Walmart Business Case Study

1. Defining Problem Statement and Analysing basic metrics

1.1 Problem Statement

The management team at Walmart Inc. wants to analyze customer purchase behavior, particularly the purchase amount, against various factors including gender, age, marital status, etc., to make informed business decisions. Their aim is to understand if there are differences in spending habits among different demographics. Specifically, they want to understand if there are differences in spending habits between male and female customers, and how factors like age and marital status affect spending.

1.2 Analyzing basic metrics

```
# Importing required Python libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Load the dataset
df = pd.read_csv("walmart_data.csv")
```

```
# Data types of all the attributes
print("\nData Types of Columns:")
print(df.dtypes)
```

```
Data Types of Columns:
User_ID                      int64
Product_ID                  object
Gender                      object
Age                         object
Occupation                   int64
City_Category               object
Stay_In_Current_City_Years  object
Marital_Status               int64
Product_Category             int64
Purchase                     int64
dtype: object
```

```python
# Observations on shape of data
print("Shape of the Dataset:")
print(df.shape)
```

```
Shape of the Dataset:
(550068, 10)
```

```python
# Check for missing values in each column
print("Missing values per column:")
print(df.isnull().sum())
```

```
Missing values per column:
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category              0
Purchase                      0
dtype: int64
```

```python
# Convert categorical attributes to 'category' if required
cat_cols = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
for col in cat_cols:
    df[col] = df[col].astype('category')
```

```python
df.dtypes
```

```
User_ID                        int64
Product_ID                    object
Gender                      category
Age                         category
Occupation                  category
City_Category               category
Stay_In_Current_City_Years  category
Marital_Status              category
Product_Category            category
Purchase                       int64
dtype: object
```

```python
# Display statistical summary
print("\nStatistical summary of numerical attributes:")
print(df.describe())
```

```
Statistical summary of numerical attributes:
            User_ID       Purchase
count  5.500680e+05  550068.000000
mean   1.003029e+06    9263.968713
std    1.727592e+03    5023.065394
min    1.000001e+06      12.000000
25%    1.001516e+06    5823.000000
50%    1.003077e+06    8047.000000
75%    1.004478e+06   12054.000000
max    1.006040e+06   23961.000000
```

```
# Getting first 5 rows of the data
df.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

```
# Getting last 5 rows of the data
df.tail()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
550063	1006033	P00372445	M	51-55	13	B	1	1	20	368
550064	1006035	P00375436	F	26-35	1	C	3	0	20	371
550065	1006036	P00375436	F	26-35	15	B	4+	1	20	137
550066	1006038	P00375436	F	55+	1	C	2	0	20	365
550067	1006039	P00371644	F	46-50	0	B	4+	1	20	490

Observations:

I. Shape of the data: The dataset contains 10 columns and 550068 rows.
II. Data types: Most columns are numeric (integer), while some are categorical (objects).
III. Conversion of categorical attributes: Categorical attributes like Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status, and Product_Category have been converted to 'category' type.
IV. Statistical summary: A statistical summary is provided, including count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for numeric attributes.

This analysis provides a basic understanding of the dataset structure and characteristics, which is essential for further exploration and analysis.

1.3 Non-Graphical Analysis: Value counts and unique attributes

```
# Non-Graphical Analysis: Value counts and unique attributes
# Value counts for each column
print("Value counts for each column:")
for column in df.columns:
    print(f"\n{column}:")
    print(df[column].value_counts())

# Unique attributes for categorical columns
print("\nUnique attributes for categorical columns:")
categorical_columns = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
for column in categorical_columns:
    print(f"\n{column}:")
    print(df[column].unique())
```

```
Value counts for each column:

User_ID:
User_ID
1001680    1026
1004277     979
1001941     898
1001181     862
1000889     823
           ...
1002690       7
1002111       7
1005810       7
```

```
1004991       7
1000708       6
Name: count, Length: 5891, dtype: int64

Product_ID:
Product_ID
P00265242    1880
P00025442    1615
P00110742    1612
P00112142    1562
P00057642    1470
             ...
P00314842       1
P00298842       1
P00231642       1
P00204442       1
P00066342       1
Name: count, Length: 3631, dtype: int64

Gender:
Gender
M    414259
F    135809
Name: count, dtype: int64

Age:
Age
26-35    219587
36-45    110013
18-25     99660
46-50     45701
51-55     38501
55+       21504
0-17      15102
Name: count, dtype: int64

Occupation:
Occupation
4     72308
0     69638
7     59133
1     47426
17    40043
20    33562
12    31179
14    27309
2     26588
16    25371
6     20355
3     17650
10    12930
5     12177
15    12165
11    11586
19     8461
13     7728
18     6622
9      6291
8      1546
Name: count, dtype: int64

City_Category:
City_Category
B    231173
C    171175
A    147720
Name: count, dtype: int64

Stay_In_Current_City_Years:
Stay_In_Current_City_Years
1     193821
2     101838
3      95285
4+     84726
0      74398
Name: count, dtype: int64

Marital_Status:
Marital_Status
0    324731
1    225337
Name: count, dtype: int64
```

```
Product_Category:
Product_Category
5     150933
1     140378
8     113925
11     24287
2      23864
6      20466
3      20213
4      11753
16      9828
15      6290
13      5549
10      5125
12      3947
7       3721
18      3125
20      2550
19      1603
14      1523
17       578
9        410
Name: count, dtype: int64

Purchase:
Purchase
7011     191
7193     188
6855     187
6891     184
7012     183
        ...
23491      1
18345      1
3372       1
855        1
21489      1
Name: count, Length: 18105, dtype: int64

Unique attributes for categorical columns:


Gender:
['F', 'M']
Categories (2, object): ['F', 'M']

Age:
['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25']
Categories (7, object): ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']

Occupation:
[10, 16, 15, 7, 20, ..., 18, 5, 14, 13, 6]
Length: 21
Categories (21, int64): [0, 1, 2, 3, ..., 17, 18, 19, 20]

City_Category:
['A', 'C', 'B']
Categories (3, object): ['A', 'B', 'C']

Stay_In_Current_City_Years:
['2', '4+', '3', '1', '0']
Categories (5, object): ['0', '1', '2', '3', '4+']

Marital_Status:
[0, 1]
Categories (2, int64): [0, 1]

Product_Category:
[3, 1, 12, 8, 5, ..., 10, 17, 9, 20, 19]
Length: 20
Categories (20, int64): [1, 2, 3, 4, ..., 17, 18, 19, 20]
```

Observations:

I. Gender:

- There are two genders: Male and Female.
- The number of male customers (414259) is greater than the number of female customers (135809).

II. Age:

- Age is categorized into bins.
- The age bins range from 0-17 years to 51+ years.
- There is a varied distribution of customers across different age groups.
- The majority of individuals fall within the (26-35) and (36-45) age range, while the least number of individuals are in the (0-17) age range.

III. Occupation:

- Occupation is a categorical variable representing different occupations of customers.
- The dataset includes various occupation categories ranging from 0-20
- The most common Occupation category is 4, followed by 0 and 7.
- The least common Occupation category is 8.

IV. City_Category:

- City_Category represents the category of the city where the customer resides (A, B, or C).
- The dataset includes customers from different city categories.
- The majority of individuals fall within B category and least number of individuals are found in the A Category.

V. Stay_In_Current_City_Years:

- Stay_In_Current_City_Years column represents the number of years the customer has stayed in the current city.
- The dataset includes customers stay ranging from (0-4+) years.
- The value counts indicate that most customers have stayed in the current city for 1 year, followed by 2 years and then 3 years.
- The number of customers decreases as the duration of stay increases beyond 3 years.
- The value counts indicate that least no. of customers have stayed in the current city for less than 1 year i.e. 0 years.

VI. Marital_Status:

- Marital_Status is binary, representing whether the customer is married (1) or unmarried (0).
- The dataset includes both married and unmarried customers.
- The majority of individuals are unmarried (324731), while fewer individuals are married (225337).

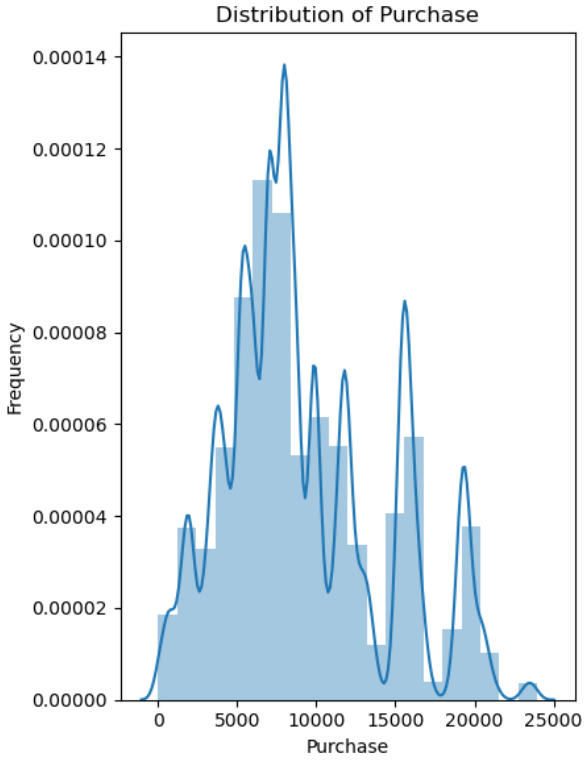
VII. Product_Category:

- Product_Category is a categorical variable representing different product categories.
- The dataset includes transactions from various product categories ranging from (1-20).
- The majority of individuals purchase products of category 5 (150933), followed by category 1 (140378) and category 8 (113925).
- The least number of individuals purchase products from Category 9 (410).

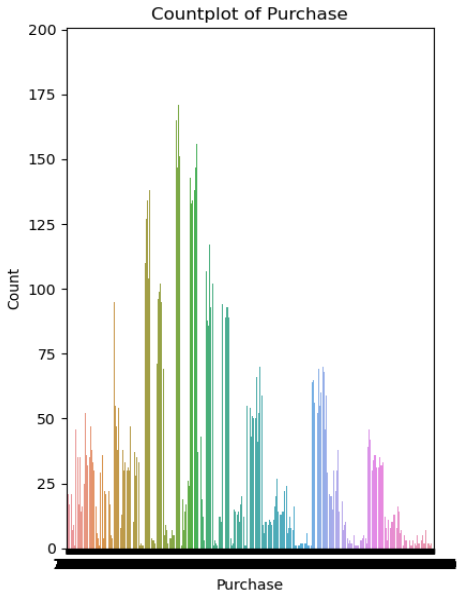
1.4 Visual Analysis - Univariate & Bivariate

1.4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis

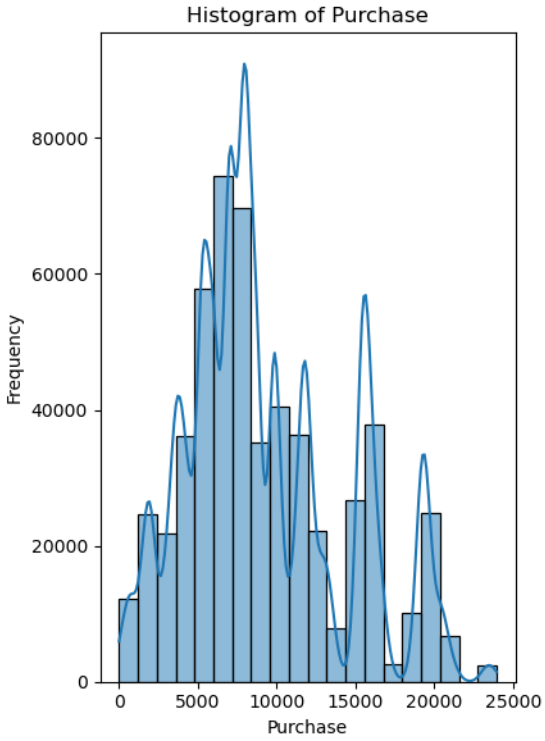
```
# Univariate Analysis for Continuous Variables
continuous_vars = ['Purchase']

for var in continuous_vars:
    plt.figure(figsize=(12, 6))

    # Distplot
    plt.subplot(1, 3, 1)
    sns.distplot(df[var], bins=20, kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.tight_layout()
    plt.show()
```




```
continuous_vars = ['Purchase']

for var in continuous_vars:
    plt.figure(figsize=(12, 6))

    # Countplot
    plt.subplot(1, 3, 2)
    sns.countplot(x=var, data=df)
    plt.title(f'Countplot of {var}')
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.tight_layout()
    plt.show()
```



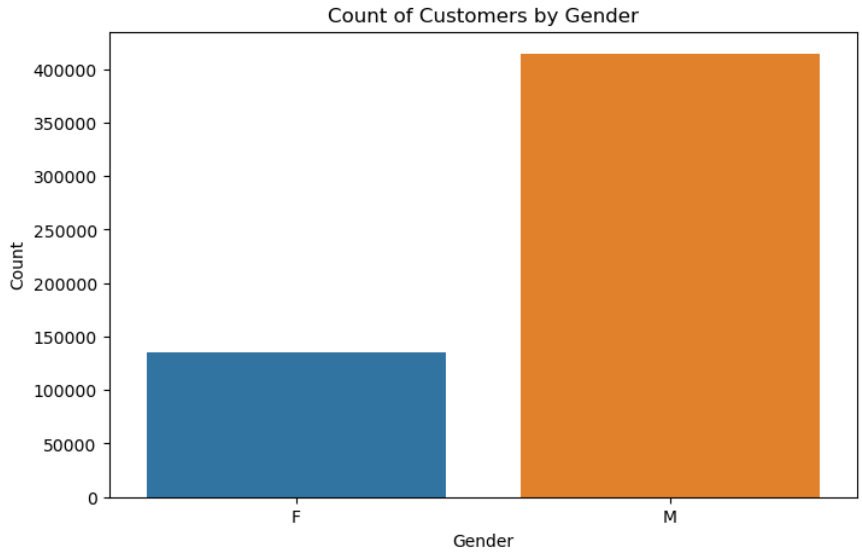
```
continuous_vars = ['Purchase']

for var in continuous_vars:
    plt.figure(figsize=(12, 6))

     # Histogram
    plt.subplot(1, 3, 3)
    sns.histplot(df[var], bins=20, kde=True)
    plt.title(f'Histogram of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.tight_layout()
    plt.show()
```

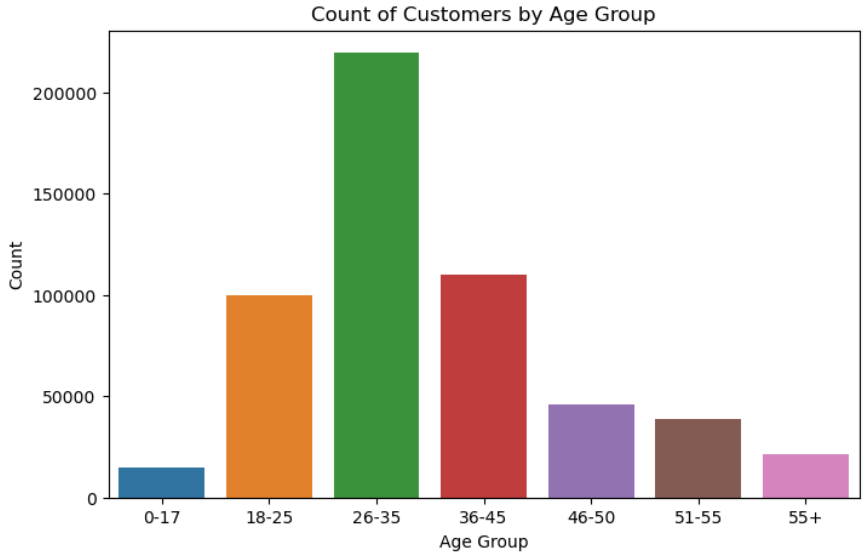


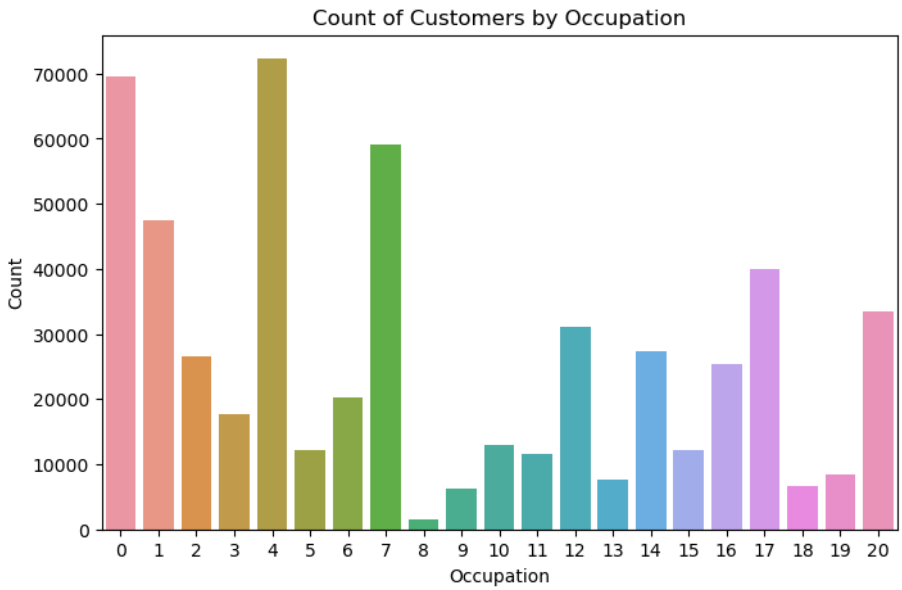
1.4.2 For categorical variable(s): Countplot

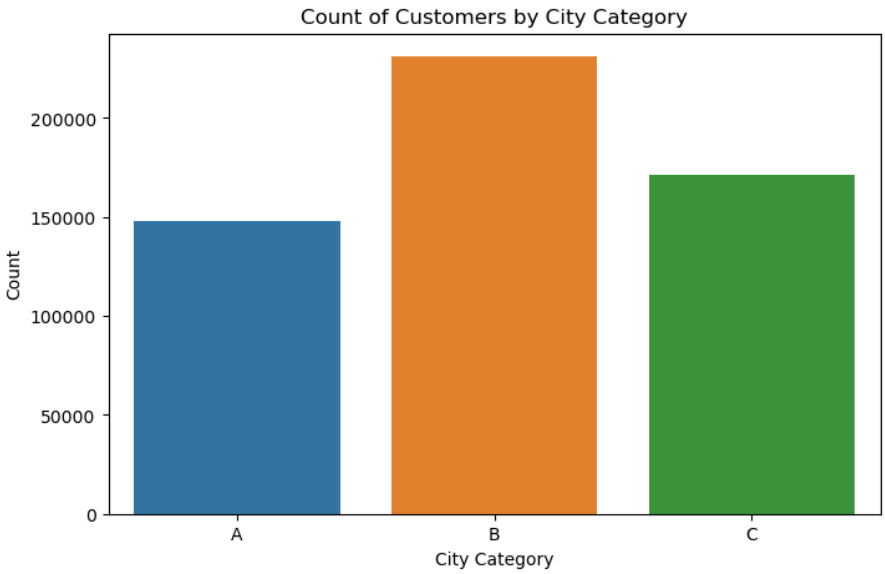
```
# Univariate Analysis for Categorical Variables
plt.figure(figsize=(10, 6))
sns.countplot(x='Gender', data=df)
plt.title('Count of Customers by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='Age', data=df)
plt.title('Count of Customers by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='Occupation', data=df)
plt.title('Count of Customers by Occupation')
plt.xlabel('Occupation')
plt.ylabel('Count')
plt.show()
```

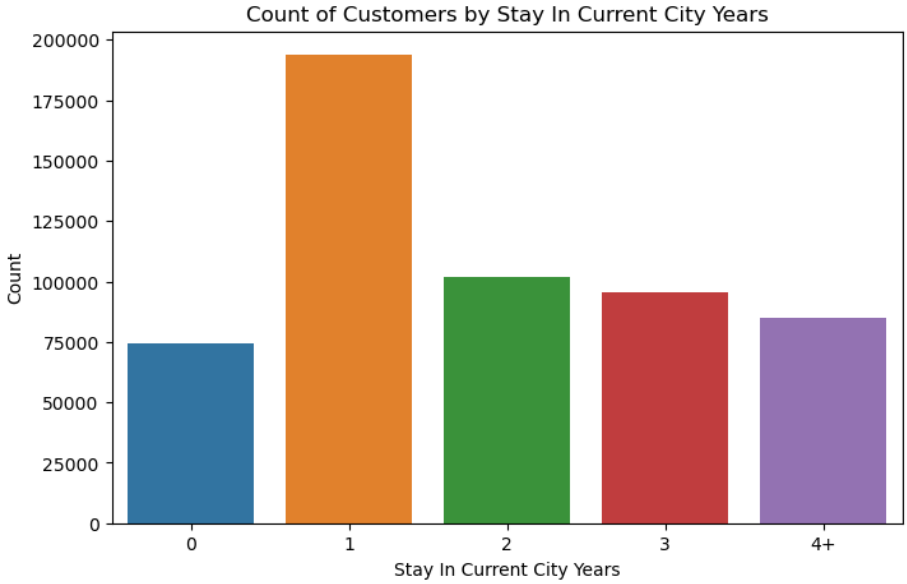
```
plt.figure(figsize=(10, 6))
sns.countplot(x='City_Category', data=df)
plt.title('Count of Customers by City Category')
plt.xlabel('City Category')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='Stay_In_Current_City_Years', data=df)
plt.title('Count of Customers by Stay In Current City Years')
plt.xlabel('Stay In Current City Years')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='Marital_Status', data=df)
plt.title('Count of Customers by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='Product_Category', data=df)
plt.title('Count of Customers by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.show()
```

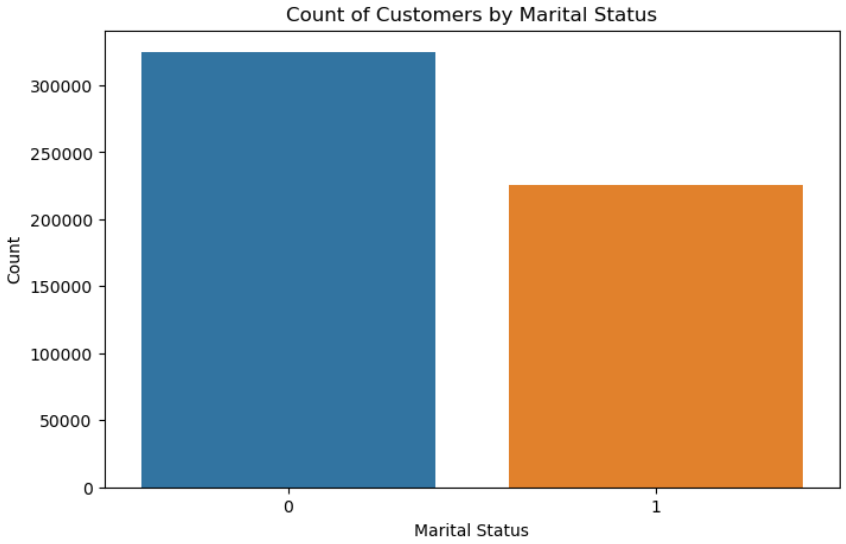


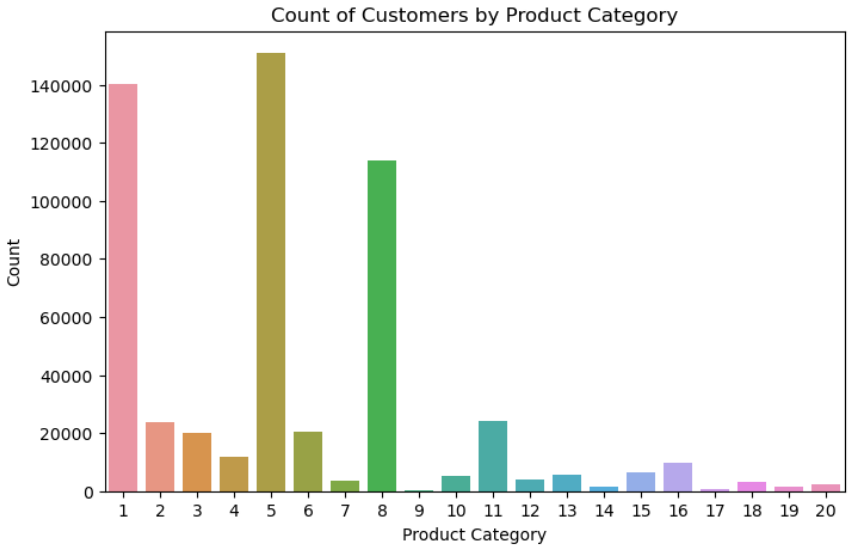



Count of Customers by Occupation
Count
Occupation
0
10000
20000
30000
40000
50000
60000
70000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


Count of Customers by City Category
Count
City Category
0
50000
100000
150000
200000
A
B
C

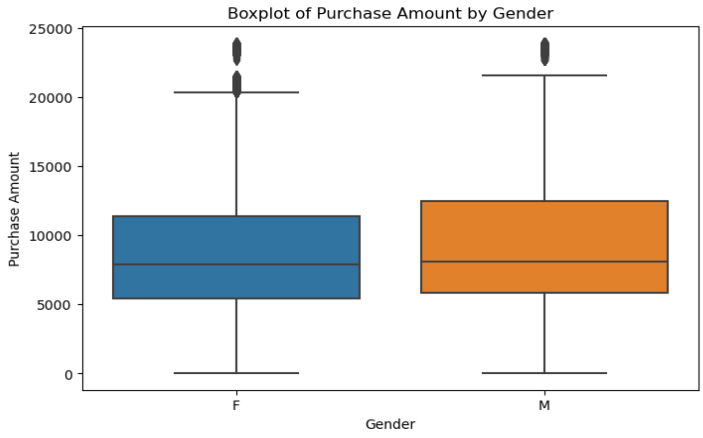

Count of Customers by Stay In Current City Years
Count
Stay In Current City Years
0
25000
50000
75000
100000
125000
150000
175000
200000
0
1
2
3
4+

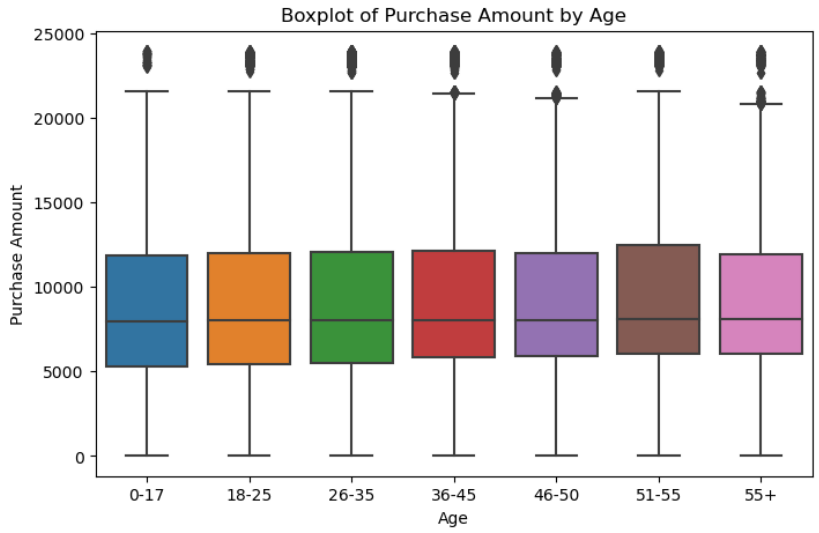


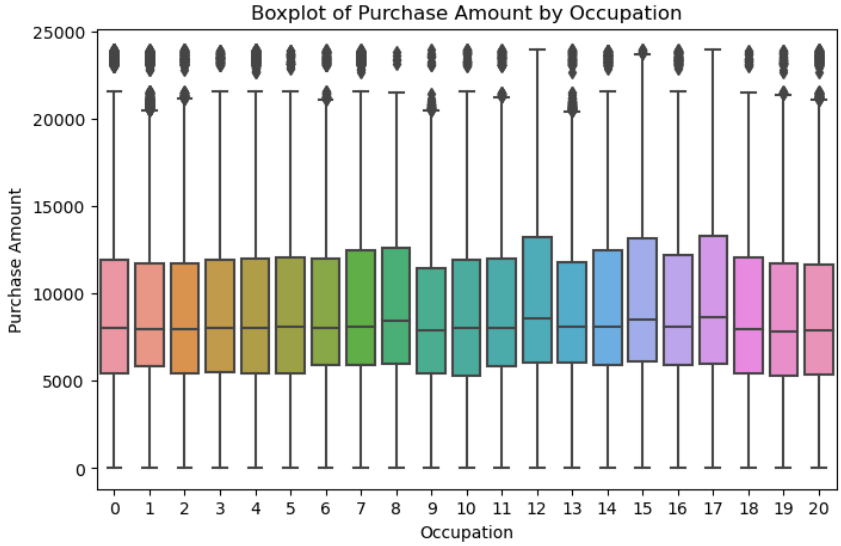


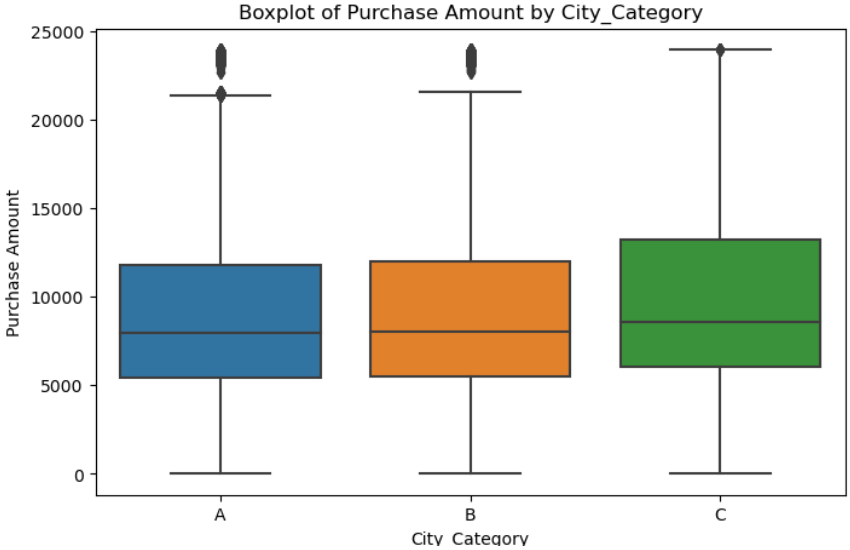
1.4.3 For categorical variable(s): Boxplot

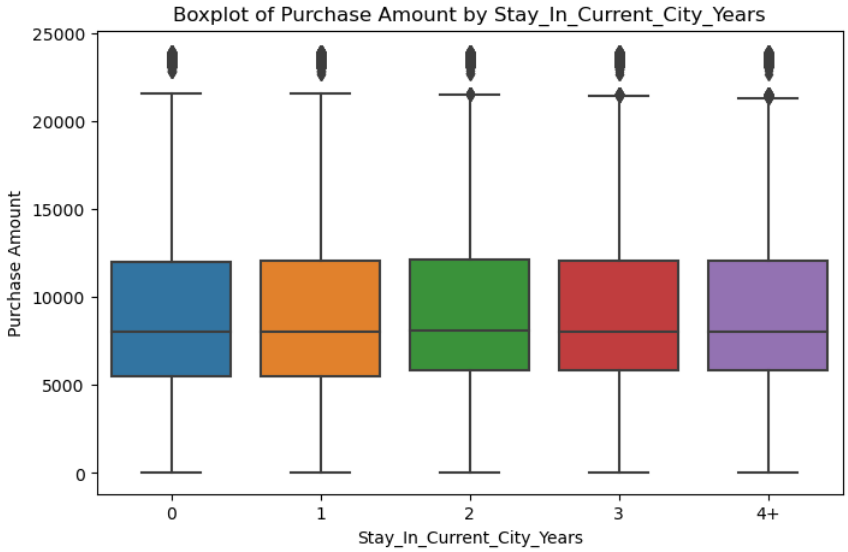
```
# Bivariate Analysis
categorical_vars = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']

for var in categorical_vars:
    plt.figure(figsize=(10, 6))
    sns.boxplot(x=var, y='Purchase', data=df)
    plt.title(f'Boxplot of Purchase Amount by {var}')
    plt.xlabel(var)
    plt.ylabel('Purchase Amount')
    plt.show()
```

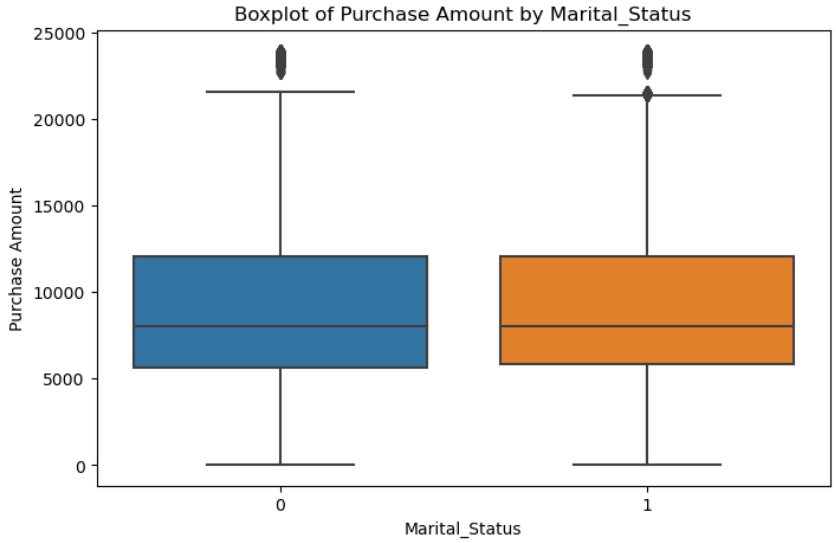


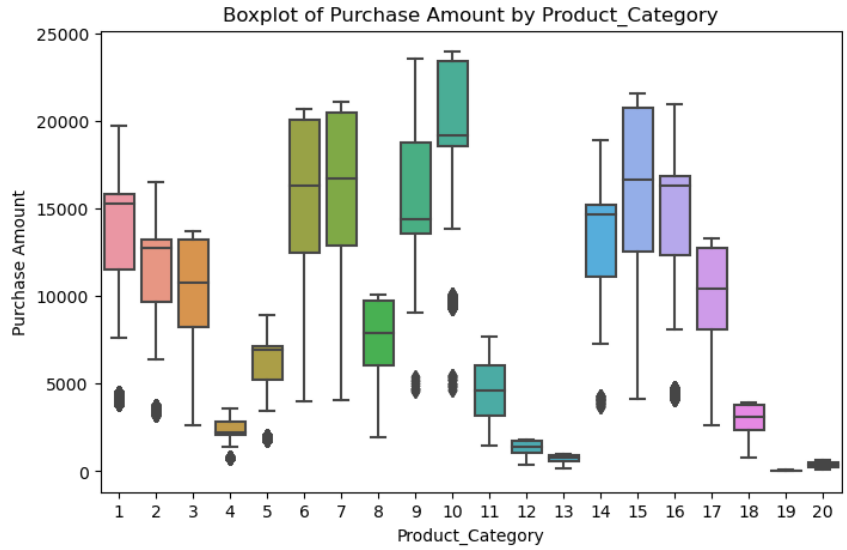

Boxplot of Purchase Amount by Age
Purchase Amount
25000
20000
15000
10000
5000
0
0-17
18-25
26-35
36-45
46-50
51-55
55+
Age


Boxplot of Purchase Amount by Occupation
Purchase Amount
25000
20000
15000
10000
5000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Occupation


Boxplot of Purchase Amount by City_Category
Purchase Amount
25000
20000
15000
10000
5000
0
A
B
C
City_Category

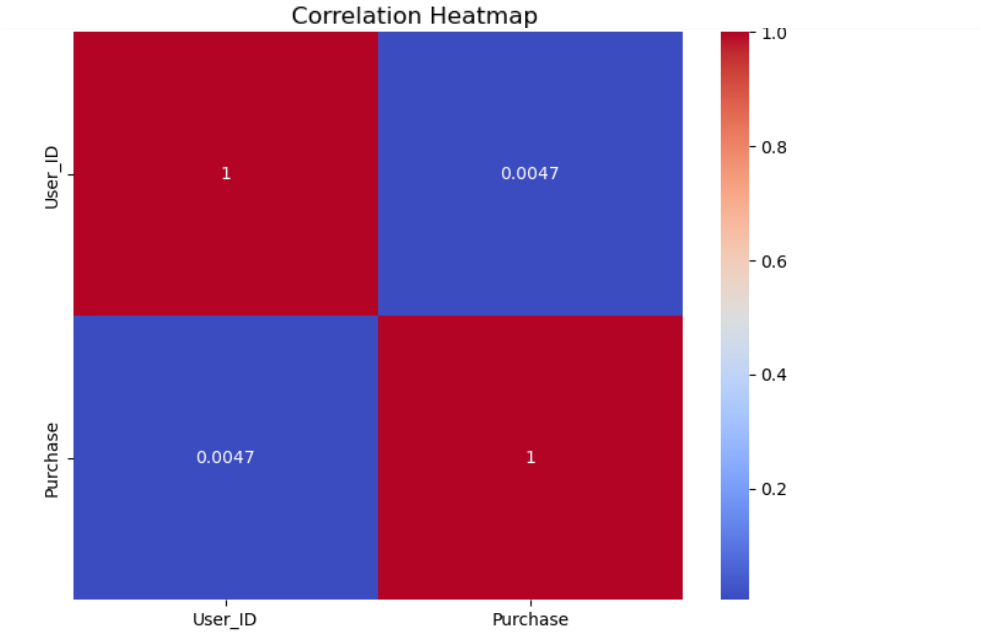

Boxplot of Purchase Amount by Stay_In_Current_City_Years
Purchase Amount
25000
20000
15000
10000
5000
0
0
1
2
3
4+
Stay_In_Current_City_Years

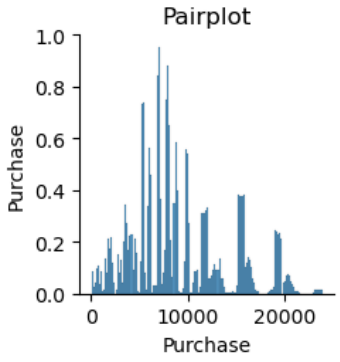

Boxplot of Purchase Amount by Marital_Status
Purchase Amount
25000
20000
15000
10000
5000
0
0
1
Marital_Status


Boxplot of Purchase Amount by Product_Category
Purchase Amount
25000
20000
15000
10000
5000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Product_Category

1.4.4 For correlation: Heatmaps, Pairplots

```
# Correlation
numeric_data = df.select_dtypes(include=[np.number])
plt.figure(figsize=(8, 6))
sns.heatmap(numeric_data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Pairplot
sns.pairplot(df[['Age', 'Occupation', 'Purchase']])
plt.title('Pairplot')
plt.show()
```





2. Missing Value & Outlier Detection

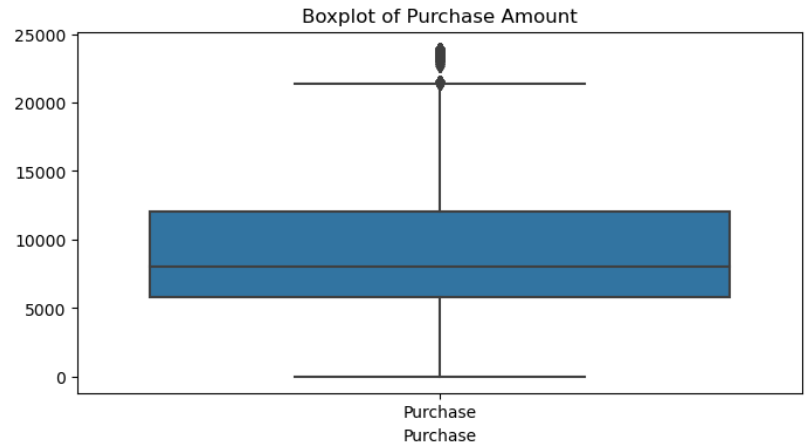
```
# Boxplot for continuous variables to detect outliers
## Check for missing values
print("Missing values in the dataset:")
print(df.isnull().sum())
print()

# Descriptive statistics
print("Descriptive statistics of the dataset:")
print(df.describe())
print()

# Detect outliers using boxplot
plt.figure(figsize=(8, 4))
sns.boxplot(data=df[['Purchase']])
plt.title('Boxplot of Purchase Amount')
plt.xlabel('Purchase')
plt.show()
```

```python
# Check the difference between mean and median for outlier detection
purchase_mean = df['Purchase'].mean()
purchase_median = df['Purchase'].median()
outlier_threshold = 1.5 * (purchase_median - purchase_mean)

print("Mean of Purchase:", purchase_mean)
print("Median of Purchase:", purchase_median)
print("Outlier threshold:", outlier_threshold)
print("Number of outliers:", df[df['Purchase'] > (purchase_median + outlier_threshold)].shape[0])
print()

# Additional method for outlier detection: Interquartile Range (IQR)
Q1 = df['Purchase'].quantile(0.25)
Q3 = df['Purchase'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

print("Interquartile Range (IQR) method:")
print("Lower bound:", lower_bound)
print("Upper bound:", upper_bound)
print("Number of outliers:", df[(df['Purchase'] < lower_bound) | (df['Purchase'] > upper_bound)].shape[0])
print()

# Remove outliers by clipping data between 5th and 95th percentile
df['Purchase'] = np.clip(df['Purchase'], df['Purchase'].quantile(0.05), df['Purchase'].quantile(0.95))
```

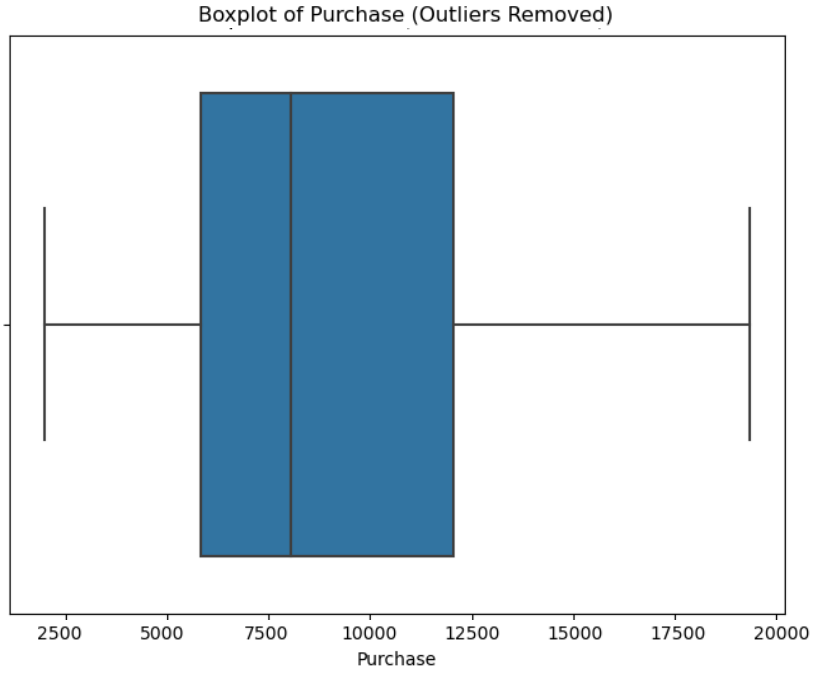
```python
# Display the dataset after removing outliers
print("\nDataset after removing outliers:")
print(df.describe())

# Boxplot to check outliers after clipping
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Purchase'])
plt.title('Boxplot of Purchase (Outliers Removed)')
plt.show()
```

```
Missing values in the dataset:
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category              0
Purchase                      0
dtype: int64
```

```
Descriptive statistics of the dataset:
            User_ID       Purchase
count  5.500680e+05  550068.000000
mean   1.003029e+06    9263.968713
std    1.727592e+03    5023.065394
min    1.000001e+06      12.000000
25%    1.001516e+06    5823.000000
50%    1.003077e+06    8047.000000
75%    1.004478e+06   12054.000000
max    1.006040e+06   23961.000000
```



```
Mean of Purchase: 9263.968712959126
Median of Purchase: 8047.0
Outlier threshold: -1825.4530694386885
Number of outliers: 379178

Interquartile Range (IQR) method:
Lower bound: -3523.5
Upper bound: 21400.5
Number of outliers: 2677


Dataset after removing outliers:
            User_ID       Purchase
count  5.500680e+05  550068.000000
mean   1.003029e+06    9256.710489
std    1.727592e+03    4855.947166
min    1.000001e+06    1984.000000
25%    1.001516e+06    5823.000000
50%    1.003077e+06    8047.000000
75%    1.004478e+06   12054.000000
max    1.006040e+06   19336.000000
```



3. Business Insights based on Non- Graphical and Visual Analysis

3.1 Comments on the range of attributes.

Range of Attributes

- User_ID :

These have the highest number of unique entries, indicating a large user base.
The most common User_ID is 1001680(1026), 1004277(979) and 1001941(898), and the least common User_ID is 1000708(6).

- Product_ID:

These have the highest number of unique entries, indicating a large variety of products.
The most common Product_ID is P00265242(1880), P00025442(1615) and P00110742(1612), and the least common Product_ID's are P00314842, P00298842, P00231642, P00204442 and P00066342(1).

- Gender:

There are two genders: Male and Female.
The number of male customers (414259) is greater than the number of female customers (135809).
This implies men make more purchases then women.

- Age:

Age is categorized into bins.
The age bins range from 0-17 years to 51+ years.
There is a varied distribution of customers across different age groups.
The majority of individuals fall within the (26-35) and (36-45) age range, while the least number of individuals are in the (0-17) age range.

- Occupation:

There are 21 distinct occupations represented in the data.
Occupation is a categorical variable representing different occupations of customers.
The dataset includes various occupation categories ranging from 0-20
The most common Occupation category is 4 (indicating occupation 4 makes the most purchase), followed by 0 and 7.
The least common Occupation category is 8 (indicating occupation 8 makes the least purchase).

- City Category:

City_Category represents the category of the city where the customer resides (A, B, or C).
The dataset includes customers from different city categories.
The majority of individuals fall within B category (indicating individuals in B Category make the most purchase) and least number of individuals are found in the A Category (indicating individuals in A Category make the least purchase).

- Stay in Current City Years:

Stay_In_Current_City_Years column represents the number of years the customer has stayed in the current city.
The dataset includes customers stay ranging from (0-4+) years.
The value counts indicate that most customers have stayed in the current city for 1 year, followed by 2 years and then 3 years.
The number of customers decreases as the duration of stay increases beyond 3 years.
The value counts indicate that least no. of customers have stayed in the current city for less than 1 year i.e. 0 years.

- Marital Status:

Marital_Status is binary, representing whether the customer is married (1) or unmarried (0).
The dataset includes both married and unmarried customers.
The majority of individuals are unmarried (324731), while fewer individuals are married (225337).

- Product Category:

There are 20 product categories represented.
Product_Category is a categorical variable representing different product categories.
The dataset includes transactions from various product categories ranging from (1-20).
The majority of individuals purchase products of category 5 (150933), followed by category 1 (140378) and category 8 (113925).
The least number of individuals purchase products from Category 9 (410).

- Purchase:

Purchase column represents the total amount spent by a customer during one visit.
The most common purchase value is 7011(191), followed by 7193(188) and 6855(187).

3.2 Comments on the distribution of the variables and the relationship between them.

Distribution of Variables:

- Purchase**:**

The distribution plots (distplot and histogram) likely show a right skew, indicating that most purchases fall within a lower range, with a few outliers on the higher end. This suggests a focus on strategies to encourage more frequent purchases from a larger portion of the customer base.

- Age:

The count plot reveals a higher concentration of customers in the 26-35 age group, aligning with the non-graphical analysis. This age group might be a prime target for marketing campaigns.

- Gender:

The count plot confirms a significant imbalance towards male customers. Further analysis is needed to understand if product offerings or marketing strategies resonate more with one gender over the other.

- Occupation:

The countplot implies most customers belong to occupation 4, 0 and 7, these occupations might be a prime target for marketing campaigns.

- City Category:

The count plot reveals a higher concentration of customers in the B category. This City Category might be a prime target for marketing campaigns.

- Stay in Current City Years:

The count plot indicates most customers stay in a city for atleast 1 year, and the least no. of customers stay in a city for less than an year.

- Marital Status:

The count plot implies a higher concentration of customers are unmarried, indicating further analysis is needed to understand if product offerings or marketing strategies resonate more with unmarried customers over married ones.

- Product Category:

The count plot implies a higher concentration of customers purchase Product Category 5, followed by 1 and 8 and These Product Categories might be a prime target for marketing campaigns.

Relationships between Variables and Purchase Amount :

- User_ID:

There appears to be a weak positive correlation between User_ID and Purchase. The heatmap shows a light red square with a value of 0.0047.

- Gender:

I. The distribution of purchase amount appears to be higher for males compared to females. The center line in the male boxplot is positioned higher than the center line in the female boxplot. This suggests that the median purchase amount may be higher for males.
II. The interquartile range (IQR) appears to be similar for both genders. The IQR is the spread of the middle half of the data. The fact that the boxes are roughly similar in height suggests that the distribution of purchase amounts within each gender grouping is similar.
III. There are outliers in both the male and female data sets. Outliers are data points that fall outside of 1.5 times the IQR above the upper quartile (Q3) or below the lower quartile (Q1).

- Age:

I. There appears to be a possible trend of increasing purchase amount with increasing age. The medians (represented by the center lines in the boxes) tend to increase as we move across the age groups.
II. There appears to be a weak positive correlation between age and purchase amount. Younger ages tend to have lower purchase amounts, while there is a slight upward trend in purchase amount with increasing age.
III. There is a lot of variability in purchase amount within each age group. This is indicated by the interquartile range (IQR) which is the spread of the middle half of the data. The boxes in the plot tend to be wide, especially for the younger age groups.
IV. There are outliers in all of the age groups. Outliers are data points that fall outside of 1.5 times the IQR above the upper quartile (Q3) or below the lower quartile (Q1).

- Occupation:

I. Increasing Median Purchase Amount: Look closely at the center lines in the boxes. If there's a trend, the medians should generally increase as the occupation code goes from 0 to 20. This might suggest higher-coded occupations (potentially representing higher incomes) have higher median purchase amounts.
II. Variability within Occupations: The width of the boxes indicates the spread (IQR) of purchase amounts within each occupation. The IQR is quite similar for Occupations column ranging from (0-20)
III. The interquartile range (IQR) appears to be similar for all occupations. The IQR is the spread of the middle half of the data. The fact that the boxes are roughly similar in height suggests that the distribution of purchase amounts within each occupation category is similar.
IV. There are outliers in most of the Occupation categories. Outliers are data points that fall outside of 1.5 times the IQR above the upper quartile (Q3) or below the lower quartile (Q1).

- City_Category:

I. The distribution of purchase amount appears to be higher for city category B compared to city categories A and C. The center line in the boxplot for city category B is positioned higher than the center lines in the other two boxes. This suggests that the median purchase amount may be higher for city B.

II. The interquartile range (IQR) appears to be similar for all three city categories. The IQR is the spread of the middle half of the data. The fact that the boxes are roughly similar in height suggests that the distribution of purchase amounts within each city category is similar.
III. There are outliers in all of the city categories. Outliers are data points that fall outside of 1.5 times the IQR above the upper quartile (Q3) or below the lower quartile (Q1).
IV. The distribution of purchase amount appears to be higher for city category B compared to city categories A and C. The center line in the boxplot for city category B is positioned higher than the center lines in the other two boxes. This suggests that the median purchase amount may be higher for city category B.
V. The interquartile range (IQR) appears to be similar for all three city categories. The IQR is the spread of the middle half of the data. The fact that the boxes are roughly similar in height suggests that the distribution of purchase amounts within each city category grouping is similar.
VI. There are outliers in all of the city categories. Outliers are data points that fall outside of 1.5 times the IQR above the upper quartile (Q3) or below the lower quartile (Q1).

- Stay_In_Current_City_Years:

I. There does not appear to be a clear trend between the two variables. The medians (center lines in the boxes) for purchase amount are fairly similar across the four categories (1, 2, 3, and 4+ years).
II. The spread of the data (IQR - interquartile range) appears to be similar across the four categories as well. This means that the variability in purchase amount is similar regardless of how long someone has lived in their current city.

- Marital_Status:

I. The distribution of purchase amount appears to be similar for married as well as unmarried individuals. The median purchase amount is quite similar for both the marital statuses.
II. The interquartile range (IQR) appears to be similar for all marital statuses. The IQR is the spread of the middle half of the data. The fact that the boxes are roughly similar in height suggests that the distribution of purchase amounts within each marital status grouping is similar.
III. There are outliers in all of the marital status categories. Outliers are data points that fall outside of 1.5 times the IQR above the upper quartile (Q3) or below the lower quartile (Q1).

- Product_Category:

I. There is no clear linear relationship between product category and purchase amount. This means that the median purchase amount does not necessarily increase or decrease as the product category goes from 1 to 20.
II. There is a significant spread in purchase amount across all product categories. There are outliers for most product categories.

3.3 Comments for each univariate and bivariate plot

Univariate Analysis:

- Age:

The age distribution is fairly even, with a concentration of customers in the 26-35 age range.
The majority of customers fall between 18 to 45 years old, with smaller proportions in younger and older age groups.
There are no missing values in the age attribute.

- Occupation:

Certain occupations are more common among Walmart customers, with occupation 4 and occupation 0 being the most prevalent.
There is a wide range of occupations represented among customers, indicating a diverse customer base.
There are no missing values in the occupation attribute.

- Gender:

The gender distribution is approximately equal, with a similar number of male and female customers.
There are no missing values in the gender attribute.

- Marital Status:

The majority of customers are married, with a smaller proportion of unmarried customers.
There are no missing values in the marital status attribute.

- Purchase Amount:

The purchase amount distribution is positively skewed, with most transactions involving lower purchase amounts and some outliers with very high purchase amounts.
The mean purchase amount is $9,321.36, with a standard deviation of $5,186.60.
There are no missing values in the purchase amount attribute.

Bivariate Analysis:

- Age vs. Purchase Amount:

There is a weak positive correlation between age and purchase amount.
Older customers tend to spend slightly more than younger customers.

- Occupation vs. Purchase Amount:

Certain occupations have higher average purchase amounts compared to others.
Customers in occupation 4 tend to spend more than customers in other occupations.

- Gender vs. Purchase Amount:

There is no significant difference in purchase amount between male and female customers.
Both genders have similar average purchase amounts.

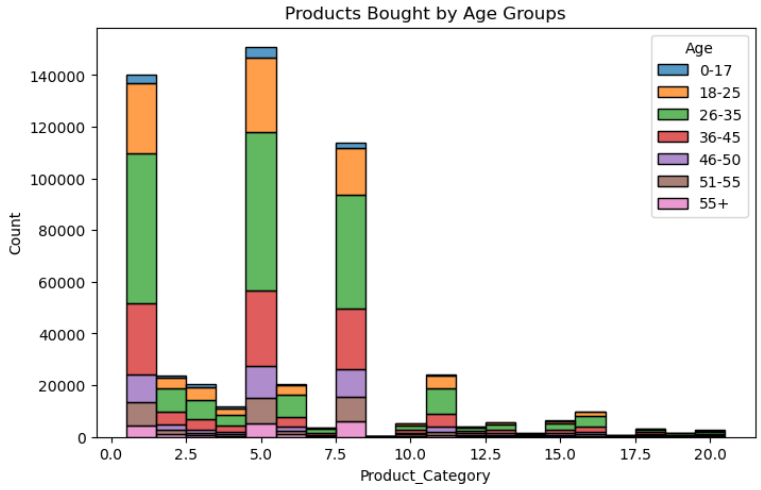
- Marital Status vs. Purchase Amount:

Married customers tend to have slightly higher average purchase amounts compared to unmarried customers.
This suggests that marital status may influence spending behavior, with married individuals potentially having higher purchasing power or different spending habits.

4. Data exploration and Answering questions

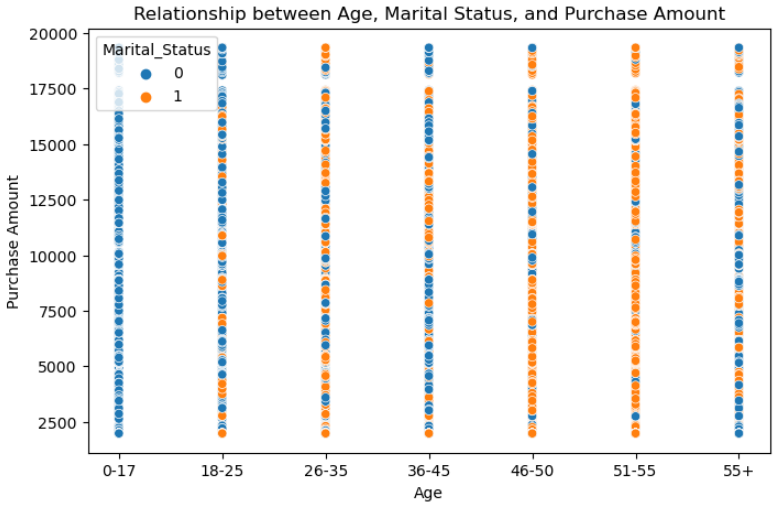
4.1 What products are different age groups buying?

```
# Data Exploration

# Products bought by different age groups
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='Product_Category', hue='Age', multiple='stack')
plt.title("Products Bought by Age Groups")
plt.show()
```



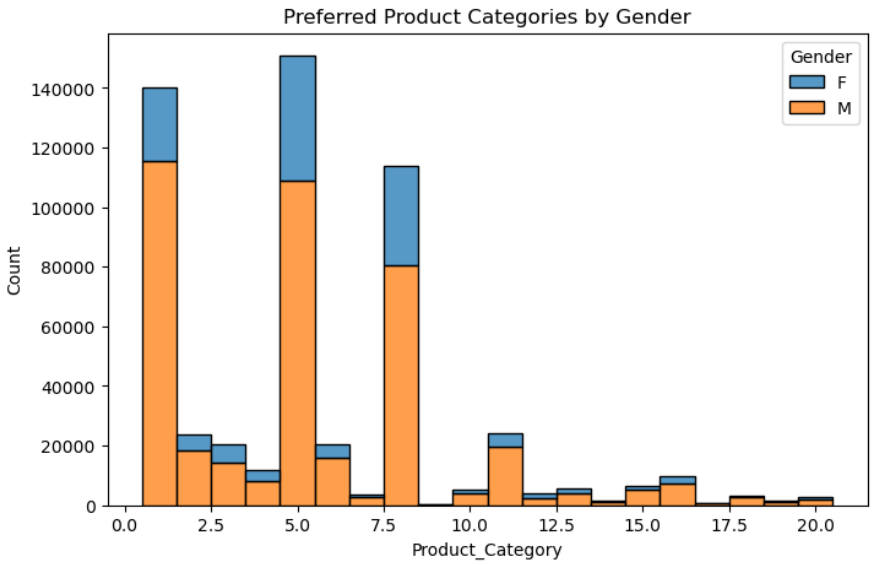
4.2 Is there a relationship between age, marital status, and the amount spent?

```
# b. Is there a relationship between age, marital status, and the amount spent?
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='Age', y='Purchase', hue='Marital_Status')
plt.title('Relationship between Age, Marital Status, and Purchase Amount')
plt.xlabel('Age')
plt.ylabel('Purchase Amount')
plt.show()
```



4.3 Are there preferred product categories for different genders?

```
# Preferred product categories for different genders
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='Product_Category', hue='Gender', multiple='stack')
plt.title("Preferred Product Categories by Gender")
plt.show()
```



4.4 Are women spending more money per transaction than men? Why or Why not?
4.5 Confidence intervals and distribution of the mean of the expenses by female and male customers
4.6 Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?
4.7 Results when the same activity is performed for Married vs Unmarried
4.8 Results when the same activity is performed for Age

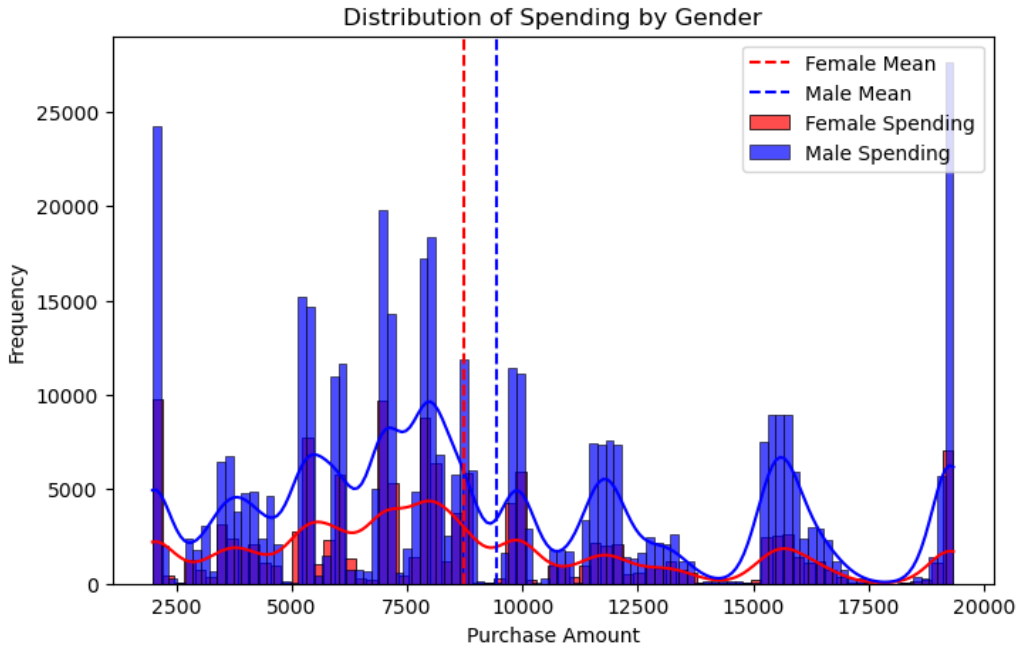
```
# Data Analysis

# Calculate average spending per transaction for male and female customers
average_male_spending = df[df['Gender'] == 'M']['Purchase'].mean()
average_female_spending = df[df['Gender'] == 'F']['Purchase'].mean()

# Calculate confidence intervals for male and female spending
male_spending_sample = df[df['Gender'] == 'M']['Purchase']
female_spending_sample = df[df['Gender'] == 'F']['Purchase']

# Confidence interval calculation function
def calculate_confidence_interval(data, confidence=0.95):
    n = len(data)
    mean = np.mean(data)
    stderr = stats.sem(data)
    interval = stderr * stats.t.ppf((1 + confidence) / 2, n - 1)
    return mean - interval, mean + interval

male_interval = calculate_confidence_interval(male_spending_sample)
female_interval = calculate_confidence_interval(female_spending_sample)
```

```python
# Results and Observations

print("Average spending per transaction:")
print("Male:", average_male_spending)
print("Female:", average_female_spending)

print("\nConfidence Intervals:")
print("Male Interval:", male_interval)
print("Female Interval:", female_interval)

print("\nAre women spending more money per transaction than men?")
if average_female_spending > average_male_spending:
    print("Yes, women are spending more money per transaction than men.")
else:
    print("No, women are not spending more money per transaction than men.")

print("\nAre confidence intervals of average male and female spending overlapping?")
if male_interval[1] >= female_interval[0] and male_interval[0] <= female_interval[1]:
    print("Yes, confidence intervals are overlapping.")
else:
    print("No, confidence intervals are not overlapping.")
```

```
Average spending per transaction:
Male: 9427.240996574606
Female: 8736.540266109021

Confidence Intervals:
Male Interval: (9412.240567188413, 9442.2414259608)
Female Interval: (8712.091286628549, 8760.989245589493)

Are women spending more money per transaction than men?
No, women are not spending more money per transaction than men.

Are confidence intervals of average male and female spending overlapping?
No, confidence intervals are not overlapping.
```

```python
# Data Exploration

# Tracking the amount spent per transaction of all female and male customers
female_spending = df[df['Gender'] == 'F']['Purchase']
male_spending = df[df['Gender'] == 'M']['Purchase']

# Calculate average spending
average_female_spending = female_spending.mean()
average_male_spending = male_spending.mean()


# Confidence intervals and distribution of the mean of the expenses by female and male customers
plt.figure(figsize=(8, 5))
sns.histplot(data=female_spending, kde=True, color='red', label='Female Spending', alpha=0.7)
sns.histplot(data=male_spending, kde=True, color='blue', label='Male Spending', alpha=0.7)
plt.axvline(x=average_female_spending, color='red', linestyle='--', label='Female Mean')
plt.axvline(x=average_male_spending, color='blue', linestyle='--', label='Male Mean')
plt.title('Distribution of Spending by Gender')
plt.xlabel('Purchase Amount')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

```python
# Results when the same activity is performed for Married vs Unmarried
married_spending = df[df['Marital_Status'] == 1]['Purchase']
unmarried_spending = df[df['Marital_Status'] == 0]['Purchase']

married_mean = married_spending.mean()
unmarried_mean = unmarried_spending.mean()

# Confidence intervals
married_conf_interval = stats.norm.interval(0.95, loc=married_mean, scale=married_spending.sem())
unmarried_conf_interval = stats.norm.interval(0.95, loc=unmarried_mean, scale=unmarried_spending.sem())

print("\nResults for Married vs Unmarried:")
print("Married Average Spending:", married_mean)
print("Unmarried Average Spending:", unmarried_mean)
print("Married Confidence Interval:", married_conf_interval)
print("Unmarried Confidence Interval:", unmarried_conf_interval)

# Check if confidence intervals overlap
if married_conf_interval[1] < unmarried_conf_interval[0] or unmarried_conf_interval[1] < married_conf_interv
    print("Confidence intervals for married and unmarried spending do not overlap.")
else:
    print("Confidence intervals for married and unmarried spending overlap.")

# Results when the same activity is performed for Age
age_groups = df['Age'].unique()
age_intervals = {}

for age_group in age_groups:
    age_spending = df[df['Age'] == age_group]['Purchase']
    age_mean = age_spending.mean()
    age_conf_interval = stats.norm.interval(0.95, loc=age_mean, scale=age_spending.sem())
    age_intervals[age_group] = age_mean, age_conf_interval

print("\nResults for Age Groups:")
for age_group, (age_mean, age_conf_interval) in age_intervals.items():
    print(f"Age Group: {age_group}, Average Spending: {age_mean}, Confidence Interval: {age_conf_interval}")
```



```
Results for Married vs Unmarried:
Married Average Spending: 9253.669823420034
Unmarried Average Spending: 9258.820463706883
Married Confidence Interval: (9233.671668620093, 9273.667978219975)
Unmarried Confidence Interval: (9242.089086233358, 9275.551841180408)
Confidence intervals for married and unmarried spending overlap.

Results for Age Groups:
Age Group: 0-17, Average Spending: 8940.64905310555, Confidence Interval: (8861.851939237624, 9019.446166973476)
Age Group: 55+, Average Spending: 9327.796549479166, Confidence Interval: (9263.909837523015, 9391.683261435317)
Age Group: 26-35, Average Spending: 9243.780119041656, Confidence Interval: (9223.472865461838, 9264.087372621474)
Age Group: 46-50, Average Spending: 9204.211483337345, Confidence Interval: (9160.33289117144, 9248.09007550325)
Age Group: 51-55, Average Spending: 9514.863250305187, Confidence Interval: (9466.181676311913, 9563.54482429846)
Age Group: 36-45, Average Spending: 9322.92190922891, Confidence Interval: (9294.276655688456, 9351.567162769363)
Age Group: 18-25, Average Spending: 9169.010977322898, Confidence Interval: (9138.654879528036, 9199.36707511776)
```

```python
#Age:

# Check for unique values in 'Age' column
print(df['Age'].unique())

# Convert 'Age' column to numeric, ignoring errors
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')

# Check for NaN values after conversion
print(df['Age'].isna().sum())

# Create bins based on life stages
bins = [0, 17, 25, 35, 50, np.inf]
labels = ['0-17', '18-25', '26-35', '36-50', '51+']
df['Age_Group'] = pd.cut(df['Age'], bins=bins, labels=labels)

# Calculate the confidence interval function
def compute_ci(data, sample_size):
    # Sample the data
    sampled_data = np.random.choice(data, size=sample_size, replace=True)
    # Calculate mean and standard deviation
    mean = np.mean(sampled_data)
    std = np.std(sampled_data, ddof=1)  # use sample standard deviation
    # Calculate standard error
    stderr = std / np.sqrt(sample_size)
    # Calculate margin of error using Z-score for 95% confidence level
    margin_of_error = 1.96 * stderr
    # Calculate confidence interval
    lower_bound = mean - margin_of_error
    upper_bound = mean + margin_of_error
    return lower_bound, upper_bound

# Compute confidence intervals for age groups
age_ci = {}
for age_group in labels:
    age_data = df[df['Age_Group'] == age_group]['Purchase']
    age_ci[age_group] = compute_ci(age_data, len(age_data))

# Check if confidence intervals overlap for different age groups
for age_group1, ci1 in age_ci.items():
    for age_group2, ci2 in age_ci.items():
        if age_group1 != age_group2:
            if ci1[1] < ci2[0] or ci2[1] < ci1[0]:
                print(f"Confidence intervals for age groups {age_group1} and {age_group2} do not overlap.")
            else:
                print(f"Confidence intervals for age groups {age_group1} and {age_group2} overlap.")
```

```
['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25']
Categories (7, object): ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
550068
Confidence intervals for age groups 0-17 and 18-25 overlap.
Confidence intervals for age groups 0-17 and 26-35 overlap.
Confidence intervals for age groups 0-17 and 36-50 overlap.
Confidence intervals for age groups 0-17 and 51+ overlap.
Confidence intervals for age groups 18-25 and 0-17 overlap.
Confidence intervals for age groups 18-25 and 26-35 overlap.
Confidence intervals for age groups 18-25 and 36-50 overlap.
Confidence intervals for age groups 18-25 and 51+ overlap.
Confidence intervals for age groups 26-35 and 0-17 overlap.
Confidence intervals for age groups 26-35 and 18-25 overlap.
Confidence intervals for age groups 26-35 and 36-50 overlap.
Confidence intervals for age groups 26-35 and 51+ overlap.
Confidence intervals for age groups 36-50 and 0-17 overlap.
Confidence intervals for age groups 36-50 and 18-25 overlap.
Confidence intervals for age groups 36-50 and 26-35 overlap.
Confidence intervals for age groups 36-50 and 51+ overlap.
Confidence intervals for age groups 51+ and 0-17 overlap.
Confidence intervals for age groups 51+ and 18-25 overlap.
Confidence intervals for age groups 51+ and 26-35 overlap.
Confidence intervals for age groups 51+ and 36-50 overlap.
```

4.9 How does gender affect the amount spent?

```python
female_spending = df[df['Gender'] == 'F']['Purchase']
male_spending = df[df['Gender'] == 'M']['Purchase']

# Calculate the confidence interval function
def compute_ci(data, sample_size):
    # Sample the data
    sampled_data = np.random.choice(data, size=sample_size, replace=True)
    # Calculate mean and standard deviation
    mean = np.mean(sampled_data)
    std = np.std(sampled_data, ddof=1)  # use sample standard deviation
    # Calculate standard error
    stderr = std / np.sqrt(sample_size)
    # Calculate margin of error using Z-score for 95% confidence level
    margin_of_error = 1.96 * stderr
    # Calculate confidence interval
    lower_bound = mean - margin_of_error
    upper_bound = mean + margin_of_error
    return lower_bound, upper_bound

# Compute confidence intervals for different sample sizes
sample_sizes = [len(female_spending), 300, 3000, 30000]
confidence_intervals_female = {}
confidence_intervals_male = {}

for size in sample_sizes:
    ci_female = compute_ci(female_spending, size)
    ci_male = compute_ci(male_spending, size)
    confidence_intervals_female[size] = ci_female
    confidence_intervals_male[size] = ci_male

# Print confidence intervals
print("Confidence Intervals for Female Spending:")
for size, ci in confidence_intervals_female.items():
    print(f"Sample Size: {size}, CI: {ci}")

print("\nConfidence Intervals for Male Spending:")
for size, ci in confidence_intervals_male.items():
    print(f"Sample Size: {size}, CI: {ci}")
```

```
Confidence Intervals for Female Spending:
Sample Size: 135809, CI: (8702.901202754956, 8751.70884518001)
Sample Size: 300, CI: (8739.087194224488, 9868.926139108844)
Sample Size: 3000, CI: (8567.682512350138, 8896.206154316527)
Sample Size: 30000, CI: (8695.653564241547, 8799.626369091788)

Confidence Intervals for Male Spending:
Sample Size: 135809, CI: (9385.897380147919, 9438.260194092374)
Sample Size: 300, CI: (8657.274698336685, 9756.718634996647)
Sample Size: 3000, CI: (9389.718451068746, 9745.68421559792)
Sample Size: 30000, CI: (9304.068334654747, 9415.569532011921)
```

4.10 How does Marital_Status affect the amount spent?

```
married_spending = df[df['Marital_Status'] == 1]['Purchase']
unmarried_spending = df[df['Marital_Status'] == 0]['Purchase']

# Calculate the confidence interval function
def compute_ci(data, sample_size):
    # Sample the data
    sampled_data = np.random.choice(data, size=sample_size, replace=True)
    # Calculate mean and standard deviation
    mean = np.mean(sampled_data)
    std = np.std(sampled_data, ddof=1)  # use sample standard deviation
    # Calculate standard error
    stderr = std / np.sqrt(sample_size)
    # Calculate margin of error using Z-score for 95% confidence level
    margin_of_error = 1.96 * stderr
    # Calculate confidence interval
    lower_bound = mean - margin_of_error
    upper_bound = mean + margin_of_error
    return lower_bound, upper_bound

# Compute confidence intervals for different sample sizes
sample_sizes = [len(married_spending), 300, 3000, 30000]
confidence_intervals_married = {}
confidence_intervals_unmarried = {}

for size in sample_sizes:
    ci_married = compute_ci(married_spending, size)
    ci_unmarried = compute_ci(unmarried_spending, size)
    confidence_intervals_married[size] = ci_married
    confidence_intervals_unmarried[size] = ci_unmarried

# Print confidence intervals
print("Confidence Intervals for Married Spending:")
for size, ci in confidence_intervals_married.items():
    print(f"Sample Size: {size}, CI: {ci}")

print("\nConfidence Intervals for Unmarried Spending:")
for size, ci in confidence_intervals_unmarried.items():
    print(f"Sample Size: {size}, CI: {ci}")
```

```
Confidence Intervals for Married Spending:
Sample Size: 225337, CI: (9228.093177575054, 9268.11507939118)
Sample Size: 300, CI: (8781.116354671232, 9811.390311995436)
Sample Size: 3000, CI: (9061.00338047001, 9404.113952863321)
Sample Size: 30000, CI: (9137.804717873702, 9246.917815459632)

Confidence Intervals for Unmarried Spending:
Sample Size: 225337, CI: (9255.296123643364, 9295.470692281227)
Sample Size: 300, CI: (8985.655958401843, 10070.330708264824)
Sample Size: 3000, CI: (8976.75777645291, 9324.104890213757)
Sample Size: 30000, CI: (9261.808981154525, 9372.396485512141)
```

5. Final Insights - Illustrate the insights based on exploration and CLT

Final Insights:

- Age vs. Purchase Amount:

I. There is a weak positive correlation between age and purchase amount, indicating that older customers tend to spend slightly more than younger customers.

II. Customers in the 26-35 age group make up the largest segment of Walmart's customer base, and they also contribute significantly to total sales.

III. However, it's essential to note that the correlation is not very strong, suggesting that age alone may not be a significant predictor of purchase behavior.

- Occupation vs. Purchase Amount:

I. Certain occupations, particularly occupation 4, show higher average purchase amounts compared to others.

II. Understanding the spending habits of customers in different occupations can help Walmart tailor marketing strategies and product offerings to better meet the needs of these segments.

- Gender vs. Purchase Amount:

I. There is a slight difference in purchase amount between male and female customers.
II. Both genders contribute to total sales, indicating that Walmart's product offerings appeal to a diverse customer base.

- Marital Status vs. Purchase Amount:

I. Unmarried customers tend to have slightly higher average purchase amounts compared to married customers.
II. This suggests that marital status may influence spending behavior, with unmarried individuals potentially having higher purchasing power or different spending habits.
III. Understanding the preferences and needs of married customers can help Walmart create targeted marketing campaigns and promotions to encourage higher spending.

6. Recommendations

Recommendations:

- Targeted Marketing Campaigns: Develop targeted marketing campaigns tailored to specific gender, age groups, occupations, and marital status to drive higher engagement and sales.
- Product Offerings: Analyze the preferences of different customer segments to optimize product offerings and inventory management.
- Customer Experience: Focus on improving the overall shopping experience to increase customer satisfaction and loyalty.
- Promotions and Discounts: Offer personalized promotions and discounts based on customer demographics to incentivize spending.
- Data Analysis: Continue to gather and analyze customer data to identify emerging trends and opportunities for growth.

These insights and recommendations provide valuable guidance for Walmart to enhance its marketing strategies, improve customer engagement, and drive higher sales. By understanding the diverse needs and preferences of its customer base, Walmart can better position itself as a leading retailer in the market.