Assignment 5 - Hierarchial Clustering

ankith dasu

R Markdown

library(readr) library(tidyverse) library(cluster) library(caret) library(factoextra) library(dendextend) library(dplyr) library(treemap)

#Loading the data unto the dataframe

```
getwd()
```

```
## [1] "/Users/ankithdasu/Desktop/Spring 2022/Fundamentals of Machine Learning/Assignment 5"
```

```
setwd("/Users/ankithdasu/Desktop/Spring 2022/Fundamentals of Machine Learning/Assignment 5")
CerealData  <- read.csv("cereals.csv")

#Determining the structure of the data cereals
str(CerealData)
```

```
## '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 ...
```

Data Preparation

```
#checking for Null values , there are 4 missing values
colSums(is.na(CerealData))
```

```
##     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
```

```
#removing the missing values
CerealData <- na.omit(CerealData)

# Considering only numerical data from column 4 to 16
CerealData <- CerealData[4:16]
str(CerealData)
```

```
## 'data.frame':    74 obs. of  13 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 ...
##  $ shelf   : int  3 3 3 3 1 2 3 1 3 2 ...
##  $ 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 ...
```

```
#scaling the dataset(Z standards)

ScaledCereal <- as.data.frame(scale(CerealData))
```

Q1) Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method.

#Using Agnes method we can compare the clusting from single,complete, Average and war linkage method

```
library(cluster)

complete <- agnes(ScaledCereal, method = "complete")
complete$ac # Agglomerative coefficient here is 83 %, stronger cluster structure
```

```
## [1] 0.8353712
```

#"Single method computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2, and considers the smallest of these dissimilarities as a linkage criterion. It tends to produce long clusters."

```
single <- agnes(ScaledCereal, method = "single")
single$ac # Agglomerative coefficient is 60 %, weaker cluster structure.
```

```
## [1] 0.6067859
```

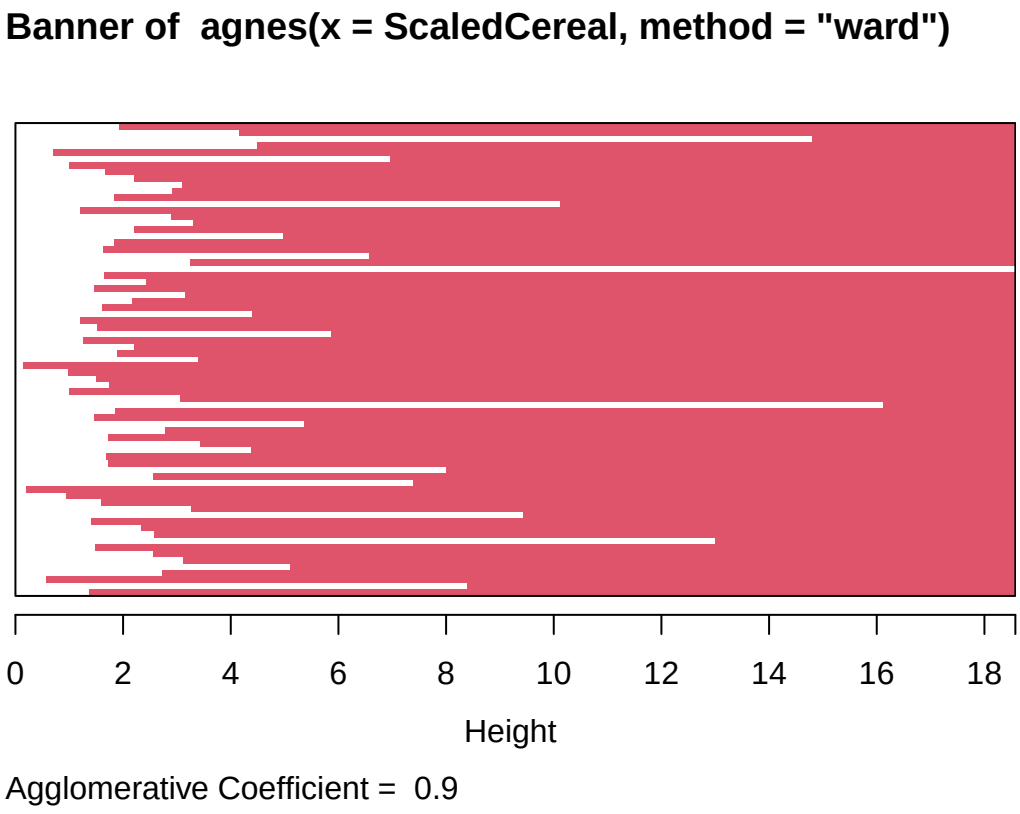
#"Average method computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2, and considers the average of these dissimilarities as the distance between the two clusters"

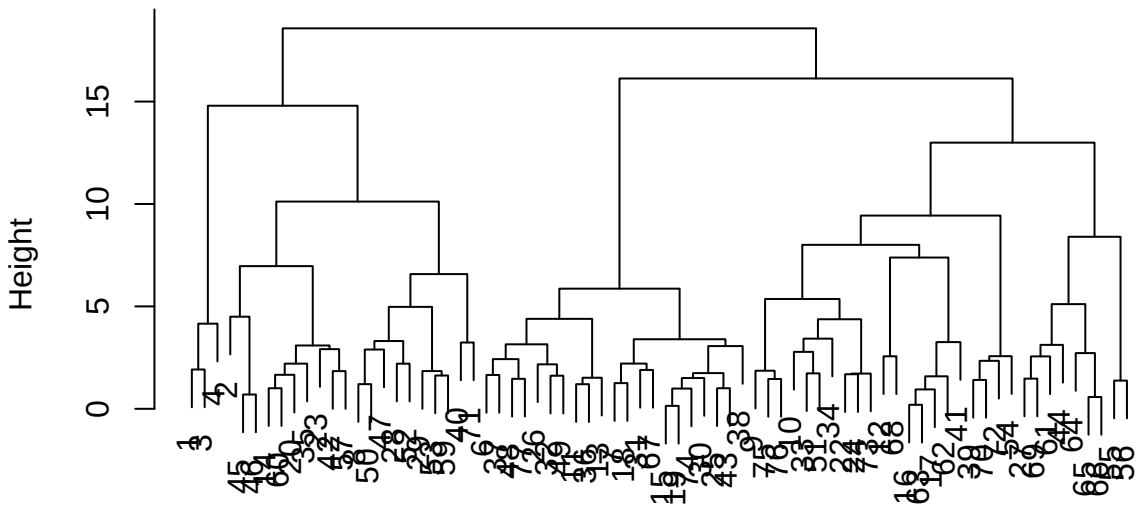
```
average <- agnes(ScaledCereal, method = "average")
average$ac #Agglomerative coefficient is 77 %, average cluster structure
```

```
## [1] 0.7766075
```

#Ward method minimizes the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged.

```
Ward  <- agnes(ScaledCereal, method = "ward")
plot(Ward)
```






```
Ward$ac # Agglomerative coefficeint is 90 %,stronger cluster structure
```

```
## [1] 0.9046042
```

#From the below Agnes function we notice that the Ward method has the strong cluster structure with the coefficient of 90%. This indicates that structure is stronger and the value is close to 1

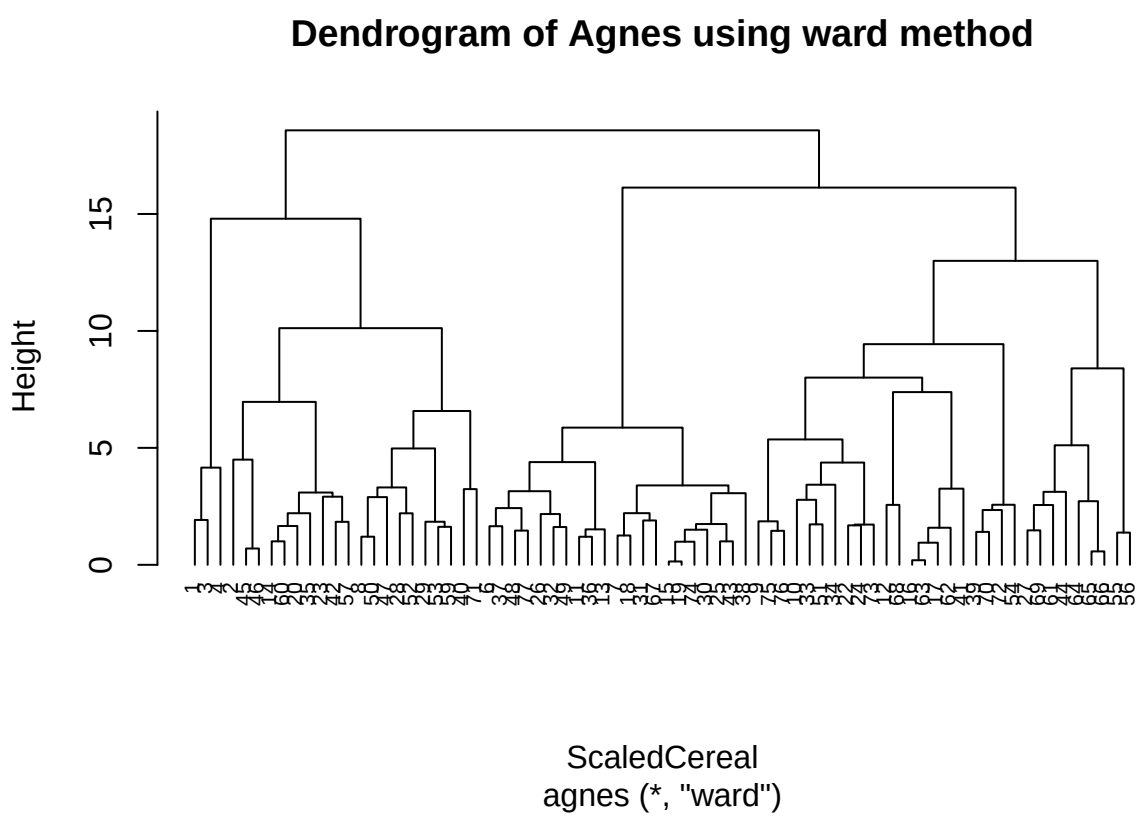
```
data.frame(complete$ac,single$ac,average$ac,Ward$ac)
```

```
##   complete.ac single.ac average.ac   Ward.ac
## 1   0.8353712 0.6067859  0.7766075 0.9046042
```

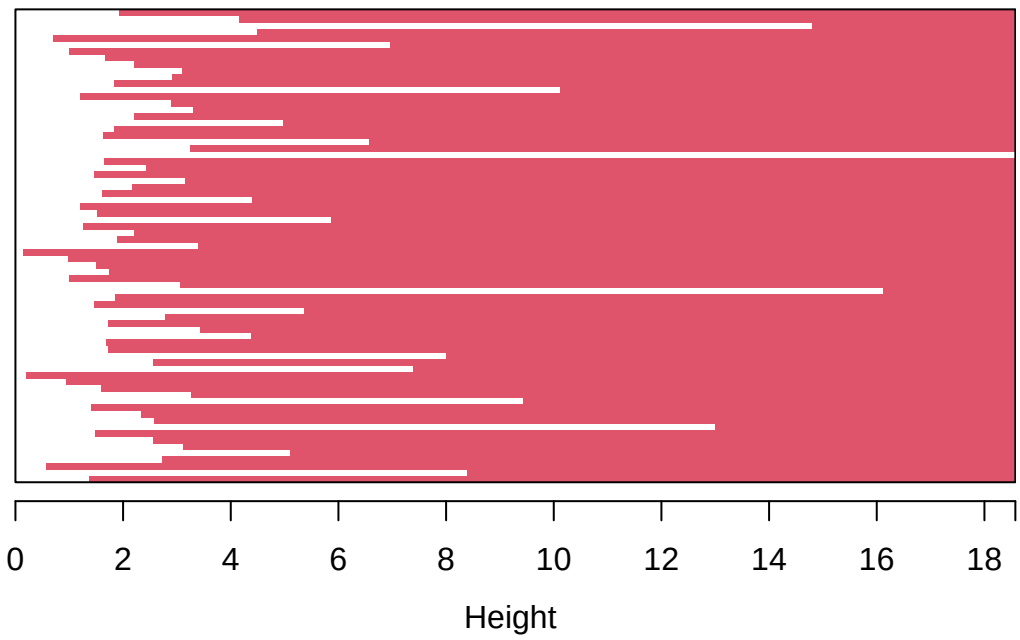
Q2) How many clusters would you choose?

```
HI_Cereal <- agnes(ScaledCereal,method="ward")

#visualizing the dendogram
pltree(HI_Cereal,cex = 0.7 , hang = -1 , main = "Dendrogram of Agnes using ward method")
```

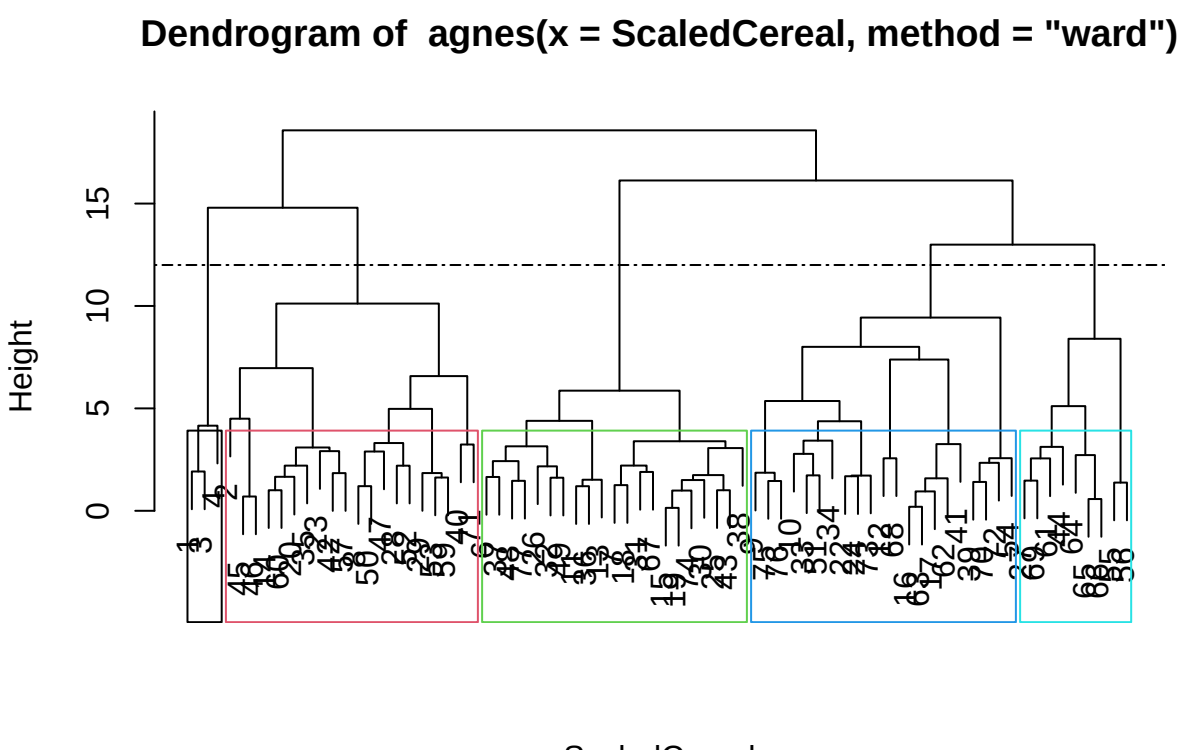



```
# k value can be determined by looking at the largest difference of height, so K =5 is the optimum.
plot(HI_Cereal)
```




```
rect.hclust(HI_Cereal, k = 5, border = 1:5,)

abline(h=12,lty=12)
```




```
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
```

```
#cutting tree
cluster_Tree = cutree(HI_Cereal,k=5)
Cereals_C <- mutate(ScaledCereal,cluster=cluster_Tree)
Cereals_C$cluster # no of cluster
```

```
##  [1] 1 2 1 1 3 3 2 4 4 3 4 3 2 3 4 4 3 3 2 4 2 4 3 3 5 2 2 3 3 3 4 4 2 3 3 3 4 2
## [39] 4 2 3 5 2 2 2 3 3 2 4 2 2 4 5 5 2 2 2 5 4 4 5 5 5 3 4 5 4 2 4 4 3 4 4 3
```

#Partitioning the data

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(tidyr)
library(factoextra)
```

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

```
# creating 80% partition of cluster data
set.seed(1234)
Partition <- createDataPartition(Cereals_C$cluster, p = 0.8 , list = FALSE)
A <- Cereals_C[Partition,]
B <- Cereals_C[-Partition,]

#Finding centroids for A by gathering features and values in partition and grouping is applied by clust

Centroid_A <- A %>% gather("features","values",-cluster) %>% group_by(cluster,features) %>% summarise(m
```

```
## `summarise()` has grouped output by 'cluster'. You can override using the
## `.groups` argument.
```

```
Centroid_A # Centroids for each cluster per column
```

```
## # A tibble: 5 x 14
## # Groups:   cluster [5]
##   cluster calories  carbo   cups    fat   fiber potass protein rating  shelf
##     <int>    <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>  <dbl>
## 1       1   -2.87  -1.73  -1.36  -0.993  4.88    3.27   1.38    3.66   0.942
## 2       2    0.738 -0.330 -0.615  0.883  0.386   0.750  0.555  -0.298  0.808
## 3       3    0.239 -0.573  0.208  0     -0.691  -0.772 -0.915  -1.01  -0.543
## 4       4   -0.176  0.825  0.355 -0.409 -0.145  -0.269  0.343   0.312  0.164
## 5       5   -1.30   0.166  0.301 -0.869 -0.0209 -0.147 -0.0126  1.54  -0.410
## # ... with 4 more variables: sodium <dbl>, sugars <dbl>, vitamins <dbl>,
## #   weight <dbl>
```

```
Cluster_B <- data.frame(data=seq(1,nrow(B),1),BClus = rep(0,nrow(B))) # Finding cluster B from the colu

#Finding the minimum distance between centroids in A and each observation in B

#this 'for' loop will  calculate the distance between centroid and each observation of B for complete l

for (x in 1:nrow(B))
  {

  Cluster_B$BClus[x] <- which.min(as.matrix(get_dist(as.data.frame(rbind(Centroid_A[-1],B[x,-length(B)]]

}
```

```
#Comparing B partition with original data
Cluster_B <- Cluster_B %>% mutate(originalcluster = B$cluster)
mean(Cluster_B$BClus == Cluster_B$originalcluster)
```

```
## [1] 0.9230769
```

"Based on the above analysis both the original and predicted cluste matches at a percentage of 92. So here, the cluster has a good stability but some distances might be huge, which is causing it not to be 100%. This might be due to the data partition."

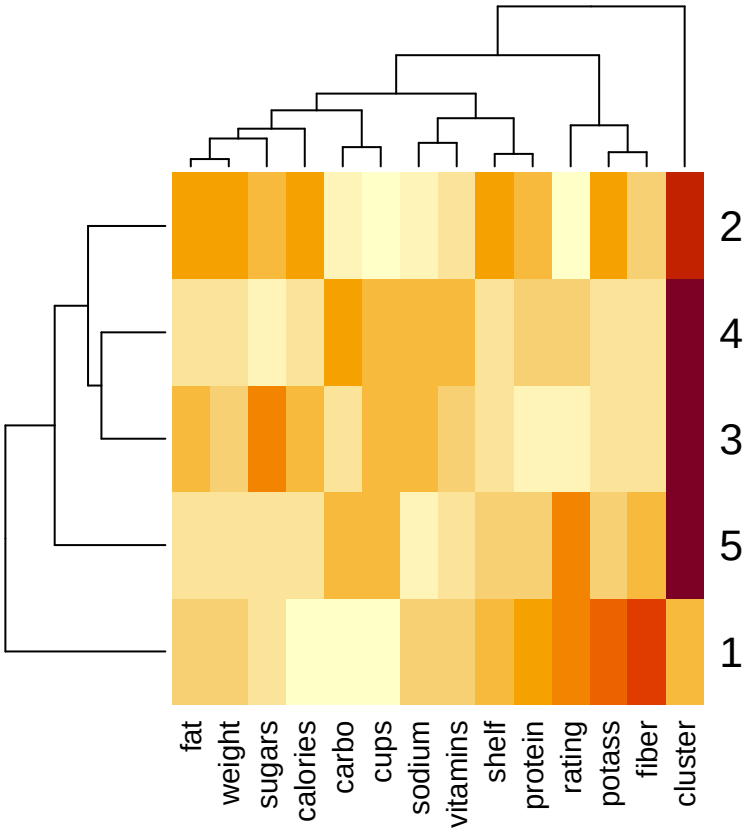
Q3)

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."

```
Healthy_Cereals <- split(Cereals_C,Cereals_C$cluster) # splitting Cereal cluster from cereal_c data fra
Healthy_MC <-lapply(Healthy_Cereals,colMeans) # Lapply is used to apply a function over a dataframe and
(centroids <- do.call(rbind,Healthy_MC)) # binding data frame
```

```
##     calories     protein        fat      sodium       fiber      carbo
## 1 -2.2018711  1.38174776 -0.3310734  0.17279012  3.64131237 -2.0718749
## 2  0.8553248  0.59163927  0.9435592 -0.08898011  0.38141771 -0.2003584
## 3  0.1978117 -0.91996886  0.0000000  0.12101140 -0.66198437 -0.5423583
## 4 -0.1621407  0.18662567 -0.4729620  0.77112209 -0.21003997  0.9626860
## 5 -1.2499969 -0.06420242 -0.8828625 -1.94150793 -0.02664224  0.1551013
##       sugars     potass    vitamins      shelf     weight       cups     rating
## 1 -0.7894824  2.9837813 -0.18184220  0.9419715 -0.2008324 -1.8452553  2.2426479
## 2  0.5143002  0.7475659  0.09849786  0.8217889  0.9235649 -0.5477863 -0.2928786
## 3  0.9583619 -0.7415648 -0.18184220 -0.6604628 -0.2008324  0.2779676 -0.9636465
## 4 -0.8659505 -0.3485391  0.45893508 -0.1453946 -0.2008324  0.4577648  0.2916795
## 5 -1.0953551 -0.1122758 -0.80482011 -0.2598542 -1.0482044  0.1156788  1.4712151
##   cluster
## 1       1
## 2       2
## 3       3
## 4       4
## 5       5
```

```
heatmap(centroids) # heatmap of the centroids.
```



"Based on the above analysis, here we observe that the Bran Cerelas (Cluster 1) is recommended for children as it has high fiber, protein,potassium and low on sugars,calories and carbs."