R: An overview

www.r-project.org

Didier Chauveau

MAPMO - UMR 6628 - Université d'Orléans

CaSciModOT Tours – Juillet 2010

Plan de l'exposé

1. Généralités
   - Origines et environnement
   - Notions de base
2. Quelques illustrations
   - Analyse exploratoire de données
   - Statistiques Inférentielles
3. Un bref panorama des packages
   - Un exemple : mixtools

Genèse : Les origines de R

- Initié par **R**oss Ihaka et **R**obert Gentleman (1993)
  (University of Auckland, New Zealand)
- Implémentation du langage “S”
  (Chambers et al., Bell labs 1984–1998)
- ACM Software System Award, 1998.
- De facto devenu :
  - le standard de la recherche en “computational statistics” et exploration de données
  - l’environnement idéal pour vulgariser les nouvelles méthodologies en statistique

Quelques avantages de R

- Gratuit (GPL2) et Open Source (écrit en `C`)
- Multi-plateforme (UNIX, LINUX, MacOS X, Windows...)
- Développé et maintenu par les meilleurs experts en "Statistical Computing"= The R Core Team :
  17 créateurs/développeurs de S et R et statisticiens de renom (B. Ripley, L. Tierney, J. Chambers...)
- Langage interactif, orienté objet, extensible
- Pensé pour l'exploration et la modélisation de données
  (à la différence de e.g. MATLAB, Scilab,...)
- Interface simple vers du code `C`, `Fortran`,...

“Philosophie” de R

Une analyse statistique implique :

- Exploration des données (résumés numériques, graphiques)
- Choix des outils guidés par la visualisation données
- Possibilité d’adapter les outils existants : l’écriture de fonctions est naturelle

Extension du système R et sites CRAN

Un ensemble de fonctions qui implémentent une méthode gagne a être organisé sous forme de **package**

- R inclut un environnement de :
  - développement (peut contenir du code R, `C`, `FORTRAN`)
  - documentation (LaTeX-like)
  - test de package
- un package peut *dépendre* (“hériter”) d’autres packages

Les packages “validés” sont accessibles via les sites miroirs

CRAN = **C**omprehensive **R** **A**rchive **N**etwork

2447 packages à ce jour ($\approx$ 1400 en 2009. . .)

Objets R

Structures communes à tous les langages “de haut niveau” :
`vector`, `matrix`, `array`, `list`,...

Classes d’objets spécialisées pour les statistiques, eg :

- `factor` : vecteur à valeurs *modalités*
  = facteur qualitatif (e.g. CSP, sexe,...)
- `data.frame` : liste de vecteurs de même longueur, de *classes* quelconques
  = table individus-caractères en statistique

“Toy example” : Données *State Facts*

Source : *Bureau of the Census, US (1977)*.
Un des nombreux jeux de données inclus dans R

Sélection de 7 variables :

Etat	Noms des “individus” = 50 états (code à 2 lettres)
Pop	Population estimée en 1975
Revenu	Revenu moyen par habitant
Apb	taux d’analphabétisme (en % de population)
Meurtre	taux de criminalité pour 100 000 habitants
Diplome	% de population de niveau équivalent au Bac
Region	région d’appartenance des états (Northeast, South, North Central, West)

Exemple : Données “State Facts”

Table individus-caractères (lignes $i = 1, \ldots, 5$) :

```
> states[1:5,]
     Pop Revenu Apb Meurtre Diplome Region
AL  3615   3624 2.1    15.1    41.3  South
AK   365   6315 1.5    11.3    66.7   West
AZ  2212   4530 1.8     7.8    58.1   West
AR  2110   3378 1.9    10.1    39.9  South
CA 21198   5114 1.1    10.3    62.6   West

> class(states)
[1] "data.frame"

> class(states$Region)       # Region est un facteur
[1] "factor"
```

Classes et Méthodes

Langage orienté objet

Les fonctions peuvent disposer de méthodes adaptées aux *classes* de leur argument

Exemple : résumé de structures

```
> summary(states)   # methode de summary pour data.frame
```

Méthodes associées

```
> methods(summary)
  [1] summary.aov           summary.aovlist      summary.asp
  [4] summary.connection    summary.data.frame   summary.Dat
  [7] summary.default       summary.ecdf*        summary.fac
  ...
```

Résumés numériques

```
> summary(states)

    Meurtre          Diplome                 Region
 Min.   : 1.400   Min.   :37.80   North_Central:12
 1st Qu.: 4.350   1st Qu.:48.05   Northeast    : 9
 Median : 6.850   Median :53.25   South        :16
 Mean   : 7.378   Mean   :53.11   West         :13
 3rd Qu.:10.675   3rd Qu.:59.15
 Max.   :15.100   Max.   :67.30

> print(cor(states[,-6]),3)

             Pop Revenu    Apb Meurtre Diplome
Pop       1.0000  0.208  0.108   0.344 -0.0985
Revenu    0.2082  1.000 -0.437  -0.230  0.6199
Apb       0.1076 -0.437  1.000   0.703 -0.6572
Meurtre   0.3436 -0.230  0.703   1.000 -0.4880
Diplome -0.0985  0.620 -0.657  -0.488  1.0000
```

Résumés numériques (2)

```
> attach(states)    # rend visibles les variables
> sd(Apb)   # Apb au lieu de states$Apb
[1] 0.6095331
```

Statistiques par niveaux de facteur(s) :

```
> by(states[,2:5], Region, mean) # fonction mean param.

Region: North_Central
    Revenu         Apb    Meurtre     Diplome
4611.08333    0.70000    5.27500   54.51667
------------------------------------------------------------
Region: Northeast
     Revenu         Apb     Meurtre      Diplome
4570.222222    1.000000    4.722222   53.966667
------------------------------------------------------------
[etc...]
```

Recodage d'une variable numérique en facteur

Classes égales avec modalités spécifiées

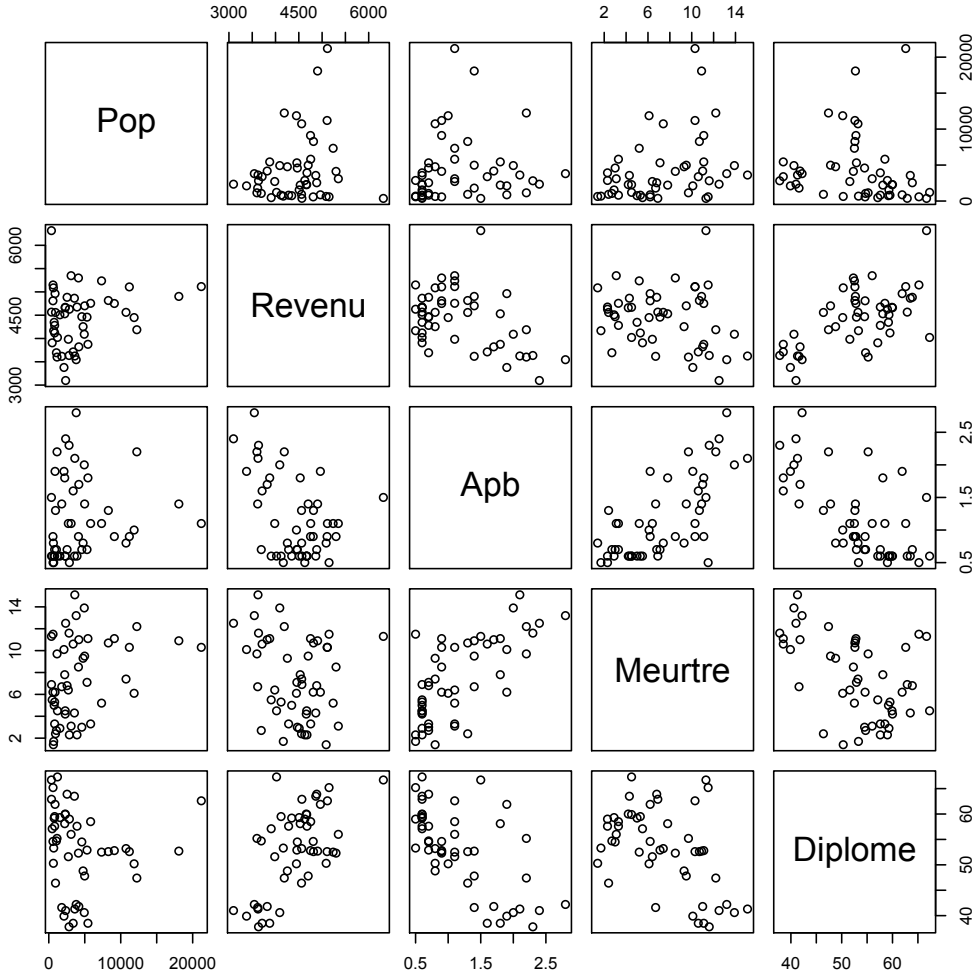
```
FD <- cut(Diplome,breaks=3,
    labels=c("faible","moyen","fort"))
table(FD)     # table de fréquence (loi empirique)
FD
faible  moyen   fort
    12     20     18
```

Discrétisation en classes égales en probabilité (empiriques)

```
q <- quantile(Diplome,probs=seq(0,1,1/3)) # 3 classes
FDq <- cut(Diplome,breaks=q,
        labels=c("faible","moyen","fort"))
table(FDq)
FDq
faible  moyen   fort
    16     17     16
```

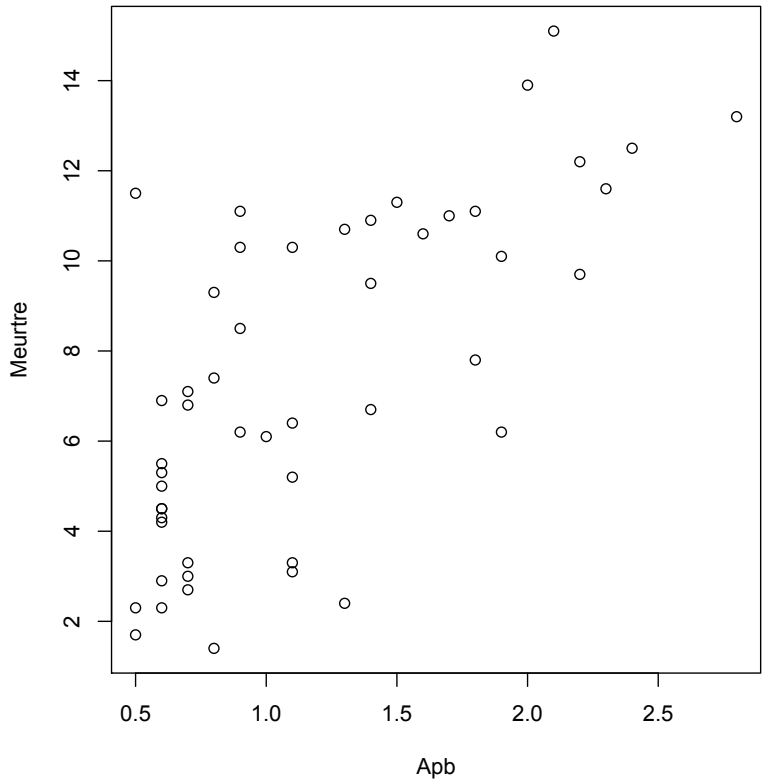
Méthode plot.data.frame

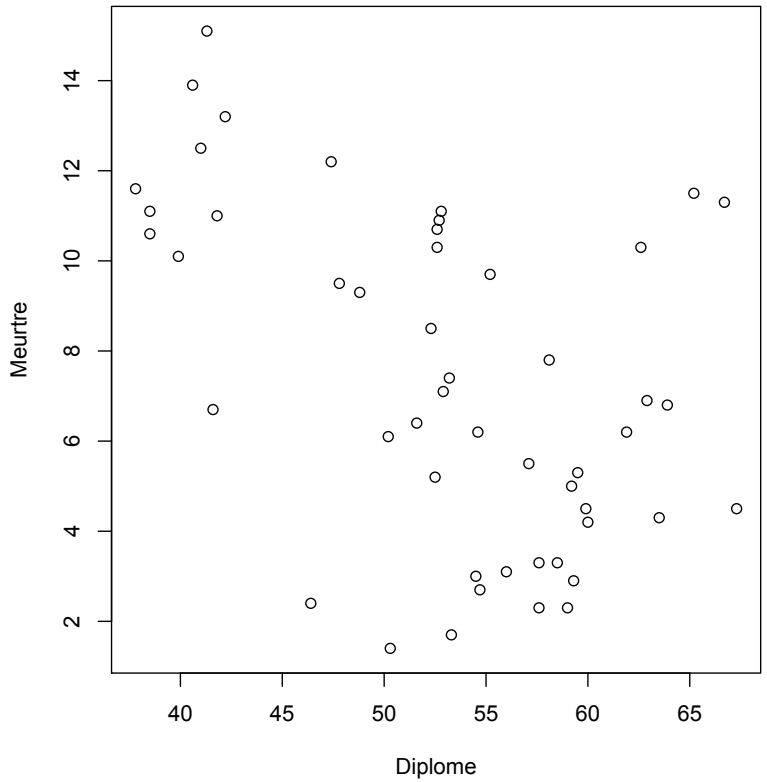
```
plot(states[,-6]) # tout sauf Region
```



Méthode plot.formula

```
par(mfrow=c(1,2))                    # subplots
plot(Meurtre ~ Apb + Diplome) # formule en R
```





Résumés graphiques

Nuages avec labels individus ou modalités d'un facteur

Facteur Region avec modalités en 1 car

```
R2 <- factor(Region,labels=c("C","E","S","W"))
plot(Apb,Meurtre,type="n",xlab="Analphabétisme")
text(Apb,Meurtre,R2,col=as.numeric(R2))
```

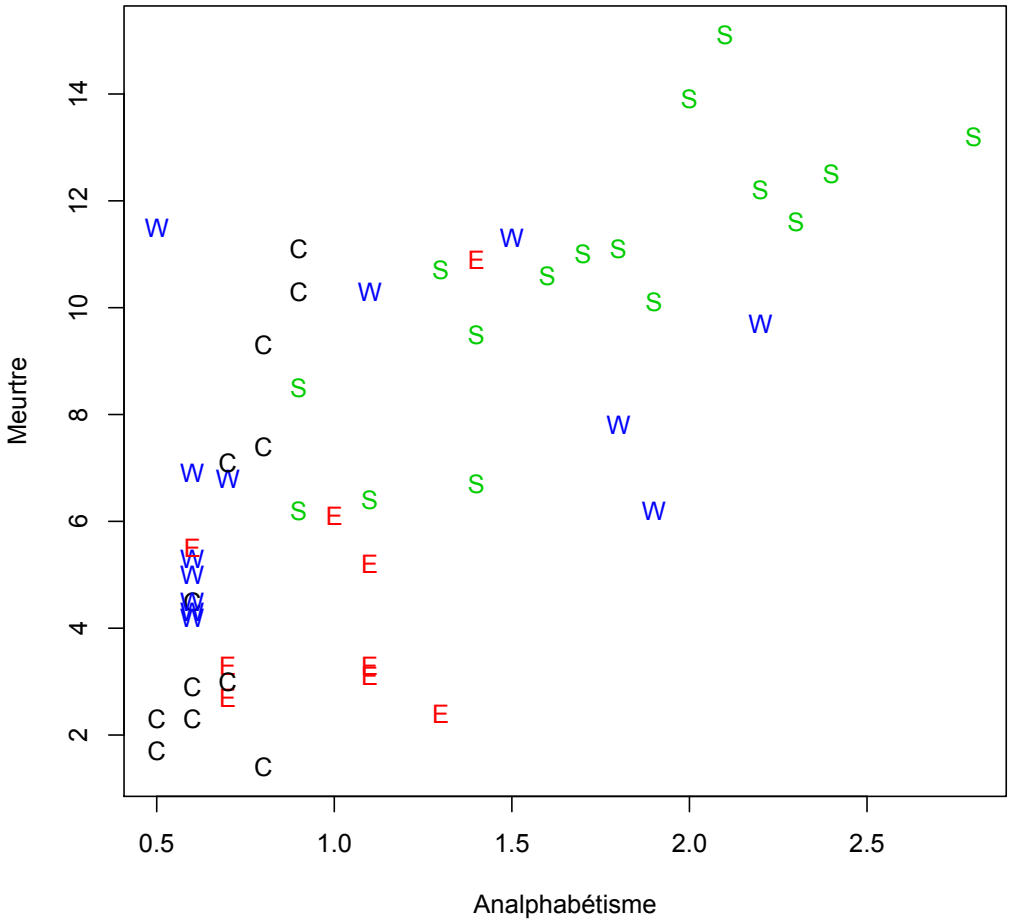
Chargement du package `ade4` méthodes factorielles ACP, AFC,. . . (D. Chessel, Lyon I)

```
library(ade4)
```

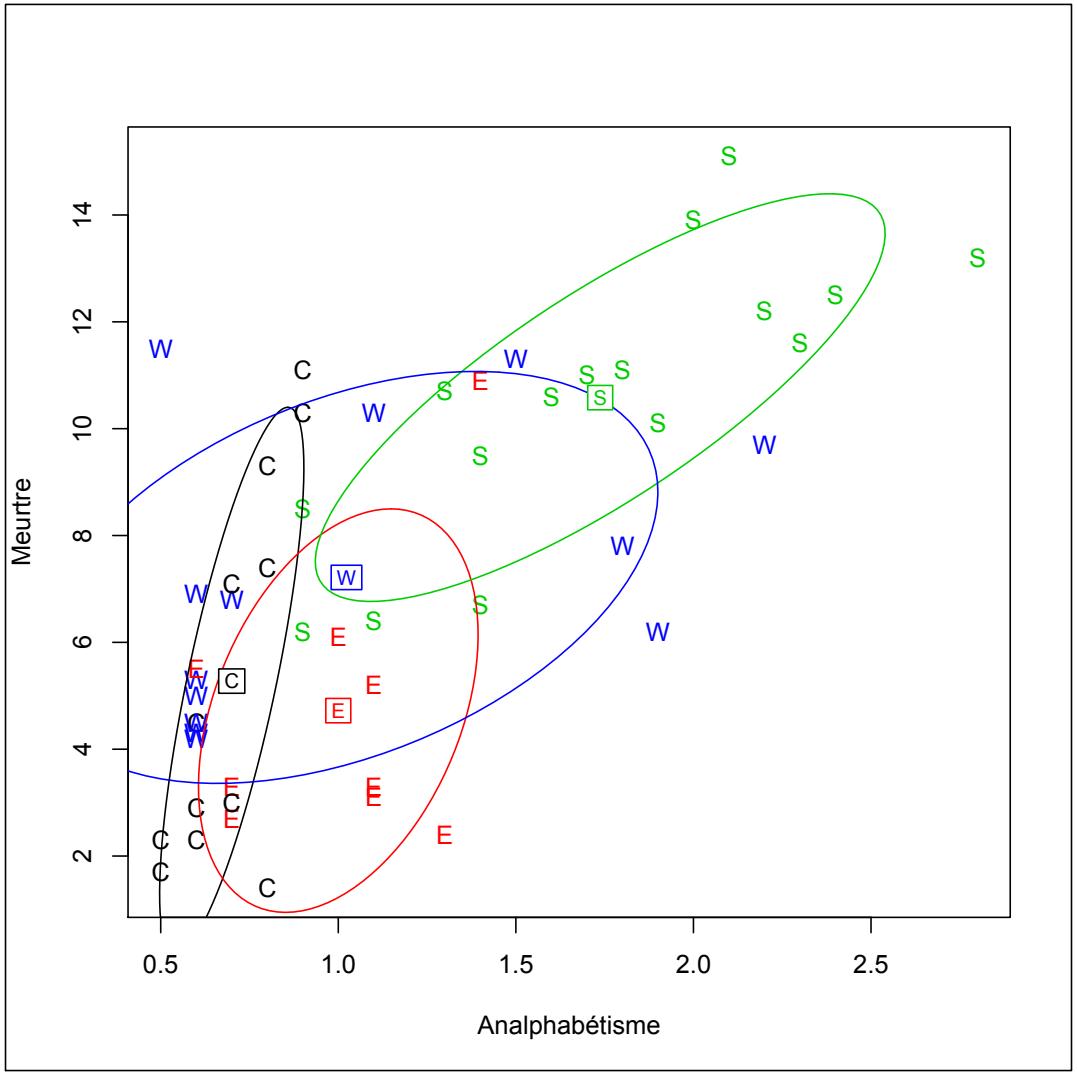
Ajout des barycentres et ellipses d'inertie par modalités d'un facteur

```
s.class(data.frame(Apb,Meurtre),R2, cstar=0,
        cpoint=0, clabel=0.8,col=1:length(levels(R2)),
        axesell=FALSE, add.plot=TRUE)
```

Nuage avec modalités du facteur Region. . .

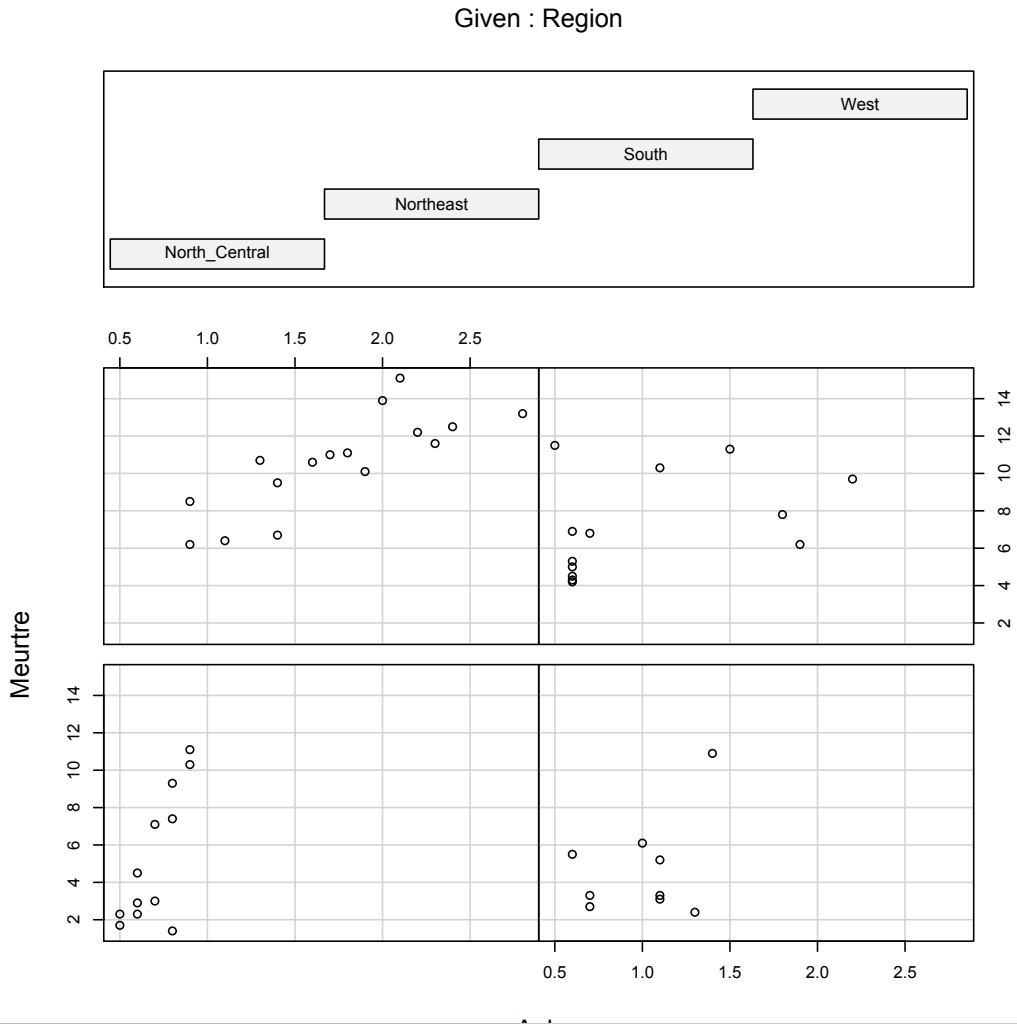


+ barycentres et ellipses d'inertie



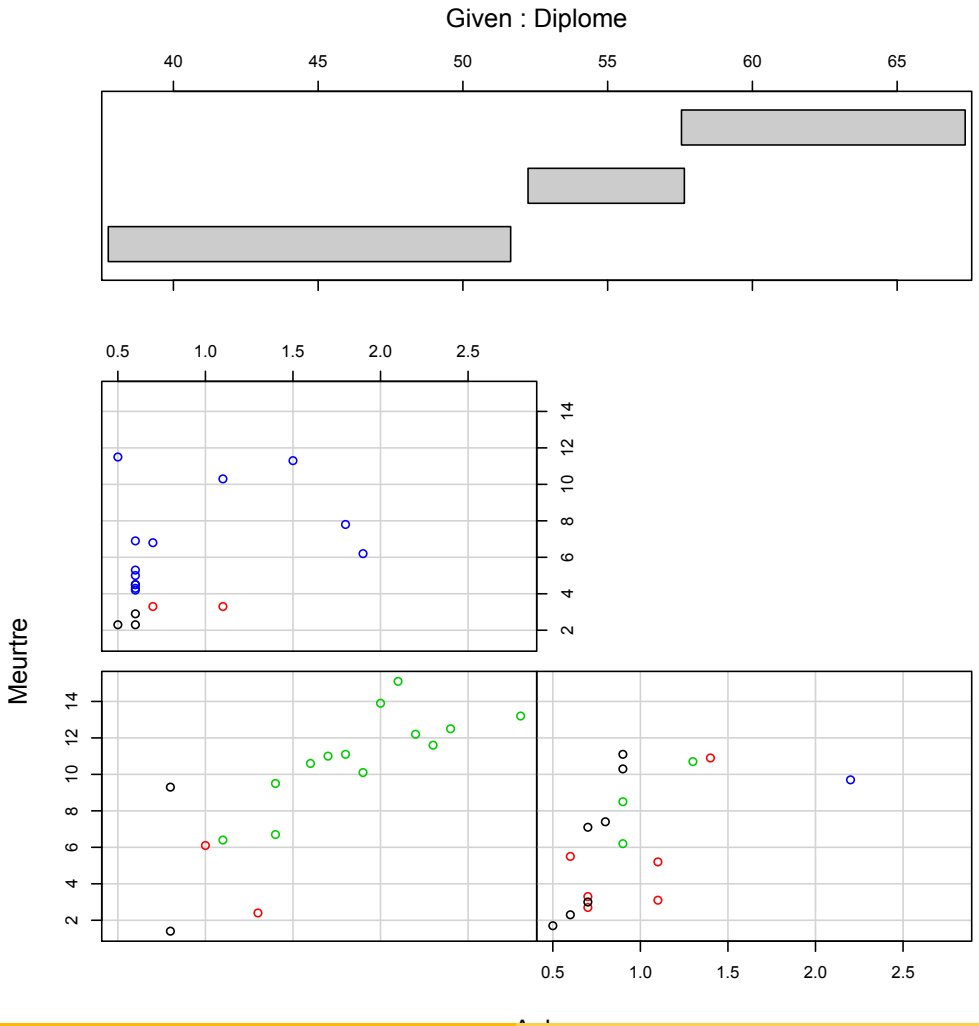
Nuage conditionnel à un facteur

```
coplot(Meurtre ~ Apb | Region)
```



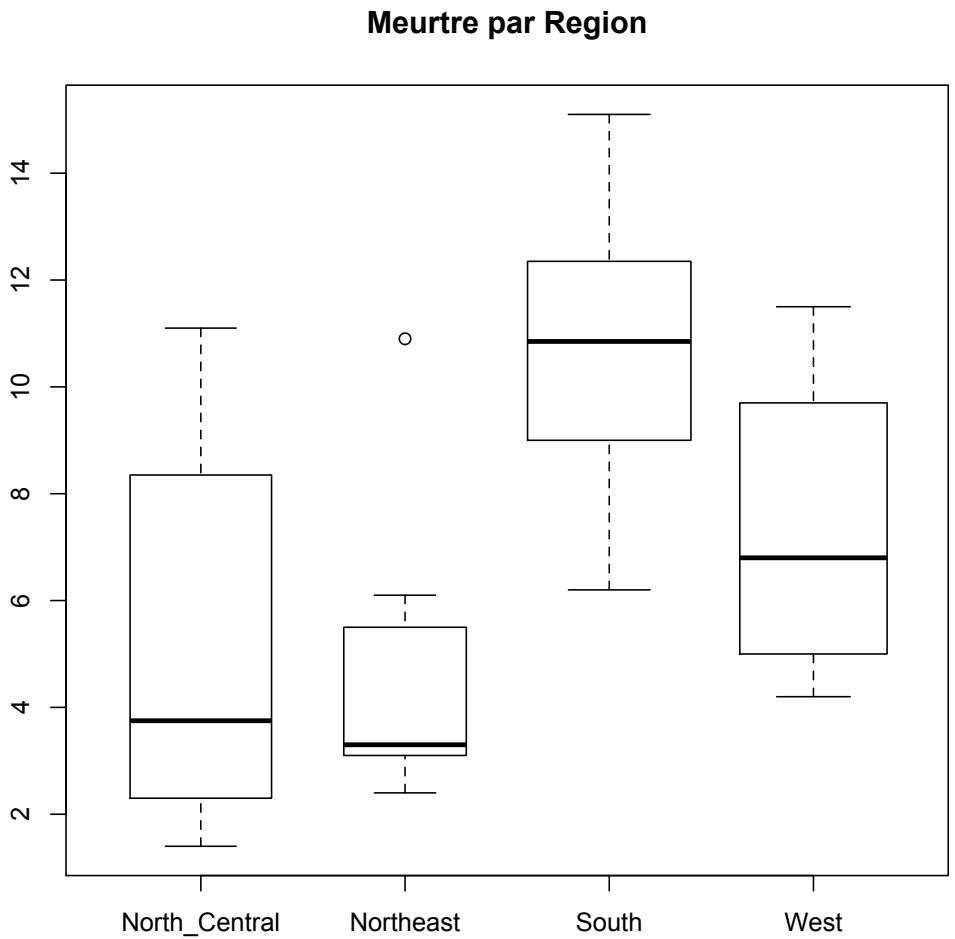
Nuage conditionnel à une var. discrétisée

```
coplot(Meurtre~Apb|Diplome,num=3,over=0,col=R2)
```



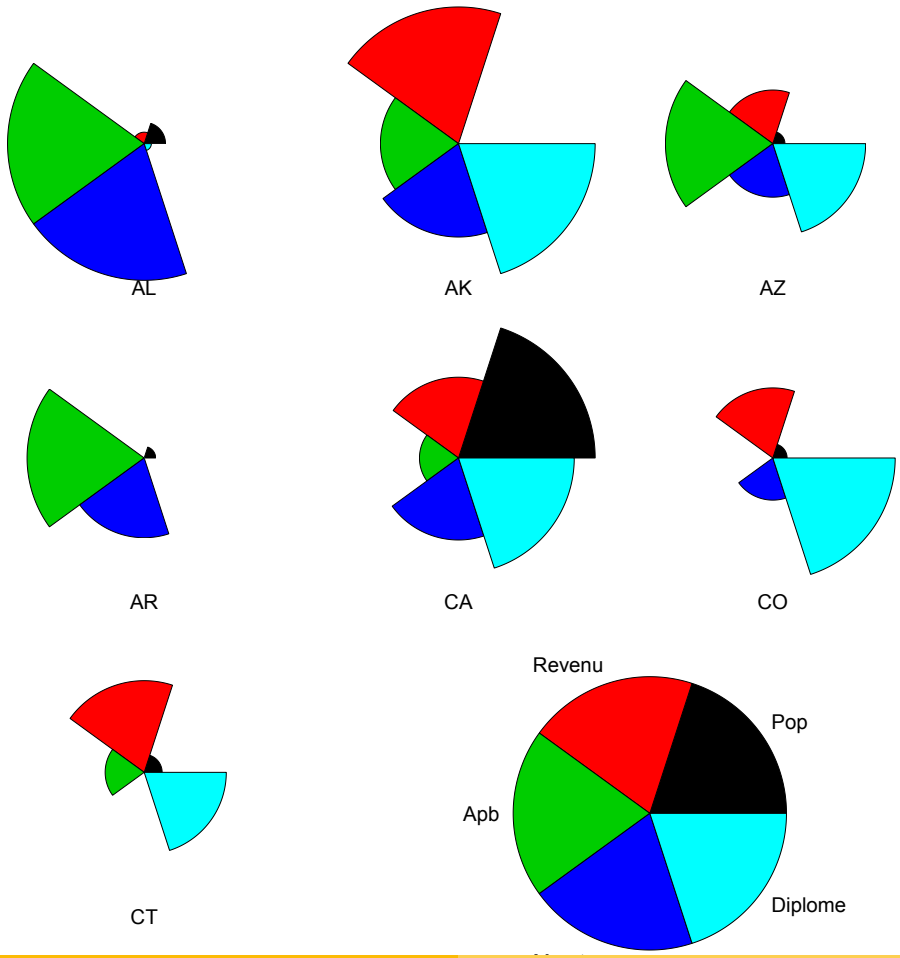
Boxplots par modalités d'un facteur

```
boxplot(Meurtre ~ Region,varwidth=T)
```



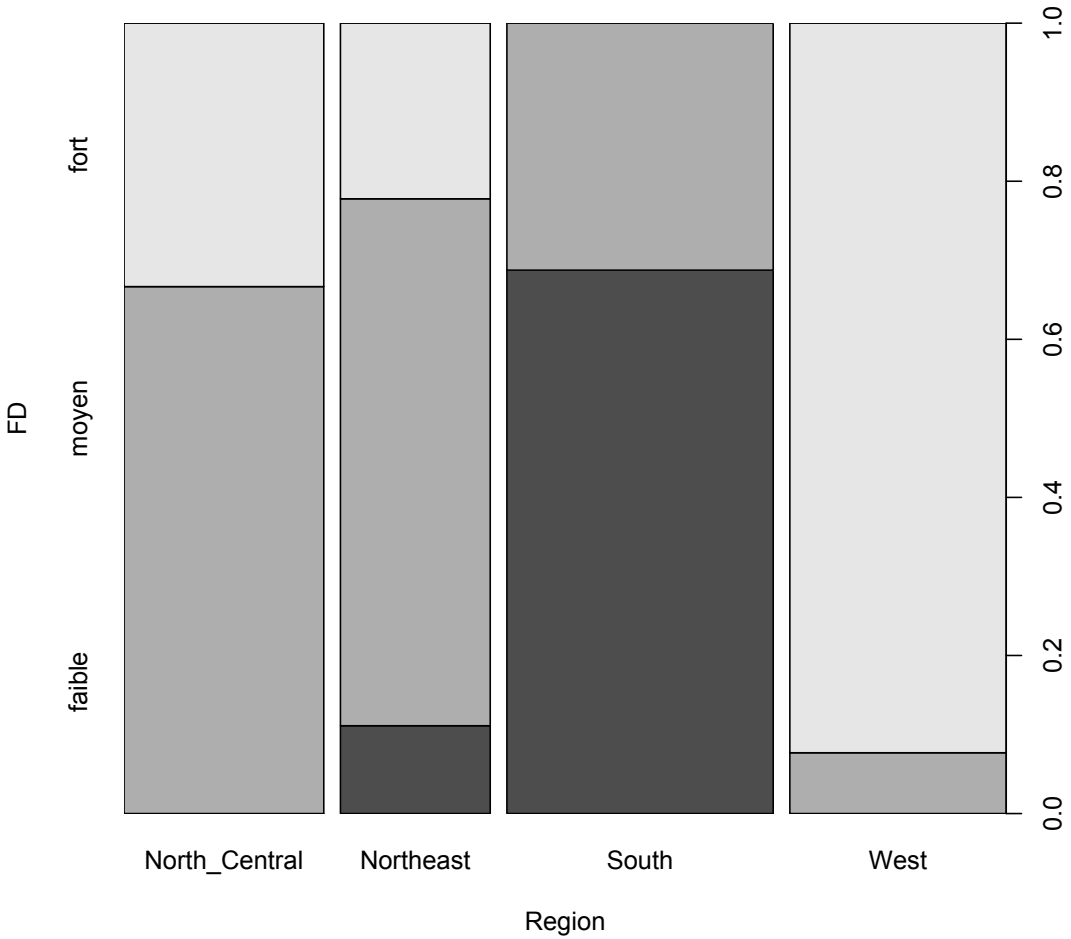
Graphiques “radar” ou “segments”

```
stars(states[1 :10,-6],key.loc=c(6,2),draw=T)
```



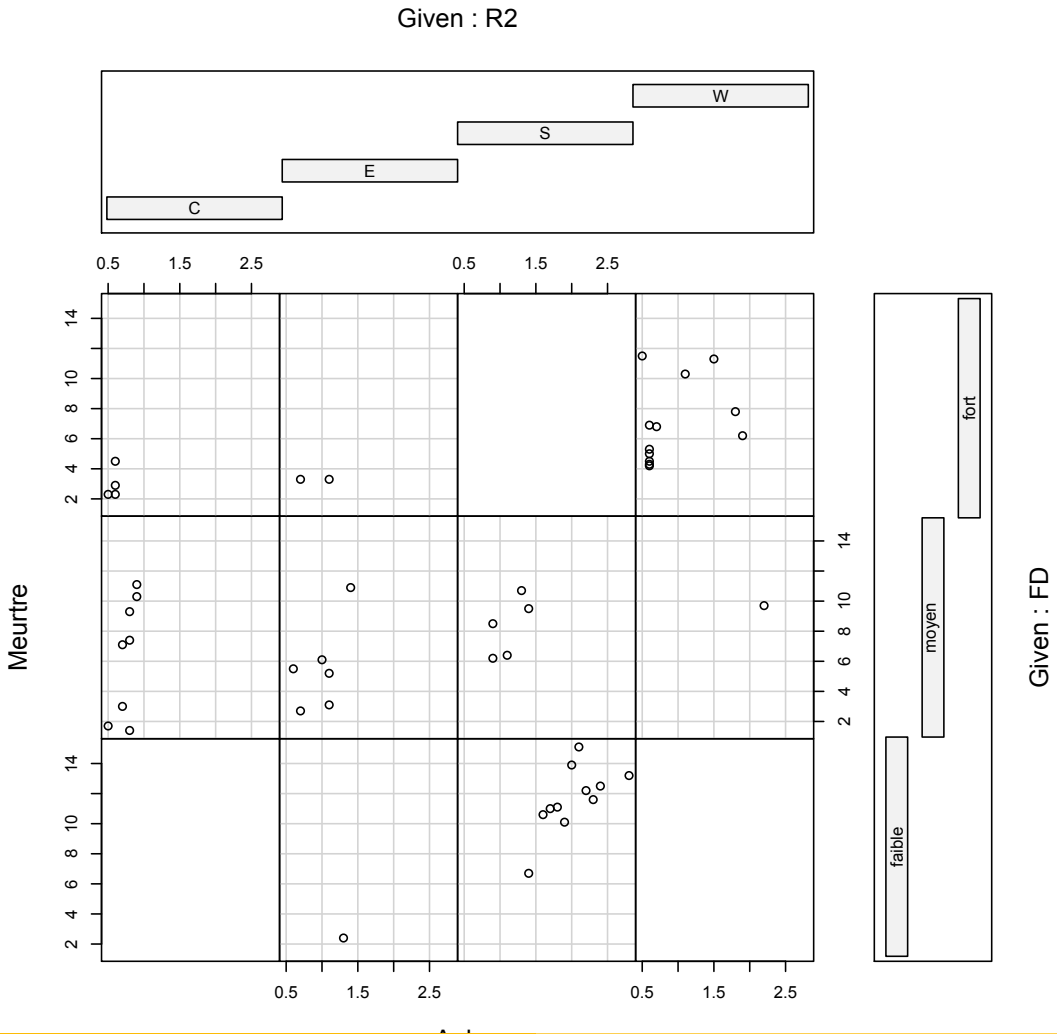
Histogrammes de profils conditionnels

```
spineplot(Region,FD)
```



Coplots a deux facteurs

```
coplot(Meurtre ~ Apb | R2 + FD)
```



Modèles statistiques dans R

Dans la distribution standard (package `stats`) :

- Lois usuelles, simulation, bootstrap...
- Tests standards (Student, Kolmogorov, Mann-Whitney,...)
- Modèles linéaires (anova, régression)
- Modèles non linéaires, mixtes, analyse de survie...

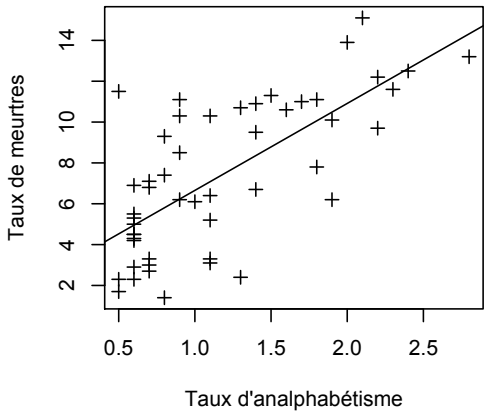
La spécification des modèles se fait par une *formule*, e.g.,

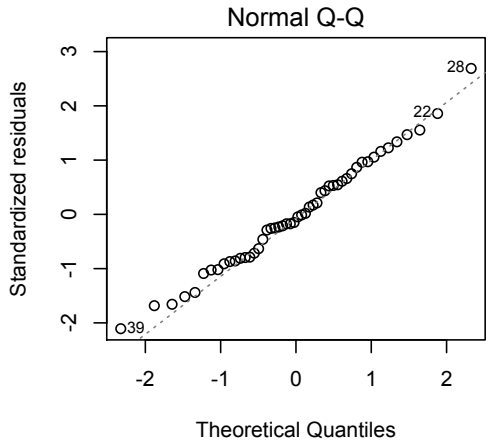
$$\text{Meurtre} = \beta + \alpha \text{Apb} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

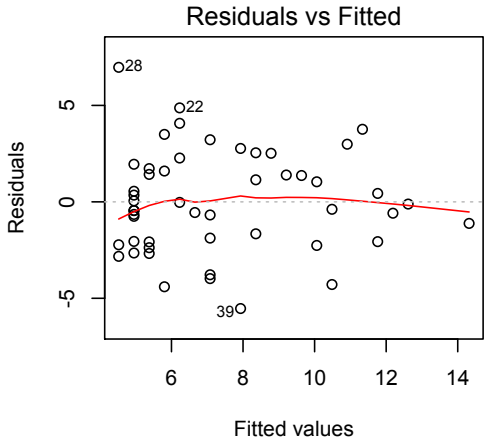
```
rg <- lm(Meurtre ~ Apb)  # class(rg) = "lm"
> rg                     # method print.lm
Coefficients:
(Intercept)          Apb
      2.397        4.257
```

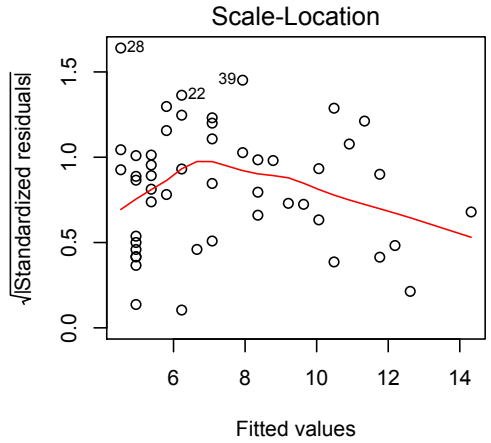
Régression linéaire simple

```
par(mfcol=c(2,2)); plot(Apb, Meurtre, pch=3)
abline(reg=rg); plot(rg, which=1:3)
```

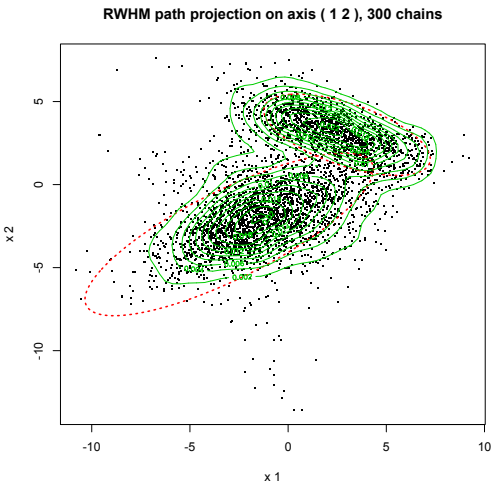


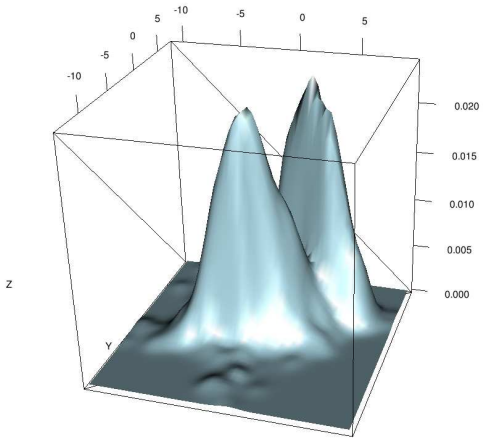


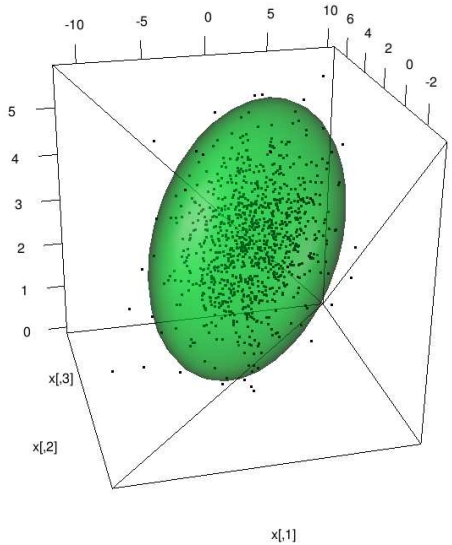


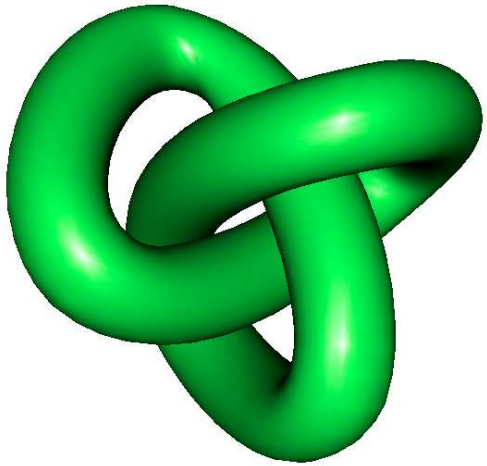


Petite gallerie graphique (R appelle OpenGL)



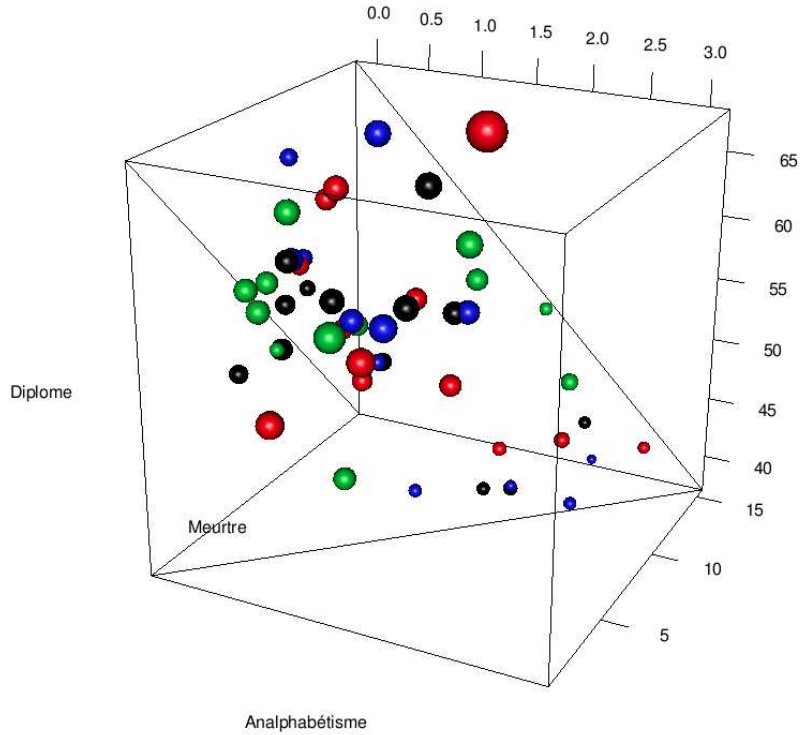






Scatterplot 3D avec

- coloration par Région
- rayon $\propto$ Revenu



Beaucoup plus dans les packages

Quelques exemples en vrac :

- Clustering et modèles de mélange
- Géostatistique (N. Saby, INRA Orléans)
  - Analyse multivariée sous contrainte spatiale
  - Analyse bayésienne des corrélations spatiales
  - Modèles mixtes avec corrélation spatiale
- Statistique et Génome
  - Multiple testing, FWER, FDR,. . .
  - Projet BioConductor (380 packages !)
- Méthodes MCMC (Metropolis, Gibbs sampler,. . .)
- Graphiques 3D OpenGL
- ...

Pour s'y retrouver : CRAN Task Views

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Design, Monitoring, and Analysis of Clinical Trials
Cluster	Cluster Analysis & Finite Mixture Models
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
gR	gRaphical Models in R
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis

Package mixtools

mixtools: Tools for analyzing finite mixture models

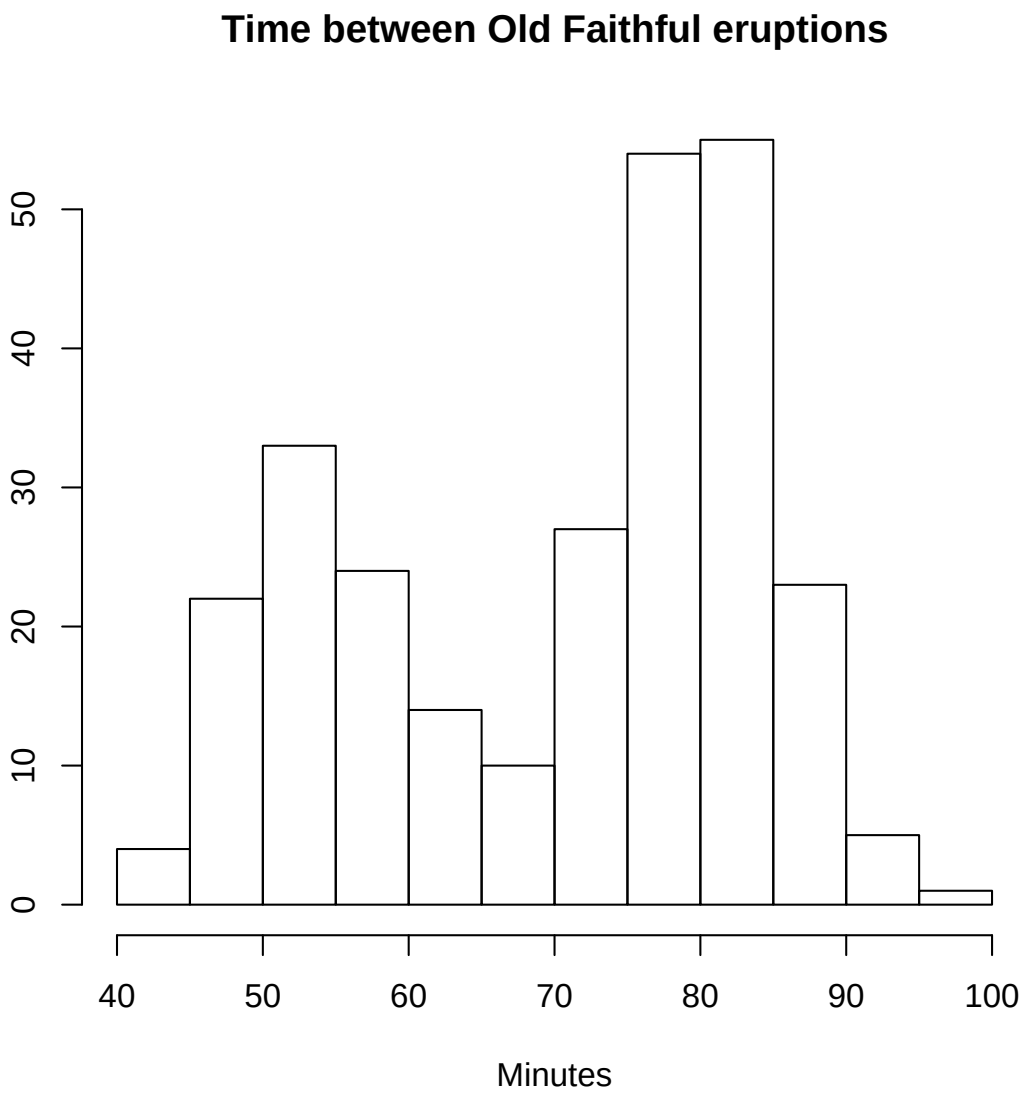
A collection of R functions for analyzing finite mixture models. This package is based upon work supported by the National Science Foundation under Grant No. SES-0518772.

Version:	0.4.4
Depends:	boot, MASS, R (≥ 2.0.0)
Published:	2010-06-28
Author:	Derek Young Tatiana Benaglia Didier Chauveau Ryan Elmore Tom Hettmansperger David Hunter Hoben Thomas Fengjuan Xuan
Maintainer:	Derek Young <dsy109 at stat.psu.edu>
License:	GPL (≥ 2)
Citation:	mixtools citation info
In views:	Cluster, Distributions
CRAN checks:	mixtools results

Downloads:

Package source:	mixtools_0.4.4.tar.gz
MacOS X binary:	mixtools_0.4.4.tgz
Windows binary:	mixtools_0.4.4.zip
Reference manual:	mixtools.pdf
Vignettes:	mixtools for mixture models

Données univariées : Old Faithful wait times (mn)



from www.nps.gov/yell

- Bimodalité évidente
- Modèle de mélange ?
- Allure gaussienne des composantes ?

Mélange fini (univarié ici) de lois

Objectif : Estimer λ_j et f_j (ou f_{ξ_j}) à partir d'un éch. i.i.d. de

Cas paramétrique :

$$g(x) = \sum_{j=1}^{m} \lambda_j f_{\xi_j}(x)$$

paramètre $\theta = (\boldsymbol{\lambda}, \boldsymbol{\xi})$,
e.g. $f_{\xi_j} \equiv \mathcal{N}(\mu_j, \sigma_j^2)$

Cas semi-paramétrique :

$$g(x) = \sum_{j=1}^{m} \lambda_j f(x - \mu_j)$$

paramètre $\theta = (\boldsymbol{\lambda}, \boldsymbol{\mu}, f)$,
f densité sur $\mathbb{R}$ "libre"

Un des apports de mixtools :

Proposer des algorithmes EM pour des modèles avec le minimum de contraintes sur les formes paramétriques des f_j (poids des queues et formes libres,...)

Exemple de documentation incluse (html, pdf,. . .)

```
> ?spEMsymloc
```

spEMsymloc {mixtools} R Documentation

Semiparametric EM-like Algorithm for univariate symmetric location mixture

Description

Returns semiparametric EM algorithm output (Bordes et al, 2007, and Benaglia et al, 2009) for location mixtures of univariate data and symmetric component density.

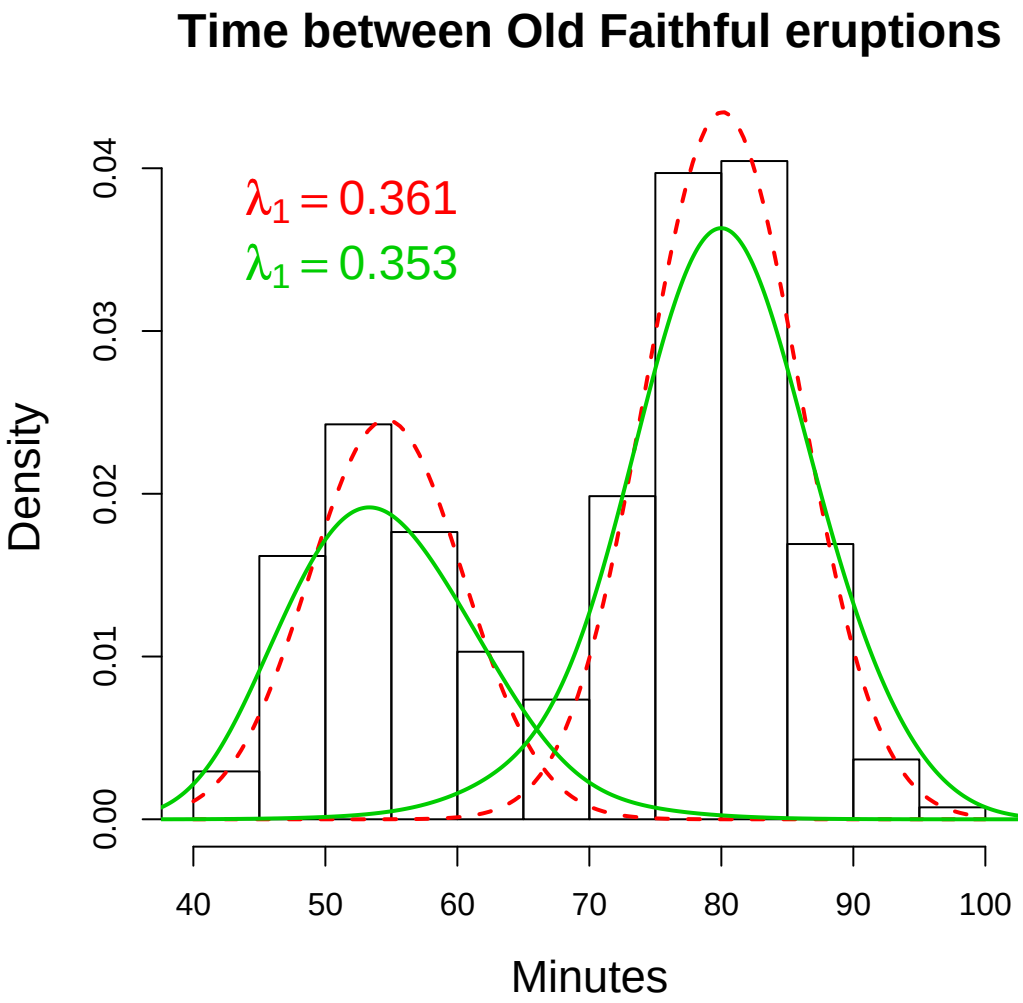
Usage

```
spEMsymloc(x, mu0, bw = bw.nrd0(x), h=bw, eps = 1e-8, maxiter = 100,
    stochastic = FALSE, verbose = FALSE)
```

Arguments

`x`	A vector of length *n* consisting of the data.
`mu0`	Either a vector specifying the initial centers for the kmeans function, and from which the number of component is obtained, or an integer *m* specifying the number of initial centers, which are then choosen randomly in kmeans.
`bw`	Bandwidth for density estimation, equal to the standard deviation of the kernel density.
`h`	Alternative way to specify the bandwidth, to provide backward compatibility.
`eps`	Tolerance limit for declaring algorithm convergence. Convergence is declared before `maxiter` iterations whenever the maximum change in any coordinate of the `lambda` (mixing proportion estimates) and `mu` (means) vector does not exceed `eps`.
`maxiter`	The maximum number of iterations allowed, for both stochastic and non-stochastic versions; for non-stochastic algorithms (`stochastic = FALSE`), convergence may be declared before `maxiter` iterations (see

Ajustement de mélanges sur les données Old Faithful



- EM mélange gaussien
-
```
R> data(faithful)
R> attach(faithful)
R> a<-normalmixEM(waiting,
R+     mu=c(55,80),
R+     sigma=5)
R> plot(a, density=T)
```
- $\hat{\mu} = (54.6, 80.1)$
- EM Semi-paramétrique

```
R> spEMsymloc(waiting,
R+     mu=c(55,80),h=4)
```

$\hat{\mu} = (54.7, 79.8)$

Quelques Références

- Chambers, J.M. (1998), *Programming with Data*
- Dalgaard, *Introductory Statistics with R*
- Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*
- R Wiki `http://wiki.r-project.org`
- R graphics gallery `http://addictedtor.free.fr/graphiques`
- and much more !. . .

En bref :