

# Assignment - 5

Nikitha Chigurupati

11/30/2022

```
#Loading the Packages  
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.1.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##  
## 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
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

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

```
library(dendextend)
```

```
## Warning: package 'dendextend' was built under R version 4.1.3
```

```
##
## -----
## 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/issu
## es
## 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)
```

```
## Warning: package 'knitr' was built under R version 4.1.3
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.1.3
```

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

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
#Importing the dataset
Cereals <- read.csv("C:/Users/Nikitha/Downloads/Cereals.csv")
head(Cereals)
```

```
##           name mfr type calories protein fat sodium fiber carbo
## 1      100%_Bran   N   C       70        4  1   130  10.0   5.0
## 2    100%_Natural_Bran   Q   C      120        3  5    15   2.0   8.0
## 3          All-Bran   K   C       70        4  1   260   9.0   7.0
## 4 All-Bran_with_Extra_Fiber   K   C       50        4  0   140  14.0   8.0
## 5          Almond_Delight   R   C      110        2  2   200   1.0  14.0
## 6  Apple_Cinnamon_Cheerios   G   C      110        2  2   180   1.5  10.5
##  sugars potass vitamins shelf weight cups  rating
## 1      6      280       25     3      1 0.33 68.40297
## 2      8      135        0     3      1 1.00 33.98368
## 3      5      320       25     3      1 0.33 59.42551
## 4      0      330       25     3      1 0.50 93.70491
## 5      8       NA       25     3      1 0.75 34.38484
## 6     10       70       25     1      1 0.75 29.50954
```

```
dim(Cereals)
```

```
## [1] 77 16
```

```
#Omitting the NULL values
Cereals <- na.omit(Cereals)
dim(Cereals)
```

```
## [1] 74 16
```

```
head(Cereals)
```

```
##           name mfr type calories protein fat sodium fiber carbo
## 1      100%_Bran   N   C       70        4  1   130  10.0   5.0
## 2    100%_Natural_Bran   Q   C      120        3  5    15   2.0   8.0
## 3          All-Bran   K   C       70        4  1   260   9.0   7.0
## 4 All-Bran_with_Extra_Fiber   K   C       50        4  0   140  14.0   8.0
## 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
##  sugars potass vitamins shelf weight cups  rating
## 1      6      280       25     3      1 0.33 68.40297
## 2      8      135        0     3      1 1.00 33.98368
## 3      5      320       25     3      1 0.33 59.42551
## 4      0      330       25     3      1 0.50 93.70491
## 6     10       70       25     1      1 0.75 29.50954
## 7     14       30       25     2      1 1.00 33.17409
```

```
#Creating a dataset with the Numeric Values
Cereals_Num<- data.frame(Cereals[,4:16])
Cereals_Num <- na.omit(Cereals_Num)
```

```
#Normalizing the data
```

```
Cereals_norm <- scale(Cereals_Num)
```

```
head(Cereals_norm)
```

```
##      calories  protein      fat    sodium      fiber      carbo      sugars
## 1 -1.8659155  1.3817478  0.0000000 -0.3910227  3.22866747 -2.5001396 -0.2542051
## 2  0.6537514  0.4522084  3.9728810 -1.7804186 -0.07249167 -1.7292632  0.2046041
## 3 -1.8659155  1.3817478  0.0000000  1.1795987  2.81602258 -1.9862220 -0.4836096
## 4 -2.8737823  1.3817478 -0.9932203 -0.2702057  4.87924705 -1.7292632 -1.6306324
## 6  0.1498180 -0.4773310  0.9932203  0.2130625 -0.27881412 -1.0868662  0.6634132
## 7  0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656 -0.9583868  1.5810314
##      potass  vitamins      shelf    weight      cups      rating
## 1  2.5605229 -0.1818422  0.9419715 -0.2008324 -2.0856582  1.8549038
## 2  0.5147738 -1.3032024  0.9419715 -0.2008324  0.7567534 -0.5977113
## 3  3.1248675 -0.1818422  0.9419715 -0.2008324 -2.0856582  1.2151965
## 4  3.2659536 -0.1818422  0.9419715 -0.2008324 -1.3644493  3.6578436
## 6 -0.4022862 -0.1818422 -1.4616799 -0.2008324 -0.3038480 -0.9165248
## 7 -0.9666308 -0.1818422 -0.2598542 -0.2008324  0.7567534 -0.6553998
```

```
#Applying hierarchical clustering using Euclidean distance method.
```

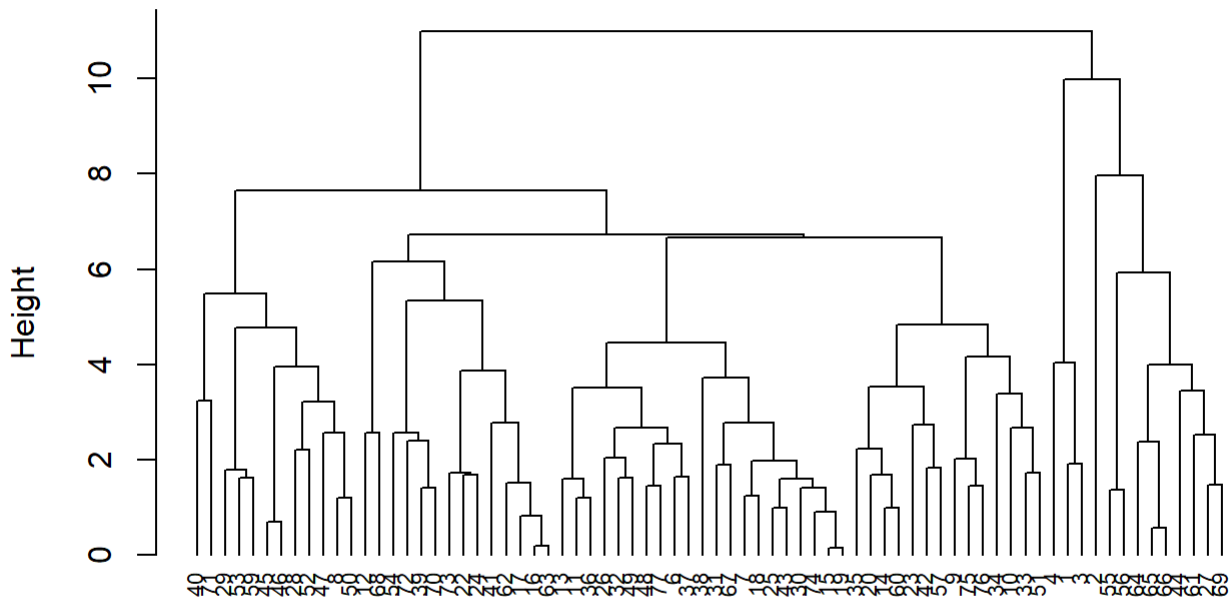
```
dist <- dist(Cereals_norm, method= "euclidean")
```

```
Hist_clust <- hclust(dist, method = "complete")
```

```
#Plotting of the dendrogram
```

```
plot(Hist_clust, cex = 0.7, hang = -1)
```

## Cluster Dendrogram



```
#Using Agnes function to perform clustering with single linkage, complete linkage average linkage and Ward.
```

```
hc_single <- agnes(Cereals_norm, method = "single")  
hc_complete <- agnes(Cereals_norm, method = "complete")  
hc_average <- agnes(Cereals_norm, method = "average")  
hc_ward <- agnes(Cereals_norm, method = "ward")
```

```
#Determining the best method  
print(hc_single$ac)
```

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

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

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

```
print(hc_average$ac)
```

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

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

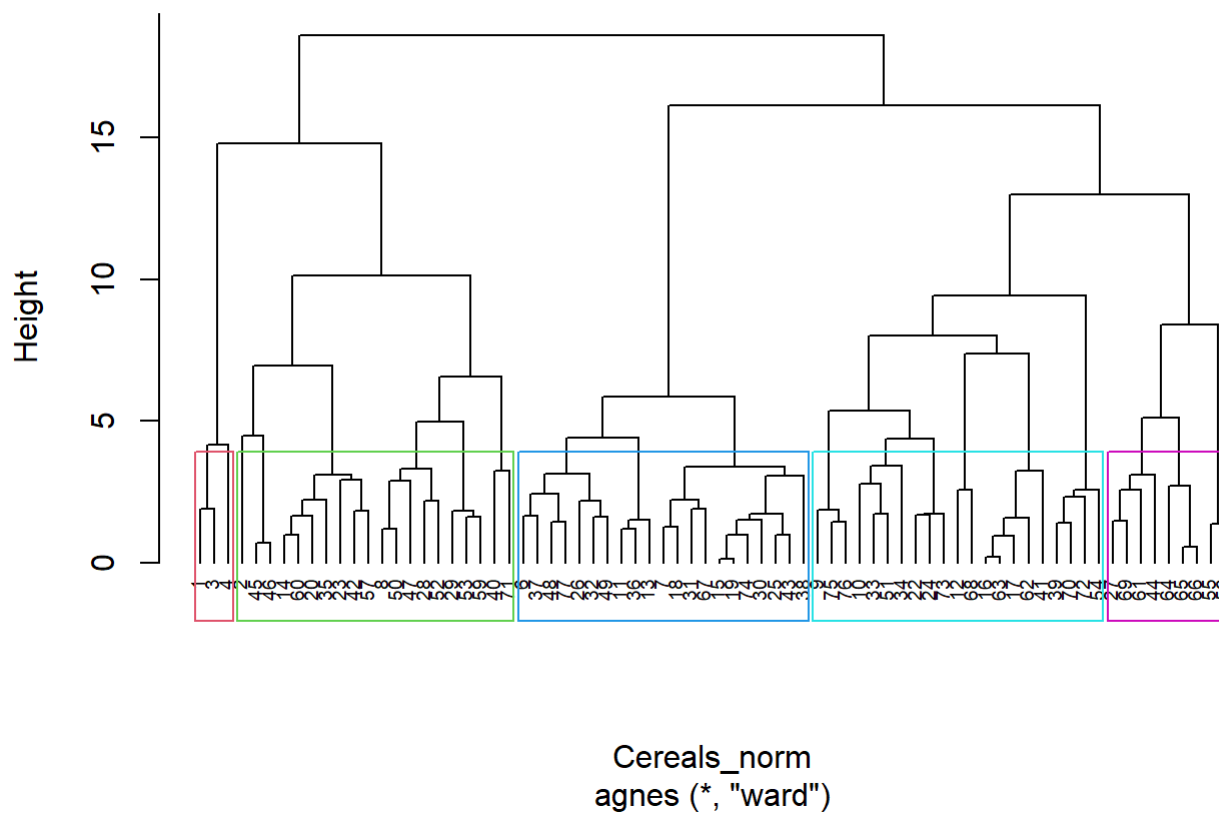
```
## [1] 0.9046042
```

*#The ward method is the best as compared to the other methods with a value of 0.9046042*

*#Choosing the number of clusters*

```
pltree(hc_ward, cex = 0.6, hang = -1, main = "Dendrogram of agnes")  
rect.hclust(hc_ward, k=5, border = 2:7)
```

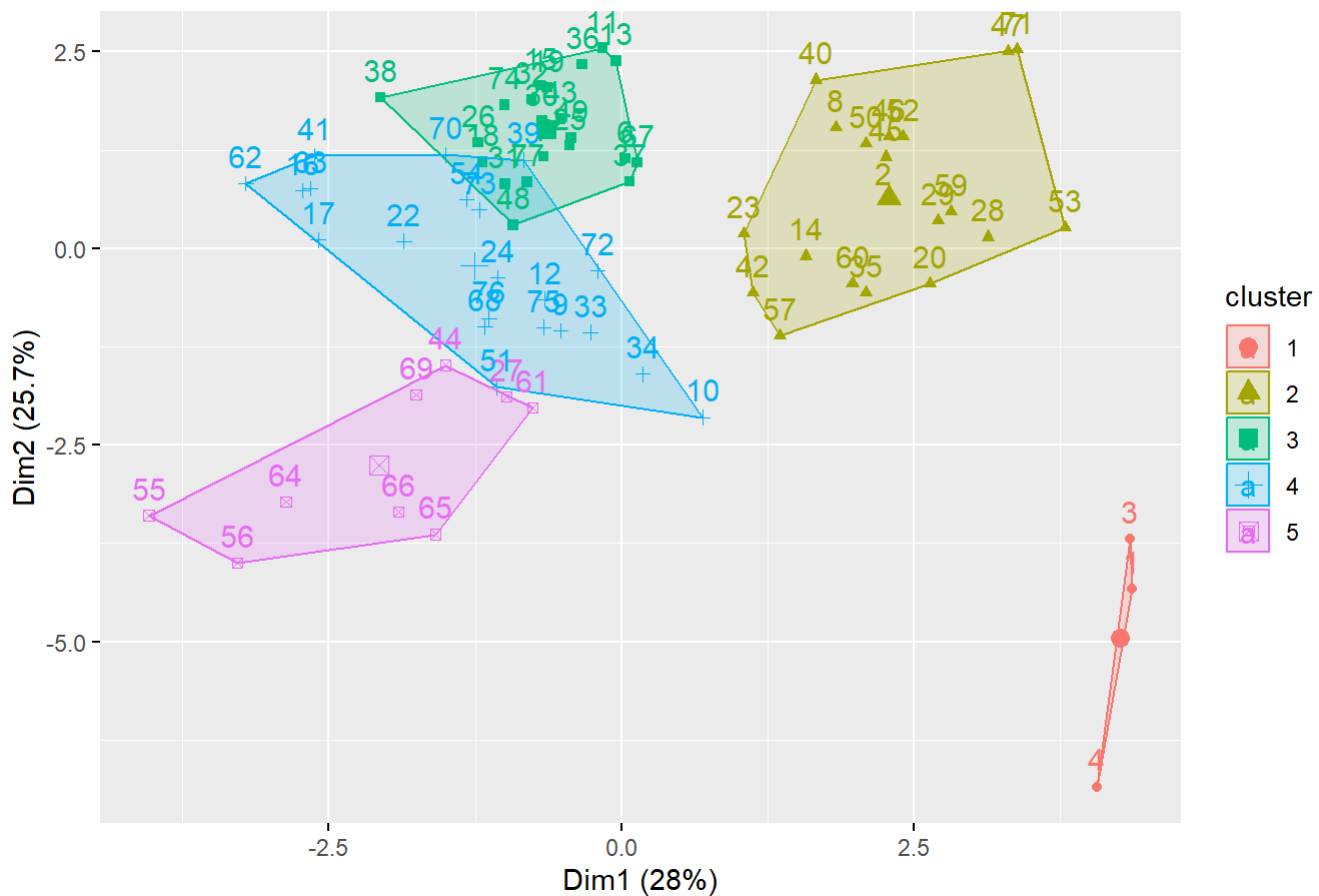
## Dendrogram of agnes



```
subGroup <- cutree(hc_ward, k=5)
```

```
df <- as.data.frame(cbind(Cereals_norm, subGroup))  
fviz_cluster(list(data=df, cluster = subGroup))
```

Cluster plot



*#It is concluded that 5 clusters can be selected.*

*#Creating Partitions*

```
set.seed(123)
partition1 <- Cereals_Num [1:50,]
partition2 <- Cereals_Num [51:74,]
```

*#Performing Hierarchical Clustering, considering k = 5.*

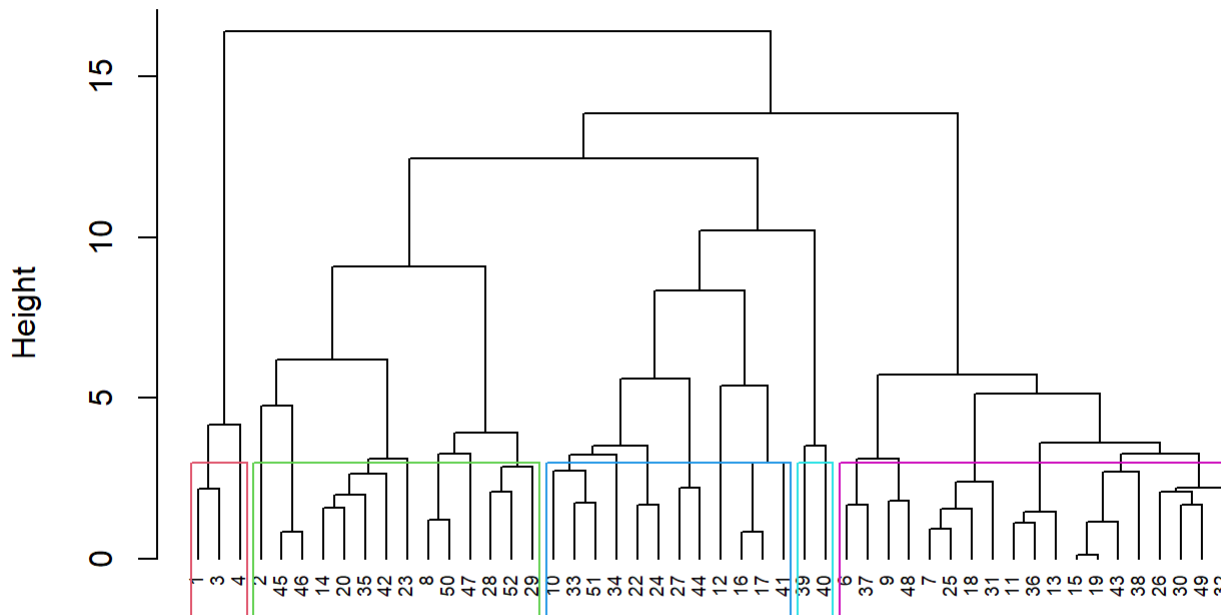
```
Ag_single <- agnes(scale(partition1), method = "single")
Ag_complete <- agnes(scale(partition1), method = "complete")
Ag_average <- agnes(scale(partition1), method = "average")
Ag_ward <- agnes(scale(partition1), method = "ward")

cbind(single= Ag_single$ac , complete=Ag_complete$ac , average= Ag_average$ac , ward= Ag_ward$ac)
```

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

```
pltree(Ag_ward, cex = 0.6, hang = -1, main = "Dendrogram of Agnes Using Ward")
rect.hclust(Ag_ward, k = 5, border = 2:7)
```

## Dendrogram of Agnes Using Ward



```
scale(partition1)
agnes (*, "ward")
```

```
cut2 <- cutree(Ag_ward, k = 5)
```

*#Calculating the centroids.*

```
Result <- as.data.frame(cbind(partition1, cut2))
Result[Result$cut2==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 cut2
## 1 0.33 68.40297    1
## 3 0.33 59.42551    1
## 4 0.50 93.70491    1
```

```
Centroid1 <- colMeans(Result[Result$cut2==1,])
Result[Result$cut2==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 cut2
## 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
```

```
Centroid2 <- colMeans(Result[Result$cut2==2,])
Result[Result$cut2==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 cut2
## 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
```

```
Centroid3 <- colMeans(Result[Result$cut2==3,])
Result[Result$cut2==4,]
```

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

##      cups   rating cut2
## 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
## 41 1.50 39.24111    4
## 44 1.00 54.85092    4
## 51 1.00 59.64284    4
```

```
Centroid4 <- colMeans(Result[Result$cut2==4,])
```

```
Centroids <- rbind(Centroid1, Centroid2, Centroid3, Centroid4)
```

```
x2 <- as.data.frame(rbind(Centroids[, -14], partition2))
```

```
#Calculating the Distance.
```

```
Dist1 <- get_dist(x2)
```

```
Matrix <- as.matrix(Dist1)
```

```
data.frame <- data.frame(data=seq(1,nrow(partition2),1), Clusters = rep(0,nrow(partition2)))
```

```
for(i in 1:nrow(partition2))
```

```
{data.frame[i,2] <- which.min(Matrix[i+4, 1:4])}
```

```
data.frame
```

```
##      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(df$SubGroup[51:74], data.frame$Clusters)
```

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

```
table(df$SubGroup[51:74] == data.frame$Clusters)
```

```
## < table of extent 0 >
```

```
#We can conclude that it is partially stable.
```

```
#Clustering Healthy Cereals.
Healthy_Cereals <- Cereals
Healthy_Cereals_na <- na.omit(Healthy_Cereals)
Clusthealthy <- cbind(Healthy_Cereals_na, subGroup)

Clusthealthy[Clusthealthy$subGroup==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 subGroup
## 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
```

```
Clusthealthy[Clusthealthy$subGroup==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	subGroup
## 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

```
Clusthealthy[Clusthealthy$subGroup==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	subGroup		
## 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		

```
Clusthealthy[Clusthealthy$subGroup==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
## 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 subGroup
## 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
## 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(Clusthealthy[Clusthealthy$subGroup==1,"rating"])
```

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



```
mean(Clusthealthy[Clusthealthy$subGroup==2,"rating"])
```

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

```
mean(Clusthealthy[Clusthealthy$subGroup==3,"rating"])
```

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

```
mean(Clusthealthy[Clusthealthy$subGroup==4,"rating"])
```

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

*#It can be concluded that cluster 1 can be chosen as it has the highest value. Hence, cluster 1 can be considered as a Healthy Cluster.*