Load and Inspect the Data:

In [1]:
```python
import pandas as pd

# Load the dataset
data = pd.read_csv("retail_sales.csv")

# Display the first few rows
print(data.head())

# Display dataset information
print(data.info())
```

```
   Transaction ID        Date Customer ID  Gender  Age Product Category  \
0               1  24-11-2023     CUST001    Male   34           Beauty
1               2  27-02-2023     CUST002  Female   26         Clothing
2               3  13-01-2023     CUST003    Male   50      Electronics
3               4  21-05-2023     CUST004    Male   37         Clothing
4               5  06-05-2023     CUST005    Male   30           Beauty

   Quantity  Price per Unit  Total Amount
0         3              50           150
1         2             500          1000
2         1              30            30
3         1             500           500
4         2              50           100
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction ID    1000 non-null   int64
 1   Date              1000 non-null   object
 2   Customer ID       1000 non-null   object
 3   Gender            1000 non-null   object
 4   Age               1000 non-null   int64
 5   Product Category  1000 non-null   object
 6   Quantity          1000 non-null   int64
 7   Price per Unit    1000 non-null   int64
 8   Total Amount      1000 non-null   int64
dtypes: int64(5), object(4)
memory usage: 70.4+ KB
None
```

Data Cleaning:

- Check for Missing Values:

In [2]:
```python
# Check for missing values
print(data.isnull().sum())
```

```
Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
dtype: int64
```

- Handle Missing Values: Depending on the results, you can decide to fill or drop missing values.

In [3]:
```
# Example: Drop rows with missing values
data = data.dropna()
```

Data Analysis:

- Total Sales and Revenue:

In [4]:
```
# Calculate total revenue
data['Total Amount'] = data['Quantity'] * data['Price per Unit']
total_revenue = data['Total Amount'].sum()
print(f"Total Revenue: {total_revenue}")
```

```
Total Revenue: 456000
```

- Sales by Product Category:

In [5]:
```
# Group by Product Category
category_sales = data.groupby('Product Category')['Total Amount'].sum().reset_in
print(category_sales)
```

```
  Product Category  Total Amount
0           Beauty        143515
1         Clothing        155580
2      Electronics        156905
```

- Monthly Sales Trend:

In [6]:
```
data['Date'] = pd.to_datetime(data['Date'], dayfirst=True)
```

In [7]:
```
data['Date'] = pd.to_datetime(data['Date'], format='%d-%m-%Y')
```

In [8]:
```
# Convert Date column to datetime
data['Date'] = pd.to_datetime(data['Date'])

# Extract month and year
data['Month'] = data['Date'].dt.to_period('M')

# Group by Month
monthly_sales = data.groupby('Month')['Total Amount'].sum().reset_index()
```

```
# Sort by Month
monthly_sales = monthly_sales.sort_values('Month')
print(monthly_sales)
```

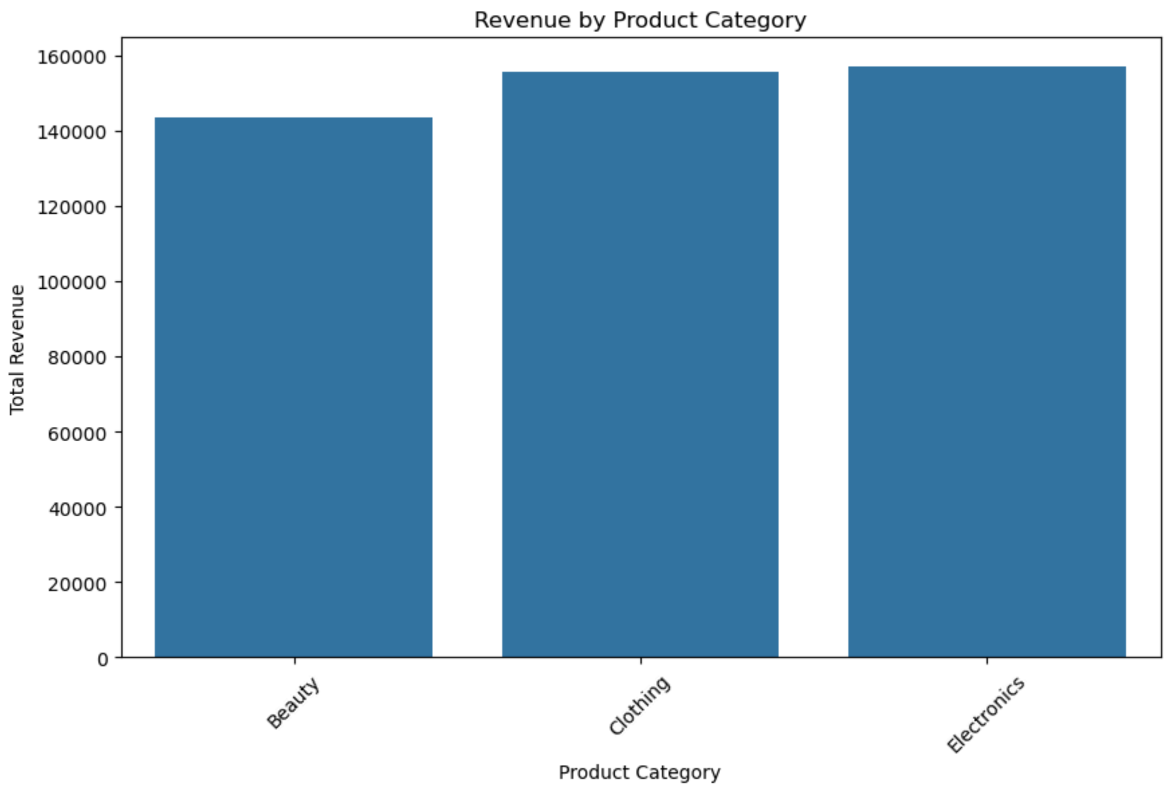
```
      Month  Total Amount
0   2023-01         35450
1   2023-02         44060
2   2023-03         28990
3   2023-04         33870
4   2023-05         53150
5   2023-06         36715
6   2023-07         35465
7   2023-08         36960
8   2023-09         23620
9   2023-10         46580
10  2023-11         34920
11  2023-12         44690
12  2024-01          1530
```

Data Visualization:

- Import Libraries:

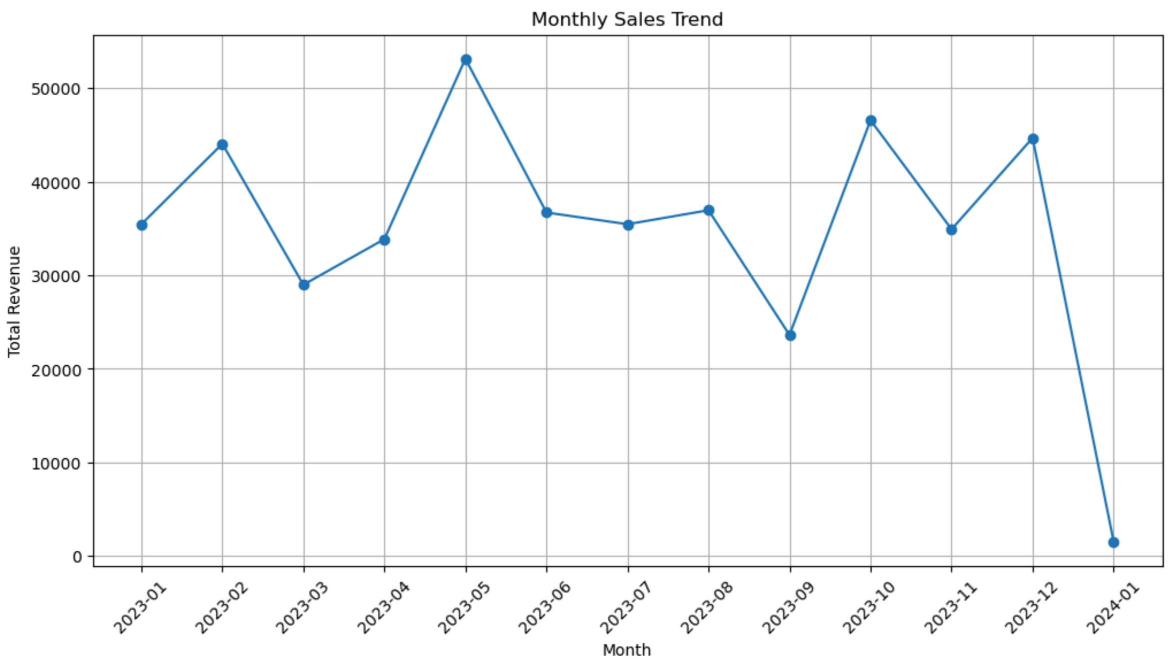
In [9]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [10]:
```
# Bar plot for sales by product category
plt.figure(figsize=(10, 6))
sns.barplot(x='Product Category', y='Total Amount', data=category_sales)
plt.title('Revenue by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.show()
```



- Monthly Sales Trend:

In [11]:
```python
# Line plot for monthly sales trend
plt.figure(figsize=(12, 6))
plt.plot(monthly_sales['Month'].astype(str), monthly_sales['Total Amount'], mark
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



Report Findings:

- Key Metrics: Summarize total revenue, top-performing product categories, and sales trends.
- Visuals: Include the bar and line charts generated above.
- Insights: Provide actionable recommendations based on the analysis, such as identifying peak sales periods or high-performing product categories.