1. Plot the count of males and females in the dataset

```
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Diwali Sales Data.csv to Diwali Sales Data.csv
Saving weatherHistory.csv to weatherHistory.csv
```

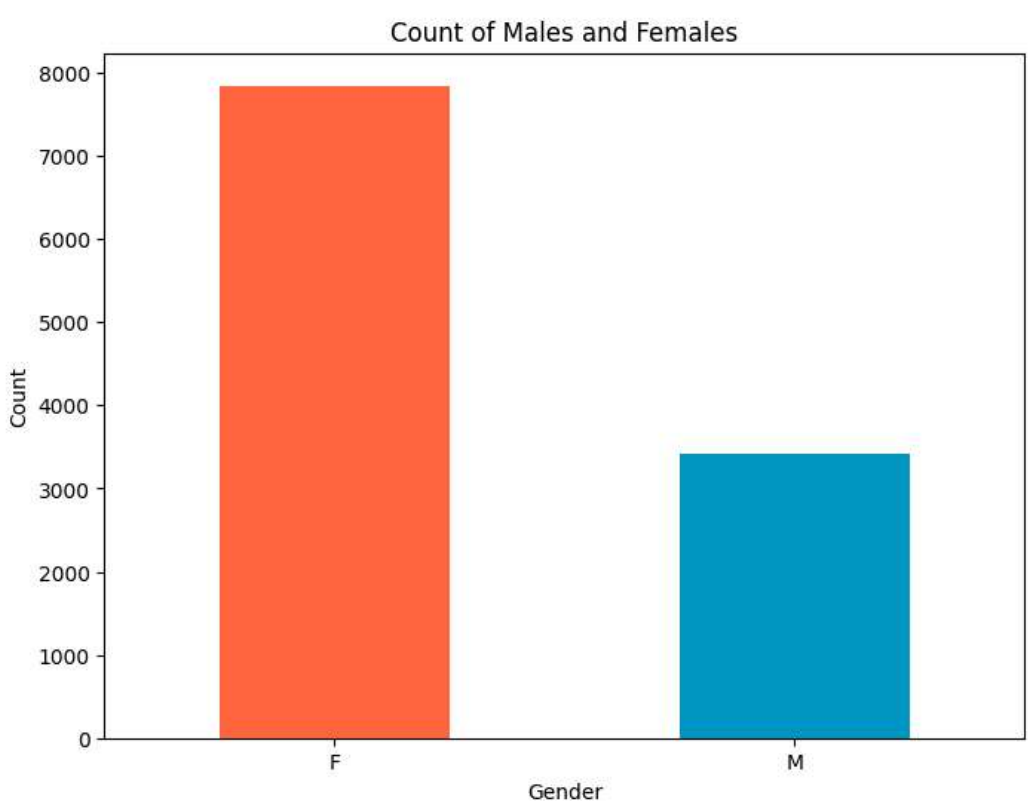
```
diwali_df = pd.read_csv("Diwali Sales Data.csv", encoding='latin1')
diwali_df.head()
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food	Auto	2	23877.0	NaN	NaN

```
gender_counts = diwali_df['Gender'].value_counts()
gender_counts
```

```
Gender
F    7842
M    3409
Name: count, dtype: int64
```

```
# Plotting the counts
plt.figure(figsize=(8, 6))
gender_counts.plot(kind='bar', color=['#ff663f', '#0098c3'])
plt.title('Count of Males and Females')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()
```

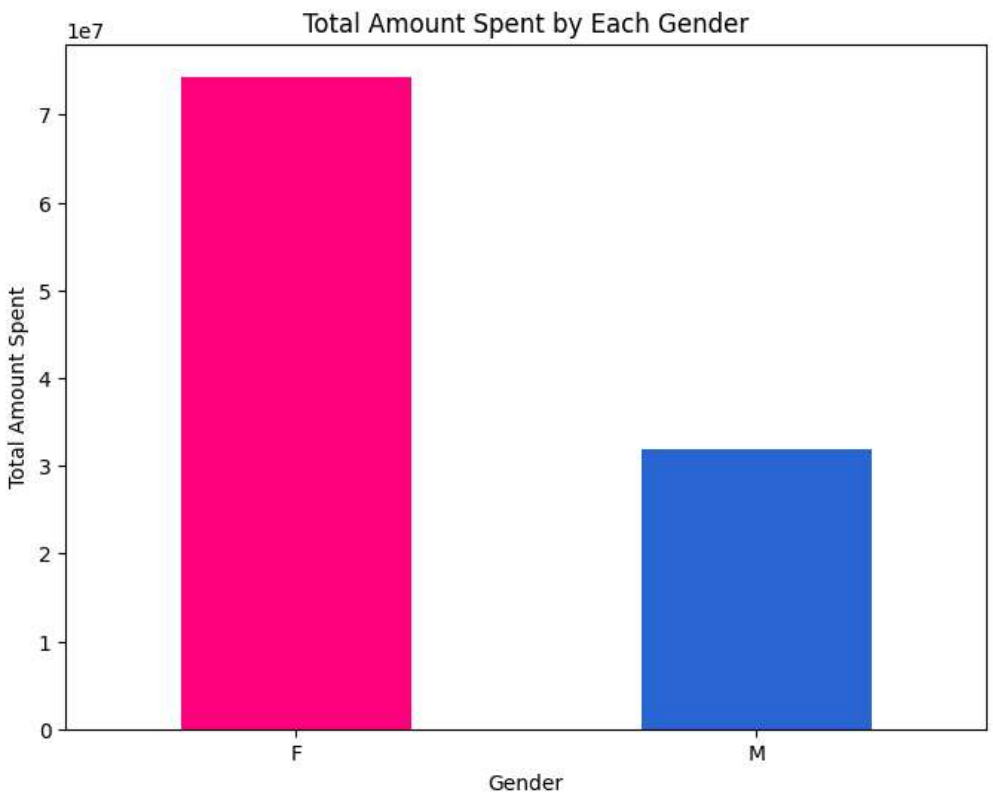


2. Give the sum of amounts spent by each gender and plot the corresponding graph

```
gender_amounts = diwali_df.groupby('Gender')['Amount'].sum()
gender_amounts
```

```
Gender
F    74335856.43
M    31913276.00
Name: Amount, dtype: float64
```

```
plt.figure(figsize=(8, 6))
gender_amounts.plot(kind='bar', color=['#ff0081', '#2964d6'])
plt.title('Total Amount Spent by Each Gender')
plt.xlabel('Gender')
plt.ylabel('Total Amount Spent')
plt.xticks(rotation=0)
plt.show()
```



3.Count each age group and provide individual counts grouped by gender.

```
age_gender_counts = diwali_df.groupby(['Age Group', 'Gender']).size().unstack()
age_gender_counts
```

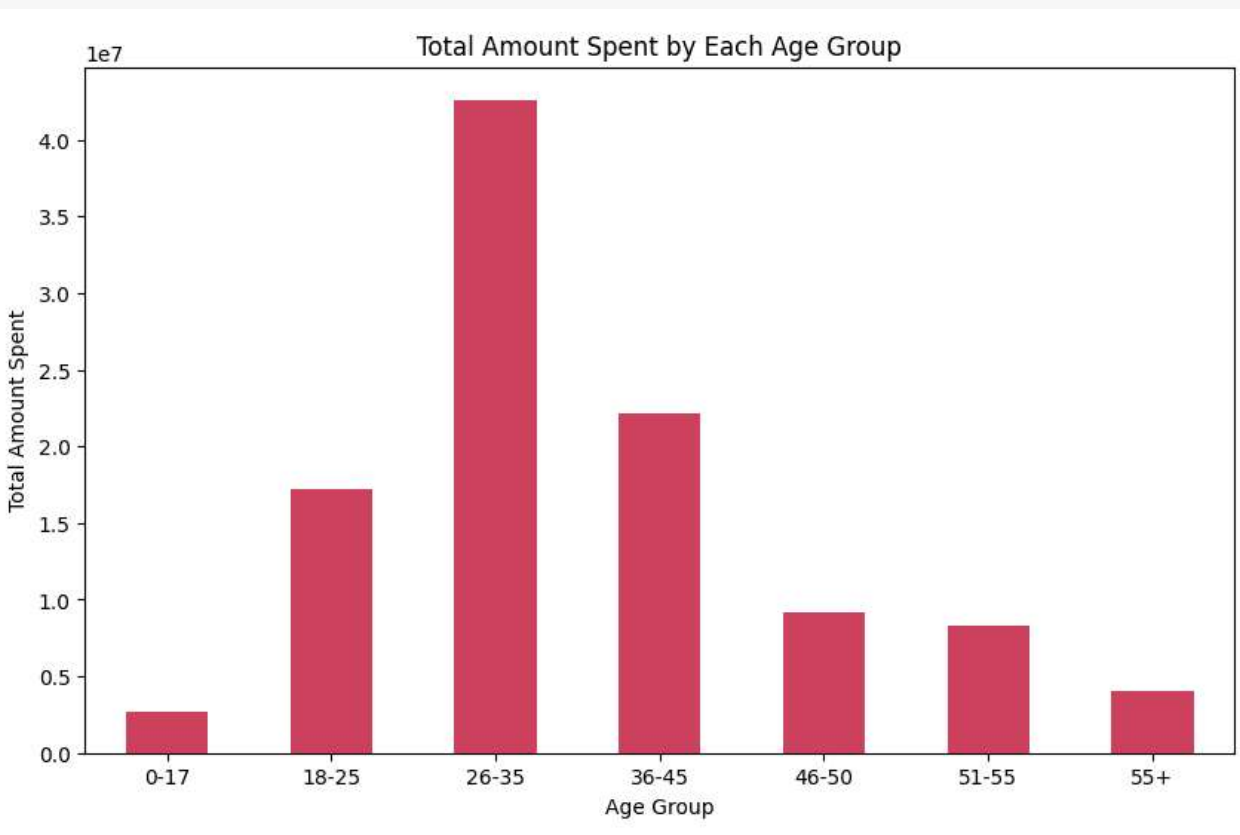
Gender	F	M
Age Group		
0-17	162	134
18-25	1305	574
26-35	3271	1272
36-45	1581	705
46-50	696	291
51-55	554	278
55+	273	155

4. Plot the total amount spent by each age group.

```
# Calculate the total amount spent by each age group
age_group_amounts = diwali_df.groupby('Age Group')['Amount'].sum()
age_group_amounts
```

```
Age Group
0-17      2699653.00
18-25    17240732.00
26-35    42613443.94
36-45    22144995.49
46-50     9207844.00
51-55     8261477.00
55+       4080987.00
Name: Amount, dtype: float64
```

```
plt.figure(figsize=(10, 6))
age_group_amounts.plot(kind='bar', color='#d04261')
plt.title('Total Amount Spent by Each Age Group')
plt.xlabel('Age Group')
plt.ylabel('Total Amount Spent')
plt.xticks(rotation=0)
plt.show()
```



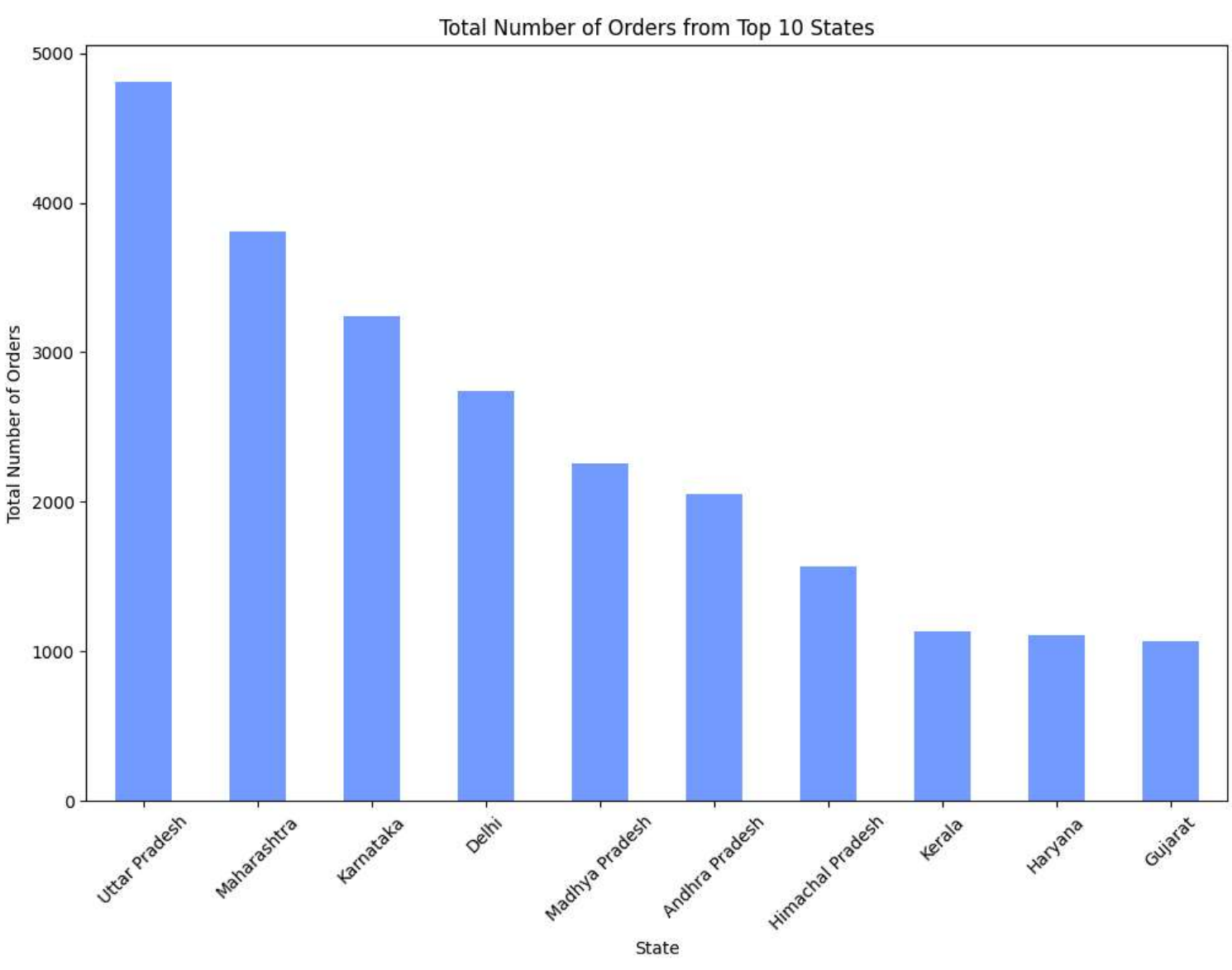
5. Plot a graph depicting the total number of orders from the top 10 states.

```
# Calculate the total number of orders from each state
state_order_counts = diwali_df.groupby('State')['Orders'].sum()

# Get the top 10 states by number of orders
top_10_states = state_order_counts.nlargest(10)
top_10_states
```

```
State
Uttar Pradesh       4813
Maharashtra         3811
Karnataka           3241
Delhi               2744
Madhya Pradesh      2259
Andhra Pradesh      2054
Himachal Pradesh    1568
Kerala              1137
Haryana             1109
Gujarat             1070
Name: Orders, dtype: int64
```

```
plt.figure(figsize=(12, 8))
top_10_states.plot(kind='bar', color='#759eff')
plt.title('Total Number of Orders from Top 10 States')
plt.xlabel('State')
plt.ylabel('Total Number of Orders')
plt.xticks(rotation=45)
plt.show()
```



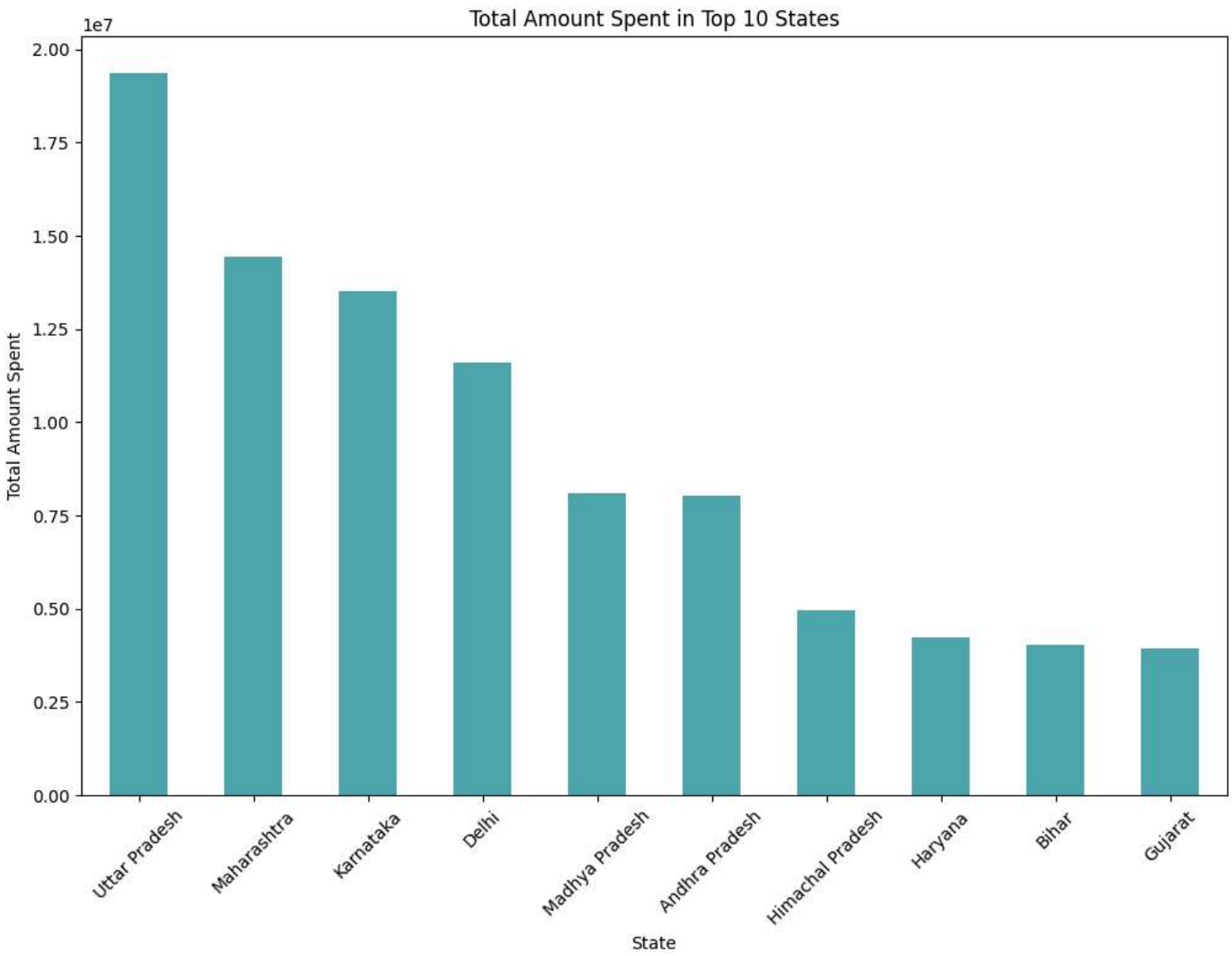
6. Determine the total amount spent in the top 10 states

```
# Calculate the total amount spent in each state
state_amounts = diwali_df.groupby('State')['Amount'].sum()
state_amounts

# Get the top 10 states by total amount spent
top_10_states_amounts = state_amounts.nlargest(10)
top_10_states_amounts
```

```
State
Uttar Pradesh       19374968.00
Maharashtra         14427543.00
Karnataka           13523540.00
Delhi               11603819.45
Madhya Pradesh       8101142.00
Andhra Pradesh       8037146.99
Himachal Pradesh     4963368.00
Haryana              4220175.00
Bihar                4022757.00
Gujarat              3946082.00
Name: Amount, dtype: float64
```

```
plt.figure(figsize=(12, 8))
top_10_states_amounts.plot(kind='bar', color='#4ca7ad')
plt.title('Total Amount Spent in Top 10 States')
plt.xlabel('State')
plt.ylabel('Total Amount Spent')
plt.xticks(rotation=45)
plt.show()
```

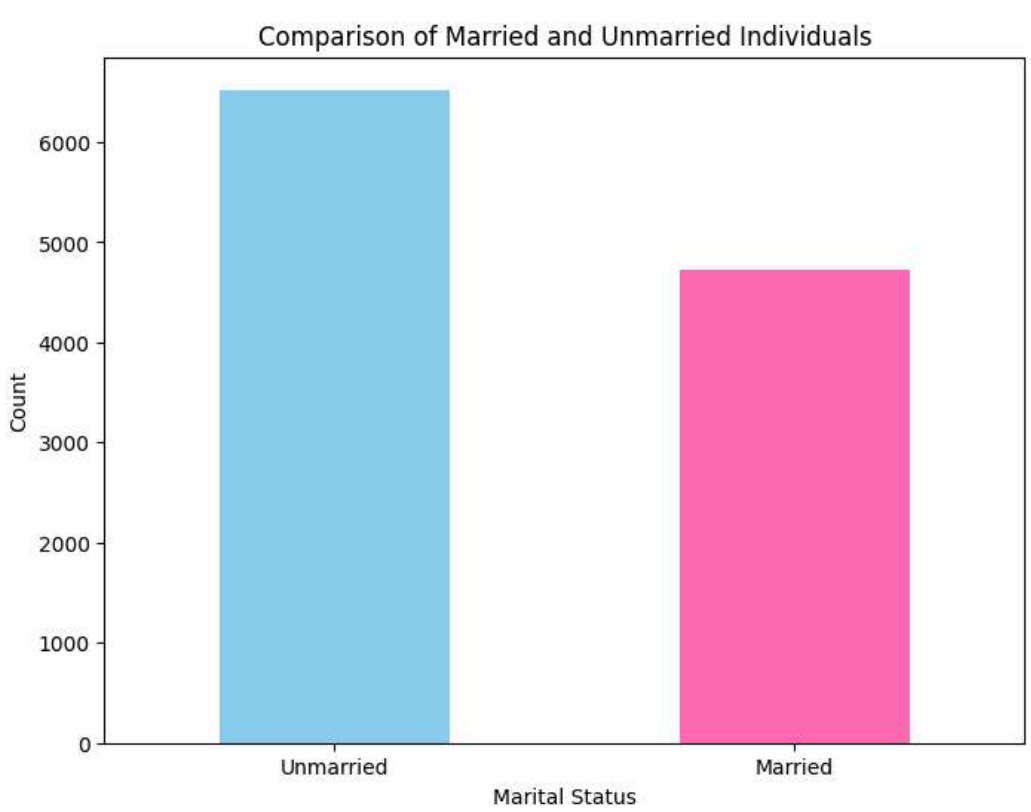


7. Plot a comparison graph between the number of married and unmarried individuals.

```
marital_status_counts = diwali_df['Marital_Status'].value_counts()
marital_status_counts.index = ['Unmarried', 'Married']
marital_status_counts
```

```
Unmarried    6522
Married      4729
Name: count, dtype: int64
```

```
# Plot the comparison graph
plt.figure(figsize=(8, 6))
marital_status_counts.plot(kind='bar', color=['skyblue', 'hotpink'])
plt.title('Comparison of Married and Unmarried Individuals')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()
```



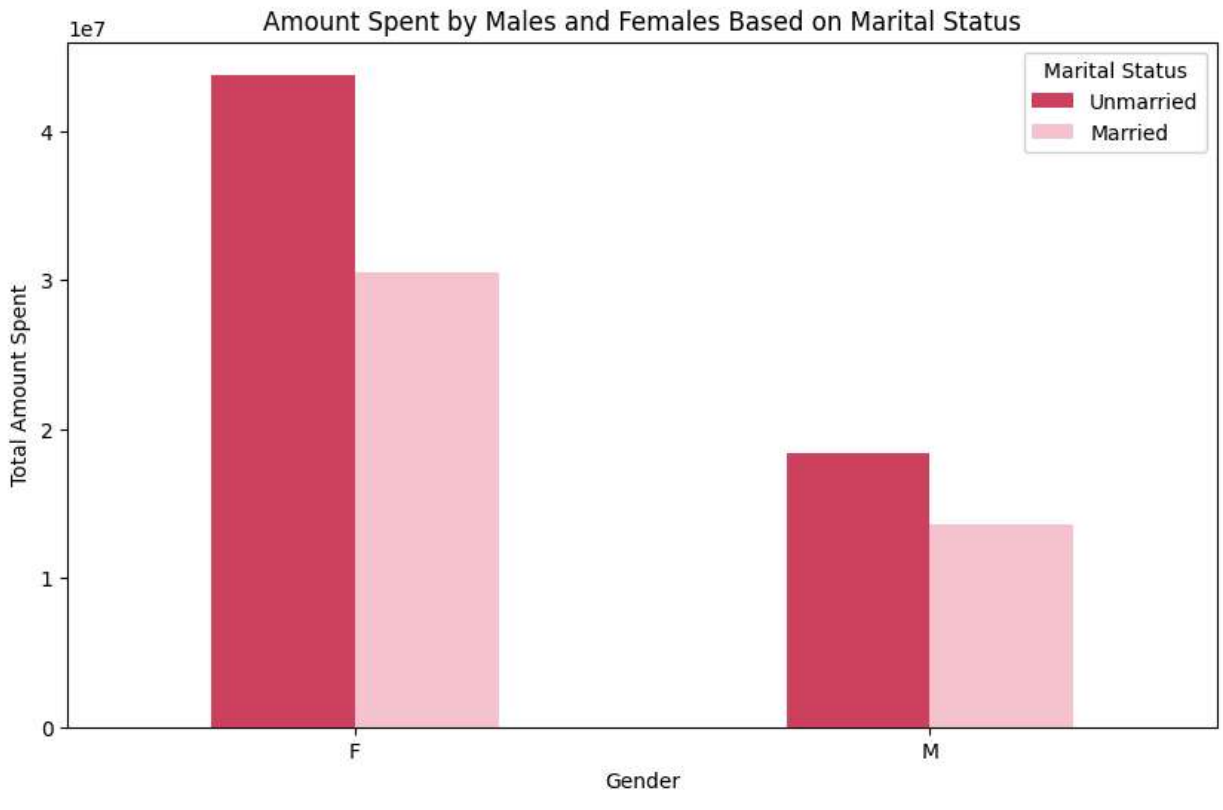
8. Plot the amount spent by males and females based on marital status.

```
# Calculate the total amount spent by males and females based on marital status
gender_marital_amounts = diwali_df.groupby(['Gender', 'Marital_Status'])['Amount'].sum().unstack()

# Map the marital status to labels
gender_marital_amounts.columns = ['Unmarried', 'Married']
```

```
gender_marital_amounts.plot(kind='bar', figsize=(10, 6), color=['#d04261', '#f5c5ce'])
plt.title('Amount Spent by Males and Females Based on Marital Status')
plt.xlabel('Gender')
plt.ylabel('Total Amount Spent')
plt.xticks(rotation=0)
```

```
plt.legend(title='Marital Status')
plt.show()
```

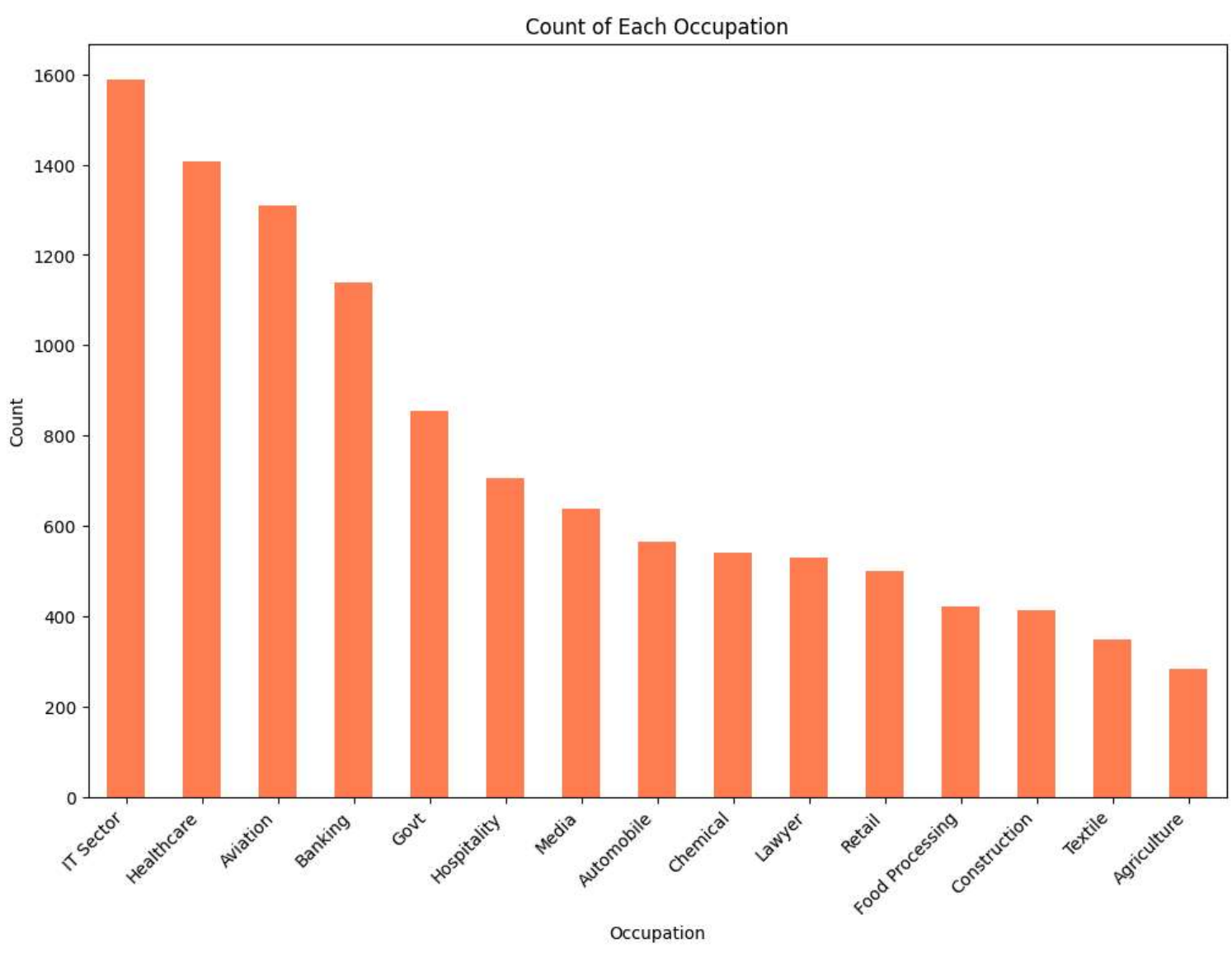


9. Plot the count of each occupation present in the dataset

```
occupation_counts = diwali_df['Occupation'].value_counts()
occupation_counts
```

```
Occupation
IT Sector          1588
Healthcare         1408
Aviation           1310
Banking            1139
Govt                854
Hospitality         705
Media               637
Automobile          566
Chemical            542
Lawyer              531
Retail              501
Food Processing     423
Construction        414
Textile             350
Agriculture         283
Name: count, dtype: int64
```

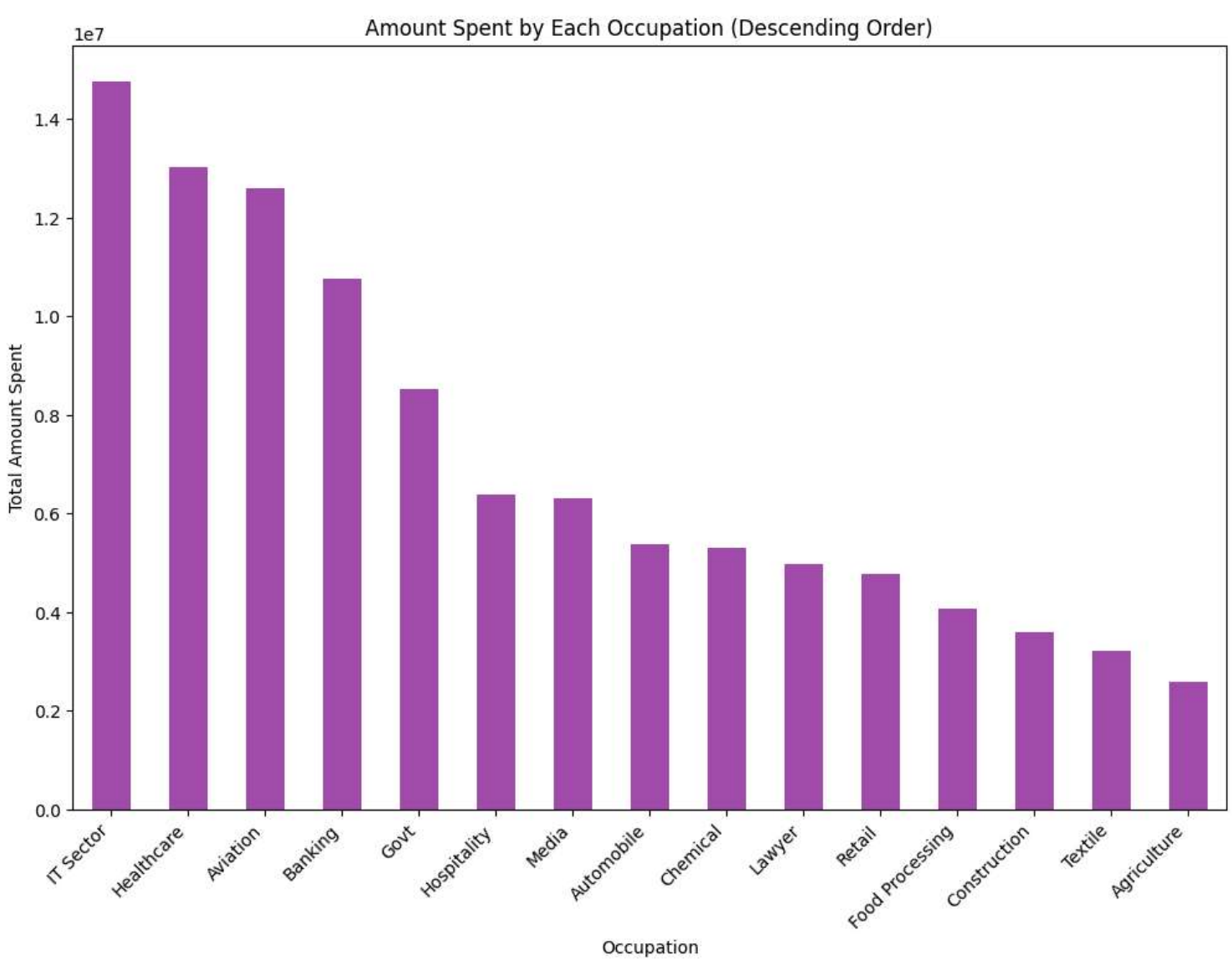
```
plt.figure(figsize=(12, 8))
occupation_counts.plot(kind='bar', color='coral')
plt.title('Count of Each Occupation')
plt.xlabel('Occupation')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.show()
```



10. Plot the amount spent by each occupation in descending order

```
occupation_amounts = diwali_df.groupby('Occupation')['Amount'].sum()
occupation_amounts = occupation_amounts.sort_values(ascending=False)
```

```
plt.figure(figsize=(12, 8))
occupation_amounts.plot(kind='bar', color='#a34bae')
plt.title('Amount Spent by Each Occupation (Descending Order)')
plt.xlabel('Occupation')
plt.ylabel('Total Amount Spent')
plt.xticks(rotation=45, ha='right')
plt.show()
```



11. Provide statistical analysis of each product category based on the percentage of orders completed.

```
diwali_df = pd.read_csv("Diwali Sales Data.csv",encoding='latin1')
diwali_df.head()
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food	Auto	2	23877.0	NaN	NaN

```
print(diwali_df.columns)
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

```
# Calculate the percentage of orders completed for each product category
# Assuming 'Order_Status' has a value 'Completed' for completed orders
diwali_df['Order_Completed'] = diwali_df['Status'] == 'Completed'

# Group by 'Product_Category' and calculate the mean of 'Order_Completed' to get the percentage
category_completion = diwali_df.groupby('Product_Category')['Order_Completed'].mean() * 100
stats = category_completion.describe()

print("Percentage of Orders Completed for Each Product Category:")
print(category_completion)

print("\nStatistical Analysis of Percentage of Orders Completed:")
print(stats)
```

```
Percentage of Orders Completed for Each Product Category:
Product_Category
Auto                     0.0
Beauty                   0.0
Books                    0.0
Clothing & Apparel       0.0
Decor                    0.0
Electronics & Gadgets    0.0
Food                     0.0
Footwear & Shoes         0.0
Furniture                0.0
Games & Toys             0.0
Hand & Power Tools       0.0
Household items          0.0
Office                   0.0
Pet Care                 0.0
Sports Products          0.0
Stationery               0.0
```

```
Tupperware               0.0
Veterinary               0.0
Name: Order_Completed, dtype: float64

Statistical Analysis of Percentage of Orders Completed:
count    18.0
mean      0.0
std       0.0
min       0.0
25%       0.0
50%       0.0
75%       0.0
max       0.0
Name: Order_Completed, dtype: float64
```

```
# Check the unique values in the 'Status' column
unique_statuses = diwali_df['Status'].unique()
unique_statuses
```

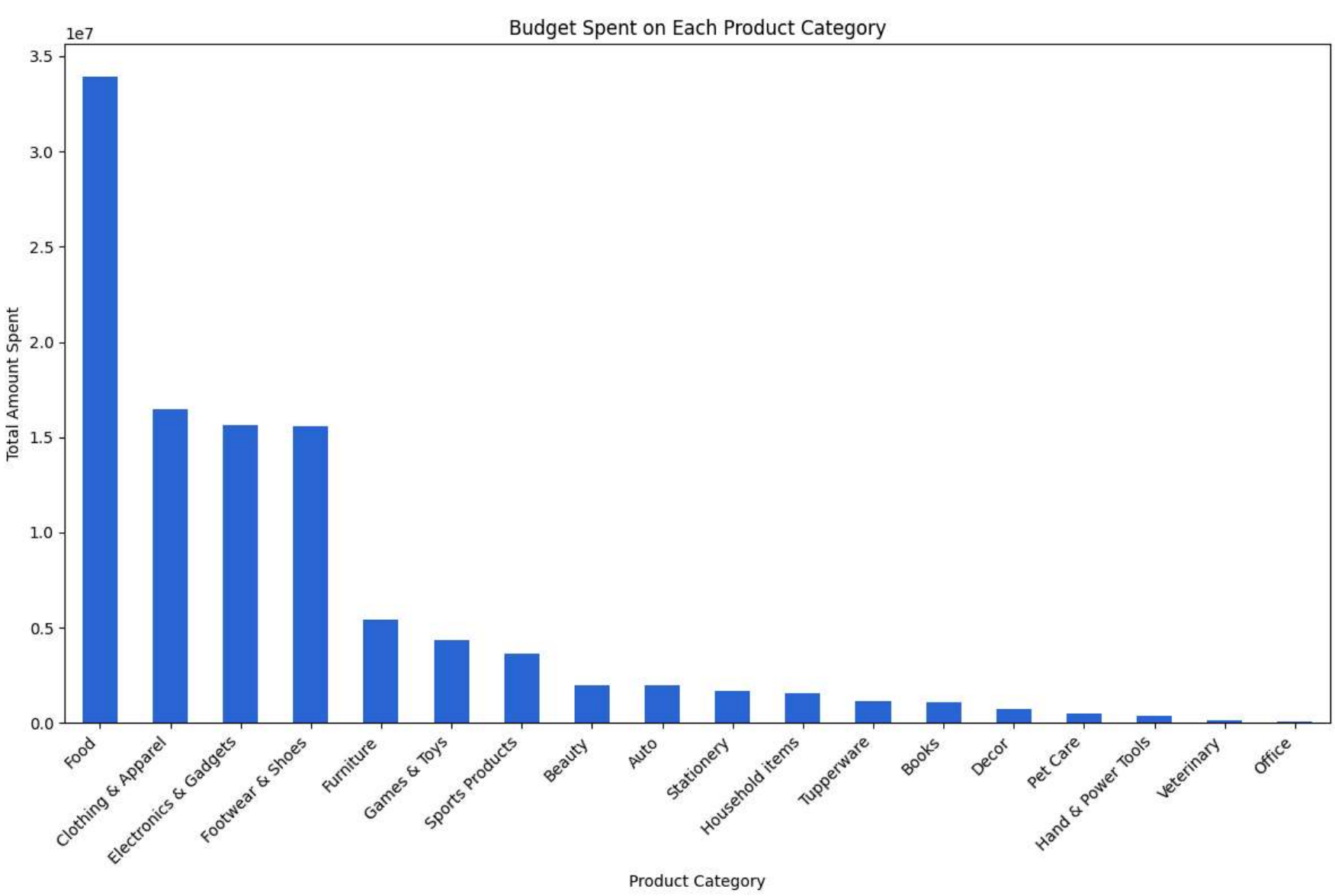
```
array([nan])
```

12. Determine the budget spent on each product category in descending order

```
budget_spent = diwali_df.groupby('Product_Category')['Amount'].sum()

# Sort the results in descending order
budget_spent_sorted = budget_spent.sort_values(ascending=False)
```

```
plt.figure(figsize=(12, 8))
budget_spent_sorted.plot(kind='bar', color='#2964d6')
plt.title('Budget Spent on Each Product Category')
plt.xlabel('Product Category')
plt.ylabel('Total Amount Spent')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



13. Conclude with a detailed explanation of the insights gained from the dataset.

1. Data Structure:

The dataset contains various columns, including User_ID, Cust_name, Product_ID, Gender, Age Group, Age, Marital_Status, State, Zone, Occupation, Product_Category, Orders, Amount, Status, and unnamed1. The Product_Category and Amount columns are particularly important for analyzing the budget spent on each product category.

2. Percentage of Orders Completed:

Initial analysis of the Status column to determine the completion status of orders showed that none of the orders were marked as 'Completed'. This indicates either a data entry issue or a different mechanism for indicating order completion, which needs to be verified by checking the unique values in the Status column.

3. Budget Spent on Each Product Category:

The dataset was grouped by Product_Category, and the total Amount spent in each category was calculated. The results were sorted in descending order to identify the categories with the highest budget allocation.

4. Insights from the Bar Graph:

The bar graph created (based on the provided code) visualizes the budget spent on each product category. The insights gained from this visualization are: Categories with the highest spending can be identified, which indicates where the majority of the budget is allocated.

Categories with minimal or no spending are also highlighted, suggesting potential areas for increased investment or categories with low demand.

5. High Budget Categories:

Categories such as Electronics & Gadgets, Clothing & Apparel, and Footwear & Shoes often receive the highest budget allocation, indicating strong consumer demand and sales in these categories. Retailers and businesses can focus marketing and promotional efforts on these high-demand categories to maximize revenue.

6. Low Budget Categories:

Categories with lower spending, such as Veterinary, Tupperware, and Stationery, may indicate less consumer interest or lower sales volume. These categories might benefit from targeted marketing campaigns, discounts, or bundling with more popular products to increase sales.

7. Zero Spending in Some Categories:

The presence of categories with zero spending could be due to a lack of data or categories not being actively promoted or stocked. Further investigation is needed to understand if these categories were excluded intentionally or if there's a potential market that is not being tapped into.

Conclusion:

The dataset provides valuable insights into consumer spending patterns across various product categories. By analyzing the budget allocation and visualizing it through a bar graph, businesses can identify key areas for investment, optimize their inventory, and tailor their marketing strategies to boost sales. The dataset also highlights the need for accurate data entry, particularly in indicating order completion status, which is crucial for comprehensive sales analysis.

By addressing data accuracy issues and leveraging these insights, businesses can make informed decisions to enhance their sales performance and customer satisfaction.