Importing Liabraries

In [1]:

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

Reading the Dataset

In [6]:

```
df = pd.read_csv("D:/Scaler/Data Visualization/d2beiqkhq929f0.cloudfront.net_public_assets_assets_00
```

In [239]:

```
df
```

Out[239]:

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

550068 rows × 10 columns

Analysing basic metrics

In [5]:

```
df.shape
```

Out[5]:

```
(550068, 10)
```

In the Given sample data Set 550068 records and 10 attributes

In [6]:

```
# Data Type of all columns
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 [219]:

```
#  Statical Summary
df.describe()
```

Out[219]:

	User_ID	Occupation	Marital_Status	Product_Category	Purchase
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.000000
mean	1.003029e+06	8.076707	0.409653	5.404270	9263.968713
std	1.727592e+03	6.522660	0.491770	3.936211	5023.065394
min	1.000001e+06	0.000000	0.000000	1.000000	12.000000
25%	1.001516e+06	2.000000	0.000000	1.000000	5823.000000
50%	1.003077e+06	7.000000	0.000000	5.000000	8047.000000
75%	1.004478e+06	14.000000	1.000000	8.000000	12054.000000
max	1.006040e+06	20.000000	1.000000	20.000000	23961.000000

In [76]:

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

Out[76]:

	Product_ID	Gender	Age	City_Category	Stay_In_Current_City_Years
count	550068	550068	550068	550068	550068
unique	3631	2	7	3	5
top	P00265242	M	26-35	B	1
freq	1880	414259	219587	231173	193821

In [88]:

```
df.isna().sum()
```

Out[88]:

```
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
Purchase_Category           0
dtype: int64
```

No Null Value present in the given data set

In [89]:

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

Out[89]:

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

In [237]:

```
df.groupby(["Gender"])["Purchase"].mean()
```

Out[237]:

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

In the Given Sample Data set 414259 male and 135809 Female Customer's records are present

In [235]:

```
df.groupby(["Age"])["Purchase"].mean()
```

Out[235]:

```
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
```

According to the Given Data set all age group customer's mean Purchasing amount is lies between [8933,9535]

In [71]:

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

Out[71]:

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

According to the given Data set Unmarried customer's are frequently purchasing as compared to married customer's.

In [236]:

```
df.groupby(["Marital_Status"])["Purchase"].mean()
```

Out[236]:

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

In [72]:

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

Out[72]:

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

In [79]:

```
df.groupby(["Gender","Marital_Status"])["Gender"].value_counts()
```

Out[79]:

```
Gender  Marital_Status  Gender
F       0               F          78821
        1               F          56988
M       0               M         245910
        1               M         168349
Name: Gender, dtype: int64
```

Male customer's are frequently purchasing in both(Married or Unmarried) the case

In [80]:

```
df["Purchase"].min()
```

Out[80]:

```
12
```

In [81]:

```
df["Purchase"].max()
```

Out[81]:

```
23961
```

In [240]:

```
def purchase_cat(x):
    if x<=2000:
        return "Low_spend"
    elif x>2000 and x<=8000:
        return "Medium_spend"
    else:
        return "High_spend"
```

In [241]:

```
df["Purchase_Category"]= df["Purchase"].apply(purchase_cat)
```

In [242]:

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

Out[242]:

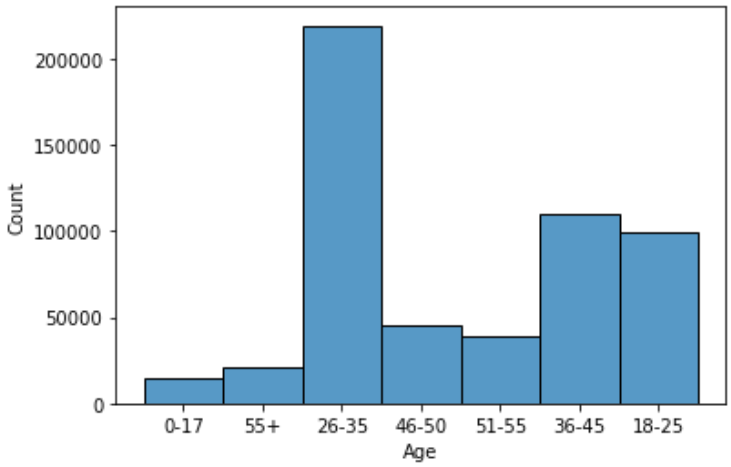
```
High_spend      281775
Medium_spend    240267
Low_spend        28026
Name: Purchase_Category, dtype: int64
```

If we calculate according to purchase category then i found that High spend customer's are greater that other two category

Univariate & Bivariate Analysis

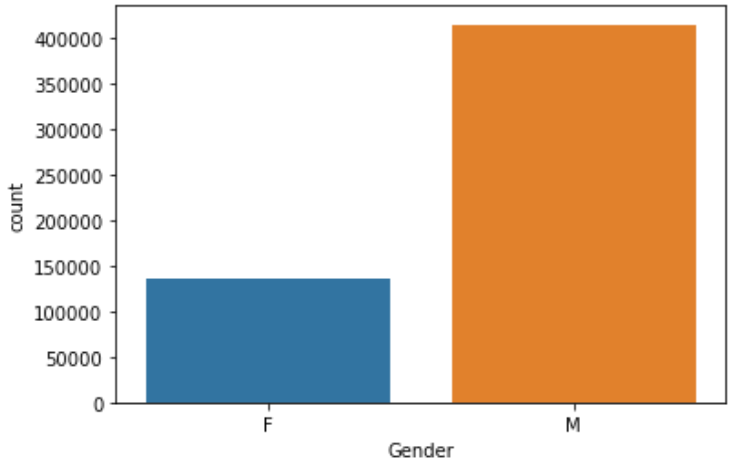
In [92]:

```
sns.histplot(df["Age"])
plt.show()
```



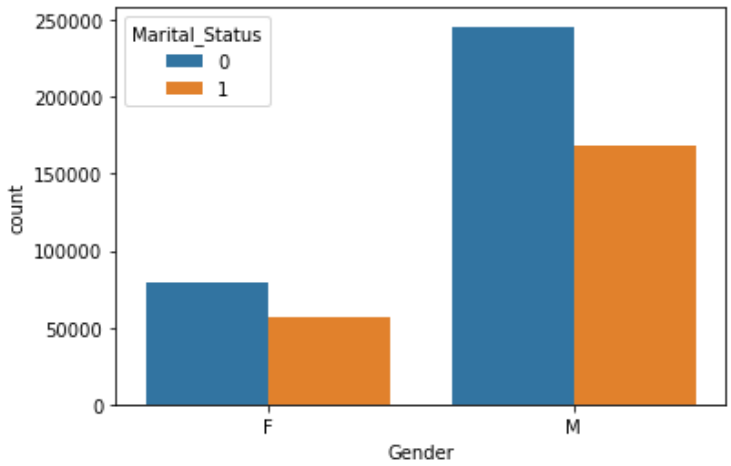
In [95]:

```
sns.countplot(x=df["Gender"])
plt.show()
```



In [96]:

```
sns.countplot(x=df["Gender"],hue=df["Marital_Status"])
plt.show()
```

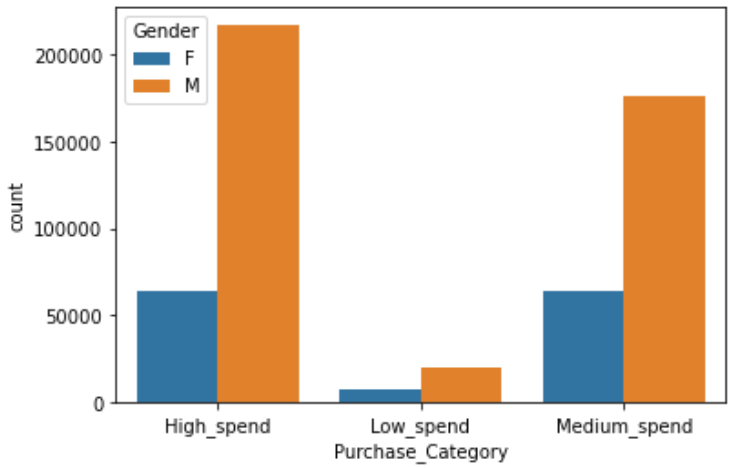


In [97]:

```
sns.countplot(x=df["Purchase_Category"],hue=df["Gender"])
```

Out[97]:

```
<AxesSubplot:xlabel='Purchase_Category', ylabel='count'>
```

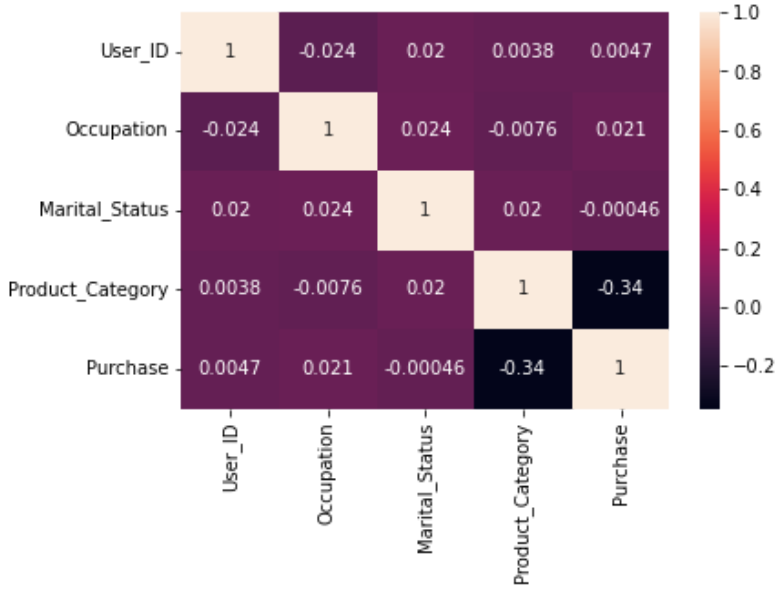


In [103]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[103]:

```
<AxesSubplot:>
```



In [104]:

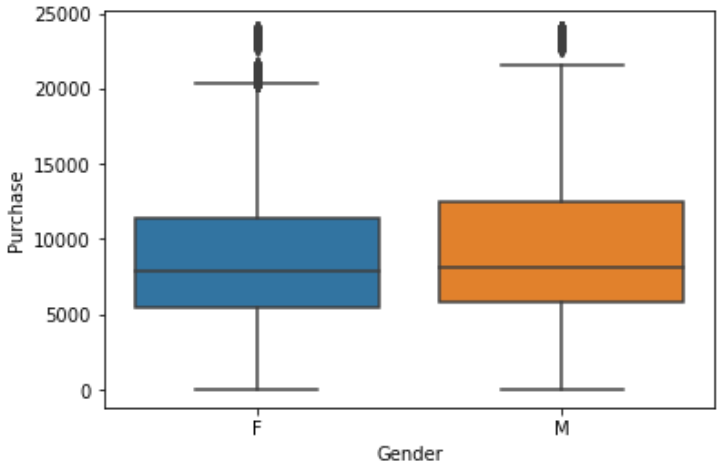
```
sns.boxplot?
```

In [106]:

```
sns.boxplot(data=df,y="Purchase",x="Gender")
```

Out[106]:

```
<AxesSubplot:xlabel='Gender', ylabel='Purchase'>
```



1-After analysing the data i found that Median purchase amount of male and female both are approximately equal.

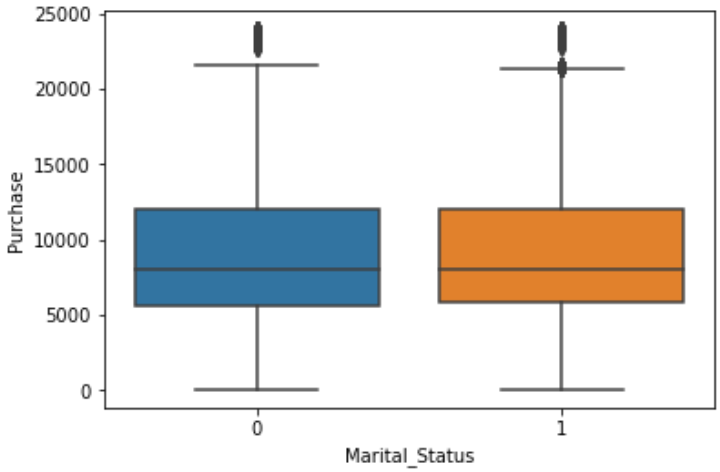
2- In the case of female customer's approximately gretaer than 20k purchase amount customer's are outliers and in the other side approximately gretaer than 22k purchase amount customer's are outliers.

In [107]:

```
sns.boxplot(data=df,y="Purchase",x="Marital_Status")
```

Out[107]:

```
<AxesSubplot:xlabel='Marital_Status', ylabel='Purchase'>
```

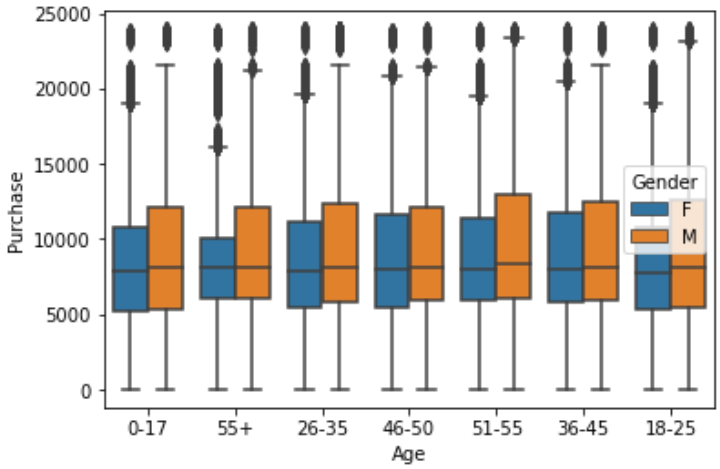


In [110]:

```
sns.boxplot(data=df,y="Purchase",x="Age",hue="Gender")
```

Out[110]:

```
<AxesSubplot:xlabel='Age', ylabel='Purchase'>
```



Calculation of true Mean range of Spend Per Transaction of 50 Million Male and Female ,Using CLT

In [47]:

```
df_male= df.loc[df["Gender"]=="M"]
df_male
```

Out[47]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
4	1000002	P00285442	M	55+	16	C	4+	
5	1000003	P00193542	M	26-35	15	A	3	
6	1000004	P00184942	M	46-50	7	B	2	
7	1000004	P00346142	M	46-50	7	B	2	
8	1000004	P0097242	M	46-50	7	B	2	
...	...	...	...	...	...	...	...	
550057	1006023	P00370853	M	26-35	0	C	2	
550058	1006024	P00372445	M	26-35	12	A	0	
550060	1006026	P00371644	M	36-45	6	C	1	
550062	1006032	P00372445	M	46-50	7	A	3	
550063	1006033	P00372445	M	51-55	13	B	1	

414259 rows × 10 columns

In [48]:

```
df_female= df.loc[df["Gender"]=="F"]
df_female
```

Out[48]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
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	
14	1000006	P00231342	F	51-55	9	A	1	
...	...	...	...	...	...	...	...	
550061	1006029	P00372445	F	26-35	1	C	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	
550066	1006038	P00375436	F	55+	1	C	2	
550067	1006039	P00371644	F	46-50	0	B	4+	

135809 rows × 10 columns

sample mean of male per transaction.

In [53]:

```
mean_male=df_male["Purchase"].mean()
mean_male
```

Out[53]:

```
9437.526040472265
```

Standard Deviation of male sample

In [56]:

```
std_male= df_male["Purchase"].std()
std_male
```

Out[56]:

```
5092.186209777949
```

Number of Sample of Male customer

In [70]:

```
No_male_sample= df_male.shape[0]
No_male_sample
```

Out[70]:

414259

In [71]:

```
# 50 milion male average spend per transaction with 90% confidance interval
norm.interval(0.90,loc=mean_male,scale=(std_male/np.sqrt(No_male_sample)))
```

Out[71]:

(9424.512497305488, 9450.539583639042)

50 Milion male customer's mean spend per transaction will lies between [9424.512497305488, 9450.539583639042] with 90% confidance Interval

In [72]:

```
# 50 milion male average spend per transaction with 95% confidance interval
norm.interval(0.95,loc=mean_male,scale=(std_male/np.sqrt(No_male_sample)))
```

Out[72]:

(9422.01944736257, 9453.032633581959)

50 Milion male customer's mean spend per transaction will lies between [9422.01944736257, 9453.032633581959] with 95% confidance Interval

In [73]:

```
# 50 milion male average spend per transaction with 99% confidance interval
norm.interval(0.99,loc=mean_male,scale=(std_male/np.sqrt(No_male_sample)))
```

Out[73]:

(9417.146922669479, 9457.90515827505)

50 Milion male customer's mean spend per transaction will lies between [9417.146922669479, 9457.90515827505] with 99% confidance Interval

In [ ]:

```

```

Sample Mean of female per Transaction

In [54]:

```
mean_female=df_female["Purchase"].mean()
mean_female
```

Out[54]:

8734.565765155476

Sample Standard Deviation Of Female

In [58]:

```
std_female= df_female["Purchase"].std()
std_female
```

Out[58]:

```
4767.233289291444
```

Number Of sample Of female Customer

In [74]:

```
No_female_sample= df_female.shape[0]
No_female_sample
```

Out[74]:

```
135809
```

In [75]:

```
# 50 milion female average spend per transaction with 90% confidance interval
norm.interval(0.90,loc=mean_female,scale=(std_female/np.sqrt(No_female_sample)))
```

Out[75]:

```
(8713.287834648021, 8755.84369566293)
```

50 Milion female customer's mean spend per transaction will lies between [8713.287834648021, 8755.84369566293] with 90% confidance Interval

In [76]:

```
# 50 milion female average spend per transaction with 95% confidance interval
norm.interval(0.95,loc=mean_female,scale=(std_female/np.sqrt(No_female_sample)))
```

Out[76]:

```
(8709.21154714068, 8759.919983170272)
```

50 Milion female customer's mean spend per transaction will lies between [8709.21154714068, 8759.919983170272] with 95% confidance Interval

In [77]:

```
# 50 milion female average spend per transaction with 99% confidance interval
norm.interval(0.99,loc=mean_female,scale=(std_female/np.sqrt(No_female_sample)))
```

Out[77]:

```
(8701.24467443839, 8767.88685587256)
```

50 Milion female customer's mean spend per transaction will lies between [8701.24467443839, 8767.88685587256] with 99% confidance Interval

Conclusion-

After Calculating the Mean Spend Per Transaction Interval of all male and female customer's with 90%,95%,99% Confidance Interval I found that mean spend per trasaction of male's are greater are than female.

Calculation of true Mean range of Spend Per Transaction for Married and Unmarried Customer's ,Using CLT

In [84]:

```
df_married = df.loc[df["Marital_Status"]==1]
df_married
```

Out[84]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
6	1000004	P00184942	M	46-50	7	B	2	
7	1000004	P00346142	M	46-50	7	B	2	
8	1000004	P0097242	M	46-50	7	B	2	
9	1000005	P00274942	M	26-35	20	A	1	
10	1000005	P00251242	M	26-35	20	A	1	
...	...	...	...	...	...	...	...	
550060	1006026	P00371644	M	36-45	6	C	1	
550061	1006029	P00372445	F	26-35	1	C	1	
550063	1006033	P00372445	M	51-55	13	B	1	
550065	1006036	P00375436	F	26-35	15	B	4+	
550067	1006039	P00371644	F	46-50	0	B	4+	

225337 rows × 10 columns

In [85]:

```
df_unmarried = df.loc[df["Marital_Status"]==0]
df_unmarried
```

Out[85]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
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+	
...	...	...	...	...	...	...	...	
550056	1006022	P00375436	M	26-35	17	C	4+	
550059	1006025	P00370853	F	26-35	1	B	1	
550062	1006032	P00372445	M	46-50	7	A	3	
550064	1006035	P00375436	F	26-35	1	C	3	
550066	1006038	P00375436	F	55+	1	C	2	

324731 rows × 10 columns

Sample Mean Of Married Per Transaction

In [87]:

```
mean_married= df_married["Purchase"].mean()
mean_married
```

Out[87]:

```
9261.174574082374
```

Sample Standard Deviation Of Married

In [88]:

```
std_married= df_married["Purchase"].std()
std_married
```

Out[88]:

```
5016.89737779313
```

Number of Sample's Of married Customer's

In [90]:

```
No_married_sample= df_married.shape[0]
No_married_sample
```

Out[90]:

225337

In [91]:

```
#  Average spend per transaction Of All Married Customer of WallMart with 90% confidance interval
norm.interval(0.90,loc=mean_married,scale=(std_married/np.sqrt(No_married_sample)))
```

Out[91]:

(9243.790713903045, 9278.558434261702)

Average spend per transaction Of All Married Customer of WallMart will lies between [9243.790713903045, 9278.558434261702] with 90% confidance interval

In [92]:

```
#  Average spend per transaction Of All Married Customer of WallMart with 95% confidance interval
norm.interval(0.95,loc=mean_married,scale=(std_married/np.sqrt(No_married_sample)))
```

Out[92]:

(9240.460427057078, 9281.888721107669)

Average spend per transaction Of All Married Customer of WallMart will lies between [9240.460427057078, 9281.888721107669] with 95% confidance interval

In [93]:

```
#  Average spend per transaction Of All Married Customer of WallMart with 99% confidance interval
norm.interval(0.99,loc=mean_married,scale=(std_married/np.sqrt(No_married_sample)))
```

Out[93]:

(9233.951570329937, 9288.39757783481)

Average spend per transaction Of All Married Customer of WallMart will lies between [9233.951570329937, 9288.39757783481] with 99% confidance interval

Sample Mean Of Unmarried Per Transaction

In [95]:

```
mean_unmarried= df_unmarried["Purchase"].mean()
mean_unmarried
```

Out[95]:

9265.907618921507

Sample Standard Deviation Of Unmarried

In [96]:

```
std_unmarried= df_unmarried["Purchase"].std()
std_unmarried
```

Out[96]:

```
5027.347858674457
```

Number of Sample's Of unmarried Customer's

In [97]:

```
No_unmarried_sample= df_unmarried.shape[0]
No_unmarried_sample
```

Out[97]:

```
324731
```

In [98]:

```
#  Average spend per transaction Of All Unmarried Customer of WallMart with 90% confidance interval
norm.interval(0.90,loc=mean_unmarried,scale=(std_unmarried/np.sqrt(No_unmarried_sample)))
```

Out[98]:

```
(9251.39638582367, 9280.418852019344)
```

```
Average spend per transaction Of All Unmarried Customer of WallMart will lies between
[9251.39638582367, 9280.418852019344] with 90% confidance interval
```

In [99]:

```
#  Average spend per transaction Of All Unmarried Customer of WallMart with 95% confidance interval
norm.interval(0.95,loc=mean_unmarried,scale=(std_unmarried/np.sqrt(No_unmarried_sample)))
```

Out[99]:

```
(9248.61641818668, 9283.198819656332)
```

Average spend per transaction Of All Unmarried Customer of WallMart will lies between [9248.61641818668, 9283.198819656332] with 95% confidance interval

In [101]:

```
#  Average spend per transaction Of All Unmarried Customer of WallMart with 99% confidance interval
norm.interval(0.99,loc=mean_unmarried,scale=(std_unmarried/np.sqrt(No_unmarried_sample)))
```

Out[101]:

```
(9243.183129136169, 9288.632108706845)
```

```
Average spend per transaction Of All Unmarried Customer of WallMart will lies between
[9243.183129136169, 9288.632108706845] with 99% confidance interval
```

Conclusion-

After calculating The Mean Spend Per Transaction interval for all Married and Unmarried Customer of wallmart with 90%,95%,99% confidance Interval i found that all three interval Of Married And Unmarried customer's are overlapping

Calculation of true Mean range of Spend Per Transaction for all age group Customer's ,Using CLT

In [102]:

```
df_age_0_17= df.loc[df["Age"]=="0-17"]
df_age_0_17
```

Out[102]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
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	
85	1000019	P00112542	M	0-17	10	A	3	
...	...	...	...	...	...	...	...	
549904	1005803	P00375436	M	0-17	10	C	1	
550012	1005953	P00370853	M	0-17	10	B	0	
550024	1005973	P00370293	M	0-17	10	C	4+	
550035	1005989	P00370853	F	0-17	10	C	3	
550046	1006006	P00375436	F	0-17	0	C	1	

15102 rows × 10 columns

Sample Mean Of Age between 0-17 customer's Per Transaction

In [113]:

```
mean_0_17= df_age_0_17["Purchase"].mean()
mean_0_17
```

Out[113]:

```
8933.464640444974
```

Sample Standard deviation Of Age between 0-17 customer's Per Transaction

In [114]:

```
std_0_17= df_age_0_17["Purchase"].std()
std_0_17
```

Out[114]:

```
5111.11404600277
```

Number of sample of 0-17 age group Customer's

In [115]:

```
No_0_17_sample= df_age_0_17.shape[0]
No_0_17_sample
```

Out[115]:

```
15102
```

In [116]:

```
#  Average spend per transaction Of All 0-17 Age Group Customer's of WallMart with 90% confidance int
norm.interval(0.90,loc=mean_0_17,scale=(std_0_17/np.sqrt(No_0_17_sample)))
```

Out[116]:

```
(8865.053694527898, 9001.87558636205)
```

Average spend per transaction Of All 0-17 Age Group Customer's of WallMart will lies between [8865.053694527898, 9001.87558636205] with 90% confidance interval

In [117]:

```
#  Average spend per transaction Of All  0-17 Age Group Customer's of WallMart with 95% confidance in
norm.interval(0.95,loc=mean_0_17,scale=(std_0_17/np.sqrt(No_0_17_sample)))
```

Out[117]:

```
(8851.947970542686, 9014.981310347262)
```

Average spend per transaction Of All 0-17 Age Group Customer's of WallMart will lies between [8851.947970542686, 9014.981310347262] with 95% confidance interval

In [118]:

```
#  Average spend per transaction Of All  0-17 Age Group Customer's of WallMart with 99% confidance in
norm.interval(0.99,loc=mean_0_17,scale=(std_0_17/np.sqrt(No_0_17_sample)))
```

Out[118]:

```
(8826.333576446717, 9040.59570444323)
```

Average spend per transaction Of All 0-17 Age Group Customer's of WallMart will lies between [8826.333576446717, 9040.59570444323] with 99% confidance interval

In [103]:

```
df_age_18_25= df.loc[df["Age"]=="18-25"]
df_age_18_25
```

Out[103]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
70	1000018	P00366542	F	18-25	3	B	3	
71	1000018	P00190742	F	18-25	3	B	3	
72	1000018	P00151842	F	18-25	3	B	3	
73	1000018	P00112642	F	18-25	3	B	3	
74	1000018	P00118442	F	18-25	3	B	3	
...	...	...	...	...	...	...	...	
550000	1005936	P00370293	M	18-25	4	C	4+	
550015	1005957	P00372445	M	18-25	20	B	1	
550017	1005959	P00371644	F	18-25	4	B	2	
550020	1005964	P00370293	M	18-25	5	B	1	
550032	1005985	P00375436	F	18-25	4	C	0	

99660 rows × 10 columns

Sample Mean Of Age between 18-25 customer's Per Transaction

In [119]:

```
mean_18_25= df_age_18_25["Purchase"].mean()
mean_18_25
```

Out[119]:

```
9169.663606261289
```

Sample Standard deviation Of Age between 18-25 customer's Per Transaction

In [120]:

```
std_18_25= df_age_18_25["Purchase"].std()
std_18_25
```

Out[120]:

```
5034.321997176577
```

Number of sample of 18-25 age group Customer's

In [121]:

```
No_18_25_sample= df_age_18_25.shape[0]
No_18_25_sample
```

Out[121]:

```
99660
```

In [122]:

```
#  Average spend per transaction Of All 18-25 Age Group Customer's of WallMart with 90% confidance i
norm.interval(0.90,loc=mean_18_25,scale=(std_18_25/np.sqrt(No_18_25_sample)))
```

Out[122]:

```
(9143.433031607847, 9195.89418091473)
```

Average spend per transaction Of All 18-25 Age Group Customer's of WallMart will lies between [9143.433031607847, 9195.89418091473] with 90% confidance interval

In [123]:

```
#  Average spend per transaction Of All 18-25 Age Group Customer's of WallMart with 95% confidance i
norm.interval(0.95,loc=mean_18_25,scale=(std_18_25/np.sqrt(No_18_25_sample)))
```

Out[123]:

```
(9138.407948753442, 9200.919263769136)
```

Average spend per transaction Of All 18-25 Age Group Customer's of WallMart will lies between [9138.407948753442, 9200.919263769136] with 95% confidance interval

In [132]:

```
#  Average spend per transaction Of All 18-25 Age Group Customer's of WallMart with 99% confidance i
norm.interval(0.99,loc=mean_18_25,scale=(std_18_25/np.sqrt(No_18_25_sample)))
```

Out[132]:

```
(9128.586709366526, 9210.740503156052)
```

Average spend per transaction Of All 18-25 Age Group Customer's of WallMart will lies between [9128.586709366526, 9210.740503156052] with 99% confidance interval

In [104]:

```
df_age_26_35= df.loc[df["Age"]=="26-35"]
df_age_26_35
```

Out[104]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
5	1000003	P00193542	M	26-35	15	A	3	
9	1000005	P00274942	M	26-35	20	A	1	
10	1000005	P00251242	M	26-35	20	A	1	
11	1000005	P00014542	M	26-35	20	A	1	
12	1000005	P00031342	M	26-35	20	A	1	
...	...	...	...	...	...	...	...	
550058	1006024	P00372445	M	26-35	12	A	0	
550059	1006025	P00370853	F	26-35	1	B	1	
550061	1006029	P00372445	F	26-35	1	C	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	

219587 rows × 10 columns

```
Sample Mean Of Age between 26-35 customer's Per Transaction
```

In [125]:

```
mean_26_35= df_age_26_35["Purchase"].mean()
mean_26_35
```

Out[125]:

```
9252.690632869888
```

```
Sample Standard deviation Of Age between 26-35  customer's Per Transaction
```

In [126]:

```
std_26_35= df_age_26_35["Purchase"].std()
std_26_35
```

Out[126]:

```
5010.527303002927
```

```
Number of sample of 26-35 age group Customer's
```

In [128]:

```
No_26_35_sample= df_age_26_35.shape[0]
No_26_35_sample
```

Out[128]:

```
219587
```

In [129]:

```
#  Average spend per transaction Of All 26-35 Age Group Customer's of WallMart with 90% confidance i
norm.interval(0.90,loc=mean_26_35,scale=(std_26_35/np.sqrt(No_26_35_sample)))
```

Out[129]:

```
(9235.103000581124, 9270.278265158651)
```

Average spend per transaction Of All 26-35 Age Group Customer's of WallMart will lies between [9235.103000581124, 9270.278265158651] with 90% confidance interval

In [133]:

```
#  Average spend per transaction Of All 26-35 Age Group Customer's of WallMart with 95% confidance i
norm.interval(0.95,loc=mean_26_35,scale=(std_26_35/np.sqrt(No_26_35_sample)))
```

Out[133]:

```
(9231.73367640003, 9273.647589339746)
```

Average spend per transaction Of All 26-35 Age Group Customer's of WallMart will lies between [9231.73367640003, 9273.647589339746] with 95% confidance interval

In [134]:

```
#  Average spend per transaction Of All 26-35 Age Group Customer's of WallMart with 99% confidance i
norm.interval(0.99,loc=mean_26_35,scale=(std_26_35/np.sqrt(No_26_35_sample)))
```

Out[134]:

```
(9225.148523415806, 9280.23274232397)
```

Average spend per transaction Of All 26-35 Age Group Customer's of WallMart will lies between [9225.148523415806, 9280.23274232397] with 99% confidance interval

In [106]:

```
df_age_36_45= df.loc[df["Age"]=="36-45"]
df_age_36_45
```

Out[106]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
18	1000007	P00036842	M	36-45	1	B	1	
29	1000010	P00085942	F	36-45	1	B	4+	
30	1000010	P00118742	F	36-45	1	B	4+	
31	1000010	P00297942	F	36-45	1	B	4+	
32	1000010	P00266842	F	36-45	1	B	4+	
...	...	...	...	...	...	...	...	
550049	1006011	P00375436	M	36-45	15	C	3	
550050	1006012	P00371644	M	36-45	15	C	4+	
550053	1006017	P00371644	F	36-45	7	B	1	
550054	1006018	P00370293	M	36-45	1	C	3	
550060	1006026	P00371644	M	36-45	6	C	1	

110013 rows × 10 columns

Sample Mean Of Age between 36-45 customer's Per Transaction

In [135]:

```
mean_36_45= df_age_36_45["Purchase"].mean()
mean_36_45
```

Out[135]:

```
9331.350694917874
```

Sample Standard deviation Of Age between 36-45 customer's Per Transaction

In [136]:

```
std_36_45= df_age_36_45["Purchase"].std()
std_36_45
```

Out[136]:

```
5022.923879204652
```

Number of sample of 36-45 age group Customer's

In [137]:

```
No_36_45_sample= df_age_36_45.shape[0]
No_36_45_sample
```

Out[137]:

```
110013
```

In [138]:

```
#  Average spend per transaction Of All 36-45 Age Group Customer's of WallMart with 90% confidance i
norm.interval(0.90,loc=mean_36_45,scale=(std_36_45/np.sqrt(No_36_45_sample)))
```

Out[138]:

```
(9306.441376202305, 9356.260013633442)
```

Average spend per transaction Of All 36-45 Age Group Customer's of WallMart will lies between [9306.441376202305, 9356.260013633442] with 90% confidance interval

In [139]:

```
#  Average spend per transaction Of All 36-45 Age Group Customer's of WallMart with 95% confidance i
norm.interval(0.95,loc=mean_36_45,scale=(std_36_45/np.sqrt(No_36_45_sample)))
```

Out[139]:

```
(9301.669410965314, 9361.031978870433)
```

Average spend per transaction Of All 36-45 Age Group Customer's of WallMart will lies between [9301.669410965314, 9361.031978870433] with 95% confidance interval

In [140]:

```
#  Average spend per transaction Of All 36-45 Age Group Customer's of WallMart with 99% confidance i
norm.interval(0.99,loc=mean_36_45,scale=(std_36_45/np.sqrt(No_36_45_sample)))
```

Out[140]:

```
(9292.342875603326, 9370.358514232421)
```

Average spend per transaction Of All 36-45 Age Group Customer's of WallMart will lies between [9292.342875603326, 9370.358514232421] with 99% confidance interval

In [107]:

```
df_age_46_50= df.loc[df["Age"]=="46-50"]
df_age_46_50
```

Out[107]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
6	1000004	P00184942	M	46-50	7	B	2	
7	1000004	P00346142	M	46-50	7	B	2	
8	1000004	P0097242	M	46-50	7	B	2	
52	1000013	P00129542	M	46-50	1	C	3	
53	1000013	P00140742	M	46-50	1	C	3	
...	...	...	...	...	...	...	...	
550041	1006000	P00371644	M	46-50	17	B	2	
550043	1006003	P00370293	F	46-50	17	C	1	
550052	1006016	P00375436	M	46-50	1	B	1	
550062	1006032	P00372445	M	46-50	7	A	3	
550067	1006039	P00371644	F	46-50	0	B	4+	

45701 rows × 10 columns

Sample Mean Of Age between 46-50 customer's Per Transaction

In [141]:

```
mean_46_50= df_age_46_50["Purchase"].mean()
mean_46_50
```

Out[141]:

```
9208.625697468327
```

Sample Standard deviation Of Age between 46-50 customer's Per Transaction

In [142]:

```
std_46_50= df_age_46_50["Purchase"].std()
std_46_50
```

Out[142]:

```
4967.216367142921
```

Number of sample of 46-50 age group Customer's

In [143]:

```
No_46_50_sample= df_age_46_50.shape[0]
No_46_50_sample
```

Out[143]:

```
45701
```

In [144]:

```
#  Average spend per transaction Of All 46-50 Age Group Customer's of WallMart with 90% confidance i
norm.interval(0.90,loc=mean_46_50,scale=(std_46_50/np.sqrt(No_46_50_sample)))
```

Out[144]:

```
(9170.406859081897, 9246.844535854758)
```

Average spend per transaction Of All 46-50 Age Group Customer's of WallMart will lies between [9170.406859081897, 9246.844535854758] with 90% confidance interval

In [145]:

```
#  Average spend per transaction Of All 46-50 Age Group Customer's of WallMart with 95% confidance i
norm.interval(0.95,loc=mean_46_50,scale=(std_46_50/np.sqrt(No_46_50_sample)))
```

Out[145]:

```
(9163.085142648752, 9254.166252287903)
```

Average spend per transaction Of All 46-50 Age Group Customer's of WallMart will lies between [9163.085142648752, 9254.166252287903] with 95% confidance interval

In [146]:

```
#  Average spend per transaction Of All 46-50 Age Group Customer's of WallMart with 99% confidance i
norm.interval(0.99,loc=mean_46_50,scale=(std_46_50/np.sqrt(No_46_50_sample)))
```

Out[146]:

```
(9148.775263210646, 9268.476131726009)
```

Average spend per transaction Of All 46-50 Age Group Customer's of WallMart will lies between [9148.775263210646, 9268.476131726009] with 99% confidance interval

In [108]:

```
df_age_51_55= df.loc[df["Age"]=="51-55"]
df_age_51_55
```

Out[108]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
14	1000006	P00231342	F	51-55	9	A	1	
15	1000006	P00190242	F	51-55	9	A	1	
16	1000006	P0096642	F	51-55	9	A	1	
17	1000006	P00058442	F	51-55	9	A	1	
67	1000017	P00019342	M	51-55	1	C	0	
...	...	...	...	...	...	...	...	
549985	1005916	P00370853	M	51-55	20	B	1	
550004	1005940	P00370853	M	51-55	12	C	1	
550037	1005993	P00370293	F	51-55	20	C	1	
550042	1006002	P00371644	M	51-55	0	C	1	
550063	1006033	P00372445	M	51-55	13	B	1	

38501 rows × 10 columns

Sample Mean Of Age between 51-55 customer's Per Transaction

In [147]:

```
mean_51_55= df_age_51_55["Purchase"].mean()
mean_51_55
```

Out[147]:

```
9534.808030960236
```

Sample Standard deviation Of Age between 51-55 customer's Per Transaction

In [ ]:

```
std_51_55= df_age_51_55["Purchase"].std()
std_51_55
```

Number of sample of 51-55 age group Customer's

In [149]:

```
No_51_55_sample= df_age_51_55.shape[0]
No_51_55_sample
```

Out[149]:

```
38501
```

In [150]:

```
#  Average spend per transaction Of All 51-55 Age Group Customer's of WallMart with 90% confidance i
norm.interval(0.90,loc=mean_51_55,scale=(std_51_55/np.sqrt(No_51_55_sample)))
```

Out[150]:

```
(9492.161430973249, 9577.454630947223)
```

Average spend per transaction Of All 51-55 Age Group Customer's of WallMart will lies between [9492.161430973249, 9577.454630947223] with 90% confidance interval

In [151]:

```
#  Average spend per transaction Of All 51-55 Age Group Customer's of WallMart with 95% confidance i
norm.interval(0.95,loc=mean_51_55,scale=(std_51_55/np.sqrt(No_51_55_sample)))
```

Out[151]:

```
(9483.991472776577, 9585.624589143894)
```

Average spend per transaction Of All 51-55 Age Group Customer's of WallMart will lies between [9483.991472776577, 9585.624589143894] with 95% confidance interval

In [152]:

```
#  Average spend per transaction Of All 51-55 Age Group Customer's of WallMart with 99% confidance i
norm.interval(0.99,loc=mean_51_55,scale=(std_51_55/np.sqrt(No_51_55_sample)))
```

Out[152]:

```
(9468.02375292888, 9601.59230899159)
```

Average spend per transaction Of All 51-55 Age Group Customer's of WallMart will lies between [9468.02375292888, 9601.59230899159] with 99% confidance interval

In [111]:

```
df_age_55_= df.loc[df["Age"]=="55+"]
df_age_55_
```

Out[111]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
4	1000002	P00285442	M	55+	16	C	4+	
159	1000031	P00117442	M	55+	7	C	2	
160	1000031	P00322042	M	55+	7	C	2	
161	1000031	P00216342	M	55+	7	C	2	
162	1000031	P00329342	M	55+	7	C	2	
...	...	...	...	...	...	...	...	
549925	1005834	P00371644	M	55+	16	C	4+	
549989	1005922	P00370853	M	55+	3	C	3	
550008	1005946	P00370853	F	55+	1	A	0	
550030	1005980	P00372445	M	55+	1	C	3	
550066	1006038	P00375436	F	55+	1	C	2	

21504 rows × 10 columns

Sample Mean Of Age between 55+ customer's Per Transaction

In [153]:

```
mean_55_= df_age_55_["Purchase"].mean()
mean_55_
```

Out[153]:

```
9336.280459449405
```

Sample Standard deviation Of Age between 55+ customer's Per Transaction

In [154]:

```
std_55_= df_age_55_["Purchase"].std()
std_55_
```

Out[154]:

```
5011.493995603418
```

Number of sample of 55+ age group Customer's

In [155]:

```
No_55__sample= df_age_55_.shape[0]
No_55__sample
```

Out[155]:

```
21504
```

In [156]:

```
#  Average spend per transaction Of All  55+ Age Group Customer's of WallMart with 90% confidance int
norm.interval(0.90,loc=mean_55_,scale=(std_55_/np.sqrt(No_55__sample)))
```

Out[156]:

```
(9280.067707714425, 9392.493211184385)
```

Average spend per transaction Of All 55+ Age Group Customer's of WallMart will lies between [9280.067707714425, 9392.493211184385] with 90% confidance interval

In [157]:

```
#  Average spend per transaction Of All  55+ Age Group Customer's of WallMart with 95% confidance int
norm.interval(0.95,loc=mean_55_,scale=(std_55_/np.sqrt(No_55__sample)))
```

Out[157]:

```
(9269.29883441773, 9403.262084481079)
```

Average spend per transaction Of All 55+ Age Group Customer's of WallMart will lies between [9269.29883441773, 9403.262084481079] with 95% confidance interval

In [158]:

```
#  Average spend per transaction Of All  55+ Age Group Customer's of WallMart with 99% confidance int
norm.interval(0.99,loc=mean_55_,scale=(std_55_/np.sqrt(No_55__sample)))
```

Out[158]:

```
(9248.251682432669, 9424.30923646614)
```

Average spend per transaction Of All 55+ Age Group Customer's of WallMart will lies between [9248.251682432669, 9424.30923646614] with 99% confidance interval

Coclusion-

After calculating true interval mean of all age group customer's of walmart i found following points:

1- 55+ Age group Interval is overlapping with 36- 45 age group with all confidance interval(90%,95%,99%).

2- 55+ Age group Interval is overlapping with 26-35 age group with confidance interval 95% and 99%.

3-46-50 Age group Interval is overlapping with 18-25 age group with all confidance interval(90%,95%,99%)

Calculating True mean and Analysing the Distribution using Bootstrapping

In [162]:

```
df_male
```

Out[162]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
4	1000002	P00285442	M	55+	16	C	4+	
5	1000003	P00193542	M	26-35	15	A	3	
6	1000004	P00184942	M	46-50	7	B	2	
7	1000004	P00346142	M	46-50	7	B	2	
8	1000004	P0097242	M	46-50	7	B	2	
...	...	...	...	...	...	...	...	
550057	1006023	P00370853	M	26-35	0	C	2	
550058	1006024	P00372445	M	26-35	12	A	0	
550060	1006026	P00371644	M	36-45	6	C	1	
550062	1006032	P00372445	M	46-50	7	A	3	
550063	1006033	P00372445	M	51-55	13	B	1	

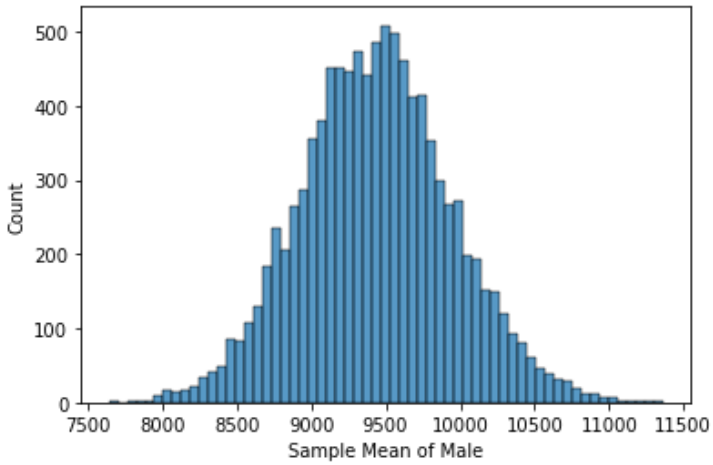
414259 rows × 10 columns

In [172]:

```
sample_Mean= [np.random.choice(df_male["Purchase"],size=100).mean() for i in range(10000)]
```

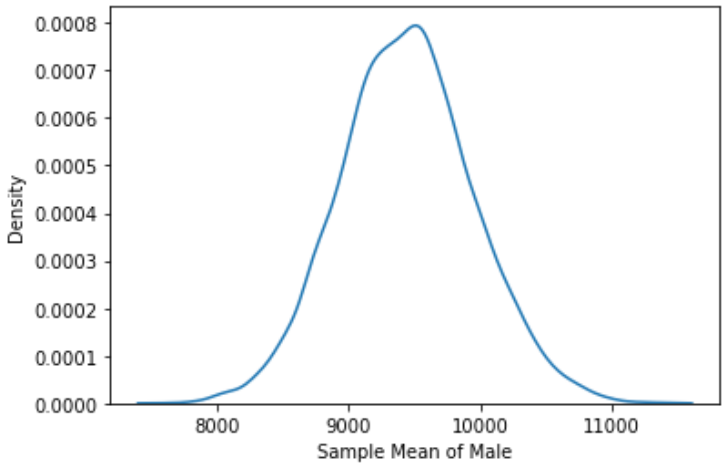
In [207]:

```
sns.histplot(sample_Mean)
plt.xlabel("Sample Mean of Male")
plt.show()
```



In [211]:

```
sns.kdeplot(sample_Mean)
plt.xlabel("Sample Mean of Male")
plt.show()
```



After taking 10000 sample of size 100 from data of male customer's then i found that the distribution of sample mean's are similar to Normal distribution

In [209]:

```
# Mean Of sample Mean
mu= np.mean(sample_Mean)
mu
```

Out[209]:

```
9440.081646999999
```

In [210]:

```
# standard deviation of sample mean distribution
sigma=np.std(sample_Mean,ddof=1)
sigma
```

Out[210]:

```
506.3836866891467
```

In [182]:

```
# Mean Spend per transaction of all male customer's with 90% confidance Interval
n=100
norm.interval(0.90,loc=mu,scale=sigma/(np.sqrt(100)))
```

Out[182]:

```
(9356.788942632029, 9523.37435136797)
```

Mean spend per transaction all Male Customer's will lies between [9356.788942632029, 9523.37435136797] with 90% confidance Interval

In [183]:

```
# Mean Spend per transaction of all male customer's with 95% confidance Interval
n=100
norm.interval(0.95,loc=mu,scale=sigma/(np.sqrt(100)))
```

Out[183]:

```
(9340.832268173064, 9539.331025826934)
```

Mean spend per transaction all Male Customer's will lies between [9340.832268173064, 9539.331025826934] with 95% confidance Interval

In [184]:

```
# Mean Spend per transaction of all male customer's with 99% confidance Interval
n=100
norm.interval(0.99,loc=mu,scale=sigma/(np.sqrt(100)))
```

Out[184]:

```
(9309.645853098697, 9570.517440901302)
```

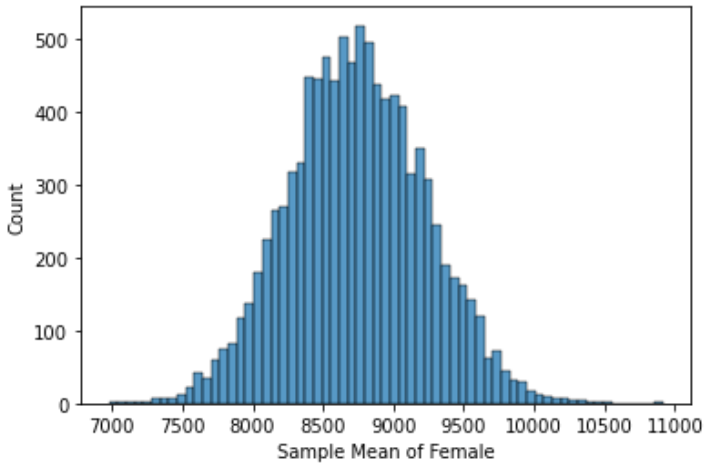
Mean spend per transaction all Male Customer's will lies between [9309.645853098697, 9570.517440901302] with 99% confidance Interval

In [188]:

```
Sample_Mean_female= [np.mean(np.random.choice(df_female["Purchase"],size=100)) for i in range(10000)]
```

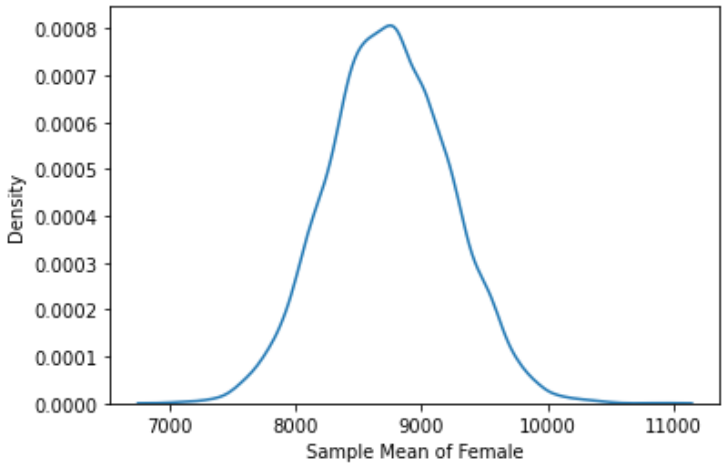
In [214]:

```
sns.histplot(Sample_Mean_female)
plt.xlabel("Sample Mean of Female")
plt.show()
```



In [215]:

```
sns.kdeplot(Sample_Mean_female)
plt.xlabel("Sample Mean of Female")
plt.show()
```




```
After taking  10000 sample of size 100 from  data of female customer's  then i found that the
distribution of sample mean's are similar to Normal distribution
```

In [216]:

```
#  Mean of female sample Mean
mu_1= np.mean(Sample_Mean_female)
mu_1
```

Out[216]:

```
8741.205073000001
```

In [217]:

```
#  Satandard Deviation of female sample mean Distribution
sigma_1=np.std(Sample_Mean_female,ddof=1)
sigma_1
```

Out[217]:

```
484.37586970016315
```

In [199]:

```
# Mean Spend per transaction of all female customer's with 90% confidance Interval
n=100
norm.interval(0.90,loc=mu_1,scale=sigma_1/(np.sqrt(100)))
```

Out[199]:

```
(8661.532332391593, 8820.87781360841)
```

Mean spend per transaction all Female Customer's will lies between [8661.532332391593, 8820.87781360841] with 90% confidance Interval

In [202]:

```
# Mean Spend per transaction of all male customer's with 95% confidance Interval
n=100
norm.interval(0.95,loc=mu_1,scale=sigma_1/(np.sqrt(100)))
```

Out[202]:

```
(8646.269147040743, 8836.14099895926)
```

Mean spend per transaction all Female Customer's will lies between [8646.269147040743, 8836.14099895926] with 95% confidance Interval

In [203]:

```
# Mean Spend per transaction of all male customer's with 99% confidance Interval
n=100
norm.interval(0.99,loc=mu_1,scale=sigma_1/(np.sqrt(100)))
```

Out[203]:

```
(8616.438117089434, 8865.972028910568)
```

Mean spend per transaction all Female Customer's will lies between [8616.438117089434, 8865.972028910568] with 99% confidance Interval

Conclusion-

After Analysing The data i found that the population Mean of Male is greater than Female with 90%, 95% and 99% confidance Interval

Business Insight

Insight based on EDA

1-After Analysing the given sample data set i observed that Male customer's are frquently purchasing as compared to female customer's.

2-After Analysing the given sample data set i observed that Mean spend per transaction of male greater than female.

3-After Analysing the given sample data set i observed that Mean spend per transaction of Unmarried slighly grater than male.

4-After Analysing the given sample data set i observed that Mean spend per transaction of 51-55 age group customer's are maximum and 0-17 age group customer's are minimum

Insight based on CLT

1- Mean Spend per transaction of 50 milion male customer's are greater than female cuatomer's with 90%, 95% and 99% confidance Interval.

2- Mean Spend per transaction of all married and unmarried customer's are overlapping with 90%, 95% and 99% confidance Interval.

3- 55+ Age group true mean Interval is overlapping with 36- 45 age group with all confidance interval(90%,95%,99%).

4- 55+ Age group true mean Interval is overlapping with 26-35 age group with confidance interval 95% and 99%.

5 46-50 Age group true mean Interval is overlapping with 18-25 age group with all confidence interval(90% 95% 99%)

Recomondatations-

Based on EDA

1-Wallmart has to be more focus on Married Customer's Because they are not frequently Purchasing.

2-Wallmart has to be more focus on 0-17 and 45+ age group customer's.

3-wallmart should stock some attractive product for teenagers so that 0-17 age group customer's will buy more.

Based on CLT

1- Wallmart has to be more focus on female Customer's beacause after analysing the data i observed that true mean of female custome's are less than male.

2-Wallmart has to be more focus on Married Customer's beacause after analysing the data i observed that true mean of Married custome's are less than Unmarried.

3-Wallmart has to be more focus on 0-17 Age Customer's beacause after analysing the data i observed that true mean of 0-17 age group custome's are less than other age group customer's.

In [ ]: