In [8]:

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

In [9]:

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

Out[9]:

	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	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0	NaN	NaN
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0	NaN	NaN
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0	NaN	NaN
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0	NaN	NaN
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0	NaN	NaN

11251 rows × 15 columns

In [10]:

```
df.shape
```

Out[10]:

```
(11251, 15)
```

In [11]:

```
df.dtypes
```

Out[11]:

```
User_ID               int64
Cust_name            object
Product_ID           object
Gender               object
Age Group            object
Age                   int64
Marital_Status        int64
State                object
Zone                 object
Occupation           object
Product_Category     object
Orders                int64
Amount              float64
Status              float64
unnamed1            float64
dtype: object
```

In [12]:

```
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 [13]:

```
# here status and unamed1 columns are full of null values
```

In [14]:

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

In [15]:

```
df.head()
```

Out[15]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0

In [16]:

```
df.isnull()
```

Out[16]:

	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 [17]:

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

Out[17]:

```
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 [18]:

```
# here we have 12 null values in the amount column.
# removing the null values
```

In [20]:

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

In [21]:

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

Out[21]:

```
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              0
dtype: int64
```

In [22]:

```
df.shape
```

Out[22]:

```
(11239, 13)
```

In [24]:

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

In [25]:

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

Out[25]:

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

In [26]:

```
df.columns
```

Out[26]:

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

In [30]:

```
df[['Age','Orders','Amount']].describe()
```

Out[30]:

	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

Explonatory Data Analysis

Gender

In [31]:

```
df.columns
```

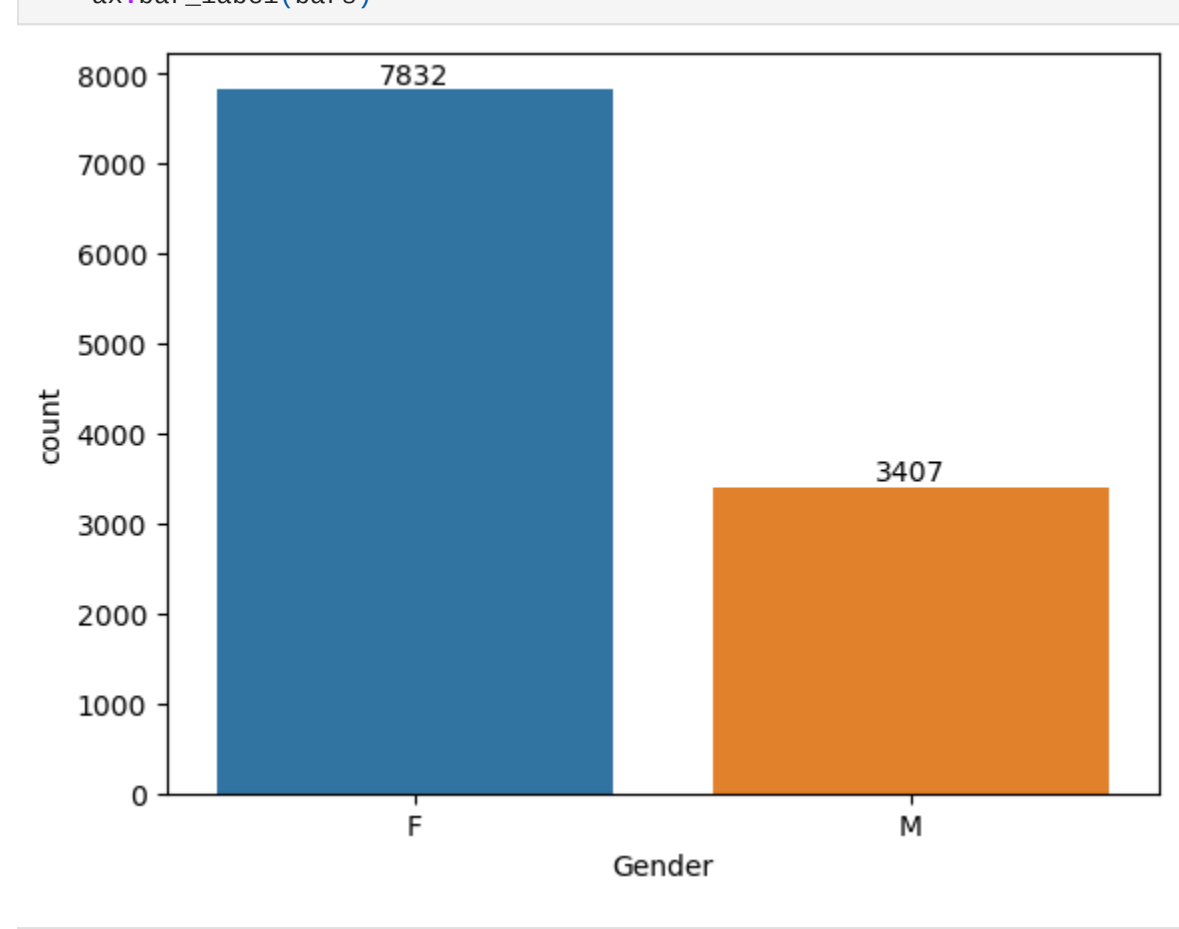
Out[31]:

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

In [35]:

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

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



In [44]:

```
gg=df.groupby(['Gender'], as_index=False)['Amount'].sum()
gg
```

Out[44]:

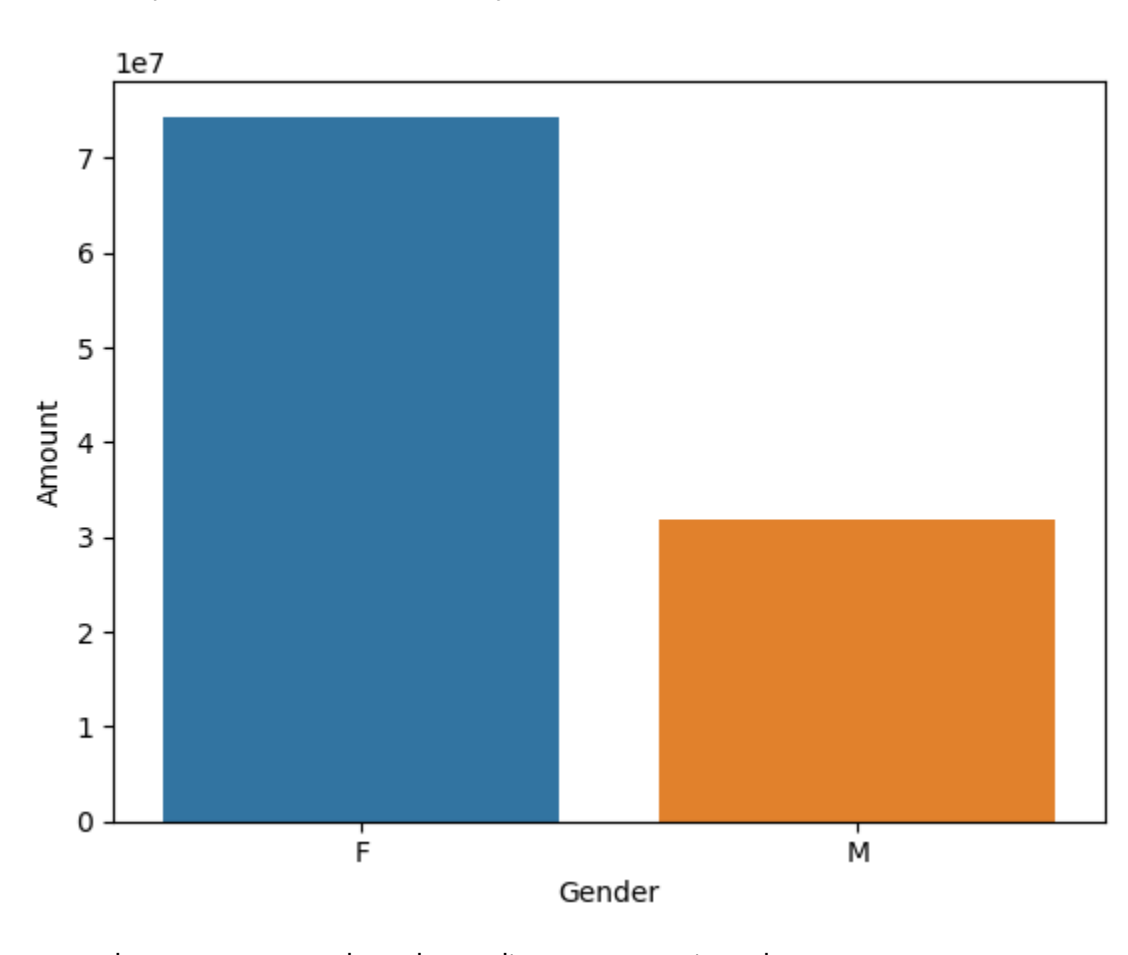
	Gender	Amount
0	F	74335853
1	M	31913276

In [45]:

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

Out[45]:

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

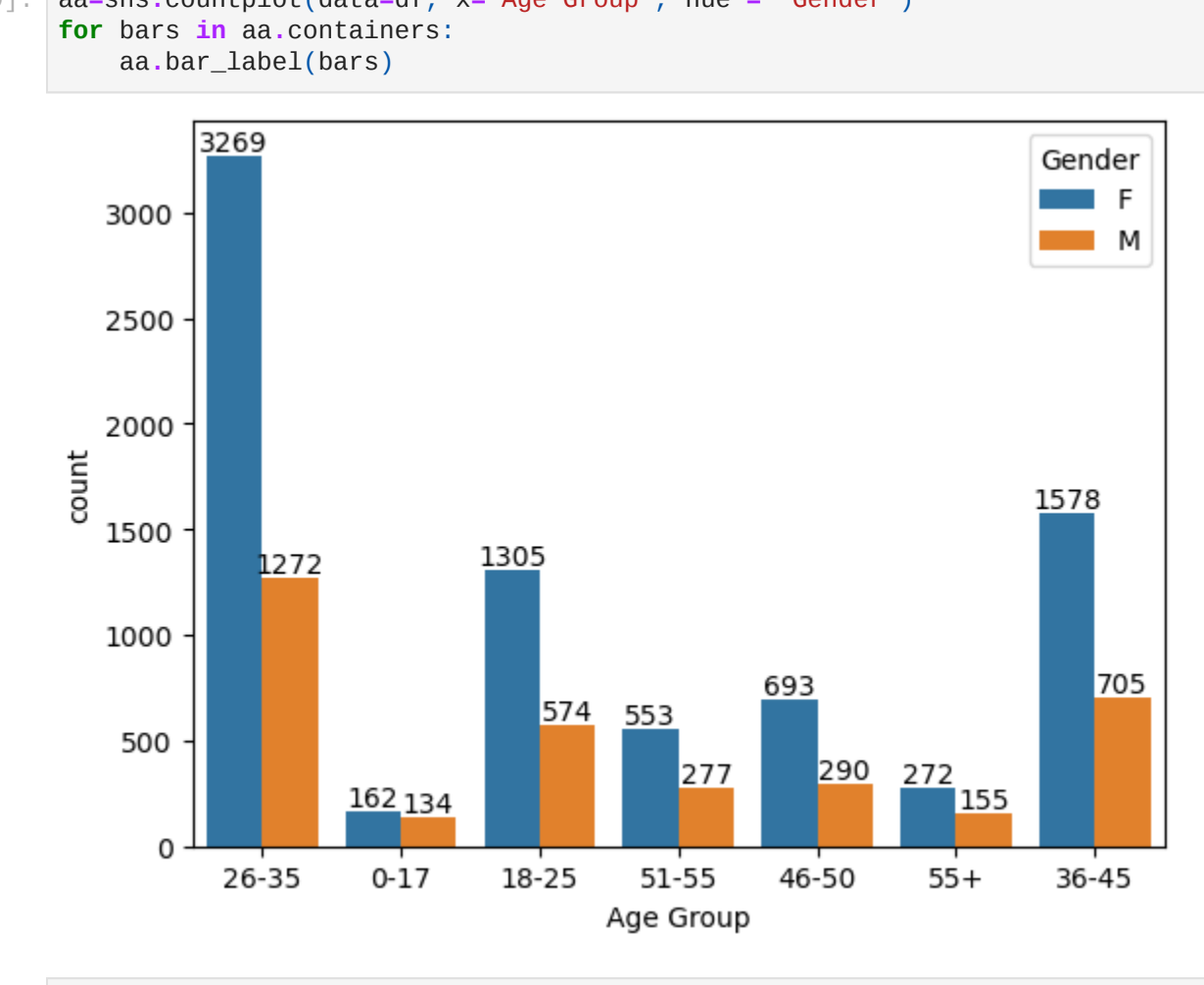


From the above graph, women are purchased more items compare to male.

AGE

In [49]:

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

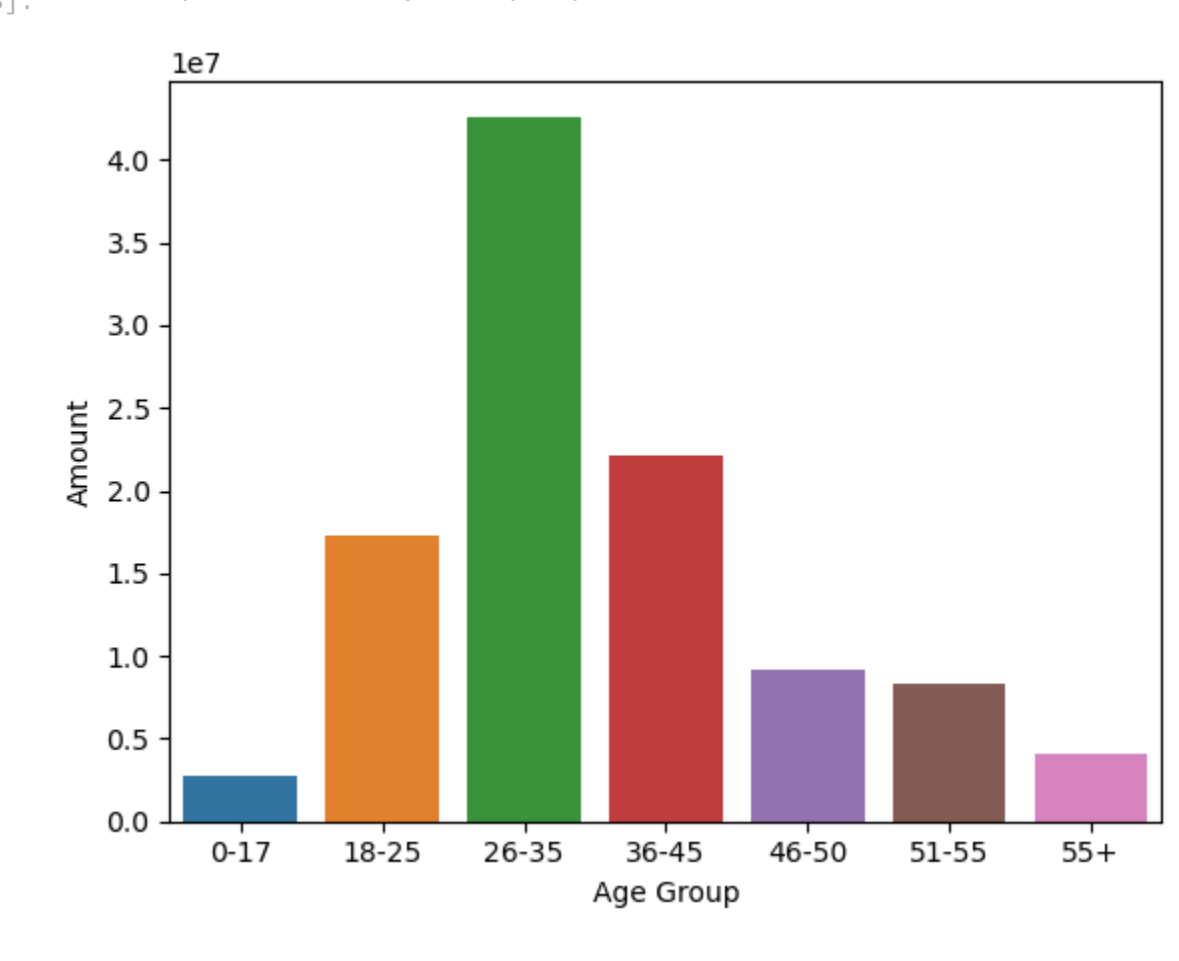


In [53]:

```
# Total Amount VS Age Group
sale_age=df.groupby(['Age Group'], as_index=False)['Amount'].sum()
sns.barplot(data=df, x='Age Group',y='Amount', data = sale_age)
```

Out[53]:

```
<AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```



From above graphs we can see that most of the buyers of age group between 26-33 yrs female

State

In [54]:

```
df.columns
```

Out[54]:

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

In [55]:

```
# total number of order from top states
```

In [67]:

```
tt=df.groupby(['State'], as_index=False)['Orders'].sum().head(10)
tt
```

Out[67]:

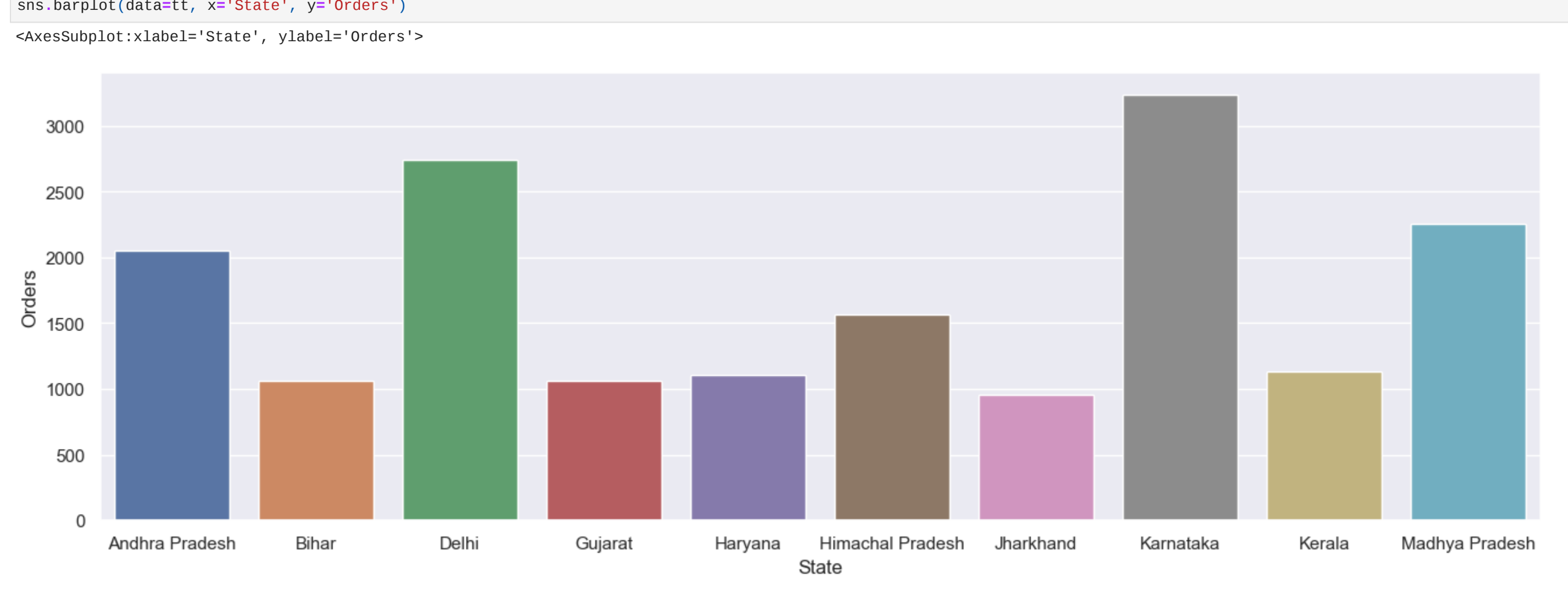
	State	Orders
0	Andhra Pradesh	2051
1	Bihar	1062
2	Delhi	2740
3	Gujarat	1066
4	Haryana	1109
5	Himachal Pradesh	1568
6	Jharkhand	953
7	Karnataka	3240
8	Kerala	1137
9	Madhya Pradesh	2252

In [68]:

```
sns.set(rc={'figure.figsize':(16,5)})
sns.barplot(data=tt, x='State',y='Orders')
```

Out[68]:

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



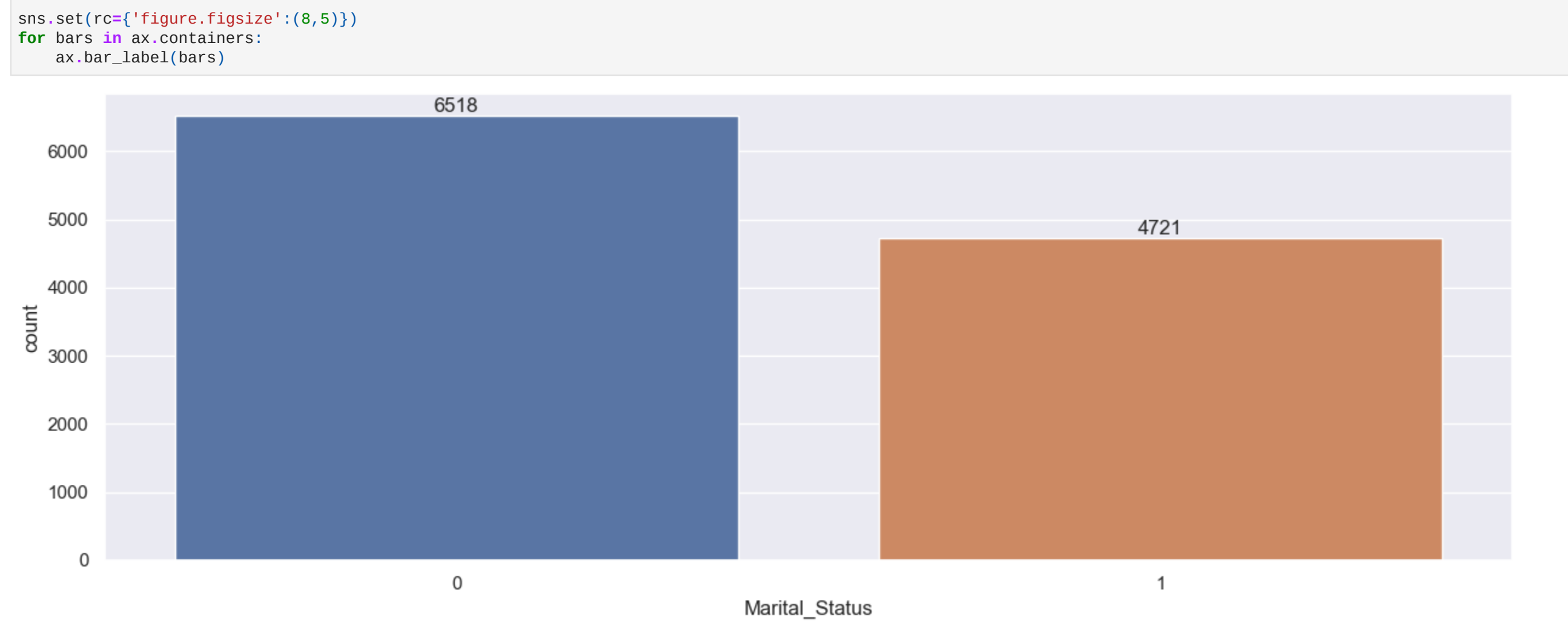
From above graph we can see that most of the orders are from uttar pradesh, maharashtra, and karnataka

Martial Status

In [72]:

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

sns.set(rc={'figure.figsize':(8,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [76]:

```
mm=df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
mm
```

Out[76]:

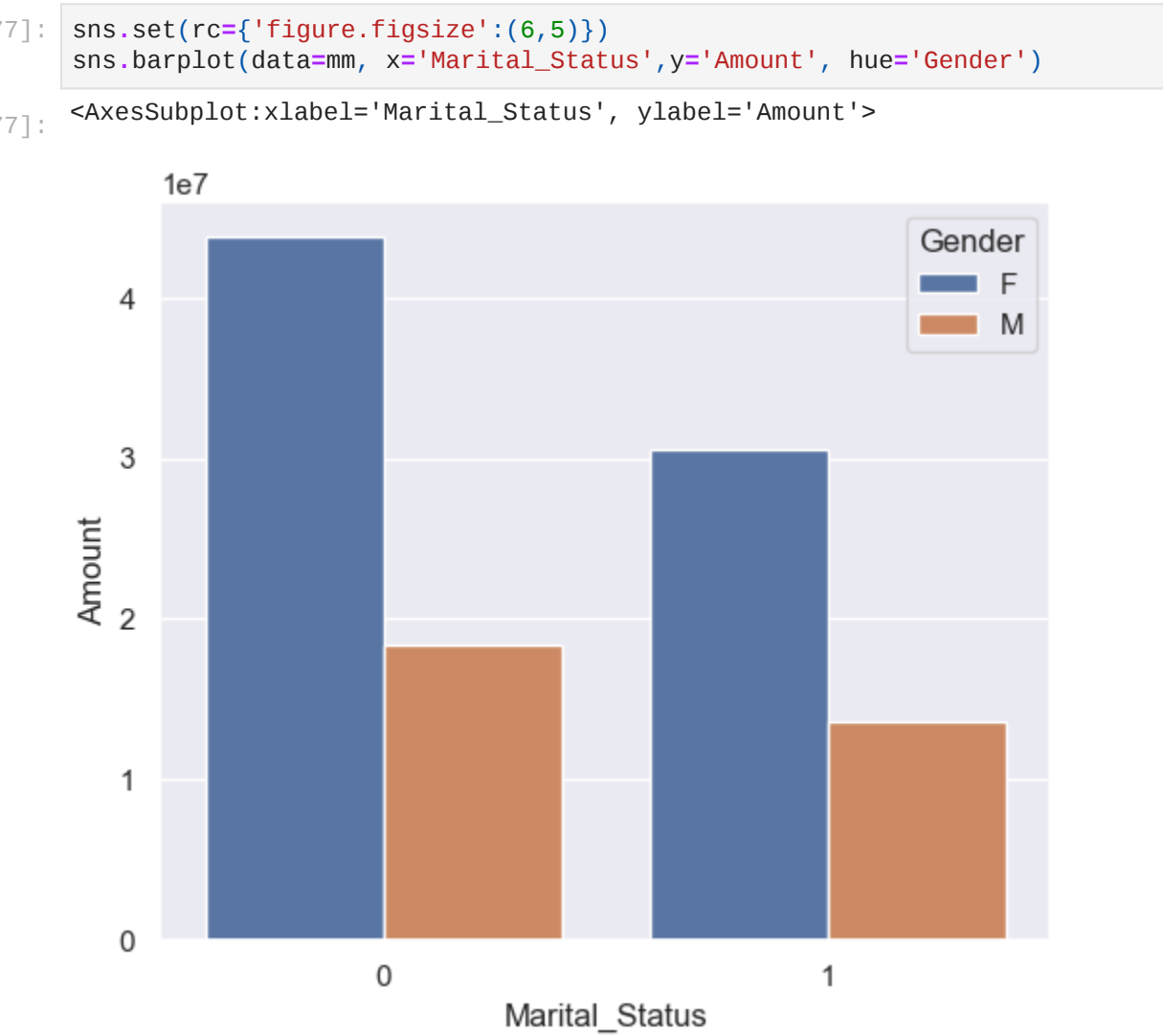
	Marital_Status	Gender	Amount
0	0	F	43786646
2	1	F	30549207
1	0	M	18338738
3	1	M	13574538

In [77]:

```
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data=mm, x='Marital_Status',y='Amount', hue='Gender')
```

Out[77]:

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



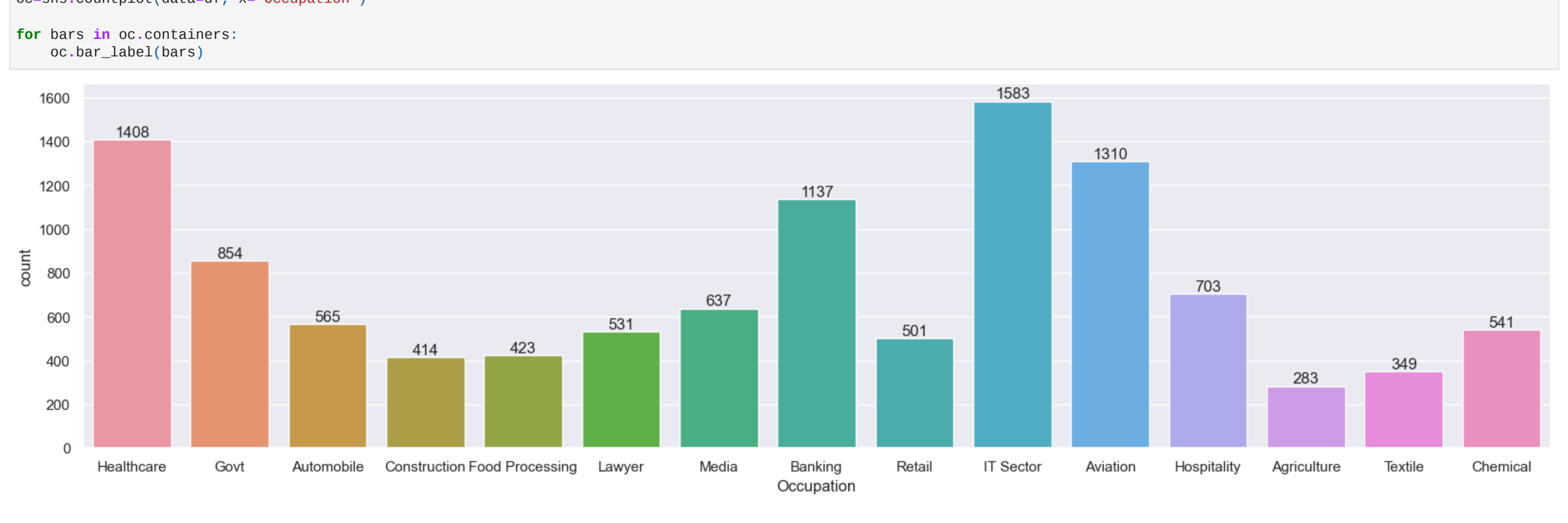
From the abouve graph, we can see that most of the buyers are married women

Occupation

In [81]:

```
sns.set(rc={'figure.figsize':(20,5)})
oc=sns.countplot(data=df, x='Occupation')

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



From the above graph , the most of the buyers are working in IT sectors

In [82]:

```
df.columns
```

Out[82]:

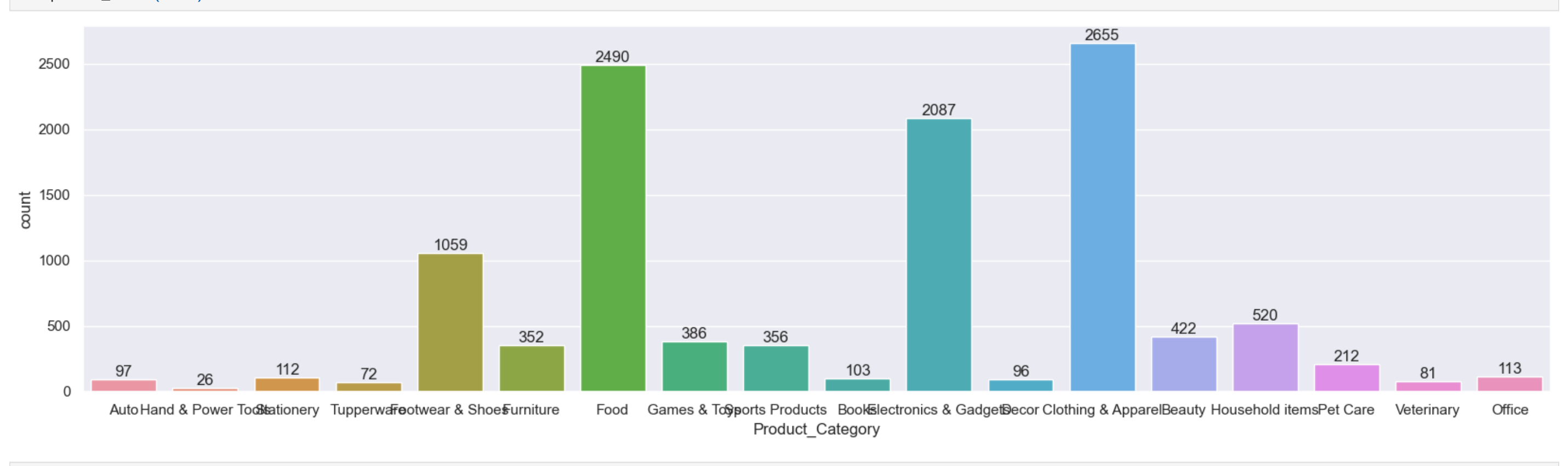
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

Product Category

In [89]:

```
sns.set(rc={'figure.figsize':(20,5)})
pc=sns.countplot(data=df, x='Product_Category')

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



In [99]:

```
pp=df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
pp
```

Out[99]:

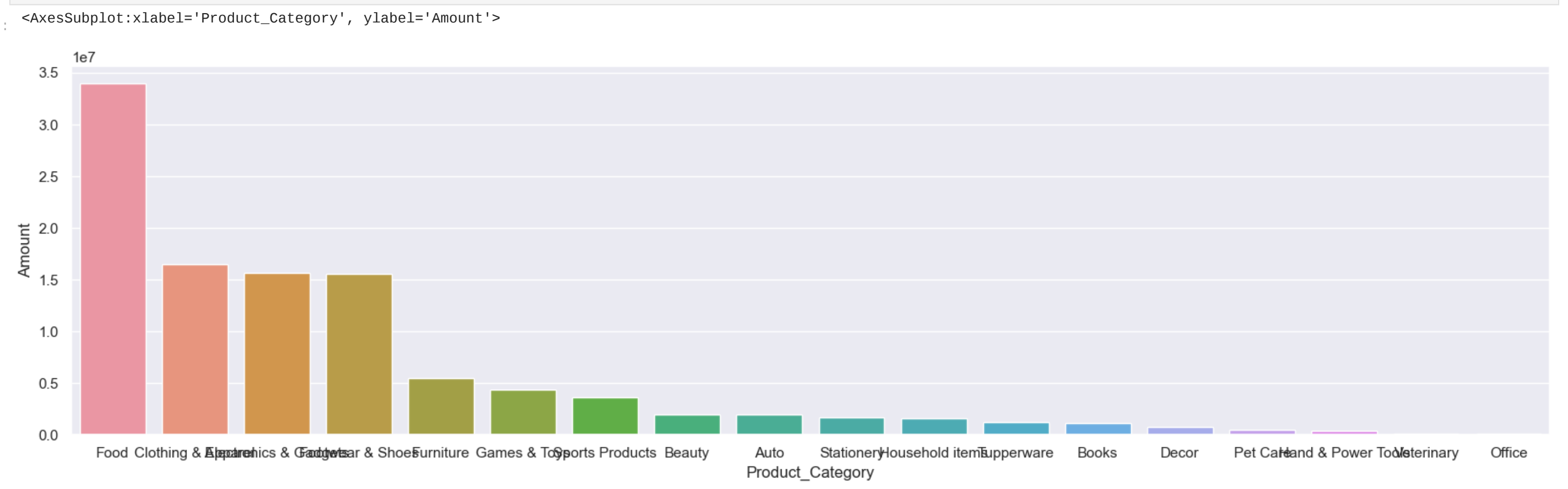
	Product_Category	Amount
6	Food	33933883
3	Clothing & Apparel	16495019
5	Electronics & Gadgets	15643846
7	Footwear & Shoes	15575209
8	Furniture	5440051
9	Games & Toys	4331694
14	Sports Products	3635933
1	Beauty	1959484
0	Auto	1958609
15	Stationery	1676051
11	Household items	1569337
16	Tupperware	1155642
2	Books	1061478
4	Decor	730360
13	Pet Care	482277
10	Hand & Power Tools	405618
17	Veterinary	112702
12	Office	81936

In [100…

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

Out[100]:

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



The Most sold Item is food, Clothing, and electronict category

In [101]:

```
# Top 3
```

In [104]:

```
oo=df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(3)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=oo, x='Product_Category',y='Amount')
```

Out[104]:

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

In [107]:

```
df.columns
```

Out[107]:

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

In [105]:

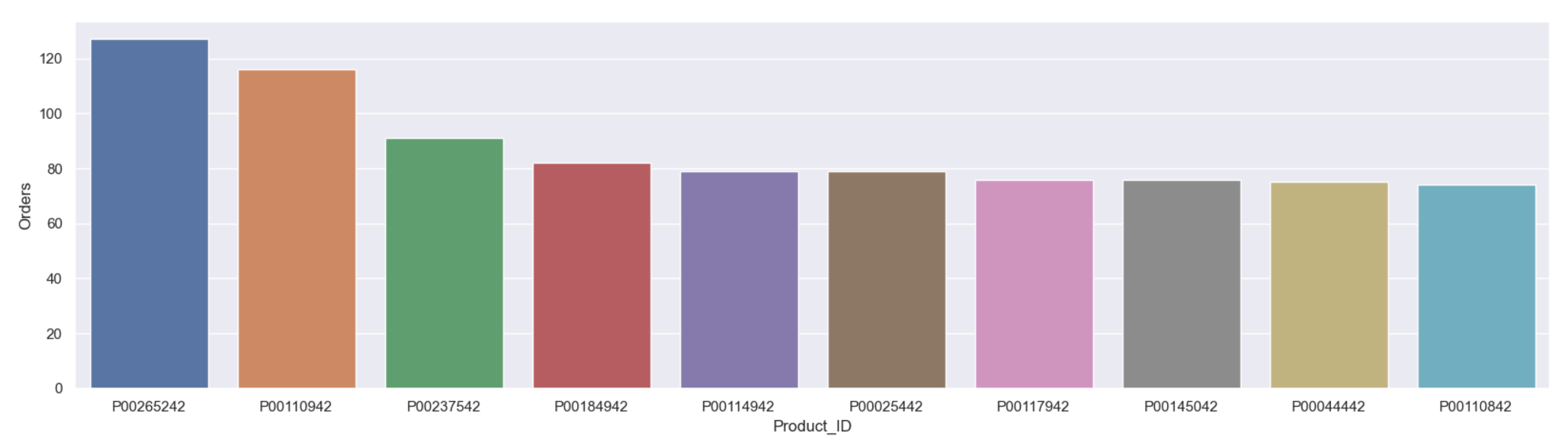
```
# Top selling product
```

In [109]:

```
pi=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=pi, x='Product_ID', y='Orders')
```

Out[109]:

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

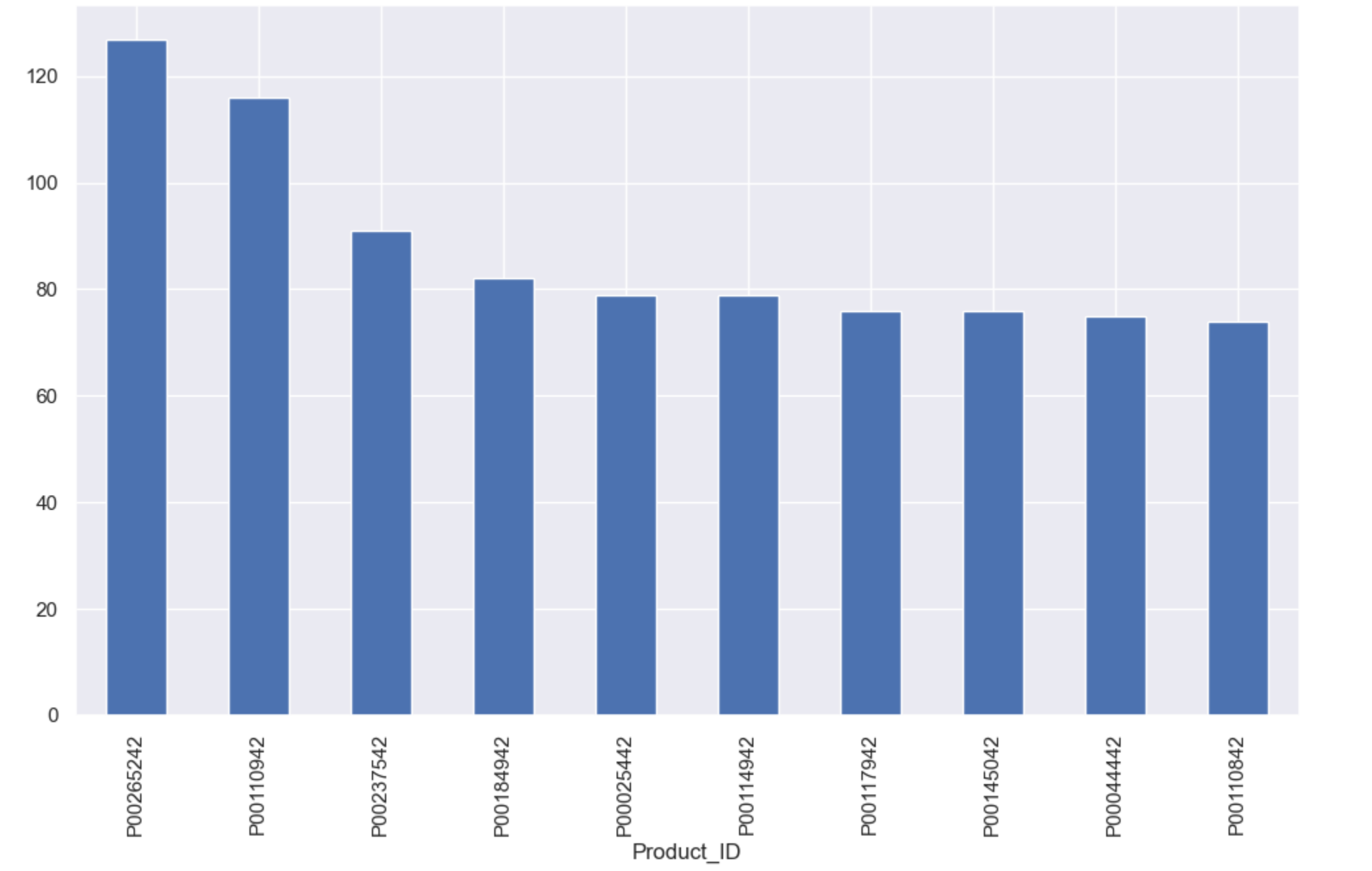


In [111]:

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

Out[111]:

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



Conclusion

Married women age group 26-35 yrs from UP, Maharashtra and Karnataka working in IT, Healthcare and Aviation are more likely buy products like food, clothing and electronics

In [ ]: