

```
In [ ]: # import python Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

```
In [2]: # import csv file
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

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

Out[3]: (11251, 15)

```
In [6]: df.head(10)
```

Out[6]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	We
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Sou
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	C
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sou
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	We
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	No
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	C
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	We
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	C
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Sou

In [10]:

pd.isnull (df)

Out[10]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	
11246	False	False	False	False	False	False	False	False	False	
11247	False	False	False	False	False	False	False	False	False	
11248	False	False	False	False	False	False	False	False	False	
11249	False	False	False	False	False	False	False	False	False	
11250	False	False	False	False	False	False	False	False	False	

11251 rows × 13 columns

In [12]:

pd.isnull (df).sum()

Out[12]:

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 [13]:

drop null values
df.dropna(inplace=True)

In [14]:

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

In [15]:

df['Amount'].dtypes

Out[15]:

dtype('int32')

In [16]:

df.columns

Out[16]:

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

```
In [17]: #rename column
df.rename(columns= {'Marital_Status':'Shaadi'})
```

Out[17]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Z
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Wesi
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	South
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Cer
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	South
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Wesi
...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Wesi
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	North
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Cer
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	South
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Wesi

11239 rows × 13 columns

```
In [18]: # describe() method returns description of the data in the DataFrame (i.e. c
df.describe()
```

Out[18]:

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

Out[14]:

	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

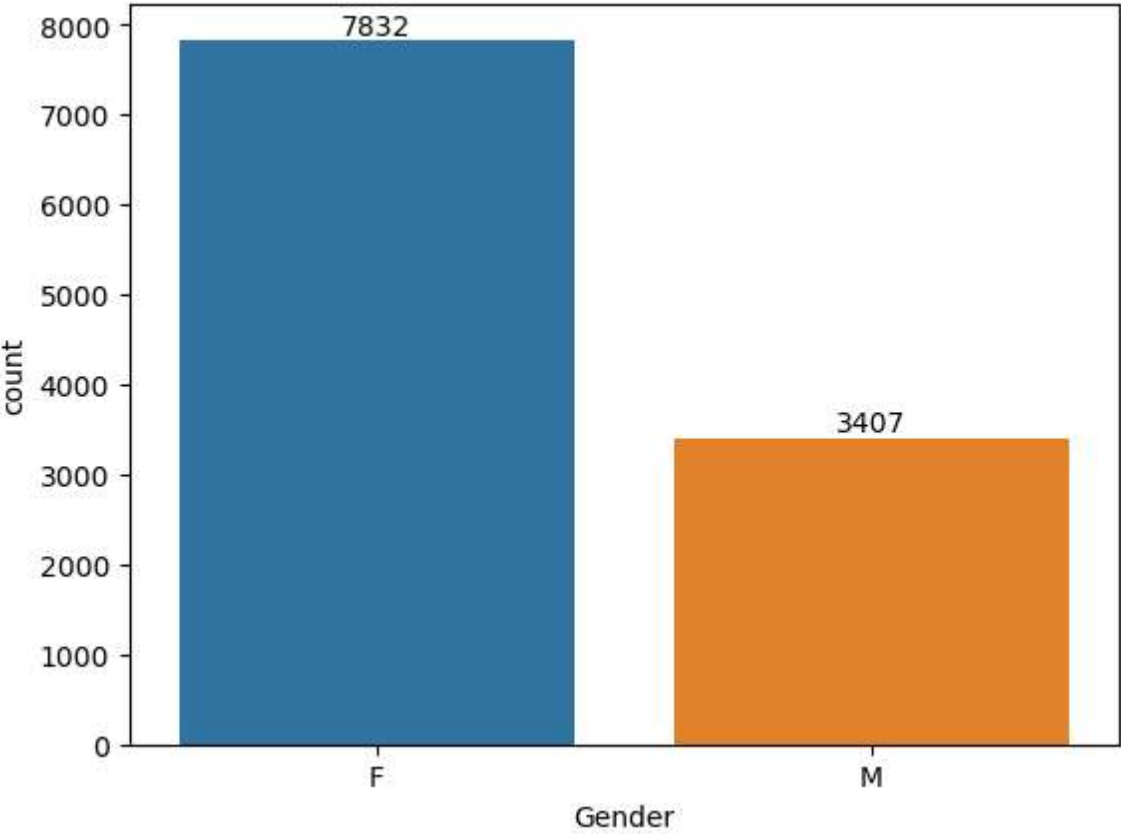
Exploratory Data Analysis

Gender

```
In [19]: # plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

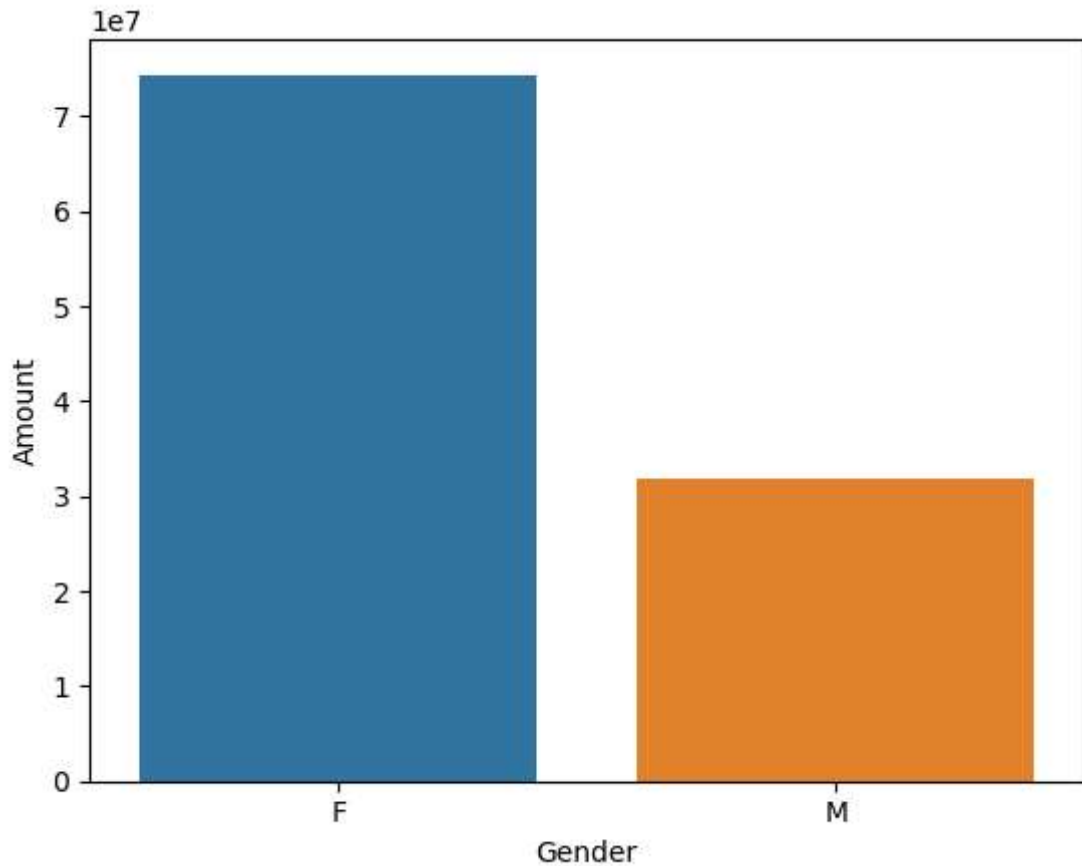
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [20]: # plotting a bar chart for gender vs total amount

sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_valu
sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gen)
```

```
Out[20]: <Axes: xlabel='Gender', ylabel='Amount'>
```



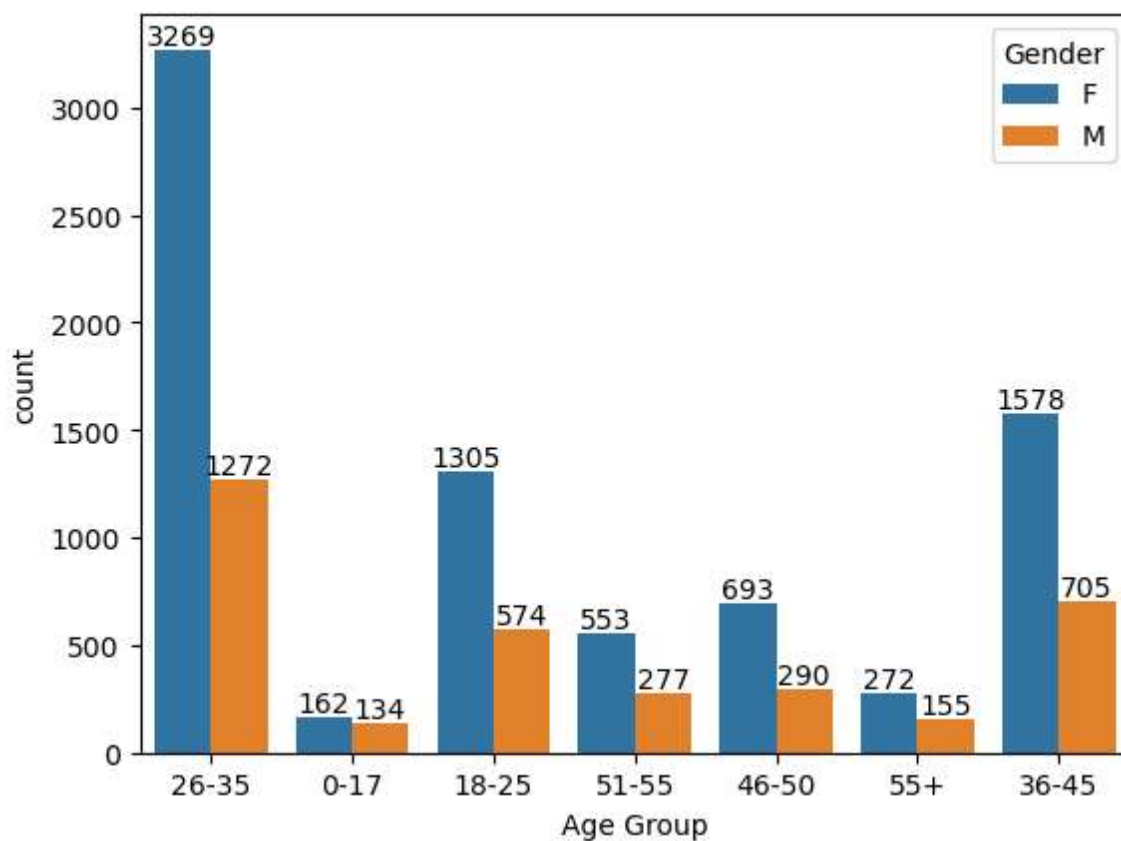
From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

Age

```
In [21]: df.columns
```

```
Out[21]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
               'Orders', 'Amount'],
              dtype='object')
```

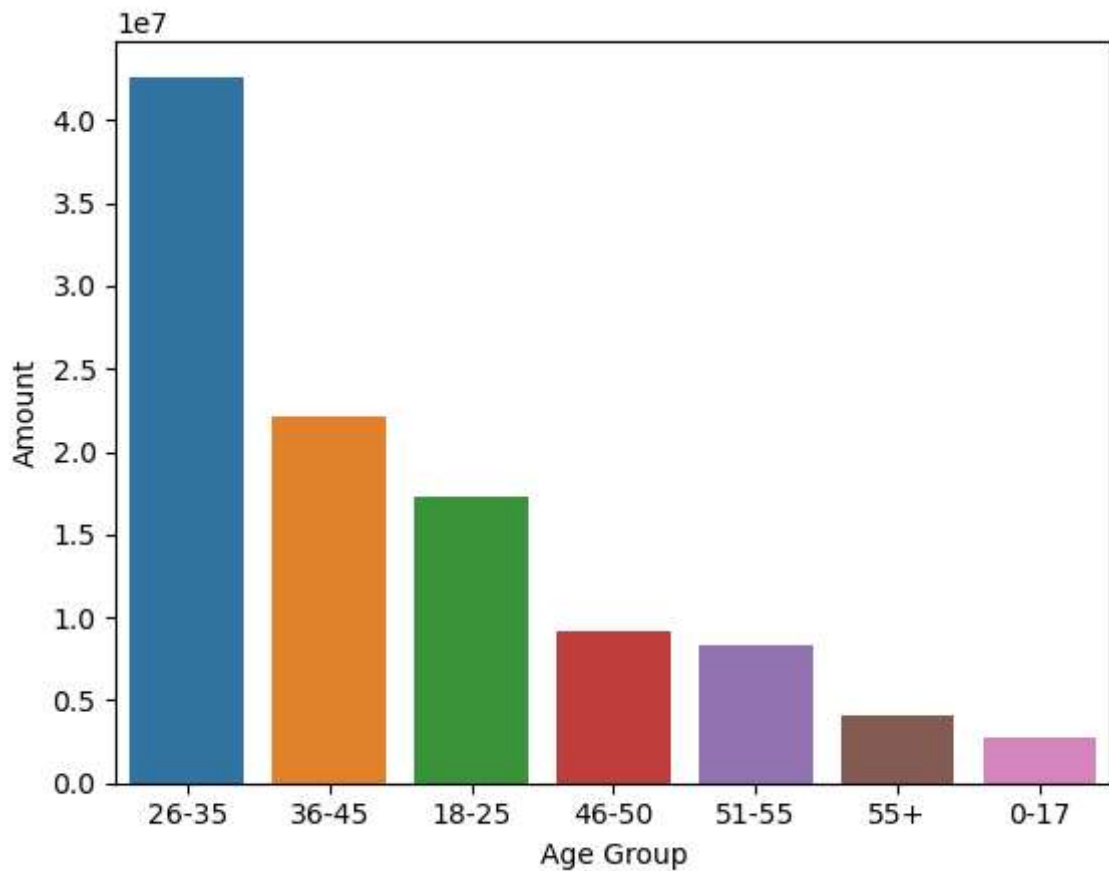
```
In [22]: ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')  
  
for bars in ax.containers:  
    ax.bar_label(bars)
```



```
In [23]: # Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_v

sns.barplot(x = 'Age Group',y= 'Amount' ,data = sales_age)
```

```
Out[23]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



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

State

```
In [24]: df.columns
```

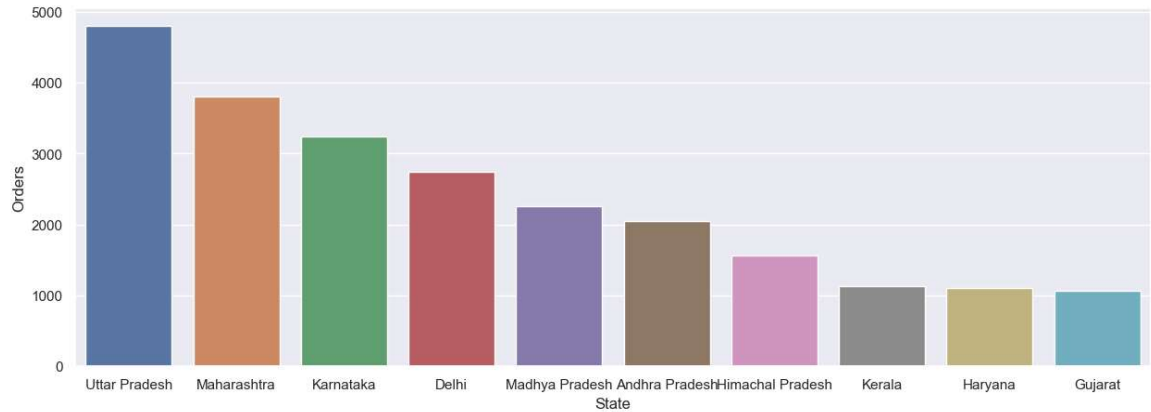
```
Out[24]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
               'Orders', 'Amount'],
              dtype='object')
```


In [25]: *# total number of orders from top 10 states*

```
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_val

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Orders')
```

Out[25]: <Axes: xlabel='State', ylabel='Orders'>

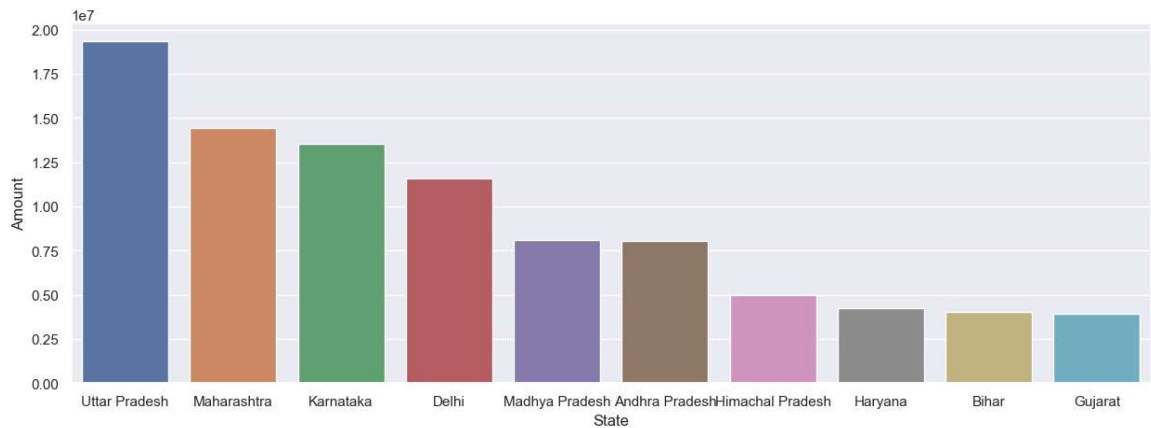


In [26]: *# total amount/sales from top 10 states*

```
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_val

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Amount')
```

Out[26]: <Axes: xlabel='State', ylabel='Amount'>

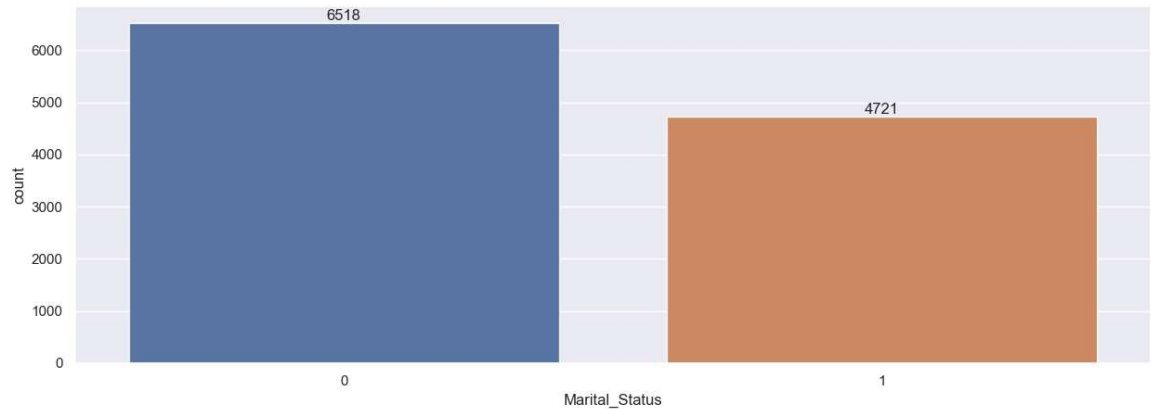


From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

Marital Status

```
In [27]: ax = sns.countplot(data = df, x = 'Marital_Status')

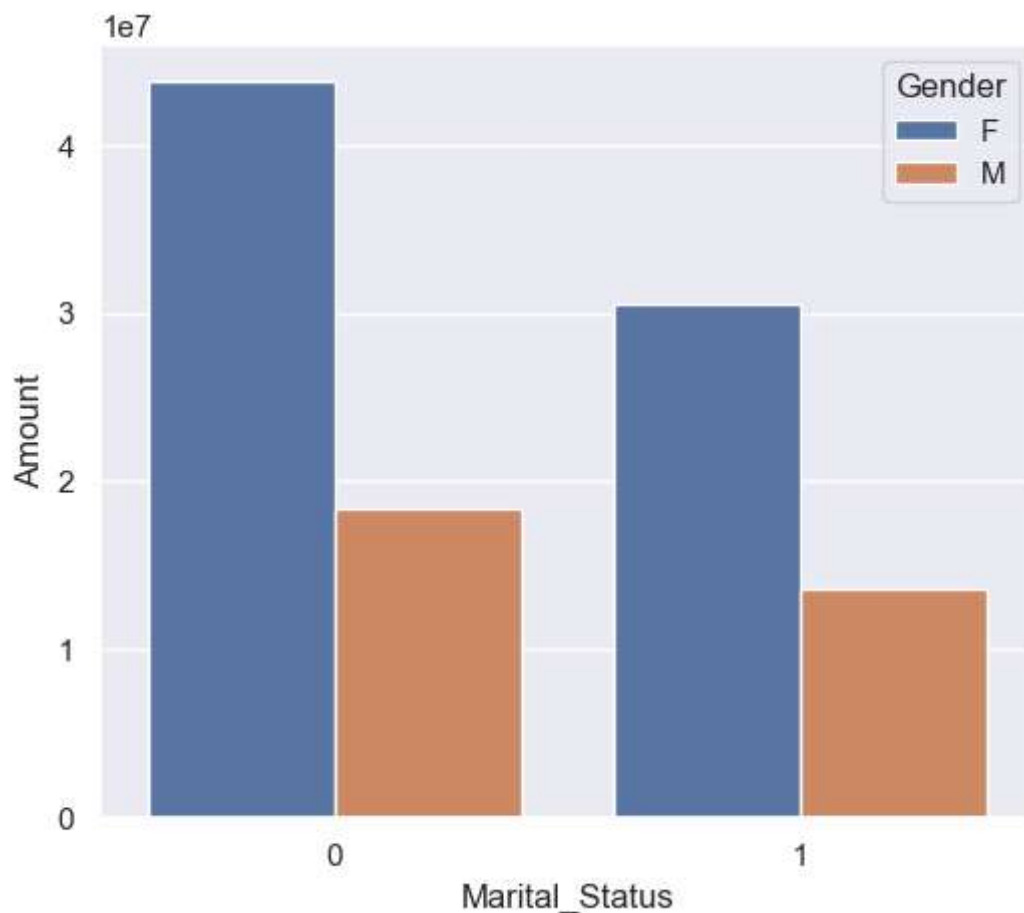
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [28]: sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount']

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status', y= 'Amount', hue='Gender')
```

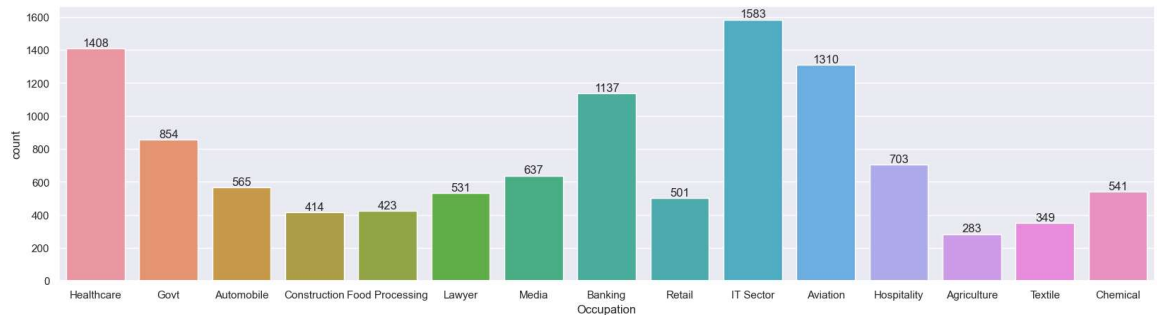
Out[28]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



Occupation

```
In [29]: sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

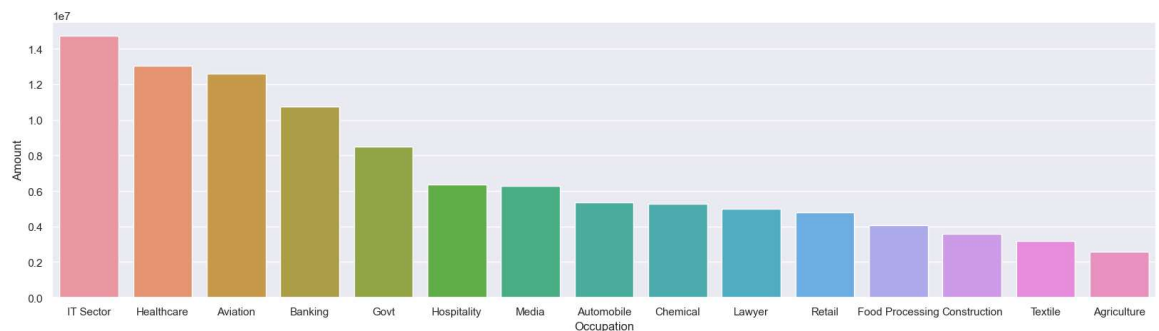
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [30]: sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Occupation', y = 'Amount')
```

Out[30]: <Axes: xlabel='Occupation', ylabel='Amount'>

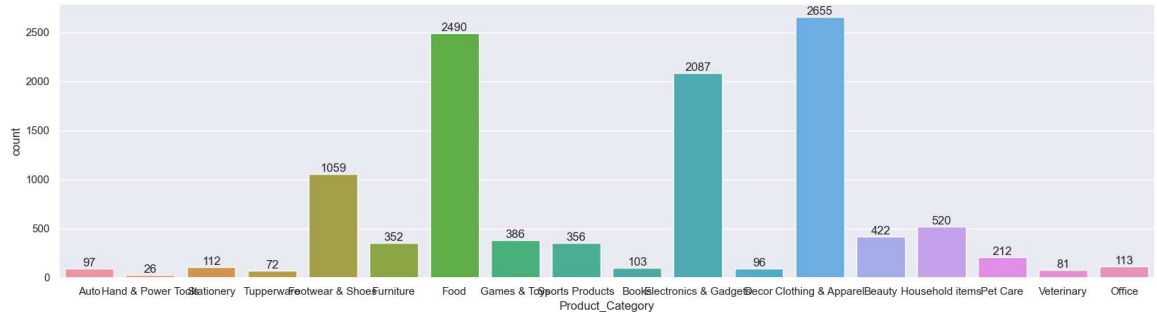


From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

Product Category

```
In [31]: sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

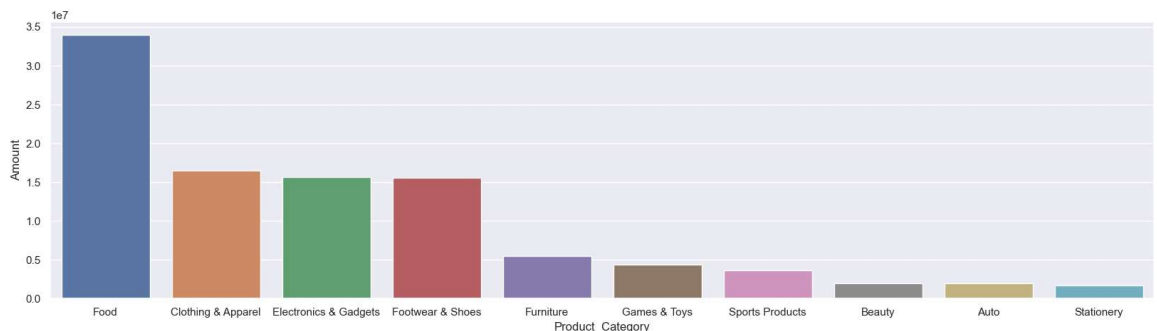
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [32]: sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum()

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_Category', y= 'Amount')
```

Out[32]: <Axes: xlabel='Product_Category', ylabel='Amount'>

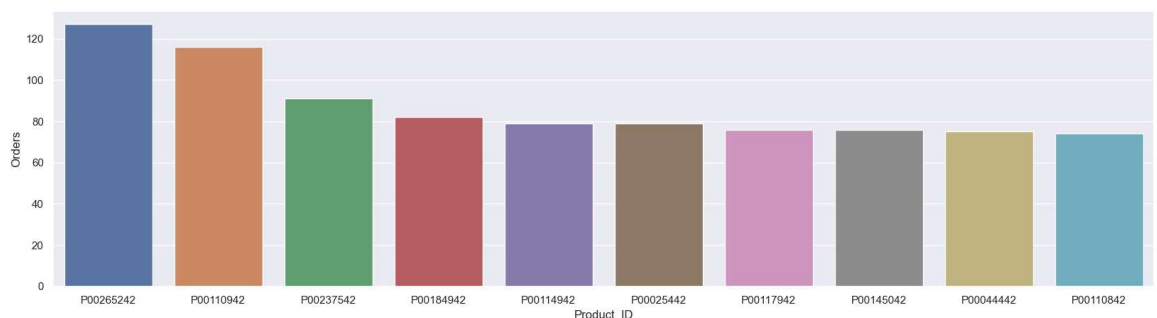


From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

```
In [33]: sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_ID', y= 'Orders')
```

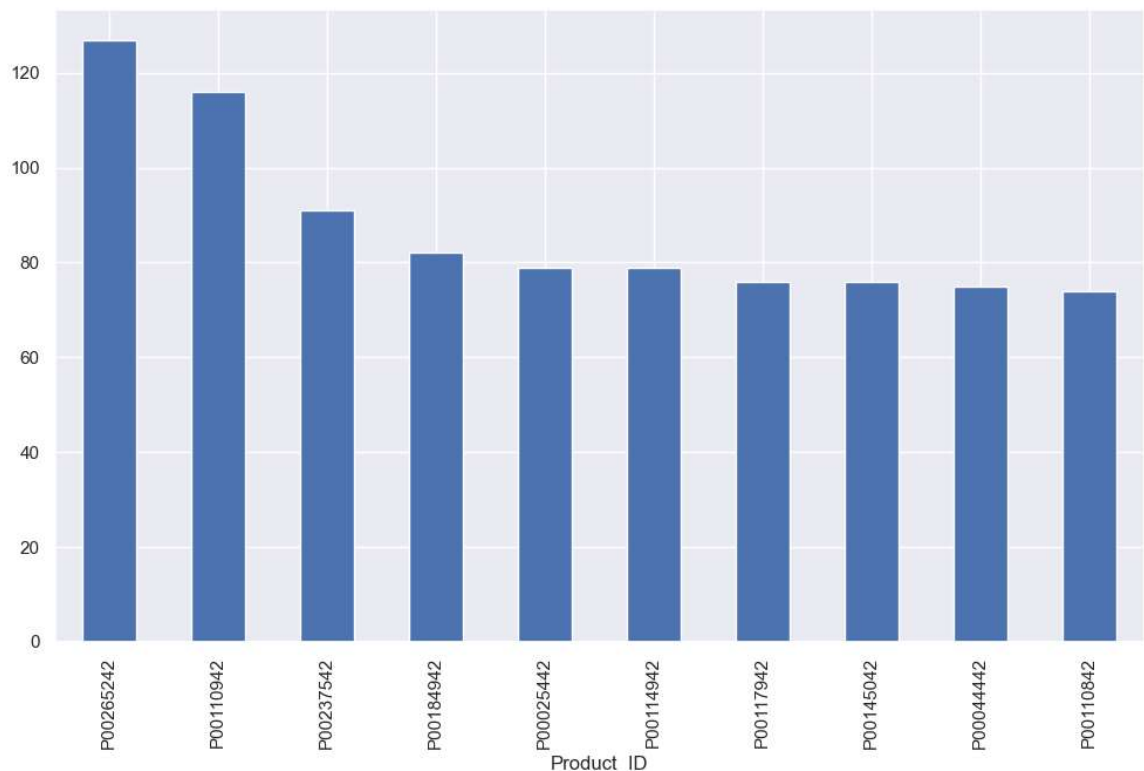
Out[33]: <Axes: xlabel='Product_ID', ylabel='Orders'>



```
In [34]: # top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=

Out[34]: <Axes: xlabel='Product_ID'>
```



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

Thank you!