IMPORTING THE NECESSARY LIBRARIES

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

LOADING THE DATASET

```

df = pd.read_csv('Diwali Sales Data.csv', encoding='unicode_escape')
df
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Order
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
...	...	...	...	...	...	...	...	...	...	...	...	..
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

11251 rows × 15 columns

THIS COMMAND GIVES THE TOP 10 ROWS OF THE DATASET

```
df.head(10)
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Am
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	239
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	239
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	239
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	239
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	238
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	238
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	238
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	238
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	237

```
df.shape
```

```
(11251, 15)
```

THIS COMMAND GIVES ALL THE DATAYPES OF ALL THE COLUMNS AND IT ALSO TELLS US ABOUT THE NULL VALUES IN THE DATASET

```
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
```

THIS SHOWS ERROR BECUASE WE HAVE DROPPED THE COLUMN AND IT IS BEING DONE IN THE MAIN DATASET SINCE WE HAVE USED INPLACE=TRUE, THEREFORE IT CAN RUN ONLY ONCE

```
# drop blank columns
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-18-28f1304f796c> in <cell line: 2>()
      1 # drop blank columns
----> 2 df.drop(['Status','unnamed1'],axis=1,inplace=True)

                                3 frames

/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
   6998         if mask.any():
   6999             if errors != "ignore":
-> 7000                 raise KeyError(f"{labels[mask].tolist()} not found in axis")
   7001             indexer = indexer[~mask]
   7002         return self.delete(indexer)

KeyError: "['Status', 'unnamed1'] not found in axis"
```

```
df
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
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

11251 rows × 13 columns

```
1 df.isnull()
```

	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

AFTER WE DROP ALL THE NULL VALUES, THE DATAFRAME DOES NOT SHOW ANY NULL VALUES

```
1 df.isnull().sum()
```

```
                  0
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
```

```
df.shape
```

```
(11251, 13)
```

```
df.dropna(inplace=True)
```

```
df.isnull().sum()
```

```
                  0
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
```

```
df['Amount'] = df['Amount'].astype('int')
df['Amount'].dtypes
```

```
dtype('int64')
```

```
df.columns
```

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

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

```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Married?	State	Zone	Occupation	Product_Category	Orders	Amou
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	239
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	239
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	239
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	239
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	238
...	...	...	...	...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	3
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	3
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	2
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	2

```
THE DESCRIBE COMMAND GIVES THE STATISTICS OF THE COLUMN
WHICH CONTAINS NUMERICAL VALUES
```

THE DESCRIBE COMMAND GIVES THE STATISTICS OF THE COLUMNS WHICH CONTAINS NUMERICAL VALUES

```
df.describe()
```

	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

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

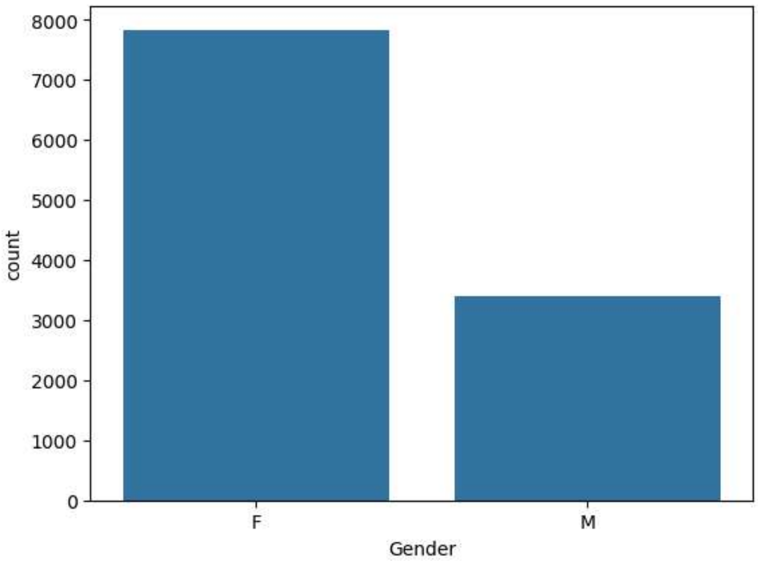
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Anlaysis

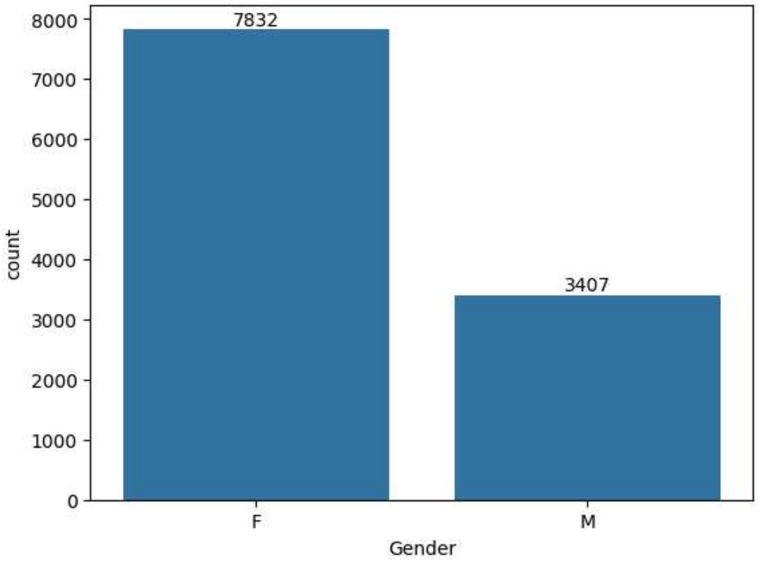
```
df.columns
```

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

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



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



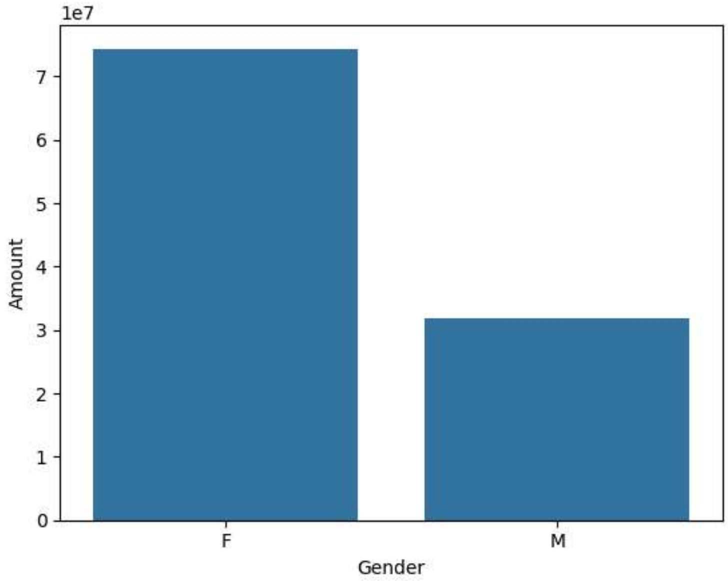
```
df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
```

	Gender	Amount
0	F	74335853
1	M	31913276

```
sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=Fals

sns.barplot(x='Gender' , y = 'Amount', data = sales_gen)
```

```
<Axes: xlabel='Gender', ylabel='Amount'>
```



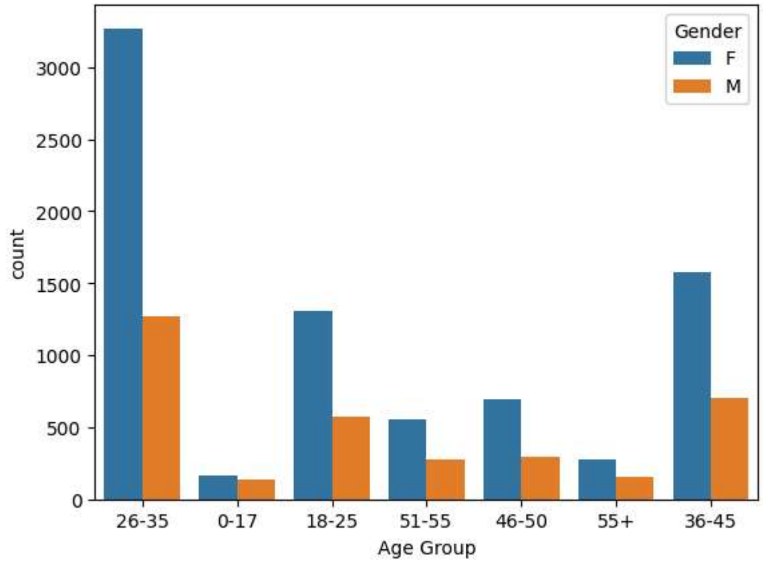
From above graphs we can see that most of the buyers are females and evne the purchasing power of females are greater than men

```
df.columns
```

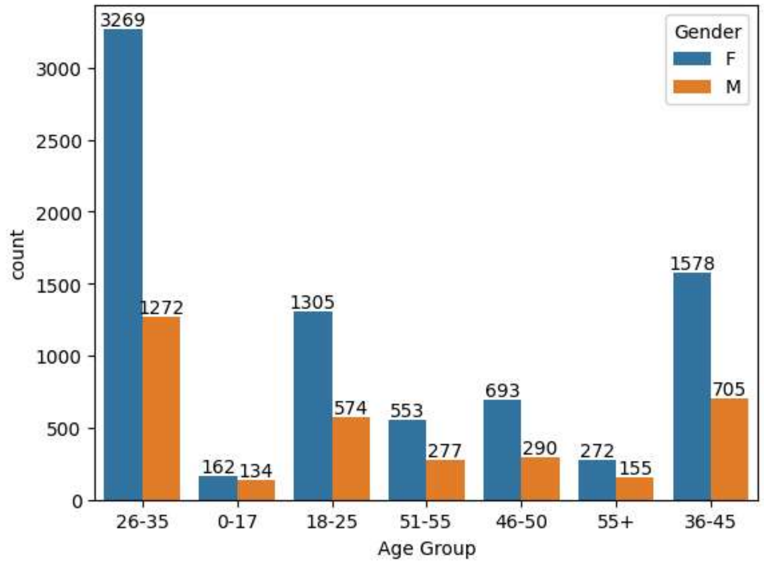
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
sns.countplot(data=df, x='Age Group', hue="Gender")
```

```
<Axes: xlabel='Age Group', ylabel='count'>
```

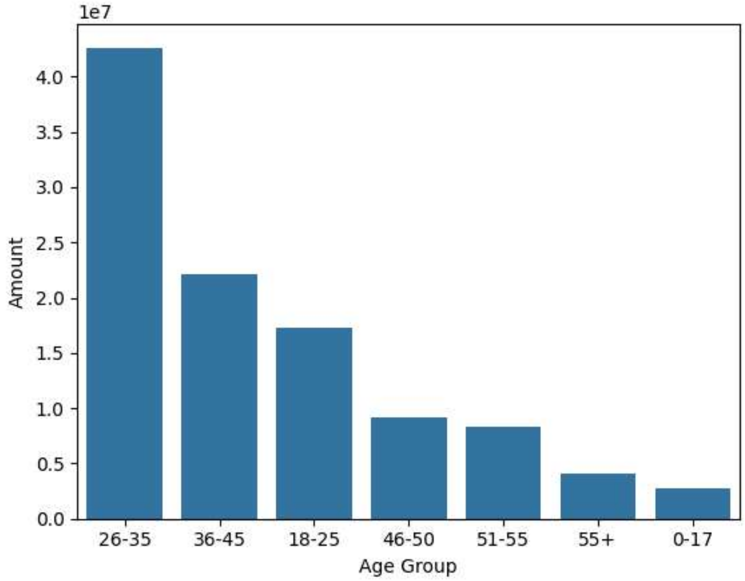


```
ax = sns.countplot(data=df, x='Age Group', hue="Gender")
for bars in ax.containers:
  ax.bar_label(bars)
```



```
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)
```

```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



```
Start coding or generate with AI.
```

From above graphs we can see thatmost of the buyers are of age group between 26-35 years females

```
df.columns
```

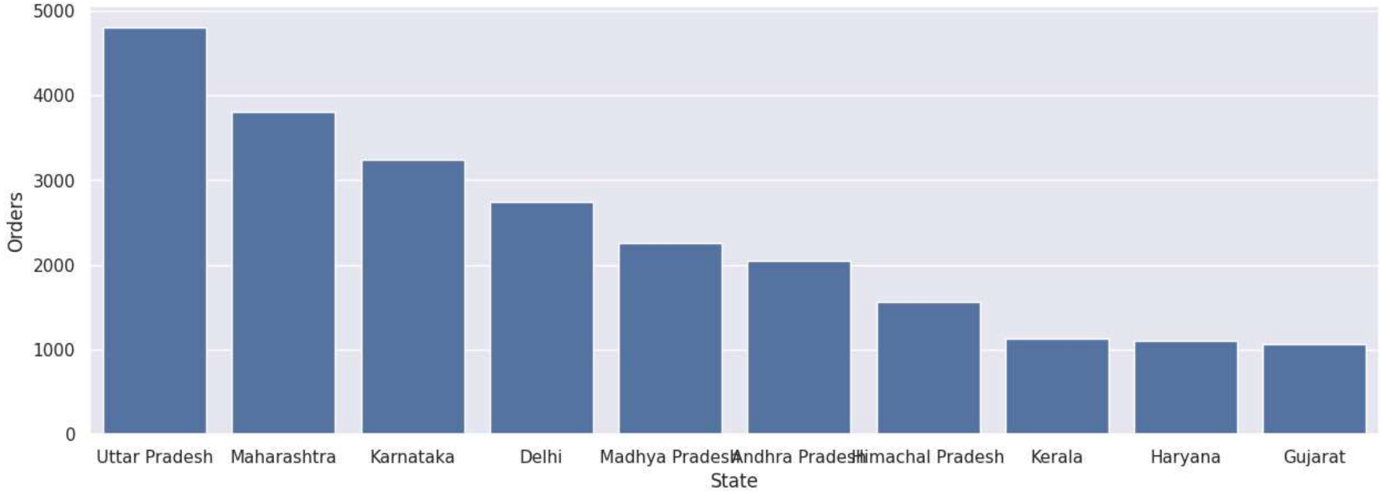
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

```
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=Fal

sns.set(rc={'figure.figsize': (15,5)})

sns.barplot(data = sales_state, x='State', y = 'Orders')
```

```
<Axes: xlabel='State', ylabel='Orders'>
```

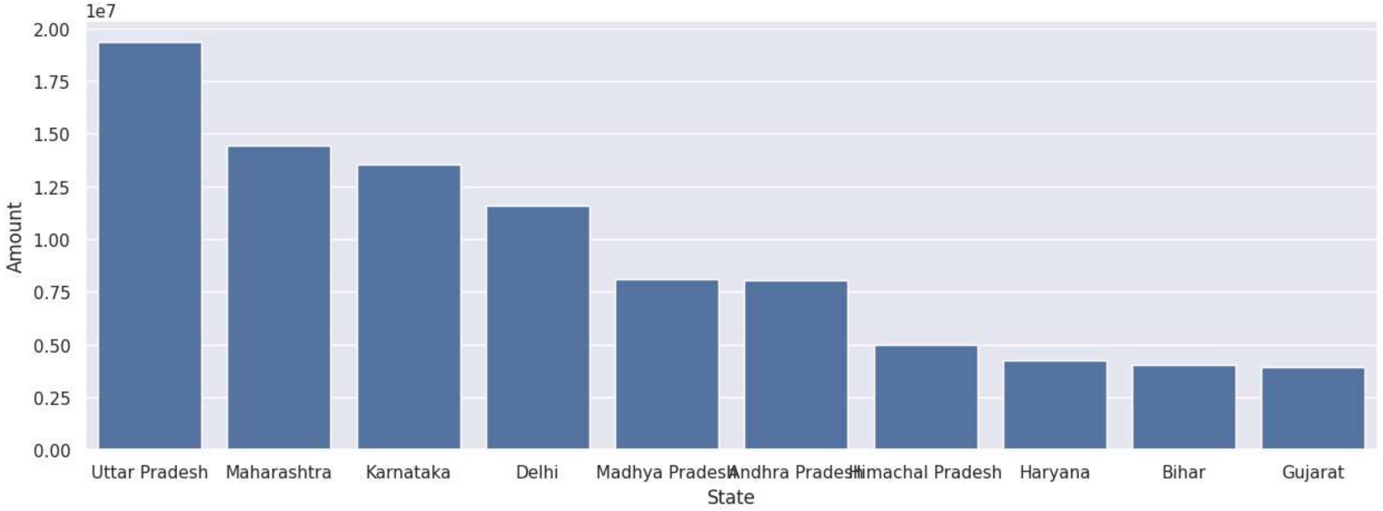


```
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=Fal

sns.set(rc={'figure.figsize': (15,5)})

sns.barplot(data = sales_state, x='State', y = 'Amount')
```

```
<Axes: xlabel='State', ylabel='Amount'>
```

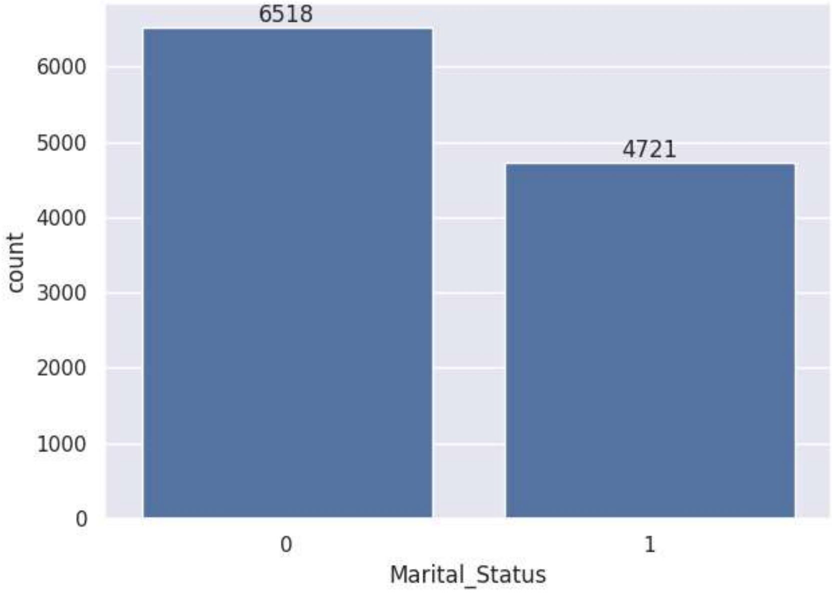


```
df.columns
```

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

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

```
ax = sns.countplot(data = df, x='Marital_Status')
sns.set(rc={'figure.figsize':(5,5)})
for bars in ax.containers:
  ax.bar_label(bars)
```

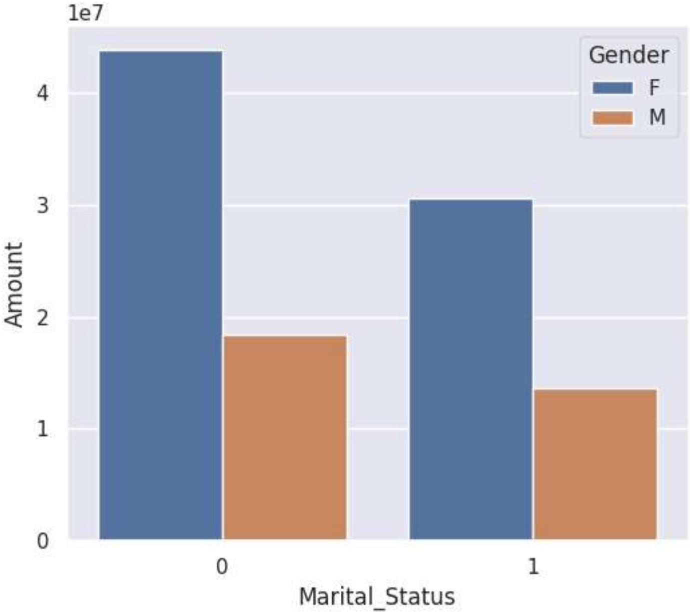


```
sales_state = df.groupby(['Marital_Status','Gender'], as_index=False)['Amount'].sum().sort_values(by='Amou

sns.set(rc={'figure.figsize': (6,5)})

sns.barplot(data = sales_state, x='Marital_Status', y = 'Amount', hue='Gender')
```

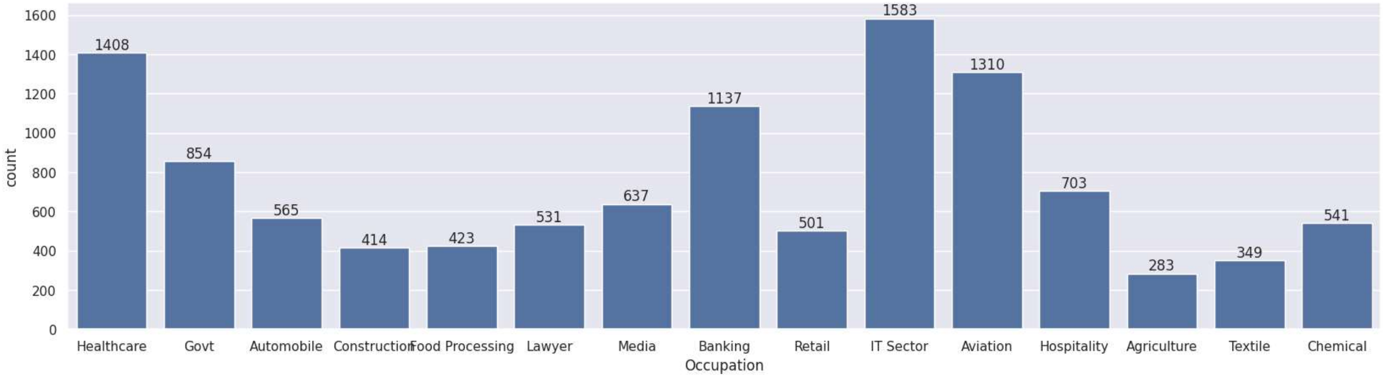
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



From above graphs we can see that most of the buyers are married women and they have hgih purchashing power

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

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

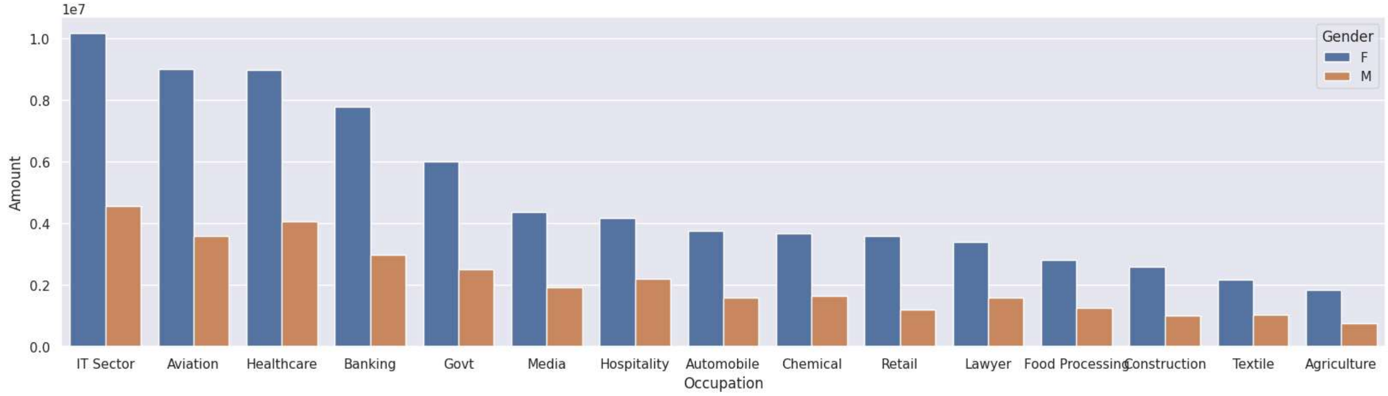


```
sales_state = df.groupby(['Occupation','Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount',

sns.set(rc={'figure.figsize': (20,5)})

sns.barplot(data = sales_state, x='Occupation', y = 'Amount', hue='Gender')
```

```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

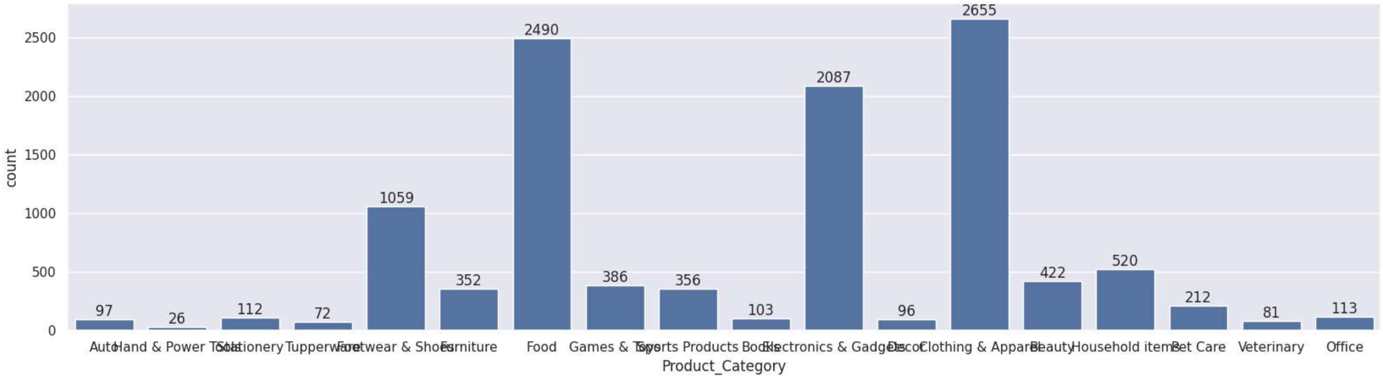


From the above graphs we cans ee that the most of the buyers are in IT sector, Healthcare and Aviation Sector

```
Start coding or generate with AI.
```

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

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

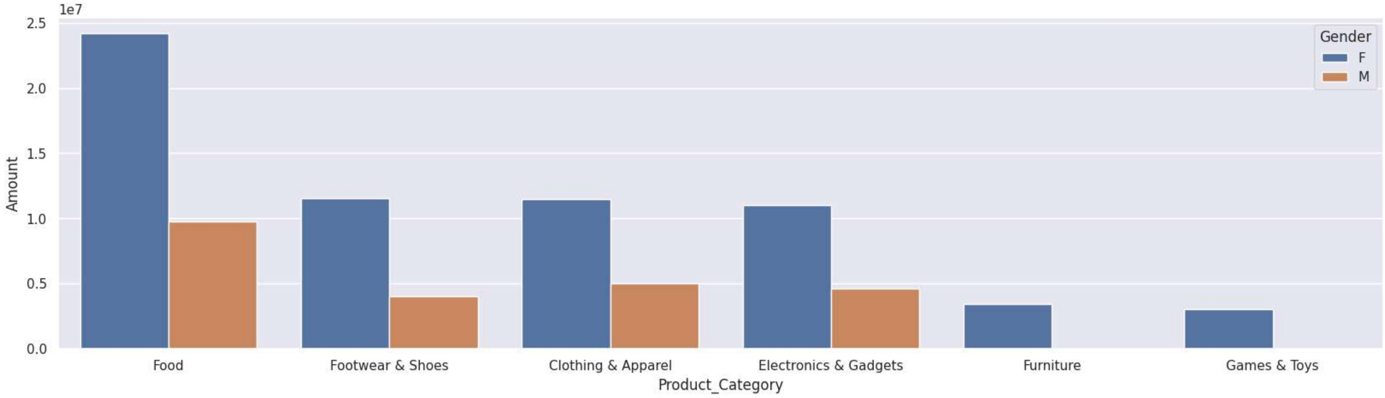


```
sales_state = df.groupby(['Product_Category','Gender'], as_index=False)['Amount'].sum().sort_values(by='Am

sns.set(rc={'figure.figsize': (20,5)})

sns.barplot(data = sales_state, x='Product_Category', y = 'Amount', hue='Gender')
```

```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```


1e7
Amount
Gender
F
M
Food
Footwear & Shoes
Clothing & Apparel
Electronics & Gadgets
Furniture
Games & Toys
Product_Category