Data Analysis

Christophe Ambroise

Section 1

Clustering

Introduction

A widespread confusion exists between the terms classification and clustering:

- Classification presupposes the existence of classes with certain objects are known, while
- Clustering tries to discover a class structure which is “natural” to the data.

In the literature related to pattern recognition, the distinction between the two approaches is often referred to by the terms “supervised” and “unsupervised” learning.

Objectives

Clustering can have different motivations:

- compress information, to describe in a simplified way large masses of data,
- structure a set of knowledge,
- reveal structures, hidden causes,
- make a diagnosis . . .

Similarity

the fist step to any clustering is to define a measure resemblance between objects (shapes vectors). Traditionally two approaches are envisaged:

- *monothetic* : we can say that two objects are similar if they share a certain feature. Basis of the Aristotelian approach [@Sutcliffe1994]. All objects in the same class share a number of characteristics (e.g. "All men are mortal");
- *polythetic* : we can also measure the similarity by using a measure of proximity (distance, dissimilarity). In this case the notion of resemblance is measured more fuzzy and two objects of the same class will have "close" characteristics within the meaning of the measure used.

Context of this lecture

polythetic classification

Classes or groups

A classification leads to the distribution of the set of vectors forms in different *homogeneous classes*. The definition of a class and the relations between classes can be very varied. In this chapter we will focus on both main classification structures:

- partition,
- hierarchy.

Partitions

Définition

Ω being a finite set, a set $P = (\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K)$ non-empty parts of Ω is a partition if:

1. $\forall i \neq j$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$,
2. $\cup_i \mathcal{C}_i = \Omega$.

In a set $\Omega = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ partitioned into K classes, each element of the set belongs to a class and only one. A practical way of describing this P partition consists of using a matrix notation.

Let $\boldsymbol{C}(P)$ the characteristic matrix of the partition $P = (\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K)$ (ou matrice de

$$\boldsymbol{C}(P) = \boldsymbol{C} = \begin{pmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{N1} & \cdots & c_{NK} \end{pmatrix}$$

where $c_{ik} = 1$ iff $\boldsymbol{x}_i \in \mathcal{C}_k$, and $c_{ik} = 0$ otherwise.

Note that the sum of the i th row is equal to 1 (an element belongs to a single class) and the sum of the values of the k th column is worth n_k the number of elements of the class $\mathcal{C}_k$. On a donc $\sum_{k=1}^{K} n_k = N$.

Hard and Fuzzy partition

The notion of hard partition is based on a set design classic. Considering the work of [@Zadeh1965] on the sets fuzzy, a definition of the concept of fuzzy partition seems "Natural". The fuzzy classification, developed at the beginning of 1970s [@Ruspini1969], generalizes an approach classical classification by broadening the concept belonging to a class.

Fuzzy sets

As part of the classic set design, an individual $\boldsymbol{x}_i$ belongs to or does not belong to a given set $\mathcal{C}_k$. In the theory of fuzzy subsets, an individual can belong to several classes with different degrees of membership.

Fuzzy clustering

In classification this amounts to authorize the vectors forms to belong to all classes, which results in the releasing the binarity constraint on the coefficients belonging to c_{ik}. A fuzzy partition is defined by a fuzzy classification matrix $\boldsymbol{C} = \{c_{ik}\}$ checking the following conditions:

1. $\forall k = 1..K,\ \forall \boldsymbol{x}_i \in \Omega,\ c_{ik} \in [0, 1]$.
2. $\forall k = 1..K,\ 0 < \sum_{i=1}^{N} c_{ik} < N$,
3. $\forall \boldsymbol{x}_i \in \Omega,\ \sum_{k=1}^{K} c_{ik}$.

The second condition reflects the fact that no class should not be empty and the third expresses the concept of total membership.

Indexed Hierarchy

Ω is a finite set. H a set of non empty subsets of Ω is a hiearchy iff

- $\Omega \in H$
- $\forall \boldsymbol{x} \in \Omega, \{\boldsymbol{x}\} \in H$
- $\forall h,\ h' \in H,\ h \cap h' = \emptyset \textit{ or } h' \subset h \textit{ or } h \subset h'$

Index

The index of a hierarchy is a function i from H to $\mathbb{R}^+$ having the following properties

- $h \subset h' \Rightarrow i(h) < i(h')$
- $\forall \boldsymbol{x} \in \Omega,\ i(\{\boldsymbol{x}\}) = 0$

(h, i) is then an indexed hierarchy

Partition and Hierarchy

- each level of a indexed hierarchy is a partition
- $\{\Omega, P1, P2, \ldots, P_K, x_1, .., x_n\}$ is a hierarchy

Clustering ideal

Ideal

2 objects of a class should be `closer` than 2 objects of 2 different classes
$\Rightarrow$ most of the time impossible to achieve

- numerical approach: definition of a criterion (with or without a unique extremum)

The Bell numbers

But the number of partitions possible, even for a problem of reasonable size, is huge. Indeed if we consider a set of N objects to partition into K classes, the number of possible partitions is:

$$NP(N,K) = \frac{1}{K!}\sum_{k=0}^{K}(-1)^{k-1} \cdot C_k^K \cdot k^N.$$

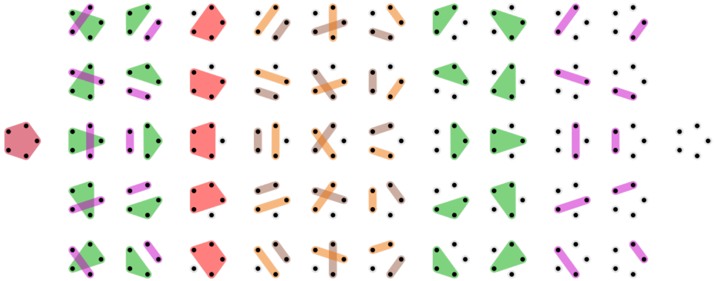


Figure 1: The 52 partitions of a set with 5 elements

The bell number

The bell number reccurence

Show that the number of partition of n objects verifies

$$B_{n+1} = \sum_{k=0}^{n} C_k^n B_k$$

Compute the number of partition of 5 objects.

The bell number

- fix an element x in a set with $n+1$ elements,
- sort the partitions according to the number k of elements outside the part containing x,
- For each value of k from 0 to n, it is necessary to choose k elements among the n different elements of x, then to give a partition.

Criteria and algorithms I

The concepts of partition and polythetic classification being specified, the following question emerges: how to find an optimal partition of a dataset, when the resemblance between two individuals is assessed by a measure of proximity?

The first thing to do is to formally clarify the meaning of the word optimal.

The solution generally adopted is to choose a digital measure of the quality of a partition.

This measure is sometimes called a criterion, functional, or still function of energy. The purpose of a classification procedure so is to find the partition or partitions that give the best value (the smallest or the largest) for a given criterion.

Rather than looking for the best score, the one that gives the optimum value of the criterion, more methods are used which converge towards “local” optima "of the criterion. thus found scores are often satisfying.

Intra-classe Inertia and partition I

Many criteria exist [@Gordon1980]. Some may be related, as we will see in the following, at the choice of a model for all the data. One of most used functions is the sum of intra-class variances:

$$\begin{aligned} I_W &= \sum_{k=1}^{K}\sum_{i=1}^{N} c_{ik}\left\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\right\|^2 \\ &= trace(\boldsymbol{S}_W) \end{aligned}$$

where the $\boldsymbol{\mu}_k$ are the prototypes (centers) of classes and the c_{ik} are the elements of a hard partition matrix. The problem is then a problem of optimization under constraints (related to c_{ik}) :

$$(\hat{\boldsymbol{c}}, \hat{\boldsymbol{\mu}}) = arg \min_{(\boldsymbol{c},\boldsymbol{\mu})} I_W((\boldsymbol{c}, \boldsymbol{\mu})) \tag{1}$$

where $\boldsymbol{\mu}$ represents all the centers of gravities.

K-means algorithm I

A common algorithm for solving this problem is the one **k-means**. Historically, this algorithm dates sixties. It has been proposed by several researchers in different areas at close dates [@ Edwards1965, @ Lloyd1957]. This algorithm based on considerations Geometric certainly owes its success to its simplicity and efficiency:

Algorithm

1. Initialization of centers: a common method is to initialize centers with coordinates of K randomly selected points.
2. Then the iterations have the following alternate form: a. given $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$, chose c_{ik} minimizing I_W, b. given $\boldsymbol{C} = \{c_{ik}\}$, minimise I_W with respect to $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$.

The first step

assigns each $\boldsymbol{x}_i$ to the nearest prototype,

the second step

recalculates the position of the prototypes considering that the prototype of the class i becomes its mean vector.

K-means algorithm II

Convervegence of kmeans algorithm

It is possible to show that each iteration decreases the criterion but no guarantee of convergence towards a global maximum does not exist in general.

Link with fuzzy clustering

If the k-means criterion is considered from the point of view of research of a fuzzy partition, that is, if the constraints on the c_{ik} are released and become $c_{ik} \in [0, 1]$ instead of $c_{ik} \in \{0, 1\}$, the optimal partition in the sense of the new criterion is the optimal one for the classical criterion [@Selim1984]. In other words, there is no interest in considering fuzzy scores when working with the criterion of k-means.

Kmeans algorithm's

Nuées dynamiques (dynamic swarm)

This form of alternate algorithm where a certain criterion is optimized, alternatively with respect to the class membership variables, then compared to the parameters defining these classes has been extensively exploited. Let's mention among others *the dynamic clouds* of Diday [@Diday1971] and the *fuzzy c-means* algorithm [@Bezdeck1974].
Note that Webster Fisher [@Fisher1958] (not to be confused with Ronald Fisher) had proposed an algorithm finding the optimal partition, within the intra-class variance, of a set of N data one-dimensional in $O(N \cdot K^2)$ operations using methods from dynamic programming.

In R, the function `kmeans {stats}` I

effectively implements the algorithm.

```r
data(iris)
kmeans.res <- iris %>%
              select(-Species,-Sepal.Length,-Sepal.Width) %>%
              kmeans(3,nstart = 10)

cluster<-as.factor(kmeans.res$cluster)
centers <-as.tibble(kmeans.res$centers)
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generate
```

```r
ggplot(iris, aes(x=Petal.Length, y=Petal.Width, color=cluster)) +
  geom_point() +
  geom_point(data=centers, color='coral',size=4,pch=21)+
  geom_point(data=centers, color='coral',size=50,alpha=0.2)
```

In R, the function `kmeans {stats}` II

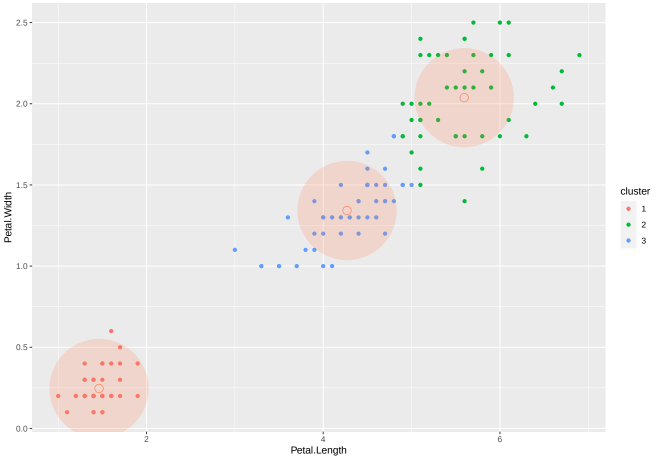


Figure 2: Automatic classification into three classes of iris by the k-means algorithm

Voronoi tiling I

By definition, `kmeans` partition the space by a Voronoi tiling defined by the centers.

```
library(deldir)
```

```
## deldir 0.2-10      Nickname: "Morpheus and Euripides"

##
##      Note 1: As of version 0.2-1, error handling in this
##      package was amended to conform to the usual R protocol.
##      The deldir() function now actually throws an error
##      when one occurs, rather than displaying an error number
##      and returning a NULL.
##
##      Note 2:  As of version 0.1-29 the arguments "col"
##      and "lty" of plot.deldir() had their names changed to
##      "cmpnt_col" and "cmpnt_lty" respectively basically
##      to allow "col" and and "lty" to be passed as "..."
##      arguments.
##
##      Note 3: As of version 0.1-29 the "plotit" argument
##      of deldir() was changed to (simply) "plot".
```

Voronoi tiling II

```
##
##      See the help for deldir() and plot.deldir().
```

```
#This creates the voronoi line segments
voronoi <- deldir(centers$Petal.Length, centers$Petal.Width)
 #Now we can make a plot
ggplot(data=iris, aes(x=Petal.Length, y=Petal.Width, color=cluster)) +
  geom_point()+
  #Plot the voronoi lines
  geom_segment(
    aes(x = x1, y = y1, xend = x2, yend = y2),
    size = 2,
    data = voronoi$dirsgs,
    linetype = 1,
    color= "coral") +
  #Plot the points
  geom_point(data=centers,
    fill='coral',
    pch=21,
    size = 4,
    color="#333333")
```

Voronoi tiling III

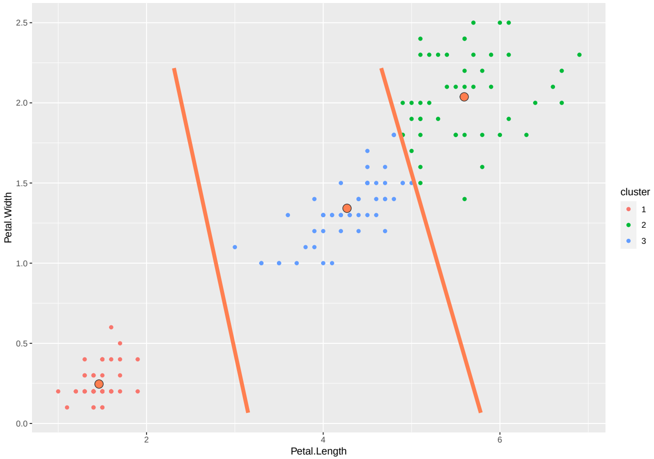


Figure 3: Classification automatique en trois classes des iris par l'algorithme des centres mobiles et pavage de Vorono<U+00EF>

Section 2

Classical algorithms

Hierachical Clustering

There are two main approaches

- Agglomerative
- Divisive

Hierarchical Agglomerative Clusering

The principle of Hierarchical Agglomerative Clusering is fairly easy

1. **Initialisation** each Ω element constitutes a class. A dissimilarity D is computed between all classes.
2. **While** number of cluster > 1

a. group the two closest classes in the sense of the dissimilarity D,
b. calculation of “distances’ ’ between the new class and the others.

Cluster dissimilarity

The dissimilarity D between two parts h and h' of Ω, can be defined in many ways from a measure of dissimilarity d on Ω.

- single link (critère du lien minimum):

$$D(h,h') = \min\left[d(\boldsymbol{x},\boldsymbol{y})/\boldsymbol{x} \in h \textit{ and } \boldsymbol{y} \in h'\right],$$

- complete link (critère du lien maximum)

$$D(h,h') = \max\left[d(\boldsymbol{x},\boldsymbol{y})/\boldsymbol{x} \in h \textit{ and } \boldsymbol{y} \in h'\right],$$

* group average

$$D(h,h') = \frac{\sum_{i=1}^{n_h} \sum_{j=1}^{n_{h'}} d(\boldsymbol{x}_i,\boldsymbol{x}_j)}{n_h \cdot n_{h'}},$$

- Ward criterion

$$D(h,h') = \frac{n_h \cdot n_{h'}}{n_h + n_{h'}} \|\boldsymbol{m}_h - \boldsymbol{m}_{h'}\|^2.$$

Intra class criterion and Ward method I

When we have a partition in K classes, the criterion of intra-class inertia measures its homogeneity:

$$\begin{aligned} I_W &= trace(\boldsymbol{S}_W), \\ &= \sum_{k=1}^{K} \sum_{i=1}^{n_k} (\boldsymbol{x}_{ik} - \boldsymbol{m}_k)^t (\boldsymbol{x}_{ik} - \boldsymbol{m}_k). \end{aligned}$$

Let us consider two partitions

- $P = (P_1, \cdots, P_K)$,
- and P', the partition obtained by merging the classes $\mathcal{C}_k$ et $\mathcal{C}_\ell$.

We can show that the difference between the inertia of the two partitions is equal to Ward's aggregation criteria:

$$I_{W'} - I_W = \frac{n_k \cdot n_\ell}{n_k + n_\ell} \|\boldsymbol{m}_k - \boldsymbol{m}_\ell\|^2.$$

Thus, each stage of Ward's algorithm chooses a new partition that limits the increase of intra-class inertia. Note that this property does not guarantee overall optimization of the criteria.

Recursive implementation of Lance et Williams}

Computing the distances can be faster using Lance and Williams formula:

Computing distance D between classes A and $B \cup C$ from distances D between A and B, A and C :

$$D_{\text{min}} : \qquad D(A, B \cup C) = \min\{D(A, B), D(A, C)\};$$

$$D_{\text{max}} : \qquad D(A, B \cup C) = \max\{D(A, B), D(A, C)\};$$

$$D\text{moy} : \qquad D(A, B \cup C) = \frac{card(B)D(A, B) + card(C).D(A, C)}{card(B) + card(C)}.$$

$$D(A, B \cup C) = \frac{(\#A + \#B) \times D(A, B) + (\#A + \#C) \times D(A, C) - \#A \times D(B, C)}{\#A + \#B + \#C}.$$

Ultrametric distance

An ultrametric distance δ checks all the properties that define a classical distance and satisfies in addition the inequality

$$\delta(\boldsymbol{x}, \boldsymbol{z}) \leq \max\left(\delta(\boldsymbol{x}, \boldsymbol{z}), \delta(\boldsymbol{z}, \boldsymbol{y})\right),$$

stronger than the triangular inequality.

- it is possible to interpret the minimum number of nestings required for two form vectors to belong to one class, as a dissimilarity.
- this dissimilarity is an ultrametric distance.
- it is possible to understand the problem of hierarchical clustering as the search for an ultrametric δ close to d, the dissimilarity used on Ω

Divisive hierarchical clustering

- the divisive clustering approach is much less popular
- In theory, the first step of a downward method must compare $2^{N-1} - 1$ possible partitions from N vectors, in two classes.
- To avoid those impossible calculations, one solution is to apply a method of partitioning to get both classes. Repeating this process recursively on each class obtained allows to have fast algorithms

Section 3

Chosing the number of cluster

Chosing the number of cluster

No definitive answer to this question.

The “elbow method”

The Elbow method looks at the total WSS as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn’t improve much better the total WSS.

The optimal number of clusters can be defined as follow:

Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters. For each k, calculate the total within-cluster sum of square (wss). Plot the curve of wss according to the number of clusters k. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters. Note that, the elbow method is sometimes ambiguous.

Silhouette

An alternative is the average silhouette method (Kaufman and Rousseeuw 1990) which can be also used with any clustering approach.

Average silhouette method

Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k (Kaufman and Rousseeuw 1990).
Silhouette score:

- mean distance from a point to its group: $a(i) = \frac{1}{|I_k|-1} \sum_{j \in I_k, j \neq i} d(x^i, x^j)$
- mean distance from a point to the closest group $b(i) = \min_{k' \neq k} \frac{1}{|I_{k'}|} \sum_{i' \in I_{k'}} d(x^i, x^{i'})$
- $s_{sil}(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$

$$S_{sil} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|I_k|} \sum_{i \in I_k} s_{sil}(i)$$

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the average silhouette of observations (avg.sil).
3. Plot the curve of avg.sil according to the number of clusters k. The location of the maximum is considered as the appropriate number of clusters.

Gap statistic method

R. Tibshirani, G. Walther, and T. Hastie (Standford University, 2001).

1. Cluster the observed data, varying the number of clusters from $k = 1, ..., k_{max}$, and compute the corresponding total within intra-cluster variation W_k.
2. Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters $k = 1, ..., k_{max}$, and compute the corresponding total within intra-cluster variation W_{kb}.
3. Compute the estimated gap statistic as the deviation of the observed W_k value from its expected value W_{kb} under the null hypothesis: $Gap(k) = \frac{1}{B}\sum_{b=1}^{B} log(W^*_{kb}) - log(W_k)$.
4. Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at $k+1$: $Gap(k) \geq Gap(k+1) - s_{k+1}$.

Section 4

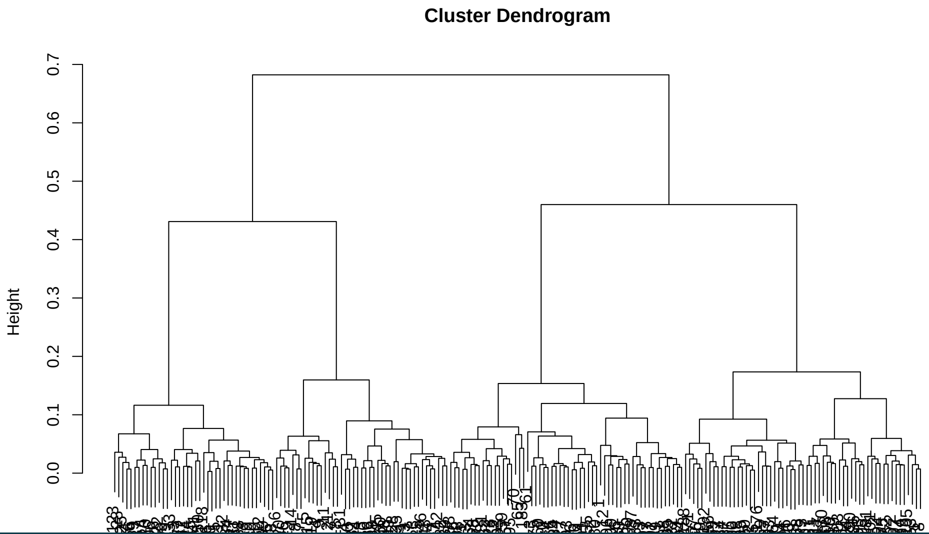
Examples

Normalized crabs

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
```

Hierarchical agglomerative clustering with R base

```r
dist_mat<-dist(crabsquant2)
hclust_ward.D2 <- hclust(dist_mat, method = 'ward.D2')
plot(hclust_ward.D2)
```



Cutting the tree

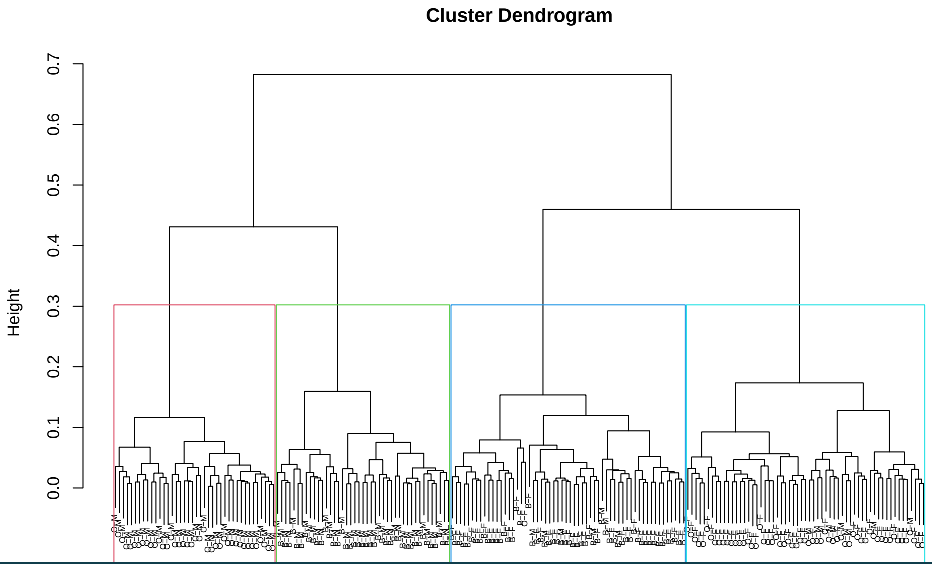
to get a partition

```
cut_ward.D2 <- cutree(hclust_ward.D2,k = 4)
table(true.classes,cut_ward.D2)
```

```
##              cut_ward.D2
## true.classes  1  2  3  4
##          B-F 50  0  0  0
##          B-M  7 43  0  0
##          O-F  1  0 49  0
##          O-M  0  0 10 40
```

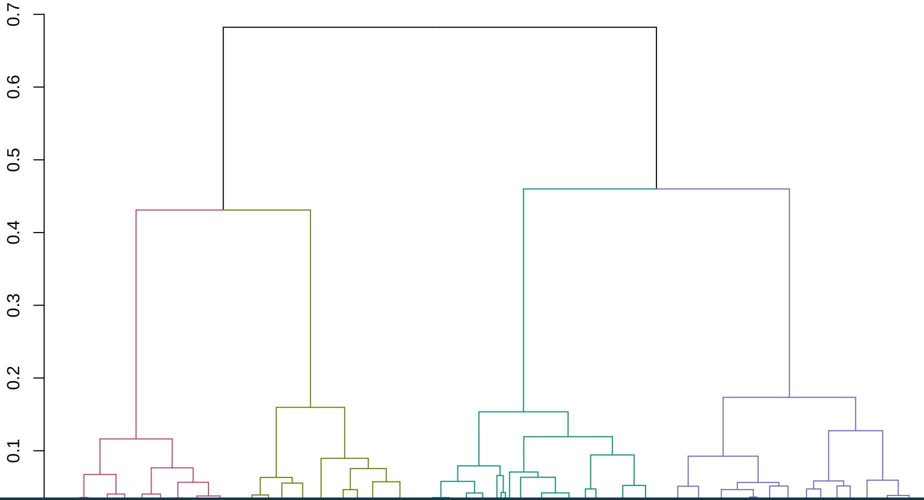
Display partition in dendrogram

```r
plot(hclust_ward.D2,labels = true.classes,cex=.5)
rect.hclust(hclust_ward.D2 , k = 4, border = 2:6)
```



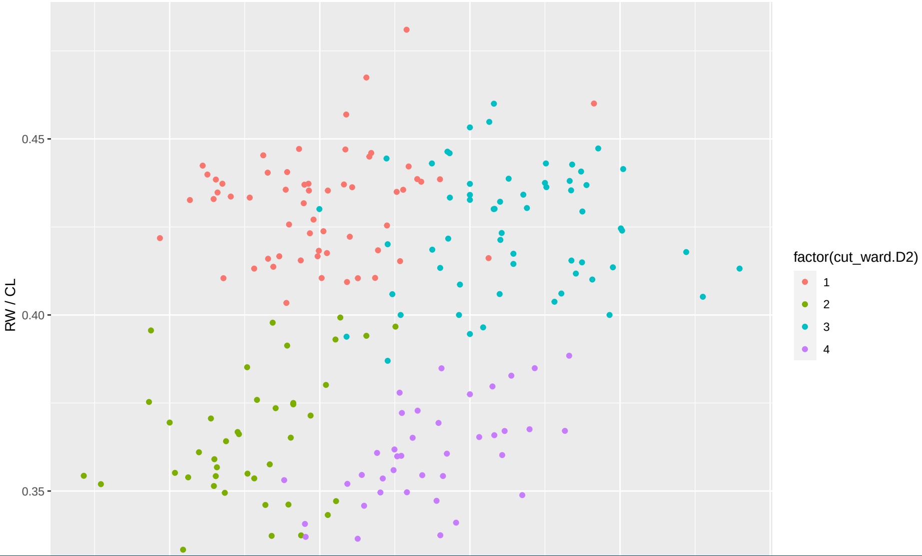
Using package “dendextend

```r
suppressPackageStartupMessages(library(dendextend))
ward.D2_dend_obj <- as.dendrogram(hclust_ward.D2)
ward.D2_col_dend <- color_branches(ward.D2_dend_obj, k = 4)
plot(ward.D2_col_dend)
```



Biplot of the hierachical clutering for 4 clusters

```r
suppressPackageStartupMessages(library(ggplot2))
ggplot(crabsquant2, aes(x=`FL / CL`, y = `RW / CL`, color = factor(cut_ward.D
```



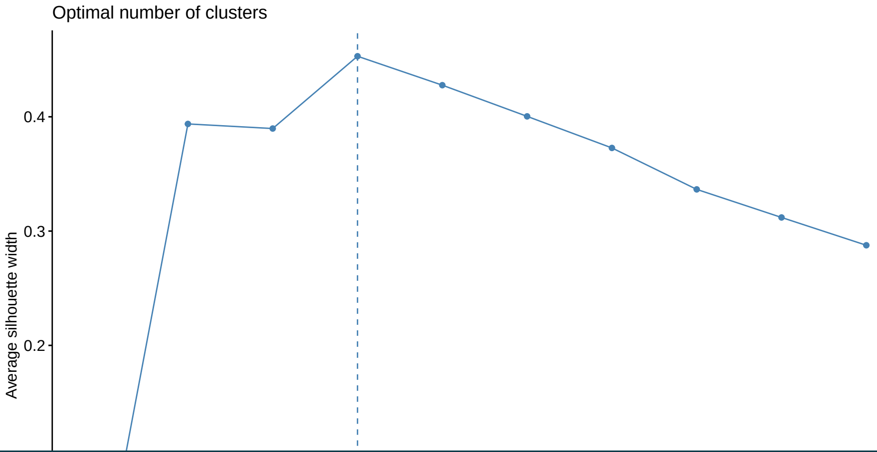
Dertermining and Visualizing the Optimal Number of Clusters

Silhouette

```
library(factoextra)
```

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

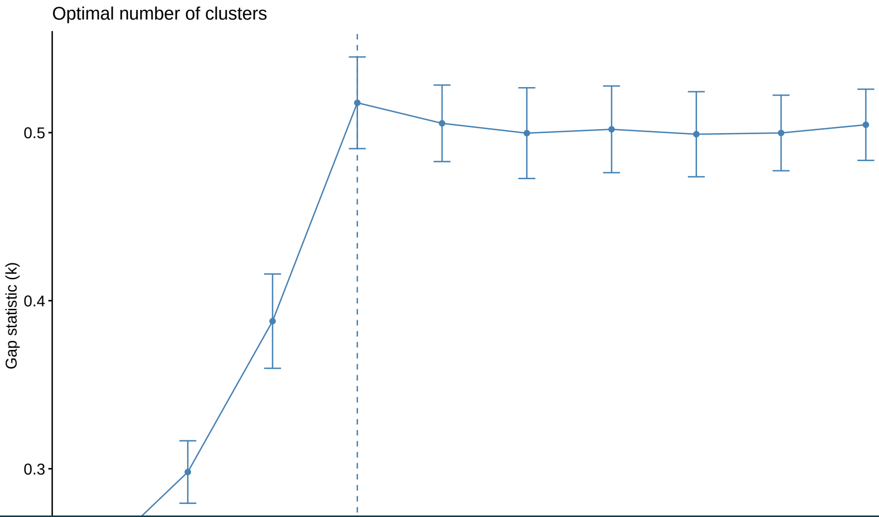
```
fviz_nbclust(as.matrix(dist_mat), FUN = hcut, method = "silhouette")
```



Dertermining and Visualizing the Optimal Number of Clusters

Gap Statistics

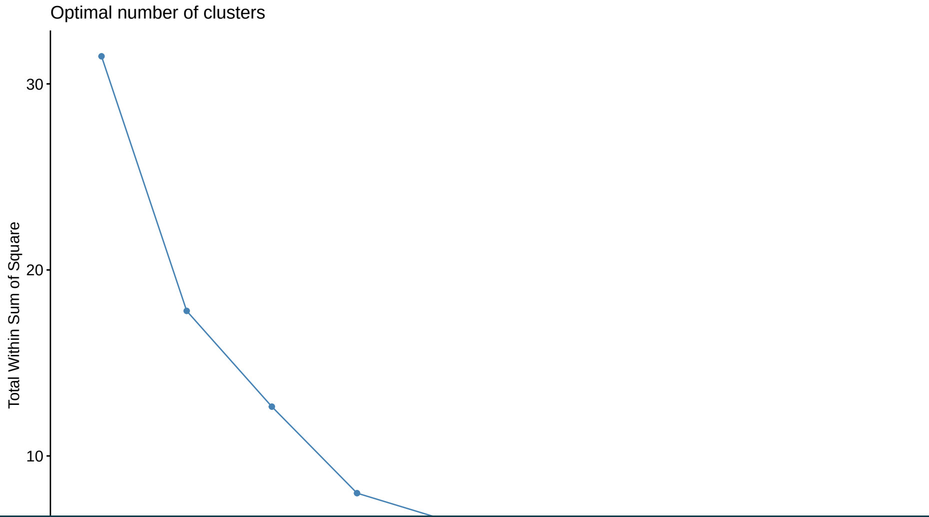
```r
gap_stat <- cluster::clusGap(crabsquant2, FUN = hcut, K.max = 10, B = 10)
fviz_gap_stat(gap_stat)
```



Dertermining and Visualizing the Optimal Number of Clusters

WSS

```
library(factoextra)
fviz_nbclust(as.matrix(dist_mat), FUN = hcut, method = "wss")
```



Hierarchical agglomerative clustering with R package cluster

agnes

Agglomerative Nesting Description: Computes agglomerative hierarchical clustering of the dataset. Usage: agnes(x, diss = inherits(x, "dist"), metric = "euclidean", stand = FALSE, method = "average", keep.diss = n < 100, keep.data = !diss)

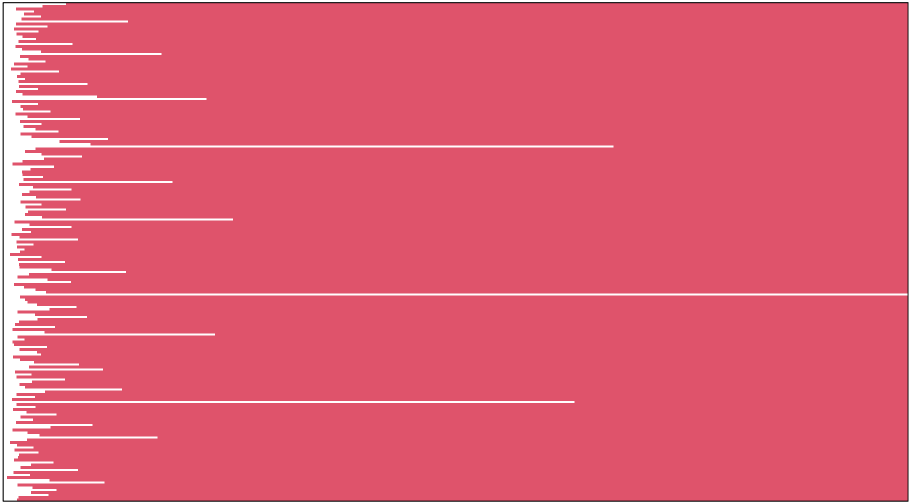
Agglomerative clustering with Ward

```r
library(cluster)
res<-agnes(crabsquant2,method="ward")
```

Banner plot

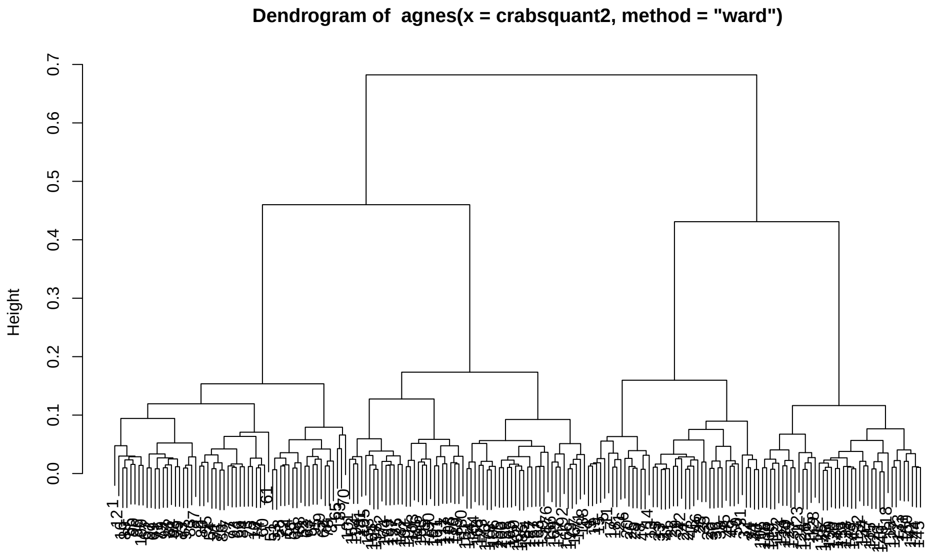
```
plot(res,which.plots=1)
```

Banner of agnes(x = crabsquant2, method = "ward")



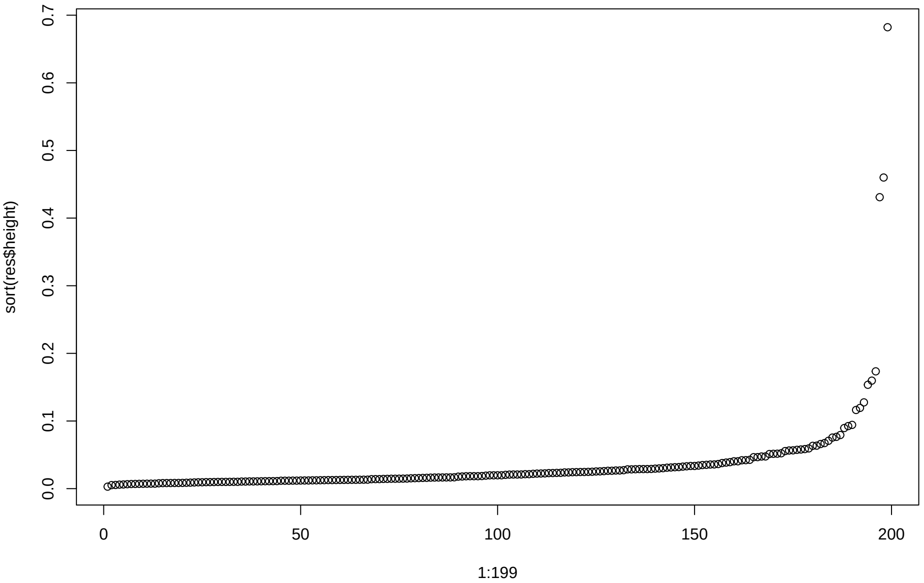
Dendogram

```
plot(res,which.plots=2)
```



Levels of fusion

```r
plot(1:199,sort(res$height))
```



Divisive Hiearchical Clustering with R

diana

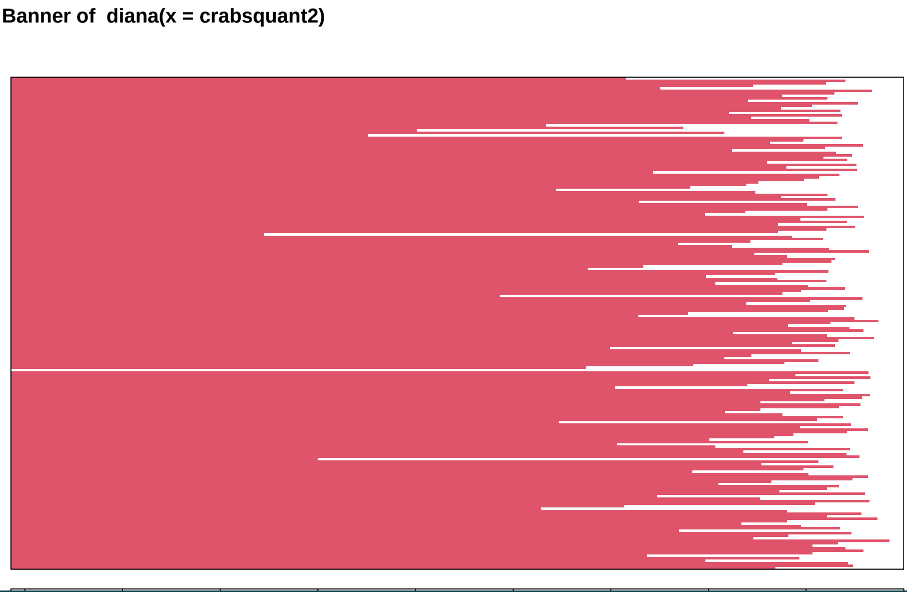
DIvisive ANAlysis Clustering Description: Computes a divisive hierarchical clustering of the dataset returning an object of class 'diana'. Usage:
diana(x, diss = inherits(x, "dist"), metric = "euclidean", stand = >FALSE, keep.diss = n < 100, keep.data = !diss)

Example of use

```
res<-diana(crabsquant2)
```

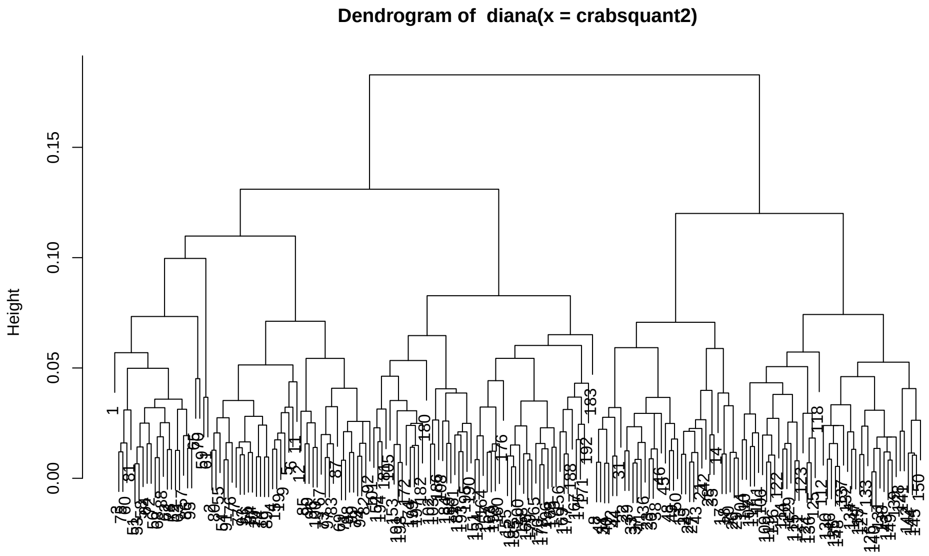
Banner plot

```r
plot(res,which.plots=1)
```



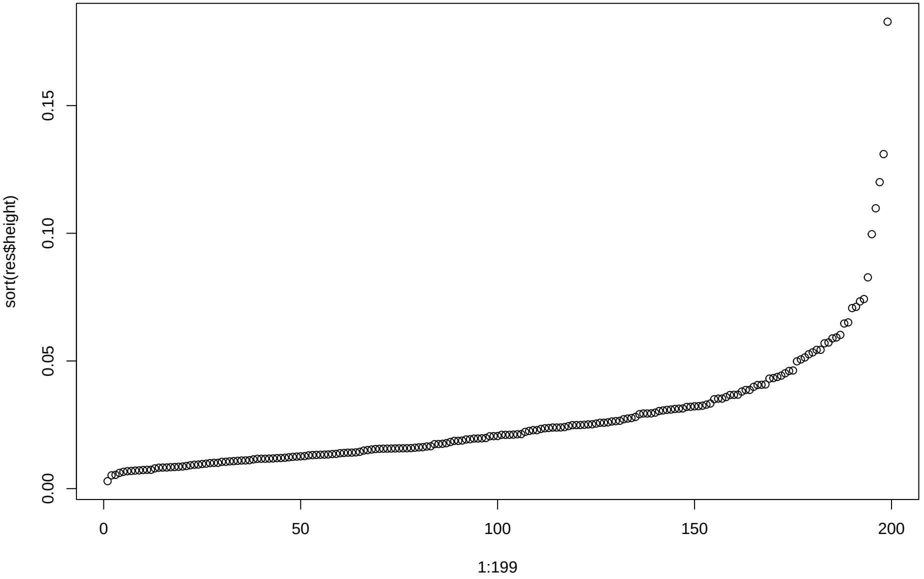
Dendogram

```
plot(res,which.plots=2)
```



Levels of division

```
plot(1:199,sort(res$height))
```



Section 5

Classification automatique (Exercices)

Partition and matrix

Consider the Iris data set. Write a R code wich produces the partition matrix.

Compute the gravity centers of the quantitative variables in the three classes using a matrix formula.

Solution

```r
data(iris)
X<-iris[,1:4]
library(nnet)
C<-class.ind(iris$Species)
t((t(X)%*%C))/diag(t(C)%*%C)
```

```
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```

The bell number

1. Show that the number of partition of n objects verifies

$$B_{n+1} = \sum_{k=0}^{n} C_k^n B_k$$

2. Compute manually the bell number for 1,2,3,4,5,6 objects.
3. Write a R program which computes the Bell number for n objects.

Solutions

The bell number recurrence relation

- fix an element x in a set with $n+1$ elements,
- sort the partitions according to the number k of elements outside the part containing x,
- For each value of k from 0 to n, it is necessary to choose k elements among the n different elements of x, then to give a partition.

Between-Within Variance relation

Consider n points from $\mathbb{R}^p$ with a partition into K classes of size $n_1, ..., n_k$. Let us note $\hat{\boldsymbol{\mu}}_k$ the gravity center of class k and $\hat{\boldsymbol{\mu}}$ the gravity center of the entire cloud of points.

Show that

$$\sum_k \sum_{i \in k} \|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k\|^2 + \sum_k n_k \|\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}}\|^2 = \sum_i \|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}\|^2$$

Solutions

Let us proceed in two steps

1. Show that
$$\sum_k \sum_{i \in k} \|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k\|^2 = \sum_i \|\boldsymbol{x}_i\|^2 - \sum_k n_k \|\hat{\boldsymbol{\mu}}_k\|^2$$
2. Show that
$$\sum_k n_k \|\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}}\|^2 = \sum_k n_k \|\hat{\boldsymbol{\mu}}_k\|^2 - n\|\hat{\boldsymbol{\mu}}\|^2$$
3. Conclude

Data set: crabs (module MASS)

Morphological Measurements on Leptograpsus Crabs
Description: The ‘crabs’ data frame has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species Leptograpsus variegatus collected at Fremantle, W. Australia.
Usage: data(crabs)

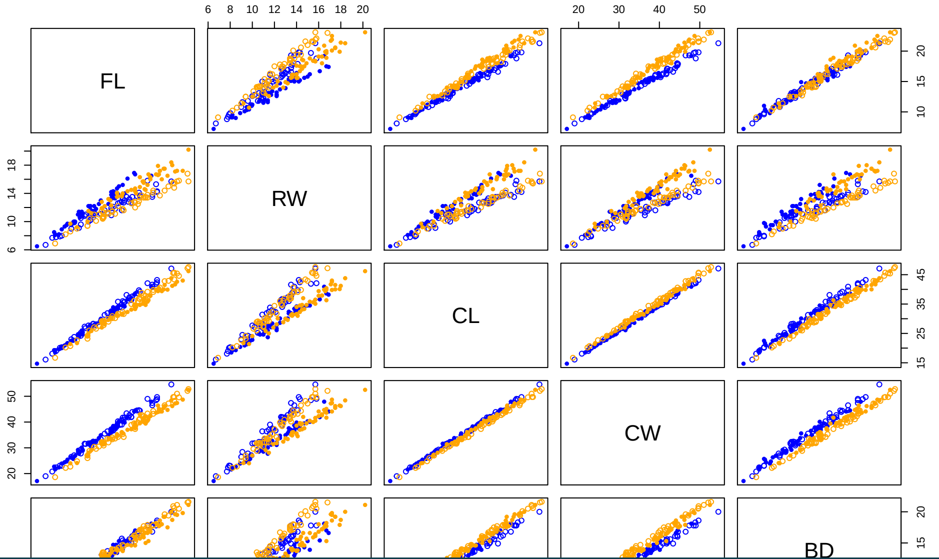
The crabs

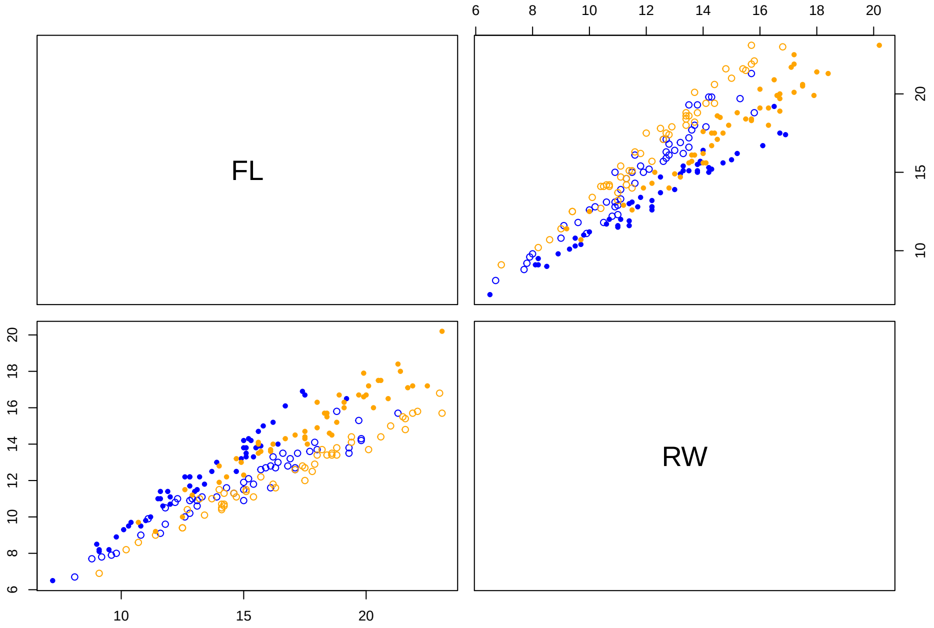
Figure 4: The crab

- Format: This data frame contains the following columns: sp' 'species' - ' "B" ' or ' "O" ' for blue or orange
    - sex' as it says
    - index' index 1:50 within each of the four groups
    - FL' frontal lobe size (mm)
    - RW' rear width (mm)
    - CL' carapace length (mm)
    - CW' carapace width (mm)
    - BD' body depth (mm)

Pairs graphs

```
library(MASS); data(crabs); crabsquant<-crabs[,4:8]
pairs(crabsquant,col=c("blue","orange")[crabs$sp],pch=c(20,21)[crabs$sex])
```




FL
RW
6
8
10
12
14
16
18
20
10
15
20
6
8
10
12
14
16
18
20
10
15
20

Clustering of raw data

```r
res<-kmeans(crabsquant,4)
str(res)
```

```
## List of 9
##  $ cluster     : Named int [1:200] 3 3 3 3 3 3 3 3 3 3 ...
##   ..- attr(*, "names")= chr [1:200] "1" "2" "3" "4" ...
##  $ centers     : num [1:4, 1:5] 14.2 17.2 10.6 20.8 11.8 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:5] "FL" "RW" "CL" "CW" ...
##  $ totss       : num 28500
##  $ withinss    : num [1:4] 836 776 798 631
##  $ tot.withinss: num 3041
##  $ betweenss   : num 25459
##  $ size        : int [1:4] 67 62 37 34
##  $ iter        : int 2
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Clustering of raw data (2)

```
table(Kmeans=res$cluster,TrueClasses)
```

```
##        TrueClasses
## Kmeans  1  2  3  4
##      1 18 16 14 19
##      2 14 17 17 14
##      3 17 10  4  6
##      4  1  7 15 11
```

Why such a bad result ?

Size effect (latent factor)

- The size information is present in all variables
- This has a masking effect

Solution

Divide all variables by one of them allows to have features which are relative

Variability of the results I

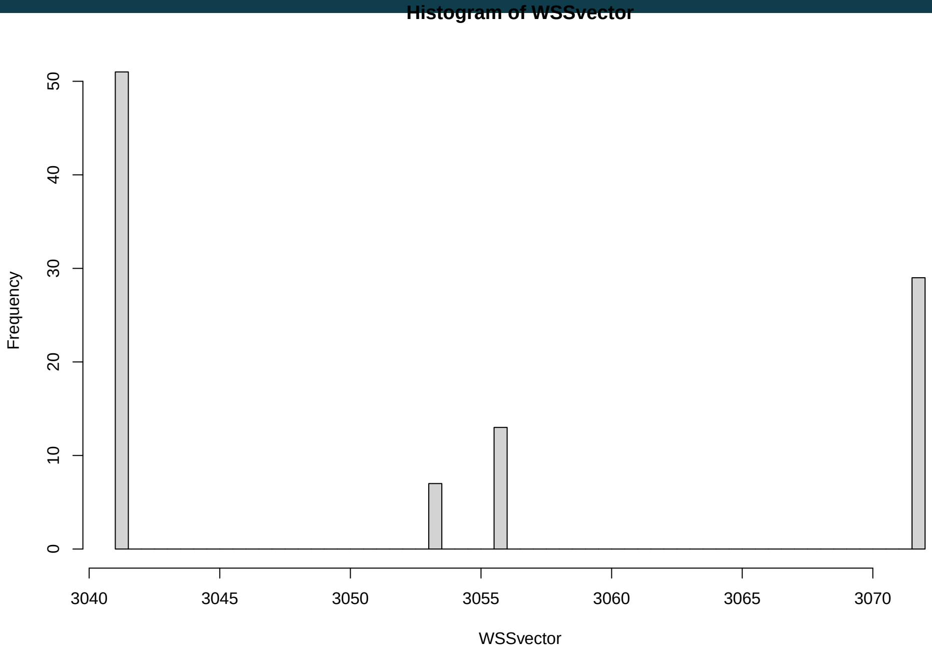
```
WSS<-function(partition)
{# Calcul l'intertie intra-classe d'une partition obtenue par k-means
 sum(partition$withinss)
}
WSSvector<-rep(0,100)
for (i in 1:100)
   {
       res<-kmeans(crabsquant,4)
       WSSvector[i]<- WSS(res)

   }
summary(WSSvector)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3041    3041    3041    3053    3072    3072
```

Variability of the results II

```r
hist(WSSvector,breaks=50)
```

Variability of the results III

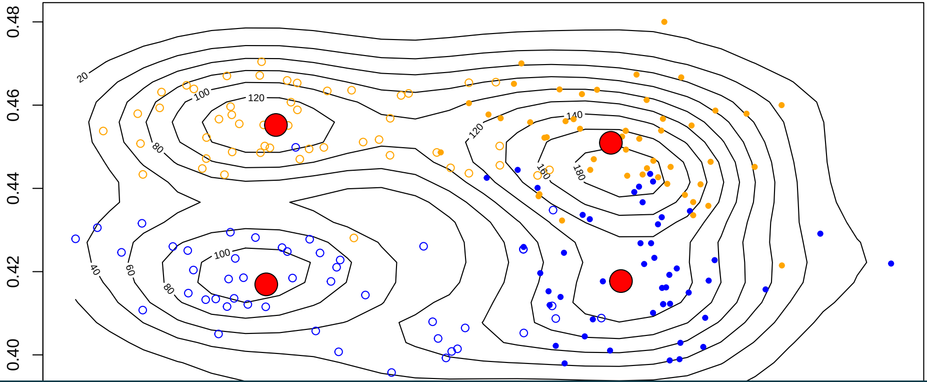


Getting rid of the size effect I

```
library(MASS); data(crabs)
n=dim(crabs)[1] # nb d'individus
crabsquant<-crabs[,4:8]
n=dim(crabs)[1] # nb d'individus
# gardons les variables quantitatives
crabsquant<-crabs[,4:8]
# Faisons une simple petite transformation pour enlever l'effet taille
crabsquant2<-(crabsquant/crabsquant[,3])[,-3]
# mettons les noms ? jour
j=0
for(i in c(1,2,4,5))
{
 j=j+1
 names(crabsquant2)[j]<-c(paste(names(crabsquant)[i],"/",names(crabsquant[
}
```

Clustering of the transformed dataset (1)

```
res<-kmeans(crabsquant2,4)
# repr?sentons les donn?es dans le plan 2, 4 avec densit? et centres
z<-kde2d(crabsquant2[,2],crabsquant2[,4])
contour(z)
# placons les points
points(crabsquant2[,c(2,4)],col=c("blue","orange")[crabs$sp],pch=c(20,21)[cr
# placons les centres des classes
points(res$center[,c(2,4)],cex=3,pch=21,bg="red")
```



Clustering of the transformed dataset (2)

```r
table(Kmeans=res$cluster,TrueClasses)
```

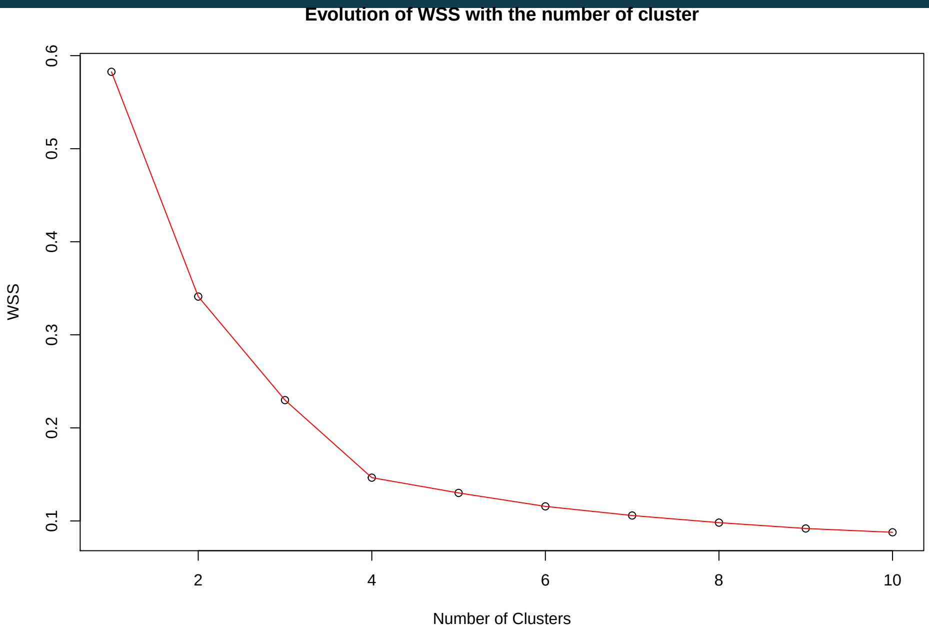
```
##       TrueClasses
## Kmeans  1  2  3  4
##      1  0  0 50  6
##      2  0 40  0  0
##      3 50 10  0  0
##      4  0  0  0 44
```

How many clusters ? I

```r
WSSkcluster<-rep(0,10)
for (k in 1:10)
{
    WSSmax<-Inf

    for (i in 1:10)
    {
       res<-kmeans(crabsquant2,k)
       if (WSS(res)<WSSmax)
        {partition<-res
         WSSmax<-WSS(res)}
    }
    WSSkcluster[k]<-WSS(partition)
}
```

```r
plot(WSSkcluster,xlab="Number of Clusters",ylab="WSS",main="Evolution of WSS
lines(WSSkcluster,col="red")
```

How many clusters ? II



Enterotypes of the human gut microbiome

(http://enterotype.embl.de/enterotypes.html)

The goal of the exercice is to understand the statistical analysis of a Nature paper published the 12 may 2011, entitled: Enterotypes of the human gut microbiome

The main conclusions come from the analysis of a metagenomic dataset Les conclusions principales sont déduites de l'analyse d' metagenomic dataset.

1. Charger les données d'abondance de genre relatives aux 33 échantillons. Chaque ligne correspond à un genre de bactérie et la première ligne correspond aux reads qui n'ont été affectés à aucun genre. Dans la suite, nous considérerons le tableau sans cette première ligne.
2. Considérons le tableau transposé où les variables sont les genres et les individus les échantillons. Utiliser l'algorithme des k-means pour trouver une partition en 3 classes.

Représenter les 3 populations par un boxplot sur la variable la plus discriminante (vous pourrez réaliser une anova sur les 248 variables) et choisir la variable de plus petite p-valeur pour ces tests.

3. Le critère ICL (Integrated Classification Likelihood) permet entre autre de comparer plusieurs partitions dans le cadre des modèles de mélange gaussiens. Il est fondé sur le principe de parcimonie: tenter de trouver le meilleur ajustement aux données avec le modèle le plus simple possible. Ainsi, il s'exprime comme un compromis entre l'ajustement du modèle aux données et le nombre de paramètres libres du modèle:

Aide à l'interprétation

Plusieurs études bio-statistiques ont produit des résultats sur le microbiote intestinal. Ci-dessous un article du New York Times intitulé "Bacterial Ecosystems Divide People Into 3 Groups, Scientists Say"" de CARL ZIMMER publié le 20 avril 2011:

> *In the early 1900s, scientists discovered that each person belonged to one of four blood types. Now they have discovered a new way to classify humanity: by bacteria. Each human being is host to thousands of different species of microbes. Yet a group of scientists now report just three distinct ecosystems in the guts of people they have studied.*
> *"It's an important advance," said Rob Knight, a biologist at the University of Colorado, who was not involved in the research. "It's the first indication that human gut ecosystems may fall into distinct types."*
> *The researchers, led by Peer Bork of the European Molecular Biology Laboratory in Heidelberg, Germany, found no link between what they called enterotypes and the ethnic background of the European, American and Japanese subjects they studied.*
> *Nor could they find a connection to sex, weight, health or age. They are now exploring other explanations. One possibility is that the guts, or intestines, of infants are randomly colonized by different pioneering species of microbes.*
> *The microbes alter the gut so that only certain species can follow them.*
> *Whatever the cause of the different enterotypes, they may end up having discrete effects on people's health. Gut microbes aid in food digestion and synthesize vitamins, using enzymes our own cells cannot make.*