Alcohol Beverage Services (ABS) Internship Capstone Project Executive Summary

Sponsor

This project was done in coordination with the Montgomery County Alcohol Beverage Services (ABS), which is responsible for regulating alcohol licensing, enforcement, and education. They are the wholesale distributor of wine, beer, and spirits in Montgomery County and also have 27 retail locations.

The ABS asked us (3 interns) to help them improve their reordering algorithm to maximize profits. They wanted us to create improvements that would reduce excess restocking, which leads to products going to waste, and prevent understocking, which leads to lost revenue. We (the 3 interns) met and discussed how to divide the workload. I decided that I would focus on determining whether the sales data that they provided us could be clustered into groups to fine-tune algorithms to different trends.

Data and Data Cleaning

We have three data sets provided to us by the ABS: one from a high-volume store ($6+ million in annual revenue), one from a medium-volume store($3-$6 million in annual revenue), and one from a low-volume store($3 or less in annual revenue). Each data set has 500 rows and 17 columns, and contains weekly sales data from 2024. The columns are as follows:

Item ID	Product Name
Bottles per Case	Cost per Bottle
Weekly Bottles Sold (53 weeks)	Annual Bottles Sold

The data is well-maintained, so there was initially minimal cleaning done. Firstly, I combined the three data sets and created a column indicating the data set to which each row belongs. Since the sales are in bottles, but there are varying amounts of bottles per case, I mutated the data to show the sales in cases of product. This is also helpful because the items are reordered by the case, not by the bottle, so using cases sold from the start will save us time mutating the data later. Some of the data is highly variable, and some of the items have zero sales for several weeks, which caused problems.

Goal and Objectives

My goal is to create clusters of products based on sales data trends. I will create several rounds of clusters using different cluster amounts, and then test these clusters to determine whether they are mathematically viable using several indices that indicate the accuracy of clusters. To ensure that the clusters are based on sales trends alone, not on sales quantity, I will normalize the sales data. Ultimately, the aim is to find the best way to cluster the sales data to augment their reordering algorithm by making adjustments to the algorithm for different sales trends.

One objective is to have clusters that reflect long-term variability in sales. By identifying products whose sales amounts vary significantly over time, we can create products that are more or less responsive to recent changes in sales that may represent significant medium-term changes.

Another objective is to identify groups based on sales volatility. Some products have sales quantities that change drastically week-to-week, while others maintain stability. Identifying short-term volatility will allow us to better ignore the “noise” of these frequent changes and focus on the underlying patterns.

Identifying seasonal patterns in the clusters is another goal. Finding a cluster of products that have increased sales every year, for example, at Christmas, will allow you to prepare for that increase in sales ahead of time instead of reacting to the change. Identifying seasonal trends in sales data will allow proactive restocking, as opposed to reactive restocking.

The final long-term objective is to identify outlying products that exhibit extremely unusual behaviors. By identifying these products, we can ensure that they aren't treated the same as other products. Such treatment could result in extreme over- or understocking, which would create a massive waste of resources. These products need special attention, and there may not be an algorithm that can be created for them. Reorder amounts and quantities may be left up to the discretion of the location managers. The products may also be removed from certain or all locations or may be given other special treatment, such as special discounts.

Overall, these clusters are meant to identify patterns and trends in sales data to allow for more individualized reordering algorithms on a massive scale. This will increase ABS revenue, which is used to pay off county debt and fund other county projects.

Tools, Methods, and Resources

I used r in rStudio for the entire project. The main packages that I used for visualizations, data cleaning, and manipulation were tidyverse, ggplot2, and reshape2, and the package that I used to cluster the data and analyze the clusters was dtwdclust. I used the tsclust() function to create the clusters and the cvi() function to analyze those clusters. I did research on different clustering methods and the indices used to analyze those methods. I also obtained significant assistance from my professor (Professor Perine) and from my ABS sponsors.

Exploratory Data Analysis

To explore the data, I first created a five-number summary of the total sales in cases of each product in each of the three stores:

High sales volume:

Min	1st	Med	3rd	Max	Mean	Var
1.008	18.08	35.33	62.80	1150	51.02	5066

Medium sales volume:

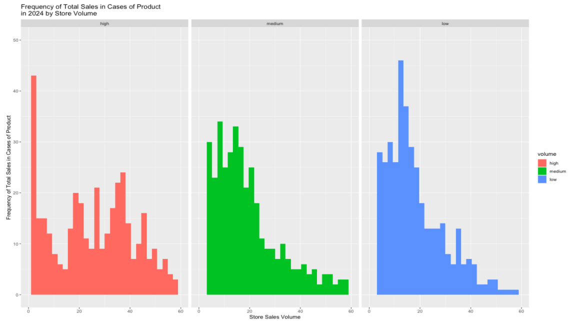
Min	1st	Med	3rd	Max	Mean	Var
1.008	3.5	12.88	22.60	371.3	18.78	759.1

Low sales volume:

Min	1st	Med	3rd	Max	Mean	Var
1.008	5.101	13.20	23.52	251.8	19.04	577.9

As we can see in the summaries above, the high volume data set tends to have higher sales in cases for its products, with a higher 1st quartile, median, 3rd quartile, max, and mean sales than the other two stores. It does, however, also have a significantly higher variance, being almost and order of magnitude greater than the variance in sales for the other two stores. Interestingly, the low volume store has a greater 1st quartile, median, 3rd quartile, and mean sales than the medium colume store, with the medium volume store only beating the low volume store in max and variance. The values, however, are close, and may not be statistically significant. These are also only the top 500 products, so the medium volume store may have a more even sales rate across all of its products, leading to more sales that aren't visible in our data. However, if the difference is statistically significant, it could indicate that the low volume store has more constituency with sales and slightly high sales across products, while the medium volume store is more likely to sell a lot of cases of a few products.

Next, I created a graph to visualize this distribution:



This histogram indicates the frequency of different amounts of cases sold for products in each of the stores. The high volume store in on the left (orange), the medium volume store is in the middle (green), and the low volume store is on the right (blue). This histogram confirms the conclusions that we made from the five number summary (with mean and variance added): The high volume store has significantly more sales of products, but it also has much more variance in sales, with the frequency of different quantities of sales spiking and falling throughout the graph. In comparison, the medium and high volume stores have a much more normal curve, though they are both heavily skewed towards the right. The low volume store has a higher peak and a smaller tail, though the peak is more to the right than that of the medium volume store. This is likely why the low volume store had higher values on the five number summary for most values. The medium volume store, however, has a greater tail, which is likely why its maximum sales value was higher, as well as its variance.

Overall, we can see some trends in the data, and we can get a sense of the sales patterns in each of the stores. This may be useful to know in its own right, as it could be helpful in determining how to reorder, sell, stock, and advertise in different locations.

Project and Results

Normalization

To create clusters of sales data that centered around sales trends, I normalized the data. I used min-max normalization for this process. I set the minimum weekly sales of each product to 0 and set the maximum weekly sales of each product to 1. However, this didn’t work with the clustering algorithm, so I multiplied this number by 1,000,000. This increased the scale of the normalization to allow it to be used with the code while not changing how the data would be analyzed.

Normalizing sales data before analysis is essential to ensure fair comparisons and accurate insights across different products. Raw sales figures can be influenced by various external factors such as varying price points or differing market sizes, which may distort patterns and lead to misleading conclusions. By normalizing the data, we can focus on the underlying patterns and trends. This process enables more meaningful comparisons, enhances the reliability of the clusters, and supports better data-driven decision-making.

Standardization

Min-max normalization isn’t optimal because it is extremely prone to outliers (a single high or low week of sales distorts the entire scale by changing the minimum or maximum value). Other methods, such as z-score standardization, are less prone to these problems. Z-score standardization involves converting all of the sales data to z-scores relative to the sales of that product. This not only avoids major issues when dealing with outlying sales data, but also more accurately measures the scale of the difference in sales, as the z-score is dependent on the standard deviation from the mean, not the position relative to the minimum and maximum sales points.

Another method of data standardization/normalization would be setting the 1st quartile of sales to -1 and the 3rd quartile of sales to 1. This would make it less susceptible to outliers while still maintaining a simple, easy-to-comprehend method of normalization/standardization.

Clustering Methods

I used the partitional clustering method of the tsclust() function, as it was the easiest to apply (fewer input variables needed) and the quickest to run. Partitional clustering involves clustering data into distinct groups, with every data entry being in exactly one cluster.

This contrasts with hierarchical clustering, which repeats several interactions of clustering on each cluster group. This leads to several levels of clusters, i.e., a hierarchy. Though this method may be appealing, especially since larger clusters can be used and, if necessary, broken down into smaller parts, it is extremely computationally expensive due to its iterative process, so it should only be used if other methods are severely lacking.

A common method of clustering time series is using Euclidean distance. This involves comparing values between points at each point in a time series. Although this method is fast, it is very bad at detecting patterns that aren't aligned. As such, I used a different method.

The dtwclust package uses dynamic time warping (dtw) clustering to find patterns in the data regardless of time shifts or time warps. Time shifts are when a pattern is present in two series, but the series have different starting points for the patterns, causing the patterns to be misaligned. Time warping occurs when a pattern in one series has a different phase length than the same pattern in another series. Though this method is much more robust and better at detecting patterns, it may not be as useful when detecting specific seasonal patterns, such as an increase in sales in December, as a similar pattern may occur for another product in another season, and those products may be grouped together. Dynamic time warping is also very computationally expensive, so it may not be the best for use on massive data sets, like the one used by ABS.

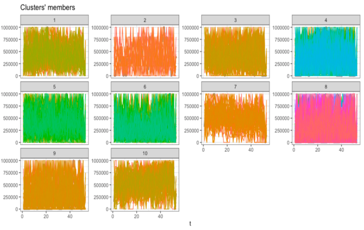
Clustering with k = clusters

I decided on one normalization technique and one clustering method (control variables), which left me with the number of clusters as the experimental variable, though the goal isn't to find an optimal number of clusters, but simply to prove that clustering works and that it can be optimized.

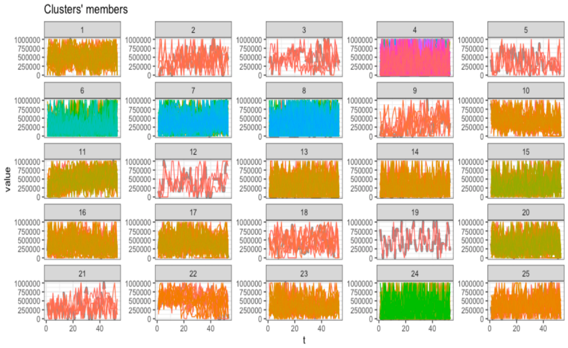
The number of clusters has a significant impact on the accuracy of the clusters that are produced. Having too few clusters leads to many dissimilar time series being clustered together, producing inaccurate clusters that aren't useful. Having too many clusters, however, results in overfitting, where many clusters only have a handful of products. This isn't useful, as different algorithms can't be efficiently applied to each of these groups. The proper balance in the number of clusters in necessary to allow the algorithm to produce accurate clusters with many data points.

I created 4 groups of clusters, where k = the number of clusters: one group with k = 10 clusters, one group with k = 25 clusters, one group with k = 50 clusters, and one group with k = 100 clusters.

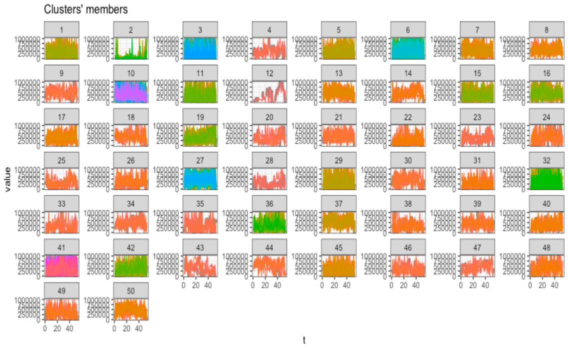
Here are the clusters from the group with k = 10 clusters:



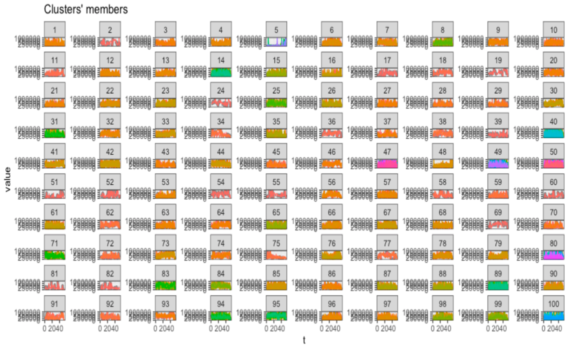
Here are the clusters from the group with k = 25 clusters:



Here are the clusters from the group with k = 50 clusters:



Here are the clusters from the group with k = 100 clusters:



Some of the clusters don't look as good as others, but it can be difficult to visualize how accurate they are. Here are some of the indices that are used to indicate how accurate data clusters are:

	K = 10	K = 25	K = 50	K = 100
Calinski-Harabasz Index	93.88816887	42.88126022	21.12122747	11.783747650
Davies-Bouldin Index	2.51595387	2.03270419	2.04084868	1.9101442736

The Calinski-Harabasz index measures the ratio of dispersion between clusters to the dispersion within clusters. For this index, a higher number indicates that the clusters have a strong connection within them and weak connections with other clusters. In short, it indicates that the clusters have done a good job separating similar series and clustering them together. Our results show that fewer clusters are associated with a better Calinski-Harabasz index, indicating that fewer clusters are better are grouping similar products and separating them from dissimilar products.

The Davies-Bouldin index measures the ratio of the distances within a cluster to the distances between a cluster and its closest neighbor. Based on the test results, there seems to be an association between the number of clusters and this index, where an increased number of clusters is associated with a lower Davies-Bouldin index. This indicates that when the number of clusters increases, the distance between a cluster and its closest neighbor is greater. This is interesting because I would expect that more clusters would force the algorithm to create similar clusters, but this may not be the case. Alternatively, the connections within the clusters may just be that much stronger that they offset similarities between the clusters.

Conclusion

In conclusion, it is possible to accurately cluster the ABS sales data. The number of clusters created is also a variable that can be optimized to increase cluster accuracy, but there is no one best method; rather, there are benefits and drawbacks of each number of clusters. The number of clusters created must be a decision made with calculated tradeoffs to ensure the best product for the clusters' specific purpose. The number of clusters created may also change over time as better clustering methods, normalization and standardization methods, and predictive algorithms are created and implemented.

Next steps

Testing Clusters

A test case should be created as a proof of concept wherein several clusters are put into predictive algorithms that predict sales based on the previous sales data. This test case is used to determine how useful, if at all, clustering the sales data might be. These test cases should be done after every step taken to improve the clustering methods to determine whether an increased accuracy in clustering translates to better predictive algorithms. As the clusters become more optimized based on normalization and standardization, cluster amounts, and clustering methods, these test cases should be done more frequently and with more clusters and algorithms to help narrow down which clustering methods actually work and which predictive algorithms they work best with.

Acknowledgements

Professors:

- Professor Perine
- Professor Saidi
- Professor Alraee

ABS Internship Sponsors:

- Gaël Le Guellec
- Victoria Liu
- Kori Wenman
- Kathy Luh
- Victoria Lewis
- Naveen Gattoju
- Jocelyn Rawat
- Melissa Romeo

Classmates and Project Partners:

- Emilio Sanchez San Martin
- Davi De Almeida
- DATA 205 Early College 2025 Cohort

References

"About Us - Alcohol Beverage Services - Montgomery County, Maryland." *Www.montgomerycountymd.gov*, www.montgomerycountymd.gov/ABS/AboutABS.html.

Hyndman, Rob, and George Athanasopoulos. "Chapter 3 the Forecaster's Toolbox | Forecasting: Principles and Practice (2nd Ed)." *Otexts.com*, 2025, otexts.com/fpp2/toolbox.html. Accessed 14 May 2025.

"Moving Averages in R." *GeeksforGeeks*, 16 Nov. 2023, www.geeksforgeeks.org/moving-averages-in-r/.

Posit Sotware, PBC. "Calculate the Similarity between Pairs of Time Series Data." *Posit Community*, 7 Jan. 2021, forum.posit.co/t/calculate-the-similarity-between-pairs-of-time-series-data/92546/2. Accessed 14 May 2025.