Formation MARBEC, Sète, 20-21 janvier 2020

Analyse multivariée avec R

Analyses à deux tableaux

Monique Simier, IRD, UMR MARBEC, Pôle Modélisation

Sommaire

1. Analyses Inter et Intra Classes

Les analyses inter et intra-classes peuvent être vues comme une extension à un schéma de dualité (ACP, AFC…) de l'analyse de variance à un facteur : décomposition de la variabilité d'une variable Y en une part expliquée par une variable qualitative X (variance inter) et une part non expliquée par X (variance intra).

1.1. Analyses Inter-classes

La function **bca()** de la librairie ade4 réalise une analyse **inter-classes** ou inter-groupes, en anglais **between-class**, permettant de se focaliser sur la part prise en compte dans une analyse simple de type dudi (x), par une variable explicative unique codée en facteur (fac) :

```
bca( x, fac, scannf = TRUE, nf = 2)
```

`x` a duality diagram, object of class dudi from one of the functions dudi.coa, dudi.pca,...

`fac` a factor partitioning the rows of dudi$tab in classes

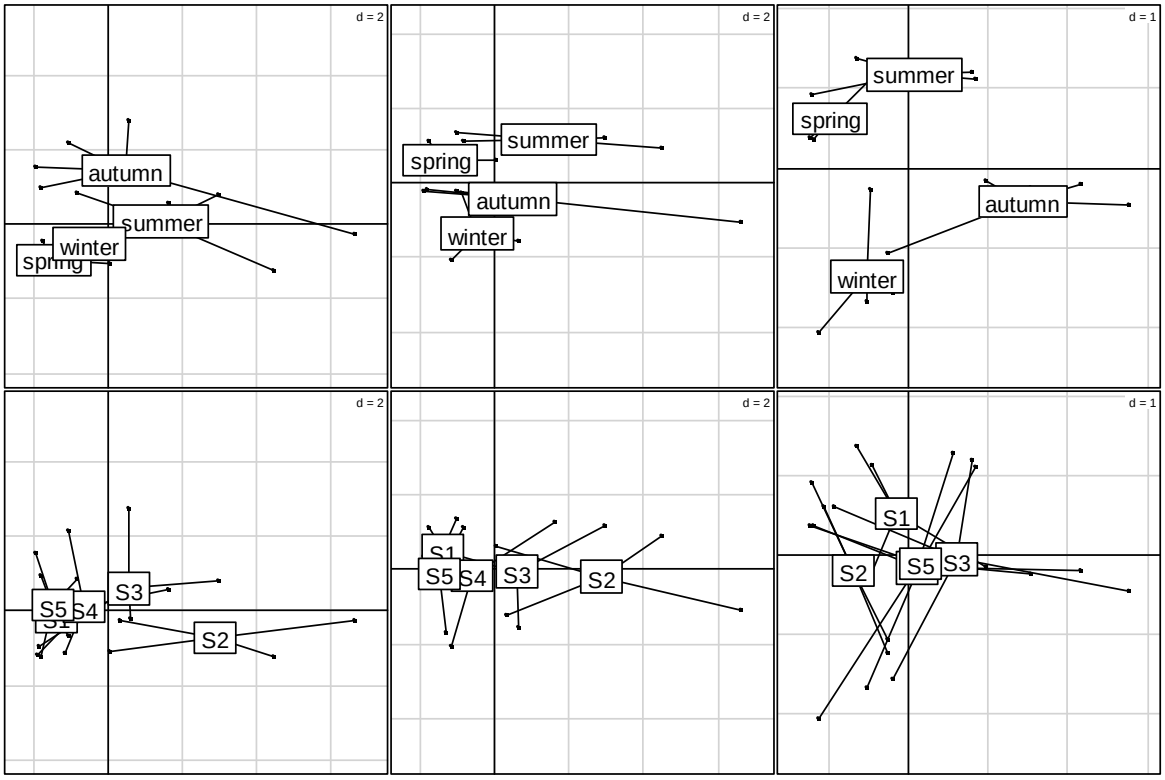
`scannf` a logical value indicating whether the eigenvalue diagram should be displayed

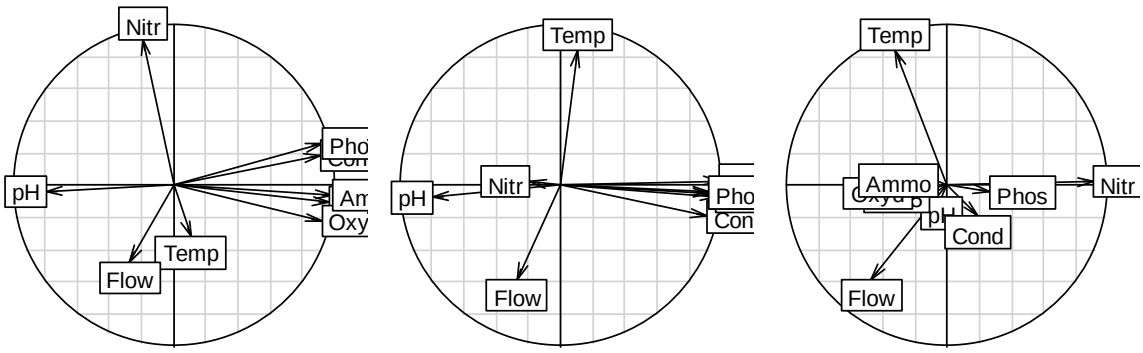
`nf` if scannf FALSE, an integer indicating the number of kept axes

La bca() est elle-même une analyse d'inertie portant sur le tableau des moyennes des variables par classe, les individus du tableau d'origine étant ensuite projetés sur les axes en individus supplémentaires. Nous allons l'appliquer ici pour quantifier les effets dus au site et à la saison dans l'ACP normée du tableau Méaudret milieu. La première étape consiste donc à effectuer l'ACP du tableau mil de meaudret. Suite à cette analyse, on procède à titre exploratoire à la projection sur les axes des points moyens par site et par saison :

```
# Chargement de la librairie ade4
library(ade4)

# Analyses Inter-classes ----

data(meaudret)
mil <- meaudret$env
pca1 <- dudi.pca(mil, scannf=F, nf=5)

# Représentation graphique des étoiles par saison ou par site
plan <- meaudret$design
plan
     season site
sp_1 spring   S1
```

```
sp_2 spring   S2
sp_3 spring   S3
…
wi_3 winter   S3
wi_4 winter   S4
wi_5 winter   S5

windows()
par(mfrow = c(3, 3))
s.class(pca1$li, plan$season, xax = 1, yax = 2, cellipse = 0, clabel =2)
s.class(pca1$li, plan$season, xax = 1, yax = 3, cellipse = 0, clabel =2)
s.class(pca1$li, plan$season, xax = 2, yax = 3, cellipse = 0, clabel =2)
s.class(pca1$li, plan$site, xax = 1, yax = 2, cellipse = 0, clabel =2)
s.class(pca1$li, plan$site, xax = 1, yax = 3, cellipse = 0, clabel =2)
s.class(pca1$li, plan$site, xax = 2, yax = 3, cellipse = 0, clabel =2)
s.corcircle(pca1$co, xax=1, yax = 2, clabel =2)
s.corcircle(pca1$co, xax=1, yax = 3, clabel =2)
s.corcircle(pca1$co, xax=2, yax = 3, clabel =2)
```





Les trois premiers axes de l'ACP normée des données physico-chimiques sont utilisés pour décrire les corrélations entre les variables qui sont liées à la structure spatio-temporelle. Le premier axe (57.5%) prend en compte le pH, la conductivité (Condu), la demande biologique en oxygène (Dbo5), l'oxygène (Oxyd), l'ammoniaque (Ammo) et l'orthophosphate (Phos). Cela peut être interprété comme un gradient de minéralisation. Ce premier axe met également en évidence un taux élevé de pollution pour le site 2 durant l'automne.

Une telle pollution induit une acidité (faible pH), une concentration en oxygène faible, des valeurs élevées de demande biologique en oxygène et d'oxydabilité. Les fortes concentrations en ammoniaque et phosphate sont aussi caractéristiques d'une pollution organique forte. Une restauration de la rivière peut être observée sur les sites 3, 4 et 5. Le site 1 représente un site non pollué. L'évolution temporelle de la pollution est différente selon le cycle saisonnier exprimé par la température de l'eau (sur l'axe 3).

Par conséquent, cette analyse mélange à la fois une typologie saisonnière et une typologie spatiale qui contrôlent le processus spatio-temporel produit par l'eau qui coule et l'évolution de la température de l'air. Ce processus peut se décomposer (au sens de la géométrie) c'est-à-dire que l'on peut choisir de se focaliser sur un composant donné (espace ou temps) du plan d'échantillonnage ou alors choisir d'éliminer ce composant.

On pourrait réaliser pour chaque variable physico-chimique une ANOVA, pour décomposer sa variance selon les effets saisonnier et spatial. Ici on veut décomposer la variabilité décrite par l'ACP de l'ensemble du tableau mil en :

- variabilité inter-sites + variabilité intra-sites

- variabilité inter-saisons + variabilité intra-saisons

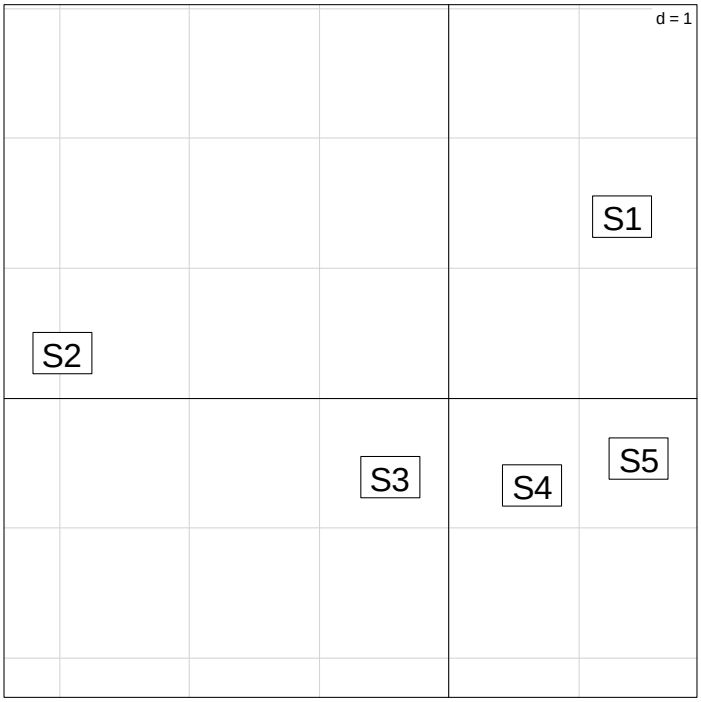
```
# inter-sites ----
# Se focaliser sur les différences entre les sites
bca1 <- bca(pca1,plan$site, scannf=F, nf=2)
bca1
Between analysis
call: bca.dudi(x = pca1, fac = plan$site, scannf = F, nf = 2)
class: between dudi

$nf (axis saved) : 2
$rank:  4
$ratio:  0.3805115

eigen values: 2.681 0.6208 0.1132 0.009503

  vector length mode    content
1 $eig   4      numeric eigen values
2 $lw    5      numeric group weigths
3 $cw    9      numeric col weigths

  data.frame nrow ncol
1 $tab       5    9
2 $li        5    2
3 $l1        5    2
4 $co        9    2
5 $c1        9    2
```

```
6 $ls        20   2
7 $as        5    2
  content
1 array class-variables
2 class coordinates
3 class normed scores
4 column coordinates
5 column normed scores
6 row coordinates
7 inertia axis onto between axis
```

L'élément $ratio est une valeur entre 0 et 1, qui indique la part prise en compte par l'effet testé (ici le site) dans l'analyse simple (ici l'ACP normée de mil). Ici bac1$ratio vaut 0.38. On vérifie que c'est le rapport de l'inertie totale de l'analyse inter-classe sum(bca1$eig) sur l'inertie totale de l'analyse simple sum(pca1$eig).

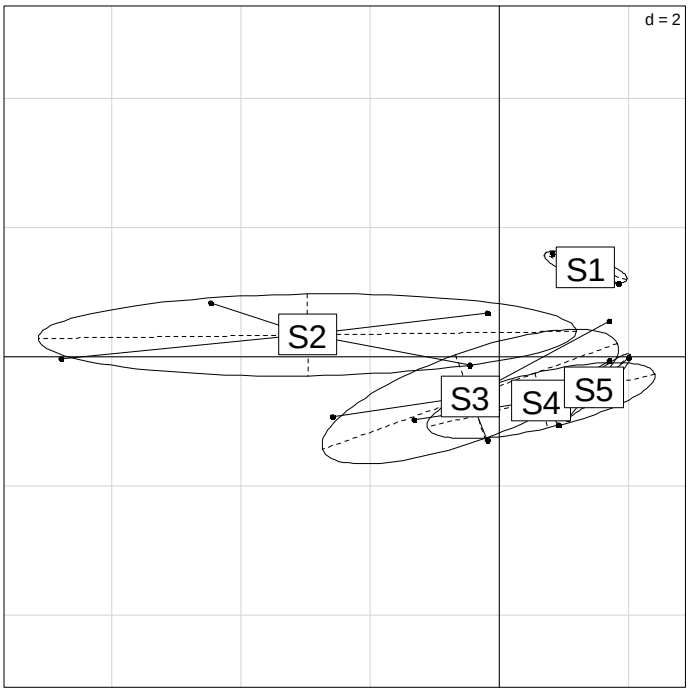
On peut représenter sur les axes les 5 individus de l'ACP inter-sites à savoir les moyennes par sites. Leurs coordonnées sont dans bca1$li.

```
s.label(bca1$li)
```

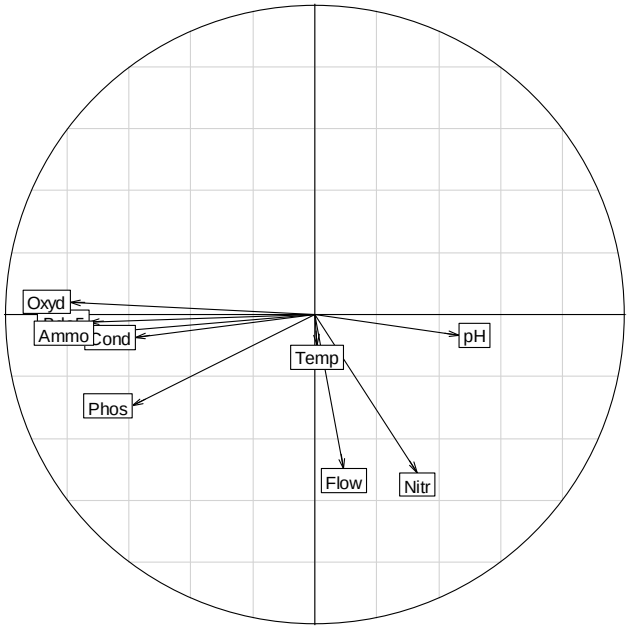


On peut aussi projeter les 20 échantillons du tableau d'origine en individus supplémentaires (coordonnées dans bca1$ls). Ici en utilisant la fonction s.class qui positionne également les centres des classes.

```
s.class(bca1$ls,plan$site)
```



Enfin, pour interpréter les axes, on peut représenter le cercle de corrélation des variables dans l'ACP inter-sites.



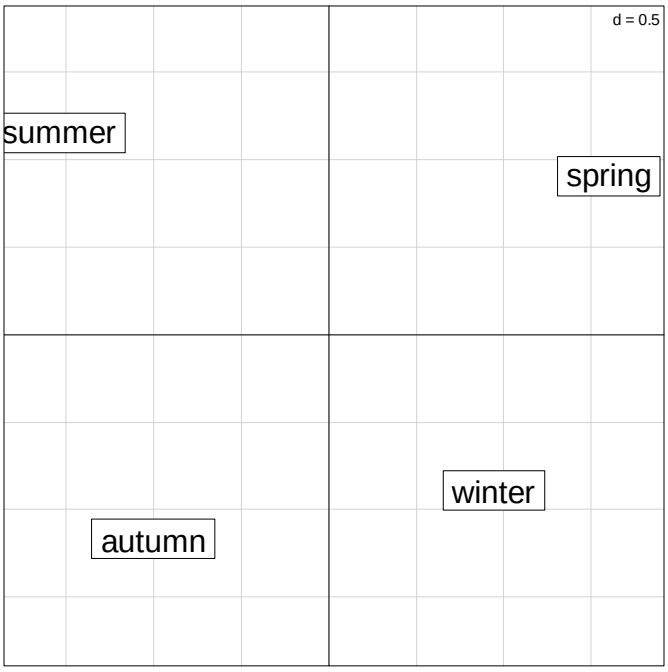
L'axe 1 de l'ACP inter-sites résume le principal phénomène spatial observé : la pollution, organique, particulièrement forte au site S2, induit une acidité (faible pH), des valeurs élevées de demande biologique en oxygène, de conductivité et d'oxydabilité. Les fortes concentrations en ammoniaque et phosphate sont aussi caractéristiques d'une pollution organique forte. Une restauration de la rivière peut être observée sur les sites 3, 4 et 5. Le site 1 représente un site non pollué. Les variables Temp (température), Flow (débit) et Nitr (Nitrate) ne relèvent pas de cet effet spatial.

On procède de la même manière pour l'effet saison.

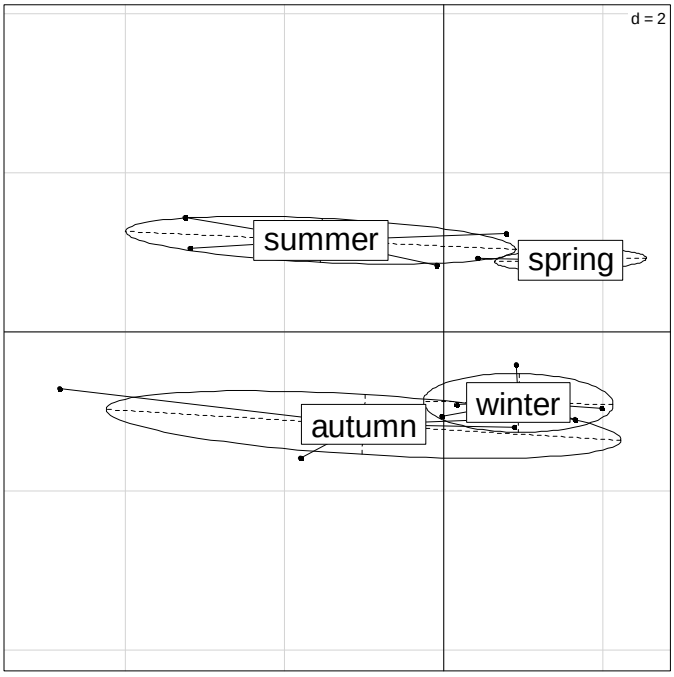
```
# inter-saisons ----
# Se focaliser sur les différences entre les saisons
bca2 <- bca(pca1,plan$season, scannf=F, nf=2)
bca2
Between analysis
call: bca.dudi(x = pca1, fac = plan$season, scannf = F, nf = 2)
class: between dudi

$nf (axis saved) : 2
$rank:  3
$ratio:  0.3722686

eigen values: 1.707 1.078 0.5652

  vector length mode    content
1 $eig   3      numeric eigen values
2 $lw    4      numeric group weigths
3 $cw    9      numeric col weigths

  data.frame nrow ncol
1 $tab       4    9
2 $li        4    2
3 $l1        4    2
4 $co        9    2
5 $c1        9    2
6 $ls        20   2
7 $as        5    2
  content
1 array class-variables
2 class coordinates
3 class normed scores
4 column coordinates
5 column normed scores
6 row coordinates
7 inertia axis onto between axis
```

La part prise en compte par l'effet saison dans l'analyse simple (bca2$ratio) vaut 0.37, ce qui est proche de la valeur obtenue pour l'effet site (0.38).

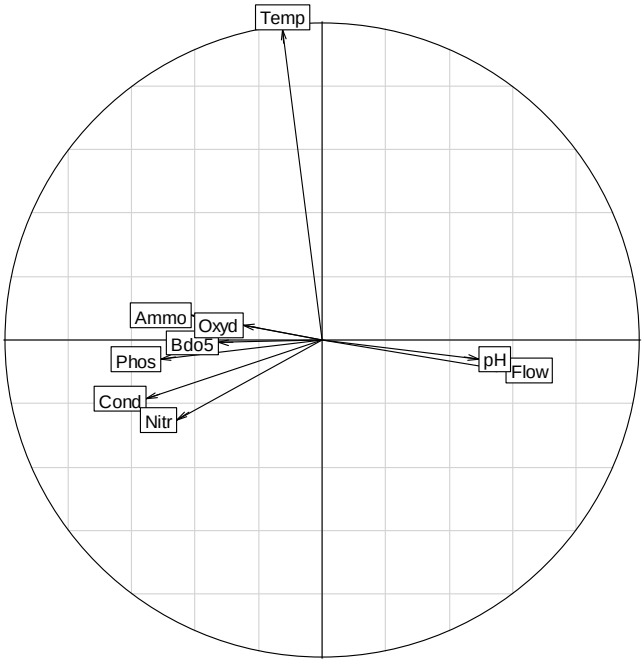
```
s.label(bca2$li)
```




```
s.class(bca2$ls,plan$season)
```




```
s.corcircle(bca2$co)
```



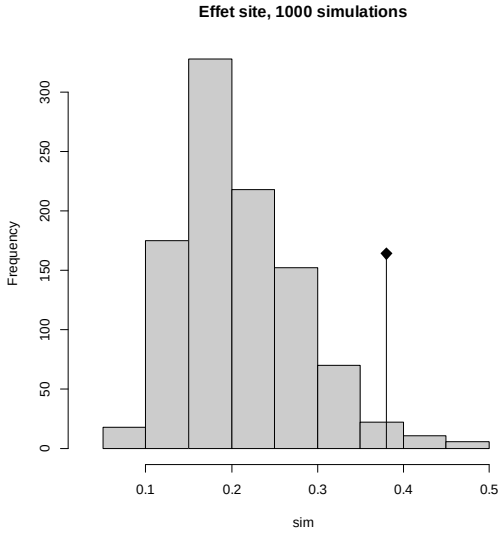
En moyenne, sur l'axe 1, on observe une pollution plus importante en automne et en été. Pendant l'hiver et surtout le printemps, la valeur élevée du débit de l'eau (Flow) conduit à une dilution de la pollution organique dans la rivière. L'axe 2 décrit l'influence du rythme saisonnier avec la température de l'eau.

Des tests de permutation de Monte-Carlo avec la fonction **rtest()** permettent de tester la significativité des effets site et saison. Le principe est de réaliser un grand nombre de permutations (ici 1000) des lignes du tableau d'origine en les attribuant aléatoirement à une classe, à réaliser autant d'analyses inter-classes, ce qui fournit autant de valeurs simulées d'inertie projetée.

Ici l'effet site est significatif avec une p-value de 0.023. Une représentation graphique adaptée permet de représenter l'histogramme de distribution des 1000 valeurs d'inertie simulées et de positionner la valeur observée de 0.38 sur cette distribution.

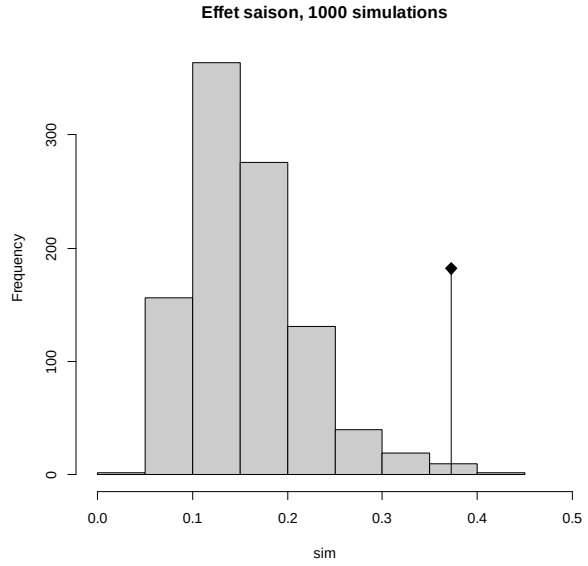
```
# Tests de permutations
rt1 <- rtest(bca1, nrepet=1000)
rt1
Monte-Carlo test
Call: rtest.between(xtest = bca1, nrepet = 1000)

Observation: 0.3805115

Based on 1000 replicates
Simulated p-value: 0.02297702
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
2.407948114 0.210269744 0.004998482
```

```
plot(rt1, main="Effet site, 1000 simulations")
```



De la même manière pour l'effet saison, on obtient une p-value de 0.005 :

```
rt2 <- rtest(bca2, nrepet=1000)
rt2
Monte-Carlo test
Call: rtest.between(xtest = bca2, nrepet = 1000)

Observation: 0.3722686

Based on 1000 replicates
Simulated p-value: 0.004995005
Alternative hypothesis: greater

    Std.Obs Expectation    Variance
3.518015732 0.157168857 0.003738387

plot(rt2, main="Effet saison, 1000 simulations")
```



1.2. Analyses Intra-classes

La function **wca()** de la librairie ade4 réalise l'analyse intra-classes (within-class), c'est-à-dire l'analyse du tableau débarrassé de l'effet considéré. Sa synthaxe est similaire à celle de bca() :

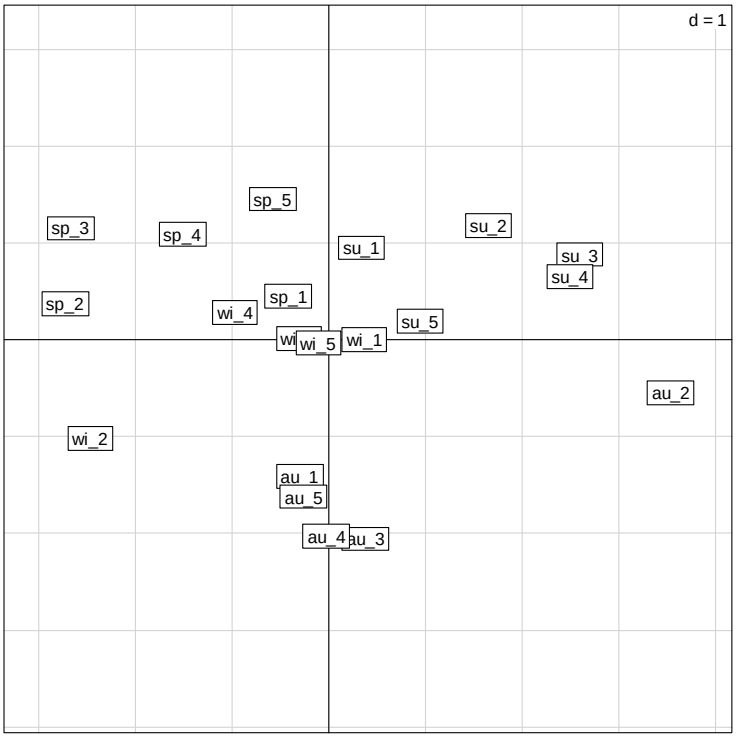
```
wca( x, fac, scannf = TRUE, nf = 2)
```

x	a duality diagram, object of class dudi from one of the functions dudi.coa, dudi.pca,...
fac	a factor partitioning the rows of dudi$tab in classes
scannf	a logical value indicating whether the eigenvalue diagram should be displayed
nf	if scannf FALSE, an integer indicating the number of kept axes

```
# Analyses Intra-classes ----

# intra-sites ----
# Eliminer l'effet site
wca1 <- wca(pca1,plan$site, scannf=F, nf=2)
wca1
Within analysis
call: wca.dudi(x = pca1, fac = plan$site, scannf = F, nf = 2)
class: within dudi

$nf (axis saved) : 2
$rank:  9
$ratio:  0.6194885

eigen values: 2.703 1.146 0.9934 0.4422 0.1846 ...

  vector length mode    content
1 $eig   9      numeric eigen values
2 $lw    20     numeric row weigths
3 $cw    9      numeric col weigths
4 $tabw  5      numeric class weigths
5 $fac   20     numeric factor for grouping

  data.frame nrow ncol
1 $tab       20   9
2 $li        20   2
3 $l1        20   2
4 $co        9    2
5 $c1        9    2
6 $ls        20   2
7 $as        5    2
  content
1 array class-variables
```

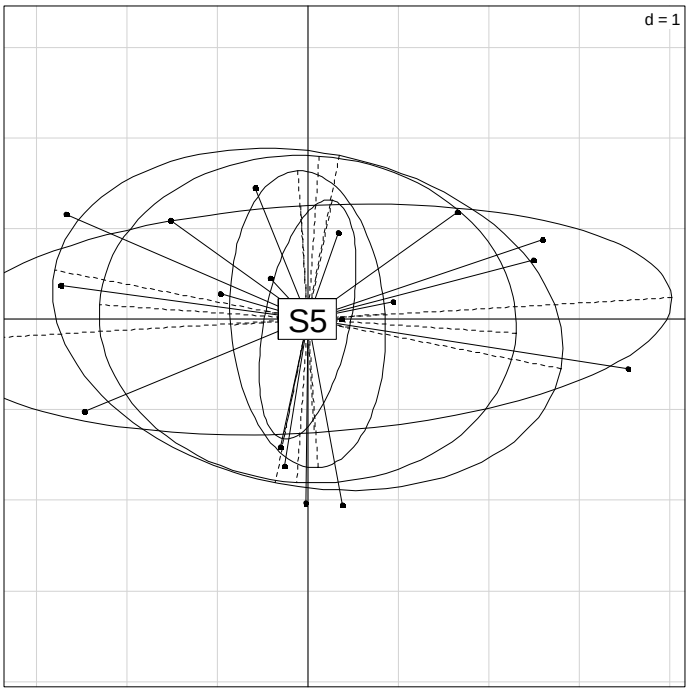
```
2 row coordinates
3 row normed scores
4 column coordinates
5 column normed scores
6 supplementary row coordinates
7 inertia axis onto within axis
```

```
s.label(wca1$li)
```



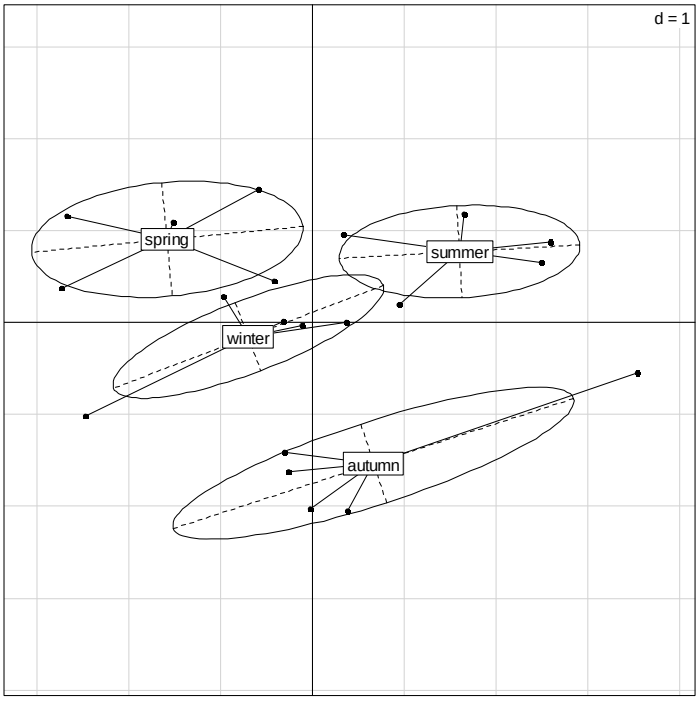
Dans le cas de l'analyse intra-classes, le tableau de données garde la même dimension que le tableau d'origine, les données étant centrées par classe. On a dans wca1$li les coordonnées des 20 échantillons. En les représentant avec s.class, on vérifie que les points-moyens par site se positionnent tous à l'origine des axes :

```
s.class(wca1$li,plan$site)
```



Les points-moyens par saison permettent de visualiser la variabilité saisonnière débarrassée de la variabilité spatiale :

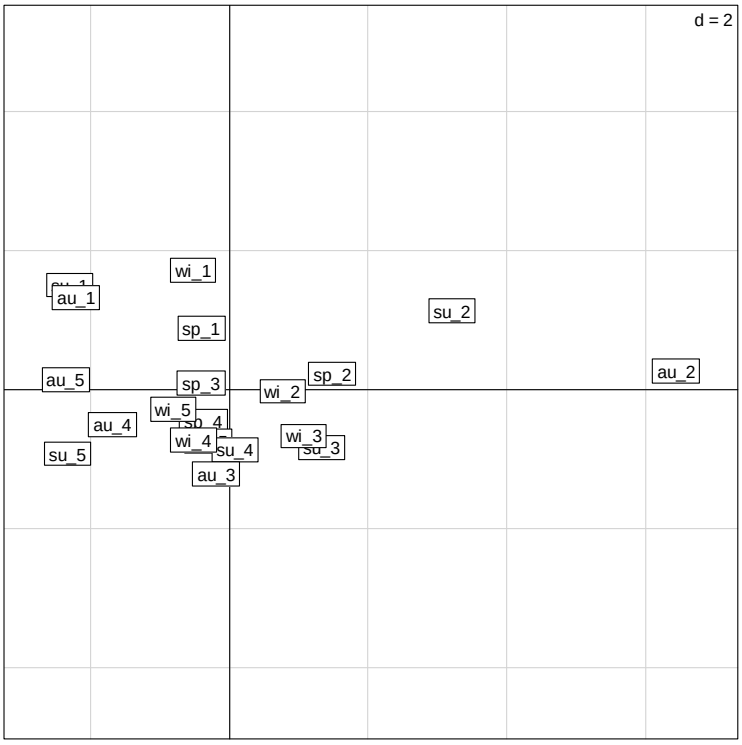
```
s.class(wca1$li,plan$season)
```



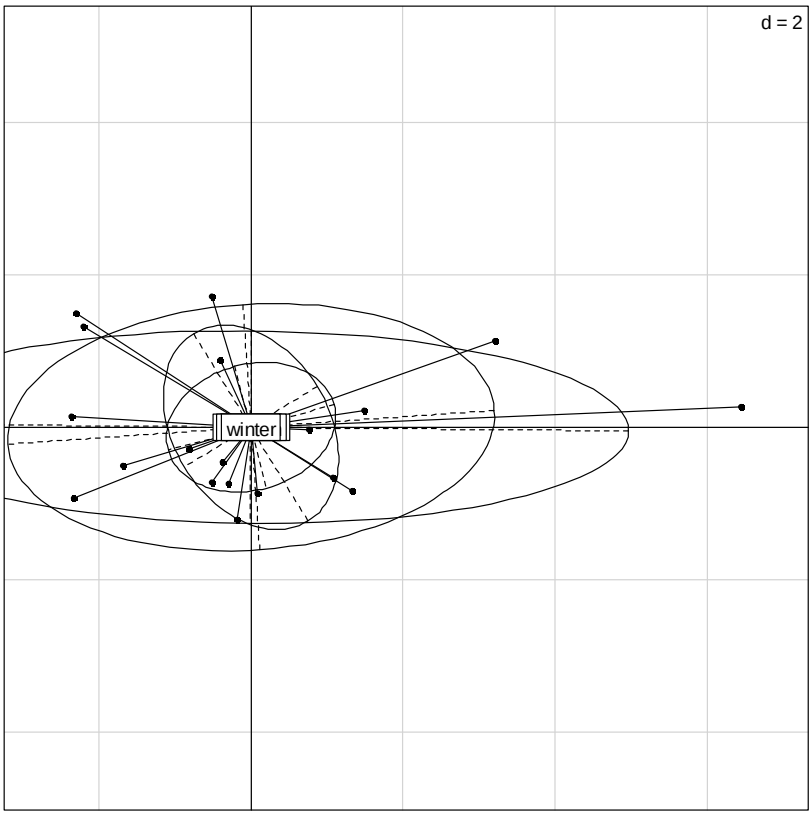

```
# intra-saisons ----
wca2 <- wca(pca1,plan$season, scannf=F, nf=2)
wca2
Within analysis
call: wca.dudi(x = pca1, fac = plan$season, scannf = F, nf = 2)
class: within dudi

$nf (axis saved) : 2
```

```
$rank:  9
$ratio:  0.6277314

eigen values: 4.158 0.7531 0.4054 0.228 0.05361 ...

  vector length mode    content
1 $eig   9      numeric eigen values
2 $lw    20     numeric row weigths
3 $cw    9      numeric col weigths
4 $tabw  4      numeric class weigths
5 $fac   20     numeric factor for grouping

  data.frame nrow ncol
1 $tab       20   9
2 $li        20   2
3 $l1        20   2
4 $co        9    2
5 $c1        9    2
6 $ls        20   2
7 $as        5    2
…
```

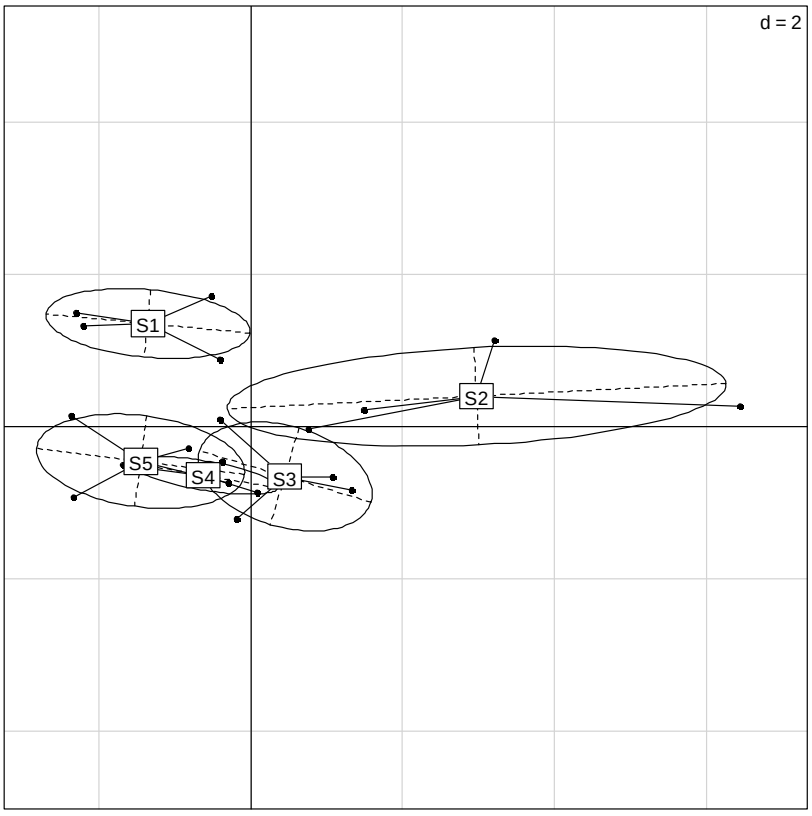
```
s.label(wca2$li)
```




```
s.class(wca2$li,plan$season)
# Points moyens des saisons à l'origine des axes
```




```
s.class(wca2$li,plan$site)
```



1.3. Bilan de la décomposition de la variabilité saisonnière et spatiale

Les ratios d'inertie expliquée par le site (inter-site bca1) et par la saison (inter-saison bca2) sont assez proches :

```
bca1$ratio
[1] 0.3805115
bca2$ratio
[1] 0.3722686
```

Inertie intra-site (wca1) et intra-saison (wca2) :

```
wca1$ratio
[1] 0.6194885
wca2$ratio
[1] 0.6277314
```

On vérifie que la somme des variances inter et intra pour un facteur donné vaut 1

```
bca1$ratio + wca1$ratio # 1 (car plan équilibré)
[1] 1
bca2$ratio + wca2$ratio # 1
[1] 1
```

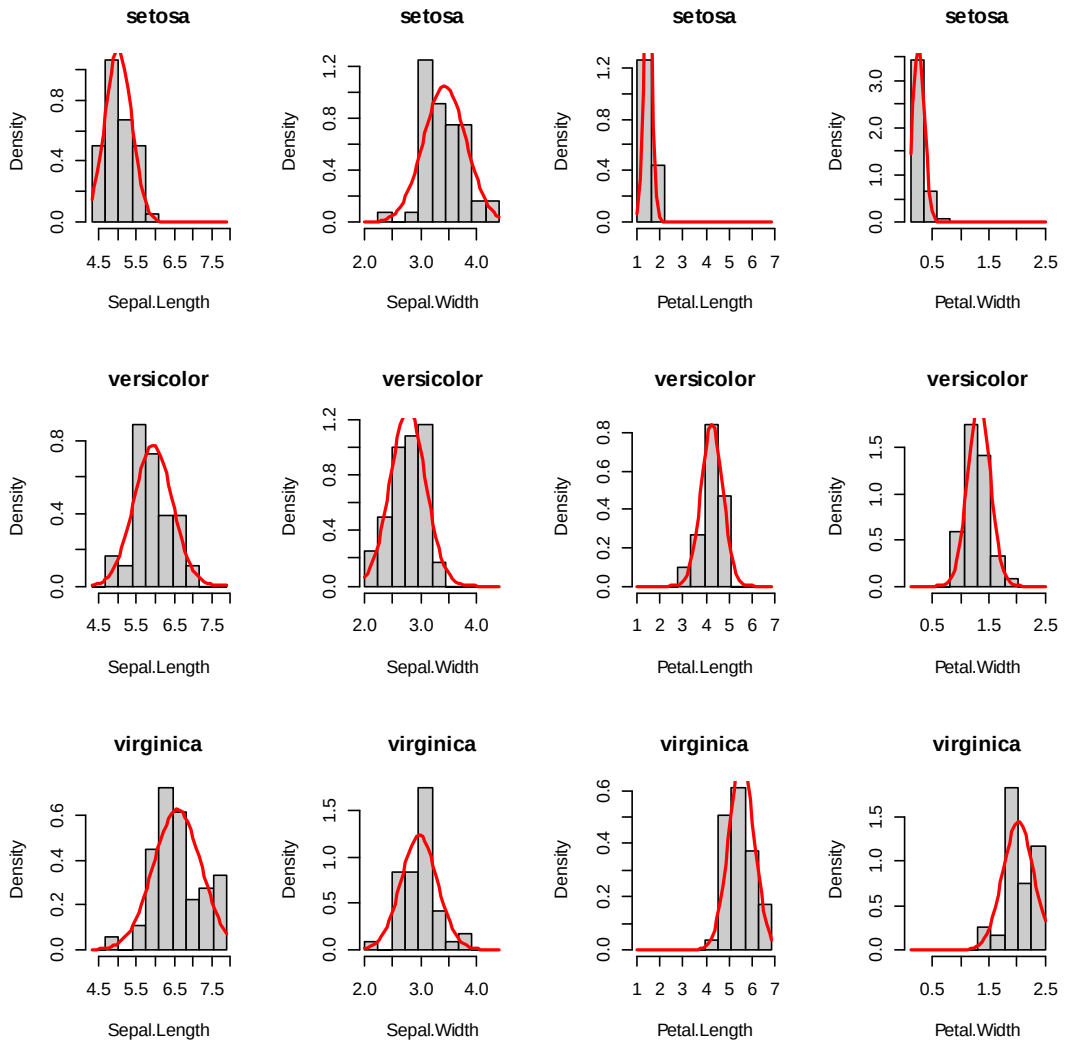
2. Analyse Discriminante Linéaire

L'analyse factorielle discriminante ou **analyse discriminante linéaire** cherche à discriminer des groupes connus *a priori* au sein d'un ensemble d'observations, à partir d'un ensemble de variables prédictives (mesures). On parle de **discrimination descriptive** quand la question est : « qu'est-ce qui sépare les groupes ? », et un problème de **discrimination prédictive** quand la question est : « à quel groupe est-ce que je peux affecter un nouvel individu dont je connais les mesures, mais pas le groupe, et avec quelle erreur ? »

L'analyse discriminante est utilisée dans de nombreux domaines, par exemple en biologie, lorsque l'on veut affecter un objet à sa famille d'appartenance à partir de ses caractéristiques physiques. Les iris de Sir Ronald Fisher — qui est à l'origine de cette méthode — en est un exemple, il s'agit de reconnaître la variété d'iris (setosa, virginica, et versicolor) à partir de la longueur/largeur de ses pétales et sépales.

```
data(iris)

# Approche univariée
# Histogrammes par Species pour les 4 variables numériques
windows()
par(mfcol = c(3, 4))
for (k in 1:4) {
  j0 <- names(iris)[k]
  br0 <- seq(min(iris[, k]), max(iris[, k]), le = 11)
  x0 <- seq(min(iris[, k]), max(iris[, k]), le = 50)
```

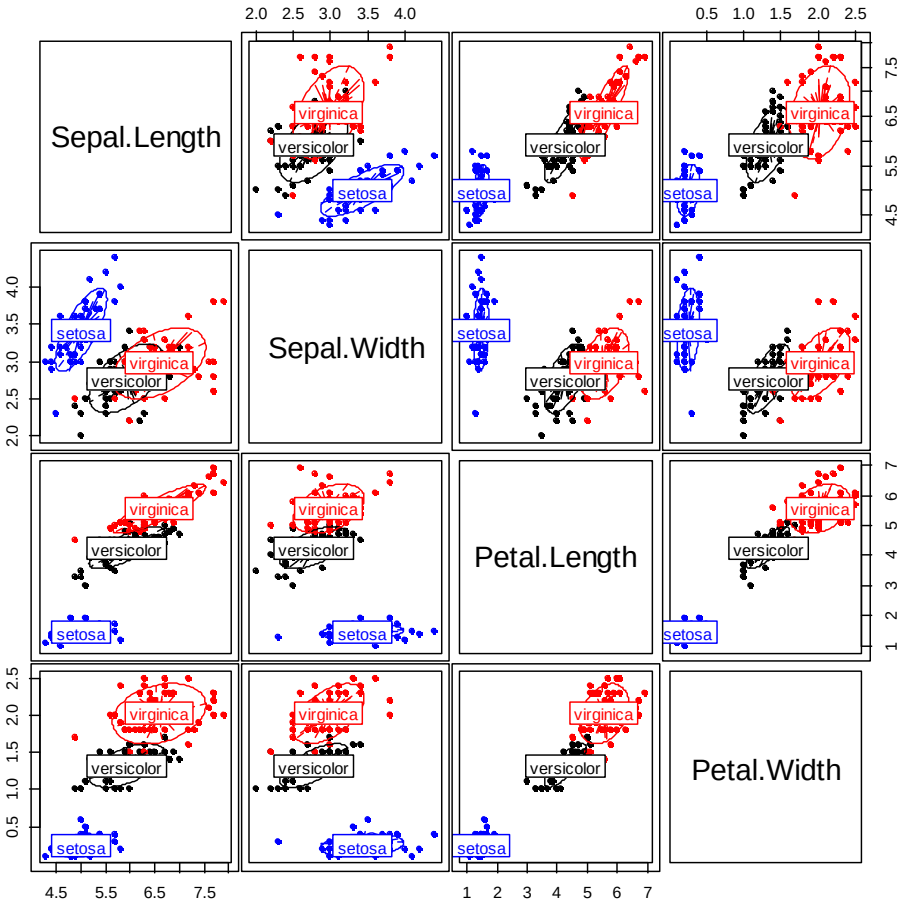
```
  for (i in 1:3) {
    i0 <- levels(iris$Species)[i]
    x <- iris[iris$Species == i0, j0]
    hist(x, br = br0, proba = T, col = grey(0.8), main = i0,
         xlab = j0)
    lines(x0, dnorm(x0, mean(x), sd(x)), col = "red", lwd = 2)
  }
}
```




```
# Approche bivariée
# Analyses de variance à un facteur (Species) pour chaque variable
numérique
# Exemple : effet espèce sur Sepal.Length

# Moyennes et écarts-types par groupe
tapply(iris$Sepal.Length, iris$Species, mean)
    setosa versicolor  virginica
     5.006      5.936      6.588
tapply(iris$Sepal.Length, iris$Species, sd)
    setosa versicolor  virginica
 0.3524897  0.5161711  0.6358796

# ANOVA à un facteur
anova(lm(iris$Sepal.Length ~ iris$Species))
Analysis of Variance Table
```

```
Response: iris$Sepal.Length
              Df Sum Sq Mean Sq F value    Pr(>F)
iris$Species   2 63.212  31.606  119.26 < 2.2e-16
Residuals    147 38.956   0.265

iris$Species ***
Residuals
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
# Nuages de points de toutes les variables numériques 2 à 2
windows()
par(mar = c(0, 0, 0, 0))
# Nuages de points simples
pairs(iris[, 1:4])
# Avec des étoiles par Species
pan1 <- function(x, y, ...) {
  xy <- cbind.data.frame(x, y)
  s.class(xy, iris$Species, include.origin = F, add.plot = T, clab = 1.5,
          col = c("blue", "black", "red"), cpoint = 2, cstar = 0.5)
}
pairs(iris[, 1:4], panel = pan1)
```



On peut se poser d'abord la question de la valeur discriminante de chaque mesure. Ici la réponse est oui pour toutes les mesures si on considère l'ANOVA à un facteur (Species) réalisée sur chacune des 4 mesures :

```
apply(iris[, 1:4], 2, function(x) summary(lm(x ~ iris[, 5])))
```

```
$Sepal.Length
Call:
lm(formula = x ~ iris[, 5])

Residuals:
    Min      1Q  Median      3Q     Max
-1.6880 -0.3285 -0.0060  0.3120  1.3120

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           5.0060     0.0728  68.762  < 2e-16 ***
iris[, 5]versicolor   0.9300     0.1030   9.033 8.77e-16 ***
iris[, 5]virginica    1.5820     0.1030  15.366  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5148 on 147 degrees of freedom
Multiple R-squared:  0.6187,  Adjusted R-squared:  0.6135
F-statistic: 119.3 on 2 and 147 DF,  p-value: < 2.2e-16

$Sepal.Width
Call:
lm(formula = x ~ iris[, 5])

Residuals:
   Min     1Q Median     3Q    Max
-1.128 -0.228  0.026  0.226  0.972

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          3.42800    0.04804  71.359  < 2e-16 ***
iris[, 5]versicolor -0.65800    0.06794  -9.685  < 2e-16 ***
iris[, 5]virginica  -0.45400    0.06794  -6.683 4.54e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3397 on 147 degrees of freedom
Multiple R-squared:  0.4008,  Adjusted R-squared:  0.3926
F-statistic: 49.16 on 2 and 147 DF,  p-value: < 2.2e-16

$Petal.Length
Call:
lm(formula = x ~ iris[, 5])

Residuals:
   Min     1Q Median     3Q    Max
-1.260 -0.258  0.038  0.240  1.348
```

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.46200    0.06086   24.02   <2e-16 ***
iris[, 5]versicolor  2.79800    0.08607   32.51   <2e-16 ***
iris[, 5]virginica   4.09000    0.08607   47.52   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4303 on 147 degrees of freedom
Multiple R-squared:  0.9414,  Adjusted R-squared:  0.9406
F-statistic:  1180 on 2 and 147 DF,  p-value: < 2.2e-16

$Petal.Width
Call:
lm(formula = x ~ iris[, 5])

Residuals:
   Min     1Q Median     3Q    Max
-0.626 -0.126 -0.026  0.154  0.474

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.24600    0.02894    8.50 1.96e-14 ***
iris[, 5]versicolor  1.08000    0.04093   26.39  < 2e-16 ***
iris[, 5]virginica   1.78000    0.04093   43.49  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2047 on 147 degrees of freedom
Multiple R-squared:  0.9289,  Adjusted R-squared:  0.9279
F-statistic:   960 on 2 and 147 DF,  p-value: < 2.2e-16
```

L’idée de l’analyse discriminante est de chercher une variable discriminante synthétique qui est une combinaison linéaire des variables d’origine permettant de discriminer au mieux les groupes. On va comparer ici deux fonctions disponibles sous R pour l’analyse discriminante.

On réalise d’abord une Analyse Discriminante Linéaire (LDA en anglais) avec la fonction **lda()** du package MASS

```
library(MASS)
?lda
lda1 <- lda(as.matrix(iris[, 1:4]), iris$Species)
lda1
Call:
lda(as.matrix(iris[, 1:4]), grouping = iris$Species)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
```

```
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785

Proportion of trace:
   LD1    LD2
0.9912 0.0088
```

lda() fournit une combinaison linéaire des variables de départ, avec les coefficients qui sont dans la colonne LD1 et que l'on retrouve dans l'objet lda1$scaling :

```
lda1$scaling
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785
```

On peut calculer cette combinaison w1 :

```
w1 <- as.vector(as.matrix(iris[, 1:4]) %*% lda1$scaling[, 1])
head(w1)
[1] 5.956693 5.023581 5.384722 4.708094 6.027203 5.596840
tail(w1)
[1] -8.952466 -7.750110 -7.284671 -7.072847 -7.991252 -6.788261
```

On réalise maintenant une Analyse Discriminante Linéaire avec la fonction **discrimin()** du package ade4

```
# Analyse discriminante linéaire
# avec la fonction discrimin d'ade4
dis1 <- discrimin(dudi.pca(iris[, 1:4], scannf = F), iris$Species, scannf =
F)
dis1
Discriminant analysis
call: discrimin(dudi = dudi.pca(iris[, 1:4], scannf = F), fac =
iris$Species,
    scannf = F)
class: discrimin

$nf (axis saved) : 2

eigen values: 0.9699 0.222

  data.frame nrow ncol
1 $fa        4    2
```

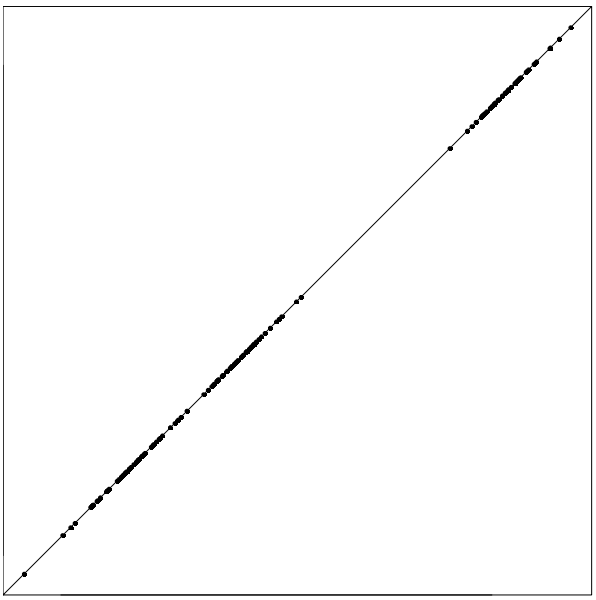
```
2 $li        150  2
3 $va        4    2
4 $cp        4    2
5 $gc        3    2
  content
1 loadings / canonical weights
2 canonical scores
3 cos(variables, canonical scores)
4 cos(components, canonical scores)
5 class scores
```

discrimin() fournit une combinaison linéaire des variables normalisées (en 1/n) avec les coefficients qui sont dans la composante dis1$fa (fa pour facteur, dans le vocabulaire du schéma de dualité).

```
dis1$fa
                    DS1         DS2
Sepal.Length  0.1200150  0.01772302
Sepal.Width   0.1168775  0.83778380
Petal.Length -0.6790443 -1.46087856
Petal.Width  -0.3743571  1.92176982

w2 <- as.vector(scalewt(iris[, 1:4]) %*% dis1$fa[, 1])
head(w2)
[1] 1.413522 1.249914 1.313235 1.194598 1.425885 1.350427
tail(w2)
[1] -1.2005876 -0.9897715 -0.9081633 -0.8710231 -1.0320523 -0.8211248
```

Ces coefficients w1 et w2 sont cohérents :

```
plot(w1, w2, pch = 20)
abline(lm(w2 ~ w1))
```



discrimin donne une combinaison linéaire de variance totale=1 (en 1/n) qui maximise la variance inter-classe (première valeur propre) :

```
var(w2) * 149/150
[1] 1
dis1$eig
[1] 0.9698722 0.2220266

summary(lm(w2 ~ iris[, 5]))$r.squared
[1] 0.9698722
```

lda donne une combinaison linéaire de variance intra-classe unité qui maximise la variance inter-classe :

```
tapply(w1, iris[, 5], var)
    setosa versicolor  virginica
 0.7181898  1.0736485  1.2081617
mean(tapply(w1, iris[, 5], var))
[1] 1
```

On explore le lien avec la MANOVA (ANOVA multivariée)

```
size <- as.matrix(iris[, 1:4])
spec <- iris[, 5]
m1 <- manova(size ~ spec)
summary(m1, test = "Pillai")
           Df Pillai approx F num Df den Df
spec        2 1.1919   53.466      8    290
Residuals 147
             Pr(>F)
spec      < 2.2e-16 ***
Residuals
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

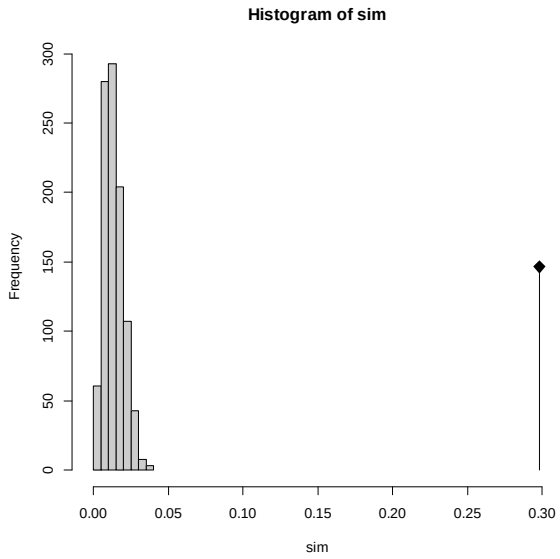
Le **critère de Pillai** de la MANOVA correspond à la somme des valeurs propres de l'analyse discriminante :

```
sum(dis1$eig)
[1] 1.191899
```

A la fonction discrimin() est associé un test non paramétrique de significativité **randtest()** basé sur le même principe de permutations aléatoires que celui de l’analyse inter-classes.

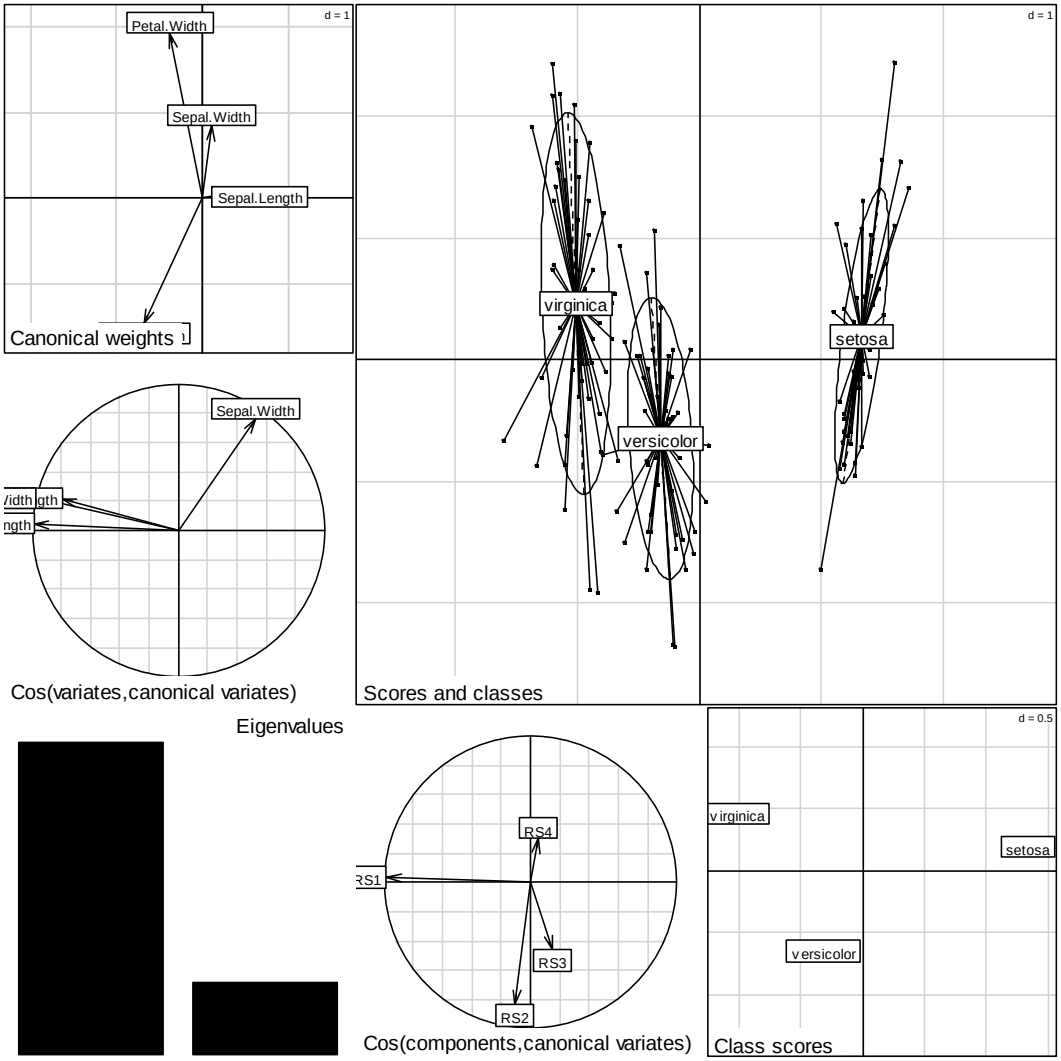
La statistique observée est le critère de Pillai divisé par le rang de l'analyse de départ :

```
sum(dis1$eig)/4
[1] 0.2979747

plot(randtest(dis1))
```



Si on utilise l'analyse discriminante dans un but **descriptif**, la fonction plot() associée à la fonction discrimin() d'ade4 permet de représenter graphiquement les résultats :

```
plot(dis1)
```



Ce graphe contient :

- Canonical weights : les poids canoniques ou loadings (coefficients des combinaisons linéaires de variance unité et de variance inter maximales). Les variables utilisées sont les colonnes normalisées de l'analyse en composantes principales préalable.
- Scores and classes : les variables canoniques ou scores (combinaisons linéaires de variance unité et de variance inter maximales) et les groupes ou classes - les ellipses - qui donnent le mode de discrimination opérée.
- Cos(variates, canonical weigths) : les corrélations entre variables canoniques et les variables de départ. Si les graphes 1 et 3 ne sont pas cohérents, c'est l'indice d'une instabilité numérique qui remet en cause l'analyse.
- le graphe des valeurs propres.
- Cos(components, canonical variates) : les corrélations entre les variables canoniques et le composantes principales de l'analyse de départ. On peut ainsi savoir si la discrimination se fait dans la partie interprétable de l'analyse préliminaire (sinon il faut être méfiant, des variables discriminantes pouvant être non interprétables).
- Class scores : les moyennes des variables canoniques par classe.

On peut réaliser indépendamment chaque élément de ce graphe avec :

```
# Canonical weights = loadings = dis1$fa
s.arrow(dis1$fa)
# Cos(variates,canonical variates) = dis1$va
s.corcircle(dis1$va)
# Cos(components,canonical variates) = dis1$cp
s.corcircle(dis1$cp)
# Scores and classes = dis1$li et dis1$gc
s.class(dis1$li, iris$Species)
# Class scores = dis1$gc
s.label(dis1$gc)
```

Si on utilise l'analyse discriminante dans un but **prédictif**, utiliser de préférence la fonction **lda**() qui est centrée sur la question de l'affectation d'un individu à une classe : peut-on prédire à quelle classe appartient un individu dont on connaît les mesures ?

Pour illustrer cette prédiction sur le jeu de données iris, on divise au hasard le tableau de données en deux parties :

- la première (ref) pour chercher une fonction discriminante,
- la seconde (sup) pour déterminer l'espèce à l'aide de cette fonction.

On compare ensuite avec la fonction table() le résultat obtenu (espestim) et les vraies valeurs (espsup).

```
echa <- sample(1:150, 50)
tabref <- iris[echa, 1:4]
espref <- iris[echa, 5]
tabsup <- iris[-echa, 1:4]
espsup <- iris[-echa, 5]
lda2 <- lda(tabref, espref)
lda2
Call:
lda(tabref, espref)

Prior probabilities of groups:
    setosa versicolor  virginica
      0.38       0.30       0.32
```

```
Group means:
           Sepal.Length Sepal.Width Petal.Length
setosa         4.973684    3.321053     1.421053
versicolor     5.986667    2.733333     4.346667
virginica      6.593750    3.100000     5.481250
           Petal.Width
setosa       0.2315789
versicolor   1.3600000
virginica    2.0937500

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.3622915 -0.04877296
Sepal.Width   1.7110732 -1.80116597
Petal.Length -2.3565720  1.55899539
Petal.Width  -2.4100374 -3.84019148

Proportion of trace:
   LD1    LD2
0.9866 0.0134

espestim <- predict(lda2, tabsup)$class

table(espestim, espsup)
            espsup
espestim     setosa versicolor virginica
  setosa         31          0         0
  versicolor      0         33         1
  virginica       0          2        33
```

Conclusion : à partir d'un sous-ensemble de 50 observations tirées au hasard parmi les 150, on prédit presque sans erreur les variétés des 100 observations « inconnues ».

3. Analyses sur Variables Instrumentales

Les analyses sur variables instrumentales sont des **méthodes de couplage de tableaux dissymétriques**. Elles permettent de coupler un tableau Y de variables à expliquer, préalablement soumis à une analyse de type dudi (ACP, AFC…), à un tableau X de variables explicatives. En écologie on se place dans le cas de l'analyse d'un tableau d'observations floristiques ou faunistiques que l'on cherche à projeter sur un tableau de variables environnementales mesurées sur les mêmes observations. Lorsque Y est analysé par ACP, on parle d'**ACPVI**, encore appelée **RDA** (Redundancy Analysis). Lorsque Y est analysé par une AFC, c'est une **AFCVI**, encore appelée **CCA** (Canonical Correspondence Analysis). Toutes ces analyses peuvent être réalisées avec la même fonction **pcaiv()** d'ade4.

Remarque : les analyses inter et intra-classes (bca et wca) sont des cas particuliers d'analyses sur variables instrumentales, où le tableau X contient une seule variable qualitative. Dans ce cas, prédire Y par X revient à remplacer la valeur d'une variable pour un individu par la moyenne des individus de la même classe pour la même variable. L'analyse inter-classes est l'analyse de ce tableau de moyennes. Elle recherche des combinaisons des variables de Y maximisant la variance inter-classes. L'analyse intra-classes, qui étudie ce qui reste une fois enlevé l'effet de la variable qualitative est l'ACPVI orthogonale.

3.1. ACP sur variables instrumentales (ACPVI)

On utilisera le jeu de données **doubs**, sur le cours d'eau du même nom, qui contient pour 30 échantillons (en lignes) un tableau de relevés faunistiques fish (abondances de 27 espèces de poissons) et un tableau de variables environnementales env (mesures de 11 variables physicochimiques). Le tableau faunistique est analysé par une ACP centrée, puis couplé avec le tableau environnemental par une ACPVI.

```
# ACPVI ou Analyse des redondances
# Exemple Doubs : couplage entre l'ACP centrée du tableau faunistique poi
# et les variables de milieu (mil)

data(doubs)
poi <- doubs$fish
mil <- doubs$env
pcafau <- dudi.pca(poi, scale = F, scannf = F, nf = 2)
pcaivdoubs <- pcaiv(pcafau, mil, scannf = F, nf = 2)
pcaivdoubs

Principal Component Analysis with Instrumental Variables
call: pcaiv(dudi = pcafau, df = mil, scannf = F, nf = 2)
class: pcaiv dudi

$rank (rank)     : 11
$nf (axis saved) : 2

eigen values: 38.42 5.954 2.416 1.339 0.7431 ...

 vector length mode    content
 $eig   11     numeric eigen values
 $lw    30     numeric row weigths (from dudi)
 $cw    27     numeric col weigths (from dudi)

 data.frame nrow ncol content
 $Y         30   27   Dependant variables
 $X         30   11   Explanatory variables
 $tab       30   27   modified array (projected variables)

 data.frame nrow ncol content
 $c1        27   2    PPA Pseudo Principal Axes
 $as        2    2    Principal axis of dudi$tab on PAP
 $ls        30   2    projection of lines of dudi$tab on PPA
 $li        30   2    $ls predicted by X

 data.frame nrow ncol content
 $fa        12   2    Loadings (CPC as linear combinations of X
 $l1        30   2    CPC Constraint Principal Components
 $co        27   2    inner product CPC - Y
 $cor       11   2    correlation CPC - X
```

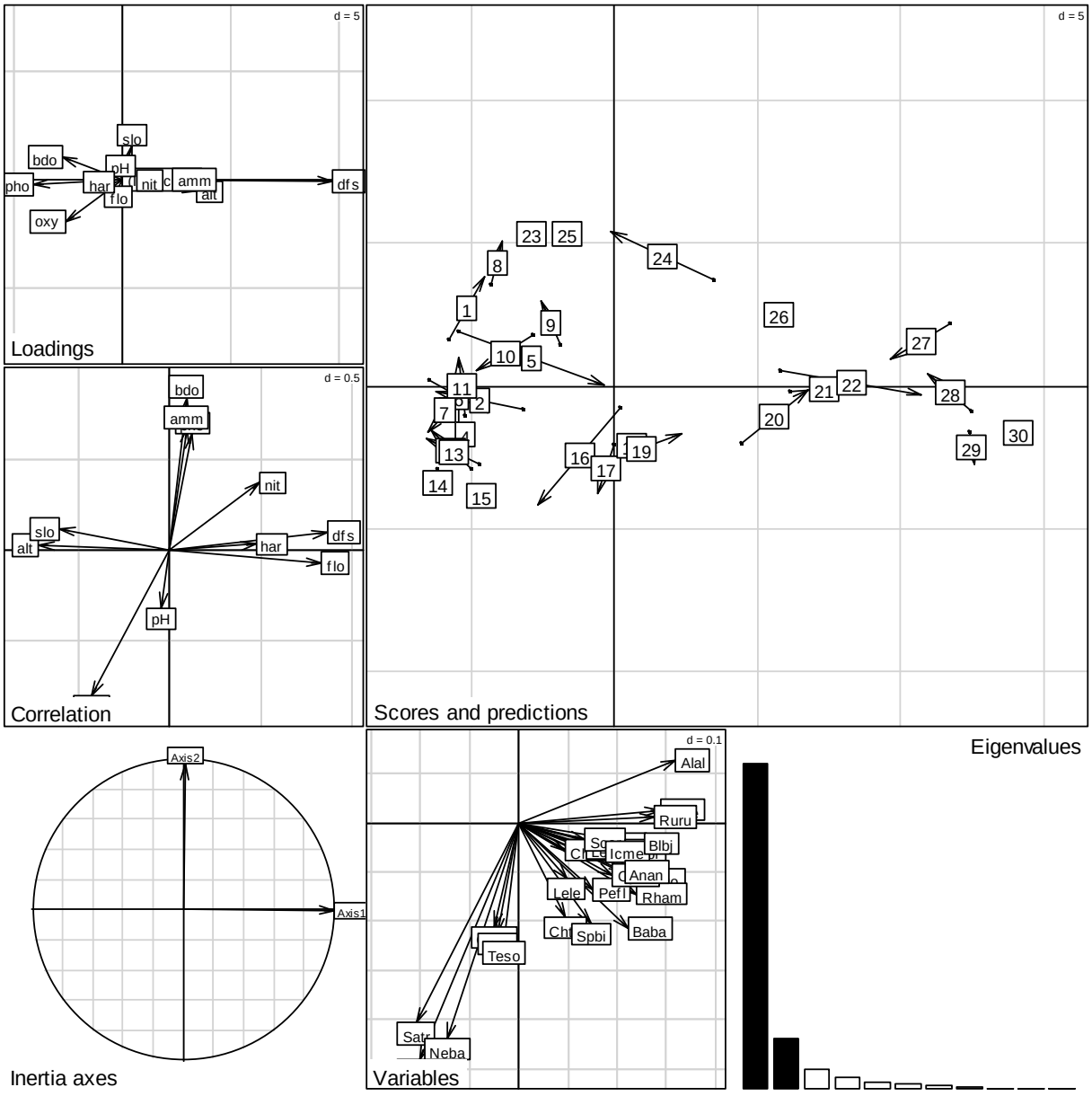
tab : le tableau PxY des modèles linéaires des colonnes de Y par X (variables projetées).
cw : le poids des colonnes provenant du dudi (1 pour chacune des m colonnes de Y).
lw : le poids des lignes provenant du dudi (1/n pour chacune des n colonnes de Y).

co : les corrélations entre les CPC et les variables de Y.

l1 : les composantes principales sous contrainte ou CPC (combinaisons linéaires de variables de X maximisant le critère de l'analyse).
eig : les valeurs propres, optimum du critère "somme des carrés des corrélations entre CPC et variables de Y".

c1 : les pseudo-axes principaux ou PAP, vecteurs normés de R_n.
ls : les coordonnées des projections des lignes de Y sur les PAP.
as : les coordonnées des projections des axes principaux (AP) de Y sur les PAP. Ceci permet de comparer les AP et les PAP. Les PAP ont une propriété d'optimalité originale.
li : les prédictions des coordonnées des projections des lignes de Y sur les PAP par régressions multiples sur X. Ces régressions définissent des carrés de corrélation multiple ou pourcentage de variance expliquée. On peut superposer ls (projections sur les PAP) et li (prédictions des positions).

```
plot(pcaivdoubs)
```



Si on veut réaliser indépendamment chaque élément de ce graphe :

```
# Loadings
```

```
s.arrow(pcaivdoubs$fa)

# Correlations
s.arrow(pcaivdoubs$cor)

# Scores and predictions
s.match(pcaivdoubs$li, pcaivdoubs$ls)

# Inertia axes
s.corcircle(pcaivdoubs$as)

# Variables
s.arrow(pcaivdoubs$c1)

# Eigenvalues
screeplot(pcaivdoubs)
```

Il existe deux possibilités pour interpréter une ACPVI. L'analyse recherche des coefficients (**fa**) des variables de X. La combinaison linéaire obtenue est une composante principale ou composante explicative (**l1**). La composante explicative maximise la somme des carrés de corrélations (si Y est analysé par une ACP normée) ou de covariances (dans le cas d'une ACP centrée) avec les variables de Y. Les colonnes de Y sont alors représentées par leurs corrélations ou covariances (**co**) avec la composante explicative. Les corrélations entre X et la composante explicative sont dans **cor**.

```
var(pcaivdoubs$l1)/30 * 29
             RS1          RS2
RS1 1.000000e+00 1.965969e-15
RS2 1.965969e-15 1.000000e+00

head(cov(poi, pcaivdoubs$l1)/30 * 29)
            RS1        RS2
Cogo -0.3010175 -0.5167752
Satr -1.2944349 -0.9924101
Phph -1.2484550 -1.1721456
Neba -0.9025339 -1.0704726
Thth -0.2711770 -0.5454216
Teso -0.1758249 -0.5916634

head(pcaivdoubs$co)
          Comp1      Comp2
Cogo -0.3010175 -0.5167752
Satr -1.2944349 -0.9924101
Phph -1.2484550 -1.1721456
Neba -0.9025339 -1.0704726
Thth -0.2711770 -0.5454216
Teso -0.1758249 -0.5916634

sum(pcaivdoubs$co[, 1]^2)
[1] 38.41774

pcaivdoubs$eig[1]
[1] 38.41774
```

La deuxième interprétation de l'ACPVI consiste à calculer un pseudo axe principal (**c1**). Les lignes de Y sont projetées sur les pseudo axes principaux. Ces projections **ls** sont des

combinaisons des variables de Y maximisant la variance expliquée par X. Les prédictions de ces projections par X sont contenues dans **li**

```
t(as.matrix(pcaivdoubs$c1)) %*% as.matrix(pcaivdoubs$c1)
              CS1           CS2
CS1  1.000000e+00 -3.712308e-16
CS2 -3.712308e-16  1.000000e+00

head(as.matrix(pcafau$tab) %*% as.matrix(pcaivdoubs$c1))
        CS1         CS2
1 -4.571026  3.84649959
2 -6.231231 -0.20450758
3 -6.544715 -1.60614068
4 -5.247418 -1.76428284
5 -0.365054  0.06064447
6 -4.432451 -0.93028165

head(pcaivdoubs$ls)
      Axis1       Axis2
1 -4.571026  3.84649959
2 -6.231231 -0.20450758
3 -6.544715 -1.60614068
4 -5.247418 -1.76428284
5 -0.365054  0.06064447
6 -4.432451 -0.93028165

lmprovi <- lm(pcaivdoubs$ls[, 1] ~ as.matrix(mil))
predict(lmprovi)[1:5]
        1         2         3         4         5
-5.789858 -3.213591 -5.071321 -5.186859 -5.450346

pcaivdoubs$li[1:5, 1]
[1] -5.789858 -3.213591 -5.071321 -5.186859 -5.450346

sum(predict(lmprovi)^2)/30
[1] 38.41774
```

L'ACPVI fournit donc un compromis entre l'analyse canonique (maximisation du carré de la corrélation multiple) et l'analyse en composantes principales (maximisation de la variance) en maximisant la variance expliquée (maximisation du produit).

```
summary(pcaivdoubs)
Principal component analysis with instrumental variables

Class: pcaiv dudi
Call: pcaiv(dudi = pcafau, df = mil, scannf = F, nf = 2)

Total inertia: 50.26

Eigenvalues:
    Ax1     Ax2     Ax3     Ax4     Ax5
38.4177  5.9540  2.4162  1.3387  0.7431

Projected inertia (%):
    Ax1     Ax2     Ax3     Ax4     Ax5
```

```
 76.441  11.847   4.808   2.664   1.478

Cumulative projected inertia (%):
    Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
  76.44   88.29   93.10   95.76   97.24

(Only 5 dimensions (out of 11) are shown)

Total unconstrained inertia (pcafau): 66.08

Inertia of pcafau explained by mil (%): 76.06

Decomposition per axis:
   iner inercum inerC inercumC ratio    R2 lambda
1 42.75    42.7 42.59     42.6 0.996 0.902  38.42
2  8.16    50.9  7.76     50.4 0.989 0.767   5.95
```

iner = valeurs propres de l'ACP simple
inercum = valeurs propres cumulées de l'ACP simple
lambda = valeurs propres de l'ACPVI
inerC = somme (pondérée par les lw) des carrés des coordonnées des lignes de l'ACPVI (ls)
inercumC = inerC cumulés
R2 = carré de corrélation multiple
ratio = rapport inerC / iner

L'analyse simple trouve des combinaisons des variables de Y de variance maximale (**iner et inercum** en cumulé). Les valeurs propres de l'ACPVI (**lambda**) sont des variances expliquées. Elles correspondent au produit de la variance (**inerC**) par le carré de la corrélation multiple (**R2**). Exemple pour l'axe 1 : 38.42 = 42.6 * 0.902

En maximisant un compromis (la variance expliquée), on rajoute une contrainte (prédiction par les variables de X) et la maximisation de la variance n'est donc plus optimale (elle l'est pour l'analyse simple). On mesure l'importance de cette contrainte par le **ratio** des variances des combinaisons des variables de Y des deux analyses :

```
pcafau$eig[1] # iner
[1] 42.74627

pcaivdoubs$eig[1] # lambda
[1] 38.41774

sum(pcaivdoubs$lw * pcaivdoubs$ls[, 1]^2) # inerC
[1] 42.59456

sum(pcaivdoubs$lw * pcaivdoubs$ls[, 1]^2)/pcafau$eig[1] # ratio inerC/iner
[1] 0.9964509
```

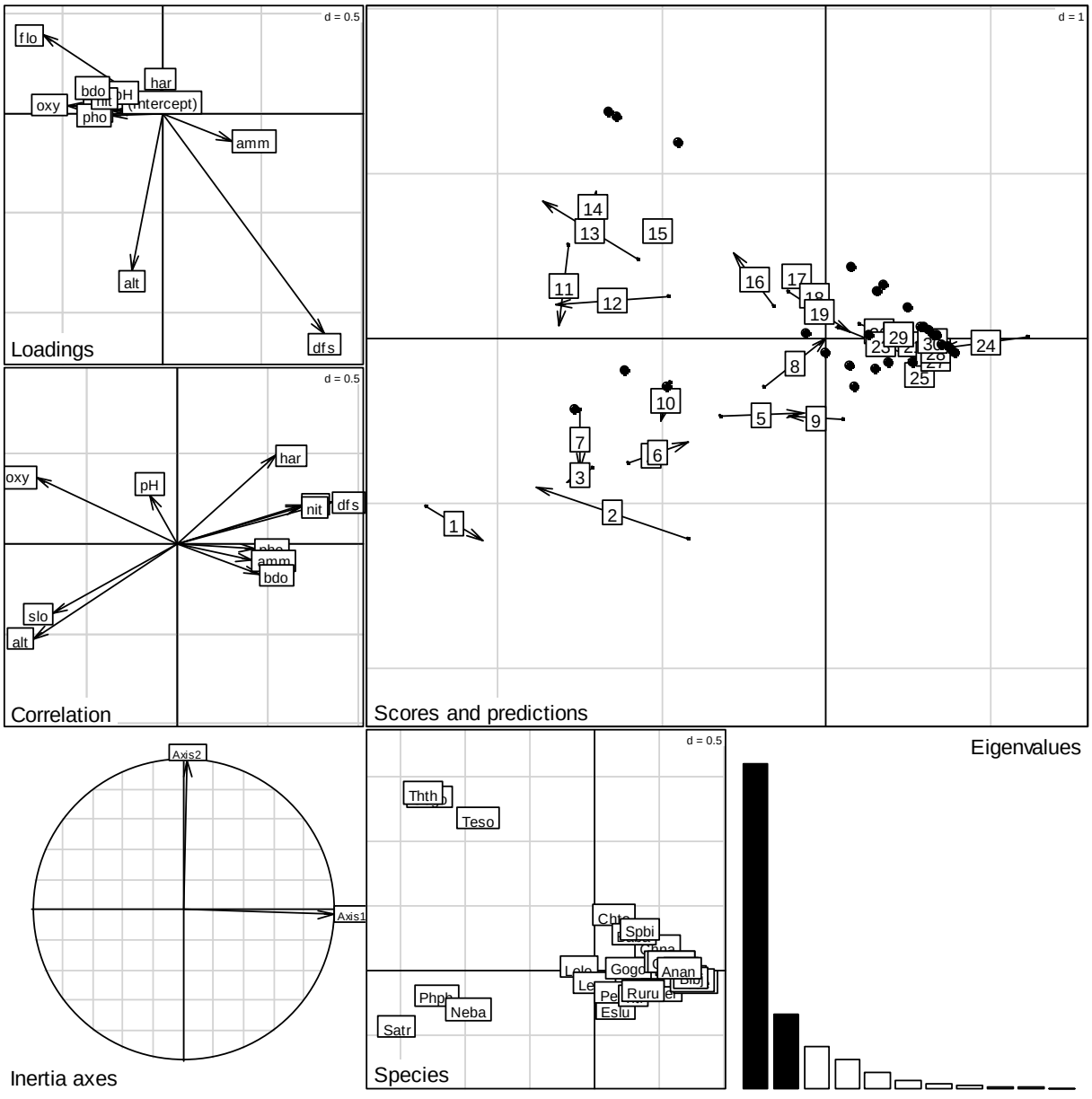
3.2. AFC sur variables instrumentales (AFCVI)

Dans l'ACFVI (ou CCA, Canonical Correspondence Analysis), l'analyse appliquée au tableau Y est une AFC. On procède alors au couplage de cette AFC avec le tableau de variables instrumentales. Pour illustrer cette analyse, on utilise comme pour l'ACPVI l'exemple du Doubs, avec une AFC du tableau faunistique. A cette différence près, le déroulement de l'analyse est identique à celui de l'ACPVI et son dépouillement également.

```
# AFCVI ou Analyse Canonique des Correspondances
# Exemple Doubs : couplage entre l'AFC du tableau fau
# et les variables de milieu (mil)

data(doubs)
poi <- doubs$fish
mil <- doubs$env

coafau <- dudi.coa(poi, scannf = F, nf = 2)
ccadoubs <- pcaiv(coafau, mil, scannf = F, nf = 2)
ccadoubs
Canonical correspondence analysis
call: pcaiv(dudi = coafau, df = mil, scannf = F, nf = 2)
class: caiv pcaiv dudi

$rank (rank)     : 11
$nf (axis saved) : 2

eigen values: 0.5345 0.1218 0.0687 0.04917 0.02709 ...

 vector length mode    content
 $eig   11     numeric eigen values
 $lw    30     numeric row weigths (from dudi)
 $cw    27     numeric col weigths (from dudi)

 data.frame nrow ncol
 $Y         30   27
 $X         30   11
 $tab       30   27
 content
 Dependant variables
 Explanatory variables
 modified array (projected variables)

 data.frame nrow ncol
 $c1        27   2
 $as        2    2
 $ls        30   2
 $li        30   2
 content
 PPA Pseudo Principal Axes
 Principal axis of dudi$tab on PAP
 projection of lines of dudi$tab on PPA
 $ls predicted by X

 data.frame nrow ncol
 $fa        12   2
 $l1        30   2
 $co        27   2
 $cor       11   2
 content
 Loadings (CPC as linear combinations of X
 CPC Constraint Principal Components
```

```
inner product CPC - Y
correlation CPC - X
```

```
plot(ccadoubs)
```



L'analyse recherche des coefficients ou loadings (**fa**) des variables de X. La combinaison linéaire obtenue est une composante principale sous contrainte (**l1**). C'est un score des relevés de variance unité, combinaison linéaire des variables de milieu. Les espèces (**co**) sont positionnées à la moyenne des relevés. L'analyse maximise la variance des moyennes conditionnelles par un double centrage. Cette vision est parfaitement adaptée à la vision de la **niche écologique** et des **gradients environnementaux** sur lesquels se séparent les niches des espèces. On réalise une **Analyse Canonique des Correspondances (CCA) selon Ter Braak, 1986**.

Il existe un deuxième point de vue qui consiste à calculer un pseudo axe principal (**c1**). Les lignes de Y (sites) sont projetées sur les pseudo-axes principaux et positionnés à la moyenne des espèces qu'ils contiennent **(ls).** Les prédictions de ces projections par X sont contenues dans **li**. Ce deuxième point de vue est celui de **l'AFCVI de Lebreton et al., 1991.**

4. Analyse de coinertie

L'analyse de **coinertie** (ou costructure) propose une approche symétrique du couplage de tableaux. Chacun des tableaux (Y et X) fait l'objet d'une analyse factorielle préalable de type dudi. L'analyse de coinertie est une approche unifiée qui regroupe entre autres méthodes **l'analyse inter-batterie** de Tucker, 1958 (couplage ACP-ACP), **l'analyse canonique sur variables explicatives** de Cazes, 1980 (couplage ACM-ACM) et **l'analyse des correspondances d'un tableau de profils écologiques** (Romane, 1972).

L'analyse de coinertie a été proposée par Dolédec et Chessel (1994) et par Dray, Chessel et Thioulouse (2003). Elle est disponible uniquement dans la librairie ade4 par la fonction **coinertia()**. Elle présente l'avantage de permettre le couplage de n'importe quel type d'analyses (par exemple le cas où les variables de X sont qualitatives). De plus elle n'est pas sensible à la proportion entre nombre de lignes et nombre de variables du tableau X, contrairement aux analyses sur variables instrumentales où un trop petit nombre d'observations par rapport au nombre de variables de X peut conduire à un résultat non fiable en raison de la surparamétrisation du modèle (comme en régression linéaire multiple).

Dolédec, S. and Chessel, D. (1994) Co-inertia analysis: an alternative method for studying species-environment relationships. *Freshwater Biology*, **31**, 277–294.

Dray, S., Chessel, D. and J. Thioulouse (2003) Co-inertia analysis and the linking of the ecological data tables. *Ecology*, **84**, 11, 3078–3089.

Dans la pratique, on commence par effectuer une analyse simple de type dudi (ACP, AFC, ACM) sur chaque tableau de données à coupler (Y et X), puis on couple les deux analyses en les appelant par **coinertia(dudiX, dudiY)**. Une contrainte à respecter est que les deux analyses doivent avoir la même pondération sur les lignes. Cela n'est pas un problème en ACP ou en ACM, mais dans le cas du couplage entre une AFC et une ACP, il faut imposer à l'ACP la pondération des lignes de l'AFC.

Voici d'après l'aide de la fonction coinertia la liste des éléments de l'objet de class coinertia généré par la fonction :

`nf` a numeric value indicating the number of kept axes

`RV` a numeric value, the RV coefficient

`eig` a numeric vector with all the eigenvalues

`lw` a numeric vector with the rows weigths (crossed table)

`cw` a numeric vector with the columns weigths (crossed table)

`tab` a crossed table (CT)

`li` CT row scores (cols of dudiY)

`l1` Principal components (loadings for cols of dudiY)

`co` CT col scores (cols of dudiX)

`c1` Principal axes (cols of dudiX)

`lX` Row scores (rows of dudiX)

`mX` Normed row scores (rows of dudiX)

`lY` Row scores (rows of dudiY)

`mY` Normed row scores (rows of dudiY)

`aX` Correlations between dudiX axes and coinertia axes

`aY` Correlations between dudiY axes and coinertia axes

Le coefficient 'RV' (Escoufier, 1973) est un coefficient de corrélation vectorielle entre les deux tableaux, extension du coefficient de corrélation entre deux variables. Il permet de juger de la ressemblance entre les deux tableaux. Le tableau croisé 'tab' donne les covariances entre les variables de Y et celles de X. C'est ce tableau qui est diagonalisé pour en extraire des axes, sur lesquels on pourra projeter les variables de X (co) et les variables de Y (li). Enfin on pourra projeter en éléments supplémentaires sur ces axes les lignes du tableau X (lX) et du tableau Y (lY) et visualiser graphiquement la ressemblance entre les deux structures.

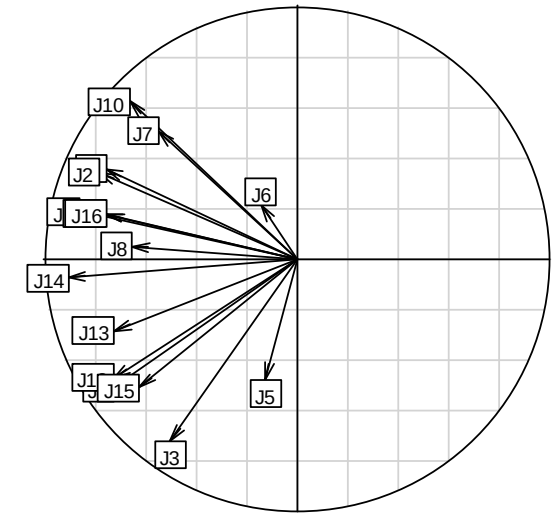
4.1 Couplage ACP-ACP

On prendra pour exemple le jeu de données **fruits** d'ade4, portant sur 28 lots de fruits (pêches et nectarines) jugés de deux manières différentes : un classement par ordre de préférence sans ex aequo par 16 juges, et 15 variables quantitatives décrivant chaque lot.

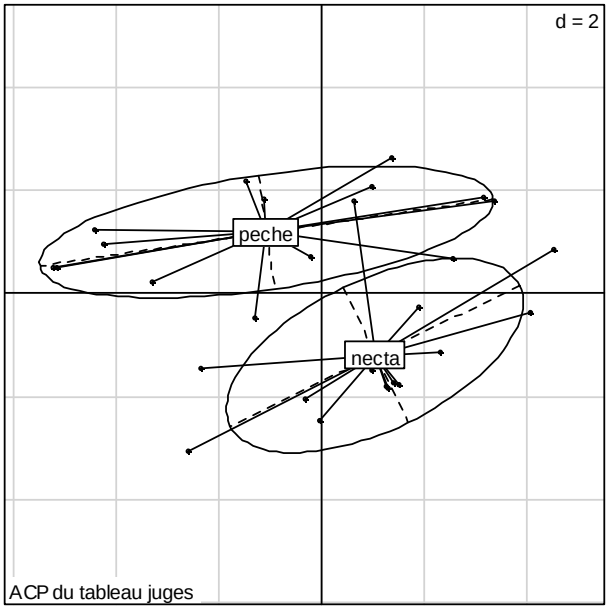
Ce jeu de données contient 3 éléments : **typ** est un vecteur de longueur 28 identifiant chaque lot, **jug** est un tableau donnant les notes des 16 juges aux 28 lots et **var** est un tableau à 15 colonnes décrivant les variables pour les 28 lots.

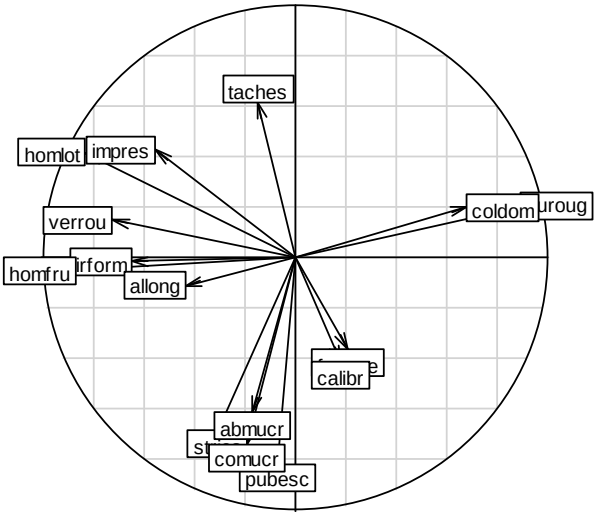
Sur chacun des deux tableaux jug et var on effectue d'abord une ACP normée, puis on couple ces deux ACP dans une analyse de coinertie.

```
data(fruits)
?fruits
pcajug <- dudi.pca(fruits$jug, scann = FALSE)
pcavar <- dudi.pca(fruits$var, scann = FALSE)

windows()
par(mfrow = c(2,2))
s.corcircle(pcajug$co, sub="ACP du tableau juges")
s.class(pcajug$li, fac = fruits$type, sub="ACP du tableau juges")
s.corcircle(pcavar$co, sub="ACP du tableau variables")
s.class(pcavar$li, fac = fruits$type, sub="ACP du tableau variables")
par(mfrow = c(1,1))
```

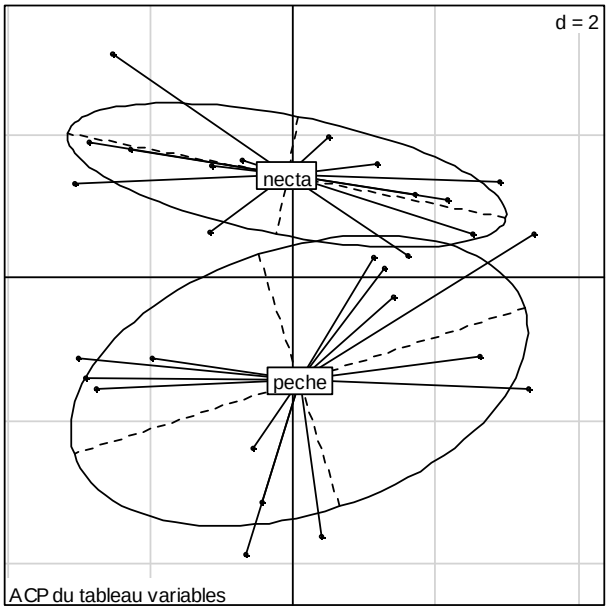


ACP du tableau juges





ACP du tableau variables




```
coi.fruits <- coinertia(pcajug, pcavar, scan = FALSE)
coi.fruits

Coinertia analysis
call: coinertia(dudiX = pcajug, dudiY = pcavar, scannf = FALSE)
class: coinertia dudi

$rank (rank)     : 15
$nf (axis saved) : 2
$RV (RV coeff)   : 0.4927474

eigenvalues: 15.13 5.704 2.728 0.8568 0.5648 ...

  vector length mode    content
1 $eig   15     numeric Eigenvalues
2 $lw    15     numeric Row weigths (for pcavar cols)
3 $cw    16     numeric Col weigths (for pcajug cols)
```

```
   data.frame nrow ncol content
1  $tab       15   16   Crossed Table (CT): cols(pcavar) x cols(pcajug)
2  $li        15   2    CT row scores (cols of pcavar)
3  $l1        15   2    Principal components (loadings for pcavar cols)
4  $co        16   2    CT col scores (cols of pcajug)
5  $c1        16   2    Principal axes (loadings for pcajug)
6  $lX        28   2    Row scores (rows of pcajug cols)
7  $mX        28   2    Normed row scores (rows of pcajug)
8  $lY        28   2    Row scores (rows of pcavar)
9  $mY        28   2    Normed row scores (rows of pcavar)
10 $aX        2    2    Corr pcajug axes / coinertia axes
11 $aY        2    2    Corr pcavar axes / coinertia axes

CT rows = cols of pcavar (15) / CT cols = cols of pcajug (16)

summary(coi.fruits)

Coinertia analysis

Class: coinertia dudi
Call: coinertia(dudiX = pcajug, dudiY = pcavar, scannf = FALSE)

Total inertia: 25.35

Eigenvalues:
    Ax1     Ax2     Ax3     Ax4     Ax5
15.1338  5.7037  2.7282  0.8568  0.5648

Projected inertia (%):
    Ax1     Ax2     Ax3     Ax4     Ax5
 59.690  22.496  10.761   3.379   2.228

Cumulative projected inertia (%):
    Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
  59.69   82.19   92.95   96.33   98.55

(Only 5 dimensions (out of 15) are shown)

Eigenvalues decomposition:
        eig    covar      sdX      sdY      corr
1 15.133835 3.890223 2.607581 1.864335 0.8002263
2  5.703734 2.388249 1.550666 1.776134 0.8671329

Inertia & coinertia X (pcajug):
    inertia      max     ratio
1  6.799477 7.318882 0.9290322
12 9.204041 9.930650 0.9268317

Inertia & coinertia Y (pcavar):
    inertia      max     ratio
1  3.475745 4.391663 0.7914416
12 6.630397 7.620306 0.8700960
```

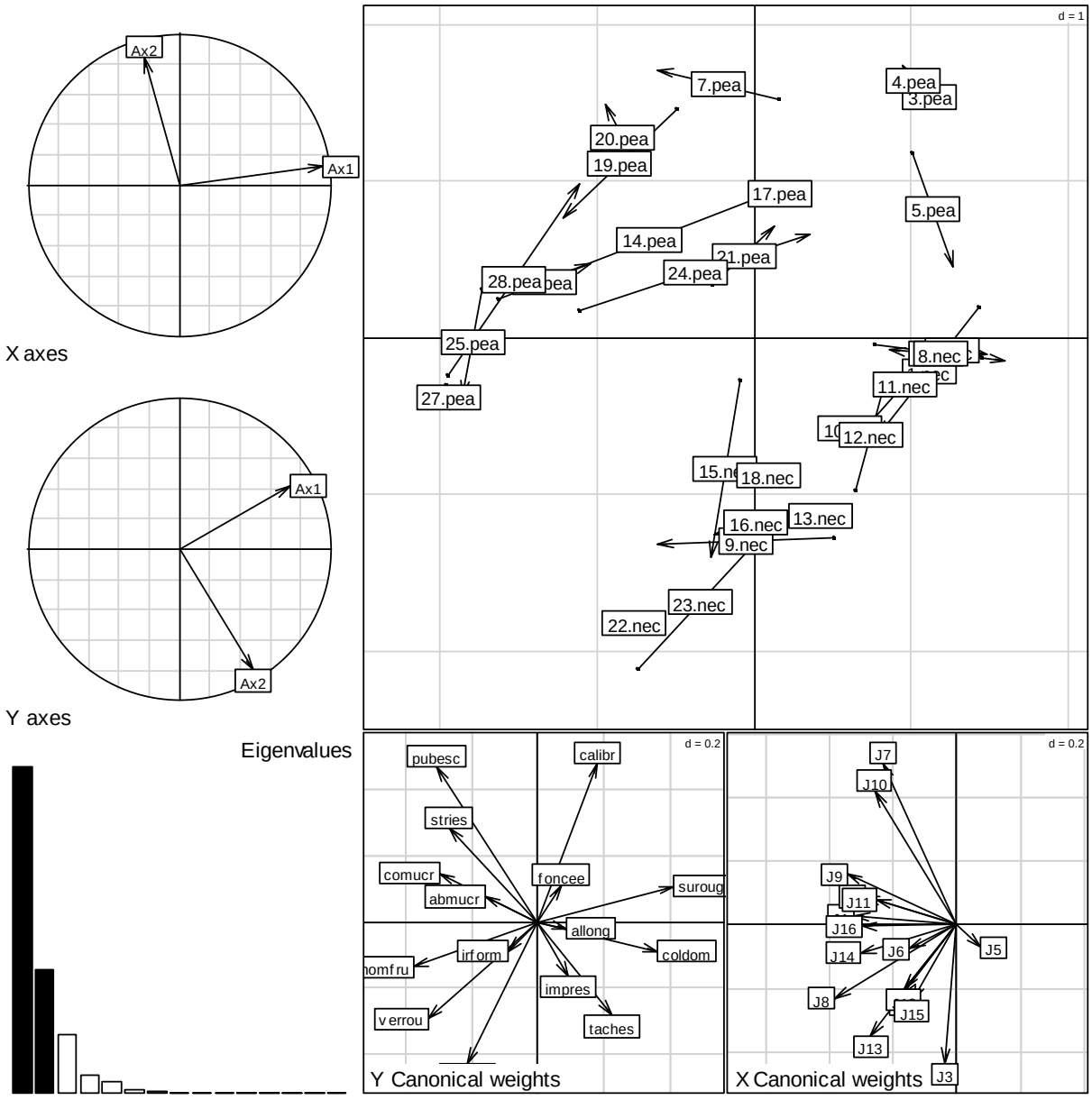
```
RV:
 0.4927474
```

La première valeur propre de la coinertie vaut 15.134, pour une covariance de 3.89, qui est le produit d'une corrélation de 0.8002 par les deux écarts types dans sdX et sdY, respectivement 2.607 et 1.864. La corrélation (entre 0 et 1) exprime la ressemblance entre les deux ACP de départ. Les tableaux 'Inertia & coinertia X' et 'Inertia & coinertia Y' permettent de donner le ratio d'inertie projetée sur le premier axe et sur le plan 1-2 pour X et pour Y.

Dans le cas du couplage de deux ACP normées, on peut retrouver le coefficient RV par :

```
sum(cor(pcajug$tab, pcavar$tab)^2)/sqrt(sum(cor(pcajug$tab, pcajug$tab)^2)
*sum(cor(pcavar$tab, pcavar$tab)^2))
[1] 0.4927474
```

```
plot(coi.fruits)
```



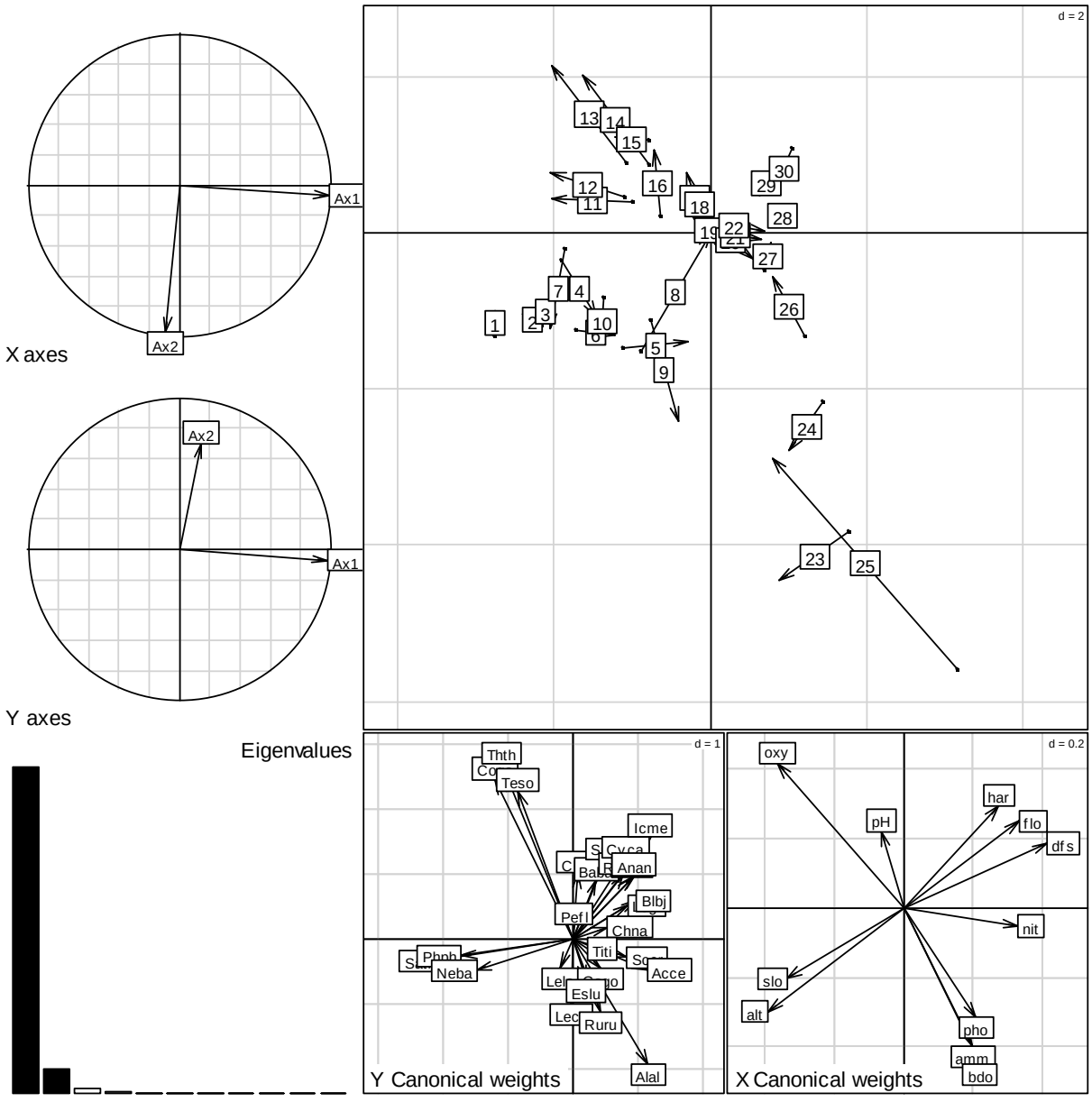
4.2 Couplage AFC-ACP

Le couplage AFC-ACP nécessite d'imposer une pondération des lignes dans l'ACP à partir du vecteur de pondérations de l'AFC. On revient sur le jeu de données sur le doubs déjà analysé en AFCVI. L'analyse est dépouillée comme précédemment

```
# Analyse de coinertie ----

coafau <- dudi.coa(poi, scannf = F, nf = 2)
pcamil <- dudi.pca(mil, row.w = coafau$lw, scannf = F, nf = 2)
coidoubs <- coinertia(pcamil, coafau, scannf = F, nf = 2)
coidoubs
Coinertia analysis
call: coinertia(dudiX = pcamil, dudiY = coafau, scannf = F, nf = 2)
class: coinertia dudi

$rank (rank)     : 11
$nf (axis saved) : 2
$RV (RV coeff)   : 0.636319

eigenvalues: 2.342 0.175 0.03947 0.01908 0.00658 ...

  vector length mode
1 $eig   11     numeric
2 $lw    27     numeric
3 $cw    11     numeric
  content
1 Eigenvalues
2 Row weigths (for coafau cols)
3 Col weigths (for pcamil cols)

   data.frame nrow ncol
1  $tab       27   11
2  $li        27   2
3  $l1        27   2
4  $co        11   2
5  $c1        11   2
6  $lX        30   2
7  $mX        30   2
8  $lY        30   2
9  $mY        30   2
10 $aX        2    2
11 $aY        2    2
   content
1  Crossed Table (CT): cols(coafau) x cols(pcamil)
2  CT row scores (cols of coafau)
3  Principal components (loadings for coafau cols)
4  CT col scores (cols of pcamil)
5  Principal axes (loadings for pcamil)
6  Row scores (rows of pcamil cols)
7  Normed row scores (rows of pcamil)
8  Row scores (rows of coafau)
```

```
9  Normed row scores (rows of coafau)
10 Corr pcamil axes / coinertia axes
11 Corr coafau axes / coinertia axes

CT rows = cols of coafau (27) / CT cols = cols of pcamil (11)
```

```
plot(coidoubs)
```




```
summary(coidoubs)

Coinertia analysis

Class: coinertia dudi
Call: coinertia(dudiX = pcamil, dudiY = coafau, scannf = F, nf = 2)

Total inertia: 2.59
```

```
Eigenvalues:
    Ax1     Ax2     Ax3     Ax4     Ax5
2.34163 0.17496 0.03947 0.01908 0.00658

Projected inertia (%):
    Ax1     Ax2     Ax3     Ax4     Ax5
90.4004  6.7545  1.5238  0.7367  0.2540

Cumulative projected inertia (%):
    Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
  90.40   97.15   98.68   99.42   99.67

(Only 5 dimensions (out of 11) are shown)

Eigenvalues decomposition:
        eig     covar      sdX       sdY      corr
1 2.3416297 1.5302384 2.366115 0.7591393 0.8519259
2 0.1749618 0.4182844 1.533416 0.3336151 0.8176473

Inertia & coinertia X (pcamil):
    inertia      max     ratio
1  5.598498 5.727595 0.9774606
12 7.949863 8.153563 0.9750170

Inertia & coinertia Y (coafau):
     inertia       max     ratio
1  0.5762925 0.6009926 0.9589011
12 0.6875916 0.7453635 0.9224915

RV:
 0.636319
```

Bibliographie

Fichiers PDF

- **tdr621_ACP_Inter_Intra.pdf** *(A.B. Dufour)*
- **tdr63_Analyses_Discriminantes.pdf** *(D. Chessel, A.B. Dufour & J. Thioulouse)*
- **tdr65_VI.pdf** *(A.B. Dufour, D. Chessel & J. Thioulouse)*
- **tdr64_coinertie.pdf** *(D. Chessel A.B. Dufour & S. Dray)*
- **stage5_couplage_tableaux.pdf** *(D. Chessel, A.B. Dufour & J. Thioulouse)*

http://pbil.univ-lyon1.fr/ADE-4 : all the pdf files can be downloaded from this website