Diwali Sales Analysis Report

Presented by Picnic Rautaray

In [48]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [65]:
```
df=pd.read_csv('Diwali Sales Data.csv',encoding='unicode_escape')
```

In [66]:
```
print("Number of Rows:",df.shape[0])
print("Number of Columns:",df.shape[1])
```

```
Number of Rows: 11251
Number of Columns: 15
```

In [51]:
```
df.head(10)
```

Out[51]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	M
1	1000732	Kartik	P00110942	F	26-35	35	1	Andh
2	1001990	Bindu	P00118542	F	26-35	35	1	Ut
3	1001425	Sudevi	P00237842	M	0-17	16	0	
4	1000588	Joni	P00057942	M	26-35	28	1	
5	1000588	Joni	P00057942	M	26-35	28	1	
6	1001132	Balk	P00018042	F	18-25	25	1	Ut
7	1002092	Shivangi	P00273442	F	55+	61	0	M
8	1003224	Kushal	P00205642	M	26-35	35	0	Ut
9	1003650	Ginny	P00031142	F	26-35	26	1	Andh

In [52]:
```
df.tail(10)
```

Out[52]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status
11241	1003032	Matthias	P00058042	F	26-35	33	0
11242	1004344	Hildebrand	P00185442	F	26-35	27	1
11243	1005446	Sheetal	P00297742	M	51-55	53	0
11244	1005446	Sheetal	P00297742	M	51-55	53	0
11245	1004140	Bertelson	P00057442	F	26-35	31	1
11246	1000695	Manning	P00296942	M	18-25	19	1
11247	1004089	Reichenbach	P00171342	M	26-35	33	0
11248	1001209	Oshin	P00201342	F	36-45	40	0
11249	1004023	Noonan	P00059442	M	36-45	37	0
11250	1002744	Brumley	P00281742	F	18-25	19	0

In [67]:
```
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 [68]:
```
df.isnull().sum()
```

Out[68]:
```
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
Status             11251
unnamed1           11251
dtype: int64
```

In [69]:
```
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

In [70]:
```
df.isnull().sum()
```

Out[70]:
```
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 [71]:
```
df.dropna(inplace=True)
```

In [72]:
```
df.isnull().sum()
```

Out[72]:
```
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 [73]:
```
df.dtypes
```

Out[73]:
```
User_ID              int64
Cust_name           object
Product_ID          object
Gender              object
Age Group           object
Age                  int64
Marital_Status       int64
State               object
Zone                object
Occupation          object
Product_Category    object
Orders               int64
Amount             float64
dtype: object
```

In [74]:
```
df['Amount']=df['Amount'].astype('int')
```

In [75]:
```
df.dtypes
```

Out[75]:
```
User_ID              int64
Cust_name           object
Product_ID          object
Gender              object
Age Group           object
Age                  int64
Marital_Status       int64
State               object
Zone                object
Occupation          object
Product_Category    object
Orders               int64
Amount               int32
dtype: object
```

In [76]:
```
list[df.columns]
```

Out[76]:
```
list[Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'A
ge',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')]
```

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

Out[77]:

	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 [78]:
```
df[['Age','Orders','Amount']].describe()
```

Out[78]:

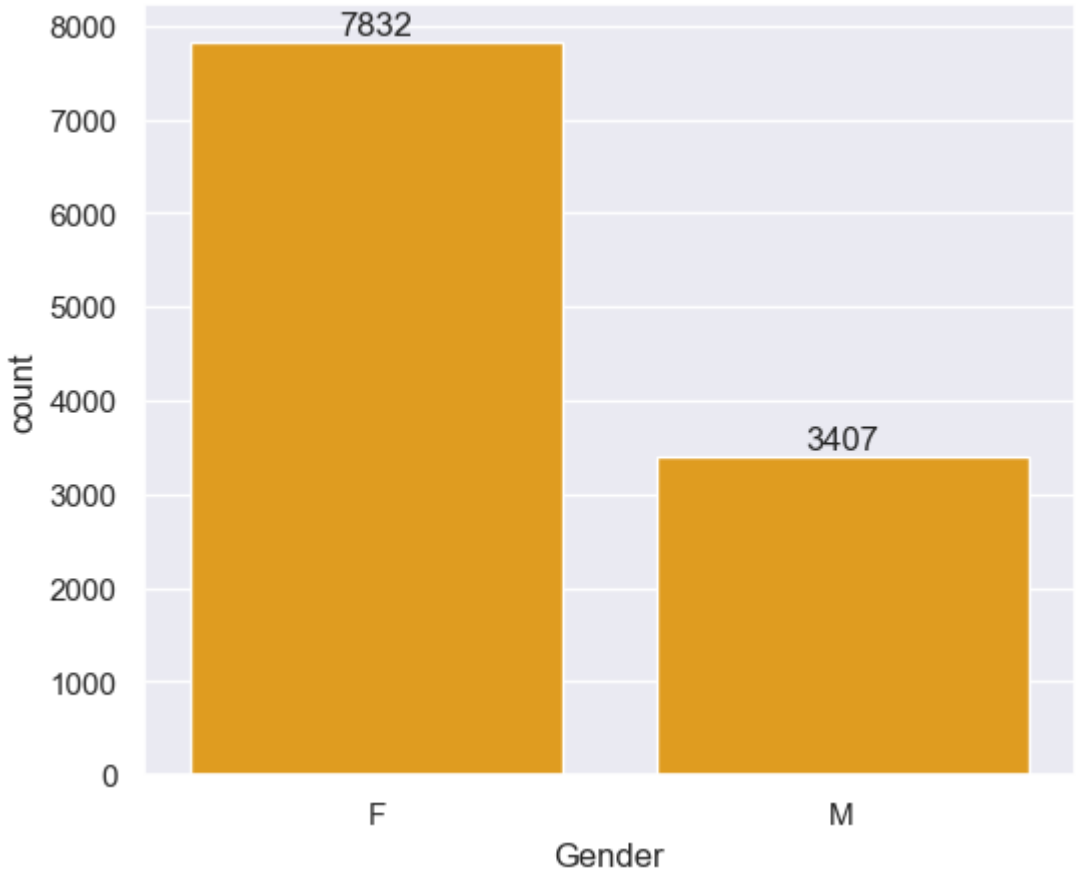
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis

Gender

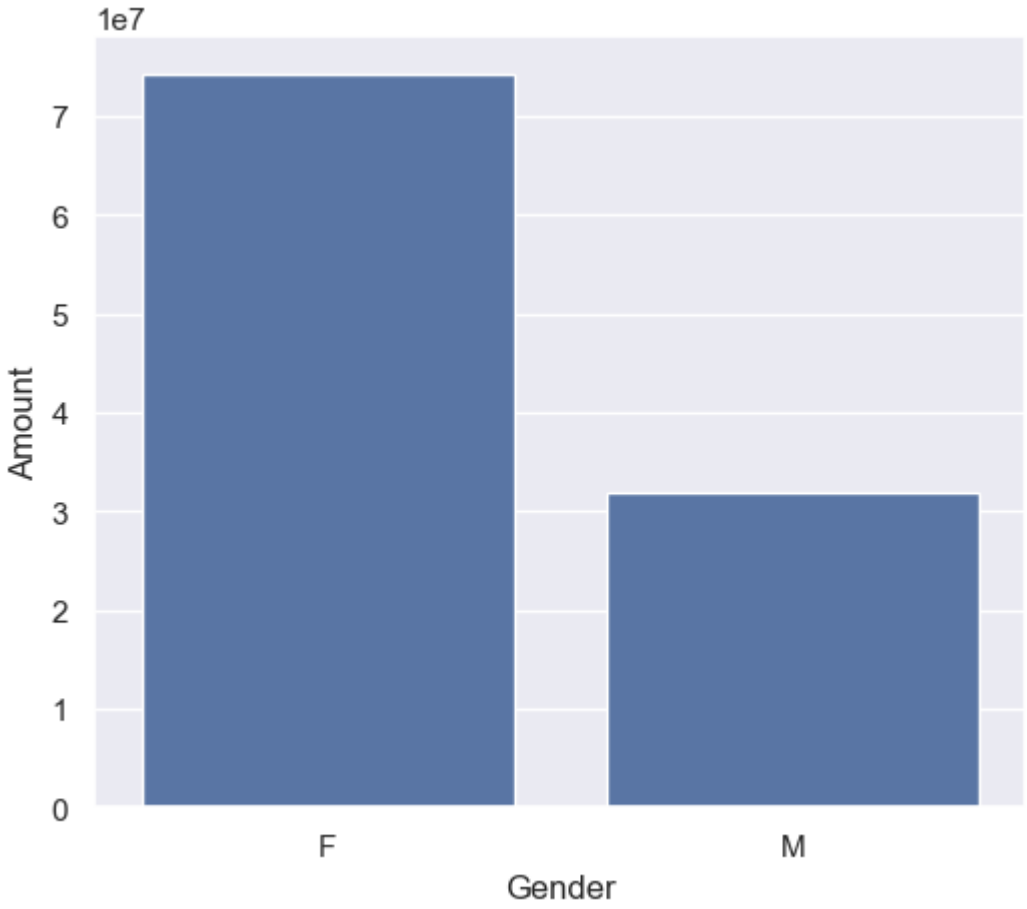
In [79]:
```
ax = sns.countplot(x='Gender',color = 'Orange',data = df)

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



In [80]:
```
sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_value
sns.barplot(x='Gender',y= 'Amount',data= sales_gen)
```

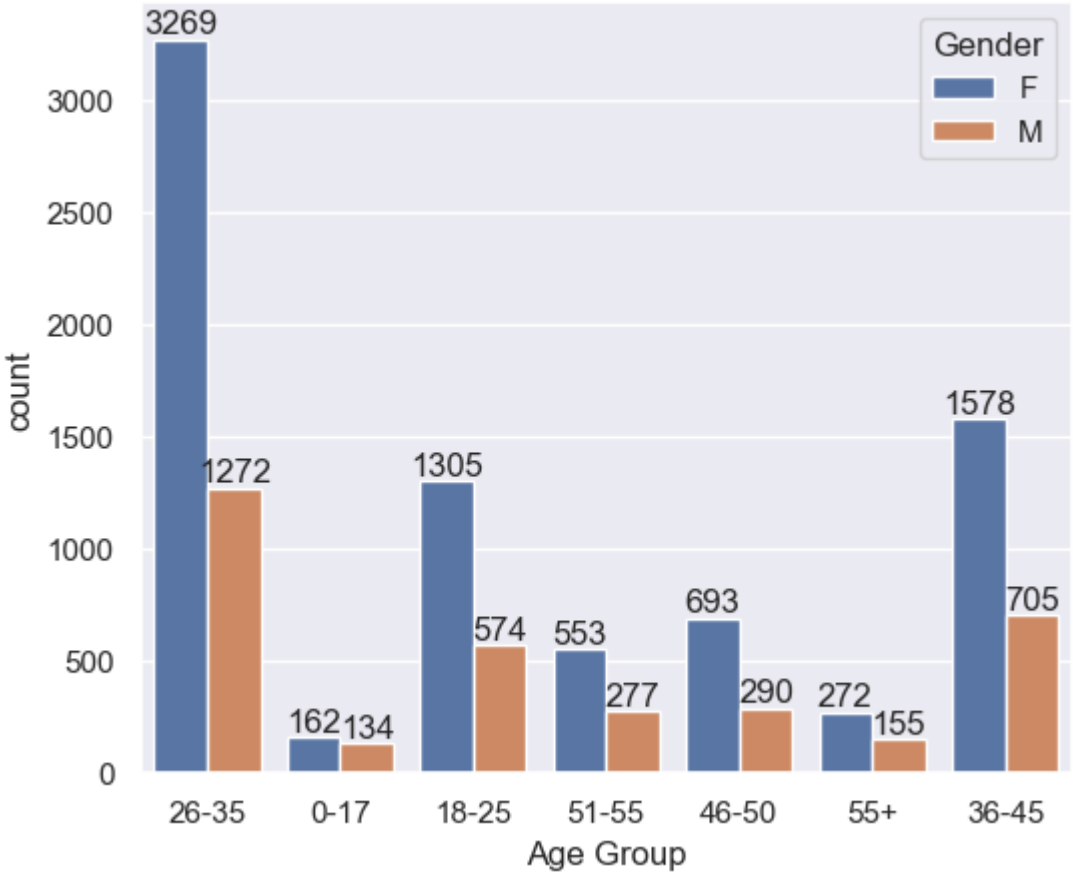
Out[80]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



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

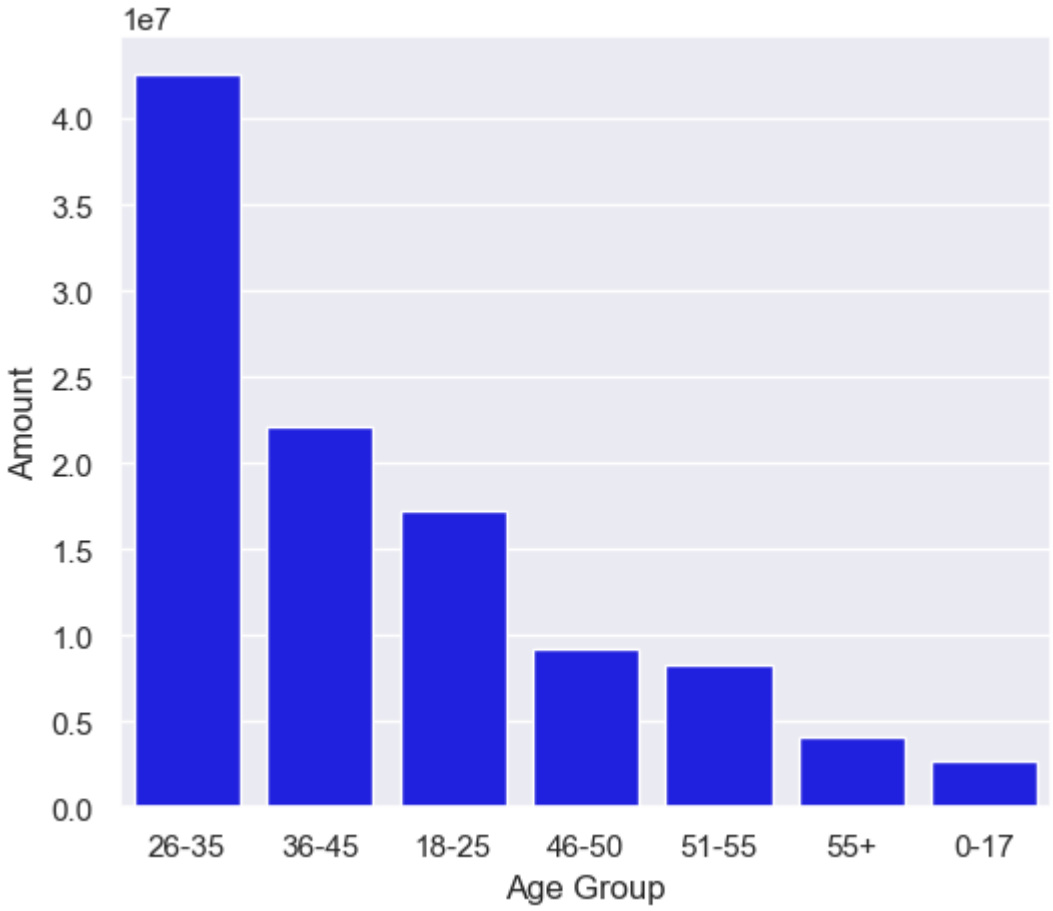
Age

In [81]:
```
ax = sns.countplot(data=df,x='Age Group',hue = 'Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [82]:
```
sales_age = df.groupby(['Age Group'],as_index = False)['Amount'].sum().sort_
sns.barplot(x='Age Group',y = 'Amount',data=sales_age,color='Blue')
```

Out[82]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



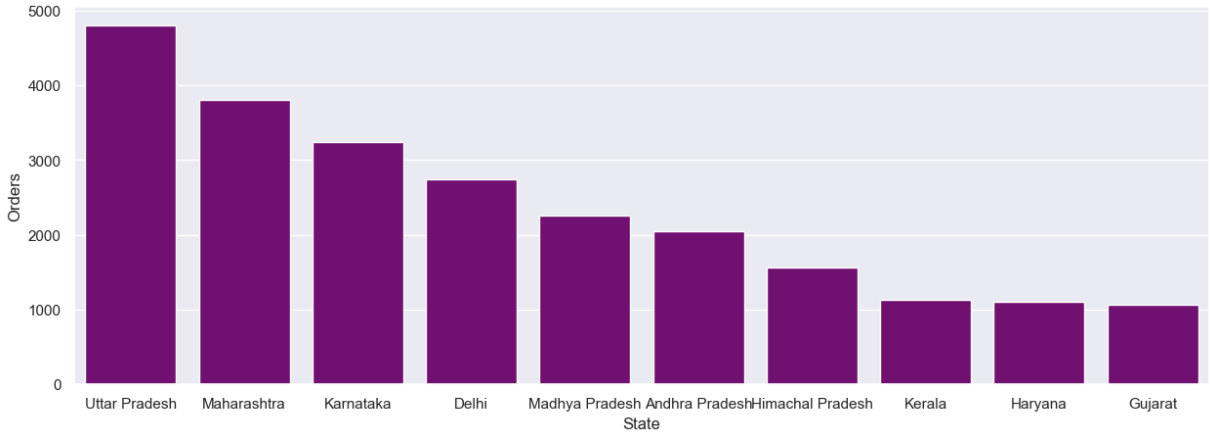
From above graphs we can see that most of the buyers are age group between 26-35 ys female

State

Total number of orders from top 10 States

In [83]:
```
sales_state = df.groupby(['State'],as_index=False)['Orders'].sum().sort_valu
sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state,x='State',y='Orders',color='Purple')
```

Out[83]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



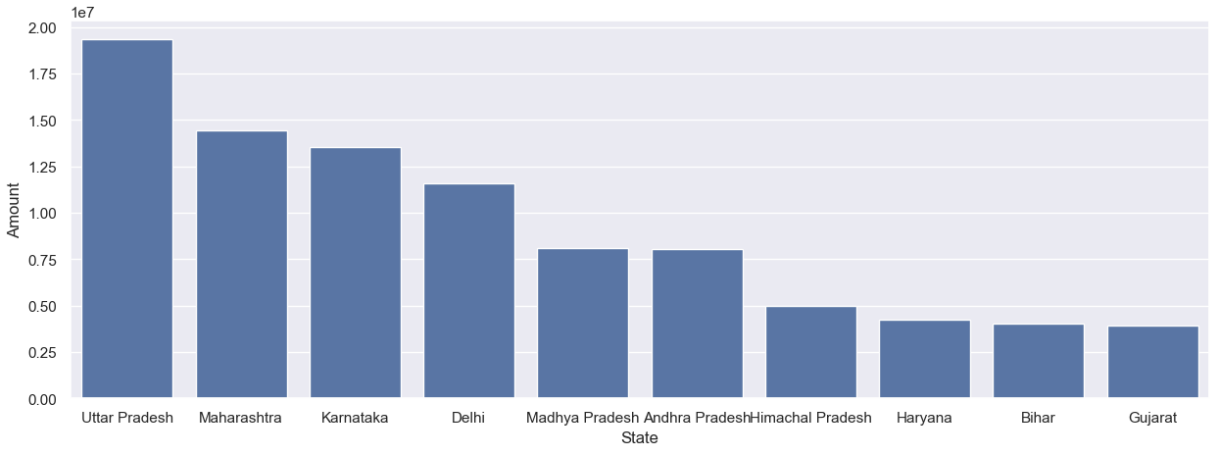
Total amount/sales from the top 10 states

In [84]:
```
sales_state = df.groupby('State')['Amount'].sum().sort_values(ascending=Fals

# Importing seaborn as sns if not already imported
import seaborn as sns

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

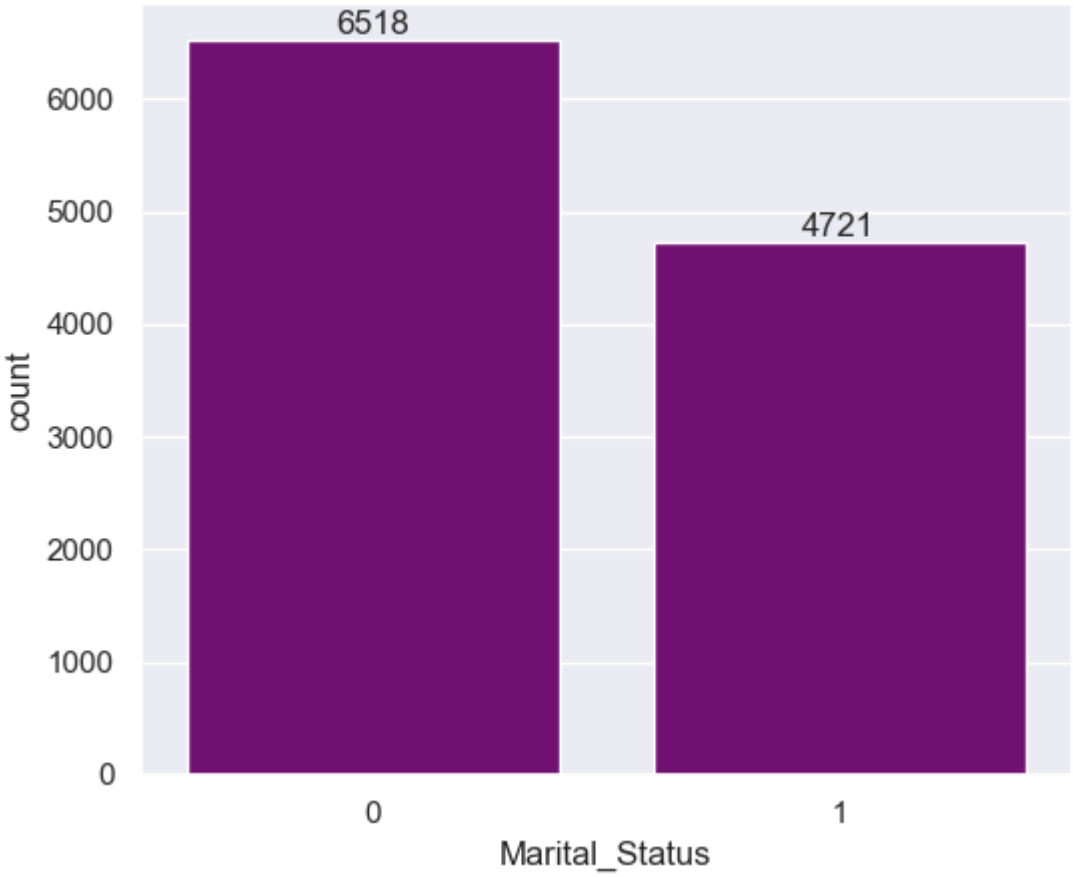
Out[84]: <Axes: xlabel='State', ylabel='Amount'>



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

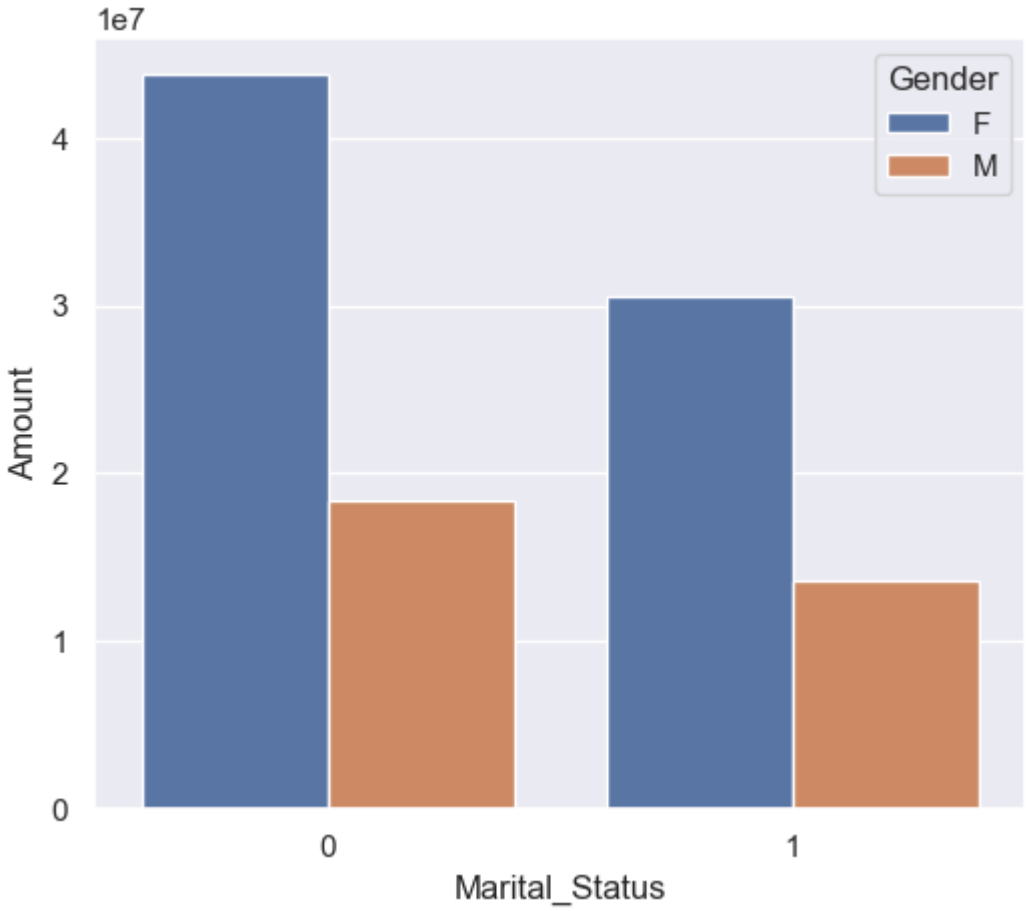
Marital Status

In [92]:
```
ax = sns.countplot(data = df,x = 'Marital_Status',color='Purple')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [86]:
```
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[86]: <Axes: xlabel='Marital_Status', ylabel='Amount'>

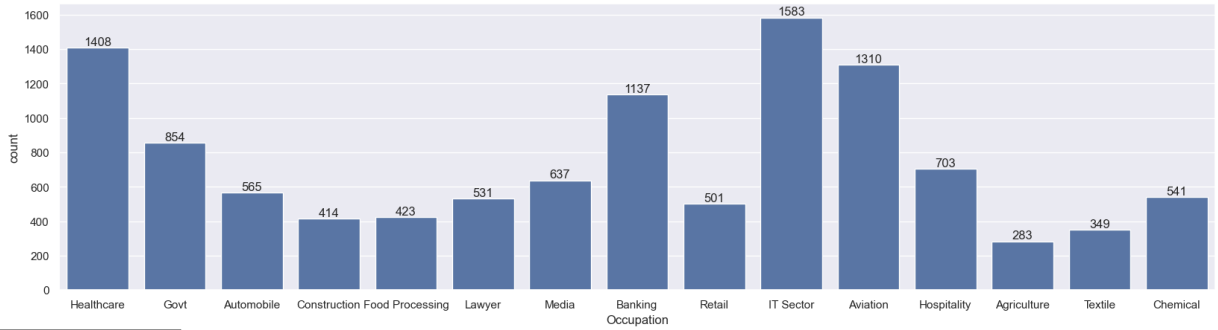


Occuption

In [87]:
```
df.columns
```

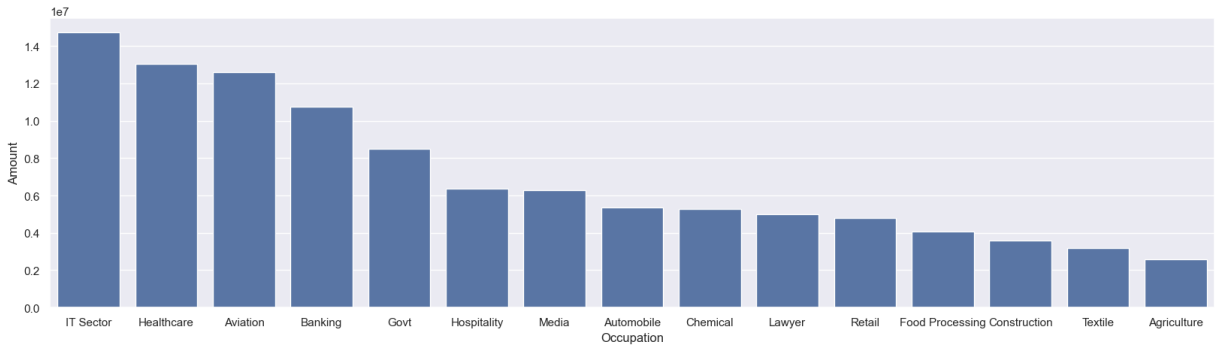
Out[87]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [90]:
```
ax = sns.countplot(data=df,x = 'Occupation')
sns.set(rc={'figure.figsize':(25,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [96]:
```
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')
```

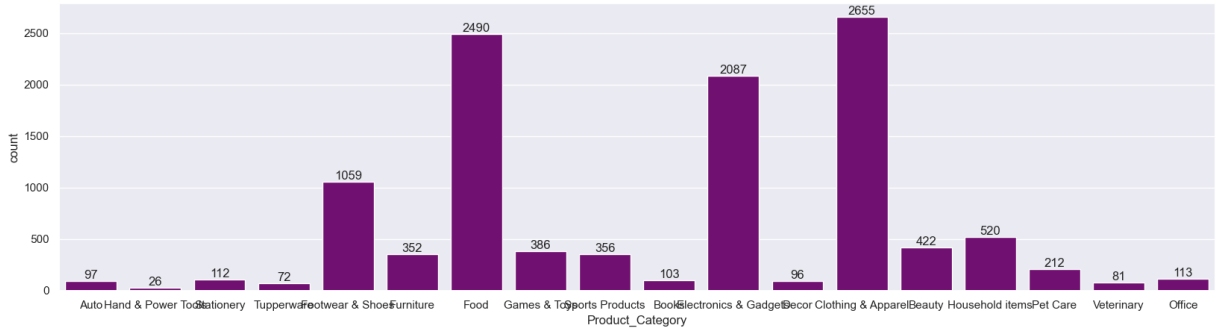
Out[96]: <Axes: xlabel='Occupation', ylabel='Amount'>



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

Product Category

In [102…
```
ax = sns.countplot(x = 'Product_Category',data = df,color = 'purple')
sns.set(rc={'figure.figsize':(20,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```

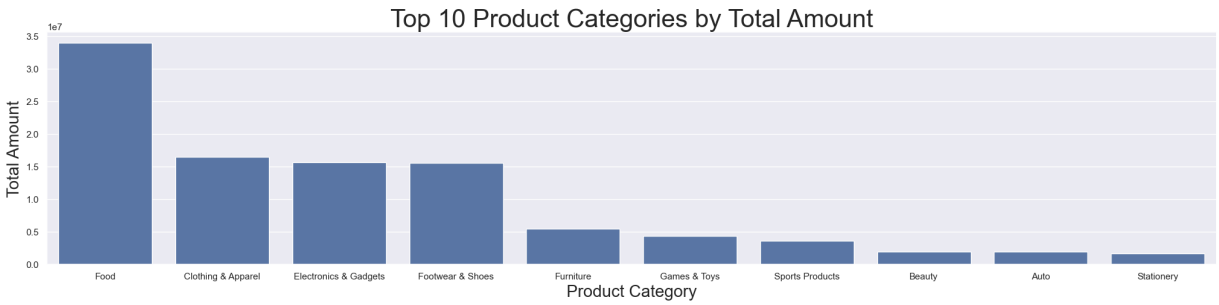


In [112…
```
sales_state = df.groupby(['Product_Category'],as_index=False)['Amount'].sum(
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data= sales_state,x = 'Product_Category',y = 'Amount')

plt.xlabel('Product Category', fontsize=20)
plt.ylabel('Total Amount', fontsize=20)

plt.title('Top 10 Product Categories by Total Amount', fontsize=30)

plt.show()
```

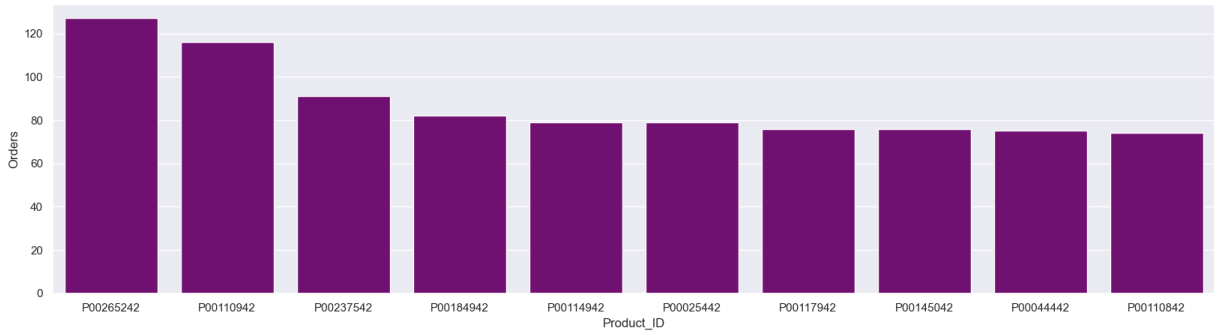


From above graphs we can see thatmost of the sold products are from food,clothing,electronics categories.

Total Orders per ProductID

In [115…
```
Ord_per_product = df.groupby(['Product_ID'],as_index=False)['Orders'].sum().
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=Ord_per_product,x ='Product_ID',y = 'Orders',color = 'Purpl
```

Out[115… <Axes: xlabel='Product_ID', ylabel='Orders'>



Conclusion

Married women age-group yrs from UP, Maharastra and karnataka working in IT, Healthcare and Aviation are more likely to by products from food Clothing and Electronics Category

In [ ]: