Analysis on diwali Sales

Importing the necessary libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

Importing the data to be analyse

In [2]:
```
df=pd.read_csv("Diwali Sales Data.csv",encoding="unicode_escape")
```

In [3]:
```
df.head()
```

Out[3]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	35	0	Maharashtra	Western	Healthcare	Auto	1	23902.0	NaN	NaN
1	1003132	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23924.0	NaN	NaN
2	1001990	Bindu	P00110942	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237942	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN

In [4]:
```
df.shape
```

Out[4]:
```
(11251, 15)
```

Performing Data Cleaning

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

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

Out[7]:
```
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 [8]:
```
df['Amount'] = df['Amount'].fillna(df['Amount'].mean())
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          11251 non-null  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

In [9]:
```
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          11251 non-null  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

In [10]:
```
df['Amount']=df['Amount'].astype('int')
```

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

Out[11]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [12]:
```
df.rename(columns={'Marital_Status':'Marriage'})
```

Out[12]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marriage	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23902
1	1003132	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23924
2	1001990	Bindu	P00110942	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237942	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11249	1004023	Noonan	P00090942	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00287042	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11251 rows × 13 columns

In [13]:
```
df[["Age","Orders","Amount"]].describe()
```

Out[13]:

	Age	Orders	Amount
count	11251.000000	11251.000000	11251.000000
mean	35.421337	2.489290	9453.609801
std	12.754122	1.115047	5219.580969
min	12.000000	1.000000	188.000000
25%	27.000000	1.000000	5443.500000
50%	33.000000	2.000000	8110.000000
75%	43.000000	3.000000	12671.000000
max	92.000000	4.000000	23902.000000

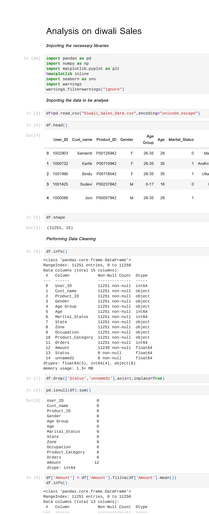
Exploratory Data Analysis: EDA

In [14]:
```
df.columns
```

Out[14]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [15]:
```
sns=sns.countplot(data=df,x='Age Group',hue="Gender")
for bars in sns.containers:
    sns.bar_label(bars)
```

Out[15]:



In [17]:
```
sales_age = df.groupby(["Age Group"], as_index=False).sum().sort_values(by="Amount", ascending=False)
sns.barplot(x="Age Group",y="Amount",data=sales_age)
for bar in ax.patches:
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.0f}', ha='center', va='bottom')
```

Out[17]:

From above graph we can see that the most of the buyer are of age group between 26-35 years

In [20]:
```
sales_state = df.groupby(["State"], as_index=False).sum().sort_values(by="Orders", ascending=False).head(10)
sns.set(rc={"figure.figsize":(20,5)})
sns.barplot(x="State",y="Orders",data=sales_state,hue="State")
```

Out[20]:

So from above graph we see that the Uttarpradesh have maximum number of orders and Gujrat have minimum of orders

In [18]:
```
sales_state_Amt = df.groupby(["State"], as_index=False).sum().sort_values(by="Amount", ascending=False).head(10)
sns.set(rc={"figure.figsize":(14,5)})
sns.barplot(x="State",y="Amount",data=sales_state_Amt,hue="State")
```

Out[18]:

Now we can see that the Haryana have less orders then Bihar but Haryana spends more finacially

In [21]:
```
sales_Marital_Status = df.groupby(["Marital_Status","Gender"], as_index=False).sum().sort_values(by="Amount", ascending=False)
sns.set(rc={"figure.figsize":(4,5)})
# Hacker_numerical values with categorical labels
sales_Marital_Status["Marital_Status"] = sales_Marital_Status['Marital_Status'].replace([0, 1], ['Married', 'Unmarried'], inplace=True)
sns=sns.barplot(x="Marital_Status",y="Amount",hue="Gender",data=sales_Marital_Status)
for bars in sns.containers:
    for bar in bars:
        height = bar.get_height() / 100000  # Dividing by 1,000,000 to convert to millions
        sns.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.1f}M', ha='center', va='bottom')
```

Out[21]:

From above graph we can see that most of the buyers are married womens

In [24]:
```
sns.set(rc={"figure.figsize":(20,5)})
sns=sns.countplot(data=df,x="Occupation")
for bar in sns.patches:
    height = bar.get_height()
    sns.text(bar.get_x() + bar.get_width() / 2, height, f'{height}', ha='center', va='bottom')
```

Out[24]:

In [25]:
```
sales_Occupation = df.groupby(["Occupation"], as_index=False).sum().sort_values(by="Amount", ascending=False).head(10)
sns.set(rc={"figure.figsize":(15,5)})
sns.barplot(x="Occupation",y="Amount",data=sales_Occupation ,hue="Occupation")
```

Out[25]:

From above graph we can see that most of the buyers are working in IT Sector,Healthcare,Aviation.***

In [26]:
```
sales_Product_Category = df.groupby(["Product_Category"], as_index=False).sum().sort_values(by="Amount", ascending=False).head(10)
sns.set(rc={"figure.figsize":(15,5)})
sns.barplot(x="Product_Category",y="Amount",data=sales_Product_Category ,hue="Product_Category")
```

Out[26]:

In [27]:
```
sales_Product_Category = df.groupby(["Product_Category"], as_index=False).sum().sort_values(by="Orders", ascending=False).head(10)
sns.set(rc={"figure.figsize":(15,5)})
sns.barplot(x="Product_Category",y="Orders",data=sales_Product_Category ,hue="Product_Category")
```

Out[27]:

From above two graph we can see that the clothing&apparel have highest number of order but amount spent on food is higher then clothing&apparel

Conclusion

The Diwali sales data analysis reveals key insights into consumer behavior and preferences during the festive season. To maximize sales during Diwali: Target marketing efforts towards the age group between 26-35 years, especially focusing on married women. Tailor promotional campaigns to resonate with buyers from states like Uttar Pradesh and Haryana, considering their high contribution to sales. Focus on product categories like clothing & apparel and food, as they attract significant orders and spending.

In [ ]:
```

```