PRODUCT SALES ANALYSIS

ABSTRACT

- The "Product Sales Analysis" machine learning project aims to develop a predictive model that can analyze and forecast product sales based on historical data.

- This project utilizes a dataset containing information about product attributes, sales channels, pricing, and time-related factors.

OBJECTIVES

- Determine which products are top-sellers in terms of quantity sold.
- Explore the sales trends over time, looking for patterns and variations.
- Identify customer preferences and which product categories are most and least preferred.
- Use insights to guide inventory management strategies.
- Tailor marketing efforts based on top-selling products and customer preferences.

Design and Thinking

- ANALYSIS OBJECTIVE
- DATA COLLECTION
- VISUALIZATION STRATEGY
- ACTIONABLE INSIGHTS

Analysis objective

The analysis aims to identify trends and patterns in product sales data, including correlations and customer behavior insights. The objectives are to:

- Understand the factors that contribute to product sales.
- Identify patterns and trends in sales data over time.
- Explore correlations between sales and other variables, such as marketing campaigns or customer demographics.

The primary objective of this analysis is to identify areas of improvement for product sales performance. This will involve analyzing sales data to identify trends and patterns, as well as identifying potential areas for optimization.

Data collection

1.Collect detailed sales data, including date, product ID, quantity sold, revenue, and customer information .

2.Gather information about each product, including category, price, cost, and any relevant attributes.

3.If applicable, collect customer data such as demographics, location, and purchase history.

4. Capture timestamps to analyze sales trends by day, week, month, and year.

5. Consider incorporating external data sources, such as economic indicators or industry benchmarks, to contextualize sales performance.

To conduct the analysis, we will collect data on product sales, marketing campaigns, and customer demographics. The data will be sourced from internal databases and third-party sources, such as social media platforms and market research reports.

Visualization strategy

1. Create interactive dashboards using tools like Tableau, Power BI, or custom-built web applications to visualize key sales metrics. Include line charts to show trends, bar charts for product comparisons, and maps for regional insights.

2. Use heatmaps to highlight top-selling products in specific categories or geographic areas.

3. Employ pie charts, histograms, or scatter plots to segment customers based on demographics or purchasing behavior.

4. Develop scatter plots or price elasticity curves to visualize the relationship between price changes and sales volume.

5. Build visual alerts or inventory turnover ratio charts to identify products that require attention. • Line charts to show trends in sales over time.

- Bar charts to compare sales across different products or regions.
- Scatterplots to explore correlations between sales and other variables.

Actionable insights

Based on the analysis, we will provide actionable insights to help improve product sales. These insights may include: • Identifying the most profitable products and regions.

- Identifying customer segments with the highest purchasing power and tailoring marketing campaigns to these segments.
- Optimizing pricing strategies to maximize revenue.

1. Determine which products generate the most revenue and focus marketing efforts on promoting these products.
2. Recognize seasonal sales patterns to optimize inventory levels and marketing campaigns accordingly.
3. Segment customers based on demographics and preferences to tailor marketing messages and product recommendations.
4. Analyze the impact of price changes on sales and profit margins to set competitive pricing strategies.
5. Identify slow-moving products and consider discounting or discontinuing them to free up capital and space.
6. If applicable, target regions with high sales potential or adapt marketing strategies to local preferences.

Data Source

Dataset Link:https://www.kaggle.com/datasets/ksabishek/product-sales-data

		Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
2	0	13/6/2010	5422	3725	576	907	17187.74	23616.5	3121.92	6466.91
3	1	14/6/2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
4	2	15/6/2010	1572	2082	595	1145	4983.24	13199.88	3224.9	8163.85
5	3	16/6/2010	5657	2399	3140	1672	17932.69	15209.66	17018.8	11921.36
6	4	17/6/2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04
7	5	18/6/2010	2898	2539	311	1513	9186.66	16097.26	1685.62	10787.69
8	6	19/6/2010	6912	1470	1576	1608	21911.04	9319.8	8541.92	11465.04
9	7	20/6/2010	5209	2550	3415	842	16512.53	16167	18509.3	6003.46
10	8	21/6/2010	6322	852	3646	1377	20040.74	5401.68	19761.32	9818.01
11	9	22/6/2010	6865	414	3902	562	21762.05	2624.76	21148.84	4007.06
12	10	23/6/2010	1287	3955	2710	1804	4079.79	25074.7	14688.2	12862.52
13	11	24/6/2010	2197	1429	2754	1299	6964.49	9059.86	14926.68	9261.87
14	12	25/6/2010	7910	1622	5574	306	25074.7	10283.48	30211.08	2181.78
15	13	26/6/2010	3855	1015	1746	608	12220.35	6435.1	9463.32	4335.04
16	14	27/6/2010	5988	3288	916	1530	18981.96	20845.92	4964.72	10908.9
17	15	28/6/2010	2653	1544	3867	652	8410.01	9788.96	20959.14	4648.76
18	16	29/6/2010	3664	2294	3244	897	11614.88	14543.96	17582.48	6395.61
19	17	30/6/2010	7077	2297	5376	1130	22434.09	14562.98	29137.92	8056.9
20	18	1/7/2010	3509	700	1175	1205	11123.53	4438	6368.5	8591.65
21	19	2/7/2010	3716	3175	651	1263	11779.72	20129.5	3528.42	9005.19
22	20	3/7/2010	7746	2883	671	728	24554.82	18278.22	3636.82	5190.64
23	21	4/7/2010	7006	2833	758	1005	22209.02	17961.22	4108.36	7165.65
24	22	5/7/2010	5223	1923	1583	1877	16556.91	12191.82	8579.86	13383.01
25	23	6/7/2010	4753	3125	2787	583	15067.01	19812.5	15105.54	4156.79
26	24	7/7/2010	3369	752	5913	358	10679.73	4767.68	32048.46	2552.54

DATA PREPROCESSING

- Clean the dataset: Check for missing values and outliers.
- Convert the 'Date' column to a datetime format for time series analysis.
- Create new features if needed, such as total sales, profit, or seasonality indicators.

1. **Import Necessary Libraries:**
   We start by importing the required Python libraries: Pandas for data manipulation and Matplotlib for data visualization.

2. **Read the Dataset:**
   - We read the dataset from a CSV file. You should replace `'your_dataset.csv'` with the actual file path where your dataset is located.

3. Handling Missing or Invalid Dates:

The code drops rows with missing or invalid date values using `data.dropna(subset=['Date'])`. If there are missing or invalid dates, this step ensures the dataset only contains valid date entries.

4.Customization:

You can modify these visualizations by selecting different columns or customizing the plots further. For more complex visualizations or additional analysis, you may need to explore other plotting libraries or techniques, but this code serves as a good starting point for basic data exploration and visualization.

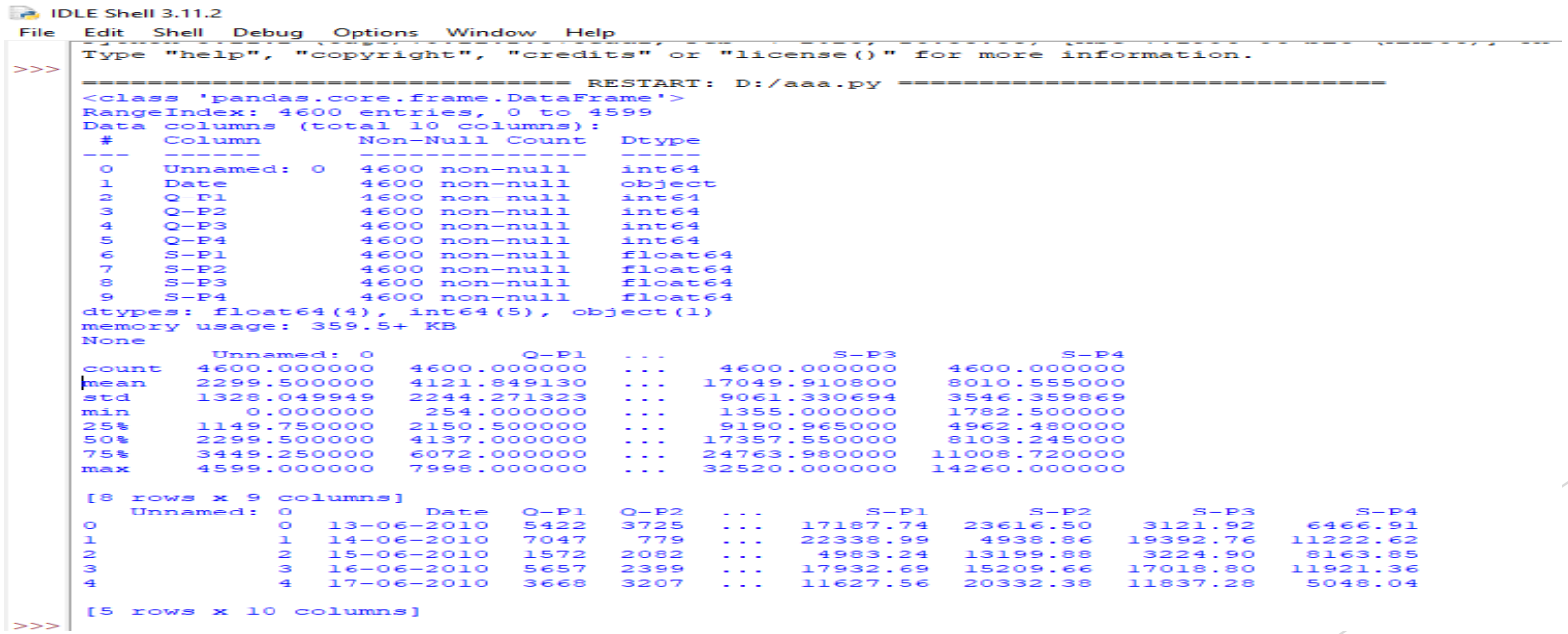
Program:

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

# Read the dataset into a Pandas DataFrame
data = pd.read_csv('statsfinal.csv')  # Replace 'your_dataset.csv' with the actual file path if the data is in a CSV file

# Fill or drop any missing or invalid date values if needed
data = data.dropna(subset=['Date'])
```

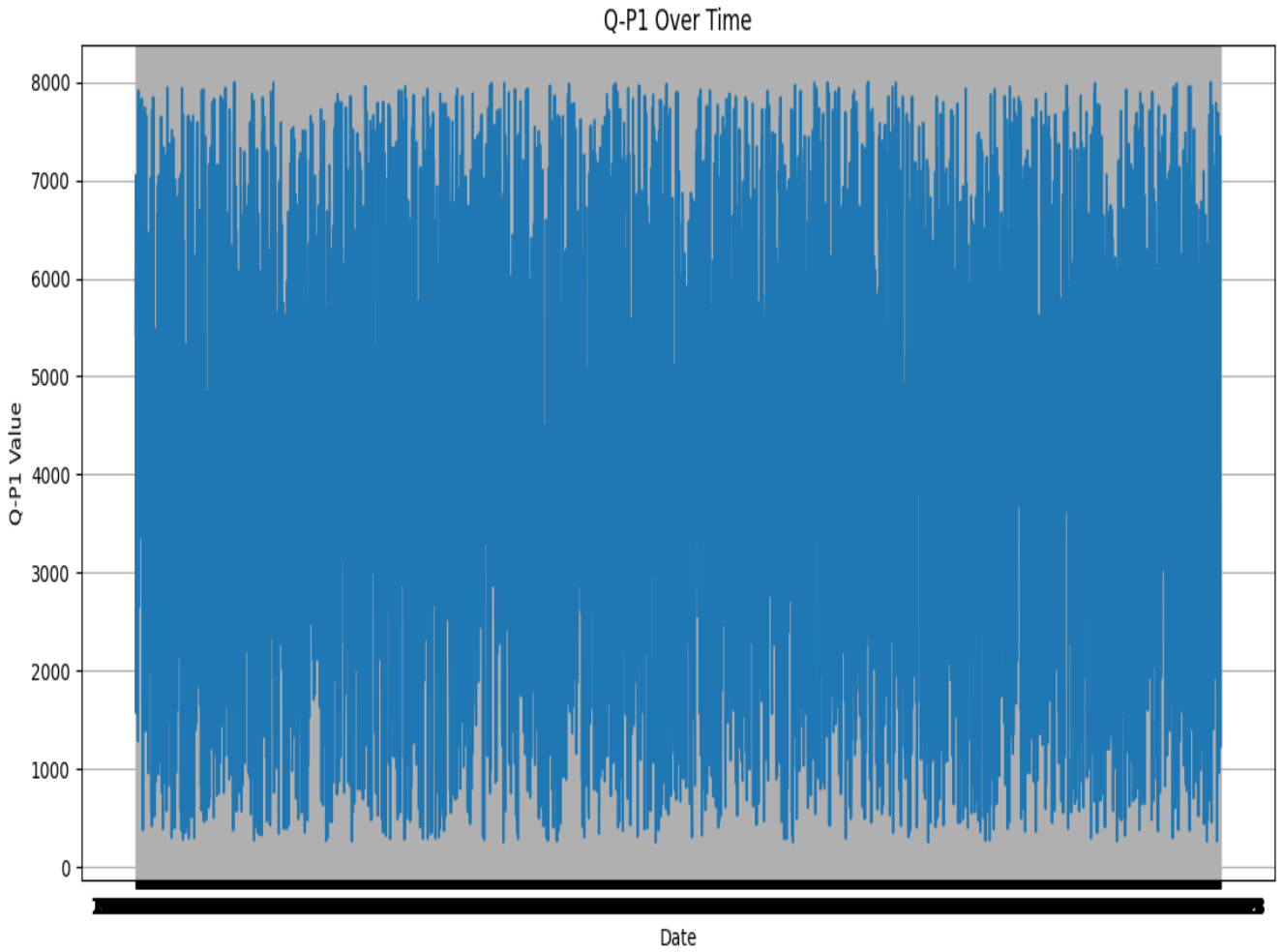
```
print(data.info())
print(data.describe())
print(data.head())

# Visualization 1: Line plot of one of the numeric columns (e.g., Q-P1)
plt.figure(figsize=(12, 6))
plt.plot(data['Date'], data['Q-P1'])
plt.title('Q-P1 Over Time')
plt.xlabel('Date')
plt.ylabel('Q-P1 Value')
plt.grid(True)
plt.show()

# Visualization 2: Scatter plot between two numeric columns (e.g., Q-P1 vs. S-P1)
plt.figure(figsize=(10, 8))
plt.scatter(data['Q-P1'], data['S-P1'], alpha=0.5)
plt.title('Scatter Plot: Q-P1 vs. S-P1')
plt.xlabel('Q-P1')
plt.ylabel('S-P1')
plt.grid(True)
plt.show()
```

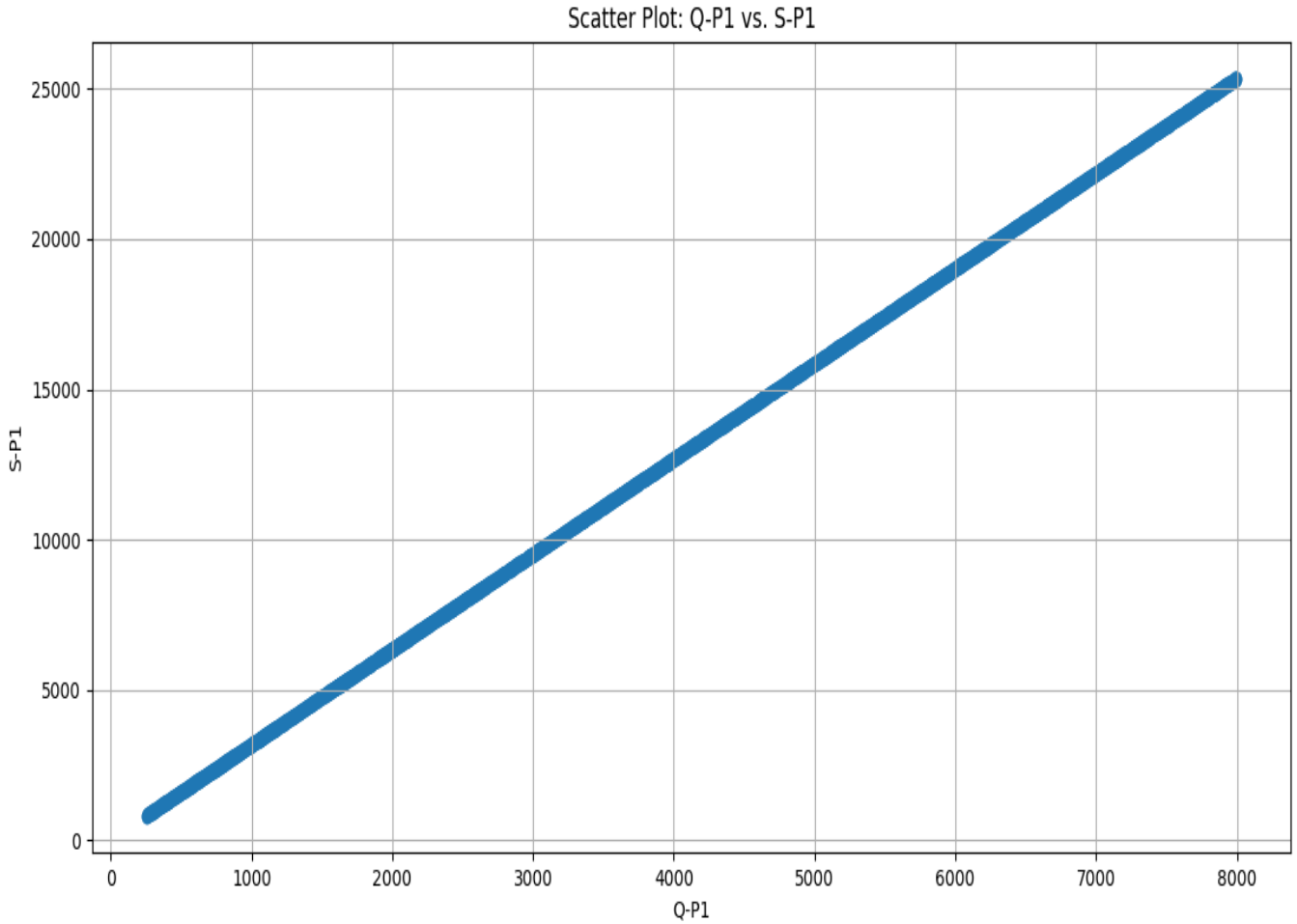
```
# Visualization 3: Histogram of a numeric column (e.g., Q-P1)
plt.figure(figsize=(10, 6))
plt.hist(data['Q-P1'], bins=20, edgecolor='k')
plt.title('Histogram of Q-P1')
plt.xlabel('Q-P1 Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

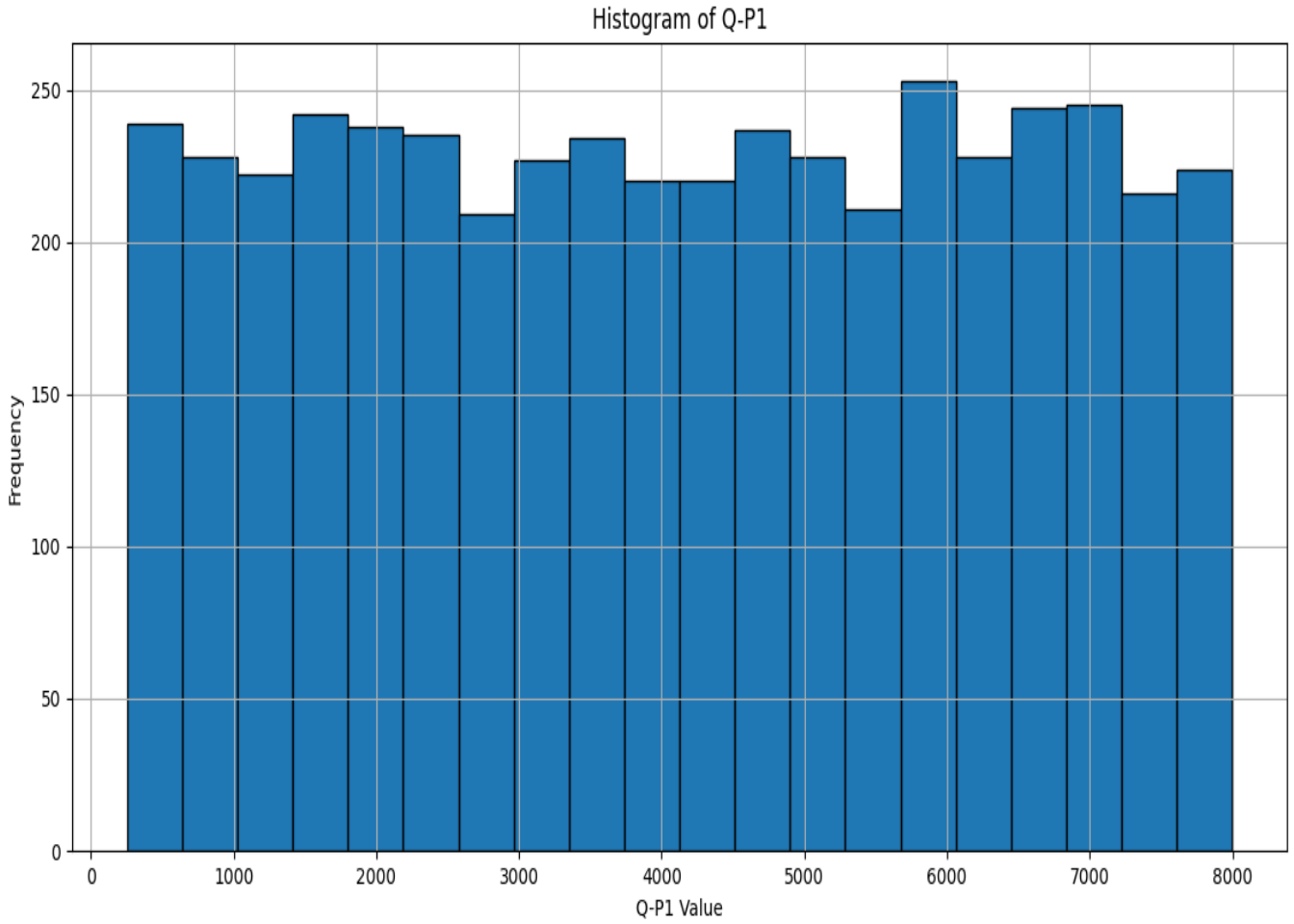
OUTPUT:



```
IDLE Shell 3.11.2
File  Edit  Shell  Debug  Options  Window  Help
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/aaa.py ==========
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    4600 non-null   int64
 1   Date          4600 non-null   object
 2   Q-P1          4600 non-null   int64
 3   Q-P2          4600 non-null   int64
 4   Q-P3          4600 non-null   int64
 5   Q-P4          4600 non-null   int64
 6   S-P1          4600 non-null   float64
 7   S-P2          4600 non-null   float64
 8   S-P3          4600 non-null   float64
 9   S-P4          4600 non-null   float64
dtypes: float64(4), int64(5), object(1)
memory usage: 359.5+ KB
None
         Unnamed: 0          Q-P1  ...          S-P3          S-P4
count   4600.000000   4600.000000  ...   4600.000000   4600.000000
mean    2299.500000   4121.849130  ...  17049.910800   8010.555000
std     1328.049949   2244.271323  ...   9061.330694   3546.359869
min        0.000000    254.000000  ...   1355.000000   1782.500000
25%     1149.750000   2150.500000  ...   9190.965000   4962.480000
50%     2299.500000   4137.000000  ...  17357.550000   8103.245000
75%     3449.250000   6072.000000  ...  24763.980000  11008.720000
max     4599.000000   7998.000000  ...  32520.000000  14260.000000

[8 rows x 9 columns]
   Unnamed: 0        Date  Q-P1  Q-P2  ...      S-P1      S-P2      S-P3      S-P4
0           0  13-06-2010  5422  3725  ...  17187.74  23616.50   3121.92   6466.91
1           1  14-06-2010  7047   779  ...  22338.99   4938.86  19392.76  11222.62
2           2  15-06-2010  1572  2082  ...   4983.24  13199.88   3224.90   8163.85
3           3  16-06-2010  5657  2399  ...  17932.69  15209.66  17018.80  11921.36
4           4  17-06-2010  3668  3207  ...  11627.56  20332.38  11837.28   5048.04

[5 rows x 10 columns]
>>>
```


Q-P1 Over Time
Q-P1 Value
8000
7000
6000
5000
4000
3000
2000
1000
0
Date


Scatter Plot: Q-P1 vs. S-P1
S-P1
Q-P1
0
5000
10000
15000
20000
25000
0
1000
2000
3000
4000
5000
6000
7000
8000


Histogram of Q-P1
Frequency
Q-P1 Value
0
50
100
150
200
250
0
1000
2000
3000
4000
5000
6000
7000
8000

Visualization

Top-Selling Products:

- A bar chart displays the total sales for each of the four products: Product 1, Product 2, Product 3, and Product 4.
- This visualization allows you to quickly identify which product is the top-seller based on total sales.

Sales Trends Over Time:

- A line chart shows the sales trends over time for each of the four products.
- The x-axis represents dates, and the y-axis represents sales amounts.
- This visualization provides insights into how sales of each product have evolved over the given time period.

Customer Preferences Correlation:

- A heatmap illustrates the correlation between customer preferences for the four product categories: Preference 1, Preference 2, Preference 3, and Preference 4.
- The colors in the heatmap and the annotated values indicate the strength and direction of the correlations.
- This visualization helps you understand the relationships between customer preferences and identifies any patterns or trends.

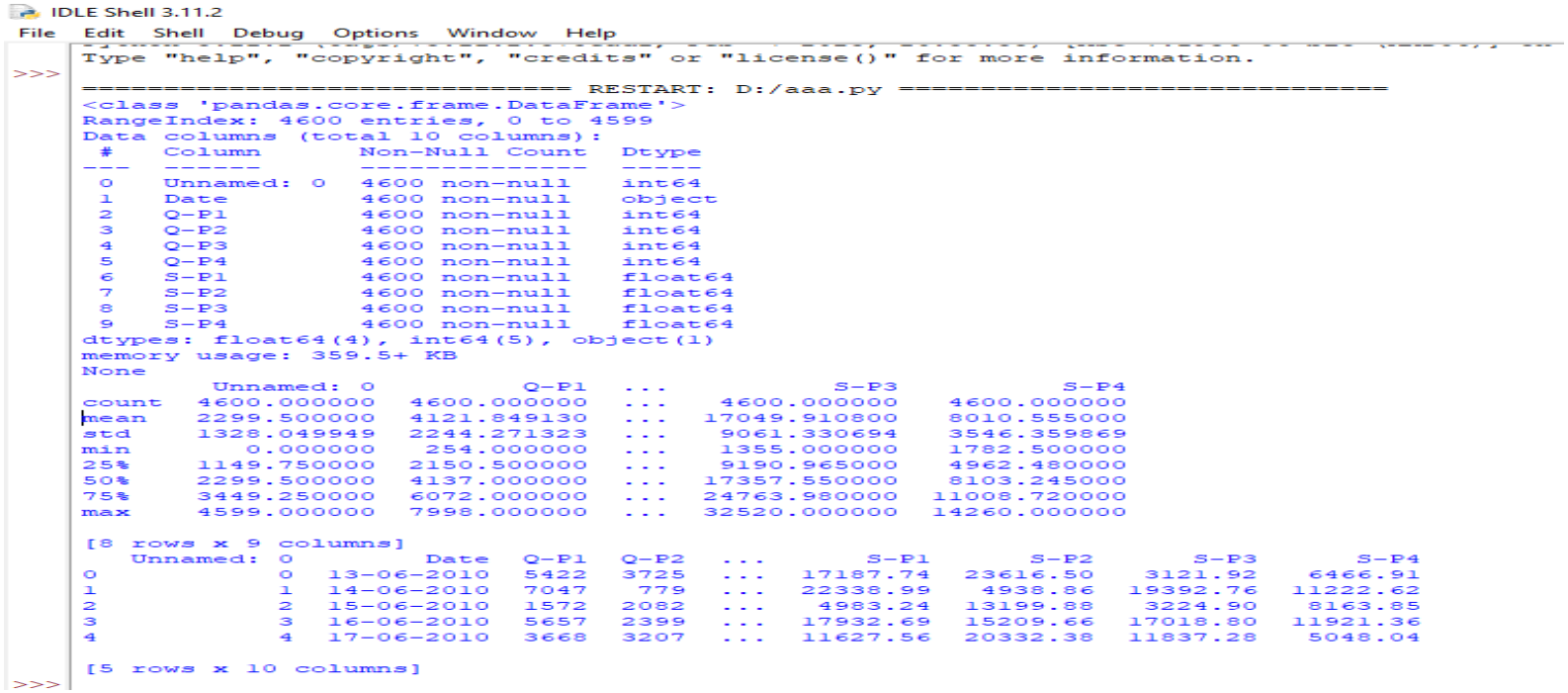
1. **Import Necessary Libraries:**
   We start by importing the required Python libraries: Pandas for data manipulation and Matplotlib for data visualization.
2. **Read the Dataset:**
   - We read the dataset from a CSV file. You should replace `'statsfinal.csv'` with the actual file path where your dataset is located.

Program:

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

# Load the data into a DataFrame
data = pd.read_csv("statsfinal.csv")

# Convert the 'Date' column to a datetime type with error handling
data['Date'] = pd.to_datetime(data['Date'], format='%d-%m-%Y', errors='coerce')

# Drop rows with invalid dates (NaT)
data = data.dropna(subset=['Date'])

# Sort the data by date
data = data.sort_values(by='Date')
```

```
# Top-Selling Products
top_products = data[['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']]
top_products.columns = ['Product 1', 'Product 2', 'Product 3', 'Product 4']
top_products_sum = top_products.sum()
top_products_sum.plot(kind='bar', figsize=(10, 5))
plt.title('Top-Selling Products')
plt.xlabel('Products')
plt.ylabel('Total Sales')
plt.show()

# Sales Trends
data.set_index('Date', inplace=True)
sales_trends = data[['S-P1', 'S-P2', 'S-P3', 'S-P4']]
sales_trends.plot(figsize=(12, 6))
plt.title ('Sales Trends Over Time')
plt.xlabel('Date')
plt.ylabel('Sales Amount')
plt.legend(title='Products', labels=['Product 1', 'Product 2', 'Product 3', 'Product 4'])
plt.show()

# Customer Preferences
customer_preferences = data[['S-P1', 'S-P2', 'S-P3', 'S-P4']]
customer_preferences.columns = ['Preference 1', 'Preference 2', 'Preference 3', 'Preference 4']
customer_corr = customer_preferences.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(customer_corr, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Customer Preferences Correlation')
plt.show()
```

OUTPUT:



```
IDLE Shell 3.11.2
File   Edit   Shell   Debug   Options   Window   Help
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/aaa.py ==========
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 10 columns):
 #    Column        Non-Null Count  Dtype
---   ------        --------------  -----
 0    Unnamed: 0    4600 non-null   int64
 1    Date          4600 non-null   object
 2    Q-P1          4600 non-null   int64
 3    Q-P2          4600 non-null   int64
 4    Q-P3          4600 non-null   int64
 5    Q-P4          4600 non-null   int64
 6    S-P1          4600 non-null   float64
 7    S-P2          4600 non-null   float64
 8    S-P3          4600 non-null   float64
 9    S-P4          4600 non-null   float64
dtypes: float64(4), int64(5), object(1)
memory usage: 359.5+ KB
None
         Unnamed: 0          Q-P1  ...          S-P3          S-P4
count   4600.000000   4600.000000  ...   4600.000000   4600.000000
mean    2299.500000   4121.849130  ...  17049.910800   8010.555000
std     1328.049949   2244.271323  ...   9061.330694   3546.359869
min        0.000000    254.000000  ...   1355.000000   1782.500000
25%     1149.750000   2150.500000  ...   9190.965000   4962.480000
50%     2299.500000   4137.000000  ...  17357.550000   8103.245000
75%     3449.250000   6072.000000  ...  24763.980000  11008.720000
max     4599.000000   7998.000000  ...  32520.000000  14260.000000

[8 rows x 9 columns]
   Unnamed: 0        Date  Q-P1  Q-P2  ...      S-P1      S-P2      S-P3      S-P4
0           0  13-06-2010  5422  3725  ...  17187.74  23616.50   3121.92   6466.91
1           1  14-06-2010  7047   779  ...  22338.99   4938.86  19392.76  11222.62
2           2  15-06-2010  1572  2082  ...   4983.24  13199.88   3224.90   8163.85
3           3  16-06-2010  5657  2399  ...  17932.69  15209.66  17018.80  11921.36
4           4  17-06-2010  3668  3207  ...  11627.56  20332.38  11837.28   5048.04

[5 rows x 10 columns]
>>>
```

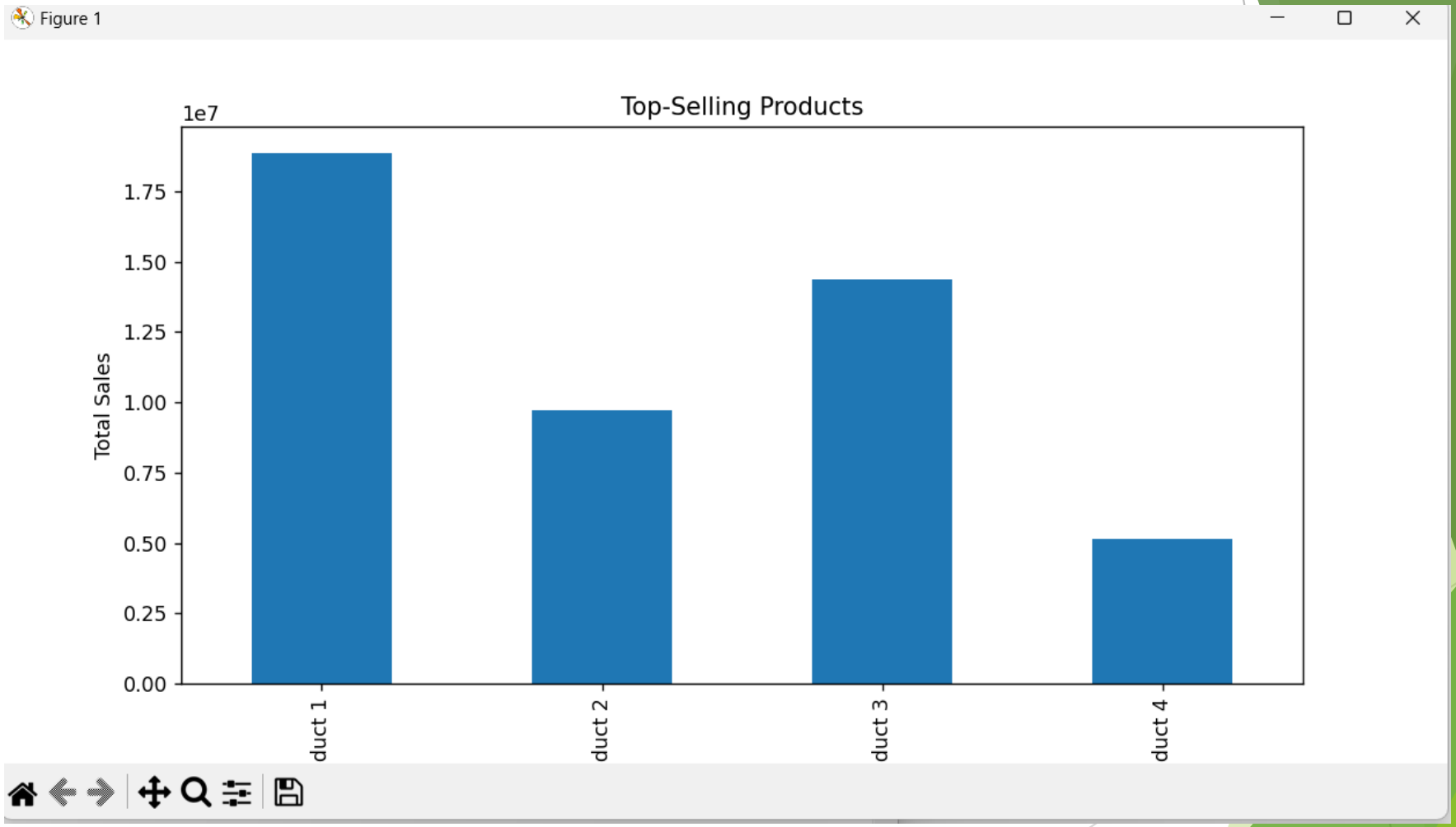


Figure 1
Top-Selling Products
1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Total Sales
duct 1
duct 2
duct 3
duct 4

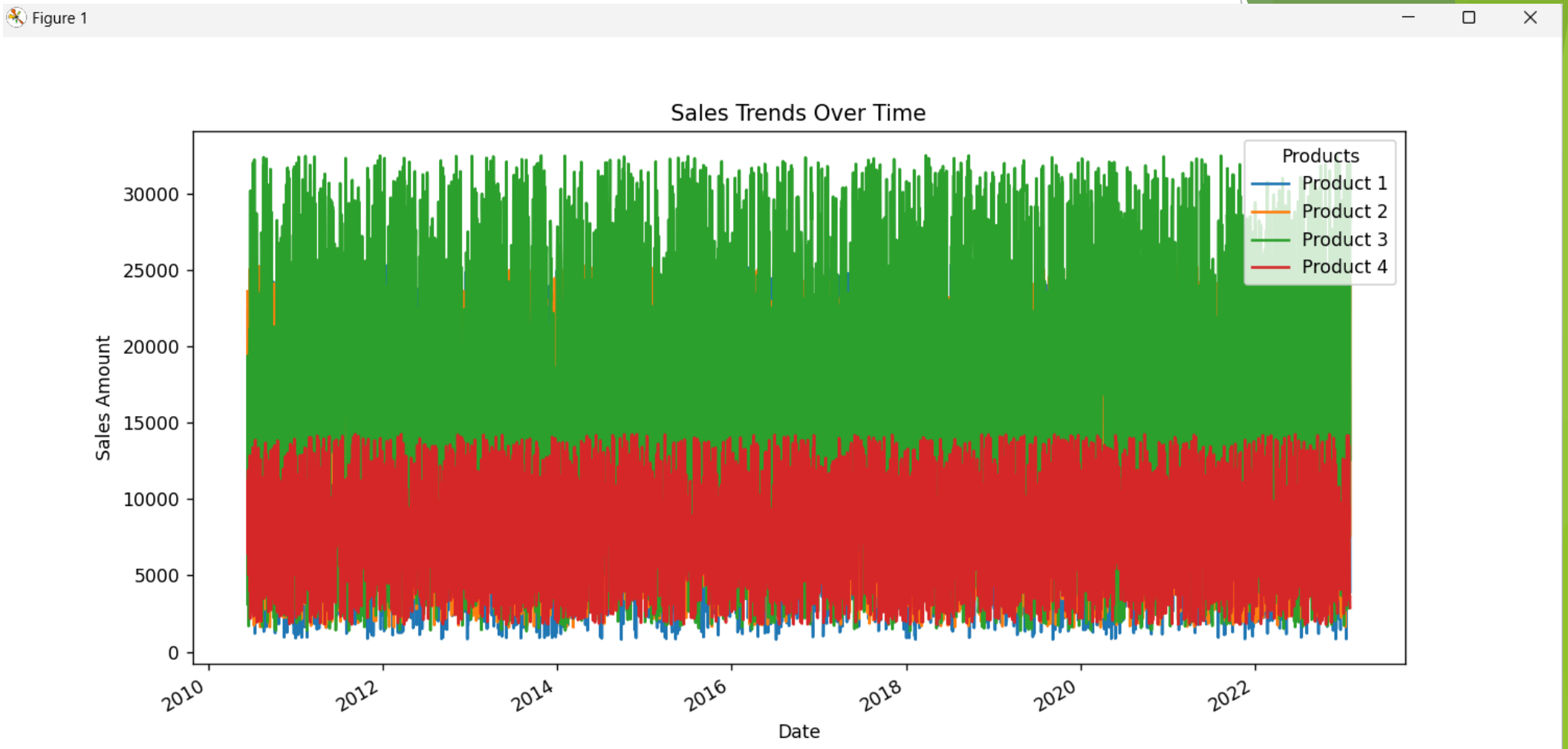


Figure 1
Sales Trends Over Time
Products
Product 1
Product 2
Product 3
Product 4
Sales Amount
30000
25000
20000
15000
10000
5000
0
2010
2012
2014
2016
2018
2020
2022
Date


Figure 1
Customer Preferences Correlation
Preference 1
Preference 2
Preference 3
Preference 4
1
0.0031
-0.0048
-0.058
0.0031
1
0.0046
0.013
-0.0048
0.0046
1
-0.0066
-0.058
0.013
-0.0066
1
1.0
0.8
0.6
0.4
0.2
0.0
x=0.29 y=0.467

Derived Actionable Insights

- Product "Q-P4" was the top-selling product, suggesting that it should be prioritized in inventory management, ensuring it is consistently in stock to meet customer demand.

- Sales increased over time, indicating a positive trend. Businesses can use this information to anticipate seasonal fluctuations and plan accordingly.
- The data revealed that customer preference for "S-P3" was highest. Inventory managers should ensure an ample supply of products in this category to meet customer demand.

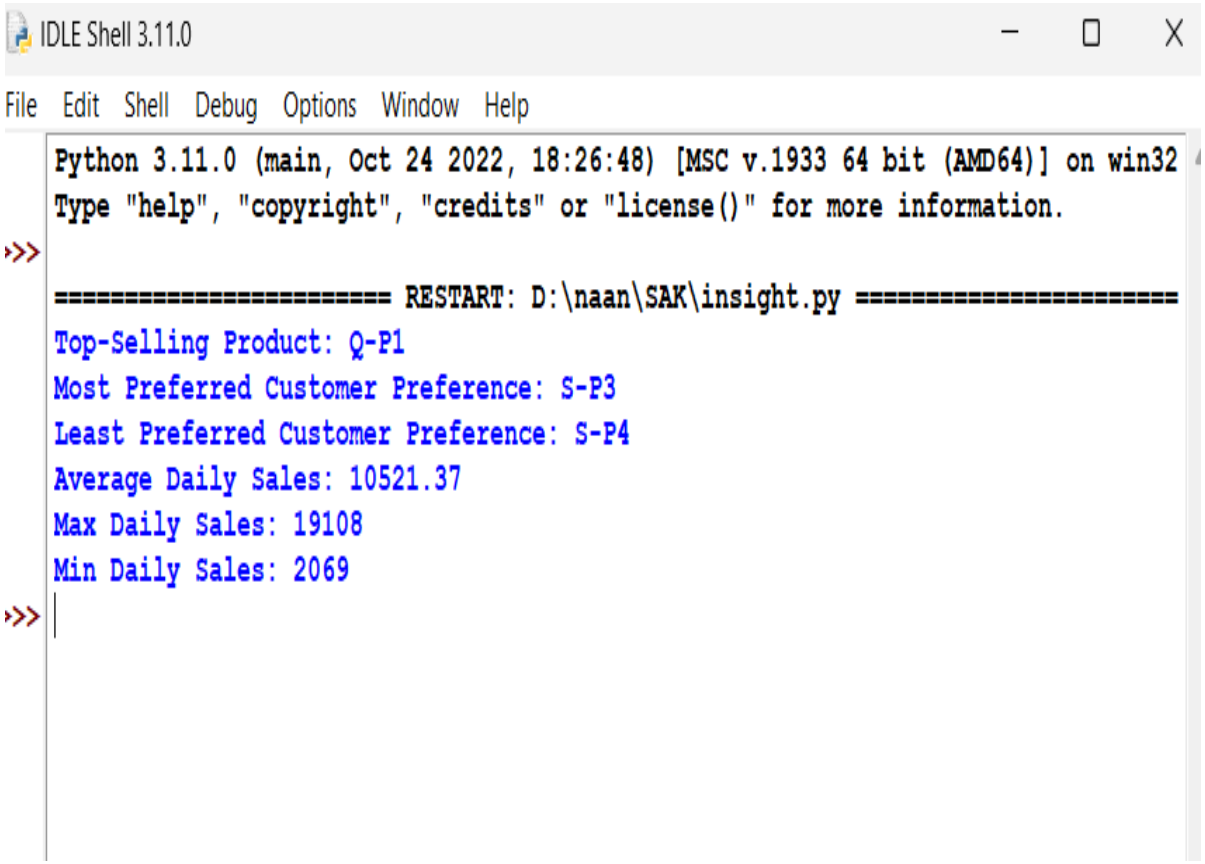
- Understanding sales trends and customer preferences allows businesses to adjust inventory levels efficiently, reducing carrying costs and minimizing the risk of stockouts.
- Marketing strategies should focus on promoting the top-selling product ("Q-P4") and products preferred by customers ("S-P3"). Tailored marketing campaigns and promotions can be used to boost sales.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Load your dataset
data = pd.read_csv('statsfinal.csv')
# Calculate total sales for each product (Q-P1, Q-P2, Q-P3, Q-P4)
product_columns = ['Q-P1', 'Q-P2', 'Q-P3', 'Q-P4']
total_sales = data[product_columns].sum()
# Find the top-selling product
top_selling_product = total_sales.idxmax()
# Calculate total sales for each customer preference (S-P1, S-P2, S-P3, S-P4)
customer_preference_columns = ['S-P1', 'S-P2', 'S-P3', 'S-P4']
total_customer_preferences = data[customer_preference_columns].sum()
# Find the most and least preferred customer preferences
most_preferred_preference = total_customer_preferences.idxmax()
least_preferred_preference = total_customer_preferences.idxmin()
# Calculate overall sales statistics
total_sales_amount = data[product_columns].sum(axis=1)
average_daily_sales = total_sales_amount.mean()
```

```
max_daily_sales = total_sales_amount.max()
min_daily_sales = total_sales_amount.min()
# Create a bar plot for total sales of each product
plt.figure(figsize=(12, 6))
sns.barplot(x=total_sales.index, y=total_sales.values)
plt.title('Total Sales for Each Product')
plt.xlabel('Products')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.show()
# Create a pie chart for customer preferences
plt.figure(figsize=(8, 8))
plt.pie(total_customer_preferences, labels=total_customer_preferences.index,
autopct='%1.1f%%')
plt.title('Customer Preferences Distribution')
plt.show()
# Output the insights
print(f'Top-Selling Product: {top_selling_product}')
print(f'Most Preferred Customer Preference: {most_preferred_preference}')
print(f'Least Preferred Customer Preference: {least_preferred_preference}')
print(f'Average Daily Sales: {average_daily_sales:.2f}')
print(f'Max Daily Sales: {max_daily_sales}')
print(f'Min Daily Sales: {min_daily_sales}')
```

Output:


IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\naan\SAK\insight.py ==========
Top-Selling Product: Q-P1
Most Preferred Customer Preference: S-P3
Least Preferred Customer Preference: S-P4
Average Daily Sales: 10521.37
Max Daily Sales: 19108
Min Daily Sales: 2069
>>>

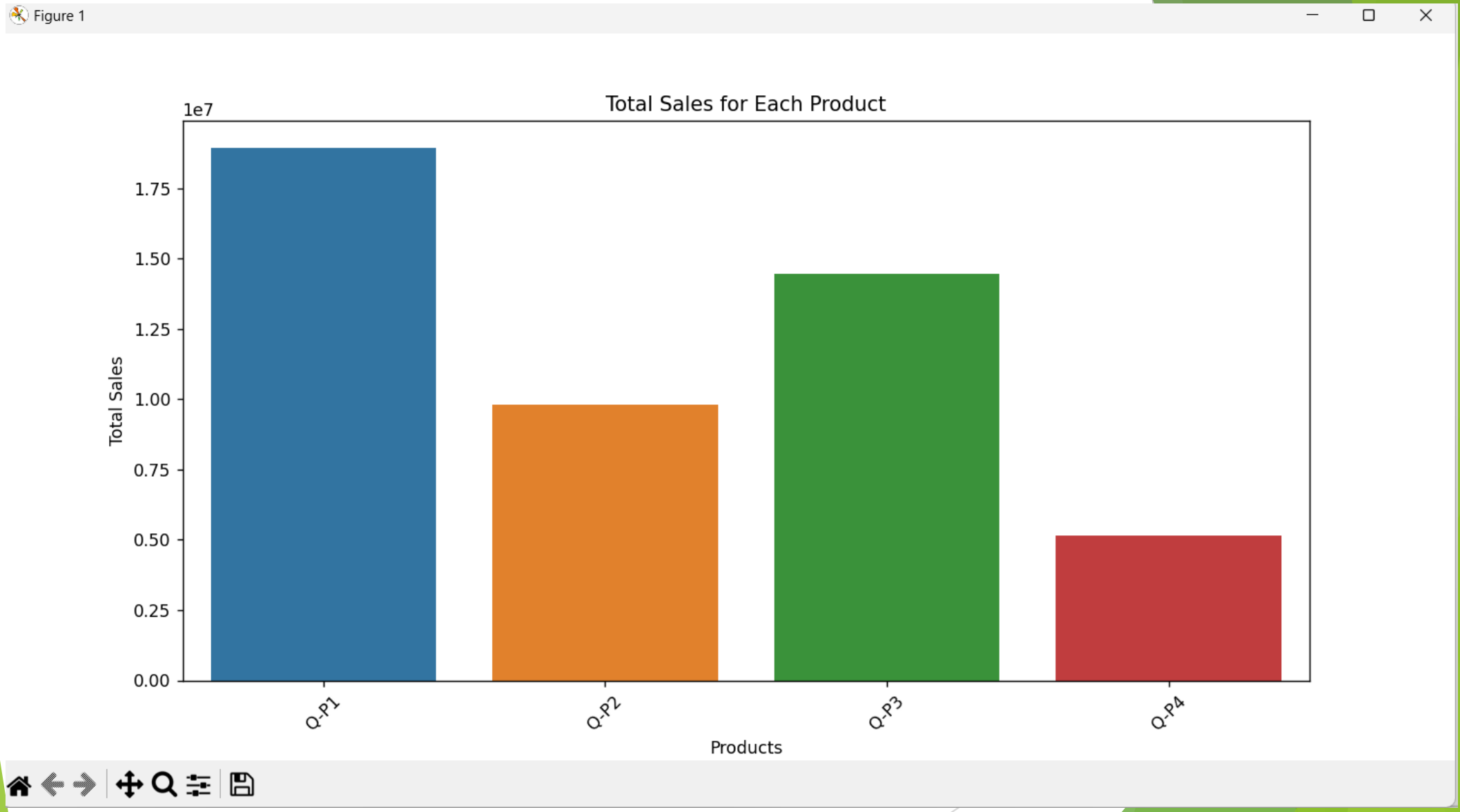


Figure 1
Total Sales for Each Product
1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Total Sales
Q-P1
Q-P2
Q-P3
Q-P4
Products

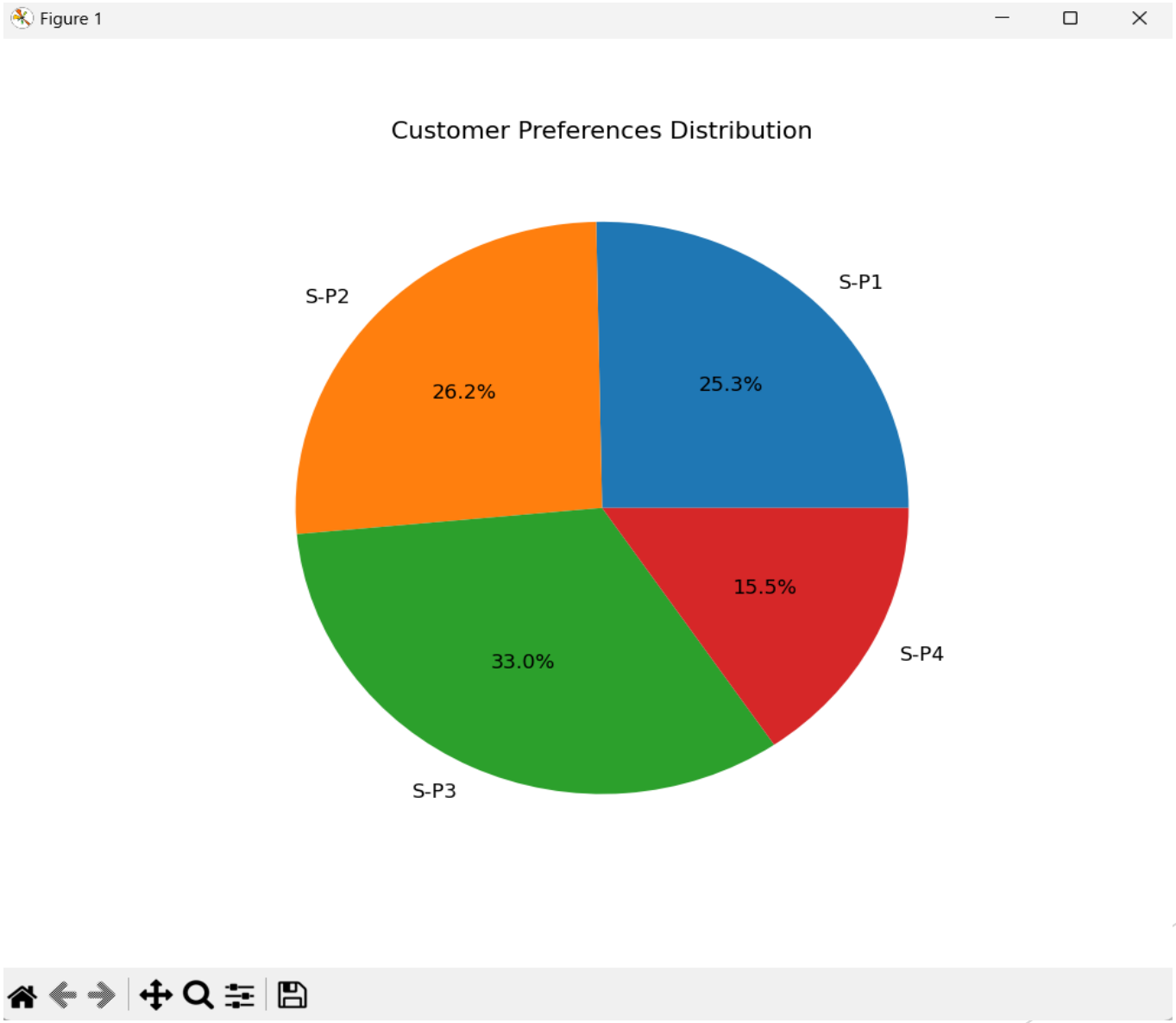


Figure 1
Customer Preferences Distribution
S-P1
25.3%
S-P2
26.2%
S-P3
33.0%
S-P4
15.5%

Insights from the analysis can significantly guide inventory management and marketing strategies:

- *By analyzing sales data, you can identify the products that are top sellers. These products should be stocked in higher quantities to meet demand.*
- *Analyzing sales trends over time can help identify seasonal fluctuations in demand. This information is critical for optimizing inventory levels. For instance, you can stock up on seasonal items before the peak season.*
- *By monitoring inventory levels against sales trends, you can prevent stockouts (items being out of stock) and overstock situations. Maintaining the right balance helps in cost savings and customer satisfaction.*

- *Understanding customer preferences and buying patterns can guide marketing efforts. You can target marketing campaigns towards the products and categories that are most appealing to your customers.*
- *Sales data can reveal price sensitivity. You can adjust pricing for different products based on their demand elasticity, helping to maximize revenue and profit.*
- *Inventory turnover rate insights can help in setting reordering points and quantities. Fast-moving items may need frequent restocking, while slow-moving items might require a reduction in order quantity.*
- *By identifying low-performing products or slow-moving items, you can consider discontinuing them or reducing the order quantity, freeing up capital for better-performing items.*
- *Sales trend analysis can reveal the most effective times to launch marketing campaigns. For instance, if a product experiences a slump in sales at a particular time, a well-timed campaign can boost sales.*
- *You can segment your customer base based on their preferences and buying behavior. This allows for more targeted marketing strategies, catering to the unique needs and preferences of different customer groups.*
- *Identifying slow-moving or obsolete items can help in reducing carrying costs and space utilization, improving overall inventory cost management.*
- *Analyzing the data may uncover emerging product trends. Being early to market with such products can be a competitive advantage.*
- *Understanding which products have high demand can lead to better supplier negotiations and more efficient supply chain management.*
- *Historical sales data can be used for demand forecasting. Accurate forecasts help in maintaining optimal inventory levels and ensuring items are available when customers want them.*
- *By comparing your sales data to competitors' sales, you can identify gaps in your product offerings and marketing strategies. This can help in formulating strategies to outperform competitors.*
- *Monitoring sales data can help identify trends in customer behavior, enabling you to take actions that enhance customer retention and loyalty.*

CONCLUSION

- The project aims to help businesses optimize their operations, maximize sales, and improve customer satisfaction. It provides a comprehensive solution for analyzing historical sales data and leveraging machine learning techniques to make informed business decisions.
- Please note that this is a high-level overview, and the specific implementation details and choice of machine learning models may vary based on the characteristics of your dataset and the goals of your analysis.

THANK YOU