Cluster Analysis on Breakfast Cereal

## Problem Description

* In the following Unsupervised Learning activity, we try to cluster various types of breakfast cereal based on their nutritional content.

setwd("~/R\_KSU/ML/Assignment5")  
cereals\_data <- read.csv("Cereals.csv", header=T)  
data <- cereals\_data  
str(cereals\_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 ...

summary(cereals\_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   
##

head(cereals\_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

tail(cereals\_data)

## name mfr type calories protein fat sodium fiber carbo sugars  
## 72 Total\_Whole\_Grain G C 100 3 1 200 3 16 3  
## 73 Triples G C 110 2 1 250 0 21 3  
## 74 Trix G C 110 1 1 140 0 13 12  
## 75 Wheat\_Chex R C 100 3 1 230 3 17 3  
## 76 Wheaties G C 100 3 1 200 3 17 3  
## 77 Wheaties\_Honey\_Gold G C 110 2 1 200 1 16 8  
## potass vitamins shelf weight cups rating  
## 72 110 100 3 1 1.00 46.65884  
## 73 60 25 3 1 0.75 39.10617  
## 74 25 25 2 1 1.00 27.75330  
## 75 115 25 1 1 0.67 49.78744  
## 76 110 25 1 1 1.00 51.59219  
## 77 60 25 1 1 0.75 36.18756

# Data Pre-Processing

# Total number of NA values in the data set  
colSums(is.na(cereals\_data))

## name mfr type calories protein fat sodium fiber   
## 0 0 0 0 0 0 0 0   
## carbo sugars potass vitamins shelf weight cups rating   
## 1 1 2 0 0 0 0 0

# comment: There are 4 NA values in dataset we shall remove those.   
cereals\_data <- na.omit(cereals\_data)  
  
#check for NA values again  
colSums(is.na(cereals\_data))

## name mfr type calories protein fat sodium fiber   
## 0 0 0 0 0 0 0 0   
## carbo sugars potass vitamins shelf weight cups rating   
## 0 0 0 0 0 0 0 0

# Setting the rownames of the breakfast cereals to the row names, as this will later help us in visualizing the clusters.  
data <- cereals\_data  
rownames(cereals\_data) <- cereals\_data$name  
cereals\_data$name = NULL  
head(cereals\_data)

## mfr type calories protein fat sodium fiber carbo  
## 100%\_Bran N C 70 4 1 130 10.0 5.0  
## 100%\_Natural\_Bran Q C 120 3 5 15 2.0 8.0  
## All-Bran K C 70 4 1 260 9.0 7.0  
## All-Bran\_with\_Extra\_Fiber K C 50 4 0 140 14.0 8.0  
## Apple\_Cinnamon\_Cheerios G C 110 2 2 180 1.5 10.5  
## Apple\_Jacks K C 110 2 0 125 1.0 11.0  
## sugars potass vitamins shelf weight cups rating  
## 100%\_Bran 6 280 25 3 1 0.33 68.40297  
## 100%\_Natural\_Bran 8 135 0 3 1 1.00 33.98368  
## All-Bran 5 320 25 3 1 0.33 59.42551  
## All-Bran\_with\_Extra\_Fiber 0 330 25 3 1 0.50 93.70491  
## Apple\_Cinnamon\_Cheerios 10 70 25 1 1 0.75 29.50954  
## Apple\_Jacks 14 30 25 2 1 1.00 33.17409

## Converting categorical variables into dummy variables   
library(fastDummies)  
cereals\_data <- fastDummies::dummy\_cols(cereals\_data, select\_columns = "mfr")[,-1]  
cereals\_data <- fastDummies::dummy\_cols(cereals\_data, select\_columns = "type")[,-1]  
cereals\_data <- fastDummies::dummy\_cols(cereals\_data, select\_columns = "shelf")[,-10]  
str(cereals\_data)

## 'data.frame': 74 obs. of 24 variables:  
## $ calories: int 70 120 70 50 110 110 130 90 90 120 ...  
## $ protein : int 4 3 4 4 2 2 3 2 3 1 ...  
## $ fat : int 1 5 1 0 2 0 2 1 0 2 ...  
## $ sodium : int 130 15 260 140 180 125 210 200 210 220 ...  
## $ fiber : num 10 2 9 14 1.5 1 2 4 5 0 ...  
## $ carbo : num 5 8 7 8 10.5 11 18 15 13 12 ...  
## $ sugars : int 6 8 5 0 10 14 8 6 5 12 ...  
## $ potass : int 280 135 320 330 70 30 100 125 190 35 ...  
## $ vitamins: int 25 0 25 25 25 25 25 25 25 25 ...  
## $ weight : num 1 1 1 1 1 1 1.33 1 1 1 ...  
## $ cups : num 0.33 1 0.33 0.5 0.75 1 0.75 0.67 0.67 0.75 ...  
## $ rating : num 68.4 34 59.4 93.7 29.5 ...  
## $ mfr\_A : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ mfr\_G : int 0 0 0 0 1 0 1 0 0 0 ...  
## $ mfr\_K : int 0 0 1 1 0 1 0 0 0 0 ...  
## $ mfr\_N : int 1 0 0 0 0 0 0 0 0 0 ...  
## $ mfr\_P : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ mfr\_Q : int 0 1 0 0 0 0 0 0 0 1 ...  
## $ mfr\_R : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ type\_C : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ type\_H : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ shelf\_1 : int 0 0 0 0 1 0 0 1 0 0 ...  
## $ shelf\_2 : int 0 0 0 0 0 1 0 0 0 1 ...  
## $ shelf\_3 : int 1 1 1 1 0 0 1 0 1 0 ...

# Assigning cereal lables as row names of the data frame.  
rownames(cereals\_data) <- data$name  
head(cereals\_data)

## calories protein fat sodium fiber carbo sugars potass  
## 100%\_Bran 70 4 1 130 10.0 5.0 6 280  
## 100%\_Natural\_Bran 120 3 5 15 2.0 8.0 8 135  
## All-Bran 70 4 1 260 9.0 7.0 5 320  
## All-Bran\_with\_Extra\_Fiber 50 4 0 140 14.0 8.0 0 330  
## 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  
## vitamins weight cups rating mfr\_A mfr\_G mfr\_K mfr\_N  
## 100%\_Bran 25 1 0.33 68.40297 0 0 0 1  
## 100%\_Natural\_Bran 0 1 1.00 33.98368 0 0 0 0  
## All-Bran 25 1 0.33 59.42551 0 0 1 0  
## All-Bran\_with\_Extra\_Fiber 25 1 0.50 93.70491 0 0 1 0  
## Apple\_Cinnamon\_Cheerios 25 1 0.75 29.50954 0 1 0 0  
## Apple\_Jacks 25 1 1.00 33.17409 0 0 1 0  
## mfr\_P mfr\_Q mfr\_R type\_C type\_H shelf\_1 shelf\_2  
## 100%\_Bran 0 0 0 1 0 0 0  
## 100%\_Natural\_Bran 0 1 0 1 0 0 0  
## All-Bran 0 0 0 1 0 0 0  
## All-Bran\_with\_Extra\_Fiber 0 0 0 1 0 0 0  
## Apple\_Cinnamon\_Cheerios 0 0 0 1 0 1 0  
## Apple\_Jacks 0 0 0 1 0 0 1  
## shelf\_3  
## 100%\_Bran 1  
## 100%\_Natural\_Bran 1  
## All-Bran 1  
## All-Bran\_with\_Extra\_Fiber 1  
## Apple\_Cinnamon\_Cheerios 0  
## Apple\_Jacks 0

## Data Normalization

## Data Scaling   
mean\_norm\_minmax <- function(x){  
 (x- mean(x)) /(max(x)-min(x))  
}  
  
cereals\_data <- as.data.frame(lapply(cereals\_data, mean\_norm\_minmax))  
rownames(cereals\_data) <- data$name  
#cereals\_data\_norm <- scale(cereals\_data\_dum, center = T, scale = T)  
head(cereals\_data)

## calories protein fat sodium fiber  
## 100%\_Bran -0.33660934 0.2972973 0.0 -0.1011402 0.55888031  
## 100%\_Natural\_Bran 0.11793612 0.0972973 0.8 -0.4605152 -0.01254826  
## All-Bran -0.33660934 0.2972973 0.0 0.3051098 0.48745174  
## All-Bran\_with\_Extra\_Fiber -0.51842752 0.2972973 -0.2 -0.0698902 0.84459459  
## Apple\_Cinnamon\_Cheerios 0.02702703 -0.1027027 0.2 0.0551098 -0.04826255  
## Apple\_Jacks 0.02702703 -0.1027027 -0.2 -0.1167652 -0.08397683  
## carbo sugars potass vitamins  
## 100%\_Bran -0.5405405 -0.07387387 0.57614758 -0.04054054  
## 100%\_Natural\_Bran -0.3738739 0.05945946 0.11583012 -0.29054054  
## All-Bran -0.4294294 -0.14054054 0.70313170 -0.04054054  
## All-Bran\_with\_Extra\_Fiber -0.3738739 -0.47387387 0.73487773 -0.04054054  
## Apple\_Cinnamon\_Cheerios -0.2349850 0.19279279 -0.09051909 -0.04054054  
## Apple\_Jacks -0.2072072 0.45945946 -0.21750322 -0.04054054  
## weight cups rating mfr\_A  
## 100%\_Bran -0.03081081 -0.3932973 0.3440454 -0.01351351  
## 100%\_Natural\_Bran -0.03081081 0.1427027 -0.1108628 -0.01351351  
## All-Bran -0.03081081 -0.3932973 0.2253933 -0.01351351  
## All-Bran\_with\_Extra\_Fiber -0.03081081 -0.2572973 0.6784526 -0.01351351  
## Apple\_Cinnamon\_Cheerios -0.03081081 -0.0572973 -0.1699960 -0.01351351  
## Apple\_Jacks -0.03081081 0.1427027 -0.1215628 -0.01351351  
## mfr\_G mfr\_K mfr\_N mfr\_P  
## 100%\_Bran -0.2972973 -0.3108108 0.93243243 -0.1216216  
## 100%\_Natural\_Bran -0.2972973 -0.3108108 -0.06756757 -0.1216216  
## All-Bran -0.2972973 0.6891892 -0.06756757 -0.1216216  
## All-Bran\_with\_Extra\_Fiber -0.2972973 0.6891892 -0.06756757 -0.1216216  
## Apple\_Cinnamon\_Cheerios 0.7027027 -0.3108108 -0.06756757 -0.1216216  
## Apple\_Jacks -0.2972973 0.6891892 -0.06756757 -0.1216216  
## mfr\_Q mfr\_R type\_C type\_H  
## 100%\_Bran -0.09459459 -0.09459459 0.01351351 -0.01351351  
## 100%\_Natural\_Bran 0.90540541 -0.09459459 0.01351351 -0.01351351  
## All-Bran -0.09459459 -0.09459459 0.01351351 -0.01351351  
## All-Bran\_with\_Extra\_Fiber -0.09459459 -0.09459459 0.01351351 -0.01351351  
## Apple\_Cinnamon\_Cheerios -0.09459459 -0.09459459 0.01351351 -0.01351351  
## Apple\_Jacks -0.09459459 -0.09459459 0.01351351 -0.01351351  
## shelf\_1 shelf\_2 shelf\_3  
## 100%\_Bran -0.2567568 -0.2702703 0.527027  
## 100%\_Natural\_Bran -0.2567568 -0.2702703 0.527027  
## All-Bran -0.2567568 -0.2702703 0.527027  
## All-Bran\_with\_Extra\_Fiber -0.2567568 -0.2702703 0.527027  
## Apple\_Cinnamon\_Cheerios 0.7432432 -0.2702703 -0.472973  
## Apple\_Jacks -0.2567568 0.7297297 -0.472973

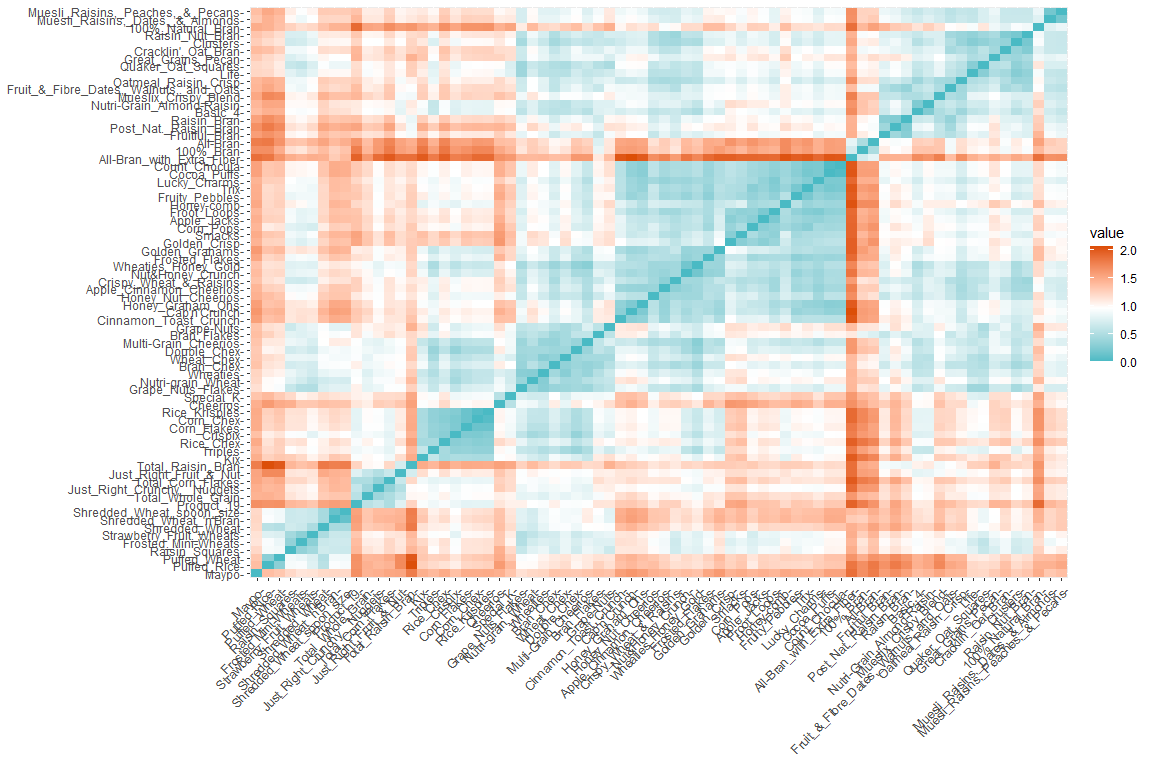
## DATA EXPLORATION

# Correlation chart avoiding the dummified variables  
library(factoextra)

## Loading required package: ggplot2

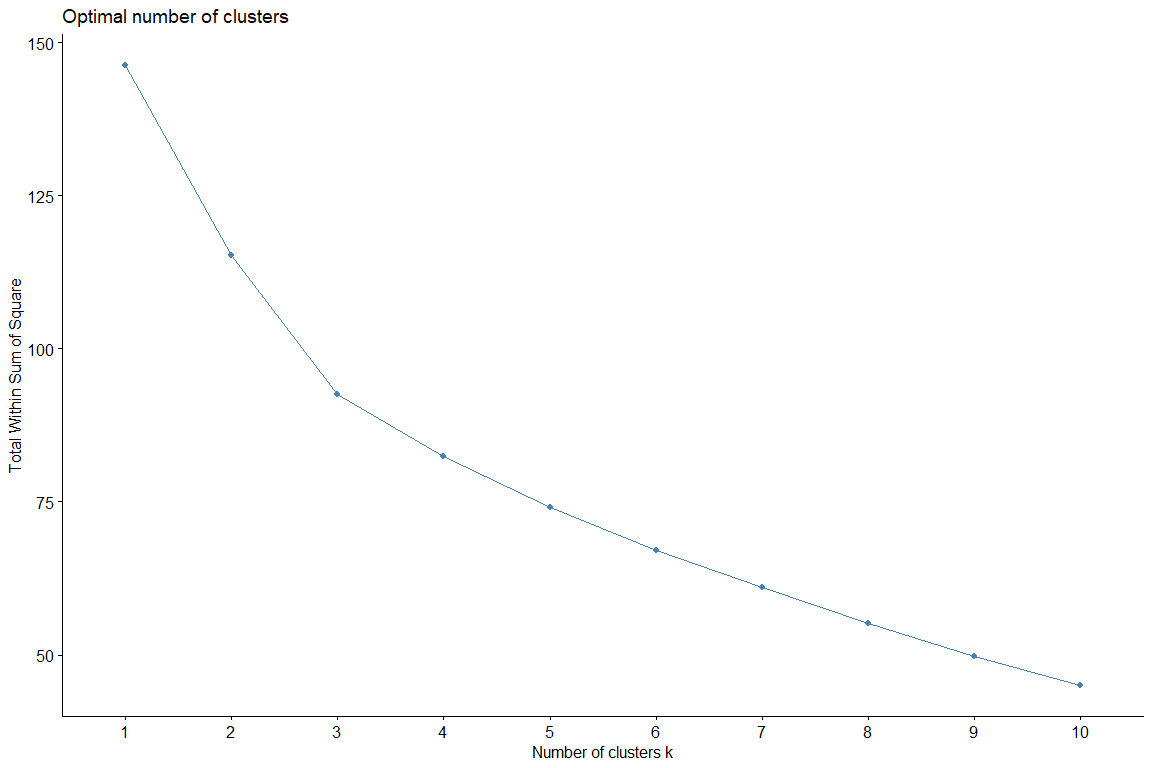
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

distance <- get\_dist(cereals\_data[,2:13])  
fviz\_dist(distance, gradient = list(low= "#00AFBB", mid = "white", high = "#DC4E07"))



### Determining Optimal Clusters

fviz\_nbclust(cereals\_data, FUN = hcut, method = "wss")



* From these estimators, lets assume the optimum K would be 3 We shall evaluate its stability later.

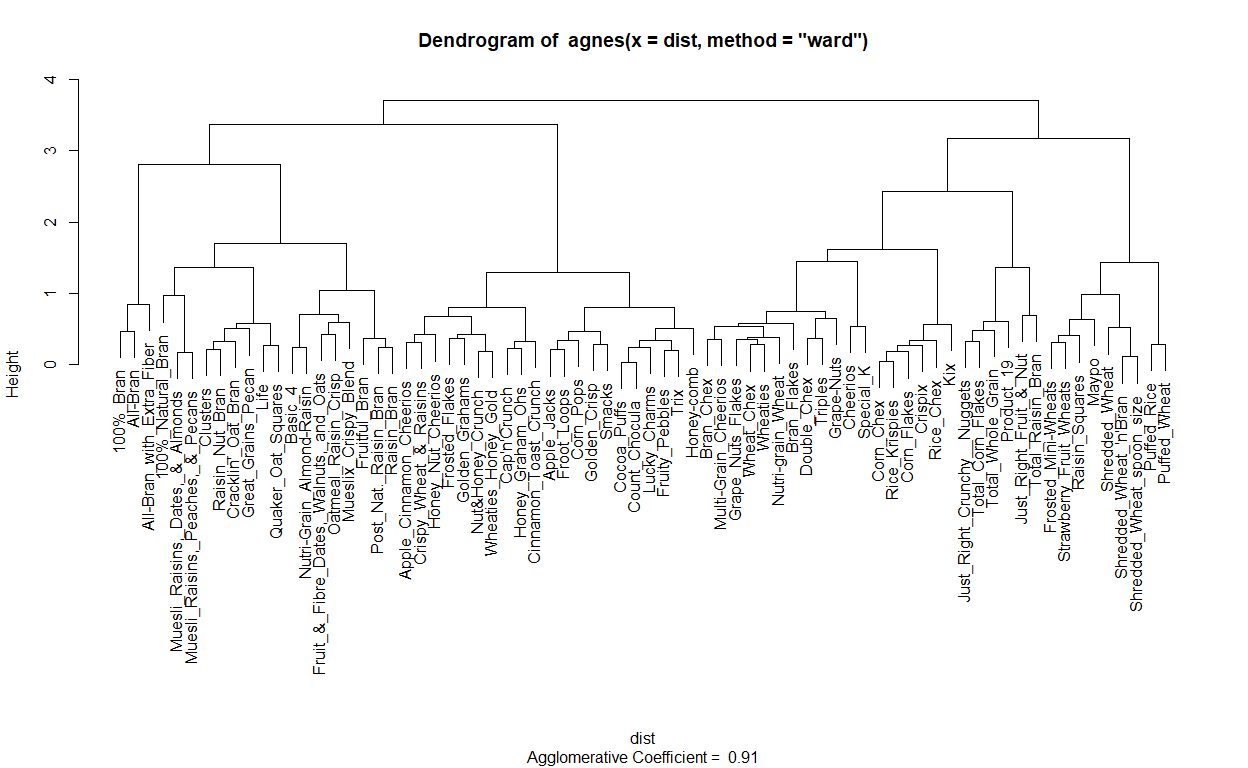
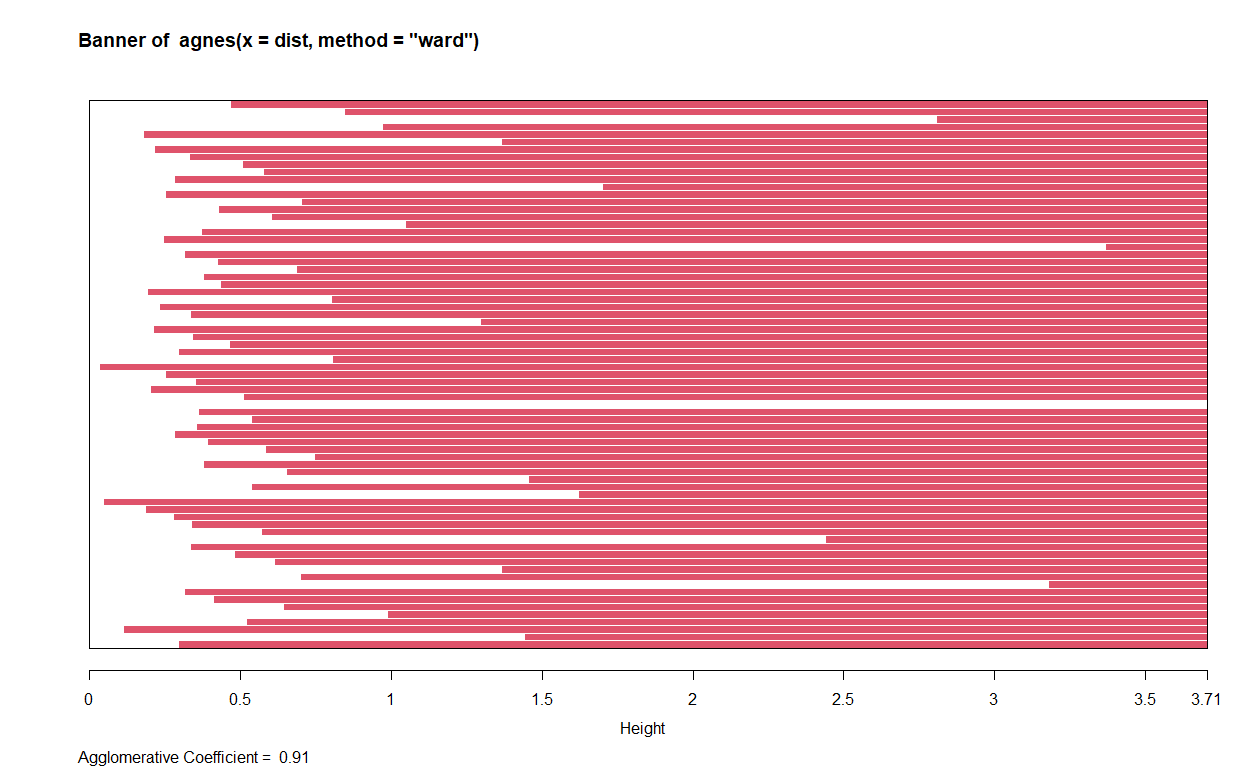
## Hierarchical Clustering

### I will use the euclidean distance measure distance.

dist <- dist(cereals\_data[,1:12], method="euclidean")

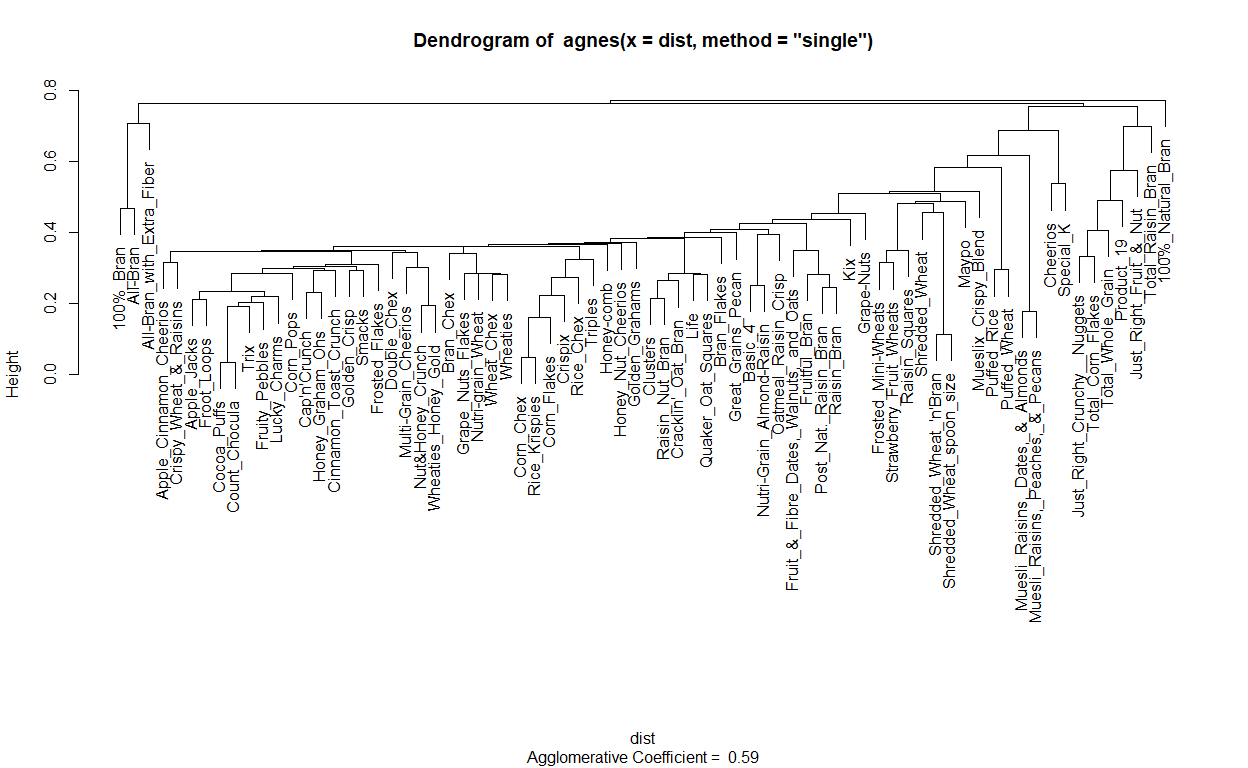
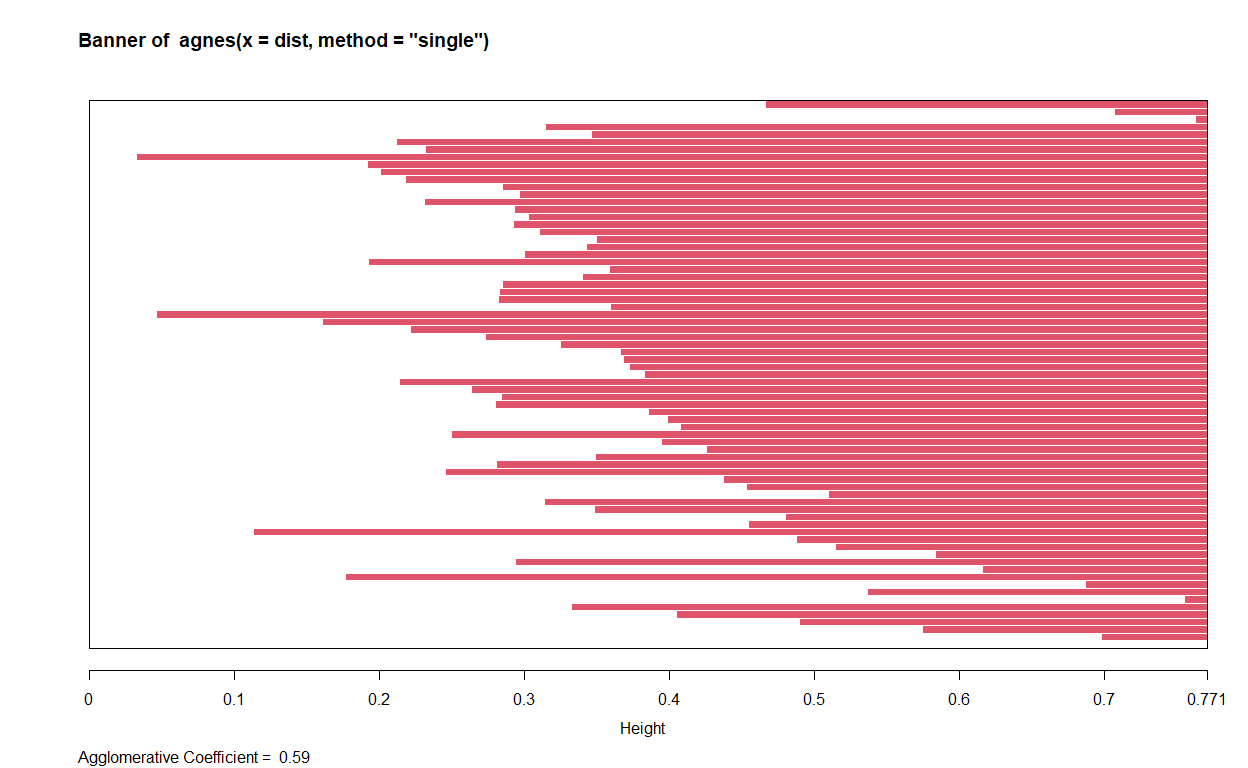
* hierarchical clustering using ward linkage method.

library(cluster)  
hc\_fit\_wd <- agnes(dist, method="ward")  
plot(hc\_fit\_wd)



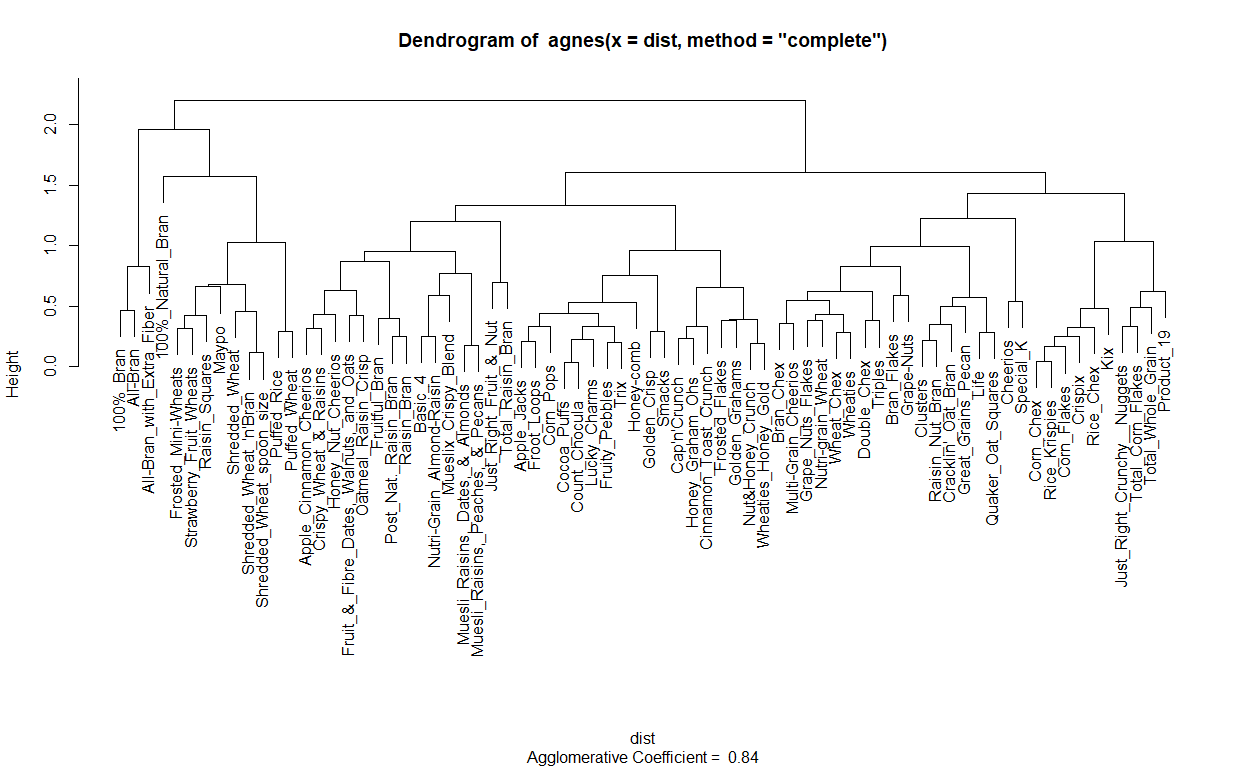
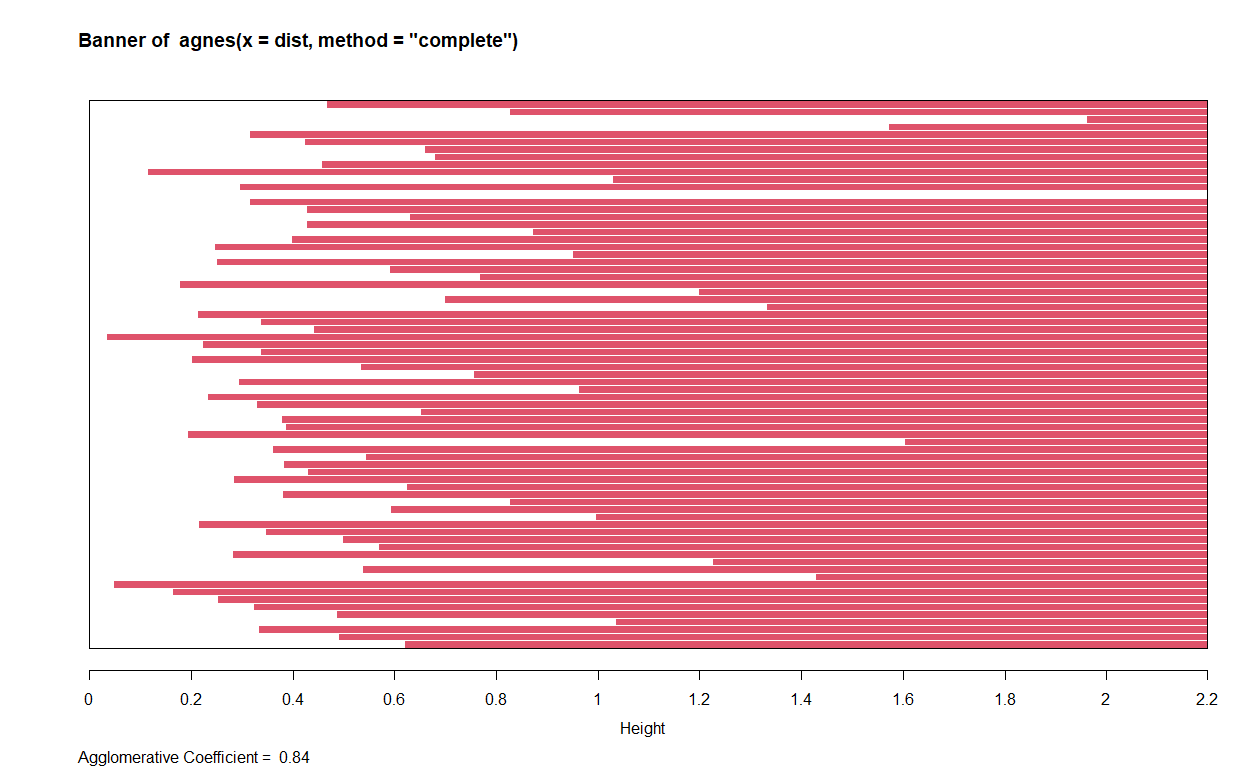
* hierarchical clustering using single linkage method.

hc\_fit\_sg <- agnes(dist, method="single")  
plot(hc\_fit\_sg)



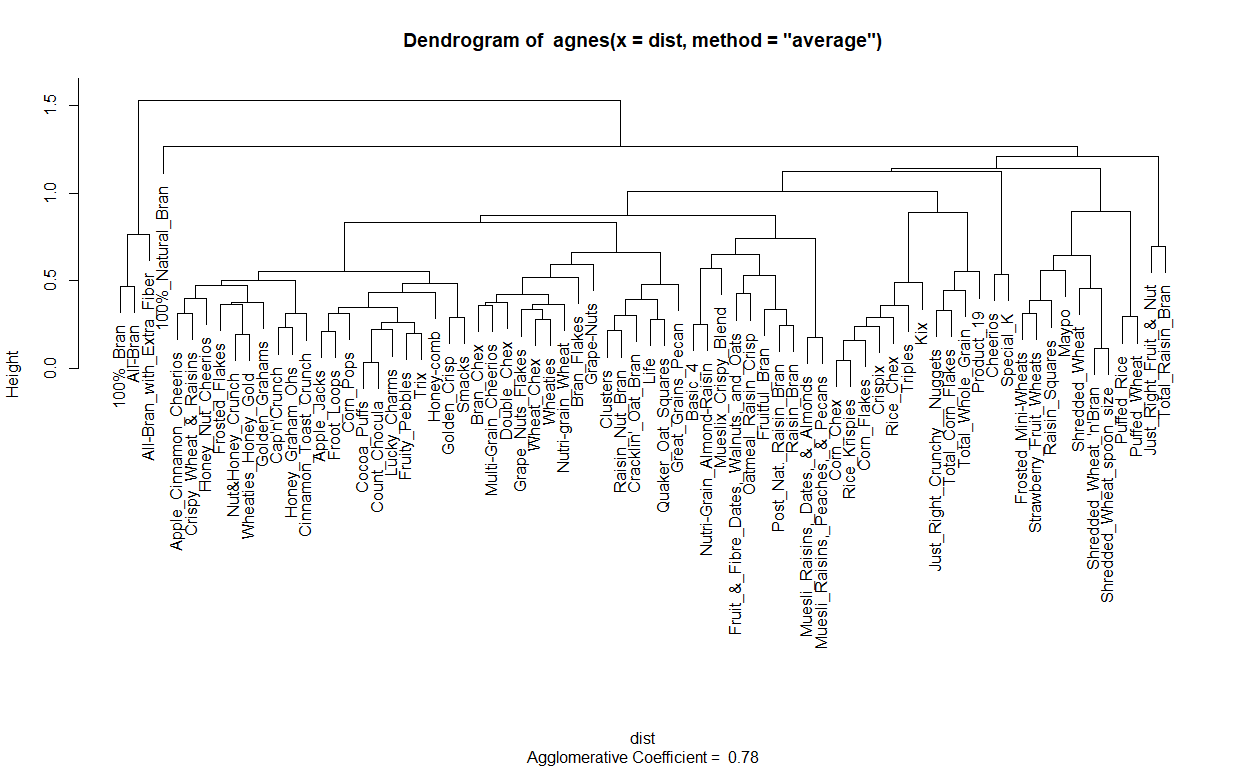
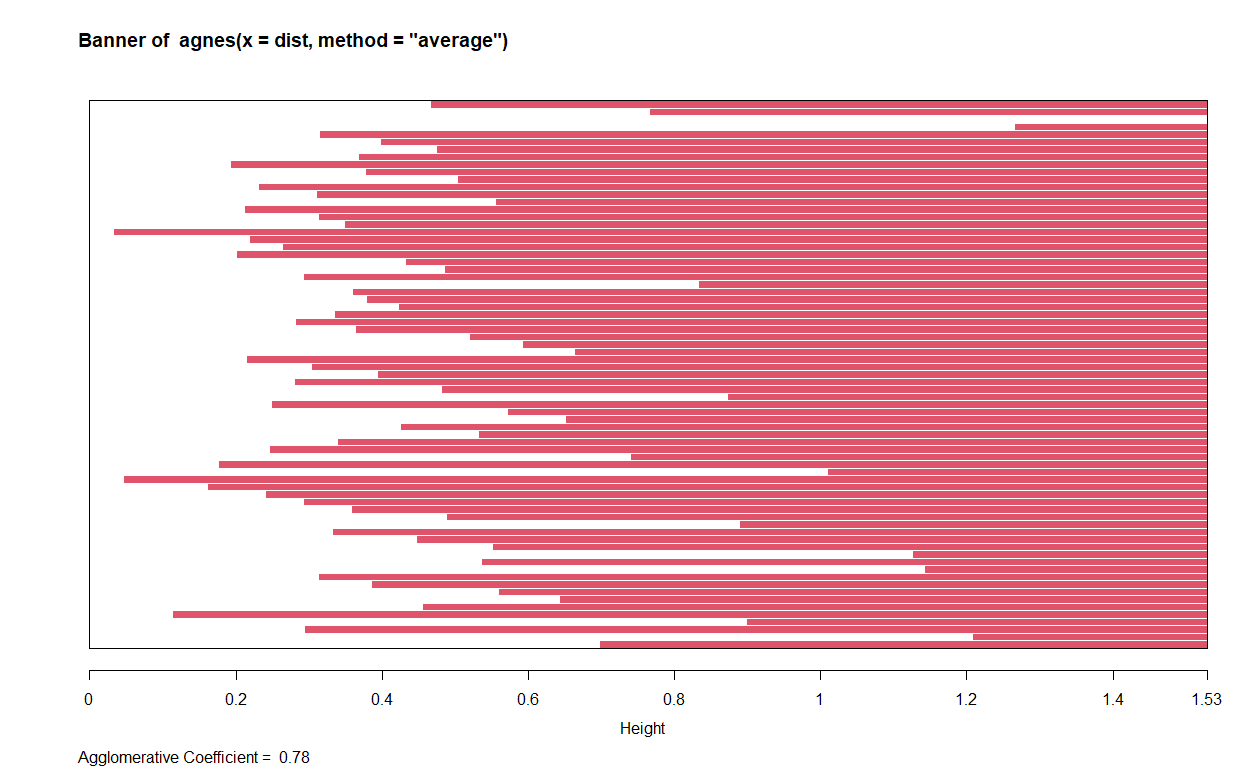
* hierarchical clustering using complete linkage method.

hc\_fit\_cmp <- agnes(dist, method="complete")  
plot(hc\_fit\_cmp)



* hierarchical clustering using ward linkage method.

hc\_fit\_avg <- agnes(dist, method="average")  
plot(hc\_fit\_avg)

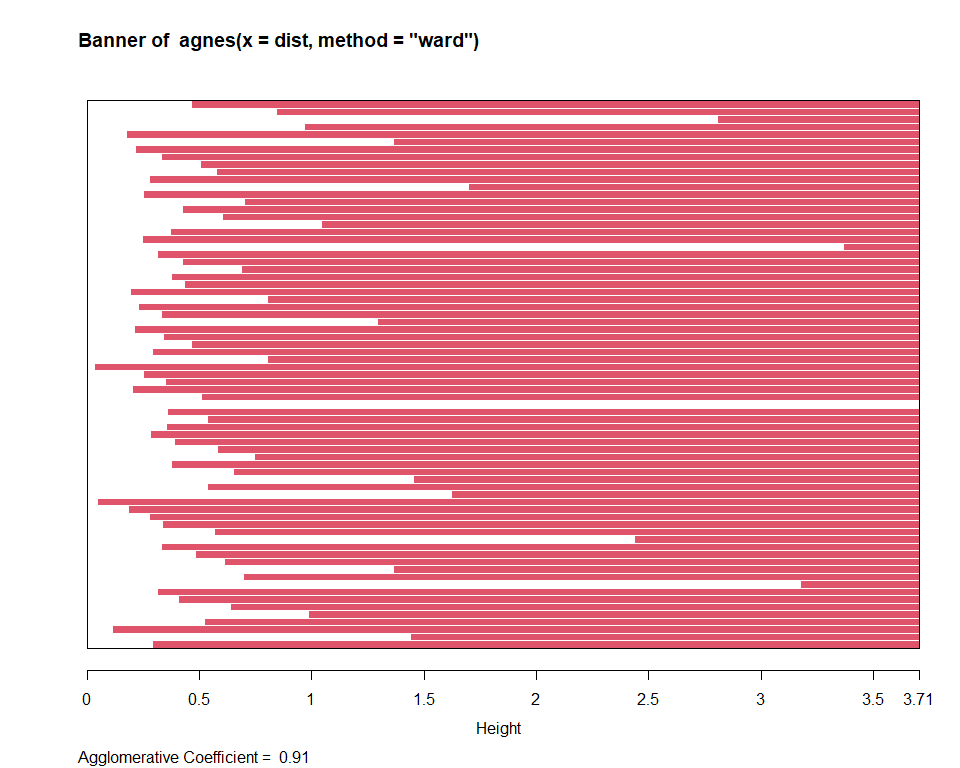


#### Based on the agglomerative coefficients, “WARD” is the most efficient method to proceed further.

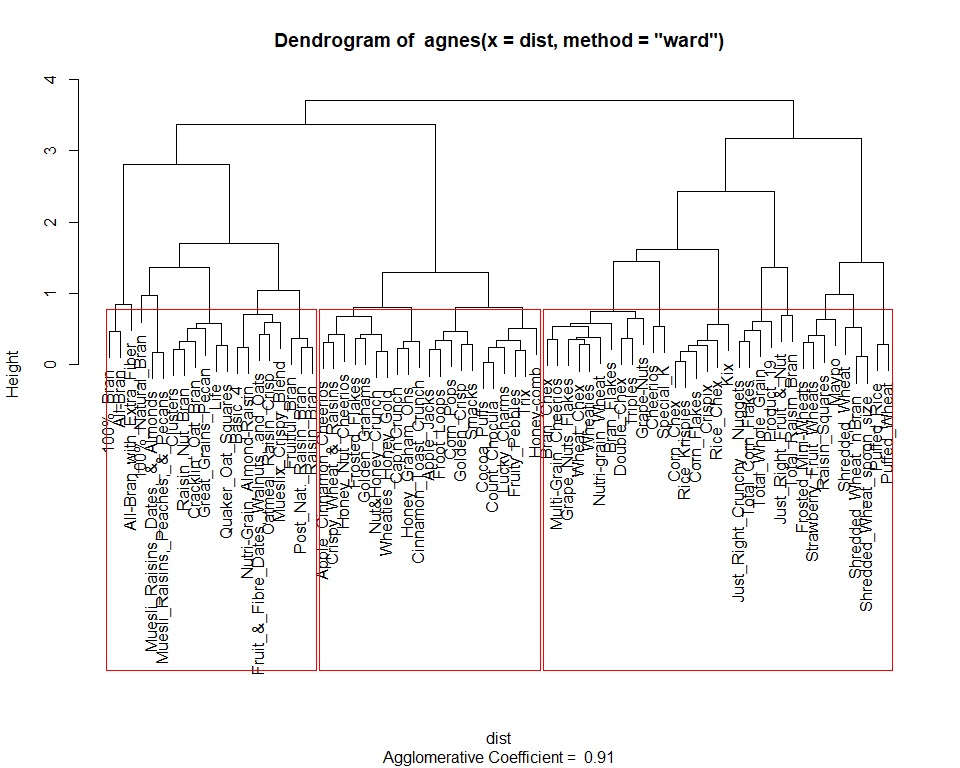
points\_hc <- cutree(hc\_fit\_wd, k=3)  
cereals\_clusts\_hc <- cbind(points\_hc, cereals\_data)  
  
colnames(cereals\_clusts\_hc)[1] <- "cluster\_hc"  
head(cereals\_clusts\_hc)

## cluster\_hc calories protein fat sodium  
## 100%\_Bran 1 -0.33660934 0.2972973 0.0 -0.1011402  
## 100%\_Natural\_Bran 1 0.11793612 0.0972973 0.8 -0.4605152  
## All-Bran 1 -0.33660934 0.2972973 0.0 0.3051098  
## All-Bran\_with\_Extra\_Fiber 1 -0.51842752 0.2972973 -0.2 -0.0698902  
## Apple\_Cinnamon\_Cheerios 2 0.02702703 -0.1027027 0.2 0.0551098  
## Apple\_Jacks 2 0.02702703 -0.1027027 -0.2 -0.1167652  
## fiber carbo sugars potass  
## 100%\_Bran 0.55888031 -0.5405405 -0.07387387 0.57614758  
## 100%\_Natural\_Bran -0.01254826 -0.3738739 0.05945946 0.11583012  
## All-Bran 0.48745174 -0.4294294 -0.14054054 0.70313170  
## All-Bran\_with\_Extra\_Fiber 0.84459459 -0.3738739 -0.47387387 0.73487773  
## Apple\_Cinnamon\_Cheerios -0.04826255 -0.2349850 0.19279279 -0.09051909  
## Apple\_Jacks -0.08397683 -0.2072072 0.45945946 -0.21750322  
## vitamins weight cups rating  
## 100%\_Bran -0.04054054 -0.03081081 -0.3932973 0.3440454  
## 100%\_Natural\_Bran -0.29054054 -0.03081081 0.1427027 -0.1108628  
## All-Bran -0.04054054 -0.03081081 -0.3932973 0.2253933  
## All-Bran\_with\_Extra\_Fiber -0.04054054 -0.03081081 -0.2572973 0.6784526  
## Apple\_Cinnamon\_Cheerios -0.04054054 -0.03081081 -0.0572973 -0.1699960  
## Apple\_Jacks -0.04054054 -0.03081081 0.1427027 -0.1215628  
## mfr\_A mfr\_G mfr\_K mfr\_N  
## 100%\_Bran -0.01351351 -0.2972973 -0.3108108 0.93243243  
## 100%\_Natural\_Bran -0.01351351 -0.2972973 -0.3108108 -0.06756757  
## All-Bran -0.01351351 -0.2972973 0.6891892 -0.06756757  
## All-Bran\_with\_Extra\_Fiber -0.01351351 -0.2972973 0.6891892 -0.06756757  
## Apple\_Cinnamon\_Cheerios -0.01351351 0.7027027 -0.3108108 -0.06756757  
## Apple\_Jacks -0.01351351 -0.2972973 0.6891892 -0.06756757  
## mfr\_P mfr\_Q mfr\_R type\_C  
## 100%\_Bran -0.1216216 -0.09459459 -0.09459459 0.01351351  
## 100%\_Natural\_Bran -0.1216216 0.90540541 -0.09459459 0.01351351  
## All-Bran -0.1216216 -0.09459459 -0.09459459 0.01351351  
## All-Bran\_with\_Extra\_Fiber -0.1216216 -0.09459459 -0.09459459 0.01351351  
## Apple\_Cinnamon\_Cheerios -0.1216216 -0.09459459 -0.09459459 0.01351351  
## Apple\_Jacks -0.1216216 -0.09459459 -0.09459459 0.01351351  
## type\_H shelf\_1 shelf\_2 shelf\_3  
## 100%\_Bran -0.01351351 -0.2567568 -0.2702703 0.527027  
## 100%\_Natural\_Bran -0.01351351 -0.2567568 -0.2702703 0.527027  
## All-Bran -0.01351351 -0.2567568 -0.2702703 0.527027  
## All-Bran\_with\_Extra\_Fiber -0.01351351 -0.2567568 -0.2702703 0.527027  
## Apple\_Cinnamon\_Cheerios -0.01351351 0.7432432 -0.2702703 -0.472973  
## Apple\_Jacks -0.01351351 -0.2567568 0.7297297 -0.472973

library(cluster)  
plot(hc\_fit\_wd)



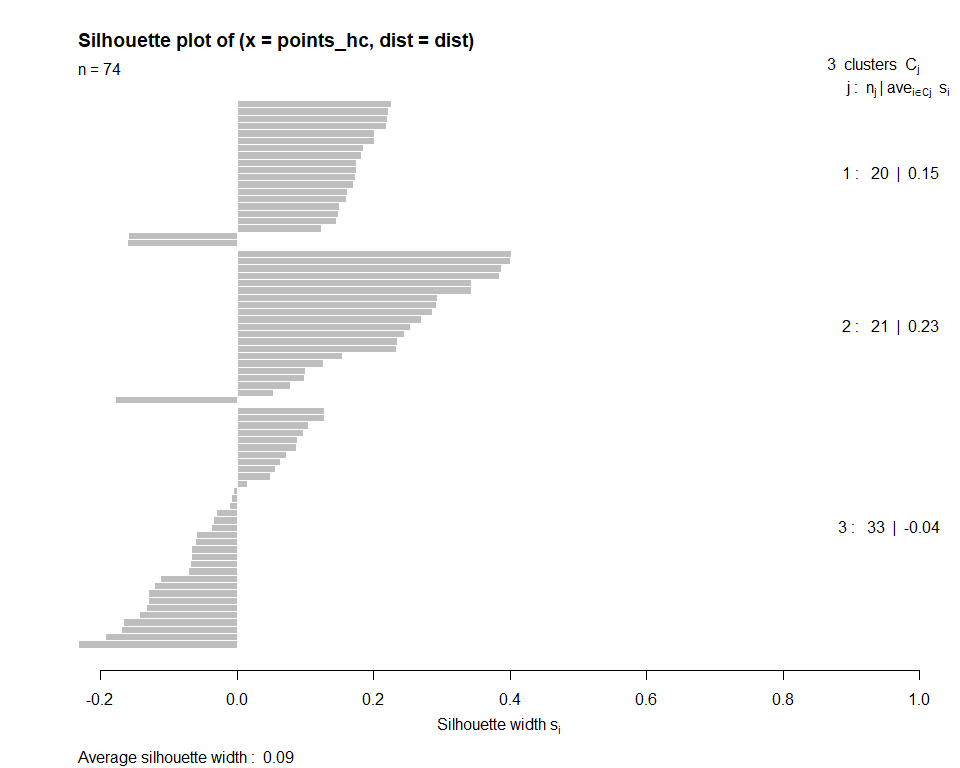
rect.hclust(hc\_fit\_wd, k = 3, border = "red")



### Checking Quality of clusters Created

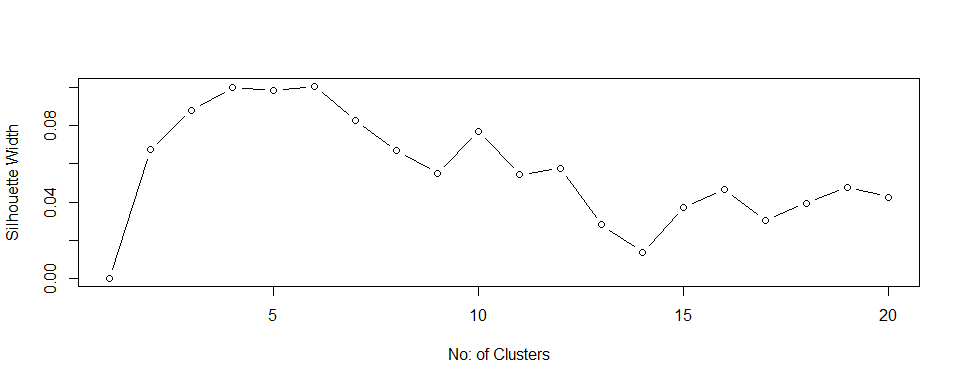
- The silhouette width/value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation) [i.e., intra-cluster cohesion and inter-cluster separation]  
- Ranges from -1 to +1   
- Values closer to 1 means higher quality of the cluster created

library(cluster)  
dist = daisy(x = cereals\_data, metric = "euclidean")  
sil\_value = silhouette(points\_hc, dist = dist)  
plot(sil\_value)



* finding the optimal number of clusters where silhouette width would be maximum

sil\_value\_hc = 0  
for (i in 2:20) {  
 points\_hc <- cutree(hc\_fit\_wd, k = i)  
 sil\_value\_hc[i] = mean(silhouette(points\_hc, dist = dist)[,3])  
}  
plot(1:20, sil\_value\_hc, type = "b", xlab = "No: of Clusters", ylab = "Silhouette Width")



* According to the Silhoutte, the optimized cluster value is 4 and 6. Lets check the stability of both 4 and 6 clusters with clusterboot method now.

### Cluster Stability

#### Checking for Stability of k=4.

* clusterboot is an integrated function that computes the clustering as well, using interface functions for various clustering methods implemented in R (several interface functions are provided, but you can implement further ones for your favourite clustering method)
* Clusterboot function using library(fpc)

library(fpc)

## Warning: package 'fpc' was built under R version 4.0.5

#Input the scaled cereals\_data  
hclust\_stability = clusterboot(cereals\_data, clustermethod=hclustCBI, method="ward.D2", k=4, count = FALSE)

* What are the cluster stability values? Values > 0.85 denote very stable clusters. 0.6 - 0.75 means the clusters show some patterns but needs to be investigated further

#Cluster stability values  
hclust\_stability$bootmean

## [1] 0.6541825 0.6004057 0.6193093 0.6900000

* How many times the different clusters were dissolved

#Cluster dissolution rate. If maximum Jaccard coefficient < 0.5, that cluster is assumed to be dissolved. Below code shows the number of times each cluster was dissolved. The lower the value, the better.  
hclust\_stability$bootbrd

## [1] 35 41 36 31

#### Checking for Stability of k=6.

hclust\_stability = clusterboot(cereals\_data, clustermethod=hclustCBI, method="ward.D2", k=6, count = FALSE)

* What are the cluster stability values? Values > 0.85 denote very stable clusters. 0.6 - 0.75 means the clusters show some patterns but needs to be investigated further

#Cluster stability values  
hclust\_stability$bootmean

## [1] 0.6783651 0.4780906 0.6388820 0.7087018 0.6867587 0.5800000

* How many times the different clusters were dissolved

#Cluster dissolution rate. If maximum Jaccard coefficient < 0.5, that cluster is assumed to be dissolved. Below code shows the number of times each cluster was dissolved. The lower the value, the better.  
hclust\_stability$bootbrd

## [1] 31 71 24 11 19 42

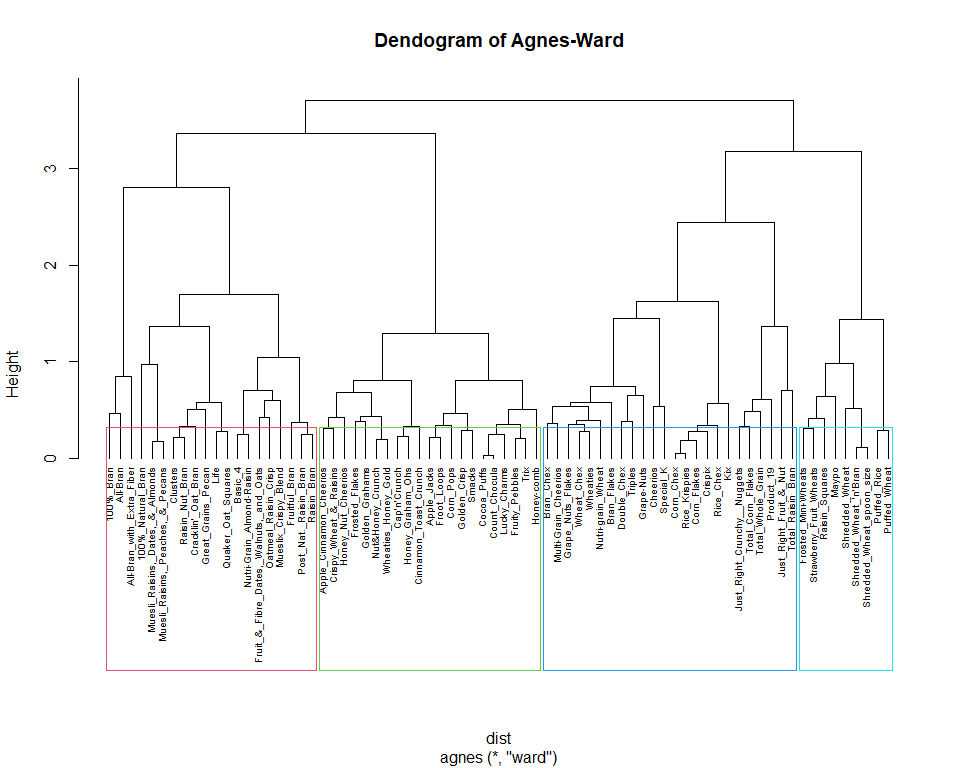
#### Hence, after checking the stability of both posible values of K; the best choice is 4.

* Implementing the the hierarchical clustering with k=4

points\_hc\_4 <- cutree(hc\_fit\_wd, k=4)  
cereals\_clusts\_hc\_4 <- cbind(points\_hc\_4, cereals\_data)  
  
colnames(cereals\_clusts\_hc\_4)[1] <- "cluster\_hc"  
head(cereals\_clusts\_hc\_4)

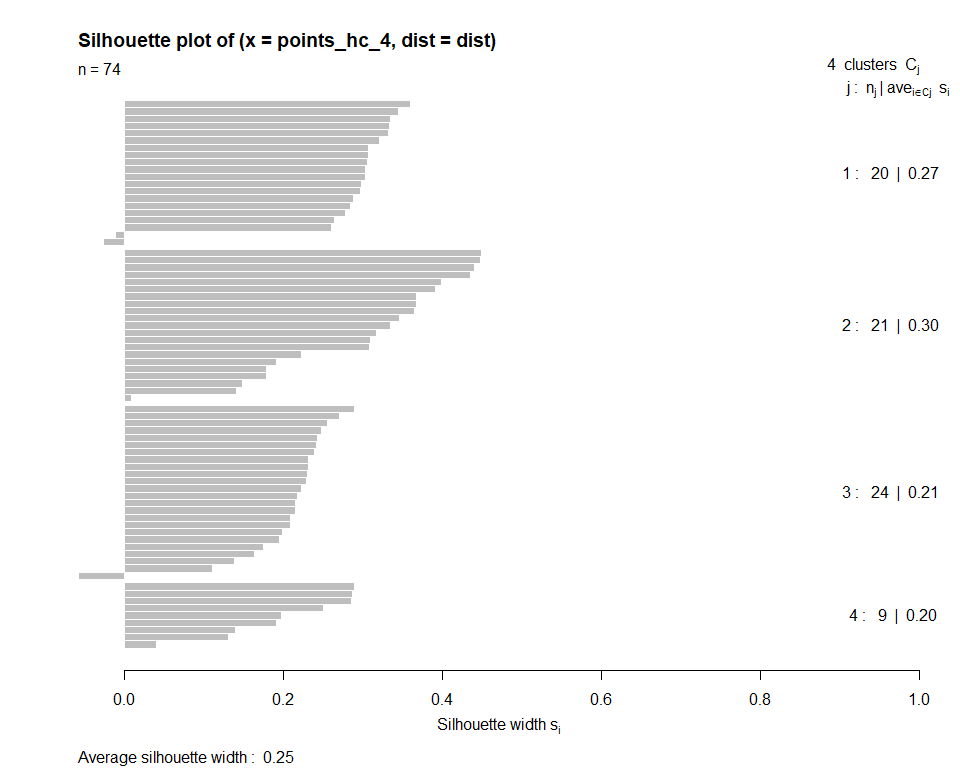
## cluster\_hc calories protein fat sodium  
## 100%\_Bran 1 -0.33660934 0.2972973 0.0 -0.1011402  
## 100%\_Natural\_Bran 1 0.11793612 0.0972973 0.8 -0.4605152  
## All-Bran 1 -0.33660934 0.2972973 0.0 0.3051098  
## All-Bran\_with\_Extra\_Fiber 1 -0.51842752 0.2972973 -0.2 -0.0698902  
## Apple\_Cinnamon\_Cheerios 2 0.02702703 -0.1027027 0.2 0.0551098  
## Apple\_Jacks 2 0.02702703 -0.1027027 -0.2 -0.1167652  
## fiber carbo sugars potass  
## 100%\_Bran 0.55888031 -0.5405405 -0.07387387 0.57614758  
## 100%\_Natural\_Bran -0.01254826 -0.3738739 0.05945946 0.11583012  
## All-Bran 0.48745174 -0.4294294 -0.14054054 0.70313170  
## All-Bran\_with\_Extra\_Fiber 0.84459459 -0.3738739 -0.47387387 0.73487773  
## Apple\_Cinnamon\_Cheerios -0.04826255 -0.2349850 0.19279279 -0.09051909  
## Apple\_Jacks -0.08397683 -0.2072072 0.45945946 -0.21750322  
## vitamins weight cups rating  
## 100%\_Bran -0.04054054 -0.03081081 -0.3932973 0.3440454  
## 100%\_Natural\_Bran -0.29054054 -0.03081081 0.1427027 -0.1108628  
## All-Bran -0.04054054 -0.03081081 -0.3932973 0.2253933  
## All-Bran\_with\_Extra\_Fiber -0.04054054 -0.03081081 -0.2572973 0.6784526  
## Apple\_Cinnamon\_Cheerios -0.04054054 -0.03081081 -0.0572973 -0.1699960  
## Apple\_Jacks -0.04054054 -0.03081081 0.1427027 -0.1215628  
## mfr\_A mfr\_G mfr\_K mfr\_N  
## 100%\_Bran -0.01351351 -0.2972973 -0.3108108 0.93243243  
## 100%\_Natural\_Bran -0.01351351 -0.2972973 -0.3108108 -0.06756757  
## All-Bran -0.01351351 -0.2972973 0.6891892 -0.06756757  
## All-Bran\_with\_Extra\_Fiber -0.01351351 -0.2972973 0.6891892 -0.06756757  
## Apple\_Cinnamon\_Cheerios -0.01351351 0.7027027 -0.3108108 -0.06756757  
## Apple\_Jacks -0.01351351 -0.2972973 0.6891892 -0.06756757  
## mfr\_P mfr\_Q mfr\_R type\_C  
## 100%\_Bran -0.1216216 -0.09459459 -0.09459459 0.01351351  
## 100%\_Natural\_Bran -0.1216216 0.90540541 -0.09459459 0.01351351  
## All-Bran -0.1216216 -0.09459459 -0.09459459 0.01351351  
## All-Bran\_with\_Extra\_Fiber -0.1216216 -0.09459459 -0.09459459 0.01351351  
## Apple\_Cinnamon\_Cheerios -0.1216216 -0.09459459 -0.09459459 0.01351351  
## Apple\_Jacks -0.1216216 -0.09459459 -0.09459459 0.01351351  
## type\_H shelf\_1 shelf\_2 shelf\_3  
## 100%\_Bran -0.01351351 -0.2567568 -0.2702703 0.527027  
## 100%\_Natural\_Bran -0.01351351 -0.2567568 -0.2702703 0.527027  
## All-Bran -0.01351351 -0.2567568 -0.2702703 0.527027  
## All-Bran\_with\_Extra\_Fiber -0.01351351 -0.2567568 -0.2702703 0.527027  
## Apple\_Cinnamon\_Cheerios -0.01351351 0.7432432 -0.2702703 -0.472973  
## Apple\_Jacks -0.01351351 -0.2567568 0.7297297 -0.472973

pltree(hc\_fit\_wd, cex = 0.6, hang = -1, main = "Dendogram of Agnes-Ward")  
rect.hclust(hc\_fit\_wd, k = 4, border = 2:5)



#### checking Quality of clusters Created

library(cluster)  
dist = daisy(x = cereals\_clusts\_hc\_4, metric = "euclidean")  
sil\_value = silhouette(points\_hc\_4, dist = dist)  
plot(sil\_value)



* A significant improvement in the silhouette width from the case when k was 3.

##### Selection of the cluster that would be the best cereal for breakfast are based on:

- Sodium and Sugar content should be minimal

* Cluster 3 has the best options of healthy cereals that students can be served all 5 workdays - so that everyday they can be served with different cereals.

library(hrbrthemes)

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

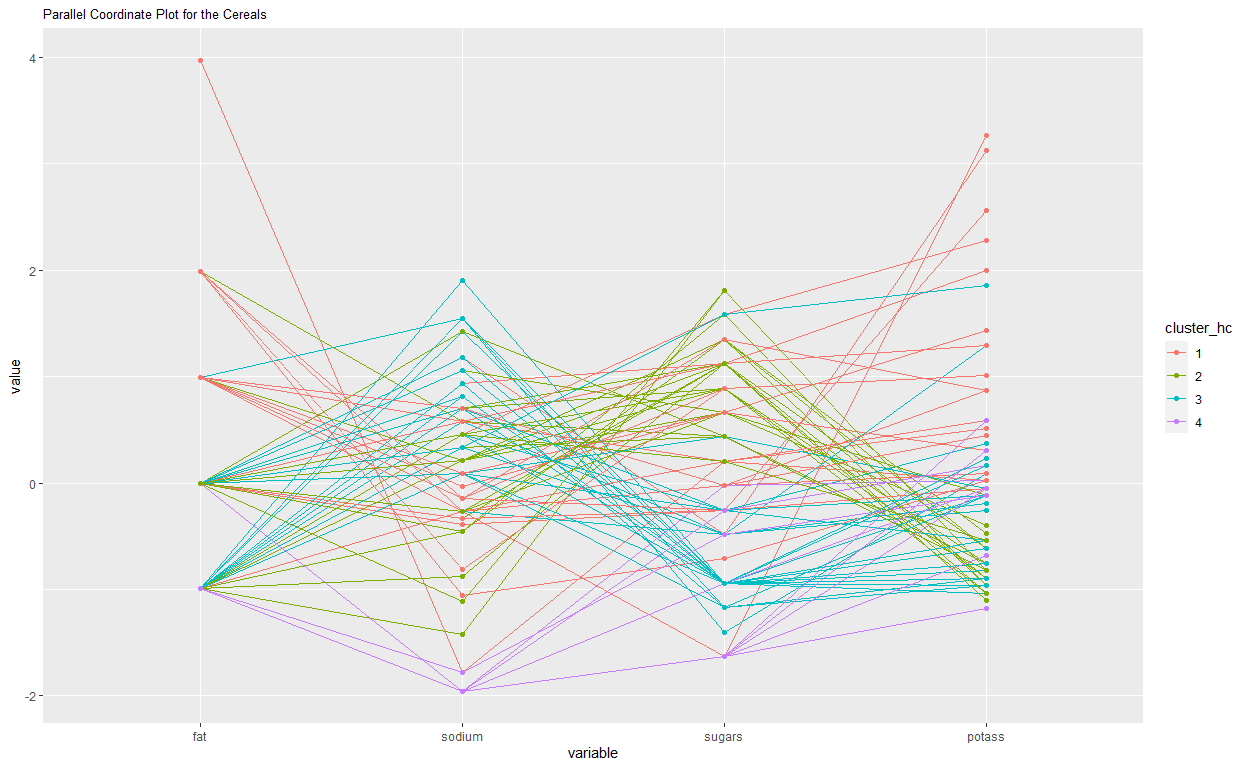
## Please use hrbrthemes::import\_roboto\_condensed() to install Roboto Condensed and

## if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

cereals\_clusts\_hc\_4$cluster\_hc <- as.factor(cereals\_clusts\_hc\_4$cluster\_hc)  
  
ggparcoord(cereals\_clusts\_hc\_4,  
 columns = c(4,5,8,9), groupColumn = 1,  
 showPoints = TRUE,   
 title = "Parallel Coordinate Plot for the Cereals",  
 alphaLines = 1  
 ) + theme(plot.title = element\_text(size=10))



* Approaching forward with an elimination technique, we will cancel a few parameters to be considered while choosing the best cluster for healthy breakfast.
  + We will consider fat, sodium, sugars and potash for choosing; the cereal having the least nutrition value in these criteria should be eliminated from selection.
  + Cluster 4 seems to be the best choice.