**Clustering:**

**Hierarchical and Partitional**

**Project Report**

**Submitted By: Date:**

***Team 06*** March 26, 2020

Amit Verma (100329477)

Sparsh Sharma (100329093)

Simardeep Kaur (100329207)

Charanjeet Singh (100329356)

Contents

[Hierarchical Clustering 4](#_Toc35709124)

[ Introduction 4](#_Toc35709125)

[ Data Preparation 4](#_Toc35709126)

[ Distance Matrix 5](#_Toc35709127)

[Euclidean based distance 5](#_Toc35709128)

[Correlation based distance 6](#_Toc35709129)

[ Agglomerative Clustering 6](#_Toc35709130)

[Hierarchical Tree 6](#_Toc35709131)

[Dendrogram 7](#_Toc35709132)

[Verify Cluster Tree 8](#_Toc35709133)

[Cut Dendrogram into different groups 8](#_Toc35709134)

[ Comparing Dendrogram 12](#_Toc35709135)

[Visual Comparison of Dendrograms 12](#_Toc35709136)

[Correlation Matrix between list of dendrograms 13](#_Toc35709137)

[Visualizing Correlation matrix using corrplot package 14](#_Toc35709138)

[Visualizing Dendrogram 15](#_Toc35709139)

[Basic Dendrogram 16](#_Toc35709140)

[Circular Dendrogram 16](#_Toc35709141)

[Phylogenic Dendrogram 17](#_Toc35709142)

[ Heatmap 18](#_Toc35709143)

[Basic heatmap 18](#_Toc35709144)

[Pretty heatmap 19](#_Toc35709145)

[Determining optimal number of clusters 20](#_Toc35709146)

[Elbow method 20](#_Toc35709147)

[Silhouette Method 21](#_Toc35709148)

[Gap Statitistics 22](#_Toc35709149)

[Clustering Analysis 23](#_Toc35709150)

[ Cluster Validation 24](#_Toc35709151)

[Silhouette Measure 24](#_Toc35709152)

[Dunn Index 26](#_Toc35709153)

[ Choosing best clustering method 27](#_Toc35709154)

[Partitional clustering 28](#_Toc35709155)

[ Introduction 28](#_Toc35709156)

[ Data Preparation 28](#_Toc35709157)

[ Elbow Method to find optimal number of clusters 29](#_Toc35709158)

[ Computing k-means 30](#_Toc35709159)

[Computing Mean 31](#_Toc35709160)

[Visualize k-means clusters 31](#_Toc35709161)

[ PAM: Partitioning Around Medoids 32](#_Toc35709162)

[Performing PAM 32](#_Toc35709163)

[Silhouette Method 33](#_Toc35709164)

[Performing PCA 33](#_Toc35709165)

[ Heatmap 34](#_Toc35709166)

[ Evaluating Clustering Tendency 35](#_Toc35709167)

[Hopkins Test 35](#_Toc35709168)

[Dissimilarity Matrix 35](#_Toc35709169)

[ Optimal Number of Clusters 36](#_Toc35709170)

[Gap Statistics 36](#_Toc35709171)

[Hubert Index 37](#_Toc35709172)

[Nb\_Clust 38](#_Toc35709173)

[ Dunn Index 39](#_Toc35709174)

[ Cluster Statistics 39](#_Toc35709175)

[ Choosing best clustering method 39](#_Toc35709176)

[Cluster validation: Stability measures 40](#_Toc35709177)

# Hierarchical Clustering

Hierarchical clustering*,*also known as hierarchical cluster analysis*,* is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters*,*where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

## Introduction

This dataset refers to clients of a wholesale distributor. It includes the annual spending in monetary units (m.u) on diverse product categories. Our aim is to make clusters from this data that can segment similar clients together.

It has **440 rows** and **8 variables**.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Column Type** | **Description** |
| Channel | int | customer Channel - (Hotel/Restaurant/Cafe)  (Nominal) |
| Region | int | Customer region Lisnon, Oporto or Other (Nominal) |
| Fresh | int | annual spending (m.u.) on fresh products (Continuous); |
| Milk | int | annual spending (m.u.) on milk products (Continuous); |
| Grocery | int | annual spending (m.u.) on grocery products (Continuous); |
| Frozen | int | annual spending (m.u.) on frozen products (Continuous) |
| Detergents\_Paper | int | annual spending (m.u.) on detergents and paper products (Continuous) |
| Delicassen | int | annual spending (m.u) on and delicatessen products (Continuous); |

## Data Preparation

We have accessed data from <https://archive.ics.uci.edu/ml/datasets/Wholesale+customers>

To normalize this data, we have used scale () function in R.

**R Code:**

#Reading file

data<-read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale customers data.csv")

#Data structure and preparation

df<-data[3:8]

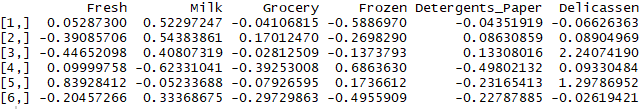
# Standardize the data

dfscaled<-scale(df)

# Show the first 6 rows

head(dfscaled,nrow=5)

**Output:**



## Distance Matrix

### Euclidean based distance

We have used Euclidean distance to compute the distance matrix with dist () function. Also, to make it into a presentable form we have converted it into a matrix.

**R Code:**

# Similarity Measures

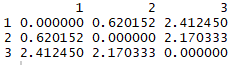
#Compute dissimilarity matrix

res.dist <- dist(dfscaled, method = "euclidean")

## ------------------------------------------------------------------------

as.matrix(res.dist)[1:3,1:3]

**Output:**



### Correlation based distance

Correlation based distance is computed using get\_dist() function in factoextra package. There are three methods to find correlation distance i.e. pearson, spearman and kendall.

We have used pearson based distance to show our analysis.

**R Code:**

# Correlation based distance

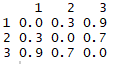
library("factoextra")

dist.cor <- get\_dist(dfscaled, method = "pearson")

# Display a subset

round(as.matrix(dist.cor)[1:3, 1:3], 1)

**Output:**



## Agglomerative Clustering

Agglomerative hierarchical method starts with individual objects. Thus, there are as many clusters as objects. The most similar objects are grouped first and these initial groups are merged according to their similarities. Eventually, as the similarity decreases all subgroups are fused into single cluster.

### Hierarchical Tree

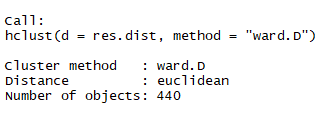
After calculating the distance matrix (Euclidean), we have used hclut() function to create hierarchical tree.

**R Code:**

res.dist <- dist(dfscaled, method = "euclidean")

res.hc <- hclust(d = res.dist, method = "ward.D")

**Output:**



### Dendrogram

To visually represent the hierarchical tree, we have created a dendrogram using fviz\_dend() function in factoextra package.

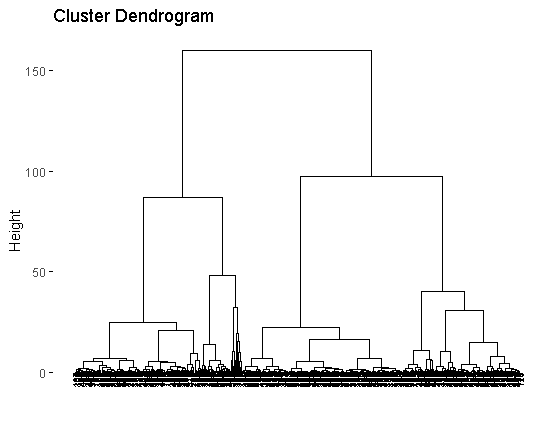
**R Code:**

#Dendrogram

library("factoextra")

fviz\_dend(res.hc, cex = 0.5)

**Output:**



**Interpretation:**

From the dendrogram above we can conclude that we can divide the customers into 3 segments after the height of 25.

### Verify Cluster Tree

To verify the cluster tree created above, correlation between cophenetic distance and original distance is computed.

**R Code:**

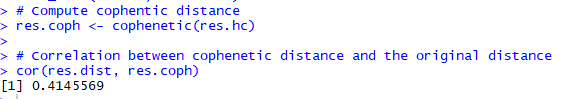
# Compute cophentic distance

res.coph <- cophenetic(res.hc)

# Correlation between cophenetic distance and the original distance

cor(res.dist, res.coph)

**Output:**



Also, we have executed hclust() function with average linkage method to get another clustering solution.

**R Code:**

res.hc2 <- hclust(res.dist, method = "average")

cor(res.dist, cophenetic(res.hc2))

**Output:**



**Interpretation:**

We can see that the correlation is better when using a different linkage method as calculated above.

### Cut Dendrogram into different groups

To partition our data, we have cut the hierarchical tree using cutree() function by specifying the number of clusters.

**R Code:**

#Cut tree into 2 groups

grp <- cutree(res.hc, k = 2)

head(grp, n = 4)

**Output:**



To check the number of members in each cluster, we have used the table() function.

**R Code:**

# Number of members in each cluster

table(grp)

**Output:**



**Interpretation:**

From the table above it can be observed that the group 1 contains 165 observations and group 2 contains 275 observations.

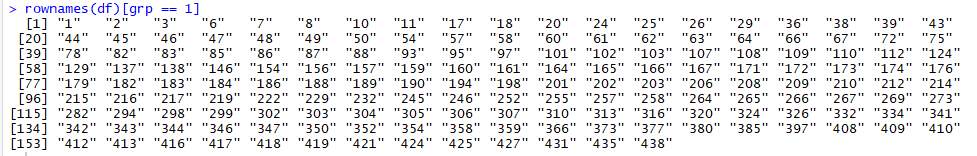
The function rownames() has been used to get the names for the members of cluster 1.

**R Code:**

# Get the names for the members of cluster 1

rownames(df)[grp == 1]

**Output:**



These 2 groups can be visually represented in the form of dendrogram using fviz\_dend() function.

We have used “color” parameter in order to segregate them into 2 different clusters.

**R Code:**

# Cut in 2 groups and color by groups

fviz\_dend(res.hc, k = 2, # Cut in 2 groups

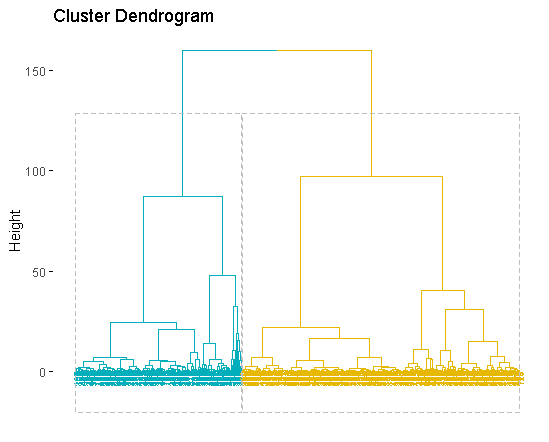
cex = 0.5, # label size

k\_colors = c("#00AFBB", "#E7B800"),

color\_labels\_by\_k = TRUE, # color labels by groups

rect = TRUE)

**Output:**



To represent these groups in the form of clusters, we have used fviz\_cluster() function in factoextra package.

**R Code:**

#Clusterplot

fviz\_cluster(list(data = dfscaled, cluster = grp),

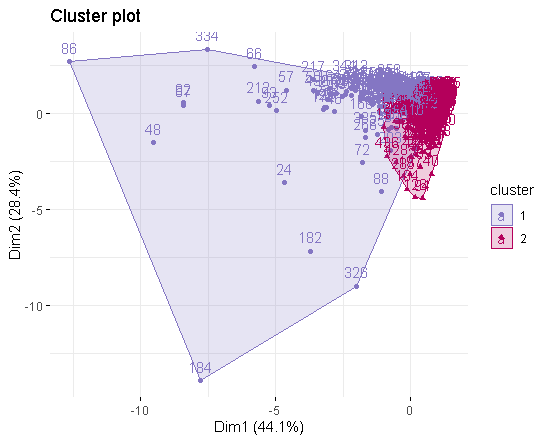
palette = c("#8476c3", "#b4005b"),

ellipse.type = "convex", # Concentration ellipse

repel = FALSE, # Avoid label overplotting (slow)

show.clust.cent = FALSE, ggtheme = theme\_minimal())

**Output:**



## Comparing Dendrogram

A list of two dendrograms has been created using two different methods for clustering i.e. average and ward.D.

**R Code:**

#Comparing Dendrogram

library(dendextend)

# Compute distance matrix

res.dist <- dist(dfscaled, method = "euclidean")

# Compute 2 hierarchical clusterings

hc1 <- hclust(res.dist, method = "average")

hc2 <- hclust(res.dist, method = "ward.D")

# Create two dendrograms

dend1 <- as.dendrogram (hc1)

dend2 <- as.dendrogram (hc2)

# Create a list to hold dendrograms

dend\_list <- dendlist(dend1, dend2)

### Visual Comparison of Dendrograms

To visually compare the dendrogram created above, we have used tanglegram() function, which plots the dendrograms sideways. Also, we have used entanglement() function to check the quality of alignment between the two dendrograms.

**R Code:**

#Visual comparison of two dendrograms

tanglegram(dend1, dend2,

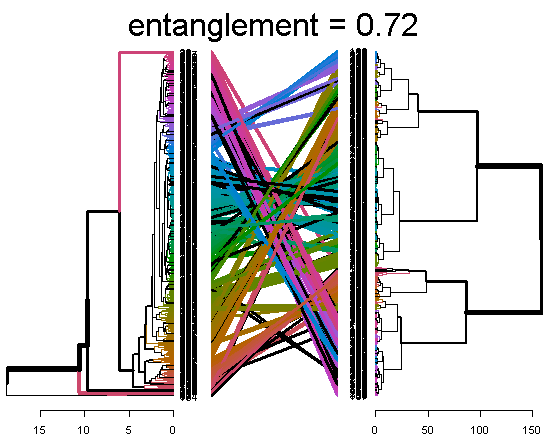
highlight\_distinct\_edges = FALSE, # Turn-off dashed lines

common\_subtrees\_color\_lines = TRUE, # Turn-off line colors

common\_subtrees\_color\_branches = TRUE, # Color common branches

main = paste("entanglement =", round(entanglement(dend\_list), 2)))

**Output:**



**Interpretation:**

The entanglement is 0.72 which means that the quality of alignment is poor as it should be near to 0.

### Correlation Matrix between list of dendrograms

We have computed the correlation matix between a list of two dendrograms using cor.dendlist() function w.r.t two methods that are cophenetic and baker.

**R Code:**

# Cophenetic correlation matrix

cor.dendlist(dend\_list, method = "cophenetic")

**Output:**



**R Code:**

# Baker correlation matrix

cor.dendlist(dend\_list, method = "baker")

**Output:**



**Interpretation:**

Since the correlation coefficient values calculated from baker and cophenetic are near to 0 so they are not similar.

### Visualizing Correlation matrix using corrplot package

We have visualized multiple dendrograms at the same time by computing a correlation matrix between different dendrograms. This correlation matrix has been visualized using corrplot() function in corrplot package in R.

**R Code:**

# Create multiple dendrograms by chaining

dend1 <- dfscaled %>% dist %>% hclust("complete") %>% as.dendrogram

dend2 <- dfscaled %>% dist %>% hclust("single") %>% as.dendrogram

dend3 <- dfscaled %>% dist %>% hclust("average") %>% as.dendrogram

dend4 <- dfscaled %>% dist %>% hclust("centroid") %>% as.dendrogram

# Compute correlation matrix

dend\_list <- dendlist("Complete" = dend1, "Single" = dend2,

"Average" = dend3, "Centroid" = dend4)

cors <- cor.dendlist(dend\_list)

# Print correlation matrix

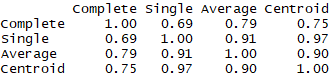
round(cors, 2)

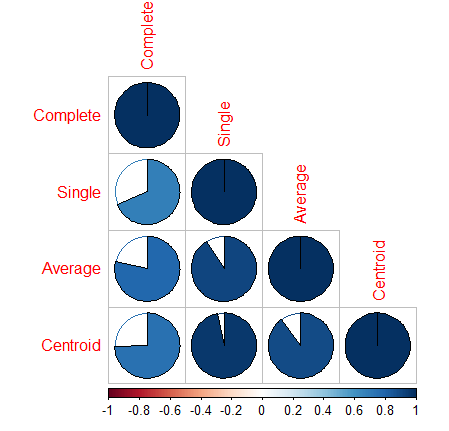
# Visualize the correlation matrix using corrplot package

library(corrplot)

corrplot(cors, "pie", "lower")

**Output:**

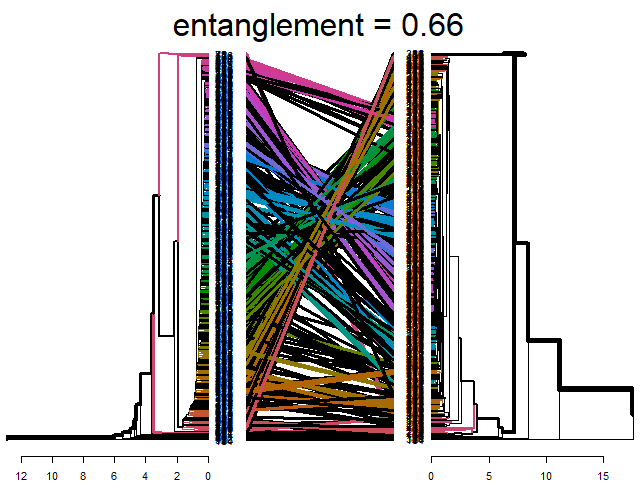




**Interpretation:**

Since the Single and Centroid has the maximum correlation coefficient as the area covered by pie is the most so we should use Single and Centroid for comparison.

So, we checked the entanglement quality again with single and centroid where we got the entanglement as 0.66.



## Visualizing Dendrogram

To make beautiful dendrograms, we have used fviz\_dend() function in factoextra package. We have created three different types of dendrograms i.e. basic, circular and phylogenic.

### Basic Dendrogram

**R Code:**

#Visualizing dendrograms

fviz\_dend(res.hc, k = 2, # Cut in 2 groups

cex = 0.5, # label size

k\_colors = c("#00AFBB", "#E7B800"),

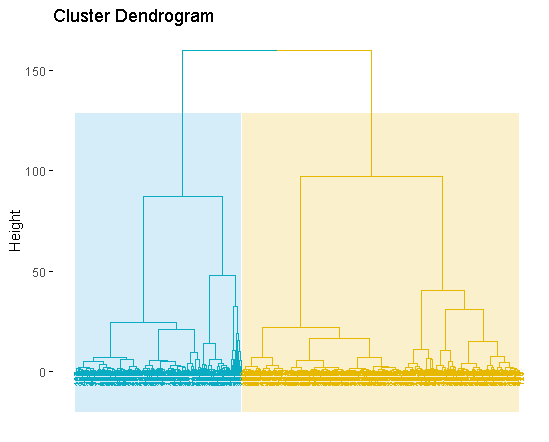
color\_labels\_by\_k = TRUE, # color labels by groups

rect = TRUE, # Add rectangle around groups

rect\_border = c("#2E9FDF", "#E7B800"),

rect\_fill = TRUE)

**Output:**



### Circular Dendrogram

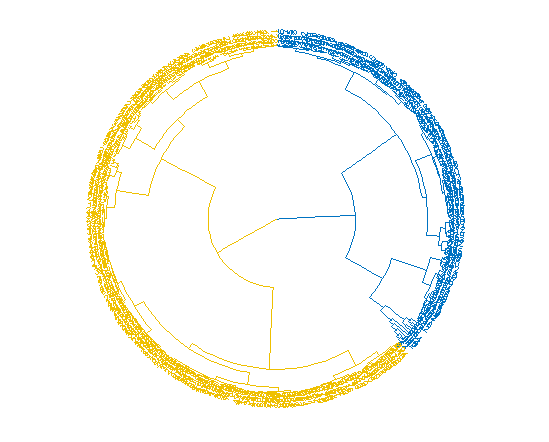
**R Code:**

#Circular Dendrogram

fviz\_dend(res.hc, cex = 0.5, k = 2,

k\_colors = "jco", type = "circular")

**Output:**



### Phylogenic Dendrogram

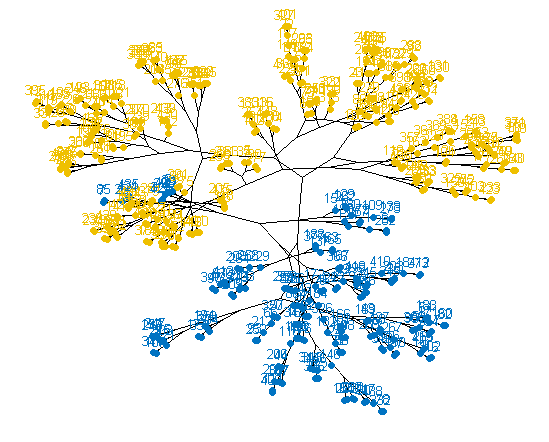
**R Code:**

require("igraph")

fviz\_dend(res.hc, k = 2, k\_colors = "jco",

type = "phylogenic", repel = FALSE)

**Output:**



## Heatmap

To visualize hierarchical clustering in a different way, we have plotted two different kinds of heatmaps.

### Basic heatmap

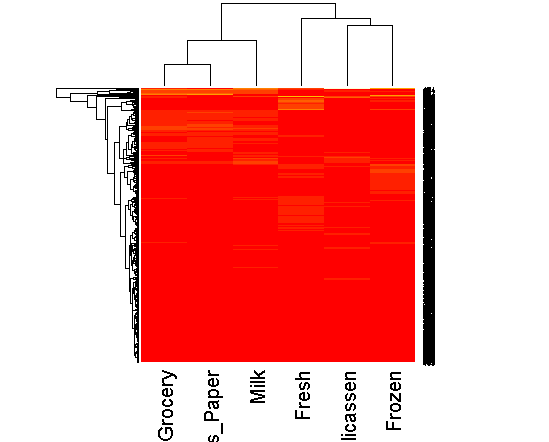
Basic heatmap an be created using heatmap() function in stats package.

**R Code:**

# Default plot

heatmap(dfscaled, scale = "none")

**Output:**



**Interpretation:**

From the above heatmap we can conclude that the Frozen, Delicassen and Fresh food form a single cluster as they are all store selling cold cuts, cheeses, and a variety of salads, as well as a selection of unusual or foreign prepared foods.

Where as grocery, Milk and Detergent Paper form a separate cluster as Grocery and milk goes hand in hand together.

### Pretty heatmap

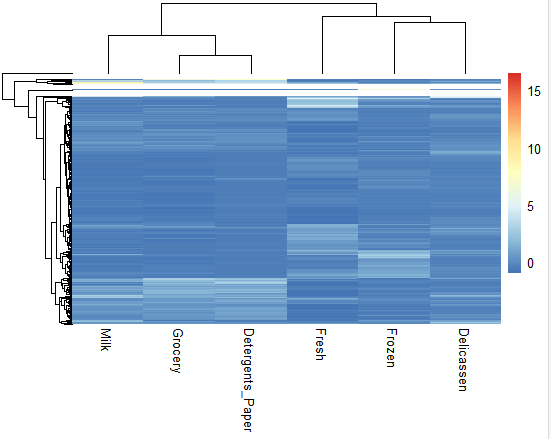
Another function to create a heatmap is by using function pheatmap() in pheatmap package. It uses Euclidean distance by default.

**R Code:**

library("pheatmap")

pheatmap(dfscaled, cutree\_rows = 4)

**Output:**



## Determining optimal number of clusters

To check optimal number of clusters for our data, we have used elbow method, silhouette method and gap statistics in factoextra and nbclust package.

### Elbow method

**R Code:**

library(factoextra)

library(NbClust)

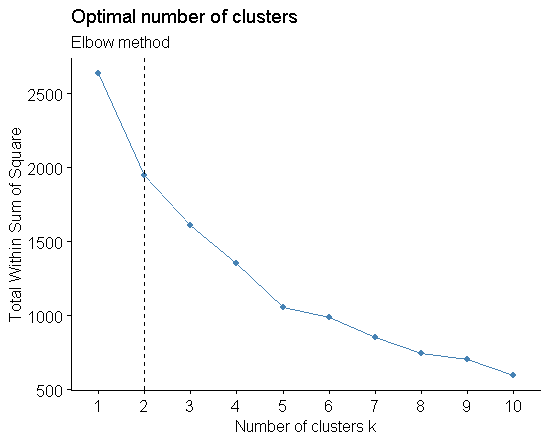
# Elbow method

fviz\_nbclust(dfscaled, kmeans, method = "wss") +

geom\_vline(xintercept = 2, linetype = 2)+

labs(subtitle = "Elbow method")

**Output:**



### Silhouette Method

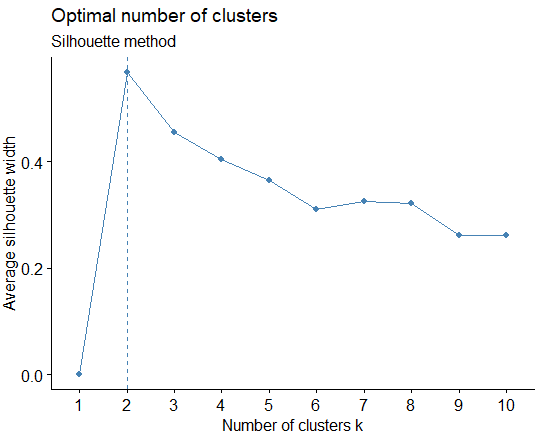
**R Code:**

# Silhouette method

fviz\_nbclust(dfscaled, kmeans, method = "silhouette")+

labs(subtitle = "Silhouette method")

**Output:**



### Gap Statitistics

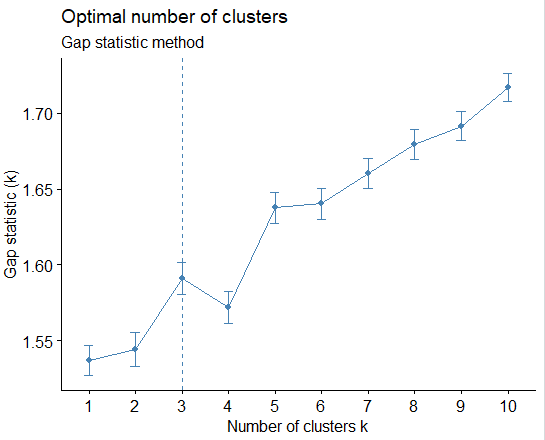
**R Code:**

# Gap statistic

fviz\_nbclust(dfscaled, kmeans, nstart = 25, method = "gap\_stat", nboot = 50)+

labs(subtitle = "Gap statistic method")

**Output:**



**Interpretation:**

From the above 3 methods to find out optimal number of clusters, it can be observed that

* Elbow Method: 2 clusters solution suggested
* Gap Statistics: 3 clusters solution suggested
* Silhouette Method: 2 clusters solution suggested

So, we will consider 2 clusters for our analysis.

## Clustering Analysis

### Hierarchical Clustering

Hierarchical clustering can also be represented visually using eclust() function.

**R Code:**

# Hierarchical clustering

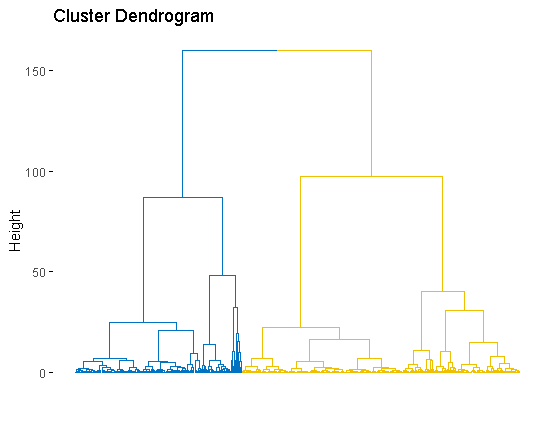
hc.res <- eclust(dfscaled, "hclust", k = 2, hc\_metric = "euclidean",hc\_method = "ward.D2", graph = FALSE)

# Visualize dendrograms

fviz\_dend(hc.res, show\_labels = FALSE,

palette = "jco", as.ggplot = TRUE)

**Output:**



## Cluster Validation

### Silhouette Measure

To check how similar an object is to the other objects in its own cluster versus those in neighbor cluster, we have used fviz\_silhouette() function in factoextra package.

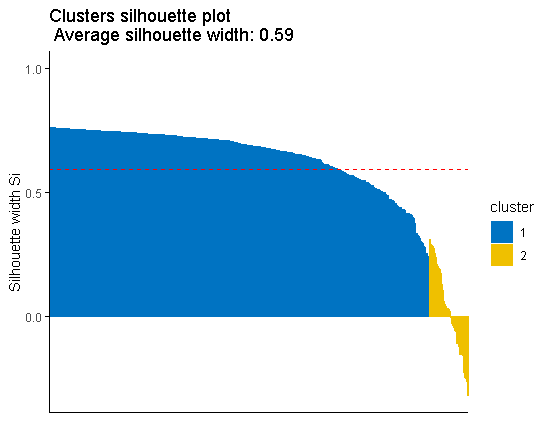
**R Code:**

km.res <- eclust(dfscaled, "kmeans", k = 2, nstart = 25, graph = FALSE)

fviz\_silhouette(km.res, palette = "jco",

ggtheme = theme\_classic())

**Output:**





#### Silhouette information

To extract the silhouette information we can use the following commands.

**R Code:**

silinfo <- km.res$silinfo

names(silinfo)

**Output:**



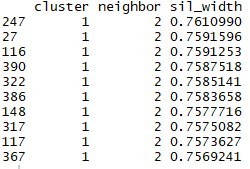
To check silhouette width of each observation:

**R Code:**

# Silhouette widths of each observation

head(silinfo$widths[, 1:3], 10)

**Output:**



We have checked the objects with negative silhouette coefficient. We have found the name of these samples and identified the neighbor clusters.

**R Code:**

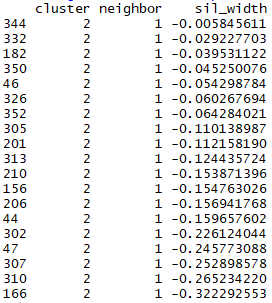
sil <- km.res$silinfo$widths[, 1:3]

# Objects with negative silhouette

neg\_sil\_index <- which(sil[, 'sil\_width'] < 0)

sil[neg\_sil\_index, , drop = FALSE]

**Output:**



**Interpretation:**

The above elements with negative silhouette width belong to wrong cluster and they should belong to its neighbouring cluster i.e. cluster 1.

### Dunn Index

We have performed cluster validation using dunn index in fpc package of R.

**R Code:**

#Cluster Validation with Dunn index

library(fpc)

# Statistics for k-means clustering

km\_stats <- cluster.stats(dist(dfscaled), km.res$cluster)

# Dun index

km\_stats$dunn

**Output:**



**Interpretation:**

The Dunn’s index is 0.014 which is very small value which depicts that the clusters are not well separated. Higher Dunn’s index implies more compact and well separated clusters.

## Choosing best clustering method

To check the best algorithm for clustering, we have used clalid() function in clValid package of R.

**R Code:**

#Choosing the Best Clustering Algorithms

library("clValid")

# Compute clValid

clmethods <- c("hierarchical","kmeans","pam")

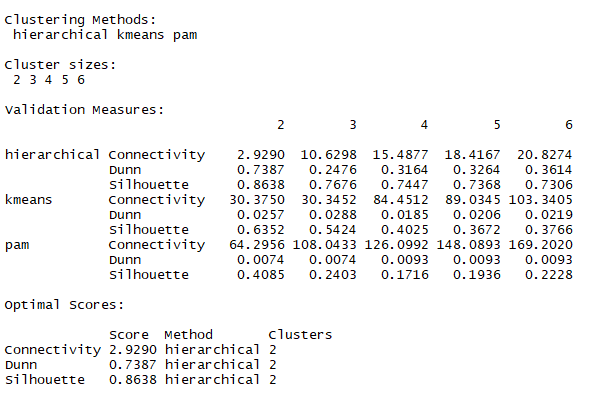
intern <- clValid(dfscaled, nClust = 2:6,

clMethods = clmethods, validation = "internal")

# Summary

summary(intern)

**Output:**



**Interpretation:**

Since the Hierarchical clustering has the lowest connectivity value and highest Dunn’s and Silhouette value so we will choose “Hierarchical” clustering for our analysis.

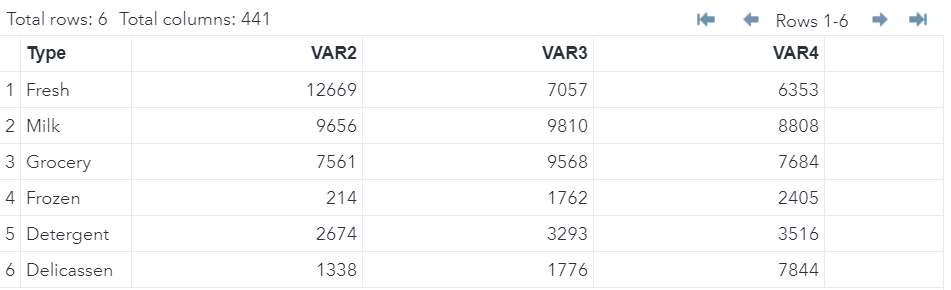
## SAS Analysis (Hierarchical)

### Data Preparation

**SAS code:**

proc import datafile= "/folders/myfolders/Wholesale.csv"  
OUT=WORK.wholesale   
DBMS=CSV   
REPLACE;   
RUN;

**Output:**



### Ward Pseudo Method in SAS

**SAS code:**

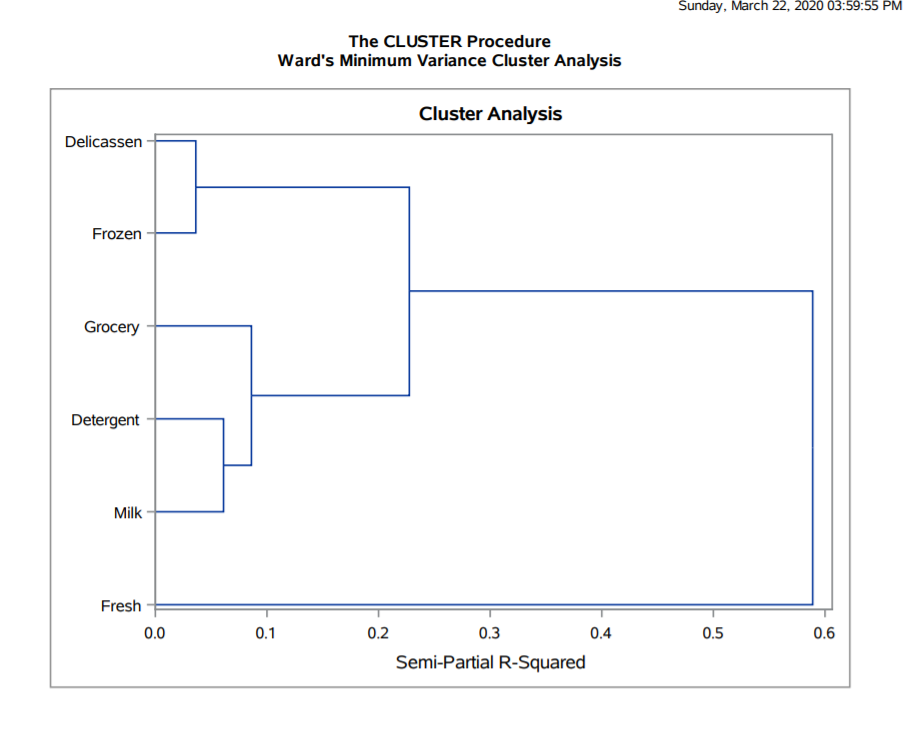
proc cluster data=WORK.wholesale method=ward pseudo;

id Type;

run;

**Output:**

Embedded output below in a pdf file. Click to view the whole output.



**Interpretation:**

From the above output, it can be seen that in the initial stage, Delicassen and Frozen are joined. In the second stage Detergent and Milk form the cluster. In the third stage, Grocery is joined with (Detergent, Milk) cluster. In the fourth stage, (Delicassen, Frozen) cluster is joined with (Milk, Detergent, Grocery) cluster. In the final stage, Fresh is joined with (Delicassen, Frozen, Grocery, Detergent, Milk) cluster.

Thus, there are 2 clusters formed:

* Cluster 1: Milk, Grocery, Detergent
* Cluster 2: Delicassen, Frozen

Finally, they are combined to form last cluster with fresh

### Proc Tree Method in SAS

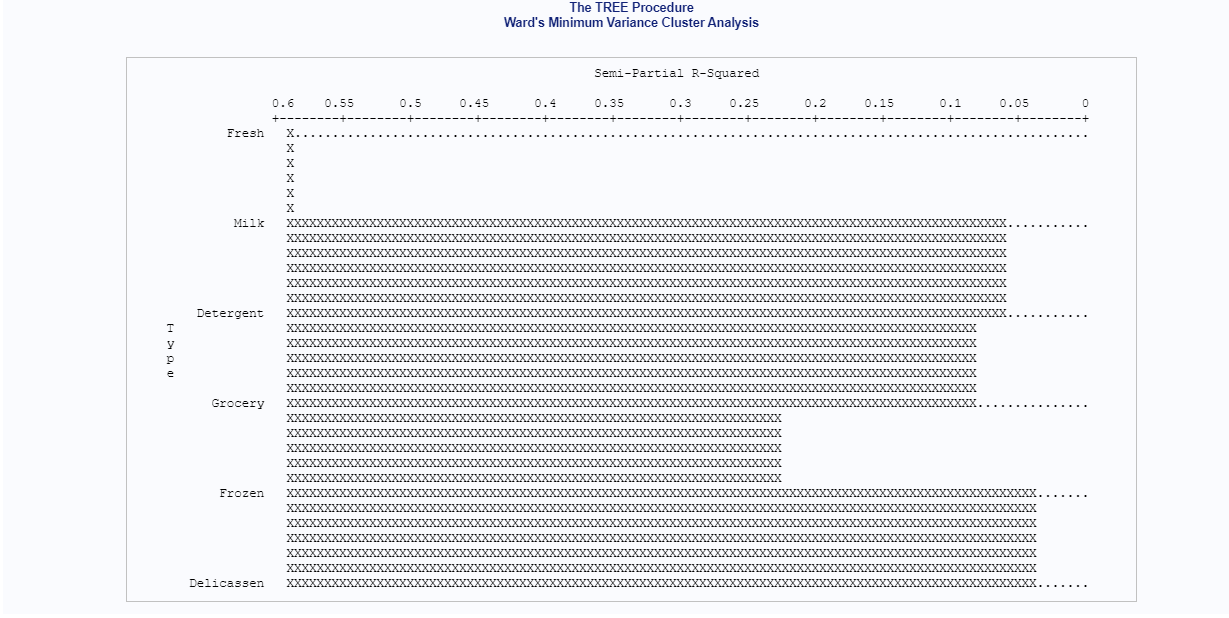
**SAS code:**

proc tree horizontal;

id Type;

run;

**Output:**



**Interpretation:**

The above plot indicates that Fresh has the highest R-squared value (i.e. 0.6) while Frozen and Delicassen have the lowest R-squared value (i.e. 0.05).

# Partitional clustering

Partitional clustering are clustering methods used to classify observations, within a data set, into multiple groups based on their similarity.

## Introduction

Measurement of geometrical properties of kernels belonging to three different varieties of wheat. A soft X-ray technique and GRAINS package were used to construct all seven, real-valued attributes.

This dataset has **210 rows and 9 variables**.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Column Type** | **Description** |
| ID | int | Id |
| Area | num | Area of seed |
| perimeter | num | Perimeter of kernel |
| compactness | num | Compactness of kernel |
| lengthOfKernel | num | length of kernel |
| widthOfKernel | num | Width of kernel |
| asymmetryCoefficient | num | Asymmetry coefficient |
| lengthOfKernelGroove | num | Length of the Kernel groove |
| seedType | int | Type of seed |

## Data Preparation

**R-Code**:

#Reading the file

wheat\_dataset<-read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/00236/seeds\_dataset.txt",header=F)

head(wheat\_dataset)

**Output:**

V1 V2 V3 V4 V5 V6 V7 V8

1 15.26 14.84 0.8710 5.763 3.312 2.221 5.220 1

2 14.88 14.57 0.8811 5.554 3.333 1.018 4.956 1

3 14.29 14.09 0.9050 5.291 3.337 2.699 4.825 1

4 13.84 13.94 0.8955 5.324 3.379 2.259 4.805 1

5 16.14 14.99 0.9034 5.658 3.562 1.355 5.175 1

6 14.38 14.21 0.8951 5.386 3.312 2.462 4.956 1

**R-Code**:

#Taking appropriate columns

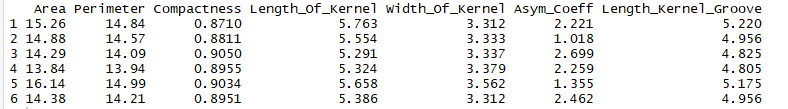
wheat\_data<-wheat\_dataset[ ,-8]

#Renaming columns

colnames(wheat\_data)<-c('Area','Perimeter','Compactness','Length\_Of\_Kernel','Width\_Of\_Kernel','Asym\_Coeff','Length\_Kernel\_Groove')

head(wheat\_data)

**Output:**



## Elbow Method to find optimal number of clusters

**R-Code:**

#Scaling the data

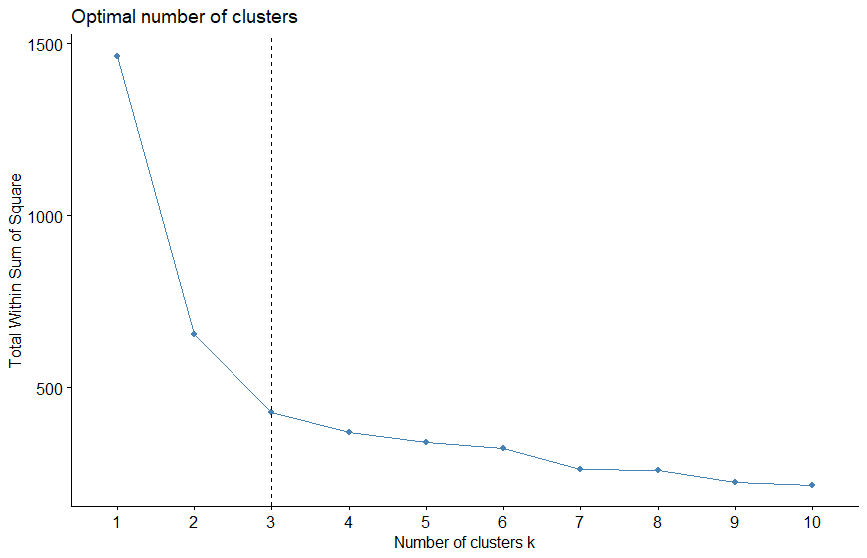
wheat\_data <- scale(wheat\_data)

#Elbow Method

fviz\_nbclust(wheat\_data, kmeans, method = "wss") +

geom\_vline(xintercept = 3, linetype = 2)

**Output:**



**Interpretation:**

The plot above represents the variance within the clusters. We can observe a bend (or "elbow") at k = 3. This bend indicates that additional clusters beyond the third cluster have a small variance. So, we'll classify the observations into 3 clusters.

## Computing k-means

**R-Code:**

# Computing k-means with k = 3

set.seed(123)

wheat\_km\_res <- kmeans(wheat\_data, 3, nstart = 25)

# Print the results

print(wheat\_km\_res)

**Output:**

K-means clustering with 3 clusters of sizes 71, 67, 72

Cluster means:

Area Perimeter Compactness Length\_Of\_Kernel Width\_Of\_Kernel Asym\_Coeff Length\_Kernel\_Groove

1 -0.1407831 -0.1696372 0.4485346 -0.2571999 0.001643014 -0.66034079 -0.5844965

2 1.2536860 1.2589580 0.5591283 1.2349319 1.162075101 -0.04511157 1.2892273

3 -1.0277967 -1.0042491 -0.9626050 -0.8955451 -1.082995635 0.69314821 -0.6233191

Clustering vector:

[1] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[59] 1 3 3 3 1 3 1 1 1 1 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

[117] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 1 2 1 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3

[175] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 1 3 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:

[1] 144.4586 139.5542 144.5954

(between\_SS / total\_SS = 70.7 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size"

[8] "iter" "ifault"

**Interpretation:**

We can observe that there are 3 clusters formed and also the within cluster sum of squares for 3 different clusters represents the distance of each observation from its cluster center. So, we need to have a lower within sum of squares.

### Computing Mean

**R-Code:**

#Taking aggregate mean

aggregate(wheat\_data, by=list(cluster=wheat\_km\_res$cluster), mean)

**Output:**



**Interpretation:**

The kmeans() output depicts the centroid vectors (cluster means), the group in which each observation was allocated (clustering vector). The percentage i.e. 70.7% represents the compactness of the clustering meaning that how similar are the members within the same group.

### Visualize k-means clusters

**R-Code:**

#To perform Principal Component Analysis (PCA) and to plot data points according to the first two principal components coordinates

fviz\_cluster(wheat\_km\_res, data = wheat\_data,

palette = c("#2E9FDF", "#00AFBB", "#E7B800"),

ellipse.type = "euclid",

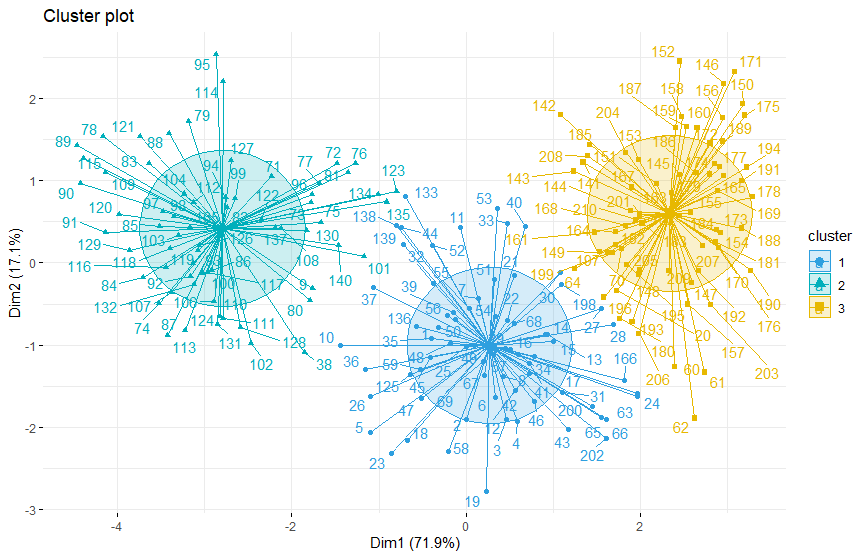
star.plot = TRUE,

repel = TRUE,

ggtheme = theme\_minimal()

)

**Output:**



**Interpretation:**

The plot above depicts the data points are placed in the three clusters w.r.t the first two principal components coordinates.

## PAM: Partitioning Around Medoids

### Performing PAM

**R-Code:**

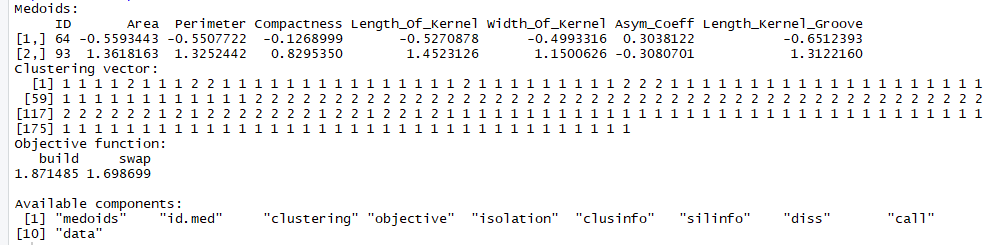
Library(‘cluster’)

#Performing PAM on dataset

wheat\_pam\_res <- pam(wheat\_data, 2)

print(wheat\_pam\_res)

**Output:**



**Interpretation:**

The PAM output shows the centroid vectors (cluster medoids), the group in which each observation was allocated (clustering vector). The objective function refers to the sum of distances of points to their medoid at the two stages i.e. build and swap stage.

### Silhouette Method

**R-Code:**

library('NbClust')

library('factoextra')

library('cluster')

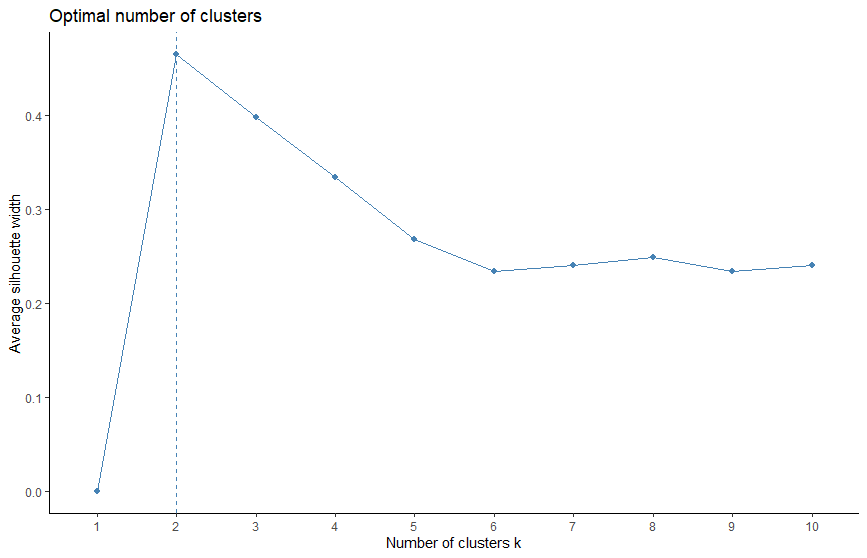
#Silhouette Method

fviz\_nbclust(wheat\_data, pam, method = "silhouette")+

theme\_classic()

**Output:**

**Output:**



**Interpretation:**

From the plot, the suggested optimal number of clusters is 2. So, we'll classify the observations into 2 clusters.

### Performing PCA

**R-Code:**

#Performing PCA and plotting points

fviz\_cluster(wheat\_pam\_res, palette = c("#00AFBB", "#FC4E07"), ellipse.type = "t", repel = TRUE,

ggtheme = theme\_classic())

**Output:**



**Interpretation:**

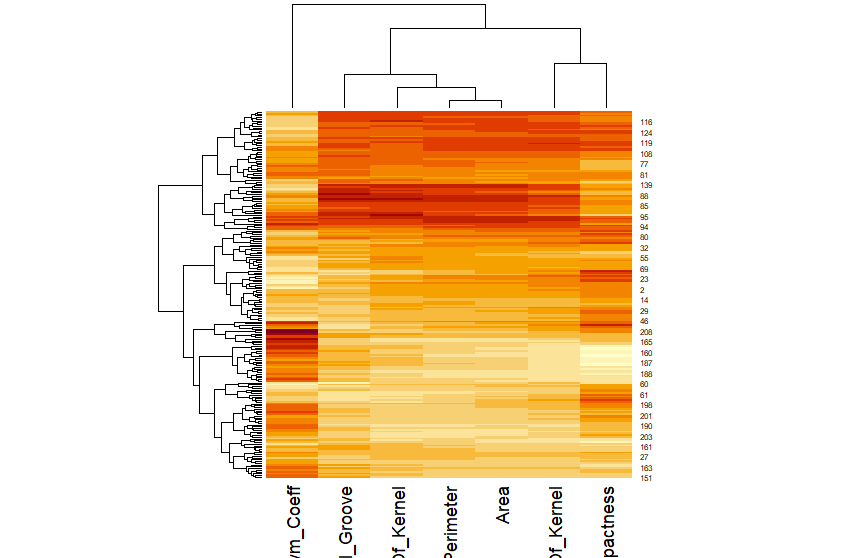
The plot depicts the data points grouped in the two clusters according to the first two principal components coordinates.

## Heatmap

**R-Code:**

heatmap(wheat\_data, scale = "none")

**Output:**



**Interpretation:**

In the plot above, we can see that there are 3 clusters formed which represents 3 different types of kernels.

## Evaluating Clustering Tendency

### Hopkins Test

**R-Code:**

# Computing Hopkins statistic for dataset

wheat\_res <- get\_clust\_tendency(wheat\_data, n = nrow(wheat\_data)-1, graph = FALSE)

wheat\_res$hopkins\_stat

**Output:**

[1] 0.8027413

**Interpretation:**

The kernel data set is not clusterable as the Hopkins statistic value = 0.80 which is far above the threshold 0.5.

### Dissimilarity Matrix

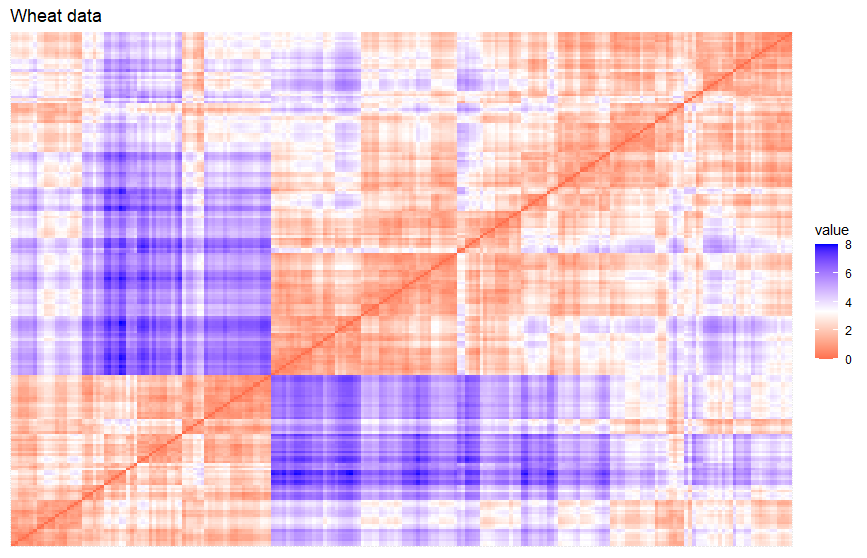
**R-Code:**

#Dissimilarity matrix (DM)

fviz\_dist(dist(wheat\_data), show\_labels = FALSE)+

labs(title = "Wheat data")

**Output:**



**Interpretation:**

The dissimilarity matrix image confirms that there is a cluster structure in the wheat data set. From the above output it seems that there are 3 possible clusters.

## Optimal Number of Clusters

### Gap Statistics

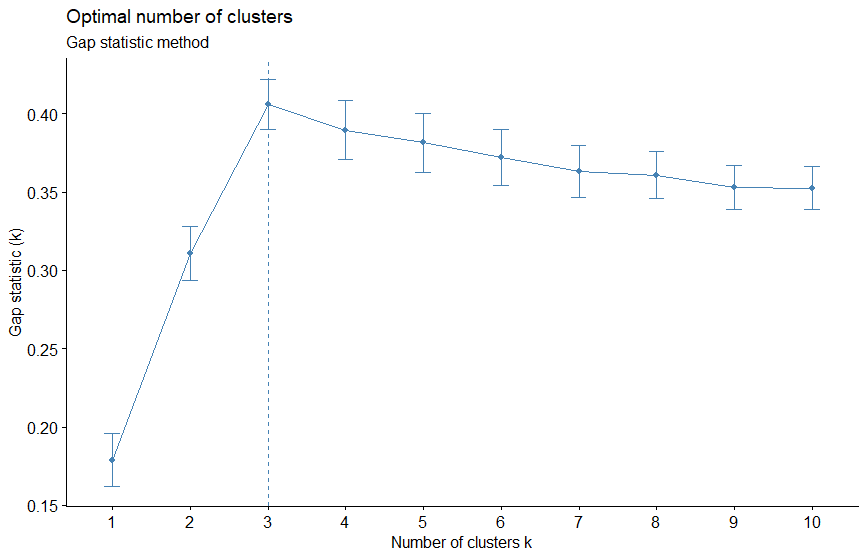
**R-Code:**

set.seed(123)

fviz\_nbclust(wheat\_data, kmeans, nstart = 25, method = "gap\_stat", nboot = 50)+

labs(subtitle = "Gap statistic method")

**Output:**



**Interpretation:**

According to Gap statistic method, the optimal number of clusters are 3.

### Hubert Index

**R-Code:**

library('NbClust')

nb <- NbClust(wheat\_data, distance = "euclidean", min.nc = 2,

max.nc = 10, method = "kmeans")

**Output:**

\*\*\* : The Hubert index is a graphical method of determining the number of clusters.

In the plot of Hubert index, we seek a significant knee that corresponds to a

significant increase of the value of the measure i.e the significant peak in Hubert

index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dindex

second differences plot) that corresponds to a significant increase of the value of

the measure.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Among all indices:

\* 10 proposed 2 as the best number of clusters

\* 11 proposed 3 as the best number of clusters

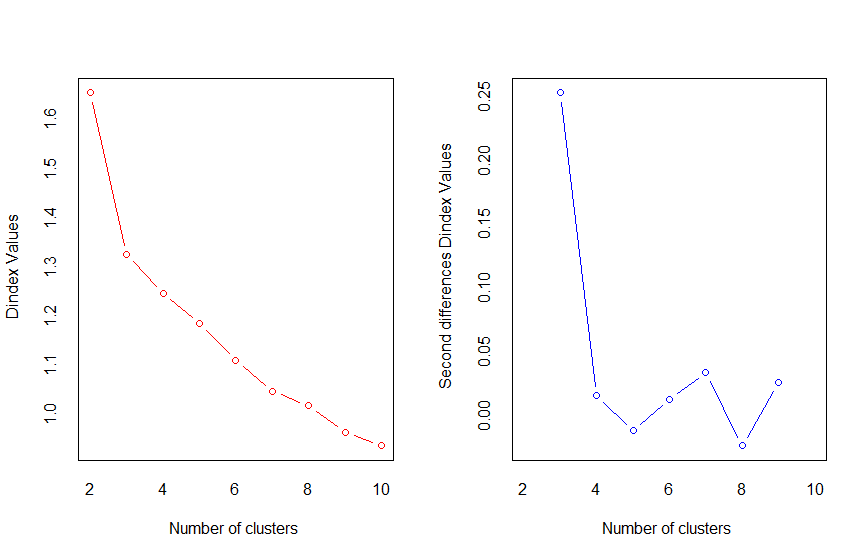
\* 1 proposed 6 as the best number of clusters

\* 1 proposed 10 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

\* According to the majority rule, the best number of clusters is 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**Interpretation:**

According to Hubert index and D index graphical method, the optimal number of clusters is 3.

### Nb\_Clust

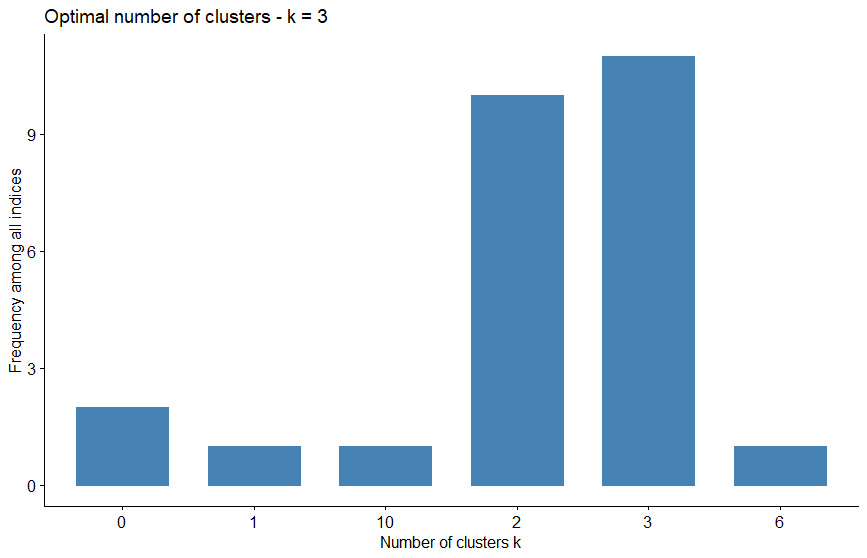
**R-Code:**

#Visualize Nb

fviz\_nbclust(nb)

**Output:**

**Output:**



**Interpretation:**

The plot shows the frequency among all indices and indicates the optimal number of clusters to be 3.

## Dunn Index

**R-Code:**

# Statistics for k-means clustering

wheat\_km\_stats <- cluster.stats(dist(wheat\_data), wheat\_km\_res$cluster)

# Dun index

wheat\_km\_stats$dunn

**Output:**

[1] 0.1052665

**Interpretation:**

The value of Dunn index is high which implies more compact and well separated clusters.

## Cluster Statistics

**R-Code:**

# Compute cluster stats

seed\_type <- as.numeric(wheat\_dataset$Seed\_Type)

clust\_stats <- cluster.stats(d = dist(wheat\_dataset), seed\_type,wheat\_km\_res$cluster)

# Corrected Rand index

clust\_stats$corrected.rand

# Computing VI index

clust\_stats$vi

**Output:**

[1] 0.7732937

[1] 0.5978132

**Interpretation:**

The values of Rand index and VI index are close to 1 which means perfect agreement between seed type and k-means clustering.

## Choosing best clustering method

**R-Code:**

# Compute clValid

clmethods <- c("hierarchical","kmeans","pam")

wheat\_valid <- clValid(wheat\_data, nClust = 2:6,

clMethods = clmethods, validation = "internal")

# Summary

summary(wheat\_valid)

**Output:**

Clustering Methods:

hierarchical kmeans pam

Cluster sizes:

2 3 4 5 6

Validation Measures:

2 3 4 5 6

hierarchical Connectivity 17.7754 39.1413 42.9798 46.1087 54.3317

Dunn 0.1132 0.0997 0.1058 0.1058 0.1157

Silhouette 0.4413 0.3760 0.3549 0.2752 0.2903

kmeans Connectivity 16.8964 39.5476 63.6409 79.9706 98.7131

Dunn 0.0870 0.1188 0.0888 0.0814 0.0814

Silhouette 0.4658 0.4007 0.3379 0.2881 0.2740

pam Connectivity 28.5028 41.4413 55.4944 73.2921 96.9325

Dunn 0.0588 0.1115 0.1114 0.0816 0.0817

Silhouette 0.4648 0.3982 0.3335 0.2678 0.2336

Optimal Scores:

Score Method Clusters

Connectivity 16.8964 kmeans 2

Dunn 0.1188 kmeans 3

Silhouette 0.4658 kmeans 2

**Interpretation:**

Kmeans clustering performs the best in each case i.e. for least Connectivity and maximum values for Dunn & Silhouette measures.

### Cluster validation: Stability measures

**R-Code:**

# Stability measures

clmethods <- c("hierarchical","kmeans","pam")

wheat\_stab <- clValid(wheat\_data, nClust = 2:6, clMethods = clmethods,

validation = "stability")

# Display only optimal Scores

optimalScores(wheat\_stab)

**Output:**

Score Method Clusters

APN 0.03511934 kmeans 2

AD 1.66075161 pam 6

ADM 0.16316514 kmeans 2

FOM 0.50952585 pam 5

**Observation:**

For the APN and ADM measures, kmeans clustering with two clusters gives the best score. For the AD measure, PAM with six clusters has the best score while for FOM measure, PAM with 5 clusters has the best score.

## SAS Analysis

### Data Preparation

**SAS code:**

proc import datafile= "/folders/myfolders/seeds.txt"

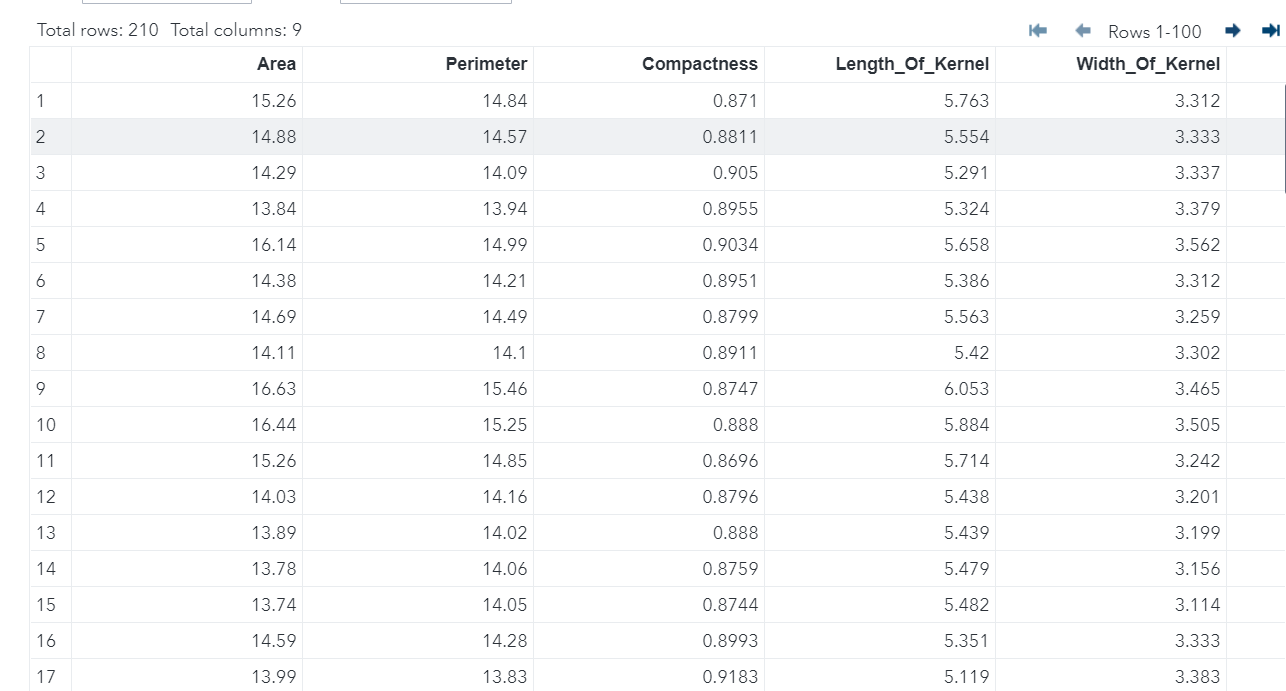
OUT=WORK.wheat

DBMS=TAB

REPLACE;

RUN;

**Output:**



### K-Means Clustering in SAS

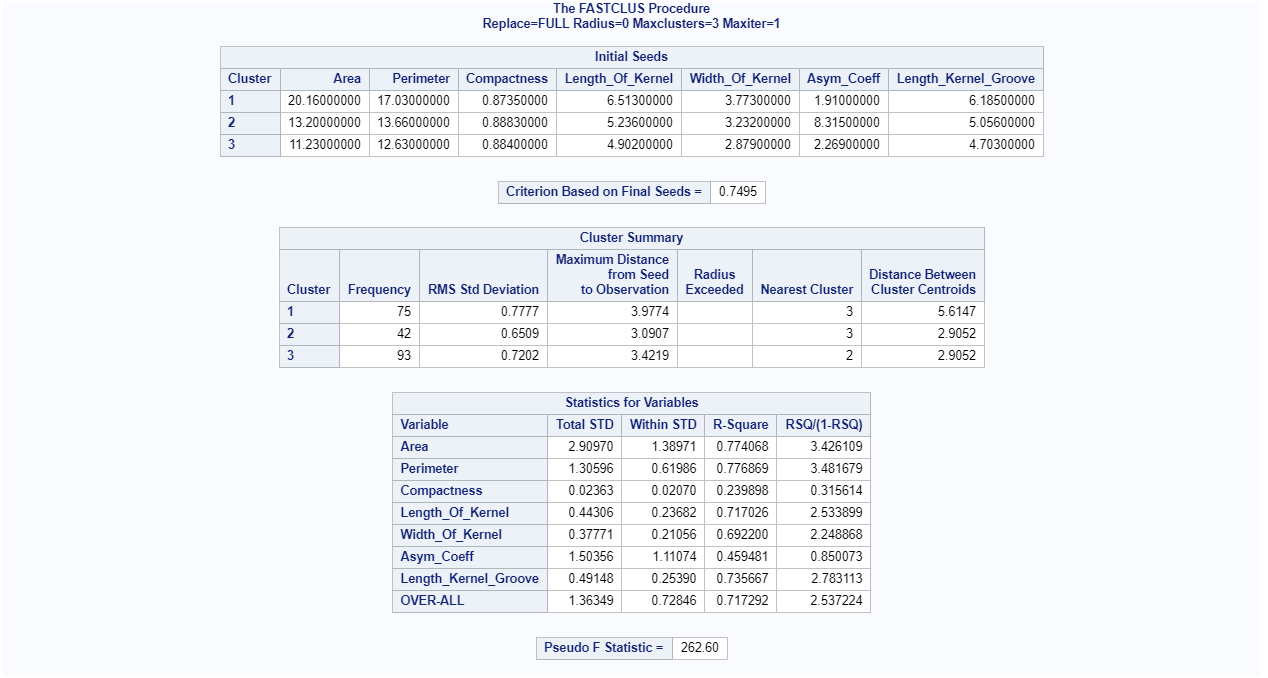
**SAS code:**

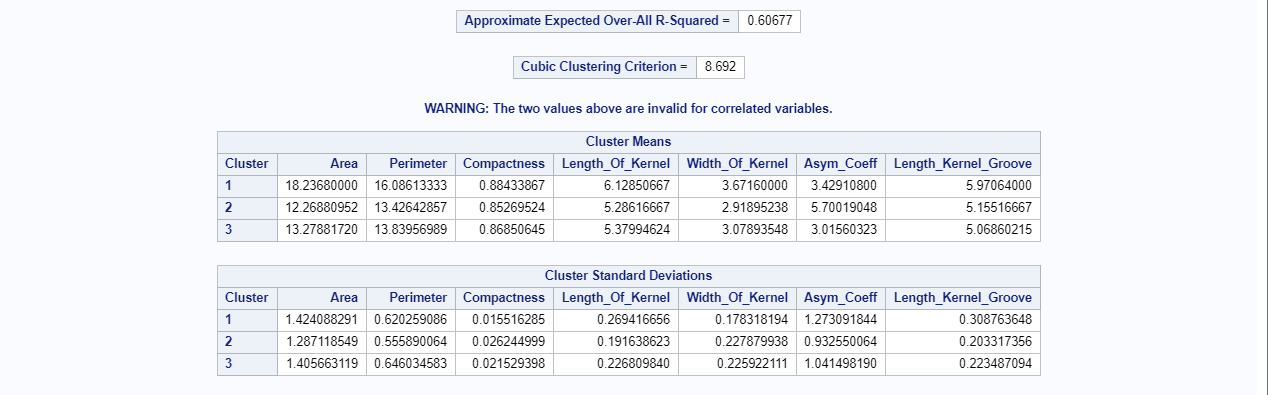
proc fastclus data=WORK.wheat maxc=3;

var Area Perimeter Compactness Length\_Of\_Kernel Width\_Of\_Kernel Asym\_Coeff Length\_Kernel\_Groove;

run;

**Output:**





**Interpretation:**

According to the kmeans clustering output, we can see cluster means, cluster summary and statistics of all the variables.