Walmart Business Problem

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).

In [316…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [317…
```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/293/original/walmart_data.csv
To: /content/walmart_data.csv
100% 23.0M/23.0M [00:00<00:00, 70.5MB/s]
```

Loading Data Set

In [318…
```
df = pd.read_csv("walmart_data.csv")
```

In [319…
```
df.head()
```

Out[319]:

	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

The above data set can be defined as follows

- User_ID : User ID
- Product_ID : Product ID
- Gender : Sex of User
- Age : Age in bins
- Occupation : Occupation(Masked)
- City_Category : Category of the City (A,B,C)
- StayInCurrentCityYears : Number of years stay in current city
- Marital_Status : Marital Status
- ProductCategory : Product Category (Masked)
- Purchase : Purchase Amount

In [320...
```
df.shape
```
Out[320]:
```
(550068, 10)
```

In [321...
```
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
```

In [322...
```
df.isnull().sum()
```

Out[322]:
```
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
```

No Null values found in the data set

In [323…
```
df.describe()
```

Out[323]:

	User_ID	Occupation	Marital_Status	Product_Category	Purchase
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.000000
mean	1.003029e+06	8.076707	0.409653	5.404270	9263.968713
std	1.727592e+03	6.522660	0.491770	3.936211	5023.065394
min	1.000001e+06	0.000000	0.000000	1.000000	12.000000
25%	1.001516e+06	2.000000	0.000000	1.000000	5823.000000
50%	1.003077e+06	7.000000	0.000000	5.000000	8047.000000
75%	1.004478e+06	14.000000	1.000000	8.000000	12054.000000
max	1.006040e+06	20.000000	1.000000	20.000000	23961.000000

In [324…
```
df.describe(include = "object")
```

Out[324]:

	Product_ID	Gender	Age	City_Category	Stay_In_Current_City_Years
count	550068	550068	550068	550068	550068
unique	3631	2	7	3	5
top	P00265242	M	26-35	B	1
freq	1880	414259	219587	231173	193821

In [325…

```
df['Gender'].value_counts()
```

Out[325]:

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

In [326…

```
df['Occupation'].value_counts()
```

Out[326]:

```
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: Occupation, dtype: int64
```

In [327…

```
df['City_Category'].value_counts()
```

Out[327]:
```
B    231173
C    171175
A    147720
Name: City_Category, dtype: int64
```

In [328…
```
df['Marital_Status'].value_counts()
```

Out[328]:
```
0    324731
1    225337
Name: Marital_Status, dtype: int64
```

In [329…
```
df.corr()
```

Out[329]:

	User_ID	Occupation	Marital_Status	Product_Category	Purchase
User_ID	1.000000	-0.023971	0.020443	0.003825	0.004716
Occupation	-0.023971	1.000000	0.024280	-0.007618	0.020833
Marital_Status	0.020443	0.024280	1.000000	0.019888	-0.000463
Product_Category	0.003825	-0.007618	0.019888	1.000000	-0.343703
Purchase	0.004716	0.020833	-0.000463	-0.343703	1.000000

In [330…
```
df['User_ID'].nunique()
```

Out[330]:
```
5891
```

In [331…
```
df['Product_ID'].nunique()
```

Out[331]:
```
3631
```

In [332…
```
df['Product_Category'].nunique()
```

Out[332]:
```
20
```

There are 5981 unique customers and 3631 Products with 20 Product Categories

In [333…
```
df["Purchase_Categories"] = pd.cut(df['Purchase'],bins = [0,100,1000,10000,20000,50000],labels = ["Affordable Cost", "Moderate Co
```

In [334…
```
df.head()
```

Out[334]:

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

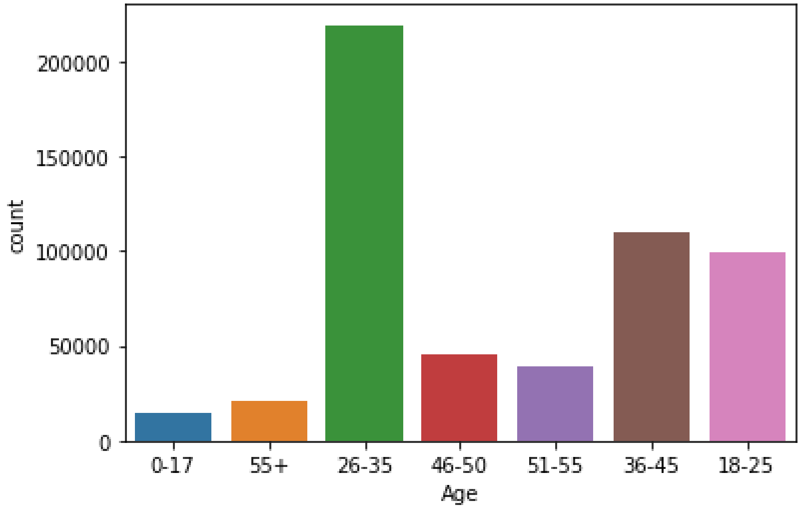
In [335…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 11 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
 10  Purchase_Categories         550068 non-null  category
dtypes: category(1), int64(5), object(5)
memory usage: 42.5+ MB
```

Univariate Analysis

In [336…
```
sns.countplot(x='Age',data=df)
```

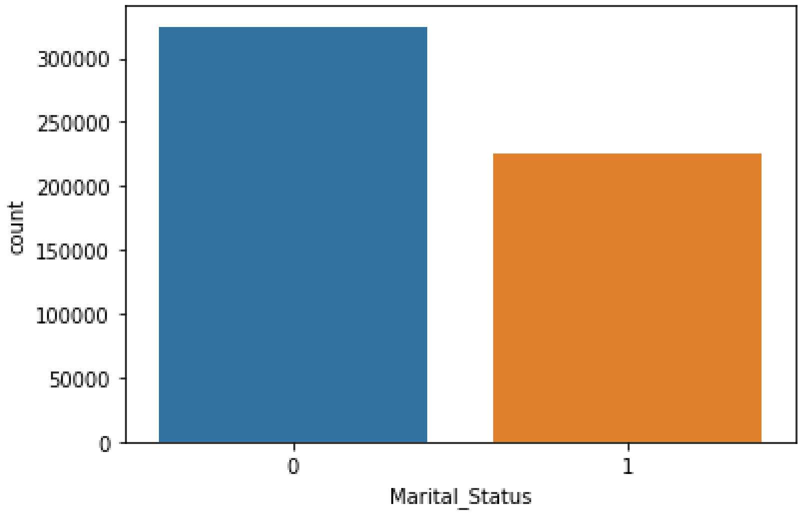
Out[336]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7278a8f610>
```

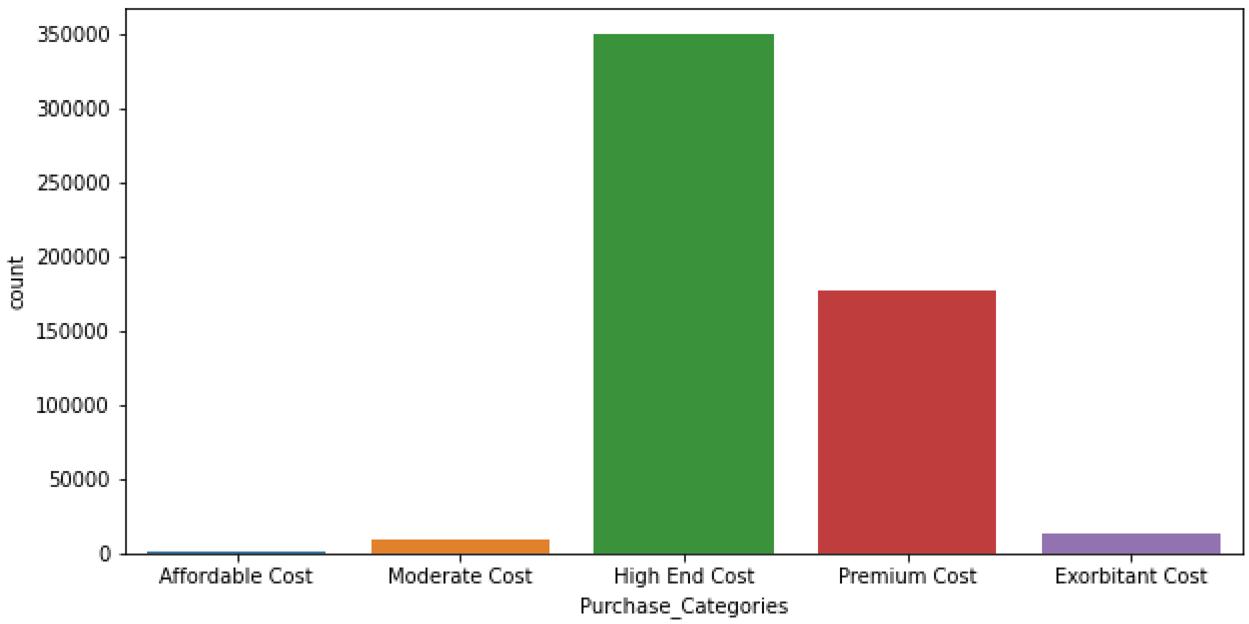


The people between age 26-35 had made more purchases

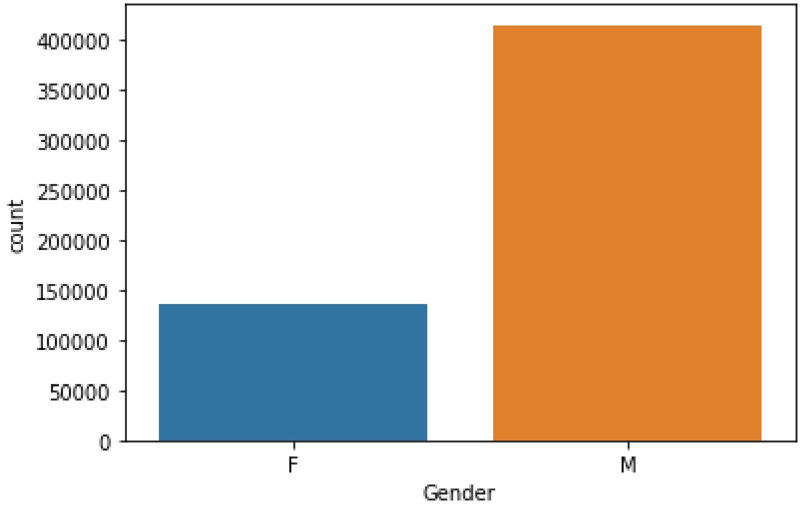
In [337…
```
sns.countplot(x="Marital_Status",data=df)
```

Out[337]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f72788b2af0>
```



In [338…
```
plt.figure(figsize=(10,5))
sns.countplot(x="Purchase_Categories",data=df)
```
Out[338]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f72788a71f0>
```


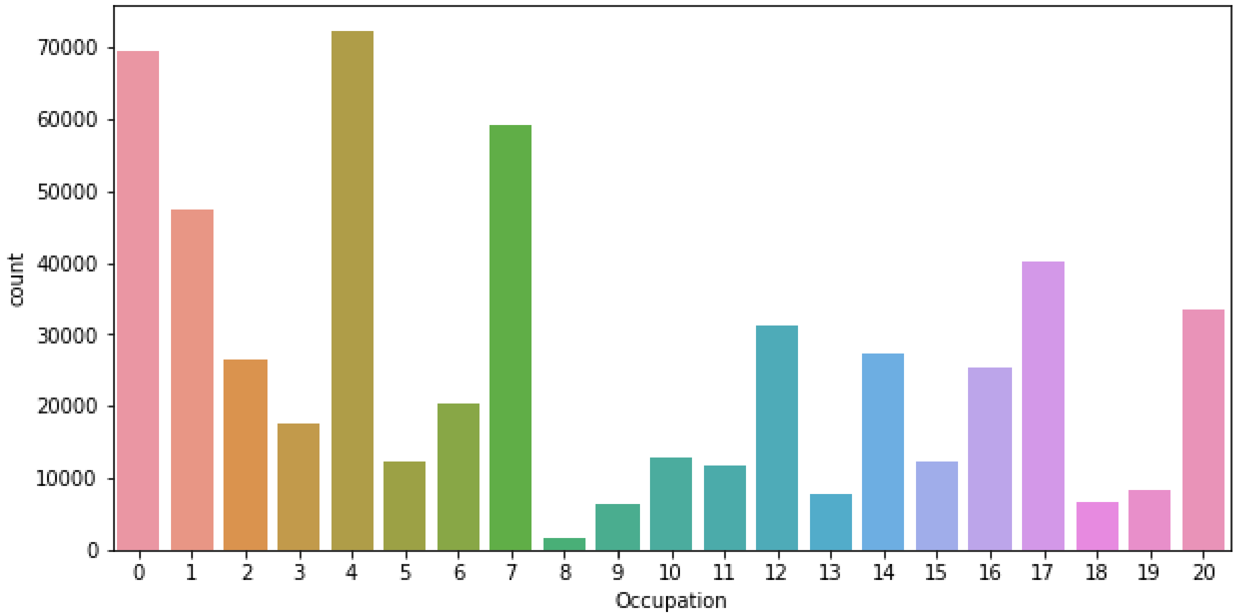
In [339…
```
sns.countplot(data=df, x='Gender')
```
Out[339]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7278bb7580>
```



In [340…

```
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Occupation')
```

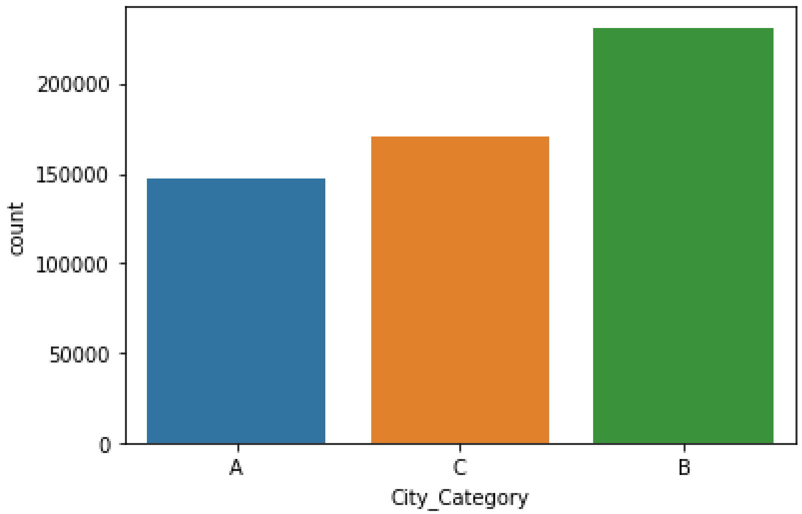
Out[340]: <matplotlib.axes._subplots.AxesSubplot at 0x7f72787859d0>



In [341…

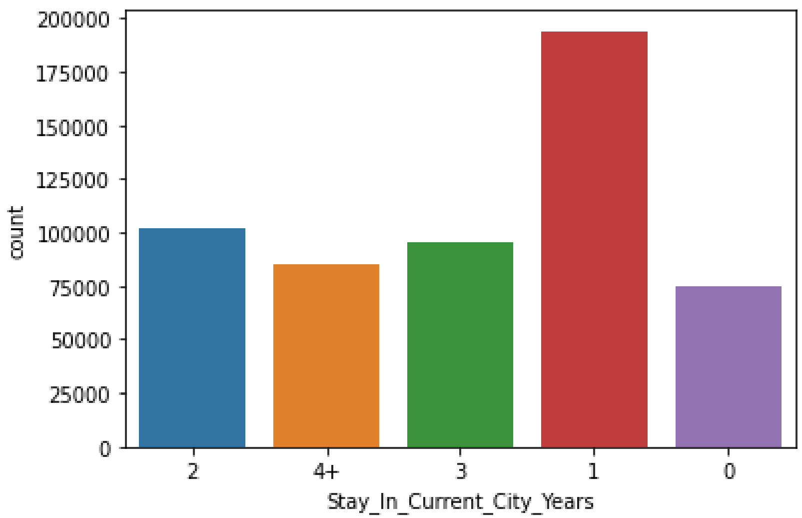
```
sns.countplot(data=df, x='City_Category')
```

Out[341]: <matplotlib.axes._subplots.AxesSubplot at 0x7f72786ce250>



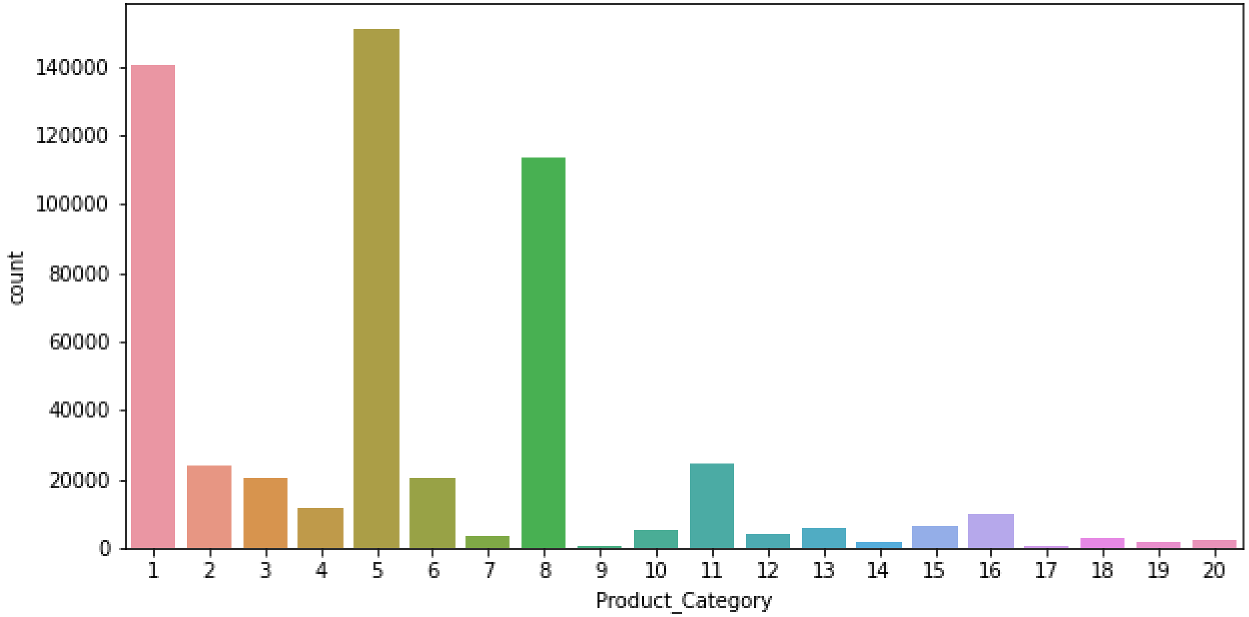
In [342…
```
sns.countplot(x='Stay_In_Current_City_Years', data=df)
```

Out[342]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7278aec8b0>



In [343…
```
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Product_Category')
```

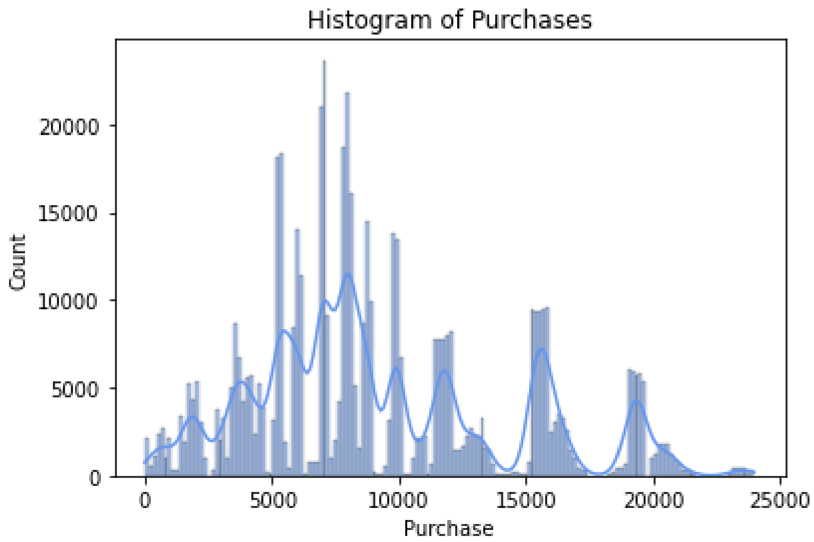
Out[343]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7278d12eb0>



In [344…

```
sns.histplot(df["Purchase"],color="cornflowerblue",kde=True)
plt.title("Histogram of Purchases")
```

Out[344]: Text(0.5, 1.0, 'Histogram of Purchases')



Observations

- Most of the customers are male
- Product Categort 5,1 and 8 are most sold categories
- **B** City has maximum number of users
- There are 20 different types of Occupation and Product_Category
- 4 and 0 occupation has brought most number of products
- More users are Single as compare to Married

In [345…
```
df['Purchase_Categories'].value_counts()
```

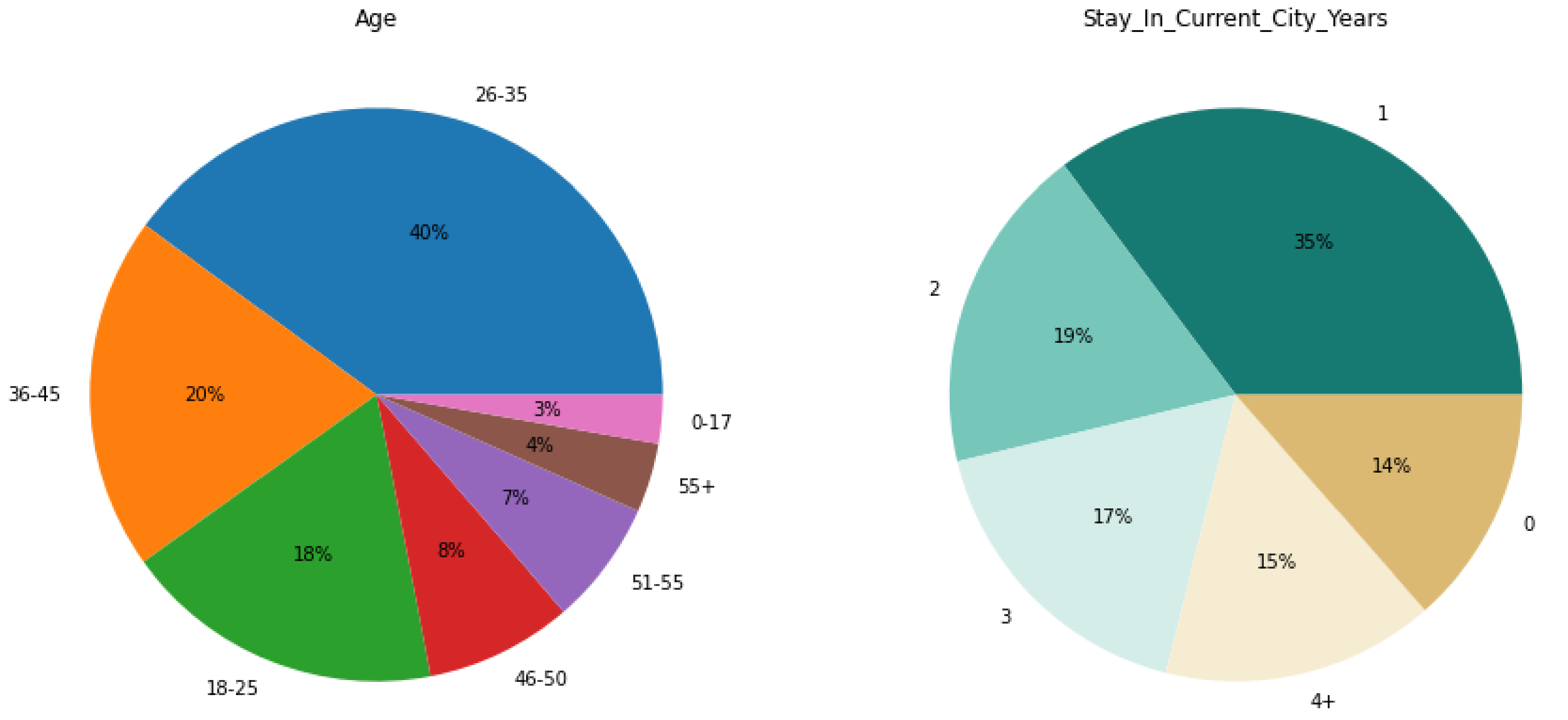
Out[345]:
```
High End Cost      349288
Premium Cost       176759
Exorbitant Cost     12691
Moderate Cost        9727
Affordable Cost      1603
Name: Purchase_Categories, dtype: int64
```

In [346…
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 10))

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

```
data = df['Stay_In_Current_City_Years'].value_counts(normalize=True)*100
palette_color = sns.color_palette('BrBG_r')
axs[1].pie(x=data.values, labels=data.index, autopct='%.0f%%',colors=palette_color)
axs[1].set_title("Stay_In_Current_City_Years")


plt.show()
```

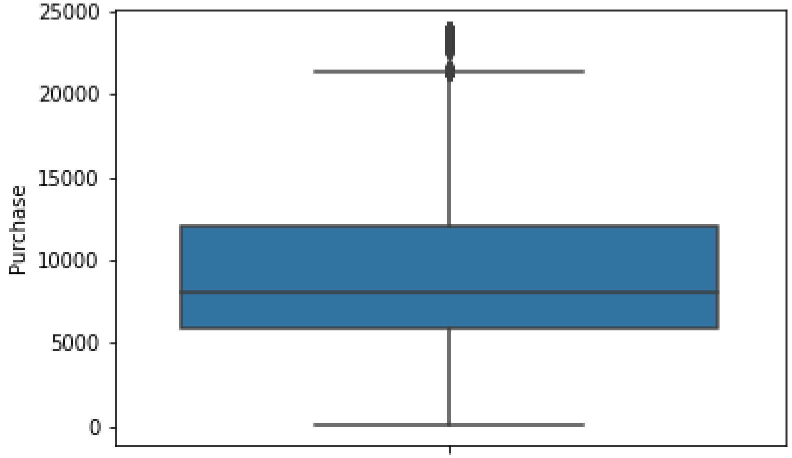


In [347…

```
sns.boxplot(data=df,y='Purchase')
```

Out[347]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f727a937430>
```



In [348…

```
#sns.pairplot(df,hue='Gender',diag_kind='hist', plot_kws={'alpha': 0.5})
```

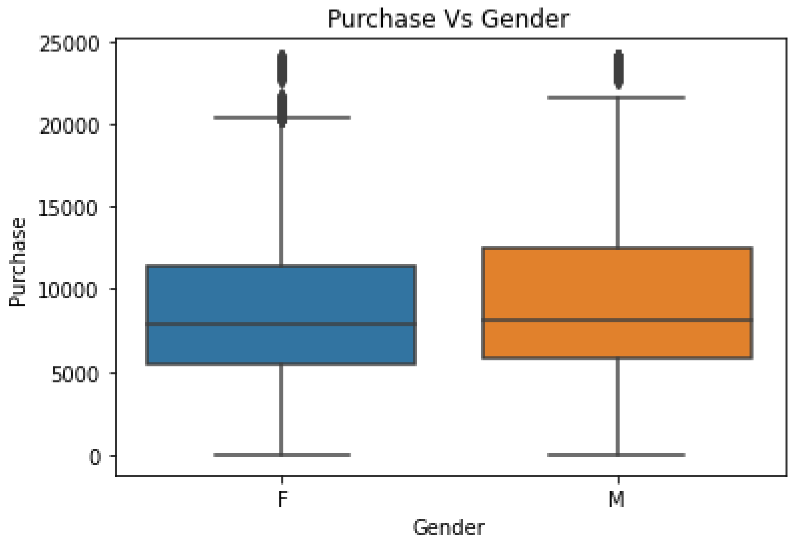
Bivariate Analysis

In [349…

```
sns.boxplot(x='Gender',y='Purchase',data=df)
plt.title("Purchase Vs Gender")
```

Out[349]:

```
Text(0.5, 1.0, 'Purchase Vs Gender')
```

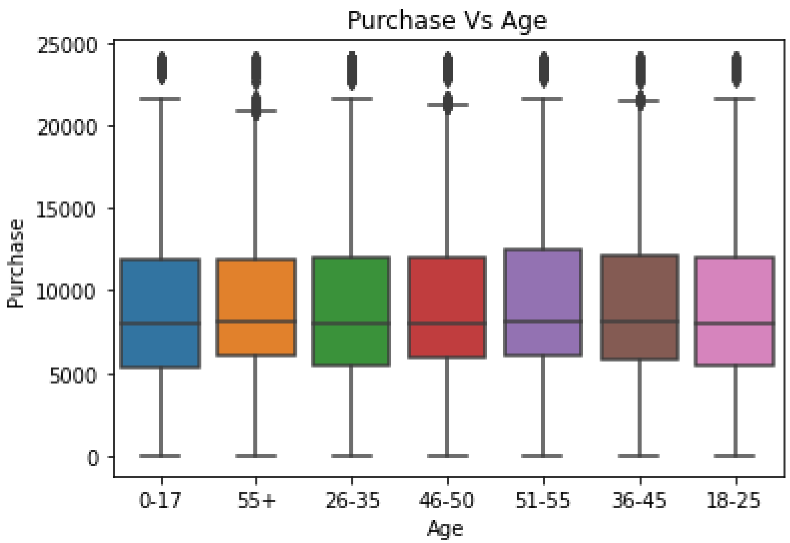


In [350…

```
sns.boxplot(x='Age',y='Purchase',data=df)
plt.title("Purchase Vs Age")
```

Out[350]:

```
Text(0.5, 1.0, 'Purchase Vs Age')
```

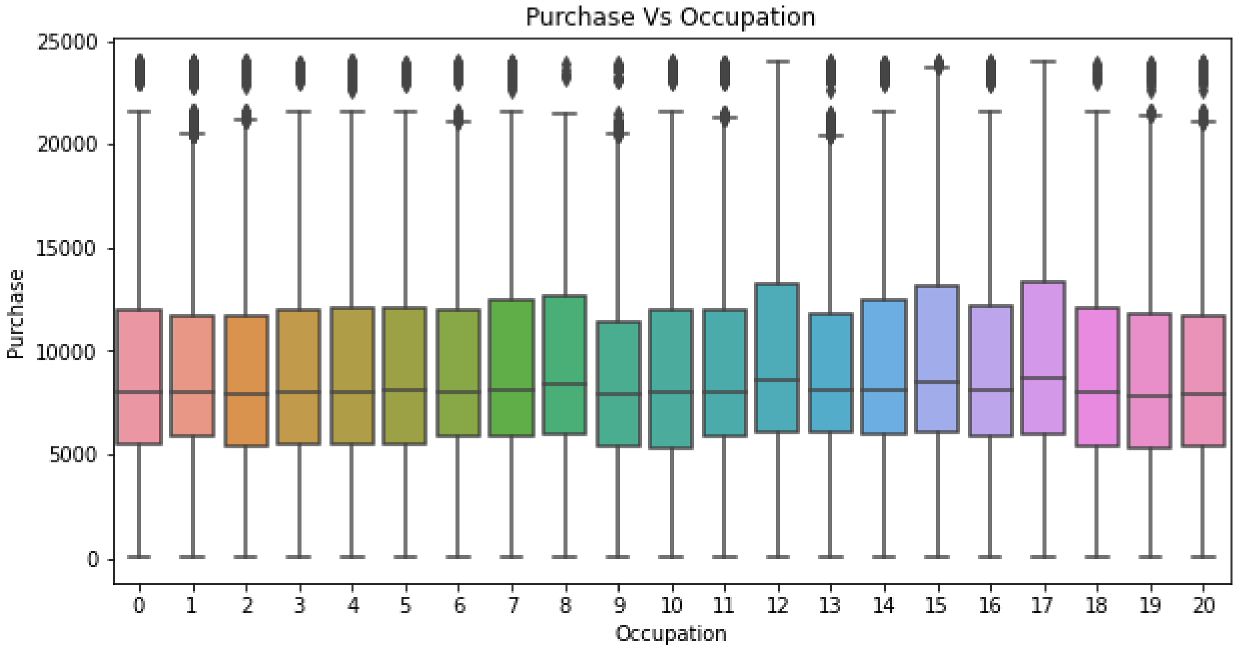


In [351…

```
plt.figure(figsize=(10,5))
sns.boxplot(x='Occupation',y='Purchase',data=df)
plt.title("Purchase Vs Occupation")
```

Out[351]:

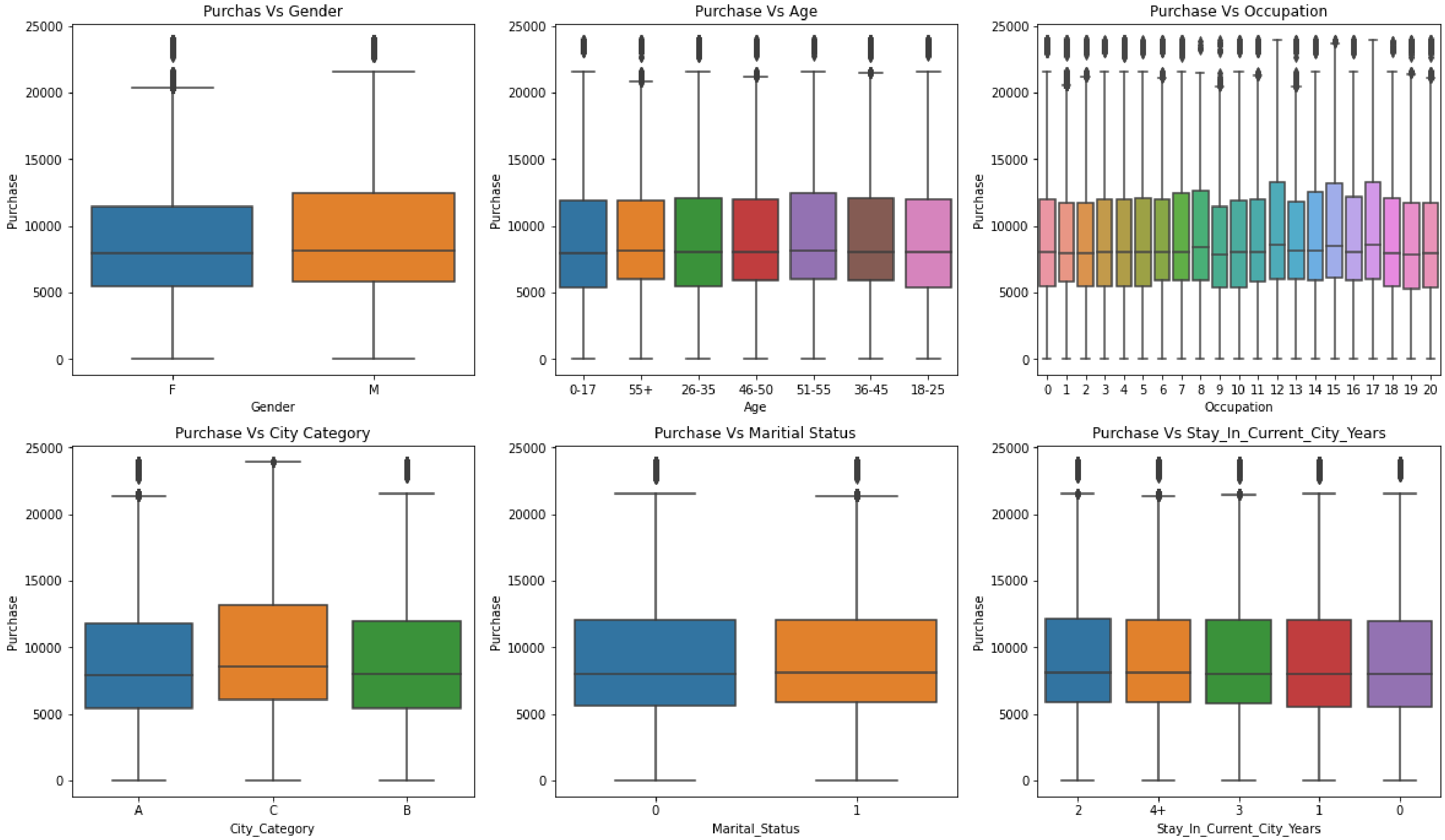
```
Text(0.5, 1.0, 'Purchase Vs Occupation')
```



In [352…

```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(20,10))
fig.subplots_adjust(top=1)

sns.boxplot(x='Gender',y='Purchase',data=df,ax=axis[0,0]).set(title="Purchas Vs Gender")
sns.boxplot(x='Age',y='Purchase',data=df,ax=axis[0,1]).set(title = "Purchase Vs Age")
sns.boxplot(x='Occupation',y='Purchase',data=df,ax=axis[0,2]).set(title = "Purchase Vs Occupation")
sns.boxplot(x='City_Category',y='Purchase',data=df,ax=axis[1,0]).set(title = "Purchase Vs City Category")
sns.boxplot(x='Marital_Status',y='Purchase',data=df,ax=axis[1,1]).set(title = "Purchase Vs Maritial Status")
sns.boxplot(x='Stay_In_Current_City_Years',y='Purchase',data=df,ax=axis[1,2]).set(title = "Purchase Vs Stay_In_Current_City_Years
plt.show()
```

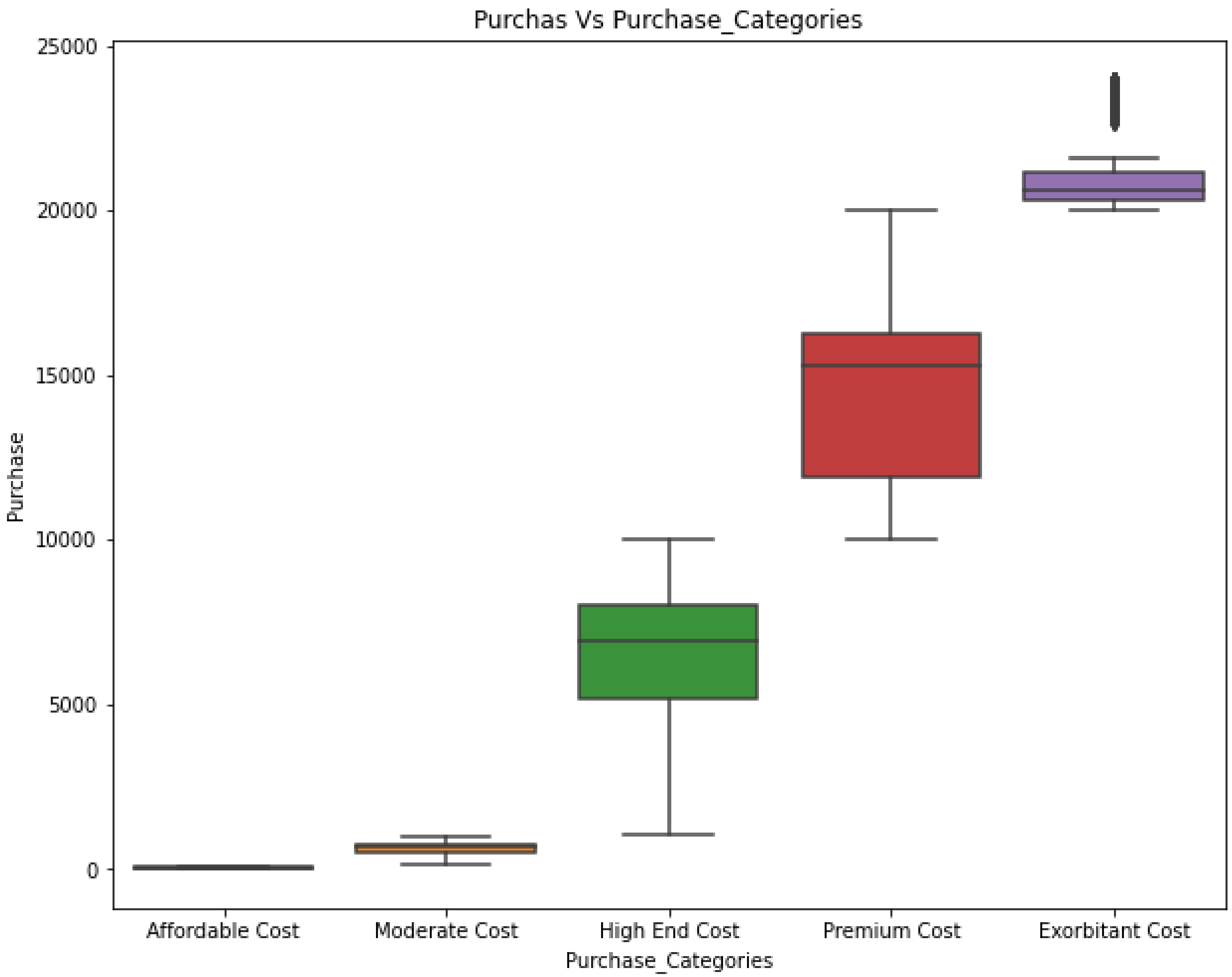


In [353…

```
plt.figure(figsize=(10,8))
sns.boxplot(x='Purchase_Categories',y='Purchase',data=df).set(title="Purchas Vs Purchase_Categories")
```

Out[353]:

```
[Text(0.5, 1.0, 'Purchas Vs Purchase_Categories')]
```

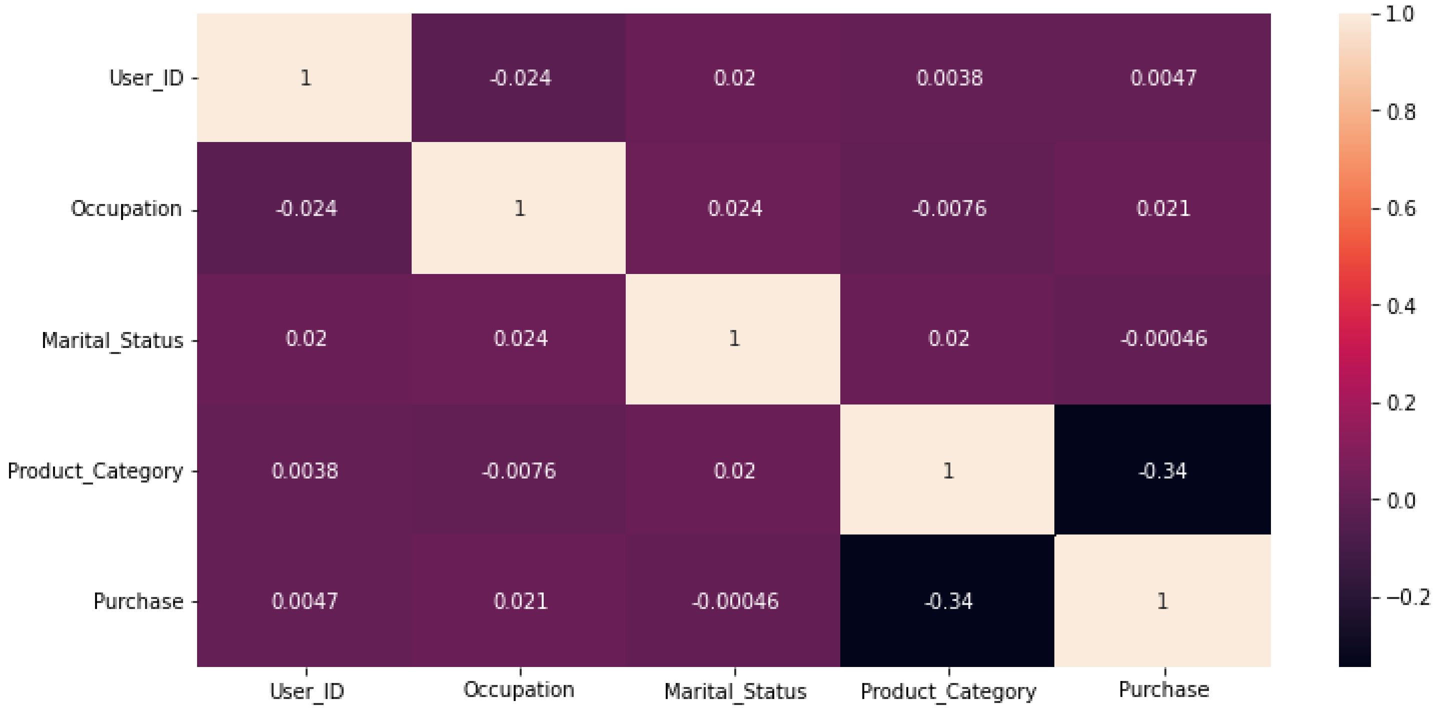


Multivariate Anaysis

In [354…

```
plt.figure(figsize=(12,6))
sns.heatmap(df.corr(), annot=True)
```

Out[354]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7277efd9a0>

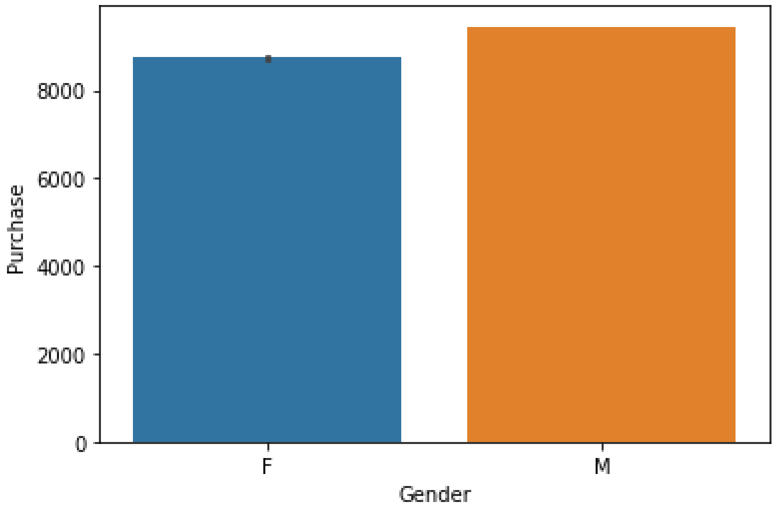


In [355…

```
sns.barplot(x = "Gender", y = "Purchase", data = df)
```

Out[355]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7277e50dc0>
```

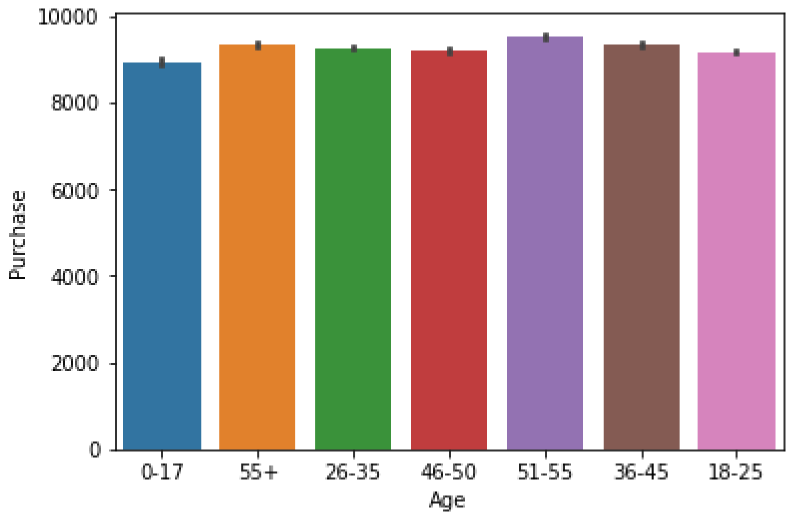


In [356…

```
sns.barplot(x = "Age", y = "Purchase", data = df)
```

Out[356]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7277e1bf70>
```

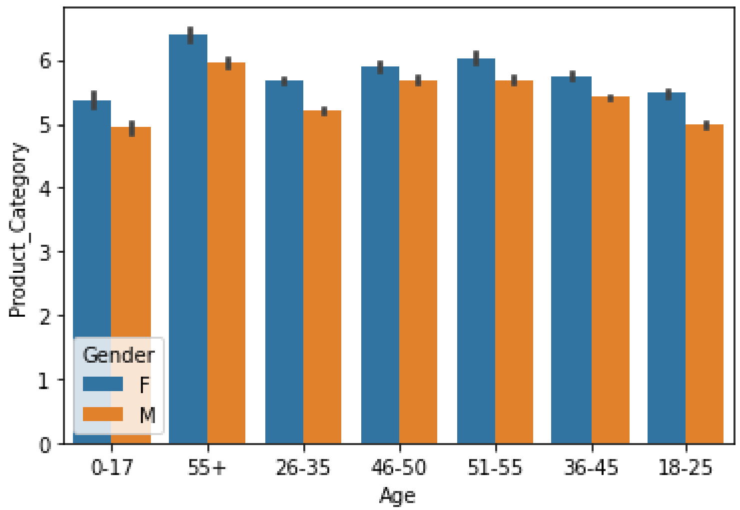


In [357…

```
sns.barplot(x='Age',y='Product_Category',hue='Gender',data=df)
```

Out[357]:

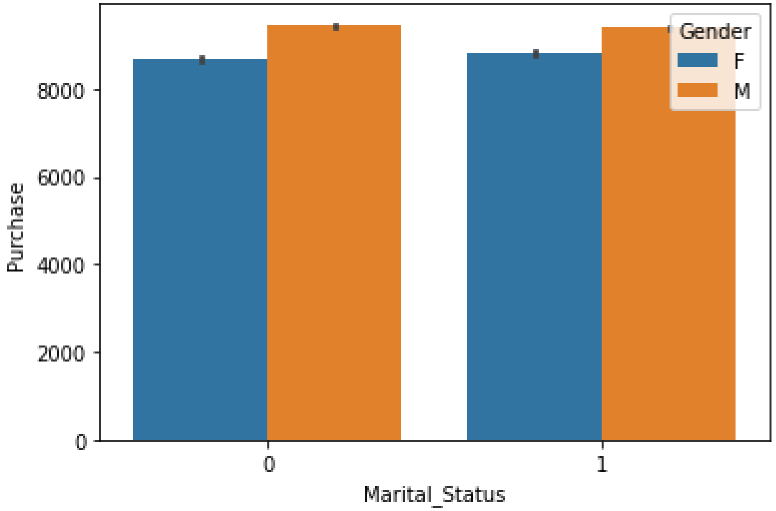
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7277d99a00>
```



In [358…

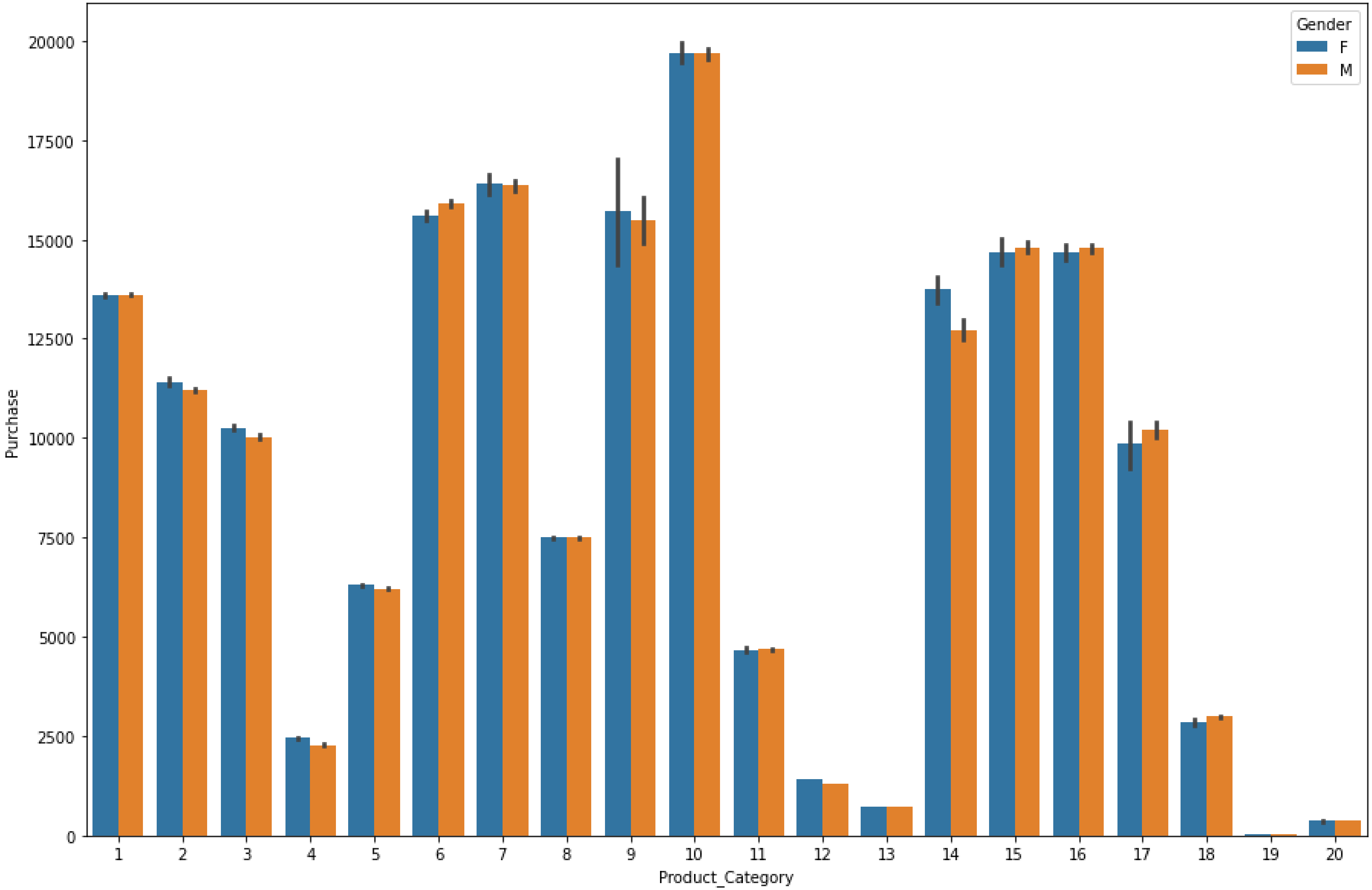
```
sns.barplot(y='Purchase',x='Marital_Status',hue='Gender',data=df)
```

Out[358]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7277d8abe0>



In [359…
```
plt.figure(figsize=(15,10))
sns.barplot(y='Purchase',x='Product_Category',hue='Gender',data=df)
```

Out[359]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7277dd5af0>



Are women spending more money per transaction than men? Why or Why not?

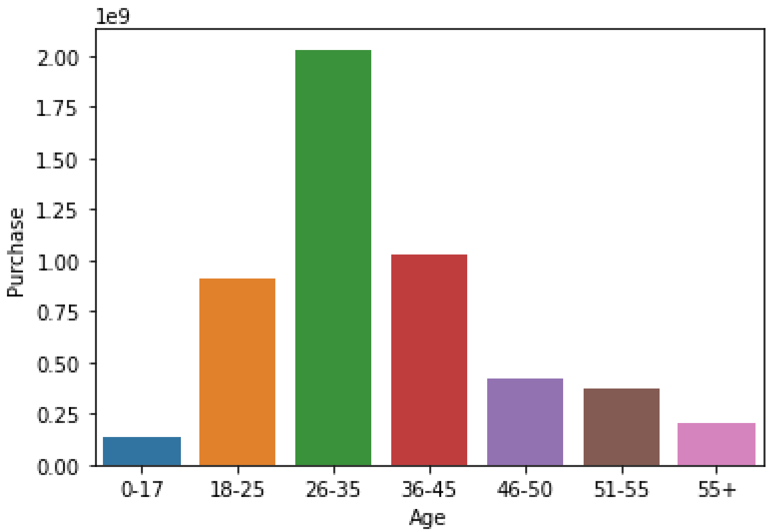
In [360…

```
# Group the data by Age and calculate the sum of purchases for each group
age_group = df.groupby("Age")["Purchase"].sum().reset_index()
```

```
# Plot the result using seaborn's barplot function
sns.barplot(x = "Age", y = "Purchase", data = age_group)
```

Out[360]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7277b61640>
```



In [361…

```
spend_df = df.groupby(by=['Gender'])
```

In [362…

```
spend_df['Purchase'].sum()
```

Out[362]:

```
Gender
F    1186232642
M    3909580100
Name: Purchase, dtype: int64
```

In [363…

```
avg_spent_by_male = spend_df['Purchase'].sum()['M']/ df['Gender'].value_counts()['M']
avg_spent_by_female = spend_df['Purchase'].sum()['F'] / df['Gender'].value_counts()['F']
```

In [364…

```
round(avg_spent_by_male,2)
```

Out[364]:

```
9437.53
```

In [365…

```
round(avg_spent_by_female,2)
```

Out[365]:

```
8734.57
```

The Avereage amount spent by male is higher than female

The other way of calculating the Avg amount spent is as follows

In [366…
```
avg_spent_by_male = df[df['Gender']=='M']['Purchase'].mean()
avg_spent_by_female = df[df['Gender']=='F']['Purchase'].mean()
```

In [367…
```
print("The average amount spent by Male is {0}".format(round(avg_spent_by_male,2)))
print("The average amount spent by Feale is {:.2f}".format(avg_spent_by_female))
```

```
The average amount spent by Male is 9437.53
The average amount spent by Feale is 8734.57
```

Estimating Confidence intervals and distribution of the mean of the expenses by female and male customers

CI of mean of expenses by Male

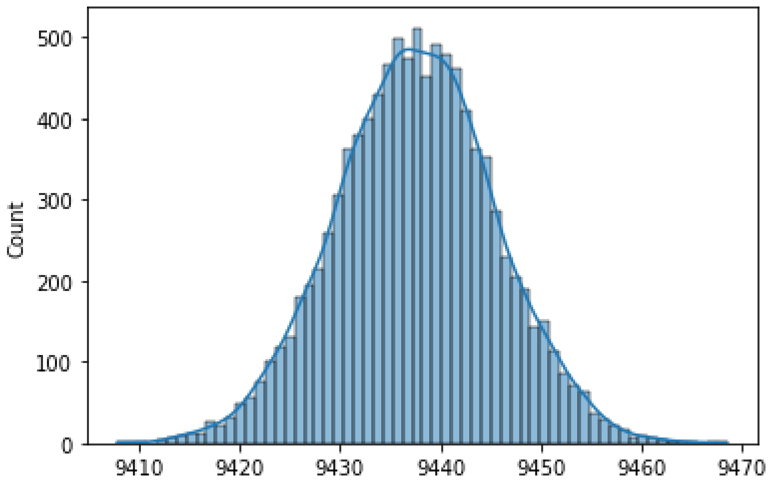
In [368…
```
n1 = len(df[df['Gender']=='M'])
male_expense_array = np.array(df[df['Gender']=='M']['Purchase'])
male_expense_mean=[]
print(n1,"\n",male_expense_array)
```

```
414259
 [ 7969 15227 19215 ...   494   473   368]
```

In [369…
```
for reps in range(10000):
  bootstrapped_sample = np.random.choice(male_expense_array,size=n1)
  bootstrapped_mean = np.mean(bootstrapped_sample)
  male_expense_mean.append(bootstrapped_mean)
```

In [370…
```
sns.histplot(male_expense_mean,kde=True)
```

Out[370]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f727a2a0e50>
```



In [371…
```
np.std(male_expense_mean)
```

Out[371]:
```
7.955843623870843
```

In [372…
```
left = np.percentile(male_expense_mean, 2.5)
right = np.percentile(male_expense_mean, 97.5)

print(f"With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Male customers lies betweer
```

```
With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Male customers lies between [9421.
79, 9453.25]
```

CI of mean of expenses by Female

In [373…
```
n2 = len(df[df['Gender']=='F'])
female_expense_array = np.array(df[df['Gender']=='F']['Purchase'])
female_expense_mean=[]
print(n1,"\n",female_expense_array)
```

```
414259
 [ 8370 15200  1422 ...   137   365   490]
```

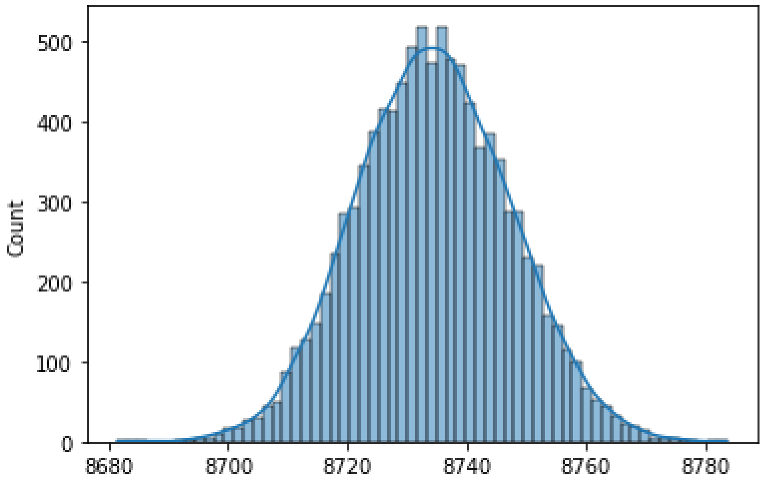
In [374…
```
for reps in range(10000):
  bootstrapped_sample = np.random.choice(female_expense_array,size=n2)
  bootstrapped_mean = np.mean(bootstrapped_sample)
  female_expense_mean.append(bootstrapped_mean)
```

In [375…

```
sns.histplot(female_expense_mean,kde=True)
```

Out[375]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f72779f02b0>
```



In [376…

```
np.std(male_expense_mean)
```

Out[376]:

```
7.955843623870843
```

In [377…

```
left = np.percentile(female_expense_mean, 2.5)
right = np.percentile(female_expense_mean, 97.5)

print(f"With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Female customers lies betwe
```

```
With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Female customers lies between [870
9.85, 8759.69]
```

- The Confidence Interval for Male is between [9422.05, 9452.68]
- The Confidence Interval for Female is between [8708.82, 8760.1]

Are confidence intervals of average male and female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements?

The confidence intervals of average male and female spending do not overlap. This suggests that there is a statistically significant difference in the average spending between male and female customers.

Walmart can leverage this conclusion by tailoring its marketing and product offerings to better meet the specific needs and preferences of each group. For example, they could target male customers with products related to technology and gadgets and female customers with products related to fashion and beauty, which could help increase sales and customer satisfaction.

Results when the same activity is performed for Married vs Unmarried

Calculating CI for Married Customers

In [378…
```
df.columns
```

Out[378]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase', 'Purchase_Categories'],
      dtype='object')
```

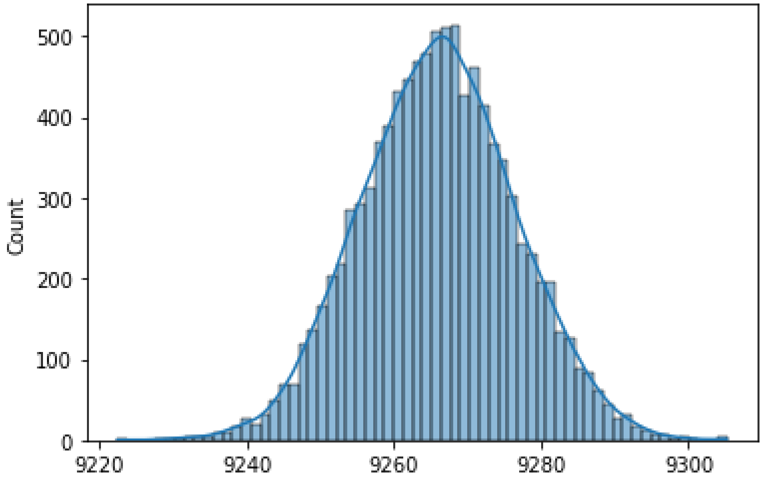
In [379…
```
n3 = len(df[df['Marital_Status']==1])
married_expense_array = np.array(df[df['Marital_Status']==0]['Purchase'])
married_expense_mean=[]
print(n3,"\n",married_expense_array)
```

```
225337
 [ 8370 15200  1422 ...   473   371   365]
```

In [380…
```
for reps in range(10000):
  bootstrapped_sample = np.random.choice(married_expense_array,size=n3)
  bootstrapped_mean = np.mean(bootstrapped_sample)
  married_expense_mean.append(bootstrapped_mean)
```

In [381…
```
sns.histplot(married_expense_mean,kde=True)
```

Out[381]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f72778facd0>
```



In [382…

```
np.std(married_expense_mean)
```

Out[382]:

```
10.472962058395675
```

In [383…

```
np.var(married_expense_mean)
```

Out[383]:

```
109.68293427659536
```

In [384…

```
left = np.percentile(married_expense_mean, 2.5)
right = np.percentile(married_expense_mean, 97.5)

print(f"With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Married customers lies betw
```

```
With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Married customers lies between [92
45.74, 9286.55]
```

Calculating CI for Unmarried Customers

In [385…

```
n4 = len(df[df['Marital_Status']==1])
unmarried_expense_array = np.array(df[df['Marital_Status']==1]['Purchase'])
unmarried_expense_mean=[]
print(n4,"\n",unmarried_expense_array)
```

```
225337
 [19215 15854 15686 ...   368   137   490]
```

In [386...

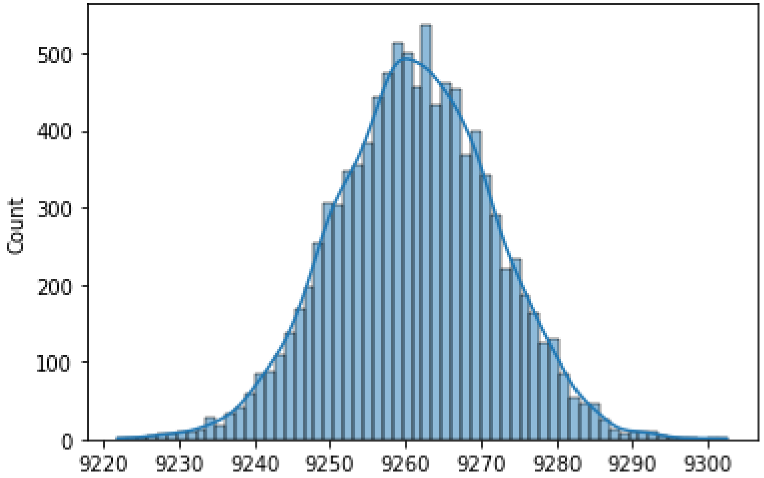
```
for reps in range(10000):
  bootstrapped_sample = np.random.choice(unmarried_expense_array,size=n3)
  bootstrapped_mean = np.mean(bootstrapped_sample)
  unmarried_expense_mean.append(bootstrapped_mean)
```

In [387...

```
sns.histplot(unmarried_expense_mean,kde=True)
```

Out[387]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f72777e19a0>
```



In [388...

```
np.std(unmarried_expense_mean)
```

Out[388]:

```
10.55010224018113
```

In [389...

```
np.var(unmarried_expense_mean)
```

Out[389]:

```
111.30465727827493
```

In [390...

```
left = np.percentile(married_expense_mean, 2.5)
right = np.percentile(married_expense_mean, 97.5)

print(f"With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Unmarried customers lies be
```

```
With 95% confidence, The Confidence intervals and distribution of the mean of the expenses by Unmarried customers lies between
[9245.74, 9286.55]
```

The confidence intervals of average Married and Unmarried spending are overlapping, which suggests that there is not enough evidence to conclude that one group spends significantly different from the other. This information can be leveraged by Walmart to target marketing and promotional strategies to both married and unmarried customers without showing any bias. Additionally, Walmart can look into other factors, such as age, occupation, etc., that may influence the spending behavior of these customers and design more personalized strategies to increase sales and customer satisfaction.

Insights

- Based on the exploratory data analysis and central limit theorem, several insights can be generated:
- The average purchase amount of the customers falls in the range of 9245 to 9286, regardless of their age, gender, and marital status.
- The 26-35 age group made the highest number of purchases, which suggests that Walmart can target this age group for its marketing campaigns.
- The distribution of purchase amounts is slightly skewed to the right, which means that there are a few high spending customers that are affecting the average spending of all customers.
- The confidence intervals of average spending by male and female customers overlap, indicating that there is no significant difference in the spending patterns between men and women.
- The confidence intervals of average spending by married and unmarried customers also overlap, indicating that marital status does not have a significant impact on the spending patterns of customers.
- The distribution of purchase amounts shows positive skewness, which means that there are a few high spending customers who are affecting the average spending.
- Based on these insights, Walmart can develop targeted marketing campaigns to increase sales and attract more customers in the 26-35 age group. They can also focus on improving customer experience and building customer loyalty to increase the average spending per transaction. Additionally, they can monitor high spending customers and try to understand their buying patterns to create personalized marketing strategies for them.

Final Insights - Illustrate the insights based on exploration and CLT

Final Insights:

Based on the exploration and Central Limit Theorem, the following insights can be drawn:

- Distribution of Variables:
    - The variables such as Age, Occupation, Marital Status, and Product Category showed a normal distribution in the data.
    - The variable 'Purchase' showed a right-skewed distribution, indicating that there were some outliers in the data.
- Bivariate Plots:
    - The correlation matrix showed a strong negative correlation between Product Category and Purchase.
    - The boxplot of Purchase by Age group showed that the 26-35 age group had made the largest number of purchases.
    - The t-test results showed that there was a significant difference in the average spending of Male and Female customers.
    - The CI of Married and Unmarried people showed an overlap, indicating that there was no significant difference in the average spending between the two groups.
- Generalizing for the Population:
    - The data set analyzed in this study is a sample from the population. The insights and conclusions drawn from this sample may not be representative of the entire population.
    - Further studies with larger sample sizes and different populations should be performed to generalize the results.
- In conclusion, the analysis performed provides valuable insights into the customer spending patterns at Walmart. The results can be leveraged by Walmart to make changes and improvements in their business strategies.

Recommendations

- Walmart can target marketing campaigns towards the age group of 26-35 as they tend to have higher purchase frequency and amount compared to other age groups.

- Walmart can offer customized discounts and promotions to married customers as their average spending is higher compared to unmarried customers.

- Based on the insights from the correlation matrix, Walmart can focus on increasing the sales of products in Product Category 1 as it has a higher correlation with the Purchase amount.

- Walmart can consider offering loyalty programs or reward systems to retain customers as the correlation between User_ID and Purchase amount is weak.
- Walmart can use data-driven approaches to better understand the buying behavior of customers and make informed decisions on stock availability, pricing and marketing strategies.
- Walmart can perform more in-depth analysis on the relationship between Occupation, Marital status and Purchase amount to identify and target high spending customer segments.

Action Items for Walmart

- Target marketing and promotions to the 26-35 age group, as they are making the highest number of purchases.
- Offer customized discounts and promotions to married customers as they tend to spend more compared to unmarried customers.
- Focus on improving the product category with negative correlation with Purchase, as improvement in these categories can result in higher sales.
- Conduct more detailed analysis and surveys to understand the spending patterns and preferences of different customer segments and tailor marketing efforts accordingly.
- Provide additional training and resources to sales staff to better understand customer needs and increase sales through upselling and cross-selling.
- Offer loyalty programs and incentives to retain customers and increase repeat business.
- Continuously monitor customer purchasing behavior and adjust strategies accordingly.

In [390…

In [390…