In [14]:
```
!pip install seaborn # install seaborn library
```

```
ERROR: Invalid requirement: '#'
WARNING: You are using pip version 20.2.1; however, version 24.3.1 is available.
You should consider upgrading via the 'c:\users\pushp\appdata\local\programs\pyth
on\python38\python.exe -m pip install --upgrade pip' command.
```

In [15]:
```
import seaborn as sns
import pandas as pd #read data_sets
import numpy as np #working with arrays
import matplotlib.pyplot as plt #plot graph and pychart
%matplotlib inline #is a backend comand in jupyter notebook that enables the ren
```

```
UsageError: unrecognized arguments: #is a backend comand in jupyter notebook that
enables the rendaring of matplotlib plot directly below of the code cells
```

In [16]:
```
mca = pd.read_csv("Diwali Sales Data.csv", encoding="unicode_escape")
mca
```

Out[16]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Mah
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar
3	1001425	Sudevi	P00237842	M	0-17	16	0	Ka
4	1000588	Joni	P00057942	M	26-35	28	1	
...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Mah
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	
11248	1001209	Oshin	P00201342	F	36-45	40	0	
11249	1004023	Noonan	P00059442	M	36-45	37	0	Ka
11250	1002744	Brumley	P00281742	F	18-25	19	0	Mah

11251 rows × 15 columns

In [17]:
```
mca.shape #return the numbers of row
```

Out[17]:
```
(11251, 15)
```

In [18]:
```
mca.head() # display first five row and column
```

Out[18]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	Stat
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtı
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prades
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prades
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnatak
4	1000588	Joni	P00057942	M	26-35	28	1	Gujara

In [19]:
```
mca.tail() # display last five row
```

Out[19]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	S
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharas
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Har
11248	1001209	Oshin	P00201342	F	36-45	40	0	Ma Pra
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karna
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharas

In [20]:
```
mca.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 [21]:
```
#drop bank columns
mca.drop(['Status', 'unnamed1'], axis=1, inplace=True, errors='ignore')
```

```
print(mca.columns)
```

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

In [22]:
```
pd.isnull(mca).sum()
```

Out[22]:
```
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 [23]:
```
#drop null values
mca.dropna(inplace=True)
```

In [24]:
```
mca.shape
```

Out[24]:
```
(11239, 13)
```

In [25]:
```
#change data type
mca['Amount']=mca['Amount'].astype('int')
```

In [26]:
```
mca['Amount'].dtypes
```

Out[26]:
```
dtype('int32')
```

In [27]:
```
mca.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]:
```
#rename columns
mca.rename(columns={'Marital_Status':'Shaadi'})
```

Out[28]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat
...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra

11239 rows × 13 columns

In [29]:
```
#describe() method returns description of the data in the DataFrame (i.e. count,
mca.describe()
```

Out[29]:

	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 [30]:
```
#use describe() for specific columns
mca[['Age','Orders','Amount']].describe()
```

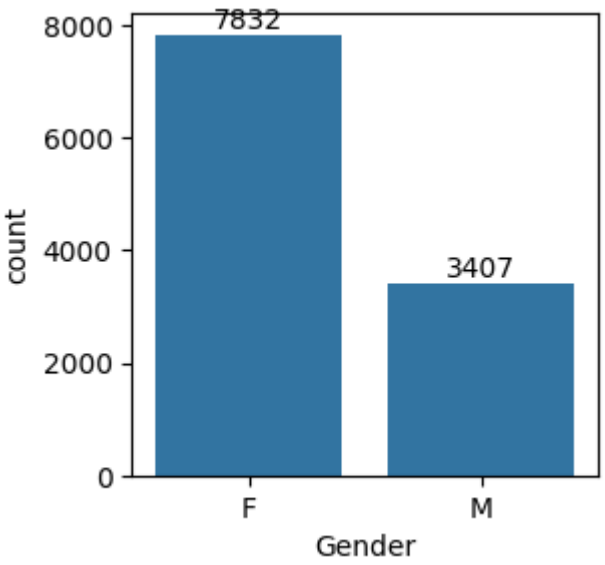
Out[30]:

	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

Exploratory Data Analysis

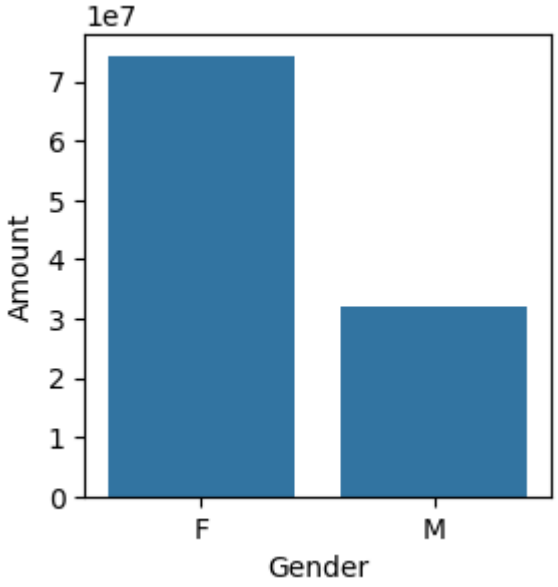
Gender

In [43]:
```
plt.figure(figsize=(3,3))
ax=sns.countplot(x='Gender',data=mca)
for bars in ax.containers:
    ax.bar_label(bars)
```



In [44]:
```
plt.figure(figsize=(3,3))
sales_gen=mca.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by=
sns.barplot(x='Gender',y='Amount',data=sales_gen)
```

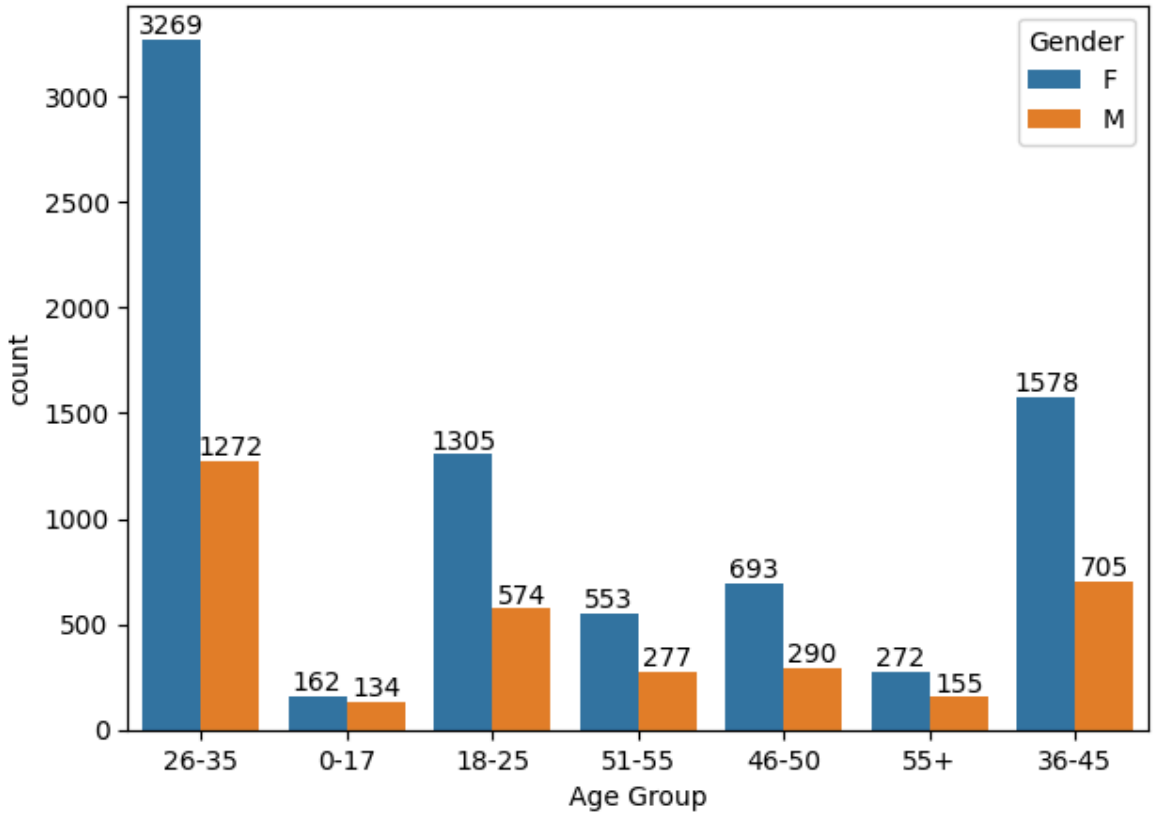
Out[44]: <Axes: xlabel='Gender', ylabel='Amount'>



from above graphs we can see that most of the buyars are females and even the purchesing power of females are greater then man

Age

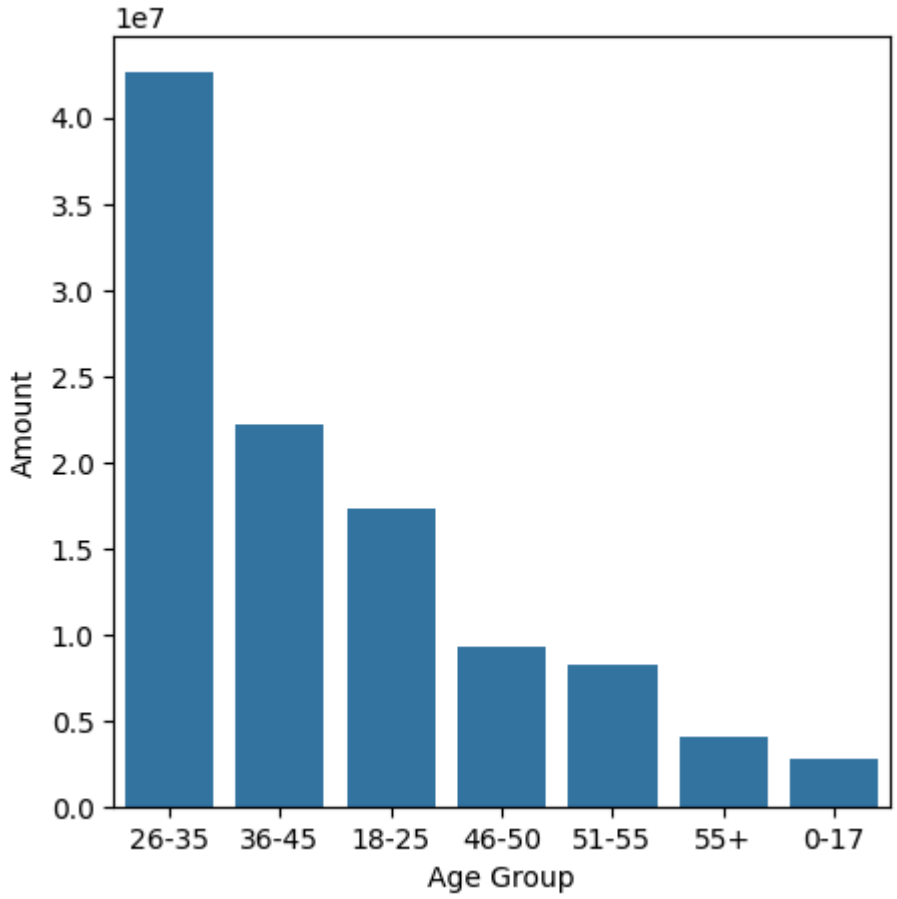
In [50]:
```
plt.figure(figsize=(7,5))
ax=sns.countplot(data= mca , x='Age Group',hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [48]:
```
#Total Amount vs Age Group
plt.figure(figsize=(5,5))
```

```
sales_age=mca.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(
sns.barplot(x='Age Group',y='Amount',data=sales_age)
```

Out[48]: <Axes: xlabel='Age Group', ylabel='Amount'>

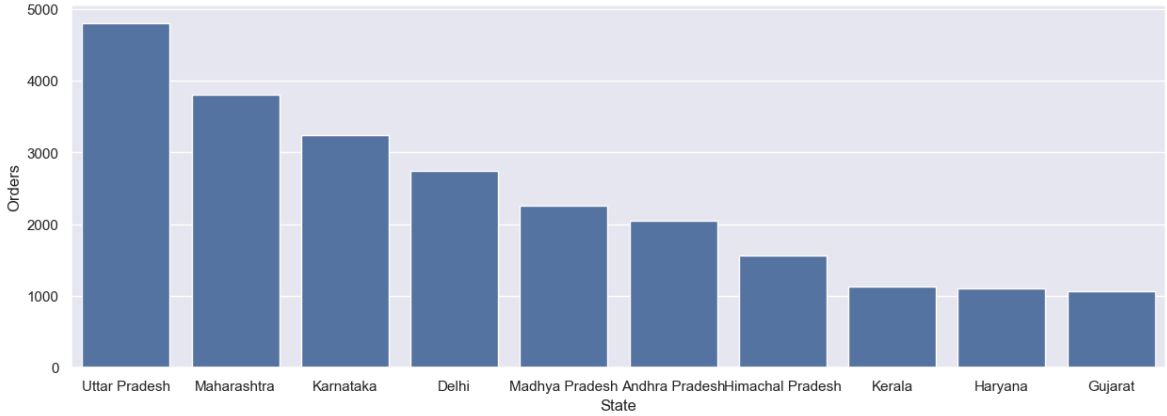


from above graphs we can see that most of the buyers are of age group between 26-35 yrs females

State

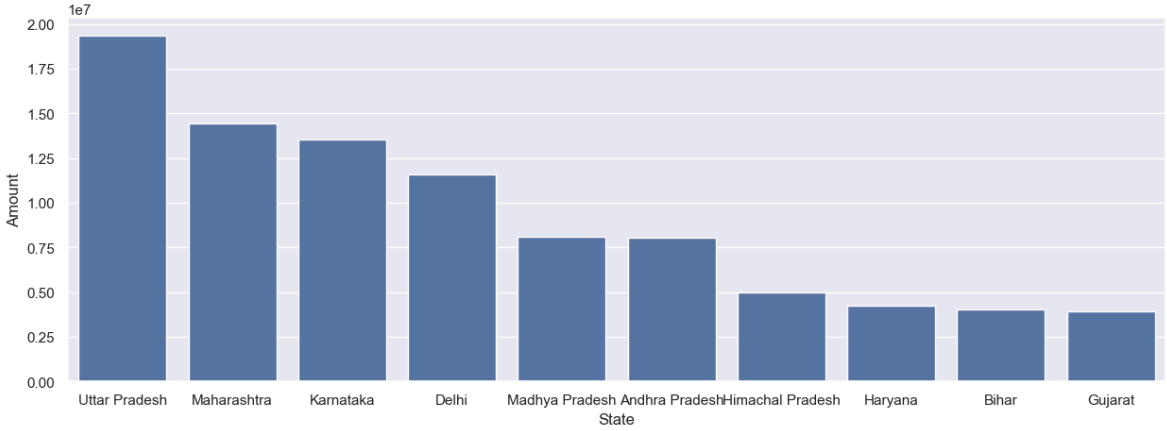
In [51]:
```
# total number of orders from top 10 states
sales_state=mca.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x='State',y='Orders',data=sales_state)
```

Out[51]: <Axes: xlabel='State', ylabel='Orders'>



In [53]:
```
# total amount/sales from top 10 states
sales_state=mca.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x='State',y='Amount',data=sales_state)
```

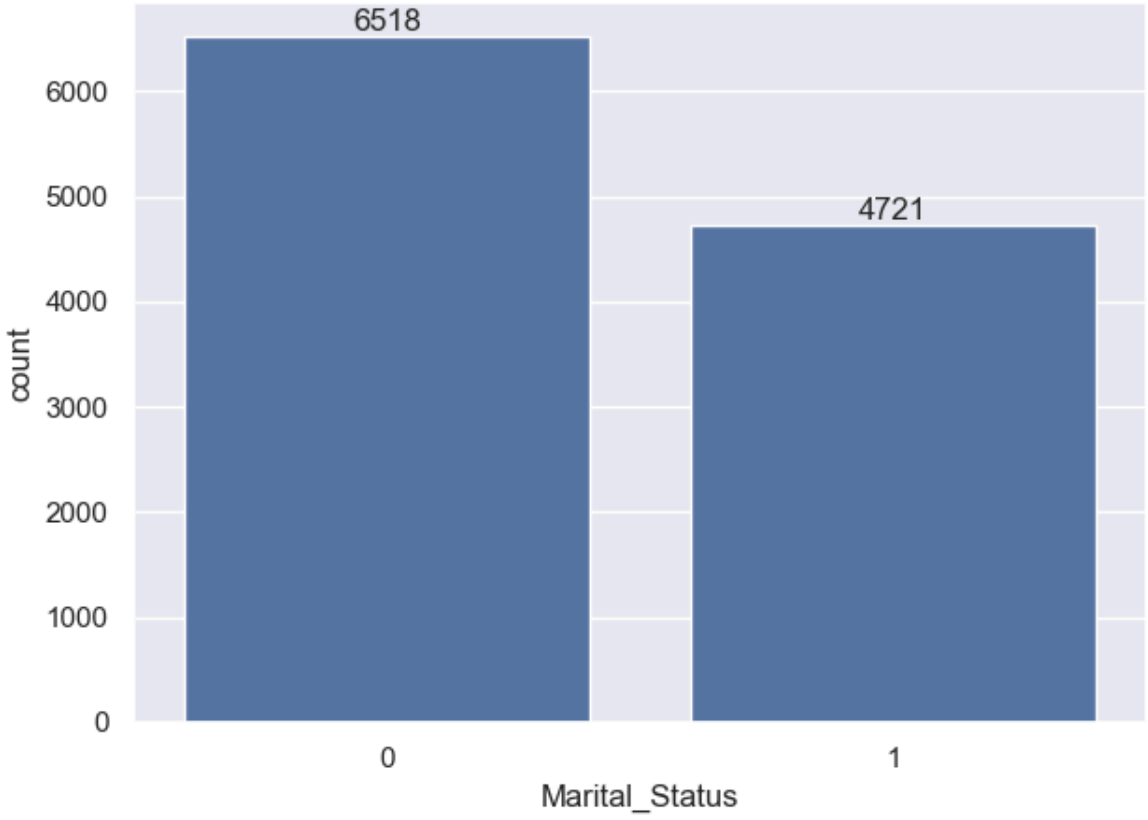
Out[53]: <Axes: xlabel='State', ylabel='Amount'>



from above graphs we can see that most of the orders & total sales/amount are from uttar pradesh,maharashtra and karnataka respectively.

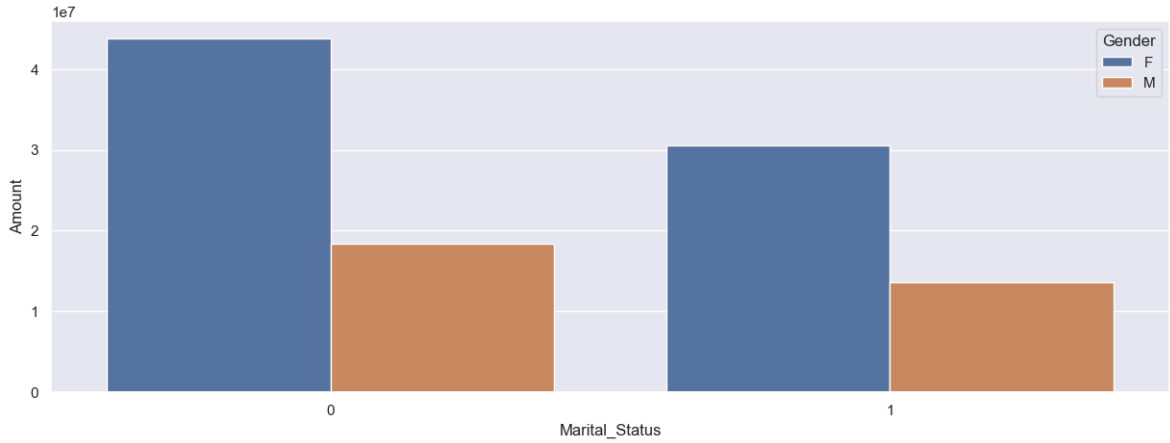
Marital Status

In [55]:
```
plt.figure(figsize=(7,5))
ax=sns.countplot(x='Marital_Status',data=mca)
for bars in ax.containers:
    ax.bar_label(bars)
```



In [56]:
```
sales_state=mca.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].su
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x='Marital_Status',y='Amount',hue='Gender',data=sales_state)
```

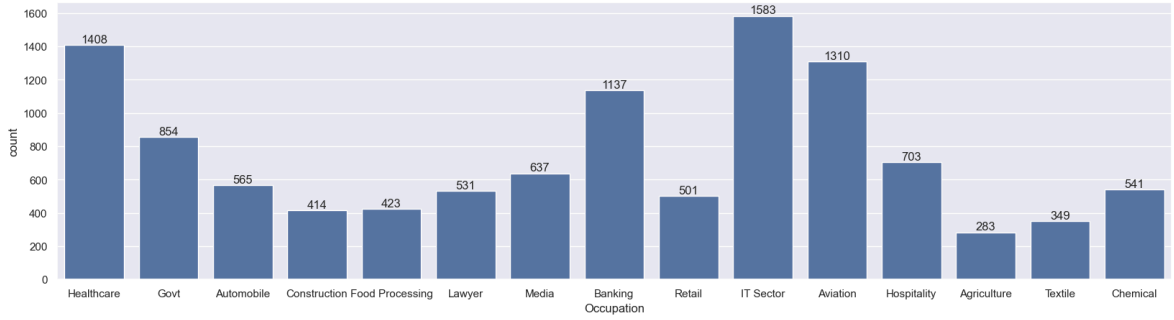
Out[56]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



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

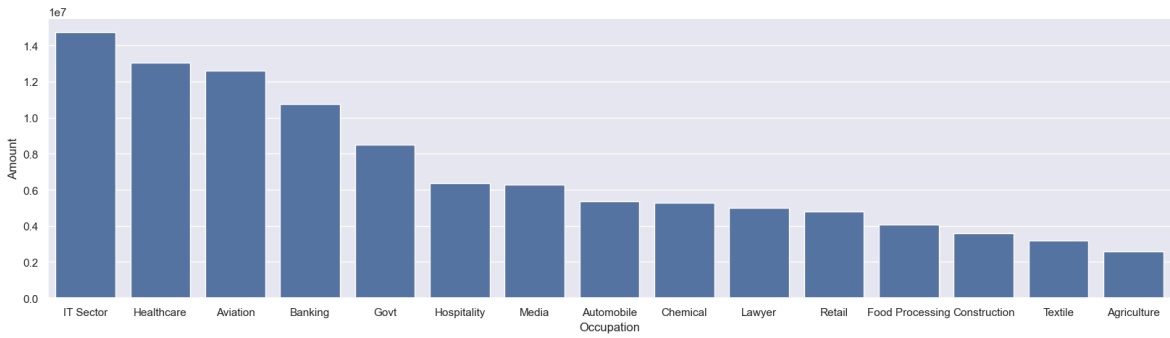
Occupation

In [58]:
```
plt.figure(figsize=(20,5))
ax=sns.countplot(data= mca , x='Occupation')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [61]:
```
sales_state=mca.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_valu
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(x='Occupation',y='Amount',data=sales_state)
```

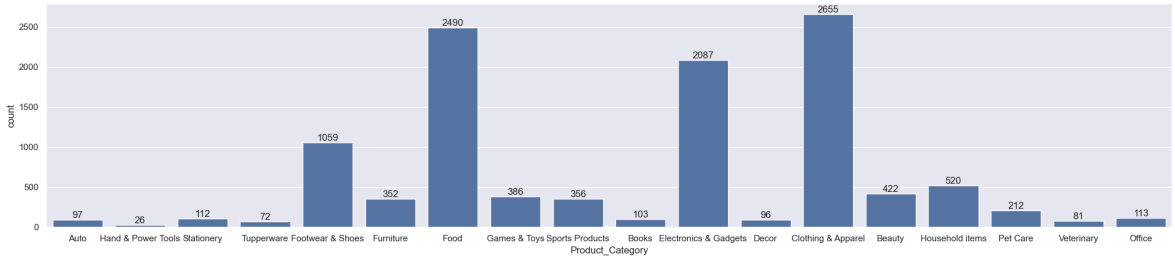
Out[61]: <Axes: xlabel='Occupation', ylabel='Amount'>



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

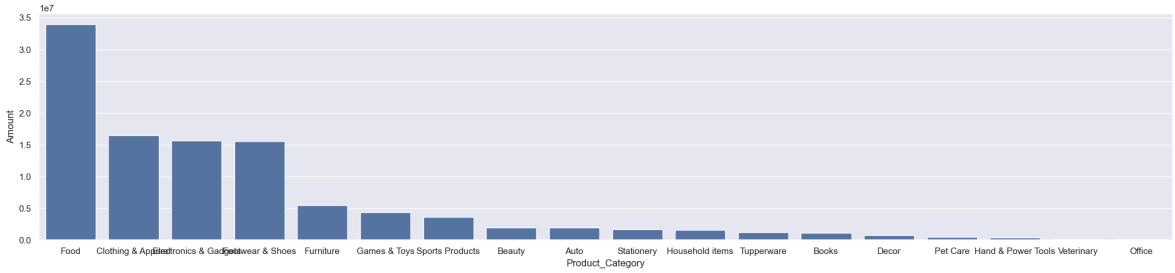
Product Category

In [65]:
```
plt.figure(figsize=(25,5))
ax=sns.countplot(data= mca , x='Product_Category')
for bars in ax.containers:
    ax.bar_label(bars)
```



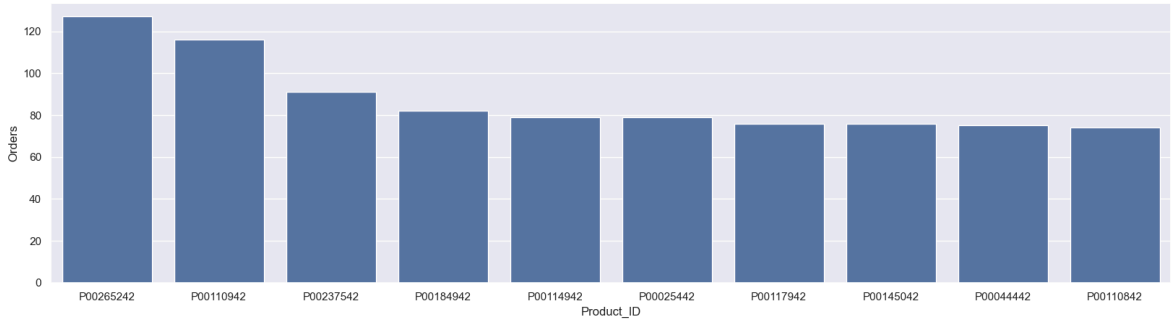
In [67]:
```
sales_state=mca.groupby(['Product_Category'],as_index=False)['Amount'].sum().sor
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(x='Product_Category',y='Amount',data=sales_state)
```

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



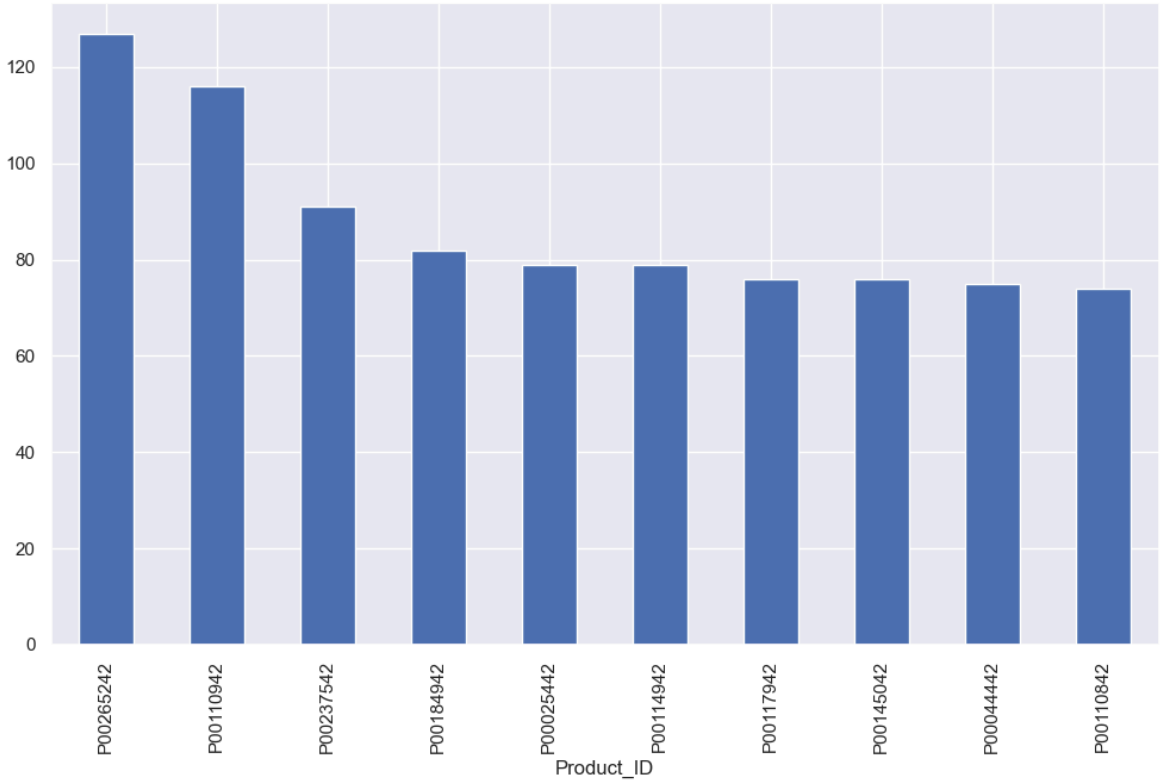
In [68]:
```
sales_state=mca.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_valu
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(x='Product_ID',y='Orders',data=sales_state)
```

Out[68]: <Axes: xlabel='Product_ID', ylabel='Orders'>



In [69]:
```
#top 10 most sold products(same thing as above)
fig1,ax1=plt.subplots(figsize=(12,7))
mca.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=Fal
```

Out[69]: <Axes: xlabel='Product_ID'>



Conclusion

married women are group 26-35 yrs from up maharastra and karnataka working in IT ,helthcare and aviation are more like buy products from food ,clothing and electronics category.

In [ ]: