IMPORTING LIBRARIES

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

In [2]:
```
df=pd.read_csv("Diwali Sales Data.csv",encoding="unicode_escape")
#to avoid unicode error use "unicode_escape"
```

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

Out[3]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	Stat
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtr
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
...	...	...	...	...	...	...	...	.
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtr
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryan
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhy Prades
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnatak
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtr

11251 rows × 15 columns

In [4]:
```
df.shape                                    #to get how many rows and
```

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

In [5]:
```
df.head()                                   #to get upper five record
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	W
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	So
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	(
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	So
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W

In [6]:
```
df.info()                                    #to get the info about the dat
```

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

In [8]:
```
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 [9]:
```
pd.isnull(df)
```

Out[9]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

In [10]:
```
pd.isnull(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
dtype: int64
```

In [11]:
```
df.dropna(inplace=True)                                    #to drop all the
```

In [12]:
```
df.info()                                                    #to check weathe
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.2+ MB
```

In [13]:
```
df.shape                                                     #to check weather the
```

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

In [14]:
```
df.describe()                                                #used to see the numer
```

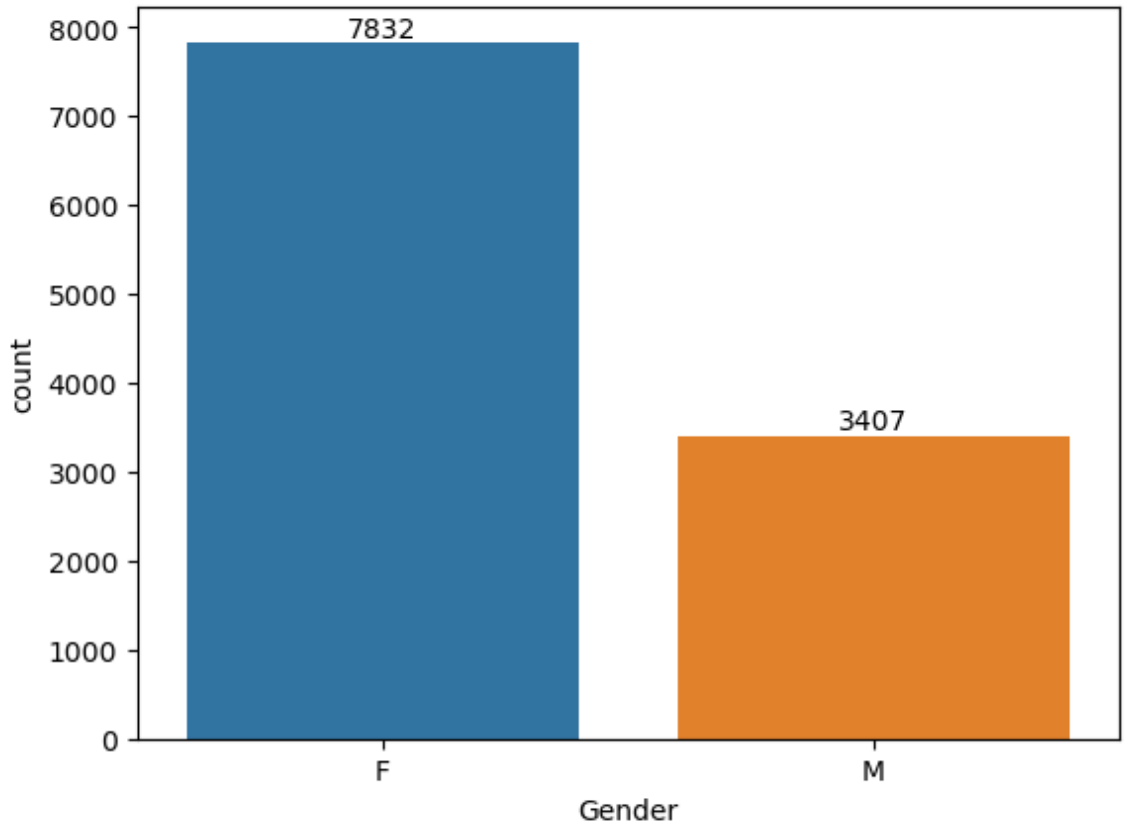
Out[14]:

	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	9453.610858
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355869
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

Gender

In [15]:
```
ax=sns.countplot(x="Gender",data=df)                    #seaborn countplo
for bar in ax.containers:
    ax.bar_label(bar)
```



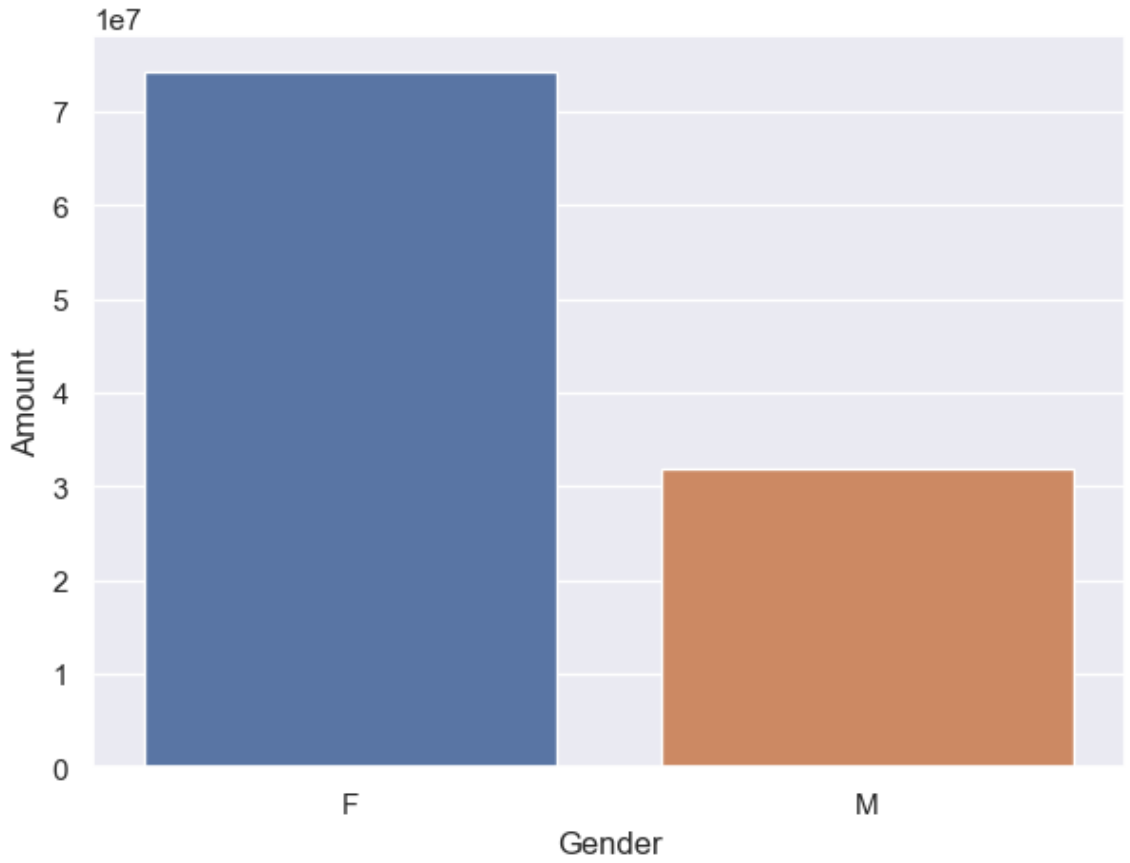
In [16]:
```
df.groupby(["Gender"], as_index=False)["Amount"].sum().sort_values(by="Amou
#grouping by and sorting values to get the analysis according to the gender
```

Out[16]:

	Gender	Amount
0	F	74335856.43
1	M	31913276.00

In [32]:
```
sales_gen=df.groupby(["Gender"], as_index=False)["Amount"].sum().sort_value
sns.barplot(x="Gender",y="Amount",data=sales_gen)           #barplot for an
```

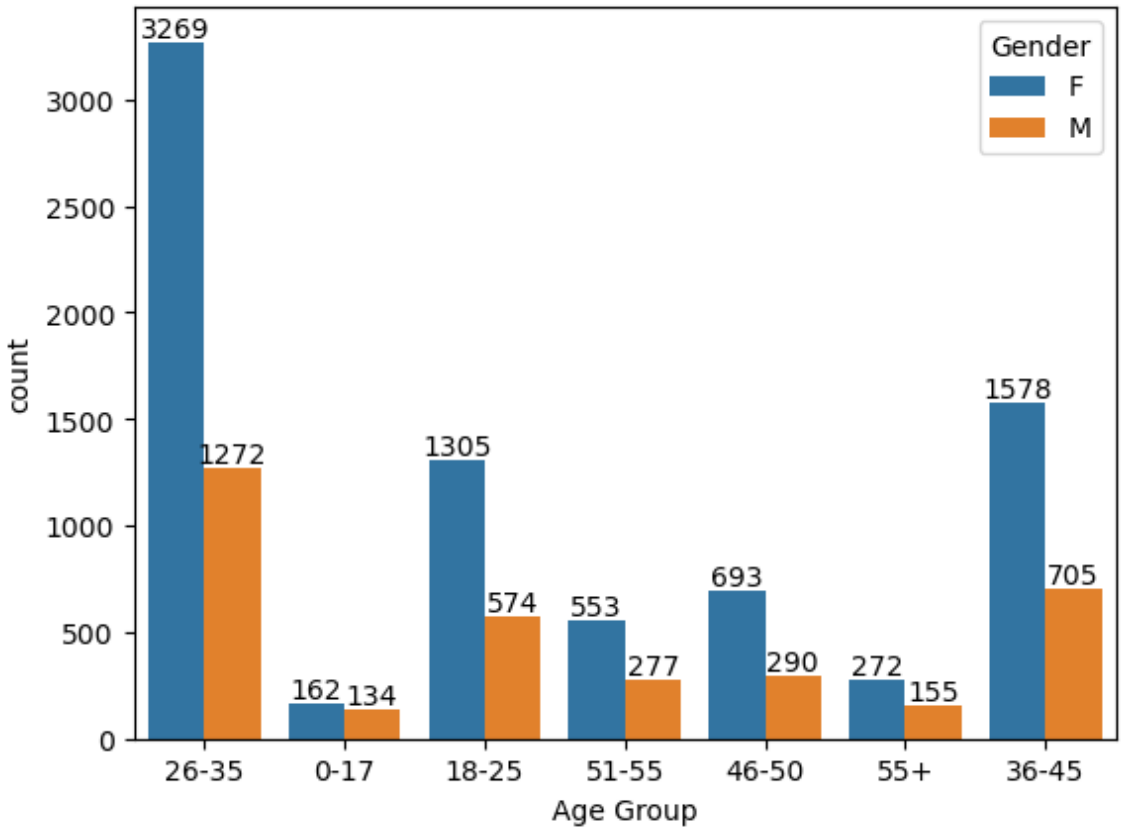
Out[32]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



From the above graps we can say most of the buyers are female and the purchasing power of femnale is much higher then male

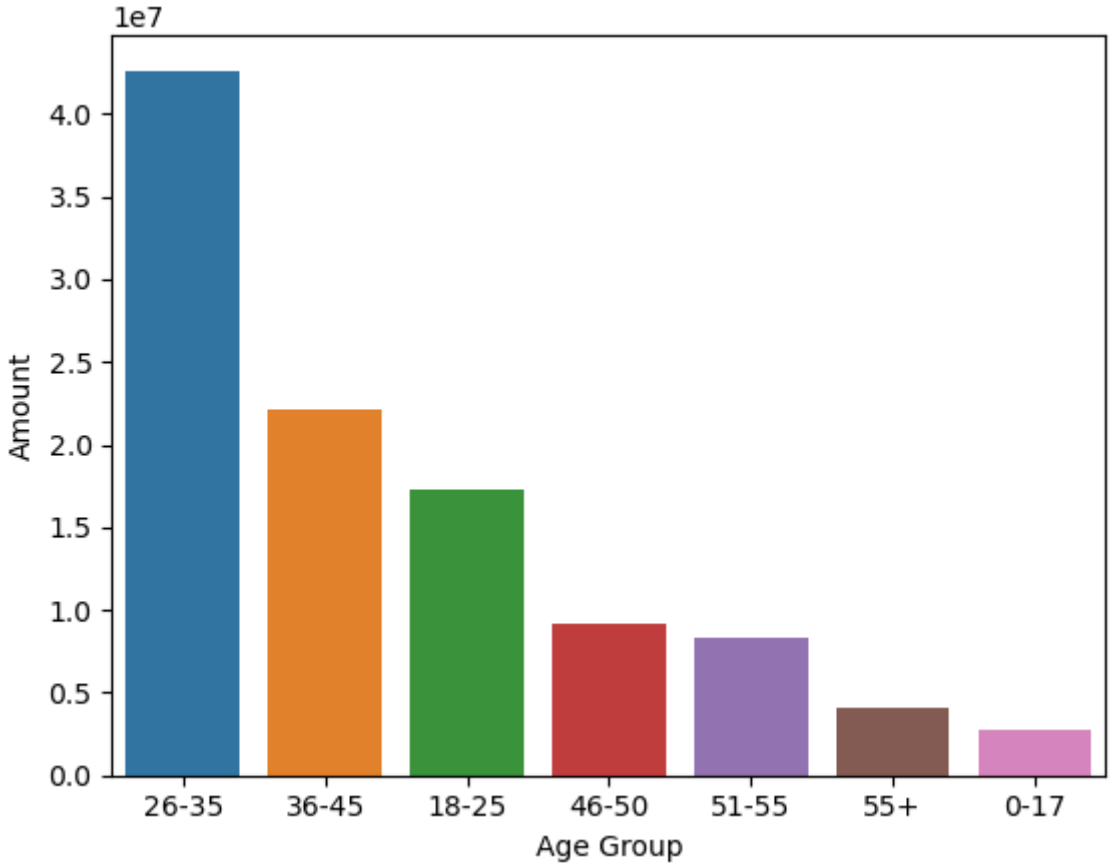
Age

In [18]:
```
ax=sns.countplot(x="Age Group",hue="Gender",data=df)                    #countp
for bar in ax.containers:
    ax.bar_label(bar)
```



In [19]:
```
sales_age=df.groupby(["Age Group"],as_index=False)["Amount"].sum().sort_valu
sns.barplot(x="Age Group",y="Amount",data=sales_age)
#barplot analysis according to the agegroup and amount
```

Out[19]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

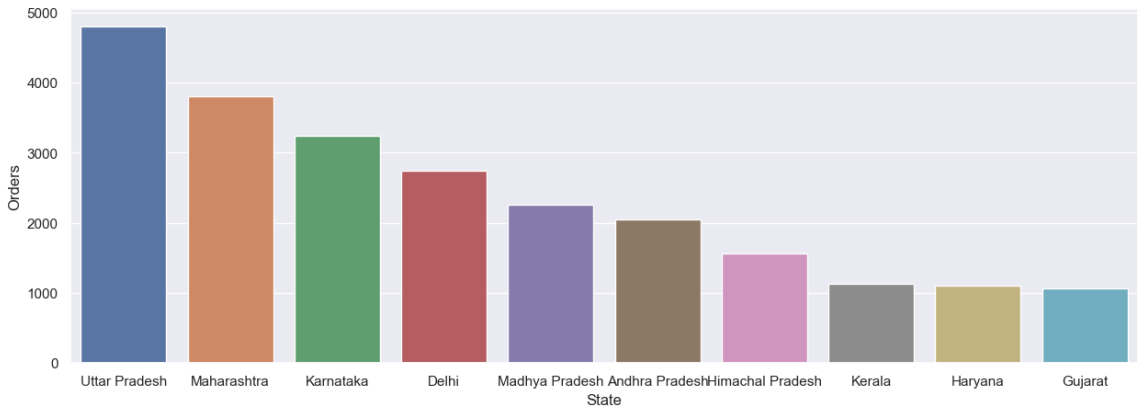


From above Graph we came to know that most of the buyers are between the age group 26-35 and most of them are females

States

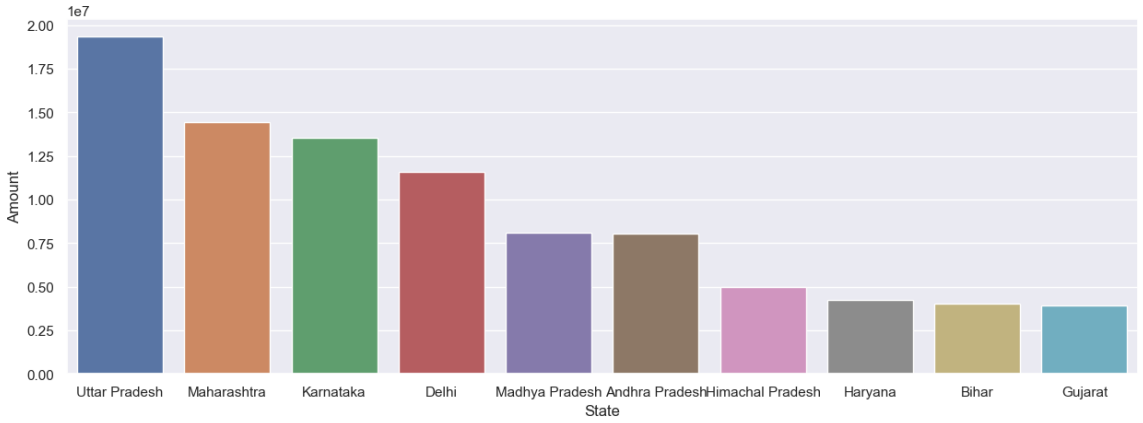
In [20]:
```
sales_state=df.groupby(["State"],as_index=False)["Orders"].sum().sort_value
sns.set(rc={"figure.figsize":(15,5)})
sns.barplot(x="State",y="Orders",data=sales_state)
#barplot analysis aacording to states and orders
```

Out[20]: <Axes: xlabel='State', ylabel='Orders'>



In [21]:
```
sales_state=df.groupby(["State"],as_index=False)["Amount"].sum().sort_value
sns.set(rc={"figure.figsize":(15,5)})
sns.barplot(x="State",y="Amount",data=sales_state)
#barplot analysis according to State and Amount
```

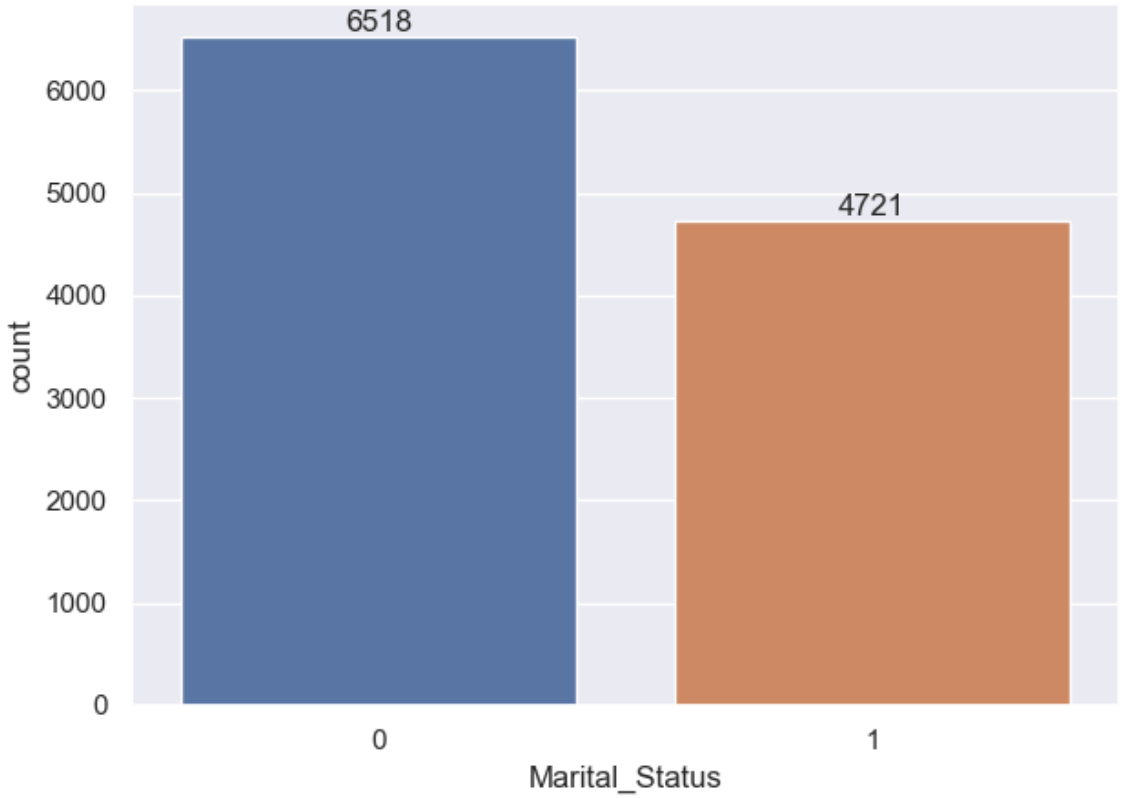
Out[21]: <Axes: xlabel='State', ylabel='Amount'>



From the above graph we can see that most of the orders and Total Sales/Amount are from UttarPradesh,Maharastra and Karnataka respectively

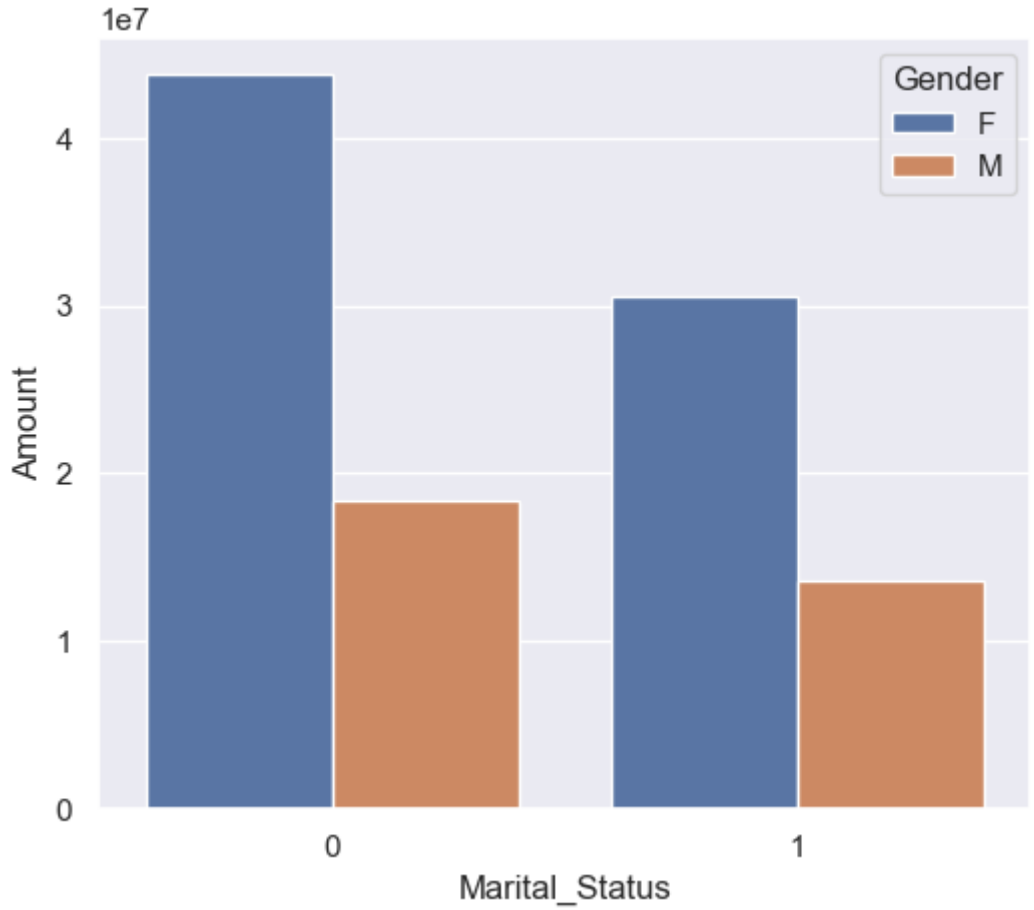
Marital Status

In [30]:
```
ax=sns.countplot(data=df,x="Marital_Status")
sns.set(rc={"figure.figsize":(7,5)})
for bar in ax.containers:
    ax.bar_label(bar)
#countplot graph analysis according to marital status
```



In [23]:
```
sales_state = df.groupby(["Marital_Status", "Gender"], as_index=False)["Amo
sns.set(rc={"figure.figsize":(6,5)})
sns.barplot(x="Marital_Status",y="Amount",hue="Gender",data=sales_state)
#bar graph analysis using marital status and amount distinguished by gender
```

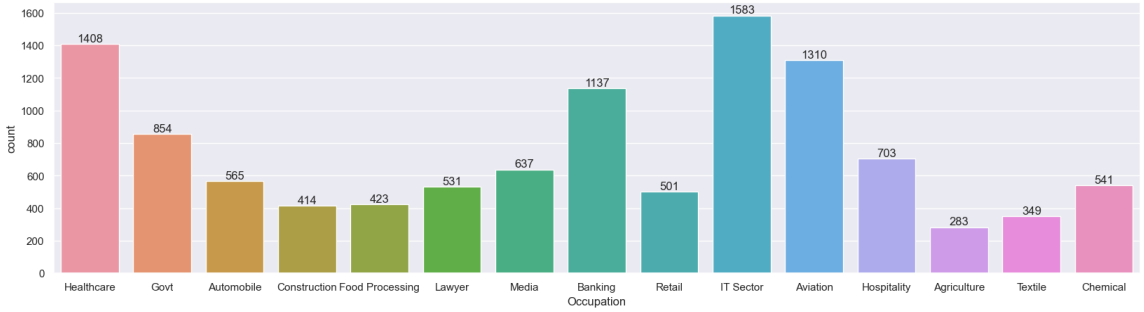
Out[23]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



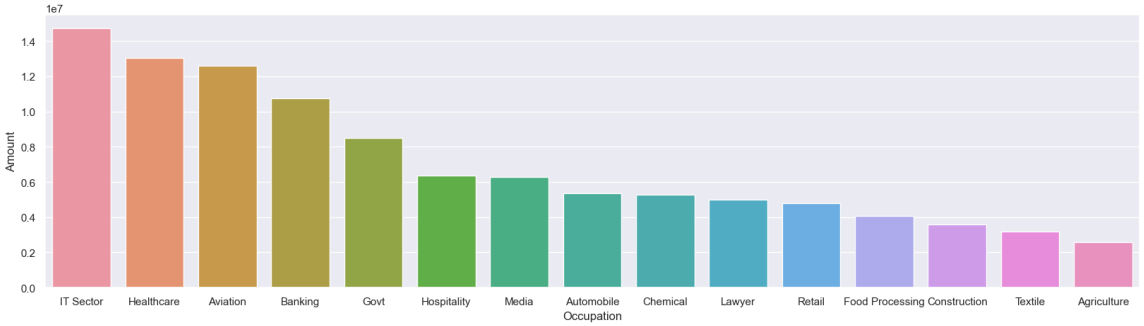
From this above graph we can see that cost of buyers that are married(Female) are more and they have high purchasing power.

Occupation

In [24]:
```
sns.set(rc={"figure.figsize":(20,5)})
ax=sns.countplot(x="Occupation",data=df)
for bar in ax.containers:
    ax.bar_label(bar)
#countplot graph analysis according to occuption
```



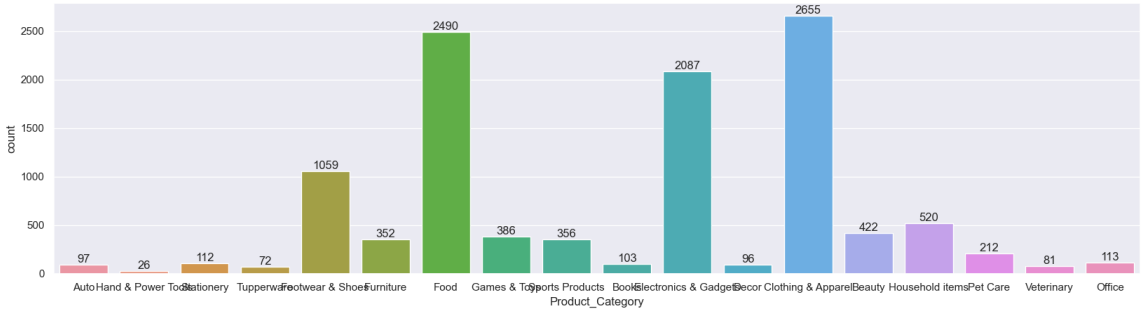
In [25]:
```
sales_state=df.groupby(["Occupation"],as_index=False)["Amount"].sum().sort_
ax=sns.barplot(x="Occupation",y="Amount",data=sales_state)
#barplot analysis according to occupation and amount
```



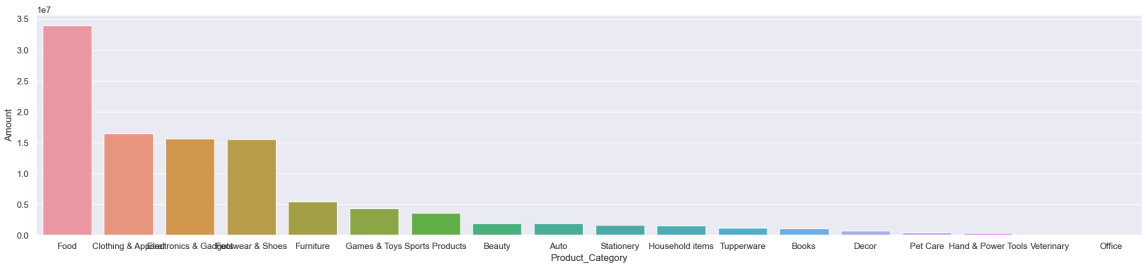
From the above graph we can see most of the buyers are from IT,Healthcare and aviation Sector

Product Category

In [26]:
```
ax=sns.countplot(x="Product_Category",data=df)
sns.set(rc={"figure.figsize":(25,5)})
for bar in ax.containers:
    ax.bar_label(bar)
#countplot analysis according to product category
```



In [27]:
```
sales_state=df.groupby(["Product_Category"],as_index=False)["Amount"].sum()
ax=sns.barplot(x="Product_Category",y="Amount",data=sales_state)
sns.set(rc={"figure.figsize":(30,5)})
#barplot analysis according to product category and amount
```

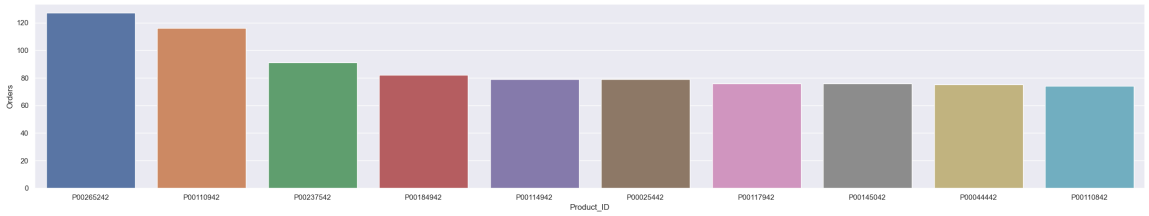


From the above graph we can see that most of the products are from Food,Clothing&Appareal and Electronics&Gadgets

Top 10 Selling Products

In [28]:
```
sales_state=df.groupby(["Product_ID"],as_index=False)["Orders"].sum().sort_
sns.barplot(x="Product_ID",y="Orders",data=sales_state)
#barplot analysis according to product id and orders to get the top 10 produ
```

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



Conclusion

From all the above graphs we come to know that most of the buyers are married women in the age group of 26-35 and they work in IT,Healthcare and Aviation Sector and most of the product category are Food,Clothing&Appearal and Electronic Gadgets

In [ ]: