```
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns

#import CSV file
df = pd.read_csv('C:/Users/DELL/Downloads/Data
Science/Python_Diwali_Sales_Analysis/Diwali Sales Data.csv', encoding
= 'unicode_escape')
#to avoid encoding error, use 'unicode_escape'

# to find no. of rows and columns in dataFrame
df.shape
```

```
(11251, 15)
```

```
# to fetch the top 10 data from starting
df.head(10)
```

```
   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

5  1000588       Joni  P00057942      M     26-35   28               1

6  1001132       Balk  P00018042      F     18-25   25               1

7  1002092   Shivangi  P00273442      F       55+   61               0

8  1003224     Kushal  P00205642      M     26-35   35               0

9  1003650      Ginny  P00031142      F     26-35   26               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
5  Himachal Pradesh  Northern  Food Processing             Auto
1
6     Uttar Pradesh   Central           Lawyer             Auto
4
7       Maharashtra   Western        IT Sector             Auto
1
8     Uttar Pradesh   Central             Govt             Auto
2
9    Andhra Pradesh  Southern            Media             Auto
4

     Amount  Status  unnamed1
0  23952.00     NaN       NaN
1  23934.00     NaN       NaN
2  23924.00     NaN       NaN
3  23912.00     NaN       NaN
4  23877.00     NaN       NaN
5  23877.00     NaN       NaN
6  23841.00     NaN       NaN
7       NaN     NaN       NaN
8  23809.00     NaN       NaN
9  23799.99     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
```

```
#drop unrelated/blank columns from DataFrame
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
```

```
# to check null value in dF, if its true -> null value is available,
if its false -> No null value
pd.isnull(df)
```

```
# check null value of all coulumns
pd.isnull(df).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
```

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

#change the 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')
```

```
# describe() method returns description of the data in the dataFrame (
i.e. count, mean, std, etc)
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
```

```
# use describe() for specific columns
df[['Age','Orders','Amount']].describe()
```

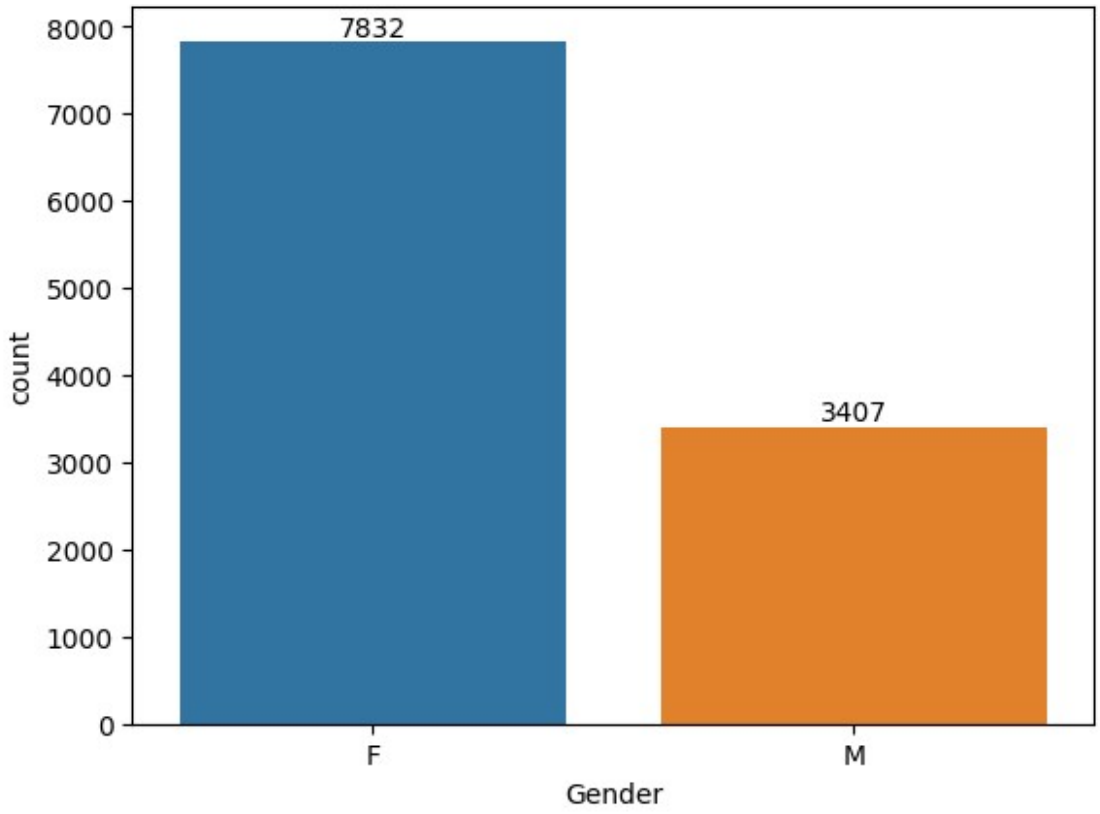
```
                Age        Orders        Amount
count  11239.000000  11239.000000  11239.000000
mean      35.410357      2.489634   9453.610553
std       12.753866      1.114967   5222.355168
min       12.000000      1.000000    188.000000
25%       27.000000      2.000000   5443.000000
```

```
50%       33.000000      2.000000   8109.000000
75%       43.000000      3.000000  12675.000000
max       92.000000      4.000000  23952.000000
```

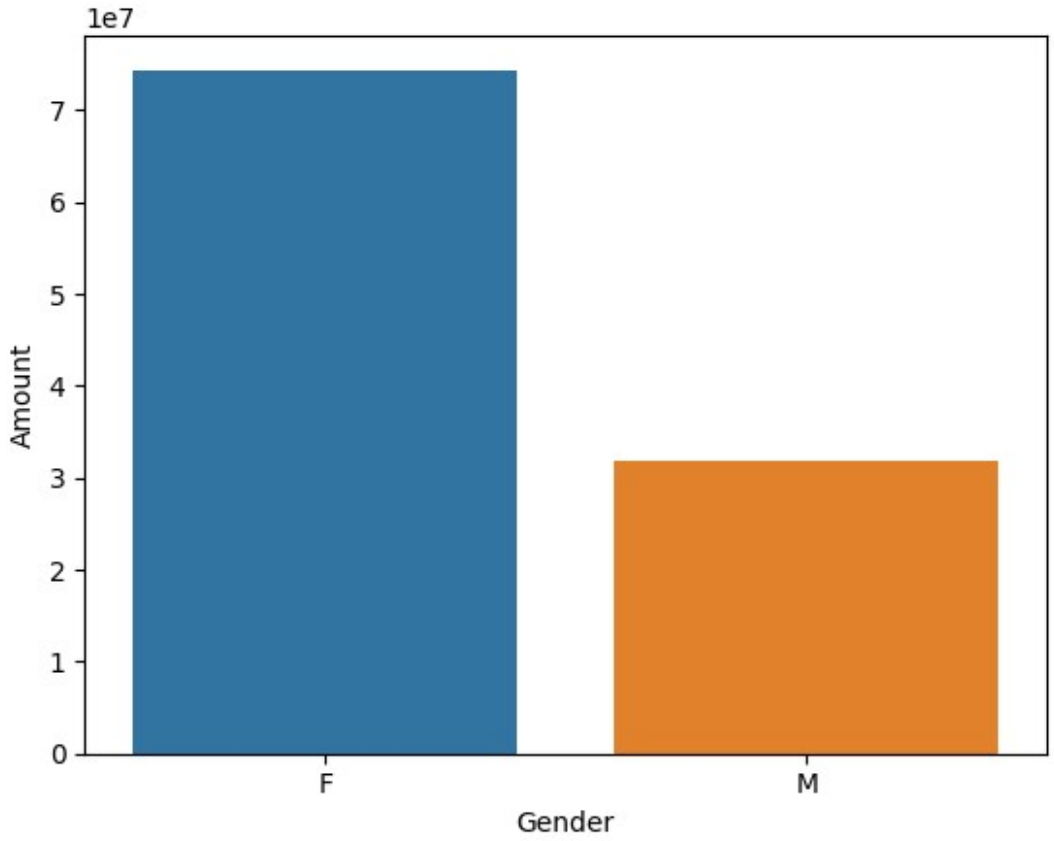
Exploratory Data Analysis

Gender

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



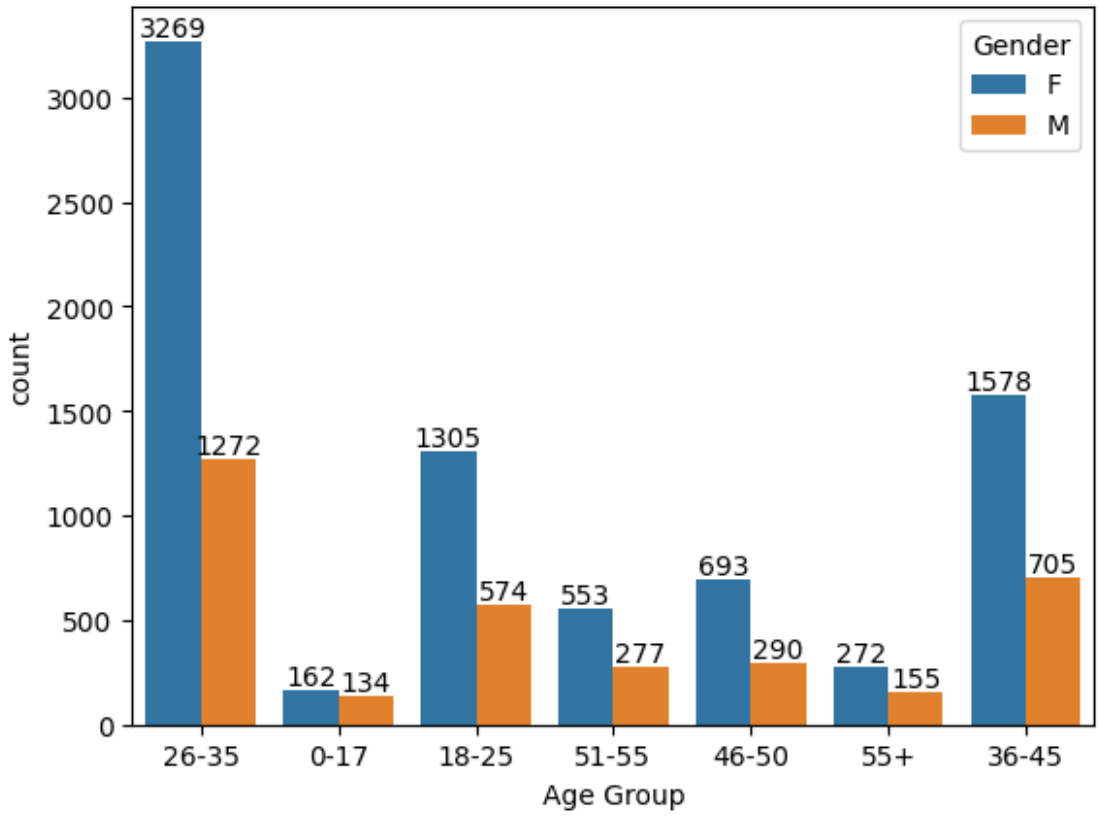

```
sales_catg = df.groupby(['Gender'], as_index = False)
['Amount'].sum().sort_values(by='Amount', ascending = False)
sns.barplot(x='Gender', y='Amount', data = sales_catg,hue='Gender')
plt.show()
```



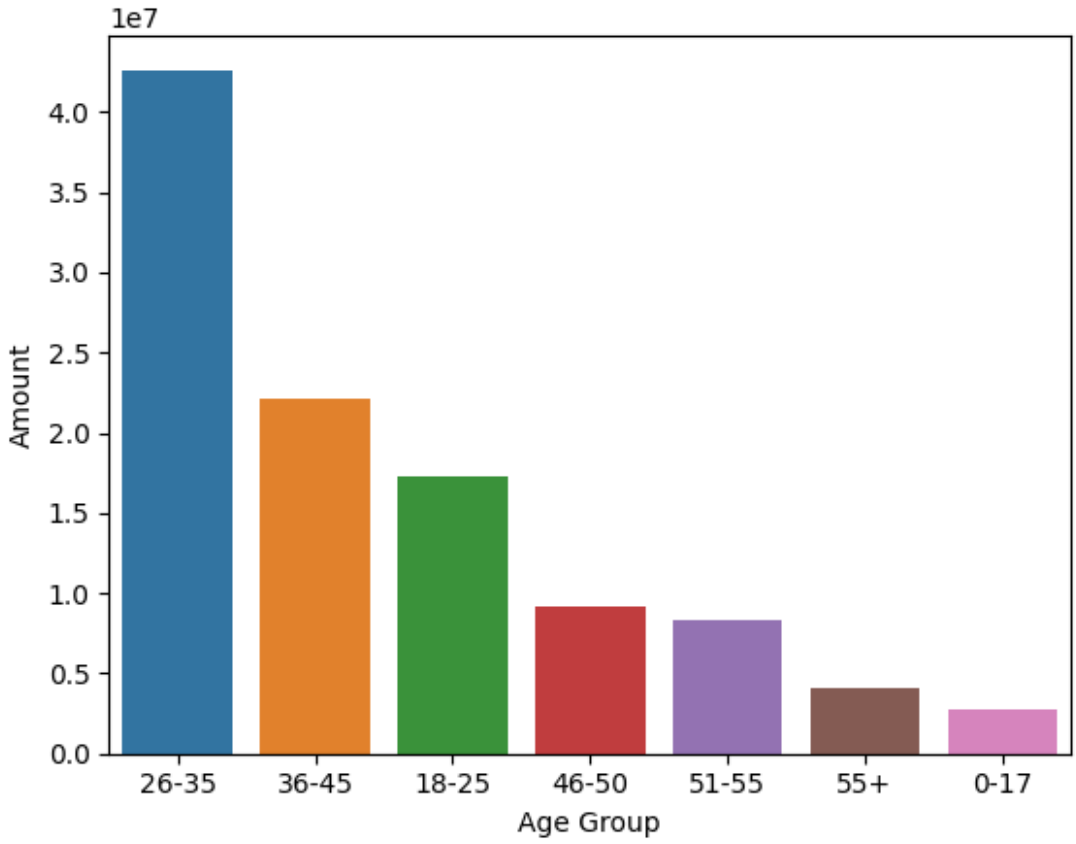
From the above graphs we can see that most of the buyers are females and even the purchasing power of females is greater than men

Age Group

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



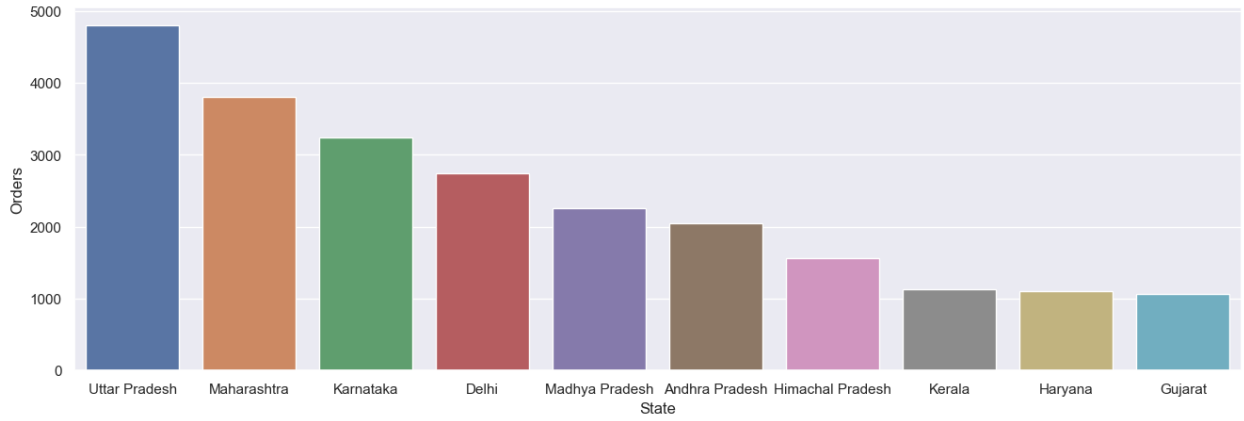

```
# Total amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index = False)
['Amount'].sum().sort_values(by='Amount', ascending = False)
sns.barplot(x='Age Group', y='Amount', data = sales_age, hue='Age
Group')
plt.show()
```



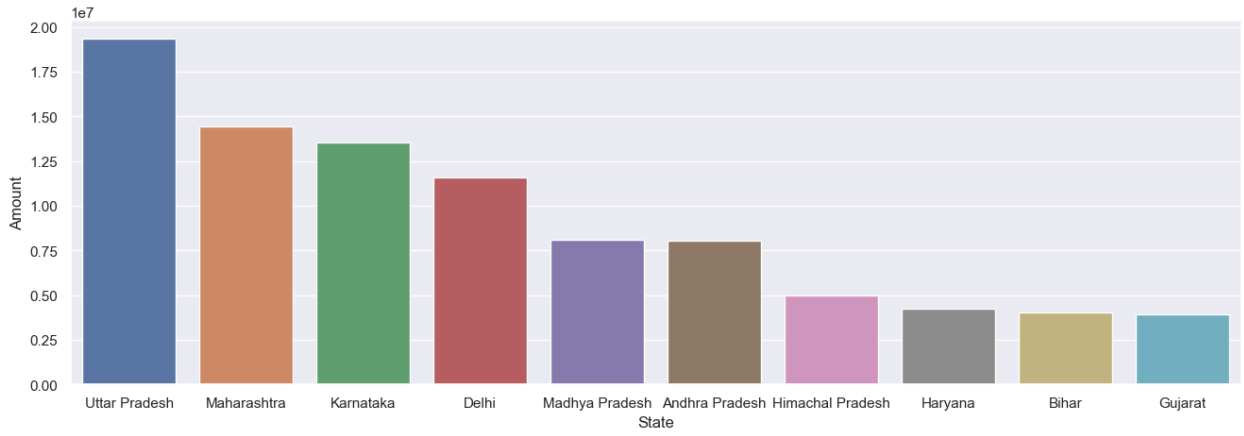
From the above graphs we can see that most of the buyers are of age group between 26-35 years female

State

```
# total number of orders from top 10 states

sales_state = df.groupby(['State'],as_index = False)
['Orders'].sum().sort_values(by='Orders', ascending = False).head(10)
sns.set(rc={'figure.figsize':(16,5)})
sns.barplot(x='State',y='Orders',data = sales_state,hue='State')
plt.show()
```



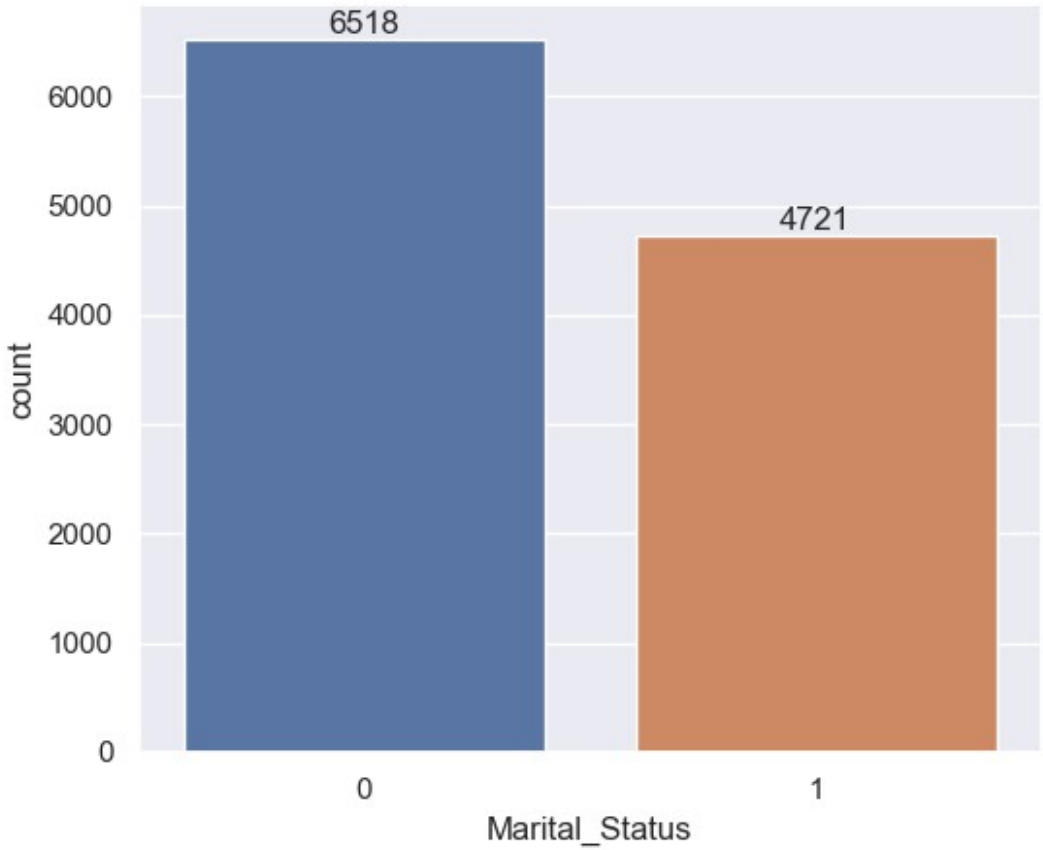

```
# total amount/saless from top 10 states

sales_state = df.groupby(['State'],as_index = False)
['Amount'].sum().sort_values(by='Amount', ascending = False).head(10)
sns.set(rc={'figure.figsize':(16,5)})
sns.barplot(x='State',y='Amount',data = sales_state, hue='State')
plt.show()
```



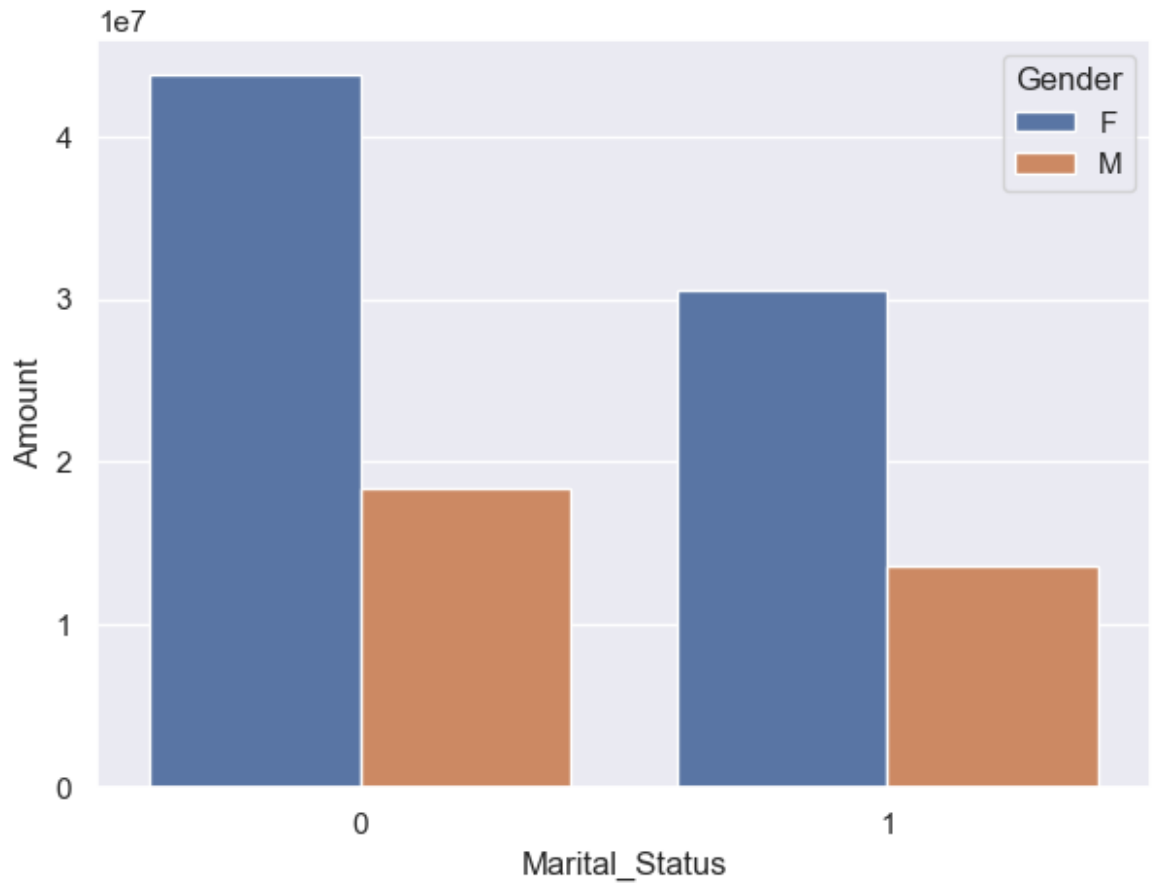
From above we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

Marital Status

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



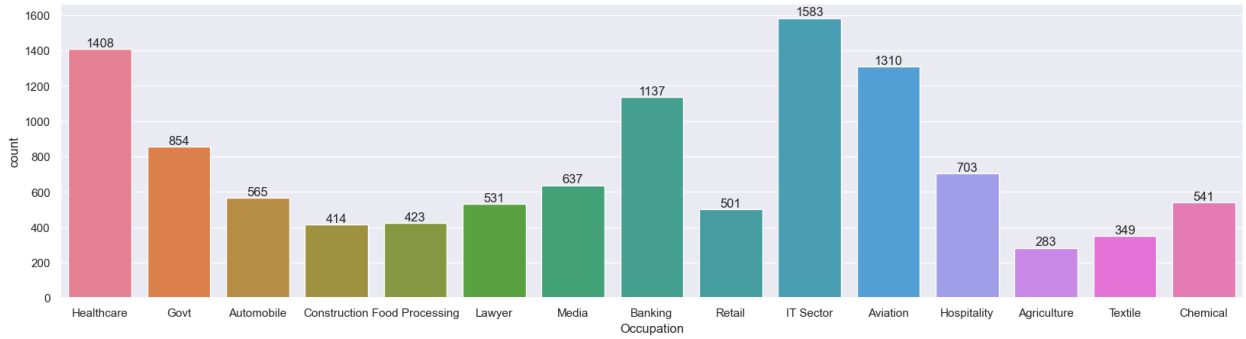

```
sales_state = df.groupby(['Marital_Status', 'Gender'],as_index =
False)['Amount'].sum().sort_values(by='Amount', ascending = False)
sns.set(rc={'figure.figsize':(7,5)})
sns.barplot(x='Marital_Status',y='Amount',data = sales_state,
hue='Gender')
plt.show()
```



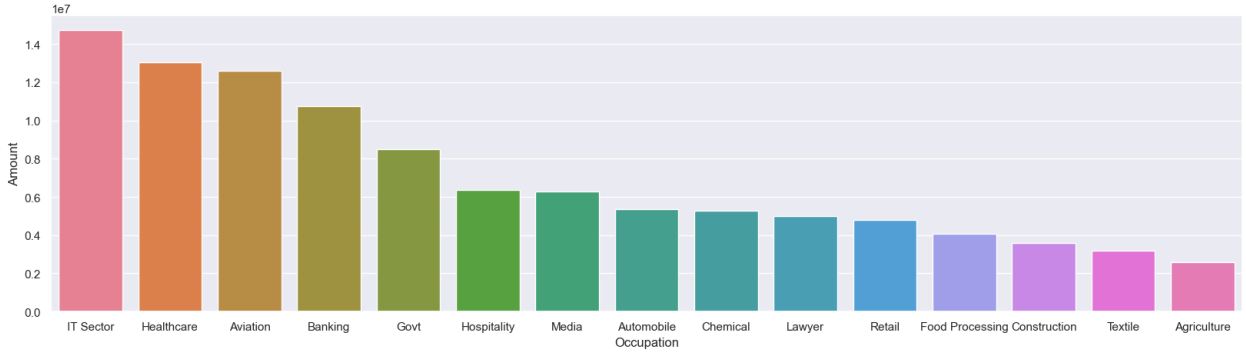
From the above graphs we can see that most of the buyers are married(women) and they have high purchasing power

Occupation

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



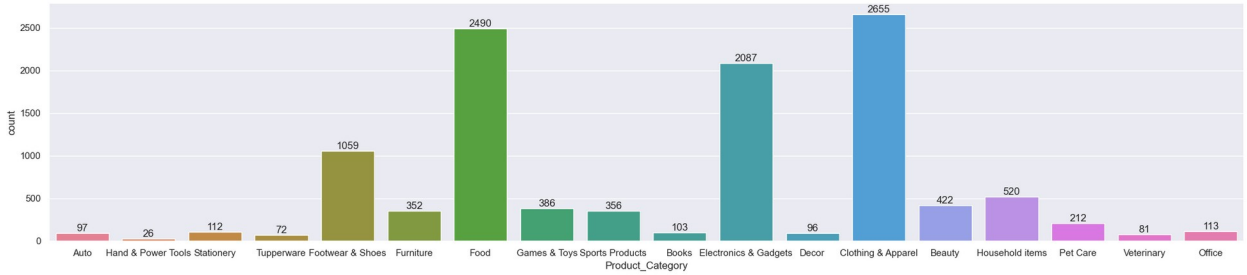
```
sales_sector = df.groupby(['Occupation'],as_index = False)
['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_sector,x='Occupation',y='Amount',hue='Occupatio
n')
plt.show()
```



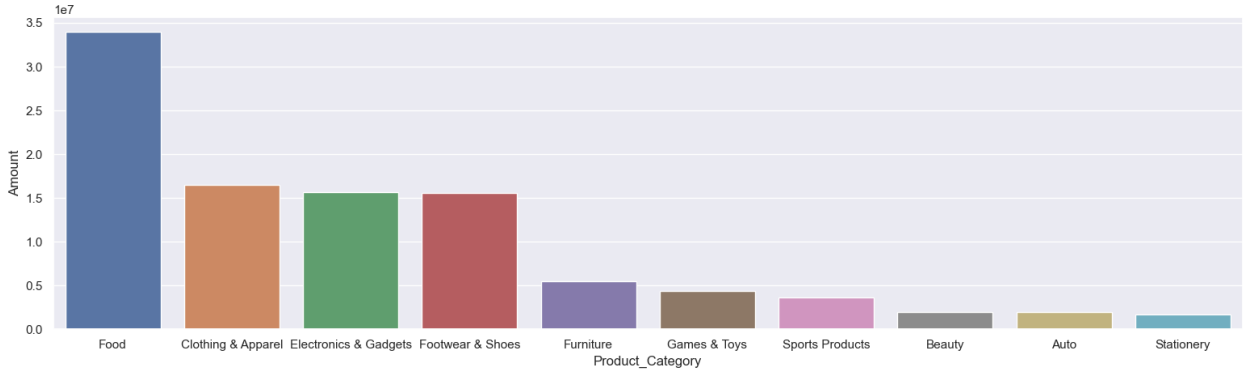
From above graphs we can see that most of the buyers are from IT, Aviation and Healthcare sector

Product Category

```
sns.set(rc={'figure.figsize':(25,5)})
ax = sns.countplot(data = df, x = 'Product_Category',
hue='Product_Category')

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

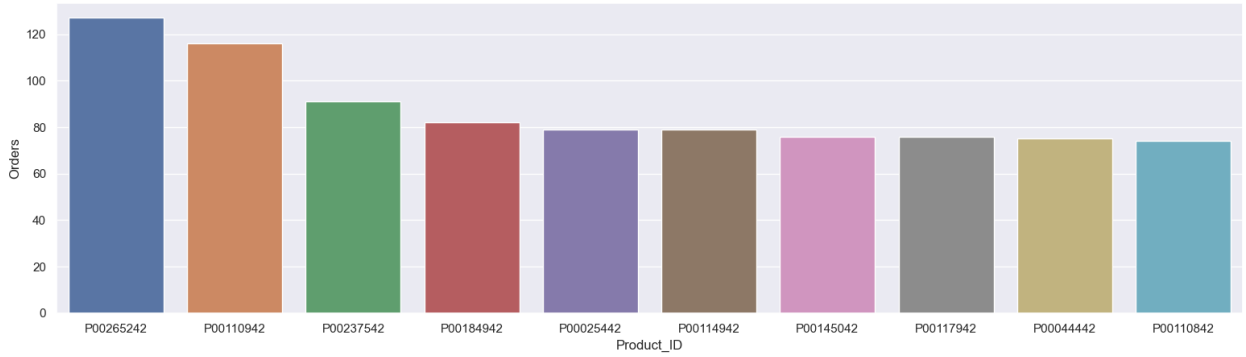



```
sales_prod = df.groupby(['Product_Category'],as_index = False)
['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)
sns.set(rc={'figure.figsize':(19,5)})
sns.barplot(data=sales_prod,x='Product_Category',y='Amount',hue='Produ
ct_Category')
plt.show()
```

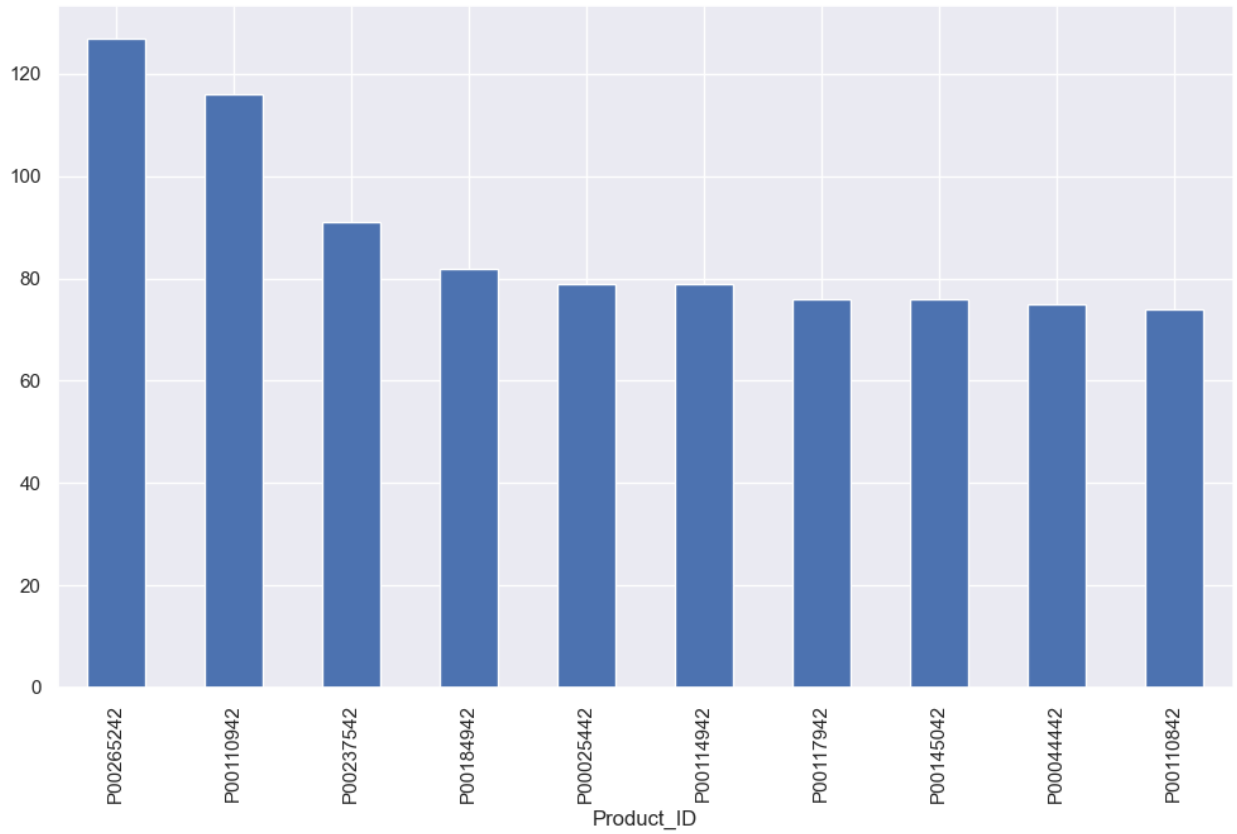


From the above graphs we can see that most of the sold products are from Food, Clothing and Electronics Category

```
# top 10 most sold products
sales_prod = df.groupby(['Product_ID'],as_index = False)
['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)
sns.set(rc={'figure.figsize':(19,5)})
sns.barplot(data=sales_prod,x='Product_ID',y='Orders',hue='Product_ID'
)
plt.show()
```




```
fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')
['Orders'].sum().nlargest(10).sort_values(ascending =
False).plot(kind='bar')
plt.show()
```



Conclusion :

Married women age group 26-35 years from UP, Maharashtra and Karnataka working in IT, Healthcare and Aviation are more likely buy products from Food, Clothing and Electronics category