Diwali Sales Analysis Report

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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
dataset = '/content/drive/MyDrive/Diwali Sales Data.csv'
```

```
dataset = pd.read_csv('/content/drive/MyDrive/Diwali Sales Data.csv', encoding= 'unicode_escape')
# to avoid encoding error, use 'unicode_escape'
```

```
dataset.shape
#row and column find
```

```
(11251, 15)
```

```
dataset.head(5)
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amo
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23

```
dataset.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
```

```
#drop uncrelated/Bank column
dataset.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

```
dataset.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
```

```
pd.isnull(dataset).sum()
```

```
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
```

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

```
pd.isnull(dataset).sum()
```

```
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
```

```
dataset['Amount'].dtypes
```

```
dtype('float64')
```

```
#change datatypes
dataset['Amount'] = dataset['Amount'].astype('int')
```

```
dataset['Amount'].dtypes
```

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

```
#Use describe() for specific columns
dataset[['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 Analysis

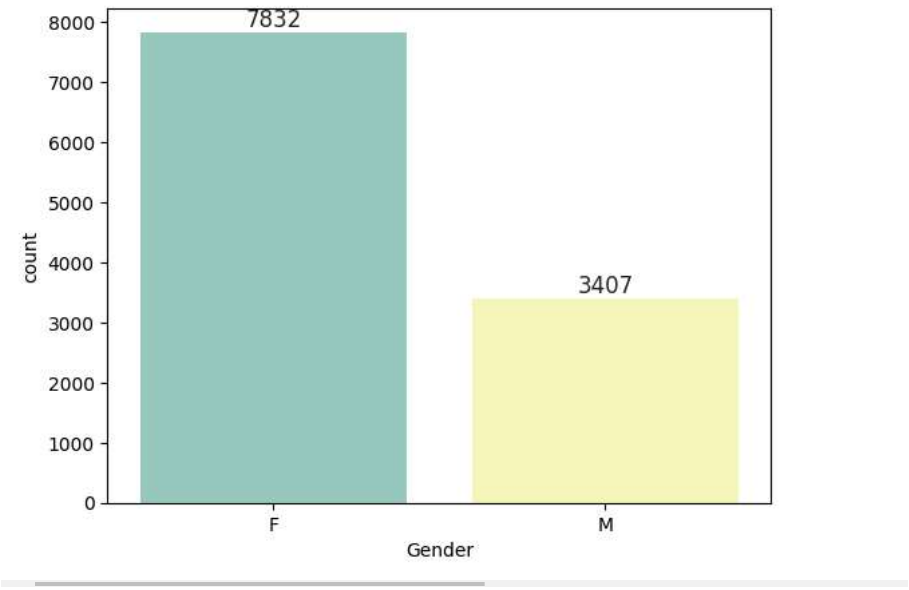
Gender

```
dataset.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 = dataset, palette='Set3')
sns.set(rc={'figure.figsize': (10,5)})
for bars in ax.containers:
  ax.bar_label(bars)
```

```
<ipython-input-20-99151e4abd06>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  ax = sns.countplot(x = 'Gender', data = dataset, palette='Set3')
```

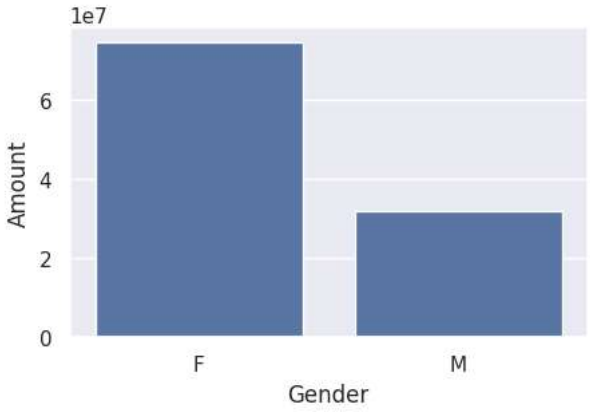



```
dataset.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
```

	Gender	Amount
0	F	74335853
1	M	31913276

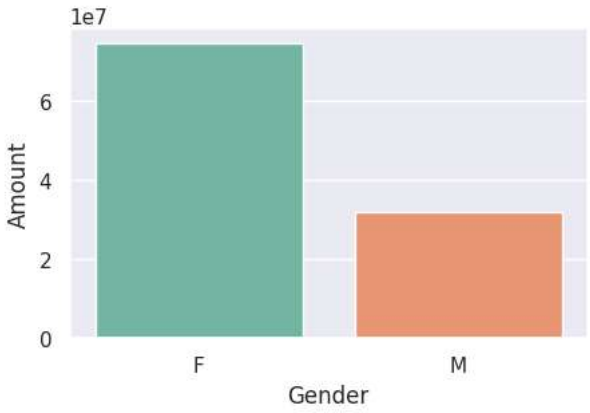
```
Sales_gen = dataset.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize': (5,3)})
sns.barplot(x = 'Gender', y = 'Amount' , data = Sales_gen)
```

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




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

sns.set(rc={'figure.figsize': (5,3)})
sns.barplot(x='Gender', y='Amount', data=Sales_gen, palette='Set2')
```

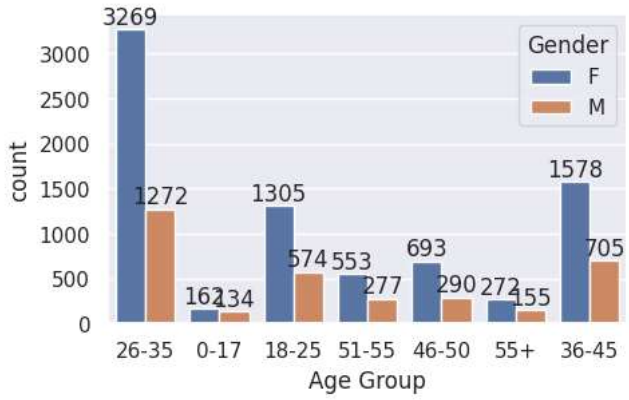
```
<ipython-input-23-dfd0b73693a7>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x='Gender', y='Amount', data=Sales_gen, palette='Set2')
<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

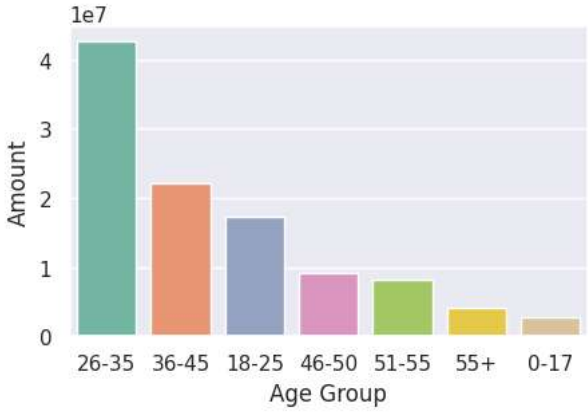
```
ax = sns.countplot(data = dataset, x = 'Age Group', hue = 'Gender')
for bars in ax.containers:
  ax.bar_label(bars)
```



Total Amount vs Age Group

```
sales_age = dataset.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending=False)

sns.barplot(x ='Age Group', y = 'Amount', data = sales_age, palette='Set2')
```

```
<ipython-input-25-60d5e06811f8>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x ='Age Group', y = 'Amount', data = sales_age, palette='Set2')
<Axes: xlabel='Age Group', ylabel='Amount'>
```



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

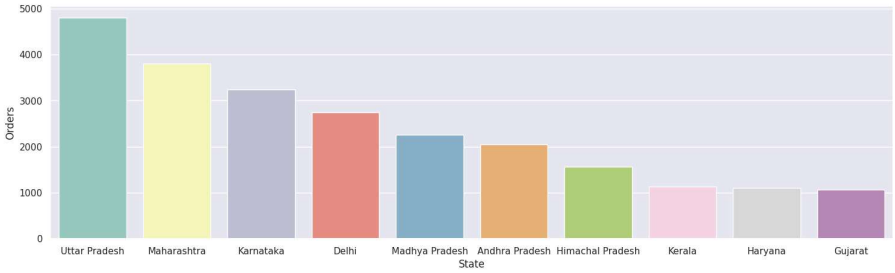
State

```
dataset.columns
```

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

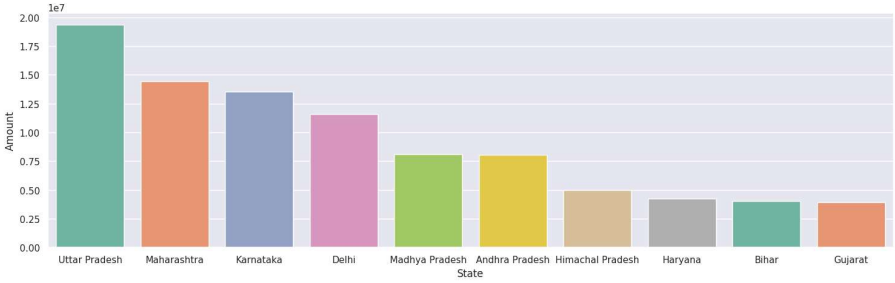
```
# Total number of orders from top 10 states
sales_state = dataset.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

sns.set(rc={'figure.figsize':(18,5)})
sns.barplot(data = sales_state, x = 'State' ,y='Orders', palette='Set3')
```

```
<ipython-input-27-f05c7687dcea>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(data = sales_state, x = 'State' ,y='Orders', palette='Set3')
<Axes: xlabel='State', ylabel='Orders'>
```




```
# Total number of Amount from top 10 states
sales_state = dataset.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize': (18,5)})
sns.barplot(data = sales_state, x = 'State', y='Amount', palette= 'Set2')
```

```
<ipython-input-28-d50cc17dfd0e>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(data = sales_state, x = 'State', y='Amount', palette= 'Set2')
<Axes: xlabel='State', ylabel='Amount'>
```

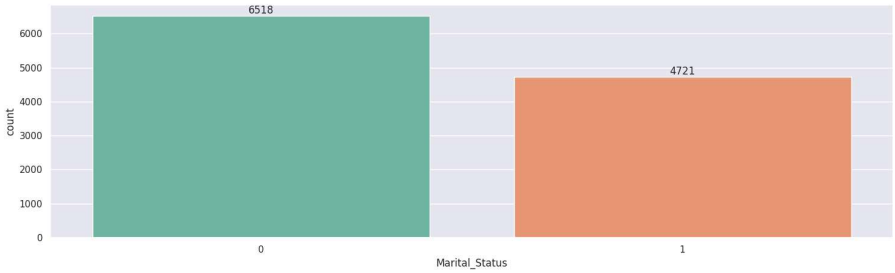


From above graphs we can see that most of the orders & total sales/amount are from Utter Pradesh, Maharashtra, Karnataka respectively

Marital Status

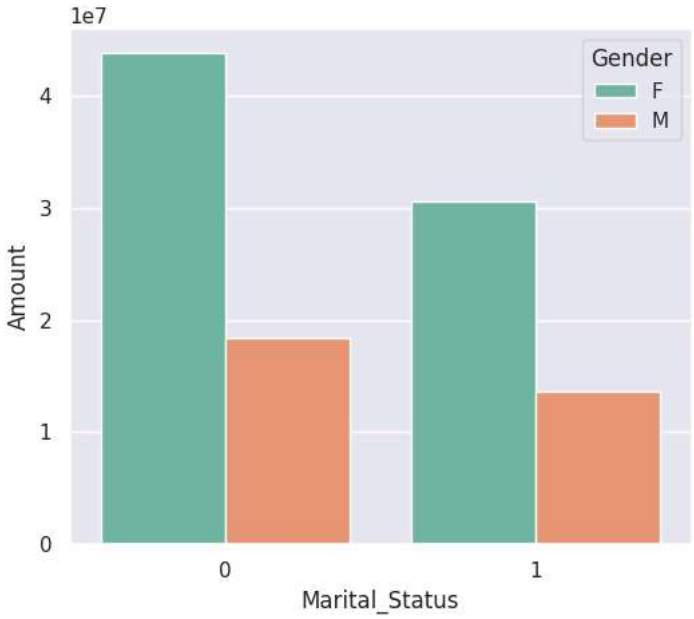
```
ax = sns.countplot(data = dataset, x = 'Marital_Status', palette= 'Set2')
sns.set(rc={'figure.figsize':(8,5)})
for bars in ax.containers:
  ax.bar_label(bars)
```

```
<ipython-input-29-ea3296314001>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  ax = sns.countplot(data = dataset, x = 'Marital_Status', palette= 'Set2')
```




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

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data=sales_state, x='Marital_Status', y='Amount', hue='Gender', palette= 'Set2')
```

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

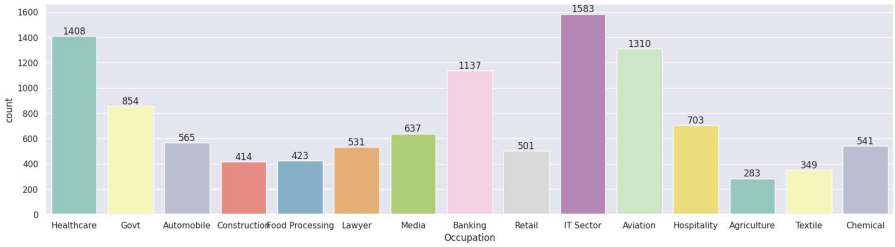


From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

Occupation

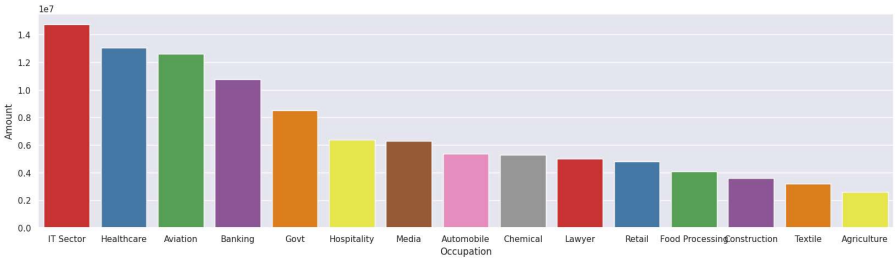
```
sns.set(rc={'figure.figsize': (20,5)})
ax = sns.countplot(data = dataset, x = 'Occupation', palette='Set3')

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

```
<ipython-input-31-e57a38deba3b>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  ax = sns.countplot(data = dataset, x = 'Occupation', palette='Set3')
```




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

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state, x = 'Occupation', y = 'Amount', palette='Set1')
```

```
<ipython-input-32-709cee42ad38>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(data=sales_state, x = 'Occupation', y = 'Amount', palette='Set1')
<Axes: xlabel='Occupation', ylabel='Amount'>
```

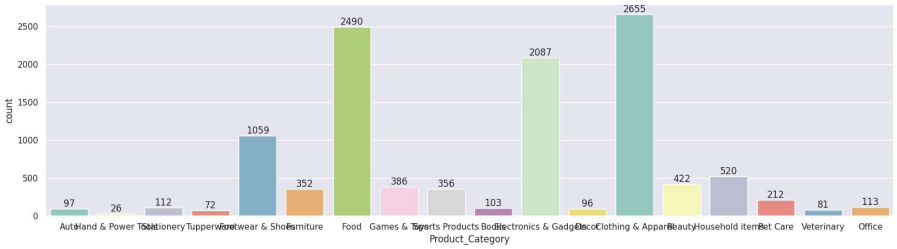


From above graphs we can see that most of the buyers are working in IT, Healyhcare and Avistion sector

Product Category

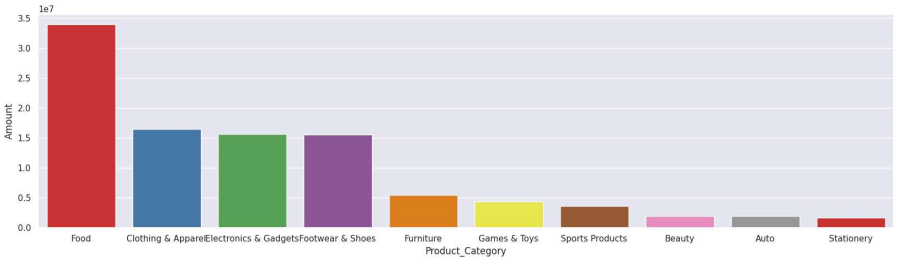
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = dataset, x = 'Product_Category', palette='Set3')

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

```
<ipython-input-33-eaa6449ac7a1>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  ax = sns.countplot(data = dataset, x = 'Product_Category', palette='Set3')
```




```
sales_state = dataset.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state, x = 'Product_Category', y = 'Amount', palette='Set1')
```

```
<ipython-input-34-2ffb300835b4>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(data=sales_state, x = 'Product_Category', y = 'Amount', palette='Set1')
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



From Above graphs we can see that most of the sold products are from food, clothing and Electronics category

```
# Top 10 most sold product (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,5))
dataset.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')
```

```
<Axes: xlabel='Product_ID'>
```