In [40]:
```python
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```python
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode escape')
```

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

Out[3]:

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

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	23952.0	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 [5]:
```python
df.tail()
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
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

In [23]:
```python
df.shape
```

Out[23]:
```
(11239, 13)
```

In [25]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           11239 non-null  int64
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender            11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64
 6   Marital_Status    11239 non-null  int64
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64
 12  Amount            11239 non-null  int32
dtypes: int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

In [9]:
```python
df.nunique()
```

Out[9]:
```
User_ID             3755
Cust_name           1250
Product_ID          2351
Gender                 2
Age Group              7
Age                   81
Marital_Status         2
State                 16
Zone                   5
Occupation            15
Product_Category      18
Orders                 4
Amount              6584
Status                 0
unnamed1               0
dtype: int64
```

In [10]:
```python
#Here we are counting NA values in the data set
pd.isna(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
Status              11251
unnamed1            11251
dtype: int64
```

In [11]:
```python
#Above we know that 'status' & 'unnamed1' columns has the most NA values and we do not require them further so we are droping them.

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

In [12]:
```python
#We can see now both columns ('Status'&'unnamed1') have been removed from the table.

df.head()
```

Out[12]:

	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 [14]:
```python
#We also know that 'Amount' column has 12 NA values and we need to clear it to make meaningful insight

df.dropna(inplace=True)
```

In [15]:
```python
#Here can see that all the NA values have been removed from the table.

pd.isna(df).sum()
```

Out[15]:
```
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 [17]:
```python
#Here We will check the data types of all the columns and if it requires to change it we will do it.

df.dtypes
```

Out[17]:
```
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
dtype: object
```

In [18]:
```python
#Here we found that 'Amount' column should be int type then float so now will change it

df['Amount'] = df['Amount'].astype('int')

df.dtypes
```

Out[18]:
```
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               int32
dtype: object
```

In [34]:
```python
#We need to change the name of a column called 'Marital_Status' to 'Shaadi'

df.rename(columns={'Marital_Status' : 'Shaadi'}, inplace=True)

df.head()
```

Out[34]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	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

In [38]:
```python
df.describe().round()
```

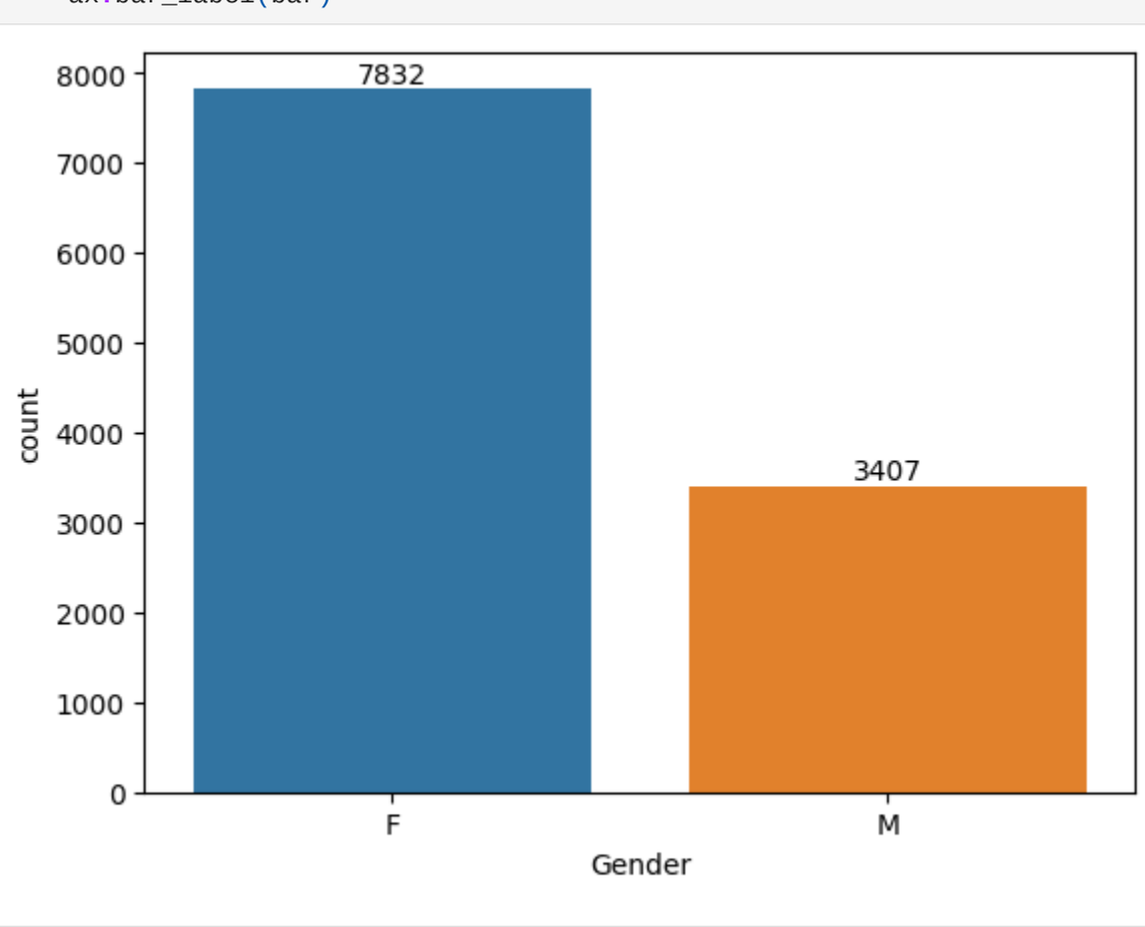
Out[38]:

	User_ID	Age	Shaadi	Orders	Amount
count	11239.0	11239.0	11239.0	11239.0	11239.0
mean	1003004.0	35.0	0.0	2.0	9454.0
std	1716.0	13.0	0.0	1.0	5222.0
min	1000001.0	12.0	0.0	1.0	188.0
25%	1001492.0	27.0	0.0	2.0	5443.0
50%	1003064.0	33.0	0.0	2.0	8109.0
75%	1004426.0	43.0	1.0	3.0	12675.0
max	1006040.0	92.0	1.0	4.0	23952.0

Exploratory Data Analysis

Gender

In [48]:
```python
#Here we can see women customers are more then men

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

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



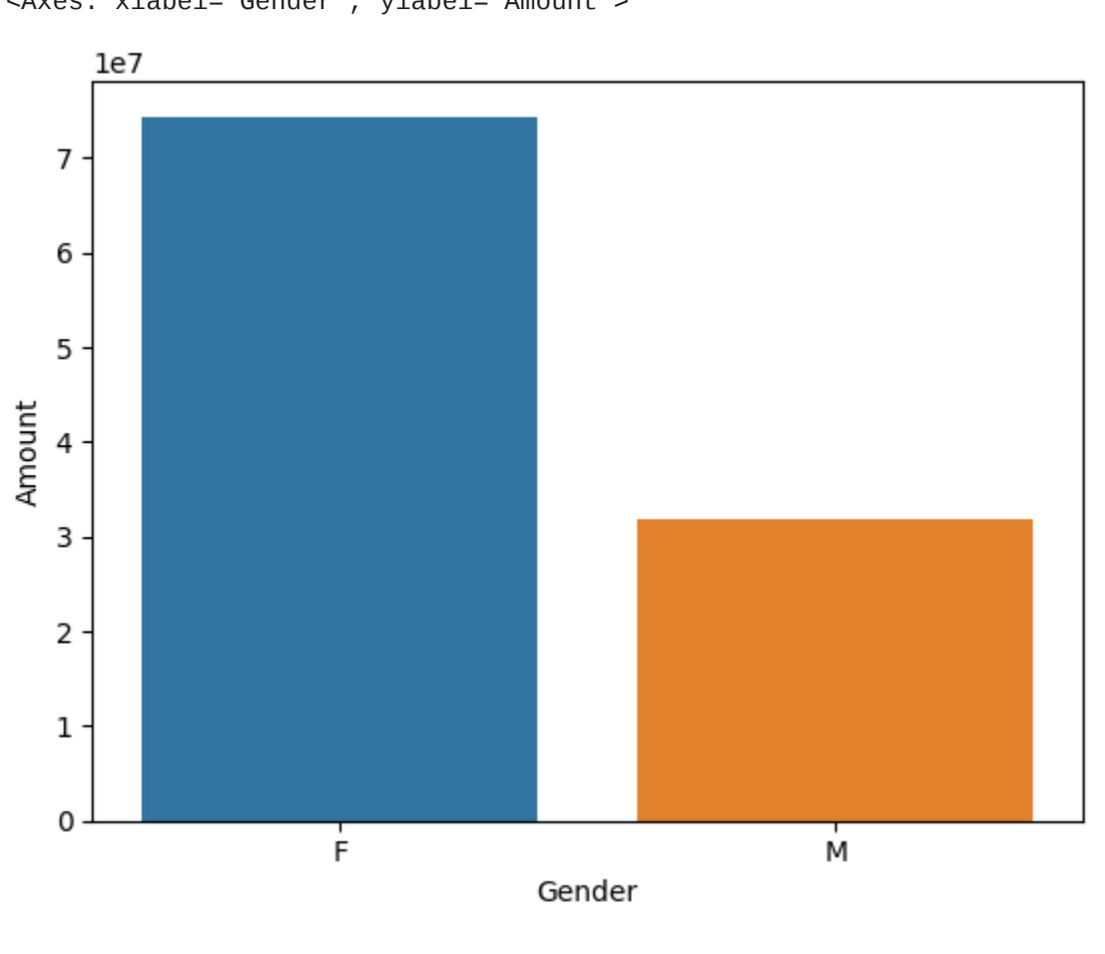
In [50]:
```python
#Here we are seeing which gender is spending more

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

Out[50]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [59]:
```python
#We can see the same in graph form :

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

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

Out[59]: <Axes: xlabel='Gender', ylabel='Amount'>



The above chart shows women are willing to pay more than men

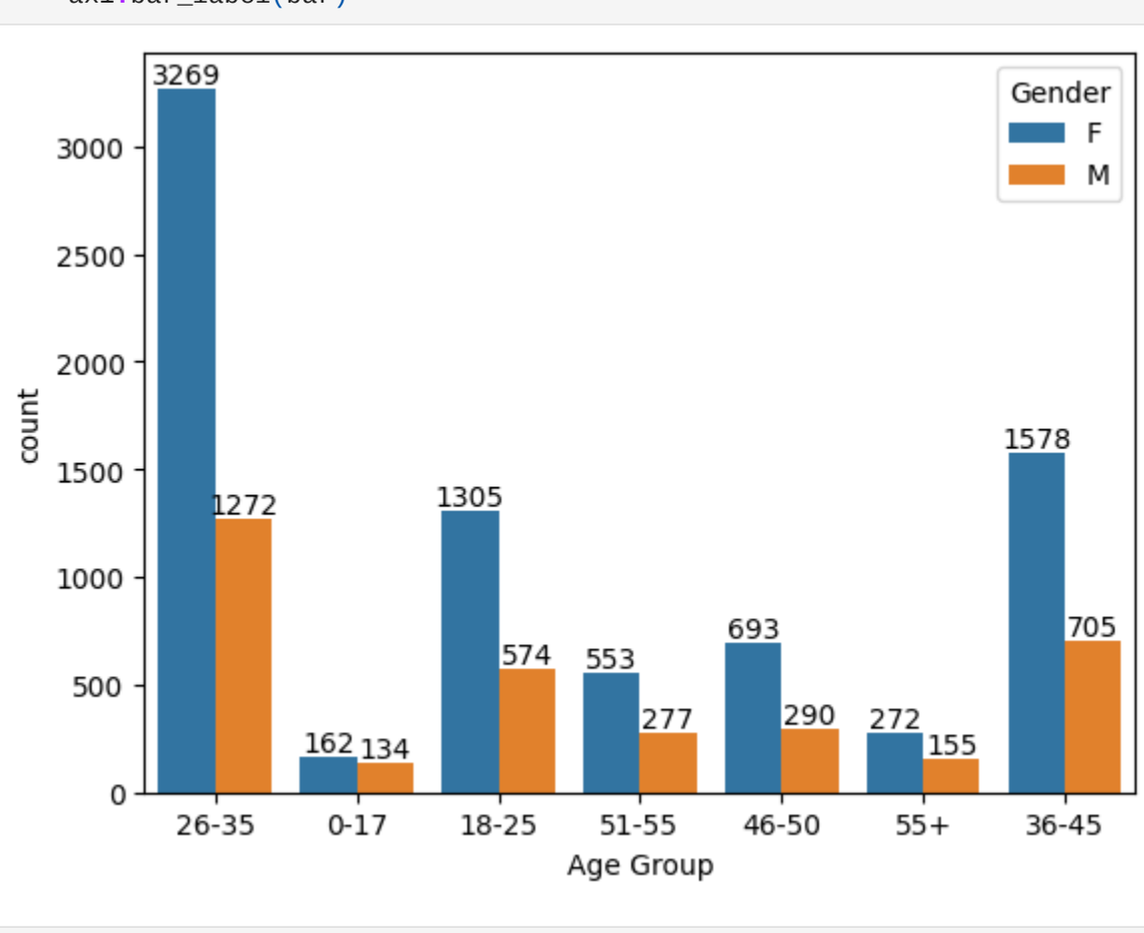
Age

Here we will see which age group purchases higher

In [60]:
```python
df.columns
```

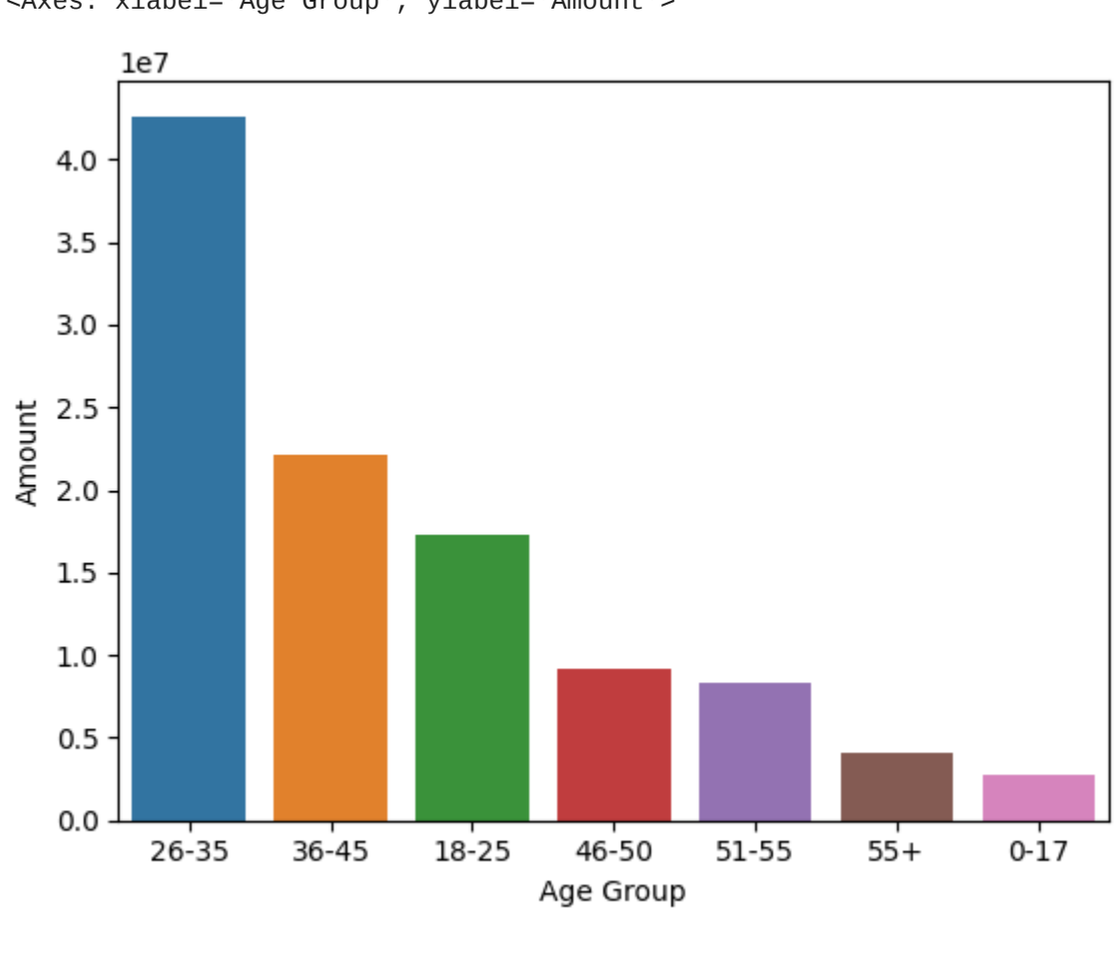
Out[60]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Shaadi', 'State', 'Zone', 'Occupation', 'Product_Category', 'Orders',
       'Amount'],
      dtype='object')
```

In [67]:
```python
ax1 = sns.countplot(x='Age Group', data = df, hue ='Gender')

for bar in ax1.containers:
    ax1.bar_label(bar)
```



In [66]:
```python
Sales_ag =  df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)

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

Out[66]: <Axes: xlabel='Age Group', ylabel='Amount'>



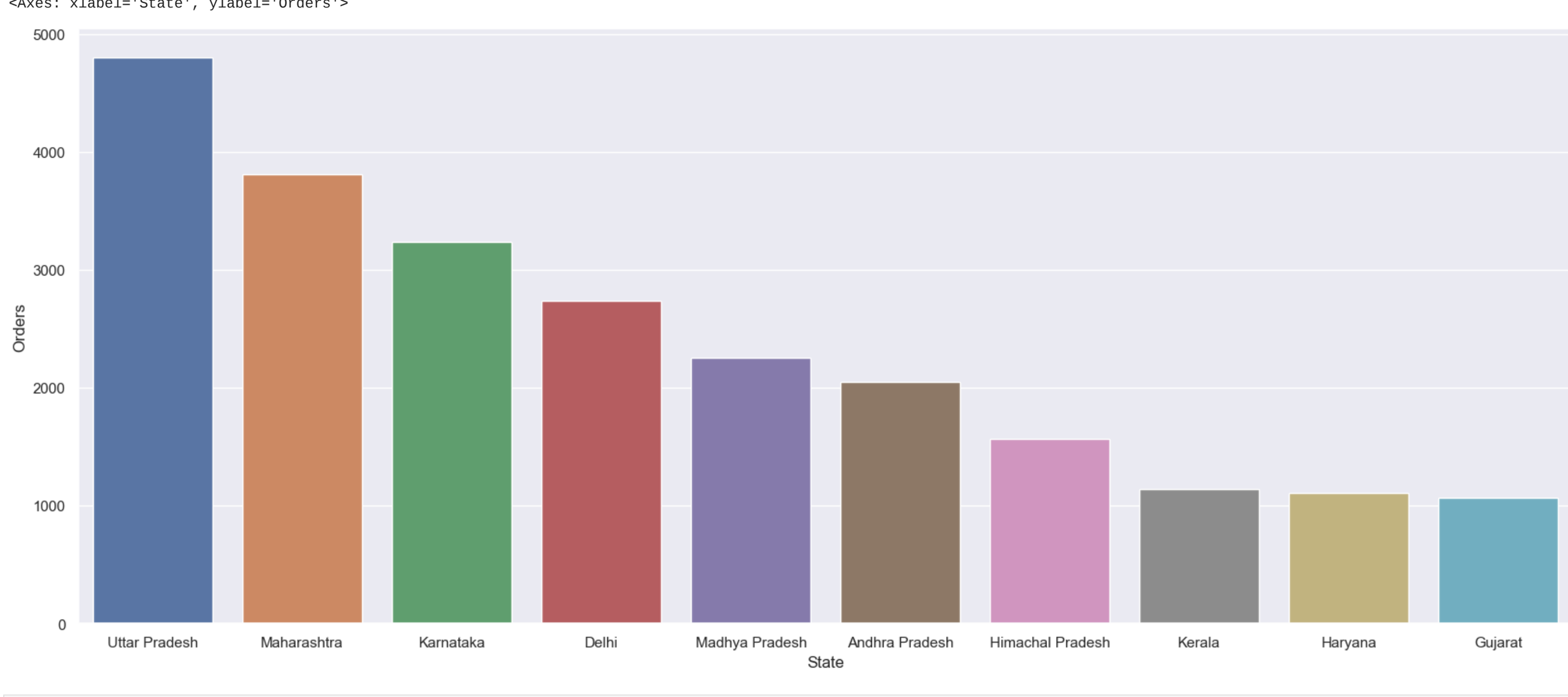
we can see in the above graph women buying more in the age group between 26-35

State

In this section we will see the insights of states

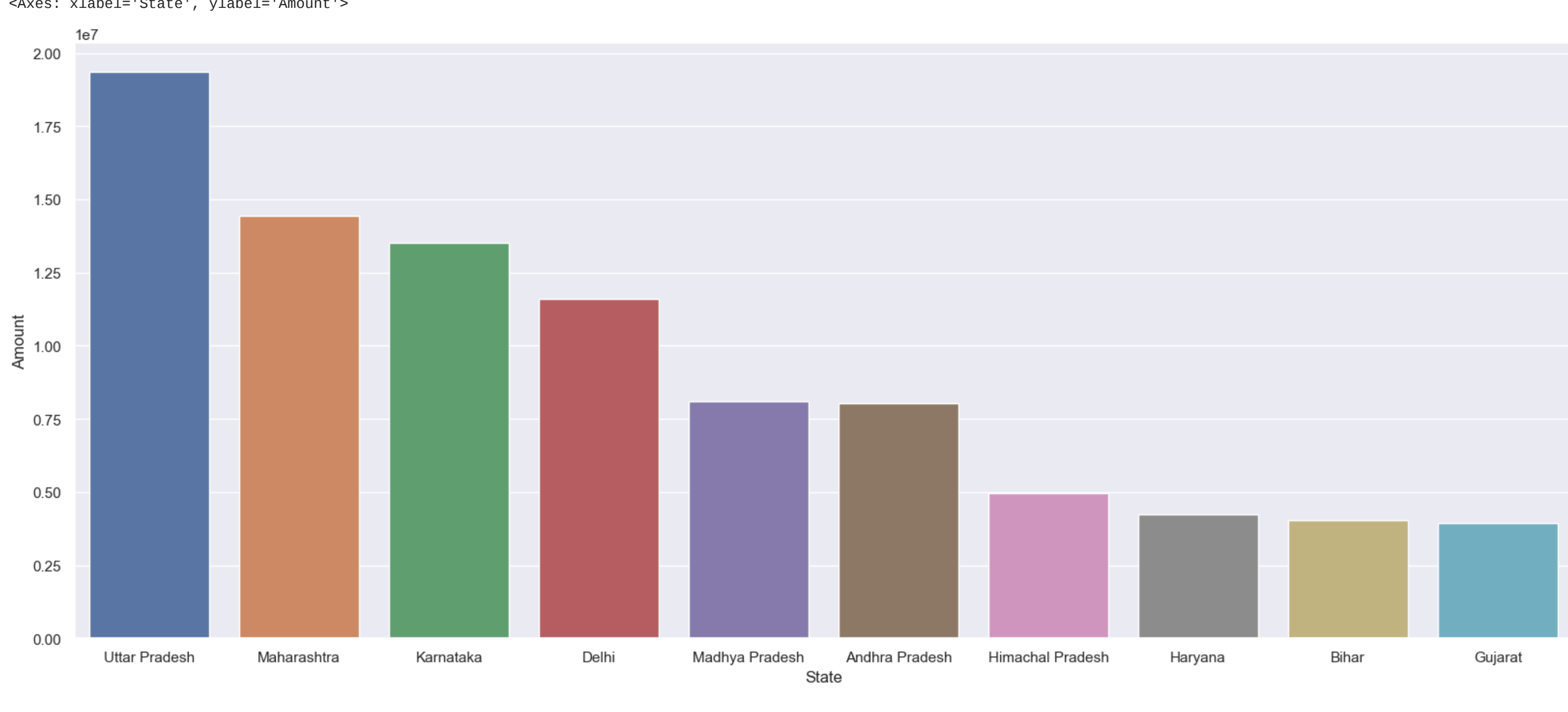
In [80]:
```python
Order_State = df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by= 'Orders', ascending = False).head(10)


sns.set(rc = {'figure.figsize': (20,8)})
sns.barplot(x='State',y='Orders', data = Order_State)
```

Out[80]: <Axes: xlabel='State', ylabel='Orders'>



In [88]:
```python
Amount_State =  df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by = 'Amount', ascending = False).head(10)


sns.set(rc={'figure.figsize':(20,8)})
sns.barplot(x='State', y='Amount', data = Amount_State)
```

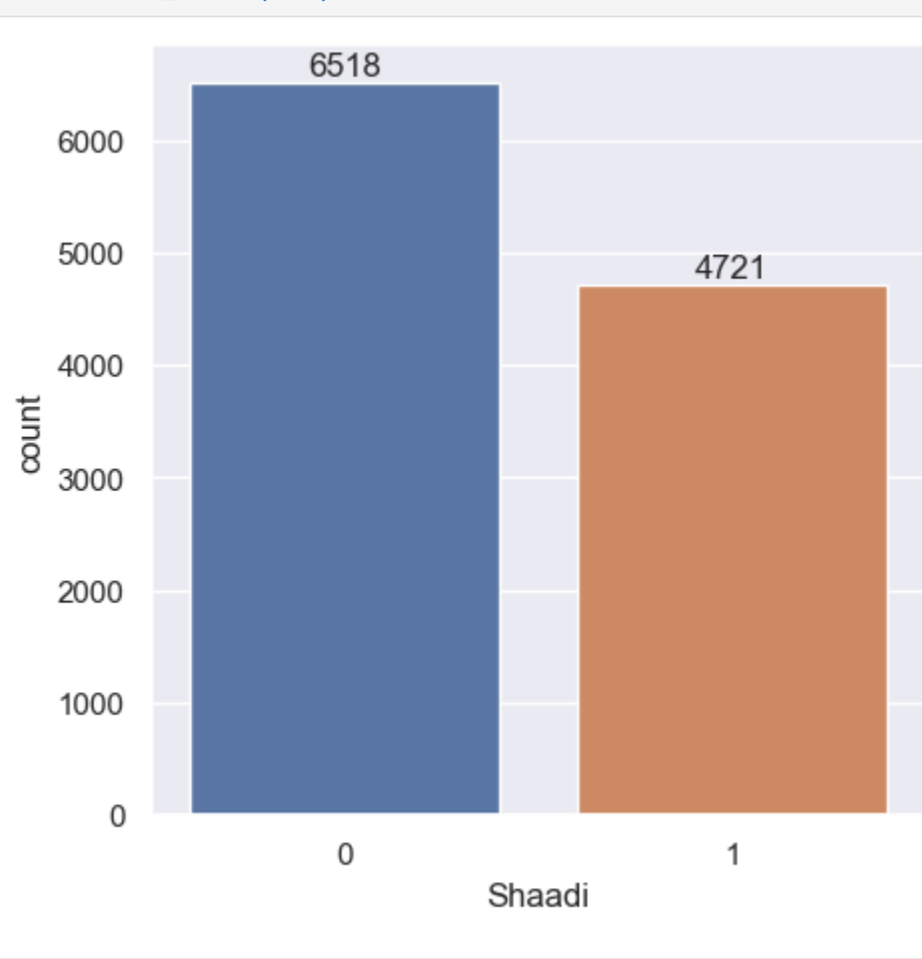
Out[88]: <Axes: xlabel='State', ylabel='Amount'>



From the Above state graphs, we can that most orders are coming from Uttar Pradesh, Maharashtra, and Karnataka respectively and most sales/amounts is from Uttar Pradesh, Maharashtra, and Karnataka respectively.

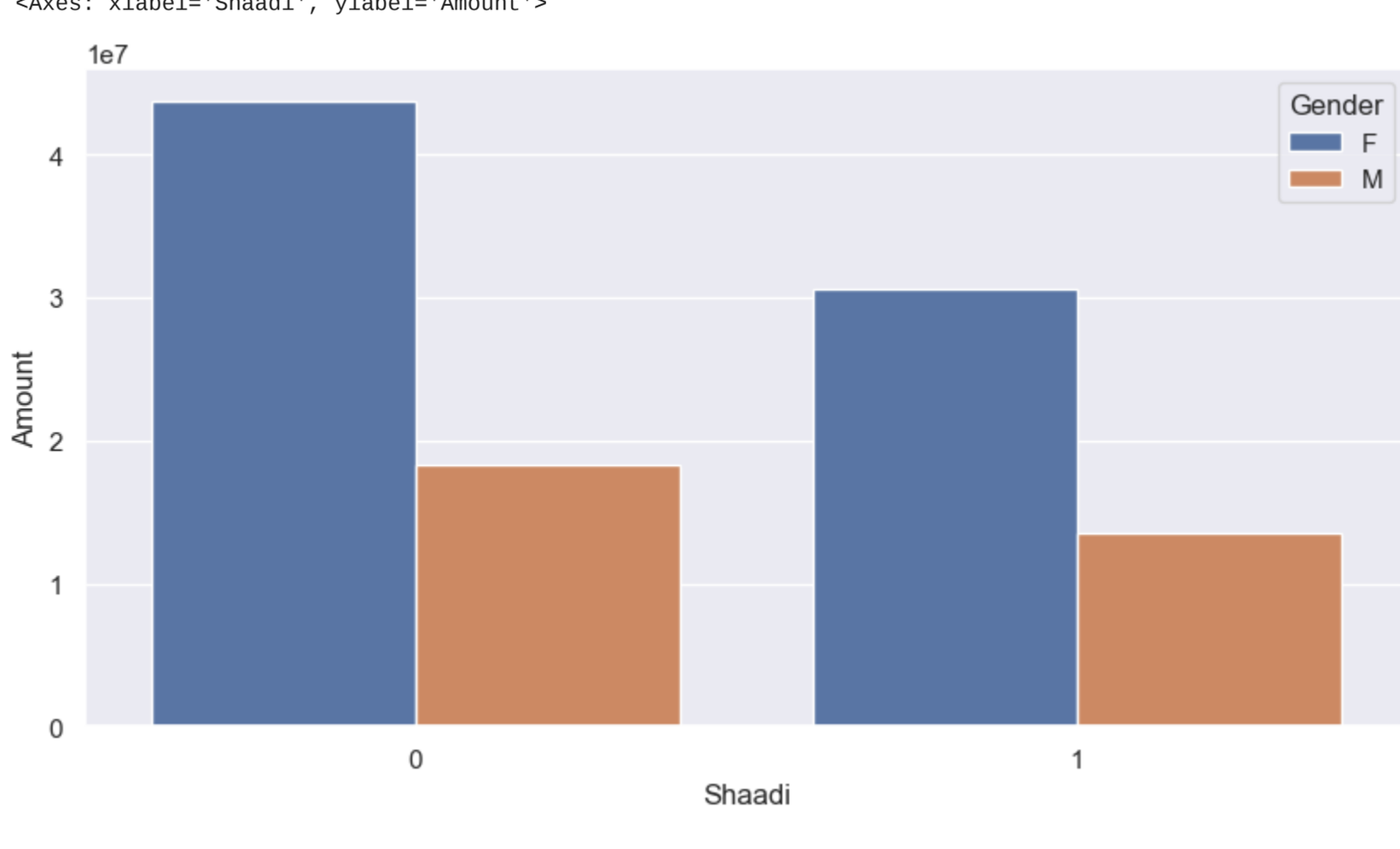
Marital Status

In [98]:
```python
ma = sns.countplot(x='Shaadi', data = df)

sns.set(rc={'figure.figsize':(2,5)})
for bar in ma.containers:
    ma.bar_label(bar)
```



In [102…
```python
SGA = df.groupby(['Shaadi', 'Gender'], as_index=False)['Amount'].sum().sort_values(by = 'Amount', ascending = False)


sns.set(rc={'figure.figsize':(10,5)})
sns.barplot(x='Shaadi', y='Amount', data = SGA, hue = 'Gender')
```

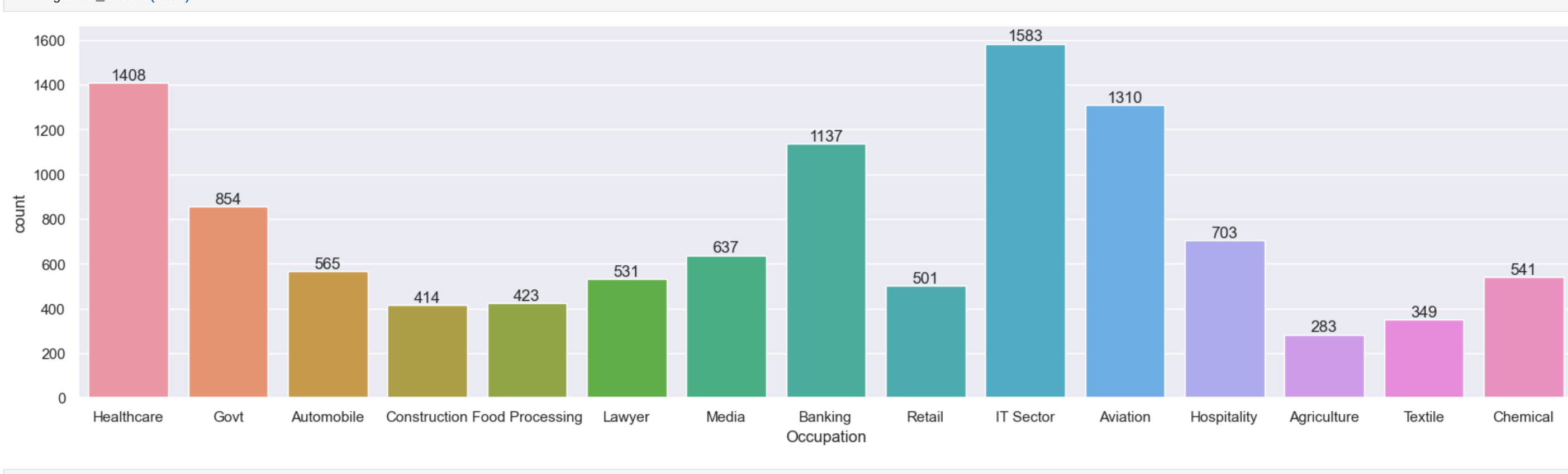
Out[102]: <Axes: xlabel='Shaadi', ylabel='Amount'>



From the above graphs we can see female are buying more and the wallet is also bigger than men

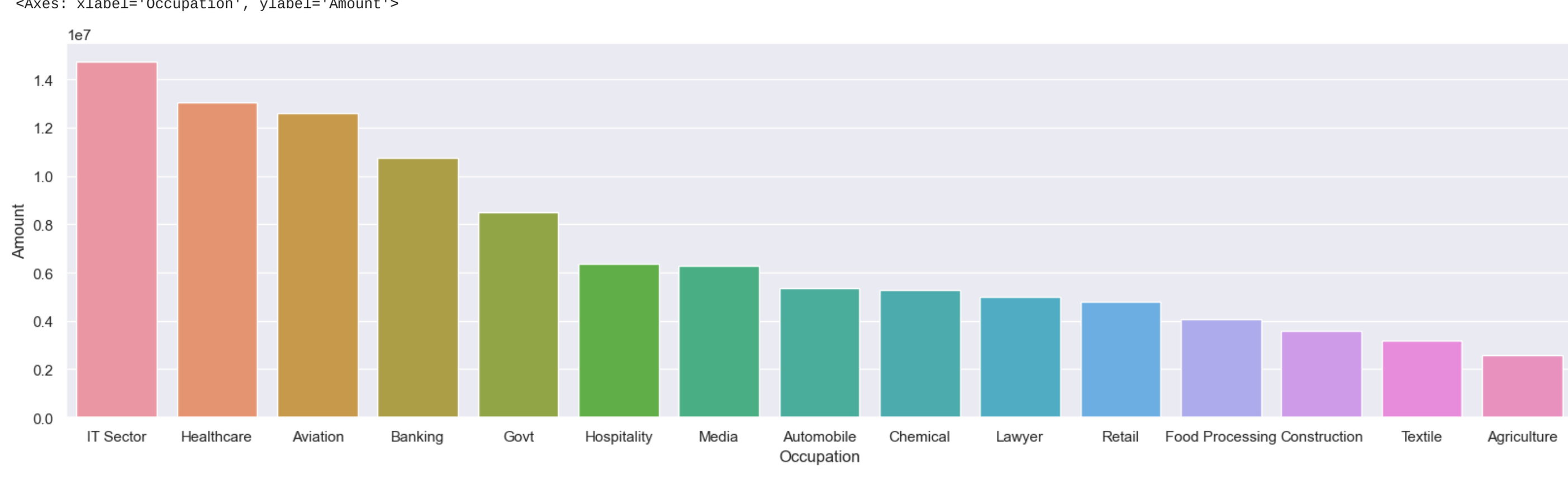
Occupation

In [107…
```python
og = sns.countplot(x='Occupation', data = df)

sns.set(rc={'figure.figsize':(20,5)})
for bar in og.containers:
    og.bar_label(bar)
```



In [116…
```python
oga = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by = 'Amount', ascending = False)


sns.barplot(x='Occupation', y='Amount', data = oga)
```

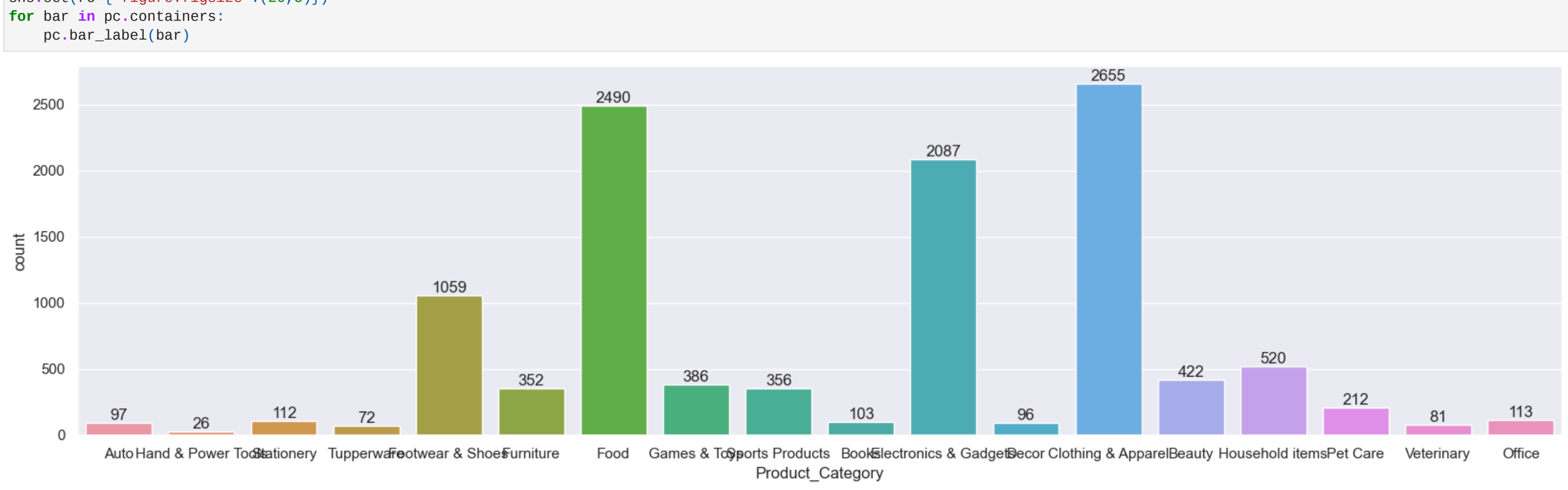
Out[116]: <Axes: xlabel='Occupation', ylabel='Amount'>



From the above graphs, we can see people who work in the IT sector expect to spend more.

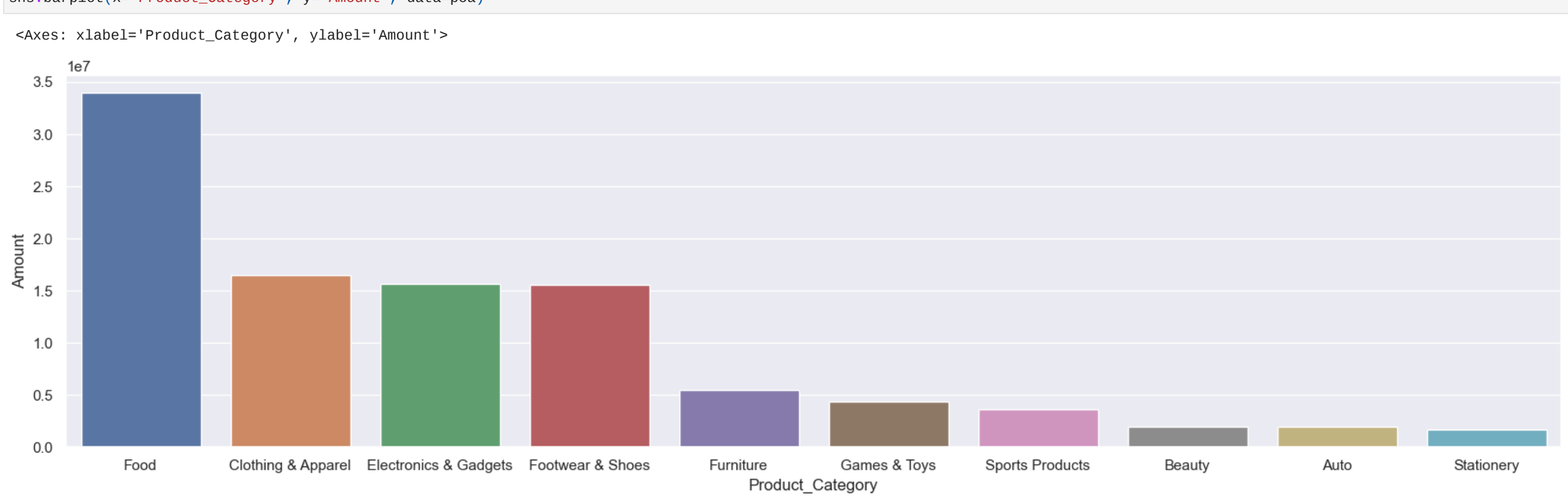
Product Category

In [139…
```python
pc = sns.countplot(x='Product_Category', data = df)


sns.set(rc={'figure.figsize':(20,5)})
for bar in pc.containers:
    pc.bar_label(bar)
```



In [131…
```python
pca = 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(x='Product_Category', y='Amount', data=pca)
```

Out[131]: <Axes: xlabel='Product_Category', ylabel='Amount'>



From above graphs we can see the clothing category got most number of orders but customer spent the most on food items

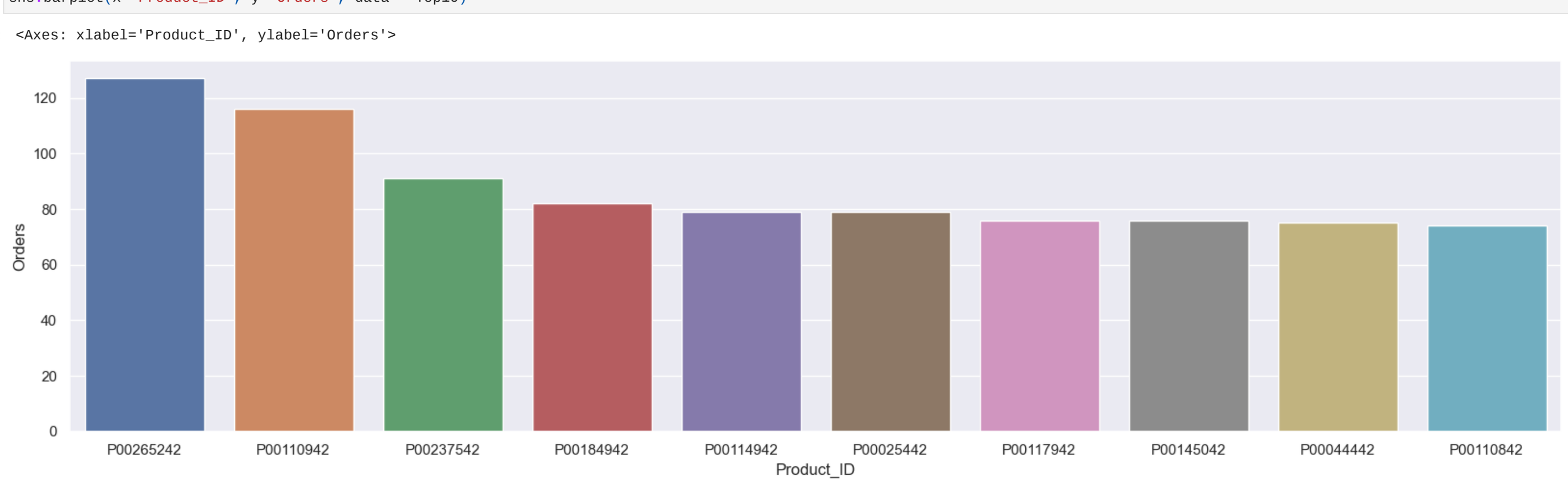
10 Most Sold Products

In [133…
```python
df.columns
```

Out[133]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Shaadi', 'State', 'Zone', 'Occupation', 'Product_Category', 'Orders',
       'Amount'],
      dtype='object')
```

In [138…
```python
Top10 = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by = 'Orders', ascending = False).head(10)



sns.barplot(x='Product_ID', y='Orders', data = Top10)
```

Out[138]: <Axes: xlabel='Product_ID', ylabel='Orders'>



Conclusions

Married women age group between 26-and 35 years from UP, Maharashtra, and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food Clothing and Electronics categories.