Diwali Data Analysis Project

Data Cleaning

importing libraries

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

importing data through pandas library

In [4]:
```
df = pd.read_csv("Diwali Sales Data.csv" , encoding = "unicode_escape")
```

In [3]:
```
df.shape
```

Out[3]:
```
(11251, 15)
```

In [5]:
```
df.head(10)
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	2
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	2
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	2
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	2
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	2
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	2
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	2
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	2
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	2

In [6]:
```
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 [7]:
```
df.drop(["Status" , "unnamed1"] , axis = 1 , inplace = True)
```

In [8]:
```
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
```

In [9]:
```
pd.isnull(df)
```

Out[9]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
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	False	False	False	False	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

Checking for null values in data

In [10]:
```
pd.isnull(df).sum()
```

Out[10]:
```
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
```

Deleting null values from data

In [14]:
```
df.dropna(inplace = True )
```

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

Out[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              0
dtype: int64
```

In [16]:
```
df['Amount'] = df['Amount'].astype('int')
```

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

Out[17]:
```
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               int32
dtype: object
```

In [18]:
```
df.columns
```

Out[18]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [19]:
```
df.describe()
```

Out[19]:

	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

In [20]:
```
df[["Age" , "Orders" , "Amount"]].describe()
```

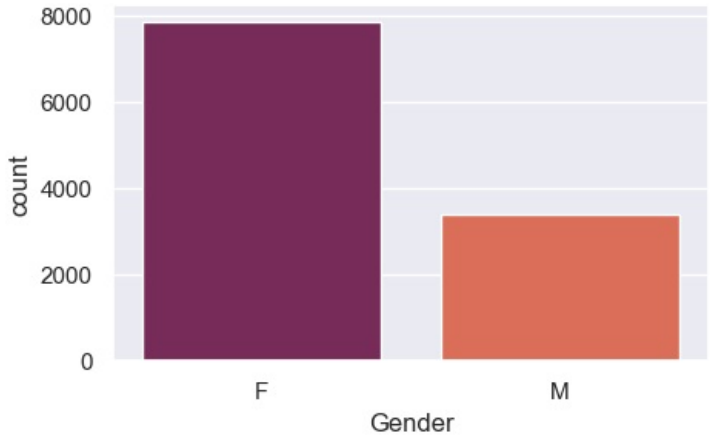
Out[20]:

	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

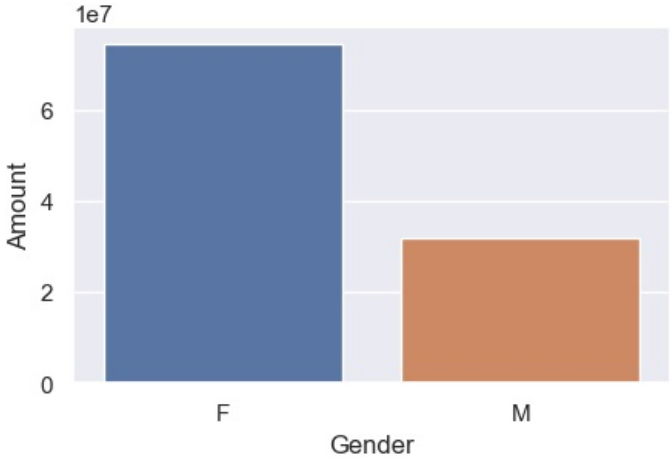
Exploratory Data Analysis

Gender

In [22]:
```
x = sns.countplot(x  = "Gender" , data = df , palette = "rocket")
sns.set(rc = {'figure.figsize' :(6,3)})
plt.show()
```



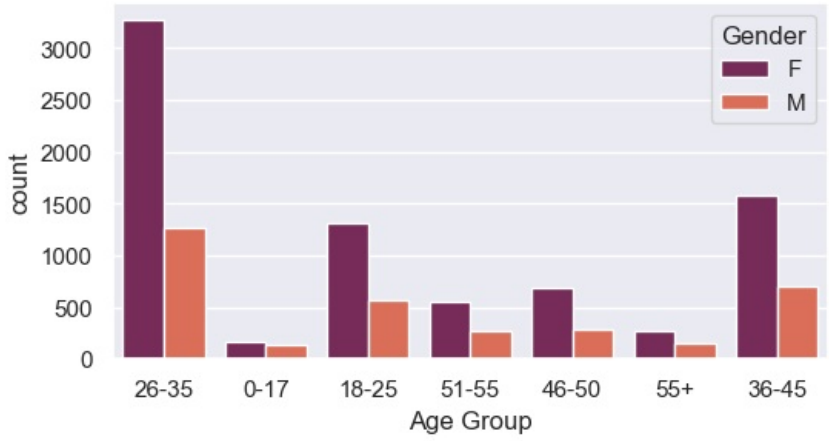
In [24]:
```
z = df.groupby(["Gender"],as_index = False)["Amount"].sum().sort_values(by = "Amount" , ascending = False)
sns.barplot(x = "Gender" , y ="Amount" , data  = z)
sns.set(rc = {'figure.figsize' :(6,3)})
plt.show()
```



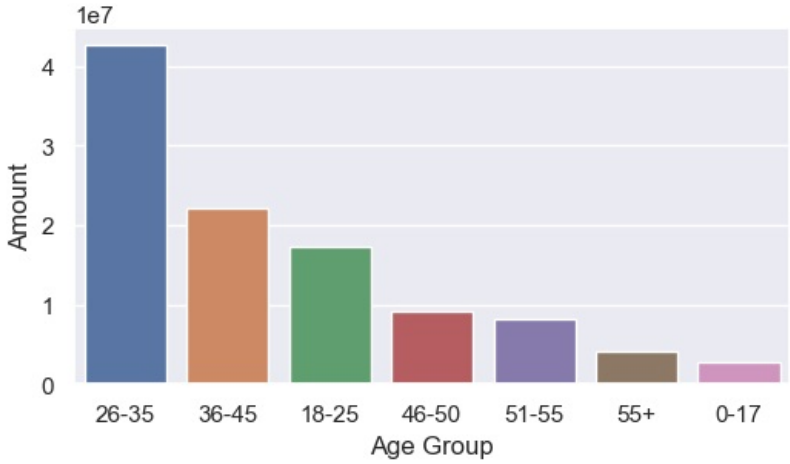
Interpretation : From the above plot we can analyse that most of the buyers are Female and the purchasing power of the female is more as compared to male

Age

In [25]:
```
y = sns.countplot(x = "Age Group" , data = df , hue = "Gender" , palette="rocket")
```



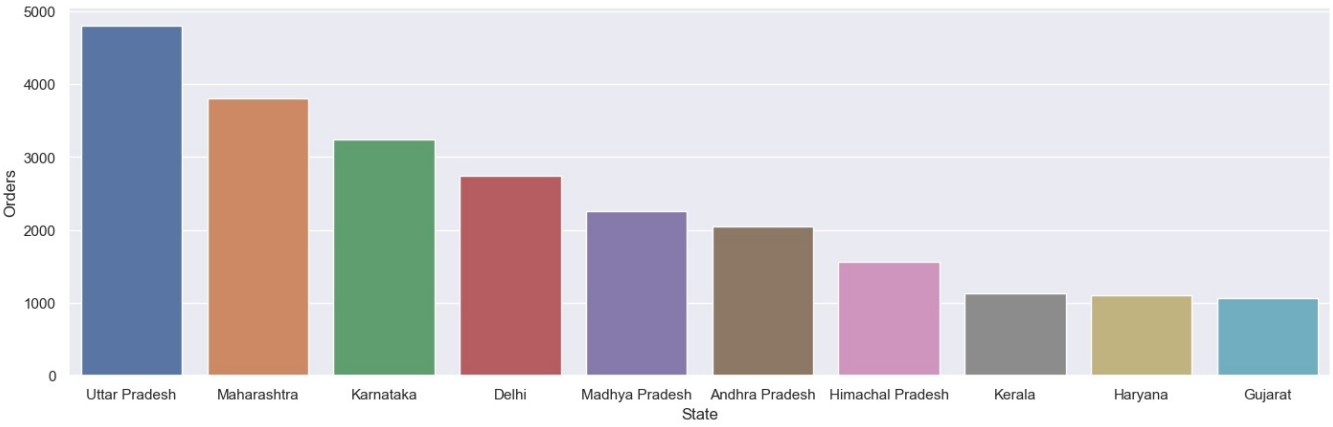
In [26]:
```
# Age group and Amount spent
sales_age = df.groupby(['Age Group'] , as_index = False )['Amount'].sum().sort_values(by = "Amount" , ascending
sns.barplot(x = "Age Group" , y ="Amount" , data  = sales_age)
plt.show()
```



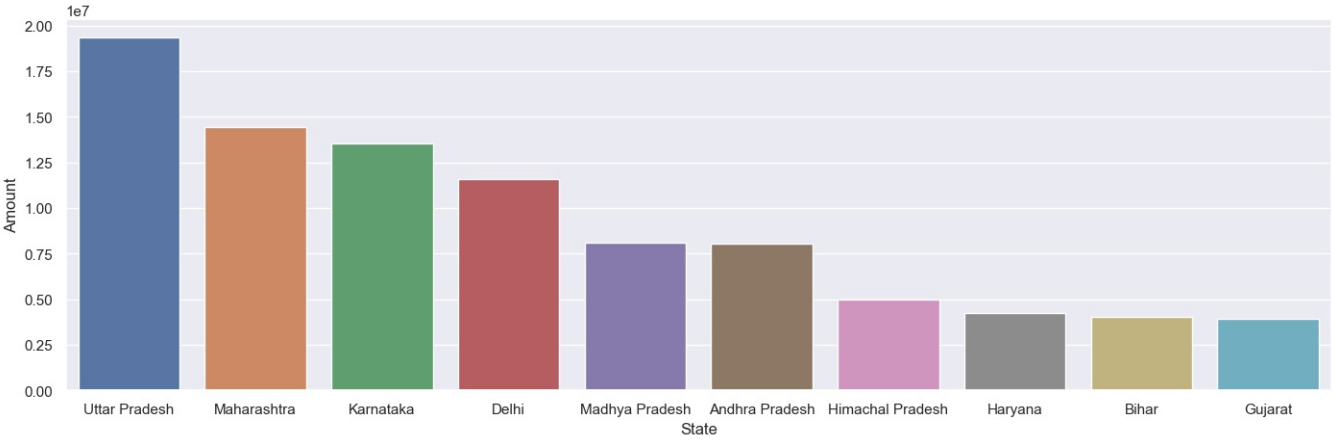
Interpretation : From the above plot we can analyse that most of the buyers are from age group 26-35 and female

State

In [37]:
```
# Total number of Orders from Top-10 States

sales_state = df.groupby(["State"] , as_index = False ) ["Orders"].sum().sort_values(by = "Orders" , ascending
sns.set(rc ={"figure.figsize":(17,5)})
sns.barplot(x = "State" , y = "Orders" , data = sales_state)
plt.show()
```



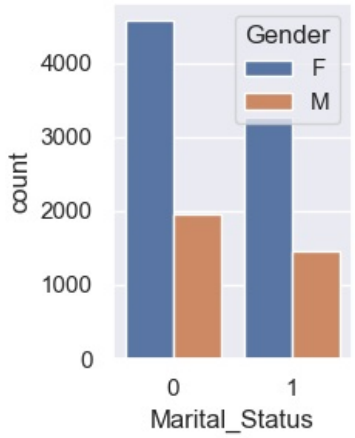
In [38]:
```
# Total Amount/Sales from Top-10 States

sales_state1 = df.groupby(['State'] , as_index = False)["Amount"].sum().sort_values(by = "Amount", ascending =
sns.set(rc = { 'figure.figsize' :(17, 5)})
sns.barplot(x = "State"  , y = "Amount" , data = sales_state1)
plt.show()
```



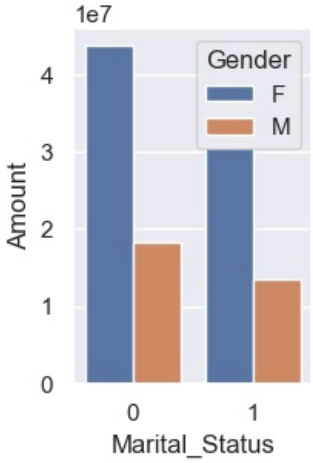
Interpretation : From above plot we can see that most of the orders as well as amount/sales are from Uttarpradesh , Maharashtra and Karnataka States

Marital Status

In [41]:
```
a = sns.countplot(x = "Marital_Status" , hue = "Gender", data = df)
sns.set(rc = {"figure.figsize" :(2,3)})
```



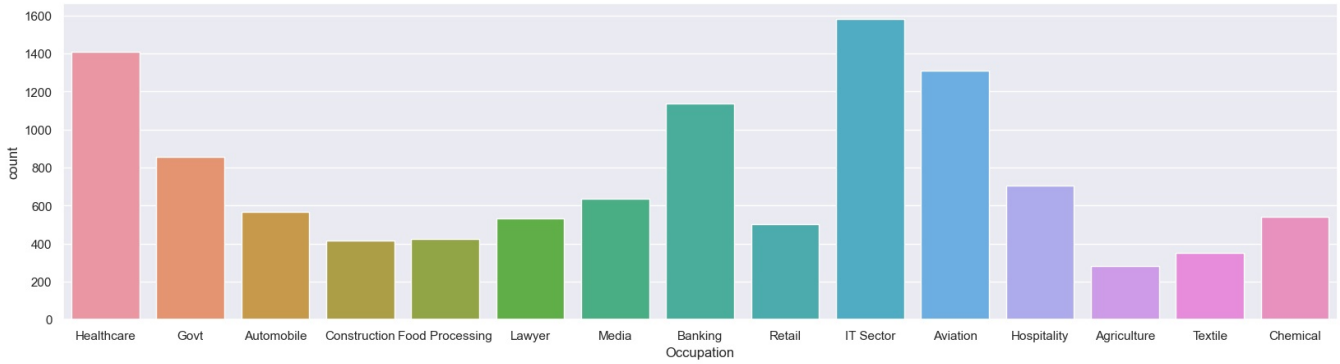
In [43]:
```
sales = df.groupby(["Marital_Status" , "Gender"] , as_index= False)["Amount"].sum().sort_values(by = "Amount" ,
sns.barplot(x = "Marital_Status" , y = "Amount" , hue = "Gender", data = sales)
plt.show()
```



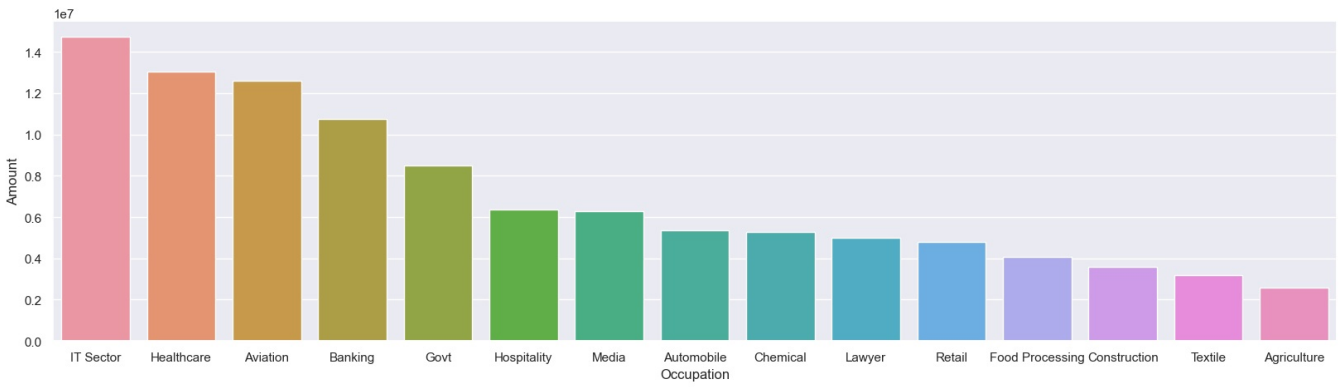
Interpretation : From above graph we can analyse that most of the buyers are Married(Female) and they have high purchasing power

Occupation

In [45]:
```
a = sns.countplot(x = "Occupation" , data = df)
sns.set(rc = {"figure.figsize" :(20,5)})
```



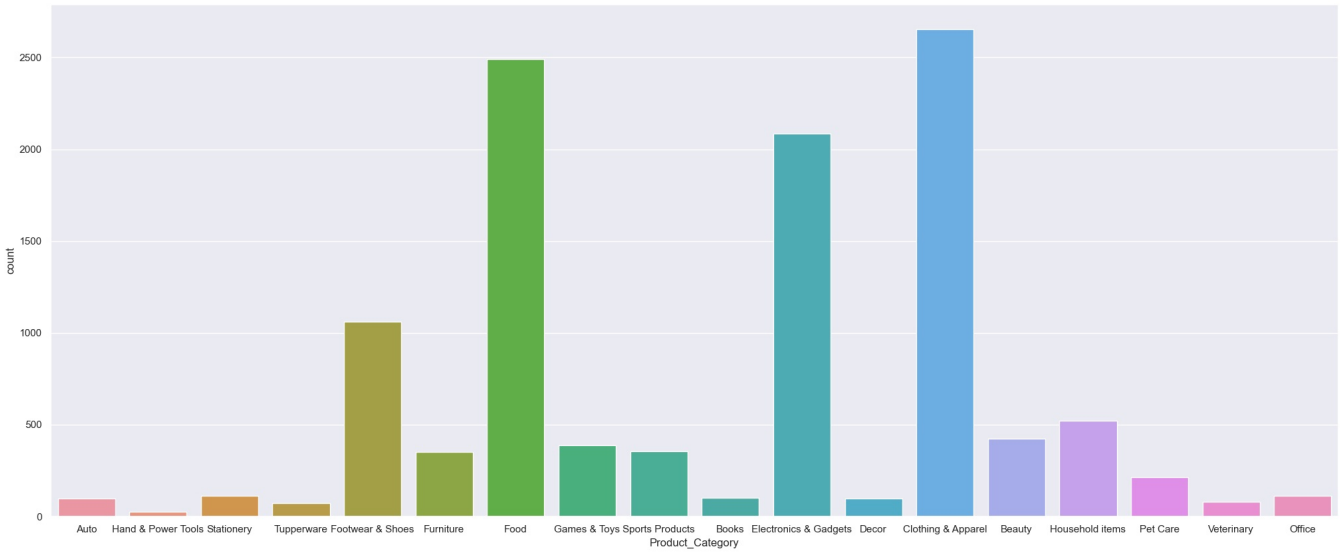
In [46]:
```
sales = df.groupby(["Occupation"] , as_index = False)["Amount"].sum().sort_values("Amount" , ascending = False)
sns.barplot(x = "Occupation"  , y = "Amount"  , data = sales )
plt.show()
```



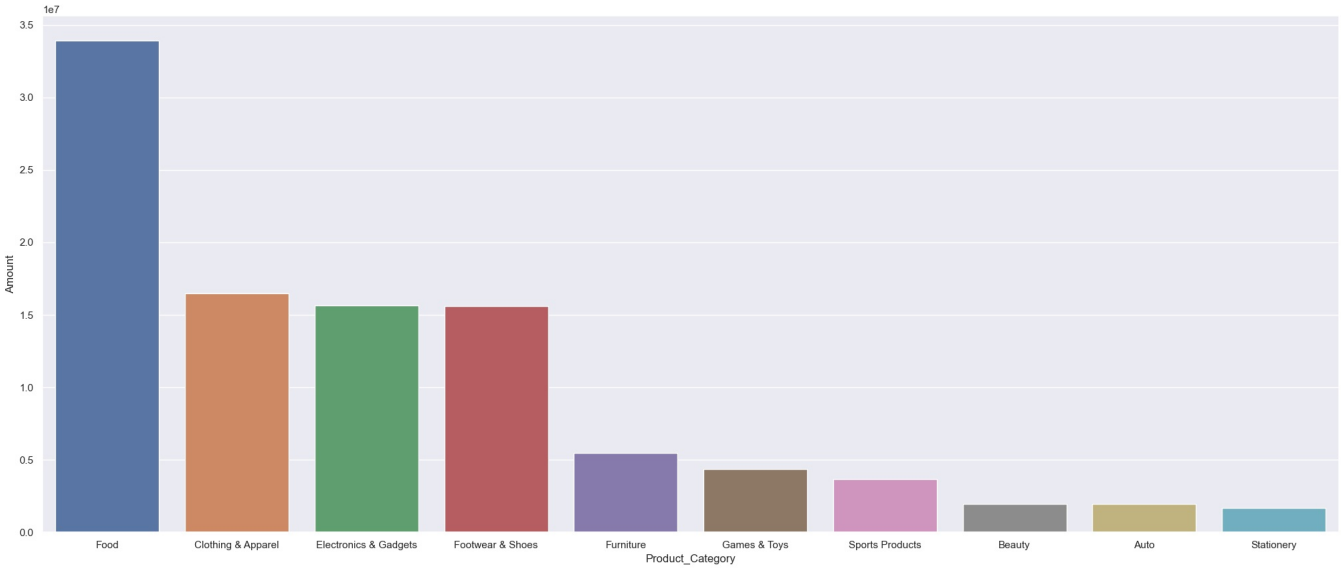
Interpretaion : From the above graphs we can analyse that most of the buyers are working in IT , Healthcare , Aviation sector

Product Category

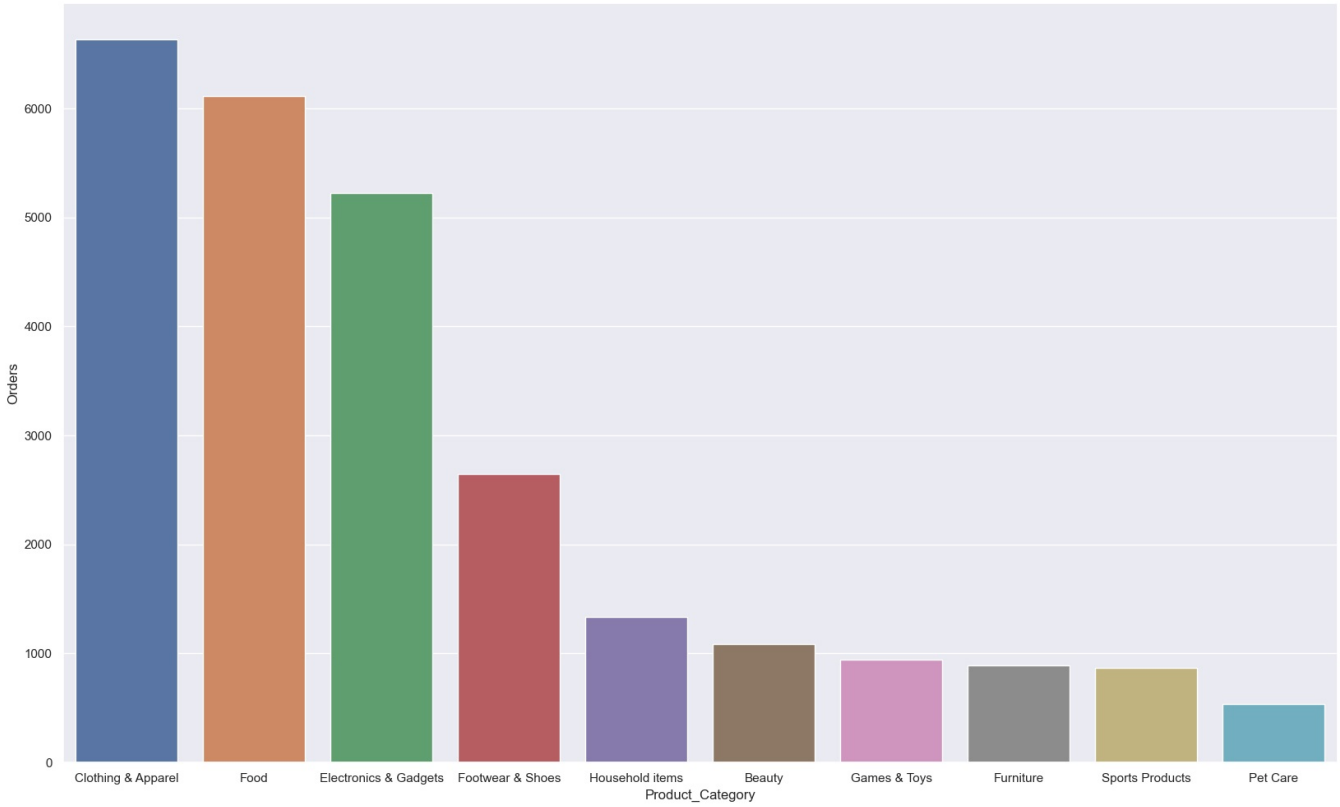
In [52]:
```
a = sns.countplot(x = "Product_Category", data = df)
sns.set(rc = {'figure.figsize' :(25,10)})
plt.show()
```



In [53]:
```
sales = df.groupby(["Product_Category"] , as_index=False)["Amount"].sum().sort_values(by = "Amount" , ascending
sns.barplot(x = "Product_Category" , y = "Amount" , data = sales)
sns.set(rc = {'figure.figsize' :(20,12)})
plt.show()
```



In [54]:
```
sales = df.groupby(["Product_Category"] , as_index = False)["Orders"].sum().sort_values("Orders" , ascending =
sns.barplot(x = "Product_Category" , y = "Orders" , data = sales )
plt.show()
```



Interpretation : From the above graphs we can analyse that most orders are from Clothing & Apparel and amount/sales are from food category

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

In [ ]: