Hierarchical Clustering

Vincent Siiren

3/31/2021

## Example   
# ---  
# Question: Implement the hierarchical clustering algorithm using the Arrests dataset  
# ---  
#   
  
# Loading the data set  
# ---  
#  
data("USArrests")

# Remove any missing value (i.e, NA values for not available)  
# That might be present in the data  
# ---  
#   
df <- na.omit(USArrests)

# Previewing our dataset  
# ---  
#  
head(df)

## Murder Assault UrbanPop Rape  
## Alabama 13.2 236 58 21.2  
## Alaska 10.0 263 48 44.5  
## Arizona 8.1 294 80 31.0  
## Arkansas 8.8 190 50 19.5  
## California 9.0 276 91 40.6  
## Colorado 7.9 204 78 38.7

# Before hierarchical clustering, we can compute some descriptive statistics  
# ---  
#   
desc\_stats <- data.frame(  
 Min = apply(df, 2, min), # minimum  
 Med = apply(df, 2, median), # median  
 Mean = apply(df, 2, mean), # mean  
 SD = apply(df, 2, sd), # Standard deviation  
 Max = apply(df, 2, max) # Maximum  
)  
desc\_stats <- round(desc\_stats, 1)  
head(desc\_stats)

## Min Med Mean SD Max  
## Murder 0.8 7.2 7.8 4.4 17.4  
## Assault 45.0 159.0 170.8 83.3 337.0  
## UrbanPop 32.0 66.0 65.5 14.5 91.0  
## Rape 7.3 20.1 21.2 9.4 46.0

# We note that the variables have a large different means and variances.   
# This is explained by the fact that the variables are measured in different   
# units; Murder, Rape, and Assault are measured as the number of occurrences per 100 000 people,   
# and UrbanPop is the percentage of the stateâs population that lives in an urban area.  
# They must be standardized (i.e., scaled) to make them comparable. Recall that,   
# standardization consists of transforming the variables such that   
# they have mean zero and standard deviation one.

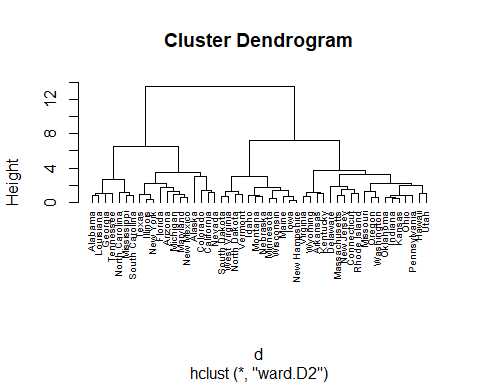
# As we donât want the hierarchical clustering result to depend to an arbitrary variable unit,   
# we start by scaling the data using the R function scale() as follows  
# ---  
#   
df <- scale(df)  
head(df)

## Murder Assault UrbanPop Rape  
## Alabama 1.24256408 0.7828393 -0.5209066 -0.003416473  
## Alaska 0.50786248 1.1068225 -1.2117642 2.484202941  
## Arizona 0.07163341 1.4788032 0.9989801 1.042878388  
## Arkansas 0.23234938 0.2308680 -1.0735927 -0.184916602  
## California 0.27826823 1.2628144 1.7589234 2.067820292  
## Colorado 0.02571456 0.3988593 0.8608085 1.864967207

# We now use the R function hclust() for hierarchical clustering  
# ---  
#   
  
# First we use the dist() function to compute the Euclidean distance between observations,   
# d will be the first argument in the hclust() function dissimilarity matrix  
# ---  
#  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc <- hclust(d, method = "ward.D2" )

# We now use the R function hclust() for hierarchical clustering  
# ---  
#   
  
# First we use the dist() function to compute the Euclidean distance between observations,   
# d will be the first argument in the hclust() function dissimilarity matrix  
# ---  
#  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc <- hclust(d, method = "ward.D2" )

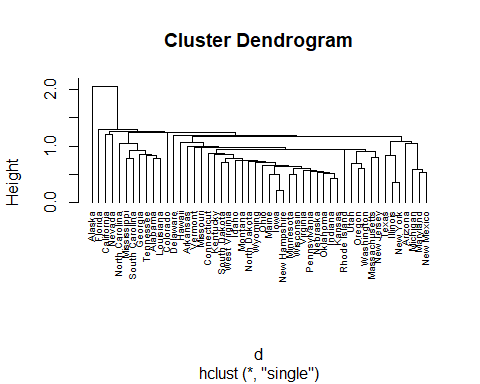
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc, cex = 0.6, hang = -1)



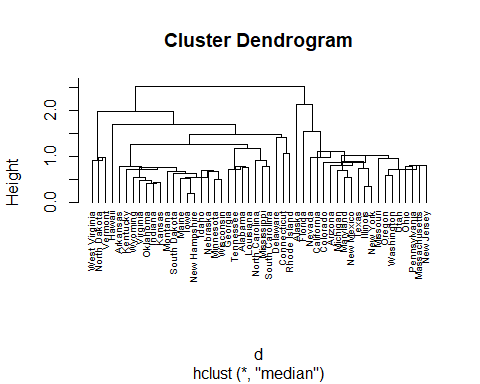
## Challenge 1

### Single

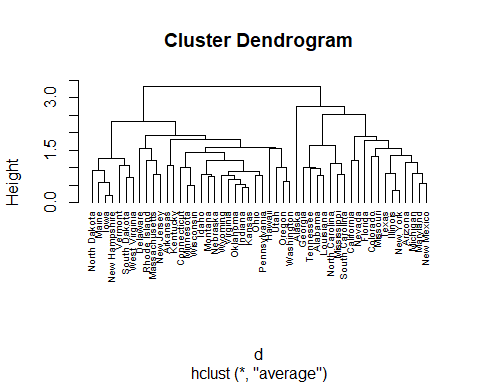
## Challenge 1  
# ---  
# Question: Using the USArrests datasets in the above example,   
# compute hierarchical clustering with other linkage methods,   
# such as single, median, average, centroid, Wardâs and McQuittyâs.  
# ---  
# Hint: You can refer to the R documentation in the suggested resources  
# ---  
  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc1 <- hclust(d, method = "single" )  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc1, cex = 0.6, hang = -1)

 ### Median

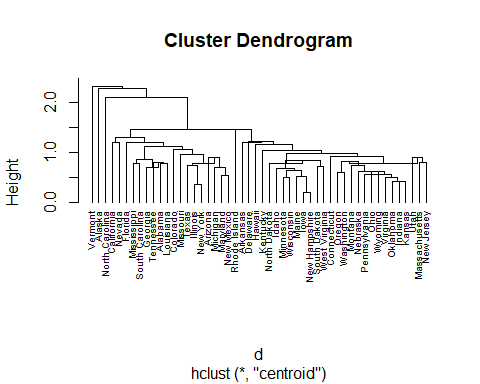
# Question: Using the USArrests datasets in the above example,   
# compute hierarchical clustering with other linkage methods,   
# such as single, median, average, centroid, Wardâs and McQuittyâs.  
# ---  
# Hint: You can refer to the R documentation in the suggested resources  
# ---  
  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc2 <- hclust(d, method = "median" )  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc2, cex = 0.6, hang = -1)

 ### Average

# Question: Using the USArrests datasets in the above example,   
# compute hierarchical clustering with other linkage methods,   
# such as single, median, average, centroid, Wardâs and McQuittyâs.  
# ---  
# Hint: You can refer to the R documentation in the suggested resources  
# ---  
  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc3 <- hclust(d, method = "average" )  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc3, cex = 0.6, hang = -1)

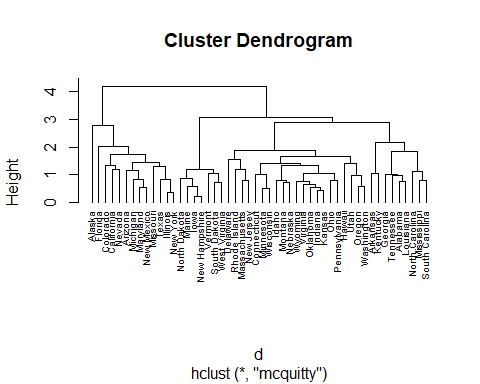
 ### Centoid

# Question: Using the USArrests datasets in the above example,   
# compute hierarchical clustering with other linkage methods,   
# such as single, median, average, centroid, Wardâs and McQuittyâs.  
# ---  
# Hint: You can refer to the R documentation in the suggested resources  
# ---  
  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc4 <- hclust(d, method = "centroid" )  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc4, cex = 0.6, hang = -1)



### MCQuitty’s

# Question: Using the USArrests datasets in the above example,   
# compute hierarchical clustering with other linkage methods,   
# such as single, median, average, centroid, Wardâs and McQuittyâs.  
# ---  
# Hint: You can refer to the R documentation in the suggested resources  
# ---  
  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc1 <- hclust(d, method = "mcquitty" )  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc1, cex = 0.6, hang = -1)



## Challenge 2

## Challenge 2  
# ---  
# Question: Perform hierarchical clustering using the mtcars dataset  
# ---  
#   
  
# Loading our dataset below  
# ---  
#   
df <- mtcars  
  
# Previewing our dataset  
# ---  
#   
head(df)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

# Before hierarchical clustering, we can compute some descriptive statistics  
# ---  
#   
desc\_stats1 <- data.frame(  
 Min = apply(df, 2, min), # minimum  
 Med = apply(df, 2, median), # median  
 Mean = apply(df, 2, mean), # mean  
 SD = apply(df, 2, sd), # Standard deviation  
 Max = apply(df, 2, max) # Maximum  
)  
desc\_stats1 <- round(desc\_stats1, 1)  
head(desc\_stats1)

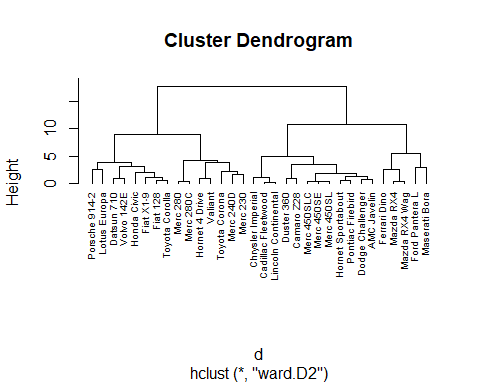
## Min Med Mean SD Max  
## mpg 10.4 19.2 20.1 6.0 33.9  
## cyl 4.0 6.0 6.2 1.8 8.0  
## disp 71.1 196.3 230.7 123.9 472.0  
## hp 52.0 123.0 146.7 68.6 335.0  
## drat 2.8 3.7 3.6 0.5 4.9  
## wt 1.5 3.3 3.2 1.0 5.4

# As we donât want the hierarchical clustering result to depend to an arbitrary variable unit,   
# we start by scaling the data using the R function scale() as follows  
# ---  
#   
df <- scale(df)  
head(df)

## mpg cyl disp hp drat  
## Mazda RX4 0.1508848 -0.1049878 -0.57061982 -0.5350928 0.5675137  
## Mazda RX4 Wag 0.1508848 -0.1049878 -0.57061982 -0.5350928 0.5675137  
## Datsun 710 0.4495434 -1.2248578 -0.99018209 -0.7830405 0.4739996  
## Hornet 4 Drive 0.2172534 -0.1049878 0.22009369 -0.5350928 -0.9661175  
## Hornet Sportabout -0.2307345 1.0148821 1.04308123 0.4129422 -0.8351978  
## Valiant -0.3302874 -0.1049878 -0.04616698 -0.6080186 -1.5646078  
## wt qsec vs am gear  
## Mazda RX4 -0.610399567 -0.7771651 -0.8680278 1.1899014 0.4235542  
## Mazda RX4 Wag -0.349785269 -0.4637808 -0.8680278 1.1899014 0.4235542  
## Datsun 710 -0.917004624 0.4260068 1.1160357 1.1899014 0.4235542  
## Hornet 4 Drive -0.002299538 0.8904872 1.1160357 -0.8141431 -0.9318192  
## Hornet Sportabout 0.227654255 -0.4637808 -0.8680278 -0.8141431 -0.9318192  
## Valiant 0.248094592 1.3269868 1.1160357 -0.8141431 -0.9318192  
## carb  
## Mazda RX4 0.7352031  
## Mazda RX4 Wag 0.7352031  
## Datsun 710 -1.1221521  
## Hornet 4 Drive -1.1221521  
## Hornet Sportabout -0.5030337  
## Valiant -1.1221521

# We now use the R function hclust() for hierarchical clustering  
# ---  
#   
  
# function dissimilarity matrix  
# ---  
#  
d <- dist(df, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc.car <- hclust(d, method = "ward.D2" )

# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc.car, cex = 0.6, hang = -1)



## Challenge 3  
# ---  
# Question: Perform hierarchical clustering using the iris dataset  
# ---  
# OUR CODE GOES BELOW   
#   
  
# Loading our dataset below  
# ---  
#   
df <- iris  
  
# Previewing our dataset  
# ---  
#   
head(df)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

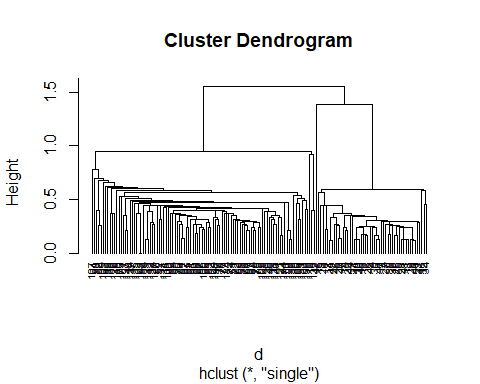
# Before hierarchical clustering, we can compute some descriptive statistics  
# ---  
#   
iris2 <- df[,-5]  
desc\_stats2 <- data.frame(  
 Min = apply(iris2, 2, min), # minimum  
 Med = apply(iris2, 2, median), # median  
 Mean = apply(iris2, 2, mean), # mean  
 SD = apply(iris2, 2, sd), # Standard deviation  
 Max = apply(iris2, 2, max) # Maximum  
)  
desc\_stats2 <- round(desc\_stats2, 1)  
print(head(desc\_stats2))

## Min Med Mean SD Max  
## Sepal.Length 4.3 5.8 5.8 0.8 7.9  
## Sepal.Width 2.0 3.0 3.1 0.4 4.4  
## Petal.Length 1.0 4.3 3.8 1.8 6.9  
## Petal.Width 0.1 1.3 1.2 0.8 2.5

# we start by scaling the data using the R function scale() as follows  
# ---  
#   
iris2 <- scale(iris2)  
head(iris2)

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## [1,] -0.8976739 1.01560199 -1.335752 -1.311052  
## [2,] -1.1392005 -0.13153881 -1.335752 -1.311052  
## [3,] -1.3807271 0.32731751 -1.392399 -1.311052  
## [4,] -1.5014904 0.09788935 -1.279104 -1.311052  
## [5,] -1.0184372 1.24503015 -1.335752 -1.311052  
## [6,] -0.5353840 1.93331463 -1.165809 -1.048667

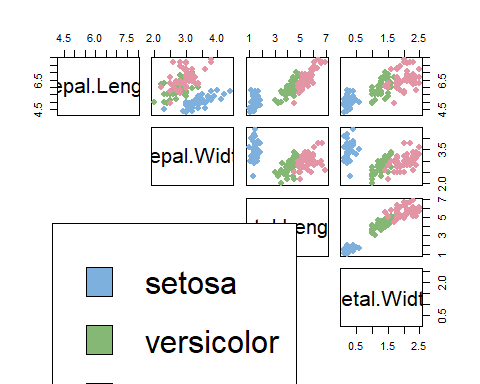
# We now use the R function hclust() for hierarchical clustering  
# ---  
#   
  
# function dissimilarity matrix  
# ---  
#  
d <- dist(iris2, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc.iris <- hclust(d, method = "single" )  
  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc.iris, cex = 0.6, hang = -1)



#### Challenge 3.1

iris2 <- iris[,-5]  
species\_labels <- iris[,5]  
library(colorspace) # get nice colors  
species\_col <- rev(rainbow\_hcl(3))[as.numeric(species\_labels)]

# Plot a SPLOM:  
pairs(iris2, col = species\_col,  
 lower.panel = NULL,  
 cex.labels=2, pch=19, cex = 1.2)  
  
# Add a legend  
par(xpd = TRUE)  
legend(x = 0.05, y = 0.4, cex = 2,  
 legend = as.character(levels(species\_labels)),  
 fill = unique(species\_col))



par(xpd = NA)

d\_iris <- dist(iris2) # method="man" # is a bit better  
hc\_iris <- hclust(d\_iris, method = "complete")  
iris\_species <- rev(levels(iris[,5]))

library(dendextend)

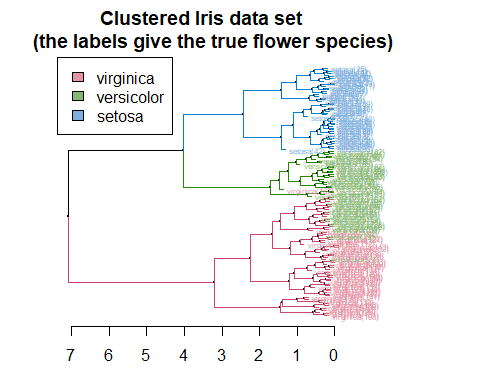
##   
## ---------------------  
## Welcome to dendextend version 1.14.0  
## Type citation('dendextend') for how to cite the package.  
##   
## Type browseVignettes(package = 'dendextend') for the package vignette.  
## The github page is: https://github.com/talgalili/dendextend/  
##   
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues  
## 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

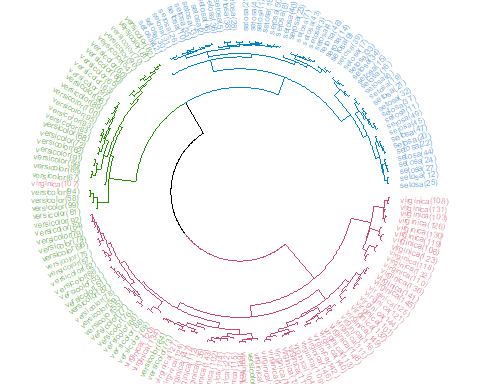
dend <- as.dendrogram(hc\_iris)  
# order it the closest we can to the order of the observations:

# Color the branches based on the clusters:  
dend <- color\_branches(dend, k=3) #, groupLabels=iris\_species)  
  
# Manually match the labels, as much as possible, to the real classification of the flowers:  
labels\_colors(dend) <-  
 rainbow\_hcl(3)[sort\_levels\_values(  
 as.numeric(iris[,5])[order.dendrogram(dend)]  
 )]  
  
# We shall add the flower type to the labels:  
labels(dend) <- paste(as.character(iris[,5])[order.dendrogram(dend)],  
 "(",labels(dend),")",   
 sep = "")  
# We hang the dendrogram a bit:  
dend <- hang.dendrogram(dend,hang\_height=0.1)  
# reduce the size of the labels:  
# dend <- assign\_values\_to\_leaves\_nodePar(dend, 0.5, "lab.cex")  
dend <- set(dend, "labels\_cex", 0.5)  
# And plot:  
par(mar = c(3,3,3,7))  
plot(dend,   
 main = "Clustered Iris data set  
 (the labels give the true flower species)",   
 horiz = TRUE, nodePar = list(cex = .007))  
legend("topleft", legend = iris\_species, fill = rainbow\_hcl(3))



# Requires that the circlize package will be installed  
par(mar = rep(0,4))  
circlize\_dendrogram(dend)

## Loading required namespace: circlize



## Challenge 4

## Challenge 4  
# ---  
# Question: Perform hierarchical cluster analysis on the given protein consumption.  
# ---  
# Dataset url = http://bit.ly/HierarchicalClusteringDataset  
# ---  
library(MASS)  
library(data.table)

##   
## Attaching package: 'data.table'

## The following object is masked from 'package:dendextend':  
##   
## set

data <- fread("http://bit.ly/HierarchicalClusteringDataset")  
head(data)

## Country RedMeat WhiteMeat Eggs Milk Fish Cereals Starch Nuts Fr&Veg  
## 1: Albania 10.1 1.4 0.5 8.9 0.2 42.3 0.6 5.5 1.7  
## 2: Austria 8.9 14.0 4.3 19.9 2.1 28.0 3.6 1.3 4.3  
## 3: Belgium 13.5 9.3 4.1 17.5 4.5 26.6 5.7 2.1 4.0  
## 4: Bulgaria 7.8 6.0 1.6 8.3 1.2 56.7 1.1 3.7 4.2  
## 5: Czechoslovakia 9.7 11.4 2.8 12.5 2.0 34.3 5.0 1.1 4.0  
## 6: Denmark 10.6 10.8 3.7 25.0 9.9 21.9 4.8 0.7 2.4

# Before hierarchical clustering, we can compute some descriptive statistics  
# ---  
#   
data2 <- data[,-1]  
desc\_stats2 <- data.frame(  
 Min = apply(data2, 2, min), # minimum  
 Med = apply(data2, 2, median), # median  
 Mean = apply(data2, 2, mean), # mean  
 SD = apply(data2, 2, sd), # Standard deviation  
 Max = apply(data2, 2, max) # Maximum  
)  
desc\_stats2 <- round(desc\_stats2, 1)  
print(head(desc\_stats2))

## Min Med Mean SD Max  
## RedMeat 4.4 9.5 9.8 3.3 18.0  
## WhiteMeat 1.4 7.8 7.9 3.7 14.0  
## Eggs 0.5 2.9 2.9 1.1 4.7  
## Milk 4.9 17.6 17.1 7.1 33.7  
## Fish 0.2 3.4 4.3 3.4 14.2  
## Cereals 18.6 28.0 32.2 11.0 56.7

# we start by scaling the data using the R function scale() as follows  
# ---  
#   
data2 <- scale(data2)  
head(data2)

## RedMeat WhiteMeat Eggs Milk Fish Cereals  
## [1,] 0.08126490 -1.7584889 -2.1796385 -1.15573814 -1.20028213 0.9159176  
## [2,] -0.27725673 1.6523731 1.2204544 0.39237676 -0.64187467 -0.3870690  
## [3,] 1.09707621 0.3800675 1.0415022 0.05460623 0.06348211 -0.5146342  
## [4,] -0.60590157 -0.5132535 -1.1954011 -1.24018077 -0.90638347 2.2280161  
## [5,] -0.03824231 0.9485445 -0.1216875 -0.64908235 -0.67126454 0.1869740  
## [6,] 0.23064892 0.7861225 0.6835976 1.11013912 1.65053488 -0.9428885  
## Starch Nuts Fr&Veg  
## [1,] -2.2495772 1.2227536 -1.35040507  
## [2,] -0.4136872 -0.8923886 0.09091397  
## [3,] 0.8714358 -0.4895043 -0.07539207  
## [4,] -1.9435955 0.3162641 0.03547862  
## [5,] 0.4430614 -0.9931096 -0.07539207  
## [6,] 0.3206688 -1.1945517 -0.96235764

# We now use the R function hclust() for hierarchical clustering  
# ---  
#   
  
# function dissimilarity matrix  
# ---  
#  
d <- dist(data2, method = "euclidean")  
  
# We then hierarchical clustering using the Ward's method  
# ---  
#   
res.hc.iris <- hclust(d, method = "single" )  
  
  
# Lastly, we plot the obtained dendrogram  
# ---  
#   
plot(res.hc.iris, cex = 0.6, hang = -1)

