

```
In [6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

Matplotlib is building the font cache; this may take a moment.

```
In [10]: df = pd.read_csv(r'C:\Users\lenovo\Downloads\Python_Diwali_Sales_Analysis\Diwali Sales
```

```
In [11]: df.shape
```

```
Out[11]: (11251, 15)
```

```
In [12]: df.head()
```

```
Out[12]:
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status		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	C	
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western		

```
In [13]: 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 [16]: df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

```
In [17]: 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 [19]: pd.isnull(df)
```

```
Out[19]:
```

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

11251 rows × 13 columns

```
In [21]: pd.isnull(df).sum()
```

```
Out[21]: 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 [22]: df.shape
```

```
Out[22]: (11251, 13)
```

```
In [23]: df.dropna(inplace=True)
```

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

```
Out[24]: (11239, 13)
```

```
In [25]: pd.isnull(df).sum()
```

```
Out[25]: 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 [26]: # change data type#  
df['Amount']= df['Amount'].astype('int')
```

```
In [27]: df['Amount'].dtypes
```

```
Out[27]: dtype('int64')
```

```
In [28]: #checking of column names#  
df.columns
```

```
Out[28]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',  
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',  
               'Orders', 'Amount'],  
               dtype='object')
```

```
In [31]: #name changing#
```

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

```
Out[31]:
```

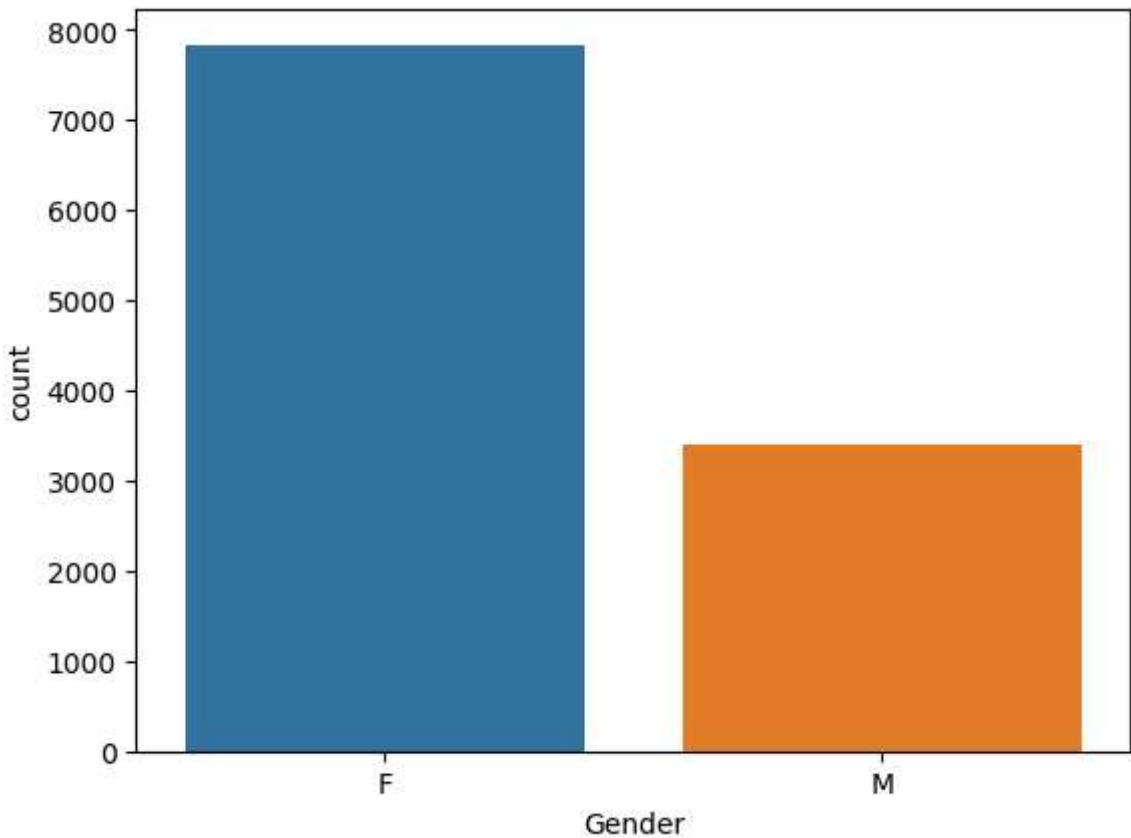
	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status		State	Zo
0	1002903	Sanskriti	P00125942	F	26-35	28		0	Maharashtra	West
1	1000732	Kartik	P00110942	F	26-35	35		1	Andhra Pradesh	South
2	1001990	Bindu	P00118542	F	26-35	35		1	Uttar Pradesh	Cent
3	1001425	Sudevi	P00237842	M	0-17	16		0	Karnataka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

11239 rows × 13 columns

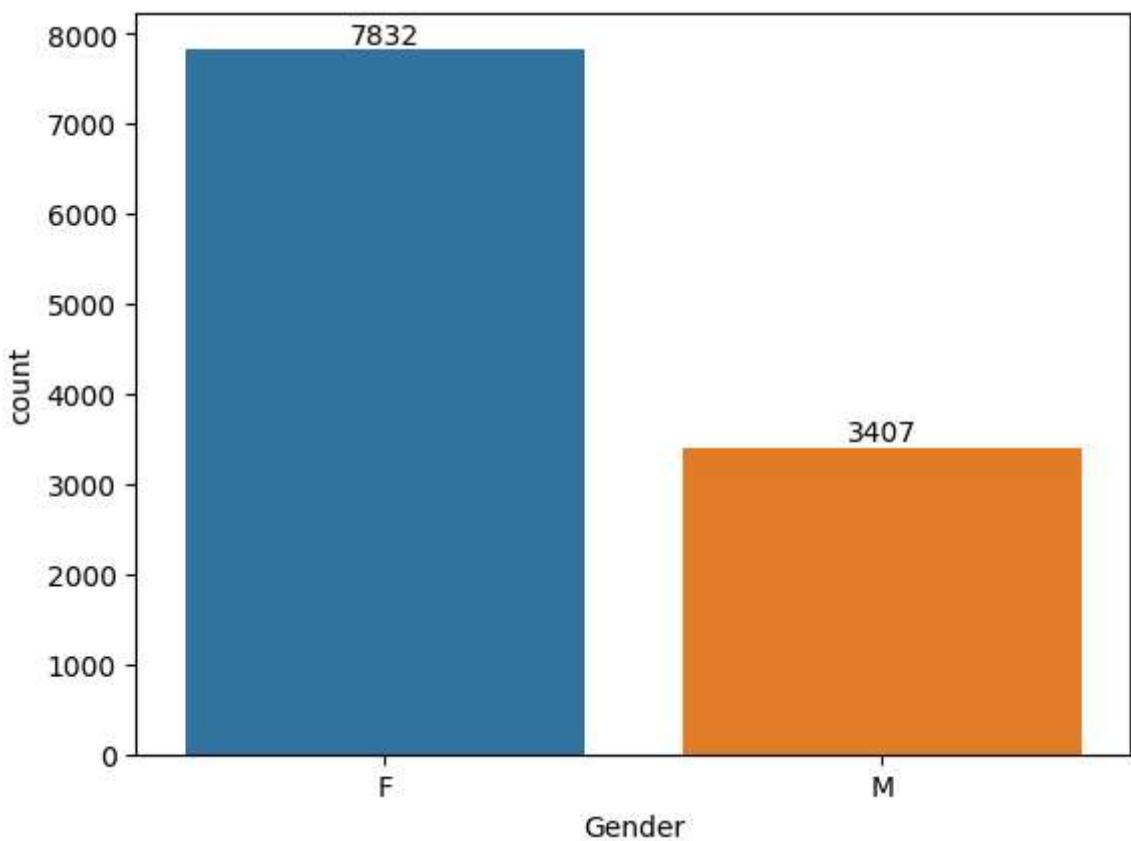
## GENDER

```
In [47]: sns.countplot(data=df,x='Gender',hue='Gender')
```

```
Out[47]: <Axes: xlabel='Gender', ylabel='count'>
```



```
In [48]: ax=sns.countplot(x='Gender',data=df,hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



from above we can say that female purchase is 4425 more than male)

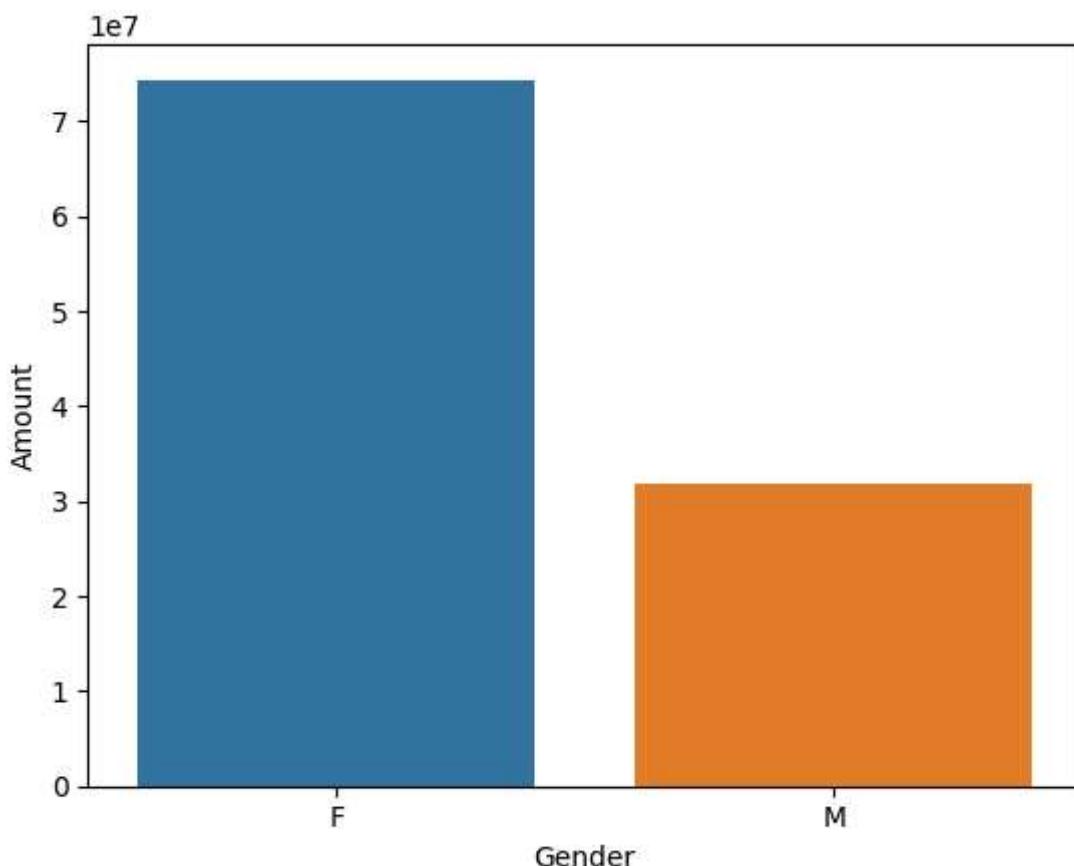
```
In [35]: df.groupby(['Gender'],as_index=False)[ 'Amount' ].sum().sort_values(by='Amount',ascending=False)
```

```
Out[35]:   Gender  Amount
```

	Gender	Amount
0	F	74335853
1	M	31913276

```
In [49]: sales_gen= df.groupby(['Gender'],as_index=False)[ 'Amount' ].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Gender',y='Amount',data=sales_gen,hue='Gender')
```

```
Out[49]: <Axes: xlabel='Gender', ylabel='Amount'>
```



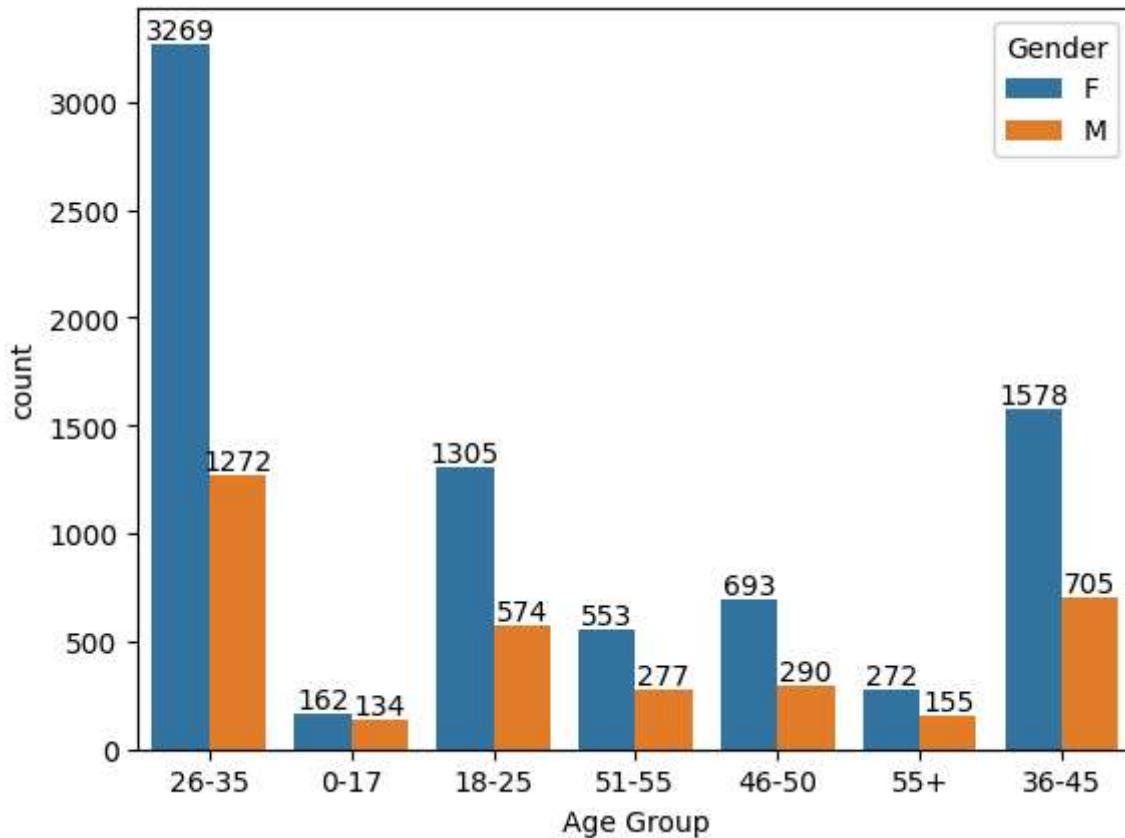
From above graphs we can see the most of the buyers are females and even the purchasing power of females are greater than men

## AGE

```
In [38]: df.columns
```

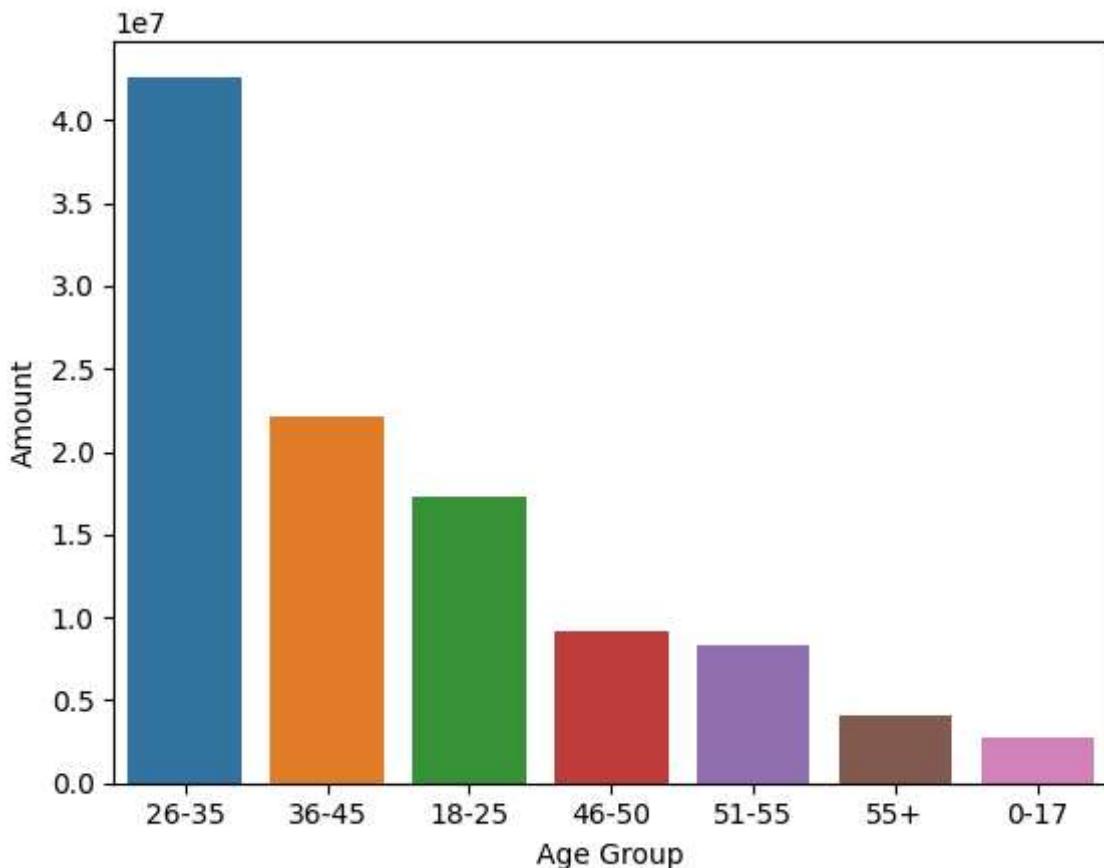
```
Out[38]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
In [40]: ax=sns.countplot(data=df,x='Age Group',hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [50]: sales_age = df.groupby(['Age Group'],as_index = False)[['Amount']].sum().sort_values(by='Amount', ascending=False)
sns.barplot(x='Age Group',y='Amount', data = sales_age,hue='Age Group')
```

```
Out[50]: <Axes: xlabel='Age Group', ylabel='Amount'>
```

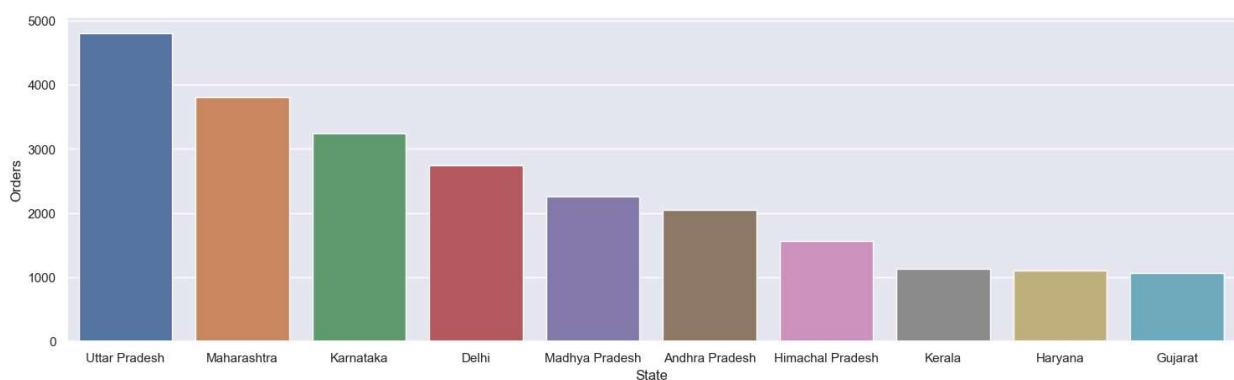


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

## STATE

```
In [58]: #Total number of orders 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':(18,5)})
sns.barplot(x='State',y='Orders', data = sales_state,hue='State')
```

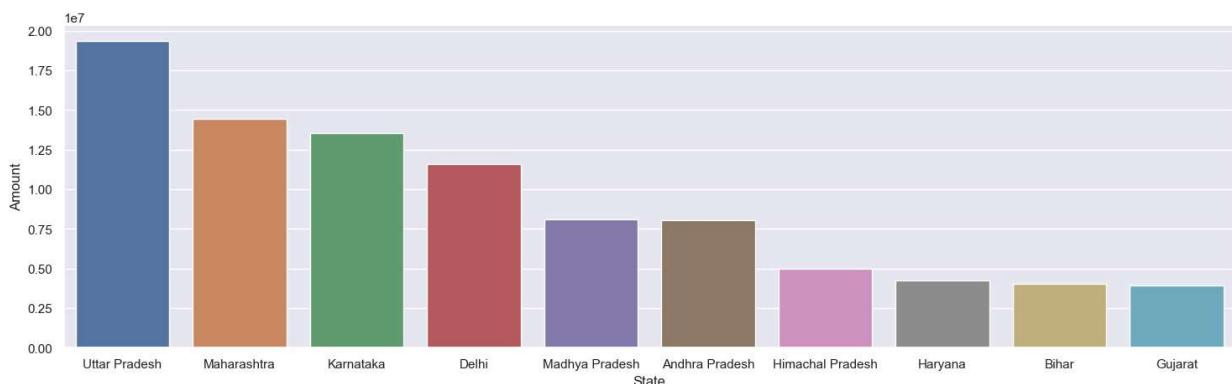
```
Out[58]: <Axes: xlabel='State', ylabel='Orders'>
```



```
In [59]: ##Total amount /sales from top 10 states##
sales_state = df.groupby(['State'],as_index = False)[ 'Amount'].sum().sort_values(by='Amount', ascending=False)
```

```
sns.set(rc={'figure.figsize':(18,5)})  
sns.barplot(x='State',y='Amount', data = sales_state,hue='State')
```

Out[59]: <Axes: xlabel='State', ylabel='Amount'>



Top 7 states retain its position in both cases, more number of orders as well as the purchasing power. Or we can say that most of the orders and total total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectevily.

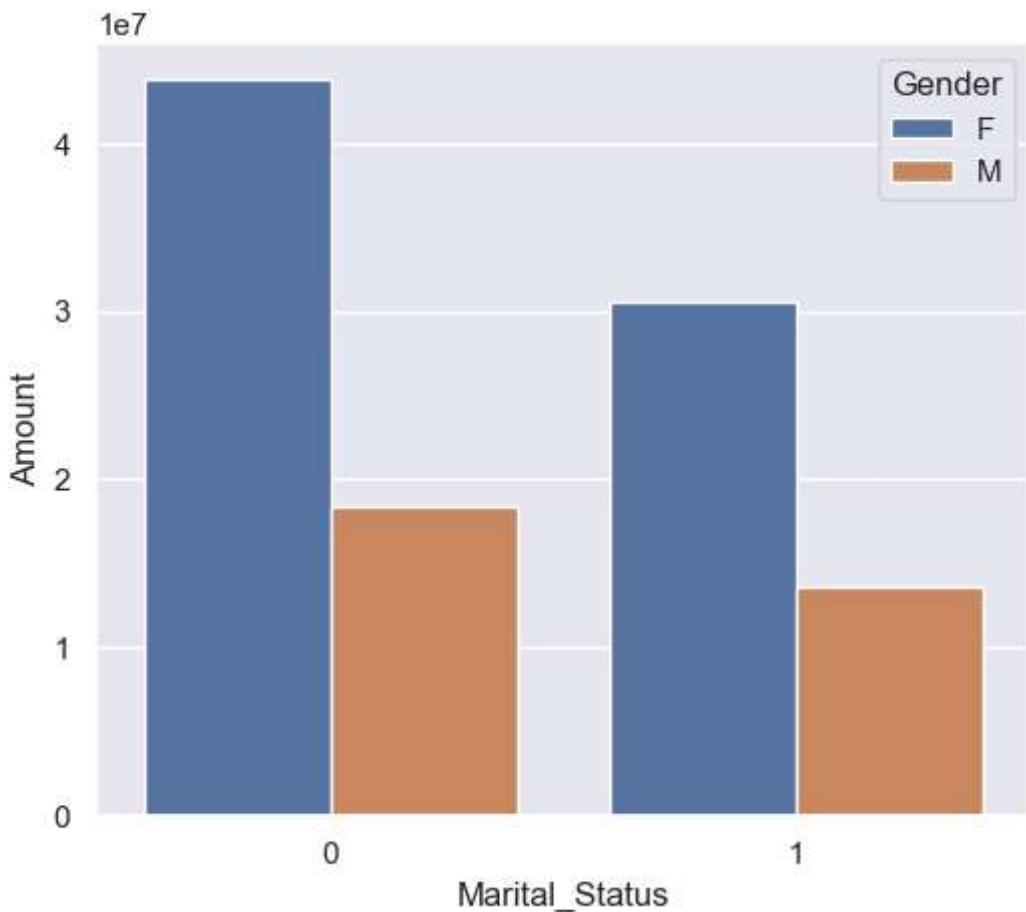
## MARITAL STATUS

```
In [66]: ax=sns.countplot(x='Marital_Status',data=df,hue='Marital_Status')  
for bars in ax.containers:  
    sns.set(rc={'figure.figsize':(6,8)})  
    ax.bar_label(bars)
```



```
In [74]: sales_Marital = df.groupby(['Marital_Status','Gender'],as_index = False)[['Amount']].sum()
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(x='Marital_Status',y='Amount', data = sales_Marital,hue='Gender')

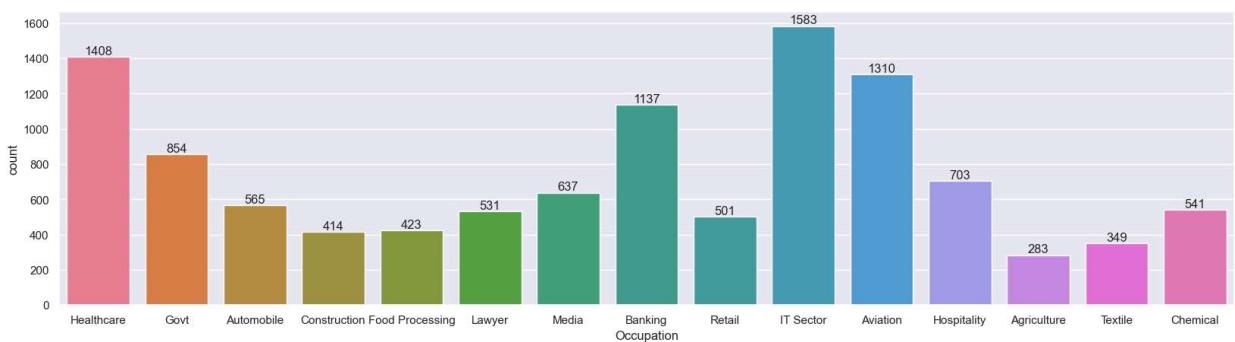
Out[74]: <Axes: xlabel='Marital_Status', ylabel='Amount'>
```



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

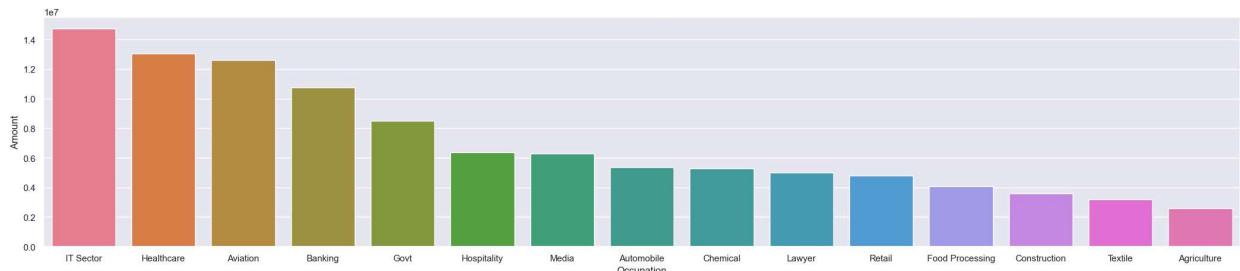
## OCCUPATION

```
In [76]: ax=sns.countplot(x='Occupation',data=df,hue='Occupation')
for bars in ax.containers:
    sns.set(rc={'figure.figsize':(26,5)})
    ax.bar_label(bars)
```



```
In [77]: sales_Marital = df.groupby(['Occupation'],as_index = False)[ 'Amount'].sum().sort_values
sns.set(rc={'figure.figsize':(26,5)})
sns.barplot(x='Occupation',y='Amount', data = sales_Marital,hue='Occupation')
```

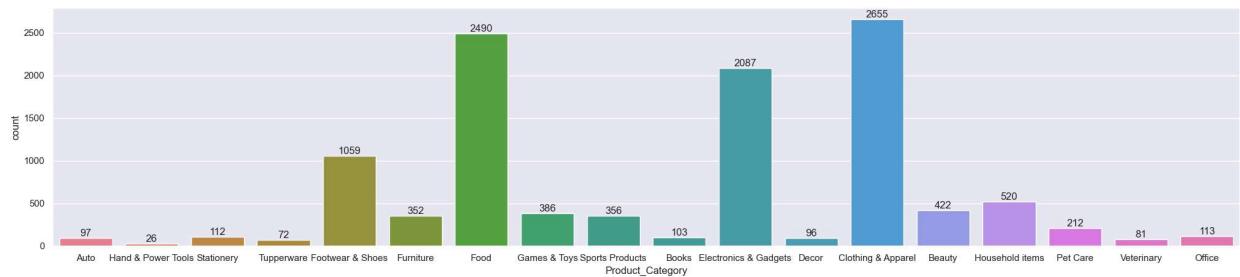
```
Out[77]: <Axes: xlabel='Occupation', ylabel='Amount'>
```



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

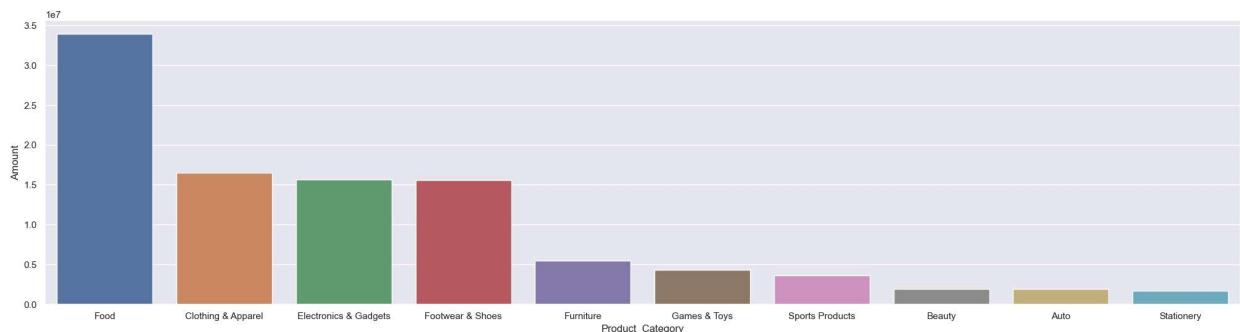
## PRODUCT CATEGORY

```
In [85]: ax=sns.countplot(x='Product_Category',data=df,hue='Product_Category')
for bars in ax.containers:
    sns.set(rc={'figure.figsize':(25,4)})
    ax.bar_label(bars)
```



```
In [88]: sales_Product = df.groupby(['Product_Category'],as_index = False)['Amount'].sum().sort
sns.set(rc={'figure.figsize':(25,6)})
sns.barplot(x='Product_Category',y='Amount', data = sales_Product,hue='Product_Cat
```

```
Out[88]: <Axes: xlabel='Product_Cat', ylabel='Amount'>
```

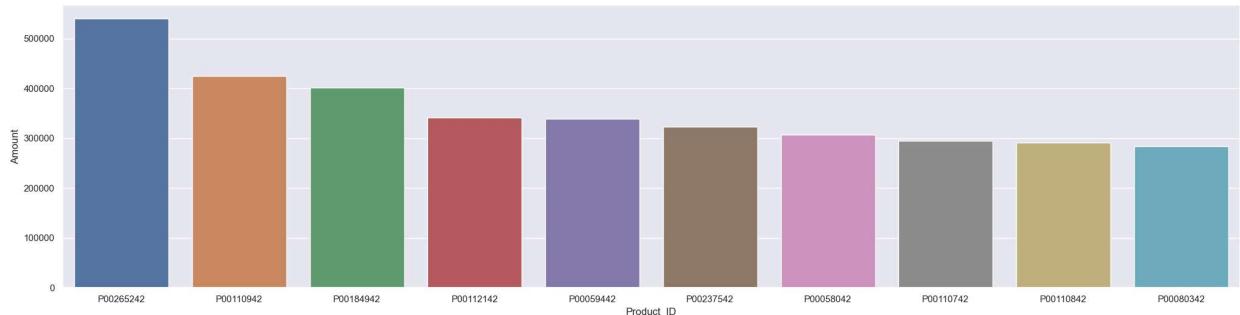


We can observe that most orderes product is clothing followed by food but in case of amount headed by food and followed by clothing.

# PRODUCT ID

```
In [89]: sales_Product_ID = df.groupby(['Product_ID'],as_index = False)[['Amount']].sum().sort_values(by='Amount', ascending=False)
sns.set(rc={'figure.figsize':(25,6)})
sns.barplot(x='Product_ID',y='Amount', data = sales_Product_ID,hue='Product_ID')
```

```
Out[89]: <Axes: xlabel='Product_ID', ylabel='Amount'>
```



## Conclusion:

Married women aged between 26 and 35 years, predominantly hailing from Uttar Pradesh and Maharashtra, with a significant representation from Karnataka, are actively employed in the IT, healthcare, and aviation industries. This demographic shows a strong preference for purchasing products within the food, clothing, and electronics categories.

```
In [ ]:
```