN     NaN
4.400000
max                100.000000      NaN   NaN    NaN     NaN
5.000000

       Subscription Status  Shipping Type Discount Applied Promo Code
Used  \
count                 3900           3900             3900
3900
unique                   2              6                2
2
top                     No  Free Shipping               No
No
freq                  2847            675             2223
2223
mean                   NaN            NaN              NaN
NaN
std                    NaN            NaN              NaN
NaN
min                    NaN            NaN              NaN
NaN
25%                    NaN            NaN              NaN
NaN
50%                    NaN            NaN              NaN
NaN
75%                    NaN            NaN              NaN
NaN
max                    NaN            NaN              NaN
NaN

        Previous Purchases Payment Method Frequency of Purchases
count          3900.000000           3900                   3900
```

```
unique                 NaN              6                      7
top                    NaN         PayPal         Every 3 Months
freq                   NaN            677                    584
mean             25.351538            NaN                    NaN
std              14.447125            NaN                    NaN
min               1.000000            NaN                    NaN
25%              13.000000            NaN                    NaN
50%              25.000000            NaN                    NaN
75%              38.000000            NaN                    NaN
max              50.000000            NaN                    NaN
```

```
# Checking if missing data or null values are present in the dataset

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

```
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
```

```
# Imputing missing values in Review Rating column with the median
rating of the product category

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

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

```
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
```

```
# Renaming columns according to snake casing for better readability
and documentation

df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ','_')
df = df.rename(columns={'purchase_amount_(usd)':'purchase_amount'})

df.columns
```

```
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')
```

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

df[['age','age_group']].head(10)
```

```
   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
```

```
# create new 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)

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

```
   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
```

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

```
  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
```

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

```
True
```

```
# Dropping promo code used column

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

df.columns
```

```
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

```
!pip install psycopg2-binary sqlalchemy

Requirement already satisfied: psycopg2-binary in c:\users\kiit\
anaconda3\lib\site-packages (2.9.10)
Requirement already satisfied: sqlalchemy in c:\users\kiit\anaconda3\
lib\site-packages (1.4.22)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\kiit\
anaconda3\lib\site-packages (from sqlalchemy) (1.1.1)
Note: you may need to restart the kernel to use updated packages.

from sqlalchemy import create_engine

# Step 1: Connect to PostgreSQL
# Replace placeholders with your actual details
username = "postgres"      # default user
password = "amlan123" # 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 for MySQL

```
!pip install pymysql sqlalchemy

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

```
!pip install pyodbc sqlalchemy

Requirement already satisfied: pyodbc in c:\users\kiit\anaconda3\lib\
site-packages (4.0.0-unsupported)
Requirement already satisfied: sqlalchemy in c:\users\kiit\anaconda3\
lib\site-packages (1.4.22)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\kiit\
anaconda3\lib\site-packages (from sqlalchemy) (1.1.1)

# 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)
```