Comprehensive Case Study: Festive Sales Analysis and EDA

Introduction

The client, a retail company, aims to understand customer behavior and product performance during the festive season. The goal is to identify key demographics, top-selling products, and sales trends to optimize marketing strategies and inventory management for future festive seasons.

Data Overview

The dataset comprises 11,251 rows and 15 columns, containing details such as user ID, customer name, product ID, gender, age group, age, marital status, state, zone, occupation, product category, orders, and amount spent. Initial data inspection revealed two columns, `Status` and `unnamed1`, which were not relevant to the analysis and were subsequently dropped. Missing values in the `Amount` column were also addressed.

The analysis is divided into the following key sections:

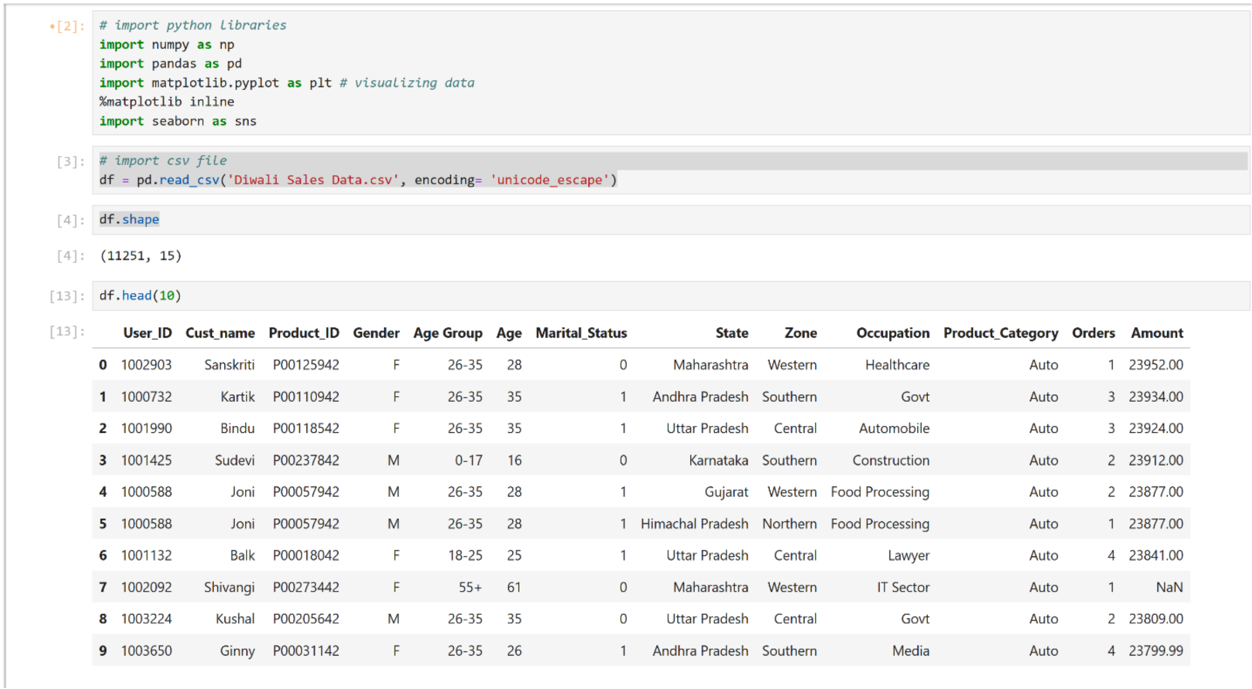
1. **Data Cleaning and Preparation**
2. **Exploratory Data Analysis (EDA)**
3. **Key Insights and Visualizations**
4. **Advanced Analysis**
5. **Conclusion**

1. Data Cleaning and Preparation

Initial steps involve loading the dataset and handling missing values. The missing values are imputed with suitable replacements or dropped if necessary. The data types are converted to appropriate formats to facilitate analysis.

- **Data Cleaning and Preprocessing**

1. **Dropping Unnecessary Columns**: The ‘`Status′` and ‘`unnamed1′` columns, which contained no data, were removed.
2. **Handling Missing Values**: Rows with missing values in the ‘`Amount′` column were dropped, resulting in a dataset of 11,239 rows and 13 columns.
3. **Data Type Conversion**: The ‘`Amount′` column's data type was converted from float to integer for consistency in analysis.
4. **Renaming Columns**: The ‘`Marital_Status′` column was renamed to ‘`Shaadi′` for a more intuitive understanding.



```
•[2]: # import python libraries
      import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt # visualizing data
      %matplotlib inline
      import seaborn as sns
```

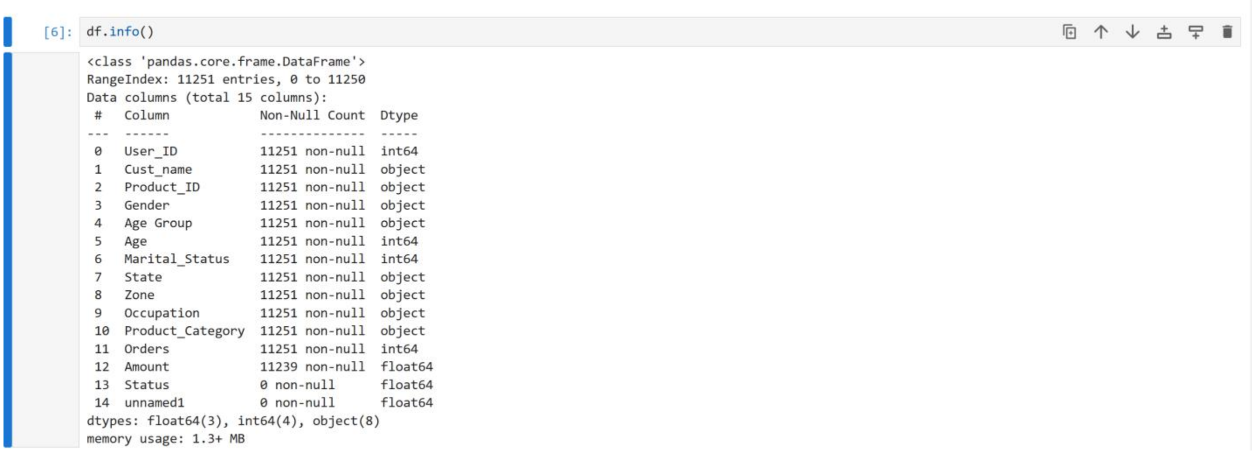
```
[3]: # import csv file
     df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

```
[4]: df.shape

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

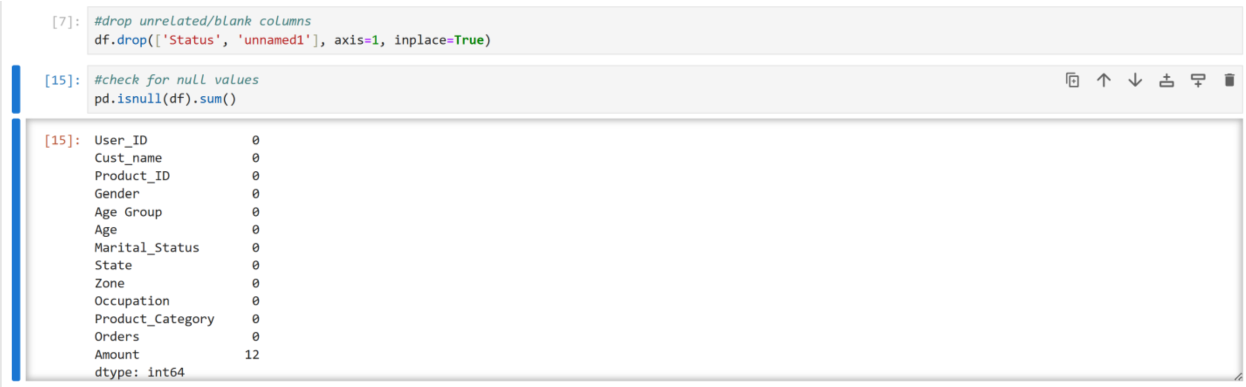
```
[13]: df.head(10)
```

[13]:

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



```
[6]: 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
```



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

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

[15]: 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
```

```
[16]: df.shape
```

```
[16]: (11251, 13)
```

```
[17]: # drop null values
      df.dropna(inplace=True)
```

```
[18]: df.shape
```

```
[18]: (11239, 13)
```

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

```
[19]: 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
```

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

```
[21]: df['Amount'].dtypes
```

```
[21]: dtype('int32')
```

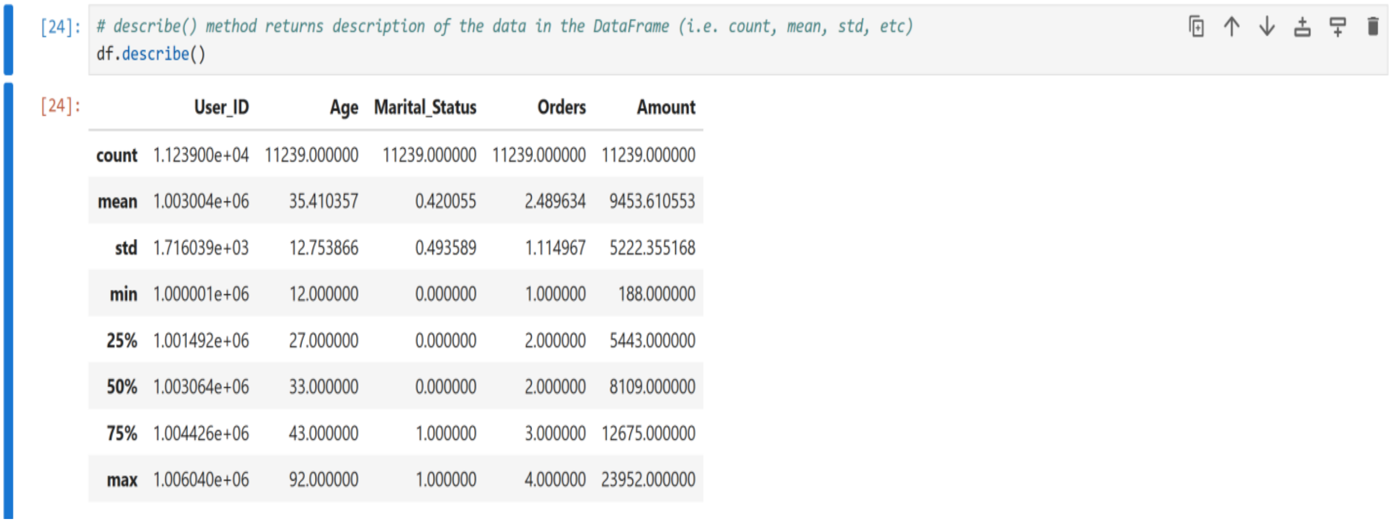
```
[22]: df.columns
```

```
[23]: #rename column
      df.rename(columns= {'Marital_Status':'Shaadi'})
```

[23]:

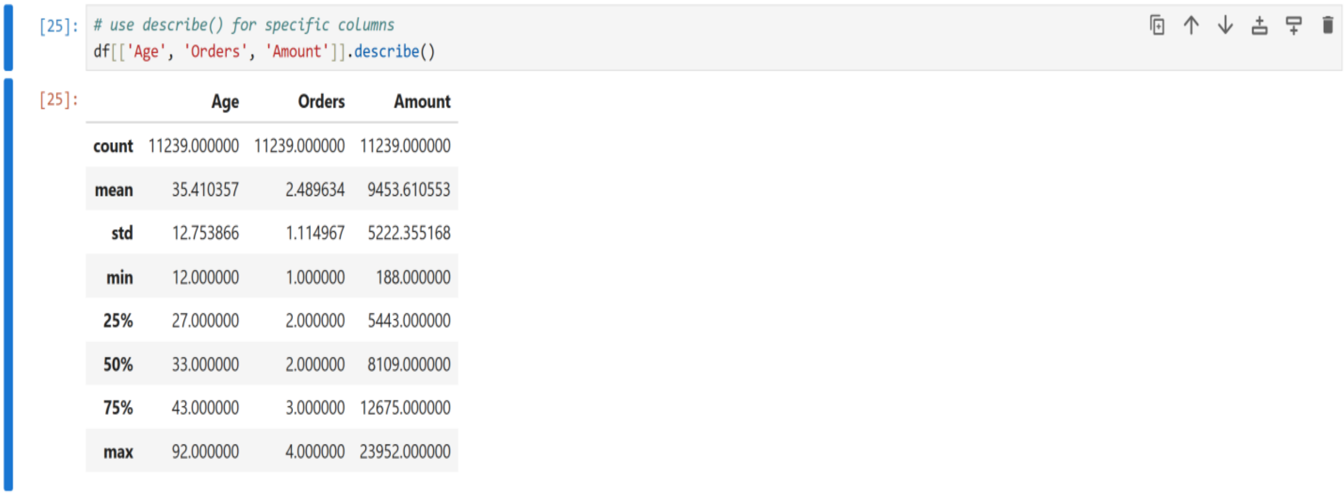
	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237842	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	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns



[24]:
```
# describe() method returns description of the data in the DataFrame (i.e. count, mean, std, etc)
df.describe()
```

[24]:
```
            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
```



[25]:
```
# use describe() for specific columns
df[['Age', 'Orders', 'Amount']].describe()
```

[25]:
```
                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
```

2. Exploratory Data Analysis (EDA)

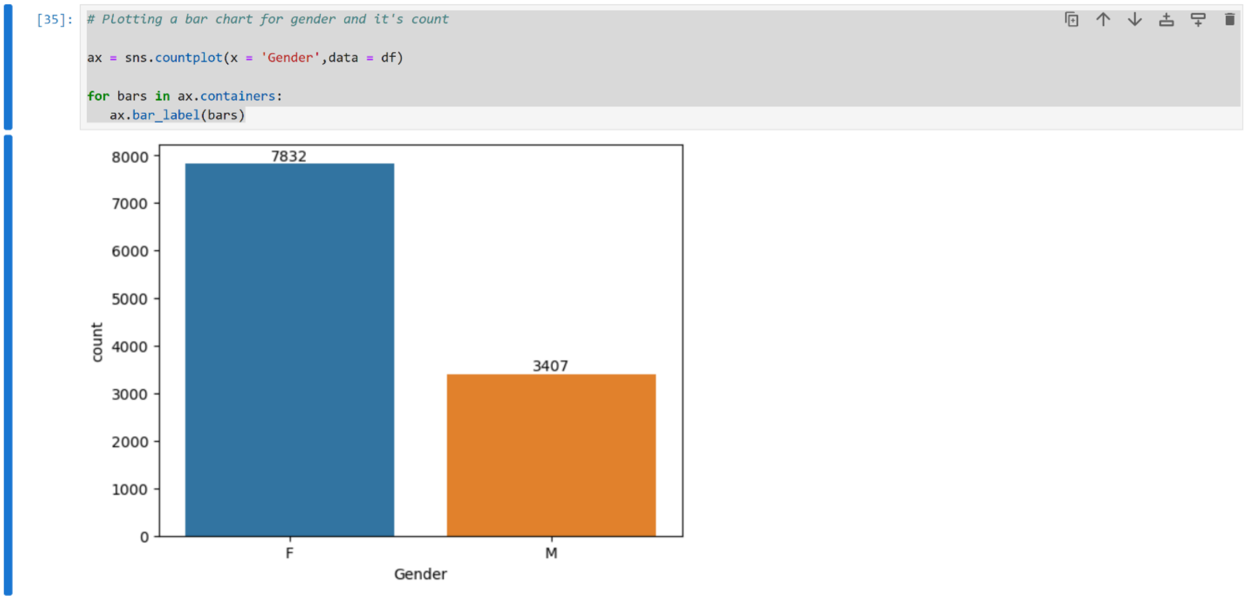
- ➢ EDA is performed to uncover patterns, trends, and relationships within the data. EDA phase involved visualizing various aspects of the data to uncover insights.

Key areas of focus include:

- **Demographic Analysis:** Understanding the distribution of customers based on age, gender, marital status, and occupation.
- **Geographical Analysis:** Analyzing sales trends across different city categories and the duration of customer stay in their current city.
- **Product Analysis:** Identifying the most popular product categories and the top-selling products.
- **Sales Trends:** Examining purchase behavior across different age groups, occupations, and marital statuses.

Gender Distribution:

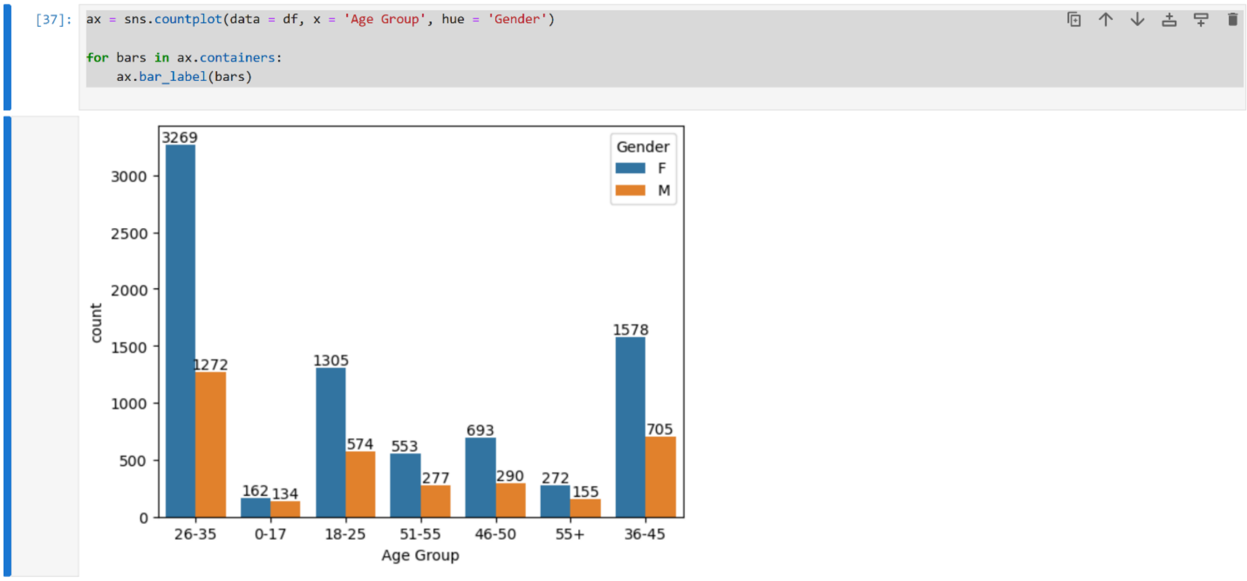
- Most buyers were females, indicating higher purchasing power among women during the festive season.



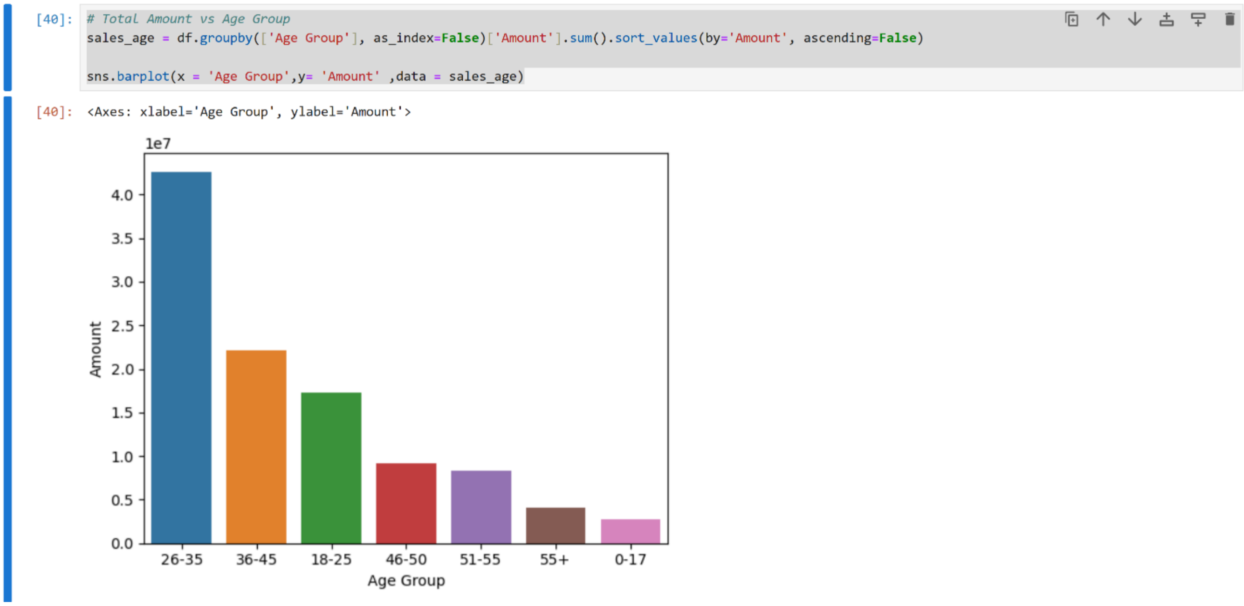
```
[35]: # Plotting a bar chart for gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

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

Age Distribution:

- The 26-36 age group had the highest number of buyers and total sales, making it the most significant age segment.



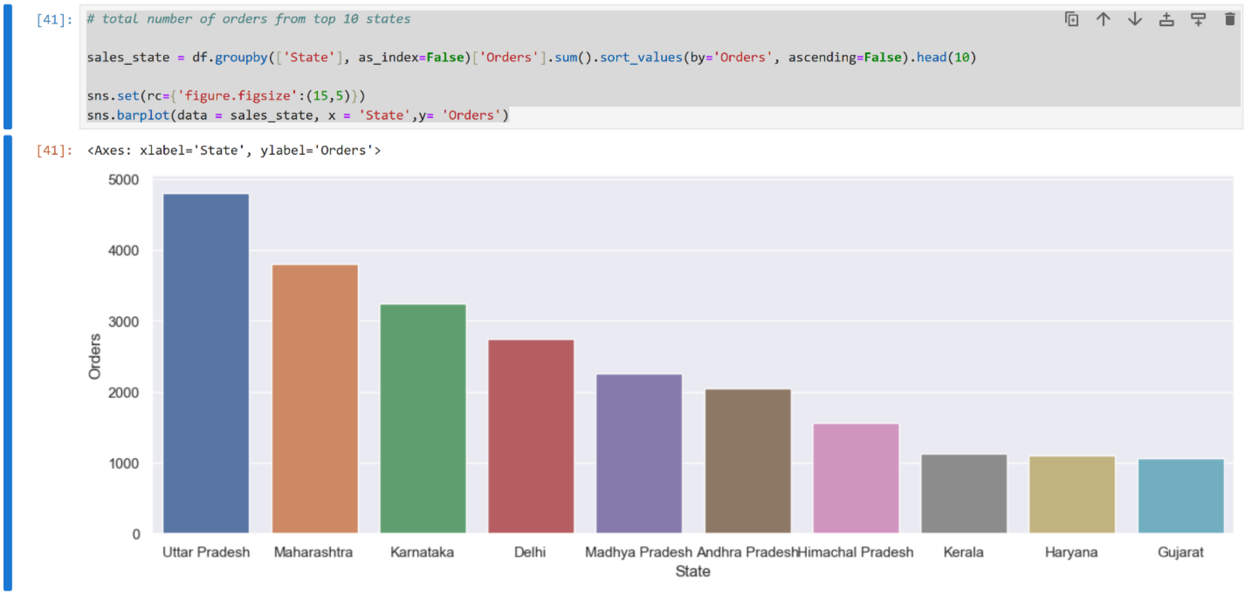
```
[37]: ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

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



```
[40]: # Total Amount vs 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)
```

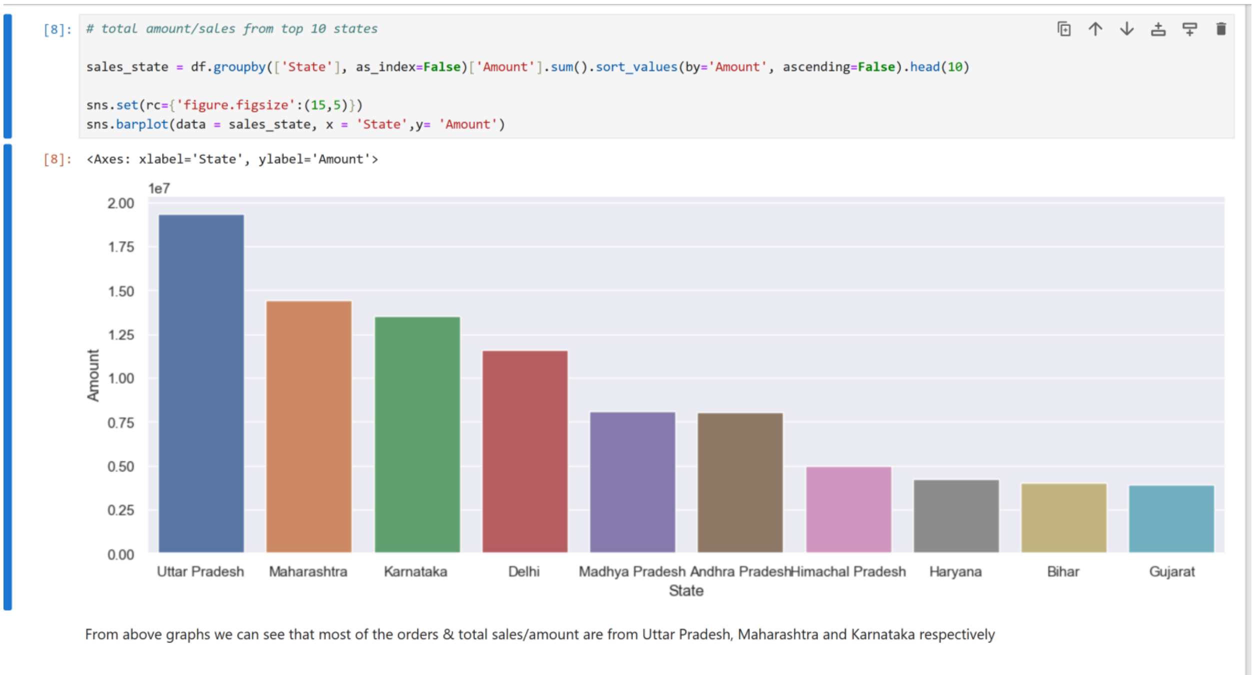
```
[40]: <Axes: xlabel='Age Group', ylabel='Amount'>
```

- **Geographical Distribution**:
  - **Top 10 States by Orders**: Uttar Pradesh, Maharashtra, and Karnataka led in the number of orders.
  - **Top 10 States by Sales Amount**: These states also had the highest sales amounts, emphasizing their importance as major markets.



```
[41]: # total number of orders from top 10 states

      sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

      sns.set(rc={'figure.figsize':(15,5)})
      sns.barplot(data = sales_state, x = 'State',y= 'Orders')
```

```
[41]: <Axes: xlabel='State', ylabel='Orders'>
```



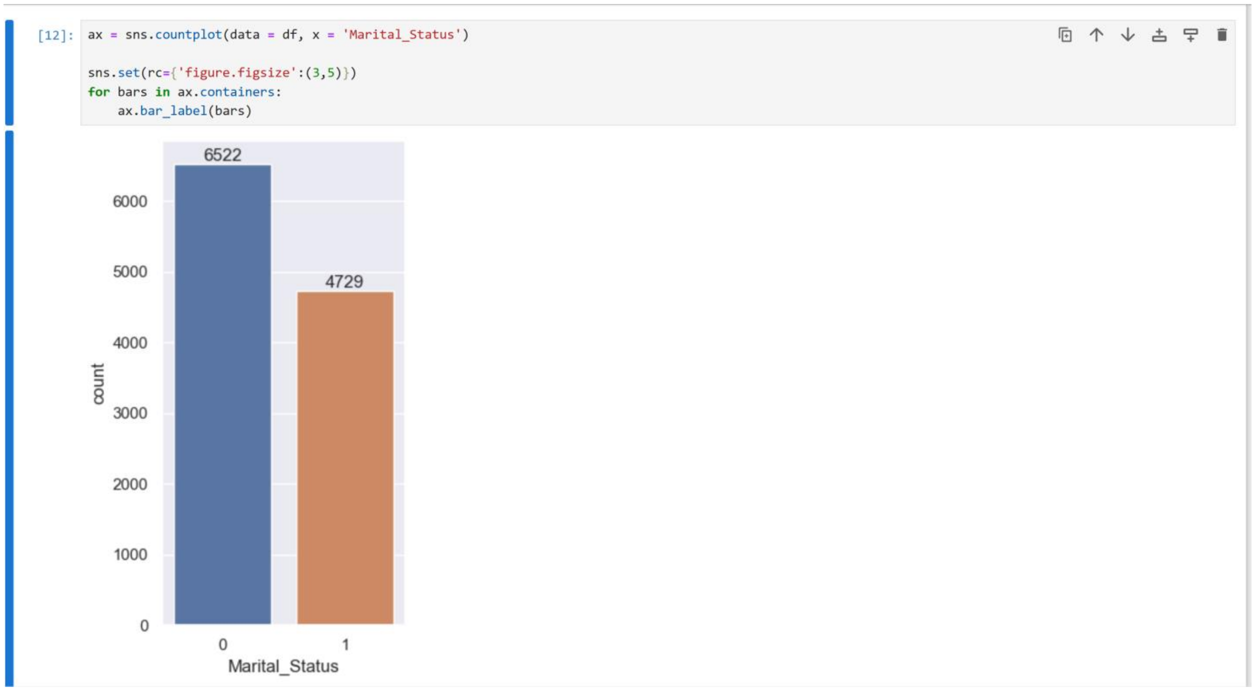
```
[8]: # total amount/sales from top 10 states

     sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

     sns.set(rc={'figure.figsize':(15,5)})
     sns.barplot(data = sales_state, x = 'State',y= 'Amount')
```

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

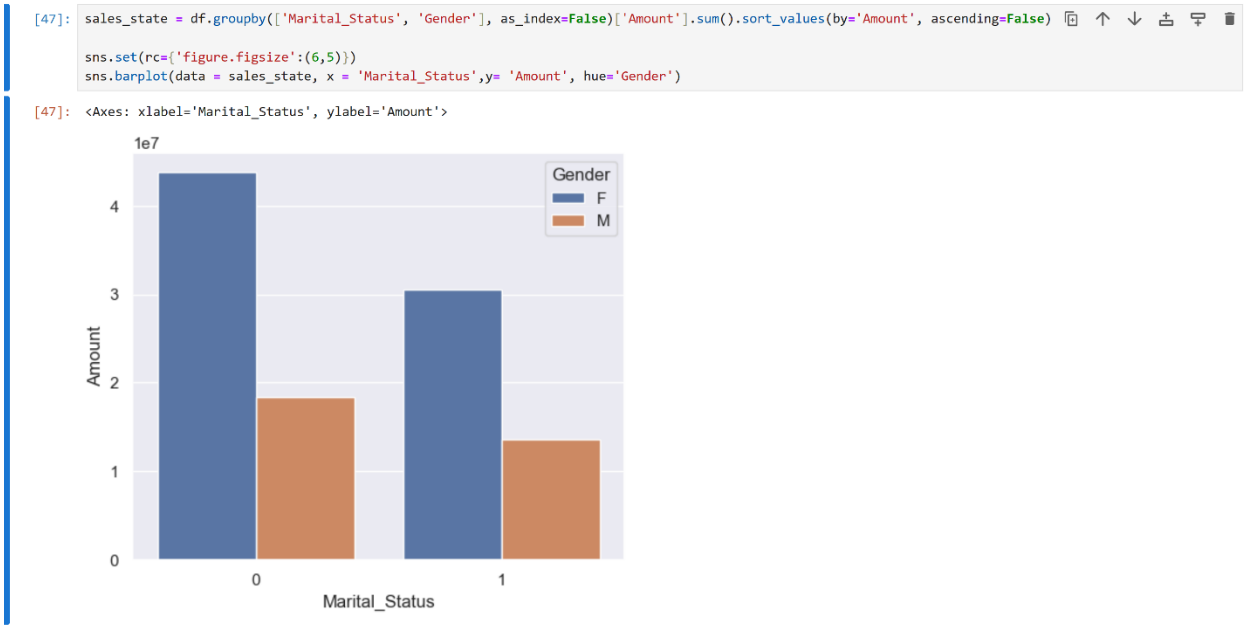
From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

Marital Status:

- Married individuals, particularly women, had a higher purchasing power, with significant contributions to total sales.



```
[12]: ax = sns.countplot(data = df, x = 'Marital_Status')

      sns.set(rc={'figure.figsize':(3,5)})
      for bars in ax.containers:
          ax.bar_label(bars)
```

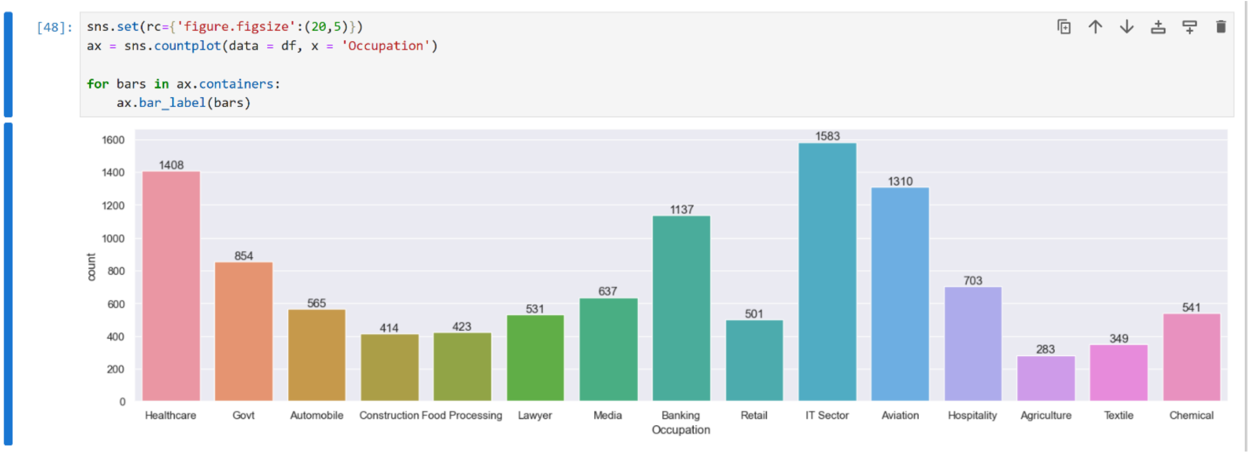


```
[47]: sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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

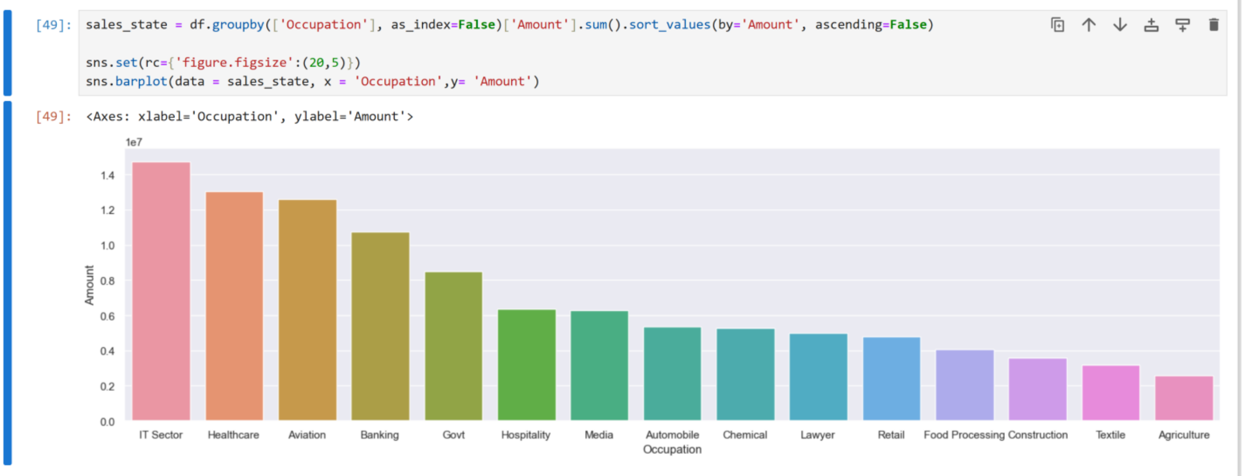
```
[47]: <Axes: xlabel='Marital_Status', ylabel='Amount'>
```

Occupation:

- **Distribution of Buyers by Occupation**: Most buyers were employed in IT, Healthcare, and Aviation sectors.
- **Total Sales by Occupation**: These occupations also accounted for the highest sales, indicating higher disposable incomes.



```
[48]: sns.set(rc={'figure.figsize':(20,5)})
      ax = sns.countplot(data = df, x = 'Occupation')

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

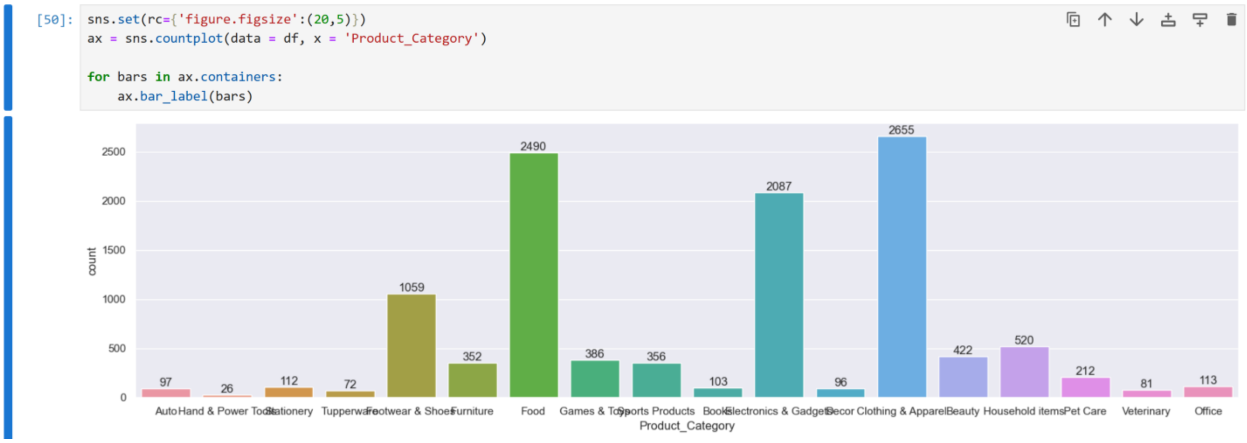


```
[49]: sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

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

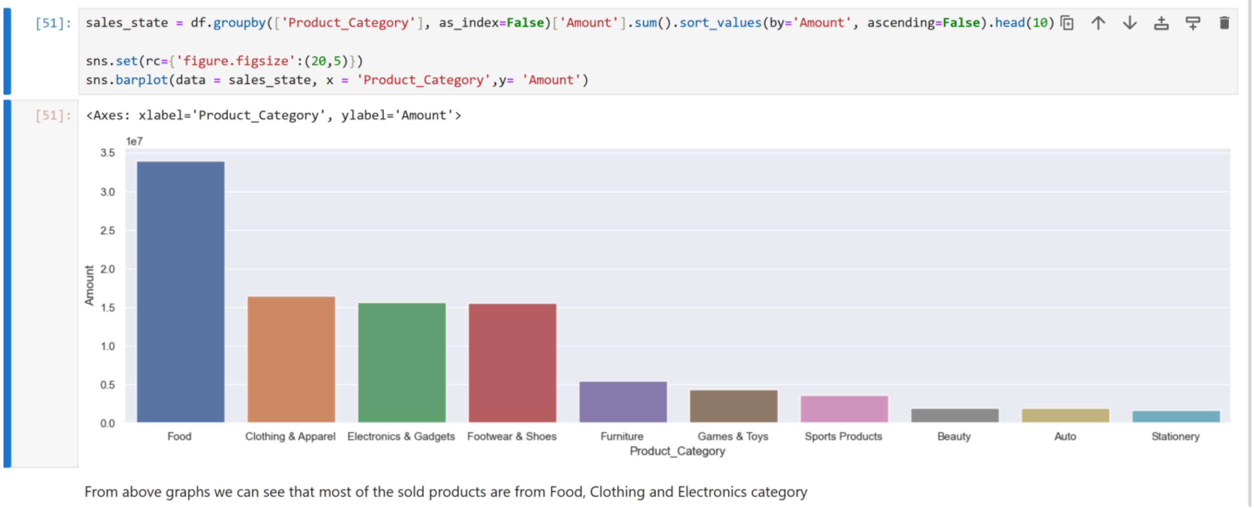
```
[49]: <Axes: xlabel='Occupation', ylabel='Amount'>
```

Product Categories:

- **Distribution of Orders by Product Category**: Food, Clothing, and Electronics were the most popular categories.



```
[50]: sns.set(rc={'figure.figsize':(20,5)})
      ax = sns.countplot(data = df, x = 'Product_Category')

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

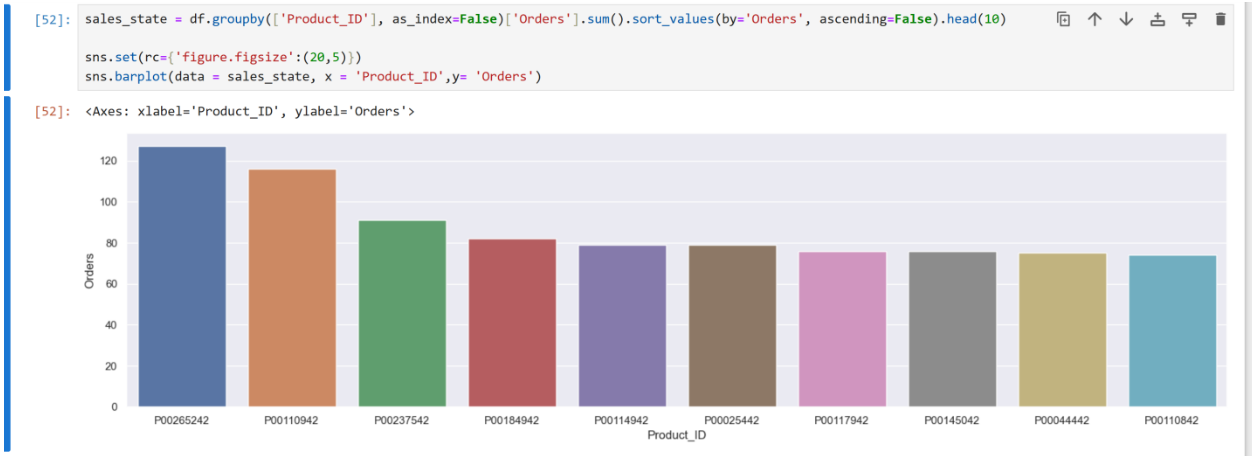


```
[51]: sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

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

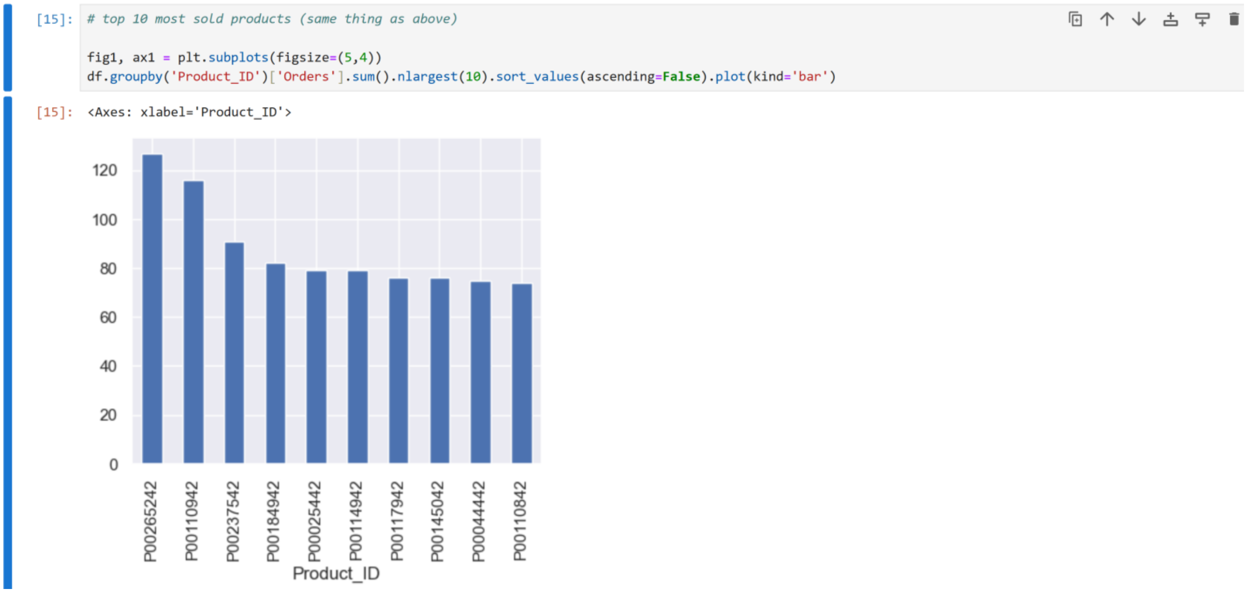
```
[51]: <Axes: xlabel='Product_Category', ylabel='Amount'>
```

From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

○ **Top 10 Most Sold Products**: Detailed analysis of the top-selling products provided insights into consumer preferences.



```
[52]: sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

      sns.set(rc={'figure.figsize':(20,5)})
      sns.barplot(data = sales_state, x = 'Product_ID',y= 'Orders')

[52]: <Axes: xlabel='Product_ID', ylabel='Orders'>
```



```
[15]: # top 10 most sold products (same thing as above)

      fig1, ax1 = plt.subplots(figsize=(5,4))
      df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')

[15]: <Axes: xlabel='Product_ID'>
```

Insights and Recommendations

Based on the analysis, several key insights were derived:

- **Target Demographics**: Marketing strategies should focus on females aged 26-35, particularly those who are married, as they represent the most significant customer segment.
- **Geographical Focus**: Promotional efforts should be concentrated in Uttar Pradesh, Maharashtra, and Karnataka to capitalize on these high-performing regions.
- **Occupation-Based Marketing**: Customized marketing campaigns targeting IT, Healthcare, and Aviation professionals could enhance sales, given their higher purchasing power.
- **Product Category Promotion**: Emphasizing products in the Food, Clothing, and Electronics categories can drive higher sales, given their popularity among buyers.

Conclusion

To conclude, the analysis of festive sales data highlights the pivotal role of understanding consumer demographics in refining marketing strategies. Through rigorous data cleaning, preprocessing, and in-depth exploration using Python, I've uncovered valuable insights. The attached results and visuals vividly illustrate trends such as age-specific purchasing behaviors, geographic preferences, and popular product categories.

This analytical approach equips my client with actionable intelligence to drive strategic marketing decisions. By honing in on specific segments, like married women aged 26-36, and leveraging insights into regional dynamics and occupational preferences, our client can significantly amplify sales during crucial festive periods. These findings not only guide efficient marketing expenditure but also enhance the precision and impact of promotional efforts.

In essence, this case study underscores the transformative potential of data analytics. It empowers businesses to achieve sustained growth and maximize profitability by ensuring targeted marketing strategies that resonate with consumer preferences and behaviors.