ASSIGNMENT_05

Yash_Patel

2022-12-01

Loading the required packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
library(caret)
```

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

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.16.0
## 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(knitr)
library(factoextra)
```

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

Importing and cleaning the dataset

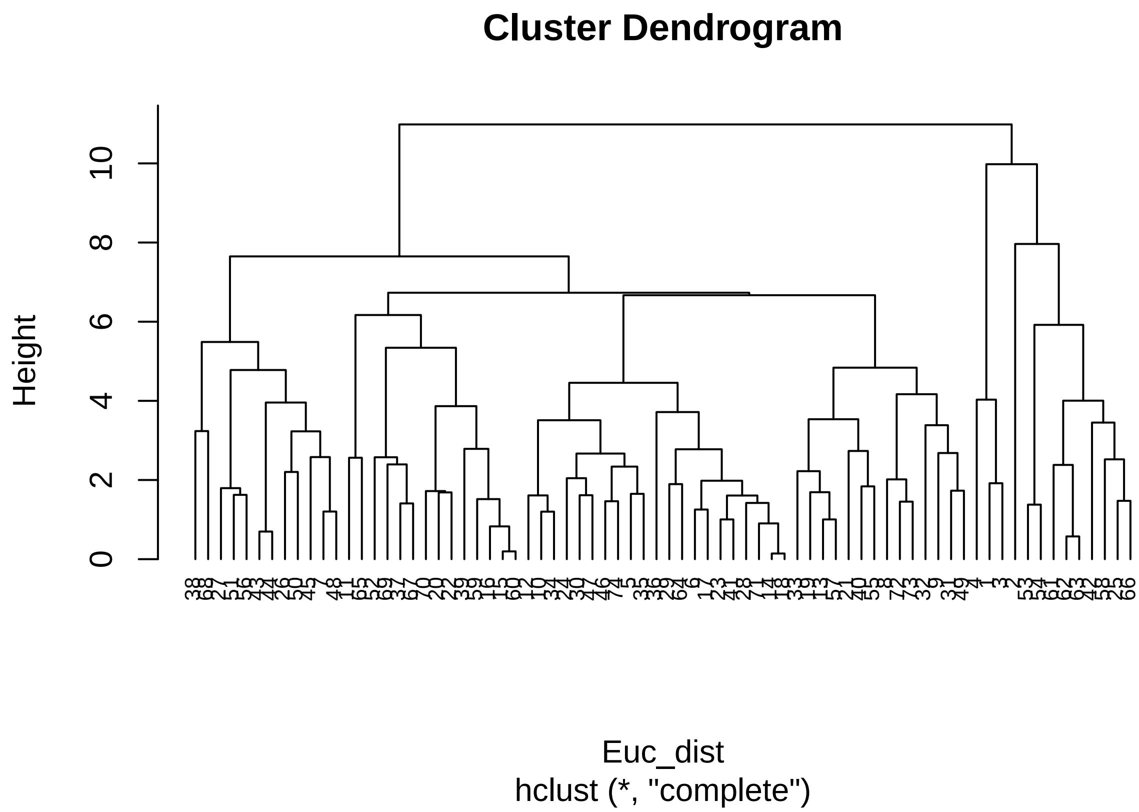
```
Cereals<- read.csv("C:/Users/YASH/Downloads/Cereals.csv")
Data_cereals <- data.frame(Cereals[,4:16]) %>% drop_na()
```

data normalization

```
Cereals_norm <- scale(Data_cereals)
```

Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements.

```
Euc_dist <- dist(Cereals_norm, method = "euclidean")
hc_complete <- hclust(Euc_dist, method = "complete")
#Plotting the dendogram
plot(hc_complete, cex = 0.7, hang = -1)
```



Using agnes function to perfrom clustering with single linkage, complete linkage, average linkage and Ward. And finding the best method

```
hc_single <- agnes(Cereals_norm, method = "single")
hc_complete <- agnes(Cereals_norm, method = "complete")
hc_avg <- agnes(Cereals_norm, method = "average")
hc_ward <- agnes(Cereals_norm, method = "ward")
print(hc_single$ac)
```

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

```
print(hc_complete$ac)
```

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

```
print(hc_avg$ac)
```

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

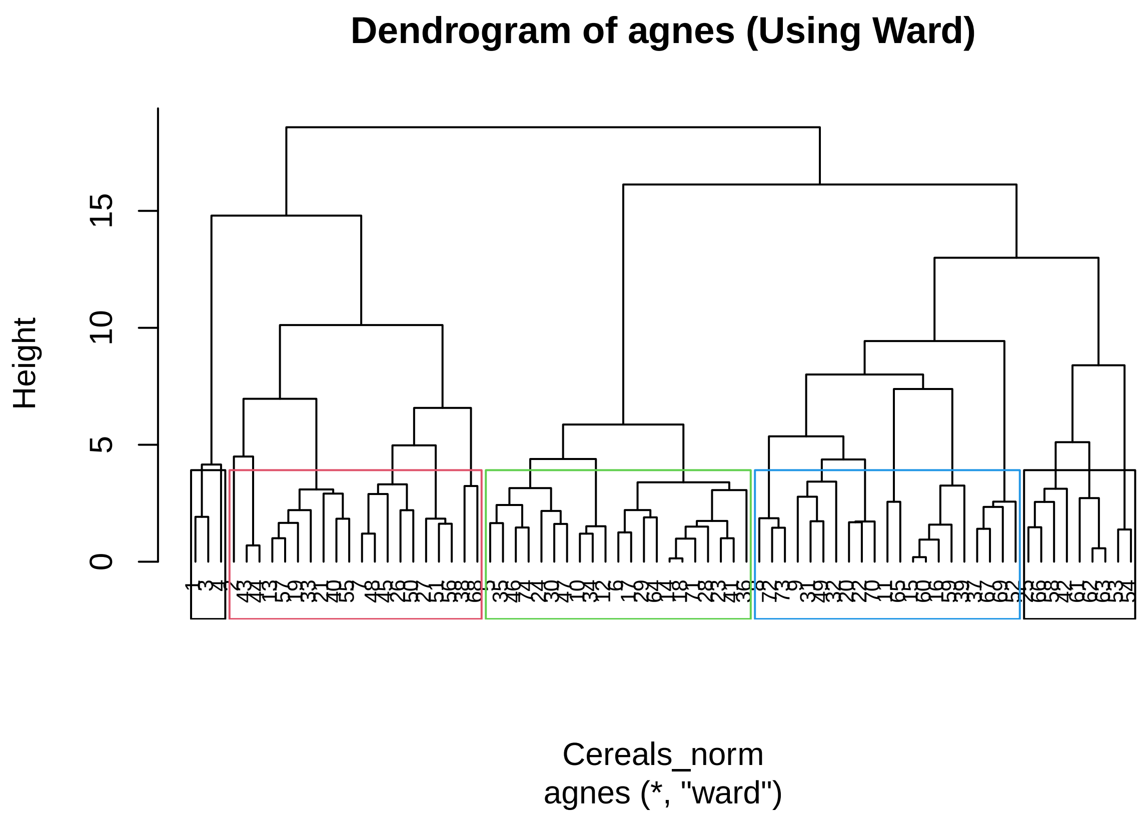
```
print(hc_ward$ac)
```

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

Here as we can see the accuracy of the Ward method is High (0.9597071) so we can consider it as a best linkage method.

How many clusters would you choose?

```
pltree(hc_ward, cex = 0.7, hang = -1, main = "Dendrogram of agnes (Using Ward)")
rect.hclust(hc_ward, k = 5, border = 1:4)
```




```
Clust_01 <- cutree(hc_ward, k=5)
clust_a <- as.data.frame(cbind(Cereals_norm,Clust_01))
```

5 clusters.

Comment on the structure of the clusters and on their stability.

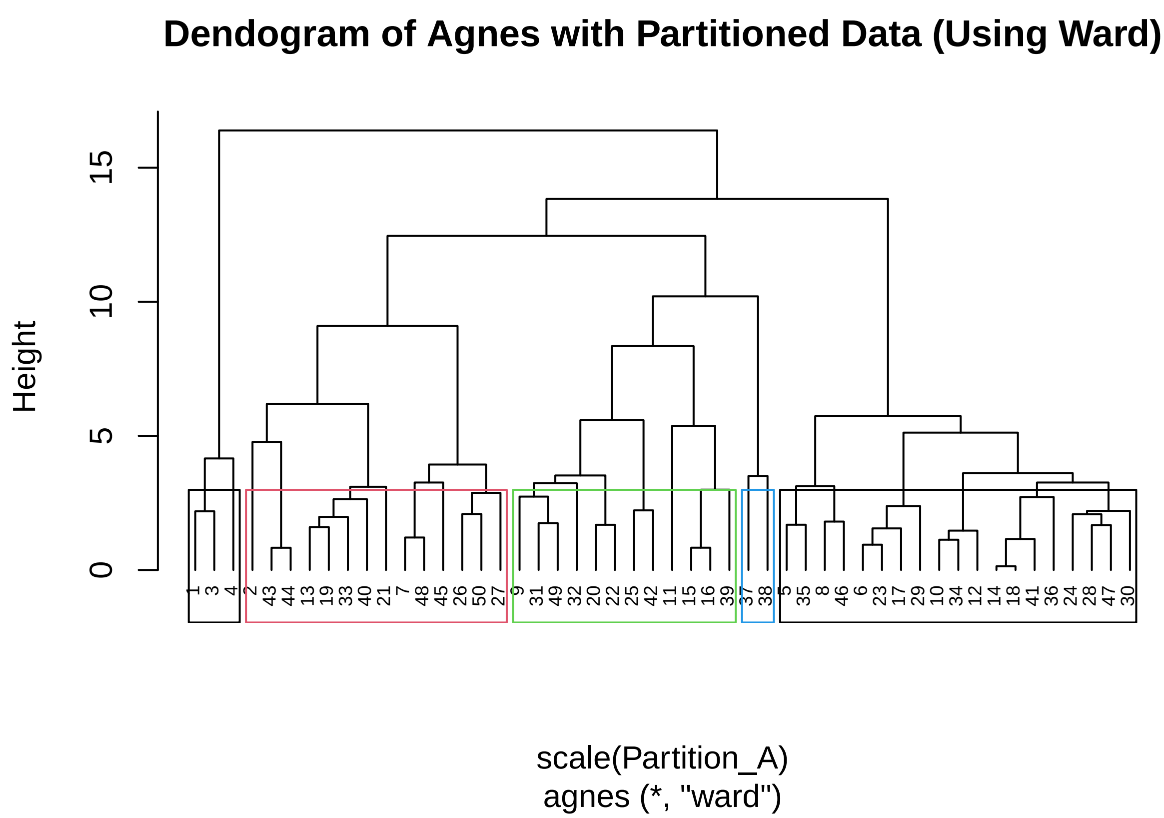
```
#Creating Partitions
set.seed(45320)
Partition_A <- Data_cereals[1:50,]
Partition_B <- Data_cereals[51:74,]
```

Performing Hierarchial Clustering,taking value of K as 5.

```
Hc_single_01 <- agnes(scale(Partition_A), method = "single")
Hc_complete_01 <- agnes(scale(Partition_A), method = "complete")
Hc_avg_01 <- agnes(scale(Partition_A), method = "average")
Hc_ward_01 <- agnes(scale(Partition_A), method = "ward")
cbind(single=Hc_single_01$ac , complete=Hc_complete_01$ac , average= Hc_avg_01$ac , ward= Hc_ward_01$ac
```

```
##          single  complete   average      ward
## [1,] 0.6393338 0.8138238 0.7408904 0.8764323
```

```
pltree(Hc_ward_01, cex = 0.6, hang = -1, main = "Dendogram of Agnes with Partitioned Data (Using Ward)"
rect.hclust(Hc_ward_01, k = 5, border = 1:4)
```




```
clust_02 <- cutree(Hc_ward_01, k = 5)
```

Use the cluster centroids from A to assign each record in partition B (each record is assigned to the cluster with the closest centroid).Assess how consistent the cluster assignments are compared to the assignments based on all the data.

```r
clust_b <- as.data.frame(cbind(Partition_A, clust_02))
clust_b[clust_b$clust_02==1,]
```

```
##   calories protein fat sodium fiber carbo sugars potass vitamins shelf weight
## 1       70       4   1    130    10     5      6    280       25     3      1
## 3       70       4   1    260     9     7      5    320       25     3      1
## 4       50       4   0    140    14     8      0    330       25     3      1
##   cups   rating clust_02
## 1 0.33 68.40297        1
## 3 0.33 59.42551        1
## 4 0.50 93.70491        1
```

```r
centroid_01  <- colMeans(clust_b[clust_b$clust_02==1,])
clust_b[clust_b$clust_02==2,]
```

```
##    calories protein fat sodium fiber carbo sugars potass vitamins shelf weight
## 2       120       3   5     15   2.0   8.0      8    135        0     3   1.00
## 7       130       3   2    210   2.0  18.0      8    100       25     3   1.33
## 13      110       3   2    140   2.0  13.0      7    105       25     3   1.00
## 19      110       3   3    140   4.0  10.0      7    160       25     3   1.00
## 21      100       2   1    140   2.0  11.0     10    120       25     3   1.00
## 26      120       3   2    160   5.0  12.0     10    200       25     3   1.25
## 27      120       3   0    240   5.0  14.0     12    190       25     3   1.33
## 33      120       3   3     75   3.0  13.0      4    100       25     3   1.00
## 40      100       4   2    150   2.0  12.0      6     95       25     2   1.00
## 43      150       4   3     95   3.0  16.0     11    170       25     3   1.00
## 44      150       4   3    150   3.0  16.0     11    170       25     3   1.00
## 45      160       3   2    150   3.0  17.0     13    160       25     3   1.50
## 48      140       3   2    220   3.0  21.0      7    130       25     3   1.33
## 50      130       3   2    170   1.5  13.5     10    120       25     3   1.25
##    cups   rating clust_02
## 2  1.00 33.98368        2
## 7  0.75 37.03856        2
## 13 0.50 40.40021        2
## 19 0.50 40.44877        2
## 21 0.75 36.17620        2
## 26 0.67 40.91705        2
## 27 0.67 41.01549        2
## 33 0.33 45.81172        2
## 40 0.67 45.32807        2
## 43 1.00 37.13686        2
## 44 1.00 34.13976        2
## 45 0.67 30.31335        2
## 48 0.67 40.69232        2
## 50 0.50 30.45084        2
```

```r
centroid_02  <- colMeans(clust_b[clust_b$clust_02==2,])
clust_b[clust_b$clust_02==3,]
```

```
##    calories protein fat sodium fiber carbo sugars potass vitamins shelf weight
## 5       110       2   2    180   1.5  10.5     10     70       25     1      1
```

```
## 6        110        2   0    125   1.0  11.0      14     30        25     2       1
## 8         90        2   1    200   4.0  15.0       6    125        25     1       1
## 10       120        1   2    220   0.0  12.0      12     35        25     2       1
## 12       120        1   3    210   0.0  13.0       9     45        25     2       1
## 14       110        1   1    180   0.0  12.0      13     55        25     2       1
## 17       110        1   0     90   1.0  13.0      12     20        25     2       1
## 18       110        1   1    180   0.0  12.0      13     65        25     2       1
## 23       110        2   1    125   1.0  11.0      13     30        25     2       1
## 24       110        1   0    200   1.0  14.0      11     25        25     1       1
## 28       110        1   1    135   0.0  13.0      12     25        25     2       1
## 29       100        2   0     45   0.0  11.0      15     40        25     1       1
## 30       110        1   1    280   0.0  15.0       9     45        25     2       1
## 34       120        1   2    220   1.0  12.0      11     45        25     2       1
## 35       110        3   1    250   1.5  11.5      10     90        25     1       1
## 36       110        1   0    180   0.0  14.0      11     35        25     1       1
## 41       110        2   1    180   0.0  12.0      12     55        25     2       1
## 46       100        2   1    220   2.0  15.0       6     90        25     1       1
## 47       120        2   1    190   0.0  15.0       9     40        25     2       1
##    cups   rating clust_02
## 5  0.75 29.50954        3
## 6  1.00 33.17409        3
## 8  0.67 49.12025        3
## 10 0.75 18.04285        3
## 12 0.75 19.82357        3
## 14 1.00 22.73645        3
## 17 1.00 35.78279        3
## 18 1.00 22.39651        3
## 23 1.00 32.20758        3
## 24 0.75 31.43597        3
## 28 0.75 28.02576        3
## 29 0.88 35.25244        3
## 30 0.75 23.80404        3
## 34 1.00 21.87129        3
## 35 0.75 31.07222        3
## 36 1.33 28.74241        3
## 41 1.00 26.73451        3
## 46 1.00 40.10596        3
## 47 0.67 29.92429        3
```

```
centroid_03 <- colMeans(clust_b[clust_b$clust_02==3,])
clust_b[clust_b$clust_02==4,]
```

```
##    calories protein fat sodium fiber carbo sugars potass vitamins shelf weight
## 9        90       3   0    210     5    13      5    190       25     3      1
## 11      110       6   2    290     2    17      1    105       25     1      1
## 15      110       2   0    280     0    22      3     25       25     1      1
## 16      100       2   0    290     1    21      2     35       25     1      1
## 20      110       2   0    220     1    21      3     30       25     3      1
## 22      100       2   0    190     1    18      5     80       25     3      1
## 25      100       3   0      0     3    14      7    100       25     2      1
## 31      100       3   1    140     3    15      5     85       25     3      1
## 32      110       3   0    170     3    17      3     90       25     3      1
## 39      110       2   1    260     0    21      3     40       25     2      1
## 42      100       4   1      0     0    16      3     95       25     2      1
```

```
## 49       90       3   0    170     3    18      2     90       25     3      1
##    cups   rating clust_02
## 9  0.67 53.31381        4
## 11 1.25 50.76500        4
## 15 1.00 41.44502        4
## 16 1.00 45.86332        4
## 20 1.00 46.89564        4
## 22 0.75 44.33086        4
## 25 0.80 58.34514        4
## 31 0.88 52.07690        4
## 32 0.25 53.37101        4
## 39 1.50 39.24111        4
## 42 1.00 54.85092        4
## 49 1.00 59.64284        4
```

```
centroid_04 <- colMeans(clust_b[clust_b$clust_02==4,])
main.centroid <- rbind(centroid_01 , centroid_02 , centroid_03, centroid_04)
var_x <- as.data.frame(rbind(main.centroid[,-14], Partition_B))
Dist_1  <- get_dist(var_x)
Data_cere_mat <- as.matrix(Dist_1 )
clust_c <- data.frame(data=seq(1,nrow(Partition_B),1), Clusters = rep(0,nrow(Partition_B)))
for(i in 1:nrow(Partition_B))
{clust_c[i,2] <- which.min(Data_cere_mat[i+4, 1:4])}
clust_c
```

```
##    data Clusters
## 1     1        1
## 2     2        4
## 3     3        3
## 4     4        2
## 5     5        2
## 6     6        1
## 7     7        2
## 8     8        2
## 9     9        3
## 10   10        3
## 11   11        2
## 12   12        2
## 13   13        2
## 14   14        3
## 15   15        4
## 16   16        2
## 17   17        3
## 18   18        2
## 19   19        4
## 20   20        4
## 21   21        3
## 22   22        4
## 23   23        4
## 24   24        3
```

```
cbind(clust_a$Clust_01[51:74], clust_c$Clusters)
```

```
##       [,1] [,2]
```

```
## [1,]    2    1
## [2,]    4    4
## [3,]    5    3
## [4,]    5    2
## [5,]    2    2
## [6,]    2    1
## [7,]    2    2
## [8,]    5    2
## [9,]    4    3
## [10,]    4    3
## [11,]    5    2
## [12,]    5    2
## [13,]    5    2
## [14,]    3    3
## [15,]    4    4
## [16,]    5    2
## [17,]    4    3
## [18,]    2    2
## [19,]    4    4
## [20,]    4    4
## [21,]    3    3
## [22,]    4    4
## [23,]    4    4
## [24,]    3    3
```

```
table(clust_a$Clust_01[51:74] == clust_c$Clusters)
```

```
##
## FALSE  TRUE
##    12    12
```

Given that we receive 12 FALSE and 12 TRUE, we can claim that the model is only partially stable.

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?

```
Healthy_Cereals <- Cereals %>% drop_na()
Healthy_diet_clust <- cbind(Healthy_Cereals, Clust_01)
Healthy_diet_clust[Healthy_diet_clust$Clust_01==1,]
```

```
##                          name mfr type calories protein fat sodium fiber carbo
## 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 Clust_01
## 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
```

```
Healthy_diet_clust[Healthy_diet_clust$Clust_01==2,]
```

```
##                                     name mfr type calories protein fat sodium
## 2                      100%_Natural_Bran   Q    C      120       3   5     15
## 7                                Basic_4   G    C      130       3   2    210
## 13                              Clusters   G    C      110       3   2    140
## 19                    Cracklin'_Oat_Bran   K    C      110       3   3    140
## 21                Crispy_Wheat_&_Raisins   G    C      100       2   1    140
## 26 Fruit_&_Fibre_Dates,_Walnuts,_and_Oats   P    C      120       3   2    160
## 27                          Fruitful_Bran   K    C      120       3   0    240
## 33                    Great_Grains_Pecan   P    C      120       3   3     75
## 38                Just_Right_Fruit_&_Nut   K    C      140       3   1    170
## 40                                  Life   Q    C      100       4   2    150
## 43      Muesli_Raisins,_Dates,_&_Almonds   R    C      150       4   3     95
## 44     Muesli_Raisins,_Peaches,_&_Pecans   R    C      150       4   3    150
## 45                  Mueslix_Crispy_Blend   K    C      160       3   2    150
## 48             Nutri-Grain_Almond-Raisin   K    C      140       3   2    220
## 50                  Oatmeal_Raisin_Crisp   G    C      130       3   2    170
## 51                 Post_Nat._Raisin_Bran   P    C      120       3   1    200
## 55                    Quaker_Oat_Squares   Q    C      100       4   1    135
## 56                           Raisin_Bran   K    C      120       3   1    210
## 57                       Raisin_Nut_Bran   G    C      100       3   2    140
## 68                      Total_Raisin_Bran   G    C      140       3   1    190
##    fiber carbo sugars potass vitamins shelf weight cups   rating Clust_01
## 2    2.0   8.0      8    135        0     3   1.00 1.00 33.98368        2
## 7    2.0  18.0      8    100       25     3   1.33 0.75 37.03856        2
## 13   2.0  13.0      7    105       25     3   1.00 0.50 40.40021        2
## 19   4.0  10.0      7    160       25     3   1.00 0.50 40.44877        2
## 21   2.0  11.0     10    120       25     3   1.00 0.75 36.17620        2
## 26   5.0  12.0     10    200       25     3   1.25 0.67 40.91705        2
## 27   5.0  14.0     12    190       25     3   1.33 0.67 41.01549        2
## 33   3.0  13.0      4    100       25     3   1.00 0.33 45.81172        2
## 38   2.0  20.0      9     95      100     3   1.30 0.75 36.47151        2
## 40   2.0  12.0      6     95       25     2   1.00 0.67 45.32807        2
## 43   3.0  16.0     11    170       25     3   1.00 1.00 37.13686        2
## 44   3.0  16.0     11    170       25     3   1.00 1.00 34.13976        2
## 45   3.0  17.0     13    160       25     3   1.50 0.67 30.31335        2
## 48   3.0  21.0      7    130       25     3   1.33 0.67 40.69232        2
## 50   1.5  13.5     10    120       25     3   1.25 0.50 30.45084        2
## 51   6.0  11.0     14    260       25     3   1.33 0.67 37.84059        2
## 55   2.0  14.0      6    110       25     3   1.00 0.50 49.51187        2
## 56   5.0  14.0     12    240       25     2   1.33 0.75 39.25920        2
## 57   2.5  10.5      8    140       25     3   1.00 0.50 39.70340        2
## 68   4.0  15.0     14    230      100     3   1.50 1.00 28.59278        2
```

```
Healthy_diet_clust[Healthy_diet_clust$Clust_01==3,]
```

```
##                       name mfr type calories protein fat sodium fiber carbo
## 5  Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
## 6              Apple_Jacks   K    C      110       2   0    125   1.0  11.0
## 10             Cap'n'Crunch   Q    C      120       1   2    220   0.0  12.0
## 12    Cinnamon_Toast_Crunch   G    C      120       1   3    210   0.0  13.0
```

```
## 14              Cocoa_Puffs   G    C      110        1   1    180   0.0  12.0
## 17                Corn_Pops   K    C      110        1   0     90   1.0  13.0
## 18           Count_Chocula   G    C      110        1   1    180   0.0  12.0
## 23              Froot_Loops   K    C      110        2   1    125   1.0  11.0
## 24           Frosted_Flakes   K    C      110        1   0    200   1.0  14.0
## 28           Fruity_Pebbles   P    C      110        1   1    135   0.0  13.0
## 29             Golden_Crisp   P    C      100        2   0     45   0.0  11.0
## 30           Golden_Grahams   G    C      110        1   1    280   0.0  15.0
## 34         Honey_Graham_Ohs   Q    C      120        1   2    220   1.0  12.0
## 35       Honey_Nut_Cheerios   G    C      110        3   1    250   1.5  11.5
## 36               Honey-comb   P    C      110        1   0    180   0.0  14.0
## 41             Lucky_Charms   G    C      110        2   1    180   0.0  12.0
## 46     Multi-Grain_Cheerios   G    C      100        2   1    220   2.0  15.0
## 47         Nut&Honey_Crunch   K    C      120        2   1    190   0.0  15.0
## 64                   Smacks   K    C      110        2   1     70   1.0   9.0
## 71                     Trix   G    C      110        1   1    140   0.0  13.0
## 74      Wheaties_Honey_Gold   G    C      110        2   1    200   1.0  16.0
##    sugars potass vitamins shelf weight cups   rating Clust_01
## 5      10     70       25     1      1 0.75 29.50954        3
## 6      14     30       25     2      1 1.00 33.17409        3
## 10     12     35       25     2      1 0.75 18.04285        3
## 12      9     45       25     2      1 0.75 19.82357        3
## 14     13     55       25     2      1 1.00 22.73645        3
## 17     12     20       25     2      1 1.00 35.78279        3
## 18     13     65       25     2      1 1.00 22.39651        3
## 23     13     30       25     2      1 1.00 32.20758        3
## 24     11     25       25     1      1 0.75 31.43597        3
## 28     12     25       25     2      1 0.75 28.02576        3
## 29     15     40       25     1      1 0.88 35.25244        3
## 30      9     45       25     2      1 0.75 23.80404        3
## 34     11     45       25     2      1 1.00 21.87129        3
## 35     10     90       25     1      1 0.75 31.07222        3
## 36     11     35       25     1      1 1.33 28.74241        3
## 41     12     55       25     2      1 1.00 26.73451        3
## 46      6     90       25     1      1 1.00 40.10596        3
## 47      9     40       25     2      1 0.67 29.92429        3
## 64     15     40       25     2      1 0.75 31.23005        3
## 71     12     25       25     2      1 1.00 27.75330        3
## 74      8     60       25     1      1 0.75 36.18756        3
```

```
Healthy_diet_clust[Healthy_diet_clust$Clust_01==4,]
```

```
##                               name mfr type calories protein fat sodium fiber carbo
## 8                        Bran_Chex   R    C       90       2   1    200     4    15
## 9                      Bran_Flakes   P    C       90       3   0    210     5    13
## 11                        Cheerios   G    C      110       6   2    290     2    17
## 15                       Corn_Chex   R    C      110       2   0    280     0    22
## 16                     Corn_Flakes   K    C      100       2   0    290     1    21
## 20                         Crispix   K    C      110       2   0    220     1    21
## 22                     Double_Chex   R    C      100       2   0    190     1    18
## 31                Grape_Nuts_Flakes   P    C      100       3   1    140     3    15
## 32                      Grape-Nuts   P    C      110       3   0    170     3    17
## 37 Just_Right_Crunchy__Nuggets   K    C      110       2   1    170     1    17
## 39                             Kix   G    C      110       2   1    260     0    21
```

```
## 49          Nutri-grain_Wheat   K     C       90        3    0     170      3     18
## 52                 Product_19   K     C      100        3    0     320      1     20
## 59                  Rice_Chex   R     C      110        1    0     240      0     23
## 60               Rice_Krispies  K     C      110        2    0     290      0     22
## 65                  Special_K   K     C      110        6    0     230      1     16
## 67          Total_Corn_Flakes   G     C      110        2    1     200      0     21
## 69          Total_Whole_Grain   G     C      100        3    1     200      3     16
## 70                    Triples   G     C      110        2    1     250      0     21
## 72                 Wheat_Chex   R     C      100        3    1     230      3     17
## 73                   Wheaties   G     C      100        3    1     200      3     17
##    sugars potass vitamins shelf weight cups   rating Clust_01
## 8       6    125       25     1      1 0.67 49.12025        4
## 9       5    190       25     3      1 0.67 53.31381        4
## 11      1    105       25     1      1 1.25 50.76500        4
## 15      3     25       25     1      1 1.00 41.44502        4
## 16      2     35       25     1      1 1.00 45.86332        4
## 20      3     30       25     3      1 1.00 46.89564        4
## 22      5     80       25     3      1 0.75 44.33086        4
## 31      5     85       25     3      1 0.88 52.07690        4
## 32      3     90       25     3      1 0.25 53.37101        4
## 37      6     60      100     3      1 1.00 36.52368        4
## 39      3     40       25     2      1 1.50 39.24111        4
## 49      2     90       25     3      1 1.00 59.64284        4
## 52      3     45      100     3      1 1.00 41.50354        4
## 59      2     30       25     1      1 1.13 41.99893        4
## 60      3     35       25     1      1 1.00 40.56016        4
## 65      3     55       25     1      1 1.00 53.13132        4
## 67      3     35      100     3      1 1.00 38.83975        4
## 69      3    110      100     3      1 1.00 46.65884        4
## 70      3     60       25     3      1 0.75 39.10617        4
## 72      3    115       25     1      1 0.67 49.78744        4
## 73      3    110       25     1      1 1.00 51.59219        4
```

```r
#Mean ratings to determine the best cluster.
mean(Healthy_diet_clust[Healthy_diet_clust$Clust_01==1,"rating"])
```

```
## [1] 73.84446
```

```r
mean(Healthy_diet_clust[Healthy_diet_clust$Clust_01==2,"rating"])
```

```
## [1] 38.26161
```

```r
mean(Healthy_diet_clust[Healthy_diet_clust$Clust_01==3,"rating"])
```

```
## [1] 28.84825
```

```r
mean(Healthy_diet_clust[Healthy_diet_clust$Clust_01==4,"rating"])
```

```
## [1] 46.46513
```

Since cluster 1's mean ratings are the highest (i.e. 73.84446), we can take that into consideration.