Importing Libraries

In [1]:

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

In [2]:

```
df=pd.read_csv("Diwali Sales Data.csv", encoding= 'unicode_escape')
```

Getting number of Rows and Columns

In [3]:

```
df.shape
```

Out[3]:

```
(11251, 15)
```

In [4]:

```
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 Blank Columns

In [5]:

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

In [6]:

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

Out[6]:

```
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 Null Values

In [7]:

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

In [8]:

```
#changing 'Amount' data type to integer
df['Amount']=df['Amount'].astype('int')
```

In [9]:

```
df['Amount'].dtypes
```

Out[9]:

```
dtype('int32')
```

In [10]:

```
#Column Details
df.columns
```

Out[10]:

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

In [11]:

```
df.head()
```

Out[11]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western

In [12]:

```
df.tail()
```

Out[12]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zon
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Wester
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Norther
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Centr
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Souther
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Wester

Viewing Statistical information of Numeric Data

In [13]:

```
df.describe()
```

Out[13]:

	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

In [14]:

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

Out[14]:

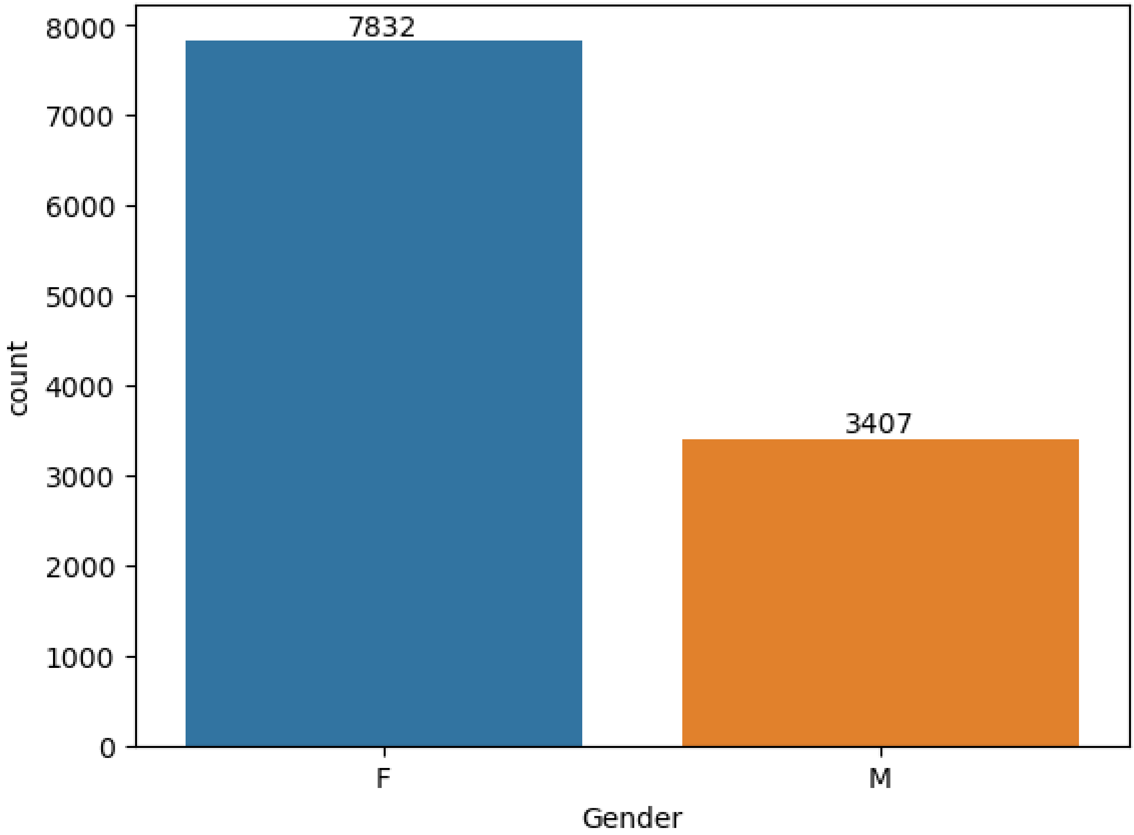
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis (EDA)

Gender

In [15]:

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

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

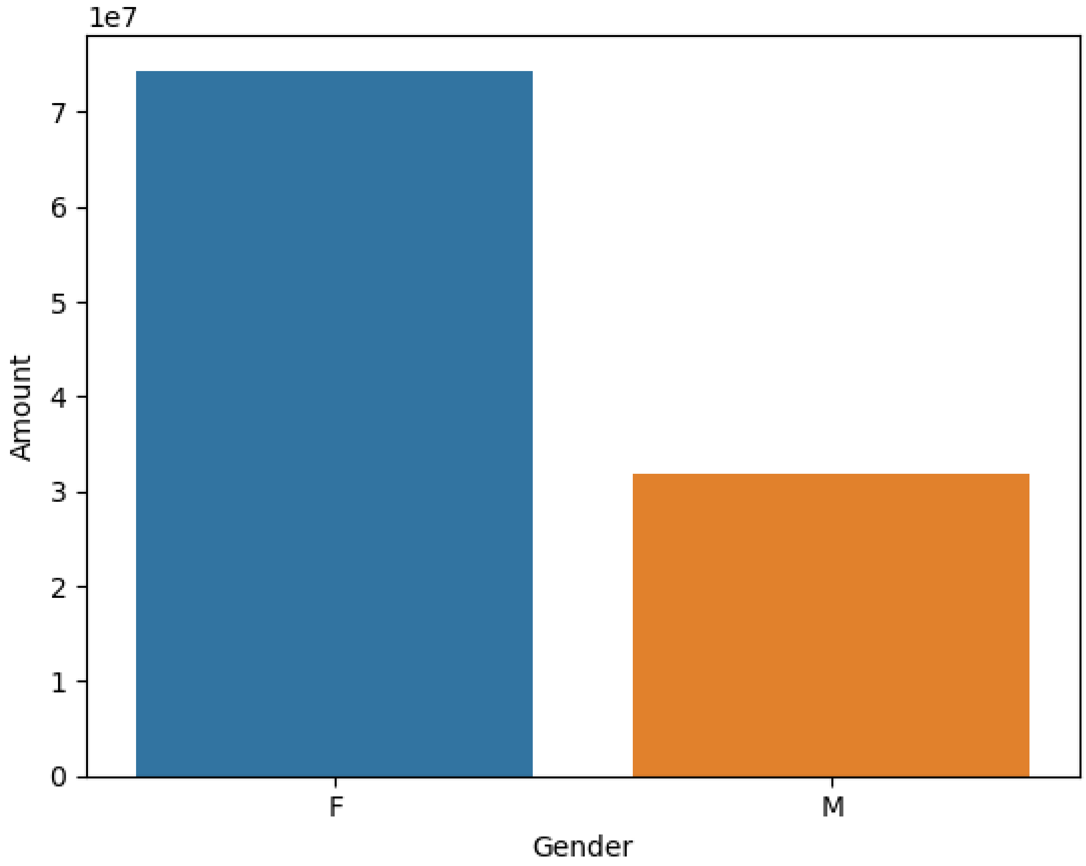


In [16]:

```
#Bar chart for gender vs total amount
sales_gender=df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(

sns.barplot(x='Gender',y='Amount',data=sales_gender)
```

Out[16]:

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

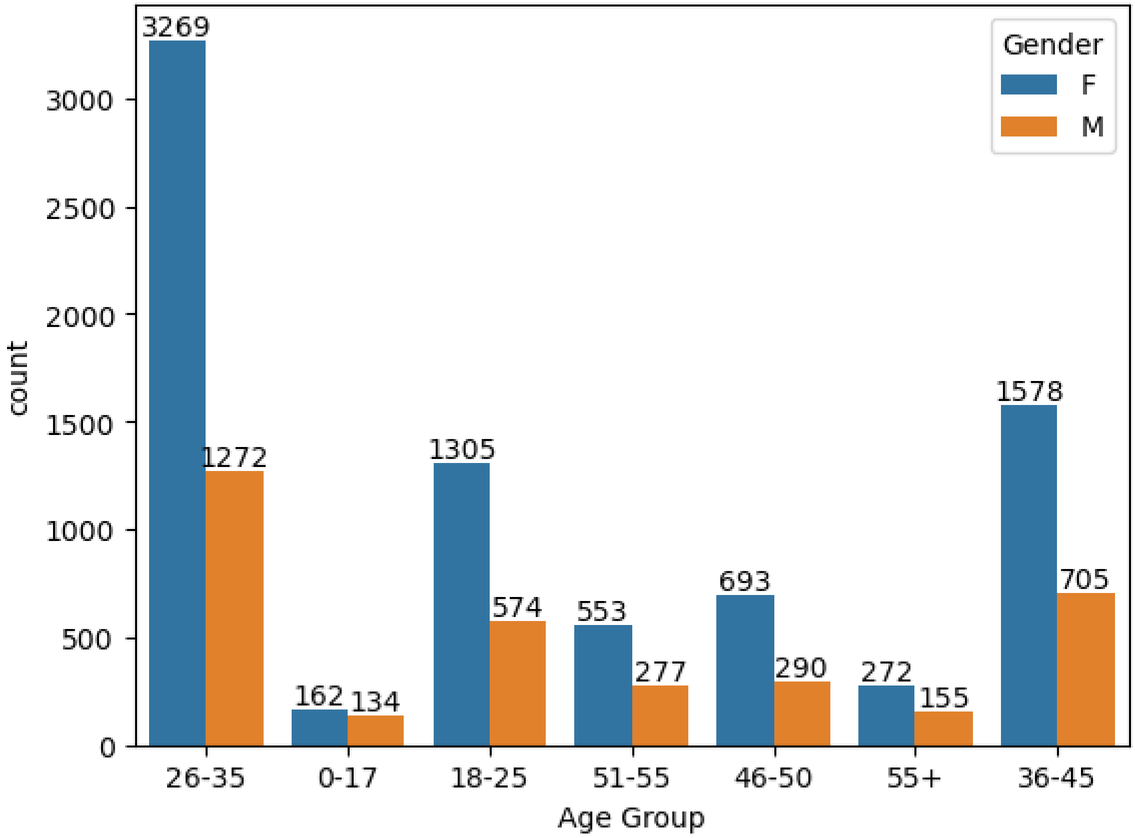


From the above graphs we can see that most of the buyers are females and even the purchasing power of females is greater than man

Age

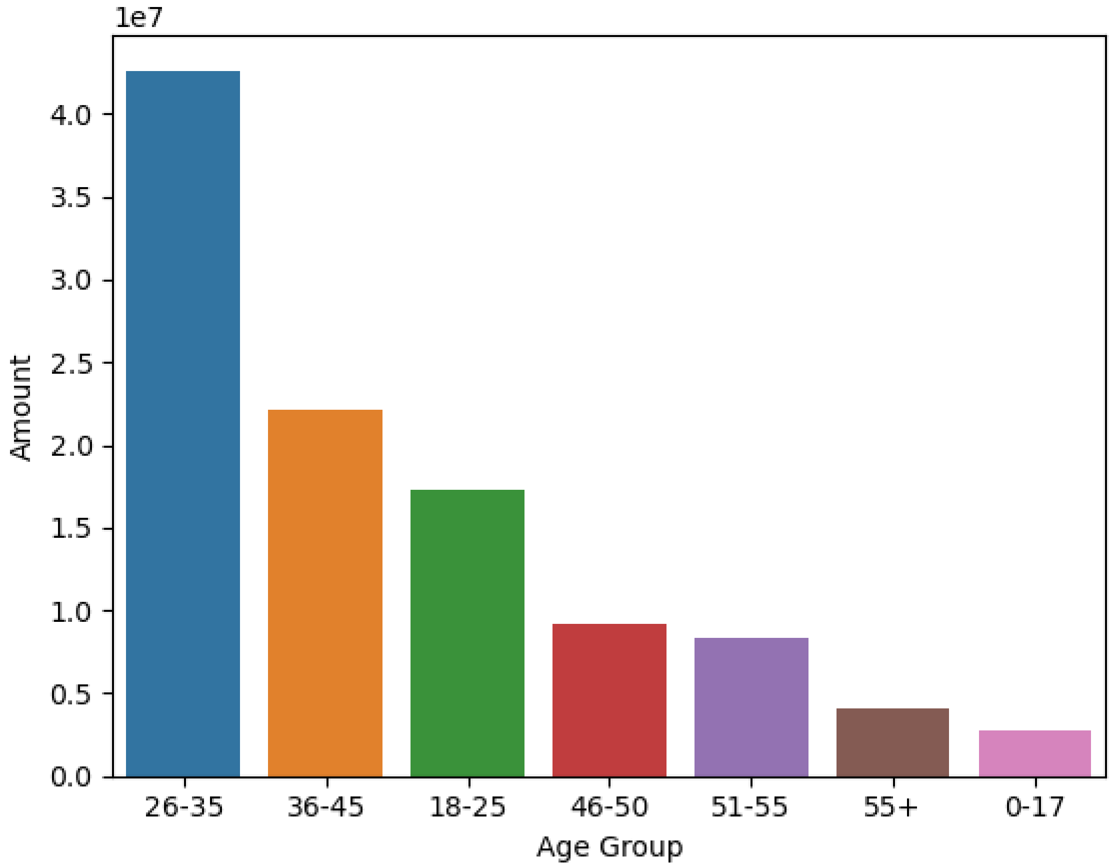
In [17]:

```
ag=sns.countplot(data=df, x='Age Group', hue='Gender')

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



In [18]:
```
#Total Amount vs Age Group
sales_age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by

sns.barplot(x='Age Group', y='Amount', data=sales_age)
```

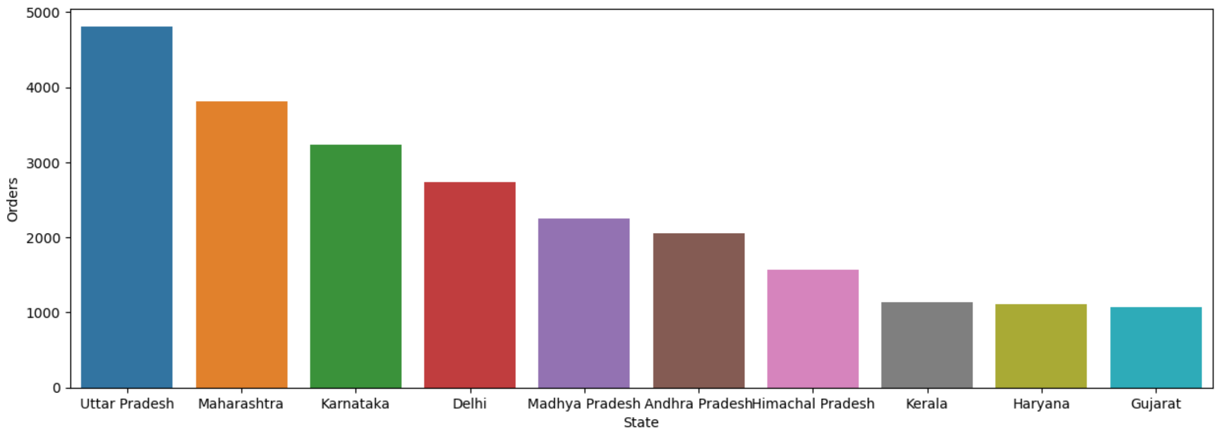
Out[18]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



State

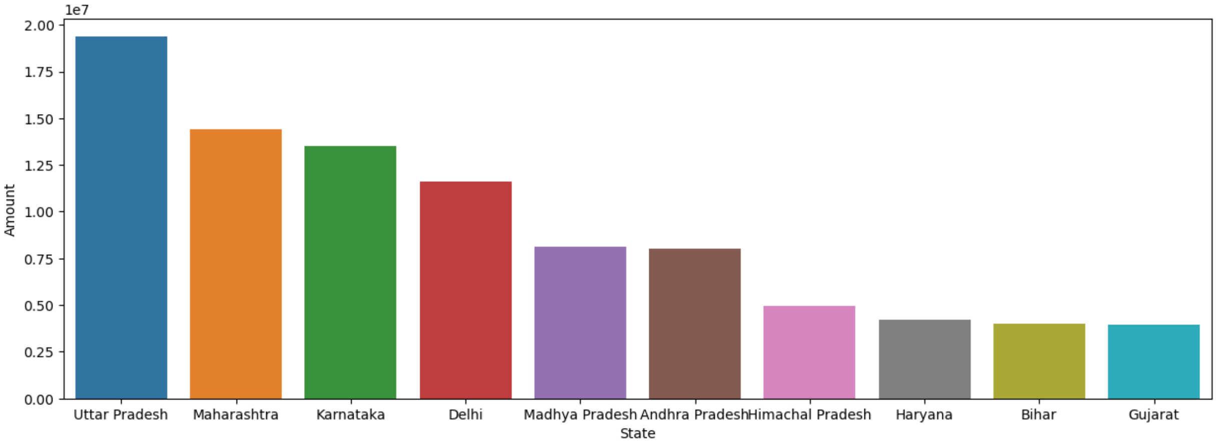
In [19]:
```
#Total number of orders from top 10 states

sales_state=df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by

plt.figure(figsize=(15,5))
sns.barplot(x='State', y='Orders', data=sales_state)
```

Out[19]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



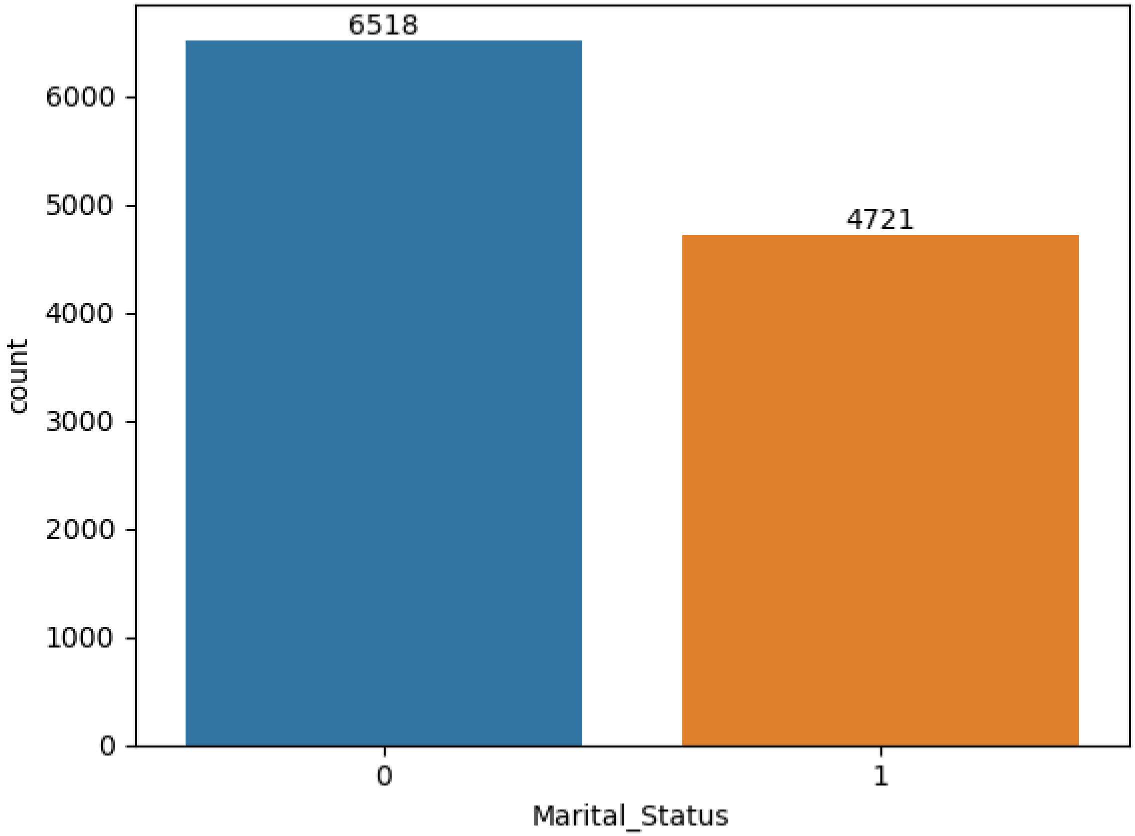
```
In [20]:  #Total amount/sales from top 10 states

          sales_state=df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by

          plt.figure(figsize=(15,5))
          sns.barplot(x='State', y='Amount', data=sales_state)
```

```
Out[20]: <Axes: xlabel='State', ylabel='Amount'>
```



Marital Status

```
In [21]:  ms=sns.countplot(x='Marital_Status',data=df)

          plt.figure(figsize=(7,5))
          for bars in ms.containers:
              ms.bar_label(bars)
```



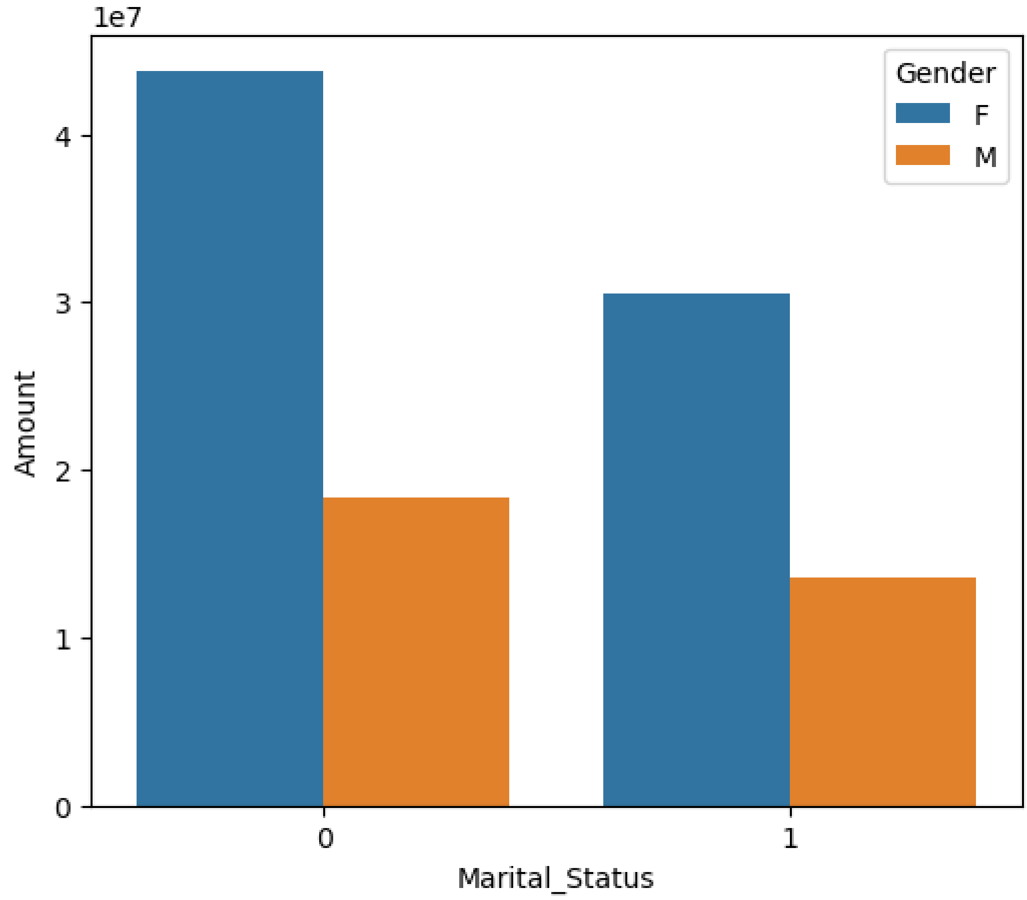
```
<Figure size 700x500 with 0 Axes>
```

In [22]:

```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount']

plt.figure(figsize=(6,5))
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gender')
```

Out[22]:

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

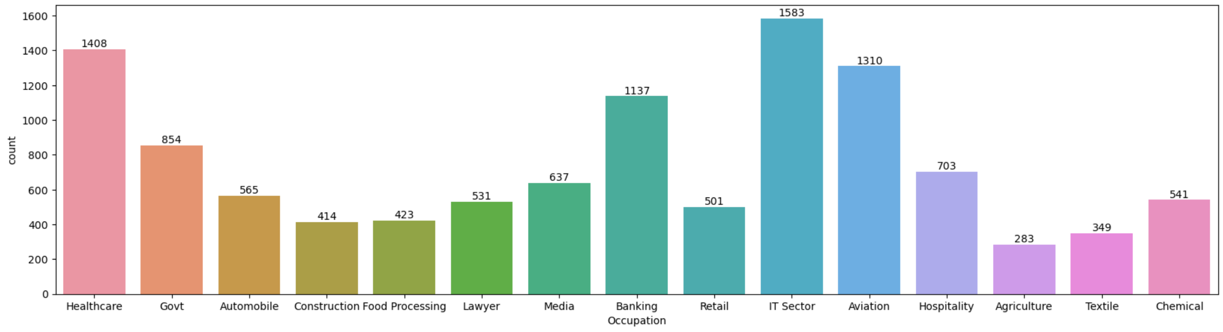


From the above graphs we see that most of the buyers are married women and they have high purchasing power.

Occupation

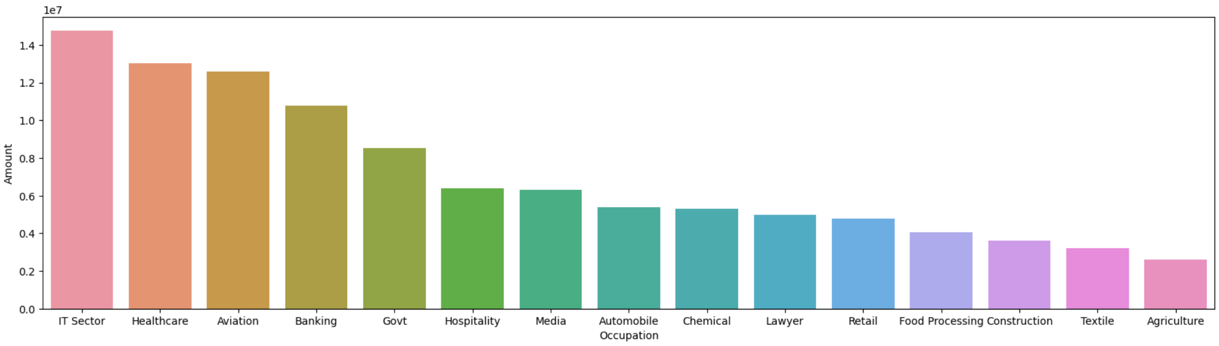
In [23]:

```
plt.figure(figsize=(20,5))
occ=sns.countplot(x='Occupation',data=df)

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



```
In [24]:  sales_state=df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_value

          plt.figure(figsize=(20,5))
          sns.barplot(data=sales_state, x='Occupation', y='Amount')
```

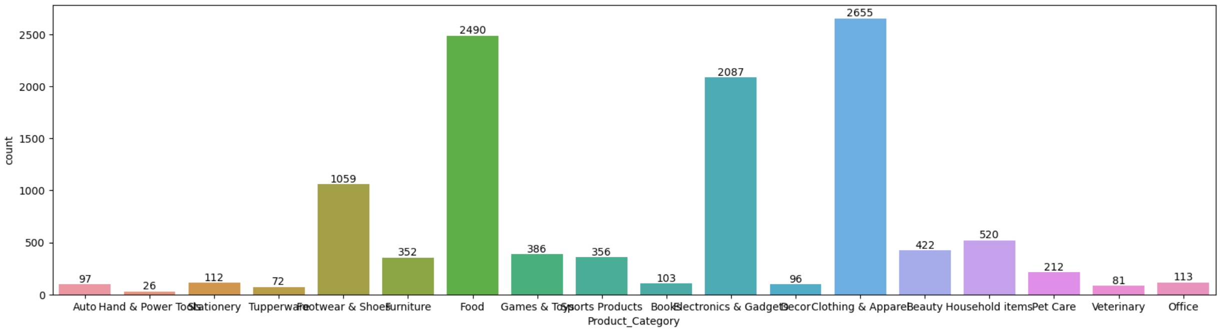
```
Out[24]: <Axes: xlabel='Occupation', ylabel='Amount'>
```



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

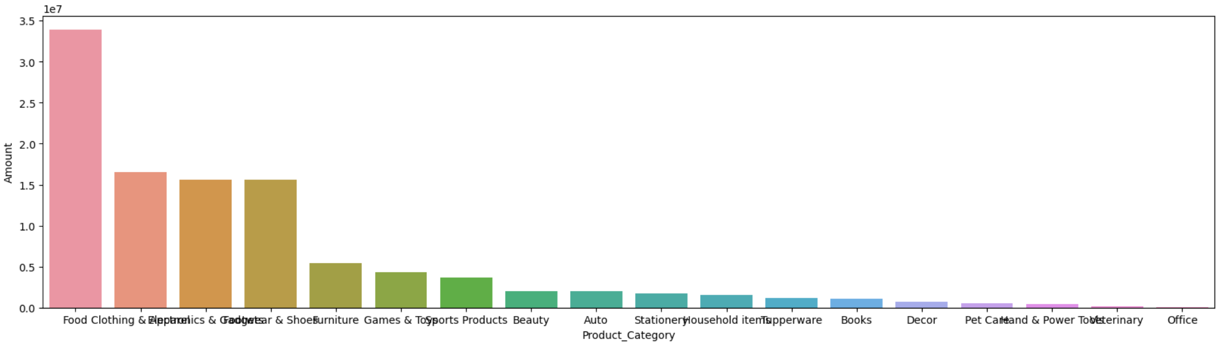
Product Category

```
In [25]:  plt.figure(figsize=(20,5))
          pc=sns.countplot(x='Product_Category',data=df)

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



```
In [26]:  sales_state=df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sor

          plt.figure(figsize=(20,5))
          sns.barplot(data=sales_state, x='Product_Category', y='Amount')
```

```
Out[26]: <Axes: xlabel='Product_Category', ylabel='Amount'>
```



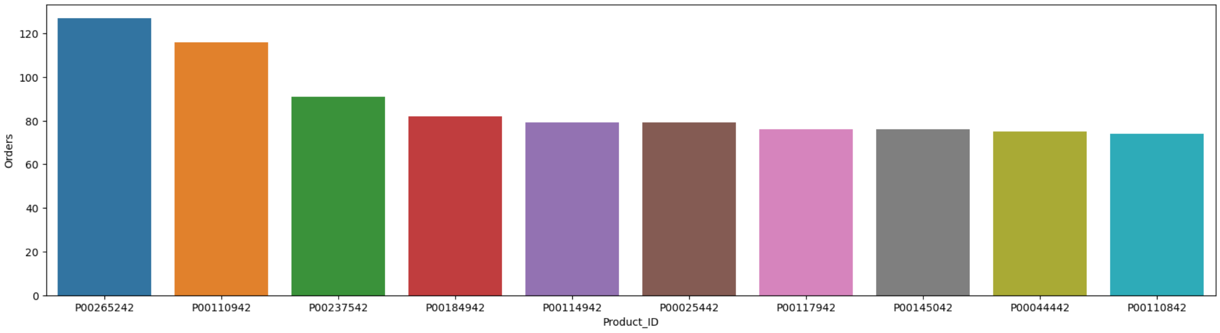
From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

In [27]:

```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_va

plt.figure(figsize=(20,5))
sns.barplot(data = sales_state, x = 'Product_ID',y= 'Orders')
```

Out[27]:

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

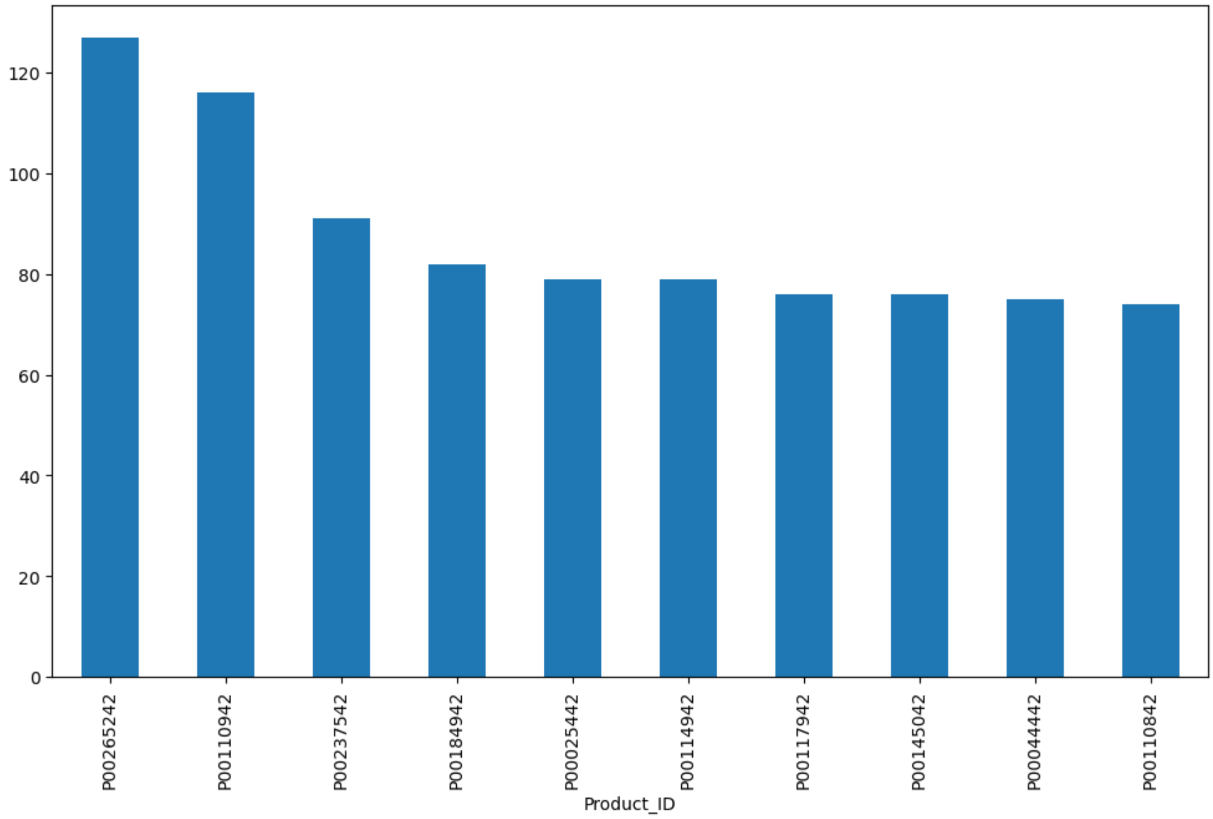


In [28]:

```
#top 10 most sold products

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

Out[28]:

```
<Axes: xlabel='Product_ID'>
```



Conclusion:

Married women in the age group 26-35 yrs from UP, Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothing and Electronics category.