Capstone Project Presentation

🏢 **Montgomery Alcohol Beverage Services Government Project**

Emilio Sanchez San Martin

Data 205 | Spring Semester 2025

Prof. Perine

OVERVIEW / GOAL

OVERALL PROBLEM: Stores in MOCO County have either <u>too much</u> (🥴) or <u>too little</u> (😫) inventory of alcohol.

TOO MUCH = ***Waste!***

TOO LITTLE = ***Lost sales!***

- **Goal🎯: Making a BETTER inventory algorithm to find JUST the right amount of stock level for each product!**
- **Solution 💡 :** *Using* **weekly 2024 data** to explore **patterns and build logic from HIGH, MEDIUM, and LOW volume stores!**
  - *Ensuring stores for ABS maintain the* ***<u>BEST</u>*** *stock levels*

ABS
ALCOHOL BEVERAGE SERVICES
A BUSINESS OF MONTGOMERY COUNTY GOVERNMENT

The Data Cleaning Process - FIRST starting off!

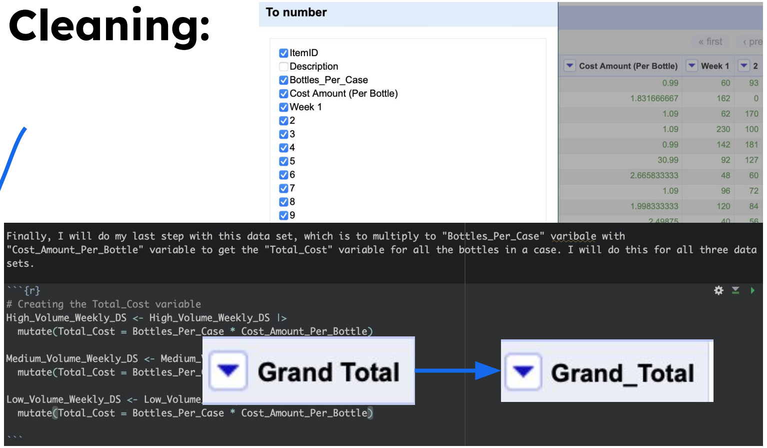
📦 Three datasets: High, Medium, & Low Volume Stores

ItemID	Description *(product name)*
Bottles Per Case	Cost Amount (Per Bottle)
*Total Cost (NEW) (Calculation of bottles per case * cost amount)	*WEEKS (1-53) Year 2024 Grand Total

- Renamed each number of the week to "Week #"
- Added "_" for every space in the variable.
- No NA's
- Multiplied **Bottles Per Case** ✖ **Cost Amount (Per Bottle)** for Total_Cost (Per case)

🧹 Data Cleaning:

Sources: OpenRefine 💎 RStudio



- 500 Rows | 59 columns
- Sales Data for each **week**

Implementing Algorithm!

📦 **EVERY volume store dataset** gained additional variables **AFTER** implementing ABS Algorithm.

🔍 Step 1: Collect Sales Data

Sales for each product are grouped into three 5 week periods (15 weeks total) each store

📈 Step 2: Calculating Sales Trends

For each item:

- Daily averages (each 5-week period) = calculated
- Combined average = computed 🧮
- Standard deviation (📏 STDev) need to see consistency

📦 Step 3: Set Minimum Shelf Stock (MSS)

- If total sales < 1 case ➡️ MSS = 0
- If sales > 1 case:
- Use PD1 or Combined Avg depending on STDev
- Multiply chosen avg by the Reorder Threshold

(🍺 Beer = 14 days, 🥃 Others = 10 days)

Step 4: Calculate Reorder Quantity

- Daily avg × Lead Time (10 days) + MSS = Reorder Qty
- 📦 Reorder Qty is then rounded to full cases

LIMITATIONS:

- No data on inventory to really measure ABS Stores' capacity.
- Some products are stable in sales; others DON'T (the key)
- Using weekly data, but daily would be beneficial to see sales (would, however, take forever; that's why we use weekly!)

Continue:

```r
beer_brands <- c("CORONA","STELLA", "HEINEKEN", "MODELO","BLUE MOON", "GUINNESS", "SAPPORO", "MICHELOB", "SAM ADAMS", "NEGRA
MODELO", "PERONI", "PILSNER URQUELL", "FLYING DOG", "LEFFE", "ASAHI", "DC BRAU", "DOS EQUIS", "SIERRA NEVADA", "TSINGTAO", "RED
STRIPE", "ALLAGASH", "DENIZENS", "SINGHA", "BEER FARM", "DOGFISH HEAD SLIGHTLY MIGHTY LO-CAL IPA 4/6PK", "DOGFISH HEAD SEAQUENCH
ALE 4/6 CAN", "DOGFISH HEAD PUNKIN ALE 4/6 CAN", "DOGFISH HEAD IPA 2/12 VP CANS", "DOGFISH HEAD HAZY SQUALL 4/6 CAN", "DOGFISH
HEAD FESTINA PECHE 4/6 CN", "DOGFISH HEAD 90 MINUTE IMPERIAL IPA 4/6 NR", "DOGFISH HEAD 60 MIN IPA 4/6 NR - 12OZ", "DOGFISH HEAD
(SUMMER) VP 2/12PK CAN", "DOGFISH HEAD (FALL) VP 2/12", "NEW BELGIUM")
# I will use the beer brands to create a new column called "Category" and put the category of the product in there.
High_Volume_Weekly_DS$Category <- ifelse(grepl(paste(beer_brands, collapse = "|"), High_Volume_Weekly_DS$Description,
ignore.case = TRUE), "Beer", "Other")
Medium_Volume_Weekly_DS$Category <- ifelse(grepl(paste(beer_brands, collapse = "|"), Medium_Volume_Weekly_DS$Description,
ignore.case = TRUE), "Beer", "Other")
Low_Volume_Weekly_DS$Category <- ifelse(grepl(paste(beer_brands, collapse = "|"), Low_Volume_Weekly_DS$Description, ignore.case
= TRUE), "Beer", "Other")
```

Made sure products were categorized by "Beer" for different Reorder Treshold:

- Beer: 14 days 🍻
- Other: 10 days 🥃

This directly impacts **MSS** and **Reorder Qty** calculations.

Gotten functions:

🧠 **Reorder Quantity Algorithm** (Function: calculate_reorder_sliding) = Master Planning Process

🧊 **Minimum Shelf Stock** (Function: calculate_mss_sliding) = Same for MSS (buffer stock)

📦 **Reorder in Cases** (Function: convert_reorder_to_cases)

- ABS Stores reorders **in full cases** (not bottles).
- Converts bottle reorder quantity to **rounded-up case amounts.**

```
ceiling(reorder_qty / Bottles_Per_Case)
```

- 500 Rows | columns

NEW VARIABLES

Reorder_Weeks(16-53)	**MSS_Weeks(16-53)**	**Reorder Cases(16-53)**

LET'S BREAK THE ALGORITHM THROUGH!

The three datasets for ABS **need to fulfill requirements** (depending on the unique store)

1. High-volume, small footprint, two deliveries per week

- **High sales** but **limited storage space**.
- Needs more frequent restocking (hence has two deliveries).
- May need to keep lower inventory levels but reorder more often to avoid running out.

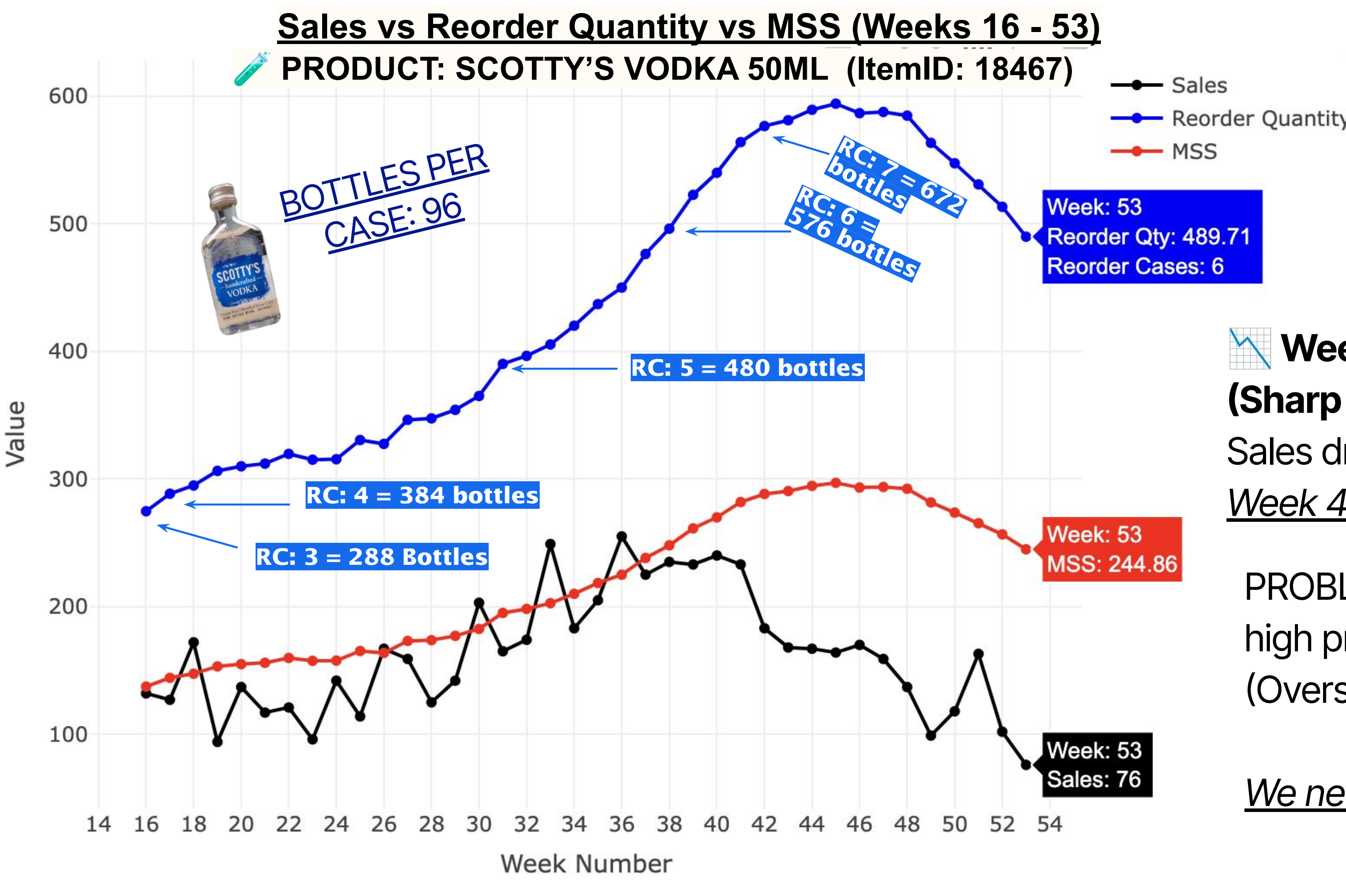
2. Mid-volume, large footprint, one delivery per week

- **Decent sales** with **plenty of storage**.
- Can hold more backup inventory.
- May tolerate slightly larger reorder quantities less frequently.

3. Low-volume, small footprint, one delivery per week

- **Lower sales** and **limited space.**
- Doesn't need to hold much but can't store much either.
- Reorder quantities should be tight and accurate to avoid overstock

HIGH STORE VOLUME: Results of Algorithm (OVERSTOCK)



📊 Weeks 16–36 (High Sales Period)

<u>Week 16, 30, 36 SALES > MSS.</u>

(Product moving fast those periods, triggering MSS)

📉 Weeks 42–53 (Sharp Sales Decline)

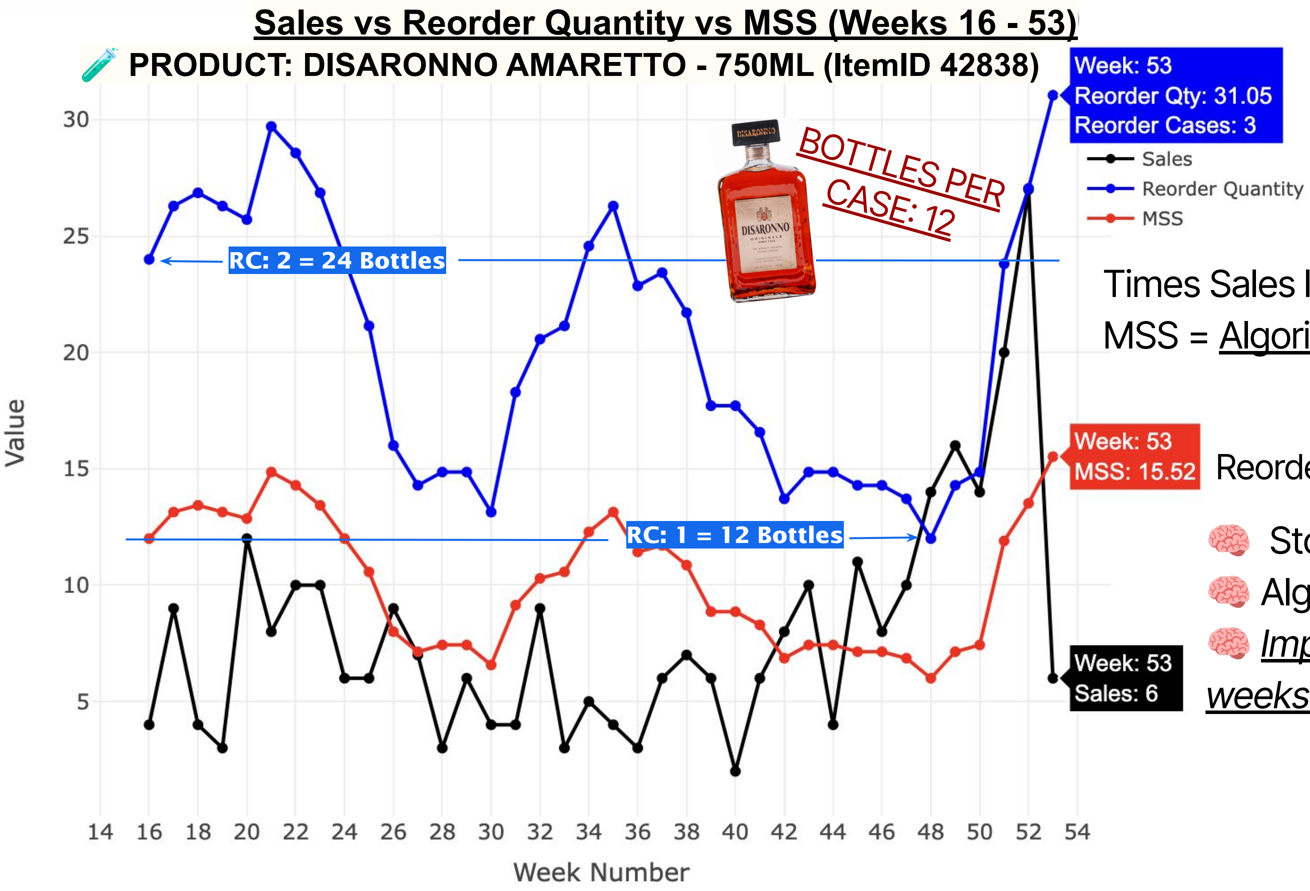
Sales drop **ALOT** below MSS.

<u>Week 42 & Week 53: LOW SALES ⚠</u>

PROBLEM: The algorithm uses high previous sales, but it declines! (Overstocking)

<u>We need to adapt to sales drops.</u>

HIGH STORE VOLUME: Results of Algorithm (UNDERSTOCK)



🟠 **SALES PATTERN:**

- SALES are NOT Stable, Spikes randomly (Sales low/high), algorithm is not adapting to change well

Times Sales lower or higher than MSS = Algorithm slow to respond

Weeks 45 - 52: **Sales explode!**

Reorder quantity got touched/exceeded!!

🧠 Store running close to stockouts.

🧠 Algorithm reacts but lags a bit

🧠 *Implementing more weight to recent weeks would help!*

POSSIBLE SOLUTION! 💡

🔁 PROBLEM RECAP:

- 📉 Sales drop fast (weeks 42–53) | MSS and reorder quantities stay high.
- 📈 Sales spike in certain weeks (weeks 42 - 52) | System doesn't respond fast enough.

✅ GOAL: Making the algorithm adapt quickly to changing sales trends!

MSS and Sales closely align, but

- MSS should be slightly above Sales (safety buffer).
- MSS should not lag behind if Sales are increasing quickly.
- MSS should drop fast when Sales go down (to prevent overstocking).

🔧 HOW TO IMPROVE THE ALGORITHM:

FIRST IDEA: Weight Averages! Giving MORE IMPORTANCE to Recent Trends

🧠 Logic: Weighting periods 3 (more recent), 2 a bit less, and 1 the least.

Something like:

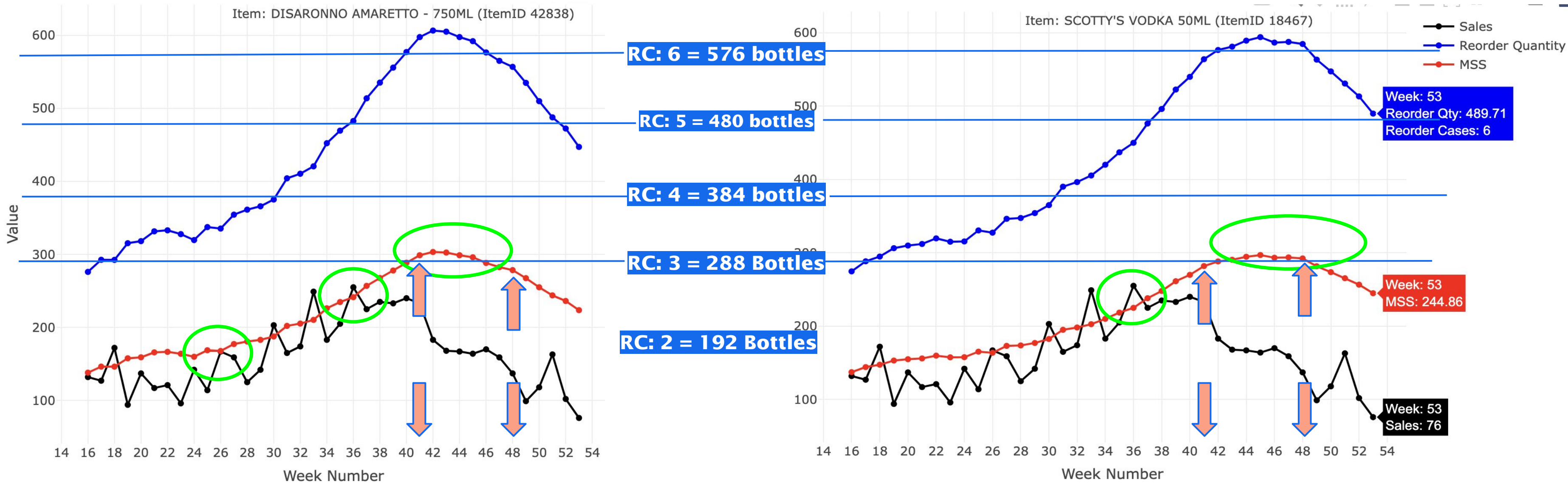
```
weighted_avg <- (avg1 * 0.2 + avg2 * 0.3 + avg3 * 0.5)
```

SECOND IDEA: Coefficient Variation! This is a statistical measure to track variability for products sold throughout the week.

CV = Standard Deviation of PD1, PD2, PD3/Mean of PD1, PD2, PD3

- If CV > 0.3 (or another threshold), switch to **shorter periods.**

Implementing FIRST IDEA ONLY



- Small difference
- MSS typically goes HIGHER When needed, which is good to capture higher reorder quantity. But…
- *It does not take MSS down when needed* **(Needs to track instability)**

My Algorithm VS ABS

🔄 Enhancing ABS's Reorder Algorithm: Smarter, More Adaptive Predictions

✅ Key Improvements

- 📊 **Adaptive Periods:**
  - Original: Fixed 5-5-5 week splits
  - Mine: Adjusts to 5-4-3 if sales are volatile (using **coefficient of variation** or trend reversals)
    → Responds better to shifting demand patterns
- 📉 **Instability Detection:**
  - Mine flags unstable patterns (up-down-up)
    → Improves predictions for erratic or seasonal items

```
unstable <- (cv > 0.25) || ((avg1 < avg2 & avg2 > avg3) | (avg1 >
avg2 & avg2 < avg3))
```

- ⚖️ **Weighted Averages:**
  - Original: Simple average or avg1
  - Mine: Weighted (20%-30%-50%) toward recent sales
    → Captures trends without overreacting to short-term noise
- 📦 **Reorder Logic:**
  - Same base formula, but my algorithm gives **smarter averages**
    → More accurate MSS and reorder quantities
    → Reduces overstock + stockouts

Explanation of metrics

🔍 Metric Breakdown – Quick Definitions

- 📉 **Stockout Risk**
  % of weeks where sales exceeded MSS → risk of running out (When MSS Red line is over Black Sales line)
  → Lower = better product availability

- 📦 **Avg Excess Stock**
  Avg # of units above what was actually sold
  → Lower = better shelf space use & cost efficiency

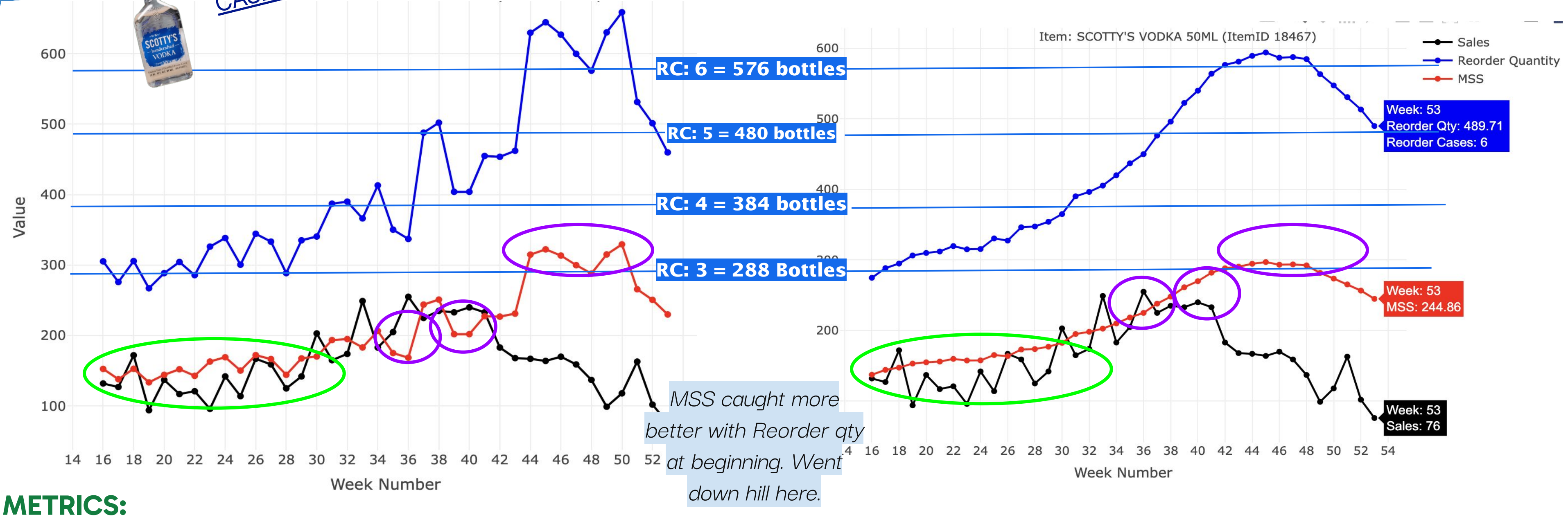
- 🔄 **Reorder Volatility**
  Measures how much reorder quantities change week to week
  → Lower = smoother operations, easier planning

- ⚖️ **MSS Fit Breakdown**
  - **Good Fit**: MSS ≈ Sales
    → balanced inventory
  - **Too Low**: MSS < Sales
    → understock risk
  - **Too High**: MSS > Sales
    → overstock waste
    → Aim for more Good Fit weeks

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

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

BOTTLES PER CASE: 96



METRICS:

⚠ Stockout Risk

- Mine: 21.1% | ABS: 13.2%
- My model is more prone to understocking, increasing the risk of stockouts.

📦 Avg Excess Stock

- Mine: 55.6 | ABS: 60.5
- My algorithm is slightly more efficient with shelf space and storage.

⚖️ Balance (MSS Fit)

- ✅ Good Fit: 18.4% (tie for both)
- 📈 Too High: 63.2% (Mine) vs. 71.1% (ABS)

→ My algorithm slightly reduces overstocking.

- 🛑 Too Low: 18.4% (Mine) vs. 10.5% (ABS)

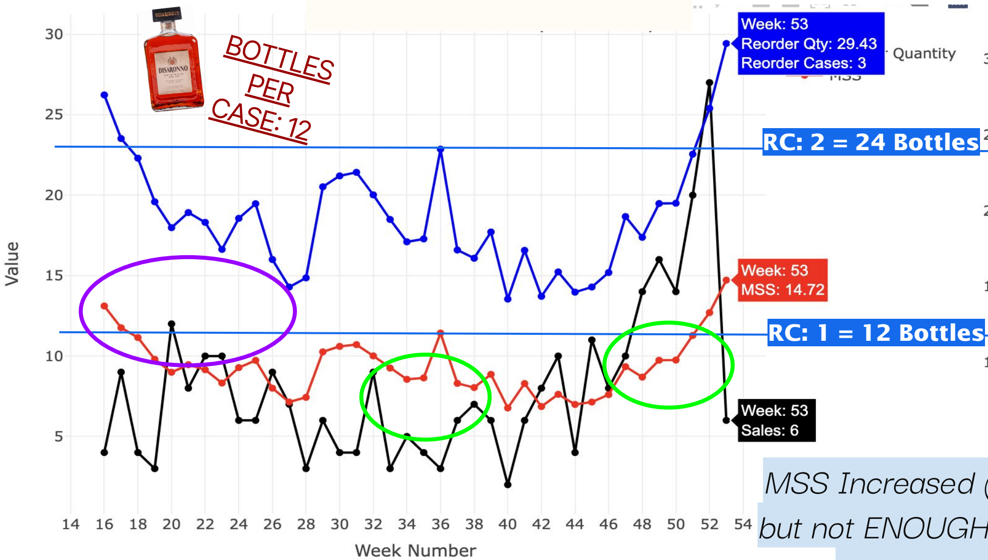
→ ABS has better protection against demand spikes.

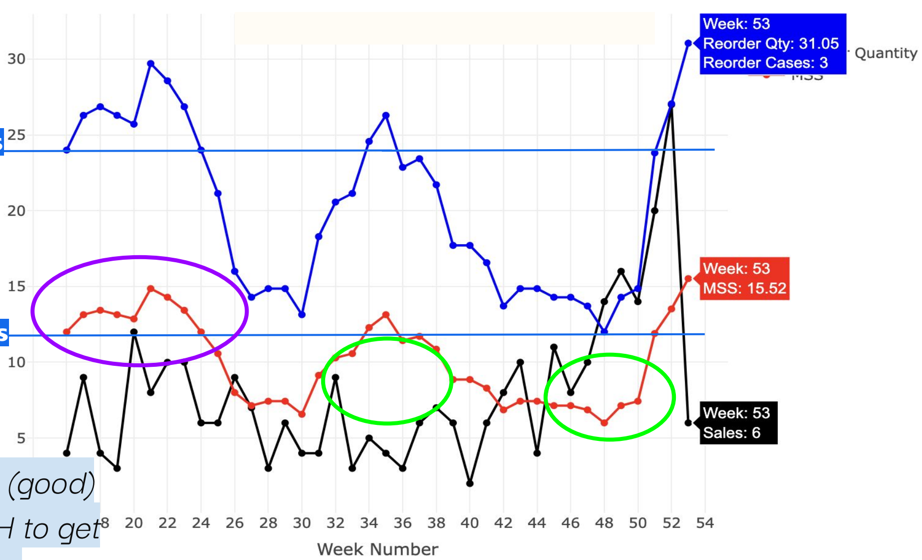
🔄 Reorder Volatility

- Mine: 120.9 | ABS: 111.1
- My reorder amounts are more erratic, which may cause planning issues for future algorithms

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

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





MSS Increased (good) but not ENOUGH to get more cases!

METRICS:

⚠ Stockout Risk

- Mine: 36.8% | ABS: 28.9%
- My algorithm is more aggressive, increasing the risk of running out.

📦 Avg Excess Stock

- Mine: 2.77 | ABS: 3.68
- I hold less unnecessary inventory, which is efficient.

🔄 Reorder Volatility

- Mine: 3.65 | ABS: 5.62
- My orders are more stable, reducing operational disruptions.

⚖️ MSS Fit Breakdown

- ✅ Good Fit: 10.5% (Mine) vs. 5.3% (ABS)

→ I slightly outperform ABS on matching MSS to sales.

- 📈 Too High: 60.5% (Mine) vs. 65.8% (ABS)

→ I'm slightly better at avoiding overstocking.

- 🛑 Too Low: 28.9% (both)

→ Same understock risk for both algorithms.

Conclusion/Experience

I started off with a simple question:
Can we build a smarter inventory algorithm to help ABS stores avoid overstock and stockouts?

This project explored two algorithms — the ABS baseline and my customized model — to optimize Minimum Shelf Stock (MSS) and Reorder Quantities for alcohol inventory across different store types.

Insights:
•📉 Stockout Risk: Both algorithms agreed most of the time, with few minor differences on reorder quantities to catch up with sales based on two different types of products.
•📦 Excess Inventory: At some points, there was just enough inventory. But in
•⚖️ MSS Fit: Continued to keep inventory close to sales
•🔁 Reorder Stability: Weighted averages helped smooth out overreactions and cut down on inconsistent ordering.

- While neither algorithm is perfect, my version may offer a customizable alternative that can be fine-tuned by product type or store profile, because it can definitely be considered when looking at oversold products, as we've seen with Disaronno Amaretto.
- Next steps could include testing different thresholds, adding external factors like promotions, and automating MSS updates for responsiveness based on several years of data.

References & Acknowledgements

Montgomery College Data Professors

- **Rachel Saidi** – For introducing me to data science and laying a strong foundation from the beginning.
- **Lori Perine** – For her continued guidance, support, and feedback throughout this project.

Mentors at Montgomery College & Montgomery County Alcohol and Beverage Services (ABS)

- **Victoria Liu**
  Kori Wenman
- **Le Guellec**
- **Naveen Gattoju**

Thank you for your time, insights, and guidance during this project. Your perspectives and feedback on the planning for our algorithm and data access were critical to the success of this project.

Peer Support

- **Emilio Difilippantonio** – A fellow student working on a similar project with ABS. His ideas, feedback, and collaboration were valuable— thank you for being a great partner and creative thinker throughout this journey.

End!
Questions?