Assignment_5

Brianna Viñas

11/28/2021

```
#Adding the libraries necessary for the codes that will be used.
library(cluster) #needed for the clustering algorithms
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     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(tidyverse)#needed in order to be able to manipulate the data
library(factoextra) #needed in order to visualize the clustering taking place
```

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

```
library(dendextend)# this library is used to compare the two dendrograms.
```

```
##
## ---------------------
## 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/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)
```

```
#Reading the Cereals .csv file.
Cer<- read.csv("Cereals.csv")
```

```
#I will now proceed with removing all cereals with missing variables.
sum(is.na(Cer))
```

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

```
BV<-na.omit(Cer)
sum(is.na(BV))# This was done to verify that the missing variables were removed.
```

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

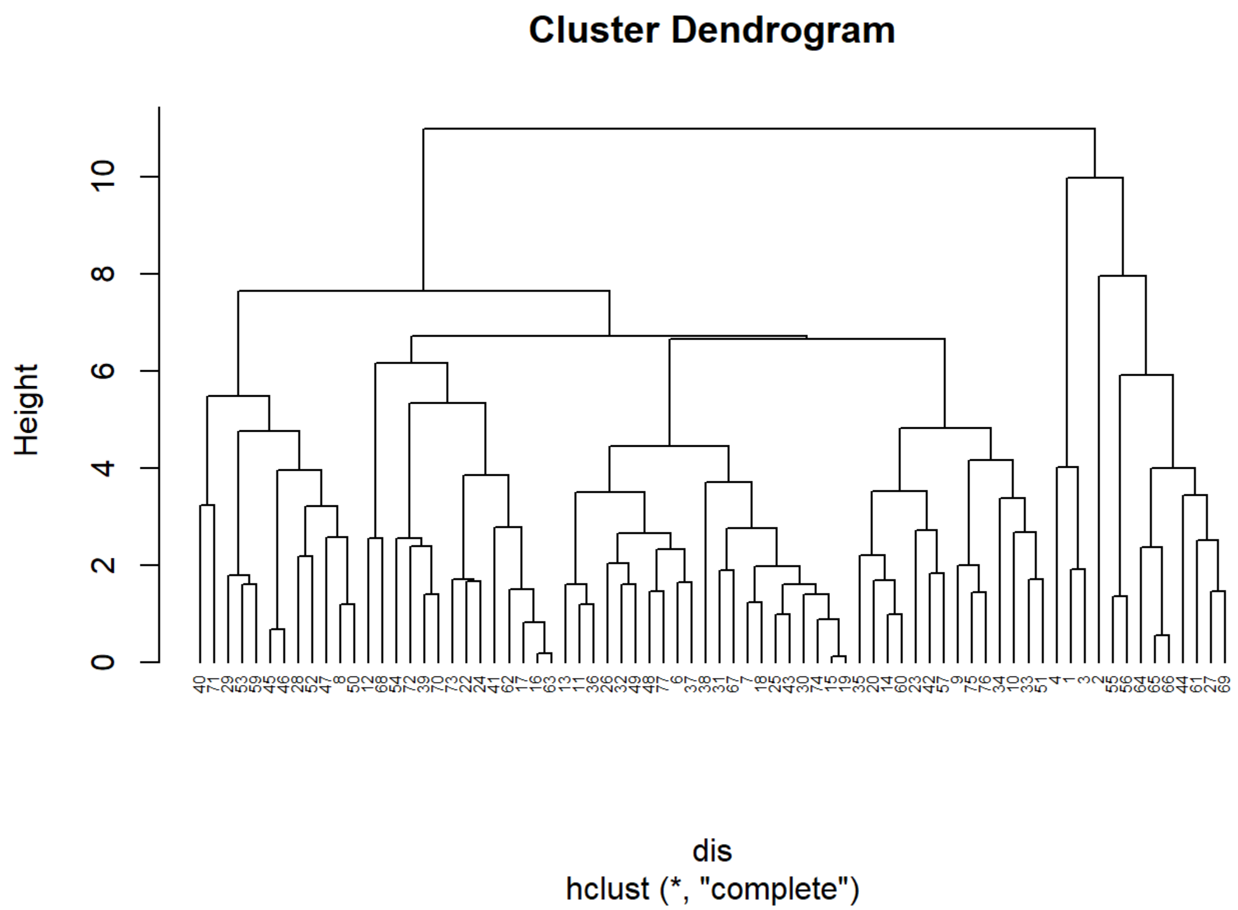
```
BV1<-BV[c(-1,-2,-3)]
```

```
#Now, I will be scaling the remaining data.
BV2<-scale(BV1)
```

```
# Question # 1: Apply hierarchical clustering to the data using Euclidean distance to the normal
ized measurements.Use Agnes to compare the clustering from single linkage, complete linkage, ave
rage linkage, and Ward.

# To answer this question, I will apply hierarchical clustering to the data using the Euclidean
 distance and the dissimilarity matrix.

dis<- dist(BV2, method= "euclidean")
```

```
#Continuation of answering question 1.
#I will be continuing to answer question one by using Hierarchical Clustering with the use of co
mplete linkage.

hc.complete<-hclust(dis, method = "complete")
```

```
#continuation of answering question 1.
# Now, I will plot the obtained dendogram.
plot(hc.complete,cex= 0.5, hang=-1)
```




```
#continuation of answering question 1.
# I will now continue by using Agnes to compare the clustering from single linkage, complete lin
kage average linkage, and ward.

c_single<- agnes(BV2, method ="single")
c_complete<- agnes (BV2, method = "complete")
c_average <- agnes (BV2, method= "average")
c_ward<- agnes(BV2, method = "ward")
```

```
# Continuation of answering question 1.
#Now, I will continue by comparing the agglomerative coefficients which includes the single, com
plete, average, and ward methods.

c_single$ac
```

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

```
c_complete$ac
```

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

```
c_average$ac
```

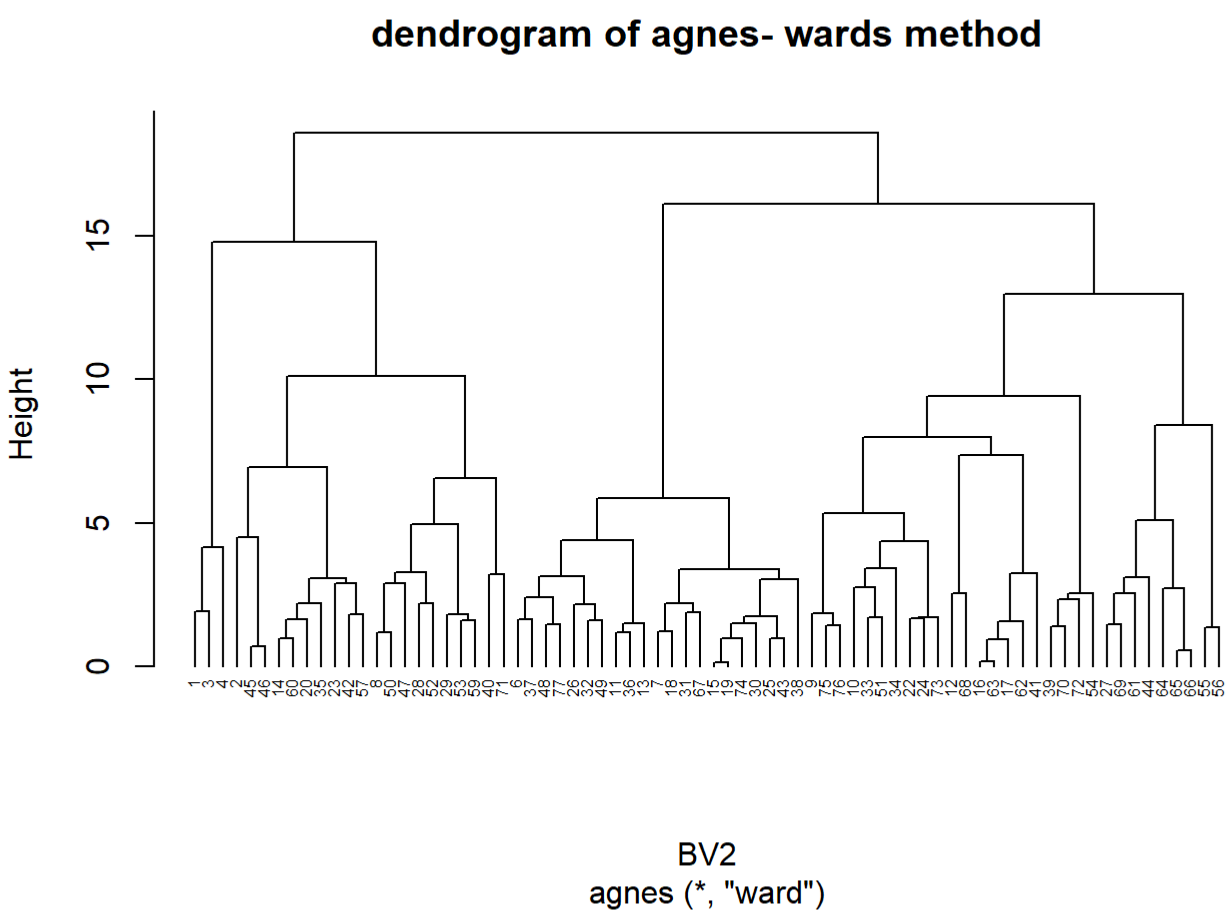
```
## [1] 0.7766075
```

```
c_ward$ac
```

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

```
#Answer to Question # 1: By observing the aforementioned values it can be concluded that the bes
t linkage method is ward with the agloermative coefficient of 0.9046042.
```

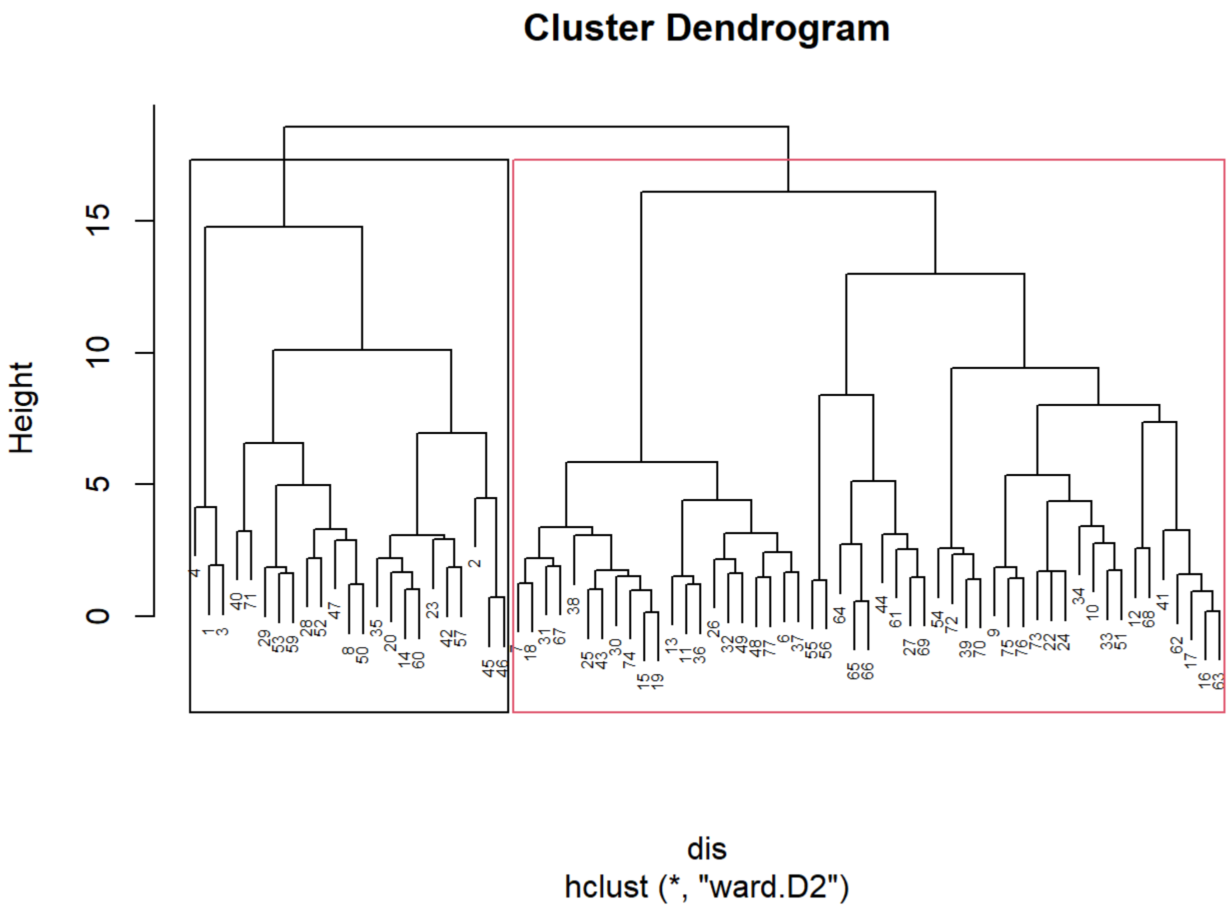
```
# Before moving onto Question #2, I will visualize the dendogram using the ward method.

al<-pltree(c_ward, cex= 0.5, hang = -1, main = "dendrogram of agnes- wards method")
```



```
#Question # 2: How many clusters would you choose?
# First, I will create the distance matrix.

dis<- dist(BV2, method = "euclidean")
```

```
# Continuation of answering question 2.
c_sv<-hclust(dis,method = "ward.D2")
```

```
# Continuation of question 2.
# Here I will plot the dendrogram and take k=2 by observing the distance.

plot(c_sv, cex=0.5)
rect.hclust(c_sv, k=2, border = 1:2)
```




```
#Continuation of question 2.
# In order to identify the clusters, I will cut the dendrogram with cutree ().

cus<-cutree(c_sv, k=2)

#Number of members in each cluster.

table(cus)
```

```
## cus
##  1  2
## 23 51
```

```
#Answer to Question # 2: It can be concluded that k=2 is cutting the longest path, so in this ca
se I will select k=2.
```

```
# Question # 3: # Comment on the structure of the clusters and on their stability.
```

```
#Continuation of question 3.
# Now, I will set the seed.

set.seed (15)
Cer_Ne <-Cer
```

```
#Continuation of question 3.
# To continue, I will remove any missing values that may be present in the data.

cs<-na.omit(Cer_Ne)
cs1<-cs[,c(-1,-2,-3)]
cs2<-scale(cs1)
cs3<-as.data.frame(cs2)
```

```
#Continuation of question 3.
#Now, I will divide the data and create the partitions.

c1<-cs[1:55,]
c2<-cs[56:74,]
```

```
#Continuation of question 3.
# I will now perform clustering uses Agnes () with single, complete, average, and ward with part
itioned data.

b1<-agnes(scale(c1[,-c(1:3)]), method = "ward")
b2<-agnes(scale(c1[,-c(1:3)]), method = "average")
b3<-agnes(scale(c1[-c(1:3)]), method = "complete")
b4 <-agnes(scale(c1[-c(1:3)]), method = "single")
cbind(ward=b1$ac, average=b2$ac, complete=b3$ac, single=b4$ac)
```

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

```
d3<-cutree(b1,k=2)
```

```
# Continuation of question 3.
# Here I will calculate the centers.

bb<-as.data.frame(cbind(scale(c1[,-c(1:3)]),d3))
cen1<-colMeans(bb[bb$d3==1,])
cen2<-colMeans(bb[bb$d3==2,])
```

```
#Continuation of question 3.
# Here I will bind the two centers together.

cen<- rbind(cen1,cen2)
cen
```

```
##        calories    protein        fat      sodium      fiber      carbo
## cen1  0.2870363  0.7424779  0.7043608 -0.18054158  0.7754394 -0.3921999
## cen2 -0.1514914 -0.3918634 -0.3717460  0.09528584 -0.4092597  0.2069944
##           sugars     potass   vitamins      shelf     weight       cups
## cen1  0.11389860  0.9717483 -0.2193799  0.7766318  0.5853498 -0.7197837
## cen2 -0.06011315 -0.5128671  0.1157838 -0.4098890 -0.3089346  0.3798858
##          rating d3
## cen1  0.2618963  1
## cen2 -0.1382230  2
```

```
#Continuation of question 3.
#Here I will calculate the distance the distance.

z<- as.data.frame(rbind(cen[,-14],scale(c2[,-c(1:3)])))
x1<-get_dist(z)
x2<-as.matrix(x1)
a1<-data.frame(data=seq(1,nrow(c2),1), clusters =rep(0,nrow(c2)))

for(i in 1:nrow(c2))
{
  a1[i,2]<-which.min(x2[i+2,1:2])
}
  a1
```

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

```
a3<-as.data.frame(cbind(BV2,cus))
cbind(a3$cus[56:74],a1$clusters)
```

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

```
table(a1$cus[56:74]==a1$clusters)
```

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

```
#Answer to Question #3: From the aforementioned values, we can say that the clusters are rather
 stable.
```

```
#Question # 4: Healthy Cereals

n<-cbind(cs3,cus)
n[n$cus==1,]
```

```
##      calories   protein        fat      sodium       fiber       carbo
## 1  -1.8659155  1.3817478  0.0000000 -0.39102269  3.22866747 -2.50013957
## 2   0.6537514  0.4522084  3.9728810 -1.78041856 -0.07249167 -1.72926320
## 3  -1.8659155  1.3817478  0.0000000  1.17959872  2.81602258 -1.98622199
## 4  -2.8737823  1.3817478 -0.9932203 -0.27020566  4.87924705 -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
## 1  -0.25420505  2.56052289 -0.1818422  0.9419715 -0.2008324 -2.0856582
## 2   0.20460407  0.51477378 -1.3032024  0.9419715 -0.2008324  0.7567534
## 3  -0.48360961  3.12486748 -0.1818422  0.9419715 -0.2008324 -2.0856582
## 4  -1.63063240  3.26595362 -0.1818422  0.9419715 -0.2008324 -1.3644493
## 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 cus
## 1   1.85490376   1
## 2  -0.59771126   1
## 3   1.21519648   1
```

```
## 4   3.65784358   1
## 8  -0.38002951   1
## 14 -0.14048876   1
## 20 -0.13702824   1
## 23 -0.44147911   1
## 28 -0.10366038   1
## 29 -0.09664548   1
## 35  0.24511896   1
## 40 -0.42043579   1
## 42  0.21065609   1
## 45 -0.37302488   1
## 46 -0.58658904   1
## 47 -0.85924775   1
## 50 -0.11967375   1
## 52 -0.84945049   1
## 53 -0.32287913   1
## 57  0.50878106   1
## 59 -0.22179377   1
## 60 -0.19014120   1
## 71 -0.98185009   1
```

```
n[n$cus==2,]
```

```
##      calories    protein        fat      sodium       fiber       carbo
## 6   0.1498180 -0.4773310  0.9932203  0.21306247 -0.27881412 -1.08686623
## 7   0.1498180 -0.4773310 -0.9932203 -0.45143121 -0.48513656 -0.95838683
## 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
## 11  0.6537514 -1.4068705  0.9932203  0.69633060 -0.89778146 -0.70142805
## 12  0.1498180  3.2408266  0.9932203  1.54204982 -0.07249167  0.58336590
## 13  0.6537514 -1.4068705  1.9864405  0.57551356 -0.89778146 -0.44446926
## 15  0.1498180 -1.4068705  0.0000000  0.21306247 -0.89778146 -0.70142805
## 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
## 18  0.1498180 -1.4068705 -0.9932203 -0.87429082 -0.48513656 -0.44446926
## 19  0.1498180 -1.4068705  0.0000000  0.21306247 -0.89778146 -0.70142805
## 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
## 25  0.1498180 -0.4773310  0.0000000 -0.45143121 -0.48513656 -0.95838683
## 26  0.1498180 -1.4068705 -0.9932203  0.45469653 -0.48513656 -0.18751047
## 27 -0.3541153  0.4522084 -0.9932203 -1.96164410  0.34015322 -0.18751047
## 30  0.1498180 -1.4068705  0.0000000 -0.33061417 -0.89778146 -0.44446926
## 31 -0.3541153 -0.4773310 -0.9932203 -1.41796746 -0.89778146 -0.95838683
## 32  0.1498180 -1.4068705  0.0000000  1.42123279 -0.89778146  0.06944832
## 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
## 36  0.6537514 -1.4068705  0.9932203  0.69633060 -0.48513656 -0.70142805
## 37  0.1498180  0.4522084  0.0000000  1.05878169 -0.27881412 -0.82990744
## 38  0.1498180 -1.4068705 -0.9932203  0.21306247 -0.89778146 -0.18751047
## 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
## 43  0.1498180 -0.4773310  0.0000000  0.21306247 -0.89778146 -0.70142805
## 44 -0.3541153  1.3817478  0.0000000 -1.96164410 -0.89778146  0.32640711
## 48 -0.3541153 -0.4773310  0.0000000  0.69633060 -0.07249167  0.06944832
## 49  0.6537514 -0.4773310  0.0000000  0.33387950 -0.89778146  0.06944832
## 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
## 67  0.1498180 -0.4773310  0.0000000 -1.11592488 -0.48513656 -1.47230441
## 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
## 74  0.1498180 -1.4068705  0.0000000 -0.27020566 -0.89778146 -0.44446926
## 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
## 77  0.1498180 -0.4773310  0.0000000  0.45469653 -0.48513656  0.32640711
```

```
##         sugars      potass   vitamins      shelf     weight        cups
## 6   0.66341318 -0.40228617 -0.1818422 -1.4616799 -0.2008324 -0.30384795
## 7   1.58103142 -0.96663076 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 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
## 11  1.12222230 -0.89608768 -0.1818422 -0.2598542 -0.2008324 -0.30384795
## 12 -1.40122785  0.09151534 -0.1818422 -1.4616799 -0.2008324  1.81735475
## 13  0.43400862 -0.75500154 -0.1818422 -0.2598542 -0.2008324 -0.30384795
## 15  1.35162686 -0.61391539 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 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
## 18  1.12222230 -1.10771690 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 19  1.35162686 -0.47282925 -0.1818422 -0.2598542 -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
## 25  1.35162686 -0.96663076 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 26  0.89281774 -1.03717383 -0.1818422 -1.4616799 -0.2008324 -0.30384795
## 27 -0.02480049  0.02097226 -0.1818422 -0.2598542 -0.2008324 -0.09172768
## 30  1.12222230 -1.03717383 -0.1818422 -0.2598542 -0.2008324 -0.30384795
## 31  1.81043598 -0.82554461 -0.1818422 -1.4616799 -0.2008324  0.24766475
## 32  0.43400862 -0.75500154 -0.1818422 -0.2598542 -0.2008324 -0.30384795
## 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
## 36  0.89281774 -0.75500154 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 37  0.66341318 -0.12011388 -0.1818422 -1.4616799 -0.2008324 -0.30384795
## 38  0.89281774 -0.89608768 -0.1818422 -1.4616799 -0.2008324  2.15674718
## 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
## 43  1.12222230 -0.61391539 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 44 -0.94241873 -0.04957081 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 48 -0.25420505 -0.12011388 -0.1818422 -1.4616799 -0.2008324  0.75675340
## 49  0.43400862 -0.82554461 -0.1818422 -0.2598542 -0.2008324 -0.64324039
## 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
## 67  1.81043598 -0.82554461 -0.1818422 -0.2598542 -0.2008324 -0.30384795
## 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
## 74  1.12222230 -1.03717383 -0.1818422 -0.2598542 -0.2008324  0.75675340
## 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
## 77  0.20460407 -0.54337232 -0.1818422 -1.4616799 -0.2008324 -0.30384795
```

```
##         rating cus
## 6  -0.91652483   2
## 7  -0.65539984   2
## 9   0.48087533   2
## 10  0.77969576   2
## 11 -1.73360655   2
## 12  0.59807496   2
## 13 -1.60671768   2
## 15 -1.39915514   2
## 16 -0.06603869   2
## 17  0.24879639   2
## 18 -0.46951197   2
## 19 -1.42337774   2
## 22  0.32235640   2
## 24  0.13959735   2
## 25 -0.72427057   2
## 26 -0.77925310   2
## 27  1.13821301   2
## 30 -1.02225423   2
## 31 -0.50730289   2
## 32 -1.32308140   2
## 33  0.69155685   2
## 34  0.78377123   2
## 36 -1.46080340   2
## 37 -0.80517325   2
## 38 -0.97118798   2
## 39 -0.41671824   2
## 41 -0.22308231   2
## 43 -1.11426481   2
## 44  0.88922515   2
## 48 -0.16145563   2
## 49 -0.88697142   2
## 51  1.23068291   2
## 54 -0.06186866   2
## 55  1.31001152   2
## 56  1.47030646   2
## 61  0.92358705   2
## 62 -0.02656845   2
## 63 -0.12909114   2
## 64  1.84299757   2
## 65  2.28743193   2
## 66  2.16834997   2
## 67 -0.79392626   2
## 68  0.76669214   2
## 69  1.21081332   2
## 70 -0.25168258   2
## 72  0.30548275   2
## 73 -0.23269772   2
## 74 -1.04166919   2
## 75  0.52841741   2
## 76  0.65701831   2
## 77 -0.44066942   2
```

```
#Continuation of question # 4.
#Here I am going to calculate the mean ratings to determine the best cluster.

mean(n[n$cus==1,"rating"])
```

```
## [1] 0.03784223
```

```
mean(n[n$cus==2,"rating"])
```

```
## [1] -0.0170661
```

```
#Answer to question # 4: Here we can see from the above cluster analysis has high rating values.
Thus, we can infer that this cluster has more nutrition values. So, Cluster1 is healthier for ch
ildren. Given that we have used the distance metric algorithm we essentially needed to normalize
the data as the factors of the data were different. Thus, we needed to scale it to similar featu
res.
```