Assignment5

Anujeeth Veerla

28/11/2021

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 (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:

```
library(cluster)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(factoextra)
```

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

```
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/i
ssues
## 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)
```

```
Cereals_df<-read.csv("Cereals.csv")
#Removing all cereal missing values from the CSV
sum(is.na(Cereals_df))
```

```
## [1] 4
```

```
C<-na.omit(Cereals_df) #C has 74 obs where as initial file has 77.
sum(is.na(C))
```

```
## [1] 0
```

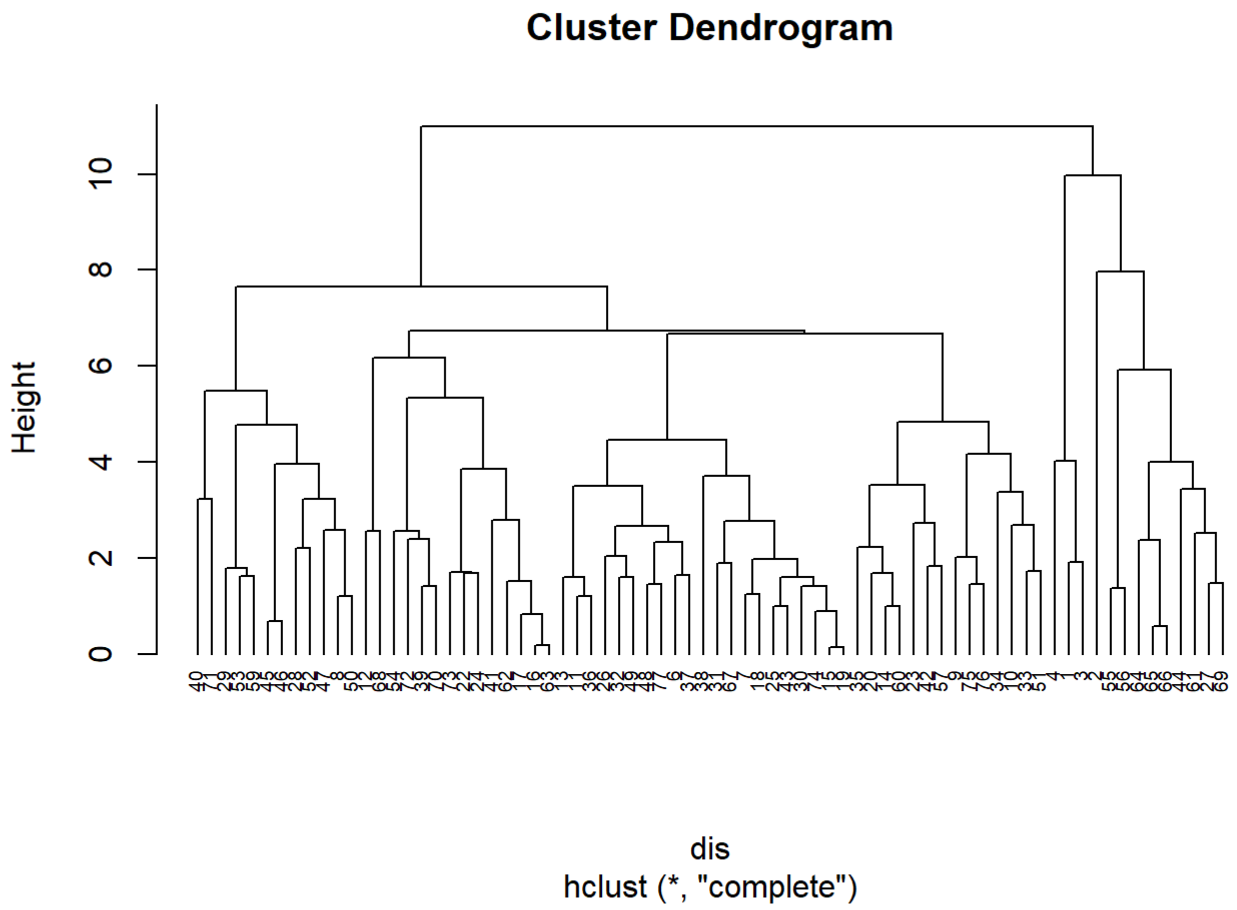
```
#removing categorical variables before scaling
C1<-C[,c(-1,-2,-3)]
head(C1)
```

```
##   calories protein fat sodium fiber carbo sugars potass vitamins shelf weight
## 1       70       4   1    130  10.0   5.0      6    280       25     3      1
## 2      120       3   5     15   2.0   8.0      8    135        0     3      1
## 3       70       4   1    260   9.0   7.0      5    320       25     3      1
## 4       50       4   0    140  14.0   8.0      0    330       25     3      1
## 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
##   cups   rating
## 1 0.33 68.40297
## 2 1.00 33.98368
## 3 0.33 59.42551
## 4 0.50 93.70491
## 6 0.75 29.50954
## 7 1.00 33.17409
```

```
#Scaling the data
Cs<-scale(C1)
```

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.

```
#Calculating dissimilarity matrix
dis<-dist(Cs, method = "euclidean")
#Performing hierarchical clustering using complete linkage
hc_complete<-hclust(dis, method = "complete")
plot(hc_complete, cex=0.6, hang=-1)
```



```
#Performing clustering using Agnes
c.single<-agnes(Cs, method = "single")
c.complete<-agnes(Cs, method = "complete")
c.average<-agnes(Cs,method = "average")
c.ward<-agnes(Cs, method = "ward")
#Now, we can compare agglomerative coefficients of single,complete,average and ward methods.
print(c.single$ac)
```

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

```
print(c.complete$ac)
```

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

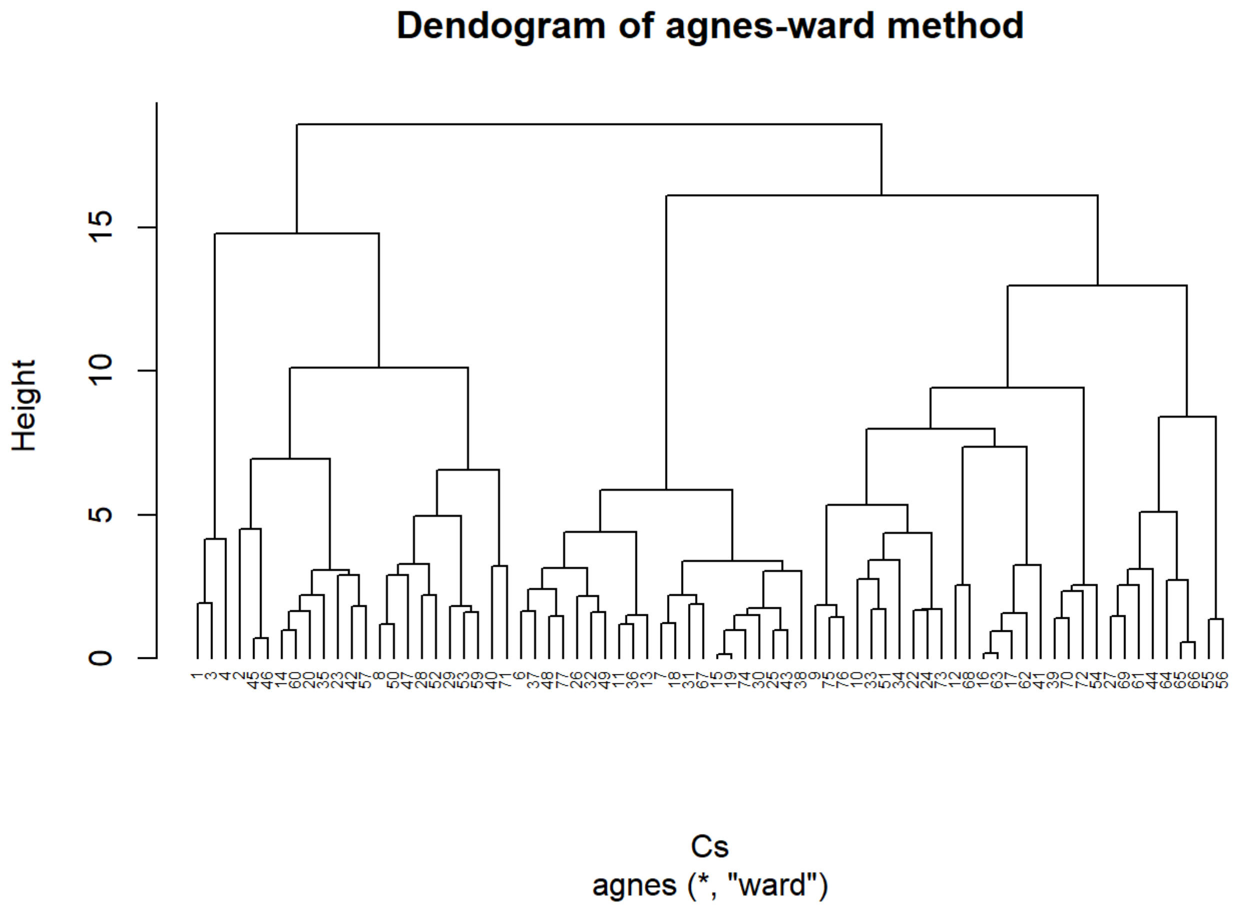
```
print(c.average$ac)
```

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

```
print(c.ward$ac)
```

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

```
#Plotting the dendogram using wards method
pl<-pltree(c.ward,cex=0.5,hang=-1,main = "Dendogram of agnes-ward method")
```



Q. How many clusters would you choose? #Distance matrix

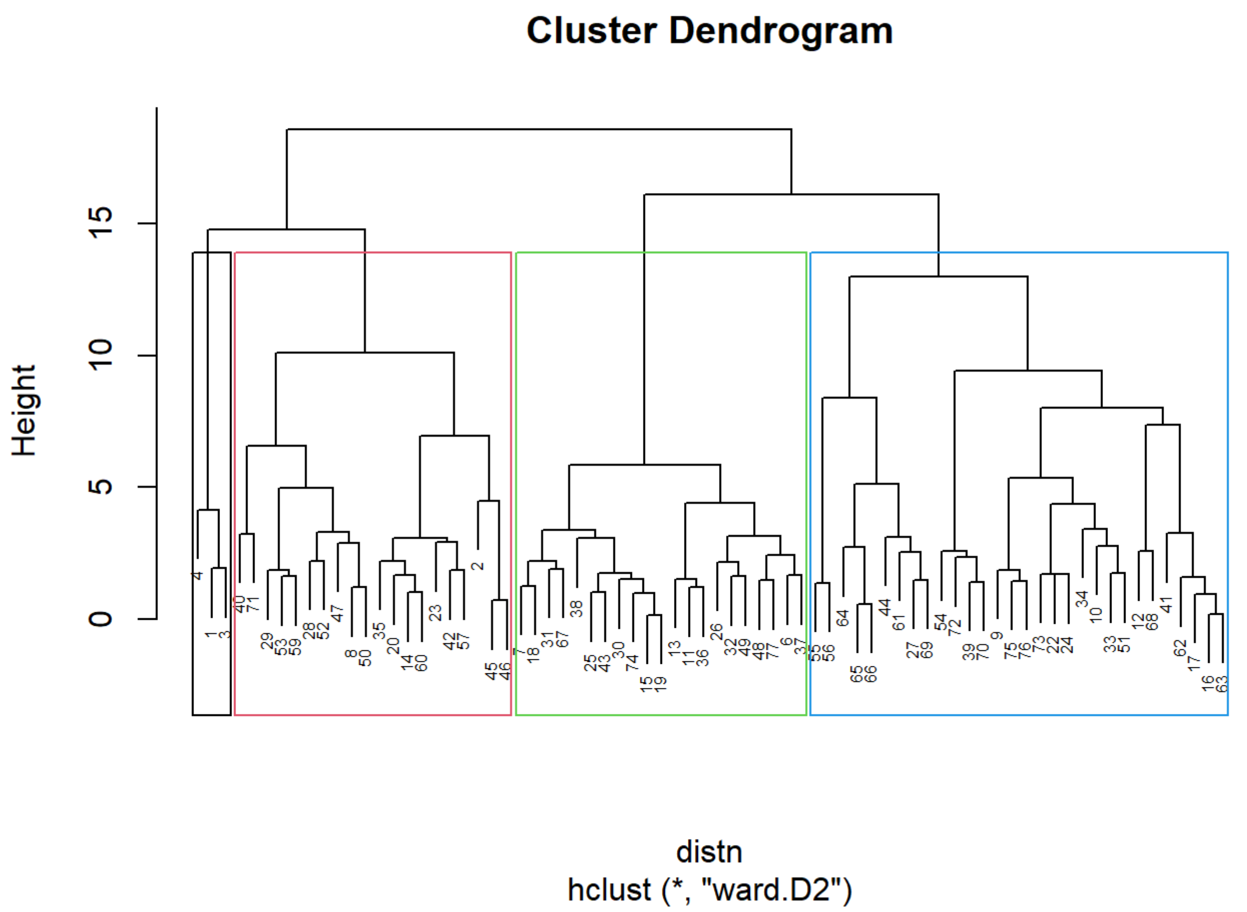
```
distn<-dist(Cs, method = "euclidean")
```

#Wards method for hierarchial cluster

```
w_hc<-hclust(distn,method = "ward.D2")
```

plotting the dendogram and value of k is taken as 4 by observing the tree

```
plot(w_hc,cex=0.5)
rect.hclust(w_hc,k=4,border=1:4)
```



Identifying clusters

```
clust<-cutree(w_hc,k=4)
```

total no. of members in each cluster

```
table(clust)
```

```
## clust
##  1  2  3  4 
##  3 20 21 30
```

Q. Comment on the structure of the clusters and on their stability. Hint: To check stability, partition the data and see how well clusters formed based on one part apply to the other part.

```
set.seed(123)
Cereals_new<-Cereals_df
```

removing missing values from the csv dataset

```
n<-na.omit(Cereals_new)
n1<-n[,c(-1,-2,-3)]
n2<-scale(n1)
n3<-as.data.frame(n2)
```

Creating partitions

```
par1<-n[1:55,]
par2<-n[56:74,]
```

Clustering using agnes with partitioned data

```
r1<-agnes(scale(par1[,-c(1:3)]),method = "ward")
r2<-agnes(scale(par1[,-c(1:3)]),method = "average")
r3<-agnes(scale(par1[,-c(1:3)]),method = "complete")
r4<-agnes(scale(par1[,-c(1:3)]),method = "single")
cbind(ward=r1$ac,average=r2$ac,complete=r3$ac,single=r4$ac)
```

```
##           ward   average  complete    single
## [1,] 0.8808195 0.7449303 0.8120228 0.6564842
```

```
c2<-cutree(r1,k=4)
```

Calculating the centers

```
ce<-as.data.frame(cbind(scale(par1[,-c(1:3)]),c2))
ce1<-colMeans(ce[ce$c2==1,])
ce2<-colMeans(ce[ce$c2==2,])
ce3<-colMeans(ce[ce$c2==3,])
ce4<-colMeans(ce[ce$c2==4,])
```

Binding 4 centers

```
centers<-rbind(ce1, ce2, ce3, ce4)
centers
```

```
##       calories    protein        fat      sodium      fiber       carbo
## ce1 -2.0643907  1.3722190 -0.4227336  0.12077927  3.2987660 -1.96959911
## ce2  0.7279288  0.6244015  0.9156909 -0.23703924  0.3023157 -0.09643751
## ce3  0.0924354 -0.9045372 -0.1168080  0.15758181 -0.5604394 -0.35725011
## ce4 -0.4241154  0.1811250 -0.6566767  0.02566092 -0.2402941  0.83762056
##         sugars     potass   vitamins      shelf     weight       cups
## ce1 -0.9815196  2.8376723 -0.1479841  0.8445408 -0.1860883 -1.7018175
## ce2  0.3192895  0.6218875 -0.2327666  0.7638988  0.7299944 -0.5356524
## ce3  0.7942560 -0.6498707 -0.1479841 -0.9210927 -0.1860883  0.2889243
## ce4 -1.0149963 -0.3597455  0.4105833  0.1614563 -0.4462334  0.4815488
##         rating c2
## ce1  2.4007287  1
## ce2 -0.1391348  2
## ce3 -0.8182967  3
## ce4  0.6218593  4
```

#Calculating distance

```
a<-as.data.frame(rbind(centers[,-14],scale(par2[,-c(1:3)])))
a1<-get_dist(a)
a2<-as.matrix(a1)
d1<-data.frame(data=seq(1,nrow(par2),1),clusters=rep(0,nrow(par2)))
for (i in 1:nrow(par2))
  {
  d1[i,2]<-which.min(a2[i+4,1:4])
}
d1
```

```
##    data clusters
## 1     1        2
## 2     2        2
## 3     3        4
## 4     4        4
## 5     5        4
## 6     6        4
## 7     7        4
## 8     8        4
## 9     9        3
## 10   10        4
## 11   11        4
## 12   12        4
## 13   13        2
## 14   14        4
## 15   15        4
## 16   16        3
## 17   17        2
## 18   18        4
## 19   19        3
```

```
a3<-as.data.frame(cbind(Cs,clust))
cbind(a3$clust[56:74],d1$clusters)
```

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

```
table(a3$clust[56:74]==d1$clusters)
```

```
## 
## FALSE  TRUE 
##     1    18
```

As we have more no of trues, model is stable

Q)find a cluster of “healthy cereals.”

```
Cr<-cbind(n3,clust)
Cr[Cr$clust==1,]
```

```
##    calories  protein        fat     sodium    fiber     carbo     sugars
## 1 -1.865915 1.381748  0.0000000 -0.3910227 3.228667 -2.500140 -0.2542051
## 3 -1.865915 1.381748  0.0000000  1.1795987 2.816023 -1.986222 -0.4836096
## 4 -2.873782 1.381748 -0.9932203 -0.2702057 4.879247 -1.729263 -1.6306324
##     potass   vitamins     shelf     weight      cups   rating clust
## 1 2.560523 -0.1818422 0.9419715 -0.2008324 -2.085658 1.854904     1
## 3 3.124867 -0.1818422 0.9419715 -0.2008324 -2.085658 1.215196     1
## 4 3.265954 -0.1818422 0.9419715 -0.2008324 -1.364449 3.657844     1
```

```
Cr[Cr$clust==2,]
```

```
##      calories    protein        fat      sodium       fiber       carbo
## 2   0.6537514  0.4522084  3.9728810 -1.78041856 -0.07249167 -1.72926320
## 8   1.1576848  0.4522084  0.9932203  0.57551356 -0.07249167  0.84032469
## 14  0.1498180  0.4522084  0.9932203 -0.27020566 -0.07249167 -0.44446926
## 20  0.1498180  0.4522084  1.9864405 -0.27020566  0.75279812 -1.21534562
## 23 -0.3541153 -0.4773310  0.0000000 -0.27020566 -0.07249167 -0.95838683
## 28  0.6537514  0.4522084  0.9932203 -0.02857160  1.16544301 -0.70142805
## 29  0.6537514  0.4522084 -0.9932203  0.93796466  1.16544301 -0.18751047
## 35  0.6537514  0.4522084  1.9864405 -1.05551637  0.34015322 -0.44446926
## 40  1.6616182  0.4522084  0.0000000  0.09224544 -0.07249167  1.35424227
## 42 -0.3541153  1.3817478  0.9932203 -0.14938863 -0.07249167 -0.70142805
## 45  2.1655516  1.3817478  1.9864405 -0.81388230  0.34015322  0.32640711
## 46  2.1655516  1.3817478  1.9864405 -0.14938863  0.34015322  0.32640711
## 47  2.6694849  0.4522084  0.9932203 -0.14938863  0.34015322  0.58336590
## 50  1.6616182  0.4522084  0.9932203  0.69633060  0.34015322  1.61120105
## 52  1.1576848  0.4522084  0.9932203  0.09224544 -0.27881412 -0.31598986
## 53  0.6537514  0.4522084  0.0000000  0.45469653  1.57808790 -0.95838683
## 57 -0.3541153  1.3817478  0.0000000 -0.33061417 -0.07249167 -0.18751047
## 59  0.6537514  0.4522084  0.0000000  0.57551356  1.16544301 -0.18751047
## 60 -0.3541153  0.4522084  0.9932203 -0.27020566  0.13383078 -1.08686623
## 71  1.6616182  0.4522084  0.0000000  0.33387950  0.75279812  0.06944832
##         sugars      potass   vitamins      shelf     weight       cups
## 2   0.20460407  0.51477378 -1.3032024  0.9419715 -0.2008324  0.7567534
## 8   0.20460407  0.02097226 -0.1818422  0.9419715  1.9501886 -0.3038480
## 14 -0.02480049  0.09151534 -0.1818422  0.9419715 -0.2008324 -1.3644493
## 20 -0.02480049  0.86748914 -0.1818422  0.9419715 -0.2008324 -1.3644493
## 23  0.66341318  0.30314456 -0.1818422  0.9419715 -0.2008324 -0.3038480
## 28  0.66341318  1.43183372 -0.1818422  0.9419715  1.4287290 -0.6432404
## 29  1.12222230  1.29074758 -0.1818422  0.9419715  1.9501886 -0.6432404
## 35 -0.71301417  0.02097226 -0.1818422  0.9419715 -0.2008324 -2.0856582
## 40  0.43400862 -0.04957081  3.1822385  0.9419715  1.7546413 -0.3038480
## 42 -0.25420505 -0.04957081 -0.1818422 -0.2598542 -0.2008324 -0.6432404
## 45  0.89281774  1.00857529 -0.1818422  0.9419715 -0.2008324  0.7567534
## 46  0.89281774  1.00857529 -0.1818422  0.9419715 -0.2008324  0.7567534
## 47  1.35162686  0.86748914 -0.1818422  0.9419715  3.0582904 -0.6432404
## 50 -0.02480049  0.44423070 -0.1818422  0.9419715  1.9501886 -0.6432404
## 52  0.66341318  0.30314456 -0.1818422  0.9419715  1.4287290 -1.3644493
## 53  1.58103142  2.27835060 -0.1818422  0.9419715  1.9501886 -0.6432404
## 57 -0.25420505  0.16205841 -0.1818422  0.9419715 -0.2008324 -1.3644493
## 59  1.12222230  1.99617831 -0.1818422 -0.2598542  1.9501886 -0.3038480
## 60  0.20460407  0.58531685 -0.1818422  0.9419715 -0.2008324 -1.3644493
## 71  1.58103142  1.85509216  3.1822385  0.9419715  3.0582904  0.7567534
##         rating clust
## 2  -0.59771126     2
## 8  -0.38002951     2
## 14 -0.14048876     2
## 20 -0.13702824     2
## 23 -0.44147911     2
## 28 -0.10366038     2
## 29 -0.09664548     2
## 35  0.24511896     2
## 40 -0.42043579     2
## 42  0.21065609     2
## 45 -0.37302488     2
## 46 -0.58658904     2
## 47 -0.85924775     2
## 50 -0.11967375     2
## 52 -0.84945049     2
## 53 -0.32287913     2
## 57  0.50878106     2
## 59 -0.22179377     2
## 60 -0.19014120     2
## 71 -0.98185009     2
```

```
Cr[Cr$clust==3,]
```

```
##      calories    protein        fat     sodium       fiber       carbo
## 6   0.1498180 -0.4773310  0.9932203  0.2130625 -0.27881412 -1.08686623
## 7   0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656 -0.95838683
## 11  0.6537514 -1.4068705  0.9932203  0.6963306 -0.89778146 -0.70142805
## 13  0.6537514 -1.4068705  1.9864405  0.5755136 -0.89778146 -0.44446926
## 15  0.1498180 -1.4068705  0.0000000  0.2130625 -0.89778146 -0.70142805
## 18  0.1498180 -1.4068705 -0.9932203 -0.8742908 -0.48513656 -0.44446926
## 19  0.1498180 -1.4068705  0.0000000  0.2130625 -0.89778146 -0.70142805
## 25  0.1498180 -0.4773310  0.0000000 -0.4514312 -0.48513656 -0.95838683
## 26  0.1498180 -1.4068705 -0.9932203  0.4546965 -0.48513656 -0.18751047
## 30  0.1498180 -1.4068705  0.0000000 -0.3306142 -0.89778146 -0.44446926
## 31 -0.3541153 -0.4773310 -0.9932203 -1.4179675 -0.89778146 -0.95838683
## 32  0.1498180 -1.4068705  0.0000000  1.4212328 -0.89778146  0.06944832
## 36  0.6537514 -1.4068705  0.9932203  0.6963306 -0.48513656 -0.70142805
## 37  0.1498180  0.4522084  0.0000000  1.0587817 -0.27881412 -0.82990744
## 38  0.1498180 -1.4068705 -0.9932203  0.2130625 -0.89778146 -0.18751047
## 43  0.1498180 -0.4773310  0.0000000  0.2130625 -0.89778146 -0.70142805
## 48 -0.3541153 -0.4773310  0.0000000  0.6963306 -0.07249167  0.06944832
## 49  0.6537514 -0.4773310  0.0000000  0.3338795 -0.89778146  0.06944832
## 67  0.1498180 -0.4773310  0.0000000 -1.1159249 -0.48513656 -1.47230441
## 74  0.1498180 -1.4068705  0.0000000 -0.2702057 -0.89778146 -0.44446926
## 77  0.1498180 -0.4773310  0.0000000  0.4546965 -0.48513656  0.32640711
##        sugars     potass   vitamins      shelf     weight       cups     rating
## 6   0.6634132 -0.4022862 -0.1818422 -1.4616799 -0.2008324 -0.3038480 -0.9165248
## 7   1.5810314 -0.9666308 -0.1818422 -0.2598542 -0.2008324  0.7567534 -0.6553998
## 11  1.1222223 -0.8960877 -0.1818422 -0.2598542 -0.2008324 -0.3038480 -1.7336066
## 13  0.4340086 -0.7550015 -0.1818422 -0.2598542 -0.2008324 -0.3038480 -1.6067177
## 15  1.3516269 -0.6139154 -0.1818422 -0.2598542 -0.2008324  0.7567534 -1.3991551
## 18  1.1222223 -1.1077169 -0.1818422 -0.2598542 -0.2008324  0.7567534 -0.4695120
## 19  1.3516269 -0.4728292 -0.1818422 -0.2598542 -0.2008324  0.7567534 -1.4233777
## 25  1.3516269 -0.9666308 -0.1818422 -0.2598542 -0.2008324  0.7567534 -0.7242706
## 26  0.8928177 -1.0371738 -0.1818422 -1.4616799 -0.2008324 -0.3038480 -0.7792531
## 30  1.1222223 -1.0371738 -0.1818422 -0.2598542 -0.2008324 -0.3038480 -1.0222542
## 31  1.8104360 -0.8255446 -0.1818422 -1.4616799 -0.2008324  0.2476647 -0.5073029
## 32  0.4340086 -0.7550015 -0.1818422 -0.2598542 -0.2008324 -0.3038480 -1.3230814
## 36  0.8928177 -0.7550015 -0.1818422 -0.2598542 -0.2008324  0.7567534 -1.4608034
## 37  0.6634132 -0.1201139 -0.1818422 -1.4616799 -0.2008324 -0.3038480 -0.8051733
## 38  0.8928177 -0.8960877 -0.1818422 -1.4616799 -0.2008324  2.1567472 -0.9711880
## 43  1.1222223 -0.6139154 -0.1818422 -0.2598542 -0.2008324  0.7567534 -1.1142648
## 48 -0.2542051 -0.1201139 -0.1818422 -1.4616799 -0.2008324  0.7567534 -0.1614556
## 49  0.4340086 -0.8255446 -0.1818422 -0.2598542 -0.2008324 -0.6432404 -0.8869714
## 67  1.8104360 -0.8255446 -0.1818422 -0.2598542 -0.2008324 -0.3038480 -0.7939263
## 74  1.1222223 -1.0371738 -0.1818422 -0.2598542 -0.2008324  0.7567534 -1.0416692
## 77  0.2046041 -0.5433723 -0.1818422 -1.4616799 -0.2008324 -0.3038480 -0.4406694
##    clust
## 6      3
## 7      3
## 11     3
## 13     3
## 15     3
## 18     3
## 19     3
## 25     3
## 26     3
## 30     3
## 31      3
## 32      3
## 36      3
## 37      3
## 38      3
## 43      3
## 48      3
## 49      3
## 67      3
## 74      3
## 77      3
```

```
Cr[Cr$clust==4,]
```

```
##      calories    protein        fat      sodium       fiber       carbo
## 9  -0.8580487 -0.4773310  0.0000000  0.45469653  0.75279812  0.06944832
## 10 -0.8580487  0.4522084 -0.9932203  0.57551356  1.16544301 -0.44446926
## 12  0.1498180  3.2408266  0.9932203  1.54204982 -0.07249167  0.58336590
## 16  0.1498180 -0.4773310 -0.9932203  1.42123279 -0.89778146  1.86815984
## 17 -0.3541153 -0.4773310 -0.9932203  1.54204982 -0.48513656  1.61120105
## 22  0.1498180 -0.4773310 -0.9932203  0.69633060 -0.48513656  1.61120105
## 24 -0.3541153 -0.4773310 -0.9932203  0.33387950 -0.48513656  0.84032469
## 27 -0.3541153  0.4522084 -0.9932203 -1.96164410  0.34015322 -0.18751047
## 33 -0.3541153  0.4522084  0.0000000 -0.27020566  0.34015322  0.06944832
## 34  0.1498180  0.4522084 -0.9932203  0.09224544  0.34015322  0.58336590
## 39  0.1498180 -0.4773310  0.0000000  0.09224544 -0.48513656  0.58336590
## 41  0.1498180 -0.4773310  0.0000000  1.17959872 -0.89778146  1.61120105
## 44 -0.3541153  1.3817478  0.0000000 -1.96164410 -0.89778146  0.32640711
## 51 -0.8580487  0.4522084 -0.9932203  0.09224544  0.34015322  0.84032469
## 54 -0.3541153  0.4522084 -0.9932203  1.90450091 -0.48513656  1.35424227
## 55 -2.8737823 -1.4068705 -0.9932203 -1.96164410 -0.89778146 -0.44446926
## 56 -2.8737823 -0.4773310 -0.9932203 -1.96164410 -0.48513656 -1.21534562
## 61 -0.8580487 -0.4773310 -0.9932203 -1.96164410 -0.07249167  0.06944832
## 62  0.1498180 -1.4068705 -0.9932203  0.93796466 -0.89778146  2.12511863
## 63  0.1498180 -0.4773310 -0.9932203  1.54204982 -0.89778146  1.86815984
## 64 -1.3619821 -0.4773310 -0.9932203 -1.96164410  0.34015322  0.32640711
## 65 -0.8580487  0.4522084 -0.9932203 -1.96164410  0.75279812  1.09728348
## 66 -0.8580487  0.4522084 -0.9932203 -1.96164410  0.34015322  1.35424227
## 68  0.1498180  3.2408266 -0.9932203  0.81714763 -0.48513656  0.32640711
## 69 -0.8580487 -0.4773310 -0.9932203 -1.78041856  0.34015322  0.06944832
## 70  0.1498180 -0.4773310  0.0000000  0.45469653 -0.89778146  1.61120105
## 72 -0.3541153  0.4522084  0.0000000  0.45469653  0.34015322  0.32640711
## 73  0.1498180 -0.4773310  0.0000000  1.05878169 -0.89778146  1.61120105
## 75 -0.3541153  0.4522084  0.0000000  0.81714763  0.34015322  0.58336590
## 76 -0.3541153  0.4522084  0.0000000  0.45469653  0.34015322  0.58336590
##         sugars      potass   vitamins      shelf     weight        cups
## 9  -0.25420505  0.37368763 -0.1818422 -1.4616799 -0.2008324 -0.64324039
## 10 -0.48360961  1.29074758 -0.1818422  0.9419715 -0.2008324 -0.64324039
## 12 -1.40122785  0.09151534 -0.1818422 -1.4616799 -0.2008324  1.81735475
## 16 -0.94241873 -1.03717383 -0.1818422 -1.4616799 -0.2008324  0.75675340
## 17 -1.17182329 -0.89608768 -0.1818422 -1.4616799 -0.2008324  0.75675340
## 22 -0.94241873 -0.96663076 -0.1818422  0.9419715 -0.2008324  0.75675340
## 24 -0.48360961 -0.26120003 -0.1818422  0.9419715 -0.2008324 -0.30384795
## 27 -0.02480049  0.02097226 -0.1818422 -0.2598542 -0.2008324 -0.09172768
## 33 -0.48360961 -0.19065695 -0.1818422  0.9419715 -0.2008324  0.24766475
## 34 -0.94241873 -0.12011388 -0.1818422  0.9419715 -0.2008324 -2.42505066
## 39 -0.25420505 -0.54337232  3.1822385  0.9419715 -0.2008324  0.75675340
## 41 -0.94241873 -0.82554461 -0.1818422 -0.2598542 -0.2008324  2.87795610
## 44 -0.94241873 -0.04957081 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 51 -1.17182329 -0.12011388 -0.1818422  0.9419715 -0.2008324  0.75675340
## 54 -0.94241873 -0.75500154  3.1822385  0.9419715 -0.2008324  0.75675340
## 55 -1.63063240 -1.17825998 -1.3032024  0.9419715 -3.4599552  0.75675340
## 56 -1.63063240 -0.68445846 -1.3032024  0.9419715 -3.4599552  0.75675340
## 61 -0.25420505  0.16205841 -0.1818422  0.9419715 -0.2008324 -1.36444931
## 62 -1.17182329 -0.96663076 -0.1818422 -1.4616799 -0.2008324  1.30826610
## 63 -0.94241873 -0.89608768 -0.1818422 -1.4616799 -0.2008324  0.75675340
## 64 -1.63063240 -0.04957081 -1.3032024 -1.4616799 -1.3089342  0.75675340
## 65 -1.63063240  0.58531685 -1.3032024 -1.4616799 -0.2008324 -0.64324039
## 66 -1.63063240  0.30314456 -1.3032024 -1.4616799 -0.2008324 -0.64324039
## 68 -0.94241873 -0.61391539 -0.1818422 -1.4616799 -0.2008324  0.75675340
## 69 -0.48360961 -0.12011388 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 70 -0.94241873 -0.89608768  3.1822385  0.9419715 -0.2008324  0.75675340
## 72 -0.94241873  0.16205841  3.1822385  0.9419715 -0.2008324  0.75675340
## 73 -0.94241873 -0.54337232 -0.1818422  0.9419715 -0.2008324 -0.30384795
## 75 -0.94241873  0.23260148 -0.1818422 -1.4616799 -0.2008324 -0.64324039
## 76 -0.94241873  0.16205841 -0.1818422 -1.4616799 -0.2008324  0.75675340
##         rating clust
## 9   0.48087533     4
## 10  0.77969576     4
## 12  0.59807496     4
## 16 -0.06603869     4
## 17  0.24879639     4
## 22  0.32235640     4
## 24  0.13959735     4
## 27  1.13821301     4
## 33  0.69155685     4
## 34  0.78377123     4
## 39 -0.41671824     4
## 41 -0.22308231     4
## 44  0.88922515     4
## 51  1.23068291     4
## 54 -0.06186866     4
## 55  1.31001152     4
## 56  1.47030646     4
## 61  0.92358705     4
## 62 -0.02656845     4
## 63 -0.12909114     4
## 64  1.84299757     4
## 65  2.28743193     4
## 66  2.16834997     4
## 68  0.76669214     4
## 69  1.21081332     4
## 70 -0.25168258     4
## 72  0.30548275     4
## 73 -0.23269772     4
## 75  0.52841741     4
## 76  0.65701831     4
```

Determining the best cluster using mean values

```
mean(Cr[Cr$clust==1,"rating"])
```

```
## [1] 2.242648
```

```
mean(Cr[Cr$clust==2,"rating"])
```

```
## [1] -0.2928786
```

```
mean(Cr[Cr$clust==3,"rating"])
```

```
## [1] -0.9636465
```

```
mean(Cr[Cr$clust==4,"rating"])
```

```
## [1] 0.6455402
```

Cluster 1 has the highest mean rating. So, we can choose cluster 1.