Fiche TD avec le logiciel R : tdr610

Analyse en Composantes Principales - Base de l'analyse des données

A.B. Dufour

Présentation de l'ACP - niveau intermédiaire. Rappels des bases. Quelques exemples à analyser par soi-même.

Table des matières

1 Introduction

Les méthodes d'analyses multivariées sont une branche à part entière des statistiques. Nous présentons une introduction ou une ré-introduction, sommaire, à l'analyse en composantes principales (ACP). Après avoir repris les éléments théoriques principaux et les procédures informatiques permettant de réaliser une analyse, nous proposons quelques jeux de données typiques.
Cette fiche reprend des éléments du `tdr601.pdf` et du `tdr61.pdf`.

2 Les données

Afin de bien cerner les différentes étapes de la méthode, nous avons choisi de présenter un jeu de données réduit.

```
notesBA <- read.table("http://pbil.univ-lyon1.fr/R/donnees/notesBA.txt",h=T,dec=",")
notesBA
```

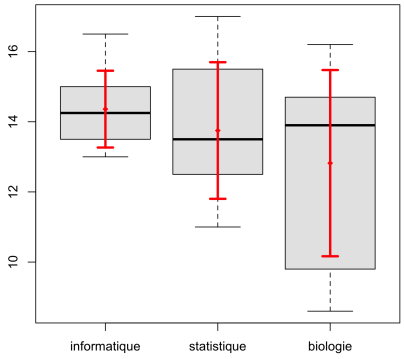
```
   informatique statistique biologie option
1          13.5        15.5     14.5      3
2          14.0        11.5     14.1      2
3          15.3        14.0     13.7      1
4          14.0        13.0     15.0      2
5          13.0        11.0      9.5      3
6          15.0        13.0     12.1      2
7          14.5        16.0      9.8      1
8          16.5        17.0     16.2      1
9          14.8        14.0     14.7      2
10         13.0        12.5      8.6      1
```

Les trois variables d'intérêt sont les notes obtenues en informatique, en statistique et en biologie par 10 étudiants du M2 "Biométrie Appliquée".
La note de biologie correspond à l'une des spécialités suivantes : (1) épidémiologie, (2) gènes et protéomique, (3) gestion des populations et des habitats. Le choix est réalisé en fonction des objectifs professionnels de chaque étudiant. Il est noté dans la quatrième variable `option`.

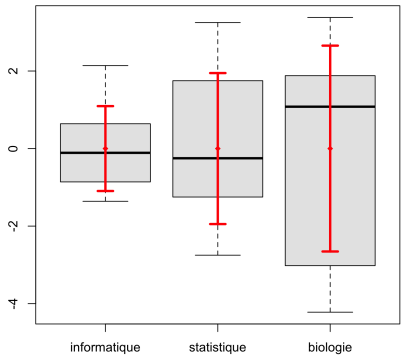
```
notes <- notesBA[,1:3]
options <- factor(notesBA$option)
```

Le data frame `notes` correspond, d'un point de vue mathématique, à une matrice à $n = 10$ lignes et $p = 3$ colonnes que l'on note $\mathbf{X}$.
Avant de se pencher sur l'analyse en composantes principales, il est toujours indispensable de regarder les données. Pour ce faire, on propose une représentation qui cumule deux types d'information : la boîte à moustaches (basée sur les quartiles) et la dispersion des individus autour de la moyenne ($\pm$ un écart-type).

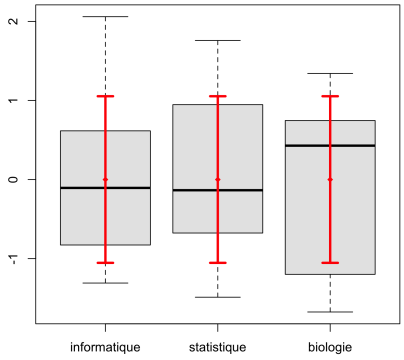
```
moyt <- sapply(notes, mean)
sdt <- sapply(notes, sd)
liminf <- min(moyt-sdt)
limsup <- max(moyt+sdt)
rb <- boxplot(notes, col=grey(0.9))
axex <- seq(rb$n)
points(axex, moyt, col = "red", pch = 18)
arrows(axex, moyt - sdt, axex, moyt + sdt, code = 3, col = "red", angle = 90, length = .1, lwd=3)
```



On représente le même graphique après avoir centré les variables,

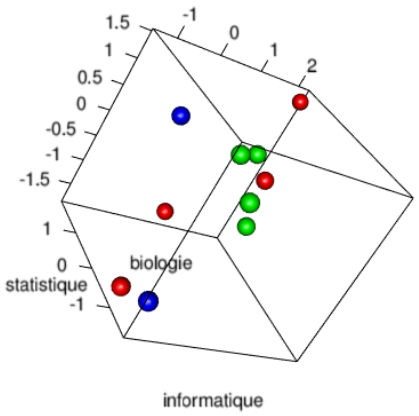


et après les avoir centrées réduites ...



Conclusion. Bien réfléchir au type de transformation que l'on souhaite avant de réaliser une analyse. Le tableau des données brutes est noté $\mathbf{X}$, le tableau des données centrées $\mathbf{X}_0$, le tableau des données centrées réduites $\mathbf{X}_*$.
Un des objectifs de l'analyse en composantes principales, si on se place dans l'espace des trois variables, est de rechercher une combinaison linéaire qui maximise la variance du nuage de points.

```
library(rgl)
plot3d(notescr,type="s",col=rainbow(3)[notesBA$option])
```

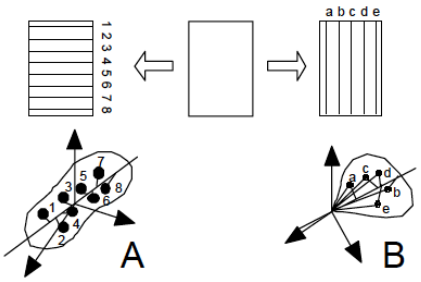


Faites tourner avec le curseur de la souris cette représentation pour en avoir différents points de vue

3 Analyse en composantes principales

3.1 Synthèse théorique

Analyser un tableau en soi ne veut rien dire. On étudie les lignes d'un tableau, les colonnes d'un tableau et on cherche parfois à relier les lignes et les colonnes.



A Point de vue des individus. Chacun des 10 individus est caractérisé par trois variables. On dit alors que l'on a $n = 10$ points dans l'espace $\mathbb{R}^3$.

B Point de vue des variables. Chacune des 3 variables est définie par 10 points. On dit alors que l'on a $p = 3$ points dans l'espace $\mathbb{R}^{10}$.

Les différentes étapes de l'analyse sont résumées ci-dessous.

Tableau à analyser

$$\mathbf{X}_*$$

n points dans $\mathbb{R}^p$	p points dans $\mathbb{R}^n$
$\mathbf{R} = \frac{1}{n}\mathbf{X}_*^T\mathbf{X}_*$	$\mathbf{S} = \frac{1}{n}\mathbf{X}_*\mathbf{X}_*^T$

Diagonalisation

Valeurs et Vecteurs propres

$\mathbf{\Lambda}_p$	$\mathbf{\Lambda}_n$
$\mathbf{U}$	$\mathbf{V}$
(axes principaux)	(composantes principales)

Propriété

Mêmes valeurs propres non nulles
On diagonalise dans l'espace de dimension le plus petit.
On note $\mathbf{\Lambda}$ la matrice diagonale contenant les valeurs propres non nulles.

Coordonnées des individus	Coordonnées des variables
$\mathbf{L} = \mathbf{X}_*\mathbf{U}$	$\mathbf{C} = \frac{1}{\sqrt{n}}\mathbf{X}_*^T\mathbf{V}$
$\Downarrow$	$\Downarrow$
Carte Factorielle	Carte Factorielle ou Cercle des corrélations

Propriété

Les coordonnées des individus $\mathbf{L}$ sont les vecteurs propres de $\mathbf{S}$ normés à $\mathbf{\Lambda}$.
On veut les vecteurs propres $\mathbf{V}$ normés à 1. Ce qui conduit à la relation :

$$\mathbf{V} = \mathbf{L}\mathbf{\Lambda}^{-1/2}$$

3.2 Mise en oeuvre sous 

3.2.1 Les individus dans un espace de dimension p

```
matR <- t(notescr)%*%notescr
matR <- matR/10
matR
```

```
             informatique statistique  biologie
informatique    1.0000000   0.6333591 0.5953104
statistique     0.6333591   1.0000000 0.3923080
biologie        0.5953104   0.3923080 1.0000000
```

```
cor(notescr)
```

```
             informatique statistique  biologie
informatique    1.0000000   0.6333591 0.5953104
statistique     0.6333591   1.0000000 0.3923080
biologie        0.5953104   0.3923080 1.0000000
```

```
eigR <- eigen(matR)
eigR
```

```
$values
[1] 2.0870135 0.6089564 0.3040302
$vectors
           [,1]        [,2]       [,3]
[1,] -0.6244712 -0.04687728  0.7796398
[2,] -0.5602567 -0.66860908 -0.4889523
[3,] -0.5441950  0.74213504 -0.3912638
```

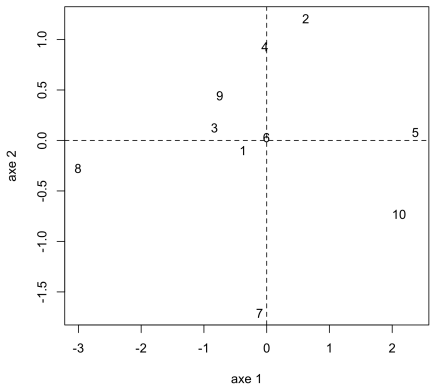
On calcule les coordonnées des individus.

```
matL <- notescr%*%eigR$vectors
matL
```

```
             [,1]        [,2]        [,3]
 [1,] -0.376641552 -0.09947283 -1.36988110
 [2,]  0.622266219  1.20783354  0.12639198
 [3,] -0.831252057  0.12642049  0.50282125
 [4,] -0.027151200  0.93015641 -0.41047029
 [5,]  2.369312776  0.07831385  0.22267670
 [6,] -0.001803939  0.03037862  0.79087100
 [7,] -0.113956373 -1.71068448 -0.02121459
 [8,] -3.002968963 -0.27681153  0.22125505
 [9,] -0.746680603  0.44369335 -0.02793240
[10,]  2.108875692 -0.72982741 -0.03451759
```

On représente les individus sur le premier plan factoriel (axes 1 et 2).

```
plot(matL[,1], matL[,2], type="n", xlab="axe 1", ylab="axe 2")
text(matL[,1], matL[,2],as.character(1:10))
abline(h=0, lty=2)
abline(v=0, lty=2)
```



3.2.2 Les variables dans un espace de dimension n

```
matS <- notescr%*%t(notescr)
matS <- matS/10
matS
```

```
             [,1]         [,2]         [,3]         [,4]         [,5]          [,6]
 [1,]  0.20283279 -0.052765993 -0.038829667  0.047999647 -0.120521235 -0.1085741648
 [2,] -0.05276599  0.186205204 -0.030101259  0.105469867  0.159707795  0.0135529542
 [3,] -0.03882967 -0.030101259  0.095979133 -0.006623287 -0.184762907  0.0403006752
 [4,]  0.04799965  0.105469867 -0.006623287  0.103441399 -0.008288773 -0.0296323202
 [5,] -0.12052123  0.159707795 -0.184762907 -0.008288773  0.566936100  0.0174213516
 [6,] -0.10857416  0.013552954  0.040300675 -0.029632320  0.017421352  0.0626403055
 [7,]  0.02421488 -0.213981464 -0.013220624 -0.157940212 -0.040869258 -0.0068540672
 [8,]  0.08554850 -0.217502353  0.257248122 -0.026676243 -0.708738256  0.0171992225
 [9,]  0.02753596  0.006774316  0.066272671  0.044444291 -0.174059245 -0.0007265169
[10,] -0.06744072  0.042640934 -0.186262857 -0.072194370  0.493174426 -0.0053274399
             [,7]        [,8]         [,9]        [,10]
 [1,]  0.024214881  0.08554850  0.0275359578 -0.06744072
 [2,] -0.213981464 -0.21750235  0.0067743158  0.04264093
 [3,] -0.013220624  0.25724812  0.0662726710 -0.18626286
 [4,] -0.157940212 -0.02667624  0.0444442906 -0.07219437
 [5,] -0.040869258 -0.70873826 -0.1740592453  0.49317443
 [6,] -0.006854067  0.01719922 -0.0007265169 -0.00532744
 [7,]  0.293987750  0.08110508 -0.0673337736  0.10089169
 [8,]  0.081105080  0.91434010  0.2113259057 -0.61385008
 [9,] -0.067333774  0.21132591  0.0755175929 -0.18975120
[10,]  0.100891687 -0.61385008 -0.1897511981  0.49811962
```

```
eigS <- eigen(matS)
eigS
```

```
$values
 [1]  2.087013e+00  6.089564e-01  3.040302e-01  3.585534e-16  1.764175e-16
 [6]  1.182563e-16  2.776152e-17  2.230967e-17  1.908873e-17 -2.092890e-17
$vectors
              [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
 [1,] -0.0824452433  0.04030987  0.78564172  0.46492882  0.00000000  0.00000000
 [2,]  0.1362114443 -0.48945639 -0.07248718  0.35035840 -0.66898464 -0.13070514
 [3,] -0.1819575605 -0.05123000 -0.28837346  0.63914374  0.38438210  0.46020406
 [4,] -0.0059432829 -0.37693190  0.23540918 -0.12073431  0.34014704  0.17176564
 [5,]  0.5186325486 -0.03173551 -0.12770751 -0.10206150 -0.15672355  0.65657517
 [6,] -0.0003948747 -0.01231048 -0.45357313  0.36363605  0.13867342 -0.40727541
 [7,] -0.0249445682  0.69322917  0.01216680  0.18401899 -0.33931061  0.18634326
 [8,] -0.6573372088  0.11217371 -0.12689218 -0.19288643 -0.17136850  0.12708294
 [9,] -0.1634452269 -0.17980006  0.01601954 -0.15369992  0.09803945  0.09595969
[10,]  0.4616239722  0.29575159  0.01979622  0.02586772  0.30183897 -0.29082400
             [,7]        [,8]        [,9]       [,10]
 [1,] -0.397726252 0.000000000  0.00000000  0.00000000
 [2,]  0.188528808 0.291867115  0.09292215  0.14101475
 [3,]  0.210030355 0.061495731 -0.15285827  0.20446384
 [4,]  0.286906717 0.359406089  0.41450108 -0.50930342
 [5,] -0.482295863 0.074230595  0.03597427 -0.09066711
 [6,] -0.472045204 -0.063993587 0.34458371 -0.36211211
 [7,]  0.314575707 -0.004057342 0.45513656 -0.16976728
 [8,] -0.328502876 0.586331063 -0.06796812  0.01352381
 [9,] -0.132368281 -0.197635191 0.67990000  0.61596451
[10,]  0.003626591 0.624014038  0.04608291  0.36215688
```

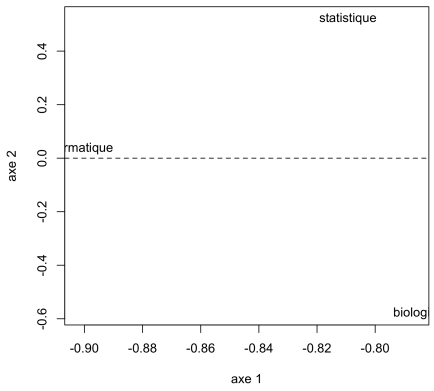
On calcule les coordonnées des variables.

```
matC <- (t(notescr)%*%eigS$vectors)/sqrt(10)
matC
```

```
                  [,1]        [,2]       [,3]          [,4]          [,5]
informatique -0.9021423  0.03658099 -0.4298851 -1.250734e-16 -1.404333e-16
statistique  -0.8093747  0.52175348  0.2696031 -2.413698e-17  2.633125e-17
biologie     -0.7861713 -0.57912995  0.2157387 -1.294620e-16  7.021667e-17
                  [,6]          [,7]          [,8]          [,9]          [,10]
informatique 0.000000e+00 -8.914226e-17  1.053250e-16 -3.291406e-17 -1.755417e-17
statistique  1.755417e-17  1.915873e-16  5.266250e-17 -2.194271e-18 -4.388542e-17
biologie     1.228792e-16  1.091650e-16 -7.021667e-17  1.755417e-17 -7.021667e-17
```

On représente les variables sur le premier plan factoriel (axes 1 et 2).

```
plot(matC[,1], matC[,2], type="n", xlab="axe 1", ylab="axe 2")
text(matC[,1],matC[,2],colnames(notes))
abline(h=0, lty=2)
abline(v=0, lty=2)
```



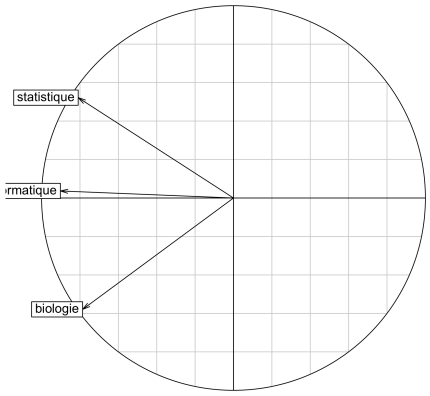
Si on calcule les corrélations entre les individus du tableau **X** et les individus du tableau **L**, on retrouve les coordonnées des variables **C** (à un signe près). Ces

dernières sont les corrélations entre les variables de départ et les variables de synthèse. On peut donc adopter une autre représentation : le cercle des corrélations (que l'on trouve dans la librairie `ade4`).

```
cor(notes,matL)
```

```
                [,1]       [,2]       [,3]
informatique -0.9021423 -0.03658099  0.4298851
statistique  -0.8093747 -0.52175348 -0.2696031
biologie     -0.7861713  0.57912995 -0.2157387
```

```
s.corcircle(as.data.frame(matC[,1:2]))
```



3.3 Mise en oeuvre dans `ade4`

Utiliser la fonction `dudi.pca()` du package `ade4` pour exécuter une ACP centrée réduite :

```
library(ade4)
acp <- dudi.pca(notes, center=TRUE, scale=TRUE, scann = FALSE, nf = 3)
names(acp)
```

```
 [1] "tab"  "cw"   "lw"   "eig"  "rank" "nf"   "c1"   "li"   "co"   "l1"   "call"
[12] "cent" "norm"
```

On a utilisé ici les options `scann = FALSE` pour conserver automatiquement `nf = 3` facteurs. En général on ne procède pas ainsi : on commence par examiner le graphe des valeurs propres qui exprime la part de la variance totale prise en compte par les axes successifs. Essayer avec :

```
acp <- dudi.pca(notes)
```

Répondre 3 à la question `"Select the number of axes:"`. Dans la pratique, on ne conserve qu'un nombre réduit d'axes, ici on les a tous conservés pour des raisons pédagogiques.
L'objet renvoyé par la fonction `dudi.pca()` est très riche. Nous allons examiner tous ses composants un à un.

`tab`

Le data frame `tab` contient les données du tableau initial $\mathbf{X}$ après centrage et réduction $\mathbf{X}_*$. C'est le tableau obtenu `notescr` avec la fonction `scalewt()`.

```
 acp$tab
   informatique statistique   biologie
1    -0.8281491   0.9473309  0.6671291
2    -0.3466670  -1.2179969  0.5082888
3     0.9051862   0.1353330  0.3494486
4    -0.3466670  -0.4059990  0.8656794
5    -1.3096311  -1.4886629 -1.3183741
6     0.6162970  -0.4059990 -0.2859125
7     0.1348150   1.2179969 -1.1992439
8     2.0607430   1.7593289  1.3422001
9     0.4237042   0.1353330  0.7465492
10   -1.3096311  -0.6766650 -1.6757647
```

`cw`

Le vecteur `cw` donne le poids des colonnes (*column weight*), c'est-à-dire le poids des variables. Par défaut, chaque variable a un poids de 1.

```
 acp$cw
[1] 1 1 1
```

`lw`

Le vecteur `lw` donne le poids des lignes (*line weight*), c'est-à-dire le poids des individus. Par défaut, chaque individu a un poids de $\frac{1}{n}$.

```
 acp$lw
 [1] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
 1/10
[1] 0.1
```

`eig`

Le vecteur `eig` donne les valeurs propres (*eigen values*) dans le plus petit des deux espaces diagonalisés.

```
 acp$eig
[1] 2.0870135 0.6089564 0.3040302
 sum(acp$eig)
[1] 3
```

Les valeurs propres nous renseignent sur la fraction de l'inertie totale prise en compte par chaque axe :

```
 (pve <- 100*acp$eig/sum(acp$eig))
[1] 69.56712 20.29855 10.13434
 cumsum(pve)
[1]  69.56712  89.86566 100.00000
```

Dans l'exemple, le premier axe factoriel extrait 69.6 % de l'inertie totale, le deuxième axe factoriel 20.3 % de l'inertie totale. Le premier plan factoriel représente donc 89.9 % de l'inertie initiale. Ceci signifie que lorsqu'on projette le nuage de points initial dans $\mathbb{R}^3$ sur le plan défini par les deux premiers axes factoriels, on en perd que peu d'information (10.1%).

`rank`

Cet entier donne le rang (*rank*) de la matrice diagonalisée, dans ce cas le nombre de variables indépendantes.

```
acp$rank
```

```
[1] 3
```

Généralement, le rang de la matrice est la dimension de l'espace le plus petit. Voici quelques exemples où ce n'est pas le cas :
- une variable est une combinaison de deux autres variables (i.e. chez l'homme, la hauteur de la jambe plus la hauteur de la cuisse donne la hauteur du membre inférieur)
- la somme des lignes est constante (i.e. la répartition des trois secteurs d'activité primaire, secondaire et tertiaire par pays).

`nf`

Cet entier donne le nombre de facteurs conservés dans l'analyse :

```
acp$nf
```

```
[1] 3
```

`c1`

`c1` donne les coordonnées des variables (colonnes) normées à l'unité. Ce sont les vecteurs propres **U**.

```
acp$c1
```

```
                    CS1         CS2        CS3
informatique -0.6244712 -0.04687728  0.7796398
statistique  -0.5602567 -0.66860908 -0.4889523
biologie     -0.5441950  0.74213504 -0.3912638
```

```
sum(acp$cw * acp$c1$CS1^2)
```

```
[1] 1
```

`l1`

`l1` donne les coordonnées des individus (lignes) normées à l'unité :

```
head(acp$l1)
```

```
          RS1         RS2        RS3
1 -0.260714751 -0.12747101 -2.4844173
2  0.430738407  1.54779700  0.2292246
3 -0.575400329  0.16200349  0.9119169
4 -0.018794311  1.19196334 -0.7444292
5  1.640060122  0.10035650  0.4038466
6 -0.001248703  0.03892916  1.4343242
```

```
sum(acp$lw * acp$l1$RS1^2)
```

```
[1] 1
```

`co`

`co` donne les coordonnées des variables (colonnes) normées à la racine carré de la valeur propre correspondante :

```
 acp$co
```

```
                Comp1       Comp2      Comp3
informatique -0.9021423 -0.03658099  0.4298851
statistique  -0.8093747 -0.52175348 -0.2696031
biologie     -0.7861713  0.57912995 -0.2157387
```

```
 sum(acp$cw * acp$co$Comp1^2)
```

```
[1] 2.087013
```

Le lien entre les `c1` et les `co` s'obtient par :

```
 acp$c1$CS1 * sqrt(acp$eig[1])
```

```
[1] -0.9021423 -0.8093747 -0.7861713
```

```
 t(t(acp$c1) * sqrt(acp$eig))
```

```
                 CS1         CS2        CS3
informatique -0.9021423 -0.03658099  0.4298851
statistique  -0.8093747 -0.52175348 -0.2696031
biologie     -0.7861713  0.57912995 -0.2157387
```

`li`

`li` donne les coordonnées des individus (lignes) normées à la racine carré de la valeur propre correspondante :

```
 head(acp$li)
```

```
        Axis1       Axis2      Axis3
1 -0.376641552 -0.09947283 -1.3698811
2  0.622266219  1.20783354  0.1263920
3 -0.831252057  0.12642049  0.5028212
4 -0.027151200  0.93015641 -0.4104703
5  2.369312776  0.07831385  0.2226767
6 -0.001803939  0.03037862  0.7908710
```

```
 sum(acp$lw * acp$li$Axis1^2)
```

```
[1] 2.087013
```

```
 head(acp$l1$RS1 * sqrt(acp$eig[1]))
```

```
[1] -0.376641552  0.622266219 -0.831252057 -0.027151200  2.369312776 -0.001803939
```

```
 head(t(t(acp$l1) * sqrt(acp$eig)))
```

```
          RS1         RS2        RS3
1 -0.376641552 -0.09947283 -1.3698811
2  0.622266219  1.20783354  0.1263920
3 -0.831252057  0.12642049  0.5028212
4 -0.027151200  0.93015641 -0.4104703
5  2.369312776  0.07831385  0.2226767
6 -0.001803939  0.03037862  0.7908710
```

`call`

Cet objet garde une trace de la façon dont ont été conduits les calculs lors de l'appel de la fonction `dudi.pca()` :

```
 acp$call
```

```
dudi.pca(df = notes, center = TRUE, scale = TRUE, scannf = FALSE,
    nf = 3)
```

`cent`

Ce vecteur donne les moyennes (cent pour centrage) des variables analysées :

```
 acp$cent
informatique  statistique     biologie
       14.36        13.75        12.82
 colMeans(notes)
informatique  statistique     biologie
       14.36        13.75        12.82
```

`norm`

Ce vecteur donne les écarts-types (sur $\sqrt{n}$) des variables analysées :

```
 acp$norm
informatique  statistique     biologie
    1.038460     1.847295     2.518253
 sd.n <- function(x) sqrt(var(x)*(length(x)-1)/length(x))
 apply(notes, 2, sd.n)
informatique  statistique     biologie
    1.038460     1.847295     2.518253
```

Remarque.
La représentation simultanée des lignes et des colonnes est obtenue par la fonction [scatter(acp)]. Elle est double et correspond à un des deux choix ci-dessous :

`acp$li` (normées à λ)	$\leftrightarrow$	`acp$c1` (normées à 1)
`acp$l1` (normées à 1)	$\leftrightarrow$	`acp$co` (normées à λ)

3.4 Exercice

1. Reprendre la synthèse théorique précédente sur des données centrées.
2. Appliquer l'analyse en composantes principales centrée sur les notes des 10 étudiants.
3. Discuter des différences entre l'ACP normée et l'ACP centrée.
4. Les deux analyses ci-dessus répondent-elles à la même question ?

Conclusion. Bien réfléchir à la question biologique à laquelle on veut répondre avant de se lancer dans une analyse multivariée.

4 Exemples

4.1 Le signe des corrélations

Les données (exemple n°357 dans [3] d'après Lunn, A. D. & McNeil, D.R. (1991) *Computer-Interactive Data Analysis*, Wiley, New York) sont dans la librairie `ade4`. Le coefficient de corrélation statistique entre deux variables n'a pas forcément le même sens que la relation biologique.
Examiner l'objet `olympic` :

```
data(olympic)
names(olympic)
```

```
[1] "tab"   "score"
```

```
is.list(olympic)
```

```
[1] TRUE
```

```
head(olympic$tab)
```

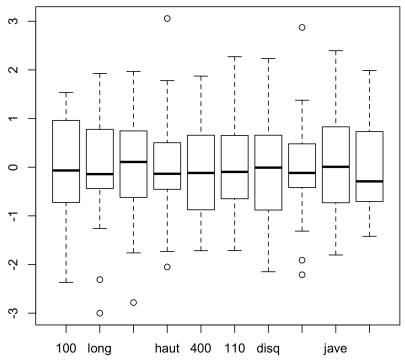
```
    100 long  poid haut   400   110  disq perc  jave   1500
1 11.25 7.43 15.48 2.27 48.90 15.13 49.28  4.7 61.32 268.95
2 10.87 7.45 14.97 1.97 47.71 14.46 44.36  5.1 61.76 273.02
3 11.18 7.44 14.20 1.97 48.29 14.81 43.66  5.2 64.16 263.20
4 10.62 7.38 15.02 2.03 49.06 14.72 44.80  4.9 64.04 285.11
5 11.02 7.43 12.92 1.97 47.44 14.40 41.20  5.2 57.46 256.64
6 10.83 7.72 13.58 2.12 48.34 14.18 43.06  4.9 52.18 274.07
```

```
dim(olympic$tab)
```

```
[1] 33 10
```

```
boxplot(as.data.frame(scale(olympic$tab)))
olympic$score
```

```
 [1] 8488 8399 8328 8306 8286 8272 8216 8189 8180 8167 8143 8114 8093 8083 8036 8021
[17] 7869 7860 7859 7781 7753 7745 7743 7623 7579 7517 7505 7422 7310 7237 7231 7016
[33] 6907
```



1. Commenter les boîtes à moustaches.
2. Réaliser une analyse en composantes principales, discuter des valeurs propres, interpréter les axes et discuter la position des athlètes.
3. Utiliser la fonction graphique `score` et discuter-la.
4. Calculer la corrélation entre les scores des individus sur les axes de l'ACP conservés et `olympic$score`.

4.2 La cohérence d'un jury

Les données sont définies par un ensemble de produits classés par des juges selon leur ordre de préférence.
16 juges ont rangé 28 lots de fruits [4].

```
data(fruits)
is.list(fruits)
```

```
[1] TRUE
```

```
head(fruits$jug)
```

```
      J1 J2 J3 J4 J5 J6 J7 J8 J9 J10 J11 J12 J13 J14 J15 J16
1.nec 10  5  8  3  1 18  5 17  3   4   1   2   5   3   1   1
2.nec  3  1  9  8  6 16  8 10  2   1   8   8   9   5   6   4
3.pea  5 11  5  2  8  8 18  3  4  15  14   4   7   1   3  13
4.pea  6 12  3  4  4  7 17  2  1  16  13   7   3   8   4  14
5.pea  4  2  4 14 17 10 16  1  5  19  21  13   6   2   5  15
6.nec  2  6 16 10 13  2 11  5 13   8  10  14  10   9  15   8
```

Deux questions peuvent alors être soulevées et résolues par des ACP.

- ⋆ Quelle est la cohérence du jury ? Peut-on exprimer un compromis entre jugements, un choix collectif ? Peut-on mettre en évidence la ressemblance entre juges ?
  Les juges sont en colonnes dans une ACP normée. Les moyennes sont toutes égales, les variances aussi, la normalisation n'est ni nécessaire, ni nuisible. On peut l'utiliser pour tracer les cercles de corrélation.

- ⋆ Peut-on faire une typologie des juges ? mettre en évidence ce qui les oppose, montrer qu'il existe plusieurs types de jugements ?
  Les juges sont en lignes dans une ACP centrée. On laissera ainsi dominer dans l'analyse les produits qui ont reçu les appréciations les plus variables. Les deux approches sont antinomiques.

Réaliser chacune des analyses précédentes et discuter les résultats.

4.3 La reconstitution de données

Les données (Carrel et al, 1986 [2]) sont dans la librairie `ade4`. Elles sont caractéristiques des situations rencontrées en écologie : la qualité physico-chimique d'une station du Rhône, située à 70km en amont de Lyon.

```
data(rhone)
names(rhone)
```

```
[1] "tab"   "date"  "disch"
```

```
head(rhone$tab)
```

```
  air.temp wat.temp conduc  pH oxygen secchi caco3 totca  mg  so4  no2  hco3
1        2      5.9    359 8.2     93     67   186  62.9 7.1 35.0 0.55 176.9
2        2      3.4    348 7.9     92    203   176  57.7 7.8 42.1 0.78 158.6
3       10      7.5    260 8.0     94    176   176  60.1 6.3 32.9 0.54 169.6
4       16      9.1    298 7.9    101     85   165  57.7 5.1 32.8 0.63 161.0
5       15      9.6    287 8.2     96     40   167  58.9 4.9 24.4 0.48 176.9
6       10     10.1    277 8.2     98     28   165  57.3 5.3 28.6 0.48 170.8
  suspension organique chloro
1       17.3       2.6    1.4
2        3.7       0.9    1.6
3        4.4       1.2    5.7
4       22.0       3.7    6.2
5       44.9       5.6    2.9
6       92.4       8.8    9.2
```

```
dim(rhone$tab)
```

```
[1] 39 15
```

`air.temp` : température de l'air
`wat.temp` : température de l'eau
`conduc` : conductivité
`pH` potentiel Hydrogène
`oxygen` : saturation en oxygène
`secchi` : transparence (disque Secchi)
`caco3` : Dureté totale ($CaCO_3$)
`totca` : Dureté Calcique (Ca^{++})

`mg` : Dureté Magnésienne (Mg^{++})
`so4` : Sulfates
`no2` : Azote Nitrique
`hco3` : Titre Alcalimétrique complet (HCO_3^-)
`suspension` : matières en suspension
`organique` : matières organiques particulières
`chloro` : chlorophylle a

Réaliser une analyse en composantes principales normées.

1. Donner le nombre d'axes conservés, le pourcentage d'inertie de chaque axe, le pourcentage cumulé.
2. Interpréter le cercle des corrélations des axes 1 et 2.
3. Interpréter la carte factorielle des 39 relevés.

Le vecteur `rhone$date` contient le nombre de jours séparant chaque relevé du premier janvier 1983. Après avoir repris un calendrier de 1983, le jour du printemps était le 20 mars (79 jours à compter du 1er janvier), celui de l'été le 21 juin (172 jours), celui de l'automne le 23 septembre (266 jours) et celui de l'hiver le 21 décembre (355 jours).
On reconstitue ainsi une variable qualitative 'saison'.

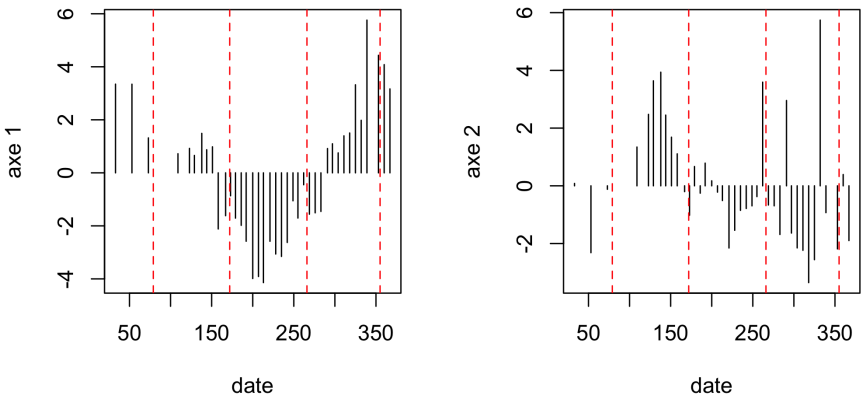
```
saison <- c(rep("H",3),rep(c("P","E","A"),c(8,14,12)),rep("H",2))
saison <- factor(saison)
```

4 Représenter, à l'aide de la fonction `s.class`, le premier plan factoriel en remplaçant les individus par la variable saison.

4 Interpréter à nouveau les résultats de l'analyse à l'aide de cette variable illustrative.

> *La carte factorielle des individus F1 × F2 permet d'observer le cycle thermique (coordonnées des échantillons sur F1) et la répartition des dates en fonction du débit sur l'axe 2...L'interprétation de cette carte factorielle peut être faite de manière plus fonctionnelle en représentant les coordonnées factorielles des individus en fonction du temps*

```
coupure <- c(79,172,266,355)
par(mfrow=c(1,2))
plot(rhone$date,acp$li[,1],type="h",xlab="date",ylab="axe 1")
abline(v=coupure, col="red",lty=2)
plot(rhone$date,acp$li[,2],type="h",xlab="date",ylab="axe 2")
abline(v=coupure, col="red",lty=2)
```

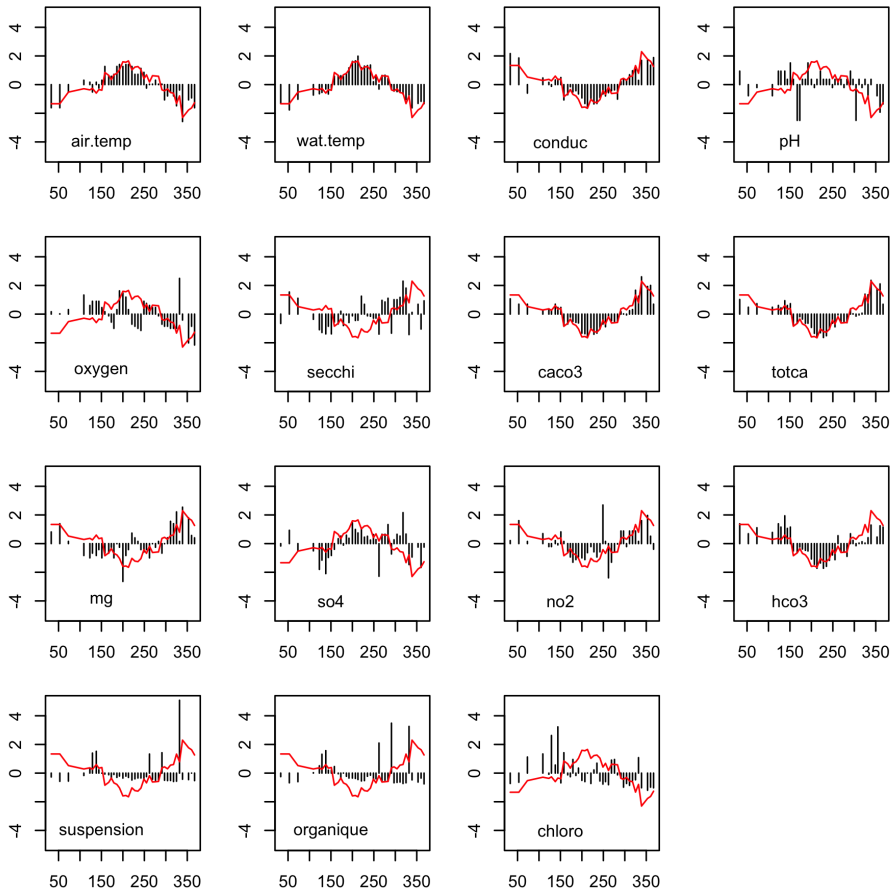


Si les cartes factorielles fournissent les principales bases de l'interprétation, elles ne sont toutefois pas très performantes pour le dépouillement des facteurs lointains attachés à des paramètres souvent complexes, ou à des dates particulières. L'utilisation de la reconstitution progressive des données et la superposition des modèles aux valeurs initiales facilitent singulièrement l'interprétation par une simple visualisation graphique

$$\mathbf{X}_* = \sum \sqrt{(\lambda_k)} \mathbf{v}_k \mathbf{u}_k^T$$

où $\mathbf{u}_k$ et $\mathbf{v}_k$ sont respectivement les k^{eme} vecteurs des matrices $\mathbf{U}$ et $\mathbf{V}$.
Pour une reconstitution des données sur l'axe 1, on a donc : $\widehat{\mathbf{X}_{*1}} = \sqrt{\lambda_1} \mathbf{v}_1 \mathbf{u}_1^T$.

```
rh1 <- reconst(acp,1)
par(mfrow = c(4,4))
par(mar = c(2.6,2.6,1.1,1.1))
for (i in 1:15) {
    plot(rhone$date, scalewt(rhone$tab[,i],center=T, scale=T),type="h",ylim=c(-5,5))
    text(150,-4,label=colnames(rhone$tab)[i])
    lines(rhone$date, scalewt(rh1[,i],center=T, scale=T), lty = 1, col="red")}
```



L'axe 1, représentatif de l'effet saisonnier, explique totalement l'évolution de 5 paramètres : températures de l'air et de l'eau, conductivité, dureté totale et dureté calcique.

4.4 La donnée compositionnelle

Les données compositionnelles sont telles que les éléments d'une ligne sont des proportions et que la somme, en ligne, est égale à 1. On prend, par exemple, un jeu de données proposé par Aitchison [1] sur l'activité journalière d'un enseignant chercheur en statistique. Cette activité est décomposée en six tâches : enseignement, consultation, administration, recherche, autre (autres activités hors travail) et sommeil. 20 jours ont été pris au hasard dans l'année.

```
activite <- read.table("http://pbil.univ-lyon1.fr/R/donnees/activite.txt", h=T)
head(activite)
```

```
  enseignement consultation administration recherche autre sommeil
1        0.162        0.041          0.138     0.123 0.254   0.282
2        0.200        0.039          0.073     0.076 0.346   0.266
3        0.201        0.082          0.115     0.146 0.194   0.261
4        0.134        0.077          0.107     0.146 0.214   0.321
5        0.224        0.080          0.091     0.162 0.195   0.248
6        0.144        0.063          0.103     0.123 0.316   0.252
```

```
apply(activite, 1, sum)
```

```
 [1] 1.000 1.000 0.999 0.999 1.000 1.001 1.000 1.000 1.000 1.000 0.999 0.992 1.001
[14] 1.000 1.000 1.001 1.001 1.000 1.000 1.000
```

Si on réalise une ACP sur le tableau de départ $\mathbf{X}_0$, on obtient une composante principale associée à une valeur propre nulle qui identifie la contrainte que l'on a sur les données.

```
eigen(cov(activite))
```

```
$values
[1] 5.123899e-03 1.600059e-03 4.566554e-04 3.804363e-04 1.486729e-04 3.805366e-07
$vectors
           [,1]       [,2]        [,3]        [,4]        [,5]       [,6]
[1,] -0.35975718  0.4701659  0.63012768  0.292894785  0.026624872 -0.4073839
[2,] -0.13796415  0.2038419 -0.31340430 -0.304247207 -0.768946281 -0.3966704
[3,] -0.04737588 -0.3941692 -0.33881054  0.751910729 -0.049188088 -0.3997554
[4,] -0.29575652  0.1701169 -0.45078348 -0.318971771  0.628871492 -0.4279722
[5,]  0.87217365  0.2351747  0.05564543 -0.009829784  0.100240765 -0.4132369
[6,] -0.03372530 -0.7055326  0.42851150 -0.392984633  0.008105648 -0.4036851
```

Cette dernière valeur propre peut être ignorée parce qu'elle est prédite par la forme des données. Les autres composantes peuvent être interprétées comme d'habitude.

Réaliser l'ACP centrée et commenter les résultats.

Les covariances, les corrélations et par voie de conséquence, les composantes principales ne peuvent être interprétées normalement, en présence de cette contrainte.

Si $\mathbf{x}_i = (x_{i1}\ x_{i2}\ \ldots\ x_{ip})$ est une ligne du tableau $\mathbf{X} = [x_{ij}]$, Aitchison propose de travailler sur la variable transformée suivante :

$$y_{ij} = ln\left(\frac{x_{ij}}{g(\mathbf{x}_i)}\right) \text{ avec } g(\mathbf{x}_i) = \left(\Pi_{j=1}^{p} x_{ij}\right)^{1/p}$$

que l'on peut écrire autrement :

$$y_{ij} = ln(x_{ij}) - \frac{1}{p}\sum_{j=1}^{p} ln(x_{ij})$$

```
logactiv <- log(activite)
temp <- scale(t(logactiv),center=T, scale=F)
neoactiv <- data.frame(t(temp))
```

Il existe toujours une valeur propre nulle mais elle est associée à un vecteur propre proportionnelle au vecteur unité.

```
eigen(cov(neoactiv))
```

```
$values
[1] 1.452970e-01 6.800850e-02 3.639510e-02 1.476292e-02 1.301093e-02 4.880821e-18
$vectors
           [,1]        [,2]        [,3]       [,4]        [,5]       [,6]
[1,] -0.2800657 -0.03704705  0.76257356  0.3922715 -0.13464283 -0.4082483
[2,] -0.4954118 -0.37068291 -0.57585345  0.3345337  0.08351462 -0.4082483
[3,]  0.2617375  0.67433972 -0.28752114  0.2439753 -0.40975650 -0.4082483
[4,] -0.3698204  0.11313404  0.04101785 -0.8111942 -0.15507570 -0.4082483
[5,]  0.6424258 -0.59086171  0.01070204 -0.1240448 -0.23665001 -0.4082483
[6,]  0.2411346  0.21111790  0.04908115 -0.0355416  0.85261042 -0.4082483
```

Comme précédemment, les vecteurs propres sont deux à deux orthogonaux entre eux et en particulier, orthogonaux au dernier. Cela définit des contrastes pour les données en transformées logarithmes et on rejoint ainsi les modèles linéaires classiques.

Réaliser l'ACP sur les données transformées et interpréter les résultats.

Références

[1] J. Aitchison. Principal component analysis of compositional data. *Biometrika*, 70 :57–65, 1983.

[2] G. Carrel, D. Barthelemy, Y. Auda, and D. Chessel. Approche graphique de l'analyse en composantes principales normée : utilisation en hydrobiologie. *Acta Œcologica, Œcologia Generalis*, 7 :189–203, 1986.

[3] D.J. Hand, F. Daly, A.D. Lunn, K.J. McConway, and E. Ostrowski. *A handbook of small data sets*. Chapman & Hall, London, 1994.

[4] J. Kervella. Analyse de l'attrait d'un produit : exemple d'une comparaison de lots de pêches. In *2émes journées européennes Agro-Industrie et Méthodes Statistiques*, pages 103–106. Association pour la Statistique et ses Utilisations, Paris, Nantes 13-14 juin 1991, 1991.