```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/Diwali Sales Data.csv', encoding="unicode_escape")
df.shape
```

```
(11251, 15)
```

```
df.head()
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amc
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	239
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	239
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	239.
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	239
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	238

Next steps: Generate code with df   View recommended plots   New interactive sheet

```
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
```

```
#drop unrealted/blank column
df.drop(['Status','unnamed1'],inplace=True,axis=1)
```

```
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
```

```
#check for null values
pd.isnull(df).sum()
```

	0
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

```
df.shape
```

```
(11251, 13)
```

```
#drop null values
df.dropna(inplace=True)
```

```
df.shape
```

```
(11239, 13)
```

```
#change data type
df['Amount']=df['Amount'].astype('int')
```

```
df['Amount'].dtypes
```

```
dtype('int64')
```

```
df.columns
```

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

```
#rename column
df.rename(columns={'Marital_Status':'Shadi'})
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shadi	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

```
#describre method returns description of the data in the dataframe(count,mean.std etc)
df.describe()
```

	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

```
#use describe() for specific column
df[['Age','Orders','Amount']].describe()
```

	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

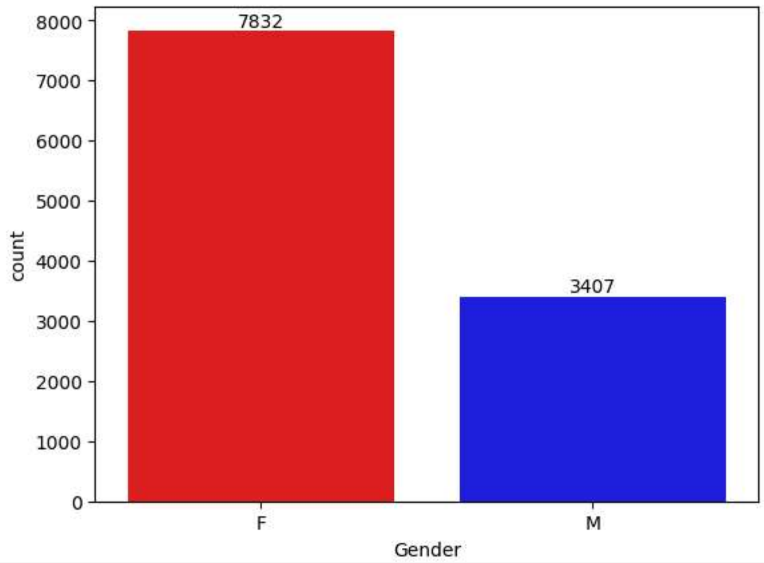
Exploratory Data Analysis

1. Gender

```
df.columns
```

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

```
ax = sns.countplot(x='Gender',data=df,hue='Gender', palette={'M': 'blue', 'F': 'red'})

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

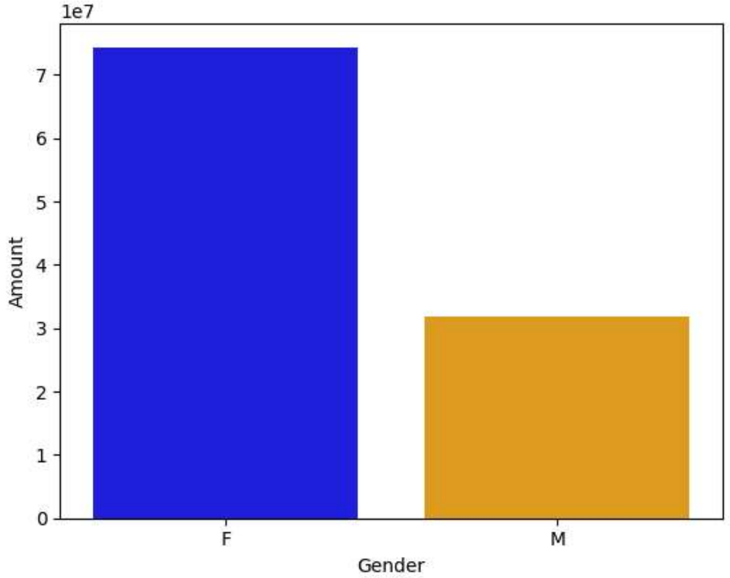


```
df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
```

	Gender	Amount
0	F	74335853
1	M	31913276

```
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={'M': 'orange','F':'blue'})
```

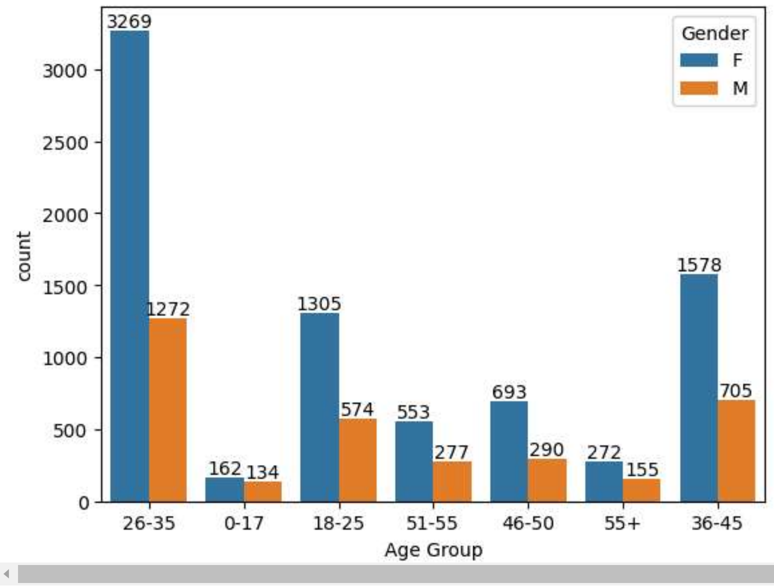
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



From the above Graphs, we can see that females has made more purchases than men.

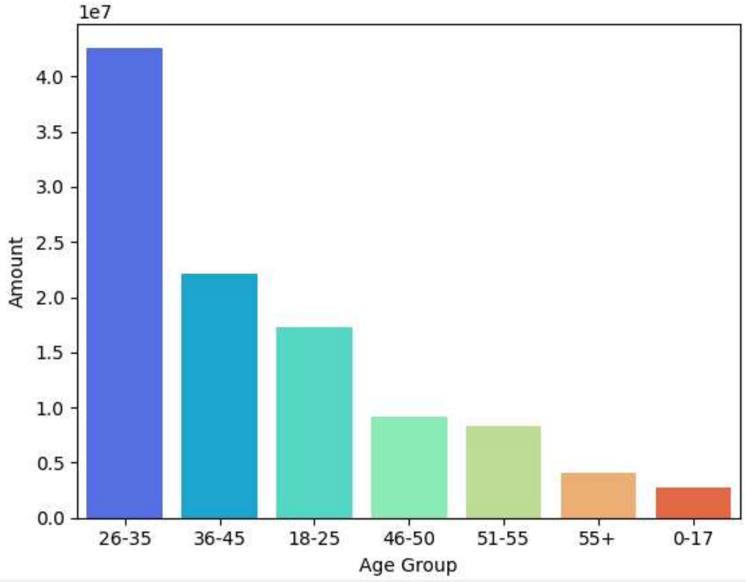
2. Age

```
ax=sns.countplot(x='Age Group',data=df,hue='Gender')

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



```
#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='rainbow')
```

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

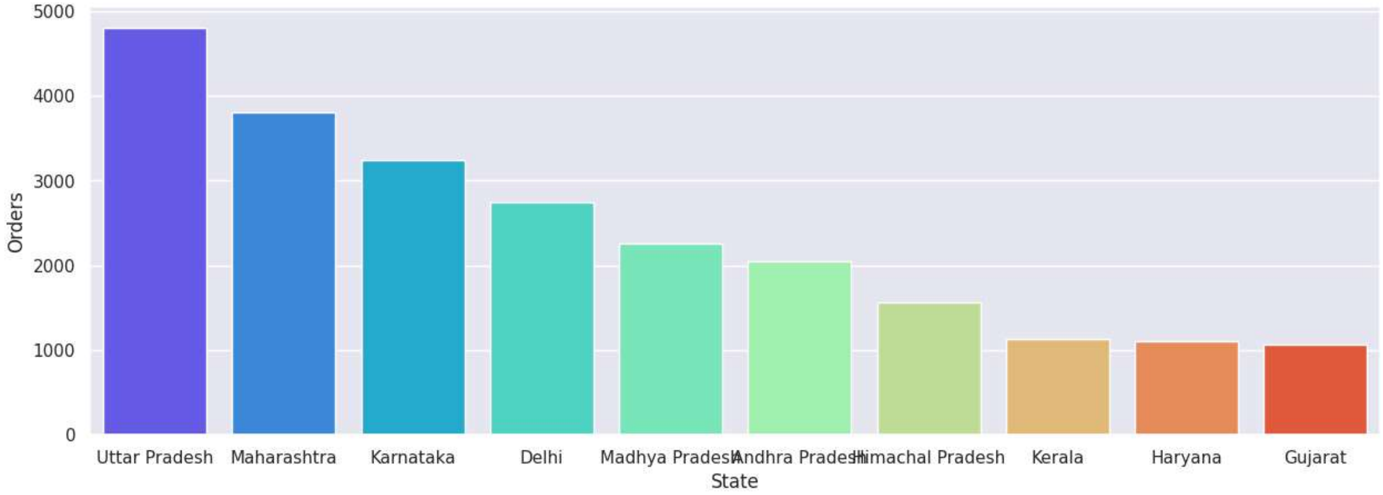


From above Graph we can see that most of the buyers are of age group between 26-35 yrs female

3. State

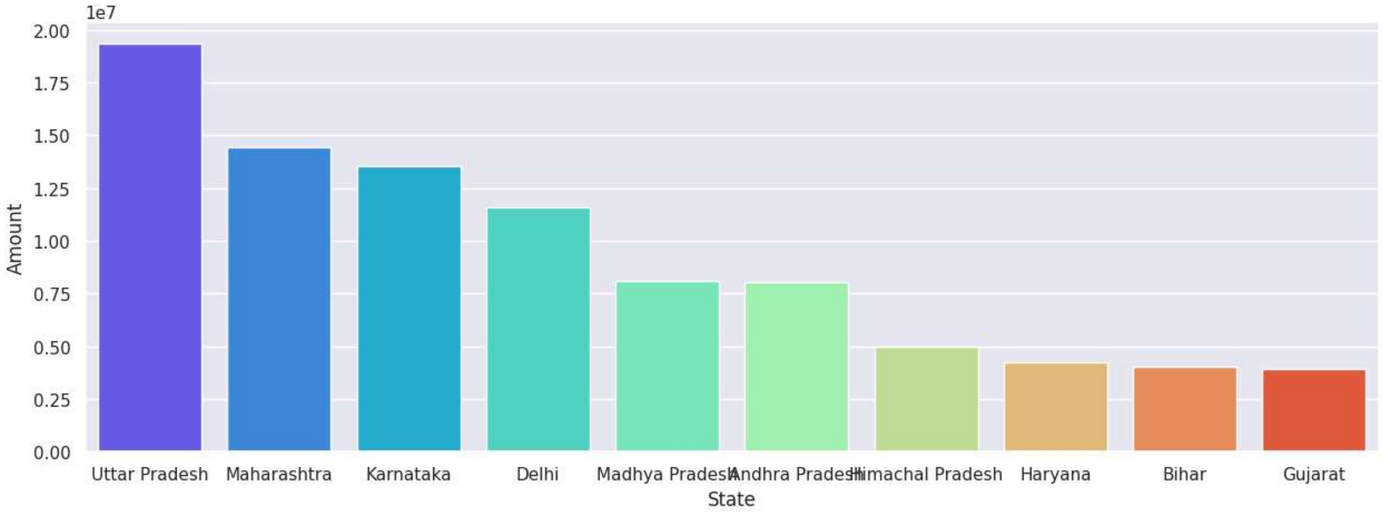
```
#total number of orders from top 10 states
sales_state = df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x ='State',y='Orders',data=sales_state,hue='State',palette='rainbow')
```

```
<Axes: xlabel='State', ylabel='Orders'>
```



```
#total number of Amount/sales from top 10 states
sales_state = df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x ='State',y='Amount',data=sales_state,hue='State',palette='rainbow')
```

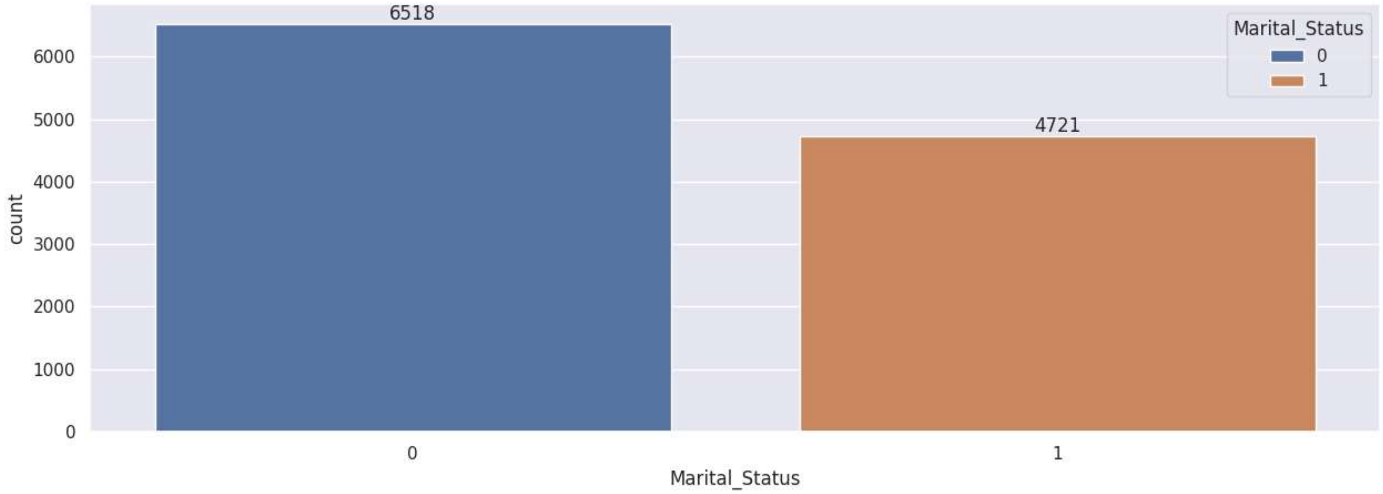
```
<Axes: xlabel='State', ylabel='Amount'>
```



From above graph we can see that most of the orders & sales/amount are from Uttar Pradesh , Maharastra , karnataka respectively.

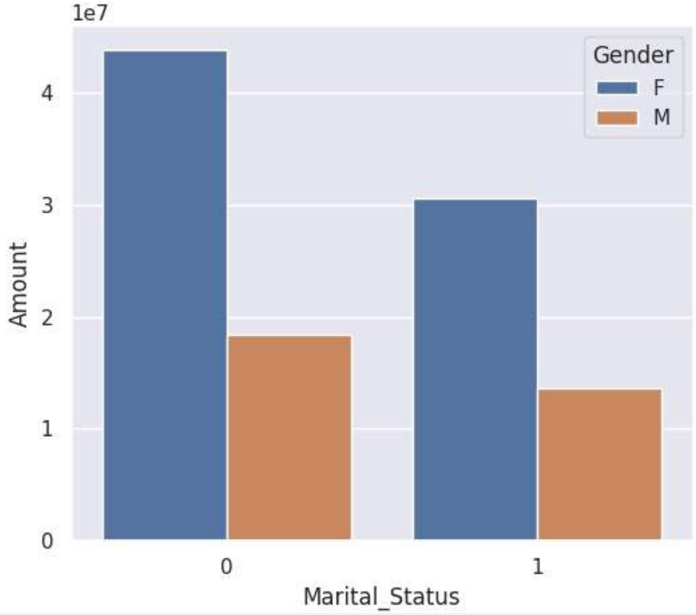
4. Marital Status

```
ax=sns.countplot(x='Marital_Status',data=df,hue='Marital_Status')

sns.set(rc={'figure.figsize':(6,3)})
for bars in ax.containers:
    ax.bar_label(bars)
```



```
sales_ms = df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(x ='Marital_Status',y='Amount',data=sales_ms,hue='Gender')
```

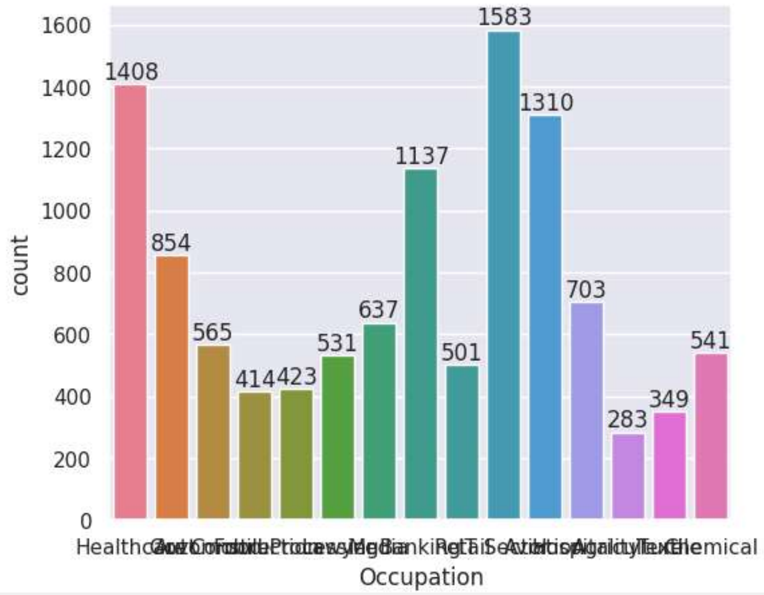
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



From the above graph we can see that most of the buyers are married women and they have high purchasing power.

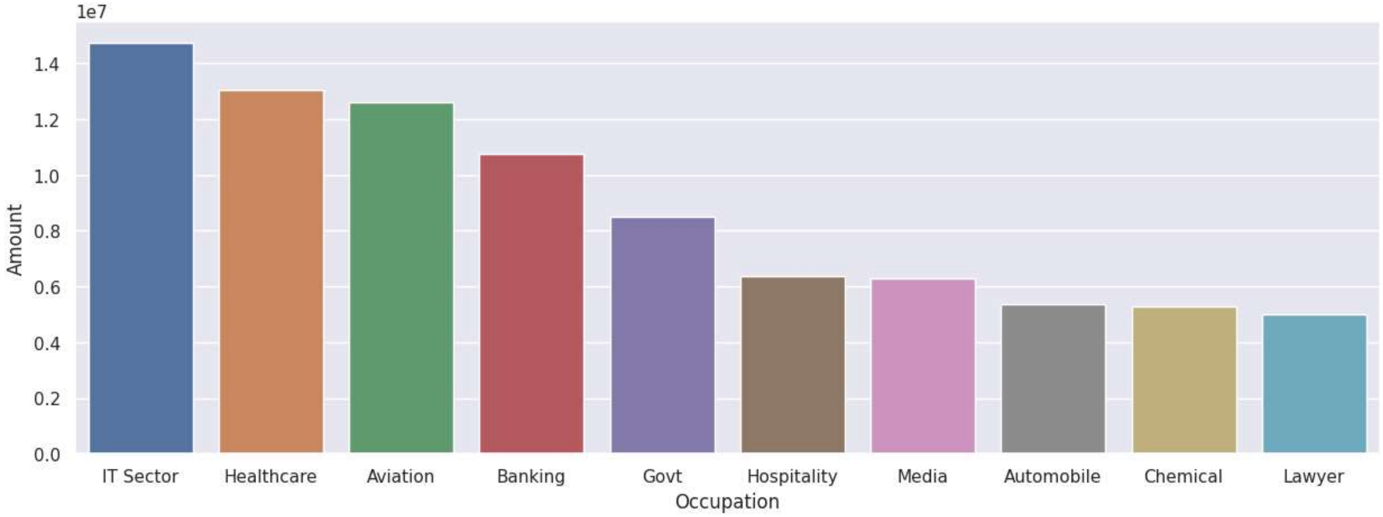
5. Occupation

```
ax=sns.countplot(x='Occupation',data=df,hue='Occupation')

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



```
sales_occupation = df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x ='Occupation',y='Amount',data=sales_occupation,hue='Occupation')
```

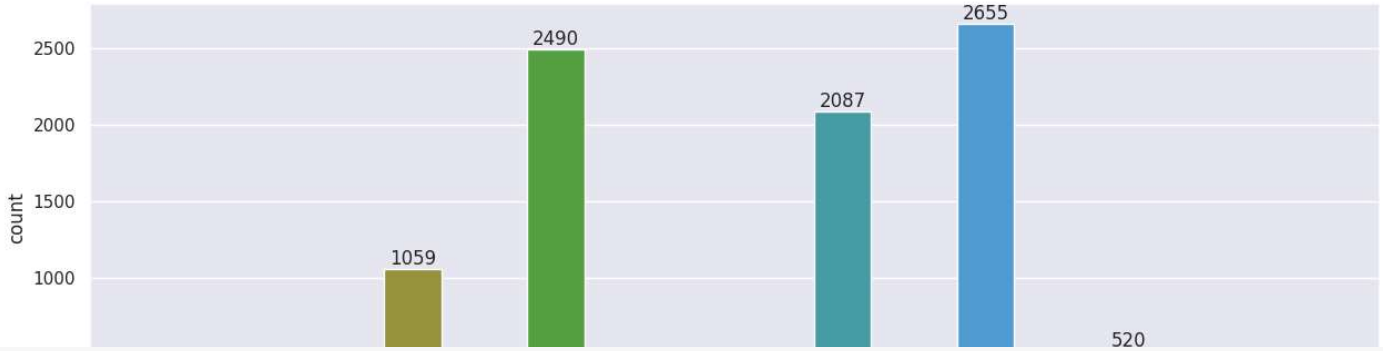
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



From above graph we can see that most of the buyers are wokring in IT, Healthcare and Aviation sector.

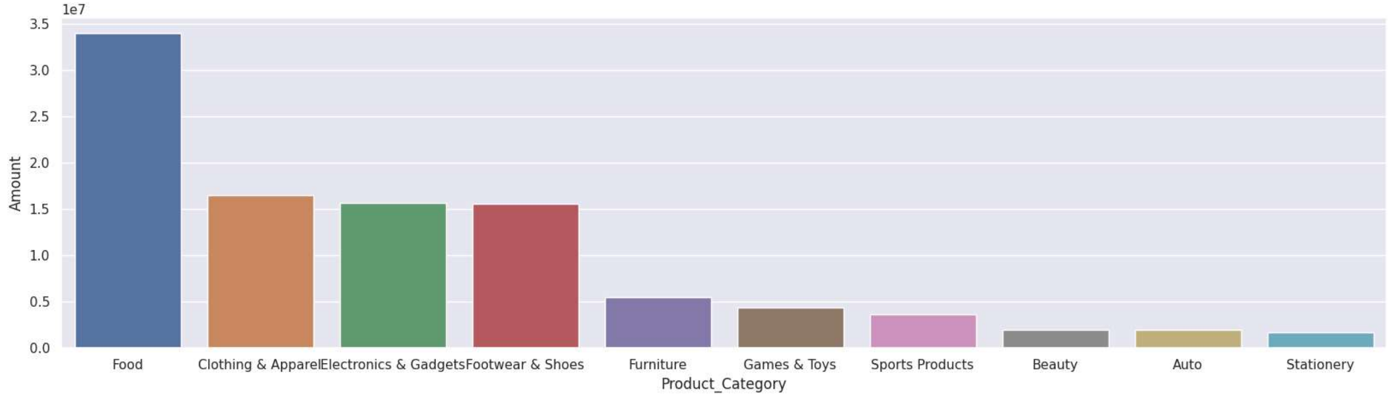
6. Product category

```
ax=sns.countplot(x='Product_Category',data=df,hue='Product_Category')

sns.set(rc={'figure.figsize':(30,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



```
sales_pcategory = 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=sales_pcategory,hue='Product_Category')
```

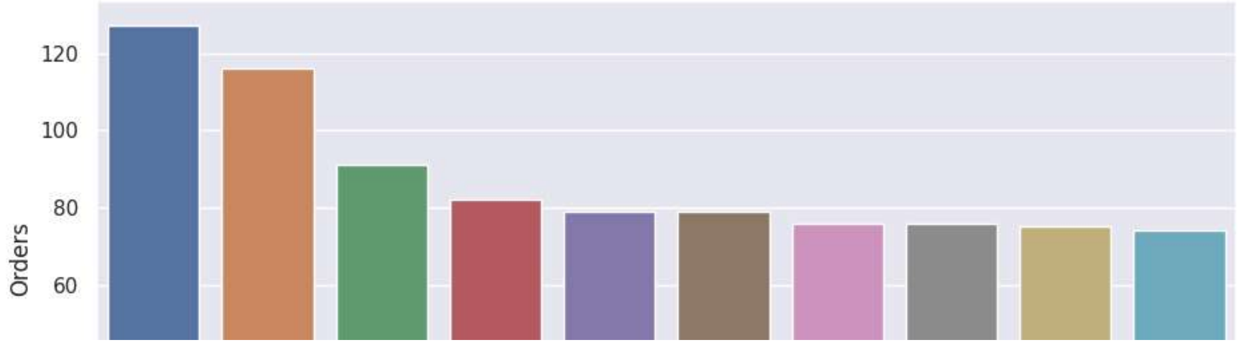
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



From above graph we can see that most selling products are Food, Clothing and Electronics category.

```
sales_pid = df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)

sns.set(rc={'figure.figsize':(11,5)})
sns.barplot(x ='Product_ID',y='Orders',data=sales_pid,hue='Product_ID')
```

```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```


Orders