```
import pandas as pd
import numpy as np

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

	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
...	...	...	...	...	...	...	...	...	...	...
550063	1006033	P00372445	M	51-55	13	B	1	1	20	368
550064	1006035	P00375436	F	26-35	1	C	3	0	20	371
550065	1006036	P00375436	F	26-35	15	B	4+	1	20	137
550066	1006038	P00375436	F	55+	1	C	2	0	20	365

1-| Defining Problem Statement and Analyzing basic metrics

1a-| Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

```
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	46-50	7	B	2	1	1	19215
7	1000004	P00346142	M	46-50	7	B	2	1	1	15854
8	1000004	P0097242	M	46-50	7	B	2	1	1	15686
9	1000005	P00274942	M	26-35	20	A	1	1	8	7871

```
df.shape
```

```
(550068, 10)
```

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

Gender	count
M	414259
F	135809

```
df.describe()
```

	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

- Average of purchase value is 9263

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

	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

- Product_ID P00265242 is widely popular product because the sales of this product is high(1880)
- Customers between Age group of 26-35 are the highest purchasers.
- Male customers are the highest purchasers

1b-| Non-Graphical Analysis: Value counts and unique attributes

Q) Which gender contributes more to the revenue?

```
np.round((df['Gender'].value_counts(normalize=True))*100,2)
```

Gender	proportion
M	75.31
F	24.69

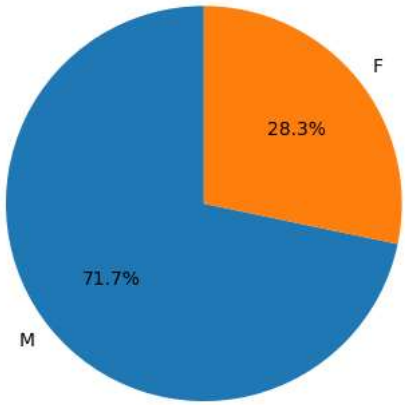
- Male Customers(75%) contribute more towards the revenue compared to Female customers(25%)

```
import math as m
import matplotlib.pyplot as plt
import seaborn as sns


Uniq_gen_id=df[['Gender','User_ID']].drop_duplicates()
Uniq_gen_id['Gender'].value_counts()
```

Gender	count
M	4225
F	1666

```
plt.pie(Uniq_gen_id['Gender'].value_counts(),labels=Uniq_gen_id['Gender'].value_counts().index,autopct='%1.1f%%',startangle=90)
plt.show()
```



72% of Customers are Unique in Male Category and 28% of are unique in Female Category

```
np.round((df.groupby(['Gender'])['Purchase'].sum())/(df['Purchase'].sum())*100,2)
```

Gender	Purchase
F	23.28
M	76.72

```
np.round((df.groupby(['Marital_Status'])['User_ID'].nunique()/df['User_ID'].nunique())*100,2)
```

Marital_Status	User_ID
0	58.0
1	42.0

58% of Customers are Unmarried 42% of Customers are Married

```
np.round((df.groupby(['Age'])['User_ID'].nunique())/(df['User_ID'].nunique())*100,2)
```

Age	User_ID
0-17	3.70
18-25	18.15
26-35	34.85
36-45	19.81
46-50	9.01
51-55	8.16
55+	6.31

35% of unique customers are between age group of 26-35 and as a age increases the purchasing proportion drops

```
np.round((df.groupby(df['Age'])['Purchase'].sum())/(df['Purchase'].sum())*100,2)
```

Age	Purchase
0-17	2.65
18-25	17.93
26-35	39.87
36-45	20.15
46-50	8.26
51-55	7.20
55+	3.94

```
np.round((df.groupby(['Occupation'])['User_ID'].nunique())/(df['User_ID'].nunique())*100,2)
```

Occupation	User_ID
0	11.68
1	8.78
2	4.35
3	2.89
4	12.56
5	1.88
6	3.87
7	11.36
8	0.29
9	1.49
10	3.26
11	2.17
12	6.38
13	2.38
14	4.99
15	2.38
16	3.99
17	8.33
18	1.14
19	1.21
20	4.63

11-12% of Customers are from 0,4,7 Occupation

```
np.round(df['City_Category'].value_counts(normalize=True)*100,2)
```

City_Category	proportion
B	42.03
C	31.12
A	26.85

Majority of orders from B Category followed by C and A

Q) Which city category has the lowest spending?

```
np.round((df.groupby(['City_Category'])['User_ID'].nunique())/(df['User_ID'].nunique())*100,2)
```

City_Category	User_ID
A	17.74
B	28.98
C	53.28

C Category(53) of City_Category as More unique customers compared to A(18) and B(29) Comparing above tables we found that despite of having more unique customers Ctaegory C has less orders with compared to Category B and Category A

```
np.round((df.groupby(['City_Category'])['Purchase'].sum())/(df['Purchase'].sum())*100,2)
```

City_Category	Purchase
A	25.83
B	41.52
C	32.65

```
np.round((df.groupby(['City_Category'])['Purchase']).mean(),2)
```

City_Category	Purchase
A	8911.94
B	9151.30
C	9719.92

Category B has less number of users but they spend heavily. depite having less users compared to C yet they spend more compared to C Category. Category A also have less users but they also spend more compared to Category C. **This proves that despite of having more users category C Spends less compared to other categories.**

Q) "Which product category shows high purchase frequency in one city category while already performing well in others?"

```
df.groupby(['City_Category'])['Product_Category'].value_counts()
```

City_Category	Product_Category	count
A	5	42211
	1	35081
	8	32179
	11	6601
	2	6141
	6	5507
	3	4943
	4	3050
	16	2848
	15	1717
	13	1614
	10	1333
	7	1226
	12	1063
	18	753
	14	481
	20	468
	19	273
	17	121
	9	110
B	5	64138
	1	58253
	8	47553
	11	10485
	2	10444
	3	8587
	6	8526
	4	5226
	16	4038
	15	2638
	13	2271
	10	2063
	12	1675
	7	1599
	18	1389
	20	753
	14	632
	19	462
	17	267
	9	174
C	1	47044
	5	44584
	8	34193
	2	7279
	11	7201
	3	6683
	6	6433
	4	3477
	16	2942
	15	1935
	10	1729

13	1664
20	1329
12	1209
18	983
7	896
19	868
14	410
17	190
9	126

City Category B's total purchasing amount higher than C, even with fewer customers: Number of orders for Product_category of 11,2 and 10 are much higher comapare to other city category and amongs them 10 is the costliest category ,whereas product_Category of 5,1,8 are most bought from every city category.

Q) Given the higher customer base in City Category C, what strategies can drive increased sales there?

```
df[(df['City_Category']=='C')|(df['Gender']== 'F')]['Product_Category'].value_counts()
```

Product_Category	count
5	74140
1	63614
8	57664
2	11187
3	10516
11	10506
6	9592
4	5973
16	4597
15	2673
13	2672
10	2483
12	2216
20	1671
7	1599
18	1239
19	1086
14	849
17	225
9	173

```
df[(df['City_Category']=='C')|(df['Gender']== 'M')]['Product_Category'].value_counts()
```

Product_Category	count
1	123808
5	121377
8	90454
11	20982
2	19956
6	17307
3	16380
4	9257
16	8173
15	5552
13	4541
10	4371
7	3018
12	2940
18	2869
20	2208
19	1385
14	1084
17	543
9	363

We can advertise and promote products like 10 as it is underused but has strong demand. Categories like 2,6,10 should be highly promoted across Gender for better reach

```
df.dtypes
```

	0
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

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

	0
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

There are no null values

```
bin=[-1,2500,5000,6500,8000,9500,11000,13500,15000,20000,25000]
labl=['<2.5K','2.5K-5K','5K-6.5K','6.5K-8K','8K-9.5K','9.5K-11K','11K-13.5K','13.5K-15K','15K-20K','>20K']
df['Purchase_Range']=pd.cut(df['Purchase'],bin,labels=labl)
df
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase	Purcha
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	
...	...	...	...	...	...	...	...	...	...	...	...
550063	1006033	P00372445	M	51-55	13	B	1	1	20	368	
550064	1006035	P00375436	F	26-35	1	C	3	0	20	371	

1C-| Visual Analysis

```
import seaborn as sns
import matplotlib.pyplot as plt


df_copy = df[['Gender','Age','Occupation','Marital_Status','Product_Category','Purchase','Stay_In_Current_City_Years']].copy()
df_copy['Gender'].replace(['M','F'],[1,0],inplace = True)
df_copy['Age'].replace(['0-17','18-25','26-35','36-45','46-50','51-55','55+'],[0,1,2,3,4,5,6],inplace = True)
df_copy['Stay_In_Current_City_Years'].replace(['1','2','3','4+','0'],[0,1,2,3,4],inplace = True)
```

```
/tmp/ipython-input-28-205219528.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment usi
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value


  df_copy['Gender'].replace(['M','F'],[1,0],inplace = True)
/tmp/ipython-input-28-205219528.py:2: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To r
  df_copy['Gender'].replace(['M','F'],[1,0],inplace = True)
/tmp/ipython-input-28-205219528.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment usi
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value


  df_copy['Age'].replace(['0-17','18-25','26-35','36-45','46-50','51-55','55+'],[0,1,2,3,4,5,6],inplace = True)
/tmp/ipython-input-28-205219528.py:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To r
  df_copy['Age'].replace(['0-17','18-25','26-35','36-45','46-50','51-55','55+'],[0,1,2,3,4,5,6],inplace = True)
/tmp/ipython-input-28-205219528.py:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment usi
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value


  df_copy['Stay_In_Current_City_Years'].replace(['1','2','3','4+','0'],[0,1,2,3,4],inplace = True)
/tmp/ipython-input-28-205219528.py:4: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To r
```

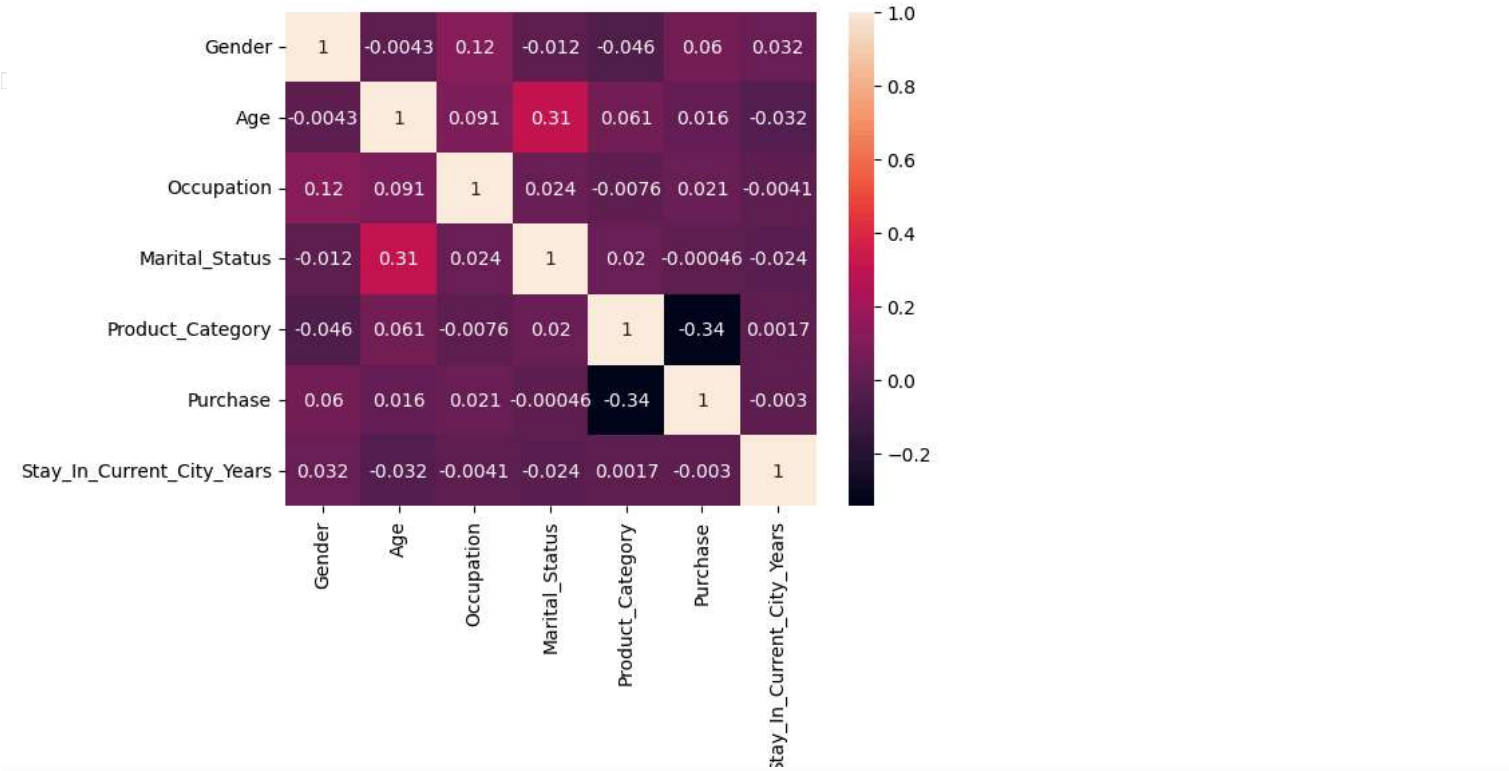
```
df_copy['Stay_In_Current_City_Years'].replace(['1','2','3','4+','0'],[0,1,2,3,4],inplace = True)
```

```
df_copy.corr()
```

	Gender	Age	Occupation	Marital_Status	Product_Category	Purchase	Stay_In_Current_City_Years
Gender	1.000000	-0.004262	0.117291	-0.011603	-0.045594	0.060346	0.032242
Age	-0.004262	1.000000	0.091463	0.311738	0.061197	0.015839	-0.031528
Occupation	0.117291	0.091463	1.000000	0.024280	-0.007618	0.020833	-0.004073
Marital_Status	-0.011603	0.311738	0.024280	1.000000	0.019888	-0.000463	-0.024155
Product_Category	-0.045594	0.061197	-0.007618	0.019888	1.000000	-0.343703	0.001661
Purchase	0.060346	0.015839	0.020833	-0.000463	-0.343703	1.000000	-0.002989
Stay_In_Current_City_Years	0.032242	-0.031528	-0.004073	-0.024155	0.001661	-0.002989	1.000000

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

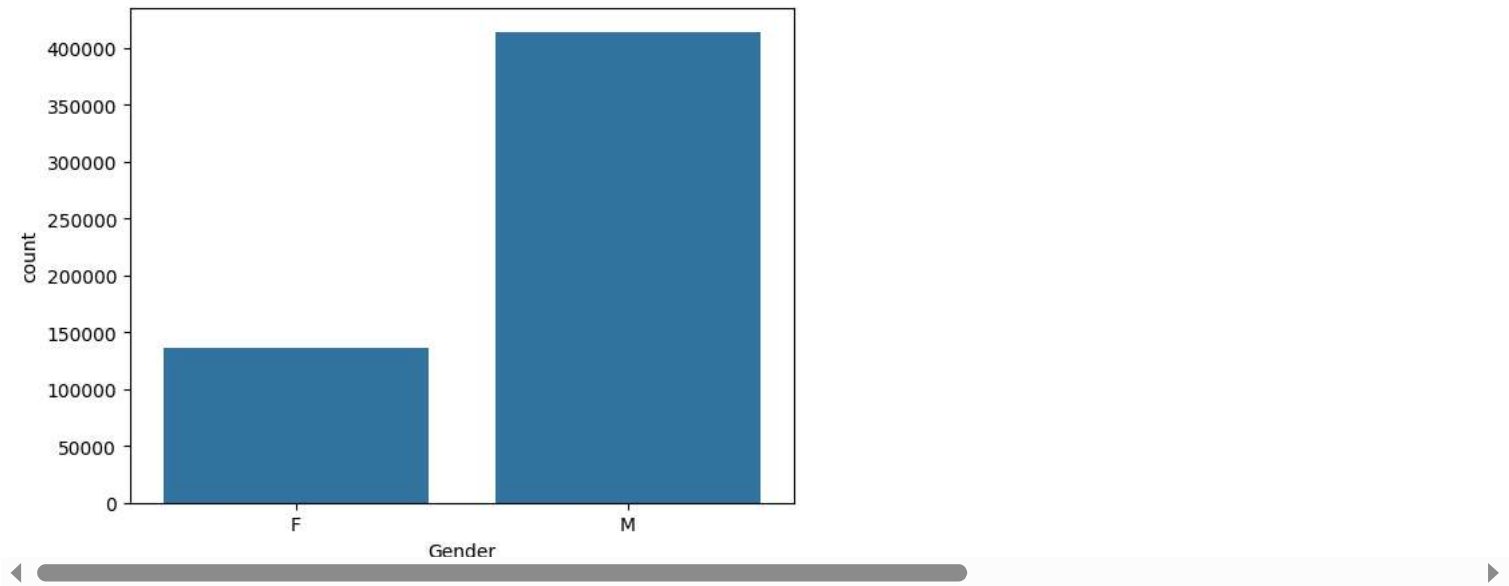
```
<Axes: >
```



We can conclude from the Heatmap that data attributes are not correlated to each other

```
sns.countplot(x='Gender',data=df)
```

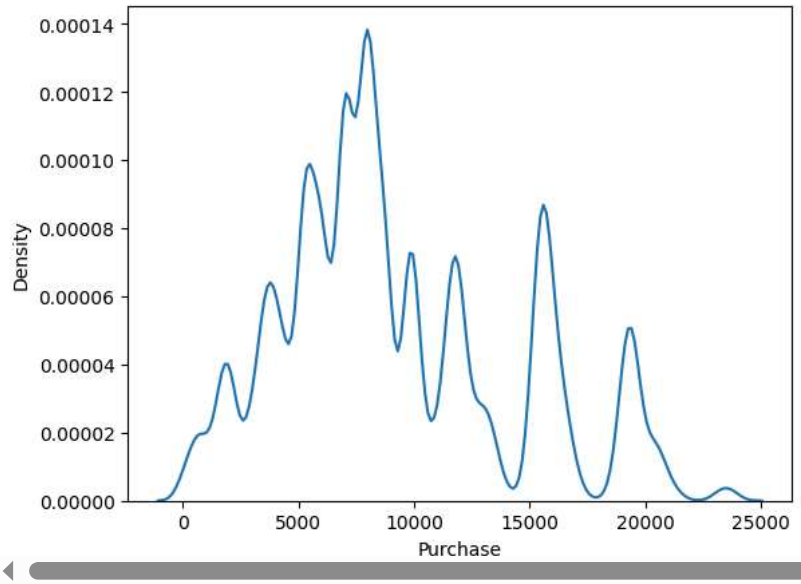
```
<Axes: xlabel='Gender', ylabel='count'>
```



This plot gives us the distribution of Gender wise with respect to no of orders No of orders from Male is higher

```
sns.kdeplot(data= df,x='Purchase')
```

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



This kdeplot gives us density curve of orders Purchase amount 6K-10K which is more density region

Q) What is the total amount spent by male and female customers?

```
df.groupby(['Gender'])['Purchase'].sum().reset_index()
```

	Gender	Purchase
0	F	1186232642
1	M	3909580100

Q)What is the average spending of male and female customers?

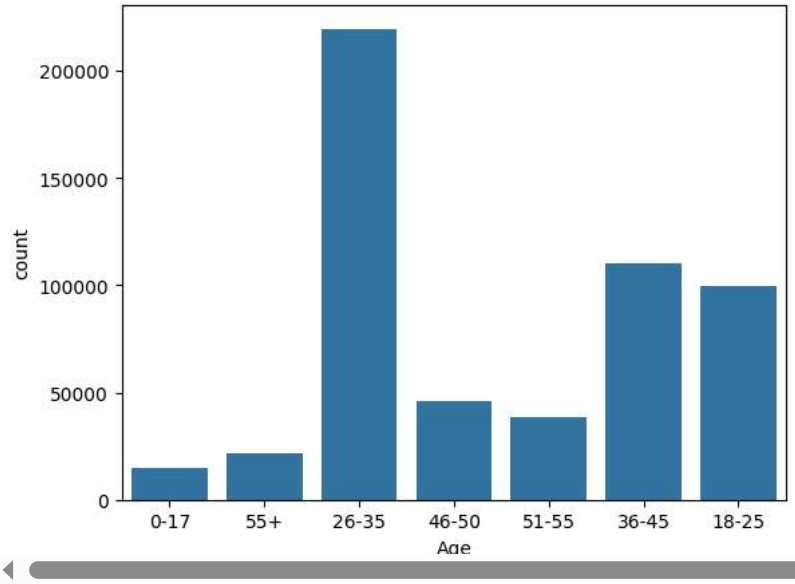
```
df.groupby(['Gender'])['Purchase'].mean().reset_index()
```

	Gender	Purchase
0	F	8734.565765
1	M	9437.526040

Q)Which age group ordered the most?

```
sns.countplot(x='Age',data=df)
```

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



Age group 26-35 Order the most.

There is a drastical drop in the order between the age group of 46-50 and 51-55.

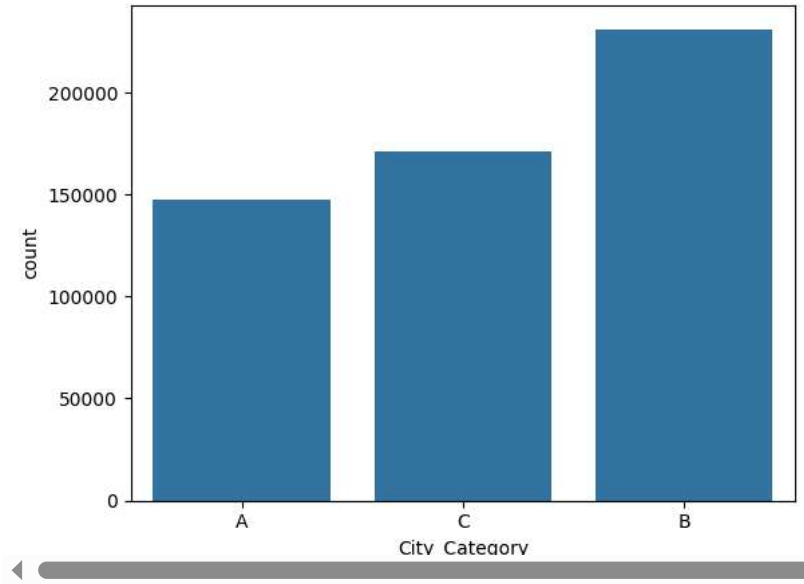
Age group between 36-45 and 18-25 also order less in compared to 26-35 age group.

Age 0-17 also have least orders due to less purchasing power and 55+ age group also have least orders

Q)Which City category ordererd the most?

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

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

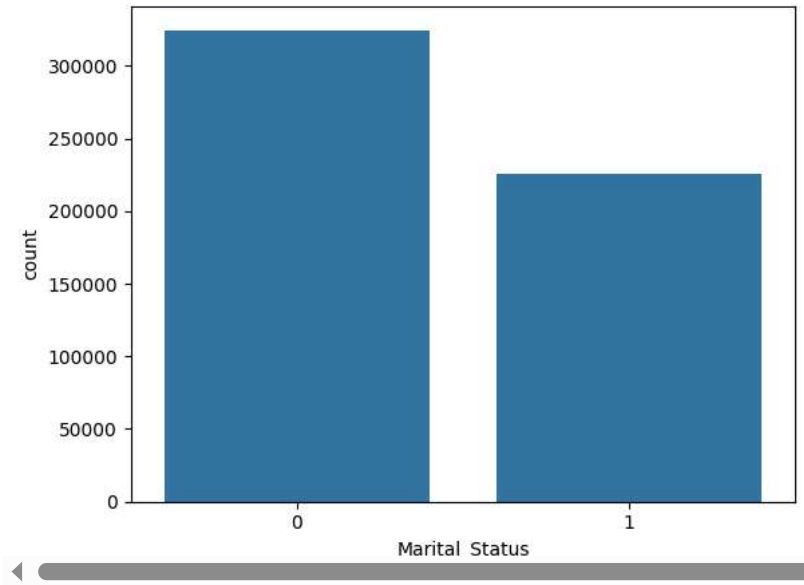


City Category B Orders most number of times

Q) Which marital status category ordered the most?

```
sns.countplot(data=df, x='Marital_Status')
```

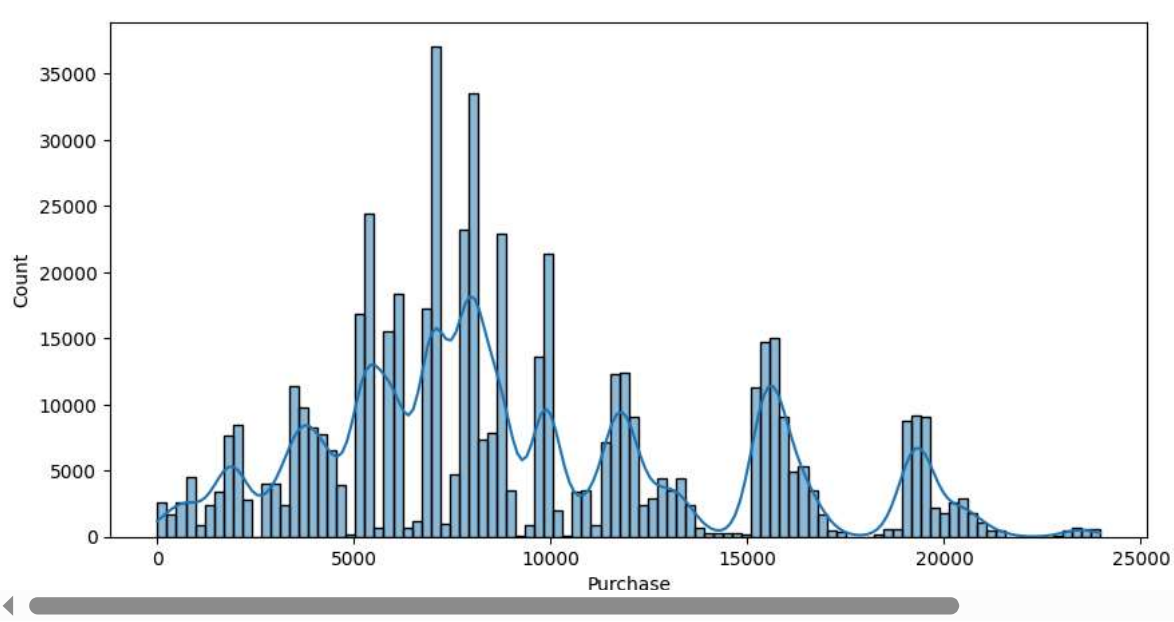
```
<Axes: xlabel='Marital_Status', ylabel='count'>
```



Unmarried people order more

Q)What is the distribution of purchase amounts among customers?

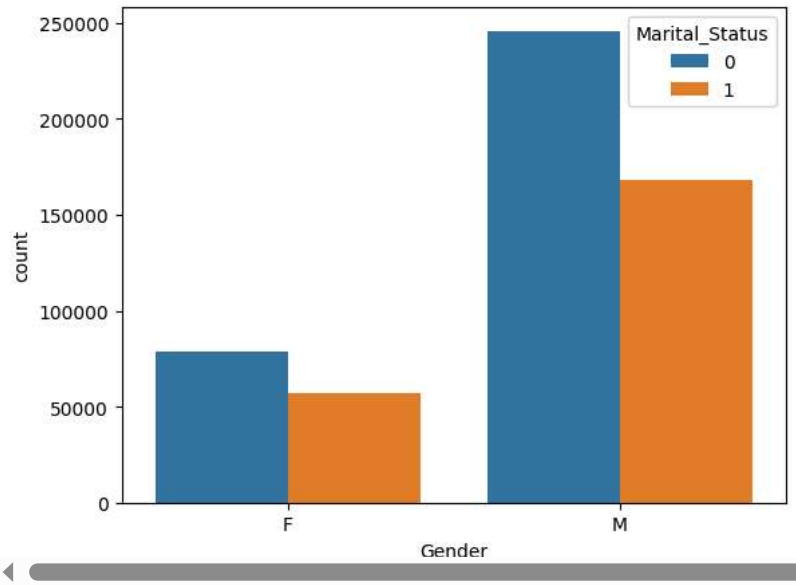
```
plt.figure(figsize=(10,5))
sns.histplot(data=df,x='Purchase',bins=100,kde=True,linewidth=1)
plt.show()
```



This plot shows that the purchasing pattern is not uniformly distributed. it has more peaks

```
sns.countplot(data=df,x='Gender',hue='Marital_Status')
```

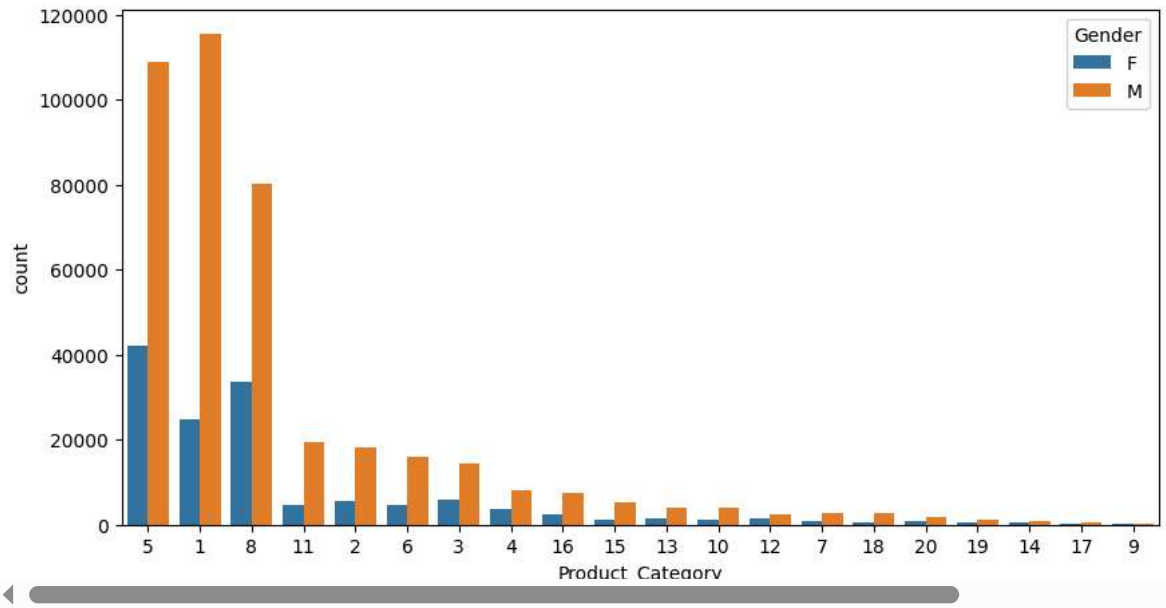
```
<Axes: xlabel='Gender', ylabel='count'>
```



Unmarried People order more compared to married people in both genders

```
order= df['Product_Category'].value_counts().index
plt.figure(figsize=(10,5))
sns.countplot(data=df,hue='Gender',x='Product_Category',order=order)
```

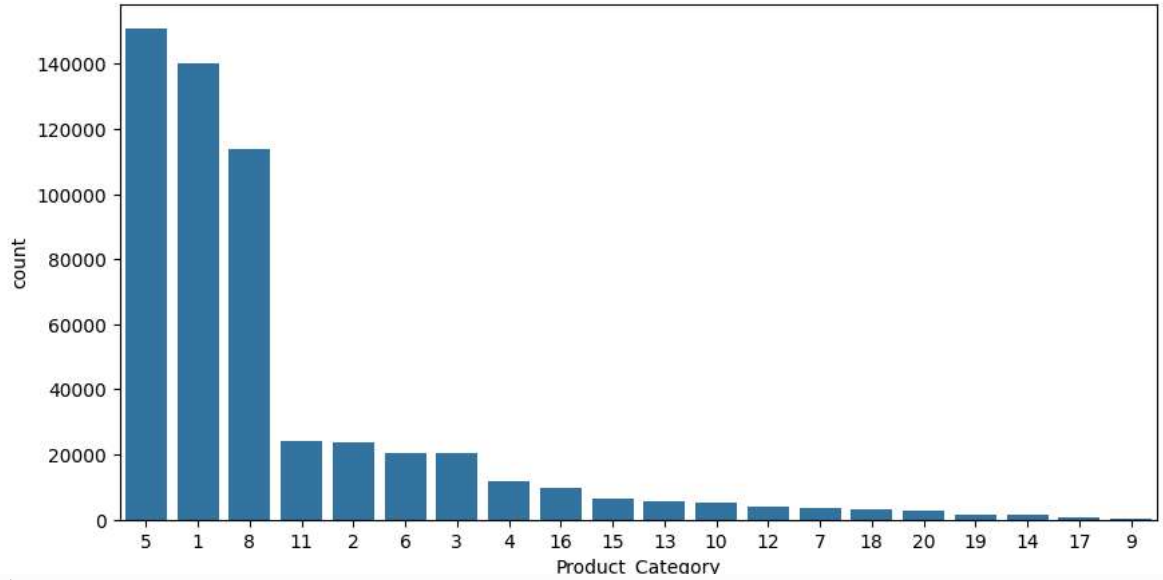
```
<Axes: xlabel='Product_Category', ylabel='count'>
```



Product category 5,1 and 8 are the top3 most purchased among both the genders

```
order= df['Product_Category'].value_counts().index
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Product_Category',order=order)
```

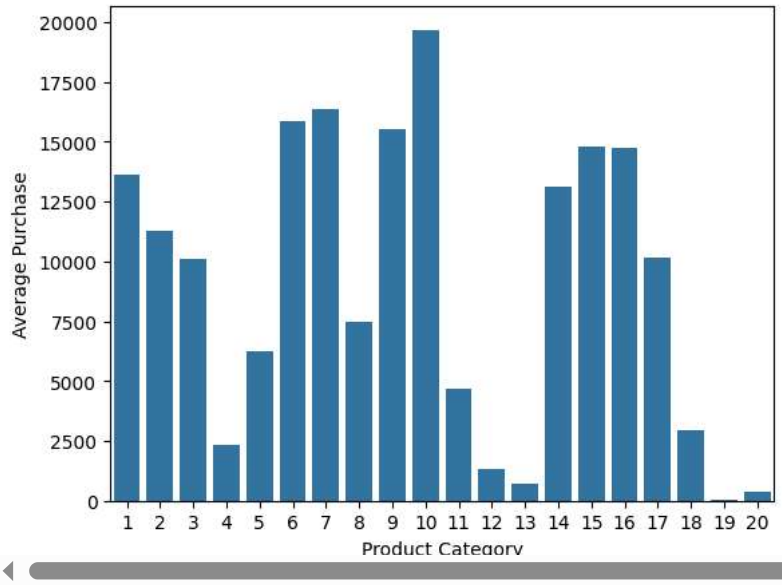
```
<Axes: xlabel='Product_Category', ylabel='count'>
```



Q)Which product category has the highest average purchase amount?

```
sns.barplot(x=df.groupby(['Product_Category'])['Purchase'].mean().sort_values(ascending=False).index,y=df.groupby(['Product_Category'])['Purchase'].mean
plt.ylabel('Average Purchase')
plt.xlabel('Product Category')
```

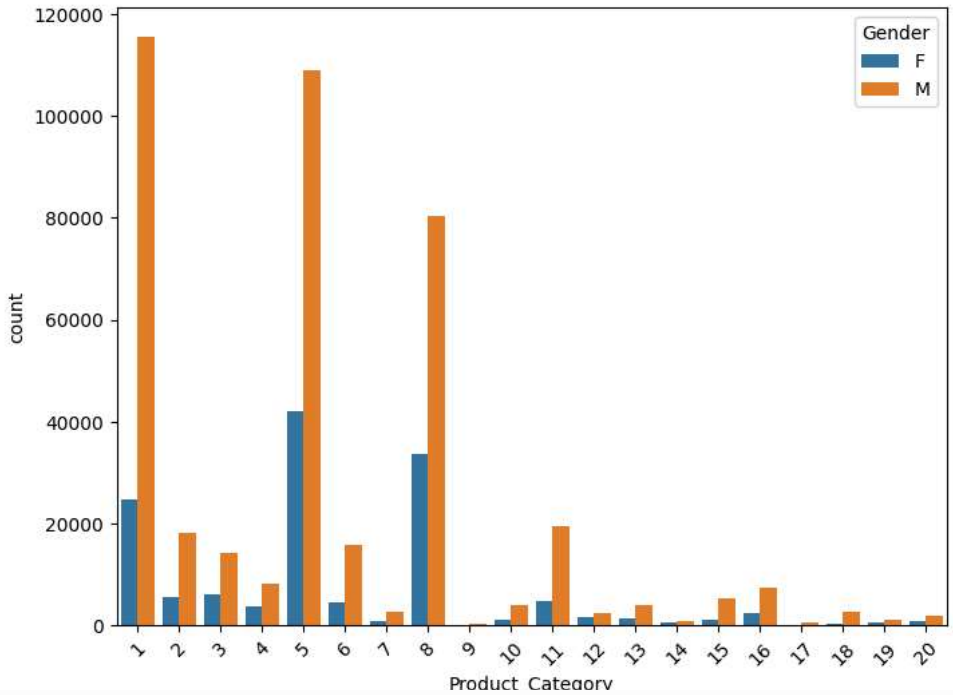
```
Text(0.5, 0, 'Product Category')
```



Product Category 10 has highest average purchase amount

Q) How does product category category distribution vary by gender?

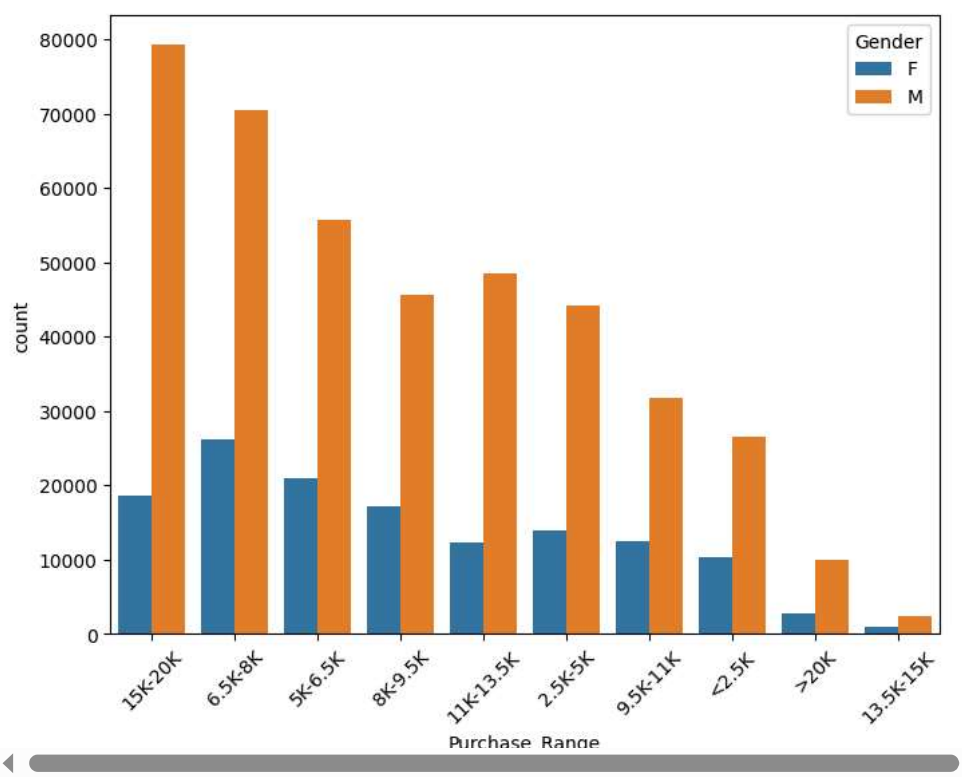
```
plt.figure(figsize = (8,6))
sns.countplot(data =df,x = 'Product_Category',hue = 'Gender')
plt.xticks(rotation = 45)
plt.show()
```



Product Category 1,5,8 has highest order from Male and also in all the category Male order most

Double-click (or enter) to edit

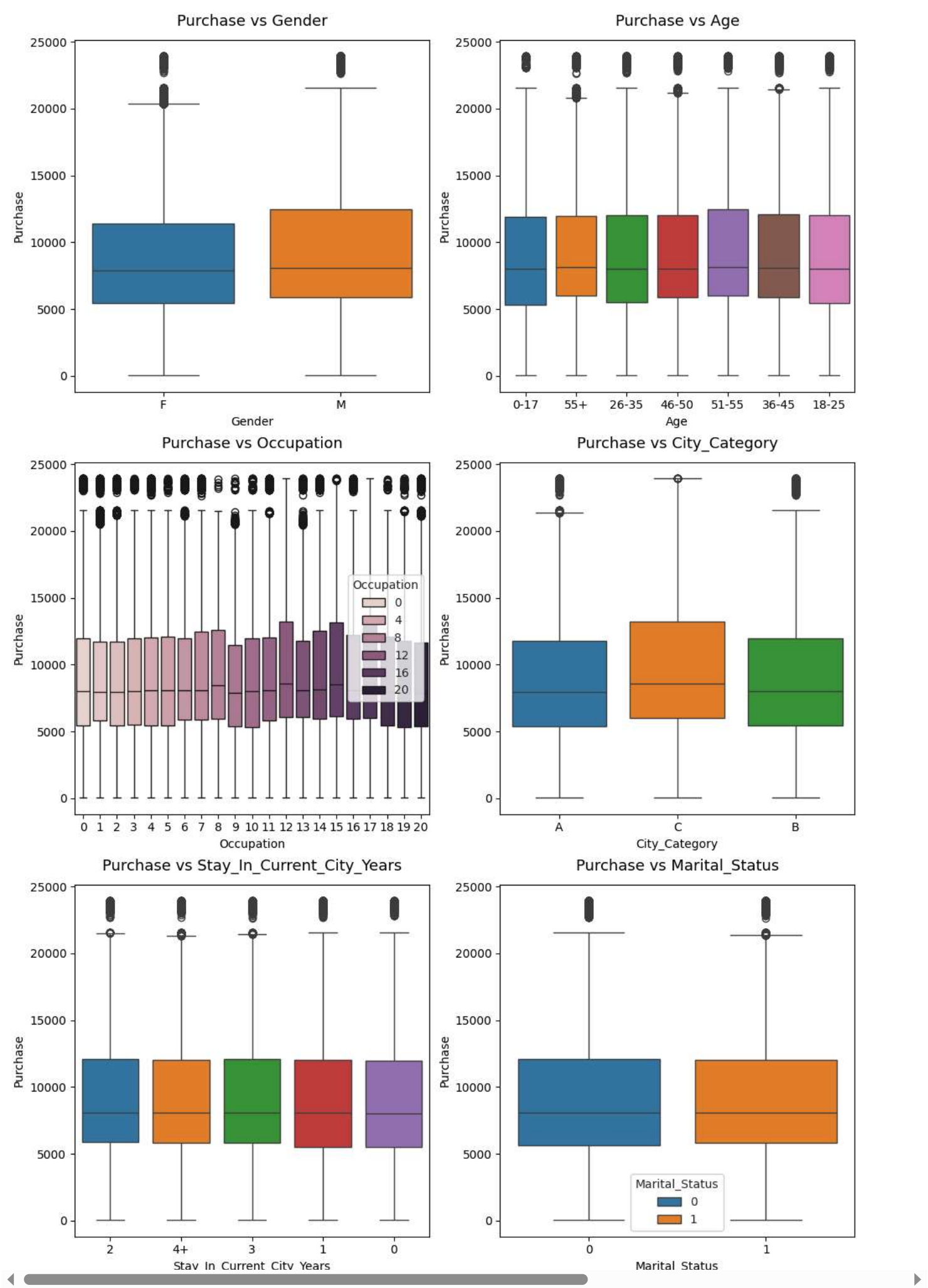
```
plt.figure(figsize = (8,6))
sns.countplot(data =df,x = 'Purchase_Range',order=df['Purchase_Range'].value_counts().index,hue = 'Gender')
plt.xticks(rotation = 45)
plt.show()
```



Most of the orders are from purchase category of 15K-20K in male and 6.5K-8K purchase category in Females

```
col_list = ['Gender','Age','Occupation','City_Category','Stay_In_Current_City_Years', 'Marital_Status']
fig,axs = plt.subplots(nrows = 3,ncols = 2,figsize = (12,12))
sns.set_style("white")
fig.subplots_adjust(top=1.3)

count = 0
for row in range(3):
  for col in range(2):
        sns.boxplot(data=df, y='Purchase', x=col_list[count], hue=col_list[count], ax=axs[row, col])
        axs[row, col].set_title(f"Purchase vs {col_list[count]}", pad=12, fontsize=13)
        count+=1
plt.show()
```



For male and female graph we can observer there are many outliers whose purchase amount for an order above 20k.

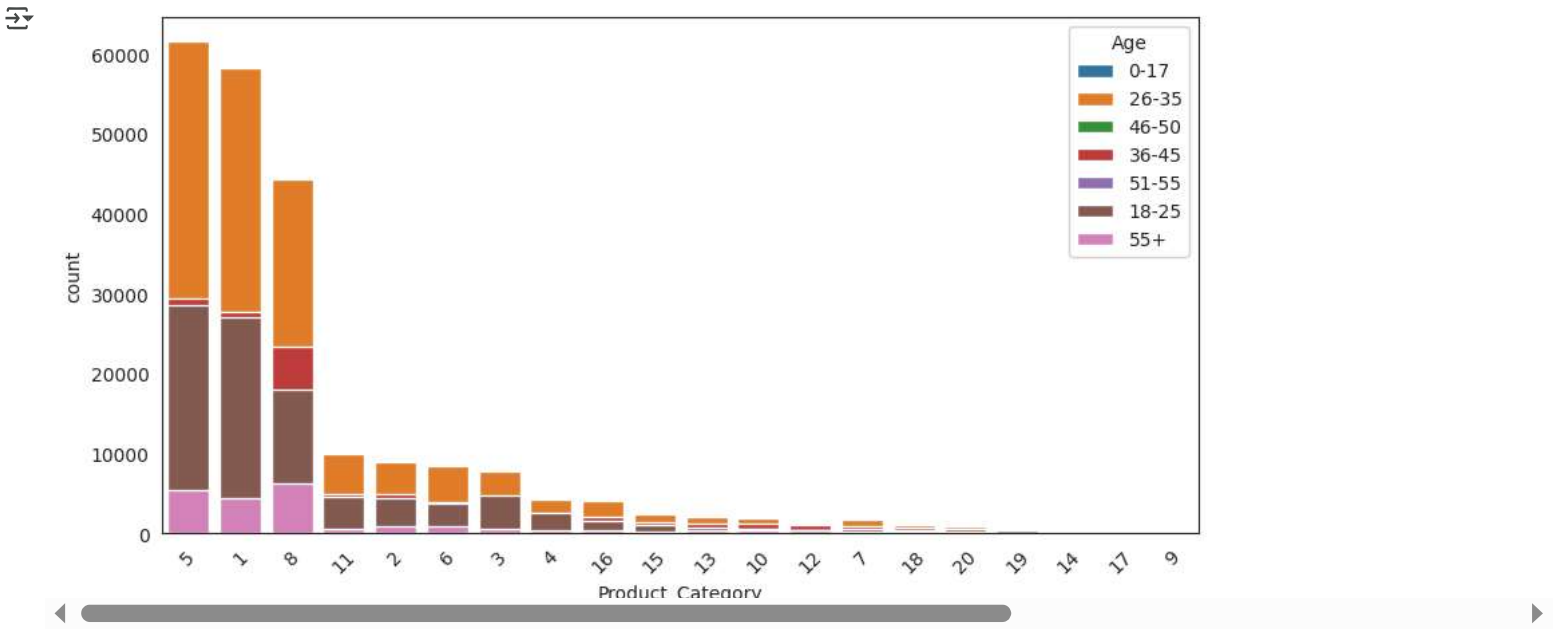
Median for category of 'stay_in_current_city_year','marital_status','age' are almost same amongs them.

We can conclude one thing that for all categorys there are outliers.

Product_category of 10 is having mean of purchase amount of almost 20k, from this we can conclude that this outlier orders are from prodouct category of 10 mostly.

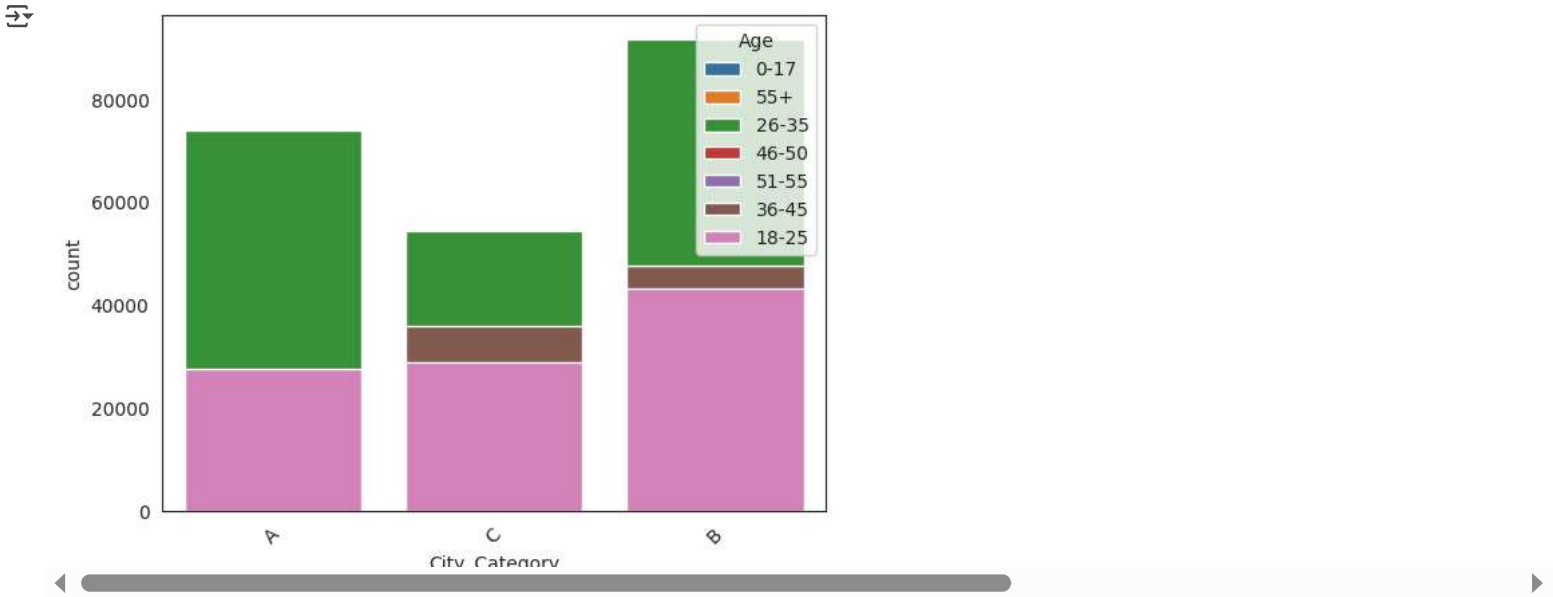
Q) Which age groups are purchasing each product category the most?

```
plt.figure(figsize=(10,5))
sns.countplot(data=df,x='Product_Category',hue='Age',dodge=False,order=df['Product_Category'].value_counts().index)
plt.xticks(rotation=45)
plt.show()
```



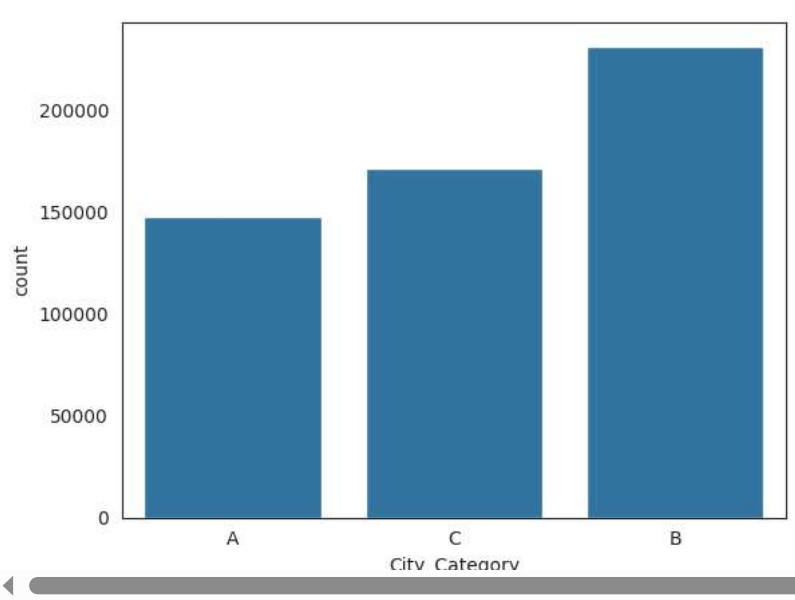
No of order for Product Category 5,1,8 are highly ordered in 26-35 age group

```
sns.countplot(data=df,x='City_Category',hue='Age',dodge=False)
plt.xticks(rotation=45)
plt.show()
```

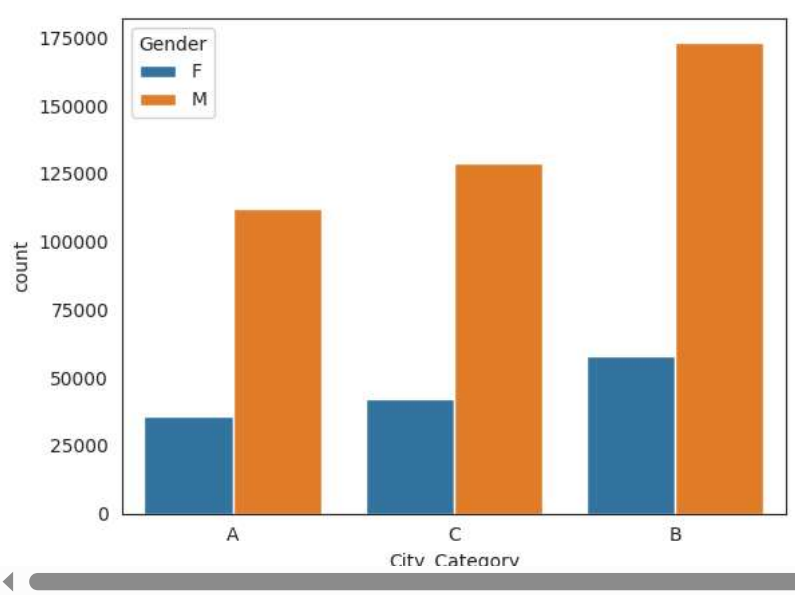


Age group between 26-35 and 18-25 have most no of orders in all city categories

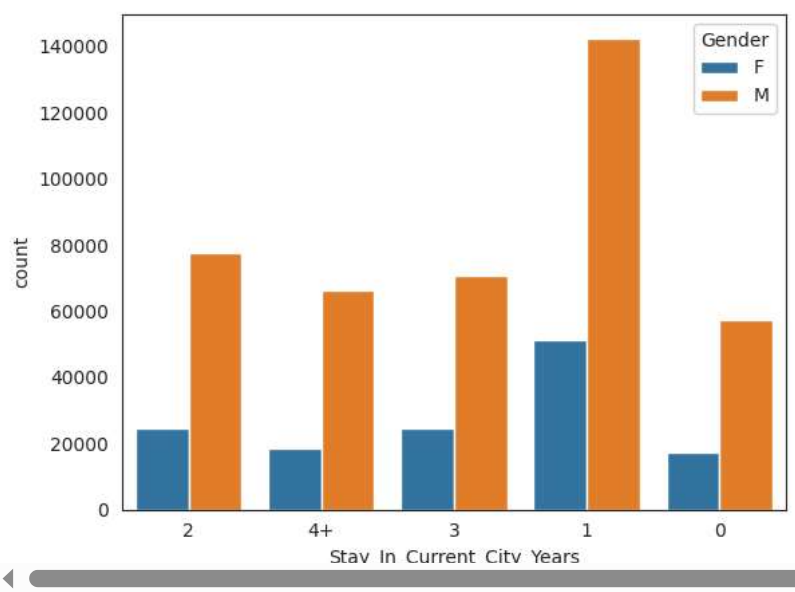
```
sns.countplot(data=df,x='City_Category')
plt.show()
```




```
sns.countplot(data=df,x='City_Category',hue='Gender')
plt.show()
```

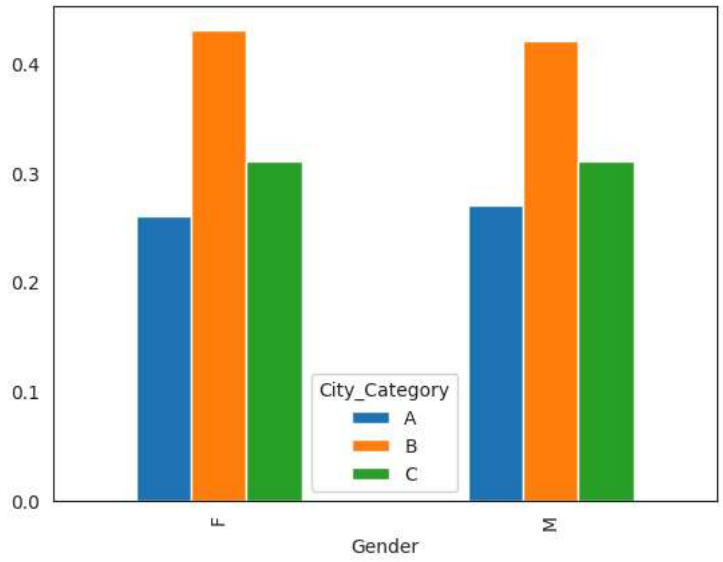


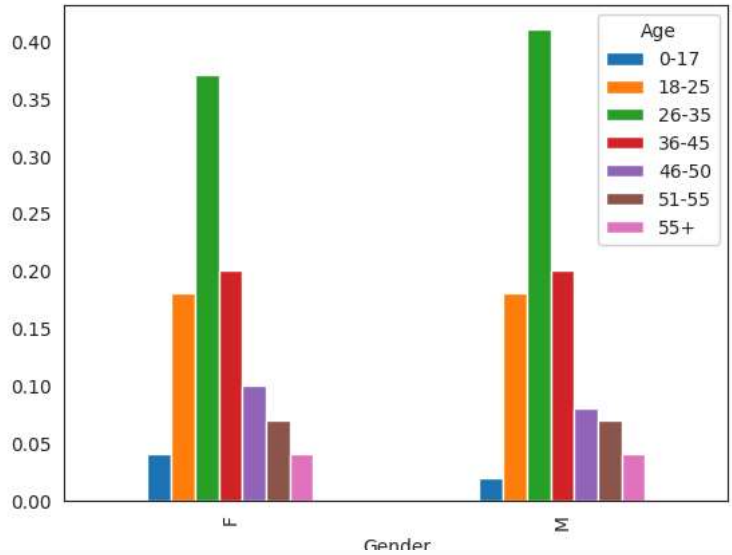

```
sns.countplot(data=df,x='Stay_In_Current_City_Years',hue='Gender')
plt.show()
```



Q)How are City Category and Age Group distributed by gender?

```
Cat_col = ['City_Category','Age']
for i in Cat_col:
  np.round(pd.crosstab(index=df['Gender'],columns=df[i],normalize='index'),2).plot(kind='bar')
  plt.show()
```



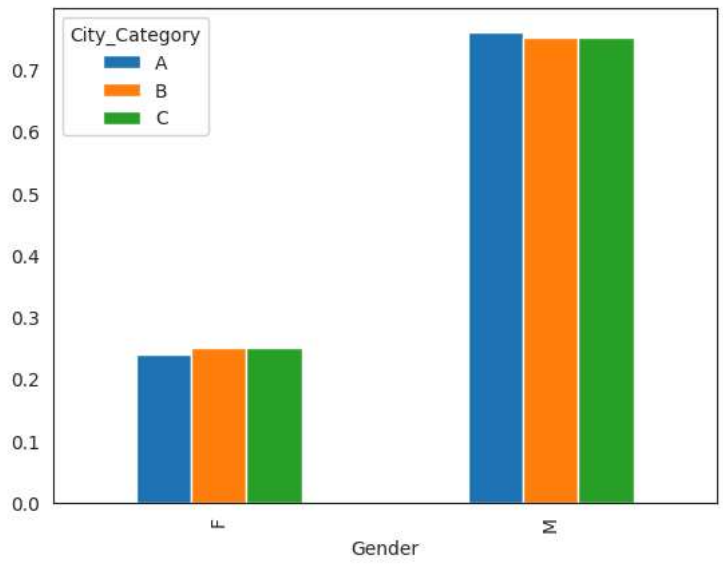


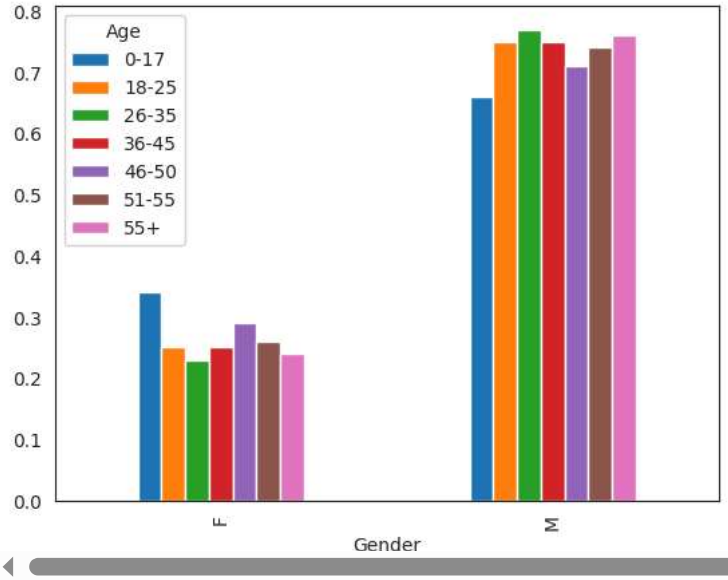
City_category are almost equally distributed gender wise.

In Age vs Gender, 26-35 group are more in both Male and Female and other age groups are almost equal in both genders.

Q)"What is the proportion of male and female customers within each City Category and Age Group

```
Cat_col=['City_Category','Age']
for i in Cat_col:
  np.round(pd.crosstab(index=df['Gender'],columns=df[i],normalize='columns'),2).plot(kind='bar')
  plt.show()
```





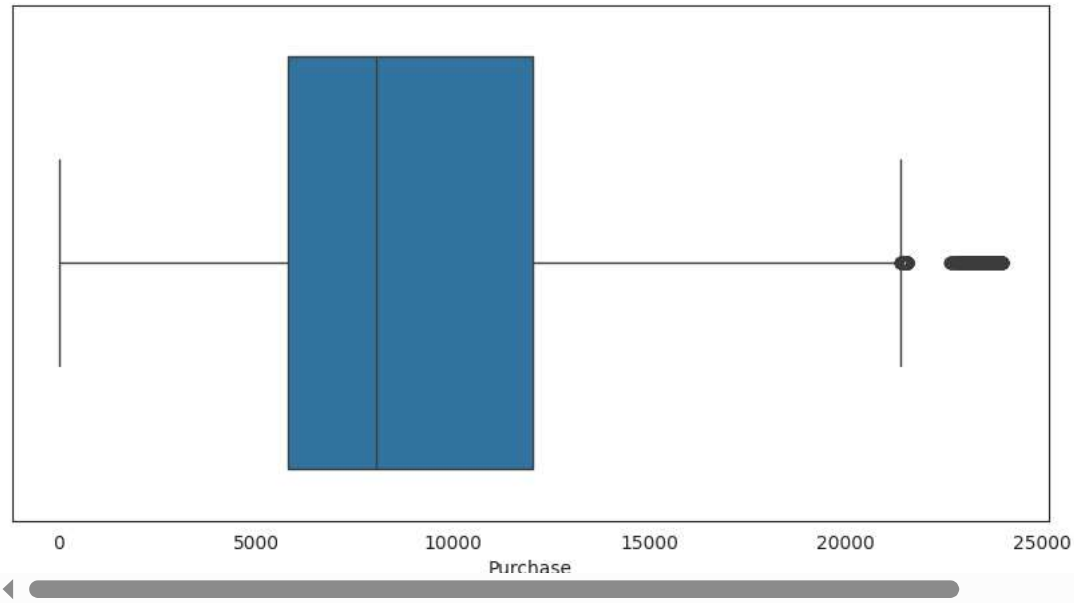
Male and Female Proportion in each city category is almost same Male proportion in all the age groups are same and 26-35 age group is little higher than all the age groups.

Female proportion is very less compared to male proportion but 0-17 age group contributes more and other age groups are less.

Company should focus on Female customers to increase their sales.

2-| Missing Value & Outlier Detection

```
plt.figure(figsize=(10,5))
sns.boxplot(data=df, x='Purchase')
plt.show()
```



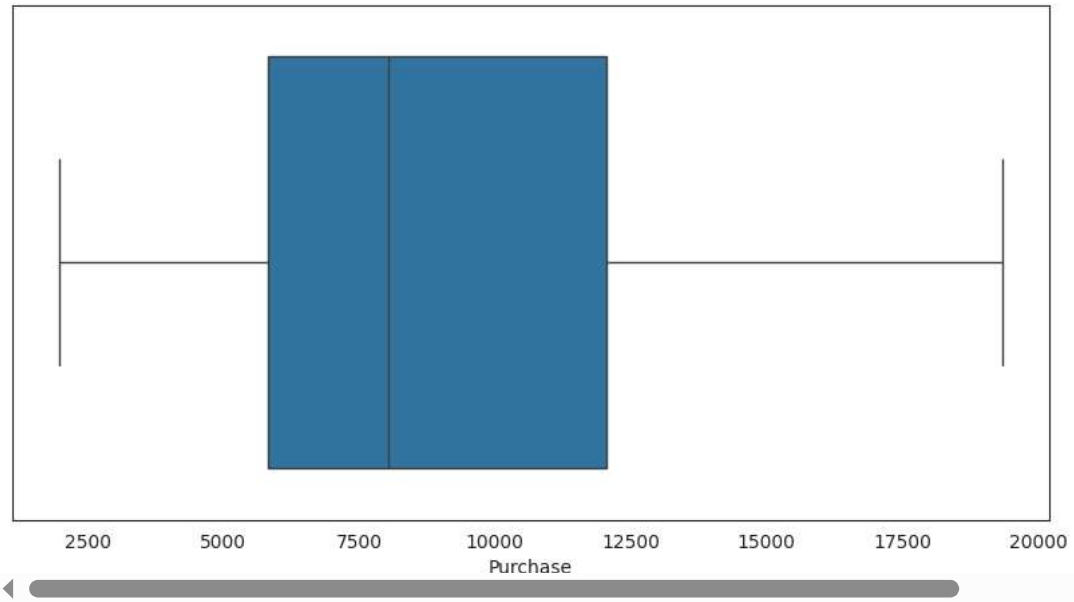
There are few outliers in purchase column

```
df_copy=df.copy()
percentile=df['Purchase'].quantile([0.05,0.95]).values
df_copy['Purchase']=np.clip(df['Purchase'],percentile[0],percentile[1])
```

```
df_copy
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase	Purcha
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	1984	
3	1000001	P00085442	F	0-17	10	A	2	0	12	1984	
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969	
...	...	...	...	...	...	...	...	...	...	...	...
550063	1006033	P00372445	M	51-55	13	B	1	1	20	1984	
550064	1006035	P00375436	F	26-35	1	C	3	0	20	1984	

```
plt.figure(figsize=(10,5))
sns.boxplot(data=df_copy, x='Purchase')
plt.show()
```



Outliers have been cleared using clipping by 5th percentile and 95th percentile for the purchase columns

Q) Which gender has the highest number of purchase outliers, and how can these outliers be leveraged to boost sales?

```
df_copyM=df[df['Gender']=='M']
float(np.percentile(df_copyM['Purchase'].dropna(),25)),float(np.percentile(df_copyM['Purchase'].dropna(),75))
```

```
(5863.0, 12454.0)
```

```
IQRM=(float(np.percentile(df_copyM['Purchase'].dropna(),75))-float(np.percentile(df_copyM['Purchase'].dropna(),25)))
IQRM
```

```
6591.0
```

```
UpperlimtM=float(np.percentile(df_copyM['Purchase'].dropna(),75))+1.5*IQRM
UpperlimtM
```

```
22340.5
```

```
df_copyM[df_copyM['Purchase']>UpperlimtM].shape[0]
```

```
1812
```

```
df_copyF=df[df['Gender']=='F']
float(np.percentile(df_copyF['Purchase'].dropna(),25)),float(np.percentile(df_copyF['Purchase'].dropna(),75))
```

```
(5433.0, 11400.0)
```

```
IQRF=float(np.percentile(df_copyF['Purchase'].dropna(),75)-float(np.percentile(df_copyF['Purchase'].dropna(),25)))
IQRF
```

```
5967.0
```

```
UpperlimtF=float(np.percentile(df_copyF['Purchase'].dropna(),75))+1.5*IQRF
UpperlimtF
```

```
20350.5
```

```
df_copyF[df_copyF['Purchase']>UpperlimtF].shape[0]
```

```
2065
```

Male outliers are 648 and Female outliers are 697 Company should focus on Male customers and promote or advertise product category because Male have more unique customers but they have less in purchasing high value products. so we should focus to advertise high value product i.e., product category 10.

CLT and Confidence Interval

```
import pandas as pd
import numpy as np
from scipy.stats import norm
```

```
df.groupby(['Gender'])['Purchase'].agg(['mean','std','count'])
```

Gender	mean	std	count
F	8734.565765	4767.233289	135809
M	9437.526040	5092.186210	414259

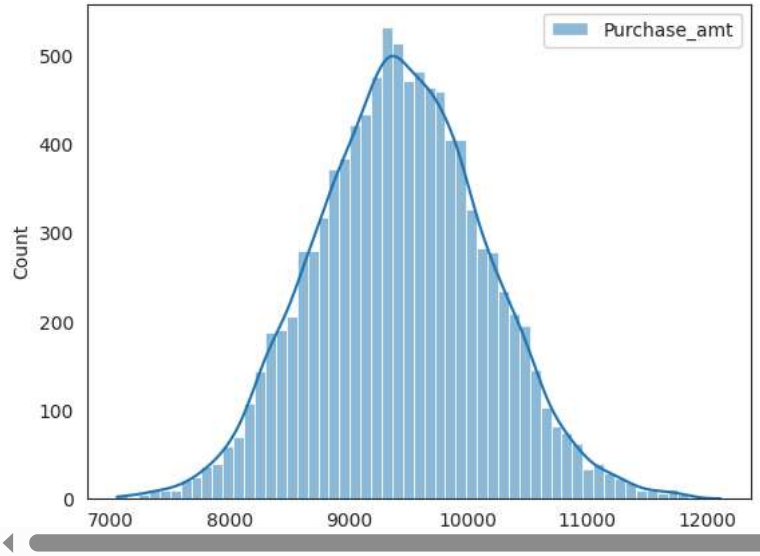
Do women spend more per transaction compared to men?

```
df_M=df.loc[df['Gender']=='M']['Purchase']
df_F=df.loc[df['Gender']=='F']['Purchase']
```

```
def anlaysis(data, sample_size,iterations):
  anlaysis_means=[]
  for reps in range(iterations):
    anlaysis_sample=np.random.choice(data,size=sample_size)
    anlaysis_means.append(np.mean(anlaysis_sample))
  exp_df = pd.DataFrame(anlaysis_means,columns =['Purchase_amt'])
  return sns.histplot(exp_df,kde= True)
```

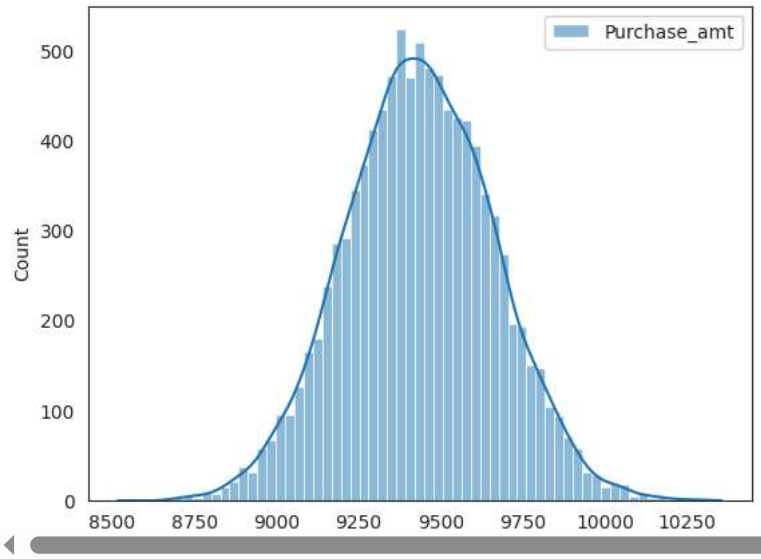
```
anlaysis(df_M,50,10000)
```

```
<Axes: ylabel='Count'>
```



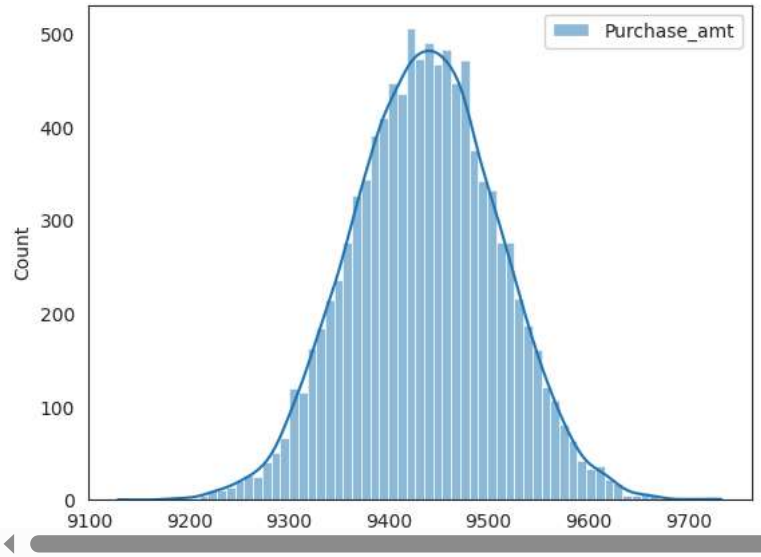

```
anlaysis(df_M,500,10000)
```

```
<Axes: ylabel='Count'>
```



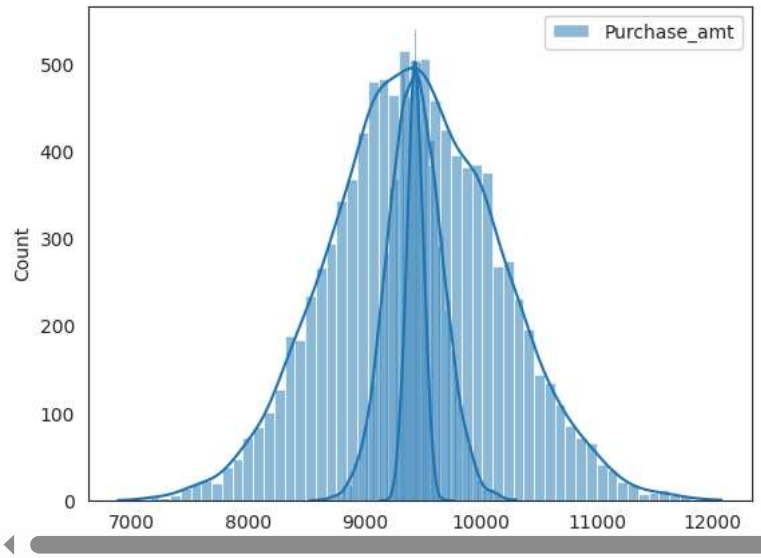

```
anlaysis(df_M,5000,10000)
```

```
<Axes: ylabel='Count'>
```




```
anlaysis(df_M,50,10000),anlaysis(df_M,500,10000),anlaysis(df_M,5000,10000)
```

```
(<Axes: ylabel='Count'>, <Axes: ylabel='Count'>, <Axes: ylabel='Count'>)
```



Q)Q:) How does the sample size affect the shape of the distributions of the means?

As the sample size increases, the more the narrow the bellcurve will become.

The more the sample size, less is the variance and our CI width also gets reduced making point estimates to population mean.

This distribution is normal now.

```
def bootstrapping_genstats(data,sample_size,iterations,confidence_level):
  bootstrapped_means =[]
```

```
  for reps in range(iterations):
    bootstr_sample = np.random.choice(data,size= sample_size)
    bootstrapped_means.append(np.mean(bootstr_sample))
  #return bootstrapped_means
  exp_df = pd.DataFrame(bootstrapped_means,columns =['Purchase_amt'])
  CI_lower = np.percentile(exp_df['Purchase_amt'], (100-confidence_level)/2)
  CI_Upper = np.percentile(exp_df['Purchase_amt'], ((100-confidence_level)/2 + confidence_level))
  CI_Width = CI_Upper - CI_lower
  return f"exp_df_mean {exp_df.mean()}",f"Pop_mean {data.mean()}", f"CI_lower {CI_lower}",f"CI_Upper {CI_Upper}",f"CI_Width {CI_Width}"


bootstrapping_genstats(df_M,50,1000,90),bootstrapping_genstats(df_M,500,1000,90),bootstrapping_genstats(df_M,5000,1000,90)
```

```
(('exp_df_mean Purchase_amt    9431.7679\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 8325.007',
  'CI_Upper 10563.055999999999',
  'CI_Width 2238.048999999999'),
 ('exp_df_mean Purchase_amt    9434.718068\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 9067.8516',
  'CI_Upper 9792.0753',
  'CI_Width 724.2237000000005'),
 ('exp_df_mean Purchase_amt    9436.21013\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 9330.209719999999',
  'CI_Upper 9552.27448',
  'CI_Width 222.06476000000112'))
```

```
bootstrapping_genstats(df_F,50,1000,90),bootstrapping_genstats(df_F,500,1000,90),bootstrapping_genstats(df_F,5000,1000,90)
```

```
(('exp_df_mean Purchase_amt    8744.70196\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 7685.499',
  'CI_Upper 9862.583999999999',
  'CI_Width 2177.084999999999'),
 ('exp_df_mean Purchase_amt    8740.511988\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 8381.3659',
  'CI_Upper 9101.9524',
  'CI_Width 720.5864999999994'),
 ('exp_df_mean Purchase_amt    8736.223964\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 8628.53607',
  'CI_Upper 8848.18456',
  'CI_Width 219.64848999999958'))
```

```
bootstrapping_genstats(df_M,50,1000,95),bootstrapping_genstats(df_M,500,1000,95),bootstrapping_genstats(df_M,5000,1000,95)
```

```
(('exp_df_mean Purchase_amt    9453.25516\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 8061.2745',
  'CI_Upper 10814.744',
  'CI_Width 2753.4695'),
 ('exp_df_mean Purchase_amt    9434.813606\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 8983.196899999999',
  'CI_Upper 9867.222549999999',
  'CI_Width 884.0256499999996'),
 ('exp_df_mean Purchase_amt    9433.894908\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 9288.268645',
  'CI_Upper 9577.447175',
  'CI_Width 289.1785299999992'))
```

```
bootstrapping_genstats(df_F,50,1000,95),bootstrapping_genstats(df_F,500,1000,95),bootstrapping_genstats(df_F,5000,1000,95)
```

```
(('exp_df_mean Purchase_amt    8762.8784\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 7489.22',
  'CI_Upper 10148.771999999999',
  'CI_Width 2659.5519999999988'),
 ('exp_df_mean Purchase_amt    8725.562944\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 8326.932649999999',
  'CI_Upper 9152.0938',
  'CI_Width 825.1611500000017'),
 ('exp_df_mean Purchase_amt    8733.525721\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 8605.324050000001',
  'CI_Upper 8863.98704',
  'CI_Width 258.6629899999989'))
```

```
bootstrapping_genstats(df_M,50,1000,99),bootstrapping_genstats(df_M,500,1000,99),bootstrapping_genstats(df_M,5000,1000,99)
```

```
(('exp_df_mean Purchase_amt    9466.10972\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 7821.4972',
  'CI_Upper 11270.6083',
  'CI_Width 3449.1111'),
 ('exp_df_mean Purchase_amt    9439.756006\ndtype: float64',
  'Pop_mean 9437.526040472265',
```

```
 'CI_lower 8925.27824',
 'CI_Upper 9978.26446',
 'CI_Width 1052.9862200000007'),
('exp_df_mean Purchase_amt    9442.053715\ndtype: float64',
 'Pop_mean 9437.526040472265',
 'CI_lower 9260.136745999998',
 'CI_Upper 9633.093942000001',
 'CI_Width 372.957196000003'))
```

```
bootstrapping_genstats(df_F,50,1000,99),bootstrapping_genstats(df_M,500,1000,99),bootstrapping_genstats(df_M,5000,1000,99)
```

```
(('exp_df_mean Purchase_amt    8749.57388\ndtype: float64',
  'Pop_mean 8734.565765155476',
  'CI_lower 7091.4508',
  'CI_Upper 10474.2663',
  'CI_Width 3382.8154999999997'),
 ('exp_df_mean Purchase_amt    9436.89357\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 8830.39673',
  'CI_Upper 10005.18013',
  'CI_Width 1174.7834000000003'),
 ('exp_df_mean Purchase_amt    9437.865795\ndtype: float64',
  'Pop_mean 9437.526040472265',
  'CI_lower 9258.748828',
  'CI_Upper 9605.994318',
  'CI_Width 347.24548999999934'))
```

Q) From the above calculated CLT answer the following questions.

1. Is the confidence interval computed using the entire dataset wider for one of the genders? Why is this the case?
2. How is the width of the confidence interval affected by the sample size?
3. Do the confidence intervals for different sample sizes overlap?

1. CI is calculated for confidence level of 90,95,99 and the conclusion is that none of gender has significant wider CI than other, there is slightly wider CI for male compare to female.
2. Width of CI gets tighten i.e shorter as we increase sample size(50,500,5000) shown above for both Male and Female and width of CI gets wider as we increase our confidence level.
3. Yes they do overlap for different sample sizes.As we increase sample size our CI width gets reduced and it give range of CI where our population mean will lie.

CI intervals are overlapping for male and female for small sample size 50 but when we increase to 500,5000 CI intervals are not overlapping and we can clearly see the difference in both range, this concludes for all type of confidence level.(90,95,99%)

Female

CI_Interval =(8500-8900) Male

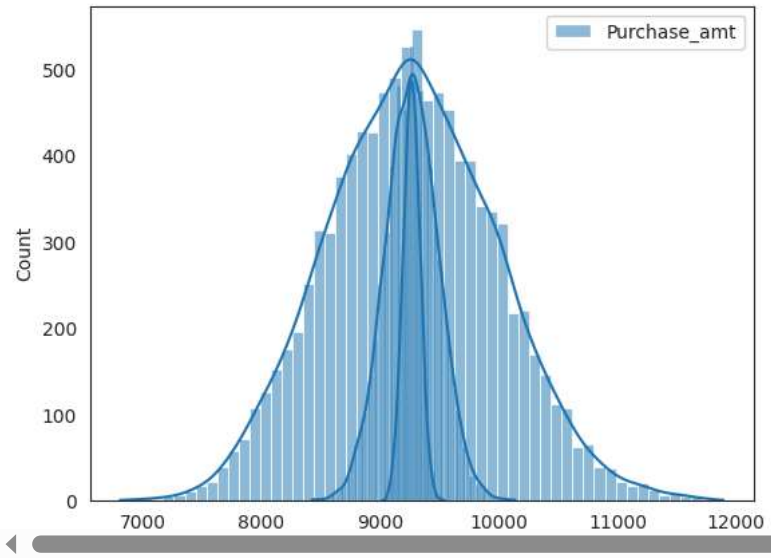
CI_Interval =(9200-9700) This approximate CI interval considering all type of 90,95,99% confidence. So in totality Male spendings are higher than female.

Q) How does Marital_Status affect the amount spent?

```
df_0=df[df['Marital_Status']==0]['Purchase']
df_1=df[df['Marital_Status']==1]['Purchase']
```

```
anlaysis(df_0,50,10000),anlaysis(df_0,500,10000),anlaysis(df_0,5000,10000)
```

```
(<Axes: ylabel='Count'>, <Axes: ylabel='Count'>, <Axes: ylabel='Count'>)
```



Q) How does the sample size affect the shape of the distributions of the means?

As we go higher for sample size , the more narrow the bell curve becomes. The more the sample size, less is the variance and our CI width also gets reduced making point estimates to population mean. This distribution is normal now.

```
bootstrapping_genstats(df_0,50,1000,95),bootstrapping_genstats(df_0,500,1000,95),bootstrapping_genstats(df_0,5000,1000,95)
```

```
((('exp_df_mean Purchase_amt    9250.76024\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 7915.513',
   'CI_Upper 10679.625',
   'CI_Width 2764.112'),
  ('exp_df_mean Purchase_amt    9263.397918\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 8795.75775',
   'CI_Upper 9689.8682',
   'CI_Width 894.1104500000001'),
  ('exp_df_mean Purchase_amt    9265.897595\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 9133.7659',
   'CI_Upper 9393.56999',
   'CI_Width 259.80408999999963'))
```

```
bootstrapping_genstats(df_1,50,1000,95),bootstrapping_genstats(df_1,500,1000,95),bootstrapping_genstats(df_1,5000,1000,95)
```

```
((('exp_df_mean Purchase_amt    9286.9902\ndtype: float64',
   'Pop_mean 9261.174574082374',
   'CI_lower 7922.312',
   'CI_Upper 10608.609',
   'CI_Width 2686.2970000000005'),
  ('exp_df_mean Purchase_amt    9265.308566\ndtype: float64',
   'Pop_mean 9261.174574082374',
   'CI_lower 8843.401',
   'CI_Upper 9700.9235',
   'CI_Width 857.522500000001'),
  ('exp_df_mean Purchase_amt    9259.785797\ndtype: float64',
   'Pop_mean 9261.174574082374',
   'CI_lower 9120.040589999999',
   'CI_Upper 9391.14524',
   'CI_Width 271.10465000000113'))
```

```
bootstrapping_genstats(df_0,50,1000,90),bootstrapping_genstats(df_0,500,1000,90),bootstrapping_genstats(df_0,5000,1000,90)
```

```
((('exp_df_mean Purchase_amt    9266.10122\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 8126.224',
   'CI_Upper 10388.276',
   'CI_Width 2262.0519999999997'),
  ('exp_df_mean Purchase_amt    9253.005286\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 8886.08',
   'CI_Upper 9626.1648',
   'CI_Width 740.0848000000005'),
  ('exp_df_mean Purchase_amt    9270.598078\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 9154.15527',
   'CI_Upper 9390.967260000001',
   'CI_Width 236.81199000000197'))
```

```
bootstrapping_genstats(df_1,50,1000,90),bootstrapping_genstats(df_1,500,1000,90),bootstrapping_genstats(df_1,5000,1000,90)
```

```
((('exp_df_mean Purchase_amt    9314.70818\ndtype: float64',
   'Pop_mean 9261.174574082374',
   'CI_lower 8108.241',
   'CI_Upper 10535.693',
   'CI_Width 2427.4519999999993'),
  ('exp_df_mean Purchase_amt    9254.291196\ndtype: float64',
   'Pop_mean 9261.174574082374',
   'CI_lower 8873.098199999999',
   'CI_Upper 9620.2269',
   'CI_Width 747.1287000000011'),
  ('exp_df_mean Purchase_amt    9258.914205\ndtype: float64',
   'Pop_mean 9261.174574082374',
   'CI_lower 9143.15861',
   'CI_Upper 9371.67431',
   'CI_Width 228.51569999999992'))
```

```
bootstrapping_genstats(df_0,50,1000,99),bootstrapping_genstats(df_0,500,1000,99),bootstrapping_genstats(df_0,5000,1000,99)
```

```
((('exp_df_mean Purchase_amt    9254.2518\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 7465.9637',
   'CI_Upper 11156.9898',
   'CI_Width 3691.026099999999'),
  ('exp_df_mean Purchase_amt    9261.623086\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 8664.2445',
   'CI_Upper 9873.28012',
   'CI_Width 1209.0356199999987'),
  ('exp_df_mean Purchase_amt    9267.284296\ndtype: float64',
   'Pop_mean 9265.907618921507',
   'CI_lower 9074.725788000002',
```

```
 'CI_Upper 9452.875914',
 'CI_Width 378.15012599999864'))
```

```
bootstrapping_genstats(df_1,50,1000,99),bootstrapping_genstats(df_1,500,1000,99),bootstrapping_genstats(df_1,5000,1000,99)
```

```
(('exp_df_mean Purchase_amt    9263.11268\ndtype: float64',
  'Pop_mean 9261.174574082374',
  'CI_lower 7620.825900000001',
  'CI_Upper 10985.371500000001',
  'CI_Width 3364.5456000000004'),
 ('exp_df_mean Purchase_amt    9260.886016\ndtype: float64',
  'Pop_mean 9261.174574082374',
  'CI_lower 8693.663569999999',
  'CI_Upper 9816.887850000001',
  'CI_Width 1123.2242800000022'),
 ('exp_df_mean Purchase_amt    9261.736964\ndtype: float64',
  'Pop_mean 9261.174574082374',
  'CI_lower 9063.575262999999',
  'CI_Upper 9436.618666',
  'CI_Width 373.0434030000015'))
```

Q:- From the above calculated CLT answer the following questions.

1. Is the confidence interval computed using the entire dataset wider for one of the marital_status? Why is this the case?
2. How is the width of the confidence interval affected by the sample size?
3. Do the confidence intervals for different sample sizes overlap?

Ans

1. For 90%,95%,99% Confidence level we cant conclude there is significant wider CI for marital_status of 0 or 1.
2. Width of CI gets tighten as we increase sample size(50,500,5000) shown above for both 0 and 1 marital_status for 90%,95%,99% confidence level giving us the point estimates to population mean.
3. Yes they do overlap for different sample sizes.As we increase sample size our CI width gets reduced and it give range of CI where our population mean will lie. CI intervals are overlapping for 0 and 1 for 90,95%,99% for sample of 50,500 and 5000 and we cant conclude anything for 0 or 1 independently.We cannot confidently say that there is a significant difference between them in spending based on marital status.

CI intervals for 0 and 1 9000-9500 for both 0 and 1 marital_status for 90,95%,99% condifence. For every confidence level range lies almost same.

Q) How does Age affect the amount spent?

```
df['Age'].value_counts()
```

Age	count
26-35	219587
36-45	110013
18-25	99660
46-50	45701
51-55	38501
55+	21504
0-17	15102

```
df_26_35 = df.loc[df['Age']=='26-35']['Purchase']
df_36_45 = df.loc[df['Age']=='36-45']['Purchase']
df_18_25 = df.loc[df['Age']=='18-25']['Purchase']
df_0_17 = df.loc[df['Age']=='0-17']['Purchase']
df_55_ = df.loc[df['Age']=='55+']['Purchase']
df_46_50 = df.loc[df['Age']=='46-50']['Purchase']
df_51_55 = df.loc[df['Age']=='51-55']['Purchase']
```

```
bootstrapping_genstats(df_0_17,5000,1000,90)
```

```
('exp_df_mean Purchase_amt    8933.707367\ndtype: float64',
 'Pop_mean 8933.464640444974',
 'CI_lower 8812.5122',
 'CI_Upper 9052.97852',
 'CI_Width 240.46632000000136')
```

```
bootstrapping_genstats(df_18_25,5000,1000,90)
```

```
('exp_df_mean Purchase_amt    9168.548803\ndtype: float64',
 'Pop_mean 9169.663606261289',
```

```
       'CI_lower 9054.799079999999',
       'CI_Upper 9288.65714',
       'CI_Width 233.85806000000048')


bootstrapping_genstats(df_26_35,5000,1000,90)

('exp_df_mean Purchase_amt    9251.797787\ndtype: float64',
       'Pop_mean 9252.690632869888',
       'CI_lower 9140.520980000001',
       'CI_Upper 9360.89',
       'CI_Width 220.36901999999827')


bootstrapping_genstats(df_36_45,5000,1000,90)

('exp_df_mean Purchase_amt    9333.786854\ndtype: float64',
       'Pop_mean 9331.350694917874',
       'CI_lower 9211.728650000001',
       'CI_Upper 9446.76561',
       'CI_Width 235.03695999999945')


bootstrapping_genstats(df_46_50,5000,1000,90)

('exp_df_mean Purchase_amt    9211.47752\ndtype: float64',
       'Pop_mean 9208.625697468327',
       'CI_lower 9095.18186',
       'CI_Upper 9324.43132',
       'CI_Width 229.2494599999991')


bootstrapping_genstats(df_51_55,5000,1000,90)

('exp_df_mean Purchase_amt    9527.96805\ndtype: float64',
       'Pop_mean 9534.808030960236',
       'CI_lower 9409.6388',
       'CI_Upper 9644.313',
       'CI_Width 234.67419999999947')


bootstrapping_genstats(df_55_,5000,1000,90)

('exp_df_mean Purchase_amt    9337.505091\ndtype: float64',
       'Pop_mean 9336.280459449405',
       'CI_lower 9222.014949999999',
       'CI_Upper 9456.17398',
       'CI_Width 234.1590300000007')


bootstrapping_genstats(df_0_17,5000,1000,95)

('exp_df_mean Purchase_amt    8936.634118\ndtype: float64',
       'Pop_mean 8933.464640444974',
       'CI_lower 8802.781665',
       'CI_Upper 9070.709055000001',
       'CI_Width 267.92739000000074')


bootstrapping_genstats(df_18_25,5000,1000,95)

('exp_df_mean Purchase_amt    9168.712286\ndtype: float64',
       'Pop_mean 9169.663606261289',
       'CI_lower 9022.302959999999',
       'CI_Upper 9311.712145000001',
       'CI_Width 289.40918500000225')


bootstrapping_genstats(df_26_35,5000,1000,95)

('exp_df_mean Purchase_amt    9249.848964\ndtype: float64',
       'Pop_mean 9252.690632869888',
       'CI_lower 9096.742390000001',
       'CI_Upper 9385.645695000001',
       'CI_Width 288.9033049999998')
```

```
bootstrapping_genstats(df_26_35,5000,1000,95)
```

```
('exp_df_mean Purchase_amt    9249.848964\ndtype: float64',
 'Pop_mean 9252.690632869888',
 'CI_lower 9096.742390000001',
 'CI_Upper 9385.645695000001',
 'CI_Width 288.9033049999998')
```

```
[119] bootstrapping_genstats(df_36_45,5000,1000,95)
```

```
('exp_df_mean Purchase_amt    9328.605812\ndtype: float64',
 'Pop_mean 9331.350694917874',
 'CI_lower 9193.463815000001',
 'CI_Upper 9463.164395',
 'CI_Width 269.7005799999988')
```

```
[120] bootstrapping_genstats(df_46_50,5000,1000,95)
```

```
('exp_df_mean Purchase_amt    9206.440566\ndtype: float64',
 'Pop_mean 9208.625697468327',
 'CI_lower 9065.212725',
 'CI_Upper 9348.271444999998',
 'CI_Width 283.0587199999991')
```

```
[123] bootstrapping_genstats(df_51_55,5000,1000,95)
```

```
('exp_df_mean Purchase_amt    9534.660306\ndtype: float64',
 'Pop_mean 9534.808030960236',
 'CI_lower 9394.78107',
 'CI_Upper 9680.752204999999',
 'CI_Width 285.97113499999796')
```

```
[124] bootstrapping_genstats(df_55_,5000,1000,95)
```

```
('exp_df_mean Purchase_amt    9336.02058\ndtype: float64',
 'Pop_mean 9336.280459449405',
 'CI_lower 9201.220685',
 'CI_Upper 9478.237635',
 'CI_Width 277.01694999999927')
```

```
bootstrapping_genstats(df_0_17,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    8937.617664\ndtype: float64',
 'Pop_mean 8933.464640444974',
 'CI_lower 8760.692088',
 'CI_Upper 9120.106701',
 'CI_Width 359.4146130000008')
```

```
[126] bootstrapping_genstats(df_18_25,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    9170.495257\ndtype: float64',
 'Pop_mean 9169.663606261289',
 'CI_lower 8990.98926',
 'CI_Upper 9345.173622',
 'CI_Width 354.18436199999996')
```

```
[127] bootstrapping_genstats(df_26_35,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    9258.750223\ndtype: float64',
 'Pop_mean 9252.690632869888',
 'CI_lower 9088.236534',
 'CI_Upper 9435.178326',
 'CI_Width 346.9417919999996')
```

```
[128] bootstrapping_genstats(df_36_45,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    9331.012068\ndtype: float64',
 'Pop_mean 9331.350694917874',
 'CI_lower 9164.226907999999',
 'CI_Upper 9507.648595999999',
 'CI_Width 343.42168800000036')
```

```
[129] bootstrapping_genstats(df_46_50,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    9211.387298\ndtype: float64',
 'Pop_mean 9208.625697468327',
 'CI_lower 9037.976326999998',
 'CI_Upper 9392.942647',
 'CI_Width 354.96632000000136')
```

```
[130] bootstrapping_genstats(df_51_55,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    9532.673477\ndtype: float64',
 'Pop_mean 9534.808030960236',
 'CI_lower 9344.841872',
 'CI_Upper 9703.500471',
 'CI_Width 358.6585989999985')
```

```
[131] bootstrapping_genstats(df_55_,5000,1000,99)
```

```
('exp_df_mean Purchase_amt    9333.934533\ndtype: float64',
 'Pop_mean 9336.280459449405',
 'CI_lower 9148.246265',
 'CI_Upper 9510.423663',
 'CI_Width 362.1773979999998')
```

Q) From the above calculated CLT answer the following questions.

1. Is the confidence interval computed using the entire dataset wider for one of the marital_status? Why is this the case?
2. How is the width of the confidence interval affected by the sample size?
3. Do the confidence intervals for different sample sizes overlap?

1. For 90%,95%,99% Confidence level we cant conclude there is significant wider CI for any age group.
2. Width of CI gets tighten as we increase sample size(50,500,5000) shown above 3. for 90%,95%,99% confidence level giving us the point estimates to population mean.

CI interval for different sample sizes overlap for 5000 sample size. CI intervals are overlapping for every age group except 0-17 for 90,95%,99% for sample size of 5000. There are no significant difference in range of any group.

Age group of 51-55 has spend more compare to other age group i.e in the range of (9300-9700 approx) We can also conclude for age group of 0-17 spendings are less as their CI intervals are also lower compare to other age group (8700-9100 approx) and can be differentiated from others. All the other group are spending in the range of (9100-9500 approx)

Final Insights:

1. Unique male customers are 72% and female 28% and total amount spent by them also are in same proportion i.e we can say company is earning more from male customers.
2. Age group 26-35 made orders the most i.e 40% orders are from these age group although contribution of female in this group is very small.
3. Age group of 0-17 has significant proportion of females compare to male. For age group after 35, female contribution is being reduced.
4. Unique customers are much higher for city category C i.e 54% whereas for B it is 29% but number of orders(42%) and total amount spent(42%) by city category B are higher than C.
5. Average order amount for all age group is around (9100-9500) except for age group 0-17 which has mean of range(8700-9100).
6. Average order amount for male is in range (9100-9700), where as for female it is low in range of (8500-8900).
7. Average order amount in terms of marital status 0 and 1 are almost same.We can say there is no differentiation in terms of spending power between marital status.
8. Numbers of female customers are low as compare to male in all the age group except 0-17.
9. Females are more contributing to outliers(2065 number of order)i.e orders above approx 20k given their unique customers are very low,whereas for male 1812 number of order.
10. Producs in range 15-20k were ordered most number of times.Product category of 5,1,8 were mostly bought.
11. Product_category 2,10 are being bought mostly from city category B. C city category has highest average amount of order value i.e 9800 but lesser number of orders.
12. Product category 10 is the costliest category. Numbers of orders for product category 5,1,8 for age group 26-35 are much higher.

Recommendations:

1. Company should focus on giving promotional offers to female who are in the age group of 26-35 more because most number of orders are from that age group and female contribution is very low
2. Company should work on introducing products which has around 15-20k average value almost equivalent to product category 10.
3. Company should give offer/ad campaign to male customers for the product which has an average value of above 20k because female portion for this value range is significant compare to male.(outlier of female are almost comparable to male shown above in INSIGHTS)
4. Sales are very low for city category C and to increase sales in this mangement should advertisement for product_category 2,10 which are low in C but in city B number of order for this product category is strong given the unique customers are very low compare to city C.
5. Cashback should be given to customer of city category A for doing 1st time purchase because unique customers are very low, management should work on for customer acquisition in city category A.
6. Management should work on more products for age group of above 35 and above because those age group spending are reducing.
7. Management should give festival offers on products category other than 5,1,8 to boost sales.
8. Management in overall should focus on customer acquisation for female as this will increase sales.