In [2]:
```python
# import python libraries

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

In [3]:
```python
# import csv file
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

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

Out[4]: (11251, 15)

In [5]:
```python
df.head(10)
```

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	23952.00	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 [6]:
```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 [7]:
```python
df.drop(['Status','unnamed1'],axis=1, inplace=True)
```

In [8]:
```python
df.head(10)
```

Out[8]:

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

In [9]:
```python
pd.isnull(df).sum()
```

Out[9]:
```
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 [10]:
```python
df.shape
```

Out[10]: (11251, 13)

In [11]:
```python
df.dropna(inplace=True)
```

In [12]:
```python
df['Amount']= df['Amount'].astype(int)
```

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
df.rename(columns={'Marital_Status':'Marriage'})
```

Out[14]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marriage	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 [15]:
```python
df.describe()
```

Out[15]:

	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 [16]:
```python
df[['Amount']].describe()
```

Out[16]:

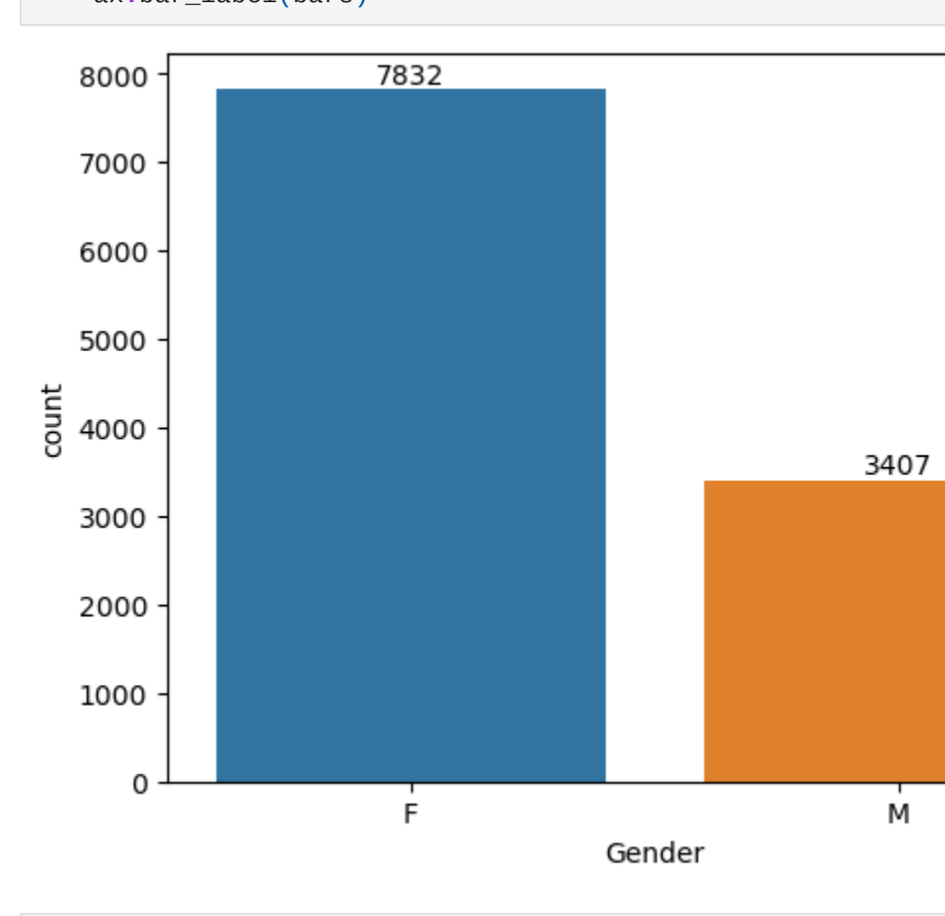
	Amount
count	11239.000000
mean	9453.610553
std	5222.355168
min	188.000000
25%	5443.000000
50%	8109.000000
75%	12675.000000
max	23952.000000

EDA

In [17]:
```python
ax = sns.countplot(x='Gender',data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```

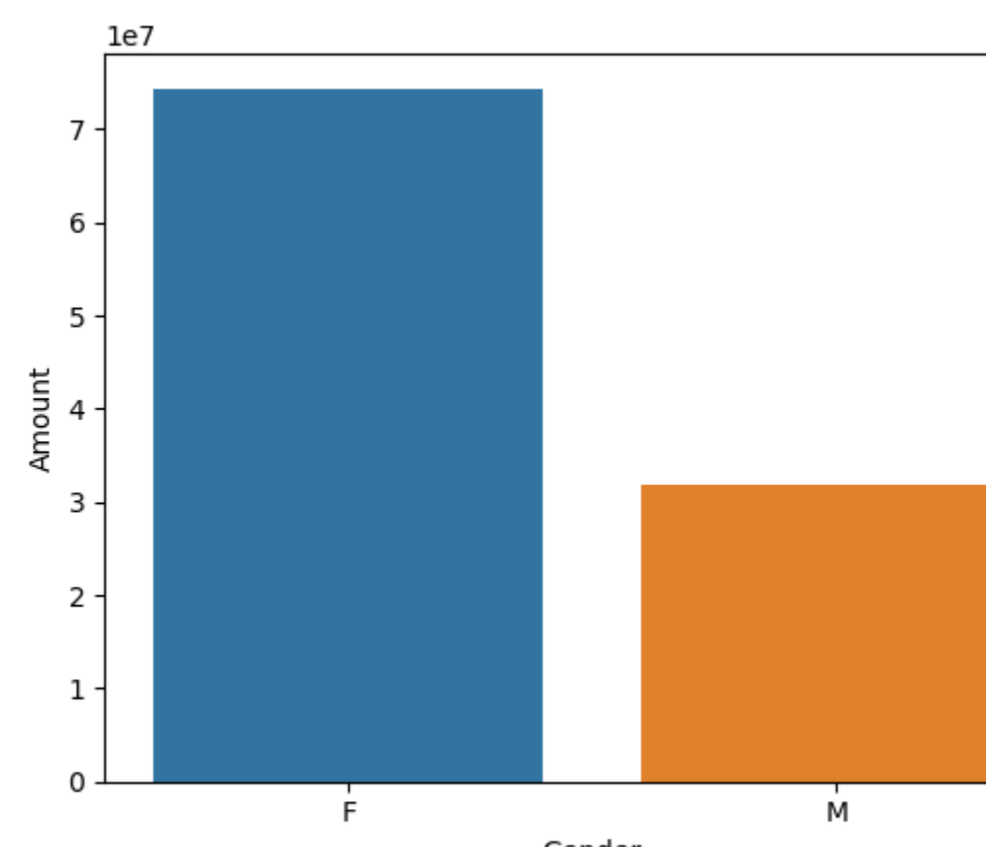
In [18]:
```python
sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.barplot(x='Gender', y='Amount', data = sales_gen)
```

Out[18]: <Axes: xlabel='Gender', ylabel='Amount'>

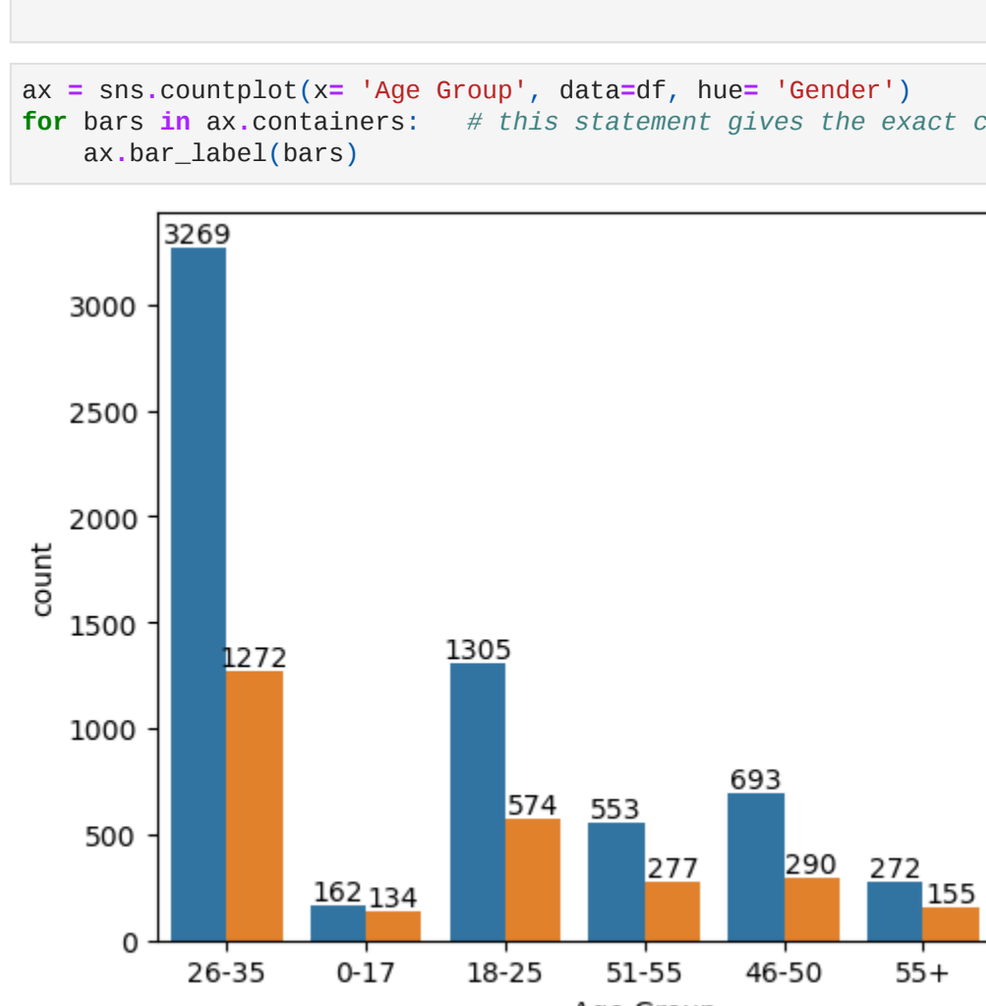


In [ ]:

In [19]:
```python
ax = sns.countplot(x= 'Age Group', data=df, hue= 'Gender')
for bars in ax.containers:   # this statement gives the exact count above the bars as shown below
    ax.bar_label(bars)
```

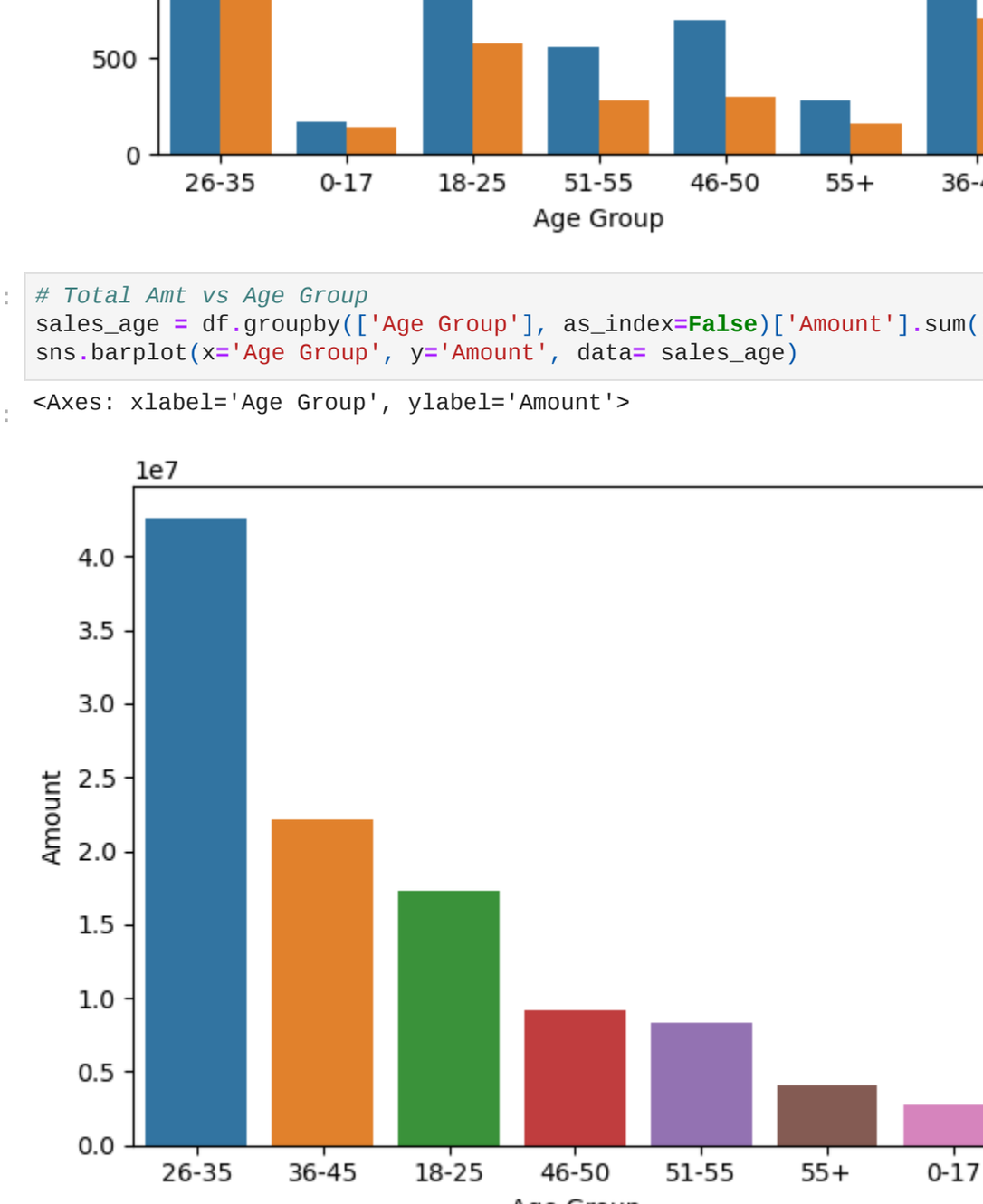


In [20]:
```python
ax = sns.countplot(x= 'Age Group', data=df, hue= 'Gender')
```



In [21]:
```python
# Total Amt 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)
```

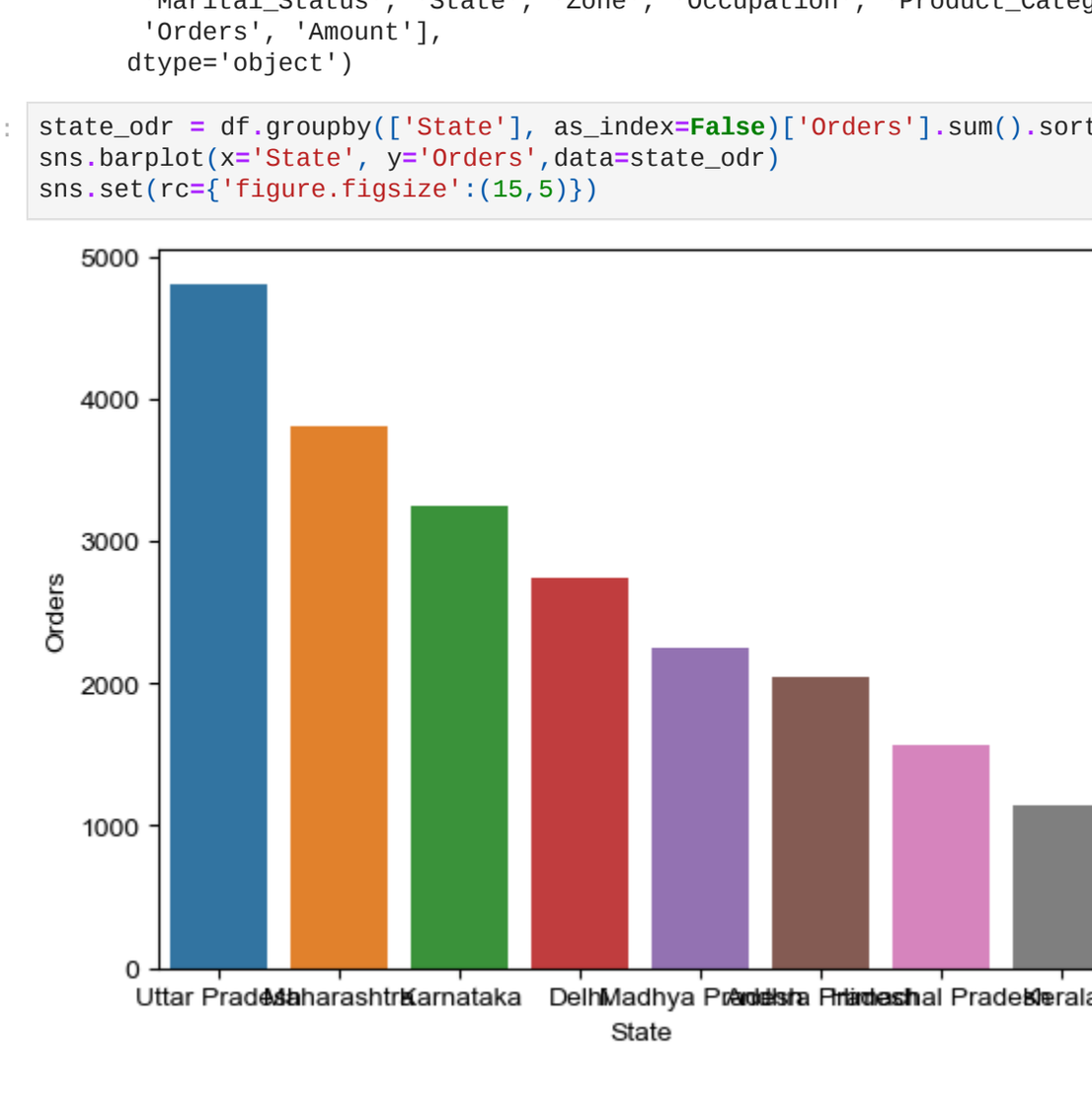
Out[21]: <Axes: xlabel='Age Group', ylabel='Amount'>



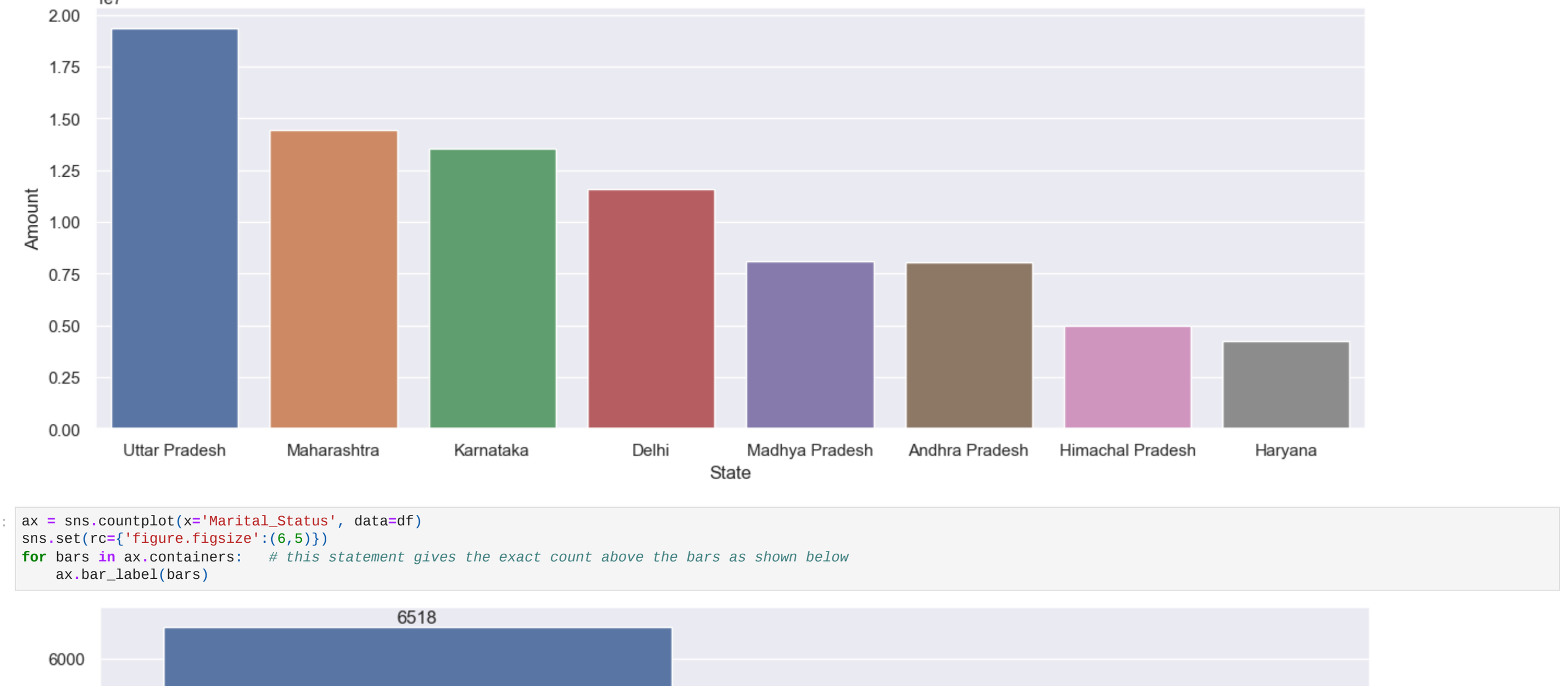
In [22]:
```python
df.columns
```

Out[22]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

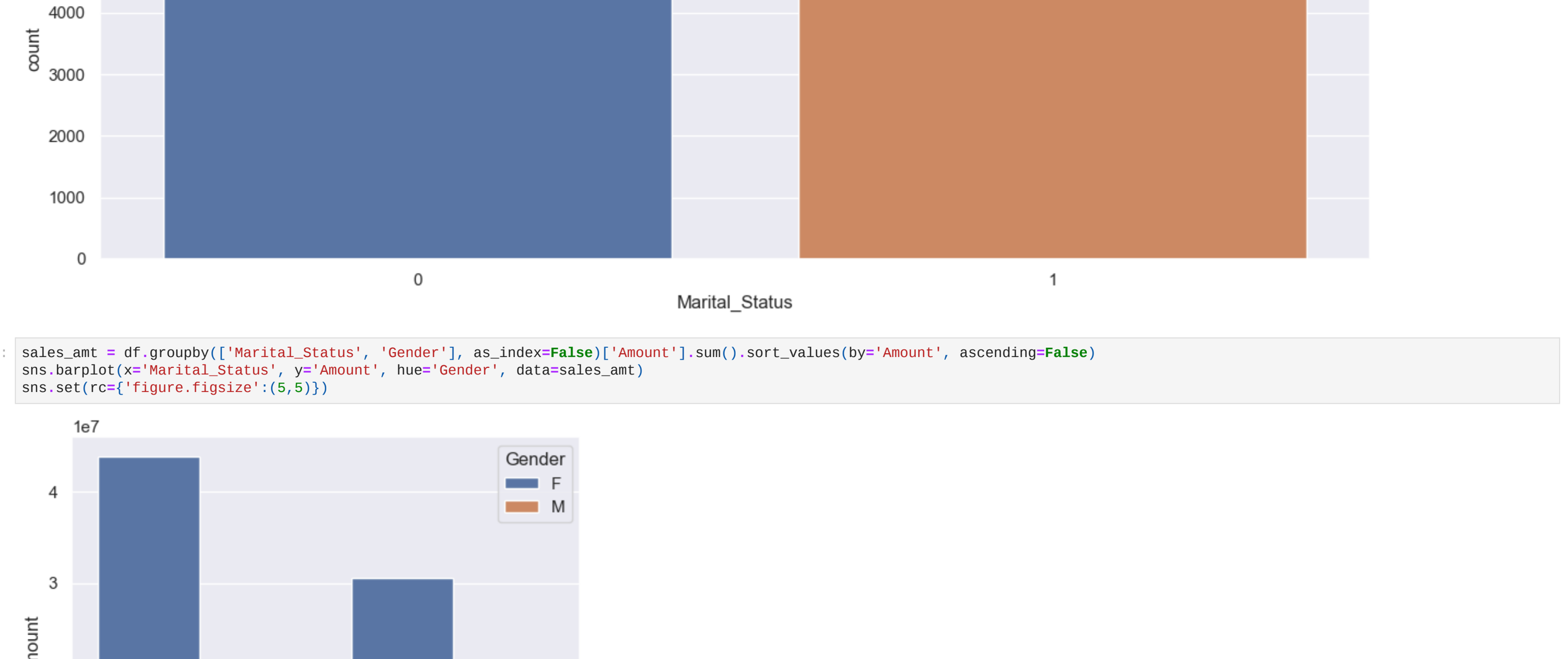
In [23]:
```python
state_odr = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(8)
sns.barplot(x='State', y='Orders',data=state_odr)
sns.set(rc={'figure.figsize':(15,5)})
```



In [24]:
```python
states_amt= df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(8)
sns.barplot(x='State', y='Amount', data=states_amt)
sns.set(rc={'figure.figsize':(15,5)})
```



In [25]:
```python
ax = sns.countplot(x='Marital_Status', data=df)
sns.set(rc={'figure.figsize':(6,5)})
for bars in ax.containers:   # this statement gives the exact count above the bars as shown below
    ax.bar_label(bars)
```



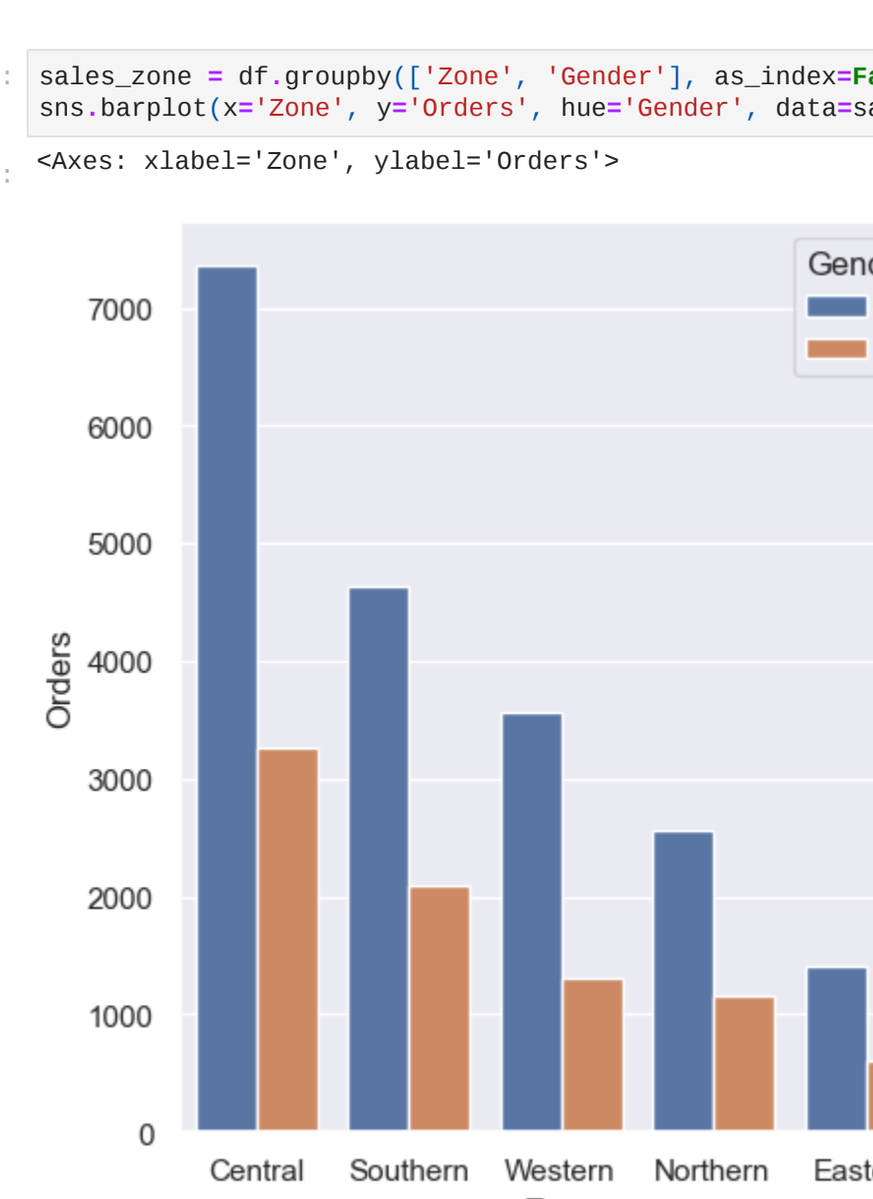
In [26]:
```python
sales_amt = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.barplot(x='Marital_Status', y='Amount', hue='Gender', data=sales_amt)
sns.set(rc={'figure.figsize':(5,5)})
```

In [27]:
```python
df.columns
```

Out[27]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

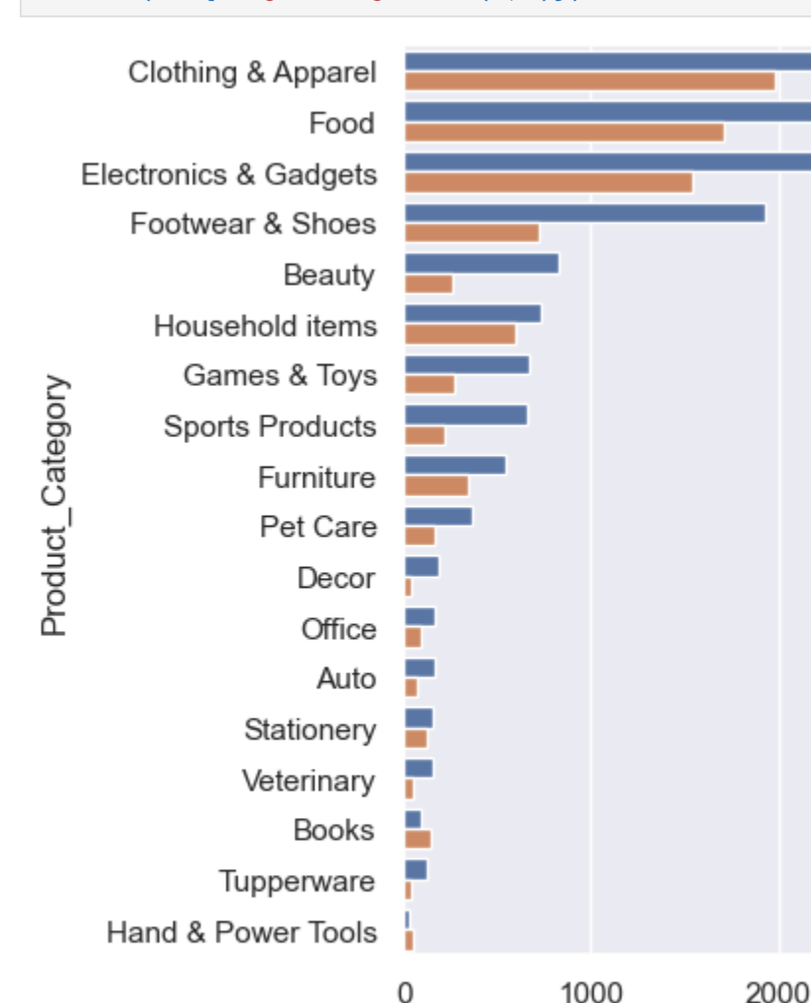
In [28]:
```python
sales_zone = df.groupby(['Zone', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Zone', y='Amount', hue='Gender', data=sales_zone)
```

Out[28]: <Axes: xlabel='Zone', ylabel='Amount'>

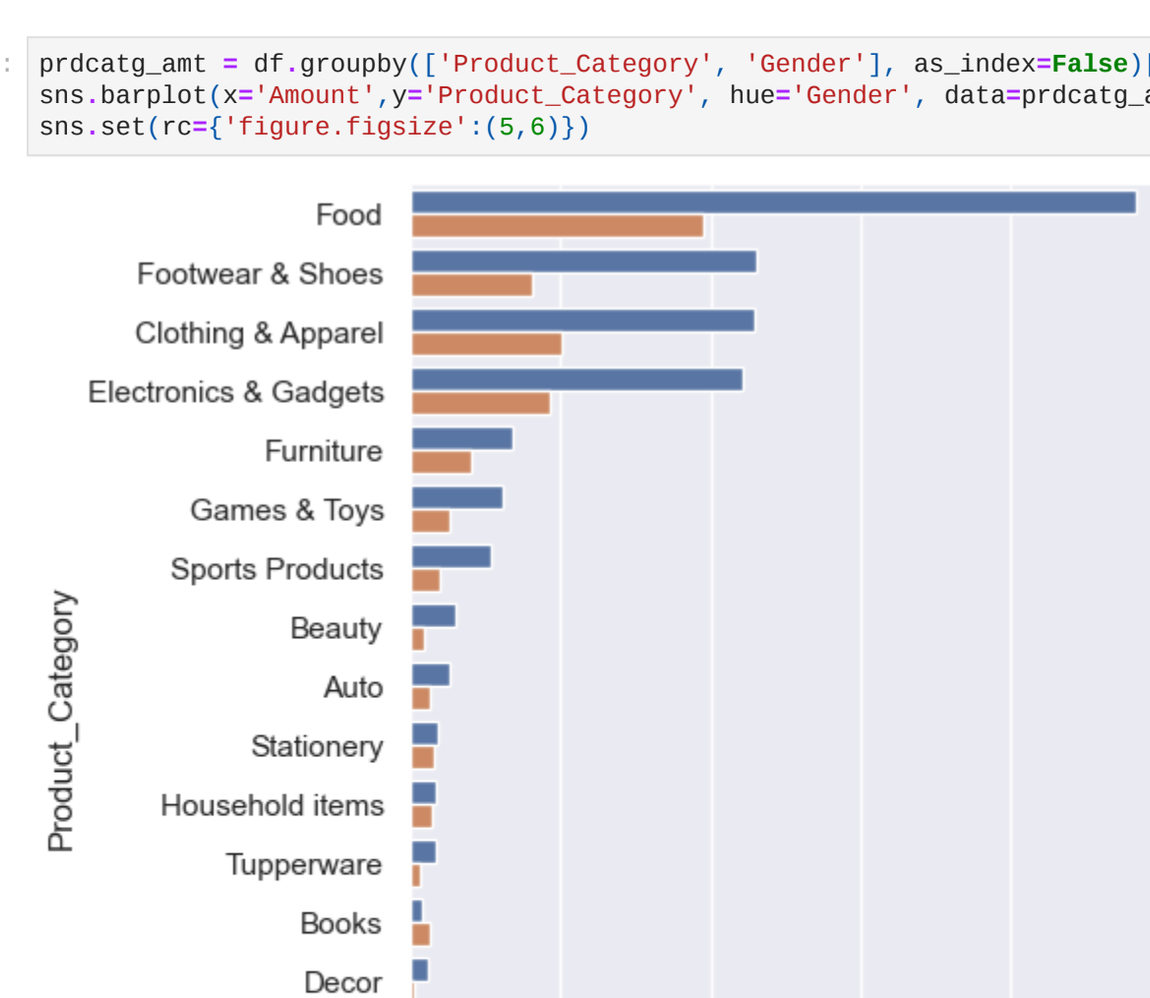


In [39]:
```python
sales_zone = df.groupby(['Zone', 'Gender'], as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False)
sns.barplot(x='Zone', y='Orders', hue='Gender', data=sales_zone)
```

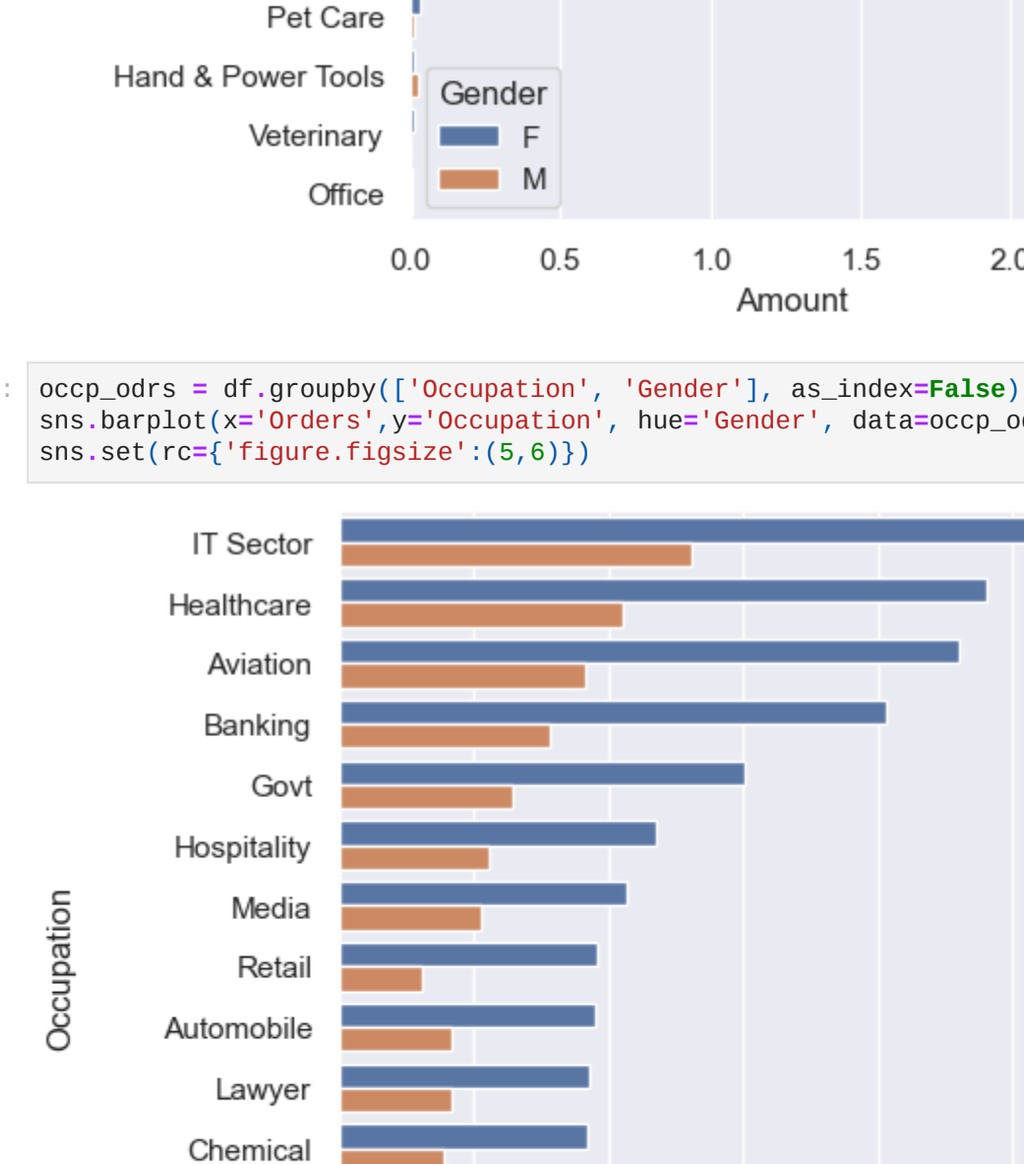
Out[39]: <Axes: xlabel='Zone', ylabel='Orders'>



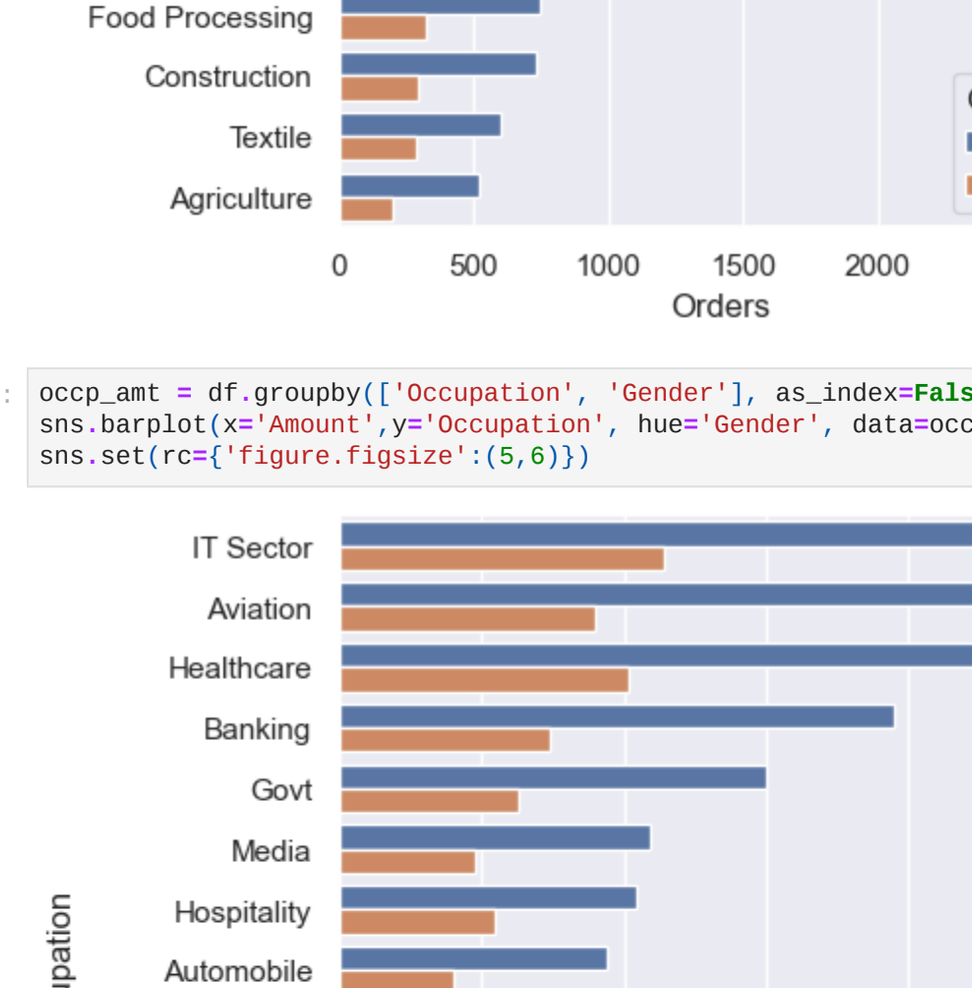
In [38]:
```python
prdcatg_odrs = df.groupby(['Product_Category', 'Gender'], as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False)
sns.barplot(x='Orders',y='Product_Category', hue='Gender', data=prdcatg_odrs)
sns.set(rc={'figure.figsize':(5,6)})
```



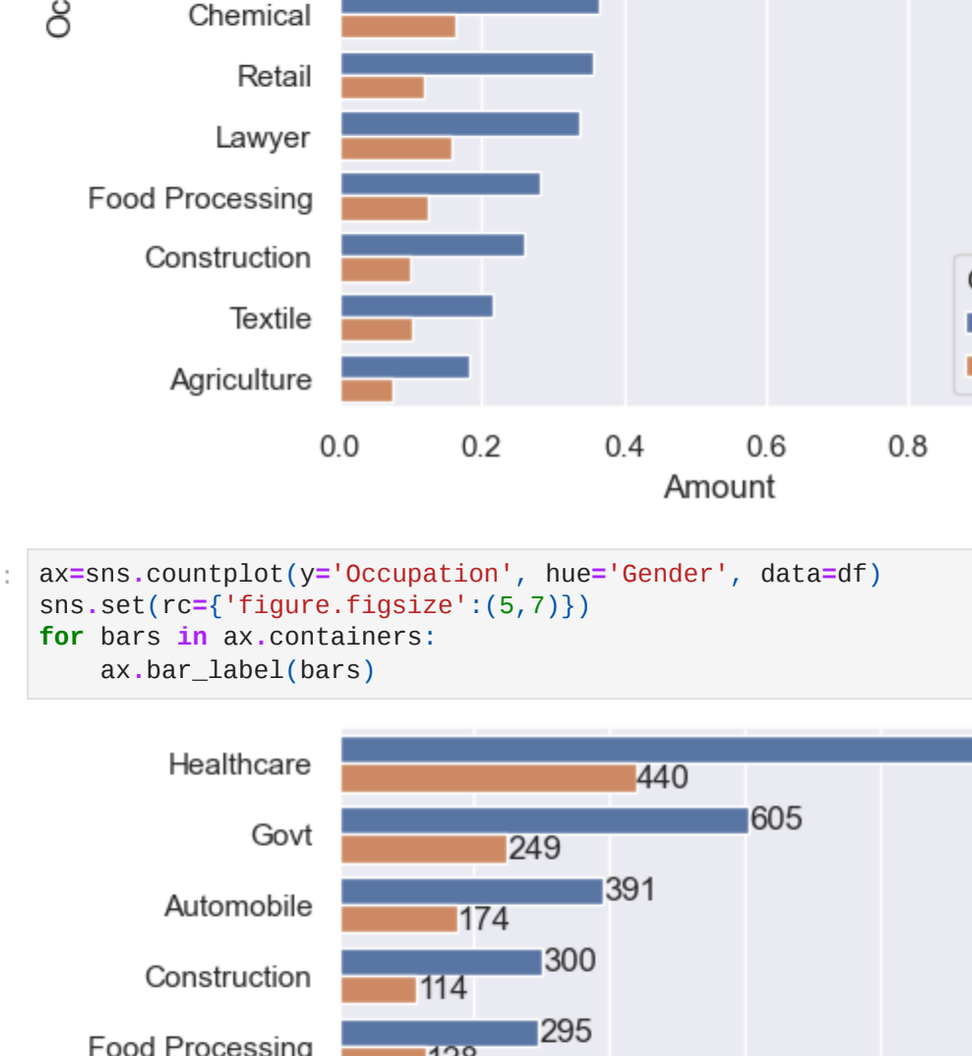
In [56]:
```python
prdcatg_amt = df.groupby(['Product_Category', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Amount',y='Product_Category', hue='Gender', data=prdcatg_amt)
sns.set(rc={'figure.figsize':(5,6)})
```



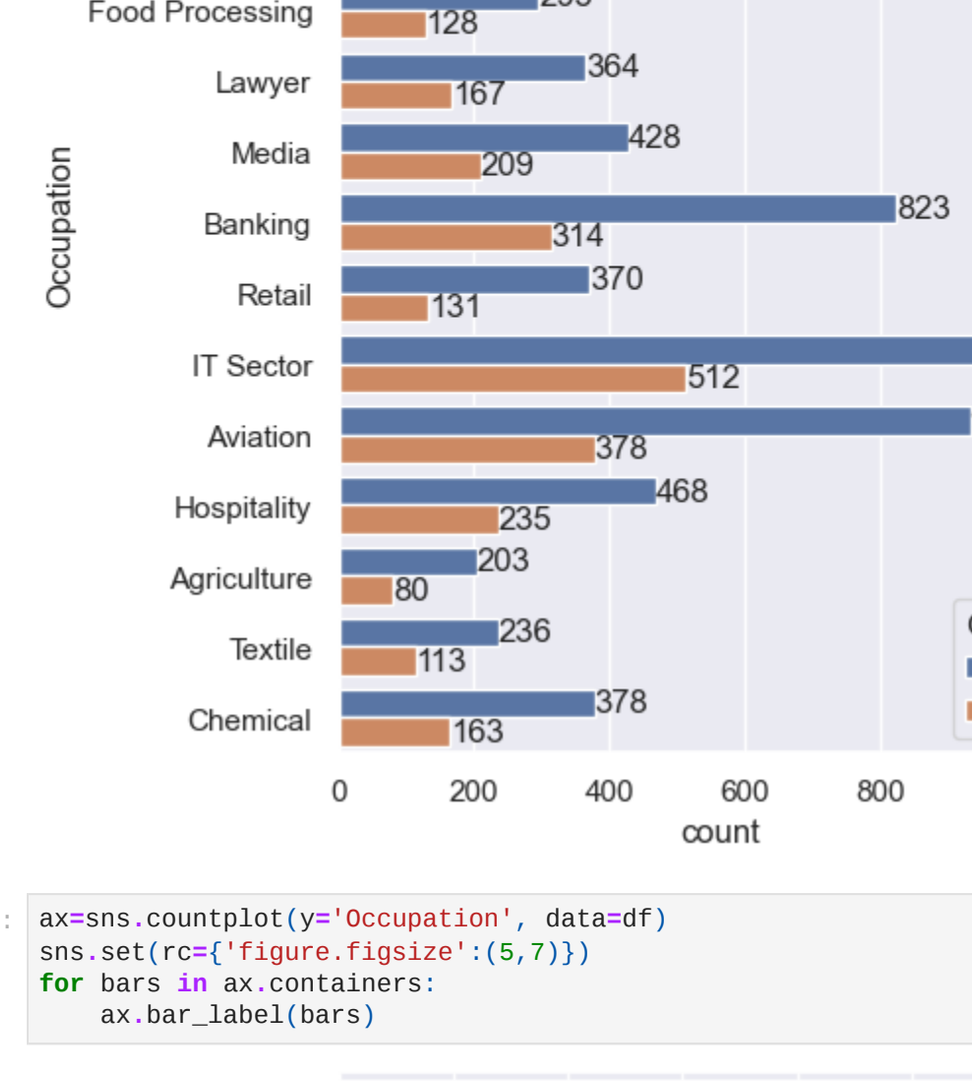
In [42]:
```python
occp_odrs = df.groupby(['Occupation', 'Gender'], as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False)
sns.barplot(x='Orders',y='Occupation', hue='Gender', data=occp_odrs)
sns.set(rc={'figure.figsize':(5,6)})
```



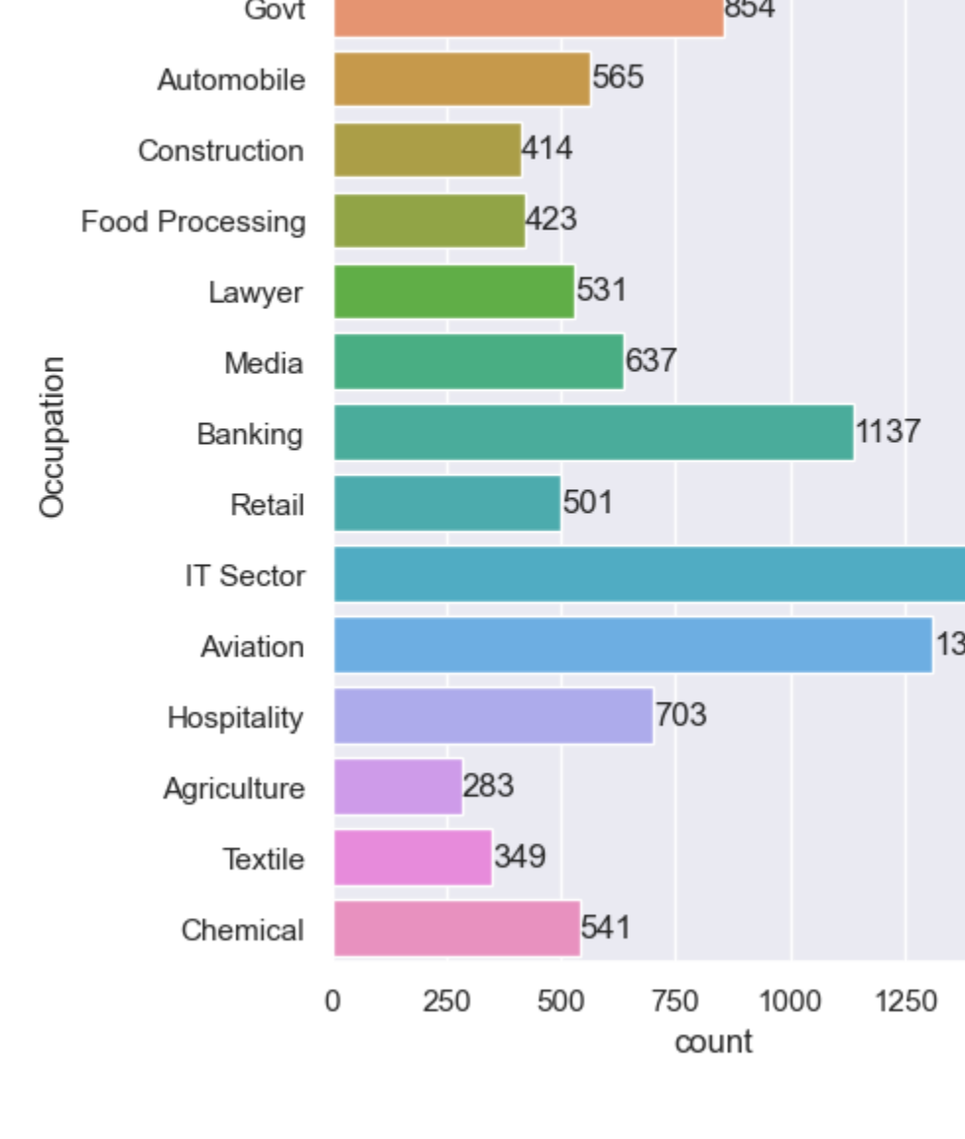
In [45]:
```python
occp_amt = df.groupby(['Occupation', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Amount',y='Occupation', hue='Gender', data=occp_amt)
sns.set(rc={'figure.figsize':(5,6)})
```



In [53]:
```python
ax=sns.countplot(y='Occupation', hue='Gender', data=df)
sns.set(rc={'figure.figsize':(5,7)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [54]:
```python
ax=sns.countplot(y='Occupation', data=df)
sns.set(rc={'figure.figsize':(5,7)})
for bars in ax.containers:
    ax.bar_label(bars)
```



Conclusions

1. Most of the customers are unmarried females and amount spent by them is also greater than males.
2. Most of the customers are of age group 26-35, followed by 36-45 age grp.
3. Most amount spent is by age group of 26-35 followed by 36-45 age grp.
4. Most of the orders are from UPfollowed by Maharastra and Karnataka,also amount spent by states also follow the same trend.
5. Most of the customers are from IT sector followed by Healthcare and Aviation and amount spent by different sectors also follow the same trend.
6. Most amount is spent by female customers from central zone followed by southern and western.
7. Most orders come from central zone followed by southern and western.
8. Most orders came for product category Clothing followed by Food and Electronics mostly by females.
9. Most amount was spent on Food followed by Footwear and Clothing mostly by females.