Goals of the Project:

1. **Data Cleaning and Preparation: To clean the dataset by handling missing values, duplicates, and ensuring data types are appropriate for analysis.**
2. **Exploratory Data Analysis (EDA): To perform EDA to understand sales trends, customer behavior, and product performance.**
3. **Sales Analysis: To analyze sales data by region, product category, and customer segments to identify top-performing areas and products.**
4. **Time-Based Analysis: To examine sales trends over time and understand seasonal patterns in sales.**
5. **Customer Behavior Analysis: To analyze customer spending patterns and identify top customers.**
6. **Reporting Insights: To summarize findings and provide actionable insights for business decision-making.**

Materials and Methods:

- **Data Source: The project utilizes a sales dataset in CSV format, which includes various attributes such as date, product details, sales figures, and customer information.**
- **Libraries Used:**
- **Pandas: For data manipulation and analysis.**
- **NumPy: For numerical operations.**
- **Seaborn and Matplotlib: For data visualization.**
- **Warnings: To suppress warnings during execution.**
- **Data Processing Steps:**

1. **Load the dataset and perform an initial overview (info, summary statistics, missing values).**
2. **Handle missing values and duplicates.**
3. **Convert date columns to datetime format and create new features for analysis.**
4. **Conduct univariate and bivariate analyses to visualize distributions and relationships.**
5. **Perform group-by operations to summarize sales by different categories (region, product, customer.**
6. **Analyze trends over time and customer spending behavior.**

Project Outcome & Insights:

1. **Sales Performance: Identified top-performing regions and product categories, providing insights into where the business is thriving.**
2. **Customer Insights: Recognized top customers based on spending, which can inform targeted marketing strategies.**
3. **Discount Analysis: Analyzed the impact of discounts on sales, revealing potential areas for optimizing pricing strategies.**
4. **Sales Trends: Visualized sales trends over time, helping to identify seasonal patterns and inform inventory management.**
5. **Return Rate Analysis: Provided insights into customer return rates by category, which can guide product development and marketing efforts.**

Feature Engineering:

1. **Date Conversion: Converted the 'Date' column to a datetime format to facilitate time-based analysis.**
2. **New Features Created:**

- **Order_year: Extracted the year from the date for yearly analysis.**
- **Order_month: Extracted the month from the date for monthly analysis.**
- **Order_weekday: Extracted the day of the week for understanding weekly sales patterns.**
- **Discount_amount: Calculated the discount amount based on quantity, price per unit, and discount percentage.**
- **Sales_after_discount: Calculated the sales amount after applying discounts.**

3. **Categorical Encoding: Ensured categorical variables are ready for analysis by summarizing counts and distributions.**

This structured overview encapsulates the essence of the project, providing clarity on its objectives, methodologies, outcomes, and the feature engineering process employed.

Key Questions and Insights to be Addressed:

 # Assuming df is your cleaned DataFrame

 # 1. Which regions generate the most sales?

```
sales_by_region = df.groupby('Region')['Total
Sales'].sum().sort_values(ascending=False)

print("\nSales by Region:\n", sales_by_region)
```

Sales by Region:

Region

East 131345.903921

North 116908.619474

West 115677.868884

South 96385.051108

 # 2. What product categories are performing best?

```
sales_by_category = df.groupby('Category')['Total
Sales'].sum().sort_values(ascending=False)

print("\nSales by Category:\n", sales_by_category)
```

Sales by Category:

Category

Stationery 162224.983280

Appliances 151590.299004

Clothing 146502.161102

 # 3. Who are the top customers and how much do they spend?

```
top_customers = df.groupby('Customer ID')['Total
Sales'].sum().sort_values(ascending=False).head(10)

print("\nTop 10 Customers by Spending:\n", top_customers)
```

```
Top 10 Customers by Spending:

 Customer ID

C226   1837.005330

C595   1784.815556

C888   1685.488822

C907   1621.021129

C265   1585.502994

C304   1575.373551

C942   1565.698321

C531   1564.776752

C61    1543.345728

C206   1538.725299
```

```
# 4. How does discounting impact total sales?

discount_sales_correlation = df[['Total Sales', 'Discount (%)']].corr()

print("\nCorrelation between Total Sales and Discounts:\n", discount_sales_correlation)
```

```
Correlation between Total Sales and Discounts:

             Total Sales  Discount (%)

Total Sales     1.000000       -0.069925

Discount (%)  -0.069925        1.000000
```

```
5. What are the sales trends over time?

df['Order_month'] = df['Date'].dt.to_period('M')  # Ensure Order_month is in the correct
format
```

```
sales_trends = df.groupby('Order_month')['Total Sales'].sum()

print("\nMonthly Revenue:\n", sales_trends)
```

```
Monthly Revenue:

 Order_month
2023-01    10886.465695
2023-02    15406.581363
2023-03    13185.271353
2023-04    12918.833348
2023-05    13363.724454
2023-06    14649.261927
2023-07    15155.325017
2023-08    12423.623065
2023-09    12904.277375
2023-10    15677.205469
2023-11    14578.996698
2023-12    13185.178206
2024-01    14245.335981
2024-02    10758.990608
2024-03    14400.918315
2024-04    12235.995520
2024-05    12685.092354
2024-06    16653.999609
2024-07    13181.873489
2024-08    13683.947252
```

2024-09 15979.522158

2024-10 10643.330429

2024-11 13600.277111

2024-12 16567.778696

2025-01 15189.298916

2025-02 11250.693735

2025-03 16216.432530

2025-04 12343.591900

2025-05 18296.945166

2025-06 14505.874709

2025-07 16433.812009

2025-08 15491.082436

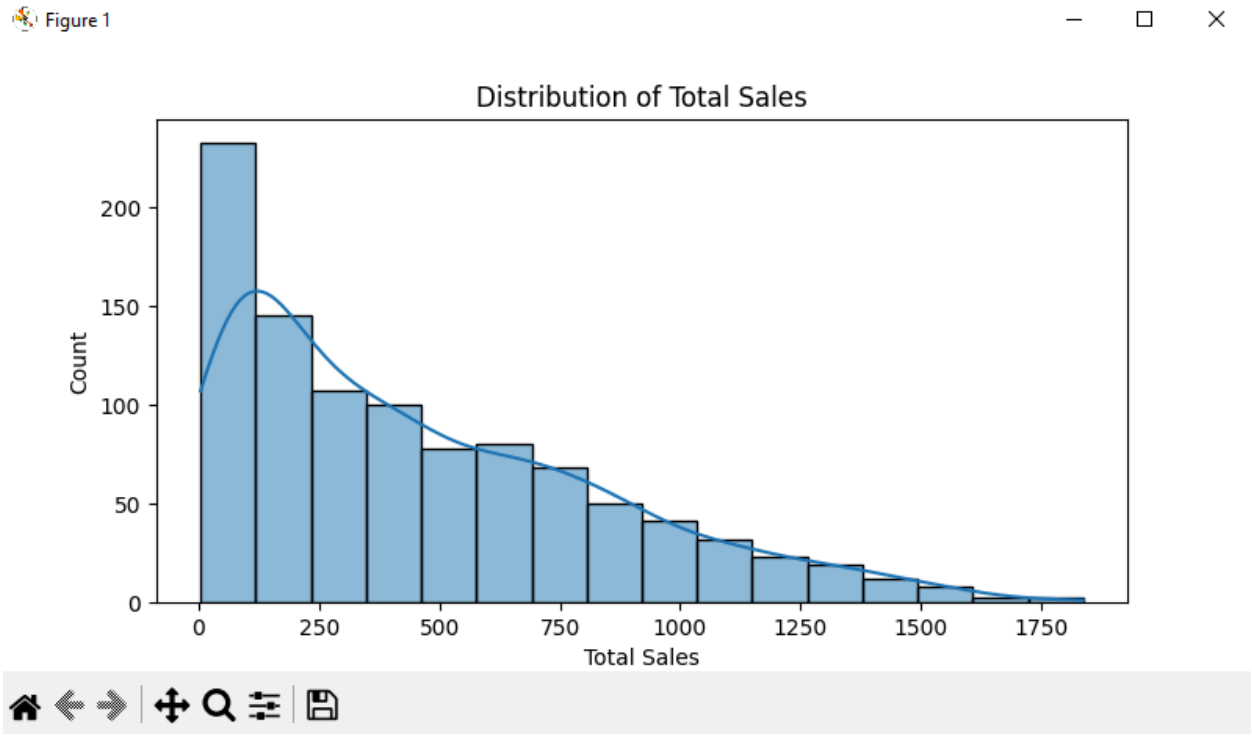
2025-09 11617.906494

```
# Plotting sales trends over time
plt.figure(figsize=(12, 6))
sales_trends.plot()
plt.title("Sales Trend Over Time")
plt.xlabel("Order Month")
plt.ylabel("Total Sales")
plt.grid()
plt.show()
# 6. What is the average discount rate by payment method?
average_discount_by_payment = df.groupby('Payment Method')['Discount (%)'].mean()
print("\nAverage Discount Rate by Payment Method:\n", average_discount_by_payment)
```

Average Discount Rate by Payment Method:

Payment Method	
Cash	9.856540
Credit Card	10.278810
Debit Card	10.053279
PayPal	9.892000

Visualization:



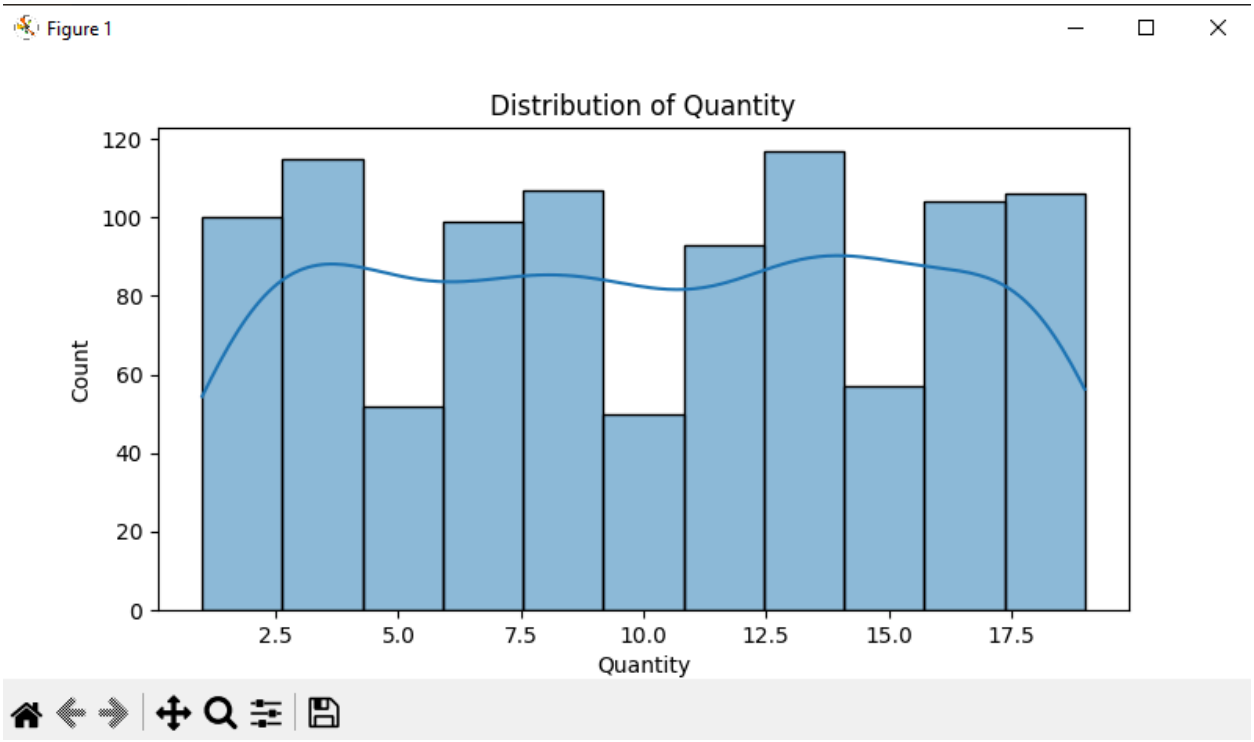


Figure 1
Distribution of Quantity
Count
Quantity
0
20
40
60
80
100
120
2.5
5.0
7.5
10.0
12.5
15.0
17.5

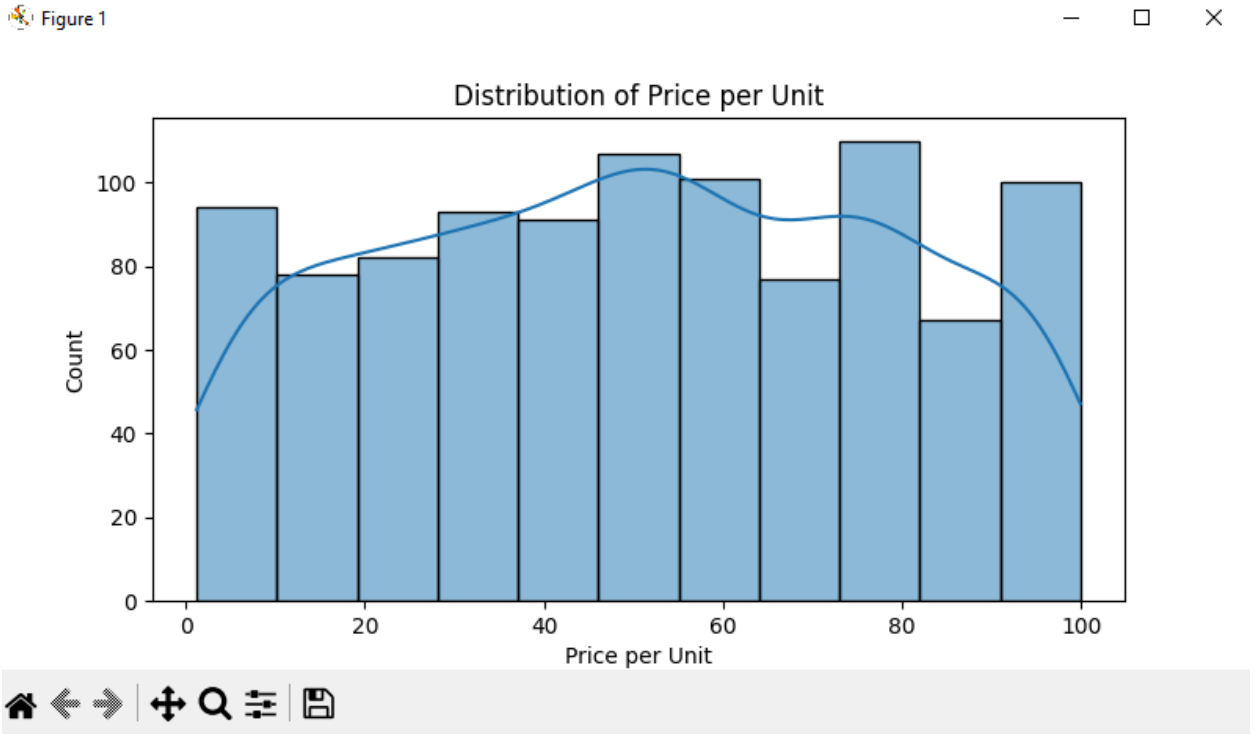


Figure 1
Distribution of Price per Unit
Count
Price per Unit
0
20
40
60
80
100
0
20
40
60
80
100

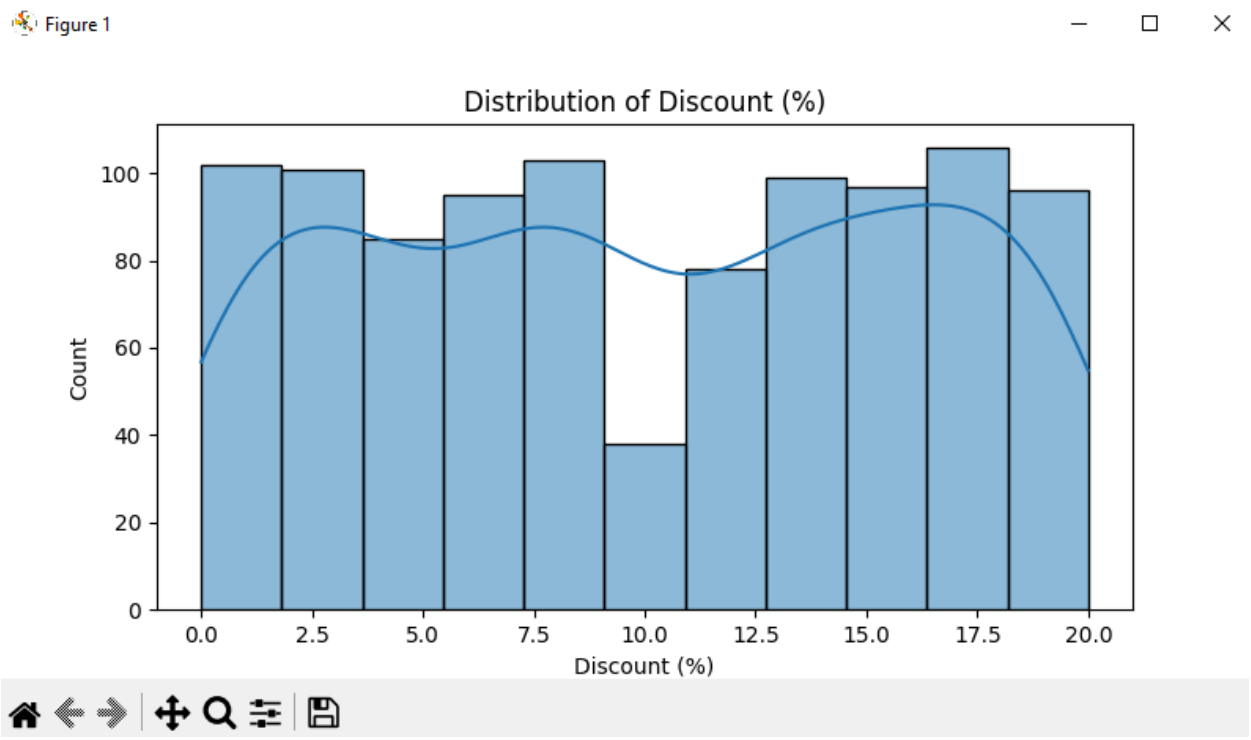


Figure 1
Distribution of Discount (%)
Count
Discount (%)
0
20
40
60
80
100
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Figure 1
Distribution of Sales_after_discount
Count
Sales_after_discount
0
50
100
150
200
0
250
500
750
1000
1250
1500
1750

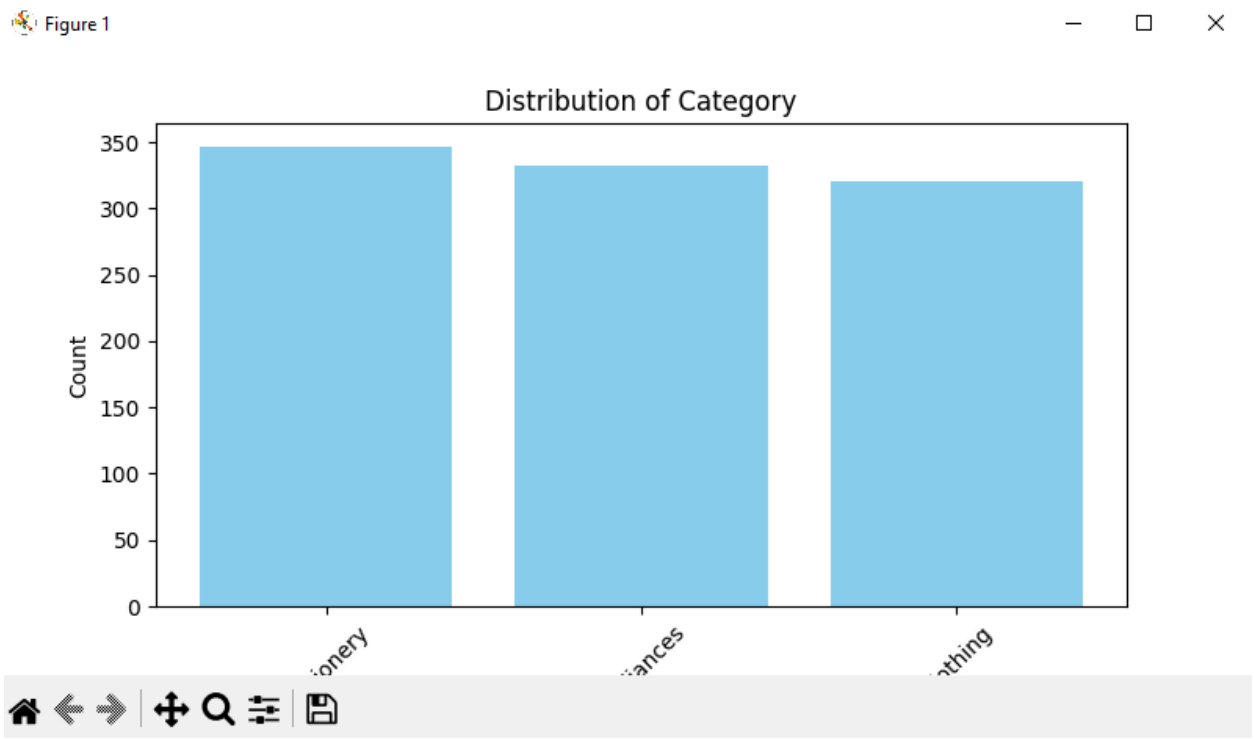


Figure 1
Distribution of Category
Count
350
300
250
200
150
100
50
0

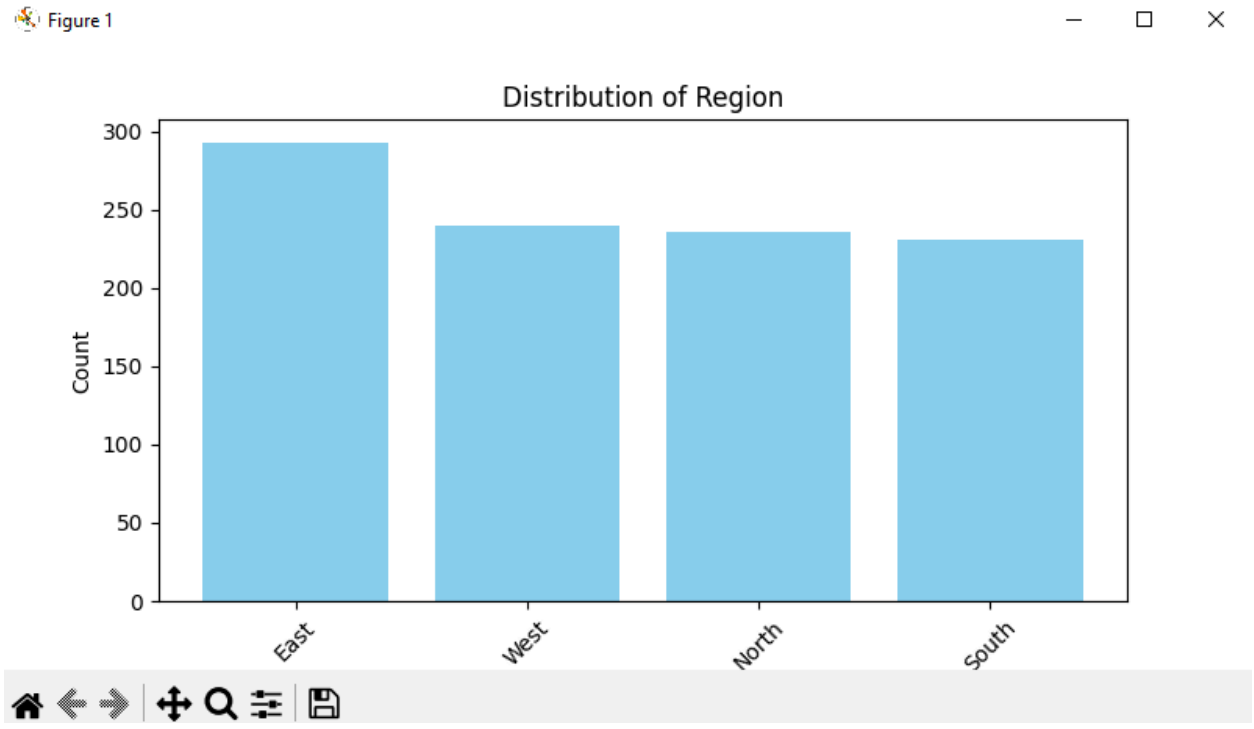


Figure 1
Distribution of Region
Count
300
250
200
150
100
50
0
East
West
North
South

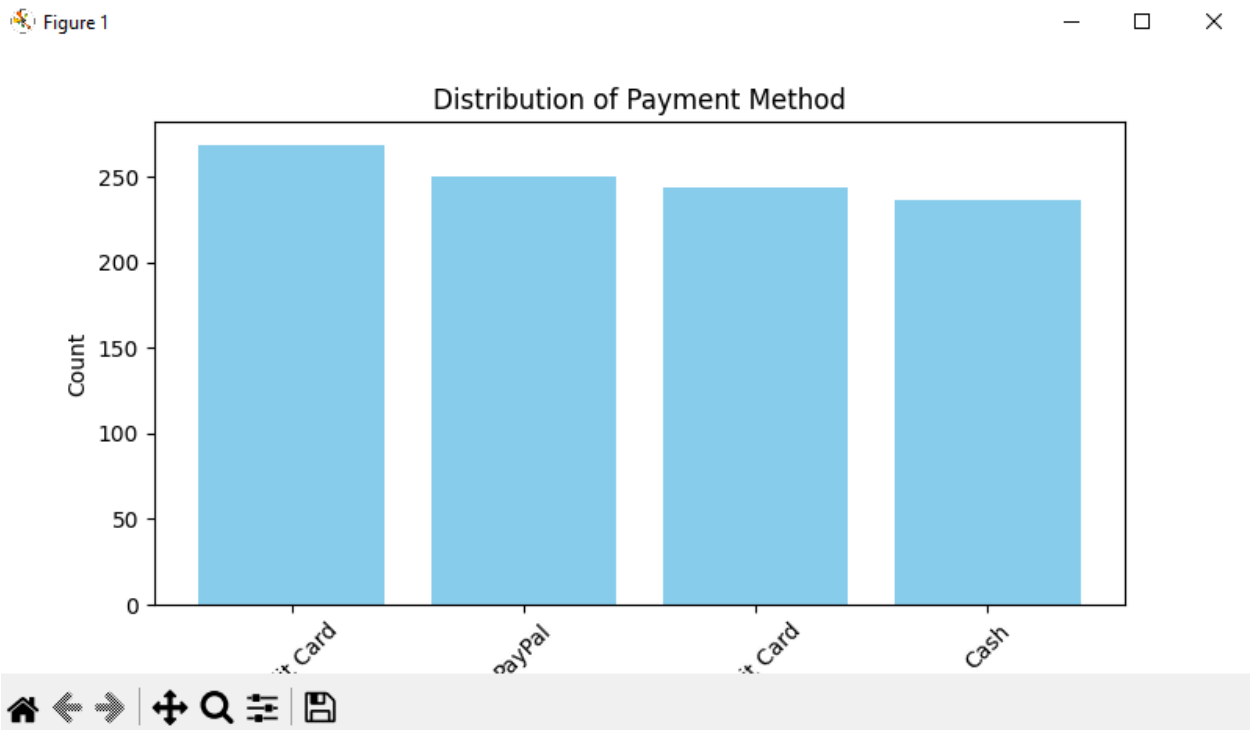


Figure 1
Distribution of Payment Method
Count
250
200
150
100
50
0
Cash

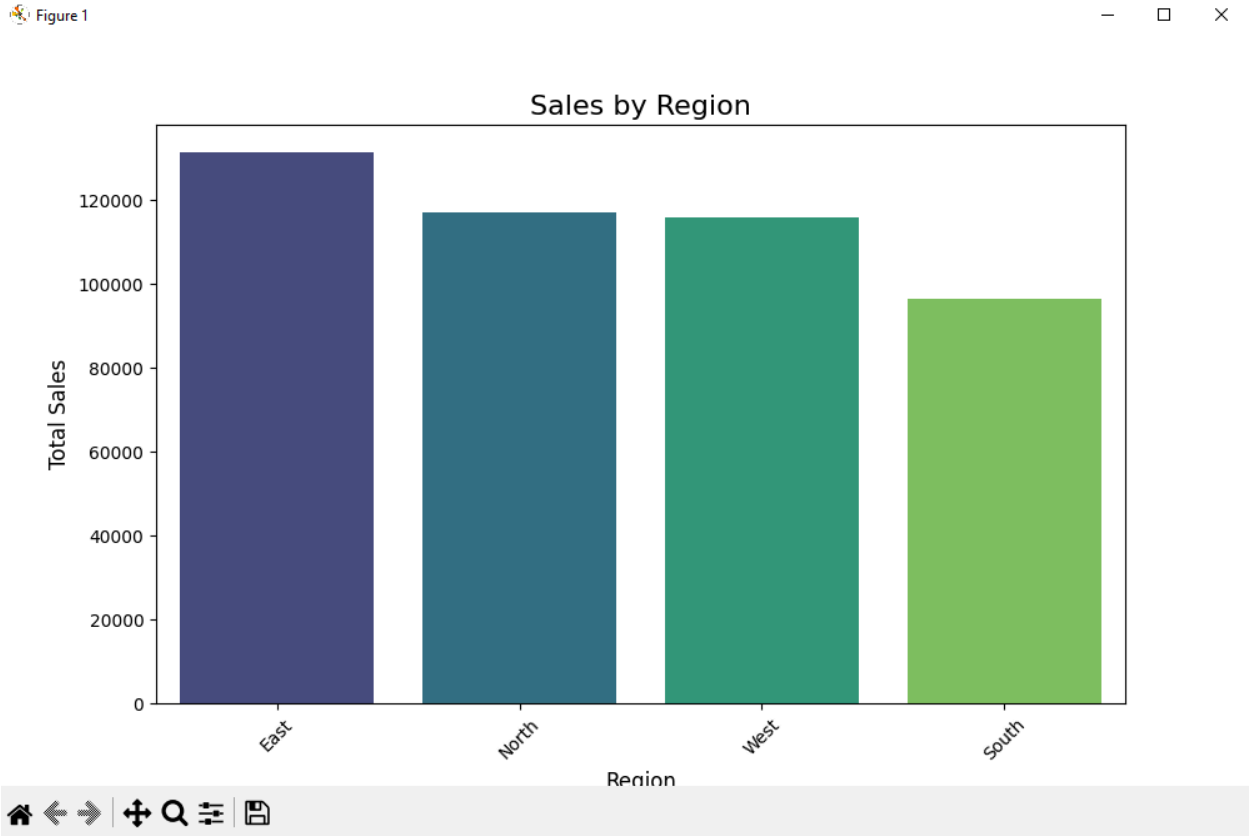


Figure 1
Sales by Region
Total Sales
120000
100000
80000
60000
40000
20000
0
East
North
West
South
Region

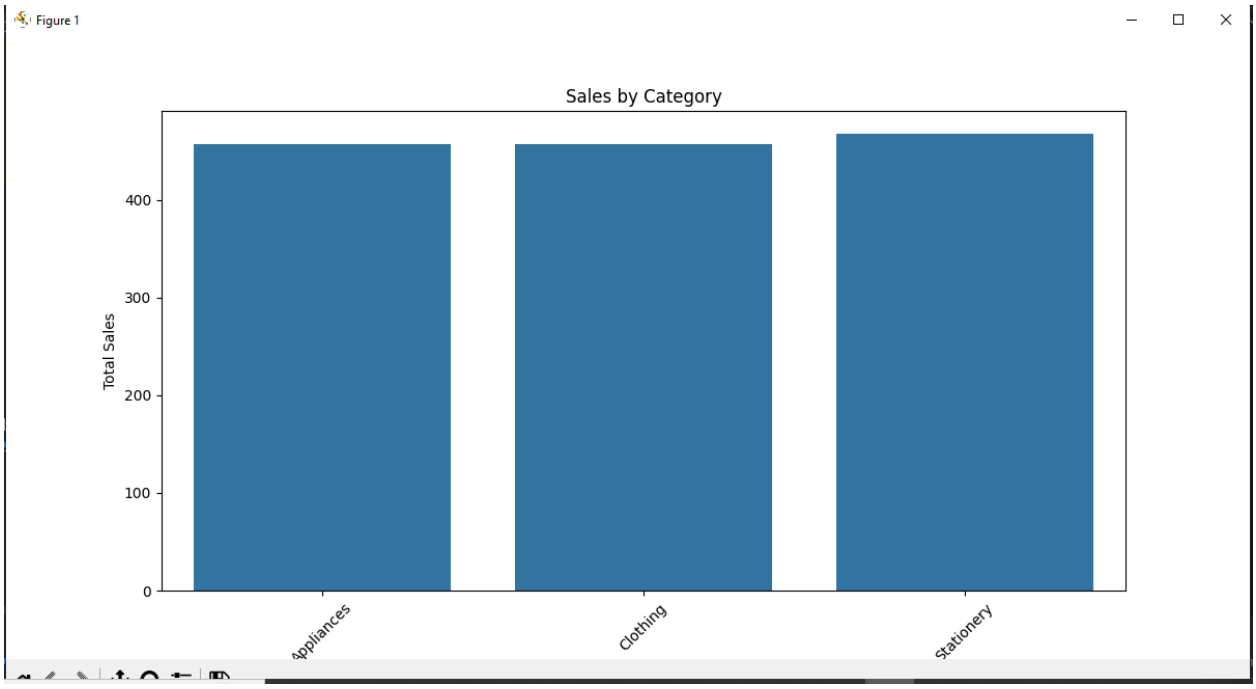


Figure 1
Sales by Category
Total Sales
0
100
200
300
400
Clothing
Stationery

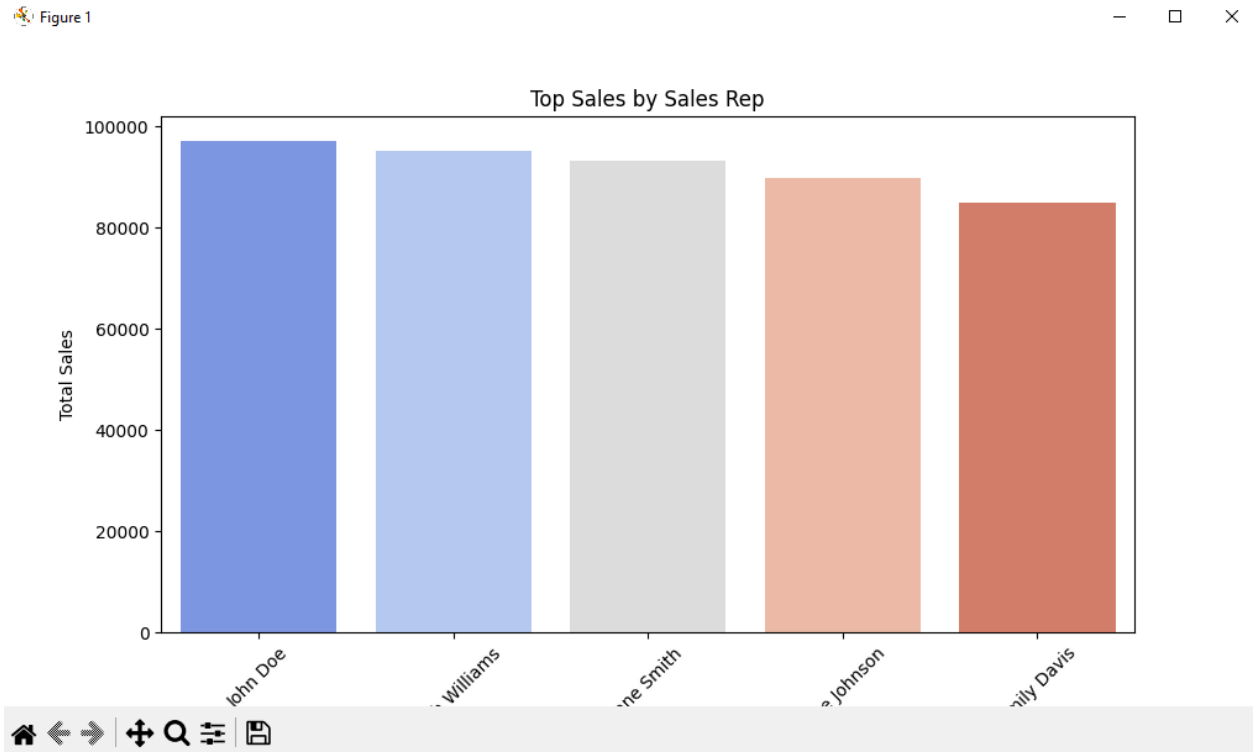


Figure 1
Top Sales by Sales Rep
Total Sales
0
20000
40000
60000
80000
100000
John Doe

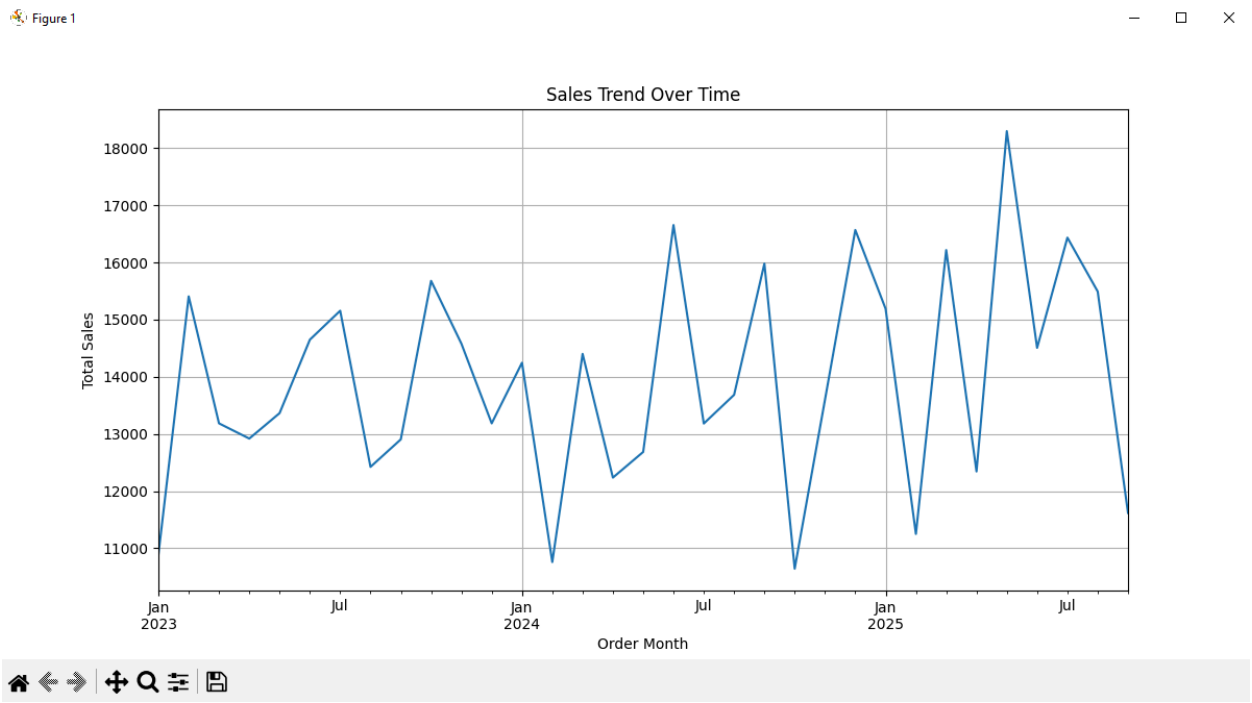


Figure 1
Sales Trend Over Time
Total Sales
18000
17000
16000
15000
14000
13000
12000
11000
Jan
2023
Jul
Jan
2024
Jul
Jan
2025
Jul
Order Month