Project Title: Diwali Sales Analysis

Project Summary:

I have worked on this project in which there is a company that has given me its Diwali period data to perform Exploratory Data Analysis functions on its Diwali sales data.

Import Libraries for EDA

In [272…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Read csv file

In [273…
```
df = pd.read_csv("Diwali Sales Data.csv", encoding = "unicode_escape")
```

Data Cleaning

In [274…
```
df.shape
```

Out[274]:
```
(11251, 15)
```

In [275…
```
df.head(10)
```

Out[275]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4

In [276…
```
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 [277…
```
#drop unrelated/blank columns
df.drop(['Status','unnamed1'], axis=1, inplace=True)
```

In [278…
```
#check for null values
pd.isnull(df).sum()
```

Out[278]:
```
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 [279…
```
#drop null values
df.dropna(inplace=True)
```

In [280…
```
#change datatype
df['Amount'] = df['Amount'].astype('int')
```

In [281…
```
df['Amount'].dtypes
```

Out[281]:
```
dtype('int32')
```

In [282…
```
df.columns
```

Out[282]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [283…
```
#rename column
df.rename(columns = {'Gender':'Sex'}, inplace=True)
```

In [284…
```
#describe() method returns statistical description of data in the dataframe(i.e. count, min, max, std, mean, et
df.describe()
```

Out[284]:

	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 [285…
```
#use describe() for specif columns
df[['Orders','Amount']].describe()
```

Out[285]:

	Orders	Amount
count	11239.000000	11239.000000
mean	2.489634	9453.610553
std	1.114967	5222.355168
min	1.000000	188.000000
25%	2.000000	5443.000000
50%	2.000000	8109.000000
75%	3.000000	12675.000000
max	4.000000	23952.000000

In [286…
```
#The data has been cleaned and all the changes have been made. You can see here.
df
```

Out[286]:

	User_ID	Cust_name	Product_ID	Sex	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2
...	...	...	...	...	...	...	...	...	...	...	...	..
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3

11239 rows × 13 columns

Exploratory Data Analysis - EDA

Sex

In [287…

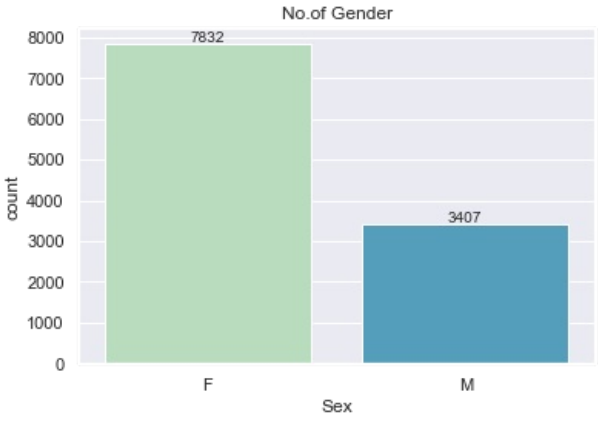
```
#No. of gender
ax = sns.countplot(x = 'Sex', data = df, palette = 'GnBu')

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

plt.title('No.of Gender')
```

Out[287]:

```
Text(0.5, 1.0, 'No.of Gender')
```



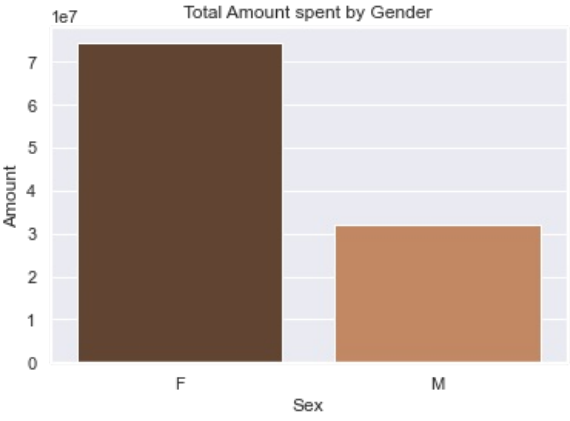
In [288…

```
#Total Amount vs Sex
sales_sex = df.groupby(['Sex'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending = False)

sns.barplot(x = 'Sex', y= 'Amount', data = sales_sex, palette = 'copper')
plt.title('Total Amount spent by Gender')
```

Out[288]:

```
Text(0.5, 1.0, 'Total Amount spent by Gender')
```



from above graph we can see that the most of the buyers are Females and even the purchasing power of female's are greater than men.

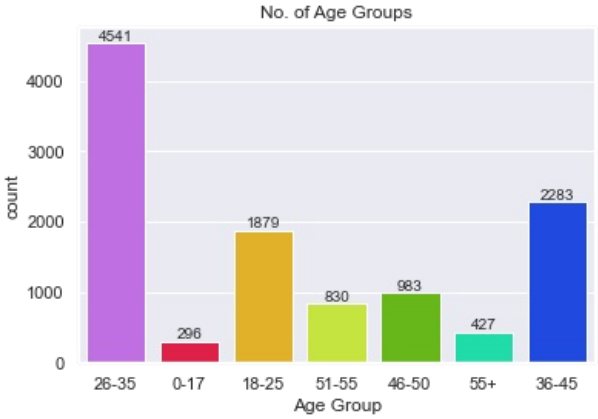
Age

In [289…
```
#No. of Age Groups
ax = sns.countplot(x = 'Age Group', data = df, palette = 'gist_ncar_r')

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

plt.title('No. of Age Groups')
```

Out[289]:
```
Text(0.5, 1.0, 'No. of Age Groups')
```

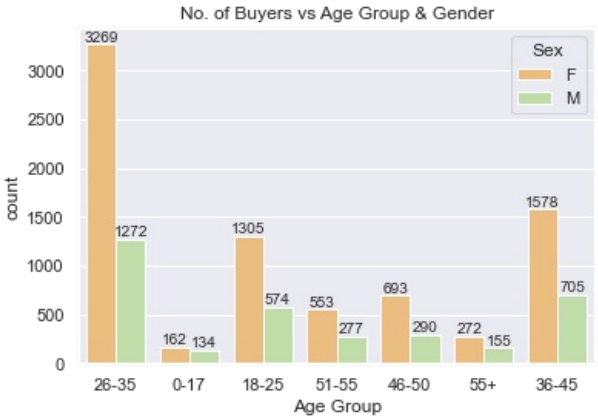


In [290…
```
#No. of purchases gender wise according to age group
ax = sns.countplot(x = 'Age Group', hue = 'Sex', data = df, palette = 'Spectral')

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

plt.title('No. of Buyers vs Age Group & Gender')
```

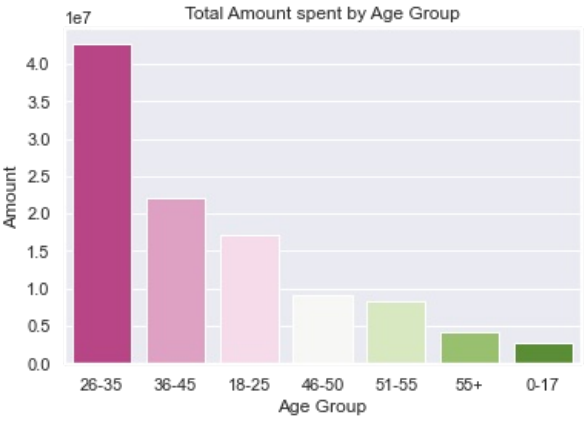
Out[290]:
```
Text(0.5, 1.0, 'No. of Buyers vs Age Group & Gender')
```



In [291…
```
#Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending =

sns.barplot(x = 'Age Group', y= 'Amount', data = sales_age, palette = 'PiYG')
plt.title('Total Amount spent by Age Group')
```

Out[291]:
```
Text(0.5, 1.0, 'Total Amount spent by Age Group')
```



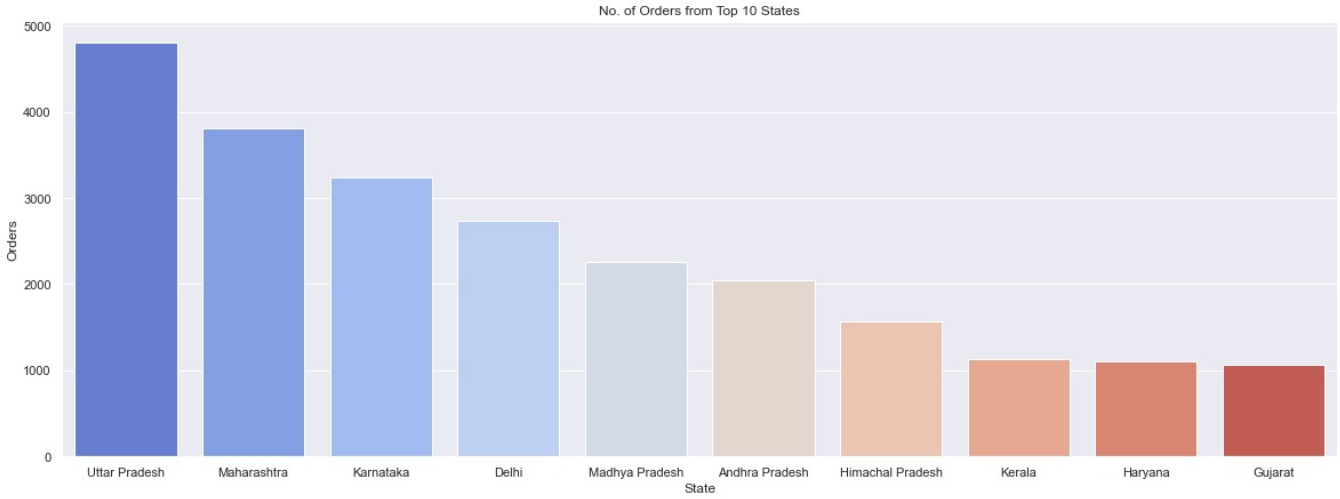
from above graphs we can see that most of the buyers are of age group between 26-35 and most of them are Females

State

In [292…
```
#Total no. of orders from top 10 states
sales_state = df.groupby(['State'], as_index = False)['Orders'].sum().sort_values(by = 'Orders', ascending = Fa

sns.set(rc={'figure.figsize':(20,7)})
sns.barplot(x = 'State', y= 'Orders', data = sales_state, palette = 'coolwarm')
plt.title('No. of Orders from Top 10 States')
```

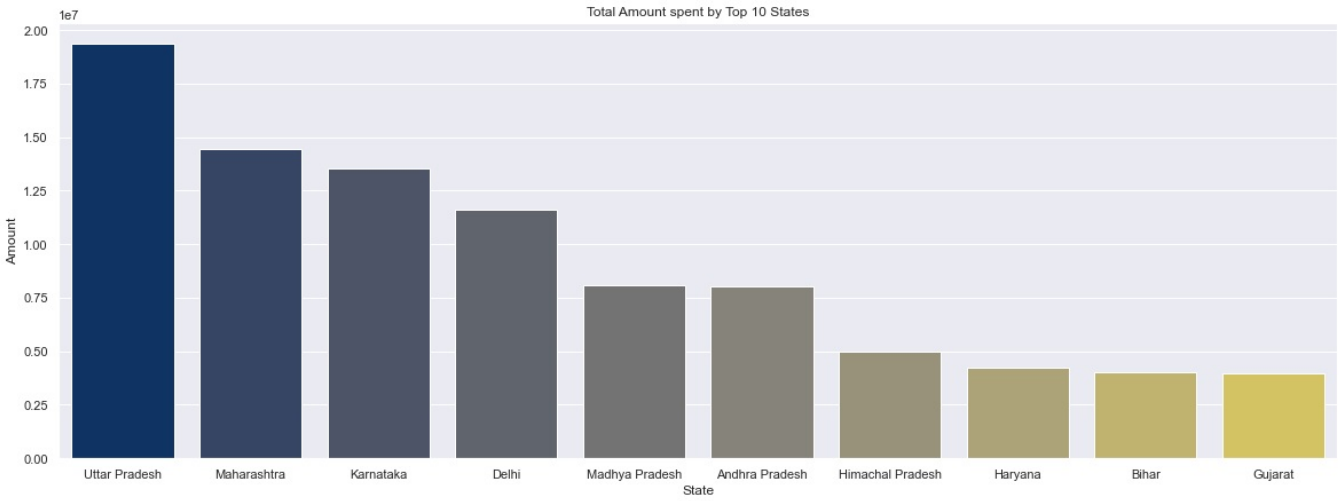
Out[292]:
```
Text(0.5, 1.0, 'No. of Orders from Top 10 States')
```



In [293…
```
# Total amount of sales from the top 10 states.
sales_state = df.groupby(['State'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending = Fa

sns.set(rc={'figure.figsize':(20,7)})
sns.barplot(x = 'State', y= 'Amount', data = sales_state, palette = 'cividis')
plt.title('Total Amount spent by Top 10 States')
```

Out[293]:
```
Text(0.5, 1.0, 'Total Amount spent by Top 10 States')
```



From the above graphs, we can see that the no. of orders & the total amount of sales are primarily from Uttar Pradesh, Maharashtra, and Karnataka, respectively.

And if you look carefully at both the graphs, there is a small difference, i.e. Kerala is at number 8 in the total number of orders but in the graph of the total amount Kerala is not there. This shows that Kerala is only in no. of sales and that too in low-average price items and not inexpensive items. The store earn better from Haryana and Bihar instead of Kerala.

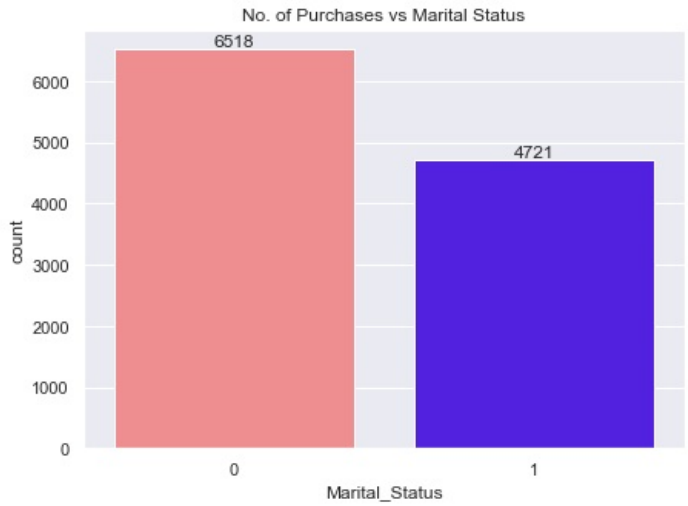
Marital Status

In [296…
```
#No. of Purchases according to Marital Status
ax = sns.countplot(x = 'Marital_Status', data = df, palette = 'gnuplot2_r')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)

plt.title('No. of Purchases vs Marital Status')
```

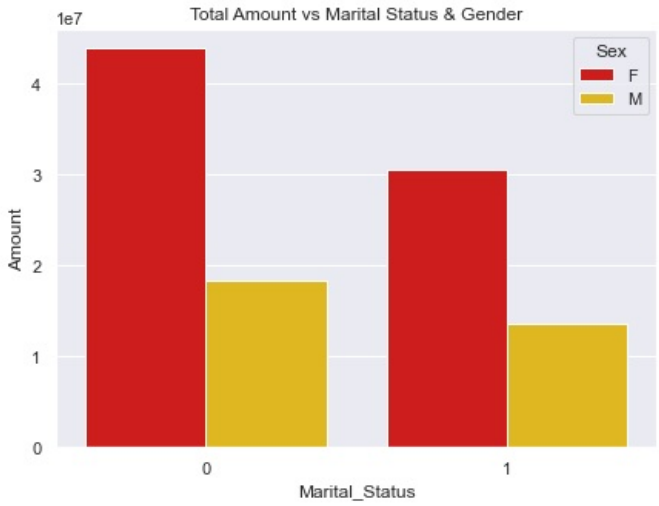
Out[296]:
```
Text(0.5, 1.0, 'No. of Purchases vs Marital Status')
```



In [297…
```
#Total amount of sales according to marital status and sex
sales_ms = df.groupby(['Marital_Status', 'Sex'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', a

sns.barplot(x = 'Marital_Status', y= 'Amount', data = sales_ms, hue = 'Sex', palette = 'hot')
plt.title('Total Amount vs Marital Status & Gender')
```

Out[297]:
```
Text(0.5, 1.0, 'Total Amount vs Marital Status & Gender')
```



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

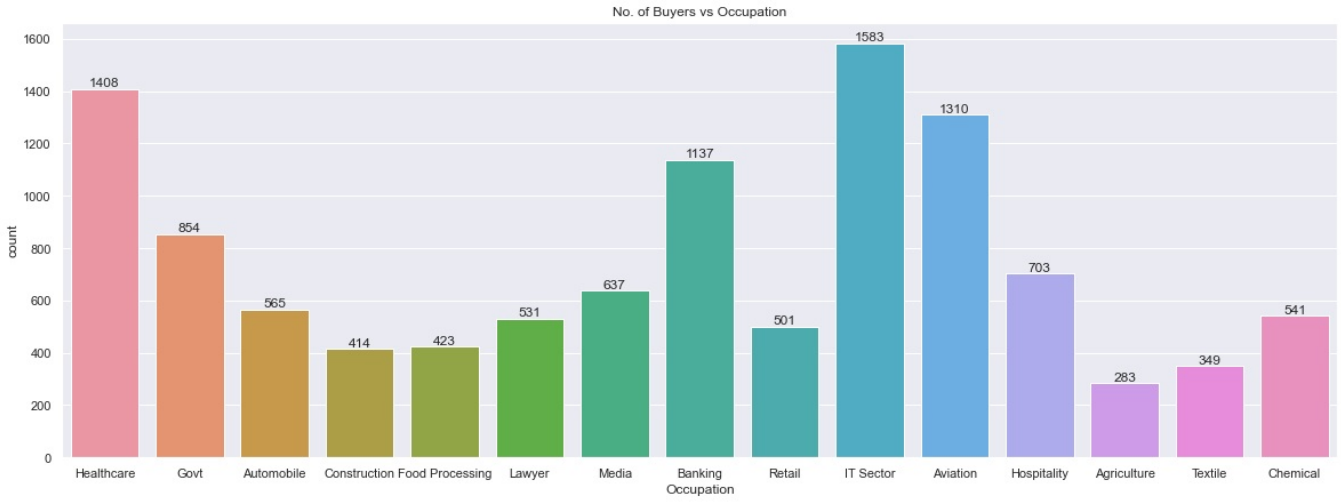
Occupation

In [299…
```
#No. of Purchases according to Occupation
ax = sns.countplot(x = 'Occupation', data = df)

sns.set(rc={'figure.figsize':(20,7)})
for bars in ax.containers:
    ax.bar_label(bars)

plt.title('No. of Buyers vs Occupation')
```

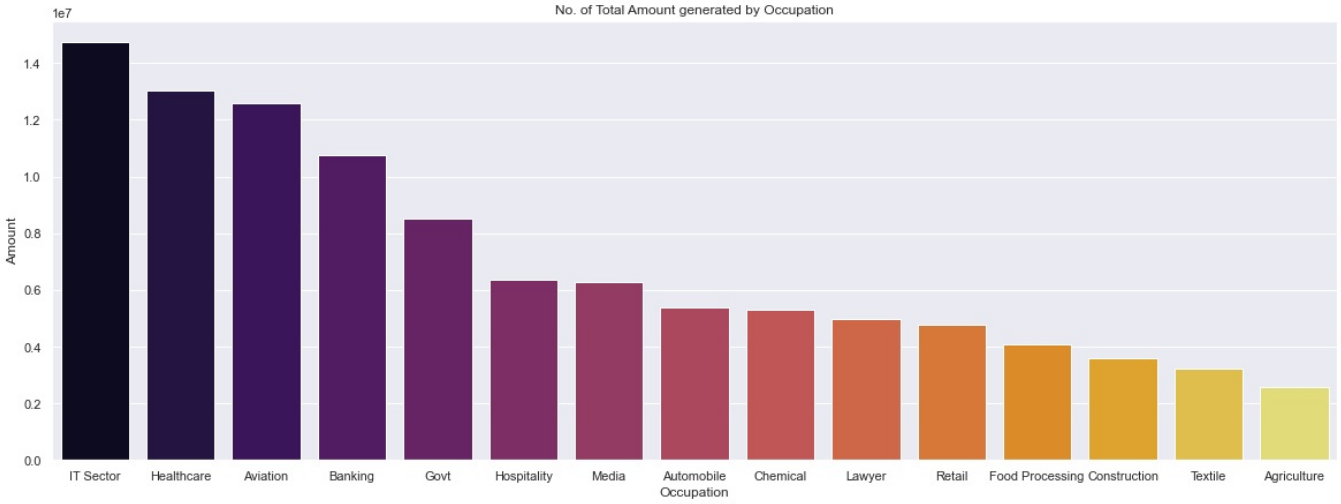
Out[299]:
```
Text(0.5, 1.0, 'No. of Buyers vs Occupation')
```



In [300…
```
#Total amount of sales according to Occupation
sales_occu = df.groupby(['Occupation'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascending

sns.barplot(x = 'Occupation', y= 'Amount', data = sales_occu, palette = 'inferno')
plt.title('No. of Total Amount generated by Occupation')
```

Out[300]:
```
Text(0.5, 1.0, 'No. of Total Amount generated by Occupation')
```



From above graphs we can see that most of the buyers are working in IT, Healtcare, and Aviation Sector then so on.

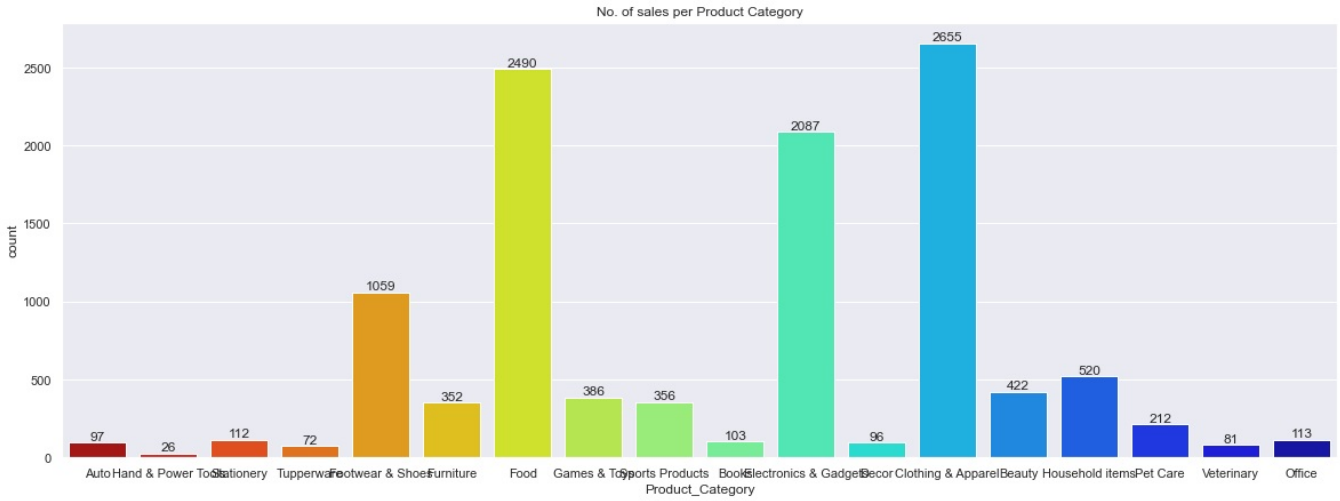
Product Category

In [301…
```
#No. of Sales according to Product Category
ax = sns.countplot(x = 'Product_Category', data = df, palette = 'jet_r')

sns.set(rc={'figure.figsize':(32,10)})
for bars in ax.containers:
    ax.bar_label(bars)

plt.title('No. of sales per Product Category')
```

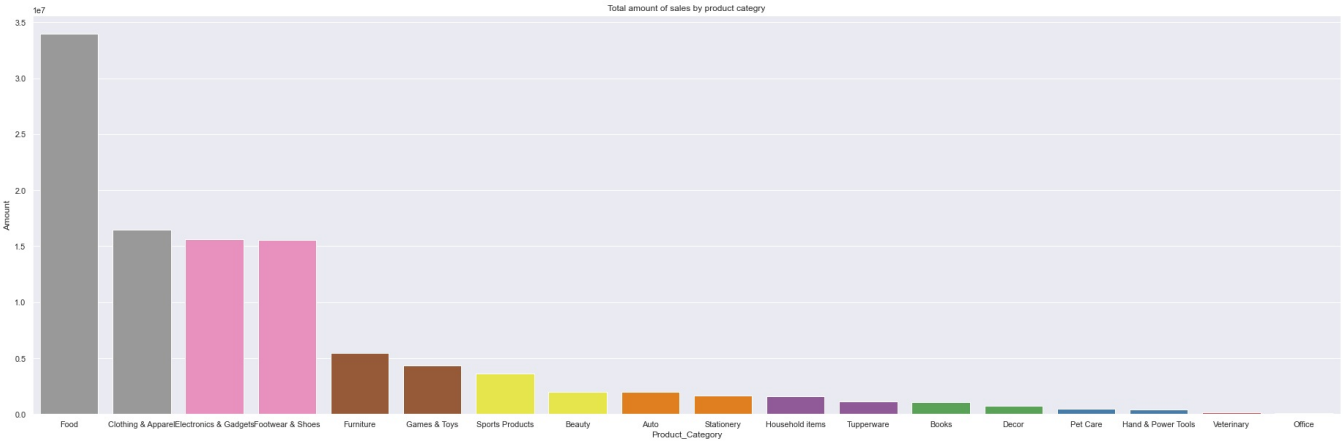
Out[301]:
```
Text(0.5, 1.0, 'No. of sales per Product Category')
```



In [302…
```
#Total amount of sales according to Product category
sales_pc = df.groupby(['Product_Category'], as_index = False)['Amount'].sum().sort_values(by = 'Amount', ascend

sns.barplot(x = 'Product_Category', y= 'Amount', data = sales_pc, palette = 'Set1_r')
plt.title('Total amount of sales by product categry')
```

Out[302]:
```
Text(0.5, 1.0, 'Total amount of sales by product categry')
```



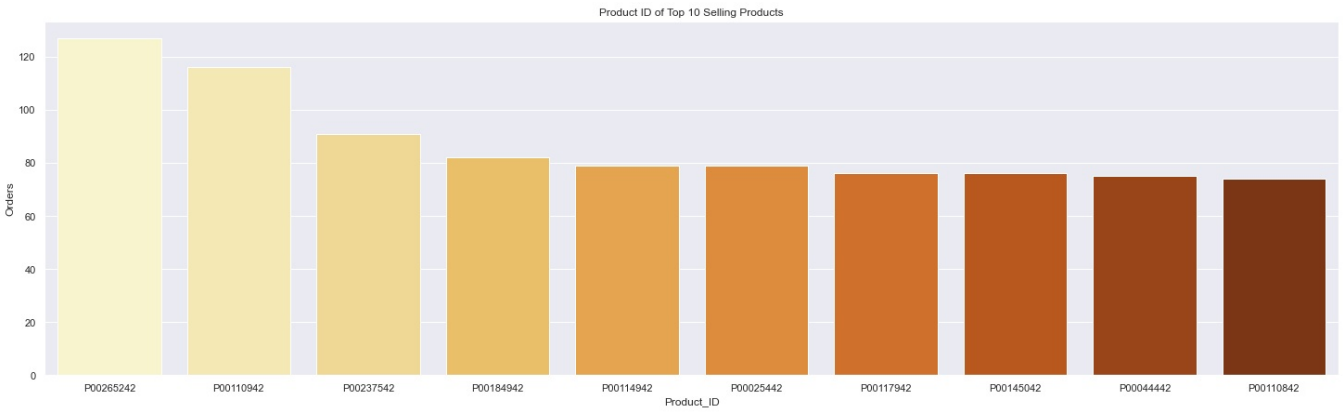
From the above graphs, we can see that the no. of sales by per product category are primarily from Clothing & Apparel, Food, and Electronics & Gadgets respectively.

And from the total amount of sales by product category are primarily from Food, Clothing & Apparel, and Elctronics & Gadgets respectively.

In [303…
```
#Fetch product ID of Top 10 selling Products
#1st Method
sales_ID = df.groupby(['Product_ID'], as_index = False)['Orders'].sum().sort_values(by = 'Orders', ascending =

sns.set(rc={'figure.figsize':(25,7)})
sns.barplot(x = 'Product_ID', y= 'Orders', data = sales_ID, palette = 'YlOrBr')
plt.title('Product ID of Top 10 Selling Products')
```

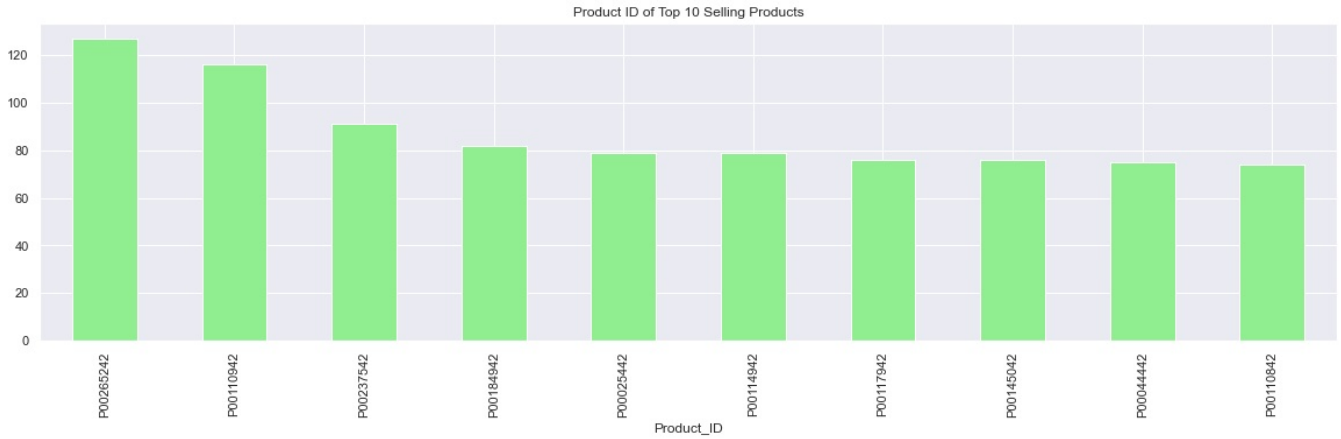
Out[303]:
```
Text(0.5, 1.0, 'Product ID of Top 10 Selling Products')
```



In [304…
```
#2nd Method
```

```
sns.set(rc={'figure.figsize':(20,5)})
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending = False).plot(kind='bar', color='li
plt.title('Product ID of Top 10 Selling Products')
```

Out[304]: Text(0.5, 1.0, 'Product ID of Top 10 Selling Products')



Conclusion:

The Married women of the age group 26-35 years from UP, Maharashtra, and Karnataka working in IT, Healthcare, and Aviation sectors are more likely to buy products from Food, Clothing, and Electronics categories.