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).

Defining Problem Statement and Analysing basic metrics

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

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

```
df.head(10)
```

	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
5	1000003	P00193542	M	26-35	15	A	3	0	1	15227
6	1000004	P00184942	M	46-	7	B	2	1	1	19215

```
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
```

Occupation, Marital_Status and Product Category needs to be in object type

```
df['Occupation']=df['Occupation'].astype('object')
df['Marital_Status']=df['Marital_Status'].astype('object')
df['Product_Category']=df['Product_Category'].astype('object')
```

```
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  object
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  object
 8   Product_Category            550068 non-null  object
 9   Purchase                    550068 non-null  int64
dtypes: int64(2), object(8)
memory usage: 42.0+ MB
```

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%
User_ID	550068.0	1.003029e+06	1727.591586	1000001.0	1001516.0	1003077.0	1004478.0
Purchase	550068.0	9.263969e+03	5023.065394	12.0	5823.0	8047.0	12054.0

```
df.describe(include=object).T
```

	count	unique	top	freq
Product_ID	550068	3631	P00265242	1880
Gender	550068	2	M	414259
Age	550068	7	26-35	219587
Occupation	550068	21	4	72308
City_Category	550068	3	B	231173
Stay_In_Current_City_Years	550068	5	1	193821
Marital_Status	550068	2	0	324731
Product_Category	550068	20	5	150933

Observations:

- 2 unique genders where Male are more
- 7 unique Age categories where 26-35 are most
- 21 unique Occupation categories where 4 number occupation are more
- 3 unique City_category where B city are more
- 2 unique Marital_Status where 0 are more (more unmarried)
- 20 unique Product_Category where 5th product category are most

Value Count and Frequency count of categorical columns

```
col=list(df.columns)
col.remove('Purchase')
col.remove('User_ID')
col.remove('Product_ID')
temp=[]
unique_categories=[]
for i in col:
  temp=df[i].unique()
  unique_categories.append(temp)
```

```
Unique_df=pd.DataFrame(unique_categories)
```

```
Unique_df.index=col
```

```
Unique_df
```

	0	1	2	3	4	5	6	7	8	9	...
Gender	F	M	None	None	None	None	None	NaN	NaN	NaN	...
Age	0-17	55+	26-35	46-50	51-55	36-45	18-25	NaN	NaN	NaN	...
Occupation	10	16	15	7	20	9	1	12.0	17.0	0.0	...
City_Category	A	C	B	None	None	None	None	NaN	NaN	NaN	...
Stay_In_Current_City_Years	2	4+	3	1	0	None	None	NaN	NaN	NaN	...
Marital_Status	0	1	None	None	None	None	None	NaN	NaN	NaN	...
Product_Category	3	1	12	8	5	4	2	6.0	14.0	11.0	...

7 rows × 21 columns

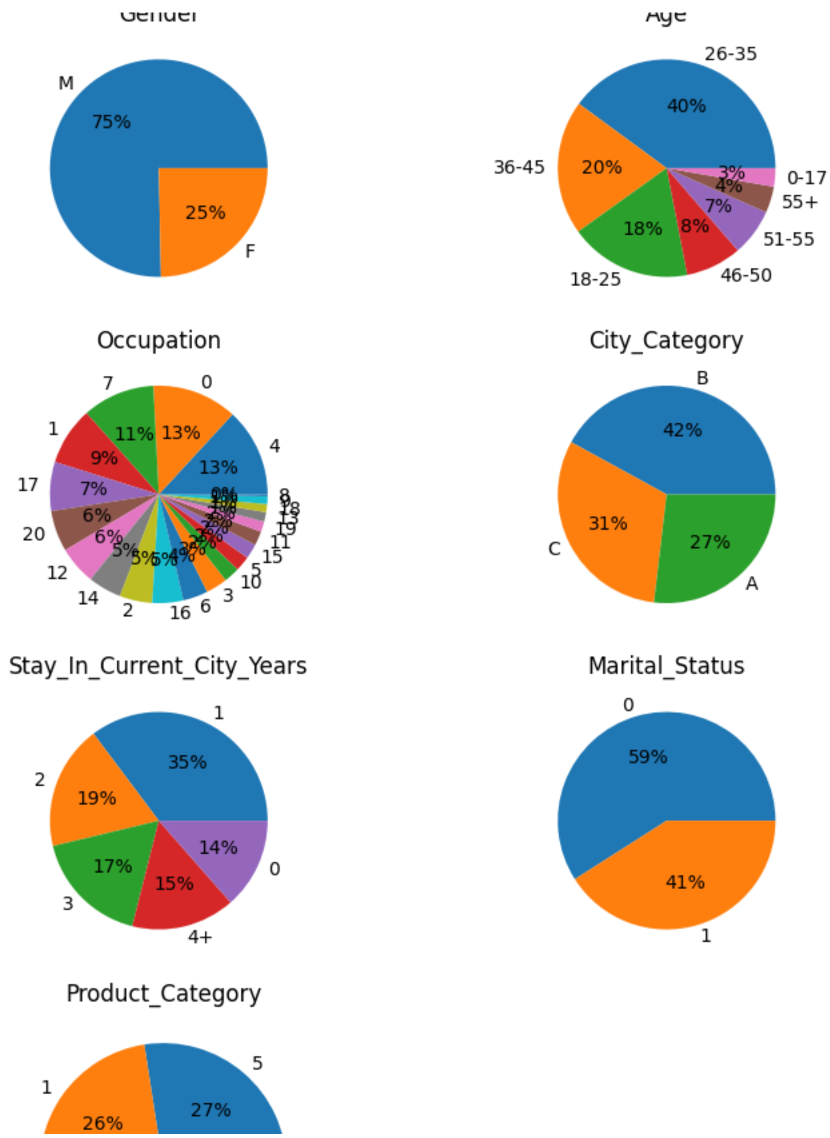
```
categorical_cols = ['Gender', 'Age', 'Occupation', 'City_Category','Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
df[categorical_cols].melt().groupby(['variable','value'])[['value']].count()/len(df)
```

	14	0.049647
	15	0.022115
	16	0.046123
	17	0.072796
	18	0.012039
	19	0.015382
	20	0.061014
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
	19	0.002914
	20	0.004636
Stay_In_Current_City_Years	**0**	0.135252
	1	0.352358
	2	0.185137
	4+	0.154028

Categorical columns visualization

```
categorical_cols=categorical_cols[:6]
c_cols=np.array(categorical_cols).reshape(3,2)


fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(9,9))
for i in range(3):
  for j in range(2):
    data=df[c_cols[i][j]].value_counts(normalize=True)*100
    axs[i,j].pie(x=data.values,labels=data.index,autopct='%.0f%%')
    axs[i,j].set_title(c_cols[i][j])
plt.figure(figsize=(3,3))
data=df['Product_Category'].value_counts(normalize=True)*100
plt.pie(x=data.values,labels=data.index,autopct='%.0f%%')
plt.title('Product_Category')
plt.show()
```



Observations

- ~ 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45)
- 75% of the users are Male and 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
- Product_Category - 1, 5 & 8 have highest purchasing frequency.
- There are 20 different types of occupations in the city
- More users belong to B City_Category

Unique categories are well ranged without any gaps

Checking Null Values

```
df.isnull().sum()
```

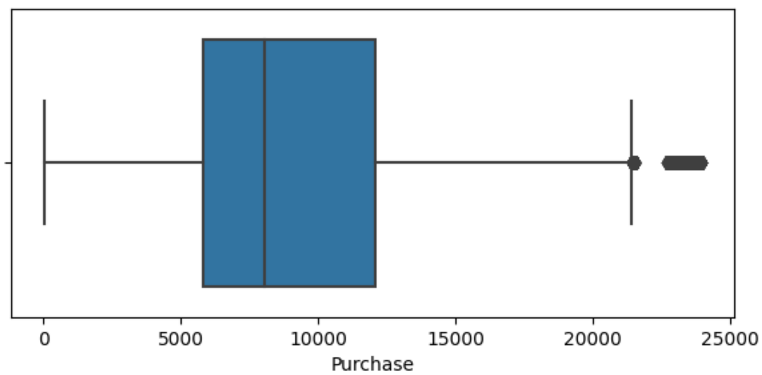
```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category              0
Purchase                      0
dtype: int64
```

No null values

Visual Analysis

Only Purchase is numerical column where outliers might be there

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



Observation:

- Purchase have outliers on right side

```
def find_outliers_IQR(df):
  q1=df.quantile(0.25)
  q3=df.quantile(0.75)
  IQR=q3-q1
  #as the outliers are on right side we will check right side outliers
  outliers = df[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()))
```

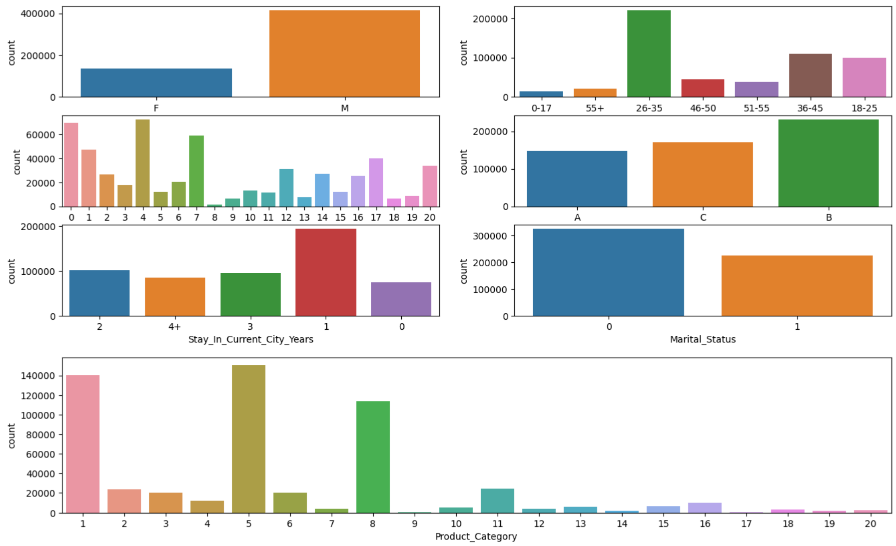
```
number of outliers: 2677
max outlier value:23961
min outlier value: 21401
```

Categorical variables

```
col1=df.columns
col1=col1[2:8]
col1=np.array(col1)
col1=col1.reshape(3,2)
```

```
fig,axs=plt.subplots(nrows=3,ncols=2,figsize=(16,6))
for i in range(3):
  for j in range(2):
    sns.countplot(data=df,x=col1[i][j],ax=axs[i,j])

plt.figure(figsize=(16,3))
sns.countplot(data=df,x='Product_Category')
plt.show()
```

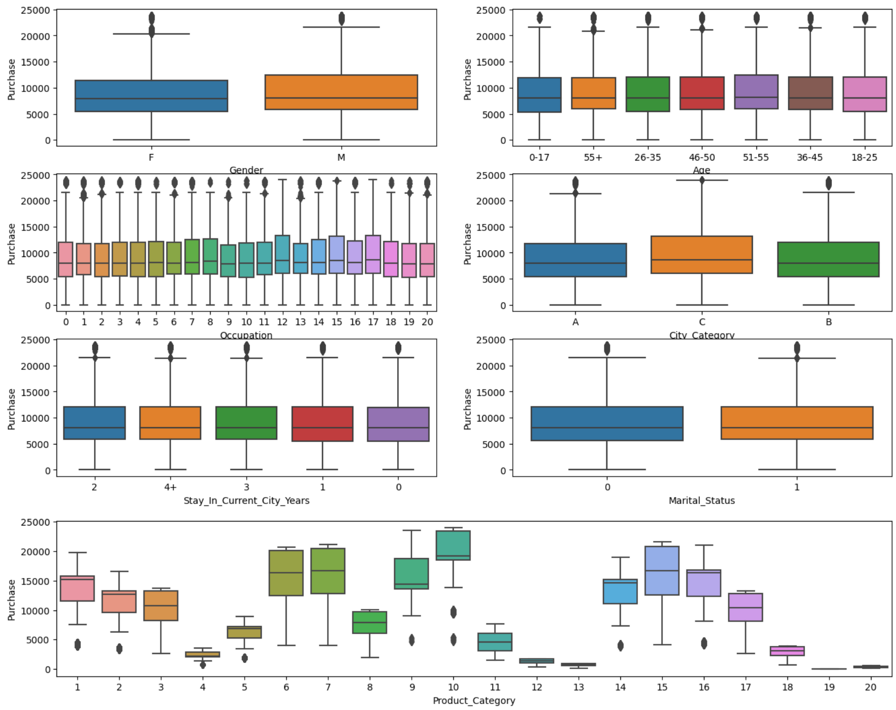


Observations:

- Most of the users are Male
- There are 20 different types of Occupation and Product_Category
- More users belong to B City_Category
- More users are Single as compare to Married
- Product_Category - 1, 5 & 8 have highest purchasing frequency.
- 26-35 age group are more frequent buyers

```
col2=df.columns
col2=col2[2:8]
```

```
fig,axs=plt.subplots(nrows=3,ncols=2,figsize=(16,9))
count=0
for i in range(3):
  for j in range(2):
    sns.boxplot(data=df,x=col2[count],y='Purchase',ax=axs[i,j])
    count+=1
plt.show()
plt.figure(figsize=(16,3))
sns.boxplot(data=df,x='Product_Category',y='Purchase')
plt.show()
```



Answers of questions

1. Are women spending more money per transaction than men? Why or Why not? Average amount spends per customer for Male and Female

```
df.head()
```

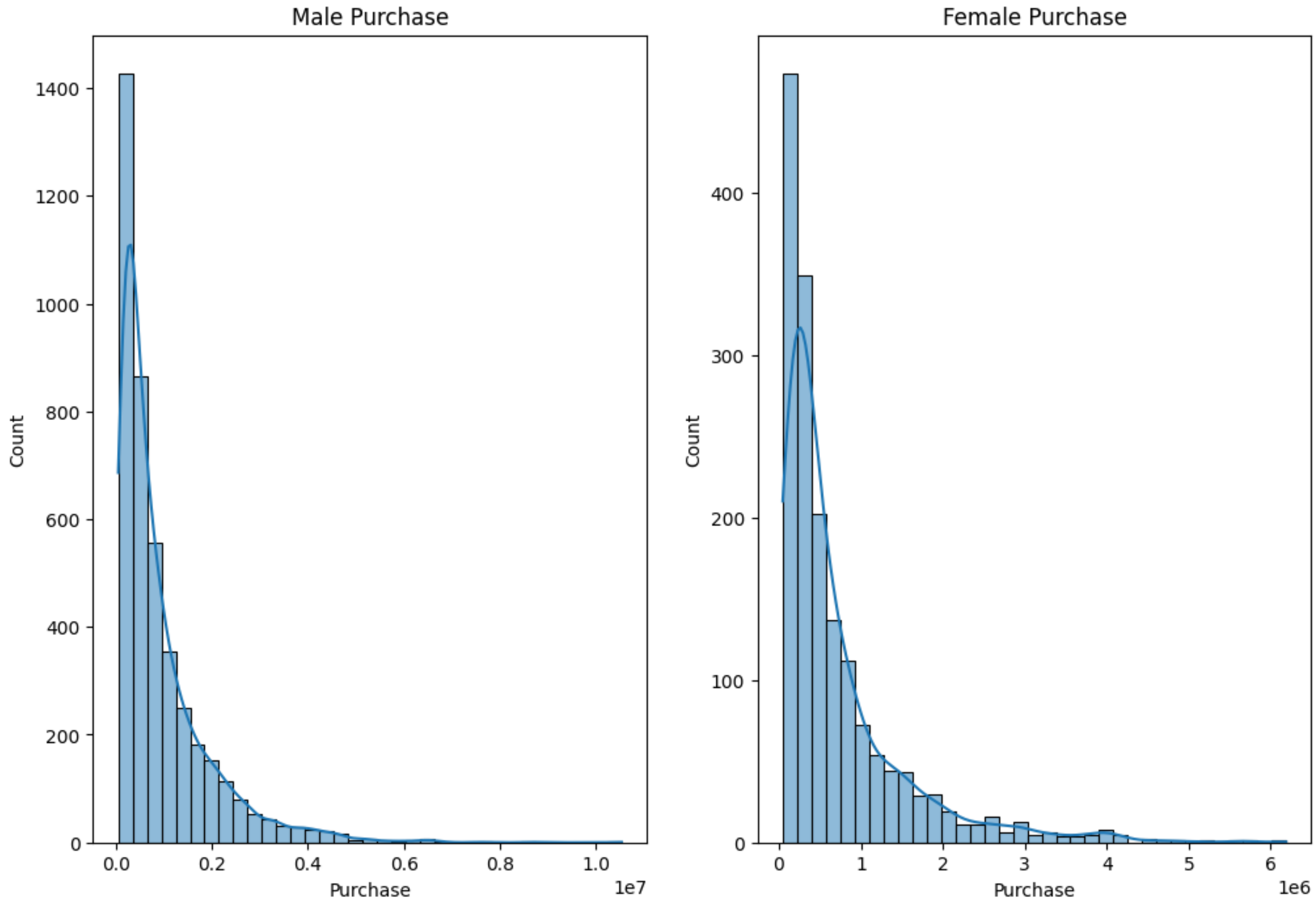
	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422

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

```
df1=data[data['Gender']=='M']
df2=data[data['Gender']=='F']
fig, axs=plt.subplots(nrows=1,ncols=2,figsize=(12,8))
sns.histplot(data=df1,x='Purchase',kde=True,bins=35,ax=axs[0])
axs[0].set_title('Male Purchase')
sns.histplot(data=df2,x='Purchase',kde=True,bins=35,ax=axs[1])
axs[1].set_title('Female Purchase')
```

```
Text(0.5, 1.0, 'Female Purchase')
```




```
male_avg=df1['Purchase'].mean()
m_s=df1["Purchase"].std()
female_avg=df2['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 on average on a unit

```
df1=df[df['Gender']=='M']
df2=df[df['Gender']=='F']
male_top10_product_cat=df1.groupby('Product_Category')['Purchase'].mean().nlargest(10)
```

```
female_top10_product_cat=df2.groupby('Product_Category')['Purchase'].mean().nlargest(10)
```

```
male_top10_product_cat,female_top10_product_cat
```

```
(Product_Category
 10    19670.731264
 7     16355.789777
 6     15907.851009
 9     15498.888235
 15    14797.431350
 16    14793.384056
 1     13608.164721
 14    12722.321111
 2     11203.590520
 17    10209.732558
 Name: Purchase, dtype: float64,
 Product_Category
 10    19692.076592
 7     16394.853659
```

```
9     15724.314286
6     15596.428164
15    14695.326960
16    14681.491257
14    13747.362761
1     13597.162619
2     11407.496819
3     10262.656677
Name: Purchase, dtype: float64)
```

Observation:

- Top 10 average price for product category for male and female are quite similar just male buy 17th category more and female buy 3rd category more then other

```
col2
```

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

```
for i in col2:
  male_top10=df1.groupby(i)['Purchase'].mean().nlargest(10)
  female_top10=df2.groupby(i)['Purchase'].mean().nlargest(10)
  print(i)
  print(f'Male Top 10 {i}{male_top10},Female Top 10 {i}{female_top10}')
```

```
0-17     8338.771985
Name: Purchase, dtype: float64
Occupation
Male Top 10 OccupationOccupation
12    9876.847492
15    9872.778721
17    9851.727696
14    9804.566923
8     9584.729114
7     9493.818898
13    9485.148154
16    9477.371520
5     9446.089083
4     9435.676366
Name: Purchase, dtype: float64,Female Top 10 OccupationOccupation
18    10074.608696
17     9543.435734
15     9394.894979
8      9361.451524
12     9155.953301
7      9092.302553
11     9090.800000
6      9078.405882
3      9055.138149
16     8965.212320
Name: Purchase, dtype: float64
City_Category
Male Top 10 City_CategoryCity_Category
C    9913.567248
B    9354.854433
A    9017.834470
```

```
1    9413.817605
Name: Purchase, dtype: float64,Female Top 10 Marital_StatusMarital_Status
1    8810.249789
0    8679.845815
Name: Purchase, dtype: float64
```

Observations:

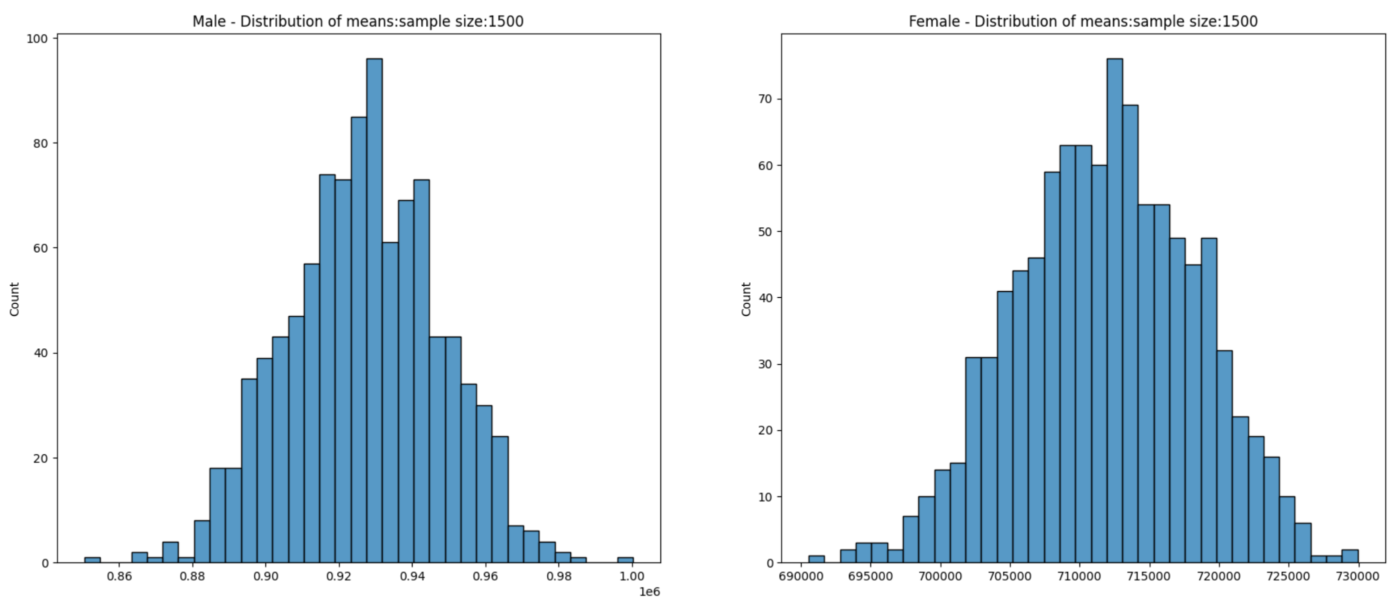
- Age range of 0-50 age female's mean purchase cost is lower then 0-50 age male's
- Leaving occupation type 18,17 & 8 all are below average purchase mean of male whereas all occupation type of male is above average purchase mean of male
- Clearly from Marital_Status,Stay_in_Current_city & city_category we can determine female's purchase mean is lower whereas male's of every type have higher and similar mean than female

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

```
df1=data[data['Gender']=='M']
df2=data[data['Gender']=='F']
sample_size=1500
male_means=[]
female_means=[]
for i in range(1000):
  male_mean=df1.sample(sample_size)['Purchase'].mean()
  male_means.append(male_mean)
  female_mean=df2.sample(sample_size)['Purchase'].mean()
  female_means.append(female_mean)
```

```
fig,axs= plt.subplots(nrows=1,ncols=2,figsize=(20,8))

sns.histplot(data=male_means,bins=35,ax=axs[0])
sns.histplot(data=female_means,bins=35,ax=axs[1])
axs[0].set_title('Male - Distribution of means:sample size:1500')
axs[1].set_title('Female - Distribution of means:sample size:1500')
plt.show()
```




```
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 forFemale: {:.2f}".format(np.mean(female_means)))
print("\nMale - Sample mean: {:.2f} Sample std:{:.2f}".format(df1['Purchase'].mean(), df1['Purchase'].std()))
print("Female - Sample mean: {:.2f} Sample std:{:.2f}".format(df2['Purchase'].mean(),df2['Purchase'].std()))
```

```
Population mean - Mean of sample means of amount spend for Male:926901.98
Population mean - Mean of sample means of amount spend forFemale: 711849.24

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

Observation:

- Average amount spend by male customers is 9,26,341.86
- Average amount spend by female customers is 7,11,704.09

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

```
male_df=df1
female_df=df2


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:

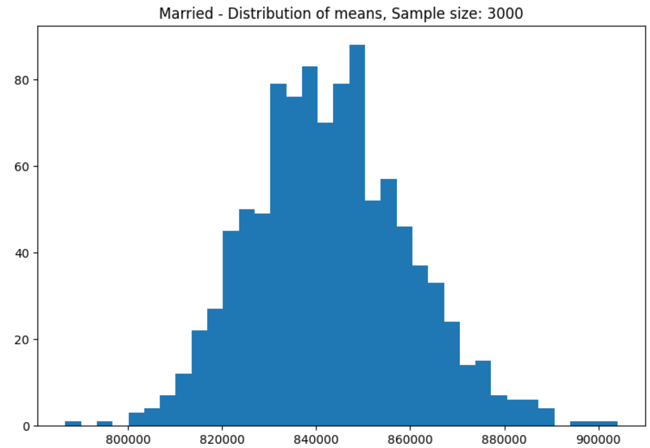
1. Average amount spend by male customer will lie in between: (895617.83, 955070.97)
2. Average amount spend by female customer will lie in between: (673254.77, 750794.02)

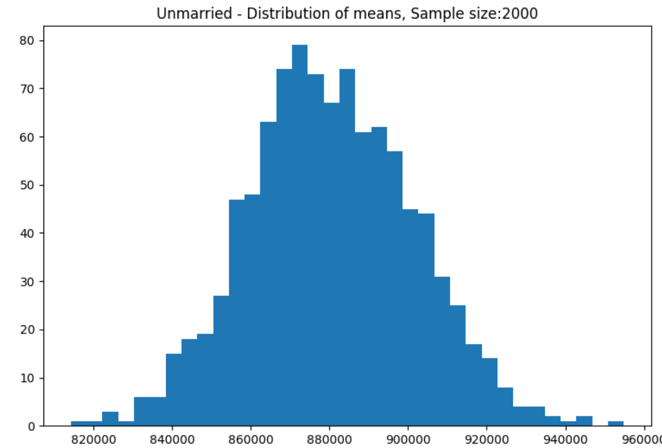
4: Results when the same activity is performed for Married vs Unmarried:

```
amt_df=data
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 forMarried: {:.2f}".format(np.mean(marid_means)))
print("Population mean - Mean of sample means of amount spend forUnmarried: {:.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['Marit
print("Unmarried - Sample mean: {:.2f} Sample std:{:.2f}".format(amt_df[amt_df['Marital_Status']==0]['Purchase'].mean(),amt_df[amt_df['Marit
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 forMarried: 843007.66
Population mean - Mean of sample means of amount spend forUnmarried: 880417.21

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)
```

5: Results when the same activity is performed for 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 0-17 --> confidence interval of means: (527662.46,710073.17)
```

Insights

- ~ 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45)
- 75% of the users are Male and 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 diffrenent types of occupations in the city
- Most of the users are Male
- There are 20 different types of Occupation and Product_Category
- More users belong to B City_Category
- More users are Single as compare to Married
- Product_Category - 1, 5, 8, & 11 have highest purchasing frequency.
- Average amount spend by Male customers: 925344.40

- Average amount spend 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:
3. Average amount spend by male customer will lie in between: (895617.83, 955070.97)
4. Average amount spend by female customer will lie in between: (673254.77, 750794.02) Confidence Interval by Marital_Status
5. Married confidence interval of means: (806668.83, 880384.76)
6. Unmarried confidence interval of means: (848741.18, 912410.38) Confidence Interval by Age
7. For age 26-35 --> confidence interval of means: (945034.42, 1034284.21)
8. For age 36-45 --> confidence interval of means: (823347.80, 935983.62)
9. For age 18-25 --> confidence interval of means: (801632.78, 908093.46)
10. For age 46-50 --> confidence interval of means: (713505.63, 871591.93)
11. For age 51-55 --> confidence interval of means: (692392.43, 834009.42)
12. For age 55+ --> confidence interval of means: (476948.26, 602446.23)
13. 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.