

```
In [74]: #import python Librabies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
In [75]: df = pd.read_csv(r"C:\Users\chirag choudhary\Downloads\Diwali Sales Data.csv",
```

```
In [76]: df.shape
```

```
Out[76]: (11251, 15)
```

```
In [77]: df.head()
```

```
Out[77]:
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zip
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	West
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	South
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Cent
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	South
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	West

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

```
Out[78]:
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zip
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	We
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Nor
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Ce
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Sou
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	We

In [79]: 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 [80]: *#dropping unrelaed data/ blank column*  
df.drop(['Status', 'unnamed1'], axis=1, inplace= True)

In [81]: 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 [82]: pd.isnull(df).sum()
```

```
Out[82]: 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 [83]: df.dropna(inplace = True)
```

```
In [84]: pd.isnull(df).sum() # null value is dropped
```

```
Out[84]: 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 [85]: df.columns
```

```
Out[85]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
               'Orders', 'Amount'],
              dtype='object')
```

```
In [86]: df.rename(columns= {'Marital_Status': 'Shaadi'})
```

Out[86]:

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

11239 rows × 13 columns



```
In [87]: df.describe() #provide the math of the data
```

Out[87]:

	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	9453.610858
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355869
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 [88]: df[['Age', 'Orders', 'Amount']].describe()
```

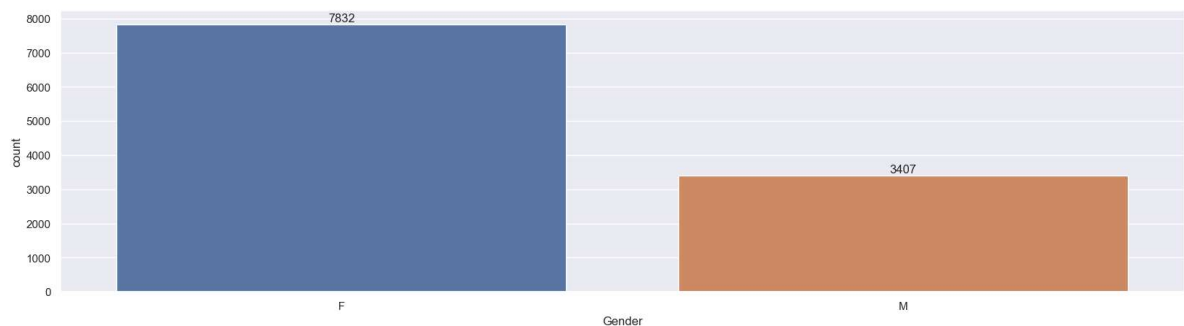
```
Out[88]:
```

	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610858
std	12.753866	1.114967	5222.355869
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

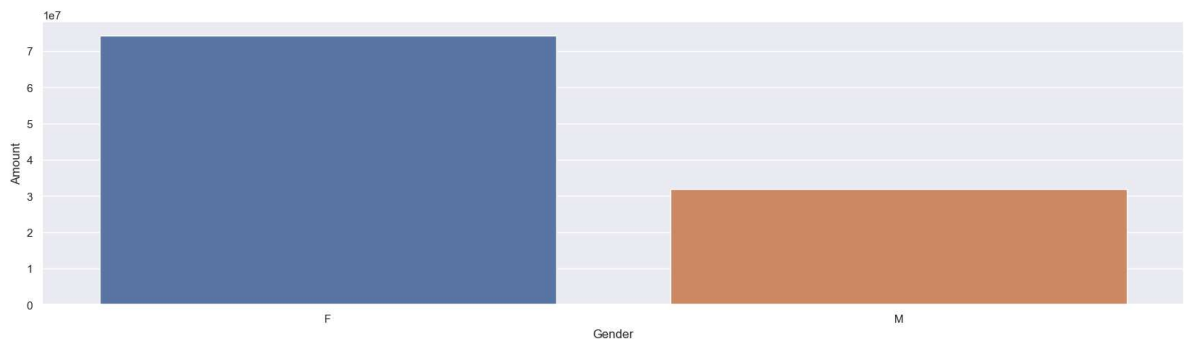
```
In [89]: #now using matplotlib and seaborn to visulaizing the data
```

```
In [90]: sx = sns.countplot(x = 'Gender',data=df)
for bars in sx.containers:
    sx.bar_label(bars)
```



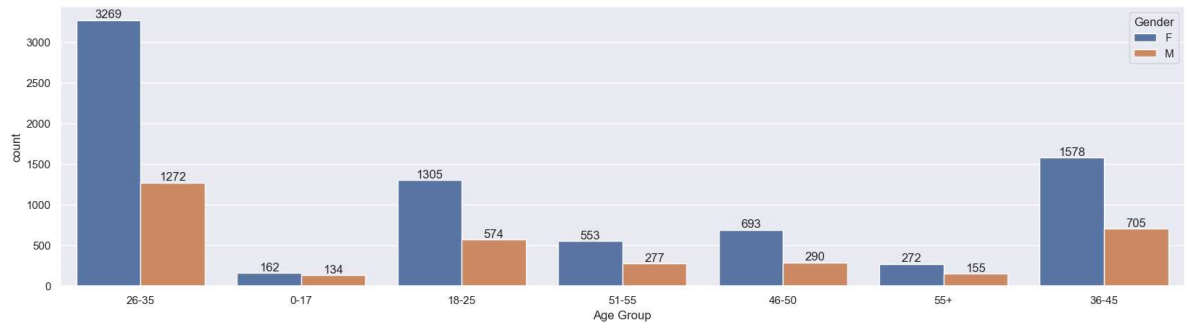
```
In [91]: #plotting a bar chart for gender vs total amount
sales_gen = df.groupby(['Gender'],as_index = False)['Amount'].sum().sort_values
sns.barplot(x = 'Gender',y='Amount',data = sales_gen)
```

```
Out[91]: <Axes: xlabel='Gender', ylabel='Amount'>
```



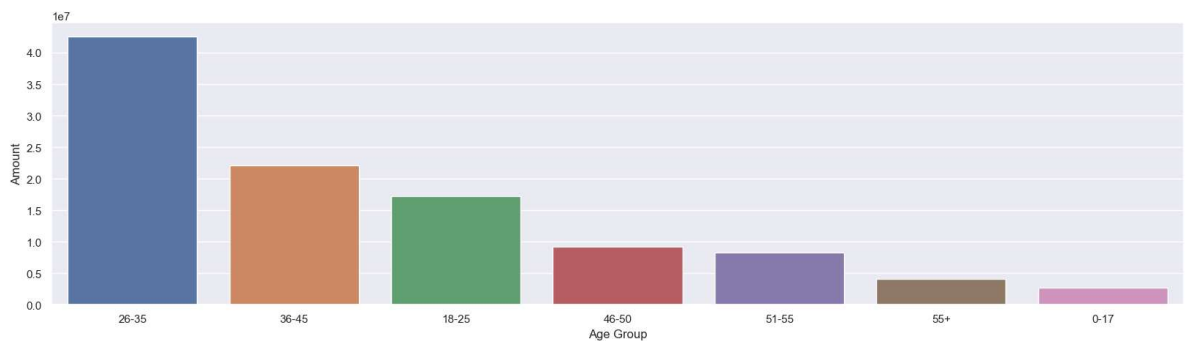
*From above chart we can conclude that most of buyers are female and even the purchasing power of females are greater than men.*

```
In [92]: sx = sns.countplot(data=df,x = 'Age Group',hue = 'Gender')
for bars in sx.containers:
    sx.bar_label(bars)
```



```
In [93]: sales_gen = df.groupby(['Age Group'],as_index = False)['Amount'].sum().sort_values
sns.barplot(x = 'Age Group',y='Amount',data = sales_gen)
```

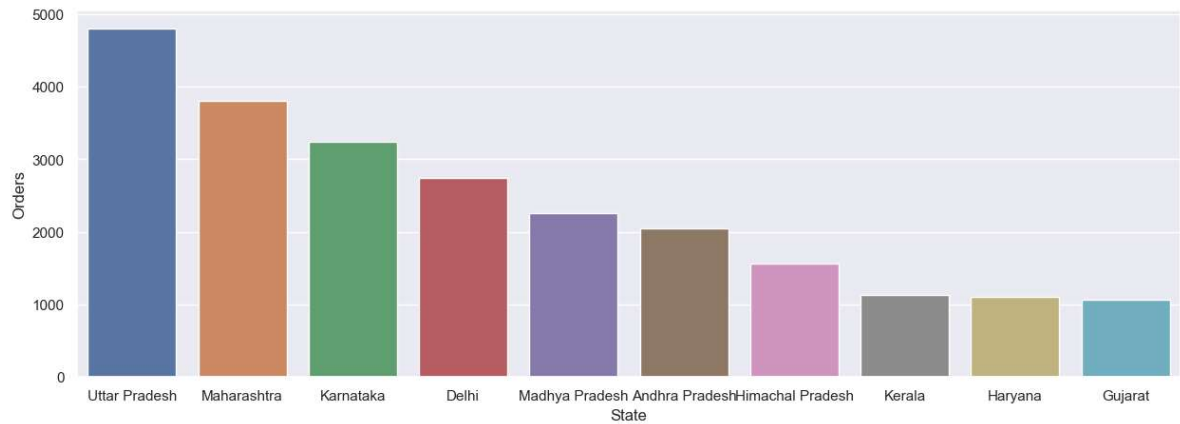
```
Out[93]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



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

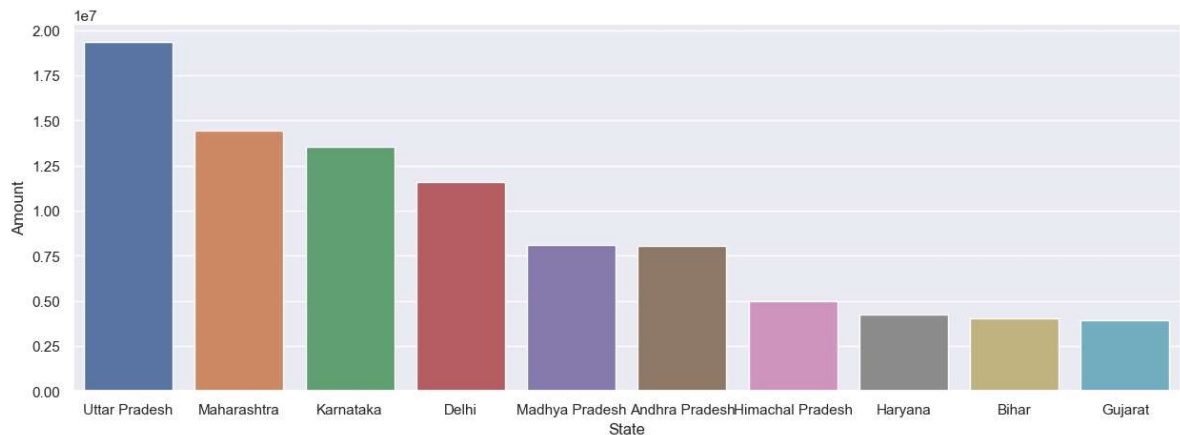
```
In [94]: sales_gen = df.groupby(['State'],as_index = False)['Orders'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x = 'State',y='Orders',data = sales_gen)
```

Out[94]: <Axes: xlabel='State', ylabel='Orders'>



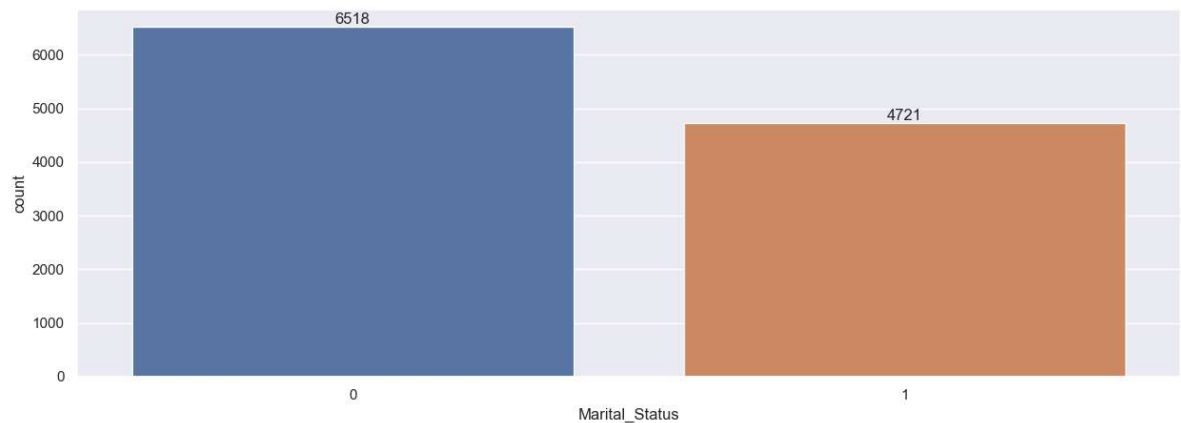
```
In [95]: sales_gen = df.groupby(['State'],as_index = False)['Amount'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(x = 'State',y='Amount',data = sales_gen)
```

Out[95]: <Axes: xlabel='State', ylabel='Amount'>



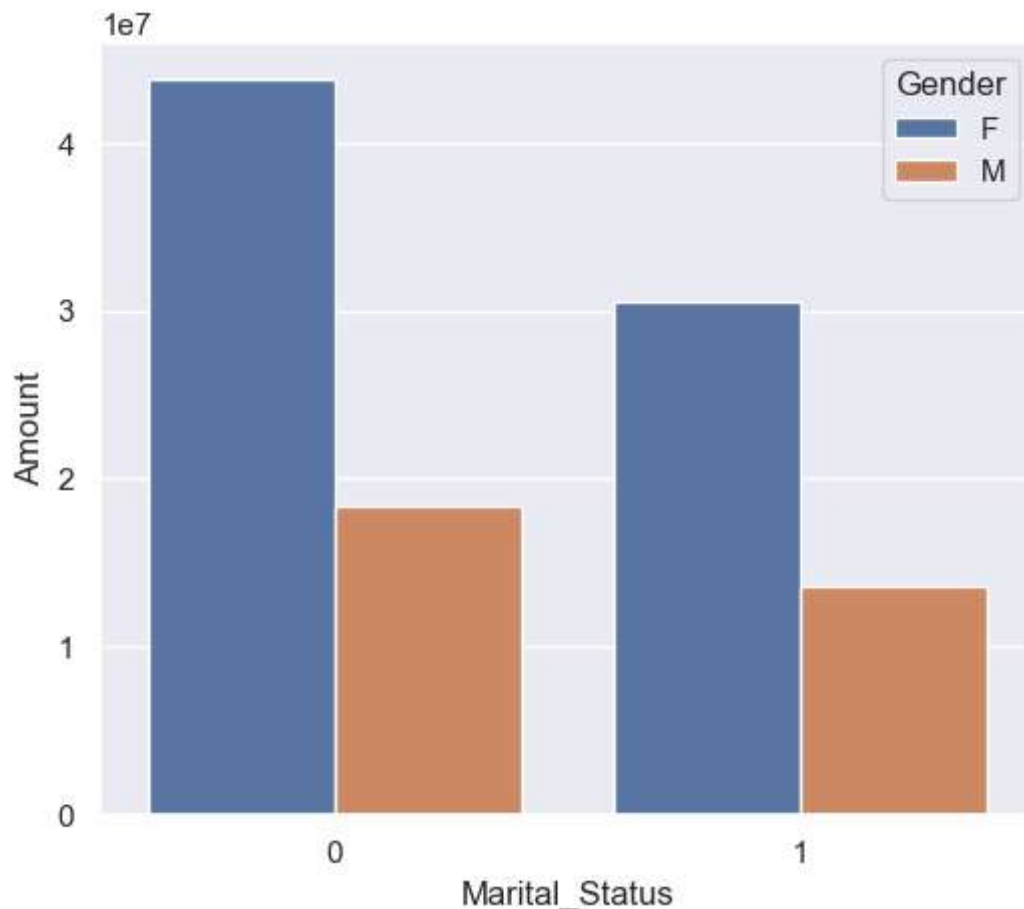
*from above graphs we can see that the most of orders & total sales/amount are from uttar pradesh, maharashtra and karnataka respectively*

```
In [96]: ax = sns.countplot(data = df, x = 'Marital_Status')
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [97]: 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[97]: <Axes: xlabel='Marital\_Status', ylabel='Amount'>

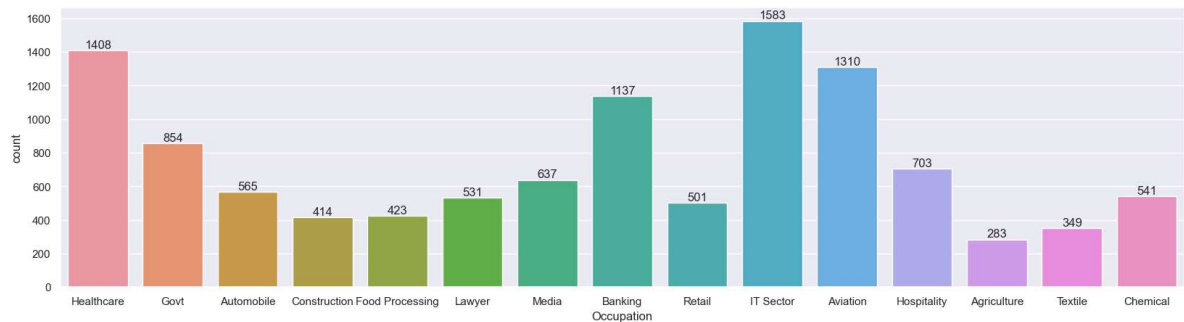




from above graph we can see that most of the buyers are married (women) and they have high purchasing power

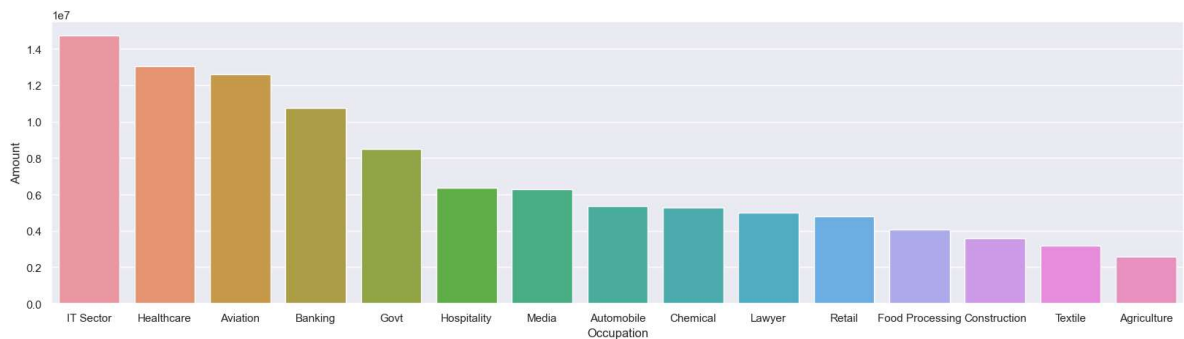
```
In [98]: sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

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



```
In [102]: sales_state = df.groupby(['Occupation'],as_index = False)['Amount'].sum().sort
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state,x = 'Occupation', y= 'Amount')
```

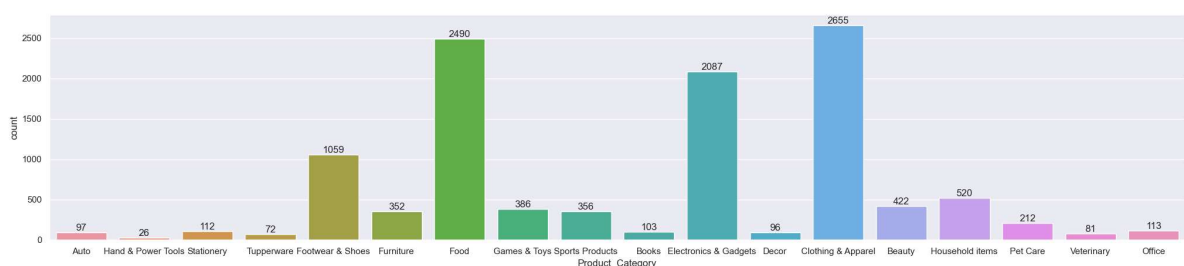
Out[102]: <Axes: xlabel='Occupation', ylabel='Amount'>



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

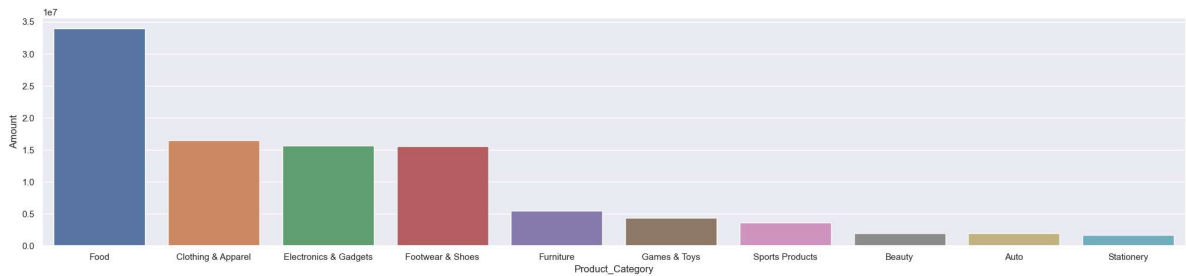
```
In [104]: sns.set(rc={'figure.figsize':(25,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

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



```
In [107]: sales_state = df.groupby(['Product_Category'],as_index = False)['Amount'].sum()
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data = sales_state,x = 'Product_Category', y= 'Amount')
```

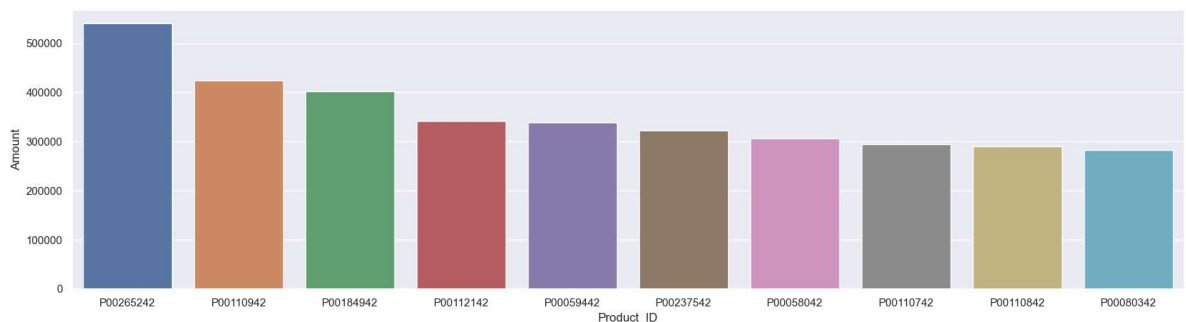
Out[107]: <Axes: xlabel='Product\_Category', ylabel='Amount'>



From above graphs we can see that most of the sold product are food, clothing and electronics.

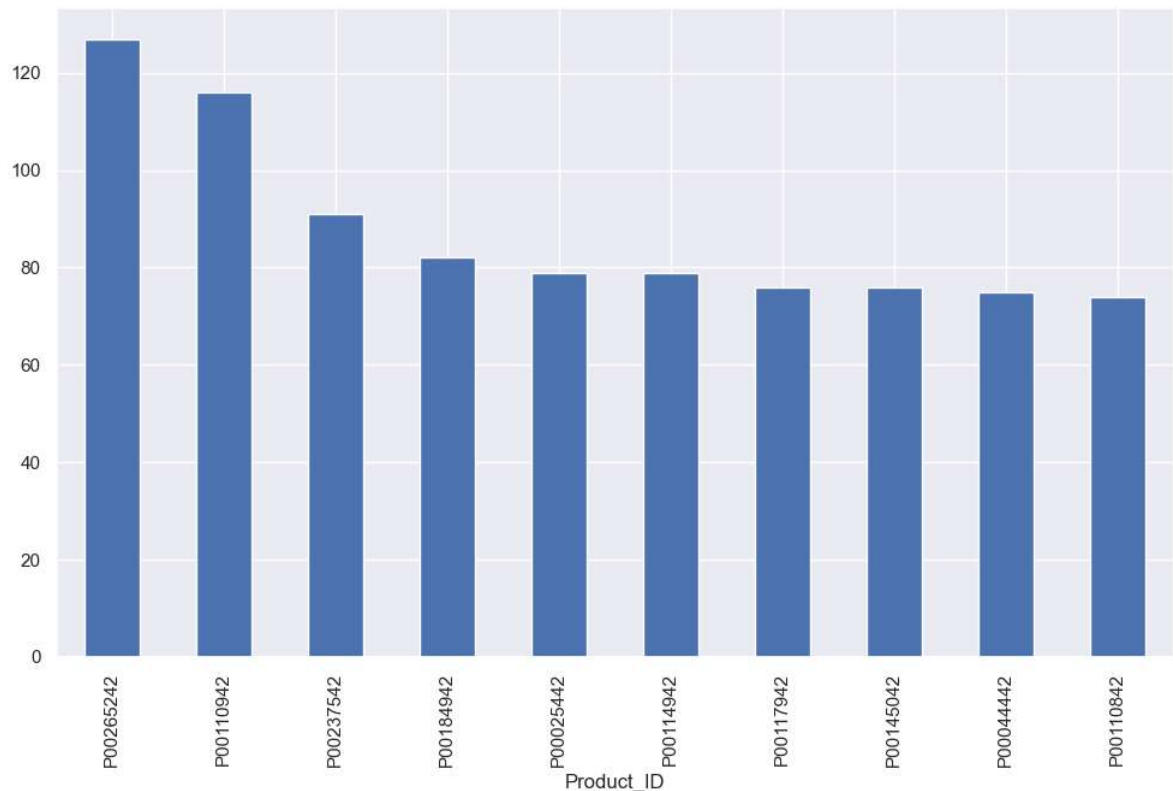
```
In [110]: sales_state = df.groupby(['Product_ID'],as_index = False)['Amount'].sum().sort()
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state,x = 'Product_ID', y= 'Amount')
```

Out[110]: <Axes: xlabel='Product\_ID', ylabel='Amount'>



```
In [111]: #top 10 product most sold
fig1, ax1 = plt.subplots(figsize = (12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending =
```

```
Out[111]: <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*