FML\_Assignment\_5

Swetha

4/14/2022

# installing required packages  
library(ISLR)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(cluster)

## Warning: package 'cluster' was built under R version 4.1.3

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.1.3

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

library(NbClust)  
library(ppclust)

## Warning: package 'ppclust' was built under R version 4.1.3

library(dendextend)

## Warning: package 'dendextend' was built under R version 4.1.3

##   
## ---------------------  
## Welcome to dendextend version 1.15.2  
## 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

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.6 v purrr 0.3.4  
## v tidyr 1.2.0 v stringr 1.4.0  
## v readr 2.1.2 v forcats 0.5.1

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()

library(ggplot2)  
library(proxy)

##   
## Attaching package: 'proxy'

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

# to import the "cereal" data set  
Cereals <- read.csv("Cereals.csv")  
#using head to get first few rows of the data set  
head(Cereals)

## 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  
## sugars potass vitamins shelf weight cups rating  
## 1 6 280 25 3 1 0.33 68.40297  
## 2 8 135 0 3 1 1.00 33.98368  
## 3 5 320 25 3 1 0.33 59.42551  
## 4 0 330 25 3 1 0.50 93.70491  
## 5 8 NA 25 3 1 0.75 34.38484  
## 6 10 70 25 1 1 0.75 29.50954

#using str to analyse the structure of the data set  
str(Cereals)

## 'data.frame': 77 obs. of 16 variables:  
## $ name : chr "100%\_Bran" "100%\_Natural\_Bran" "All-Bran" "All-Bran\_with\_Extra\_Fiber" ...  
## $ mfr : chr "N" "Q" "K" "K" ...  
## $ type : chr "C" "C" "C" "C" ...  
## $ calories: int 70 120 70 50 110 110 110 130 90 90 ...  
## $ protein : int 4 3 4 4 2 2 2 3 2 3 ...  
## $ fat : int 1 5 1 0 2 2 0 2 1 0 ...  
## $ sodium : int 130 15 260 140 200 180 125 210 200 210 ...  
## $ fiber : num 10 2 9 14 1 1.5 1 2 4 5 ...  
## $ carbo : num 5 8 7 8 14 10.5 11 18 15 13 ...  
## $ sugars : int 6 8 5 0 8 10 14 8 6 5 ...  
## $ potass : int 280 135 320 330 NA 70 30 100 125 190 ...  
## $ vitamins: int 25 0 25 25 25 25 25 25 25 25 ...  
## $ shelf : int 3 3 3 3 3 1 2 3 1 3 ...  
## $ weight : num 1 1 1 1 1 1 1 1.33 1 1 ...  
## $ cups : num 0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...  
## $ rating : num 68.4 34 59.4 93.7 34.4 ...

# using summary to analyse the summary of the data set  
summary(Cereals)

## name mfr type calories   
## Length:77 Length:77 Length:77 Min. : 50.0   
## Class :character Class :character Class :character 1st Qu.:100.0   
## Mode :character Mode :character Mode :character Median :110.0   
## Mean :106.9   
## 3rd Qu.:110.0   
## Max. :160.0   
##   
## protein fat sodium fiber   
## Min. :1.000 Min. :0.000 Min. : 0.0 Min. : 0.000   
## 1st Qu.:2.000 1st Qu.:0.000 1st Qu.:130.0 1st Qu.: 1.000   
## Median :3.000 Median :1.000 Median :180.0 Median : 2.000   
## Mean :2.545 Mean :1.013 Mean :159.7 Mean : 2.152   
## 3rd Qu.:3.000 3rd Qu.:2.000 3rd Qu.:210.0 3rd Qu.: 3.000   
## Max. :6.000 Max. :5.000 Max. :320.0 Max. :14.000   
##   
## carbo sugars potass vitamins   
## Min. : 5.0 Min. : 0.000 Min. : 15.00 Min. : 0.00   
## 1st Qu.:12.0 1st Qu.: 3.000 1st Qu.: 42.50 1st Qu.: 25.00   
## Median :14.5 Median : 7.000 Median : 90.00 Median : 25.00   
## Mean :14.8 Mean : 7.026 Mean : 98.67 Mean : 28.25   
## 3rd Qu.:17.0 3rd Qu.:11.000 3rd Qu.:120.00 3rd Qu.: 25.00   
## Max. :23.0 Max. :15.000 Max. :330.00 Max. :100.00   
## NA's :1 NA's :1 NA's :2   
## shelf weight cups rating   
## Min. :1.000 Min. :0.50 Min. :0.250 Min. :18.04   
## 1st Qu.:1.000 1st Qu.:1.00 1st Qu.:0.670 1st Qu.:33.17   
## Median :2.000 Median :1.00 Median :0.750 Median :40.40   
## Mean :2.208 Mean :1.03 Mean :0.821 Mean :42.67   
## 3rd Qu.:3.000 3rd Qu.:1.00 3rd Qu.:1.000 3rd Qu.:50.83   
## Max. :3.000 Max. :1.50 Max. :1.500 Max. :93.70   
##

Now I am scaling the data to remove NA values from the data set.

# Here I am Creating duplicate of data set for preprocessing  
Scaled\_Cereals <- Cereals  
# Now I am scaling the data set to place it into a clustering algorithm  
Scaled\_Cereals[ , c(4:16)] <- scale(Cereals[ , c(4:16)])  
# Here I am Removing NA values from data set using omit function  
Preprocessed\_Cereal <- na.omit(Scaled\_Cereals)  
#Using head to display first few rows after removing NA  
head(Preprocessed\_Cereal)

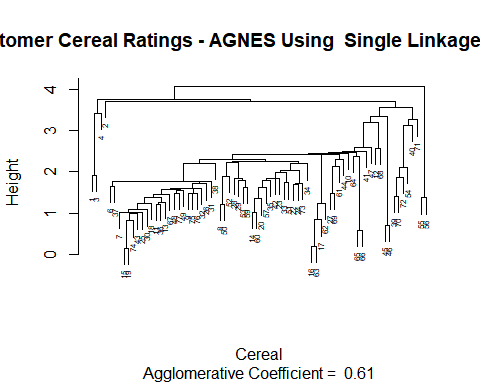
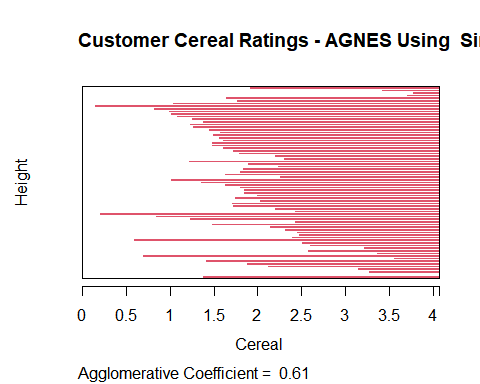
## name mfr type calories protein fat  
## 1 100%\_Bran N C -1.8929836 1.3286071 -0.01290349  
## 2 100%\_Natural\_Bran Q C 0.6732089 0.4151897 3.96137277  
## 3 All-Bran K C -1.8929836 1.3286071 -0.01290349  
## 4 All-Bran\_with\_Extra\_Fiber K C -2.9194605 1.3286071 -1.00647256  
## 6 Apple\_Cinnamon\_Cheerios G C 0.1599704 -0.4982277 0.98066557  
## 7 Apple\_Jacks K C 0.1599704 -0.4982277 -1.00647256  
## sodium fiber carbo sugars potass vitamins shelf  
## 1 -0.3539844 3.29284661 -2.5087829 -0.2343906 2.5753685 -0.1453172 0.9515734  
## 2 -1.7257708 -0.06375361 -1.7409943 0.2223705 0.5160205 -1.2642598 0.9515734  
## 3 1.1967306 2.87327158 -1.9969238 -0.4627711 3.1434645 -0.1453172 0.9515734  
## 4 -0.2346986 4.97114672 -1.7409943 -1.6046739 3.2854885 -0.1453172 0.9515734  
## 6 0.2424445 -0.27354112 -1.1011705 0.6791317 -0.4071355 -0.1453172 -1.4507595  
## 7 -0.4136273 -0.48332864 -0.9732057 1.5926539 -0.9752315 -0.1453172 -0.2495930  
## weight cups rating  
## 1 -0.1967771 -2.1100340 1.8321876  
## 2 -0.1967771 0.7690100 -0.6180571  
## 3 -0.1967771 -2.1100340 1.1930986  
## 4 -0.1967771 -1.3795303 3.6333849  
## 6 -0.1967771 -0.3052601 -0.9365625  
## 7 -0.1967771 0.7690100 -0.6756899

The total number of observations,after pre-processing and scaling the data, went from 77 to 74. So, there were only 3 records with “NA” value.

## Q) 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.

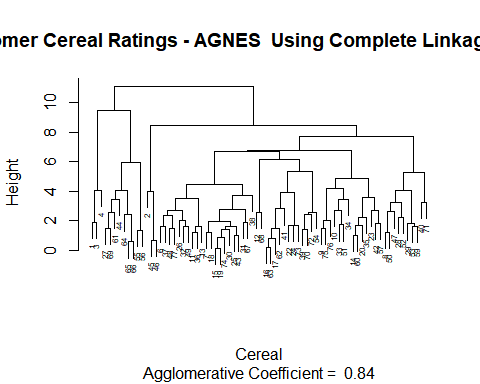
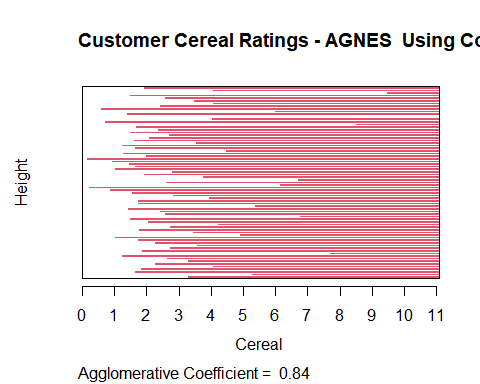
## Single Linkage:

# Creating the dissimilarity matrix for all the numeric values in the data set through Euclidean distance measurements  
Cereal\_Euclidean <- dist(Preprocessed\_Cereal[ , c(4:16)], method = "euclidean")  
# Performing an hierarchical clustering through the single linkage method  
HC\_Single <- agnes(Cereal\_Euclidean, method = "single")  
# Here I am Plotting the results of the different methods  
plot(HC\_Single,   
 main = "Customer Cereal Ratings - AGNES Using Single Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.50)



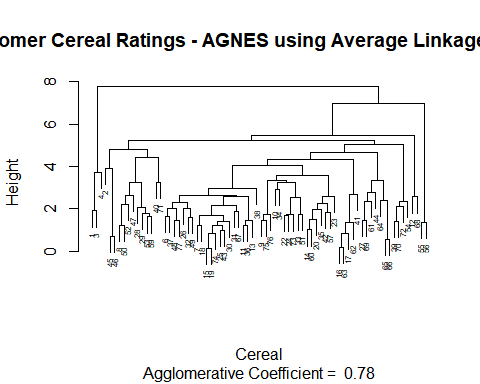
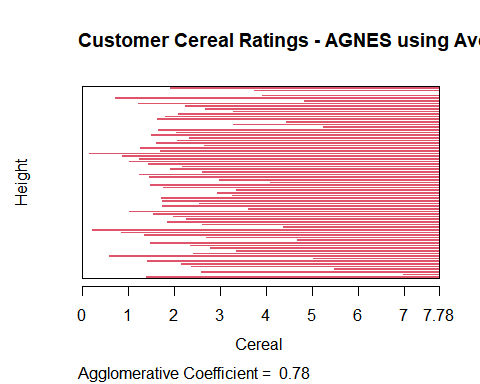
# Complete Linkage:

# Performing the hierarchical clustering through the complete linkage method  
HC\_Complete <- agnes(Cereal\_Euclidean, method = "complete")  
# Here I am Plotting the results of the different methods  
plot(HC\_Complete,   
 main = "Customer Cereal Ratings - AGNES Using Complete Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.50)



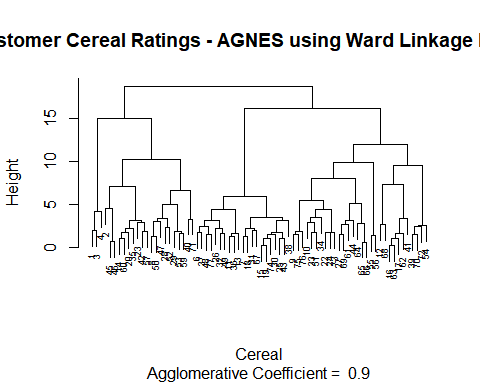
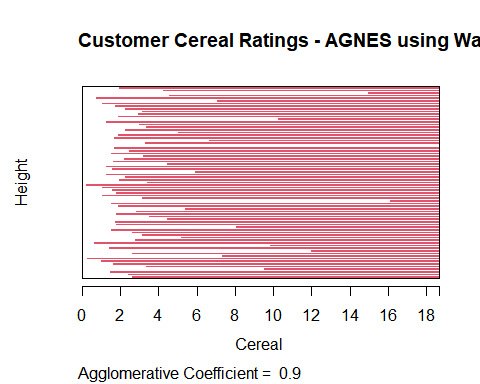
# Average Linkage:

# Performing the hierarchical clustering through the average linkage method  
HC\_Average <- agnes(Cereal\_Euclidean, method = "average")  
# Here I am Plotting the results of the different methods  
plot(HC\_Average,   
 main = "Customer Cereal Ratings - AGNES using Average Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.50)



# Ward Method:

# Performing the hierarchical clustering through the ward linkage method  
HC\_Ward <- agnes(Cereal\_Euclidean, method = "ward")  
#Here I am Plotting the results of the different methods  
plot(HC\_Ward,   
 main = "Customer Cereal Ratings - AGNES using Ward Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.55)

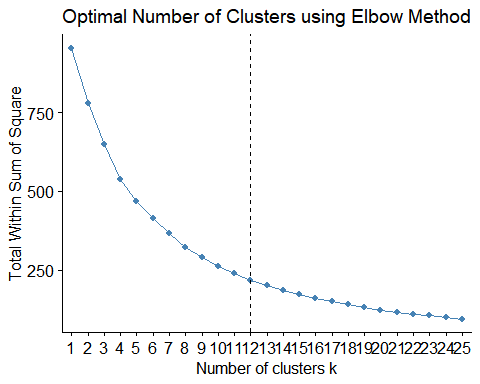
 The clustering structure is closer if the value is close to 1.0. As a result, the method with the closest value to 1.0 will be chosen. Single Linkage: 0.61 Complete Linkage: 0.84 Average Linkage: 0.78 Ward Method: 0.90 Here From the result, The best clustering model is the Ward method.

## Q) How many clusters would you choose?

# Here I am using elbow and silhouette methods to determine the appropriate number of clusters.

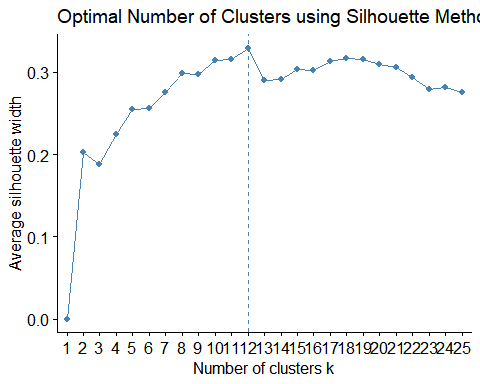
## Elbow Method:

fviz\_nbclust(Preprocessed\_Cereal[ , c(4:16)], hcut, method = "wss", k.max = 25) +  
 labs(title = "Optimal Number of Clusters using Elbow Method") +  
 geom\_vline(xintercept = 12, linetype = 2)

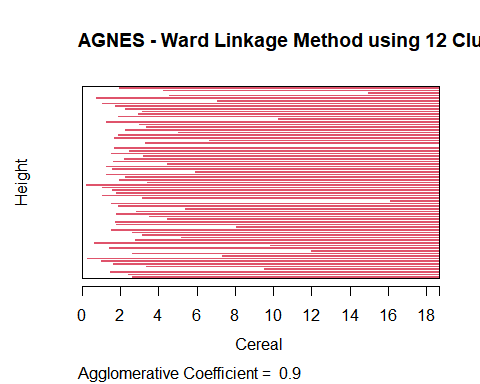


##Silhouette Method:

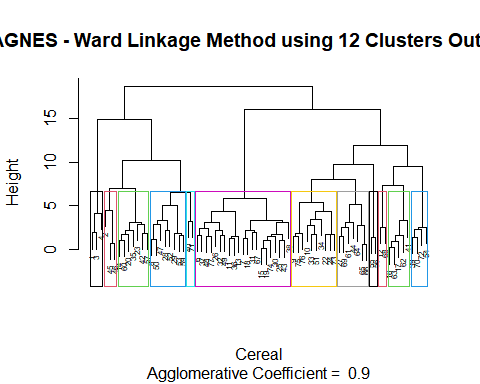
fviz\_nbclust(Preprocessed\_Cereal[ , c(4:16)],   
 hcut,   
 method = "silhouette",   
 k.max = 25) +  
 labs(title = "Optimal Number of Clusters using Silhouette Method")

 Here from the results of the elbow and silhouette methods,we can see that the optimal number of clusters would be 12.

#Here I am Plotting the Ward hierarchical tree with the 12 clusters outlined for reference  
plot(HC\_Ward,   
 main = "AGNES - Ward Linkage Method using 12 Clusters Outlined",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.50,)



rect.hclust(HC\_Ward, k = 12, border = 1:12)



## Q) 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?

Since the nutritional information for cereal is normalized based on the sample of cereal being assessed, normalizing the data would not be appropriate in this circumstance. As a result, the data collected could only include Cereals with a very high sugar content and very little fiber, iron, or other nutritional information. Once the cereal has been normalized throughout the sample set, it’s impossible to say how much nutrition it will supply a child. We might assume that a cereal with an iron level of 0.999 includes almost all of the nutrional iron that a child requires; nevertheless, it could simply be the best of the worst in the sample set with no nutrional value. As a result, converting the data to a ratio of daily suggested calories, fiber, carbs, and other nutrients for a child would be a better approach to preprocess it. This would allow analysts to make more educated cluster judgments during the review process while also preventing a few larger variables from overriding the distance estimations. When examining the clusters, an analyst may examine the cluster average to determine what percentage of a student’s daily nutritional needs would be met by XX cereal. Employees would be able to make well-informed decisions on which “healthy” cereal clusters to choose as a result of this.