In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # Visualization data
%matplotlib inline
import seaborn as sns
```

In [2]:
```python
df=pd.read_csv("C://Users//HP//OneDrive//Desktop//Diwali_Sales_Data.csv")
```

In [3]:
```python
df.shape
```

Out[3]:
```
(11251, 15)
```

In [4]:
```python
df.head(10)
```

Out[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN	NaN
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	23877.00	NaN	NaN
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN	NaN
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	NaN	NaN	NaN
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN	NaN
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23799.99	NaN	NaN

In [5]:
```python
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
```

In [ ]:
```python
# drop unrelated/ blanks columns
```

In [6]:
```python
df.drop(['Status','unnamed1'], axis=1, inplace=True)
```

In [7]:
```python
pd.isnull(df)
```

Out[7]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

In [10]:
```python
# cheack for null values
pd.isnull(df).sum()
```

Out[10]:
```
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
```

In [8]:
```python
df.shape
```

Out[8]:
```
(11251, 13)
```

In [9]:
```python
#drop null values
df.dropna(inplace=True)
```

In [10]:
```python
df.shape
```

Out[10]:
```
(11239, 13)
```

In [11]:
```python
# change data type
df['Amount']=df['Amount'].astype('int')
```

In [12]:
```python
df['Amount'].dtypes
```

Out[12]:
```
dtype('int32')
```

In [13]:
```python
df.columns
```

Out[13]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [14]:
```python
# rename column
df.rename(columns={'Marital_Status':'Shadii'})
```

Out[14]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shadii	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns

In [19]:
```python
# describe()method to returns description of the data in the Dataframe(i.e count,mean,std etc)
df.describe()
```

Out[19]:

	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 [15]:
```python
# use describe() for specific columns
df[['Age','Orders','Amount']].describe()
```

Out[15]:

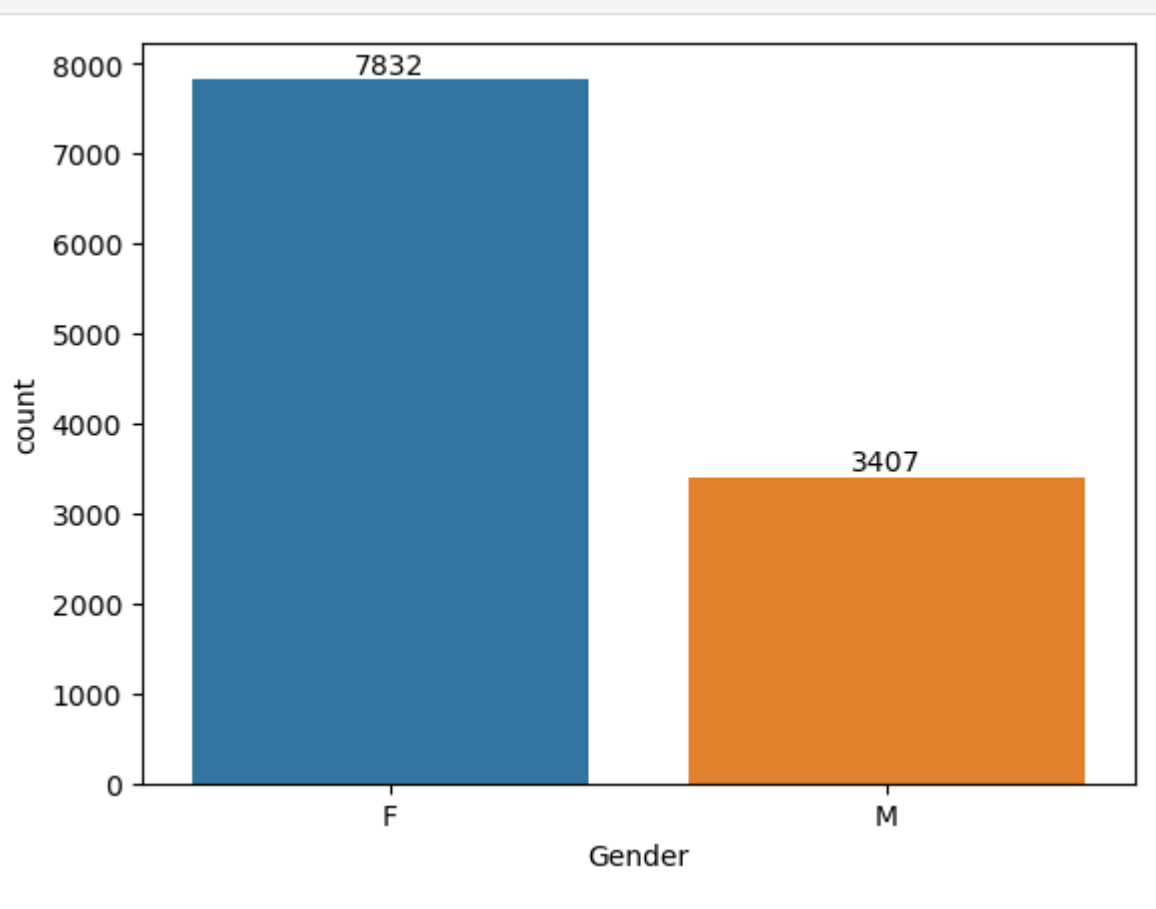
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

In [ ]:
```python
# EDA----------------                EXPLORATORY DATA ANALYSIS
```

In [17]:
```python
df.columns
```

Out[17]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [19]:
```python
ax=sns.countplot(x='Gender',data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```



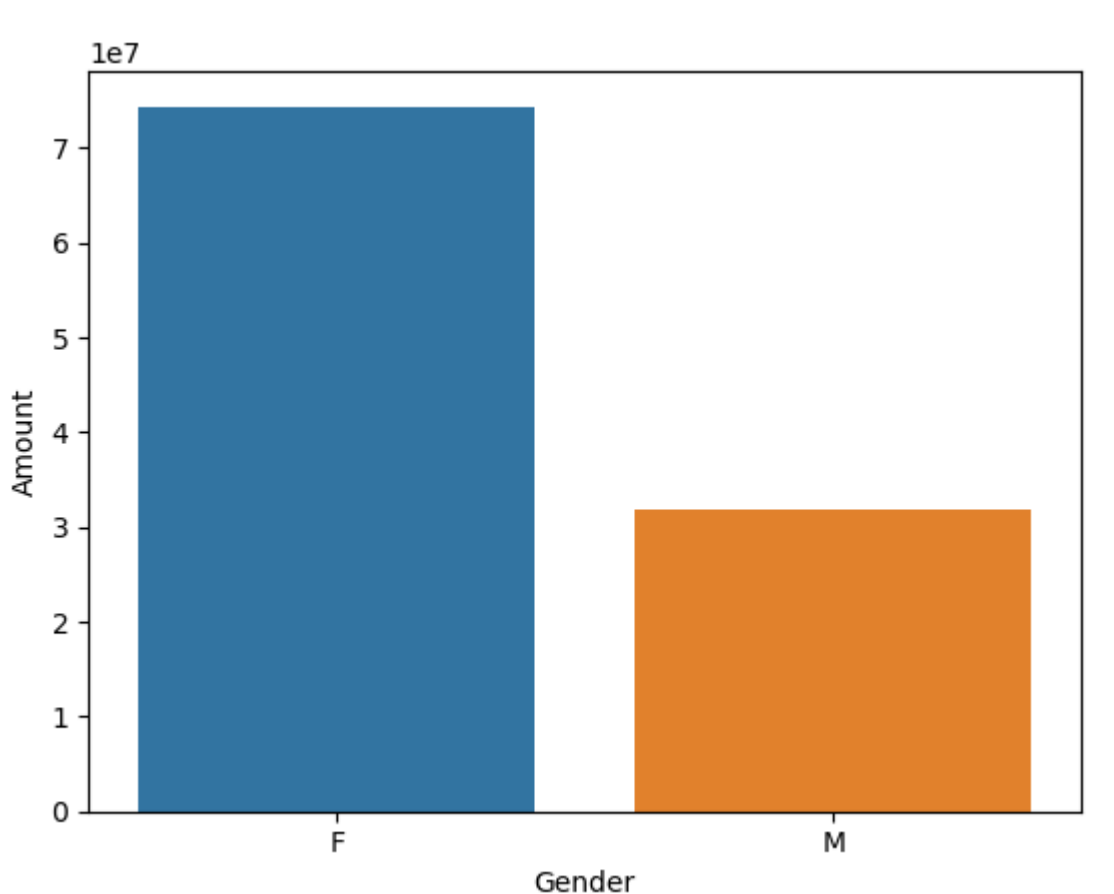
In [20]:
```python
df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
```

Out[20]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [21]:
```python
sales_gen=df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Gender',y='Amount',data=sales_gen)
```

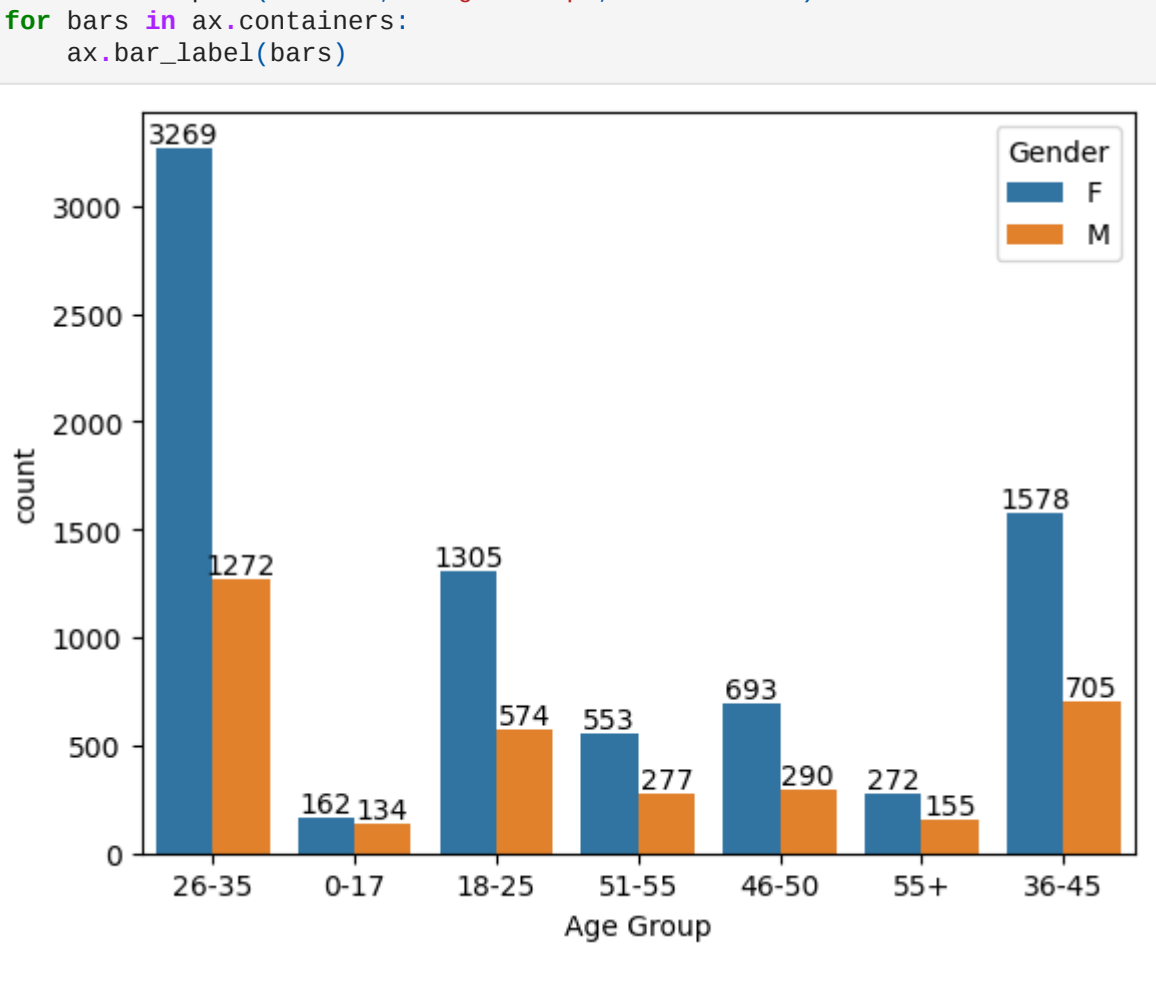
Out[21]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



In [ ]:
```python
# ish graph ko dekhne se ye pta chalta hai ki jada saman females ne hi kharida hai mens ke mukabale
```

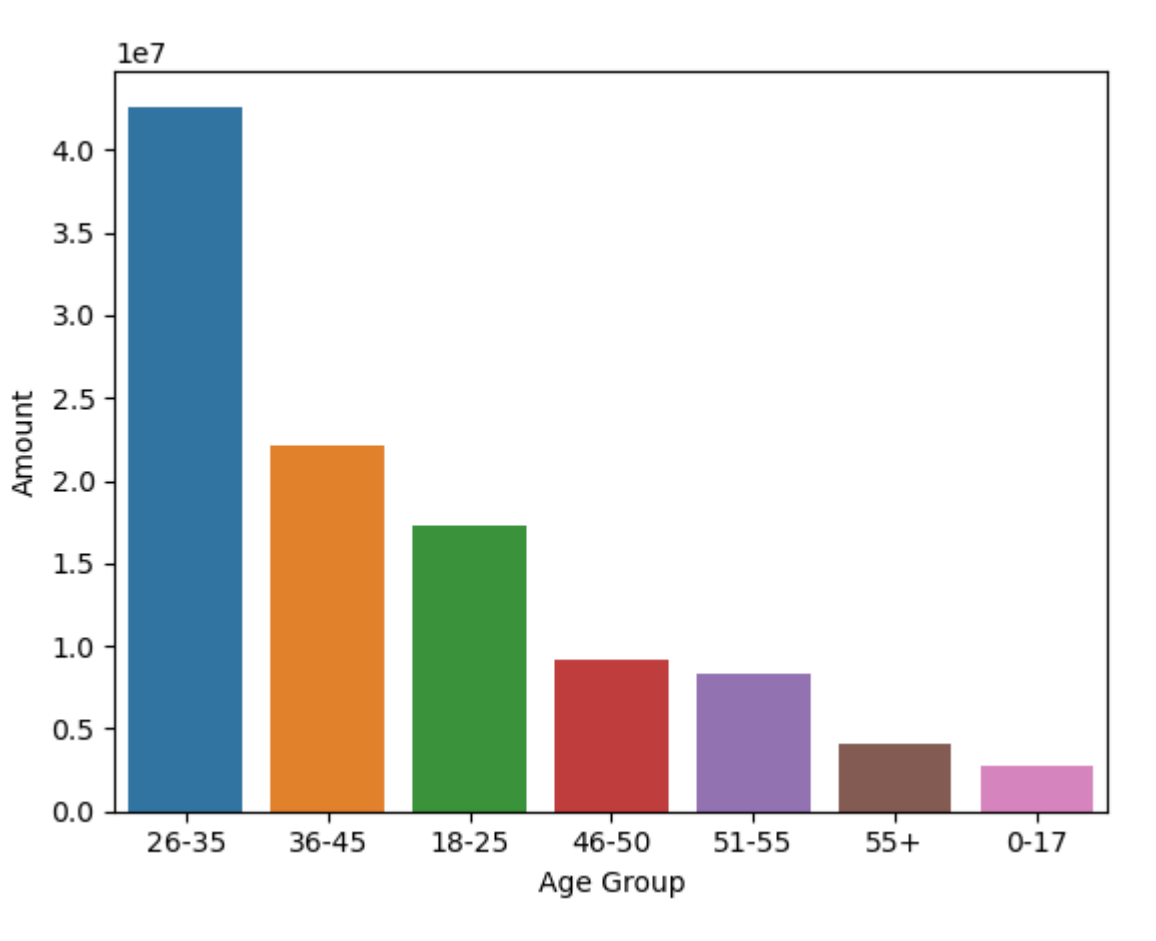
In [ ]:
```python
##-------------------AGE
```

In [22]:
```python
ax=sns.countplot(data=df,x='Age Group',hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [23]:
```python
# 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)
```

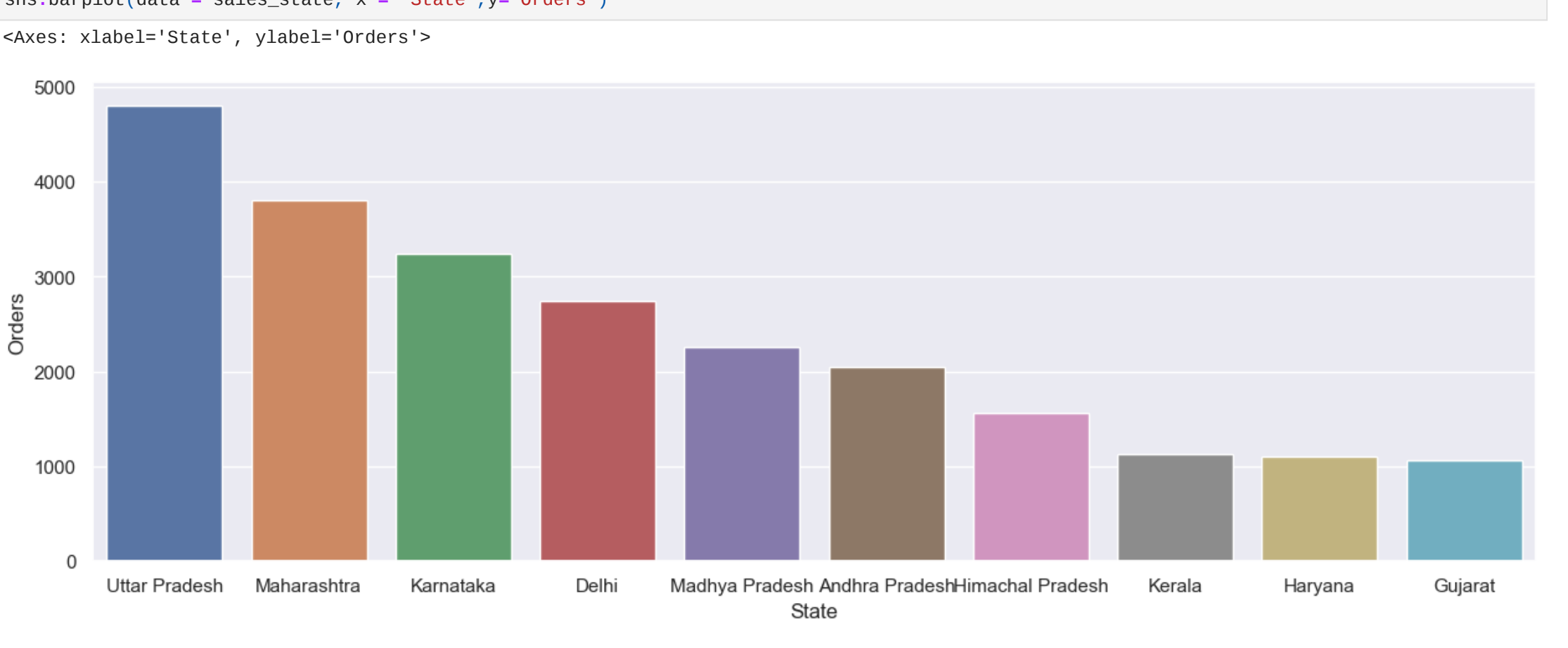
Out[23]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



In [ ]:
```python
#graph ko dekhne se ye pta chalta hai ki jadatar kharidari karne wali females 26-35 age ki hai .
```

In [24]:
```python
# Totals numbers of order 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':(15,5)})
sns.barplot(data = sales_state, x = 'State',y='Orders')
```

Out[24]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



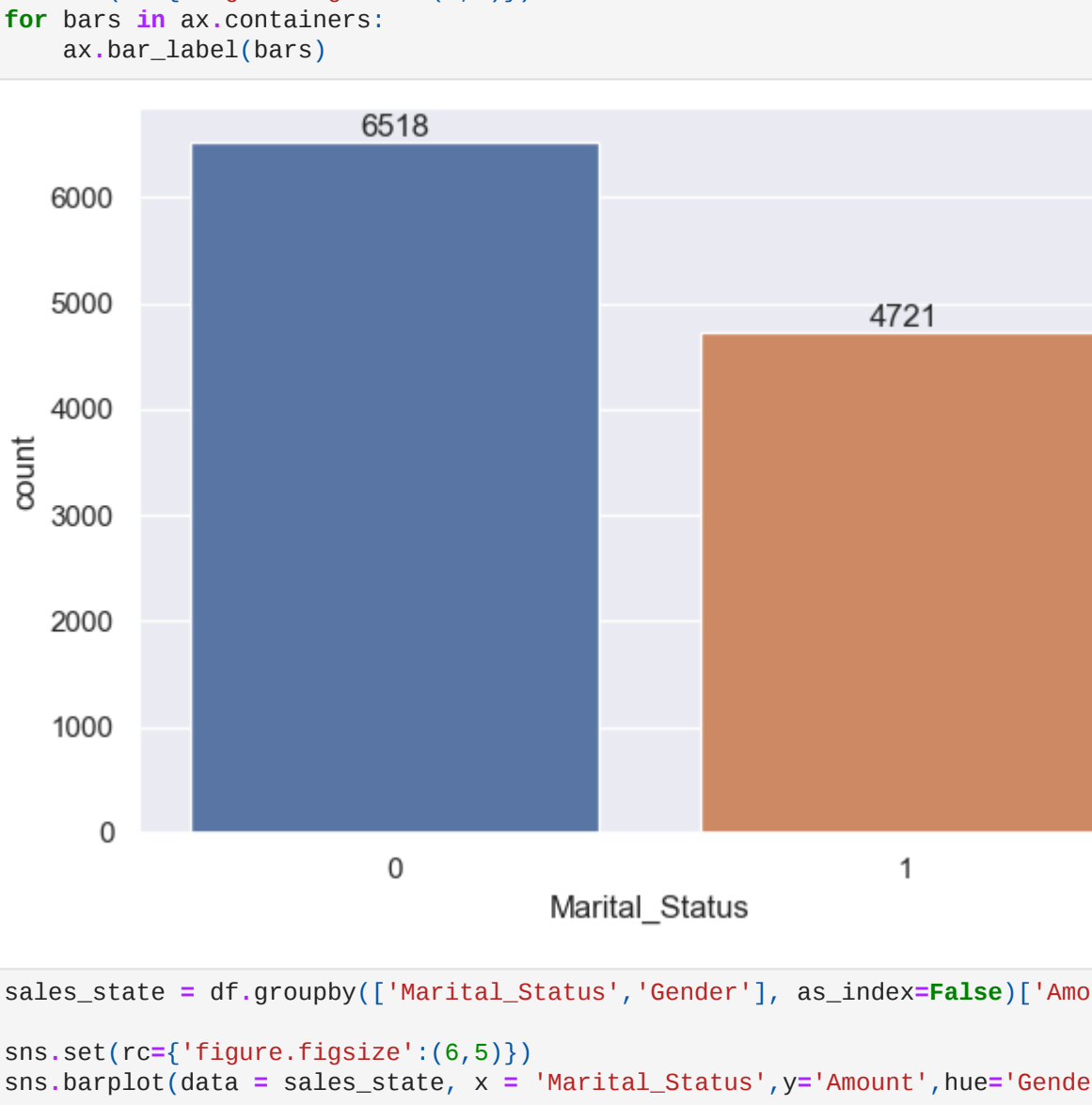
In [25]:
```python
# Totals amount/sales from  top 10 states

sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y='Amount')
```

Out[25]:
```
<Axes: xlabel='State', ylabel='Amount'>
```

In [ ]:
```
hmey graph dekhne se ye pta chalta hai ki jada tar order Uttarpradesh, Maharashtra,and Karnataka aur sales amount v ishi inhi 3 saharo ka hi
```

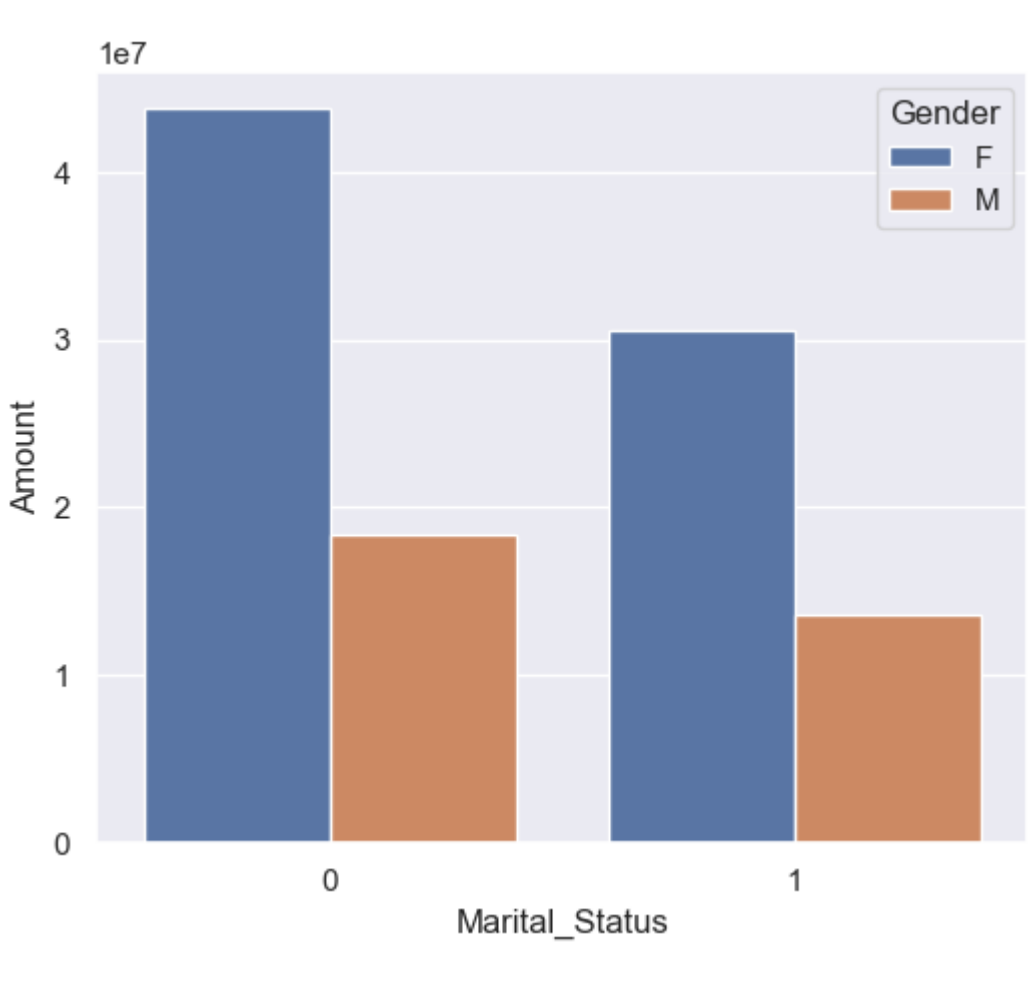
In [ ]:
```python
# Marital_status
```

In [28]:
```python
ax= sns.countplot(data=df,x='Marital_Status')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [29]:
```python
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':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y='Amount',hue='Gender')
```

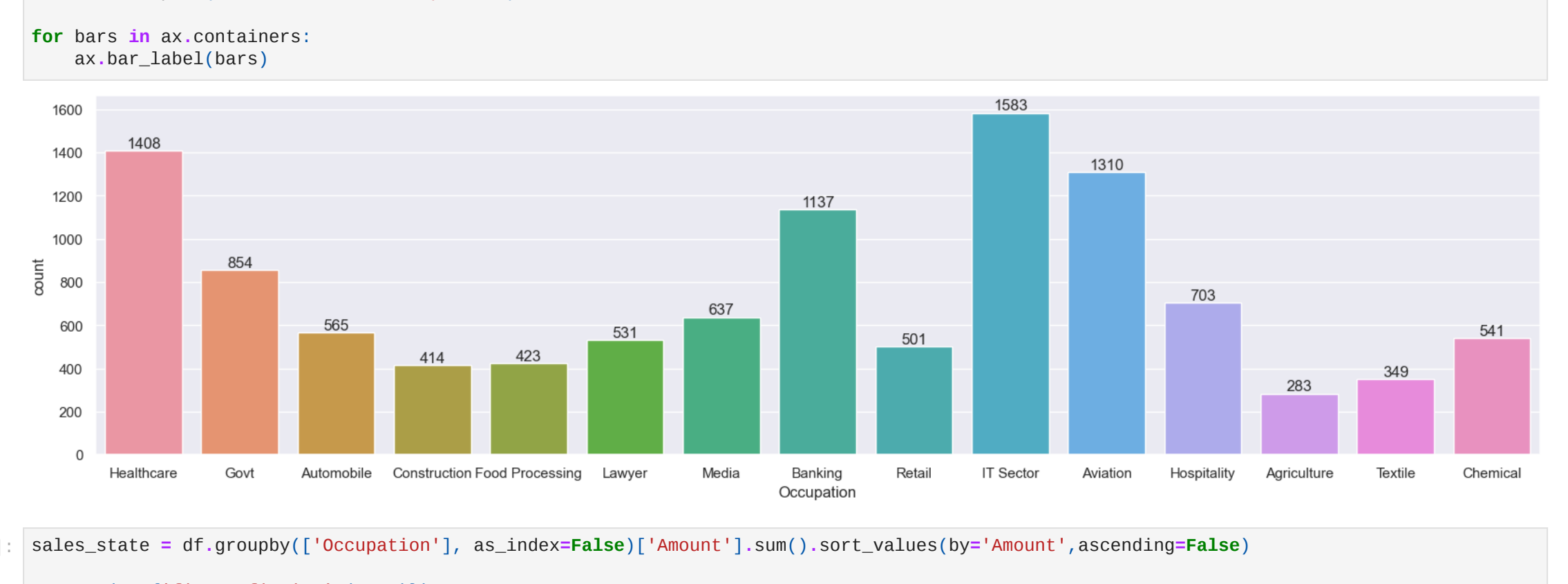
Out[29]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



In [ ]:
```python
# jo sabsey jada shopping ki hai wo married women hai .aur inka purchasing power v jada hai.
```

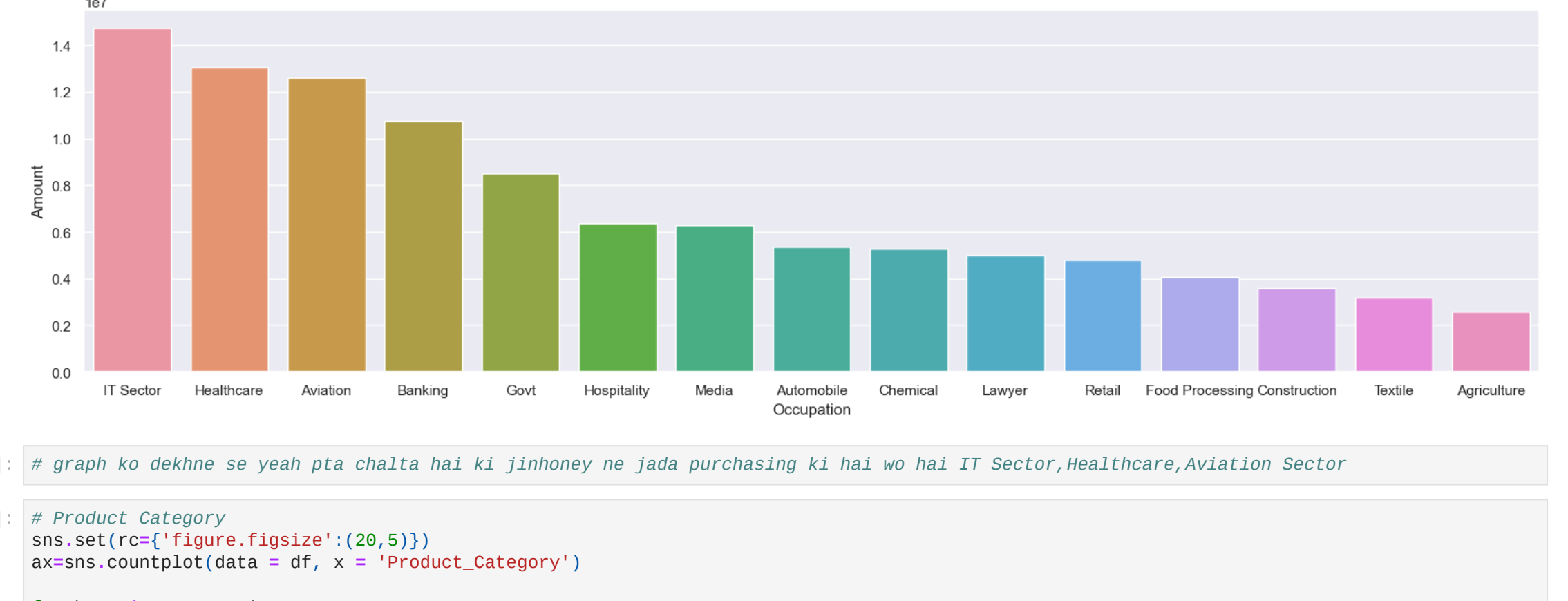
In [ ]:
```python
# Occupation
```

In [30]:
```python
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data = df, x = 'Occupation')

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



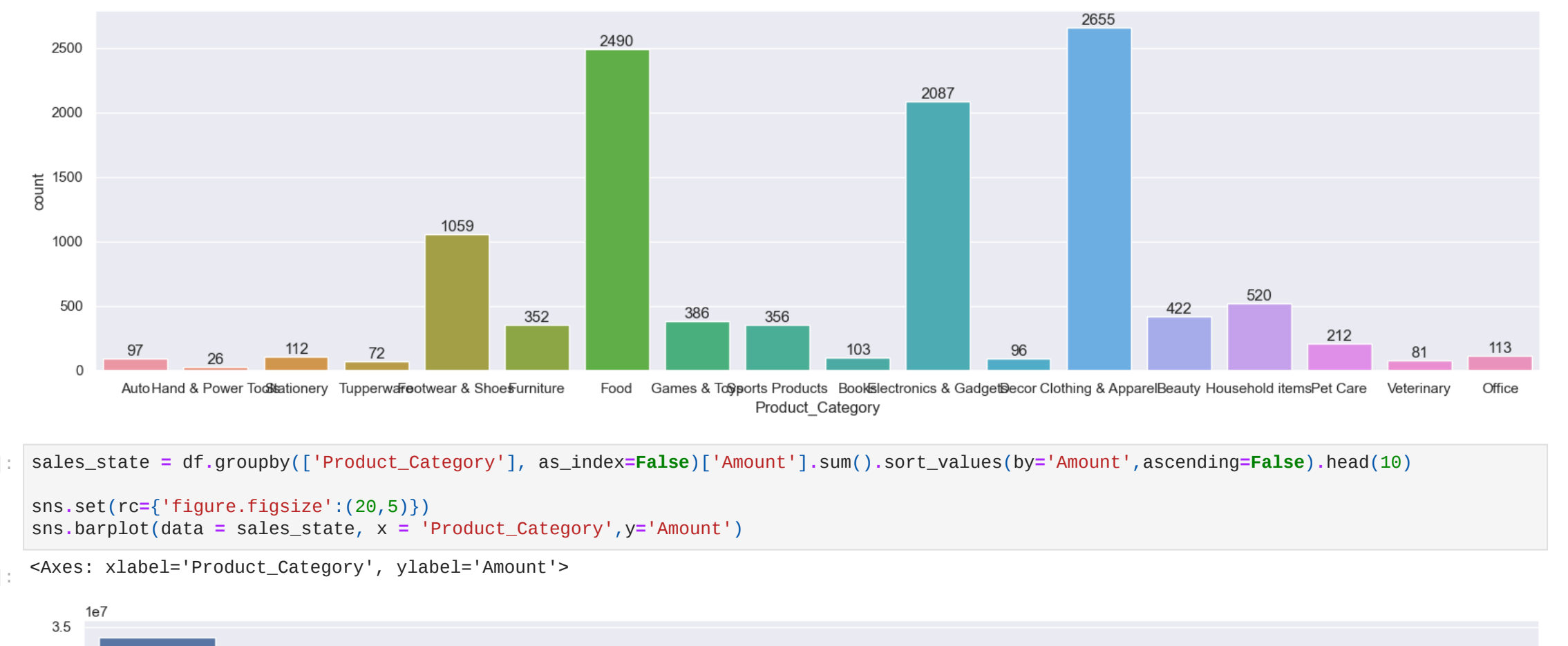
In [31]:
```python
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Occupation',y='Amount')
```

Out[31]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



In [ ]:
```python
# graph ko dekhne se yeah pta chalta hai ki jinhoney ne jada purchasing ki hai wo hai IT Sector,Healthcare,Aviation Sector
```

In [37]:
```python
# Product Category
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data = df, x = 'Product_Category')

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



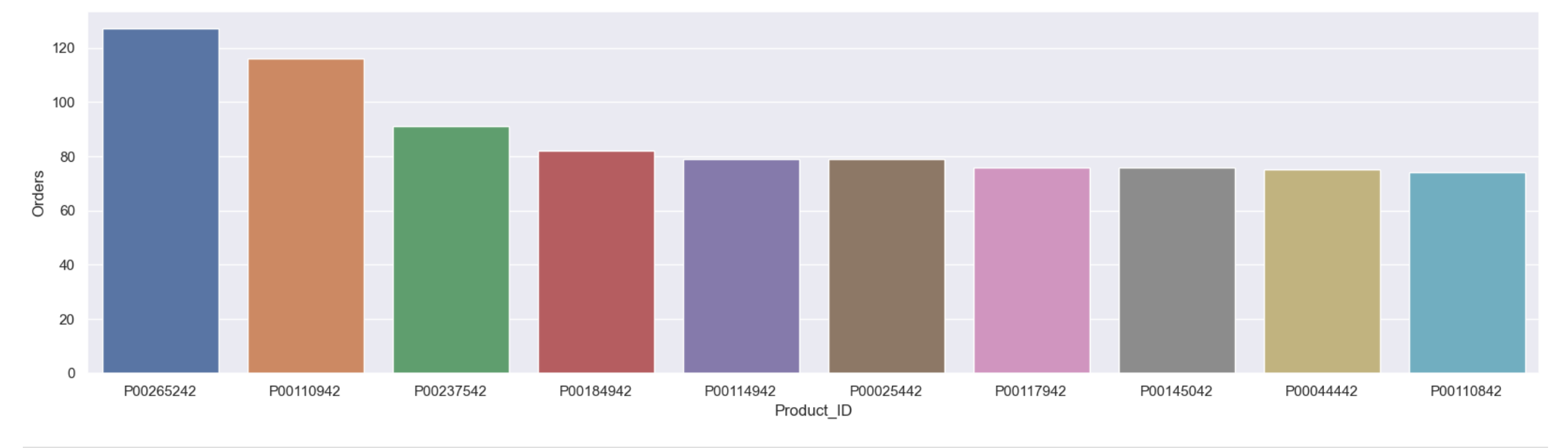
In [35]:
```python
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_Category',y='Amount')
```

Out[35]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```

In [ ]:
```python
# graph ko dekhne se ye pta chal rha hai ki jis product ki sales hui hai wo hai Food,Cloths,and Electronic Gadgets
```

In [38]:
```python
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_ID',y='Orders')
```

Out[38]:
```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```



In [ ]:
```
# Conclusion:
 Married Women age group 26-35 yrs from UP,Maharastra and Kerla working in IT,Healthcare,and Aviation are more likely to
    to buy products from Food, Clothing and Electronic Category ...
```