UNIVERSITÉ
PAUL
SABATIER

TOULOUSE III

Data Mining

M2 SE

Xavier Gendre

xavier.gendre@math.univ-toulouse.fr

October 9, 2017

License

Contents

Chapter 1

Exploratory data analysis and dimension reduction

1.1 Data and definitions

1.1.1 Kind of data

From a general point of view, the aim of statistics is to describe phenomena based on observations of variables related to these phenomena. In the sequel, we call **variable** anything that we can observe. Let us consider some variable x, we call **data** any set of observations of x. For any positive integer n, we denote by $x_1, \dots, x_n$ the observations of x. We have to distinguish two kinds of data:

- if the data belong to some real vector space (*e.g.* if there exists some integer $k > 0$ such that $x_1, \dots, x_n \in \mathbb{R}^k$), we say that the data are **quantitative**. For instance, physical measurements (pressure, temperature, ...) lead to quantitative data.
- if the data belong to some unordered finite set, we say that the data are **qualitative**. For example, occupational categories (worker, student, unemployed, ...) lead to qualitative data.

Hereafter, we will often consider quantitative data because it is easier for computation. But we also will see that we are not restricted to quantitative data and that we can extend some technics in order to deal with qualitative data.

1.1.2 The real case

Let us begin by considering the simplest kind of quantitative data, real data, $x_1, \dots, x_n \in \mathbb{R}$ related to some variable x. We also introduce **weights** $w_1, \dots, w_n > 0$ associated to these data. For any $i \in \{1, \dots, n\}$, the weight w_i represents the importance of the observation x_i among the data. Usually, this importance is measured relatively to the total weight $w_1 + \cdots + w_n$ by the fraction

$$\tilde{w}_i = \frac{w_i}{w_1 + \cdots + w_n} \ .$$

A first quantity that we can define in relation to quantitative data is the **mean**,

$$\overline{x} = \frac{1}{w_1 + \cdots + w_n} \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} \tilde{w}_i x_i \ .$$

Note that it is equivalent to work with the weights $w_1, \dots, w_n$ and with $\tilde{w}_1, \dots, \tilde{w}_n$. The main difference is that the weights $\tilde{w}_1, \dots, \tilde{w}_n$ are said to be **normalized** because $\tilde{w}_1 + \cdots + \tilde{w}_n = 1$. In particular, the **uniform weights** $w_1 = \cdots = w_n = 1/n$ are normalized. For the sake of legibility, in the sequel, we will always assume that we deal with normalized weights (*i.e.* $w_1 + \cdots + w_n = 1$). When you handle the following formulas, be careful to this hinted assumption.

The mean $\overline{x}$ indicates around what the data are distributed. An other useful information is how far the data are from the mean. To measure this dispersal, a classical approach consists in considering the **variance**,

$$\sigma^2(x) = \sum_{i=1}^{n} w_i (x_i - \overline{x})^2 \ .$$

The variance is the mean of the squared distances between each observation and $\overline{x}$. This quantity is nonnegative and is close to zero only if the observations are all close to the mean. The **standard deviation** $\sigma(x)$ is the square root of the variance.

In order to get more precise information about how the data are distributed, we introduce the **quantiles**. Let $p \in [0, 1]$, the p-quantile $q_p(x)$ of the data is given by

$$q_p(x) = \inf \left\{ t \in \mathbb{R} \text{ such that } \sum_{i=1}^{n} w_i \mathbb{1}_{(-\infty, t]}(x_i) \geqslant p \right\} \ .$$

Note that the funtion $p \in [0, 1] \mapsto q_p(x)$ is nondecreasing (see Figure 1.1). The quantile $q_p(x)$ is the point that divides the data into two sets such that the set of observations smaller than $q_p(x)$ has a total weight greater or equal to p. Some particular quantiles have names:

- $q_{0.5}(x)$ is called the **median**,
- $q_{0.25}(x)$ and $q_{0.75}(x)$ are called the first and third **quartiles**, respectively.

The difference between the third and the first quartiles is called the **interquartile range**, $IQR = q_{0.75}(x) - q_{0.25}(x)$.

```
summary(cars$dist)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   26.00   36.00   42.98   56.00  120.00

p <- (0:100)/100
qp <- quantile(cars$dist, probs=p)
plot(p, qp, type="l", col="red")
```

A convenient way for graphically representing the distribution of the data is the **box plot**. Basically, five informations are summarized in a box plot : the smallest observation, the first quartile, the median, the third quartile and the largest observation. Moreover, one adds two whiskers to the box to indicate some additional informations. The lengths of these whiskers can vary from a representation to an other but a usual choice is to use the lowest observation still within $1.5 \times IQR$ of the first quartile and the highest observation still within $1.5 \times IQR$ of the third quartile (see Figure 1.2).

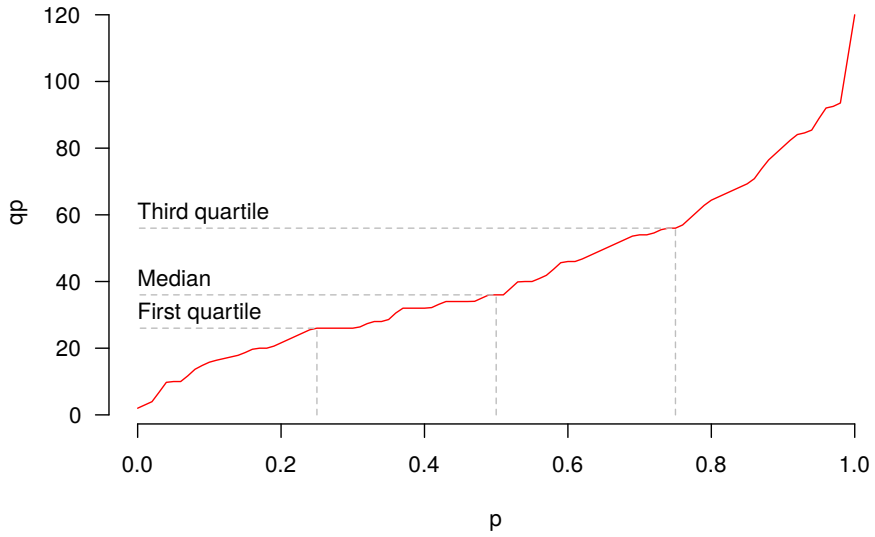

Figure 1.1: Quantile function of "Stopping Distances of Cars" with uniform weights

```
boxplot(cars$dist,horizontal=TRUE,col="orange")
```

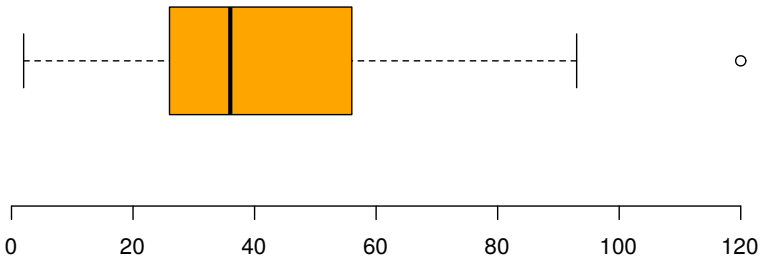

Figure 1.2: Example of box plot based on the "Stopping Distances of Cars" with uniform weights

An other graphical representation of the distribution of the data is the **histogram**. Let $b_0, \ldots, b_r$ be given such that all the data are in the open interval (b_0, b_r). For each $k \in$

$\{1, \dots, r\}$, the **frequency** f_k of the interval $[b_{k-1}, b_k]$ is

$$f_k = \sum_{i=1}^{n} w_i \mathbb{1}_{(b_{k-1}, b_k]}(x_i) \ .$$

Then, the histogram associated to the data is the piecewise constant function, defined for any $t \in \mathbb{R}$,

$$h_x(t) = \begin{cases} 0 & \text{if } t \leqslant b_0 \\ f_k/(b_k - b_{k-1}) & \text{if } t \in (b_{k-1}, b_k],\ k \in \{1, \dots, r\} \\ 0 & \text{if } t > b_r \end{cases} \ .$$

Note that all the frequencies are normalized by the length of the corresponding interval. In such a way, the area below the curve is equal to one. A similar approach consists in considering a symmetric nonnegative function K such that $\int_0^1 K(t)dt = 1$ and in defining, for any $t \in \mathbb{R}$,

$$h_{x,\lambda}(t) = \frac{1}{\lambda} \sum_{i=1}^{n} w_i K\left(\frac{t - x_i}{\lambda}\right)$$

for some $\lambda > 0$. It is easy to verify that the area below the curve of $h_{x,\lambda}$ is also equal to one. Figure 1.3 gives an example of an histogram h_x and a function $h_{x,\lambda}$ with uniform weights, $K(t) = \exp(-t^2/2)/\sqrt{2\pi}$ and $\lambda = 7.5$.

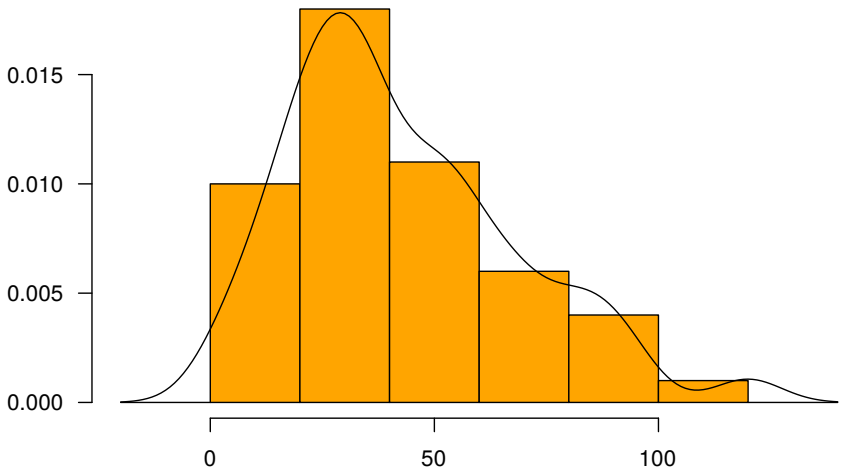

Figure 1.3: Histogram of "Stopping Distances of Cars" with uniform weights

1.1.3 Two real variables

In many situations, we do not observe only one quantitative variable. Let us assume, for the moment, that we observe a pair (x, y) of real variables. Thus, we have at our disposal

quantitative data given by n points $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^2$. As previously, for any $i \in \{1, \ldots, n\}$, the i-th observation (x_i, y_i) is weighted by some $w_i > 0$. A very classical way for representing such a data set is the **scatter plot** that is simply the plot of our n points (see Figure 1.4).

```
plot(cars$speed, cars$dist, pch=4)
```

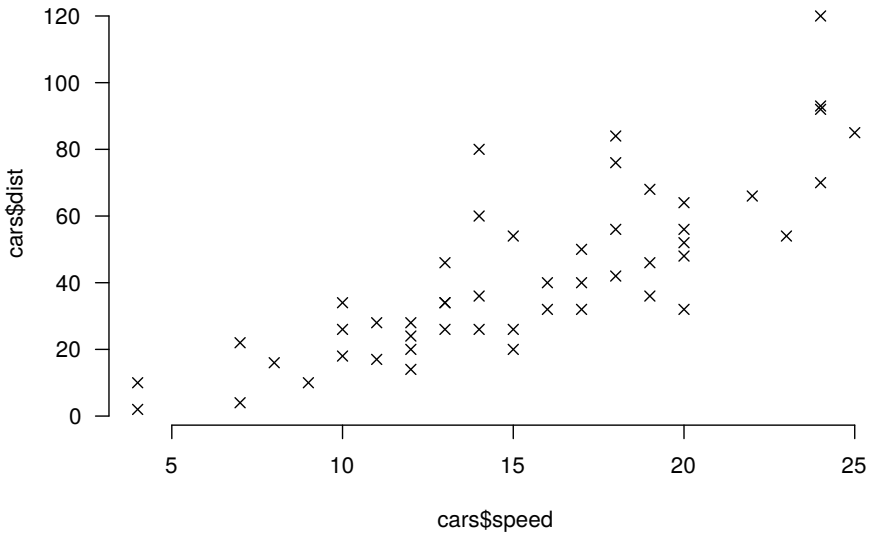

Figure 1.4: Scatter plot of "Speed and Stopping Distances of Cars"

The natural question that arises is the existence of a relation between the variables x and y. Many quantities have been introduced in order to deal with such a question. One of the most famous is the **covariance**,

$$\sigma(x, y) = \sum_{i=1}^{n} w_i (x_i - \overline{x})(y_i - \overline{y}) \ .$$

The covariance $\sigma(x, y)$ tends to be larger than 0 if the observations x_i and y_i are together greater or smaller than their means (*i.e.* "$x_i \geqslant \overline{x}$ and $y_i \geqslant \overline{y}$" or "$x_i \leqslant \overline{x}$ and $y_i \leqslant \overline{y}$"). Thus, positive covariance suggests that the variables x and y vary in the same direction. From the other side, $\sigma(x, y)$ tends to be smaller than 0 if the variables x and y vary in opposite direction. Note that, by definition, the covariance between a variable and itself is the variance,

$$\sigma(x, x) = \sigma^2(x) \ .$$

The underlying problem is that "larger than 0" is not well defined because the covariance is scale dependent. In order to avoid this problem, we can consider the covariance between normalized data, called **Pearson product-moment correlation coefficient**,

$$\rho(x, y) = \frac{\sigma(x, y)}{\sqrt{\sigma^2(x)\sigma^2(y)}} \ .$$

Cauchy–Schwarz inequality directly gives that $\rho(x,y) \in [-1,1]$. Moreover, we can prove that $|\rho(x,y)| = 1$ if and only if the points $(x_1,y_1),\dots,(x_n,y_n)$ are all distributed on a staight line. Thus, $\rho(x,y)$ allows to measure how the relation between x and y is far from an affine relation (such a relation will be suggested by $|\rho(x,y)|$ close to 1). Notice that there exist a lot of other correlation measurments (Spearmann, Kendall, ...). According to the studied problem, these other coefficient should be taken into account.

Clearly, in practice, data are never perfectly distributed along a straight line. Nevertheless, if $|\rho(x,y)|$ is close to 1, it can be interesting to know what kind of affine relation could rely the variables x and y. In other words, we are looking for two reals a and b such that the **least square criterion** $\gamma_{x,y}(a,b)$ is minimal,

$$\gamma_{x,y}(a,b) = \sum_{i=1}^{n} w_i \left(y_i - a x_i - b\right)^2 \ .$$

It is easy to see that the minimal value of $\gamma_{x,y}(a,b)$ is reached by

$$\hat{a} = \frac{\sigma(x,y)}{\sigma^2(x)} \quad \text{and} \quad \hat{b} = \overline{y} - \hat{a} \times \overline{x}$$

and so, the **linear regression line** is given by the equation $y = \hat{a}x + \hat{b}$ (see Figure 1.5).

```
cor(cars$speed, cars$dist)

## [1] 0.8068949

a <- cov(cars$speed,cars$dist)/var(cars$speed)
b <- mean(cars$dist)-a*mean(cars$speed)
abline(b, a, col="red")
```

1.1.4 Real vectors of observations

From a practical point of view, we often deal with much more than one or two variables. Thus, we now consider p quantitative variables $x^1,\dots,x^p$ that we observe n times. In other words, we have at our disposal the observations $x_1,\dots,x_n \in \mathbb{R}^p$ with, for any $i \in \{1,\dots,n\}$,

$$x_i = \begin{pmatrix} x_i^1 \\ \vdots \\ x_i^p \end{pmatrix} \in \mathbb{R}^p \ .$$

A natural way for giving sense to the mean of the x_i's is to define the **center of gravity**,

$$g(x) = \sum_{i=1}^{n} w_i x_i = \begin{pmatrix} \overline{x^1} \\ \vdots \\ \overline{x^p} \end{pmatrix} \in \mathbb{R}^p \ ,$$

i.e. the mean of the observed vectors is the vector of the observed means.

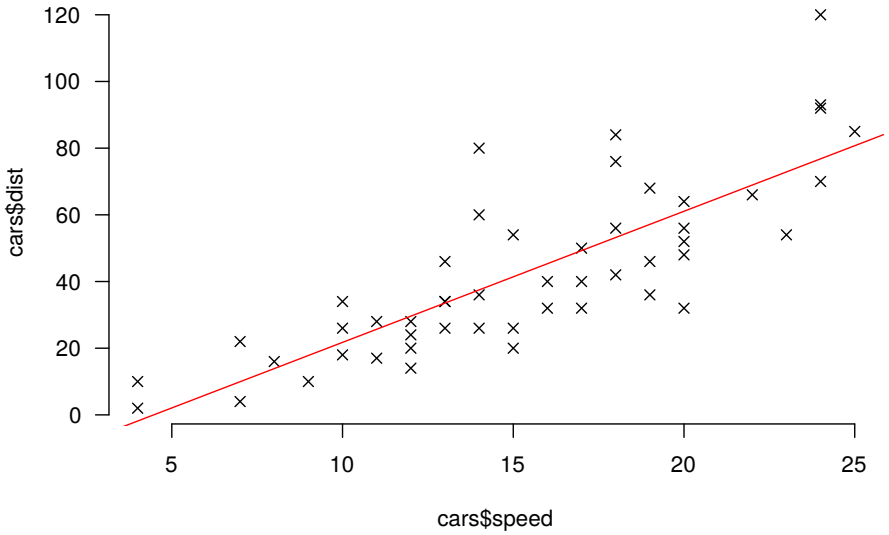

Figure 1.5: Linear regression line for "Stopping Distances" with respect to "Speed" and uniform weights

Dealing with vectorial data could implies some difficulties for the notations. For this reason, it is convenient to use matrix notations. The weights $w_1, \ldots, w_n >$ are given by the diagonal $n \times n$-**weight matrix**,

$$W = \begin{bmatrix} w_1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & w_n \end{bmatrix},$$

and the observations are given by the $n \times p$-**data matrix**,

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \ldots & x_1^p \\ x_2^1 & x_2^2 & \ldots & x_2^p \\ \vdots & \vdots & \ldots & \vdots \\ x_n^1 & x_n^2 & \ldots & x_n^p \end{bmatrix},$$

where each line is relative to an individual and each column is relative to a variable. We also often use the $n \times p$-**centered data matrix** $\bar{X}$ that is obtained from X by considering centered observations,

$$\bar{X} = \begin{bmatrix} x_1^1 - \overline{x^1} & x_1^2 - \overline{x^2} & \ldots & x_1^p - \overline{x^p} \\ x_2^1 - \overline{x^1} & x_2^2 - \overline{x^2} & \ldots & x_2^p - \overline{x^p} \\ \vdots & \vdots & \ldots & \vdots \\ x_n^1 - \overline{x^1} & x_n^2 - \overline{x^2} & \ldots & x_n^p - \overline{x^p} \end{bmatrix}.$$

By the aid of this matrix $\bar{X}$, we can define the $p \times p$-**covariance matrix** that contains all the covariances that we can compute between two observed variables,

$$\Sigma = {}^t\bar{X}W\bar{X} = \begin{bmatrix} \sigma^2(x^1) & \sigma(x^1,x^2) & \dots & \sigma(x^1,x^p) \\ \sigma(x^1,x^2) & \sigma^2(x^2) & \dots & \sigma(x^2,x^p) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(x^1,x^p) & \sigma(x^2,x^p) & \dots & \sigma^2(x^p) \end{bmatrix} .$$

Indeed, for any $i \in \{1,\dots,n\}$ and any $j \in \{1,\dots,p\}$, we have

$$\left({}^t\bar{X}W\bar{X}\right)_{ij} = \sum_{k=1}^{n}\sum_{\ell=1}^{n} {}^t\bar{X}_{ik}W_{k\ell}\bar{X}_{\ell j} = \sum_{k=1}^{n} w_k\bar{X}_{ki}\bar{X}_{kj} = \sum_{k=1}^{n} w_k(x_k^i - \overline{x^i})(x_k^j - \overline{x^j}) = \sigma(x^i,x^j) .$$

Note that, by definition, we know that Σ is symmetric. It is also easy to check that Σ is positive. Indeed, let $u \in \mathbb{R}^p$,

$${}^tu\Sigma u = {}^t(\bar{X}u)W\bar{X}u = \sum_{i=1}^{n} w_i(\bar{X}u)_i^2 \geqslant 0 .$$

To generalize the variance, a basic approach consists in considering the **standard inertia**,

$$I(x) = \sum_{j=1}^{p} \sigma^2(x^j) ,$$

i.e. to measure the dispersal of the observed vectors, we sum of the measurements of the dispersal according to each coordinate. To get a more general outlook, we can rewrite $I(x)$ in a particular way,

$$I(x) = \sum_{j=1}^{p}\sum_{i=1}^{n} w_i(x_i^j - \overline{x^j})^2 = \sum_{i=1}^{n} w_i \sum_{j=1}^{p}(x_i^j - \overline{x^j})^2 = \sum_{i=1}^{n} w_i\|x_i - g(x)\|^2_{\mathrm{Id}_p}$$

where we have set, for any positive $p \times p$-matrix M,

$$\forall u,v \in \mathbb{R}^p,\ \langle u,v\rangle_M = {}^tuMv \text{ and } \|u-v\|_M^2 = {}^t(u-v)M(u-v) .$$

Obviously, if M is a symmetric positive matrix, $\langle\cdot,\cdot\rangle_M$ is a scalar product in $\mathbb{R}^p$ and $\|\cdot\|_M$ is a norm in $\mathbb{R}^p$. Note that replacing M by $(M + {}^tM)/2$ does not change these definitions. Thus, we will always can consider that $\langle\cdot,\cdot\rangle_M$ and $\|\cdot\|_M$ are given by a symmetric matrix M in the sequel. Moreover, this leads to the general definition of **inertia**,

$$I_M(x) = \sum_{i=1}^{n} w_i \|x_i - g(x)\|_M^2 .$$

Particular choices of M lead to interesting situations. If $M = \mathrm{diag}(1/\sigma^2(x^1),\dots,1/\sigma^2(x^p))$, it amounts to deal with normalized data, *i.e.* $\tilde{x}_i^j = (x_i^j - \overline{x^j})/\sigma(x^j)$. If Σ is invertible and $M = \Sigma^{-1}$, $\|u-v\|_M$ is known as the **Mahalanobis distance** between u and v.

All these considerations bring us to consider two spaces for dealing with our data:

- the **space of variables** : isomorphic to $\mathbb{R}^n$, this is the space of the observations of one variable. It is equiped with a metric relative to the weight matrix W,

$$\forall u, v \in \mathbb{R}^n,\ \|u-v\|_W^2 = {}^t(u-v)W(u-v) = \sum_{i=1}^{n} w_i(u_i - v_i)^2 \ .$$

- the **space of individuals** : isomorphic to $\mathbb{R}^p$, this is the space of the observations related to one individual. It is equiped with a metric relative to some positive matrix M,

$$\forall u, v \in \mathbb{R}^p,\ \|u-v\|_M^2 = {}^t(u-v)M(u-v) \ .$$

We will often use this duality in the sequel.

1.2 Principal component analysis

1.2.1 Principle

In this section, we assume that we dispose of quantitative data $x_1, \dots, x_n \in \mathbb{R}^p$ relative to some p real variables $x^1, \dots, x^p$. The goal of PCA (Principal component analysis) is to provide a method for reducing the dimension:

- to produce some "optimal" graphical representation of the data in $\mathbb{R}^2$ or $\mathbb{R}^3$ with $p \geqslant 3$,
- to understand the correlation structure of the variables,
- to compress the data, *i.e.* to build q variables with $q < p$ to explain the data without losing too much information.

An important idea to keep in mind is the following one : we want to try to conserve the distances between the observations in $\mathbb{R}^p$ and their versions in the reduced space. These distances are measured according to a metric given by for some well chosen positive $p \times p$-matrix M. So, in other words, we want to conserve the inertia $I_M(x)$ of the data.

Let $E_d \subset \mathbb{R}^p$ be some linear space of dimension $d \leqslant p$ and $v^1, \dots, v^d \in \mathbb{R}^p$ be a M-orthonormal basis of E_d (*i.e.* orthonormal basis according to the scalar product $\langle \cdot, \cdot \rangle_M$). For any $i \in \{1, \dots, n\}$, we denote $\tilde{x}_i = x_i - g(x)$ and we know that the orthogonal projection of $\tilde{x}_i$ in E_d is given by

$$\pi_{E_d}(\tilde{x}_i) = \sum_{j=1}^{d} \langle \tilde{x}_i, v^j \rangle_M v^j = \sum_{j=1}^{d} {}^t v^j M \tilde{x}_i v_j \ .$$

Conserving the inertia means that we want to find a space E_d of fixed dimension d such that the inertia $I_M(x, E_d)$ of the projected observations is as close as possible to the inertia $I_M(x)$

of the data. Let us write the inertia of the projected observations,

$$\begin{aligned} I_M(x, E_d) &= \sum_{i=1}^{n} w_i \|\pi_{E_d}(\tilde{x}_i)\|_M^2 = \sum_{i=1}^{n} w_i \sum_{j=1}^{d} \langle \tilde{x}_i, v^j \rangle_M^2 \\ &= \sum_{i=1}^{n} w_i \sum_{j=1}^{d} \left({}^t\tilde{x}_i M v^j \right)^2 = \sum_{i=1}^{n} w_i \sum_{j=1}^{d} {}^t v^j \, {}^t M \tilde{x}_i \, {}^t\tilde{x}_i M v^j \\ &= \sum_{j=1}^{d} {}^t v^j \, {}^t M \left(\sum_{i=1}^{n} w_i \tilde{x}_i \, {}^t\tilde{x}_i \right) M v^j = \sum_{j=1}^{d} {}^t v^j \, {}^t M \Sigma M v^j \\ &= \sum_{j=1}^{d} {}^t\left(\Sigma M v^j \right) M v^j = \sum_{j=1}^{d} \langle \Sigma M v^j, v^j \rangle_M \ . \end{aligned}$$

Thus, to find the "best" space E_d, we just have to get M-orthonormal vectors $v^1, \ldots, v^d \in \mathbb{R}^p$ that maximize this last sum.

Of course, this optimization problem is connected to singular values decomposition. Note that, because M is symmetric, we know that the square matrix ΣM is self-adjoint for $\langle \cdot, \cdot \rangle_M$,

$$\forall u, v \in \mathbb{R}^p, \ \langle \Sigma M u, v \rangle_M = {}^t(\Sigma M u) \, M v = {}^t u \, {}^t M \, {}^t \Sigma M v = {}^t u M \Sigma M v = \langle u, \Sigma M v \rangle_M \ ,$$

and positive,

$$\forall u \in \mathbb{R}^p, \ \langle \Sigma M u, u \rangle_M = {}^t(M u) \Sigma M u \geqslant 0 \ .$$

Thus, we know that ΣM admits p nonnegative eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_p \geqslant 0$ and we denote by $v^1, \ldots, v^p \in \mathbb{R}^p$ the M-orthonormal basis of eigenvectors associated to the λ_i's. So, E_d is the space generated by $v^1, \ldots, v^d$ (called **principal vectors**) that are M-orthonormal eigenvectors associated to the d largest eigenvalues of ΣM,

$$I_M(x, E_d) = \sum_{j=1}^{d} \langle \Sigma M v^j, v^j \rangle_M = \sum_{j=1}^{d} \langle \lambda_j v^j, v^j \rangle_M = \sum_{j=1}^{d} \lambda_j \ .$$

Note that the spaces E_d are nested ($E_1 \subset E_2 \subset \cdots \subset E_p$) and, for $d = p$, we have $I_M(x, E_p) = I_M(x) = \lambda_1 + \cdots + \lambda_p$. We call **explained inertia** of E_d the fraction

$$\frac{\sum_{j=1}^{d} \lambda_j}{\sum_{j=1}^{p} \lambda_j} \ .$$

This quantity gives us a way for choosing a "good" dimension d for explaining the data. An explained inertia largest than 80% is commonly considered as good.

In practice, we do not directly compute the M-orthonormal vectors $v^1, \ldots, v^p$. Indeed, M-orthonormality is not an easy-to-handle notion and we prefer to come down to classic orthonormality by noting that, by invertibility of M,

$$\text{"}v \text{ is an eigenvalue of } \Sigma M\text{"} \Leftrightarrow \text{"}\tilde{v} = M^{1/2} v \text{ is an eigenvalue of } M^{1/2} \Sigma M^{1/2}\text{"} \ .$$

Moreover, we have

$$\langle \Sigma M v, v\rangle_M = {}^t v M \Sigma M v = {}^t \tilde{v} M^{1/2} \Sigma M^{1/2} \tilde{v} = \langle M^{1/2}\Sigma M^{1/2}\tilde{v}, \tilde{v}\rangle$$

where $\langle \cdot, \cdot \rangle$ is the usual scalar product. The matrix $M^{1/2}\Sigma M^{1/2}$ is positive symmetric, so we can find $\tilde{v}^1, \dots, \tilde{v}^p$ orthonormal for $\langle \cdot, \cdot \rangle$ that are eigenvectors of $M^{1/2}\Sigma M^{1/2}$ associated to eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_p \geqslant 0$. To get our M-orthonormal eigenvectors of ΣM, we set, for any $j \in \{1, \dots, p\}$, $v^j = M^{-1/2}\tilde{v}^j$ (easy to check that they are M-orthonormal eigenvectors).

1.2.2 Toy examples

We introduce two examples that we will handle in the sequel of this chapter.

Grades

We consider data that are grades of $n = 9$ students in $p = 4$ topics:

	Maths	Physics	French	English
Benny	6	6	5	5.5
Bobby	8	8	8	8
Brandy	6	7	11	9.5
Coby	14.5	14.5	15.5	15
Daisy	14	14	12	12.5
Emily	11	10	5.5	7
Judy	5.5	7	14	11.5
Marty	13	12.5	8.5	9.5
Sandy	9	9.5	12.5	12

To not favour anyone, we consider uniform weights $w_1 = \cdots = w_n = 1/n$ (*i.e.* $W = \frac{1}{n}\mathrm{Id}_n$). Thus, we have

$$X = \begin{bmatrix} 6 & 6 & 5 & 5.5 \\ 8 & 8 & 8 & 8 \\ 6 & 7 & 11 & 9.5 \\ 14.5 & 14.5 & 15.5 & 15 \\ 14 & 14 & 12 & 12.5 \\ 11 & 10 & 5.5 & 7 \\ 5.5 & 7 & 14 & 11.5 \\ 13 & 12.5 & 8.5 & 9.5 \\ 9 & 9.5 & 12.5 & 12 \end{bmatrix}$$

and the covariance matrix

$$\Sigma = \frac{1}{n}\,{}^t\bar{X}\bar{X} = \begin{bmatrix} 11.3888889 & 9.9166667 & 2.6574074 & 4.8240741 \\ 9.9166667 & 8.9444444 & 4.1203704 & 5.4814815 \\ 2.6574074 & 4.1203704 & 12.0617284 & 9.2932099 \\ 4.8240741 & 5.4814815 & 9.2932099 & 7.9135802 \end{bmatrix}.$$

Let us have a look to the distribution of these variables (see Figure 1.6). We see that the 4 variables are "normally" distributed. We decide to compare them in the classical way with $M = \mathrm{Id}_4$ (*i.e* $\langle \cdot, \cdot \rangle_M$ is the usual scalar product in $\mathbb{R}^p$).

```
boxplot(X,col="orange")
```

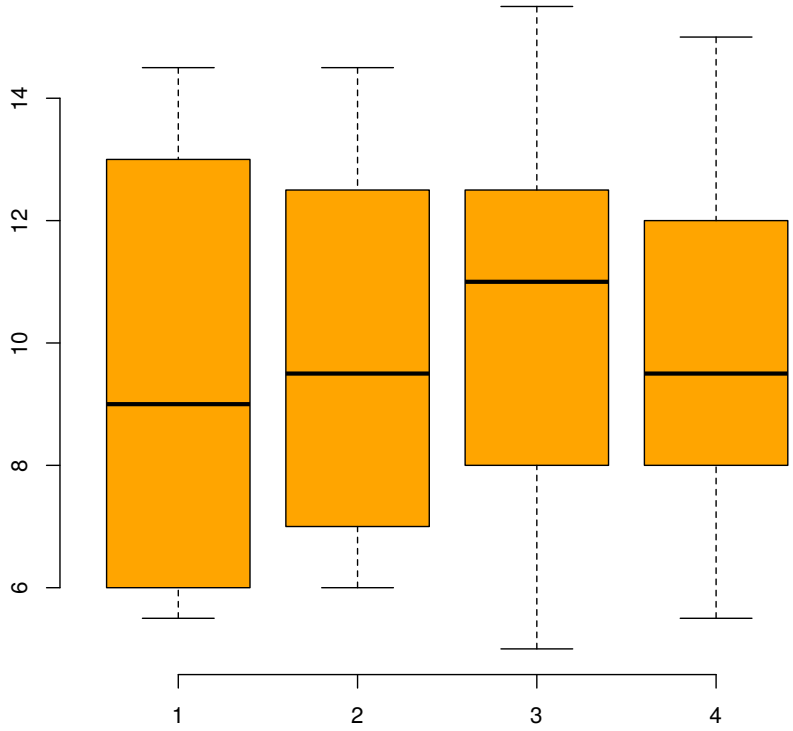

Figure 1.6: Box plot of the grades data

Then, we can get the ordered eigenvalues of $\Sigma M = \Sigma$ and consider that $d = 2$ is a good trade-off for representing the data (99.89% of explained inertia). A usefull tool for helping us to see what is a good value of d is to plot the ordered eigenvalues λ_k according to k (see Figure 1.7).

```
dg <- eigen(Sigma)
dg$values
```

```
## [1] 28.23487122 12.03054605  0.03263201  0.01059269
```

```
cumsum(dg$values)/sum(dg$values)
```

```
## [1] 0.7004669 0.9989277 0.9997372 1.0000000
```

```
plot(1:4, dg$values, type="b", col="orange")
```

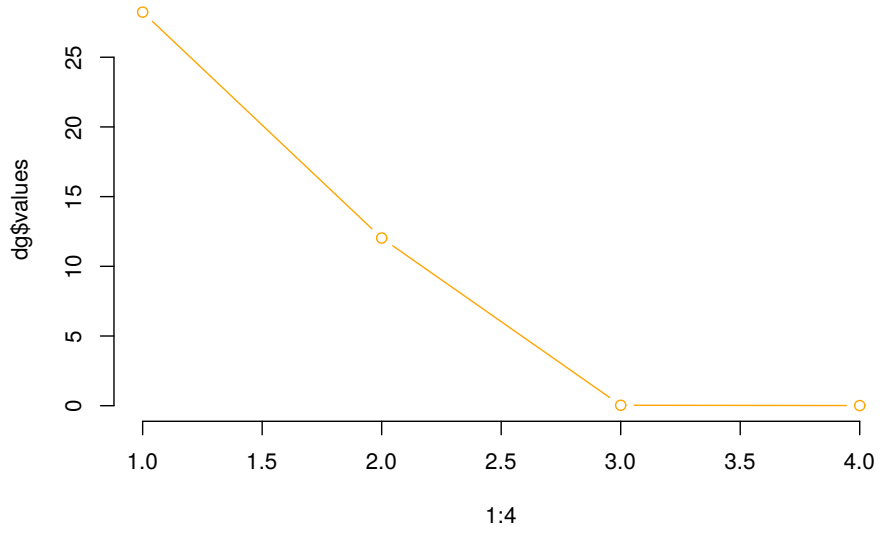

Figure 1.7: Ordered eigenvalues of the grades data

Skyrim bows

The data of the second example are the weight, value, damage and speed (so, $p = 4$) of the $n = 14$ bows of the game "The Elder Scrolls V: Skyrim". As above, we consider uniform weights $W = \frac{1}{n}\mathrm{Id}_n$, X denotes the data and Σ the covariance matrix. The box plot of the data (see Figure 1.8) shows that the scale of the each variable can not be compared (the variable "Value" is too important). To avoid this problem, we deal with normalized data (*i.e.* M is the diagonal matrix with $1/\sigma(x^i)$'s).

	Weight	Value	Damage	Speed
Long Bow	5	30	6	1
Hunting Bow	7	50	7	0.9375
Orcish Bow	9	150	10	0.8125
Nord Hero Bow	7	200	11	0.875
Dwarven Bow	10	270	12	0.75
Elven Bow	12	470	13	0.6875
Glass Bow	14	820	15	0.625
Ebony Bow	16	1440	17	0.5625
Daedric Bow	18	2500	19	0.5
Dragonbone Bow	20	2725	20	0.75
Crossbow	14	120	19	1
Enhanced Crossbow	15	200	19	1
Dwarven Crossbow	20	350	22	1
Enhanced Dwarven Crossbow	21	550	22	1

$$X = \begin{bmatrix} 5 & 30 & 6 & 1 \\ 7 & 50 & 7 & 0.9375 \\ 9 & 150 & 10 & 0.8125 \\ 7 & 200 & 11 & 0.875 \\ 10 & 270 & 12 & 0.75 \\ 12 & 470 & 13 & 0.6875 \\ 14 & 820 & 15 & 0.625 \\ 16 & 1440 & 17 & 0.5625 \\ 18 & 2500 & 19 & 0.5 \\ 20 & 2725 & 20 & 0.75 \\ 14 & 120 & 19 & 1 \\ 15 & 200 & 19 & 1 \\ 20 & 350 & 22 & 1 \\ 21 & 550 & 22 & 1 \end{bmatrix}$$

$$\Sigma = \frac{1}{n} {}^t\bar{X}\bar{X} = \begin{bmatrix} 25.8163265 & 2588.0612245 & 25.1530612 & -0.1377551 \\ 2588.0612245 & 7.3538023 \times 10^5 & 2050.3061224 & -98.2844388 \\ 25.1530612 & 2050.3061224 & 26.6938776 & -0.0325255 \\ -0.1377551 & -98.2844388 & -0.0325255 & 0.0294962 \end{bmatrix}$$

$$M = \begin{bmatrix} 1/\sigma^2(x^1) & 0 & 0 & 0 \\ 0 & 1/\sigma^2(x^2) & 0 & 0 \\ 0 & 0 & 1/\sigma^2(x^3) & 0 \\ 0 & 0 & 0 & 1/\sigma^2(x^4) \end{bmatrix} = \begin{bmatrix} 0.0387352 & 0 & 0 & 0 \\ 0 & 1.3598407 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 0.0374618 & 0 \\ 0 & 0 & 0 & 33.9027027 \end{bmatrix}$$

```
boxplot(Xbar,col="orange")
boxplot(Xbar %*% sqrt(M),col="blue")
```

Two dimensions seems to be a good trade-off (94.03% of explained inertia).

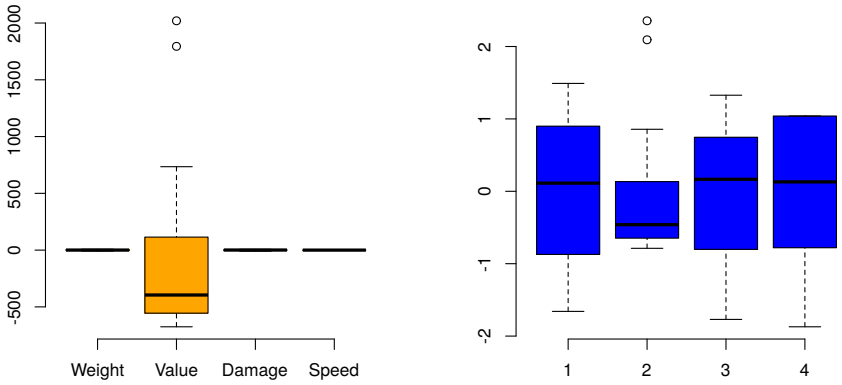

Figure 1.8: Box plot of the centered "Skyrim bows" data with uniform weigths (not normalized and normalized)

```
Mhalf <- diag(sqrt(diag(M)))
dg <- eigen(Mhalf %*% Sigma %*% Mhalf)
dg$vectors <- diag(1/sqrt(diag(M))) %*% dg$vectors
dg$values
```

```
## [1] 2.5110468 1.2502687 0.2094900 0.0291945
```

```
cumsum(dg$values)/sum(dg$values)
```

```
## [1] 0.6277617 0.9403289 0.9927014 1.0000000
```

```
plot(1:4, dg$values, type="b", col="orange")
```

1.2.3 Principal components and representation of the individuals

The M-orthonormal eigenvectors $v^1, \dots, v^p$ are a basis of $\mathbb{R}^p$. Thus, we can write the observed data X into this basis. For $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, p\}$, we denote by c_i^j the coordinate of $\tilde{x}_i = x_i - g(x)$ along the vector v^j,

$$c_i^j = \langle \tilde{x}_i, v^j \rangle_M = {}^t\tilde{x}_i M v^j \ .$$

This leads us to consider p new centered vectors of observations $c^1, \dots, c^p \in \mathbb{R}^n$ given by

$$c^j = \begin{pmatrix} c_1^j \\ \vdots \\ c_n^j \end{pmatrix} = \bar{X} M v^j \ .$$

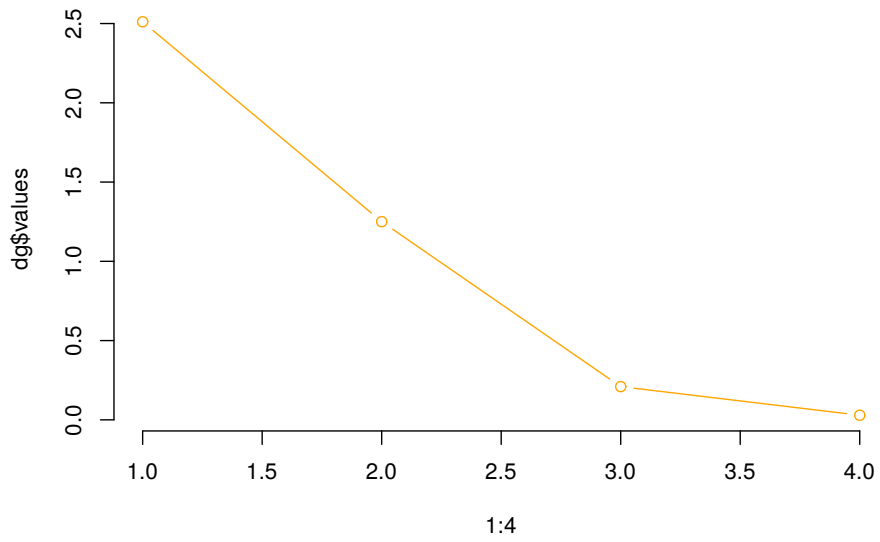

Figure 1.9: Ordered eigenvalues of the "Skyrim bows" data

$c^1, \dots, c^p$ are called the **principal components** and the $n \times p$-data matrix C associated to them is the **principal components matrix**,

$$C = \bar{X}MV$$

where V is the transformation matrix from the standard basis to the basis of the v^j's,

$$V = \left[v^1 \dots v^p\right] = \begin{bmatrix} v_1^1 & \dots & v_1^p \\ \vdots & \dots & \vdots \\ v_p^1 & \dots & v_p^p \end{bmatrix} .$$

The principal components are centered and noncorrelated. Indeed, we can easily compute the covariance matrix associated to C,

$${}^tCWC = {}^tV\,{}^tM\,{}^t\bar{X}W\bar{X}MV = {}^tV\left(M\Sigma M\right)V = \Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_p \end{bmatrix} ,$$

because V is an M-orthogonal matrix formed by the M-orthonormal eigenvectors of ΣM. Thus, for any $j \neq k \in \{1, \dots, p\}$, we have

$$\sigma^2(c^j) = \lambda_j \text{ and } \sigma(c^j, c^k) = 0 .$$

Thus, the first column of C contains the coordinates of the n individuals along the first principal axe, the second columns contains their coordinates along the second principal axe,

... This allows us to represent the individuals in E_d. In particular, E_2 is called the **principal plan** and the coordinates of the n individuals in E_2 are given by the two first columns of C. Moreover, for any $i \in \{1, \dots, n\}$, we can measure the quality of the representation of x_i by the quantity

$$\cos^2 \theta_i = \frac{\|\pi_{E_2}(\tilde{x}_i)\|_M^2}{\|\tilde{x}_i\|_M^2} = \frac{\left(c_i^1\right)^2 + \left(c_i^2\right)^2}{\left(c_i^1\right)^2 + \cdots + \left(c_i^p\right)^2} \ .$$

When we produce the scatter plot of the data in E_2, it can be usefull to draw the points with a size proportionnal to this quantity.

To discuss about each individual and detect outliers, we can now define several quantities as the contribution of the i-th individual to the inertia,

$$\frac{w_i \|\tilde{x}_i\|_M^2}{I_M(x)} \ ,$$

the contribution of the i-th individual to the j-th principal component,

$$\frac{w_i (c_i^j)^2}{\lambda_j} \ , \ \cdots$$

If we have at our disposal a new individual s, we do not need to compute again all the stuff for adding s to the plot in E_2. Indeed, it suffices to consider $\tilde{s} = s - g(x)$ and to plot

$$\left(\langle \tilde{s}, v^1 \rangle_M, \langle \tilde{s}, v^2 \rangle_M\right) = \left({}^t\tilde{s} M v^1, {}^t\tilde{s} M v^2\right) \ .$$

Grades

Because $M = \mathrm{Id}_p$, we have $C = \bar{X}V$ and we can easily compute the quality of the representation of each individual in E_2 and the contribution of each individual to the two first principal components.

```
C <- Xbar %*% dg$vectors
cos2 <- rowSums(C[,1:2]^2)/rowSums(C^2)
cos2
```

```
##     Benny     Bobby    Brandy      Coby     Daisy     Emily      Judy
## 0.9998728 0.9996600 0.9986273 0.9997552 0.9990726 0.9992720 0.9993354
##     Marty     Sandy
## 0.9980322 0.9807683
```

```
rbind(C[,1]^2/(nrow(X)*dg$values[1]),C[,2]^2/(nrow(X)*dg$values[2]))
```

```
##           Benny       Bobby    Brandy        Coby      Daisy     Emily
## [1,] 0.29186747 0.059205877 0.0406348 0.381947288 0.16151891 0.0362031
## [2,] 0.01834526 0.002329559 0.1110988 0.003319477 0.03868406 0.2236650
##            Judy      Marty       Sandy
## [1,] 0.00413805 0.01502477 0.009459738
## [2,] 0.37559607 0.16288732 0.064074536
```

Thus, we obtain Figure 1.10 when we draw the scatter plot in E_2

```
plot(C[,1:2], pch=2, cex=cos2, col="orange", xlab="", ylab="")
```

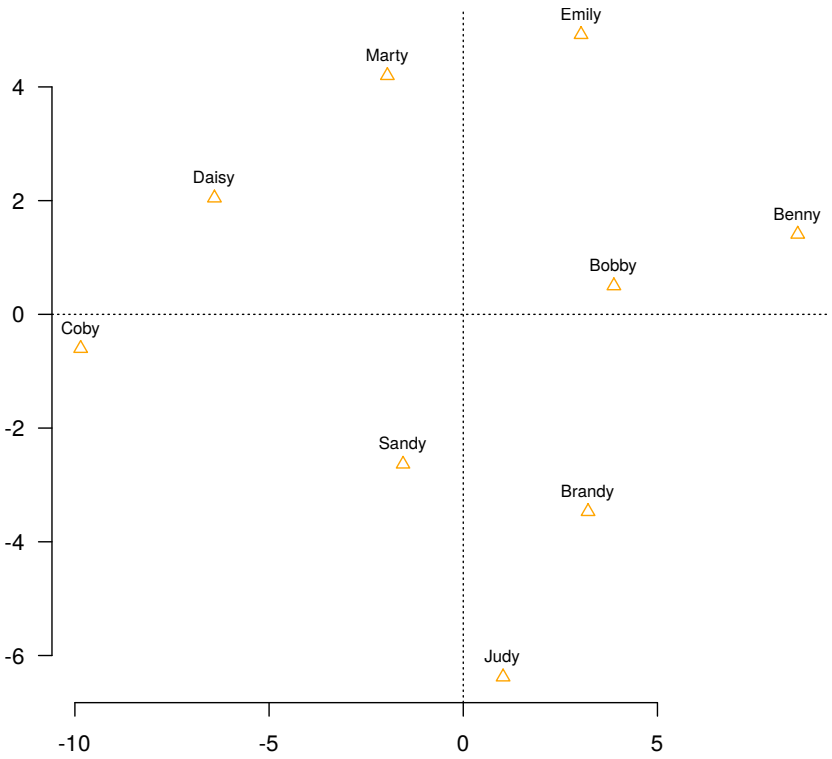

Figure 1.10: Scatter plot of the grades data in E_2

Skyrim bows

We have $C = \bar{X}MV$ and we can proceed as above. In particular, we get Figure 1.11 in the principal plan.

```
C <- Xbar %*% M %*% dg$vectors
cos2 <- rowSums(C[,1:2]^2)/rowSums(C^2)
cos2
```

```
##                  Long Bow                Hunting Bow
##                 0.9348944                  0.9590128
##                Orcish Bow              Nord Hero Bow
##                 0.9852860                  0.9574949
##               Dwarven Bow                  Elven Bow
##                 0.8813076                  0.6941128
##                 Glass Bow                  Ebony Bow
##                 0.7384271                  0.9496699
##               Daedric Bow             Dragonbone Bow
##                 0.9962952                  0.8511160
##                  Crossbow          Enhanced Crossbow
##                 0.9466848                  0.9816916
##          Dwarven Crossbow Enhanced Dwarven Crossbow
##                 0.9880126                  0.9878681
```

```
plot(C[,1:2], pch=2, cex=cos2, col="orange", xlab="", ylab="")
```

1.2.4 Representation of the variables and biplot

In order to get a graphical representation of the variables, we can proceed as above but with ${}^t\bar{X}$ in the place of $\bar{X}$. In particular, W and M are inverted and we deal with a scalar product $\langle \cdot, \cdot \rangle_W$. Thus, we need to get two W-orthonormal vectors $u^1, u^2 \in \mathbb{R}^n$ that are eigenvectors of $\bar{X}M\,{}^t\bar{X}W$ associated to the two largest eigenvalues. Hereafter, we assume that $\min(n, p) \geqslant 2$ which is a quite common situation. Basic linear algebra gives us

$$u^1 = \frac{c^1}{\sqrt{\lambda_1}} \text{ and } u^2 = \frac{c^2}{\sqrt{\lambda_2}} \ .$$

Indeed, we have

$$\begin{aligned} \bar{X}M\,{}^t\bar{X}Wu^1 &= \bar{X}M\,{}^t\bar{X}W\frac{c^1}{\sqrt{\lambda_1}} \\ &= \frac{1}{\sqrt{\lambda_1}}\bar{X}M\,{}^t\bar{X}W\bar{X}Mv^1 \\ &= \frac{1}{\sqrt{\lambda_1}}\bar{X}M\Sigma Mv^1 \\ &= \frac{\lambda_1}{\sqrt{\lambda_1}}\bar{X}Mv^1 = \frac{\lambda_1}{\sqrt{\lambda_1}}c^1 = \lambda_1 u^1 \ . \end{aligned}$$

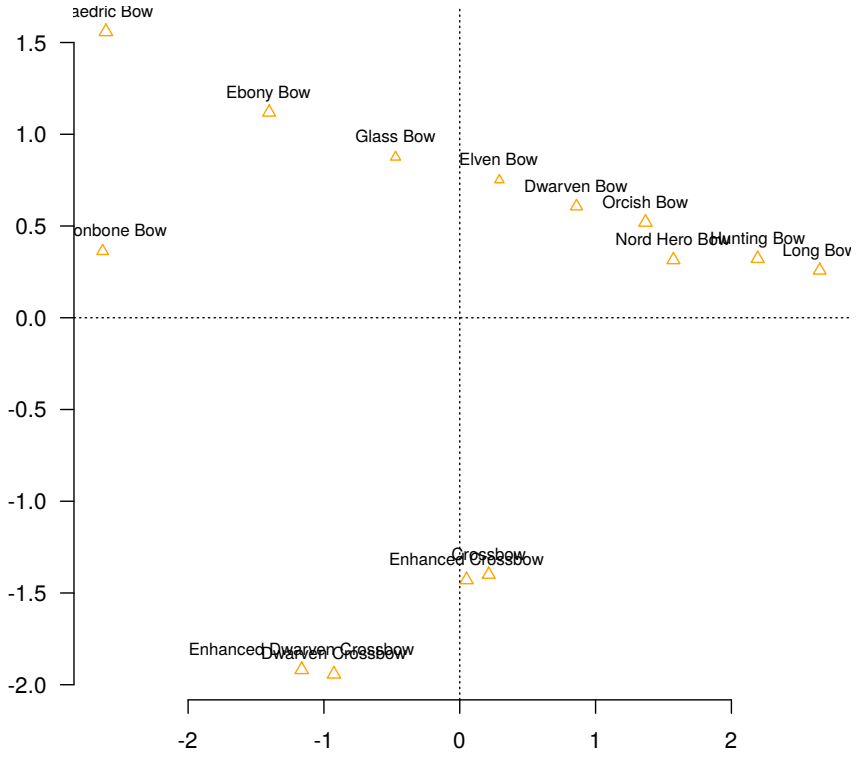

Figure 1.11: Scatter plot of the "Skyrim bows" data in E_2

Arguing in the same way shows that u^1 and u^2 are the two W-orthonormal eigenvectors of $\bar{X}M\,{}^t\bar{X}W$ associated to the two largest eigenvalues $\lambda_1 \geqslant \lambda_2$. As a consequence, for any $j \in \{1, \ldots, p\}$, the variable x^j can be represented by

$$\left(\langle x^j, u^1\rangle_W, \langle x^j, u^2\rangle_W\right) = \left(\sqrt{\lambda_1}v_j^1, \sqrt{\lambda_2}v_j^2\right) .$$

Note that, more generally, this is the two first columns of $V\Lambda^{1/2}$.

In order to interpret the role of the initial variable in the construction of the principal components, it is usefull to consider the correlation between them,

$$\rho(x^j, c^k) = \frac{\sigma(x^j, c^k)}{\sigma(x^j)\sqrt{\lambda_k}} = \frac{\langle x^j, c^k\rangle_W}{\sigma(x^j)\sqrt{\lambda_k}} = \frac{\langle x^j, u^k\rangle_W}{\sigma(x^j)} = \frac{\sqrt{\lambda_k}}{\sigma(x^j)}v_j^k$$

because c^k is centered and then $\sigma(x^j, c^k) = \langle x^j, c^k\rangle_W$, by construction. It is easy to verify that the points

$$\mathfrak{P}_j(\rho(x^j, c^1), \rho(x^j, c^2))$$

are in the unit circle (compute $\sigma^2(x^j) = \|\tilde{x}^j\|_W^2$ w.r.t. the W-orthonormal basis $u^1, \ldots, u^p \in \mathbb{R}^n$). Representation of these points is known as the **correlation circle** (see Figures 1.12

and 1.14). The closer a variable is to the circle, the better it is represented in the considered direction of E_2. Note that the last notice $(\lambda_1(v_j^1)^2 + \cdots + \lambda_p(v_j^p)^2 = \sigma^2(x^j))$ allows us to interpret $\rho(x^j, c^k)^2$ as the contribution of c^k to x^j.

The fact that the axes of E_2 and the correlation circle are the same up to a homothetic transformation allow us to represent E_2 and the correlation circle on the same graphic. Such a representation is known as a **biplot** and we generally use two different scales for keeping the biplot as clear as possible (see Figures 1.13 and 1.15). It is important to notice that there exists other choices for the scale of the axes in biplot. Indeed, we use C and $D_\sigma^{-1}V\Lambda^{1/2}$ for plotting but one also can use C and V (line isometry, this is what R does with prcomp and biplot), U and $V\Lambda^{1/2}$ (column isometry), ...

Grades

```
Rho <- diag(1/sqrt(diag(Sigma))) %*% dg$vectors %*% diag(sqrt(dg$values))
Rho[,1:2]
```

```
##             [,1]       [,2]
## Math -0.8111521  0.5844514
## Phys -0.9018802  0.4305779
## Fr   -0.7531811 -0.6573021
## Eng  -0.9148759 -0.4007291
```

```
rowSums(Rho[,1:2]^2)
```

```
##      Math      Phys        Fr       Eng
## 0.9995511 0.9987852 0.9993277 0.9975817
```

```
DrawCorCircle(Rho)
```

```
DrawBiplot(C,Rho)
```

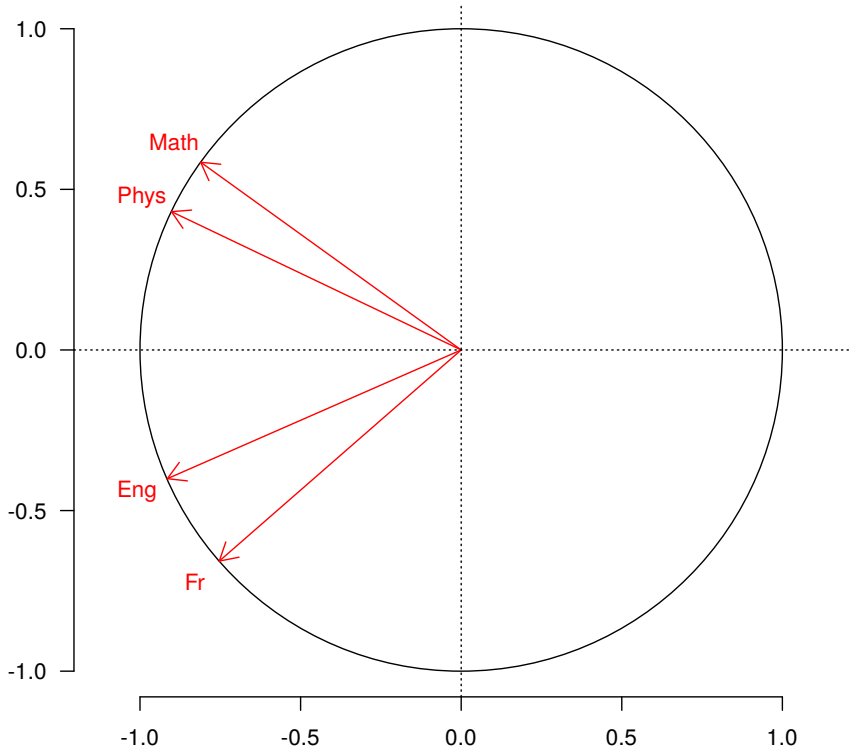

Figure 1.12: Correlation circle of the grades data

Skyrim bows

```
Rho <- diag(1/sqrt(diag(Sigma))) %*% dg$vectors %*% diag(sqrt(dg$values))
Rho[,1:2]

##               [,1]       [,2]
## Weight -0.9202730 -0.3667223
## Value  -0.8375940  0.4252481
## Damage -0.8518698 -0.4961416
## Speed   0.4867222 -0.8299343

rowSums(Rho[,1:2]^2)

##    Weight     Value    Damage     Speed
## 0.9813877 0.8823996 0.9718387 0.9256895

DrawCorCircle(Rho)
```

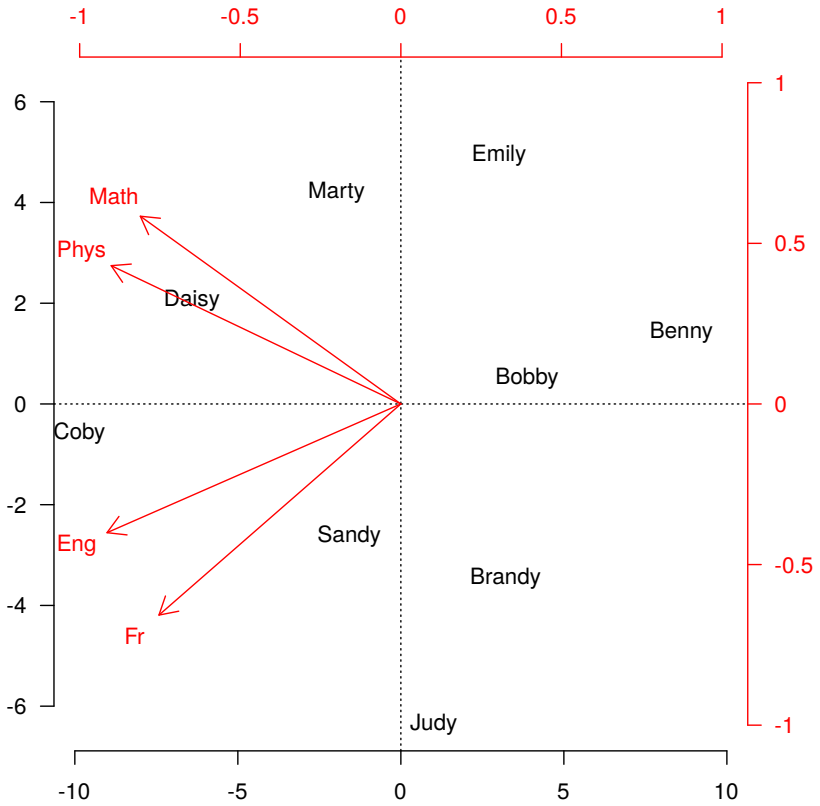

Figure 1.13: Biplot of the grades data

```
DrawBiplot(C,Rho)
```

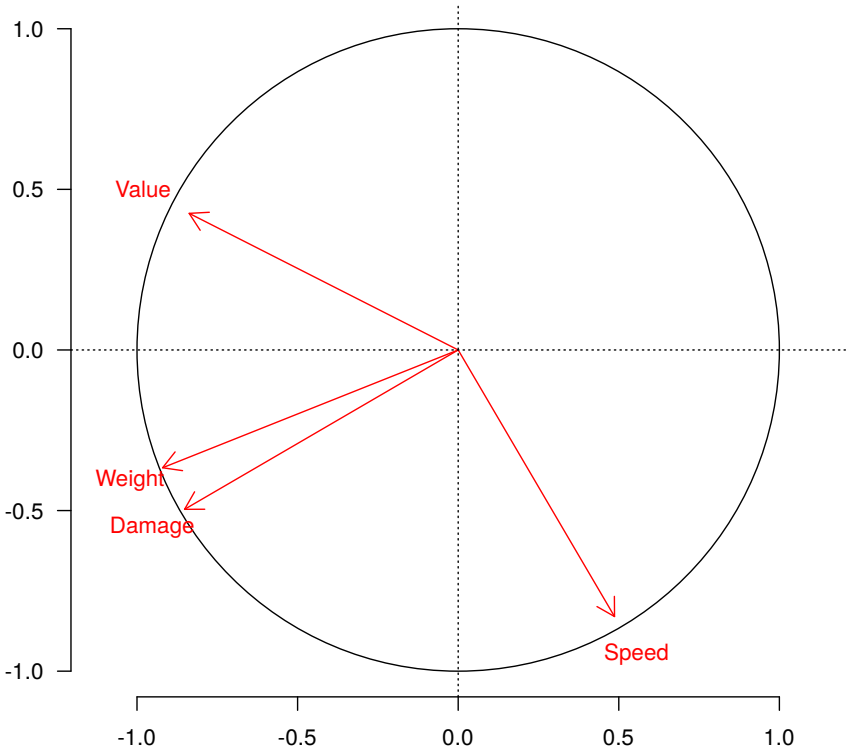

Figure 1.14: Correlation circle of the "Skyrim bows" data

1.2.5 Summary

We brievely summarize the various notations that we handle in this section. First, we had the input matrices:

- X : $n \times p$ matrix of the data,
- W : $n \times n$ matrix (gives the scalar product $\langle \cdot, \cdot \rangle_W$ in the space of variables),
- M : $p \times p$ matrix (gives the scalar product $\langle \cdot, \cdot \rangle_M$ in the space of individuals).

Based on these objects, we have defined the following ones:

- $\bar{X}$: $n \times p$ matrix of centered data,
- $\Sigma = {}^t\bar{X}W\bar{X}$: $p \times p$ covariance matrix,
- V : $p \times p$ transformation matrix from the standard basis to the M-orthonormal basis given by the eigenvectors $v^1, \ldots, v^p$ of ΣM,
- $C = \bar{X}MV$: $n \times p$ principal components matrix.

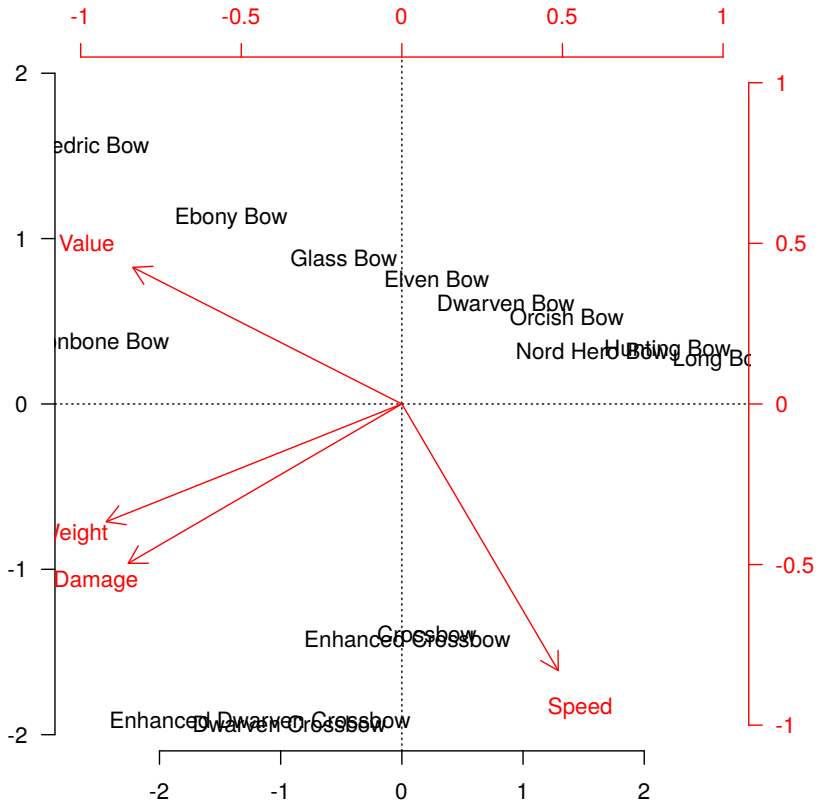

Figure 1.15: Biplot of the "Skyrim bows" data

1.3 Correspondence analysis

In the previous section, we have introduced PCA. One of the application of this method is that it allows us to discuss about the correlation structure of the variables. Moreover, it helps us to give a graphical representation of this structure. A priori, this work is only available for quantitative data because we need to quantify the distances and the variations. In this section, we will see that we can handle quantitative data and use PCA in order to reach a similar goal. The important fact is to understand what we call "individual" and "variable". Thus, the general outlook that we had on PCA (X, W and M) will can be applied to other kinds of problems. In particular, we are going to see that we can deal with the question of correspondences between the states of two qualitative variables. This procedure is known as the **correspondence analysis**.

1.3.1 Introduction

Let us consider two qualitative variables x and y such that:

- x can take the p values of $\{x_1, \ldots, x_p\}$,

- y can take the q values of $\{y_1, \dots, y_q\}$.

We observe n times the variable couple (x, y) and we have at our disposal the data associated to these observations. Usually, this kind of data set is represented by the **contingency table** $T = (n_{ij})_{1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q}$ that is the $p \times q$ matrix given by

$$\forall (i,j) \in \{1, \dots, p\} \times \{1, \dots, q\},\ n_{ij} = \sharp \{i \in \{1, \dots, n\} \text{ s.t. } (x_i, y_j) \text{ is observed})\} \ .$$

When T is represented, we add the **marginal totals** of each line,

$$n_{i\cdot} = \sum_{j=1}^{q} n_{ij},\ i \in \{1, \dots, p\} \ ,$$

and of each column,

$$n_{\cdot j} = \sum_{i=1}^{p} n_{ij},\ j \in \{1, \dots, q\} \ .$$

Of course, the **grand total** is n,

$$\sum_{i=1}^{p} n_{i\cdot} = \sum_{j=1}^{q} n_{\cdot j} = n \ ,$$

and T take the following form,

	y_1	$\dots$	y_j	$\dots$	y_q	Total
x_1	n_{11}	$\dots$	n_{1j}	$\dots$	n_{1q}	$n_{1\cdot}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_i	n_{i1}	$\dots$	n_{ij}	$\dots$	n_{iq}	$n_{i\cdot}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_p	n_{p1}	$\dots$	n_{pj}	$\dots$	n_{pq}	$n_{p\cdot}$
Total	$n_{\cdot 1}$	$\dots$	$n_{\cdot j}$	$\dots$	$n_{\cdot q}$	n

The goals of CA (correspondence analysis) are

- to describe the correspondences between the values taken by the variables x and y (*i.e.* does the observation of x_1 influence the fact that I also observe y_5?),
- to give a graphical representation of the correspondence.

To reach these goals, the method consists in doing a double PCA on the lines and on the columns of T.

Note that this framework implies that the order of the values that can take x and y are ignored. Moreover, we assume that each individual has exactly one value for x and one value for y (no multiple choices allowed) and that all the values are observed at least one time (if not, we can throw it away).

Example

All along this section, we will handle the following example. The data come from the work of the french sociologist Pierre Bourdieu and are based on $n = 8869$ students. They represent the socioprofessional category of his/her father (variable x):

EAG Exploitant agricole

SAG Salarié agricole

PT Patron

PLCS Profession libérale & cadre supérieur

CM Cadre moyen

EMP Employé

OUV Ouvrier

OTH Other

and the kind of studies followed by the student (variable y):

DR Droit

SCE Sciences éco.

LET Lettres

SC Sciences

MD Médecine ou dentaire

PH Pharmacie

PD Pluridisciplinaire

IUT IUT

	DR	**SCE**	**LET**	**SC**	**MD**	**PH**	**PD**	**IUT**	Total
EAG	80	36	134	99	65	28	11	58	511
SAG	6	2	15	6	4	1	1	4	39
PT	168	74	312	137	208	53	21	62	1035
PLCS	470	191	806	400	876	164	45	79	3031
CM	236	99	493	264	281	56	36	87	1552
EMP	145	52	281	133	135	30	20	54	850
OUV	16	6	27	11	8	2	2	8	80
OTH	305	115	624	247	301	47	42	90	1771
Total	1426	575	2692	1297	1878	381	178	442	8869

1.3.2 Profiles

The two PCA that we will do are based on vectors of frequencies called **profiles**. According to the case, we consider the x_i's as the individuals and the y_j's as the variables (PCA on the line profiles) or the y_j's as the individuals and the x_i's as the variables (PCA on the column profiles).

Line profiles

Let $i \in \{1, \dots, p\}$, the **line profile** associated to the observations of x_i is the vector $f^{(x_i)} \in [0,1]^q$ of the frequencies of the observations of each value of y with x_i, namely

$$f^{(x_i)} = \begin{pmatrix} n_{i1}/n_{i\cdot} \\ \vdots \\ n_{iq}/n_{i\cdot} \end{pmatrix} = \frac{1}{n_{i\cdot}} \begin{pmatrix} n_{i1} \\ \vdots \\ n_{iq} \end{pmatrix} .$$

Putting these p profiles in the lines of a $p \times q$ matrix P_1 leads to the **line profiles matrix**,

$$P_1 = D_1 T = \begin{bmatrix} {}^t f^{(x_1)} \\ \vdots \\ {}^t f^{(x_p)} \end{bmatrix} = \begin{bmatrix} n_{11}/n_{1\cdot} & \dots & n_{1q}/n_{1\cdot} \\ \vdots & \ddots & \vdots \\ n_{p1}/n_{p\cdot} & \dots & n_{pq}/n_{p\cdot} \end{bmatrix}$$

with $D_1 = \mathrm{diag}(1/n_{1\cdot}, \dots, 1/n_{p\cdot})$.

By definition, the line profiles are all in the subspace of $\mathbb{R}^q$ of dimension $q-1$ defined by

$$Z_q = \left\{ v \in \mathbb{R}^q \text{ s.t. } \sum_{j=1}^{q} v_j = 1 \right\} .$$

Seeing P_1 as the data matrix, we handle q "variables" that correspond to the various values of y and we observe these "variables" on p "individuals" that corresponds to the values of x. For $i \in \{1, \dots, p\}$, we associate to the "individual" x_i the weight corresponding to its frequency,

$$w_{1,i} = \frac{n_{i\cdot}}{n} .$$

Note that this choice corresponds to take uniform weights $1/n$ on the n observations. Then, we can compute the center of gravity $g_1 = (g_{1,1}, \dots, g_{1,q})' \in \mathbb{R}^q$ associated to it,

$$\forall j \in \{1, \dots, q\},\ g_{1,j} = \sum_{i=1}^{p} w_{1,i} f_j^{(x_i)} = \frac{1}{n} \sum_{i=1}^{p} n_{i\cdot} \times \frac{n_{ij}}{n_{i\cdot}} = \frac{n_{\cdot j}}{n} .$$

Thus, g_1 is the vector of the **marginal frequencies** of the variable y.

Column profiles

Let $j \in \{1, \dots, q\}$, we define in the same way the column profile associated to the observations of y_j by

$$f^{(y_j)} = \begin{pmatrix} n_{1j}/n_{\cdot j} \\ \vdots \\ n_{pj}/n_{\cdot j} \end{pmatrix} = \frac{1}{n_{\cdot j}} \begin{pmatrix} n_{1j} \\ \vdots \\ n_{pj} \end{pmatrix} \in [0,1]^p .$$

Putting these q profiles in the lines of a $q \times p$ matrix P_2 leads to the **column profiles matrix**,

$$P_2 = D_2\, {}^tT = \begin{bmatrix} {}^tf^{(y_1)} \\ \vdots \\ {}^tf^{(y_q)} \end{bmatrix} = \begin{bmatrix} n_{11}/n_{\cdot 1} & \dots & n_{p1}/n_{\cdot 1} \\ \vdots & \ddots & \vdots \\ n_{1q}/n_{\cdot q} & \dots & n_{pq}/n_{\cdot q} \end{bmatrix}$$

with $D_2 = \mathrm{diag}(1/n_{\cdot 1}, \dots, 1/n_{\cdot q})$.

As above, the column profiles are all in the subspace of $\mathbb{R}^p$ of dimension $p-1$ defined by

$$Z_p = \left\{ u \in \mathbb{R}^p \text{ s.t. } \sum_{i=1}^{p} u_i = 1 \right\} .$$

Moreover, for any $j \in \{1, \dots, q\}$, the weight of the "individual" y_j is its frequency,

$$w_{2,j} = \frac{n_{\cdot j}}{n}$$

and we can compute the center of gravity $g_2 = (g_{2,1}, \dots, g_{2,p})' \in \mathbb{R}^p$,

$$\forall i \in \{1, \dots, p\},\ g_{2,i} = \sum_{j=1}^{q} w_{2,j} f_i^{(y_j)} = \frac{1}{n} \sum_{j=1}^{q} n_{\cdot j} \times \frac{n_{ij}}{n_{\cdot j}} = \frac{n_{i\cdot}}{n} .$$

Again, g_2 is the vector of the **marginal frequencies** of the variable x.

Profiles of independence

As explained above, we are interested in describing the correspondences between the values taken by x and by y. To discuss, we compare the profiles obtained from the table T to theoritical profiles that should appear if there is no significative correspondence. Such a case is known as **independence** between the variables x and y and can be express as follows.

Let $i \in \{1, \dots, p\}$, if we observe x_i, the variable y take its various values $y_1, \dots, y_q$ with frequencies that are independent from x_i. So, the frequency of simultaneous observation of x_i and y_j is simply the marginal frequency of y_j, namely $n_{\cdot j}/n$. Thus, the theoritical line profile associated to the observations of x_i, in case of independence, is given by the center of gravity g_1,

$$\tilde{f}^{(x_i)} = g_1 = \begin{pmatrix} n_{\cdot 1}/n \\ \vdots \\ n_{\cdot q}/n \end{pmatrix} .$$

Let $j \in \{1, \dots, q\}$, arguing in the same way as above, the theoritical column profile associated to the observations of y_j, in case of independence, is given by the center of gravity g_2,

$$\tilde{f}^{(y_j)} = g_2 = \begin{pmatrix} n_{1\cdot}/n \\ \vdots \\ n_{p\cdot}/n \end{pmatrix} .$$

We now want to be able to compare the observed profiles with the theoritical ones. Let $u, v \in Z_q$ be two line profiles, we compare them by introducing the χ^2 **distance** between u and v,

$$\sum_{j=1}^{q} \frac{(u_j - v_j)^2}{g_{1,j}} = \sum_{j=1}^{q} \frac{n}{n_{\cdot j}} (u_j - v_j)^2 = \|u - v\|^2_{nD_2} .$$

Note that the sum is weighted by the coordinates of g_1. It amounts to give more importance to the relative differences between u and v for the less frequent values of y_j. Similarly, for two columns profiles $u, v \in Z_p$, the χ^2 distance is defined by

$$\sum_{i=1}^{p} \frac{(u_i - v_i)^2}{g_{2,i}} = \sum_{i=1}^{p} \frac{n}{n_{i\cdot}}(u_j - v_j)^2 = \|u - v\|^2_{nD_1} \ .$$

1.3.3 Double PCA

To study the correspondences, we use two PCA procedures with the following parameters:

- PCA on line profiles:
  - Data matrix $P_1 = D_1 T$,
  - Center of gravity $g_1 = \frac{1}{n} D_2^{-1} \mathbf{1}_q = \frac{1}{n}\, {}^t T \mathbf{1}_p$,
  - Weight matrix $W_1 = \frac{1}{n} D_1^{-1}$,
  - Distance on space of individuals given by $M_1 = nD_2$,
- PCA on column profiles:
  - Data matrix $P_2 = D_2\, {}^t T$,
  - Center of gravity $g_2 = \frac{1}{n} D_1^{-1} \mathbf{1}_p = \frac{1}{n} T \mathbf{1}_q$,
  - Weight matrix $W_2 = \frac{1}{n} D_2^{-1}$,
  - Distance on space of individuals given by $M_2 = nD_1$,

where, for any k, $\mathbf{1}_k = (1, \ldots, 1)' \in \mathbb{R}^k$.

Line profiles

Before dealing with PCA, let us make the following remark. Using $\sigma(x, y) = \overline{xy} - \overline{x} \times \overline{y}$, it is easy to verify that the covariance matrix Σ_1 associated to the data matrix P_1 is given by

$$\Sigma_1 = {}^t P_1 W_1 P_1 - g_1\, {}^t g_1 = \frac{1}{n}\, {}^t T D_1 T - g_1\, {}^t g_1 \ .$$

Thus, we are looking for the eigenvalues of

$$\Sigma_1 M_1 = {}^t T D_1 T D_2 - n g_1\, {}^t g_1 D_2 \ .$$

The rank of $n g_1\, {}^t g_1 D_2$ is 1 and its only nontrivial eigenvector is g_1 (associated to the eigenvalue 1). Indeed,

$$n g_1\, {}^t g_1 D_2 g_1 = g_1\, {}^t g_1 \mathbf{1}_q = g_1 \ .$$

Moreover, g_1 is also an eigenvector of ${}^t T D_1 T D_2$ associated to the eigenvalue 1,

$${}^t T D_1 T D_2 g_1 = \frac{1}{n}\, {}^t T D_1 T \mathbf{1}_q = {}^t T D_1 g_2 = g_1 \ .$$

Thus, basic linear algebra gives us that $\Sigma_1 M_1$ and ${}^t T D_1 T D_2$ have the same eigenvalues apart from the one associated to g_1 (that is trivial due to the stochastic nature of the matrix P_1).

We know that ${}^tTD_1TD_2$ admits at most $\kappa = \min(p,q) - 1$ nontrivial eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_\kappa \geqslant 0$. We consider the M_1-orthonormal eigenvectors associated to these eigenvalues and we obtain the $q \times \kappa$ matrix V_1. Then, we define the $p \times \kappa$ principal components matrix

$$C^{(1)} = P_1(nD_2)V_1 = nD_1TD_2V_1$$

and the coordinates of the "individual" x_i are given by the i-th line of $C^{(1)}$. Note that, despite the data are not centered, the principal component are centered because the trivial eigenvector, given by the center of gravity g_1, is not considered in V_1.

Column profiles

Arguing in the same way for the column profiles, we get κ nontrivial eigenvalues $\lambda'_1 \geqslant \cdots \geqslant \lambda'_\kappa \geqslant 0$ for the matrix $TD_2\,{}^tTD_1$ and the M_2-orthonormal associated eigenvectors in the $p \times \kappa$ matrix V_2. Then, we define the $q \times \kappa$ principal components matrix

$$C^{(2)} = P_2(nD_1)V_2 = nD_2\,{}^tTD_1V_2$$

and the coordinates of the "individual" y_j are given by the j-th line of $C^{(2)}$.

Transition principle

Note that the λ_i are the eigenvalues of $({}^tTD_1) \times (TD_2) = {}^tP_1\,{}^tP_2$ and that the λ'_i are the eigenvalues of $(TD_2) \times ({}^tTD_1) = {}^tP_2\,{}^tP_1$. So, we know that, for any $i \in \{1,\ldots,\kappa\}$, $\lambda_i = \lambda'_i$ and we denote $\Lambda = \mathrm{diag}(\lambda_1,\ldots,\lambda_\kappa)$. Moreover, by definition, we know that V_1 and V_2 are the unique matrices (up to a column permutation in case of multiple eigenvalues) such that

$${}^tP_1\,{}^tP_2V_1 = V_1\Lambda \text{ and } {}^tP_2\,{}^tP_1V_2 = V_2\Lambda\ .$$

Simple computation gives

$${}^tP_2\,{}^tP_1 \times {}^tP_2V_1\Lambda^{-1/2} = {}^tP_2V_1\Lambda^{1/2} = ({}^tP_2V_1\Lambda^{-1/2})\Lambda\ .$$

Similar argument leads to

$$V_1 = {}^tP_1V_2\Lambda^{-1/2} \text{ and } V_2 = {}^tP_2V_1\Lambda^{-1/2}\ .$$

So, we obtain

$$C^{(1)} = nD_1\,{}^tP_2V_1 = nD_1 \times \underbrace{{}^tP_2\,{}^tP_1V_2}_{=V_2\Lambda} \times \Lambda^{-1/2}$$

and, finally,

$$C^{(1)} = nD_1V_2\Lambda^{1/2} \text{ and } C^{(2)} = nD_2V_1\Lambda^{1/2}\ .$$

We also can write these equalities in the following way, known as **transition formulae**,

$$C^{(1)} = P_1C^{(2)}\Lambda^{-1/2} \text{ and } C^{(2)} = P_2C^{(1)}\Lambda^{-1/2}\ .$$

The main advantage of these formulae is that they allow to get the two principal components matrices by computing only one of them. Indeed, if you have at your disposal V_1 and $C^{(1)}$, you directly get V_2 and $C^{(2)}$ without working to get the eigenvalues.

1.3.4 Graphical representation

As for the biplot of PCA, there is no justified choice for normalizing the axes of the CA representation. A common choice is to plot the x_i's and the y_j's in the plan by using the two first columns of $C^{(1)}$ and $C^{(2)}$ respectively. Other choices are often used by normalizing one axe or the other (*e.g.* barycentric representation, ...).

All the criteria inherited from PCA are available. We can measure the global quality of the representation by the explained inertia,

$$\frac{\lambda_1 + \lambda_2}{\sum_{k=1}^{\kappa} \lambda_k},$$

measure the quality of the representation of x_i by

$$\frac{\left(C_{i1}^{(1)}\right)^2 + \left(C_{i2}^{(1)}\right)^2}{\sum_{k=1}^{\kappa} \left(C_{ik}^{(1)}\right)^2},$$

or the one of y_j by

$$\frac{\left(C_{j1}^{(2)}\right)^2 + \left(C_{j2}^{(2)}\right)^2}{\sum_{k=1}^{\kappa} \left(C_{jk}^{(2)}\right)^2}.$$

We also can consider the contribution of x_i to each principal component, ...

Bourdieu data

We begin by computing the line and column profiles matrices,

```
D1 <- diag(1/rowSums(T))
P1 <- D1 %*% T
D2 <- diag(1/colSums(T))
P2 <- D2 %*% t(T)
```

We now proceed to get the PCA of the line profiles matrix, as we did in the previous section, and the principal components matrix $C^{(1)}$.

```
M1half <- diag(sqrt(n/colSums(T)))
M1halfInv <- diag(sqrt(colSums(T)/n))
dg <- eigen(M1half %*% t(T) %*% D1 %*% T %*% D2 %*% M1halfInv)
dg$values <- dg$values[2:8] # Avoid the trivial eigenvlue
dg$vectors <- (M1halfInv %*% dg$vectors)[,2:8] # Idem
C1 <- P1 %*% (n*D2) %*% dg$vectors
```

With the aid of transition formulae, we directly get the principal components matrix $C^{(2)}$ and we can compute the various quantities relative to the quality of the representation, for instance.

```
C2 <- P2 %*% C1 %*% diag(1/sqrt(dg$values))
cumsum(dg$values)/sum(dg$values)
```

```
## [1] 0.7939074 0.9612612 0.9928056 0.9968960 0.9986875 0.9999757 1.0000000
```

```
# Representation quality of the x's values
rowSums(C1[,1:2]^2)/rowSums(C1^2)
```

```
##       EAG       SAG        PT      PLCS        CM       EMP       OUV
## 0.9987801 0.9029523 0.3170859 0.9994798 0.7107327 0.9778616 0.8647598
##       OTH
## 0.9914724
```

```
# Representation quality of the y's values
rowSums(C2[,1:2]^2)/rowSums(C2^2)
```

```
##        DR       SCE       LET        SC        MD        PH        PD
## 0.5035835 0.2708271 0.9927045 0.6677505 0.9990910 0.9605193 0.9764390
##       IUT
## 0.9889960
```

```
plot(dg$values,type="b",col="orange",xlab="",ylab="")
```

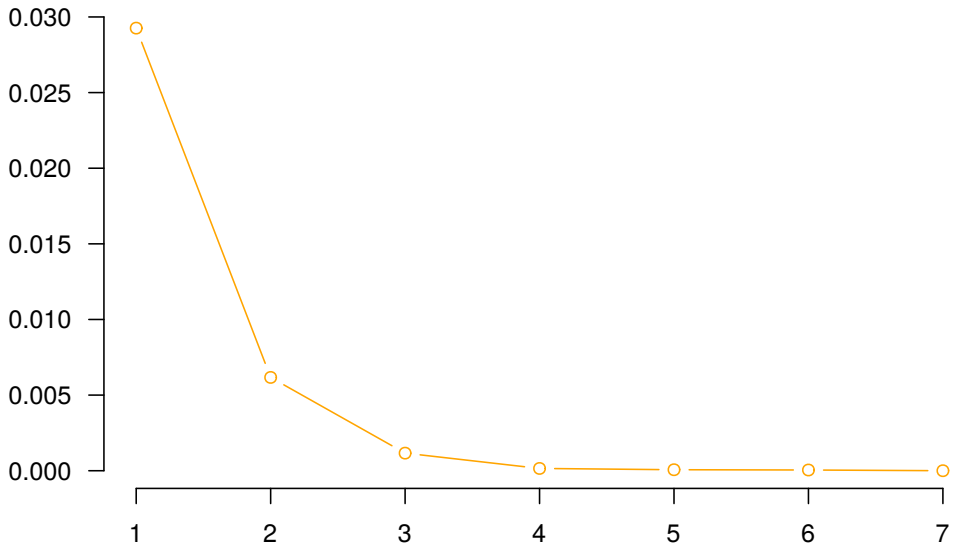

Figure 1.16: Eigenvalues of "Bourdieu data"

Finally, we can plot the CA representation and interpret the correspondences by considering the proximity between points.

```
DrawCA(C1,C2)
```

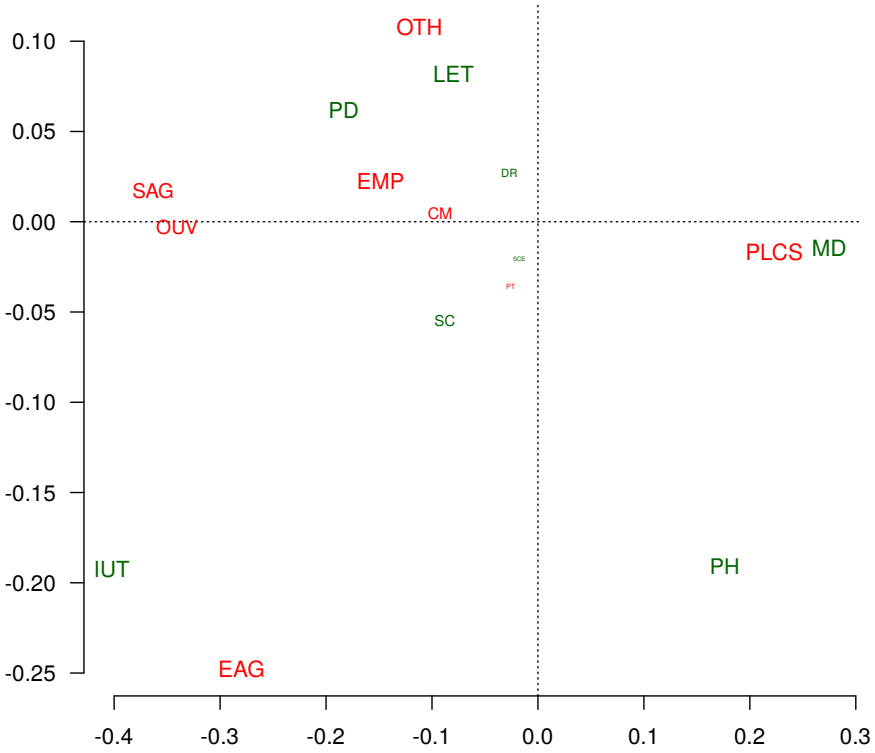

Figure 1.17: Correspondence analysis of "Bourdieu data"

1.4 Multiple correspondence analysis

By arguing in a similar way, it is also possible to deal with more than two variables. See the practical sessions.

1.5 Bibliography

- *The Elements of Statistical Learning : Data Mining, Inference and Prediction*, J. Friedman, T. Hastie and R. Tibshirani (2009)
- *Principal Component Analysis*, I.T. Jolliffe (2002)

- *Wiki Stat*, http://wikistat.fr/

Chapter 2

Supervised learning

The aim of supervised learning is to deal with labeled data in order to construct a procedure that predict the label of a new data. Various approach have been proposed to tackle this very common problem. We introduce here three such methods.

2.1 Multiple discriminant analysis

2.1.1 Introduction

After having seen the PCA procedure, we could naturally use it to deal with supervised learning. Indeed, when we interpreted the results of PCA, we have often consider that the principal axes allows us to distinguish different groups of data. But how far such an approach is valid? Actually, we never claim that PCA was a good tool for discrimating groups. PCA is only well adapted for getting better representation in a space of fixed dimension. A priori, there is no connection with label discrimination.

Lubischew data

To illustrate MDA (Multiple Discriminant Analysis), we handle the Lubischew data set. This set is about $n = 74$ insects and, for each one, we measure $p = 6$ morphologic sizes. Moreover, for each insect, we know its species (denoted by "A", "B" and "C") that plays the role of label information. We can begin by look at the result of a PCA procedure for this data set with uniform weights and standard Euclidean metric.

```
Xbar <- scale(X,scale=F)
Sigma <- t(Xbar) %*% Xbar / nrow(Xbar)
dg <- eigen(Sigma)
C <- Xbar %*% dg$vectors
cos2 <- rowSums(C[,1:2]^2)/rowSums(C^2)
plot(C[,1:2],pch=2,cex=cos2,xlab="",ylab="")
```

We see that the PCA helps us to suggest some groups but are they related to the species? Moreover, we know that we are dealing with three species. But how can we retrieve this information with unlabeled PCA? Let us make the species appear on this scatter plot.

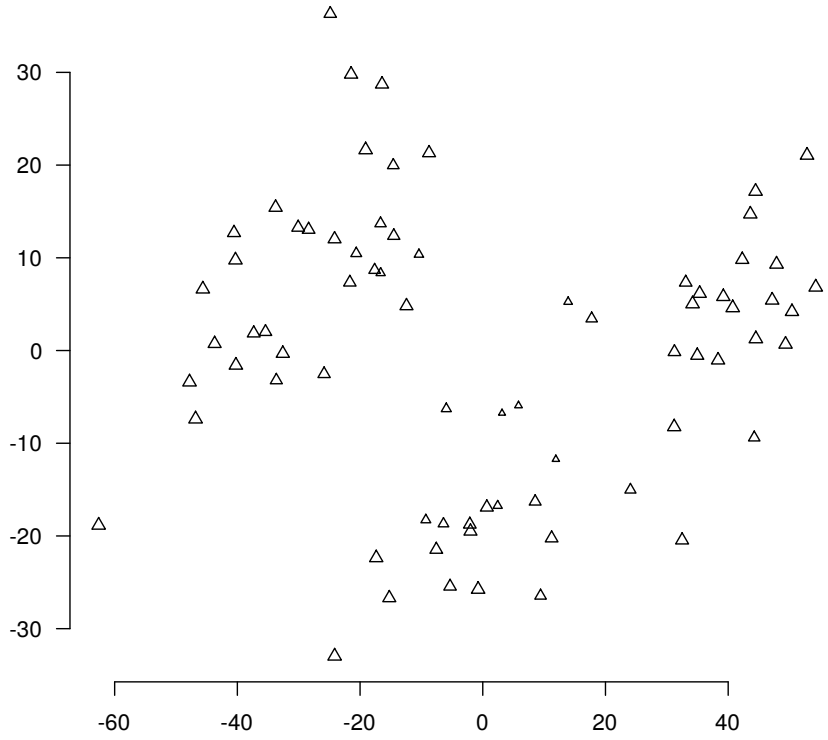

Figure 2.1: Scatter plot of the "Lubischew data" in E_2

```
text(C[,1:2],labels=data$V8,cex=cos2,col=species)
```

This time, information is clearer and we could use it to construct our supervised learning procedure. The remaining problem is how to choose a good frontier between the colored groups in Figure 2.2? Indeed, it could be easier to handle and more precise if the groups were organised in a clearer way. This is the aim of MDA procedure.

Framework

From a general point of view, we handle p quantitative variables $x^1, \ldots, x^p$ and one qualitative variable t that can take m different values in $\{\tau_1, \ldots, \tau_m\}$. Our data set is relative to n observations of these $p+1$ variables, *i.e.* for each $i \in \{1, \ldots, n\}$, the observation of the i-th individual gives

$$x_i = (x_i^1, \ldots x_i^p)' \in \mathbb{R}^p \text{ and } t_i \in \{\tau_1, \ldots, \tau_m\} \ .$$

Moreover, we associate a weight w_i to the i-th individual and we denote by $W = \text{diag}(w_1, \ldots, w_n)$ the weight matrix.

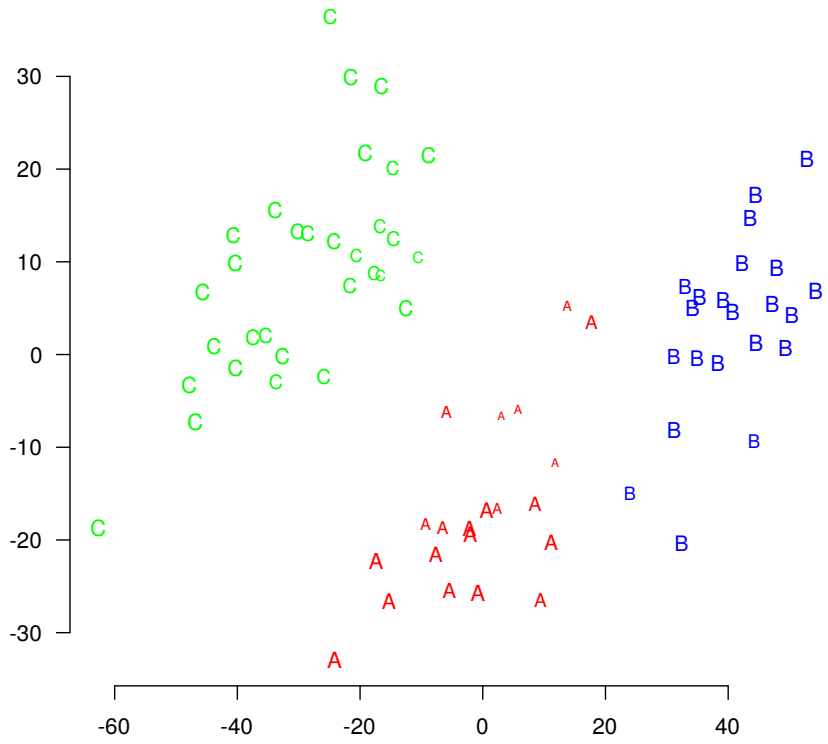

Figure 2.2: Scatter plot of the "Lubischew data" in E_2 with species

The variable t induce a partition of $\{1, \ldots, n\}$ given by the m blocks $\Omega_1, \ldots, \Omega_m$ defined as, for any $\ell \in \{1, \ldots, m\}$,

$$\Omega_\ell = \{i \in \{1, \ldots, n\} \text{ s.t. } t_i = \tau_\ell\} \ .$$

This partition can be represented in a matrix by the $n \times m$ matrix T given by

$$T_{i\ell} = \begin{cases} 1 & \text{if } i \in \Omega_\ell \\ 0 & \text{else} \end{cases} \ , \ i \in \{1, \ldots, n\}, \ \ell \in \{1, \ldots, m\} \ .$$

Note that, by construction, there is only one nonzero cell on each line of T. Moreover, we can associate a weight to a block Ω_ℓ by suming the weights of its elements,

$$\bar{w}_\ell = \sum_{i \in \Omega_\ell} w_i, \ \ell \in \{1, \ldots, m\} \ .$$

These grouped weights are given by the $m \times m$ diagonal matrix,

$$\bar{W} = {}^tTWT \ .$$

In the sequel, we implicitely assume that each value allowed for t is observed, at least, one time (*i.e.* $\bar{W}$ is of full rank). This hypothesis is quite free because, if one value is not observed, it suffices to throw it away from the allowed values of the variable t.

The goals of MDA are to:

- get the best representation of the groups Ω_ℓ's,
- allow us to predict the value of t for any new vector of observations of the variables $x^1, \ldots, x^p$.

You can notice that the two goals are quite similar. The first one take a PCA point of view and the second one is quite the definition of supervised learning.

To deal with our data, we introduce some additional notations. As usual, we denote by X the $n \times p$ data matrix associated to the observations of $x^1, \ldots, x^p$ and $\bar{X}$ its centered version. Let $\ell \in \{1, \ldots, m\}$, we can define the center of gravity $g_\ell \in \mathbb{R}^p$ of the group Ω_ℓ as we did for the "global" center of gravity g,

$$g_\ell = \frac{1}{\bar{w}_\ell} \sum_{i \in \Omega_\ell} w_i x_i \ ,$$

and we can put these centers of gravity in the lines of a $m \times p$ matrix G,

$$G = \bar{W}^{-1} \, {}^t T W X = \begin{bmatrix} {}^t g_1 \\ \vdots \\ {}^t g_m \end{bmatrix} \ .$$

These centers of gravity will help us to separate the groups in the way that we will put them the most far away from the others. To do that, we avoid global centering and we prefer to center according to each class. For this reason, let us introduce the $n \times p$ **matrix X_b of repeated centers of gravity**,

$$X_b = TG$$

and the following decomposition,

$$\bar{X} = \bar{X}_w + \bar{X}_b$$

where we have set

$$\bar{X}_w = X - X_b \text{ and } \bar{X}_b = X_b - \mathbf{1}_n \, {}^t g \ .$$

2.1.2 Covariance matrix decomposition

Note that this last decomposition of $\bar{X}$ is the similar basic argument for proving the decomposition of the variance of a real variable along some partition of the set of the index. It leads to the famous decomposition of the variance as the sum of the **variance within** (*i.e.* the mean of the variances) and the **variance between** (*i.e.* the variance of the means). Thus, we proceed in the same way by considering the $m \times p$ **centered centers of gravity matrix**,

$$\bar{G} = G - \mathbf{1}_m \, {}^t g \ ,$$

the $p \times p$ **covariance within matrix**,

$$\Sigma_w = {}^t \bar{X}_w W \bar{X}_w = \sum_{\ell=1}^{m} \sum_{i \in \Omega_\ell} w_i (x_i - g_\ell) \, {}^t (x_i - g_\ell)$$

and the $p \times p$ **covariance between matrix**,

$$\Sigma_b = {}^t\bar{G}\bar{W}\bar{G} = {}^t\bar{X}_b W \bar{X}_b = \sum_{\ell=1}^{m} \bar{w}_\ell (g_\ell - g)\,{}^t(g_\ell - g) \ .$$

Note that we obtain the same decomposition relation for the covariance matrix,

$$\Sigma = \Sigma_w + \Sigma_b \ .$$

This decomposition contains an important statistical idea. Indeed, Σ_b stands for what happens between the groups and Σ_w is related to what happens in the groups. The aim of MDA is to get the best discrimination between the groups. Thus, this is the term related to Σ_b that will lead us to our goal. Let M be some $p \times p$ positive symmetric matrix and $v^1, \ldots, v^d \in \mathbb{R}^p$ be M-orthonormal vectors, we know that the inertia of the projections of the observations in the set generated by the v^j's is given by

$$\sum_{j=1}^{d} \langle \Sigma M v^j, v^j \rangle_M = \underbrace{\sum_{j=1}^{d} \langle \Sigma_w M v^j, v^j \rangle_M}_{\text{dispersal in the groups}} + \underbrace{\sum_{j=1}^{d} \langle \Sigma_b M v^j, v^j \rangle_M}_{\text{dispersal between the groups}} \ .$$

Thus, we want to maximize what happens between the groups with respect to the global dispersal,

$$\frac{\displaystyle\sum_{j=1}^{d} \langle \Sigma_b M v^j, v^j \rangle_M}{\displaystyle\sum_{j=1}^{d} \langle \Sigma M v^j, v^j \rangle_M} \ .$$

The problem is the presence of the v^j's both in numerator and denominator. To avoid that and keep a linear problem to solve, we take $M = \Sigma^{-1}$ (assuming that this is possible) and we obtain, up to the numerical constant factor $1/d$,

$$\sum_{j=1}^{d} \langle \Sigma_b \Sigma^{-1} v^j, v^j \rangle_{\Sigma^{-1}} \ .$$

In other word, we are doing the PCA of $\bar{G}$ with the weight matrix $\bar{W}$ and a distance on the space of individuals given by $M = \Sigma^{-1}$. This distance is known as the **Mahalanobis distance**.

2.1.3 MDA procedure

Let $\kappa = \min(m-1, p)$, we know that we can find κ eigenvectors $v^1, \ldots, v^\kappa \in \mathbb{R}^p$ of the matrix $\Sigma_b \Sigma^{-1}$ that are Σ^{-1}-orthonormal and associated to the ordered eigenvalues $1 \geqslant \lambda_1 \geqslant \cdots \geqslant \lambda_\kappa \geqslant 0$ (the eigenvalues are in $[0,1]$ because of the normalization, see PCA on correlation matrix). The v^j's are called **discriminant axis** and λ_1 is the **power of discrimination**. If $\lambda_1 = 1$ all the points of each group are equal to the center of gravity of their group and if $\lambda_1 = 0$, no linear separation is possible (back to the choice of M).

Denoting by V the matrix of eigenvectors, we obtain the principal components matrix, as usual, for the centers of gravity,

$$\bar{C} = \bar{G}\Sigma^{-1}V \ ,$$

and also for the individuals,

$$C = \bar{X}\Sigma^{-1}V \ .$$

Then, we simultaneously plot the data and the centers of gravity. The quality of the representation is measured by the classical tools of PCA (cosine, ...).

Lubischew data

We compute the usefull matrix that we need to deal with MDA.

```
W <- diag(rep(1/n,n))
Wbar <- t(T) %*% W %*% T
WbarInv <- diag(1/diag(Wbar))
G <- WbarInv %*% t(T) %*% W %*% X
g <- colMeans(X)
Gbar <- G - rep(1,3) %*% t(g)
XB <- T %*% G
XBbar <- XB - rep(1,n) %*% t(g)
SigmaB <- t(Gbar) %*% Wbar %*% Gbar
```

Thus, we compute the eigenvectors as we did for PCA,

```
M <- solve(Sigma)
dgM <- eigen(M)
P <- dgM$vectors
Mhalf <- P %*% diag(sqrt(dgM$values)) %*% t(P)
dg <- eigen(Mhalf %*% SigmaB %*% Mhalf)
dg$vectors <- P %*% diag(1/sqrt(dgM$values)) %*% t(P) %*% dg$vectors
C <- Xbar %*% M %*% dg$vectors
Cbar <- Gbar %*% M %*% dg$vectors
```

Finally, we can measure the quality of the representation and plot the MDA (recall that $\kappa = 2$ here),

```
dg$values[1]

## [1] 0.94675

cumsum(dg$values)/sum(dg$values)

## [1] 0.5434695 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000

text(C[,1:2],labels=data$V8,col=species)
points(Cbar[,1:2],pch=15)
```

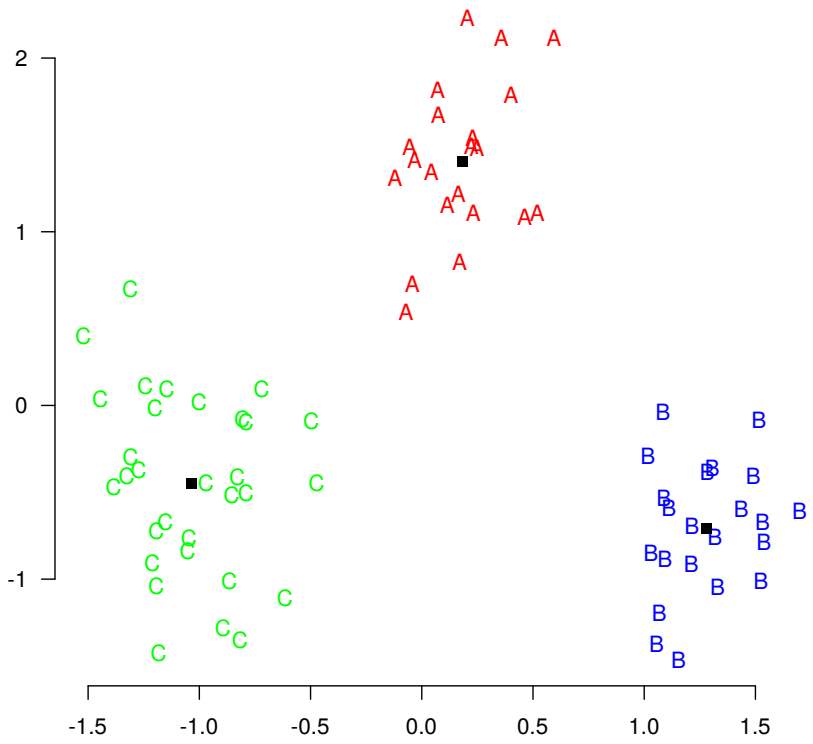

Figure 2.3: Multiple discriminant analysis of "Lubischew data"

2.1.4 Notes about MDA

Other normalizations are used. A common consists in getting the eigenvectors of $\Sigma_b\Sigma_w^{-1}$. It doesn't modify the interpretation of MDA but be careful when you use a software.

To predict the label of a new individual, we use the graphical representation. This tool is useful because it is easy to handle but it is also known to be too optimistic (penalization procedure is needed to get a trade-off with the number of groups).

Other example : Fisher iris data

To illustrate MDA (Multiple Discriminant Analysis), we handle the well known iris data set of Fisher. This set is about $n = 150$ iris and, for each one, we measure the sepal length and width and the petal length and width ($p = 4$ variables). Moreover, for each iris, we know its species ("Setosa", "Versicolor" and "Virginica") that plays the role of label information. We can begin by look at the result of a PCA procedure for this data set with uniform weights and standard Euclidean metric.

```
X <- as.matrix(iris[,1:4])
Xbar <- scale(X,scale=F)
Sigma <- t(Xbar) %*% Xbar / nrow(Xbar)
dg <- eigen(Sigma)
C <- Xbar %*% dg$vectors
cos2 <- rowSums(C[,1:2]^2)/rowSums(C^2)
plot(C[,1:2],pch=2,cex=cos2,xlab="",ylab="")
```

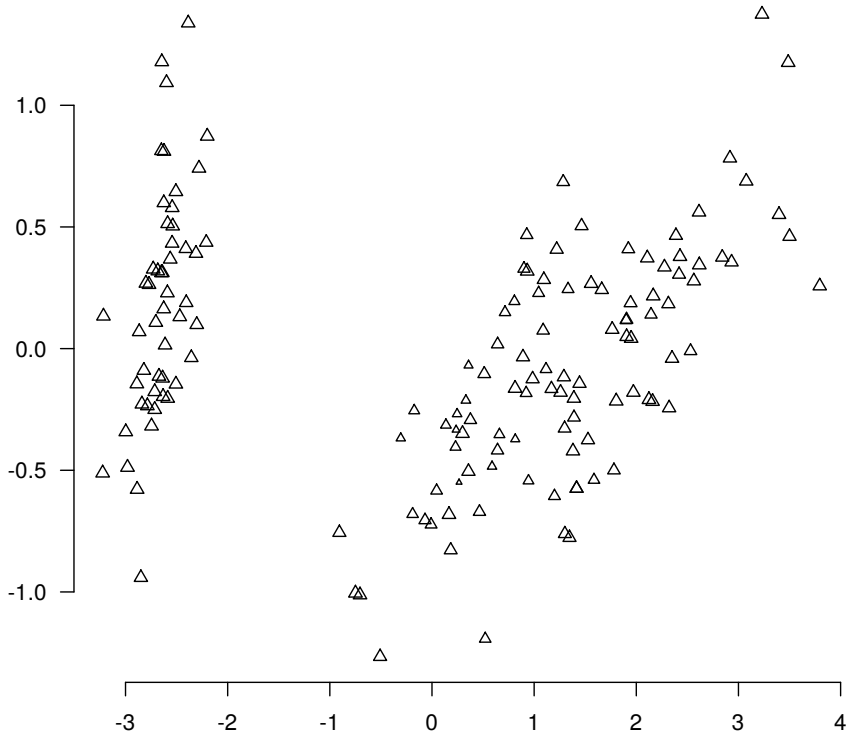

Figure 2.4: Scatter plot of the "Iris data" in E_2

We clearly see that the PCA helps us to discriminate two groups but are they related to the species? Moreover, we know that we are dealing with three species. But we only discriminate two groups. Let us make the species appear on this scatter plot.

```
plot(C[,1:2],pch=2,cex=cos2,xlab="",ylab="",col=species)
```

This time, information is clearer and we could use it to construct our supervised learning procedure. The remaining problem is how to choose a good frontier between the blue and the

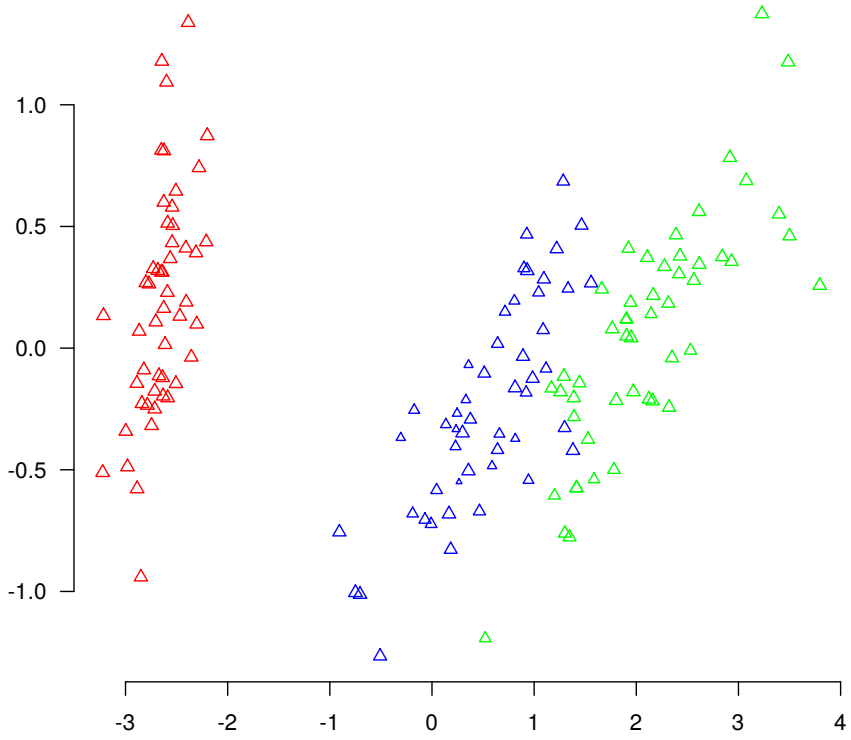

Figure 2.5: Scatter plot of the "Iris data" in E_2 with species

green set in Figure 2.5? Indeed, it could be easier to handle and more precise if the groups were organised in a clearer way. This is the aim of MDA procedure.

We compute the usefull matrix that we need to deal with MDA.

```
W <- diag(rep(1/n,n))
Wbar <- t(T) %*% W %*% T
WbarInv <- diag(1/diag(Wbar))
G <- WbarInv %*% t(T) %*% W %*% X
g <- colMeans(X)
Gbar <- G - rep(1,3) %*% t(g)
XB <- T %*% G
XBbar <- XB - rep(1,n) %*% t(g)
SigmaB <- t(Gbar) %*% Wbar %*% Gbar
```

Thus, we compute the eigenvectors as we did for PCA,

```
M <- solve(Sigma)
dgM <- eigen(M)
P <- dgM$vectors
Mhalf <- P %*% diag(sqrt(dgM$values)) %*% t(P)
dg <- eigen(Mhalf %*% SigmaB %*% Mhalf)
dg$vectors <- P %*% diag(1/sqrt(dgM$values)) %*% t(P) %*% dg$vectors
C <- Xbar %*% M %*% dg$vectors
Cbar <- Gbar %*% M %*% dg$vectors
```

Finally, we can measure the quality of the representation and plot the MDA,

```
dg$values[1]

## [1] 0.9698722

cumsum(dg$values)/sum(dg$values)

## [1] 0.8137202 1.0000000 1.0000000 1.0000000

points(Cbar[,1:2],pch=15)
```

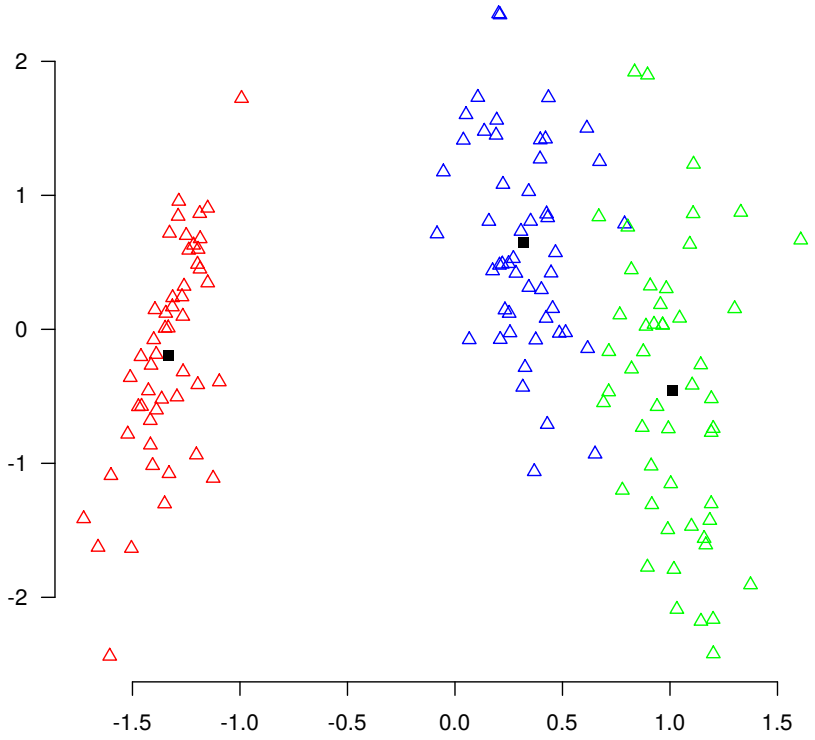

Figure 2.6: Multiple discriminant analysis of "Iris data"

2.2 CART

As above, we consider labeled data given by n observations of $p+1$ variables: p real variables $x^1, \ldots, x^p$ and 1 categorical variable t with values in $\{\tau_1, \ldots, \tau_m\}$. Moreover, to each individual $i \in \{1, \ldots, n\}$, we associate a weight w_i.

2.2.1 Introduction

From a general point of view, a **decision tree** is a tool used to provide a predictive model that makes correspond individuals to labels. There exist two classical such a procedure, known as **classification tree** and **regression tree**. *Classification And Regression Tree* (CART) is a generic name that cover these two approaches. In this section, we are going to deal with classification trees but there are few differences for dealing with regression trees (see the literature).

Let us describe by explaining how a tree can be built in order to represent a classification rule on an example. We consider the case of $p = 2$ with x^1 and x^2 both with values in $[0, 1]$. To keep the topic simple, we restrict ourself to cases given by partitions of $[0, 1]^2$ with blocks paralell to the axes (see Figure 2.7). We have at our disposal 5 regions $R_1, \ldots, R_5$. It is clear

that such a partition always can be described by binary paritions (*i.e.* test if $x^1 < 0.2$, then, if yes test if $x_2 < 0.6$, ...). Such sequence of tests can easily been represented by a tree as the one presented in Figure 2.8. Note that there is not unicity of the tree that represents the partition. An important advantage of the recursive binary tree is its interpretability. Indeed, the space or observations, as complicated as it is, can be fully represented by a tree. Of course, it is easy to see that such trees can be used for more than only two variables. It is harder to give a graphical representation of the partitioning of the space but the idea is the same. We divide it by testing for a condition and we recursivly continue till we reach some stopping rule.

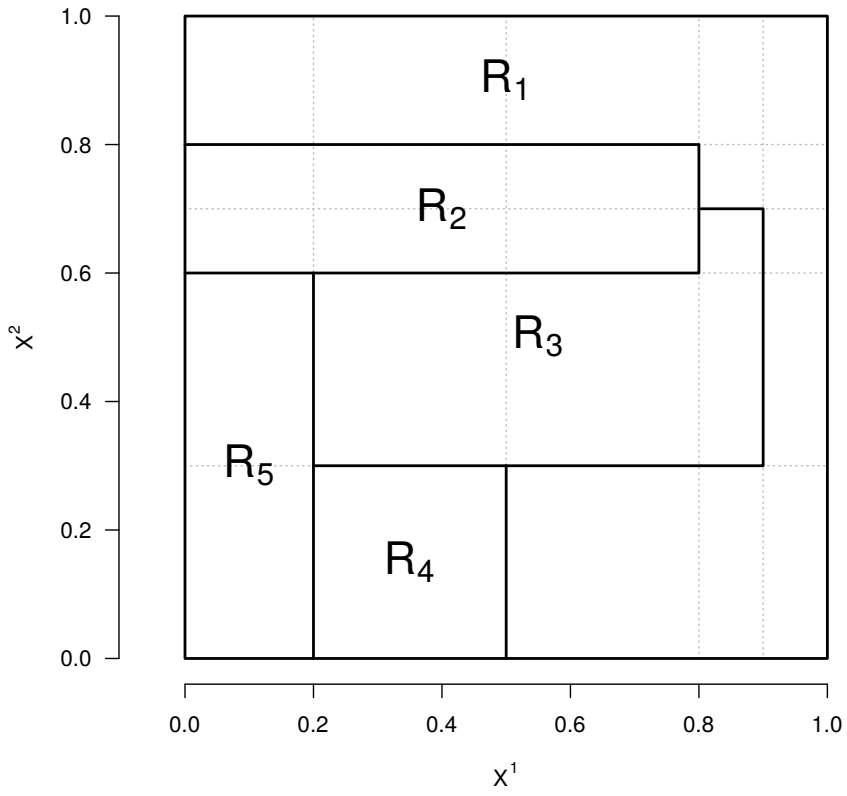

Figure 2.7: Example of binary partition of $[0, 1]^2$

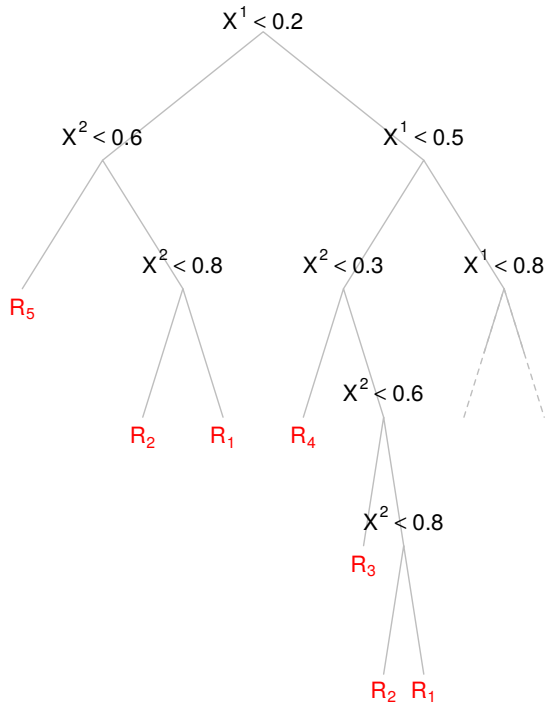

Figure 2.8: Example of binary tree representing the partition

Spam data

In the sequel of this section, we handle a data set relative to some measures on $n = 4601$ mails labeled as "mail" or "spam". This data set was first considered for Hewlett-Packard technical report and is now famous as a training/testing set. The aim is to determine if a mail is spam or not. To do this, the data set contains $p = 57$ real variables $x^1, \ldots, x^{57}$ and one qualitative variable (0 for mail and 1 for spam). The data set contains 1813 spams (*i.e.* 39.4% of spams). The variables measures various things as the frequencies of some particular words ("make", "address", "george", ...), the frequencies of some characters (';', '$', '#', ...), lengths of uninterrupted sequences of capital letters, ...

```
 # Mail n°1

## [1] "spam"
```

```
##     WFmake  WFaddress      WFall       WF3d      WFour     WFover
##       0.00       0.64       0.64       0.00       0.32       0.00
##   WFremove WFinternet    WForder     WFmail
```

```
##       0.00       0.00       0.00       0.00
## [ ... ]
##        WFedu      WFtable WFconference          CF;          CF(
##        0.000        0.000        0.000        0.000        0.000
##          CF[          CF!          CF$          CF#       CAPave
##        0.000        0.778        0.000        0.000        3.756
##       CAPlon       CAPtot
##       61.000      278.000
```

```
 # Mail n°1900

## [1] "mail"
```

```
##     WFmake  WFaddress      WFall       WF3d      WFour     WFover
##          0          0          0          0          0          0
##   WFremove WFinternet    WForder     WFmail
##          0          0          0          0
## [ ... ]
##        WFedu      WFtable WFconference          CF;          CF(
##         7.14         0.00         0.00         0.00         0.00
##          CF[          CF!          CF$          CF#       CAPave
##         0.00         0.00         0.00         0.00         5.50
##       CAPlon       CAPtot
##        10.00        11.00
```

2.2.2 Procedure

Our aim is to construct a classification procedure for the variable t, *i.e.* we want to get a tree based rule that lead to a good partition (in the sense of t) of the data solely based on the observations $x_i = (x_i^1, \ldots, x_i^p)' \in \mathbb{R}^p$. The question is how to grow such a classification tree, *i.e.* how can we find the splitting variables and the split points? To illustrate the difficulty of this question, let us assume that we know a partition of $\mathbb{R}^p$ into M regions $R_1, \ldots, R_M$ such that the variable t is given by the $\mathbb{R}^p$-valued vector of variables $x = (x^1, \ldots, x^p)'$,

$$t = t(x) = \tau_{\lambda(k)} \text{ if } x \in R_k \ ,$$

where the function $\lambda : \{1, \ldots, M\} \to \{1, \ldots, m\}$ is simply the function that associate one of the labels to each region. Of course, if we know the regions $R_1, \ldots, R_M$, the supervised learning problem amounts to get this function λ. This is just a way for rewriting the problem.

A basic way to estimate λ in such a case and, then, to build the classification procedure, is to consider the index sets, for any $k \in \{1, \ldots, M\}$,

$$\mathcal{R}_k = \{i \in \{1, \ldots, n\} \text{ s.t. } x_i \in R_k\} \ ,$$

and the weight of $\mathcal{R}_k$,

$$\bar{w}_k = \sum_{i \in \mathcal{R}_k} w_i \ .$$

Then, we can compute the frequencies of each value τ_ℓ of the elements of $\mathcal{R}_k$,

$$\hat{p}_{k\ell} = \frac{1}{\bar{w}_k} \sum_{i \in \mathcal{R}_k} w_i \mathbb{1}_{t_i = \tau_\ell}, \ \ell \in \{1, \ldots, m\} \ ,$$

and attribute to R_k the most represented element,

$$\hat{\lambda}(k) = \underset{\ell \in \{1,\ldots,m\}}{\operatorname{argmax}} \ \hat{p}_{k\ell} \ .$$

Because of the categorical nature of the variable t, we can not use a least-square criterion to measure the "quality" of the estimated vector of frequencies $\hat{p}_k = (\hat{p}_{k1}, \ldots, \hat{p}_{km})'$. Such a quantity is known as a measure of **node impurity**. Many impurities have been considered in the literature. For classification trees, a very common choice is the **Gini index**,

$$\mathcal{G}_k = \sum_{\ell=1}^{m} \hat{p}_{k\ell}(1 - \hat{p}_{k\ell}) \ .$$

This index is related to the error rate obtained by choosing the label of R_k at random with the probabilites given by $\hat{p}_k$.

Of course, in practice, we have no idea about what is a "good" partition $R_1, \ldots, R_M$ and, in our framework, we have to find the best (in a sense to precise later) binary partition. Generally, computing such a partition is infeasible (we can prove that this is a NP-complete problem). Hence, we consider the following **greedy algorithm** (*i.e.* a step by step algorithm that aims to solve a general problem by only solving it at each step, hoping that the constructed object is an acceptable solution for the global problem).

Let us describe the first step of this algorithm. For any $j \in \{1, \ldots, p\}$ and $s \in \mathbb{R}$, we define the pair of half-planes given by splitting the variable x^j at s,

$$R_1(j, s) = \left\{x = (x^1, \ldots, x^p)' \in \mathbb{R}^p \text{ s.t. } x^j \leqslant s\right\}$$

and

$$R_2(j, s) = \left\{x = (x^1, \ldots, x^p)' \in \mathbb{R}^p \text{ s.t. } x^j > s\right\} \ .$$

These two sets give us a trivial partition of $\mathbb{R}^p$ and we can use them to proceed as above to compute the frequencies $\hat{p}_{1\ell}(j, s)$ and $\hat{p}_{2\ell}(j, s)$, $\ell \in \{1, \ldots, m\}$, of the τ_ℓ's in $R_1(j, s)$ and $R_2(j, s)$, respectively. Moreover, we can associate a label

$$\tau_1(j, s) = \tau_{\lambda(1)}(j, s) \text{ and } \tau_2(j, s) = \tau_{\lambda(2)}(j, s)$$

to each region $R_1(j, s)$ and $R_2(j, s)$ by taking the most represented value. In order to choose good index $j \in \{1, \ldots, p\}$ and split point $s \in \mathbb{R}$, we take the values that make the Gini index minimal. Precisely, we have at our disposal $\mathcal{G}_1(j, s)$ and $\mathcal{G}_2(j, s)$ and the region weights $\bar{w}_1 = \bar{w}_1(j, s)$ and $\bar{w}_2 = \bar{w}_2(j, s)$. Thus, we take $j^* \in \{1, \ldots, p\}$ and $s^* \in \mathbb{R}$ which reach the minimal value

$$\min_{j,s} \left\{\bar{w}_1 \mathcal{G}_1(j, s) + \bar{w}_2 \mathcal{G}_2(j, s)\right\} \ .$$

The sequel of the algorithm then consists in repeating the same procedure in each region. In such a way, we divide the considered region by two at each step.

Spam data

To deal with the spam data set, we use uniform weights. Thus, each mail is weighted by $1/n$ and each region is weighted by its cardinal divided by n. In Figures 2.9, 2.10 and 2.11, we show the CART for fixed depth equal to 1, 2 and 3 respectively.

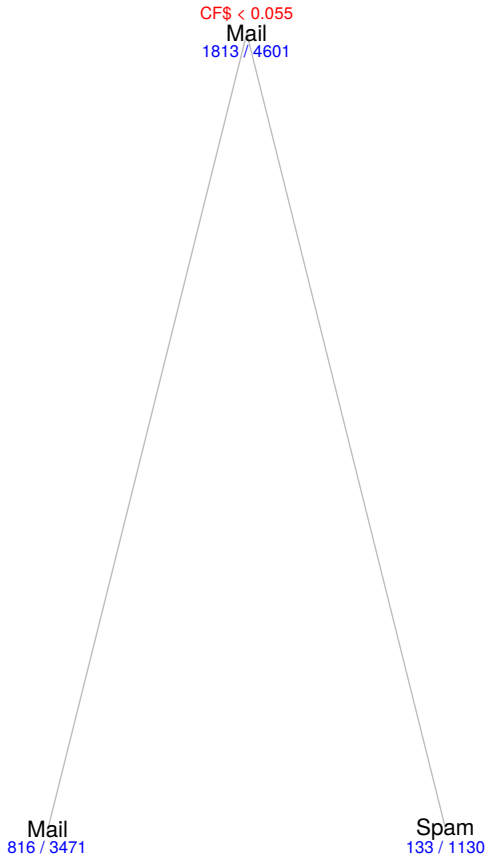

Figure 2.9: CART for Spam data with depth 1

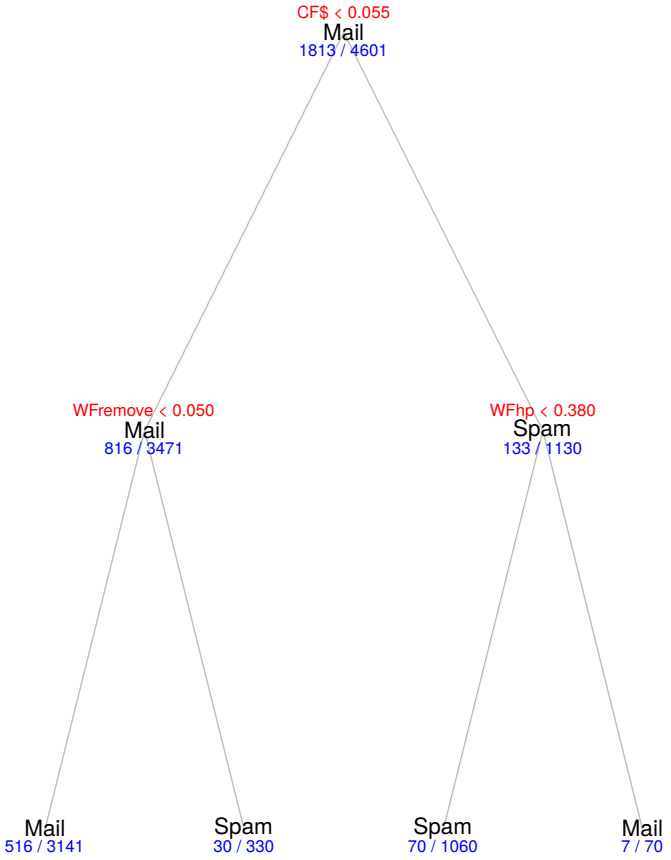

Figure 2.10: CART for Spam data with depth 2

2.2.3 Pruning the tree

The problem with our procedure is when should we stop it? In other terms, how large should we grow the tree? This question leads us to the important statistical dilemma: a large tree might overfit the data, while a small tree might not capture the important structure. We have to find some tradeoff between the capacity of the tree to represent a complicated classification rule and its complexity.

The idea of pruning is due to Breiman and can be done as follows. Let us consider a large tree T_0 constructed as above by stopping the process only when some minimum node size M_{T_0} is reached (*e.g.* level 5 reached with 2^5 terminal leaves, for instance). We now want to prune this large tree in order to only keep the important part. Let $T \subset T_0$ be any subtree of T_0, this tree leads to $1 \leqslant M_T \leqslant M_{T_0}$ regions $R_1, \ldots, R_{M_T}$ and, for each one, we have a weight $\bar{w}_k$, an estimated label $\hat{\lambda}(k)$ and a Gini index $\mathcal{G}_k(T)$. To get the announced trade-off, we consider the **penalized criterion**

$$\gamma_\alpha(T) = \sum_{k=1}^{M_T} \bar{w}_k \mathcal{G}_k(T) + \frac{\alpha M_T}{n}$$

for some $\alpha \geqslant 0$. The first part is relative to the goodness of the tree to fit to the data and

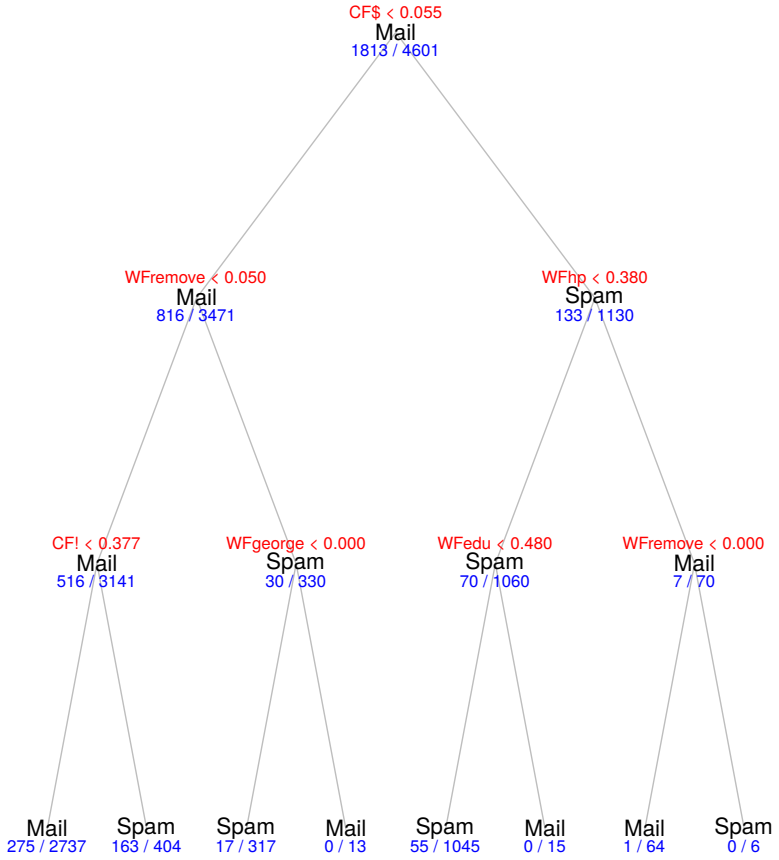

Figure 2.11: CART for Spam data with depth 3

the second term is a **penalty** that punishes the trees with too much leaves. For any $\alpha \geqslant 0$, we can choose $T_\alpha \subset T$ that minimizes $\gamma_\alpha(T)$. Of course, if $\alpha = 0$, we get $T_\alpha = T_0$ and, by convexity, if α grows towards infinity, T_α tends to be the trivial tree with only one (root) node. Finally, the problem amounts to adaptively choose α. This last point is generally treated by a cross-validation procedure and it is hard to get a theoritical justification of what is a good choice for α.

To help us in practice to compute T_α (because there are a lot of subtrees), we use **weakest link pruning**: we successively collapse the internal node that produces the smallest per-node increase of the mean of the Gini indices, and continue until we produce the single node tree. This gives a finite sequence of subtrees and one can show this sequence must contain T_α.

2.3 Perceptron (Practical session)

2.4 Bibliography

- *The Elements of Statistical Learning : Data Mining, Inference and Prediction*, J. Friedman, T. Hastie and R. Tibshirani (2009)

- *Classification and Regression Trees*, L. Breiman, J. Friedman, R. Olshen and C. Stone (1984)
- *Sélection de variables pour la discrimination en grande dimension et classification de données fonctionnelles (PhD. Thesis)*, C. Tuleau (2005)
- *Wiki Stat*, http://wikistat.fr/

Chapter 3

Clustering

3.1 Introduction

In contrast with the previous chapter, we are now interested in unlabeled data for which we want to create "good" groups. In other terms, we want to construct a partition of the observed individuals that leads to groups of similar individuals. Because we do not have access to any prior partition of the data, such a problem is called **clustering** (or **unsupervised learning**). Note that the number of partitions of $\{1, \ldots, n\}$ in k groups is given by

$$N_{n,k} = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n \ .$$

Then, the total number of partitions of $\{1, \ldots, n\}$ is $N_{n,0} + N_{n,1} + \cdots + N_{n,n}$. This number is huge, even for small values of n (*e.g.* for $n = 5$, there are 541 partitions), and this is not feasable to explore all such partitions (the problem is said to be NP-complete). Thus, we will have to construct the one that we are looking for.

In order to give a sense to the word "similar", we have to deal with a notion of distance or with the weaker concept of **similarity**. To be precise, if the observed data take their values in a space $\mathcal{X}$, we measure the dissimilarity between x and $x' \in \mathcal{X}$ with the aid of a function

$$d : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}_+$$

that is such that

1. $d(x, x') = d(x', x)$,
2. $d(x, x) = 0$.

Note that this definition is weaker than the one of a distance on $\mathcal{X}$. Indeed, if we want that δ is a distance, we need to impose some additional properties, namely

1. $d(x, x') = 0 \Longrightarrow x = x'$,
2. $d(x, x') \leqslant d(x, x'') + d(x'', x'),\ \forall x'' \in \mathcal{X}$.

Moreover, we say that d is an Euclidean distance if there exists some norm $\|\cdot\|$ over $\mathcal{X}$ such that $d(x, x') = \|x - x'\|$.

In the sequel, we assume that we do not directly handle observed data $x_1, \ldots, x_n \in \mathcal{X}$ but only the $n \times n$ matrix of their mutual distances,

$$X = \big[d(x_i, x_{i'})\big]_{1 \leqslant i,i' \leqslant n} \ .$$

Note that we did not assume anything about the space $\mathcal{X}$. We now discuss a bit about what kind of measure of dissimilarity we can handle in various basic situations.

Quantitative data

If all the observed variables $x^1, \ldots, x^p$ are quantitative, for instance in $\mathbb{R}$, we still have seen that we can define Euclidean distances by considering some positive symmetric matrix M. The distance is given by,

$$d_M(x, y) = \sqrt{{}^t(x-y)M(x-y)}, \ \forall x, y \in \mathbb{R}^p \ .$$

Particular choices are famous:

- $M = \mathrm{Id}_p$ gives the standard Euclidean distance,
- $M = \mathrm{diag}(1/\sigma^2(x^1), \ldots, 1/\sigma^2(x^p))$ gives the normalized distance,
- $M = \Sigma^{-1}$, with Σ the invertible variance matrix of the observations, gives the Mahalanobis distance.

More generally, we can deal with non-Euclidean distances (*e.g.* geodesic on some manifold to deal with GPS data).

Qualitative data

If all the variables $x^1, \ldots, x^p$ are qualitative, we can deal with a distance on the space of the line profiles. To introduce this generalized χ^2 distance, let us introduce some notations. For any $j \in \{1, \ldots, p\}$, the variable x^j takes its values in $\{1, \ldots, m_j\}$ and, for any $i_1, i_2 \in \{1, \ldots, n\}$, we can check if two individuals share the same value for x^j by considering

$$\delta_{j\ell}(i_1, i_2) = \begin{cases} 0 & \text{if } x^j_{i_1} \text{ and } x^j_{i_2} \text{ are both equal or unequal to } \ell \ , \\ 1 & \text{else} \ , \end{cases} \quad , \forall \ell \in \{1, \ldots, m_j\} \ .$$

Then, the distance is given by

$$d_{\chi^2}(i_1, i_2) = \frac{n}{p} \sum_{j=1}^{p} \sum_{\ell=1}^{m_j} \frac{\delta_{j\ell}(i_1, i_2)}{\sharp\{i \text{ s.t. } x^j_i = \ell\}} \ .$$

Binary data

We often deal with qualitative data that only can take two values (*e.g.* yes/no question, presence or no of something, ...). If all the variables are like that, we deal with p variables $x^1, \ldots, x^p$ with values in $\{0, 1\}$ and each individual is represented by some vector $x_i \in \{0, 1\}^n$, $i \in \{1, \ldots, n\}$. In such a particular case, we can consider the following quantities defined for any pair $i, i' \in \{1, \ldots, n\}$,

- $A_{i,i'}$ number of common 1 in x_i and $x_{i'}$,
- $B_{i,i'}$ number of digits equal to 0 in x_i and to 1 in $x_{i'}$,
- $C_{i,i'}$ number of digits equal to 1 in x_i and to 0 in $x_{i'}$,
- $D_{i,i'}$ number of common 0 in x_i and $x_{i'}$.

Of course $A_{i,i'}+B_{i,i'}+C_{i,i'}+D_{i,i'}=p$. Then, we can define the **Hamming distance** between x_i and $x_{i'}$,

$$d_{Ham}(x_i,x_{i'})=B_{i,i'}+C_{i,i'}\ ,$$

the **Jaccard distance** between x_i and $x_{i'}$,

$$d_{Jac}(x_i,x_{i'})=\frac{B_{i,i'}+C_{i,i'}}{A_{i,i'}+B_{i,i'}+C_{i,i'}}\ ,$$

the **concordance distance** between x_i and $x_{i'}$,

$$d_{Con}(x_i,x_{i'})=\frac{B_{i,i'}+C_{i,i'}}{p}\ ,$$

or the **Dice distance** between x_i and $x_{i'}$,

$$d_{Dic}(x_i,x_{i'})=\frac{B_{i,i'}+C_{i,i'}}{2A_{i,i'}+B_{i,i'}+C_{i,i'}}\ .$$

Mixed data

Last but not least, we can have to deal with a mix of quantitative and qualitative data. In such a situation, we have to choose between:

- make all variable qualitative by slicing the quantitative variables to get groups of intervals (quantiles, ...),
- make all variable quantitative by doing a CA or a MCA and keeping some principal components as compressed data.

More generally, all the methods seen in the first chapter (PCA, CA, ...) can be handled as a preliminar step for the clustering.

Distance between cities

To illustrate the clustering procedures that we are going to study, we will deal with a data set that contains the distances between $n=47$ cities. Our aim, in the sequel, will be to use clustering to automaticaly detect geographical groups. Note that we do not handle position coordinates but only spatial distances between cities.

	Amiens	Andorre	Angers	Bâle	LaBaule	Besançon	Bordeaux	...
Amiens	0	1020	440	560	590	560	730	...
Andorre	1020	0	760	1130	830	970	430	...
Angers	440	760	0	770	160	620	340	...
Bâle	560	1130	770	0	940	160	840	...
LaBaule	590	830	160	940	0	770	400	...
Besançon	560	970	620	160	770	0	700	...
Bordeaux	730	430	340	840	400	700	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

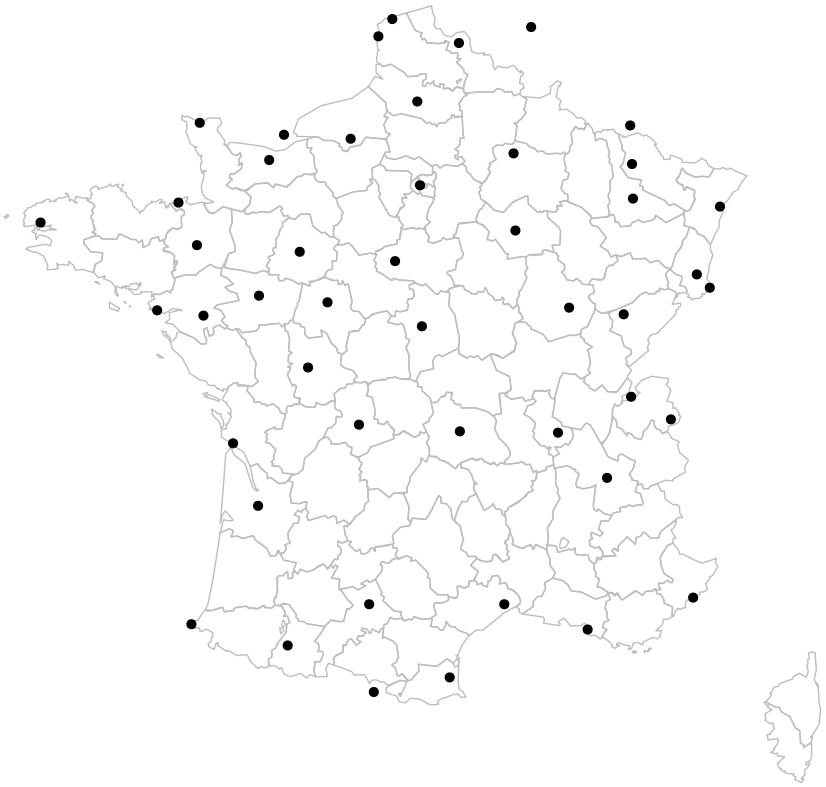

Figure 3.1: Map of cities

3.2 K-means and K-medoids

Let us assume that we have at our disposal the $n \times n$ matrix of the similarity measurements of our n individuals. The first approach of clustering is very simple and intuitive and it is known as K-means procedure (or K-medoids, see below). A drawback of such an approach is that we need to consider, a priori, some fixed number K of clusters. Base on the same idea as

MDA, the principle of this procedure is to provide a "good" partition of the data in the sense that the inertia within is minimal (*i.e.* each group contains similar elements). The procedure is iterative and a step of the algorithm can be describe in few words: each individual is put in the closest group.

3.2.1 K-means clustering

The K-means procedure is the oldest one and is not only based on X but also on the coordinates of the individuals. Indeed, we need to deal with points in $\mathbb{R}^p$, for instance, or with data that allow us to compute centers of gravity. Each cluster is given by a **centroid** c_k and the associated individuals.

1. Pick K individuals at random in the data set. These points are the initial centroids of the K clusters.

2. For each individual, put it in the cluster of the closest centroid.

3. Compute the centers of gravity of each cluster. These centers of gravity become the new centroids.

4. Compute the inertia within criterion,

$$\sum_{k=1}^{K} \frac{1}{\bar{w}_k} \sum_{i \in C_k} w_i d^2(x_i, c_k) \ .$$

   If it is smaller, go back to step 2, else stop.

This procedure converges to some local minimal. It should be sharper to repeat the operation some times in order to avoid bad clustering.

Variant 1 (Faster)

We can update the center of gravity of a cluster (thus, its centroid) each time we add or remove an individual. This variant is faster to converge towards a local minimum. In particular, it means that it is faster but it is more liable to fall in a local minimum that is far from a global minimum.

Variant 2 (Nuées dynamiques)

In order to only deal with the matrix X, we can avoid the computation of the centers of gravity. The centroids are taken equal to some element of the cluster that minimizes the inertia within criterion among the elements of the cluster,

$$\sum_{k=1}^{K} \frac{1}{\bar{w}_k} \sum_{i \in C_k} w_i X^2_{ic_k} \ .$$

Note that, in this variant, the stopping time is reached when the the centroids do not change.

Cities data set

Here are the initial settings for the K-means procedure that we use for the three variants (results in Figures 3.2, 3.3 and 3.4).

```
K <- 5
VilleNom[start]

## [1] "Nice"    "Rouen"   "Orléans" "Reims"   "LeHavre"
```

```
step # K-means (no variant)

## [1] 6
```

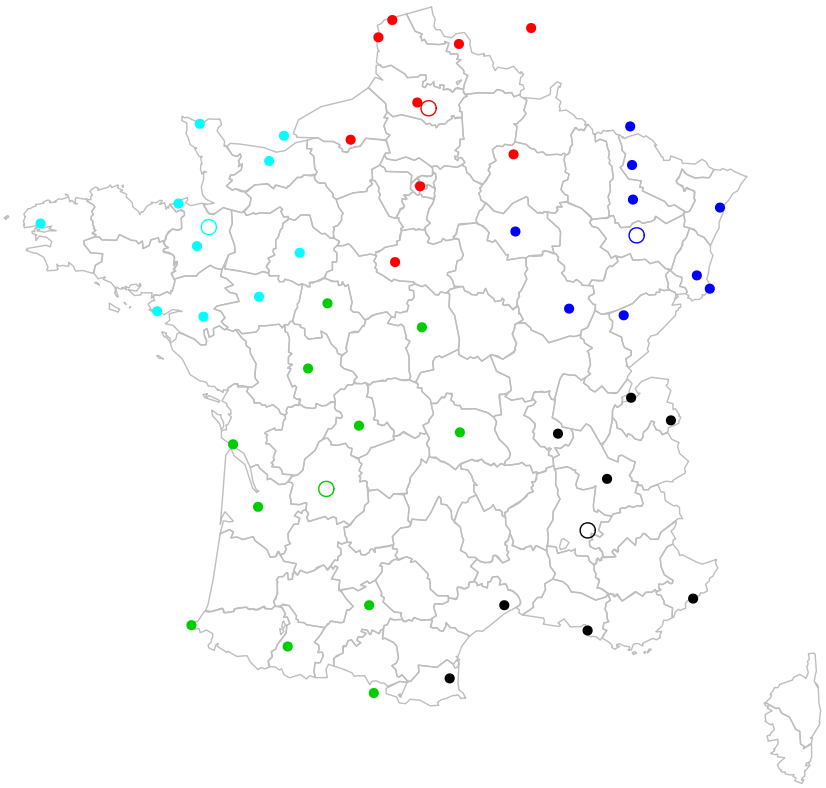

Figure 3.2: K-means clustering (no variant)

```
step # K-means (variant 1)

## [1] 3
```

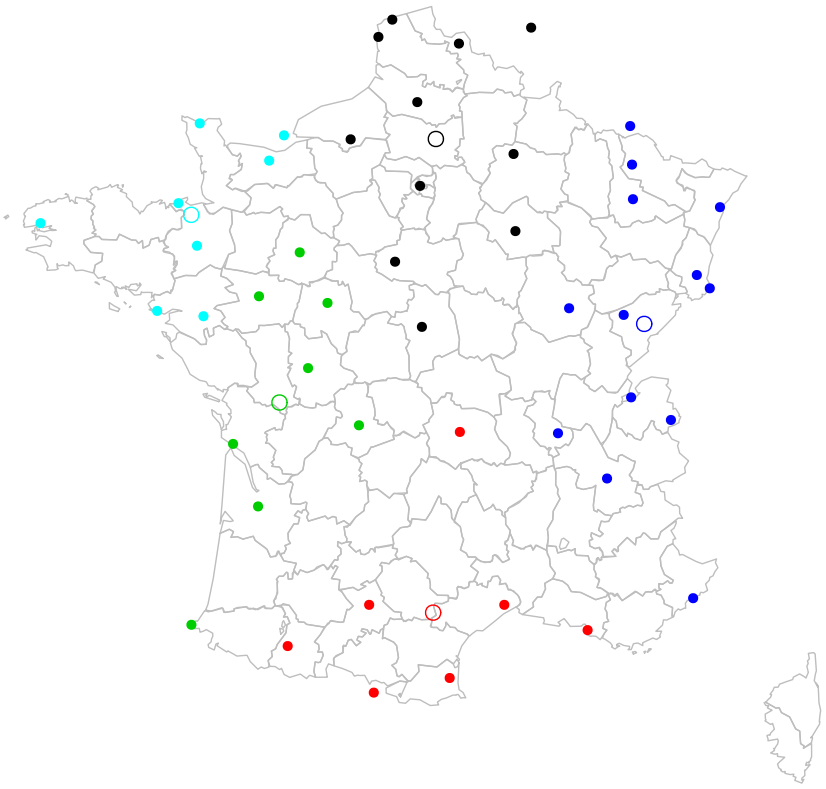

Figure 3.3: K-means clustering (variant 1)

```
step # K-means (variant 2)

## [1] 4
```

3.2.2 K-medoids clustering

The procedure called K-medoids is very similar to the K-means but is based on medians with non-Euclidean distances rather than on the means of square Euclidean distance. The main advantage of this slight variant is that K-medoids are more robust to noise and outliers.

The basic algorithm of K-medoids is the same as the K-means procedure, only subsituting "means" by "medians" and deleting the square in the powers of distances. Note that the variant 2 is probably the most famous K-medoids procedure, known as **Partitioning Around**

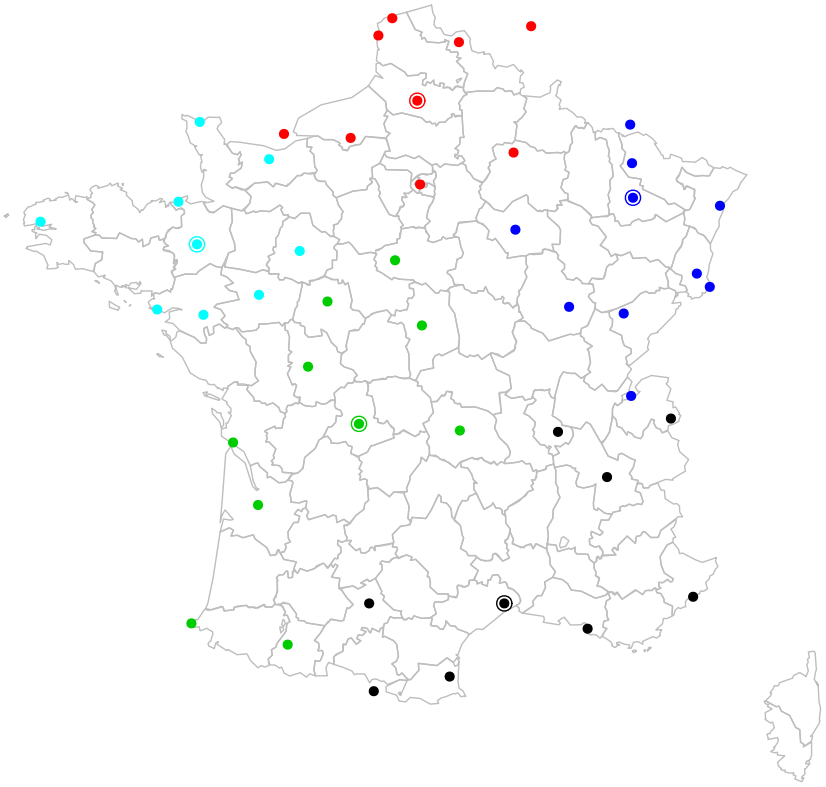

Figure 3.4: K-means clustering (variant 2)

Medoids (PAM). It is only based on the disimilarity matrix X and the main difference with K-means is that there is no square in the criterion.

3.3 Hierarchical clustering

3.3.1 Introduction

Hierarchical clustering is a clustering procedure based on aggregating groups of individuals that are closed. This procedure does not need to know the number of cluster. The idea is to start with all the individuals and to group the two that are the closest. It leads to a group of individuals and, if we know how to quantify the distance between groups, we can do it again till we acheive an only-one-group configuration. As for the CART procedure, usually, we represent the result by some binary tree, called **dendrogram**.

In order to apply such a procedure, we only need to define what we call the distance/disimilarity between two groups. We have to give sense to this distance only by using the weights of the individuals and the distance/disimilarity matrix X.

3.3.2 Distance between groups

There is no standard way for defining how far are two clusters. Among the most used procedure, we can list the following ones that make sense for any dissimilarity matrix X. Let A and B be two disjoint subset of $\{1, \dots, n\}$, we introduce:

- Single linkage:
$$d(A, B) = \min_{i \in A, j \in B} X_{ij} \ ,$$

- Complete linkage:
$$d(A, B) = \max_{i \in A, j \in B} X_{ij} \ ,$$

- Group average linkage:
$$d(A, B) = \frac{1}{\sharp A \times \sharp B} \sum_{i \in A, j \in B} X_{ij} \ .$$

If we are dealing with some Euclidean distance and if we can compute the centers of gravity (knowledge of the coordinates), we also can consider

- Centroid distance:
$$d(A, B) = d(g_A, g_B) \ ,$$

- Ward distance:
$$d(A, B) = \frac{w_A w_B}{w_A + w_B} d(g_A, g_B) \ .$$

The two most used definition are the group average linkage and the Ward distance. Note that the Ward distance is very natural because it corresponds to the loss of inertia between by joining A and B (*i.e.* maximization of inertia between by loosing the minimal quantity).

3.3.3 Procedure

The hierarchical clustering is an easy algorithm to apply:

1. Take the n singletons of individual and compute/get the distances.

2. Repeat till you only get one cluster:

   (a) group the two closest clusters in the sense of chosen group distance.

   (b) update the distance matrix by replacing the joined cluster by the new one and its distances to other clusters.

It leads us to the dendrogram and we now need to choose where we cut it to get our clustering.

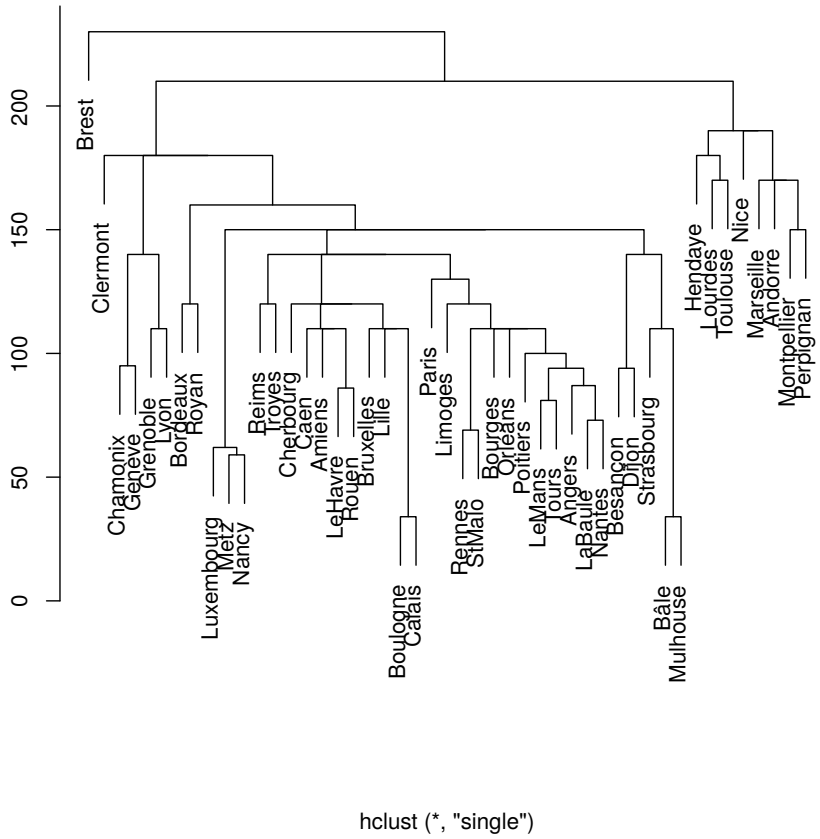

Figure 3.5: Dendrogram of hierarchical clustering with single linkage

Cities data set

To illustrate the important role played by the choice of the group distance, Figures 3.5, 3.6 and 3.7 represent the results obtained for the single linkage, average linkage and complete linkage, respectively.

Moreover, in order to be able to use the clustering in practice, we have to choose where we cut the tree. Figure 3.8 illustrate this point by presenting the dendrogram obtained with Ward distance and the groups that we get by taking 3, 4 and 9 clusters. What are the "good" choices?

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

3.3.4 Cutting the tree

As usual, there are various method for dealing with the question of how many clusters we should consider. This is again a tradeoff to find between the capacity to fit to the data and the complexity of the obtained clustering. A usual choice (see practical sessions for other

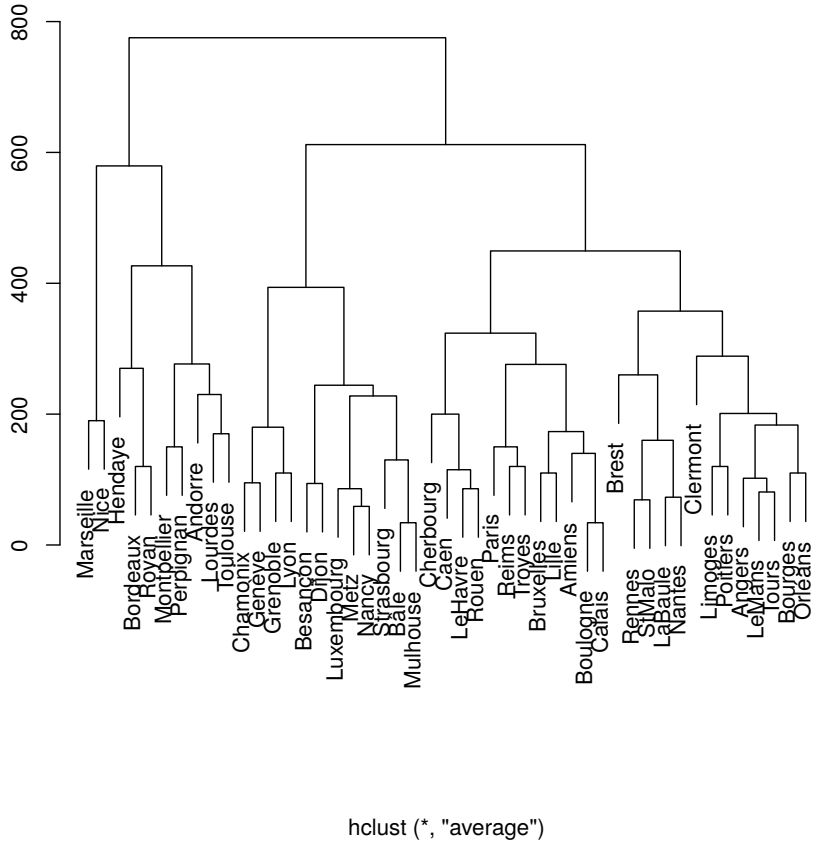

Figure 3.6: Dendrogram of hierarchical clustering with average linkage

ways) is to mimic the PCA by cutting the tree when the gain in term of variance between becomes small. This approch is mainly used for the hierarchcal clustering obtained with the Ward method because it is exactly the interpretation of the Ward distance. It can be done graphically by considering the fall of the clustering heights.

Cities data set

To illustrate the important role played by the choice of the group distance, Figures 3.5, 3.6 and 3.7 represent the results obtained for the single linkage, average linkage and complete linkage, respectively.

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

3.4 Bibliography

- *The Elements of Statistical Learning : Data Mining, Inference and Prediction*, J. Friedman, T. Hastie and R. Tibshirani (2009)

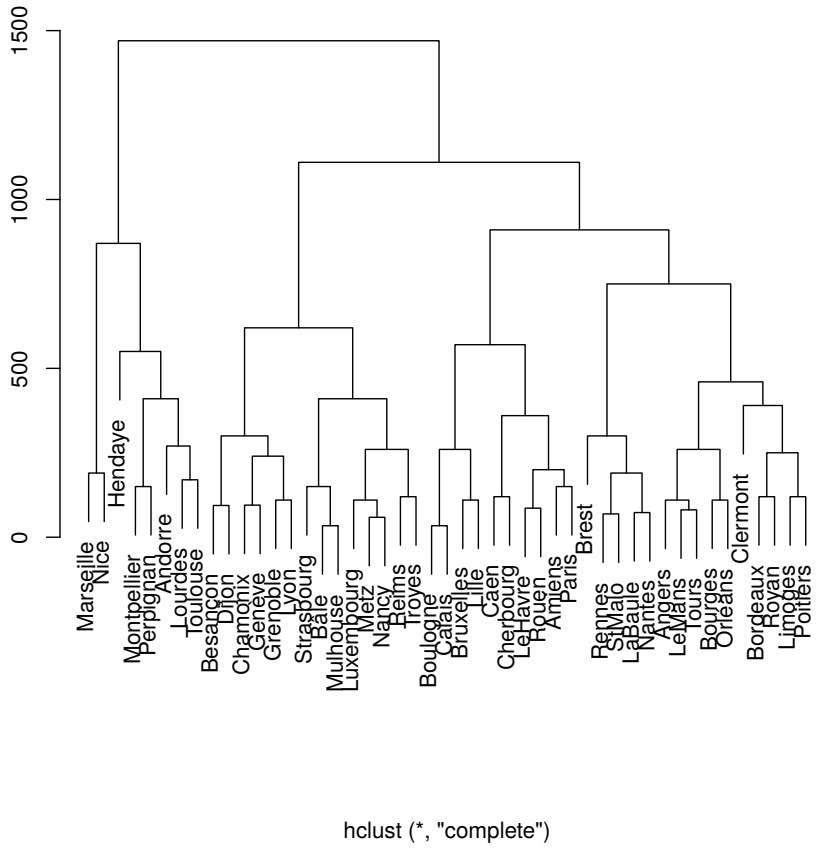

Figure 3.7: Dendrogram of hierarchical clustering with complete linkage

- *Wiki Stat*, http://wikistat.fr/
- *Quick-R*, http://www.statmethods.net/

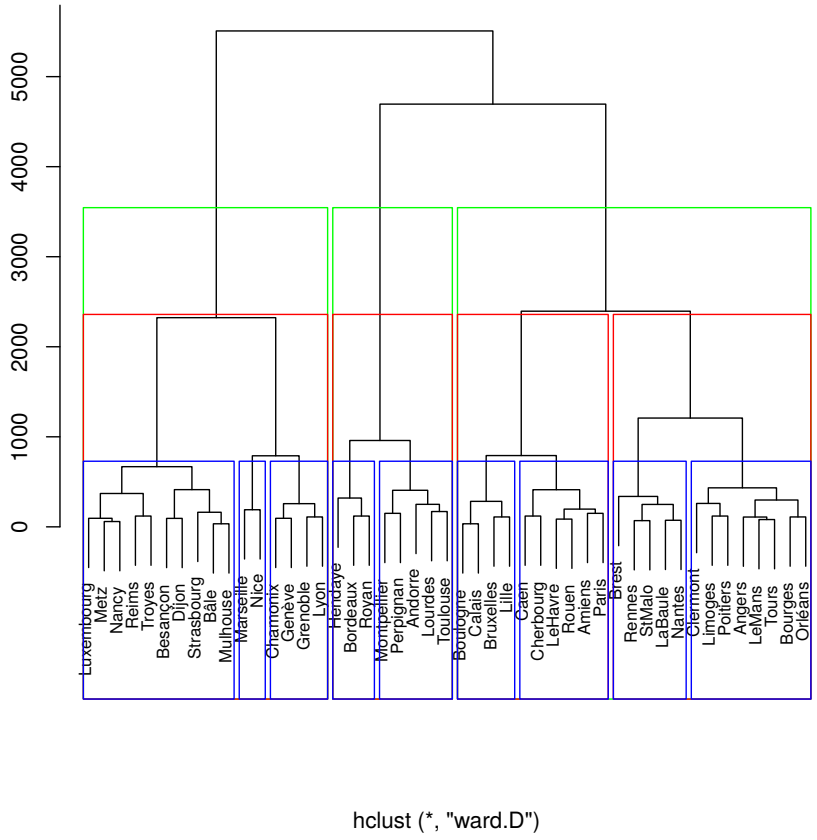

Figure 3.8: Dendrogram of hierarchical clustering with Ward linkage

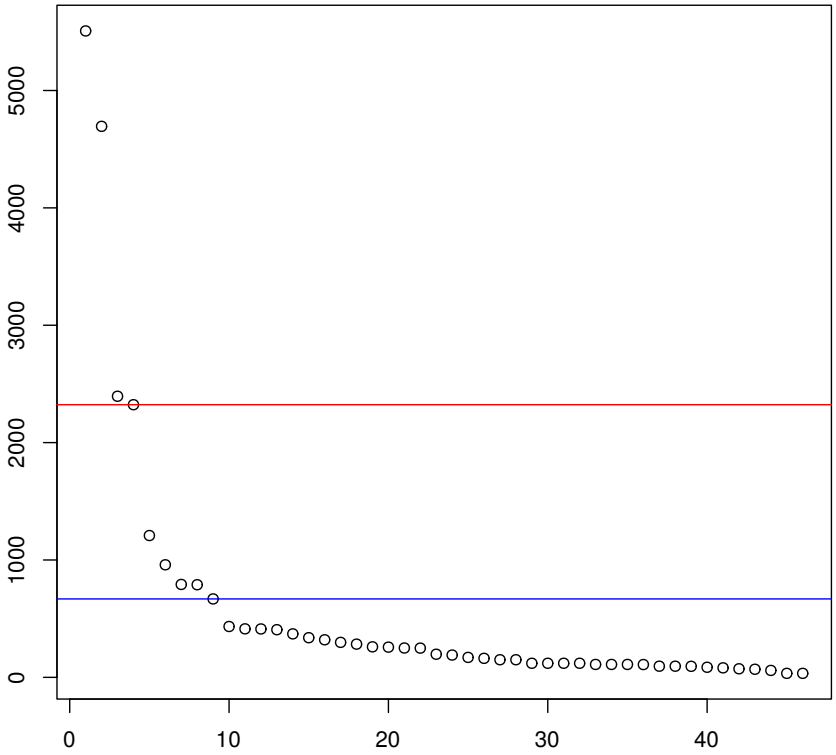

Figure 3.9: Fall of the clustering heights with Ward method

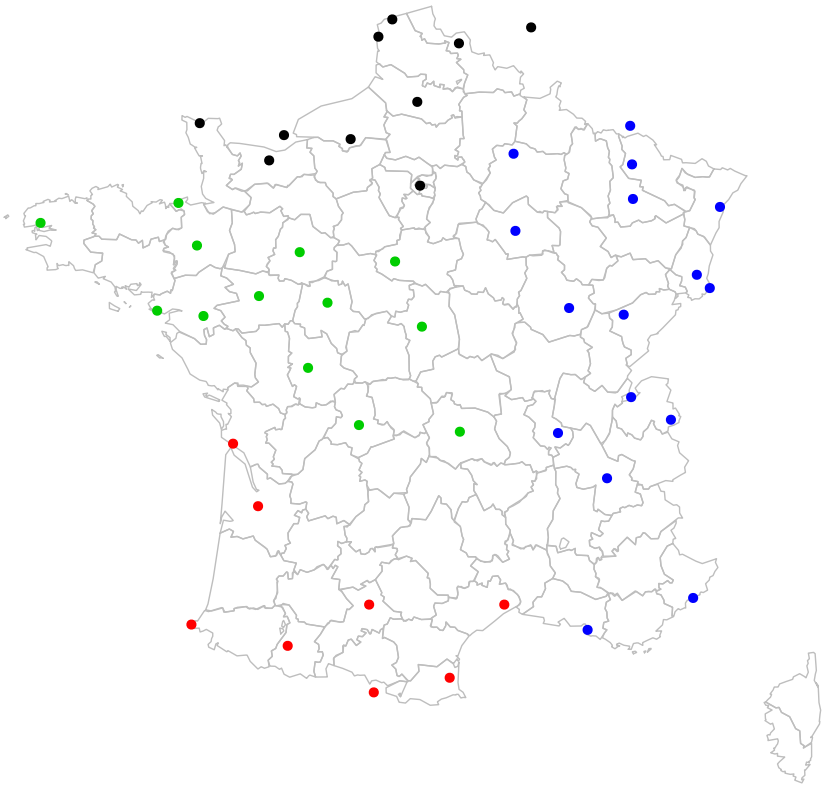

Figure 3.10: Hierarchical clustering with 4 clusters

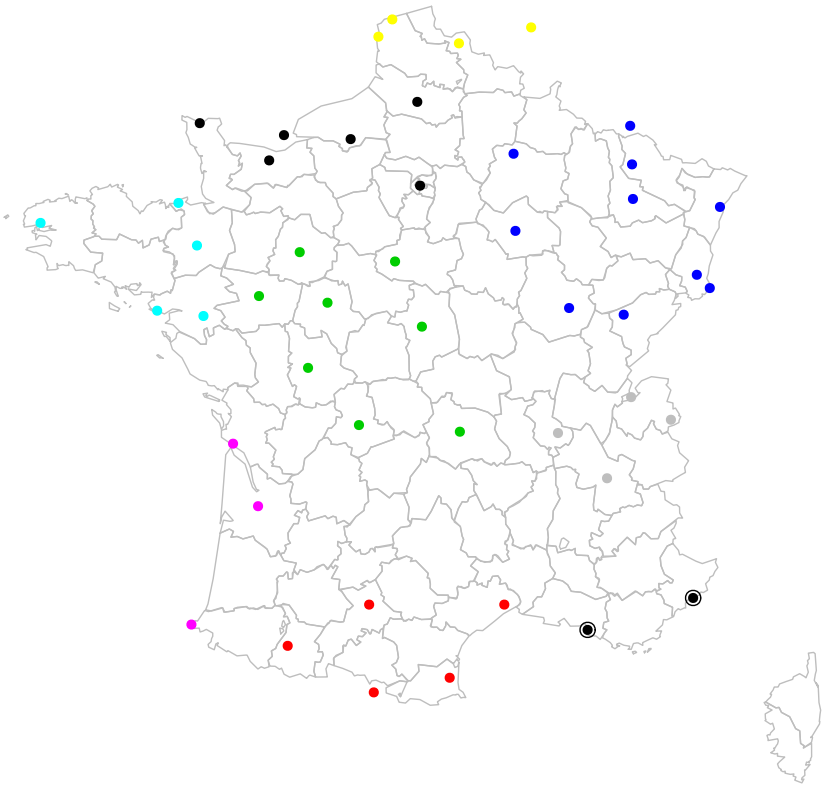

Figure 3.11: Hierarchical clustering with 9 clusters

Chapter 4

Model selection and calibration