# Title: K-Means Clustering

Course: MBA563 Module: 08 Author: Kariuki \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Run this code before you start.

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.5   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

#install.packages('factoextra')  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.2.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

options(tibble.print\_max = Inf)  
options(tibble.width = Inf)

We will now us the K-means clustering algorithm to address our business problem.

TECA has a problem. They have a customer loyalty program, but it is not as effective as it could be. TECA would like to do two things to improve their ROI on their customer loyalty program. First, they would like to increase the number of their loyalty customers. Over the last three years, the amount and percentage of transactions from loyalty customers has steadily increased, but overall, those percentages are still lowâ€”never reaching higher than 30%. TECA management knows there is more they can do. Second, TECA would like to increase the yield and profitability of their loyalty customers. They get the sense that some customers are spending good money on high margin products, but others seem to just dash in and out of the store, purchasing gas and cigarettes on the weekend and nothing more. From prior research, TECA learned that they do not have a good model for which transactions are going to come from loyalty customersâ€”the accuracy of their model is only about 60%. So, all in all, they realize that loyalty customers are good, but some are probably better than others. What TECA would like to do is find a way to figure out what types of customers they have. Can they separate their customers into subgroups or clusters? If so, they can market to their loyalty customers in different ways and promote those products and habits that will be most beneficial for TECA.

The first thing we need to do to help TECA is to bring in the dataset we are going to work with. This is TECA data that has been transformed to be used in our analysis. Let’s load it and examine it a bit. We first notice that the dataset has a little over 200,000 rows and 12 features. This data is aggregated at the level of an individual loyalty card holder. Thus, each row is the sum of all purchases for that person for a three-year period. So, what features are included? As you can see, each feature is a sum of the items purchase in a particular area of the store. I describe each column below: \* area.fresh: products in this area include, for example, pizza, hot sandwiches, salads, and roller grill items; \* area.snacks: products in this area include, for example, candy, gum, chips, and salty snacks; and \* area.cooler: products in this area include, for example, energy drinks, canned soda, and juice; \* area.grocery: products in this area include, for example, milk, eggs, and cheese; \* area.nongrocery: products in this area include, for example, clothing, magazines, medicine, and newspapers; \* area.alcohol: products in this area include, for example, wine and beer; \* area.tobacco: products in this area include, for example, cigarettes and chewing tobacco. \* area.fuel: products in this area include, for example, gas; \* area.dispensed: products in this area include, for example, cold and hot dispensed (fountain) drinks; \* area.lottery: products in this area include, for example, lottery tickets; \* area.miscellaneous: products in this area include, for example, store services and coupons; \* refill.yes indicates whether or not the purchase was a refill of fountain soda or coffee.

# Explore the data

clustering\_input2 <- readRDS("C:/Users/User/OneDrive/Documents/projects/k-means.rds")  
str(clustering\_input2)

## tibble [272,417 × 15] (S3: tbl\_df/tbl/data.frame)  
## $ customer\_id : num [1:272417] 1 1.02 1.04 1.19 1.25 ...  
## $ area.fresh : int [1:272417] 0 0 1 0 0 2 0 1 0 0 ...  
## $ area.snacks : int [1:272417] 0 0 0 1 0 0 0 0 1 0 ...  
## $ area.cooler : int [1:272417] 0 0 0 1 0 0 0 0 0 0 ...  
## $ area.grocery : int [1:272417] 0 0 0 0 0 0 0 0 0 0 ...  
## $ area.nongrocery : int [1:272417] 0 0 0 0 0 0 0 0 0 0 ...  
## $ area.alcohol : int [1:272417] 0 0 0 0 0 0 0 0 0 0 ...  
## $ area.tobacco : int [1:272417] 0 0 0 0 0 0 0 0 0 0 ...  
## $ area.fuel : int [1:272417] 1 0 0 0 0 0 0 0 0 0 ...  
## $ area.dispensed : int [1:272417] 0 0 0 0 0 0 1 0 0 1 ...  
## $ area.lottery : int [1:272417] 0 1 0 0 4 0 0 0 0 0 ...  
## $ area.miscellaneous: int [1:272417] 0 0 0 0 0 0 0 0 0 0 ...  
## $ refill.no : int [1:272417] 1 1 1 2 4 2 0 1 1 0 ...  
## $ refill.yes : int [1:272417] 0 0 0 0 0 0 1 0 0 1 ...  
## $ mean\_revenue : num [1:272417] 57.39 6 2.99 2.73 -0.5 ...

slice\_sample(clustering\_input2, n=10)

## # A tibble: 10 × 15  
## customer\_id area.fresh area.snacks area.cooler area.grocery area.nongrocery  
## <dbl> <int> <int> <int> <int> <int>  
## 1 200. 0 0 0 0 0  
## 2 4672. 0 0 0 1 0  
## 3 5882. 0 0 0 0 0  
## 4 7075. 3 1 5 0 1  
## 5 6343. 0 0 1 0 0  
## 6 3790. 2 0 1 0 0  
## 7 4071. 0 0 0 0 0  
## 8 7306. 0 1 0 0 0  
## 9 6333. 0 0 1 0 0  
## 10 9110. 0 0 0 0 0  
## area.alcohol area.tobacco area.fuel area.dispensed area.lottery  
## <int> <int> <int> <int> <int>  
## 1 0 0 2 0 0  
## 2 0 0 0 0 0  
## 3 0 0 1 0 0  
## 4 0 1 0 0 0  
## 5 0 0 1 0 0  
## 6 0 1 1 5 0  
## 7 0 0 1 0 0  
## 8 0 0 0 0 0  
## 9 2 2 0 0 1  
## 10 1 0 0 0 0  
## area.miscellaneous refill.no refill.yes mean\_revenue  
## <int> <int> <int> <dbl>  
## 1 0 2 0 50.9   
## 2 0 1 0 2.59  
## 3 0 1 0 38.8   
## 4 0 11 0 2.23  
## 5 0 2 0 12.4   
## 6 0 10 0 2.88  
## 7 0 1 0 10.2   
## 8 0 1 0 5   
## 9 0 6 0 2.64  
## 10 0 1 0 26.0

Of course, since there is no right answer here, no target value to predict, there is no need to split the data into testing and training data sets (though we could use hold out data to see what the clusters look like on new data). Rather, all of the data will be used to form clusters. Our first task then, is to standardize our data. Recall, that k-means uses a distance function to determine which cluster mean or centroid each data point belongs to. Thus, we need all of the variables to be on similar scales. This function takes the existing dataframe and converts all of the variables using z-score standardization. That is, the mean of each feature is subtracted from each individual feature and then divided by that featureâ€™s standard deviation. This transformation rescales the feature such it has a mean of zero and a standard deviation of one. Thus, it is measured in how many standard deviations it falls above or below its mean. The as.data.frame() function is used here to keep clustering\_input2 as a dataframe, instead of a matrix.

# Z-score standardization

clustering\_input2 <- as.data.frame(scale(clustering\_input2))

Next, let’s run the model. The first line below sets the starting point of the random number generator in the kmeans function. That way if we run this again we can get similar results. You can put any number into the set.seed() function, but you want to keep it the same every time. If you try to replicate my results, you will want to use the number I put here. (Of course, replicating results is not always possible due to differences in R versions and package versions, etc.)

To implement k-means we use the kmeans() function that comes with the stats package. The main inputs to the function are, of course, first, the data. Second, we enter k, the number of centers. Here I have picked 7 centers. This number was selected to have a realistic number of clusters for use in our business case. Too many clusters would not allow TECA to market to each group. Too few clusters would not be enough information to have meaningful groups. The inter.max option is the number of iterations that are allowed that the k-means procedure will go through to find new centroids and new cluster groups. The default is 10, but we selected a higher number here to ensure that the model converges (finds a solution). nstart is the number of random starting positions for centroids the algorithm will try. The default is 1, but it is advisable to make this greater than 1 since the outcome is so dependent on the initial random positioning of the centroids. Nevertheless, we will select one here to reduce the computation time.

# Run the model

set.seed(777)  
clusters <- kmeans(clustering\_input2, centers=7, iter.max=20, nstart=1)

Next, let’s check the size of the clusters. This is one by just printing the size “element” of the clusters object. Ideally, we split our clusters into fairly even groups. Sure enough, our results show a fairly even split. The largest of our clusters is about less than twice the size of the next largest, and overall, several of the clusters are fairly similar in size.

# Size of the k clusters

clusters$size

## [1] 691 2690 54301 27224 135268 43054 9189

Of course, what we really want to know is what these clusters are. We want to know what types of customers they represent. We can examine this by just printing the centers “element” of the clusters object. We interpret this data by looking at each cluster and subjectively identifying what the pattern of purchases mean. Thus, we look for high, positive amounts to indicate that this is column that defines that cluster. We just pick the cluster, which is indicated by a row and read the numbers across the row. Any high, positive numbers indicate high “loadings” of these variables on that cluster, and we can use those to define the cluster. For example, let’s look at cluster three. Cluster three has negative numbers for all but one column–fuel. This column has the only positive number for this cluster and is the highest number for all other clusters for fuel. Thus, it is safe to say that this cluster is comprised of customers that mostly purchase only gas. So, if we were to label this cluster, we would likely call it something like “Gas only”. Let’s think about how this cluster affects TECA. This is a problematic cluster because it looks like these customers don’t really even come into the store. TECA knows that the profitability, the gross profit margin, of fuel is quite low. So. while TECA appreciates this revenue, they would like to market to these customers to encourage them to come into the store and buy some higher-margin items.

Below, I have pasted a quick analysis and labeling of the clusters from Excel. I have added colors to indicate the size of the mean in each row. Green indicates higher numbers. For example, in cluster 7, the 0.90 in the area.cooler column is clearly the highest number in that row. This column then defines this cluster. Next, I added arrows to indicate the size of that number within each column. This is not needed but helps indicate the strength of that cluster loading. For example, cluster one has a 1.18 for lottery. While this is not that high for this cluster, no other cluster has a loading this high. This indicates that while lottery is not the main column that defines cluster 1, it is important. Finally, I pasted the size of each column.

Let’s talk a bit about these labels. Cluster 1 is interesting. It is small but significant. These loyalty customers seem to come into the store and buy a lot of different products. These are loyal customers that TECA will want to keep happy. Cluster 2 is clearly individuals that come into the store to purchase soda and coffee, mainly refilling a cup they already have. They occasionally purchase food and lottery tickets, and miscellaneous items, but they mainly get a refill. While soda and coffee refills are high profit, they are low revenue. TECA should work with these customers to encourage them to spend more money. Cluster 4 seems to come into the stores only to buy cigarettes, cigars, and other tobacco products. These items are low profit and similar to cluster 3, TECA should work to expand this group’s purchasing habits. Cluster 5 is a great cluster for TECA and they need to encourage and expand it. This cluster appears to be getting meals and the convenience stores. These are profitable items. Further, this cluster is large. Clusters 6 and 7 each have one or two items they focus on. These are profitable items, but TECA should work to expand these habits even more and try to convert these loyalty customers to purchase more like the customers in cluster 1.

# A matrix indicating the mean values for each feature and cluster combination

clusters$centers

## customer\_id area.fresh area.snacks area.cooler area.grocery  
## 1 0.11259951 -0.390167992 -0.3987902 -0.465270604 -0.205707054  
## 2 -0.29704612 0.884405830 0.4587509 0.118434855 0.114094906  
## 3 -0.01333504 -0.236222271 -0.3558500 -0.264656741 -0.131790700  
## 4 -0.02721843 -0.170303007 -0.2003601 -0.111448912 -0.050314033  
## 5 0.03017179 0.005978093 -0.4226683 -0.006745433 -0.007074293  
## 6 0.01176481 -0.139344018 1.5145051 -0.149231390 -0.067594763  
## 7 -0.26133973 2.235788625 1.7180469 2.692955996 1.330775354  
## area.nongrocery area.alcohol area.tobacco area.fuel area.dispensed  
## 1 -0.12133350 -0.136071823 -0.2768854 1.4831316 -0.417225270  
## 2 0.35667143 0.070681198 0.4708858 0.3900085 4.581096879  
## 3 -0.09489001 -0.087590485 -0.2701521 1.4371469 -0.233252047  
## 4 -0.02901367 0.015963350 1.9342146 -0.2653425 -0.151612043  
## 5 -0.01136452 0.002561231 -0.3000301 -0.4978769 -0.006697117  
## 6 -0.08064356 -0.074483505 -0.2761326 -0.3083327 -0.162604448  
## 7 1.09654696 0.771130505 1.4593762 0.8415471 1.378293090  
## area.lottery area.miscellaneous refill.no refill.yes mean\_revenue  
## 1 -0.14427266 -0.125712016 -0.30178931 -0.14268696 14.43906246  
## 2 0.57057404 0.441797683 1.47515426 7.30762394 -0.23147115  
## 3 -0.07244517 -0.070701693 -0.07716424 -0.11332912 0.80950960  
## 4 -0.04185402 -0.022057190 0.16889581 -0.09671666 -0.06551272  
## 5 -0.02212520 -0.009043723 -0.30096015 -0.05999071 -0.28797918  
## 6 -0.07442516 -0.063238710 0.03896564 -0.09641792 -0.25771484  
## 7 1.07032972 0.792697203 3.79422114 0.16258440 -0.16088523

Find the picture of the Excel output, called “seven clusters.png”, below, or open it outside of RStudio to view. Examining the clusters in Excel

The next step is to add these labels to the actual customers, so that a marketing campaign can be undertaken. The following code reads in the original dataframe with the customer id number present. It then adds the cluster number, 1 through 7. Finally, I add the labels to the clusters in a new column called cluster\_labels. I do this with the mutate() function from the tidyverse package.

Let’s look at three loyalty customers. As we scan the purchases of these three customers, we see that they are labeled accurately. The first customer bought snacks and was labeled as such. The second customer purchased both snacks and tobacco but seems appropriately labeled. The third customer purchased 10 of gas and was correctly labeled as such.

# Add cluster to the original a more complete dataframe and add labels

clustering\_input1 <- readRDS("C:/Users/User/OneDrive/Documents/projects/k-means.rds")  
clustering\_input1$cluster <- clusters$cluster  
clustering\_input1 <- clustering\_input1 %>% mutate(cluster\_labels = case\_when(  
 cluster==1 ~ 'everything',   
 cluster==2 ~ 'fountain soda/coffee',  
 cluster==3 ~ 'gas only',  
 cluster==4 ~ 'cigarettes',  
 cluster==5 ~ 'meals',  
 cluster==6 ~ 'snacks',  
 cluster==7 ~ 'cooler'))  
  
clustering\_input1[53:55, ]

## # A tibble: 3 × 17  
## customer\_id area.fresh area.snacks area.cooler area.grocery area.nongrocery  
## <dbl> <int> <int> <int> <int> <int>  
## 1 4.22 0 1 0 0 0  
## 2 4.24 0 1 0 0 0  
## 3 4.36 0 0 0 0 0  
## area.alcohol area.tobacco area.fuel area.dispensed area.lottery  
## <int> <int> <int> <int> <int>  
## 1 0 0 0 0 0  
## 2 0 1 0 0 0  
## 3 0 0 1 0 0  
## area.miscellaneous refill.no refill.yes mean\_revenue cluster cluster\_labels  
## <int> <int> <int> <dbl> <int> <chr>   
## 1 0 1 0 1.49 6 snacks   
## 2 0 2 0 8.89 4 cigarettes   
## 3 0 1 0 10 3 gas only

Finally, let’s try to visualize the multi-dimensional clustering space in two dimensions. This is done with the factoextra package. It uses something called principal component analysis to collapse our 12-dimensional space into two dimensions. The result is a bit messy, but kind of cool. It highlights that our clusters, at least in two dimensions are not easy to separate and are quite dense. It further illustrates that there are some significant outliers in our data. This is potentially problematic since k-means is very affected by outliers since it measures clusters using distance.

# Visualize the clustering

fviz\_cluster(clusters, clustering\_input2, geom = "point", show.clust.cent = FALSE, palette = "jco", ggtheme = theme\_classic())

