R Notebook

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v ggplot2 3.2.1     v forcats 0.4.0
```

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

```
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

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

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.12.0
## 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
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

Read the data

```
Cereals <- read_csv("Cereals.csv")
```

```
## Parsed with column specification:
## cols(
##   name = col_character(),
##   mfr = col_character(),
##   type = col_character(),
##   calories = col_double(),
##   protein = col_double(),
##   fat = col_double(),
##   sodium = col_double(),
##   fiber = col_double(),
##   carbo = col_double(),
##   sugars = col_double(),
##   potass = col_double(),
##   vitamins = col_double(),
##   shelf = col_double(),
##   weight = col_double(),
##   cups = col_double(),
##   rating = col_double()
## )
```

```
head(Cereals)
```

name <chr>	… <chr>	ty… <chr>	calories <dbl>	protein <dbl>	fat <dbl>	sodi… <dbl>	fiber <dbl>	car… <dbl>	sugars <dbl>
100%_Bran	N	C	70	4	1	130	10.0	5.0	6
100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8
All-Bran	K	C	70	4	1	260	9.0	7.0	5
All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14.0	8.0	0
Almond_Delight	R	C	110	2	2	200	1.0	14.0	8
Apple_Cinnamon_Cheerios	G	C	110	2	2	180	1.5	10.5	10

6 rows | 1-10 of 16 columns

```
set.seed(15)
```

Data Preprocessing. Remove all cereals with missing values

```
# Number of missing values
sum(is.na(Cereals))
```

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

```
# Remove all cereals with missing values
MyData <- na.omit(Cereals)
#str(MyData)
```

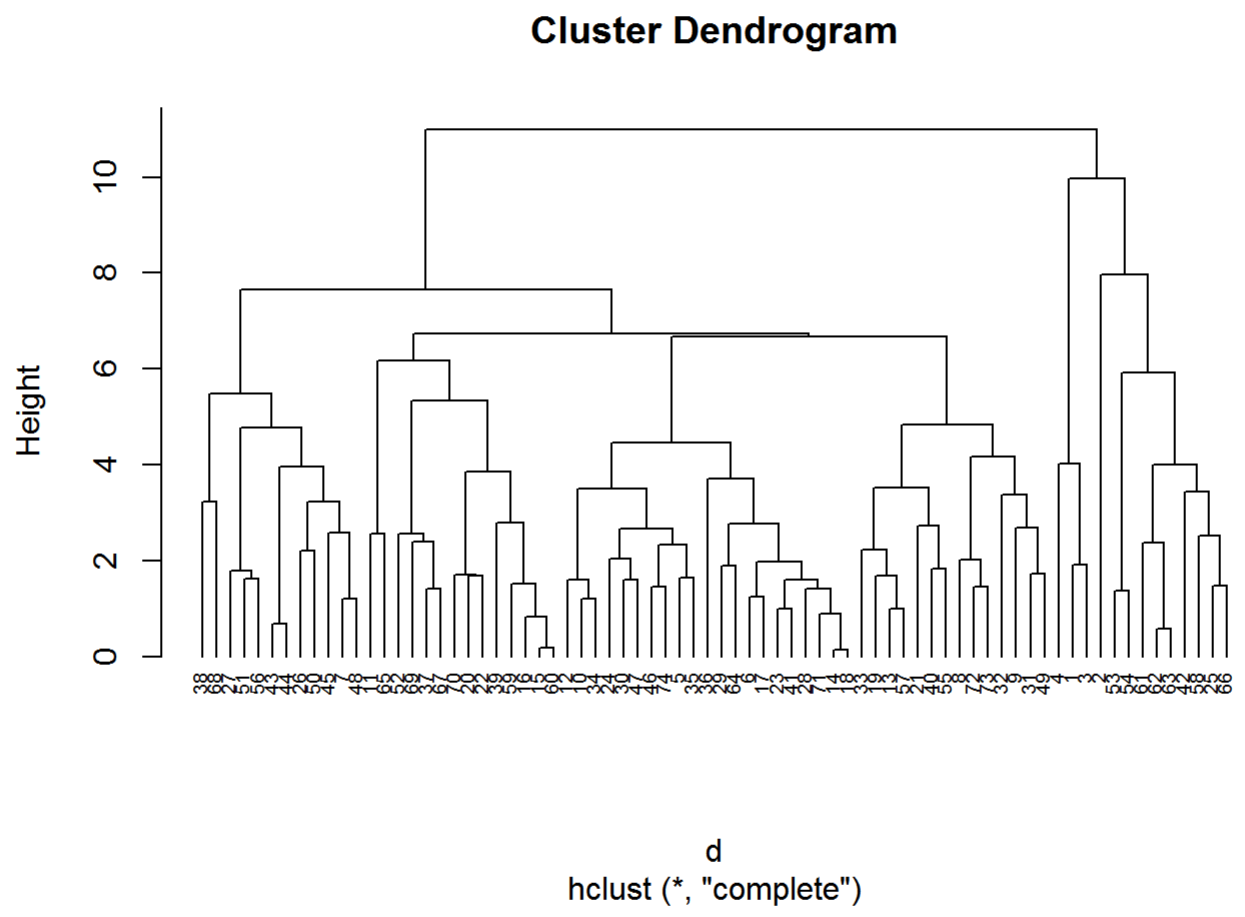
Normalization and Scale the Data

```
Cerealnames <- MyData$name
# Drop the Categorical Columns
MyData <- MyData[, c(-1, -2, -3)]
MyData <- scale(MyData, center = T, scale = T)
head(MyData)
```

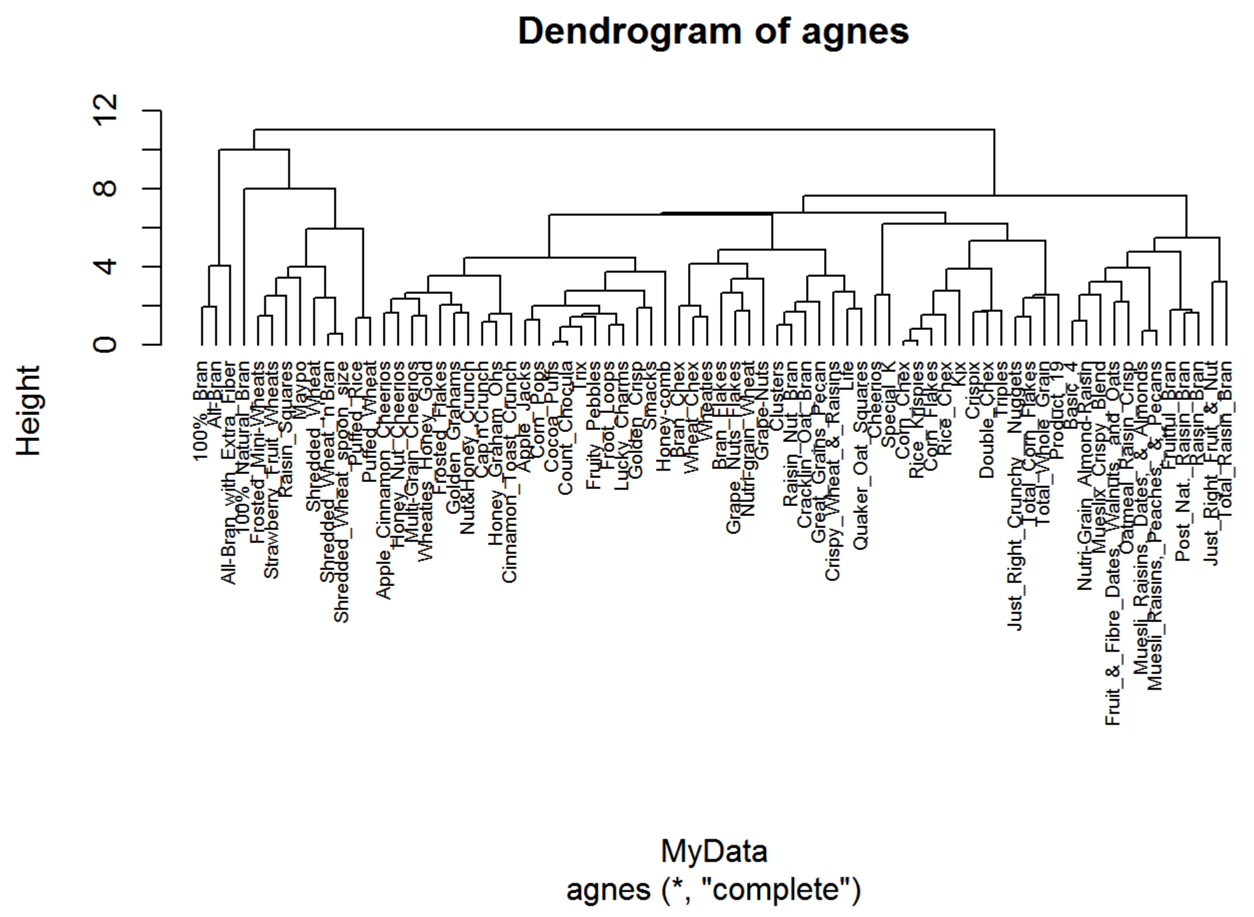
```
##        calories    protein        fat     sodium       fiber      carbo
## [1,] -1.8659155  1.3817478  0.0000000 -0.3910227  3.22866747 -2.5001396
## [2,]  0.6537514  0.4522084  3.9728810 -1.7804186 -0.07249167 -1.7292632
## [3,] -1.8659155  1.3817478  0.0000000  1.1795987  2.81602258 -1.9862220
## [4,] -2.8737823  1.3817478 -0.9932203 -0.2702057  4.87924705 -1.7292632
## [5,]  0.1498180 -0.4773310  0.9932203  0.2130625 -0.27881412 -1.0868662
## [6,]  0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656 -0.9583868
##          sugars     potass   vitamins      shelf     weight       cups
## [1,] -0.2542051  2.5605229 -0.1818422  0.9419715 -0.2008324 -2.0856582
## [2,]  0.2046041  0.5147738 -1.3032024  0.9419715 -0.2008324  0.7567534
## [3,] -0.4836096  3.1248675 -0.1818422  0.9419715 -0.2008324 -2.0856582
## [4,] -1.6306324  3.2659536 -0.1818422  0.9419715 -0.2008324 -1.3644493
## [5,]  0.6634132 -0.4022862 -0.1818422 -1.4616799 -0.2008324 -0.3038480
## [6,]  1.5810314 -0.9666308 -0.1818422 -0.2598542 -0.2008324  0.7567534
##          rating
## [1,]  1.8549038
## [2,] -0.5977113
## [3,]  1.2151965
## [4,]  3.6578436
## [5,] -0.9165248
## [6,] -0.6553998
```

1. Apply hierarchical clustering to the data using Euclidean distance to the normaliMyDataed measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method.

```
# Dissimilarity matrix
d <- dist(MyData, method = "euclidean")

# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```



```
# Compute with agnes and with different linkage methods
hc_single <- agnes(MyData, method = "single")
hc_complete <- agnes(MyData, method = "complete")
hc_average <- agnes(MyData, method = "average")
hc_ward <- agnes(MyData, method = 'ward')

pltree(hc_complete, cex = 0.6, hang = -1, main = "Dendrogram of agnes", labels = Cerealnames)
```

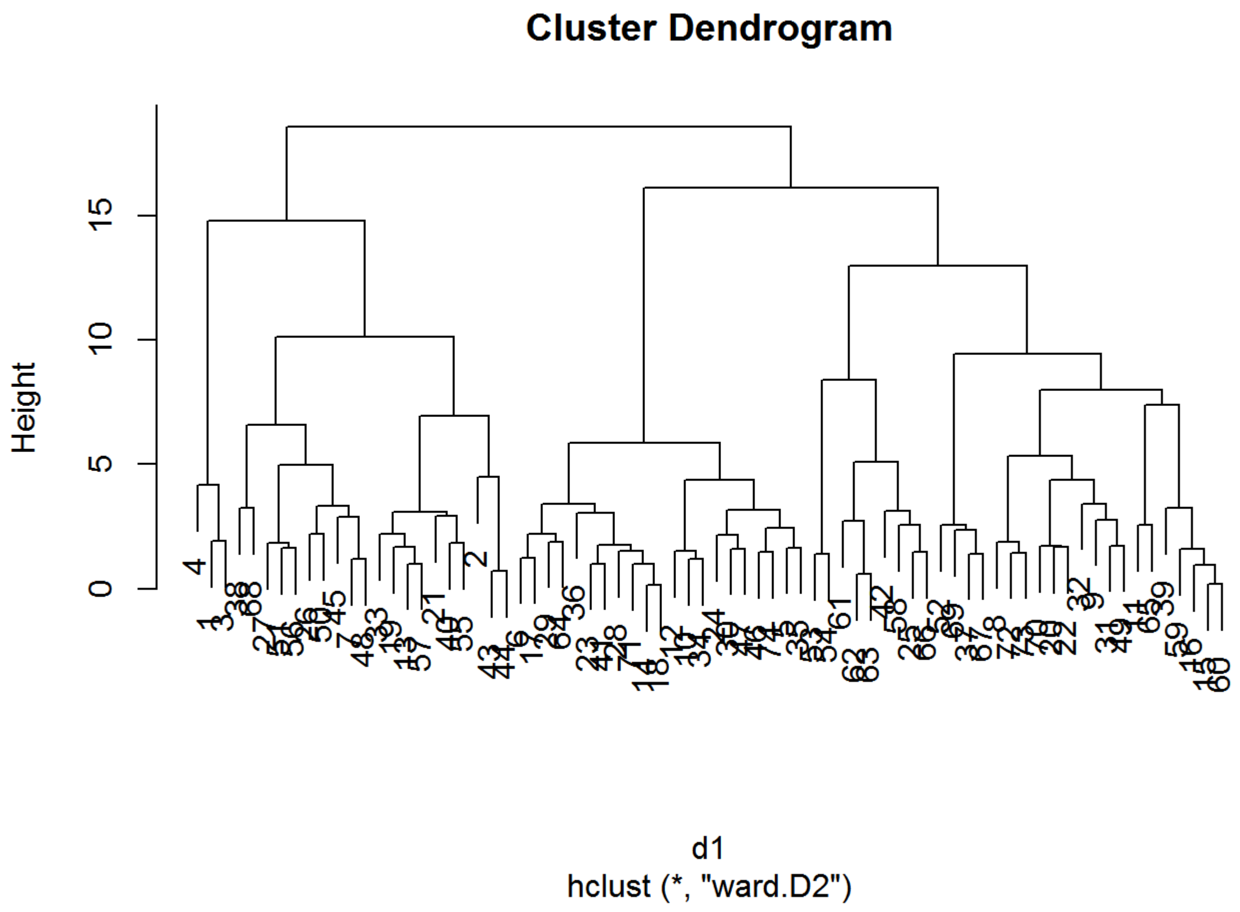


```r
# Compare Agglomerative Coefficients

m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient

ac <- function(x) {
  agnes(MyData, method = x)$ac
}

map_dbl(m, ac)
```

```
##   average    single  complete      ward
## 0.7766075 0.6067859 0.8353712 0.9046042
```

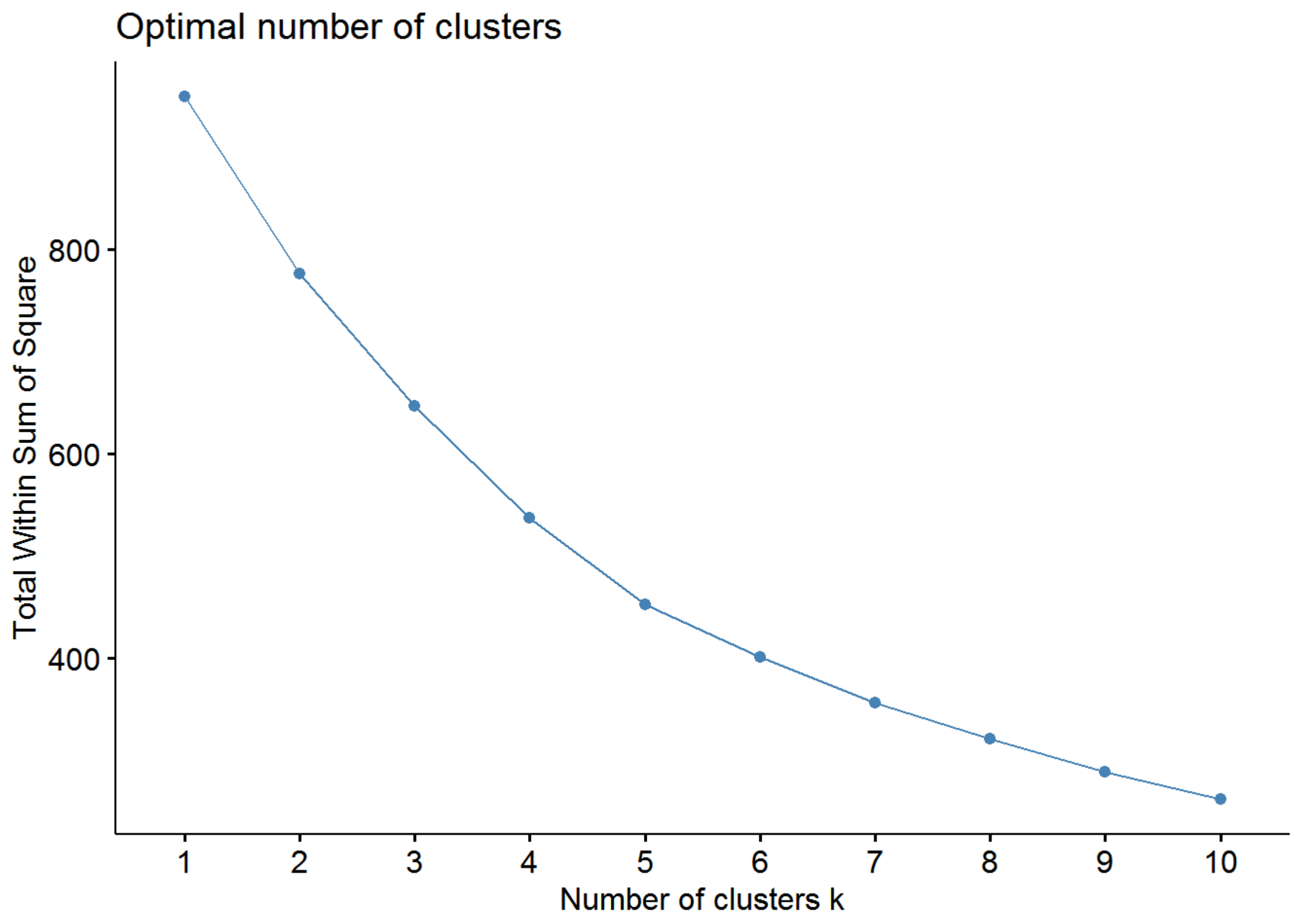
On comparing the best method is ‘Ward’

```
# Dissimilarity matrix
d1 <- dist(MyData, method = "euclidean")

# Hierarchical clustering using Ward Linkage
hc2 <- hclust(d1, method = "ward.D2" )

# Plot the obtained dendrogram
plot(hc2)
```



2. How many clusters would you choose?

```
# Elbow method to chosse K
fviz_nbclust(MyData, FUN = hcut, method = "wss")
```



Not able to determine K based on Elbow methid.

Let’s try based on Dendogram using rect.hclust method by cutting the tree into 4 clusters

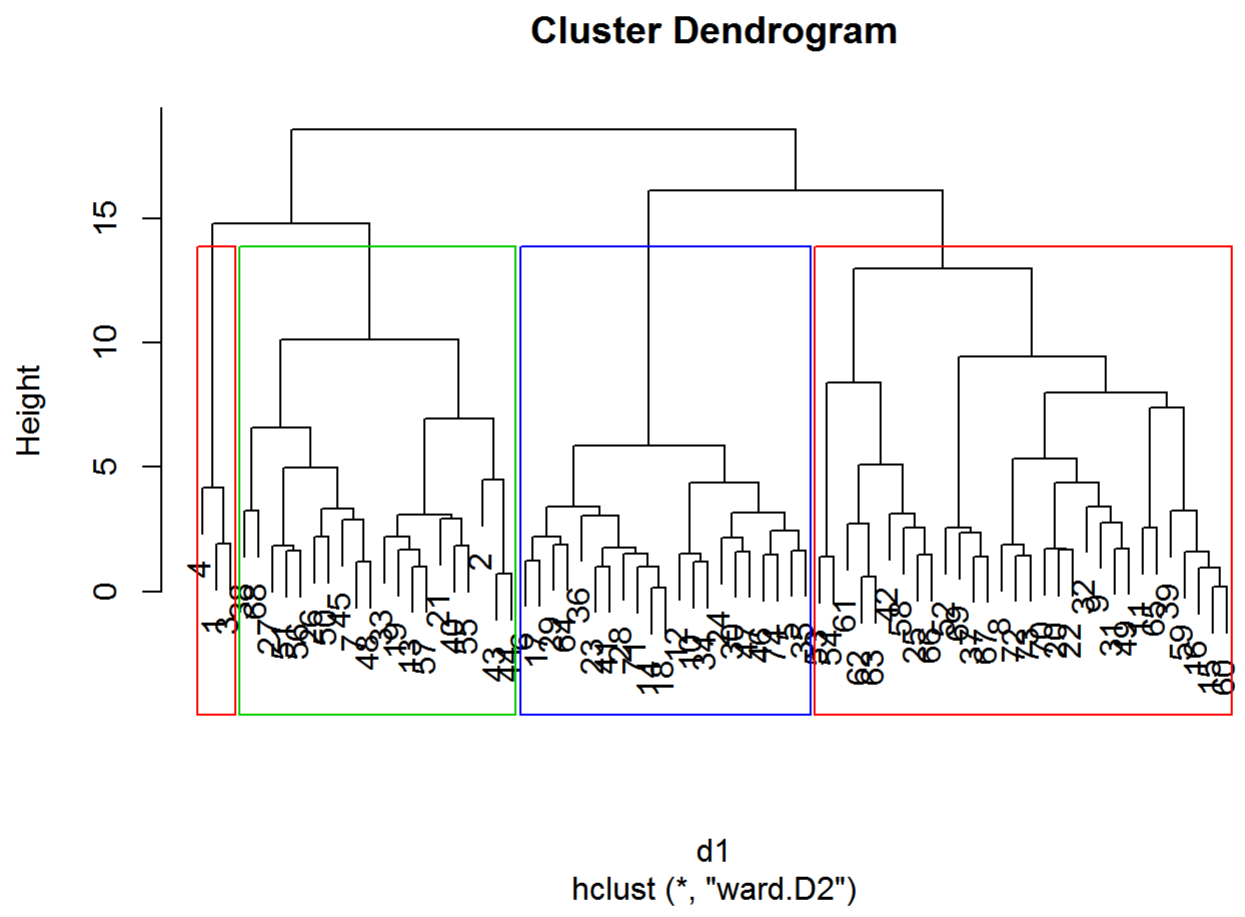
```
# Cut the tree to 4 clusters, using the cutree() function
hc3 <- cutree(hc2, k = 4)

# Number of Cereals in each cluster
table(hc3)
```

```
## hc3
##  1  2  3  4
##  3 20 21 30
```

```
# Store the clusters in a data frame along with the cereals data

cereals_hc <- cbind(hc3, MyData)

# We can also use the cutree output to add the the cluster each observation belongs to to our or
iginal data.

colnames(cereals_hc)[1] <- "cluster"

head(cereals_hc)
```

```
##      cluster   calories    protein        fat     sodium       fiber
## [1,]       1 -1.8659155  1.3817478  0.0000000 -0.3910227  3.22866747
## [2,]       2  0.6537514  0.4522084  3.9728810 -1.7804186 -0.07249167
## [3,]       1 -1.8659155  1.3817478  0.0000000  1.1795987  2.81602258
## [4,]       1 -2.8737823  1.3817478 -0.9932203 -0.2702057  4.87924705
## [5,]       3  0.1498180 -0.4773310  0.9932203  0.2130625 -0.27881412
## [6,]       3  0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656
##           carbo     sugars     potass   vitamins      shelf     weight
## [1,] -2.5001396 -0.2542051  2.5605229 -0.1818422  0.9419715 -0.2008324
## [2,] -1.7292632  0.2046041  0.5147738 -1.3032024  0.9419715 -0.2008324
## [3,] -1.9862220 -0.4836096  3.1248675 -0.1818422  0.9419715 -0.2008324
## [4,] -1.7292632 -1.6306324  3.2659536 -0.1818422  0.9419715 -0.2008324
## [5,] -1.0868662  0.6634132 -0.4022862 -0.1818422 -1.4616799 -0.2008324
## [6,] -0.9583868  1.5810314 -0.9666308 -0.1818422 -0.2598542 -0.2008324
##            cups     rating
## [1,] -2.0856582  1.8549038
## [2,]  0.7567534 -0.5977113
## [3,] -2.0856582  1.2151965
## [4,] -1.3644493  3.6578436
## [5,] -0.3038480 -0.9165248
## [6,]  0.7567534 -0.6553998
```

```
plot(hc2)

rect.hclust(hc2, k = 4, border = 2:4)
```



Based on the dendogram above the optimal value of K is 4 since 4 boxes are cut properly on the graph.

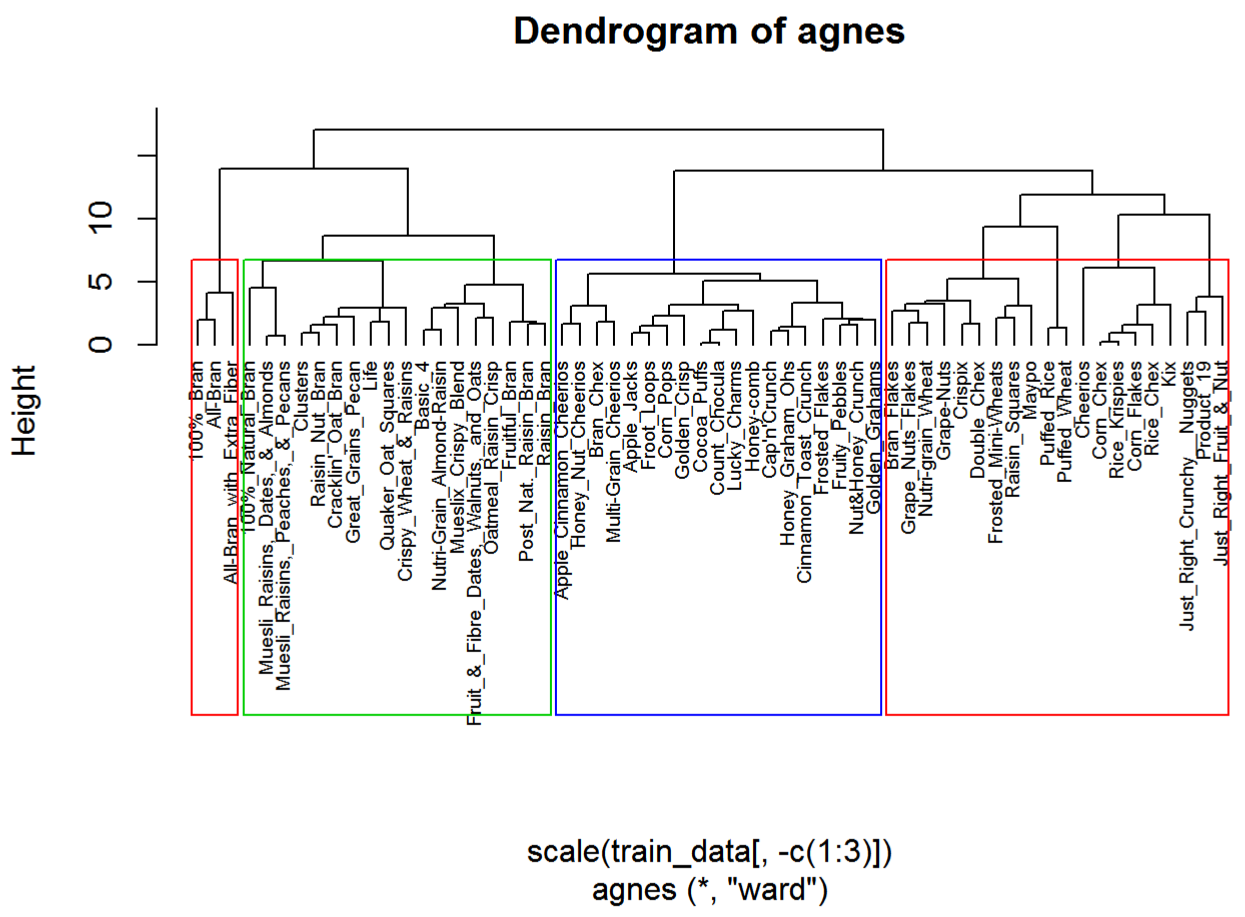
c. Comment on the structure of the clusters and on their stability. Hint: To check stability, partition the data and see how well clusters formed based on one part apply to the other part. To do this:

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

```
# Checking the stability of the cluster
newdata<-Cereals
newdata1<-na.omit(newdata)
#newdata_index<-createDataPartition(newdata1$calories,p=0.75,list=FALSE)
train_data<-newdata1[1:60,] # Partition A
test_data<-newdata1[61:74,] # Partition B


NormTrain_Data <- scale(train_data[, -c(1:3)])
NormTest_Data <- scale(test_data[, -c(1:3)])

#For Partition A the best mwethod is "ward"
hc11<- agnes(scale(train_data[,-c(1:3)]),method = "ward")
hc12<-agnes(scale(train_data[,-c(1:3)]),method="average")
hc13<-agnes(scale(train_data[,-c(1:3)]),method="complete")
hc14<-agnes(scale(train_data[,-c(1:3)]),method="single")
kable(cbind(ward=hc11$ac,average=hc12$ac,complete=hc13$ac,single=hc14$ac))
```

ward	average	complete	single
0.8898441	0.7567786	0.8207517	0.6672134

```
pltree(hc11,cex=0.6,hang=-1,main="Dendrogram of agnes",labels = train_data$name)
rect.hclust(hc11, k = 4, border = 2:4)
```



```r
clust2<-cutree(hc11, k=4)

result<-as.data.frame(cbind(NormTrain_Data,clust2))

# Determine centroids for all 4 clusters
centroid1<-data.frame(column=seq(1,13,1),mean=rep(0,13))
centroid2<-data.frame(column=seq(1,13,1),mean=rep(0,13))
centroid3<-data.frame(column=seq(1,13,1),mean=rep(0,13))
centroid4<-data.frame(column=seq(1,13,1),mean=rep(0,13))
for(i in 1:13)
{
  centroid1[i,2]<-mean(result[result$clust2==1,i])
  centroid2[i,2]<-mean(result[result$clust2==2,i])
  centroid3[i,2]<-mean(result[result$clust2==3,i])
  centroid4[i,2]<-mean(result[result$clust2==4,i])
}
centroidResult<-t(cbind(centroid1$mean,centroid2$mean,centroid3$mean,centroid4$mean))
colnames(centroidResult)<-colnames(newdata1[,-c(1:3)])

centroidResult
```

```
##        calories     protein         fat      sodium      fiber      carbo
## [1,] -2.1272666  1.42341904 -0.38646082  0.10806859  3.3813952 -1.9394118
## [2,]  0.6878427  0.64132067  0.90174191 -0.20271690  0.3500940 -0.1929553
## [3,]  0.1035748 -0.89817824 -0.07729216  0.14375688 -0.5622176 -0.4005138
## [4,] -0.3983645  0.06256787 -0.68017104  0.02966589 -0.2881871  0.8450596
##          sugars     potass   vitamins      shelf     weight       cups
## [1,] -0.9555926  2.8120550 -0.1416617  0.8630890 -0.2075900 -1.6847616
## [2,]  0.3784256  0.6724075 -0.2203626  0.7227493  0.7327750 -0.5456305
## [3,]  0.8283522 -0.6603432 -0.1416617 -0.8653049 -0.2075900  0.3000279
## [4,] -0.9841787 -0.3996489  0.3541542  0.0421019 -0.4311485  0.4587551
##          rating
## [1,]  2.4671291
## [2,] -0.1548936
## [3,] -0.8629557
## [4,]  0.5891428
```

```r
Dumm1 <- data.frame(data=seq(1,14,1), cluster=rep(0,14))
for(i in 1:14)
{
  R <- as.data.frame(rbind(centroidResult,NormTest_Data[i,]))
  U <- as.matrix(get_dist(R))
  Dumm1[i,2] <- which.min(U[5,-5])

}
Dumm1
```

data <dbl>	cluster <dbl>
1	4

data <dbl>	cluster <dbl>
2	4
3	4
4	3
5	4
6	4
7	4
8	2
9	4
10	4

1-10 of 14 rows Previous **1** 2 Next

```
NewClusterData <- as.data.frame(cereals_hc[61:74,])

cbind(Label1 = Dumm1$cluster, Label2 = NewClusterData$cluster)
```

```
##       Label1 Label2
##  [1,]      4      4
##  [2,]      4      4
##  [3,]      4      4
##  [4,]      3      3
##  [5,]      4      4
##  [6,]      4      4
##  [7,]      4      4
##  [8,]      2      2
##  [9,]      4      4
## [10,]      4      4
## [11,]      3      3
## [12,]      2      4
## [13,]      4      4
## [14,]      3      3
```

```
table(Dumm1$cluster == NewClusterData$cluster)
```

```
##
## FALSE  TRUE
##     1    13
```

Out of 14 rows 13 are True. Accuarcy is 92%. Hence stability of cluster is 92%.

d. The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of “healthy cereals.” Should the data be normalized? If not, how should they be used in the cluster analysis?

```
MyDataResult<-cbind(newdata1,hc3)
MyDataResult[MyDataResult$hc3==1,]
```

	name	…	type	calories	protein	fat	sodium	fiber	carbo
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
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

3 rows | 1-10 of 18 columns

```
MyDataResult[MyDataResult$hc3==2,]
```

	name	…	t…	calories	protein	…	sodi…	fiber
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2	100%_Natural_Bran	Q	C	120	3	5	15	2.0
7	Basic_4	G	C	130	3	2	210	2.0
13	Clusters	G	C	110	3	2	140	2.0
19	Cracklin'_Oat_Bran	K	C	110	3	3	140	4.0
21	Crispy_Wheat_&_Raisins	G	C	100	2	1	140	2.0
26	Fruit_&_Fibre_Dates,_Walnuts,_and_Oats	P	C	120	3	2	160	5.0
27	Fruitful_Bran	K	C	120	3	0	240	5.0
33	Great_Grains_Pecan	P	C	120	3	3	75	3.0
38	Just_Right_Fruit_&_Nut	K	C	140	3	1	170	2.0
40	Life	Q	C	100	4	2	150	2.0

1-10 of 20 rows | 1-10 of 18 columns Previous **1** 2 Next

```
MyDataResult[MyDataResult$hc3==3,]
```

	name <chr>	… <chr>	type <chr>	calories <dbl>	protein <dbl>	fat <dbl>	sodium <dbl>	fiber <dbl>	carbo <dbl>
5	Apple_Cinnamon_Cheerios	G	C	110	2	2	180	1.5	10.5
6	Apple_Jacks	K	C	110	2	0	125	1.0	11.0
10	Cap'n'Crunch	Q	C	120	1	2	220	0.0	12.0
12	Cinnamon_Toast_Crunch	G	C	120	1	3	210	0.0	13.0
14	Cocoa_Puffs	G	C	110	1	1	180	0.0	12.0
17	Corn_Pops	K	C	110	1	0	90	1.0	13.0
18	Count_Chocula	G	C	110	1	1	180	0.0	12.0
23	Froot_Loops	K	C	110	2	1	125	1.0	11.0
24	Frosted_Flakes	K	C	110	1	0	200	1.0	14.0
28	Fruity_Pebbles	P	C	110	1	1	135	0.0	13.0

1-10 of 21 rows | 1-10 of 18 columns Previous **1** 2 3 Next

```
MyDataResult[MyDataResult$hc3==4,]
```

	name <chr>	… <chr>	ty… <chr>	calories <dbl>	protein <dbl>	fat <dbl>	sodi… <dbl>	fiber <dbl>	carbo <dbl>
8	Bran_Chex	R	C	90	2	1	200	4	15
9	Bran_Flakes	P	C	90	3	0	210	5	13
11	Cheerios	G	C	110	6	2	290	2	17
15	Corn_Chex	R	C	110	2	0	280	0	22
16	Corn_Flakes	K	C	100	2	0	290	1	21
20	Crispix	K	C	110	2	0	220	1	21
22	Double_Chex	R	C	100	2	0	190	1	18
25	Frosted_Mini-Wheats	K	C	100	3	0	0	3	14
31	Grape_Nuts_Flakes	P	C	100	3	1	140	3	15
32	Grape-Nuts	P	C	110	3	0	170	3	17

1-10 of 30 rows | 1-10 of 18 columns Previous **1** 2 3 Next

From the above Cluster 1 has highest ratings. So Cluster 1 is a cluster of “Healthy Cereals”