
Conception de graphiques avec



Christophe Lalanne <christophe.lalanne@gmx.net>

Table of Contents

Introduction	1
La gestion des graphiques sous R	3
Variables numériques	3
Histogramme et estimateur de densité	3
Graphiques conditionnels	10
Diagrammes de dispersion	12
Boîtes à moustaches	20
Variables qualitatives	25
Distributions univariées	25
Représentations conjointes	32
Représentations conditionnelles	34
Données de survie	35
Séries temporelles	35
Données spatiales	37
Divers	37
Références	39
Index	39

Introduction

Nous explorons dans ce document les possibilités graphiques de **R**, à l'aide des fonctions graphiques standards et celles contenues dans les packages `lattice`, `grid`, et `ggplot`.

Le lecteur intéressé trouvera d'autres illustrations, sans doute bien meilleures que celles présentées ici, sur le web, notamment :

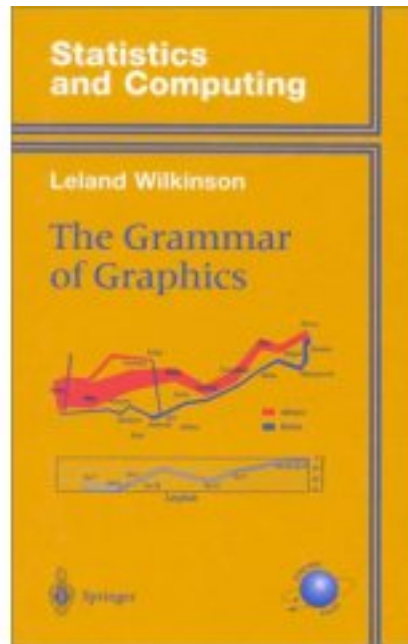
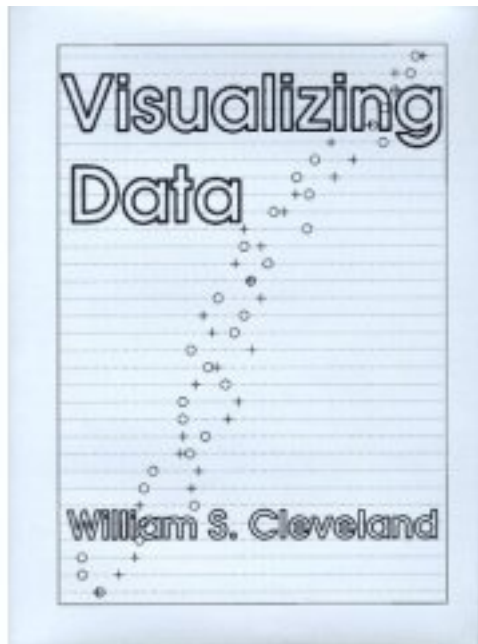
- R Graphics (P. Murrell) : <http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html>
- R Graph Gallery : <http://addictedtor.free.fr/graphiques/>
- Statistiques avec R (V. Zoonekynd) : http://zoonek2.free.fr/UNIX/48_R_2004/all.html
- Treillis displays : <http://netlib.bell-labs.com/cm/ms/departments/sia/project/trellis/>
- Categorical data analysis with graphics : <http://www.math.yorku.ca/SCS/Courses/grcat/>

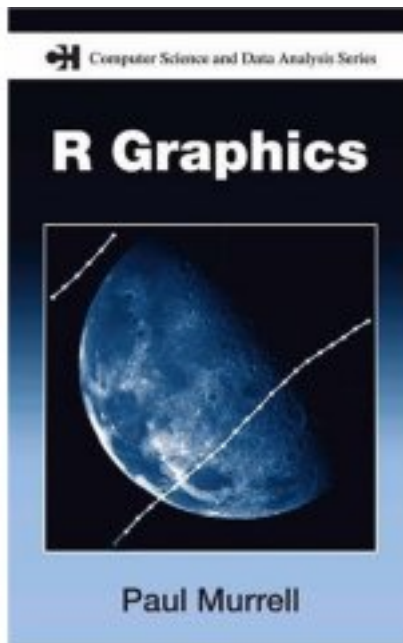
ainsi qu'en consultant les documents disponibles sur le site officiel de R, CRAN[<http://cran.r-project.org/>],

en particulier :

- R pour les débutants (E. Paradis) : http://cran.armsmachinandi.it/doc/contrib/Paradis-rdebuts_fr.pdf
- S-PLUS (and R) Manual to Accompany Agresti's Categorical Data Analysis (L. Thompson) : <https://home.comcast.net/~lthompson221/Splusdiscrete2.pdf>
- Using R for Data Analysis and Graphics - Introduction, Examples and Commentary (J. Maindonald) : <http://cran.armsmachinandi.it/doc/contrib/usingR-2.pdf>
- Statistical Computing and Graphics Course Notes (F. Harrell) : <http://cran.armsmachinandi.it/doc/contrib/Harrell-statcomp-notes.pdf>

Du côté ouvrage de référence, on pourra citer : [Cleveland1993], [Wilkinson1999], [Friendly2000], [Murrell2005]. Un résumé de ces deux derniers ouvrages est disponible sur le site de Vincent Zoonekynd (<http://zoonek.free.fr>).





Les figures proposées sont au format `png` et ont été produites à l'aide de la commande :

```
png("file.png",width=500,height=500,pointsize=12,bg="white")
```

La liste des fonctions graphiques utilisées peut être consultée directement à partir de l'index situé en fin de document.

Note

Ce document est plutôt “mal organisé”, et le lecteur intéressé pourra consulter un autre document, peut-être un peu mieux structuré et rédigé avec `LaTeX` + `Sweave`. Le document est disponible à cette adresse : Analyse exploratoire des données avec `R` [<http://www.aliquote.org/articles/tech/aed/aed.pdf>]. Celui-ci est beaucoup plus orienté sur l'utilisation des graphiques en fonction du type d'analyse effectué (ANOVA, Régression, etc.) et de l'objectif recherché (exploration, diagnostic, synthèse).

Avertissement

Ce document est en cours de construction (au cas où cela ne se verrait pas dans les premiers paragraphes, le lecteur s'en rendra vite compte par la suite...)

La gestion des graphiques sous R

Il existe deux principaux types de systèmes graphiques sous `R` : `grid` et `lattice`.

Variables numériques

Histogramme et estimateur de densité

L'histogramme est un outil très utile pour visualiser la répartition des valeurs d'une variable numérique. Cependant, le choix de l'intervalle de découpage des valeurs n'est pas toujours aisé. En effet, avec un intervalle trop faible, on fait apparaître trop de variations souvent insignifiantes, tandis qu'avec un intervalle trop élevé les variations de la répartition s'effacent au profit d'une distribution peu "discriminante" et ne laissent pas apparaître une éventuelle distribution bimodale. Le choix de la taille de l'intervalle s'effectue avec l'option `breaks` de la fonction `hist`. Si cette option n'est pas renseignée, **R** calcule un intervalle par défaut (option `Sturges`) selon laquelle $h = \text{range}(x) / \log_2(n) + 1$. Il existe également d'autres estimations de la taille optimale de l'intervalle de classes à partir d'algorithmes d'optimisation entre biais et variance pour des distributions de référence considérées comme normales [Venables2002] (p. 112). Comme indiqué dans l'aide en ligne (`?hist`), il faut spécifier la méthode `Scott` ou `FD` (pour "Friedman-Diaconis"). Pour ces méthodes, l'intervalle choisi est calculé comme

```
h = 3.5*s*n^(-1/3)
```

(Scott, 1979, voir [Venables2002])

ou

```
h = 2*R*n^(-1/3)
```

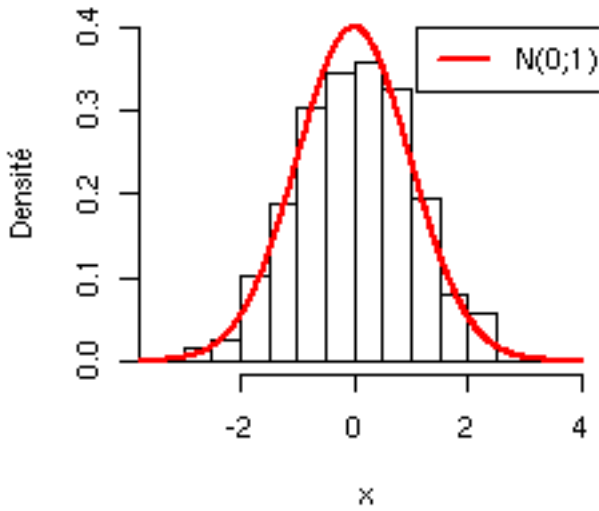
(Freedman & Diaconis, 1981, *ibidem.*).

On peut tester la première méthode à l'aide en simulant un jeu d'observations gaussiennes :

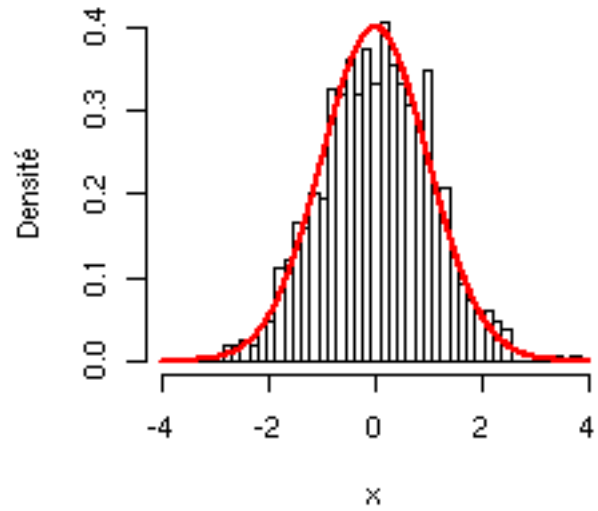
```
# on vérifie l'évolution de h(n)
x <- rnorm(10000)
n <- length(x)
h <- vector("numeric",n-1)
for (i in 2:n) {
  h[i-1] <- (max(x)-min(x))/(log(i,2)+1)
}
plot(h,type="l",xlab="n")
# comparaison entre les choix pour `breaks` et l'option "Sturges"
xx <- seq(-4,4,by=.01)
idx <- c(50,20,8)
par(mfrow=c(2,2))
hist(x,main="Méthode Sturges",ylab="Densité",ylim=c(0,0.4),proba=T)
lines(xx,dnorm(xx),col="red",lwd=2)
legend(1.1,0.4,legend="N(0;1)",lty=1,lwd=2,col="red")
for (i in idx) {
  hist(x,breaks=seq(-4,4,len=i),main=paste("h=",8/i,"(n=50)"),ylim=c(0,0.4),
  ylab="Densité",proba=T)
  lines(xx,dnorm(xx),col="red",lwd=2)
}
```

Figure 1. Choix de l'intervalle de classe pour un histogramme

Méthode Sturges



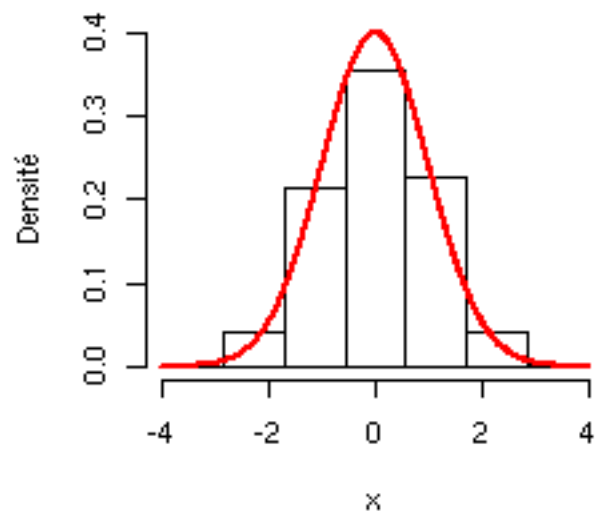
h=0.16 (n=50)



h=0.4 (n=20)



h=1 (n=8)



Des estimateurs locaux de densité sont superposés sur chacun des graphiques précédents. Ces estimations de densité (non-paramétrique) sont accessibles à l'aide de la fonction `density`, qui accepte pour principaux paramètres :

- la taille de la fenêtre de lissage ;
- le type de noyau à choisir parmi : "gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine", et "optcosine".

On peut voir ce que cela donne en variant le paramètre `adj` :

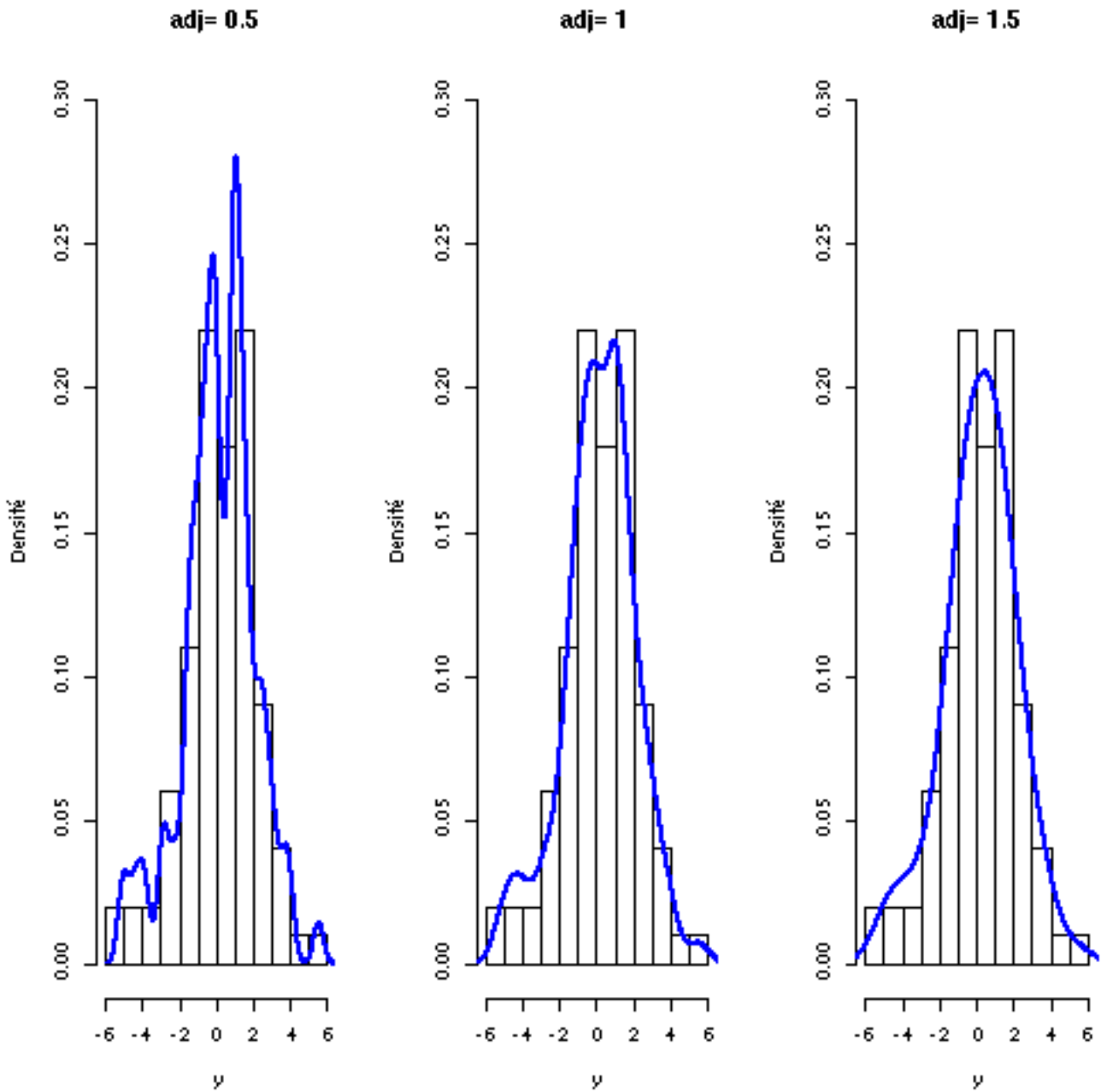
```
y <- rnorm(100,0,(1+2*rbinom(100,1,0.35))) # [Venables2002]
par(mfrow=c(1,3))
```

```

for (i in seq(0.5,1.5,by=0.5)) {
  hist(y,main=paste("adj=",i),ylim=c(0,0.3),ylab="Densité",proba=T)
  lines(density(y,adj=i),col="blue",lwd=2)
}

```

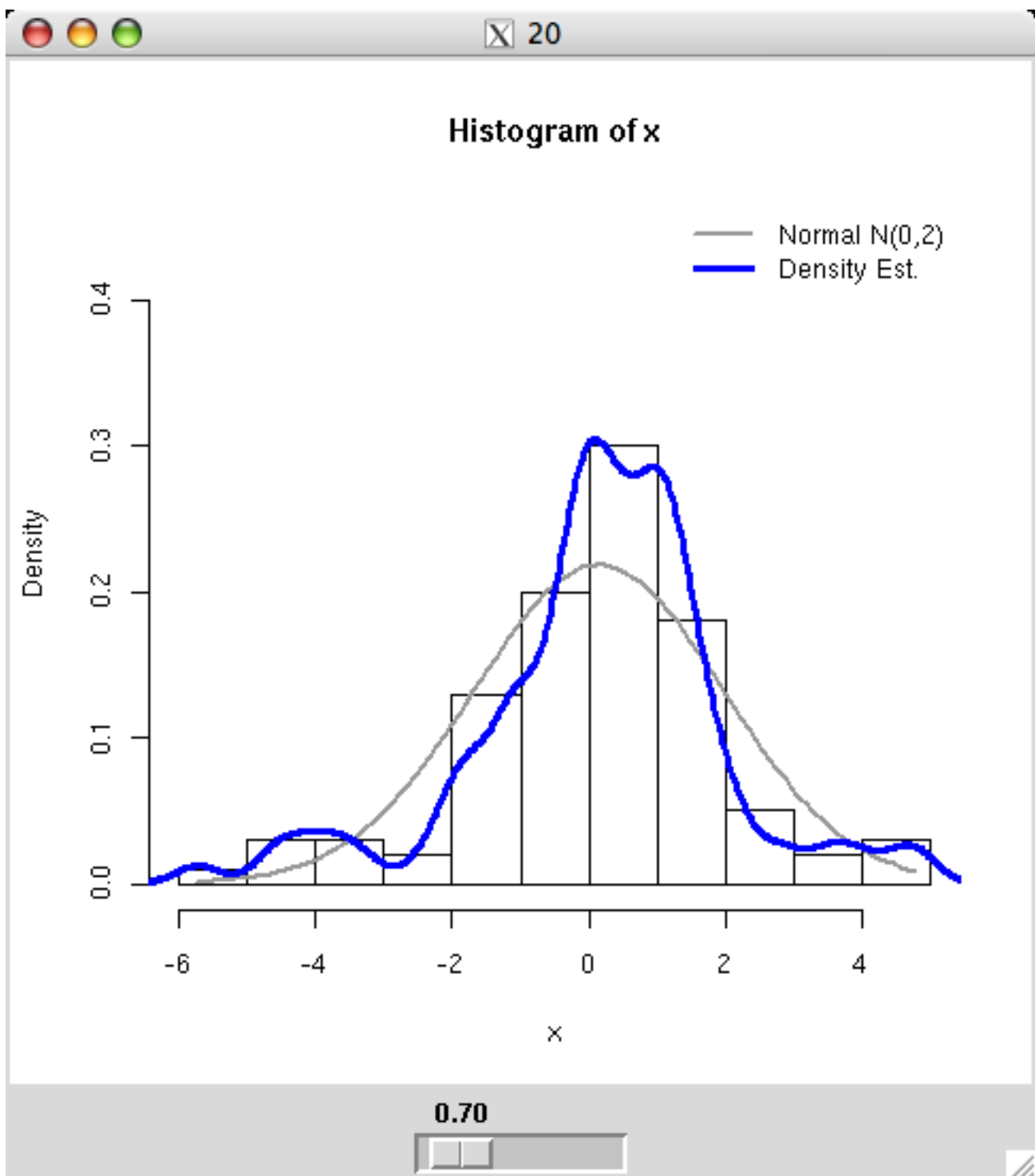
Figure 2. Influence de la taille de la fenêtre de lissage



Lorsque le paramètre de lissage est faible (e.g. $\text{adj}=0.5$), la courbe de densité souligne beaucoup de variations d'effectifs qui ne semblent pas particulièrement remarquables, tandis qu'à l'inverse, avec un fort lissage ($\text{adj}=1.5$), seule subsiste la tendance générale unimodale. On notera que l'asymétrie gauche reste visible, comme avec $\text{adj}=1$ (option par défaut sous **R**).

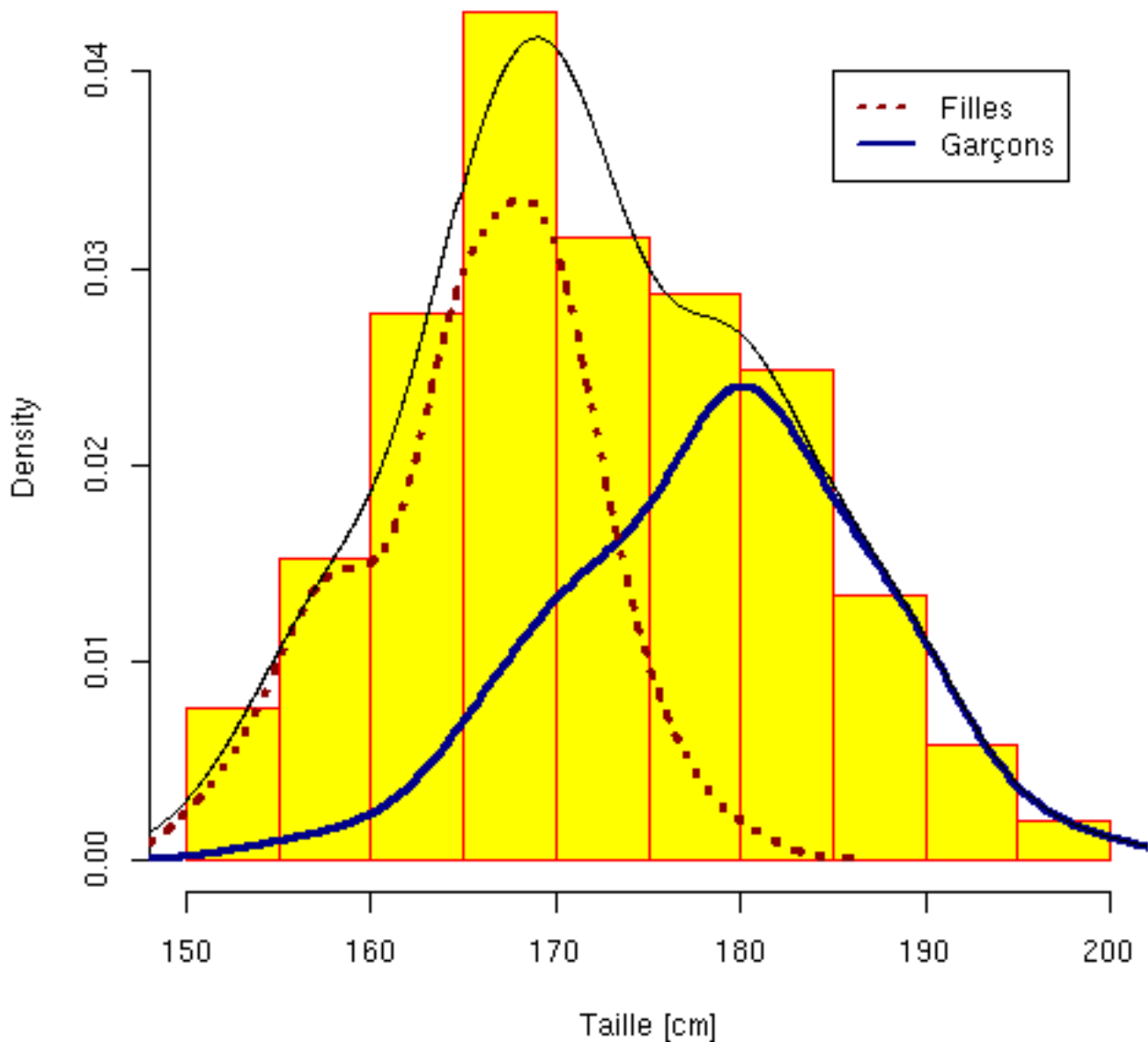
On peut visualiser une démonstration de ce procédé dynamiquement à l'aide du script suivant :

density_tk.R. Ce script nécessite la librairie `tkrplot`.



Voici un autre exemple d'utilisation de l'histogramme sous **R** (`hist1.R`) :

Taille de 237 étudiants



On peut enfin s'amuser à reproduire un des exemples donnés par John Fox (<http://socserv.mcmaster.ca/jfox/>).

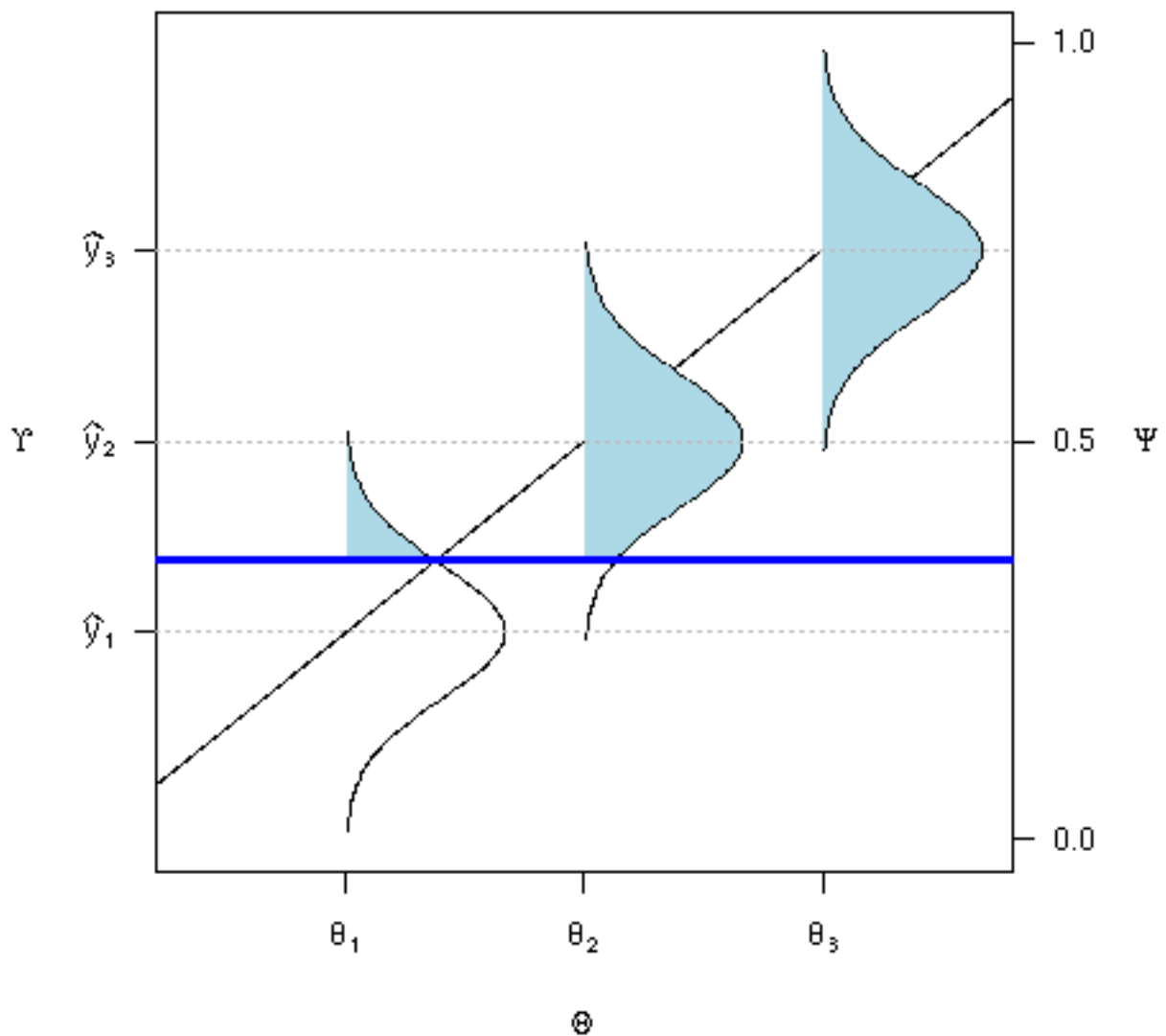
```
old.par <- par(mar=c(5,4,4,5))
xx <- c(.2, .5, .8)
yy <- xx*0.8+0.1
plot(c(0,1), c(0,1), type="n", axes=FALSE, xlab=expression(Theta), ylab="")
axis(1, at=xx, labels=c(expression(theta[1]),expression(theta[2]),expression(theta[3])))
axis(2, at=yy, las=1, labels=c(expression(widehat(y)[1]),
expression(widehat(y)[2]),expression(widehat(y)[3])))
axis(4, at=seq(0,1,by=0.5), las=1)
box()
# draw the regression line
abline(0.1,0.8)
# add normal deviates for each Y|x_i plus corresponding probability
# density for a binary response
x <- seq(-3,3,length=100)
y <- dnorm(x)/2
```

```

threshold <- 0.35
for (i in 1:3) {
  y1 <- x/12 + yy[i]
  x1 <- y + xx[i]
  lines(x1, y1)
  whichy <- y1>=threshold
  new.x1 <- c(xx[i],x1[whichy])
  new.y1 <- c(y1[whichy][1],y1[whichy])
  polygon(new.x1, new.y1, col="light blue", border=NA)
}
abline(h=yy, lty=3, col="gray")
abline(h=threshold,lwd=3,col="blue")
mtext(c(expression(Upsilon1),expression(Psi)),side=c(2,4),line=3,las=1)
par(old.par)

```

Figure 3. Une variante de la fonction `densityplot`

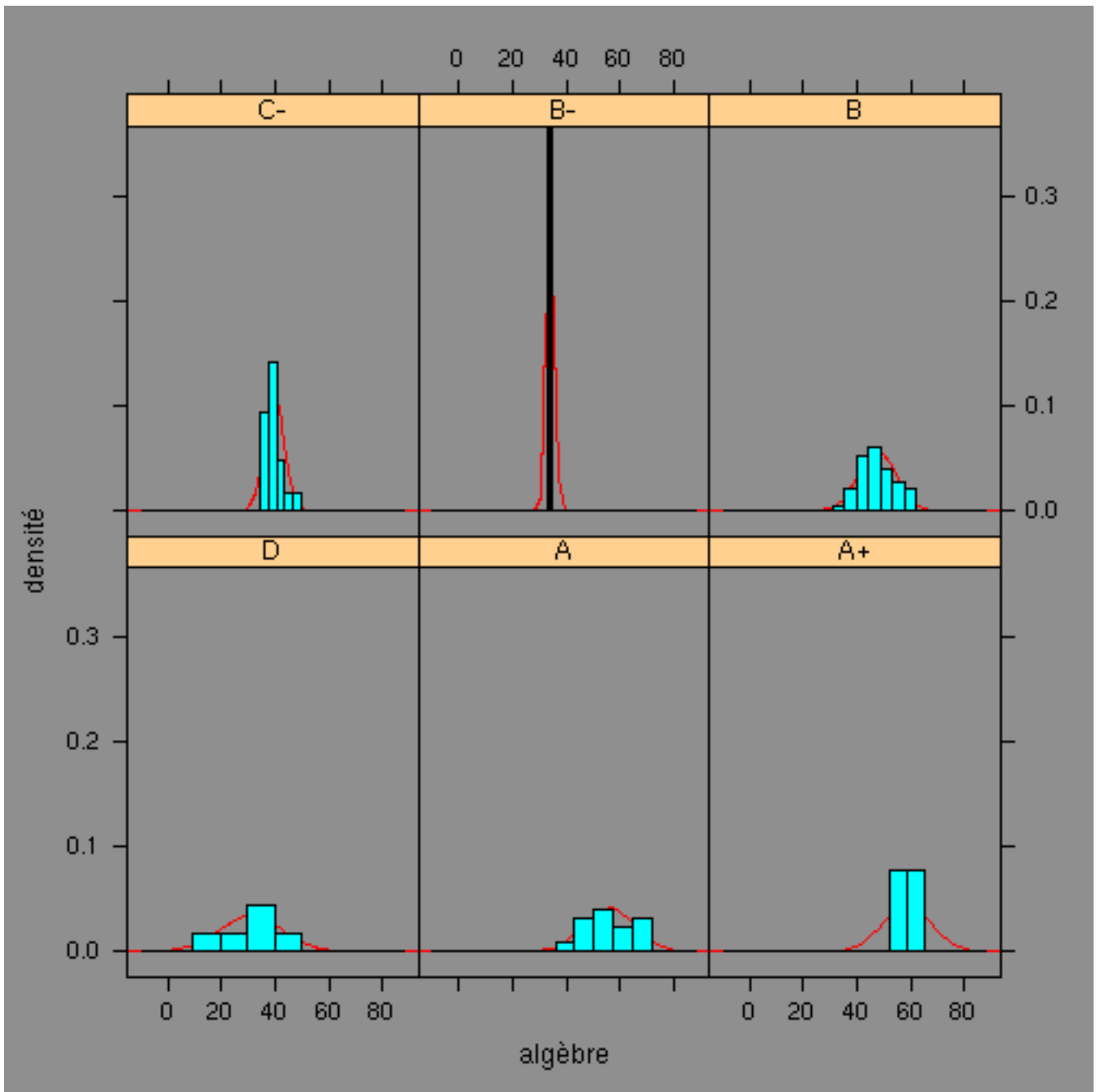


Graphiques conditionnels

La librairie `lattice` offre des possibilités graphiques additionnelles puisqu'elle propose les mêmes classes d'outils graphique (histogramme, boîte à moustaches, diagramme, etc.) mais avec la possibilité d'effectuer des représentations graphiques conditionnellement aux valeurs ou niveaux d'une variable concomitante.

```
library(ade4)
library(lattice)
data(deug)
x <- deug$tab$Algebra
y <- deug$result
densityplot(~x|y,xlab="algèbre",ylab="densité",panel=function(x,...) {
  panel.mathdensity(dmath=dnorm,args=list(mean=mean(x),sd=sd(x)),col="red")
  panel.histogram(x,breaks=NULL,col="cyan")})
```

Figure 4. La fonction `densityplot`



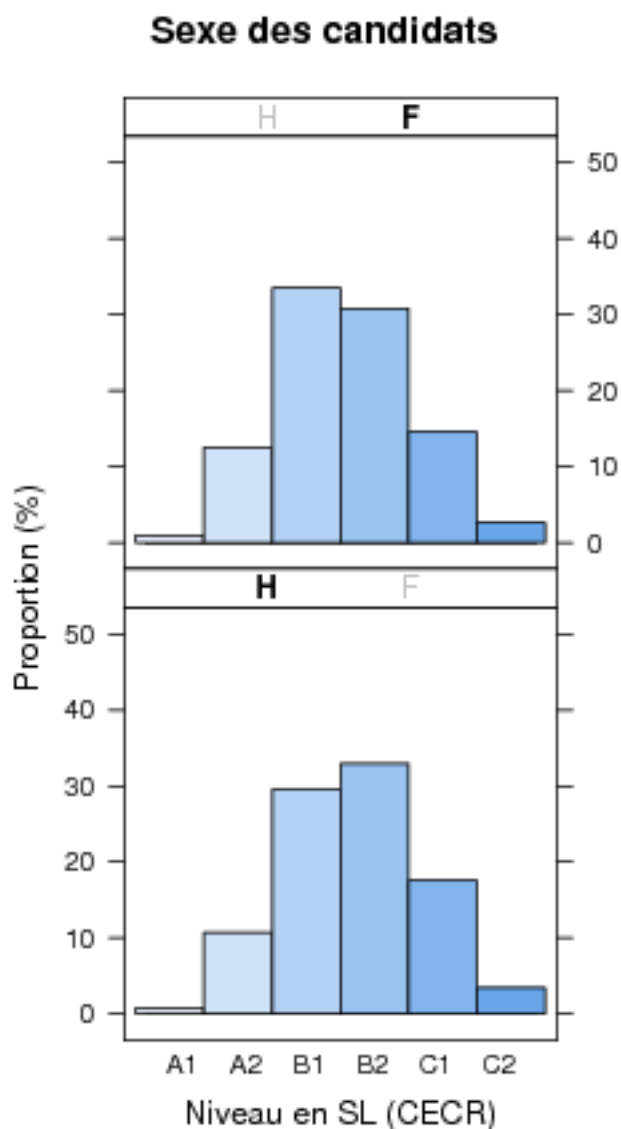
Exportation des graphiques `lattice`

On peut également exporter le graphique produit au format postscript, qui donne un meilleur rendu pour une insertion dans un document et qui offre la possibilité d'avoir un fond uniforme blanc, et non pas gris comme par défaut. On utilisera par exemple la commande

```
postscript("densityplot.eps", width = 10.0, height = 10.0,
           horizontal = FALSE, onefile = FALSE, paper = "special",
           encoding = "ISOLatin1.enc")
```

pour produire la figure `densityplot.eps` [`img/densityplot.eps`]. On veillera dans ce cas à utiliser `print(densityplot(...))` pour générer correctement le graphique postscript.

Figure 5. Exemple de distribution conditionnelle



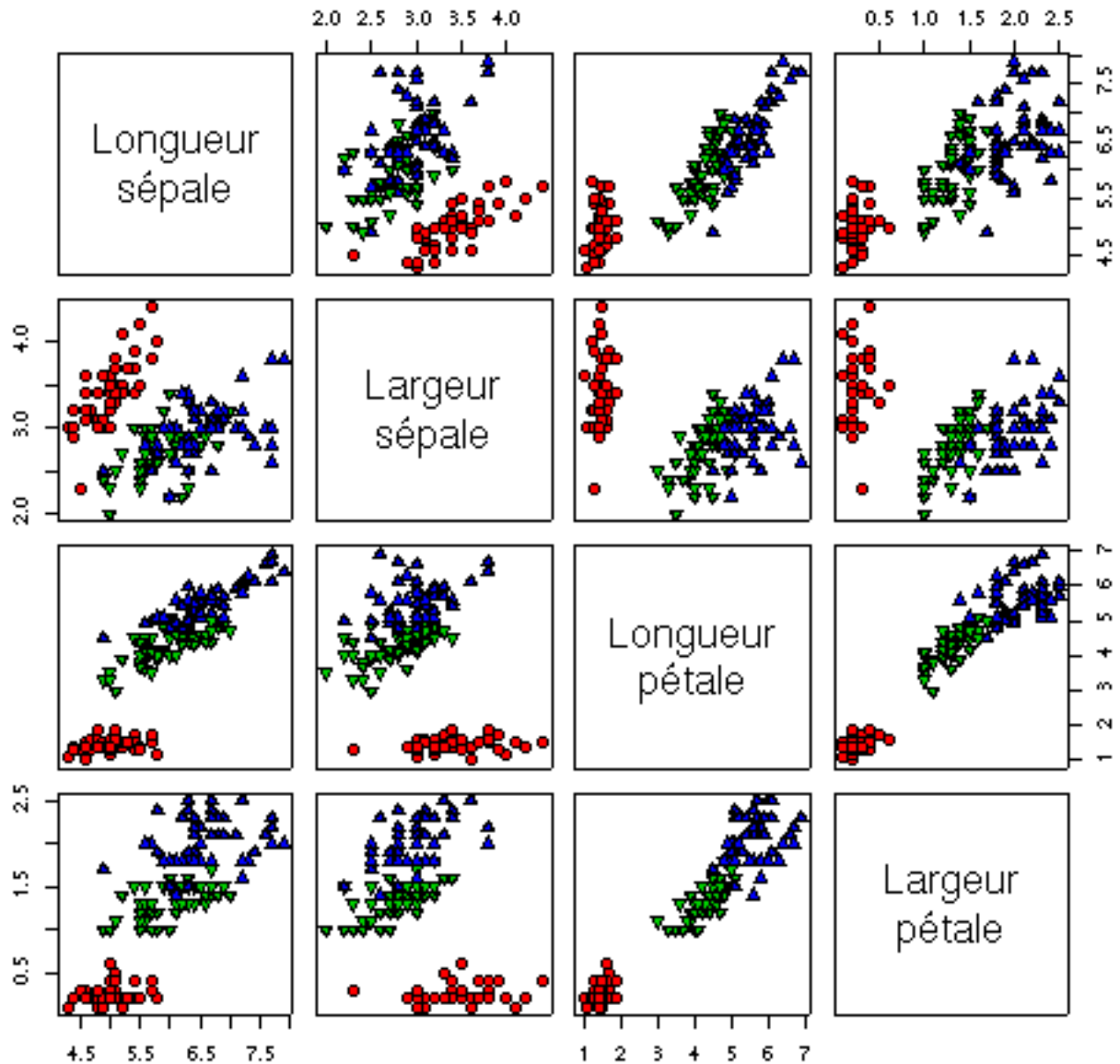
Diagrammes de dispersion

On peut utiliser ce type de représentation dans le cas où l'on croise plusieurs variables, par exemple :

```
plot(iris[,1:4],bg=c("red","green3","blue")[iris[,5]],  
     pch=c(21,25,24)[iris[,5]],main="Les iris de Fisher",  
     labels=c("Longueur\nsépale","Largeur\nsépale",  
             "Longueur\npétale","Largeur\npétale"))
```

Figure 6. Diagramme de dispersion pour les iris de Fisher

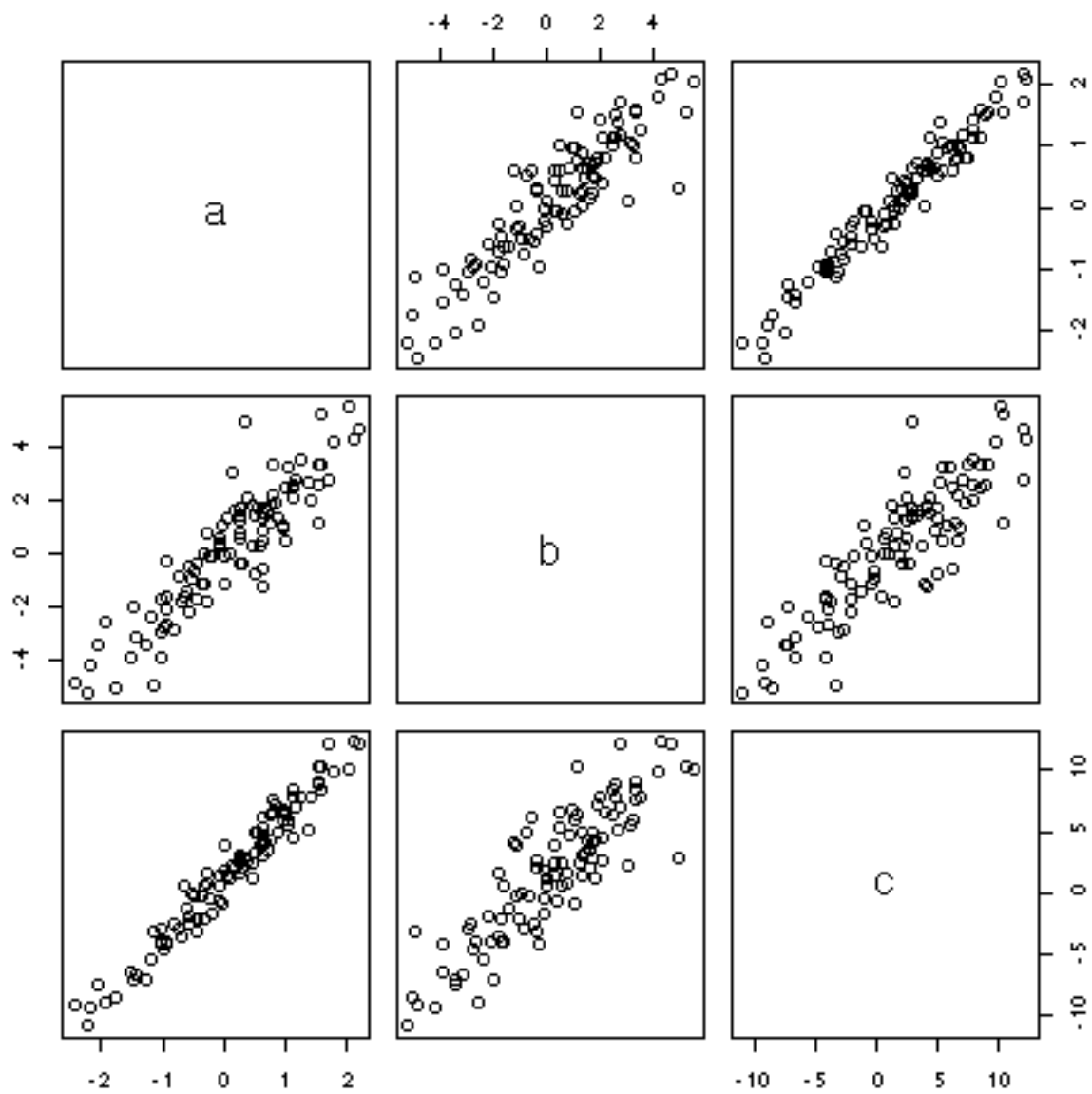
Les iris de Fisher



La fonction `pairs` permet également de représenter une matrice graphique :

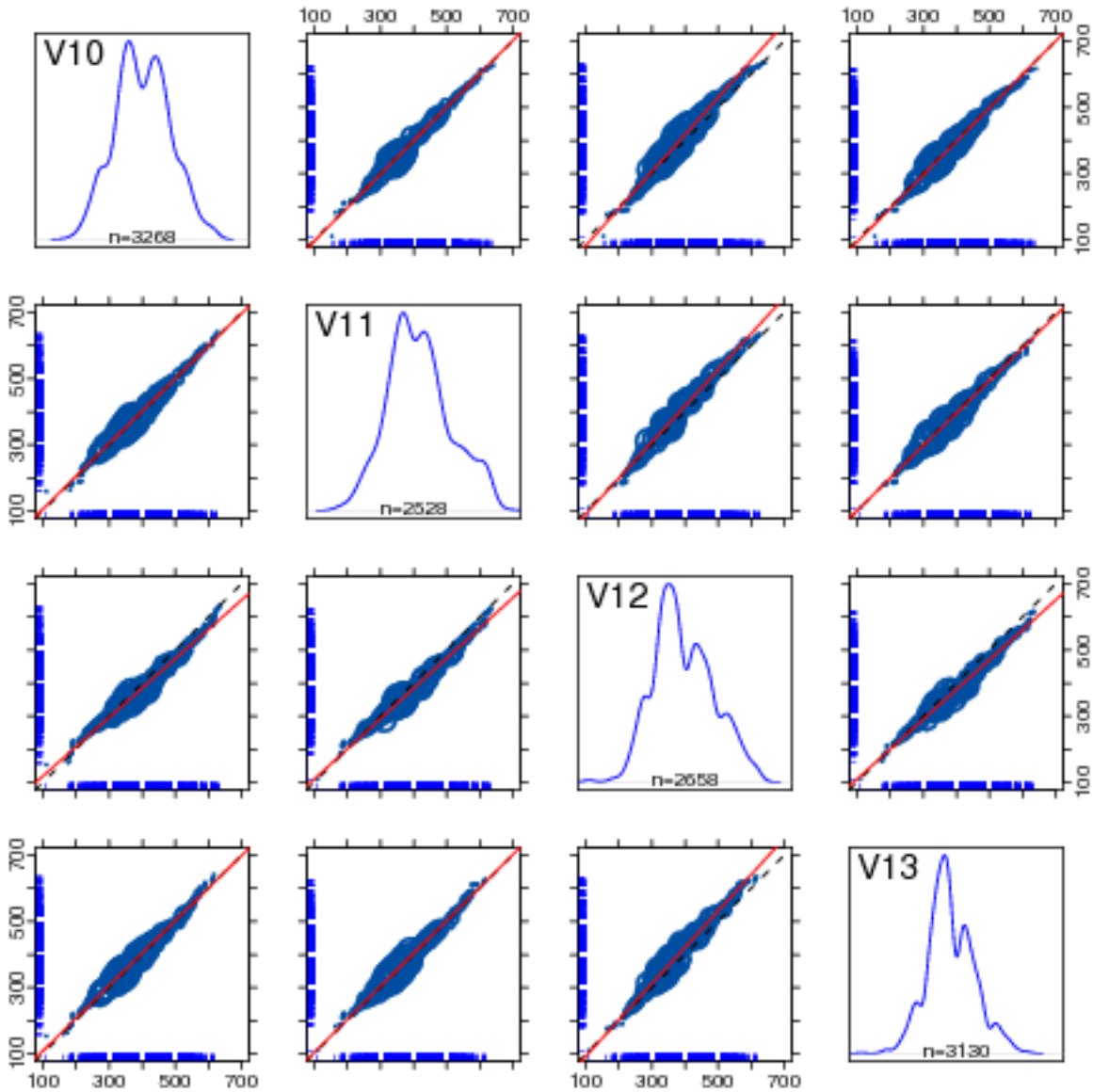
```
a<-rnorm(100)
b<-2*a+rnorm(100)
c<-5*a+rnorm(100)+runif(100)*2
pairs(cbind(a,b,c))
```

Figure 7. Les diagrammes de dispersion multivariés

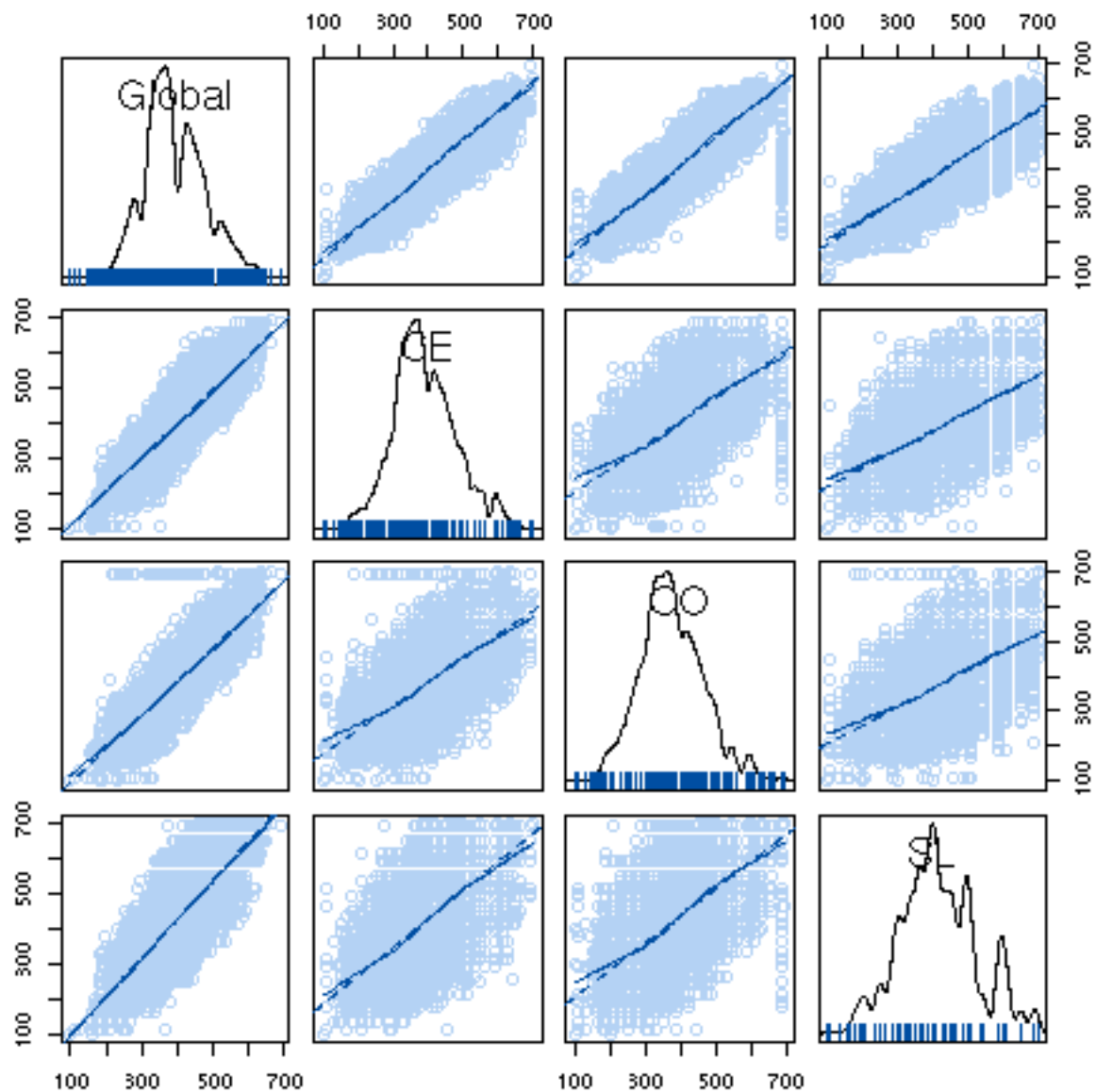


On peut personnaliser la diagonale de ce graphique grâce à la librairie

Figure 8. Personnalisation du diagramme de dispersion



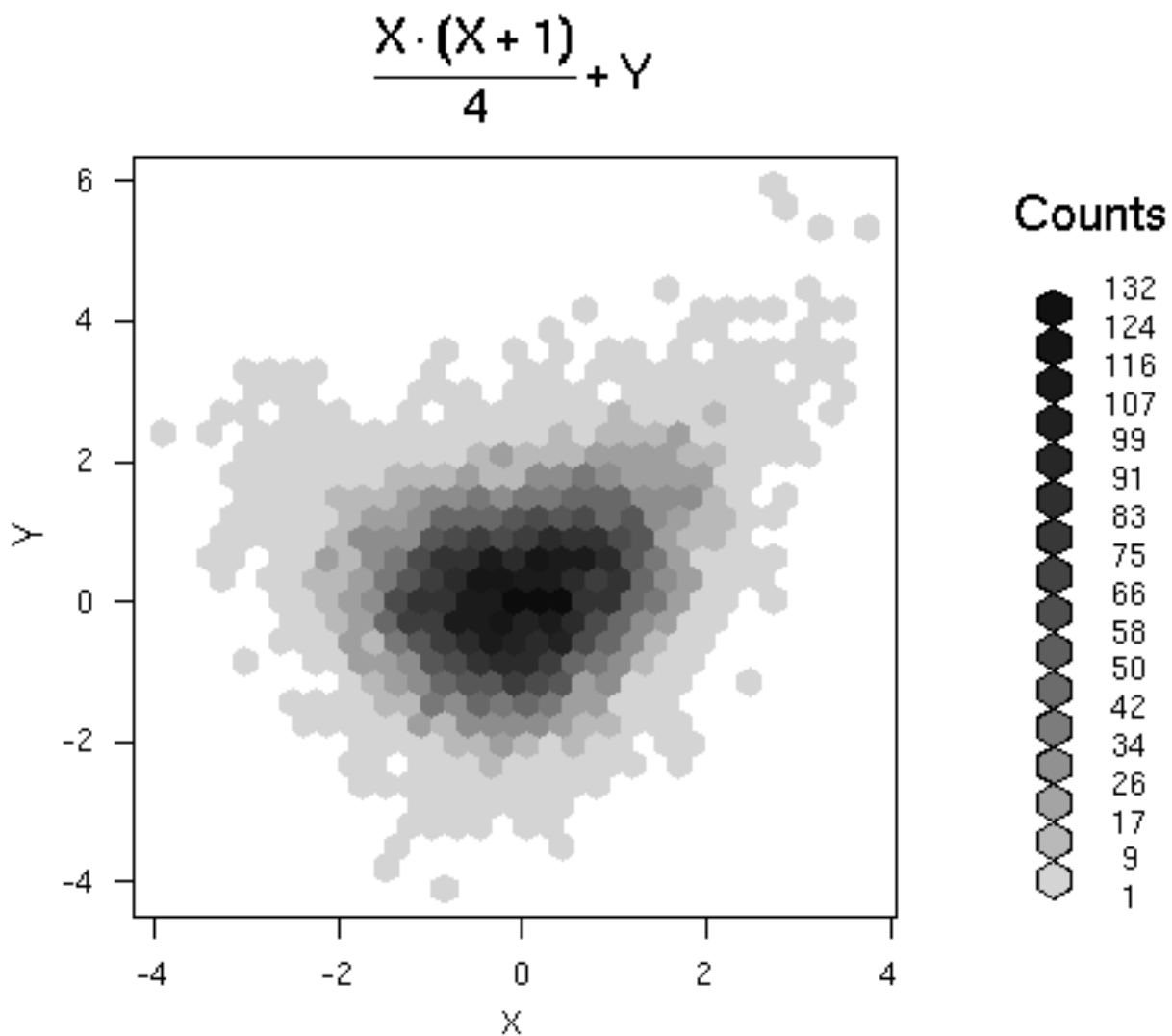
Mais dans le cas où le nombre de points est très important ($n > 10000$), cela devient difficile d'exporter directement l'image à moins d'accepter de gérer des images `postscript` ou `png` de 12 Mo, comme par exemple l'image suivante !



La librairie `hexbin` fournie avec les extensions de Bioconductor (www.bioconductor.org) permet de repr senter de larges volumes de donn es :

```
library(hexbin)
# adapted from the example in ?hexbin
x <- rnorm(10000)
y <- rnorm(10000)
plot(hexbin(x, y + x*(x+1)/4), ylab="Y",
      main=expression(frac(X %.% (X+1),4) + Y))
```

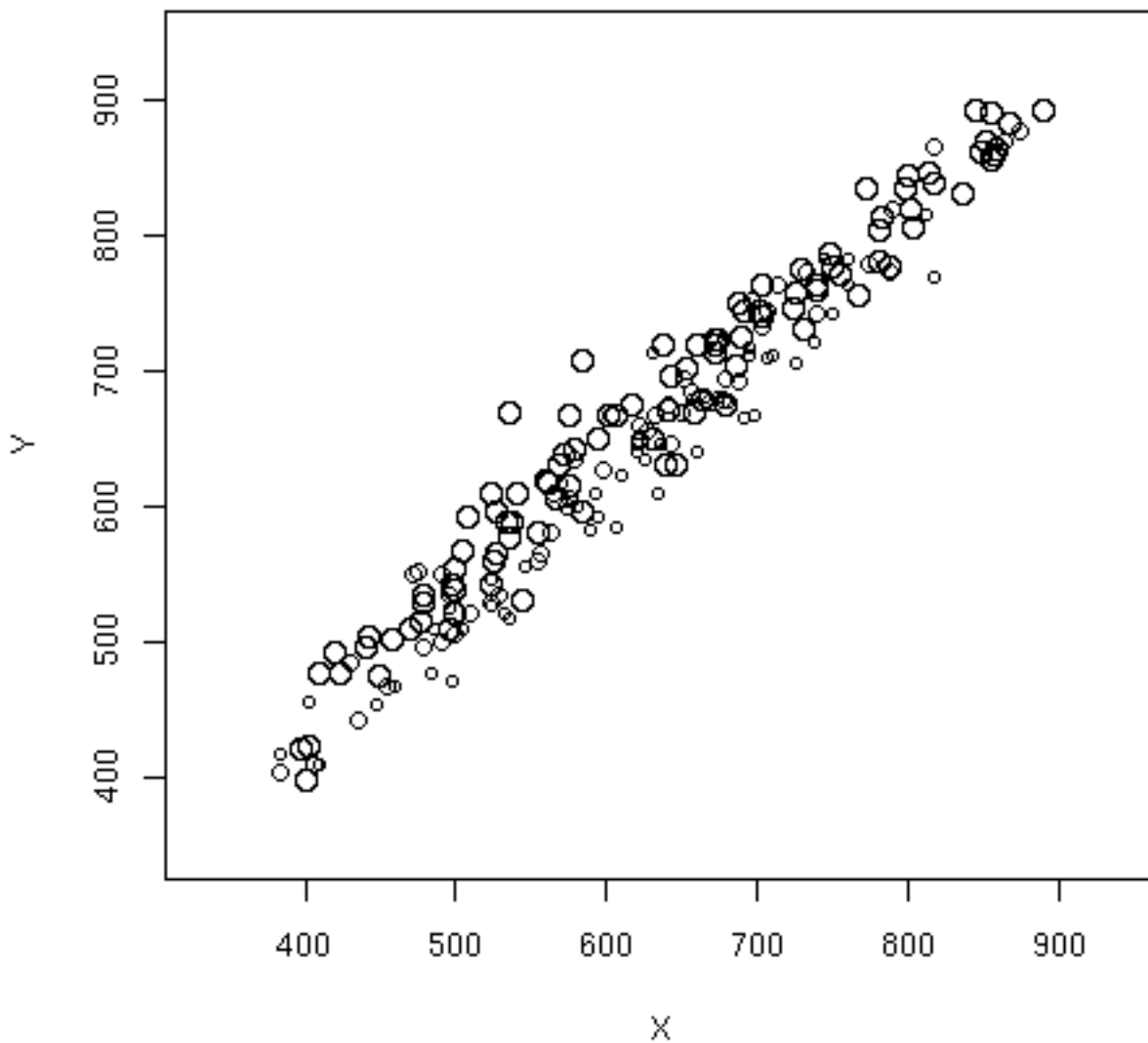
Figure 9. Diagramme de dispersion optimis  pour les gros volumes de donn es



On retrouve une représentation des effectifs à peu près comparable dans le package `survey`. Dans un premier temps, on peut représenter les points de coordonnées (x, y) avec une taille qui est proportionnelle à l'effectif associé:

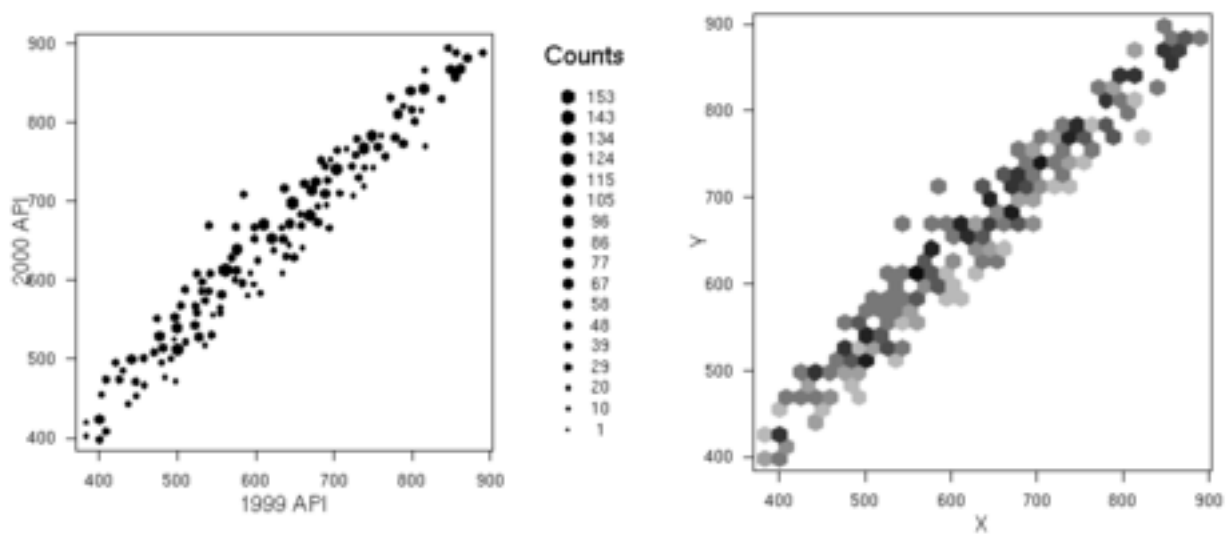
```
# from ?svyplot
data(api)
dstrat<-svydesign(id=~1, strata=~stype, weights=~pw, data=apistrat, fpc=~fpc)
svyplot(api00~api99, design=dstrat, style="bubble")
```

Figure 10. Application pour la représentation des données de sondage



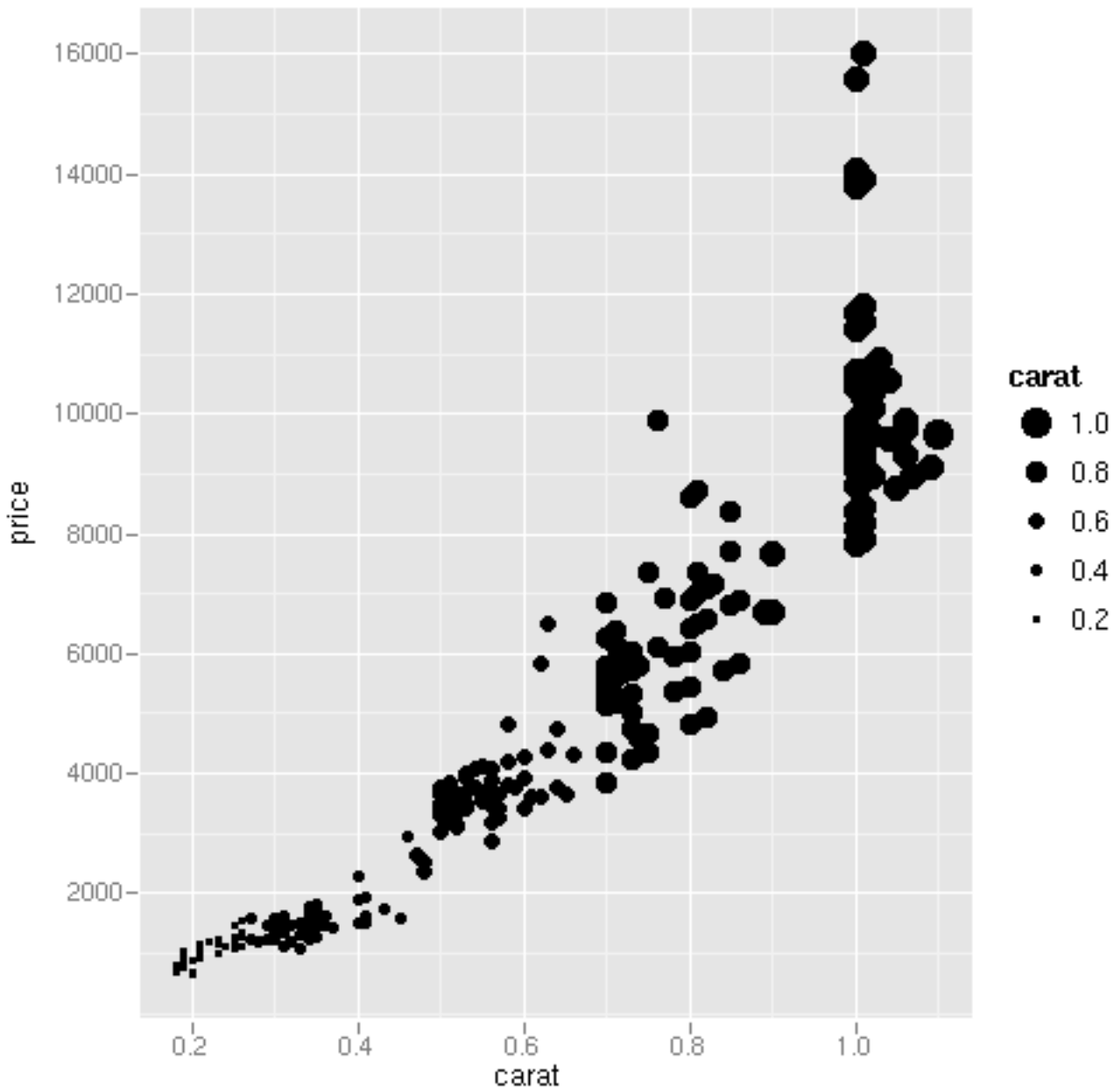
ou alors faire appel à une représentation reposant sur la librairie `hexbin` grâce à l'option `style="hex"`:

```
svyplot(api00~api99, design=dstrat, style="hex", xlab="1999 API",  
        ylab="2000 API")  
svyplot(api00~api99, design=dstrat, style="grayhex", legend=0)
```



La librairie `ggplot` (remplacée par `ggplot2` à l'heure actuelle) permet également ce type de représentation :

Figure 11. Utilisation de la librairie `ggplot`

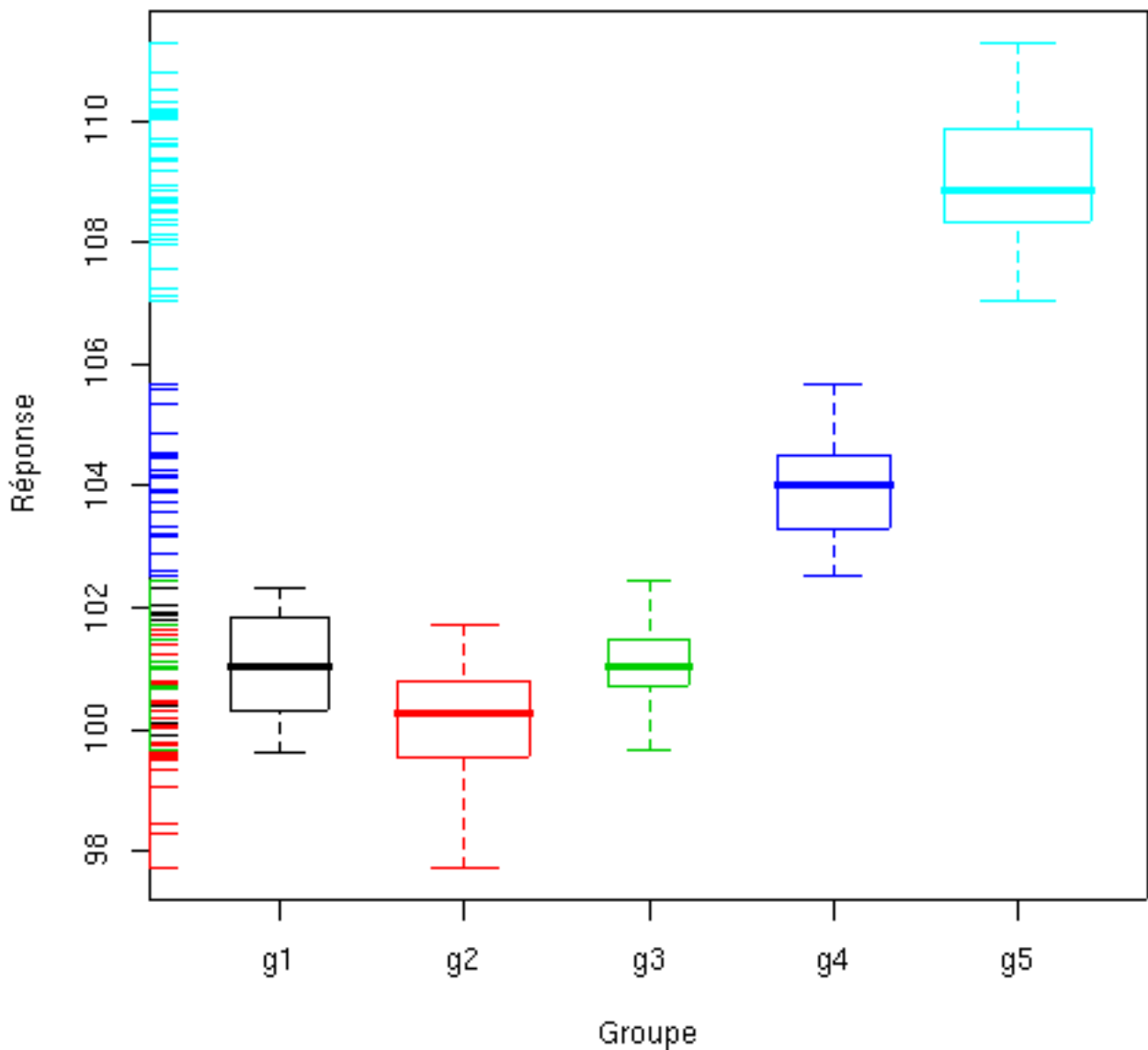


En dernier lieu, on peut imaginer une représentation de densité 2D avec

Boîtes à moustaches

```
ns <- c(15,28,10,20,35)
n <- length(ns)
group <- factor(rep(1:n,ns),labels=paste("g",1:n,sep=""))
data <- rnorm(length(group),mean=100+(as.numeric(group)-2)^2)
boxplot(data~group,border=1:n,xlab="Groupe",ylab="Réponse",varwidth=T)
for (i in 1:n) {
  rug(data[as.numeric(group)==i],side=2,col=i)
}
```

Figure 12. Boîte à moustaches et projection de la distribution univariée

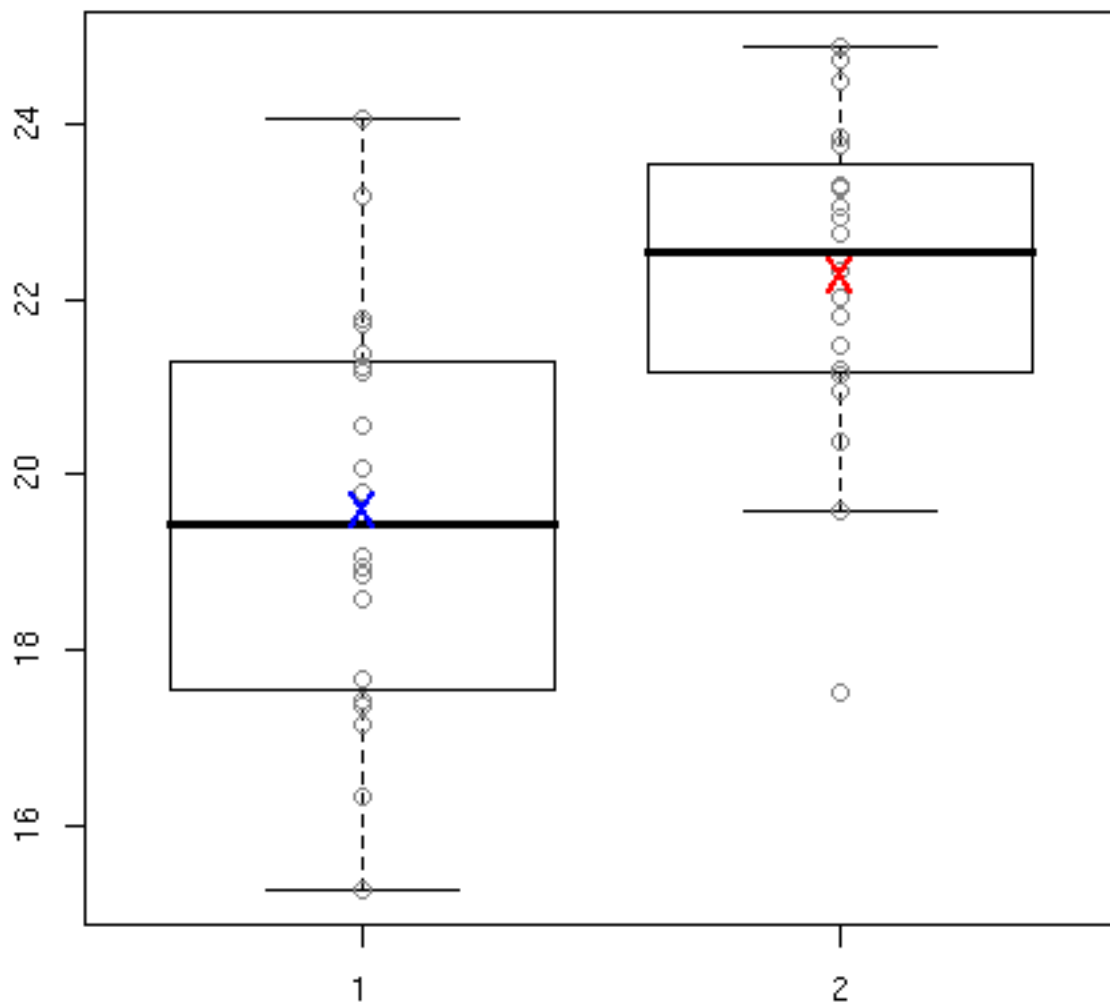


Il peut être intéressant de superposer sur cette représentation synthétique de la distribution de la variable étudiée la distribution univariée des observations :

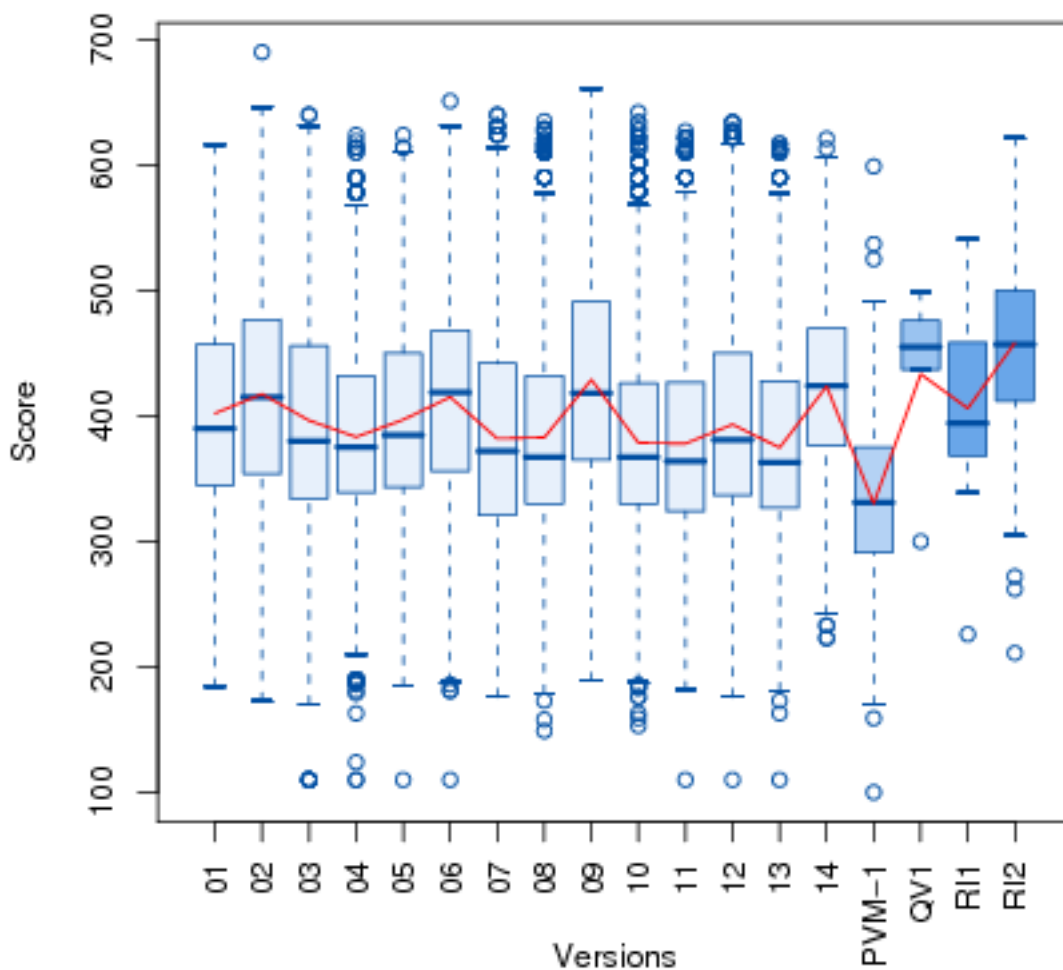
```
x <- rnorm(20,mean=20,sd=2.5)
y <- rnorm(20,mean=22,sd=2.3)

boxplot(x,y)
points(c(rep(1,20),rep(2,20)),c(x,y),col='gray50')
points(c(1,2),c(mean(x),mean(y)),pch='x',cex=2,col=c('blue','red'))
```

Figure 13. Boîte à moustaches et distribution des observations



Les paramètres graphiques sont extrêmement faciles à moduler et il est très facile d'obtenir des figures un peu plus "sexy" :



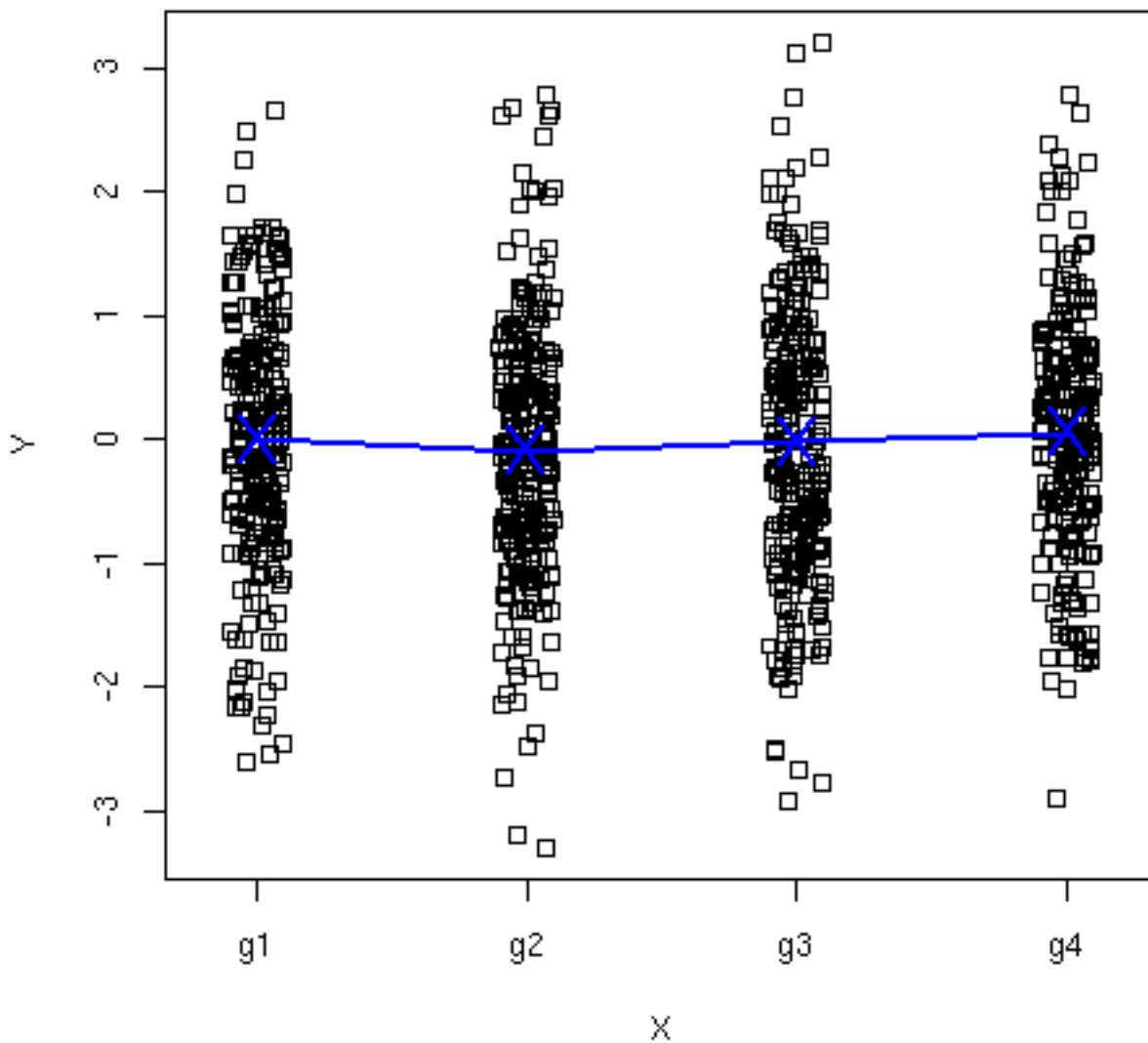
On peut également utiliser la fonction `stripchart` qui permet de ne pas superposer les points les uns sur les autres (utile quand il y a beaucoup d'observations et que certaines d'entre elles prennent la même valeur), comme ceci :

```

y <- rnorm(1000)
x <- gl(4,25,1000,labels=paste("g",1:4,sep=""))
stripchart(y~x,method="jitter",ylab="Y",xlab="X",vertical=T)
(moy <- as.numeric(tapply(y,x,mean)))
points(1:4,moy,pch="X",col='blue',cex=2)
lines(1:4,moy,col='blue',lwd=2)

```

Figure 14. Représentation de la distribution des observations avec `stripchart` (option `jitter`)



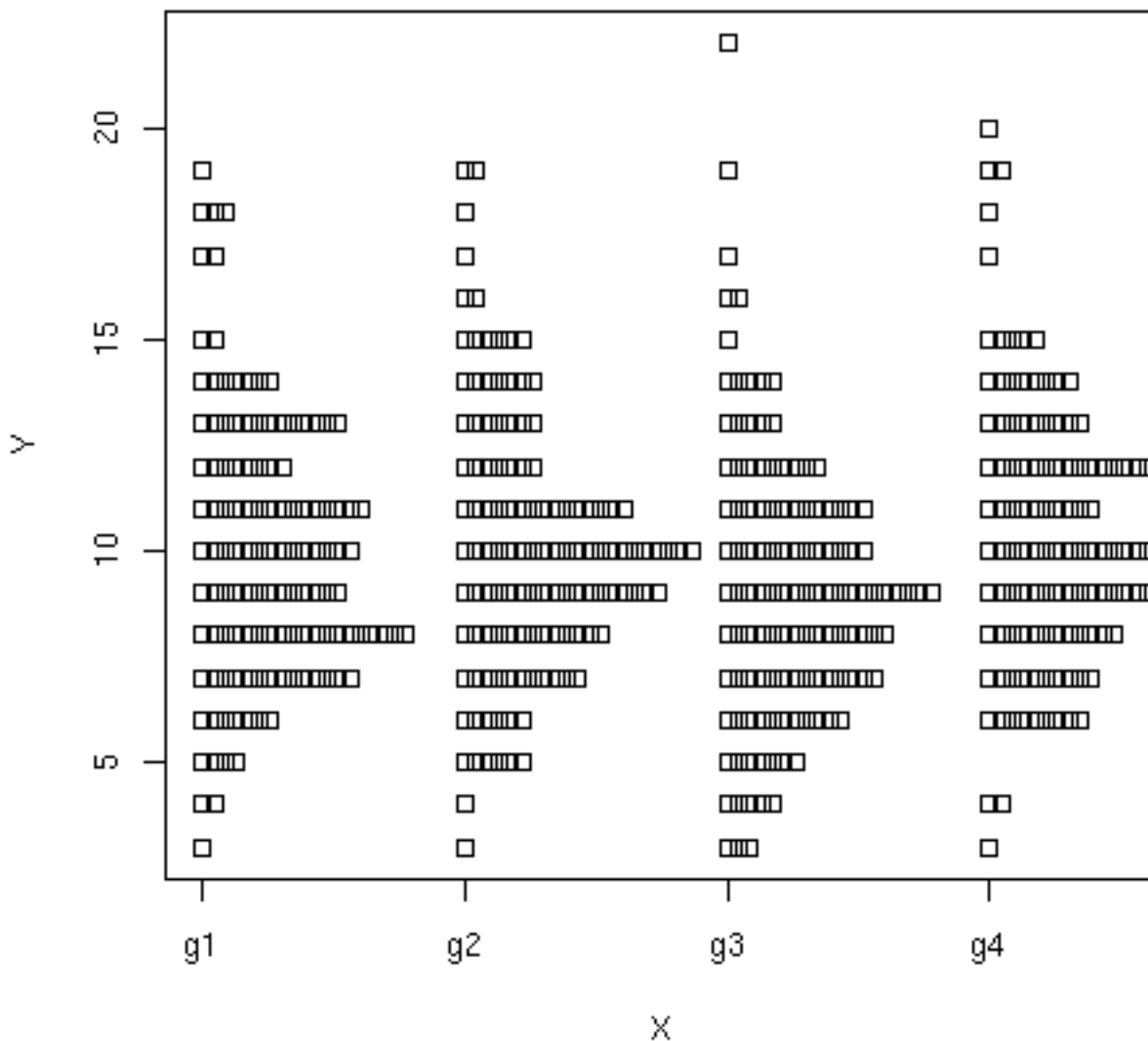
L'option `method="stack"` peut également se révéler intéressante :

```

y <- rpois(500,10)
x <- gl(4,25,500,labels=paste("g",1:4,sep=""))
stripchart(y~x,method="stack",ylab="Y",xlab="X",vertical=T)

```

Figure 15. Représentation de la distribution des observations avec `stripchart` (option `stack`)



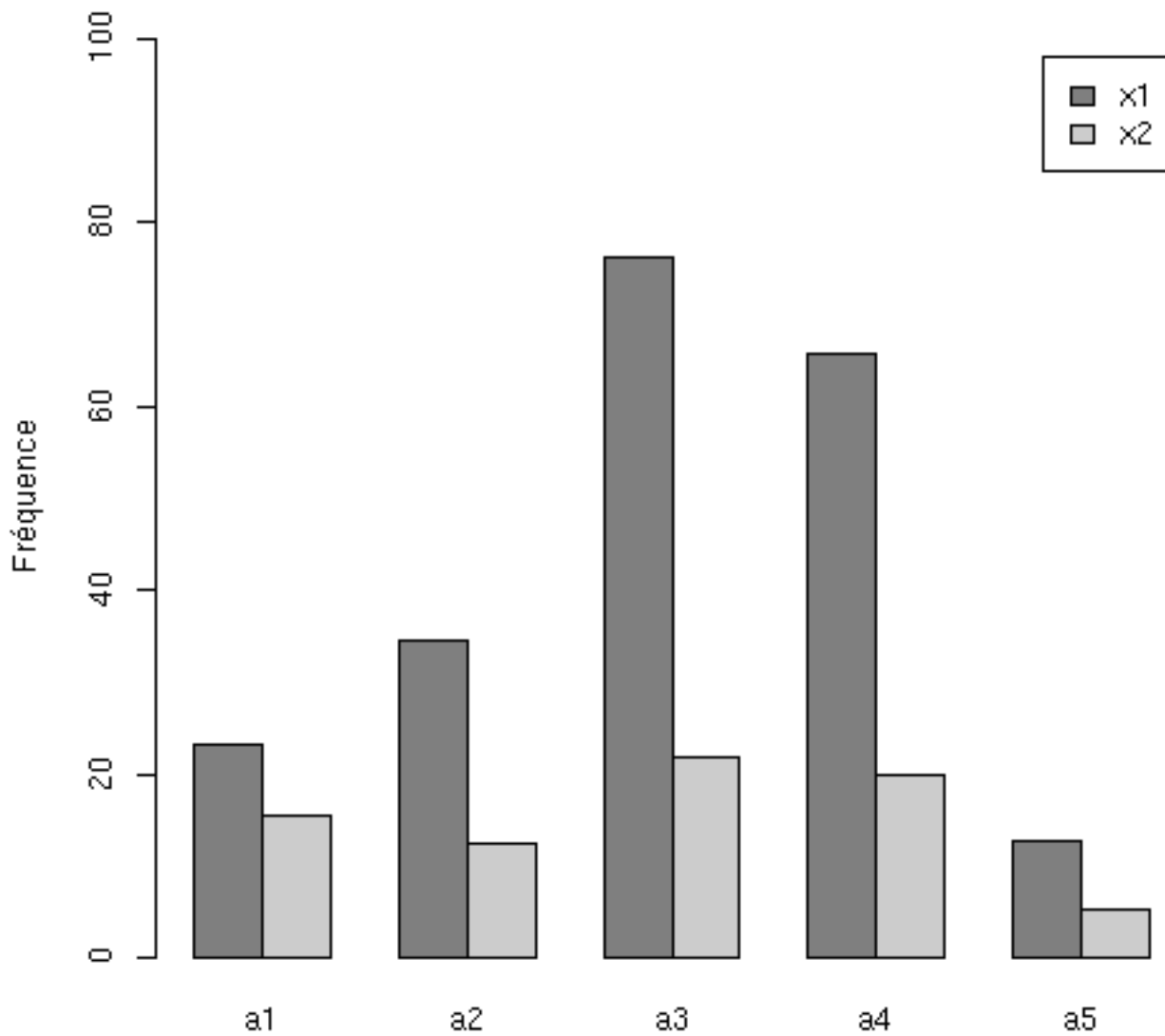
Variabes qualitatives

Distributions univariées

Les graphiques en barres sont sans doute les plus utilisés dans le cadre de la représentation de la distribution des effectifs en fonction des modalités d'une variable qualitative. Sous **R**, on peut les générer grâce à la fonction `barplot`.

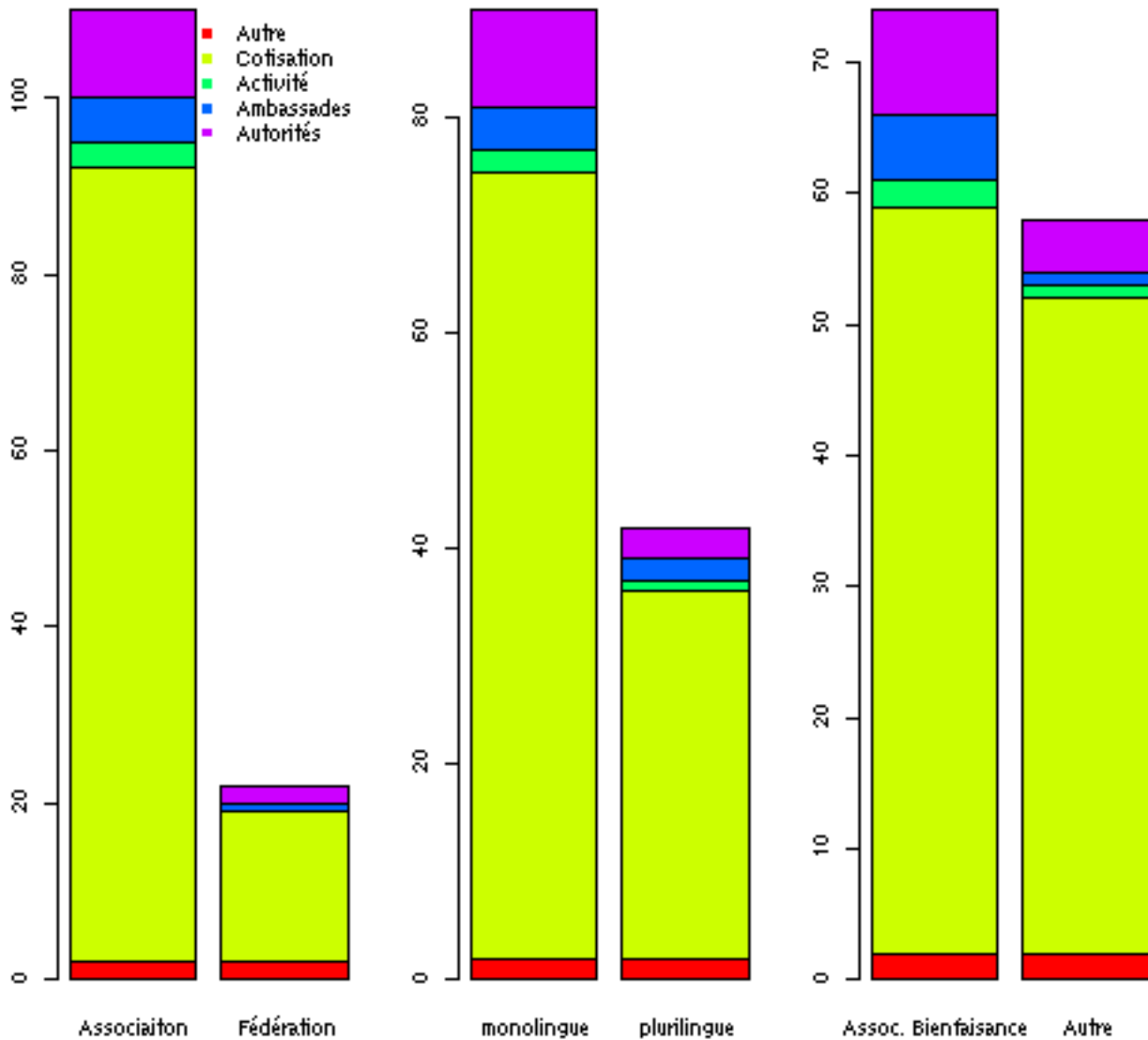
```
x1 <- c(23.2,34.5,76.3,65.8,12.6)
x2 <- c(15.6,12.4,21.8,20,5.2)
A <- gl(5,1,5,labels=c("a1","a2","a3","a4","a5"))
data <- cbind(x1,x2)
rownames(data) <- levels(A)
barplot(x1,names.arg=levels(A))
barplot(t(data),beside=T,ylim=c(0,100),legend.text=colnames(data),
        col=c("grey50","grey80"),ylab="Fréquence")
```

Figure 16. Exemple de diagramme en barres avec `barplot`



L'option `beside=TRUE` permet de ne pas superposer les différentes catégories, ce qui est parfois plus lisible.

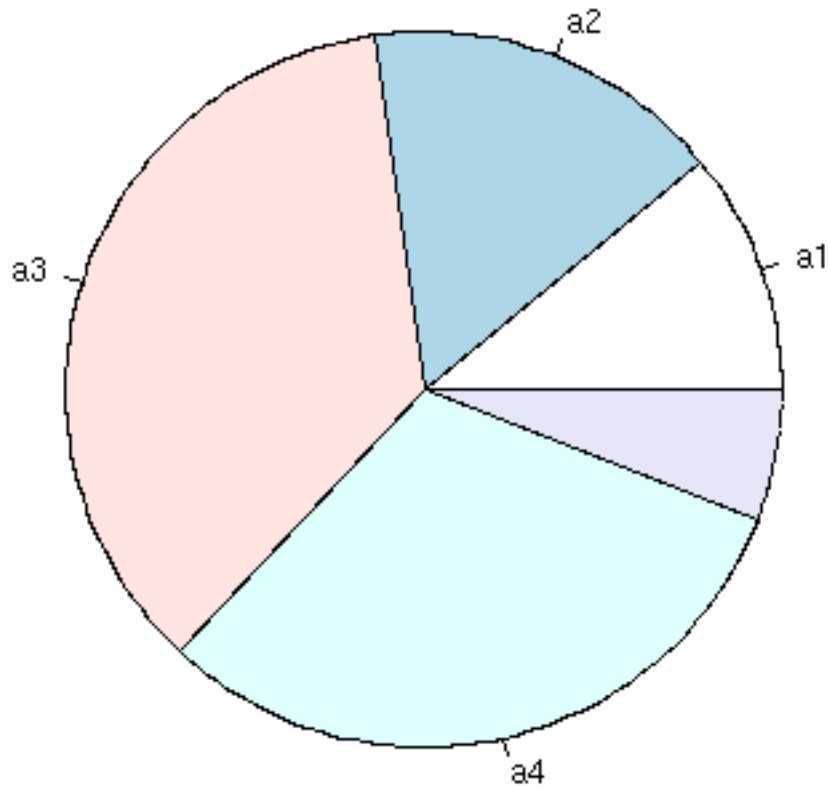
Mode de financement principal



On peut également utiliser une boîte à camembert pour représenter les effectifs ventilés sur les modalités de la variable d'intérêt :

```
names(x1) <- levels(A)
pie(x1/100)
```

Figure 17. Les diagrammes circulaires (“camemberts”)



Remarquons cependant que ces représentations dans lesquelles les proportions relatives sont évaluées par des secteurs angulaires deviennent vite difficiles à analyser lorsqu'il y a beaucoup de modalités (cf. Cleveland (1985), page 264 ¹) ou lorsque l'on souhaite croiser différentes variables, et il est préférable d'utiliser des diagrammes en barres.

On peut utiliser cette représentation en barre dans le cas d'une variable continue ventilée sur différentes modalités :

```
y <- c(rnorm(50, mean=20), rnorm(50, mean=15, sd=3))
x1 <- gl(2, 50, 100, labels=paste("c", 1:2, sep=" "))
x2 <- factor(rep(1:4, 25), labels=letters[1:4])
y.means <- tapply(y, list(x1, x2), mean)
```

¹"Data that can be shown by pie charts always can be shown by a dot chart. This means that judgements of position along a common scale can be made instead of the less accurate angle judgements." This statement is based on the empirical investigations of Cleveland and McGill as well as investigations by perceptual psychologists. (?pie)

```

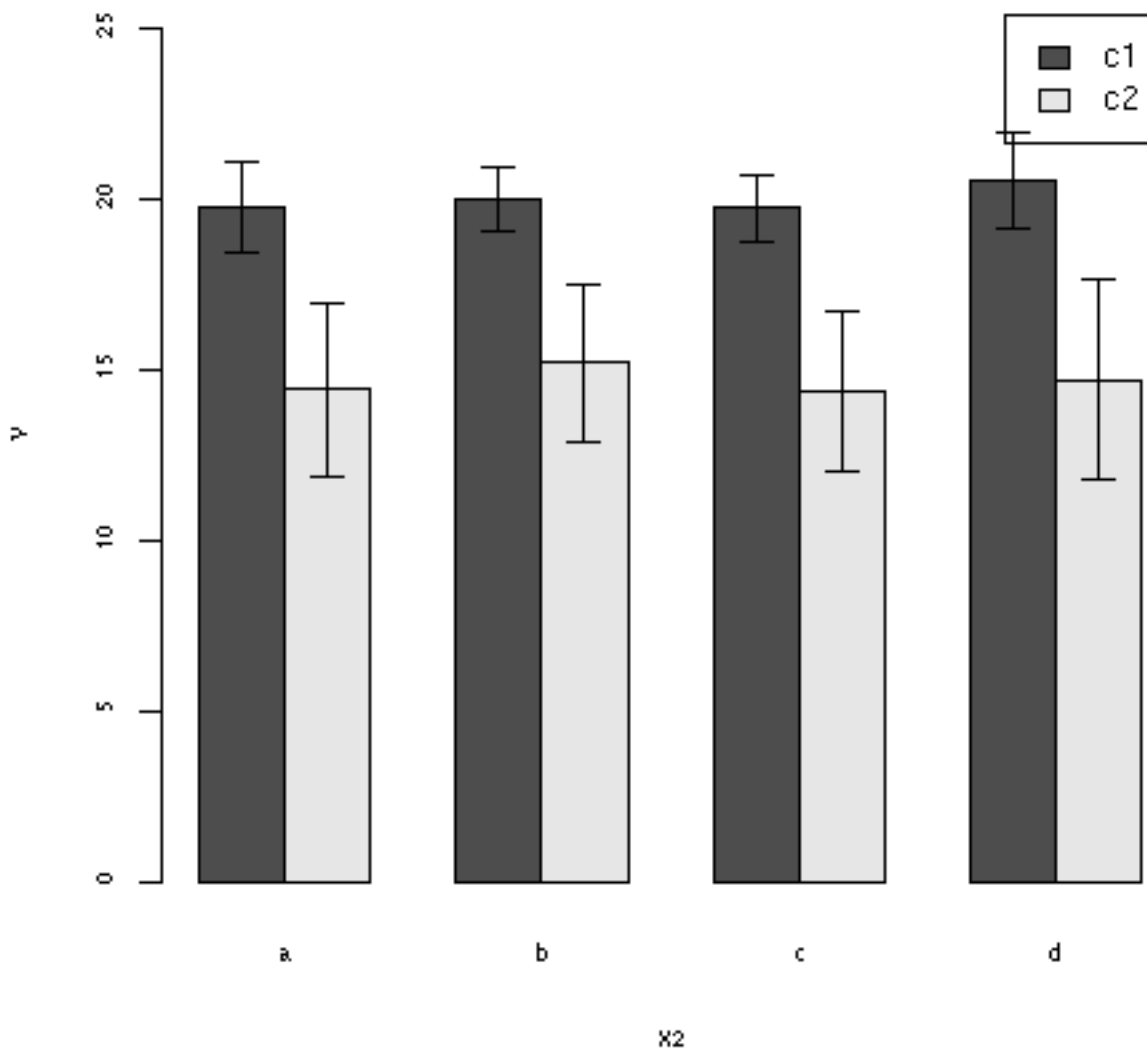
y.sd <- tapply(y,list(x1,x2),sd)

my.barplot <- function(data, err, ...) {
  tmp <- barplot(data, ...)
  if (length(err) != length(data))
    stop("la longueur de 'data' et 'err' ne correspond pas")
  for (i in 1:length(err))
    arrows(tmp[i],data[i]-err[i],tmp[i],data[i]+err[i],
          code=3,angle=90,length=0.08)
}

my.barplot(y.means,y.sd,beside=T,ylab="Y",xlab="X2",
  ylim=c(0,26),names.arg=levels(x2),legend.text=levels(x1),
  cex.axis=.7,cex.names=.7,cex.lab=.7)

```

Figure 18. Personnaliser la fonction barplot



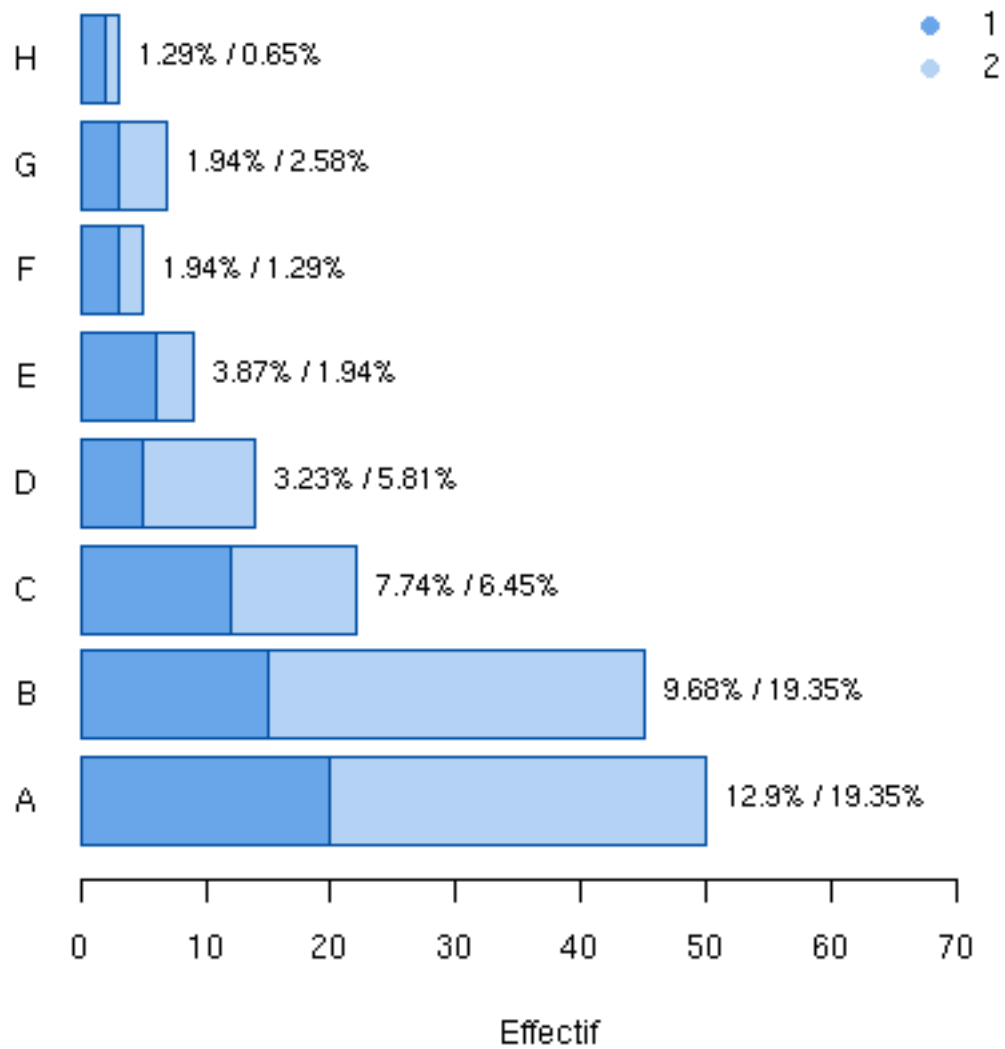
On peut encore pousser la personnalisation un peu plus loin :

```
a <- matrix(NA,nrow=8,ncol=2)
a[,1] <- rev(c(2,3,3,6,5,12,15,20))
a[,2] <- rev(c(1,4,2,3,9,10,30,30))
rownames(a) <- LETTERS[1:8]
colnames(a) <- c("1","2")
cols <- c(rgb(105/255,166/255,233/255),
          rgb(180/255,210/255,244/255),
          rgb(0/255,78/255,162/255))

a.prop <- round(a/sum(a)*100,2)
a.margin <- apply(a,1,sum)
x.offset <- 25

par(mar=c(5,6,4,2))
bp <- barplot(t(a),names.arg=rownames(a),horiz=T,col=cols[1:2],
             las=1,xlab="Effectif",main="",xlim=c(0,max(a.margin)+x.offset),
             border=cols[3])
legend("topright",c("1","2"),pch=rep(19,2),col=cols[1:2],bty="n")
labs <- paste(paste(as.character(a.prop[,1]),"%",sep=""),
             paste(as.character(a.prop[,2]),"%",sep=""),sep=" / ")
text(a.margin,bp,labs,pos=4,cex=.8)
```

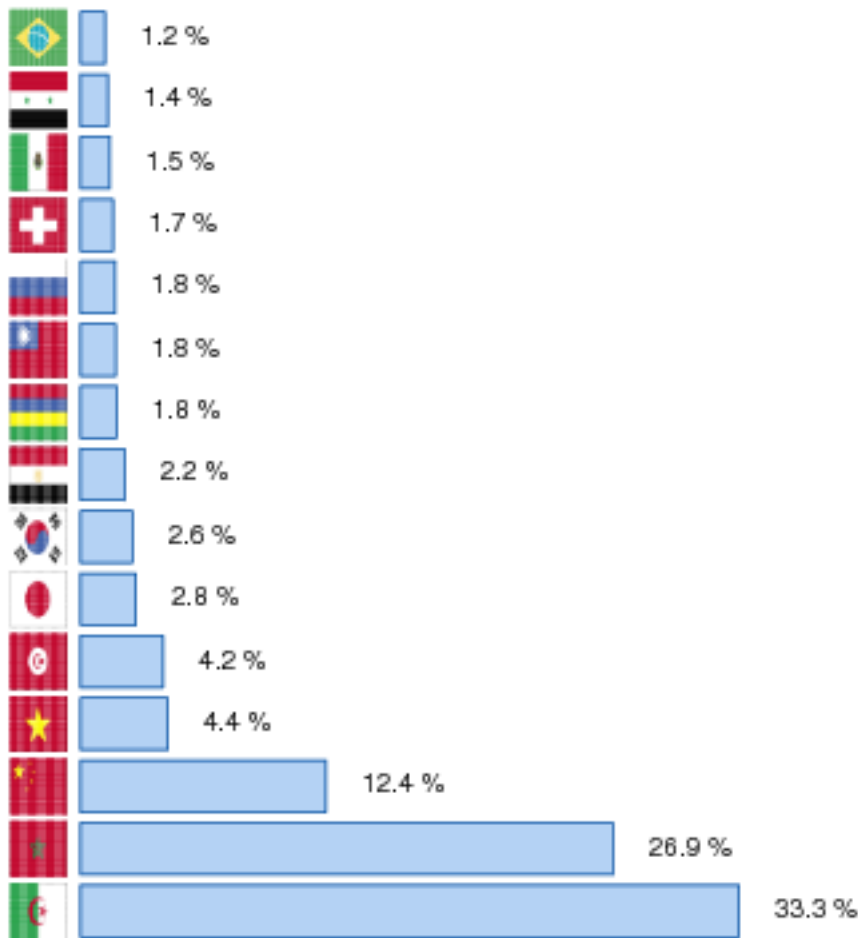
Figure 19. Personnaliser la fonction `barplot` (2)



ou écrire une fonction qui permettent de représenter conjointement une distribution exprimée en termes d'effectifs et de fréquences : `my.barplot.R`.

Autre type de personnalisation possible :

Pays de naissance des candidats



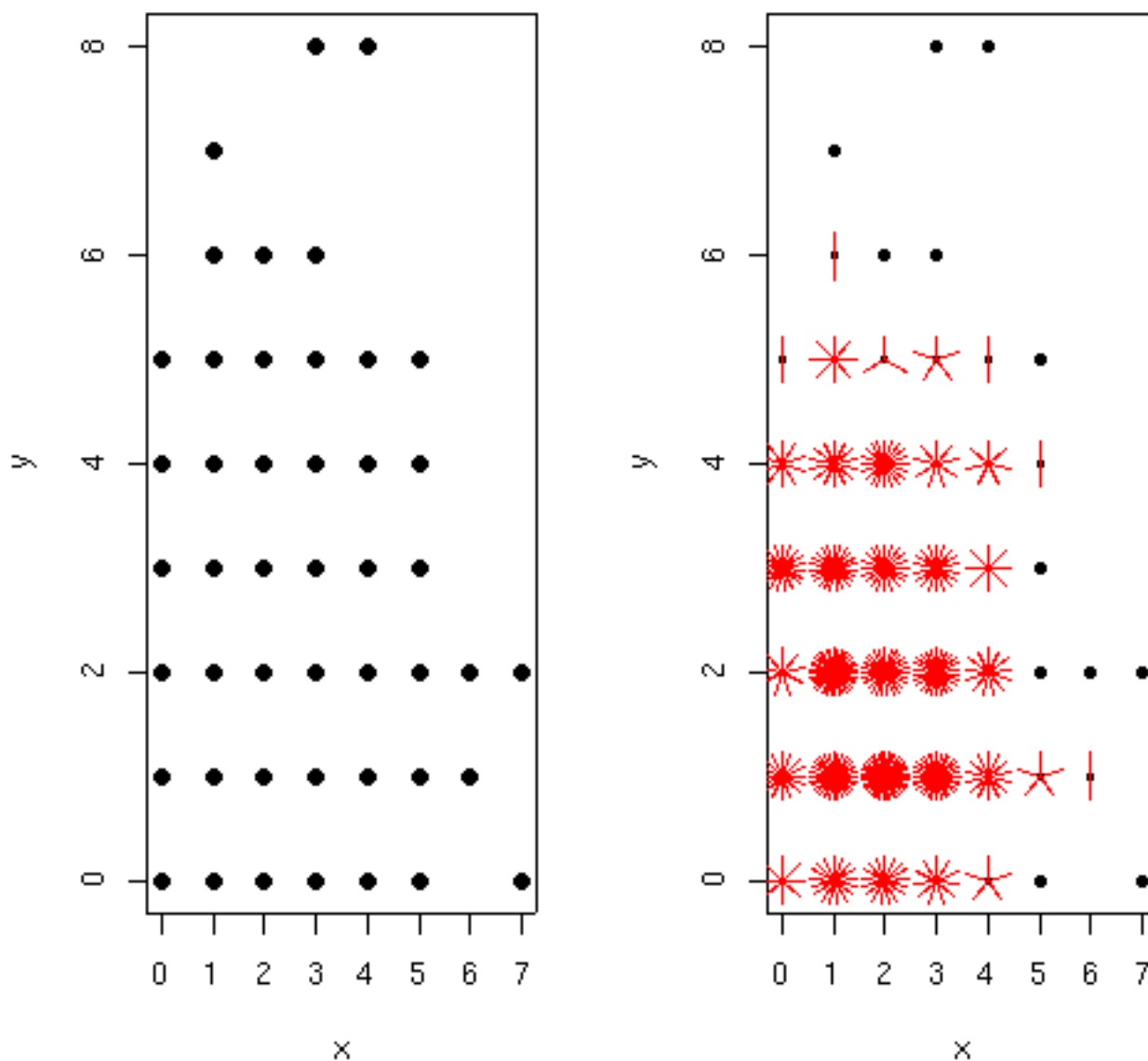
Représentations conjointes

Il existe plusieurs fonctions utiles dans le cadre de la représentation des effectifs ventilés sur les modalités de deux variables qualitatives. Ces méthodes de représentation permettent d'éviter l'éventuel superposition des points telle qu'on peut l'observer dans un simple diagramme de dispersion.

Par exemple, la fonction `sunflowerplot` représente les effectifs sous forme de diagramme étoilé pour chacun des croisements des modalités des variables :

```
x <- rpois(500,lambda=2)
y <- rpois(500,lambda=2)
layout(t(matrix(1:2)))
plot(x,y,pch=19)
sunflowerplot(x,y,pch=19)
```

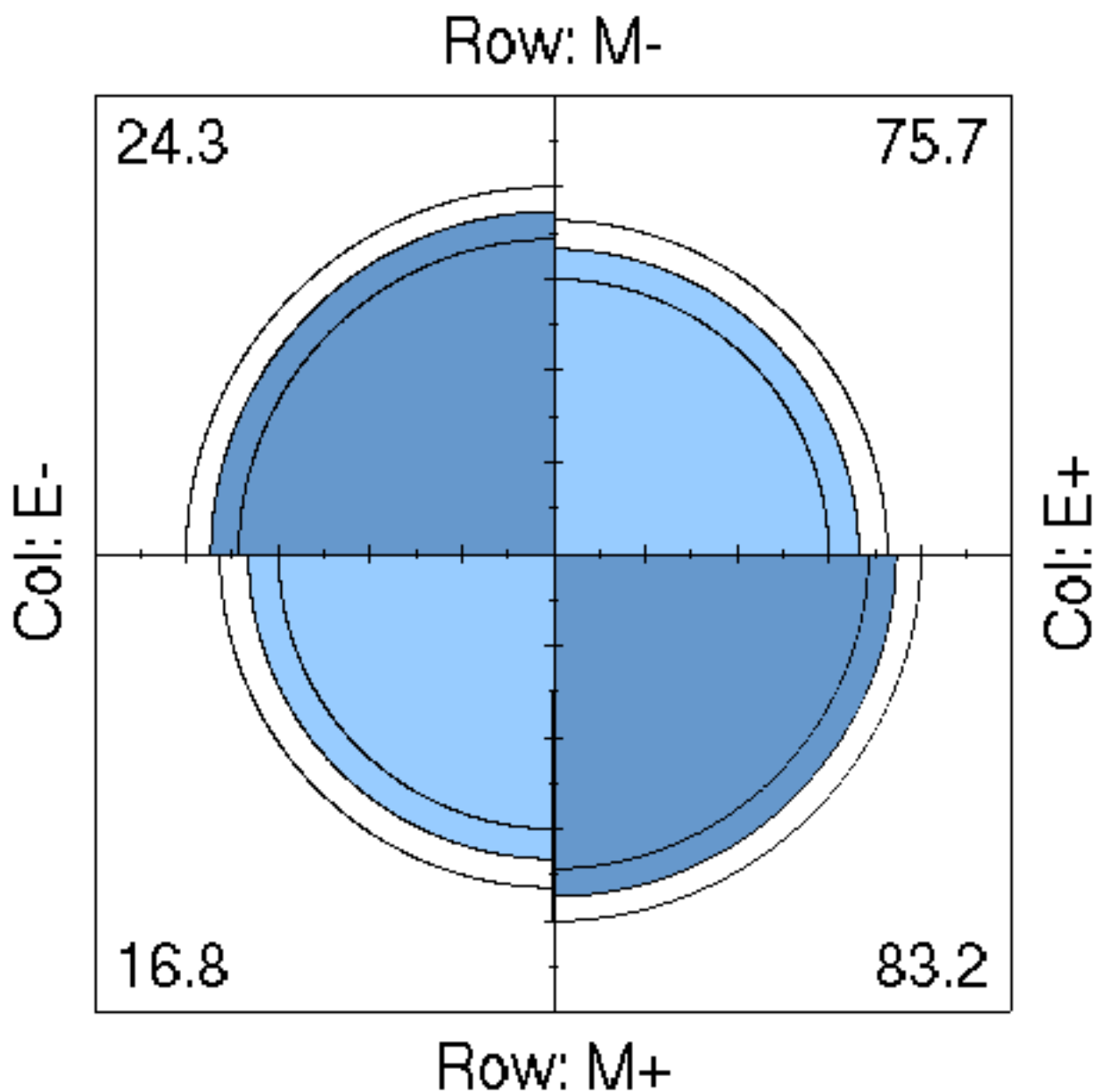
Figure 20. Diagramme en étoiles



Dans le cas de deux variables dichotomiques (e.g. en épidémiologie, exposition vs. maladie), on peut utiliser une représentation en quarts de cercle de l'association entre les deux variables :

```
x <- c(24.3,75.7,16.8,83.2)
a <- matrix(x,nr=2,byrow=T)
var1 <- c("E-","E+")
var2 <- c("M-","M+")
dimnames(a) <- list(var2,var1)
fourfoldplot(a)
```

Figure 21. Diagramme en quart de cercle

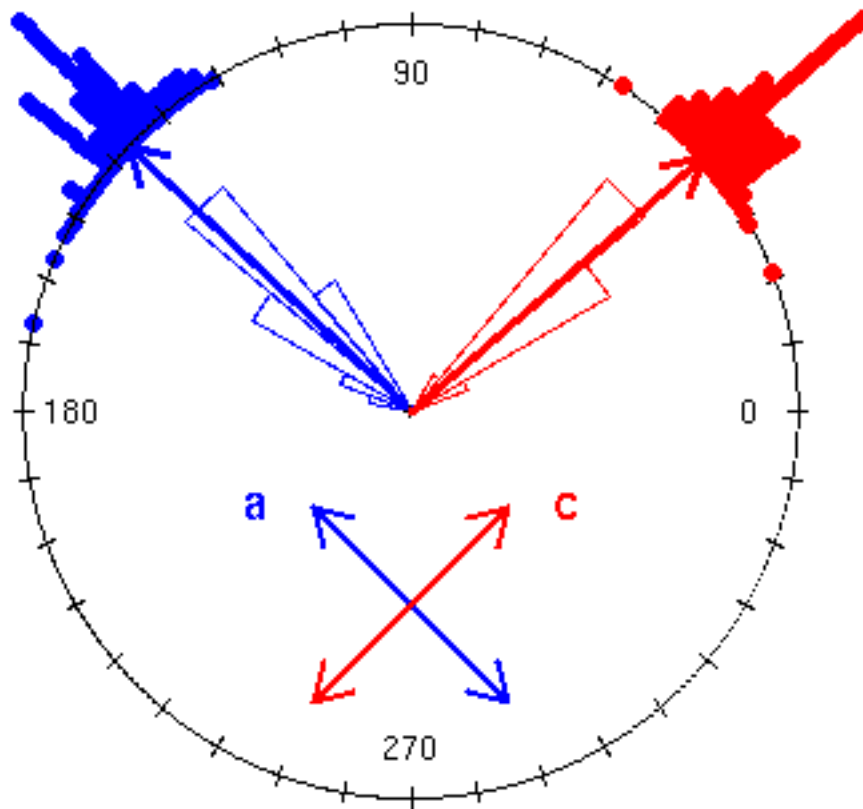


Représentations conditionnelles

Voici un exemple d'utilisation d'un diagramme de fréquence conditionnel (deux variables qualitatives), d'après l'exemple trouvé dans l'aide en ligne (?dotchart) :

```
dotchart(VADeaths, main = "Death Rates in Virginia - 1940")
```

Figure 22. Distribution de données circulaires



Données de survie

A faire.

Séries temporelles

Avec les données contenues dans l'image `ex_st.Rdata` [`data/ex_st.Rdata`], on peut illustrer quelques-unes des facilités de **R** pour la personnalisation des graphiques.

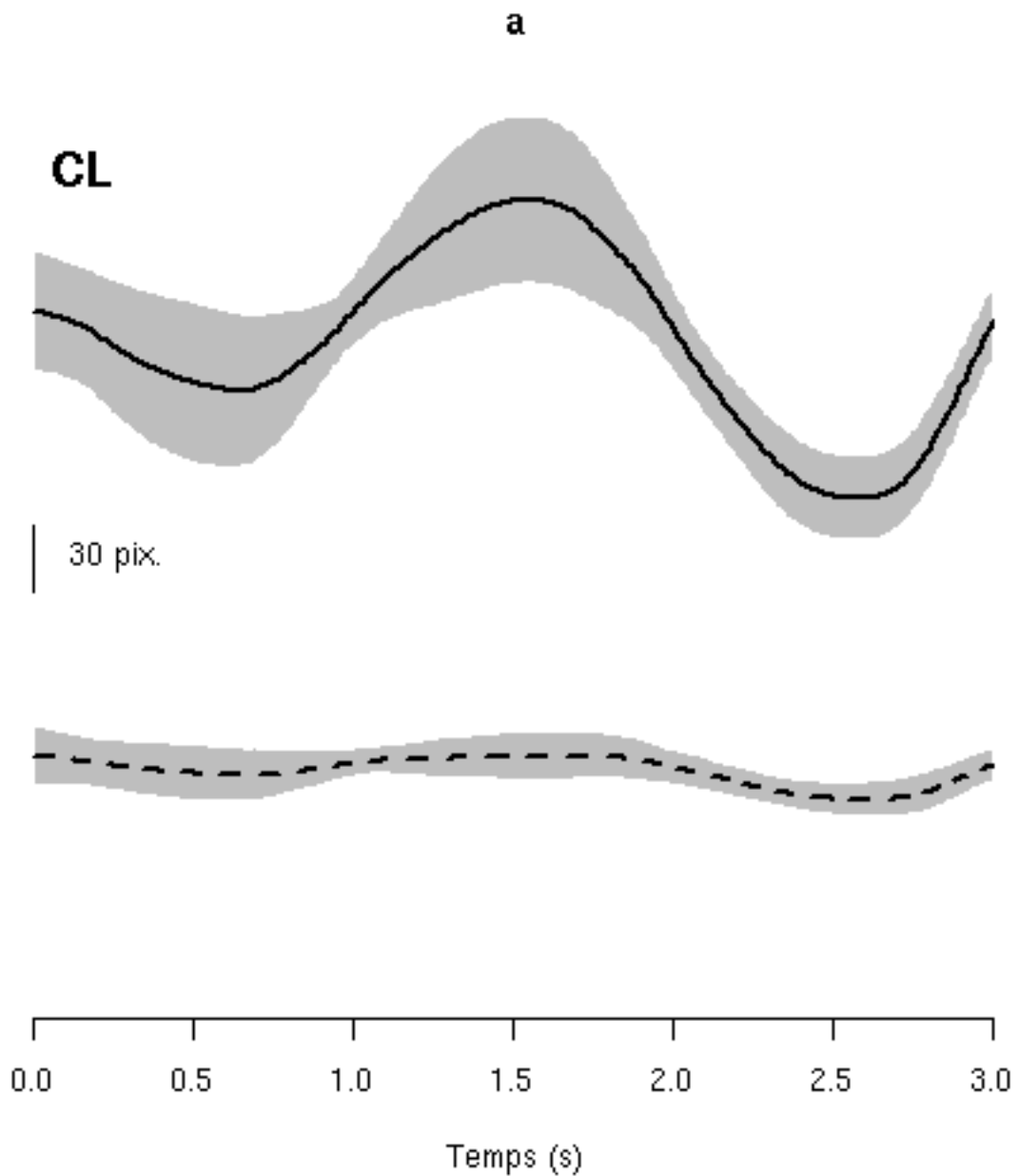
```
load("ex_st.Rdata")
offset <- 100
xx <- 1:length(a.mean[,1])*0.01 # le temps en secondes (@100 Hz)
plot(xx,a.mean[,1],type="n",xlab="Temps (s)",ylab="",ylim=c(-200,200),
```

```

main="a",axes=F)
xsda <- c(xx,rev(xx))
ysda <- c(a.mean[,1]+(a.sd[,1]/2)+offset,rev(a.mean[,1]-(a.sd[,1]/2)+offset))
polygon(xsda,ysda,col="grey",border=NA)
xsdb <- c(xx,rev(xx))
ysdb <- c(b.mean[,1]+b.sd[,1]-offset,rev(b.mean[,1]-b.sd[,1]-offset))
polygon(xsdb,ysdb,col="grey",border=NA)
lines(xx,a.mean[,1]+offset,type="l",lwd=2)
lines(xx,b.mean[,1]-offset,type="l",lwd=2,lty=2)
axis(1)
xx <- 0
yy <- 0
lwb <- yy-15;
upb <- yy+15;
arrows(xx,lwb,xx,upb,length=0,angle=90,code=3)
text(xx+.05,yy,labels="30 pix.",pos=4)
text(0,180,"CL",cex=1.5,font=2,pos=4)

```

Figure 23. Evolution d'une série chronologique



Données spatiales

A faire.

Divers

La librairie `pixmap` permet d'afficher des images au format `ppm` sur une fenêtre graphique.

Par exemple, on peut afficher un ensemble de drapeaux sous forme de matrice dans un graphique :

```
library(pixmap)
my.path <- "../flags.pnm"
scale <- 1/50
```

```

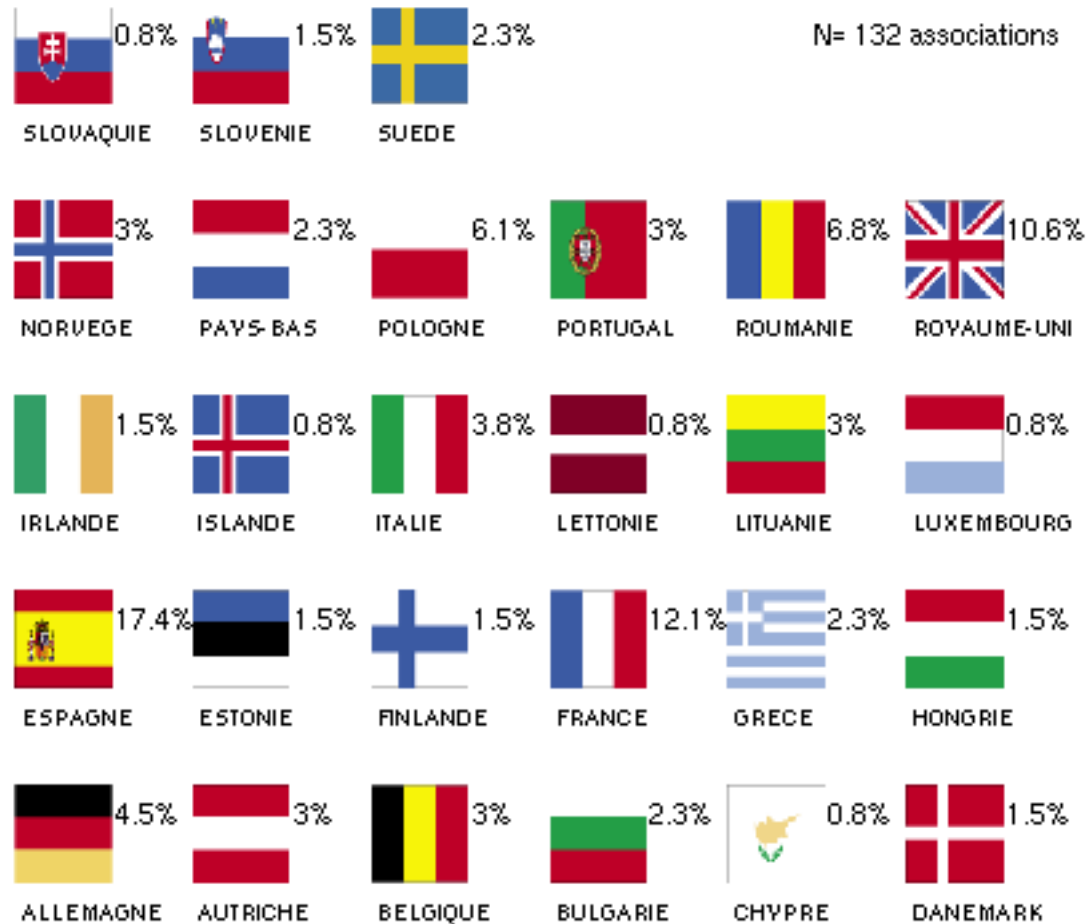
dx <- 2.75
dy <- 1
flags <- list.files(path=my.path, full.names=T)
N <- length(pays.prop)

par(mar=c(2,2,2,2))
plot(seq(0,16),seq(0,16),ylim=c(0,8),type="n",axes=F)
k <- 0
for (i in 1:5) {
  for (j in 1:6) {
    k <- k+1
    if (k == N) break # prevents from exceeding max number of flags (28 < 30)
    x <- read.pnm(flags[grep(names(pays.prop[k]),flags)])
    # Note: all flags are 81x54 pixels
    addlogo(x, px=c(j-dx+1.75*j,j-dx+1.75*j+1.5),py=c(i-dy+.5*i,i-dy+.5*i+.75))
    text(j-dx-.25+1.75*j,i-dy+.5*i-.25,names(pays.prop[k]),pos=4,cex=.7)
    text(j-dx+1.75*j+1.25,i-dy+.5*i+.5,paste(pays.prop[k],"%",sep=""),pos=4,cex=.8)
  }
}
title(main="Localisation géographique des associations")
text(12,7,paste("N=",sum(pays),"associations",sep=" "),pos=4,cex=.8)

```

Figure 24. Représenter des cartes grâce à `pixmap`

Localisation géographique des associations



Références

[Venables2002] Venables, W.N. & Ripley, B.D. (2002). *Modern Applied Statistics with S*. Springer-Verlag.

[Everitt2005] Everitt, B. (2005). *An R and S-PLUS Companion to Multivariate Analysis*. Springer-Verlag.

[Cleveland1993] William S. Cleveland (1993). *Visualizing Data*. Hobart Press.

[Wilkinson1999] Leland Wilkinson (1999). *The Grammar of Graphics*. Springer-Verlag.

[Friendly2000] Michael Friendly (2000). *Visualizing Categorical Data*. SAS Publishing.

[Murrell2005] Paul Murrell (2005). *R Graphics*. Chapman & Hall/CRC.

Index

B

barplot, 25

D

density, 5

G

ggplot, 19

ggplot2, 19

H

hexbin, 16, 18

hist, 4

L

lattice, 10

P

pairs, 13

pixmap, 37

S

stripchart, 23

sunflowerplot, 32

survey, 17