```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt #visualizing data


df=pd.read_csv('Diwali Sales Data.csv',encoding = 'unicode_escape')
#to avoid encoding error,use unicode_escape
#Using 'unicode_escape' tells pandas to treat those escape sequences
as actual characters, not control codes that would otherwise break
reading.


df.shape
```

```
(11251, 15)
```

```python
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
```

```python
df.describe()
```

```
            User_ID           Age  Marital_Status        Orders
Amount  \
count  1.125100e+04  11251.000000    11251.000000  11251.000000
11239.000000
mean   1.003004e+06     35.421207        0.420318      2.489290
9453.610858
std    1.716125e+03     12.754122        0.493632      1.115047
```

```
5222.355869
min    1.000001e+06     12.000000         0.000000       1.000000
188.000000
25%    1.001492e+06     27.000000         0.000000       1.500000
5443.000000
50%    1.003065e+06     33.000000         0.000000       2.000000
8109.000000
75%    1.004430e+06     43.000000         1.000000       3.000000
12675.000000
max    1.006040e+06     92.000000         1.000000       4.000000
23952.000000

       Status  unnamed1
count     0.0       0.0
mean      NaN       NaN
std       NaN       NaN
min       NaN       NaN
25%       NaN       NaN
50%       NaN       NaN
75%       NaN       NaN
max       NaN       NaN
```

```
df.head()
```

```
   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


            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


    Amount  Status  unnamed1
```

```
0  23952.0     NaN       NaN
1  23934.0     NaN       NaN
2  23924.0     NaN       NaN
3  23912.0     NaN       NaN
4  23877.0     NaN       NaN
```

```
df.drop([' Status', 'unnamed1'], axis=1, inplace=True)
#drop/remove irrelevant columns #inplace(true) used to save
```

```
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
```

```
df.head()
```

```
   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


            State      Zone       Occupation Product_Category  Orders
Amount
0     Maharashtra   Western       Healthcare             Auto       1
23952.0
```

```
1  Andhra Pradesh  Southern             Govt             Auto       3
23934.0
2   Uttar Pradesh   Central       Automobile             Auto       3
23924.0
3       Karnataka  Southern     Construction             Auto       2
23912.0
4         Gujarat   Western  Food Processing             Auto       2
23877.0
```

```
df.dropna(inplace = True)
df.shape
```

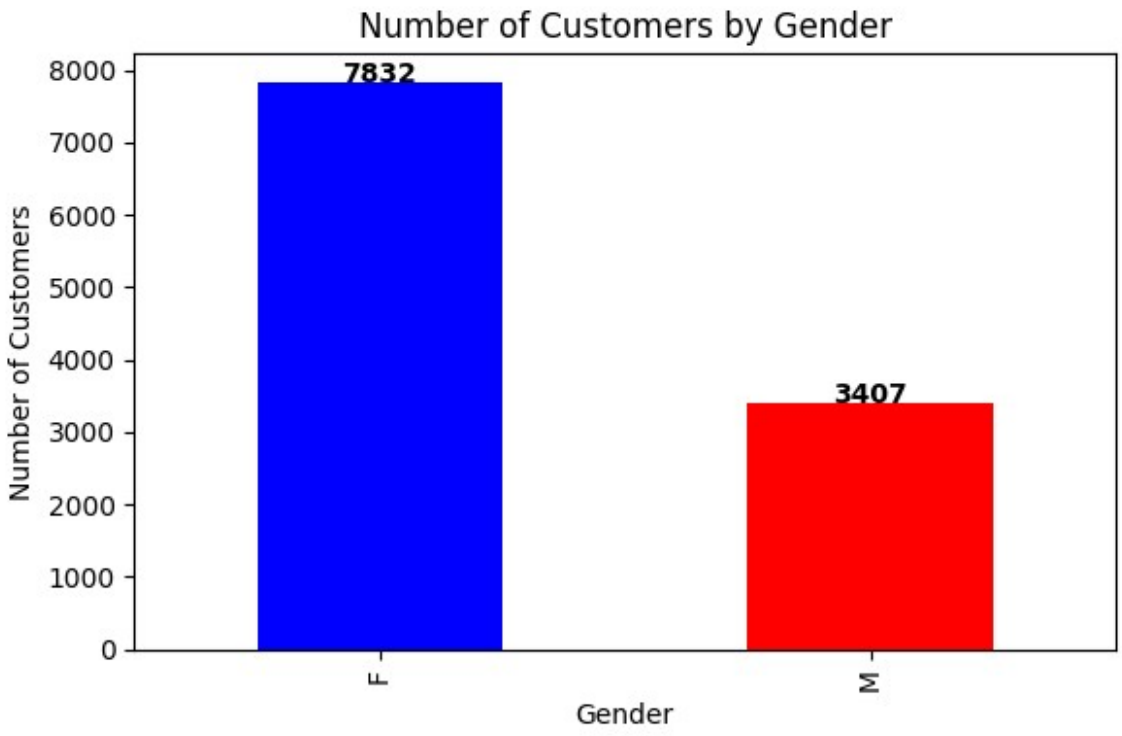
```
(11239, 13)
```

```
df.columns
```

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

BY GENDER

```
import matplotlib.pyplot as plt

# Count values by gender
gender_counts = df['Gender'].value_counts()

# Plot bar chart
plt.figure(figsize=(6,4))
gender_counts.plot(kind='bar', color=['blue', 'red'])

# Add labels and title
plt.xlabel('Gender')
plt.ylabel('Number of Customers')
plt.title('Number of Customers by Gender')

# Show values on bars
for i, val in enumerate(gender_counts):
    plt.text(i, val + 5, str(val), ha='center', fontweight='bold')

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



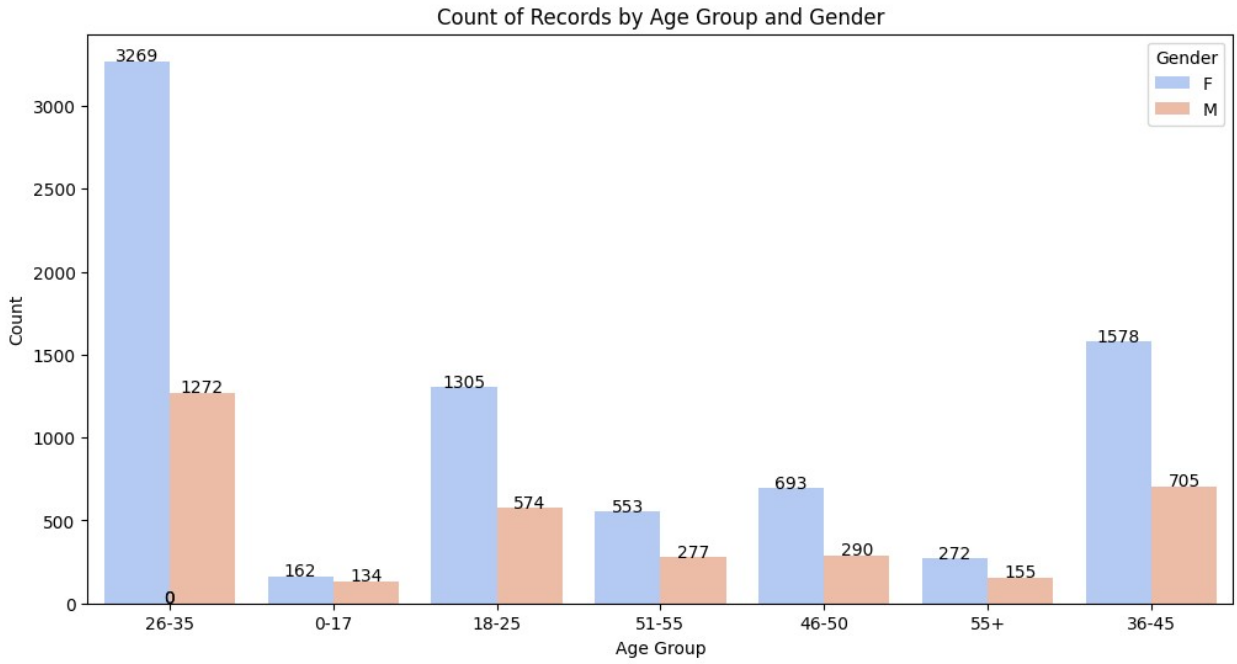

```
df.groupby(['Gender'],as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending = False)
```

| | Gender | Amount |
|---|---|---|
| 0 | F | 74335856.43 |
| 1 | M | 31913276.00 |

BY AGE-GROUP

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

plt.figure(figsize=(12,6))

# Create countplot
ax = sns.countplot(data=df, x='Age Group',
hue='Gender',palette='coolwarm')

plt.title('Count of Records by Age Group and Gender')
plt.xlabel('Age Group')
plt.ylabel('Count')

# Add counts on top of bars
for p in ax.patches:
    height = p.get_height()
    ax.text(
        p.get_x() + p.get_width() / 2.,  # X position: center of the
```

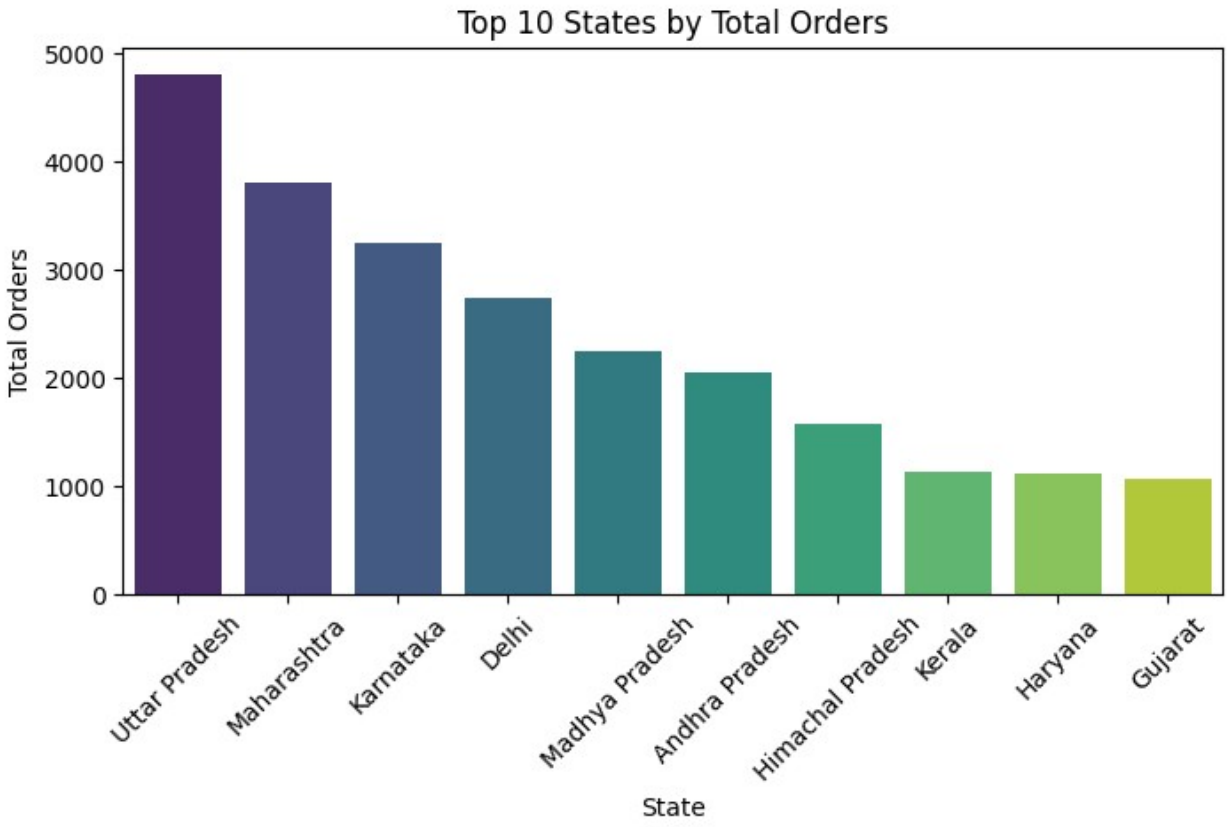
```
bar
        height + 0.5,                    # Y position: just above the
bar
        int(height),                    # Text to display: the height
(count)
        ha="center"                    # Horizontal alignment: center
    )

plt.show()
```




BY STATE

```
df.groupby(['State'],as_index=False)
['Orders'].sum().sort_values(by='Orders',ascending = False).head(10)
import seaborn as sns
import matplotlib.pyplot as plt

# Aggregate total Orders by State and get top 10
top_states = df.groupby('State', as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

plt.figure(figsize=(8,4))

# Colorful barplot for top 10 states by Orders
sns.barplot(data=top_states, x='State', y='Orders', palette='viridis')

plt.title('Top 10 States by Total Orders')
plt.xlabel('State')
plt.ylabel('Total Orders')
```

```
plt.xticks(rotation=45)  # Rotate state names for readability
plt.show()


C:\Users\SBI\AppData\Local\Temp\ipykernel_16552\2352320976.py:11:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=top_states, x='State', y='Orders',
palette='viridis')
```

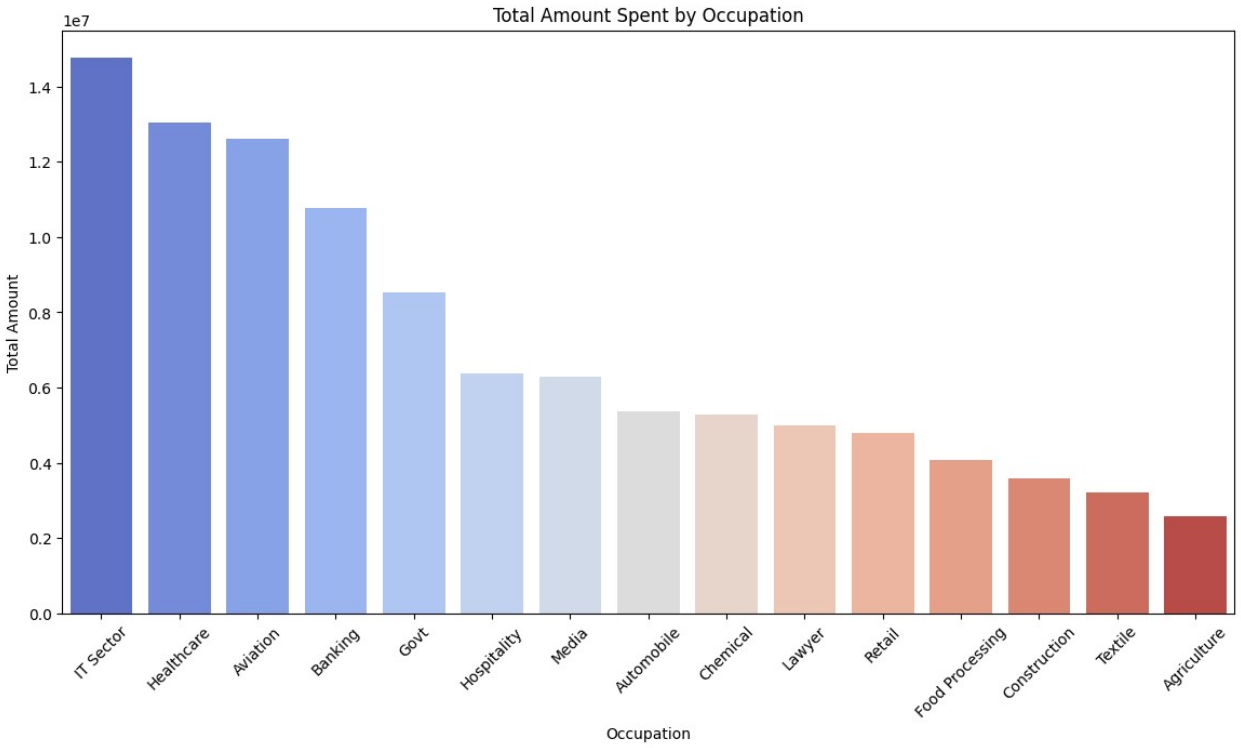



OCCUPATION

```
df.groupby(['Occupation'],as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending = False)
import seaborn as sns
import matplotlib.pyplot as plt

# Aggregate total Amount by Occupation and sort descending
occupation_amount = df.groupby('Occupation', as_index=False)
```

```
['Amount'].sum().sort_values(by='Amount', ascending=False)

plt.figure(figsize=(14,7))

# Barplot for total Amount by Occupation with colorful palette
sns.barplot(x='Occupation', y='Amount', data=occupation_amount,
palette='coolwarm')

plt.title('Total Amount Spent by Occupation')
plt.xlabel('Occupation')
plt.ylabel('Total Amount')

plt.xticks(rotation=45)  # Rotate labels for readability
plt.show()
```

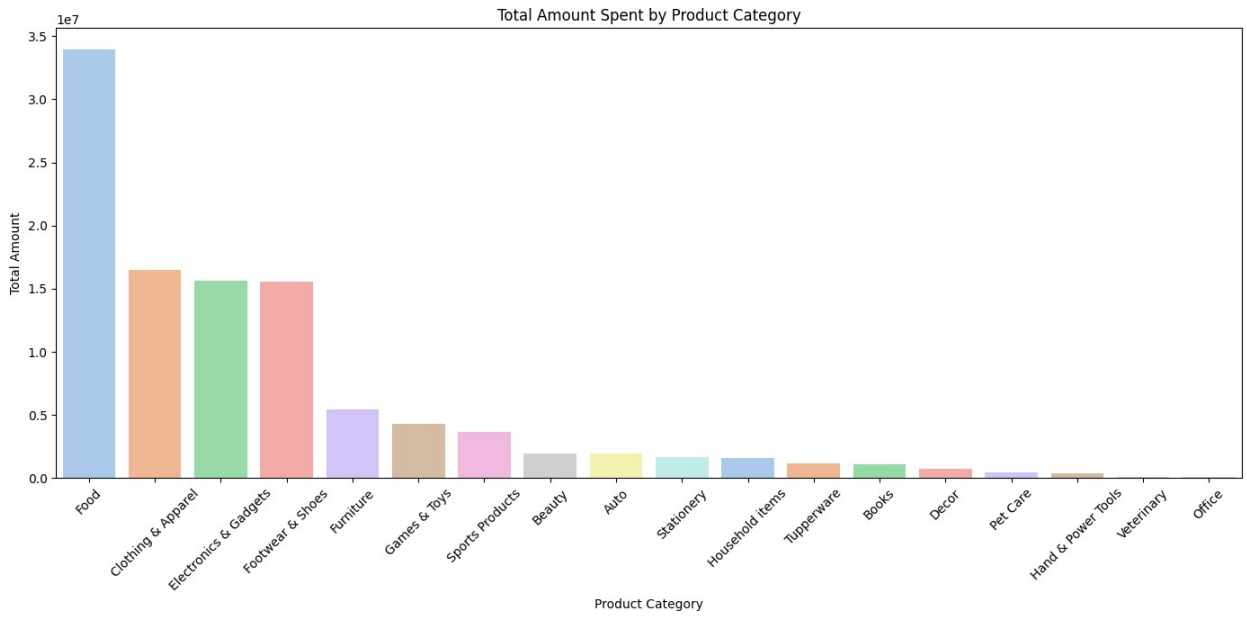
```
C:\Users\SBI\AppData\Local\Temp\ipykernel_16552\3867451347.py:11:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='Occupation', y='Amount', data=occupation_amount,
palette='coolwarm')
```




PRODUCT CATEGORY

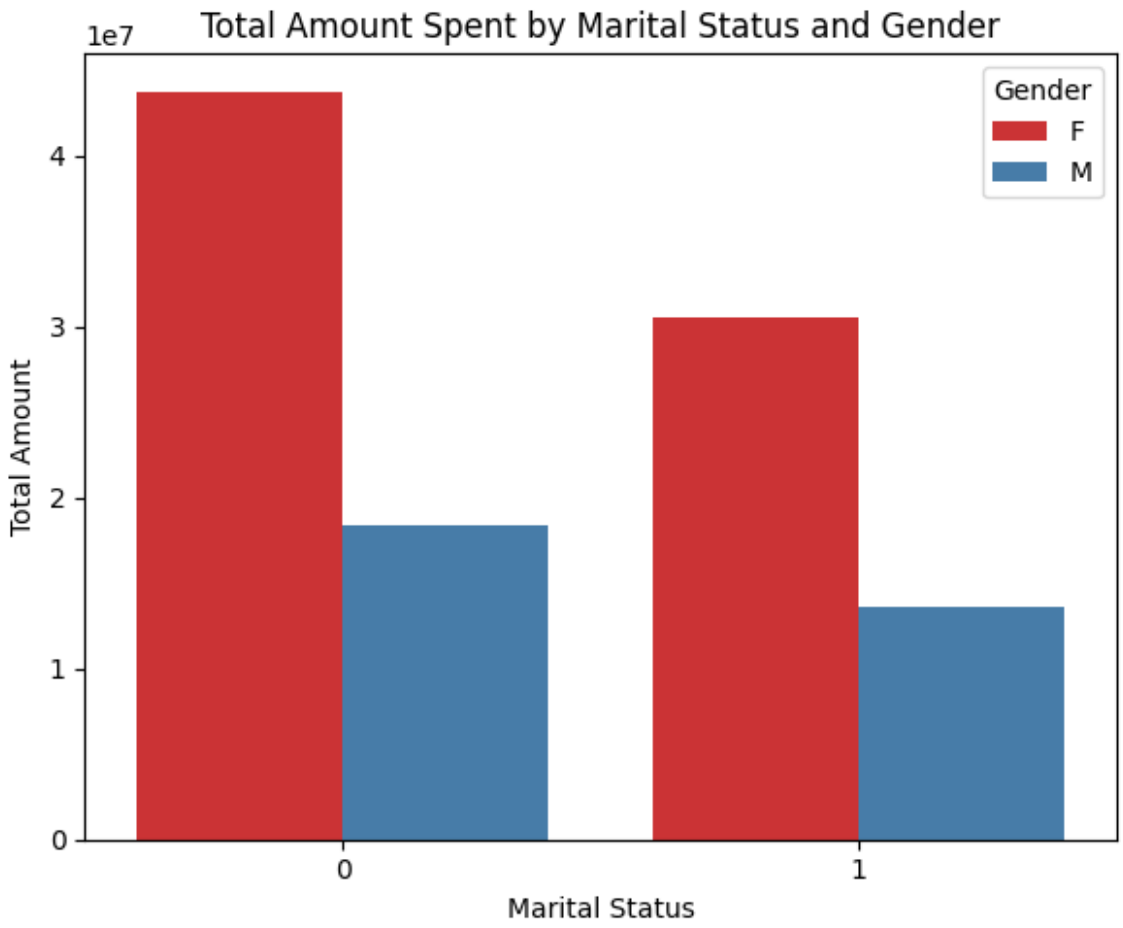
```
df.groupby(['Product_Category'],as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending = False)
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Group and sort data
category_amount = df.groupby(['Product_Category'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)

# Step 2: Create the plot
plt.figure(figsize=(14,7))
sns.barplot(data=category_amount, x='Product_Category', y='Amount',
palette='pastel')

# Step 3: Add labels and title
plt.title('Total Amount Spent by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Total Amount')

# Step 4: Rotate x-axis labels for clarity
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\SBI\AppData\Local\Temp\ipykernel_16552\622134834.py:10:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=category_amount, x='Product_Category', y='Amount',
palette='pastel')
```




MARITAL STATUS

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

# Step 1: Group data by Marital_Status and Gender, then sum Amount
marital_gender_amount = df.groupby(['Marital_Status', 'Gender'],
as_index=False)['Amount'].sum()

# Step 2: Plot with seaborn
plt.figure(figsize=(6,5))
sns.barplot(data=marital_gender_amount, x='Marital_Status',
y='Amount', hue='Gender', palette='Set1')

# Step 3: Add labels and title
plt.title('Total Amount Spent by Marital Status and Gender')
plt.xlabel('Marital Status')
plt.ylabel('Total Amount')

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


Total Amount Spent by Marital Status and Gender
1e7
Gender
F
M
Total Amount
0
1
2
3
4
0
1
Marital Status