Assignment\_5

Krishna Kumar Tavva - 811283461

2023-04-15

# call the libraries

library(readr)

## Warning: package 'readr' was built under R version 4.2.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.1 ✔ purrr 1.0.1  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(caret)

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(ISLR)  
library(dplyr)  
library(factoextra)

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

library(stats)  
library(cluster)

## Warning: package 'cluster' was built under R version 4.2.3

library(ggplot2)  
library(knitr)  
library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.2.3

library(e1071)

## Warning: package 'e1071' was built under R version 4.2.3

library(reshape2)

##   
## Attaching package: 'reshape2'  
##   
## The following object is masked from 'package:tidyr':  
##   
## smiths

library(cowplot)

##   
## Attaching package: 'cowplot'  
##   
## The following object is masked from 'package:lubridate':  
##   
## stamp

library(pander)  
library(kernlab)

##   
## Attaching package: 'kernlab'  
##   
## The following object is masked from 'package:purrr':  
##   
## cross  
##   
## The following object is masked from 'package:ggplot2':  
##   
## alpha

library(tidyr)  
library(fastDummies)  
library(FactoMineR)

## Warning: package 'FactoMineR' was built under R version 4.2.3

# Loading the data

cs <- read.csv("E:\\Fundamentals of Machine Learning\\Module 8\\Cereals.csv")  
summary(cs)

## 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   
##

row.names(cs) <- cs[,1] #changing column name of Cereals data set to row name

# Looking for null values & omitting null values

any(is.na.data.frame(cs))

## [1] TRUE

cs1 <- na.omit(cs) #Remove NA (missing) values

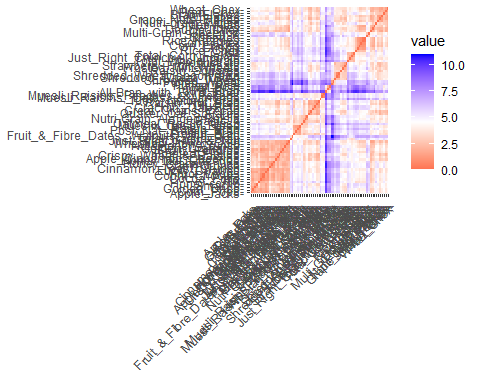
# Normalize the data and finding the optimal k value by Elbow chart & Silhouette method

set.seed(1)  
cs2 <- scale(cs1[,-c(1:3,13)])  
head(cs2)

## 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 weight cups rating  
## 100%\_Bran -0.1818422 -0.2008324 -2.0856582 1.8549038  
## 100%\_Natural\_Bran -1.3032024 -0.2008324 0.7567534 -0.5977113  
## All-Bran -0.1818422 -0.2008324 -2.0856582 1.2151965  
## All-Bran\_with\_Extra\_Fiber -0.1818422 -0.2008324 -1.3644493 3.6578436  
## Apple\_Cinnamon\_Cheerios -0.1818422 -0.2008324 -0.3038480 -0.9165248  
## Apple\_Jacks -0.1818422 -0.2008324 0.7567534 -0.6553998

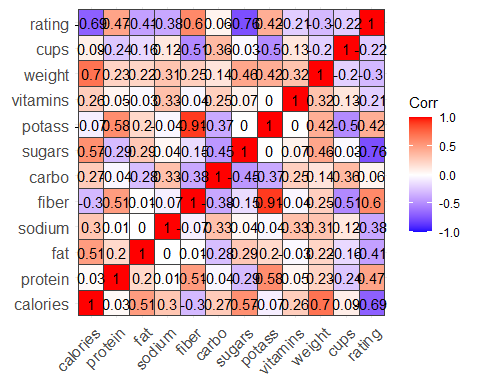
#Q1:(PartA):Using Euclidean distance to the normalized measurements

distance\_table <- get\_dist(cs2) #Compute the distances. Euclidean distance is default.  
fviz\_dist(distance\_table) #fviz\_dist() function visualizes a distance matrix

 #This graph is a distance matrix. As we can see, the diagonal values are zeros (dark orange) because it is showing the distance between any point against itself. The purple and blue represent the furthest distance between any pair of observations.

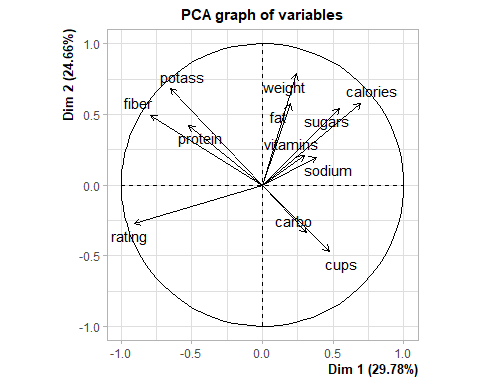
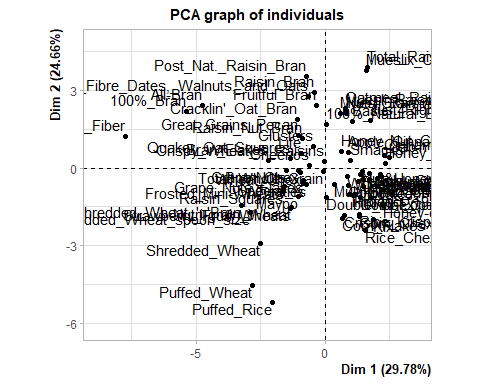
#Looking at the Correlation between Variables.

corr <- cor(cs2)  
ggcorrplot(corr, outline.color = "grey25", lab = TRUE, hc.order = FALSE, type = "full")

 #Sugar and calories are highly negatively correlated with rating. Also, Potass is highly positively correlated with fiber and Protien.

#Trying to Understand the variable variance by performing principle component analysis

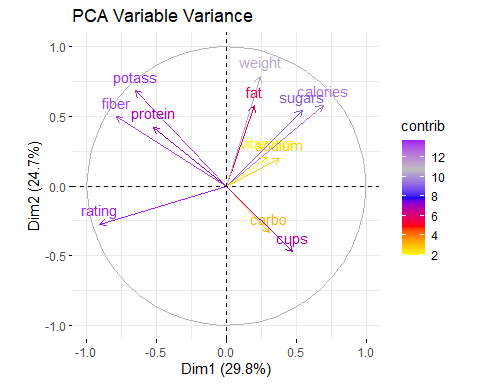
pca\_cereal <- PCA(cs2) #perform principal component analysis



pca\_cereal <- prcomp(cs2, scale = TRUE) #variable has mean zero and standard deviation one  
loadings <- pca\_cereal$rotation #extract loading  
print(loadings[, 1:2])#print loading for the first two PCs

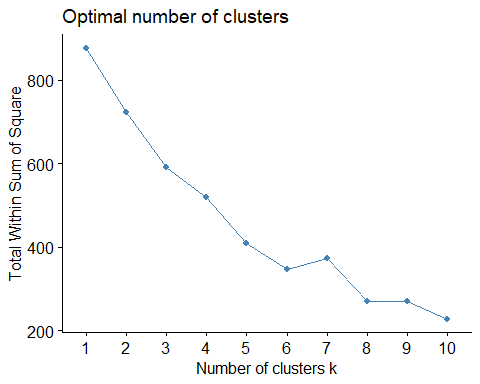
## PC1 PC2  
## calories 0.3670078 0.3370596  
## protein -0.2772241 0.2449194  
## fat 0.1049438 0.3343406  
## sodium 0.2015610 0.1150253  
## fiber -0.4180964 0.2880965  
## carbo 0.1636662 -0.1948813  
## sugars 0.2874024 0.3141683  
## potass -0.3443208 0.3970189  
## vitamins 0.1557868 0.1196450  
## weight 0.1281124 0.4578598  
## cups 0.2510333 -0.2738283  
## rating -0.4799624 -0.1586012

var <- get\_pca\_var(pca\_cereal)  
fviz\_pca\_var(pca\_cereal, col.var="contrib",  
gradient.cols = c("yellow","red","blue","grey","purple"),  
ggrepel = TRUE ) + labs( title = "PCA Variable Variance")

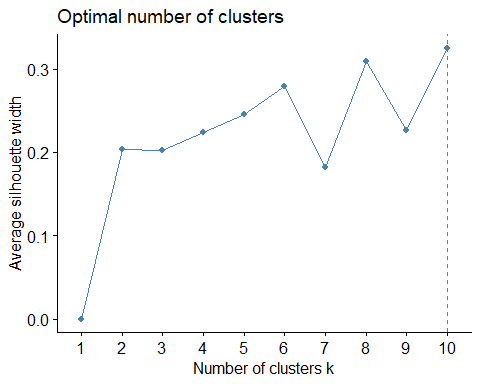


#From PCA Variable Variance, we can infer that Sugar , calories, protien potass and fiber contribute more in the two PCA components/dimensions (Variables)

Elbow <- fviz\_nbclust(cs2, kmeans, method="wss")  
Elbow



silhouette <- fviz\_nbclust(cs2,kmeans,method="silhouette")  
silhouette



set.seed(1)  
k10 <- kmeans(cs2, centers = 10, nstart = 25) # k = 10, number of restarts = 25  
k5 <- kmeans(cs2, centers = 5, nstart = 25) # k = 5, number of restarts = 25

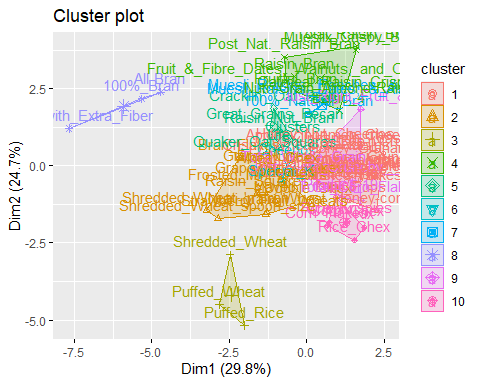
k10$centers

## calories protein fat sodium fiber carbo  
## 1 0.17272410 -0.8998490 1.009294e-17 0.1032288 -0.6351893 -0.56126871  
## 2 -0.57008680 0.2530213 -6.384987e-01 -0.6456014 0.2812040 0.41817811  
## 3 -2.36984887 -0.7871775 -9.932203e-01 -1.9616441 -0.3475883 -0.44446926  
## 4 1.21367739 0.4522084 4.414312e-01 0.3875760 0.6840240 0.08372381  
## 5 -0.01815976 0.7620548 1.158757e+00 -0.3910227 0.1682179 -0.68001481  
## 6 0.14981803 3.2408266 0.000000e+00 1.1795987 -0.2788141 0.45488650  
## 7 1.66161818 1.0719013 2.648587e+00 -0.9145632 0.2026049 -0.35881633  
## 8 -2.20187108 1.3817478 -3.310734e-01 0.1727901 3.6413124 -2.07187492  
## 9 0.25060471 0.0803926 -1.986441e-01 0.5996770 -0.3200786 1.04589172  
## 10 0.07782755 -0.6101224 -7.094430e-01 1.1968583 -0.7798829 1.75803465  
## sugars potass vitamins weight cups rating  
## 1 0.9449551 -0.6940780 -0.1818422 -0.2008324 0.2515215 -0.93991166  
## 2 -0.7621723 0.1620584 -0.3420365 -0.2008324 -0.2947571 0.99351683  
## 3 -1.6306324 -0.6374297 -1.3032024 -2.7429482 0.7567534 1.54110518  
## 4 0.9183071 1.1653377 0.1919445 2.0805535 -0.4923993 -0.43724782  
## 5 -0.1777369 0.2796302 -0.1818422 -0.2008324 -1.3644493 0.08281632  
## 6 -1.1718233 -0.2612000 -0.1818422 -0.2008324 1.2870541 0.68238355  
## 7 0.6634132 0.8439748 -0.5556289 -0.2008324 0.7567534 -0.51910839  
## 8 -0.7894824 2.9837813 -0.1818422 -0.2008324 -1.8452553 2.24264794  
## 9 -0.5294905 -0.4163948 3.1822385 0.1902623 0.5446331 -0.16904450  
## 10 -1.0079629 -0.8759325 -0.1818422 -0.2008324 0.9870554 -0.01518936

k10$size

## [1] 22 14 3 9 6 2 3 3 5 7

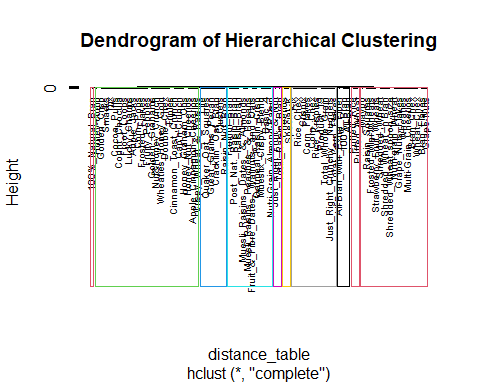
fviz\_cluster(k10, data = cs2)



#After applying both the silhouette method and elbow method, we obtained K value as 10, which we used to plot the 10 clusters. However, upon observing the plot, we noticed that some of the clusters were overlapping, indicating that using only K-means clustering may not be the best option for optimization. Therefore, we will apply hierarchical clustering to obtain an optimal number of clusters.

#Q1:(PartB) Apply hierarchical clustering. Use Agnes to compare the clustering from single linkage, completelinkage, average linkage, and Ward. Choose the best method.

set.seed(1)  
hierarchical\_cluster <- hclust(distance\_table, method = "complete") #hierarchical clustering using Complete Link  
plot(hierarchical\_cluster, cex = 0.6, hang = -1, main = "Dendrogram of Hierarchical Clustering") #Plot the obtained dendrogram  
rect.hclust(hierarchical\_cluster, k = 10, border = 2:10)



# Compute with agnes and with different linkage methods

hc\_single<-agnes(distance\_table, method ="single")  
hc\_complete<-agnes(distance\_table, method ="complete")  
hc\_average<-agnes(distance\_table, method ="average")  
hc\_ward <- agnes(distance\_table, method = "ward")

#Compare Agglomerative coefficients

print(hc\_single$ac)

## [1] 0.6072384

print(hc\_complete$ac)

## [1] 0.8469328

print(hc\_average$ac)

## [1] 0.7881955

print(hc\_ward$ac)

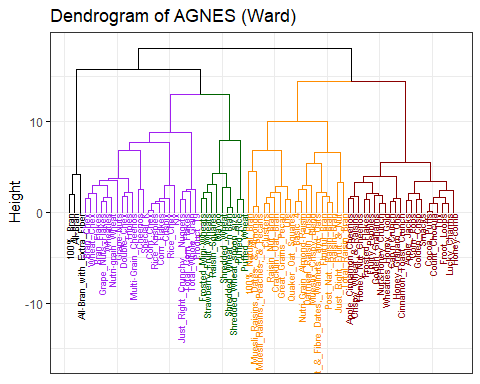
## [1] 0.9087265

#After comparing the Agglomerative coefficients the best linakage method is ward linkage i.e. 0.90 accuracy

#Q2: How many clusters would you choose?

#Utilizing the Ward linkage, 5 clusters seem to be a good number to group the data  
set.seed(1)  
fviz\_dend(hc\_ward, k = 5,main = "Dendrogram of AGNES (Ward)",  
cex = 0.5, k\_colors = c("black", "purple", "darkgreen", "darkorange", "darkred"),   
color\_labels\_by\_k = TRUE,labels\_track\_height = 16,ggtheme = theme\_bw()) #Plot the Dendrogram of AGNES

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as  
## of ggplot2 3.3.4.  
## ℹ The deprecated feature was likely used in the factoextra package.  
## Please report the issue at <https://github.com/kassambara/factoextra/issues>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



cs2\_5 <- cutree(hc\_ward, k = 5)  
Clustered\_df <-as.data.frame(cbind (cs2, cs2\_5 ))

#Q3:Comment on the structure of the clusters and their stability. Hint: To check stability, partition the data, and see how well clusters formed based on one part apply to the other part. #Q3: PartA: Cluster partition A

#We will partition the dataset into two groups: Training A and Validation B.  
set.seed(1) #To get the same random variables  
TrainingA <- cs2[1:55,]  
nrow(TrainingA)

## [1] 55

ValidationB <- cs2[56:74,]  
nrow(ValidationB)

## [1] 19

# Compute the distances. Euclidean distance is used by default. Looking at the cluster of trainingA and ValidationB data set.

set.seed(1) # To maintain same values  
distance\_TrainA <- get\_dist(TrainingA)  
# Compute with AGNES and with different linkage methods For Training Dataset  
hc\_single\_TrainA <- agnes(distance\_TrainA, method = "single")  
hc\_complete\_TrainA <- agnes(distance\_TrainA, method = "complete")  
hc\_average\_TrainA <- agnes(distance\_TrainA, method = "average")  
hc\_ward\_TrainA <- agnes(distance\_TrainA, method = "ward")  
print(hc\_single\_TrainA$ac)

## [1] 0.6663587

print(hc\_complete\_TrainA$ac)

## [1] 0.8285192

print(hc\_average\_TrainA$ac)

## [1] 0.7646836

print(hc\_ward\_TrainA$ac)

## [1] 0.8891086

#It allows us to determine that the best linkage is Ward with 88.91% accuracy for validationA

## Compute with AGNES and with different linkage methods For Training Dataset

set.seed(1) # To maintain same values  
distance\_ValidB <- get\_dist(ValidationB)  
  
# Compare AGNES (agglomerative) coefficients  
hc\_single\_ValidB <- agnes(distance\_ValidB, method = "single")  
hc\_complete\_ValidB <- agnes(distance\_ValidB, method = "complete")  
hc\_average\_ValidB <- agnes(distance\_ValidB, method = "average")  
hc\_ward\_ValidB <- agnes(distance\_ValidB, method = "ward")  
print(hc\_single\_ValidB$ac)

## [1] 0.4805129

print(hc\_complete\_ValidB$ac)

## [1] 0.71298

print(hc\_average\_ValidB$ac)

## [1] 0.6232053

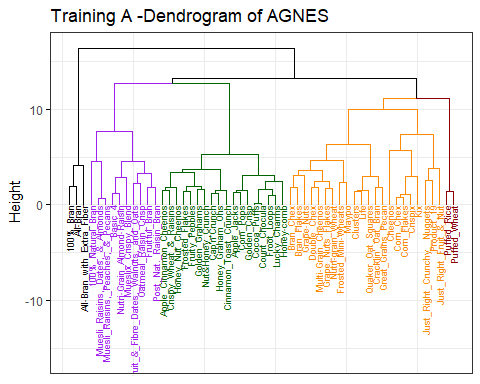
print(hc\_ward\_ValidB$ac)

## [1] 0.7710122

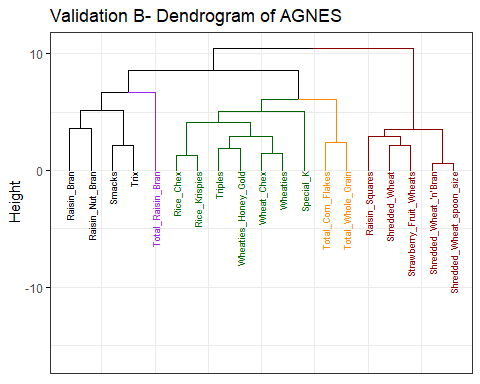
#It allows us to determine that the best linkage is Ward with 77.10% accuracy for validationB

#Dendrogram for TrainingA and ValidationB dataset

fviz\_dend(hc\_ward\_TrainA, k = 5,main = "Training A -Dendrogram of AGNES",  
cex = 0.5, k\_colors = c("black", "purple", "darkgreen", "darkorange", "darkred"),  
color\_labels\_by\_k = TRUE,labels\_track\_height = 16,ggtheme = theme\_bw()) #Plot the Dendrogram of AGNES



fviz\_dend(hc\_ward\_ValidB, k = 5,main = "Validation B- Dendrogram of AGNES",  
cex = 0.5, k\_colors = c("black", "purple", "darkgreen", "darkorange", "darkred"),   
color\_labels\_by\_k = TRUE,labels\_track\_height = 16,ggtheme = theme\_bw()) #Plot the Dendrogram of AGNES



#Q3:PartB: Method1 Use the cluster centroids from A to assign each record in partition B (each record is assigned to the cluster with the closest centroid)

Clustered\_df\_A <-cutree (hc\_ward\_TrainA, k=5)  
Clusters\_A <-as.data.frame(cbind(TrainingA, Clustered\_df\_A))  
nrow(Clusters\_A)#55

## [1] 55

Clust\_1 <- colMeans (Clusters\_A [Clusters\_A$ Clustered\_df\_A == "1" ,]) #This results in a vector of mean values for each column of the data, which represents the centroid of cluster 1   
Clustered\_df\_B <-cutree (hc\_ward\_ValidB, k=5)  
Clusters\_B <-as.data.frame(cbind(ValidationB, Clustered\_df\_B))  
nrow(Clusters\_B)#55

## [1] 19

Clust\_2 <- colMeans (Clusters\_B [Clusters\_B$ Clustered\_df\_B == "1" ,]) #This results in a vector of mean values for each column of the data, which represents the centroid of cluster 2   
Centroid <-rbind(Clust\_1, Clust\_2)  
Centroid

## calories protein fat sodium fiber carbo  
## Clust\_1 -2.201871 1.3817478 -0.3310734 0.1727901 3.64131237 -2.0718749  
## Clust\_2 0.149818 -0.2449462 0.2483051 -0.2702057 -0.02091106 -0.7977876  
## sugars potass vitamins weight cups rating  
## Clust\_1 -0.7894824 2.9837813 -0.1818422 -0.2008324 -1.845255 2.2426479  
## Clust\_2 1.0648712 0.1796942 -0.1818422 0.3369228 -0.303848 -0.5618826  
## Clustered\_df\_A  
## Clust\_1 1  
## Clust\_2 1

#On overall level the both the cluster seems fine but also a slight difference is - #Cluster\_1 has a higher fiber and potassium content compared to Cluster\_2, which may suggest that cereals in this cluster are healthier or more nutrient-dense. #Cluster\_2 has a higher sugar content compared to Cluster\_1, which may suggest that cereals in this cluster are less healthy or have more added sugars.

#Q3:PartB: Method2 Use the cluster centroids from A to assign each record in partition B (each record is assigned to the cluster with the closest centroid)

#In order to predict the calculate distances between each record in data set B and the cluster centroids

distances <- dist(ValidationB[, -1], TrainingA, method = "euclidean") #This line calculates the pairwise distances between the validation and the training samples, using the Euclidean distance metric.  
hc <- hclust(distances) #This line performs hierarchical clustering on the distances object, using the default "complete" linkage method  
clusterB <- cutree(hc, k = 5) #This line cuts the hierarchical tree into five clusters based on the hc object, using the cutree()  
ValidationB$cluster <- clusterB #This line adds a new column to the ValidationB data frame called "cluster"

## Warning in ValidationB$cluster <- clusterB: Coercing LHS to a list

ValidationB$cluster

## Raisin\_Bran Raisin\_Nut\_Bran Raisin\_Squares   
## 1 1 2   
## Rice\_Chex Rice\_Krispies Shredded\_Wheat   
## 3 3 2   
## Shredded\_Wheat\_'n'Bran Shredded\_Wheat\_spoon\_size Smacks   
## 2 2 1   
## Special\_K Strawberry\_Fruit\_Wheats Total\_Corn\_Flakes   
## 4 2 3   
## Total\_Raisin\_Bran Total\_Whole\_Grain Triples   
## 5 3 3   
## Trix Wheat\_Chex Wheaties   
## 1 3 3   
## Wheaties\_Honey\_Gold   
## 3

#The predicted clusters of B on the basis of centroids of A almost classified same except 3 cereals which are “special\_K”, “Total\_CF” and “Total\_WG”. Out of 19 only 3 observation changed their cluster after comparing the validation data set with Training dataset.It means the stability of clusters are really high.

#Q3:PartC: Assess how consistent the cluster assignments are compared to the assignments based on all the data

#Method 1: We are comparing the mean values of each feature for the two clusters identified in the two datasets. These centroids can be used to compare the features of the two clusters and explore differences or similarities between them.Here we can see that Cluster\_1 has a higher fiber and potassium content compared to Cluster\_2, which may suggest that cereals in this cluster are healthier or more nutrient-dense.Cluster\_2 has a higher sugar content compared to Cluster\_1, which may suggest that cereals in this cluster are less healthy or have more added sugars hence cluster 2 rating is really low compared to cluster 1.

#Method 2:This method calculates the pairwise Euclidean distances between the records in the ValidationB dataset and the cluster centroids obtained from the TrainingA dataset using hierarchical clustering with complete linkage method.This enables the prediction of the cluster labels for the validation dataset using the centroids obtained from the training dataset. hence we can see the stability of validation data set on the basis of training dataset. We can see the cereals are cluster exactly the same except “special\_K”, “Total\_CF” and “Total\_WG”. Out of 19 only 3 observation changed their cluster after comparing the validation data set with Training dataset.It means the stability of clusters are really high.

#Q4: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?

#To analyze which group of cereals are healthier to distribute daily in cafeterias in elementary public schools,we will use the non-standardized dataset. In my opinion, it is more meaningful and easier to compare if we look at the variables in their original scale.Here is a table summarizing the number of cereals per cluster:

Healthy\_data <-as.data.frame(cbind (cs1, cs2\_5 ))  
Healthy\_data\_sort <- Healthy\_data[order(Healthy\_data$cs2\_5),c(1,17)]  
Count\_cluster <- Healthy\_data\_sort %>% group\_by(cs2\_5) %>% summarise(count = n())  
print(Count\_cluster)

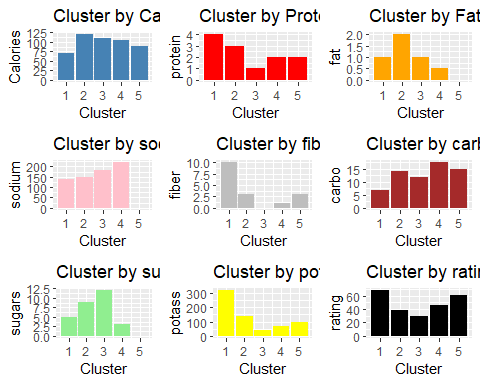
## # A tibble: 5 × 2  
## cs2\_5 count  
## <int> <int>  
## 1 1 3  
## 2 2 19  
## 3 3 21  
## 4 4 22  
## 5 5 9

#Summary table showing the median of each variable  
Healthy\_data\_Var <- Healthy\_data [,4:17]  
cluster\_table <- Healthy\_data\_Var %>% group\_by(cs2\_5) %>%   
summarize(across(.cols = everything(), .fns = median))  
print(cluster\_table)

## # A tibble: 5 × 14  
## cs2\_5 calories protein fat sodium fiber carbo sugars potass vitamins shelf  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 70 4 1 140 10 7 5 320 25 3   
## 2 2 120 3 2 150 3 14 9 140 25 3   
## 3 3 110 1 1 180 0 12 12 40 25 2   
## 4 4 105 2 0.5 220 1 17.5 3 70 25 2.5  
## 5 5 90 2 0 0 3 15 0 95 0 2   
## # ℹ 3 more variables: weight <dbl>, cups <dbl>, rating <dbl>

# Create bar graph

calories <- ggplot(cluster\_table, aes(x = cs2\_5, y = calories)) +   
 geom\_bar(stat = "identity", fill = "steelblue") +  
 labs(x = "Cluster", y = "Calories") +  
 ggtitle("Cluster by Calories")  
  
protein <- ggplot(cluster\_table, aes(x = cs2\_5, y = protein)) +   
 geom\_bar(stat = "identity", fill = "red") +  
 labs(x = "Cluster", y = "protein") +  
 ggtitle("Cluster by Protein")  
  
fat <- ggplot(cluster\_table, aes(x = cs2\_5, y = fat)) +   
 geom\_bar(stat = "identity", fill = "orange") +  
 labs(x = "Cluster", y = "fat") +  
 ggtitle("Cluster by Fat")  
  
sodium <- ggplot(cluster\_table, aes(x = cs2\_5, y = sodium)) +   
 geom\_bar(stat = "identity", fill = "pink") +  
 labs(x = "Cluster", y = "sodium") +  
 ggtitle("Cluster by sodium")  
  
fiber <- ggplot(cluster\_table, aes(x = cs2\_5, y = fiber)) +   
 geom\_bar(stat = "identity", fill = "gray") +  
 labs(x = "Cluster", y = "fiber") +  
 ggtitle("Cluster by fiber")  
  
carbo <- ggplot(cluster\_table, aes(x = cs2\_5, y = carbo)) +   
 geom\_bar(stat = "identity", fill = "brown") +  
 labs(x = "Cluster", y = "carbo") +  
 ggtitle("Cluster by carbo")  
  
sugars <- ggplot(cluster\_table, aes(x = cs2\_5, y = sugars)) +   
 geom\_bar(stat = "identity", fill = "lightgreen") +  
 labs(x = "Cluster", y = "sugars") +  
 ggtitle("Cluster by sugars")  
  
potass <- ggplot(cluster\_table, aes(x = cs2\_5, y = potass)) +   
 geom\_bar(stat = "identity", fill = "yellow") +  
 labs(x = "Cluster", y = "potass") +  
 ggtitle("Cluster by potass")  
  
rating <- ggplot(cluster\_table, aes(x = cs2\_5, y = rating)) +   
 geom\_bar(stat = "identity", fill = "black") +  
 labs(x = "Cluster", y = "rating") +  
 ggtitle("Cluster by rating")  
  
plot\_grid(calories, protein, fat, sodium, fiber, carbo, sugars, potass, rating)



#Based on the graphs, we can see that Cluster 1 has the lowest values for calories, fat, and sugars and the highest values for protein, fiber, and vitamins, which suggests that it may contain cereals that are generally considered healthier options and thats why it has very high rating as well. That why Cluster 1 fits the needs of our client! Nevertheless, part of our client’s petition is to have a different cereal per day, which this cluster does not satisfy this need. For this reason, we will also recommend cluster 5 to satisfy this request. Cluster 5 has zero fats, Zero sugars, and it has the second-lowest number of calories after cluster 1. It also has a good number of proteins and fiber.On the other hand, Cluster 3 has the highest values for calories and sugars and the lowest values for protein, fiber, and vitamins, which suggests that it may contain cereals that are generally considered less healthy. we saw the same insight from our correlation plot high sugar less rating because its less healthy. However, it’s important to note that this is just a general observation and individual cereals within each cluster may vary in terms of their nutritional value.