[77]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

[78]:
```
# import csv file

df = pd.read_csv("C:\\Users\\DELL\\Downloads\\Diwali Sales Data.csv",␣
 ↪encoding='unicode_escape')
df.head()
```

[78]:

	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

1 Data Cleaning

[79]:
```
df.shape
```

[79]:
```
(11251, 15)
```

[80]:
```
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
```

[81]:
```
#drop unrelated/blank columns
df.drop(["Status","unnamed1"], axis=1, inplace=True)
```

[82]:
```
df
```

[82]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	\
0	1002903	Sanskriti	P00125942	F	26-35	28	0	
1	1000732	Kartik	P00110942	F	26-35	35	1	
2	1001990	Bindu	P00118542	F	26-35	35	1	
3	1001425	Sudevi	P00237842	M	0-17	16	0	
4	1000588	Joni	P00057942	M	26-35	28	1	
…	…	…	…	…	…	…	…	
11246	1000695	Manning	P00296942	M	18-25	19	1	
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	

```
11250  1002744      Brumley  P00281742      F     18-25   19                0

                State      Zone       Occupation Product_Category  Orders  \
0         Maharashtra   Western       Healthcare             Auto       1
1      Andhra Pradesh  Southern             Govt             Auto       3
2       Uttar Pradesh   Central       Automobile             Auto       3
3           Karnataka  Southern     Construction             Auto       2
4             Gujarat   Western  Food Processing             Auto       2
…                …       …               …              …     …
11246     Maharashtra   Western         Chemical           Office       4
11247         Haryana  Northern       Healthcare       Veterinary       3
11248  Madhya Pradesh   Central          Textile           Office       4
11249       Karnataka  Southern      Agriculture           Office       3
11250     Maharashtra   Western       Healthcare           Office       3

        Amount
0      23952.0
1      23934.0
2      23924.0
3      23912.0
4      23877.0
…        …
11246    370.0
11247    367.0
11248    213.0
11249    206.0
11250    188.0

[11251 rows x 13 columns]
```

[83]:
```
#check for null values
df.isnull().sum()
```

[83]:
```
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
```

[84]:
```
# dlt null values
df.dropna(inplace=True)
```

[85]:
```
df.shape
```

[85]:
```
(11239, 13)
```

[86]:
```
# change data type
df["Amount"] = df["Amount"].astype("int")
```

[87]:
```
df["Amount"].dtype
```

[87]:
```
dtype('int64')
```

[88]:
```
# describe() method returns description of the data in the DataFrame (i.e.␣
  ↪count, mean, std, etc)
df[["Age","Orders","Amount"]].describe()
```

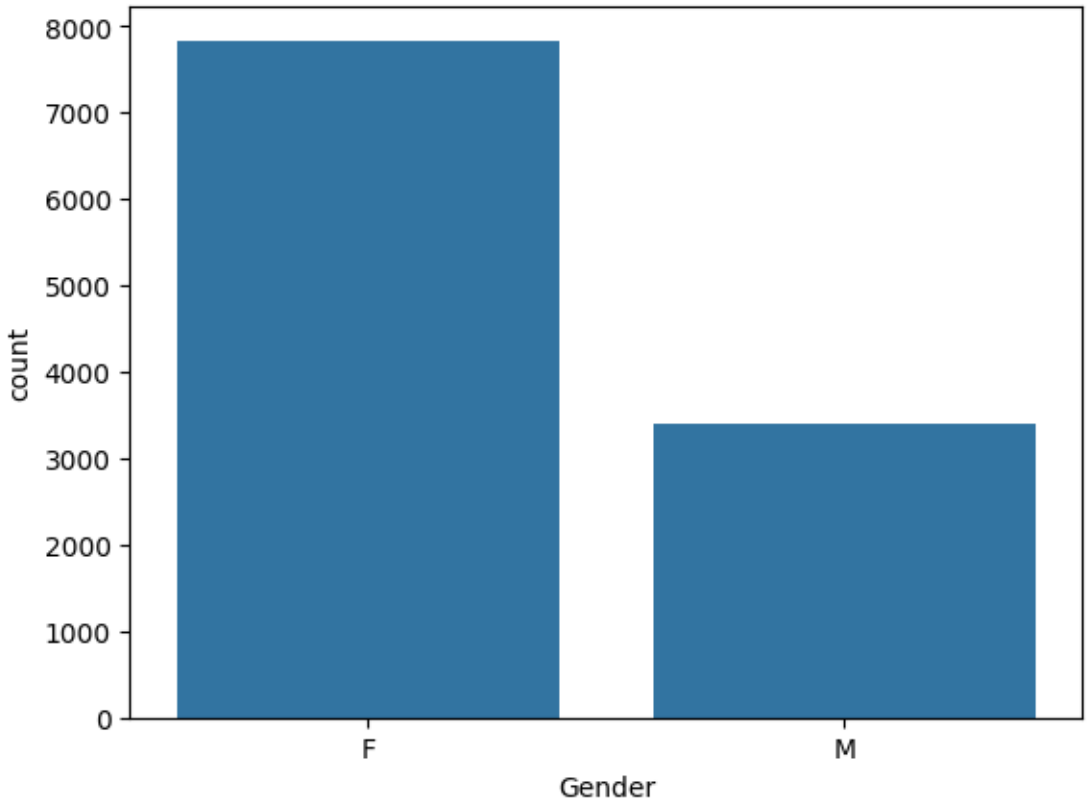
[88]:

	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

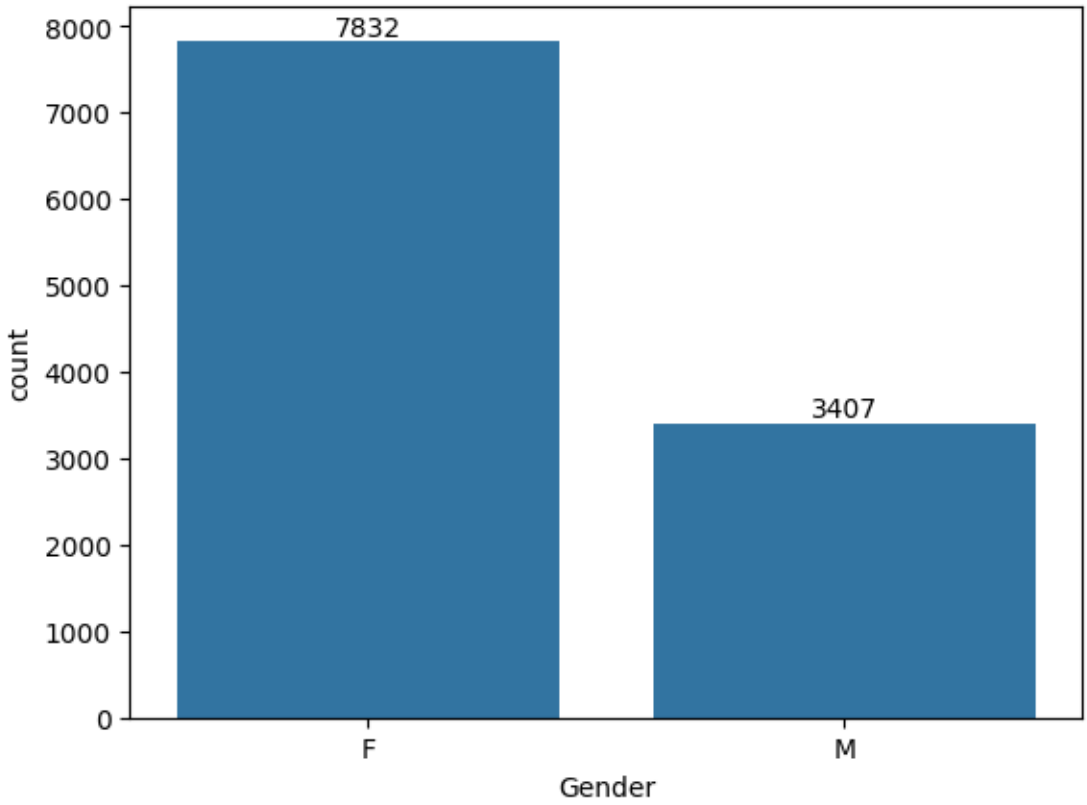
2 Exploratory Data Analysis

2.0.1 Gender

[89]:
```
ax = sns.countplot(x = "Gender", data=df)
```

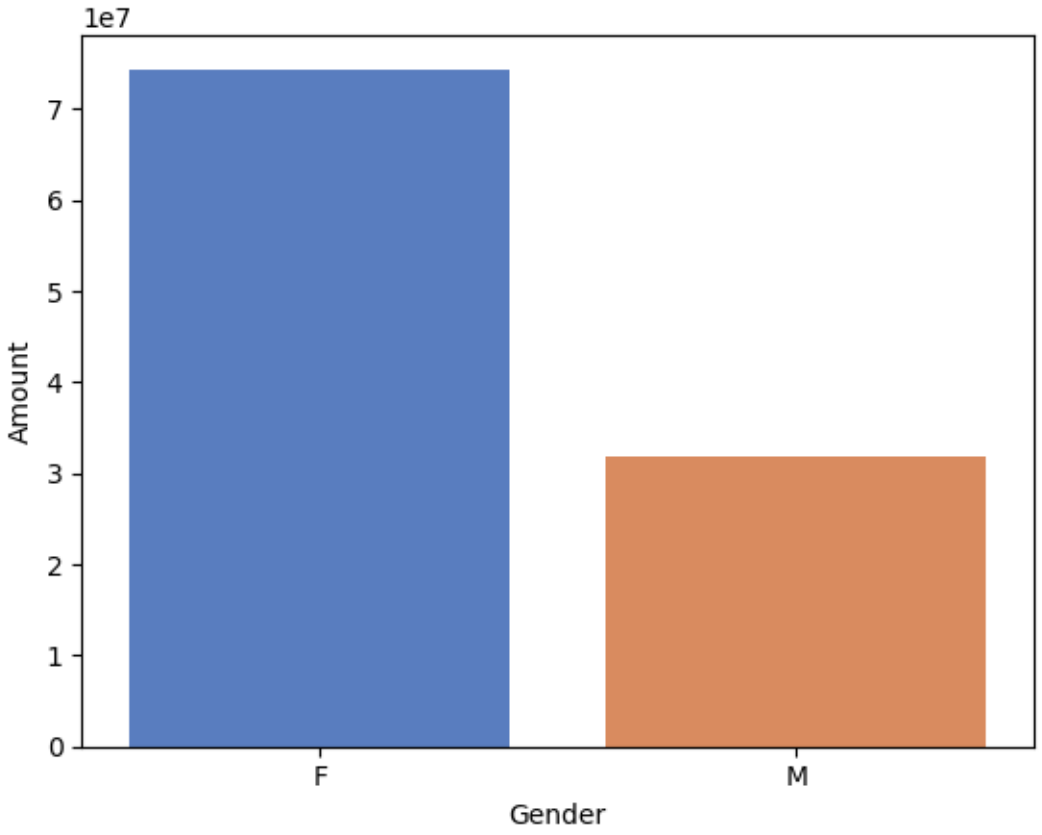


[90]:
```
# plotting a bar chart for Gender and it's count
ax = sns.countplot(x = "Gender", data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```



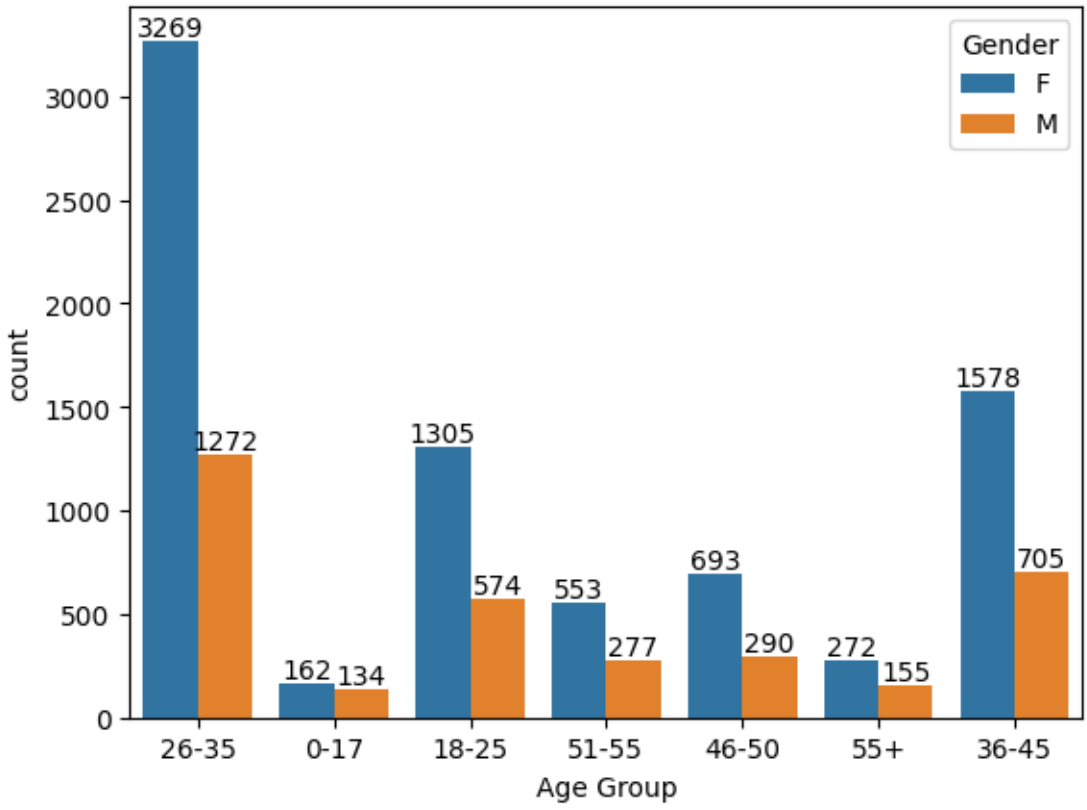
[91]:
```
# plotting a bar chart for gender vs total amount
sales_gen = df.groupby(["Gender"], as_index=False)["Amount"].sum().sort_values(␣
 ↪by = "Amount",ascending=False)
sns.barplot(x="Gender",y="Amount", data=sales_gen, hue="Gender",␣
 ↪palette="muted")
```

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



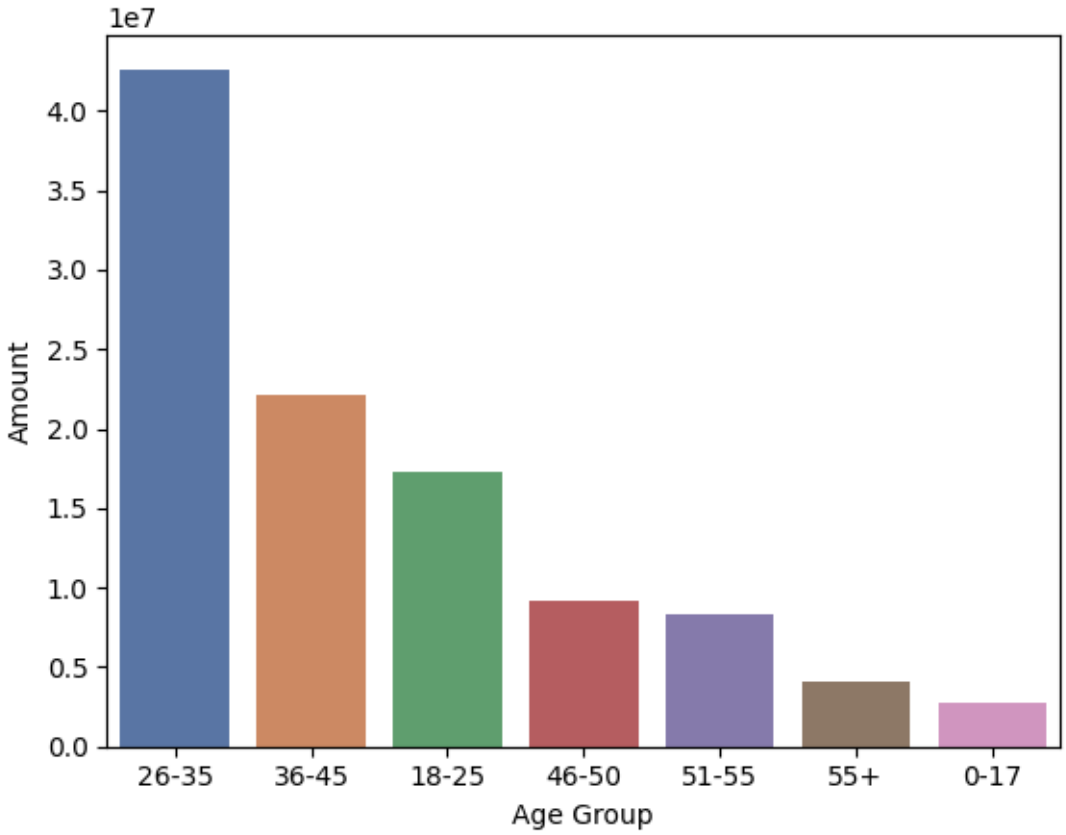
2.0.2 Age

[92]:
```
ax = sns.countplot(x = "Age Group", data=df, hue="Gender")
for bars in ax.containers:
    ax.bar_label(bars)
```



[93]:
```
# Total Amount 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, hue="Age Group",␣
 ↪palette="deep")
```

[93]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

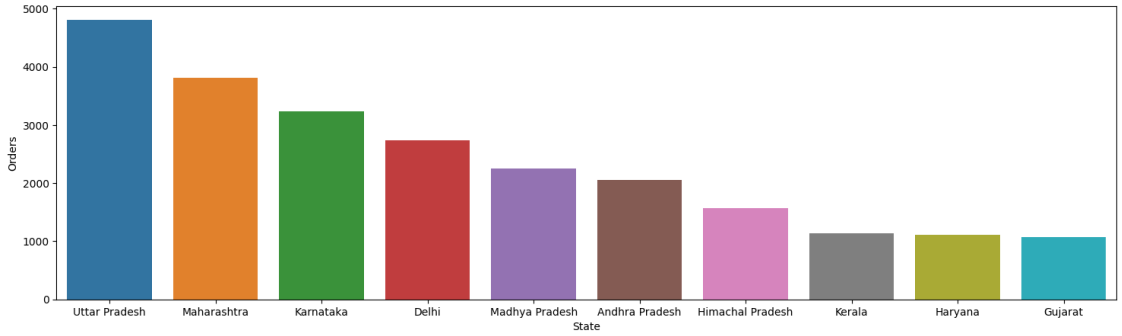


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

2.0.3 State

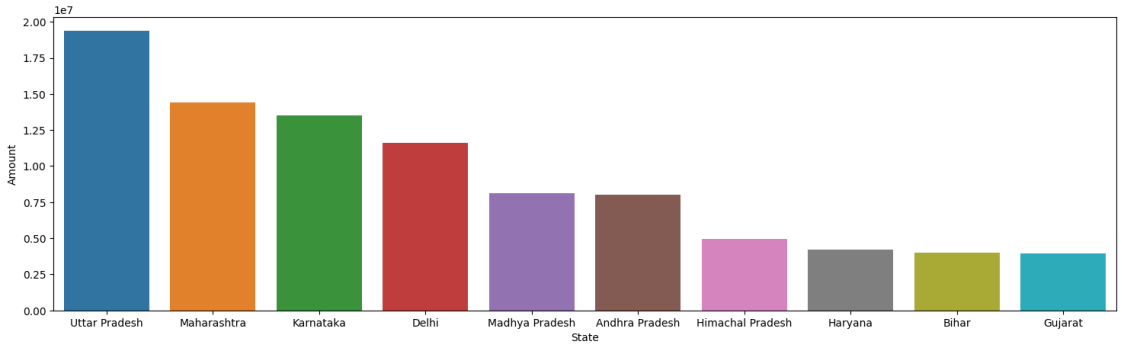
[94]:
```
# total orders from top 10 states
sales_state = df.groupby(["State"], as_index=False)["Orders"].sum().
  ↪sort_values(by="Orders", ascending=False).head(10)
plt.figure(figsize=(18, 5))
sns.barplot(x="State", y="Orders", hue="State", data=sales_state)
```

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



[95]:
```
# total amount/sales from top 10 states
ord_state = df.groupby(["State"], as_index=False)["Amount"].sum().
 ↪sort_values(by="Amount", ascending=False).head(10)
plt.figure(figsize=(18, 5))
sns.barplot(x="State", y="Amount", hue="State", data=ord_state)
```

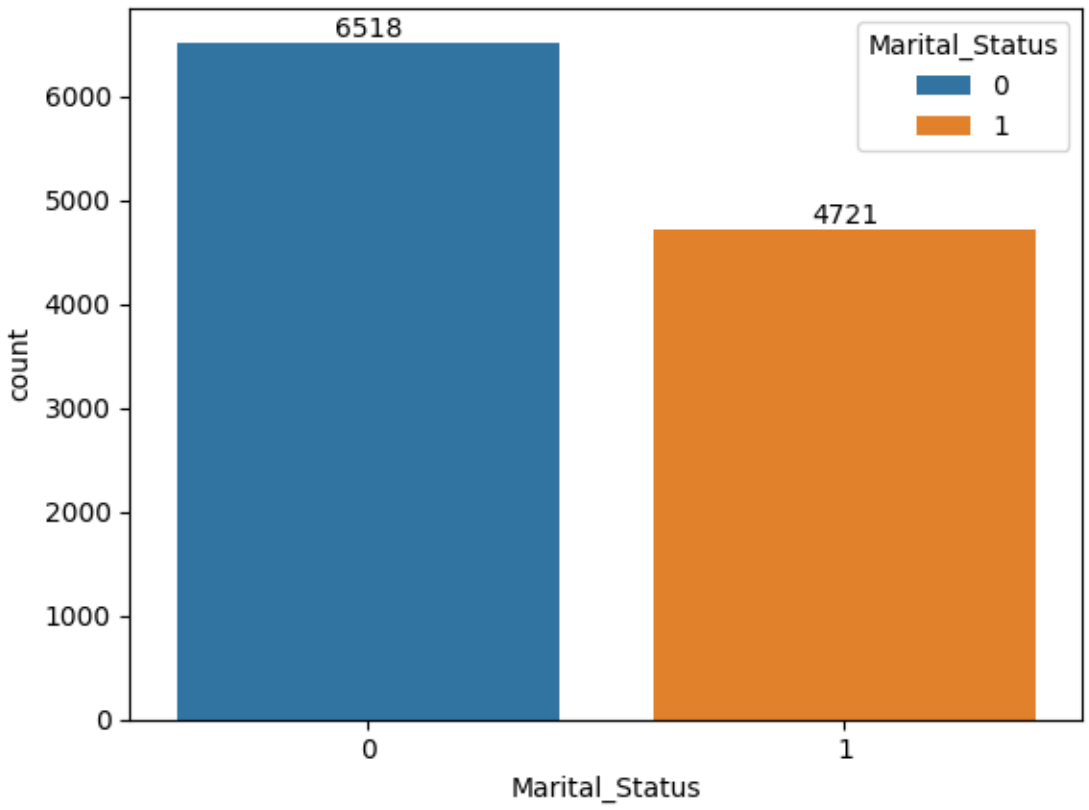
[95]: <Axes: xlabel='State', ylabel='Amount'>



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

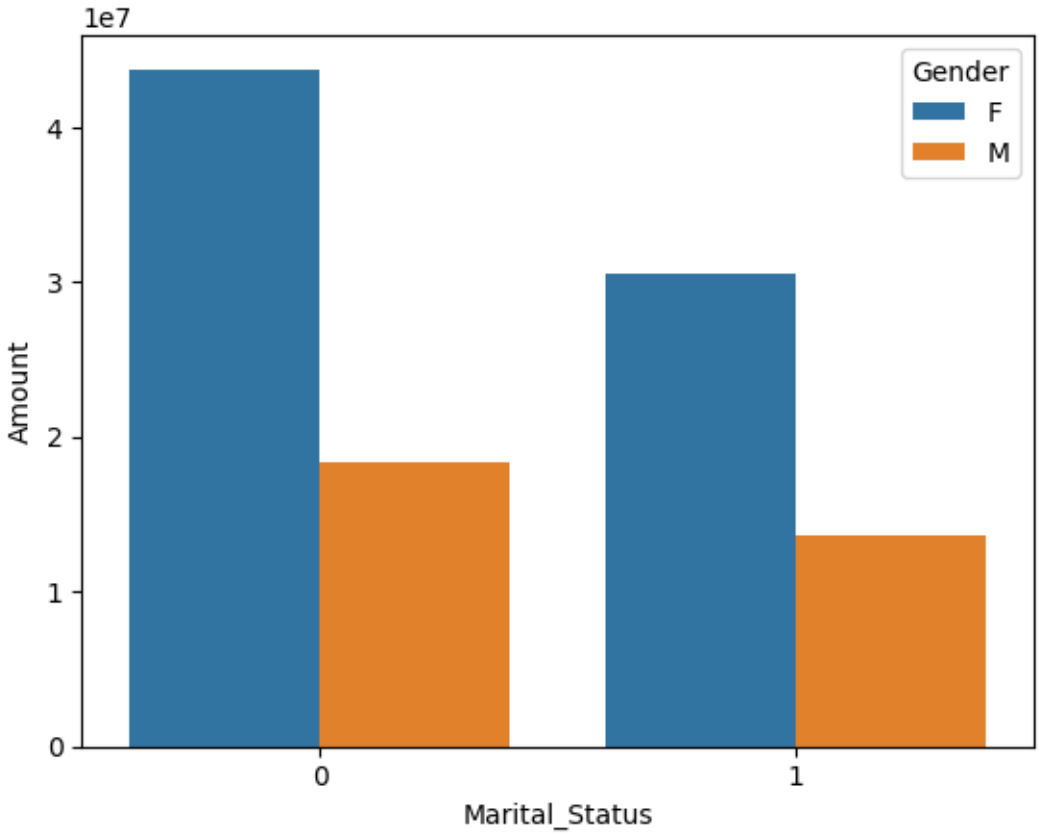
2.0.4 Marital Status

[96]:
```
ax = sns.countplot(x = "Marital_Status", data=df, hue="Marital_Status")
for bars in ax.containers:
    ax.bar_label(bars)
```



[97]:
```
married = 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=married)
```

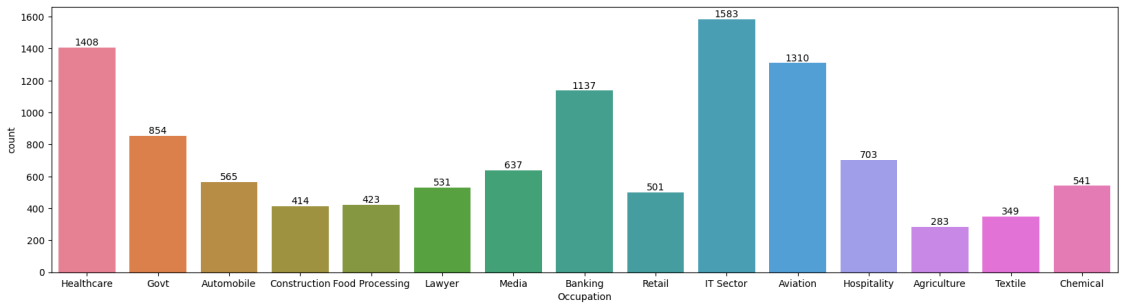
[97]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



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

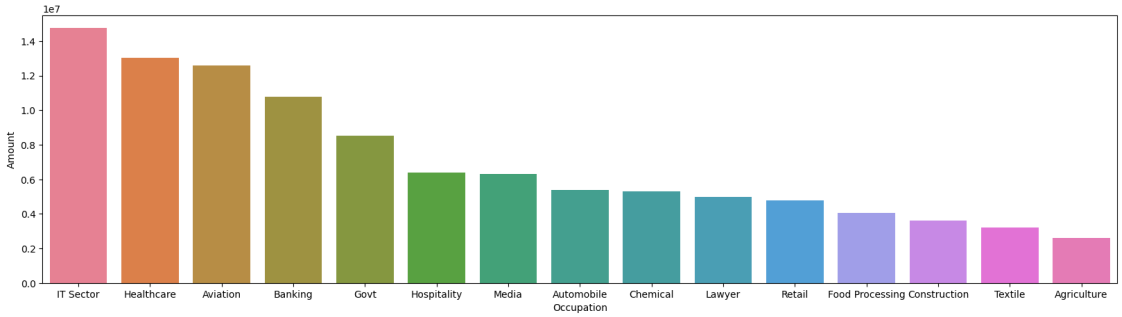
2.0.5 Occupation

[98]:
```
plt.figure(figsize=(20,5))
ax = sns.countplot(x="Occupation", data=df, hue="Occupation")
for bars in ax.containers:
    ax.bar_label(bars)
```



[99]:
```
occ = df.groupby(["Occupation"], as_index=False)["Amount"].sum().
 ↪sort_values(by="Amount", ascending=False)
plt.figure(figsize=(20,5))
sns.barplot(x="Occupation", y="Amount", data=occ, hue="Occupation")
```

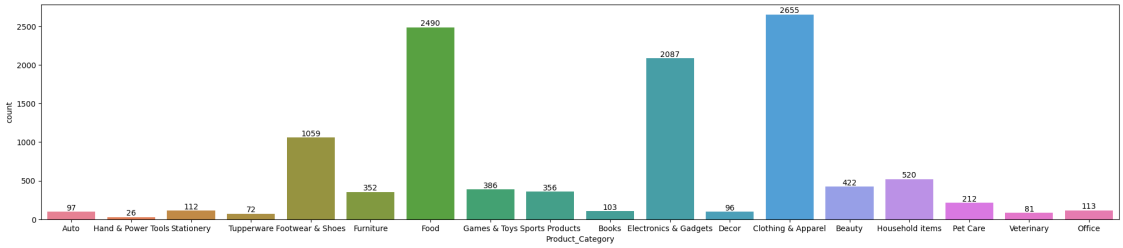
[99]: <Axes: xlabel='Occupation', ylabel='Amount'>



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

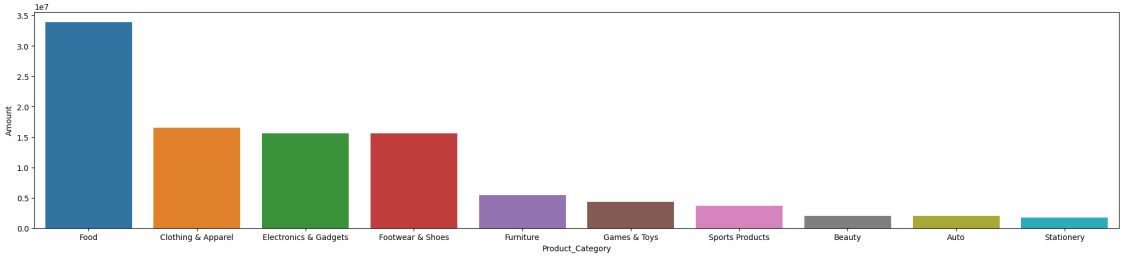
2.0.6 Product Category

[100]:
```
plt.figure(figsize=(25,5))
ax = sns.countplot(x="Product_Category", data=df, hue="Product_Category")
for bars in ax.containers:
    ax.bar_label(bars)
```



[101]:
```
prod = df.groupby(["Product_Category"], as_index=False)["Amount"].sum().
 ↪sort_values(by="Amount",ascending=False).head(10)
plt.figure(figsize=(25,5))
sns.barplot(x = "Product_Category", y = "Amount", data=prod,␣
 ↪hue="Product_Category")
```

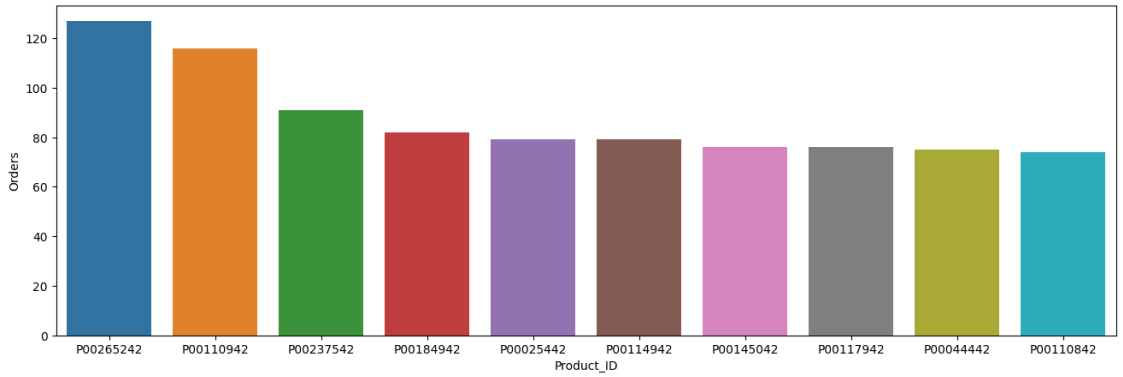
[101]: <Axes: xlabel='Product_Category', ylabel='Amount'>



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

[102]:
```
# top 10 most sold products
most_sell_prod = df.groupby(["Product_ID"], as_index=False)["Orders"].sum().
 ↪sort_values(by="Orders",ascending=False).head(10)
plt.figure(figsize=(16,5))
sns.barplot(x = "Product_ID", y = "Orders", data=most_sell_prod,␣
 ↪hue="Product_ID")
```

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



2.0.7 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