FML ASSIGNMENT-5

LOKESH JETANGI

2023-04-16

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#install.packages("cluster")
#install.packages("caret")
#install.packages("dendextend")
#install.packages("factoextra")
#install.packages("knitr")

library(cluster)

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

library(caret)

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

## Loading required package: ggplot2

## Loading required package: lattice

library(dendextend)

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

##
## ---------------------
## 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/talgal
ili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
```

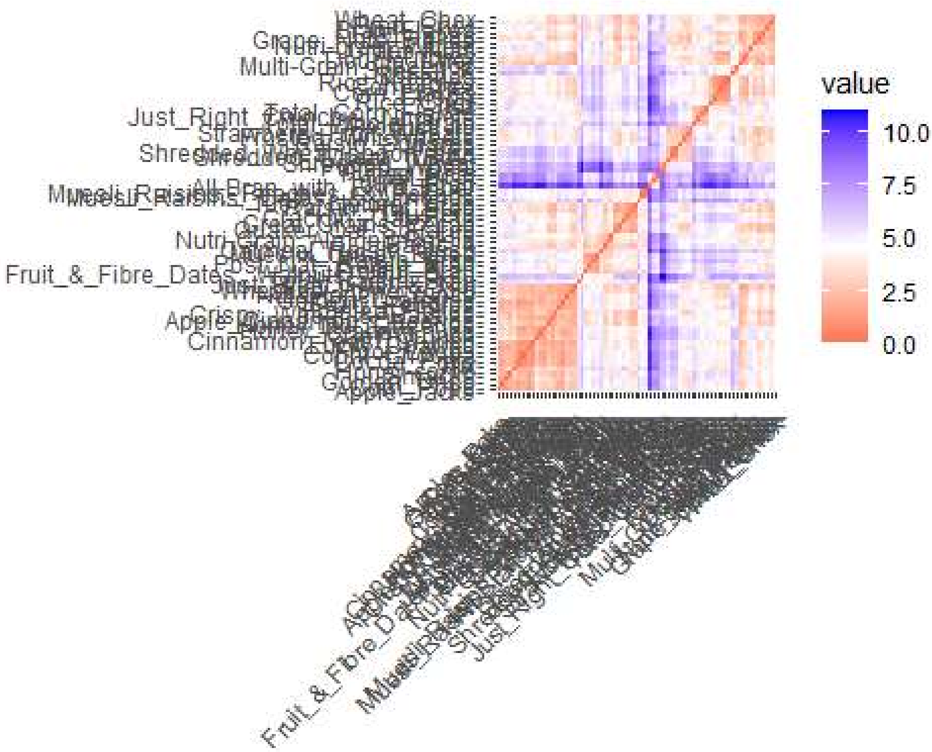
```
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(den
dextend))
## ---------------------

##
## Attaching package: 'dendextend'

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

library(knitr)
library(factoextra)

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

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

cereals_data <- read.csv("C:/Users/jetan/Downloads/Cereals.csv")
str(cereals_data)

## 'data.frame':    77 obs. of  16 variables:
##  $ name    : chr  "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-Bran_wit
h_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 ...

#Removing missing values
sum(is.na(cereals_data))

## [1] 4

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 

cereals1<- na.omit(cereals_data) #missing values removed
colMeans(is.na(cereals1))

##     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 

cerealsnames <- cereals1[,c(1,2)]
cerealsnames

##                                      name mfr
## 1                               100%_Bran   N
## 2                       100%_Natural_Bran   Q
## 3                                All-Bran   K
## 4               All-Bran_with_Extra_Fiber   K
## 6                 Apple_Cinnamon_Cheerios   G
## 7                             Apple_Jacks   K
## 8                                 Basic_4   G
## 9                               Bran_Chex   R
## 10                            Bran_Flakes   P
## 11                           Cap'n'Crunch   Q
## 12                               Cheerios   G
## 13                  Cinnamon_Toast_Crunch   G
## 14                               Clusters   G
## 15                            Cocoa_Puffs   G
## 16                              Corn_Chex   R
## 17                            Corn_Flakes   K
## 18                              Corn_Pops   K
## 19                          Count_Chocula   G
## 20                     Cracklin'_Oat_Bran   K
## 22                                Crispix   K
## 23                 Crispy_Wheat_&_Raisins   G
## 24                            Double_Chex   R
## 25                            Froot_Loops   K
## 26                         Frosted_Flakes   K
## 27                    Frosted_Mini-Wheats   K
## 28 Fruit_&_Fibre_Dates,_Walnuts,_and_Oats   P
## 29                          Fruitful_Bran   K
## 30                         Fruity_Pebbles   P
## 31                           Golden_Crisp   P
## 32                         Golden_Grahams   G
## 33                      Grape_Nuts_Flakes   P
## 34                             Grape-Nuts   P
## 35                     Great_Grains_Pecan   P
## 36                       Honey_Graham_Ohs   Q
## 37                      Honey_Nut_Cheerios   G
## 38                             Honey-comb   P
```

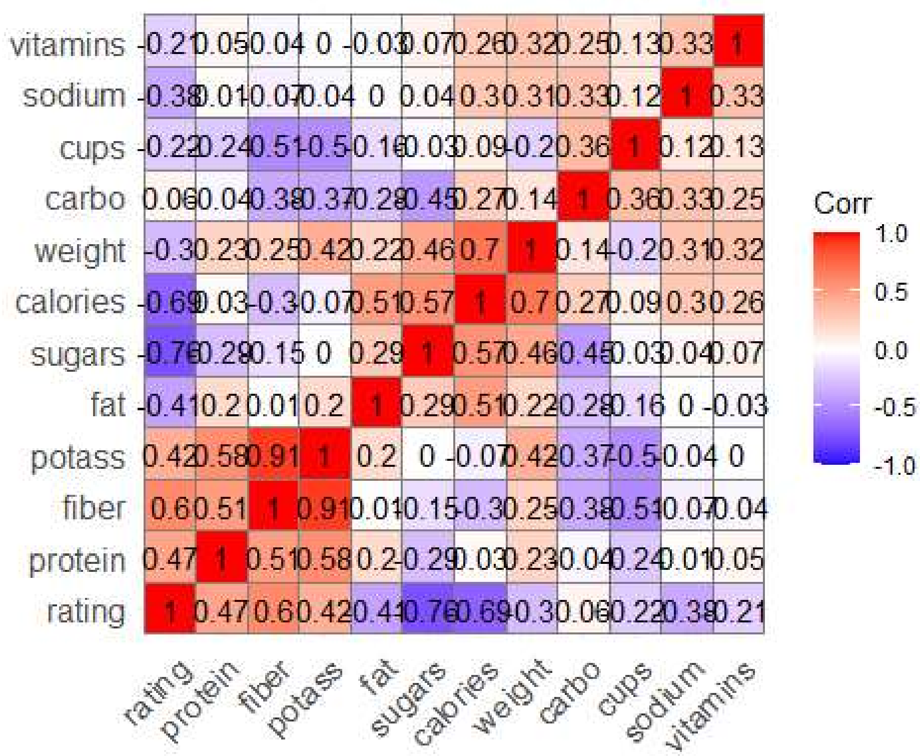
```
## 39            Just_Right_Crunchy__Nuggets   K
## 40                 Just_Right_Fruit_&_Nut   K
## 41                                    Kix   G
## 42                                   Life   Q
## 43                           Lucky_Charms   G
## 44                                  Maypo   A
## 45      Muesli_Raisins,_Dates,_&_Almonds   R
## 46     Muesli_Raisins,_Peaches,_&_Pecans   R
## 47                   Mueslix_Crispy_Blend   K
## 48                   Multi-Grain_Cheerios   G
## 49                       Nut&Honey_Crunch   K
## 50              Nutri-Grain_Almond-Raisin   K
## 51                      Nutri-grain_Wheat   K
## 52                   Oatmeal_Raisin_Crisp   G
## 53                  Post_Nat._Raisin_Bran   P
## 54                             Product_19   K
## 55                            Puffed_Rice   Q
## 56                           Puffed_Wheat   Q
## 57                     Quaker_Oat_Squares   Q
## 59                            Raisin_Bran   K
## 60                        Raisin_Nut_Bran   G
## 61                         Raisin_Squares   K
## 62                              Rice_Chex   R
## 63                          Rice_Krispies   K
## 64                         Shredded_Wheat   N
## 65                 Shredded_Wheat_'n'Bran   N
## 66              Shredded_Wheat_spoon_size   N
## 67                                 Smacks   K
## 68                              Special_K   K
## 69                Strawberry_Fruit_Wheats   N
## 70                      Total_Corn_Flakes   G
## 71                      Total_Raisin_Bran   G
## 72                      Total_Whole_Grain   G
## 73                                Triples   G
## 74                                   Trix   G
## 75                             Wheat_Chex   R
## 76                               Wheaties   G
## 77                   Wheaties_Honey_Gold   G
```

```
#It shows us the column name of Cereal’s dataset changed from column to row n
ame
# Extract full cereal names
cerealsnames<- cereals1[,1]

# Remove duplicate rows
cereals_unique <- cereals1[!duplicated(cereals1),]

# Change column names to row names
row.names(cereals_unique) <- cereals_unique[, 1]
cereals_unique <- cereals_unique[, c(4:12, 14:16)]
```

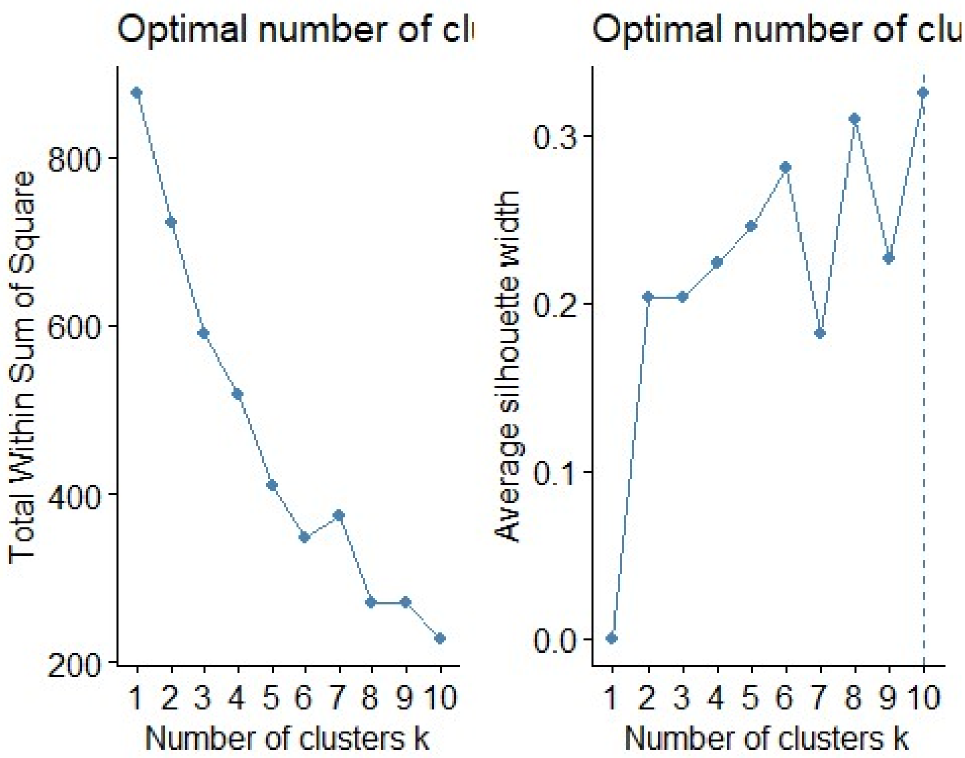
```r
# Normalize the dataset
cereals_normalized <- scale(cereals_unique)

# View only first 6 rows
head(cereals_normalized)

##                            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

distancetable <- get_dist(cereals_normalized )
fviz_dist(distancetable)
```




```
# Correlation between Variables.
library(ggcorrplot)
Correlation <- cor(cereals_normalized )
ggcorrplot(Correlation, outline.color = "grey50", lab = TRUE, hc.order = TRUE
, type = "full")
```



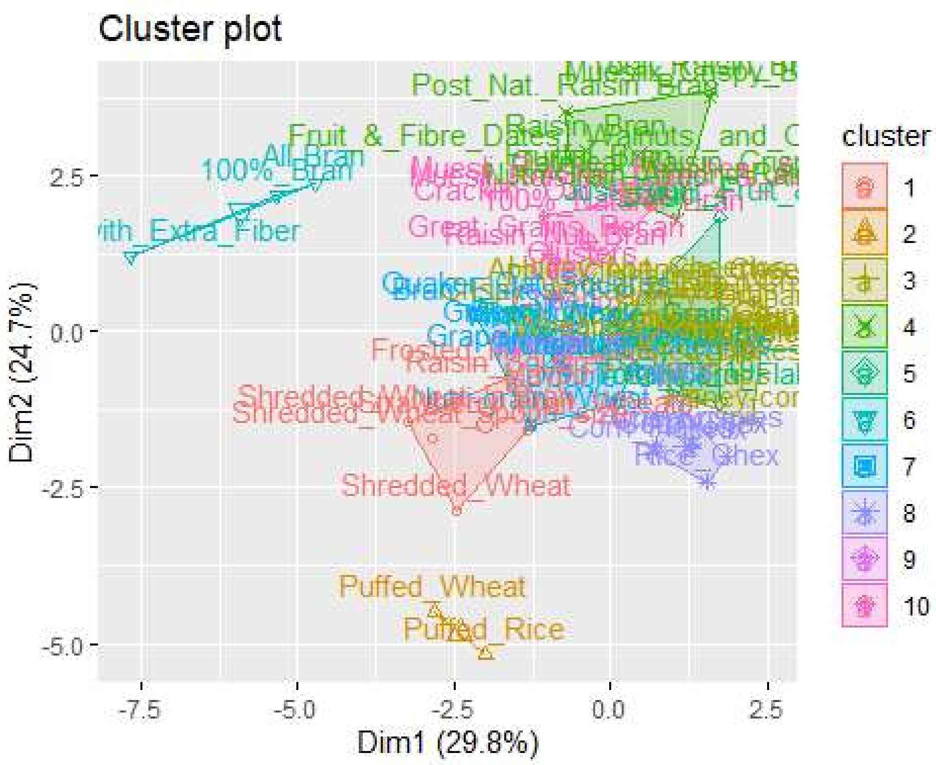

```r
# K value using Kmeans firs and Using both the values elbow and silhouette to see K value.
library(cowplot)
# K value using Kmeans first and Using both the values elbow and silhouette to see K value.
library(cowplot)
Elbowmethod <- fviz_nbclust(cereals_normalized, kmeans, method = "wss")
Silhouettemethod <- fviz_nbclust(cereals_normalized, kmeans, method = "silhouette")
plot_grid(Elbowmethod, Silhouettemethod, nrow = 1)
#from both the methods k=10
```



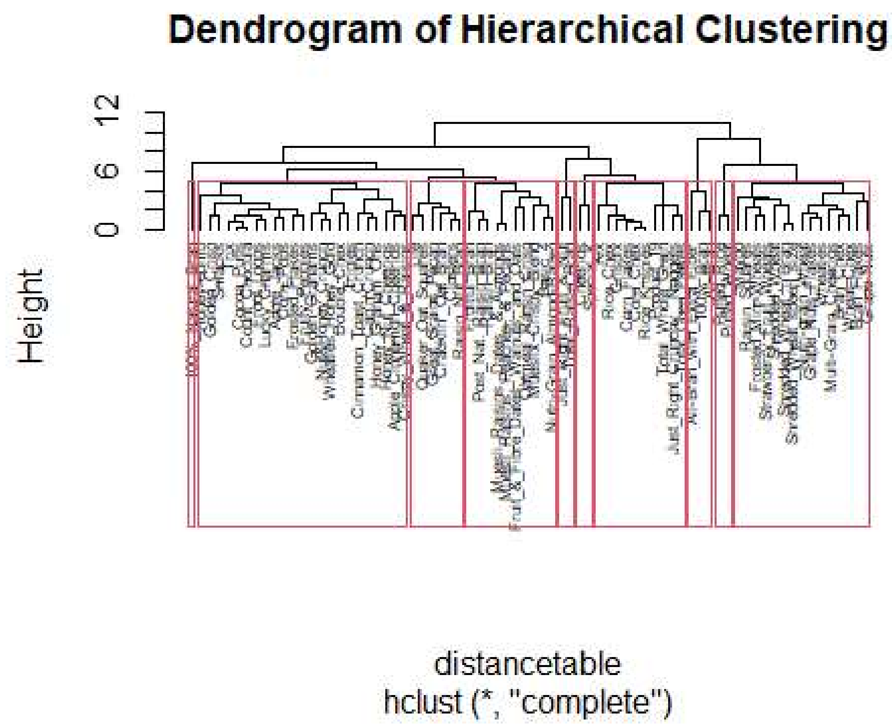

```r
set.seed(1234)
k10 <- kmeans(cereals_normalized, centers = 10, nstart = 25)
k5 <- kmeans(cereals_normalized, centers = 5, nstart = 25)
k10$centers
```

```
##       calories   protein           fat      sodium      fiber       carbo
## 1  -0.78605825 0.1866257 -8.513317e-01 -1.93575474  0.1633054  0.43653231
## 2  -2.87378226 -0.9421007 -9.932203e-01 -1.96164410 -0.6914590 -0.82990744
## 3   0.19781169 -0.9199689  1.057355e-17  0.07498586 -0.6619844 -0.59130285
## 4   1.21367739  0.4522084  4.414312e-01  0.38757596  0.6840240  0.08372381
## 5   0.25060471  0.0803926 -1.986441e-01  0.59967697 -0.3200786  1.04589172
## 6  -2.20187108  1.3817478 -3.310734e-01  0.17279012  3.6413124 -2.07187492
## 7  -0.45490203  0.2663005 -3.972881e-01  0.29159354  0.2988887  0.30071123
## 8   0.07782755 -0.6101224 -7.094430e-01  1.19685830 -0.7798829  1.75803465
## 9   0.14981803  3.2408266  0.000000e+00  1.17959872 -0.2788141  0.45488650
## 10  0.65375141  0.8007856  1.862288e+00 -0.59490143  0.2112017 -0.62112842
##        sugars     potass   vitamins     weight        cups      rating
## 1  -0.9424187  0.1217481 -0.6624252 -0.3591327 -0.06748537  1.49437400
## 2  -1.6306324 -0.9313592 -1.3032024 -3.4599552  0.75675340  1.39015899
## 3   1.0020580 -0.7214096 -0.1818422 -0.2008324  0.22746282 -0.97698099
## 4   0.9183071  1.1653377  0.1919445  2.0805535 -0.49239931 -0.43724782
## 5  -0.5294905 -0.4163948  3.1822385  0.1902623  0.54463313 -0.16904450
## 6  -0.7894824  2.9837813 -0.1818422 -0.2008324 -1.84525525  2.24264794
## 7  -0.6212523  0.1408955 -0.1818422 -0.2008324 -0.35051441  0.56389406
## 8  -1.0079629 -0.8759325 -0.1818422 -0.2008324  0.98705540 -0.01518936
## 9  -1.1718233 -0.2612000 -0.1818422 -0.2008324  1.28705407  0.68238355
## 10  0.1472529  0.5059559 -0.3220122 -0.2008324 -0.56899829 -0.19615104
```

```
k10$size

##  [1]  7  2 21  9  5  3 10  7  2  8

fviz_cluster(k10, data = cereals_normalized)
```

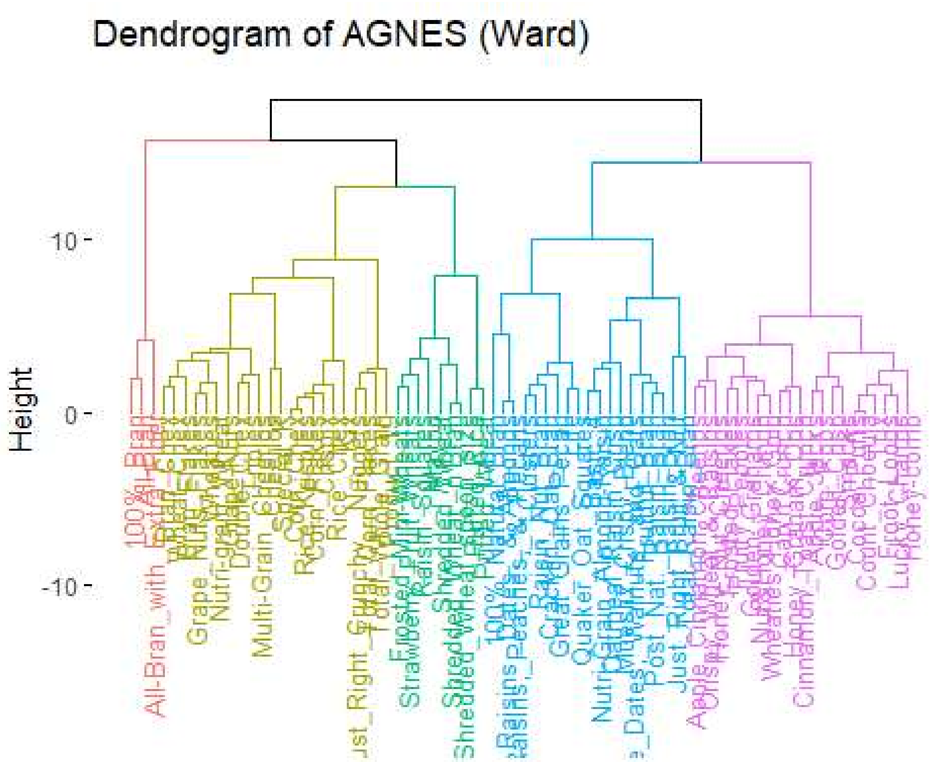



```
#After performing both the silhouette and elbow methods, a K value of 10 was
obtained for clustering. However, after visualizing the resulting 10 clusters
,we observed overlapping of some clusters, suggesting that K-means clustering
alone may not be the best approach for optimization. As a result, we will app
ly hierarchical clustering to determine the optimal number of clusters. This
technique creates a dendrogram that shows the relationships between data poin
ts and can help us identify the appropriate number of clusters based on our d
esired outcome.

set.seed(1234)
hc<- hclust(distancetable,
          method = "complete" )# Hc using Complete Linkage
plot(hc,
     cex = 0.5,
     hang = -1,
     main = "Dendrogram of Hierarchical Clustering")
     rect.hclust(hc, k = 10)
```




```
set.seed(1234)
# Perform clustering using AGNES with single linkage method
hc_single <- agnes(cereals_normalized, method = "single")
hc_complete <- agnes(cereals_normalized, method = "complete")
hc_average <- agnes(cereals_normalized, method = "average")
hc_ward <- agnes(cereals_normalized, method = "ward")
##Compare the agglomerative coefficients for single,complete,average and ward
.
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

#The agglomerative coefficient (ac) is a measure of the quality of the
clustering,with higher values indicating better clustering. Based on the prin
ted coefficients, we can see that the ward linkage method has the highest
coefficient i.e 0.9046042, followed by the average linkage method, the comple
```

```
te linkage method,and finally the single linkage method.
Therefore, we can conclude that the ward linkage method is the best method fo
r clustering the cereal data based on Euclidean distance to the normalized
measurements.

# clusters 5 appear to be a good number to group the data using the Ward link
age.
set.seed(123)
fviz_dend(hc_ward, k = 5,main = "Dendrogram of AGNES (Ward)",cex = 0.7, color
_labels_by_k = TRUE,labels_track_height = 18)

## 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 <]8;;https://github.com/kassambara/factoextra
/issueshttps://github.com/kassambara/factoextra/issues]8;;>.
```



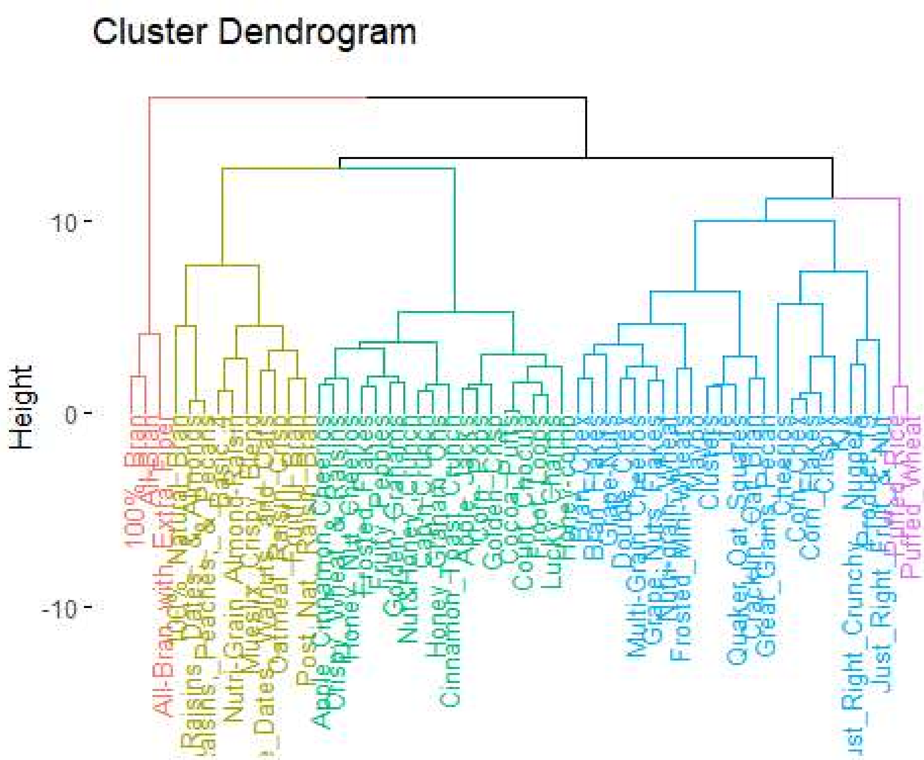

```
cereals2 <- cutree(hc_ward, k = 5)
Clustered_df <-as.data.frame(cbind ( cereals_normalized, cereals2 ))
```

#Q3:Comment on the structure of the clusters and their stability.

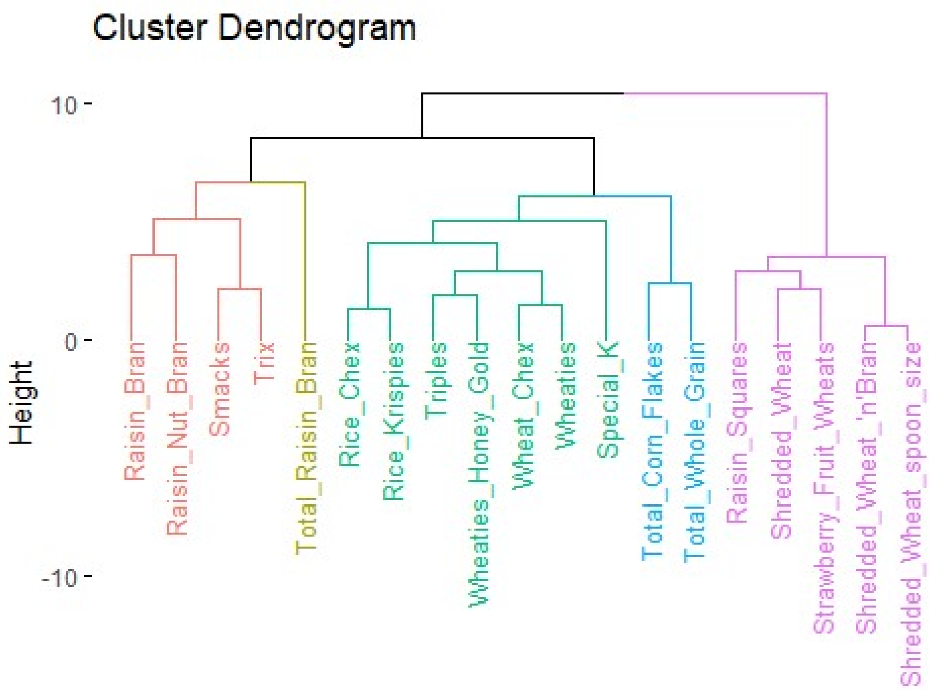
```
# partitioning the dataset into two groups  Training A and Validation B.
set.seed(2023)# To get the same random variables
Train_A <-cereals_normalized [1:55,]
nrow(Train_A)
```

```
## [1] 55

Valid_B <-cereals_normalized [56:74,]
nrow(Valid_B)

## [1] 19

#Figure out the distances,Keep in mind that Euclidean distance is the standar
d.examining the trainingA and validationB data set cluster.
set.seed(1234)
distancetrain_A <- get_dist(Train_A)
# Compute with AGNES and with different linkage methods For Training Dataset
hc_single_Train_A <- agnes(distancetrain_A, method = "single")
hc_complete_Train_A <- agnes(distancetrain_A, method = "complete")
hc_average_Train_A <- agnes(distancetrain_A, method = "average")
hc_ward_Train_A <- agnes(distancetrain_A, method = "ward")
print(hc_single_Train_A$ac)

## [1] 0.6663587

print(hc_complete_Train_A$ac)

## [1] 0.8285192

print(hc_average_Train_A$ac)

## [1] 0.7646836

print(hc_ward_Train_A$ac)

## [1] 0.8891086

## Compute with AGNES and with different linkage methods For Training Dataset
set.seed(2023)# To maintain same values
distance_validB <- get_dist(Valid_B)
hc_single_Valid_B <- agnes(distance_validB, method = "single")
hc_complete_Valid_B <- agnes(distance_validB, method = "complete")
hc_average_Valid_B <- agnes(distance_validB, method = "average")
hc_ward_Valid_B<- agnes(distance_validB, method = "ward")
# Compare AGNES (agglomerative) coefficients
print(hc_single_Valid_B$ac)

## [1] 0.4805129

print(hc_complete_Valid_B$ac)

## [1] 0.71298

print(hc_average_Valid_B$ac)

## [1] 0.6232053

print(hc_ward_Valid_B$ac)
```

```
## [1] 0.7710122

#Dendrogram for TrainingA and ValidationB
fviz_dend(hc_ward_Train_A, k = 5,cex = 0.7, color_labels_by_k = TRUE,labels_t
rack_height = 16,)
```




```
fviz_dend(hc_ward_Valid_B, k = 5,
          cex = 0.7,  color_labels_by_k = TRUE,labels_track_height = 16,)
```



To assign each store in partition B, use centroids from A

```
Clustered_DF_A <-cutree (hc_ward_Train_A, k=5)
Clusters_A <-as.data.frame(cbind(Train_A, Clustered_DF_A))
nrow(Clusters_A)

## [1] 55

Clust_1 <- colMeans (Clusters_A [Clusters_A$ Clustered_DF_A == "1" ,])
Clustered_DF_B <-cutree (hc_ward_Valid_B, k=5)
Clusters_B <-as.data.frame(cbind(Valid_B, Clustered_DF_B))
nrow(Clusters_B)

## [1] 19

Clust_2 <- colMeans (Clusters_B [Clusters_B$ Clustered_DF_B == "2" ,])
Centroid <-rbind(Clust_1, Clust_2)
Centroid

##            calories    protein        fat     sodium     fiber      carbo
## Clust_1 -2.2018711  1.3817478 -0.3310734  0.1727901 3.6413124 -2.0718749
## Clust_2 -0.9588354 -0.1055153 -0.9932203 -1.9253990 0.3401532  0.5833659
##              sugars    potass   vitamins     weight       cups   rating
## Clust_1 -0.7894824 2.983781 -0.1818422 -0.2008324 -1.8452553 2.242648
## Clust_2 -1.1259424 0.176167 -0.8546583 -0.4224528 -0.2274847 1.686636
##          Clustered_DF_A
## Clust_1              1
## Clust_2              2
```

```
#On a general level, both clusters appear to be fine, but there is a small variation between those Cluster1 has more fiber and potassium than Cluster2, which may indicate that the cereals in this cluster are healthier or more nut rient-dense.Compared to Cluster_1, Cluster_2 has a higher sugar level, which may indicate that the cereals in this cluster are less healthful or include more added sugars.
```

#Q3{B} Method-2 Use the cluster centroids from A to assign each record in partition B

```
distance <- dist(Valid_B[,-1], Train_A, method = "euclidean")
hc <- hclust(distance)
cluster_B <- cutree(hc, k = 5)
Valid_B$cluster <- cluster_B

## Warning in Valid_B$cluster <- cluster_B: Coercing LHS to a list

Valid_B$cluster

##                Raisin_Bran            Raisin_Nut_Bran            Raisin_Squa
res
##                          1                          1
2
##                  Rice_Chex              Rice_Krispies            Shredded_Wh
eat
##                          3                          3
2
##      Shredded_Wheat_'n'Bran Shredded_Wheat_spoon_size                    Sma
cks
##                          2                          2
1
##                  Special_K   Strawberry_Fruit_Wheats         Total_Corn_Fla
kes
##                          4                          2
3
##          Total_Raisin_Bran         Total_Whole_Grain                   Trip
les
##                          5                          3
3
##                       Trix                 Wheat_Chex                 Wheat
ies
##                          1                          3
3
##       Wheaties_Honey_Gold
##                          3

#With the exception of three cereals,specialK,TotalCF, and TotalWG, the anticipated clusters of B based on the centroids of A are essentially identically classified. After comparing the validation data set with the training dataset,only 3 out of 19 observations had their clusters modified.It indicates that clusters have a very high level of stability.
```

#Q3{C} Assess how consistent the cluster assignments are compared to the assignments based on all the data

```
#The mean values of each feature for the two clusters found in the two datase
ts are being compared. The characteristics of the two clusters can be compare
d using these centroids to discover any differences or similarities.The fact
that Cluster1 has more fiber and potassium than Cluster2 may indicate that th
e cereals in this cluster are healthier or more nutrient-dense.Compared to Cl
uster1, Cluster2 has a larger sugar level, which could mean that the cereals
in this cluster are less healthier or include more added sugars. As a result,
Cluster2 scored far lower than Cluster1.

#method-2
#In this method cluster centroids from the TrainA dataset are produced using
hc with complete linkage, and pairwise Euclidean distances between the record
s in the ValidationB dataset are calculated.By using the centroids from the t
raining dataset, this enables the prediction of the cluster labels for the va
lidation dataset. Since the validation data set is based on the training data
set, we can observe its stability. With the exception of special_K,Total_CF,
and Total_WG, the cereals are all clustered exactly the same way.When the val
idation data set was compared to the training data set, only three observatio
ns out of 19 had their cluster modified.In other words, clusters have a very
high level of stability.
```

#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?

```
HealthyCereals <- cereals_data
HealthyCereals_new<- na.omit(HealthyCereals)
HealthyClust <- cbind(HealthyCereals_new, cereals2)
HealthyClust[HealthyClust$cereals2==1,]

##                         name mfr type calories protein fat sodium fiber car
bo
## 1                  100%_Bran   N    C       70       4   1    130    10
5
## 3                   All-Bran   K    C       70       4   1    260     9
7
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140    14
8
##   sugars potass vitamins shelf weight cups   rating cereals2
## 1      6    280       25     3      1 0.33 68.40297        1
## 3      5    320       25     3      1 0.33 59.42551        1
## 4      0    330       25     3      1 0.50 93.70491        1

HealthyClust[HealthyClust$cereals2==2,]

##                                       name mfr type calories protein fat so
dium
```

```
## 2                        100%_Natural_Bran   Q    C      120        3   5
15
## 8                                  Basic_4   G    C      130        3   2
210
## 14                                Clusters   G    C      110        3   2
140
## 20                      Cracklin'_Oat_Bran   K    C      110        3   3
140
## 28 Fruit_&_Fibre_Dates,_Walnuts,_and_Oats   P    C      120        3   2
160
## 29                           Fruitful_Bran   K    C      120        3   0
240
## 35                      Great_Grains_Pecan   P    C      120        3   3
75
## 40                  Just_Right_Fruit_&_Nut   K    C      140        3   1
170
## 42                                    Life   Q    C      100        4   2
150
## 45       Muesli_Raisins,_Dates,_&_Almonds   R    C      150        4   3
95
## 46      Muesli_Raisins,_Peaches,_&_Pecans   R    C      150        4   3
150
## 47                    Mueslix_Crispy_Blend   K    C      160        3   2
150
## 50                Nutri-Grain_Almond-Raisin   K    C      140        3   2
220
## 52                    Oatmeal_Raisin_Crisp   G    C      130        3   2
170
## 53                   Post_Nat._Raisin_Bran   P    C      120        3   1
200
## 57                      Quaker_Oat_Squares   Q    C      100        4   1
135
## 59                             Raisin_Bran   K    C      120        3   1
210
## 60                         Raisin_Nut_Bran   G    C      100        3   2
140
## 71                       Total_Raisin_Bran   G    C      140        3   1
190
##    fiber carbo sugars potass vitamins shelf weight cups   rating cereals2
## 2    2.0   8.0      8    135        0     3   1.00 1.00 33.98368        2
## 8    2.0  18.0      8    100       25     3   1.33 0.75 37.03856        2
## 14   2.0  13.0      7    105       25     3   1.00 0.50 40.40021        2
## 20   4.0  10.0      7    160       25     3   1.00 0.50 40.44877        2
## 28   5.0  12.0     10    200       25     3   1.25 0.67 40.91705        2
## 29   5.0  14.0     12    190       25     3   1.33 0.67 41.01549        2
## 35   3.0  13.0      4    100       25     3   1.00 0.33 45.81172        2
## 40   2.0  20.0      9     95      100     3   1.30 0.75 36.47151        2
## 42   2.0  12.0      6     95       25     2   1.00 0.67 45.32807        2
## 45   3.0  16.0     11    170       25     3   1.00 1.00 37.13686        2
## 46   3.0  16.0     11    170       25     3   1.00 1.00 34.13976        2
```

```
## 47   3.0  17.0     13    160       25     3   1.50 0.67 30.31335        2
## 50   3.0  21.0      7    130       25     3   1.33 0.67 40.69232        2
## 52   1.5  13.5     10    120       25     3   1.25 0.50 30.45084        2
## 53   6.0  11.0     14    260       25     3   1.33 0.67 37.84059        2
## 57   2.0  14.0      6    110       25     3   1.00 0.50 49.51187        2
## 59   5.0  14.0     12    240       25     2   1.33 0.75 39.25920        2
## 60   2.5  10.5      8    140       25     3   1.00 0.50 39.70340        2
## 71   4.0  15.0     14    230      100     3   1.50 1.00 28.59278        2
```

```
HealthyClust[HealthyClust$cereals2==3,]
```

```
##                         name mfr type calories protein fat sodium fiber carb
o
## 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
## 11             Cap'n'Crunch   Q    C      120       1   2    220   0.0  12.
0
## 13   Cinnamon_Toast_Crunch   G    C      120       1   3    210   0.0  13.
0
## 15              Cocoa_Puffs   G    C      110       1   1    180   0.0  12.
0
## 18                Corn_Pops   K    C      110       1   0     90   1.0  13.
0
## 19            Count_Chocula   G    C      110       1   1    180   0.0  12.
0
## 23  Crispy_Wheat_&_Raisins   G    C      100       2   1    140   2.0  11.
0
## 25              Froot_Loops   K    C      110       2   1    125   1.0  11.
0
## 26           Frosted_Flakes   K    C      110       1   0    200   1.0  14.
0
## 30           Fruity_Pebbles   P    C      110       1   1    135   0.0  13.
0
## 31             Golden_Crisp   P    C      100       2   0     45   0.0  11.
0
## 32           Golden_Grahams   G    C      110       1   1    280   0.0  15.
0
## 36          Honey_Graham_Ohs   Q    C      120       1   2    220   1.0  12.
0
## 37       Honey_Nut_Cheerios   G    C      110       3   1    250   1.5  11.
5
## 38               Honey-comb   P    C      110       1   0    180   0.0  14.
0
## 43             Lucky_Charms   G    C      110       2   1    180   0.0  12.
0
## 49         Nut&Honey_Crunch   K    C      120       2   1    190   0.0  15.
0
## 67                   Smacks   K    C      110       2   1     70   1.0   9.
```

```
0
## 74                    Trix   G    C      110       1   1    140   0.0  13.
0
## 77     Wheaties_Honey_Gold   G    C      110       2   1    200   1.0  16.
0
##    sugars potass vitamins shelf weight cups   rating cereals2
## 6      10     70       25     1      1 0.75 29.50954        3
## 7      14     30       25     2      1 1.00 33.17409        3
## 11     12     35       25     2      1 0.75 18.04285        3
## 13      9     45       25     2      1 0.75 19.82357        3
## 15     13     55       25     2      1 1.00 22.73645        3
## 18     12     20       25     2      1 1.00 35.78279        3
## 19     13     65       25     2      1 1.00 22.39651        3
## 23     10    120       25     3      1 0.75 36.17620        3
## 25     13     30       25     2      1 1.00 32.20758        3
## 26     11     25       25     1      1 0.75 31.43597        3
## 30     12     25       25     2      1 0.75 28.02576        3
## 31     15     40       25     1      1 0.88 35.25244        3
## 32      9     45       25     2      1 0.75 23.80404        3
## 36     11     45       25     2      1 1.00 21.87129        3
## 37     10     90       25     1      1 0.75 31.07222        3
## 38     11     35       25     1      1 1.33 28.74241        3
## 43     12     55       25     2      1 1.00 26.73451        3
## 49      9     40       25     2      1 0.67 29.92429        3
## 67     15     40       25     2      1 0.75 31.23005        3
## 74     12     25       25     2      1 1.00 27.75330        3
## 77      8     60       25     1      1 0.75 36.18756        3
```

```
HealthyClust[HealthyClust$cereals2==4,]
```

```
##                             name mfr type calories protein fat sodium fiber
carbo
## 9                      Bran_Chex   R    C       90       2   1    200     4
15
## 10                   Bran_Flakes   P    C       90       3   0    210     5
13
## 12                      Cheerios   G    C      110       6   2    290     2
17
## 16                     Corn_Chex   R    C      110       2   0    280     0
22
## 17                   Corn_Flakes   K    C      100       2   0    290     1
21
## 22                       Crispix   K    C      110       2   0    220     1
21
## 24                   Double_Chex   R    C      100       2   0    190     1
18
## 33             Grape_Nuts_Flakes   P    C      100       3   1    140     3
15
## 34                    Grape-Nuts   P    C      110       3   0    170     3
17
```

```
## 39 Just_Right_Crunchy__Nuggets   K    C      110       2   1    170      1
17
## 41                         Kix   G    C      110       2   1    260      0
21
## 48        Multi-Grain_Cheerios   G    C      100       2   1    220      2
15
## 51           Nutri-grain_Wheat   K    C       90       3   0    170      3
18
## 54                  Product_19   K    C      100       3   0    320      1
20
## 62                   Rice_Chex   R    C      110       1   0    240      0
23
## 63               Rice_Krispies   K    C      110       2   0    290      0
22
## 68                   Special_K   K    C      110       6   0    230      1
16
## 70           Total_Corn_Flakes   G    C      110       2   1    200      0
21
## 72           Total_Whole_Grain   G    C      100       3   1    200      3
16
## 73                     Triples   G    C      110       2   1    250      0
21
## 75                  Wheat_Chex   R    C      100       3   1    230      3
17
## 76                    Wheaties   G    C      100       3   1    200      3
17
##    sugars potass vitamins shelf weight cups   rating cereals2
## 9       6    125       25     1      1 0.67 49.12025        4
## 10      5    190       25     3      1 0.67 53.31381        4
## 12      1    105       25     1      1 1.25 50.76500        4
## 16      3     25       25     1      1 1.00 41.44502        4
## 17      2     35       25     1      1 1.00 45.86332        4
## 22      3     30       25     3      1 1.00 46.89564        4
## 24      5     80       25     3      1 0.75 44.33086        4
## 33      5     85       25     3      1 0.88 52.07690        4
## 34      3     90       25     3      1 0.25 53.37101        4
## 39      6     60      100     3      1 1.00 36.52368        4
## 41      3     40       25     2      1 1.50 39.24111        4
## 48      6     90       25     1      1 1.00 40.10596        4
## 51      2     90       25     3      1 1.00 59.64284        4
## 54      3     45      100     3      1 1.00 41.50354        4
## 62      2     30       25     1      1 1.13 41.99893        4
## 63      3     35       25     1      1 1.00 40.56016        4
## 68      3     55       25     1      1 1.00 53.13132        4
## 70      3     35      100     3      1 1.00 38.83975        4
## 72      3    110      100     3      1 1.00 46.65884        4
## 73      3     60       25     3      1 0.75 39.10617        4
## 75      3    115       25     1      1 0.67 49.78744        4
## 76      3    110       25     1      1 1.00 51.59219        4
```

```
#Mean ratings to determine the best cluster.
mean(HealthyClust[HealthyClust$cereals2==1,"rating"])

## [1] 73.84446

mean(HealthyClust[HealthyClust$cereals2==2,"rating"])

## [1] 38.37137

mean(HealthyClust[HealthyClust$cereals2==3,"rating"])

## [1] 28.66112

mean(HealthyClust[HealthyClust$cereals2==4,"rating"])

## [1] 46.17608

#Given that cluster 1's mean ratings are the highest (73.84446), we can take
cluster 1 into consideration.Group 1 might therefore be thought of as the
cluster for a healthy diet.
```