

Biostatistiques avancées avec R

Représentation et gestion des données

Christophe Lalanne

www.aliquote.org

Synopsis

Éléments de contexte

Gestion des données avec R

Représentations graphiques

Importation et sauvegarde de données

Mesures d'association

Application

Le langage R

R est un logiciel pour le traitement et la modélisation de données statistiques^(11,18).

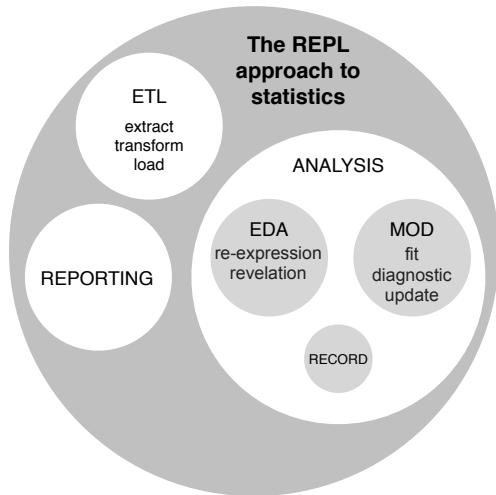
Il s'agit avant tout d'un langage de programmation inspiré du langage S développé dans les années 80^(1,2).

Le projet est maintenu par la *R Foundation for Statistical Computing* (www.r-project.org) et il est soutenu par le *R Consortium* (www.r-consortium.org).

Dans cet environnement interactif, l'utilisateur envoie des commandes, R les interprète et renvoie un résultat (« Read-Eval-Print-Loop », REPL).

Let's not kid ourselves : the most widely used piece of software for statistics is Excel — Brian Ripley

Approche interactive de l'analyse de données



Exemple de variable

On parlera de **variables** et de **commandes** pour distinguer les objets dans lesquels on stocke des données et les fonctions permettant d'opérer sur ces données, respectivement.

La commande `c()` permet d'associer une liste de valeurs à une variable ; l'opérateur d'assignation est le symbole `←` (ou `=`). Pour afficher le contenu d'une variable, il suffit de taper son nom ou d'utiliser `print()`.

```
> v <- c(1,1,2,3,5,8,13)
```

```
> v
```

```
[1] 1 1 2 3 5 8 13
```

```
> print(v)
```

```
[1] 1 1 2 3 5 8 13
```

```
> rm(v)
```

Représentation des données

Principaux types de variables : `numeric` (`integer`, `double`), `complex`, `character`, `logical`⁽²¹⁾.

```
> is.numeric(3.14)
```

```
[1] TRUE
```

```
> is.double(3.14)
```

```
[1] TRUE
```

```
> is.integer(3.14)
```

```
[1] FALSE
```

```
> as.integer(3.14)
```

```
[1] 3
```

Représentation des données

En pratique, on travaille rarement avec des variables isolées, mais plutôt avec un tableau de données où les observations sont arrangées en lignes et les variables en colonnes⁽²⁰⁾. Sous R, on appelle ce type de structure de données un « data frame ».

La commande `data()` permet d'importer des données disponibles dans les différents packages R. Généralement, les données sont immédiatement disponibles sous forme de data frame et la commande `ls()` permet de vérifier le nom du data frame importé dans l'espace de travail.

Pour visualiser l'en-tête des données, on peut utiliser la commande `head()`.

Voici par exemple les données d'une étude sur la longueur des odontoblastes (variable `len`) chez 10 cochons d'inde après administration de vitamine C à différentes doses (0,5, 1 ou 2 mg, variable `dose`) sous forme d'acide ascorbique ou de jus d'orange (variable `supp`)⁽⁵⁾.

```
> data(ToothGrowth)
```

```
> head(ToothGrowth)
```

	len	supp	dose
1	4.2	VC	0.5
2	11.5	VC	0.5
3	7.3	VC	0.5
4	5.8	VC	0.5
5	6.4	VC	0.5
6	10.0	VC	0.5

Une aide en ligne (`help()`) est généralement disponible pour les jeux de données internes.

```
> ls()
```

```
[1] "ToothGrowth"
```

```
> ## help(ToothGrowth)
```

```
> str(ToothGrowth)
```

```
'data.frame':      60 obs. of  3 variables:
 $ len : num  4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
 $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
 $ dose: num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

Propriétés d'un data frame

La commande `str()` fournit la taille du tableau de données [2], ainsi que le nom des variables, leur mode de représentation et un aperçu des 10 1^{re} observations [3-5].

```
1 > str(ToothGrowth)
2 'data.frame': 60 obs. of 3 variables:
3 $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 .
4 $ supp: Factor w/ 2 levels "OJ", "VC": 2 2 2 2 2 2 2 2 2 2
5 $ dose: num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 .
6 > names(ToothGrowth)
7 [1] "len" "supp" "dose"
8 > dim(ToothGrowth)
9 [1] 60 3
```

ToothGrowth

ToothGrowth[3,2]

Diagram illustrating the structure of the `ToothGrowth` data frame:

- `rownames()` points to the row indices (1 to 6).
- `ncol()` points to the column names (`len`, `supp`, `dose`).
- `nrow()` points to the number of rows (6).
- `colnames()` points to the column names (`len`, `supp`, `dose`).

	len	supp	dose
1	4.2	VC	0.5
2	11.5	VC	0.5
3	7.3	VC	0.5
4	5.8	VC	0.5
5	6.4	VC	0.5
6	10.0	VC	0.5
...

Sélection indexée d'observations

Notation : $[i, j]$, i^{e} ligne et j^{e} colonne.

```
> ToothGrowth[1,]
```

```
  len supp dose  
1 4.2   VC  0.5
```

```
> ToothGrowth[c(1,3),]
```

```
  len supp dose  
1 4.2   VC  0.5  
3 7.3   VC  0.5
```

```
> ToothGrowth[1:5, 2]
```

```
[1] VC VC VC VC VC  
Levels: OJ VC
```

```
> ToothGrowth[1:5, "supp"]
```

```
[1] VC VC VC VC VC  
Levels: OJ VC
```

Sélection critériée d'observations

Principe identique à la sélection indexée, sauf que les observations sont sélectionnés à partir de filtres logiques.

Notation : & (et), | (ou), ! (négation), %in% (ou ensembliste).

```
> ToothGrowth[ToothGrowth$supp == "VC" &  
+             ToothGrowth$dose == 0.5,  
+             "len"]
```

```
[1] 4.2 11.5 7.3 5.8 6.4 10.0 11.2 11.2 5.2 7.0
```

```
> ToothGrowth$supp[ToothGrowth$len < 11]
```

```
[1] VC VC VC VC VC VC VC VC OJ OJ OJ OJ OJ
```

```
Levels: OJ VC
```

Sélection critériée d'observations

La commande `subset()` est plus souple d'utilisation, et elle renvoie un data frame (et non un vecteur).

```
> subset(ToothGrowth, supp == "VC" & dose == 0.5, len)
```

```
      len
1    4.2
2   11.5
3    7.3
4    5.8
5    6.4
6   10.0
7   11.2
8   11.2
9    5.2
10   7.0
```

Résumé numérique

Pour obtenir un résumé numérique d'une variable (ou d'un data frame), on utilise `summary()`.

```
> summary(ToothGrowth$len)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.2	13.1	19.2	18.8	25.3	33.9

```
> summary(ToothGrowth$supp)
```

```
OJ VC
```

```
30 30
```

```
> summary(ToothGrowth)
```

len		supp	dose	
Min.	: 4.2	OJ:30	Min.	:0.50
1st Qu.:	13.1	VC:30	1st Qu.:	0.50
Median	:19.2		Median	:1.00
Mean	:18.8		Mean	:1.17
3rd Qu.:	25.3		3rd Qu.:	2.00
Max.	:33.9		Max.	:2.00

Résumé numérique

La variable dose peut être considérée comme numérique ou catégorielle. Dans le 2^e cas, il est nécessaire de la convertir en `factor()`.

```
> head(ToothGrowth$dose)
```

```
[1] 0.5 0.5 0.5 0.5 0.5 0.5
```

```
> unique(ToothGrowth$dose)
```

```
[1] 0.5 1.0 2.0
```

```
> head(factor(ToothGrowth$dose))
```

```
[1] 0.5 0.5 0.5 0.5 0.5 0.5
```

```
Levels: 0.5 1 2
```


A PERIODIC TABLE OF VISUALIZATION METHODS

 C circular	Data Visualization Visual representations of quantitative data in schematic form (either with or without axes)										Strategy Visualization The systematic use of complementary visual representations in the analysis, development, formulation, communication, and implementation of strategies in organizations.										 G graphic facilities		
 Tb table	 Ga cartesian coordinates	Information Visualization The use of interactive visual representations of data to analyze regions. This means that the data is transformed into an image; it is mapped to screen space. The image can be changed by users as they proceed working with it.										Metaphor Visualization Visual Metaphors position information graphically to organize and structure information. They also convey an insight about the represented information through the key characteristics of the metaphor that is employed.										 St story template	
 Pi pie chart	 L line chart	Concept Visualization Methods to elaborate (reify) qualitative concepts, ideas, plans, and analyses.										Compound Visualization The complementary use of different graphic representation formats in one single scheme or frame.										 Tr tree	
 B bar chart	 Ae area chart	 R radar chart	 Pa parallel coordinates	 Hy hierarchical tree	 Cy cycle diagram	 T timeline	 Ve venn diagram	 Mi mind map	 Sq square 4 oppositions	 Cc concentric circles	 Ar argument slide	 Sw swim lane diagram	 Cc cyclic chart	 Pm perspective diagram	 D diamond diagram	 Pr parameter ruler	 Kn knowledge map	 Me meeting trace	 Mm metro map	 Tm temple	 Br bridge	 Fu funnel	 Ri rich picture
 Hi histogram	 Sc scatterplot	 Sa sashy diagram	 In information base	 E entity relationship diagram	 Pt point net	 Fi flow chart	 Cl clustering	 Lc layer chart	 Py pyramid	 Ce cause-effect chains	 Ti tree	 Dt decision tree	 Cp open critical path method	 Cf concept fan	 Co concept map	 Ic iceberg	 Lm learning map	 Go goal diagram	 Br bridge	 Co concept	 Br bridge	 Fu funnel	 Ri rich picture
 Tk tally bar plot	 Sp spectrogram	 Da data map	 Trp treemap	 Cn cone tree	 Sy system dynamics simulation	 Df data flow diagram	 Se semantic network	 So soft system modeling	 Sn snake map	 Fo force field diagram	 Ib iceberg	 Pr process model	 Pt part chart	 Ev evolutionary map	 V vow diagram	 Hh honey 'h' cell chart	 I informal	 Go goal diagram	 Br bridge	 Co concept	 Br bridge	 Fu funnel	 Ri rich picture

Cy Process Visualization

Hy Structure Visualization

Overview
Detail

Detail AND Overview

< > Divergent thinking

> < Convergent thinking

Note: Depending on your location and connection speed it can take some time to load a pop-up picture.

© Ralph Lengler & Martin J. Eppler: www.visual-literacy.org

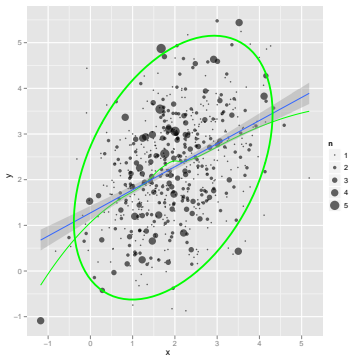
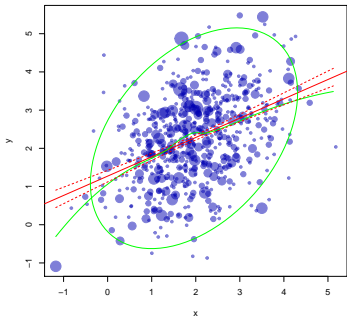
version 1.5

> < < > Su supply demand curve	> < < > Pe performance charting	> < < > St strategic map	> < < > Oc organizational chart	< < < > Ho house of quality	> < < > Fd feedback diagram	> < < > Ft failure tree	> < < > Mq map quadrant	> < < > Ld life-cycle diagram	> < < > Po porter's five forces	< < < > S s-cycle	> < < > Sm submodel map	> < < > Is islands diagram	> < < > Tc technology roadmap
> < < > Ed edge-weighted tree	> < < > Pf periodic diagram	> < < > Sg single gene board	> < < > Mz metaphor's organograph	> < < > Z zoo's morphological box	> < < > Ad advent diagram	> < < > De decision discovery diagram	> < < > Bm big matrix	> < < > Stc strategic canvas	> < < > Vc value chain	> < < > Hy hyper-cycle	> < < > Sr scheduler rating map	> < < > Ta taps	< < < > Sd spring diagram

http://www.visual-literacy.org/periodic_table/periodic_table.html

Systèmes graphiques sous R

R dispose de deux principaux système graphiques – base et grid⁽¹⁴⁾ – et de trois interfaces : graphics⁽³⁾, **lattice**⁽¹⁵⁾ et **ggplot2**⁽¹⁹⁾.



Le package **lattice** sera utilisé pour la plupart des illustrations graphiques.

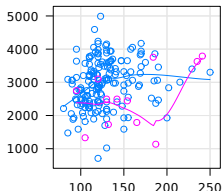
Le package lattice

histogram()	histogramme (effectif, fréquence, densité)
densityplot()	courbe de densité non-paramétrique
stripplot()	diagramme de dispersion univarié
qqmath()	« quantile plot »
bwplot()	diagramme en boîte à moustaches
barchart()	diagramme en barres
dotplot()	diagramme de Cleveland
xyplot()	diagramme de dispersion

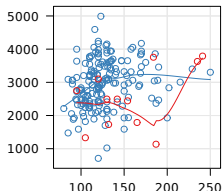
- Graphiques prêts à l'usage pour l'analyse exploratoire et la modélisation, légende automatique, conditionnement, thèmes graphiques
- Personnalisation délicate (idem pour graphics et [ggplot2](#))

```
> library(lattice)
> if (!require(latticeExtra))
+   install.packages("latticeExtra")
> lattice.options(default.args = list(axis = axis.grid))
> trellis.par.set(ggplot2like(lwd = 2.5))
```

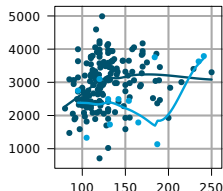
Thème par défaut



custom.theme.2()



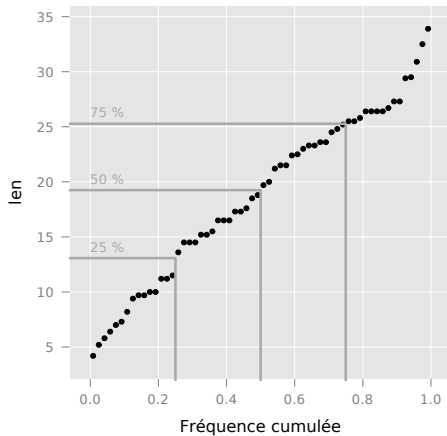
theEconomist.theme()



Use excellent graphics, liberally — Frank E Harrell

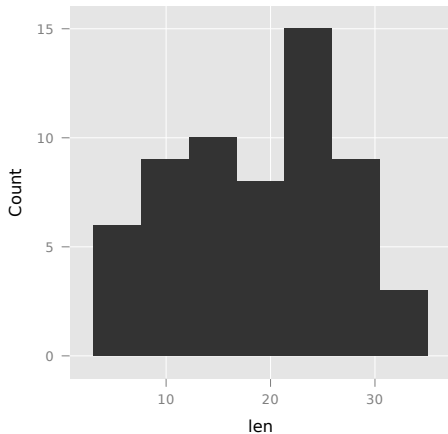
Fonction de répartition

```
> qqmath(~ len, data = ToothGrowth, dist = qunif)
```



Histogramme d'effectifs

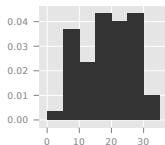
```
> histogram(~ len, data = ToothGrowth, type = "count")
```



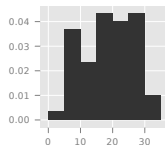
Histogramme d'effectifs

Le paramètre `breaks=` permet de changer le nombre d'intervalles construits (voir aussi `nint=`). Par défaut, la méthode utilisée est la **méthode de Sturges**⁽¹⁷⁾. L'ajout d'une loi de densité gaussienne dont les paramètres sont estimés à partir de l'échantillon est également possible.

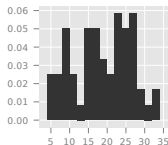
5 intervalles



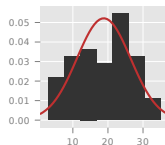
10 intervalles



15 intervalles



N(18.8;7.6) superposée



Graphiques en trellis

Les graphiques en trellis^(6,4) offrent une structure de graphique simple et efficace pour représenter des données multidimensionnelles. En particulier, ils introduisent la notion de **facettes** pour représenter des distributions conditionnelles.

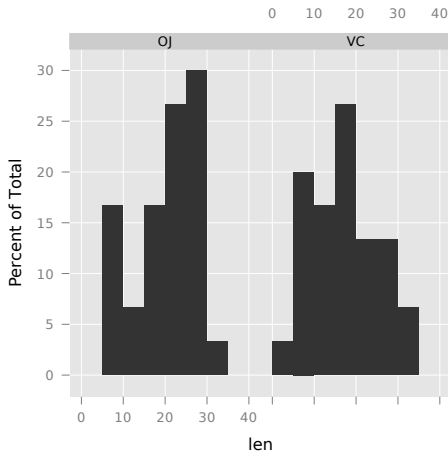
La notation utilisée est la **notation par formule** :

- $y \sim x \mid a$, y en fonction de x condit. à a
- $y \sim x \mid a + b$, y en fonction de x condit. à a et b

Les variables continues peuvent être « catégorisées » à l'aide de « shingles ». En transformant les variables numériques en facteurs, il devient possible de représenter un plus grand nombre de croisement de variables, tout en tenant compte de la nature continue des données.

Histogramme de fréquences relatives

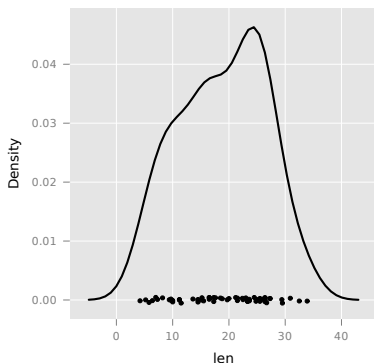
```
> histogram(~ len | supp, data = ToothGrowth,  
+           breaks = seq(0, 40, by = 5))
```

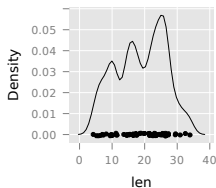
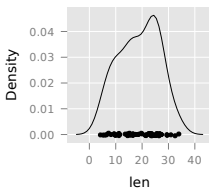
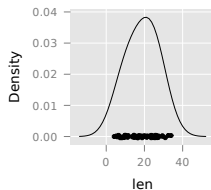


Courbe de densité

Pour pallier à l'arbitraire du choix du nombre d'intervalles, on peut préférer représenter la fonction de densité empirique^(18,16). Il reste toutefois à définir la largeur de la fenêtre de lissage associée à la fonction ou noyau (voir `help(bw.nrd0)` ou §5.6⁽¹⁸⁾).

```
> densityplot(~ len, data = ToothGrowth)
```



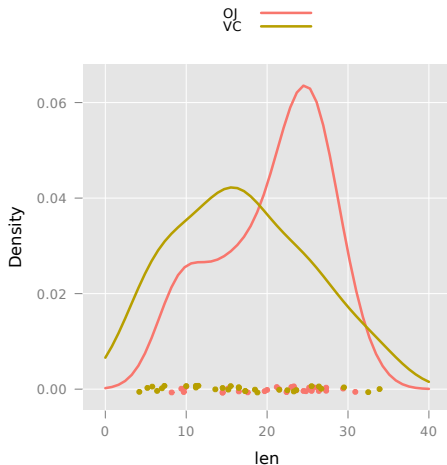
adjust = 0.5**adjust = 1 (défaut)****adjust = 2**

$$\hat{f}(x) = \frac{1}{nb} \sum_{j=1}^n K\left(\frac{x - x_j}{b}\right)$$

$K()$ noyau donné (par défaut, gaussien), et b largeur de la fenêtre de lissage : $\hat{b} = 1.06 \text{ ou } 0.9 \min(\hat{\sigma}, R/1.34) n^{-1/5}$.

Courbe de densité conditionnelle

```
> densityplot(~ len, data = ToothGrowth, groups = supp,  
+             from = 0, to = 40, auto.key = TRUE)
```

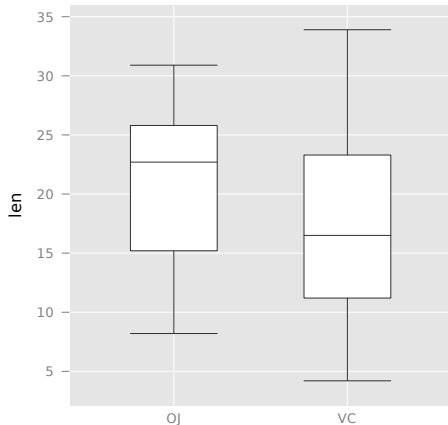


Graphics *versus* Lattice

```
1 > plot(density(ToothGrowth$len[ToothGrowth$supp == "OJ"]),
2       main = "", xlab = "len", las = 1, lwd = 2, col = "coral")
3 > lines(density(ToothGrowth$len[ToothGrowth$supp == "VC"]),
4       lwd = 2, col = "cornflowerblue")
5 > ## rug() does not allow to use a grouping factor
6 > points(x = ToothGrowth$len[ToothGrowth$supp == "OJ"],
7        y = runif(n = length(ToothGrowth$len[ToothGrowth$supp == "OJ"]),
8              min = -0.001, max = 0.001),
9        col = "coral")
10 > points(x = ToothGrowth$len[ToothGrowth$supp == "VC"],
11        y = runif(n = length(ToothGrowth$len[ToothGrowth$supp == "VC"]),
12              min = -0.001, max = 0.001),
13        col = "cornflowerblue")
14 > legend("top", levels(ToothGrowth$supp),
15        col = c("coral", "cornflowerblue"),
16        lty = 1, bty = "n")
```

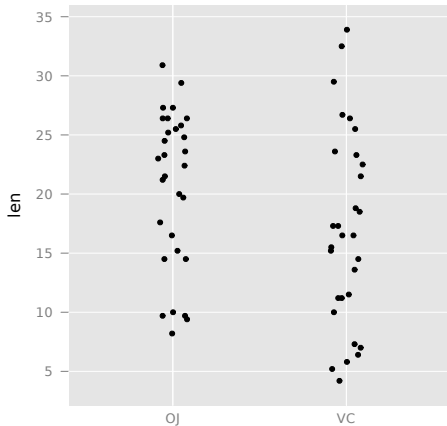
Diagramme de type boîtes à moustaches

```
> bwplot(len ~ supp, data = ToothGrowth, pch = "|")
```



Diagrammes en barres et en points

```
> dotplot(len ~ supp, ToothGrowth, jitter.x = TRUE)
```



Type de fichiers

R peut lire des données enregistrées sous de nombreux formats :

- fichiers Stata, SPSS, SAS (packages `foreign`, `Hmisc`, `readr`)
- fichiers texte (`read.table()`)
- fichiers MS Excel (packages `xlsx`, `readxl`)
- base de données relationnelles ou NoSQL (packages `DBI`, `RMySQL`, `RPostgreSQL`, `RMongo`, `RODBC`, `RSQLite`)
- données non structurées de type JSON (packages `rjson`, `jsonlite`)

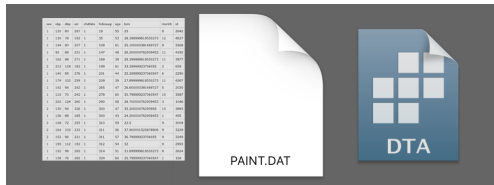
Les fichiers peuvent être enregistrés sur le disque ou lus directement depuis internet.

Lecture d'un fichier texte

help(read.table)

```
1 read.table(file, header = FALSE, sep = "", quote = "\"",
2           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
3           row.names, col.names, as.is = !stringsAsFactors,
4           na.strings = "NA", colClasses = NA, nrow = -1,
5           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
6           strip.white = FALSE, blank.lines.skip = TRUE,
7           comment.char = "#",
8           allowEscapes = FALSE, flush = FALSE,
9           stringsAsFactors = default.stringsAsFactors(),
10          fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
11
12 read.csv(file, header = TRUE, sep = ",", quote = "\"",
13         dec = ".", fill = TRUE, comment.char = "", ...)
14
15 read.csv2(file, header = TRUE, sep = ";", quote = "\"",
16         dec = ",", fill = TRUE, comment.char = "", ...)
17
18 read.delim(file, header = TRUE, sep = "\t", quote = "\"",
19         dec = ".", fill = TRUE, comment.char = "", ...)
```

Exemples de fichiers



<https://github.com/PF-BB/Biostat/data>

- Framingham.csv, « Framingham Heart Study »^(12,7)
- PAINT.DAT, « Health survey of paint sprayers »⁽⁸⁾
- polymorphism.dta, « Polymorphisme et gène du récepteur estrogène »⁽⁷⁾

L'étude Framingham

« Framingham Heart Study »^(12,7)

```
> fhs <- read.csv("data/Framingham.csv")
```

```
> fhs <- subset(fhs, fhs$id != 9999 & complete.cases(fhs))
```

```
> dim(fhs)
```

```
[1] 4658 10
```

```
> names(fhs)[1:5]
```

```
[1] "sex"      "sbp"      "dbp"      "scl"      "chdfate"
```

Données à l'inclusion

sbp	systolic blood pressure (SBP) in mm Hg
dbp	diastolic blood pressure (DBP) in mm Hg
age	age in years
scl	serum cholesterol (SCL) in mg/100ml
bmi	body mass index (BMI) = $\text{weight}/\text{height}^2$ in kg/m ²
sex	gender (1=male, 2=female)
month	month of year in which baseline exam occurred
id	patient identification variable (numbered 1 to 4699)

Données de suivi

followup	follow-up in days
chdfate	CHD outcome (1=patient develops CHD at the end of follow-up, 0=otherwise)

Recodage et agrégation de données

```
> fhs$sex <- factor(fhs$sex, levels = 1:2,  
+                   labels = c("M", "F"))  
> summary(fhs$sex)
```

```
  M    F  
2040 2618
```

```
> aggregate(sbp ~ sex, data = fhs, mean)
```

```
  sex sbp  
1  M 132  
2  F 133
```

```
> aggregate(sbp ~ sex, data = fhs, sd)
```

```
  sex sbp  
1  M 19.7  
2  F 24.9
```

Classification IMC (OMS, <http://goo.gl/JxzT>)

```
> summary(fhs$bmi)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
16.2	22.9	25.2	25.6	28.0	57.6

```
> fhs$bmi.cat <- cut(fhs$bmi, breaks=c(16,18.5,25,30,58),  
+                    right = FALSE)
```

```
> summary(fhs$bmi.cat)
```

[16,18.5)	[18.5,25)	[25,30)	[30,58)
71	2139	1852	596

```
> levels(fhs$bmi.cat) <- c("Under", "Normal", "Over", "Obese")
```

```
> ## relevel(fhs$bmi.cat, ref = "Normal")
```

Tableaux d'effectifs et de fréquences

```
> xtabs(~ sex + bmi.cat, data = fhs)
```

```
  bmi.cat
```

```
sex Under Normal Over Obese
```

```
 M    14    787 1013   226
```

```
 F    57   1352  839   370
```

```
> r <- xtabs(~ sex + bmi.cat, data = fhs)
```

```
> margin.table(r, margin = 2)
```

```
bmi.cat
```

```
 Under Normal    Over  Obese
```

```
    71   2139   1852    596
```

```
> prop.table(r, margin = 2)
```

```
  bmi.cat
```

```
sex Under Normal    Over  Obese
```

```
 M 0.197  0.368 0.547 0.379
```

```
 F 0.803  0.632 0.453 0.621
```

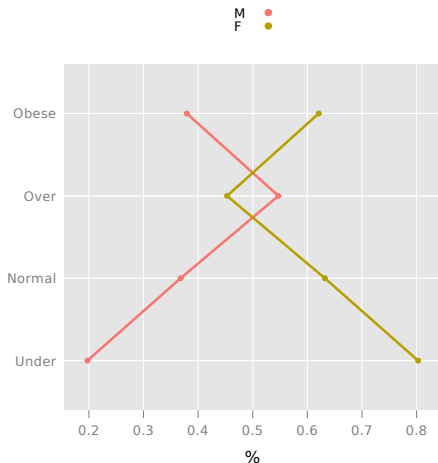
Format de représentation d'un tableau

```
> pr <- prop.table(r, margin = 2)  
> as.data.frame(pr)
```

	sex	bmi.cat	Freq
1	M	Under	0.197
2	F	Under	0.803
3	M	Normal	0.368
4	F	Normal	0.632
5	M	Over	0.547
6	F	Over	0.453
7	M	Obese	0.379
8	F	Obese	0.621

Diagramme de fréquences relatives

```
> dotplot(bmi.cat ~ Freq, data = as.data.frame(pr),  
+         groups = sex, xlab = "%",  
+         type = c("p", "l"), auto.key = TRUE)
```



Association entre deux variables numériques

```
> summary(fhs[,c("sbp", "age")])
```

sbp	age
Min. : 80	Min. : 30
1st Qu.: 116	1st Qu.: 39
Median : 130	Median : 45
Mean : 133	Mean : 46
3rd Qu.: 144	3rd Qu.: 53
Max. : 270	Max. : 66

```
> cor(fhs$sbp, fhs$age)
```

```
[1] 0.395
```

```
> cor(fhs$sbp, fhs$age, method = "spearman")
```

```
[1] 0.389
```

Diagramme de dispersion

```
> xyplot(sbp ~ age, data = fhs, type = c("p", "smooth"))
```

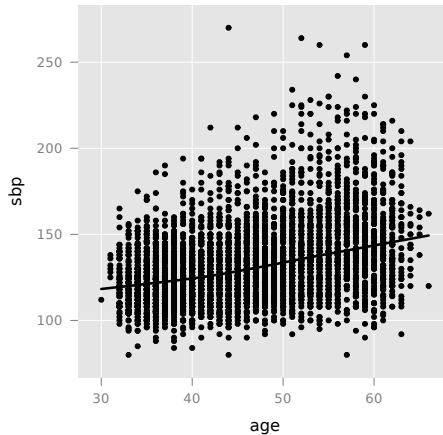
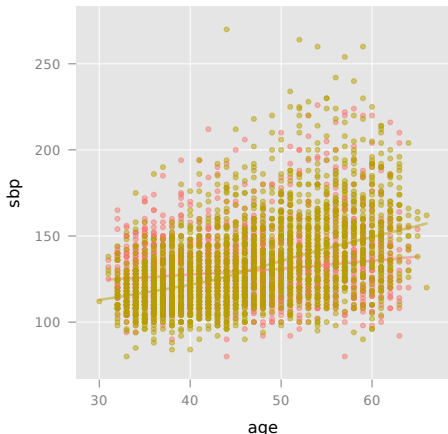


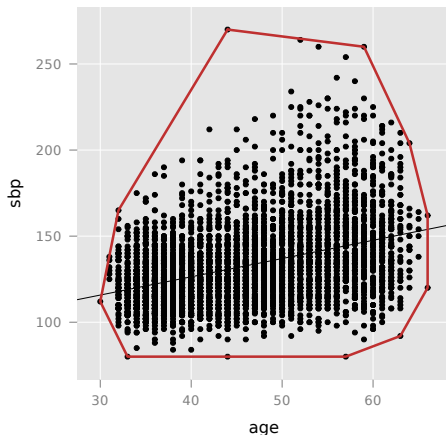
Diagramme de dispersion conditionnel

```
> xyplot(sbp ~ age, data = fhs, groups = sex,  
+        type = c("p", "smooth"), alpha = 0.5)
```



Enveloppe d'un nuage de points

```
> ch <- with(fhs, chull(sbp, age))  
> cor(fhs$sbp[-ch], fhs$age[-ch])  
[1] 0.398
```



Ce qu'il faut retenir

- Il est important de vérifier le **codage des variables**, et de recoder en fonction des besoins de l'analyse ou de la visualisation.
- On caractérise d'abord les **distributions univariées** avant de passer aux visualisations ou aux modèles multivariés. Cela permet de détecter les éventuelles valeurs aberrantes, la mauvaise représentation de certaines modalités d'une variable catégorielle, ou l'existence d'asymétrie dans les distributions.
- L'interface **lattice** utilise la même **notation par formule** que les fonctions R pour la modélisation statistique.
- De nombreux outils sont disponibles dans les packages **vcd** et **vcdExtra**^(13,9) pour la visualisation des données catégorielles.

Étude sur les poids de naissance

« The low birth weight study »

Il s'agit d'une étude prospective visant à identifier les facteurs de risque associés à la naissance de bébés dont le poids est inférieur à la norme (2,5 kg). Les données proviennent de 189 femmes, dont 59 ont accouché d'un enfant en sous poids. Parmi les variables d'intérêt figurent l'âge de la mère, le poids de la mère lors des dernières menstruations, l'ethnicité de la mère et le nombre de visites médicales durant le premier trimestre de grossesse⁽¹⁰⁾.

Elle est disponible sous R dans le package **MASS** :

```
> data(birthwt, package="MASS")
```

Exercices

1. Recoder les variables catégorielles en facteur et faire un résumé numérique et graphique de chaque variable.
2. Quel est le poids moyen des femmes qui fumait durant leur grossesse ?
3. Combien dénombre t-on d'antécédents d'hypertension chez les femmes pesant plus de 60 kg (sachant que les mesures du fichier sont exprimées en livres) ?
4. Quel est le poids minimal des bébés chez les mères n'ayant pas manifesté une irritabilité utérine ?
5. Recoder la variable `ftv` en variable binaire (0 ou 1+).
6. Construire un tableau croisant cette variable avec la variable `low` ; calculer la moyenne et l'écart-type pour la variable `bwt` dans les deux groupes d'individus définis par cette variable `ftv01`.

Références I

1. RA Becker and JM Chambers. *S : A language and system for data analysis*. Bell Laboratories Computer Information Service, Murray Hill, New Jersey, 1981.
2. RA Becker and JM Chambers. *S : An Interactive Environment for Data Analysis and Graphics*. Wadsworth, 1984.
3. RA Becker, JM Chambers, and AR Wilks. *The New S Language*. Wadsworth & Brooks/Cole, 1988.
4. RA Becker, WS Cleveland, and MJ Shyu. The visual design and control of trellis display. *Journal of Computational and Statistical Graphics*, 5(2) :123–155, 1996.
5. CI Bliss. *The Statistics of Bioassay*. Academic Press, 1952.
6. WS Cleveland. *Visualizing Data*. Hobart Press, 1993.
7. WD Dupont. *Statistical Modeling for Biomedical Researchers*. Cambridge University Press, 2ème édition, 2009.
8. B Everitt and S Rabe-Hesketh. *Analyzing Medical Data Using S-PLUS*. Springer, 2001.
9. M Friendly. Tutorial : Working with categorical data with r and the vcd package, 2011. URL <http://www.datavis.ca/courses/VCD>.
10. D Hosmer and S Lemeshow. *Applied Logistic Regression*. New York : Wiley, 1989.

Références II

11. R Ihaka and R Gentleman. R : A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3) :299–314, 1996.
12. D Levy. 50 years of discovery : Medical milestones from the national heart, lung, and blood institute's Framingham Heart Study. Hackensack, N.J. : Center for Bio-Medical Communication Inc., 1999.
13. D Meyer, A Zeilis, and K Hornik. The strucplot framework : Visualizing multi-way contingency tables with vcd. *Journal of Statistical Software*, 17(3), 2006.
14. P Murrell. *R Graphics*. Chapman & Hall/CRC, 2011. URL <https://www.stat.auckland.ac.nz/~paul/RG2e>.
15. D. Sarkar. *Lattice : Multivariate Data Visualization with R*. Springer, 2008. URL <http://lmdvr.r-forge.r-project.org>.
16. BW Silverman. *Density estimation*. Chapman and Hall, 1986.
17. HA Sturges. The choice of a class interval. *Journal of the American Statistical Association*, pages 65–66, 1926.
18. WN Venables and BD Ripley. *Modern Applied Statistics with S*. Springer, 4ème édition, 2002. ISBN 0-387-95457-0.
19. H Wickham. *ggplot2 : Elegant Graphics for Data Analysis*. Springer, 2009. URL <http://ggplot2.org>.
20. H Wickham. Tidy data. *Journal of Statistical Software*, 59 :1–23, 2014.
21. H Wickham. *Advanced R*, 2015. URL <http://adv-r.had.co.nz>.

Index des commandes

aggregate, 37	dotplot, 31, 41	lines, 29	read.table, 32, 33
as.data.frame, 40	double, 6	logical, 6	require, 20
bwplot, 30	factor, 16, 37	ls, 7, 9	rm, 5
c, 5, 29	head, 7, 8, 16	margin.table, 39	runif, 29
character, 6	help, 9, 26, 33	max, 29	str, 9, 10
chull, 45	histogram, 22, 25	min, 29	subset, 14, 35
col, 29	install.packages, 20	names, 10, 35	summary, 15, 37, 38, 42
complex, 6	integer, 6	numeric, 6	trellis.par.set, 20
cor, 42, 45	lattice.options, 20	plot, 29	unique, 16
cut, 38	legend, 29	points, 29	xtabs, 39
data, 7, 8, 47	length, 29	print, 5	xyplot, 43, 44
density, 29	levels, 10, 29, 38	prop.table, 39, 40	
densityplot, 26, 28	library, 20	qqmath, 21	
dim, 10, 35		quote, 33	
		read.csv, 35	