Supermarket Sales Analysis: Better Understand Your Customers

Payton Reaver

2023-04-21

Introduction

In this study, I will be analyzing customer sales from three different supermarkets. Specifically, I will utilize clustering techniques to try and segment what *kinds* of customers these supermarkets encounter. Clustering algorithms are unsupervised learning techniques that aim to group together similar data points into clusters based on the similarity or dissimilarity between them. So, in tandem with this data set, the overarching goal is to identify spending patterns that relate to large quantities of customers.

But what to do with this information?

Well that is an excellent question. The answer? It depends. But in general, imagine you are taking on the roll of a new general manager for Grocery Store Inc, brand new branch store. Some vital questions that will dictate your success could be:

- What products do you sell well?
- What are the products you don't sell at all? What changes can be made to fix this?
- Who is visiting your store and what are they buying?
- When do you experience the largest flow of customers? Are there any seasonal trends?

Having a concrete idea of these questions is what will drive success, but in the right direction. Through cluster analysis, brands and businesses can discover correlations between customer and product lines that aren't seemingly obvious and make them stand out from competition.

Suppose you have your customers segmented, now what?... Marketing. If brands/businesses are able to find what could high dimensional correlation between customers and sales, the customer's behavior is painting the picture itself. If I can find a large group of customers visiting my store at similar times, or buying similar items, or spending similar amounts of money per gross income, I now have a much more educated and complete idea on *what* should be marketed to *who*.

Separate from marketing, depending on the economy of your business, maybe your goal is to keep the customers you already have. Being able to visualize the components of your sales through clustering gives great insight into what is already working for you. Maybe you want to re-brand your business towards a more specific market. Maybe you want to take your current customer segmentation and make it even more refined. But, before making any rash/naive business decision, you want to gauge the success. Through clustering this is possible.

Data Set

This data set contains 1,000 unique sales transactions and is studied on 19 variables.

Variable Classification

Listed below is each variable and its corresponding definition.

Invoice ID: Computer generated sales slip invoice identification number

Branch: Branch of super center (3 branches are available identified by A, B and C).

City: Location of super centers

Customer Type: Type of customers, recorded by Members for customers using member card and Normal for without member card.

Gender: Gender type of customer

Product Tine: General item categorization groups - Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel

Unit Price: Price of each product in $

Quantity: Number of products purchased by customer

Tax: 5% tax fee for customer buying

Total: Total price including tax

Month: Month the purchase was made (January, February, March)

Day: Day of the purchase.

Year: Year of purchase (2019).

Time: Purchase time (10am to 9pm)

Payment: Payment used by customer for purchase (3 methods are available – Cash, Credit card and Ewallet)

COGS: Cost of goods sold

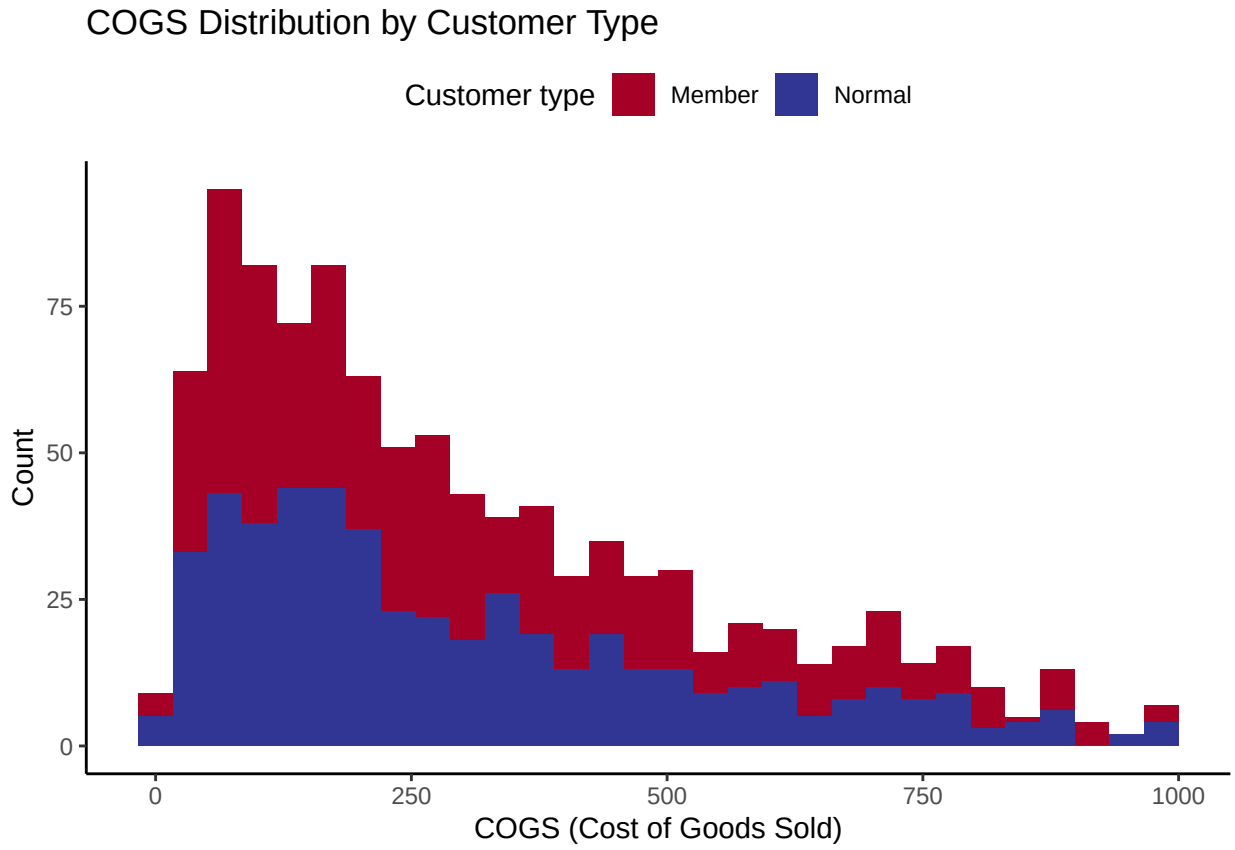
Gross Margin Percentage: Gross margin percentage

Gross Income: Gross income

Rating: Customer stratification rating on their overall shopping experience (On a scale of 1 to 10)

Graphical Analysis

We start with some graphical analysis. The purpose behind EDA is to find features or variables that have noticeable trends in the data. For example, show below is the distribution of COGS colored by Members and Non-members.

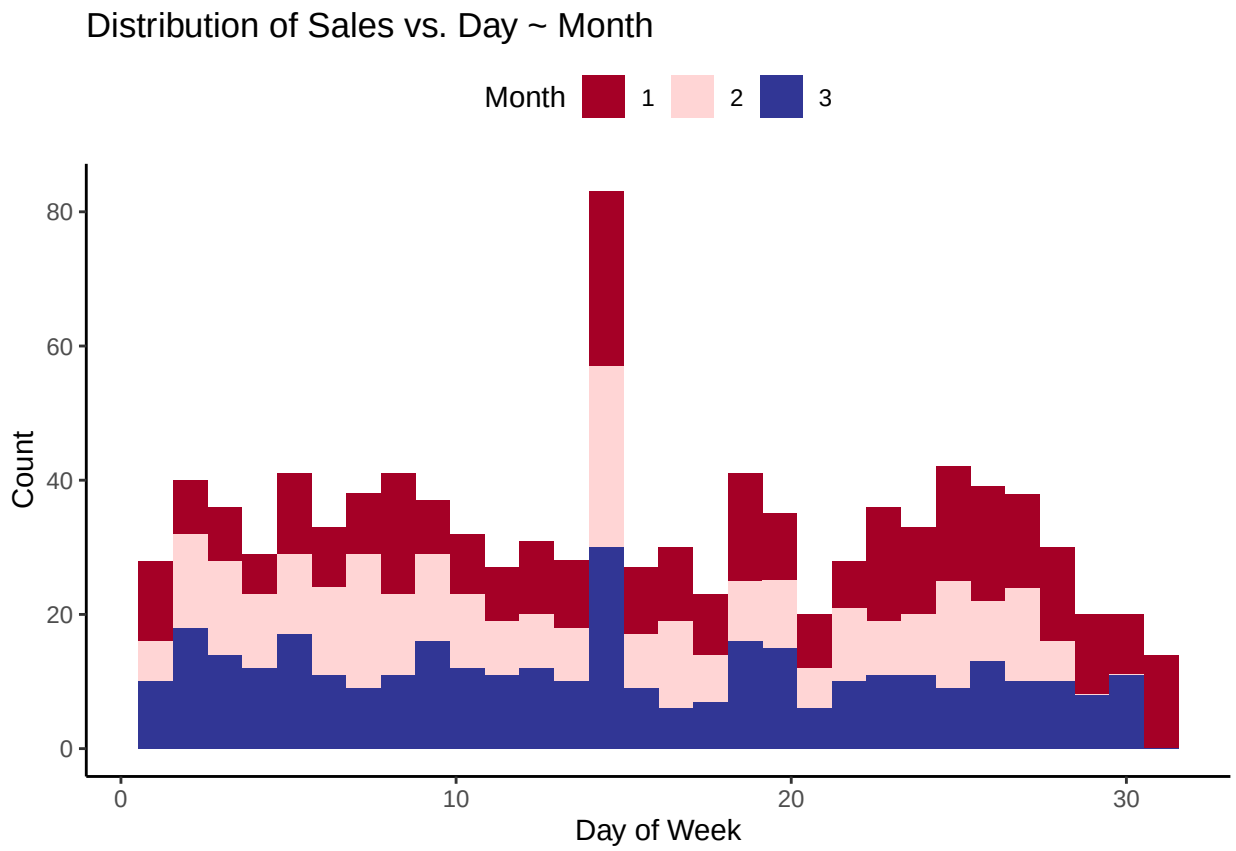


The inspiration behind this was curiosity. I was curious to see if Members of the store have different spending patterns than that of Non-members. It seems this was not the case, and it is surprising to me. If you hold membership at a specific store, then I would imagine you fancy visiting that store than others because you have chosen your allegiance specifically. Why is it that Members and Non-members are spending roughly the same amount at this store? Perhaps there is no readily evident benefit to having membership, maybe the perks should be adjusted, who knows. More information would be required to study this further.

Some other information that would have been useful is multiple customer transactions, i.e., same people different times. This would have helped as we could aggregate over a longer period of time to really get a good idea of spending patterns.

But, taking the graph as it, it still shows a great picture. We can identify a textbook right-skew of the data and this makes perfect sense. Most people don't have $700 - $1000 to spend at the supermarket, member or not.

Next up for graphical analysis is seasonal patterns. Recall back to your brand new branch store for Grocery Store Inc. What if you could tell me the best or worst times to come to your store. What if you could predict days of the week you will be busy? Well, in terms of controlling labor this is great news! If you are expecting large flows of customers, you might want all hands on deck for that day. But, on the flip side, what if you only need half of your regular staff to get through the day? That is money you are saving as a business owner.



There is a very clear winner here for "Most popular" and it seems to fall around the 13-15 day mark. What could this be? Payday. Oh yes, we are talking big bucks. Half way through the month? Is it the 14th? Time to spend some money. This makes perfect sense. When customers are given more disposable income or given more money in general, then those customers will also spend more money.

Principal Component Analysis

PCA (Principal Component Analysis) is a statistical technique used to reduce the dimensionality of high-dimensional data by identifying the most important patterns in the data and projecting it onto a smaller set of orthogonal variables called principal components. This is useful if standard EDA did not provide any obvious patterns in the data set, or maybe your data set has many many variables and is too large to work with.

Each Principal Component (PC) is a linear combination of the original variables with the coefficients of each variable indicating its contribution to that PC. The coefficients of each variable can change across different PCs, indicating that the PCs weight and combine the original variables differently. Therefore, the difference between the PCs lies in both the amount of variation they capture and how they weight and combine the original variables to capture that variation.

In other words, imagine you are given the task of digging a hole for a fire-pit. Further, suppose I gave you a crane, shovel, and a spoon. While a crane is great for digging holes, it is far too big. Similarly, a spoon is perfect for digging holes into cereal or yogurt, but cumbersome for a fire-pit. The right balance of efficiency and execution is with the shovel. So, I really only need the shovel to complete my task. This is the idea behind Principle Component Analysis.

```
sales$Month <- as.numeric(sales$Month)

sales_std <- scale(sales[,c(7,8,10,11,12,19,18,16)])

pca_result <- prcomp(sales_std)

summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6       PC7
## Standard deviation     1.982 1.0704 1.0068 0.9858 0.9254 0.29072 4.174e-16
## Proportion of Variance 0.491 0.1432 0.1267 0.1215 0.1070 0.01057 0.000e+00
## Cumulative Proportion  0.491 0.6342 0.7609 0.8824 0.9894 1.00000 1.000e+00
##                             PC8
## Standard deviation     1.209e-16
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

Shown are the results of the PCA. We can see at each iteration, we are given 3 metrics to analyze. What we are after is finding a subset of variables smaller than what we started with (8), but at the same time, we want to explain as much variance in our data as possible. Finding this balance is what we want to do.

At first glance, I would say we can grab PC1 to PC5 and run, but lets take a deeper look under the hood.

```
mat <- pca_result$rotation

print(mat)
```

```
##                      PC1          PC2          PC3         PC4          PC5
## Unit price   -0.328169581  0.268360689  0.305862790  0.57803873  0.232587010
## Quantity     -0.364862000 -0.232608010 -0.299627368 -0.50227793 -0.225226525
## Total        -0.502812106 -0.006601106 -0.002951532  0.00686331 -0.005793656
## Month         0.016070080 -0.624281304  0.278394605  0.39362372 -0.614446214
## Day          -0.000398876  0.675910572  0.234197899 -0.15655070 -0.680980220
```

```
## Rating        0.021494138  0.164812266 -0.827220072  0.48373446 -0.231942636
## gross income -0.502812106 -0.006601106 -0.002951532  0.00686331 -0.005793656
## cogs         -0.502812106 -0.006601106 -0.002951532  0.00686331 -0.005793656
##                       PC6           PC7           PC8
## Unit price   -0.581815886 -6.425828e-16 -1.344162e-16
## Quantity     -0.648060619 -9.485437e-16  4.596448e-17
## Total         0.283514306  7.520274e-01 -3.179961e-01
## Month        -0.005221880 -1.183282e-17 -1.021793e-17
## Day           0.007341891  2.292871e-17  5.163468e-18
## Rating        0.016895353 -5.794319e-18  4.058163e-19
## gross income  0.283514306 -6.514064e-01 -4.922768e-01
## cogs          0.283514306 -1.006210e-01  8.102728e-01
```

This output shows the correlation between each variable and each principal component, it is called a loadings matrix. The rows correspond to the original variables (unit price, quantity, total, month, day, rating, gross income, and cogs), and the columns correspond to the principal components (PC1, PC2, PC3, PC4, PC5, PC6, PC7, and PC8).

A positive value indicates a positive correlation, and a negative value indicates a negative correlation. The magnitude of the value indicates the strength of the correlation.

Interpretations of some of the PCs are as follows:

- PC1 gives us strong negative correlations for COGS, Gross Income, and Total. This suggests variable importance related to the total cost of the transaction with respect to a customers income, similar to our comparisons from EDA.
- PC4 tells almost an entirely different story. PC4 is telling us that Unit Price and Rating have very strong positive correlations in the data set. This makes sense as a store is seemingly "better" if it receives higher ratings.
- PC8 is where things get tricky. By inspection, it seems COGS and Total hold the largest weight in this PC. Similar to PC1, PC8 is finding association related around total cost of the transaction.

From this analysis, it is tough to say which variables are actually the best, but what we can do is find a nice balance of cumulative proportion of variance explained, and the standard deviation of the respective PC. I am still comfortable rolling with PC1 to PC5.

Hierarchial Clustering

Hierarchical clustering is a type of clustering algorithm used to group similar objects or data points into clusters based on their distance or similarity. The algorithm works by building a hierarchy of clusters, where each node in the hierarchy represents a cluster that contains a set of objects or data points.

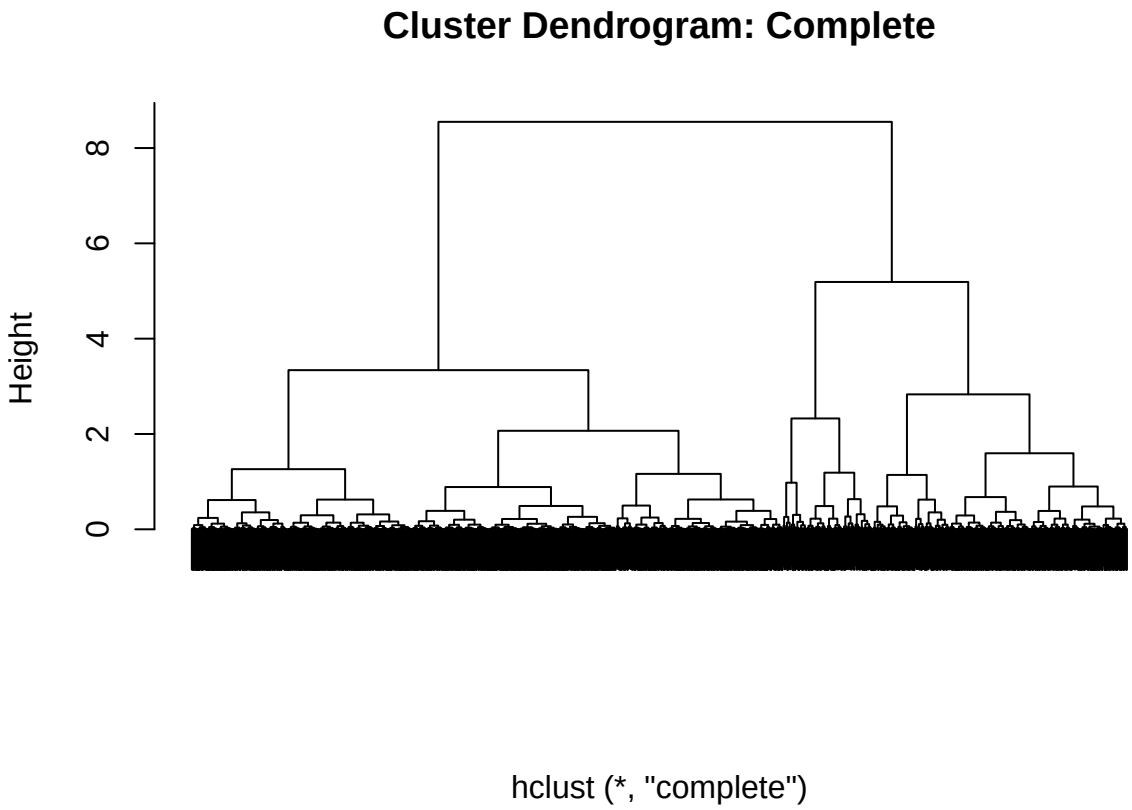
There are two types of hierarchical clustering algorithms: agglomerative and divisive.

In agglomerative clustering (this study), the algorithm starts by considering each object or data point as a separate cluster and then iteratively merges the closest pairs of clusters until a single cluster containing all objects is formed. This process creates a tree-like structure called a dendrogram, which shows the hierarchical relationships between the clusters.

The choice of distance metric and linkage method are important considerations in hierarchical clustering. The distance metric measures the dissimilarity between two objects or data points, while the linkage method determines how the distance between clusters is computed. Some common linkage methods include single linkage, complete linkage, and average linkage.

Shown in the next couple of pages are each distance metric's dendrogram using PC1 (PC1 infact has the most explanation on the given data set). When deciding on where to actually split the clusters, that is dependent on the problem at hand. Given I have 1,000 observations, I will be sure to try and find clusters that have approximately even splits. Splitting dendrograms with too many subsets of clusters can lead to "over fitting" in a sense or trying too hard to cluster your customers. While on the other hand too big of clusters does not really give further insight into the problem. We will see this one specifically (hint hint, the single linkage)

Hierarchial Clustering: Complete



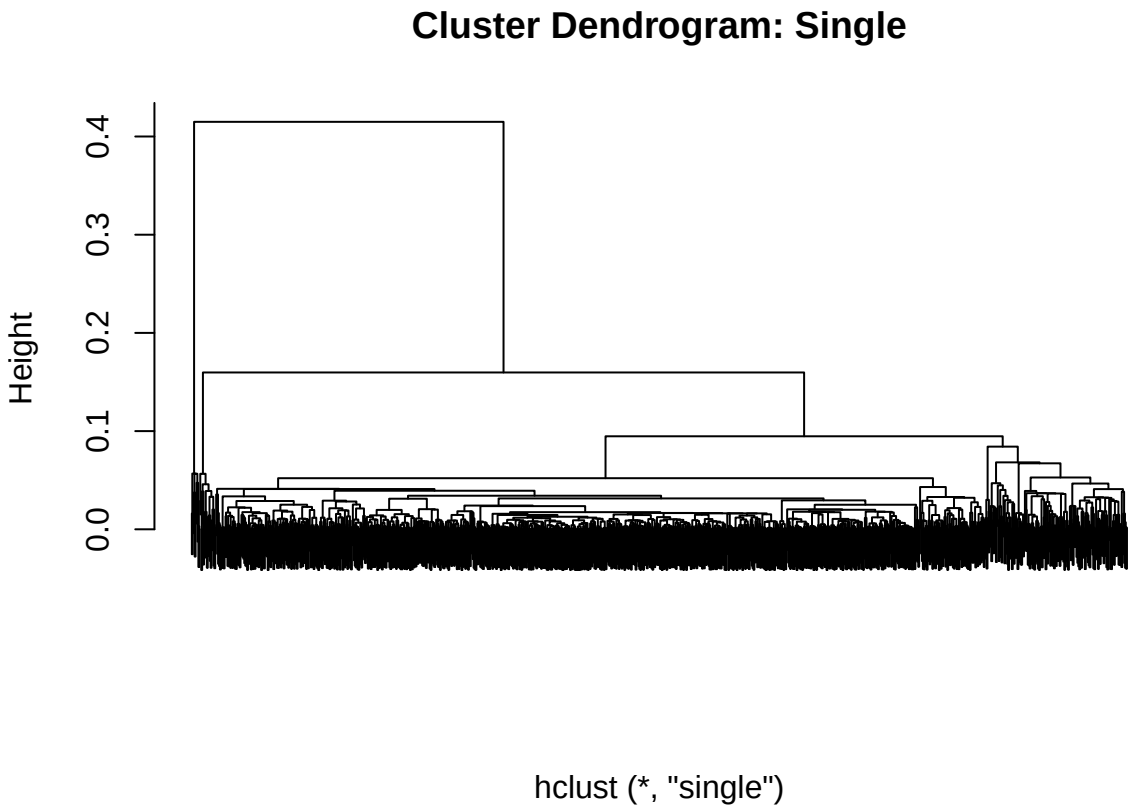
How does it work?

Complete linkage metric: The complete linkage metric, also known as the maximum distance metric, measures the similarity between two clusters as the longest distance between any two objects in different clusters. This distance metric tends to produce compact, spherical clusters and is less sensitive to noise and outliers than the single linkage metric.

Where to split?

Well, given the distribution of the dendrogram, I think a healthy spot to split this clustering is about 1.5-2 on the y-axis. While this gives a lot of different clusters, except for the really tiny cluster, they are all approximately the same size.

Hierarchial Clustering: Single



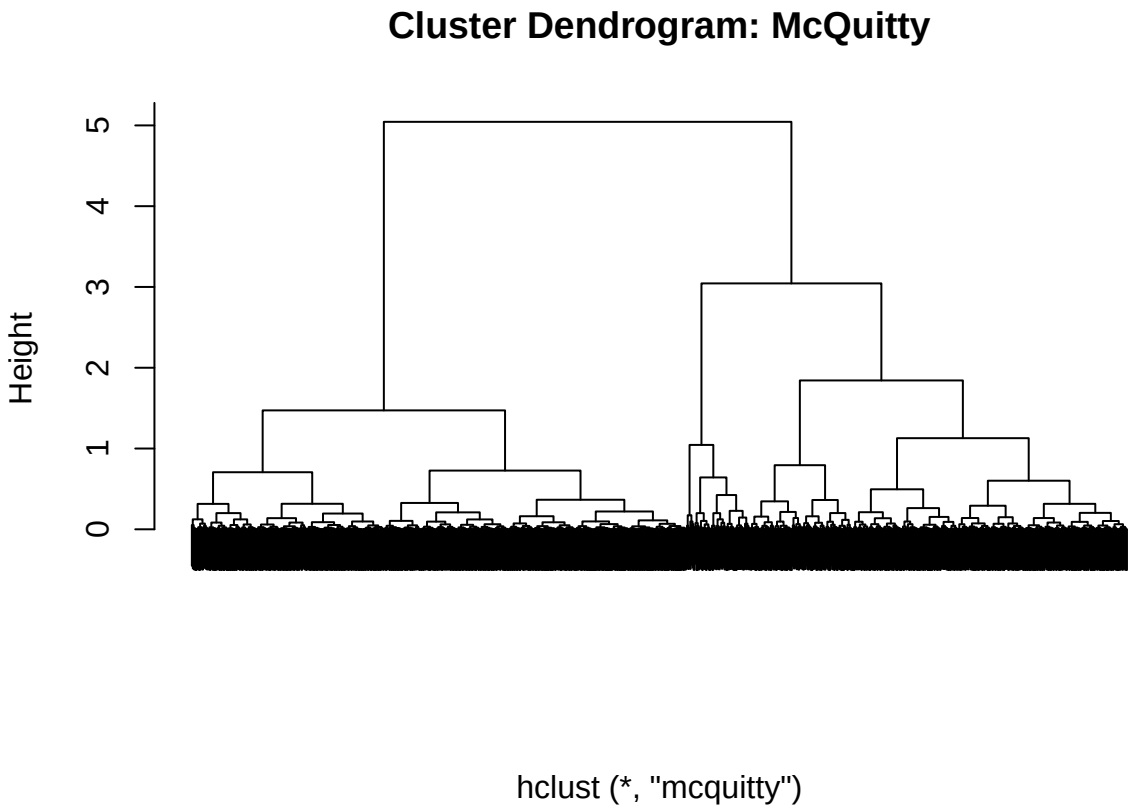
How does it work?

Single linkage metric: The single linkage metric, also known as the minimum distance metric, measures the similarity between two clusters as the shortest distance between any two objects in different clusters. This distance metric tends to produce long, thin clusters and is sensitive to noise and outliers in the data.

Where to split?

I would split this data nowhere. Overall, this was a useless calculation. Rather, this is exactly what not to submit as results. But, if I am condemned to just one number, give me 0.4+, in this dendrogram the best cluster is no cluster.

Hierarchial Clustering: McQuitty



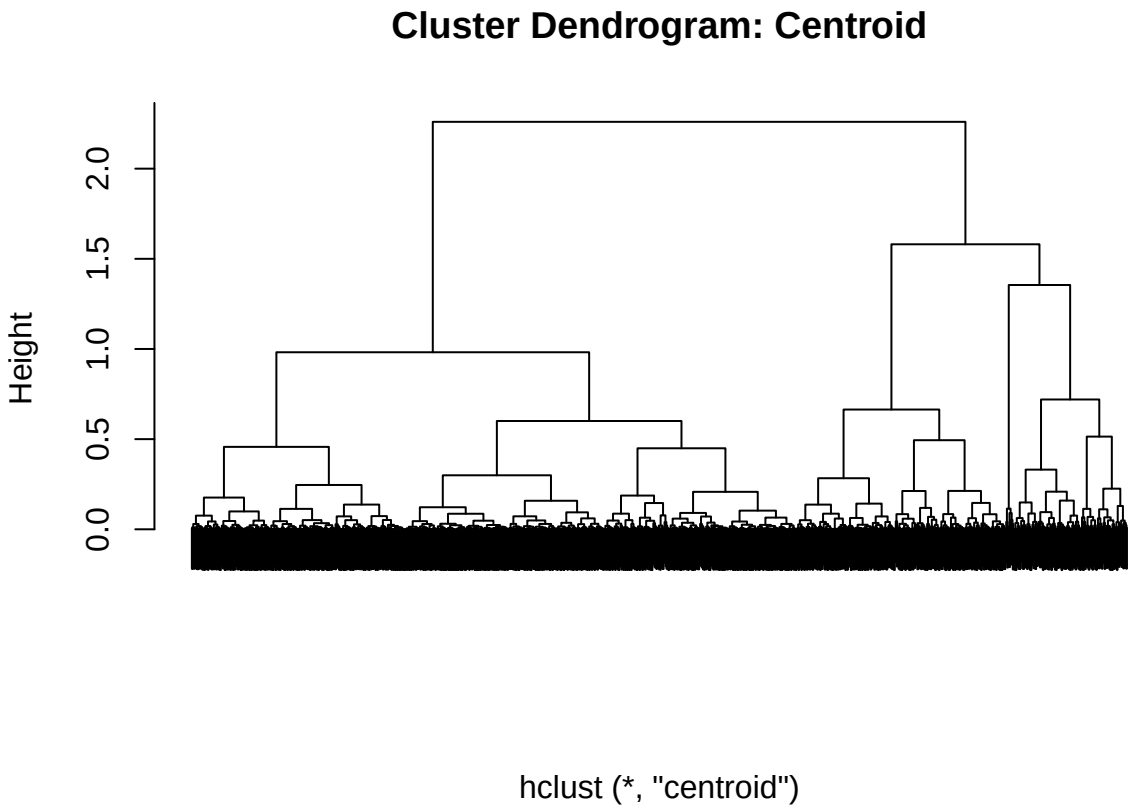
How does it work?

McQuitty distance metric: The McQuitty distance metric is a measure of the similarity between two clusters, which is defined as the maximum distance between any two objects in the clusters. This distance metric is useful when clusters have a wide range of dissimilarities between objects, as it focuses on the most extreme distances.

Where to split?

This one has a nice little subset of observations in the middle of the dendrogram which is just interesting to see. I would provide a split somewhere between 1 and 1.5ish. I am curious to see what separates the small middle subset with the two other larger subsets.

Hierarchial Clustering: Centroid



How does it work?

Centroid linkage metric: The centroid linkage metric measures the similarity between two clusters as the distance between the centroids of the clusters. This distance metric is sensitive to the shape and size of the clusters and tends to produce balanced clusters. It is also computationally efficient compared to other linkage methods.

Where to split?

This one is up in the air for me. If you would like a lot of different segments, you're looking at the .5 range. Rather if you need minimal number of clusters, I recommend you to the 1-1.5 range.

While each of these different dendrograms are different in their own way, it is important to note the ambiguity within the models. What does it mean for a linkage oif Complete to have a split at height = 1? What about a Complete linkage split at height = 0.2? It is hard to interpret these models directly, but in return we can see very nice visuals of all of the different clusters.

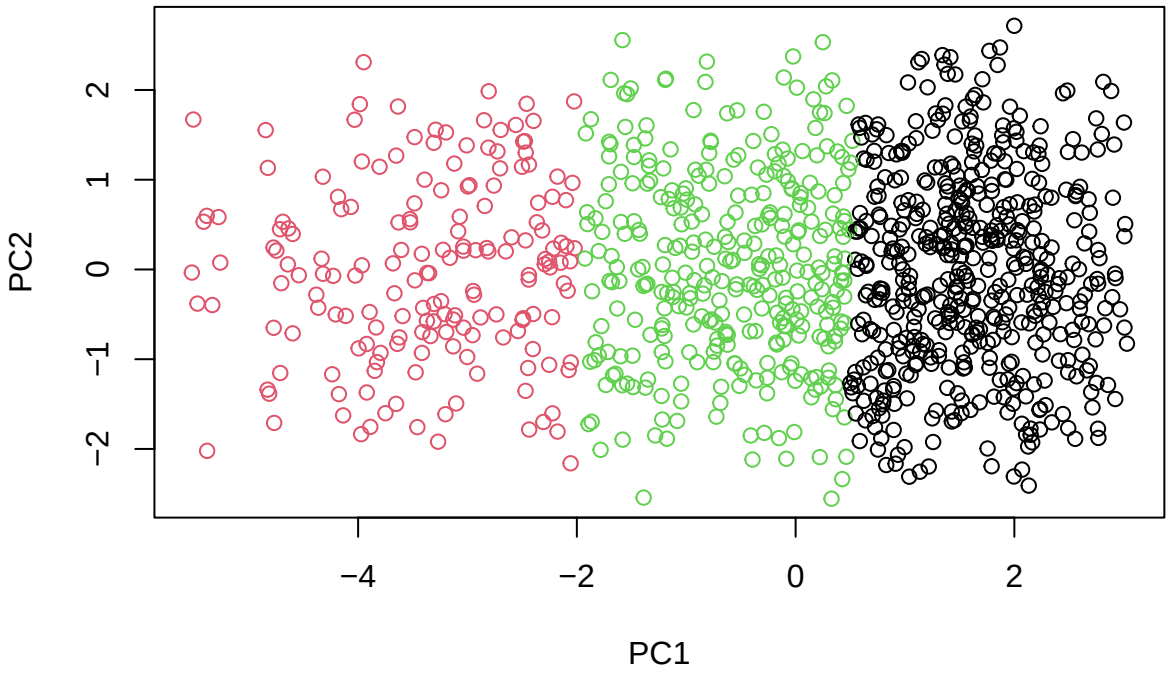
K-means Clustering

K-means clustering is a type of clustering algorithm that partitions a data set into a predetermined number of clusters, where each cluster is defined by a centroid, or mean point. The algorithm works by iteratively assigning each data point to the nearest centroid and then recalculating the centroids based on the new assignments. This process continues until the centroids no longer change significantly or a maximum number of iterations is reached.

In contrast to hierarchical clustering, K-means clustering is a flat clustering method, meaning it does not create a hierarchical structure of clusters. Instead, K-means clustering assigns each data point to a single cluster based on the nearest centroid. This makes K-means clustering more suitable for data sets where the number of clusters is known or can be estimated beforehand (such as visualizing dendrograms), whereas hierarchical clustering can be used to explore and discover the underlying structure of the data without prior knowledge of the number of clusters (that's what we did!).

Another difference between K-means clustering and hierarchical clustering is the way they handle outliers and noise in the data. K-means clustering is sensitive to outliers, as they can significantly affect the position of the centroids and the assignment of data points to clusters. On the other hand, hierarchical clustering is more robust to outliers, as they are less likely to affect the overall structure of the dendrogram. Think about it. If there is just one point way out of the expected distribution, how would you go about arguing what cluster assignment it gets? What if it is it's on little cluster. Who knows. This is a key difference between Hierarchical Clustering and K-means.

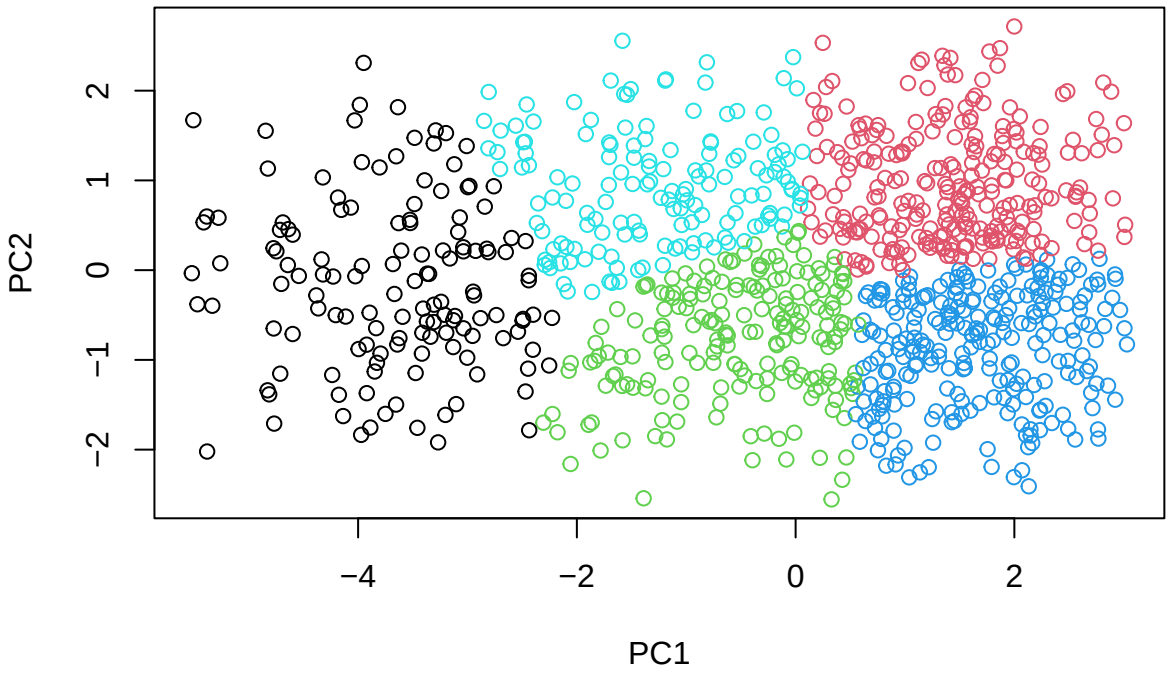
This is a very useful technique given we just analyzed multiple dendrograms and have a pretty good idea on how many clusters we can expect. Overall, I think it is safe to assume we can expect 3-5 clusters, 1-2 of which being significantly smaller than the rest. We are taking what we learned from above and applying it further.



This output shows the clustering assignments using K-means clustering. The initial parameters are PC1 and PC2 results, and 3 centers. In other words, using the first two principal components and having 3 clusters.

Also, graphing PC1 against PC2. We can see that we have strictly vertical boundaries. This shows the interaction between each PC and its respective clustering assignment. Given they are all vertical, this suggests with respect to K-means, the first two principal components do not contribute a lot of unique information on the data set.

Here is the same type of graph, but I told K-means clustering I want 5 clusters now instead of 3. This will be interesting to study because it will show me which already existing clusters will be segmented further. Also, this will give more insight on the decision boundaries. We are in a high dimensional space, so having the opportunity to visualize the decision boundaries is great!



I want to make note of where exactly these new clusters are appearing. We are still using the first two Principal Components, but more clustering. The algorithm left the left hand side of points mostly unaffected. When we want to see more segments, the right side has a weaker correlation than the left side. I.e., data points clustered on the left are closer relatives to each other than data points on the right. There must be some underlying connection to keep the left most column unaffected in comparison to the first algortihm ran using K-means.

Project Conclusions and Limitations

Perhaps the only strikingly obvious limitation is figuring out exactly why each data point is clustered into the assignment it was given. Maybe it has something do with income, or COGS, or unit price, and even a combination of all 3. While we were able to visualixe the different clustering assignments and study the interaction between the variables, we lose credibility on the interpretation part of the analysis.

But, while there is no "correct" answer in clustering problems, it is not entirely a waste. We were able to analyze different techniques and algorithms to go about trying to find useful and efficient customer segmentations.

If you want to use hierarchical clustering, it pays to try different linkages. Some results gave similar answers such as McQuitty and Centroid. Other linkages gave clusters that were just not useful at all, looking at Single here.

But, maybe you want to try K-means clustering. In a more advanced study, one could even use hierarchical clustering as a form of data processing. When doing this, you collect ideas on how many clusters seems reasonable. We were able to assume 3-5 clusters could be a good place to start.

In conclusion, customer segmentation is a complex task that requires careful consideration of the available data and the specific business goals. There are a variety of clustering algorithms that can be used to segment customers based on sales data, each with their own strengths and weaknesses. However, it is important to note that different algorithms can produce wildly different results on the same dataset. Additionally, even within a single algorithm, the relationships between variables and the resulting clusters can vary greatly depending on the specific parameters chosen. Therefore, it is important to approach customer segmentation as an iterative process that involves testing different algorithms and parameters, as well as incorporating domain knowledge and business objectives to ensure the resulting segments are meaningful and actionable.