Assignemnt5 - Hierarchical Clustering

Pranay Kumar Kodeboyina

2023-04-16

# importing the libraries and including knitr package  
library(knitr)  
# importing the libraries and including dendextend package  
library(dendextend)

##   
## ---------------------  
## Welcome to dendextend version 1.16.0  
## Type citation('dendextend') for how to cite the package.  
##   
## Type browseVignettes(package = 'dendextend') for the package vignette.  
## The github page is: https://github.com/talgalili/dendextend/  
##   
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues  
## You may ask questions at stackoverflow, use the r and dendextend tags:   
## https://stackoverflow.com/questions/tagged/dendextend  
##   
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))  
## ---------------------

##   
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':  
##   
## cutree

# importing the libraries and including factoextra package  
library(factoextra)

## Loading required package: ggplot2

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

# importing the libraries and including readr package  
library(readr)  
# importing the libraries and including cluster package  
library(cluster)  
# importing the libraries and including caret package  
library(caret)

## Loading required package: lattice

#Including the library read r. to read the data set   
library(readr)  
#Extracting the current working directory  
getwd()

## [1] "/Users/kodeboyina/Documents/Kent State/Sem1/Fundamentals of ML/Assignment5"

#setting seed  
set.seed(123)  
  
#Loading Cereals csv data Import the data set into R  
cereals <- read.csv("data/Cereals.csv", header = TRUE, sep = ",", stringsAsFactors = FALSE)  
  
#viewing the data frame to identify the numeric columns  
View(cereals)  
  
#Observing the first 10 observations of the data set  
head(cereals, n=10L)

## name mfr type calories protein fat sodium fiber carbo  
## 1 100%\_Bran N C 70 4 1 130 10.0 5.0  
## 2 100%\_Natural\_Bran Q C 120 3 5 15 2.0 8.0  
## 3 All-Bran K C 70 4 1 260 9.0 7.0  
## 4 All-Bran\_with\_Extra\_Fiber K C 50 4 0 140 14.0 8.0  
## 5 Almond\_Delight R C 110 2 2 200 1.0 14.0  
## 6 Apple\_Cinnamon\_Cheerios G C 110 2 2 180 1.5 10.5  
## 7 Apple\_Jacks K C 110 2 0 125 1.0 11.0  
## 8 Basic\_4 G C 130 3 2 210 2.0 18.0  
## 9 Bran\_Chex R C 90 2 1 200 4.0 15.0  
## 10 Bran\_Flakes P C 90 3 0 210 5.0 13.0  
## sugars potass vitamins shelf weight cups rating  
## 1 6 280 25 3 1.00 0.33 68.40297  
## 2 8 135 0 3 1.00 1.00 33.98368  
## 3 5 320 25 3 1.00 0.33 59.42551  
## 4 0 330 25 3 1.00 0.50 93.70491  
## 5 8 NA 25 3 1.00 0.75 34.38484  
## 6 10 70 25 1 1.00 0.75 29.50954  
## 7 14 30 25 2 1.00 1.00 33.17409  
## 8 8 100 25 3 1.33 0.75 37.03856  
## 9 6 125 25 1 1.00 0.67 49.12025  
## 10 5 190 25 3 1.00 0.67 53.31381

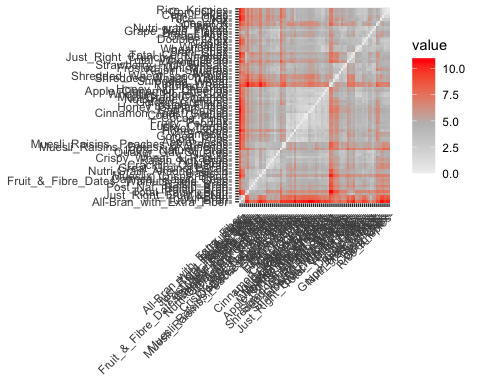
# Set row names to the Cereals column  
rownames(cereals) <- cereals[,1]  
  
#Returns the original vector of column names in their original order  
t(t(names(cereals)))

## [,1]   
## [1,] "name"   
## [2,] "mfr"   
## [3,] "type"   
## [4,] "calories"  
## [5,] "protein"   
## [6,] "fat"   
## [7,] "sodium"   
## [8,] "fiber"   
## [9,] "carbo"   
## [10,] "sugars"   
## [11,] "potass"   
## [12,] "vitamins"  
## [13,] "shelf"   
## [14,] "weight"   
## [15,] "cups"   
## [16,] "rating"

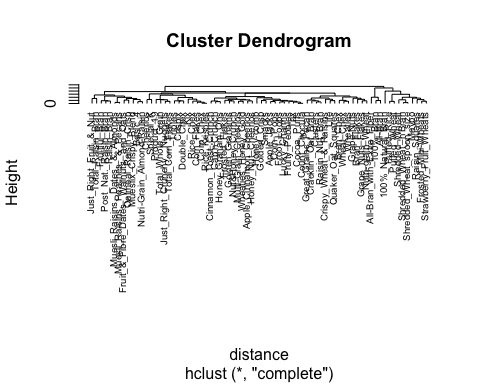
#Only identifying the numeric data from the data frame columns 4 to 16  
cereals\_df <- data.frame(cereals[,4:16])  
  
# Copy the original data to data frame two to perform normalization and fitting data set and scaling for a clustering technique  
cereals\_df1 <- na.omit(cereals\_df)  
  
# Using omit function to remove the NA values from the data set.  
cereals\_df2 <- scale(cereals\_df1)  
  
#The total number of observations after ommiting the NA data are 74 observations(3 omitted dute to empty data)  
head(cereals\_df2)

## calories protein fat sodium  
## 100%\_Bran -1.8659155 1.3817478 0.0000000 -0.3910227  
## 100%\_Natural\_Bran 0.6537514 0.4522084 3.9728810 -1.7804186  
## All-Bran -1.8659155 1.3817478 0.0000000 1.1795987  
## All-Bran\_with\_Extra\_Fiber -2.8737823 1.3817478 -0.9932203 -0.2702057  
## Apple\_Cinnamon\_Cheerios 0.1498180 -0.4773310 0.9932203 0.2130625  
## Apple\_Jacks 0.1498180 -0.4773310 -0.9932203 -0.4514312  
## fiber carbo sugars potass  
## 100%\_Bran 3.22866747 -2.5001396 -0.2542051 2.5605229  
## 100%\_Natural\_Bran -0.07249167 -1.7292632 0.2046041 0.5147738  
## All-Bran 2.81602258 -1.9862220 -0.4836096 3.1248675  
## All-Bran\_with\_Extra\_Fiber 4.87924705 -1.7292632 -1.6306324 3.2659536  
## Apple\_Cinnamon\_Cheerios -0.27881412 -1.0868662 0.6634132 -0.4022862  
## Apple\_Jacks -0.48513656 -0.9583868 1.5810314 -0.9666308  
## vitamins shelf weight cups  
## 100%\_Bran -0.1818422 0.9419715 -0.2008324 -2.0856582  
## 100%\_Natural\_Bran -1.3032024 0.9419715 -0.2008324 0.7567534  
## All-Bran -0.1818422 0.9419715 -0.2008324 -2.0856582  
## All-Bran\_with\_Extra\_Fiber -0.1818422 0.9419715 -0.2008324 -1.3644493  
## Apple\_Cinnamon\_Cheerios -0.1818422 -1.4616799 -0.2008324 -0.3038480  
## Apple\_Jacks -0.1818422 -0.2598542 -0.2008324 0.7567534  
## rating  
## 100%\_Bran 1.8549038  
## 100%\_Natural\_Bran -0.5977113  
## All-Bran 1.2151965  
## All-Bran\_with\_Extra\_Fiber 3.6578436  
## Apple\_Cinnamon\_Cheerios -0.9165248  
## Apple\_Jacks -0.6553998

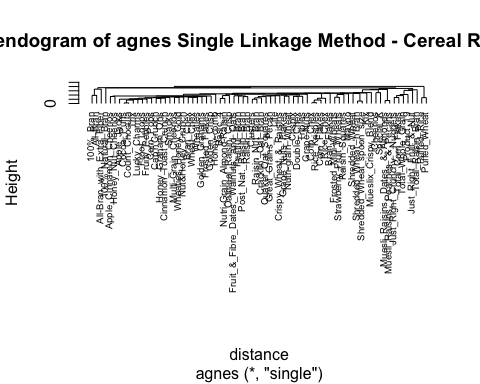
#viewing the data frame post normalization  
View(cereals\_df2)  
  
library(factoextra)  
  
# Compute the Euclidean distance matrix  
distance <- dist(cereals\_df2, method = "euclidean")  
  
# Visualize the distance matrix with the fviz\_dist function  
fviz\_dist(distance,   
 gradient = list(low = "white", mid = "grey", high = "red"), show\_labels = TRUE)



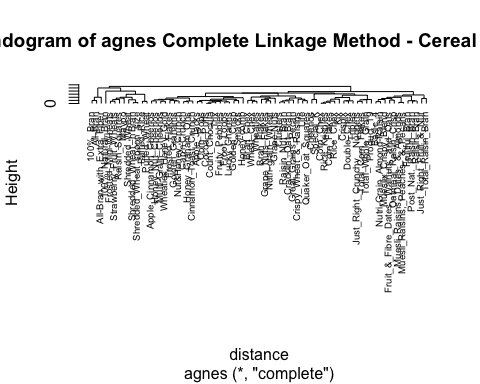
#Use the normalized data to do hierarchical clustering using the Euclidean Dist technique.  
hc1<-hclust(distance, method = "complete")  
  
#the dendogram plotting process.  
plot(hc1, hang = -1, cex = 0.6)



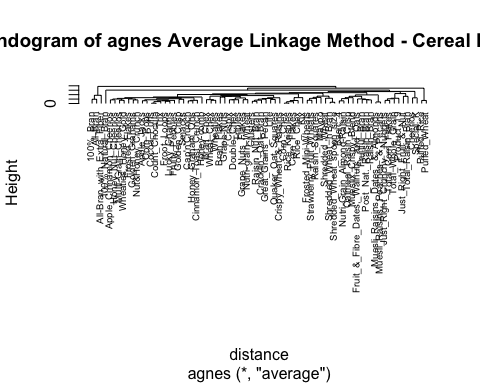
# Q1. Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method.  
  
#Comparing hierarchical clustering with different linkages  
  
#single method - Apply hierarchical clustering  
single\_hc <- agnes(distance, method = "single")  
  
# Plot the resulting dendrogram with modified appearance  
pltree(single\_hc, cex=0.6, hang = -1, main = "Dendogram of agnes Single Linkage Method - Cereal Ratings")



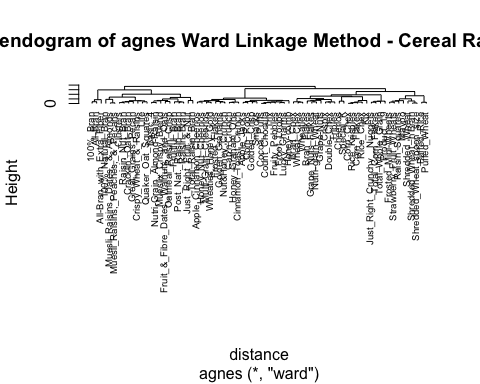
#Complete method - Apply hierarchical clustering  
complete\_hc <- agnes(distance, method = "complete")  
  
# Plot the resulting dendrogram with modified appearance  
pltree(complete\_hc, cex=0.6, hang = -1, main = "Dendogram of agnes Complete Linkage Method - Cereal Ratings")



#Average method - Apply hierarchical clustering  
average\_hc <- agnes(distance, method = "average")  
  
# Plot the resulting dendrogram with modified appearance  
pltree(average\_hc, cex=0.6, hang = -1, main = "Dendogram of agnes Average Linkage Method - Cereal Ratings")



#Ward method - Apply hierarchical clustering  
ward\_hc <- agnes(distance, method = "ward")  
  
# Plot the resulting dendrogram with modified appearance  
pltree(ward\_hc, cex=0.6, hang = -1, main = "Dendogram of agnes Ward Linkage Method - Cereal Ratings")



#Compare Agglomeration coefficients  
# Single  
print(single\_hc$ac)

## [1] 0.6067859

#Complete  
print(complete\_hc$ac)

## [1] 0.8353712

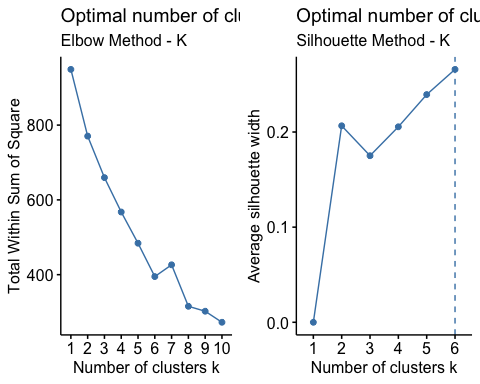
#Average  
print(average\_hc$ac)

## [1] 0.7766075

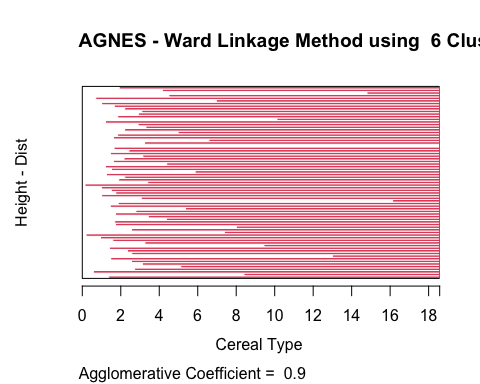
# WARD Best linkage method with the coefficient of 0.9049881  
print(ward\_hc$ac)

## [1] 0.9046042

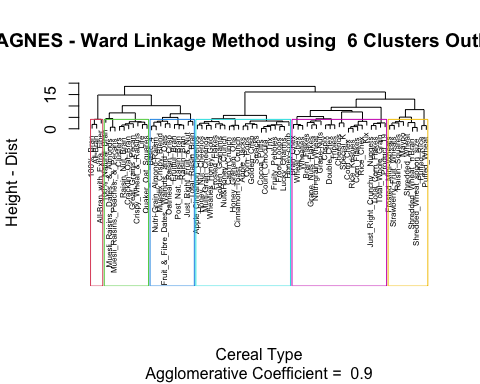
# Q2. How many clusters would you choose?   
  
#Choosing the optimal number of clusters using Elbow and Silhouette methods  
  
library(cowplot)  
  
Elbow\_method <- fviz\_nbclust(cereals\_df2, kmeans, method = "wss",k.max = 10) +labs(subtitle = "Elbow Method - K")  
  
Silhouette <- fviz\_nbclust(cereals\_df2, kmeans, method = "silhouette", k.max = 6) + labs(subtitle = "Silhouette Method - K")  
  
plot\_grid(Elbow\_method, Silhouette, nrow = 1)



#Using the Elbow method and Siloutee stat method to determine the optimal value of k which is k = 6 where there is maximum slope  
  
  
# Plot dendrogram with 6 clusters outlined  
plot(ward\_hc,   
 main = "AGNES - Ward Linkage Method using 6 Clusters Outlined",  
 xlab = "Cereal Type",  
 ylab = "Height - Dist",  
 cex.axis = 1,  
 cex = 0.50,)

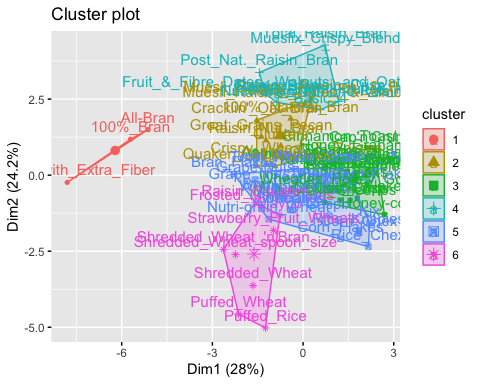


rect.hclust(ward\_hc, k = 6, border = 2:7)



# Cut the tree into 6 clusters  
clusters1 <- cutree(ward\_hc, k = 6)  
cereals\_df2\_6 <- as.data.frame(cbind(cereals\_df2,clusters1))

fviz\_cluster(list(data = cereals\_df2, cluster = clusters1))

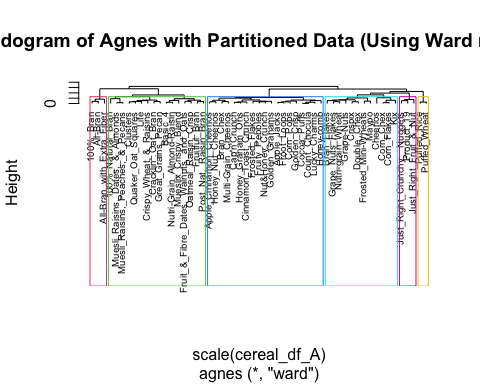


#Creating  
#For the stability of the clusters, partitioning the data into A and B  
set.seed(123)  
  
cereal\_df\_A <-cereals\_df1 [1:55,]  
cereal\_df\_B <-cereals\_df1 [56:74,]

#Performing Hierarchical Clustering while considering k = 6.  
single\_k6 <- agnes(scale(cereal\_df\_A), method = "single")  
complete\_k6 <- agnes(scale(cereal\_df\_A), method = "complete")  
average\_k6 <- agnes(scale(cereal\_df\_A), method = "average")  
ward\_k6 <- agnes(scale(cereal\_df\_A), method = "ward")  
  
  
cbind(single=single\_k6$ac , complete=complete\_k6$ac , average= average\_k6$ac , ward= ward\_k6$ac)

## single complete average ward  
## [1,] 0.6564842 0.8120228 0.7449303 0.8808195

pltree(ward\_k6, cex = 0.6, hang = -1, main = "Dendogram of Agnes with Partitioned Data (Using Ward method)")  
  
rect.hclust(ward\_k6, k = 6, border = 2:7)



cut\_2 <- cutree(ward\_k6, k = 6)

#the centroids are calculated.  
Sb\_result <- as.data.frame(cbind(cereal\_df\_A, cut\_2))  
  
Sb\_result[Sb\_result$cut\_2==1,]

## calories protein fat sodium fiber carbo sugars potass  
## 100%\_Bran 70 4 1 130 10 5 6 280  
## All-Bran 70 4 1 260 9 7 5 320  
## All-Bran\_with\_Extra\_Fiber 50 4 0 140 14 8 0 330  
## vitamins shelf weight cups rating cut\_2  
## 100%\_Bran 25 3 1 0.33 68.40297 1  
## All-Bran 25 3 1 0.33 59.42551 1  
## All-Bran\_with\_Extra\_Fiber 25 3 1 0.50 93.70491 1

one\_centroid <- colMeans(Sb\_result[Sb\_result$cut\_2==1,])  
  
Sb\_result[Sb\_result$cut\_2==2,]

## calories protein fat sodium fiber carbo  
## 100%\_Natural\_Bran 120 3 5 15 2.0 8.0  
## Basic\_4 130 3 2 210 2.0 18.0  
## Clusters 110 3 2 140 2.0 13.0  
## Cracklin'\_Oat\_Bran 110 3 3 140 4.0 10.0  
## Crispy\_Wheat\_&\_Raisins 100 2 1 140 2.0 11.0  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats 120 3 2 160 5.0 12.0  
## Fruitful\_Bran 120 3 0 240 5.0 14.0  
## Great\_Grains\_Pecan 120 3 3 75 3.0 13.0  
## Life 100 4 2 150 2.0 12.0  
## Muesli\_Raisins,\_Dates,\_&\_Almonds 150 4 3 95 3.0 16.0  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans 150 4 3 150 3.0 16.0  
## Mueslix\_Crispy\_Blend 160 3 2 150 3.0 17.0  
## Nutri-Grain\_Almond-Raisin 140 3 2 220 3.0 21.0  
## Oatmeal\_Raisin\_Crisp 130 3 2 170 1.5 13.5  
## Post\_Nat.\_Raisin\_Bran 120 3 1 200 6.0 11.0  
## Quaker\_Oat\_Squares 100 4 1 135 2.0 14.0  
## sugars potass vitamins shelf weight cups  
## 100%\_Natural\_Bran 8 135 0 3 1.00 1.00  
## Basic\_4 8 100 25 3 1.33 0.75  
## Clusters 7 105 25 3 1.00 0.50  
## Cracklin'\_Oat\_Bran 7 160 25 3 1.00 0.50  
## Crispy\_Wheat\_&\_Raisins 10 120 25 3 1.00 0.75  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats 10 200 25 3 1.25 0.67  
## Fruitful\_Bran 12 190 25 3 1.33 0.67  
## Great\_Grains\_Pecan 4 100 25 3 1.00 0.33  
## Life 6 95 25 2 1.00 0.67  
## Muesli\_Raisins,\_Dates,\_&\_Almonds 11 170 25 3 1.00 1.00  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans 11 170 25 3 1.00 1.00  
## Mueslix\_Crispy\_Blend 13 160 25 3 1.50 0.67  
## Nutri-Grain\_Almond-Raisin 7 130 25 3 1.33 0.67  
## Oatmeal\_Raisin\_Crisp 10 120 25 3 1.25 0.50  
## Post\_Nat.\_Raisin\_Bran 14 260 25 3 1.33 0.67  
## Quaker\_Oat\_Squares 6 110 25 3 1.00 0.50  
## rating cut\_2  
## 100%\_Natural\_Bran 33.98368 2  
## Basic\_4 37.03856 2  
## Clusters 40.40021 2  
## Cracklin'\_Oat\_Bran 40.44877 2  
## Crispy\_Wheat\_&\_Raisins 36.17620 2  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats 40.91705 2  
## Fruitful\_Bran 41.01549 2  
## Great\_Grains\_Pecan 45.81172 2  
## Life 45.32807 2  
## Muesli\_Raisins,\_Dates,\_&\_Almonds 37.13686 2  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans 34.13976 2  
## Mueslix\_Crispy\_Blend 30.31335 2  
## Nutri-Grain\_Almond-Raisin 40.69232 2  
## Oatmeal\_Raisin\_Crisp 30.45084 2  
## Post\_Nat.\_Raisin\_Bran 37.84059 2  
## Quaker\_Oat\_Squares 49.51187 2

two\_centroid <- colMeans(Sb\_result[Sb\_result$cut\_2==2,])  
  
Sb\_result[Sb\_result$cut\_2==3,]

## calories protein fat sodium fiber carbo sugars potass  
## Apple\_Cinnamon\_Cheerios 110 2 2 180 1.5 10.5 10 70  
## Apple\_Jacks 110 2 0 125 1.0 11.0 14 30  
## Bran\_Chex 90 2 1 200 4.0 15.0 6 125  
## Cap'n'Crunch 120 1 2 220 0.0 12.0 12 35  
## Cinnamon\_Toast\_Crunch 120 1 3 210 0.0 13.0 9 45  
## Cocoa\_Puffs 110 1 1 180 0.0 12.0 13 55  
## Corn\_Pops 110 1 0 90 1.0 13.0 12 20  
## Count\_Chocula 110 1 1 180 0.0 12.0 13 65  
## Froot\_Loops 110 2 1 125 1.0 11.0 13 30  
## Frosted\_Flakes 110 1 0 200 1.0 14.0 11 25  
## Fruity\_Pebbles 110 1 1 135 0.0 13.0 12 25  
## Golden\_Crisp 100 2 0 45 0.0 11.0 15 40  
## Golden\_Grahams 110 1 1 280 0.0 15.0 9 45  
## Honey\_Graham\_Ohs 120 1 2 220 1.0 12.0 11 45  
## Honey\_Nut\_Cheerios 110 3 1 250 1.5 11.5 10 90  
## Honey-comb 110 1 0 180 0.0 14.0 11 35  
## Lucky\_Charms 110 2 1 180 0.0 12.0 12 55  
## Multi-Grain\_Cheerios 100 2 1 220 2.0 15.0 6 90  
## Nut&Honey\_Crunch 120 2 1 190 0.0 15.0 9 40  
## vitamins shelf weight cups rating cut\_2  
## Apple\_Cinnamon\_Cheerios 25 1 1 0.75 29.50954 3  
## Apple\_Jacks 25 2 1 1.00 33.17409 3  
## Bran\_Chex 25 1 1 0.67 49.12025 3  
## Cap'n'Crunch 25 2 1 0.75 18.04285 3  
## Cinnamon\_Toast\_Crunch 25 2 1 0.75 19.82357 3  
## Cocoa\_Puffs 25 2 1 1.00 22.73645 3  
## Corn\_Pops 25 2 1 1.00 35.78279 3  
## Count\_Chocula 25 2 1 1.00 22.39651 3  
## Froot\_Loops 25 2 1 1.00 32.20758 3  
## Frosted\_Flakes 25 1 1 0.75 31.43597 3  
## Fruity\_Pebbles 25 2 1 0.75 28.02576 3  
## Golden\_Crisp 25 1 1 0.88 35.25244 3  
## Golden\_Grahams 25 2 1 0.75 23.80404 3  
## Honey\_Graham\_Ohs 25 2 1 1.00 21.87129 3  
## Honey\_Nut\_Cheerios 25 1 1 0.75 31.07222 3  
## Honey-comb 25 1 1 1.33 28.74241 3  
## Lucky\_Charms 25 2 1 1.00 26.73451 3  
## Multi-Grain\_Cheerios 25 1 1 1.00 40.10596 3  
## Nut&Honey\_Crunch 25 2 1 0.67 29.92429 3

three\_centroid <- colMeans(Sb\_result[Sb\_result$cut\_2==3,])  
  
Sb\_result[Sb\_result$cut\_2==4,]

## calories protein fat sodium fiber carbo sugars potass  
## Bran\_Flakes 90 3 0 210 5 13 5 190  
## Cheerios 110 6 2 290 2 17 1 105  
## Corn\_Chex 110 2 0 280 0 22 3 25  
## Corn\_Flakes 100 2 0 290 1 21 2 35  
## Crispix 110 2 0 220 1 21 3 30  
## Double\_Chex 100 2 0 190 1 18 5 80  
## Frosted\_Mini-Wheats 100 3 0 0 3 14 7 100  
## Grape\_Nuts\_Flakes 100 3 1 140 3 15 5 85  
## Grape-Nuts 110 3 0 170 3 17 3 90  
## Kix 110 2 1 260 0 21 3 40  
## Maypo 100 4 1 0 0 16 3 95  
## Nutri-grain\_Wheat 90 3 0 170 3 18 2 90  
## vitamins shelf weight cups rating cut\_2  
## Bran\_Flakes 25 3 1 0.67 53.31381 4  
## Cheerios 25 1 1 1.25 50.76500 4  
## Corn\_Chex 25 1 1 1.00 41.44502 4  
## Corn\_Flakes 25 1 1 1.00 45.86332 4  
## Crispix 25 3 1 1.00 46.89564 4  
## Double\_Chex 25 3 1 0.75 44.33086 4  
## Frosted\_Mini-Wheats 25 2 1 0.80 58.34514 4  
## Grape\_Nuts\_Flakes 25 3 1 0.88 52.07690 4  
## Grape-Nuts 25 3 1 0.25 53.37101 4  
## Kix 25 2 1 1.50 39.24111 4  
## Maypo 25 2 1 1.00 54.85092 4  
## Nutri-grain\_Wheat 25 3 1 1.00 59.64284 4

four\_centroid <- colMeans(Sb\_result[Sb\_result$cut\_2==4,])  
  
  
centroids <- rbind(one\_centroid, two\_centroid, three\_centroid, four\_centroid)  
  
x2 <- as.data.frame(rbind(centroids[,-14], cereal\_df\_B))

#figuring out the Dist.  
Dist\_1 <- get\_dist(x2)  
Matrix\_1 <- as.matrix(Dist\_1)  
dataframe1 <- data.frame(data=seq(1,nrow(cereal\_df\_B),1), Clusters = rep(0,nrow(cereal\_df\_B)))  
for(i in 1:nrow(cereal\_df\_B))   
 {dataframe1[i,2] <- which.min(Matrix\_1[i+4, 1:4])}  
dataframe1

## data Clusters  
## 1 1 1  
## 2 2 2  
## 3 3 2  
## 4 4 3  
## 5 5 3  
## 6 6 2  
## 7 7 2  
## 8 8 2  
## 9 9 3  
## 10 10 4  
## 11 11 2  
## 12 12 3  
## 13 13 2  
## 14 14 4  
## 15 15 4  
## 16 16 3  
## 17 17 4  
## 18 18 4  
## 19 19 3

cbind(cereals\_df2\_6$clusters1[56:74], dataframe1$Clusters)

## [,1] [,2]  
## [1,] 4 1  
## [2,] 2 2  
## [3,] 6 2  
## [4,] 5 3  
## [5,] 5 3  
## [6,] 6 2  
## [7,] 6 2  
## [8,] 6 2  
## [9,] 3 3  
## [10,] 5 4  
## [11,] 6 2  
## [12,] 5 3  
## [13,] 4 2  
## [14,] 5 4  
## [15,] 5 4  
## [16,] 3 3  
## [17,] 5 4  
## [18,] 5 4  
## [19,] 3 3

table(cereals\_df2\_6$clusters1[56:74] == dataframe1$Clusters)

##   
## FALSE TRUE   
## 15 4

#Since we are getting 15 FALSE and 4 TRUE, we can conclude that the model is partially stable.

#The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of “healthy cereals.”Should the data be normalized? If not, how should they be used in the cluster analysis?  
  
#Clustering Healthy SB\_Cereals.  
  
Healthy\_SB\_Cereals <- cereals  
  
#omitting the empty observations  
Healthy\_SB\_Cereals\_NA <- na.omit(Healthy\_SB\_Cereals)  
  
#binding the clusters  
clust <- cbind(Healthy\_SB\_Cereals\_NA, clusters1)  
  
clust[clust$clusters1==1,]

## name mfr type calories protein  
## 100%\_Bran 100%\_Bran N C 70 4  
## All-Bran All-Bran K C 70 4  
## All-Bran\_with\_Extra\_Fiber All-Bran\_with\_Extra\_Fiber K C 50 4  
## fat sodium fiber carbo sugars potass vitamins shelf  
## 100%\_Bran 1 130 10 5 6 280 25 3  
## All-Bran 1 260 9 7 5 320 25 3  
## All-Bran\_with\_Extra\_Fiber 0 140 14 8 0 330 25 3  
## weight cups rating clusters1  
## 100%\_Bran 1 0.33 68.40297 1  
## All-Bran 1 0.33 59.42551 1  
## All-Bran\_with\_Extra\_Fiber 1 0.50 93.70491 1

clust[clust$clusters1==2,]

## name mfr type  
## 100%\_Natural\_Bran 100%\_Natural\_Bran Q C  
## Clusters Clusters G C  
## Cracklin'\_Oat\_Bran Cracklin'\_Oat\_Bran K C  
## Crispy\_Wheat\_&\_Raisins Crispy\_Wheat\_&\_Raisins G C  
## Great\_Grains\_Pecan Great\_Grains\_Pecan P C  
## Life Life Q C  
## Muesli\_Raisins,\_Dates,\_&\_Almonds Muesli\_Raisins,\_Dates,\_&\_Almonds R C  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans Muesli\_Raisins,\_Peaches,\_&\_Pecans R C  
## Quaker\_Oat\_Squares Quaker\_Oat\_Squares Q C  
## Raisin\_Nut\_Bran Raisin\_Nut\_Bran G C  
## calories protein fat sodium fiber carbo  
## 100%\_Natural\_Bran 120 3 5 15 2.0 8.0  
## Clusters 110 3 2 140 2.0 13.0  
## Cracklin'\_Oat\_Bran 110 3 3 140 4.0 10.0  
## Crispy\_Wheat\_&\_Raisins 100 2 1 140 2.0 11.0  
## Great\_Grains\_Pecan 120 3 3 75 3.0 13.0  
## Life 100 4 2 150 2.0 12.0  
## Muesli\_Raisins,\_Dates,\_&\_Almonds 150 4 3 95 3.0 16.0  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans 150 4 3 150 3.0 16.0  
## Quaker\_Oat\_Squares 100 4 1 135 2.0 14.0  
## Raisin\_Nut\_Bran 100 3 2 140 2.5 10.5  
## sugars potass vitamins shelf weight cups  
## 100%\_Natural\_Bran 8 135 0 3 1 1.00  
## Clusters 7 105 25 3 1 0.50  
## Cracklin'\_Oat\_Bran 7 160 25 3 1 0.50  
## Crispy\_Wheat\_&\_Raisins 10 120 25 3 1 0.75  
## Great\_Grains\_Pecan 4 100 25 3 1 0.33  
## Life 6 95 25 2 1 0.67  
## Muesli\_Raisins,\_Dates,\_&\_Almonds 11 170 25 3 1 1.00  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans 11 170 25 3 1 1.00  
## Quaker\_Oat\_Squares 6 110 25 3 1 0.50  
## Raisin\_Nut\_Bran 8 140 25 3 1 0.50  
## rating clusters1  
## 100%\_Natural\_Bran 33.98368 2  
## Clusters 40.40021 2  
## Cracklin'\_Oat\_Bran 40.44877 2  
## Crispy\_Wheat\_&\_Raisins 36.17620 2  
## Great\_Grains\_Pecan 45.81172 2  
## Life 45.32807 2  
## Muesli\_Raisins,\_Dates,\_&\_Almonds 37.13686 2  
## Muesli\_Raisins,\_Peaches,\_&\_Pecans 34.13976 2  
## Quaker\_Oat\_Squares 49.51187 2  
## Raisin\_Nut\_Bran 39.70340 2

clust[clust$clusters1==3,]

## name mfr type calories protein fat  
## Apple\_Cinnamon\_Cheerios Apple\_Cinnamon\_Cheerios G C 110 2 2  
## Apple\_Jacks Apple\_Jacks K C 110 2 0  
## Cap'n'Crunch Cap'n'Crunch Q C 120 1 2  
## Cinnamon\_Toast\_Crunch Cinnamon\_Toast\_Crunch G C 120 1 3  
## Cocoa\_Puffs Cocoa\_Puffs G C 110 1 1  
## Corn\_Pops Corn\_Pops K C 110 1 0  
## Count\_Chocula Count\_Chocula G C 110 1 1  
## Froot\_Loops Froot\_Loops K C 110 2 1  
## Frosted\_Flakes Frosted\_Flakes K C 110 1 0  
## Fruity\_Pebbles Fruity\_Pebbles P C 110 1 1  
## Golden\_Crisp Golden\_Crisp P C 100 2 0  
## Golden\_Grahams Golden\_Grahams G C 110 1 1  
## Honey\_Graham\_Ohs Honey\_Graham\_Ohs Q C 120 1 2  
## Honey\_Nut\_Cheerios Honey\_Nut\_Cheerios G C 110 3 1  
## Honey-comb Honey-comb P C 110 1 0  
## Lucky\_Charms Lucky\_Charms G C 110 2 1  
## Multi-Grain\_Cheerios Multi-Grain\_Cheerios G C 100 2 1  
## Nut&Honey\_Crunch Nut&Honey\_Crunch K C 120 2 1  
## Smacks Smacks K C 110 2 1  
## Trix Trix G C 110 1 1  
## Wheaties\_Honey\_Gold Wheaties\_Honey\_Gold G C 110 2 1  
## sodium fiber carbo sugars potass vitamins shelf weight  
## Apple\_Cinnamon\_Cheerios 180 1.5 10.5 10 70 25 1 1  
## Apple\_Jacks 125 1.0 11.0 14 30 25 2 1  
## Cap'n'Crunch 220 0.0 12.0 12 35 25 2 1  
## Cinnamon\_Toast\_Crunch 210 0.0 13.0 9 45 25 2 1  
## Cocoa\_Puffs 180 0.0 12.0 13 55 25 2 1  
## Corn\_Pops 90 1.0 13.0 12 20 25 2 1  
## Count\_Chocula 180 0.0 12.0 13 65 25 2 1  
## Froot\_Loops 125 1.0 11.0 13 30 25 2 1  
## Frosted\_Flakes 200 1.0 14.0 11 25 25 1 1  
## Fruity\_Pebbles 135 0.0 13.0 12 25 25 2 1  
## Golden\_Crisp 45 0.0 11.0 15 40 25 1 1  
## Golden\_Grahams 280 0.0 15.0 9 45 25 2 1  
## Honey\_Graham\_Ohs 220 1.0 12.0 11 45 25 2 1  
## Honey\_Nut\_Cheerios 250 1.5 11.5 10 90 25 1 1  
## Honey-comb 180 0.0 14.0 11 35 25 1 1  
## Lucky\_Charms 180 0.0 12.0 12 55 25 2 1  
## Multi-Grain\_Cheerios 220 2.0 15.0 6 90 25 1 1  
## Nut&Honey\_Crunch 190 0.0 15.0 9 40 25 2 1  
## Smacks 70 1.0 9.0 15 40 25 2 1  
## Trix 140 0.0 13.0 12 25 25 2 1  
## Wheaties\_Honey\_Gold 200 1.0 16.0 8 60 25 1 1  
## cups rating clusters1  
## Apple\_Cinnamon\_Cheerios 0.75 29.50954 3  
## Apple\_Jacks 1.00 33.17409 3  
## Cap'n'Crunch 0.75 18.04285 3  
## Cinnamon\_Toast\_Crunch 0.75 19.82357 3  
## Cocoa\_Puffs 1.00 22.73645 3  
## Corn\_Pops 1.00 35.78279 3  
## Count\_Chocula 1.00 22.39651 3  
## Froot\_Loops 1.00 32.20758 3  
## Frosted\_Flakes 0.75 31.43597 3  
## Fruity\_Pebbles 0.75 28.02576 3  
## Golden\_Crisp 0.88 35.25244 3  
## Golden\_Grahams 0.75 23.80404 3  
## Honey\_Graham\_Ohs 1.00 21.87129 3  
## Honey\_Nut\_Cheerios 0.75 31.07222 3  
## Honey-comb 1.33 28.74241 3  
## Lucky\_Charms 1.00 26.73451 3  
## Multi-Grain\_Cheerios 1.00 40.10596 3  
## Nut&Honey\_Crunch 0.67 29.92429 3  
## Smacks 0.75 31.23005 3  
## Trix 1.00 27.75330 3  
## Wheaties\_Honey\_Gold 0.75 36.18756 3

clust[clust$clusters1==4,]

## name  
## Basic\_4 Basic\_4  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats  
## Fruitful\_Bran Fruitful\_Bran  
## Just\_Right\_Fruit\_&\_Nut Just\_Right\_Fruit\_&\_Nut  
## Mueslix\_Crispy\_Blend Mueslix\_Crispy\_Blend  
## Nutri-Grain\_Almond-Raisin Nutri-Grain\_Almond-Raisin  
## Oatmeal\_Raisin\_Crisp Oatmeal\_Raisin\_Crisp  
## Post\_Nat.\_Raisin\_Bran Post\_Nat.\_Raisin\_Bran  
## Raisin\_Bran Raisin\_Bran  
## Total\_Raisin\_Bran Total\_Raisin\_Bran  
## mfr type calories protein fat sodium  
## Basic\_4 G C 130 3 2 210  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats P C 120 3 2 160  
## Fruitful\_Bran K C 120 3 0 240  
## Just\_Right\_Fruit\_&\_Nut K C 140 3 1 170  
## Mueslix\_Crispy\_Blend K C 160 3 2 150  
## Nutri-Grain\_Almond-Raisin K C 140 3 2 220  
## Oatmeal\_Raisin\_Crisp G C 130 3 2 170  
## Post\_Nat.\_Raisin\_Bran P C 120 3 1 200  
## Raisin\_Bran K C 120 3 1 210  
## Total\_Raisin\_Bran G C 140 3 1 190  
## fiber carbo sugars potass vitamins shelf  
## Basic\_4 2.0 18.0 8 100 25 3  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats 5.0 12.0 10 200 25 3  
## Fruitful\_Bran 5.0 14.0 12 190 25 3  
## Just\_Right\_Fruit\_&\_Nut 2.0 20.0 9 95 100 3  
## Mueslix\_Crispy\_Blend 3.0 17.0 13 160 25 3  
## Nutri-Grain\_Almond-Raisin 3.0 21.0 7 130 25 3  
## Oatmeal\_Raisin\_Crisp 1.5 13.5 10 120 25 3  
## Post\_Nat.\_Raisin\_Bran 6.0 11.0 14 260 25 3  
## Raisin\_Bran 5.0 14.0 12 240 25 2  
## Total\_Raisin\_Bran 4.0 15.0 14 230 100 3  
## weight cups rating clusters1  
## Basic\_4 1.33 0.75 37.03856 4  
## Fruit\_&\_Fibre\_Dates,\_Walnuts,\_and\_Oats 1.25 0.67 40.91705 4  
## Fruitful\_Bran 1.33 0.67 41.01549 4  
## Just\_Right\_Fruit\_&\_Nut 1.30 0.75 36.47151 4  
## Mueslix\_Crispy\_Blend 1.50 0.67 30.31335 4  
## Nutri-Grain\_Almond-Raisin 1.33 0.67 40.69232 4  
## Oatmeal\_Raisin\_Crisp 1.25 0.50 30.45084 4  
## Post\_Nat.\_Raisin\_Bran 1.33 0.67 37.84059 4  
## Raisin\_Bran 1.33 0.75 39.25920 4  
## Total\_Raisin\_Bran 1.50 1.00 28.59278 4

#Mean ratings are used to select the best cluster.  
mean(clust[clust$clusters1==1,"rating"])

## [1] 73.84446

mean(clust[clust$clusters1==2,"rating"])

## [1] 40.26405

mean(clust[clust$clusters1==3,"rating"])

## [1] 28.84825

mean(clust[clust$clusters1==4,"rating"])

## [1] 36.25917

# Based on the mean ratings for each cluster, it seems that cluster 1 has the highest mean rating at 73.8, Therefore, Group 1 may be considered of as the cluster for a healthy diet.