Assignment 05- FML

Chathurani Ekanayake

2023-04-16

```
library(readxl)
Cereals <- read_excel("F:/1st sem/ML/Assignmnet 05/Cereals.xlsx")
View(Cereals)

library(dplyr)
```

```
##
## 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
```

```
# Data processing
str(Cereals) # checking the structure of dataset
```

```
## tibble [77 × 16] (S3: tbl_df/tbl/data.frame)
##  $ name    : chr [1:77] "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-Bran_with_Extra_Fibe
r" ...
##  $ mfr     : chr [1:77] "N" "Q" "K" "K" ...
##  $ type    : chr [1:77] "C" "C" "C" "C" ...
##  $ calories: num [1:77] 70 120 70 50 110 110 110 130 90 90 ...
##  $ protein : num [1:77] 4 3 4 4 2 2 2 3 2 3 ...
##  $ fat     : num [1:77] 1 5 1 0 2 2 0 2 1 0 ...
##  $ sodium  : num [1:77] 130 15 260 140 200 180 125 210 200 210 ...
##  $ fiber   : num [1:77] 10 2 9 14 1 1.5 1 2 4 5 ...
##  $ carbo   : num [1:77] 5 8 7 8 14 10.5 11 18 15 13 ...
##  $ sugars  : num [1:77] 6 8 5 0 8 10 14 8 6 5 ...
##  $ potass  : num [1:77] 280 135 320 330 NA 70 30 100 125 190 ...
##  $ vitamins: num [1:77] 25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : num [1:77] 3 3 3 3 3 1 2 3 1 3 ...
##  $ weight  : num [1:77] 1 1 1 1 1 1 1 1.33 1 1 ...
##  $ cups    : num [1:77] 0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
##  $ rating  : num [1:77] 68.4 34 59.4 93.7 34.4 ...
```

```
t(t(names(Cereals)))
```

```
##       [,1]
##  [1,] "name"
##  [2,] "mfr"
##  [3,] "type"
##  [4,] "calories"
##  [5,] "protein"
##  [6,] "fat"
##  [7,] "sodium"
##  [8,] "fiber"
##  [9,] "carbo"
## [10,] "sugars"
## [11,] "potass"
## [12,] "vitamins"
## [13,] "shelf"
## [14,] "weight"
## [15,] "cups"
## [16,] "rating"
```

```
## Remove rows with missing values
data <- na.omit(Cereals)

## select only numerical columns
df <- data[, c(4:12,14:16)]

## Normalize data using scale function
df1 <- (scale(df))

##  Set row names to the Cereals column
row.names(df1) <- data$name
head(df1)
```

```
##                              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     weight       cups     rating
## 100%_Bran                  -0.1818422 -0.2008324 -2.0856582  1.8549038
## 100%_Natural_Bran          -1.3032024 -0.2008324  0.7567534 -0.5977113
## All-Bran                   -0.1818422 -0.2008324 -2.0856582  1.2151965
## All-Bran_with_Extra_Fiber -0.1818422 -0.2008324 -1.3644493  3.6578436
## Apple_Cinnamon_Cheerios   -0.1818422 -0.2008324 -0.3038480 -0.9165248
## Apple_Jacks                -0.1818422 -0.2008324  0.7567534 -0.6553998
```

01. 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.

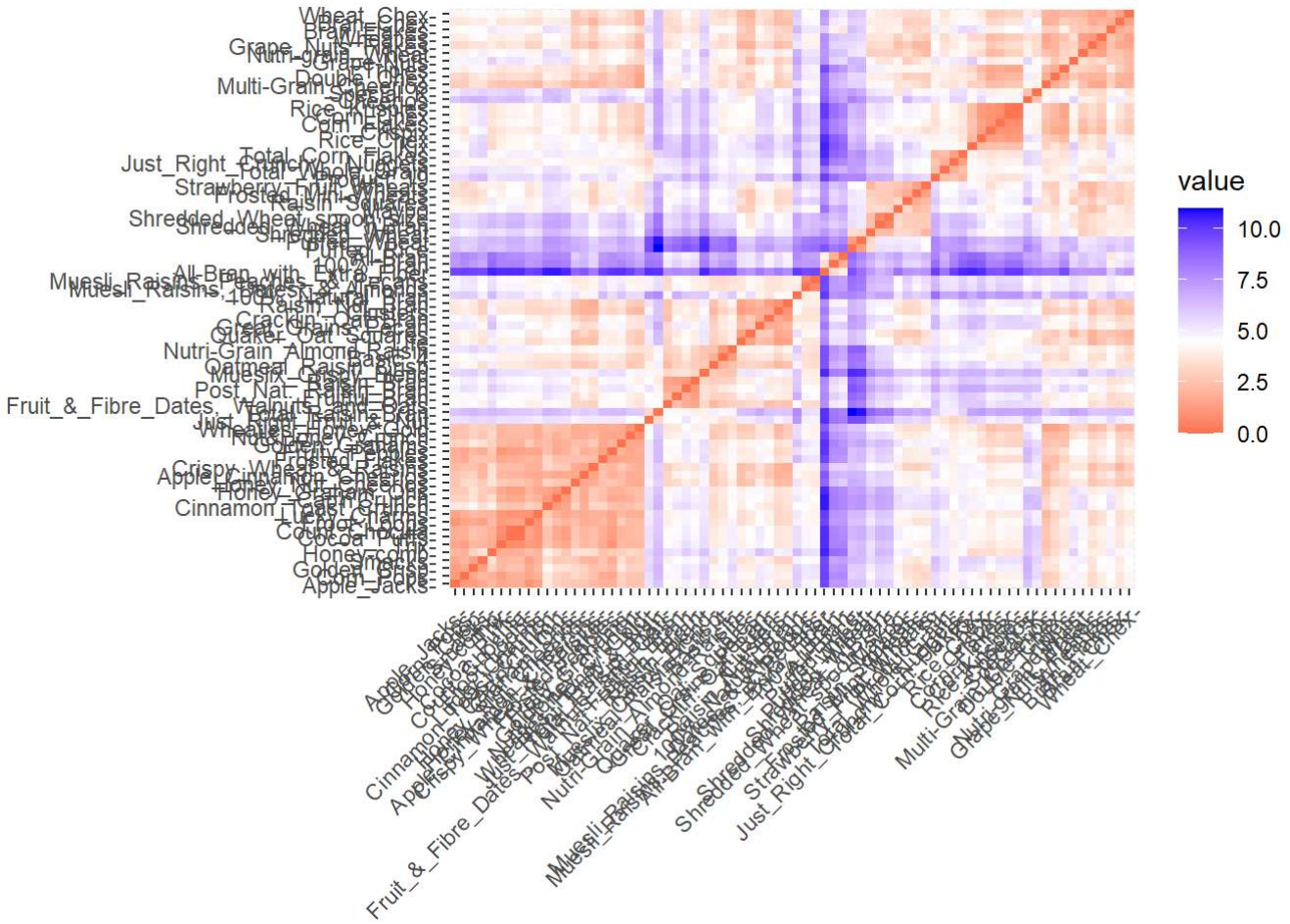
Apply euclidean distance to normalized

```
library(cluster)
library(factoextra)
```

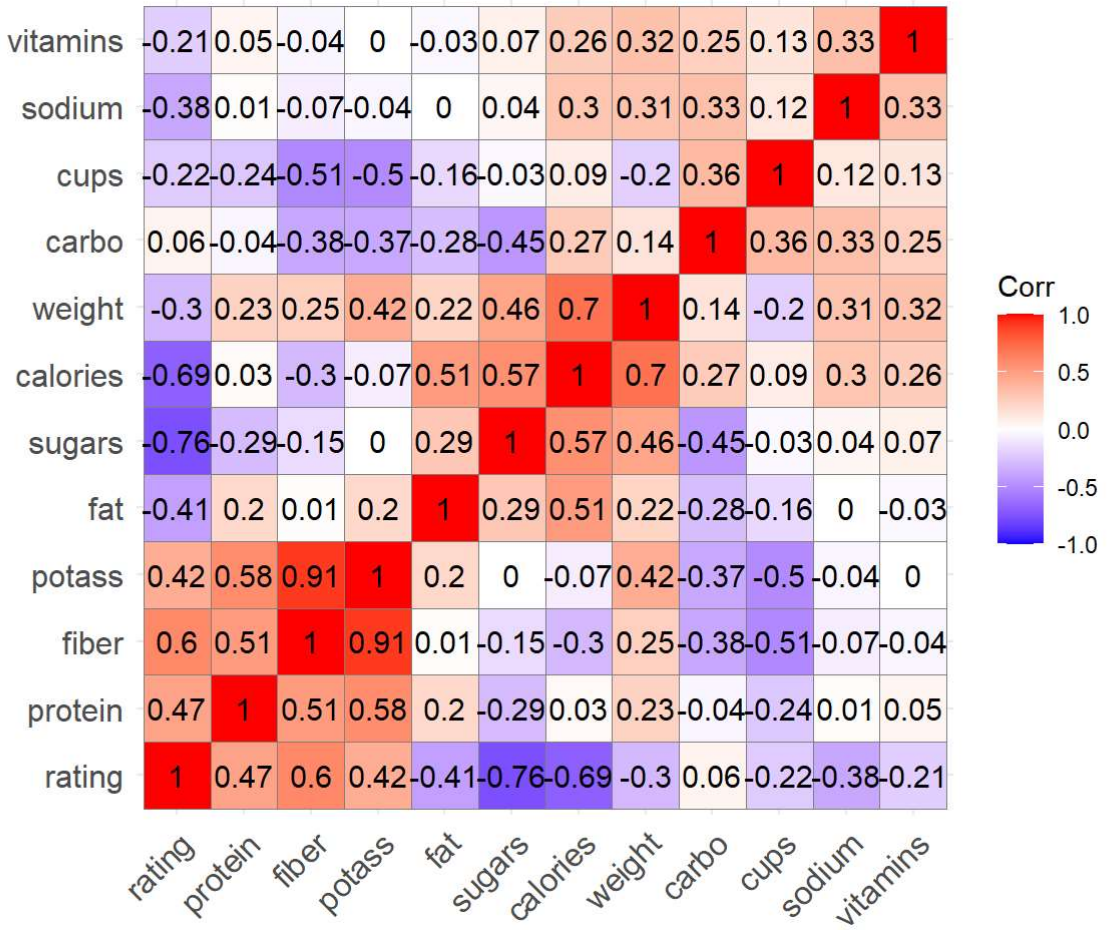
```
## Loading required package: ggplot2
```

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

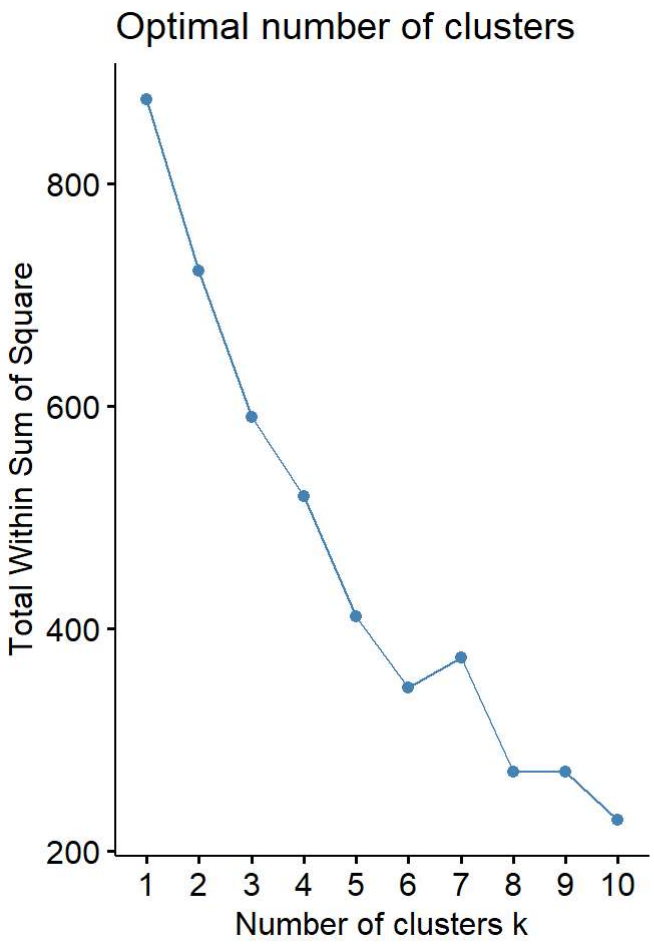
```
distance1 <- dist(df1,method = "euclidean")
fviz_dist(distance1)
```

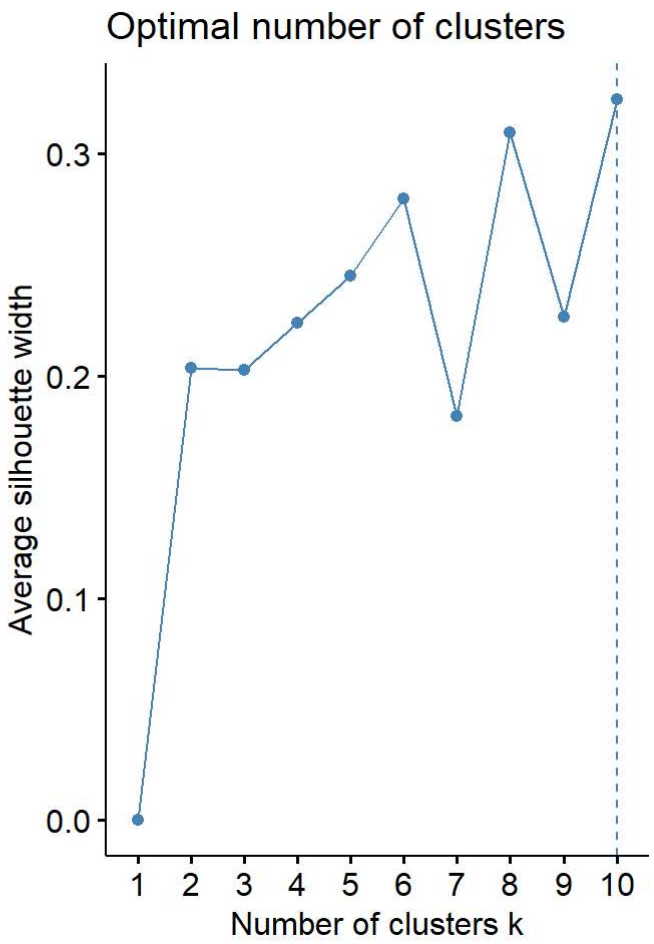



```
# The resulting visualization is a heatmap that shows the pairwise distances between all observa
tions in df1. The colors in the heatmap represent the distance values, where lighter colors indi
cate shorter distances and darker colors indicate longer distances.The diagonal of the heatmap i
s always zero, since the distance between an observation and itself is always zero. The heatmap
is symmetric around the diagonal.

## Correlation matrix to determine the relationships among variables.
library(ggcorrplot)
corr <- cor(df1)
ggcorrplot(corr, outline.color = "grey50", lab = TRUE, hc.order = TRUE, type = "full")
```




```
# According to resulting plot, we can identify the variables which has negative, positive and ze
ro correlation.There is any correlation between potass and sugar, sodium and fat. As well as the
re is a strong positive correlation between potass and fiber. There is a strong negative correla
tion between sugars and rating.

## Checking the value for K - Number of clusters
library(cowplot)
library(dplyr)
library(factoextra)

elbow_method <- fviz_nbclust(df1, kmeans, method = "wss")
silhouette <- fviz_nbclust(df1, kmeans, method = "silhouette")
plot_grid(elbow_method,silhouette, nrow = 1)
```

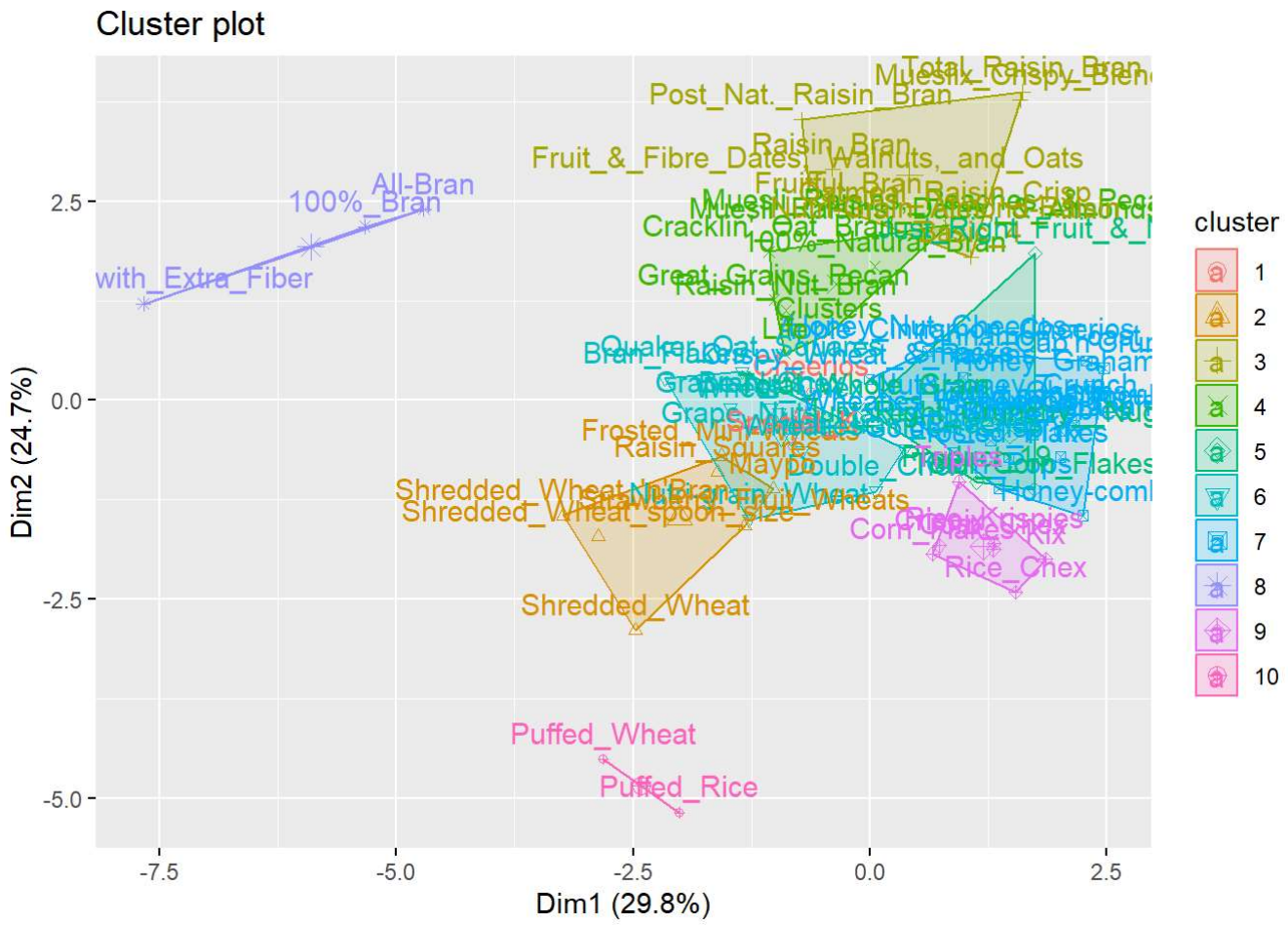




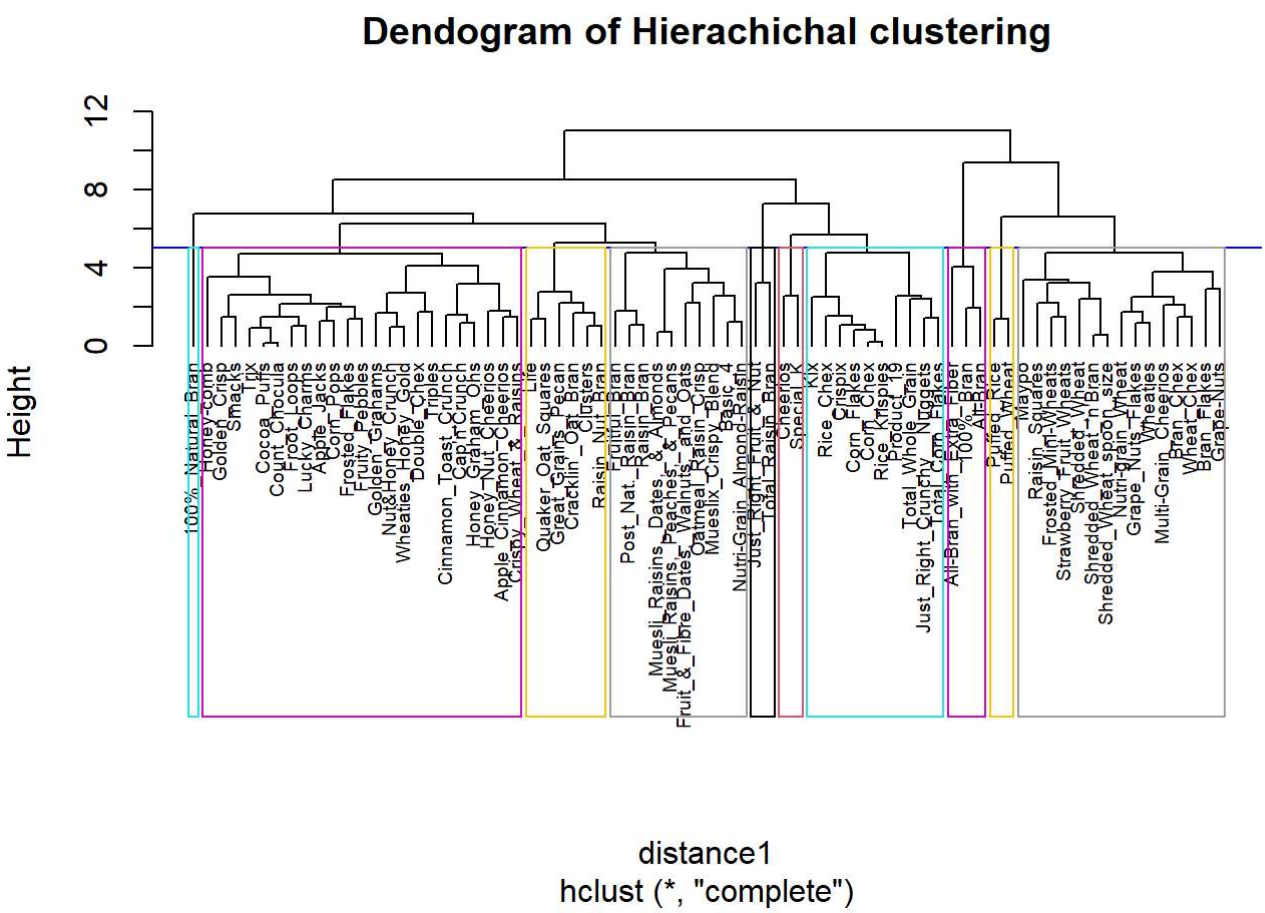

```
# As it shows by the graphs of Silhouette and WSS, we can come identify the best value for K as 10. Simply it means the number of clusters = 10 based on silhouette. The following cluster plot makes it more clear.

k10 <- kmeans(df1,centers = 10, nstart = 25)
k10$centers
```

```
##       calories    protein           fat      sodium      fiber       carbo
## 1   0.14981803  3.2408266  0.000000e+00  1.17959872 -0.2788141  0.45488650
## 2  -0.78605825  0.1866257 -8.513317e-01 -1.93575474  0.1633054  0.43653231
## 3   1.21367739  0.4522084  4.414312e-01  0.38757596  0.6840240  0.08372381
## 4   0.65375141  0.8007856  1.862288e+00 -0.59490143  0.2112017 -0.62112842
## 5   0.25060471  0.0803926 -1.986441e-01  0.59967697 -0.3200786  1.04589172
## 6  -0.45490203  0.2663005 -3.972881e-01  0.29159354  0.2988887  0.30071123
## 7   0.19781169 -0.9199689  1.057355e-17  0.07498586 -0.6619844 -0.59130285
## 8  -2.20187108  1.3817478 -3.310734e-01  0.17279012  3.6413124 -2.07187492
## 9   0.07782755 -0.6101224 -7.094430e-01  1.19685830 -0.7798829  1.75803465
## 10 -2.87378226 -0.9421007 -9.932203e-01 -1.96164410 -0.6914590 -0.82990744
##        sugars     potass   vitamins     weight        cups      rating
## 1  -1.1718233 -0.2612000 -0.1818422 -0.2008324  1.28705407  0.68238355
## 2  -0.9424187  0.1217481 -0.6624252 -0.3591327 -0.06748537  1.49437400
## 3   0.9183071  1.1653377  0.1919445  2.0805535 -0.49239931 -0.43724782
## 4   0.1472529  0.5059559 -0.3220122 -0.2008324 -0.56899829 -0.19615104
## 5  -0.5294905 -0.4163948  3.1822385  0.1902623  0.54463313 -0.16904450
## 6  -0.6212523  0.1408955 -0.1818422 -0.2008324 -0.35051441  0.56389406
## 7   1.0020580 -0.7214096 -0.1818422 -0.2008324  0.22746282 -0.97698099
## 8  -0.7894824  2.9837813 -0.1818422 -0.2008324 -1.84525525  2.24264794
## 9  -1.0079629 -0.8759325 -0.1818422 -0.2008324  0.98705540 -0.01518936
## 10 -1.6306324 -0.9313592 -1.3032024 -3.4599552  0.75675340  1.39015899
```

```
## cluster plot
fviz_cluster(k10,data = df1)
```




```
# According to the cluster plot, there are 10 clusters. But some of the clusters are overlappin
g.

## Apply hierarchical clustering.
set.seed(123)
hc <- hclust(distance1, method = "complete")
plot(hc,cex=0.6, hang= -1, main = "Dendogram of Hierachichal clustering")

## Draw a horizontal line at the height that would result 10 clusters
abline(h=5, col="blue")
clusters <-cutree(hc, k=10)

## draw rectangles around the clusters
rect.hclust(hc, k=10, border = 5:10)
```




```
# The dendogram shows clearly the number of clusters and their elements for each 10 clusters.

## Use AGNES to perform hierarchical clustering
hc_single <- agnes(distance1, method = "single")
hc_complete <- agnes(distance1, method = "complete")
hc_average <- agnes(distance1, method = "average")
hc_ward <- agnes(distance1, method = "ward")

## compare Agglomerative coefficients
print(hc_single$ac)
```

```
## [1] 0.6072384
```

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

```
## [1] 0.8469328
```

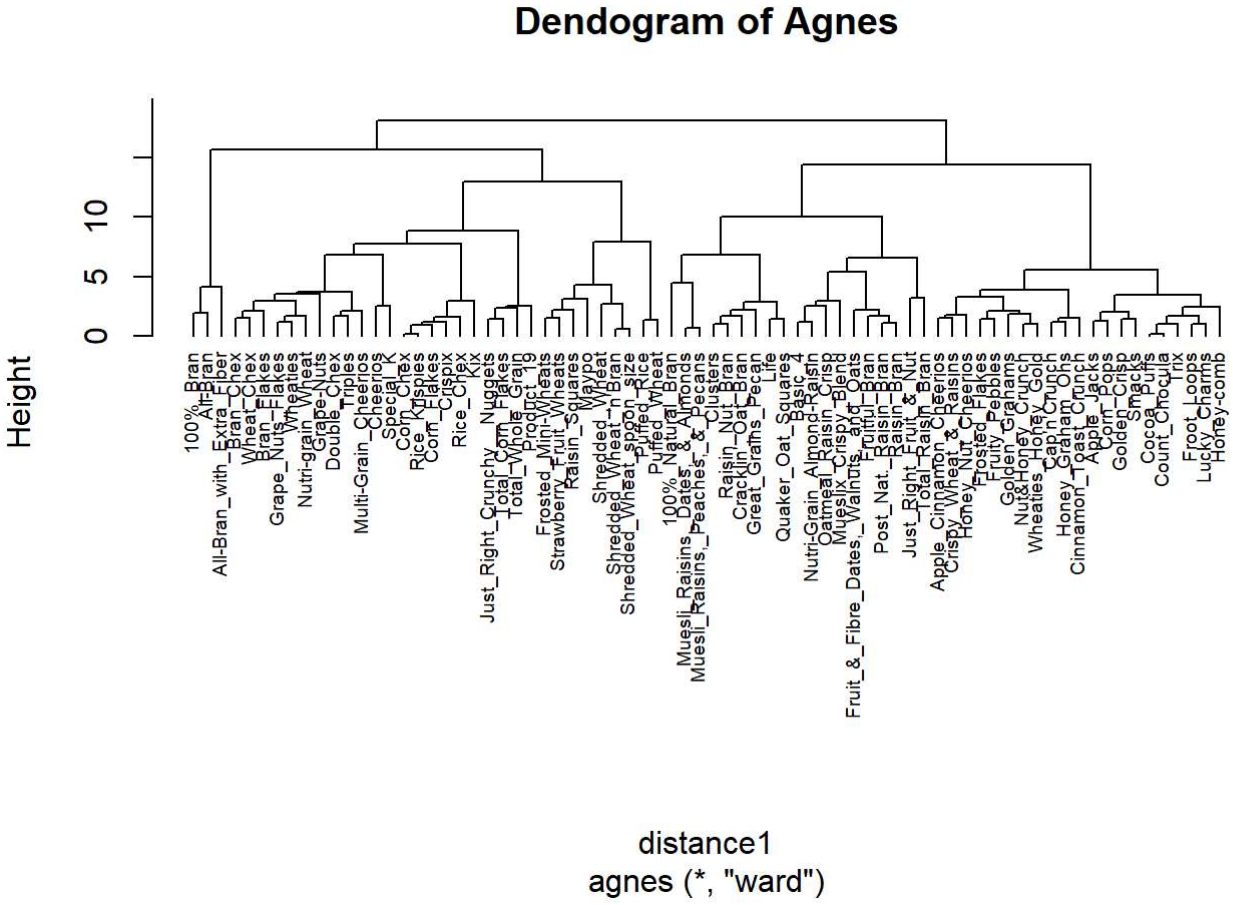
```
print(hc_average$ac)
```

```
## [1] 0.7881955
```

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

```
## [1] 0.9087265
```

```
# According to the above comparison,the highest value is generated by the "ward" method.Therefor
e "ward" can be identifies as the best linkage method which shows the 90% accuracy.

pltree(hc_ward, cex=0.6, hang= -1, main = "Dendogram of Agnes")
```

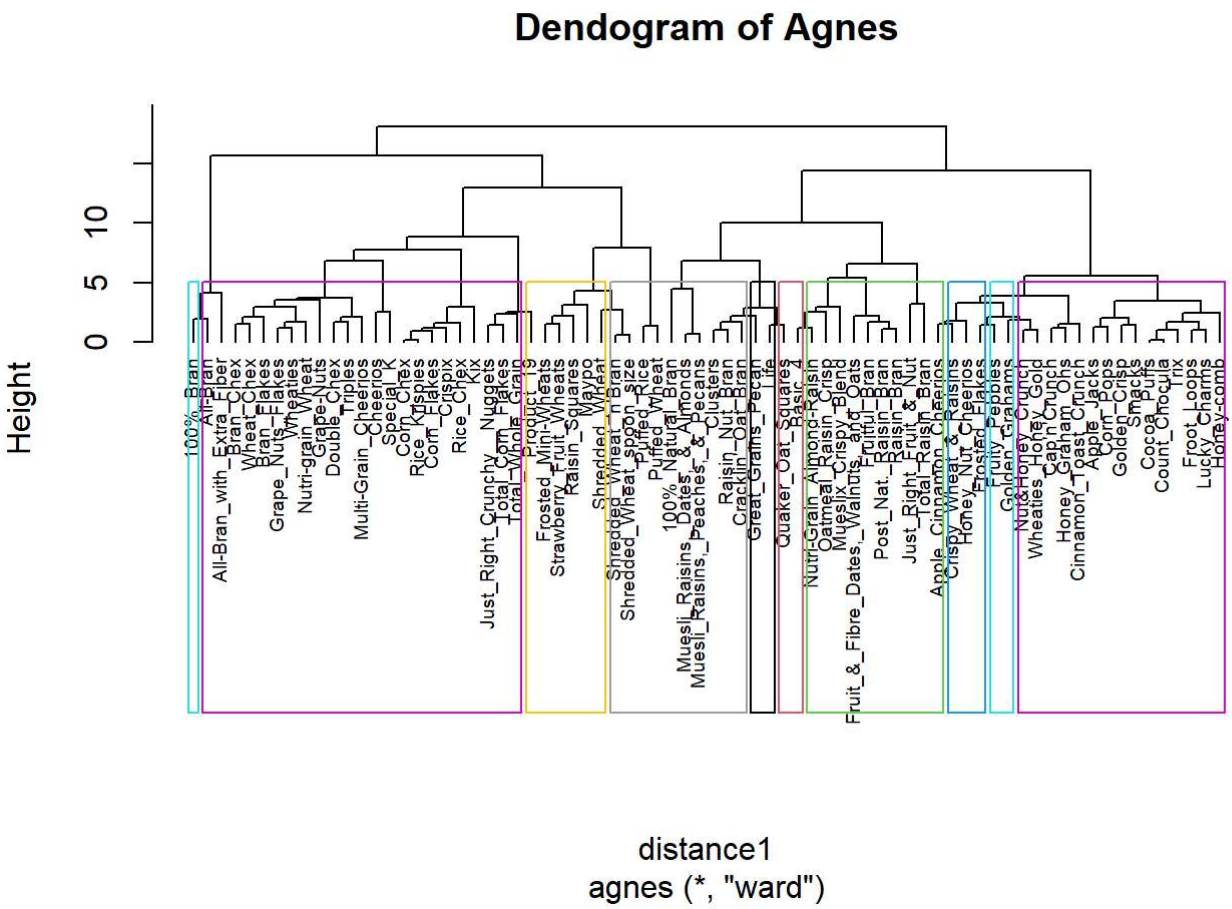


02. How many clusters would you choose?

As suggested by above methods we can select k= 10. But it may be vary based the requirement of analysis.

library(stats) library(graphics)

```
pltree(hc_ward, cex=0.6, hang= -1, main = "Dendogram of Agnes")
rect.hclust(hc, k=10, border = 5:12)
```




```
Cereals_data<-cutree(hc_ward, k=10)
clustered_d <- as.data.frame(cbind(df,Cereals_data))
```

The Dendogram of Agnus- ward shows the better hierarchical clustering which indicated 90% accuracy. And it further indicated how to cluster each element into 10 clusters.

```
# 03.A) Cluster partition A
set.seed(123)
df_A <- df1[1:55,]
df_B <- df1[56:74,]

## Find the distance for partition A
set.seed(123)
df_A_distance <- get_dist(df_A, method = "euclidean")

## check the distance with other linkage methods of AGNUS for partition A
df_A_1 <- agnes(df_A_distance, method = "single")
df_A_2 <- agnes(df_A_distance, method = "complete")
df_A_3 <- agnes(df_A_distance, method = "average")
df_A_4 <- agnes(df_A_distance, method = "ward")

## compare the values
print(df_A_1 $ac)
```

```
## [1] 0.6663587
```

```
print(df_A_2 $ac)
```

```
## [1] 0.8285192
```

```
print(df_A_3 $ac)
```

```
## [1] 0.7646836
```

```
print(df_A_4 $ac)
```

```
## [1] 0.8891086
```

```
# According to the above comparison the "ward" is the best linkage method for cluster A which gi
ves the highest accuracy of 88.91%


## Following up all the above steps for partition B
## Find the distance for partition B
set.seed(123)
df_B_distance <- get_dist(df_B, method = "euclidean")

## check the distance with other linkage methods of AGNUS for partition A
df_B_1 <- agnes(df_B_distance, method = "single")
df_B_2 <- agnes(df_B_distance, method = "complete")
df_B_3 <- agnes(df_B_distance, method = "average")
df_B_4 <- agnes(df_B_distance, method = "ward")

## compare the values
print(df_B_1 $ac)
```

```
## [1] 0.4805129
```

```
print(df_B_2 $ac)
```

```
## [1] 0.71298
```

```
print(df_B_3 $ac)
```

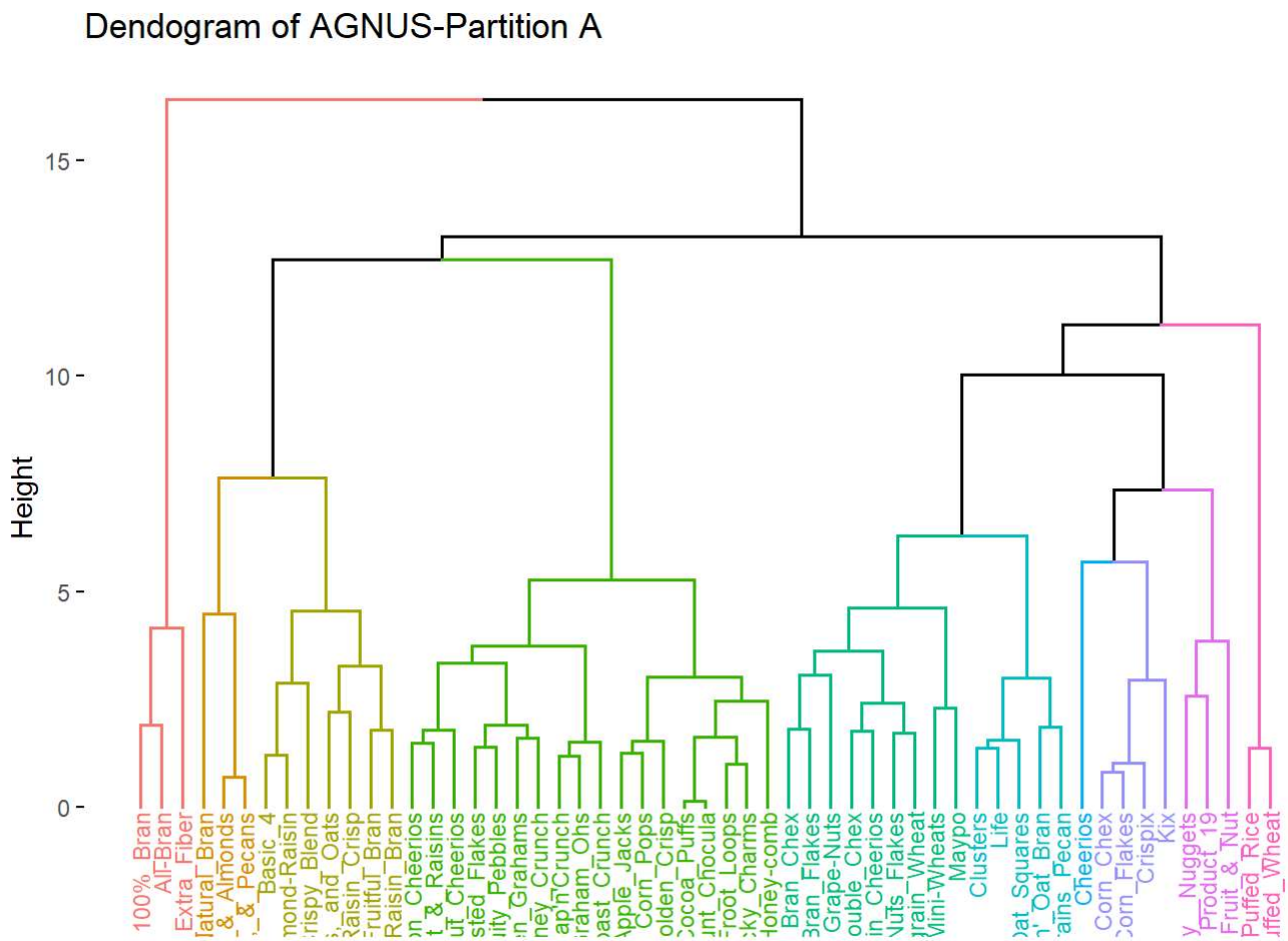
```
## [1] 0.6232053
```

```
print(df_B_4 $ac)
```

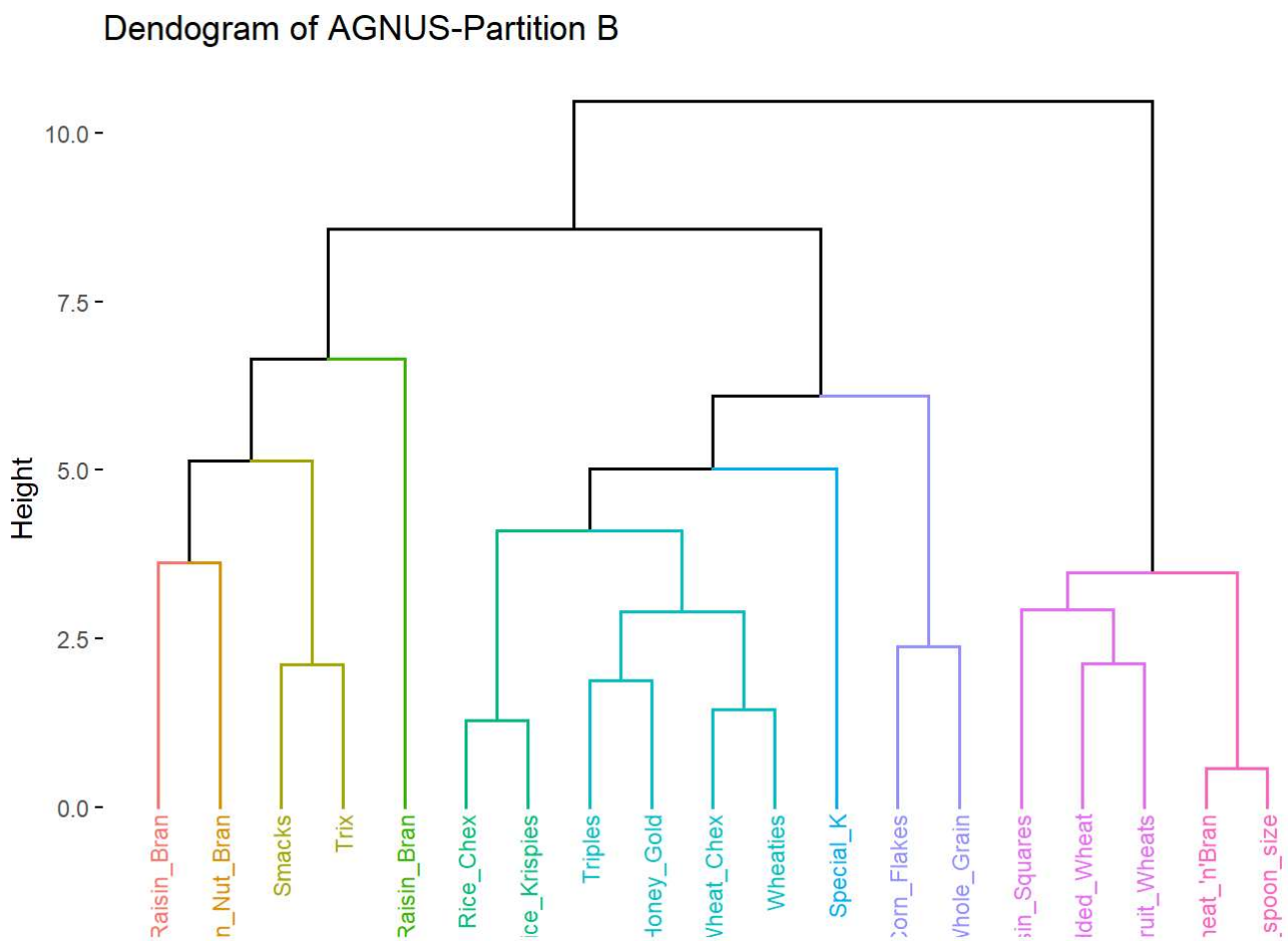
```
## [1] 0.7710122
```

```
# According to the above comparison the "ward" is the best linkage method for cluster B which gi
ves the highest accuracy of 77.10%

## Dendrogram of partition A
fviz_dend(df_A_4, k=10, cex = 0.6, main = "Dendogram of AGNUS-Partition A")
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <]8;;https://github.com/kassambara/factoextra/issues]8;;>.
```




```
## Dendrogram of partition B
fviz_dend(df_B_4, k=10, cex = 0.6, main = "Dendogram of AGNUS-Partition B")
```




```
# 03. B) Use the cluster centroids from A to assign each record in partition B
clusterd_df_A <- cutree(df_A_4, k=10)
clusters_A <- as.data.frame(cbind(df_A,clusterd_df_A))
clust_1 <- colMeans(clusters_A[clusters_A$clusterd_df_A == "1", ]) # The column means represents
the centroids

clusterd_df_B <- cutree(df_B_4, k=10)
clusters_B <- as.data.frame(cbind(df_B,clusterd_df_B))
clust_2 <- colMeans(clusters_B[clusters_B$clusterd_df_B == "1", ]) # The column means represents
the centroids


## Find the Centroid
Centroid <- rbind(clust_1,clust_2)
Centroid
```

```
##           calories   protein        fat    sodium    fiber      carbo
## clust_1 -2.2018711 1.3817478 -0.3310734 0.1727901 3.641312 -2.0718749
## clust_2  0.6537514 0.4522084  0.0000000 0.5755136 1.165443 -0.1875105
##             sugars   potass   vitamins     weight      cups     rating
## clust_1 -0.7894824 2.983781 -0.1818422 -0.2008324 -1.845255  2.2426479
## clust_2  1.1222223 1.996178 -0.1818422  1.9501886 -0.303848 -0.2217938
##         clusterd_df_A
## clust_1             1
## clust_2             1
```

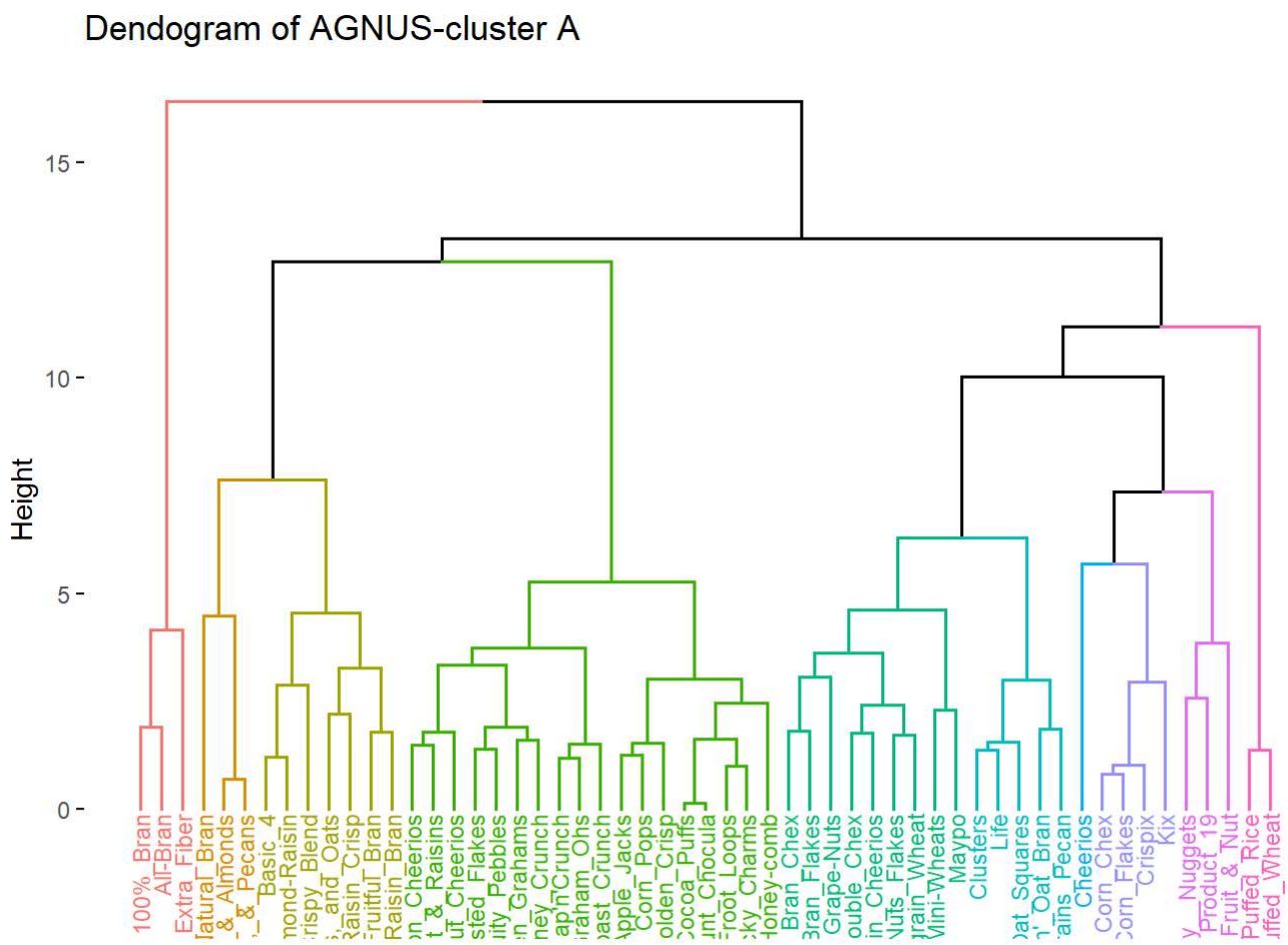
 # 03. C) Assess how consistent the cluster assignments are compared to the assignments based on all the data.
 # According to the above explanations with centroids, it can be identifies some facts regarding each cluster.In cluster 1 is high in protein, fiber and potassium compared to cluster 2. Sugar i s high in cluster 2 compared to cluster A. Therefore, it seems to be cereals in cluster A is bet ter than cereals in B in health wise.

 # 04. The elementary public schools would like to choose a set of cereals to include in their da ily cafeterias. Every day a different cereal is offered, but all cereals should support a health y 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?

Centroid

```
##           calories   protein        fat    sodium    fiber      carbo
## clust_1 -2.2018711 1.3817478 -0.3310734 0.1727901 3.641312 -2.0718749
## clust_2  0.6537514 0.4522084  0.0000000 0.5755136 1.165443 -0.1875105
##             sugars   potass   vitamins     weight      cups     rating
## clust_1 -0.7894824 2.983781 -0.1818422 -0.2008324 -1.845255  2.2426479
## clust_2  1.1222223 1.996178 -0.1818422  1.9501886 -0.303848 -0.2217938
##         clusterd_df_A
## clust_1             1
## clust_2             1
```

```
fviz_dend(df_A_4, k=10, cex = 0.6, main = "Dendogram of AGNUS-cluster A")
```




```
# Normalization of the data is typically recommended before conducting a cluster analysis, espec
ially when the variables have different units of measurement or scales. In this case, it would b
e important to normalize the data to ensure that each variable carries equal weight in the clust
ering algorithm. As it indicates from the centroid grid, cluster1 contains the more healthy cere
als compared to the cluster B. Cereals in cluster A is rich in protein, fiber, potassium and tho
se type of nutrients are essential for kids in elementary level.Therefore all the cereals which
indicate in the dendogram are good for the kids in the elementary school. But when wen consider
them individually their nutrition may vary to each other.

# Further We can use median values of each clusters for the further identification.
# Use aggregate function to calculate median of each variable
summary_table <- aggregate(df, by=list(Cereals_data), FUN=median)

# Rename the first column to "Cluster"
colnames(summary_table)[1] <- "Cluster"

# View the Summary table
View(summary_table)
```

```
library(knitr)
summary_table <- aggregate(df, by=list(Cereals_data), FUN=median)
colnames(summary_table)[1] <- "Cluster"
kable(summary_table)
```

Cluster	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	weight	cups	rating
1	70	4.0	1.0	140	10.0	7.0	5	320.0	25	1.00	0.330	68.40297
2	110	3.0	3.0	140	2.5	13.0	7	135.0	25	1.00	0.500	40.40021
3	110	1.0	1.0	180	0.0	12.0	12	40.0	25	1.00	0.750	29.50954
4	130	3.0	1.5	195	3.5	14.5	11	175.0	25	1.33	0.670	37.43958
5	100	3.0	1.0	200	3.0	17.0	4	90.0	25	1.00	0.750	50.68982
6	110	6.0	1.0	260	1.5	16.5	2	80.0	25	1.00	1.125	51.94816
7	110	2.0	0.0	270	0.0	21.5	3	32.5	25	1.00	1.000	41.72198
8	90	3.0	0.0	0	3.0	16.0	3	100.0	25	1.00	0.800	59.36399
9	105	2.5	1.0	200	1.0	18.5	3	52.5	100	1.00	1.000	40.17164
10	50	1.5	0.0	0	0.5	11.5	0	32.5	0	0.50	1.000	61.88088

```
# Cereals cluster 01
# According to the above table, when consider about the each 10 clusters, cereals in cluster one
has lot of fiber, potassium, less calories and carbohydrate. Therefore its seems to be a good ch
oice out of other clusters and it has the highest rating out of all the others.

# Get the indices of data points in cluster 1
cluster1_indices <- which(k10$cluster == 1)
# Extract the corresponding names from the original data frame
cluster1_names <- data$name[cluster1_indices]
# Print the list of names in cluster 1
print(cluster1_names)
```

```
## [1] "Cheerios"  "Special_K"
```

```
library(dplyr)
# Cereals in cluster 06
# Based on the above summary table, cereals in cluster 6 are high in protein, rich in sodium and
less sugar.
# Get the row indices of cereals in cluster 06
cluster_06_indices <- which(k10$cluster == 6)

# Get the names of cereals in cluster 06
cereals_cluster_06 <- row.names(df1)[cluster_06_indices]

# Print the names of cereals in cluster 06
cereals_cluster_06
```

```
##  [1] "Bran_Chex"            "Bran_Flakes"          "Double_Chex"
##  [4] "Grape_Nuts_Flakes"    "Grape-Nuts"           "Multi-Grain_Cheerios"
##  [7] "Nutri-grain_Wheat"    "Quaker_Oat_Squares"   "Wheat_Chex"
## [10] "Wheaties"
```

```
cat("Healthy cereals: ", paste(cluster1_names, collapse = ", "), ", ",
    "Cereals in cluster 06: ", paste(cereals_cluster_06, collapse = ", "))
```

```
## Healthy cereals:  Cheerios, Special_K ,  Cereals in cluster 06:  Bran_Chex, Bran_Flakes, Doub
le_Chex, Grape_Nuts_Flakes, Grape-Nuts, Multi-Grain_Cheerios, Nutri-grain_Wheat, Quaker_Oat_Squa
res, Wheat_Chex, Wheaties
```

```
# Therefore the above mentioned cereals seems to be good for the kids. But based on their health
concerns someone can choose another type of cluster for cereals.
```