CLIENT PROJECT DOCUMENTATION:

Project ID: CDACL005

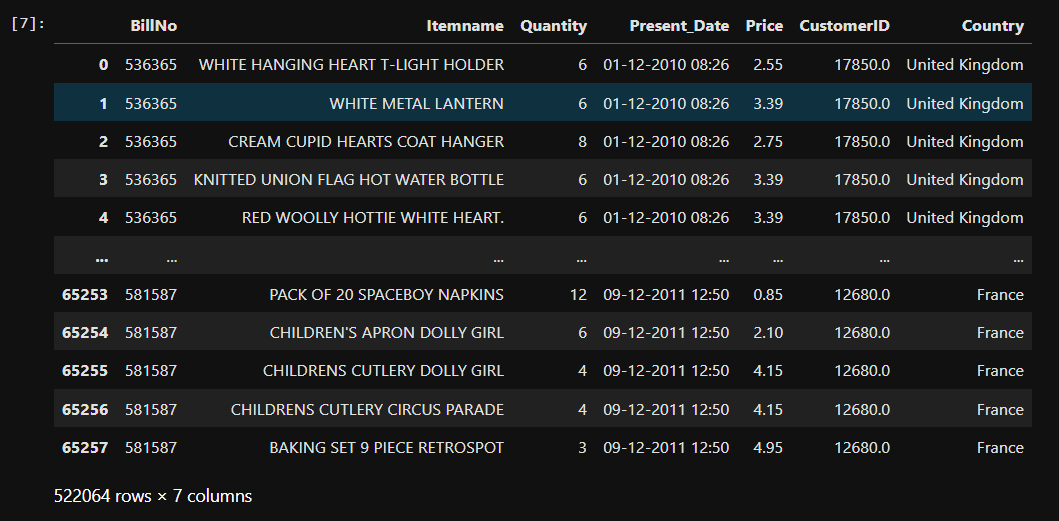
Project Team ID: PTID-CDA-MAY-25-475

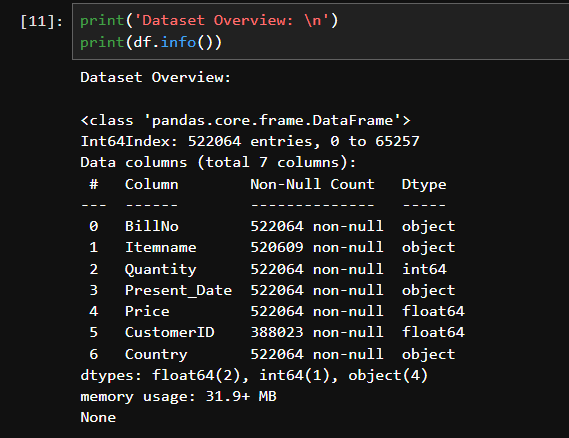
Team Members: Sridhar M, Vaamshikan M

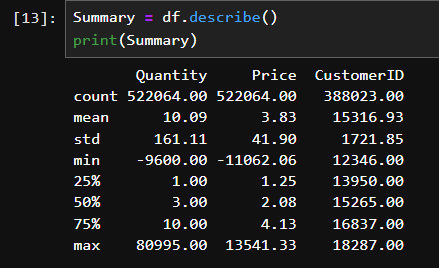
**Week 1: Project Kick-off and Data Exploration**

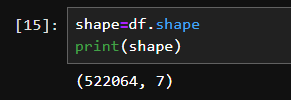
* Acquired and loaded the transaction dataset using **pandas**.
* Explored dataset structure, column types, and initial value samples.
* Performed Exploratory Data Analysis (EDA) using **pandas** and **seaborn**.
* Computed key statistics like mean, median, and mode.
* Visualized data distributions and patterns using **matplotlib** and **seaborn**.
* Identified missing values and outliers to guide data cleaning in the next phase.

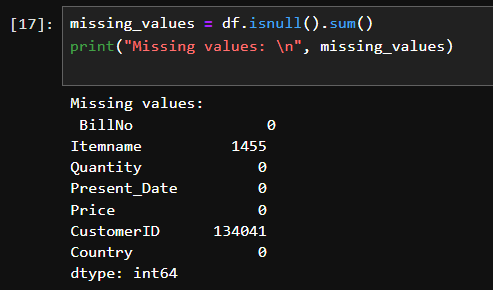
Results of Week 1:

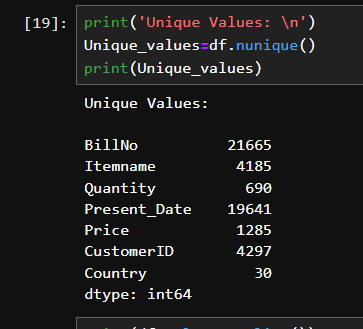


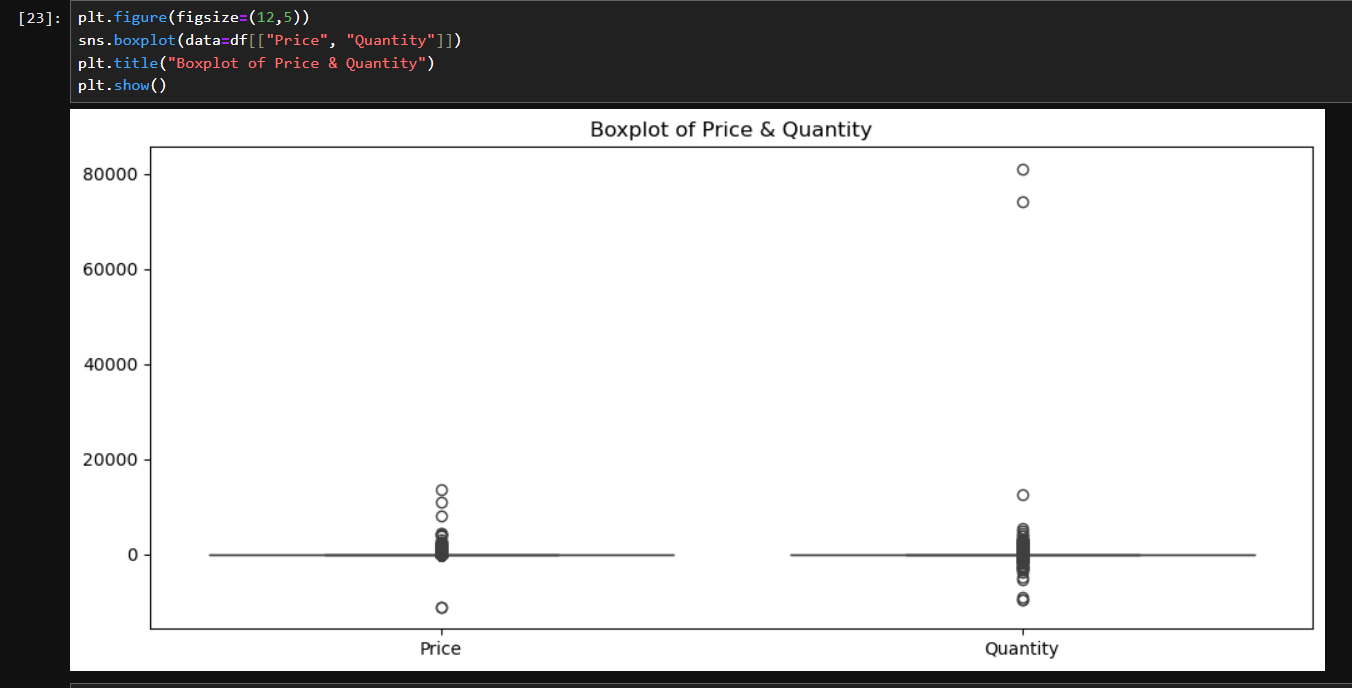


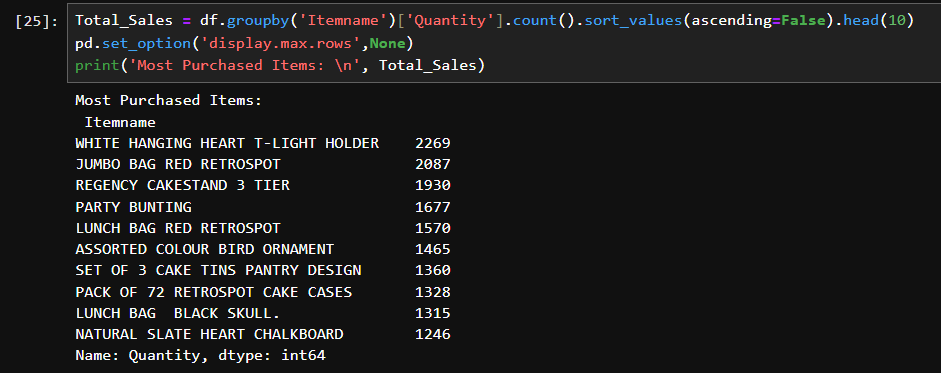


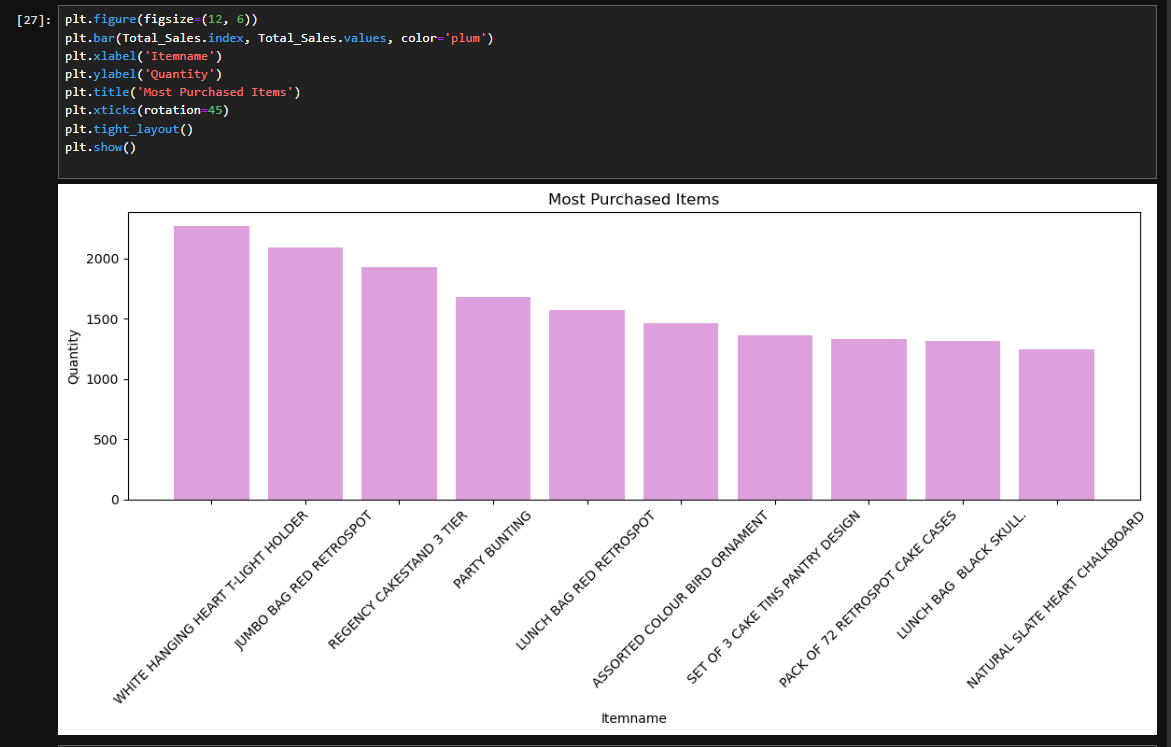


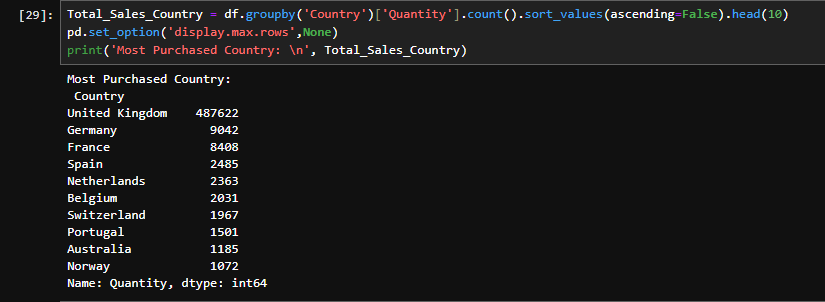


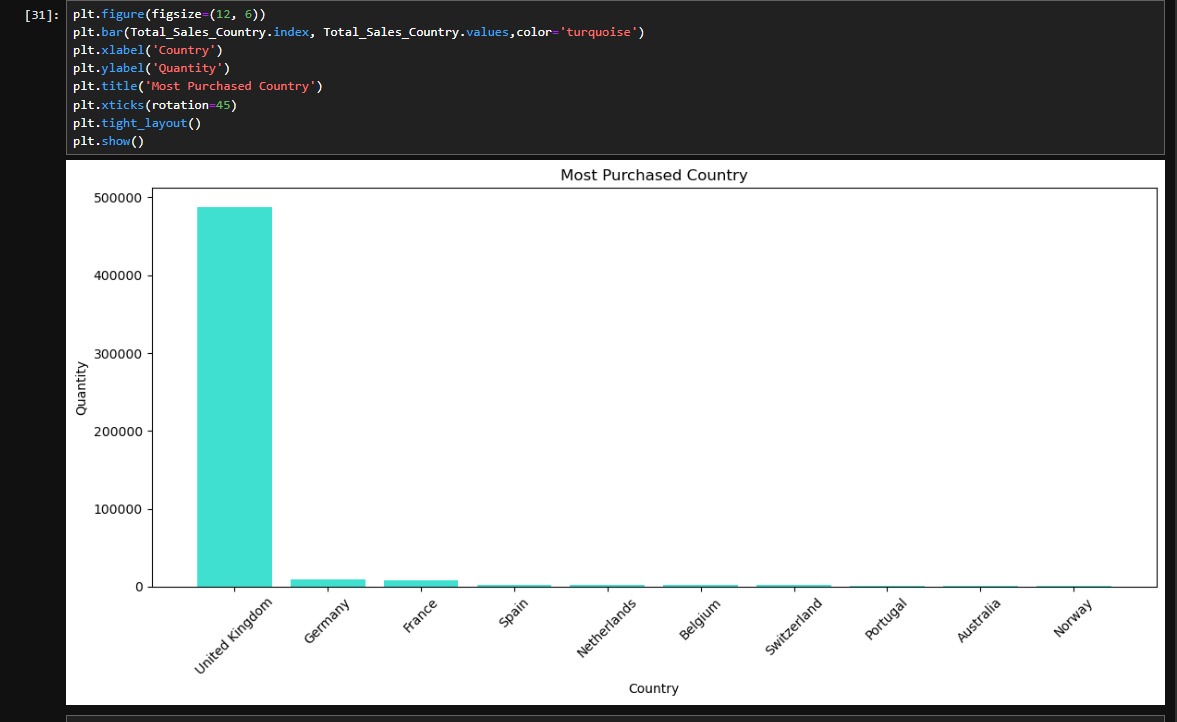
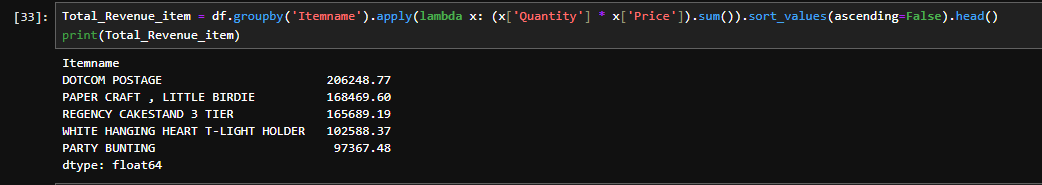


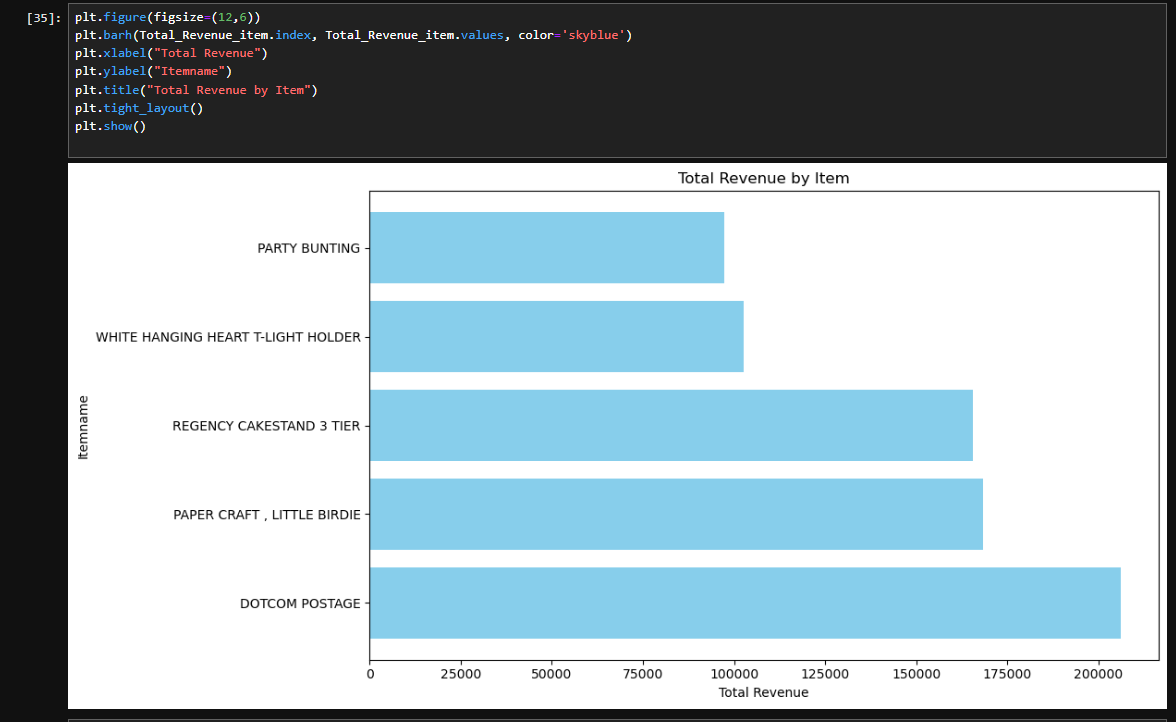


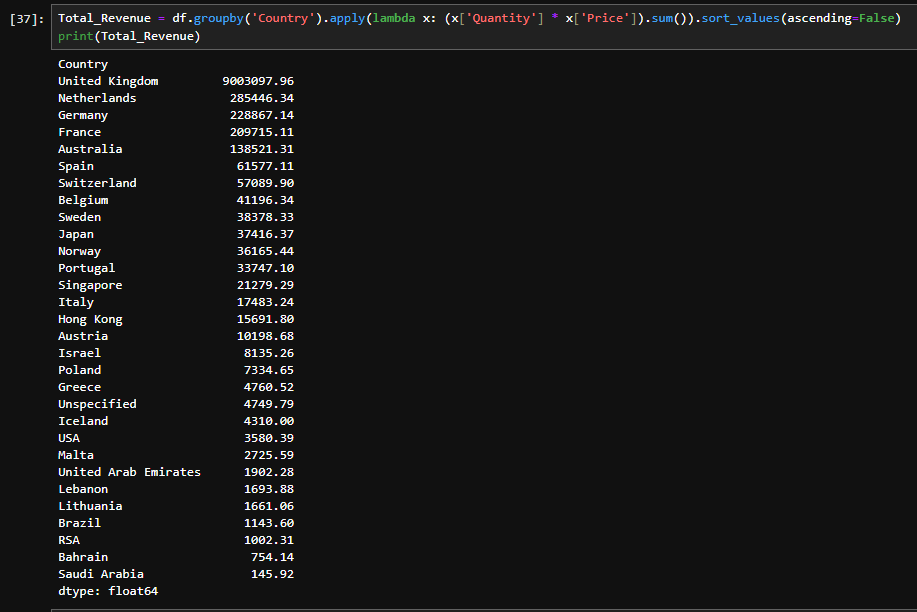


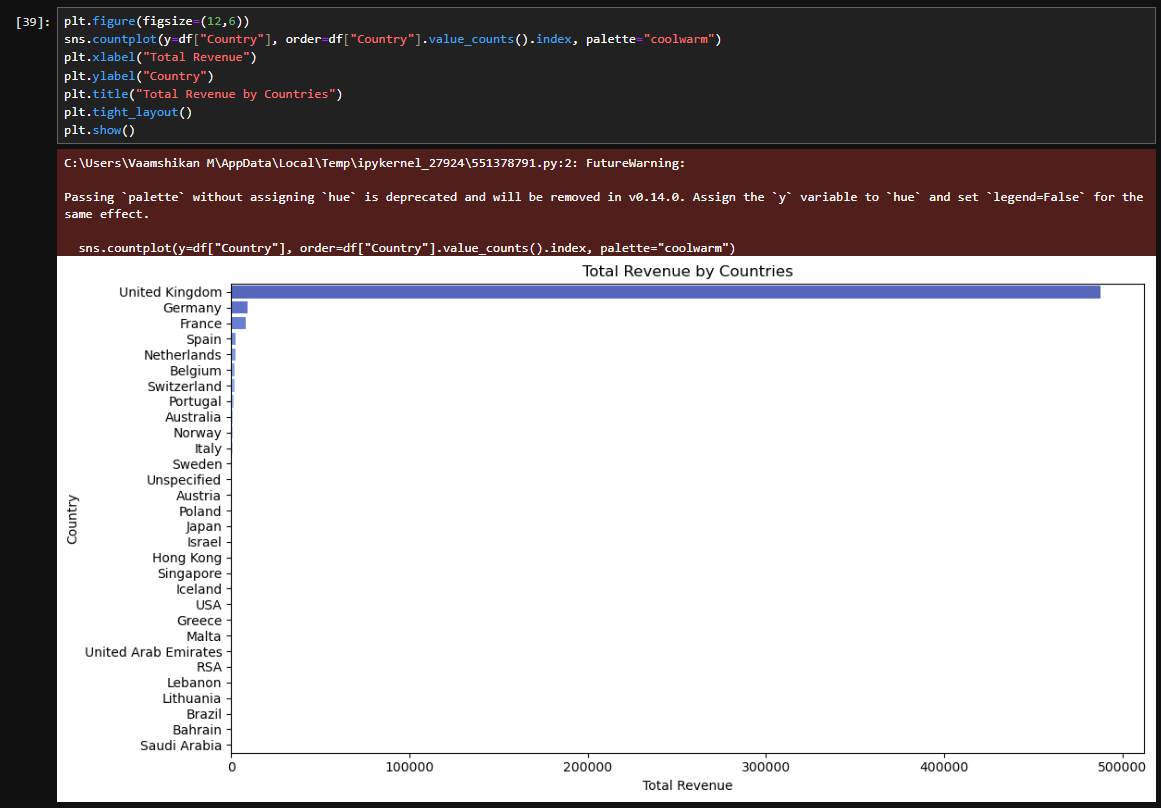


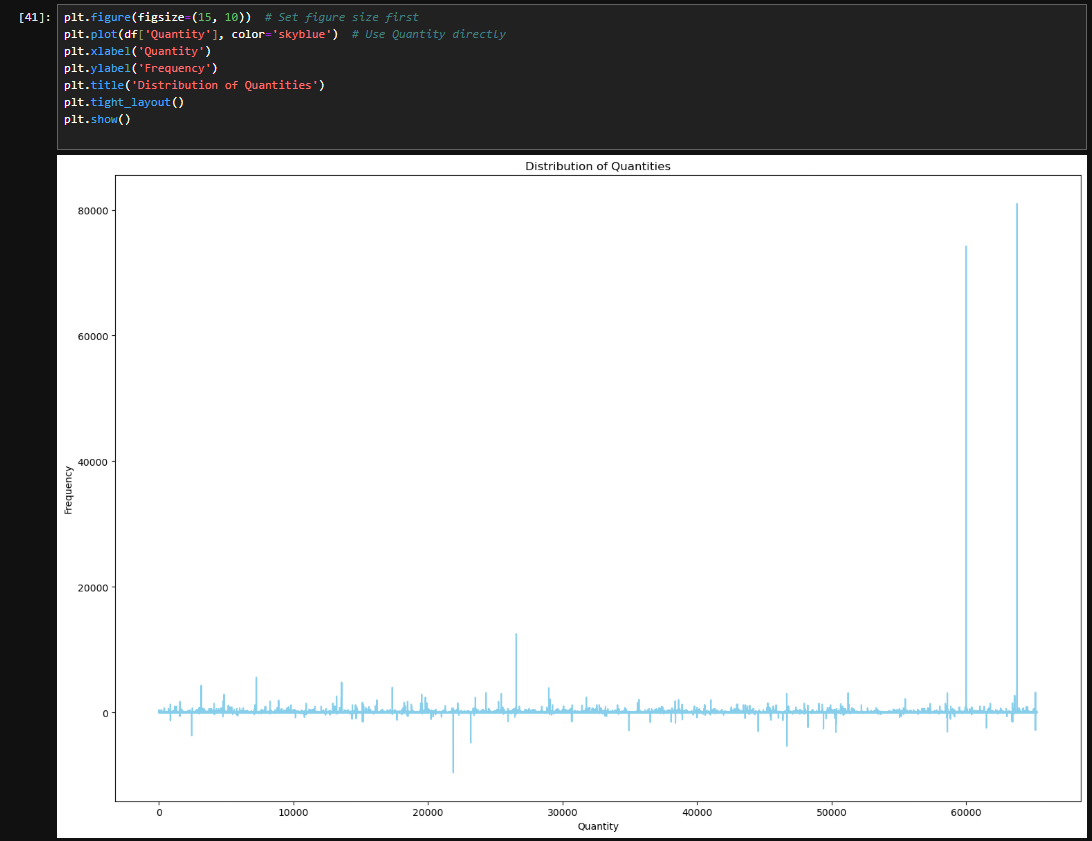


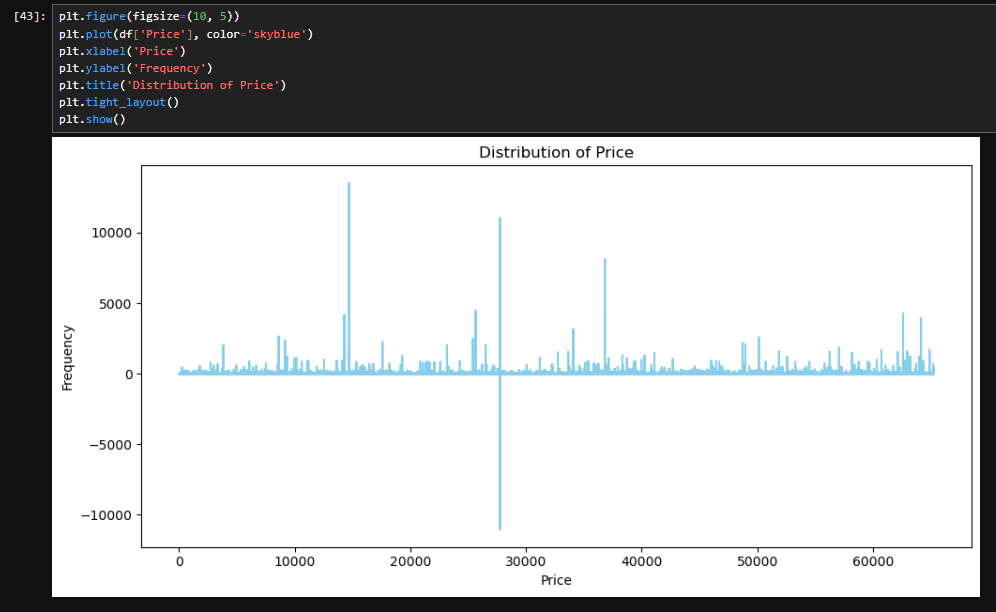
 







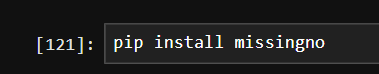


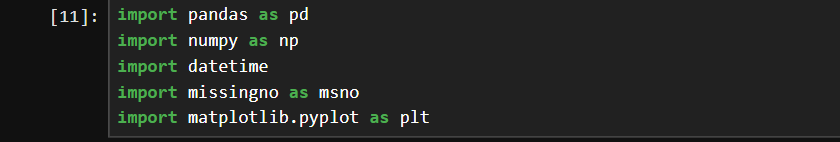


**Week 2: Data Cleaning and Transformation**

**(along with results)**

**Step:1:**



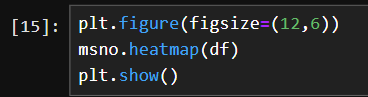


Start by installing and importing the required libraries.

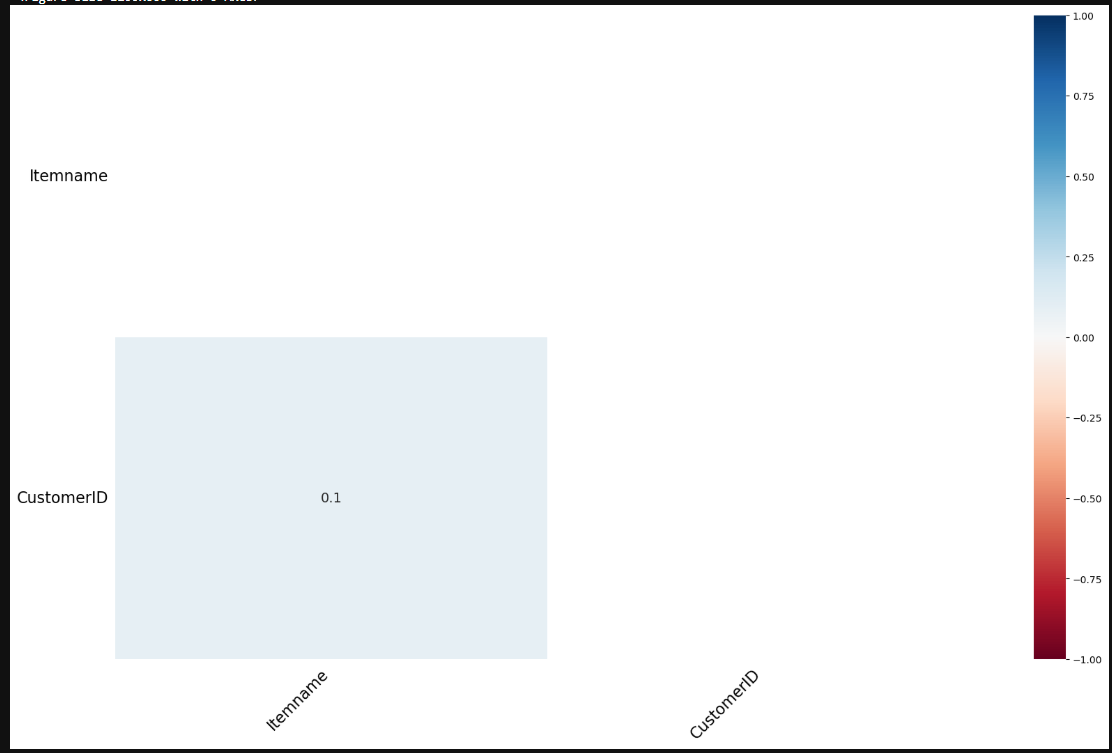
**Step:2:**

Import the required data from your designated folder using the appropriate pandas read function. Use pd.read\_csv() for CSV files, pd.read\_table() for text files, pd.read\_excel() for Excel files, and pd.read\_json() for JSON files.

**Step:3:**



A heatmap is generated using the missingno and matplotlib.pyplot libraries to effectively visualize and analyze missing data patterns.



**Step:4:**



Converts the 'Present\_Date' column to datetime format using pd.to\_datetime() to enable easier date-based operations and analysis. 

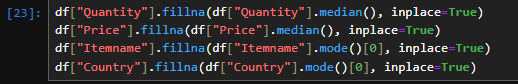
**Step:5:**



Extracts the date and time components from the 'Present\_Date' column and stores them in separate columns: 'Date' for the date and 'Time' for the time.



**Step:6:**



Fills missing 'Quantity' and 'Price' with their respective **medians** (robust to outliers).

Fills missing 'Itemname' and 'Country' with their respective **modes** (most frequent value).

Ensures in-place updates to the original DataFrame for efficiency.



**Step:7:**



Converts each country name in the 'Country' column to title case (e.g., "united kingdom" → "United Kingdom").

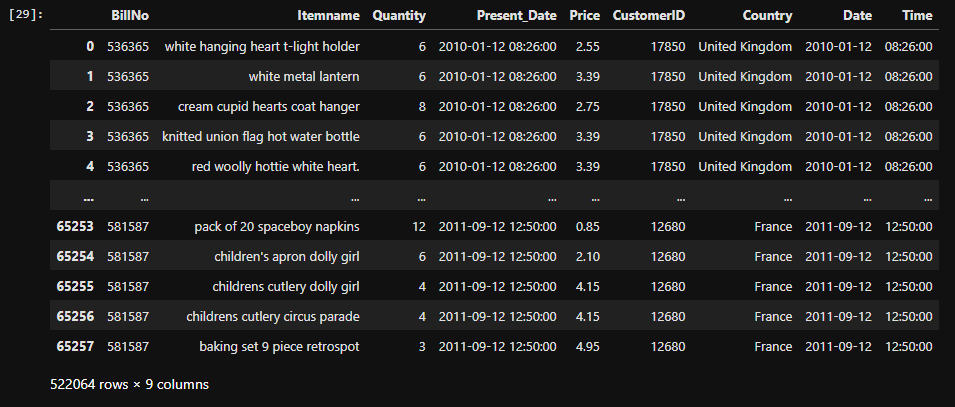
Cleans the 'Itemname' column by removing leading/trailing spaces and converting all text to lowercase for consistency.



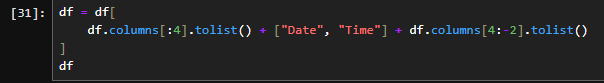
**Step:8:**



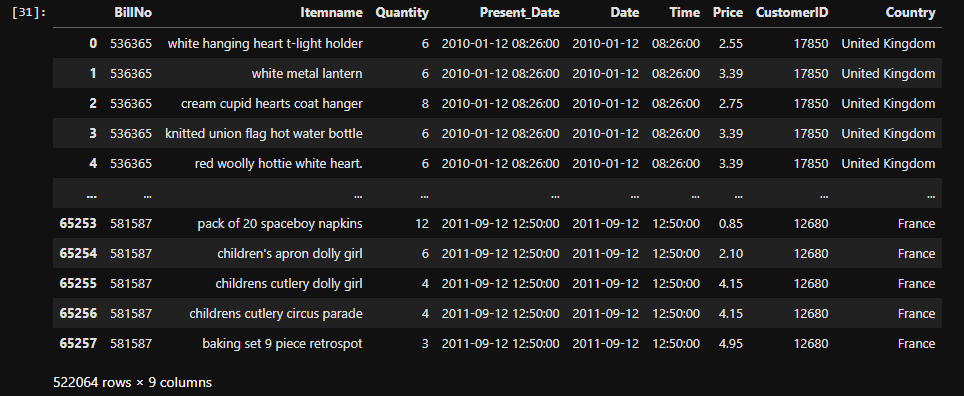
Converts the 'CustomerID' column to integer format (Int64).  
Non-numeric values are coerced into NaN using pd.to\_numeric(..., errors="coerce"), and the result is cast to Int64 to support nullable integers.



**Step:9:**



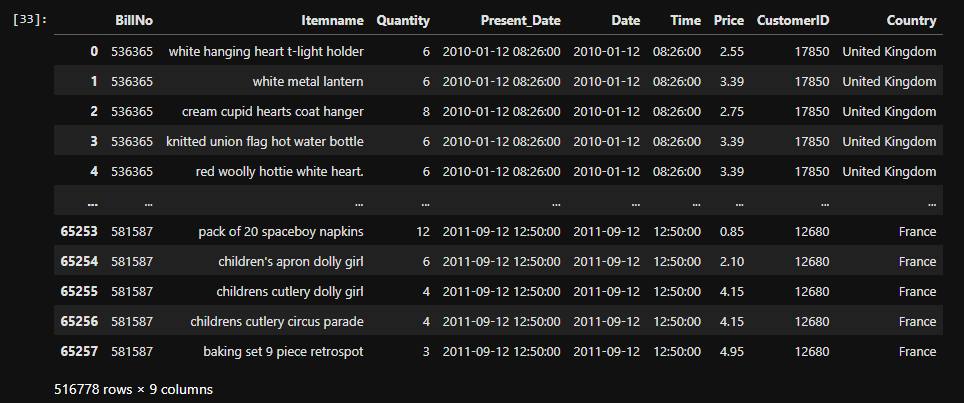
Reorders the DataFrame columns to place 'Date' and 'Time' immediately after the first four columns. This is done by slicing the column list and inserting the new columns ("Date" and "Time") between existing ones for improved readability or structure.



**Step:10:**



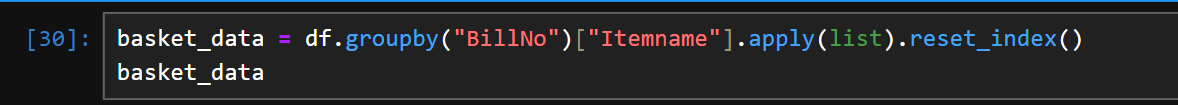
Removes duplicate rows from the DataFrame using drop\_duplicates(), keeping only the first occurrence of each unique row.



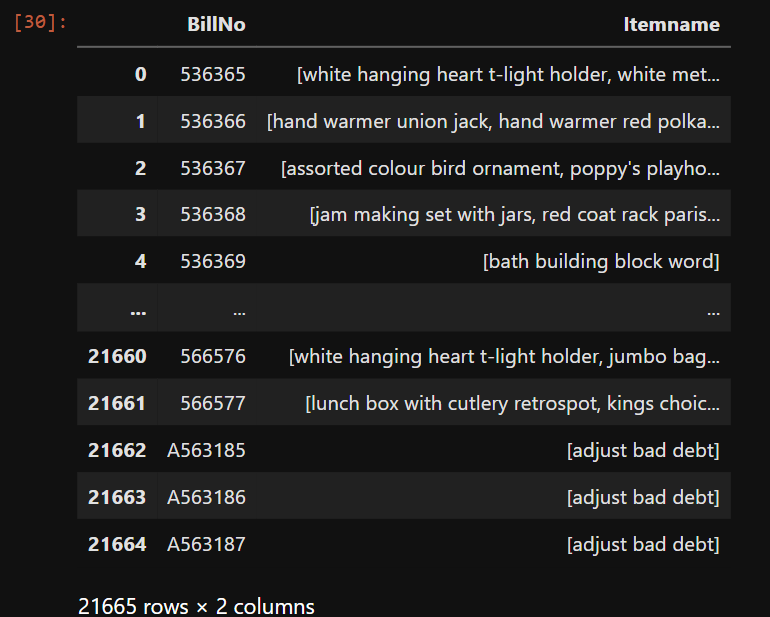
The dataset has been thoroughly prepared for analysis through a series of essential preprocessing steps:

**Week 3: Data Visualization and Apriori Algorithm Implementation**

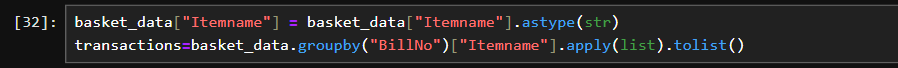
* Installed mlxtend and imported apriori, association\_rules, and TransactionEncoder.
* Grouped transactions using:

****

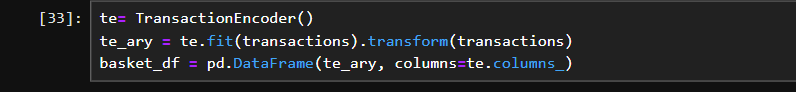
* Created item lists per transaction (BillNo).



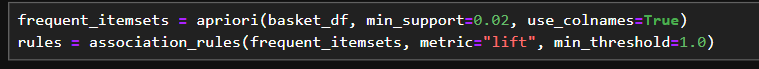
* Ensures all item names are of string type for consistency.
* Groups transactions by invoice (BillNo), aggregating purchased items into lists.



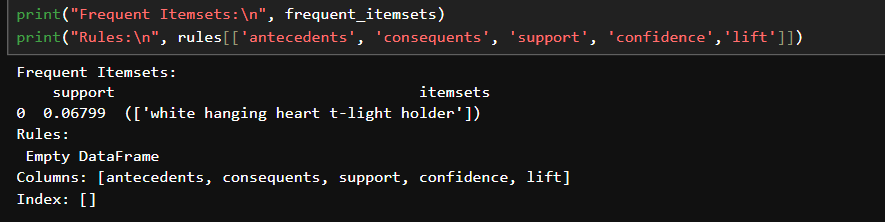
* Applies one-hot encoding to convert transactional data into a format suitable for analysis.
* Constructs a binary DataFrame indicating item presence per transaction.



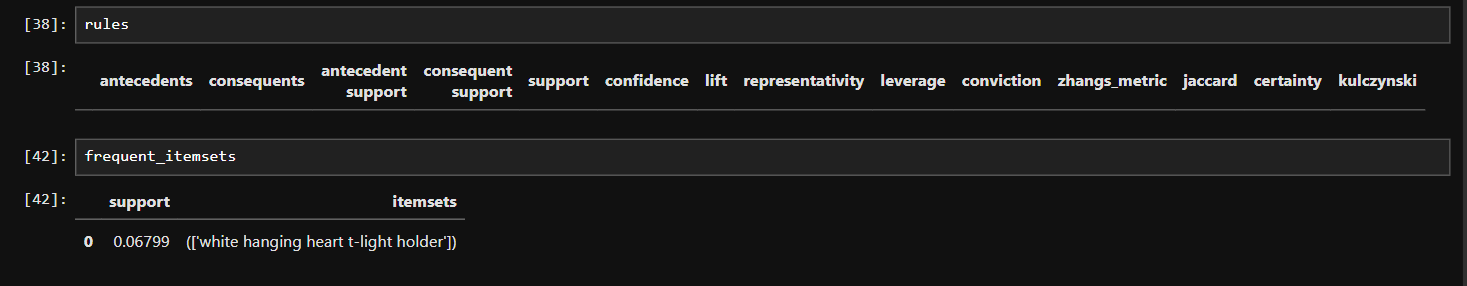
* Executes the Apriori algorithm to identify item combinations with support ≥ 2%.
* Uses item names instead of column indices for readability.
* Derives association rules from the frequent itemsets based on the lift metric.
* Filters rules with a minimum lift of 1.0 to ensure meaningful associations.



* Outputs the identified frequent itemsets and associated rule metrics (support, confidence, lift).



Result of Week 3:

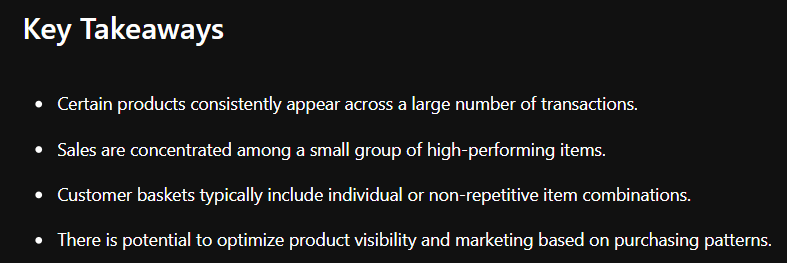


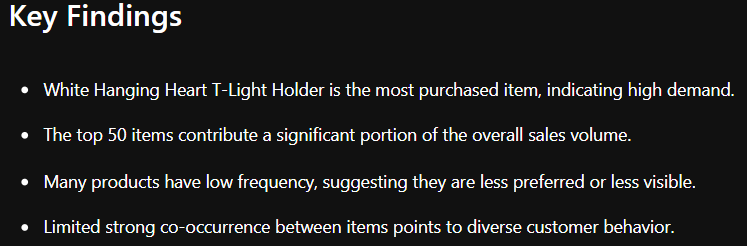
**Week 4: Insights, Recommendations, and Finalization**

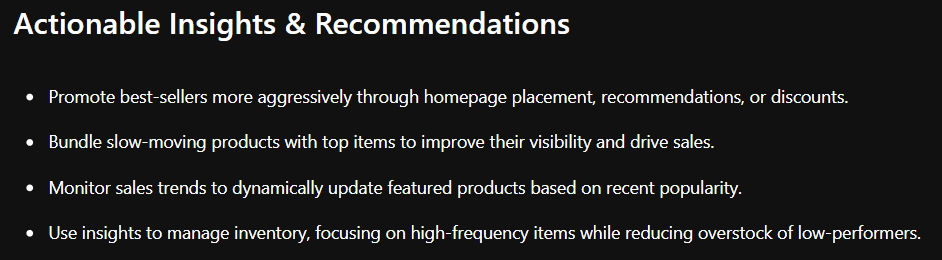
Week 4 compiles overall project outcomes:

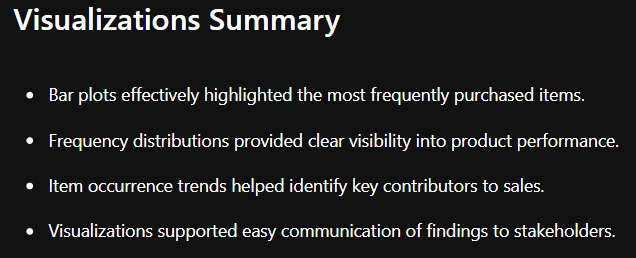
* Key Findings: Cleaned data revealed top-selling items and patterns.
* Takeaways: Noted customer purchase trends and product associations.
* Insights: Identified bundling and inventory opportunities.
* Recommendations: Improve cross-selling, layout, and promotions.
* Visual Summary: Charts supported findings and decisions.

****

****

****

****

****

**Conclusion:**

The analysis shows that a few key products drive most sales, with customers displaying varied buying patterns. This highlights strong opportunities for promoting best-sellers and bundling lesser-known items. Leveraging these insights can boost both sales and customer engagement through smarter merchandising.