```python
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# import csv file
df = pd.read_csv("Diwali Sales Data.csv",encoding="unicode_escape")

df.shape
```

```
(11251, 15)
```

```python
df.head(6)
```

```
   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


              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

    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
```

```python
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
```

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

#check for null values
pd.isnull(df).sum()
```

```
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
```

```
# drop null values
df.dropna(inplace=True)

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

df['Amount'].dtypes

dtype('int32')

df.columns

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

#rename column
df.rename(columns= {'Marital_Status':'Shaadi'})

       User_ID    Cust_name Product_ID Gender Age Group  Age
Shaadi  \
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]
```

```
# describe() method returns description of the data in the DataFrame
(i.e. count, mean, std, etc)
df.describe()
```

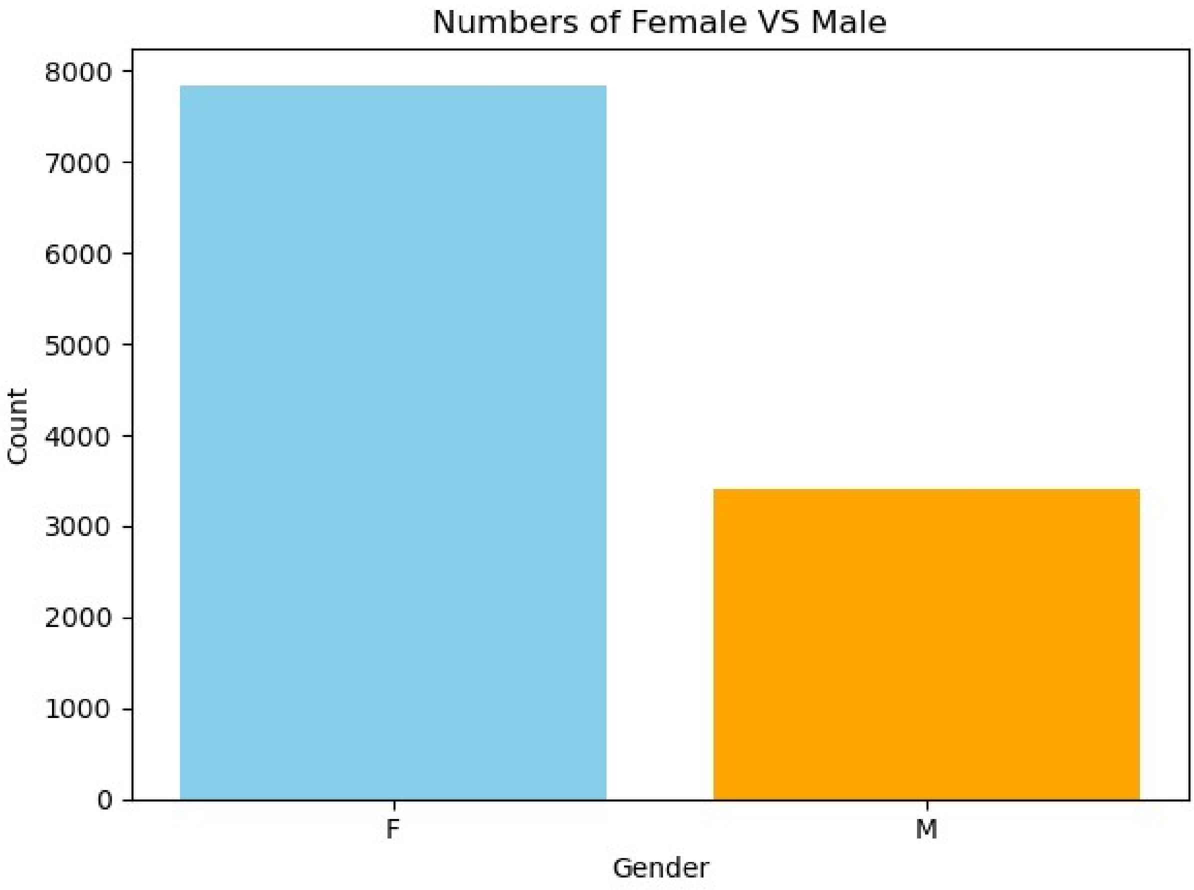
```
            User_ID           Age  Marital_Status        Orders
Amount
count  1.123900e+04  11239.000000    11239.000000  11239.000000
11239.000000
mean   1.003004e+06     35.410357        0.420055      2.489634
9453.610553
std    1.716039e+03     12.753866        0.493589      1.114967
5222.355168
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

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

                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
```

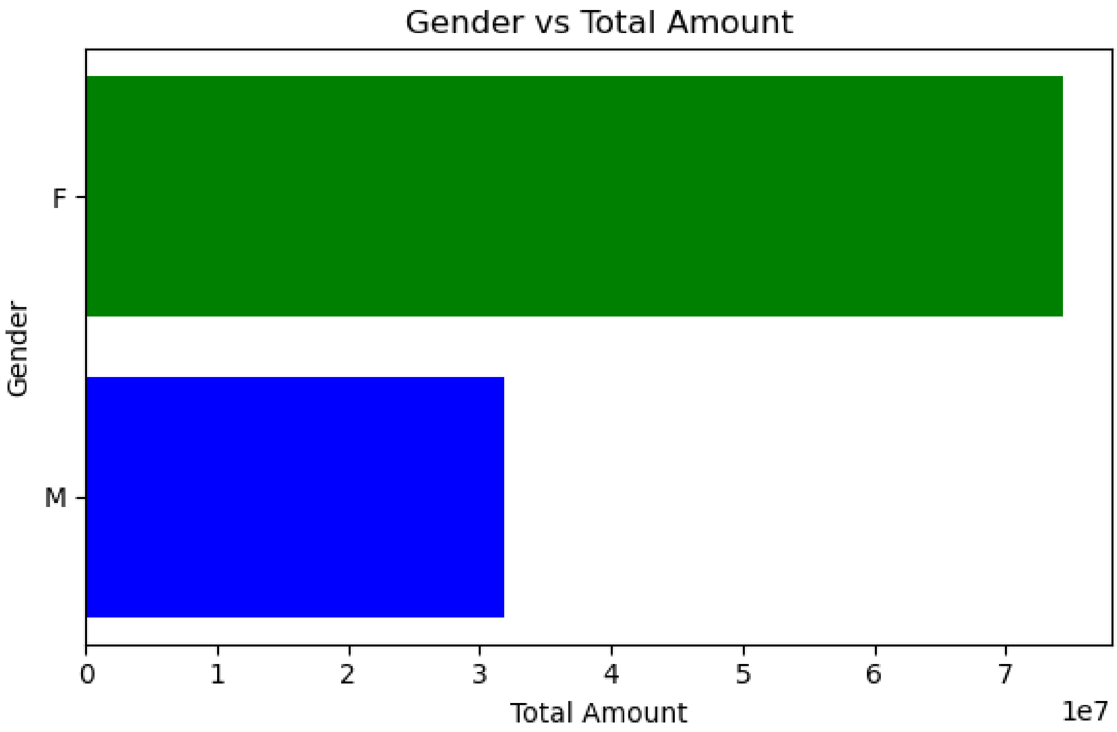
Exploratory Data Analysis

Gender

```
# plotting a bar chart for Gender and it's count
gender_counts=df["Gender"].value_counts()
plt.figure()
plt.bar(gender_counts.index,gender_counts.values,color=["skyblue","ora
nge"])
plt.title("Numbers of Female VS Male")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.tight_layout()

plt.show()
```



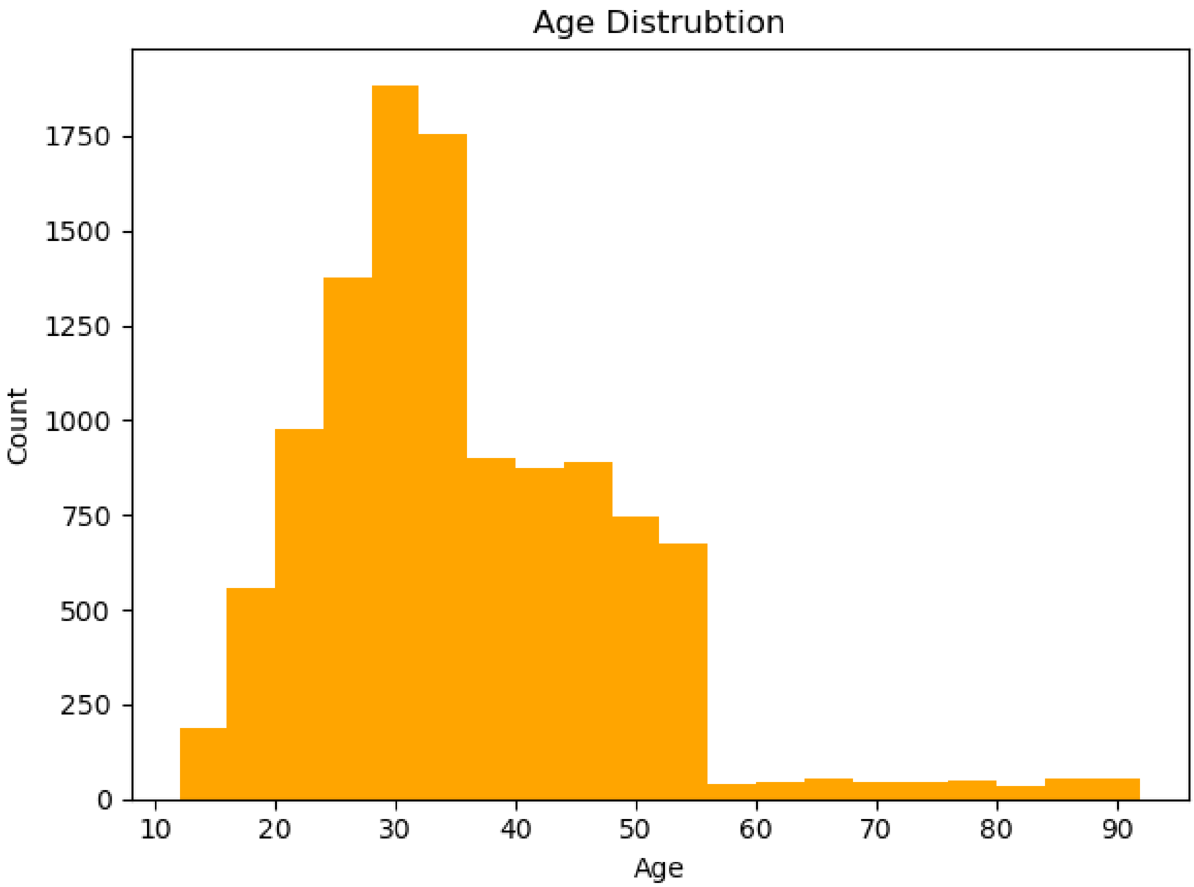
```
# plotting a bar chart for gender vs total amount
# Grouping
gender_amount = df.groupby("Gender")
["Amount"].sum().sort_values(ascending=True)
# Plot
plt.figure(figsize=(6,4))
plt.barh(gender_amount.index, gender_amount.values,
color=["blue","green"])
plt.title("Gender vs Total Amount")
plt.xlabel("Total Amount")   # X-axis = Amount
plt.ylabel("Gender")         # Y-axis = Gender
plt.tight_layout()
plt.show()
```



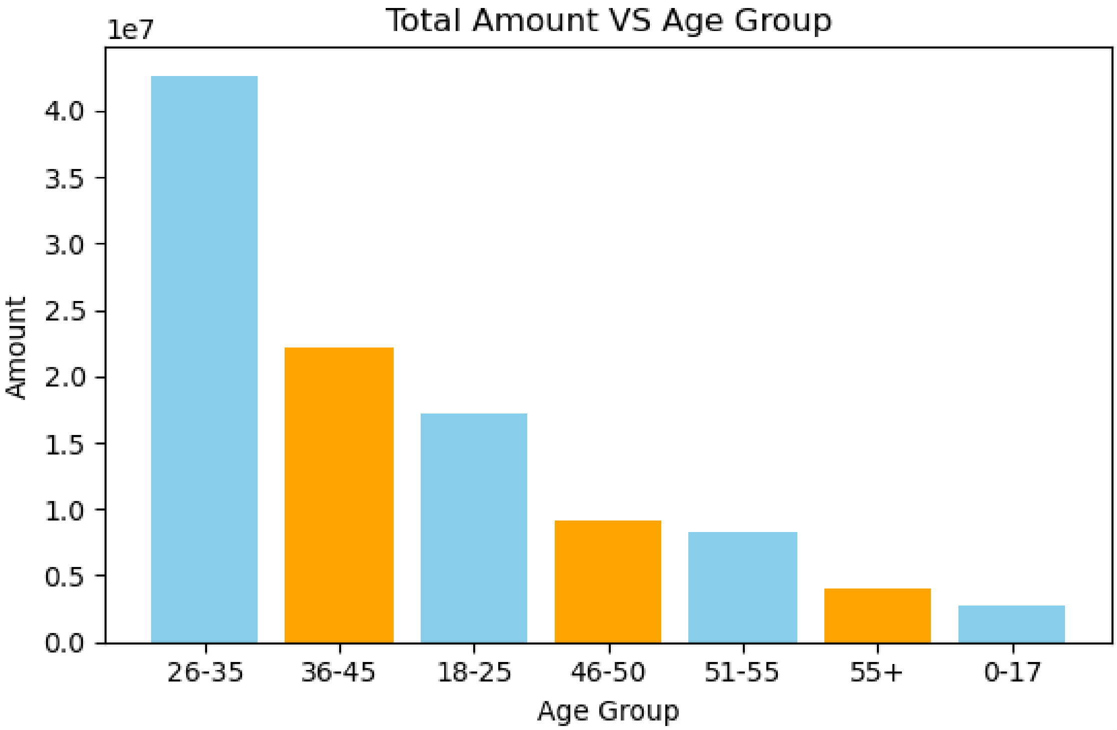
From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

Age

```
plt.figure()
plt.hist(df["Age"],bins=20,color=["orange"])
plt.xlabel("Age")
plt.ylabel("Count")
plt.title("Age Distrubtion")
plt.tight_layout()
plt.show()
```



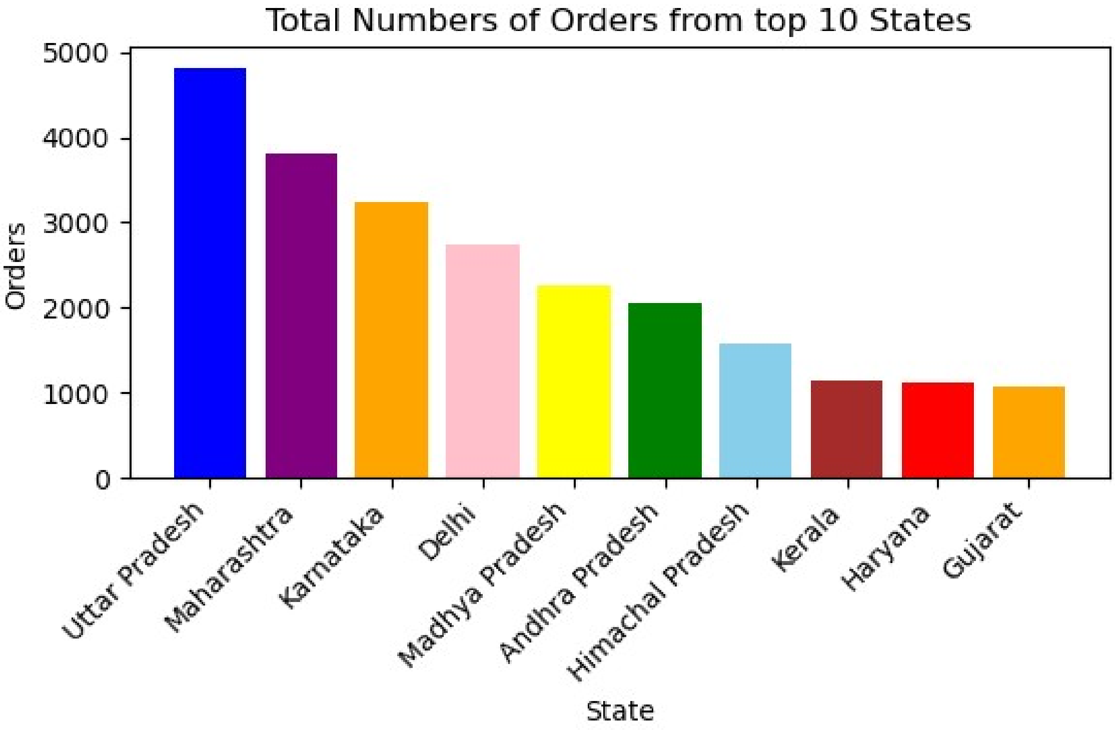
```
# Total Amount vs Age Group
sales_age = df.groupby("Age Group")
['Amount'].sum().sort_values(ascending=False)
plt.figure(figsize=(6,4))
plt.bar(sales_age.index,sales_age.values, color=["skyblue","orange"])
plt.title("Total Amount VS Age Group")
plt.xlabel("Age Group")   # X-axis = Amount
plt.ylabel("Amount")         # Y-axis = Gender
plt.tight_layout()
plt.show()
```



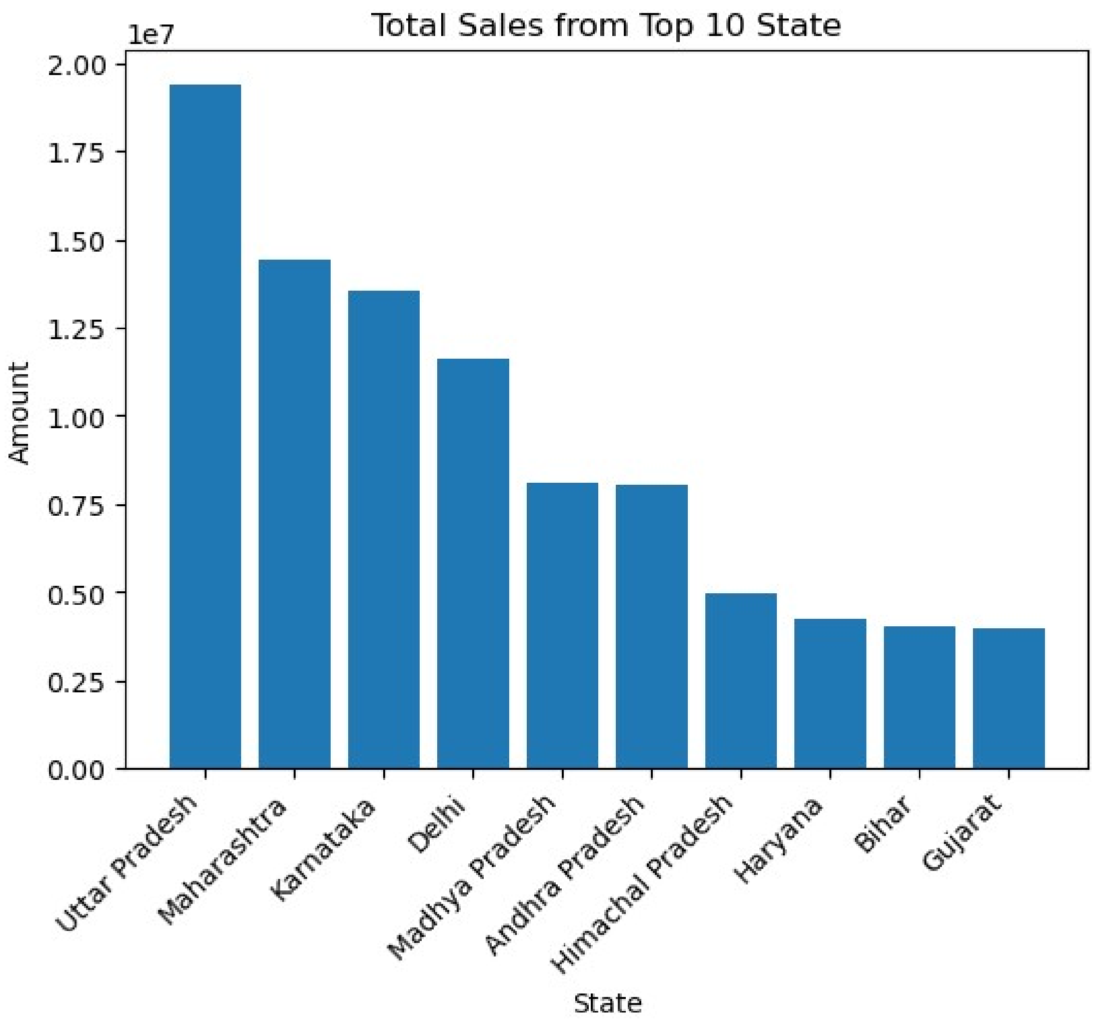
From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

State

```
# total number of orders from top 10 states
sales_state = df.groupby('State')
['Orders'].sum().sort_values(ascending=False).head(10)
plt.figure(figsize=(6,4))
plt.bar(sales_state.index,sales_state.values,color=["blue","purple","orange","pink","yellow","green","skyblue","brown","red","orange"])
plt.title("Total Numbers of Orders from top 10 States")
plt.xlabel("State")   # X-axis = Amount
plt.ylabel("Orders")
plt.xticks(rotation=45, ha="right") # Y-axis = Gender
plt.tight_layout()
plt.show()
```



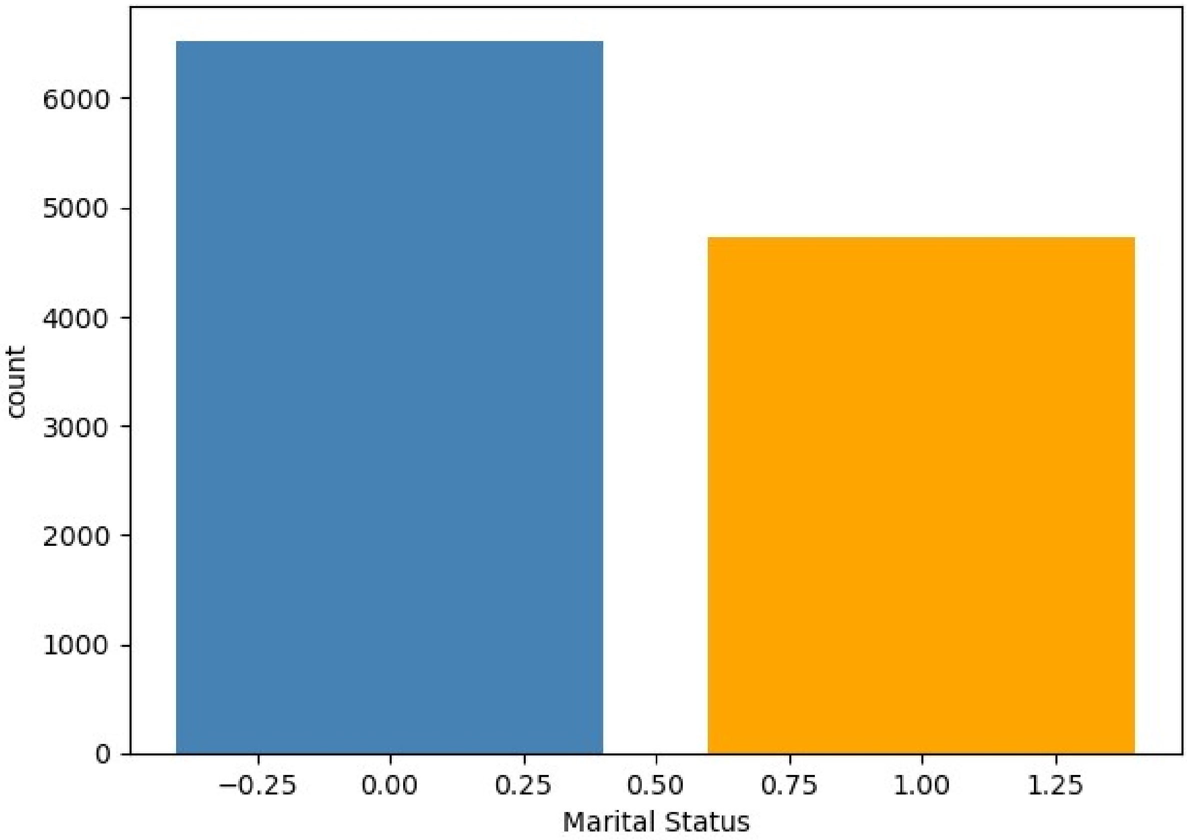
```
# total amount/sales from top 10 states
sales_state=df.groupby("State")
["Amount"].sum().sort_values(ascending=False).head(10)
plt.bar(sales_state.index,sales_state.values,)
plt.title("Total Sales from Top 10 State")
plt.xlabel("State")
plt.ylabel("Amount")
plt.xticks(rotation=45, ha="right")   # rotate labels by 45° and align
right
plt.show()
```



From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

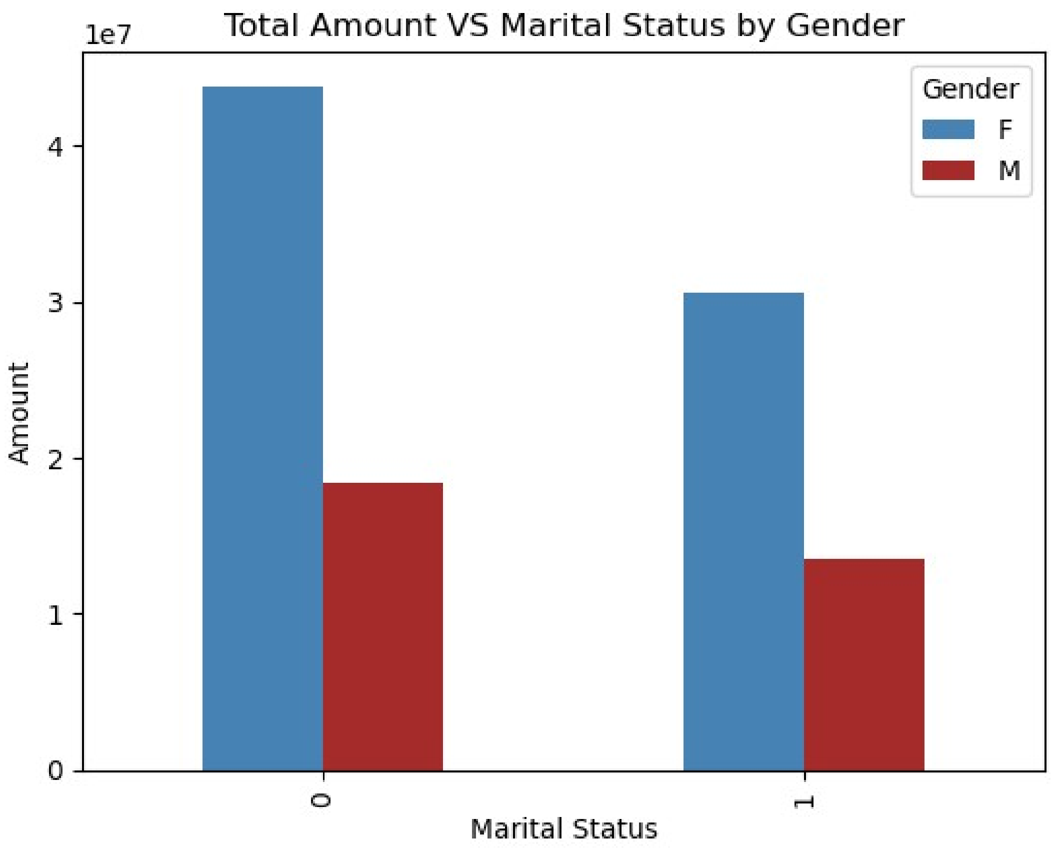
Marital Status

```
mar_status=df["Marital_Status"].value_counts().sort_values(ascending=False)
plt.figure(figsize=(7,5))
plt.bar(mar_status.index,mar_status.values,color=["steelblue","orange"])
plt.xlabel("Marital Status")
plt.ylabel("count")
plt.show()
```



```
sales_state = df.groupby(['Marital_Status','Gender'])
['Amount'].sum().unstack()
plt.figure(figsize=(7,5))
sales_state.plot(kind="bar",color=["steelblue","brown"])
plt.title("Total Amount VS Marital Status by Gender")
plt.ylabel("Amount")
plt.xlabel("Marital Status")
plt.show()
```

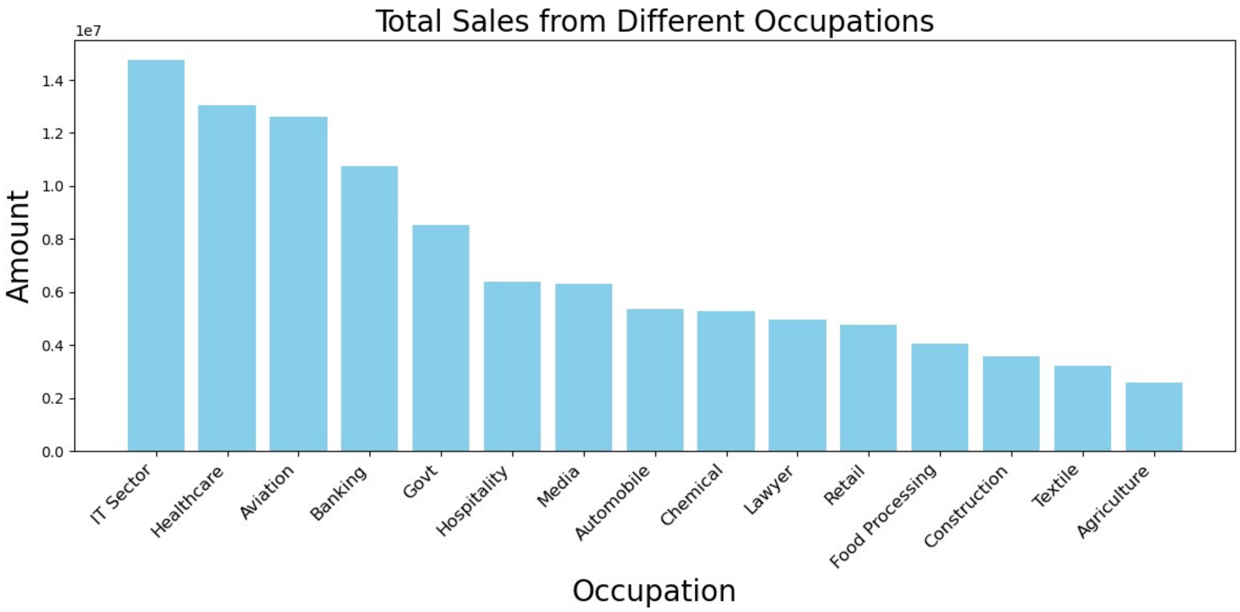
```
<Figure size 700x500 with 0 Axes>
```



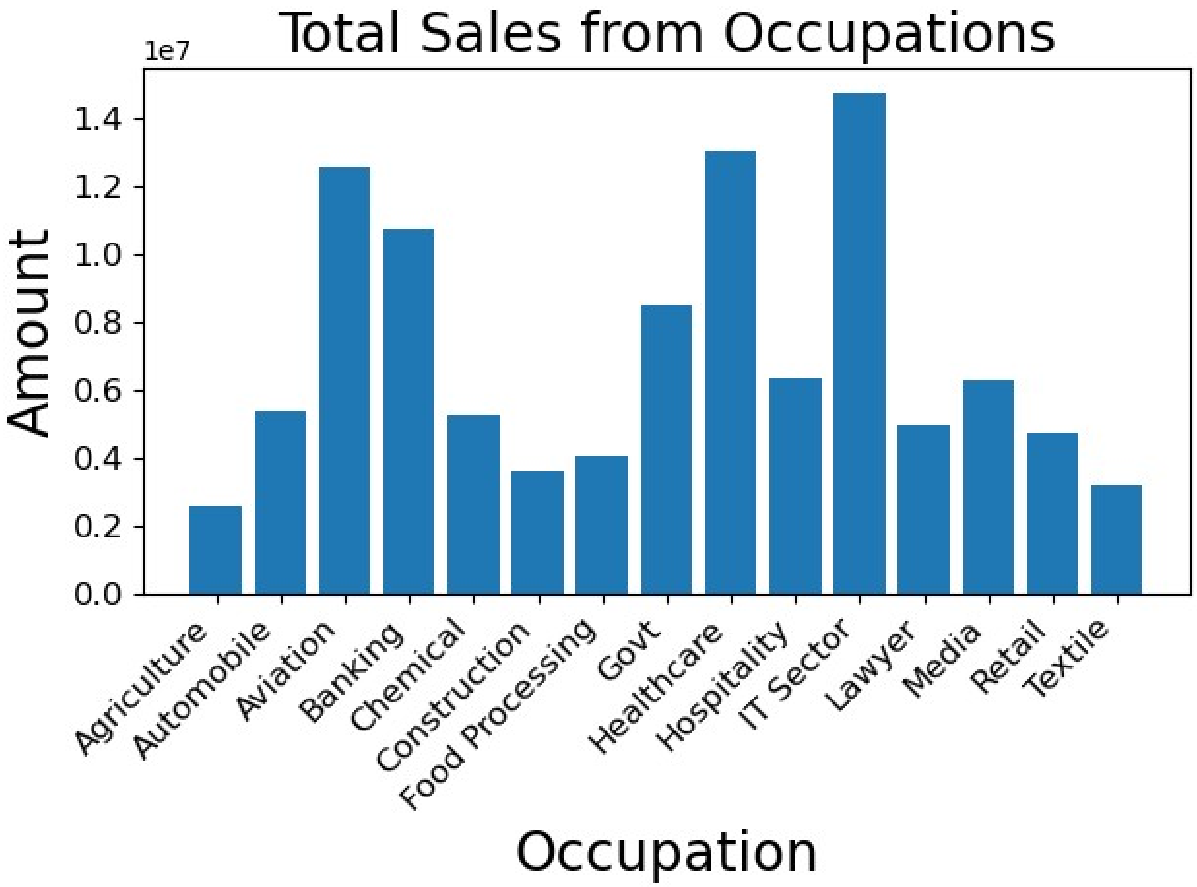
From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

Occupation

```
sales_state = df.groupby("Occupation")
["Amount"].sum().sort_values(ascending=False)

plt.figure(figsize=(12,6))
plt.bar(sales_state.index, sales_state.values, color="skyblue")
plt.title("Total Sales from Different Occupations",fontsize=20)
plt.xlabel("Occupation",fontsize=20)
plt.ylabel("Amount",fontsize=20)
plt.xticks(rotation=45, ha="right",fontsize=12)
plt.xticks(rotation=45, ha="right",fontsize=12)
plt.tight_layout()
plt.show()
```



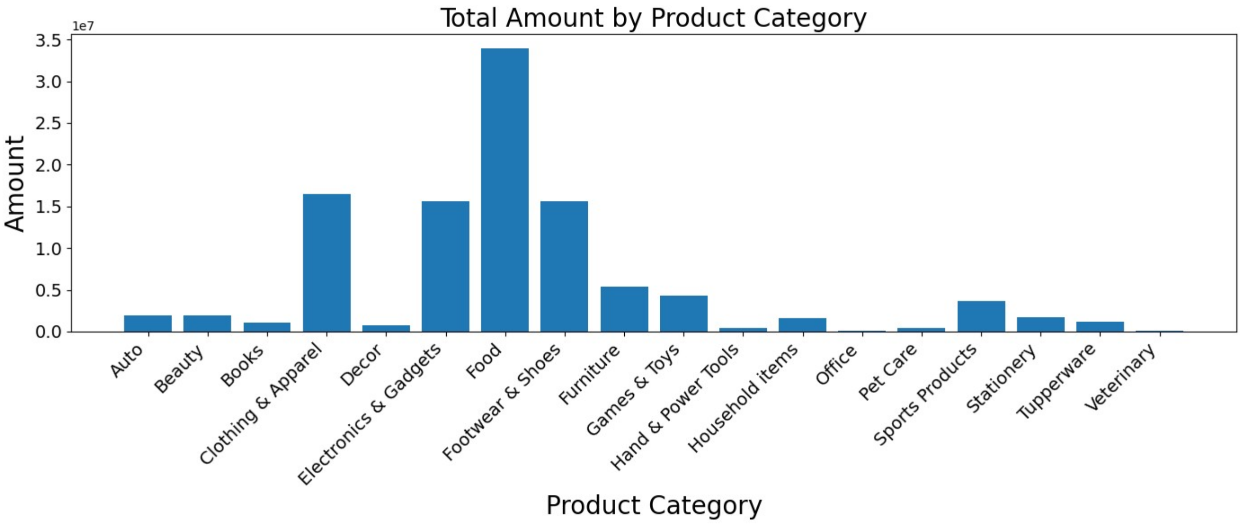
```python
sales_state = df.groupby("Occupation")["Amount"].sum()
plt.bar(sales_state.index, sales_state.values)
plt.title("Total Sales from Occupations", fontsize=20)
plt.xlabel("Occupation", fontsize=20)
plt.ylabel("Amount", fontsize=20)
plt.xticks(rotation=45, ha="right", fontsize=12)
plt.yticks(fontsize=12)   # change 12 to bigger value
plt.tight_layout()
plt.show()
```



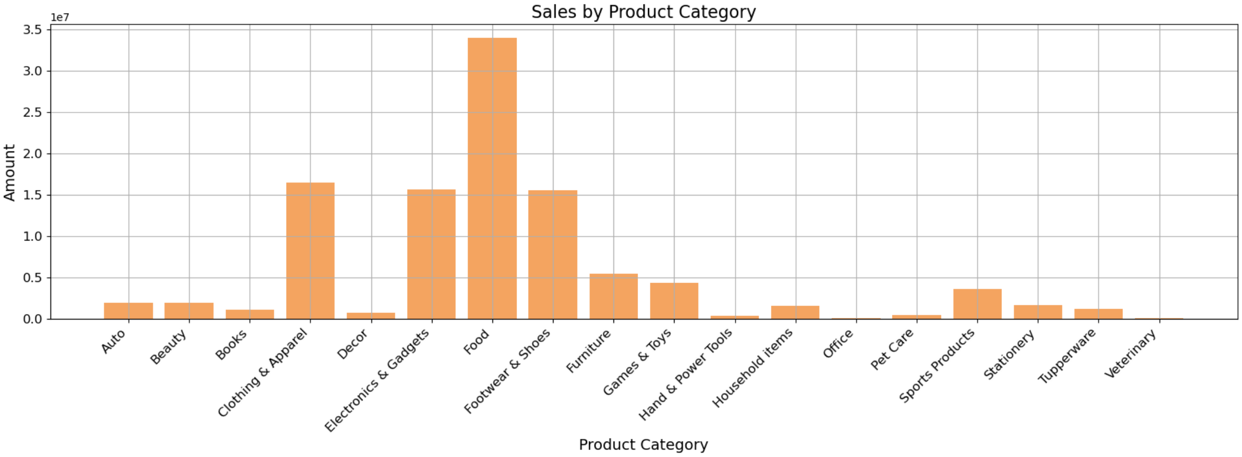
From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

Product Category

```
sales_state = df.groupby('Product_Category')['Amount'].sum()
plt.figure(figsize=(14,6))
plt.bar(sales_state.index,sales_state.values)
plt.title("Total Amount by Product Category", fontsize=20)
plt.xlabel("Product Category", fontsize=20)
plt.ylabel("Amount", fontsize=20)
plt.xticks(rotation=45, ha="right", fontsize=14)
plt.yticks(fontsize=14)
plt.tight_layout()
plt.show()
```

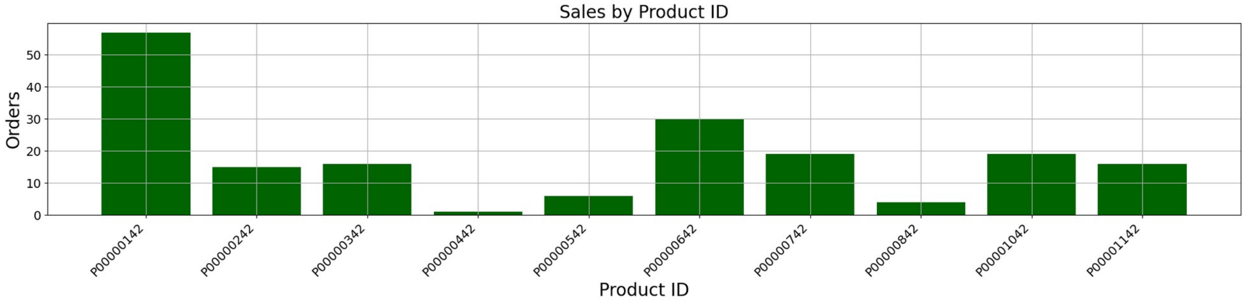


```
sales_state = df.groupby('Product_Category')['Amount'].sum()
plt.figure(figsize=(20,5))
plt.bar(sales_state.index,sales_state.values,color="sandybrown")
plt.xlabel('Product Category', fontsize=14)
plt.ylabel('Amount', fontsize=14)
plt.title('Sales by Product Category', fontsize=16)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.yticks(fontsize=12)
plt.grid()
plt.show()
```

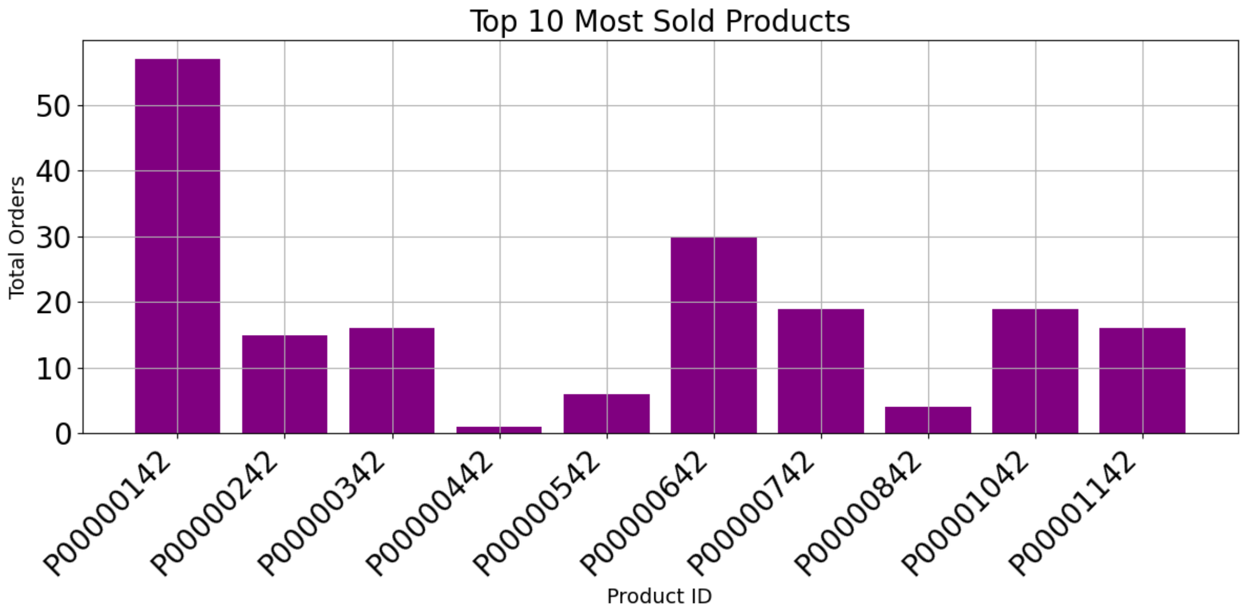


From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

```
sales_state = df.groupby('Product_ID')['Orders'].sum().head(10)
plt.figure(figsize=(20,5))
plt.bar(sales_state.index, sales_state.values, color="darkgreen")
plt.xlabel('Product ID', fontsize=20)
plt.ylabel('Orders', fontsize=20)
plt.title('Sales by Product ID', fontsize=20)
plt.xticks(rotation=45, ha='right', fontsize=14)
plt.yticks(fontsize=14)
plt.tight_layout()
plt.grid()
plt.show()
```



```
# top 10 most sold products (same thing as above)

top_products =df.groupby('Product_ID')['Orders'].sum().head(10)
plt.figure(figsize=(12,6))
plt.bar(top_products.index, top_products.values, color="purple")
plt.xlabel('Product ID', fontsize=14)
plt.ylabel('Total Orders', fontsize=14)
plt.title('Top 10 Most Sold Products', fontsize=20)
plt.xticks(rotation=45, ha='right', fontsize=20)
plt.yticks(fontsize=20)
plt.grid()
plt.tight_layout()
plt.show()
```



Conclusion:

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