```
from google.colab import files
uploaded = files.upload()
```

Choose Files retail_sales_dataset.csv

retail_sales_dataset.csv(text/csv) - 51673 bytes, last modified: 23/01/2026 - 100% done

```
Saving retail_sales_dataset.csv to retail_sales_dataset.csv
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("retail_sales_dataset.csv")

# Quick view
df.head()
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30

Next steps: Generate code with df New interactive sheet

```
df.info()
df.columns
```

```
<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
Index(['Transaction ID', 'Date', 'Customer ID', 'Gender', 'Age',
       'Product Category', 'Quantity', 'Price per Unit', 'Total Amount'],
      dtype='object')
```

Convert Date to real datetime

```
df["Date"] = pd.to_datetime(df["Date"])
```

```
df.info()
```

```
<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   datetime64[ns]
 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: datetime64[ns](1), int64(5), object(3)
memory usage: 70.4+ KB
```

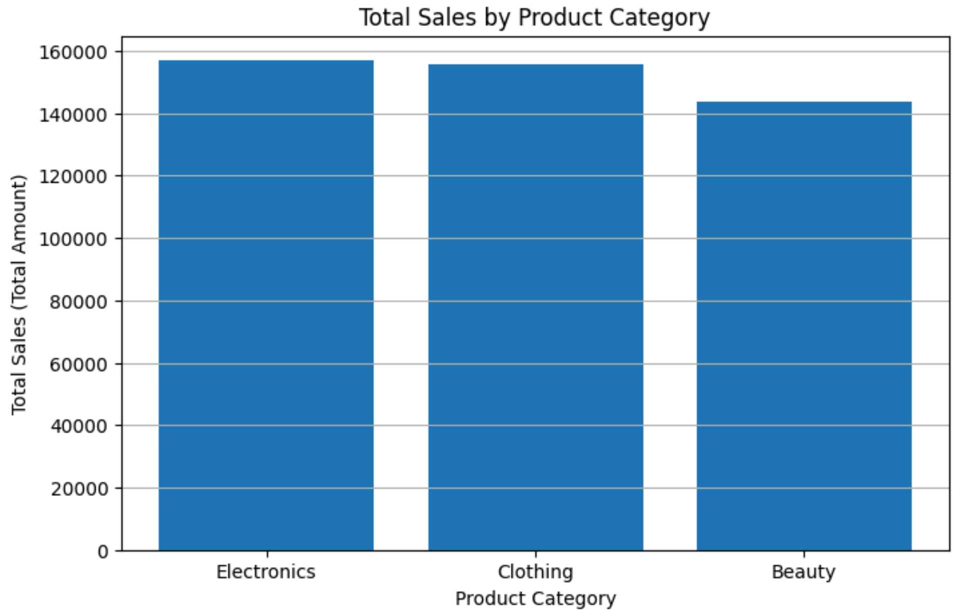
Null Value Checking

```
df.isna().sum()
```

```
                   0
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
```

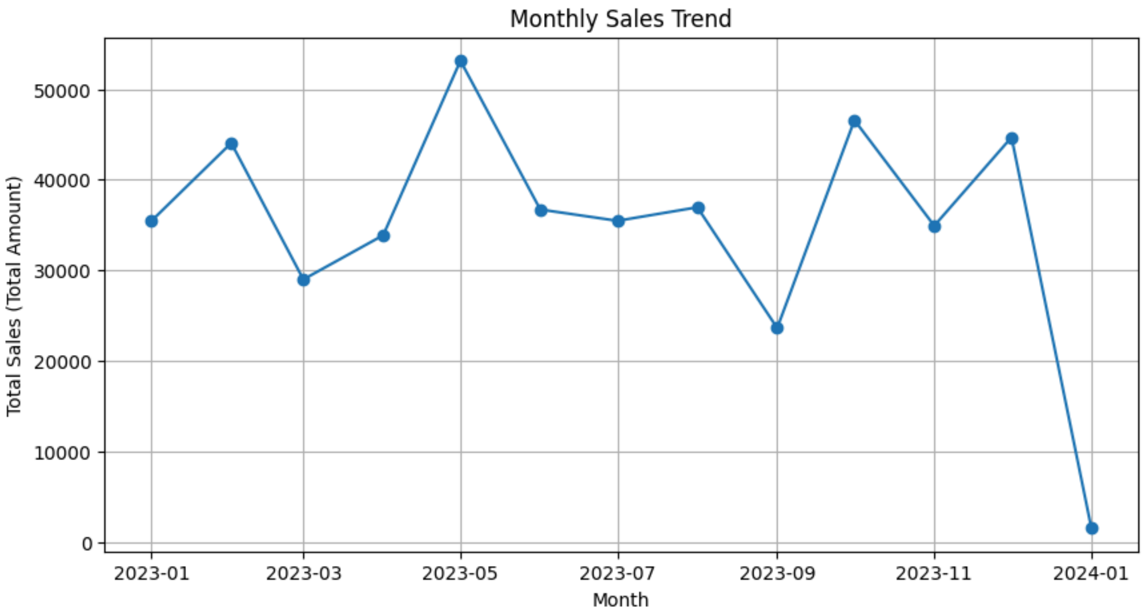
Bar Chart (Top Categories by Total Sales)

```
category_sales = (
    df.groupby("Product Category")["Total Amount"]
    .sum()
    .sort_values(ascending=False)
)

plt.figure(figsize=(8,5))
plt.bar(category_sales.index, category_sales.values)
plt.title("Total Sales by Product Category")
plt.xlabel("Product Category")
plt.ylabel("Total Sales (Total Amount)")
plt.grid(True, axis="y")
plt.show()
```

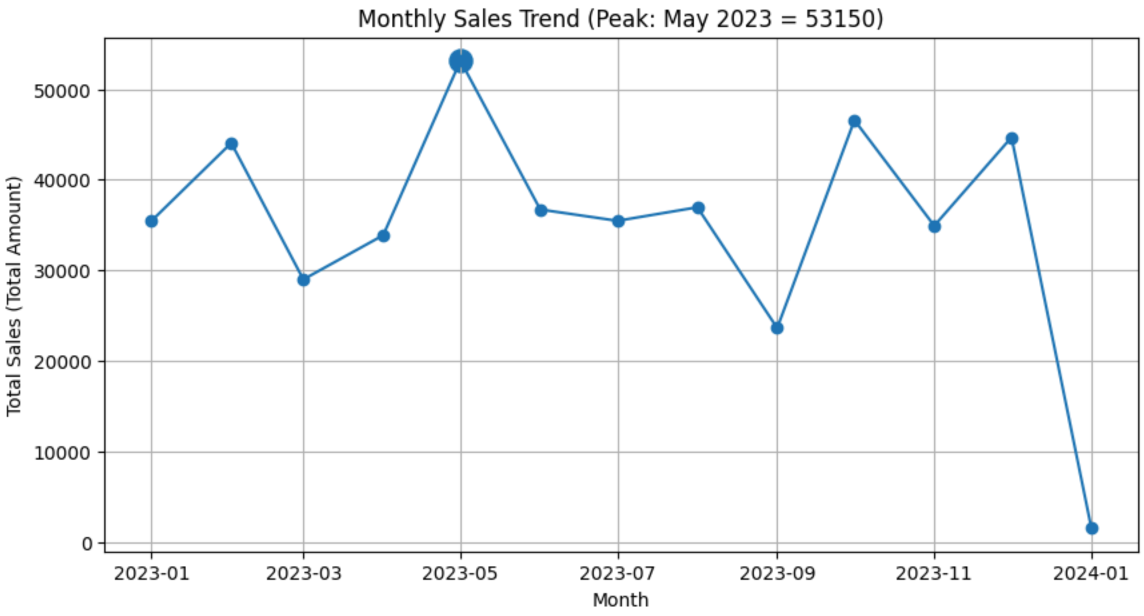


Line Chart (Monthly Sales Trend)

```
df["Month"] = df["Date"].dt.to_period("M").dt.to_timestamp()

monthly_sales = (
    df.groupby("Month")["Total Amount"]
    .sum()
    .sort_index()
)

plt.figure(figsize=(10,5))
plt.plot(monthly_sales.index, monthly_sales.values, marker="o")
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Total Sales (Total Amount)")
plt.grid(True)
plt.show()
```

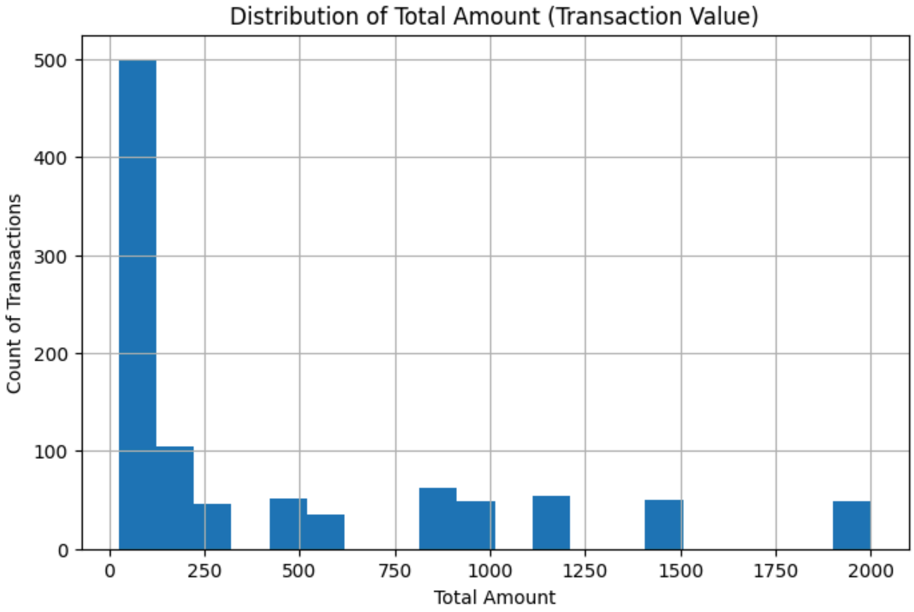


```
peak_month = monthly_sales.idxmax()
peak_value = monthly_sales.max()

plt.figure(figsize=(10,5))
plt.plot(monthly_sales.index, monthly_sales.values, marker="o")
plt.scatter([peak_month], [peak_value], s=150)  # highlight point
plt.title(f"Monthly Sales Trend (Peak: {peak_month.strftime('%b %Y')} = {peak_value})")
plt.xlabel("Month")
plt.ylabel("Total Sales (Total Amount)")
plt.grid(True)
plt.show()
```



Histogram (Distribution of Total Amount)

```
plt.figure(figsize=(8,5))
plt.hist(df["Total Amount"], bins=20)
plt.title("Distribution of Total Amount (Transaction Value)")
plt.xlabel("Total Amount")
plt.ylabel("Count of Transactions")
plt.grid(True)
plt.show()
```

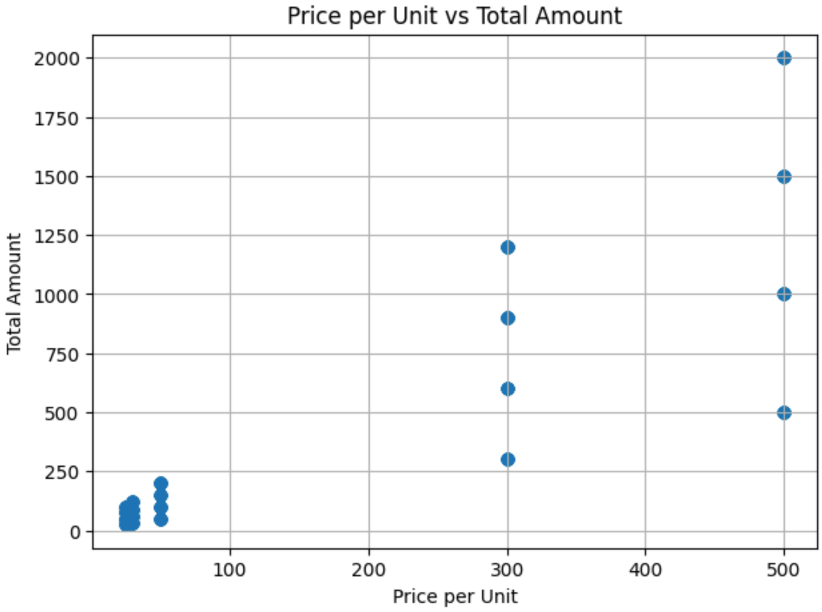


```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Scatter Plot

```
plt.figure(figsize=(7,5))
plt.scatter(df["Price per Unit"], df["Total Amount"], alpha=0.6)
plt.title("Price per Unit vs Total Amount")
plt.xlabel("Price per Unit")
plt.ylabel("Total Amount")
plt.grid(True)
plt.show()
```



3 Insights

1.Category insight: Electronics has the highest total sales, slightly above Clothing, while Beauty is the lowest. This suggests Electronics drives the most revenue overall.

2.Trend insight: Monthly sales peak around May 2023, showing a strong spike compared to other months. Also, January 2024 is extremely low, likely because it contains only a small number of days/partial data.

3.Distribution + correlation insight: The histogram shows many transactions are small, but a few very large transactions exist (outliers). The scatter plot shows Total Amount increases strongly with Price per Unit, meaning expensive items are a major reason for high bills.