Cluster course: Hierarchical and non hierarchical clustering R code

By: S. Civit

Contents

Hierarchical clustering

Dataset

Fangataufa és un atolò en "format rectangular" amb una superfície total de 45 km². Originalment no tenia cap pas a la llacuna interior, però les forces armades franceses van volar 400 m d'esculls per obrir un pas que facilités el programa de proves nuclears. És una zona militar amb accés prohibit sense autorització.

Entre el 1966 i 1996 s'hi van fer 5 explosions nuclears atmosfèriques i 10 subterrànies a una profunditat entre 500 i 700 m sota la llacuna.

Les dades procedeixen d'un estudi que pretenia detectar canvis temporals en l'estructura de la comunitat gasteròpoda en els esculls del atolò. Els mol · luscs dels esculls van ser parcialment o totalment esborrats per la calor de les proves nuclears i van ser recolonitzats per larves oceàniques. En tots els esculls, la composició de la comunitat abans de les proves era molt diferent de la que va evolucionar fins a després.

Cal doncs entendre les dades i per tant la **matriu multivariant** de dades

- Files zones del mostreig, que correspon a zones del atolò
- Columnes: espècies de gasteròpods

```
data<-read.table("fangataufa.txt",sep="\t",header=TRUE)
```

Distància

- Elecció de la distància com element clau. Comencem de forma incorrecta (en aquest cas amb la distància euclidia)

```
data2<-scale(data[,2:14])# Standardise the variables
is.matrix(data2)
```

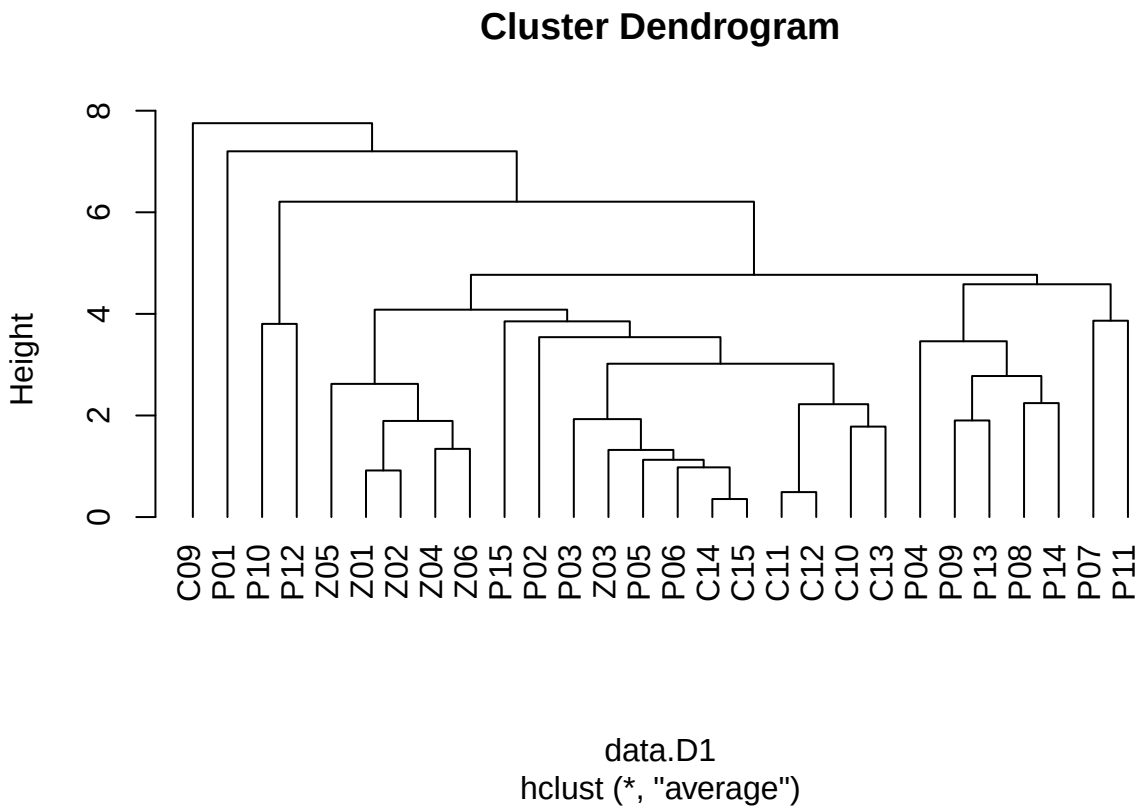
```
## [1] TRUE
```

```r
data.D1 = dist(data2, method="eucl") # Compute Euclidean distance (for illustrative purposes)
```

Dendograma

- Representem mitjançant el mètode UPGM i mètode de Ward
- Determimen la correlació coefenètica (distorsió de les dades originals a les observades en el dendograma)

```r
# Agglomerative clustering, UPGMA method:
clusterAV = hclust(data.D1, method="average")#average=UPGMA
clusterW<-hclust(data.D1^2, method="ward.D2")#ward

# Plot the dendrogram
plot(clusterAV, hang=-1, labels=data[,1])
```




```r
# Cophenetic distances of the dendrogram
Ours.coph = cophenetic(clusterAV)
cor(data.D1, Ours.coph)
```

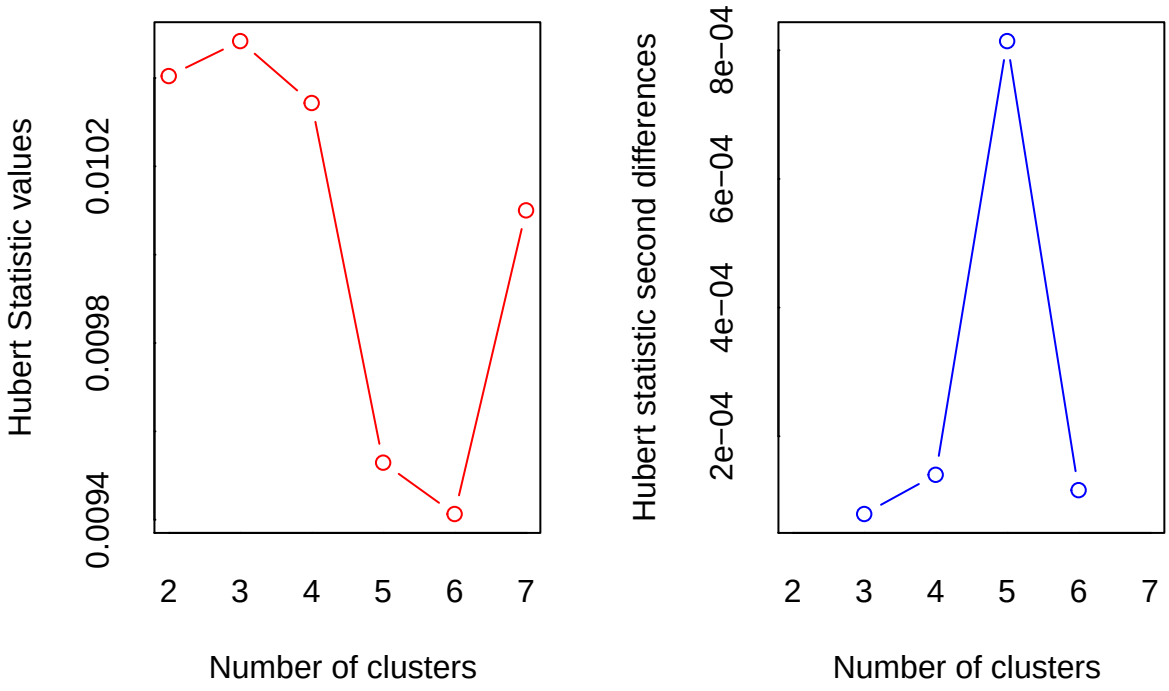
```
## [1] 0.851541
```

Nombre de grups?

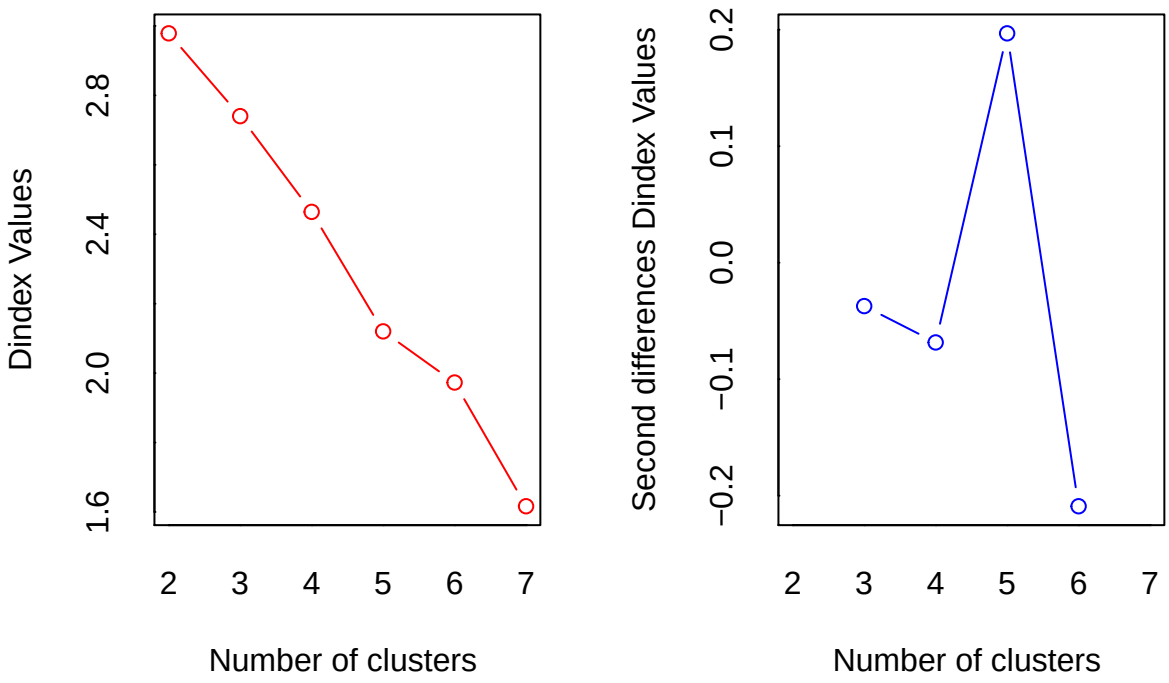
Funció **NbClust ()** [al paquet NbClust R] (Charrad et al. 2014): proporciona 30 índexs per determinar el nombre rellevant de clúster a partir de les combinacions de nombre de clústers, mesures de distància i mètodes d'agrupació.

```r
#install.packages("NbClust")
library(NbClust)

NbClust(data = data2, diss = NULL, distance = "euclidean",
        min.nc = 2, max.nc = 7, method = "average")
```

```
## Warning in pf(beale, pp, df2): NaNs produced
```



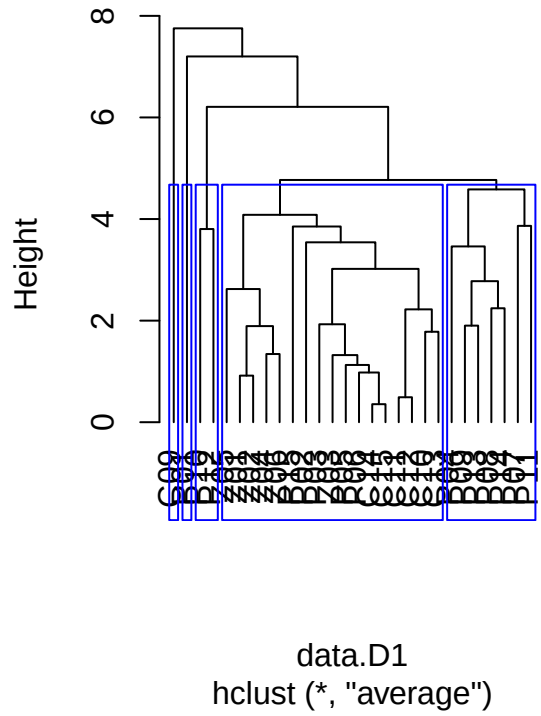

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 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:
## * 6 proposed 2 as the best number of clusters
## * 5 proposed 3 as the best number of clusters
## * 3 proposed 4 as the best number of clusters
## * 3 proposed 5 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
## * 5 proposed 7 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************

## $All.index
##        KL      CH Hartigan     CCC    Scott      Marriot    TrCovW   TraceW
## 2 0.5406  4.1846   4.0962 -2.8921 127.1457 3.437719e+13 1223.8504 302.3397
## 3 0.5686  4.2981   5.8870 -3.6442 195.8445 6.651199e+12  977.0556 261.1903
## 4 0.7255  5.2824   9.1556 -3.2498 292.6396 3.727648e+11  699.4205 211.4077
## 5 3.1174  7.4386   3.5009 -0.8025 366.3150 4.192873e+10  337.1872 153.0295
```

```
## 6 0.3963  7.2283  10.3531 -0.7780 436.8156 4.868245e+09  289.1568 132.8134
## 7 6.8481 10.1027   1.8939  2.7847 495.8941 8.033927e+08   89.1037  90.3127
##   Friedman  Rubin Cindex     DB Silhouette   Duda Pseudot2  Beale Ratkowsky
## 2 123.9969 1.1609 0.4941 0.4711     0.4344 0.9841   0.4044 0.1382    0.1691
## 3 133.5228 1.3438 0.5125 0.4726     0.4074 0.8094   5.6516 2.0089    0.2276
## 4 175.5469 1.6603 0.4976 0.6798     0.3729 0.7141   8.8089 3.4034    0.2845
## 5 218.5080 2.2937 0.5080 0.9195     0.3281 0.5554   4.0025 5.9280    0.3240
## 6 303.7856 2.6428 0.4940 0.8264     0.3319 0.5764  11.0244 6.1230    0.3114
## 7 314.3756 3.8865 0.5142 0.8129     0.3744 3.7829   0.0000 0.0000    0.3242
##       Ball Ptbiserial   Frey McClain   Dunn Hubert SDindex Dindex   SDbw
## 2 151.1699     0.4445 1.2596  0.0447 0.5867 0.0104  0.5398 2.9787 0.4687
## 3  87.0634     0.5890 1.8252  0.0936 0.6375 0.0105  0.4557 2.7402 0.2886
## 4  52.8519     0.6862 1.8754  0.2101 0.6077 0.0103  0.5506 2.4644 0.3805
## 5  30.6059     0.6252 0.6039  0.8140 0.3519 0.0095  0.7976 2.1202 0.4575
## 6  22.1356     0.6272 0.5995  0.8938 0.3519 0.0094  0.7621 1.9729 0.4198
## 7  12.9018     0.5929 0.3878  1.6722 0.2499 0.0101  0.7656 1.6165 0.3892
##
## $All.CriticalValues
##   CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2         0.7171             9.8613       0.9999
## 3         0.7127             9.6760       0.0197
## 4         0.7029             9.2971       0.0001
## 5         0.4917             5.1697       0.0000
## 6         0.6563             7.8570       0.0000
## 7         0.0916             0.0000          NaN
##
## $Best.nc
##                       KL      CH Hartigan    CCC   Scott      Marriot   TrCovW
## Number_clusters 7.0000  7.0000   7.0000 7.0000  4.0000 3.000000e+00   5.0000
## Value_Index     6.8481 10.1027   8.4592 2.7847 96.7951 2.144756e+13 362.2333
##                   TraceW Friedman   Rubin Cindex     DB Silhouette   Duda
## Number_clusters  5.0000   6.0000  5.0000  6.000 2.0000     2.0000 2.0000
## Value_Index     38.1619  85.2776 -0.2842  0.494 0.4711     0.4344 0.9841
##                 PseudoT2  Beale Ratkowsky    Ball PtBiserial   Frey McClain
## Number_clusters   2.0000 2.0000    7.0000  3.0000     4.0000 4.0000  2.0000
## Value_Index       0.4044 0.1382    0.3242 64.1064     0.6862 1.8754  0.0447
##                   Dunn Hubert SDindex Dindex   SDbw
## Number_clusters 3.0000      0  3.0000      0 3.0000
## Value_Index     0.6375      0  0.4557      0 0.2886
##
## $Best.partition
##  [1] 1 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
```

```
#How many groups we have?. It depends Suppose k= 5 groups
groups.3 <- cutree(clusterAV,5)  #3 grups?
plot(clusterAV, hang=-1, labels=data[,1], main="Cluster Dendogram (UPGMA method), k=5")
rect.hclust(clusterAV,k=5, border="blue", )
```



Caracterització dels grups

```
#number of elements and elements within groups
table(groups.3)
```

```
## groups.3
##  1  2  3  4  5
##  1 17  1  7  2
```

```
g1=data[groups.3==1,]

# Mean profile G1
G1mean<-round(apply(g1[-1],2,mean),3)
```

Possible exercici:

- 1. Repeat the analysis plotting Hierarchical clustering dendogram and coefenetic correlation according to "real distance" for this dataset.
- 2. Decide how many groups and characterize them and show a 2D plot

Non hierarchical clustering: Kmeans

Dataset

In this data set we observe the composition of different wines. Given a set of observations (x1,x2,.,xn), where each observation is a d-dimensional real vector,

```r
#install.packages('rattle')
data(wine, package='rattle')
head(wine)
```

```
##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06          0.28
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76          0.26
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24          0.30
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49          0.24
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69          0.39
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39          0.34
##   Proanthocyanins Color  Hue Dilution Proline
## 1            2.29  5.64 1.04     3.92    1065
## 2            1.28  4.38 1.05     3.40    1050
## 3            2.81  5.68 1.03     3.17    1185
## 4            2.18  7.80 0.86     3.45    1480
## 5            1.82  4.32 1.04     2.93     735
## 6            1.97  6.75 1.05     2.85    1450
```

Distància

```r
wine.stand <- scale(wine[-1])  # To standarize the variables
```

Graph

```r
# K-Means
k.means.fit <- kmeans(wine.stand, 3) # k = 3

k.means.fit$cluster
```

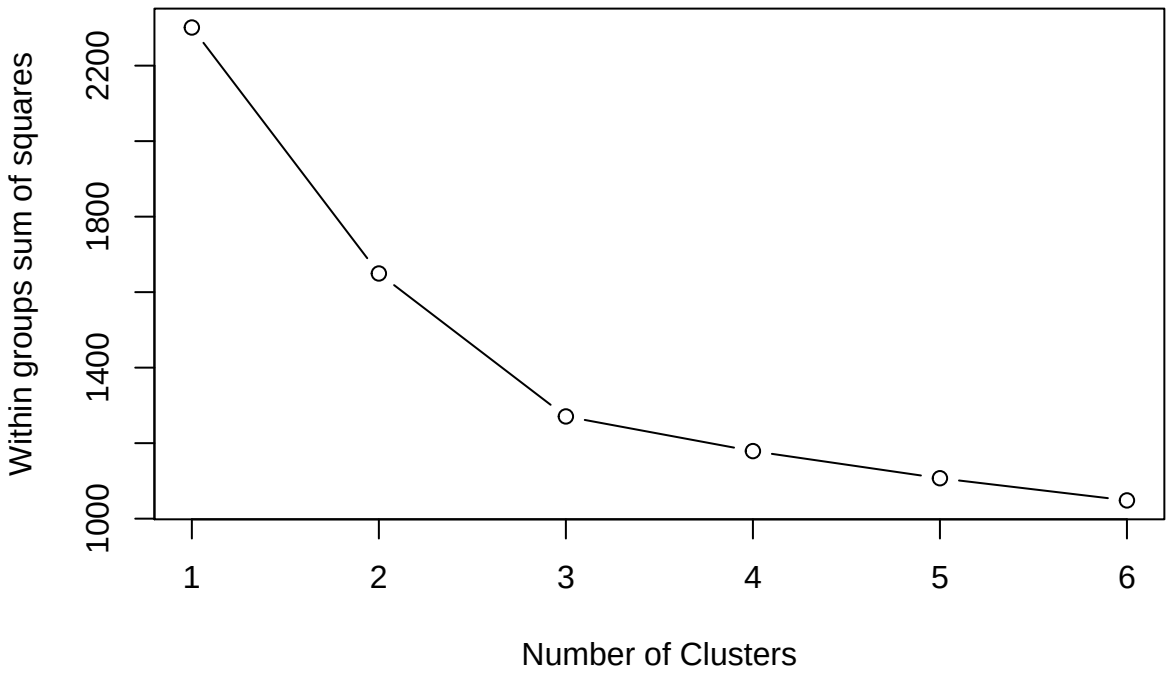
```
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 2
##  [75] 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [112] 3 3 3 3 3 3 3 1 3 3 2 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Nombre de grups?

Una qüestió fonamental és com determinar el valor del paràmetre k. Si observem el percentatge de variància explicat en funció del nombre de clústers: s'ha de triar un nombre de clústers de manera que afegir un altre clúster no doni una modelització molt millor de les dades. Més exactament, si es traça el percentatge de variància explicat pels clústers contra el nombre de clústers, els primers clústers afegiran molta informació (expliquen molta variància), però en algun moment el guany marginal caurà, donant un angle al gràfic.

- opció 1: En aquest moment es tria el nombre de clústers, d'aquí el "criteri del colze".

```r
wssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}

wssplot(wine.stand, nc=6)
```

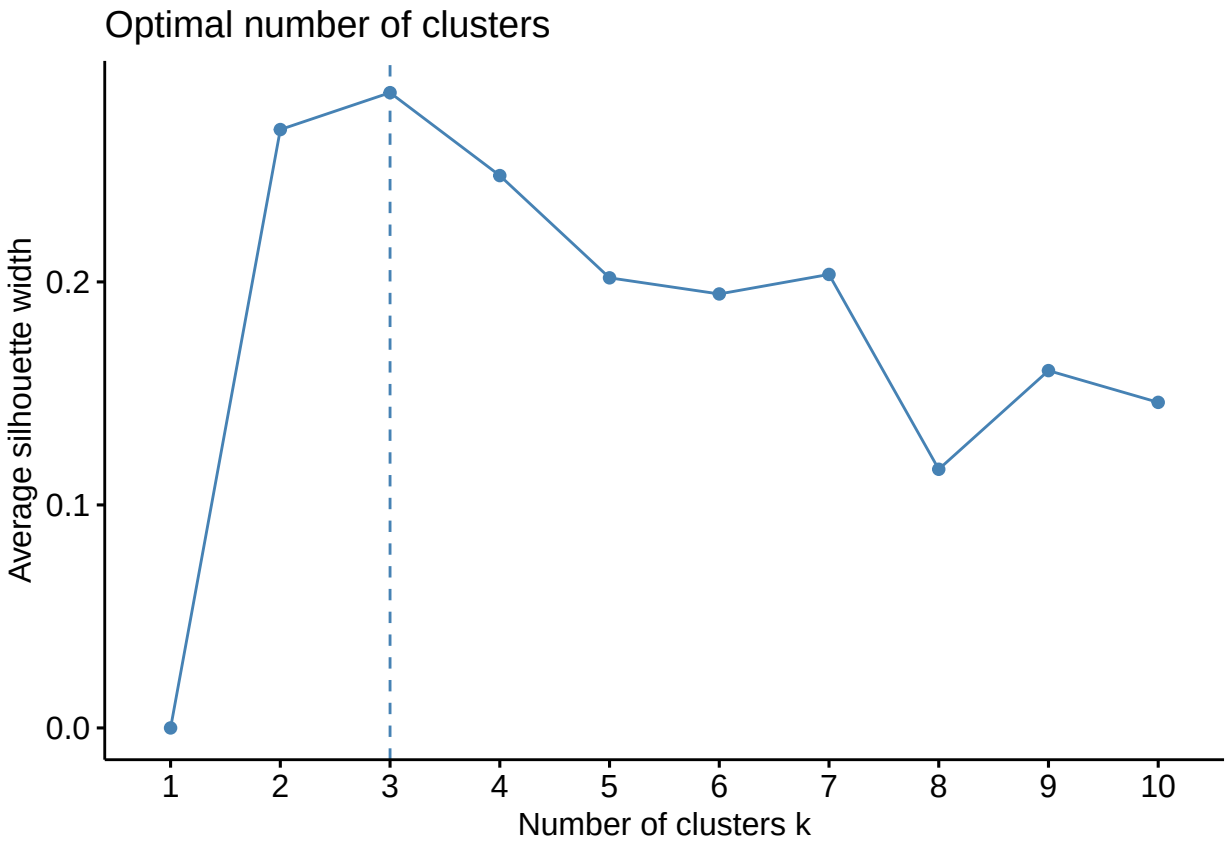


- Opció 2: Ús de la **library factoextra**

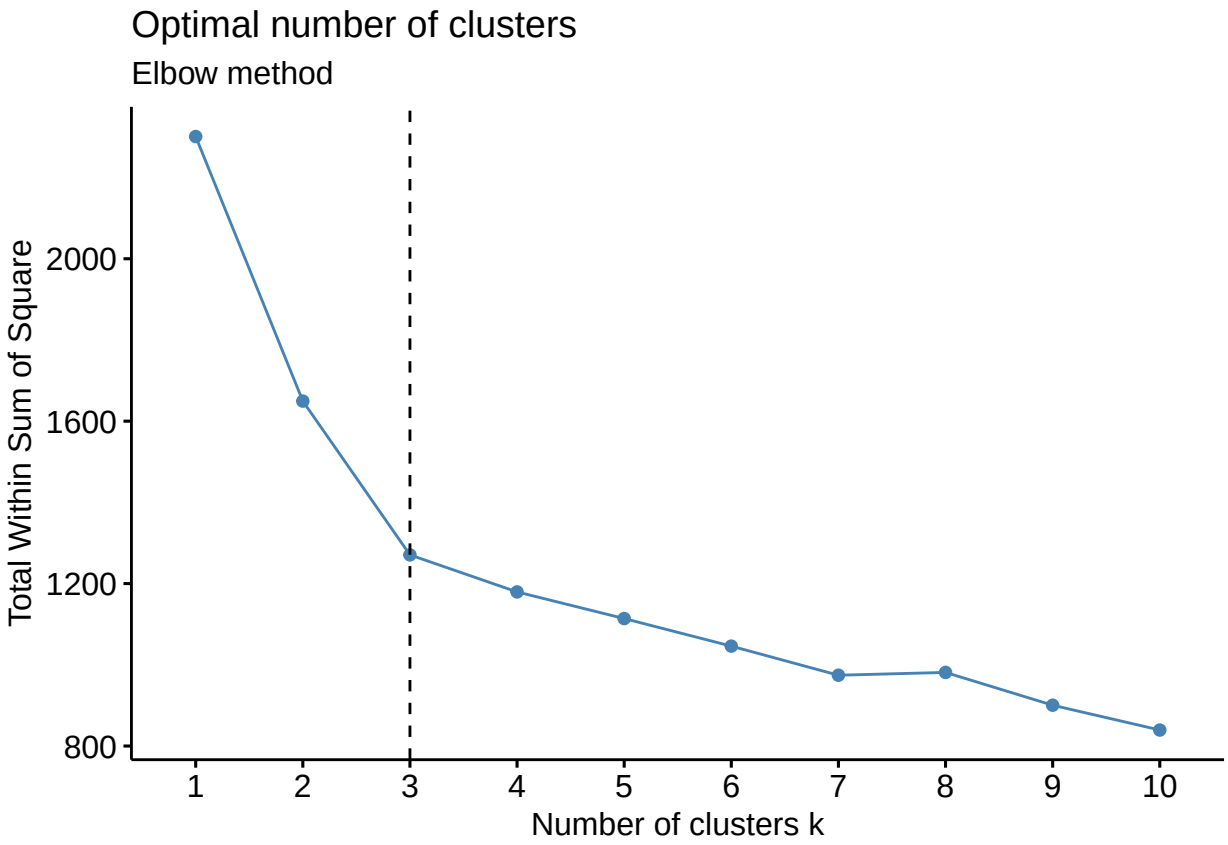
```
#install.packages(factoextra)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.6.3
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.6.3
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

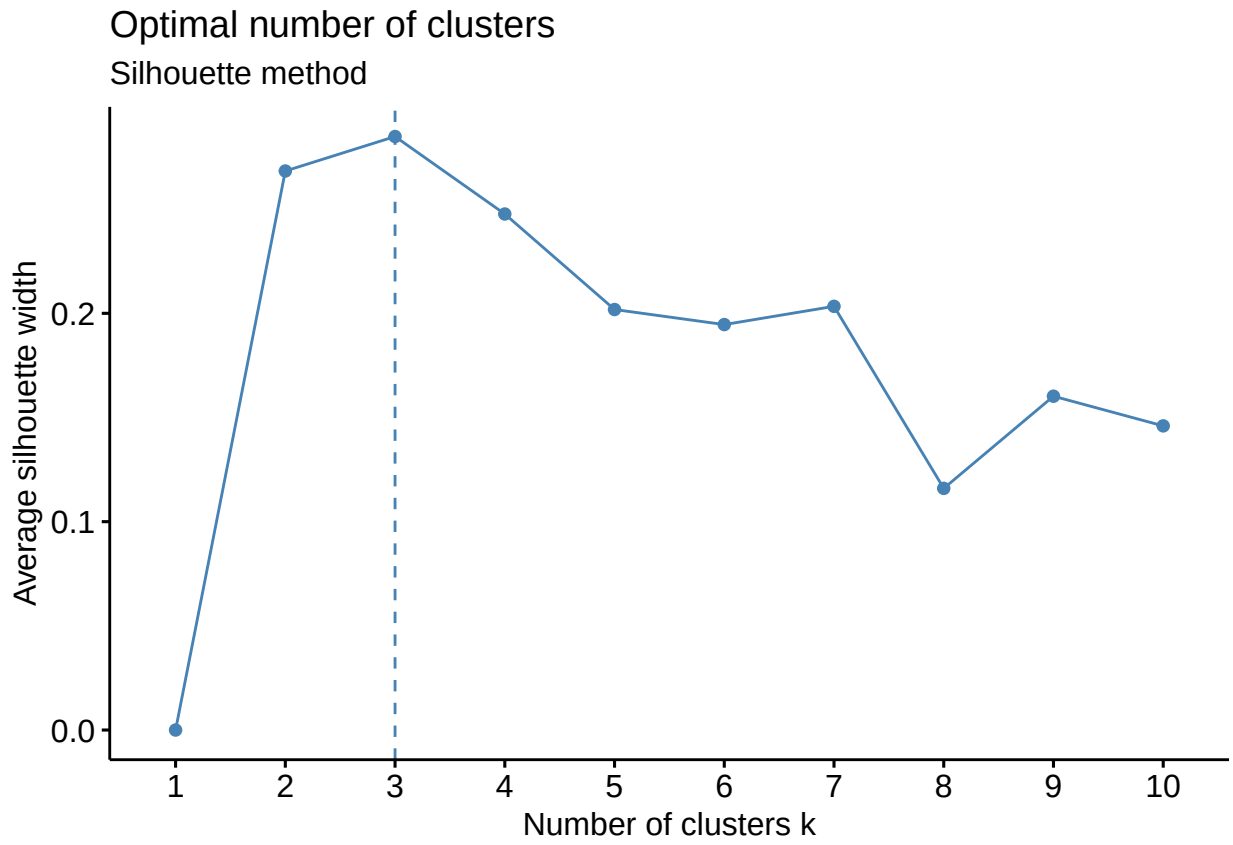
```
library(ggplot2)

fviz_nbclust(wine.stand, kmeans, method = c("silhouette", "wss", "gap_stat"))
```



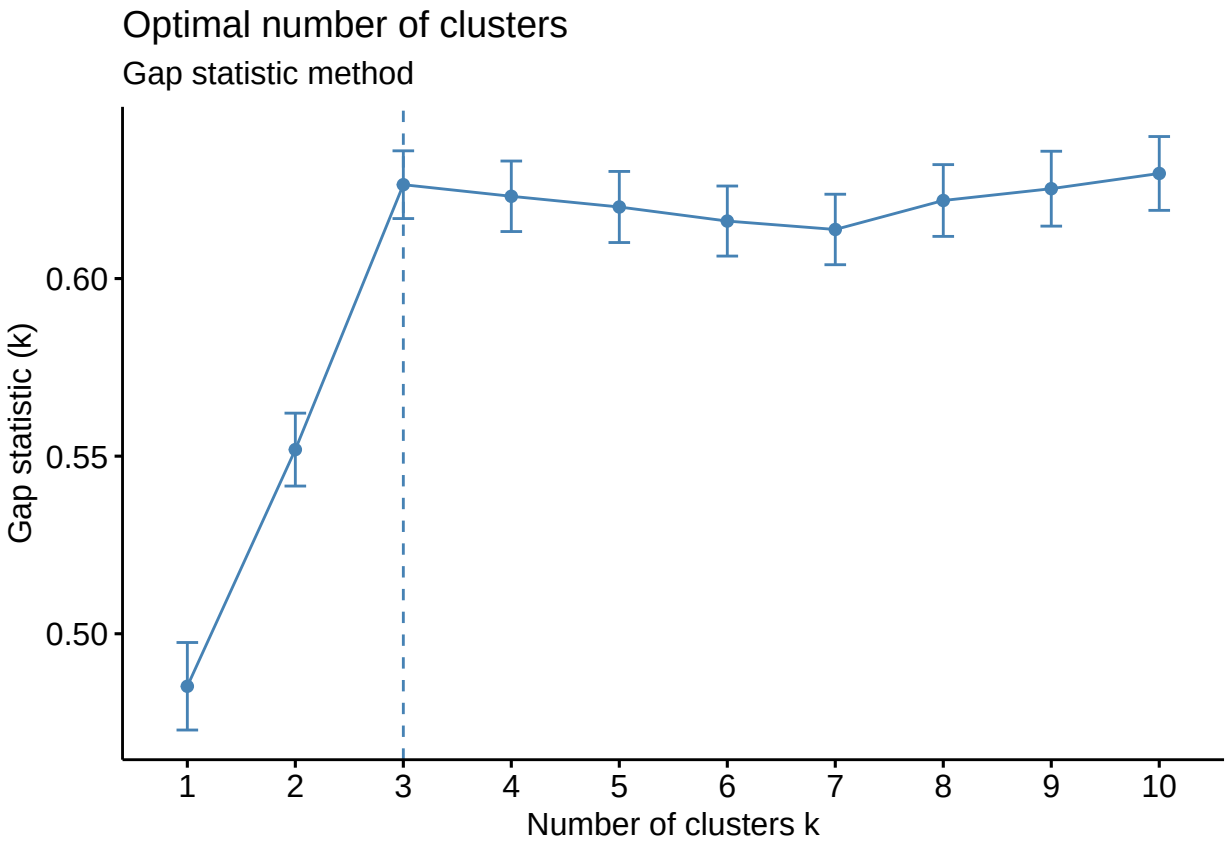

```
# Elbow method
fviz_nbclust(wine.stand, kmeans, method = "wss") +
    geom_vline(xintercept = 3, linetype = 2)+
  labs(subtitle = "Elbow method")
```




```
# Silhouette method
fviz_nbclust(wine.stand, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```




```
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(wine.stand, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```



Caracterització dels grups

```
# Centroids:
k.means.fit$centers
```

```
##      Alcohol      Malic        Ash Alcalinity   Magnesium     Phenols
## 1  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548
## 2  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724
## 3 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891
##    Flavanoids Nonflavanoids Proanthocyanins      Color        Hue   Dilution
## 1 -1.21182921    0.72402116     -0.77751312  0.9388902 -1.1615122 -1.2887761
## 2  0.97506900   -0.56050853      0.57865427  0.1705823  0.4726504  0.7770551
## 3  0.02075402   -0.03343924      0.05810161 -0.8993770  0.4605046  0.2700025
##      Proline
## 1 -0.4059428
## 2  1.1220202
## 3 -0.7517257
```

```
# Cluster size:
k.means.fit$size
```

```
## [1] 51 62 65
```

Conexió amb altres tècniques multivariants PCA

library cluster ens permeten representar (amb l'ajut de PCA) la solució de clúster en 2 dimensions

```
library(cluster)
clusplot(wine.stand, k.means.fit$cluster, main='2D representation of the Cluster solution',
         color=TRUE, shade=TRUE,
         labels=2, lines=0)
```