DIWALI SALES ANALYSIS

Utilized Power BI to design and develop an interactive and visually appealing dashboard that showcases key sales metrics, trends, and customer insights. The dashboard provides an intuitive user interface with slicers, filters, and drill-down capabilities to explore and analyze sales data from various dimensions. Leveraged Python for data preprocessing, cleaning, and advanced analysis tasks. The raw sales data was processed using libraries such as Pandas and NumPy to ensure data quality, consistency, and accuracy for reliable analysis results. The Python code also implemented statistical analysis techniques to identify patterns, correlations, and outliers within the data. Conducted in-depth analysis of the sales data, including revenue analysis, product performance analysis, customer segmentation, and market trends analysis.

In [8]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [9]:
```
df=pd.read_csv(r"C:\Desktop\Data Analyst Project\DIWALI SALES\Python_Diwali_Sales_Analysis-main\diwalisale.csv",encoding= 'unicode_escape')
```

In [10]:
```
df.shape
```

Out[10]:
```
(11248, 14)
```

In [11]:
```
df.head(10)
```

Out[11]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	unnamed
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	23877.00	NaN
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN
7	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN
8	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23799.99	NaN
9	1003829	Harshita	P00200842	M	26-35	34	0	Delhi	Central	Banking	Auto	1	23770.00	NaN

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11248 entries, 0 to 11247
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11248 non-null  int64  
 1   Cust_name         11248 non-null  object 
 2   Product_ID        11248 non-null  object 
 3   Gender            11248 non-null  object 
 4   Age Group         11248 non-null  object 
 5   Age               11248 non-null  int64  
 6   Marital_Status    11248 non-null  int64  
 7   State             11248 non-null  object 
 8   Zone              11248 non-null  object 
 9   Occupation        11248 non-null  object 
 10  Product_Category  11248 non-null  object 
 11  Orders            11248 non-null  int64  
 12  Amount            11239 non-null  float64
 13  unnamed           0 non-null      float64
dtypes: float64(2), int64(4), object(8)
memory usage: 1.2+ MB
```

In [13]:
```
df.drop(['unnamed'], axis=1, inplace=True)
```

we use drop to delete unrelated data

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11248 entries, 0 to 11247
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11248 non-null  int64  
 1   Cust_name         11248 non-null  object 
 2   Product_ID        11248 non-null  object 
 3   Gender            11248 non-null  object 
 4   Age Group         11248 non-null  object 
 5   Age               11248 non-null  int64  
 6   Marital_Status    11248 non-null  int64  
 7   State             11248 non-null  object 
 8   Zone              11248 non-null  object 
 9   Occupation        11248 non-null  object 
 10  Product_Category  11248 non-null  object 
 11  Orders            11248 non-null  int64  
 12  Amount            11239 non-null  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

In [15]:
```
pd.isnull(df) #check null values
```

Out[15]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11243	False	False	False	False	False	False	False	False	False	False	False	False	False
11244	False	False	False	False	False	False	False	False	False	False	False	False	False
11245	False	False	False	False	False	False	False	False	False	False	False	False	False
11246	False	False	False	False	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False	False	False	False	False

11248 rows × 13 columns

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

In [17]:
```
df.shape
```

Out[17]:
```
(11239, 13)
```

In [18]:
```
df['Amount']=df['Amount'].astype('int') #change data type from current to integer
```

In [19]:
```
df['Amount'].dtype
```

Out[19]:
```
dtype('int32')
```

```
In [20]: df.columns
```

```
Out[20]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
                'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
                'Orders', 'Amount'],
               dtype='object')
```

```
In [21]: df.rename(columns={'Occupation':'Service'})
```

Out[21]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Service	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11243	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11244	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11245	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11246	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11247	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns

In [22]:
```
df.info
```

Out[22]:
```
<bound method DataFrame.info of        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
...        ...          ...        ...    ...       ...  ...             ...
11243  1000695      Manning  P00296942      M     18-25   19               1
11244  1004089  Reichenbach  P00171342      M     26-35   33               0
11245  1001209        Oshin  P00201342      F     36-45   40               0
11246  1004023       Noonan  P00059442      M     36-45   37               0
11247  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
...               ...       ...              ...              ...     ...
11243     Maharashtra   Western         Chemical           Office       4
11244         Haryana  Northern       Healthcare       Veterinary       3
11245  Madhya Pradesh   Central          Textile           Office       4
11246       Karnataka  Southern      Agriculture           Office       3
11247     Maharashtra   Western       Healthcare           Office       3

       Amount
0       23952
1       23934
2       23924
3       23912
4       23877
...       ...
11243     370
11244     367
11245     213
11246     206
11247     188

[11239 rows x 13 columns]>
```

In [23]:
```
df.columns
```

Out[23]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
In [24]: df[['Cust_name','Age','Occupation','Amount']].describe() #describe show only numeric type as we can see cust_name and Occupation
```

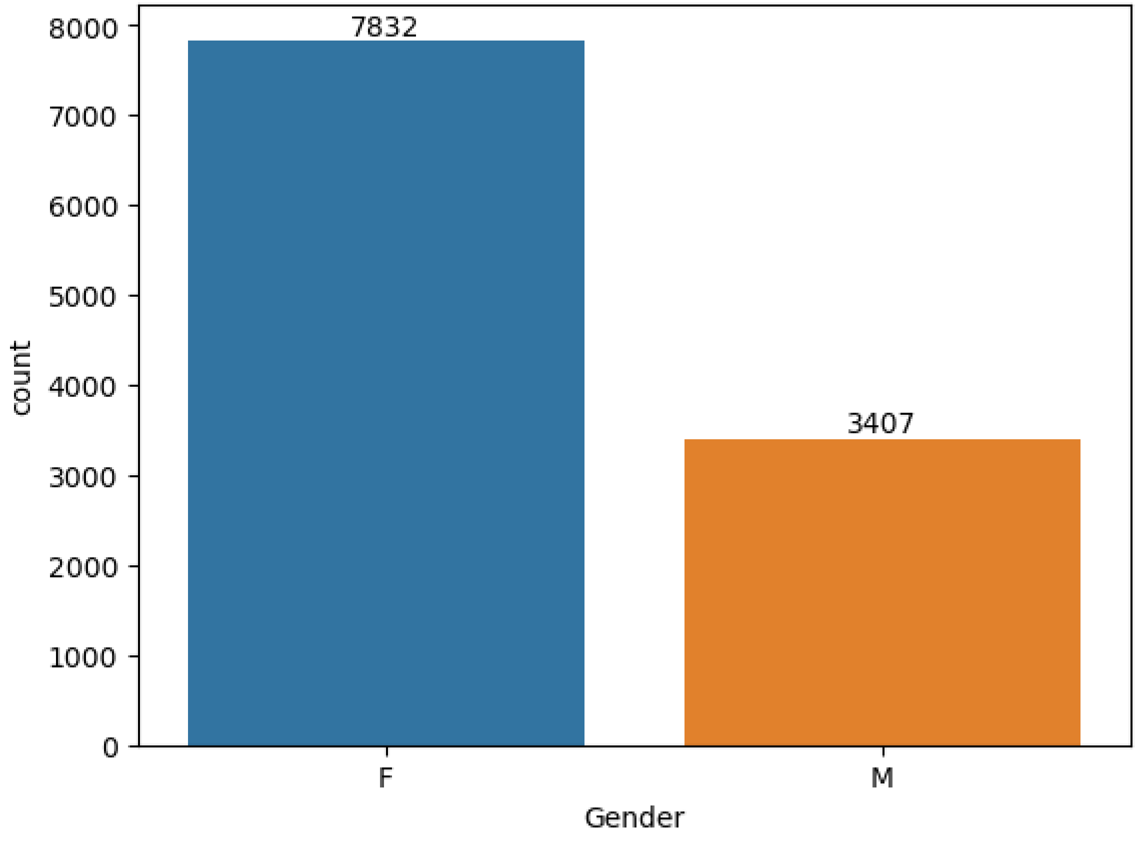
Out[24]:

	Age	Amount
count	11239.000000	11239.000000
mean	35.410357	9453.610553
std	12.753866	5222.355168
min	12.000000	188.000000
25%	27.000000	5443.000000
50%	33.000000	8109.000000
75%	43.000000	12675.000000
max	92.000000	23952.000000

```
In [25]: df.columns
```

```
Out[25]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
In [26]: ax=sns.countplot(x='Gender',data=df)
         for bars in ax.containers:
             ax.bar_label(bars)
```



most of the buyers are female as described above

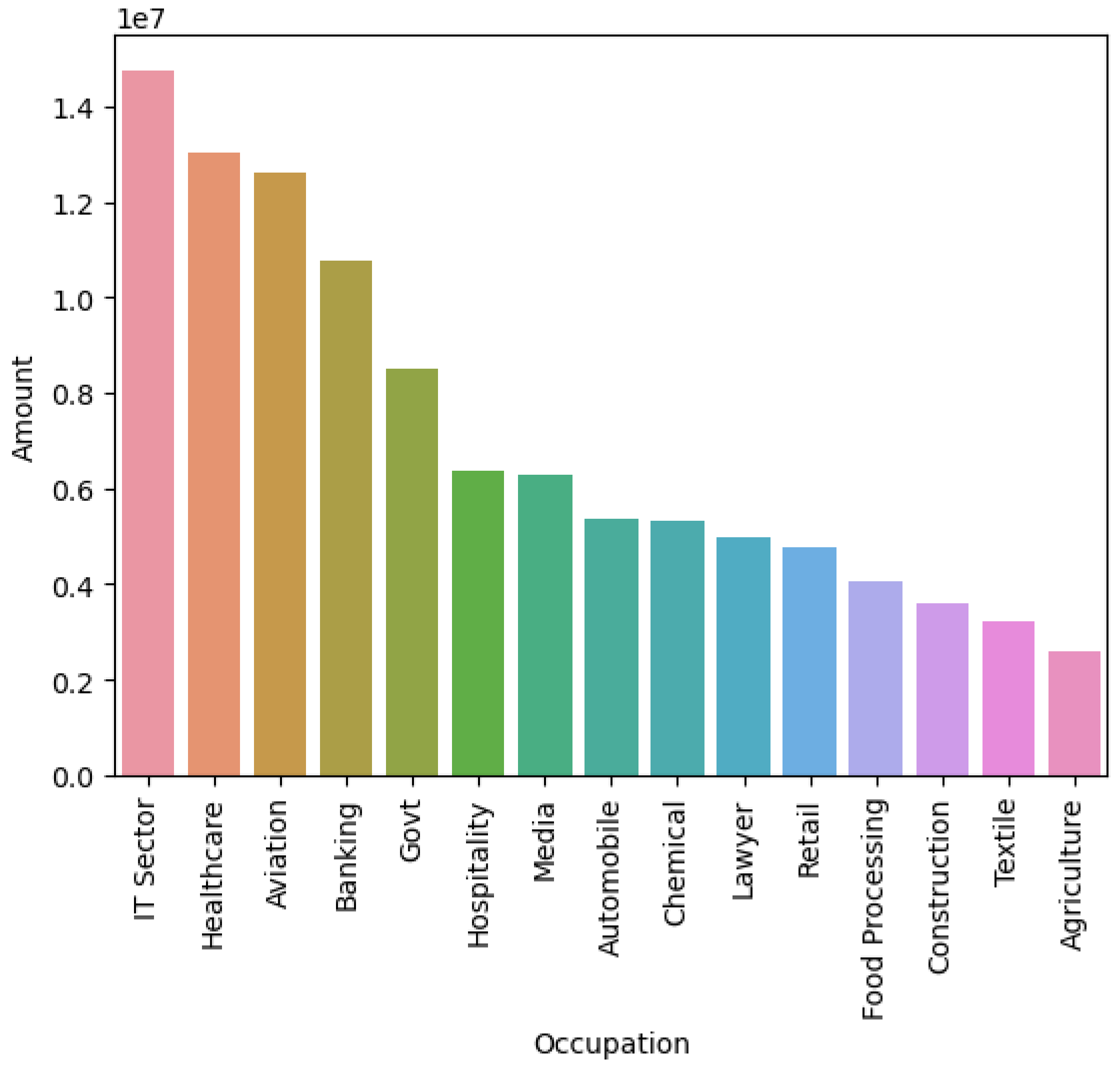
```
In [27]: oc=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
```

In [28]:

```
bx=sns.barplot(x='Occupation' ,y='Amount',data=oc)
plt.xticks(rotation=90)

#for bars in bx.containers: bx.bar_label(bars)
plt.show
```

Out[28]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



most buyrs are from It Sector,Healthvare and Aviation respectively

In [29]:

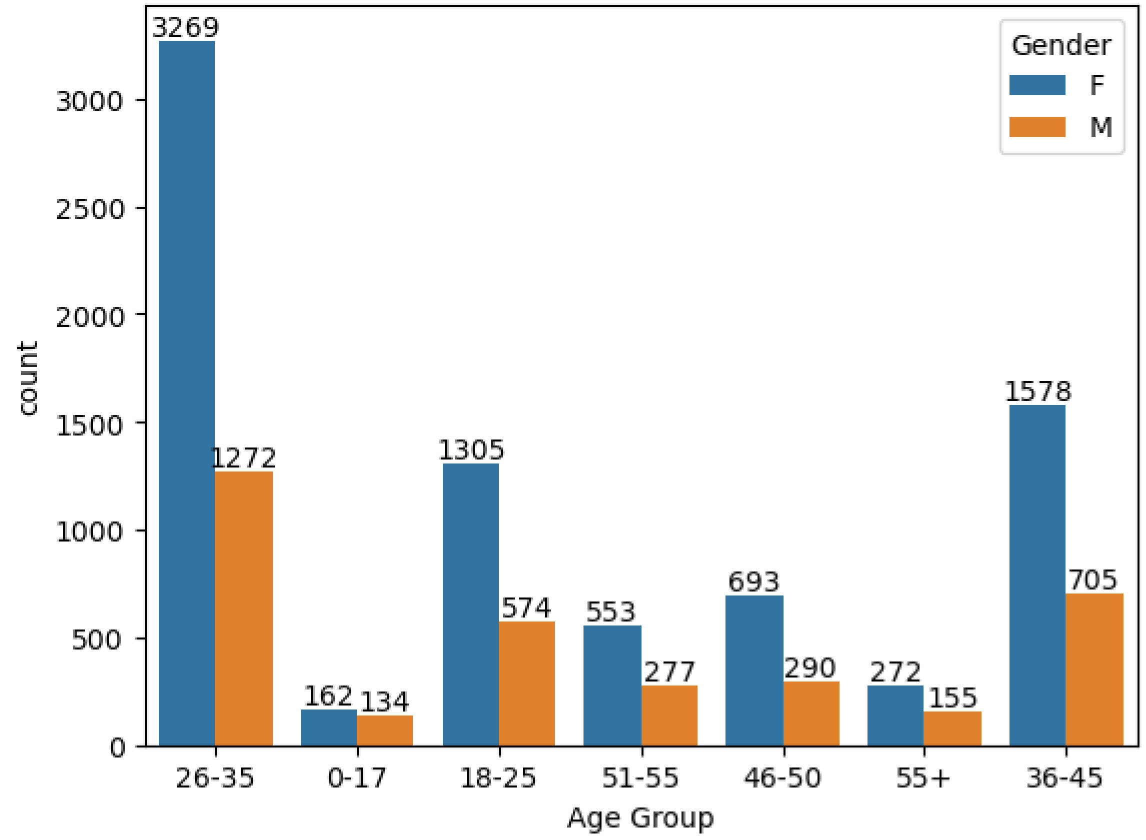
```
df.columns
```

Out[29]:

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

In [30]:

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

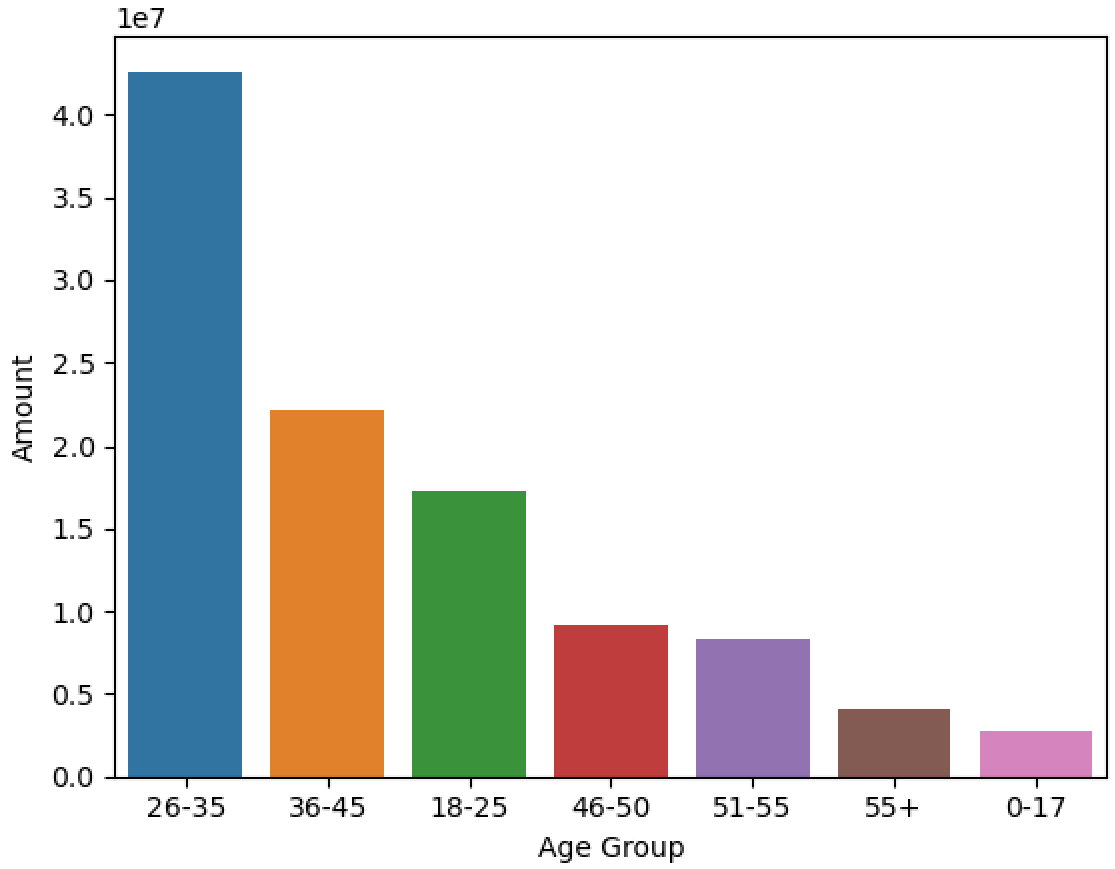


In [31]:

```
cx=df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(data=cx, x='Age Group',y='Amount')
```

Out[31]:

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



In [32]:

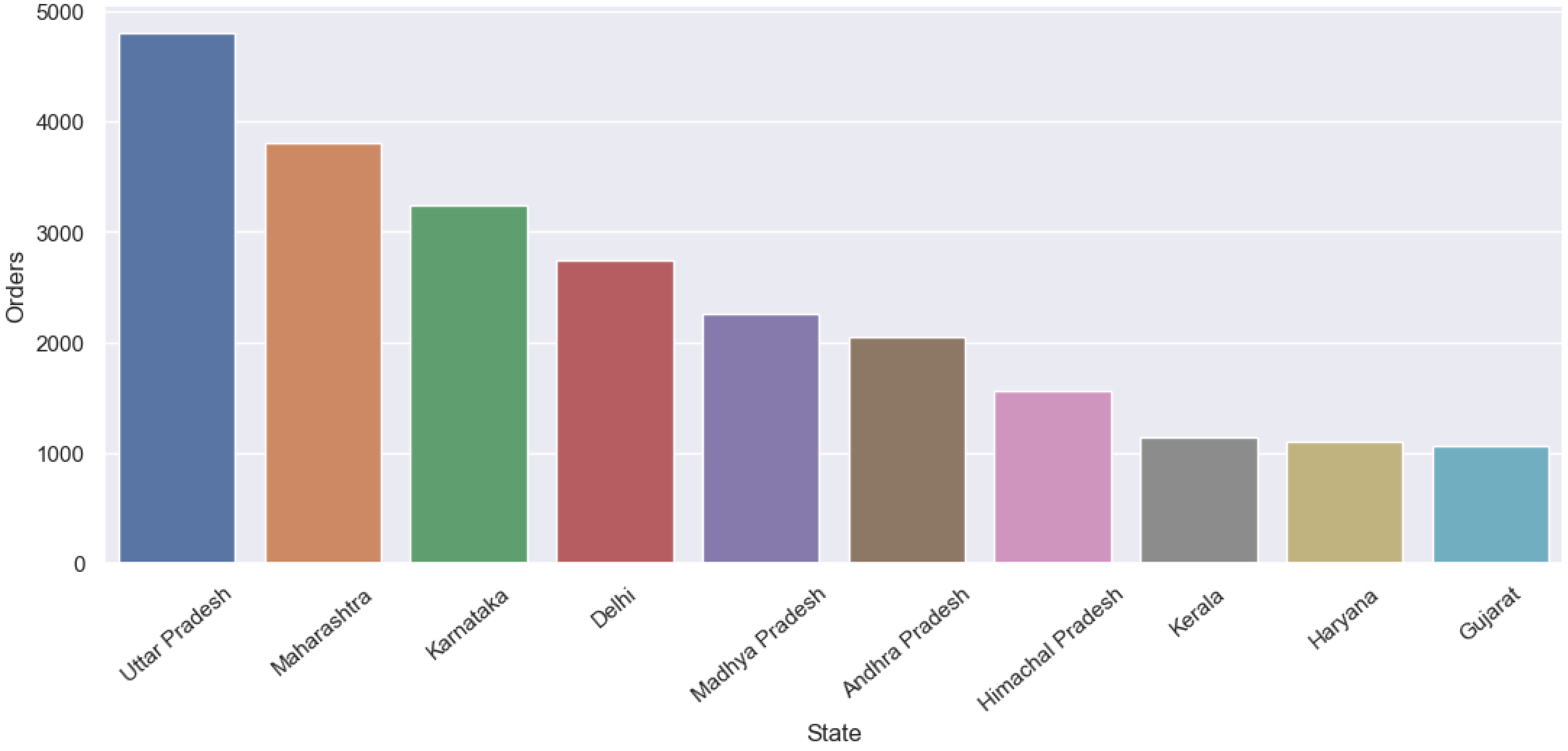
```
dx=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)
```

In [33]:

```
sns.set(rc={'figure.figsize':(13,5)})
plt.xticks(rotation=40)
sns.barplot(data=dx , x='State',y='Orders')
#top 10 states based on orders made  by tyhe customrs
```

Out[33]:

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



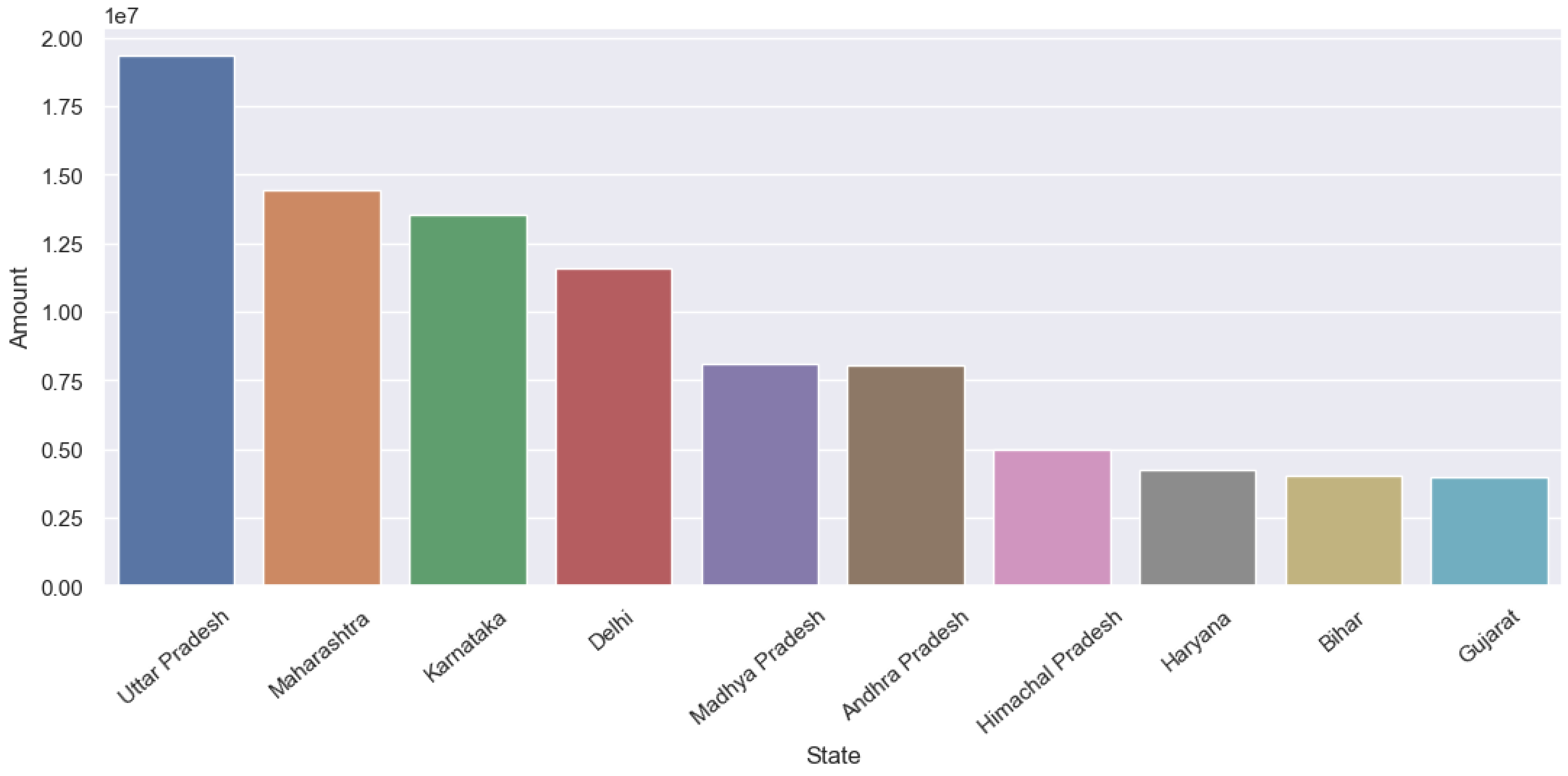
from the above graph we can see that most of the oreders from uttarpradesh,Maharashtra and Karnataka respectively

In [34]:

```
ex=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(10)
sns.barplot(x='State',y='Amount',data=ex)
plt.xticks(rotation=40)
plt.show
#top 10 states with moat spent amount by customers
```

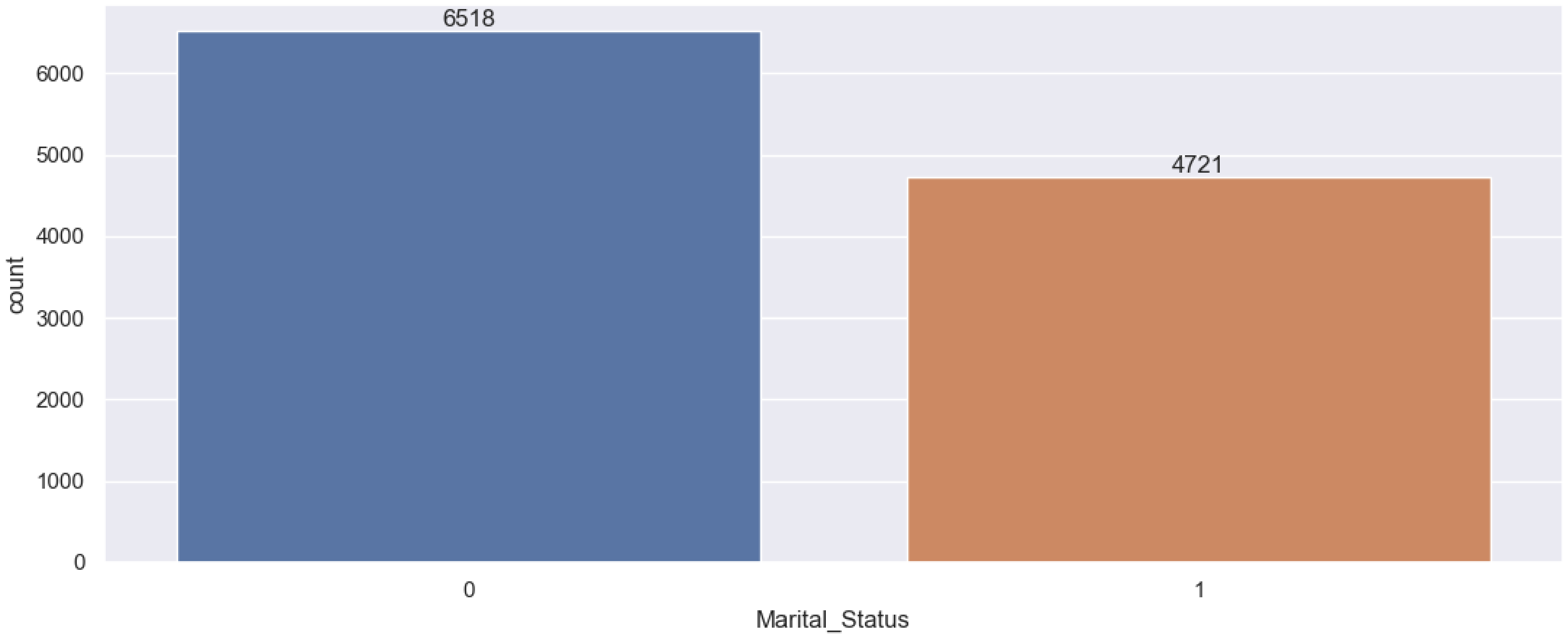
Out[34]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [35]:

```
ex=sns.countplot(data=df, x='Marital_Status')
sns.set(rc={'figure.figsize':(5,3)})
for bars in ex.containers:
    ex.bar_label(bars)
```

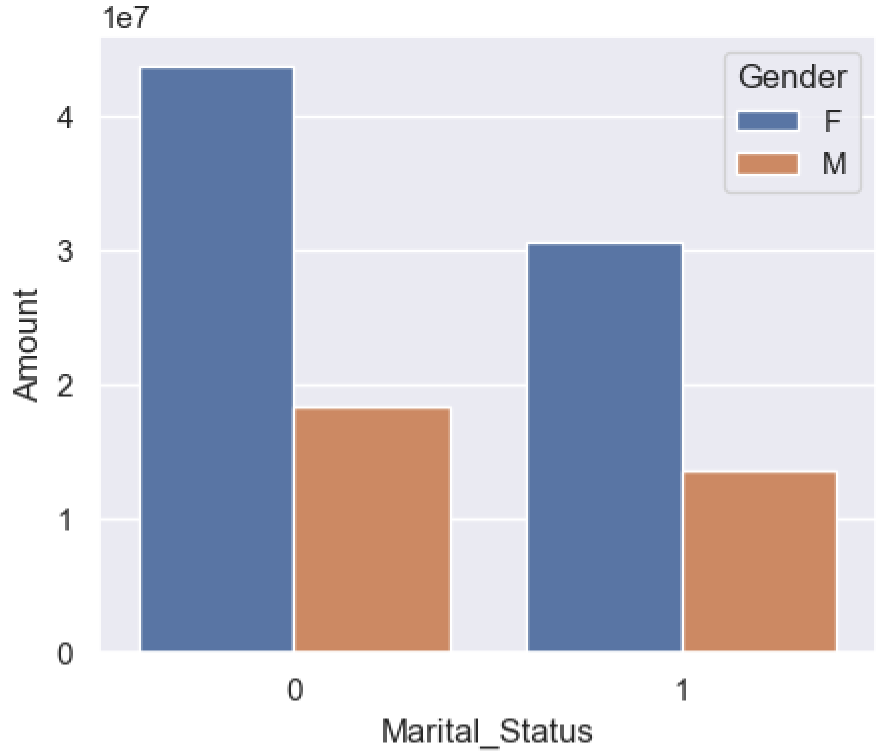


In [36]:

```
mar= df.groupby (['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(5,4)})
sns.barplot(data=mar, x = 'Marital_Status', y='Amount',hue='Gender')
```

Out[36]:

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



from above graph we can see that most of the buyers are unmarried womens

In [37]:

```
df.columns
```

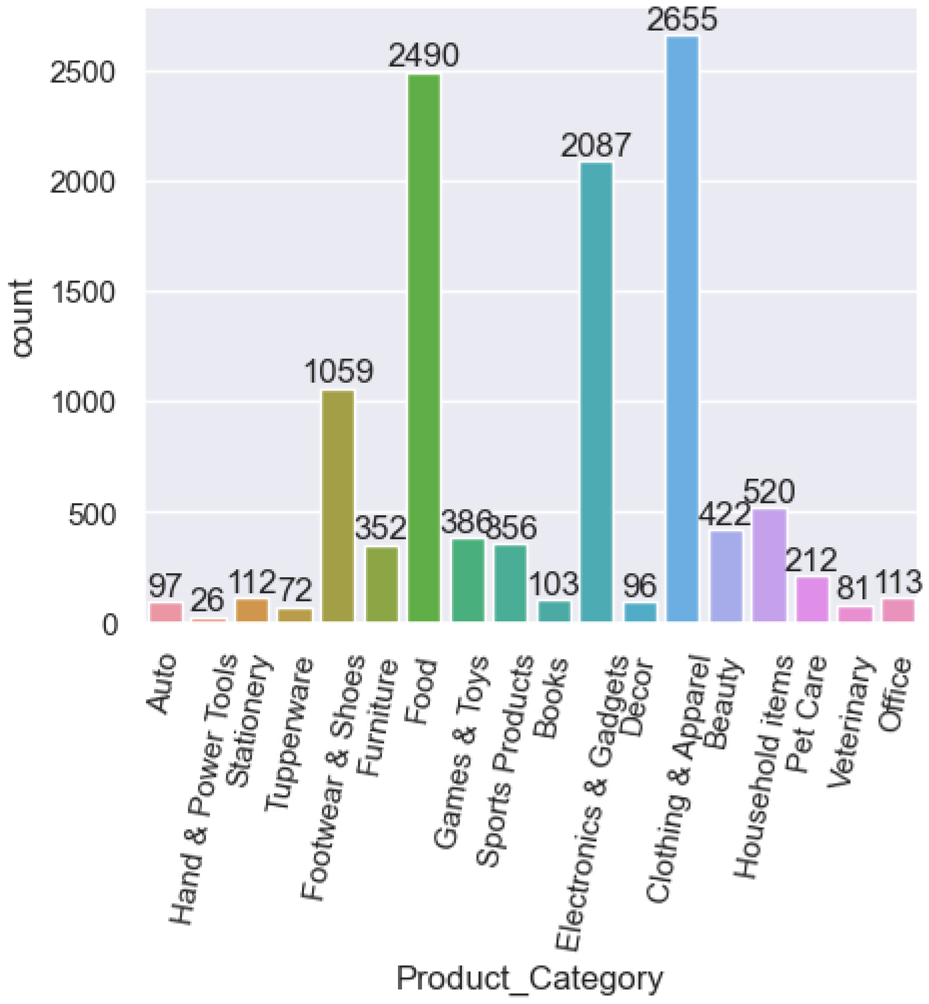
Out[37]:

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

Product_category

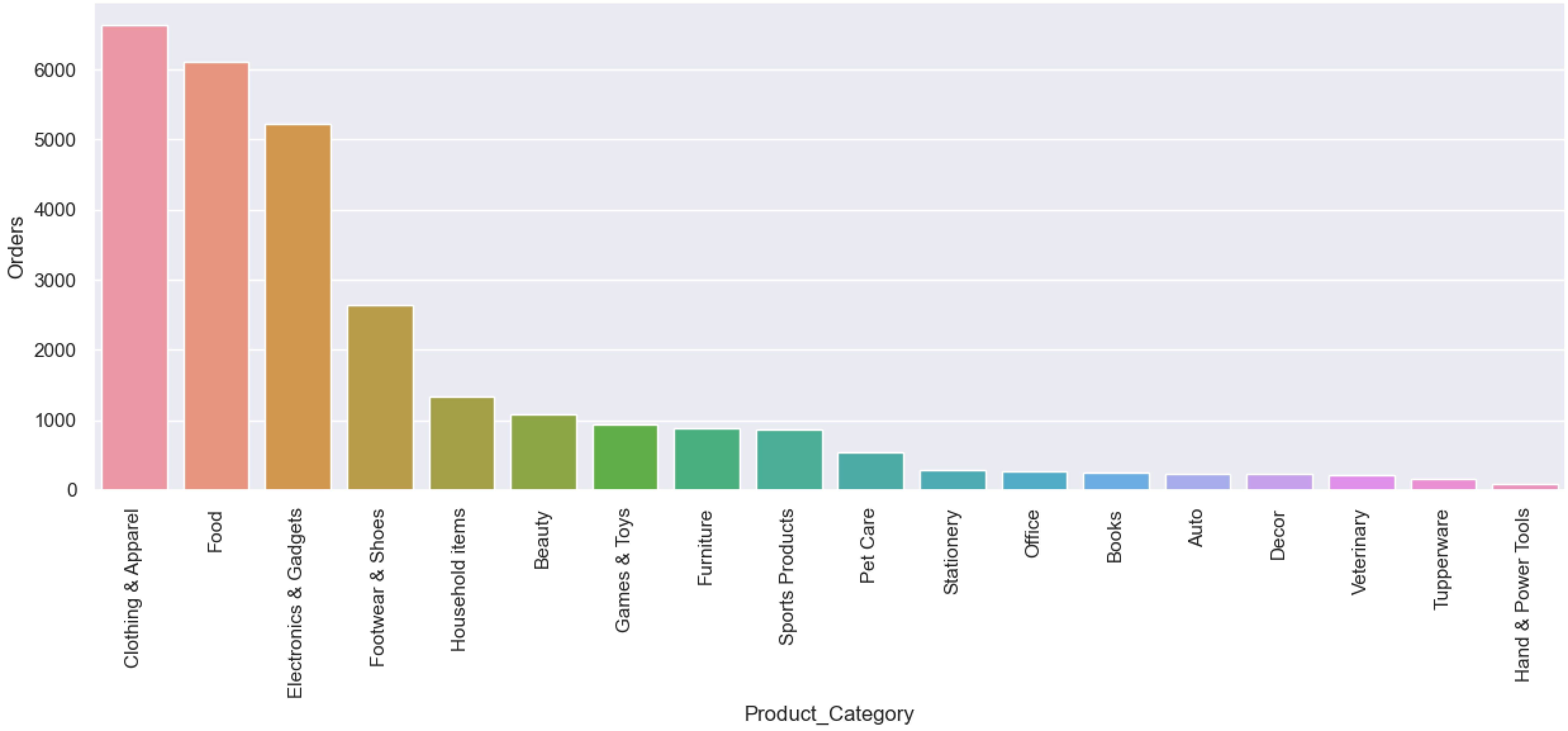
In [38]:
```
cx=sns.countplot(data=df,x='Product_Category')
sns.set(rc={'figure.figsize':(15,5)})
for bars in cx.containers:
    cx.bar_label(bars)
plt.xticks(rotation=80)
```

Out[38]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17]),
 [Text(0, 0, 'Auto'),
  Text(1, 0, 'Hand & Power Tools'),
  Text(2, 0, 'Stationery'),
  Text(3, 0, 'Tupperware'),
  Text(4, 0, 'Footwear & Shoes'),
  Text(5, 0, 'Furniture'),
  Text(6, 0, 'Food'),
  Text(7, 0, 'Games & Toys'),
  Text(8, 0, 'Sports Products'),
  Text(9, 0, 'Books'),
  Text(10, 0, 'Electronics & Gadgets'),
  Text(11, 0, 'Decor'),
  Text(12, 0, 'Clothing & Apparel'),
  Text(13, 0, 'Beauty'),
  Text(14, 0, 'Household items'),
  Text(15, 0, 'Pet Care'),
  Text(16, 0, 'Veterinary'),
  Text(17, 0, 'Office')])
```



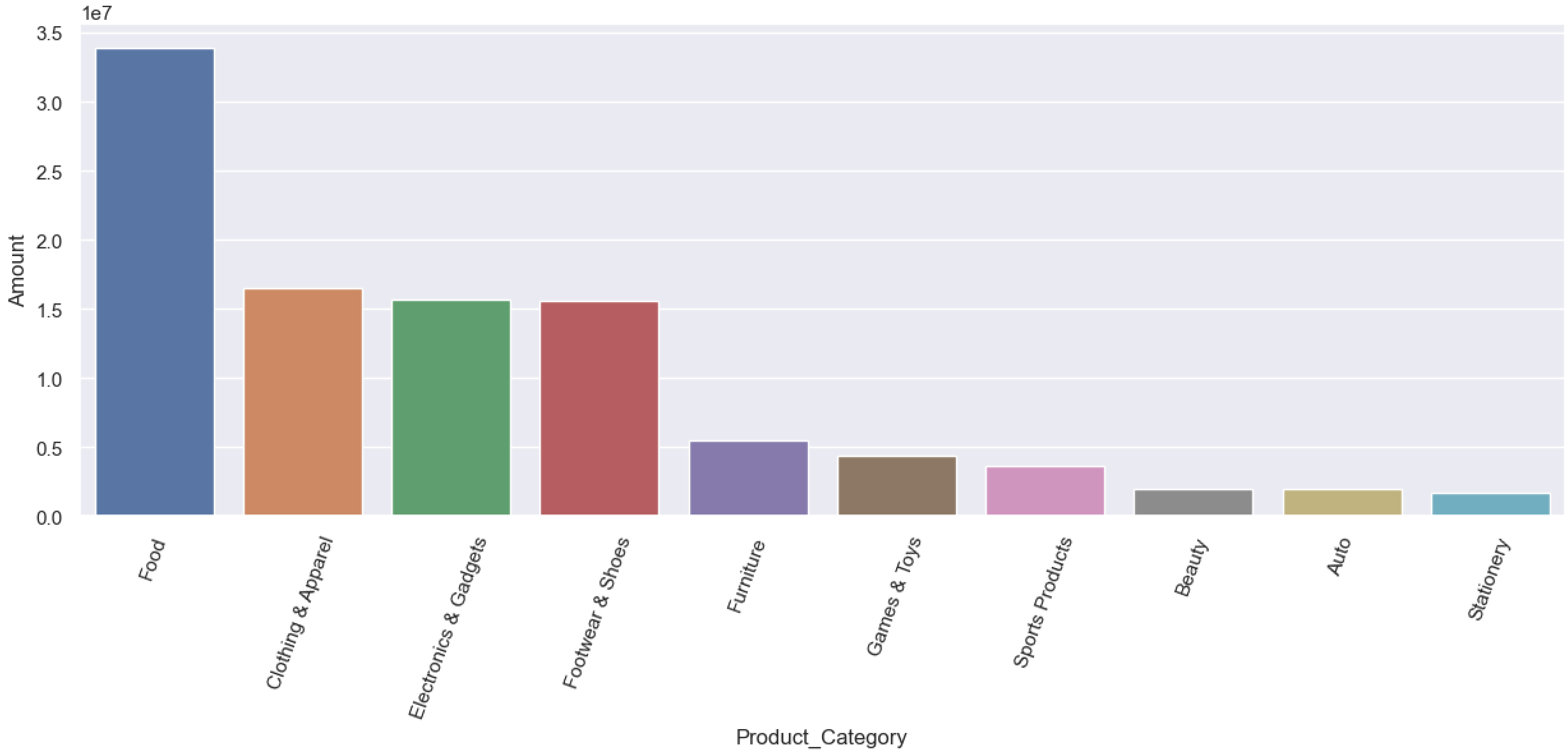
```
In [39]: dx=df.groupby(['Product_Category'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False)
         sns.barplot(data=dx,x='Product_Category',y='Orders')
         plt.xticks(rotation=90)
```

```
Out[39]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                 17]),
          [Text(0, 0, 'Clothing & Apparel'),
           Text(1, 0, 'Food'),
           Text(2, 0, 'Electronics & Gadgets'),
           Text(3, 0, 'Footwear & Shoes'),
           Text(4, 0, 'Household items'),
           Text(5, 0, 'Beauty'),
           Text(6, 0, 'Games & Toys'),
           Text(7, 0, 'Furniture'),
           Text(8, 0, 'Sports Products'),
           Text(9, 0, 'Pet Care'),
           Text(10, 0, 'Stationery'),
           Text(11, 0, 'Office'),
           Text(12, 0, 'Books'),
           Text(13, 0, 'Auto'),
           Text(14, 0, 'Decor'),
           Text(15, 0, 'Veterinary'),
           Text(16, 0, 'Tupperware'),
           Text(17, 0, 'Hand & Power Tools')])
```



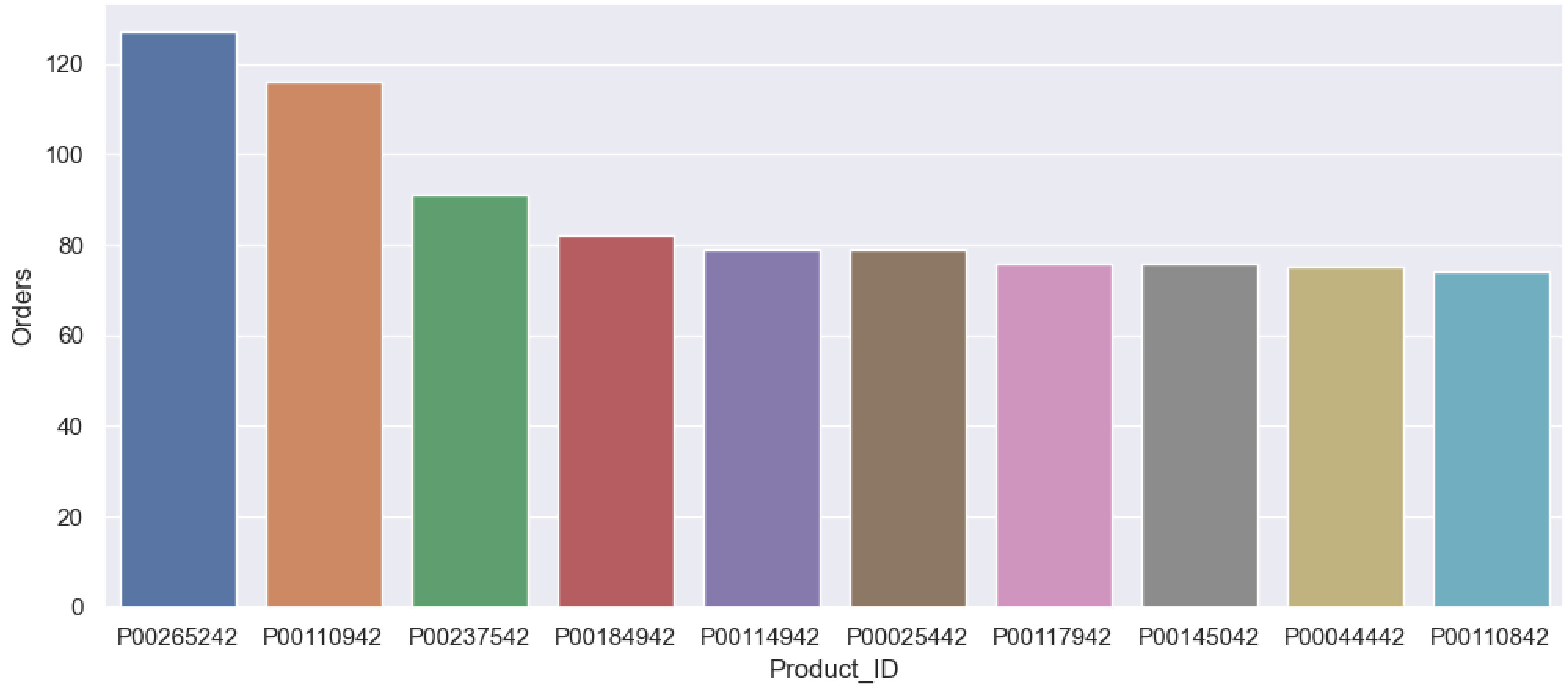
```
In [40]: px=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

         sns.barplot(x='Product_Category',y='Amount',data=px)

         plt.xticks(rotation=70)
```

```
Out[40]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
          [Text(0, 0, 'Food'),
           Text(1, 0, 'Clothing & Apparel'),
           Text(2, 0, 'Electronics & Gadgets'),
           Text(3, 0, 'Footwear & Shoes'),
           Text(4, 0, 'Furniture'),
           Text(5, 0, 'Games & Toys'),
           Text(6, 0, 'Sports Products'),
           Text(7, 0, 'Beauty'),
           Text(8, 0, 'Auto'),
           Text(9, 0, 'Stationery')])
```



```
In [41]: sells=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(10)
         sns.set(rc={'figure.figsize':(12,5)})
         sns.barplot(data=sells, x='Product_ID', y='Orders')
```

```
Out[41]: <AxesSubplot:xlabel='Product_ID', ylabel='Orders'>
```



```
In [42]: #top 10 selling product
```

conclusion:

"single women age group 25-35 years from uttarpradesh,Maharashtra and Karnataka working in IT sector ,Healthcare and Aviation are more likely to buy Products from food,clothing and electronic categories