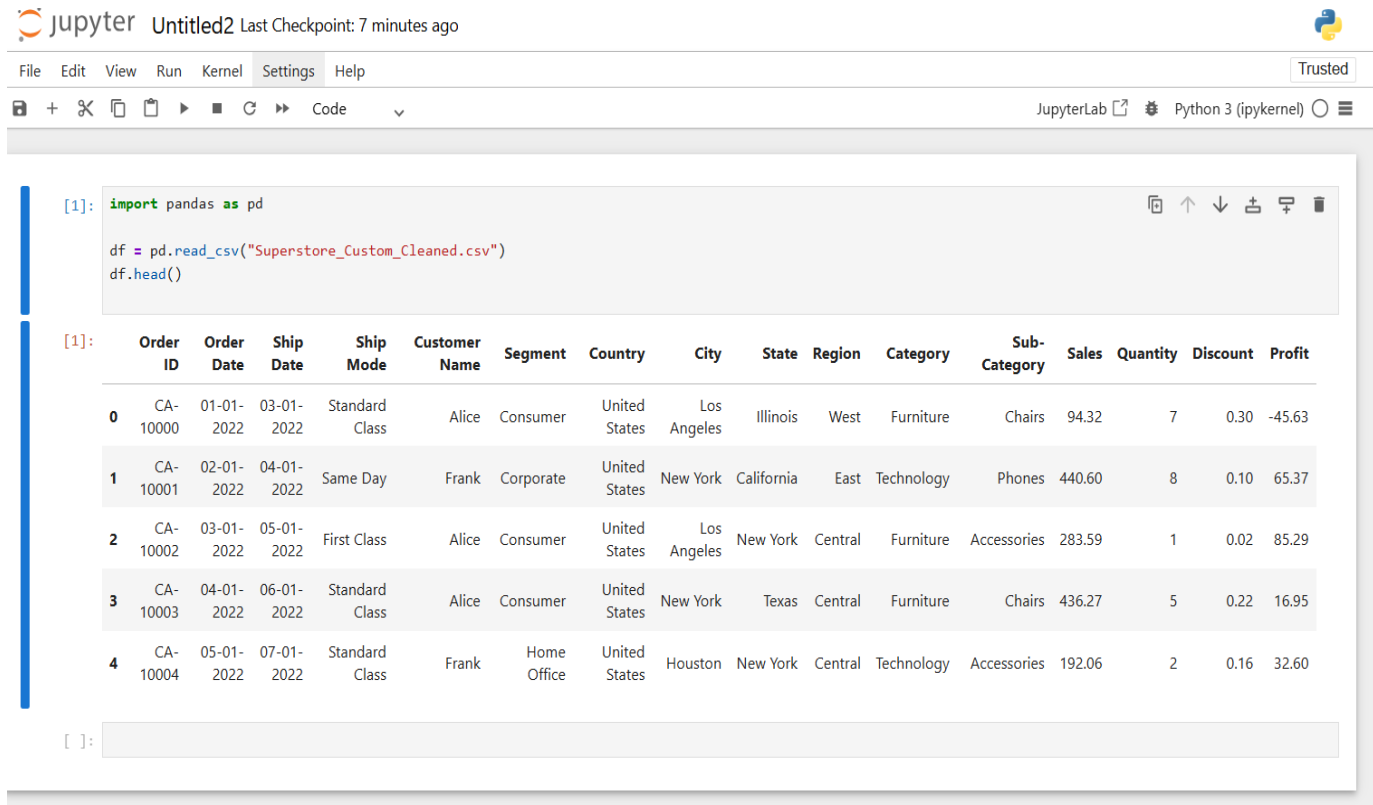


Task 5: Exploratory Data Analysis (EDA)

PDF report of findings



The screenshot shows a JupyterLab interface with a notebook titled 'Untitled2'. The last checkpoint was 7 minutes ago. The notebook has a menu bar with 'File', 'Edit', 'View', 'Run', 'Kernel', 'Settings', and 'Help'. The toolbar includes icons for file operations and a 'Code' button. The notebook content shows the following code cell:

```
[1]: import pandas as pd

df = pd.read_csv("Superstore_Custom_Cleaned.csv")
df.head()
```

The output of the code cell is a table with 17 columns: Order ID, Order Date, Ship Date, Ship Mode, Customer Name, Segment, Country, City, State, Region, Category, Sub-Category, Sales, Quantity, Discount, and Profit. The table displays the first five rows of data.

	Order ID	Order Date	Ship Date	Ship Mode	Customer Name	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	CA-10000	01-01-2022	03-01-2022	Standard Class	Alice	Consumer	United States	Los Angeles	Illinois	West	Furniture	Chairs	94.32	7	0.30	-45.63
1	CA-10001	02-01-2022	04-01-2022	Same Day	Frank	Corporate	United States	New York	California	East	Technology	Phones	440.60	8	0.10	65.37
2	CA-10002	03-01-2022	05-01-2022	First Class	Alice	Consumer	United States	Los Angeles	New York	Central	Furniture	Accessories	283.59	1	0.02	85.29
3	CA-10003	04-01-2022	06-01-2022	Standard Class	Alice	Consumer	United States	New York	Texas	Central	Furniture	Chairs	436.27	5	0.22	16.95
4	CA-10004	05-01-2022	07-01-2022	Standard Class	Frank	Home Office	United States	Houston	New York	Central	Technology	Accessories	192.06	2	0.16	32.60

In above ss I have Loaded the dataset in Jupyter notebook.

Below ss is for matplotlib libraries.

```
[2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")
```

Matplotlib is building the font cache; this may take a moment.

df.info() - Shows column names, data types, and missing values

```
[3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 16 columns):
#   Column             Non-Null Count  Dtype  
---  --
0   Order ID            800 non-null    object  
1   Order Date          800 non-null    object  
2   Ship Date           800 non-null    object  
3   Ship Mode           800 non-null    object  
4   Customer Name       800 non-null    object  
5   Segment             800 non-null    object  
6   Country             800 non-null    object  
7   City                800 non-null    object  
8   State               800 non-null    object  
9   Region              800 non-null    object  
10  Category            800 non-null    object  
11  Sub-Category        800 non-null    object  
12  Sales               800 non-null    float64  
13  Quantity            800 non-null    int64   
14  Discount            800 non-null    float64  
15  Profit              800 non-null    float64  
dtypes: float64(3), int64(1), object(12)
memory usage: 100.1+ KB
```

df.describe() - Shows statistical summary for numerical columns

```
memory usage: 100.1+ KB

[4]: df.describe()

[4]:
```

	Sales	Quantity	Discount	Profit
count	800.000000	800.000000	800.000000	800.000000
mean	250.888275	4.913750	0.150025	24.664875
std	139.809397	2.524733	0.087261	42.952770
min	10.460000	1.000000	0.000000	-49.560000
25%	130.150000	3.000000	0.080000	-10.542500
50%	248.775000	5.000000	0.150000	26.180000
75%	372.022500	7.000000	0.220000	62.850000
max	497.860000	9.000000	0.300000	99.960000

```
[ ]:
```

In above two screenshots I have checked basic details and summary.

```
[5]: print("Missing values in each column:")
      print(df.isnull().sum())

Missing values in each column:
Order ID      0
Order Date    0
Ship Date     0
Ship Mode     0
Customer Name  0
Segment       0
Country       0
City          0
State         0
Region        0
Category      0
Sub-Category  0
Sales         0
Quantity      0
Discount      0
Profit        0
dtype: int64

[6]: print("\nTotal duplicate rows:", df.duplicated().sum())

Total duplicate rows: 0

[ ]:
```

In above screenshot I have checked missing and duplicate values

Below are the categorical columns with `value_counts()` that are showing **how many unique items** are in each category like Category, Segment, Region, etc.

```
print(df['Category'].value_counts())

print(df['Sub-Category'].value_counts())

print(df['Segment'].value_counts())

print(df['Region'].value_counts())
```

```
[7]: print(df['Category'].value_counts())
```

```
Category
Technology      277
Office Supplies  263
Furniture        260
Name: count, dtype: int64
```

```
[8]: print(df['Sub-Category'].value_counts())
```

```
Sub-Category
Paper          175
Binders        165
Chairs         156
Accessories    156
Phones         148
Name: count, dtype: int64
```

```
[9]: print(df['Segment'].value_counts())
```

```
Segment
Consumer      274
Corporate      272
Home Office    254
Name: count, dtype: int64
```

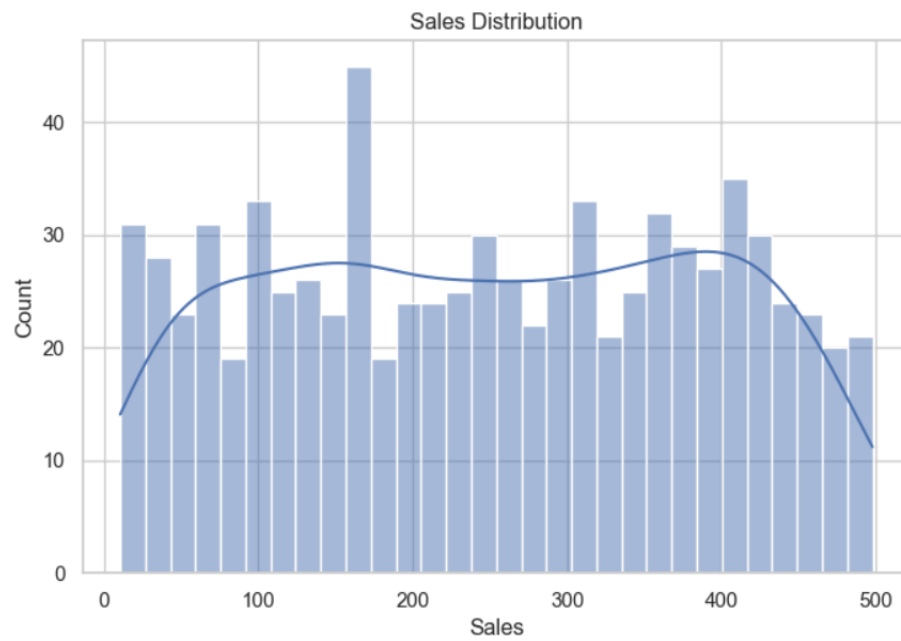
Histogram – To see distribution of numeric columns (e.g., Sales, Profit)
`plt.figure(figsize=(8,5))`

`sns.histplot(df['Sales'], bins=30, kde=True)`

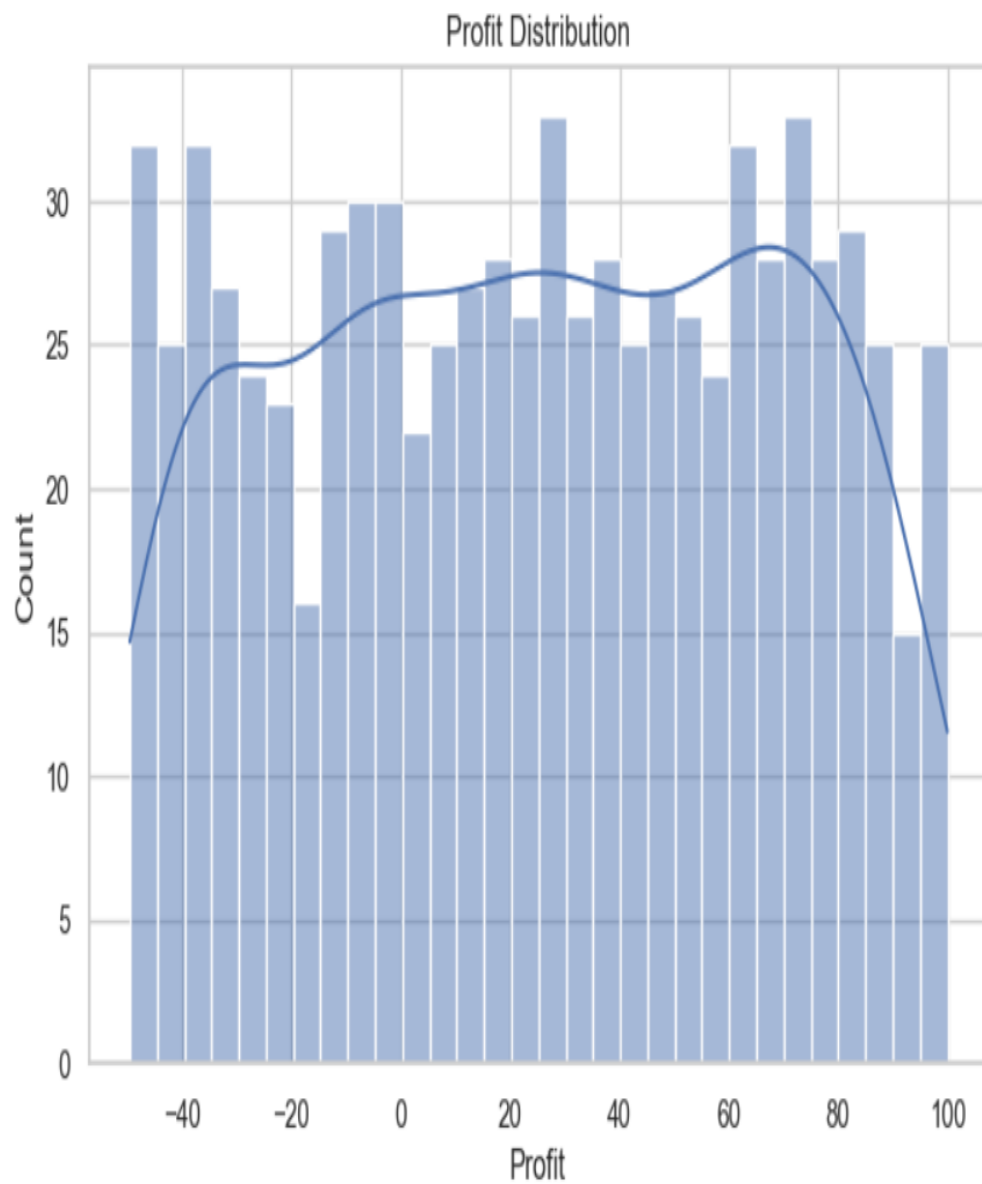
`plt.title("Sales Distribution")`

`plt.show()`

```
[11]: plt.figure(figsize=(8,5))
sns.histplot(df['Sales'], bins=30, kde=True)
plt.title("Sales Distribution")
plt.show()
```

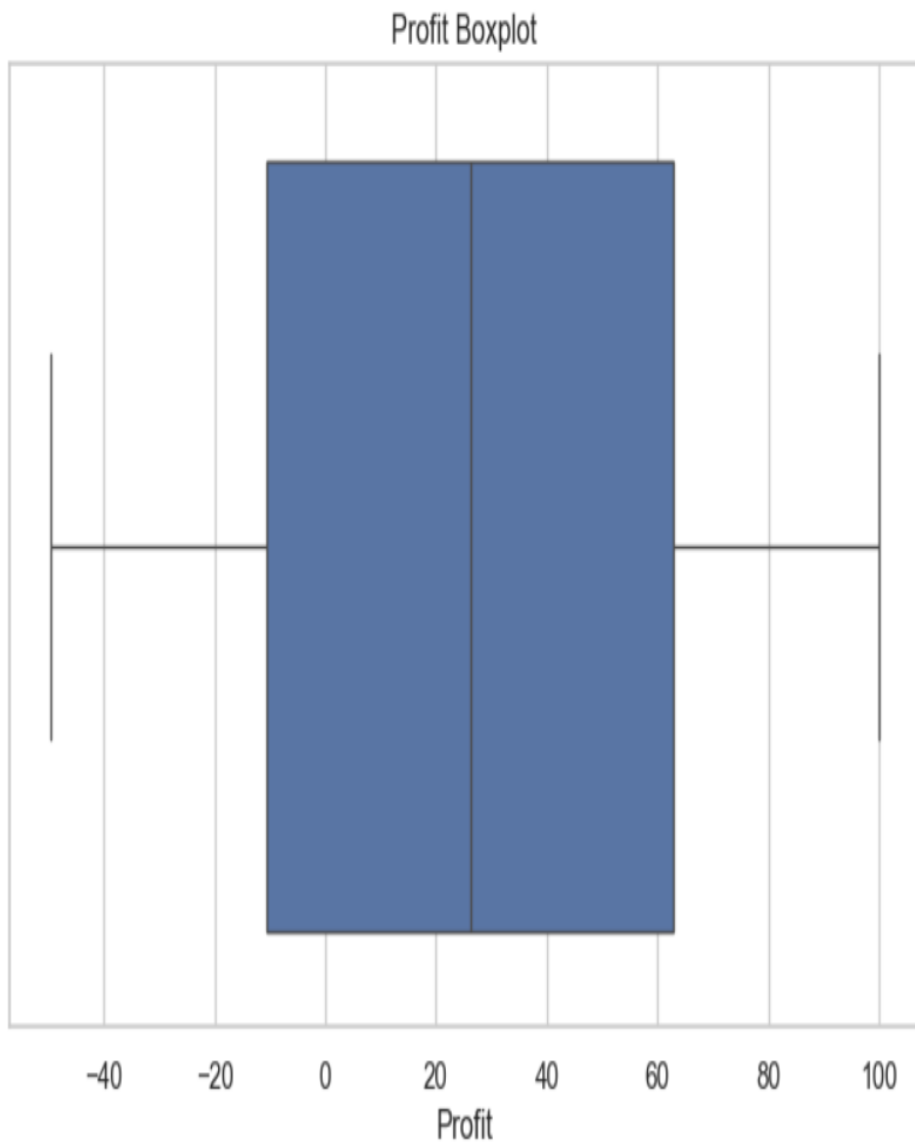


```
[12]: plt.figure(figsize=(8,5))
sns.histplot(df['Profit'], bins=30, kde=True)
plt.title("Profit Distribution")
plt.show()
```



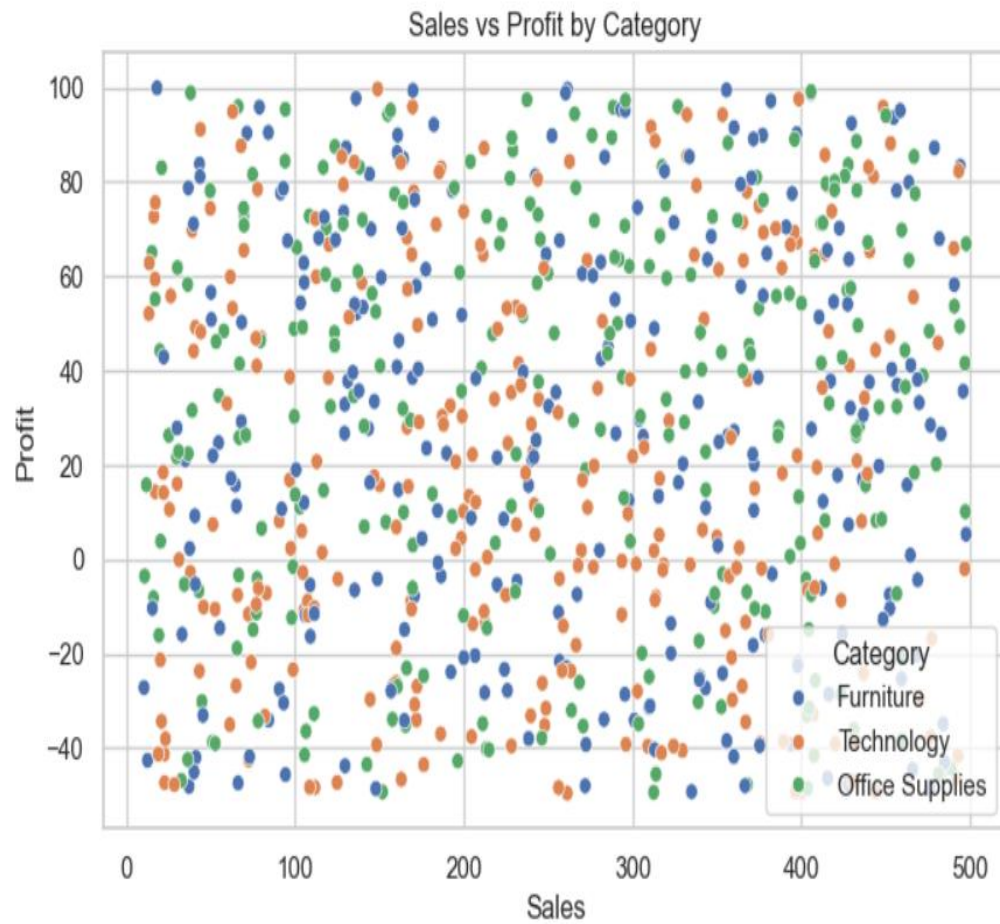
Boxplot – To detect outliers

```
[13]: plt.figure(figsize=(8,5))
sns.boxplot(x=df['Profit'])
plt.title("Profit Boxplot")
plt.show()
```



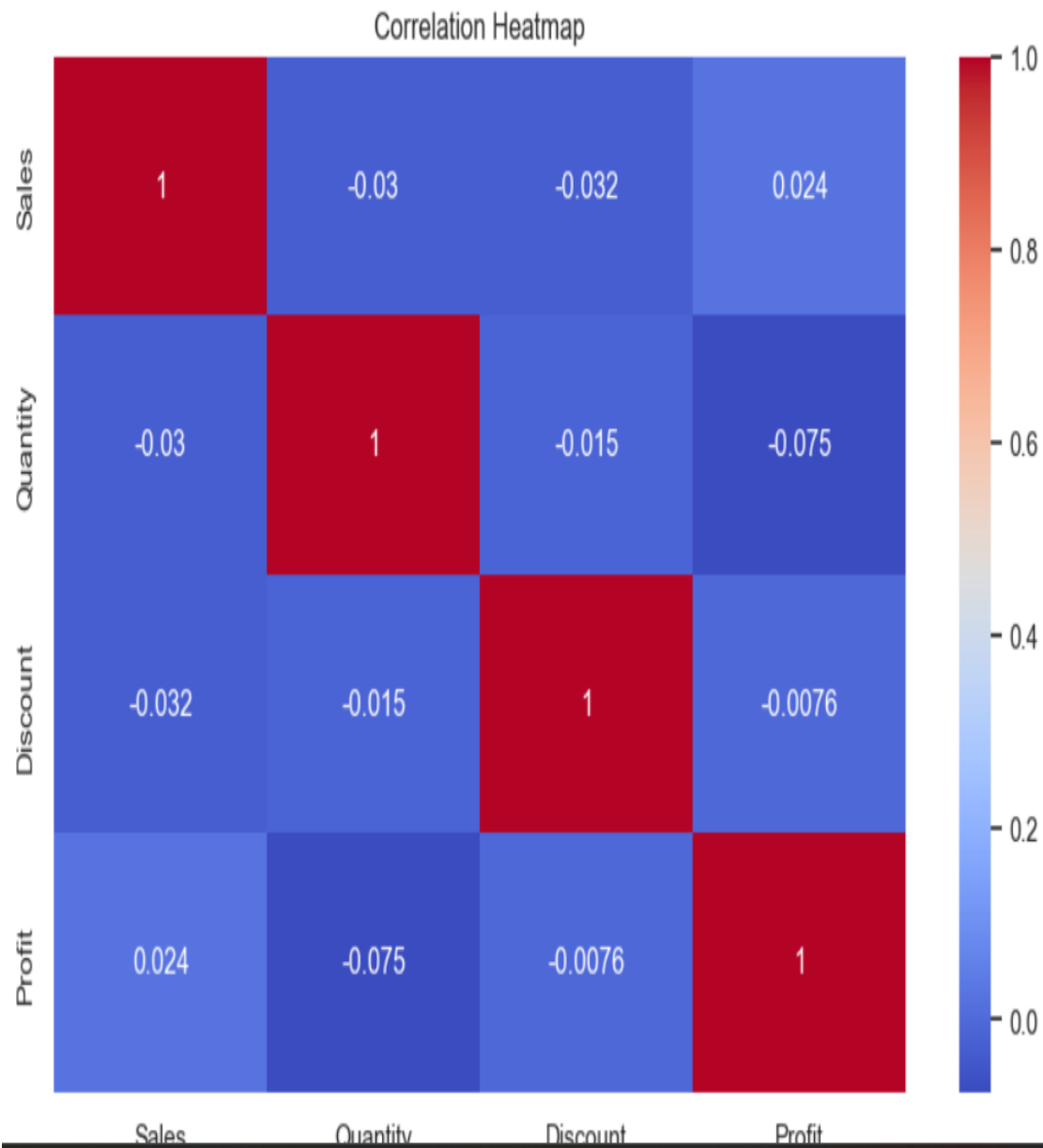
Scatterplot – To see relationship between Sales and Profit

```
[14]: plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x='Sales', y='Profit', hue='Category')
plt.title("Sales vs Profit by Category")
plt.show()
```



Heatmap – To check correlations

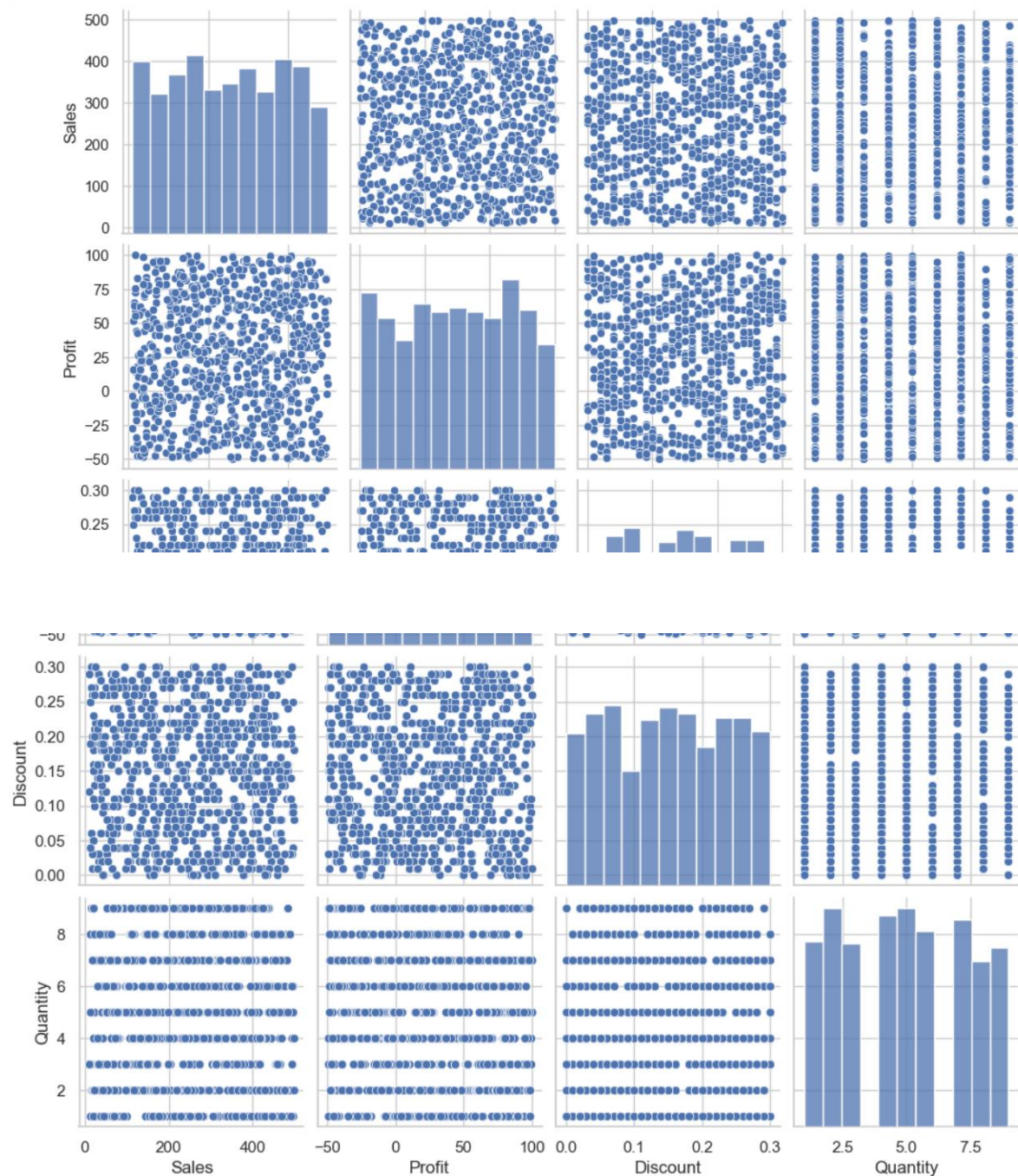

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



Pairplot – Explore relationships across multiple features

```
sns.pairplot(df[['Sales', 'Profit', 'Discount', 'Quantity']])
```

[16]: <seaborn.axisgrid.PairGrid at 0x1ce2e61cec0>



Identifying Relationships and Trends

I checked how different columns (like Sales, Profit, Discount, Quantity) are related and if there are any patterns.

Key Trends and Relationships Found:

1. Sales vs Profit:

- No strong pattern — high sales didn't always lead to high profits.
- Some high-sale orders still had **negative profit** (loss), possibly due to discounts.

2. Discount vs Profit:

- Strong **negative relationship** — more discount usually means less or negative profit.
- Giving big discounts regularly may be hurting the business.

3. Category-wise Trends:

- **Technology** category seems to perform better in terms of profit.
- **Furniture** has many losses — needs more attention.

4. Quantity and Profit:

- No clear relationship — just selling more quantity doesn't guarantee more profit.

➤ Observations for Visuals & Final Summary

This is to analyze what each chart means and summarize your findings.

Histogram – Sales & Profit

A histogram shows how values are spread — for example, how many orders had low sales vs high sales.

I'm learned

- Most of the sales values are small — like ₹100 to ₹500.
- Very few orders had very high sales — like ₹2000 or more.
- For profit, some values were even negative (loss).

This means: Most of the time, the company is selling small-value items, and sometimes, they are even losing money on a sale.

Boxplot – Profit

A boxplot shows outliers - unusual values that are very different from others.

I'm learned

- Many orders have very low or very high profit values, far away from the average.
- These faraway points are called outliers.

This means: Some orders are making a lot of profit, and some are causing big losses — the company needs to check those.

Scatterplot – Sales vs Profit

A scatterplot shows the relationship between two things — in this case, Sales and Profit.

I'm learned

- Just because a product this means: Selling more doesn't always mean earning more. Sometimes, big sales come with big losses (maybe due to discounts or high costs).
- has high sales, doesn't mean it made a high profit.
- Some sales are big, but the company still lost money on them.
- Different categories (like furniture or technology) behave differently.

Heatmap – Correlation

A heatmap with correlation shows how strongly related two things are.

I'm learned

- Sales and profit are not strongly linked.
- Discount and profit have a negative relationship — when discounts go up, profits go down.
- Quantity sold doesn't have much effect on profit.

This means Giving more discounts is dangerous — it usually reduces the company's profit.

Pairplot

This chart shows small graphs between each pair of features - it's like checking all relationships at once.

I'm learned

- You can see which variables are related.
- It gives a full picture, but might look crowded — it's good for exploring patterns and clusters.

This helps in understanding the overall data behavior, and spotting hidden patterns.

Final Summary

- Most orders are for small amounts.
- Some products or orders are causing losses — especially if discounts are high.
- Big sales do not always mean big profits.
- Giving too much discount reduces profit.
- We should be careful about which products and discounts we give.

➤ Summary of Findings

Sales and Profit Distribution: Most orders have low sales; few are very high. Profit values vary and include losses.

1. **Outliers Detected:** Extreme profit values found using boxplots — worth investigating.
2. **Relationship Between Sales and Profit:** No strong connection; some high-sales orders cause losses.
3. **Impact of Discounts:** More discounts often result in lower profits — a risky strategy.
4. **Category Behavior:** Categories like Technology perform better; Furniture has more losses.
5. **Relationships and Trends:**
 - Discounts hurt profit.
 - No direct link between quantity and profit.
 - Category and region impact results.
6. **Overall Insight:** The company must review its discount policy, focus on profitable categories, and check why losses happen on high-sale items.

➤ Outcome

Through this Exploratory Data Analysis (EDA) project on the Superstore dataset.

I have

Gained confidence in working with real-world business data
Learned how to use Pandas, Matplotlib, and Seaborn for data exploration Developed
the skill to find patterns, such as which product categories perform well
Understood how to identify trends, like how discounts affect profit
Practiced detecting anomalies and outliers, such as extreme profit/loss values
Improved my ability to summarize insights visually and statistically
Learned to communicate findings clearly using graphs and observations