keerthana\_FML\_A5

Keerthana

2024-04-08

#all Packages loaded in the background

#Taking “cereal\_data” as our dataframe, load the Cerials dataset, then provide an overview of it

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(factoextra)

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

library(ggplot2)  
library(ISLR)  
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(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0  
## ✔ readr 2.1.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

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

library(NbClust)  
library(ppclust)  
library(dendextend)

##   
## ---------------------  
## Welcome to dendextend version 1.17.1  
## 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(cluster)  
library(tinytex)  
cereal\_data <- read.csv("/Users/keerthanavonteddu/Downloads/Cereals.csv")  
head(cereal\_data)

## 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

summary(cereal\_data)

## 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   
##

str(cereal\_data)

## '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 ...

dim(cereal\_data)

## [1] 77 16

#The data will be scaled prior to the NA (Null) values being removed from the data set.

#Make a copy of the data set for preparation.   
cereal\_scaled <- cereal\_data  
#Before feeding the data set into a clustering method, scale it.   
cereal\_scaled[,c(4:16)] <- scale(cereal\_data[,c(4:16)])  
#Take NA values out of the dataset.   
Preprocessed.data <- na.omit(cereal\_scaled)  
head(Preprocessed.data)

## 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

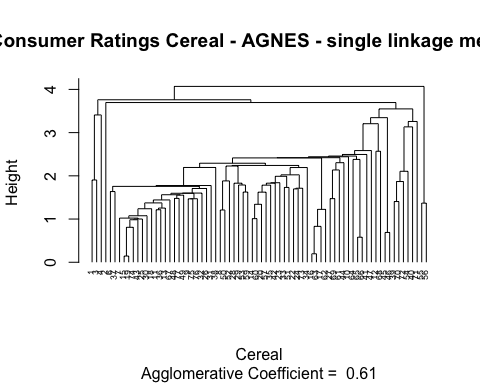
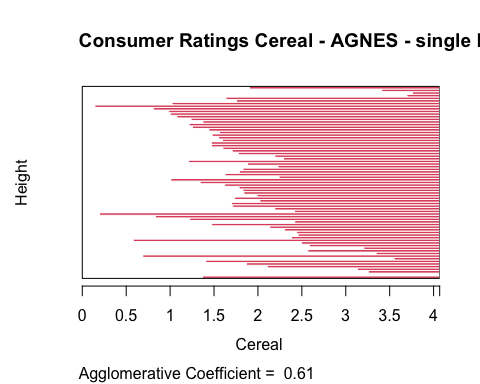
##Task A #Then, use the normalized measures as the Euclidean distance and cluster the data hierarchically. Use Agnes to compare the clustering from complete, single, average, and Ward linkages. Decide on the best course of action.

#single linkage method  
#To generate the dissimilarity matrix for the numerical values in the data set, apply Euclidean distance measurements.  
eucledian\_dist\_df <- dist(Preprocessed.data[,c(4:16)],method = "euclidean")  
#To carry out hierarchical clustering, utilize the single linkage method.   
H.cluster\_single\_Ag <- agnes(eucledian\_dist\_df,method = "single")  
#Ploting outcomes   
plot(H.cluster\_single\_Ag,main = "Consumer Ratings Cereal - AGNES - single linkage method",xlab = "Cereal",ylab = "Height",cex.axis = 1,cex = 0.56,hang = -1)

## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical  
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"  
## is not a graphical parameter

## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a  
## graphical parameter

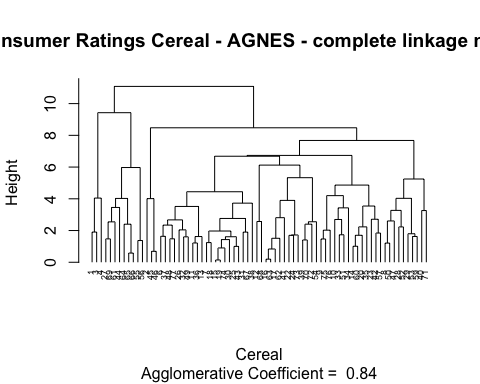
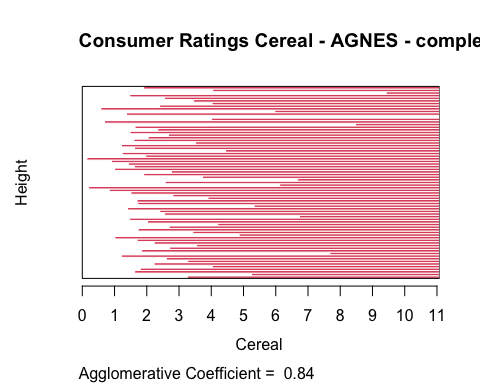


#Complete linkage method  
H.cluster\_complete\_Ag <- agnes(eucledian\_dist\_df,method = "complete")  
#Ploting the outcomes  
plot(H.cluster\_complete\_Ag,main = "Consumer Ratings Cereal - AGNES - complete linkage method",xlab = "Cereal",ylab = "Height",cex.axis = 1,cex = 0.56,hang = -1)

## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical  
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"  
## is not a graphical parameter

## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a  
## graphical parameter

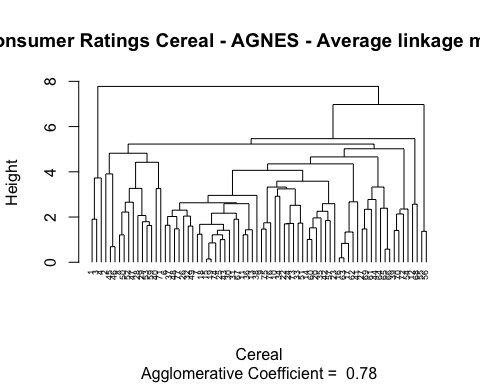
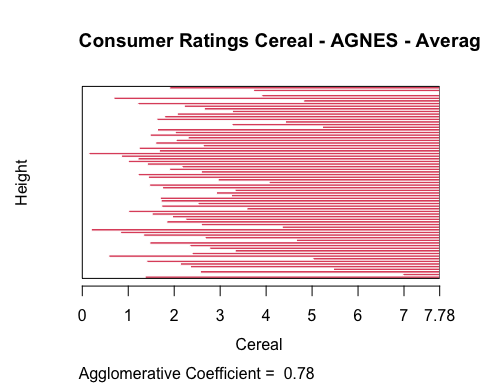


#Average Linkage method  
H.cluster\_average\_Ag <- agnes(eucledian\_dist\_df, method = "average")  
#ploting the outcomes  
plot(H.cluster\_average\_Ag,main = "Consumer Ratings Cereal - AGNES - Average linkage method",xlab = "Cereal",ylab = "Height",cex.axis = 1,cex = 0.56,hang = -1)

## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical  
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"  
## is not a graphical parameter

## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a  
## graphical parameter

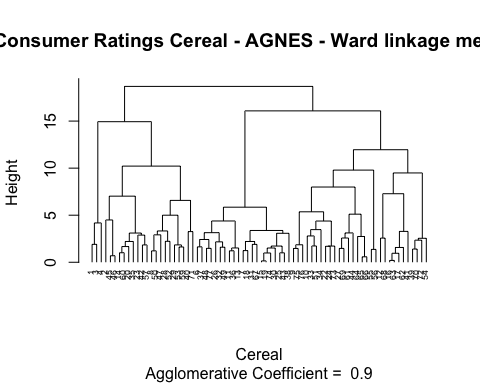
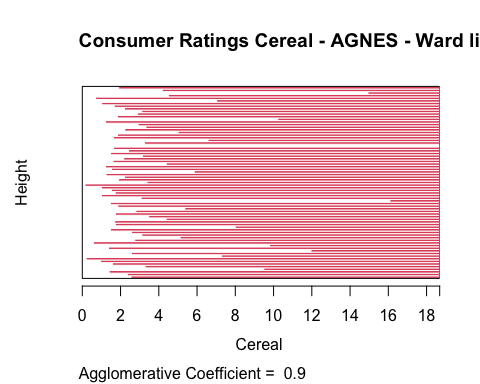


#Ward Linkage Method  
H.cluster\_ward\_Ag <- agnes(eucledian\_dist\_df,method = "ward")  
#ploting the outcomes  
plot(H.cluster\_ward\_Ag,main = "Consumer Ratings Cereal - AGNES - Ward linkage method",xlab = "Cereal",ylab = "Height",cex.axis = 1,cex = 0.56,hang = -1)

## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical  
## parameter

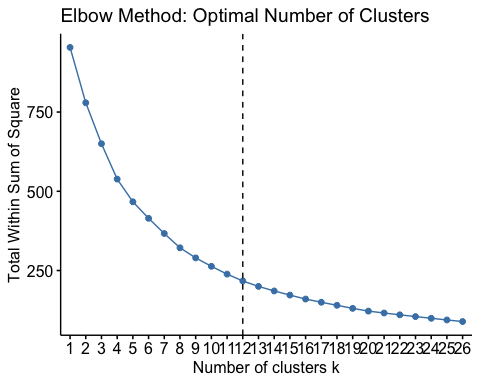
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"  
## is not a graphical parameter

## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a  
## graphical parameter

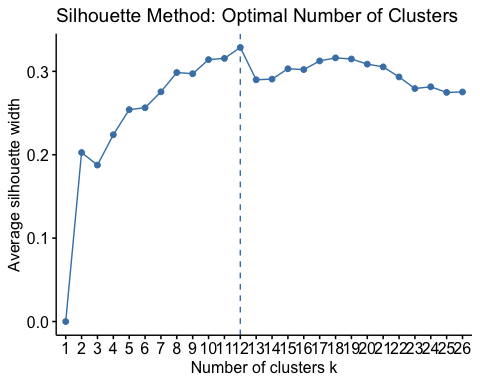
 #The best clustering strategy would be selected based on the agglomerative coefficient that is found for each method. The closer the value is to 1.0, the closer the clustering structure is. The strategy that produces a result that is closest to 1.0 will therefore be chosen.For Ward Method: 0.90, Average Linkage: 0.78, Complete Linkage: 0.84, and Single Linkage: 0.61 #In this instance, the Ward technique will be chosen as the best clustering model.

##Task B #The Ward approach will be selected as the optimal clustering model in this case

#Elbow method  
fviz\_nbclust(Preprocessed.data[,c(4:16)],hcut,method = "wss",k.max = 26) + labs(title = "Elbow Method: Optimal Number of Clusters") + geom\_vline(xintercept = 12,linetype = 2)



#Silhouette Method  
fviz\_nbclust(Preprocessed.data[,c(4:16)],hcut,method = "silhouette",k.max = 26) + labs(title = "Silhouette Method: Optimal Number of Clusters")



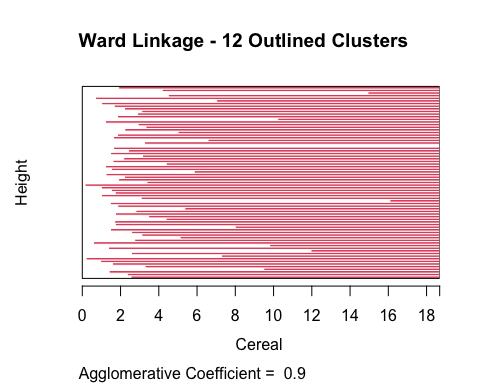
#Elbow method and silhouette concur that in this case, 12 clusters would be the appropriate quantity

#The reference plot displays the 12 clusters of the Ward hierarchical tree.   
plot(H.cluster\_ward\_Ag,main = "Ward Linkage - 12 Outlined Clusters", xlab = "Cereal", ylab = "Height",cex.axis = 1,cex = 0.56,hang = -1)

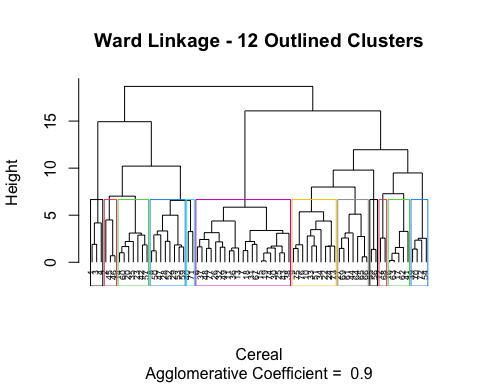
## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical  
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"  
## is not a graphical parameter

## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a  
## graphical parameter



rect.hclust(H.cluster\_ward\_Ag,k=12,border = 1:12)



##Task 3

#Each Cluster Allotted to Data: #The clusters assigned to each data set are located in “cereal\_preprocessed\_1”.

ward\_cluster\_12 <- cutree(H.cluster\_ward\_Ag,k=12)  
cereal\_preprocessed\_1 <- cbind(cluster = ward\_cluster\_12,Preprocessed.data)

#To evaluate the stability of the clusters, a 70-30 divide of the data set will be made. Thirty percent will be assigned based on the centroid nearest to them, following the creation of cluster assignments using the seventy percent.  
  
set.seed(10059)  
cereal\_index <- createDataPartition(Preprocessed.data$protein,p=0.3,list = FALSE)  
preprocessed\_divB <- Preprocessed.data[cereal\_index,]  
preprocessed\_divA <- Preprocessed.data[-cereal\_index,]

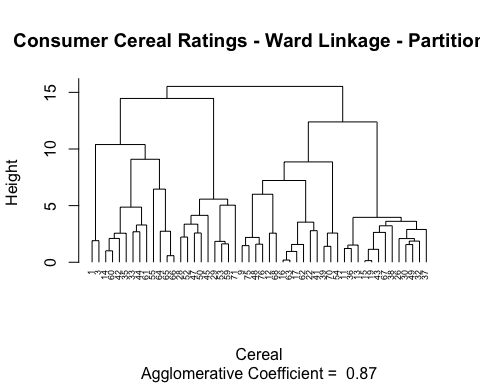
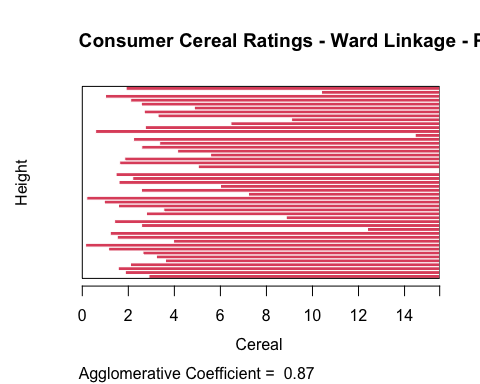
#Retake the Partitioned Data Clustering Process:For the sake of this task, we will utilize the same ward clustering technique and K value (12) to evaluate the stability of the clusters. We will next assign clusters to the nearest places in Partition B for clusters 1 through 12.

#To do hierarchical clustering on partitioned data, use the ward linkage approach   
eucledian\_A\_d <- dist(preprocessed\_divA[,c(4:16)],method = "euclidean")  
H.cluster\_ward\_A\_agnes <- agnes(eucledian\_A\_d,method = "ward")  
plot(H.cluster\_ward\_A\_agnes,main = "Consumer Cereal Ratings - Ward Linkage - Partition A",xlab="Cereal",ylab="Height",cex.axis=1,cex=0.56,hang=-1)

## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical  
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"  
## is not a graphical parameter

## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a  
## graphical parameter



#For analysis, divide the tree into 12 clusters.   
clusters\_12\_ward\_A <- cutree(H.cluster\_ward\_A\_agnes,k=12)  
preprocessed\_cereal\_A <- cbind(cluster = clusters\_12\_ward\_A,preprocessed\_divA)  
  
#We must compute the centroids for each cluster in order to identify which cluster's centroidal is closest to the data points in partition B.  
Ward\_Centroid\_A <- aggregate(preprocessed\_cereal\_A[,5:17],list(preprocessed\_cereal\_A$cluster),mean)  
Ward\_Centroid\_A <- data\_frame(cluster = Ward\_Centroid\_A[,1],centroid = rowMeans(Ward\_Centroid\_A[,-c(1:4)]))

## Warning: `data\_frame()` was deprecated in tibble 1.1.0.  
## ℹ Please use `tibble()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

Ward\_Centroid\_A <- Ward\_Centroid\_A$Centroid

## Warning: Unknown or uninitialised column: `Centroid`.

#Identify the centers of the Partition B data set.  
centers\_dviB\_preprocessed <- data.frame(preprocessed\_divB[,1:3],center = rowMeans(preprocessed\_divB[,4:16]))

#Assign the clusters based on the shortest distance between cluster centers  
preprocessed\_cereal\_B <- cbind(cluster =   
c(4,8,7,3,5,6,7,11,11,10,8,5,10,1,10,1,4,12,12,7,7,1,4,9),   
preprocessed\_divB)   
#To compare the A and B partitions to the original clusters, combine them  
preprocessed\_cereal\_A <- rbind(preprocessed\_cereal\_A,preprocessed\_cereal\_B)  
cereal\_preprocessed\_1 <- cereal\_preprocessed\_1[order(cereal\_preprocessed\_1$name),]  
preprocessed\_cereal\_A <- preprocessed\_cereal\_A[order(preprocessed\_cereal\_A$name),]  
  
#We can compare the number of matched assignments to determine the stability of the clusters after assigning the data using both methods (full data and partitioned data)  
sum(cereal\_preprocessed\_1$cluster == preprocessed\_cereal\_A$cluster)

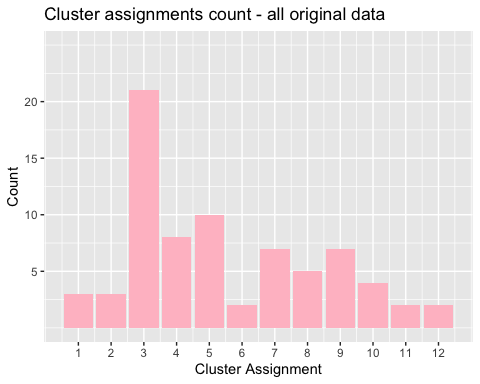
## [1] 18

#This finding clearly shows that the clusters are not very stable. When 70% of the available data were matched, only 35 out of the 74 observations had matching assignments. Consequently, 47% of the assignment is repeatable

#Visualize the cluster allocations to look for changes between the two   
#Plots of the original hierarchical clustering technique.  
ggplot(data = cereal\_preprocessed\_1,aes(cereal\_preprocessed\_1$cluster)) + geom\_bar(fill = "pink") + labs(title = "Cluster assignments count - all original data") + labs(x="Cluster Assignment",y="Count") + guides(fill = FALSE) + scale\_x\_continuous(breaks=c(1:12)) +   
scale\_y\_continuous(breaks=c(5,10,15,20), limits = c(0,25))

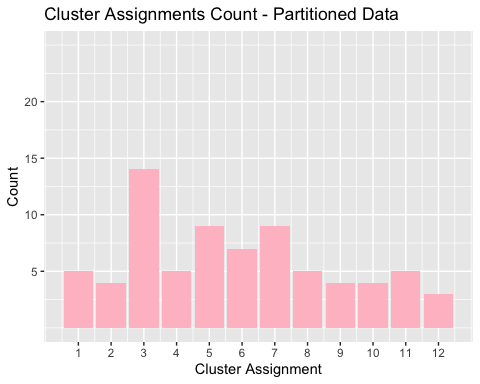
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as  
## of ggplot2 3.3.4.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: Use of `cereal\_preprocessed\_1$cluster` is discouraged.  
## ℹ Use `cluster` instead.



#Plot of the divided algorithm used to assign the remaining data   
ggplot(data = preprocessed\_cereal\_A, aes(preprocessed\_cereal\_A$cluster)) +   
geom\_bar(fill = "pink") +   
labs(title="Cluster Assignments Count - Partitioned Data") +   
labs(x="Cluster Assignment", y="Count") +   
guides(fill=FALSE) +   
scale\_x\_continuous(breaks=c(1:12)) +   
scale\_y\_continuous(breaks=c(5,10,15,20), limits = c(0,25))

## Warning: Use of `preprocessed\_cereal\_A$cluster` is discouraged.  
## ℹ Use `cluster` instead.

 #As can be seen visually, Cluster 3 shrank considerably using the partitioned data. Some of the other clusters became larger as a result. The chart suggests that a partition of the data results in a more uniform distribution of the clusters among the 12 clusters

##Task D In this case, normalizing the data would not be appropriate. It wouldn’t be appropriate since the nutritional data for cereal is adjusted and scaled according to the cereal sample under investigation. Because of this, the data set that was gathered may only include cereals that are incredibly poor in iron, fiber, and other minerals and extremely high in sugar. The amount of nutrients that a child will receive from the cereal is difficult to predict once it has been scaled or normalized throughout the sample set. Uninformed viewers would assume that a cereal with an iron score of 0.999 means it almost fully meets a child’s iron needs; nevertheless, it could just be the best option within the sample set, offering very little in the way of nutrients.

Thus, it would be more acceptable to prepare the data as a ratio to a child’s daily recommended consumption of calories, fiber, carbohydrates, etc. This would allow analysts to make better-informed conclusions about the clusters during evaluation, while preventing a few more important variables from influencing the distance calculations. By examining the cluster average, an analyst can determine the percentage of a student’s daily recommended nutrition that would come from XX cereal. This would enable the staff to make informed decisions when selecting from the “healthy” cereal clusters.