

# Bases statistiques pour la biologie

Christophe Ambroise et Cyril Dalmasso

École doctorale « *du génome aux organismes* »  
Université d'Évry

20-22 janvier 2014

[http://stat.genopole.cnrs.fr/~jchiquet/fr/initiation\\_R](http://stat.genopole.cnrs.fr/~jchiquet/fr/initiation_R)

## Equipe « Statistique & Génome »

<http://stat.genopole.cnrs.fr>



Cyril Dalmasso



Statistique

Christophe Ambroise



Statistique

`prenom.nom@genopole.cnrs.fr`

# Agenda (théorique) de la semaine

**Lundi** Statistiques descriptives, variables aléatoires et distributions

**Mardi** Estimation et intervalle de confiance

**Mercredi** Tests d'hypothèses (sous R)

## Cours (avec moi...)

1. Introduction
2. Individus, échantillon et statistiques descriptives
3. Introduction à R

*... et travaux dirigés!*

# Agenda (théorique) de la semaine

**Lundi** Statistiques descriptives, variables aléatoires et distributions

**Mardi** Estimation et intervalle de confiance

**Mercredi** Tests d'hypothèses (sous R)

## Cours (avec moi...)

1. Introduction
2. Individus, échantillon et statistiques descriptives
3. Introduction à R

*... et travaux dirigés!*

# Agenda (théorique) de la semaine

**Lundi** Statistiques descriptives, variables aléatoires et distributions

**Mardi** Estimation et intervalle de confiance

**Mercredi** Tests d'hypothèses (sous R)

## Cours (avec moi...)

1. Introduction
2. Individus, échantillon et statistiques descriptives
3. Introduction à R

... *et travaux dirigés!*

# Première partie I

## Introduction à la statistique

Concepts fondamentaux

Variables, distributions

Modes d'étude d'une population

Objectifs d'une étude statistique

Les données

### Concepts fondamentaux

Variables, distributions

Modes d'étude d'une population

Objectifs d'une étude statistique

Les données



# Qu'est ce que la statistique ?

## Statistique

Activité qui consiste dans le recueil, le traitement et l'interprétation de données d'observation.

## Population

- ▶ population, ensemble d'entités objet de l'investigation statistique, et non telle ou telle entité particulière.
- ▶ individus, définis comme les éléments d'une certaine population.

- ▶ Dans certain cas la population de référence est finie et ses éléments peuvent être explicitement dénombrés
- ▶ la notion de population revêt parfois une signification plus abstraite (exemple population de malades)
- ▶ parfois la notion de population s'identifie avec celle de procédure de génération de données (données d'expression)

Concepts fondamentaux

Variables, distributions

Modes d'étude d'une population

Objectifs d'une étude statistique

Les données

Chaque individu est décrit par un ensemble de variables :

- ▶ qualitative (sexe, nationalité, état matrimonial, ...) : les valeurs prises par le caractère sont les modalités
  - ▶ ordinale (notion d'ordre) : modalités intrinsèquement ordonnées
  - ▶ nominale : pas de structure d'ordre : par exemple le sexe.
- ▶ quantitative (taille, poids ...)
  - ▶ discrète
  - ▶ continue

La distribution d'un caractère  $X$  dans une population  $\mathcal{P}$  peut être décrite par la fonction de répartition

$$F_X : \mathbb{R} \rightarrow [0, 1]$$

qui à tout réel  $x$  associe la proportion  $P$  d'individus pour lesquels on a

$$X \leq x$$

Concepts fondamentaux

Variables, distributions

Modes d'étude d'une population

Objectifs d'une étude statistique

Les données

## Étude exhaustive

Dite par recensement.

L'étude d'une population de grande taille est souvent difficile voire impossible

## Échantillon

Le processus de sélection d'un échantillon est l'échantillonnage. Seule solution dans le cas d'une population infinie.

## Inférence statistique

Processus visant à

- ▶ déduire de conclusions générales relative à la population totale
- ▶ à partir de la connaissance particulière relative à un nombre de cas particulier

Concepts fondamentaux

Variables, distributions

Modes d'étude d'une population

Objectifs d'une étude statistique

Les données



1. synthétiser, résumer, structurer l'information :  
Statistique Descriptive ou Exploratoire
2. formuler ou valider des hypothèses relatives à la population totale :  
Statistique inférentielle

Concepts fondamentaux

Variables, distributions

Modes d'étude d'une population

Objectifs d'une étude statistique

Les données

# Le tableau de données

- ▶  $n$  individus mesurés par  $p$  variables

- ▶ Tableau  $X = (x_i^j) = \begin{pmatrix} x_1^1 & x_1^j & x_1^p \\ x_i^1 & x_i^j & x_i^p \\ \vdots & \vdots & \vdots \\ x_n^1 & x_n^j & x_n^p \end{pmatrix}$

- ▶ Chaque variable est représentée par le vecteur  $\mathbf{x}^j = (x_1^j, \dots, x_n^j)'$
- ▶ Chaque individu est représenté par le vecteur  $\mathbf{x}_i = (x_i^1, \dots, x_i^p)'$
- ▶  $X$  : réalisation d'un échantillon de taille  $n$  du vecteur aléatoire de dimension  $p$

$$\mathbf{X} = (X^1, \dots, X^p)'$$

# Deuxième partie II

## Introduction à R

Avant de démarrer

Installation et premiers contacts

Une session exemple

Avant de démarrer

Installation et premiers contacts

Une session exemple

# Qu'est-ce que R ?

En deux mots,

*R est un logiciel de développement scientifique spécialisé dans le calcul et l'**analyse statistique**.*

R est aussi

- ▶ un langage,
- ▶ un environnement,
- ▶ un projet open source (projet GNU),
- ▶ un logiciel multi-plateforme (Linux, Mac, Windows),

# Qu'est-ce que R ?

En deux mots,

*R est un logiciel de développement scientifique spécialisé dans le calcul et l'**analyse statistique**.*

R est aussi

- ▶ un langage,
- ▶ un environnement,
- ▶ un projet open source (projet GNU),
- ▶ un logiciel multi-plateforme (Linux, Mac, Windows),
- ▶ la 18<sup>e</sup> lettre de l'alphabet ☺.



# Qu'est-ce que R ?

En deux mots,

*R est un logiciel de développement scientifique spécialisé dans le calcul et l'**analyse statistique**.*

R est aussi

- ▶ un langage,
- ▶ un environnement,
- ▶ un projet open source (projet GNU),
- ▶ un logiciel multi-plateforme (Linux, Mac, Windows),
- ▶ la 18<sup>e</sup> lettre de l'alphabet ☹.

1. Gestionnaire de données
  - ▶ Lecture, manipulation, stockage.
2. Algèbre linéaire
  - ▶ Opérations classiques sur vecteurs, tableaux et matrices
3. Statistiques et analyse de données
  - ▶ Dispose d'un *grand* nombre de méthodes d'analyse de données (des plus anciennes et aux plus récentes)
4. Moteur de sorties graphiques
  - ▶ Sorties écran ou fichier
5. Système de modules
  - ▶ Alimenté par la communauté (+ de 2000 extensions !)
6. Interface facile avec C/C++, Fortran,...

1. Gestionnaire de **données**
  - ▶ Lecture, manipulation, stockage.
2. Algèbre linéaire
  - ▶ Opérations classiques sur vecteurs, tableaux et matrices
3. **Statistiques et analyse de données**
  - ▶ Dispose d'un *grand* nombre de méthodes d'analyse de données (des plus anciennes et aux plus récentes)
4. Moteur de **sorties graphiques**
  - ▶ Sorties écran ou fichier
5. Système de modules
  - ▶ Alimenté par la communauté (+ de 2000 extensions !)
6. Interface facile avec C/C++, Fortran, ...

## Approche chronologique

- 1970s développement de S au Bell labs.
- 1980s développement de S-PLUS au AT&T. Lab
- 1993 développement de R sur le modèle de S par Robert Gentleman et Ross Ihaka au département de statistique de l'université d'Auckland.
- 1995 dépôts des codes sources sous licence GNU/GPL
- 1997 élargissement du groupe
- 2002 la fondation R dépose ses statuts sous la présidence de Gentleman et Ihaka

## Développement entièrement bénévole

- ▶ « R development core team » (12aine de personnes)
- ▶ Participation de *nombreux* chercheurs (2000 packages)

## Approche chronologique

- 1970s développement de S au Bell labs.
- 1980s développement de S-PLUS au AT&T. Lab
- 1993 développement de R sur le modèle de S par Robert Gentleman et Ross Ihaka au département de statistique de l'université d'Auckland.
- 1995 dépôts des codes sources sous licence GNU/GPL
- 1997 élargissement du groupe
- 2002 la fondation R dépose ses statuts sous la présidence de Gentleman et Ihaka

## Développement entièrement bénévole

- ▶ « R development core team » (12aine de personnes)
- ▶ Participation de *nombreux* chercheurs (2000 packages)

1. La page web de la **fondation R**
  - ▶ les statuts, des liens, des références.
  - ▶ <http://www.r-project.org/>
2. La page web du **CRAN** (Comprehensive R Arxiv Network)
  - ▶ binaires d'installation, packages, documentations, ...
  - ▶ <http://cran.r-project.org/>
3. La **conférence** des utilisateurs de R :
  - ▶ annuelle, prochaine édition à Gaithersburg
  - ▶ <http://user2010.org/>
4. *The R journal* propose des articles sur
  - ▶ de nouvelles extensions, des applications, des actualités.
  - ▶ <http://journal.r-project.org/>

1. La page web de la **fondation** R
  - ▶ les statuts, des liens, des références.
  - ▶ <http://www.r-project.org/>
  
2. La page web du **CRAN** (Comprehensive R Arxiv Network)
  - ▶ binaires d'installation, packages, documentations, ...
  - ▶ <http://cran.r-project.org/>
  
3. La **conférence** des utilisateurs de R :
  - ▶ annuelle, prochaine édition à Gaithersburg
  - ▶ <http://user2010.org/>
  
4. *The R journal* propose des articles sur
  - ▶ de nouvelles extensions, des applications, des actualités.
  - ▶ <http://journal.r-project.org/>

1. La page web de la **fondation R**
  - ▶ les statuts, des liens, des références.
  - ▶ <http://www.r-project.org/>
2. La page web du **CRAN** (Comprehensive R Arxiv Network)
  - ▶ binaires d'installation, packages, documentations, ...
  - ▶ <http://cran.r-project.org/>
3. La **conférence** des utilisateurs de R : *useR!*
  - ▶ annuelle, prochaine édition à Gaithersburg
  - ▶ <http://user2010.org/>
4. *The R Journal* propose des articles sur
  - ▶ de nouvelles extensions, des applications, des actualités.
  - ▶ <http://journal.r-project.org/>



1. La page web de la **fondation R**
  - ▶ les statuts, des liens, des références.
  - ▶ <http://www.r-project.org/>
2. La page web du **CRAN** (Comprehensive R Arxiv Network)
  - ▶ binaires d'installation, packages, documentations, ...
  - ▶ <http://cran.r-project.org/>
3. La **conférence** des utilisateurs de R : *useR!*
  - ▶ annuelle, prochaine édition à Gaithersburg
  - ▶ <http://user2010.org/>
4. *The R journal* propose des articles sur
  - ▶ de nouvelles extensions, des applications, des actualités.
  - ▶ <http://journal.r-project.org/>

## Plus ☺

1. Libre et gratuit,
2. Richesse des modules (en statistique),
3. Rapidité d'exécution,
4. Développement rapide (langage de scripts),
5. Syntaxe intuitive et compact,
6. Nombreuses possibilités graphiques.

## Moins ☹

1. Aide intégrée succincte,
2. Debugger un peu sec,
3. Code parfois illisible (compacité),
4. Personnalisation des graphiques un peu lourde.

## Plus 😊

1. Libre et gratuit,
2. Richesse des modules (en statistique),
3. Rapidité d'exécution,
4. Développement rapide (langage de scripts),
5. Syntaxe intuitive et compact,
6. Nombreuses possibilités graphiques.

## Moins ☹️

1. Aide intégrée succincte,
2. Debugger un peu sec,
3. Code parfois illisible (compacité),
4. Personnalisation des graphiques un peu lourde.

Les logiciels de développement scientifique sont spécialisés en

## 1. algèbre linéaire

- ▶ Matlab (Mathworks), la référence,
- ▶ Scilab (INRIA), l'alternative libre,
- ▶ Octave (GNU), l'alternative open source ☺,

## 2. statistiques

- ▶ SAS (SAS Inc.), la référence,
- ▶ S-PLUS (TIBCO), le concurrent,
- ▶ R (GNU), l'alternative open source ☺,

## 3. calcul symbolique

- ▶ Mathematica (Wolfram), la référence,
- ▶ Maple (Maplesoft), la référence aussi,
- ▶ Maxima (GNU), l'alternative open source ☺.

Les logiciels de développement scientifique sont spécialisés en

## 1. algèbre linéaire

- ▶ Matlab (Mathworks), la référence,
- ▶ Scilab (INRIA), l'alternative libre,
- ▶ Octave (GNU), l'alternative open source ☺,

## 2. statistiques

- ▶ SAS (SAS Inc.), la référence,
- ▶ S-PLUS (TIBCO), le concurrent,
- ▶ R (GNU), l'alternative open source ☺,

## 3. calcul symbolique

- ▶ Mathematica (Wolfram), la référence,
- ▶ Maple (Maplesoft), la référence aussi,
- ▶ Maxima (GNU), l'alternative open source ☺.

Les logiciels de développement scientifique sont spécialisés en

## 1. algèbre linéaire

- ▶ Matlab (Mathworks), la référence,
- ▶ Scilab (INRIA), l'alternative libre,
- ▶ Octave (GNU), l'alternative open source ☺,

## 2. statistiques

- ▶ SAS (SAS Inc.), la référence,
- ▶ S-PLUS (TIBCO), le concurrent,
- ▶ R (GNU), l'alternative open source ☺,

## 3. calcul symbolique

- ▶ Mathematica (Wolfram), la référence,
- ▶ Maple (Maplesoft), la référence aussi,
- ▶ Maxima (GNU), l'alternative open source ☺.

## ► Obtenir de l'aide

<code>help -i</code>	<code>help.start ()</code>
<code>help</code>	<code>help(help)</code>
<code>help sort</code>	<code>help(sort) _or_ ?sort</code>

## ► Séquence de vecteurs

<code>1:10</code>	<code>1:10 _or_ seq(10)</code>
<code>1:3:10</code>	<code>seq(1,10,by=3)</code>
<code>10:-1:1</code>	<code>10:1</code>
<code>linspace(1,10,7)</code>	<code>seq(1,10,length=7)</code>

## ► Manipulation de vecteurs

<code>a=[2 7 8 5]</code>	<code>a &lt;- c(2,7,8,5)</code>
<code>a=a[3:4]</code>	<code>a &lt;- a[c(3,4)]</code>
<code>adash=[2 3 4 5]'</code>	<code>adash &lt;- t(c(2,3,4,5))</code>

Avant de démarrer

Installation et premiers contacts

Une session exemple



# Installation

Rendez-vous sur la page du CRAN <http://cran.r-project.org/>

## Macintosh

Télécharger `R-2.10.1.pkg`, cliquer.

## Windows

Télécharger `R-2.10.1-win32.exe`, cliquer (prier).

## Linux

Systèmes supportant `apt` (Debian, Ubuntu, ...)

```
$ sudo apt-get update
```

```
$ sudo apt-get install r-base
```

## Premiers pas

```
$ R
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
[...]
Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.
> 1+1
[1] 2
```

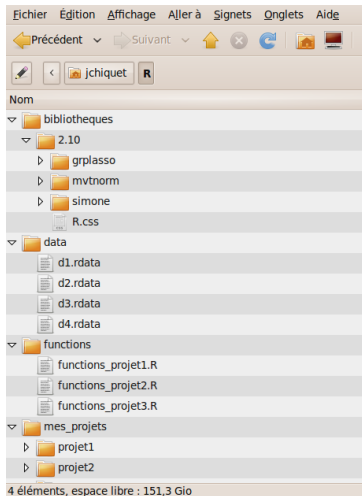
## Sortez moi de là !

```
> q()
Save workspace image? [y/n/c]:y
```

↪ Sauve l'environnement et le réouvre la prochaine fois

# Organiser un projet R

## À calquer lors des travaux dirigés



- ▶ Dans un répertoire R, placer
  - ▶ un répertoire data
  - ▶ un répertoire mes\_projets
  - ▶ un répertoire fonctions
- ▶ Créer un répertoire par projet
  - ▶ sauvegarde des données  
`save.image(file = "f.RData")`
  - ▶ sauvegarde des instructions  
`savehistory(file = "f.Rhistory")`
- ▶ bibliothèques contient les extensions installées.

FIGURE : Arborescence type

# Environnement de travail sous Linux

Un bureau de développement avec R

```
File Edit Options Buffers Tools Imenu-S ESS Help
rm(list=ls())
library(mvtnorm)
source("functions.R")
source("functions_group_l1.R")

set.seed(1002)

## données simulés (settings de Yuan et Lin - papier
er de 2006)
n <- 100
Sigma <- matrix(c(1,0.5,0.5,1),2,2)
X <- rmvnorm(n, Sigma)
y <- X[,1]^3 + X[,1]^2 - 2 * X[,1] + (1/3)*X[,2]^3
  - 0*X[,2]^2 + (2/3)*X[,2] + rnorm(n,0,3)
U:~ check_CoopLasso.R Top (11.0) SVN-385 (ESS[S] [none])--15:50 0.47--
```

```
Fichier Édition Affichage Terminal Onglets Aide
Terminal Terminal Terminal
15:03 jchiquet@term14 ~/svn/notiid/branches/regressionCoop/R R >
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

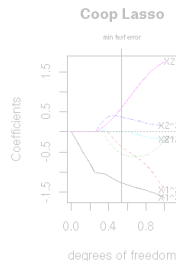
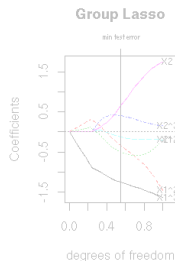
R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> source("check_CoopLasso.R")
```

1. un éditeur de texte
2. un terminal avec R
3. des sorties graphiques



# Environnement de travail sous Linux

Un bureau de développement avec R

```
File Edit Options Buffers Tools Imenu-S ESS Help
rm(list=ls())
library(mvtnorm)
source("fonctions.R")
source("fonctions_group_ll.R")

set.seed(1002)

## données simulés (settings de Yuan et Lin - papier
er de 2006)
n <- 100
Sigma <- matrix(c(1,0.5,0.5,1),2,2)
X <- rmvnorm(n, Sigma)
y <- X[,1]^3 + X[,1]^2 - 2 * X[,1] + (1/3)*X[,2]^3
  - 0*X[,2]^2 + (2/3)*X[,2] + rnorm(n,0,3)
U:~ check_CoopLasso.R Top (11.0) SVN-385 (ESS[S][none])--15:50 047~
```

```
Fichier Édition Affichage Terminal Onglets Aide
Terminal Terminal Terminal
15:03 jchiquet@term14 ~/svn/notiid/branches/regressionCoop/R R >
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

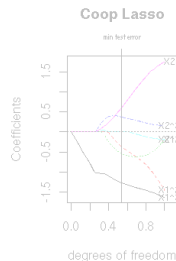
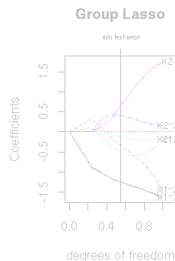
R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> source("check_CoopLasso.R")
```

1. un éditeur de texte
2. un terminal avec R
3. des sorties graphiques



# Environnement de travail sous Linux

Un bureau de développement avec R

```
File Edit Options Buffers Tools Imenu-S ESS Help
[Icons]
rm(list=ls())
library(mvtnorm)
source("functions.R")
source("functions_group_ll.R")

set.seed(1002)

## données simulés (settings de Yuan et Lin - papier de 2006)
n <- 100
Sigma <- matrix(c(1,0.5,0.5,1),2,2)
X <- rmvnorm(n, Sigma)
y <- X[,1]^3 + X[,1]^2 - 2 * X[,1] + (1/3)*X[,2]^3 - 0*X[,2]^2 + (2/3)*X[,2] + rnorm(n,0,3)
#U:-- check_CoopLasso.R Top (11.0) SVN-385 (ESS[S][none])--15:50 0.47--
```

```
Fichier Édition Affichage Terminal Onglets Aide
Terminal Terminal Terminal
15:03 jchiquet@term14 ~/svn/notiid/branches/regressionCoop/R R >
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

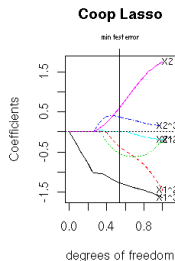
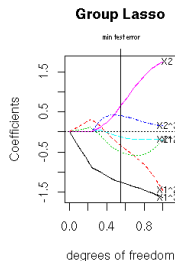
R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> source("check_CoopLasso.R")
```

1. un éditeur de texte
2. un terminal avec R
3. des sorties graphiques



## Depuis R

- ▶ `help(str)` : lance l'aide associée à la commande `str`,
- ▶ `help.search("factorial")` : cherche les commandes contenant le mot-clé `factorial`,
- ▶ `help.start()` : lance l'aide HTML.

## Sur le Web

- ▶ **Le site du CRAN** : beaucoup (trop ?) de guides d'utilisations sont répertoriés (y compris ceux en français),
- ▶ le site du module en propose une sélection.

## À tout moment

- ▶ la liste des commandes usuelles,
- ▶ le prof (pas infaillible mais rapide d'accès).

Avant de démarrer

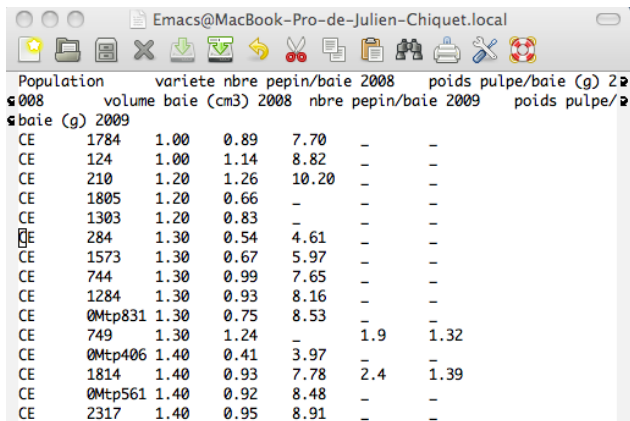
Installation et premiers contacts

Une session exemple



# Analyse élémentaire d'un jeu de données

Quelle tête ont les données ? On ouvre avec Emacs :



The image shows a screenshot of the Emacs editor window titled "Emacs@MacBook-Pro-de-Julien-Chiquet.local". The window displays a table of grape data with the following columns: Population, variete, nbre pepin/baie 2008, poids pulpe/baie (g) 2008, volume baie (cm3) 2008, nbre pepin/baie 2009, and poids pulpe/baie (g) 2009. The data rows are as follows:

Population	variete	nbre pepin/baie 2008	poids pulpe/baie (g) 2008	volume baie (cm3) 2008	nbre pepin/baie 2009	poids pulpe/baie (g) 2009
1784	CE	1.00	0.89	7.70	-	-
124	CE	1.00	1.14	8.82	-	-
210	CE	1.20	1.26	10.20	-	-
1805	CE	1.20	0.66	-	-	-
1303	CE	1.20	0.83	-	-	-
284	CE	1.30	0.54	4.61	-	-
1573	CE	1.30	0.67	5.97	-	-
744	CE	1.30	0.99	7.65	-	-
1284	CE	1.30	0.93	8.16	-	-
0Mtp831	CE	1.30	0.75	8.53	-	-
749	CE	1.30	1.24	-	1.9	1.32
0Mtp406	CE	1.40	0.41	3.97	-	-
1814	CE	1.40	0.93	7.78	2.4	1.39
0Mtp561	CE	1.40	0.92	8.48	-	-
2317	CE	1.40	0.95	8.91	-	-

FIGURE : données baies de vignes 2008/2009

Je remplace tous les \_ par du vide (R le comprendra mieux) !

Les commandes `getwd()` et `setwd()` gèrent le répertoire de travail :

```
> setwd("~/SVN/gao/BaseStatBio/0_IntroductionBaseStat/")
> getwd()

[1] "/Users/tom/SVN/gao/BaseStatBio/0_IntroductionBaseStat"
```

Les données possèdent un entête et sont délimitées par des tabulations :

```
> donnees <- read.delim("mesures_baie_raisin_2008-2009.txt")
```

Qu'est-ce qui se trouve dorénavant dans mon itinéraire de recherche ?

```
> ls()

[1] "donnees"

> objects()

[1] "donnees"
```

## Quelle tête ont mes données ?

```
> head(donnees)
```

```
  Population variete  nbre.pepin.baie.2008 poids.pulpe.baie..g..2008
1          CE      1784                1.0                0.89
2          CE       124                1.0                1.14
3          CE       210                1.2                1.26
4          CE      1805                1.2                0.66
5          CE      1303                1.2                0.83
6          CE       284                1.3                0.54
 volume.baie..cm3..2008  nbre.pepin.baie.2009 poids.pulpe.baie..g..2009
1                    7.70                    NA                    NA
2                    8.82                    NA                    NA
3                   10.20                    NA                    NA
4                    NA                    NA                    NA
5                    NA                    NA                    NA
6                    4.61                    NA                    NA
```

## Je les mets dans mon itinéraire de recherche

```
> attach(donnees)
```

Les objets suivants sont masqués from donnees (position 3):

```
nbre.pepin.baie.2008, nbre.pepin.baie.2009,  
poids.pulpe.baie..g..2008, poids.pulpe.baie..g..2009, Population,  
variete, volume.baie..cm3..2008
```

Les objets suivants sont masqués from donnees (position 4):

```
nbre.pepin.baie.2008, nbre.pepin.baie.2009,  
poids.pulpe.baie..g..2008, poids.pulpe.baie..g..2009, Population,  
variete, volume.baie..cm3..2008
```

## Et les attributs ?

```
> str(donnees)
```

```
'data.frame':      245 obs. of  7 variables:
 $ Population      : Factor w/  3 levels "CE","CO","TE": 1 1 1 1 1 1 1 1 1 1 ...
 $ variete         : Factor w/ 245 levels "OMtp1004","OMtp1005",...: 113 66 ...
 $ nbre.pepin.baie.2008 : num  1 1 1.2 1.2 1.2 1.3 1.3 1.3 1.3 1.3 ...
 $ poids.pulpe.baie..g..2008: num  0.89 1.14 1.26 0.66 0.83 0.54 0.67 0.99 0.93 0.7 ...
 $ volume.baie..cm3..2008  : num  7.7 8.82 10.2 NA NA 4.61 5.97 7.65 8.16 8.53 ...
 $ nbre.pepin.baie.2009   : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ poids.pulpe.baie..g..2009: num  NA NA NA NA NA NA NA NA NA NA NA ...
```

# Troisième partie III

## Structures de données

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données



### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

## Propriétés

- ▶ objet le plus **élémentaire** sous R,
- ▶ collection d'entités **de même nature**,
- ▶ **mode** (ou type) défini par la nature des entités qui le composent.

## Les modes possibles

1. numérique (`numeric`),
2. caractère (`character`),
3. logique (`boolean`).

## Propriétés

- ▶ objet le plus **élémentaire** sous  $\mathbb{R}$ ,
- ▶ collection d'entités **de même nature**,
- ▶ **mode** (ou type) défini par la nature des entités qui le composent.

## Les modes possibles

1. numérique (`numeric`),
2. caractère (`character`),
3. logique (`boolean`).

## Propriétés

- ▶ objet le plus **élémentaire** sous  $\mathbb{R}$ ,
- ▶ collection d'entités **de même nature**,
- ▶ **mode** (ou type) défini par la nature des entités qui le composent.

## Les modes possibles

1. numérique (`numeric`),
2. caractère (`character`),
3. logique (`boolean`).

## 1. Numérique

```
> x0 <- 0
> x1 <- c(-1,23,98.7)
> mode(x0)

[1] "numeric"
```

## 2. Caractère

```
> y0 <- "bonjour"
> y1 <- c("Pomme","Flore","Alexandre")
> mode(y1)

[1] "character"
```

## 3. Logique

```
> z0 <- TRUE
> z1 <- c(FALSE,TRUE,FALSE,TRUE,TRUE)
> z2 <- c(T,F,F)
> mode(z2)

[1] "logical"
```

## Définition (affectation)

C'est l'opération qui consiste à *attribuer une valeur* à une variable.

En R, plusieurs choix sont possibles :

- ▶ l'opérateur usuel est '`<-`' (signe inférieur suivi du signe moins)

```
> jo <- "l'indien"  
> jo  
[1] "l'indien"
```

- ▶ l'opérateur '`=`' peut être utilisé la plupart du temps

```
> nb.max.d.annees.pour.faire.une.these = 3  
> nb.max.d.annees.pour.faire.une.these  
[1] 3
```

- ▶ la commande `assign` permet cette opération (d'où l'anglicisme *assignation*)

```
> assign("x", c(8,9,-pi,sqrt(2)))  
> x  
[1] 8.000000 9.000000 -3.141593 1.414214
```

## Variabes réservées par R

- ▶ NA est le code R pour les valeurs manquantes (absentes des données),
- ▶ NaN est le code de R pour signifier un résultat numérique aberrant ,
- ▶ Inf et -Inf sont les valeurs réservées pour plus et moins  $\infty$ ,
- ▶ NULL est l'objet nul.

```
> c(4,2,NA,5)
[1] 4 2 NA 5
> 0/0
[1] NaN
> 1/0
[1] Inf
> names(1)
NULL
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires**

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données



# Opérations arithmétiques

s'effectuent terme-à-terme

Soient  $x, y$  tels que

>  $x <- c(1, 2, -3, -4)$

>  $y <- c(-5, -6, 9, 0)$

'+' addition des éléments de deux vecteurs

>  $x+y$

[1] -4 -4 6 -4

'-' soustraction des éléments de deux vecteurs

>  $x-y$

[1] 6 8 -12 -4

'\*' multiplication des éléments de deux vecteurs

>  $x*y$

[1] -5 -12 -27 0

'/' division des éléments de deux vecteurs

>  $x/y$

[1] -0.2000000 -0.3333333 -0.3333333 -Inf

# Le « recyclage » des éléments du vecteur

Lors d'une opération entre vecteurs, les vecteurs trop courts sont ajustés pour atteindre la taille du plus grand vecteur en recyclant les données.

## Exemple

```
> x <- c(10,100,1000)
> y <- c(1,2)
> 2*x + y - 1
[1] 20 201 2000
```

↪ souvent pratique mais **attention aux effets de bords!**

## Fonctions numériques élémentaires

`floor`, `ceiling`, `round`.

```
> floor(2/3)
```

```
[1] 0
```

```
> ceiling(2/3)
```

```
[1] 1
```

```
> round(2/3,3)
```

```
[1] 0.667
```

## Fonctions arithmétiques élémentaires

`^`, `%%`, `%/%`, `abs`, `log`, `exp`, `log10`, `sqrt`, `cos`, `tan`, `sin`... s'appliquent toutes **terme-à-terme**.

```
> log10(c(10,100,1000))
```

```
[1] 1 2 3
```

```
> cos(c(pi/2,pi))^2 + sin(c(pi/2,pi))^2
```

```
[1] 1 1
```

## Fonctions caractérisant un vecteur

prod, sum, max, min, range, which.min, which.max, length

```
> x <- c(-8,1.5,3)
```

```
> prod(x)
```

```
[1] -36
```

```
> sum(x)
```

```
[1] -3.5
```

```
> length(x)
```

```
[1] 3
```

```
> max(x)
```

```
[1] 3
```

```
> min(x)
```

```
[1] -8
```

```
> range(x)
```

```
[1] -8 3
```

```
> which.max(x)
```

```
[1] 3
```

```
> which.min(x)
```

```
[1] 1
```

Pour le minimum / maximum terme-à-terme : `pmin`, `pmax`.

## Fonctions appliquées le long du vecteur

`cumsum`, `cumprod`, `cummin`, `cummax`

```
> x <- c(-2, 1, -3, 2)
```

```
> cumprod(x)
```

```
[1] -2 -2 6 12
```

```
> cumsum(x)
```

```
[1] -2 -1 -4 -2
```

```
> cummax(x)
```

```
[1] -2 1 1 2
```

```
> cummin(x)
```

```
[1] -2 -2 -3 -3
```

## Fonctionnent pour tous les modes

`unique`, `intersect`, `union`, `setdiff`, `setequal`, `is.element`

```
> unique(c("banane", "citron", "banane"))
```

```
[1] "banane" "citron"
```

```
> intersect(c("banane", "citron"), c("orange", "banane"))
```

```
[1] "banane"
```

```
> union(c("banane", "citron"), c("orange", "banane"))
```

```
[1] "banane" "citron" "orange"
```

```
> setequal(c("banane", "citron"), c("orange", "banane"))
```

```
[1] FALSE
```

```
> is.element(1, sample(c(1,2,3),2))
```

```
[1] FALSE
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs**

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données



## from:to

Génère une séquence par pas de un depuis le nombre `from` jusqu'à `to` (si possible).

```
> -5:5
```

```
[1] -5 -4 -3 -2 -1  0  1  2  3  4  5
```

```
> 5:-5
```

```
[1]  5  4  3  2  1  0 -1 -2 -3 -4 -5
```

```
> pi:6
```

```
[1] 3.141593 4.141593 5.141593
```

```
> 1:6/2
```

```
[1] 0.5 1.0 1.5 2.0 2.5 3.0
```

```
> 1:(6/2)
```

```
[1] 1 2 3
```

## Plusieurs schémas possibles

- ▶ `seq(from,to)`
- ▶ `seq(from,to,by=)`
- ▶ `seq(from,to,length.out=)`

```
> seq(1,10)
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
> seq(2,10,by=2)
```

```
[1] 2 4 6 8 10
```

```
> seq(2,10,length.out=6)
```

```
[1] 2.0 3.6 5.2 6.8 8.4 10.0
```

## Fonctionne pour tous les modes

- ▶ `rep(x, times)`, où `times` peut être un vecteur,
- ▶ `rep(x, each)`.

```
> rep(1,3)
```

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

```
> rep("Mercy",3)
```

```
[1] "Mercy" "Mercy" "Mercy"
```

```
> rep(c("A", "B", "C"), c(3,2,4))
```

```
[1] "A" "A" "A" "B" "B" "C" "C" "C" "C"
```

```
> rep(c(TRUE,FALSE), each=2)
```

```
[1] TRUE TRUE FALSE FALSE
```

## Obtenus par conditions avec

- ▶ les opérateurs logiques '`<`', '`<=`', '`>`', '`>=`', '`==`' '`!=`'
- ▶ le ET, le OU, NON, OU exclusif : '`&`' (intersection), '`|`' (union), '`!`' (négation), `xor`.

```
> note1 <- c(8,9,14,3,17.5,11)
> note2 <- c("C","B","A","B","E","B")
> admis <- (note1 >= 10) & (note2 == "A" | note2 == "B")
> mention <- (note1 >= 15) & (note2 == "A")
> admis
[1] FALSE FALSE TRUE FALSE FALSE TRUE
> sum(admis)
[1] 2
> sum(mention)
[1] 0
```

## Avec 'c()'

L'opérateur 'c()' peut s'appliquer à n'importe quoi pourvu que l'on concatène des vecteurs de même type.

```
> c( c(1,2), c(3,4))
```

```
[1] 1 2 3 4
```

```
> round(c(seq(-pi,pi,len=4),rep(c(1:3),each=2),0),2)
```

```
[1] -3.14 -1.05  1.05  3.14  1.00  1.00  2.00  2.00  3.00  3.00  0.00
```

## Remarque

*Dans le second exemple, les entiers composants c(1:3) ont été forcés au typage flottant.*

## Avec `paste`

Concaténation de chaînes de caractères. Convertit en caractères les éléments passés en argument avant toute opération.

```
> paste("R", "c'est", "bien")
```

```
[1] "R c'est bien"
```

```
> paste(2:4, "ieme")
```

```
[1] "2 ieme" "3 ieme" "4 ieme"
```

```
> paste("A", 1:5, sep="")
```

```
[1] "A1" "A2" "A3" "A4" "A5"
```

```
> paste("A", 1:5, sep="", collapse="")
```

```
[1] "A1A2A3A4A5"
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs**

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

## Principe

- ▶ Permet la **sélection d'un sous-ensemble** du vecteur  $x$ .
- ▶ Le sous-ensemble est spécifié **entre crochets**  $x[\text{subset}]$ .

L'objet `subset` peut prendre 4 types différents :

1. **un vecteur logique**, qui doit être de la même taille que le vecteur  $x$  ;
2. **un vecteur numérique aux composantes positives**, qui spécifie les valeurs à inclure ;
3. **un vecteur numérique aux composantes négatives**, qui spécifie les valeurs à exclure ;
4. **un vecteur de chaînes de caractères**, qui spécifie les noms des éléments de  $x$  à conserver.



## Principe

- ▶ Permet la sélection d'un sous-ensemble du vecteur  $x$ .
- ▶ Le sous-ensemble est spécifié **entre crochets**  $x[\text{subset}]$ .

L'objet `subset` peut prendre 4 types différents :

1. **un vecteur logique**, qui doit être de la même taille que le vecteur  $x$  ;
2. **un vecteur numérique aux composantes positives**, qui spécifie les valeurs à inclure ;
3. **un vecteur numérique aux composantes négatives**, qui spécifie les valeurs à exclure ;
4. **un vecteur de chaînes de caractères**, qui spécifie les noms des éléments de  $x$  à conserver.

## Principe

- ▶ Permet la sélection d'un sous-ensemble du vecteur  $x$ .
- ▶ Le sous-ensemble est spécifié **entre crochets**  $x[\text{subset}]$ .

L'objet `subset` peut prendre 4 types différents :

1. **un vecteur logique**, qui doit être de la même taille que le vecteur  $x$  ;
2. **un vecteur numérique aux composantes positives**, qui spécifie les valeurs à inclure ;
3. **un vecteur numérique aux composantes négatives**, qui spécifie les valeurs à exclure ;
4. **un vecteur de chaînes de caractères**, qui spécifie les noms des éléments de  $x$  à conserver.

## Principe

- ▶ Permet la sélection d'un sous-ensemble du vecteur  $x$ .
- ▶ Le sous-ensemble est spécifié **entre crochets**  $x[\text{subset}]$ .

L'objet `subset` peut prendre 4 types différents :

1. **un vecteur logique**, qui doit être de la même taille que le vecteur  $x$  ;
2. **un vecteur numérique aux composantes positives**, qui spécifie les valeurs à inclure ;
3. **un vecteur numérique aux composantes négatives**, qui spécifie les valeurs à exclure ;
4. **un vecteur de chaînes de caractères**, qui spécifie les noms des éléments de  $x$  à conserver.

## Principe

- ▶ Permet la sélection d'un sous-ensemble du vecteur  $x$ .
- ▶ Le sous-ensemble est spécifié **entre crochets**  $x[\text{subset}]$ .

L'objet `subset` peut prendre 4 types différents :

1. **un vecteur logique**, qui doit être de la même taille que le vecteur  $x$  ;
2. **un vecteur numérique aux composantes positives**, qui spécifie les valeurs à inclure ;
3. **un vecteur numérique aux composantes négatives**, qui spécifie les valeurs à exclure ;
4. **un vecteur de chaînes de caractères**, qui spécifie les noms des éléments de  $x$  à conserver.

## Vecteurs logiques

```
> x <- c(3,6,-2,9,NA,sin(-pi/6))
```

```
> x[x > 0]
```

```
[1] 3 6 9 NA
```

```
> x[!is.na(x)]
```

```
[1] 3.0 6.0 -2.0 9.0 -0.5
```

```
> x[!is.na(x) & x>0]
```

```
[1] 3 6 9
```

```
> mean(x,na.rm=TRUE)
```

```
[1] 3.1
```

```
> x[x <= mean(x,na.rm=TRUE)]
```

```
[1] 3.0 -2.0 NA -0.5
```

## Vecteurs aux composantes positives ou négatives

```
> x
[1] 3.0 6.0 -2.0 9.0 NA -0.5

> x[2]
[1] 6

> x[1:5]
[1] 3 6 -2 9 NA

> x[-c(1,5)]
[1] 6.0 -2.0 9.0 -0.5

> x[-(1:5)]
[1] -0.5
```

## Vecteurs de chaînes de caractères

```
> names(x) <- c("var1", "var2", "var3", "var4", "var5", "var6")
```

```
> x
```

```
var1 var2 var3 var4 var5 var6  
 3.0  6.0 -2.0  9.0  NA -0.5
```

```
> x[c("var1", "var3")]
```

```
var1 var3  
  3   -2
```

## 1. Classer

- ▶ `sort` renvoie le vecteur classé par ordre croissant ou décroissant,
- ▶ `order` renvoie les indices d'ordre des éléments par ordre croissant ou décroissant,

## 2. Extraire

- ▶ `which` renvoie les indices de `x` vérifiant une condition ;

## 3. Échantillonner

- ▶ `sample` échantillonne aléatoirement dans un vecteur `x`, avec ou sans remise.



## 1. Classer

- ▶ `sort` renvoie le vecteur classé par ordre croissant ou décroissant,
- ▶ `order` renvoie les indices d'ordre des éléments par ordre croissant ou décroissant,

## 2. Extraire

- ▶ `which` renvoie les indices de `x` vérifiant une condition ;

## 3. Échantillonner

- ▶ `sample` échantillonne aléatoirement dans un vecteur `x`, avec ou sans remise.

## 1. Classer

- ▶ `sort` renvoie le vecteur classé par ordre croissant ou décroissant,
- ▶ `order` renvoie les indices d'ordre des éléments par ordre croissant ou décroissant,

## 2. Extraire

- ▶ `which` renvoie les indices de  $x$  vérifiant une condition ;

## 3. Échantillonner

- ▶ `sample` échantillonne aléatoirement dans un vecteur  $x$ , avec ou sans remise.

# Exemples

```
> x <- -5:5
> y <- sample(x)
> sort(y)

[1] -5 -4 -3 -2 -1  0  1  2  3  4  5

> order(y)

[1]  3  9  5  7  1  4 10  6  8  2 11

> y[order(y)]

[1] -5 -4 -3 -2 -1  0  1  2  3  4  5

> y[order(y,decreasing=TRUE)]

[1]  5  4  3  2  1  0 -1 -2 -3 -4 -5

> which(sample(x,4) > 0)

[1] 1
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

## Définition

*Un facteur est un vecteur de **variables catégorielles**. Les niveaux du facteur peuvent être ordonnés ou pas.*

## Utilisation

les facteurs s'utilisent pour **catégoriser les données** d'un vecteur (ce qui s'avère très utile pour la gestion des variables qualitatives).

↪ un facteur est souvent associé à d'autres vecteurs pour en définir une **partition**.

## Définition

*Un facteur est un vecteur de **variables catégorielles**. Les niveaux du facteur peuvent être ordonnés ou pas.*

## Utilisation

les facteurs s'utilisent pour **catégoriser les données** d'un vecteur (ce qui s'avère très utile pour la gestion des variables qualitatives).

↪ un facteur est souvent associé à d'autres vecteurs pour en définir une **partition**.

## Définition

*Un facteur est un vecteur de **variables catégorielles**. Les niveaux du facteur peuvent être ordonnés ou pas.*

## Utilisation

les facteurs s'utilisent pour **catégoriser les données** d'un vecteur (ce qui s'avère très utile pour la gestion des variables qualitatives).

↪ un facteur est souvent associé à d'autres vecteurs pour en définir une **partition**.

## Création : la fonction factor

```
> factor(sample(1:3,10,replace=TRUE))
```

```
[1] 3 3 1 2 2 2 2 3 2
```

```
Levels: 1 2 3
```

```
> factor(sample(1:3,10,replace=TRUE),levels=1:5)
```

```
[1] 3 3 1 3 2 1 3 2 1 2
```

```
Levels: 1 2 3 4 5
```

## Gestion : nlevels, levels, table

```
> x <- factor(sample(c("thésard", "CR", "MdC"),15,replace=TRUE))
```

```
> cat(nlevels(x), "niveaux:", levels(x))
```

```
3 niveaux: CR MdC thésard
```

```
> table(x)
```

```
x
```

```
CR      MdC thésard
```

```
5       4       6
```



# Un exemple de facteur associé à un vecteur

Un exemple de facteur associé à un vecteur

## Données

Chacun me donne son âge et son grade<sup>1</sup>

```
> age <- c(25,35,32,27,32,40,26,25,26,28,30,NA,36,30,30)
> grd <- c("thd", "CR", "MdC", "thd", "thd", "MdC", "MdC", "thd", "thd", "MdC", "CR")
```

Question : nombre d'individus par catégorie ?

```
> table(grd)
```

```
grd
CR MdC thd
 3   5   7
```

---

1. sauf un qui refuse :'(

## Utilisation

Applique une fonction sur un vecteur partitionné en groupes.

Question : âge moyen / écart-type par catégorie ?

```
> tapply(age,grd,mean,na.rm=TRUE)
```

```
      CR      MdC      thd  
33.66667 31.50000 27.85714
```

```
> tapply(age,grd,sd,na.rm=TRUE)
```

```
      CR      MdC      thd  
3.214550 6.191392 2.794553
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

## Définition (objet array)

*Un tableau est un vecteur muni d'un attribut dimension (*dim*), lui même défini par un vecteur. Il est défini par la commande `array(data, dim, dimnames=)`*

```
> array(1:8, c(2,2,2))
```

```
, , 1
```

	[,1]	[,2]
[1,]	1	3
[2,]	2	4

```
, , 2
```

	[,1]	[,2]
[1,]	5	7
[2,]	6	8

## Définition (objet `matrix`)

*Une matrice est un tableau à deux dimensions. Elle est définie par la commande*

```
matrix(data, nrow=, ncol=, byrow)
```

En conséquence

- ▶ Un objet `array` à deux dimensions est automatiquement converti en `matrix`
- ▶ Un vecteur auquel on ajoute un attribut `dimension` est automatiquement converti en `matrix`

```
> class(array(1:4, c(2,2)))
```

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

```
> x <- c(1,2,3,4)
```

```
> dim(x) <- c(2,2)
```

```
> class(x)
```

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

1. R range les éléments d'une matrice par défaut par **colonne**.

```
> matrix(1:6,nrow=2)
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

```
> matrix(1:6,nrow=2,byrow=TRUE)
```

```
      [,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6
```

2. Lors de la création d'une matrice, R **recycle** les éléments jusqu'à ce que les contraintes de dimension soient vérifiées.

```
> matrix(1:3,nrow=2,ncol=2)
```

```
      [,1] [,2]  
[1,]    1    3  
[2,]    2    1
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données



Étant donné qu'une matrice est un vecteur pourvu d'une dimension, on a la proposition suivante :

## Proposition

*La plupart des opérateurs vectorielles s'appliquent (arithmétiques/mathématiques, ensemblistes, d'indexation).*

```
> a <- matrix(sample(-4:4,9),3,3)
> cat(max(a),sum(a),prod(a))

4 0 0

> which(a > 0)

[1] 1 4 7 8

> cumsum(a[a > 0])

[1] 1 3 7 10

> order(a)

[1] 9 3 5 6 2 1 4 8 7

> round(exp(a),4)

      [,1]  [,2]  [,3]
[1,] 2.7183 7.3891 54.5982
[2,] 1.0000 0.1353 20.0855
[3,] 0.0498 0.3679 0.0183
```

## Opérateurs matriciels usuels

- ▶ `+`, `/`, `*`, `^` sont les opérateurs usuels terme-à-terme,
- ▶ `%*%` est le produit matriciel,
- ▶ `crossprod()` est le produit scalaire,
- ▶ `t()` transpose une matrice,
- ▶ `diag()` extrait / spécifie la diagonale.

```
> a <- matrix(sample(-4:4,9),3,3)
> b <- matrix(sample(a),3,3)
> diag(a)
[1] 0 3 -4
> diag(a) <- diag(b) <- 1
> diag(a)
[1] 1 1 1
> a + t(b) %*% b
      [,1] [,2] [,3]
[1,]  10  -5   1
[2,]  -2   3  -2
[3,]  -6   1  12
```

# Concaténation de matrices

Trois fonctions selon l'effet voulu :

1. `c()` concatène les éléments de plusieurs matrices en un vecteur,
2. `cbind()` empile **horizontalement** plusieurs matrices,
3. `rbind()` empile **verticalement** plusieurs matrices.

```
> a <- matrix(1,2,3)
> b <- matrix(2,2,3)
> c(a,b)

[1] 1 1 1 1 1 1 1 2 2 2 2 2 2
```

```
> cbind(a,b)

      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    2    2    2
[2,]    1    1    1    2    2    2
```

```
> rbind(a,b)

      [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    1    1
[3,]    2    2    2
[4,]    2    2    2
```

### Vecteurs

- Les modes ou typages

- Opérations élémentaires

- Génération de vecteurs

- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création

- Manipulation de matrices

- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

## Résolution de systèmes linéaires, inversion matricielle

La commande `solve` résout

$$\mathbf{Ax} = \mathbf{b},$$

```
> A <- matrix(c(4,2,8,-3),2,2)
> b <- c(2,3)
> solve(A,b)
```

```
[1] 1.0714286 -0.2857143
```

ou inverse une matrice :

```
> round(solve(A) %*% A,8)
```

```
      [,1] [,2]
[1,]    1    0
[2,]    0    1
```

R dispose des outils classiques d'algèbre linéaire

- ▶ `det` : calcule le **déterminant** d'une matrice ;
- ▶ `chol` : factorisation de **Cholesky** ( $A = C^T C$ , avec  $A$  symétrique,  $C$  triangulaire supérieure) ;
- ▶ `qr` : factorisation **QR** ( $A = QR$  avec  $Q$  orthogonale,  $R$  triangulaire supérieure) ;
- ▶ `eigen` : calcule valeurs propres et **vecteurs propres** d'une matrice ;
- ▶ `svd` : calcule la décomposition en **valeurs singulières**.
- ▶ ...

### Vecteurs

- Les modes ou typages
- Opérations élémentaires
- Génération de vecteurs
- Manipulation de vecteurs

### Facteurs

### Matrices (et tableaux)

- Définition, création
- Manipulation de matrices
- Opérateurs d'algèbre linéaire

### Listes et Tableaux de données

## Définition (objet list)

Une liste est une *collection d'objets hétérogènes*. Elle est définie par la commande `list(e11=, e12=, ...)`. Les éléments d'une liste peuvent posséder un nom.

```
> list(c(1,2,3),c("robert","johnson"),matrix(rnorm(4),2,2))
```

```
[[1]]  
[1] 1 2 3
```

```
[[2]]  
[1] "robert" "johnson"
```

```
[[3]]  
      [,1]      [,2]  
[1,] -0.4891467 -0.1518925  
[2,] -0.9995514  1.2755982
```

```
> list(numero = c(1,2,3), noms = c("robert","johnson"), mat = matrix(rnorm(4),2,2))
```

```
$numero  
[1] 1 2 3
```

```
$noms  
[1] "robert" "johnson"
```

```
$mat  
      [,1]      [,2]  
[1,] -0.4327840 -0.2820244  
[2,] -0.1164887 -0.3989929
```



## Deux situations

1. Les éléments de la liste **ne sont pas nommés** : on accède au  $i^{\text{e}}$  élément par indexation `nom_liste[[i]]` uniquement.
2. Les éléments de la liste **sont nommés** : on peut y accéder comme ci-dessus ou en utilisant le nom de l'élément `nom_liste$nom_elt`.

```
> maliste <- list(numero = c(1,2,3), noms = c("robert","johnson"), mat = matrix(rnorm(10), 2, 5))
> maliste$nom
```

```
[1] "robert" "johnson"
```

```
> maliste$nom[2]
```

```
[1] "johnson"
```

```
> maliste[[2]]
```

```
[1] "robert" "johnson"
```

```
> maliste[[2]][2]
```

```
[1] "johnson"
```

## Sélectionner des éléments

Fonctionne (presque) comme pour les vecteurs

```
> l1 <- list(1:2,c("a","c","g","t"))  
> l1[[-2]]  
  
[1] 1 2
```

## Commande lapply

Applique une fonction à chaque élément d'une liste

```
> lapply(maliste,length)  
  
$numero  
[1] 3  
  
$noms  
[1] 2  
  
$mat  
[1] 4
```

## Commande `c()`

Permet de concaténer deux listes.

```
> c(list(1:2,c("a","c","g","t")),list(rnorm(3),"yop"))
```

```
[[1]]
```

```
[1] 1 2
```

```
[[2]]
```

```
[1] "a" "c" "g" "t"
```

```
[[3]]
```

```
[1] -1.0758943 0.6551682 -1.0119778
```

```
[[4]]
```

```
[1] "yop"
```

## Définition (objet `data.frame`)

*C'est une liste à laquelle on impose certaines contraintes<sup>2</sup>, afin de rassembler vecteurs et facteurs sous la forme d'un tableau de données.*

- ▶ *Pratiquement, un tableau de données est une matrice dont les colonnes sont de mode différent,*
- ▶ C'est l'objet idéal pour la **manipulation de données** (**forcez-vous** à l'utiliser).

---

2. que je vous épargne

## Définition (objet `data.frame`)

*C'est une liste à laquelle on impose certaines contraintes<sup>2</sup>, afin de rassembler vecteurs et facteurs sous la forme d'un tableau de données.*

- ▶ *Pratiquement, un tableau de données est une matrice dont les colonnes sont de mode différent,*
- ▶ C'est l'objet idéal pour la **manipulation de données** (**forcez-vous** à l'utiliser).

---

2. que je vous épargne

## Syntaxe

On peut spécifier le nom des colonnes par le vecteur `row.names` ou directement comme pour une liste :

```
data.frame(e1=,e2=,...,row.names=)
```

```
> age <- c(25,35,32,27,32,40,26,25,26,28,30,NA,36,30,30)
> grd <- c("thd","CR","MdC","thd","thd","MdC","MdC","thd","thd","MdC","CR","MdC","CR")
> sexe <- factor(sample(c(rep("M",3),rep("F",12))))
> donnees <- data.frame(age=age,grade=grd,sexe=sexe)
> head(donnees)
```

	age	grade	sexe
1	25	thd	F
2	35	CR	M
3	32	MdC	F
4	27	thd	F
5	32	thd	F
6	40	MdC	F

# Manipulation des éléments du tableau de données

- ▶ Comme une liste !
- ▶ les commandes `attach()` / `detach` placent / ôtent les éléments du tableaux de données dans l'itinéraire de recherche.

```
> donnees$age
```

```
[1] 25 35 32 27 32 40 26 25 26 28 30 NA 36 30 30
```

```
> attach(donnees, warn.conflicts=FALSE)
```

```
> grade
```

```
[1] thd CR MdC thd thd MdC MdC thd thd MdC CR MdC CR thd thd
```

```
Levels: CR MdC thd
```

```
> detach(donnees)
```

# Travailler avec les tableaux de données

- ▶ beaucoup de fonctions prédéfinies
- ▶ penser aux fonctions `tapply` (ou `by`)

```
> summary(donnees)
      age      grade  sexe
Min.   :25.00   CR :3   F:12
1st Qu.:26.25   MdC:5   M: 3
Median :30.00   thd:7
Mean   :30.14
3rd Qu.:32.00
Max.   :40.00
NA's   :1

> attach(donnees, warn.conflicts=FALSE)
> by(age, sexe, mean, na.rm=TRUE)

sexe: F
[1] 29.75
-----

sexe: M
[1] 32.5

> by(age, grade, mean, na.rm=TRUE)

grade: CR
[1] 33.66667
-----

grade: MdC
[1] 31.5
-----

grade: thd
[1] 27.85714

> detach(donnees)
```



# Quatrième partie IV

## Statistiques descriptives

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

Représentations multivariées

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

Nous considérerons dans un premier temps une seule colonne du tableau de données, soient  $n$  observations :

$$x_1, \dots, x_n$$

# Variable quantitative discrète ou qualitative ordinale

La variable prend ses valeurs dans

$$V_x = \{\epsilon_1, \dots, \epsilon_K\}$$

avec

$$\epsilon_1 < \dots < \epsilon_K$$

## Tableau de fréquence

- ▶  $\epsilon_k$ , la modalité
- ▶  $n_k$ , l'effectif des observations ayant la valeur  $\epsilon_k$
- ▶  $f_k = \frac{n_k}{n}$ , la fréquence
- ▶  $F_k = \sum_{j=1}^k f_j$ , la fréquence relative cumulée

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

Même représentation sans l'ordre.  
Pas de fréquence cumulée.

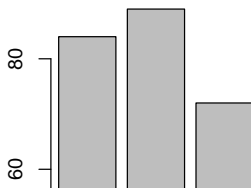
# Camembert et diagramme en barres

Couplés à la commande `table` Le diagramme en barres et le graphe en camembert permettent de visualiser le découpage d'une population en donnée catégorielle.

```
> vigne <- read.delim("mesures_baie_raisin_2008-2009.txt",header=TRUE)
> vigne <- vigne[-c(2,6,7)]
> colnames(vigne) <- c("pop", "pepin.08", "poids.08", "volcm3.08")
> head(vigne)

  pop pepin.08 poids.08 volcm3.08
1  CE        1.0     0.89      7.70
2  CE        1.0     1.14      8.82
3  CE        1.2     1.26     10.20
4  CE        1.2     0.66      NA
5  CE        1.2     0.83      NA
6  CE        1.3     0.54      4.61

> attach(vigne)
> par(mfrow=c(1,2))
> pie(table(pop))
> barplot(table(pop), las=3)
```





Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

- ▶ Moyenne :  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ 
  - ▶ Remarque : la somme des écarts à la moyenne empirique est nulle

$$\sum_i (x_i - \bar{x}) = 0$$

- ▶ Inconvénient : problème des valeurs aberrantes
- ▶ Moyenne tronquée :  $M_k = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} x_{(i)}$  où  $x_{(i)}$  est l'observation de rang  $i$
- ▶ Médiane :

$$M = \begin{cases} x_{(n/2)} & \text{si } n \text{ est pair,} \\ x_{(\lfloor n/2 \rfloor + 1)} & \text{sinon} \end{cases}$$

- ▶ Fractile empirique d'ordre  $\alpha$

$$\hat{f}_\alpha = \begin{cases} x_{(n\alpha)} & \text{si } n\alpha \text{ est entier,} \\ x_{(\lfloor n\alpha \rfloor + 1)} & \text{sinon} \end{cases}$$

- ▶ la variance empirique
- ▶ la variance empirique corrigée
- ▶ l'étendue
- ▶ l'étendue interquartile

# Reprenons l'exemple de la vigne

Renommons les variables et considérons uniquement les données de 2008 pour une manipulation plus agréable. J'enlève également la colonne « variété », car je ne vois pas à quoi elle sert.

```
> vigne <- read.delim("mesures_baie_raisin_2008-2009.txt",header=TRUE)
> vigne <- vigne[-c(2,6,7)]
> colnames(vigne) <- c("pop", "pepin.08", "poids.08", "volcm3.08")
> head(vigne)
```

	pop	pepin.08	poids.08	volcm3.08
1	CE	1.0	0.89	7.70
2	CE	1.0	1.14	8.82
3	CE	1.2	1.26	10.20
4	CE	1.2	0.66	NA
5	CE	1.2	0.83	NA
6	CE	1.3	0.54	4.61

```
> attach(vigne)
```

Les objets suivants sont masqués from vigne (position 3):

```
pepin.08, poids.08, pop, volcm3.08
```

Le résumé numérique s'adapte selon la nature des variables (univariée, multivariée, factorielle)

```
> summary(pepin.08)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	1.40	1.80	1.86	2.40	3.20	27

```
> summary(pop)
```

```
CE CO TE  
84 89 72
```

```
> summary(vigne)
```

pop	pepin.08	poids.08	volcm3.08
CE:84	Min. :0.00	Min. :0.39	Min. : 3.4
CO:89	1st Qu.:1.40	1st Qu.:0.82	1st Qu.: 7.1
TE:72	Median :1.80	Median :1.06	Median : 8.9
	Mean :1.86	Mean :1.21	Mean :10.5
	3rd Qu.:2.40	3rd Qu.:1.36	3rd Qu.:11.9
	Max. :3.20	Max. :3.75	Max. :33.8
	NA's :27	NA's :28	NA's :44

# Résumé statistique

Package Hmisc commande describe

```
> library(Hmisc)
> describe(vigne)
```

vigne

4 Variables      245 Observations

---

pop

n	missing	unique
245	0	3

CE (84, 34%), CO (89, 36%), TE (72, 29%)

---

pepin.08

n	missing	unique	Mean	.05	.10	.25
218	27	26	1.858	1.0	1.2	1.4
.50	.75	.90	.95			
1.8	2.4	2.7	2.9			

lowest : 0.0 0.5 0.8 1.0 1.1, highest: 2.8 2.9 3.0 3.1 3.2

---

poids.08

n	missing	unique	Mean	.05	.10	.25
217	28	122	1.212	0.578	0.676	0.820

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle



Dans le cas où la variable  $x$  est continue, la réalisation d'un tableau de fréquence nécessite un partitionnement préalable du domaine de définition en  $K$  classes de largeur

- ▶ constante
- ▶ ou variable

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

**Variable continue**

Résumés numériques

Tableau de fréquences

**Fonction de répartition empirique**

Histogramme et estimateur à noyaux

Boite à moustache

Représentations multivariées

Description bidimensionnelle

Description multidimensionnelle

$$\hat{F} : \mathbb{R} \mapsto [0, 1], x \mapsto \frac{1}{n} \text{card}\{i : x_i \leq x\}$$

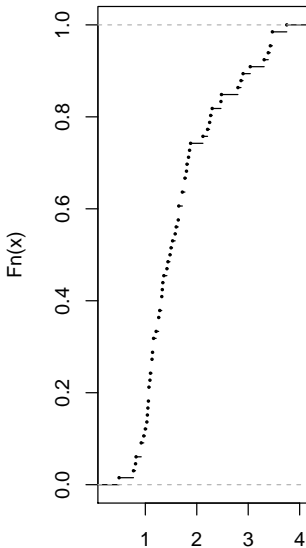
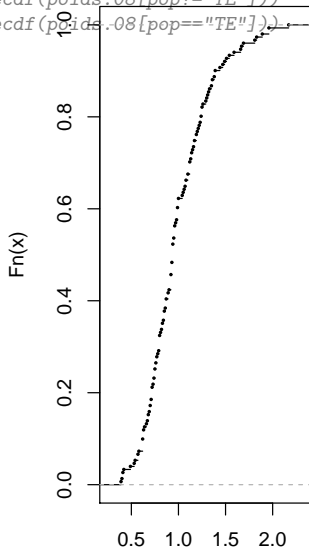
- ▶ Le graphe de la fonction de répartition est une fonction en escalier appelé diagramme cumulatif



# Fonction de répartition empirique

`ecdf` crée un objet qui peut être tracé avec `plot`.

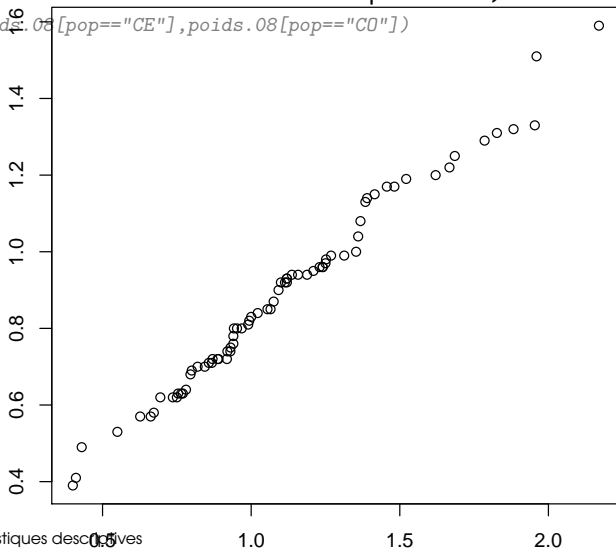
```
> par(mfrow=c(1,2))  
> plot(ecdf(poids.08[pop!="TE"]))  
> plot(ecdf(poids.08[pop=="TE"]))
```



# Comparaison de distribution

Pour comparer visuellement deux distributions, la manière la plus efficace est le graphe quantile/quantile (qui doivent correspondre si les distributions sont proches.)

```
> qqplot(poids.08[pop=="CE"],poids.08[pop=="CO"])
```



Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux**

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

- ▶ Estimateur de la fonction de densité

$$\hat{f}_n(x) = \sum_i h_i \mathbb{1}_{[a_i, a_{i+1}[}(x) \quad a_1 < \dots < a_{k+1}$$

- ▶ Découpage en intervalles
- ▶ Calcul de la fréquence
- ▶ Aire du rectangle proportionnel à la fréquence

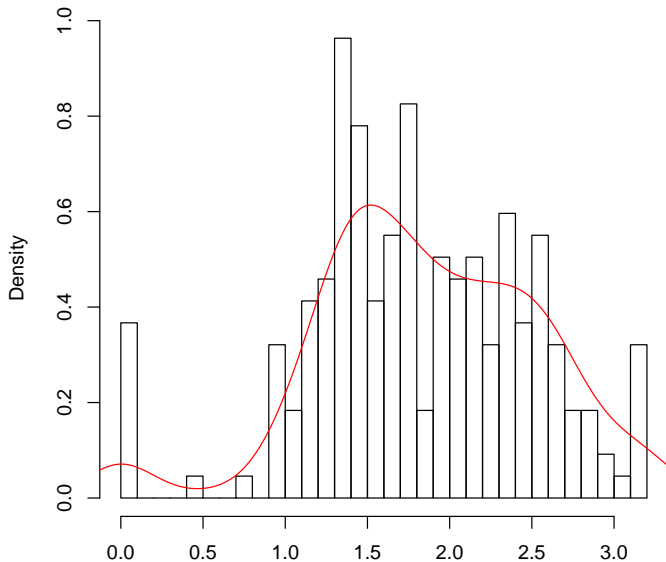
$$\sum_i h_i (a_{i+1} - a_i) = 1 \text{ et } h_i (a_{i+1} - a_i) = \hat{P}_F(X \in [a_i, a_{i+1}[)$$



- ▶ Attention : hauteur proportionnelle à la fréquence si et seulement si les intervalles ont tous la même largeur
- ▶ Nombre d'intervalles :
  - ▶ Important
  - ▶ Réglage difficile
  - ▶ Règle empirique : règle de Sturges  $1 + 10/3 * \log_{10}(n)$

# Histogramme

```
> hist(pepin.08,nclass=25,prob=TRUE)  
> lines(density(pepin.08[!is.na(pepin.08)]))
```



Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

**Variable continue**

Résumés numériques

Tableau de fréquences

Fonction de répartition empirique

Histogramme et estimateur à noyaux

**Boite à moustache**

Représentations multivariées

Description bidimensionnelle

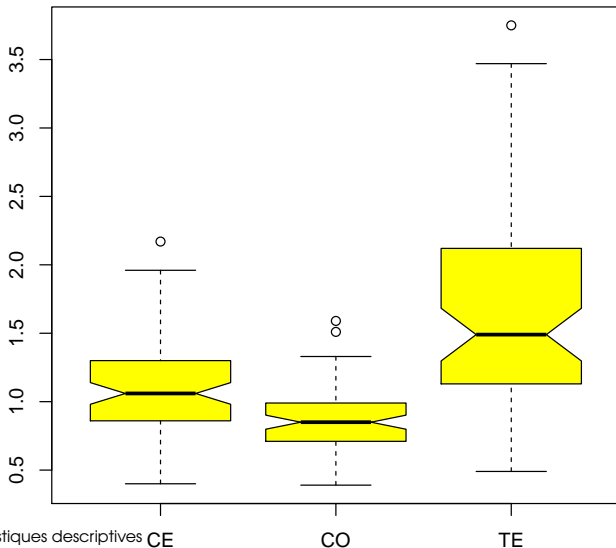
Description multidimensionnelle

- ▶ Éléments atypiques (aberrants, *outliers*)
  - ▶ Notion arbitraire
  - ▶ Règle empirique assez souvent utilisée : valeurs situées à l'extérieur de  $[q_1 - 1.5 \times Iqr, q_3 + 1.5 \times Iqr]$
- ▶ Définition : Graphique constitué
  - ▶ d'un rectangle délimité par les quartiles et partagé en deux par la médiane
  - ▶ d'une paire de moustaches : minimum et maximum de l'échantillon auquel on a ôté les éléments atypiques
  - ▶ des outliers eux-mêmes

# Boîtes à moustaches

La boîte à moustache permet de visualiser les grands traits caractéristiques d'une distribution.

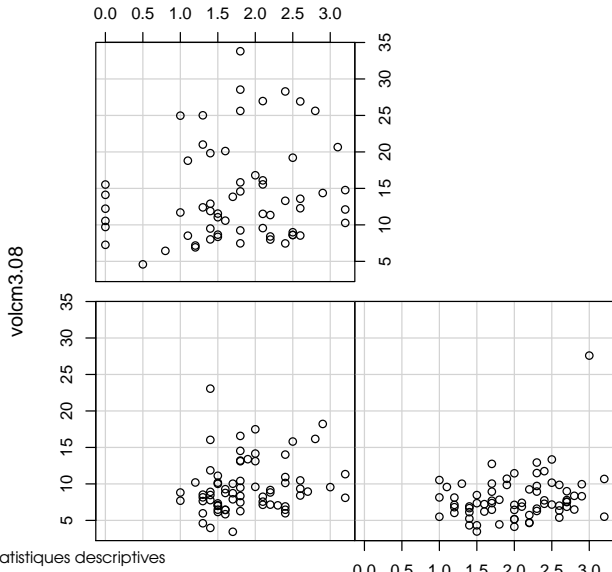
```
> boxplot(poids.08~pop,col="yellow",notch=T)
```



# Graphe conditionné par une variable

```
> coplot(volcm3.08 ~ pepin.08 | pop, show.given=FALSE)
```

Given : pop



Mise en évidence de certaines caractéristiques :

- ▶ Présence de données atypiques
- ▶ Absence de symétrie de la distribution
- ▶ Présence de populations hétérogènes
- ▶ ...

Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle



Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

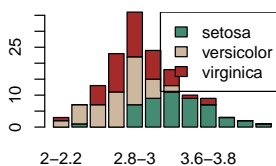
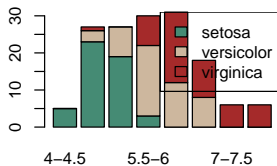
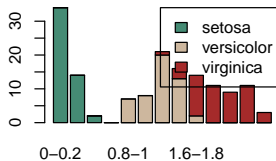
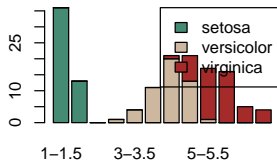
- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

# Histogrammes et variable qualitative



## ► Rappels

- $X$  réalisation d'un échantillon de taille  $n$  du vecteur aléatoire  $\mathbf{X}$
- $x_i$  réalisation de taille 1 de  $\mathbf{X}$
- $x^j$  réalisation d'un échantillon de taille  $n$  de  $X^j$

## ► Moyenne empirique

$$\bar{\mathbf{x}} = (\bar{x}^1, \dots, \bar{x}^p)' \quad \text{où} \quad \bar{x}^j = \frac{1}{n} \sum_{i=1}^n x_i^j$$

## ► Variance empirique

$$s_j^2 = \frac{1}{n} \sum_{i=1}^n (x_i^j - \bar{x}^j)^2$$

- ▶ Covariance empirique

$$s_{jj'} = \frac{1}{n} \sum_{i=1}^n (x_i^j - \bar{x}^j) \cdot (x_i^{j'} - \bar{x}^{j'})$$

- ▶ Coefficient de corrélation linéaire empirique

$$r_{jj'} = \frac{s_{jj'}}{s_j s_{j'}}$$

- ▶ Matrice de variance empirique

$$S = (s_{jj'}) = \frac{1}{n} (X - 1_n \bar{\mathbf{x}})' (X - 1_n \bar{\mathbf{x}}) = \frac{1}{n} Y' Y$$

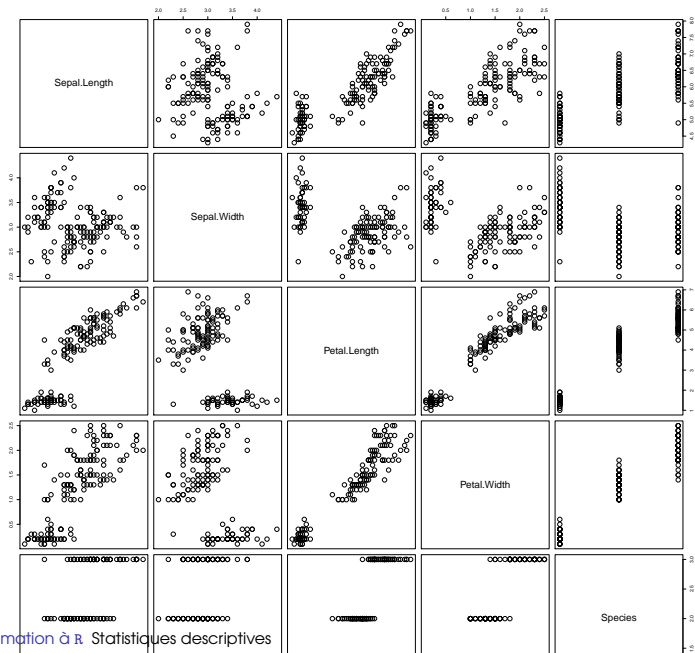
où  $1_n$  est la matrice de dimension  $(n, 1)$  remplie de 1 et  $Y$  est la matrice centrée associée à  $X$ .

- ▶ Matrice de corrélation empirique

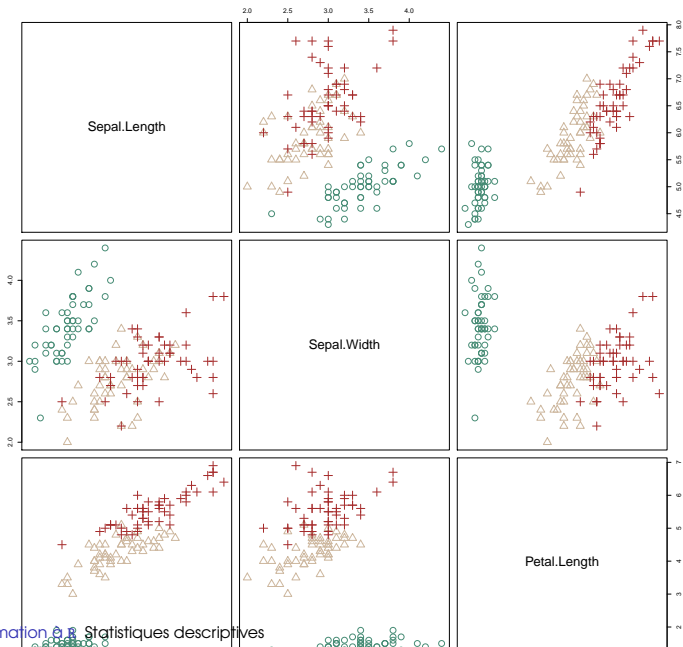
$$R = (r_{jj'}) = D_{1/s_j} S D_{1/s_{j'}}$$

- ▶ Représentation de chaque individu  $i$  par le point du plan  $(x_i^1, x_i^2)$
- ▶ Nuage de  $n$  points dans le plan
- ▶ Visualisation synthétique des données : permet de voir
  - ▶ les relations linéaires
  - ▶ les regroupements en classes homogènes

# Les 5 variables des iris



# Les 5 variables des iris en couleurs



- ▶ 2 variables : covariance et corrélation empirique
- ▶ > 2 variables : matrices de cov. et de corr. empiriques

## Les iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	0.69	-0.04	1.27	0.52
Sepal.Width	-0.04	0.19	-0.33	-0.12
Petal.Length	1.27	-0.33	3.12	1.30
Petal.Width	0.52	-0.12	1.30	0.58

TABLE : Matrice de covariance

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	0.69	-0.04	1.27	0.52
Sepal.Width	-0.04	0.19	-0.33	-0.12
Petal.Length	1.27	-0.33	3.12	1.30
Petal.Width	0.52	-0.12	1.30	0.58

TABLE : Matrice de corrélation



Variable quantitative discrète ou qualitative ordinale

Variable qualitative nominale

Variable continue

- Résumés numériques

- Tableau de fréquences

- Fonction de répartition empirique

- Histogramme et estimateur à noyaux

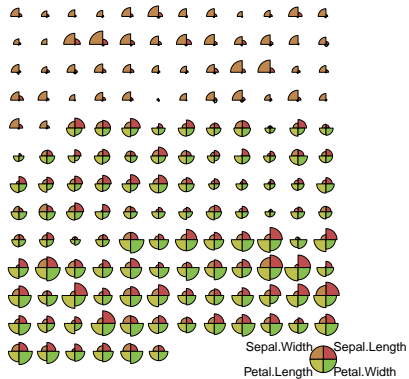
- Boite à moustache

Représentations multivariées

- Description bidimensionnelle

- Description multidimensionnelle

## Les iris



- ▶ Espace de grande dimension
- ▶ Calculs similaires à ceux du plan
- ▶ Mais difficile de généraliser
- ▶ Exemple 1 :
  - ▶ Dans  $\mathbb{R}$ 
    - ▶ Pts uniformément répartis dans  $[-1, +1]$
    - ▶ % de points situées à 1 distance  $\leq 0.75$  de l'origine : 75%
  - ▶ Dans  $\mathbb{R}^{10}$ 
    - ▶ Pts uniformément répartis dans  $[-1, +1]^{10}$
    - ▶ % de points situées à 1 distance  $\leq 0.75$  de l'origine : 5%
- ▶ Exemple 2 : on veut construire un histogramme en s'appuyant sur au moins une moyenne de 10 points par intervalle et 10 classes par variable
  - ▶  $\mathbb{R}$  : 10 classes  $n = 100$
  - ▶  $\mathbb{R}^2$  : 100 classes  $n = 1000$
  - ▶  $\mathbb{R}^{10}$  :  $10^{10}$  classes  $n = 10^{11} = 100 \text{ milliards}$

- ▶ Si  $p$  assez grand, l'espace  $\mathbb{R}^p$  est pratiquement vide et sauf si les données se situent au voisinage d'une variété de faible dimension, l'analyse des données n'apportera aucune information intéressante.
- ▶ Les points voisins d'un point donné sont tous très loin : difficultés dans l'emploi de méthodes du type  $k$ -plus proches voisins