Exploratory Data Analysis

December 20, 2023

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

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

[3]:
```
df
```

[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
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3			
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2			
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2			
…	…	…	…	…	…	…	…	…	…	…	…	…			
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4			
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3			
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4			
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3			
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3			

```
2      23924.0     NaN       NaN
3      23912.0     NaN       NaN
4      23877.0     NaN       NaN
…          …     …         …
11246    370.0     NaN       NaN
11247    367.0     NaN       NaN
11248    213.0     NaN       NaN
11249    206.0     NaN       NaN
11250    188.0     NaN       NaN

[11251 rows x 15 columns]
```

[7]:
```
df.shape
```

[7]:
```
(11251, 15)
```

[8]:
```
df.head(10)
```

[8]:
```
   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
5  1000588       Joni  P00057942      M     26-35   28               1
6  1001132       Balk  P00018042      F     18-25   25               1
7  1002092   Shivangi  P00273442      F       55+   61               0
8  1003224     Kushal  P00205642      M     26-35   35               0
9  1003650      Ginny  P00031142      F     26-35   26               1

              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
5  Himachal Pradesh  Northern  Food Processing             Auto       1
6     Uttar Pradesh   Central           Lawyer             Auto       4
7       Maharashtra   Western        IT Sector             Auto       1
8     Uttar Pradesh   Central            Govt              Auto       2
9    Andhra Pradesh  Southern           Media              Auto       4

     Amount  Status  unnamed1
0  23952.00     NaN       NaN
1  23934.00     NaN       NaN
2  23924.00     NaN       NaN
3  23912.00     NaN       NaN
```

```
 4  23877.00     NaN       NaN
 5  23877.00     NaN       NaN
 6  23841.00     NaN       NaN
 7       NaN     NaN       NaN
 8  23809.00     NaN       NaN
 9  23799.99     NaN       NaN
```

[10]:
```
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
```

[12]:
```
df.drop(['Status', 'unnamed1'],axis=1,inplace=True)
```

[13]:
```
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
```

[15]:
```
df.isnull().sum()
```

[15]:
```
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
```

[16]:
```
df.dropna(inplace=True)
```

[17]:
```
df.info()
```

```
<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
```

[18]:
```
df['Amount']=df['Amount'].astype('int')
```

[19]:
```
df.info()
```

```
<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  int32 
dtypes: int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

[21]:
```
df.columns
```

[21]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[27]:
```
df[['Age','Amount','Orders']].describe()
```

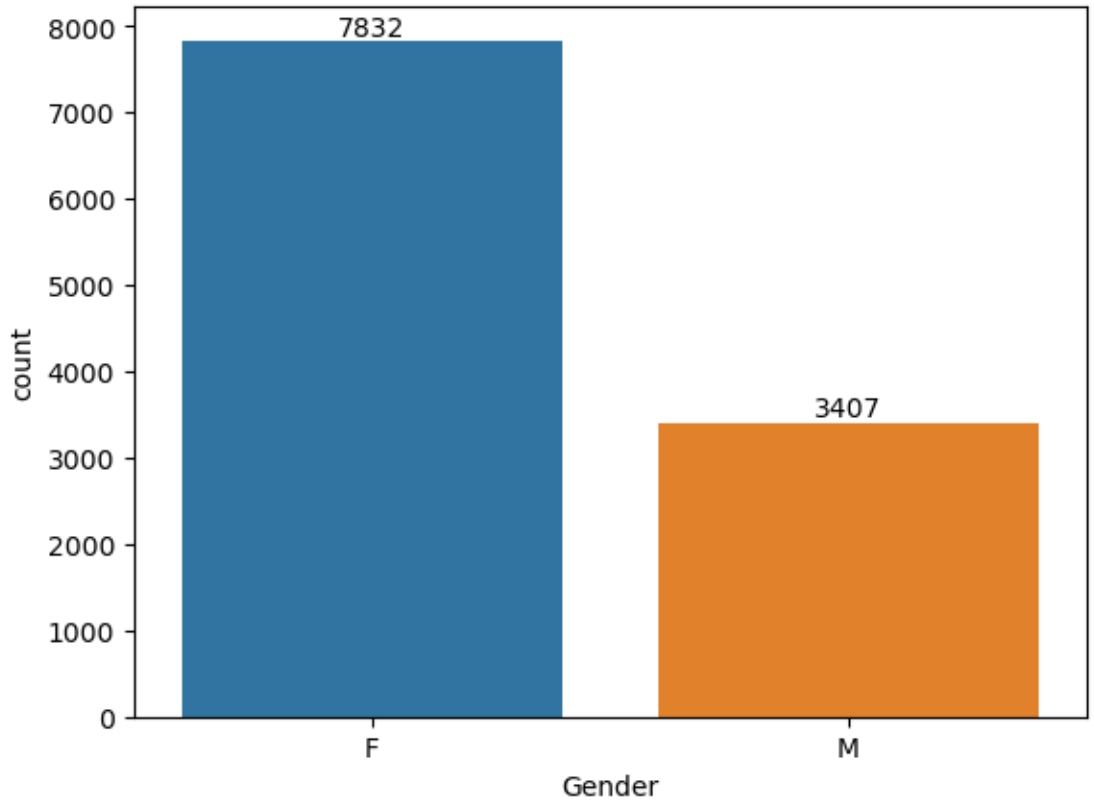
[27]:

	Age	Amount	Orders
count	11239.000000	11239.000000	11239.000000
mean	35.410357	9453.610553	2.489634
std	12.753866	5222.355168	1.114967
min	12.000000	188.000000	1.000000
25%	27.000000	5443.000000	2.000000
50%	33.000000	8109.000000	2.000000
75%	43.000000	12675.000000	3.000000
max	92.000000	23952.000000	4.000000

1 Exploratory Data Analysis

2 Gender

[29]:
```
# plotting a bar: count by gender
ax = sns.countplot(data=df, x='Gender')

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



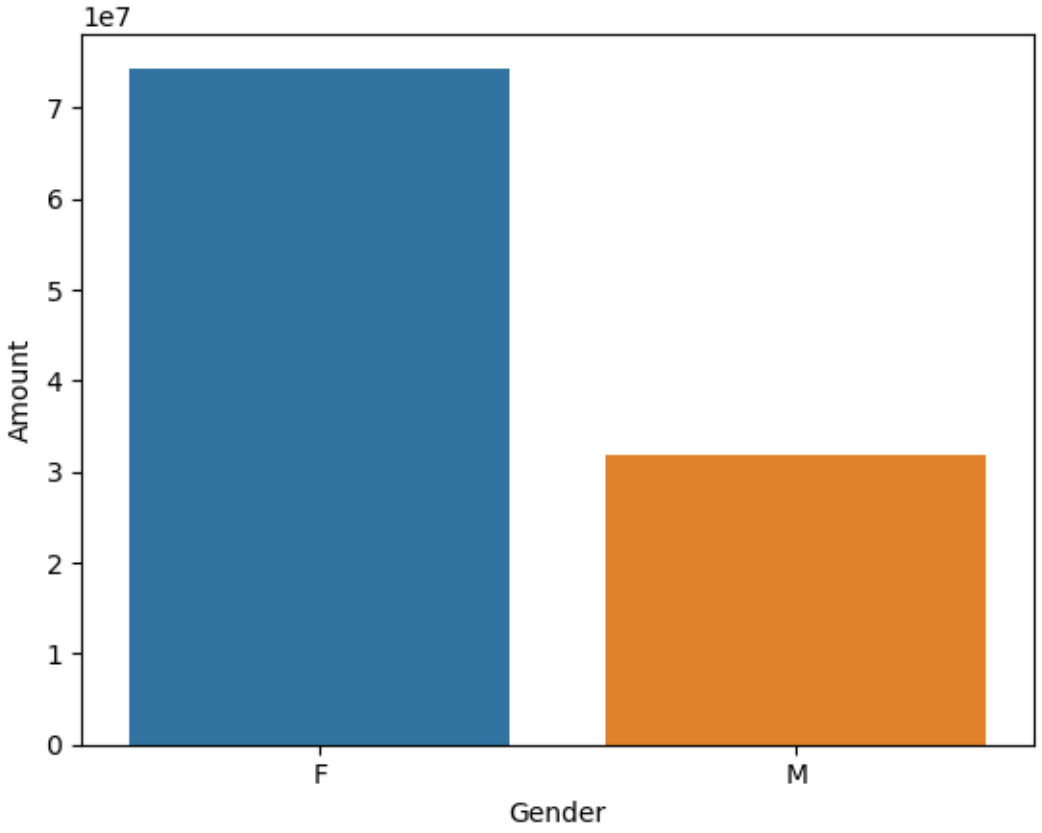
[32]:
```
# plotting a bar: purchasing power by gender

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

[32]:

	Gender	Amount
0	F	74335853
1	M	31913276

[33]:
```
sns.barplot(x='Gender', y='Amount', data=P_gender)
```

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

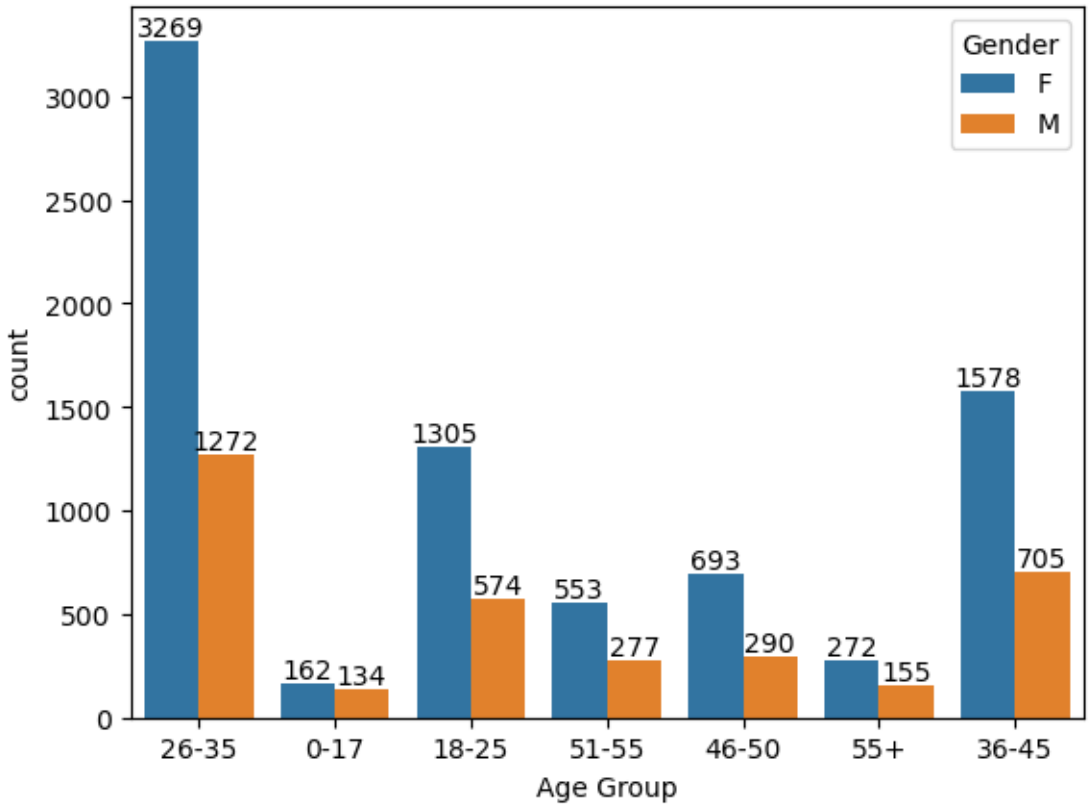


3 Age

[34]:
```
df.columns
```

[34]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[39]:
```
ax_age = sns.countplot(data=df, x= 'Age Group', hue='Gender')

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



[40]:
```
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().
↪sort_values(by='Amount', ascending=False)
```

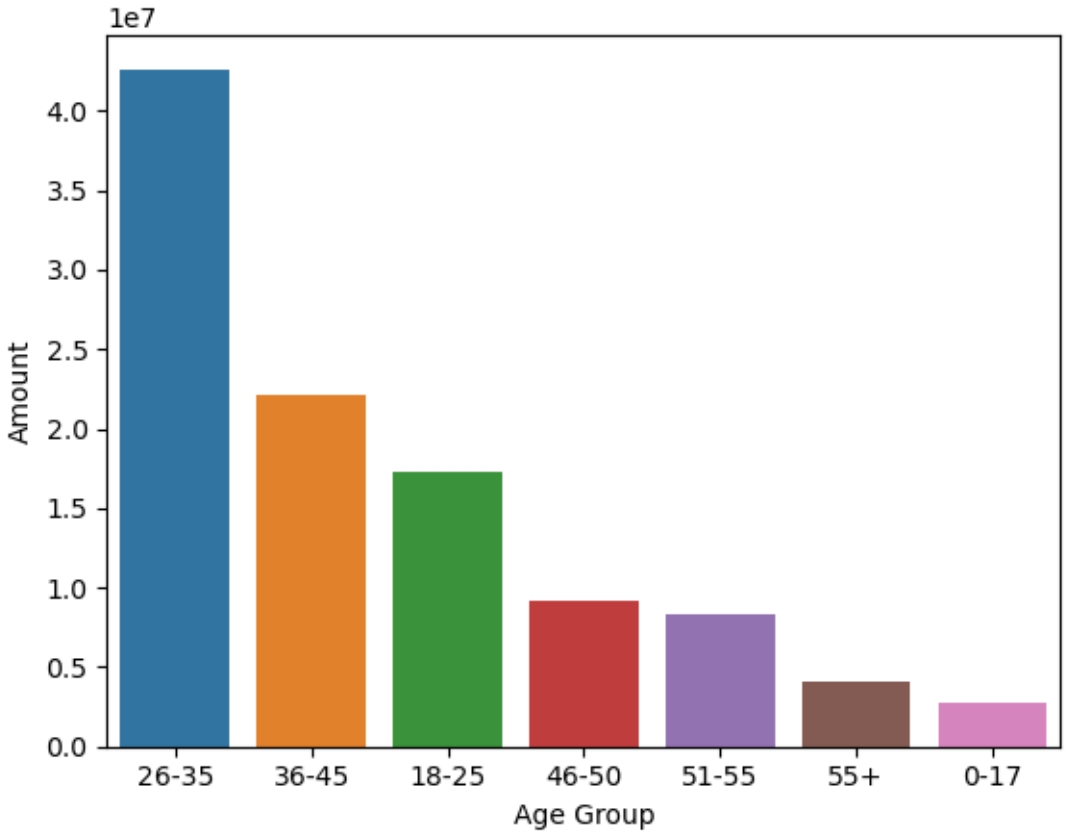
[41]:
```
sales_age
```

[41]:

	Age Group	Amount
2	26-35	42613442
3	36-45	22144994
1	18-25	17240732
4	46-50	9207844
5	51-55	8261477
6	55+	4080987
0	0-17	2699653

[45]:
```
sns.barplot(x='Age Group', y='Amount', data= sales_age)
```

[45]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

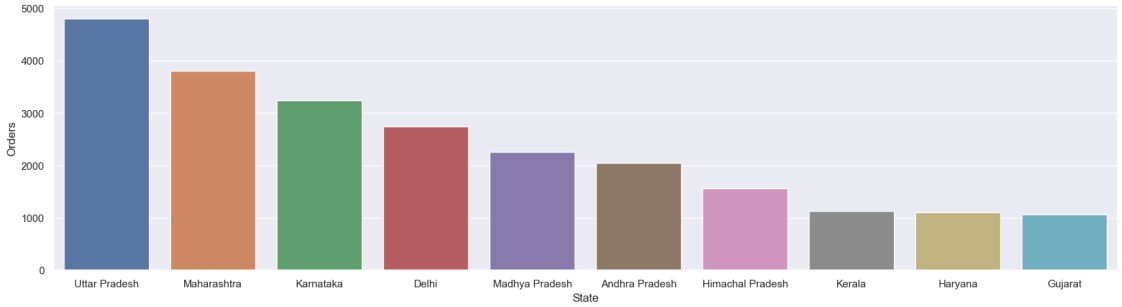


From the above graph, we find that most of the buyers are in the age group of 26-35.

4 States

[60]:
```
sales_state = df.groupby(['State'],as_index=False)['Orders'].sum().
  ↪sort_values(by='Orders', ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(x='State',y='Orders',data=sales_state)

for bars in sales_state.cont
```

[60]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



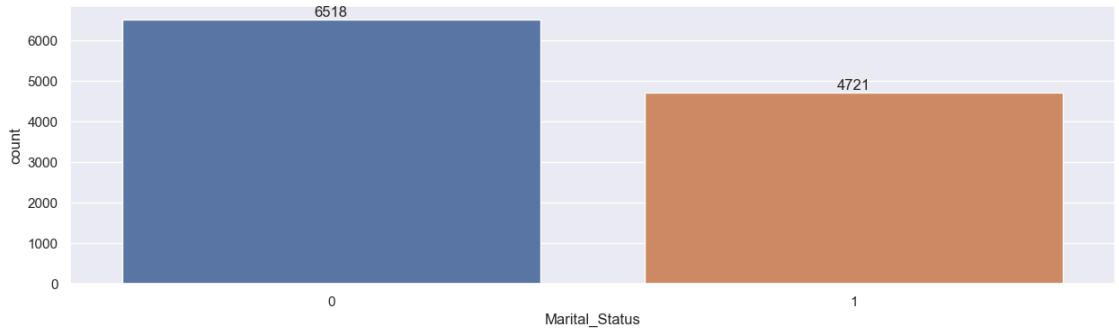
From the above graph, we find that the highest amount of sales comes from Uttar Pradesh, Maharashtra, and Karnataka.

5 Marital Status

[62]:
```
df.columns
```

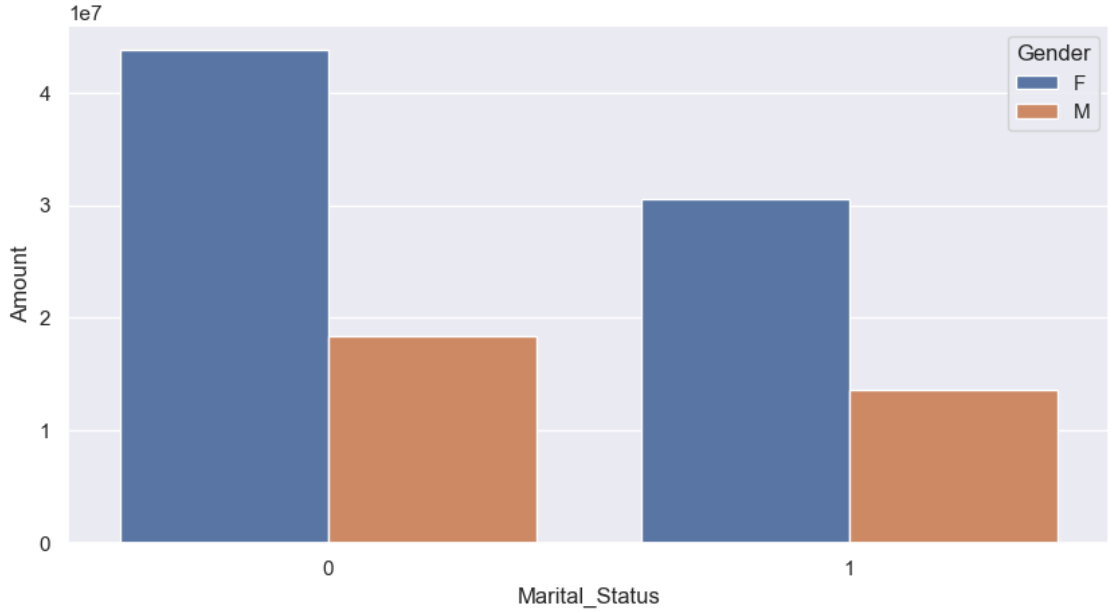
[62]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[74]:
```
ax_m = sns.countplot(x='Marital_Status', data=df)
sns.set(rc={'figure.figsize':(15,5)})

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



[81]:
```
sales_m_status = df.groupby(['Marital_Status', 'Gender'],␣
 ↪as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False )
sns.set(rc={'figure.figsize':(10,5)})
sns.barplot(x= 'Marital_Status', y='Amount', hue='Gender', data=sales_m_status)
```

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

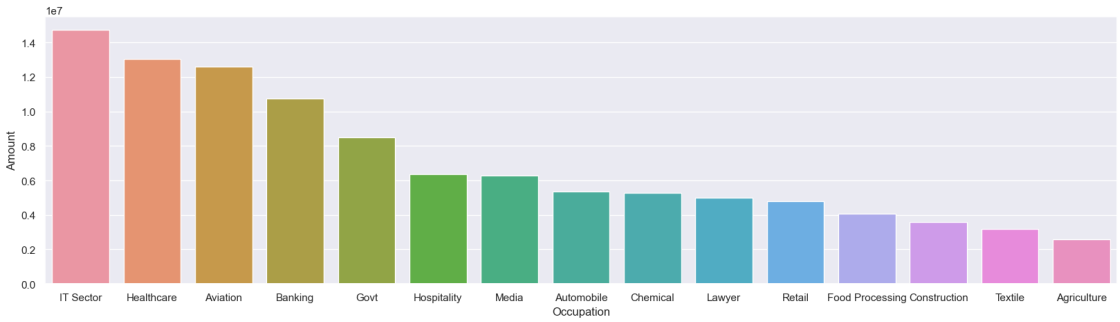


From the above graph, we find that married women have greater purchasing power.

6 Occupation

[88]:
```
O_Amt = df.groupby(['Occupation'], as_index=False)['Amount'].sum().
↪sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=O_Amt,x='Occupation',y='Amount')
```

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

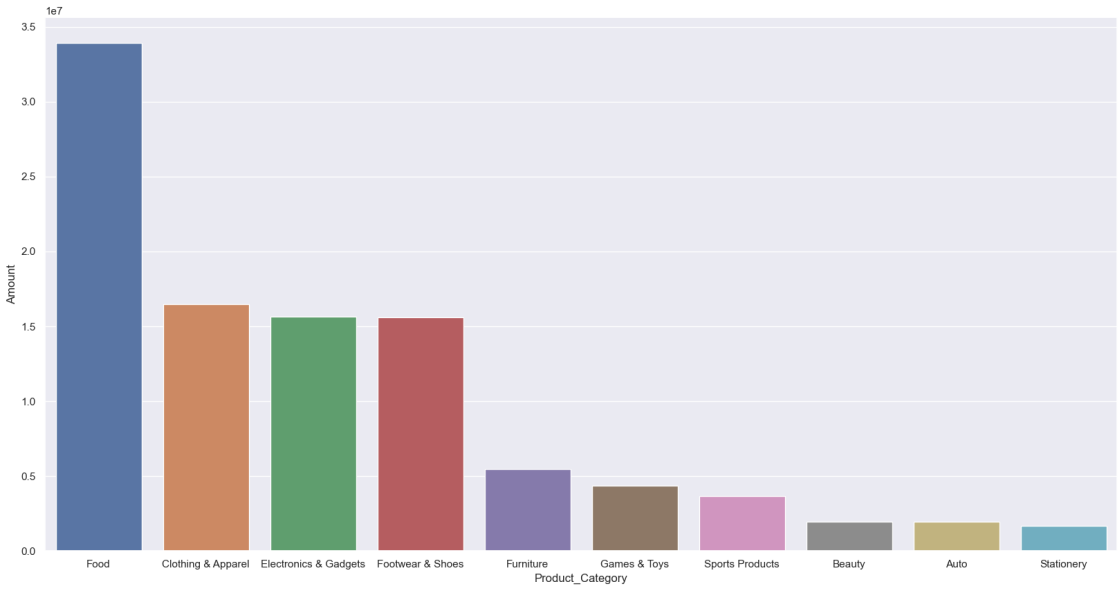


From the above graph, we find that people from the IT sector are spending much more than those

from other sectors

7 Product Category

[109]:
```
p_amt = df.groupby(['Product_Category'],as_index=False)['Amount'].sum().
  ↪sort_values(by='Amount',ascending=False).head(10)
sns.barplot(data=p_amt, x='Product_Category',y='Amount')
sns.set(rc={'figure.figsize':(20,6)})
```



From the above graph, we find that people are spending much more on food.

8 Conclusion

9 From the above analysis, we find that married women from Uttar Pradesh, Maharashtra, and Karnataka, currently working in the IT sector, are spending much more money on food items, clothing & apparel, and electronics & gadgets.

[ ]: