festival-data

April 23, 2024

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[4]:
```
df = pd.read_csv("Festival Data.csv" , encoding= 'unicode_escape')
#encoding= 'unicode_escape for avaoiding error while loading the file
```

[5]:
```
df
```

[5]:

	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	Status	unnamed1

```
0      23952.0     NaN       NaN
1      23934.0     NaN       NaN
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]
```

0.1 Data Cleaning

[12]:
```
df.shape
```

[12]:
```
(11251, 15)
```

[14]:
```
df.head(7)
```

[14]:
```
   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

              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

    Amount  Status  unnamed1
0  23952.0     NaN       NaN
1  23934.0     NaN       NaN
2  23924.0     NaN       NaN
3  23912.0     NaN       NaN
4  23877.0     NaN       NaN
5  23877.0     NaN       NaN
```

```
 6  23841.0      NaN        NaN
```

[15]:
```
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
```

[ ]:
```
df.drop(["Status","unnamed1"],axis=1,inplace=True)
# dropping two coloumn withh all the rows (axis) and permenat deletion with␣
 ↪inplace
```

[20]:
```
df
```

[20]:

	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]
```

[29]:
```
pd.isnull(df).sum()
#counting summ of null value
```

[29]:
```
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              0
dtype: int64
```

[32]:
```
df.dropna(inplace=True)
#droping all the null value
```

[33]:
```
pd.isnull(df).sum()
```

[33]:
```
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              0
dtype: int64
```

[38]:
```
df["Amount"]=  df["Amount"] .astype('int')
# changing the data type of amount column from float to int removing all the␣
 ↪decimal values
```

[39]:
```
df["Amount"].dtypes
# checking the data type of amount column
```

[39]:
```
dtype('int32')
```

[40]:
```
df
```

[40]:

	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
1       23934
2       23924
3       23912
4       23877
…         …
11246     370
11247     367
11248     213
11249     206
11250     188

[11239 rows x 13 columns]
```

[42]:
```
df.describe()
```

[42]:
```
             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
```

[44]:
```
df[['Age','Orders','Amount']].describe()
```

[44]:
```
                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
```

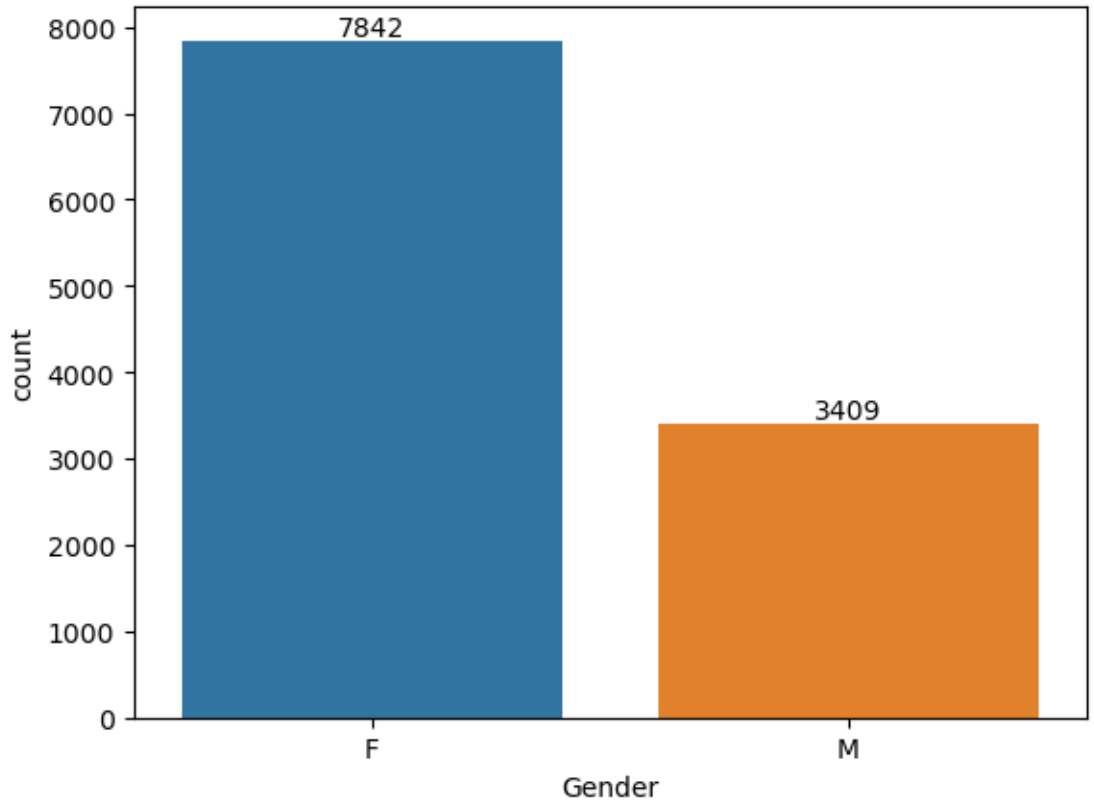
1 Exploratory Data Analysis

[47]:
```
df.columns
```

[47]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

1.1 Count of people for each gender

[7]:
```
ax = sns.countplot ( x= 'Gender', data=df)
for bars in ax.containers : ax.bar_label(bars)
```



1.2 Amount spend by each gender

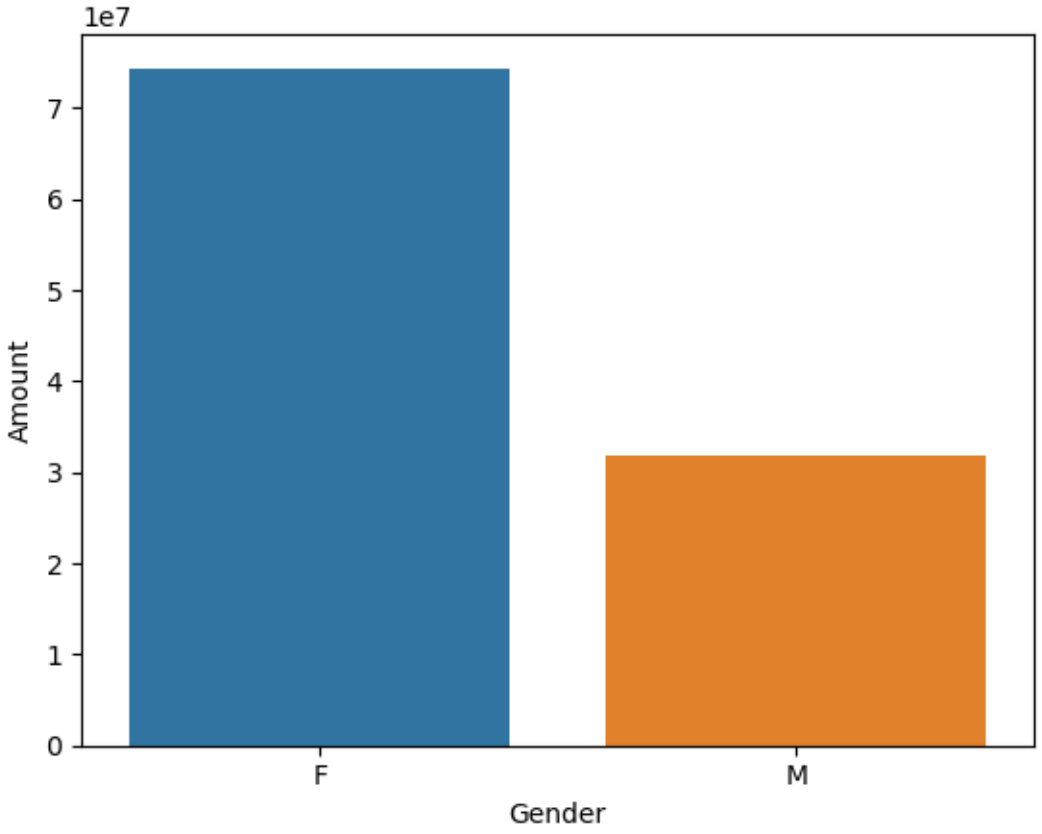
[10]:
```
df.groupby(['Gender'],as_index=False)['Amount'].sum().
↪sort_values(by='Amount',ascending=False)
```

[10]:

	Gender	Amount
0	F	74335856.43
1	M	31913276.00

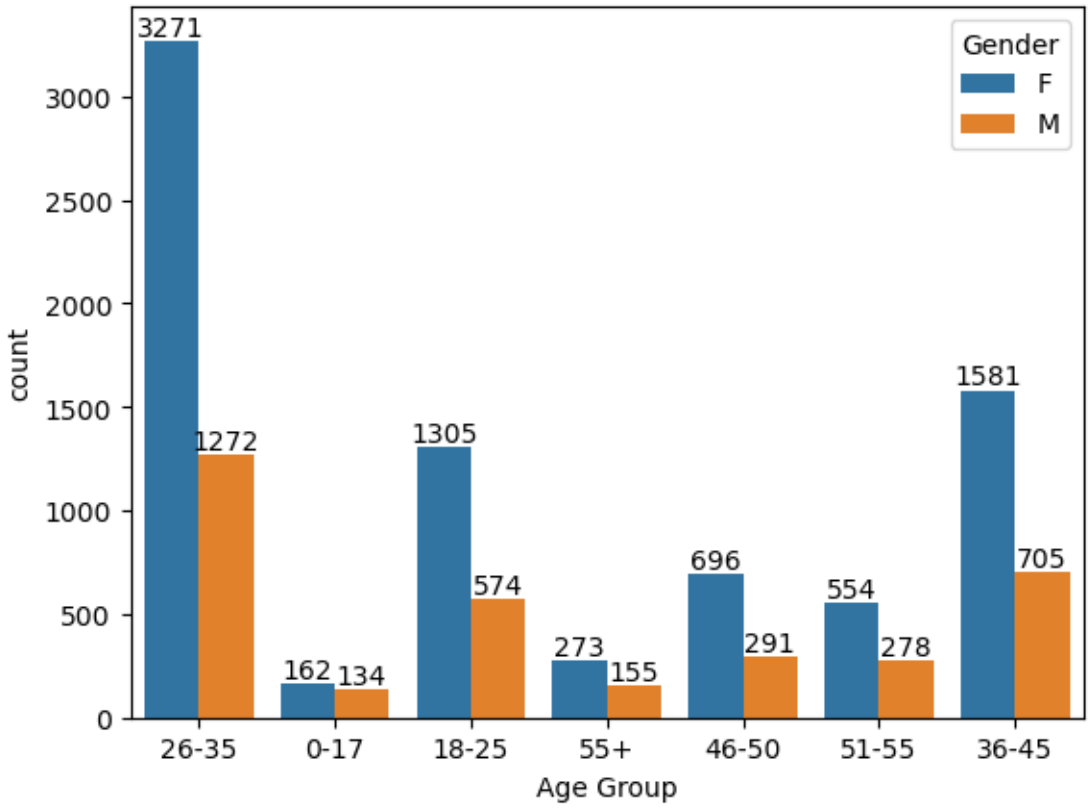
[11]:
```
sales_gender= df.groupby(['Gender'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False)
sns.barplot ( x= 'Gender', y= 'Amount',data=sales_gender)
```

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



1.3 Count of each age group people on the basis of Gender

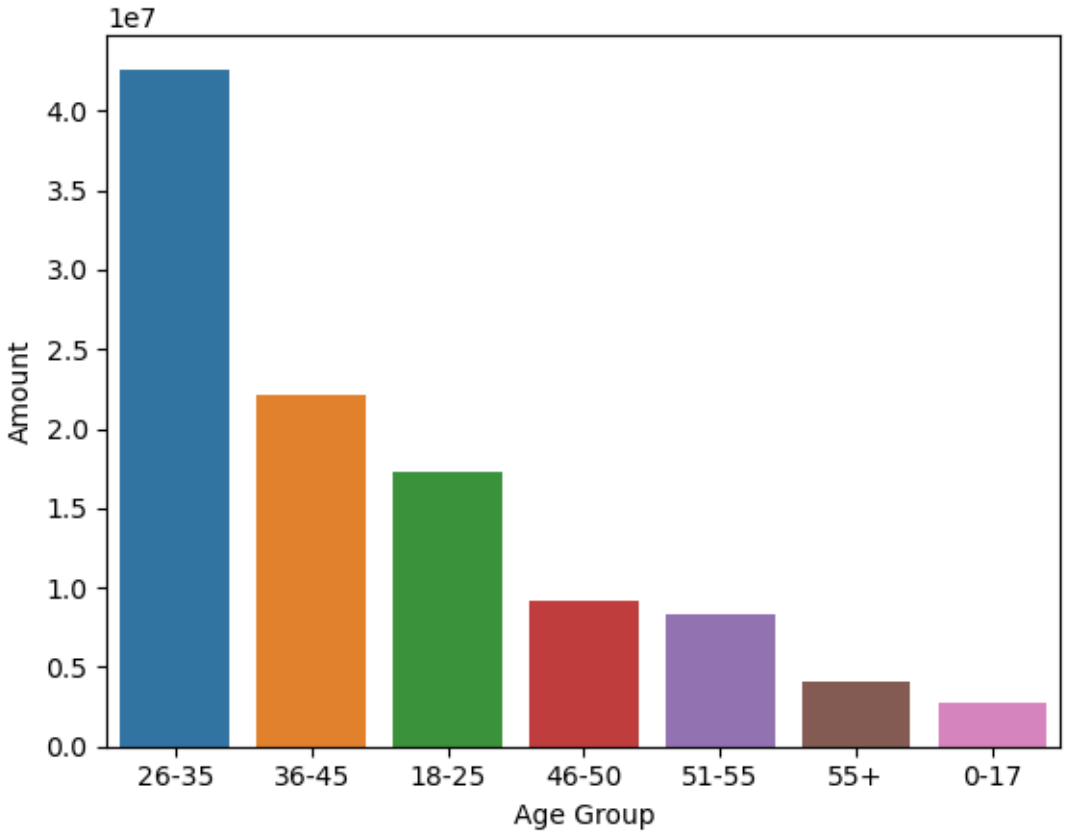
[12]:
```
ax = sns.countplot (  data=df, x= 'Age Group', hue= 'Gender')
for bars in ax.containers : ax.bar_label(bars)
```



1.4 Most sales on the basis of age group

[13]:
```
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)
```

[13]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



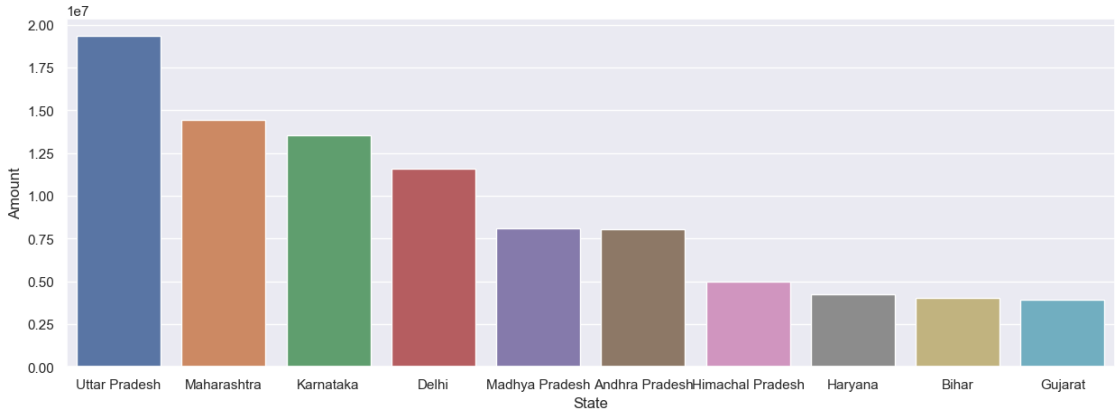
[14]:
```
df.columns
```

[14]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

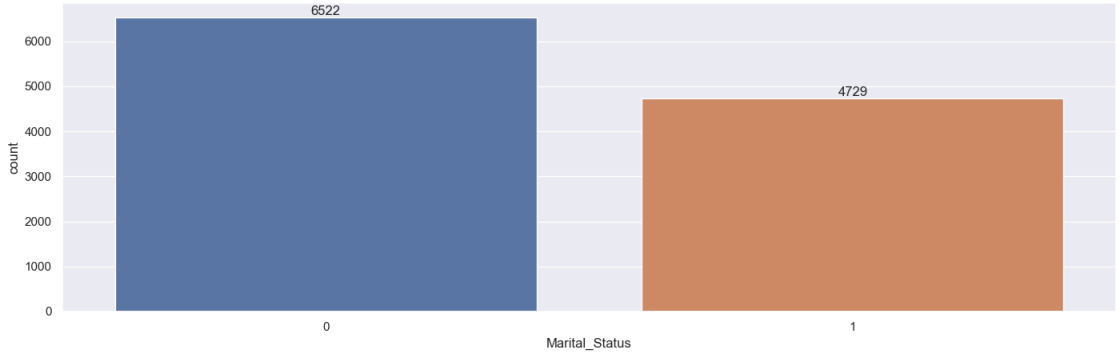
1.5 Total sales by State

[19]:
```
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')
```

[19]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



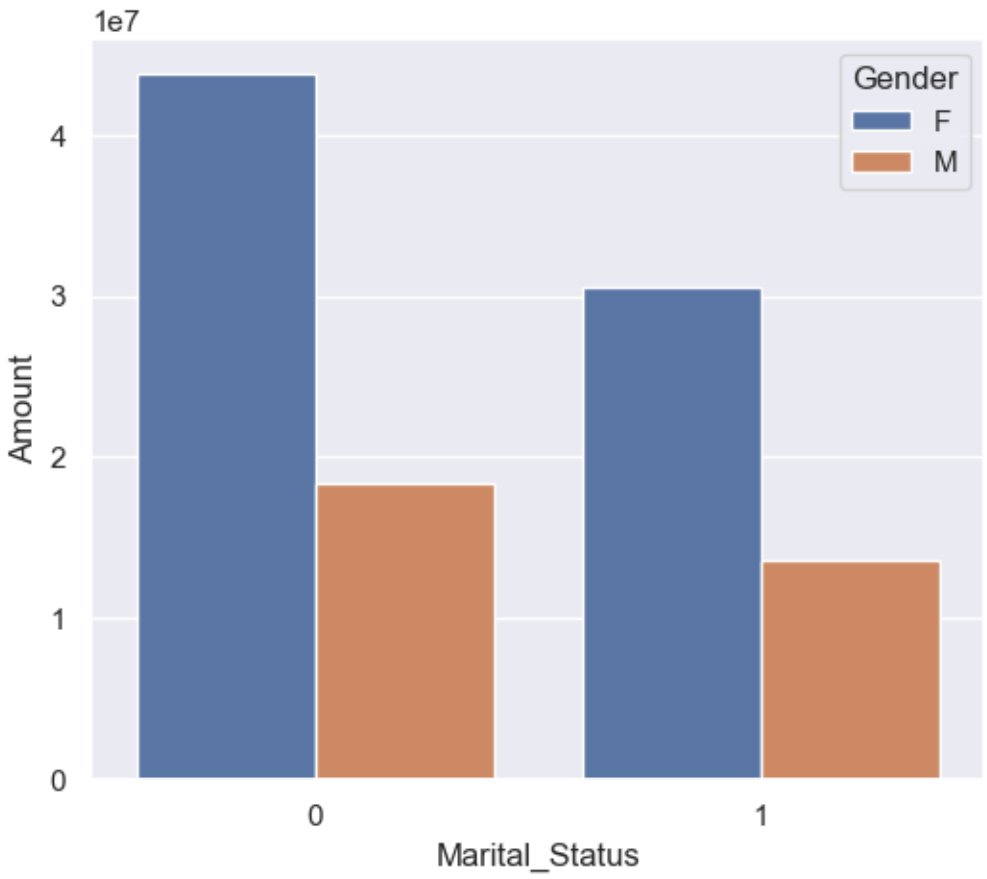
[23]:
```
ax = sns.countplot (  data=df, x= 'Marital_Status', )
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers : ax.bar_label(bars)
```



1.6 Married status of gender on the basis of sale

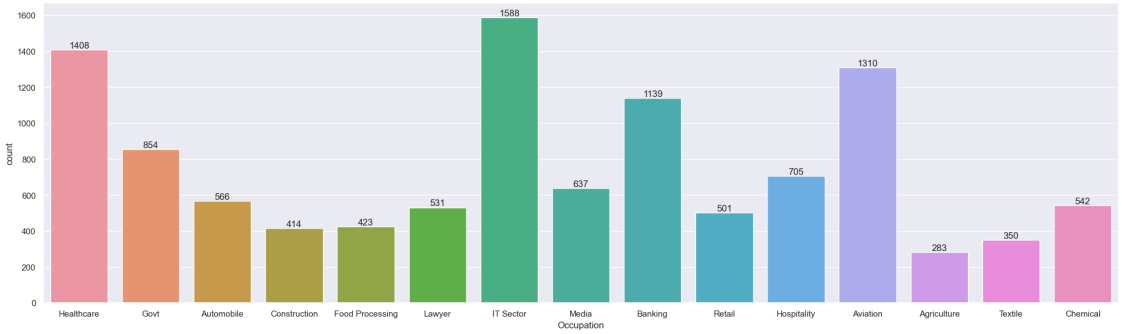
[28]:
```
sales_Marital_Status= df.
 ↪groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot (data=sales_Marital_Status, x= 'Marital_Status', y=␣
 ↪'Amount',hue='Gender')
```

[28]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



2 Count of people on the basis of occupation

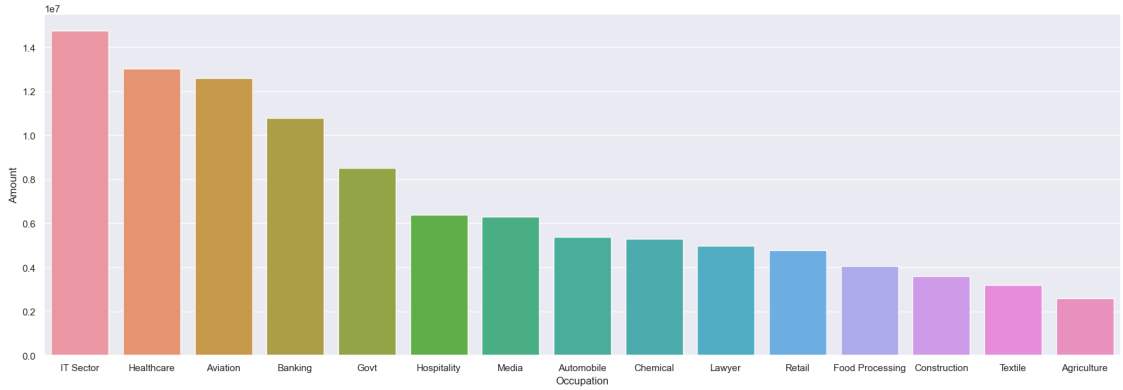
[38]:
```
ax = sns.countplot (  data=df, x= 'Occupation', )
sns.set(rc={'figure.figsize':(25,2)})
for bars in ax.containers : ax.bar_label(bars)
```



2.1 Sales of product on the basis of occupation

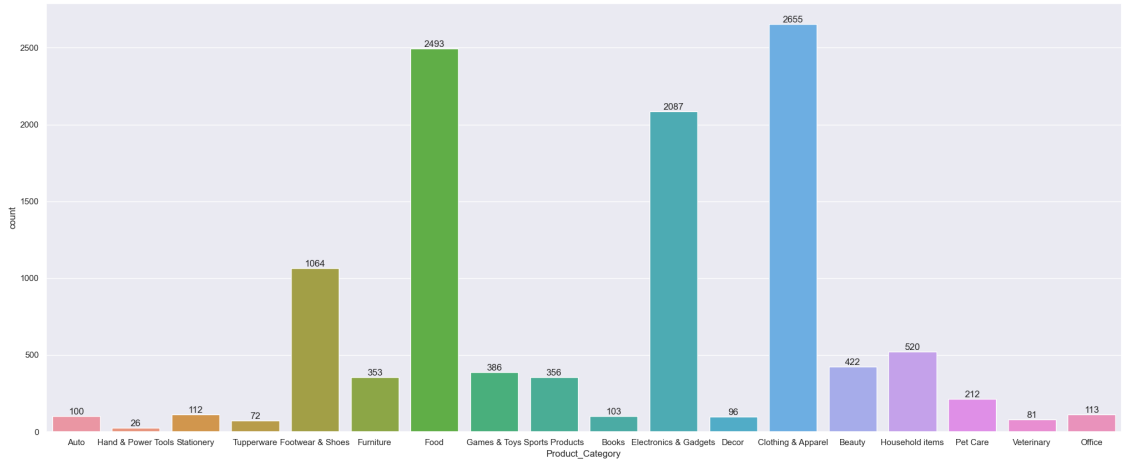
[45]:
```
sales_state= df.groupby(['Occupation'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(22,7)})
sns.barplot (data=sales_state, x= 'Occupation', y= 'Amount')
```

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



2.2 Product category

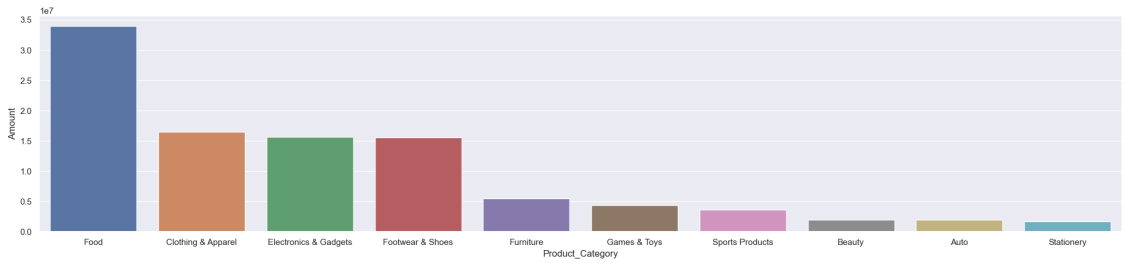
[51]:
```
ax = sns.countplot (  data=df, x= 'Product_Category', )
sns.set(rc={'figure.figsize':(15,10)})
for bars in ax.containers : ax.bar_label(bars)
```



2.3 Top 10 Product_Category sold

[67]:
```
sales_Product_Category= df.
  ↪groupby(['Product_Category'],as_index=False)['Amount'].sum().
  ↪sort_values(by='Amount',ascending=False).head(10)
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot (data=sales_Product_Category, x= 'Product_Category', y= 'Amount')
```

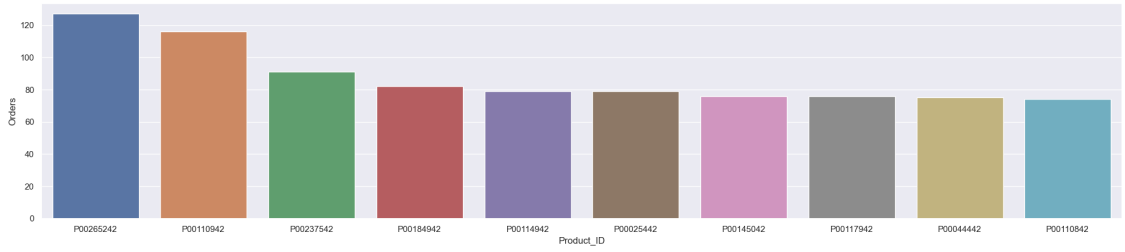
[67]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



2.4 Count of orders of each product_id

[70]:
```
sales_Product_ID= df.groupby(['Product_ID'],as_index=False)['Orders'].sum().
  ↪sort_values(by='Orders',ascending=False).head(10)
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot (data=sales_Product_ID, x= 'Product_ID', y= 'Orders')
```

[70]:
```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```

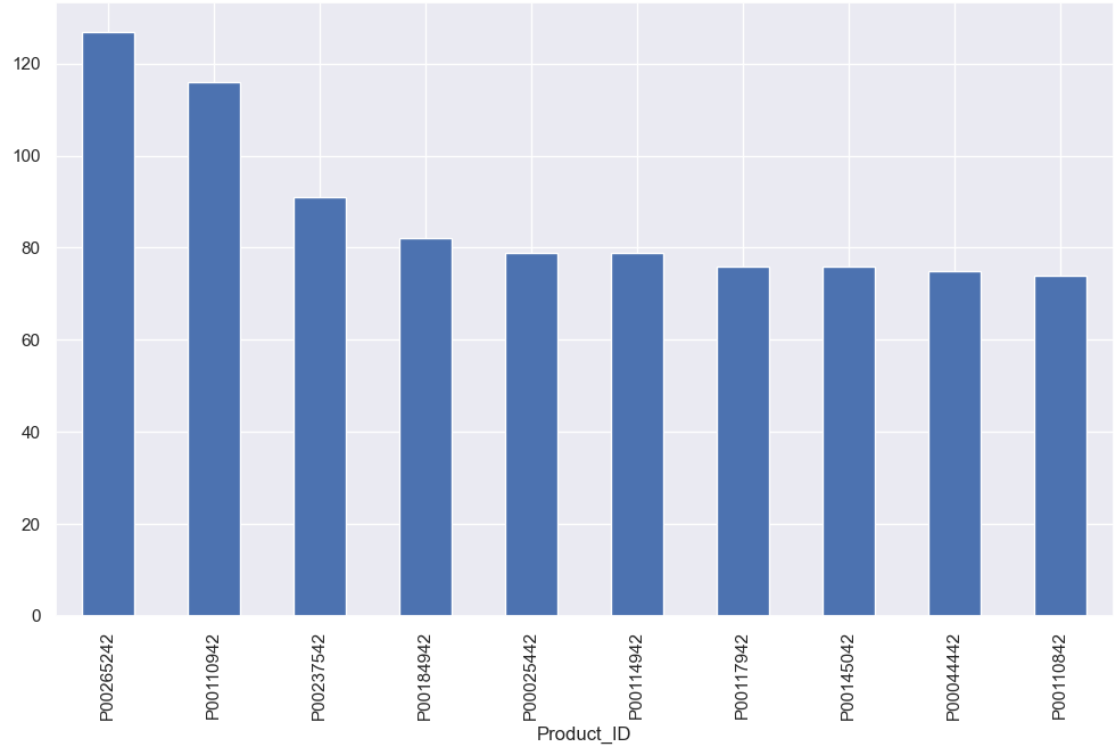


[71]:
```
df.columns
```

[71]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

[74]:
```
fig1,ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).
  ↪sort_values(ascending=False).plot(kind='bar')
```

[74]: <Axes: xlabel='Product_ID'>



3 Conclusion

3.0.1 Married women of age 26-25 yrs from UP, Maharastra, Karnataka working in IT,HEALTHCARE,AVIATION likely to buy product of Food, Clothing, Electronics and Gadgets