In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:
```
df = pd.read_csv('Diwali Sales Data.csv' ,encoding= 'unicode_escape')
```

In [11]:
```
df.shape
```

Out[11]:
```
(11251, 15)
```

In [13]:
```
df.head()
```

Out[13]:

	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 [14]:
```
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 [16]:
```
df.drop(['Status', 'unnamed1'], axis= 1 , inplace=True)
```

In [17]:
```
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
```

In [21]:
```
pd.isnull(df).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              12
dtype: int64
```

In [22]:
```
df.dropna(inplace= True)
```

In [26]:
```
pd.isnull(df).sum()
```

Out[26]:
```
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 [27]:
```
df['Amount']= df['Amount'].astype('int')
```

In [28]:
```
df['Amount'].dtype
```

Out[28]:
```
dtype('int32')
```

In [29]:
```
df.columns
```

Out[29]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [31]:
```
df.rename(columns={'Cust_name':'Name'})
```

Out[31]:

	User_ID	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
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

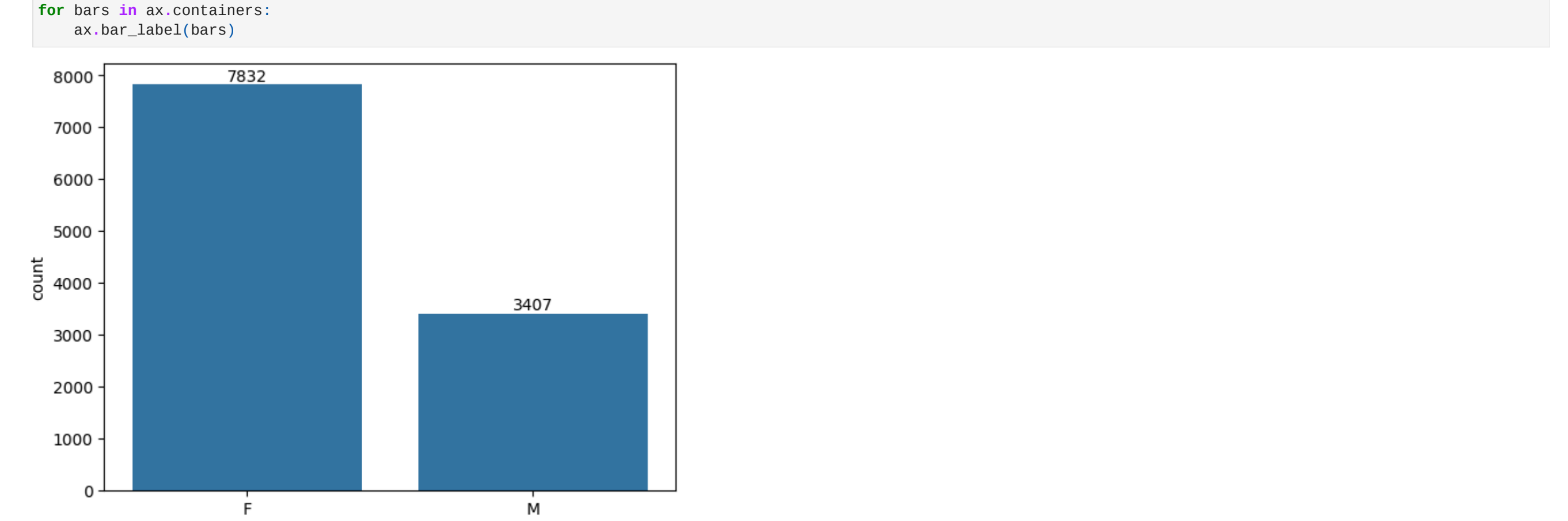
In [32]:
```
df.describe()
```

Out[32]:

	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

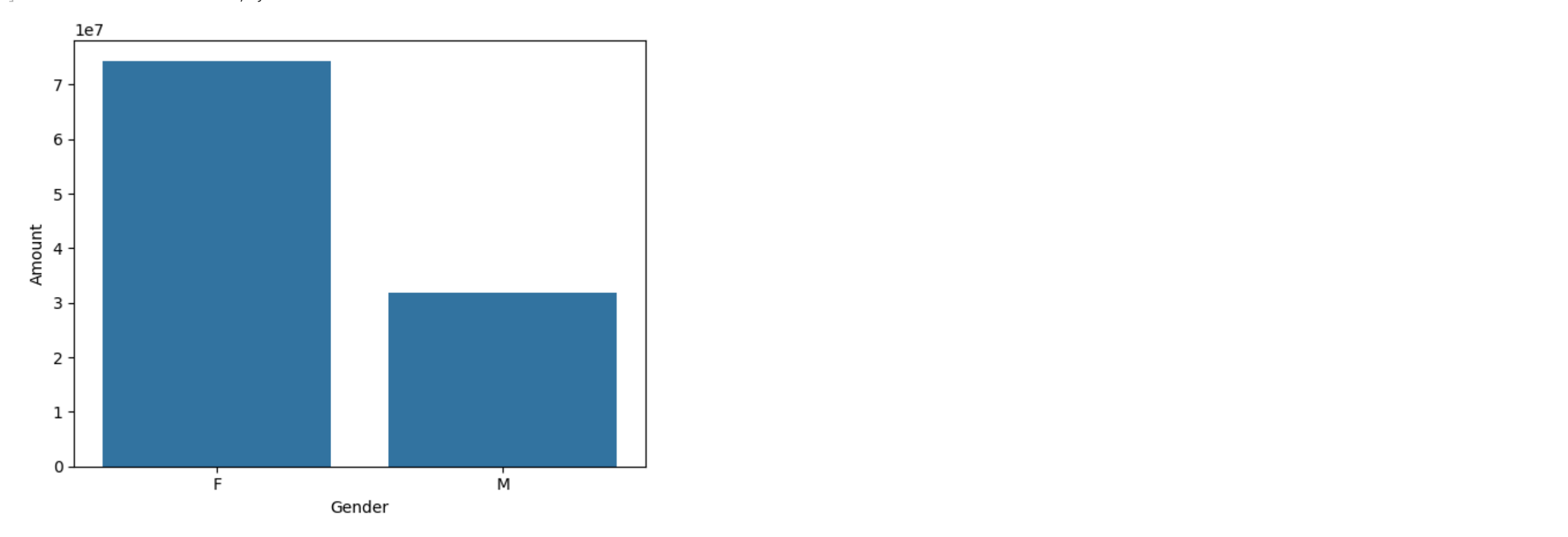
exploratory data analysis

In [44]:
```
ax = sns.countplot(x='Gender', data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```



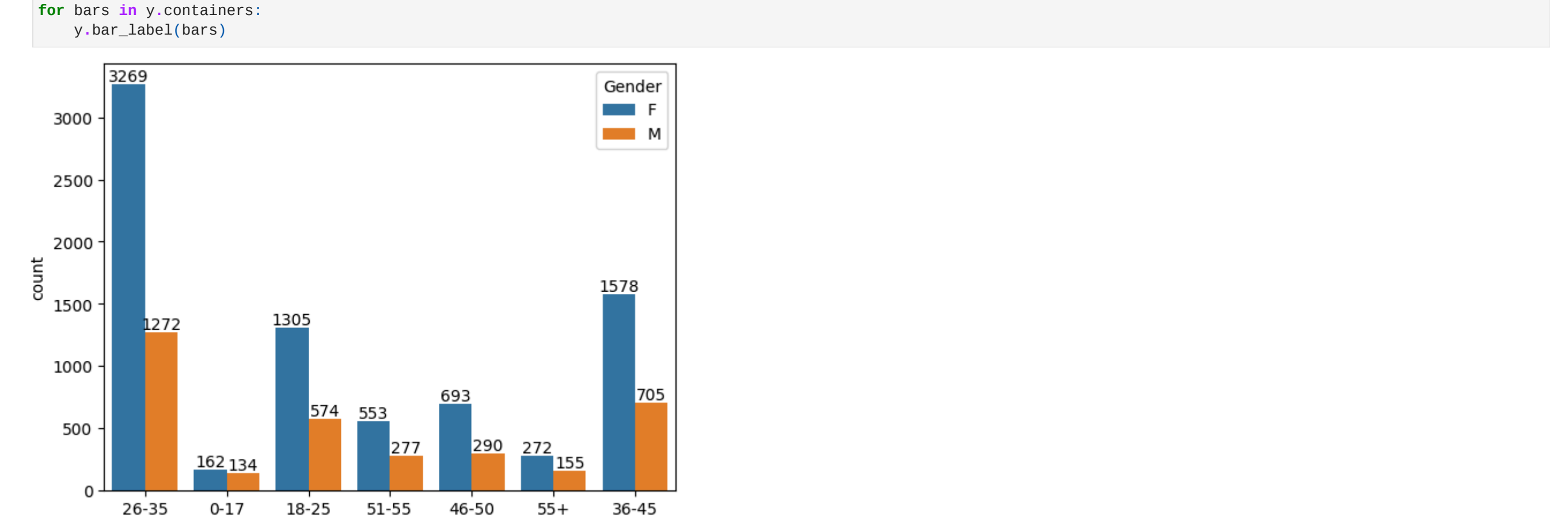
In [73]:
```
gen_amt = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending= False)
sns.barplot(x='Gender',y='Amount', data=gen_amt)
```

Out[73]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



From above barchart we can see that women have spent more money than men and have more purchasing power

In [84]:
```
y = sns.countplot(data=df, x='Age Group', hue='Gender')
for bars in y.containers:
    y.bar_label(bars)
```

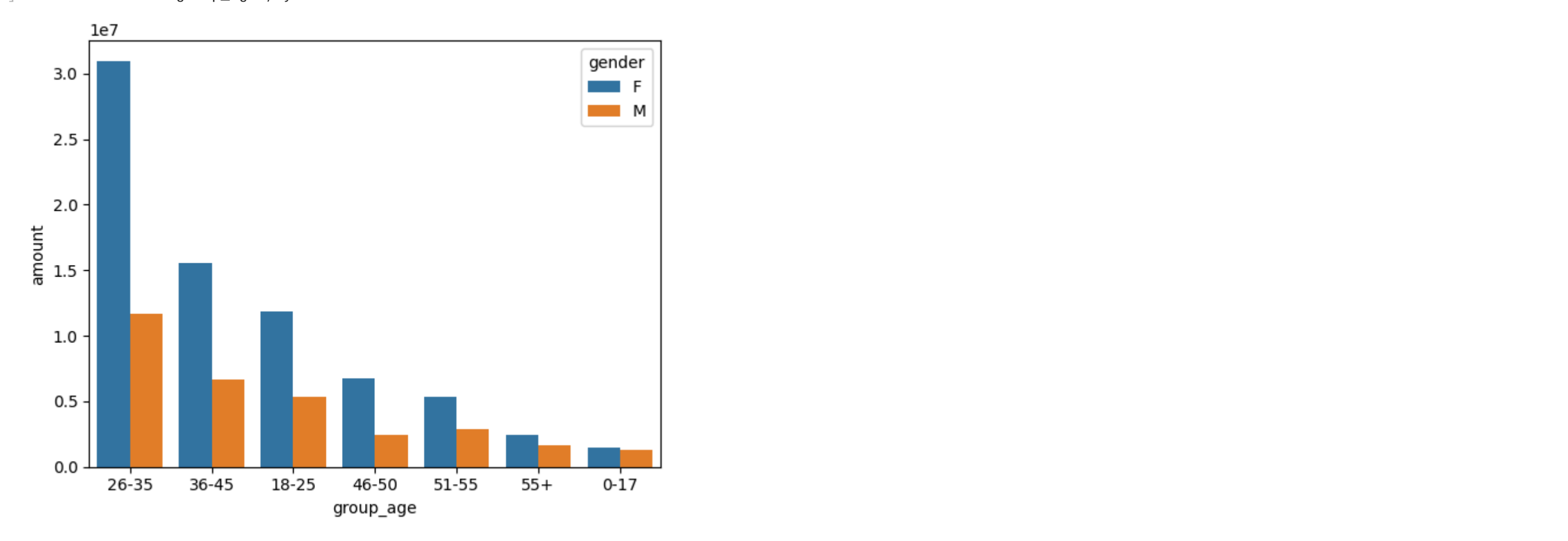


In [6]:
```
df.columns
```

Out[6]:
```
Index(['user_id', 'cust_name', 'product_id', 'gender', 'group_age', 'age',
       'marital_status', 'state', 'zone', 'occupation', 'product_category',
       'orders', 'amount'],
      dtype='object')
```

In [13]:
```
age_amt = df.groupby(['group_age','gender'], as_index=False)['amount'].sum().sort_values(by='amount', ascending= False)
sns.barplot(x='group_age',y='amount', data=age_amt, hue='gender')
```

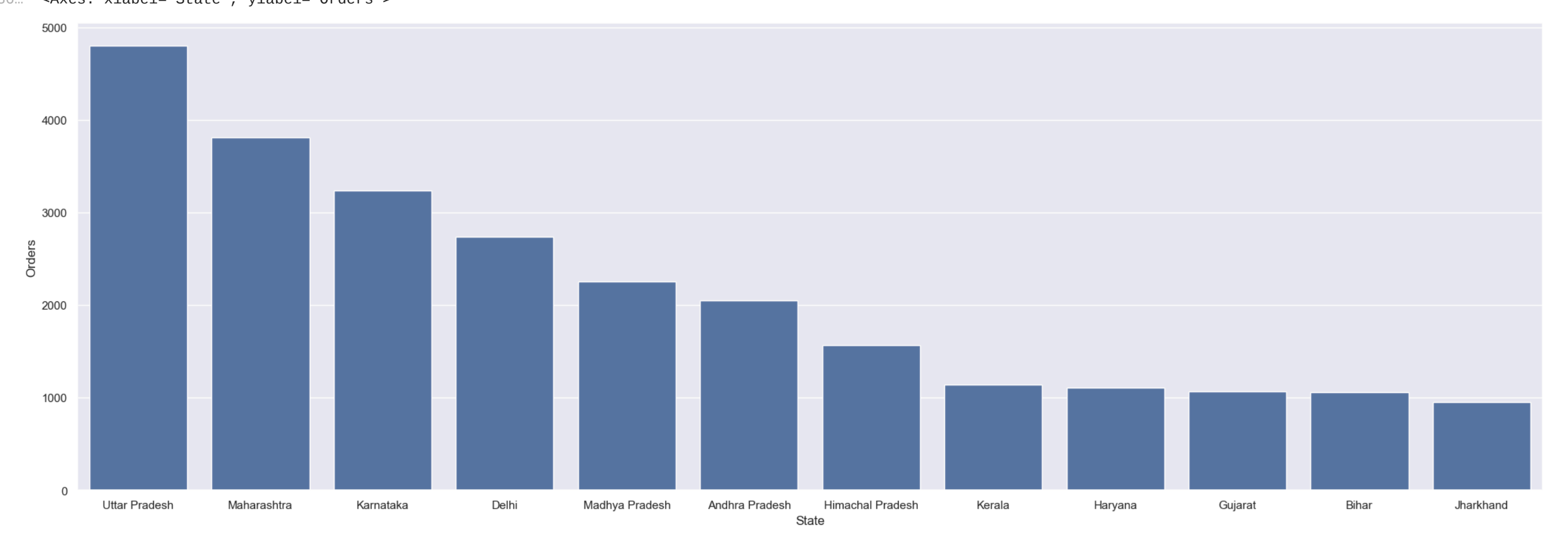
Out[13]:
```
<Axes: xlabel='group_age', ylabel='amount'>
```



from above graph we can see that female of age group 26-35 have high spending capacity

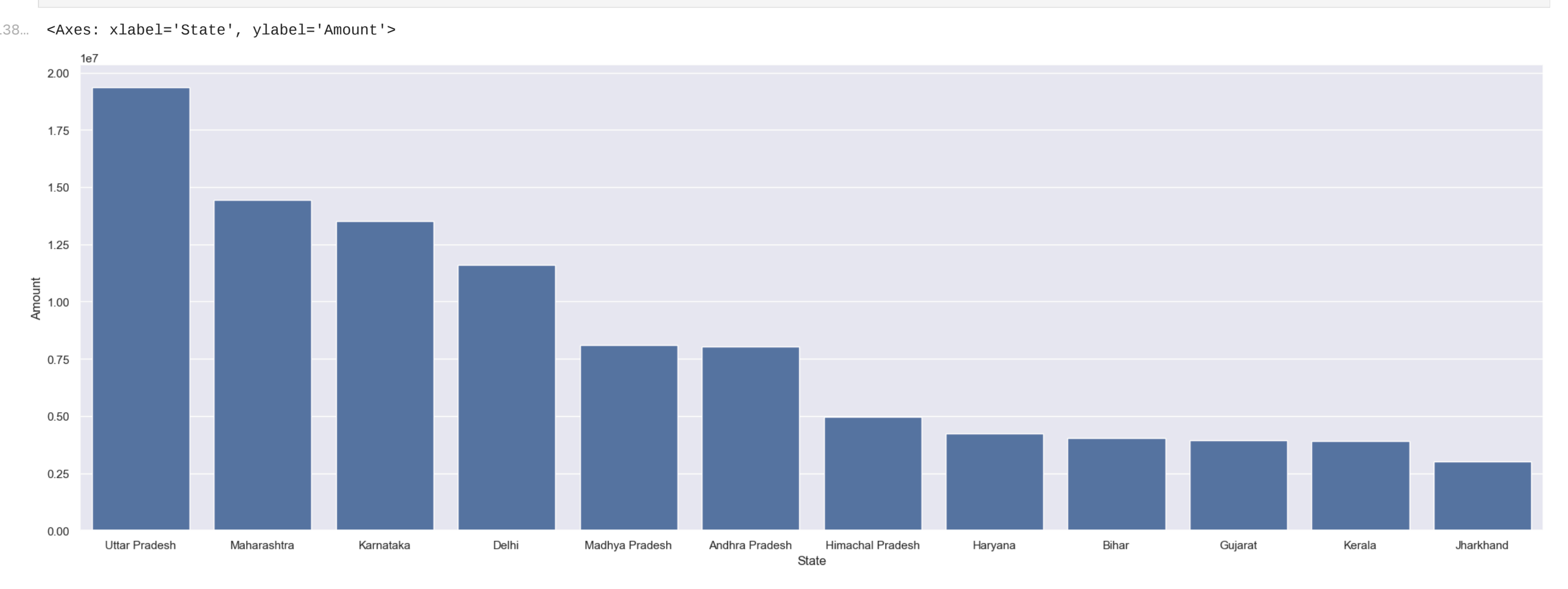
In [136…
```
state_order = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders',ascending= False).head(12)
sns.set(rc={'figure.figsize':(25,8)})
sns.barplot(data=state_order, x="State", y='Orders')
```

Out[136…
```
<Axes: xlabel='State', ylabel='Orders'>
```



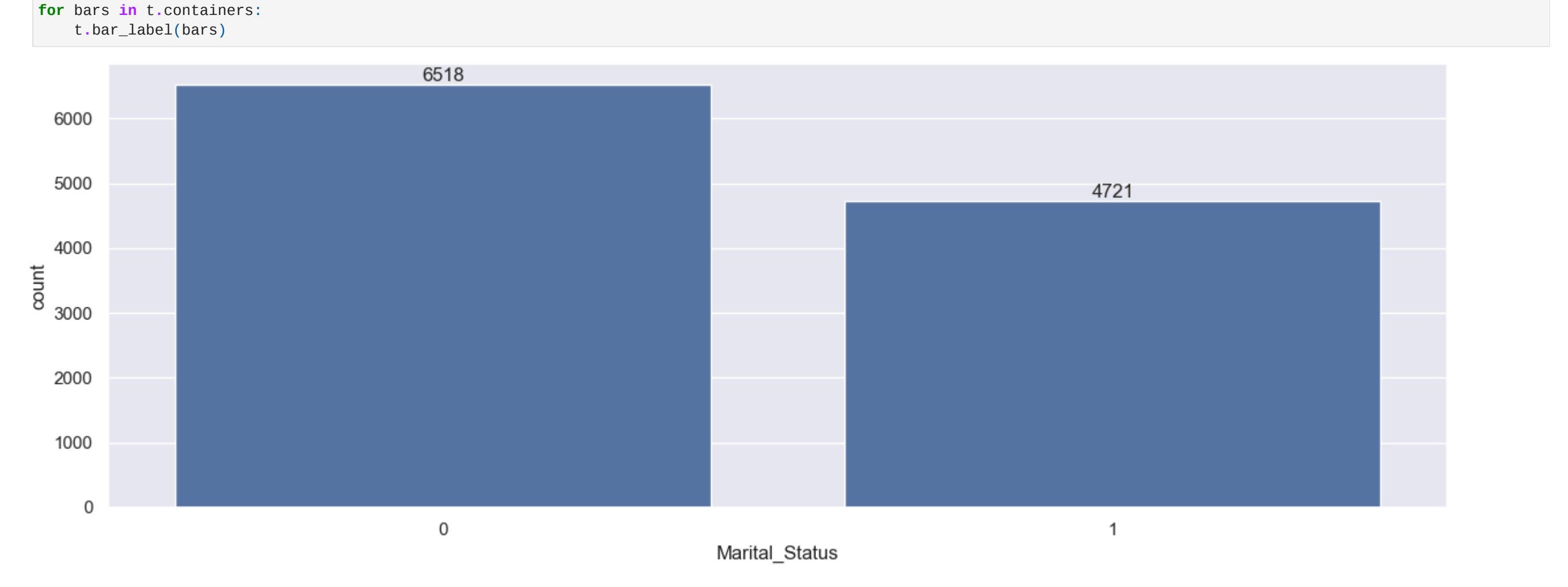
In [138…
```
state_amt = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending= False).head(12)
sns.set(rc={'figure.figsize':(25,8)})
sns.barplot(data=state_amt, x="State", y='Amount')
```

Out[138…
```
<Axes: xlabel='State', ylabel='Amount'>
```



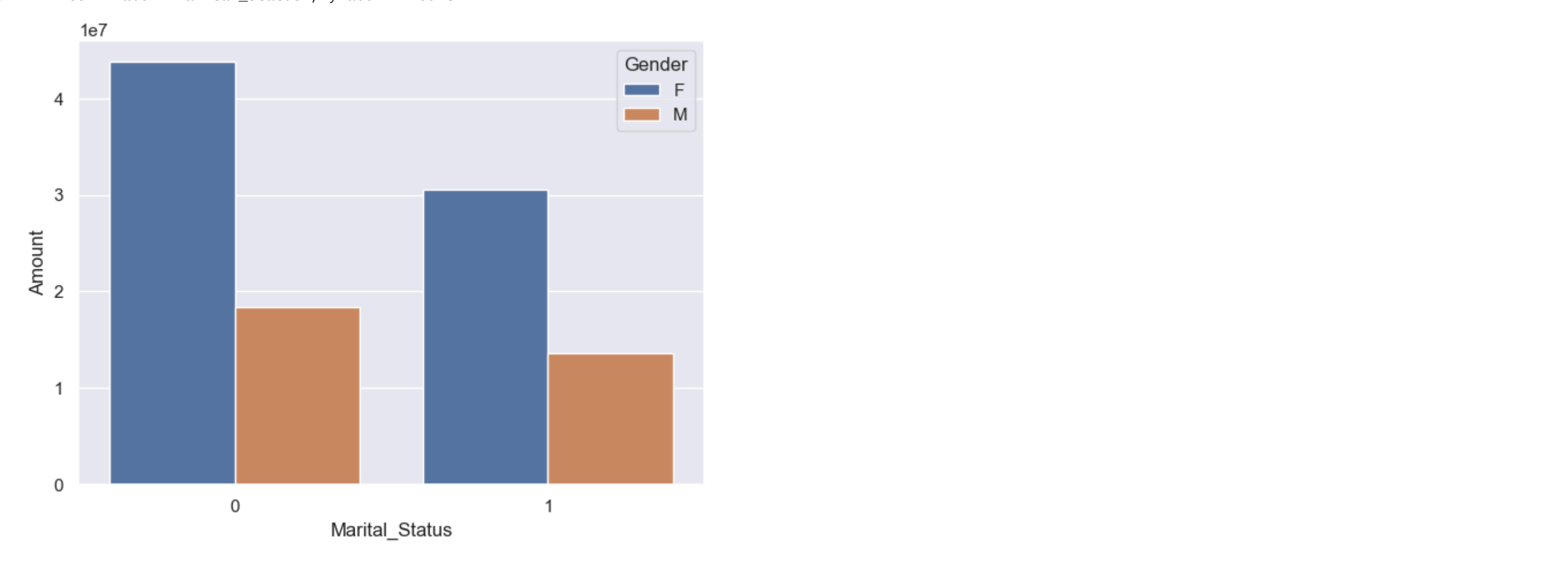
From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

In [155…
```
t = sns.countplot(data=df, x= 'Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in t.containers:
    t.bar_label(bars)
```



In [157…
```
marr_amt = df.groupby(['Marital_Status','Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending= False)
sns.barplot(x='Marital_Status',y='Amount', data=marr_amt, hue='Gender')
```

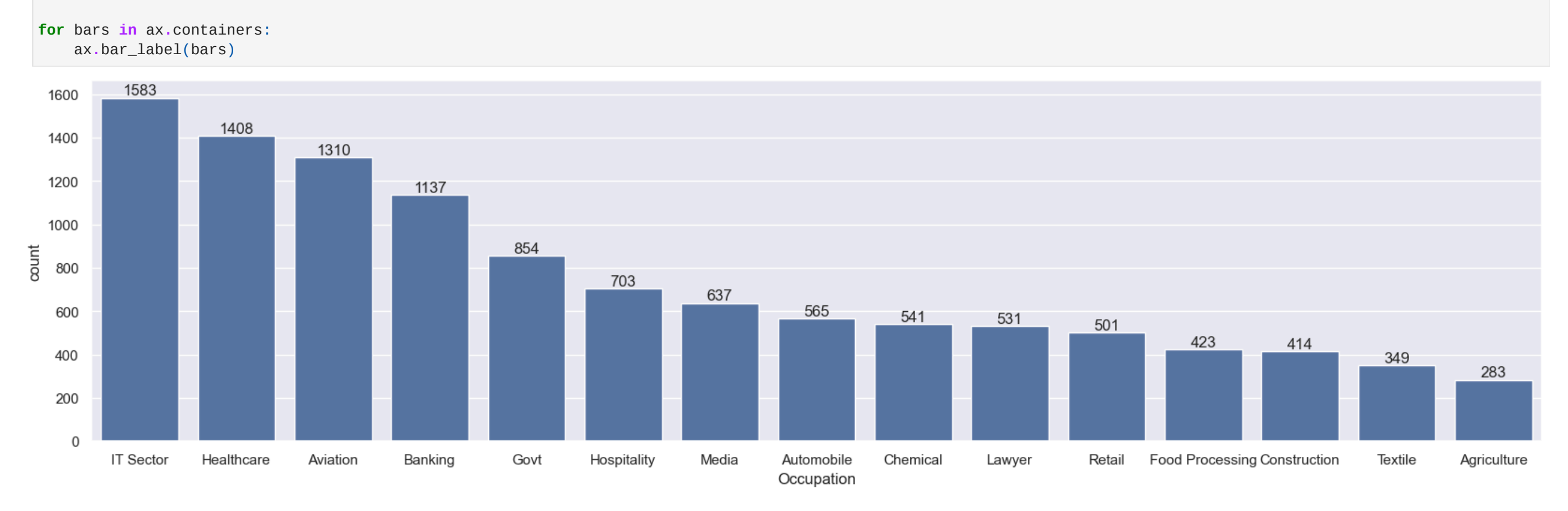
Out[157…
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



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

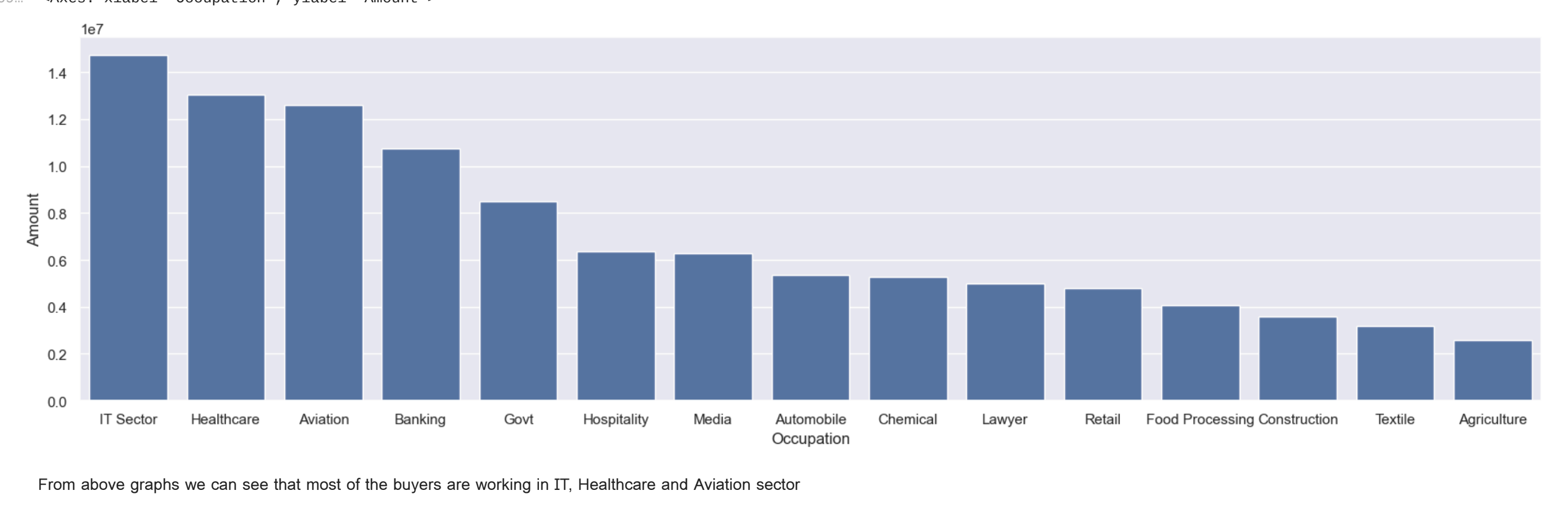
In [168…
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation', order=df['Occupation'].value_counts(ascending=False).index)

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



In [169…
```
occ_amt = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.barplot(data=occ_amt, x='Occupation', y ='Amount')
```

Out[169…
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



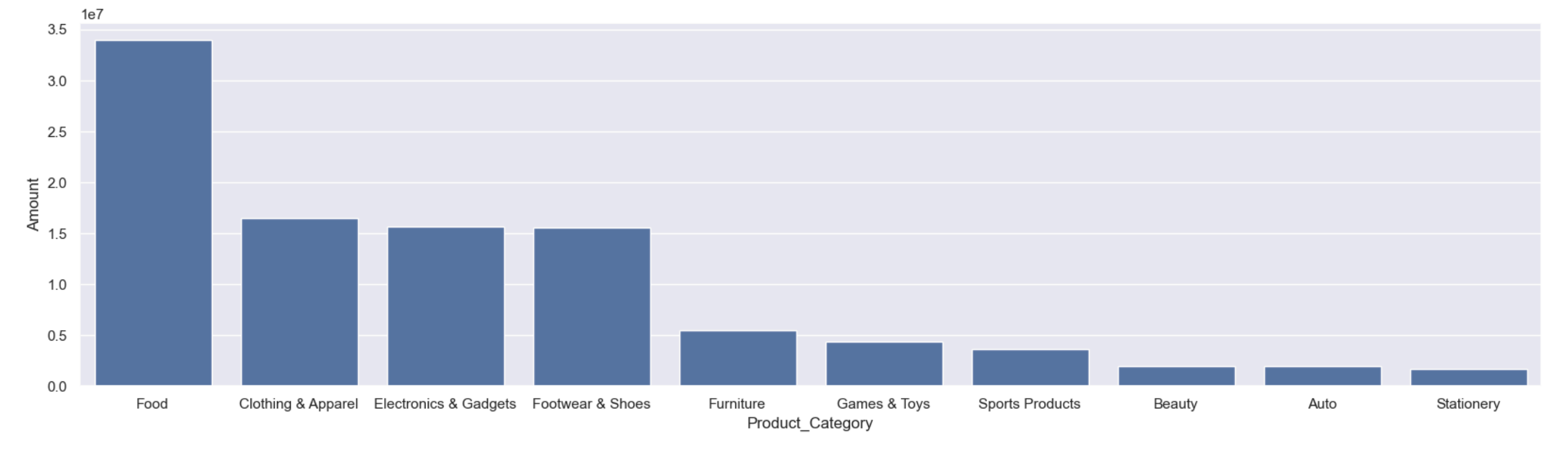
From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

product category

In [173…
```
df.columns
```

Out[173…
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [189…
```
pdt_amt = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
sns.barplot(data=pdt_amt, x='Product_Category', y ='Amount')
sns.set(rc={'figure.figsize':(28,5)})
```

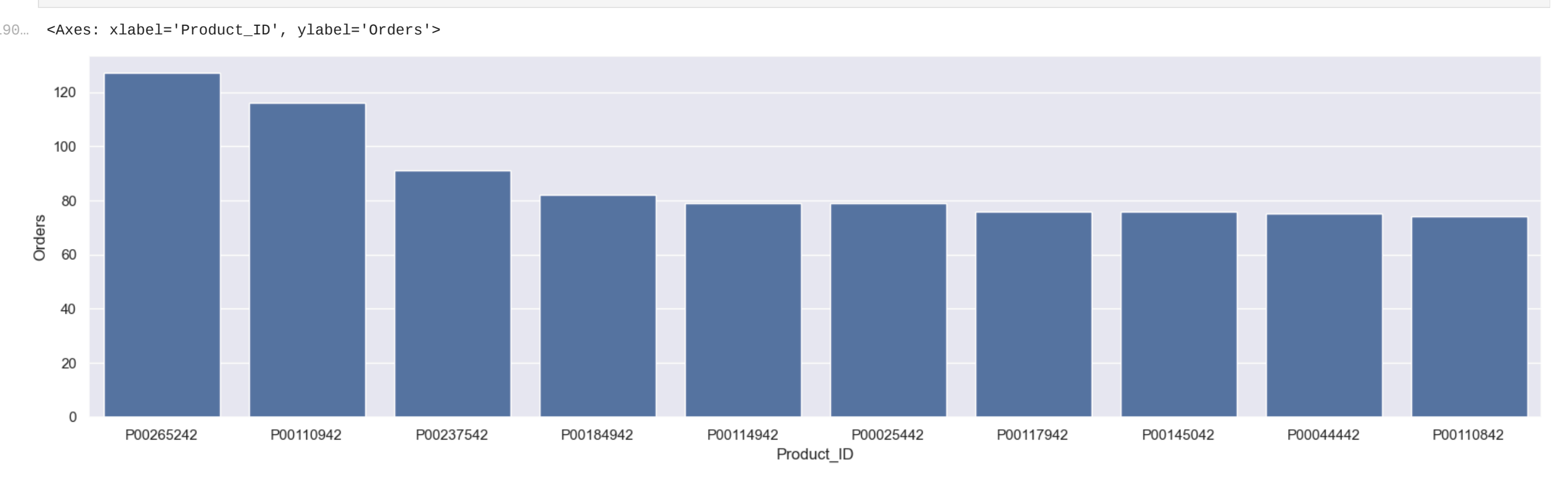


From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

In [190…
```
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[190…
```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```



Conclusion: Married women 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

In [ ]: