Introduction:

- Welcome to our Diwali sales store project analysis, where we delve into the intricacies of consumer behavior and sales trends to optimize business strategies. Through the lens of Python data analysis, we uncover valuable insights to enhance our understanding of customer preferences and maximize sales potential during the festive season.

Problem Statement:

- In the realm of Diwali sales, understanding consumer behavior and identifying key market segments are essential for driving business success. Our challenge lies in deciphering the primary consumer base, recognizing demographic patterns, and pinpointing regions where sales thrive the most. Additionally, we aim to explore the influence of marital status and profession on buying behavior while identifying the most popular product segments among buyers.

```
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns

# Import necessary libraries
import pandas as pd
# Read the CSV file
df = pd.read_csv('"C:\Users\Dell\Downloads\Python Project\Diwali Sales Data.csv"',
encoding = 'unicode_escape' )

df = pd.read_csv(r"C:\Users\Dell\Downloads\Python Project\Diwali Sales Data.csv",
encoding = 'unicode_escape' )
df.shape
```

```
(11251, 15)
```

```
df.head( 20)
```

```
 User_ID Cust_name Product_ID Gender Age Group Age Marital_Status \
0 1002903 Sanskriti P00125942 F 26-35 28 0
1 1000732 Kartik P00110942 F 26-35 35 1
2 1001990 Bindu P00118542 F 26-35 35 1
3 1001425 Sudevi P00237842 M 0-17 16 0
4 1000588 Joni P00057942 M 26-35 28 1
5 1000588 Joni P00057942 M 26-35 28 1
6 1001132 Balk P00018042 F 18-25 25 1
7 1002092 Shivangi P00273442 F 55+ 61 0
8 1003224 Kushal P00205642 M 26-35 35 0
9 1003650 Ginny P00031142 F 26-35 26 1
10 1003829 Harshita P00200842 M 26-35 34 0
11 1000214 Kargatis P00119142 F 18-25 20 0
12 1004035 Elijah P00080342 F 18-25 20 1
13 1001680 Vasudev P00324942 M 26-35 26 1
14 1003858 Cano P00293742 M 46-50 46 1
15 1000813 Lauren P00289942 F 18-25 24 0
16 1005447 Amy P00275642 F 46-50 48 1
17 1001193 Mick P00004842 F 26-35 29 0
18 1001883 Praneet P00029842 M 51-55 54 1
19 1001883 Praneet P00029842 M 51-55 54 1

State Zone Occupation Product_Category Orders \
0 Maharashtra Western Healthcare Auto 1
1 Andhra Pradesh Southern Govt Auto 3
2 Uttar Pradesh Central Automobile Auto 3
3 Karnataka Southern Construction Auto 2
4 Gujarat Western Food Processing Auto 2
5 Himachal Pradesh Northern Food Processing Auto 1
6 Uttar Pradesh Central Lawyer Auto 4
7 Maharashtra Western IT Sector Auto 1
8 Uttar Pradesh Central Govt Auto 2
9 Andhra Pradesh Southern Media Auto 4
10 Delhi Central Banking Auto 1
11 Andhra Pradesh Southern Retail Auto 2
12 Andhra Pradesh Southern IT Sector Auto 2
13 Andhra Pradesh Southern Automobile Auto 4
14 Madhya Pradesh Central Hospitality Auto 3
15 Andhra Pradesh Southern Govt Auto 2
16 Andhra Pradesh Southern IT Sector Auto 3
17 Andhra Pradesh Southern Aviation Auto 1
18 Uttar Pradesh Central Hospitality Auto 1
19 Uttar Pradesh Central Hospitality Auto 1

Amount Status unnamed1
0 23952.00 NaN NaN
1 23934.00 NaN NaN
2 23924.00 NaN NaN
3 23912.00 NaN NaN
4 23877.00 NaN NaN
5 23877.00 NaN NaN
6 23841.00 NaN NaN
7 NaN NaN NaN
8 23809.00 NaN NaN
9 23799.99 NaN NaN
10 23770.00 NaN NaN
11 23752.00 NaN NaN
12 23730.00 NaN NaN
```

```
13 23718.00 NaN NaN
14 NaN NaN NaN
15 23664.00 NaN NaN
16 NaN NaN NaN
17 23619.00 NaN NaN
18 23568.00 NaN NaN
19 23568.00 NaN NaN
```

```
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
```

```
#drop unrelated/blank columns
df.drop(['Status','unnamed1'],axis=1, inplace=True)
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 11239 non-null float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

```
pd.isnull(df).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
```

```
df.shape
```

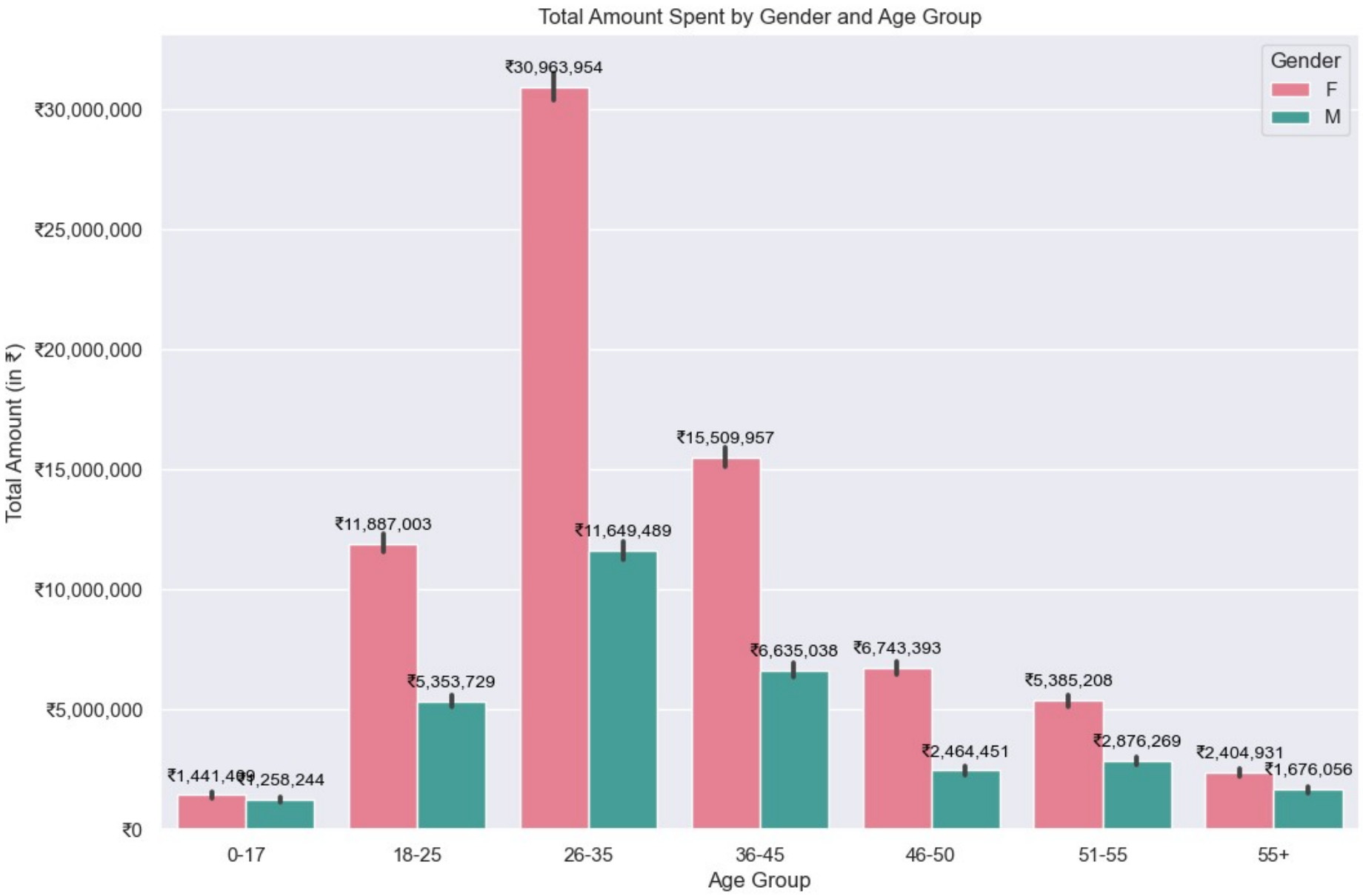
```
(11239, 13)
```

```
#drop null values
df.dropna(inplace=True)
```

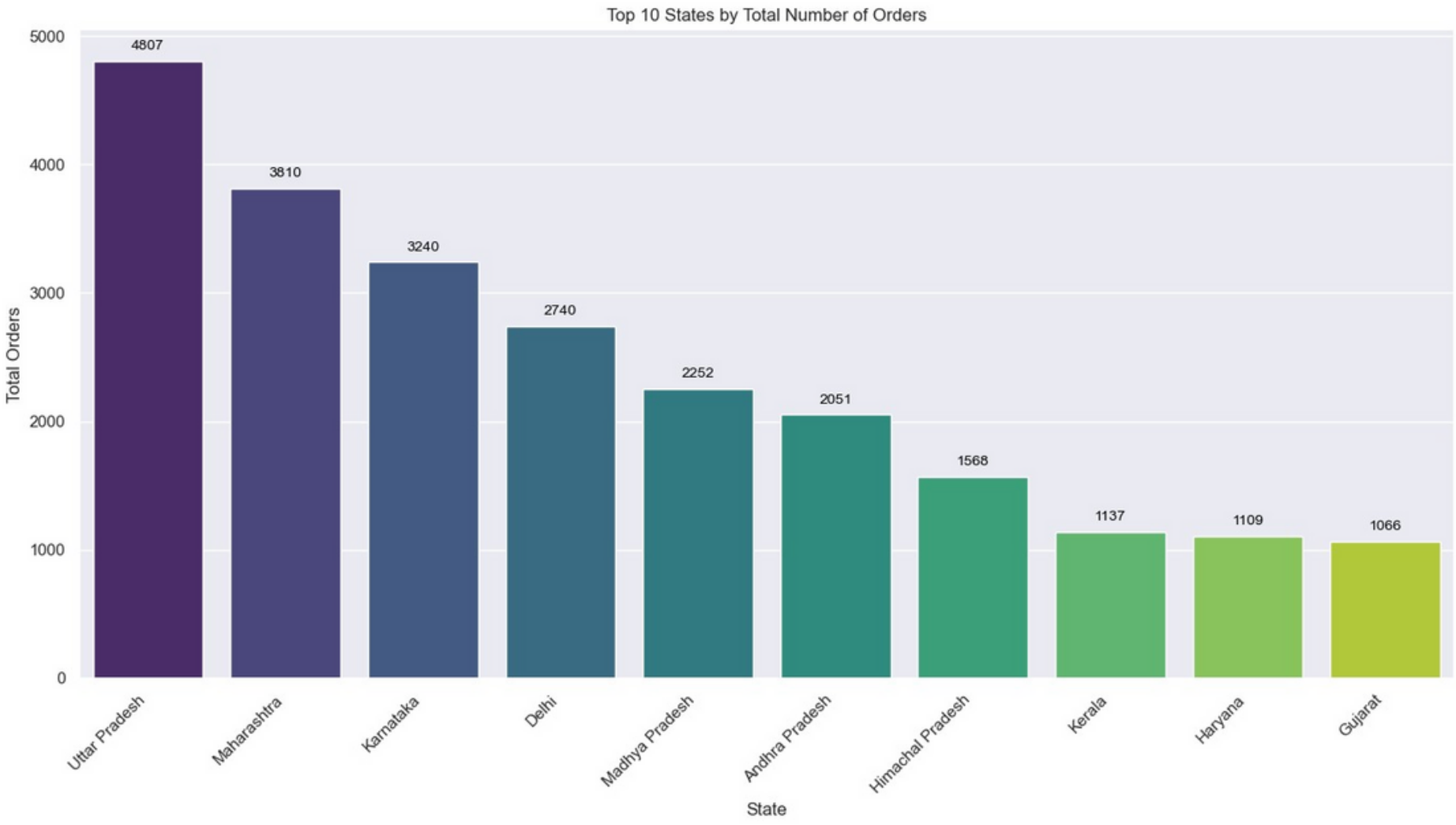
```
#change data type
df['Amount'] = df['Amount'].astype('int')

df[ 'Orders' ].dtypes

dtype('int64')

df.columns

Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
 'Orders', 'Amount'],
 dtype='object')
# rename column
df.rename(columns= {'Age':'Umar'})
 User_ID Cust_name Product_ID Gender Age Group Umar Marital_Status \
0 1002903 Sanskriti P00125942 F 26-35 28 0
1 1000732 Kartik P00110942 F 26-35 35 1
2 1001990 Bindu P00118542 F 26-35 35 1
3 1001425 Sudevi P00237842 M 0-17 16 0
4 1000588 Joni P00057942 M 26-35 28 1
... ... ... ... ... ... ... ...
11246 1000695 Manning P00296942 M 18-25 19 1
11247 1004089 Reichenbach P00171342 M 26-35 33 0
11248 1001209 Oshin P00201342 F 36-45 40 0
11249 1004023 Noonan P00059442 M 36-45 37 0
11250 1002744 Brumley P00281742 F 18-25 19 0
State Zone Occupation Product_Category Orders \
0 Maharashtra Western Healthcare Auto 1
1 Andhra Pradesh Southern Govt Auto 3
2 Uttar Pradesh Central Automobile Auto 3
3 Karnataka Southern Construction Auto 2
4 Gujarat Western Food Processing Auto 2
... ... ... ... ... ...
11246 Maharashtra Western Chemical Office 4
11247 Haryana Northern Healthcare Veterinary 3
11248 Madhya Pradesh Central Textile Office 4
11249 Karnataka Southern Agriculture Office 3
11250 Maharashtra Western Healthcare Office 3
Amount
0 23952
1 23934
2 23924
3 23912
4 23877
... ...
11246 370
11247 367
11248 213
11249 206
11250 188
[11239 rows x 13 columns]

import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Assuming 'df' is your DataFrame with the required columns ('Gender' and 'Amount')

plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Gender', y='Amount', data=df, estimator=sum)
# Function to format y-axis labels with Indian Rupee symbol
def rupee_formatter(x, pos):
 return f'₹{int(x):,}'

# Apply the formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(rupee_formatter))
# Add data labels with rupee symbols
for p in ax.patches:
 ax.annotate(f'₹{int(p.get_height()):,}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')
```

```
plt.title('Total Amount Spent by Gender')
plt.ylabel('Total Amount (in ₹)')
plt.show()
```

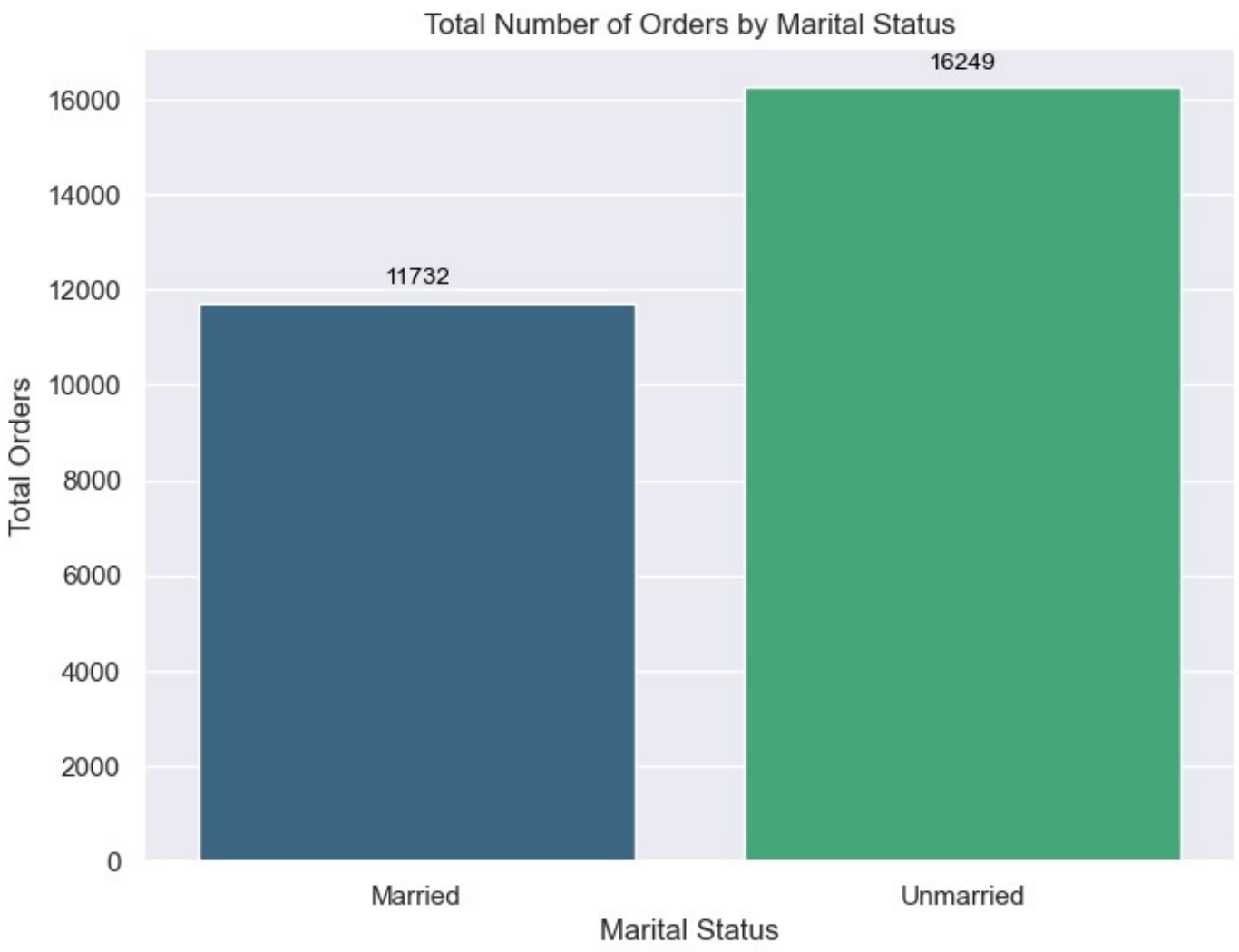

```
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Assuming 'df' is your DataFrame with the required columns ('Gender', 'Age Group', and 'Amount')

# Define the chronological order for 'Age Group'
age_group_order = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='Age Group', y='Amount', hue='Gender', data=df, estimator=sum, palette='husl',
order =age_group_order)

# Function to format y-axis labels with Indian Rupee symbol
def rupee_formatter(x, pos):
 return f'₹{int(x):,}'

# Apply the formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(rupee_formatter))

# Add data labels with rupee symbols
for p in ax.patches:
 ax.annotate(f'₹{int(p.get_height()):,}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')

plt.title('Total Amount Spent by Gender and Age Group')
plt.ylabel('Total Amount (in ₹)')
plt.xlabel('Age Group')
plt.lege nd(t itle = 'Gender' )
plt.show()
```



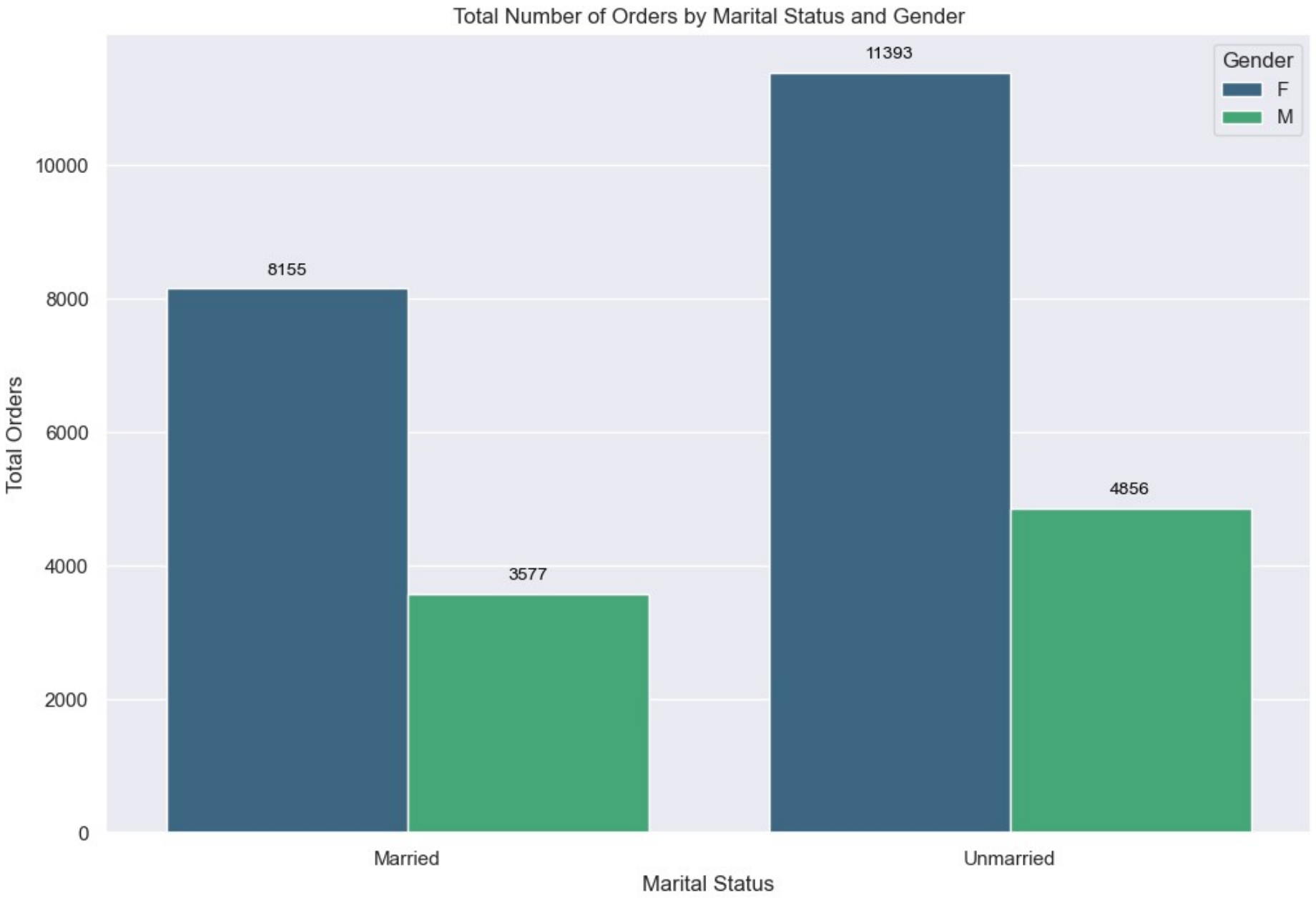

```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'df' is your DataFrame with the required columns ('State' and 'Orders')

# Calculate total orders by state
total_orders_by_state = df.groupby('State')['Orders'].sum().reset_index()
# Sort and select the top 10 states
top_10_states = total_orders_by_state.sort_values(by='Orders', ascending=False).head(10)
# Bar Chart showing top 10 states by total number of orders with data labels
plt.figure(figsize=(14, 8))
ax = sns.barplot(x='State', y='Orders', data=top_10_states, palette='viridis', capsize=0.1, errwidth=1.5)

# Rotate x-axis labels for better readability
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, horizontalalignment='right')
# Adding data labels
for p in ax.patches:
 ax.annotate(f'{p.get_height():.0f}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')

plt.title('Top 10 States by Total Number of Orders')
plt.ylabel('Total Orders')
# Adjust layout to prevent clipping of rotated x-axis labels
plt.tight_layout()
plt.show()
```



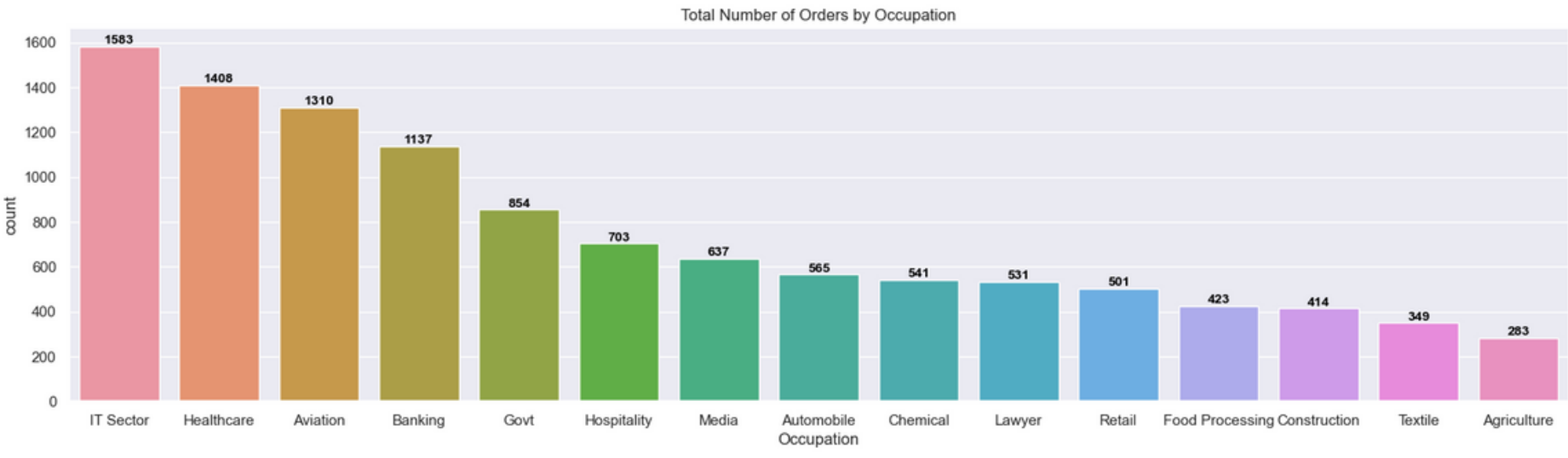

```
# Replace numeric values with labels
df['Marital_Status'] = df['Marital_Status'].replace({0: 'Unmarried', 1: 'Married'})

# Calculate total orders by marital status
total_orders_by_marital_status = df.groupby('Marital_Status')['Orders'].sum().reset_index()
# Bar Chart showing total number of orders by marital status with data labels
plt.figure(figsize=(8, 6))
ax = sns.barplot(x='Marital_Status', y='Orders', data=total_orders_by_marital_status, palette='viridis')

# Adding data labels
for p in ax.patches:
 ax.annotate(f'{p.get_height():.0f}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')

plt.title('Total Number of Orders by Marital Status')
plt.ylabel('Total Orders')
plt.xlabel('Marital Status')

plt.show()
```




```
# Replace 0 with 'Unmarried' and 1 with 'Married' in the 'Marital_Status' column
df['Marital_Status'] = df['Marital_Status'].replace({0: 'Unmarried', 1: 'Married'})

# Calculate total orders by marital status and gender
total_orders_by_marital_gender = df.groupby(['Marital_Status', 'Gender'])['Orders'].sum().reset_index()
# Bar Chart showing total number of orders by marital status and gender with data labels
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='Marital_Status', y='Orders', hue='Gender', data=total_orders_by_marital_gender,
palette='viridis')
# Adding data labels
for p in ax.patches:
 ax.annotate(f'{p.get_height():.0f}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')

plt.title('Total Number of Orders by Marital Status and Gender')
plt.ylabel('Total Orders')
plt.xlabel('Marital Status')
plt.legend(title='Gender', loc='upper right')
plt.show()
```

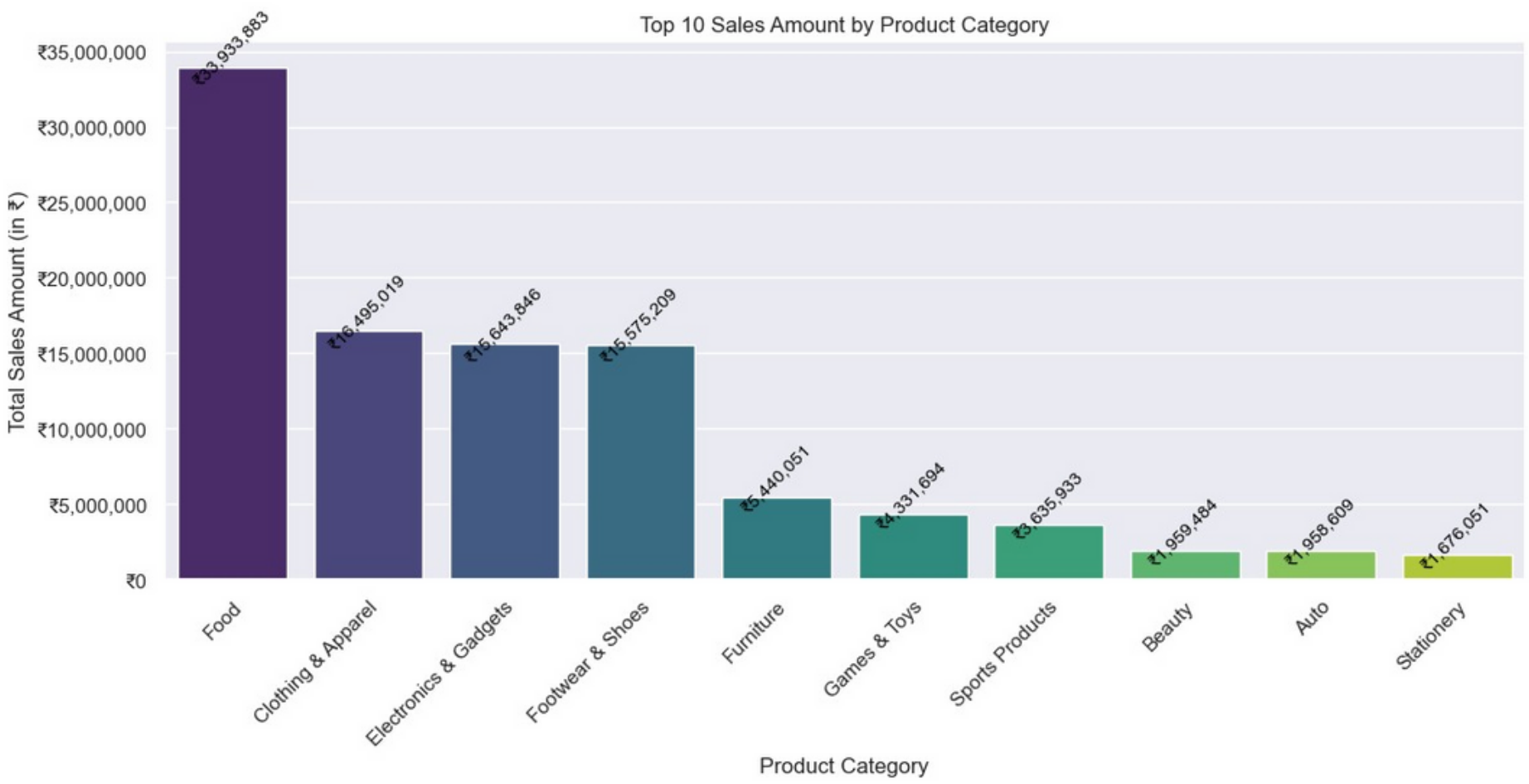



```
sns.set(rc={'figure.figsize': (20, 5)})

ax = sns.countplot(data=df, x='Occupation', order=df['Occupation'].value_counts().index)

for bars in ax.containers:
 ax.bar_label(bars, fmt='%.0f', label_type='edge', fontsize=10, color='black', weight='bold')
plt.title('Total Number of Orders by Occupation')
plt.xlabel( 'Occupation' )
plt.show()
```



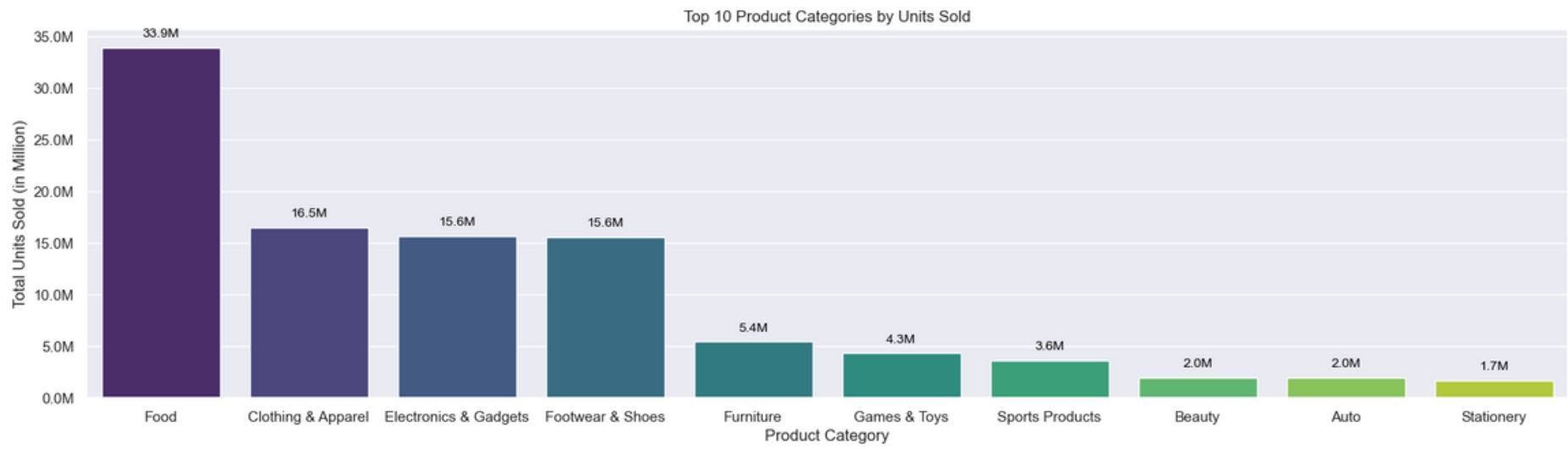

```
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Assuming 'df' is your DataFrame with the required columns ('Product_Category' and 'Amount')

# Calculate total sales amount for each product category
sales_by_category = df.groupby('Product_Category', as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)

# Select the top 10 product categories
top_10_categories = sales_by_category.head(10)
# Set the figure size
sns.set(rc={'figure.figsize':(12, 6)})
```

```
# Create a bar chart for the top 10 categories
ax = sns.barplot(data=top_10_categories, x='Product_Category', y='Amount', palette='viridis')

# Function to format y-axis labels with Indian Rupee symbol
def rupee_formatter(x, pos):
 return f'₹{int(x):,}'

# Apply the formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(rupee_formatter))
# Rotate x-axis labels and adjust spacing
plt.xticks(rotation=45, ha='right', rotation_mode='anchor')
plt.tight_layout()

# Add data labels with rupee symbols
for p in ax.patches:
 ax.annotate(f'₹{int(p.get_height()):,}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points', rotation=45, rotation_mode='anchor')

# Set plot labels and title
plt.xlabel('Product Category')
plt.ylabel('Total Sales Amount (in ₹)')
plt.title('Top 10 Sales Amount by Product Category')
# Show the plot
plt.show()
```

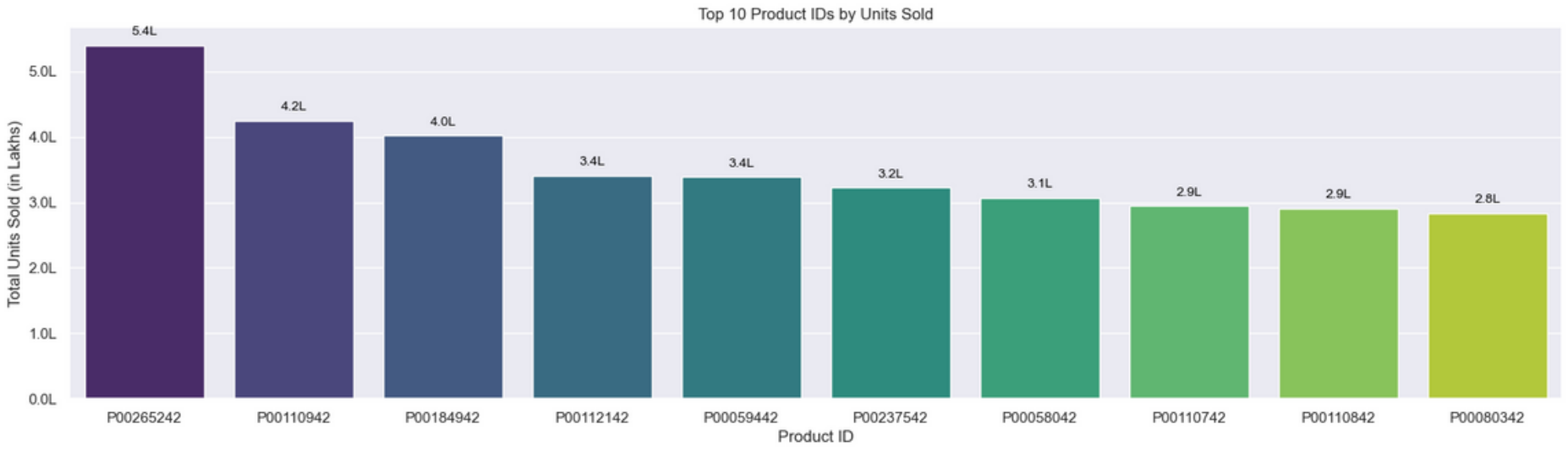



```
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Assuming 'df' is your DataFrame with the required columns ('Product_Category' and 'Amount')

# Calculate total units sold for each product category, sort, and select the top 10
units_sold_state = df.groupby('Product_Category', as_index=False)['Amount'].sum().nlargest(10, 'Amount')
# Set the figure size
sns.set(rc={'figure.figsize':(20,5)})
# Create a bar chart
ax = sns.barplot(data=units_sold_state, x='Product_Category', y='Amount', palette='viridis')
# Function to format y-axis labels in million units
def millions_formatter(x, pos):
 return f'{x/1e6:.1f}M'

# Apply the formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(millions_formatter))
```

```
# Add data labels
for p in ax.patches:
 ax.annotate(f'{p.get_height()/1e6:.1f}M', (p.get_x() + p.get_width() / 2., p.get_height()),  ha='center',
va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')

# Set plot labels and title
plt.xlabel('Product Category')
plt.ylabel('Total Units Sold (in Million)')
plt.title('Top 10 Product Categories by Units Sold')
# Show the plot
plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Assuming 'df' is your DataFrame with the required columns ('Product_ID' and 'Amount')

# Calculate total units sold for each product ID, sort, and select the top 10
units_sold_product_id = df.groupby('Product_ID', as_index=False)['Amount'].sum().nlargest(10, 'Amount')
# Set the figure size
sns.set(rc={'figure.figsize': (20, 5)})
# Create a bar chart
ax = sns.barplot(data=units_sold_product_id, x='Product_ID', y='Amount', palette='viridis')
# Function to format y-axis labels in lakhs
def lakhs_formatter(x, pos):
 return f'{x/1e5:.1f}L'

# Apply the formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(lakhs_formatter))
# Add data labels
for p in ax.patches:
 ax.annotate(f'{p.get_height()/1e5:.1f}L', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', fontsize=10, color='black', xytext=(0, 10),
 textcoords='offset points')

# Set plot labels and title
plt.xlabel('Product ID')
plt.ylabel('Total Units Sold (in Lakhs)')
plt.title('Top 10 Product IDs by Units Sold')
# Show the plot
plt.show()
```



--Married women age group 26-35 yrs from UP, Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothing and Electronics category

Insights:

- The primary consumer base comprises predominantly females, with their purchasing power surpassing that of men.

- The age demographic exhibiting the highest purchasing activity falls within the 26-35 years range, particularly among females.

- Sales thrive significantly in regions such as Uttar Pradesh, Maharashtra, and Karnataka, indicating lucrative markets for Diwali sales.

- Marital status significantly influences buying behavior, with married women exhibiting higher purchasing power.

- Top professions of buyers include those in the IT, Healthcare, and Aviation sectors, indicating potential target segments for marketing initiatives.

- The most popular product segments among buyers encompass Food, Clothing, and Electronics categories, indicating areas of high demand.

Practical Recommendations

- Targeted Marketing: Tailor marketing efforts towards the primary consumer base of married women aged 26-35 years, focusing on regions with high sales potential such as Uttar Pradesh, Maharashtra, and Karnataka.

- Product Assortment: Curate product assortments that align with popular product segments, including Food, Clothing, and Electronics, to meet the preferences of our target demographic.

- Occupational Segmentation: Explore partnerships or promotional strategies targeting professionals in IT, Healthcare, and Aviation sectors to capitalize on their purchasing power.

- Customer Engagement: Implement personalized shopping experiences and loyalty programs to enhance customer engagement and foster long-term relationships.

- Inventory Management: Optimize inventory levels to meet anticipated demand during the festive season, ensuring sufficient stock availability without excess inventory.

- Cross-Selling Opportunities: Identify opportunities for cross-selling by bundling complementary products or offering discounts on related items to increase average order value.

Conclusion

- *Through comprehensive data analysis, we have identified key insights into consumer behavior and sales trends, providing valuable guidance for optimizing business strategies during the Diwali sales season. By leveraging these insights and implementing targeted initiatives, we aim to enhance customer satisfaction, drive sales growth, and establish a competitive edge in the market.*