Assignment5_Hierarchical Clustering

Ram

11/29/2021

Setting up working directory

```
setwd("C:/Users/ramne/Desktop/ML Assignment/Hierarchical Clustering")
set.seed(123)
```

Loading required libraries.

```
library(cluster)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(dendextend)

##
## ---------------------
## Welcome to dendextend version 1.15.2
## 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
```

Data Importing cereals dataset

```
library(readr)
cereals<-read.csv("Cereals.csv")
DataFrame <- data.frame(cereals[,4:16])
```

Data Pre-Processing

To remove any missing value that might be present in the data.

```
OmitMissing <- na.omit(DataFrame)
```

Data Normalization & Data Scaling:

Normalizing the Data using Scale function.

```
Normalise <- scale(OmitMissing)
```

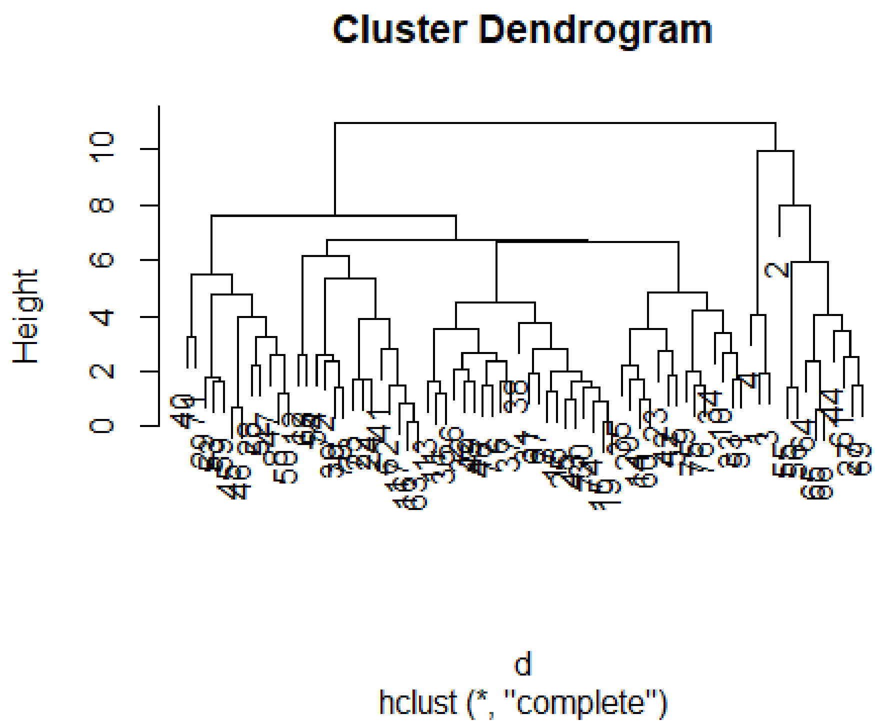
Using the euclidean distance to measure the distance:

Computing the dissimilarity matrix values by using Dist and the method is Euclidean.

```
d <- dist(Normalise, method = "euclidean")
```

Perform Hierarchical Clustering using complete linkage.

```
HC <- hclust(d, method = "complete")
plot(HC)
```



Plotting the dendogram.

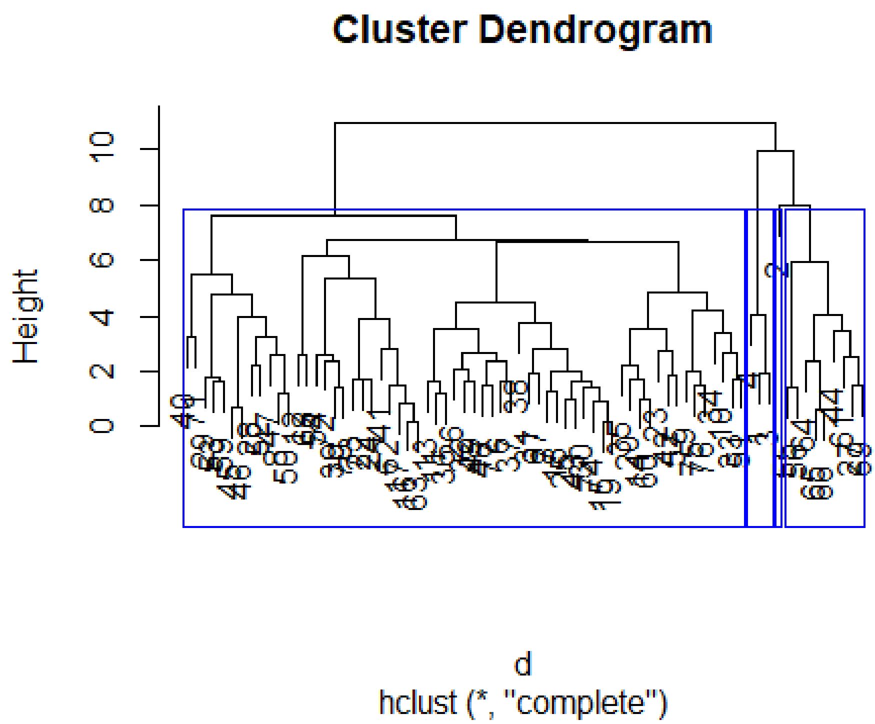
```
round(HC$height, 3)
```

```
##  [1]  0.143  0.196  0.575  0.698  0.828  0.904  1.003  1.004  1.201  1.203
## [11]  1.254  1.378  1.408  1.421  1.454  1.463  1.474  1.517  1.608  1.611
## [21]  1.616  1.625  1.650  1.687  1.692  1.720  1.730  1.795  1.839  1.897
## [31]  1.919  1.982  2.015  2.046  2.203  2.224  2.339  2.381  2.394  2.522
## [41]  2.563  2.574  2.579  2.668  2.682  2.734  2.776  2.787  3.229  3.236
## [51]  3.385  3.451  3.510  3.535  3.717  3.866  3.957  4.005  4.031  4.168
## [61]  4.456  4.779  4.839  5.342  5.488  5.920  6.169  6.669  6.731  7.650
## [71]  7.964  9.979 10.984
```

Determining Optimal Clusters:

Highliting the clusters directly in dendogram

```
plot(HC)
rect.hclust(HC,
  k = 4, # k is used to specify the number of clusters
  border = "Blue"
)
```



We can also use agnes() function to perform clustering.

Performing clustering using agnes() with single, complete, average and ward.

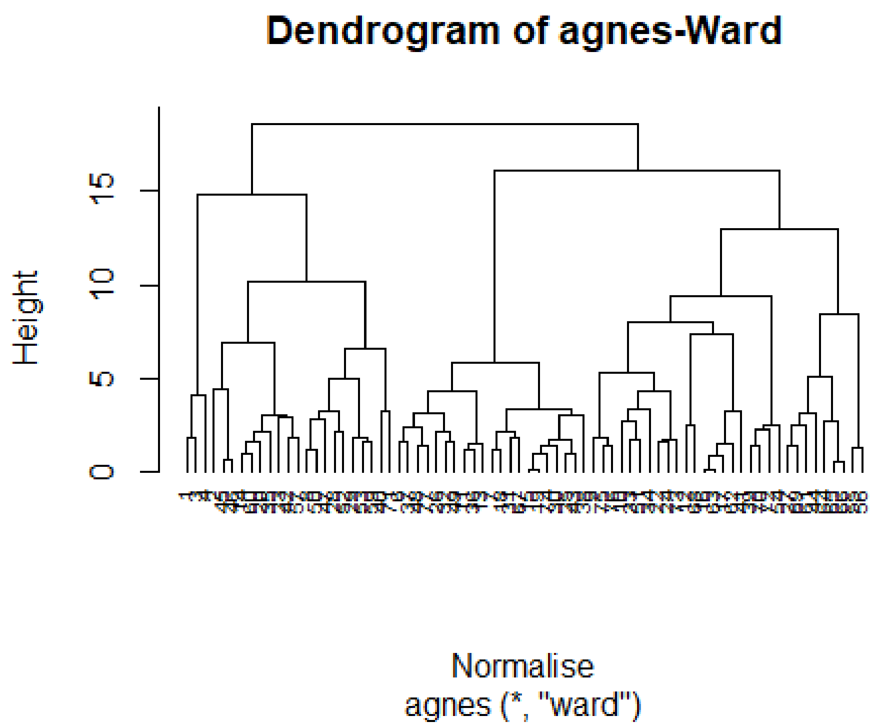
```
HCsingle <- agnes(Normalise, method = "single")
HCcomplete <- agnes(Normalise, method = "complete")
HCaverage <- agnes(Normalise, method = "average")
HCward <- agnes(Normalise, method = "ward")
```

Now we will compare the agglomerative coefficients for Single,complete, average and ward.

```
print(HCsingle$ac)

## [1] 0.6067859

print(HCcomplete$ac)

## [1] 0.8353712

print(HCaverage$ac)

## [1] 0.7766075

print(HCward$ac)

## [1] 0.9046042
```

The results say that the wards method is the best with the value of 0.904. Plotting the agnes using ward method and Cutting the Dendrogram. We will take k = 4 by observing the distance.

```
pltree(HCward, cex = 0.6, hang = -1, main = "Dendrogram of agnes-Ward")
```

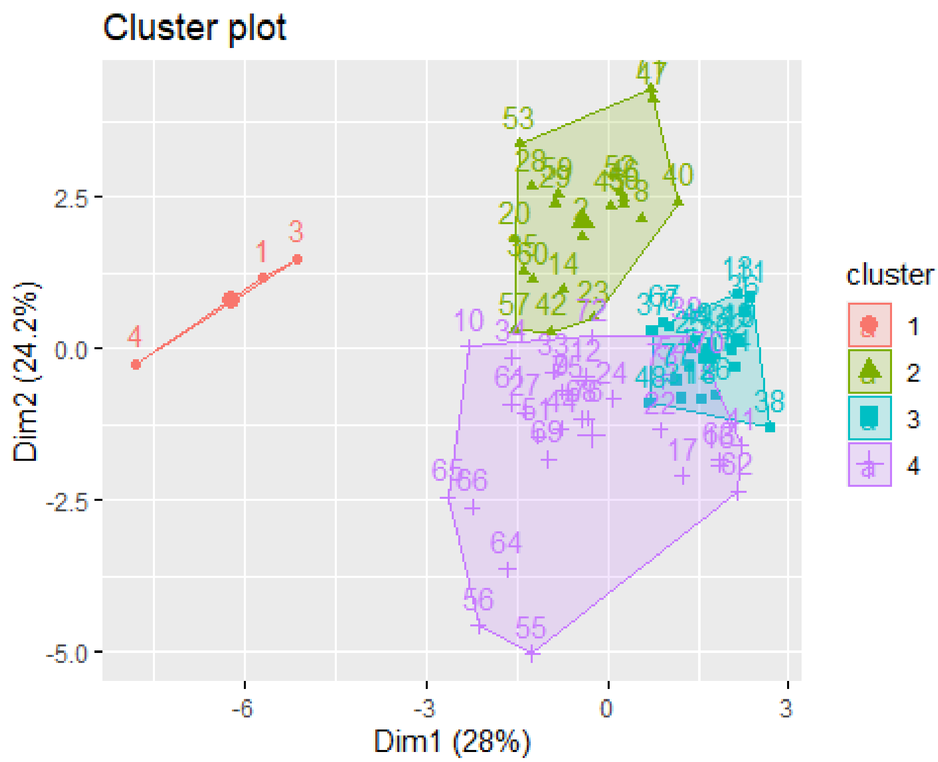


Hierarchical clustering using ward method.

```
HC1 <- hclust(d, method = "ward.D2" )
subgrp <- cutree(HC1, k = 4)
table(subgrp)

## subgrp
##  1  2  3  4
##  3 20 21 30

dataframe <- as.data.frame(cbind(Normalise,subgrp))
```

To visualiZe the results in scatter plot.

```
fviz_cluster(list(data = Normalise, cluster = subgrp))
```



To check the structure of the clusters and on their stability.

We will partition the data and apply one part to the other part

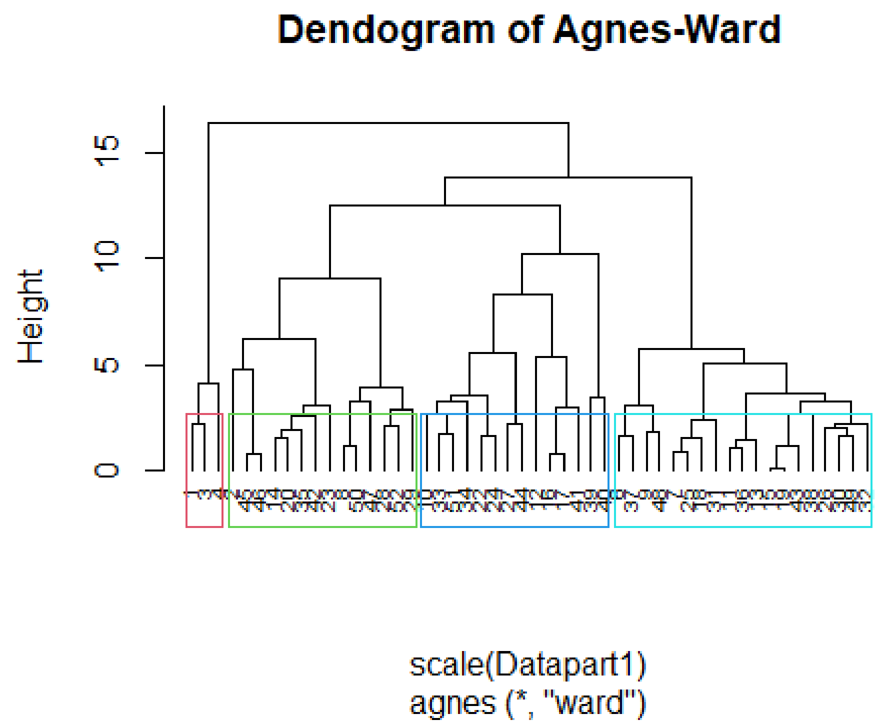
```
Datapart1 <- OmitMissing[1:50,]
Datapart2 <- OmitMissing[51:74,]
```

Performing Hierarichal Clustering using agnes() with single, complete, average and ward with partitioned data , plotting dendrogram and then cutting the dendrogram by taking k = 4.

```
Award <- agnes(scale(Datapart1), method = "ward")
Aaverage <- agnes(scale(Datapart1), method = "average")
Acomplete <- agnes(scale(Datapart1), method = "complete")
Asingle <- agnes(scale(Datapart1), method = "single")
cbind(ward=Award$ac, average=Aaverage$ac, complete=Acomplete$ac,
      single=Asingle$ac)

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

Plot dendogram for the partitioned data.

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



Using Cutree to divide into groups cluster = 4.

```
c <- cutree(Award, k = 4)
print(c)

##  1  2  3  4  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27
28
##  1  2  1  1  3  3  2  3  4  3  4  3  2  3  4  4  3  3  2  4  2  4  3  3  4
2
## 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
##  2  3  3  3  4  4  2  3  3  3  4  4  4  2  3  4  2  2  2  3  3  2  4  2
```

Calculating centers to assess the consistency of data.

```
Assess <- as.data.frame(cbind(Datapart1,c))
Assess[Assess$c==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 c
## 1 0.33 68.40297 1
```

```
## 3 0.33 59.42551 1
## 4 0.50 93.70491 1
```

```
c1 <- colMeans(Assess[Assess$c==1,])
Assess[Assess$c==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
## 8       130       3   2    210   2.0  18.0      8    100       25     3
1.33
## 14      110       3   2    140   2.0  13.0      7    105       25     3
1.00
## 20      110       3   3    140   4.0  10.0      7    160       25     3
1.00
## 23      100       2   1    140   2.0  11.0     10    120       25     3
1.00
## 28      120       3   2    160   5.0  12.0     10    200       25     3
1.25
## 29      120       3   0    240   5.0  14.0     12    190       25     3
1.33
## 35      120       3   3     75   3.0  13.0      4    100       25     3
1.00
## 42      100       4   2    150   2.0  12.0      6     95       25     2
1.00
## 45      150       4   3     95   3.0  16.0     11    170       25     3
1.00
## 46      150       4   3    150   3.0  16.0     11    170       25     3
1.00
## 47      160       3   2    150   3.0  17.0     13    160       25     3
1.50
## 50      140       3   2    220   3.0  21.0      7    130       25     3
1.33
## 52      130       3   2    170   1.5  13.5     10    120       25     3
1.25
##    cups   rating c
## 2  1.00 33.98368 2
## 8  0.75 37.03856 2
## 14 0.50 40.40021 2
## 20 0.50 40.44877 2
## 23 0.75 36.17620 2
## 28 0.67 40.91705 2
## 29 0.67 41.01549 2
## 35 0.33 45.81172 2
## 42 0.67 45.32807 2
## 45 1.00 37.13686 2
## 46 1.00 34.13976 2
## 47 0.67 30.31335 2
```

```
## 50 0.67 40.69232 2
## 52 0.50 30.45084 2
```

```
c2 <- colMeans(Assess[Assess$c==2,])
Assess[Assess$c==3,]
```

```
##    calories protein fat sodium fiber carbo sugars potass vitamins shelf
weight
## 6       110       2   2    180   1.5  10.5     10     70       25     1
1
## 7       110       2   0    125   1.0  11.0     14     30       25     2
1
## 9        90       2   1    200   4.0  15.0      6    125       25     1
1
## 11      120       1   2    220   0.0  12.0     12     35       25     2
1
## 13      120       1   3    210   0.0  13.0      9     45       25     2
1
## 15      110       1   1    180   0.0  12.0     13     55       25     2
1
## 18      110       1   0     90   1.0  13.0     12     20       25     2
1
## 19      110       1   1    180   0.0  12.0     13     65       25     2
1
## 25      110       2   1    125   1.0  11.0     13     30       25     2
1
## 26      110       1   0    200   1.0  14.0     11     25       25     1
1
## 30      110       1   1    135   0.0  13.0     12     25       25     2
1
## 31      100       2   0     45   0.0  11.0     15     40       25     1
1
## 32      110       1   1    280   0.0  15.0      9     45       25     2
1
## 36      120       1   2    220   1.0  12.0     11     45       25     2
1
## 37      110       3   1    250   1.5  11.5     10     90       25     1
1
## 38      110       1   0    180   0.0  14.0     11     35       25     1
1
## 43      110       2   1    180   0.0  12.0     12     55       25     2
1
## 48      100       2   1    220   2.0  15.0      6     90       25     1
1
## 49      120       2   1    190   0.0  15.0      9     40       25     2
1
##    cups   rating c
## 6  0.75 29.50954 3
## 7  1.00 33.17409 3
## 9  0.67 49.12025 3
```

```
## 11 0.75 18.04285 3
## 13 0.75 19.82357 3
## 15 1.00 22.73645 3
## 18 1.00 35.78279 3
## 19 1.00 22.39651 3
## 25 1.00 32.20758 3
## 26 0.75 31.43597 3
## 30 0.75 28.02576 3
## 31 0.88 35.25244 3
## 32 0.75 23.80404 3
## 36 1.00 21.87129 3
## 37 0.75 31.07222 3
## 38 1.33 28.74241 3
## 43 1.00 26.73451 3
## 48 1.00 40.10596 3
## 49 0.67 29.92429 3
```

```
c3 <- colMeans(Assess[Assess$c==3,])
Assess[Assess$c==4,]
```

```
##    calories protein fat sodium fiber carbo sugars potass vitamins shelf
weight
## 10       90       3   0    210     5    13      5    190       25     3
1.0
## 12      110       6   2    290     2    17      1    105       25     1
1.0
## 16      110       2   0    280     0    22      3     25       25     1
1.0
## 17      100       2   0    290     1    21      2     35       25     1
1.0
## 22      110       2   0    220     1    21      3     30       25     3
1.0
## 24      100       2   0    190     1    18      5     80       25     3
1.0
## 27      100       3   0      0     3    14      7    100       25     2
1.0
## 33      100       3   1    140     3    15      5     85       25     3
1.0
## 34      110       3   0    170     3    17      3     90       25     3
1.0
## 39      110       2   1    170     1    17      6     60      100     3
1.0
## 40      140       3   1    170     2    20      9     95      100     3
1.3
## 41      110       2   1    260     0    21      3     40       25     2
1.0
## 44      100       4   1      0     0    16      3     95       25     2
1.0
## 51       90       3   0    170     3    18      2     90       25     3
1.0
```

```
##    cups   rating c
## 10 0.67 53.31381 4
## 12 1.25 50.76500 4
## 16 1.00 41.44502 4
## 17 1.00 45.86332 4
## 22 1.00 46.89564 4
## 24 0.75 44.33086 4
## 27 0.80 58.34514 4
## 33 0.88 52.07690 4
## 34 0.25 53.37101 4
## 39 1.00 36.52368 4
## 40 0.75 36.47151 4
## 41 1.50 39.24111 4
## 44 1.00 54.85092 4
## 51 1.00 59.64284 4

c4 <- colMeans(Assess[Assess$c==4,])
```

Binding the 4 centers.

```
centers <- rbind(c1,c2,c3,c4)
centers

##     calories  protein       fat   sodium      fiber     carbo    sugars
## c1  63.33333 4.000000 0.6666667 176.6667 11.0000000  6.666667  3.666667
## c2 125.71429 3.142857 2.2857143 146.7857  2.8928571 13.892857  8.857143
## c3 110.00000 1.526316 1.0000000 179.4737  0.7368421 12.736842 10.947368
## c4 105.71429 2.857143 0.5000000 182.8571  1.7857143 17.857143  4.071429
##        potass vitamins    shelf   weight      cups   rating c
## c1 310.00000 25.00000 3.000000 1.000000 0.3866667 73.84446 1
## c2 139.64286 23.21429 2.928571 1.142143 0.6914286 38.13235 2
## c3  50.78947 25.00000 1.631579 1.000000 0.8842105 29.46119 3
## c4  80.00000 35.71429 2.357143 1.021429 0.9178571 48.08120 4
```

Calculating Distance and comparing the record in B with the closest centroid in A

```
d1 <- as.data.frame(rbind(centers[,-14],Datapart2))
d2 <- get_dist(d1)
matrix <- as.matrix(d2)
df1 <-
data.frame(data=seq(1,nrow(Datapart2),1),clusters=rep(0,nrow(Datapart2)))
for(i in 1:nrow(Datapart2)) {

  df1[i,2] <- which.min(matrix[i+4, 1:4])
}
df1

##    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(dataframe$subgrp[51:74], df1$clusters)
```

```
##       [,1] [,2]
##  [1,]    2    1
##  [2,]    4    4
##  [3,]    4    3
##  [4,]    4    2
##  [5,]    2    2
##  [6,]    2    1
##  [7,]    2    2
##  [8,]    4    2
##  [9,]    4    3
## [10,]    4    3
## [11,]    4    2
## [12,]    4    2
## [13,]    4    2
## [14,]    3    3
## [15,]    4    4
## [16,]    4    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(dataframe$subgrp[51:74] == df1$clusters)

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

From above Results , 12 are True and 12 are False, so we can say the model may be stable.

Selecting the cluster that is best cereal for breakfast, which will have high protein, fiber and low in sugar, sodium.

Choosing the Cluster of Healthy Cereals.

```
newdata <- cereals
newdata_omit <- na.omit(newdata)
Clust <- cbind(newdata_omit, subgrp)
Clust[Clust$subgrp==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 subgrp
## 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

Clust[Clust$subgrp==2,]

##                                       name mfr type calories protein fat
sodium
## 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
## 23                   Crispy_Wheat_&_Raisins   G    C      100       2   1
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 subgrp
## 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
## 23   2.0  11.0     10    120       25     3   1.00 0.75 36.17620      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
```

```
Clust[Clust$subgrp==3,]
```

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

```
## 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
## 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
## 48    Multi-Grain_Cheerios   G    C      100       2   1    220   2.0
15.0
## 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 subgrp
## 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
```

```
## 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
## 48       6      90        25      1       1 1.00 40.10596       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
```

```
Clust[Clust$subgrp==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
## 27          Frosted_Mini-Wheats   K    C      100       3   0      0     3
14
## 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
## 44                        Maypo   A    H      100       4   1      0     0
16
## 51            Nutri-grain_Wheat   K    C       90       3   0    170     3
18
## 54                   Product_19   K    C      100       3   0    320     1
20
## 55                  Puffed_Rice   Q    C       50       1   0      0     0
```

```
13
## 56                 Puffed_Wheat   Q    C       50       2   0       0      1
10
## 61               Raisin_Squares   K    C       90       2   0       0      2
15
## 62                    Rice_Chex   R    C      110       1   0     240      0
23
## 63                 Rice_Krispies   K    C      110       2   0     290      0
22
## 64               Shredded_Wheat   N    C       80       2   0       0      3
16
## 65       Shredded_Wheat_'n'Bran   N    C       90       3   0       0      4
19
## 66   Shredded_Wheat_spoon_size   N    C       90       3   0       0      3
20
## 68                    Special_K   K    C      110       6   0     230      1
16
## 69     Strawberry_Fruit_Wheats   N    C       90       2   0      15      3
15
## 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 subgrp
## 9       6    125       25     1   1.00 0.67 49.12025      4
## 10      5    190       25     3   1.00 0.67 53.31381      4
## 12      1    105       25     1   1.00 1.25 50.76500      4
## 16      3     25       25     1   1.00 1.00 41.44502      4
## 17      2     35       25     1   1.00 1.00 45.86332      4
## 22      3     30       25     3   1.00 1.00 46.89564      4
## 24      5     80       25     3   1.00 0.75 44.33086      4
## 27      7    100       25     2   1.00 0.80 58.34514      4
## 33      5     85       25     3   1.00 0.88 52.07690      4
## 34      3     90       25     3   1.00 0.25 53.37101      4
## 39      6     60      100     3   1.00 1.00 36.52368      4
## 41      3     40       25     2   1.00 1.50 39.24111      4
## 44      3     95       25     2   1.00 1.00 54.85092      4
## 51      2     90       25     3   1.00 1.00 59.64284      4
## 54      3     45      100     3   1.00 1.00 41.50354      4
## 55      0     15        0     3   0.50 1.00 60.75611      4
## 56      0     50        0     3   0.50 1.00 63.00565      4
## 61      6    110       25     3   1.00 0.50 55.33314      4
## 62      2     30       25     1   1.00 1.13 41.99893      4
## 63      3     35       25     1   1.00 1.00 40.56016      4
```

```
## 64       0     95         0      1   0.83 1.00 68.23588       4
## 65       0    140         0      1   1.00 0.67 74.47295       4
## 66       0    120         0      1   1.00 0.67 72.80179       4
## 68       3     55        25      1   1.00 1.00 53.13132       4
## 69       5     90        25      2   1.00 1.00 59.36399       4
## 70       3     35       100      3   1.00 1.00 38.83975       4
## 72       3    110       100      3   1.00 1.00 46.65884       4
## 73       3     60        25      3   1.00 0.75 39.10617       4
## 75       3    115        25      1   1.00 0.67 49.78744       4
## 76       3    110        25      1   1.00 1.00 51.59219       4
```

Calculating mean ratings to determine the best cluster.

```
mean(Clust[Clust$subgrp==1,"rating"])

## [1] 73.84446

mean(Clust[Clust$subgrp==2,"rating"])

## [1] 38.26161

mean(Clust[Clust$subgrp==3,"rating"])

## [1] 28.84825

mean(Clust[Clust$subgrp==4,"rating"])

## [1] 51.43111
```

As we can see that the mean ratings for the subgrp==1 is the highest(73.84), it's the best option to choose cluster 1 and the cereals in the cluster 1 for healthy diet.