Diwali sales analysis

Problem statement:

In this project, we will analyze Diwali sales data to understand customer buying behavior during the festival.
The data includes important details like:

- Gender: How male and female customers differ in their purchases.
- Age Group: Which age groups spend the most and what products they prefer.
- Marital Status: How being single or married affects spending habits.
- Occupation: How different jobs influence what people buy and how much they spend.
- Product Category: Which types of products are most popular among different customers.

objectives

- Understand Customers: Learn about the different types of customers (gender, age, marital status, job).
- Spending Patterns: Find out how much different groups spend on average.
- Popular Products: Identify which products are favorites for various customer groups.
- Business Insights: Provide useful information for businesses to improve their marketing and offers during Diwali.

In [174…
```
#importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [41]:
```
df=pd.read_csv("diwalisales.csv",encoding="unicode_escape")
```

In [42]:
```
df.head()
```

Out[42]:

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

In [43]:
```
df.sample(5)
```

Out[43]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category
1074	1001352	Sumeet	P00262642	F	36-45	42	0	Karnataka	Southern	Banking	Footwear & Shoes
1551	1002688	Dionis	P00171542	F	26-35	33	1	Karnataka	Southern	Lawyer	Footwear & Shoes
2809	1003265	Arti	P00346542	F	26-35	34	0	Himachal Pradesh	Northern	Govt	Games & Toys
1039	1002173	Kritika	P00302642	F	51-55	54	1	Uttar Pradesh	Central	Healthcare	Footwear & Shoes
708	1002257	Collister	P00117442	M	18-25	22	0	Maharashtra	Western	Chemical	Food

In [44]:
```
df.shape
```

Out[44]:
```
(11251, 15)
```

In [45]:
```
df.columns
```

Out[45]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

In [46]:
```
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
```

In [47]:
```
df.describe()
```

Out[47]:

	User_ID	Age	Marital_Status	Orders	Amount	Status	unnamed1
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000	0.0	0.0
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858	NaN	NaN
std	1.716125e+03	12.754122	0.493632	1.115047	5222.355869	NaN	NaN
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000	NaN	NaN
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.000000	NaN	NaN
50%	1.003065e+06	33.000000	0.000000	2.000000	8109.000000	NaN	NaN
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000	NaN	NaN
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000	NaN	NaN

In [48]:
```
df.isnull().sum()
```

Out[48]:
```
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
Status             11251
unnamed1           11251
dtype: int64
```

In [49]:
```
df.drop(['Status','unnamed1'], axis=1, inplace=True)
```

In [38]:
```
df.head()
```

Out[38]:

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

In [54]:
```
# again checking null values
df.isnull().sum()
```

Out[54]:
```
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
```

In [55]:
```
# Removing null values
df.dropna(inplace=True)
```

In [61]:
```
df.dtypes
```

Out[61]:
```
User_ID              int64
Cust_name           object
Product_ID          object
Gender              object
Age Group           object
Age                  int64
Marital_Status       int64
State               object
Zone                object
Occupation          object
Product_Category    object
Orders               int64
Amount               int64
dtype: object
```

Exploratory data analysis

In [165…
```
df.head()
```

Out[165…

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

In [168…
```
df["State"].nunique()
```

Out[168… 16

In [170…
```
df["Zone"].unique()
```

Out[170…
```
array(['Western', 'Southern', 'Central', 'Northern', 'Eastern'],
      dtype=object)
```

In [172…
```
df["Product_Category"].nunique()
```

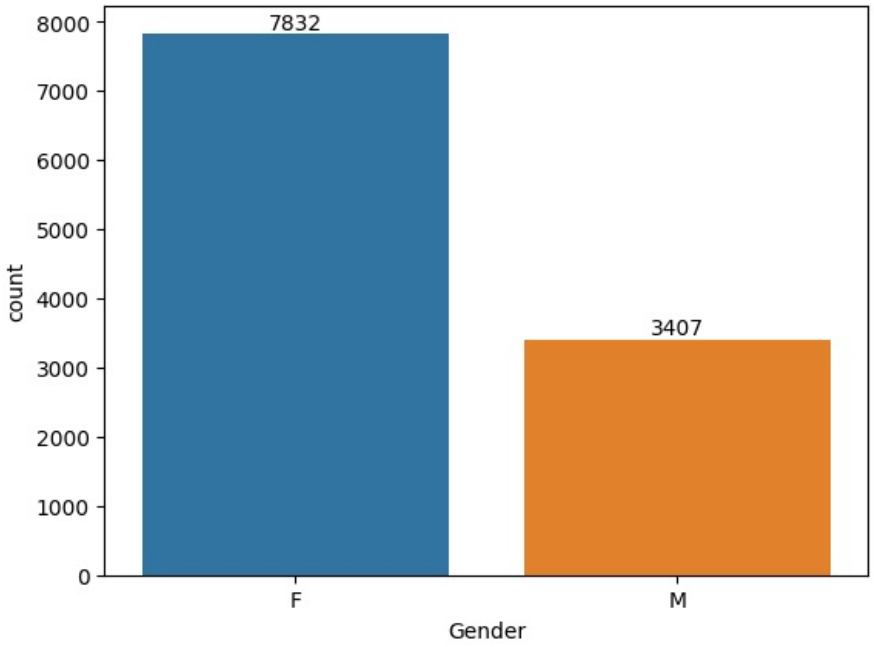
Out[172… 18

In [173…
```
df["Product_Category"].value_counts()
```

Out[173…
```
Product_Category
Clothing & Apparel       2655
Food                     2490
Electronics & Gadgets    2087
Footwear & Shoes         1059
Household items           520
Beauty                    422
Games & Toys              386
Sports Products           356
Furniture                 352
Pet Care                  212
Office                    113
Stationery                112
Books                     103
Auto                       97
Decor                      96
Veterinary                 81
Tupperware                 72
Hand & Power Tools         26
Name: count, dtype: int64
```

Gender

In [68]:
```
ax=sns.countplot(x="Gender",data=df,hue="Gender",legend=False)
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



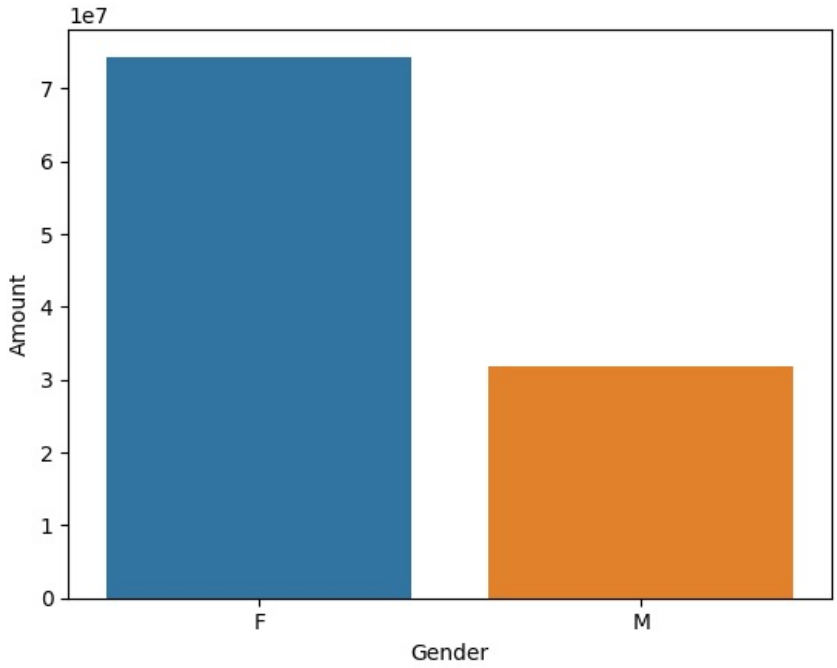
In [72]:
```
df.groupby(["Gender"],as_index=False)["Amount"].sum()
```

Out[72]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [73]:
```
sales_by_gender=df.groupby(["Gender"],as_index=False)["Amount"].sum()
```

In [76]:
```
sns.barplot(x="Gender", y="Amount", data=sales_by_gender,hue="Gender",legend=False)
plt.show()
```



Age

In [77]:
```
df.columns
```

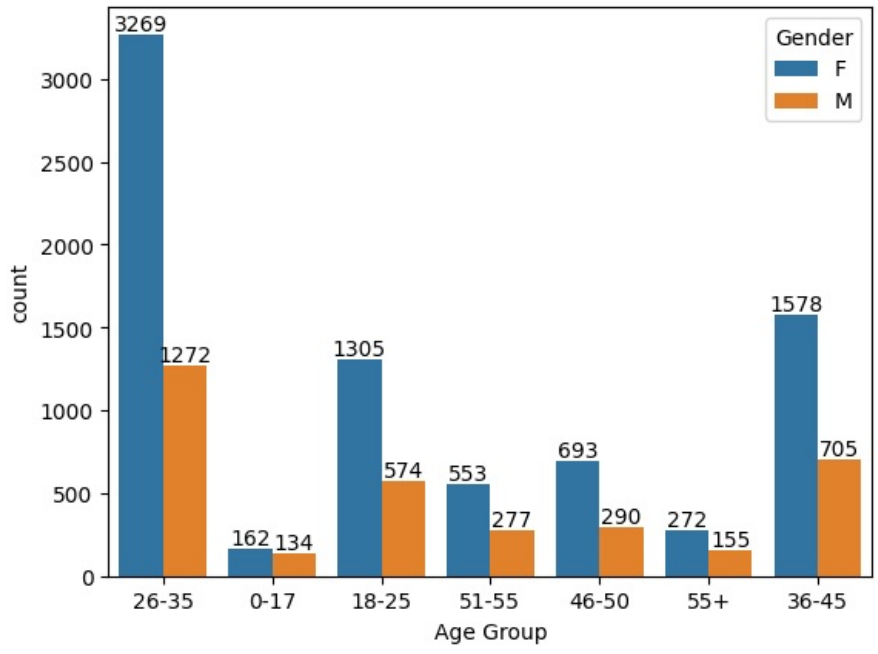
Out[77]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [78]:
```
df.head()
```

Out[78]:

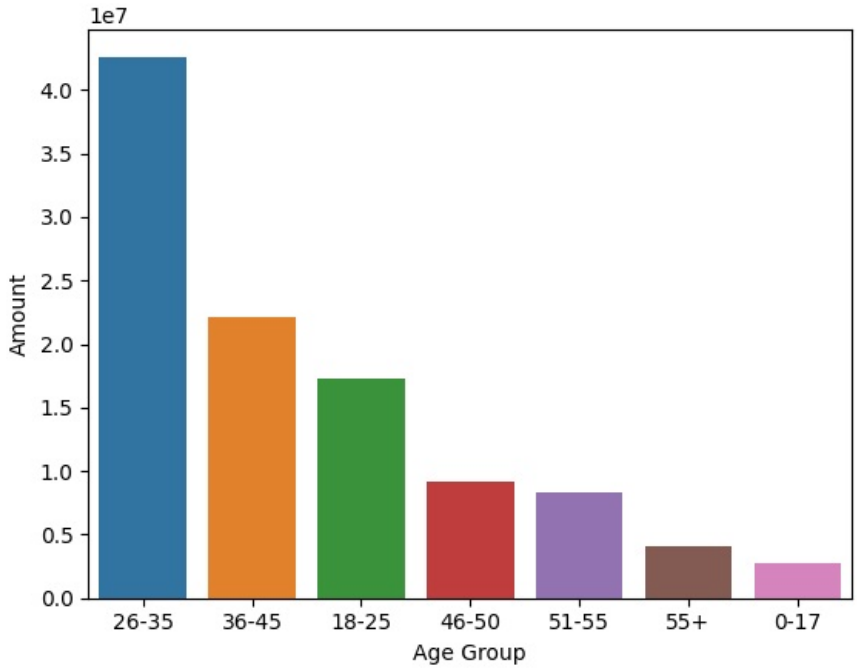
	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar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

In [81]:
```
ax=sns.countplot(data=df, x="Age Group", hue="Gender")
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



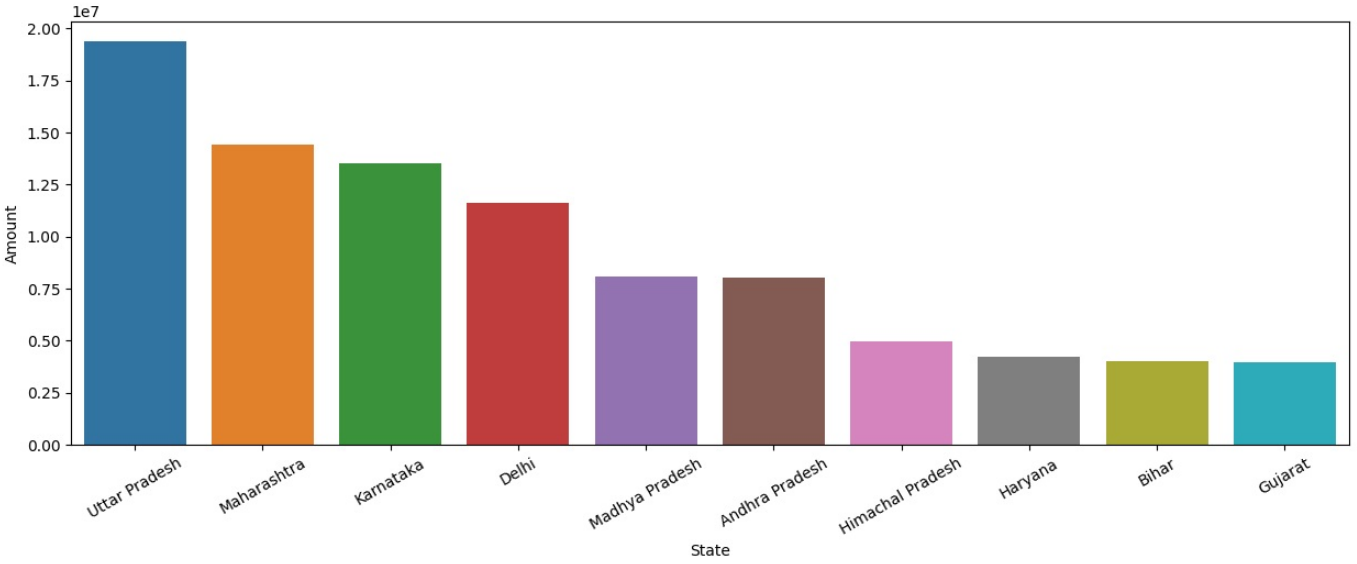
In [89]:
```
sales_by_age_group=df.groupby(["Age Group"],as_index=False)["Amount"].sum().sort_values(by="Amount",ascending=Fa
```

In [90]:
```
sns.barplot(x="Age Group", y="Amount", data=sales_by_age_group,hue="Age Group",legend=False)
plt.show()
```



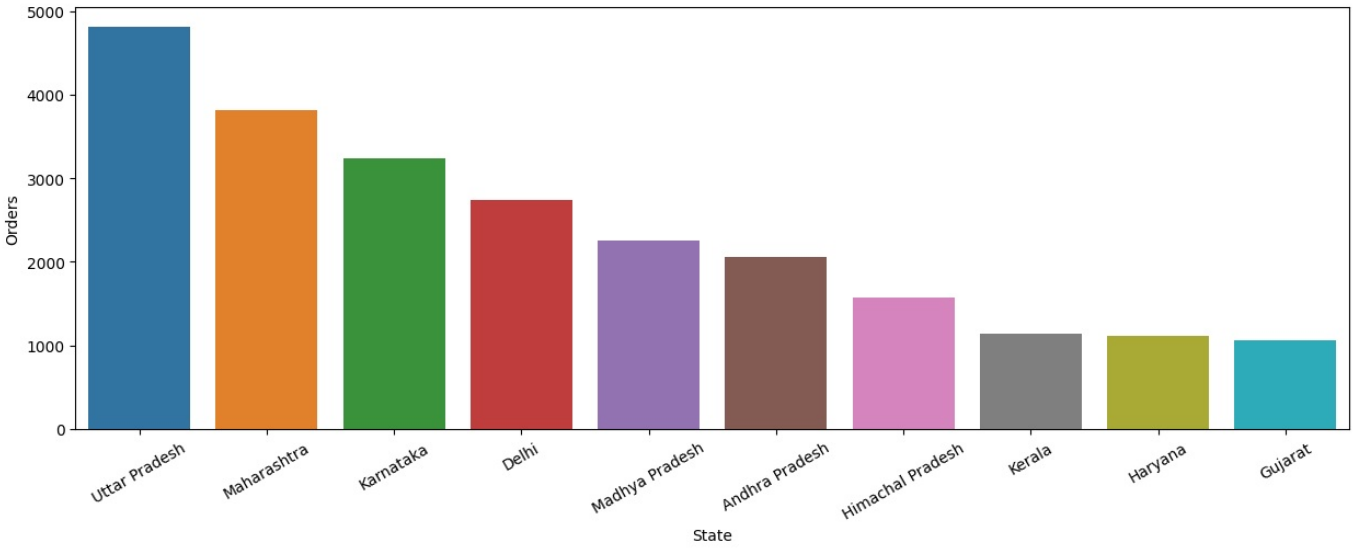
In [107…
```
sales_by_state=df.groupby(["State"],as_index=False)["Amount"].sum().sort_values(by="Amount",ascending=False).hea
```

In [108…
```
plt.figure(figsize=(15,5))
sns.barplot(x="State", y="Amount", data=sales_by_state, hue="State",legend=False)
plt.xticks(rotation=30)
plt.show()
```



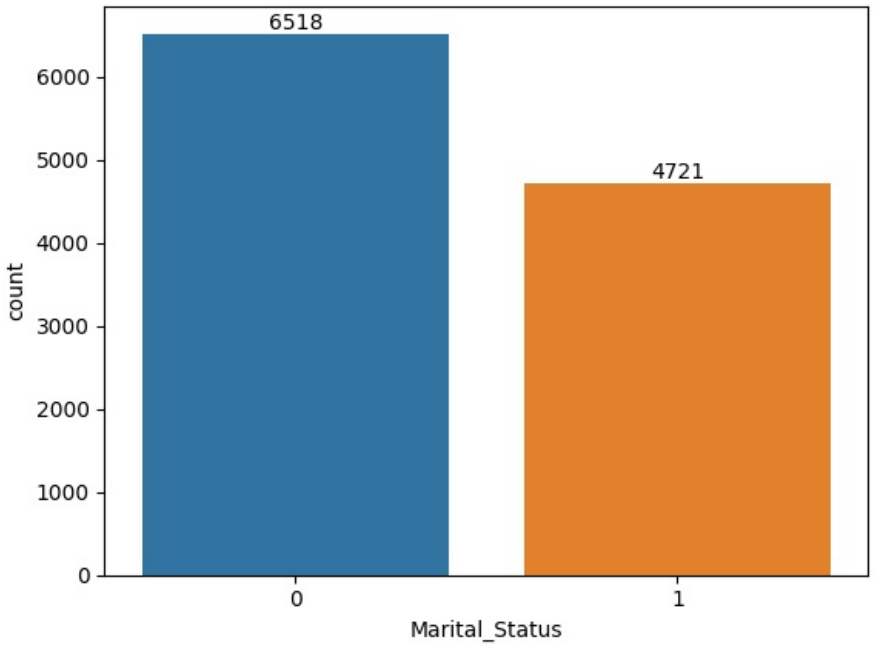
In [109…
```
order_by_state=df.groupby(["State"],as_index=False)["Orders"].sum().sort_values(by="Orders",ascending=False).hea
```

In [110…
```
plt.figure(figsize=(15,5))
sns.barplot(x="State", y="Orders", data=order_by_state, hue="State",legend=False)
plt.xticks(rotation=30)
plt.show()
```



Martial status

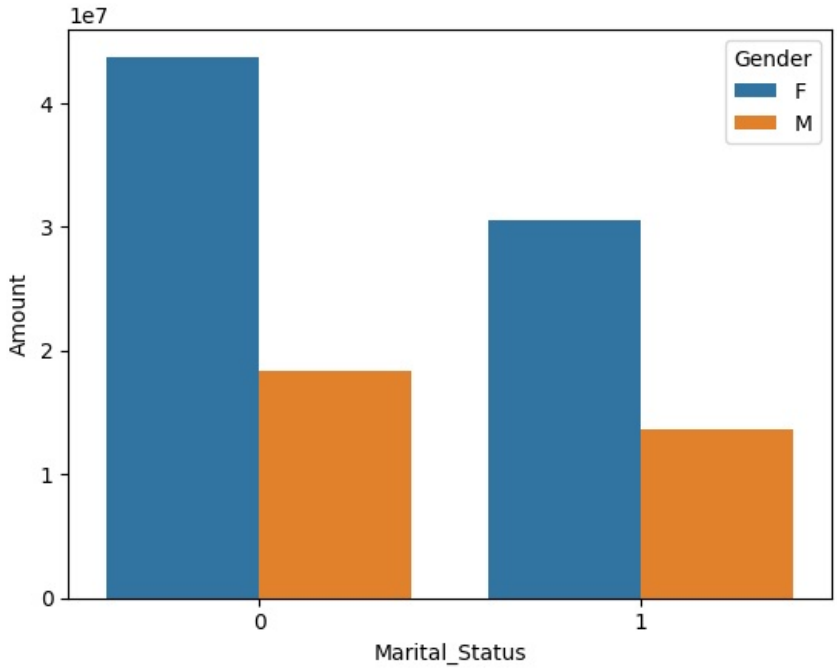
In [111…
```
ax=sns.countplot(x="Marital_Status", data=df, hue="Marital_Status",legend=False)
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



In [113…
```
sales=df.groupby(["Marital_Status","Gender"],as_index=False)["Amount"].sum().sort_values(by="Amount",ascending=
```

In [118…
```
sns.barplot(x="Marital_Status", y="Amount", data=sales, hue="Gender")

plt.show()
```



Occupation

In [119…
```
df.head()
```

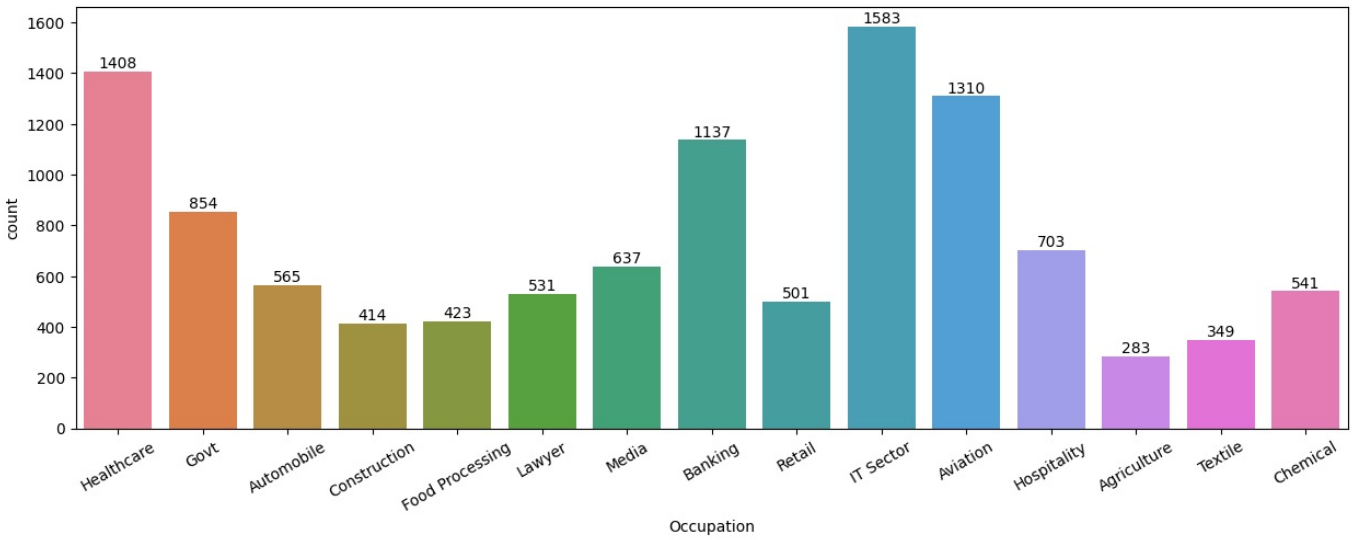
Out[119…

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

In [120…
```
df["Occupation"].unique()
```

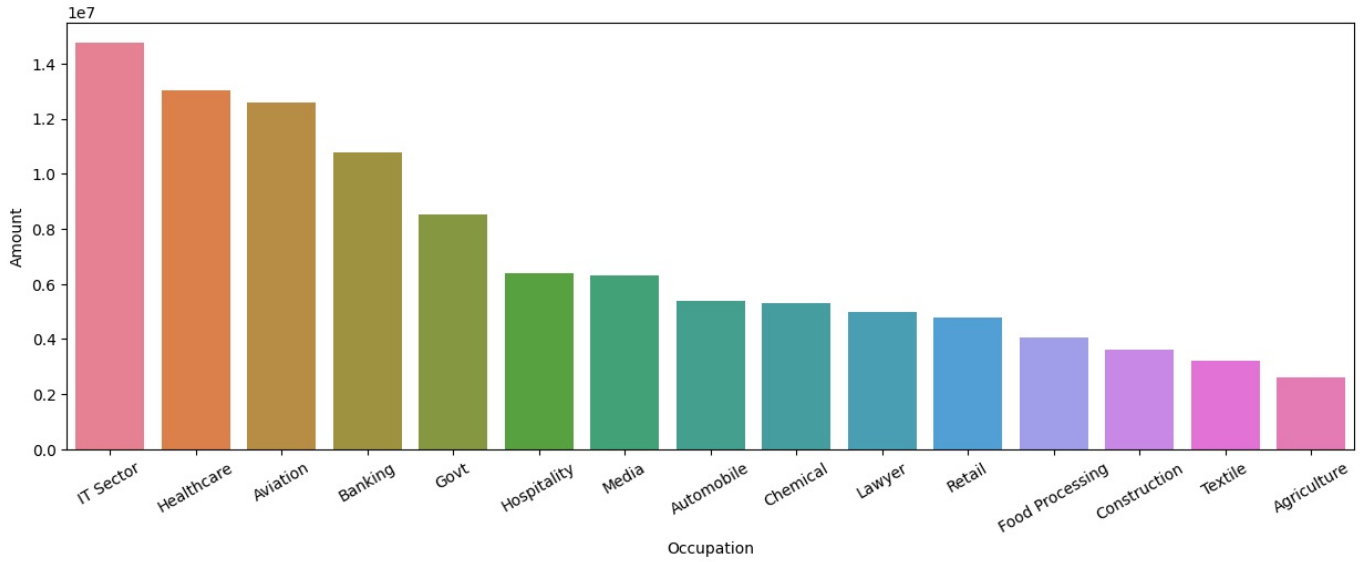
Out[120…
```
array(['Healthcare', 'Govt', 'Automobile', 'Construction',
       'Food Processing', 'Lawyer', 'Media', 'Banking', 'Retail',
       'IT Sector', 'Aviation', 'Hospitality', 'Agriculture', 'Textile',
       'Chemical'], dtype=object)
```

In [125…
```
plt.figure(figsize=(15,5))
ax=sns.countplot(x="Occupation",data=df,hue="Occupation",legend=False)
for container in ax.containers:
    ax.bar_label(container)
plt.xticks(rotation=30)
plt.show()
```



In [130…
```
Sales_by_occupation=df.groupby(["Occupation"],as_index=False)["Amount"].sum().sort_values(by="Amount",ascending
```

In [132…
```
plt.figure(figsize=(15,5))
sns.barplot(x="Occupation", y="Amount", data=Sales_by_occupation,hue="Occupation",legend=False)
plt.xticks(rotation=30)
plt.show()
```



Product category

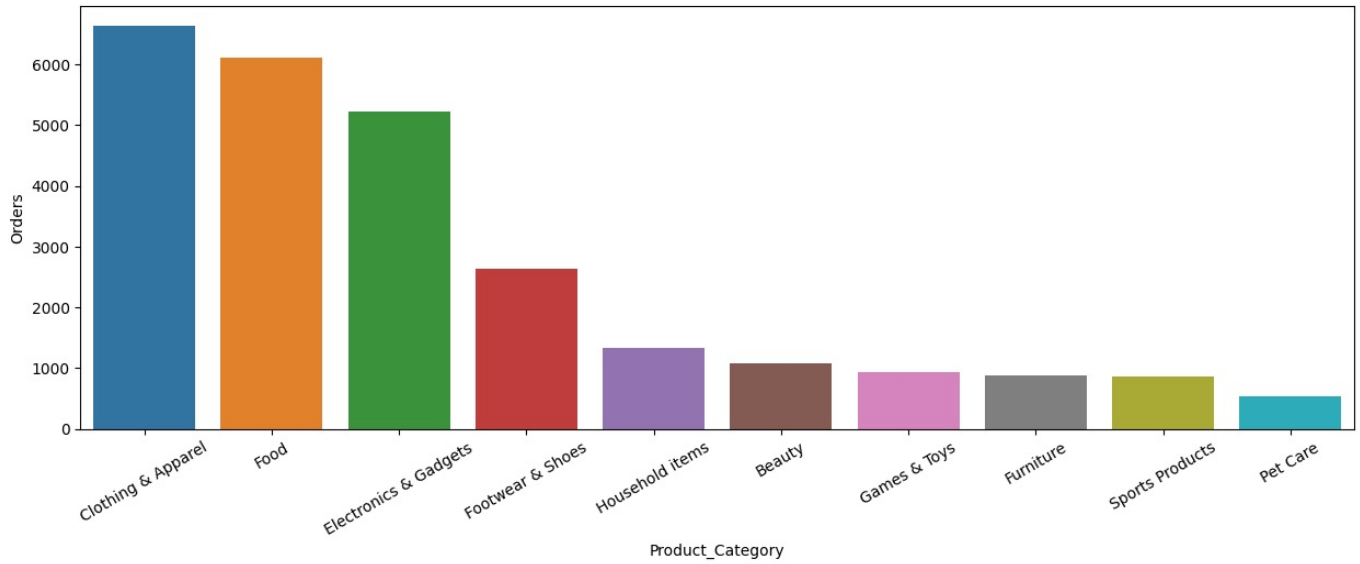
In [133…
```
df.head()
```

Out[133…

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

In [142…
```
#Top 10 sales_by_Product_Category
sales_by_Product_Category=df.groupby(["Product_Category"],as_index=False)["Orders"].sum().sort_values(by="Orders
```

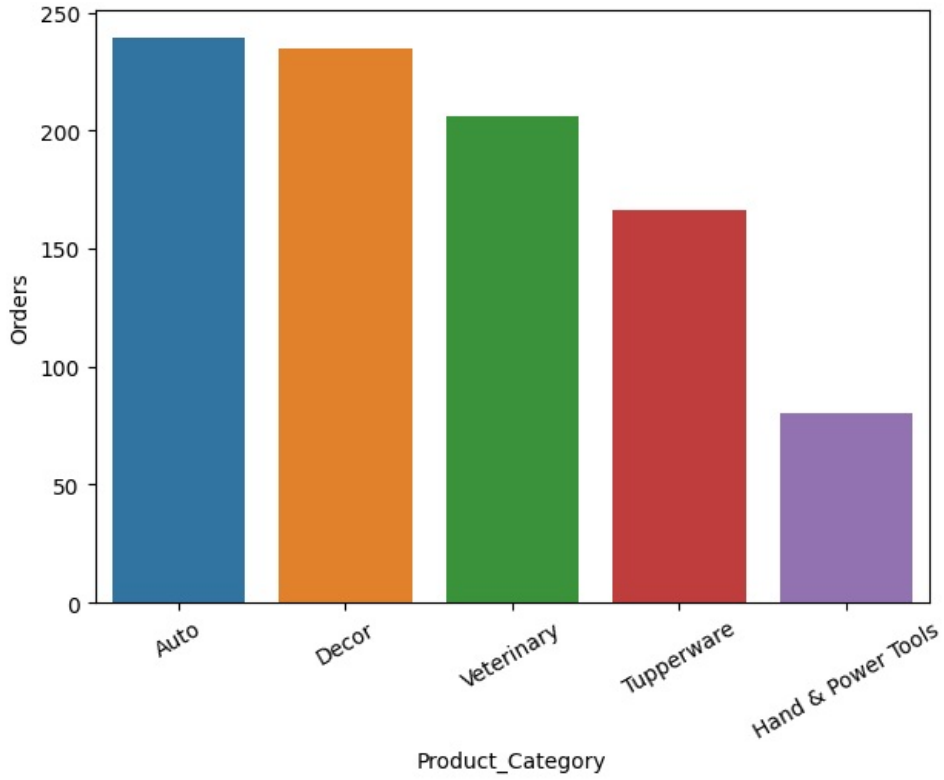
In [143…
```
plt.figure(figsize=(15,5))
sns.barplot(x="Product_Category", y="Orders", data=sales_by_Product_Category, hue="Product_Category",legend=Fals
plt.xticks(rotation=30)
plt.show()
```



In [145…
```
# Bottom 5 by sales_by_Product_Category
sales_by_Product_Category_b=df.groupby(["Product_Category"],as_index=False)["Orders"].sum().sort_values(by="Ord
```

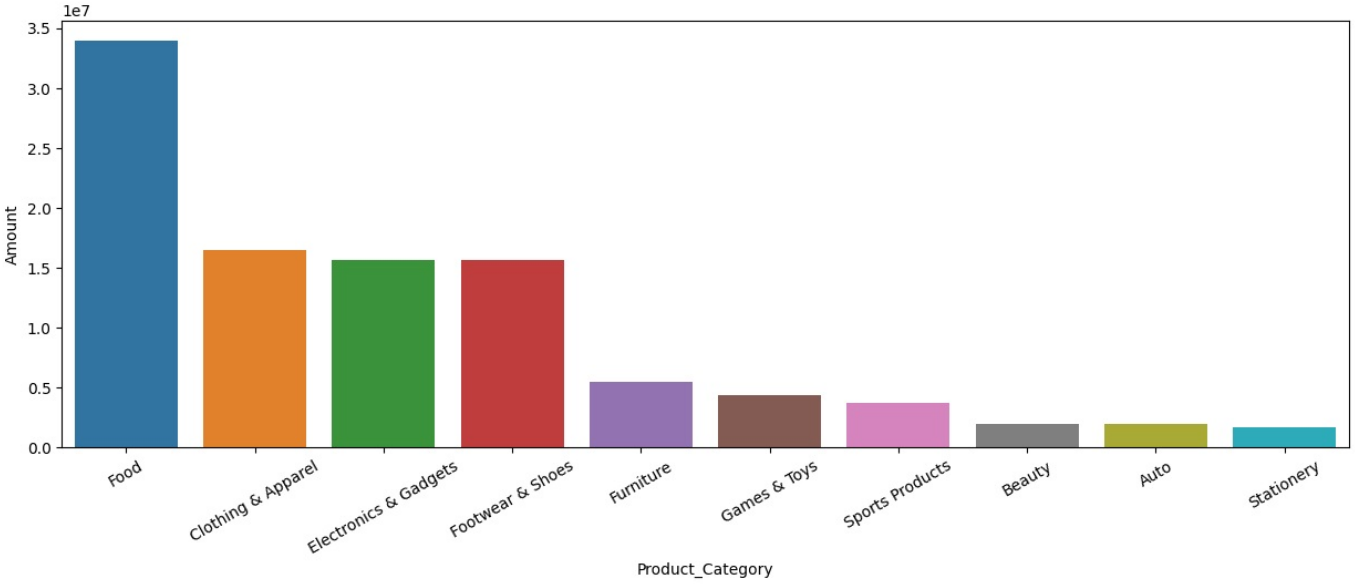
In [148…
```
plt.figure(figsize=(7,5))
sns.barplot(x="Product_Category", y="Orders", data=sales_by_Product_Category_b, hue="Product_Category",legend=Fa
```

```
plt.xticks(rotation=30)
plt.show()
```



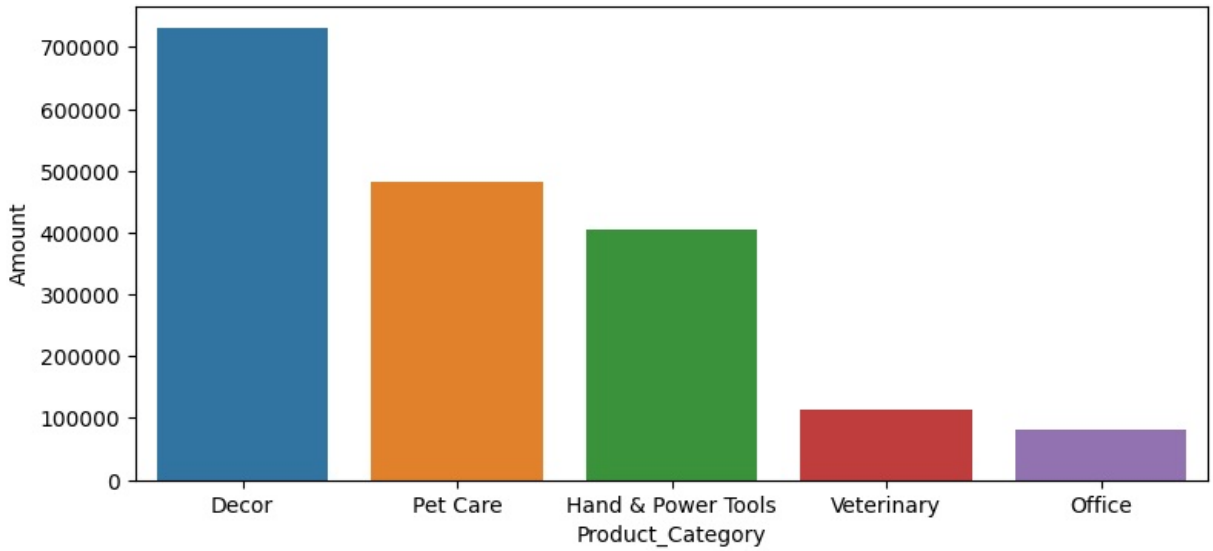
In [156…
```
# sales state by top 10
sales_state_Top=df.groupby(["Product_Category"],as_index=False)["Amount"].sum().sort_values(by="Amount",ascendi
```

In [158…
```
plt.figure(figsize=(15,5))
sns.barplot(x="Product_Category", y="Amount", data=sales_state_Top, hue="Product_Category",legend=False)
plt.xticks(rotation=30)
plt.show()
```



In [159…
```
# sales state by product_category Bottom 5
sales_state_Bot=df.groupby(["Product_Category"],as_index=False)["Amount"].sum().sort_values(by="Amount",ascendi
```

In [164…
```
plt.figure(figsize=(9,4))
sns.barplot(x="Product_Category", y="Amount", data=sales_state_Bot, hue="Product_Category",legend=False)
plt.show()
```



Coclusion:

- More female shoppers as twice 2X than men.
- Purchasing power of women are twice as that of men.
- Married women has more puechasing power than those who are single.
- 26-35 Age group has more customer base.
- Majority amount of orders and sales came from up.
- IT,Health,Aviation occupation are most spenders in terms of money and order.
- Majority of orders are from clothing and apparel product category.
- Majority of Revenue are from Food, clothing and apparel product category.

In [ ]: