Business Case Study: Walmart - Confidence Interval and CLT

About Walmart

Walmart is an American multinational retail corporation that operates a chain of supercentres, discount departmental stores, and grocery stores from the United States. Walmart has more than 100 million customers worldwide.

Business Problem

The Management team at Walmart Inc. wants to analyse the customer purchase behaviour (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).

1) Defining Problem Statement & Analysing Basic Metrics: ---

- **Importing Library**

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

- **Loading a Data Set**

```
df = pd.read_csv("/content/walmart_data.csv")
df.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10.0	A	2	0.0	3.0	8370.0
1	1000001	P00248942	F	0-17	10.0	A	2	0.0	1.0	15200.0
2	1000001	P00087842	F	0-17	10.0	A	2	0.0	12.0	1422.0
3	1000001	P00085442	F	0-17	10.0	A	2	0.0	12.0	1057.0
4	1000002	P00285442	M	55+	16.0	C	4+	0.0	8.0	7969.0

- About The Information

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50080 entries, 0 to 50079
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   User_ID                     50080 non-null  int64  
 1   Product_ID                  50080 non-null  object 
 2   Gender                      50080 non-null  object 
 3   Age                         50080 non-null  object 
 4   Occupation                  50080 non-null  int64  
 5   City_Category               50080 non-null  object 
 6   Stay_In_Current_City_Years  50080 non-null  object 
 7   Marital_Status              50080 non-null  int64  
 8   Product_Category            50079 non-null  float64
 9   Purchase                    50079 non-null  float64
dtypes: float64(2), int64(3), object(5)
memory usage: 3.8+ MB
```

So, from the information we found that it is a Pandas Data Frame which starts from 0 till 50079 & at the same time we have 9 columns which do not have any null values.

- **Conversion of Categorical Attributes to Category: -**

  Changing the Datatypes of few columns such as -> (Occupation, Marital Status & Product Category)

```
cols = ["Occupation","Marital_Status","Product_Category"]
df[cols] = df[cols].astype("object")
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                    float64
dtype: object
```

- **Statistical Summary**

```
print(df.memory_usage())
df.describe()
```

```
Index                          128
User_ID                     400640
Product_ID                  400640
Gender                      400640
Age                         400640
Occupation                  400640
City_Category               400640
Stay_In_Current_City_Years  400640
Marital_Status              400640
Product_Category            400640
Purchase                    400640
dtype: int64
```

	User_ID	Purchase
count	5.008000e+04	50079.000000
mean	1.002552e+06	9279.490525
std	1.786666e+03	4952.962432
min	1.000001e+06	185.000000
25%	1.001016e+06	5852.000000
50%	1.002100e+06	8045.000000
75%	1.004064e+06	12033.000000
max	1.006040e+06	23958.000000

- **Observation: ---**

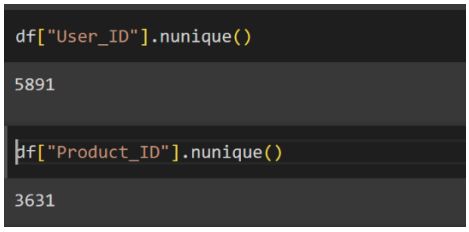
I. So, we have found that there are no missing values in the Data Set.

II. Purchase amount might have outliers -> The max purchase amount is 23958, While it's mean is 9279, The mean is sensitive to outliers, but the fact that is the mean so small compared to the max value, which indicates that the max value is an Outliers.

2) Non-Graphical Analysis: - (Value Counts & Unique Attributes):-

i. Unique Attributes: ---

So, we are Observing how many unique Users & Products are there In the Dataset: -



So, we have found that the unique values for User_ID & Product_ID are 5891 & 3631 respectively.

ii. Value Counts for the following Columns: --

- Gender
- Age
- Occupation
- City_Category
- Stay_In_Current_City_Years

- Marital_Status
- Product_Category

```
categorical_cols = ["Gender","Age","Occupation","Occupation","City_Category","Stay_In_Current_City_Years","Marital_Status","Product_Category"]
df[categorical_cols].melt().groupby(["variable","value",])[["value"]].count()/len(df)
```

Age	0-17	0.027455
	18-25	0.181178
	26-35	0.399200
	36-45	0.199999
	46-50	0.083082
	51-55	0.069993
	55+	0.039093
City_Category	A	0.268549
	B	0.420263
	C	0.311189

Gender	F	0.246895
	M	0.753105
Marital_Status	0	0.590347
	1	0.409653
Occupation	0	0.253198
	1	0.172437
	2	0.096672
	3	0.064174
	4	0.262906
	5	0.044275
	6	0.074009
	7	0.215003
	8	0.005621
	9	0.022874

	10	0.047012
	11	0.042126
	12	0.113364
	13	0.028098
	14	0.099293
	15	0.044231
	16	0.092247
	17	0.145593
	18	0.024077
	19	0.030763
	20	0.122029

Product_Category	1	0.255201
	2	0.043384
	3	0.036746
	4	0.021366
	5	0.274390
	6	0.037206
	7	0.006765
	8	0.207111
	9	0.000745
	10	0.009317
	11	0.044153
	12	0.007175
	13	0.010088
	14	0.002769
	15	0.011435
	16	0.017867
	17	0.001051
	18	0.005681

Observation: --

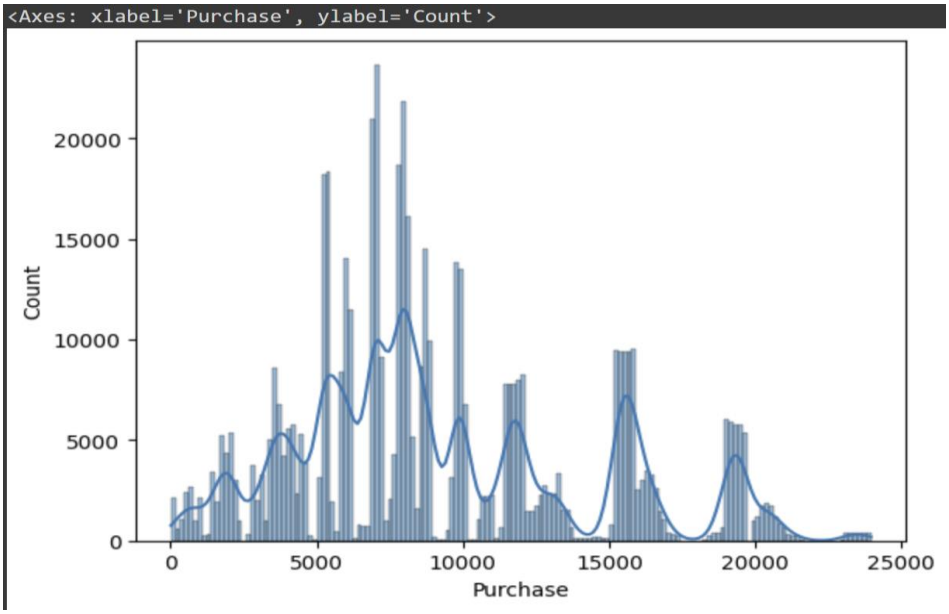
- 80 % of the users are between the age 18-50 (18-25: 18 %, 26-35: 40 %, 36-45: 20 %,46-50: 2%).
- 75 % of the users are Male & 25 % of them are Female.
- 60 % are Single & 40 % are Married.
- 35 % Staying in the city from 1 year, 18 % are from 2 years & 17 % are from 3 years.
- In Total there are 20 Product Categories.
- There are 20 different types of Occupations in the City.

3) Visual Analysis – (Univariate & Bivariate): --

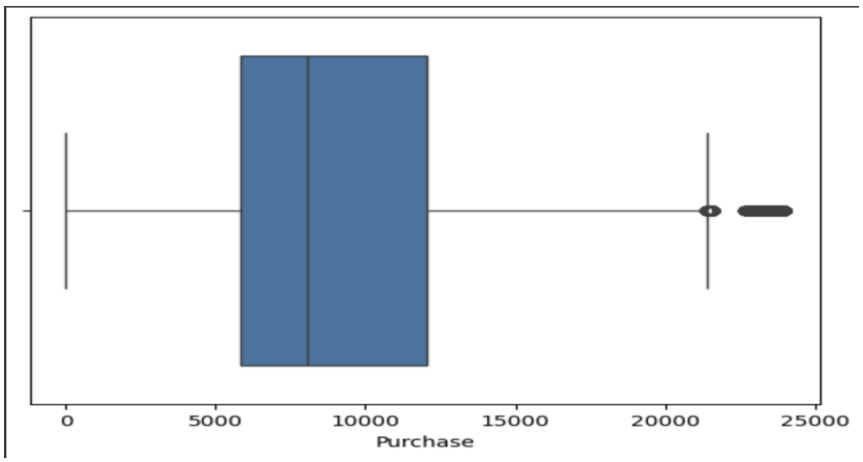
Uni-Variate: -

- For continuous Variable (s):- Boxplot & Histogram for Univariate Analysis :- Understanding the distribution of the data & detecting outliers for continuous variables.

```
sns.histplot(data = df, x = "Purchase", kde = True)
```



```
sns.boxplot(data = df, x = "Purchase", orient ="h")
plt.show()
```

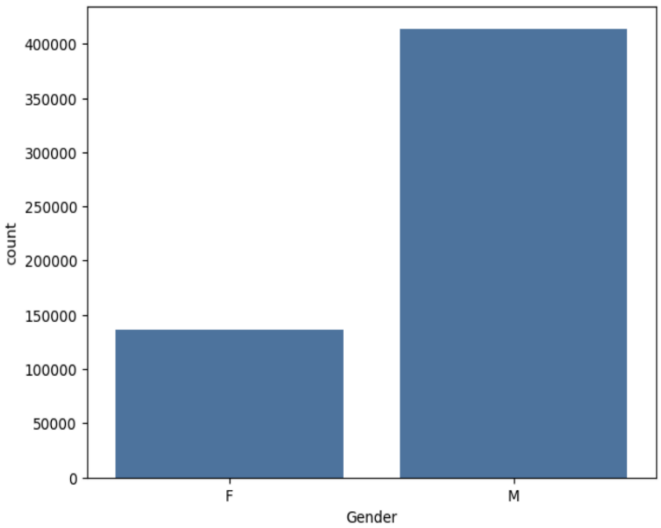


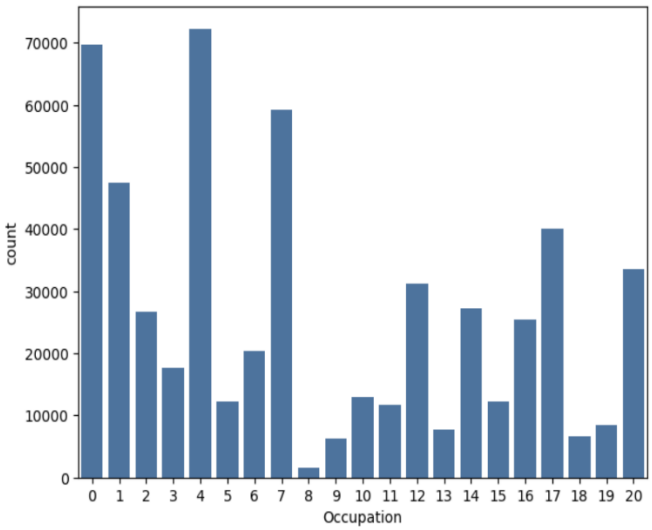
Observations: --

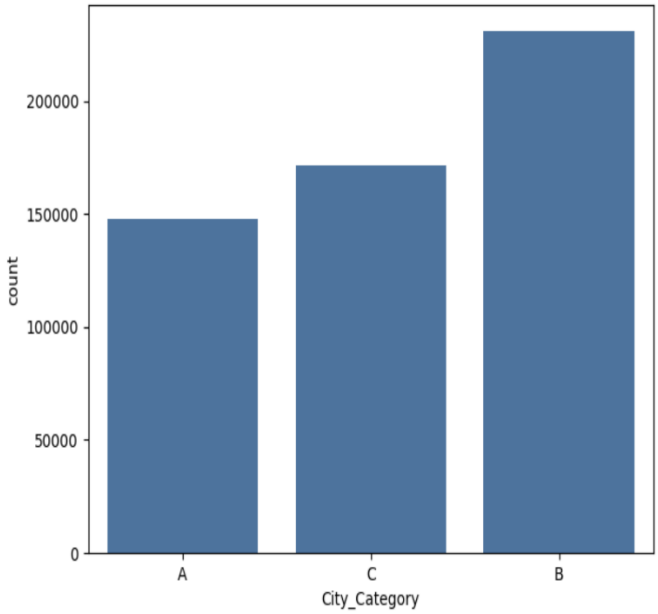
- Purchase is having the Outliers.

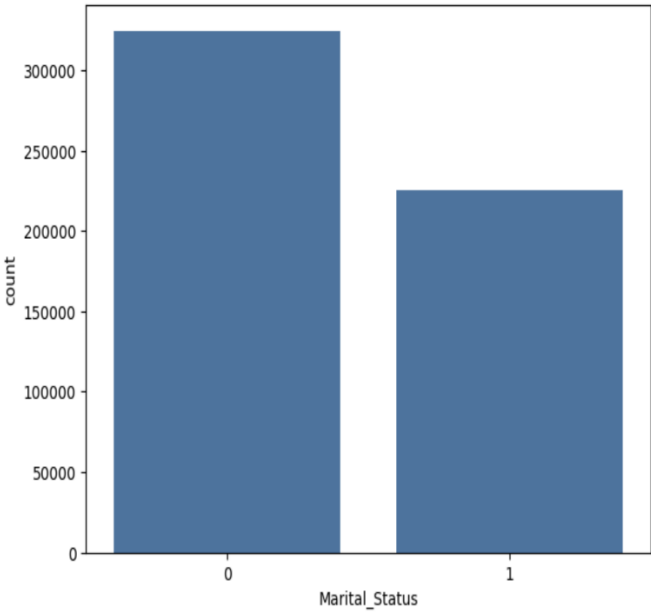
For Categorical Variable (s): Count Plot: - Understanding the distribution of the data for Categorical Variables.

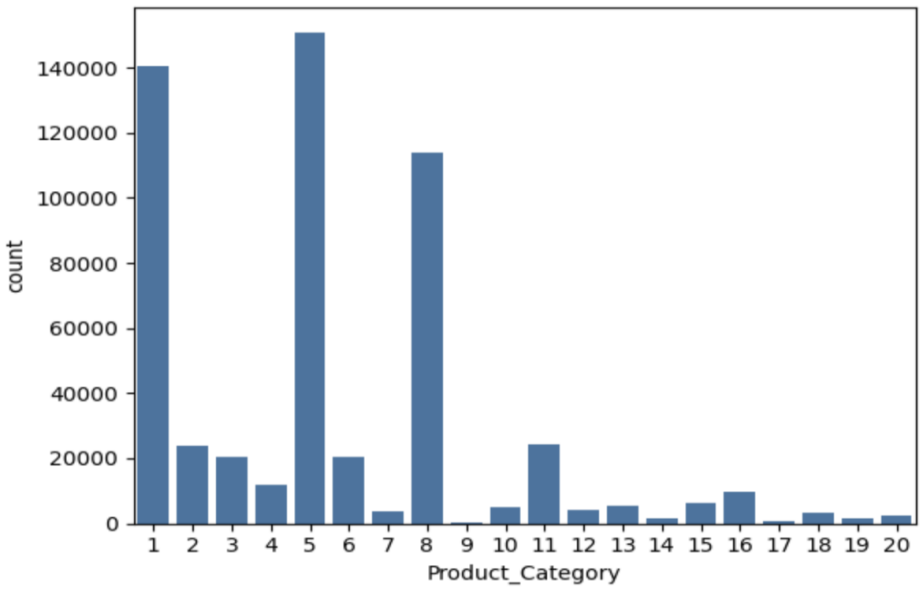
```
cat_cols = ["Gender","Occupation","City_Category","Marital_Status","Product_Category"]
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))
sns.countplot(data=df, x='Gender', ax=axs[0,0])
sns.countplot(data=df, x='Occupation', ax=axs[0,1])
sns.countplot(data=df, x='City_Category', ax=axs[1,0])
sns.countplot(data=df, x='Marital_Status', ax=axs[1,1])
plt.show()
sns.countplot(data = df,x = "Product_Category")
plt.show()
plt.figure(figsize=(10,8))
sns.countplot(data = df, x = "Product_Category")
plt.show()
```


count
400000
350000
300000
250000
200000
150000
100000
50000
0
F
M
Gender

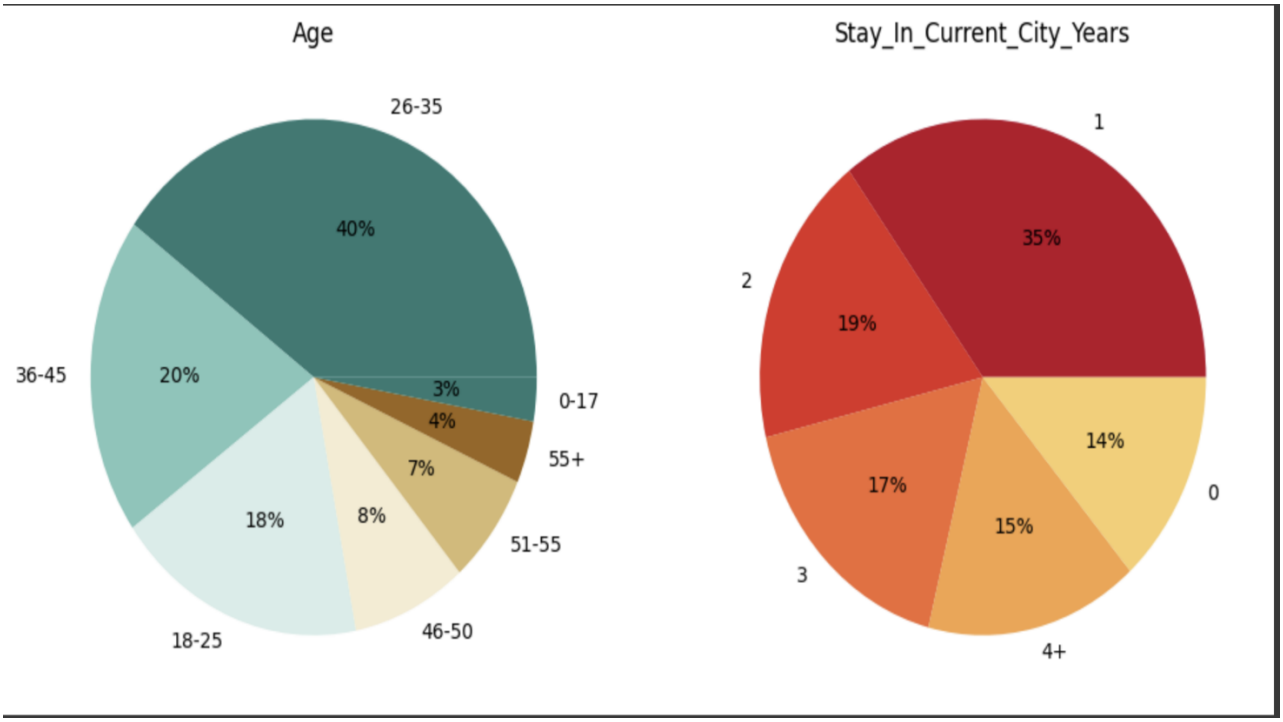

count
70000
60000
50000
40000
30000
20000
10000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Occupation


count
200000
150000
100000
50000
0
A
C
B
City_Category


count
300000
250000
200000
150000
100000
50000
0
0
1
Marital_Status


count
140000
120000
100000
80000
60000
40000
20000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Product_Category

```
fig, axs = plt.subplots(nrows = 1, ncols= 2, figsize = (12,8))
data = df['Age'].value_counts(normalize=True)*100
palette_color = sns.color_palette('BrBG_r')
axs[0].pie(x=data.values, labels=data.index, autopct='%.0f%%',
colors=palette_color)
axs[0].set_title("Age")
data = df['Stay_In_Current_City_Years'].value_counts(normalize=True)*100
palette_color = sns.color_palette('YlOrRd_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()
```

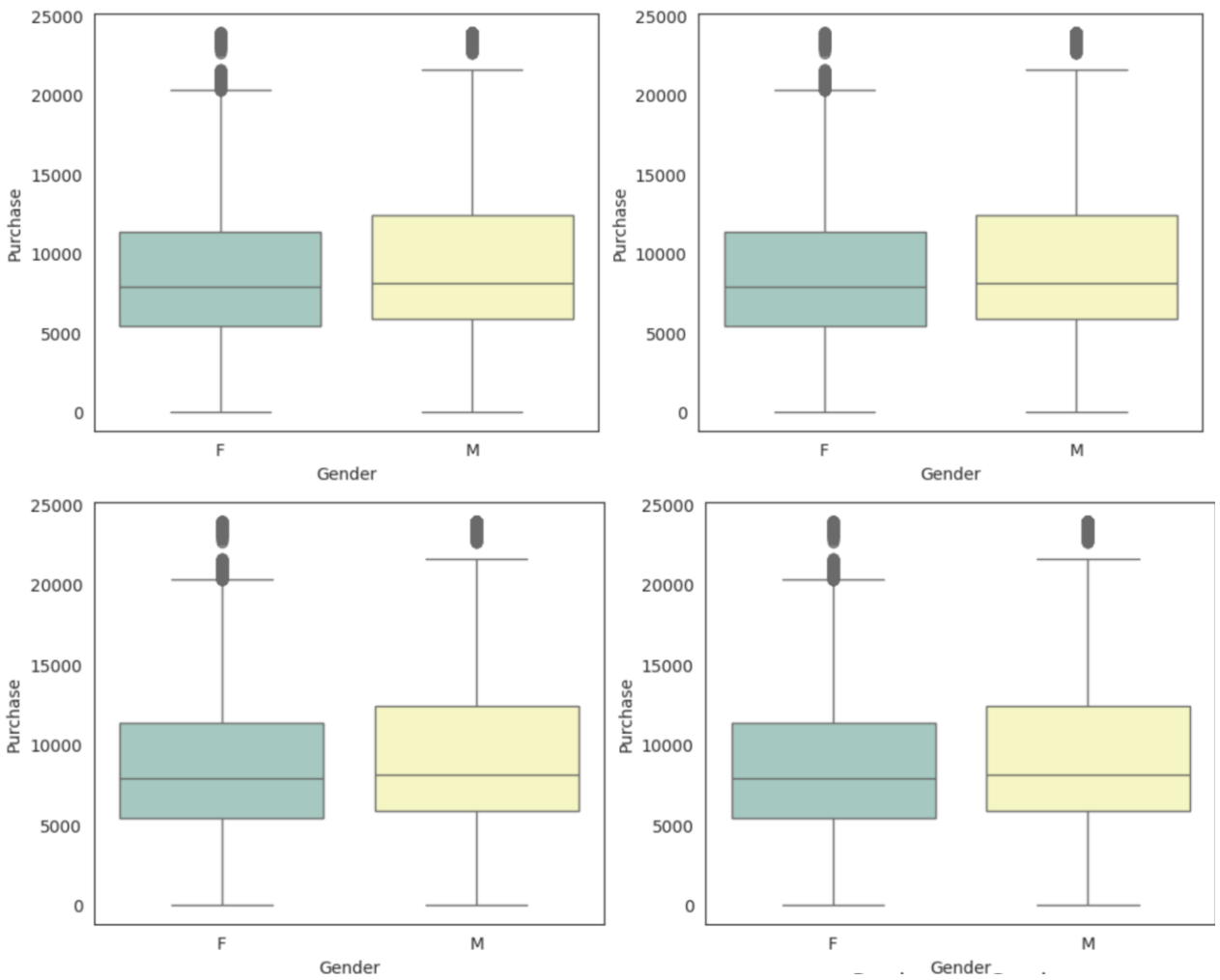


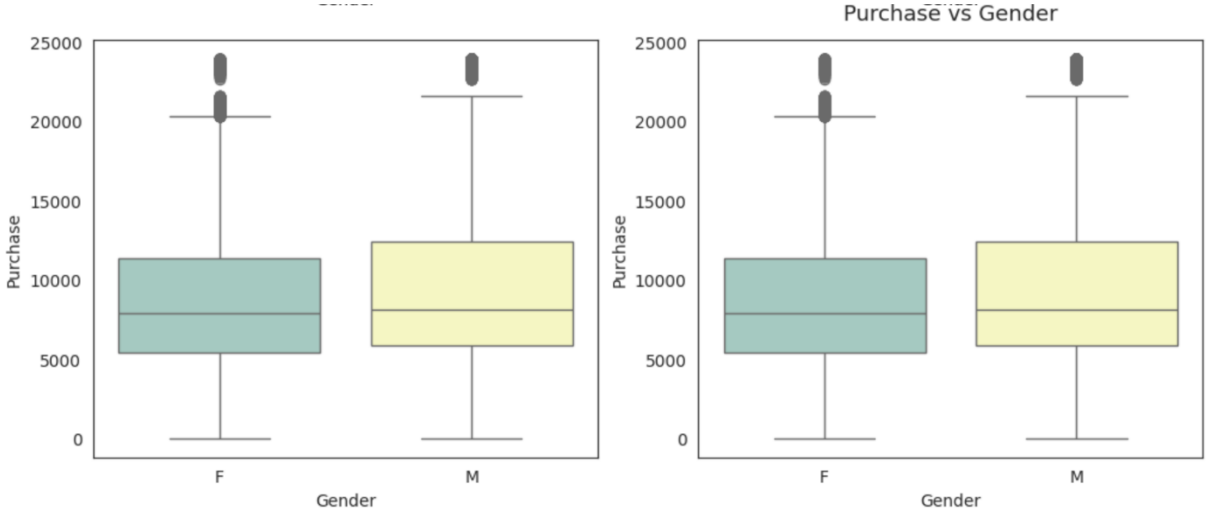
Observations: -

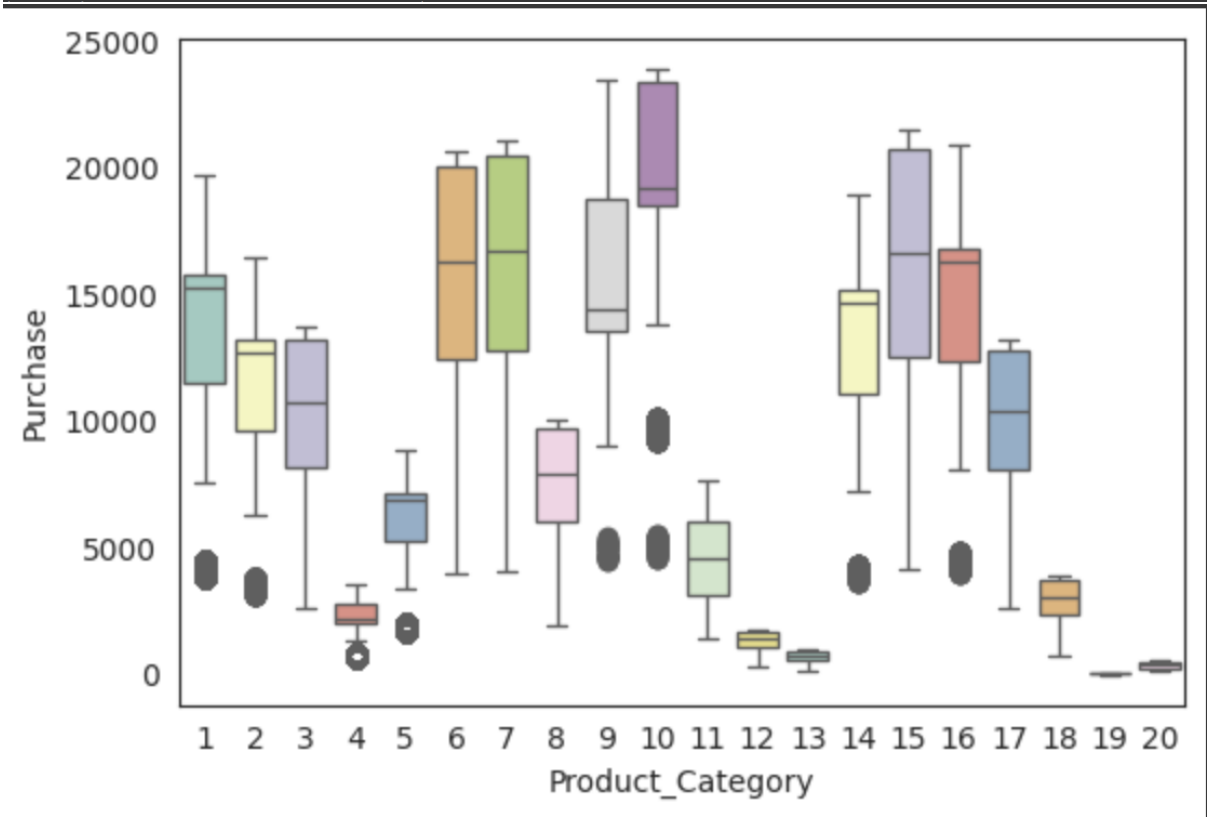
- Most of the Users are Male
- There are 20 different types of Occupations & Product Category.
- Most Users belongs to B City Category.
- Most Users are single as compared to Married.
- Product Category (-1, 5, 8 & 11) have highest purchasing frequency.

Bivariate: --

```
attrs = ['Gender', 'Age', 'Occupation', 'City_Category',
'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
sns.set_style("white")
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(12,10))
fig.subplots_adjust(top=1.3)
count = 0
for row in range(3):
  for col in range(2):
    sns.boxplot(data=df, y='Purchase', x=attrs[count], ax=axs[row,
col], palette='Set3')
axs[row,col].set_title(f"Purchase vs {attrs[count]}", pad=12,
fontsize=13)
count += 1
plt.show()
plt.figure(figsize=(10, 8))
sns.boxplot(data=df, y='Purchase', x=attrs[-1], palette='Set3')
plt.show()
```




Purchase vs Gender
Purchase
25000
20000
15000
10000
5000
0
F
M
Gender


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

4) Missing Value & Outliers Detections: ---

Missing Value: - (Checking Null Values)

```
print(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
```

Observations: -

- There are no Missing Values in the Data Set.

Finding Outliers Using Pandas Describe Function: --

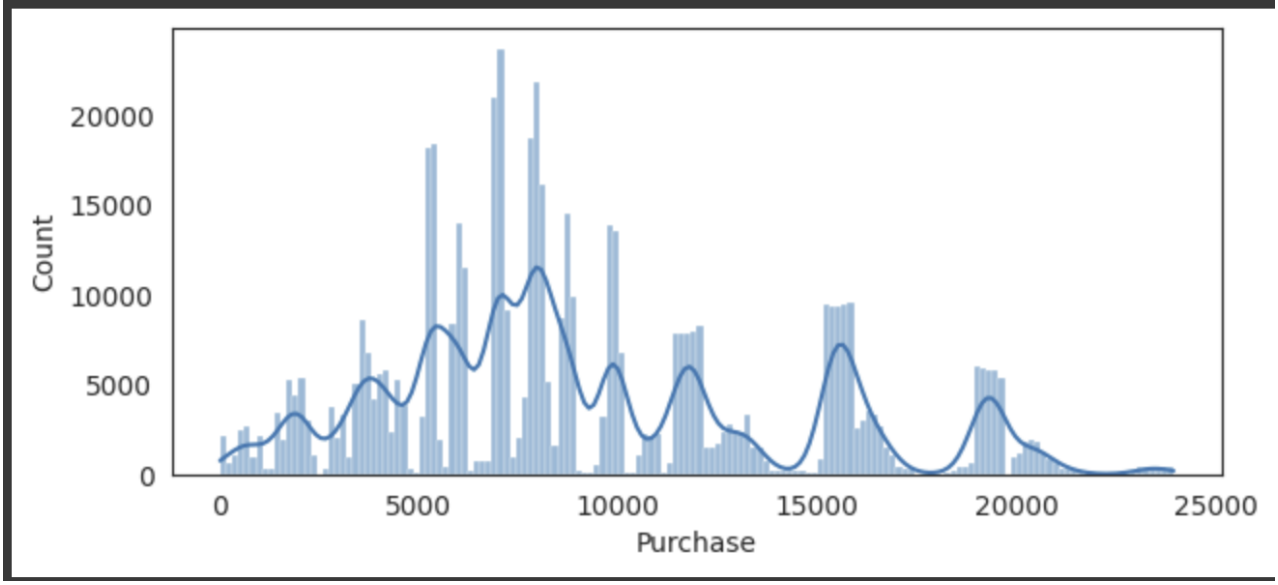
```
df.describe()
```

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

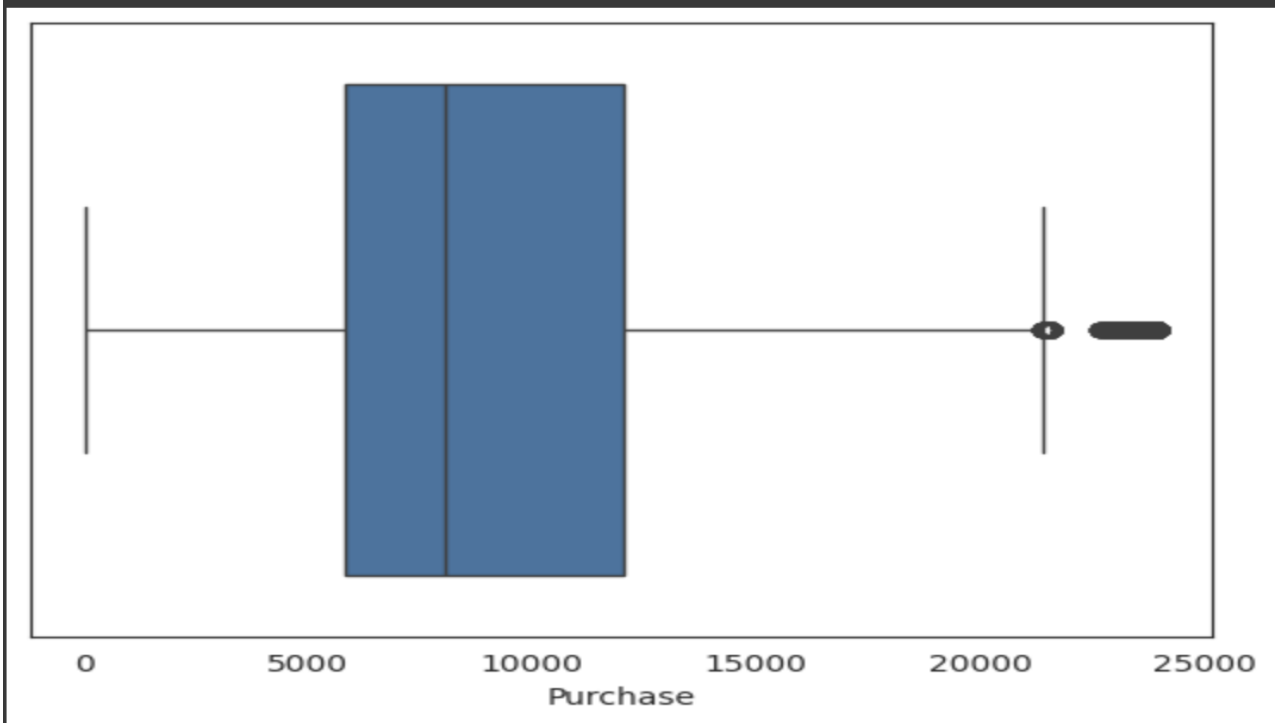
- **<u>Purchase Amount might have Outliers:</u> -** The Max Purchase amount is 23961, While it's mean is 9263.96 which is sensitive to the Outliers, but the fact that the mean is so small compared to the max value which indicates that the max value is an Outlier.

Visualize Outliers: --

```
plt.figure(figsize=(7, 3))
sns.histplot(data=df, x='Purchase', kde=True)
plt.show()
```




```
sns.boxplot(data=df, x='Purchase', orient='h')
plt.show()
```



Observations: -

- Purchase is having the Outliers.

Using The Convenient Pandas Quantile Function: ---

```
# Creating a Function to Find the Outliers using IQR:-
def find_outliers_IQR(df):
  q1=df.quantile(0.25)
  q3=df.quantile(0.75)
  IQR=q3-q1
  outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]
  return outliers
outliers = find_outliers_IQR(df["Purchase"])
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
```

Now Answering the Questions: -

Q1) Are Women Spending more money per transaction than mean.? Why or why not.?

Average amount spends per customer for Male & Female: -

```
amt_df = df.groupby(['User_ID', 'Gender'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df.head()
```

	User_ID	Gender	Purchase
0	1000001	F	334093
1	1000002	M	810472
2	1000003	M	341635
3	1000004	M	206468
4	1000005	M	821001

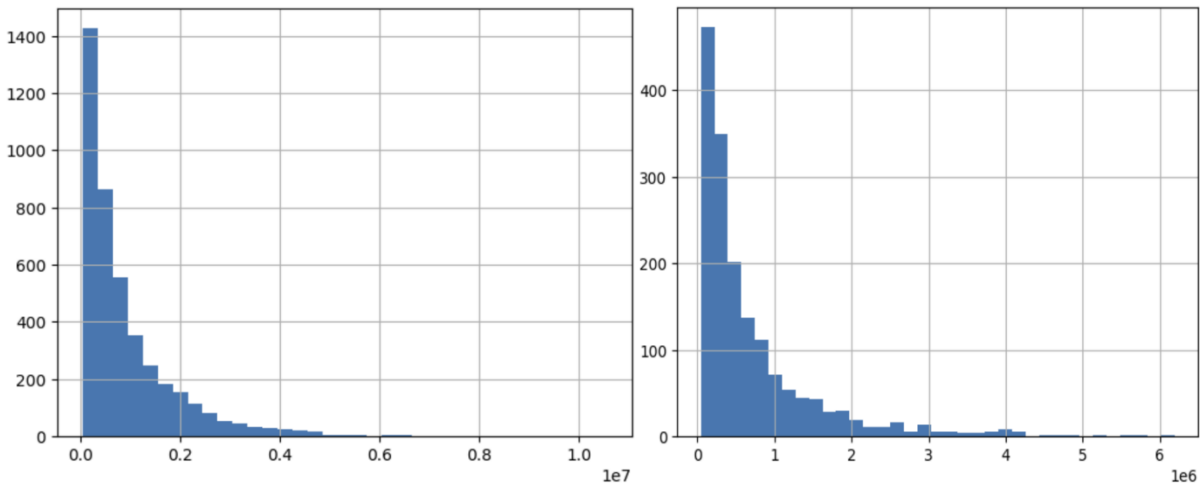
Gender Wise Value_Count:-

```
avg_amt_df = amt_df["Gender"].value_counts()
avg_amt_df
```

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

Histogram of Average amount spending on each customer Male & Female: --

```
amt_df[amt_df['Gender']=='M']['Purchase'].hist(bins=35)
plt.show()
amt_df[amt_df['Gender']=='F']['Purchase'].hist(bins=35)
plt.show()
```



```
male_avg = amt_df[amt_df['Gender']=='M']['Purchase'].mean()
female_avg = amt_df[amt_df['Gender']=='F']['Purchase'].mean()
print("Average amount spend by Male customers:{:.2f}".format(male_avg))
print("Average amount spend by Female customers:{:.2f}".format(female_avg))
```

```
Average amount spend by Male customers:925344.40
Average amount spend by Female customers:712024.39
```

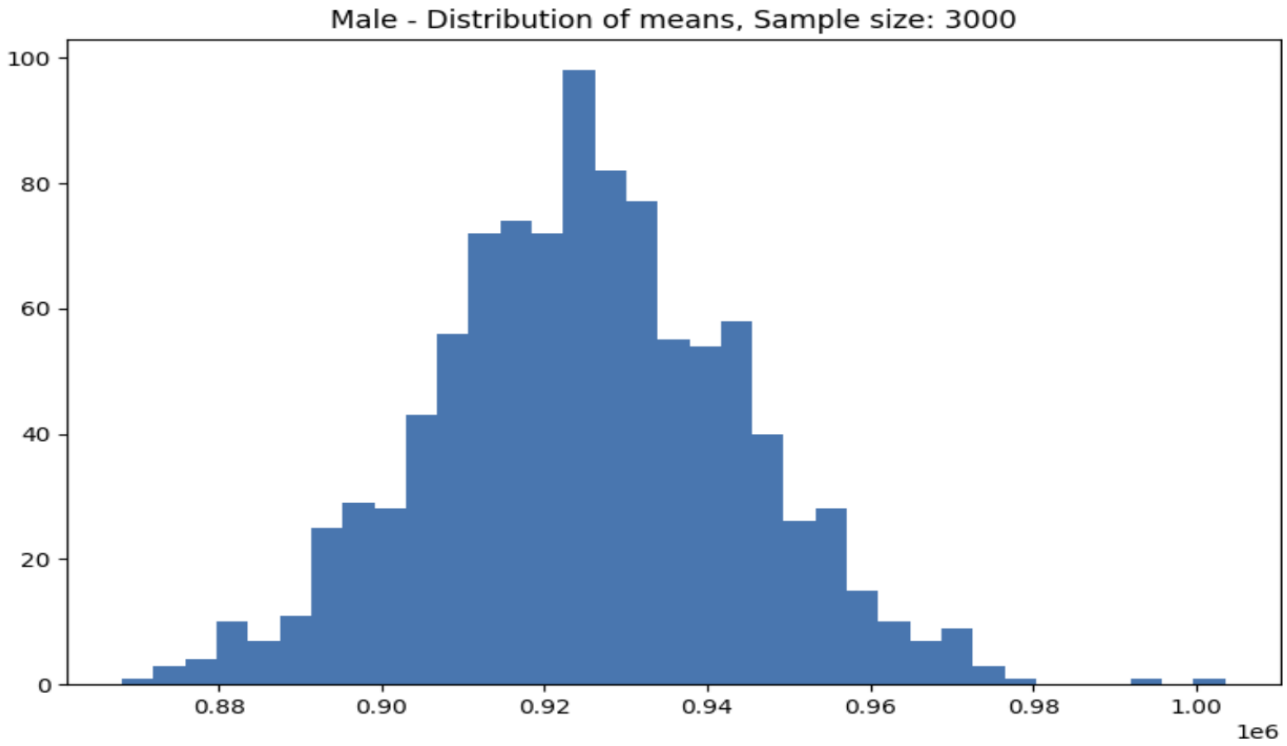
Observation: -

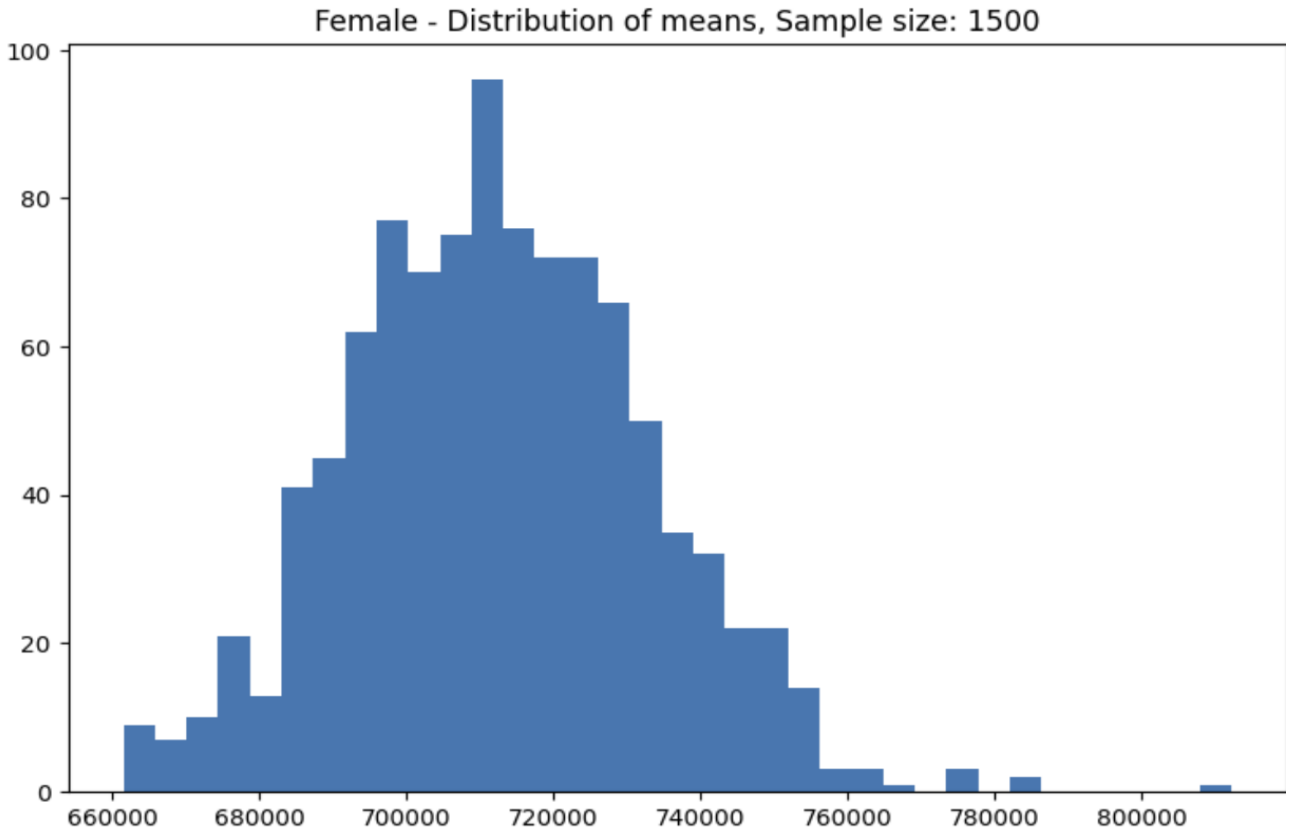
- Male customers spend more money than Female customers.

Q2) Confidence Intervals & Distribution of the Mean of the expenses by female & male customers.?

```
male_df = amt_df[amt_df['Gender']=='M']
female_df = amt_df[amt_df['Gender']=='F']
genders = ["M", "F"]
male_sample_size = 3000
female_sample_size = 1500
num_repitions = 1000
male_means = []
female_means = []
for _ in range(num_repitions):
  male_mean = male_df.sample(male_sample_size,replace=True)['Purchase'].mean()
  female_mean = female_df.sample(female_sample_size,replace=True)['Purchase'].mean()
  male_means.append(male_mean)
  female_means.append(female_mean)

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
axis[0].hist(male_means, bins=35)
axis[1].hist(female_means, 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()
```


Male - Distribution of means, Sample size: 3000
100
80
60
40
20
0
0.88
0.90
0.92
0.94
0.96
0.98
1.00
1e6


Female - Distribution of means, Sample size: 1500
100
80
60
40
20
0
660000
680000
700000
720000
740000
760000
780000
800000

```
print("Population mean - Mean of sample means of amount spend for Male:{:.2f}".format(np.mean(male_means)))
print("Population mean - Mean of sample means of amount spend for Female: {:.2f}".format(np.mean(female_means)))
print("\nMale - Sample mean: {:.2f} Sample std:{:.2f}".format(male_df['Purchase'].mean(), male_df['Purchase'].std()))
print("Female - Sample mean: {:.2f} Sample std:{:.2f}".format(female_df['Purchase'].mean(),female_df['Purchase'].std()))
```

```
Population mean - Mean of sample means of amount spend for Male:925174.41
Population mean - Mean of sample means of amount spend for Female: 712720.69

Male - Sample mean: 925344.40 Sample std:985830.10
Female - Sample mean: 712024.39 Sample std:807370.73
```

Observations: -

Now using the **Central Limit Theorem** for the population, we can say that

1. Average amount spent by male customers is **9,26,341.86**
2. Average amount spent by the female customers is **7,11,704.09**

Q3) Are confidence intervals of average male & female spending overlapping? How can Walmart leverage this conclusion to make changes or improvements.?

```
male_margin_of_error_clt = 1.96*male_df['Purchase'].std()/np.sqrt(len(male_df))
male_sample_mean = male_df['Purchase'].mean()
male_lower_lim = male_sample_mean - male_margin_of_error_clt
male_upper_lim = male_sample_mean + male_margin_of_error_clt
female_margin_of_error_clt = 1.96*female_df['Purchase'].std()/np.sqrt(len(female_df))
female_sample_mean = female_df['Purchase'].mean()
female_lower_lim = female_sample_mean - female_margin_of_error_clt
female_upper_lim = female_sample_mean + female_margin_of_error_clt
print("Male confidence interval of means: ({:.2f},{:.2f})".format(male_lower_lim, male_upper_lim))
print("Female confidence interval of means: ({:.2f},{:.2f})".format(female_lower_lim, female_upper_lim))
```

```
Male confidence interval of means: (895617.83,955070.97)
Female confidence interval of means: (673254.77,750794.02)
```

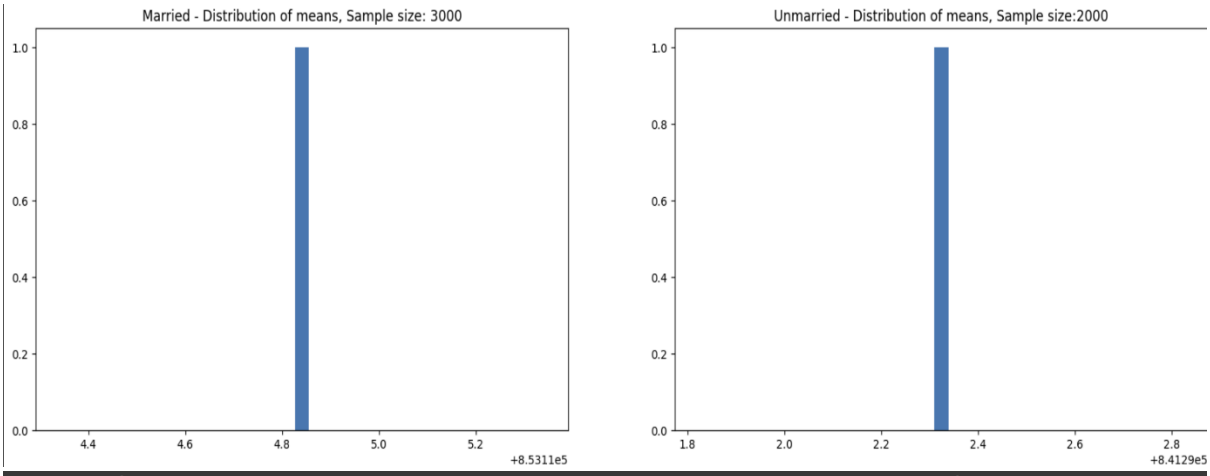
Now we can infer about the population that **95% of the times: -**

1. Average amount spent by the male customer will lie in between (895617.83, 955070.97)
2. Average amount spent by the female customers will lie in between (673254.77, 750794.02)

Q4) Result when the same activity is performed for Married vs Unmarried.?

```
amt_df
amt_df = df.groupby(['User_ID', 'Marital_Status'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
amt_df['Marital_Status'].value_counts()
marid_samp_size = 3000
unmarid_sample_size = 2000
num_repitions = 1000
marid_means = []
unmarid_means = []
```

```
for _ in range(num_repitions):
  marid_mean = amt_df[amt_df['Marital_Status']==1].sample(marid_samp_size,replace=True)['Purchase'].mean()
unmarid_mean = amt_df[amt_df['Marital_Status']==0].sample(unmarid_sample_size,replace=True)['Purchase'].mean()
marid_means.append(marid_mean)
unmarid_means.append(unmarid_mean)
fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
axis[0].hist(marid_means, bins=35)
axis[1].hist(unmarid_means, bins=35)
axis[0].set_title("Married - Distribution of means, Sample size: 3000")
axis[1].set_title("Unmarried - Distribution of means, Sample size:2000")
plt.show()
print("Population mean - Mean of sample means of amount spend for Married: {:.2f}".format(np.mean(marid_means)))
print("Population mean - Mean of sample means of amount spend for Unmarried: {:.2f}".format(np.mean(unmarid_means)))
print("\nMarried - Sample mean: {:.2f} Sample std: {:.2f}".format(amt_df[amt_df['Marital_Status']==1]['Purchase'].mean(),
                                                                 amt_df[amt_df['Marital_Status']==1]['Purchase'].std()))
print("Unmarried - Sample mean: {:.2f} Sample std:{:.2f}".format(amt_df[amt_df['Marital_Status']==0]['Purchase'].mean(),
                                                                 amt_df[amt_df['Marital_Status']==0]['Purchase'].std()))
for val in ["Married", "Unmarried"]:
  new_val = 1 if val == "Married" else 0
  new_df = amt_df[amt_df['Marital_Status']==new_val]
  margin_of_error_clt = 1.96*new_df['Purchase'].std()/np.sqrt(len(new_df))
  sample_mean = new_df['Purchase'].mean()
  lower_lim = sample_mean - margin_of_error_clt
  upper_lim = sample_mean + margin_of_error_clt
  print("{} confidence interval of means: ({:.2f},{:.2f})".format(val, lower_lim, upper_lim))
```




```
Population mean - Mean of sample means of amount spend for Married: 853114.84
Population mean - Mean of sample means of amount spend for Unmarried: 841292.32

Married - Sample mean: 843526.80 Sample std: 935352.12
Unmarried - Sample mean: 880575.78 Sample std:949436.25
Married confidence interval of means: (806668.83,880384.76)
Unmarried confidence interval of means: (848741.18,912410.38)
```

Q5) Results When the same activity is performed for Age.?

Calculating the average amount spend by age.

```
amt_df = df.groupby(['User_ID', 'Age'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
amt_df['Age'].value_counts()
sample_size = 200
num_repitions = 1000
all_means = {}
age_intervals = ['26-35', '36-45', '18-25', '46-50', '51-55', '55+','0-17']
for age_interval in age_intervals:
  all_means[age_interval] = []
for age_interval in age_intervals:
  for _ in range(num_repitions):
    mean = amt_df[amt_df['Age']==age_interval].sample(sample_size,replace=True)['Purchase'].mean()
    all_means[age_interval].append(mean)
for val in ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']:
  new_df = amt_df[amt_df['Age']==val]
  margin_of_error_clt = 1.96*new_df['Purchase'].std()/np.sqrt(len(new_df))
  sample_mean = new_df['Purchase'].mean()
  lower_lim = sample_mean - margin_of_error_clt
  upper_lim = sample_mean + margin_of_error_clt
  print("For age {} --> confidence interval of means: ({:.2f},{:.2f})".format(val, lower_lim, upper_lim))
```

```
For age 26-35 --> confidence interval of means: (945034.42,1034284.21)
For age 36-45 --> confidence interval of means: (823347.80,935983.62)
For age 18-25 --> confidence interval of means: (801632.78,908093.46)
For age 46-50 --> confidence interval of means: (713505.63,871591.93)
For age 51-55 --> confidence interval of means: (692392.43,834009.42)
For age 55+ --> confidence interval of means: (476948.26,602446.23)
For age 0-17 --> confidence interval of means: (527662.46,710073.17)
```

Insights: --

- 80% of the users are between the age of 18-50 (40%: - 26-35, 18%: - 18-25, 20%: - 36-45).
- 75% of the users are Male & 25% are Female.
- 60% Single & 40% Married
- 35% Staying in the city from 1 year, 18% from 2 years, & 17% from 3 Years.
- Total of 20 Product Categories are there.
- There are 20 different types of occupations in the city.
- Most of the users are Male.

- There are 20 different types of Occupation & Product Category
- More Users belongs to B City Category.
- More Users are Single as compared to Married.
- Product Category – 1,5,8 & 11 have the highest purchasing frequency.
- Average amount spends by Male Customers:->**925344.40**
- Average amount spends by Female Customers:->**712024.39**

Confidence Interval by Gender: -

Now using the Central Limit Theorem for the Population:

1. Average Amount spend by Male Customers is 9,26,341.86.
2. Average Amount spend by Female Customers is 7,11,704.09.

Now we can infer about the Population that 95% of the times:

1. Average Amount spend by the Male Customers will lie in between (895617.83, 955070.97)
2. Average Amount spend by female customers will lie in between (673254.77, 750794.02)

Confidence Interval by Marital Status:

1. Married Confidence interval of means (806668.83, 880384.76).
2. Unmarried Confidence interval of means (848741.18, 912410.38)

Confidence Interval By Age:

1. For age 26-35→ Confidence Interval of means (945034.42, 1034284.21).
2. For age 36-45→ Confidence Interval of means (823347.80, 935983.62)
3. For age 18-25→ Confidence Interval of means (801632.78, 908093.46)
4. For age 46-50→ Confidence Interval of means (713505.63, 871591.93)
5. For age 51-55→ Confidence Interval of means (692392.43, 834009.42)
6. For age 55+ → Confidence Interval of means (476948.26, 602446.23)
7. For age 0-17→ Confidence Interval of means (527662.46, 710073.17)

Recommendations: -

1. Men spent more money than women, so company should focus on retaining the male customers and getting more male customers.

2. Product Category - 1, 5, 8, & 11 have highest purchasing frequency. it means these are the products in these categories are liked more by customers. Company can focus on selling more of these products or selling more of the products which are purchased less.

3. Unmarried customers spend more money than married customers, so company should focus on acquisition of Unmarried customers.

4. Customers in the age 18-45 spend more money than the others, so company should focus on acquisition of customers who are in the age 18-45.

5. Male customers living in City_Category C spend more money than other male customers living in B or C, selling more products in the City_Category C will help the company increase the revenue.

6. A targeted Campaign to target 46-60 age group, in order to increase their spending. Look at the product purchase category of the age group 46-60 increase discount on those product categories.