Emilio Sanchez San Martin

Project Title: Enhancing Alcohol Inventory Reordering for ABS Using Custom Algorithms

Course: DATA 205 – Capstone Experience in Data Science

CRN: 33334

# Introduction/Overview

Montgomery County's Alcohol and Beverage Services (ABS) Retail Division has been operating 37 stores in the county, which had a balance of different tradeoffs of overstocking and stockouts for their alcohol products. Excess inventory leads to the county's tied-up funds and could lead to waste, while understocking may result in missed sales that would leave customers unable to purchase products they wish for. ABS currently uses its own Master Planning Algorithm system to help calculate inventory thresholds such as the Minimum Shelf Stock (MSS) and the Reorder Amount in bottles for each product sold in store. These values help ABS place orders and ensure stores have well-stocked products for people to purchase.

The algorithm ABS uses considers different factors, such as the daily sales across a three-time-period window, and takes minimum thresholds. Their system, however, lacks other dynamics that are as important, like delivery lead times, delivery frequency, and on-sale item states that can be a limitation, considering their impact on inventory. The ABS IT team wants to explore if their algorithm could be refined or redesigned to reflect operational realities better and improve their efficiency across stores with many different volumes and footprints.

## 

## Goals

My goal is to develop a smarter, responsive inventory algorithm that better balances product availability with inventory efficiency. Using data collected from 2024 weekly sales, I compared their existing ABS algorithm logic with another customized alternative that they could consider, evaluating the effect on performance and how they could look at different stores from high-volume, mid-volume, and low-volume stores. The goal is to reduce excess inventory and keep track of missed sales by producing a productive model that could serve the ABS team, helping them calculate MSS and reorder quantities for this. This is important to avoid waste due to overstocking, prevent loss of revenue from stockouts, and adapt more closely to store factors.

## Data

The dataset provided by ABS includes detailed weekly sales for the year 2024 from three stores: one high-volume (two weekly deliveries), one mid-volume (with one delivery), and one low-volume (one delivery). Each store's data set includes 500 unique rows of products with 59 volumes initially. These include Item ID, Description (Product Name), Bottles Per Case, Cost Amount (Per Bottle), Total Cost (Calculated as Bottles Per Case x Cost Amount), Weekly Sales Figures for Weeks 1-53, and Grand Total. During data cleaning, variables were more standardized (replacing spaces with underscores), and weekly columns were renamed to Week\_#. Zero NAs. Data cleaning was performed using **OpenRefine** and **RStudio**, documented in the GitHub repository.

## Tools and Techniques

This project was completed using **RStudio** for data processing, statistical analysis, and visualizations; **OpenRefine** for initial data cleaning and standardization; the libraries **tidyverse, ggplot2, plotly, zoo,** and **stringr** for weekly sales visualizations and statistical methods; and **statistical testing (McNemar’s Test)** for comparing performance across models. By combining exploratory analysis with my logic and testing, the project offers a support tool to help ABS rethink how inventory thresholds are set.

# Data Cleaning

To begin making an improved algorithm for ABS, I first had to clean and prepare weekly sales data for the different types of stores I worked with: high-volume, mid-volume, and low-volume. Every dataset had information for 53 weeks of 2024 and variables such as ItemID, Description, Bottles\_Per\_Case, etc.

Next, I implemented the ABS's current Master Planning Algorithm in RStudio to calculate the two main metrics per product. The minimum shelf stock and reorder quantity. I also added further processing to those steps, grouping weekly sales into 15-week windows to split into 5-week intervals. Also tagged products as "Beer" or "Other" to account for different reorder threshold periods (14 days = beer, 10 days = other). I then calculated daily averages, standard deviation, and combined averages across 5-week periods. I used the processed data to track algorithm performance to get variables: **MSS\_Weeks**: Minimum Shelf Stock values calculated weekly (Weeks 16–53), **Reorder\_Weeks**: Bottle-level Reorder Quantities calculated weekly (Weeks 16–53), and **Reorder\_Cases**: Rounded-up full case quantities converted from bottle-level reorders. Finally, I implemented three different core R functions for each, ensuring that the algorithm could adapt to changing sales patterns and look at accuracy.

# Basic Descriptive Statistics: Summary of Dataset and Key Columns

As said before, my data set consisted of many weekly sales data from 1 to 53 of the year 2024. It has used the past 15 weeks to calculate a reorder quantity and MSS. The total amount of a sold product per week, the MSS (Week 16 to Week 53): acting as the buffer stock target for each product based on demand and product category; Reorder Quantity (Week 16 to Week 53): total amount to order when stock falls short; and Reorder Cases: converting reorder quantities to full cases for ABS's use.

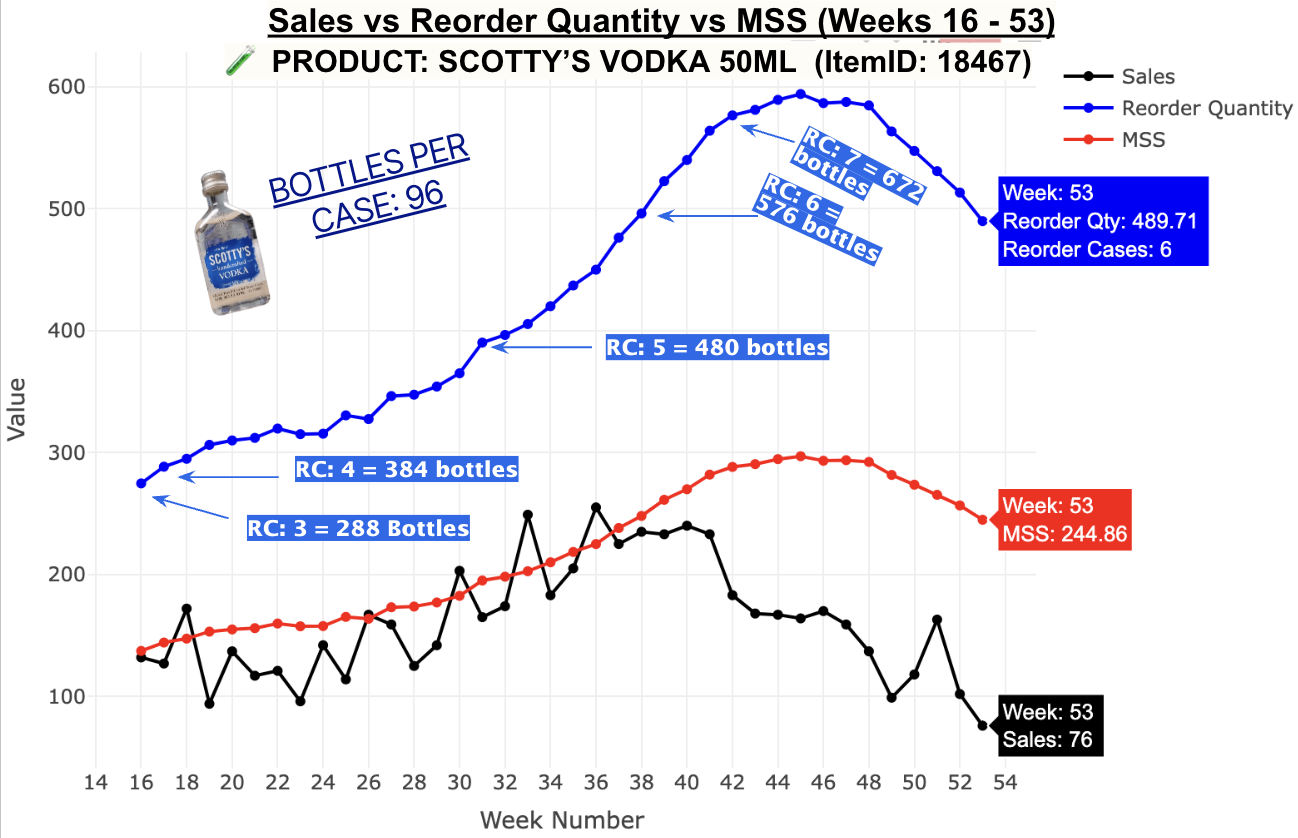
## Product Analyses:

### 1. SCOTTY’S VODKA 50 ML (ItemID: 18467)

- Pattern: Strong sales from Weeks 16–36. Sharp decline from Week 42 onward.

- Key Insight: During the high-sales period, sales consistently kept up with MSS. Accurate demand. Later, when sales dropped, the algorithm continued calculating high MSS and reorder quantities, leading to potential overstocking.

- Interpretation: The algorithm was slow in capturing the decline of sales due to its reliance on older averages (PD1), showing a need for better responsiveness.



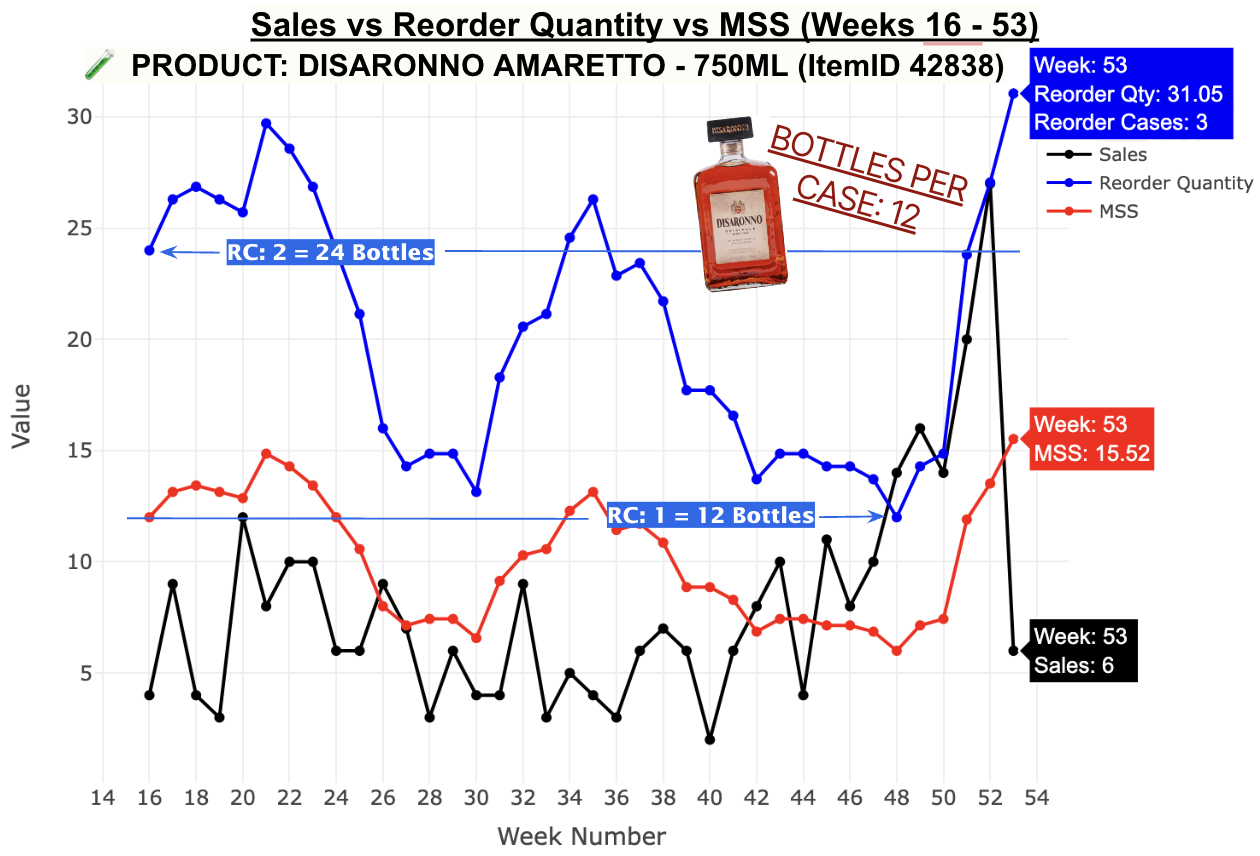
### 2. DISARONNO AMARETTO 750ML

- Pattern: Sales vary throughout the year:

* Weeks 32–38 saw sharp sales drops but slow algorithmic adjustment. Weeks 45–52 saw strong spikes in demand, with sales exceeding MSS and reorder levels.

- Key Insight: The algorithm could not capture demand drops (risk of overstock) and demand spikes (risk of stockouts).

- Interpretation: Showed limitations of underreacting and overreacting (mostly underreacting), showing a need for better adaptive inventory logic, especially for products like these.



Overall, products behave differently over time, and the ABS algorithm has shown that it may not be able to handle the rapid change in volatility effectively of different types of products. It makes reacting to high demands too lightly or demand surges too softly, causing overstock or stockouts at certain points of the week. These observations are important and good to keep in mind to focus on improving the algorithm.

# Description of Final Data Product

My final data product shows my own customized inventory optimization algorithm designed for ABS. The reorder quantities and MSS are calculated based on different factors, building on ABS's original 15-week sliding window approach with a dynamic responsiveness to sales behavior and using techniques such as **weighted averages**, **trend detection**, and the **Coefficient of Variation (CV)**. I have also done a set of visualizations and metrics to look at the performance between my new model and the ABS algorithm.

## Brief Progress Explained:

CV gets calculated from 3 sub-periods (default 5-5-5 weeks). If CV is over 0.25 or huge reversal patterns are detected (up-down-up), the algorithm flags the trend of instability, switches to a 5-4-3 structure to emphasize recent sales, and applies more weight to recent weeks (20%, 30%, 50%). This allows the weighted average to also be more predictive in capturing sales from recent weeks.

# Evaluation Metrics:

Stockout Risk (% of weeks when sales exceed the MSS for risk of missed sales)

Average Excess Stock (Average weekly amount of product sitting on shelves longer than needed)

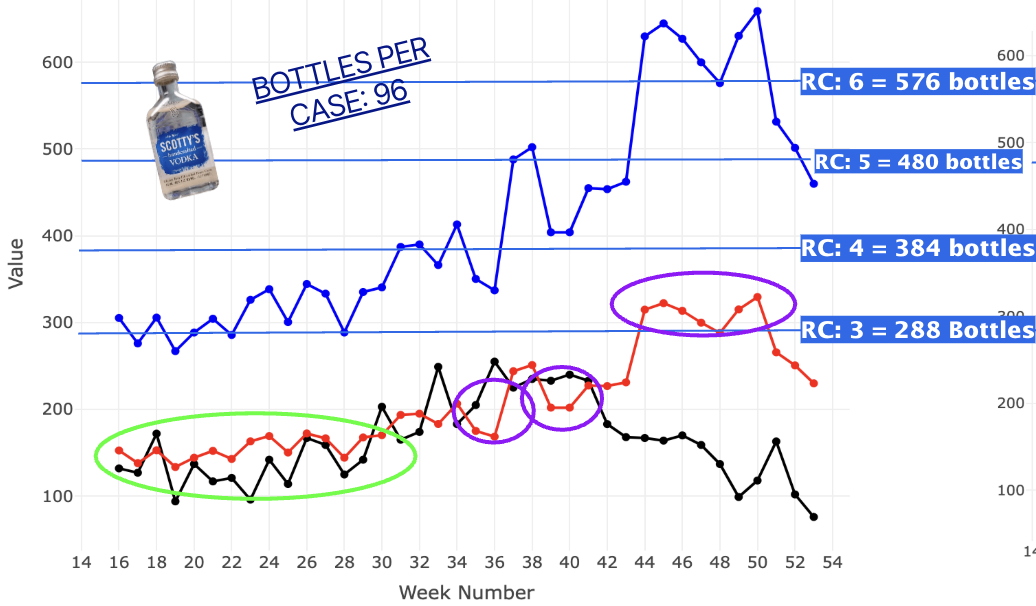
Reorder Volatility (Weekly changes in reorder quantity to see predictability)

MSS Fit Categories (Breaking down how well MSS matched sales)

# Results & Interpretation of My Algorithm

**PRODUCT: SCOTTY’S VODKA 50 ML (ItemID: 18467)**

Sales vs Reorder Quantity vs MSS (Weeks 16 - 53)

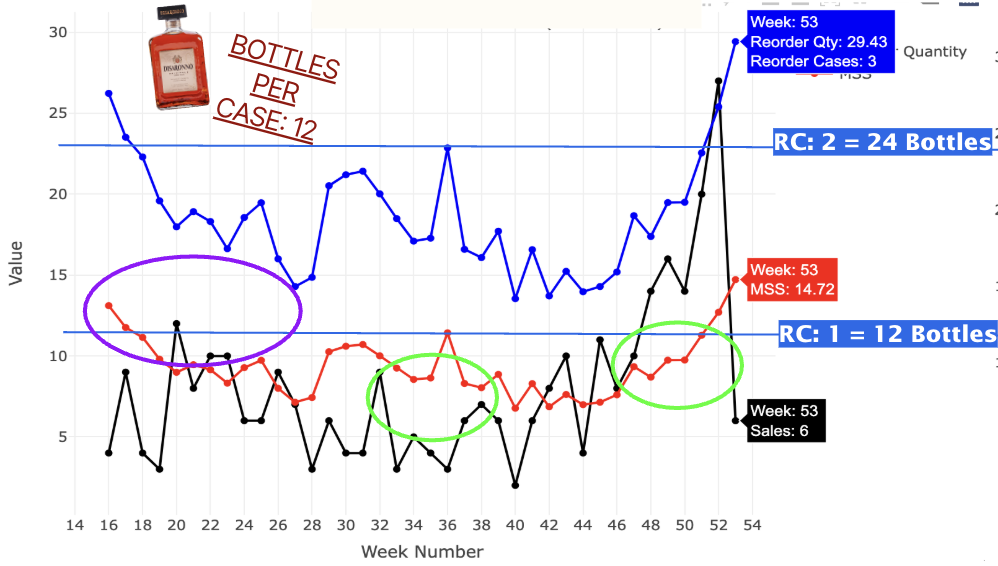


| **Metric** | **Mine** | **ABS** | **Insight** |
| --- | --- | --- | --- |
| **Stockout Risk** | 21.05% | 13.16% | My algorithm has a higher stockout risk |
| **Avg Excess Stock** | 55.62 | 60.45 | Slightly more efficient inventory use |
| **Reorder Volatility** | 120.94 | 111.13 | My reorders fluctuate more |
| **Good Fit %** | 18.4% (Tie) | 18.4% | Both align with demand equally well |
| **MSS Too Low% %** | 18.4% | 10.5% | More frequent understocking with my model |

My algorithm reduces excess inventory but increases stockout risk and variability, suggesting a need for more safety stock when an item goes high in sales.

**PRODUCT: DISARONNO AMARETTO - 750ML (ItemID 42838)**

Sales vs Reorder Quantity vs MSS (Weeks 16 - 53)

****

| **Metric** | **Mine** | **ABS** | **Insight** |
| --- | --- | --- | --- |
| **Stockout Risk** | 38.84% | 28.95% | Continued stockout risk |
| **Avg Excess Stock** | 2.77 | 3.68 | Less shelf waste |
| **Reorder Volatility** | 3.65 | 5.62 | My orders are a bit more consistent |
| **Good Fit %** | 10.5% | 5.3% | Slightly better balance |
| **MSS Too High %** | 60.5% | 65.8% | Less overstocking |

For more volatile products, my algorithm improves efficiency and reorder consistency, but can be too aggressive in cutting stock, as we saw in the beginning, leading to a risk of stockouts.

# Conclusion

In conclusion, this project tried to create an inventory algorithm that could improve ABS Stores' reorder quantities and MSS, and the results showed potential in some areas, while other areas could lack. Comparing the original ABS model with my approach, I highlighted some places for improvement to manage stockouts and stabilize reorder patterns on specific periods of the week, but there are some differences in terms of its responsiveness with different products. I wish to have the next time use more sets of products that are similar in sales patterns and test to see how the algorithm would respond, but that would take an extremely long time to process all 500 products for each store. I would also suggest refining the performance of the algorithm with more instability measures to target adjustments on certain weeks to create a more adaptable system for ABS retail stores.

# References and Acknowledgements

This project would not have been possible without the support and guidance from many different individuals and educators who were with me on my journey. I would first like to thank my Montgomery College data professors for their strong guidance in providing a strong foundation in data science and analytics.

A huge thank-you note to Professor Rachel Saidi, who introduced me initially to data science and the way the field works since the very beginning. I would also like to thank Professor Mais Alraee, who has also helped me since the beginning with programming in RStudio. I am also extremely grateful to Professor Lori Perine for her continued insightful feedback and encouragement throughout the whole project. Her guidance throughout the whole project helped me a lot in shaping my approach and analysis for this project.

Thank you as well to the mentors and staff at Montgomery College and Montgomery County Alcohol and Beverage Services (ABS)—Victoria Liu, Kori Wenman, Le Guellec, and Naveen Gattoju—for your time, insights, and support. Your perspectives and feedback on the planning for our algorithm and data access were critical to the success of this project.

Finally, I want to recognize a student who has been a part of the journey with me, Emilio Difilippantonio, who also worked alongside me on this project with ABS. His collaboration and creativity have also been helpful throughout this project and made this project more engaging and a rewarding experience.