Business Case: Walmart - Confidence Interval and CLT

By Parth Patel

Problem Statement

The Management team at Walmart Inc. wants to analyze the customer purchase behavior (specifically, purchase amount) against the customer's gender and the various other factors to help the business make better decisions. They want to understand if the spending habits differ between male and female customers: Do women spend more on Black Friday than men? (Assume 50 million customers are male and 50 million are female).

Importing all the libraries for analyzing the case study

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from scipy.stats import poisson
from scipy.stats import binom
import scipy.stats as stats
import math
```

```
df = pd.read_csv('walmart_data.csv')
df
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Cu
0	1000001	P00069042	F	0-17	10	A	
1	1000001	P00248942	F	0-17	10	A	
2	1000001	P00087842	F	0-17	10	A	
3	1000001	P00085442	F	0-17	10	A	
4	1000002	P00285442	M	55+	16	C	
...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	
550064	1006035	P00375436	F	26-35	1	C	
550065	1006036	P00375436	F	26-35	15	B	
550066	1006038	P00375436	F	55+	1	C	
550067	1006039	P00371644	F	46-50	0	B	

550068 rows × 10 columns

```
df.shape
```

(550068, 10)

Walmart dataset contains 550068 Rows and 10 Columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype 
---  ------                      --------------   ----- 
 0   User_ID                     550068 non-null  int64 
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64 
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64 
 8   Product_Category            550068 non-null  int64 
 9   Purchase                    550068 non-null  int64 
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

```
df.isna().sum()
```

```
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
```

The dataset mentioned above does not contain any null values or missing data.

```
columns = ['Occupation', 'Marital_Status', 'Product_Category']
df[columns] = df[columns].astype('object')
print(df.dtypes)
```

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

```
description = df.describe(include="all")
print(description)
```

```
             User_ID Product_ID  Gender     Age  Occupation City_Category  \
count   5.500680e+05     550068  550068  550068    550068.0        550068
unique           NaN       3631       2       7        21.0             3
top              NaN  P00265242       M   26-35         4.0             B
freq             NaN       1880  414259  219587     72308.0        231173
mean    1.003029e+06        NaN     NaN     NaN         NaN           NaN
std     1.727592e+03        NaN     NaN     NaN         NaN           NaN
min     1.000001e+06        NaN     NaN     NaN         NaN           NaN
25%     1.001516e+06        NaN     NaN     NaN         NaN           NaN
50%     1.003077e+06        NaN     NaN     NaN         NaN           NaN
75%     1.004478e+06        NaN     NaN     NaN         NaN           NaN
max     1.006040e+06        NaN     NaN     NaN         NaN           NaN

       Stay_In_Current_City_Years  Marital_Status  Product_Category  \
count                      550068        550068.0          550068.0
unique                          5             2.0              20.0
top                             1             0.0               5.0
freq                       193821        324731.0          150933.0
mean                          NaN             NaN               NaN
std                           NaN             NaN               NaN
min                           NaN             NaN               NaN
25%                           NaN             NaN               NaN
50%                           NaN             NaN               NaN
75%                           NaN             NaN               NaN
max                           NaN             NaN               NaN

             Purchase
count   550068.000000
unique            NaN
top               NaN
freq              NaN
mean      9263.968713
std       5023.065394
min         12.000000
25%       5823.000000
50%       8047.000000
75%      12054.000000
max      23961.000000
```

Observation:

1- The primary demographic making purchases falls within the 26–35 age bracket.

2- Male customers are the predominant purchasers.

3- With an average purchase amounting to 9263.96 and a maximum purchase of 23961, the mean appears sensitive to outliers. However, the considerable discrepancy between the mean and maximum value suggests that the maximum purchase may be an outlier.

Non-Graphical Analysis:

```
gender_counts = df['Gender'].value_counts()
percentage_gender_counts = (gender_counts / len(df)) * 100
print(f"Gender count : \n{gender_counts} \nGender percentage : \n{percentage_gend
```

```
Gender count :
M    414259
F    135809
Name: Gender, dtype: int64
Gender percentage :
M    75.310507
F    24.689493
Name: Gender, dtype: float64
```

```
Age_counts = df['Age'].value_counts()
percentage_Age_counts = (Age_counts / len(df)) * 100
print(f"Age count : \n{Age_counts} \nAge percentage : \n{percentage_Age_counts}")
```

```
Age count :
26-35    219587
36-45    110013
18-25     99660
46-50     45701
51-55     38501
55+       21504
0-17      15102
Name: Age, dtype: int64
Age percentage :
26-35    39.919974
36-45    19.999891
18-25    18.117760
46-50     8.308246
51-55     6.999316
55+       3.909335
0-17      2.745479
Name: Age, dtype: float64
```

Unique Attributes

```
unique_category_count = df['Product_Category'].nunique()
unique_category_count
```

```
20
```

```
unique_City_Category_count = df['City_Category'].nunique()
unique_City_Category_count
```

```
3
```

```
unique_Product_ID_count = df['Product_ID'].nunique()
unique_Product_ID_count
```

```
3631
```

```
unique_User_ID_count = df['User_ID'].nunique()
unique_User_ID_count
```

```
5891
```

Insights :

There are a total of 20 distinct product categories.

There are three unique city categories in total.

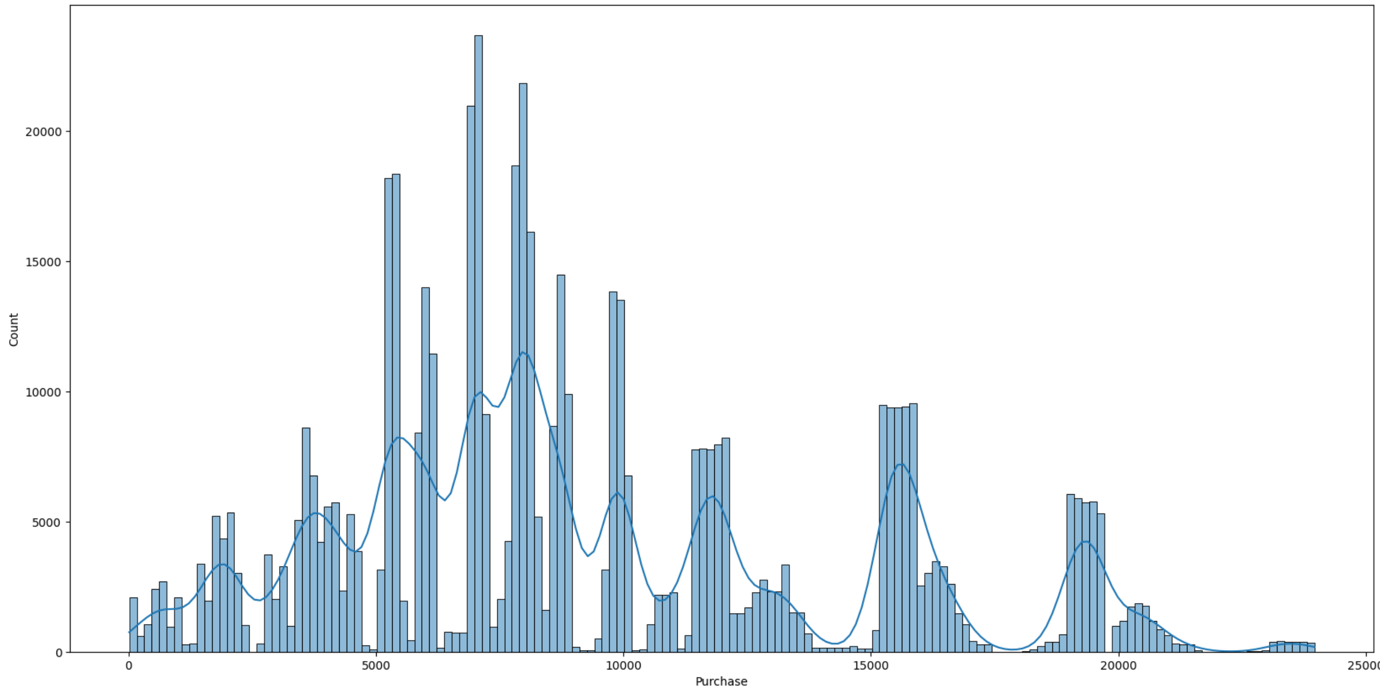
There are 3631 different product IDs in total.

The total count of unique user IDs is 5891.

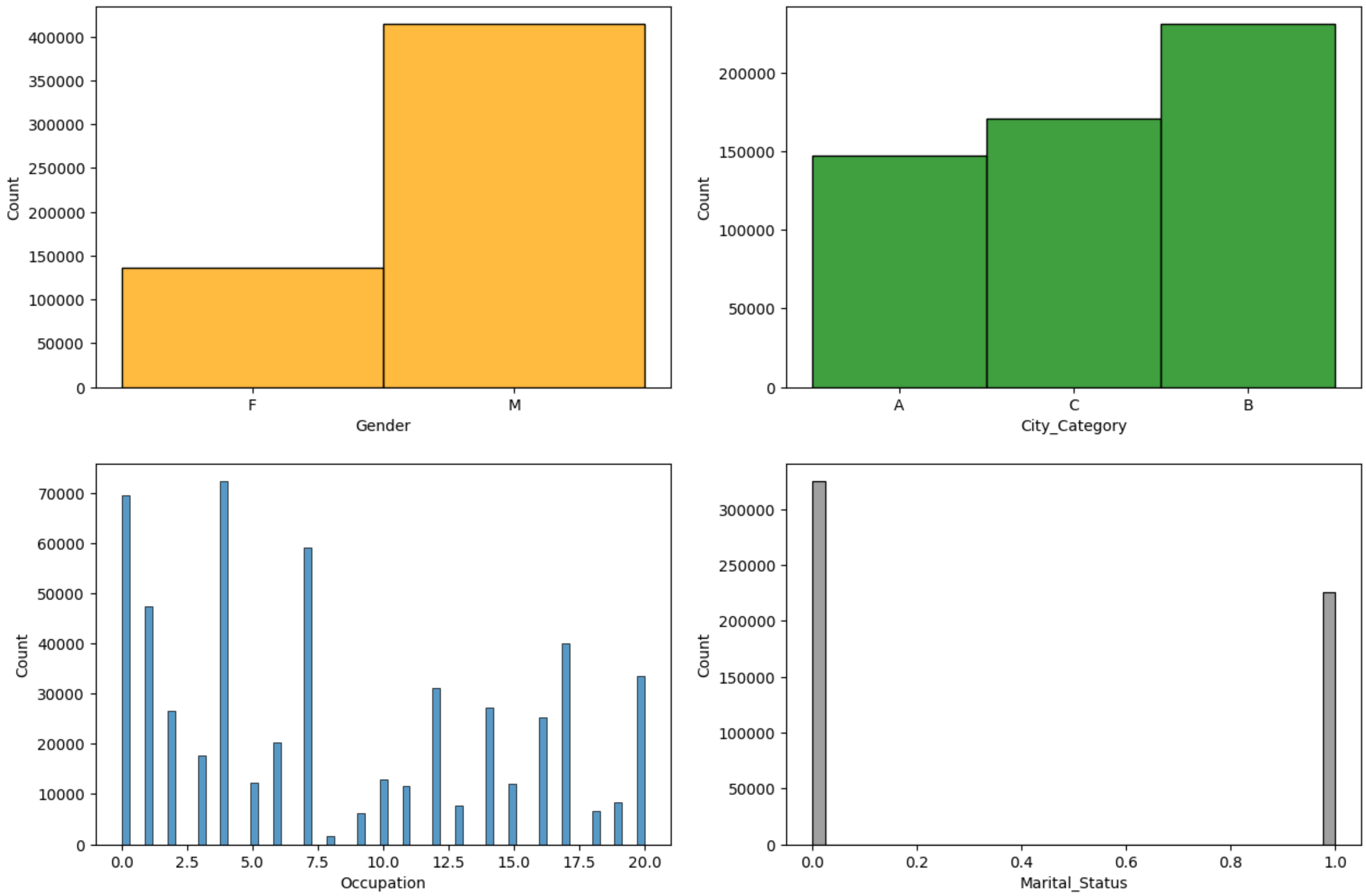
Visual Analysis - Univariate & Bivariate

Univariate

```
plt.figure(figsize=(20,10))
sns.histplot(data=df, x='Purchase', kde=True)
plt.show()
```

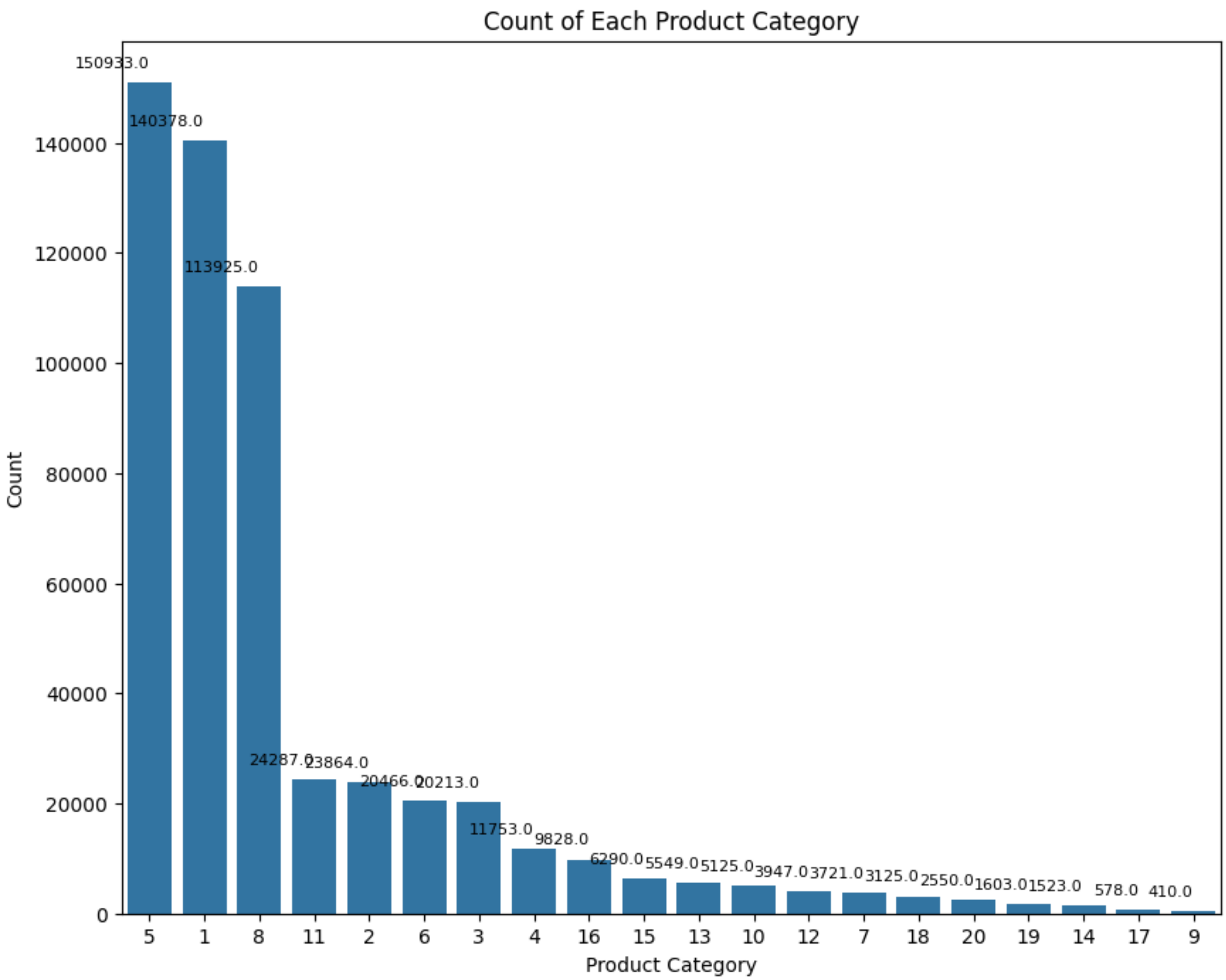



```
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(15,10))
sns.histplot(data=df, x='Gender', ax=axis[0,0],color = "orange")
sns.histplot(data=df, x='City_Category', ax=axis[0,1],color = "green")
sns.histplot(data=df, x='Occupation', ax=axis[1,0])
sns.histplot(data=df, x='Marital_Status',ax=axis[1,1],color = "grey")
plt.show()
```




```
plt.figure(figsize=(10, 8))
sns.countplot(data=df, x='Product_Category', order=df['Product_Category'].value_c
```

```
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.title('Count of Each Product Category')
for p in plt.gca().patches:
    plt.gca().annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.ge
    ha='right', va='center', fontsize=8, color='black', xytext=(0, 10), textcoord
plt.show()
```


Count of Each Product Category
150933.0
140378.0
113925.0
24287.0
23864.0
20466.0
20213.0
11753.0
9828.0
6290.0
5549.0
5125.0
3947.0
3721.0
3125.0
2550.0
1603.0
1523.0
578.0
410.0
Count
0
20000
40000
60000
80000
100000
120000
140000
5 1 8 11 2 6 3 4 16 15 13 10 12 7 18 20 19 14 17 9
Product Category

Insights:

Product categories 5, 1, and 8 are the most frequently purchased.

Men tend to have a greater purchasing capacity than women.

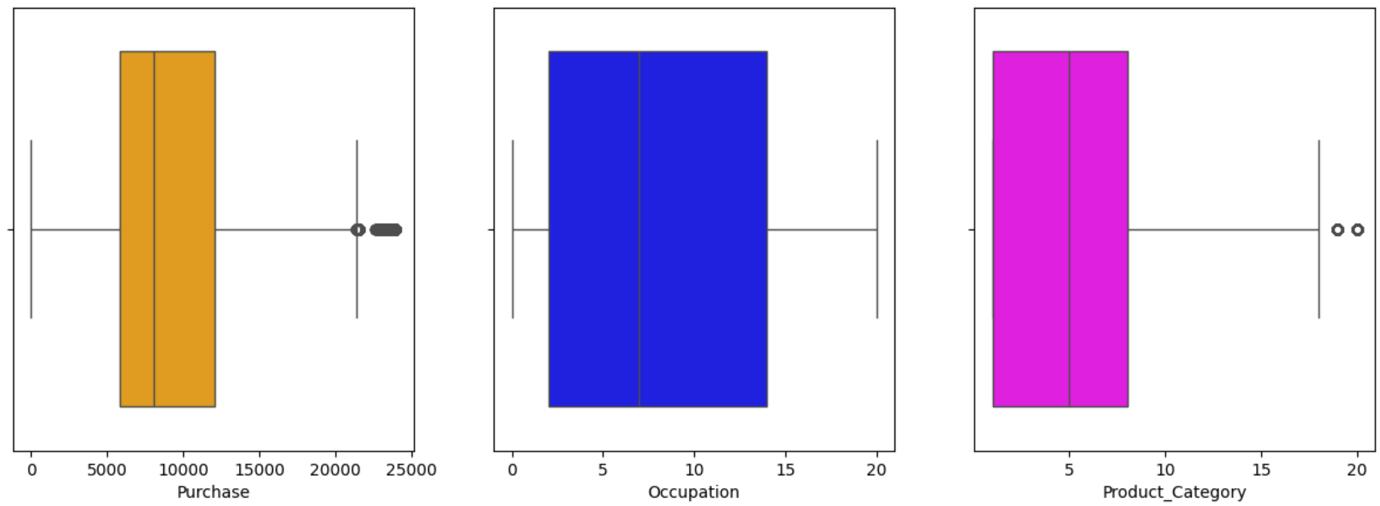
There are more users residing in the B city region.

The majority of users are unmarried.

The maximum purchase falls within the range of 5000 to 15000.

Outliers detection using BoxPlots:

```
fig, axis = plt.subplots(nrows=1, ncols=3, figsize=(15,2))
fig.subplots_adjust(top=2)
sns.boxplot(data=df, x='Purchase', ax=axis[0],color = "orange")
sns.boxplot(data=df, x='Occupation', ax=axis[1],color = "blue")
sns.boxplot(data=df, x='Product_Category', ax=axis[2],color = "magenta")
plt.show()
```



Insights:

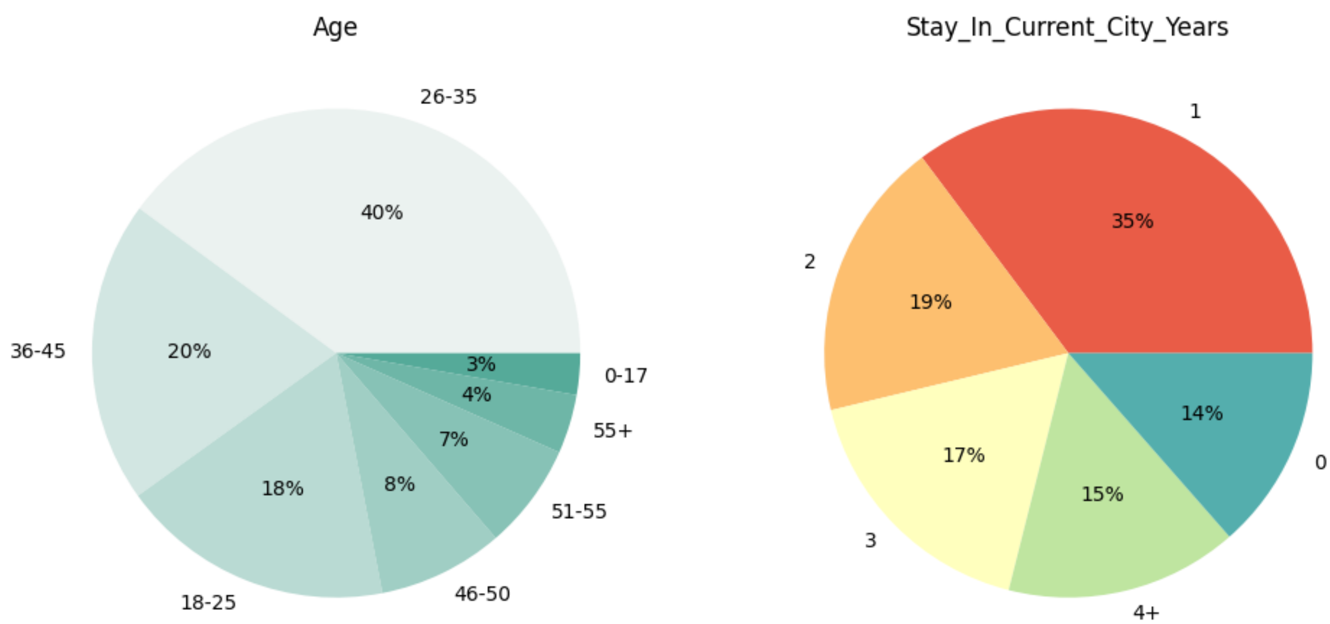
There are outliers present in the purchase data.

There are no outliers in the occupation data.

While there are some outliers in the product categories, the majority of purchases fall within the range of 1 to 8.

Using pie chart:

```
unique_colors_age = sns.color_palette("light:#5A9", len(df['Age'].unique()))
unique_colors_city_years = sns.color_palette("Spectral", len(df['Stay_In_Current_

fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(12, 8))

data_age = df['Age'].value_counts(normalize=True) * 100
axs[0].pie(x=data_age.values, labels=data_age.index, autopct='%.0f%%', colors=uni
axs[0].set_title("Age")

data_city_years = df['Stay_In_Current_City_Years'].value_counts(normalize=True) *
axs[1].pie(x=data_city_years.values, labels=data_city_years.index, autopct='%.0f%
axs[1].set_title("Stay_In_Current_City_Years")
plt.show()
```



Insights :

1- Around 40% of users fall within the age range of 26–35, followed by 20% in the 36–45 age group, 18% in the 18–25 age group, 8% in the 46–50 age group, 7% in the 51–55 age group, 4% in the 55 and above age category, and only 2% are under the age of 17.

2- Approximately 35% of individuals reside in a city for one year, while 19% stay for two years, 17% for three years, and 15% for four years or more.

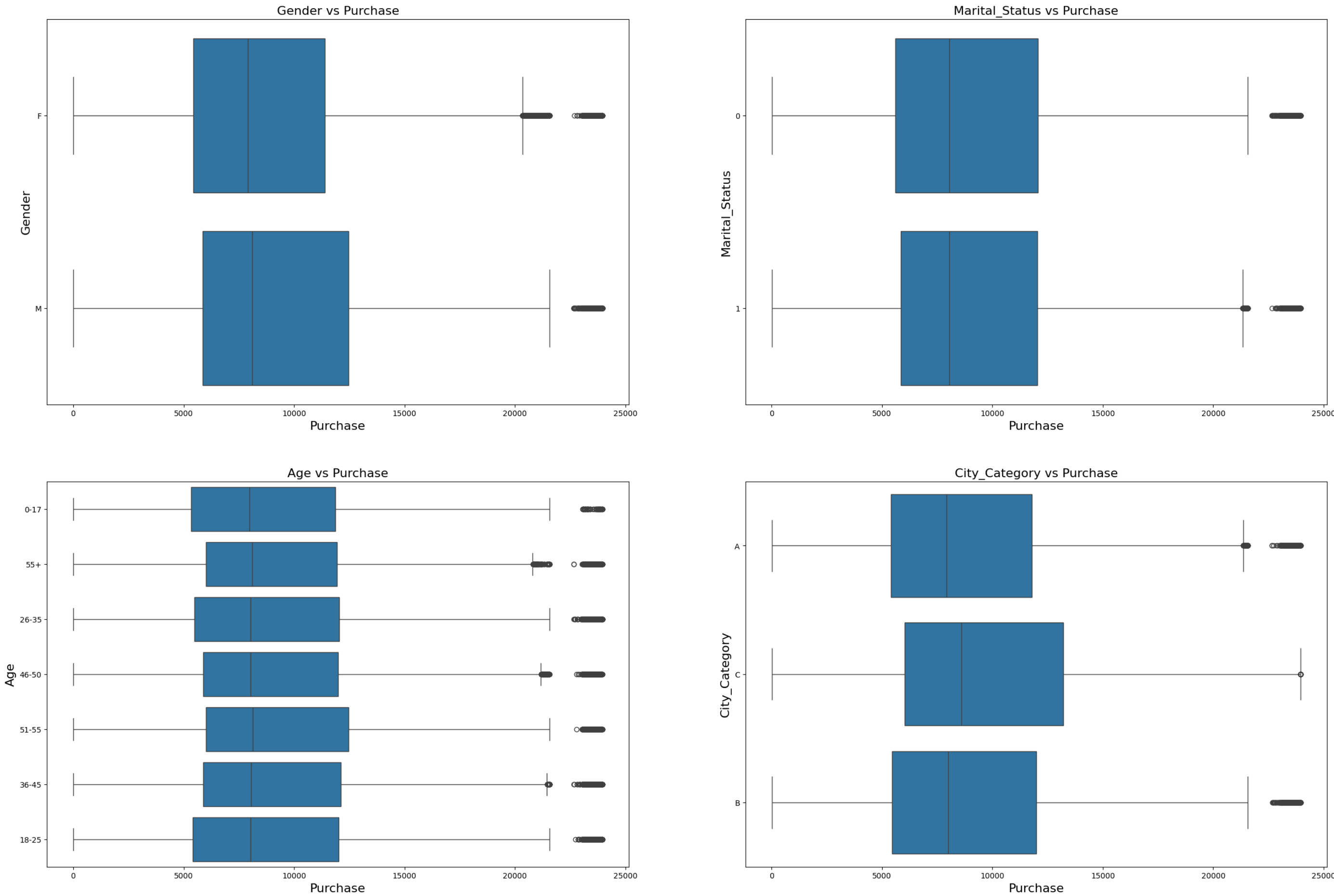
Bivariate Analysis:

Analyzing the variation in purchases with the following,

1. Gender vs Purchase
2. Martial_Status vs Purchase
3. Age vs Purchase
4. City_Category vs Purchase

```
fig1, axs=plt.subplots(nrows=2,ncols=2, figsize=(30,20))

sns.boxplot(data=df, y='Gender',x ='Purchase',orient='h',ax=axs[0,0])
axs[0,0].set_title("Gender vs Purchase", fontsize=16)
axs[0,0].set_xlabel("Purchase", fontsize=16)
axs[0,0].set_ylabel("Gender", fontsize=16)

sns.boxplot(data=df, y='Marital_Status',x ='Purchase',orient='h',ax=axs[0,1])
axs[0,1].set_title("Marital_Status vs Purchase", fontsize=16)
axs[0,1].set_xlabel("Purchase", fontsize=16)
axs[0,1].set_ylabel("Marital_Status", fontsize=16)

sns.boxplot(data=df, y='Age',x ='Purchase',orient='h',ax=axs[1,0])
axs[1,0].set_title("Age vs Purchase", fontsize=16)
axs[1,0].set_xlabel("Purchase", fontsize=16)
axs[1,0].set_ylabel("Age", fontsize=16)

sns.boxplot(data=df, y='City_Category',x ='Purchase',orient='h',ax=axs[1,1])
axs[1,1].set_title("City_Category vs Purchase", fontsize=16)
```

```
axs[1,1].set_xlabel("Purchase", fontsize=16)
axs[1,1].set_ylabel("City_Category", fontsize=16)
plt.show()
```


Gender vs Purchase
Gender
F
M
Purchase
Marital_Status vs Purchase
Marital_Status
0
1
Purchase
Age vs Purchase
Age
0-17
55+
26-35
46-50
51-55
36-45
18-25
Purchase
City_Category vs Purchase
City_Category
A
C
B
Purchase
0
5000
10000
15000
20000
25000

Insights:

1- Gender vs. Purchase

a) The median for males and females is almost equal.

b) Females have more outliers compared to males.

c) Males purchased more compared to females.

2- Martial Status vs. Purchase

a) The median for married and single people is almost equal.

b) Outliers are present in both records.

3- Age vs. Purchase

a) The median for all age groups is almost equal.

b) Outliers are present in all age groups.

4- City Category vs. Purchase

a) The C city region has very low outliers compared to other cities.

b) A and B city region medians are almost the same.

```
q1 = df["Purchase"].quantile(0.25)
q3 = df["Purchase"].quantile(0.75)
IQR = q3-q1
outliers = df["Purchase"][((df["Purchase"]<(q1-1.5*IQR)) | (df["Purchase"]>(q3+1.
print("number of outliers: "+ str(len(outliers)))
print("max outlier value:"+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))
```

```
number of outliers: 2677
max outlier value:23961
min outlier value: 21401
```

```
avg_by_gender = df.groupby('Gender')['Purchase'].mean()
print(f'Average purchase of male and female : \n{avg_by_gender}')
```

```
Average purchase of male and female :
Gender
F    8734.565765
M    9437.526040
Name: Purchase, dtype: float64
```

```
agg_df = df.groupby(['User_ID', 'Gender'])[['Purchase']].agg({'Purchase': ['sum',
agg_df = agg_df.reset_index()
agg_df = agg_df.sort_values(by='User_ID', ascending=False)

print(f"Top 10 purchase from male and female\n{agg_df.head(10)}")
```

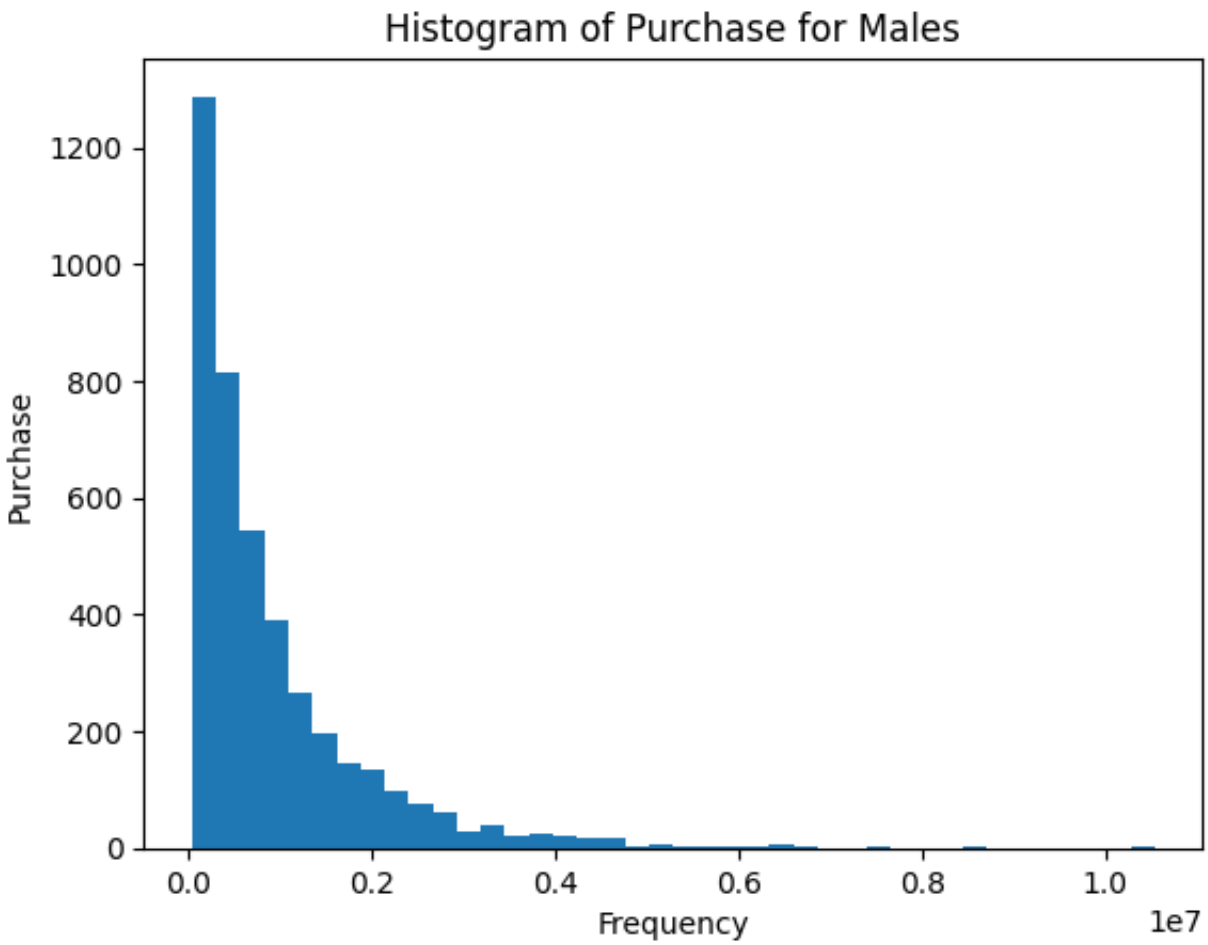
```
Top 10 purchase from male and female
      User_ID Gender Purchase
                          sum          mean
5890  1006040      M  1653299   9184.994444
5889  1006039      F   590319   7977.283784
5888  1006038      F    90034   7502.833333
5887  1006037      F  1119538   9176.540984
5886  1006036      F  4116058   8007.894942
5885  1006035      F   956645   6293.717105
5884  1006034      M   197086  16423.833333
5883  1006033      M   501843  13940.083333
5882  1006032      M   517261   9404.745455
5881  1006031      F   286374   9237.870968
```

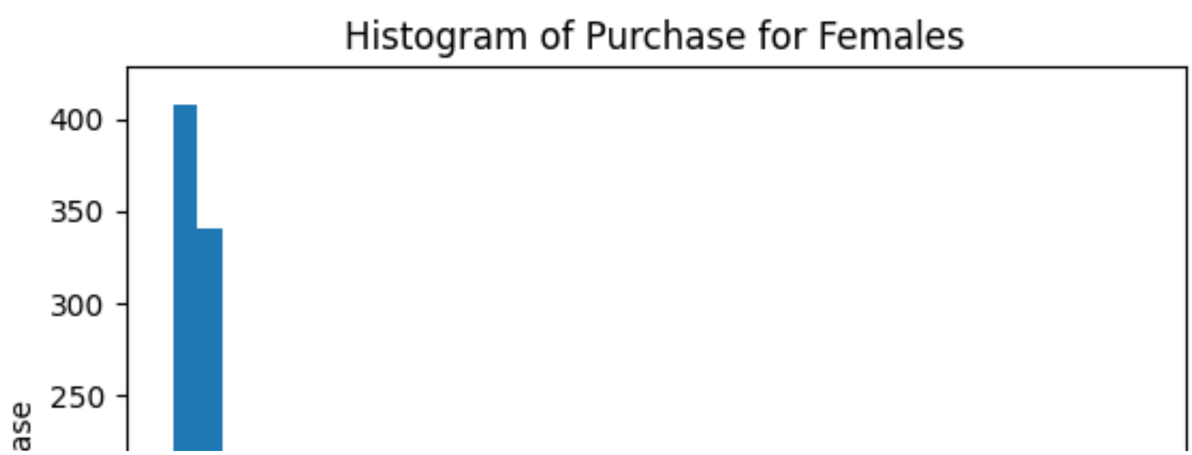
```
Gender_wise_count=agg_df['Gender'].value_counts()
print(f'Each gender wise count : \n{Gender_wise_count}')
```

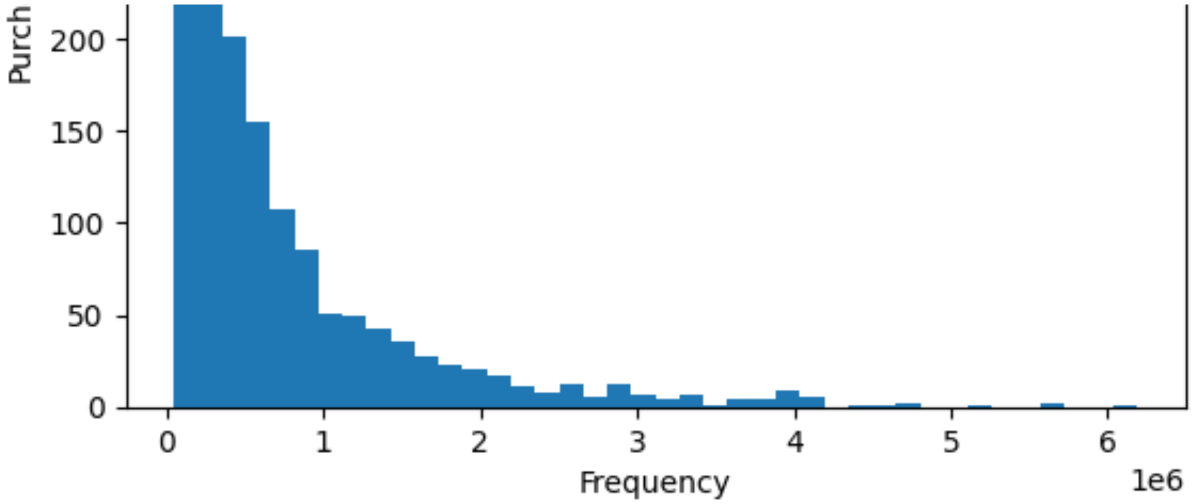
```
Each gender wise count :
M    4225
F    1666
Name: Gender, dtype: int64
```

```
sum_by_gender = df.groupby(['User_ID', 'Gender'])['Purchase'].sum()
sum_by_gender = sum_by_gender.reset_index()
sum_by_gender = sum_by_gender.sort_values(by='User_ID', ascending=False)

male_data = sum_by_gender[sum_by_gender['Gender']=='M']['Purchase']
plt.hist(male_data, bins=40)
plt.ylabel('Purchase')
plt.xlabel('Frequency')
```

```
plt.title('Histogram of Purchase for Males')
plt.show()

Female_data = sum_by_gender[sum_by_gender['Gender']=='F']['Purchase']
plt.hist(Female_data, bins=40)
plt.ylabel('Purchase')
plt.xlabel('Frequency')
plt.title('Histogram of Purchase for Females')
plt.show()
```








```
Mean_by_gender = df.groupby(['User_ID', 'Gender'])['Purchase'].sum()
Mean_by_gender = Mean_by_gender.reset_index()
Mean_by_gender = Mean_by_gender.sort_values(by='User_ID', ascending=False)
Male_cust_avg = Mean_by_gender[Mean_by_gender['Gender']=='M']['Purchase'].mean()
Female_cust_avg = Mean_by_gender[Mean_by_gender['Gender']=='F']['Purchase'].mean(
print(f'Male customer average spent amount: {Male_cust_avg}')
print(f'Female customer average spent amount: {Female_cust_avg}')
```

```
Male customer average spent amount: 925344.4023668639
Female customer average spent amount: 712024.3949579832
```

Insights:

Males outspend females on average.

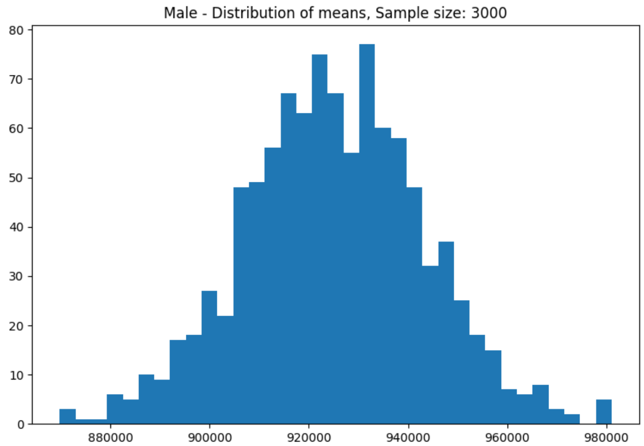
The largest purchase on record is associated with user ID 1006040, who is male.

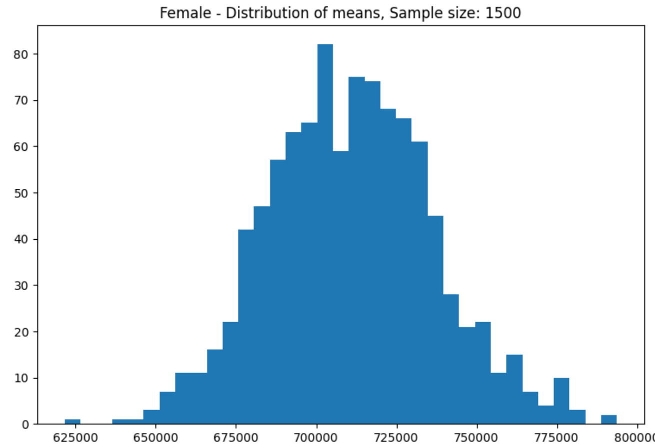
While many females make purchases, their spending tends to be lower overall.

```
male_df = sum_by_gender[sum_by_gender['Gender']=='M']
female_df = sum_by_gender[sum_by_gender['Gender']=='F']

male_sample_size = 3000
female_sample_size = 1000
num_repitions = 1000

random_sample_male = male_df.sample(n=male_sample_size)
random_sample_female = female_df.sample(n=female_sample_size)
```

```
male_means = random_sample_male['Purchase'].mean()
print(f'Population mean: random male samples mean purchase value: {male_means}')
female_means = random_sample_female['Purchase'].mean()
print(f'Population mean: random Female samples mean purchase value : {female_mean

Male_sample_mean = round(male_df['Purchase'].mean(),2)
print(f'Sample means of Male purchase : {Male_sample_mean}')
Male_std_value = round(male_df['Purchase'].std(),2)
print(f'Sample STD of Male purchase : {Male_std_value}')

Female_sample_mean = round(female_df['Purchase'].mean(),2)
print(f'Sample means of Female purchase : {Female_sample_mean}')
Female_std_value = round(female_df['Purchase'].std(),2)
print(f'Sample STD of Female purchase : {Female_std_value}')

male_means1 = []
female_means1 = []

for _ in range(num_repitions):
    male_mean2 = male_df.sample(male_sample_size,replace=True)['Purchase'].mean()
    female_mean2 = female_df.sample(female_sample_size,replace=True)['Purchase'].
    male_means1.append(male_mean2)
    female_means1.append(female_mean2)

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
axis[0].hist(male_means1, bins=35)
axis[1].hist(female_means1, bins=35)
axis[0].set_title("Male - Distribution of means, Sample size: 3000")
axis[1].set_title("Female - Distribution of means, Sample size: 1500")
plt.show()
```

```
Population mean: random male samples mean purchase value: 934842.511
Population mean: random Female samples mean purchase value : 743954.844
Sample means of Male purchase : 925344.4
Sample STD of Male purchase : 985830.1
Sample means of Female purchase : 712024.39
Sample STD of Female purchase : 807370.73
```





Insights:

Referring to the provided data and a 95% confidence level:

a) The mean spending for male customers is estimated to fall within the range of 896,453.54 to 954,235.25.

b) The mean spending for female customers is estimated to range from $683,133.53 to 740,915.24.

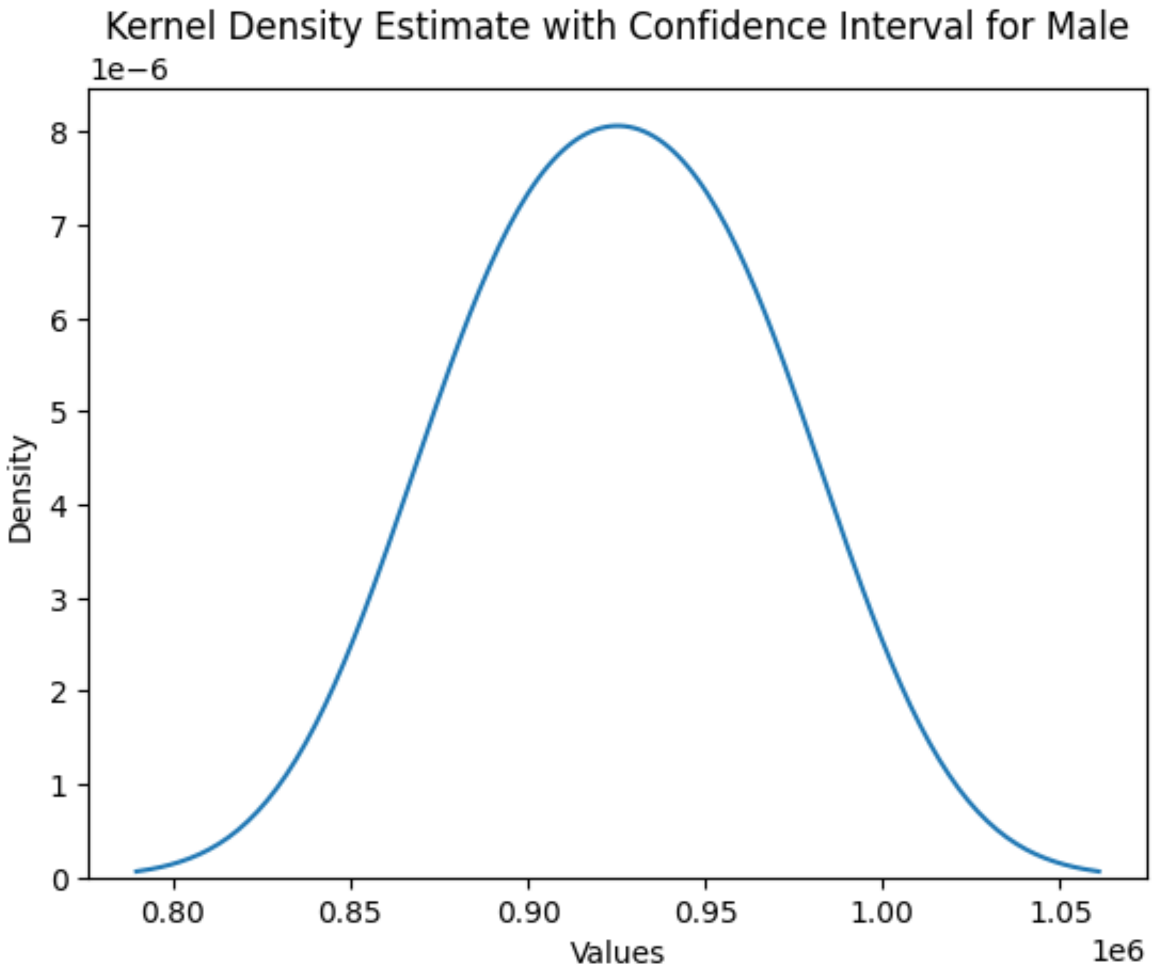
The confidence intervals for the average spending of male and female customers do not intersect.

Considering the data, it's advisable for the company to focus its marketing efforts more towards male customers, as they demonstrate higher spending compared to females.

```
sample_size = 3000
confidence_level = 0.95
z_critical = stats.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_critical * (Male_std_value / np.sqrt(sample_size))
z_critical = stats.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_critical * (Female_std_value / np.sqrt(sample_size))
```

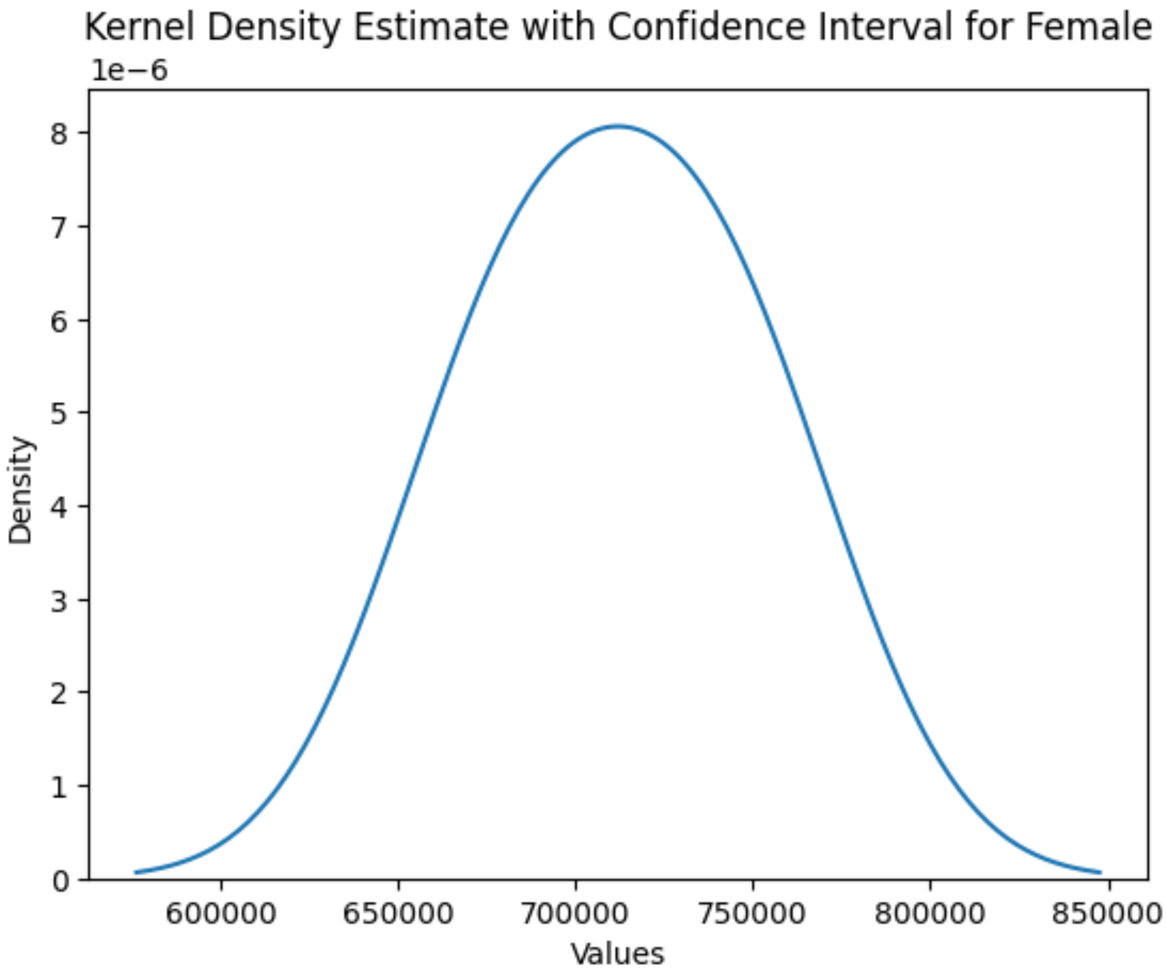
```
Male_confidence_interval = (Male_sample_mean - margin_of_error, Male_sample_mean
print("Confidence Interval 95% Male:", Male_confidence_interval)
sns.kdeplot(Male_confidence_interval)
plt.xlabel('Values')
plt.ylabel('Density')
plt.title('Kernel Density Estimate with Confidence Interval for Male')
plt.show()
```

```
Confidence Interval 95% Male: (896453.5403615071, 954235.259638493)
```



```
Female_confidence_interval = (Female_sample_mean - margin_of_error, Female_sample_
print("Confidence Interval 95% Female:", Female_confidence_interval)
sns.kdeplot(Female_confidence_interval)
plt.xlabel('Values')
plt.ylabel('Density')
plt.title('Kernel Density Estimate with Confidence Interval for Female')
plt.show()
```

```
Confidence Interval 95% Female: (683133.5303615071, 740915.2496384929)
```



Insight

1- Referring to the provided data and a 95% confidence level:

a) The mean spending for male customers is estimated to fall within the range of $896,453.54to$ 954,235.25.

b) The mean spending for female customers is estimated to range from $683,133.53to$ 740,915.24.

2- The confidence intervals for the average spending of male and female customers do not intersect.

3- Considering the data, it's advisable for the company to focus its marketing efforts more towards male customers, as they demonstrate higher spending compared to females.

Results when the same activity is performed for Married vs Unmarried

```
sum_by_Marital_Status = df.groupby(['User_ID', 'Marital_Status'])['Purchase'].sum
sum_by_Marital_Status = sum_by_Marital_Status.reset_index()
sum_by_Marital_Status = sum_by_Marital_Status.sort_values(by='User_ID', ascending
Married_cust_avg = sum_by_Marital_Status[sum_by_Marital_Status['Marital_Status']=
print(f'Married customer average spent amount: {Married_cust_avg}')
```

```
Married customer average spent amount: 843526.7966855295
```

```
sum_by_Marital_Status = df.groupby(['User_ID', 'Marital_Status'])['Purchase'].sum
sum_by_Marital_Status = sum_by_Marital_Status.reset_index()
sum_by_Marital_Status = sum_by_Marital_Status.sort_values(by='User_ID', ascending
Unmarried_cust_avg = sum_by_Marital_Status[sum_by_Marital_Status['Marital_Status'
print(f'Unmarried customer average spent amount: {Unmarried_cust_avg}')
```

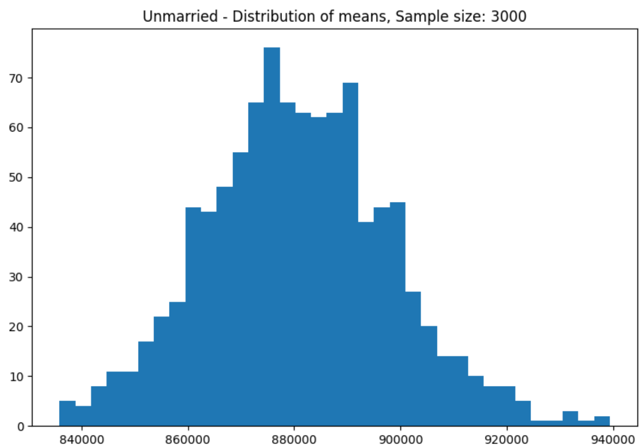
```
Unmarried customer average spent amount: 880575.7819724905
```

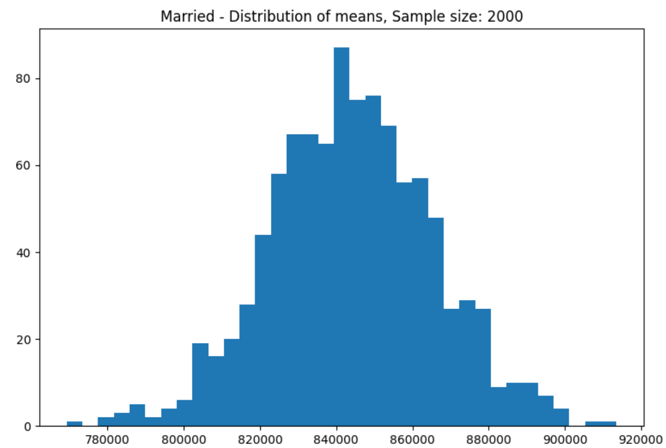
```
Unmarried_df = sum_by_Marital_Status[sum_by_Marital_Status['Marital_Status']==0]
```

```
Married_df = sum_by_Marital_Status[sum_by_Marital_Status['Marital_Status']==1]


Unmarried_sample_size = 3000
Married_sample_size = 2000
num_repitions = 1000


random_sample_Unmarried = Unmarried_df.sample(n=Unmarried_sample_size)
random_sample_Married = Married_df.sample(n=Married_sample_size)


Unmarried_means = random_sample_Unmarried['Purchase'].mean()
print(f'Population mean: random Unmarried samples mean purchase value: {Unmarried_
Married_means = random_sample_Married['Purchase'].mean()
print(f'Population mean: random Married samples mean purchase value : {Married_me


Unmarried_sample_mean = round(Unmarried_df['Purchase'].mean(),2)
print(f'Sample means of Unmarried purchase : {Unmarried_sample_mean}')
Unmarried_std_value = round(Unmarried_df['Purchase'].std(),2)
print(f'Sample STD of Unmarried purchase : {Unmarried_std_value}')


Married_sample_mean = round(Married_df['Purchase'].mean(),2)
print(f'Sample means of Married purchase : {Married_sample_mean}')
Married_std_value = round(Married_df['Purchase'].std(),2)
print(f'Sample STD of Married purchase : {Married_std_value}')


Unmarried_means1 = []
Married_means1 = []


for _ in range(num_repitions):
    Unmarried_mean2 = Unmarried_df.sample(Unmarried_sample_size,replace=True)['Pu
    Married_mean2 = Married_df.sample(Married_sample_size,replace=True)['Purchase
    Unmarried_means1.append(Unmarried_mean2)
    Married_means1.append(Married_mean2)


fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
axis[0].hist(Unmarried_means1, bins=35)
axis[1].hist(Married_means1, bins=35)
axis[0].set_title("Unmarried - Distribution of means, Sample size: 3000")
```

```
axis[1].set_title("Married - Distribution of means, Sample size: 2000")
plt.show()
```

```
Population mean: random Unmarried samples mean purchase value: 878703.41933333
Population mean: random Married samples mean purchase value : 830048.116
Sample means of Unmarried purchase : 880575.78
Sample STD of Unmarried purchase : 949436.25
Sample means of Married purchase : 843526.8
Sample STD of Married purchase : 935352.12
```





Insights:

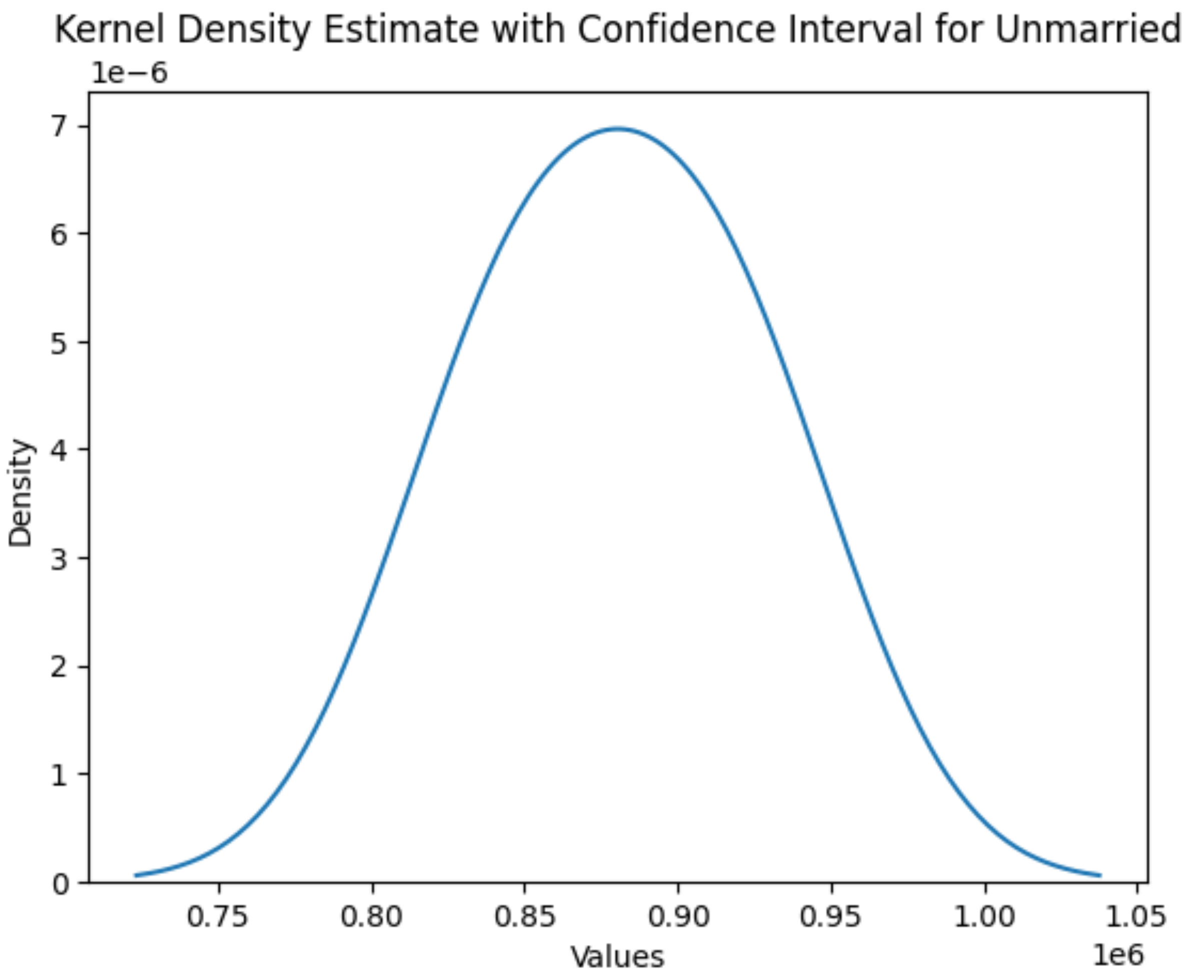
1- The average spending by unmarried customers totals $880,575.78.

2-Married customers, on average, spend $843,526.80.

3- Unmarried customers tend to spend more than married customers.

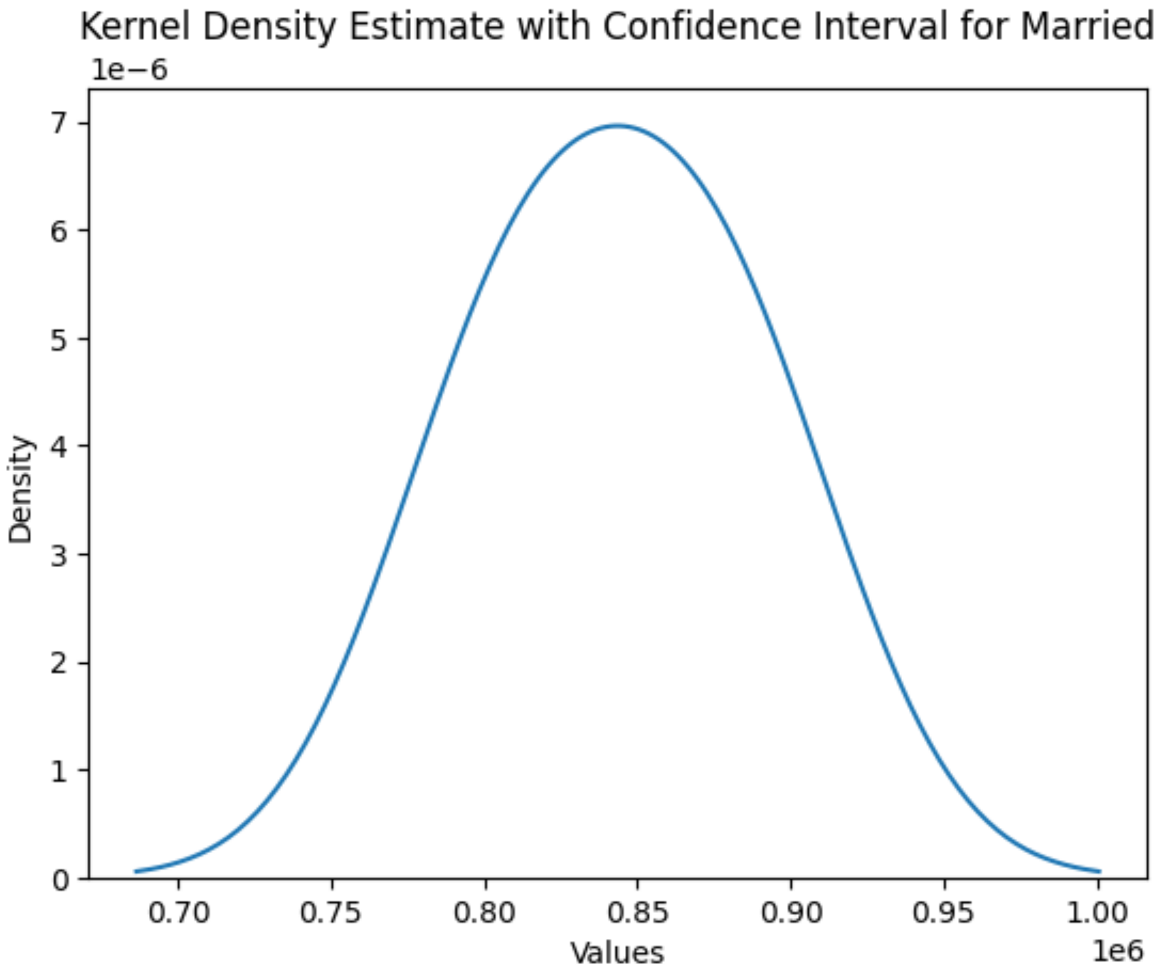
```
sample_size = 3000
confidence_level = 0.95
z_critical = stats.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_critical * (Unmarried_std_value / np.sqrt(sample_size))
z_critical = stats.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_critical * (Married_std_value / np.sqrt(sample_size))


Unmarried_confidence_interval = (Unmarried_sample_mean - margin_of_error, Unmarri
print("Confidence Interval 95% Unmarried:", Unmarried_confidence_interval)
sns.kdeplot(Unmarried_confidence_interval)
plt.xlabel('Values')
plt.ylabel('Density')
plt.title('Kernel Density Estimate with Confidence Interval for Unmarried')
plt.show()
```

```
Confidence Interval 95% Unmarried: (847105.2492916514, 914046.3107083486)
```



```
Married_confidence_interval = (Married_sample_mean - margin_of_error, Married_sam
print("Confidence Interval 95% Married:", Married_confidence_interval)
sns.kdeplot(Married_confidence_interval)
plt.xlabel('Values')
plt.ylabel('Density')
plt.title('Kernel Density Estimate with Confidence Interval for Married')
plt.show()
```

```
Confidence Interval 95% Married: (810056.2692916514, 876997.3307083487)
```



Insights:

Based on the provided data and at a 95% confidence level, the analysis reveals the following insights regarding customer spending habits across various age groups:

a) Customers aged 26 to 35 have the highest projected average spending, with the range estimated between $944, 419.9990 and$ 1,034,842.9516. b) The projected average expenditure for customers aged 36 to 45 is between $819, 003.0902 and$ 940,678.8198. c) For customers aged 18 to 25, the average spending range is estimated to be between $799, 594.4375 and$ 909,664.7362. d) Customers aged 46 to 50 are expected to have an average spending range between $711, 215.1004 and$ 874,125.3830. e) The estimated average spending for customers aged 51 to 55 lies between $685, 670.0292 and$ 840,962.3353. f) The average expenditure for customers aged 55 and above is anticipated to be between $470, 454.5225 and$ 610,200.5797. g) The lowest average spending is projected for customers aged 0 to 17, with the range between $524, 534.4423 and$ 714,973.3156.

From these insights, it is evident that the 26 to 35 age group exhibits the highest spending behavior compared to other age groups. Additionally, spending significantly declines for age groups above 55 and the youngest age group (0 to 17). Notably, the confidence intervals for the average spending of the 26 to 35 and 36 to 45 age groups do not overlap, underscoring a distinct spending pattern between these groups. Therefore, it is recommended that the company prioritizes the 26 to 35 age category for marketing and sales strategies due to their higher spending propensity.

Further recommendations for the company's strategy include:

1) Targeting male customers for retention and acquisition efforts, given their higher spending compared to female customers. 2) Focusing on Product Categories 5, 1, and 8 due to their high purchase frequency, indicating a strong preference among customers. 3) Considering strategies to enhance sales in Product Categories 11, 2, and 6, 3, where competitive purchasing behavior is observed. 4) Prioritizing unmarried customers for marketing efforts, as they tend to spend more than married customers. 5) Focusing marketing and sales initiatives on customers aged 18 to 45, who account for 86% of purchases. 6) Enhancing presence and sales efforts in City Category C, where customers spend more compared to those in City Categories B or A.