Diwali Sales Analysis

Importing Libraries

In [92]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\byash\anaconda3\lib\site-packages
(1.26.4)
Note: you may need to restart the kernel to use updated packages.
```

In [93]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\byash\anaconda3\lib\site-packages
(2.2.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\byash\anaconda3\lib\site-pa
ckages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\byash\anaconda3\li
b\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\byash\anaconda3\lib\site-pac
kages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\byash\anaconda3\lib\site-p
ackages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\byash\anaconda3\lib\site-package
s (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [94]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\byash\anaconda3\lib\site-packa
ges (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\byash\anaconda3\lib\site
-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\byash\anaconda3\lib\site-pac
kages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\byash\anaconda3\lib\sit
e-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\byash\anaconda3\lib\sit
e-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.23 in c:\users\byash\anaconda3\lib\site-pack
ages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\byash\anaconda3\lib\site-
packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\byash\anaconda3\lib\site-packag
es (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\byash\anaconda3\lib\site
-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\byash\anaconda3\lib
\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\byash\anaconda3\lib\site-package
s (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [95]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\byash\anaconda3\lib\site-packages
(0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\byash\anaconda3\lib
\site-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in c:\users\byash\anaconda3\lib\site-pack
ages (from seaborn) (2.2.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\byash\anaconda3\l
ib\site-packages (from seaborn) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\byash\anaconda3\lib\site
-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\byash\anaconda3\lib\site-pac
kages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\byash\anaconda3\lib\sit
e-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\byash\anaconda3\lib\sit
e-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\byash\anaconda3\lib\site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\byash\anaconda3\lib\site-packag
es (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\byash\anaconda3\lib\site
-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\byash\anaconda3\lib
\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\byash\anaconda3\lib\site-pac
kages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\byash\anaconda3\lib\site-p
ackages (from pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\byash\anaconda3\lib\site-package
s (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [96]:
```
pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\byash\anaconda3\lib\site-package
s (3.1.5)
Requirement already satisfied: et-xmlfile in c:\users\byash\anaconda3\lib\site-packa
ges (from openpyxl) (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [97]:
```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\byash\anaconda3\lib\site-pac
kages (1.5.1)
Requirement already satisfied: numpy>=1.19.5 in c:\users\byash\anaconda3\lib\site-pa
ckages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\byash\anaconda3\lib\site-pac
kages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\byash\anaconda3\lib\site-pa
ckages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\byash\anaconda3\lib
\site-packages (from scikit-learn) (3.5.0)
Note: you may need to restart the kernel to use updated packages.
```

In [98]:
```
pip install squarify
```

```
Requirement already satisfied: squarify in c:\users\byash\anaconda3\lib\site-package
s (0.4.4)
Note: you may need to restart the kernel to use updated packages.
```

In [99]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
```

Reading Data

In [100…
```
df_sales=pd.read_excel("C:\\Users\\byash\\Diwali Sales\\Python_Diwali_Sales_Analysi
```

In [101…
```
df_sales
```

Out[101…

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	S
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharas
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prac
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prac
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karna
4	1000588	Joni	P00057942	M	26-35	28	1	Gu
…	…	…	…	…	…	…	…	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharas
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Hary
11248	1001209	Oshin	P00201342	F	36-45	40	0	Mac Prac
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karna
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharas

11251 rows × 15 columns

In [102…
```
df_sales.columns
```

Out[102…
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

In [103…
```
df_sales.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 [104…
```
df_sales.describe()
```

Out[104…

	User_ID	Age	Marital_Status	Orders	Amount	Status	uni
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000	0.0	
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858	NaN	
std	1.716125e+03	12.754122	0.493632	1.115047	5222.355869	NaN	
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000	NaN	
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.000000	NaN	
50%	1.003065e+06	33.000000	0.000000	2.000000	8109.000000	NaN	
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000	NaN	
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000	NaN	

In [105…
```
df_sales.isnull().sum()
```

Out[105…
```
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
Status             11251
unnamed1           11251
dtype: int64
```

Removing Null Values

In [106…
```
df_sales=df_sales.drop(columns=["Status","unnamed1"])
```

In [107…
```
df_sales
```

Out[107…

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	S
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharas
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prac
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prac
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karna
4	1000588	Joni	P00057942	M	26-35	28	1	Gu
…	…	…	…	…	…	…	…	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharas
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Hary
11248	1001209	Oshin	P00201342	F	36-45	40	0	Mac Prac
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karna
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharas

11251 rows × 13 columns

Imputation (Median Impution)

The median imputation method would be suitable especially for the 'Amount' column. This is because the median is less sensitive to outliers and extreme values,The 'Amount' column might contain unusually high or low sales values, which could distort the mean. Using the median ensures that the imputed values are more representative of the typical sales amounts.

In [108…
```
df_sales['Amount'].median()
```

Out[108… 8109.0

Key insights and visuliaztion

Total Amount

In [109…
```
total_Amount=df_sales['Amount'].sum()
print("Total Amount:",total_Amount)
```

```
Total Amount: 106249132.43
```

Total Orders

In [110…
```
total_Orders=df_sales['Orders'].sum()
print("Total Orders:",total_Orders)
```

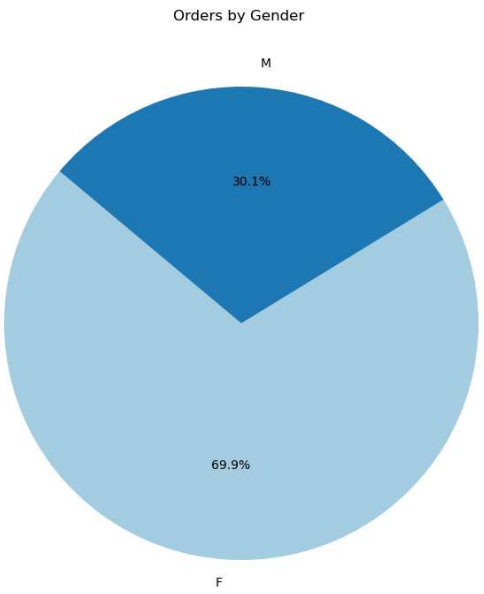
```
Total Orders: 28007
```

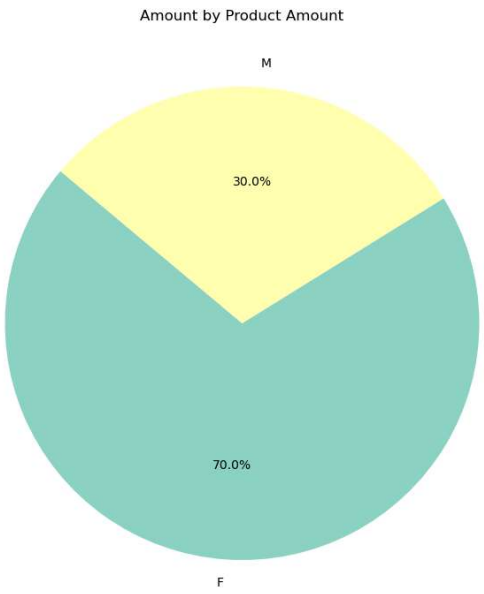
Gender vs. Order And Amount

In [111…
```
fig, axes = plt.subplots(1, 2, figsize=(14, 7))
# Pie Chart for Orders
axes[0].pie(
   orders_by_Gender,
    labels=orders_by_Gender.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=plt.cm.Paired.colors
)
axes[0].set_title('Orders by Gender ')

# Pie Chart for Amount
axes[1].pie(
    amount_by_Gender,
    labels=amount_by_Gender.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=plt.cm.Set3.colors
)
axes[1].set_title('Amount by Product Amount')

# Adjust layout
plt.tight_layout()
plt.show()
```





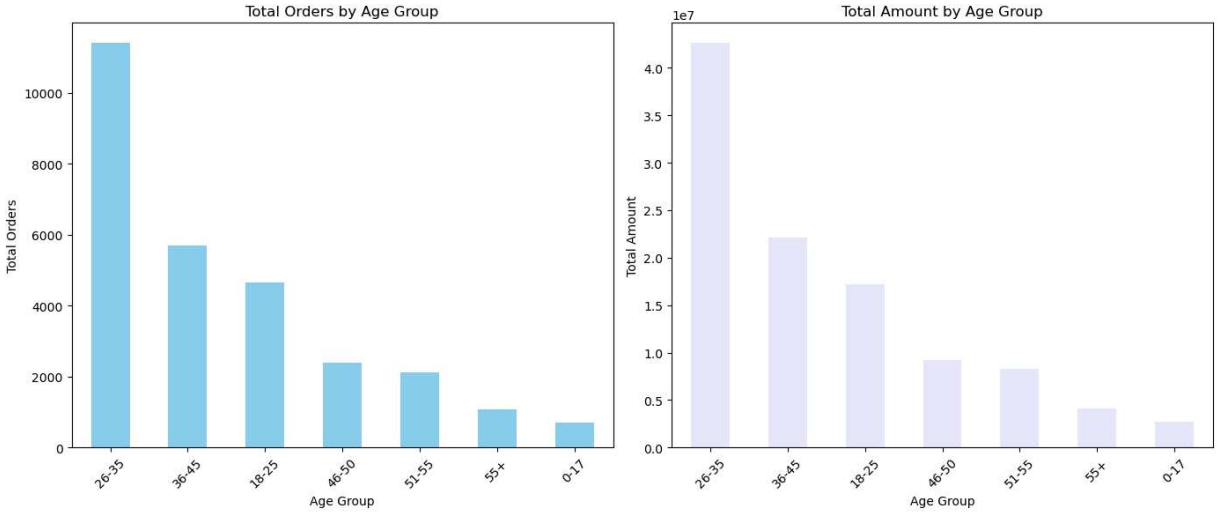
In [112…
```
orders_by_Gender = df_sales.groupby('Gender')['Orders'].sum()
amount_by_Gender = df_sales.groupby('Gender')['Amount'].sum()
```

Age Group Vs Orders and Amount

In [113…
```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 6))
# Chart 1: Total Orders by Product Category
df_sales.groupby('Age Group')['Orders'].sum().sort_values(ascending=False).plot(
    kind='bar',
    ax=axes[0],
    color='skyblue',
    title='Total Orders by Age Group'
)
axes[0].set_xlabel('Age Group')
axes[0].set_ylabel('Total Orders')
axes[0].tick_params(axis='x', rotation=45)

# Chart 2: Total Amount by Product Category
df_sales.groupby('Age Group')['Amount'].sum().sort_values(ascending=False).plot(
    kind='bar',
    ax=axes[1],
    color='Lavender',
    title='Total Amount by Age Group'
)
axes[1].set_xlabel('Age Group')
axes[1].set_ylabel('Total Amount')
axes[1].tick_params(axis='x', rotation=45)

# Adjust layout
plt.tight_layout()
plt.show()
```



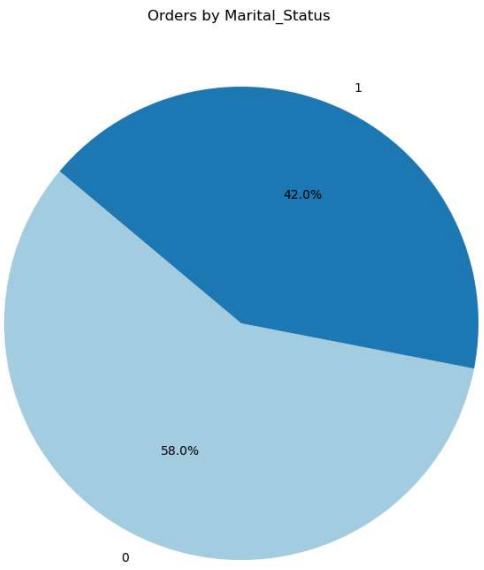
Marital_Status Vs Orders & Amount

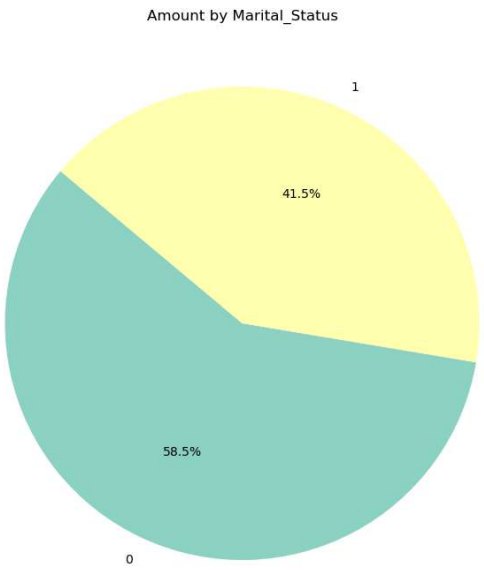
In [114…
```
orders_by_Marital_Status= df_sales.groupby('Marital_Status')['Orders'].sum()
amount_by_Marital_Status = df_sales.groupby('Marital_Status')['Amount'].sum()
```

In [115…
```
fig, axes = plt.subplots(1, 2, figsize=(14, 7))
# Pie Chart for Orders,#0-Unmarried,1-married
axes[0].pie(
   orders_by_Marital_Status,
    labels=orders_by_Marital_Status.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=plt.cm.Paired.colors
)
axes[0].set_title('Orders by Marital_Status ')

# Pie Chart for Amount
axes[1].pie(
    amount_by_Marital_Status,
    labels=amount_by_Marital_Status.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=plt.cm.Set3.colors
)
axes[1].set_title('Amount by Marital_Status')

# Adjust layout
plt.tight_layout()
plt.show()
```





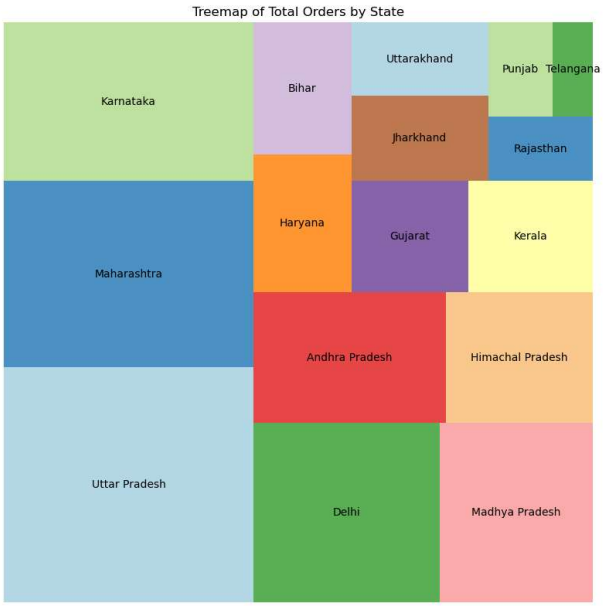
State Vs Orders and Amounts

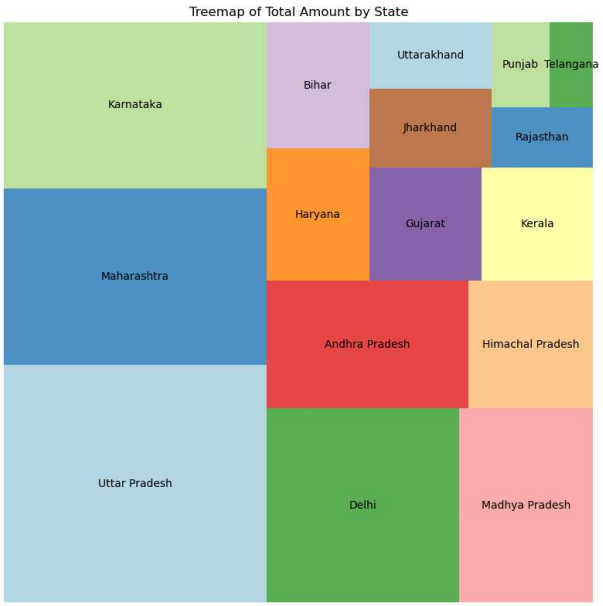
In [116…

```
state_group = state_group.sort_values(by='Amount', ascending=False)

fig, axes = plt.subplots(1, 2, figsize=(16, 8))
# Treemap for Orders
axes[0].set_title('Treemap of Total Orders by State')
squarify.plot(sizes=state_group['Orders'],
              label=state_group['State'],
              color=plt.cm.Paired.colors,
              alpha=0.8,
              ax=axes[0])
axes[0].axis('off')
# Treemap for Amount
axes[1].set_title('Treemap of Total Amount by State')
squarify.plot(sizes=state_group['Amount'],
              label=state_group['State'],
              color=plt.cm.Paired.colors,
              alpha=0.8,
              ax=axes[1])
axes[1].axis('off')

plt.tight_layout()
plt.show()
```





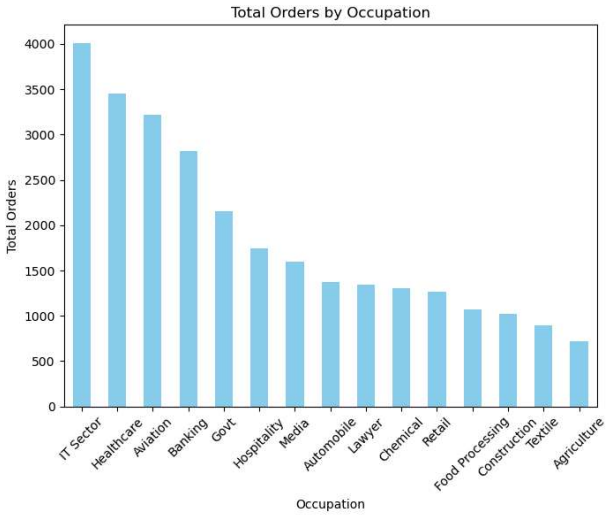
Occupation Vs Order & Amount

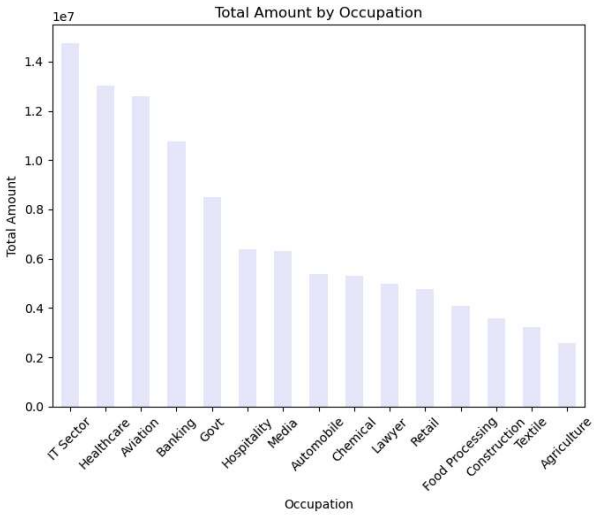
In [117…

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 6))
# Chart 1: Total Orders by Occupation
df_sales.groupby('Occupation')['Orders'].sum().sort_values(ascending=False).plot(
    kind='bar',
    ax=axes[0],
    color='skyblue',
    title='Total Orders by Occupation'
)
axes[0].set_xlabel('Occupation')
axes[0].set_ylabel('Total Orders')
axes[0].tick_params(axis='x', rotation=45)

# Chart 2: Total Amount by Occupation
df_sales.groupby('Occupation')['Amount'].sum().sort_values(ascending=False).plot(
    kind='bar',
    ax=axes[1],
    color='Lavender',
    title='Total Amount by Occupation'
)
axes[1].set_xlabel('Occupation')
axes[1].set_ylabel('Total Amount')
axes[1].tick_params(axis='x', rotation=45)

# Adjust layout
plt.tight_layout()
plt.show()
```





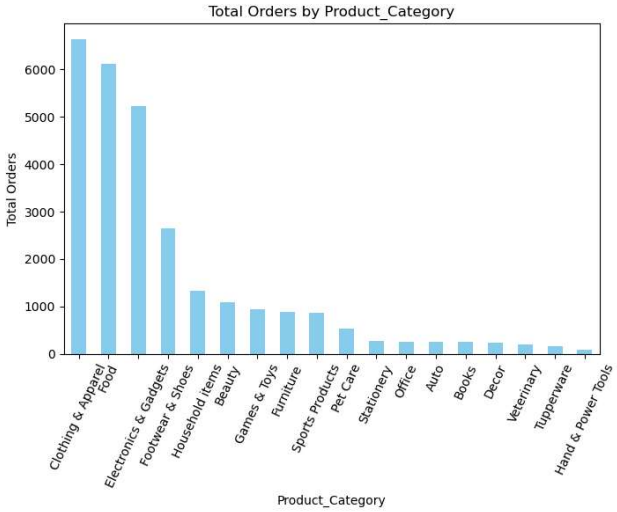
Product category vs Orders And Amount

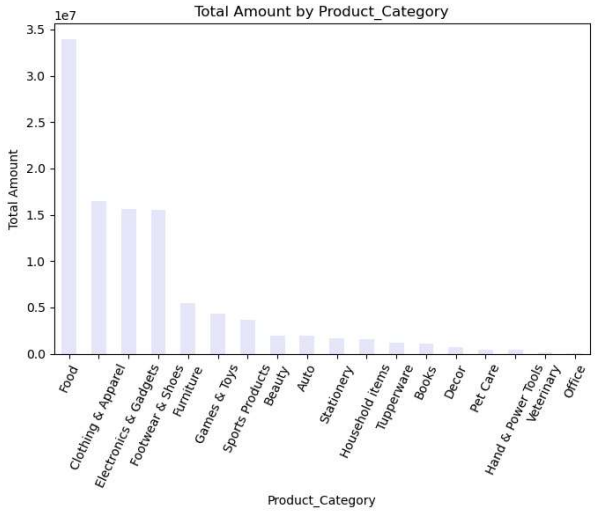
In [118…

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 6))
# Chart 1: Total Orders by Product category
df_sales.groupby('Product_Category')['Orders'].sum().sort_values(ascending=False).p
    kind='bar',
    ax=axes[0],
    color='skyblue',
    title='Total Orders by Product_Category'
)
axes[0].set_xlabel('Product_Category')
axes[0].set_ylabel('Total Orders')
axes[0].tick_params(axis='x', rotation=65)

# Chart 2: Total Amount by Product category
df_sales.groupby('Product_Category')['Amount'].sum().sort_values(ascending=False).p
    kind='bar',
    ax=axes[1],
    color='Lavender',
    title='Total Amount by Product_Category'
)
axes[1].set_xlabel('Product_Category')
axes[1].set_ylabel('Total Amount')
axes[1].tick_params(axis='x', rotation=65)

# Adjust layout
plt.tight_layout()
plt.show()
```





Order Vs Amount

In [119…

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Orders', y='Amount', data=df_sales, color='Skyblue', alpha=0.6)

plt.title('Orders vs Amount')
plt.xlabel('Orders')
plt.ylabel('Amount')

plt.tight_layout()
plt.show()
```