

# Sales Tracker Project

This notebook demonstrates how raw sales data can be cleaned, analyzed, and visualized.

It began as a simple coding exercise but was expanded into a project showcasing skills in data cleaning, statistics, and visualization.

```
In [14]: import pandas as pd
import matplotlib.pyplot as plt
```

## Load Dataset

We start by importing the raw sales dataset ( `sales.csv` ).

```
In [16]: df = pd.read_csv("sales.csv")
df.head()
```

```
Out[16]:
```

	ID	Product	Sales
0	1	Shirt	120.50
1	2	Jeans	250.00
2	3	Shoes	NaN
3	4	Hat	75.25
4	5	Shirt	130.00

## Data Cleaning

- Removed rows with missing or invalid sales values (e.g., `NaN` ).
- Converted the `Sales` column to numeric format for analysis.

```
In [18]: df_clean = df.dropna().copy()
df_clean.loc[:, "Sales"] = pd.to_numeric(df_clean["Sales"], errors="coerce")
df_clean.head()
```

Out [18]:

	ID	Product	Sales
0	1	Shirt	120.50
1	2	Jeans	250.00
3	4	Hat	75.25
4	5	Shirt	130.00
5	6	Shoes	200.75

## Basic Statistics

We compute:

- Total Sales
- Average Sale
- Top-Selling Product

```
In [20]: total_sales = df_clean["Sales"].sum()
average_sales = df_clean["Sales"].mean()
top_product = df_clean.groupby("Product")["Sales"].sum().idxmax()

print("Total Sales: $", round(total_sales, 2))
print("Average Sale: $", round(average_sales, 2))
print("Top-Selling Product:", top_product)
```

Total Sales: \$ 1237.0  
 Average Sale: \$ 154.62  
 Top-Selling Product: Jeans

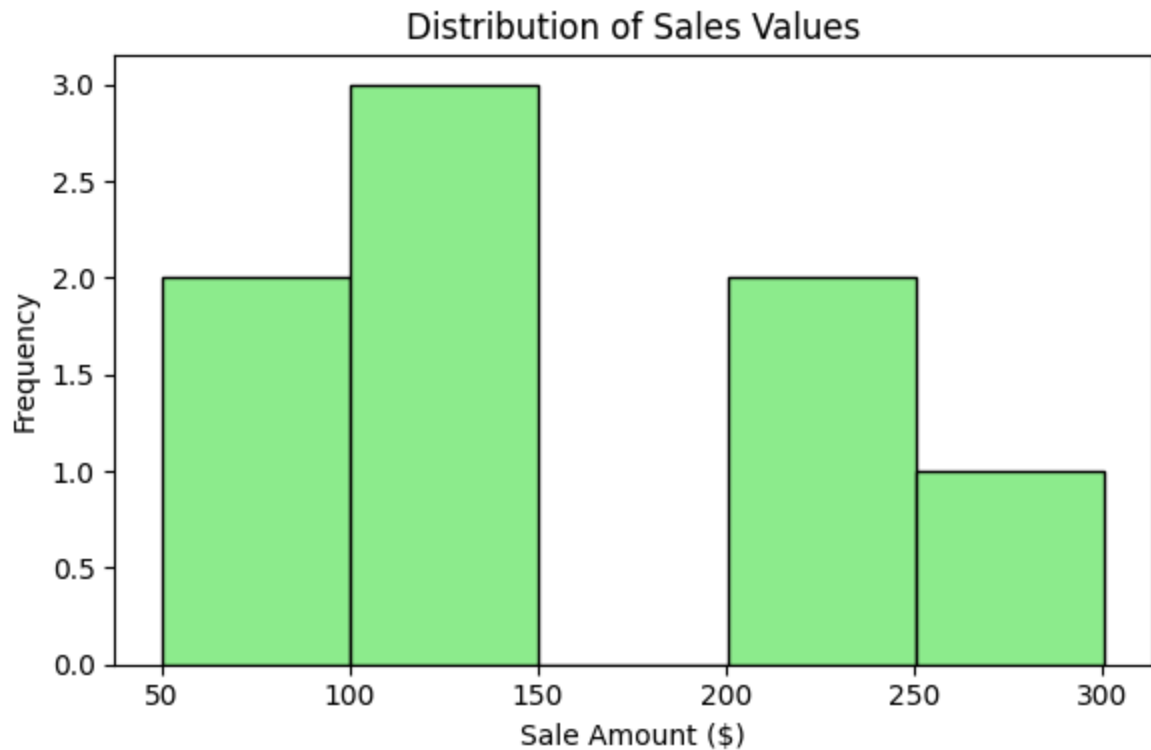
## Sales Value Distribution

This histogram shows the distribution of sales values across the dataset. It provides insight into the spread and frequency of transaction sizes.

For example:

- Lower bars indicate less common sales values.
- Taller bars highlight the ranges where most transactions fall.

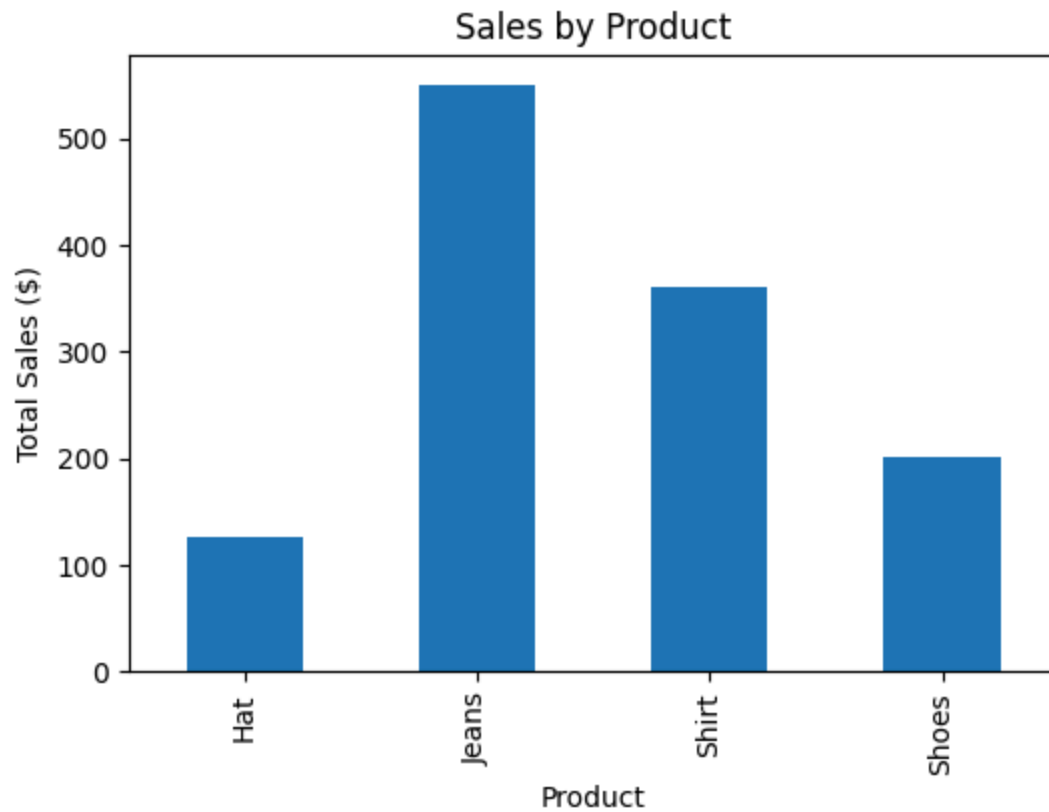
```
In [25]: # Histogram of individual sales values
plt.figure(figsize=(6,4))
plt.hist(df_clean["Sales"], bins=5, color="lightgreen", edgecolor="black")
plt.title("Distribution of Sales Values")
plt.xlabel("Sale Amount ($)")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```



## Visualization

This bar chart shows total sales by product, making it easy to spot which category performs best.

```
In [22]: product_sales = df_clean.groupby("Product")["Sales"].sum()
product_sales.plot(kind="bar", title="Sales by Product", figsize=(6,4))
plt.ylabel("Total Sales ($)")
plt.show()
```



## Reflection

This project demonstrates:

- Programming skills: cleaning and analyzing datasets with Python/pandas.
- Analytical skills: extracting insights like averages and top products.
- Visualization skills: presenting results in an intuitive chart.

It shows how I can take a classroom-style project and turn it into something closer to real-world analysis, making it relevant for internships in data analytics, business analysis, or tech consulting.