MIS 64060: Assignment_5: Hierarchical Clustering

Eyob Tadele

11/13/2021

Project Objective

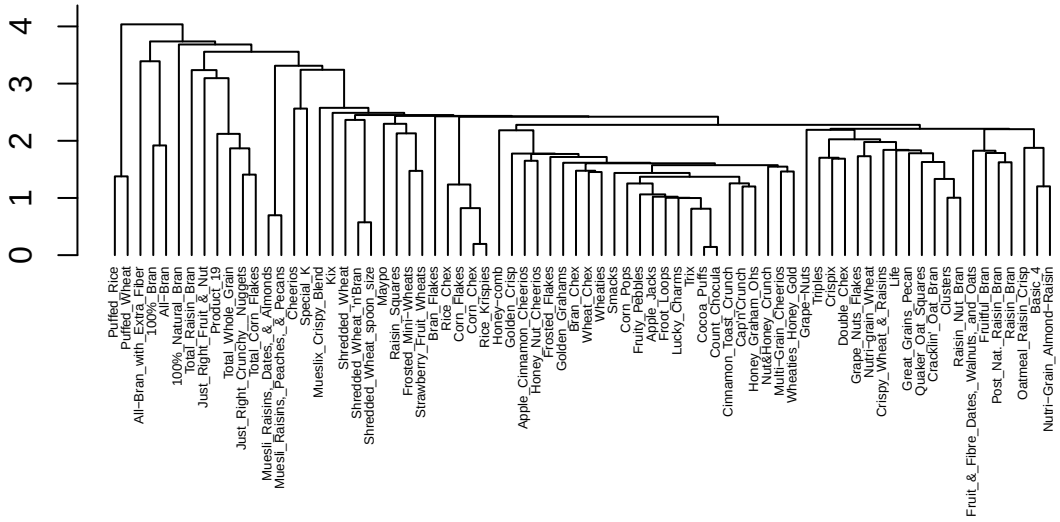
To use Hierarchical Clustering on the Cereals.csv dataset. This dataset includes nutritional information, store display, and consumer ratings for 77 breakfast cereals.

Load relevant libraries, read data and preprocess by removing all cereals with missing values.

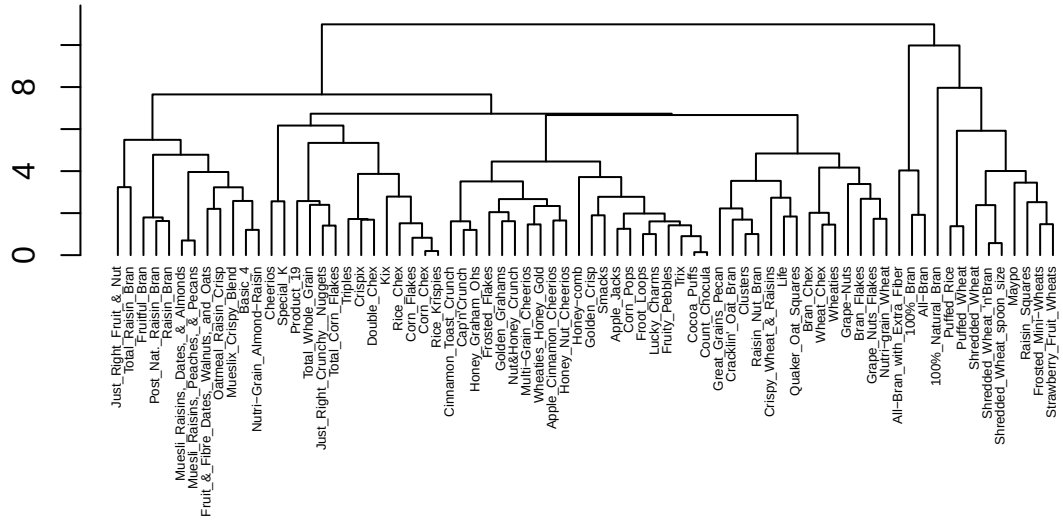
```
library(tidyverse)
library(factoextra)
library(cluster)
library(caret)
library(dplyr)

cereals.df <- read.csv("Cereals.csv")
# setting the row names to the brand name
rownames(cereals.df) <- cereals.df$name
# removing all cereals with missing values
cereals.df <- na.omit(cereals.df)
# de-selecting non-numeric variables before normalizing
cereals.df <- cereals.df[,-c(1,2,3)]
# Scaling the dataset using z-score
cereal.df.norm <- scale(cereals.df)
```

Question 1: 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.

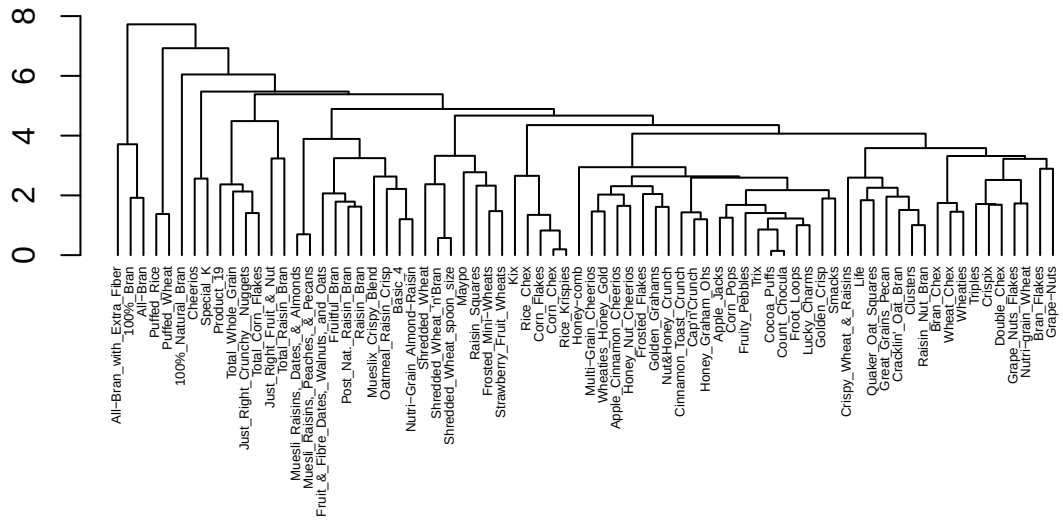
```
set.seed(420)
# computing distance based on the normalized data and the 13 variables selected
dist.norm <- dist(cereal.df.norm, method = "euclidean")

# single linkage: uses the minimum pairwise similarity between all observations
hc_single <- hclust(dist.norm, method = "single")
plot(hc_single, cex = 0.4, hang = -1, ann = FALSE)
```



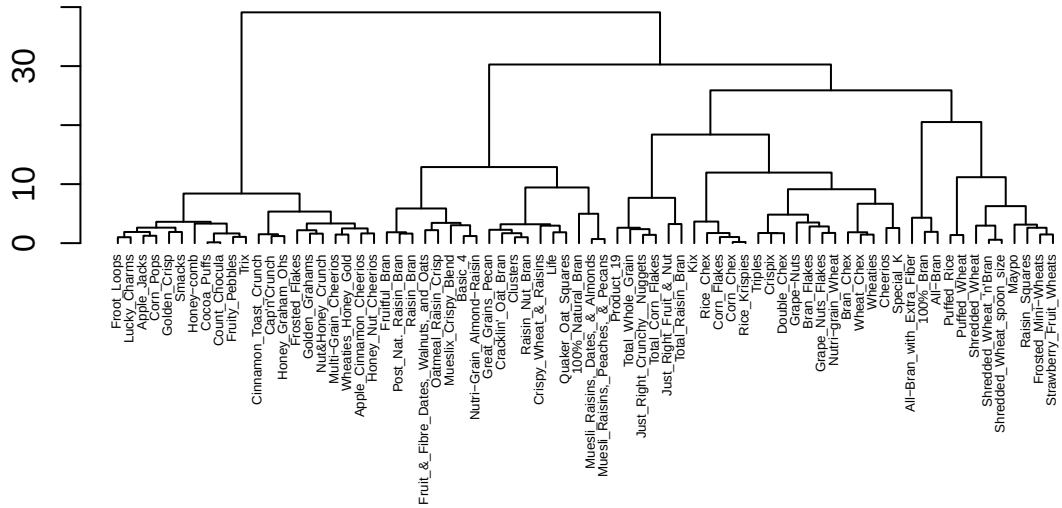

```r
# complete linkage: uses the maximum pairwise similarity between all observations
hc_comp <- hclust(dist.norm, method = "complete")
plot(hc_comp, cex = 0.4, hang = -1, ann = FALSE)
```




```r
# average linkage: uses average pairwise similarity before merging
hc_avg <- hclust(dist.norm, method = "average")
plot(hc_avg, cex = 0.4, hang = -1, ann = FALSE)
```




```
# ward: uses the Error Sum of Squares(SSE)
hc_ward <- hclust(dist.norm, method = "ward.D")
plot(hc_ward, cex = 0.4, hang = -1, ann = FALSE)
```




```
# using Agnes function to choose the best method via the agglomerative coefficient values
single.hc <- agnes(dist.norm, method = "single")
comp.hc <- agnes(dist.norm, method = "complete")
avg.hc <- agnes(dist.norm, method = "average")
ward.hc <- agnes(dist.norm, method = "ward")
#printing the Agglomerative coefficient values
single.hc$ac
```

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

```
comp.hc$ac
```

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

```
avg.hc$ac
```

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

```
ward.hc$ac
```

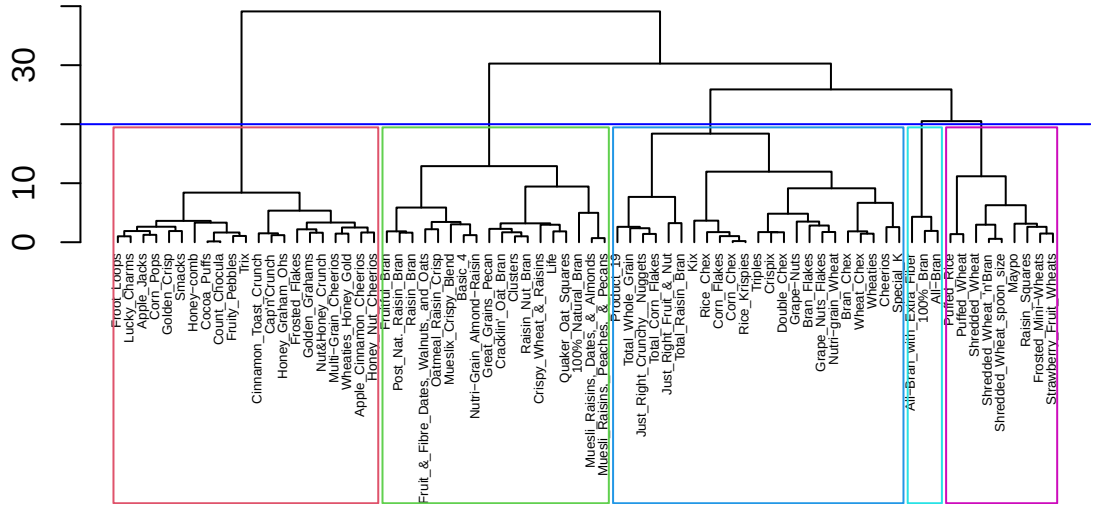
```
## [1] 0.9046042
```

We can see from the agglomerative coefficinet values above that the Ward method give is closest to 1. This indicates that the Ward method gives us the strongest clustering structure among the alternatives.

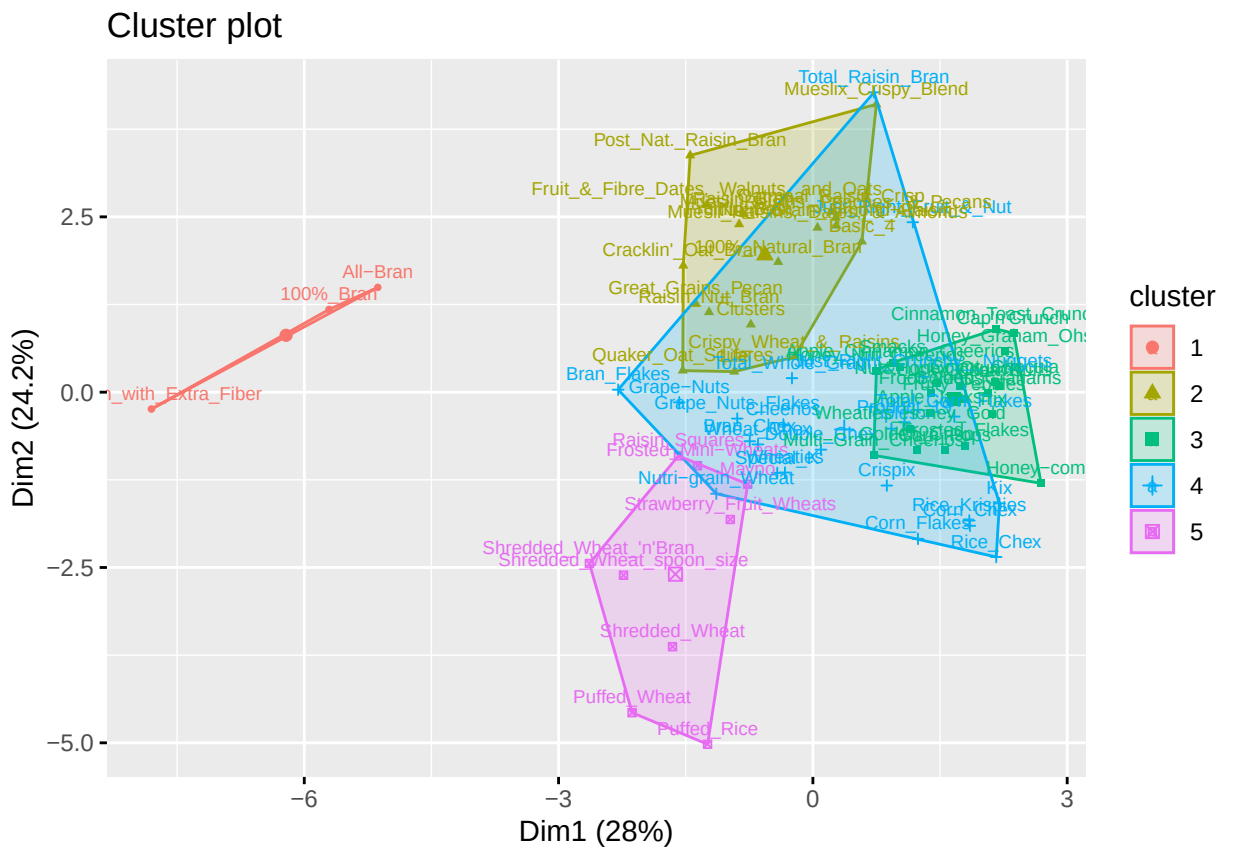
Question 2: How many clusters would you choose?

Looking at the dendrogram with Ward method, we can choose the number of clusters by first finding the largest vertical difference between the nodes and then drawing a horizontal line halfway through it. The number of vertical lines intersecting it will be the "optimal" number of clusters.

```
# re-plotting the dendrogram using ward method
plot(hc_ward, cex = 0.4, hang = -1, ann = FALSE)
# drawing a horizontal line from the halfway point of the longest difference
abline(h=20, col = "blue")
# visualizing the clusters inside the dendrogram by putting borders
rect.hclust(hc_ward, k=5, border = 2:6)
```




```
# cuts the tree into different clusters based on the height(i.e. 20 in this case)
cereal.clusters <- cutree(hc_ward, h=20)
# visualizing the clusters
fviz_cluster(list(data=cereal.df.norm, cluster = cereal.clusters), pointsize = 1, labelsize = 7)
```



We can observe from the result of the dendrogram that there are 5 clusters chosen. The clusters are also visualized using the fviz_cluster() function as seen above.

```
# save the clusters as a list into a temp variable
temp <- as.list(cutree(hc_ward, h=20))
# convert temp into a dataframe and transpose it, so as to match rows in hc_ward
df <- t(as.data.frame(temp))
df1 <- as.data.frame(df)
# copying dataframe into a new one
cereal.df.norm2 <- as.data.frame(cereal.df.norm)
cereal.df.norm2$cluster <- df1[,c(1)]
head(cereal.df.norm2)
```

```
##                             calories    protein        fat     sodium
## 100%_Bran                 -1.8659155  1.3817478  0.0000000 -0.3910227
## 100%_Natural_Bran          0.6537514  0.4522084  3.9728810 -1.7804186
## All-Bran                  -1.8659155  1.3817478  0.0000000  1.1795987
## All-Bran_with_Extra_Fiber -2.8737823  1.3817478 -0.9932203 -0.2702057
## Apple_Cinnamon_Cheerios    0.1498180 -0.4773310  0.9932203  0.2130625
## Apple_Jacks                0.1498180 -0.4773310 -0.9932203 -0.4514312
##                                 fiber      carbo     sugars     potass
## 100%_Bran                  3.22866747 -2.5001396 -0.2542051  2.5605229
## 100%_Natural_Bran         -0.07249167 -1.7292632  0.2046041  0.5147738
## All-Bran                   2.81602258 -1.9862220 -0.4836096  3.1248675
```

```
## All-Bran_with_Extra_Fiber  4.87924705 -1.7292632 -1.6306324  3.2659536
## Apple_Cinnamon_Cheerios   -0.27881412 -1.0868662  0.6634132 -0.4022862
## Apple_Jacks               -0.48513656 -0.9583868  1.5810314 -0.9666308
##                             vitamins      shelf     weight       cups
## 100%_Bran                 -0.1818422  0.9419715 -0.2008324 -2.0856582
## 100%_Natural_Bran         -1.3032024  0.9419715 -0.2008324  0.7567534
## All-Bran                  -0.1818422  0.9419715 -0.2008324 -2.0856582
## All-Bran_with_Extra_Fiber -0.1818422  0.9419715 -0.2008324 -1.3644493
## Apple_Cinnamon_Cheerios   -0.1818422 -1.4616799 -0.2008324 -0.3038480
## Apple_Jacks               -0.1818422 -0.2598542 -0.2008324  0.7567534
##                               rating cluster
## 100%_Bran                  1.8549038       1
## 100%_Natural_Bran         -0.5977113       2
## All-Bran                   1.2151965       1
## All-Bran_with_Extra_Fiber  3.6578436       1
## Apple_Cinnamon_Cheerios   -0.9165248       3
## Apple_Jacks               -0.6553998       3
```

```
str(cereal.df.norm2)
```

```
## 'data.frame':    74 obs. of  14 variables:
##  $ calories: num  -1.866 0.654 -1.866 -2.874 0.15 ...
##  $ protein : num  1.382 0.452 1.382 1.382 -0.477 ...
##  $ fat     : num  0 3.973 0 -0.993 0.993 ...
##  $ sodium  : num  -0.391 -1.78 1.18 -0.27 0.213 ...
##  $ fiber   : num  3.2287 -0.0725 2.816 4.8792 -0.2788 ...
##  $ carbo   : num  -2.5 -1.73 -1.99 -1.73 -1.09 ...
##  $ sugars  : num  -0.254 0.205 -0.484 -1.631 0.663 ...
##  $ potass  : num  2.561 0.515 3.125 3.266 -0.402 ...
##  $ vitamins: num  -0.182 -1.303 -0.182 -0.182 -0.182 ...
##  $ shelf   : num  0.942 0.942 0.942 0.942 -1.462 ...
##  $ weight  : num  -0.201 -0.201 -0.201 -0.201 -0.201 ...
##  $ cups    : num  -2.086 0.757 -2.086 -1.364 -0.304 ...
##  $ rating  : num  1.855 -0.598 1.215 3.658 -0.917 ...
##  $ cluster : int  1 2 1 1 3 3 2 4 4 3 ...
```

Question3: 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.

```
set.seed(420)
# Creating a model for partitioning the cereals data into two parts
Partition <- createDataPartition(cereal.df.norm2$cluster, p = 0.7, list = FALSE)
part.a <- cereal.df.norm2[Partition,]
part.b <- cereal.df.norm2[-Partition,]
#Grouping dataset by clusters and computing the mean of the group columns gives us the centroid values
cent.a <- aggregate(part.a[,1:13], list(part.a$cluster),mean)
cent.a
```

```
##   Group.1    calories     protein         fat       sodium      fiber
## 1       1 -1.86591549  1.38174776  0.00000000 -0.391022691  3.2286675
## 2       2  0.79773238  0.51860403  0.85133165  0.070670966  0.4875264
## 3       3  0.14981803 -0.97308540 -0.06621468  0.007673514 -0.6364397
## 4       4 -0.02804081  0.28817199 -0.35054833  0.710544364 -0.1938578
## 5       5 -1.14601066 -0.07895702 -0.99322026 -1.935754741  0.2222547
##        carbo     sugars      potass   vitamins       shelf      weight
## 1 -2.5001396 -0.2542051  2.56052289 -0.1818422  0.94197150 -0.20083243
## 2 -0.2425731  0.5323249  0.81206244 -0.1818422  0.77028211  1.03297834
## 3 -0.6072098  1.0151668 -0.76911015 -0.1818422 -0.66046278 -0.20083243
## 4  0.8856704 -0.7400029 -0.26534962  0.8075933  0.02292831 -0.08580457
## 5  0.2162819 -1.0407350  0.03104985 -0.8226195 -0.43154359 -0.82472166
##          cups     rating
## 1 -2.08565823  1.8549038
## 2 -0.72808850 -0.2752031
## 3  0.46261329 -0.9299165
## 4  0.32502626  0.1948246
## 5 -0.06748537  1.5773856
```

```
#create an empty dataframe and fill values with 0s to store information related to clusters in part.b
clust.b <- as.data.frame(matrix(0,nrow(part.b),1))

# The code below first combines the centroids from A with values from part.b and computes the distance.
# Then using which.min function, we can save the indices that represent the new clusters in clust.b
for (x in 1:nrow(part.b)) {
  clust.b$V1[x] <- which.min(as.matrix(get_dist(as.data.frame(rbind(cent.a[-1],part.b[x,-ncol(part.b)]))
}

#Here we can compute the average similarity of the clusters by comparing part.b with the primary cluste
clust.b$clust0 <- part.b$cluster
mean(clust.b$V1 == clust.b$clust0)
```

```
## [1] 0.95
```

It can be observed from the result that there is a 95% similarity among the clusters, which points to the stability of the clusters.

An alternative approch to answering this question is to use the concept of Cophenetic Correlation value as a measure of similarity between the partitions A and B.

Note: Classifications are generally pictured in the form of hierarchical trees, also called a dendrogram. A dendrogram is the graphical representation of an ultrametric (= cophenetic) matrix; so dendrograms can be compared to one another by comparing their cophenetic matrices. (Reference: Sinan Saracli, Nurhan Dogan, Ismet Dogan: Comparison of hierarchical cluster analysis methods by cophenetic correlation. Journal of Inequalities and Applications volume 2013, Article number: 203 (2013)). Sample code below attempts to implement it, but the resulting dendrograms aren't the same size(even though p=0.5). Thus preventing the calculation of Cophenetic correlation.

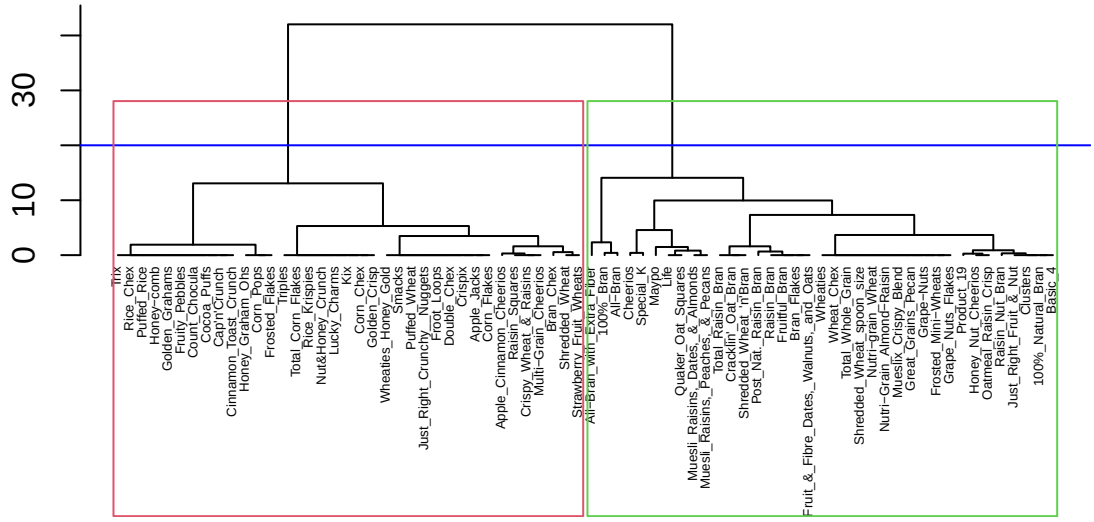
```
# Creating a model for partitioning the cereals data into two parts,
Partition_idx <- createDataPartition(cereal.df.norm2$cluster, p = 0.5, list = FALSE)
part.A <- cereal.df.norm2[Partition_idx,]
part.B <- cereal.df.norm2[-Partition_idx,]
# computing distance based on the normalized data and the 13 variables selected
```

```
dist.a <- dist(part.A, method = "euclidean")
dist.b <- dist(part.B, method = "euclidean")
# Clustering using Ward method
hc_ward_a <- hclust(dist.a, method = "ward.D")
hc_ward_b <- hclust(dist.b, method = "ward.D")
dend.a <- as.dendrogram(hc_ward_a)
dend.b <- as.dendrogram(hc_ward_b)

cor(cophenetic(dend.a), cophenetic(dend.b))
```

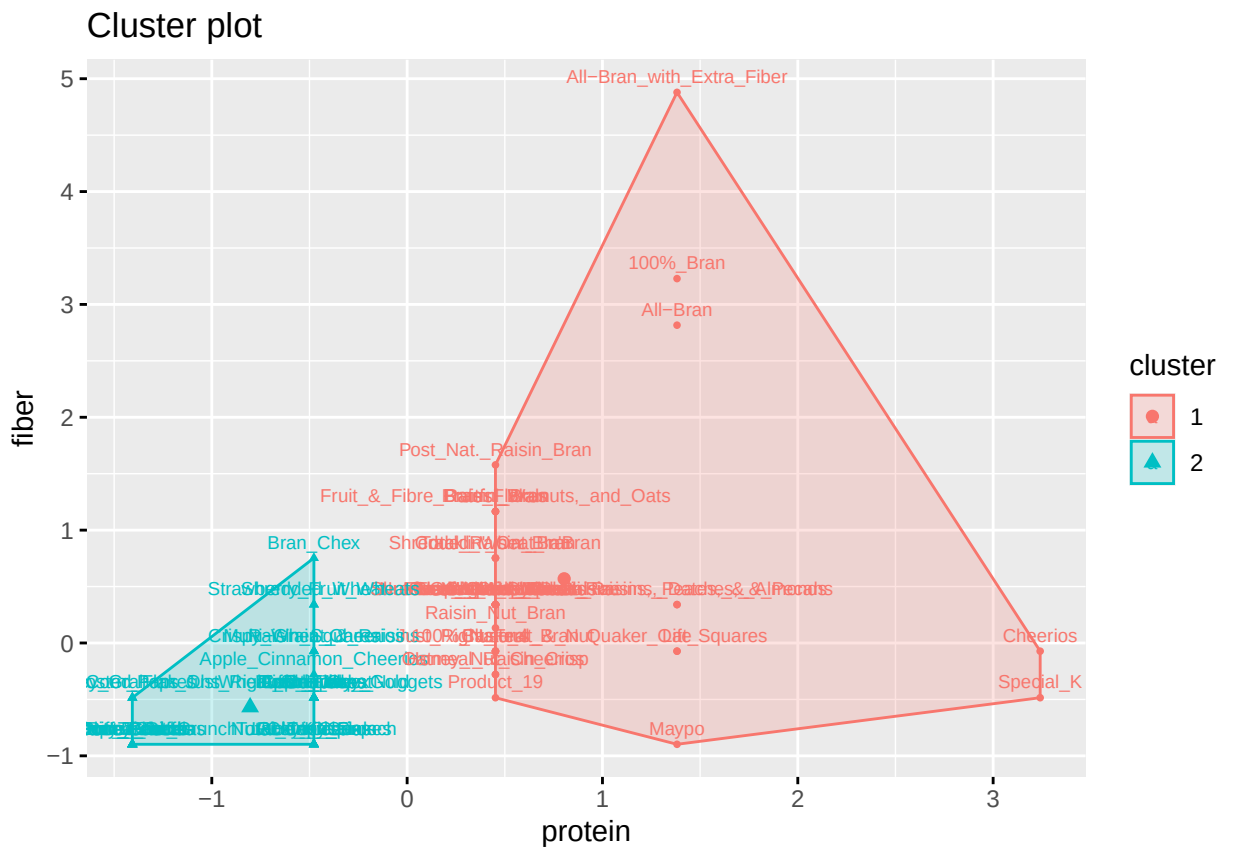
Question 4: The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of "healthy cereals." Should the data be normalized? If not, how should they be used in the cluster analysis?

The assumption taken here regarding what constitutes as a "healthy cereal" would be its protein, fiber and vitamin content. However, the vitamin feature mostly consists of the value 25, skewing the data heavily to one side. As a result, I have selected the protein and fiber features to identify those that support a healthy diet for students. The data should be normalized, as the scale of some of the features may unduly influence our clusters. For example, the range of values for protein is [1-6] and [0-14] for fiber.

```
# selecting features that are considered as healthy
healthy.cereals <- cereals.df[,c(2,5)]
# Scaling the dataset using z-score
healthy.cereals.norm <- scale(healthy.cereals)

set.seed(420)
# computing distance based on the normalized data and the 2 variables selected
dist.h <- dist(healthy.cereals.norm, method = "euclidean")
hc_healthy <- hclust(dist.h, method = "ward.D")
plot(hc_healthy, cex = 0.4, hang = -1, ann = FALSE)
# drawing a horizontal line from the halfway point of the longest difference
abline(h=20, col = "blue")
# visualizing the clusters inside the dendrogram by putting borders
rect.hclust(hc_healthy, k=2, border = 2:6)
```




```
# cuts the tree into different clusters based on the height(i.e. 20 in this case)
healthy.cereal.clusters <- cutree(hc_healthy, h=17)
# visualizing the clusters
fviz_cluster(list(data=healthy.cereals.norm, cluster = healthy.cereal.clusters), pointsize = 1, labelsi
```



We can see from the displayed plot that cereals in Cluster 1 can be considered as healthy, since they contain a relatively higher contents of fiber and protein.