Walmart Case Study (B Dinesh Prabhu DSML Dec 2022)

Importing the Required Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

```
! gdown 1jW0hRDlI20Fp21704T4x2QldtfWG_O8p
```

```
Downloading...
From: https://drive.google.com/uc?id=1jW0hRDlI20Fp21704T4x2QldtfWG_O8p
To: /content/walmartdata.txt
100% 23.0M/23.0M [00:00<00:00, 46.8MB/s]
```

```
Wal_df=pd.read_csv('walmartdata.txt')
Wal_df
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Catego
0	1000001	P00069042	F	0-17	10	A	2	0	
1	1000001	P00248942	F	0-17	10	A	2	0	
2	1000001	P00087842	F	0-17	10	A	2	0	
3	1000001	P00085442	F	0-17	10	A	2	0	
4	1000002	P00285442	M	55+	16	C	4+	0	
...	...	...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	1	1	
550064	1006035	P00375436	F	26-35	1	C	3	0	

1.Defining Problem Statement and Analysing basic metrics

1.Perform Exploratory Data Analysis (EDA) on Walmart data and Extract meaningful insights from it to improve the Business

2. Make an inference from the purchase pattern based on Age,Gender,Age Group,Marital status

> 1.1.1 Columns in the data

```
Wal_df.columns
```

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

> 1.1.2 Information About the data

```
# Checking the Structure of the data
Wal_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
```

> 1.1.3Shape of the data

```
Wal_df.shape
```

```
(550068, 10)
```

> 1.1.4 Data Types the data

```
Wal_df.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
```

> 1.1.5 Conversion of categorical objects into category

```
Wal_df['Product_ID']=Wal_df['Product_ID'].astype('category')
Wal_df['Gender']=Wal_df['Gender'].astype('category')
Wal_df['City_Category']=Wal_df['City_Category'].astype('category')
#Wal_df['Stay_In_Current_City_Years'].value_counts()
Wal_df['Age']=Wal_df['Age'].astype('category')
Wal_df.dtypes
```

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

> 1.2. **Non-Graphical Analysis: Value counts and unique attributes**

> 1.2.1 **Value Counts**

> **Observation** :: Walmart has More number of Male Customers

```
#Gender count
Wal_df['Gender'].value_counts()
```

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

> **Observation**: Most of the customers lies in the Age group 26-35

```
#Age Groups
Wal_df['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
```

> **Observation** City category B has highest number of customers

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

```
B    231173
C    171175
A    147720
Name: City_Category, dtype: int64
```

> **Observation** unmarried customers are dominating

```
Wal_df['Marital_Status'].value_counts()
```

```
0    324731
1    225337
Name: Marital_Status, dtype: int64
```

> v 13.5% of customers stayed less than one year in the current city

```
(Wal_df[ 'Stay_In_Current_City_Years'].value_counts()/len(Wal_df))*100
```

```
1     35.235825
2     18.513711
3     17.322404
4+    15.402823
0     13.525237
Name: Stay_In_Current_City_Years, dtype: float64
```

> **Observation**. less number of purchases are made in Product category **9**

```
Wal_df['Product_Category'].value_counts()
```

```
5     150933
1     140378
8     113925
11     24287
2      23864
6      20466
3      20213
4      11753
16      9828
15      6290
13      5549
10      5125
12      3947
7       3721
18      3125
20      2550
19      1603
14      1523
17       578
9        410
Name: Product_Category, dtype: int64
```

```
#Total 7 Age groups were there
Wal_df['Age'].nunique()
```

```
7
```

```
# Customers from different cities are classified into 3 groups
Wal_df['City_Category'].nunique()
```

```
3
```

```
#Average purchase amount per customer
Wal_df['Purchase'].mean()
```

```
9263.968712959126
```

```
#Mean purchase made by each customer
Wal_df.groupby('Gender')['Purchase'].mean()
```

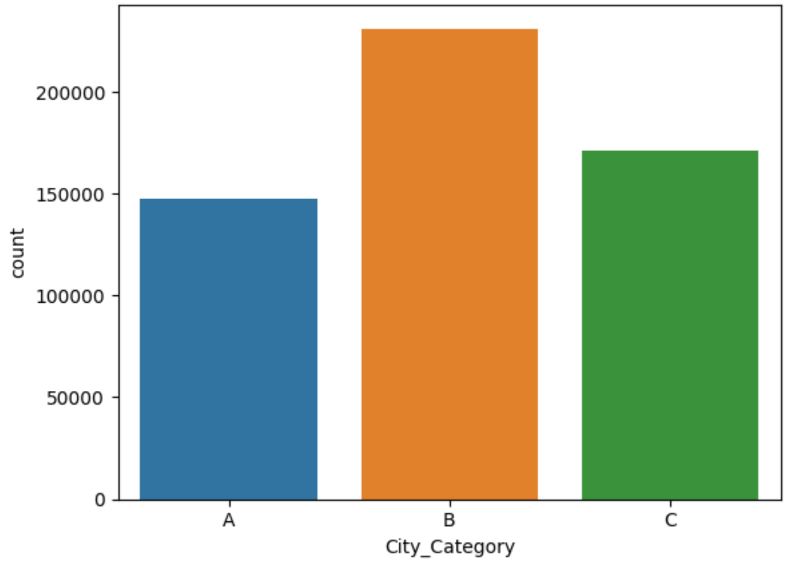
```
Gender
F    8734.565765
M    9437.526040
Name: Purchase, dtype: float64
```

> **1.3 Visual Analysis - Univariate & Bivariate**

> More number of customers are from city B

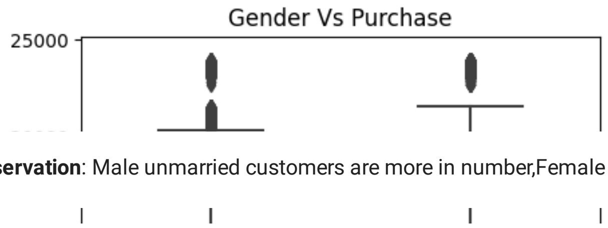
```
sns.countplot(data = Wal_df, x='City_Category')
```

```
<Axes: xlabel='City_Category', ylabel='count'>
```



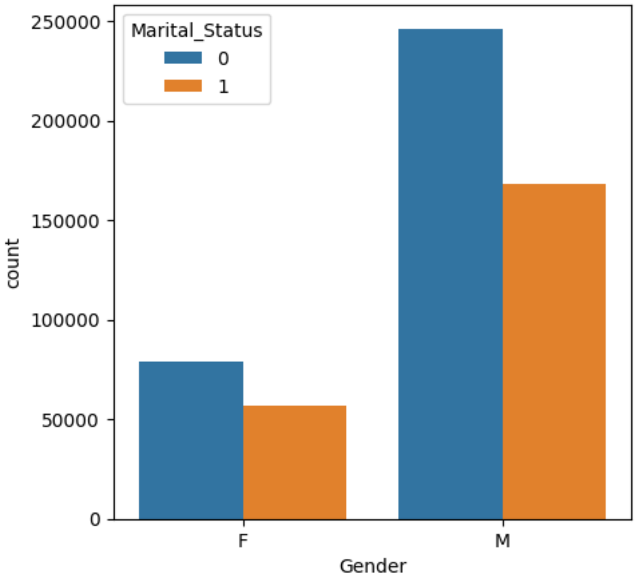
> **Observation** the mean purchase made by Male customers is slightly higher than Female customers we can observe that there are few outliers in Purchases made by Female and male customers

```
plt.figure(figsize=(5,5))
plt.title ('Gender Vs Purchase')
sns.boxplot(data=Wal_df,x='Gender',y='Purchase')
plt.show()
```



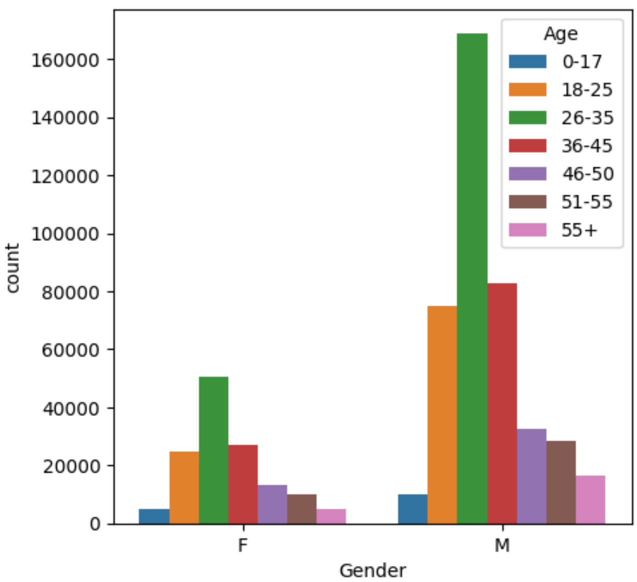
> **Observation**: Male unmarried customers are more in number,Female married customers are less in number

```
plt.figure(figsize=(5,5))
sns.countplot(x='Gender',hue='Marital_Status',data=Wal_df)
plt.show()
```



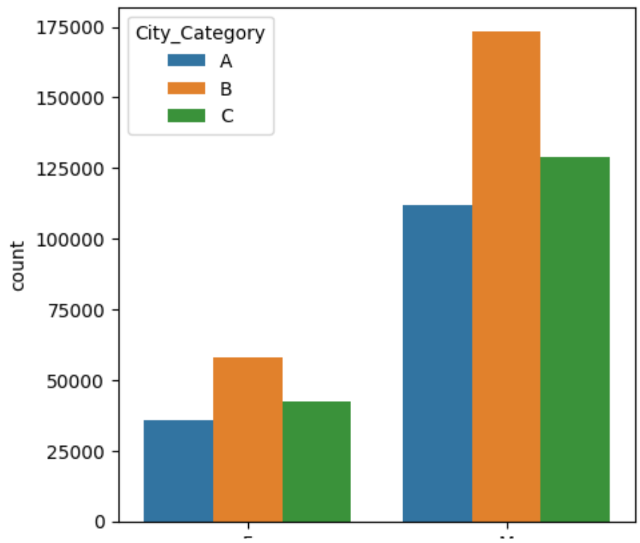
> **Observation**:: 26-35 is the Age group where most of the customers lies

```
plt.figure(figsize=(5,5))
sns.countplot(x='Gender',hue='Age',data=Wal_df)
plt.show()
```



> **Observation**:: City category B has most number of customers and Males are dominating

```
plt.figure(figsize=(5,5))
sns.countplot(x='Gender',hue='City_Category',data=Wal_df)
plt.show()
```

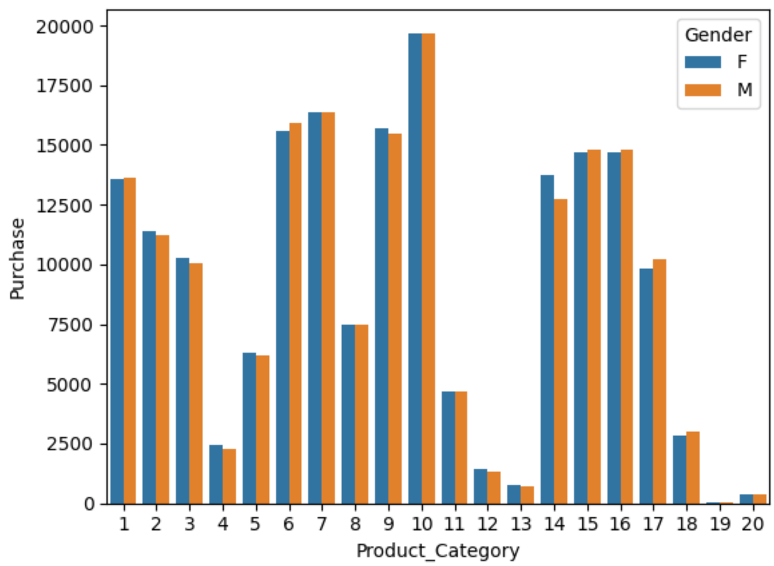


1.3.2 **Bi_Variate Analysis**

> **Observation**: Product Category 19 is where less Amount of purchases were made by both the Gender and in category 10 high number of purchases were made

```
sns.barplot(data=Wal_df,x='Product_Category',y='Purchase',hue='Gender',errorbar=None)
```

```
<Axes: xlabel='Product_Category', ylabel='Purchase'>
```

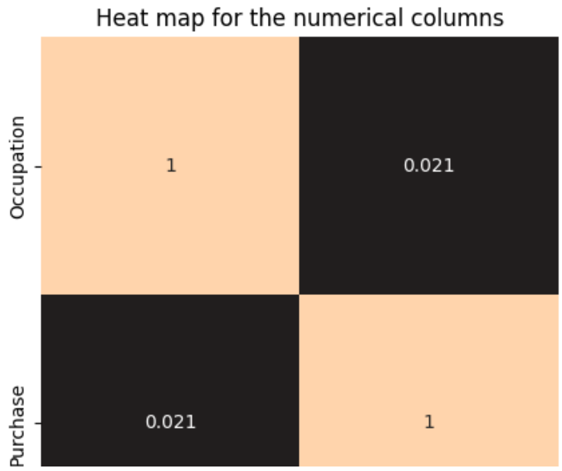



```
#sns.barplot(data=Wal_df,x='Stay_In_Current_City_Years',y='Purchase',hue='Gender')
```

```
df1=Wal_df[['Occupation' ,'Purchase']]
correlation=df1.corr()
correlation
```

	Occupation	Purchase
Occupation	1.000000	0.020833
Purchase	0.020833	1.000000

```
plt.figure(figsize=(5,5))
sns.heatmap(correlation,cbar=False,annot=True,center=0)
plt.title("Heat map for the numerical columns")
plt.show()
```



2. Missing Value & Outlier Detection

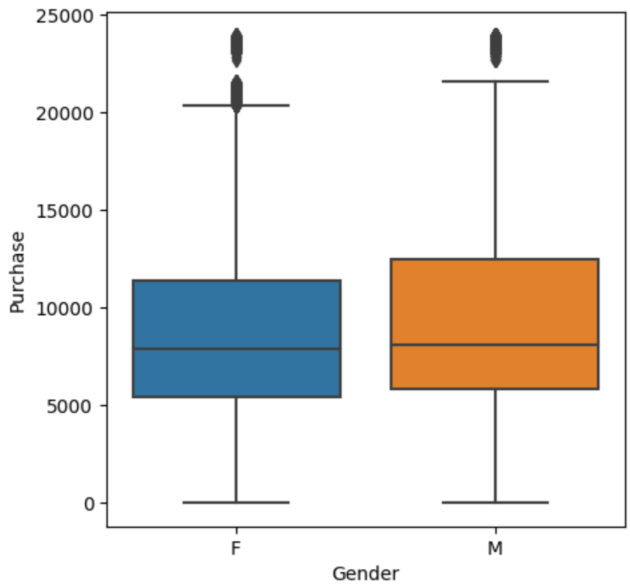
2.1 Checking If there are any Null Values in data

> **Observation** : Looking at the information of the data we can conclude that the data contains ZERO Null values

```
Wal_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
```

```
plt.figure(figsize=(5,5))
sns.boxplot(data=Wal_df,x='Gender',y='Purchase')
plt.show()
```



> **2.2 Statistical Summary Of The Data**

> **Outlier Check** looking at the numeical columns of the data frame we can conclude that data have no outliers

```
Wal_df.describe()[['Occupation','Purchase']]
```

	Occupation	Purchase
count	550068.000000	550068.000000
mean	8.076707	9263.968713
std	6.522660	5023.065394
min	0.000000	12.000000
25%	2.000000	5823.000000
50%	7.000000	8047.000000
75%	14.000000	12054.000000
max	20.000000	23961.000000

> **Observation** looking at the numeical columns of the data frame we can conclude that data have no outliers beacause the mean and median values lies within the 3 sigma standard deviation

> **CONTINGENCY TABLE**

> **Observation** customers belonging to Age group 26-35 is high in number irrespective of the Gender

```
crosstab=pd.crosstab(index=[Wal_df['Gender'],Wal_df['Age']],columns=Wal_df['Marital_Status'],margins='All')
crosstab
```

	Marital_Status	0	1	All
Gender	**Age**			
F	**0-17**	5083	0	5083
	18-25	18357	6271	24628
	26-35	30078	20674	50752
	36-45	16649	10521	27170
	46-50	3166	10033	13199
	51-55	3580	6314	9894
	55+	1908	3175	5083
M	**0-17**	10019	0	10019
	18-25	60187	14845	75032
	26-35	103218	65617	168835
	36-45	49728	33115	82843
	46-50	9524	22978	32502
	51-55	7259	21348	28607
	55+	5975	10446	16421
All		324731	225337	550068

> **Using above contingecy table we can find out the Conditional and Marginal probabilities**

```
#Normalsing values column wise to find the percentage of total contribution column wise
crosstab=pd.crosstab(index=[Wal_df['Gender'],Wal_df['Age']],columns=Wal_df['Marital_Status'],normalize='columns')*100
crosstab
```

Gender	Age	0	1
F	0-17	1.565296	0.000000
	18-25	5.652987	2.782943
	26-35	9.262436	9.174703
	36-45	5.127013	4.669007
	46-50	0.974961	4.452442
	51-55	1.102451	2.802025

```
#normalising row wise
crosstab=pd.crosstab(index=[Wal_df['Gender'],Wal_df['Age']],columns=Wal_df['Marital_Status'],normalize='index')*100
crosstab
```

Gender	Age	Marital_Status 0	1
F	0-17	100.000000	0.000000
	18-25	74.537112	25.462888
	26-35	59.264660	40.735340
	36-45	61.277144	38.722856
	46-50	23.986666	76.013334
	51-55	36.183546	63.816454
	55+	37.536888	62.463112
M	0-17	100.000000	0.000000
	18-25	80.215108	19.784892
	26-35	61.135428	38.864572
	36-45	60.026798	39.973202
	46-50	29.302812	70.697188
	51-55	25.374908	74.625092
	55+	36.386335	63.613665

```
#Normalising all the values
crosstab=pd.crosstab(index=[Wal_df['Gender'],Wal_df['Age']],columns=Wal_df['Marital_Status'],normalize='all')*100
crosstab
```

Gender	Age	Marital_Status 0	1
F	0-17	0.924068	0.000000
	18-25	3.337224	1.140041
	26-35	5.468051	3.758444
	36-45	3.026717	1.912673
	46-50	0.575565	1.823956
	51-55	0.650829	1.147858
	55+	0.346866	0.577201
M	0-17	1.821411	0.000000
	18-25	10.941738	2.698757
	26-35	18.764589	11.928889
	36-45	9.040337	6.020165
	46-50	1.731422	4.177302
	51-55	1.319655	3.880975
	55+	1.086229	1.899038

4.

> **4.1** Are women spending more money per transaction than men?

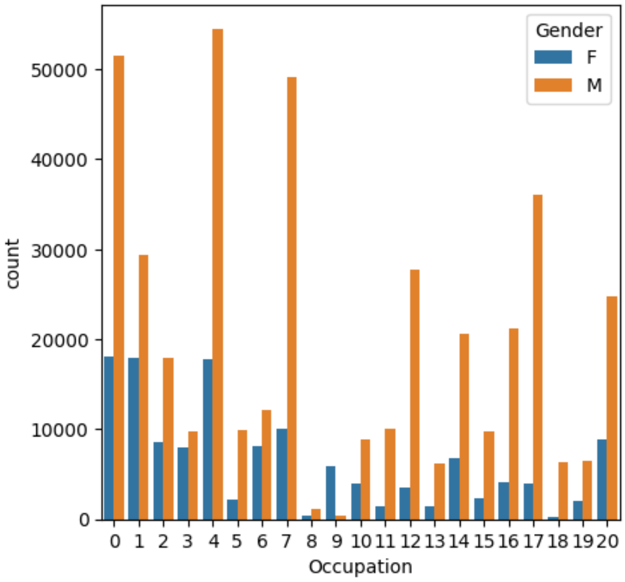
NO looking at the mean purchase amount by each gender we can confirm that Males spends more money per transaction

WHY? looking at the count plot below we can infer that Male customers occupation is very high as compared to women that could be one of the reason of why women spend less money per transaction

```
Wal_df.groupby('Gender')['Purchase'].mean()
```

```
Gender
F    8734.565765
M    9437.526040
Name: Purchase, dtype: float64
```

```
plt.figure(figsize=(5,5))
sns.countplot(data=Wal_df,x='Occupation',hue='Gender')
plt.show()
```

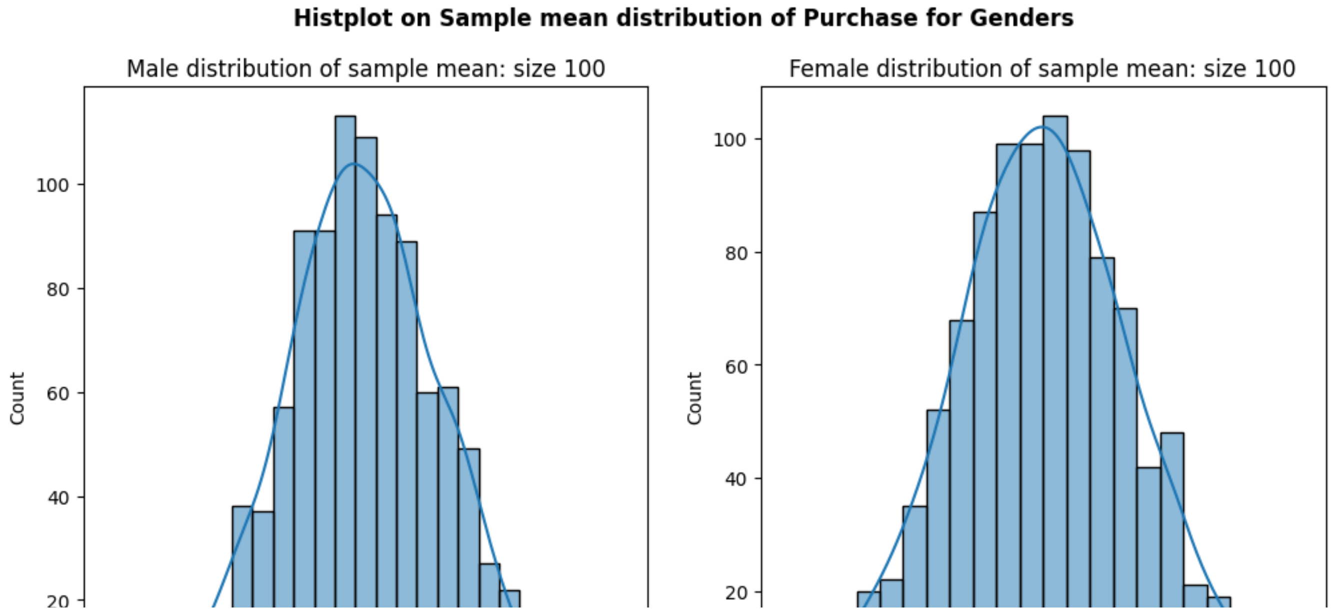


> **4.2** Confidence intervals and distribution of the mean of the expenses by female and male customers

```
male_purchase=Wal_df[Wal_df['Gender']=='M']['Purchase']
Female_purchase=Wal_df[Wal_df['Gender']=='F']['Purchase']
num_resamples=1000
dist_sample_mean_male=[np.mean(male_purchase.sample(100)) for i in range(num_resamples)]
dist_sample_mean_Female=[np.mean(Female_purchase.sample(100)) for i in range(num_resamples)]
```

```
distribution of sample mean for males purchases 9440.667850000002
distribution of sample mean for Female purchases 8742.09086
population mean for male purchases 9437.526040472265
population mean for Female purchases 8734.565765155476
```

```
plt.figure(figsize=(12,6))
plt.suptitle("Histplot on Sample mean distribution of Purchase for Genders", fontweight = 'bold')
plt.subplot(1,2,1)
sns.histplot(data = dist_sample_mean_male, kde= True)
plt.title("Male distribution of sample mean: size 100")
plt.xticks(rotation = 45)

plt.subplot(1,2,2)
sns.histplot(data = dist_sample_mean_Female, kde= True)
plt.title("Female distribution of sample mean: size 100")
plt.xticks(rotation = 45)

plt.show()
```



> I**Observation1**:: We can observe that the sample mean of both male and female purchases is almost closer to the population mean of purchase for both male and female. And also can observe that the sample mean distribution is a Gaussian distribution, hence this concludes that it meets the Central Limit theorem in Purchasing behaviour.

> **Observation2**:: We can also guess by concluding that since the sample mean of 100 for both male and female customers is closer to the population mean of 500k customer data. With this we can infere that the mean for the entire data of 100+ million customers will also lie at almost same or close by to this.

> **90% confidence interval for avg expenses of Male and Female having sample size 100**

```
#Taking the values for z at 90%, 95% and 99% confidence interval as:
z90=1.645 #90% Confidence Interval
z95=1.960 #95% Confidence Interval
z99=2.576 #99% Confidence Interval

sample_SD_male = pd.Series(dist_sample_mean_male).std()
sample_SD_female = pd.Series(dist_sample_mean_Female).std()

sample_SE_male = sample_SD_male/np.sqrt(num_resamples)
sample_SE_female = sample_SD_female/np.sqrt(num_resamples)

pur_upper_limit_male = round(np.mean(dist_sample_mean_male) + z90*sample_SE_male ,2)
pur_lower_limit_male = round(np.mean(dist_sample_mean_male) - z90*sample_SE_male ,2)

pur_upper_limit_female = round(np.mean(dist_sample_mean_Female) + z90*sample_SE_female , 2)
pur_lower_limit_female = round(np.mean(dist_sample_mean_Female) - z90*sample_SE_female , 2)

print('distribution of sample mean for males purchases',np.mean(dist_sample_mean_male))
print('distribution of sample mean for Female purchases',np.mean(dist_sample_mean_Female))
print('\n')
print('population mean for male purchases',male_purchase.mean())
print('population mean for Female purchases',Female_purchase.mean())
print('\n')
print("sample std of males:", round(sample_SD_male,2))
print("sample std of females:", round(sample_SD_female,2))
print("\n")
print("sample std error of males", round(sample_SE_male,2))
print("sample std error of females", round(sample_SE_female,2))
print("\n")
print("CI for male at 90%:",[pur_lower_limit_male, pur_upper_limit_male])
print("CI for female at 90%:" , [ pur_lower_limit_female, pur_upper_limit_female])
```

```
distribution of sample mean for males purchases 9440.667850000002
distribution of sample mean for Female purchases 8742.09086


population mean for male purchases 9437.526040472265
population mean for Female purchases 8734.565765155476


sample std of males: 507.15
sample std of females: 492.23
```

```
sample std error of males 16.04
sample std error of females 15.57


CI for male at 90%: [9414.29, 9467.05]
CI for female at 90%: [8716.49, 8767.7]
```

> **We can conclude that with 90% confidence**:

The average purchases made by Male customers will be from 9414.29 to 9467.05 The average purchases made by Female customers will be from 8716.49 to 8767.7 The sample mean is also lying in between the CI values. Also closer to the population mean.

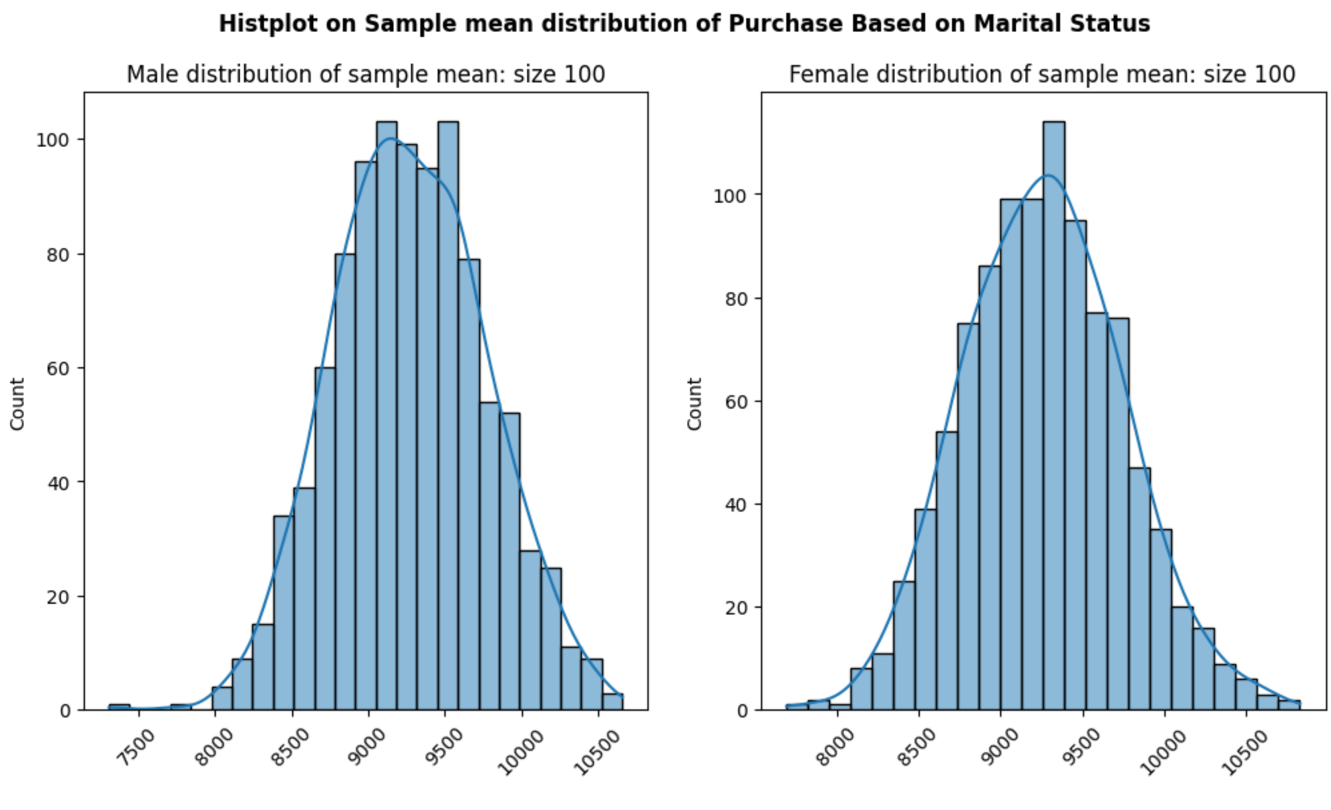
> **4.2** Purchase capability based on the Marital Status

```
Wal_df.groupby('Marital_Status')['Purchase'].mean()
```

```
Marital_Status
0    9265.907619
1    9261.174574
Name: Purchase, dtype: float64
```

```
Married_purchase=Wal_df[Wal_df['Marital_Status']==1]['Purchase']
UnMarried_purchase=Wal_df[Wal_df['Marital_Status']==0]['Purchase']
num_resamples=1000
dist_sample_mean_married=[np.mean(Married_purchase.sample(100)) for i in range(num_resamples)]
dist_sample_mean_unMarried=[np.mean(UnMarries_purchase.sample(100)) for i in range(num_resamples)]
```

```
plt.figure(figsize=(12,6))
plt.suptitle("Histplot on Sample mean distribution of Purchase Based on Marital Status", fontweight = 'bold')
plt.subplot(1,2,1)
sns.histplot(data = dist_sample_mean_married, kde= True)
plt.title("Male distribution of sample mean: size 100")
plt.xticks(rotation = 45)

plt.subplot(1,2,2)
sns.histplot(data = dist_sample_mean_unMarried, kde= True)
plt.title("Female distribution of sample mean: size 100")
plt.xticks(rotation = 45)

plt.show()
```



> **Observation** As we can observe that the population mean of purchasing for Unmarried and Married customers is almost close when performed on sample mean of Unmarried and Married customers. Hence we can conclude that the CLT will work and we can use the CI for them in the next steps.

```
sample_SD_married = pd.Series(dist_sample_mean_married).std()
sample_SD_unmarried = pd.Series(dist_sample_mean_unMarried).std()

sample_SE_married = sample_SD_married/np.sqrt(num_resamples)
sample_SE_unmarried= sample_SD_unmarried/np.sqrt(num_resamples)

pur_upper_limit_married = round(np.mean(dist_sample_mean_married) + z90*sample_SE_male ,2)
pur_lower_limit_married = round(np.mean(dist_sample_mean_married) - z90*sample_SE_male ,2)

pur_upper_limit_unmarried= round(np.mean(dist_sample_mean_unMarried) + z90*sample_SE_female , 2)
pur_lower_limit_unmarried = round(np.mean(dist_sample_mean_unMarried) - z90*sample_SE_female , 2)

print('distribution of sample mean for Married purchases',np.mean(dist_sample_mean_married))
print('distribution of sample mean for UnMarried  purchases',np.mean(dist_sample_mean_unMarried))
print('\n')
print('population mean for Married  purchases',Married_purchase.mean())
print('population mean for UnMarried  purchases',UnMarried_purchase.mean())
print('\n')
print("sample std of Married :", round(sample_SD_married,2))
print("sample std of UnMarried :", round(sample_SD_unmarried,2))
print("\n")
print("sample std error of Married ", round(sample_SE_married,2))
print("sample std error of UnMarried ", round(sample_SE_unmarried,2))
print("\n")
print("CI for Married  at 90%:",[pur_lower_limit_male, pur_upper_limit_married])
print("CI for UnMarried  at 90%:" , [ pur_lower_limit_female, pur_upper_limit_unmarried])
```

```
distribution of sample mean for Married purchases 9272.347810000001
distribution of sample mean for UnMarried  purchases 9261.53417


population mean for Married  purchases 9261.174574082374
population mean for UnMarried  purchases 9265.907618921507


sample std of Married : 504.66
sample std of UnMarried : 490.82


sample std error of Married  15.96
sample std error of UnMarried  15.52


CI for Married  at 90%: [9414.29, 9298.73]
CI for UnMarried  at 90%: [8716.49, 9287.14]
```

> **Observation** We can conclude that with 90% confidence:

- The average purchases made by Unmarried customers will be from 8716.49 to 9287.14
- The average purchases made by Married customers will be from 9414.29 to 9298.73

The sample mean is also lying in between the CI values. Also closer to the population mean.

> **4.3** Purchasability based on the Age Group

```
Wal_df.groupby('Age')['Purchase'].mean()
```

```
Age
0-17     8933.464640
18-25    9169.663606
26-35    9252.690633
36-45    9331.350695
46-50    9208.625697
51-55    9534.808031
55+      9336.280459
Name: Purchase, dtype: float64
```

```
total_pur_age_user =Wal_df.groupby(['User_ID', 'Age'])['Purchase'].sum().reset_index()
total_pur_age_user
```

	User_ID	Age	Purchase
0	1000001	0-17	334093
1	1000001	18-25	0
2	1000001	26-35	0
3	1000001	36-45	0
4	1000001	46-50	0
...	...	...	...
41232	1006040	26-35	1653299
41233	1006040	36-45	0
41234	1006040	46-50	0
41235	1006040	51-55	0
41236	1006040	55+	0

```
#Calculating sample mean purchase for each age group: considering sample size as 200 because the lowest unique age group is 218. Her
sample_size = 90
num_repitions = 1000

all_age_sample_means = {}

age_intervals = ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']
for i in age_intervals:
  all_age_sample_means[i] = []

for i in age_intervals:
  for j in range(num_repitions):
    mean = total_pur_age_user[total_pur_age_user['Age']==i].sample(sample_size, replace=True)['Purchase'].mean()
    all_age_sample_means[i].append(mean)
```

```
#Printing the population mean for all the age groups one after another
print("Population mean for Purchase at Age group 26 to 35: ", round(total_pur_age_user[total_pur_age_user['Age']=='26-35']['Purchase
print("Population mean for Purchase at Age group 36 to 45: ", round(total_pur_age_user[total_pur_age_user['Age']=='36-45']['Purchase
print("Population mean for Purchase at Age group 18 to 25: ", round(total_pur_age_user[total_pur_age_user['Age']=='18-25']['Purchase
print("Population mean for Purchase at Age group 46 to 50: ", round(total_pur_age_user[total_pur_age_user['Age']=='46-50']['Purchase
print("Population mean for Purchase at Age group 51 to 55: ", round(total_pur_age_user[total_pur_age_user['Age']=='51-55']['Purchase
print("Population mean for Purchase at Age group 55+     : ", round(total_pur_age_user[total_pur_age_user['Age']=='55+']['Purchase']
print("Population mean for Purchase at Age group 0 to 17 : ", round(total_pur_age_user[total_pur_age_user['Age']=='0-17']['Purchase
```

```
Population mean for Purchase at Age group 26 to 35:  344894.0
Population mean for Purchase at Age group 36 to 45:  174260.72
Population mean for Purchase at Age group 18 to 25:  155126.24
Population mean for Purchase at Age group 46 to 50:  71438.36
Population mean for Purchase at Age group 51 to 55:  62315.34
Population mean for Purchase at Age group 55+     :  34080.36
Population mean for Purchase at Age group 0 to 17 :  22901.58
```

```
#printing the means sample mean for each age group one after another
print("Sample Mean for purchase at Age group 26 to 35: ", round(np.mean(all_age_sample_means['26-35']),2))
print("Sample Mean for purchase at Age group 36 to 45: ", round(np.mean(all_age_sample_means['36-45']),2))
print("Sample Mean for purchase at Age group 18 to 25: ", round(np.mean(all_age_sample_means['18-25']),2))
print("Sample Mean for purchase at Age group 46 to 50: ", round(np.mean(all_age_sample_means['46-50']),2))
print("Sample Mean for purchase at Age group 51 to 55: ", round(np.mean(all_age_sample_means['51-55']),2))
print("Sample Mean for purchase at Age group 55+     : ", round(np.mean(all_age_sample_means['55+']),2))
print("Sample Mean for purchase at Age group 0 to 17 : ", round(np.mean(all_age_sample_means['0-17']),2))
```

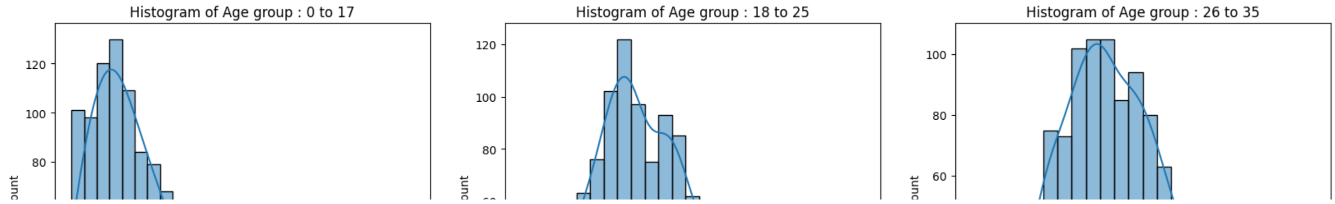
```
Sample Mean for purchase at Age group 26 to 35:  346153.07
Sample Mean for purchase at Age group 36 to 45:  174495.52
Sample Mean for purchase at Age group 18 to 25:  156480.34
Sample Mean for purchase at Age group 46 to 50:  70589.91
Sample Mean for purchase at Age group 51 to 55:  62186.33
Sample Mean for purchase at Age group 55+     :  33974.08
Sample Mean for purchase at Age group 0 to 17 :  23380.38
```

> **Observation** We can observe from the above two pieces of code that the mean of the sample means are closer to the population mean as per central limit theorem. When the number of sample size is increased the means mean get closer to the population mean, as per Central Limit Theorem.

```
#Plotting histogram for the sample mean count of purchasing for each age group
plt.figure(figsize=(20,18))
plt.suptitle("Histplot on Sample mean distribution of Purchase for Age group", fontweight = 'bold')

plt.subplot(3,3,1)
```

```
sns.histplot(data = all_age_sample_means['0-17'], kde = True)
plt.title("Histogram of Age group : 0 to 17")

plt.subplot(3,3,2)
sns.histplot(all_age_sample_means['18-25'], kde = True)
plt.title("Histogram of Age group : 18 to 25")

plt.subplot(3,3,3)
sns.histplot(all_age_sample_means['26-35'], kde = True)
plt.title("Histogram of Age group : 26 to 35")

plt.subplot(3,3,4)
sns.histplot(all_age_sample_means['36-45'], kde = True)
plt.title("Histogram of Age group : 36 to 45")

plt.subplot(3,3,5)
sns.histplot(all_age_sample_means['46-50'], kde = True)
plt.title("Histogram of Age group : 46 to 50")

plt.subplot(3,3,6)
sns.histplot(all_age_sample_means['51-55'], kde = True)
plt.title("Histogram of Age group : 51 to 55")

plt.subplot(3,3,8)
sns.histplot(all_age_sample_means['55+'], kde = True)
plt.title("Histogram of Age group : 55+")
plt.xticks(rotation = 45)

plt.show()
```

Histplot on Sample mean distribution of Purchase for Age group



> **Observation** The means sample seems to be normally distributed for 18-25 ,26-35,36-45 age groups. for other groups the distribution is slightly right skewed

```
#Calculating the CI @ 90%
z90=1.645 #90% Confidence Interval
z95=1.960 #95% Confidence Interval
z99=2.576 #99% Confidence Interval

for val in ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']:
  new_df = total_pur_age_user[total_pur_age_user['Age']==val]
  std_error = z90*new_df['Purchase'].std()/np.sqrt(len(new_df))
  sample_mean = new_df['Purchase'].mean()
  lower_lim = sample_mean - std_error
  upper_lim = sample_mean + std_error
  print("At 90% CI age {} average spent lower limit and upper limit are: ({}, {})".format(val, lower_lim, upper_lim))
```

```
At 90% CI age 26-35 average spent lower limit and upper limit are: (328387.22420281474, 361400.7839452077)
At 90% CI age 36-45 average spent lower limit and upper limit are: (162256.98257750858, 186264.45147443507)
At 90% CI age 18-25 average spent lower limit and upper limit are: (144377.1894380248, 165875.28891879067)
At 90% CI age 46-50 average spent lower limit and upper limit are: (63733.63072954032, 79143.0974999623)
At 90% CI age 51-55 average spent lower limit and upper limit are: (55714.87051785984, 68915.8013544878)
At 90% CI age 55+ average spent lower limit and upper limit are: (29727.374570839012, 38433.33668361694)
At 90% CI age 0-17 average spent lower limit and upper limit are: (19125.44701111194, 26677.704576054926)
```

> **Observation** We can see the sample means are closer to the population mean for the differnt age groups. And, with greater confidence interval we have the upper limit and lower limit range increases. As we have seen for gender and marital status, by increasing the sample size we can have the mean of the sample means closer to the population.

Recommendations

- The purchasing ability towards Men are far more better than females in all the categories. Hence Walmart can concentrate more on marketting towards male customers.
- Most of the customers purchased over the sale period are unmarried compared to married. Hence Walmart can lookout ways how they can market towards married customers as well, so they can expand thier business there.
- Most of the purchases are being made by the occupation codes: 0,4and 7 .Which are disturbuted uniformly accross all the city categories with most of them are unmarried and males.
- Most of the product categories sold are : 6, 7 and 10. So walmart can concentrate more on selling these products as thier demand is very high.
- Top three age groups which has made most of the purchases are: 26-35 (with 35% share), 36-45 (with 20% share) and 18-25 (with 18% share). With total 75% purchase share from age group 18-45.