Diwali_Sales_Analysis

April 25, 2024

[1]:
```
# import python libraries

import numpy as np #arrays mathematical
import pandas as pd # dataframe table

# visualizing data
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

[2]:
```
# import csv file
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

[3]:
```
df.head(2)
```

[3]:

	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

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

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

[5]:
```
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
```

0.0.1 DATA CLEANING

[6]:
```
#drop unrelated/blank columns
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

[7]:
```
#check for null values
pd.isnull(df).sum()
```

[7]:
```
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
```

[8]:
```
# drop null values
df.dropna(inplace=True)
```

[9]:
```
df.info() #checking again to see if dropping NA worked
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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
```

[10]:
```
# use describe() for specific columns
df[['Age', 'Orders', 'Amount']].describe()
```

[10]:

	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610858
std	12.753866	1.114967	5222.355869
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

[11]:
```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

[12]:
```
df['Amount'].dtypes
```

[12]:
```
dtype('int64')
```

0.1 Exploratory Data Analysis

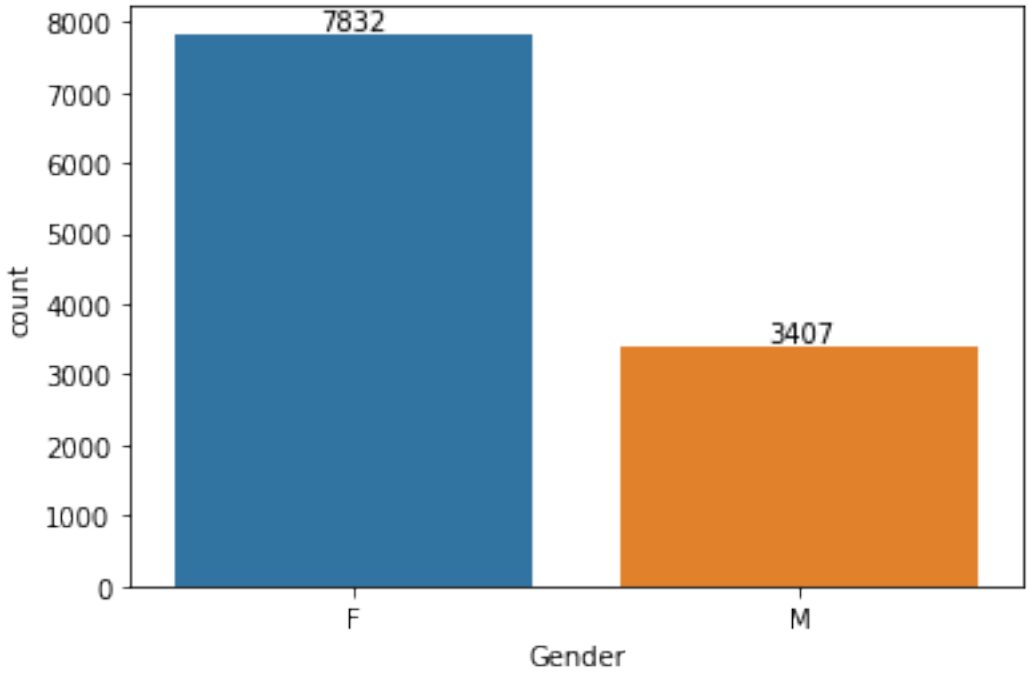
0.1.1 Gender

[13]:
```
df.columns
```

[13]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
```

```
      dtype='object')
```

[14]:
```
# Gender Bar Chart

gen_count = sns.countplot(x = 'Gender',data = df)

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



0.1.2 Therefore, Females have placed more orders than Men

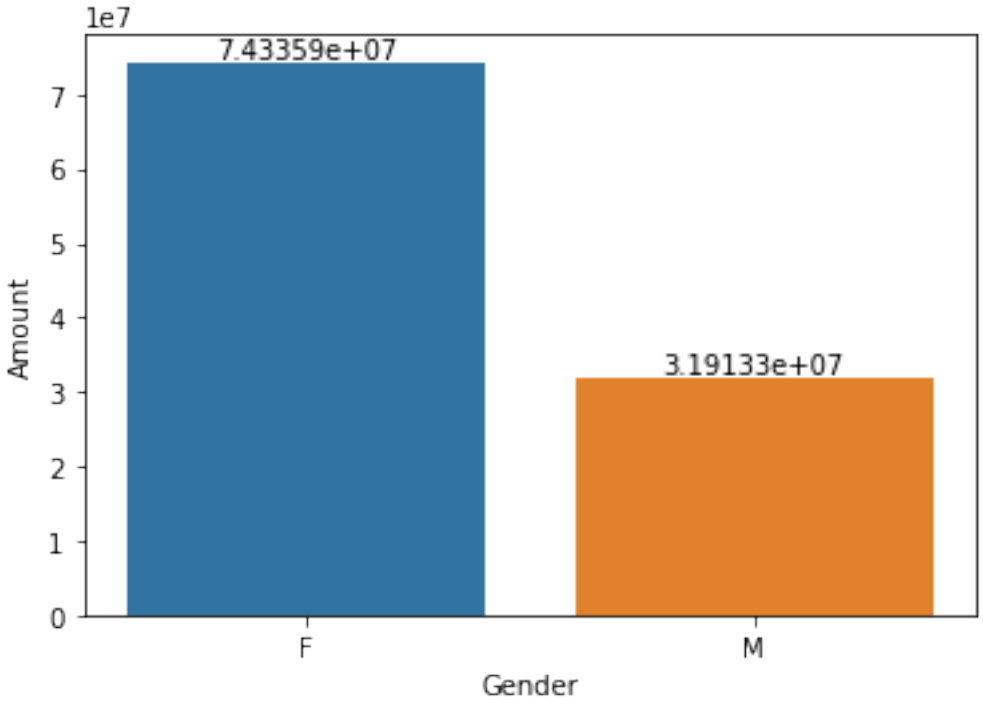
[15]:
```
# Bar chart : Gender vs total Amount

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

[15]:

	Gender	Amount
0	F	74335853
1	M	31913276

[16]:
```
gp = sns.barplot(x = 'Gender',y= 'Amount' ,data = gen_sales)

gp.bar_label(gp.containers[0])  # only 1 container needed unless using `hue`
```

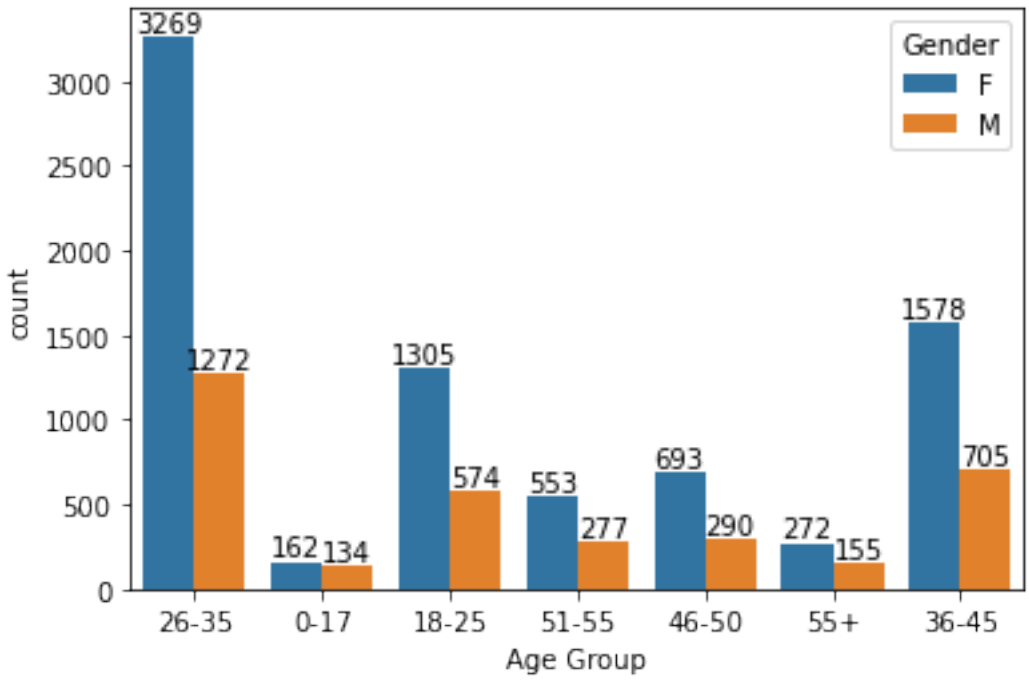
[16]: [Text(0, 0, '7.43359e+07'), Text(0, 0, '3.19133e+07')]



0.1.3 Therefore, Females buy more than Men and has greater purchasing power.

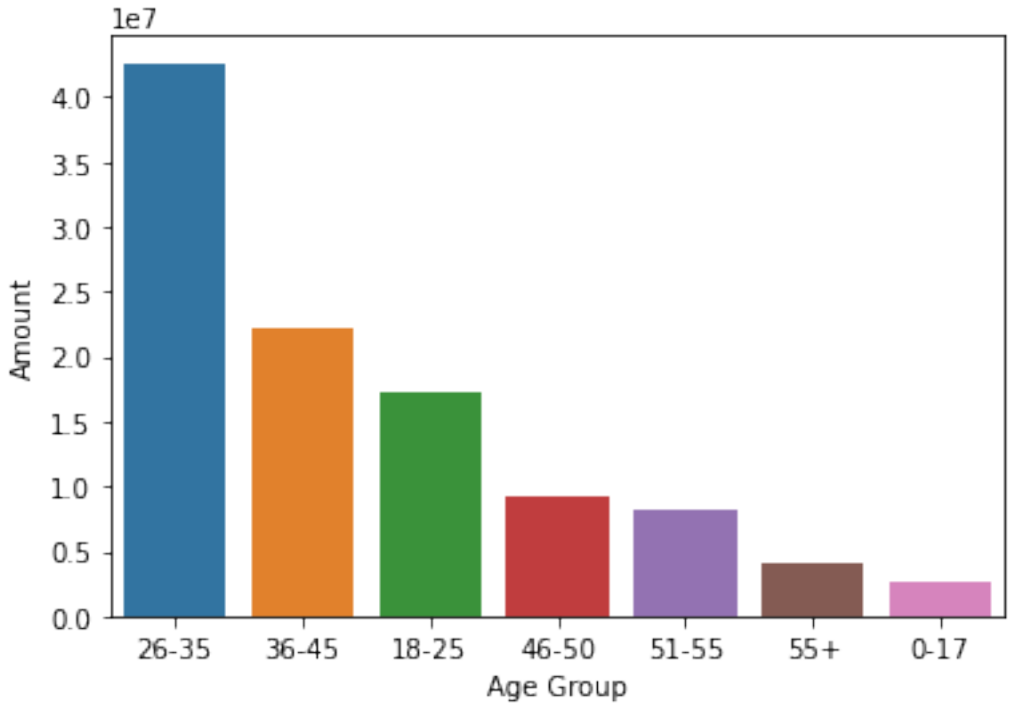
0.1.4 Age

[17]:
```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

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



[18]:
```
# 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)
```

[18]: <AxesSubplot: xlabel='Age Group', ylabel='Amount'>

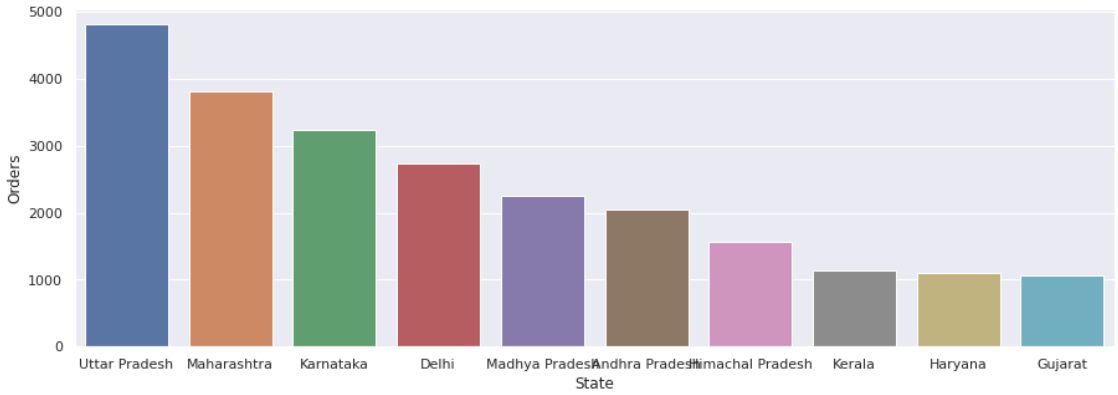


0.1.5 Therefore, most of the buyers are of age group between 26-35 yrs women buyers

0.1.6 State

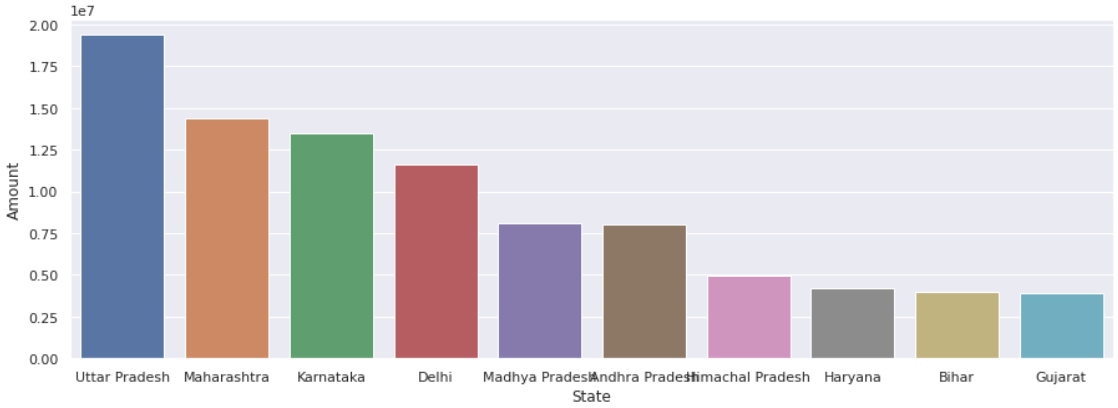
[19]:
```
# total number of orders from top 10 states

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(data = sales_state, x = 'State',y= 'Orders')
```

[19]:
```
<AxesSubplot: xlabel='State', ylabel='Orders'>
```



[20]:
```
# total 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(data = sales_state, x = 'State',y= 'Amount')
```

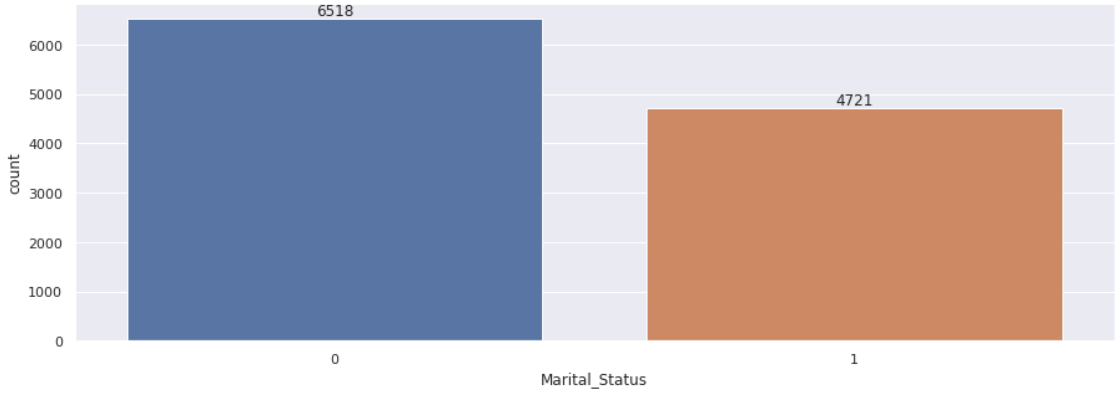
[20]: <AxesSubplot: xlabel='State', ylabel='Amount'>



From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively. Also, Haryana has more purchasing power than Kerala on comparing above 2 graphs.

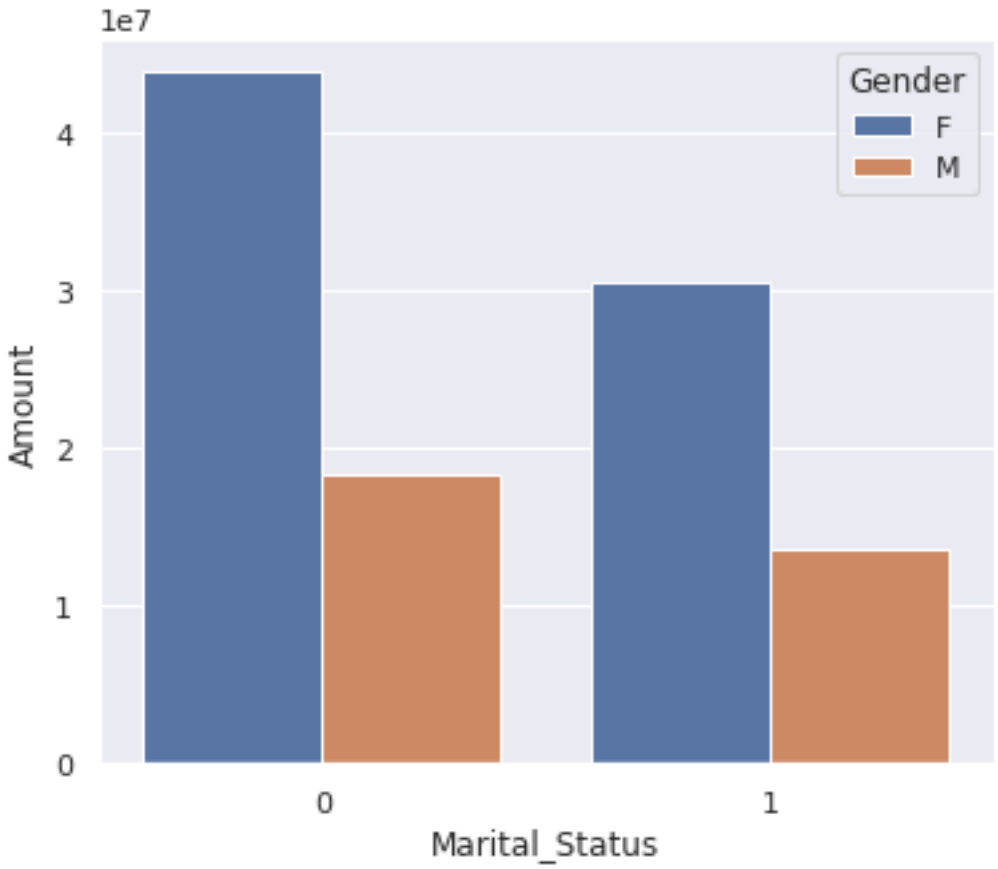
0.1.7 Marital Status

[21]:
```
ax = sns.countplot(data = df, x = 'Marital_Status')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



[22]:
```
sales_state = df.groupby(['Marital_Status', 'Gender'],␣
 ↪as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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

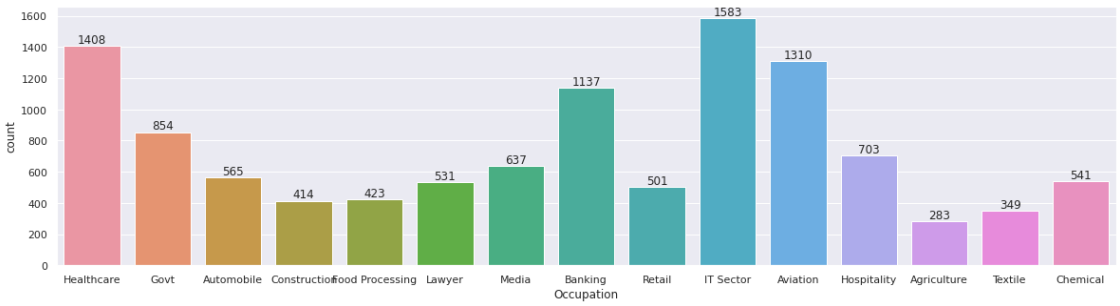
[22]:
```
<AxesSubplot: xlabel='Marital_Status', ylabel='Amount'>
```



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

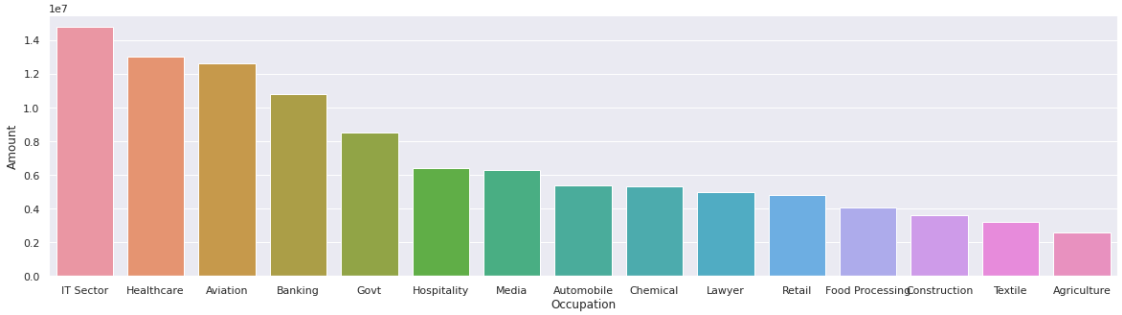
0.1.8 Occupation

[23]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

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



[24]:
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().
↪sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
```

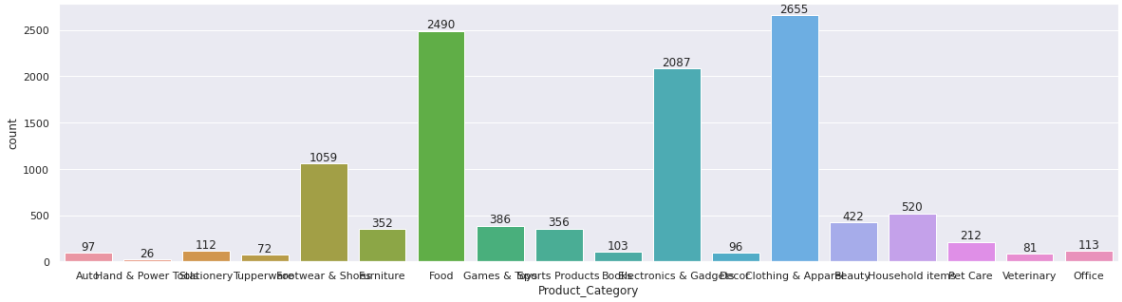
[24]:
```
<AxesSubplot: xlabel='Occupation', ylabel='Amount'>
```



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

0.1.9 Product Category

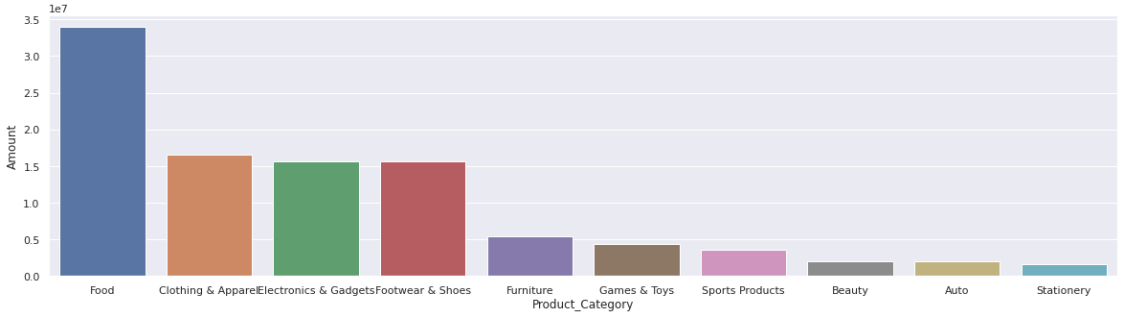
[25]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

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



[26]:
```
sales_state = 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(data = sales_state, x = 'Product_Category',y= 'Amount')
```

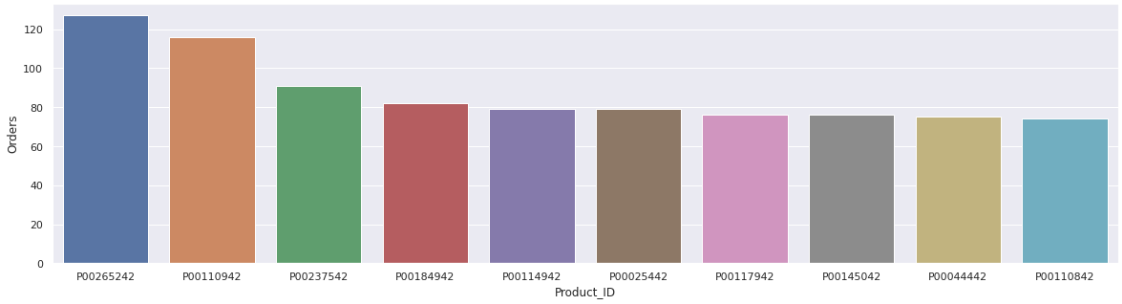
[26]: <AxesSubplot: xlabel='Product_Category', ylabel='Amount'>



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

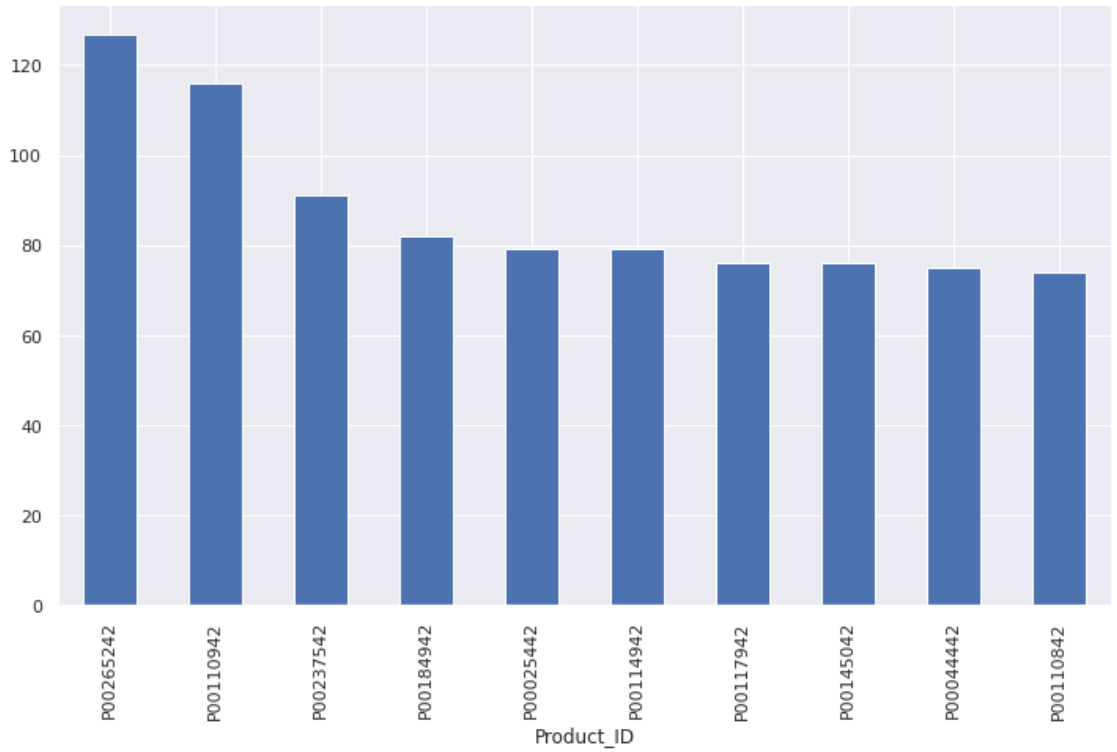
[27]:
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().
 ↪sort_values(by='Orders', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_ID',y= 'Orders')
```

[27]: <AxesSubplot: xlabel='Product_ID', ylabel='Orders'>



[28]:
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).
 ↪sort_values(ascending=False).plot(kind='bar')
```

[28]: <AxesSubplot: xlabel='Product_ID'>



0.2 Conclusion:

0.2.1

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

1 Project Learnings

1.0.1 1. Performed data cleaning and manipulation.

1.0.2 2. Performed exploratory data analysis(EDA) using Pandas, matplotlib and seaborn libraries.

1.0.3 3. Improved customer experience by identifying potential customers across different states, occupations, Gender and age groups.

1.0.4 4. Improved sales by identifying most selling product categories and products, which can help to plan inventory and hence meet the demands.

Thankyou!

[ ]: