In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
df=pd.read_excel(r"C:\Users\HP\Desktop\Diwali Sales Data project.xlsx")
df
```

Out[2]:

	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 [3]:
```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 [4]:
```python
df.shape
```

Out[4]:
```
(11251, 15)
```

In [5]:
```python
df.head()
```

Out[5]:

	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

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

In [7]:
```python
df
```

Out[7]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0

11251 rows × 13 columns

In [8]:
```python
#DATA CLEANING
#check null value
df.isnull().sum()
```

Out[8]:
```
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 [9]:
```python
#fill null value by mean
df["Amount"]=df["Amount"].fillna(df['Amount'].mean())
```

In [10]:
```python
#checking null value is fill
df.isnull().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              0
dtype: int64
```

In [11]:
```python
df
```

Out[11]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0

11251 rows × 13 columns

In [12]:
```python
#Change Data type
df["Amount"]=df["Amount"].astype('int')
```

In [13]:
```python
df["Amount"].dtype
```

Out[13]:
```
dtype('int32')
```

In [14]:
```python
#to check column name
df.columns
```

Out[14]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [15]:
```python
df1=df.rename(columns={'Gender':'Sex'})
df1
```

Out[15]:

	User_ID	Cust_name	Product_ID	Sex	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
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

11251 rows × 13 columns

In [16]:
```python
#describe function return the description of the data in Data frame(i.e count,mean,std etc)
df.describe()
```

Out[16]:

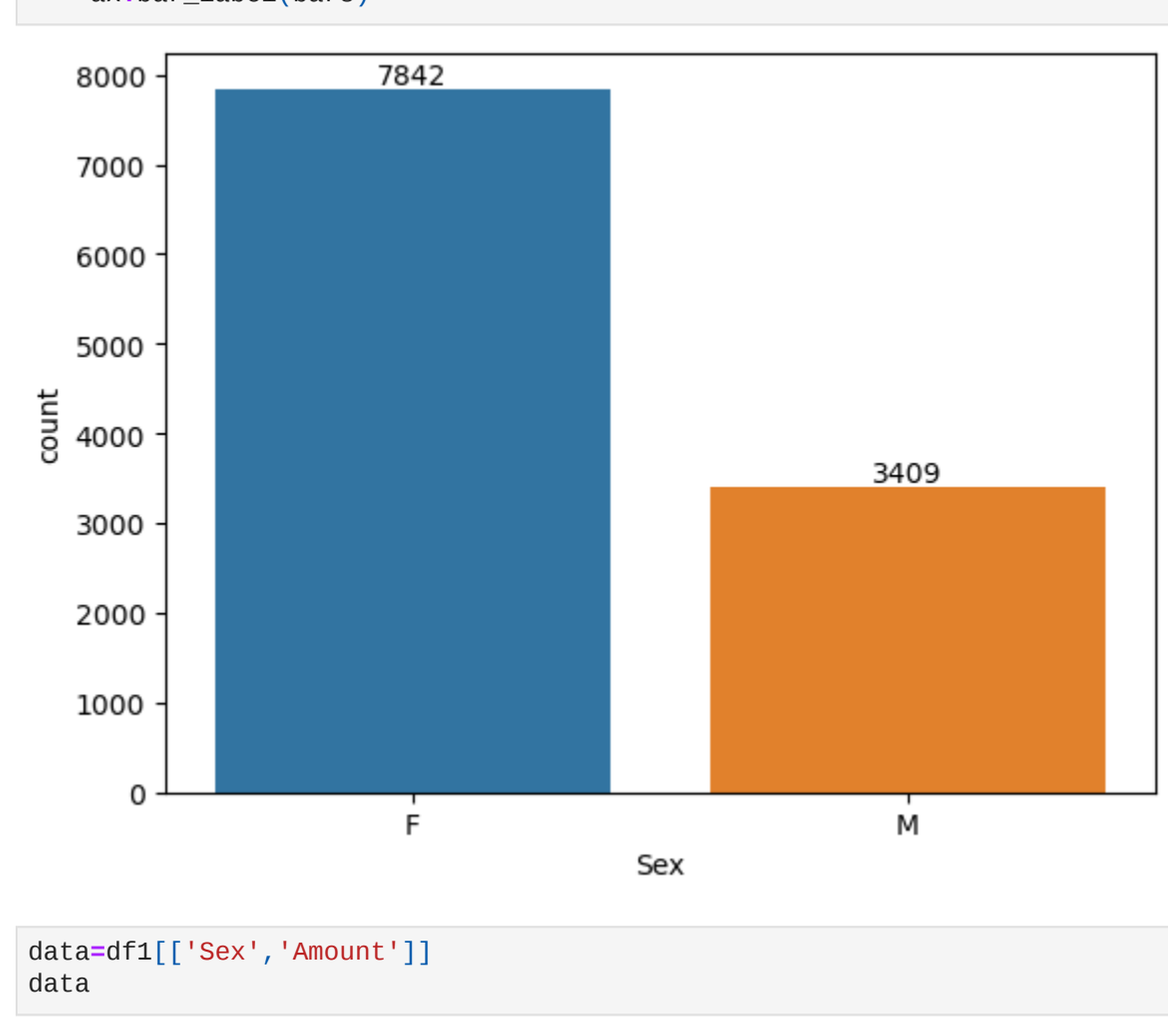
	User_ID	Age	Marital_Status	Orders	Amount
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11251.000000
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.609901
std	1.716125e+03	12.754122	0.493632	1.115047	5219.569169
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.500000
50%	1.003065e+06	33.000000	0.000000	2.000000	8110.000000
75%	1.004430e+06	43.000000	1.000000	3.000000	12671.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

In [17]:
```python
#describe function for specific column
df[['Age','Orders','Amount']].describe()
```

Out[17]:

	Age	Orders	Amount
count	11251.000000	11251.000000	11251.000000
mean	35.421207	2.489290	9453.609901
std	12.754122	1.115047	5219.569169
min	12.000000	1.000000	188.000000
25%	27.000000	1.500000	5443.500000
50%	33.000000	2.000000	8110.000000
75%	43.000000	3.000000	12671.000000
max	92.000000	4.000000	23952.000000

In [18]:
```python
ax=sns.countplot(x='Sex',data=df1)
#for data label
for bars in ax.containers:
    ax.bar_label(bars)
```



In [19]:
```python
data=df1[['Sex','Amount']]
data
```

Out[19]:

	Sex	Amount
0	F	23952
1	F	23934
2	F	23924
3	M	23912
4	M	23877
...	...	...
11246	M	370
11247	M	367
11248	F	213
11249	M	206
11250	F	188

11251 rows × 2 columns

In [20]:
```python
data.groupby('Sex').sum()
```

Out[20]:

Sex	Amount
F	74430383
M	31932182

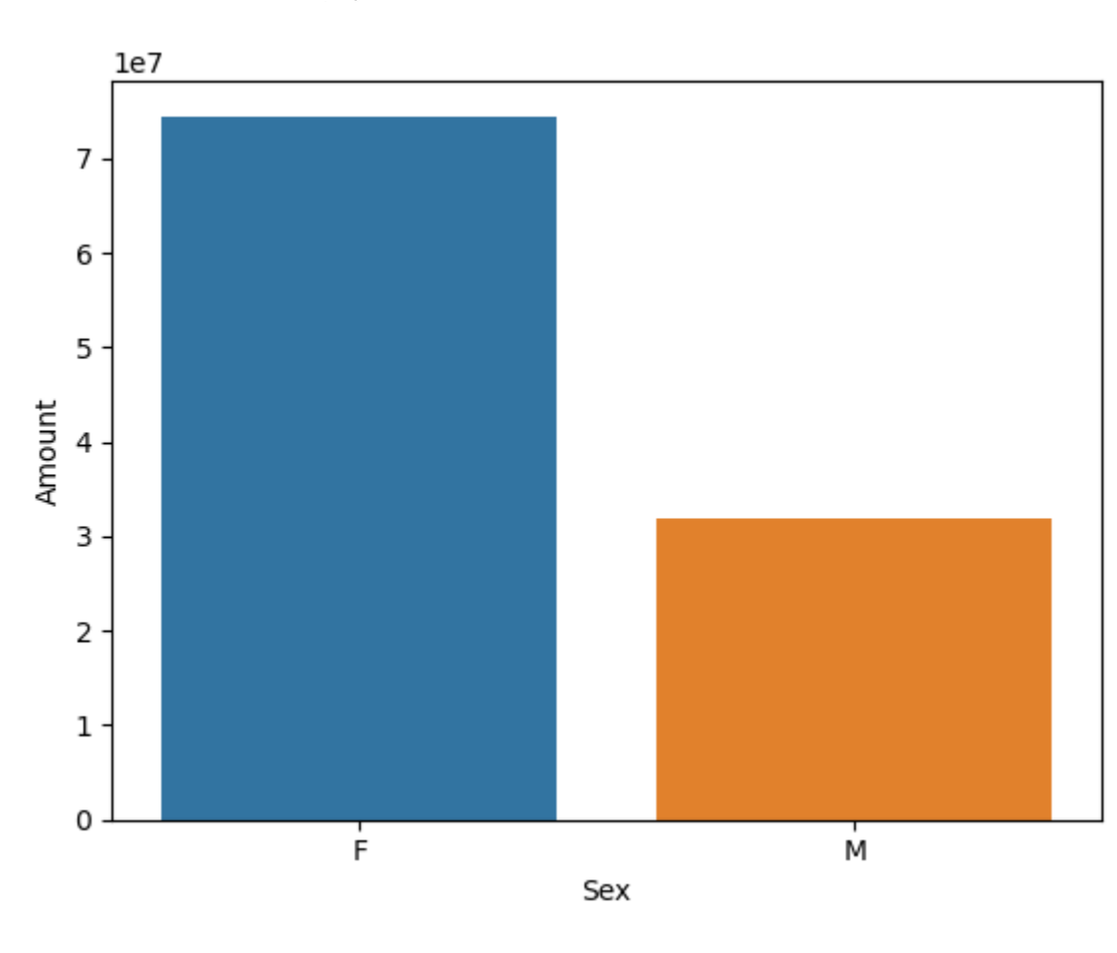
In [21]:
```python
ss=df1.groupby(['Sex'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
ss
```

Out[21]:

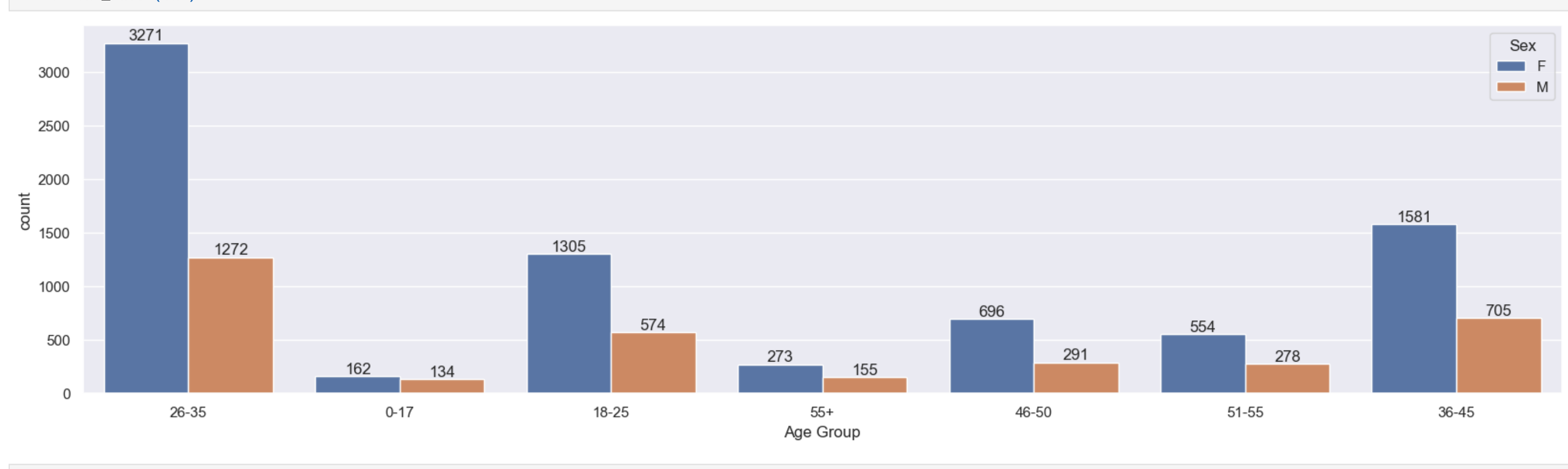
	Sex	Amount
0	F	74430383
1	M	31932182

In [22]:
```python
sns.barplot(x= 'Sex' ,y= 'Amount' ,data=ss)
```

Out[22]:
```
<Axes: xlabel='Sex', ylabel='Amount'>
```



In [68]:
```python
bx=sns.countplot(x='Age Group',data=df1,hue="Sex")
for bar in bx.containers:
    bx.bar_label(bar)
```



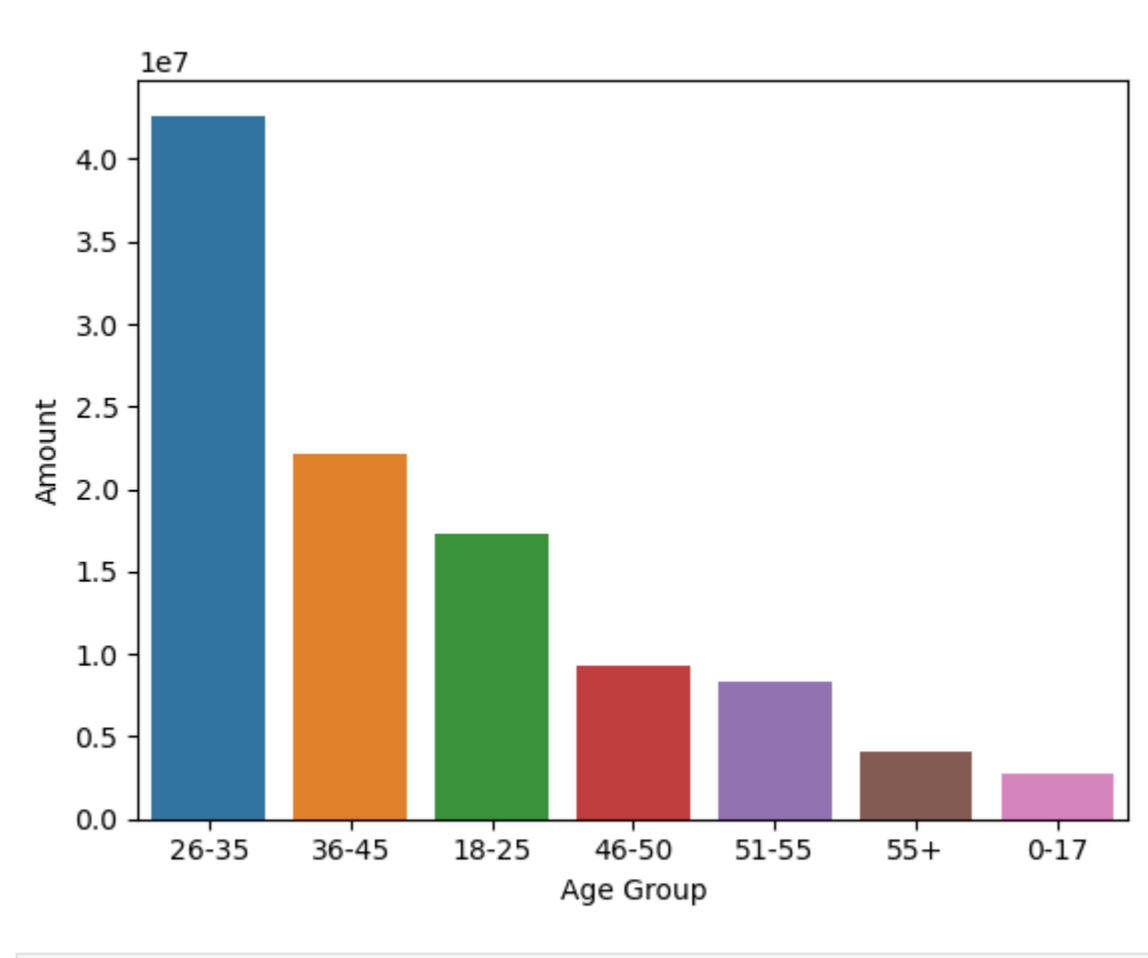
In [24]:
```python
#Total Amount vs Age Group
sales_age=df1.groupby(['Age Group'],as_index= False)['Amount'].sum().sort_values(by="Amount",ascending=False)
sales_age
```

Out[24]:

	Age Group	Amount
2	26-35	42632348
3	36-45	22173353
1	18-25	17240732
4	46-50	9245656
5	51-55	8280383
6	55+	4090440
0	0-17	2699653

In [25]:
```python
sns.barplot(x="Age Group",y="Amount", data=sales_age)
```

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



In [26]:
```python
#Total no. of order from 10 states
statewise=df1[["State","Orders"]]
statewise
```

Out[26]:

	State	Orders
0	Maharashtra	1
1	Andhra Pradesh	3
2	Uttar Pradesh	3
3	Karnataka	2
4	Gujarat	2
...	...	...
11246	Maharashtra	4
11247	Haryana	3
11248	Madhya Pradesh	4
11249	Karnataka	3
11250	Maharashtra	3

11251 rows × 2 columns

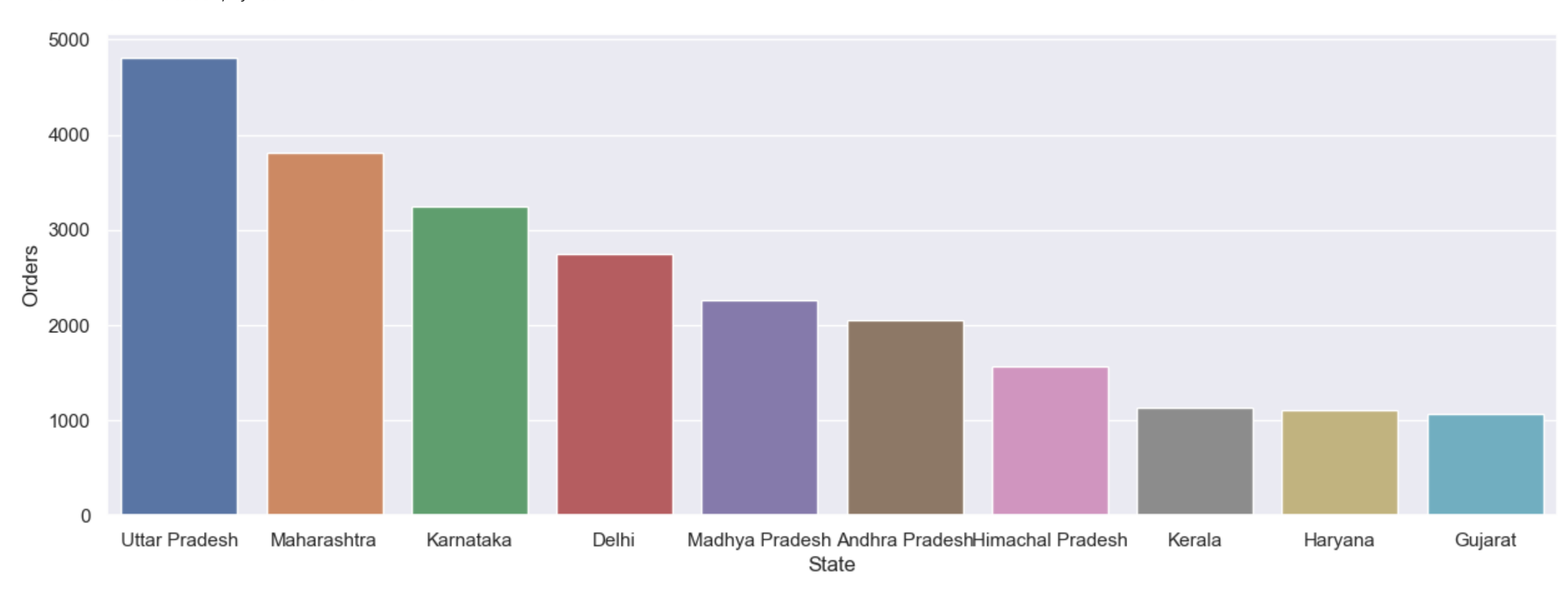
In [27]:
```python
statewise_order=df1.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False)[0:10]
statewise_order
```

Out[27]:

	State	Orders
14	Uttar Pradesh	4813
10	Maharashtra	3811
7	Karnataka	3241
2	Delhi	2744
9	Madhya Pradesh	2259
0	Andhra Pradesh	2054
5	Himachal Pradesh	1568
8	Kerala	1137
4	Haryana	1109
3	Gujarat	1070

In [28]:
```python
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x="State",y="Orders",data=statewise_order)
```

Out[28]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



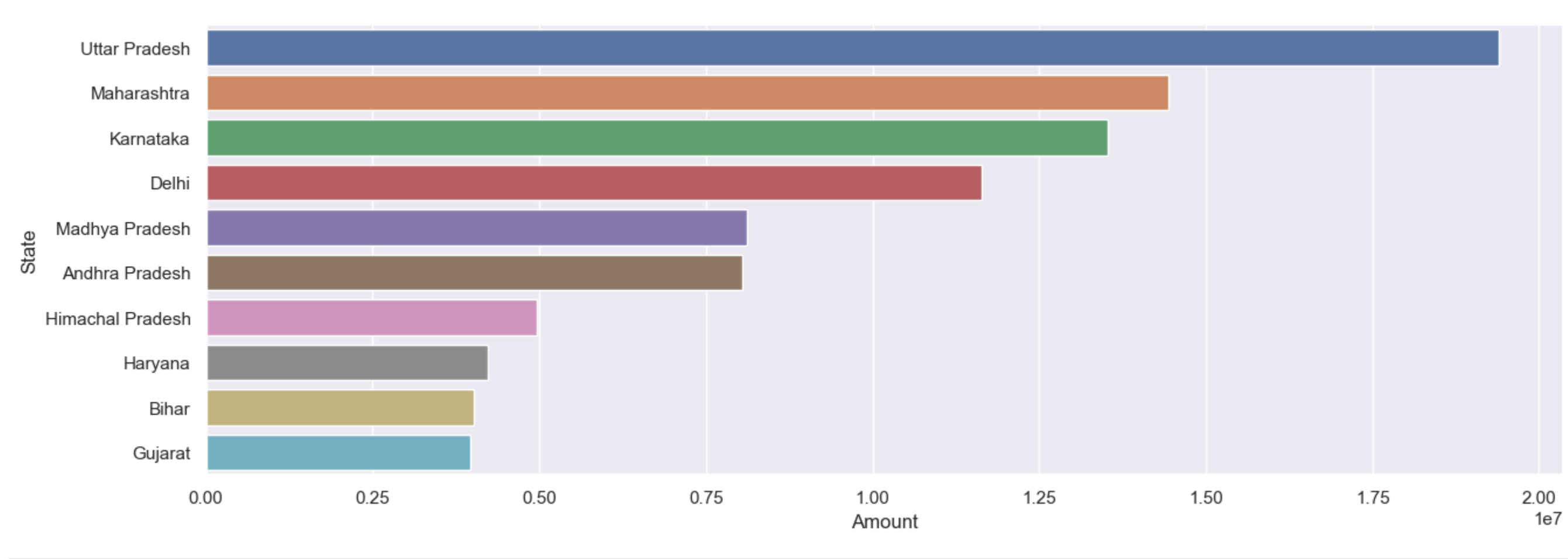
In [29]:
```python
#Sales from top 10 states
top_state_sales=df1.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)[0:10]
top_state_sales
```

Out[29]:

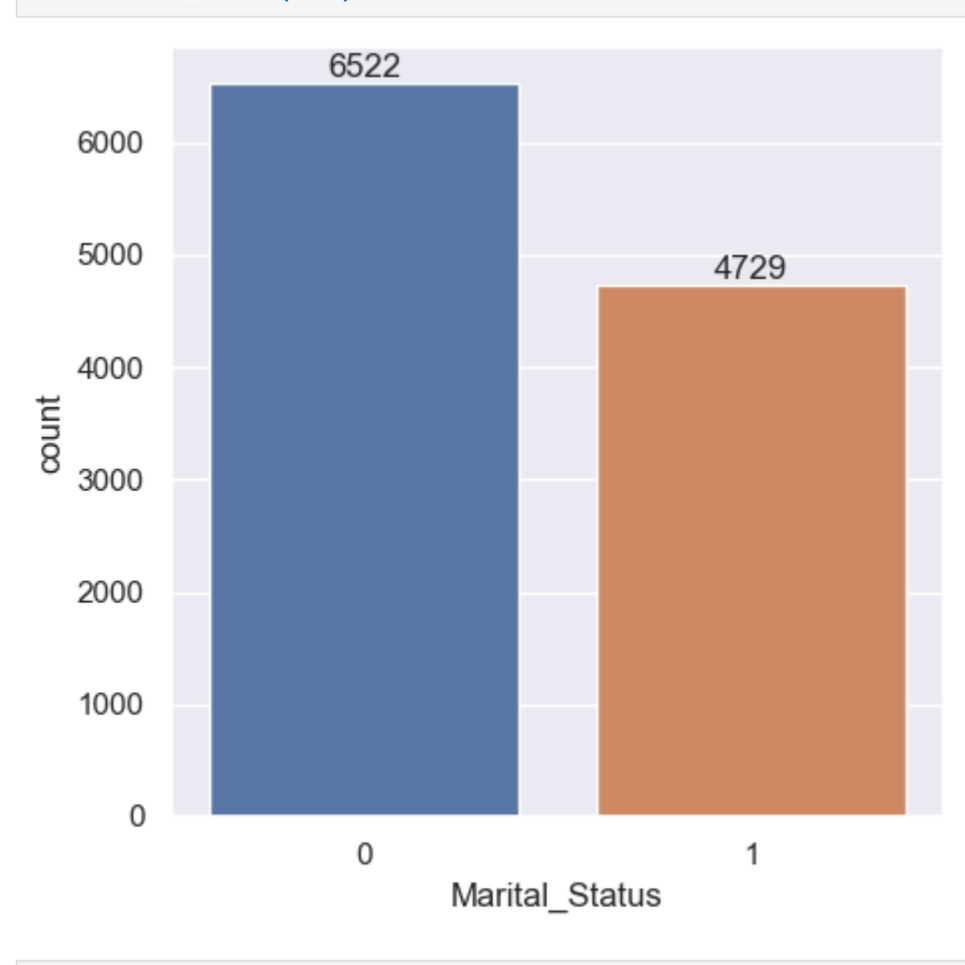
	State	Amount
14	Uttar Pradesh	19393874
10	Maharashtra	14436996
7	Karnataka	13532993
2	Delhi	11632177
9	Madhya Pradesh	8120048
0	Andhra Pradesh	8046599
5	Himachal Pradesh	4963368
4	Haryana	4220175
1	Bihar	4022757
3	Gujarat	3964988

In [30]:
```python
sns.barplot(x='Amount',y='State', data=top_state_sales)
```

Out[30]:
```
<Axes: xlabel='Amount', ylabel='State'>
```



In [38]:
```python
sns.set(rc={'figure.figsize':(5,5)})
ax=sns.countplot(x='Marital_Status',data=df1)
for bar in ax.containers:
    ax.bar_label(bar)
```



In [44]:
```python
df1
```

Out[44]:

	User_ID	Cust_name	Product_ID	Sex	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
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

11251 rows × 13 columns

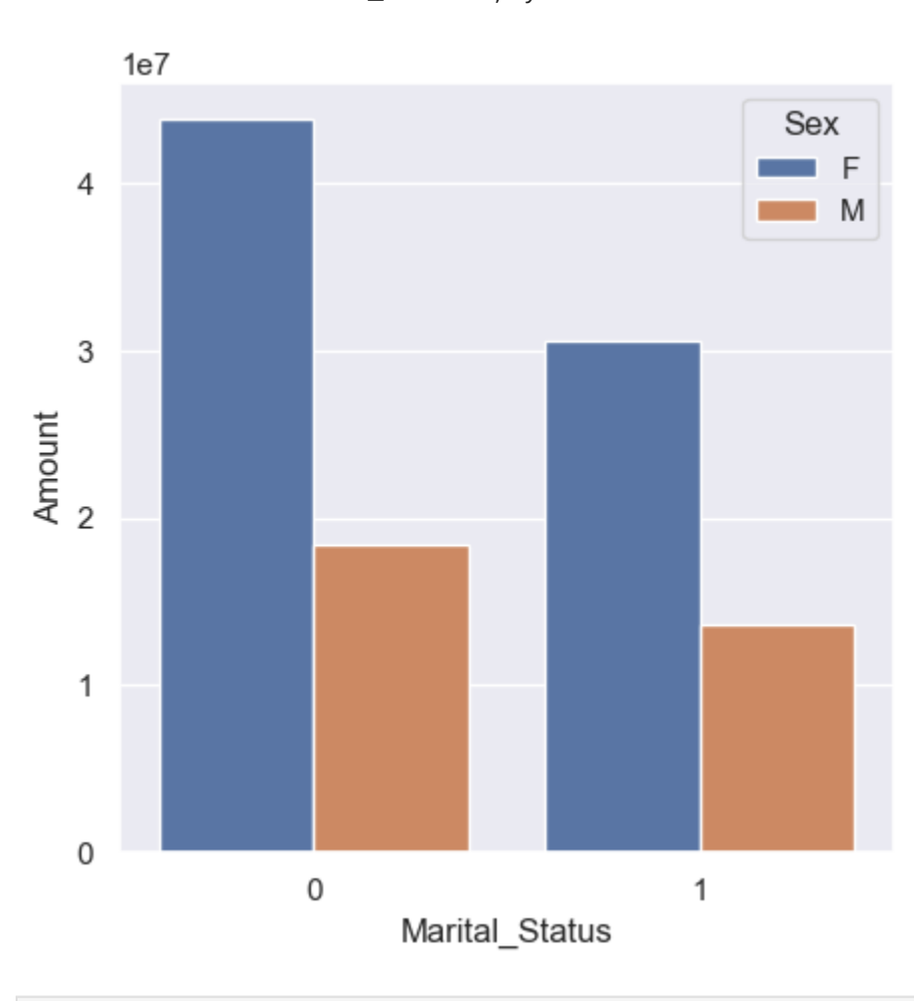
In [55]:
```python
#amount according to marital status with sex
sss=df1.groupby(['Marital_Status','Sex'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sss
```

Out[55]:

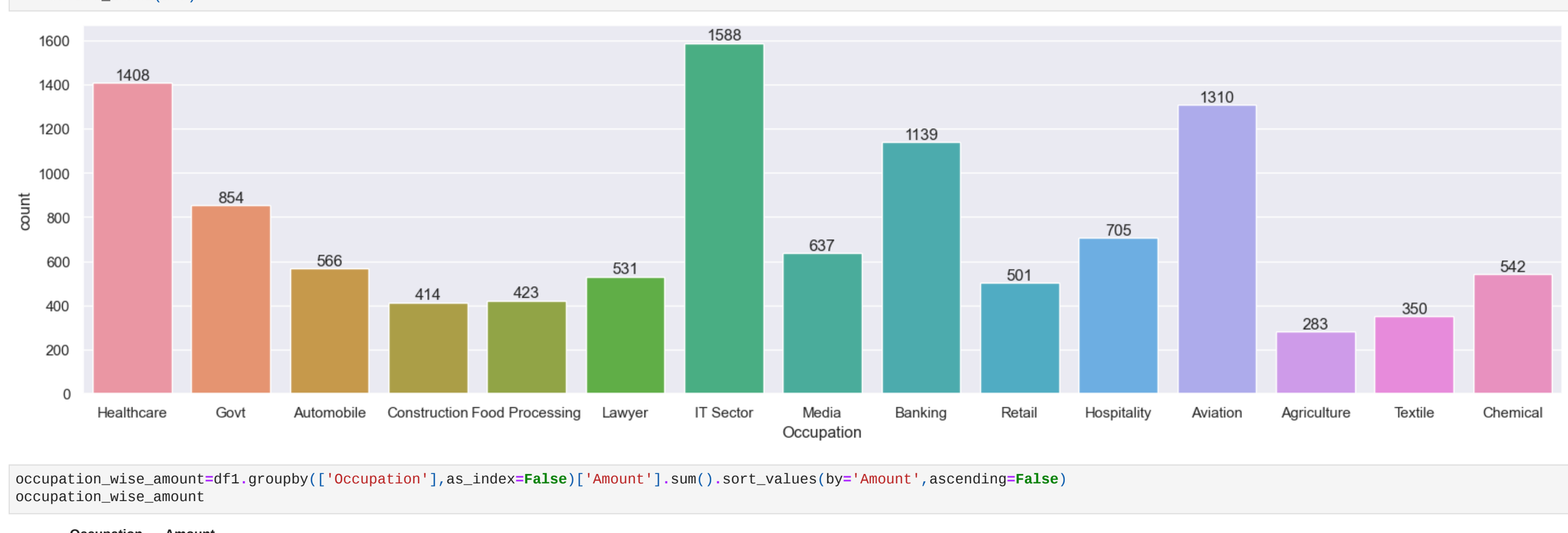
	Marital_Status	Sex	Amount
0	0	F	43815005
2	1	F	30615378
1	0	M	18348191
3	1	M	13583991

In [53]:
```python
sns.barplot(x='Marital_Status',data=sss,y='Amount',hue='Sex')
```

Out[53]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



In [63]:
```python
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(x='Occupation',data=df1)
for bar in ax.containers:
    ax.bar_label(bar)
```



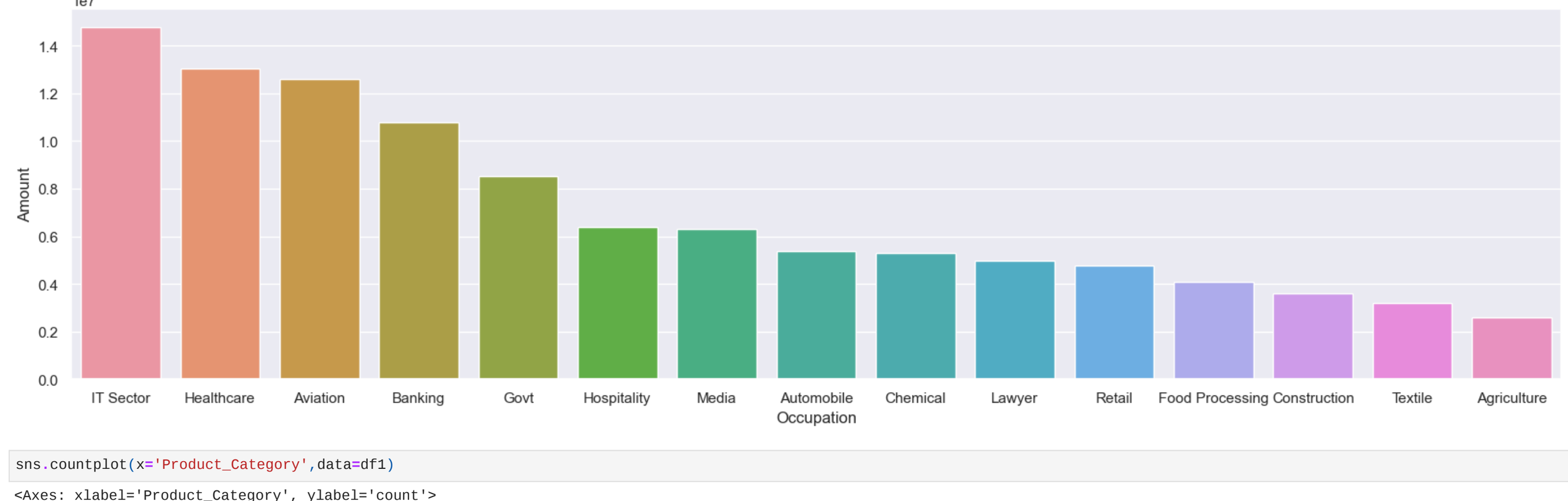
In [66]:
```python
occupation_wise_amount=df1.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
occupation_wise_amount
```

Out[66]:

	Occupation	Amount
10	IT Sector	14802344
8	Healthcare	13034586
2	Aviation	12602298
3	Banking	10789516
7	Govt	8517212
9	Hospitality	6395311
12	Media	6295832
1	Automobile	5378049
4	Chemical	5306889
11	Lawyer	4981665
13	Retail	4783170
6	Food Processing	4070670
5	Construction	3597511
14	Textile	3214425
0	Agriculture	2593087

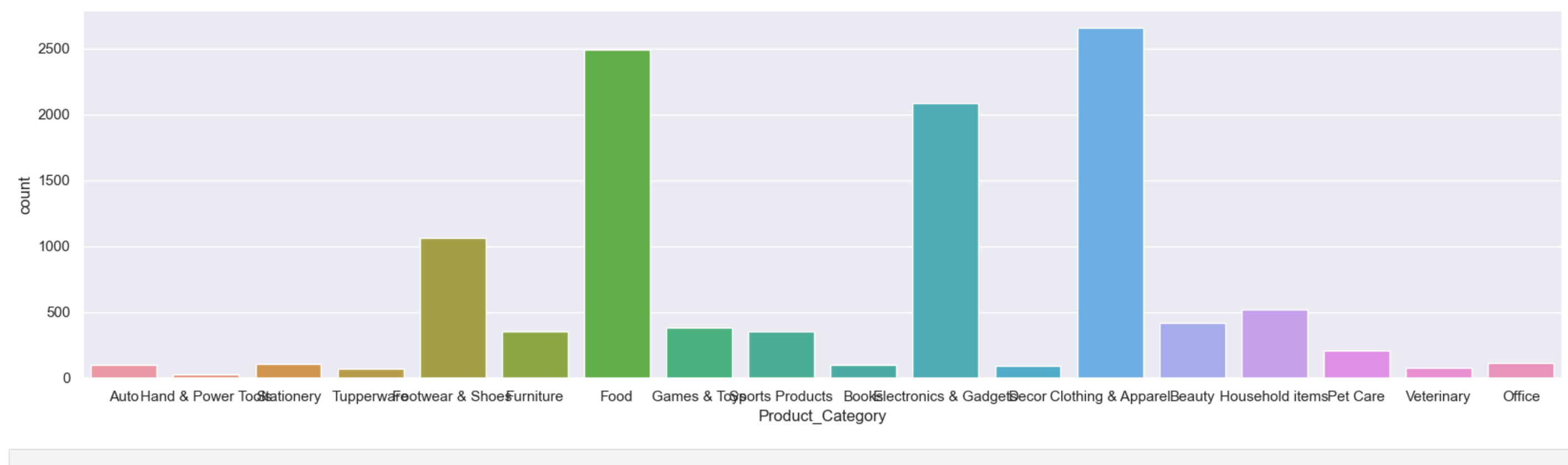
In [67]:
```python
sns.barplot(x='Occupation',y='Amount',data=occupation_wise_amount)
```

Out[67]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



In [71]:
```python
sns.countplot(x='Product_Category',data=df1)
```

Out[71]:
```
<Axes: xlabel='Product_Category', ylabel='count'>
```



In [ ]: