Open in Colab

In [15]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

In [16]:
```
df=pd.read_csv(r'/content/Diwali Sales Data.csv',encoding='unicode_escape')
```

In [17]:
```
df.shape
```

Out [17]: (11251, 15)

In [18]:
```
df.head(10)
```

Out [18]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Ar
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	NaN
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

In [19]:
```
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 [20]:
```
#drop unrelated/blank columns
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

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

Out [21]:

	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

In [22]:
```
#check for sum of null values
pd.isnull(df).sum()
```

Out [22]:
```
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 [23]:
```
#drop delete null values
df.dropna(inplace=True)
```

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

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

In [25]:
```
#initialize list of lists
data_test=[['mandeep',19],['jayesh',19],['Manu',],['Ankit',20]]
#create the pandas DataFrame using list
df_test=pd.DataFrame(data_test,columns=['Name','Age'])
df_test
```

Out [25]:

	Name	Age
0	mandeep	19.0
1	jayesh	19.0
2	Manu	NaN
3	Ankit	20.0

In [26]:
```
#using inplace for save
df_test.dropna(inplace=True)
```

In [27]:
```
df_test
```

Out [27]:

	Name	Age
0	mandeep	19.0
1	jayesh	19.0
3	Ankit	20.0

In [28]:
```
#change data type
df['Amount']=df['Amount'].astype('int')
```

In [29]:
```
df['Amount'].dtypes
```

Out [29]:
```
dtype('int64')
```

In [30]:
```
df.columns
```

Out [30]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [30]:
```

```

In [31]:
```
df.describe()
```

Out [31]:

	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 [31]:

In [32]:
```
df[['Age','Orders','Amount']].describe()
```

Out [32]:

	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

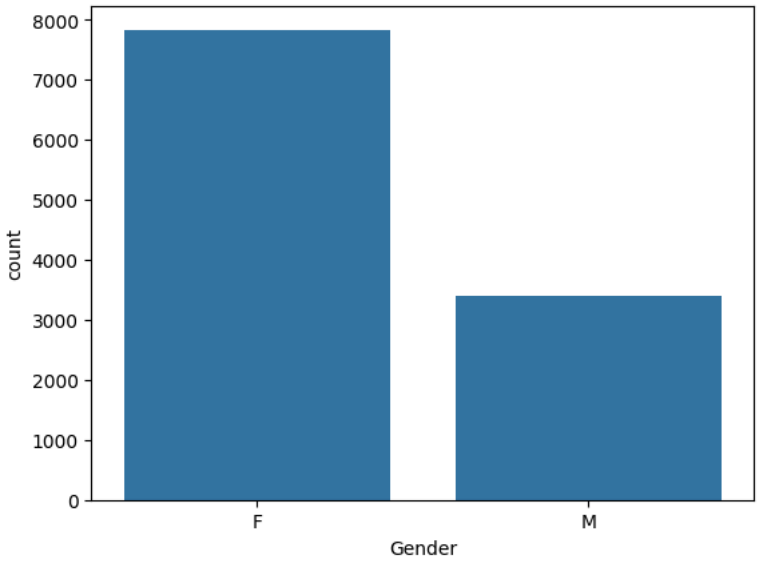
In [33]:
```
#EXPLORATORY DATAANALYTICS
```

In [34]:
```
df.columns
```

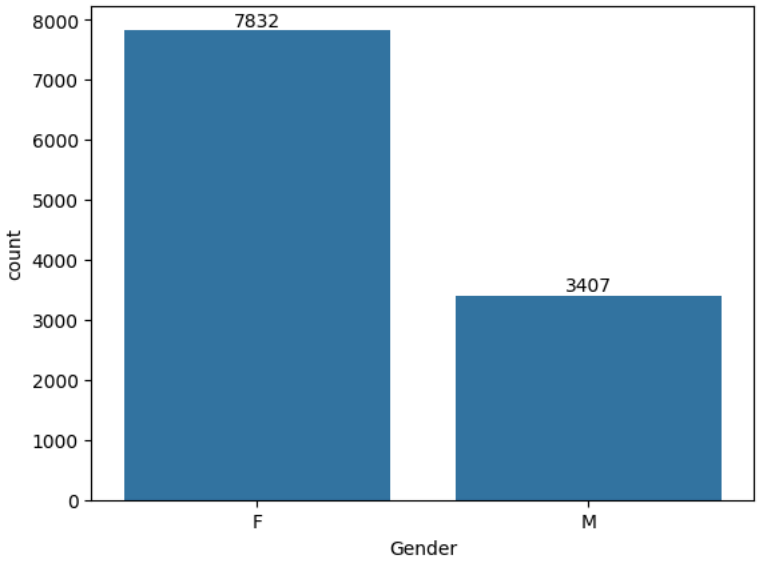
Out [34]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [35]:
```
sns.countplot(x='Gender',data=df)
```

Out [35]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



In [36]:
```
ax=sns.countplot(x='Gender',data=df)
for bars in ax.containers:
    ax.bar_label(bars)
```



In [37]:
```
df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
```

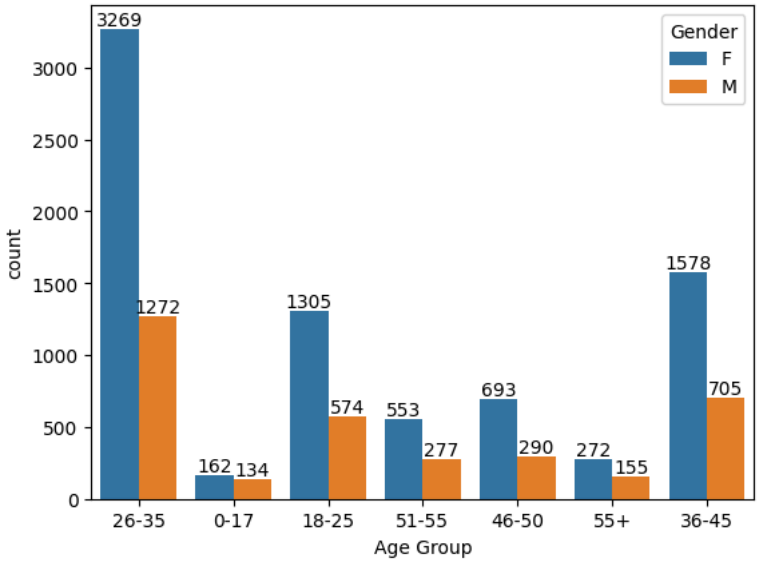
Out [37]:

	Gender	Amount
0	F	74335853
1	M	31913276

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

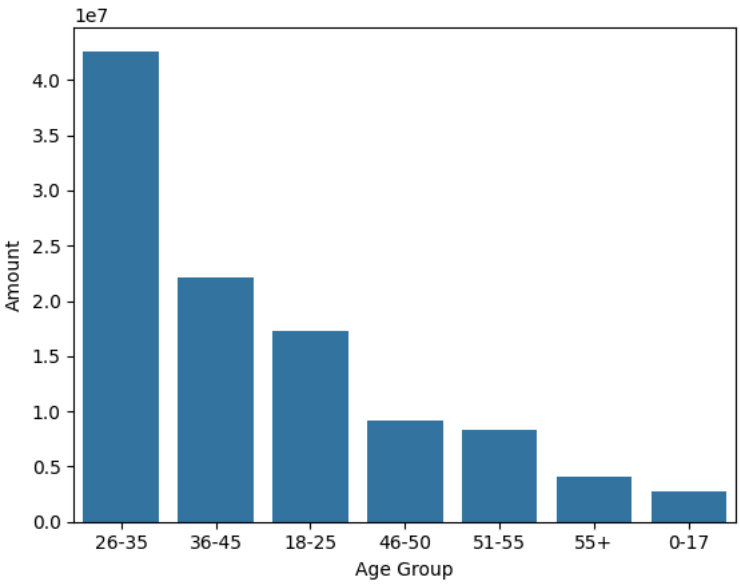
Out [38]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [39]:
```
ax= sns.countplot(data=df,x='Age Group',hue='Gender')
for bars in ax.containers:
        ax.bar_label(bars)
```



In [40]:
```
#total amount and age group
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)
```

Out [40]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

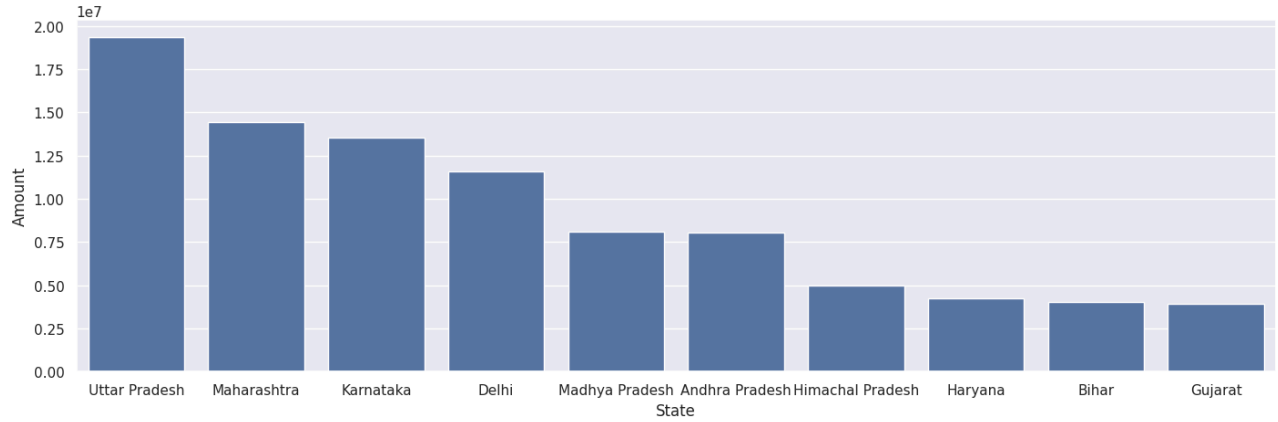


In [41]:
```
df.columns
```

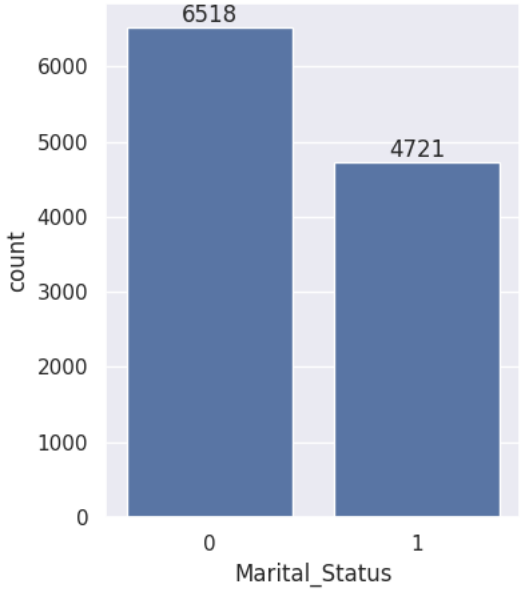
Out [41]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [42]:
```
#total amount /sales from top 10 states
sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head
sns.set(rc={'figure.figsize':(17,5)})
sns.barplot(data=sales_state,x='State',y='Amount')
```

Out [42]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



In [50]:
```
ax= sns.countplot(data=df,x='Marital_Status')
sns.set(rc={'figure.figsize':(4,2)})
for bars in ax.containers:
       ax.bar_label(bars)
```



In [51]:
```
# conclusion: married couple do more shopping and in married female doing more shopping
```