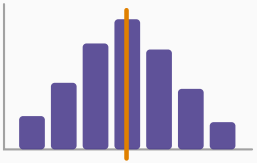
Statistique descriptive

Chapitre 15 : Mise en œuvre sur R



Jaouad Madkour

Faculté d'Économie et de Gestion de Tanger · 27 août 2026

Plan du chapitre

Refaire toute la chaîne

Ce chapitre reprend les quatorze précédents

Étape	Chapitres
Importer et décrire les variables	2
Tableau statistique et classes	3, 4
Les quatre graphiques	5
Tendance centrale, quantiles, dispersion, forme	6 à 9
Lorenz et Gini	10
Contingence et conditionnelles	11, 12
Covariance, corrélation, ajustement	13, 14

Le programme (suite)

Le but n'est pas d'apprendre R

C'est de **vérifier** que les résultats calculés à la main sont bien ceux que produit un logiciel — et de constater que quinze lignes de code remplacent un semestre de calculs.

Comprendre la formule reste indispensable : c'est ce qui permet de savoir **quelle** fonction demander et de reconnaître un résultat aberrant.

Charger et explorer

L'import

```r
# Les données : 500 salariés, six colonnes
sal <- read.csv("salaires.csv", stringsAsFactors = TRUE)

str(sal)
head(sal)
summary(sal)
```

La sortie de str()

```
'data.frame': 500 obs. of  6 variables:
 $ id         : int  1 2 3 4 5 ...
 $ salaire    : int  8540 7100 4380 ...
 $ anciennete: int  10 15 4 ...
 $ diplome    : Factor w/ 3 levels "Bac","Bac+2","Bac+5"
 $ region     : Factor w/ 4 levels "Autres","Casablanca",...
 $ sexe       : Factor w/ 2 levels "F","H"
```

Le premier réflexe : vérifier les types

`Factor` pour les qualitatives, `int` ou `num` pour les quantitatives.

Si R a lu `salaire` comme un facteur, c'est qu'une ligne contient du texte — un espace, une virgule décimale, un « n.d. ». **Toutes les moyennes seraient alors impossibles.**

C'est la première erreur à chercher, et elle survient dans un fichier réel sur deux.

```r
colSums(is.na(sal))          # combien de valeurs manquantes par colonne
nrow(sal)                    # 500
length(unique(sal$salaire))  # 381 valeurs distinctes
```

Pourquoi 381 valeurs distinctes comptent

C'est le critère du chapitre 2 : 381 valeurs pour 500 individus, donc on traitera le salaire comme **continu** et on le regroupera en classes.

Le tableau statistique

Variable qualitative

```r
n  <- nrow(sal)
ni <- table(sal$diplome)                    # effectifs
fi <- prop.table(ni)                        # fréquences
Ni <- cumsum(ni)                            # effectifs cumulés
Fi <- cumsum(fi)                            # fréquences cumulées

tableau <- data.frame(ni, fi = as.numeric(fi),
                      Ni = as.numeric(Ni), Fi = as.numeric(Fi))
print(tableau, digits = 3)
```

Le résultat

```
   Var1 Freq    fi  Ni    Fi
1   Bac  219 0.438 219 0.438
2 Bac+2  174 0.348 393 0.786
3 Bac+5  107 0.214 500 1.000
```

Exactement le tableau du chapitre 3.

Variable continue : le découpage en classes

```r
bornes <- c(2000, 4000, 6000, 8000, 10000, 14000, 20000)
cl <- cut(sal$salaire, breaks = bornes, right = FALSE)   # [a ; b[
ni <- table(cl)
ai <- diff(bornes)                          # amplitudes
di <- as.numeric(ni) / ai                   # densités

data.frame(classe = names(ni), ni = as.numeric(ni),
           fi = as.numeric(ni)/n, ai, di)
```

L'argument right = FALSE est essentiel

Par défaut, R crée des classes $]a\,;\,b]$ — **ouvertes à gauche**, fermées à droite. C'est l'inverse de la convention française.

`right = FALSE` rétablit $[a\,;\,b[$. Sans lui, tous les effectifs sont décalés aux bornes.

Les graphiques

Les quatre représentations

```r
par(mfrow = c(2, 2))

barplot(table(sal$diplome), col = "steelblue",
        main = "Diplôme", ylab = "effectif")

pie(table(sal$region), main = "Région")

hist(sal$salaire, breaks = bornes, freq = FALSE, col = "steelblue",
     main = "Salaire", xlab = "dirhams", ylab = "densité")

plot(ecdf(sal$salaire), main = "Cumulative croissante",
     xlab = "dirhams", ylab = "F(x)")
```

Les quatre représentations (suite)

L'argument qui compte : freq = FALSE

Avec `freq = TRUE` (le défaut), `hist` trace les **effectifs** : l'histogramme est faux dès que les amplitudes sont inégales.

`freq = FALSE` trace les **densités** — la règle du chapitre 4. **À écrire systématiquement.**

La boîte à moustaches

```
boxplot(salaire ~ diplome, data = sal,
        col = c("steelblue", "orange", "seagreen"),
        ylab = "salaire (dirhams)")
```

La syntaxe formule

`y ~ x` se lit « y **selon** x ». C'est la notation centrale de R, et elle servira à toute la modélisation.

Ici elle produit directement les distributions **conditionnelles** du chapitre 12.

Les indicateurs univariés

Tendance centrale et dispersion

```r
x <- sal$salaire

mean(x)                          # 7158.26
median(x)                        # 6770
quantile(x, c(.10,.25,.50,.75,.90))
IQR(x)                           # 3012
var(x)  * (n-1)/n                # 6750122   variance descriptive
sd(x)   * sqrt((n-1)/n)          # 2598.10   écart-type descriptif
sd(x)/mean(x)                    # 0.363     coefficient de variation
```

Le piège n contre n-1

R divise par $n-1$, pas par n. `var()` calcule la variance dite corrigée, celle de l'inférentielle.

La variance **descriptive** du chapitre 8 divise par n. D'où la correction `* (n-1)/n`.

Sur 500 observations l'écart est de $0{,}2\,\%$ — négligeable en pratique, mais il faut savoir qu'il existe, et il devient sensible sur de petits échantillons. **Un étudiant qui trouve un écart avec son calcul manuel doit vérifier ce point en premier.**

Forme et concentration

```r
library(moments)
skewness(x)                    # 1.164   asymétrie
kurtosis(x)                    # 4.837   aplatissement, référence 3

gini <- function(v) {
  v <- sort(v); n <- length(v)
  2*sum(seq_len(n)*v)/(n*sum(v)) - (n+1)/n
}
gini(x)                        # 0.195
```

La courbe de Lorenz

```r
# Courbe de Lorenz
v <- sort(x)
F <- seq_len(n)/n
Q <- cumsum(v)/sum(v)
plot(c(0,F), c(0,Q), type = "l", lwd = 2, col = "steelblue",
     xlab = "part cumulée des salariés",
     ylab = "part cumulée de la masse salariale")
abline(0, 1, col = "grey50")
```

Vérifier la convention d'aplatissement

`moments::kurtosis()` donne γ_2, dont la référence est 3.

D'autres implémentations donnent l'**excès** $\gamma_2 - 3$, de référence 0. Sur les mêmes données, on lirait alors $1{,}84$.

Les deux sont justes ; il faut savoir laquelle on regarde.

Le bivarié

```r
T <- table(sal$diplome, sal$region)
addmargins(T)                          # avec les marges
round(prop.table(T, 1), 3)             # profils-lignes
round(prop.table(T, 2), 3)             # profils-colonnes
round(chisq.test(T)$expected, 1)       # effectifs théoriques
```

L'argument margin de prop.table

1 divise par les totaux de **ligne**, 2 par les totaux de **colonne**, rien du tout par le total général.

C'est exactement la distinction conjointe / profil-ligne / profil-colonne du chapitre 11 — et l'erreur d'interprétation la plus fréquente tient à ce seul argument.

Conditionnelles et décomposition de la variance

```r
# Moyennes et écarts-types conditionnels
aggregate(salaire ~ diplome, data = sal,
          FUN = function(v) c(n = length(v), moy = mean(v), et = sd(v)))

# Décomposition de la variance
g   <- split(sal$salaire, sal$diplome)
nj  <- sapply(g, length)
mj  <- sapply(g, mean)
vj  <- sapply(g, function(v) mean((v - mean(v))^2))
mg  <- mean(sal$salaire)

intra <- sum(nj*vj)/n                  # 5010426
inter <- sum(nj*(mj - mg)^2)/n         # 1739696
eta2  <- inter/(intra + inter)         # 0.258
```

Le contrôle à faire

`intra + inter` doit redonner la variance descriptive totale, $6\,750\,122$.

```r
y <- sal$salaire
z <- sal$anciennete

cov(z, y) * (n-1)/n            # 9859.9   covariance descriptive
cor(z, y)                      # 0.6363   coefficient de corrélation
cor(z, y)^2                    # 0.4049   coefficient de détermination

mod <- lm(salaire ~ anciennete, data = sal)
summary(mod)
```

La sortie de lm()

```
Coefficients:
             Estimate
(Intercept)  4347.2
anciennete    277.17

Multiple R-squared:  0.4049
```

Le graphique final

```r
plot(z, y, pch = 20, col = rgb(0,0,1,0.35),
     xlab = "ancienneté (années)", ylab = "salaire (dirhams)")
abline(mod, col = "orange", lwd = 2)
points(mean(z), mean(y), pch = 19, col = "darkviolet", cex = 1.4)
```

Le contrôle visuel

Le point moyen doit être **sur** la droite. S'il ne l'est pas, le calcul est faux — c'est la propriété du chapitre 14, et elle sert ici de test.

Le script complet

Quinze lignes

```r
sal <- read.csv("salaires.csv", stringsAsFactors = TRUE)
n   <- nrow(sal); x <- sal$salaire; z <- sal$anciennete

c(moyenne = mean(x), mediane = median(x),
  ecart_type = sd(x)*sqrt((n-1)/n), CV = sd(x)/mean(x))
quantile(x, c(.10,.25,.50,.75,.90))
c(asym = moments::skewness(x), aplat = moments::kurtosis(x))
gini(x)

table(sal$diplome, sal$region)
aggregate(salaire ~ diplome, sal, mean)
c(cov = cov(z,x)*(n-1)/n, r = cor(z,x), r2 = cor(z,x)^2)
coef(lm(salaire ~ anciennete, data = sal))
```

Ce que ce script produit

Tout le cours. Les douze indicateurs univariés, le tableau croisé, les conditionnelles, la corrélation et la droite.

Ce qu’il faut retenir

Le résumé du chapitre

Six points

1. `str()` d'abord : vérifier que chaque variable a le **bon type**.
2. `cut(..., right = FALSE)` pour des classes $[a\,;\,b[$ à la française.
3. `hist(..., freq = FALSE)` pour tracer des **densités**, pas des effectifs.
4. `var()` et `sd()` divisent par $n-1$: corriger par $(n-1)/n$ pour la variance descriptive.
5. `prop.table(T, 1)` et `prop.table(T, 2)` : profils-lignes et profils-colonnes, à ne pas confondre.
6. La syntaxe `y ~ x` est la clé de R : `boxplot`, `aggregate`, `lm` l'emploient toutes.

La suite

Le chapitre 16 refait exactement la même chaîne en Python, avec pandas et matplotlib — et retrouve les mêmes nombres.