Phase 5

**PRODUCT SALES ANALYSIS**

horizontal line

**TEAM MEMBERS:**

|  |  |
| --- | --- |
| DHIVYADHARSHINI S B ([**dhivyadharshu07@gmail.com**](mailto:dhivyadharshu07@gmail.com)) | 2021115029 |
| DHARINI  ([**dhariniathithakumar@gmail.com**](mailto:dhariniathithakumar@gmail.com)) | 2021115027 |
| DHAVASRI  ([**indradhavasri212@gmail.com**](mailto:indradhavasri212@gmail.com)) | 2021115028 |
| DILEEP  ([**dilleep18@gmail.com**](mailto:dilleep18@gmail.com)) |  |

# OBJECTIVE

The objective of the project is to develop a product sales analysis system using IBM Cognos. This system will help organizations analyze and gain insights into their product sales data to make informed decisions, improve sales strategies, and optimize their sales performance.

## DESIGN THINKING PROCESS

Design thinking is a human-centered approach to problem-solving and product development. It consists of several phases that are applied to the project of creating a product sales analysis system using IBM Cognos:

* Empathize:
  + Understand the needs and pain points of the end-users and stakeholders who will be using the product sales analysis system.
  + Conduct interviews, surveys, and observations to gather insights into their requirements and expectations.
* Define:
  + Clearly define the problem or challenge that the product sales analysis system aims to address.
  + Create user personas and use cases to establish a shared understanding of the system's purpose.
* Ideate:
  + Brainstorm potential solutions and features for the product sales analysis system.
  + Encourage creative thinking to come up with innovative ways to visualize and analyze sales data.
* Prototype:
  + Develop low-fidelity and high-fidelity prototypes of the user interface and system functionality.
  + Test these prototypes with stakeholders to gather feedback and make iterative improvements.
* Test:
  + Conduct usability testing to evaluate the effectiveness of the prototypes and gather feedback from users.
  + Refine the design and functionality based on the testing results.
* Implement:
  + Begin the development of the product sales analysis system using IBM Cognos.
  + Follow best practices for data modeling, report design, and dashboard creation within Cognos.
* Deploy:
  + Roll out the system to a small group of users or a pilot program to gather further feedback and identify any issues.
* Evaluate:
  + Continuously monitor the performance of the product sales analysis system in a real-world environment.
  + Collect feedback from users and stakeholders to identify areas for improvement.

## Development Phases:

The development of the product sales analysis system using IBM Cognos can be broken down into several phases:

* Requirements Gathering:
  + Define the specific requirements for the system, including data sources, key performance indicators (KPIs), and user access levels.
* Data Integration:
  + Integrate data from various sources, such as CRM systems, sales databases, and external data feeds, into IBM Cognos.
* Data Modeling:
  + Create a well-structured data model to support the reporting and analysis needs of the system.
  + Define relationships and hierarchies within the data.
* Report and Dashboard Design:
  + Design interactive reports and dashboards that allow users to analyze product sales data.
  + Use IBM Cognos tools to create visually appealing and informative data visualizations.
* Security and Access Control:
  + Implement security measures to ensure that only authorized users have access to specific data and functionality.
  + Define user roles and permissions.
* Testing and Quality Assurance:
  + Thoroughly test the system for functionality, data accuracy, and performance.
  + Address and resolve any issues or bugs that arise during testing.
* Deployment:
  + Deploy the system to the production environment, ensuring that it is accessible to the intended users.
* Training and Documentation:
  + Provide training to users and administrators on how to use the product sales analysis system effectively.
  + Create documentation for system usage and maintenance.
* Ongoing Support and Maintenance:
  + Provide ongoing support for the system, including troubleshooting, updates, and enhancements based on user feedback and evolving business needs.

By following a design thinking approach and breaking down the development into these phases, you can create an effective product sales analysis system using IBM Cognos that meets the needs of the organization and its users.

# Analysis Objectives:

The analysis objectives for the product sales analysis using IBM Cognos may include:

1. Sales Performance Evaluation: Assess and analyze the sales performance of products across different time periods, regions, and sales channels.

2. Customer Segmentation: Segment customers based on various criteria such as demographics, purchase history, and preferences to identify target markets.

3. Inventory Management: Optimize inventory levels based on historical sales data to reduce carrying costs and stockouts.

4. Price Optimization: Analyze pricing strategies and their impact on sales volume and revenue.

5. Trend Identification: Identify sales trends, seasonality, and anomalies in product sales data.

6. Sales Forecasting: Predict future sales based on historical data to aid in demand planning and inventory management.

# Data Collection Process:

Data collection is a critical step in the analysis process. Here's an overview of how data was collected for the product sales analysis:

1. Data Sources: Identify the relevant data sources, which might include sales databases, CRM systems, e-commerce platforms, and external data feeds.

2. Data Extraction: Extract the necessary data from these sources. This may involve SQL queries, data exports, or using ETL (Extract, Transform, Load) tools to prepare the data for analysis.

3. Data Cleaning: Cleanse the data by removing duplicates, correcting errors, and handling missing values to ensure data accuracy.

4. Data Integration: Combine data from different sources into a unified dataset for analysis. Ensure consistency in data formats and structures.

5. Data Transformation: Transform data as needed, such as aggregating sales by product, region, and time, and calculating relevant metrics like revenue, profit, and sales quantities.

6. Data Loading: Load the clean and transformed data into IBM Cognos or a suitable data warehousing solution for analysis.

# Data Visualization Using IBM Cognos:

IBM Cognos provides robust capabilities for data visualization and reporting. Here's how data is visualized within Cognos:

1. Create Data Models: Define data models in Cognos Framework Manager to provide a structured foundation for reporting and analysis.

2. Report Development: Use Cognos Report Studio or Cognos Analytics to build reports and dashboards. Include a variety of visualizations such as bar charts, line graphs, pie charts, heat maps, and tables to represent sales data.

3. Interactivity: Enable interactivity in reports and dashboards by allowing users to filter, drill down, and slice-and-dice data to gain deeper insights.

4. Dashboard Design: Develop interactive dashboards that display key performance indicators (KPIs), trends, and other relevant information.

5. Scheduled Reporting: Schedule and automate report generation and distribution to relevant stakeholders on a regular basis.

# Derived Actionable Insights:

The analysis of product sales data using IBM Cognos can yield actionable insights, such as:

1. Identify Top-Performing Products: Determine which products are driving the most revenue and focus on optimizing their sales channels and marketing efforts.

2. Sales Channel Effectiveness: Assess the performance of various sales channels (e.g., online, retail, B2B) and reallocate resources accordingly.

3. Customer Segmentation: Tailor marketing and sales strategies to different customer segments, such as high-value customers or those in specific geographic regions.

4. Inventory Optimization: Use historical sales data to predict inventory needs, reducing carrying costs and stockouts.

5. Price Adjustments: Analyze the impact of pricing changes on sales and make data-driven decisions about price adjustments.

6. Trend Identification: Identify seasonal trends and respond with targeted marketing campaigns and inventory planning.

7. Sales Forecasting: Accurately forecast future sales to plan resources, inventory, and budgets effectively.

8. Performance Monitoring: Continuously monitor and compare sales performance against targets and adjust strategies as needed.

These actionable insights derived from the analysis of product sales data can help organizations make informed decisions, optimize their sales strategies, and ultimately improve their sales performance.

**Innovation Phase**

1. **Understanding the Data:**

Begin by understanding the dataset's structure, column names, data types, and identifying any missing or duplicate values. This step helps in formulating a strategy for cleaning the data effectively.

# Handling Missing Values:

Decide how to handle missing values in the dataset. Options include removing rows with missing values or filling missing values with appropriate measures like the mean or median.

# Handling Duplicates:

Identify and remove duplicate rows to ensure that the analysis is performed on unique data points, preventing skewed results.

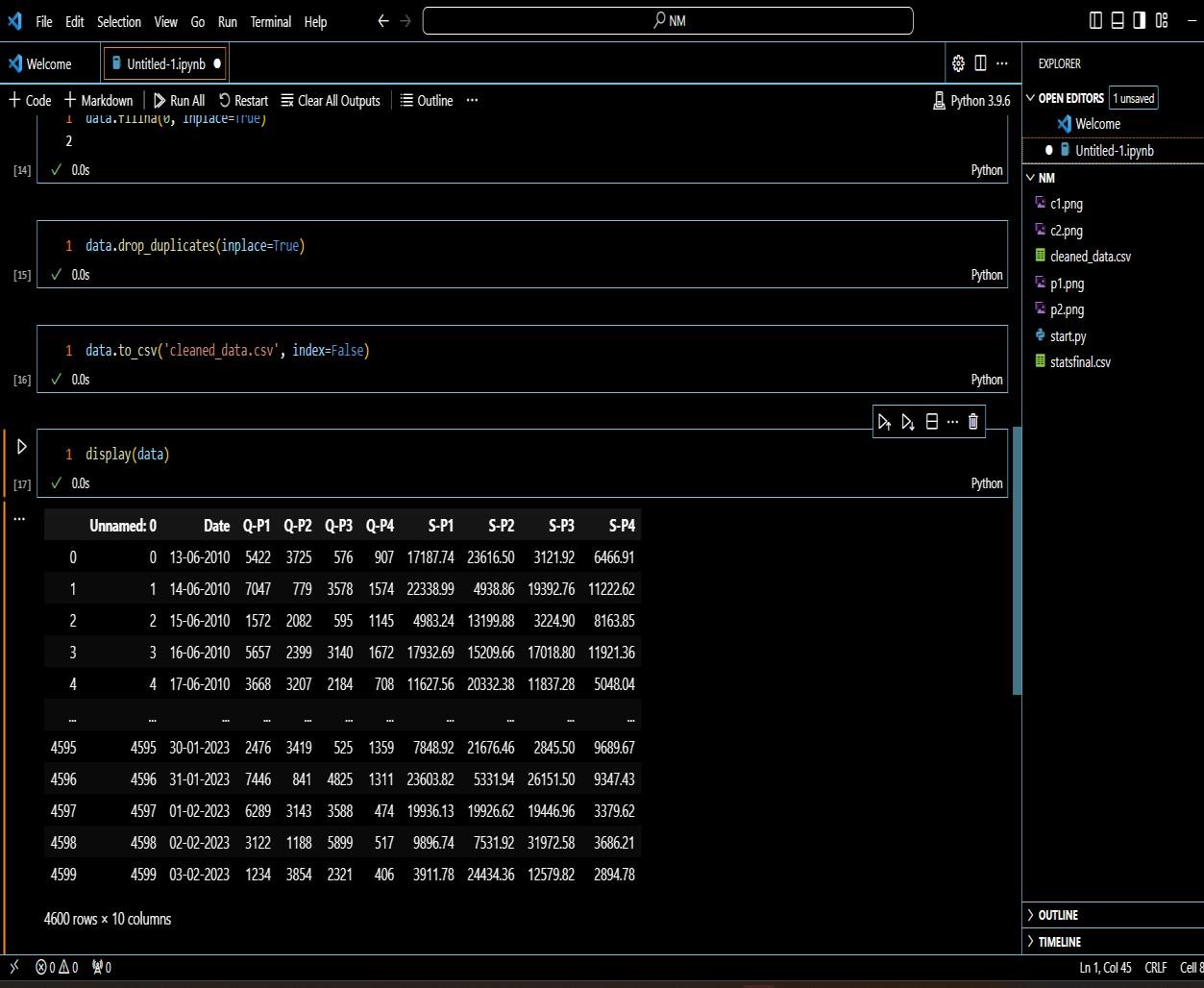
# Saving Cleaned Data:

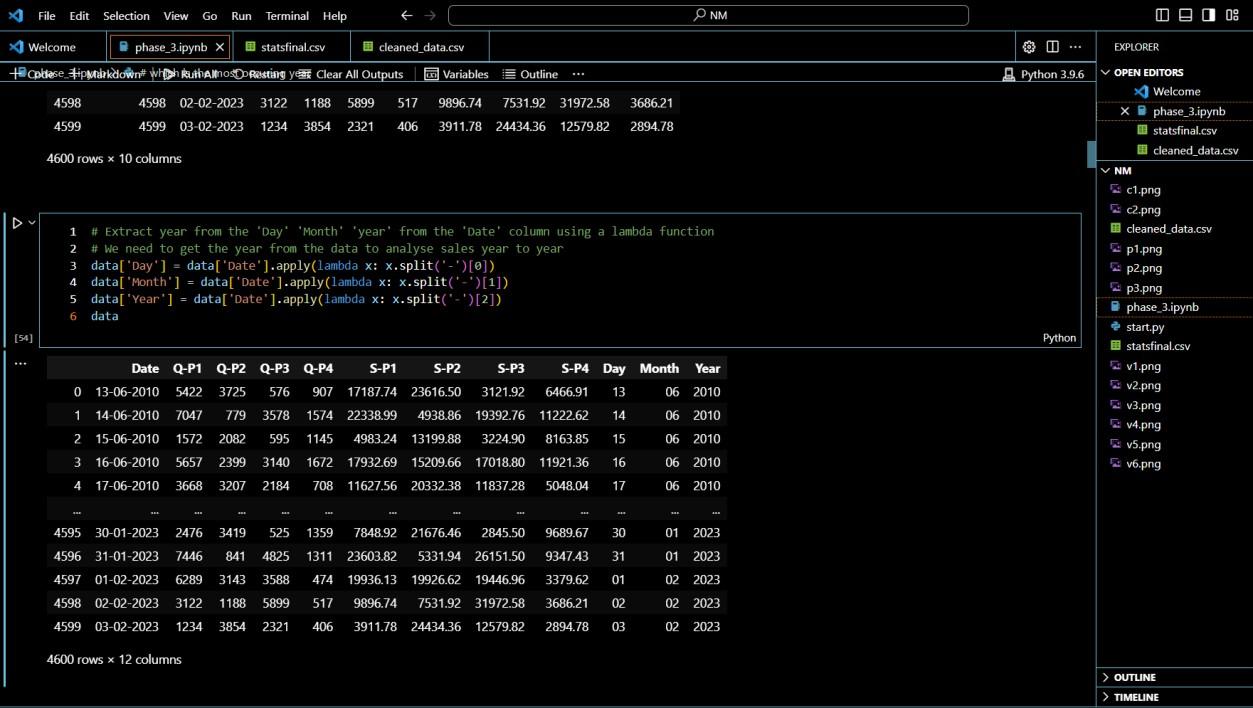
After completing the cleaning and preprocessing steps, save the cleaned dataset. This ensures that the cleaned data is available for further analysis and modeling without the need to repeat the cleaning process every time the analysis is performed.

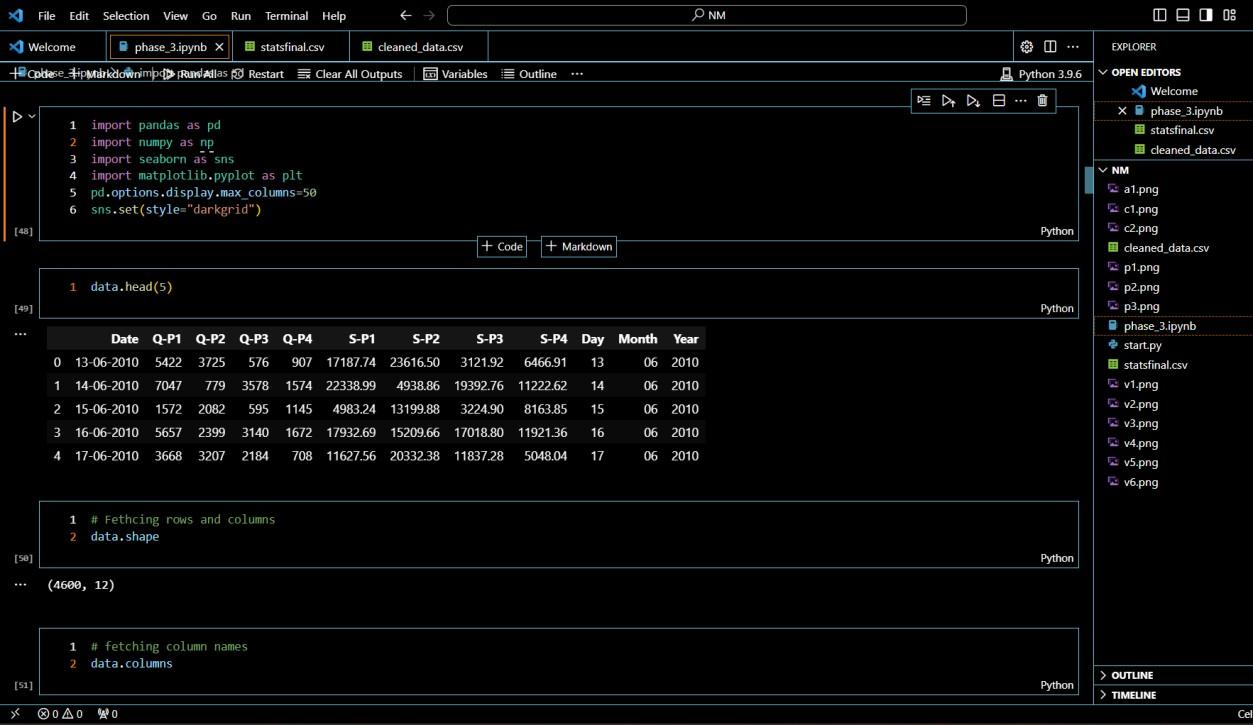
About Dataset

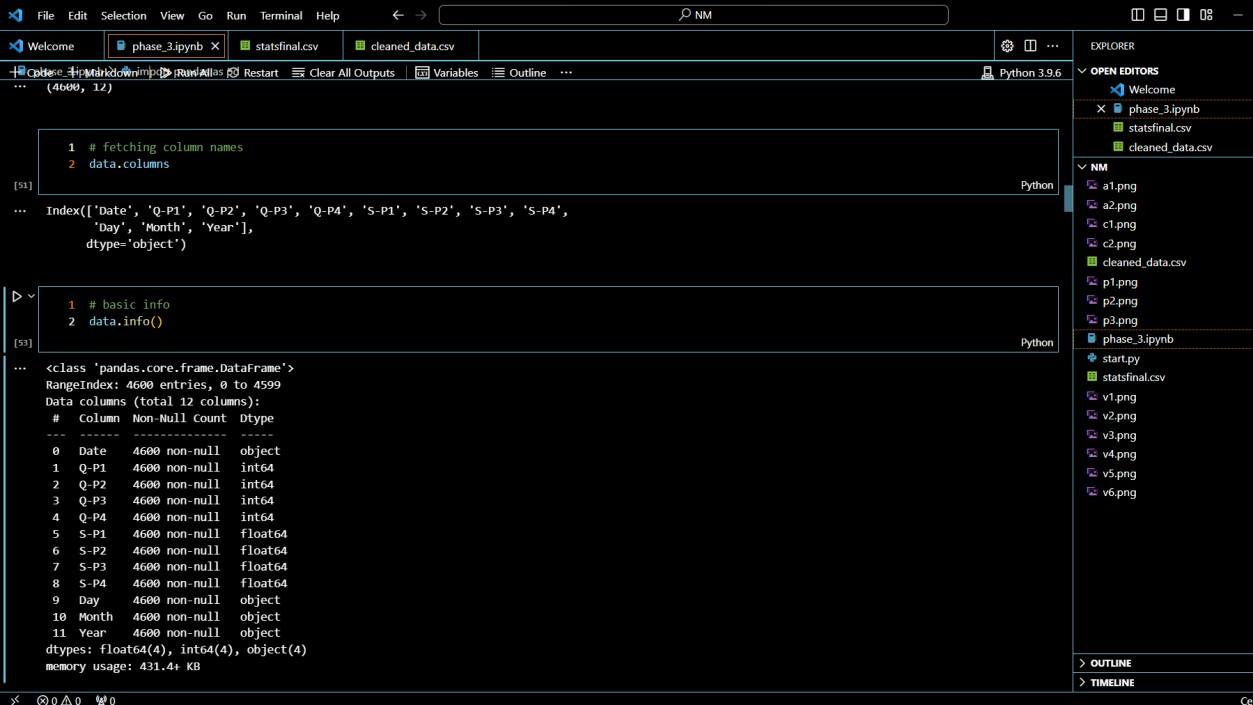
* + Q1- Total unit sales of product 1
  + Q2- Total unit sales of product 2
  + Q3- Total unit sales of product 3
  + Q4- Total unit sales of product 4
  + S1- Total revenue from product 1
  + S2- Total revenue from product 2
  + S3- Total revenue from product 3
  + S4- Total revenue from product 4

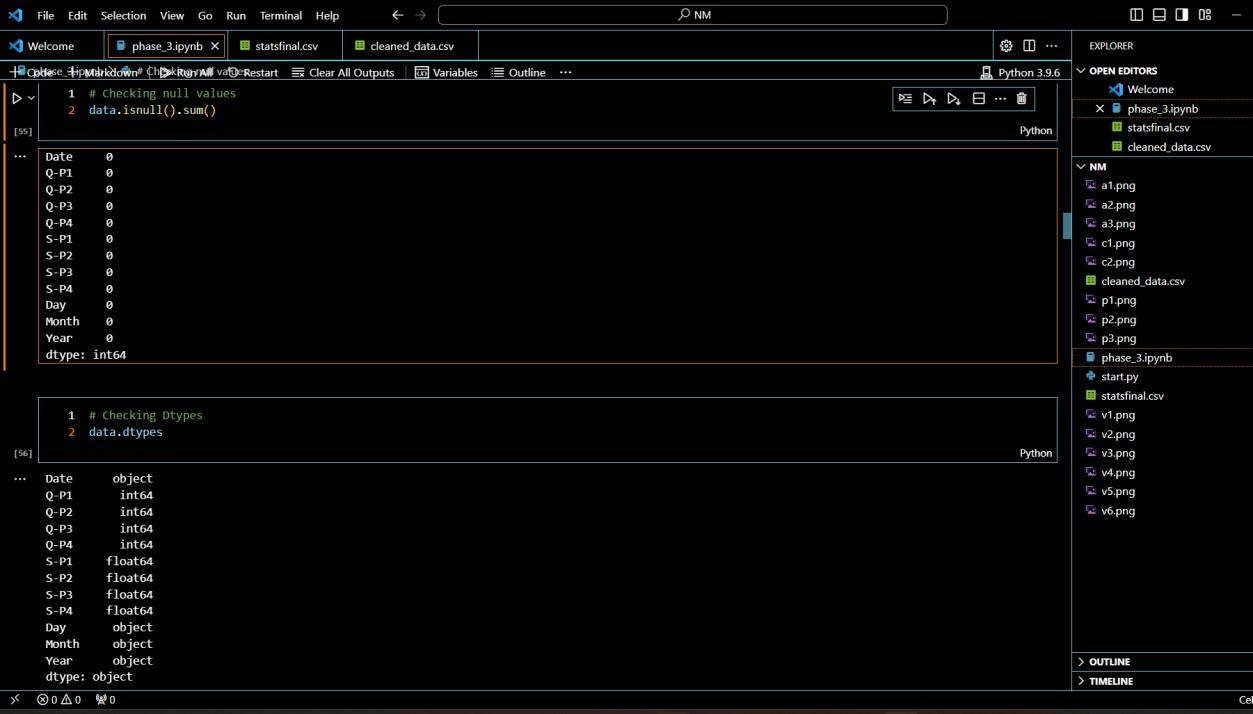
Data Cleaning



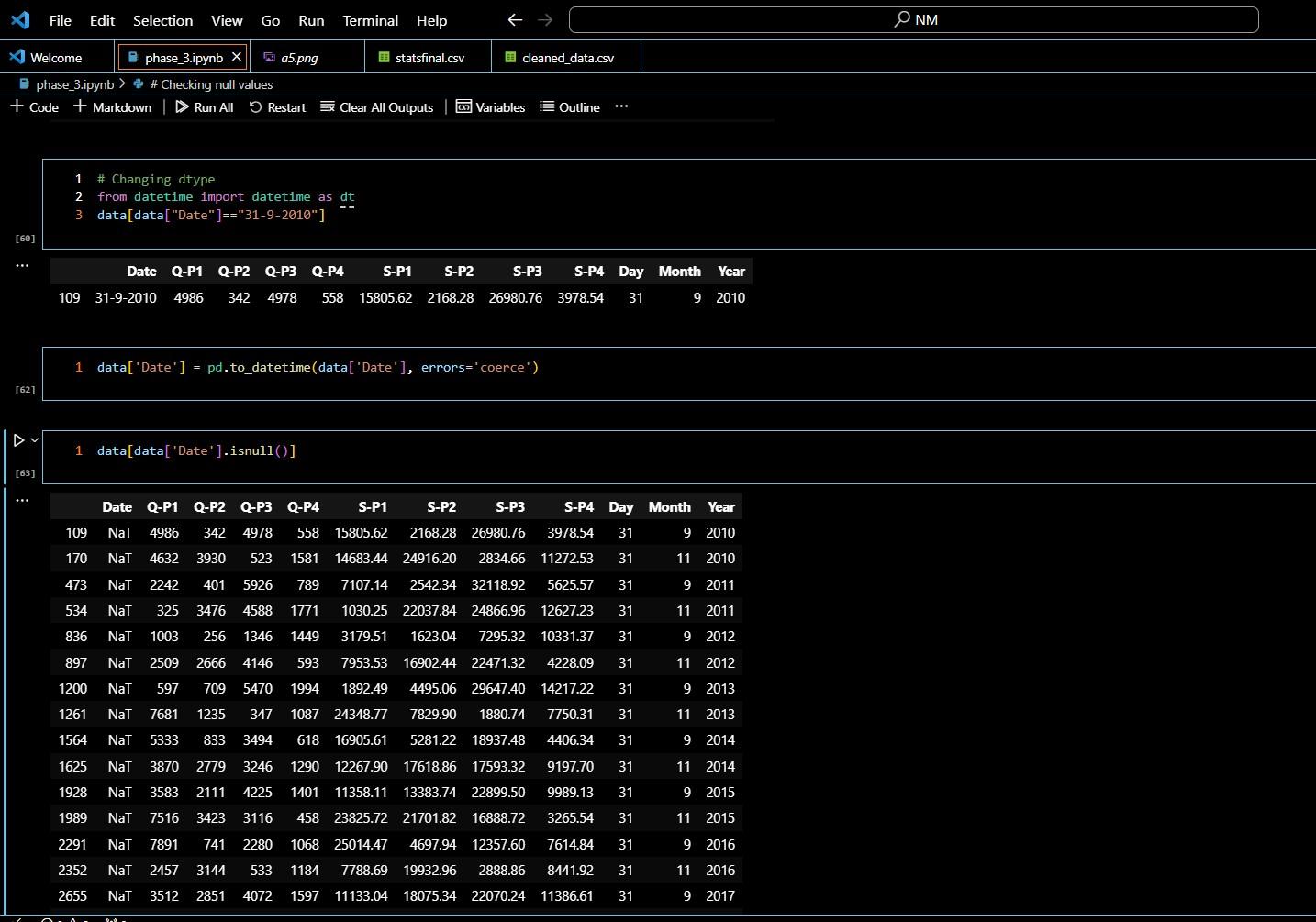


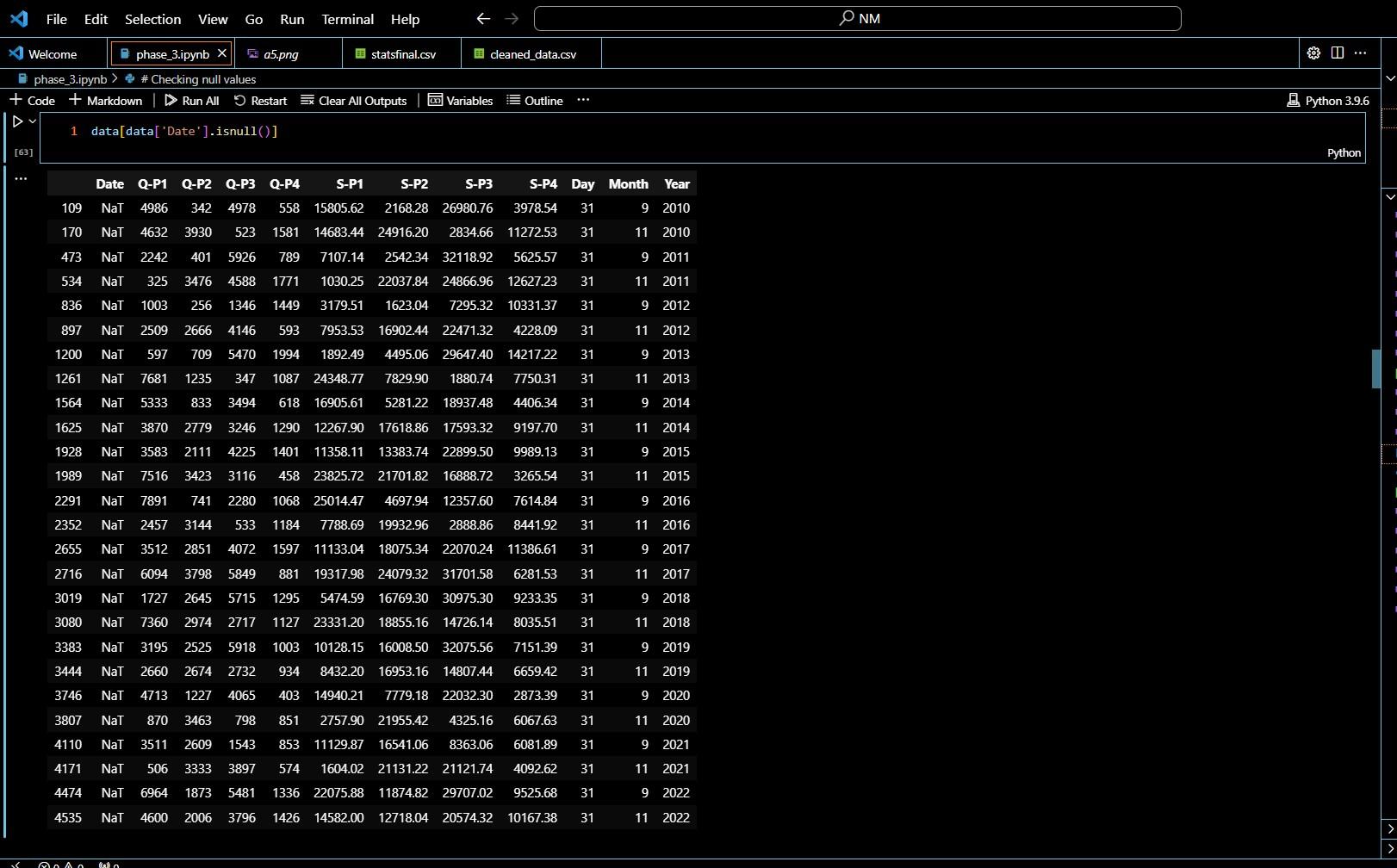




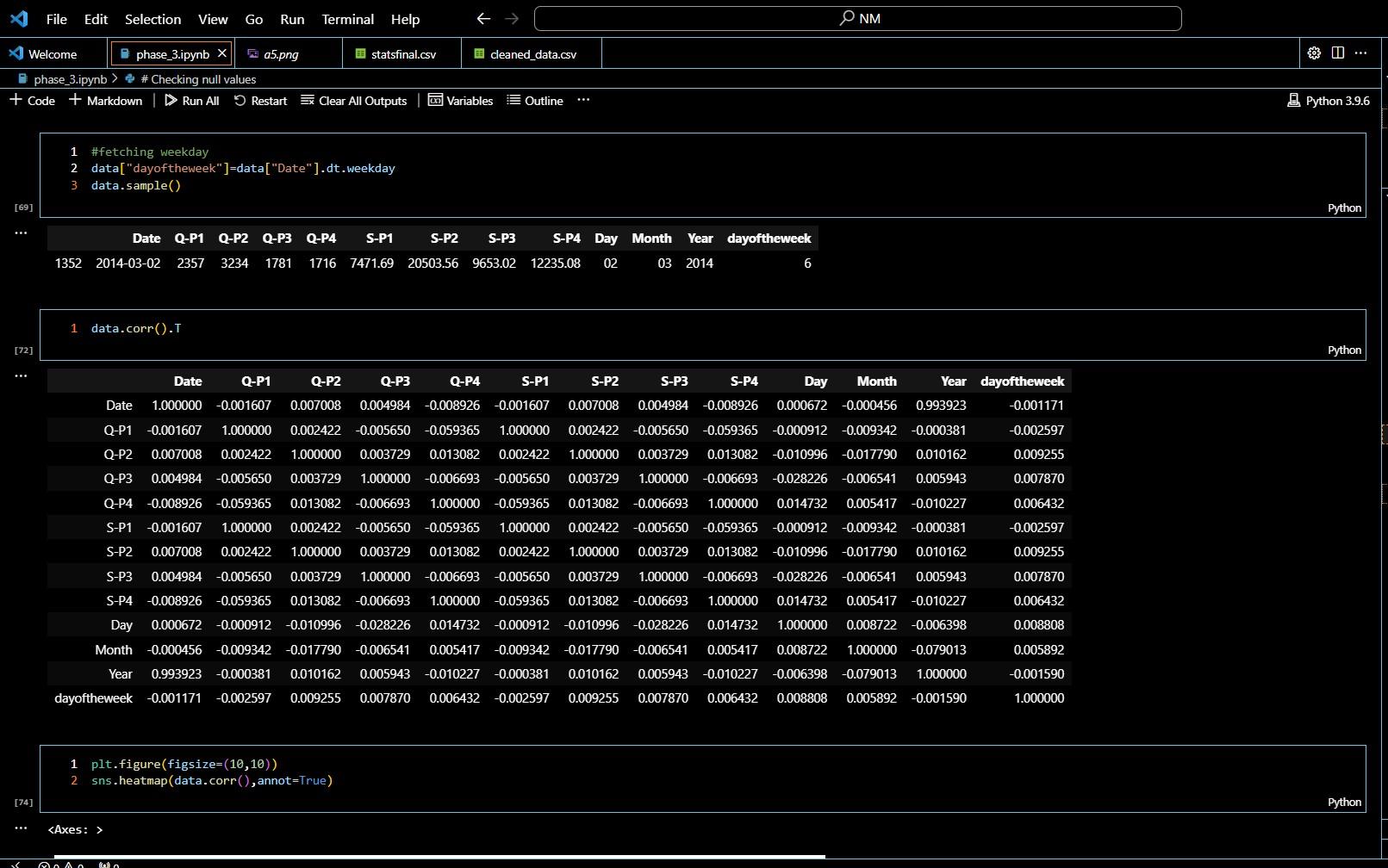


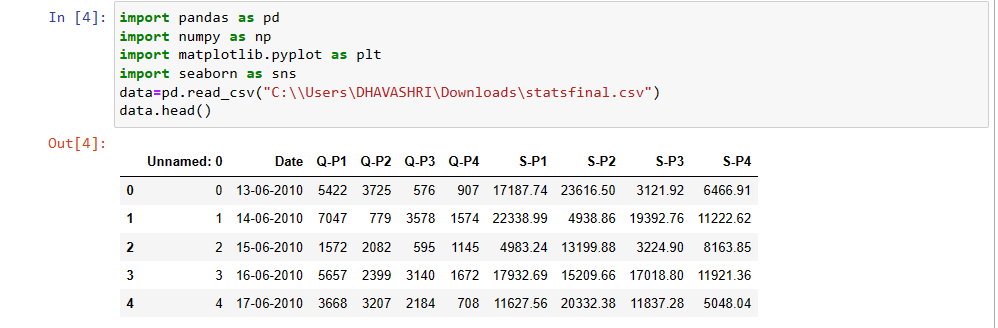












**1. Data Exploration**:

- Check the data's structure: `data.info()`

- Summary statistics: `data.describe()`

- Check for missing values: `data.isnull().sum()`

**2. Data Visualization:**

- Plot histograms or bar charts to visualize distributions.

- Create scatter plots for relationships between variables.

- Use Seaborn for more advanced visualization.

**3. Data Preprocessing**:

- Handle missing values (e.g., impute or remove rows/columns).

- Convert data types if necessary.

- Encode categorical variables.

**4. Analysis:**

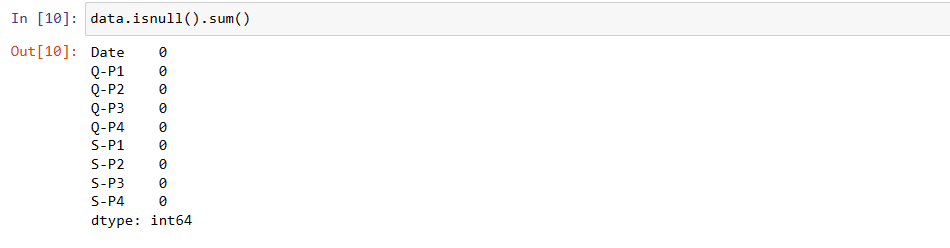
- Calculate total sales, profit, or any other relevant metrics.

- Perform time series analysis if your dataset includes timestamps.

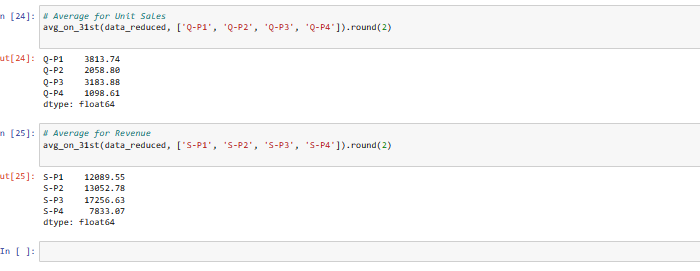
- Conduct hypothesis tests or statistical analysis if needed.

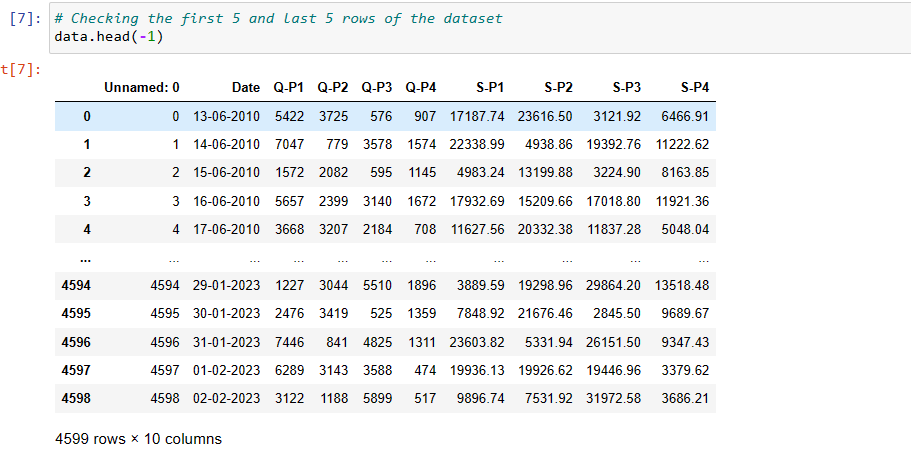
**5. Machine Learning (if applicable):**

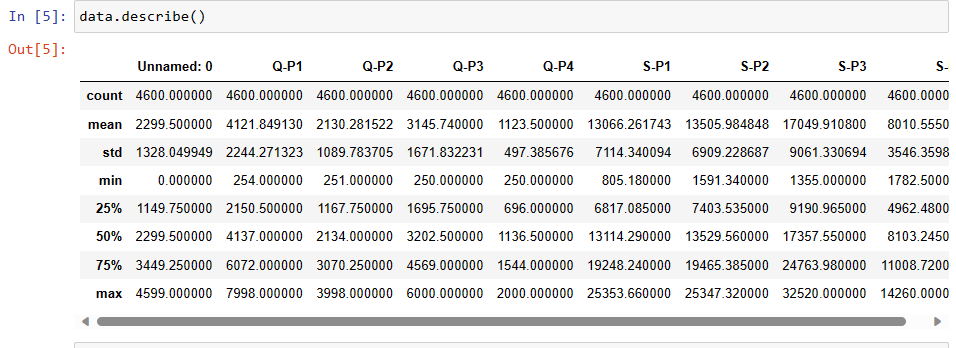
- Build predictive models for sales forecasting.



1. data.isnull(): This part of the code generates a DataFrame of the same shape as data, where each element is True if the corresponding element in data is null (missing), and False otherwise.
2. .sum(): This part of the code calculates the sum for each column in the DataFrame obtained in step 1. Since True is treated as 1 and False as 0 when you sum a boolean DataFrame, this effectively counts the number of missing values in each column.







1. Count: It counts the number of non-null (non-missing) values in each numeric column. This can give you an idea of whether there are any missing values in your dataset.

2. Mean: It calculates the arithmetic mean (average) for each numeric column, which gives you a measure of central tendency.

3. Standard Deviation (std): It measures the dispersion or spread of the data. A higher standard deviation indicates more variation in the data.

4. Minimum: It gives the minimum (lowest) value in each column, which can be useful for understanding the range of values.

5. 25th Percentile (25%): This represents the value below which 25% of the data falls. It's also called the first quartile.

6. 50th Percentile (50%): This is the median or the value below which 50% of the data falls. It's the second quartile.

7. 75th Percentile (75%): This represents the value below which 75%<class 'pandas.core.frame.DataFrame'>: This line tells you that you're working with a pandas DataFrame.

8. RangeIndex: 1000 entries, 0 to 999: It provides information about the index, including the number of entries and the range.

9. Data columns (total 5 columns): This line tells you that there are 5 columns in the DataFrame.

10. Column1 1000 non-null int64: This is information about the first column. It tells you the name of the column, the number of non-null (non-missing) values, and the data type (in this case, int64).

11. Column2 1000 non-null float64: Similar to the first column, this is information about the second column, which is a float.

12. Column3 1000 non-null object: Information about the third column, which is of type object (typically strings).

13. Column4 1000 non-null int64: Information about the fourth column, another integer column.

14. Column5 1000 non-null datetime64[ns]: Information about the fifth column, which is of type datetime.

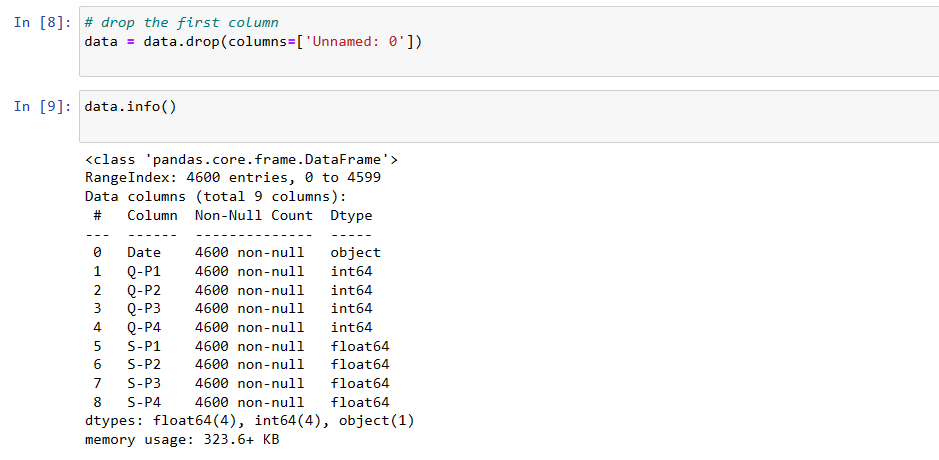
15. dtypes: This section lists the data types of all columns.

16. memory usage: It shows the memory usage of the DataFrame.

17. This information is very useful when you're cleaning, transforming, or analyzing a dataset because it gives you an initial understanding of the data's structure and helps identify potential issues with missing values or incorrect data types.

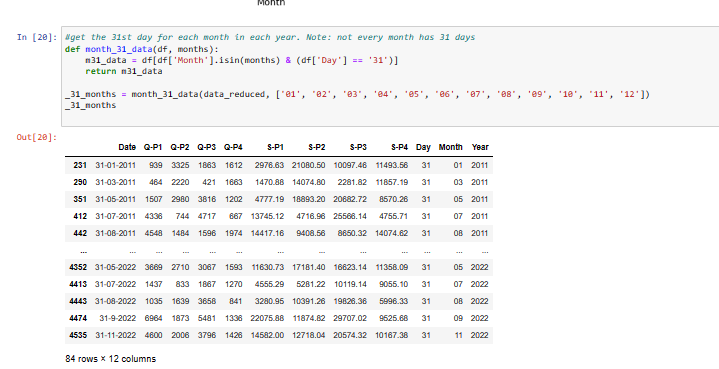
19. of the data falls. It's the third quartile.

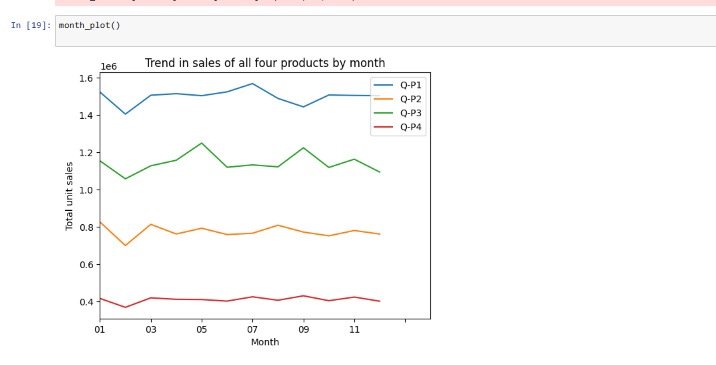
20. Maximum: It gives the maximum (highest) value in each column, which helps you understand the range of values.

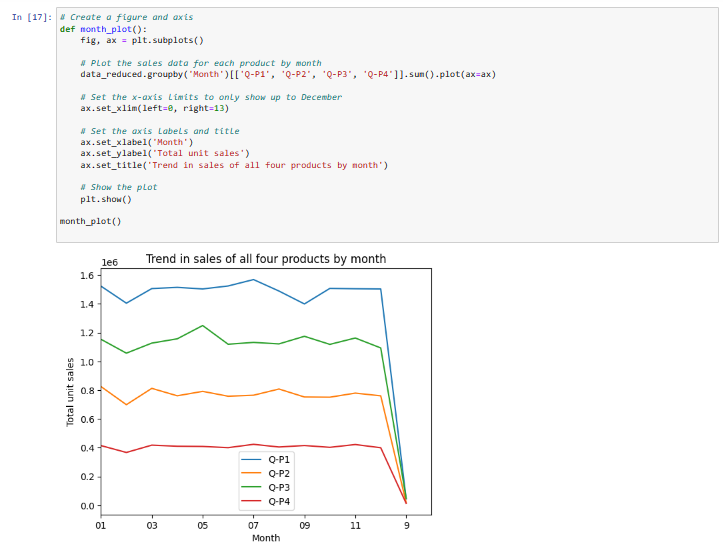


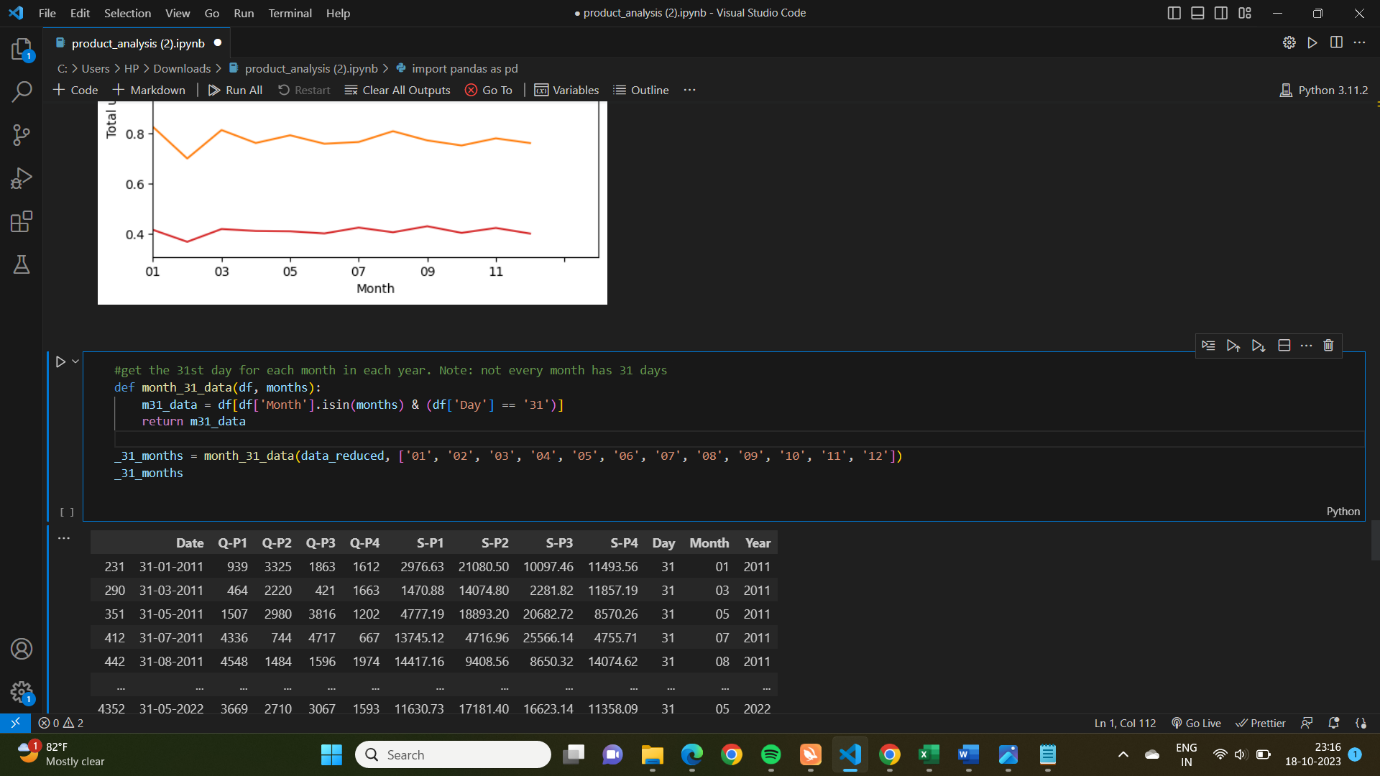
* <class 'pandas.core.frame.DataFrame'>: This line tells you that you're working with a pandas DataFrame.
* RangeIndex: 1000 entries, 0 to 999: It provides information about the index, including the number of entries and the range.
* Data columns (total 5 columns): This line tells you that there are 5 columns in the DataFrame.
* Column1 1000 non-null int64: This is information about the first column. It tells you the name of the column, the number of non-null (non-missing) values, and the data type (in this case, int64).
* Column2 1000 non-null float64: Similar to the first column, this is information about the second column, which is a float.
* Column3 1000 non-null object: Information about the third column, which is of type object (typically strings).
* Column4 1000 non-null int64: Information about the fourth column, another integer column.
* Column5 1000 non-null datetime64[ns]: Information about the fifth column, which is of type datetime.
* dtypes: This section lists the data types of all columns.
* memory usage: It shows the memory usage of the DataFrame.

This information is very useful when you're cleaning, transforming, or analyzing a dataset because it gives you an initial understanding of the data's structure and helps identify potential issues with missing values or incorrect data types.









1. df is the DataFrame containing the dataset.
2. months is a list of months (in the format '01', '02', ..., '12') for which you want to retrieve data for the 31st day.
3. The function first filters the DataFrame to include rows where the 'Month' column matches one of the specified months and the 'Day' column is '31'.

