Walmart Business Case Study: Exploratory Data Analysis (EDA)

Introduction

Walmart Inc. seeks to analyze customer purchase behavior, focusing on spending patterns based on gender and other factors. The insights derived from this analysis will help Walmart make informed business decisions and tailor marketing strategies to different customer segments.

In [1]:
```
#importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [2]:
```
# Load the dataset
df = pd.read_csv('walmart_data.csv')
df.head()
```

Out[2]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years
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+

In [3]:
```
#checking data types of all columns
print("\nData types of each column:")
print(df.dtypes)
```

```
Data types of each column:
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
```

In [4]:
```
df.dtypes
```

Out[4]:
```
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
```

- Product ID,Gender,Age,City Category, Stay in current city years are object (String).
- User ID,Occupation,Marital Status,Product Category, Purchase are in integer data type.

In [5]:
```
df.shape
```

Out[5]:
```
(550068, 10)
```

- Dataset contains 550058 rows and 10 columns

In [6]:
```
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 [7]:
```
df.nunique()
```

Out[7]:
```
User_ID                        5891
Product_ID                     3631
Gender                            2
Age                               7
Occupation                       21
City_Category                     3
Stay_In_Current_City_Years        5
Marital_Status                    2
Product_Category                 20
Purchase                      18105
dtype: int64
```

Data Cleaning

Checking for null values

In [8]:
```
df.isnull().sum().sort_values(ascending=True)
```

Out[8]:
```
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
```

- there is no null values in this dataset.

Non Graphical Analysis

In [9]:
```
# Marital status wise count and unique value
df["Marital_Status"].unique()
```

Out[9]:
```
array([0, 1], dtype=int64)
```

In [10]:
```
df["Marital_Status"].nunique()
```

Out[10]:
```
2
```

In [11]:
```
df["Marital_Status"].value_counts(normalize=True).round(2)*100
```

Out[11]:
```
0    59.0
1    41.0
Name: Marital_Status, dtype: float64
```

- Marital status is divided into two category:"0" refers single and "1" refer married .
- Most of the customer are single(59%) followed by married(41%)

In [12]:
```
df["Product_Category"].nunique()
```

Out[12]:
```
20
```

- Total 20 Products different are there in this data.

In [13]:
```
df["Product_Category"].value_counts(normalize=True).round(2)*100
```

Out[13]:
```
5     27.0
1     26.0
8     21.0
11     4.0
2      4.0
6      4.0
3      4.0
4      2.0
16     2.0
15     1.0
13     1.0
10     1.0
12     1.0
7      1.0
18     1.0
20     0.0
19     0.0
14     0.0
17     0.0
9      0.0
Name: Product_Category, dtype: float64
```

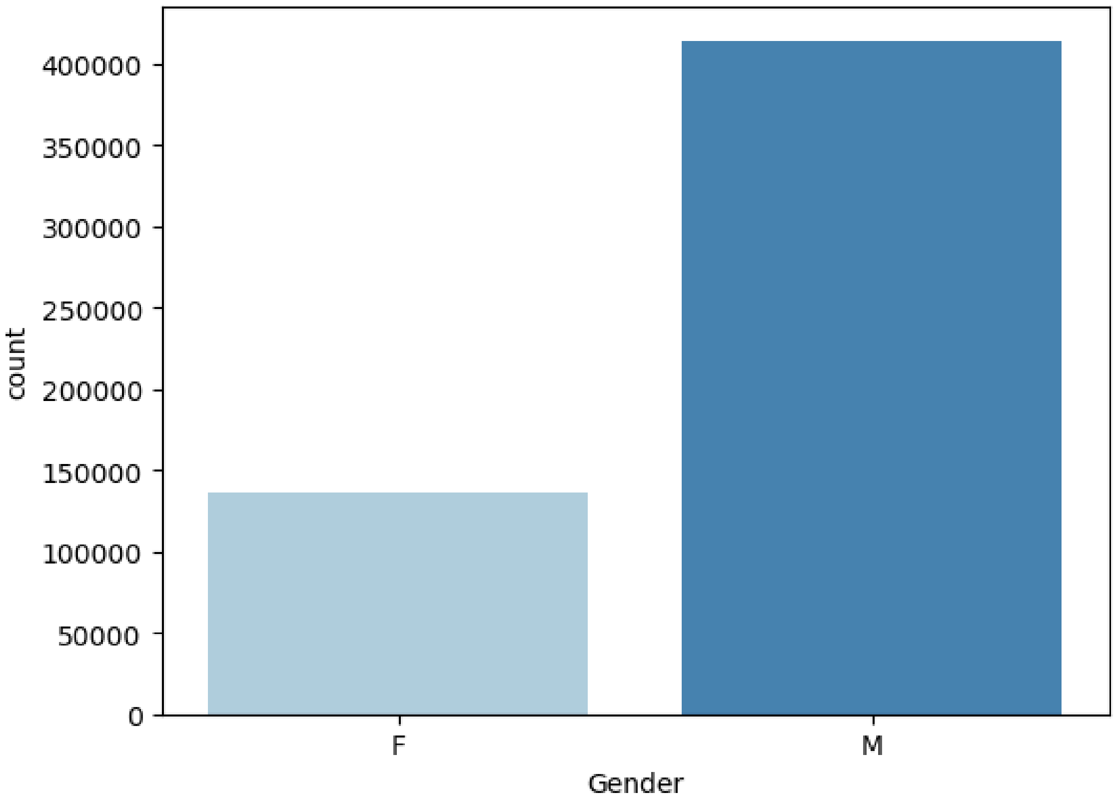
Observation :

- Walmart have 20 different Product categories in their stores. *Product_category with 5,1,8 are top three among 20 in walmart inventory.

Visual Analysis:-

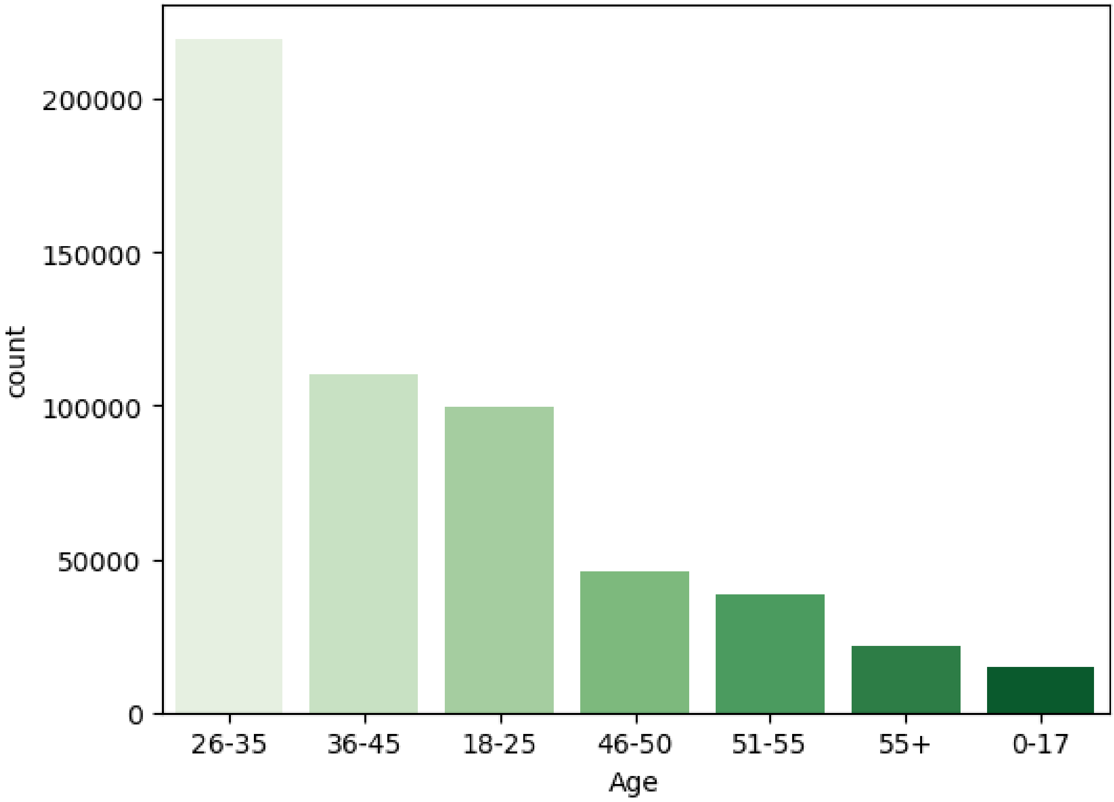
In [14]:
```
# Gender countplot
sns.countplot(data=df,x="Gender",palette="Blues")
```

Out[14]: <Axes: xlabel='Gender', ylabel='count'>



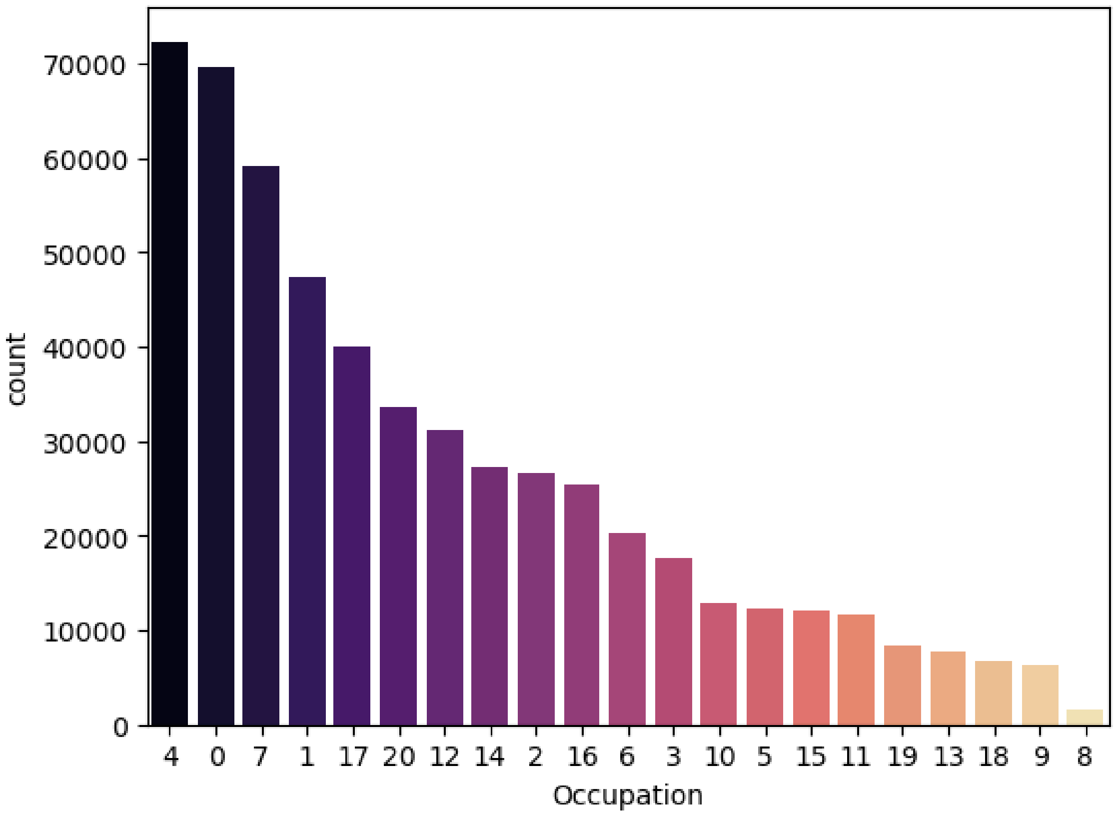
In [15]:
```
# Age countplot
sns.countplot(data=df,x="Age",palette="Greens",order=df["Age"].value_counts
```

Out[15]: <Axes: xlabel='Age', ylabel='count'>



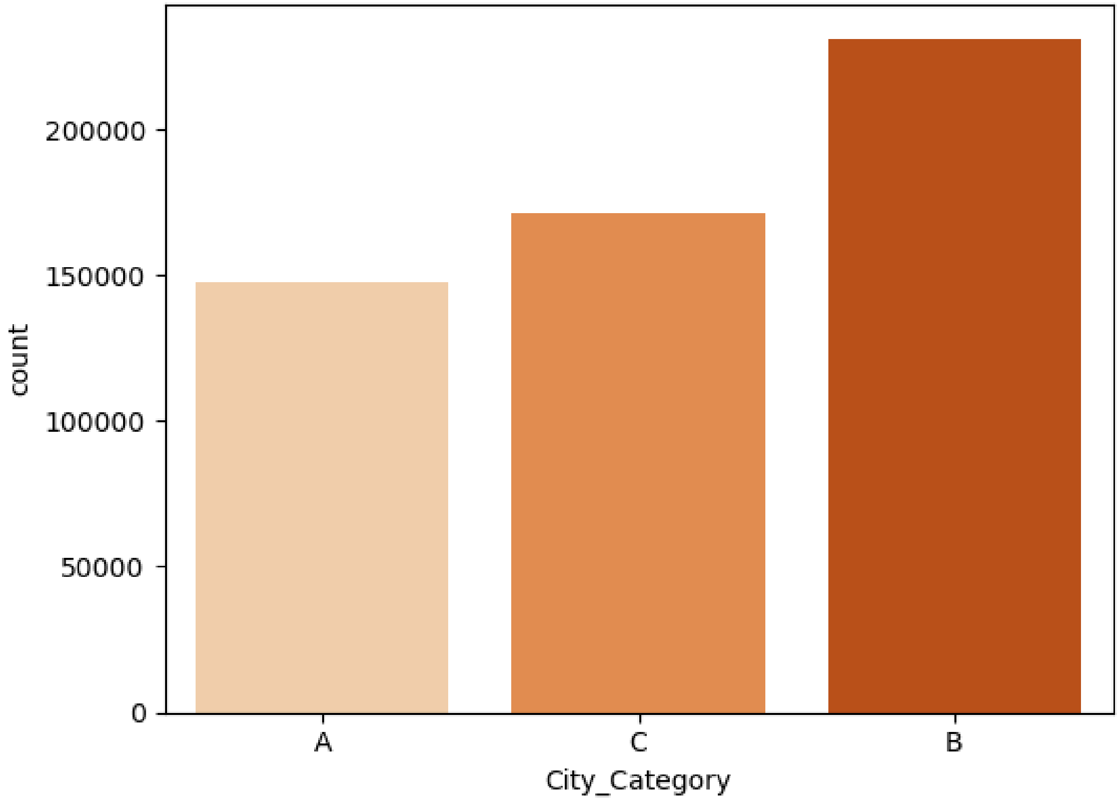
In [16]:
```
# Occupation Count plot
sns.countplot(data=df,x="Occupation",order=df["Occupation"].value_counts().
```

Out[16]: <Axes: xlabel='Occupation', ylabel='count'>



In [17]:
```
# City_category countplot
sns.countplot(data=df,x="City_Category",palette="Oranges")
```

Out[17]:
```
<Axes: xlabel='City_Category', ylabel='count'>
```

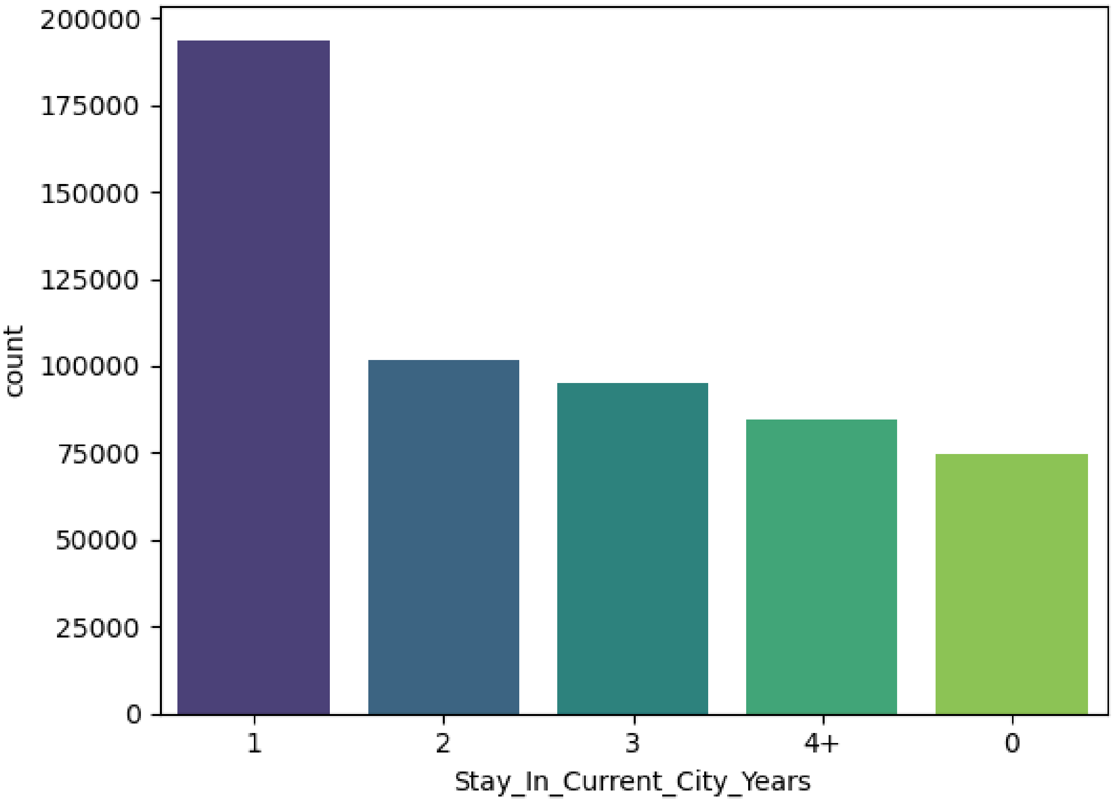


Observation :

- Most of the Customer are from the city_category B followed by A

In [18]:
```
# current city stay countplot
sns.countplot(data=df,x="Stay_In_Current_City_Years",order=df["Stay_In_Curr
```

Out[18]: <Axes: xlabel='Stay_In_Current_City_Years', ylabel='count'>

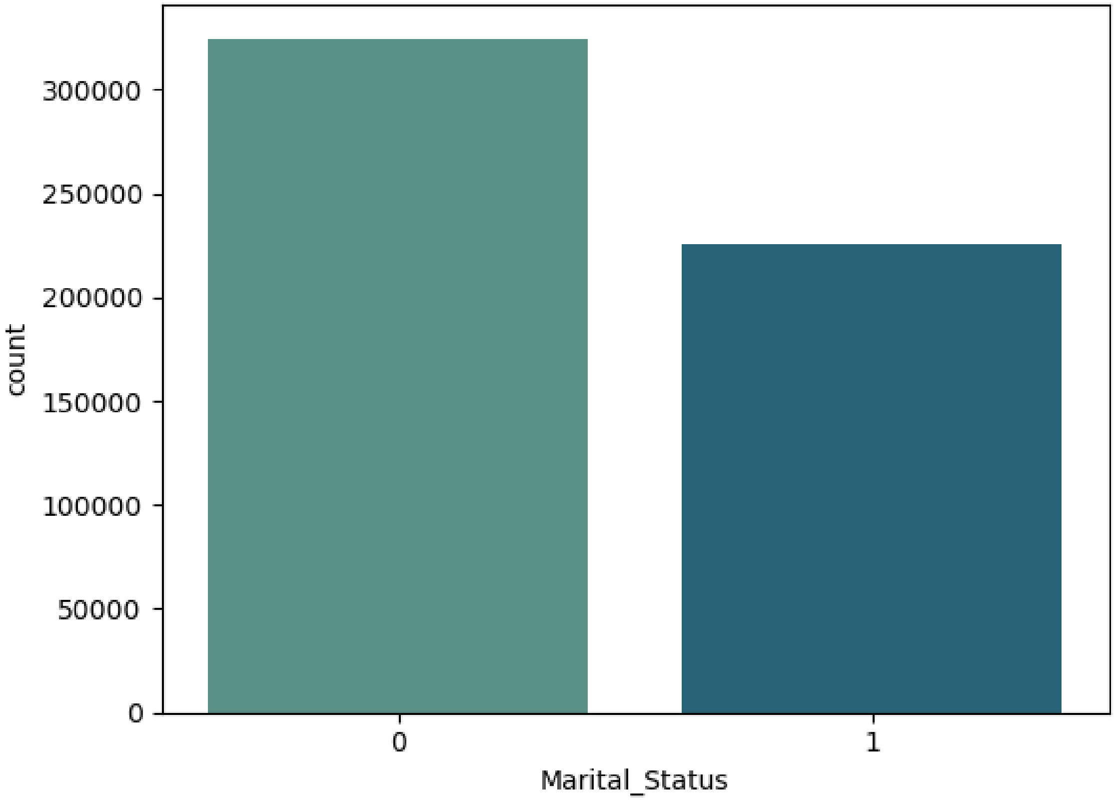


Observation :

- Most of the customer who go to walmart for shopping are residing in their current city for 1yr .

In [19]:
```
# Marital status countplot
sns.countplot(data=df,x="Marital_Status",palette="crest")
```

Out[19]:
```
<Axes: xlabel='Marital_Status', ylabel='count'>
```

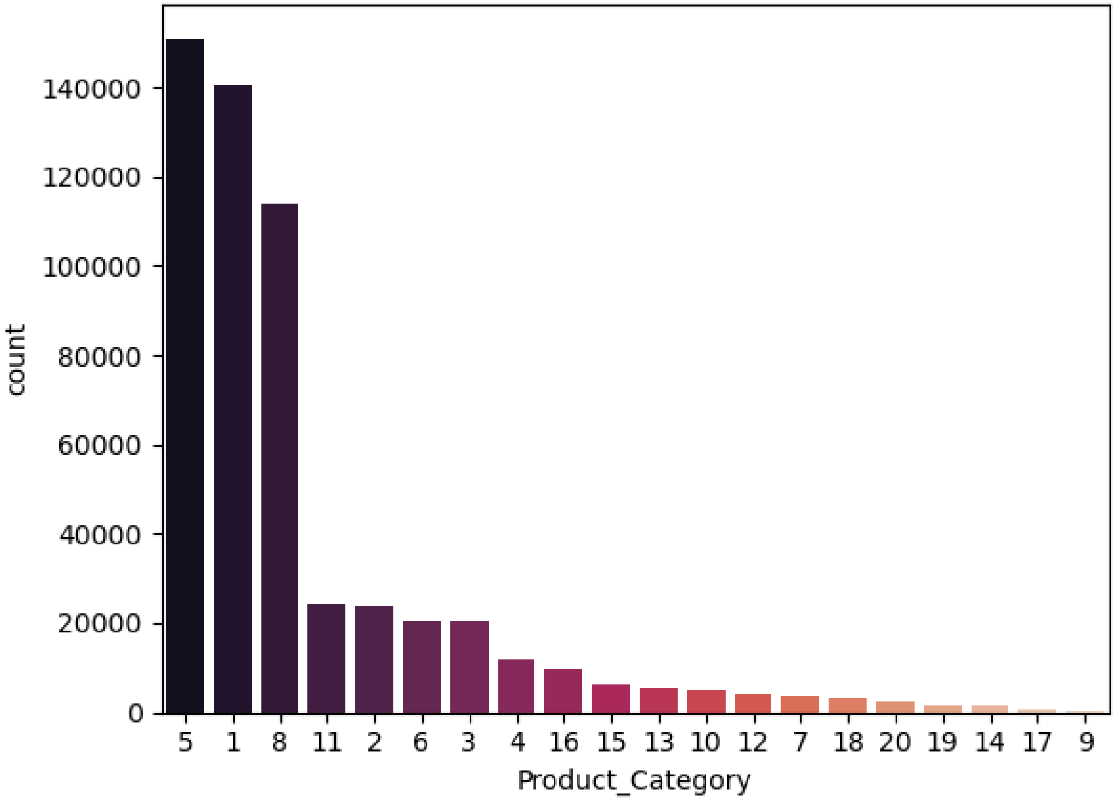


Observation :

- From the graph we can see that most of the customers are unmarried.

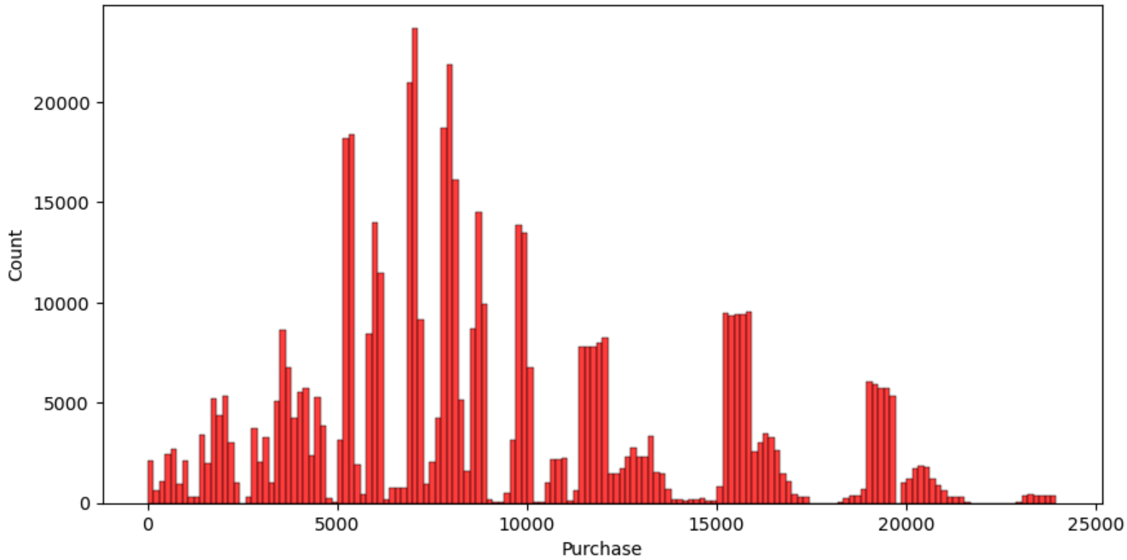
In [20]:
```
# Product_category countplot
sns.countplot(data=df,x="Product_Category",order=df["Product_Category"].val
```

Out[20]: <Axes: xlabel='Product_Category', ylabel='count'>



2.Histogram Plot:

In [21]:
```
# Purchase plot
plt.figure(figsize=(10,5))
sns.histplot(df["Purchase"],color="r")
plt.show()
```

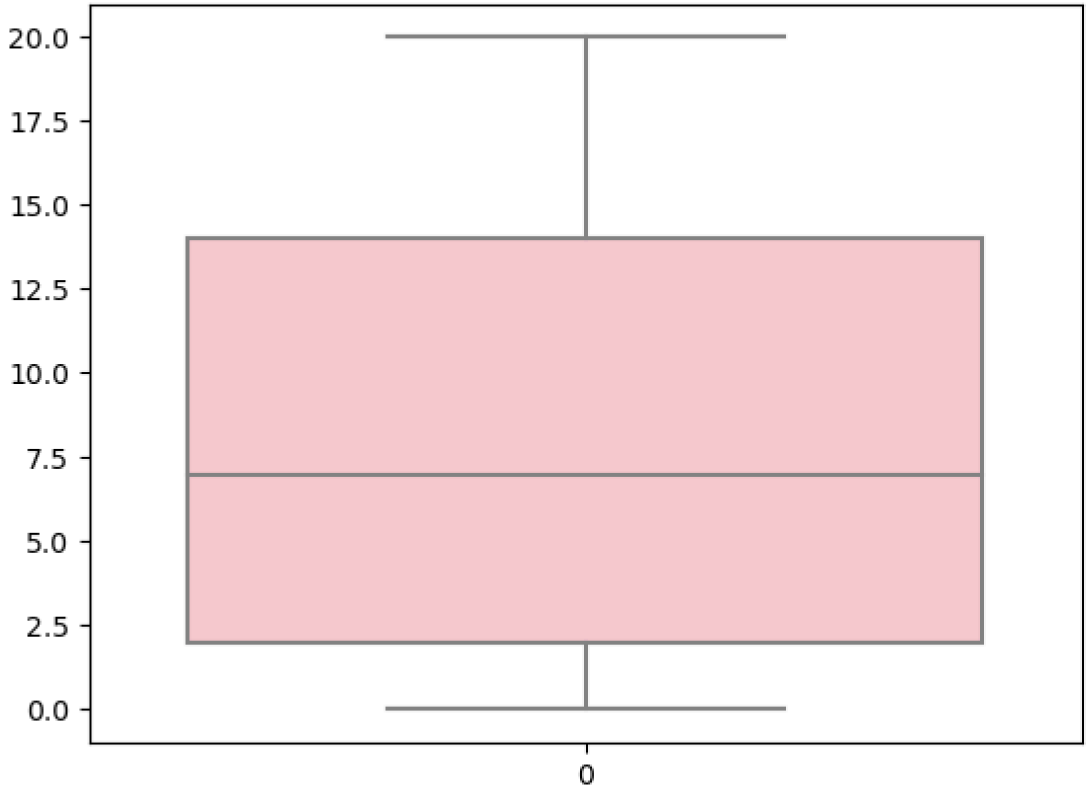


Observation :

- Customer who come to walmart for shopping most of them expend in the range of 6K-8K.

3. Box plot

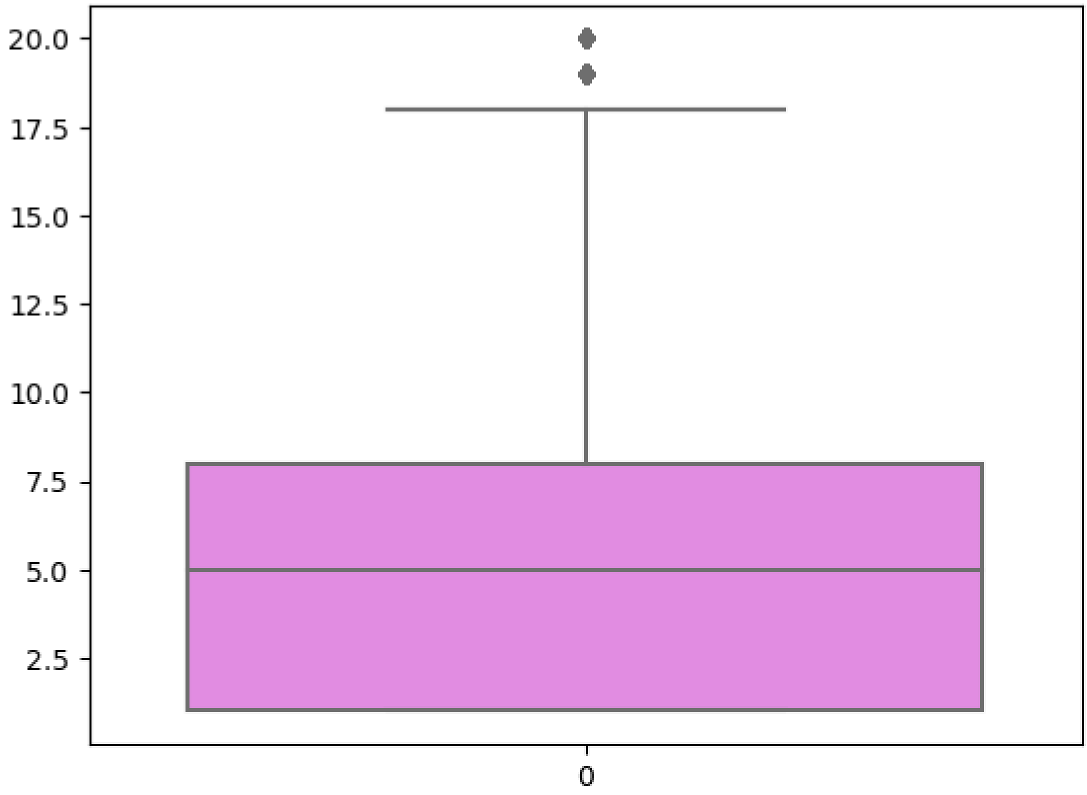
In [22]:
```
# Occupation Boxplot
sns.boxplot(df["Occupation"],orient="v",color="pink")
plt.show()
```



Observation :

- No outlier is present.

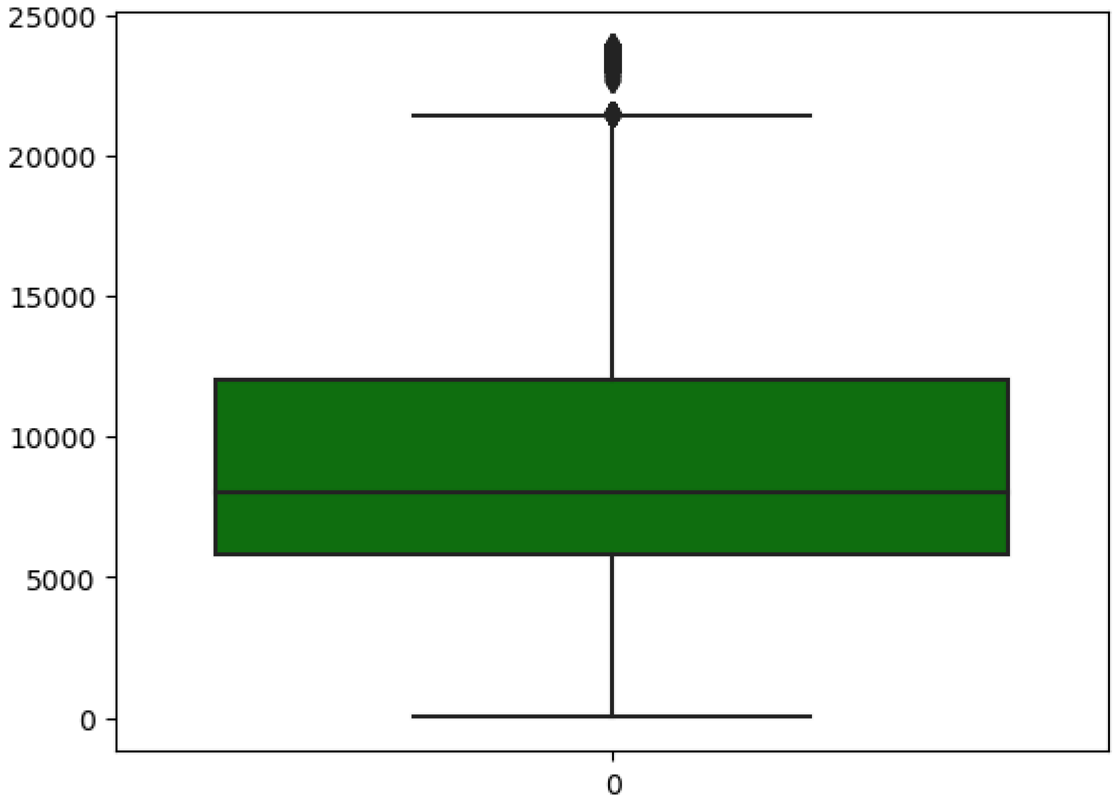
In [23]:
```
# Product_category boxplot
sns.boxplot(df["Product_Category"],orient="v",color="violet")
plt.show()
```



Observation :

- Outliers are above product_Category 17.

In [24]:
```
#Purchase boxplot
sns.boxplot(df["Purchase"],orient="v",color="g")
plt.show()
```



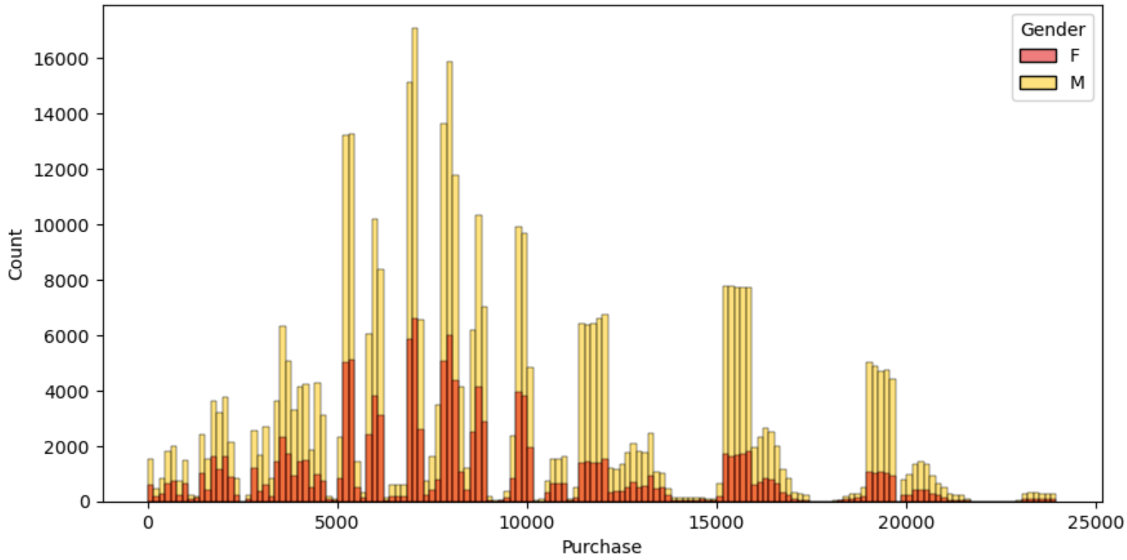
Observation :

- Outliers are above purchase amount of 20000.

Bivariate Analysis

1.Histogram plot

In [25]:
```
# Purchase with respect to gender
plt.figure(figsize=(10,5))
sns.histplot(x=df["Purchase"],hue=df['Gender'],palette="hot")
plt.show()
```

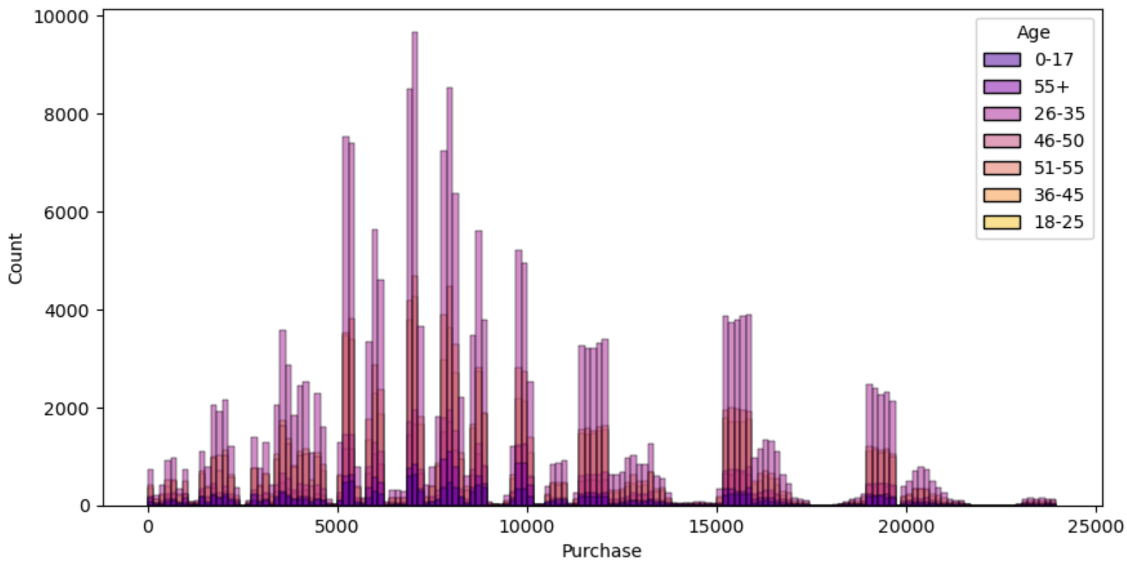


Observation :

- Male mostly prefer walmart for the shopping than women.

In [26]:
```
#Purchase with respect to Age
plt.figure(figsize=(10,5))
sns.histplot(data=df,x="Purchase",hue="Age",palette="plasma")
```

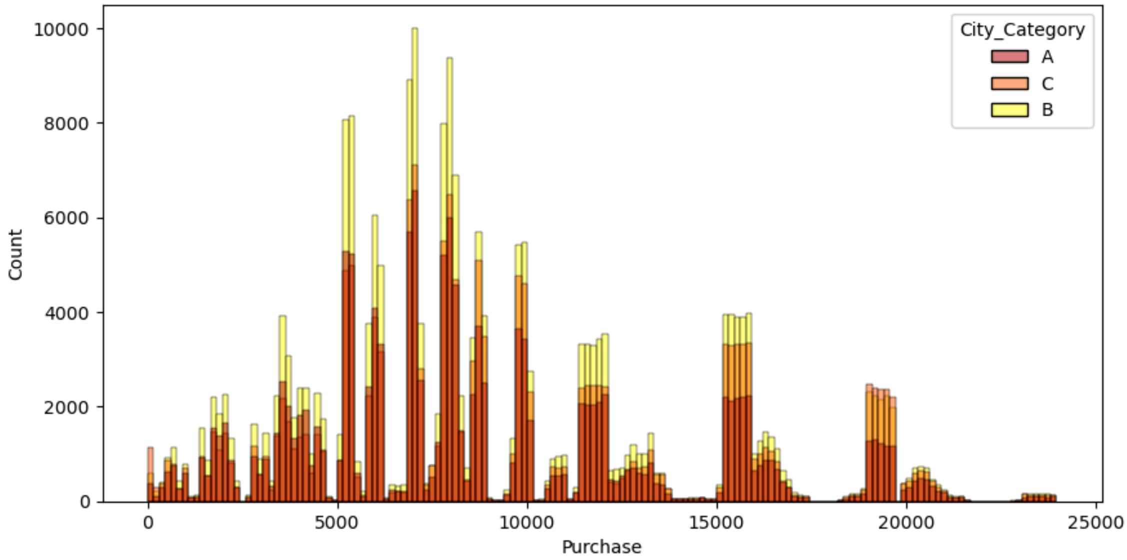
Out[26]: <Axes: xlabel='Purchase', ylabel='Count'>



Observation :

- Maximum purchase are done by the customers of Age Group 26-35.

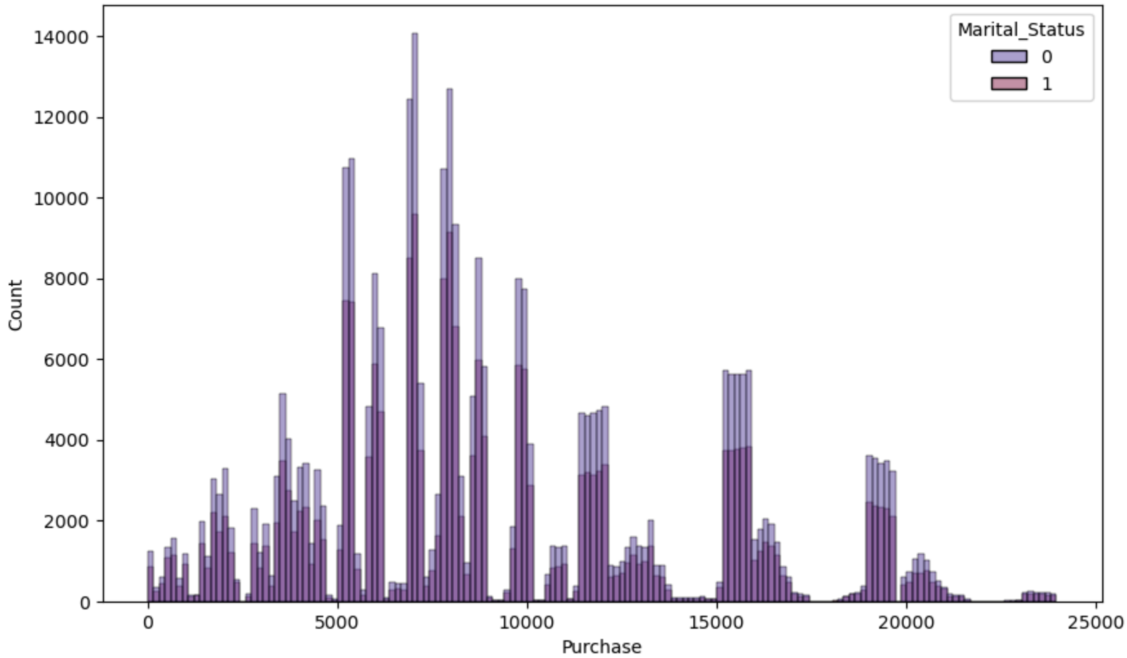
In [27]:
```
# Purchase with respect to City_category
plt.figure(figsize=(10,5))
sns.histplot(data=df,x="Purchase",hue="City_Category",palette="hot")
plt.show()
```



Observation :

- Customers belonging to the City_category of B does maximum shopping at walmart followed by C and then A.

In [28]:
```
#Purchase with respect to Marital Status
plt.figure(figsize=(10,6))
sns.histplot(data=df,x="Purchase",hue="Marital_Status",palette="twilight")
plt.show()
```

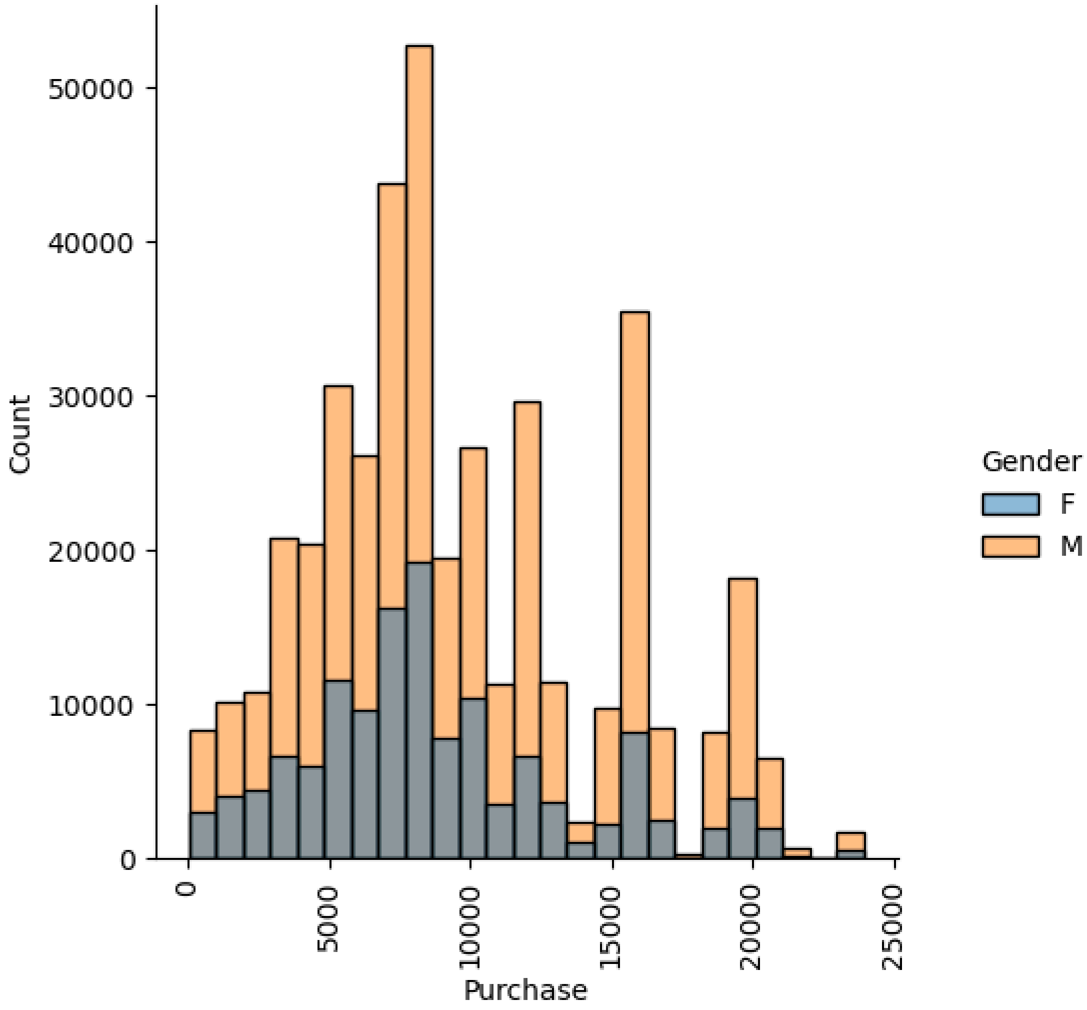


Observation :

- Mostly unmarried people do shopping from walmart in comparison to married.

2.Dis plot

In [29]:
```
sns.displot(data=df,x="Purchase",hue="Gender",bins=25,color="magma_r")
plt.xticks(rotation=90)
plt.show()
```

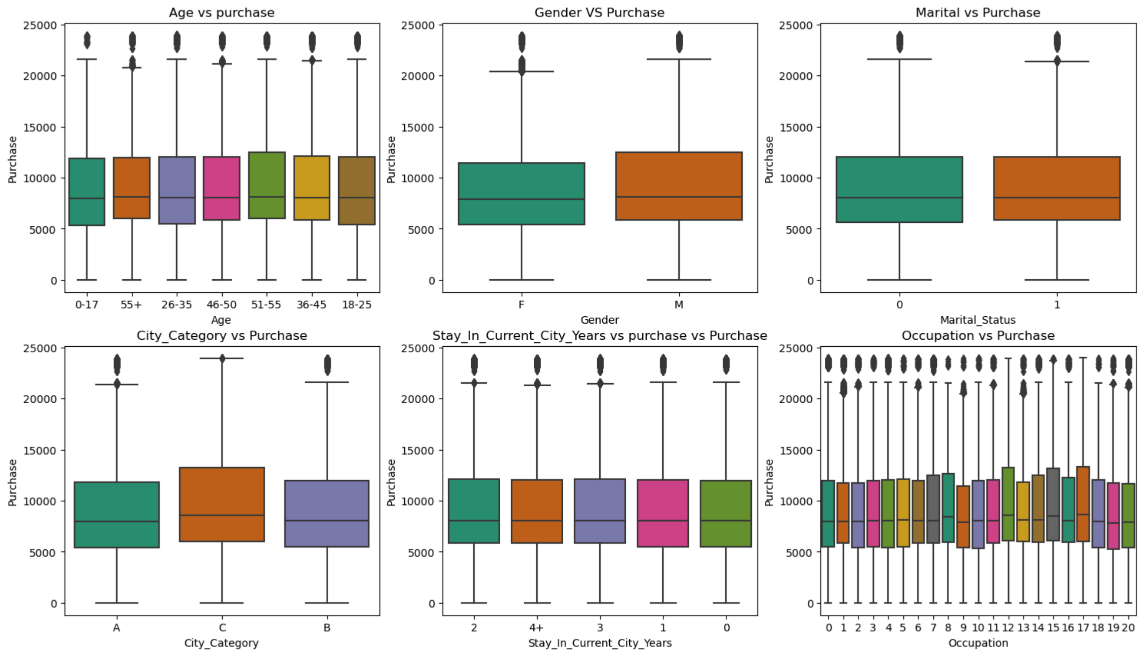


Observation :

- Males are purchasing more compare to female.

Box Plots-

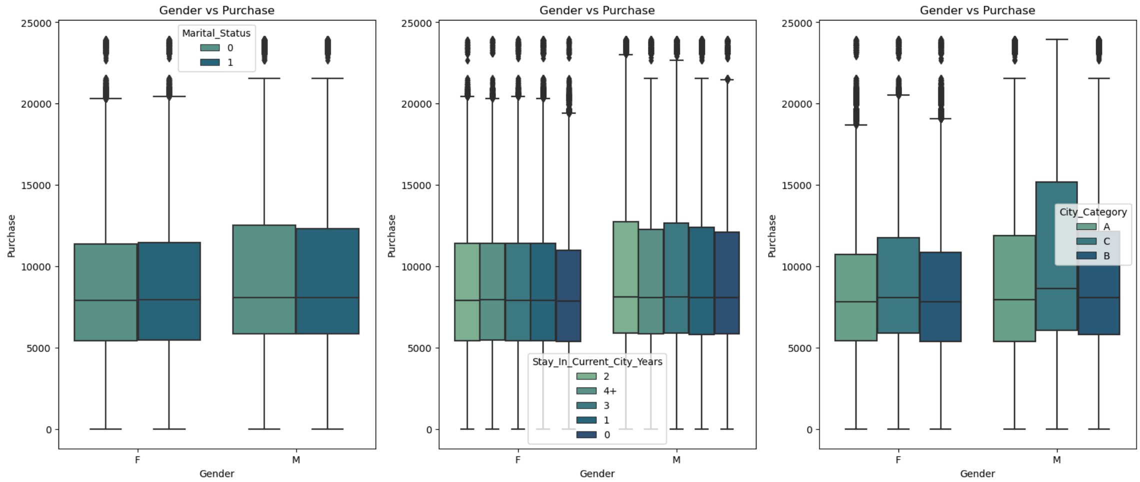
In [30]:
```
# Purchase vs Various Parameter(gender,marital_status,Age,City_category,Cur
plt.figure(figsize=(18,10))
plt.subplot(2,3,1)
sns.boxplot(data=df,x="Age",y="Purchase",palette="Dark2")
plt.title("Age vs purchase",fontsize=12)
plt.subplot(2,3,2)
sns.boxplot(data=df,x="Gender",y="Purchase",palette="Dark2")
plt.title("Gender VS Purchase",fontsize=12)
plt.subplot(2,3,3)
sns.boxplot(data=df,x="Marital_Status",y="Purchase",palette="Dark2")
plt.title("Marital vs Purchase",fontsize=12)
plt.subplot(2,3,4)
sns.boxplot(data=df,x="City_Category",y="Purchase",palette="Dark2")
plt.title("City_Category vs Purchase",fontsize=12)
plt.subplot(2,3,5)
sns.boxplot(data=df,x="Stay_In_Current_City_Years",y="Purchase",palette="Da
plt.title("Stay_In_Current_City_Years vs purchase vs Purchase",fontsize= 12
plt.subplot(2,3,6)
sns.boxplot(data=df,x="Occupation",y="Purchase",palette="Dark2")
plt.title("Occupation vs Purchase",fontsize=12)
plt.show()
```



Observation :

1. There is 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 more amongst all.
4. Median purchase for City Category 'C' is more than the rest City Category.
5. Median purchase for all current city stay is nearly equal.
6. Median purchase is almost equal for single and married people.

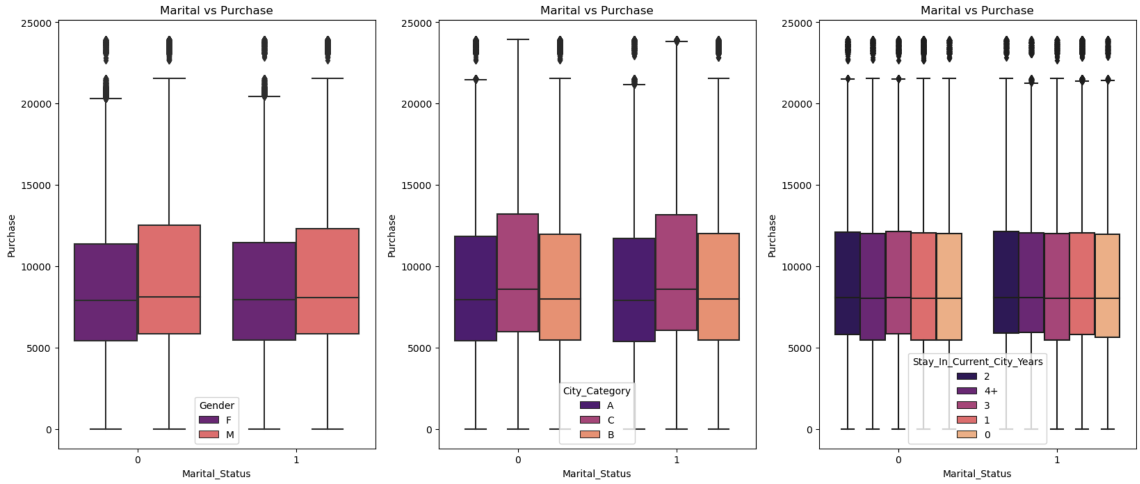
In [31]:
```
# Gender vs Purchase(hue as marital_status,city_category,current_city_stay,
plt.figure(figsize=(20,8))
plt.subplot(1,3,1)
sns.boxplot(x="Gender",y="Purchase",hue="Marital_Status",data=df,palette="c
plt.title("Gender vs Purchase",fontsize=12)
plt.subplot(1,3,2)
sns.boxplot(x="Gender",y="Purchase",data=df,hue="Stay_In_Current_City_Years
plt.title("Gender vs Purchase",fontsize=12)
plt.subplot(1,3,3)
sns.boxplot(x="Gender",y="Purchase",data=df,hue="City_Category",palette="cr
plt.title("Gender vs Purchase",fontsize=12)
plt.show()
```



Observation :

- In every cases such as marital status, city category & current stay city, male customers are slightly more purchasing the product as compared to female customers.

In [32]:
```
# Marital_status vs Purchase(with hue as Gender,Stay_in _current_city,city_
plt.figure(figsize=(20,8))
plt.subplot(1,3,1)
sns.boxplot(x="Marital_Status",y="Purchase",data=df,hue="Gender",palette="m
plt.title("Marital vs Purchase",fontsize=12)
plt.subplot(1,3,2)
sns.boxplot(x="Marital_Status",y="Purchase",data=df,hue="City_Category",pal
plt.title("Marital vs Purchase",fontsize=12)
plt.subplot(1,3,3)
sns.boxplot(x="Marital_Status",y="Purchase",data=df,hue="Stay_In_Current_Ci
plt.title("Marital vs Purchase",fontsize=12)
plt.show()
```



Observation :

- Purchase amount for both single & married customers are nearly same.

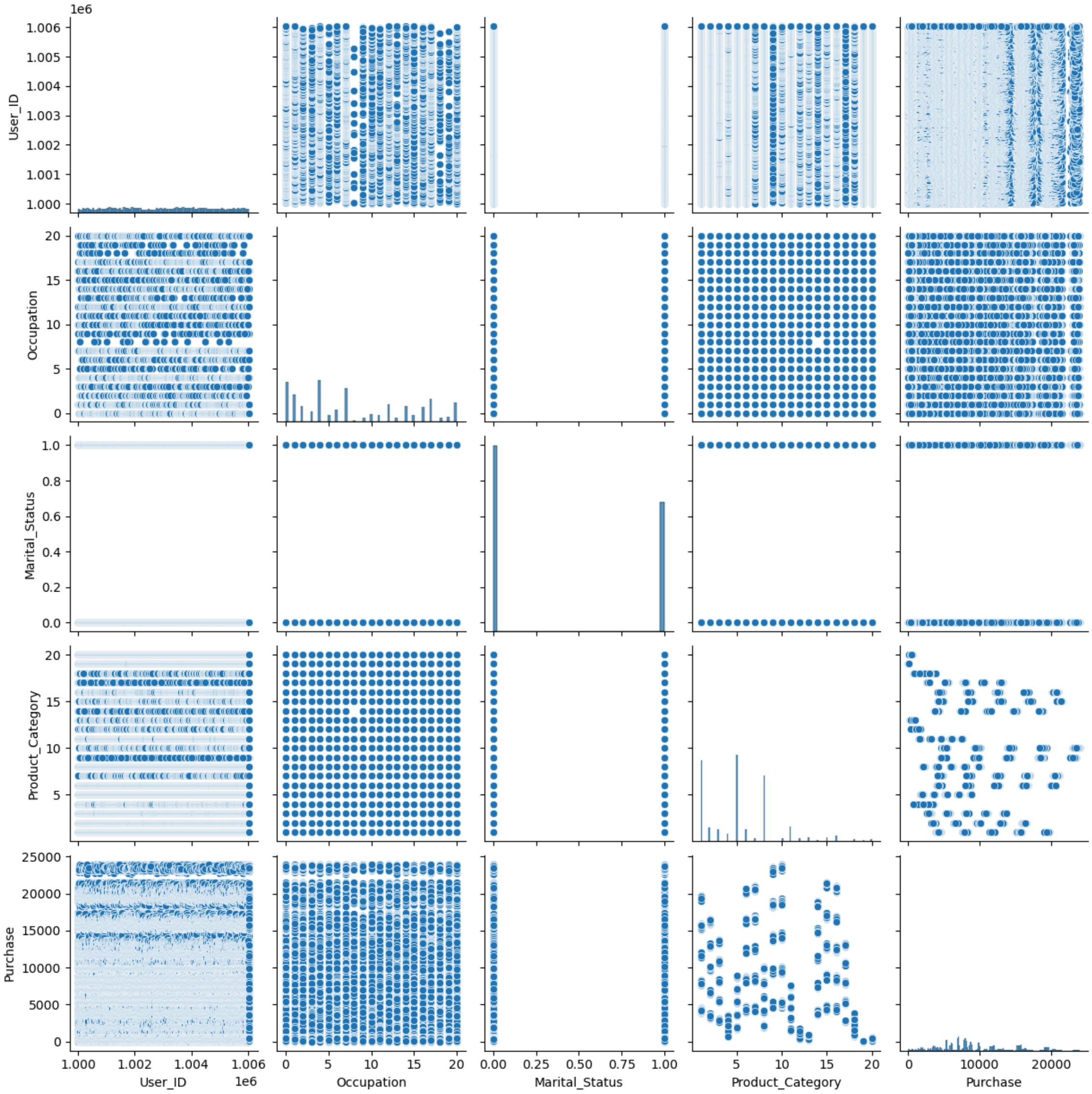
C. Multivariate Analysis -

To check correlation

1.Pair Plots -

In [34]:
```
plt.figure(figsize=(12,10))
sns.pairplot(df)
plt.show()
```

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



4. CLT & Confidence Interval Analysis

In [35]:
```
samp=df.sample(500)
samp
```

Out[35]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_
384978	1005252	P00188442	M	26-35	20	B	
428663	1006003	P00219242	F	46-50	17	C	
107674	1004543	P00124842	M	26-35	2	A	
313959	1000379	P00086042	F	36-45	1	C	
142747	1004013	P00115142	M	26-35	1	C	
...	...	...	...	...	...	...	
370537	1003095	P00243242	F	26-35	3	B	
430531	1000273	P00362642	M	18-25	4	B	
189157	1005205	P00165742	M	26-35	1	B	
465320	1005671	P00292342	M	26-35	1	C	
444006	1002285	P00306042	F	0-17	10	B	

500 rows × 10 columns

Gender Analysis -

In [36]:
```
# Mean for men
df[df["Gender"]=="M"]["Purchase"].mean()
```

Out[36]: 9437.526040472265

In [37]:
```
# Mean for women
df[df["Gender"]=="F"]["Purchase"].mean()
```

Out[37]: 8734.565765155476

In [38]:
```
# Sample Statistical Properties
samp.groupby("Gender")["Purchase"].describe()
```

Out[38]:

Gender	count	mean	std	min	25%	50%	75%	max
F	113.0	9084.840708	5125.879863	36.0	5414.0	7950.0	11812.0	23810.0
M	387.0	9526.633075	4968.303462	24.0	5891.0	8331.0	12236.0	21382.0

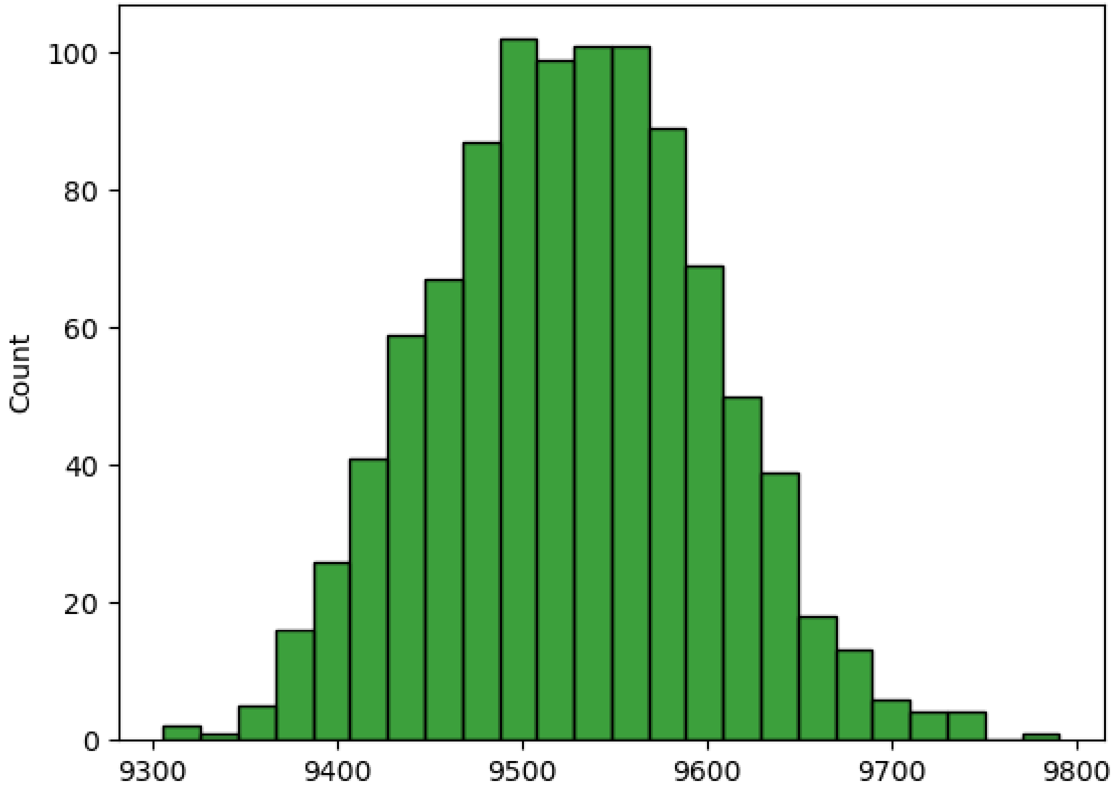
In [39]:
```
male_samp_mean = [samp[samp["Gender"] == "M"].sample(5000, replace = True)[
male_samp_mean
```

Out[39]:
```
[9511.5592,
 9523.5084,
 9456.1986,
 9494.8104,
 9499.6258,
 9639.1118,
 9548.5538,
 9471.0134,
 9660.3598,
 9558.1114,
 9492.9364,
 9502.4796,
 9557.7768,
 9494.1794,
 9507.7222,
 9522.7806,
 9609.4484,
 9578.382,
 9570.0888,
```

In [40]:
```
len(male_samp_mean)
```

Out[40]:
```
1000
```

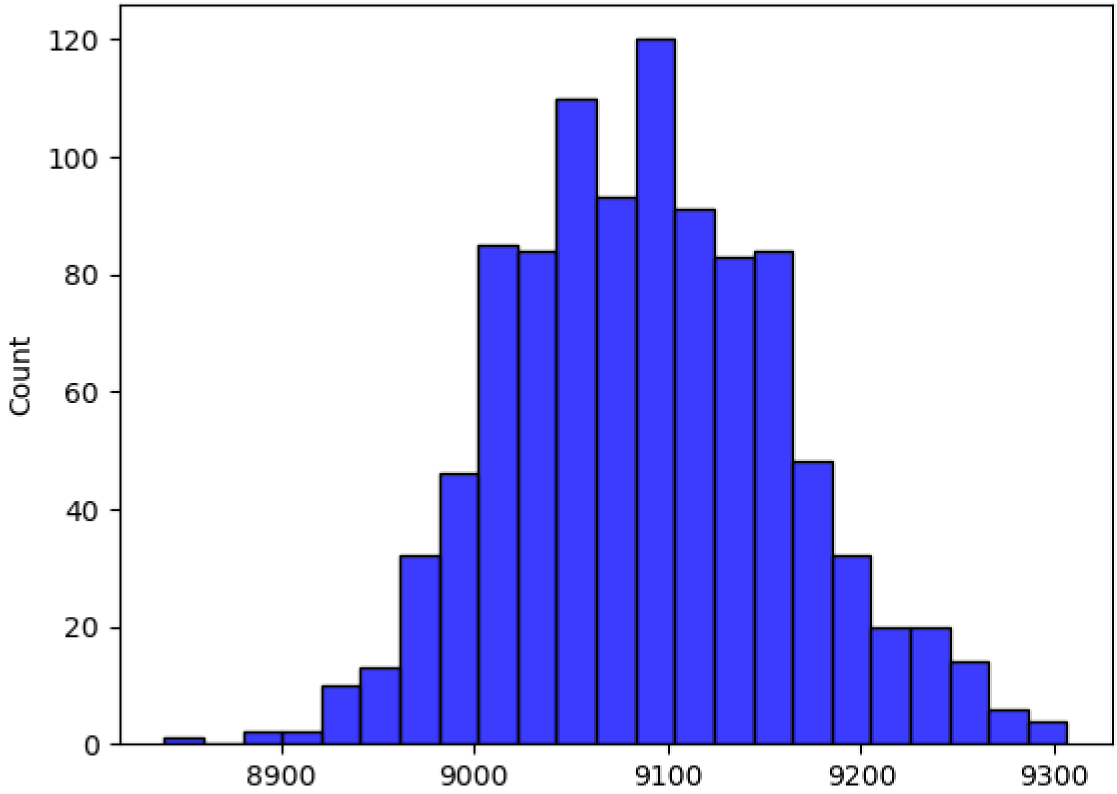
In [41]:
```
sns.histplot(male_samp_mean,color="g")
plt.show()
```



In [42]:
```
female_samp_mean = [samp[samp["Gender"] == "F"].sample(5000, replace = True
female_samp_mean
```

```
 9104.6428,
 9057.2048,
 9170.4818,
 9024.463,
 9069.7586,
 8994.4638,
 8912.8346,
 9245.6814,
 9190.5234,
 9044.1612,
 9028.5656,
 9182.1186,
 9013.6926,
 9171.1454,
 9088.5956,
 8969.1722,
 9011.5844,
 9092.5232,
 9050.23,
```

In [43]:
```
sns.histplot(female_samp_mean,color="b")
plt.show()
```



In [44]:
```
# std deviation of male sample
male_std=np.std(male_samp_mean).round(3)
male_std
```

Out[44]: 73.859

In [45]:
```
#std deviation of female sample
female_std=np.std(female_samp_mean).round(3)
female_std
```

Out[45]: 73.028

Confidence Interval - 90%

In [46]:
```
# confidence Interval of male 90%
from scipy.stats import norm

male_low=np.mean(male_samp_mean)+norm.ppf(0.05)*np.std(female_samp_mean)
male_high=np.mean(male_samp_mean)+norm.ppf(0.95)*np.std(female_samp_mean)
male_low.round(3),male_high.round(3)
```

Out[46]: (9407.4, 9647.641)

In [47]:
```
# 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_mea
female_low.round(3),female_high.round(3)
```

Out[47]: (8968.855, 9209.096)

In [48]:
```
# To check the overlapping of confidence interval
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)
```

Out[48]: (array([9407.599, 9644.057]), array([8976.78 , 9214.785]))

Confidence Interval - 95%

In [49]:
```
# Confidence Interval of male for 95% interval
male_low=np.mean(male_samp_mean)+norm.ppf(.025)*np.std(male_samp_mean)
male_high=np.mean(male_samp_mean)+norm.ppf(.975)*np.std(male_samp_mean)
male_low.round(3),male_high.round(3)
```

Out[49]: (9382.76, 9672.28)

In [50]:
```
# Confidence Interval of female for 95% interval
female_low=np.mean(female_samp_mean)+norm.ppf(.025)*np.std(female_samp_mean
female_high=np.mean(female_samp_mean)+norm.ppf(.975)*np.std(female_samp_mea
female_low.round(3),female_high.round(3)
```

Out[50]: (8945.843, 9232.108)

In [51]:
```
# To check the overlapping of confidence interval of male and female
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)
```

Out[51]: (array([9387.781, 9674.089]), array([8955.011, 9242.056]))

- From above result, for 95% CI - it is clear that confidence intervals of male & female average purchases are not overlapping.

Confidence Interval - 99%

In [52]:
```
# confidence Interval of male for CI---99
male_low=np.mean(male_samp_mean)+norm.ppf(.005)*np.std(male_samp_mean)
male_high=np.mean(male_samp_mean)+norm.ppf(.995)*np.std(male_samp_mean)
male_low.round(3),male_high.round(3)
```

Out[52]: (9337.273, 9717.767)

In [53]:
```
# confidence Interval of female for CI---99
female_low=np.mean(female_samp_mean)+norm.ppf(.005)*np.std(female_samp_mean
female_high=np.mean(female_samp_mean)+norm.ppf(.995)*np.std(female_samp_mea
female_low.round(3),female_high.round(3)
```

Out[53]:
```
(8900.867, 9277.083)
```

In [54]:
```
# checking overlapping of confidence interval of male and female
male_ci=np.percentile(male_samp_mean,[.005,.995])
female_ci=np.percentile(female_samp_mean,[.005,.995])
male_ci.round(3),female_ci.round(3)
```

Out[54]:
```
(array([9307.061, 9371.375]), array([8841.899, 8930.33 ]))
```

- From above result, for 99% CI - it is clear that confidence intervals of male & female average purchases are not overlapping.

Marital Status Analysis -

In [55]:
```
Samp2=df.sample(500)
Samp2
```

Out[55]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_
240964	1001164	P00138942	F	26-35	19	A	
26947	1004109	P00121042	M	36-45	3	B	
141862	1003878	P00345742	M	36-45	16	B	
439478	1001639	P00205942	M	26-35	17	B	
259576	1003988	P00220342	F	26-35	14	B	
...	...	...	...	...	...	...	
6895	1001101	P00213742	M	36-45	1	A	
70782	1004861	P00324942	M	26-35	4	C	
418943	1004444	P00256642	F	26-35	12	C	
130567	1002042	P00010542	M	0-17	10	C	
310118	1005788	P00122742	M	26-35	0	A	

500 rows × 10 columns

In [56]:
```
# overall mean for Single customer
df[df["Marital_Status"]==0]["Purchase"].mean().round(3)
```

Out[56]: 9265.908

In [57]:
```
# overall mean for married customer
df[df["Marital_Status"]==1]["Purchase"].mean().round(3)
```

Out[57]: 9261.175

In [58]:
```
# Sample statistical Properties
Samp2.groupby("Marital_Status").describe()
```

Out[58]:

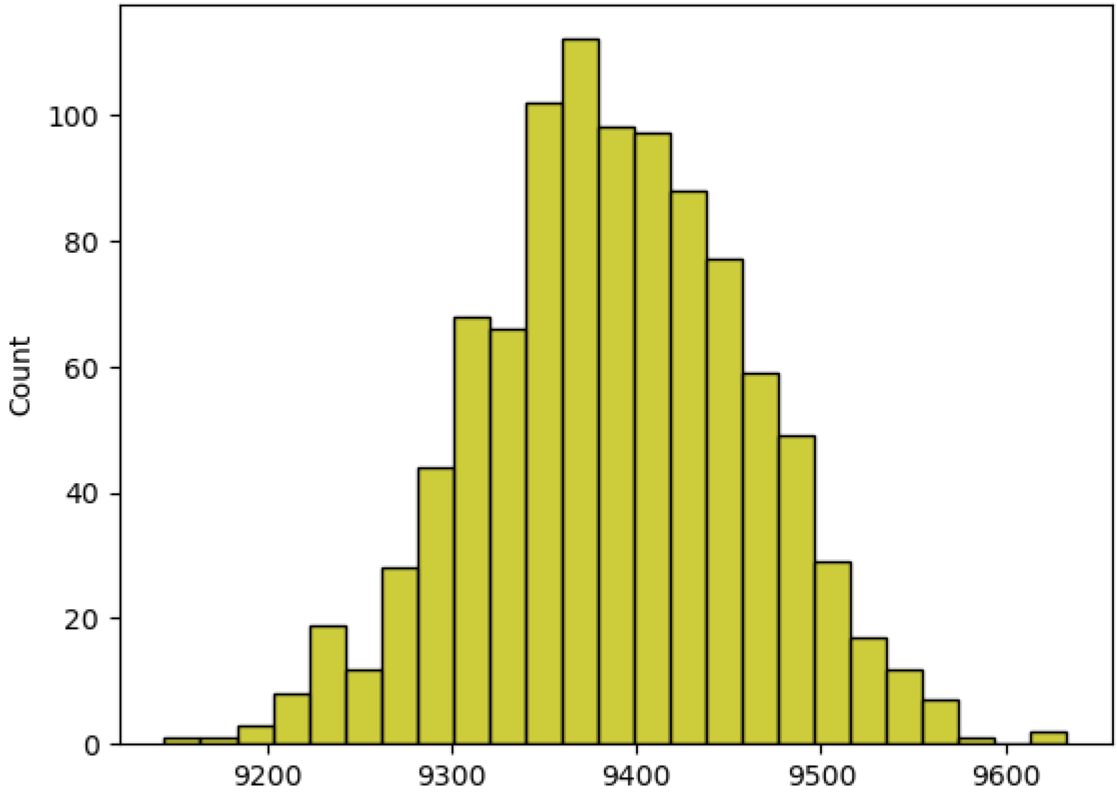
	User_ID						
	count	mean	std	min	25%	50%	75%
Marital_Status							
0	296.0	1.003019e+06	1656.320015	1000019.0	1001693.5	1003200.0	1004238.5
1	204.0	1.002993e+06	1708.055933	1000008.0	1001645.0	1002887.5	1004435.5

2 rows × 32 columns

In [59]:
```
unmarried_samp2_mean=[Samp2[Samp2["Marital_Status"]==0].sample(5000,replace
unmarried_samp2_mean
```

```
 9292.1486,
 9306.4486,
 9361.5046,
 9488.8454,
 9373.9862,
 9289.8298,
 9433.5596,
 9366.0688,
 9462.52,
 9304.0666,
 9371.6224,
 9472.9266,
 9447.578,
 9381.1742,
 9241.1208,
 9325.751,
 9215.373,
 9311.6366,
 9387.9388,
 9399.8922,
```

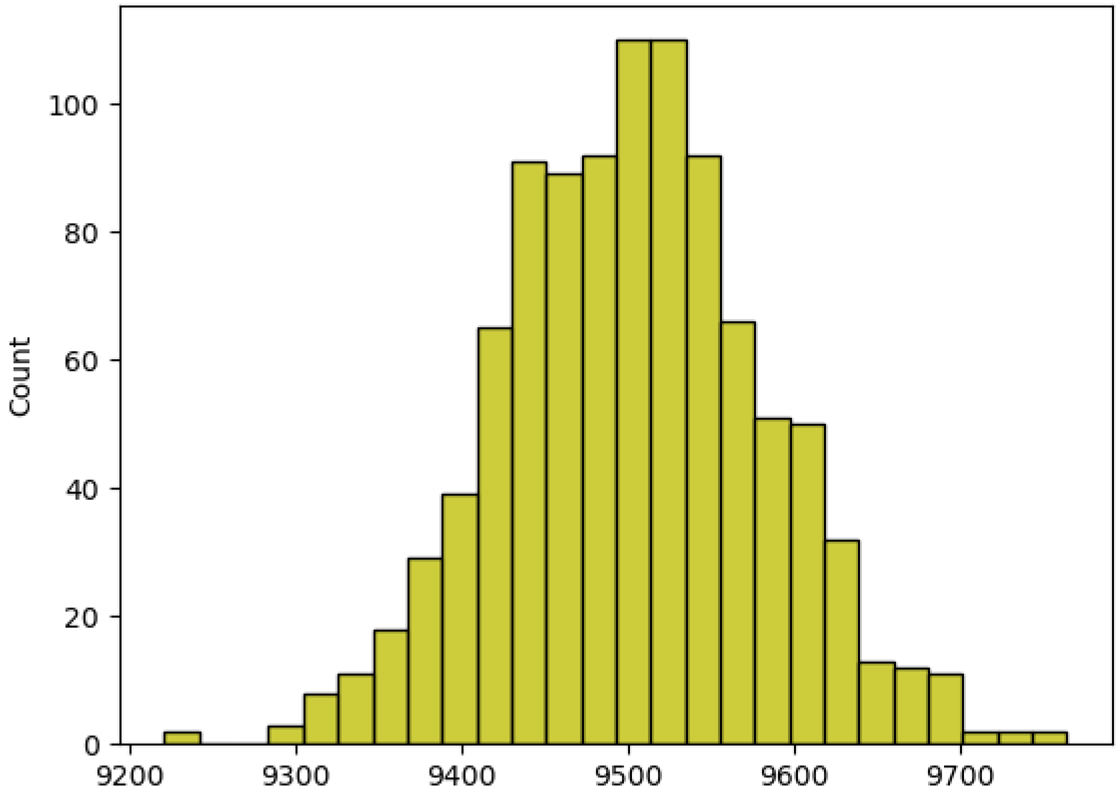
In [60]:
```
sns.histplot(unmarried_samp2_mean,color="y")
plt.show()
```



In [61]:
```
married_samp2_mean=[Samp2[Samp2["Marital_Status"]==1].sample(5000,replace=T
married_samp2_mean
```

Out[61]:
```
[9504.1852,
 9611.4436,
 9462.027,
 9499.6096,
 9562.1198,
 9630.1688,
 9634.328,
 9620.9674,
 9491.9948,
 9434.3794,
 9530.6552,
 9470.3606,
 9490.032,
 9471.3534,
 9409.5132,
 9653.085,
 9499.5732,
 9482.9782,
 9524.8298,
```

In [62]:
```
sns.histplot(married_samp2_mean,color="y")
plt.show()
```



In [63]:
```
# standard deviation of unmarried customer
np.std(unmarried_samp2_mean).round(3)
```

Out[63]: 73.611

In [64]:
```
# standard deviation of married customer
np.std(married_samp2_mean).round(3)
```

Out[64]: 79.183

CI --->90

In [65]:
```
# Confidence Interval of Single(unmarried)-->90
unmarried_low=np.mean(unmarried_samp2_mean)+norm.ppf(.05)*np.std(unmarried_
unmarried_high=np.mean(unmarried_samp2_mean)+norm.ppf(.95)*np.std(unmarried
unmarried_low.round(3),unmarried_high.round(3)
```

Out[65]: (9266.313, 9508.471)

In [66]:
```
# confidence Interval of married customer--->90
married_low=np.mean(married_samp2_mean)+norm.ppf(.05)*np.std(married_samp2_
married_high=np.mean(married_samp2_mean)+norm.ppf(.95)*np.std(married_samp2
married_low.round(3),married_high.round(3)
```

Out[66]: (9372.196, 9632.687)

In [67]:
```
#To check Overlapping of Confidence Interval
unmarried_CI=np.percentile(unmarried_samp2_mean,[5,95]).round(3)
married_CI=np.percentile(married_samp2_mean,[5,95]).round(3)
unmarried_CI,married_CI
```

Out[67]:
```
(array([9266.219, 9507.564]), array([9377.101, 9631.493]))
```

- From above result, for 90% CI - it is clear that confidence intervals of unmarried & married people average purchases are overlapping.

Confidence Interval - 95%

In [68]:
```
# Confidence Interval of single(unmarried)----95%
unmarried_low=np.mean(unmarried_samp2_mean)+norm.ppf(.025)*np.std(unmarried
unmarried_high=np.mean(unmarried_samp2_mean)+norm.ppf(.975)*np.std(unmarrie
unmarried_low.round(3),unmarried_high.round(3)
```

Out[68]:
```
(9243.118, 9531.666)
```

In [69]:
```
# confidence Interval of married---->95%
married_low=np.mean(married_samp2_mean)+norm.ppf(.025)*np.std(married_samp2
married_high=np.mean(married_samp2_mean)+norm.ppf(.975)*np.std(married_samp
married_low.round(3),married_high.round(3)
```

Out[69]:
```
(9347.245, 9657.638)
```

In [70]:
```
# checking the overlapping of confidence Interval
unmarried_CI=np.percentile(unmarried_samp2_mean,[2.5,97.5]).round(3)
married_CI=np.percentile(married_samp2_mean,[2.5,97.5]).round(3)
unmarried_CI,married_CI
```

Out[70]:
```
(array([9234.909, 9531.691]), array([9346.537, 9663.741]))
```

- From above result, for 95% CI - it is clear that confidence intervals of unmarried & married people average purchases are overlapping.

Confidence Interval - 99%

In [71]:
```
# Confidence Interval of unmarried for 99%
unmarried_low=np.mean(unmarried_samp2_mean)+norm.ppf(.005)*np.std(unmarried
unmarried_high=np.mean(unmarried_samp2_mean)+norm.ppf(.995)*np.std(unmarrie
unmarried_low.round(3),unmarried_high.round(3)
```

Out[71]:
```
(9197.784, 9577.001)
```

In [72]:
```
# confidence Interval of married for 99%
married_low=np.mean(married_samp2_mean)+norm.ppf(.005)*np.std(married_samp2
married_high=np.mean(married_samp2_mean)+norm.ppf(.995)*np.std(married_samp
married_low.round(3),married_high.round(3)
```

Out[72]:
```
(9298.478, 9706.404)
```

- From above result, for 99% CI - it is clear that confidence intervals of unmarried & married people average purchases are overlapping.

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 differnent 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 from 1 year, 18% from 2 years, 17% from 3 years.
8. From CLT graphs we have noticed that -

a) for gender samples, the confidence interval range was not overlapping.

b) for marital status samples, the confidence interval range was overlapping.

Recommendations -

1. Unmarried customers spend more money than married customers, So in order to increase sales from married customers, walmart should give some discounts offers for married people .
2. As males are purchasing more as compared to females, walmart should retain the male customer. Also walmart should think to grow sales from female perspective like giving them some discounts or do advertise about the product to attract female customer base.
3. Customers in the age group of 18-25 are the favourable age range for the business, so walmart should retain these customers. Also for the age group which is less purchasing than the above mentioned age group, walmart should come up with some ideas to involve those age groups in order to increase the sales.
4. Walmart have strong customer base in 'City C', so walmart should retain these customers, Also walmart should think to change strategies in 'City B' & 'City A' in order to increase the sales in those cities as well. Walmart can do advertising using online flatforms such as social digital flatforms such as Youtube, Instagram.
5. There are some product categories such as 1, 5, 8 & 11 which are purchased by most of the customers. So, walmart can focus on the product categories other than this so that the sales from all other categories would be increase at some sufficient level.

In [ ]:

```

```