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

```
df=pd.read_csv(r'C:\Users\pushp\OneDrive\Desktop\
Python_Diwali_Sales_Analysis-main\Diwali Sales Data.csv' ,
encoding='unicode_escape')
```

```
df.shape
```

```
(11251, 15)
```

```
df.head()
```

```
   User_ID  Cust_name Product_ID Gender Age Group  Age  Marital_Status  
\
0  1002903  Sanskriti  P00125942      F     26-35   28               0

1  1000732     Kartik  P00110942      F     26-35   35               1

2  1001990      Bindu  P00118542      F     26-35   35               1

3  1001425     Sudevi  P00237842      M      0-17   16               0

4  1000588       Joni  P00057942      M     26-35   28               1


            State       Zone       Occupation Product_Category  Orders  
\
0     Maharashtra   Western       Healthcare             Auto       1

1  Andhra Pradesh  Southern             Govt             Auto       3

2   Uttar Pradesh   Central       Automobile             Auto       3

3       Karnataka  Southern     Construction             Auto       2

4         Gujarat   Western  Food Processing             Auto       2


    Amount  Status  unnamed1
0  23952.0     NaN       NaN
1  23934.0     NaN       NaN
2  23924.0     NaN       NaN
3  23912.0     NaN       NaN
4  23877.0     NaN       NaN
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11251 non-null  int64  
 1   Cust_name         11251 non-null  object 
 2   Product_ID        11251 non-null  object 
 3   Gender            11251 non-null  object 
 4   Age Group         11251 non-null  object 
 5   Age               11251 non-null  int64  
 6   Marital_Status    11251 non-null  int64  
 7   State             11251 non-null  object 
 8   Zone              11251 non-null  object 
 9   Occupation        11251 non-null  object 
 10  Product_Category  11251 non-null  object 
 11  Orders            11251 non-null  int64  
 12  Amount            11239 non-null  float64
 13  Status            0 non-null      float64
 14  unnamed1          0 non-null      float64
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

```
df.drop(["Status" ,"unnamed1"] ,axis=1 ,inplace=True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11251 non-null  int64  
 1   Cust_name         11251 non-null  object 
 2   Product_ID        11251 non-null  object 
 3   Gender            11251 non-null  object 
 4   Age Group         11251 non-null  object 
 5   Age               11251 non-null  int64  
 6   Marital_Status    11251 non-null  int64  
 7   State             11251 non-null  object 
 8   Zone              11251 non-null  object 
 9   Occupation        11251 non-null  object 
 10  Product_Category  11251 non-null  object 
 11  Orders            11251 non-null  int64  
 12  Amount            11239 non-null  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

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

```
User_ID              0
Cust_name            0
Product_ID           0
Gender               0
Age Group            0
Age                  0
Marital_Status       0
State                0
Zone                 0
Occupation           0
Product_Category     0
Orders               0
Amount              12
dtype: int64
```

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

```
np.int64(12)
```

```
df.dropna(inplace=True)
```

```
df.shape
```

```
(11239, 13)
```

Change Data Type

```
df["Amount"]=df["Amount"].astype("int")
```

```
df["Amount"].dtypes
```

```
dtype('int64')
```

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation',
'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

Rename Columns

```
df.rename(columns={"Marital_Status" : "Married"} )
```

```
       User_ID    Cust_name Product_ID Gender Age Group  Age  Married
\
0      1002903    Sanskriti  P00125942      F     26-35   28        0
```

```
1      1000732       Kartik  P00110942      F     26-35   35         1

2      1001990        Bindu  P00118542      F     26-35   35         1

3      1001425       Sudevi  P00237842      M      0-17   16         0

4      1000588         Joni  P00057942      M     26-35   28         1

...        ...          ...        ...    ...       ...  ...       ...

11246  1000695      Manning  P00296942      M     18-25   19         1

11247  1004089  Reichenbach  P00171342      M     26-35   33         0

11248  1001209        Oshin  P00201342      F     36-45   40         0

11249  1004023       Noonan  P00059442      M     36-45   37         0

11250  1002744      Brumley  P00281742      F     18-25   19         0


                State      Zone       Occupation Product_Category
Orders  \
0         Maharashtra   Western       Healthcare             Auto
1
1      Andhra Pradesh  Southern             Govt             Auto
3
2       Uttar Pradesh   Central       Automobile             Auto
3
3           Karnataka  Southern     Construction             Auto
2
4             Gujarat   Western  Food Processing             Auto
2
...               ...       ...              ...              ...
...
11246     Maharashtra   Western         Chemical           Office
4
11247         Haryana  Northern       Healthcare       Veterinary
3
11248  Madhya Pradesh   Central         Textile           Office
4
11249       Karnataka  Southern      Agriculture           Office
3
11250     Maharashtra   Western       Healthcare           Office
3

       Amount
0       23952
1       23934
```

```
2        23924
3        23912
4        23877
...        ...
11246      370
11247      367
11248      213
11249      206
11250      188

[11239 rows x 13 columns]
```

```
df.describe()
```

```
            User_ID           Age  Marital_Status        Orders
Amount
count  1.123900e+04  11239.000000    11239.000000  11239.000000
11239.000000
mean   1.003004e+06     35.410357        0.420055      2.489634
9453.610553
std    1.716039e+03     12.753866        0.493589      1.114967
5222.355168
min    1.000001e+06     12.000000        0.000000      1.000000
188.000000
25%    1.001492e+06     27.000000        0.000000      2.000000
5443.000000
50%    1.003064e+06     33.000000        0.000000      2.000000
8109.000000
75%    1.004426e+06     43.000000        1.000000      3.000000
12675.000000
max    1.006040e+06     92.000000        1.000000      4.000000
23952.000000
```

Describe only on individual columns

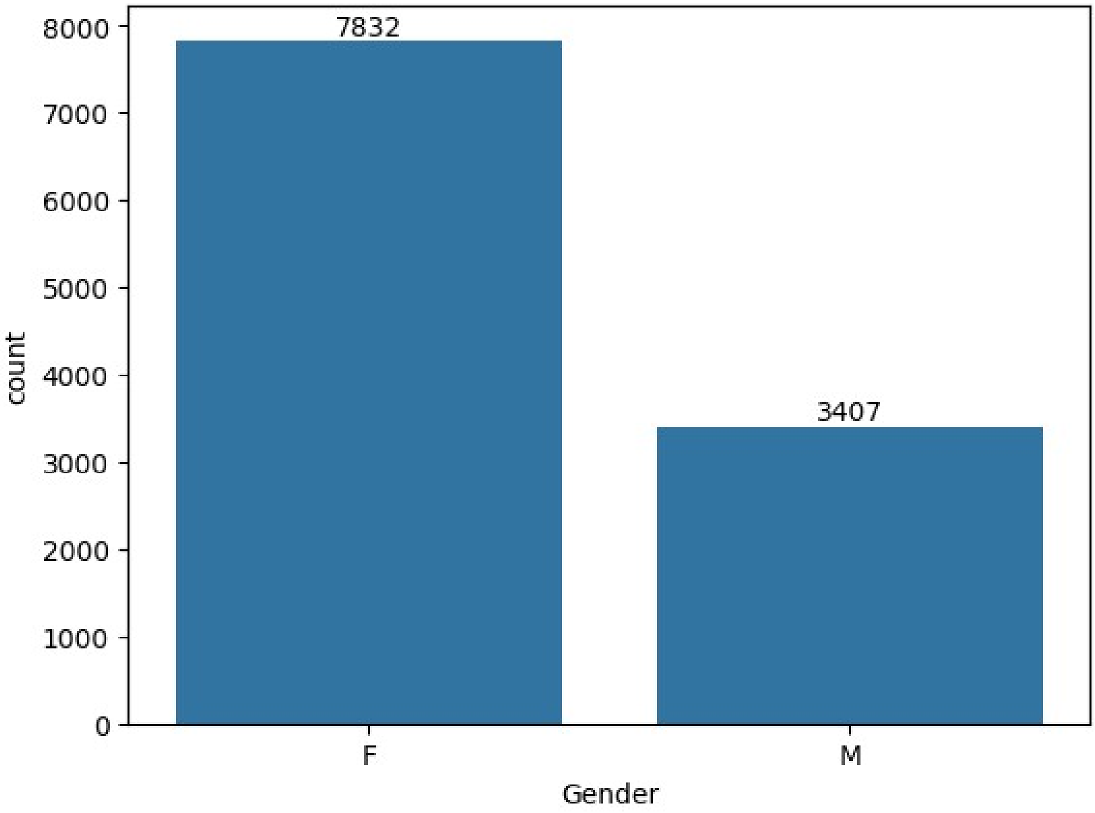
```
df["Amount"].describe()
```

```
count    11239.000000
mean      9453.610553
std       5222.355168
min        188.000000
25%       5443.000000
50%       8109.000000
75%      12675.000000
max      23952.000000
Name: Amount, dtype: float64
```

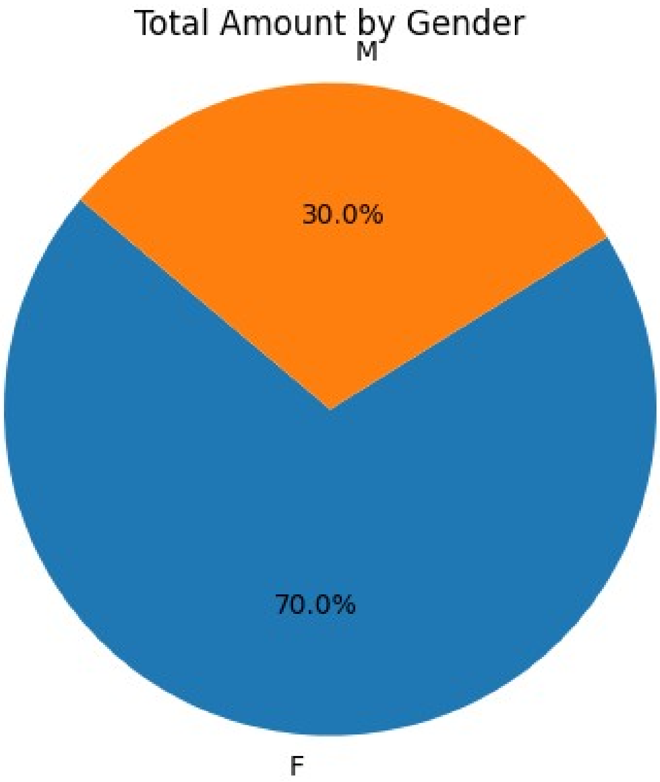
NOW ITS EDA

Gender

```
ax=sns.countplot(x="Gender" ,data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```



```
gender_amount = df.groupby("Gender")["Amount"].sum()

plt.pie(gender_amount.values, labels=gender_amount.index,
autopct='%1.1f%%', startangle=140)
plt.title("Total Amount by Gender")
plt.axis('equal')
plt.show()
```

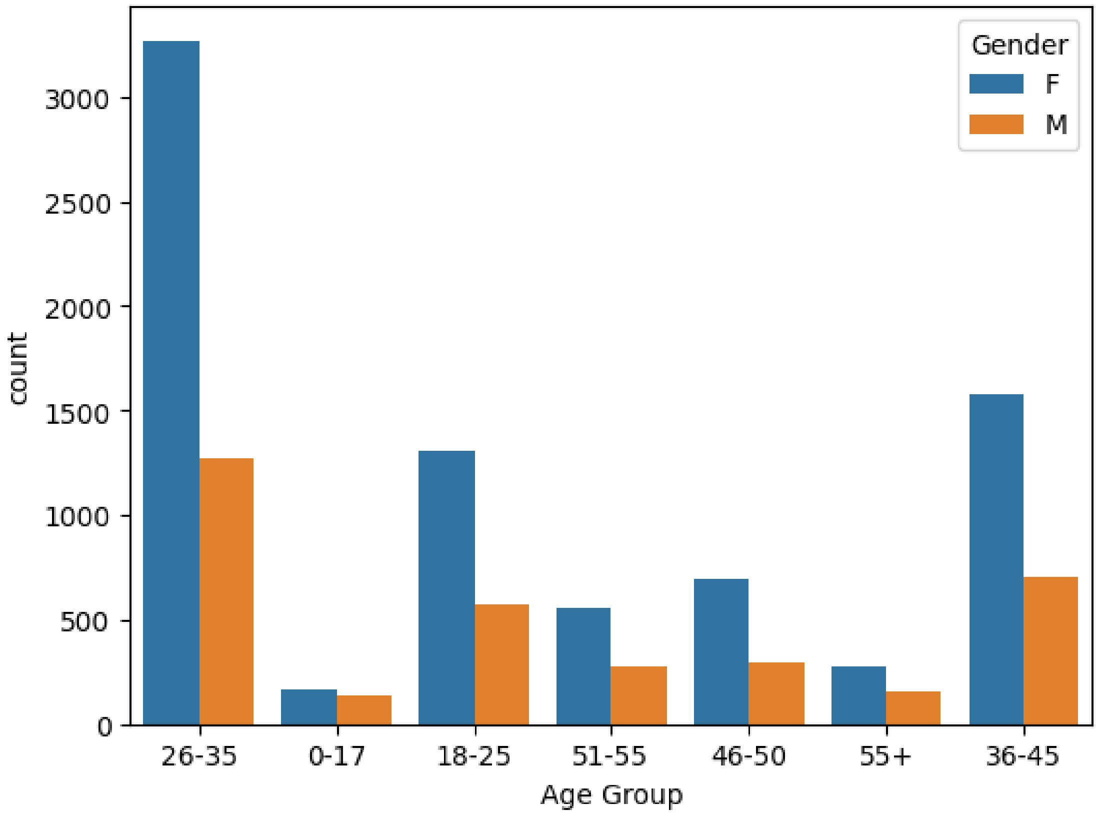


From above piechart we can see that most of the buyers are females and even the purchasing power of the females are greater than men.

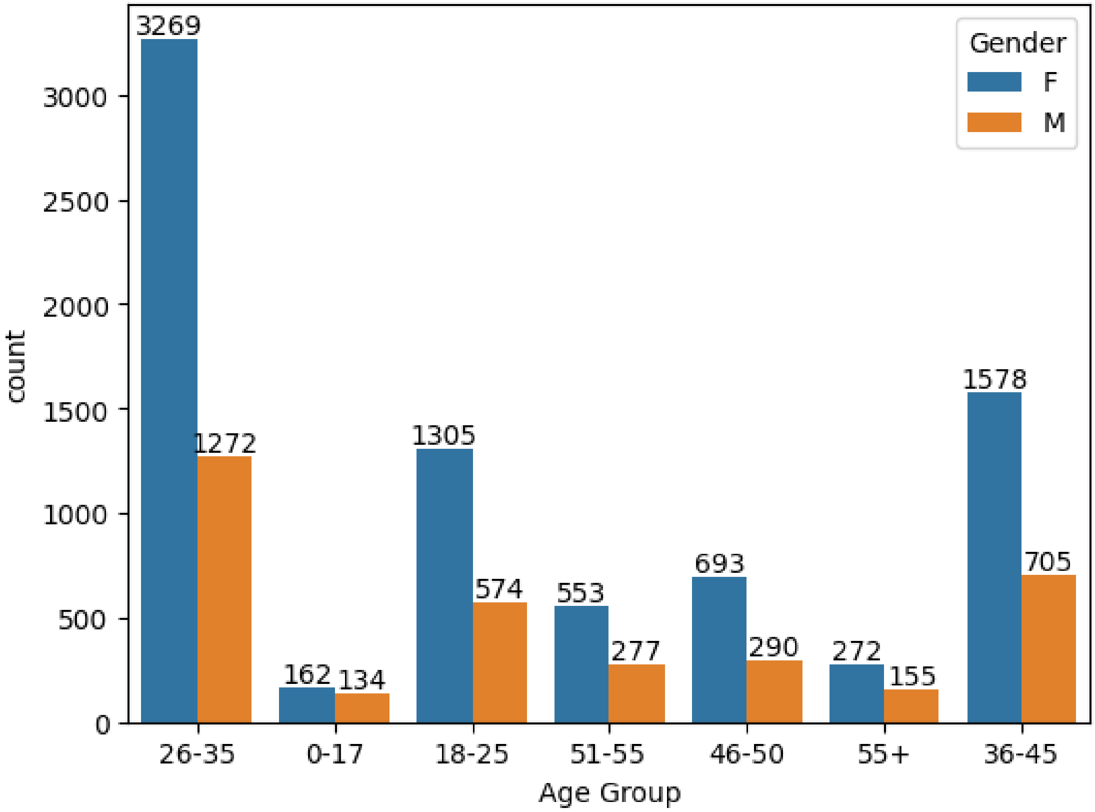
AGE

```
ax=sns.countplot(data=df , x="Age Group",hue="Gender")
```

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



```
ax=sns.countplot(data=df, x="Age Group",hue="Gender")
for bars in ax.containers:
    ax.bar_label(bars)
```



From the above age group we can see that most of the buyers are of age group between 26-35 yrs female

STATE

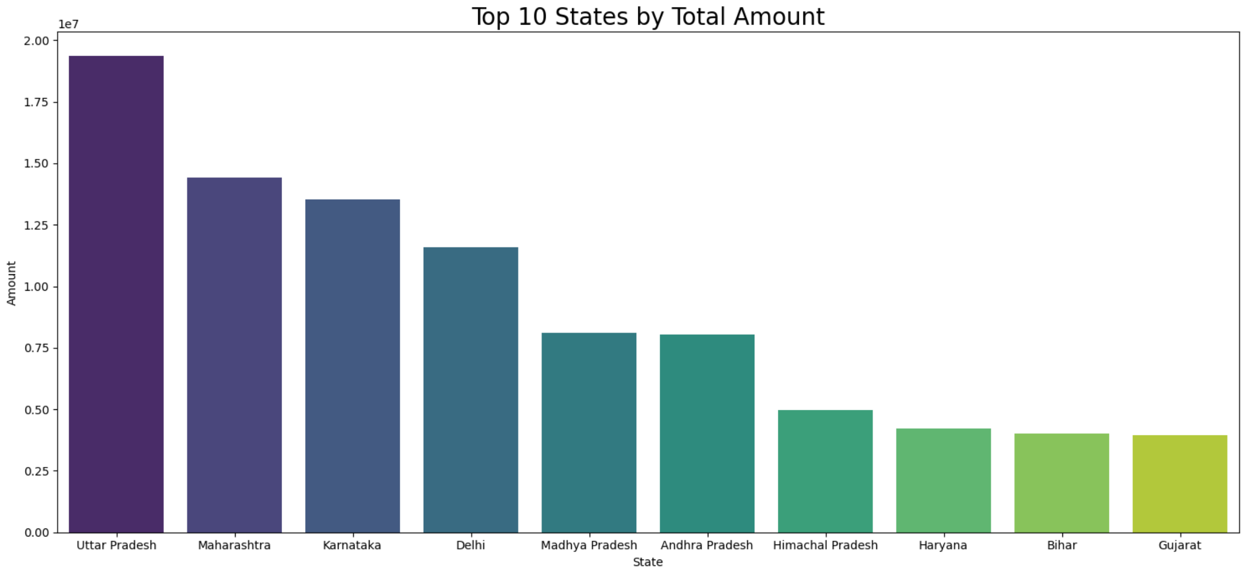
```
df.columns

Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation',
'Product_Category',
       'Orders', 'Amount'],
      dtype='object')

#TOTAL NUMBER OF ORDER IN DESCENDING ORDER BY STATE
ax=df.groupby(["State"] , as_index=False)
["Orders"].sum().sort_values(by="Orders" ,ascending=False).head(10)
ax

               State  Orders
14     Uttar Pradesh    4807
10       Maharashtra    3810
7          Karnataka    3240
2              Delhi    2740
```

```
9     Madhya Pradesh    2252
0     Andhra Pradesh    2051
5   Himachal Pradesh    1568
8             Kerala    1137
4            Haryana    1109
3            Gujarat    1066
```

```
#TOTAL NUMBER OF ORDER IN DESCENDING ORDER BY STATE
ax=df.groupby(["State"] , as_index=False)
["Amount"].sum().sort_values(by="Amount" ,ascending=False).head(10)
ax
```

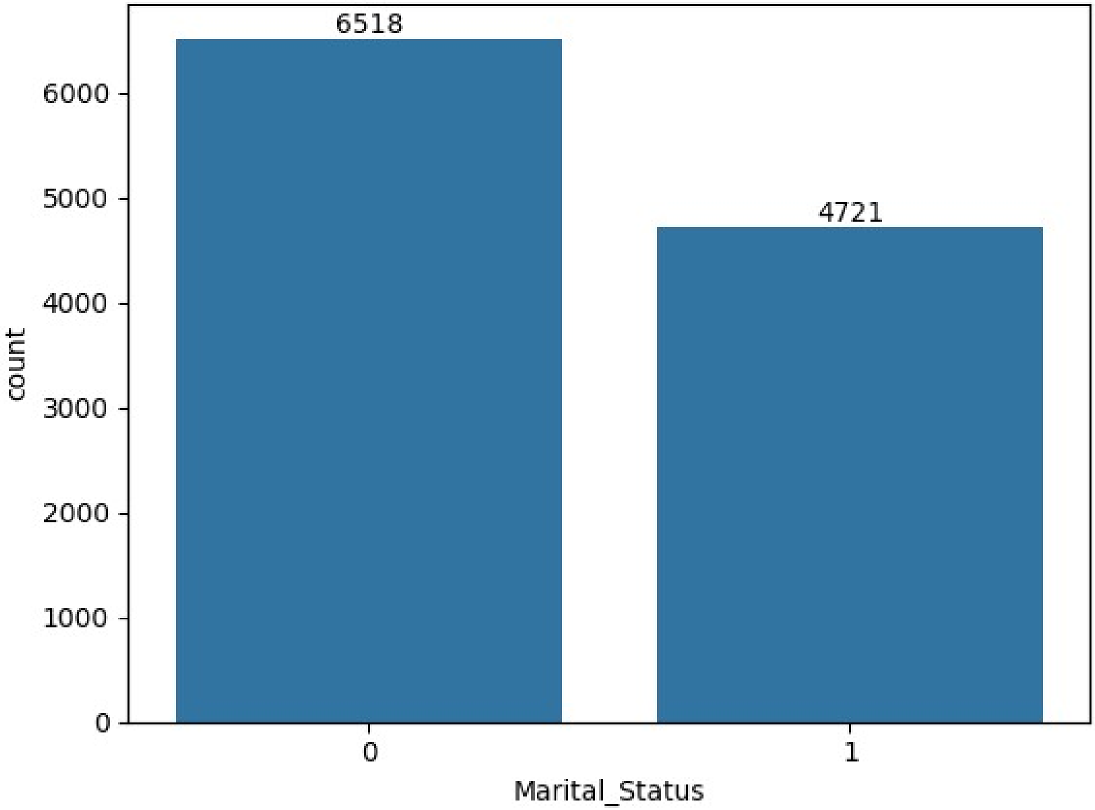
```
               State    Amount
14     Uttar Pradesh  19374968
10       Maharashtra  14427543
7          Karnataka  13523540
2              Delhi  11603818
9     Madhya Pradesh   8101142
0     Andhra Pradesh   8037146
5   Himachal Pradesh   4963368
4            Haryana   4220175
1              Bihar   4022757
3            Gujarat   3946082
```

```
plt.figure(figsize=(15,7))  # Chart ka size

sns.barplot(x="State", y="Amount", data=ax,
palette="viridis",hue="State" )
plt.title("Top 10 States by Total Amount" ,fontsize=20)
plt.xticks(rotation=0)  # State names rotate for better visibility
plt.tight_layout()  # Layout adjust so nothing cuts off
plt.show()
```



Marital Status

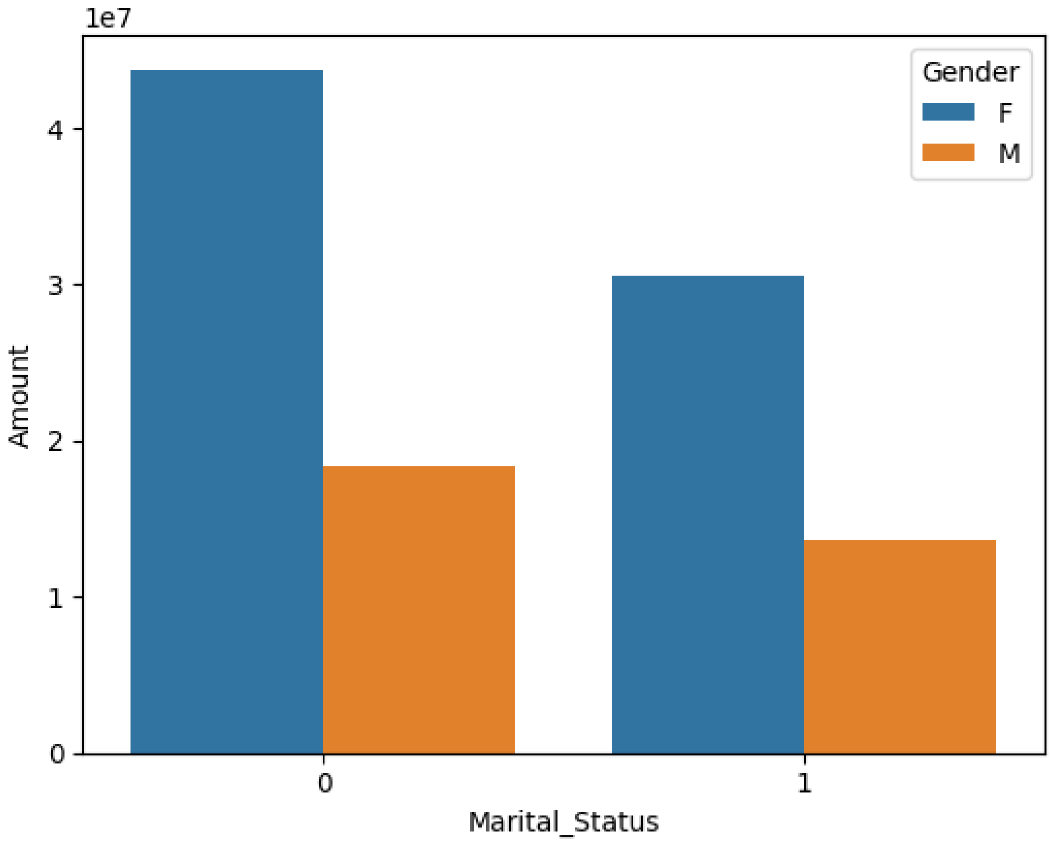
```
ax=sns.countplot(data=df, x="Marital_Status")

for bars in ax.containers:
    ax.bar_label(bars)
```



ON AMOUNT BASIS

```
ax=df.groupby(["Marital_Status","Gender"] , as_index=False)
["Amount"].sum().sort_values(by="Amount" ,ascending=False)
sns.barplot(data=ax ,x="Marital_Status",y="Amount",hue="Gender")
```

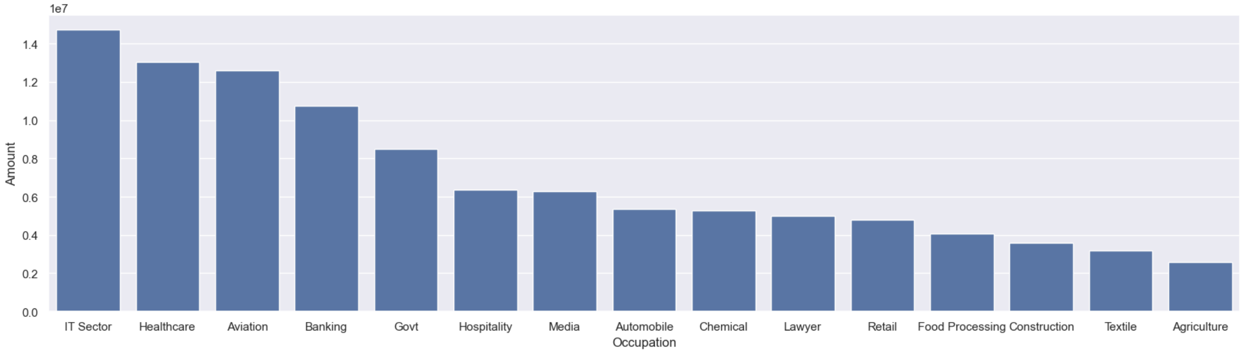
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



From the above graph we can see that married women purchasing power is more than unmarried women

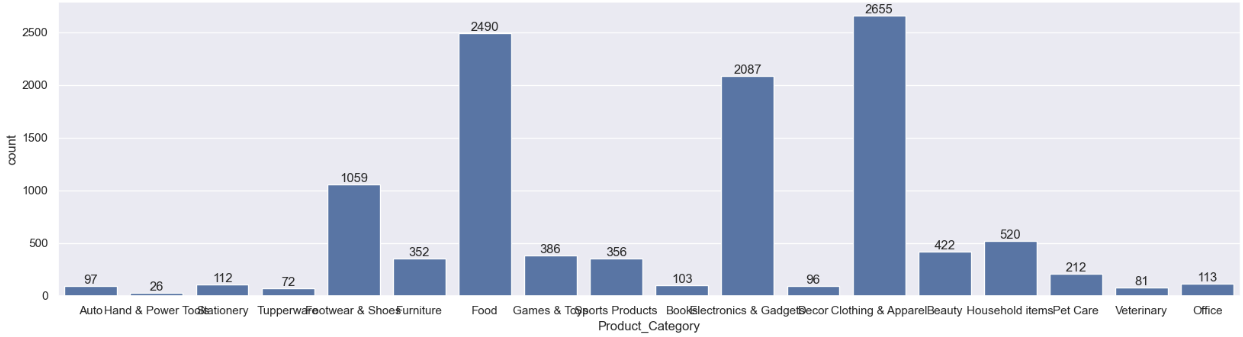
OCCUPATION

```
ax=df.groupby(["Occupation"] , as_index=False)
["Amount"].sum().sort_values(by="Amount" ,ascending=False)
sns.barplot(data=ax ,x="Occupation",y="Amount")
sns.set(rc={"figure.figsize": (20, 5)})
```

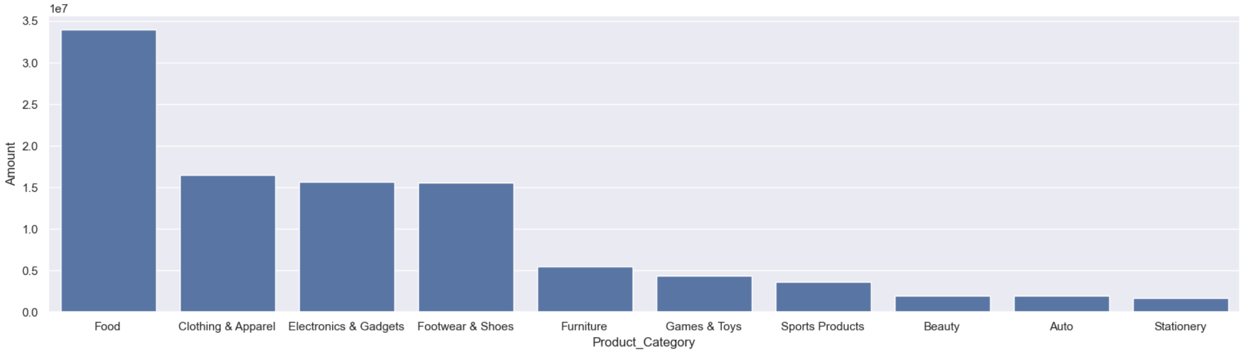


From the above graph we can see that most of the buyers are from IT sector.

```
ax=sns.countplot(data=df, x="Product_Category")

for bars in ax.containers:
    ax.bar_label(bars)
```

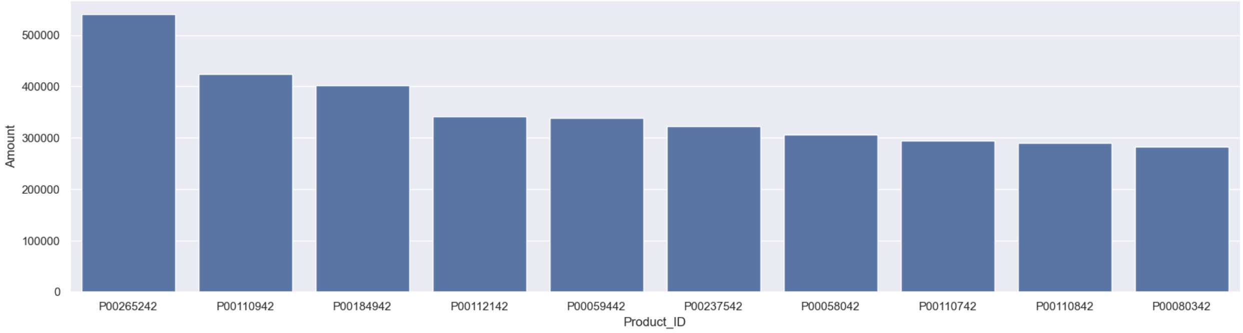


```
ax=df.groupby(["Product_Category"] , as_index=False)
["Amount"].sum().sort_values(by="Amount" ,ascending=False).head(10)
sns.barplot(data=ax ,x="Product_Category",y="Amount")
sns.set(rc={"figure.figsize": (20, 5)})
```



From above graph it's clear that most amount spent on food but most order are of clothings

```
ax=df.groupby(["Product_ID"] , as_index=False)
["Amount"].sum().sort_values(by="Amount" ,ascending=False).head(10)
sns.barplot(data=ax ,x="Product_ID",y="Amount")
sns.set(rc={"figure.figsize": (20, 5)})
```



Conclusion

Married women age group 26-35 years From Uttarpradesh working in IT Sector are more likely buy products from food and clothing