Diwali Sales Analysis Using Python

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

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

In [5]:
```
df
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Ord
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	
...	...	...	...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	

11251 rows × 15 columns

In [50]:
```
df.shape
```

Out[50]:
```
(11251, 15)
```

In [51]:
```
df.head(10)
```

Out[51]:

	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
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4
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
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4

In [52]:
```
df.tail(5)
```

Out[52]:

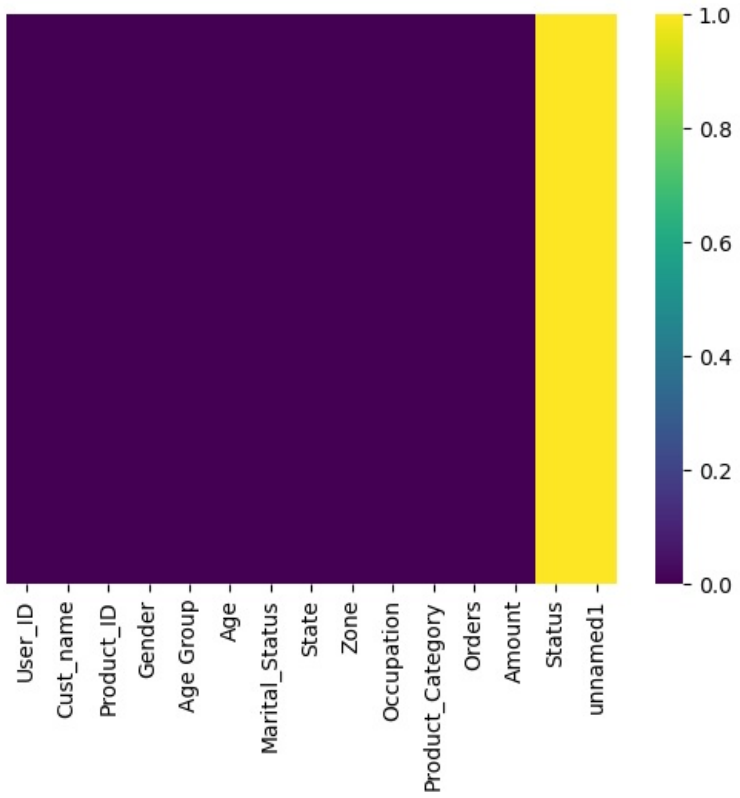
	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3

In [54]:
```
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                 12
Status              11251
unnamed1            11251
dtype: int64
```

In [55]:
```
sns.heatmap(df.isnull(),yticklabels=False,cmap ='viridis')
```

Out[55]:
```
<Axes: >
```



Here in above heatmap we can clearly observed the missing value in most in status and unnamed1 column.It is not necessesry for eda,So we can drop this column.

In [56]:
```
df.drop(['Status','unnamed1'],axis = 1 ,inplace = True)
```

In [57]:
```
df
```

Out[57]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Ord
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	
...	...	...	...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	

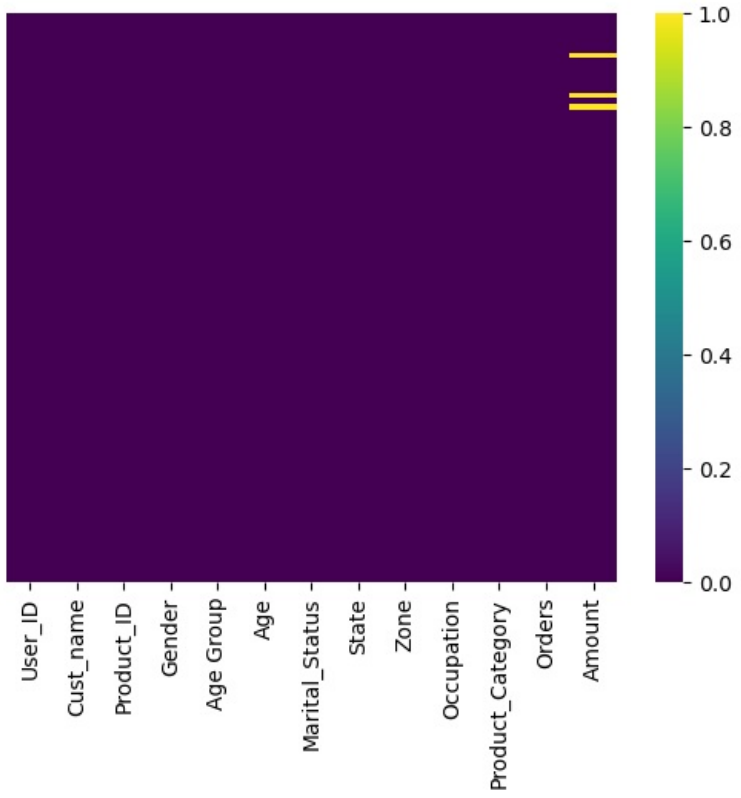
11251 rows × 13 columns

In [58]:
```
df.isnull().sum()
```

Out[58]:
```
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
```

In [59]:
```
sns.heatmap(df.isnull()[:100],yticklabels=False,cmap ='viridis')
```

Out[59]:
```
<Axes: >
```



In [60]:
```
df.dropna(inplace =True)
```

In [61]:
```
# Check Datatype of given data
df.isnull().sum()
```

Out[61]:
```
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 [63]:
```
df.dtypes
```

Out[63]:
```
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             float64
dtype: object
```

we can clearly observed the Amount datatype is float so we can convert into int

In [64]:
```
df['Amount']=df['Amount'].astype('int')
```

In [65]:
```
df.dtypes
```

Out[65]:
```
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 [66]:
```
list[df.columns]
```

Out[66]:
```
list[Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')]
```

In [6]:
```
df.describe()
```

Out[6]:

	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

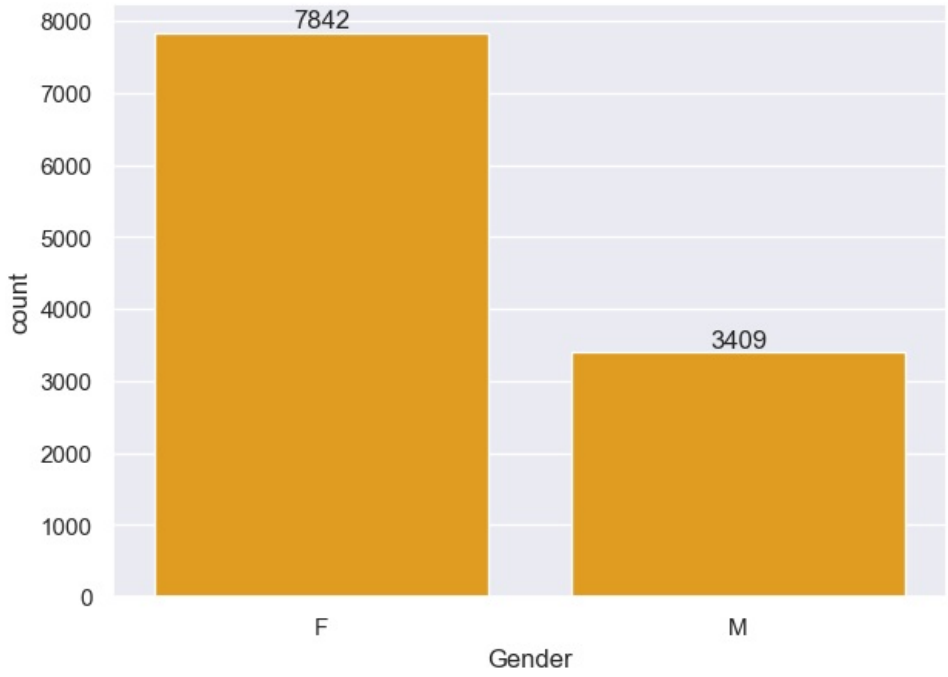
In [7]:
```
df[['Age','Orders','Amount']].describe()
```

Out[7]:

	Age	Orders	Amount
count	11251.000000	11251.000000	11239.000000
mean	35.421207	2.489290	9453.610858
std	12.754122	1.115047	5222.355869
min	12.000000	1.000000	188.000000
25%	27.000000	1.500000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

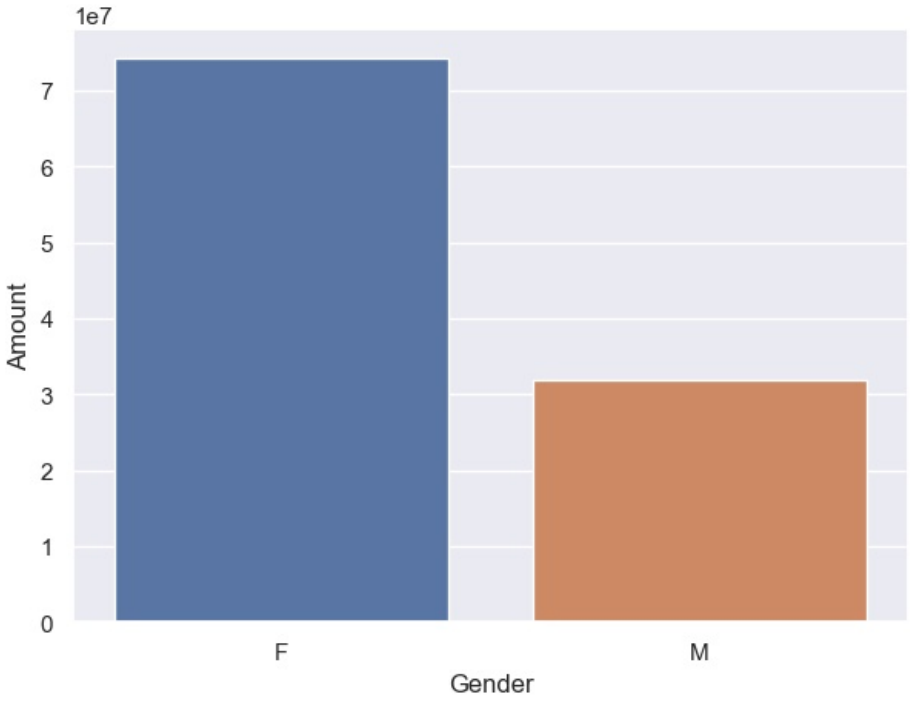
Exploratory Data Analysis

In [114…
```
ax = sns.countplot(x='Gender',color = 'Orange',data = df)
for bars in ax.containers:
 ax.bar_label(bars)
```



In [113…
```
sales_gender = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending= False)
sns.barplot(x='Gender',y='Amount' ,data = sales_data)
```

Out[113]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



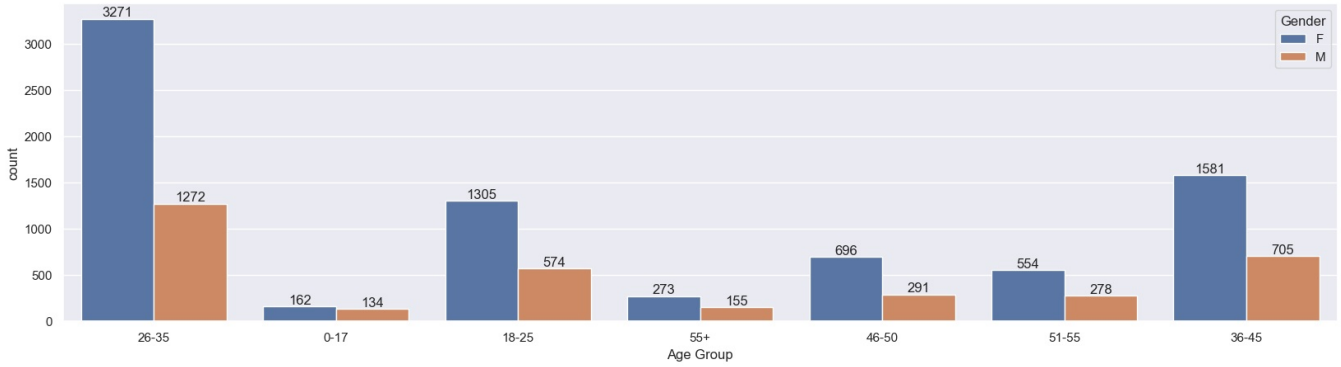
From above graphs we can see that most of the buyers are females and even the purchasing ratio of females are greater than men

Age

In [97]:
```
df.columns
```

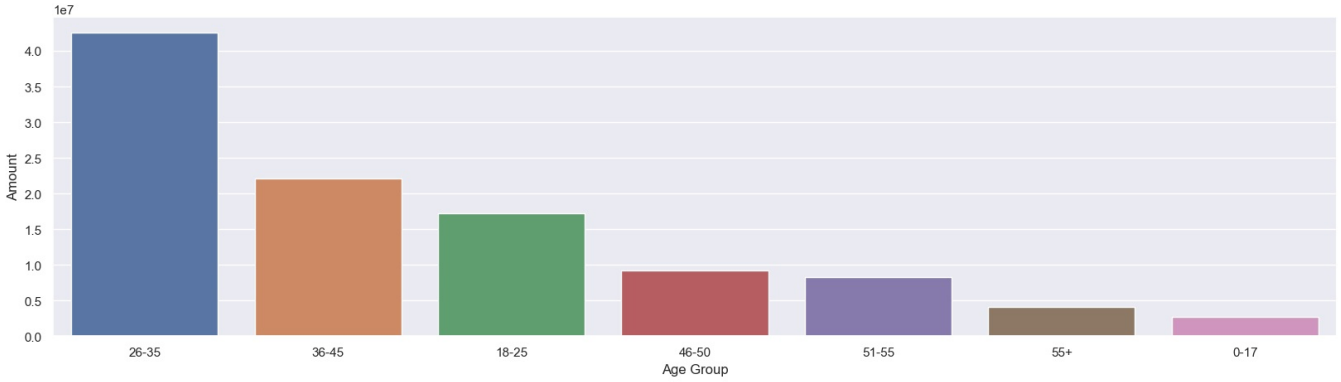
Out[97]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

In [98]:
```
ax = sns.countplot(data=df,x='Age Group',hue = 'Gender')
for bars in ax.containers:
 ax.bar_label(bars)
```



In [99]:
```
# Total Amount vs Age Group
sales_age = df.groupby(['Age Group'],as_index = False)['Amount'].sum().sort_values(by = 'Amount',ascending = Fa
sns.barplot(x='Age Group',y = 'Amount',data=sales_age)
```

Out[99]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



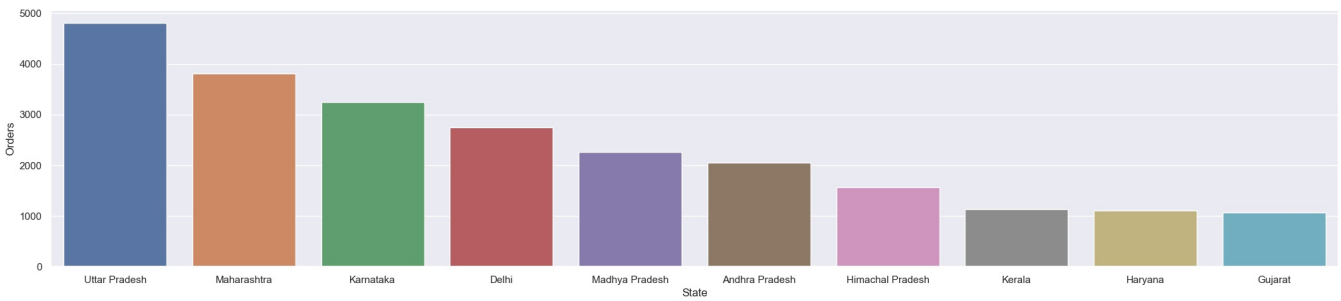
From the above graphs we can see that most of the buyers are of group between 26-35 yrs female

State

Total number of orders from top 10 States

In [100…
```
# Total no of oredrs from top 10 states
sales_state =df.groupby(['State'],as_index= False)['Orders'].sum().sort_values(by =  'Orders',ascending = False
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data = sales_state, x= 'State',y = 'Orders')
```

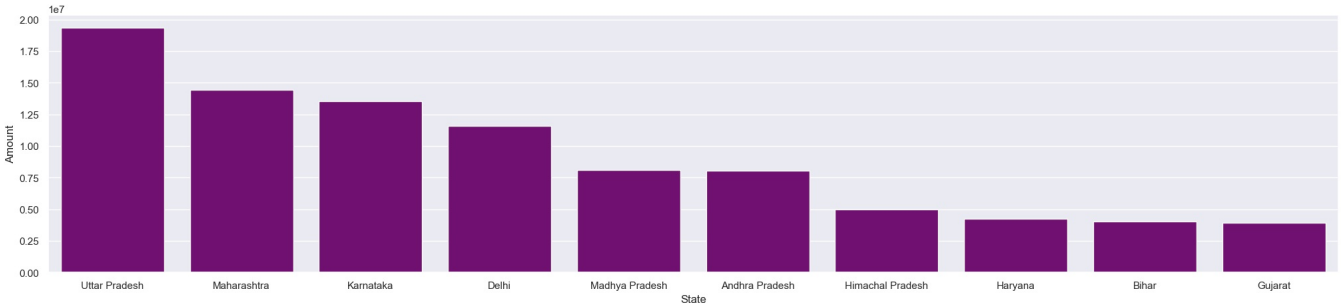
Out[100]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



Total number of Amout from top 10 States

In [101…
```
sales_state= df.groupby(['State'],as_index= False)['Amount'].sum().sort_values(by='Amount',ascending = False).h
sns.set(rc ={'figure.figsize':(25,5)})
sns.barplot(data = sales_state, x= 'State',y ='Amount', color = 'purple')
```

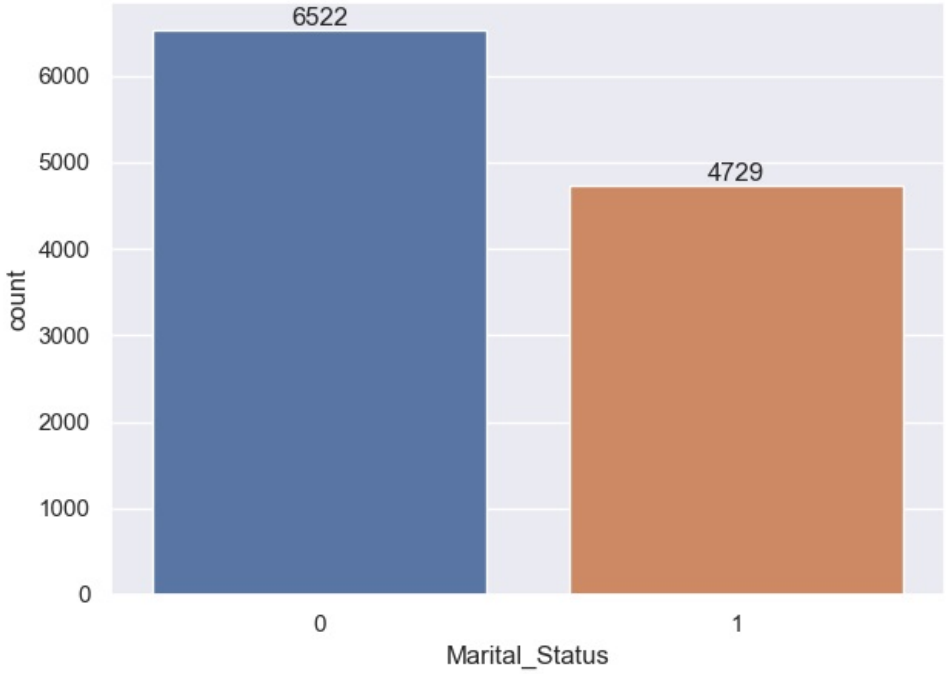
Out[101]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



from above graph we can see that most of the ordersand total amount are from Uttar Pradesh,Maharashtra and Karnataka respectively

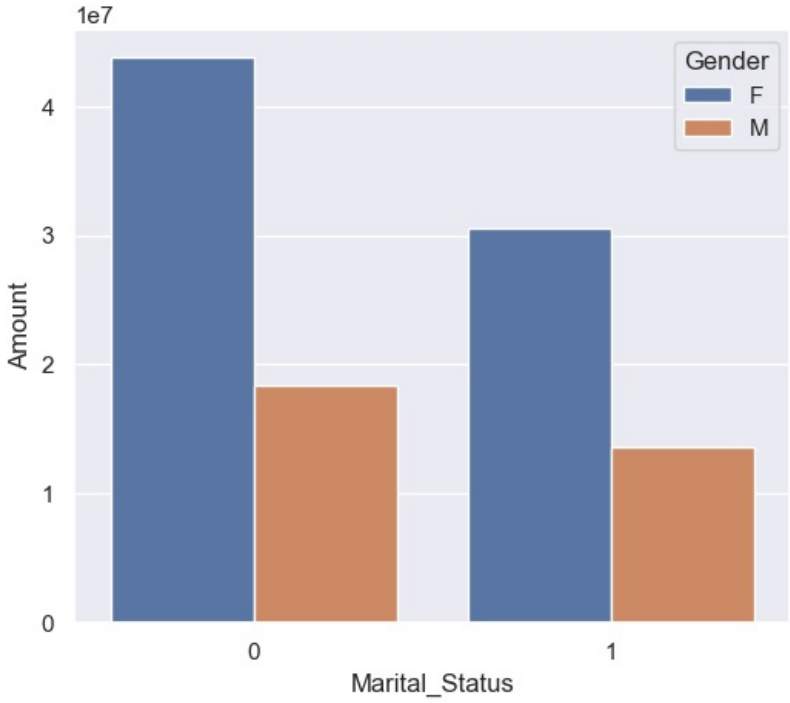
Maritial Status

In [112…
```
ax = sns.countplot(data = df,x = 'Marital_Status')
for bars in ax.containers:
 ax.bar_label(bars)
```



In [103…
```
sales_state= df.groupby(['Marital_Status','Gender'],as_index= False)['Amount'].sum().sort_values(by='Amount',as
sns.set(rc ={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x= 'Marital_Status',y ='Amount', hue ='Gender')
```

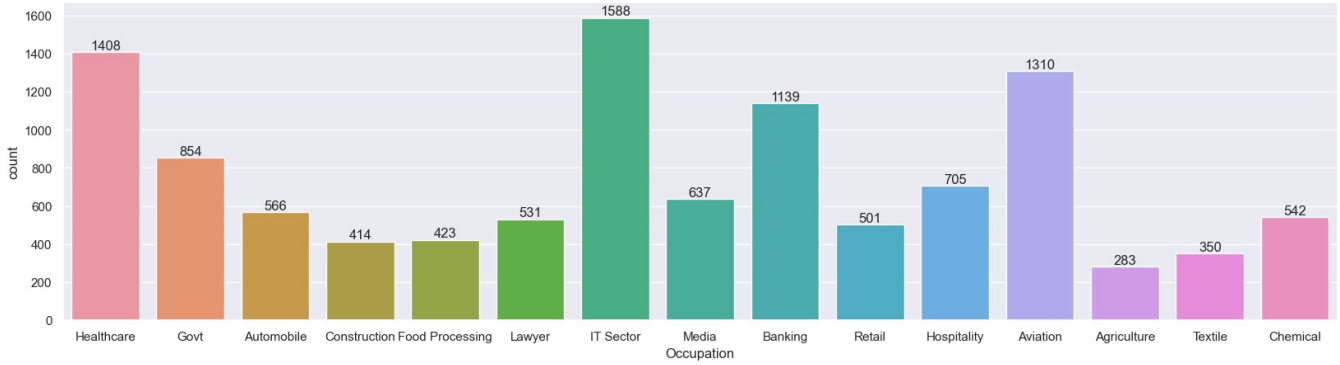
Out[103]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



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

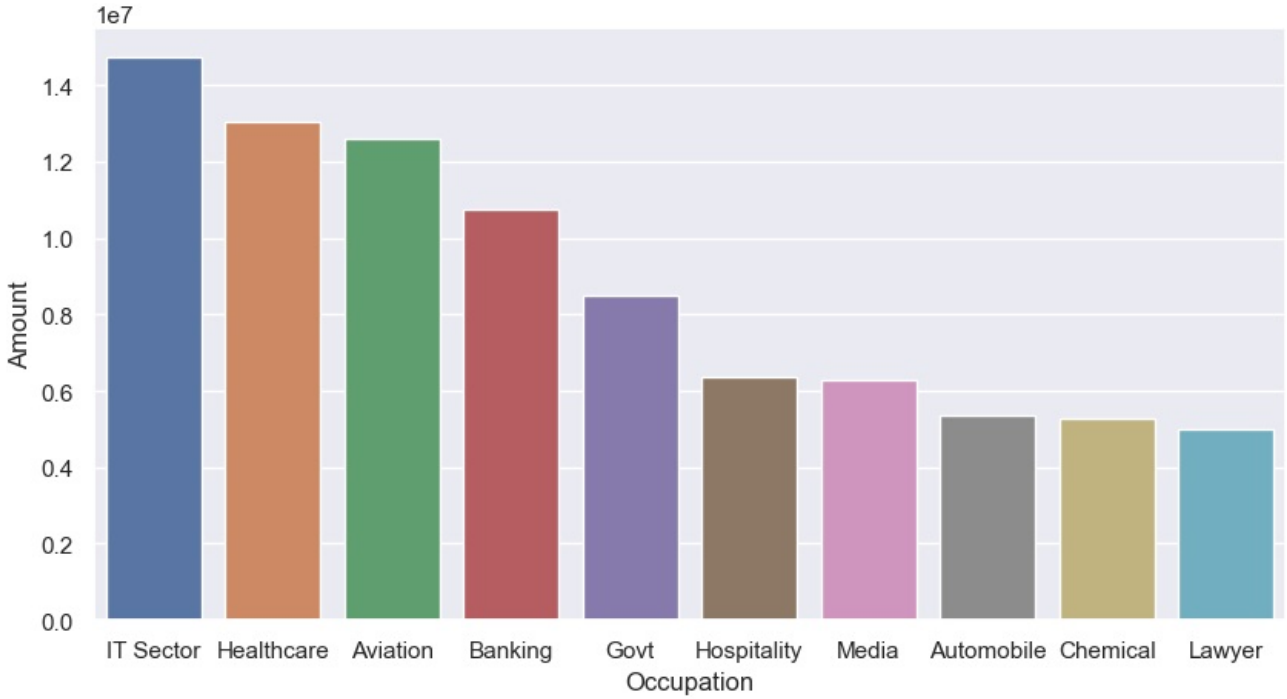
Occupation

In [104…
```
sns.set(rc={'figure.figsize':(20,5)})
ax= sns.countplot(data = df ,x= 'Occupation')
for bars in ax.containers:
    ax.bar_label(bars)
```



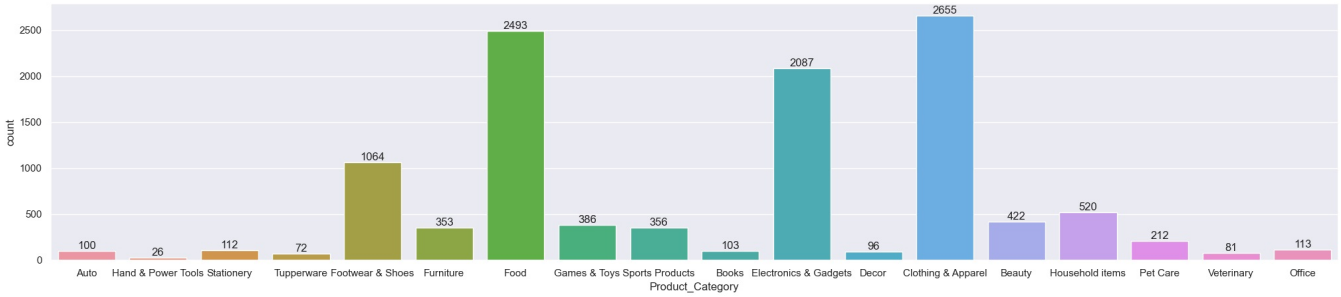
In [105…
```
sales_state= df.groupby(['Occupation'],as_index= False)['Amount'].sum().sort_values(by='Amount',ascending = Fal
sns.set(rc ={'figure.figsize':(10,5)})
sns.barplot(data = sales_state, x= 'Occupation', y ='Amount')
```

Out[105]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



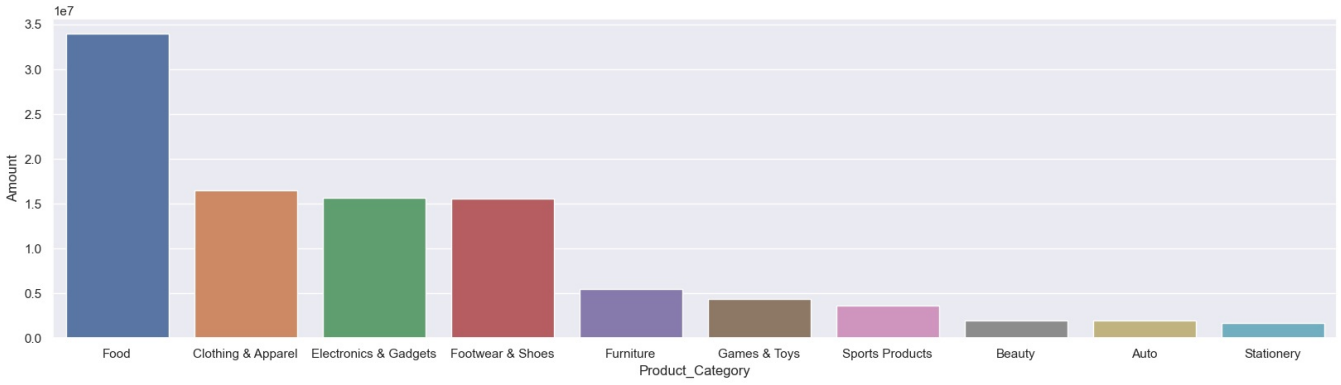
From above graph we can see that most of the buyers are working in IT,Healthcare,and Aviation sector

In [107…
```
sns.set(rc={'figure.figsize':(25,5)})
ax= sns.countplot(data = df ,x= 'Product_Category')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [108…
```
sales_state= df.groupby(['Product_Category'],as_index= False)['Amount'].sum().sort_values(by='Amount',ascending
sns.set(rc ={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x= 'Product_Category', y ='Amount')
```

Out[108]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```

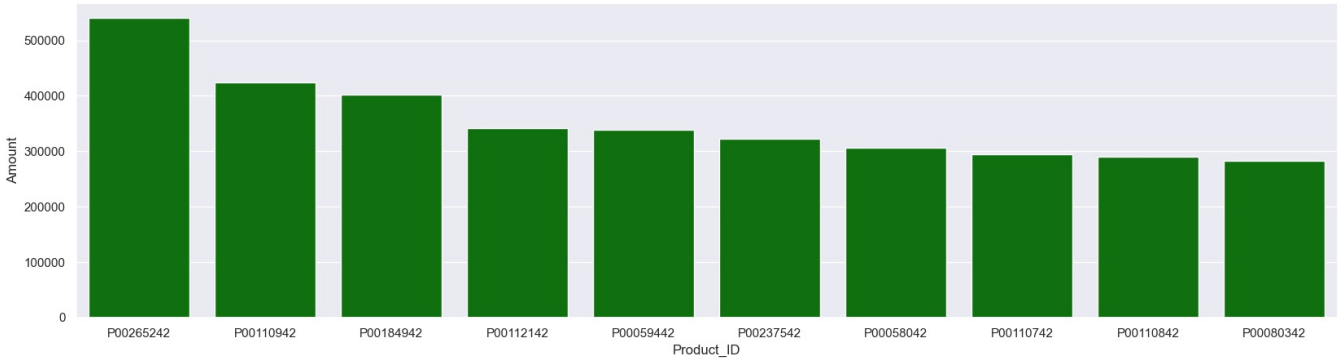


from the above graph we can see that most of the sold products are from Food,Footwear and Electronics category

Total Orders per ProductID

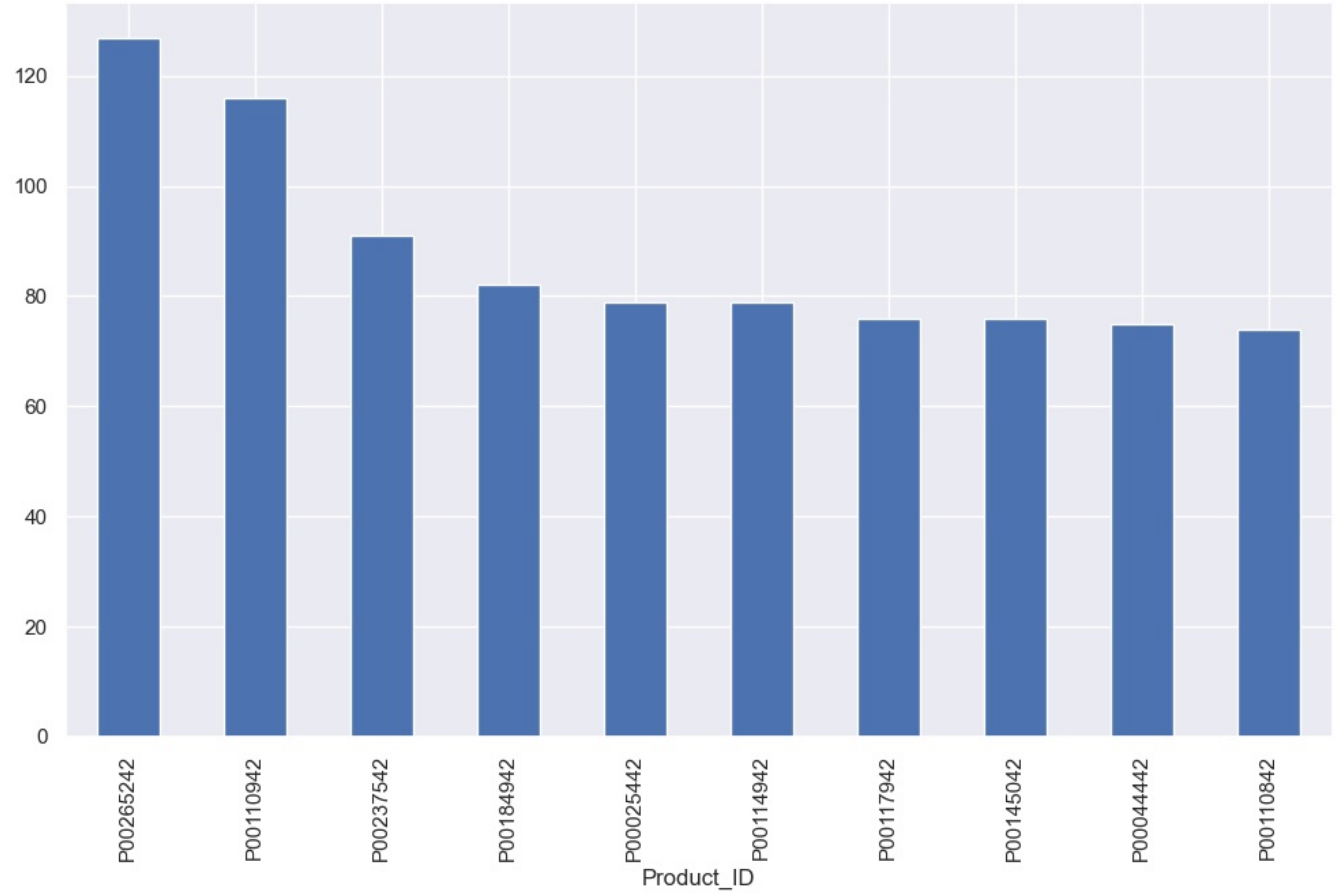
In [119…
```
sales_state= df.groupby(['Product_ID'],as_index= False)['Amount'].sum().sort_values(by='Amount',ascending = Fal
sns.set(rc ={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x= 'Product_ID', y ='Amount',color= 'green')
```

Out[119]:
```
<Axes: xlabel='Product_ID', ylabel='Amount'>
```



In [120…
```
# Top 10 Most sold product
fig1,ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending =False).plot(kind ='bar')
```

Out[120]:
```
<Axes: xlabel='Product_ID'>
```



Conclusion

Married women age-group yrs from UP, Maharastra and karnataka working in IT, Healthcare and Aviation are more likely to by products from food Clothing and Electronics Categorv.