Business Case: Walmart - Confidence Interval and CLT

Problem Statement

Help Walmart make better business decisions by analysing customer purchase behavior (specifically, purchase amount) against the customer's gender and the various other factors. They want to understand if the spending habits differ between male and female customers: Do women spend more on Black Friday than men?

Importing Python Libraries necessary to carry out data exploration & visualisation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
walmart = pd.read_csv("Walmart.csv")
```

Dataset Description

Dataset Consists of:

- 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

Inspecting Dataset & Analyzing Different Metrics

```
walmart.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Sta
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	

```
walmart.tail()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marita
550063	1006033	P00372445	M	51-55	13	B	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	
550066	1006038	P00375436	F	55+	1	C	2	
550067	1006039	P00371644	F	46-50	0	B	4+	

Observations on

- 1) shape of data 2) data types 3) Statistical summary

```
walmart.shape
```

```
(550068, 10)
```

```
walmart.columns
```

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

```
walmart.size
```

```
5500680
```

```
walmart.dtypes
```

```
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
```

```
walmart.info
```

```
<bound method DataFrame.info of         User_ID Product_ID Gender    Age  Occupation City_Category  \
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

       Stay_In_Current_City_Years  Marital_Status  Product_Category  Purchase
0                               2               0                 3      8370
1                               2               0                 1     15200
2                               2               0                12      1422
3                               2               0                12      1057
4                              4+               0                 8      7969
...                           ...             ...               ...       ...
```

```
550063                  1          1           20       368
550064                  3          0           20       371
550065                 4+          1           20       137
550066                  2          0           20       365
550067                 4+          1           20       490

[550068 rows x 10 columns]>
```

```
walmart.describe()
```

	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

```
walmart.describe(include=object)
```

	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

Data Cleaning : Optional Treatment

```
walmart.isnull().sum().sort_values(ascending =False)
```

```
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 doesn't have any null values so data cleaning is not required

Mean and Median

```
walmart.mean()
```

```
<ipython-input-15-c61f0c8f89b5>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future ve
  df.mean()
User_ID             1.003029e+06
Occupation          8.076707e+00
Marital_Status      4.096530e-01
Product_Category    5.404270e+00
Purchase            9.263969e+03
dtype: float64
```

```
walmart.median()
```

```
<ipython-input-16-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future
  df.median()
User_ID             1003077.0
Occupation                7.0
Marital_Status            0.0
```

```
Product_Category          5.0
Purchase               8047.0
dtype: float64
```

Inference: Difference between Mean and Median is not significant

Check the characteristics of the data

```
walmart.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Sta
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	

```
walmart['Product_ID'].value_counts()
```

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

There are 3417 unique products

```
walmart['Age'].value_counts()
```

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

There are 7 different age bins

Age wise unique count & value count

```
walmart["Age"].unique()
```

```
array(['0-17', '55+', '26-35', '46-50', '51-55', '36-45', '18-25'],
      dtype=object)
```

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

```
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
```

Inference: There are 21 unique occupations in the dataset

Gender wise unique count & value count

```
walmart["Gender"].unique()
```

```
array(['F', 'M'], dtype=object)
```

```
walmart["Gender"].value_counts()
```

```
M    113676
F     36579
Name: Gender, dtype: int64
```

Inference: We have more than 2.5 times male customers compared to females in the dataset.

Marital Status unique count & value count

```
walmart["Marital_Status"].unique()
```

```
array([0, 1])
```

```
walmart["Marital_Status"].value_counts()
```

```
0    88831
1    61424
Name: Marital_Status, dtype: int64
```

```
walmart["Marital_Status"].value_counts(normalize=True).round(2)*100
```

```
0.0    59.0
1.0    41.0
Name: Marital_Status, dtype: float64
```

Inference: The data contains 59% single customers and 41% married people.

```
#The below code proves that marital status code 0 is for single and 1 is for married
walmart.loc[walmart["Age"]=="0-17"]["Marital_Status"].value_counts()
```

```
0.0    2804
Name: Marital_Status, dtype: int64
```

```
walmart.loc[walmart["Age"]=="55+"]["Marital_Status"].value_counts()
```

```
1.0    2525
0.0    1406
Name: Marital_Status, dtype: int64
```

```
walmart["Marital_Status"] = walmart["Marital_Status"].apply(lambda x : "Married" if x==1 else "Single")
```

```
walmart.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Sta
0	1000001	P00069042	F	0-17	10.0	A	2	Si

Stay_In_Current_City unique count & value count

```
walmart.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Sta
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	

```
walmart["Stay_In_Current_City_Years"].unique()
```

```
array(['2', '4+', '3', '1', '0', nan], dtype=object)
```

```
walmart["Stay_In_Current_City_Years"].value_counts()
```

```
1     34949
2     18598
3     17532
4+    15539
0     13556
Name: Stay_In_Current_City_Years, dtype: int64
```

```
walmart["Stay_In_Current_City_Years"].value_counts(normalize=True).round(2)*100
```

```
1     35.0
2     19.0
3     18.0
4+    16.0
0     14.0
Name: Stay_In_Current_City_Years, dtype: float64
```

Inference: Mostly users use the threadmill 2-4 days a week. Only 5% of the customers use it 6-7 days a week.

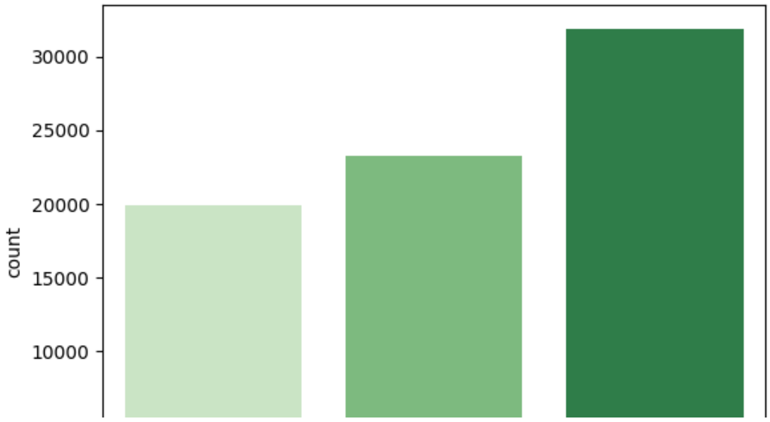
Checking the categories of cities in the dataset

```
walmart.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Sta
0	1000001	P00069042	F	0-17	10.0	A	2	
1	1000001	P00248942	F	0-17	10.0	A	2	
2	1000001	P00087842	F	0-17	10.0	A	2	
3	1000001	P00085442	F	0-17	10.0	A	2	
4	1000002	P00285442	M	55+	16.0	C	4+	

Univariate Analysis using countplots

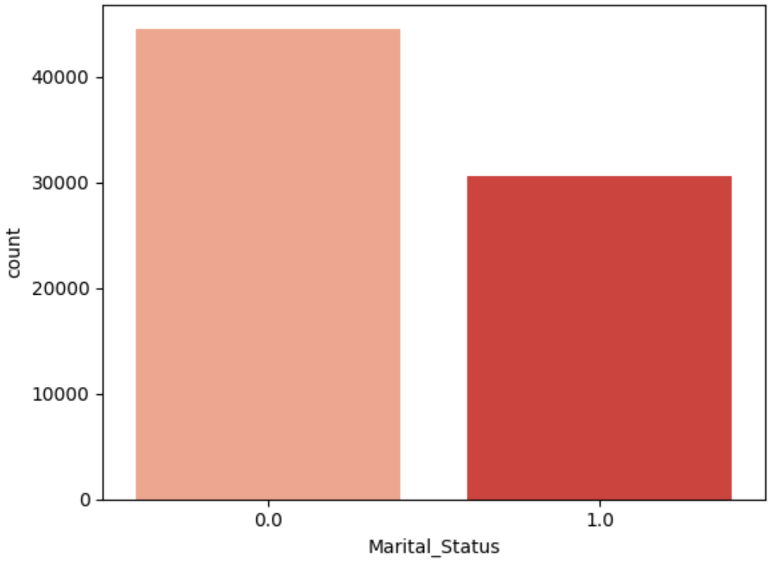
```
# Product countplot
sns.countplot(x = "City_Category", data = walmart, palette = "Greens")
plt.show()
```



Inference: Most number of cities fall in category B followed by C. Least number of cities are in category A.

Checking the data marital status wise.

```
sns.countplot(x = "Marital_Status", data = walmart, palette = "Reds")
plt.show()
```



Inference: Single people are higher in number in this dataset compared to married people.

Checking data age wise

```
plt.figure(figsize = (8,5))
sns.countplot(x = "Age", data = walmart, palette = "twilight")
plt.xticks(rotation = 90)
plt.show()
```

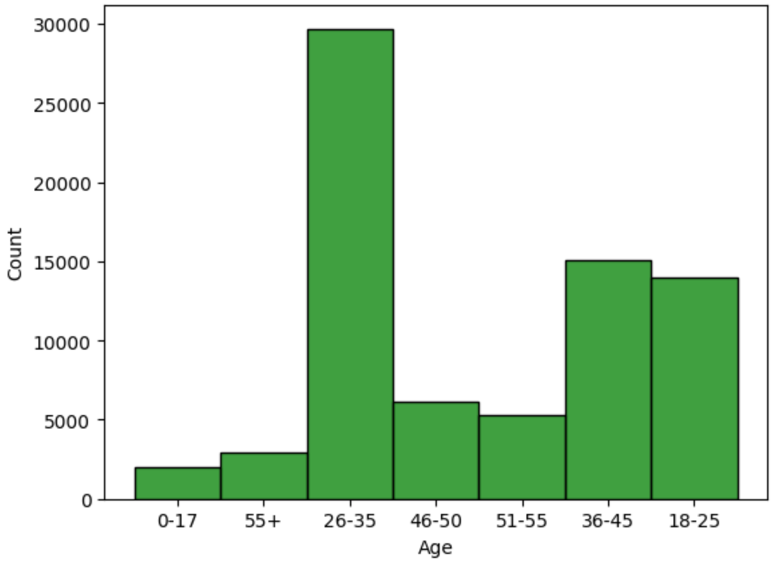
Inference : Most buyers fall in the age group of 26-35 (around 38k). The next age groups are 36-45 and 18-25. This seems natural as people in the age 26-35 group are most active in general and include more people with families.

```
walmart.head()
```

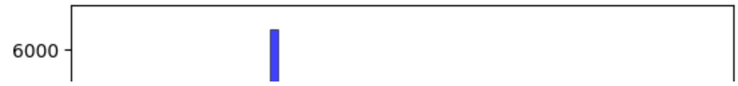
	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purch
0	1000001	P00069042	F	0-17	10.0	A	2	Single	3.0	837
1	1000001	P00248942	F	0-17	10.0	A	2	Single	1.0	1520
2	1000001	P00087842	F	0-17	10.0	A	2	Single	12.0	142
3	1000001	P00085442	F	0-17	10.0	A	2	Single	12.0	105

Univariate Analysis - Check different columns using histograms

```
sns.histplot(walmart["Age"], color = "g")
plt.show()
```

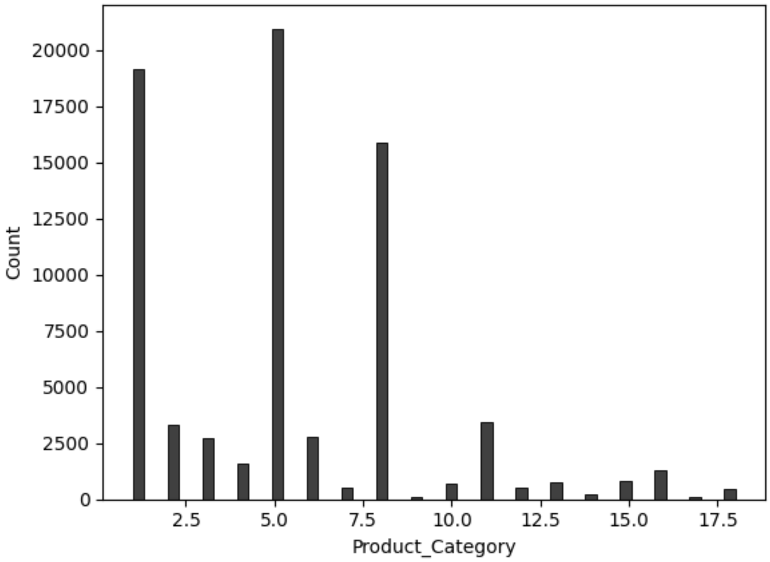



```
sns.histplot(walmart["Purchase"], color = "b")
plt.show()
```



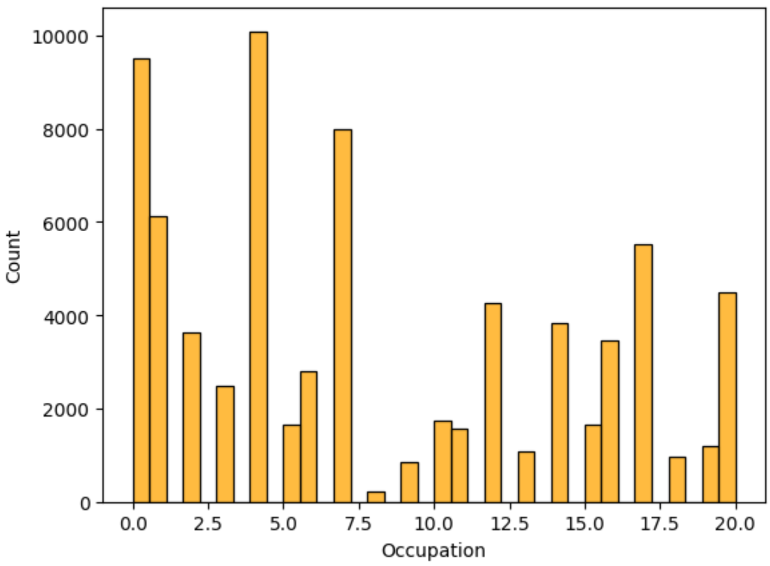
Inference: From the above graph, it can be concluded that the most number of purchases are between 5000 dollars and 10000 dollars

```
sns.histplot(walmart["Product_Category"], color = "k")
plt.show()
```



Inference: From the above graph, it is evident that the product category number 5, 1 and 8 are the most sold products in descending order of sales.

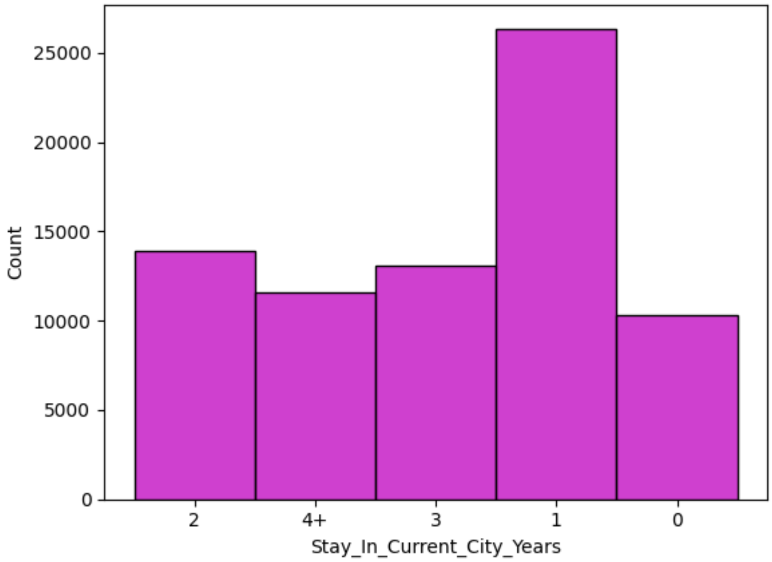
```
sns.histplot(walmart["Occupation"], color = "orange")
plt.show()
```




```
walmart.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purch
0	1000001	P00069042	F	0-17	10.0	A	2	0.0	3.0	837
1	1000001	P00248942	F	0-17	10.0	A	2	0.0	1.0	1520
2	1000001	P00087842	F	0-17	10.0	A	2	0.0	12.0	142
3	1000001	P00085442	F	0-17	10.0	A	2	0.0	12.0	105

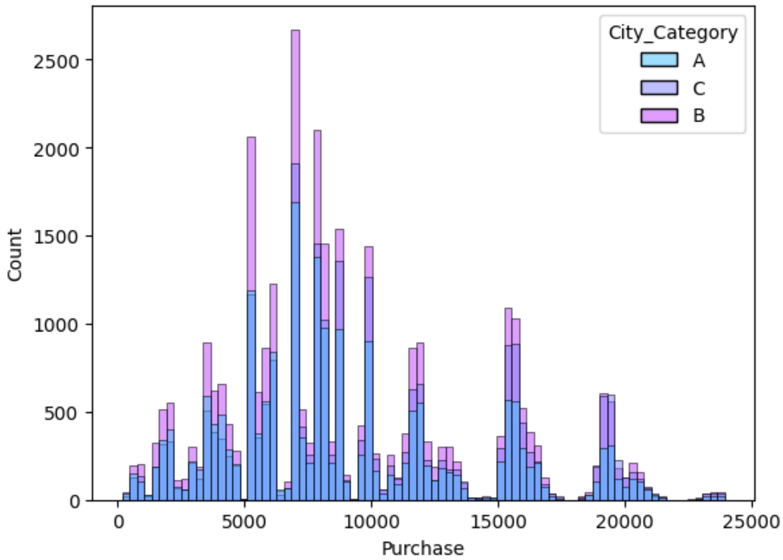
```
sns.histplot(walmart["Stay_In_Current_City_Years"], color = "m")
plt.show()
```



Inference: The maximum number of people in the dataset have stayed in their current city for one year (approximately 35k). The distribution for the rest of the options(0,2,3,4+) is close.

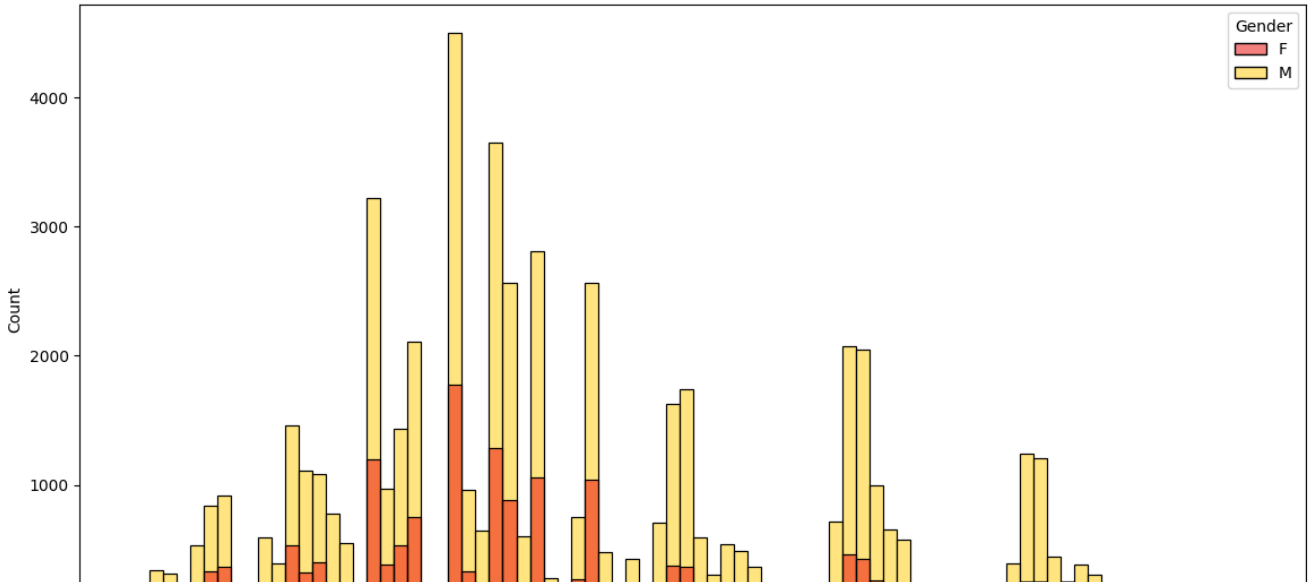
Bivariate Analysis

```
# City_Category wise Purchase
sns.histplot(x = "Purchase", hue = "City_Category", data = walmart, palette = "cool")
plt.show()
```



Inference: Purchasing power seems to be highest for people in category B cities followed by category C and then category A.

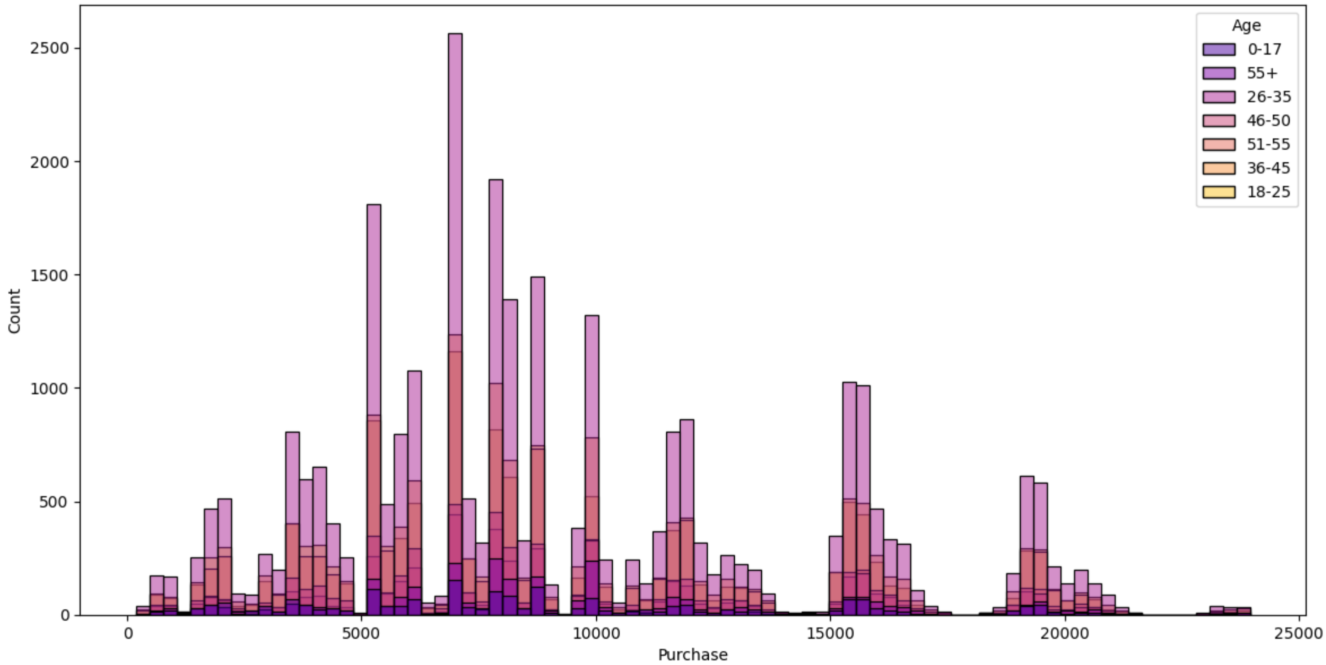
```
# Purchase with respect to gender -
plt.figure(figsize = (14, 7))
sns.histplot(x = "Purchase", data = walmart, hue = "Gender", palette = "hot")
plt.show()
```



Inference:

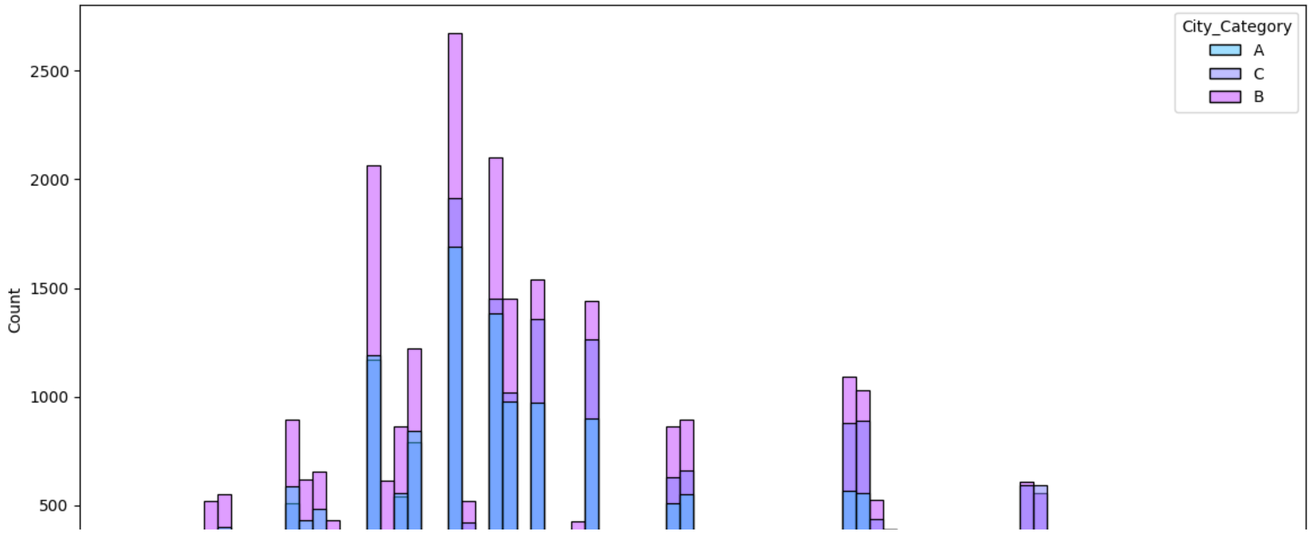
Males purchase products more often than women from Walmart.

```
# Purchase with respect to Age -
plt.figure(figsize = (14, 7))
sns.histplot(x = "Purchase", data = walmart, hue = "Age", palette = "plasma")
plt.show()
```



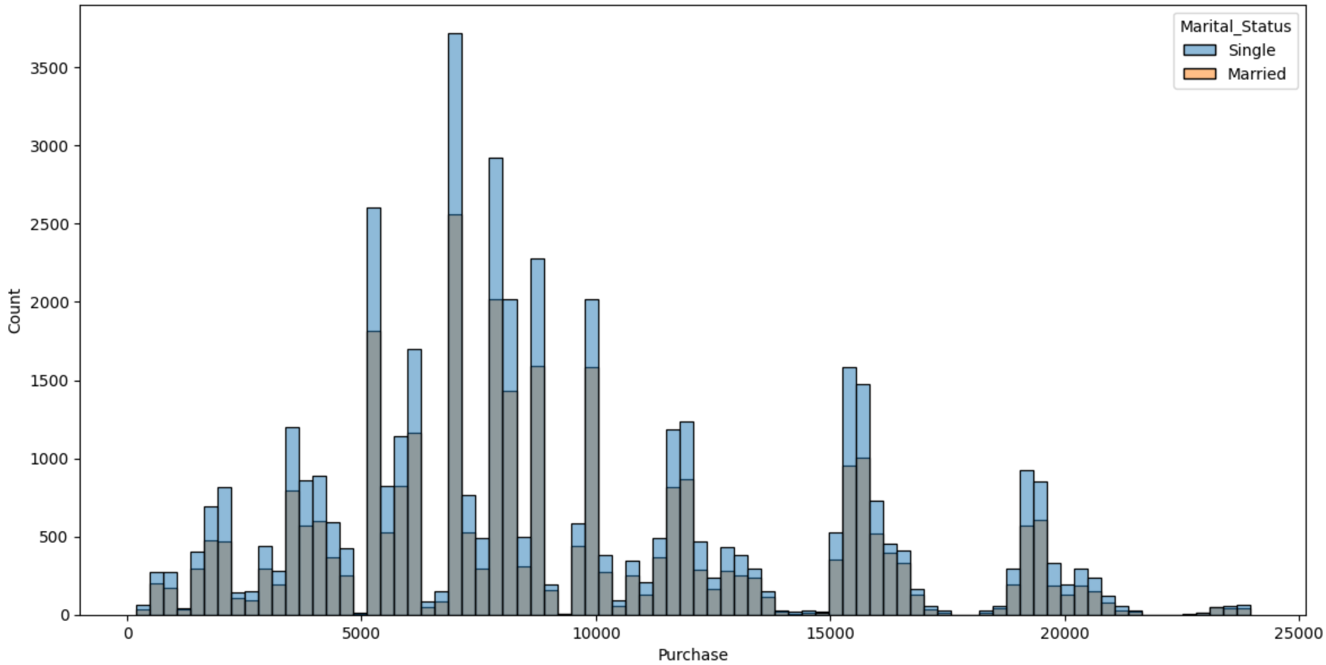
Inference: The age group between 26-35 is the biggest consumer.

```
# Purchase with respect to City Category -
plt.figure(figsize = (14, 7))
sns.histplot(x = "Purchase", data = walmart, hue = "City_Category", palette = "cool")
plt.show()
```



Inference: Purchase from City category 'B' is highest, followed by 'C' and then 'A'.

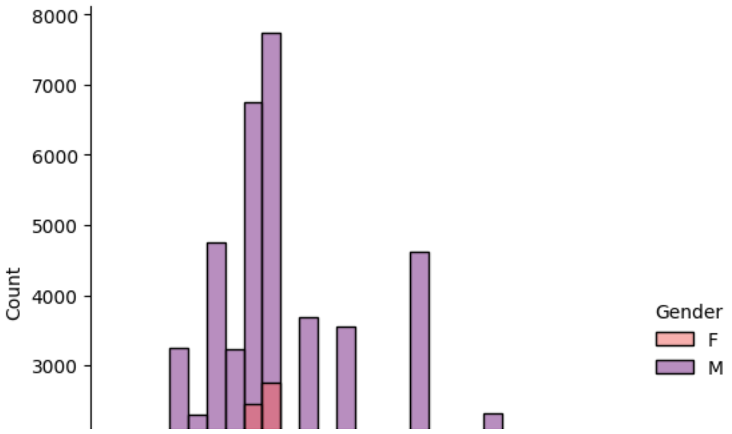
```
# Purchase with respect to Marital Status -
plt.figure(figsize = (14, 7))
sns.histplot(x = "Purchase", data = walmart, hue = "Marital_Status")
plt.show()
```



Inference: Single people have more tendency to purchase than married people.

Displots

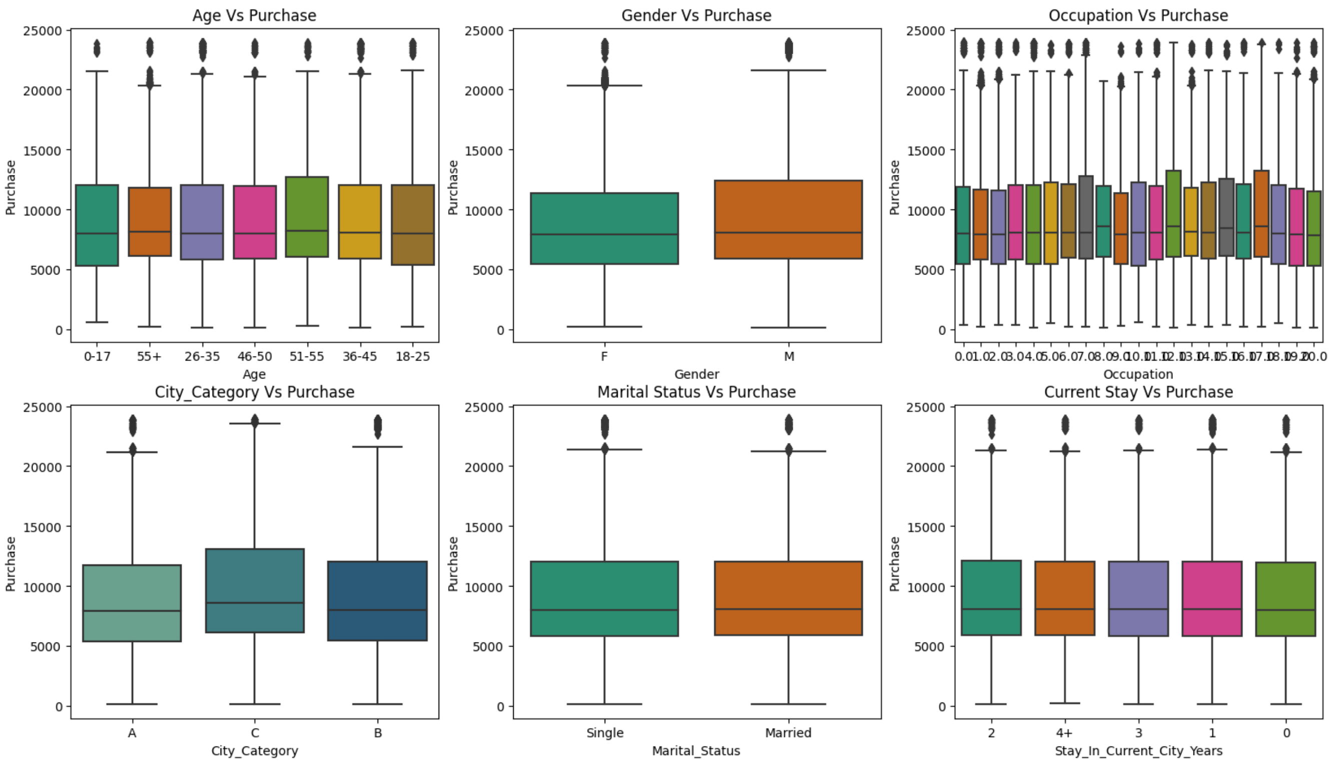
```
sns.displot(x = "Purchase", hue = "Gender", data = walmart, bins = 25, palette = "magma_r" )
plt.xticks( rotation = 90)
plt.show()
```



Inference: Males are purchasing more as compared to females.

Boxplots

```
# Various Factors vs Purchase (Gender, Age, Occupation, City Category, Current city stay, Marital Status)
plt.figure(figsize = (18, 10))
plt.subplot(2, 3, 1)
sns.boxplot(x = "Age", y = "Purchase", data = walmart, palette = "Dark2")
plt.title("Age Vs Purchase", fontsize = 12)
plt.subplot(2, 3, 2)
sns.boxplot(x = "Gender", y = "Purchase", data = walmart, palette = "Dark2")
plt.title("Gender Vs Purchase", fontsize = 12)
plt.subplot(2, 3, 3)
sns.boxplot(x = "Occupation", y = "Purchase", data = walmart, palette = "Dark2")
plt.title("Occupation Vs Purchase", fontsize = 12)
plt.subplot(2, 3, 4)
sns.boxplot(x = "City_Category", y = "Purchase", data = walmart, palette = "crest")
plt.title("City_Category Vs Purchase", fontsize = 12)
plt.subplot(2, 3, 5)
sns.boxplot(x = "Marital_Status", y = "Purchase", data = walmart, palette = "Dark2")
plt.title("Marital Status Vs Purchase", fontsize = 12)
plt.subplot(2, 3, 6)
sns.boxplot(x = "Stay_In_Current_City_Years", y = "Purchase", data = walmart, palette = "Dark2")
plt.title("Current Stay Vs Purchase", fontsize = 12)
plt.show()
```



Inference:

1) There is a slight difference in the median purchase of male and female. (slightly higher for male)

2) Median purchase of every age group is nearly similar.

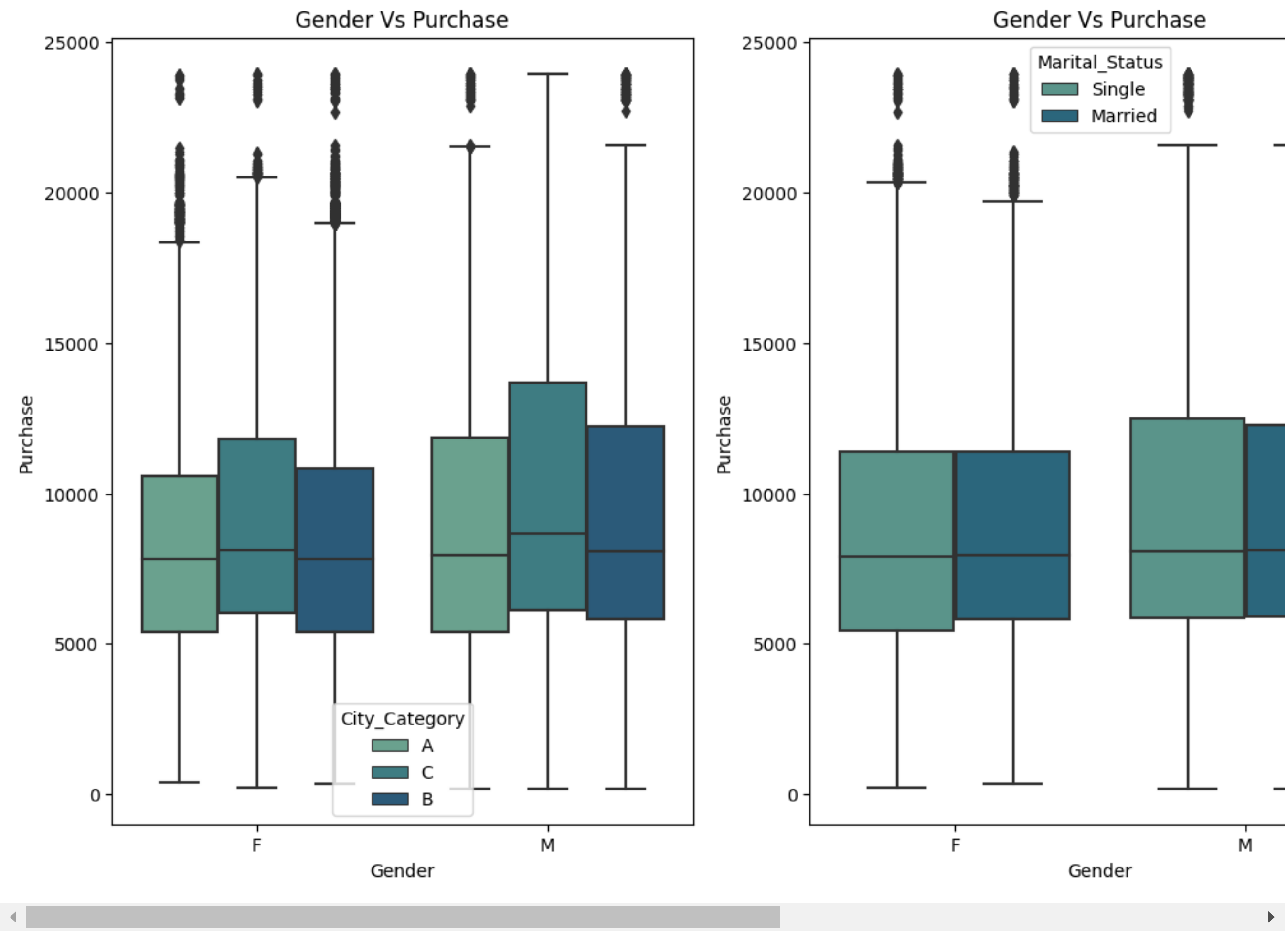
3) Median purchase of Occupational experience 12, 15 & 17 years are the highest among all occupational experience groups.

4) Median purchase for City Category 'C' is more than the other city categories.

5) Median purchase for all current city stay is nearly equal.

6) Median purchase is almost equal for single and married people.

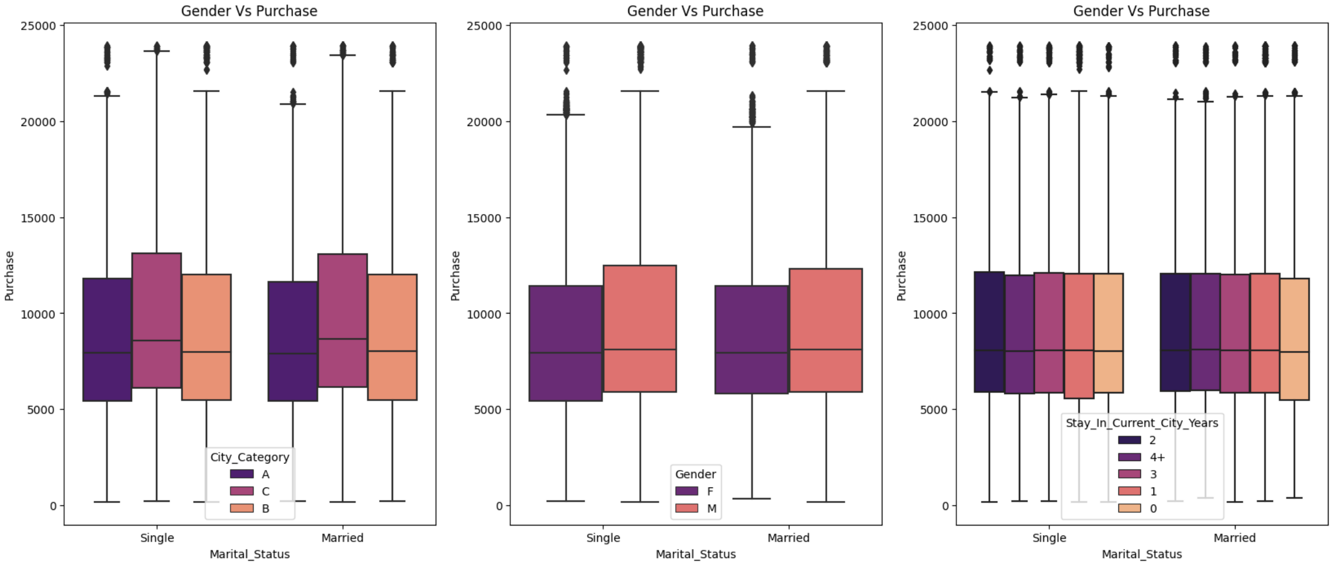
```
# Gender vs Purchase (With hue as City Category, Marital Status, Current city stay)
plt.figure(figsize = (20, 8))
plt.subplot(1, 3, 1)
sns.boxplot(data = walmart, x = "Gender", y = "Purchase", hue = "City_Category", palette = "crest")
plt.title("Gender Vs Purchase", fontsize = 12)
plt.subplot(1, 3, 2)
sns.boxplot(data = walmart, x = "Gender", y = "Purchase", hue = "Marital_Status", palette = "crest")
plt.title("Gender Vs Purchase", fontsize = 12)
plt.subplot(1, 3, 3)
sns.boxplot(data = walmart, x = "Gender", y = "Purchase", hue = "Stay_In_Current_City_Years", palette = "crest")
plt.title("Gender Vs Purchase", fontsize = 12)
plt.show()
```



Inference: Irrespective of marital status, city category & current stay city, male customers are higher purchasers of the product as compared to female customers. This may also because of the fact that there are more males than females in the dataset.

```
# Marital Status vs Purchase (With hue as City Category, Gender, Years in current city)
plt.figure(figsize = (20, 8))
plt.subplot(1, 3, 1)
sns.boxplot(data = walmart, x = "Marital_Status", y = "Purchase", hue = "City_Category", palette = "magma")
plt.title("Gender Vs Purchase", fontsize = 12)
plt.subplot(1, 3, 2)
sns.boxplot(data = walmart, x = "Marital_Status", y = "Purchase", hue = "Gender", palette = "magma")
plt.title("Gender Vs Purchase", fontsize = 12)
plt.subplot(1, 3, 3)
sns.boxplot(data = walmart, x = "Marital_Status", y = "Purchase", hue = "Stay_In_Current_City_Years", palette = "magma")
```

```
plt.title("Gender Vs Purchase", fontsize = 12)
plt.show()
```

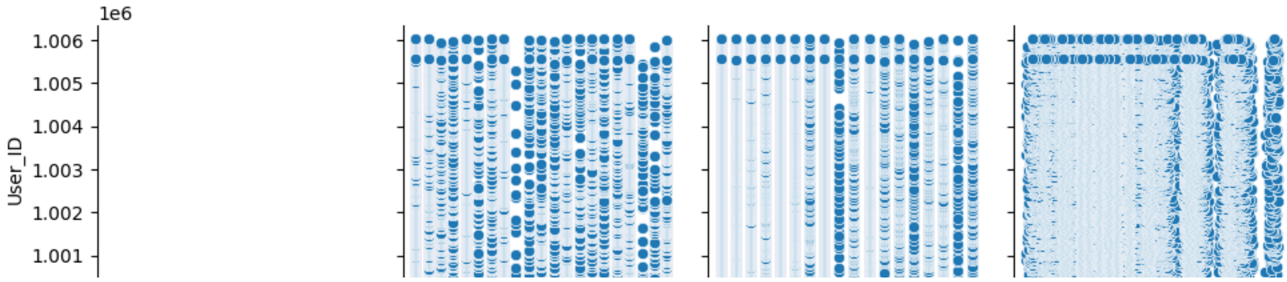


Inference: Purchase amount for both single & partnered customers are nearly same.

Multivariate Analysis

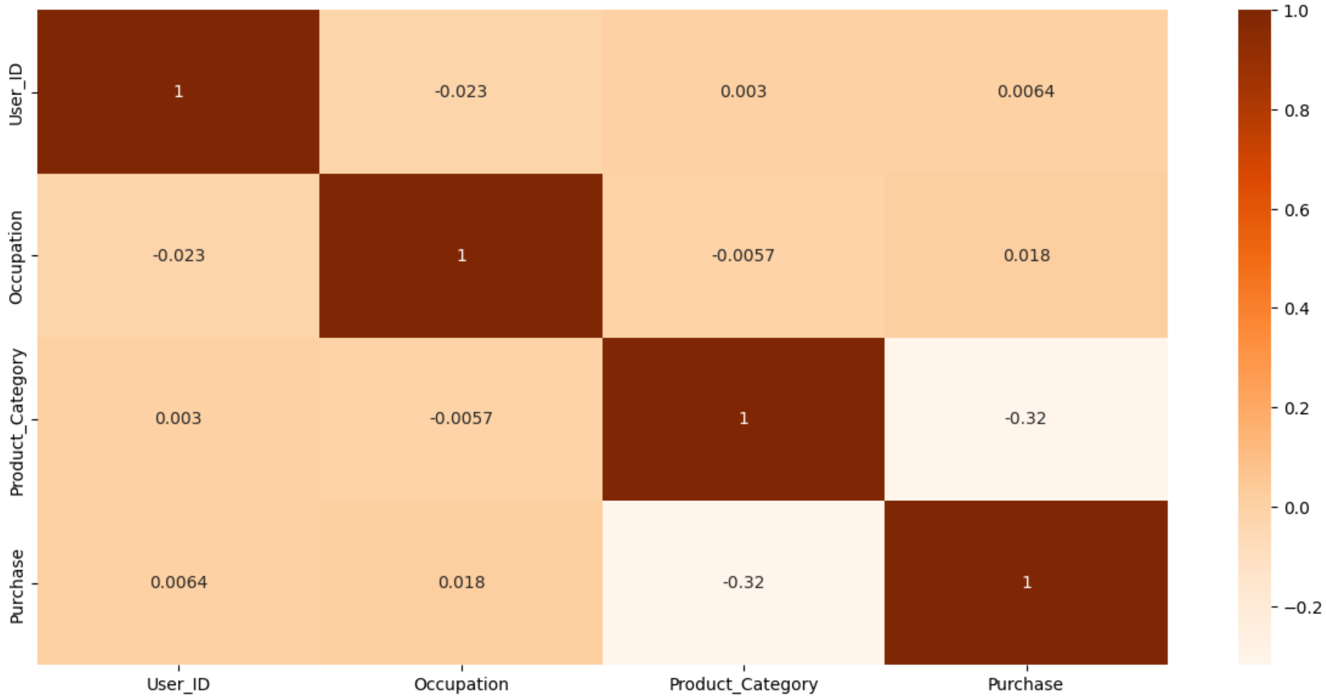
```
# Pair plots -
plt.figure(figsize = (12, 10))
sns.pairplot(walmart)
plt.show()
```

```
<Figure size 1200x1000 with 0 Axes>
```




```
# Heatmaps -
plt.figure(figsize = (15, 7))
sns.heatmap(walmart.corr(), annot = True, cmap = "Oranges")
plt.show()
```

```
<ipython-input-40-8a71837090ed>:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
  sns.heatmap(walmart.corr(), annot = True, cmap = "Oranges")
```



Inference:

No positive or negative correlations can be seen from above pair plots & heatmaps.

Confidence Interval Analysis

```
samp = walmart.sample(500)
samp
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	P
3315	1000541	P00057542	F	18-25	4.0	C	3	Single	3.0	
39845	1000133	P00032842	F	26-35	0.0	C	1	Married	8.0	

Gender Analysis

```
# Overall Mean -
walmart[walmart["Gender"] == "M"]["Purchase"].mean()
```

    9464.0596711069

```
# Overall Mean -
walmart[walmart["Gender"] == "F"]["Purchase"].mean()
```

    8780.111166198425

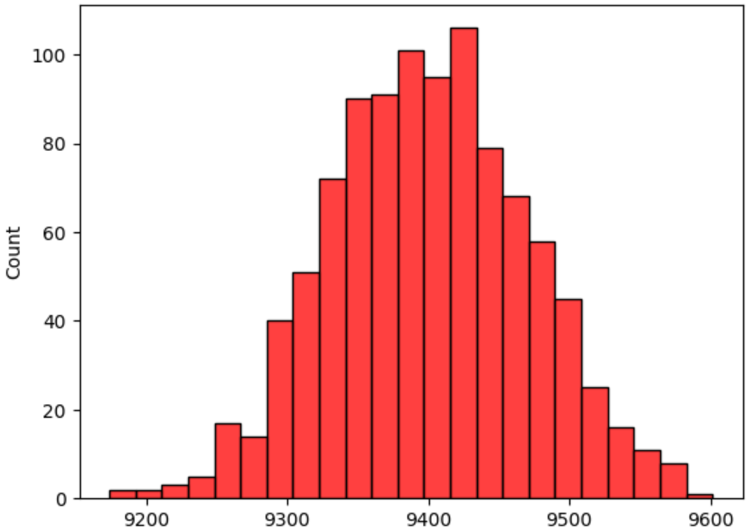
```
# Sample Statistical Properties -
samp.groupby("Gender")["Purchase"].describe()
```

Gender	count	mean	std	min	25%	50%	75%	max
F	135.0	9368.748148	4963.383893	562.0	5849.0	8320.0	12076.0	23847.0
M	365.0	9402.635616	4981.618889	734.0	6018.0	8064.0	12476.0	23802.0

```
male_samp_mean = [samp[samp["Gender"] == "M"].sample(5000, replace = True)["Purchase"].mean() for i in range(1000)]
male_samp_mean
```

```
        9311.9796,
        9403.8516,
        9309.8996,
        9399.5526,
        9424.9988,
        9366.0772,
        9459.6196,
        9434.0346]
```

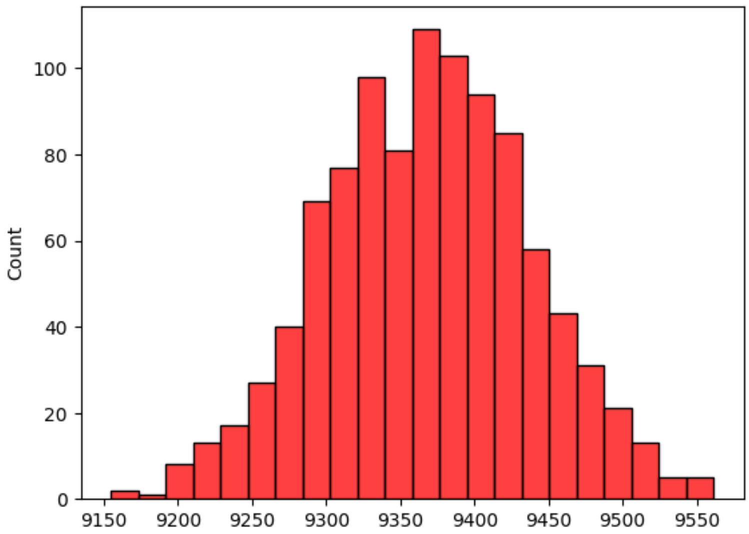
```
sns.histplot(male_samp_mean, color = "r")
plt.show()
```




```
female_samp_mean = [samp[samp["Gender"] == "F"].sample(5000, replace = True)["Purchase"].mean() for i in range(1000)]
female_samp_mean
```

```
    9408.2074,
    9366.0858,
    9295.9866,
    9344.6306,
    9234.3772,
    9374.4236,
    9223.3358,
    9414.4006,
    9390.739]
```

```
sns.histplot(female_samp_mean, color = "r")
plt.show()
```




```
# Std deviation of male sample
np.std(male_samp_mean).round(3)
```

```
69.97
```

```
# Std deviation of female sample
np.std(female_samp_mean).round(3)
```

```
69.114
```

CI - 90%

```
from scipy.stats import norm
```

```
# Confidence Interval of male = 90%
male_low = np.mean(male_samp_mean) + norm.ppf(0.05) * (np.std(male_samp_mean))
male_high = np.mean(male_samp_mean) + norm.ppf(0.95) * (np.std(male_samp_mean))
male_low.round(3), male_high.round(3)
```

```
(9285.095, 9515.277)
```

```
# Confidence Interval of female = 90%
female_low = np.mean(female_samp_mean) + norm.ppf(0.05) * (np.std(female_samp_mean))
female_high = np.mean(female_samp_mean) + norm.ppf(0.95) * (np.std(female_samp_mean))
female_low.round(3), female_high.round(3)
```

```
(9254.496, 9481.861)
```

```
# To check overlapping of Confidence Intervals
male_CI = np.percentile(male_samp_mean, [5, 95])
female_CI = np.percentile(female_samp_mean, [5, 95])
male_CI.round(3), female_CI.round(3)
```

```
(array([9291.36 , 9516.801]), array([9256.702, 9481.716]))
```

Inference: From the above results, it is clear that confidence intervals of male & female average purchases are not overlapping for 90% CI.

CI - 95%

```
# Confidence Interval of male = 95%
male_low = np.mean(male_samp_mean) + norm.ppf(0.025) * (np.std(male_samp_mean))
male_high = np.mean(male_samp_mean) + norm.ppf(0.975) * (np.std(male_samp_mean))
male_low.round(3), male_high.round(3)
```

```
(9263.047, 9537.326)
```

```
# Confidence Interval of female = 95%
female_low = np.mean(female_samp_mean) + norm.ppf(0.025) * (np.std(female_samp_mean))
female_high = np.mean(female_samp_mean) + norm.ppf(0.975) * (np.std(female_samp_mean))
female_low.round(3), female_high.round(3)
```

```
(9232.718, 9503.64)
```

```
# To check overlapping of Confidence Intervals
male_CI = np.percentile(male_samp_mean, [2.5, 97.5])
female_CI = np.percentile(female_samp_mean, [2.5, 97.5])
male_CI.round(3), female_CI.round(3)
```

```
(array([9263.378, 9538.21 ]), array([9231.055, 9503.228]))
```

Inference: From the above results, it is clear that confidence intervals of male & female average purchases are not overlapping for 95% CI.

CI - 99%

```
# Confidence Interval of male = 99%
male_low = np.mean(male_samp_mean) + norm.ppf(0.005) * (np.std(male_samp_mean))
male_high = np.mean(male_samp_mean) + norm.ppf(0.995) * (np.std(male_samp_mean))
male_low.round(3), male_high.round(3)
```

```
(9219.955, 9580.418)
```

```
# Confidence Interval of female = 95%
female_low = np.mean(female_samp_mean) + norm.ppf(0.005) * (np.std(female_samp_mean))
female_high = np.mean(female_samp_mean) + norm.ppf(0.995) * (np.std(female_samp_mean))
female_low.round(3), female_high.round(3)
```

```
(9190.153, 9546.205)
```

```
# To check overlapping of Confidence Intervals
male_CI = np.percentile(male_samp_mean, [0.5, 99.5])
female_CI = np.percentile(female_samp_mean, [0.5, 99.5])
male_CI.round(3), female_CI.round(3)
```

```
(array([9217.019, 9573.233]), array([9195.924, 9536.827]))
```

Inference: From the above results, it is clear that confidence intervals of male & female average purchases are not overlapping for 99% CI.

From above analysis, it is evident that males are purchasing more for different confidence intervals as compared to females

Insights from Data

1) 59% Single, 41% Married.

2) 75% of the users are Male and 25% are Female.

3) Nearly 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45).

4) Total of 20 product categories are there.

5) There are 20 different types of occupations in the city.

6) Customers mostly from city B(42%) followed by city C(31%) & then city A(27%).

7) 35% Staying in the city for the last 1 year, 18% for the last 2 years, 17% for the last 3 years.

8) From CLT graphs we have noticed that for gender samples, the confidence interval range was not overlapping.

Recommendations:

1) Unmarried customers spend more money than married customers. Therefore, in order to enable married people to buy more, walmart may consider curating offers targetted towards married people.

2) As males are purchasing more as compared to females, retaining male customers while deducing instruments to invite more females to purchase is advised.

3) Customers in the age group of 18-25 is the most favourable age range for the business. Thus, walmart needs to retain these customers. Also for the other age groups with less purchases, walmart should come up with some ideas to increase sales from them.

4) Walmart have strong customer base in 'City C', therefore it may be advisable to increase business through word of mouth marketing here. For 'City B' & 'City A', digital marketing might be advised using Instagram, Facebook, Google and YouTube ads, as well as SEO and other digital marketing methods.

5) Product categories such as 1, 5, 8 & 11 are purchased by most of the customers. Walmart can focus on the product categories, and also offer upsells on them to further enhance revenue. Other than that, digital marketing is advised for other products to increase visbility.

✓ 0s completed at 16:01