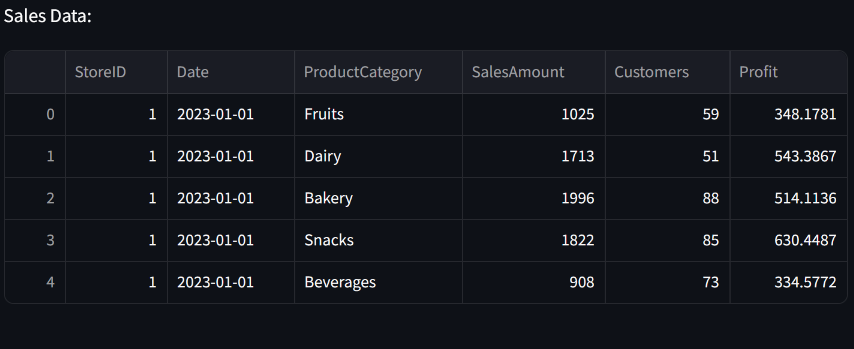
**GreenGrocer Analytics – Python Streamlit Application**

This report outlines the design and functionality of the GreenGrocer Analytics Dashboard, a Streamlit-based interactive application built to analyze sales and customer data from a grocery retail chain operating across several European cities.

The following sections correspond to key Python functionalities implemented using data science libraries such as pandas, geopandas, scikit-learn, statsmodels, and matplotlib.

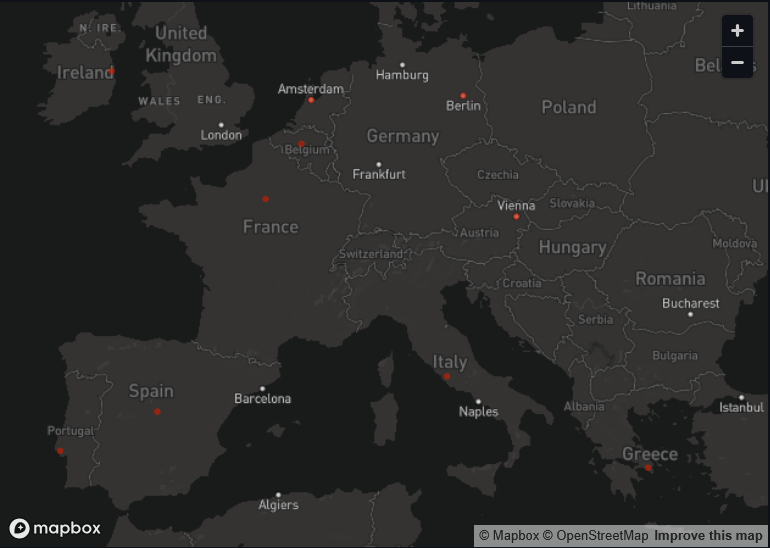
**1. Data Preview**

To begin, the application loads and displays the first rows of the sales dataset. This helps users quickly understand the structure and quality of the data.



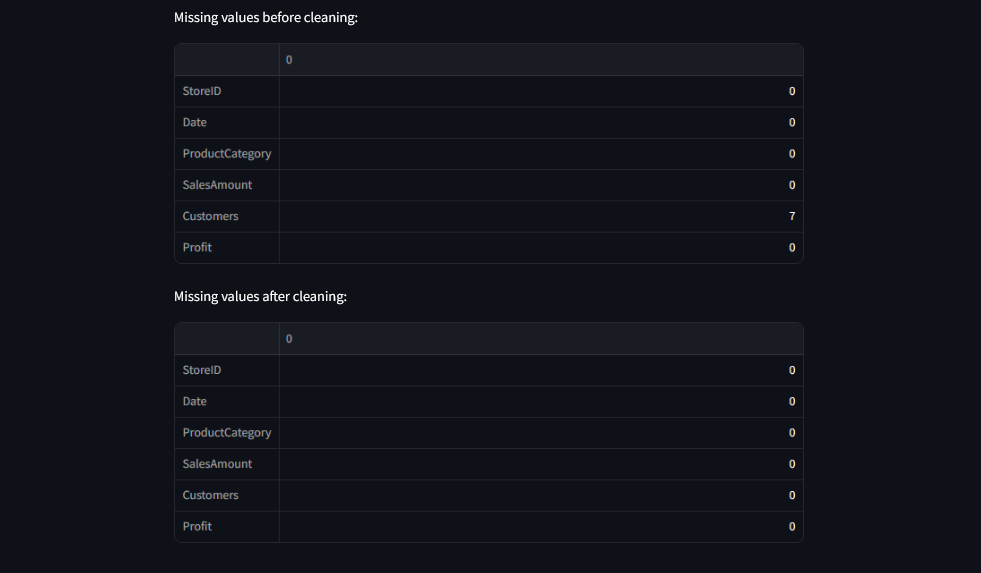
**2. Store Locations**

The second section uses Geopandas to create a spatial map of the store locations based on their latitude and longitude. This map provides a geographic view of the retail network.



**3. Handling Missing and Extreme Values**

We handle missing values by filling null entries in the "Customers" column with the column's median value. This avoids distortions caused by missing or extreme records.



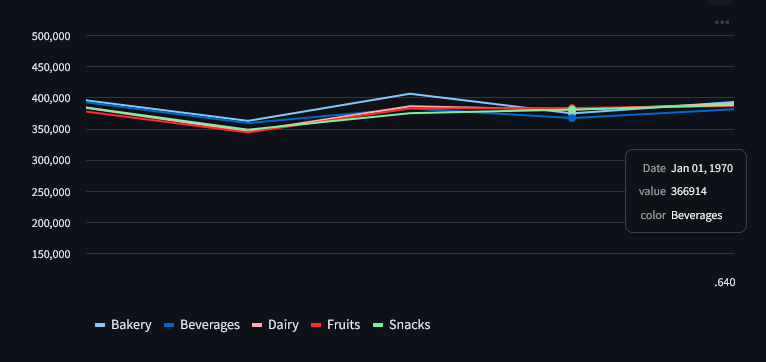
**4. Encoding and Scaling**

To prepare data for machine learning models, we encoded categorical data (ProductCategory) using label encoding and scaled numeric features using StandardScaler.



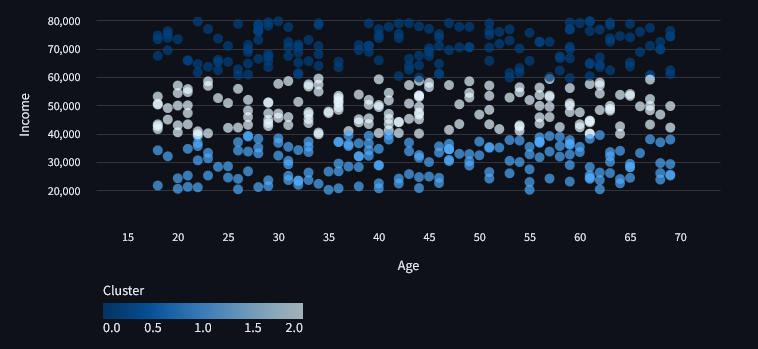
**5. Statistical Analysis: Monthly Sales**

This section aggregates monthly sales by product category using pandas groupby. The result is visualized using a Streamlit line chart, allowing users to track sales evolution over time.



**6. Clustering: Customer Segmentation**

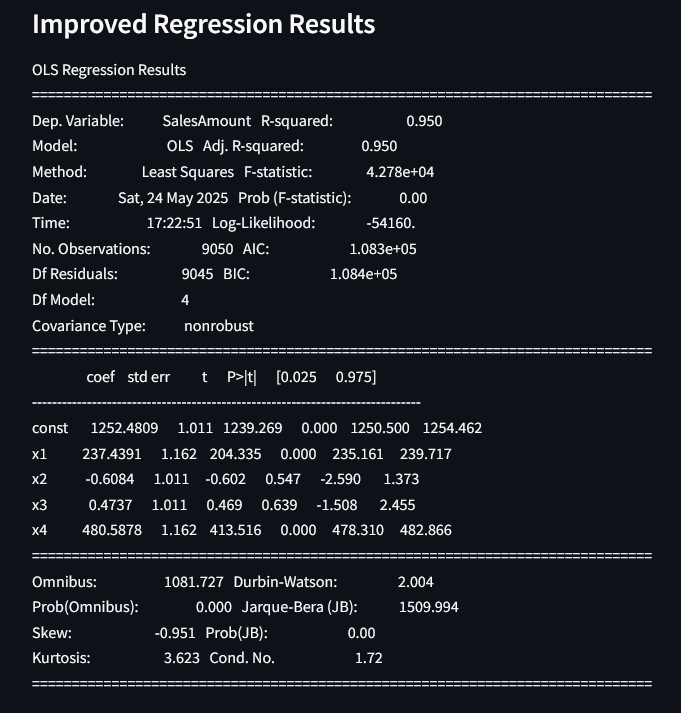
We applied K-Means clustering to group customers based on age and income. The results are visualized using a scatter plot, helping the business identify core customer segments.



**7. Regression Analysis**

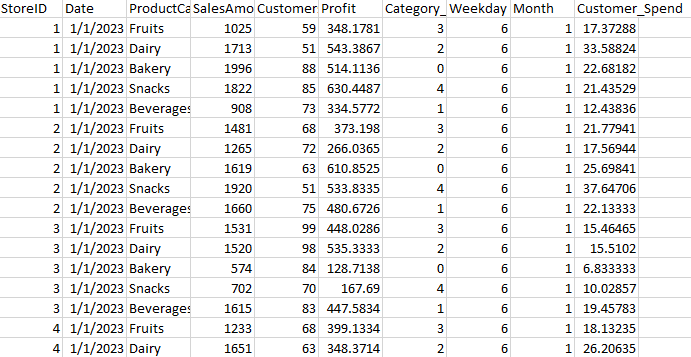
Using statsmodels, we ran a multiple linear regression to predict sales based on the number of customers, product category, weekday, and average customer spend.

Features were scaled before regression. The regression summary shows the statistical relevance of each factor.



**8. Download Cleaned Data**

To facilitate further analysis, a download button was provided. This allows the user to export the cleaned and enriched sales data as a CSV file directly from the app.  
  
Example of cleaned data:



**Conclusion**

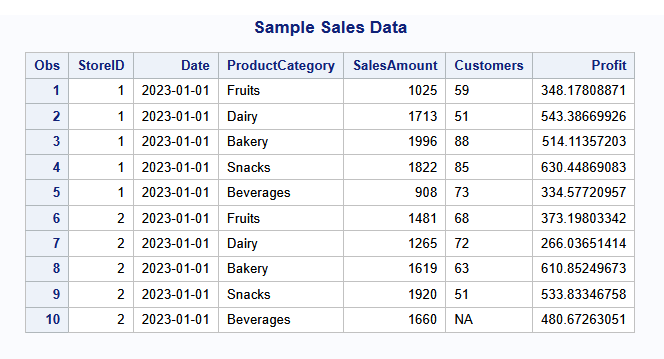
The GreenGrocer Analytics Dashboard enables dynamic exploration and analysis of sales performance, store distribution, customer behavior, and sales drivers. It is designed to support data-driven decisions for expansion and operational improvements.

**GreenGrocer Analytics – SAS Programming**

This section documents the SAS-based analysis for the GreenGrocer retail dataset. The analysis includes data import, formatting, statistical processing, SQL joins, array processing, and graphical visualization. Each task is aligned with the course requirements and implemented using SAS Studio.

**1. Import Sales Data**

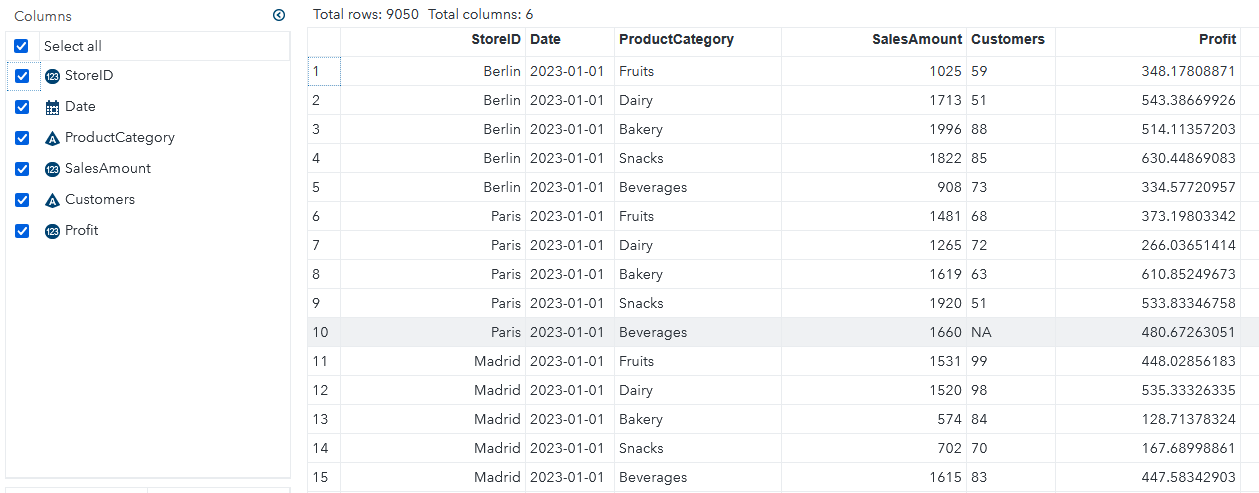
* **Problem**: Import the raw sales data into the SAS environment for analysis.
* **Information Required**: A CSV file (greengrocer\_sales.csv) with columns like StoreID, Date, SalesAmount, Profit, etc.
* **Method**: Used PROC IMPORT to load the file and PROC PRINT to view the first 10 rows.
* **Results**: The dataset was successfully imported and verified for structure.



* **Interpretation**: Having structured data in SAS enables further statistical and analytical processing.

**2. Create User-Defined Formats**

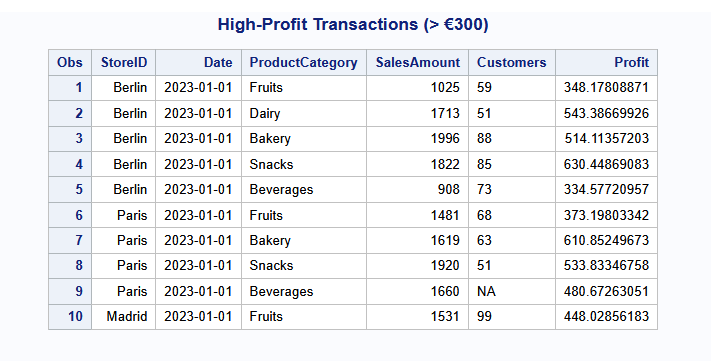
* **Problem**: Make StoreID values more readable by associating them with city names.
* **Information Required**: Mapping between store IDs and city names.
* **Method**: Defined a custom format using PROC FORMAT, then applied it in a DATA step.
* **Results**: StoreID values now display as city names.



* **Interpretation**: Enhances clarity for reporting and analysis across geographic dimensions.

**3. High-Profit Subset (Conditional Processing)**

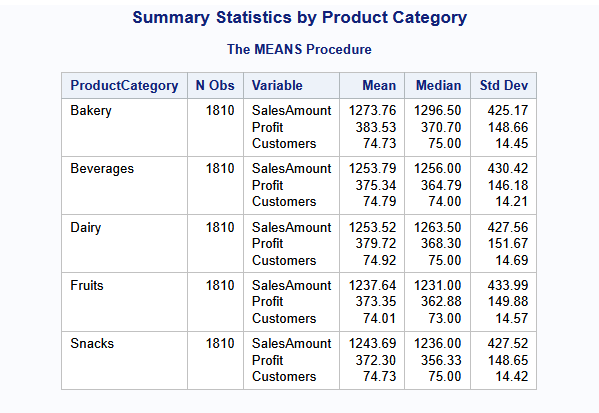
* **Problem**: Identify high-value transactions with profit greater than €300.
* **Information Required**: Profit column from the sales dataset.
* **Method**: Used an IF condition in a DATA step to filter records.
* **Results**: High-profit rows printed with PROC PRINT.



* **Interpretation**: Helps identify key revenue-driving sales or promotions.

**4. Statistical Summary by Category**

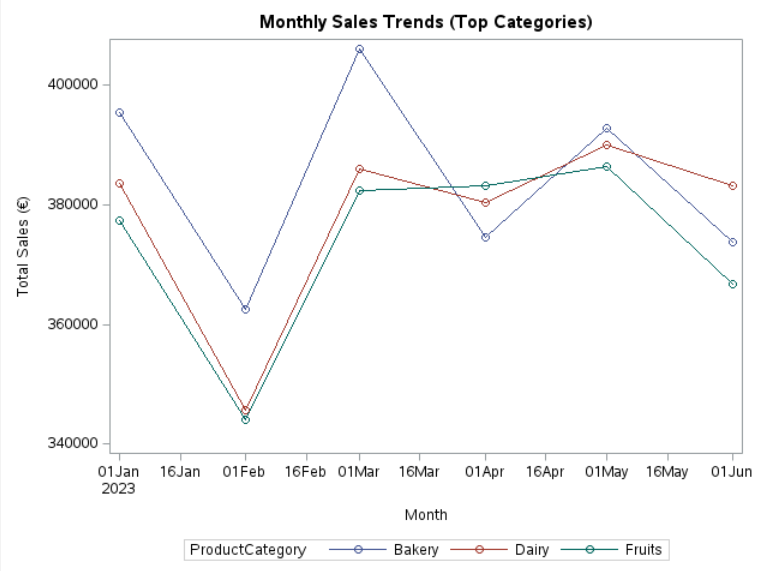
* **Problem**: Summarize performance metrics (e.g., average profit) for each product category.
* **Information Required**: Variables ProductCategory, SalesAmount, Profit, Customers.
* **Method**: Used PROC MEANS with CLASS and VAR.
* **Results**: Report includes mean, median, and standard deviation.



* **Interpretation**: Supports strategic decisions on which categories perform best.

**5. Monthly Sales Graph**

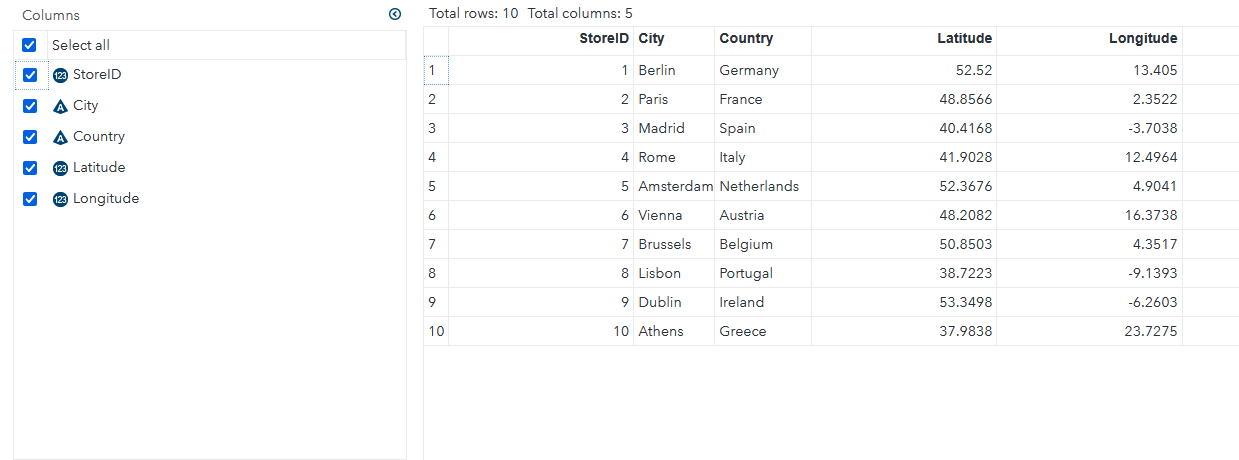
* **Problem**: Visualize monthly sales trends per category.
* **Information Required**: Date, ProductCategory, SalesAmount.
* **Method**: Used PROC SQL to group sales, and PROC SGPLOT to draw a line chart.
* **Results**: Line plot showing trends for top product categories.



* **Interpretation**: Reveals seasonal performance trends and helps forecast demand.

**6. Merge Sales with Store Locations**

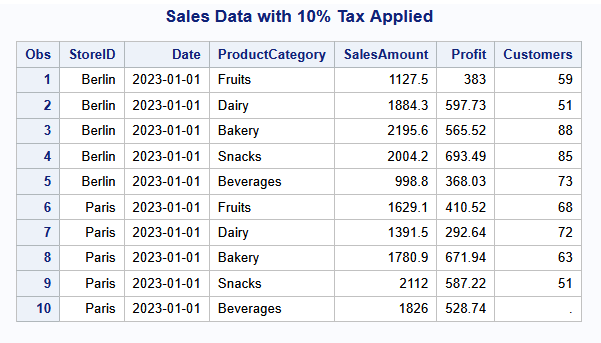
* **Problem**: Add geographic context to the sales data.
* **Information Required**: StoreID to join sales\_fmt with locations.csv.
* **Method**: Used PROC IMPORT, PROC SORT, and PROC SQL for the merge.
* **Results**: Combined dataset with fields like City, Country, Latitude, Longitude.



* **Interpretation**: Enables region-based sales analysis and mapping.

**7. Apply 10% Tax (Arrays)**

* **Problem**: Simulate post-tax values for both SalesAmount and Profit.
* **Information Required**: SalesAmount, Profit fields.
* **Method**: Used an ARRAY in a DATA step to apply 10% increase.
* **Results**: Printed 10 rows of tax-adjusted sales.



* **Interpretation**: Shows how tax policy affects revenue and margin.

**8. Sales and Profit by Country (SQL Summary)**

* **Problem**: Compute total sales and profit per country.
* **Information Required**: Country, SalesAmount, Profit fields from the merged dataset.
* **Method**: Used PROC SQL with GROUP BY and ORDER BY.
* **Results**: Table sorted by highest total sales.



* **Interpretation**: Highlights the top-performing countries for strategic investment or marketing focus.