Fiche TD avec le logiciel  : tdr1103

Initiation à l'analyse en composantes principales

A.B. Dufour & D. Clot

Une première approche très intuitive et interactive de l'ACP.

Table des matières

1 Introduction

Les méthodes d'analyse de données sont une branche à part entière de la statistique. Nous proposons ici une introduction, très sommaire, à l'une d'entre elles, l'analyse en composantes principales (ACP). Plus précisément, nous n'utiliserons qu'une des variantes : l'ACP centrée et réduite appelée aussi ACP normée. L'objectif est de comprendre les différents objets créés par la fonction du package ade4 de R. Puis, nous étudierons, par l'exemple, le rôle des pondérations associées aux lignes d'un tableau à analyser.

2 Les données

Les données représentent les dépenses de santé, pour une mutuelle, dans trois secteurs : maladie, dentaire et optique. Afin de ne pas nuire à leur confidentialité, 20 classes d'âge ont été conservées sur l'ensemble et les dépenses affichées sont les moyennes pour chacune d'elles.

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

```
[1] "age"      "groupe"   "effectif" "maladie"  "dentaire" "optique"
```

La colonne `effectif` contient le nombre d'individus du tableau original sur lequel a porté le calcul de la moyenne. Elle nous servira à discuter de la pondération des lignes.

Les âges ont été répartis en trois groupes : jeunes, actifs, anciens. Nous récupérons cette variable qualitative dans un vecteur, et définissons un vecteur de couleurs pour les distinguer facilement :

```
groupe <- depsante$groupe
summary(groupe)
```

```
actifs anciens  jeunes
     8       8       4
```

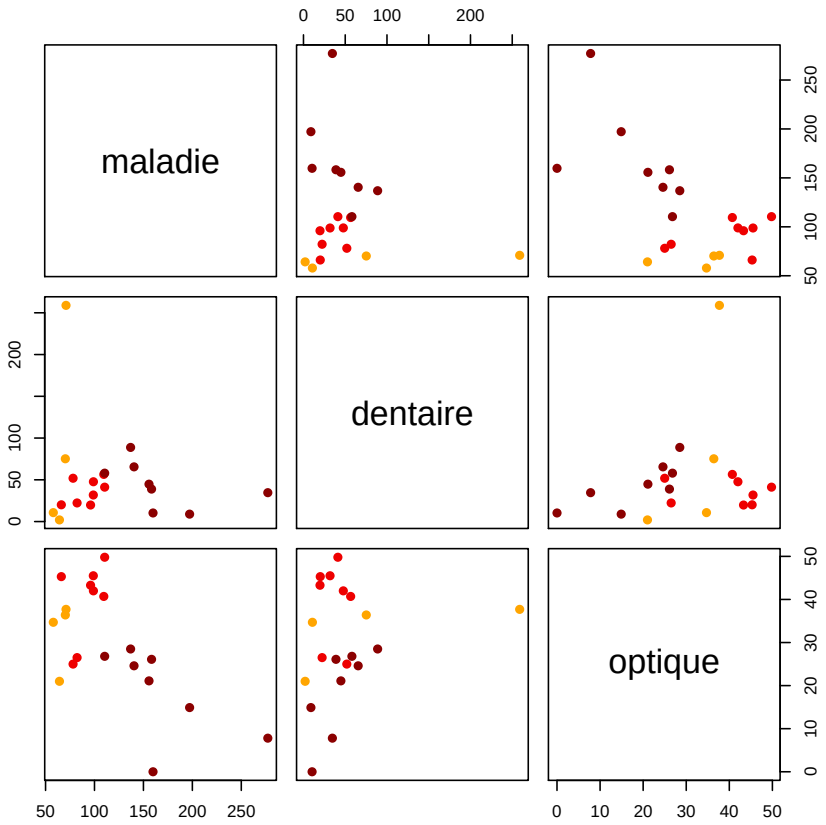
```
couleur <- rep(c("orange","red2","red4"),c(4,8,8))
```

Nous extrayons les trois variables quantitatives de ce jeu de données c'est-à-dire les dépenses dans les secteurs maladie, dentaire et optique. Toutes sont exprimées en euros.

```
mesures <- depsante[,4:6]
names(mesures)
```

```
[1] "maladie"  "dentaire" "optique"
```

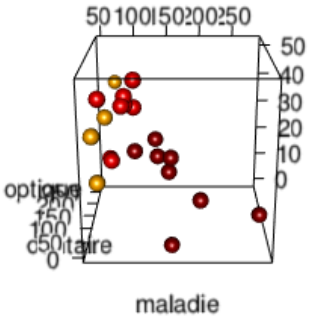
```
plot(mesures, col = couleur, pch=19)
```



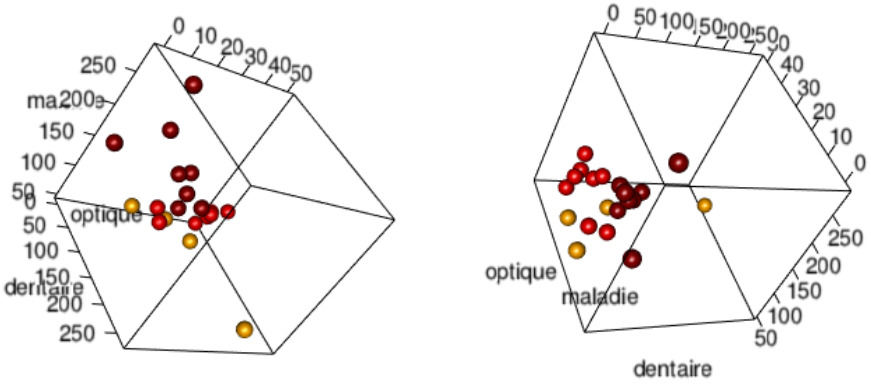
3 Visualisation des données en trois dimensions

Chaque individu est caractérisé par trois variables, soit par un point dans $\mathbb{R}^3$. La fonction `plot3d()` de la librairie `rgl` [1] vous permet d'explorer facilement un nuage de points en 3 dimensions.

```
library(rgl)
plot3d(mesures, type = "s", col = couleur)
```



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



4 Centrage et réduction

Les représentations graphiques précédentes sont trompeuses parce que nous n'avons pas utilisé la même échelle en x, y et z.

```
colMeans(mesures)
```

```
maladie dentaire  optique
116.950   49.395   29.885
```

```
varn <- function(x) var(x)*(length(x)-1)/length(x)
sapply(mesures,varn)
```

```
  maladie  dentaire   optique
2759.8585 2830.5495  164.9503
```

Les moyennes et les variances sont très différentes d'une dépense à l'autre. L'optique semble jouer un rôle à part. Pour éviter de telles différences, il est opportun de donner à chaque variable une même importance. Pour ce faire, nous réalisons un centrage et une réduction sur les données. La fonction `scalewt()` d'`ade4` permet d'effectuer directement cette opération :

```
library(ade4)
mesures.cr <- scalewt(mesures, center=TRUE, scale=TRUE)
class(mesures.cr)
```

```
[1] "matrix"
```

```
mesurescr <- as.data.frame(mesures.cr)
sapply(mesurescr,mean)
```

```
     maladie      dentaire       optique
1.100194e-17 -8.082185e-17 -9.998512e-17
```

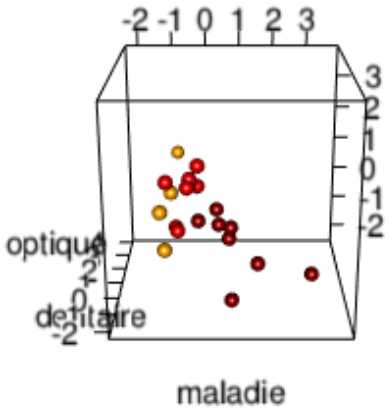
```
round(sapply(mesurescr,mean),2)
```

```
maladie dentaire  optique
      0        0        0
```

```
sapply(mesurescr,varn)
```

```
maladie dentaire  optique
      1        1        1
```

```
lims <- c(min(mesurescr),max(mesurescr))
plot3d(mesurescr, type = "s", col = couleur, xlim = lims, ylim = lims,
       zlim = lims)
```

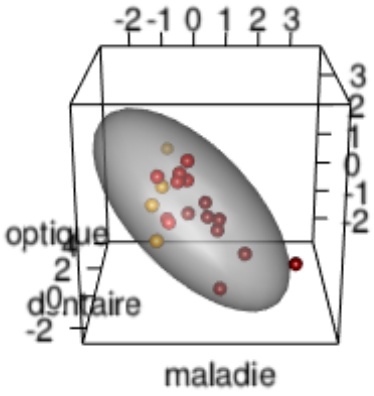


Quand une ACP normée est réalisée, les données sont donc centrées et réduites. Il est important de bien comprendre à quoi correspondent ces opérations.

5 La forme générale du nuage

Dans l'exemple que nous avons choisi, la forme générale du nuage de points est celle d'une dragée (le terme technique est un ellipsoïde).

```
plot3d(mesurescr, type = "s", col = couleur, xlim = lims, ylim = lims,
       zlim = lims)
plot3d( ellipse3d(cor(mesurescr)), col="grey", alpha=0.5, add = TRUE)
```



Une dragée est définie par ses trois axes :

1. Le premier axe correspondant au plus grand diamètre de l'ellipsoïde, la longueur de la dragée.
2. Le deuxième axe correspondant au diamètre moyen de l'ellipsoïde, la largeur de la dragée.

3. Le troisième axe correspondant au plus petit diamètre de l'ellipsoïde, l'épaisseur de la dragée.

Faites tourner le graphique précédent pour représenter le nuage de points dans le plan des axes (1,2), puis (1,3) puis (2,3).
Si vous aviez à choisir entre les trois plans que nous avons envisagés ci-dessus, vous n'hésiteriez pas à prendre le plan défini par les deux premiers axes parce que c'est dans cette représentation que nous avons le moins de perte d'information par rapport au nuage de points dans $\mathbb{R}^3$: c'est dans cette projection que les points sont les plus étalés dans le plan (on dit aussi que l'on a conservé le maximum possible de l'inertie initiale du nuage de points). Ce faisant, vous avez en fait réalisé une ACP à la main.

6 ACP centrée-réduite dans `ade4`

6.1 Calculs

Utiliser la fonction `dudi.pca()` de la librairie `ade4` [2] pour exécuter une ACP centrée réduite :

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

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

Nous avons utilisé ici les options `scannf = 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 quelle fraction de la variance totale est prise en compte par les axes successifs. Essayer avec :

```
acp <- dudi.pca(mesures, center=TRUE, scale=TRUE)
```

Répondre *3* à la question `"Select the number of axes:"`. Dans la pratique, seul un nombre restreint d'axes est conservé. Pour des raisons pédagogiques, les trois axes ont été conservés dans la présentation.
L'objet renvoyé par la fonction `dudi.pca()` est très riche. Nous allons examiner tous ses composants un à un.

`acp$tab`

Le data frame `acp$tab` contient les données du tableau initial après centrage et réduction.

```
head(acp$tab)
```

```
     maladie   dentaire    optique
1 -1.0041048 -0.8908348 -0.6918006
2 -0.8898939  0.4850299  0.5072685
3 -0.8765692  3.9378486  0.6084886
4 -1.1240264 -0.7273099  0.3749037
5 -0.9679380 -0.5506278  1.2002370
6 -0.3987867 -0.5562666  1.0445137
```

```
head(mesurescr)
```

```
     maladie   dentaire    optique
1 -1.0041048 -0.8908348 -0.6918006
2 -0.8898939  0.4850299  0.5072685
3 -0.8765692  3.9378486  0.6084886
4 -1.1240264 -0.7273099  0.3749037
5 -0.9679380 -0.5506278  1.2002370
6 -0.3987867 -0.5562666  1.0445137
```

acp$cw

Le vecteur `acp$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
```

La métrique associée aux poids des colonnes est dite **canonique**.

acp$lw

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

```
 head(acp$lw)
[1] 0.05 0.05 0.05 0.05 0.05 0.05
 head(acp$lw)*nrow(mesures)
[1] 1 1 1 1 1 1
```

La métrique associée aux poids des lignes est dite **uniforme**.

acp$eig

Le vecteur `acp$eig` donne les valeurs propres (*eigen values*) de la plus petite des matrices à diagonaliser.

```
 acp$eig
[1] 1.7371612 0.9015541 0.3612847
 sum(acp$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] 57.90537 30.05180 12.04282
 cumsum(pve)
[1]  57.90537  87.95718 100.00000
```

Notons que cette information s'obtient directement par la fonction `summary()` :

```
 summary(acp)
Class: pca dudi
Call: dudi.pca(df = mesures, center = TRUE, scale = TRUE, scannf = FALSE,
    nf = 3)
Total inertia: 3

Eigenvalues:
    Ax1     Ax2     Ax3
 1.7372  0.9016  0.3613

Projected inertia (%):
    Ax1     Ax2     Ax3
  57.91   30.05   12.04

Cumulative projected inertia (%):
    Ax1   Ax1:2   Ax1:3
  57.91   87.96  100.00
```

Dans notre exemple, le premier axe factoriel extrait 57.9 % de l'inertie totale, le deuxième axe factoriel 30.1 % de l'inertie totale. Le premier plan factoriel représente donc 88 % de l'inertie initiale. Ceci signifie que lorsque nous projetons le nuage de points initial dans $\mathbb{R}^3$ sur le plan défini par les deux premiers axes factoriels, nous avons perdu peu d'information.

acp$rank

Cet entier donne le rang (*rank*) de la matrice diagonalisée, dans notre cas le nombre de composantes principales.

```
acp$rank
```

```
[1] 3
```

```
bismesures <- cbind(mesures,mesures)
head(bismesures)
```

```
  maladie dentaire optique maladie dentaire optique
1    64.2      2.0    21.0    64.2      2.0    21.0
2    70.2     75.2    36.4    70.2     75.2    36.4
3    70.9    258.9    37.7    70.9    258.9    37.7
4    57.9     10.7    34.7    57.9     10.7    34.7
5    66.1     20.1    45.3    66.1     20.1    45.3
6    96.0     19.8    43.3    96.0     19.8    43.3
```

```
colnames(bismesures) <- c("maladie1","dentaire1","optique1",
                          "maladie2","dentaire2","optique2")
dudi.pca(bismesures,scann=F,n=3)$rank
```

```
[1] 3
```

acp$nf

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

```
acp$nf
```

```
[1] 3
```

acp$c1

`acp$c1` donne les coordonnées des variables (colonnes). Les vecteurs sont de norme unité :

```
acp$c1
```

```
                CS1        CS2         CS3
maladie   0.6582899 -0.2804235 -0.69858221
dentaire -0.3456279 -0.9370149  0.05044165
optique  -0.6687270  0.2082443 -0.71374963
```

```
sapply(1:3, function(x) sum(acp$cw*acp$c1[,x]^2))
```

```
[1] 1 1 1
```

acp$l1

`acp$l1` donne les coordonnées des individus (lignes). Les vecteurs sont de norme unité :

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

```
          RS1        RS2        RS3
1  0.08310309  1.0239436  1.9137334
2 -0.82902915 -0.1045783  0.4726025
3 -1.77917665 -3.4937239  0.6266807
4 -0.56089293  1.1319356  0.8001581
5 -0.94801891  1.0924901 -0.3464807
6 -0.58326421  0.8958098 -0.8235236
```

```
sapply(1:3, function(x) sum(acp$lw*acp$l1[,x]^2))
```

```
[1] 1 1 1
```

acp$co

`acp$co` donne les coordonnées des variables (colonnes). Les vecteurs sont normés à la racine carrée de la valeur propre correspondante :

```
 acp$co
             Comp1      Comp2       Comp3
maladie   0.8676353 -0.2662627 -0.41989654
dentaire -0.4555425 -0.8896976  0.03031894
optique  -0.8813916  0.1977284 -0.42901321
 sapply(1:3, function(x) sum(acp$cw*acp$co[,x]^2))
[1] 1.7371612 0.9015541 0.3612847
 acp$eig
[1] 1.7371612 0.9015541 0.3612847
```

Le lien entre les `acp$c1` et les `acp$co` s'obtient par :
• pour le premier axe,

```
 acp$c1$CS1 * sqrt(acp$eig[1])
[1]  0.8676353 -0.4555425 -0.8813916
```

• pour tous les axes,

```
 t(t(acp$c1) * sqrt(acp$eig))
                CS1        CS2         CS3
maladie   0.8676353 -0.2662627 -0.41989654
dentaire -0.4555425 -0.8896976  0.03031894
optique  -0.8813916  0.1977284 -0.42901321
```

acp$li

`acp$li` donne les coordonnées des individus (lignes). Les vecteurs sont normés à la racine carrée de la valeur propre correspondante :

```
 head(acp$li)
       Axis1       Axis2      Axis3
1  0.1095310  0.97223653  1.1502870
2 -1.0926721 -0.09929729  0.2840670
3 -2.3449799 -3.31729801  0.3766787
4 -0.7392648  1.07477518  0.4809507
5 -1.2495023  1.03732155 -0.2082590
6 -0.7687505  0.85057328 -0.4949950
 sapply(1:3, function(x) sum(acp$lw*acp$li[,x]^2))
[1] 1.7371612 0.9015541 0.3612847
 acp$eig
[1] 1.7371612 0.9015541 0.3612847
```

Le lien entre les `acp$l1` et les `acp$li` s'obtient par :
• pour le premier axe,

```
 head(acp$l1$RS1 * sqrt(acp$eig[1]))
[1]  0.1095310 -1.0926721 -2.3449799 -0.7392648 -1.2495023 -0.7687505
```

• pour tous les axes,

```
 head(t(t(acp$l1) * sqrt(acp$eig)))
         RS1         RS2        RS3
1  0.1095310  0.97223653  1.1502870
2 -1.0926721 -0.09929729  0.2840670
3 -2.3449799 -3.31729801  0.3766787
4 -0.7392648  1.07477518  0.4809507
5 -1.2495023  1.03732155 -0.2082590
6 -0.7687505  0.85057328 -0.4949950
```

acp$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 = mesures, center = TRUE, scale = TRUE, scannf = FALSE,
    nf = 3)
```

La fonction `eval()` permet de refaire les mêmes calculs :

```
eval(acp$call)
```

```
Duality diagramm
class: pca dudi
$call: dudi.pca(df = mesures, center = TRUE, scale = TRUE, scannf = FALSE,
    nf = 3)
$nf: 3 axis-components saved
$rank: 3
eigen values: 1.737 0.9016 0.3613
  vector length mode    content
1 $cw    3      numeric column weights
2 $lw    20     numeric row weights
3 $eig   3      numeric eigen values

  data.frame nrow ncol content
1 $tab       20   3    modified array
2 $li        20   3    row coordinates
3 $l1        20   3    row normed scores
4 $co        3    3    column coordinates
5 $c1        3    3    column normed scores
other elements: cent norm
```

```
identical(eval(acp$call), acp)
```

```
[1] TRUE
```

acp$cent

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

```
acp$cent
```

```
maladie dentaire  optique
 116.950   49.395   29.885
```

```
colMeans(mesures)
```

```
maladie dentaire  optique
 116.950   49.395   29.885
```

acp$norm

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

```
acp$norm
```

```
 maladie dentaire  optique
52.53436 53.20291 12.84330
```

```
var.n <- function(x) sum((x-mean(x))^2)/length(x)
sd.n <- function(x) sqrt(var.n(x))
apply(mesures, 2, sd.n)
```

```
 maladie dentaire  optique
52.53436 53.20291 12.84330
```

Exercice : dé-réduction et dé-centrage

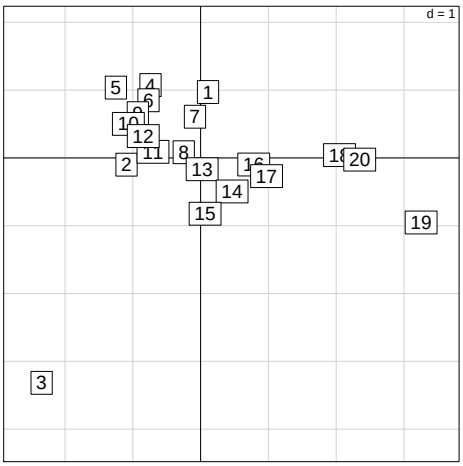
Si vous avez bien compris en quoi consiste l'opération de centrage et réduction, vous devez pouvoir être capables de faire l'opération inverse. À partir des objets `acp$tab`, `acp$cent` et `acp$norm`, reconstituez les données de départ, placez le résultat dans l'objet `recon`.

6.2 Représentations graphiques dans `ade4`

6.2.1 Représentation des individus

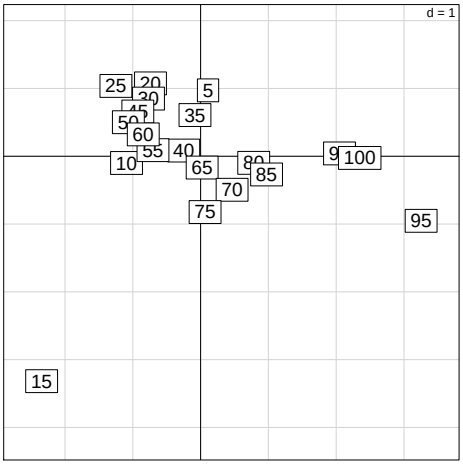
La fonction `s.label()` permet de représenter les individus sur les différents plans factoriels, par exemple sur le premier plan factoriel :

```
s.label(acp$li, xax = 1, yax = 2, clabel=1.5)
```



Afin de bien interpréter les données, il est préférable d'utiliser comme étiquette d'un individu l'âge moyen.

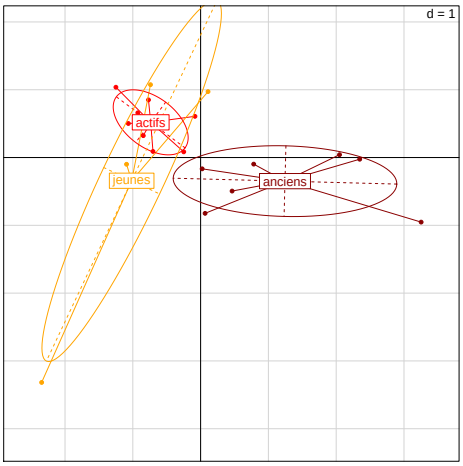
```
s.label(acp$li, xax = 1, yax = 2, label=as.character(depsante$age), clabel=1.5)
```



Exercice. Faire les représentations dans les plans (1,2), (1,3) et (2,3) avec une échelle commune pour tous les graphiques.

La fonction `s.class()` permet de porter en information supplémentaire une variable qualitative définissant des groupes d'individus, par exemple :

```
gcol <- c("red1", "red4","orange")
s.class(dfxy = acp$li, fac = depsante$groupe, col = gcol, xax = 1, yax = 2)
```

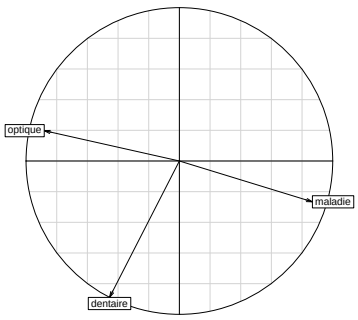


Exercice. Faire les représentation dans les trois plans factoriels avec une échelle commune pour tous les graphiques.

6.2.2 Représentation des variables

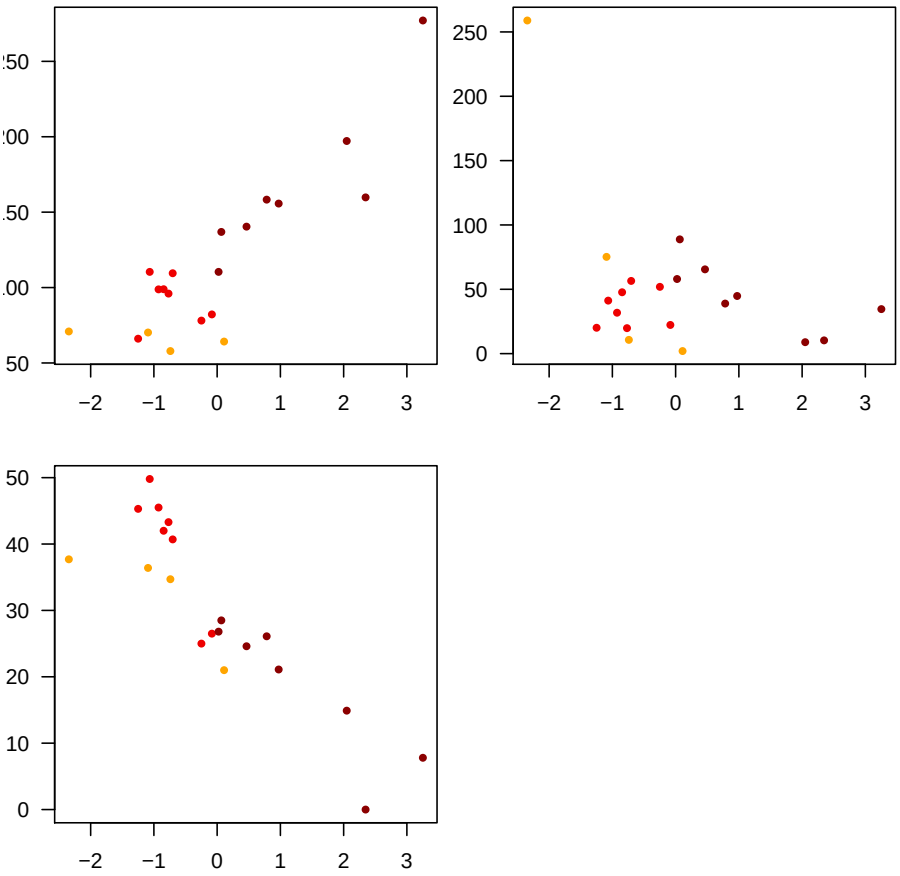
La fonction `s.corcircle()` représente les variables initiales dans le nouvel espace. Cette représentation est appelée cercle des corrélations :

```
s.corcircle(acp$co, xax = 1, yax = 2)
```



Sur le premier facteur de l'ACP, nous notons une opposition entre les secteurs optique et maladie. Le second facteur est associé au secteur dentaire.

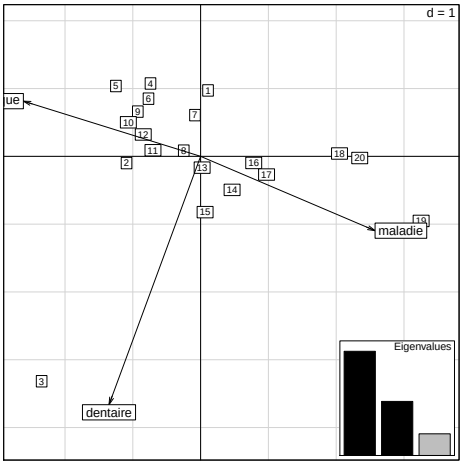
```
par(mfrow=c(2,2))
par(mar=c(2,2,2,1))
graphe <- function (i) plot(x = acp$li[,1], y = mesures[,i], pch = 20, col = couleur,
xlab = "Axe 1", las = 1,
ylab = colnames(mesures)[i])
sapply(1:3,graphe)
```



6.2.3 Représentation simultanée des individus et des variables

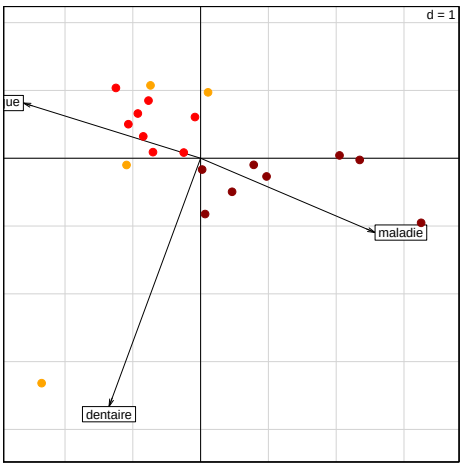
La fonction `scatter()` permet de représenter simultanément les individus et les variables. C'est une fonction générique associée à un objet de la classe `dudi`.

```
scatter(acp,posieig="bottomright")
```



Enrichir le graphique en portant l'information sur les groupes :

```
scatter(acp, clab.row = 0, posieig = "none")
s.class(acp$li,groupe,col=gcol, add.plot = TRUE,
cstar = 0, clabel = 0, cellipse = 0, cpoint=2)
```



Faire le lien avec ce qui avait été obtenu à la main avec la fonction `plot3d()`, notez en particulier comment les axes de la base initiale se projettent sur le premier plan factoriel.

7 Changement de pondération sur les individus

Dans une analyse en composantes principales classique, la pondération associée aux individus est uniforme.

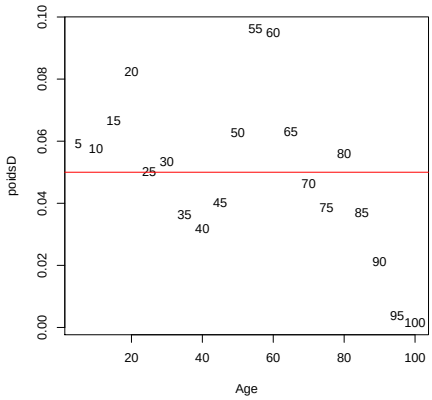
```
acp$lw
```

```
 [1] 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
[17] 0.05 0.05 0.05 0.05
```

Les données sur lesquelles nous travaillons sont particulières car ce sont des moyennes de dépenses pour des classes d'âge. Nous possédons le nombre d'individus sur lequel les moyennes ont porté. Nous allons donc associé à chaque classe d'âge une pondération liée aux nombres d'individus appartenant à cette classe.

```
poidsU <- acp$lw
#
poidsD <- depsante$effectif
poidsD <- poidsD/sum(poidsD)
round(poidsD,3)
```

```
 [1] 0.059 0.058 0.067 0.083 0.050 0.054 0.036 0.032 0.040 0.063 0.096 0.095 0.063
[14] 0.046 0.039 0.056 0.037 0.021 0.004 0.001
```



Notons par exemple que les poids associés aux classes d'âge les plus élevés (90,95 et 100) sont très faibles, que les poids associés aux 55-60 ans sont les plus élevés. Quelle incidence cela va-t-il avoir sur les résultats de l'ACP ?

Ces nouvelles pondérations ont une incidence sur la moyenne et la variance.

```
moypond <- function(x,pond) sum(pond*x)
varpond <- function(x,pond) sum(pond*(x-sum(pond*x))^2)
(resmU <- sapply(mesures,moypond,poidsU))
```

```
maladie dentaire  optique
116.950   49.395   29.885
```

```
(resmD <- sapply(mesures,moypond,poidsD))
```

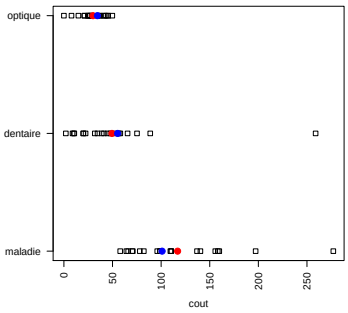
```
  maladie  dentaire   optique
101.03121  55.27092  34.46188
```

```
(resvU <- sapply(mesures,varpond,poidsU))
```

```
  maladie  dentaire   optique
2759.8585 2830.5495  164.9503
```

```
(resvD <- sapply(mesures,varpond,poidsD))
```

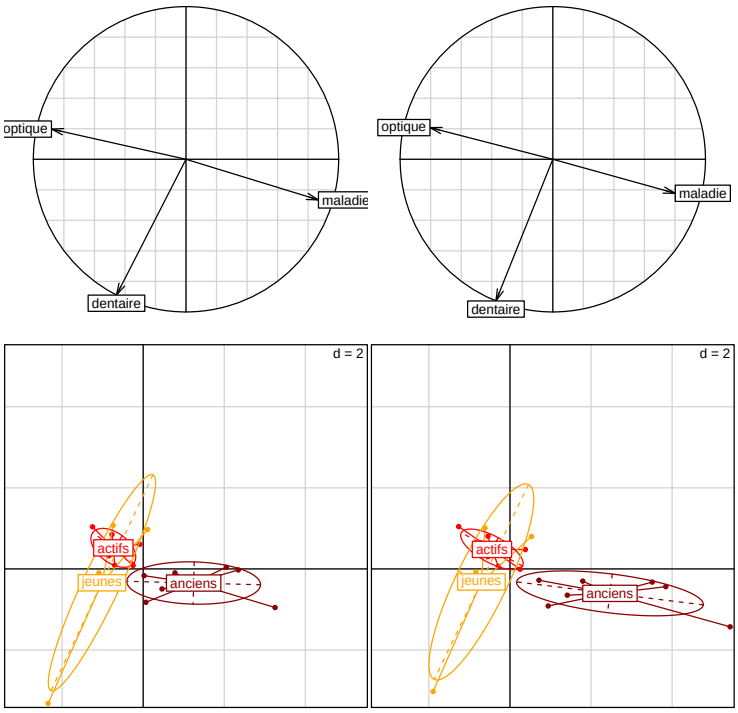
```
   maladie   dentaire    optique
1198.40706 3446.77153   88.41082
```



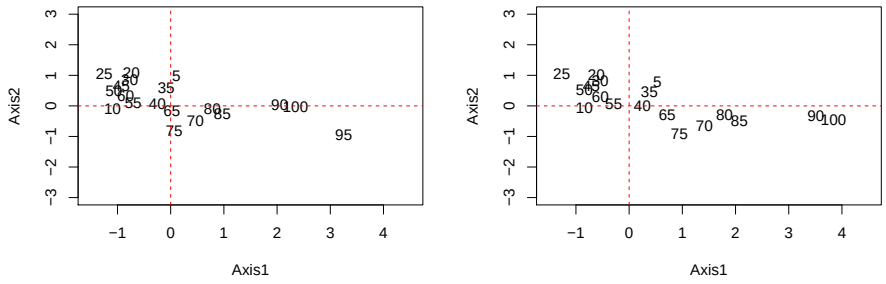

```
acpD <- dudi.pca(mesures, row.w=poidsD, scannf=FALSE, nf=3)
```

Nous visualisons à gauche les résultats de la première ACP avec la pondération uniforme et à droite la seconde ACP avec la pondération liée aux nombres d'individus par classe d'âge.

```
liminf <- min(cbind(acp$li,acpD$li))-0.10
limsup <- max(cbind(acp$li,acpD$li))+0.10
lims <- c(liminf,limsup)
par(mfrow=c(2,2))
s.corcircle(acp$co)
s.corcircle(acpD$co)
s.class(dfxy = acp$li, fac = groupe, col = gcol, xlim=lims, ylim=lims)
s.class(dfxy = acpD$li, fac = groupe, col = gcol, xlim=lims, ylim=lims)
```



Nous réalisons un zoom sur les deux cartes factorielles (nous ne représentons pas la classe des 15 ans caractérisée par une dépense dentaire importante.)



En choisisant quelques classes d'âge, quelle conclusion pouvons nous tirer de cette dernière ACP sur pondération non uniforme ?

Exercice

1. On connait la répartition de la population française (en %) au premier janvier 2015.

Classe d'âge	Pourcentage
moins de 15 ans	18.6
15 ans	6.1
20 ans	5.7
25 ans	6.0
30 ans	6.2
35 ans	6.1
40 ans	6.9
45 ans	6.8
50 ans	6.7
55 ans	6.4
60 ans	6.1
65 ans	5.6
70 ans	3.7
75 ans et plus	9.1

2. Transformer le tableau initial en fonction des classes d'âge populationnelles.
3. Réaliser les trois analyse en composantes principales :
   (a) avec la pondération uniforme,
   (b) avec la pondération liée à l'échantillonnage,
   (c) avec la pondération liée à la population française au premier janvier 2015.
4. Comparer les résultats.

Références

[1] Daniel Adler and Duncan Murdoch. *rgl : 3D visualization device system (OpenGL)*, 2014. R package version 0.93.996.

[2] D. Chessel, A.-.B. Dufour, and J. Thioulouse. The ade4 package-I- One-table methods. *R News*, 4 :5–10, 2004.