```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings as wrn
wrn.filterwarnings('ignore')
import os

#read dataset
os.chdir("C:\\Users\hp\Downloads\Python_Diwali_Sales_Analysis\
Python_Diwali_Sales_Analysis")
df =pd.read_csv("Diwali Sales Data.csv" , encoding= 'unicode_escape')
df
```

```
       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
...        ...          ...        ...    ...       ...  ...
...
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  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
...        ...     ...       ...
11246    370.0     NaN       NaN
11247    367.0     NaN       NaN
11248    213.0     NaN       NaN
11249    206.0     NaN       NaN
11250    188.0     NaN       NaN

[11251 rows x 15 columns]
```

```
#shape of data
df.shape
```

```
(11251, 15)
```

```
#data information
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
```

```
# describe data
df.describe()
```

```
            User_ID           Age  Marital_Status        Orders
Amount  \
count  1.125100e+04  11251.000000    11251.000000  11251.000000
11239.000000
mean   1.003004e+06     35.421207        0.420318      2.489290
9453.610858
std    1.716125e+03     12.754122        0.493632      1.115047
5222.355869
min    1.000001e+06     12.000000        0.000000      1.000000
188.000000
25%    1.001492e+06     27.000000        0.000000      1.500000
5443.000000
50%    1.003065e+06     33.000000        0.000000      2.000000
8109.000000
75%    1.004430e+06     43.000000        1.000000      3.000000
12675.000000
max    1.006040e+06     92.000000        1.000000      4.000000
23952.000000

       Status  unnamed1
count     0.0       0.0
mean      NaN       NaN
std       NaN       NaN
min       NaN       NaN
25%       NaN       NaN
50%       NaN       NaN
75%       NaN       NaN
max       NaN       NaN
```

```
#checking columns (covert df to python list)
df.columns.tolist()
```

```
['User_ID',
 'Cust_name',
 'Product_ID',
 'Gender',
```

```
 'Age Group',
 'Age',
 'Marital_Status',
 'State',
 'Zone',
 'Occupation',
 'Product_Category',
 'Orders',
 'Amount',
 'Status',
 'unnamed1']
```

```
#check missing values
df.isnull().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
Status              11251
unnamed1            11251
dtype: int64
```

```
#check duplicate values
df.nunique()
```

```
User_ID             3755
Cust_name           1250
Product_ID          2351
Gender                 2
Age Group              7
Age                   81
Marital_Status         2
State                 16
Zone                   5
Occupation            15
Product_Category      18
Orders                 4
Amount              6584
Status                 0
```

```
unnamed1                0
dtype: int64
```

```
#drop unrelated/blank column
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
```

```
# drop null values
df.dropna(inplace=True)
```

```
df.shape
```

```
(11239, 13)
```

```
# change datatype
df['Amount']=df['Amount'].astype('int')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11239 non-null  int64  
 1   Cust_name         11239 non-null  object 
 2   Product_ID        11239 non-null  object 
 3   Gender            11239 non-null  object 
 4   Age Group         11239 non-null  object
```

```
 5   Age                 11239 non-null  int64
 6   Marital_Status      11239 non-null  int64
 7   State               11239 non-null  object
 8   Zone                11239 non-null  object
 9   Occupation          11239 non-null  object
 10  Product_Category    11239 non-null  object
 11  Orders              11239 non-null  int64
 12  Amount              11239 non-null  int32
dtypes: int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

```
# rename column
df.rename(columns={'Zone' : 'Region'})
```

```
       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
...        ...          ...        ...    ...       ...  ...
...
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    Region       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
```

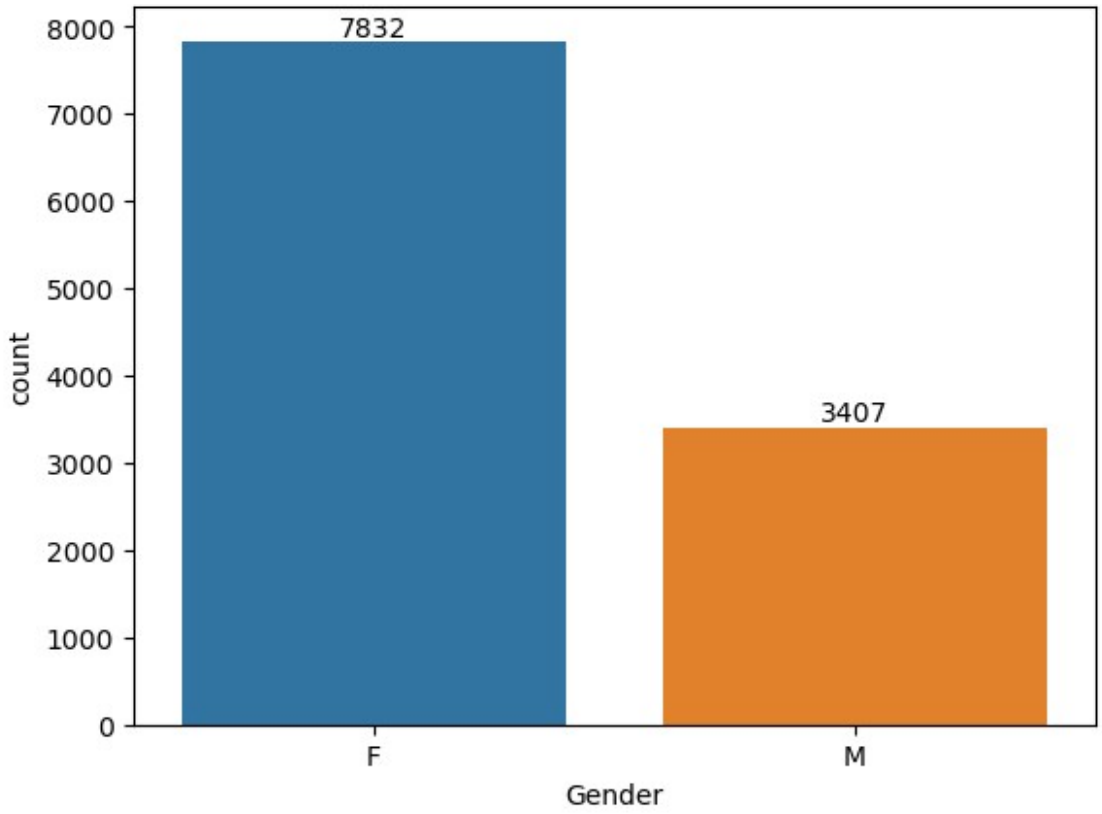
```
2
...               ...       ...               ...               ...
...
11246     Maharashtra   Western         Chemical            Office
4
11247         Haryana  Northern       Healthcare       Veterinary
3
11248  Madhya Pradesh   Central          Textile            Office
4
11249       Karnataka  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]
```

Seaborn

```
df.columns
```

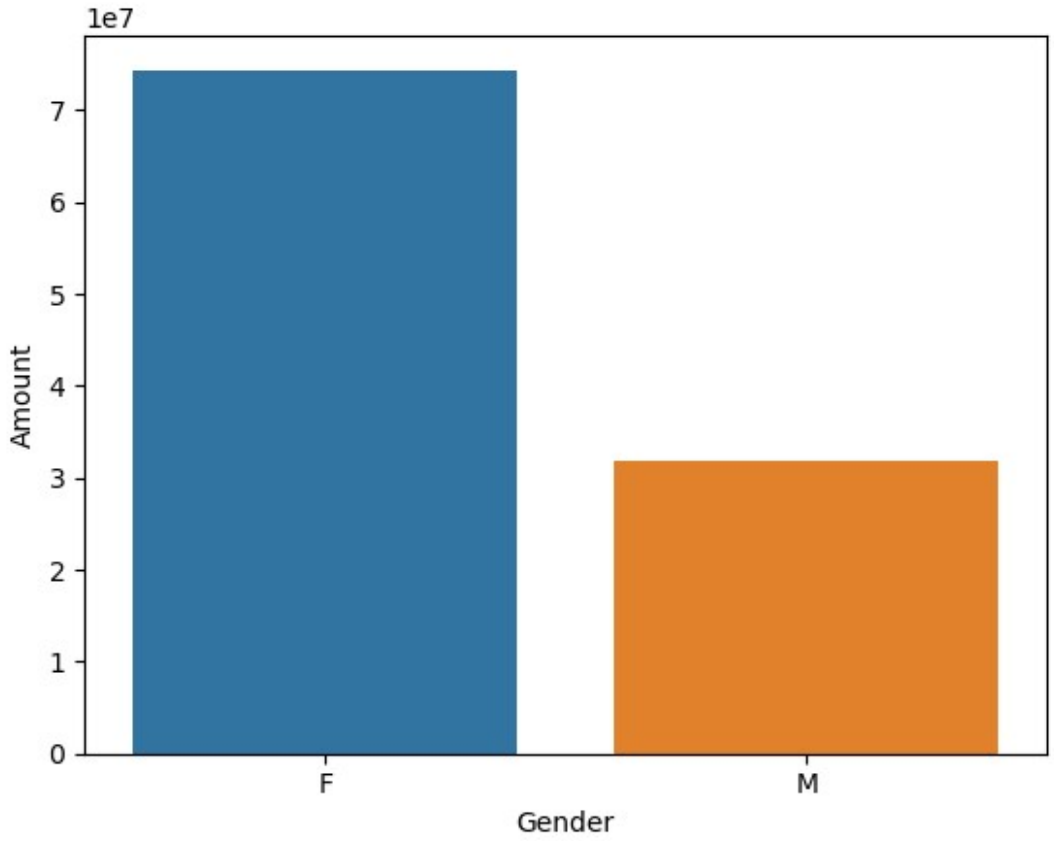
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation',
'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
# count plot syntax :
# sns.countplot(x= name, data = dataframe name)
#sns.countplot(x= 'Gender' ,data=df )
# to add lables
ax= sns.countplot(x= 'Gender' ,data=df )
for bars in ax.containers:
    ax.bar_label(bars)
```




```
sales_by_gender= df.groupby(['Gender'] , as_index= False )
['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x= 'Gender' , y='Amount', data= sales_by_gender)
```

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

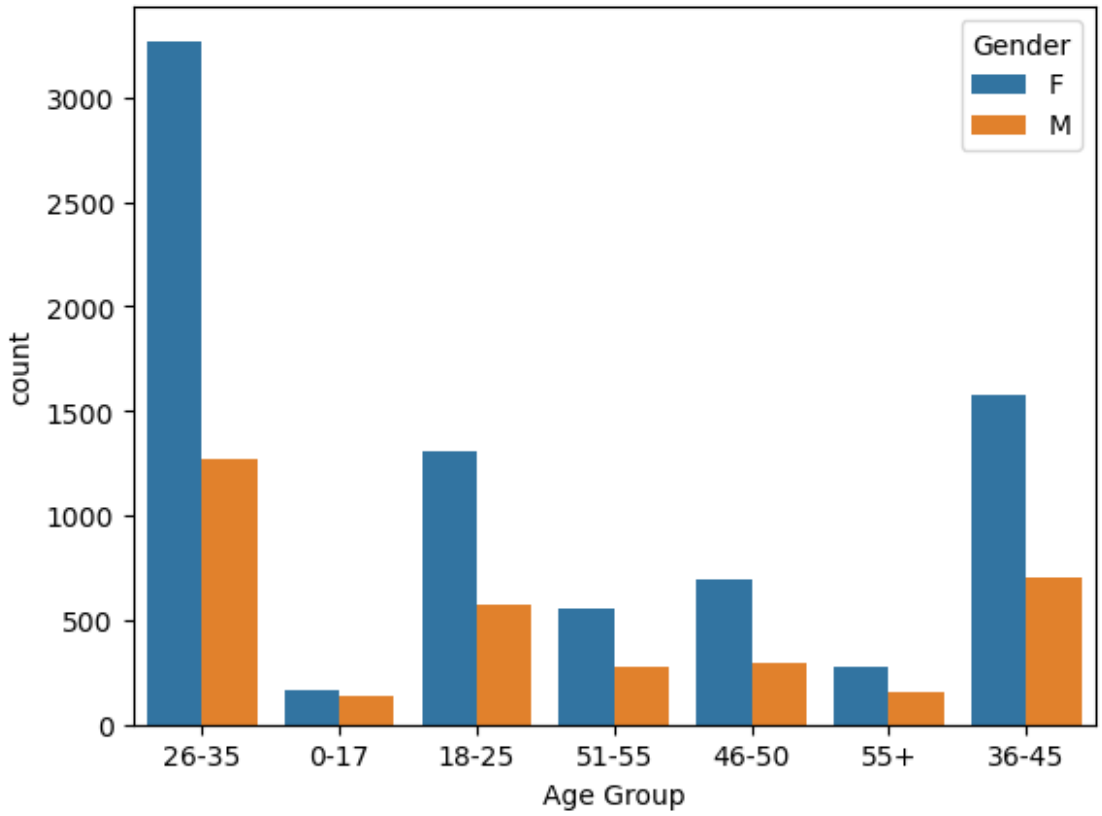


From the above graphs we can see that the most buyers are females and even purchasing power of females are greater than man

sales according to Age groups

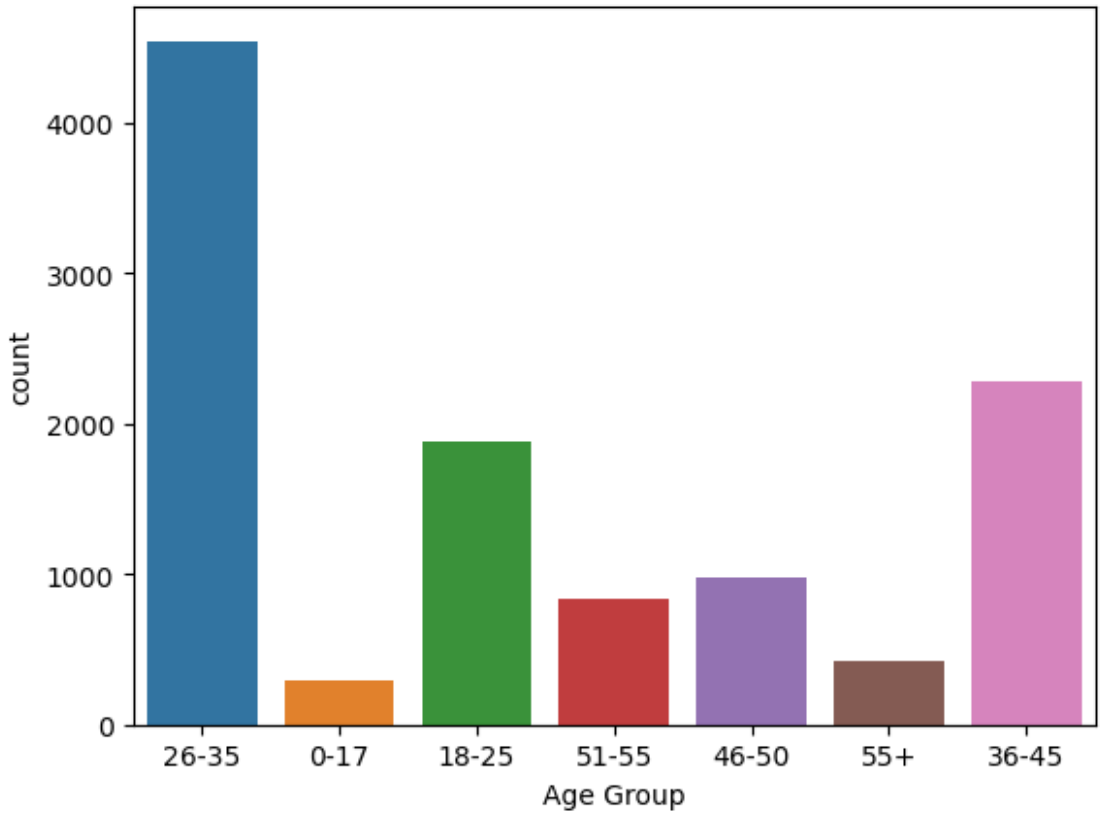
```
sns.countplot(x='Age Group' , data =df , hue= 'Gender')
# hue = according to genders agegroup data will show
```

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




```
# only see which age group shopping most
sns.countplot(x='Age Group', data = df)
```

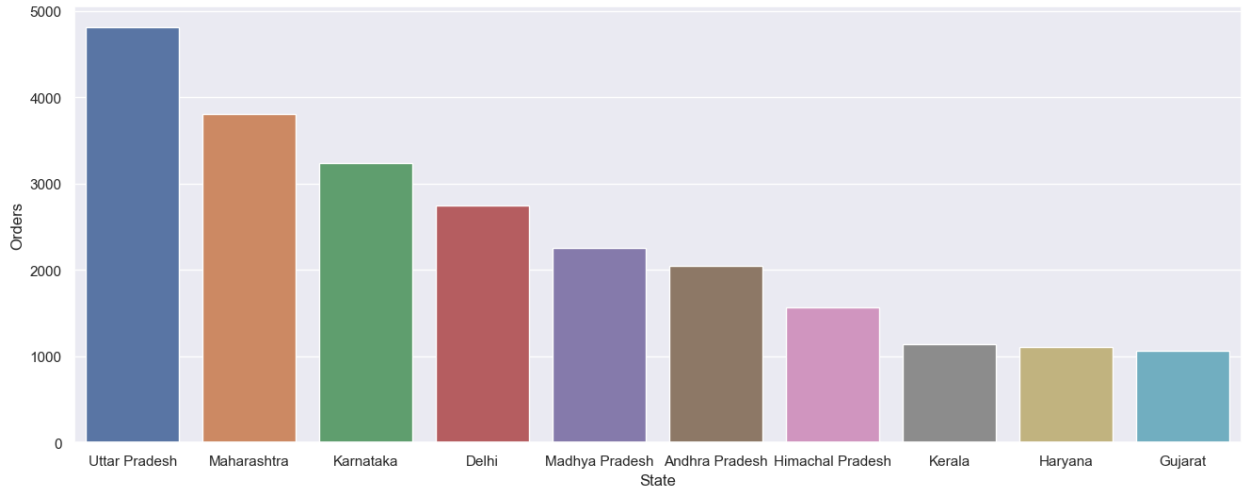
```
<Axes: xlabel='Age Group', ylabel='count'>
```



From the above graph we can see that the people of age group between 26-35 purchases more

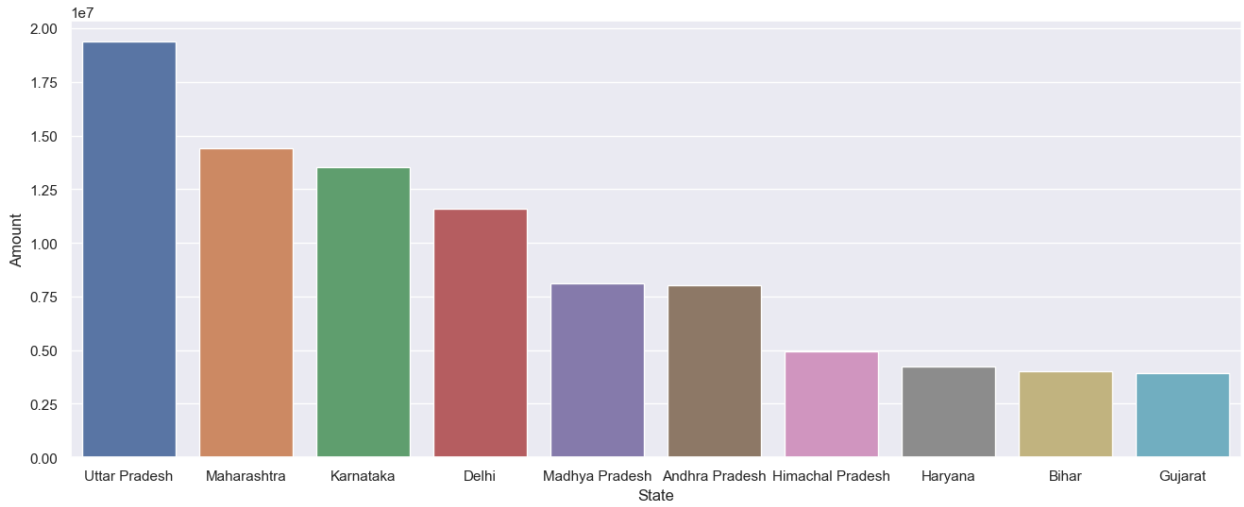
```
# According to state wise see which state which state purchases more
(.head())= no. of state which u want to see
sales_state= df.groupby(['State'] , as_index= False )
['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)

sns.set(rc={'figure.figsize':(16,6)})

sns.barplot(data=sales_state , x = 'State' , y = 'Orders')
```

```
<Axes: xlabel='State', ylabel='Orders'>
```




```
# Total amount/sales from top 10 states
sales_state= df.groupby(['State'] , as_index= False )
['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(16,6)})

sns.barplot(data=sales_state , x = 'State' , y = 'Amount')
```

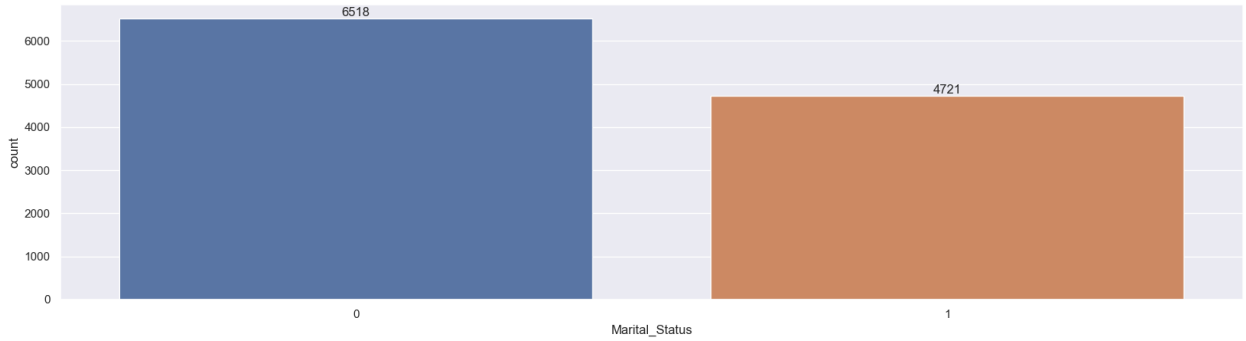
```
<Axes: xlabel='State', ylabel='Amount'>
```



From the above graphs we can see that the most orders are place in Uttar Pradesh , Maharashtra , Karnatka respectively

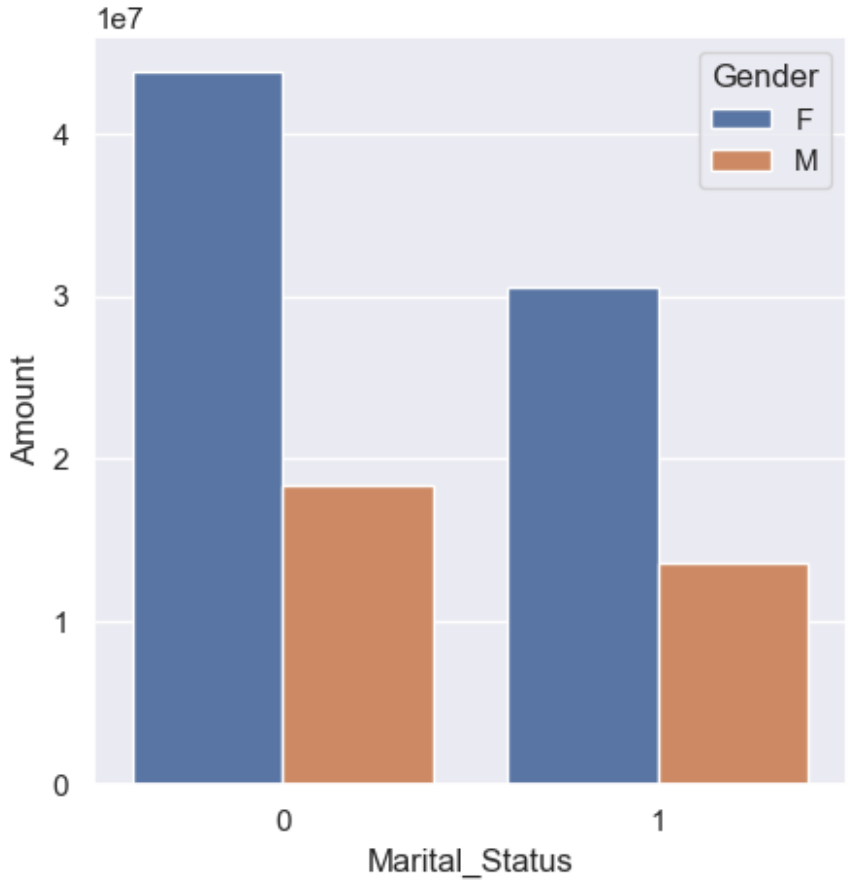
Marital Status

```
ax= sns.countplot(data= df , x= 'Marital_Status')
sns.set(rc={'figure.figsize':(10,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```




```
# according to amount
sales_state= df.groupby(['Marital_Status' , 'Gender'] , as_index=
False )
['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(5,5)})

sns.barplot(data=sales_state , x = 'Marital_Status' , y = 'Amount' ,
hue= 'Gender')
```

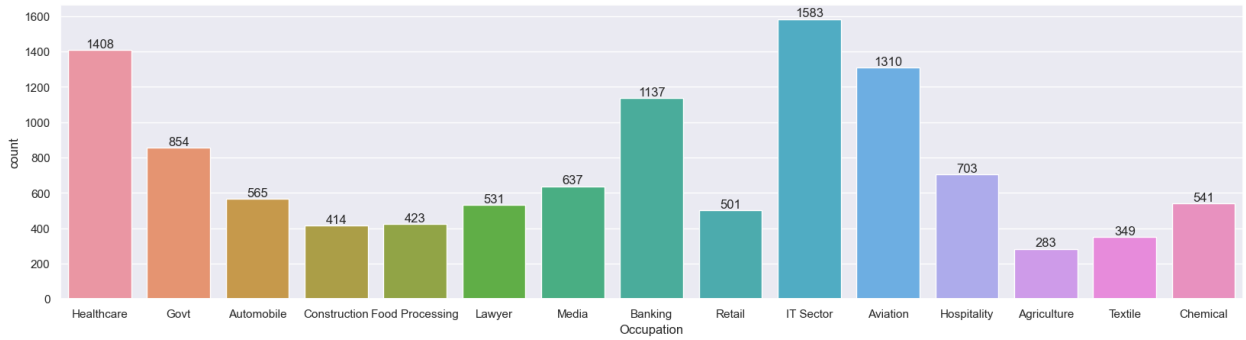
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



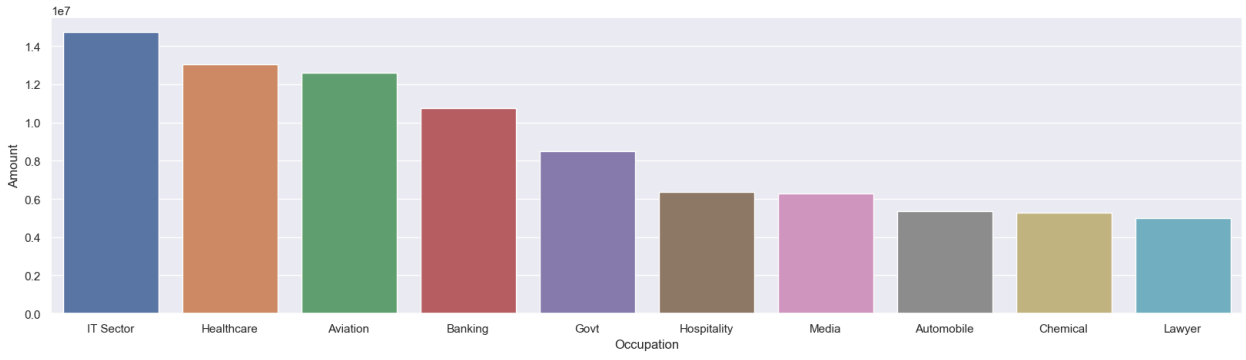
From the above graph we can see that the most of the buyers are married(women) also the purchasing power is also high of women

Occupation

```
sns.set(rc={'figure.figsize':(20,5)})
ax= sns.countplot(data= df , x= 'Occupation')
for bars in ax.containers:
    ax.bar_label(bars)
```




```
# According to Amount
sales=df.groupby(['Occupation'], as_index= False)
['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})

sns.barplot(data=sales , x = 'Occupation' , y = 'Amount')


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

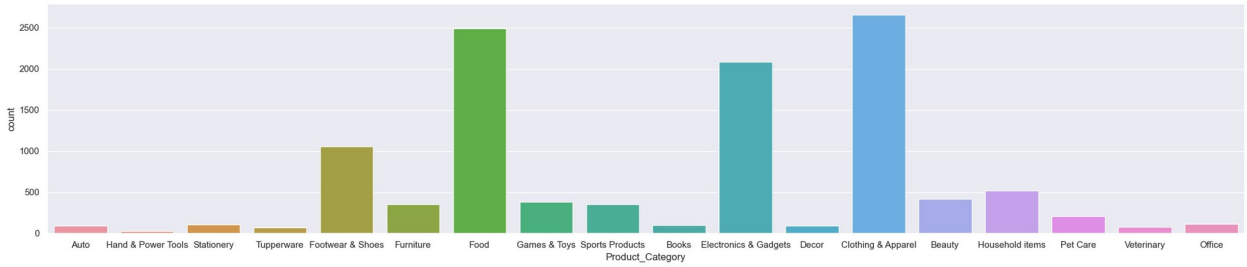


From the above graphs we can see that most buyers are from IT sector,Healthcare and Aviation

Product Category

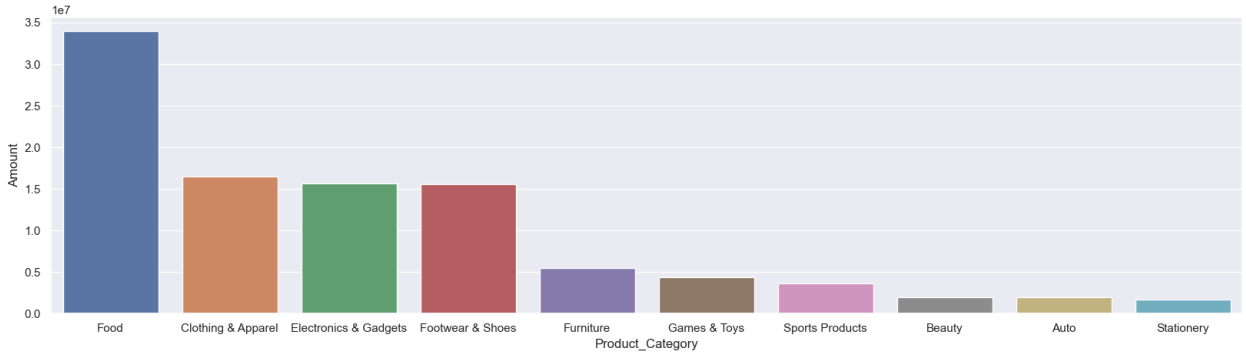
```
sns.set(rc={'figure.figsize':(26,5)})
a= sns.countplot(data= df, x= 'Product_Category')
```

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




```
prod_cat= df.groupby(['Product_Category'],as_index= False)
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
#sales=df.groupby(['Occupation'], as_index= False)
['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})

sns.barplot(data=prod_cat, x = 'Product_Category' , y = 'Amount')
```

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



From above graphs we can see that the most buy products are Food, Clothing and Electronic

Conclusion

> Married women from the age group between 26-35 from Uttarpradesh , Maharashtra, karnatka from occupation IT sector,Healthcare and Aviation purschases item Food , clothing items amd electromnics.