Diwali Sales Data Visualization

A Python Data Visualization Project

Narayana Lavanya

Topics

- Introduction
- Tools & Libraries used
- Queries & Data Visualizations
- Key insights
- Conclusion

Introduction

This project analyzes Diwali sales data using Python to uncover key insights about customer behavior, sales trends, and product performance. The goal is to help businesses understand which products performed well, who their top customers are, and seasonal trends during Diwali sales.

Tools & Libraries Used

- **Python Libraries:**
  - * Pandas - Data cleaning & Manipulation
  - * Matplotlib & Seaborn - Data visualization
  - * Jupyter Notebook - Execution environment

Queries & Data Visualizations

1. Importing Libraries & Data

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from matplotlib.ticker import PercentFormatter
```

In [4]:
```
df = pd.read_csv("Diwali Sales Data.csv", encoding = 'unicode_escape')
df.head(10)
```

Out[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	u
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN	
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN	
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN	
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN	
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN	
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	23877.00	NaN	
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN	
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	NaN	NaN	
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN	
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23799.99	NaN	

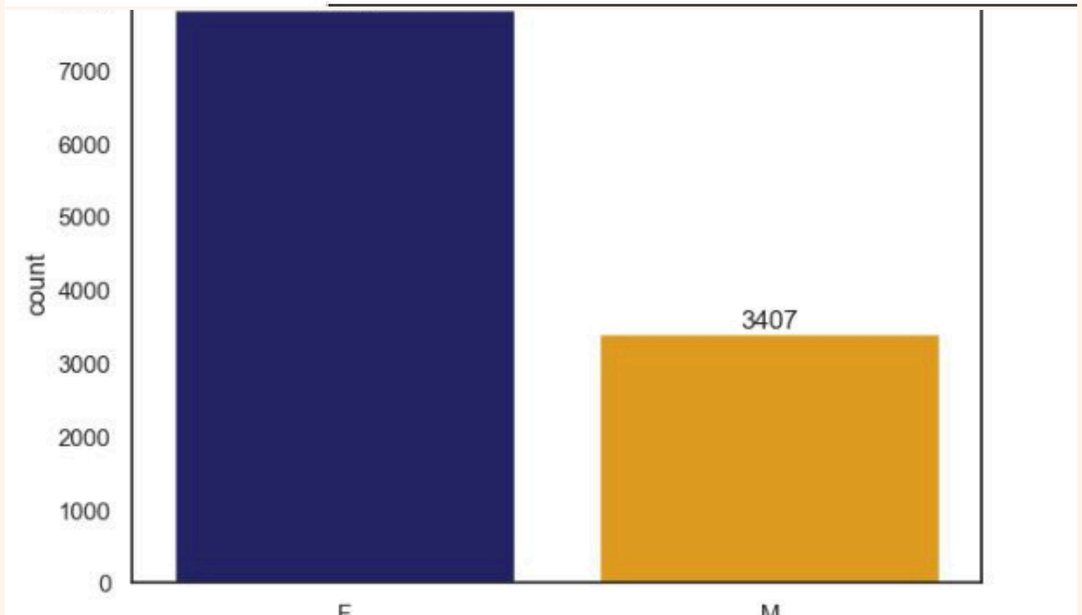
Queries & Data Visualizations

2. Customer counnt by Gender

GENDER

```
In [19]: sns.set_palette('colorblind')

In [28]: ax = sns.countplot(x = 'Gender', data = df_sales_data,
                            palette = ['midnightblue','orange'] )
         sns.set(rc = {'figure.figsize': (8,5)})
         sns.set_style('white')
         for bars in ax.containers:
             ax.bar_label(bars)
```

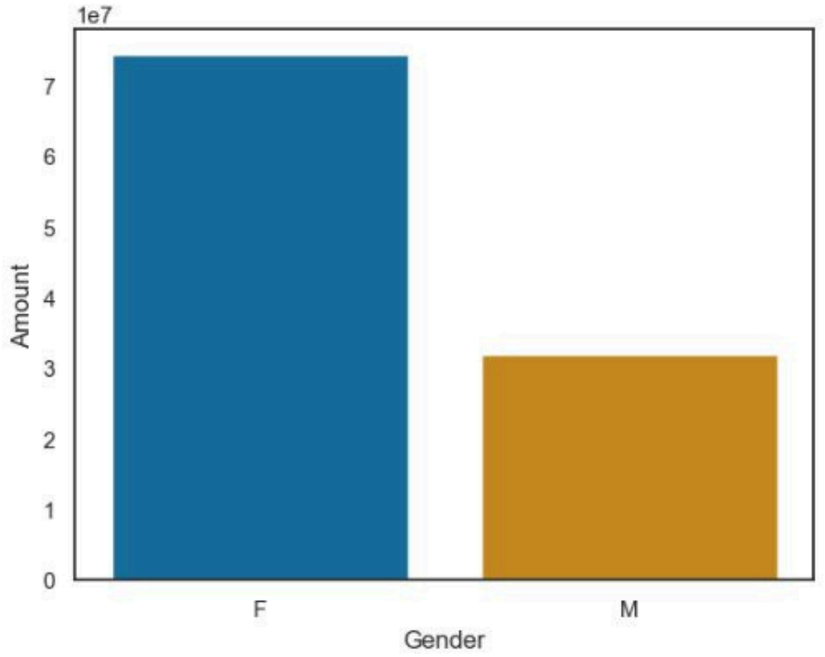


3. Total sales by Gender

```
In [22]: sales_gen = df_sales_data.groupby(['Gender'], as_index = False)
                    ['Amount'].sum().sort_values(by = 'Amount', ascending = False)

         sns.barplot(x = 'Gender', y = 'Amount', data = sales_gen)
```

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



from the above graph most of the buyers are females and the purchasing power of female is greater than me

Queries & Data Visualizations

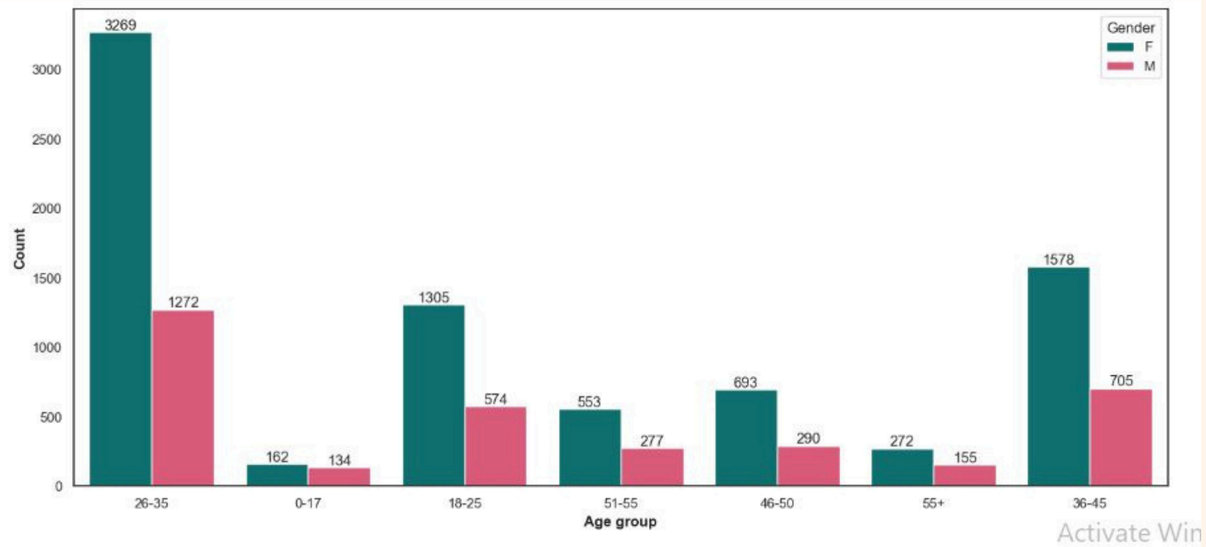
4.Age group distribution by Gender

AGE

```
In [38]: ax = sns.countplot(x = 'Age Group', hue = 'Gender',
                           data = df_sales_data, palette = ['teal','#EF476F'])

         sns.set(rc = {'figure.figsize': (7,5)})
         sns.set_style('white')
         plt.xlabel("Age group", weight = 'bold')
         plt.ylabel("Count", weight = 'bold')

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

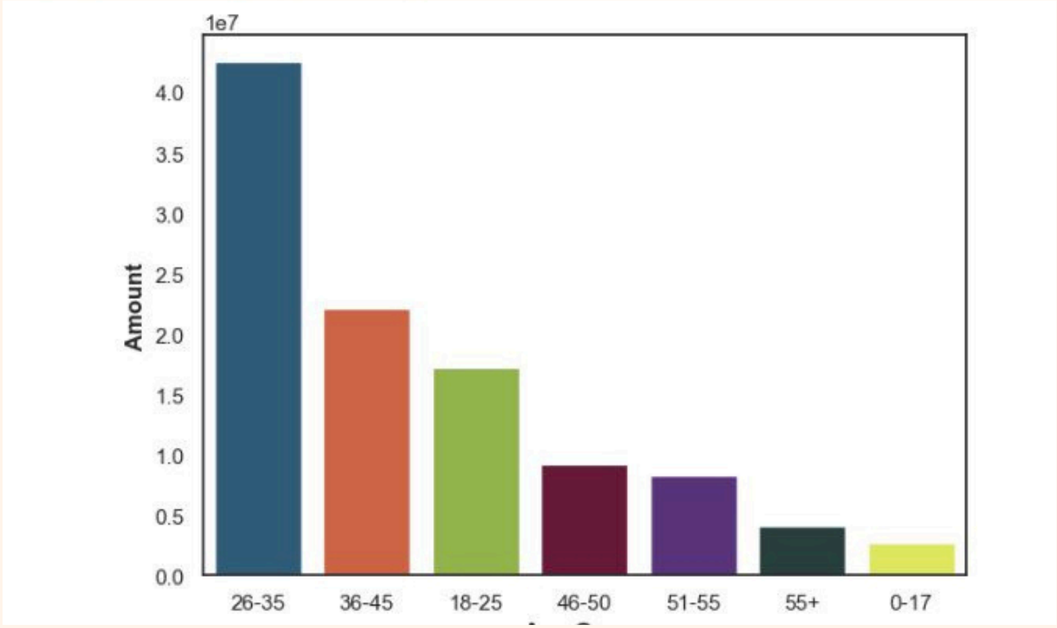


5. Total sales by Age group

```
In [30]: sales_age = df_sales_data.groupby(['Age Group'], as_index = False)
                    ['Amount'].sum().sort_values(by = 'Amount', ascending = False)

         sns.set(rc = {'figure.figsize': (7,5)})
         sns.set_style('white')
         plt.xlabel("Age group", weight = 'bold')
         plt.ylabel("Amount", weight = 'bold')


         sns.barplot(x = 'Age Group', y = 'Amount', data = sales_age,
                     palette = ['#255F85','#E4572E','#9BC53D','#750D37'
                                ,'#5B2A86','#254441','#F2FF49'])
```



Queries & Data Visualizations

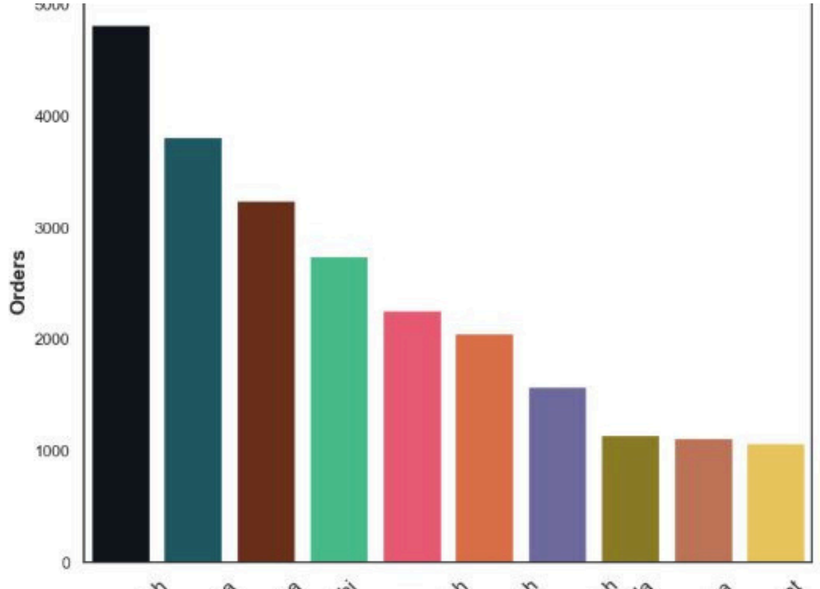
6. Orders by Top 10 States

STATE

In [31]:
```
# total number of orders from top 10 states
sales_state = df_sales_data.groupby(['State'], as_index = False)
    ['Orders'].sum().sort_values(by = 'Orders', ascending = False).head(10)


sns.set(rc = {'figure.figsize': (9,7)})
sns.set_style('white')
plt.xlabel("State", weight = 'bold', fontsize = 14)
plt.ylabel("Orders", weight = 'bold', fontsize = 14)
plt.xticks(rotation = 45, fontsize = 14)


sns.barplot(x = 'State', y = 'Orders', data = sales_state,
           palette = ['#0E131F','#15616D','#78290F','#35CE8D',
              '#FF4365','#F26430','#6761A8','#9B8816','#CE6C47','#FFD046'])
```



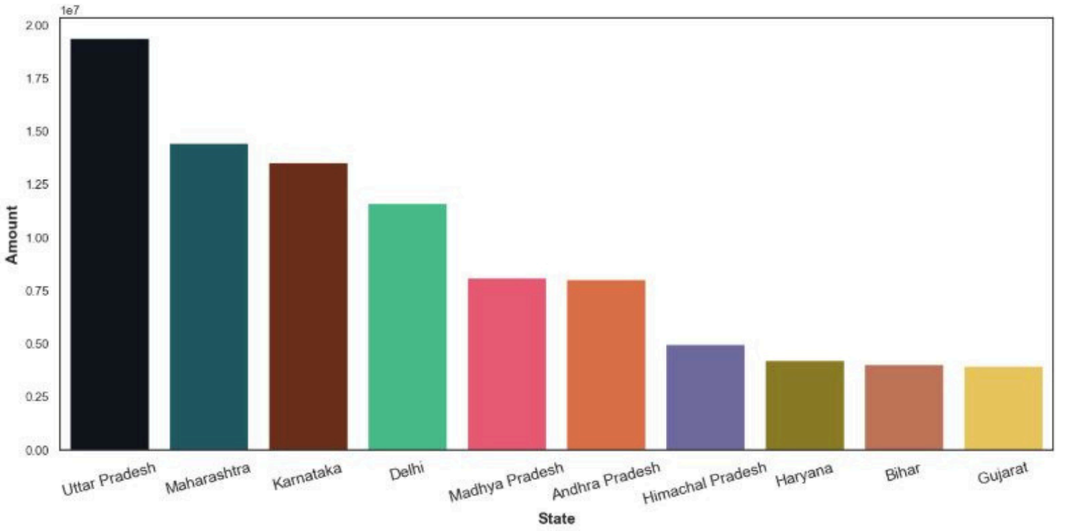
7. Total sales of Top 10 States

In [37]:
```
# total of amount from top 10 states
sales_state = df_sales_data.groupby(['State'], as_index = False)
    ['Amount'].sum().sort_values(by = 'Amount',ascending = False).head(10)

sns.set(rc = {'figure.figsize': (16,7)})
sns.set_style('white')
plt.xlabel("State", weight = 'bold', fontsize = 14)
plt.ylabel("Amount", weight = 'bold', fontsize = 14)
plt.xticks(fontsize = 14, rotation = 15)

sns.barplot(x = 'State', y = 'Amount', data = sales_state,
           palette = ['#0E131F','#15616D','#78290F','#35CE8D',
          '#FF4365','#F26430','#6761A8','#9B8816','#CE6C47','#FFD046'])
```

Out[37]: <Axes: xlabel='State', ylabel='Amount'>



from the above graphs we can see that most of the orders are from UP, maharastra and karnataka and total amount also highest from UP, maharastra and karnataka

Queries & Data Visualizations

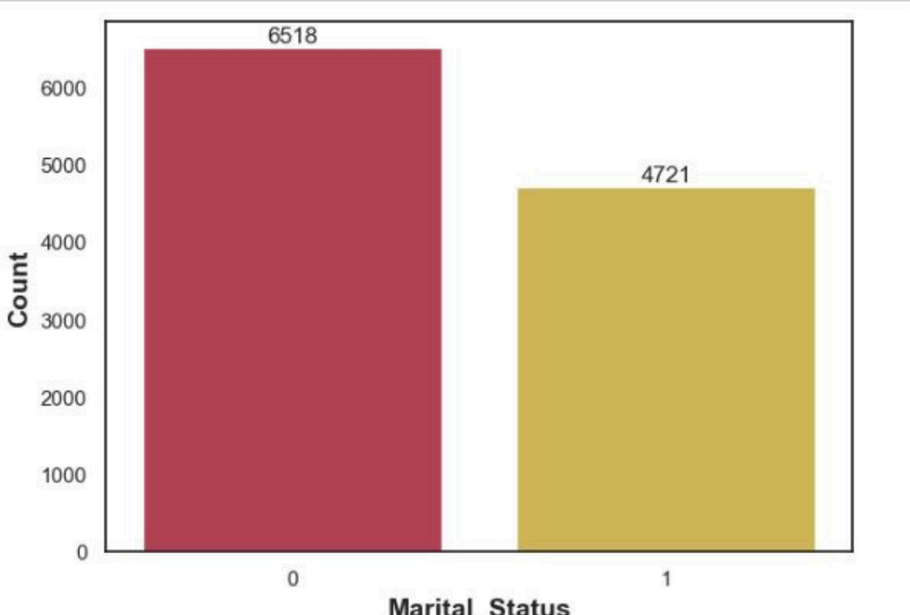
8. Customer count by Marital status

MARITAL STATUS

```
In [53]: ax = sns.countplot(x = 'Marital_Status', data = df_sales_data,
                           palette = ['#C33149','#E2C044'])

         sns.set(rc = {'figure.figsize': (7,4)})
         sns.set_style('white')
         plt.xlabel("Marital_Status", weight = 'bold', fontsize = 14)
         plt.ylabel("Count", weight = 'bold', fontsize = 14)

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



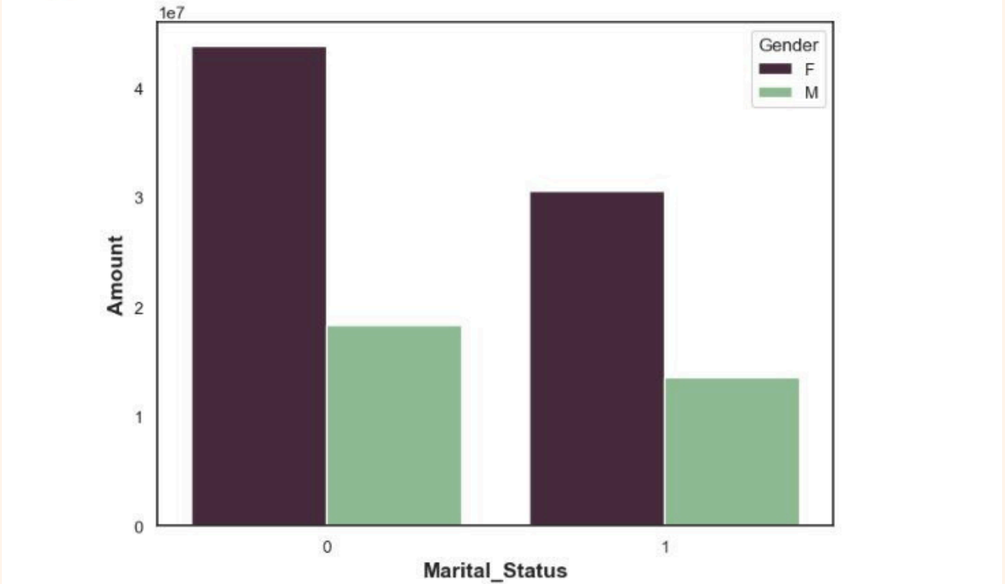
9. Total sales based on marital status.

```
In [54]: sales_state = df_sales_data.groupby(['Marital_Status', 'Gender'],
                as_index = False)['Amount'].sum().sort_values(by = 'Amount',
                                            ascending = False).head(10)

         sns.set(rc = {'figure.figsize': (8,6)})
         sns.set_style('white')
         plt.xlabel("Marital_Status", weight = 'bold', fontsize = 14)
         plt.ylabel("Amount", weight = 'bold', fontsize = 14)

         sns.barplot(x = 'Marital_Status', y = 'Amount',
                     data = sales_state, hue = 'Gender',
                    palette = ['#4B2840','#87C38F'])
```

Out[54]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



Queries & Data Visualizations

10. Customer count by Occupation

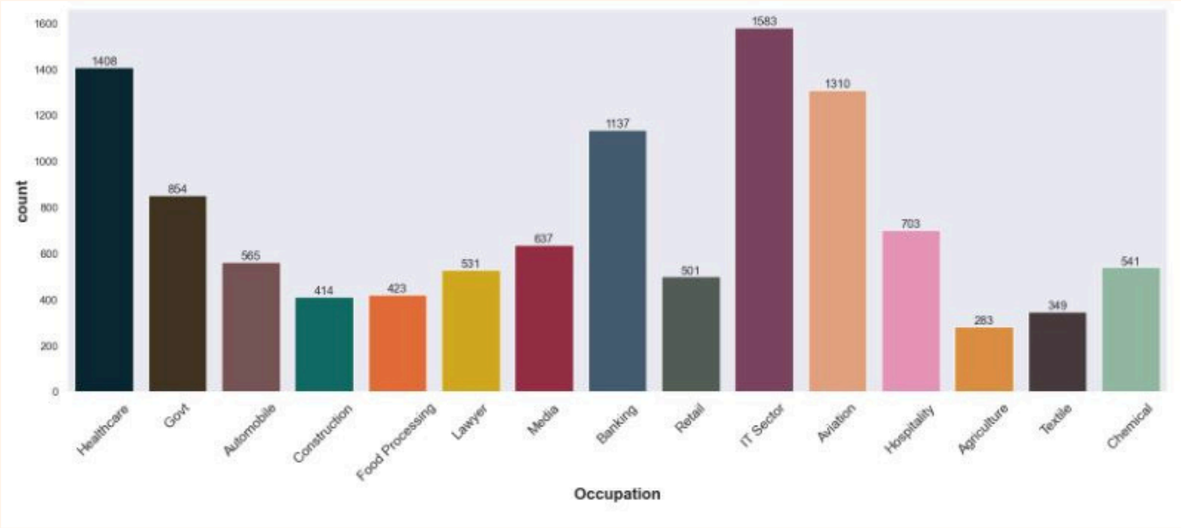
OCCUPATION

```
In [79]: sns.set(rc = {'figure.figsize': (20,7)})

         ax = sns.countplot(x = 'Occupation', data = df_sales_data,
                  palette = ['#022B3A','#46351D','#7D4F50','#037971','#FE621D',
                             '#EEB902','#A61C3C','#3E5C76','#545E56','#843B62',
                             '#F49D6E','#F487B6','#F58F29','#49393B','#8BBF9F'])
         sns.axes_style('white', {'axes.grid': False})
         ax.grid(False)

         plt.xlabel("Occupation", weight = 'bold', fontsize = 16)
         plt.ylabel("count", weight = 'bold', fontsize = 16)
         plt.xticks(rotation = 45, fontsize = 14)


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



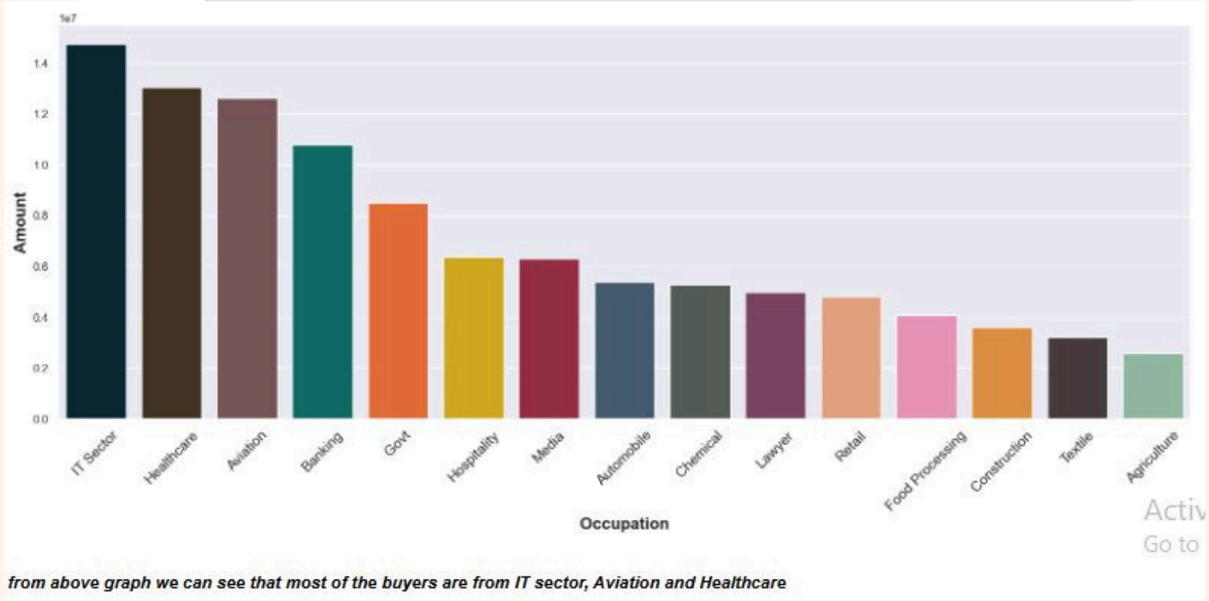
11. Total sales by Occupation

```
In [81]: sales_occupation = df_sales_data.groupby(['Occupation'], as_index = False)
                    ['Amount'].sum().sort_values(by = 'Amount',ascending = False)

         sns.set(rc = {'figure.figsize': (20,7)})
         sns.barplot(x = 'Occupation', y = 'Amount', data = sales_occupation,
                    palette = ['#022B3A','#46351D','#7D4F50','#037971','#FE621D',
                               '#EEB902','#A61C3C','#3E5C76','#545E56','#843B62',
                               '#F49D6E','#F487B6','#F58F29','#49393B','#8BBF9F'])
         sns.set_style('white')

         ax.grid(False)

         plt.xlabel("Occupation", weight = 'bold', fontsize = 16)
         plt.ylabel("Amount", weight = 'bold', fontsize = 16)
         plt.xticks(rotation = 45, fontsize = 14)
```



from above graph we can see that most of the buyers are from IT sector, Aviation and Healthcare

Queries & Data Visualizations

12. Customer count by Product category

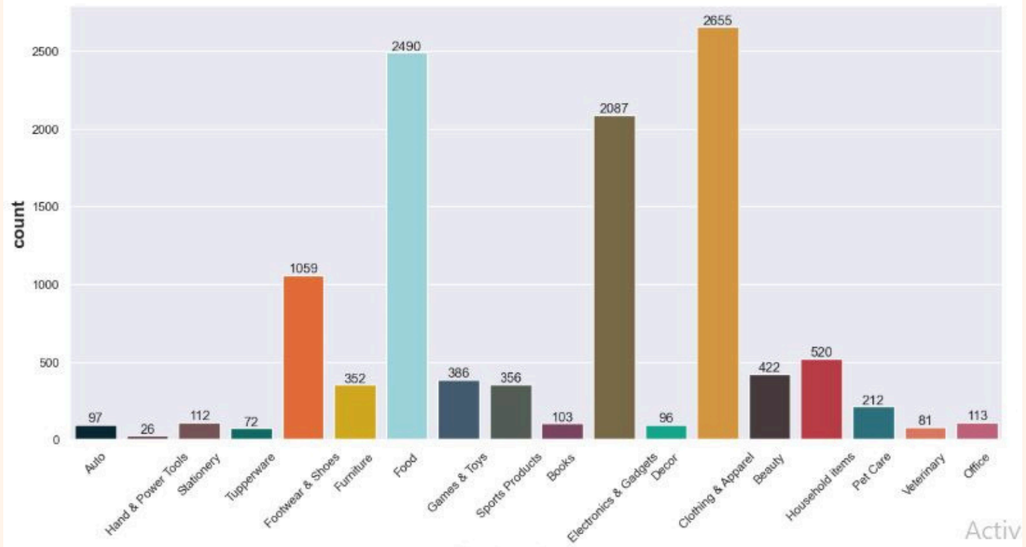
PRODUCT CATEGORY

```
In [73]: sns.set(rc = {'figure.figsize': (15,7)})

         ax = sns.countplot(x = 'Product_Category', data = df_sales_data,
                  palette = ['#022B3A','#6D435A','#7D4F50','#037971','#FE621D',
                             '#EEB902','#92DCE5','#3E5C76','#545E56','#843B62',
                             '#806D40','#00BD9D','#EC9A29','#49393B','#CC2936',
                              '#1F7A8C','#EE6C4D','#CE5374'])

         sns.set_style('white')
         plt.xlabel("Product_Category", weight = 'bold', fontsize = 16)
         plt.ylabel("count", weight = 'bold', fontsize = 16)
         plt.xticks(rotation = 45)

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

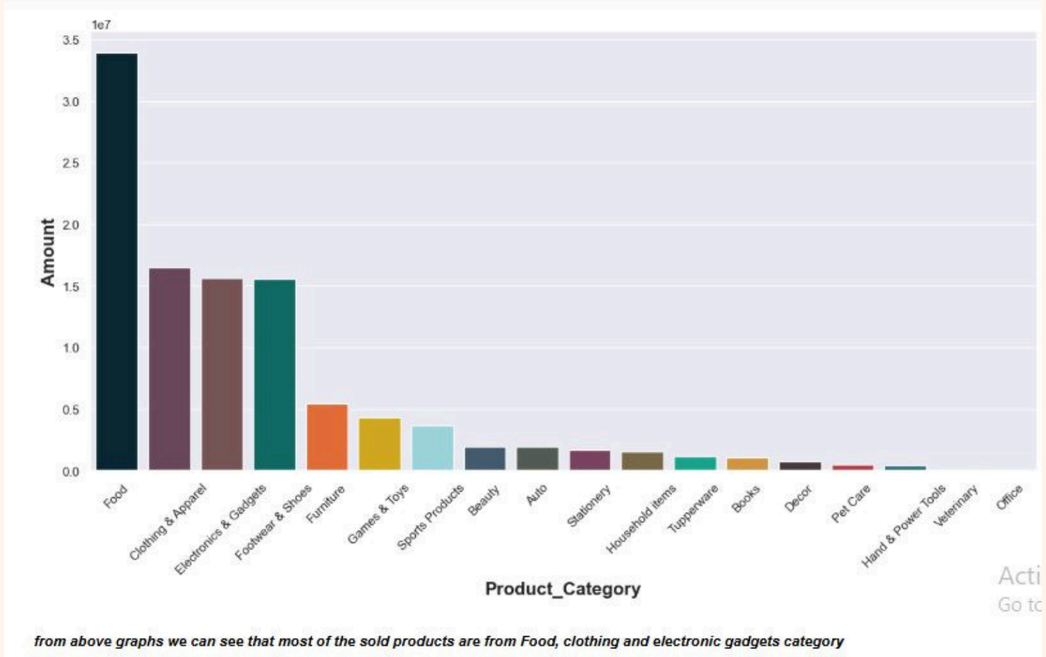


13. Total sales by Product category

```
In [63]: sales_occupation = df_sales_data.groupby(['Product_Category'], as_index = False)
                ['Amount'].sum().sort_values(by = 'Amount',ascending = False)

         sns.set(rc = {'figure.figsize': (15,7)})
         sns.barplot(x = 'Product_Category', y = 'Amount', data = sales_occupation,
                     palette = ['#022B3A','#6D435A','#7D4F50','#037971','#FE621D',
                                '#EEB902','#92DCE5','#3E5C76','#545E56','#843B62',
                                '#806D40','#00BD9D','#EC9A29','#49393B','#CC2936',
                                '#1F7A8C','#EE6C4D','#CE5374'])

         sns.set_style('white')
         plt.xlabel("Product_Category", weight = 'bold', fontsize = 16)
         plt.ylabel("Amount", weight = 'bold', fontsize = 16)
         plt.xticks(rotation = 45)
```



from above graphs we can see that most of the sold products are from Food, clothing and electronic gadgets category

Queries & Data Visualizations

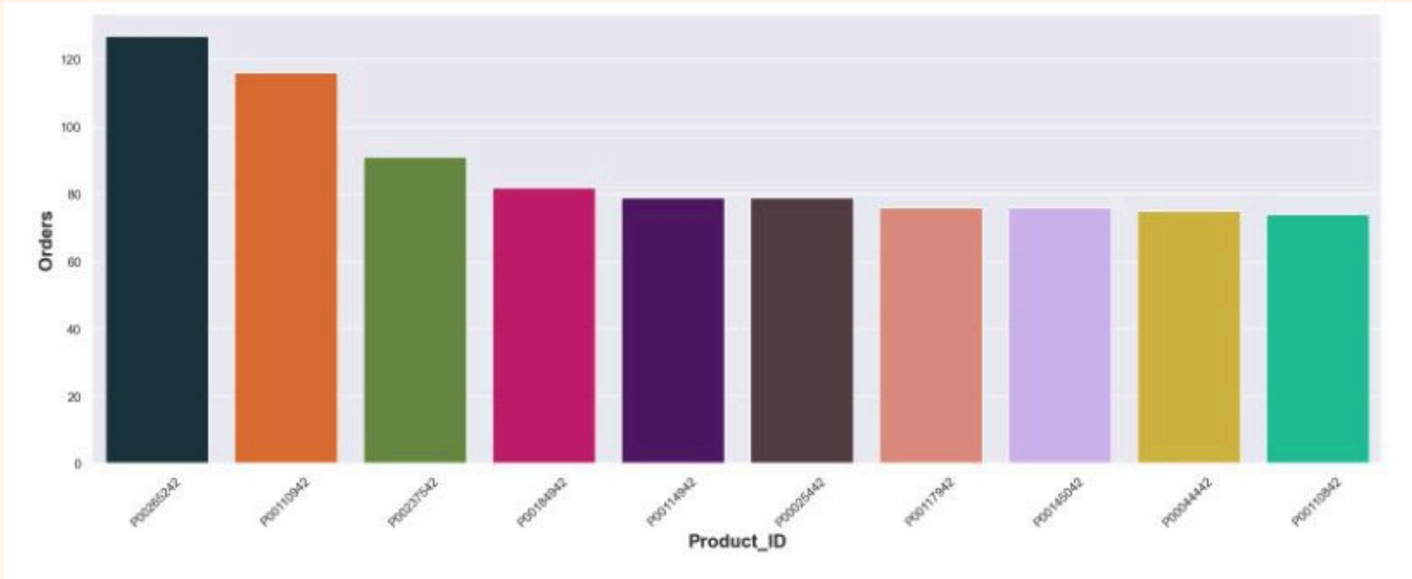
14. Top 10 most sold products

PRODUCTS

[71]:
```
# top 10 most sold products
sales_productID = df_sales_data.groupby(['Product_ID'], as_index = False)
    ['Orders'].sum().sort_values(by = 'Orders',ascending = False).head(10)

sns.set(rc = {'figure.figsize': (20,7)})
sns.barplot(x = 'Product_ID', y = 'Orders', data = sales_productID,
            palette = ['#143642','#F56416','#679436','#D90368','#540D6E',
                       '#553A41','#E9806E','#CAA8F5','#E6C229','#06D6A0'])

sns.set_style('white')
plt.xlabel("Product_ID", weight = 'bold', fontsize = 16)
plt.ylabel("Orders", weight = 'bold', fontsize = 16)
plt.xticks(rotation = 45)
```



Key insights

- High Spending Customers: Identified demographics based on gender, age group, marital status ,states ,product category and occupation .
- Top-Selling states: UP, Maharastra and Karnataka are high in demand.
- Top-Selling product categories: Food, clothing & Electronics had the highest sales.
- Sales Trends: Peak Sales observed before and during DIwali.

Conclusion

This project demonstrates how data visualization can transform raw sales data into meaningful insights, helping businesses make informed decisions for better profitability.