DIWALI SALES DATA ANALYSIS

Import Libraries

In [7]:
```
# Import Python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [9]:
```
df= pd.read_csv('Diwali Sales Data.csv',encoding='unicode_escape')
df
```

Out[9]:

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

11251 rows × 15 columns

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

Out[3]: (11251, 15)

In [4]:
```
df.head(10)
```

Out[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	W
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	So
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	C
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	So
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W
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	W
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	C
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	So

In [5]:
```
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 [6]:
```
#drop Unreleated/blank columns
df.drop(['Status','unnamed1'], axis=1, inplace=True)
```

In [7]:
```
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 [8]:
```
pd.isnull(df)
```

Out[8]:

	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 [9]:
```
pd.isnull(df).sum()
```

Out[9]:
```
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 [10]:
```
df.shape
```

Out[10]:
```
(11251, 13)
```

In [11]:
```
#drop null values
df.dropna(inplace=True)
```

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

In [14]:
```
df['Amount'].dtypes
```

Out[14]:
```
dtype('int32')
```

In [15]:
```
df.columns
```

Out[15]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [16]:
```
#rename column
df.rename(columns={'Marital_Status':'Shaadi'})
```

Out[16]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Z
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Wes
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	Wes
...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Wes
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	Wes

11239 rows × 13 columns

In [17]:
```
df.describe()
```

Out[17]:

	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 [18]:
```
df[['Age','Orders','Amount']].describe()
```

Out[18]:

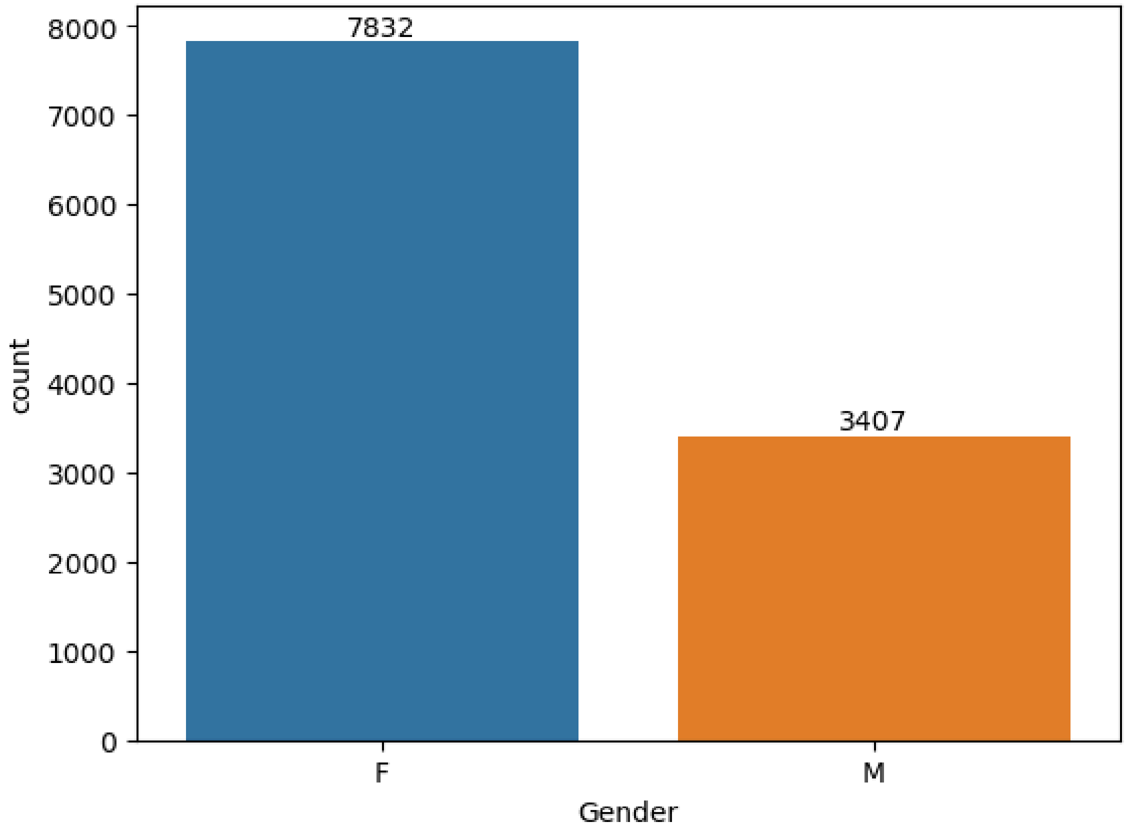
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis

Gender

In [19]:
```
ax= sns.countplot(x='Gender',data=df)

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



Note :- in that graph show female purchase more than male

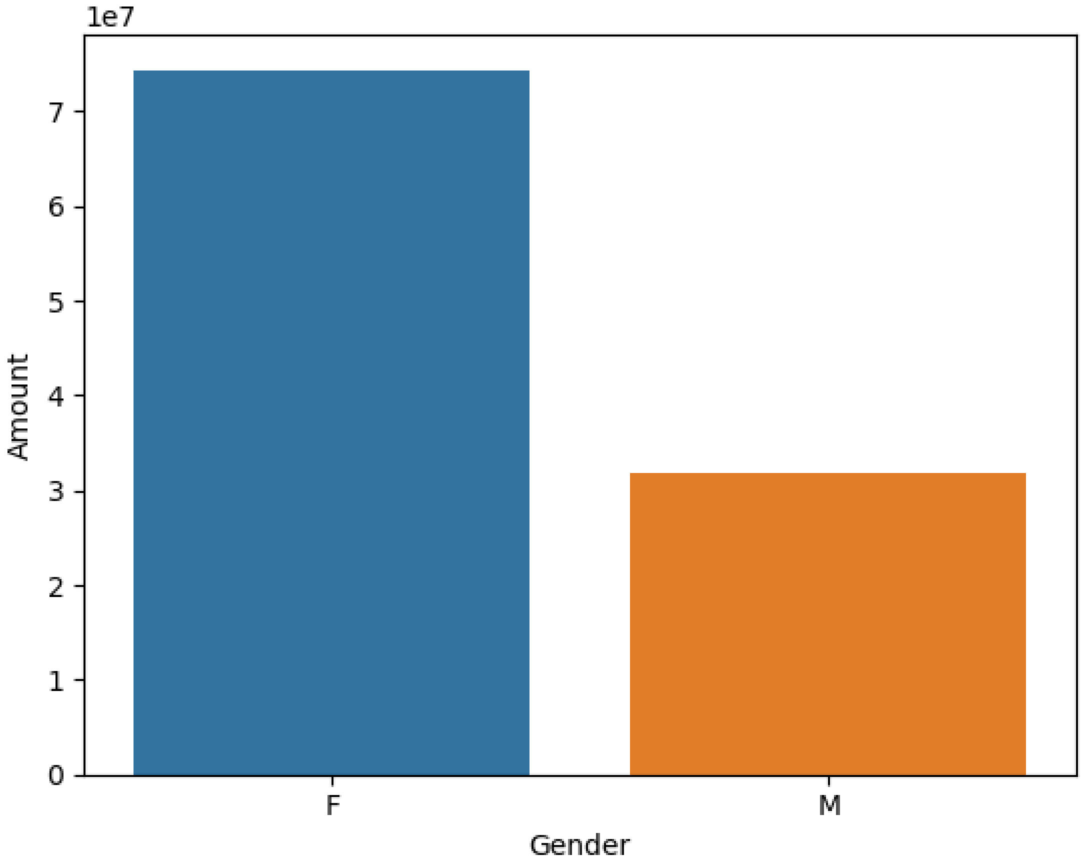
In [20]:
```
df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount
```

Out[20]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [21]:
```
sales_gen=df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values
```

In [22]:
```
sns.barplot(data=sales_gen, x='Gender', y='Amount')
plt.show()
```



Note: From the above graph we can see that most of the buyers are females and even the purchasing power of female are greater than men

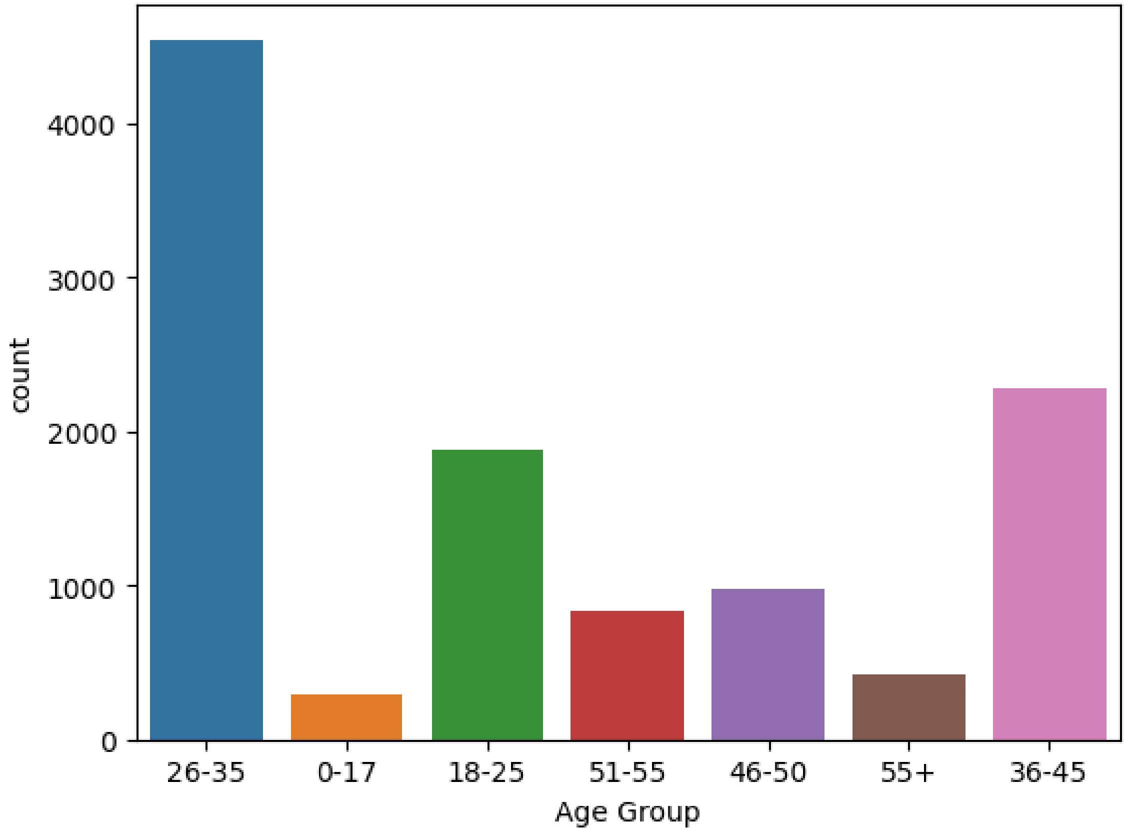
Age

In [23]:
```
df.columns
```

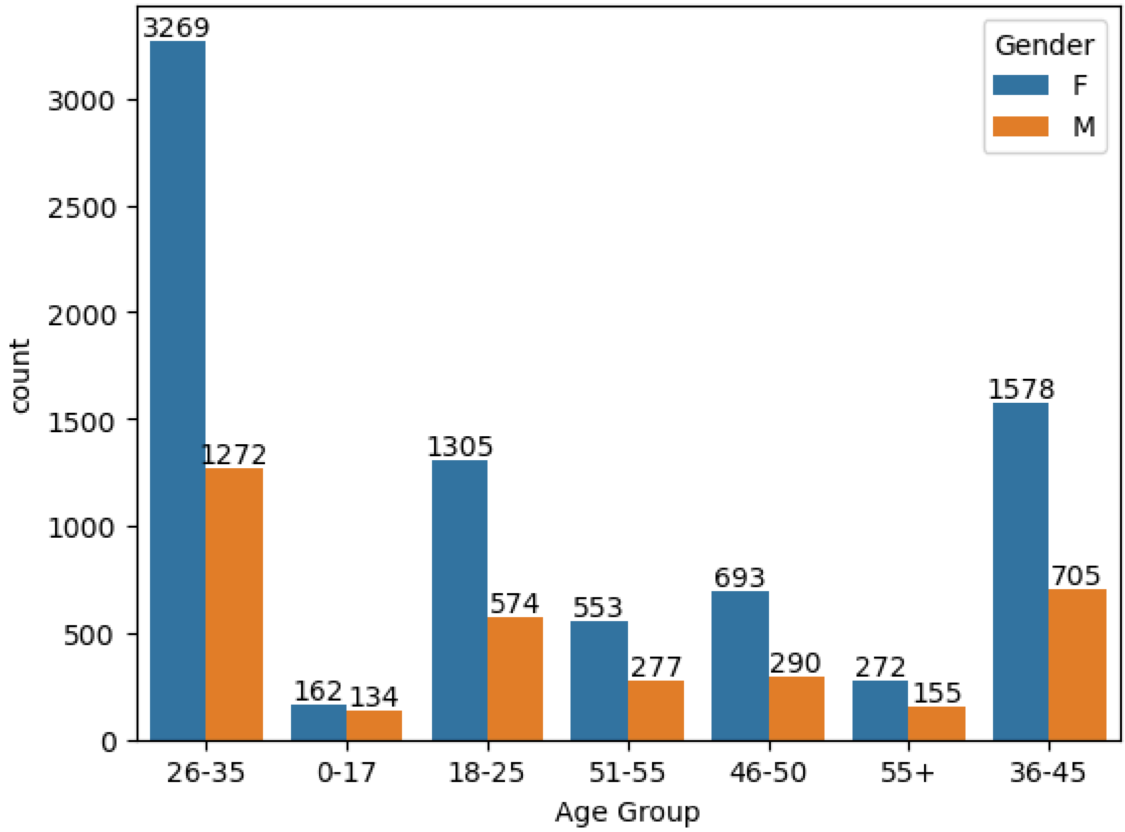
Out[23]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [24]:
```
sns.countplot(data=df,x='Age Group')
```

Out[24]: <Axes: xlabel='Age Group', ylabel='count'>



In [25]:
```
ax2=sns.countplot(data=df,x='Age Group', hue='Gender')
for bars in ax2.containers:
    ax2.bar_label(bars)
```



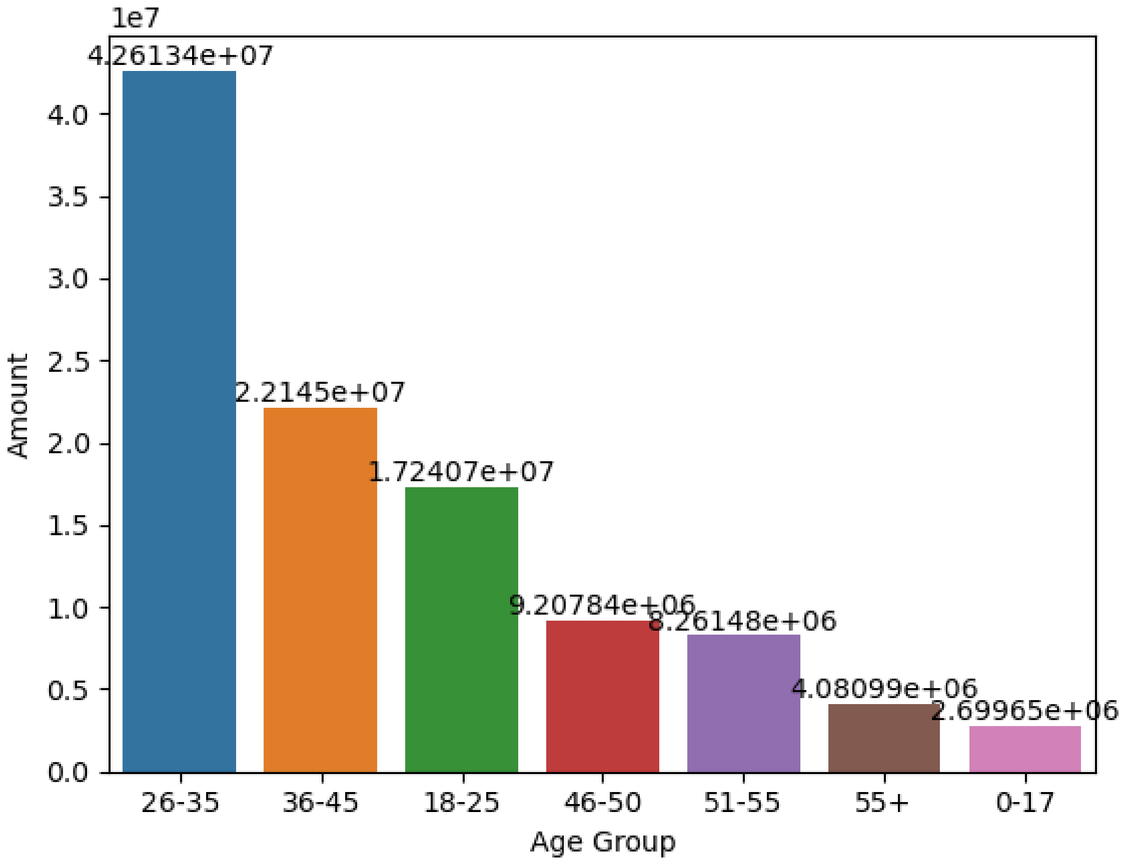
In [26]:
```
# Total Amount vs Age Group
sales_age=df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_val
```

In [27]:
```
sales_age
```

Out[27]:

	Age Group	Amount
2	26-35	42613442
3	36-45	22144994
1	18-25	17240732
4	46-50	9207844
5	51-55	8261477
6	55+	4080987
0	0-17	2699653

In [28]:
```
ax3= sns.barplot(data=sales_age, x='Age Group' , y='Amount')
for bars in ax3.containers:
    ax3.bar_label(bars)
```



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

State

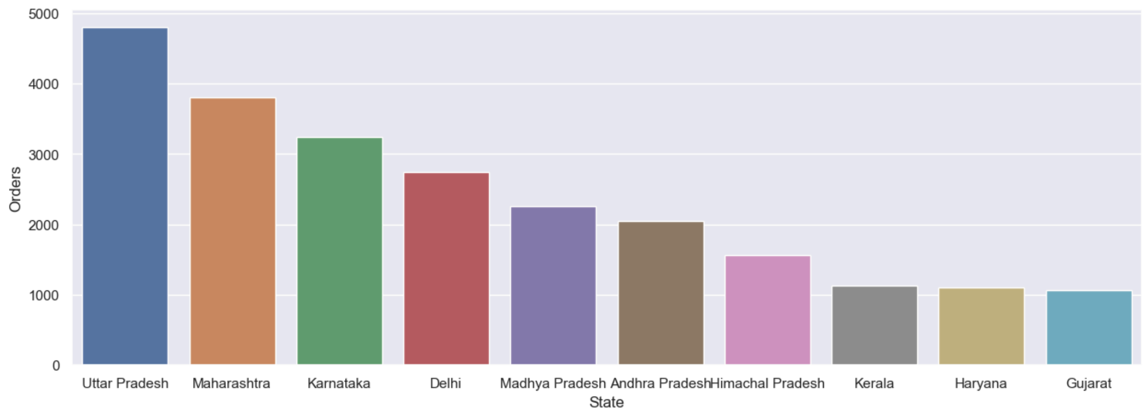
In [29]:
```
df.columns
```

Out[29]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [30]:
```
# Number of order from  10 state
order_state= df.groupby(['State'], as_index=False)['Orders'].sum().sort_valu

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

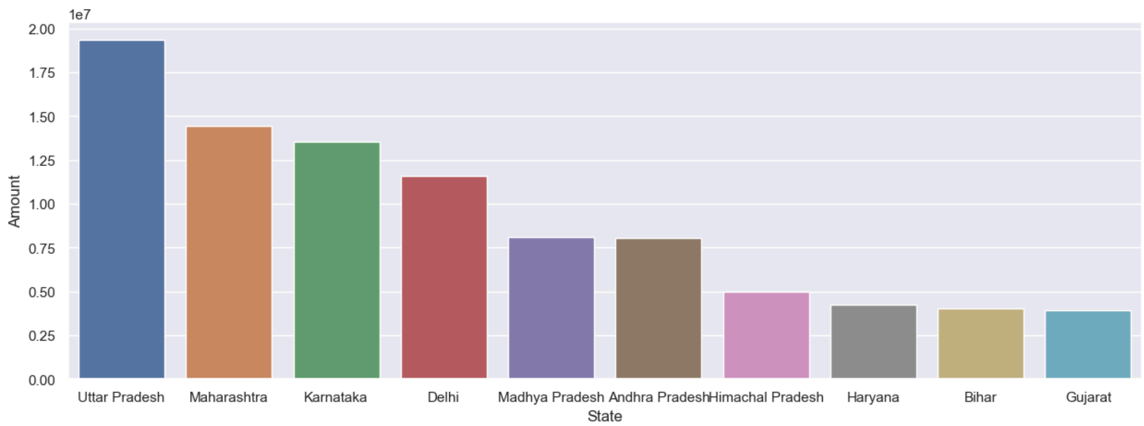
Out[30]: <Axes: xlabel='State', ylabel='Orders'>



In [31]:
```
# Total amount of sales from 10 state
sales_am=df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(b

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

Out[31]: <Axes: xlabel='State', ylabel='Amount'>



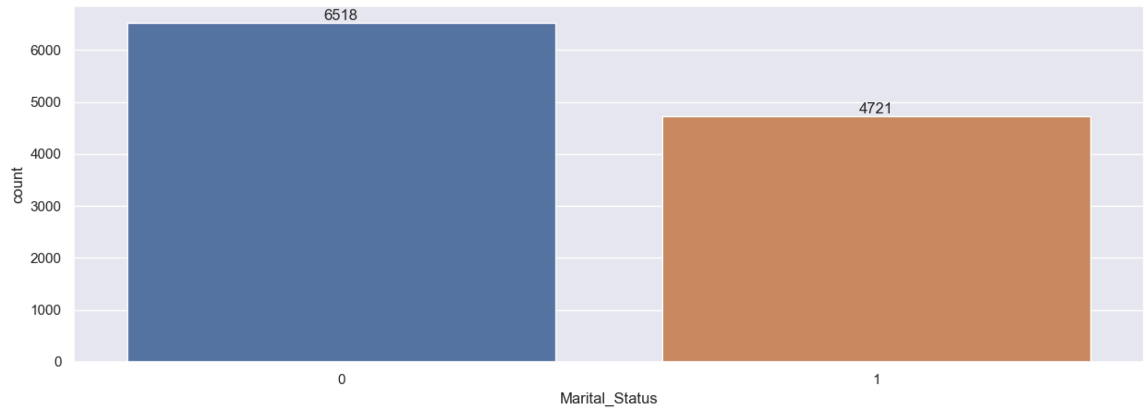
From the above Graph, we can see that most of the orders are from Uttar Pradesh, Maharashtra, and Karnataka respectively but total

sales/Amount is the UP, Karnataka, and the Maharashtra

Marital_Status

In [ ]:

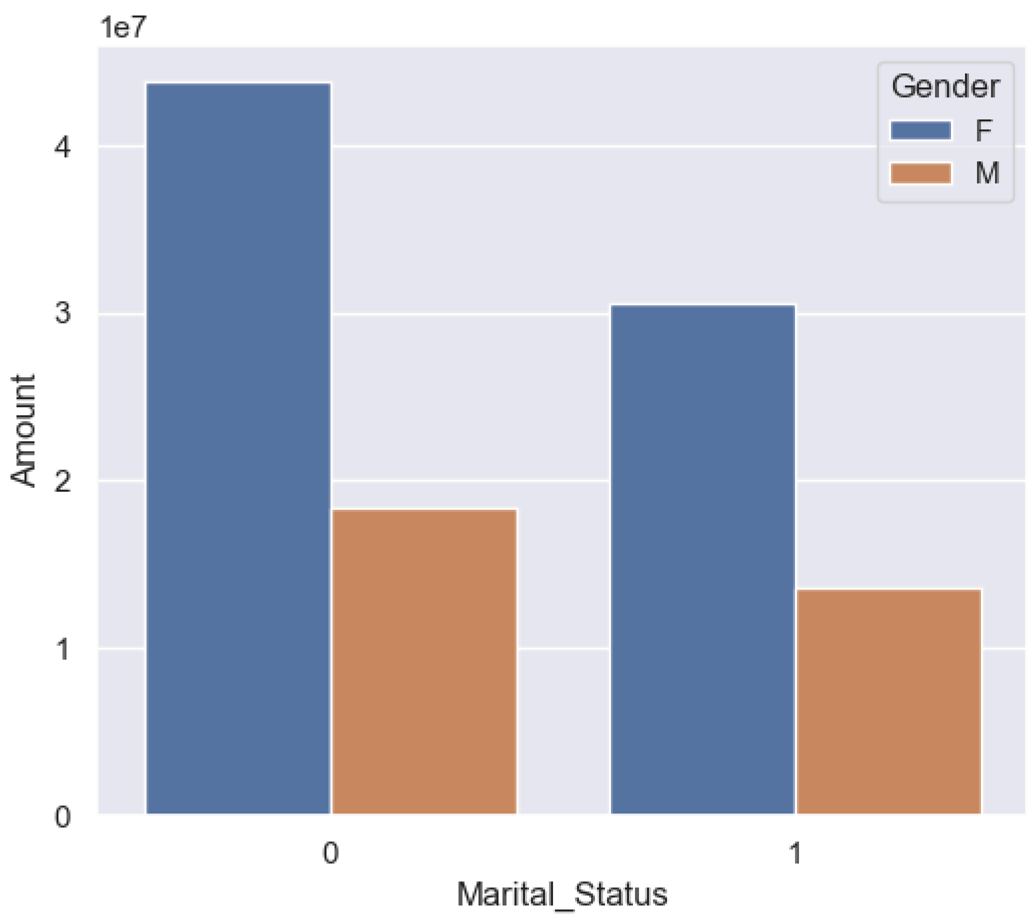
In [32]:
```
ax=sns.countplot(data=df, x='Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [33]:
```
sales=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum()

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

Out[33]: <Axes: xlabel='Marital_Status', ylabel='Amount'>

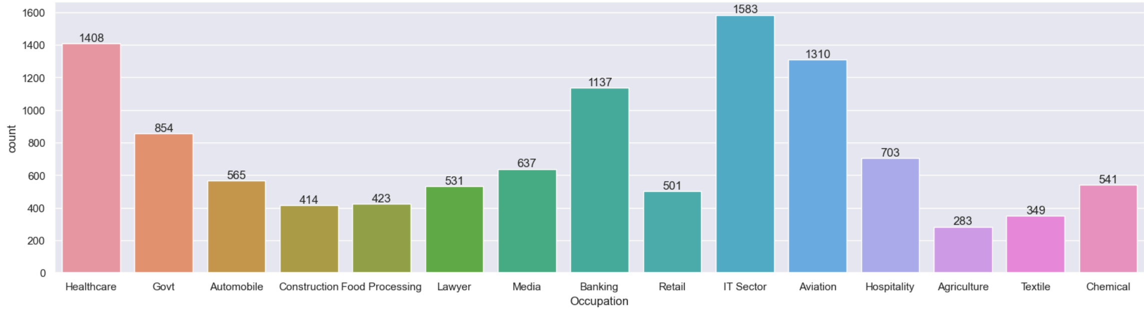


from above graph we can see that most of the buyers are married women and they purchasing power

Occupation

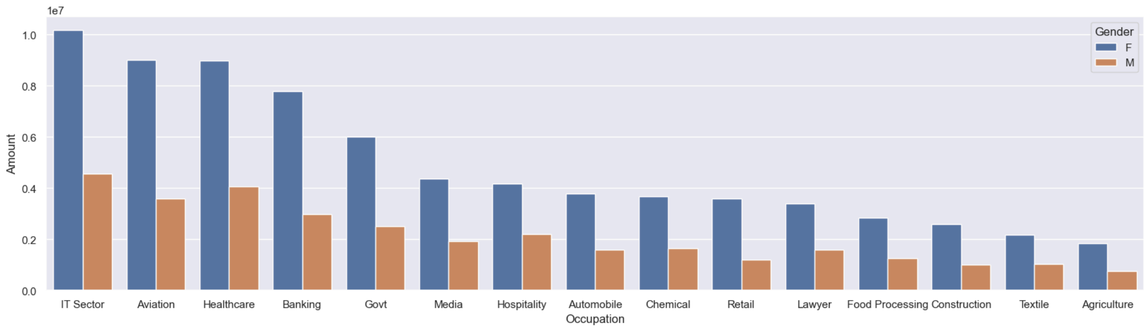
In [34]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(data=df,x='Occupation')

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



In [35]:
```
sales=df.groupby(['Occupation','Gender'], as_index=False)['Amount'].sum().s
sns.barplot(data=sales, x='Occupation', y='Amount', hue='Gender')
```

Out[35]: <Axes: xlabel='Occupation', ylabel='Amount'>



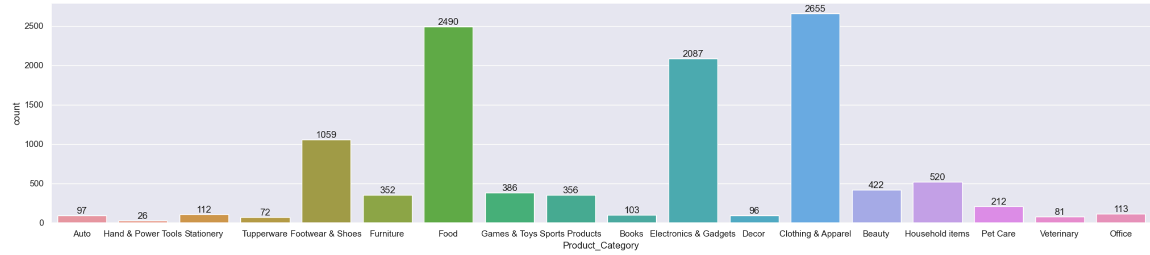
From above graph we can see that most of the buyers are working in IT, Aviation and Healthcare sector

In [36]:
```
df.columns
```

Out[36]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

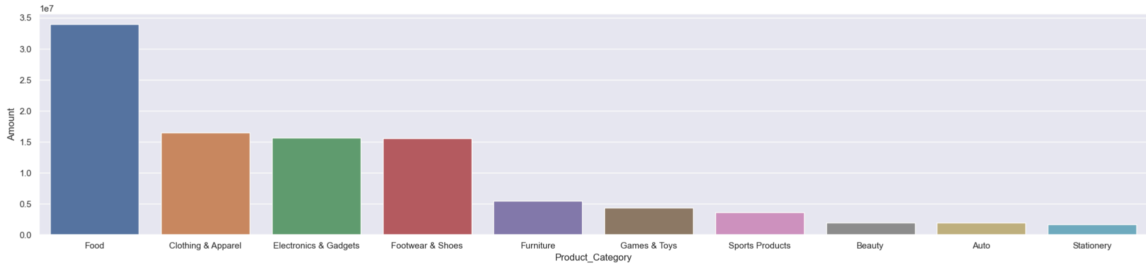
In [37]:
```
sns.set(rc={'figure.figsize':(25,5)})
ax5=sns.countplot(data=df, x='Product_Category')

for bars in ax5.containers:
    ax5.bar_label(bars)
```



In [38]:
```
sales= df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_
sns.barplot(data=sales, x='Product_Category', y='Amount')
```

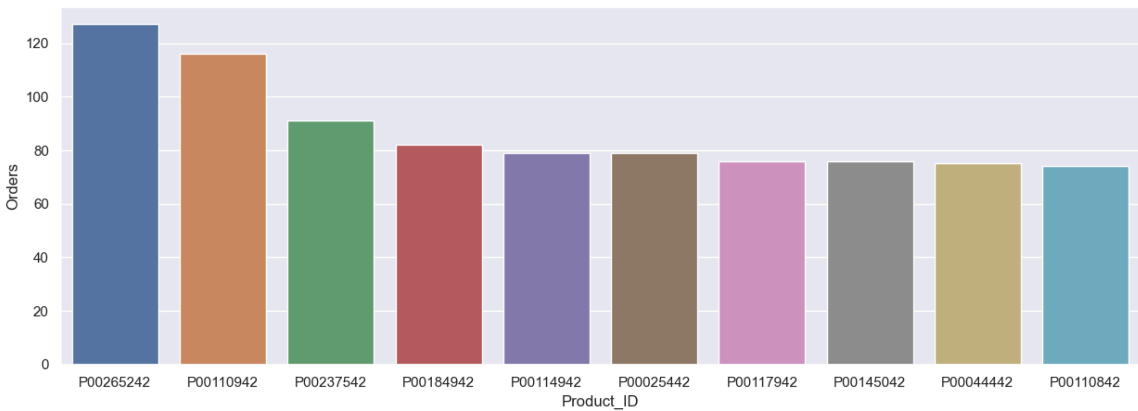
Out[38]: <Axes: xlabel='Product_Category', ylabel='Amount'>



From above graph we can see that most of the sold products are from food,clothing and electronic category

In [39]:
```
sales_state =df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state, x='Product_ID', y='Orders')
```

Out[39]: <Axes: xlabel='Product_ID', ylabel='Orders'>



Conclusion:

The Diwali sales data analysis reveals a significant increase in consumer spending, with electronics and apparel sectors showing the highest growth rates. Online shopping platforms experienced a notable surge, driven by exclusive discounts and the convenience of home delivery. Additionally, regional trends indicated a higher expenditure in urban areas compared to rural regions. The overall positive sales performance underscores the festive season's critical role in boosting economic activity, highlighting the effectiveness of targeted marketing strategies and promotional campaigns. This analysis provides valuable insights

In [ ]:

```

```