

Statistique descriptive

Chapitre 14 : Mise en œuvre sur R

Jaouad Madkour

12 octobre 2026

Plan du chapitre

Refaire toute la chaîne	2
Le programme	2
Avant de commencer	3
Charger et explorer	3
L'import	3
Le premier réflexe : vérifier les types	3
Lire summary()	4
Le contrôle des manquants	4
Le tableau statistique	4
Variable qualitative	4
Variable continue : le découpage en classes	5
Le tableau de référence du chapitre 3	5
Le piège des bornes	5
Sur données groupées : moyenne et écart-type	6
Sur données groupées : mode et médiane	6
Les graphiques	7
Les quatre représentations	7
Ce que R affiche	7
L'argument qui compte : freq = FALSE	8
La boîte à moustaches	8
La syntaxe formule	8
Les indicateurs univariés	9
Tendance centrale et dispersion	9
Le piège n contre n − 1	9
La forme, sans package	10
La concentration	10
La courbe de Lorenz	10
La courbe obtenue	11
Le bivarié	11

Contingence	11
L'argument <code>margin</code> de <code>prop.table</code>	11
Conditionnelles	12
Décomposition de la variance	12
Covariance, corrélation, ajustement	12
La sortie de <code>lm()</code>	13
Le graphique final	13
Le nuage et sa droite	14
Le script complet	14
Une quinzaine de lignes	14
Ce que ce script produit	14
S'entraîner	15
Exercice 1 : n ou $n - 1$ sur dix salariés	15
Exercice 1 : solution	15
Exercice 2 : la région explique-t-elle le salaire ?	15
Exercice 2 : solution	16
Exercice 2 : le code et l'interprétation	16
Exercice 3 : cinq erreurs dans un script	16
Exercice 3 : solution	16
Ce qu'il faut retenir	17
Le résumé du chapitre	17

Refaire toute la chaîne

Le programme

Ce chapitre reprend les treize précédents

Étape	Chapitres
Importer et décrire les variables	1
Tableau statistique et classes	2, 3
Les quatre graphiques	4
Tendance centrale, quantiles, dispersion, forme	5 à 8
Lorenz et Gini	9
Contingence et conditionnelles	10, 11
Covariance, corrélation, ajustement	12, 13

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 qu'une quinzaine de lignes de code remplacent un semestre de calculs.

Avant de commencer

Ce qu'il faut installer, une seule fois

1. **R**, le langage, et **RStudio**, l'éditeur : tous deux gratuits.
2. Le fichier `salaires.csv` dans le **dossier de travail** de RStudio.
3. Un seul package supplémentaire, pour l'asymétrie et l'aplatissement :
`install.packages("moments")`.

Comprendre la formule reste indispensable

C'est ce qui permet de savoir **quelle** fonction demander, avec **quels** arguments — et de reconnaître un résultat aberrant. Les cinq pièges de ce chapitre sont tous des pièges de **convention**, invisibles pour qui ne connaît pas la formule.

Charger et explorer

L'import

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

La sortie de str()

```
'data.frame': 500 obs. of 6 variables:
 $ id      : int  1 2 3 4 5 ...
 $ salaire : int  8540 7100 4380 11790 ...
 $ anciennete: int  10 15 4 19 19 ...
 $ diplome  : Factor w/ 3 levels "Bac","Bac+2",...: 3 1 1 ...
 $ region   : Factor w/ 4 levels "Autres","Casablanca",...: 3 ...
 $ sexe     : Factor w/ 2 levels "F","H": 2 1 1 ...
```

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 dans un fichier réel.

```
class(sal$salaire)           # "integer" : correct
# Fichier français, avec « n.d. » pour les manquants :
sal <- read.csv2("salaires.csv", na.strings = c("", "n.d."))
```

read.csv ou read.csv2

`read.csv` attend des virgules entre colonnes et un point décimal ; `read.csv2` attend des **points-virgules** et une **virgule décimale** — le format d'un tableur réglé en français.

Lire summary()

```
summary(sal[, c("salaire", "anciennete", "diplome")])
```

La sortie

salaire	anciennete	diplome
Min. : 2810	Min. : 0.00	Bac :219
1st Qu.: 5318	1st Qu.: 6.00	Bac+2:174
Median : 6770	Median : 9.00	Bac+5:107
Mean : 7158	Mean :10.14	
3rd Qu.: 8330	3rd Qu.:13.00	
Max. :19050	Max. :35.00	

summary() arrondit à quatre chiffres significatifs

5318 est le $Q_1 = 5317,5$ du chapitre 6, 7158 la moyenne 7 158,26. Pour les valeurs exactes, il faut demander chaque indicateur.

Le contrôle des manquants

```
colSums(is.na(sal))      # 0 partout : aucune valeur manquante
nrow(sal)                # 500
length(unique(sal$salaire)) # 381 valeurs distinctes
```

Pourquoi 381 valeurs distinctes comptent

C'est le critère du chapitre 1 : 381 valeurs pour 500 individus, donc on traitera le salaire comme **continu** et on le regroupera en classes.

Un manquant contamine tout

Avec une seule valeur NA, `mean(x)` renvoie NA. L'argument `na.rm = TRUE` l'écarte — mais il faut alors **dire** sur combien d'individus porte le calcul.

Le tableau statistique

Variable qualitative

```
n <- nrow(sal)
ni <- table(sal$diplome)      # effectifs
fi <- prop.table(ni)          # fréquences
Ni <- cumsum(ni); Fi <- cumsum(fi) # cumuls
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 2.

Variable continue : le découpage en classes

```
bornes <- c(2000, 4000, 6000, 8000, 10000, 14000, 20000)
cl <- cut(sal$salaire, breaks = bornes, right = FALSE,
          dig.lab = 5)                                # classes [a ; b[
ni <- as.numeric(table(cl))                            # effectifs
ai <- diff(bornes)                                     # amplitudes
di <- ni / ai                                          # densités
print(data.frame(classe = levels(cl), ni, fi = ni/n, ai, di),
          digits = 3)
```

Les bornes, les amplitudes, les densités

`bornes` fixe les sept bornes des six classes ; `diff()` en déduit les amplitudes, et la densité divise chaque effectif par son amplitude — la règle du chapitre 3, quand les classes sont inégales.

Le tableau de référence du chapitre 3

La sortie de R

	classe	ni	fi	ai	di
1	[2000,4000)	31	0.062	2000	0.01550
2	[4000,6000)	157	0.314	2000	0.07850
3	[6000,8000)	166	0.332	2000	0.08300
4	[8000,10000)	79	0.158	2000	0.03950
5	[10000,14000)	57	0.114	4000	0.01425
6	[14000,20000)	10	0.020	6000	0.00167

Ce que la colonne di montre

La classe modale est [6 000 ; 8 000[, de densité 0,083. La cinquième classe compte presque deux fois plus de salariés que la première (57 contre 31), mais sa densité est **plus faible** (0,0143 contre 0,0155) : elle est deux fois plus large.

Le piège des bornes

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[` ; `dig.lab = 5` évite les bornes en notation scientifique, `[2e+03,4e+03)`.

Sur nos données, un salarié change de classe

Un salarié gagne **exactement** 6 000 dirhams. Avec $[a; b[$, il est dans la troisième classe ; sans `right = FALSE`, il passe dans la deuxième.

Classe	$[4\,000; 6\,000[$	$[6\,000; 8\,000[$
Avec <code>right = FALSE</code>	157	166
Sans (défaut de R)	158	165

Ici l'écart est minime. Sur des salaires arrondis à la centaine, des dizaines d'individus tombent sur les bornes.

Sur données groupées : moyenne et écart-type

```
ci <- (head(bornes, -1) + bornes[-1]) / 2 # centres des classes
mg <- sum(ni * ci) / n # 7210 moyenne groupée
vg <- sum(ni * ci^2) / n - mg^2 # 7685900 König-Huygens
sqrt(vg) # 2772.35 écart-type groupé
```

Les valeurs des chapitres 5 et 7, retrouvées

	Groupé	Exact	Écart
Moyenne	7 210	7 158,26	+0,7 %
Écart-type	2 772,35	2 598,10	+6,7 %

Les centres de classe ignorent la répartition **à l'intérieur** des classes : l'écart-type groupé surestime la dispersion, surtout à cause des deux dernières classes, très larges.

Sur données groupées : mode et médiane

```
k <- which.max(di) # classe modale : la 3e
d1 <- di[k] - di[k-1]; d2 <- di[k] - di[k+1]
bornes[k] + ai[k] * d1 / (d1 + d2) # 6187.5 mode
Fk <- cumsum(ni) / n
j <- which(Fk >= 0.5)[1] # classe médiane : la 3e
bornes[j] + ai[j] * (0.5 - Fk[j-1]) / (ni[j]/n) # 6747 médiane
```

Le code est la formule

Chaque ligne traduit une formule du chapitre 5 : la classe de **densité** maximale, les deux écarts Δ_1 et Δ_2 , puis l'interpolation linéaire dans la classe où F franchit 0,5. Écrire le code oblige à savoir **exactement** ce qu'on calcule.

Les graphiques

Les quatre représentations

```
par(mfrow = c(2, 2))          # quatre graphiques sur une page

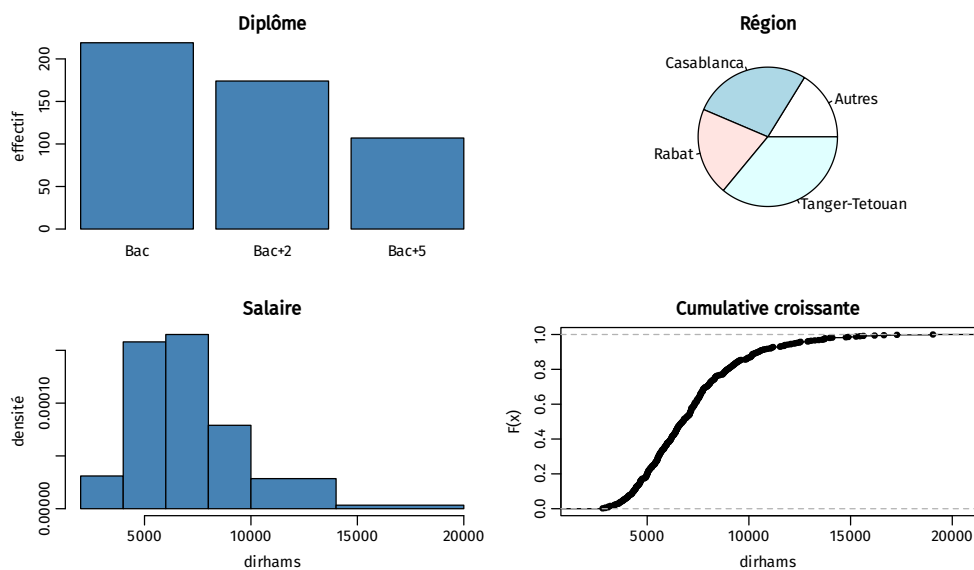
barplot(table(sal$diplome), col = "steelblue",
        main = "Diplôme", ylab = "effectif")

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

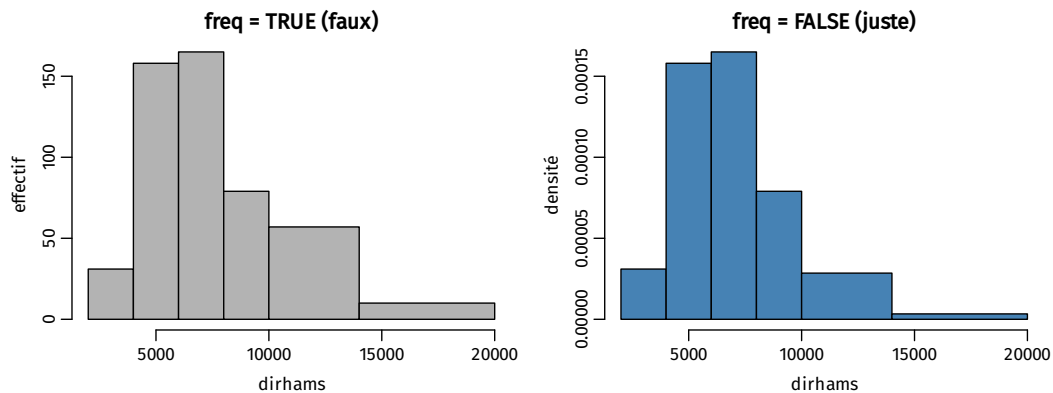
hist(sal$salaire, breaks = bornes, freq = FALSE, col = "steelblue",
     main = "Salaire", xlab = "dirhams", ylab = "densité")

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

Ce que R affiche



L'argument qui compte : `freq = FALSE`

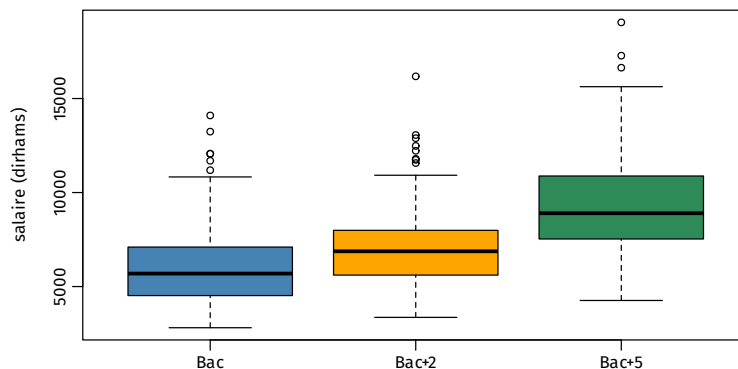


Deux histogrammes des mêmes données

Avec `freq = TRUE` (le défaut), `hist` trace les **effectifs** : la classe $[10\,000; 14\,000[$ paraît aussi fournie que $[8\,000; 10\,000[$. R prévient d'ailleurs : *the AREAS in the plot are wrong*. `freq = FALSE` trace les **densités**, la règle du chapitre 3.

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. Dans `boxplot`, elle produit directement les distributions **conditionnelles** du chapitre 11 : une boîte par diplôme.

```
aggregate(salaire ~ sexe, data = sal, FUN = median)
```

```
sexe salaire
1    F   6580
```

La même syntaxe, une autre fonction

`aggregate` applique FUN à chaque groupe : la médiane des hommes dépasse celle des femmes de 240 dirhams, alors que les **moyennes** sont égales à quelques centimes près (7 158,42 et 7 158,14). Deux indicateurs, deux lectures — le chapitre 11 l'avait annoncé.

Les indicateurs univariés

Tendance centrale et dispersion

```
x <- sal$salaire

mean(x)           # 7158.26
median(x)         # 6770
quantile(x, c(.10, .25, .50, .75, .90))
# 4348  5317.5  6770  8330  10473
IQR(x)            # 3012.5
var(x) * (n-1)/n   # 6750122  variance descriptive
sd(x) * sqrt((n-1)/n) # 2598.10  écart-type descriptif
sd(x) * sqrt((n-1)/n) / mean(x) # 0.363  CV
```

Les quantiles de R sont ceux du chapitre 6

Par défaut, `quantile()` place x_p au rang $1+p(n-1)$ et interpole : c'est la convention retenue au chapitre 6.

Le piège n contre $n - 1$

R divise par $n - 1$, pas par n

`var()` et `sd()` calculent la variance dite **corrigée**, celle de la statistique inférentielle. La variance **descriptive** du chapitre 7 divise par n :

$$V_R = \frac{1}{n-1} \sum (x_i - \bar{x})^2 = \frac{n}{n-1} V \quad \Rightarrow \quad V = \frac{n-1}{n} V_R$$

Sur nos 500 salariés

	var(), sd()	Descriptif	Rapport
Variance	6 763 649	6 750 122	0,998
Écart-type	2 600,70	2 598,10	0,999

Négligeable sur 500 observations ; sur 10, le rapport tombe à 0,9 (exercice 1). **Un écart avec un calcul manuel se vérifie d'abord ici.**

La forme, sans package

```
m <- mean(x)
mean((x - m)^3) / mean((x - m)^2)^1.5    # 1.164    asymétrie
mean((x - m)^4) / mean((x - m)^2)^2      # 4.837    aplatissement

library(moments)                          # après install.packages
skewness(x); kurtosis(x)                  # les mêmes valeurs
```

La formule du chapitre 8, en une ligne

$\gamma_1 = m_3/\sigma^3$ et $\gamma_2 = m_4/\sigma^4$, avec des moments centrés divisés par n : `mean()` fait exactement cela. Le package ne fait rien de plus.

Vérifier la convention d'aplatissement

`moments::kurtosis()` donne γ_2 , de référence 3. D'autres fonctions donnent l'excès $\gamma_2 - 3$, de référence 0 : on lirait 1,84. Les deux sont justes ; il faut savoir laquelle on regarde.

La concentration

```
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
tapply(x, sal$diplome, gini)           # Bac 0.181    Bac+2 0.155    Bac+5 0.165
```

La formule des rangs, vérifiée sur deux cas extrêmes

Avec les $x_{(i)}$ rangés par ordre croissant, $G = \frac{2 \sum i x_{(i)}}{n \sum x_i} - \frac{n+1}{n}$.

- **Égalité parfaite**, $x_{(i)} = c : \sum i c = c \frac{n(n+1)}{2}$, d'où $G = \frac{n+1}{n} - \frac{n+1}{n} = 0$.
- **Un seul individu a tout**, $x_{(n)} = S : G = \frac{2nS}{nS} - \frac{n+1}{n} = \frac{n-1}{n}$, proche de 1.

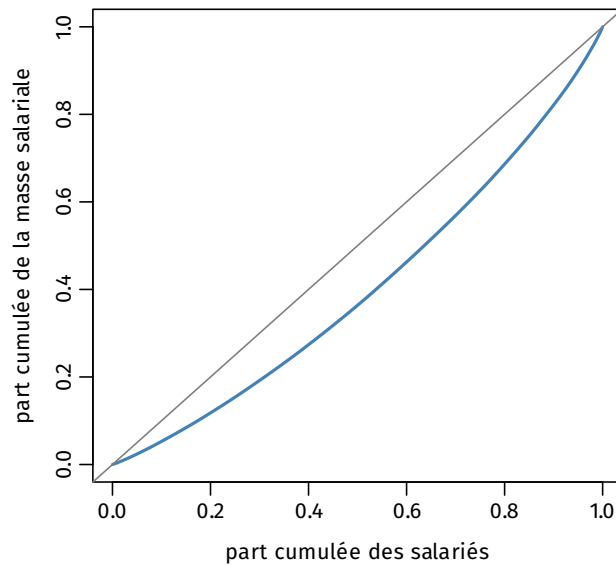
La courbe de Lorenz

```
v <- sort(x)
F <- seq_len(n) / n                # part cumulée des salariés
Q <- cumsum(v) / sum(v)            # part cumulée de la masse
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")      # la diagonale d'égalité
```

Le 0 ajouté en tête

`c(0, F)` et `c(0, Q)` font partir la courbe de l'origine : sans eux, elle commencerait au premier salarié, en $(1/500; x_{(1)}/S)$.

La courbe obtenue



Le bivarié

Contingence

```
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
```

La sortie de addmargins()

	Autres	Casablanca	Rabat	Tanger-Tetouan	Sum
Bac	40	59	44	76	219
Bac+2	30	46	35	63	174
Bac+5	11	32	23	41	107
Sum	81	137	102	180	500

L'argument margin de prop.table

1, 2 ou rien

1 divise par les totaux de **ligne**, 2 par les totaux de **colonne**, rien du tout par le total général. C'est la distinction conjointe / profil-ligne / profil-colonne du chapitre 10.

```
> round(prop.table(T, 2), 3)                 # profils-colonnes
      Autres Casablanca Rabat Tanger-Tetouan
Bac    0.494      0.431 0.431      0.422
Bac+2  0.370      0.336 0.343      0.350
Bac+5  0.136      0.234 0.225      0.228
```

Ce qu'on y lit

Les Bac+5 font 13,6 % des salariés de la région « Autres », contre 22 à 23 % ailleurs. Chaque colonne somme à 1 : on compare des **régions**, à effectifs différents.

Conditionnelles

```
# l'écart-type descriptif, en divisant par n
sdd <- function(v) sqrt(mean((v - mean(v))^2))
data.frame(n = tapply(x, sal$diplome, length),
           moy = tapply(x, sal$diplome, mean),
           et = tapply(x, sal$diplome, sdd))
```

Le résultat

	n	moy	et
Bac	219	6058.904	2021.222
Bac+2	174	7080.747	2045.281
Bac+5	107	9534.393	2872.122

Écrire sa propre fonction

sdd règle une fois pour toutes le piège $n - 1$: deux lignes de code, et tous les écarts-types du chapitre deviennent **descriptifs**.

Décomposition de la variance

```
g <- split(x, sal$diplome)
nj <- sapply(g, length)
mj <- sapply(g, mean)
vj <- sapply(g, function(v) mean((v - mean(v))^2))

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

Le contrôle à faire

$\text{intra} + \text{inter}$ doit redonner la variance descriptive totale, 6 750 122. Si ce n'est pas le cas, c'est presque toujours que vj a été calculé avec `var()`, donc en $n_j - 1$ (exercice 3).

Covariance, corrélation, ajustement

```
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)
```

cor() n'a pas besoin de correction

$r = \text{cov}/(\sigma_z \sigma_y)$: le facteur $\frac{n}{n-1}$ apparaît au numérateur et au dénominateur, et disparaît. Les **rapports** sont indifférents à la convention.

La sortie de lm()

Un extrait de summary(mod)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4347.23	177.18	24.54	<2e-16 ***
anciennete	277.17	15.06	18.41	<2e-16 ***

Residual standard error: 2008 on 498 degrees of freedom

Multiple R-squared: 0.4049, Adjusted R-squared: 0.4037

Ce qu'il faut lire, ce qu'on laisse pour plus tard

La colonne Estimate donne $\hat{a} = 4347$ et $\hat{b} = 277$; Multiple R-squared donne $r^2 = 0,405$: les valeurs du chapitre 13. Les autres colonnes relèvent de l'**inférence**. Même l'écart-type des résidus, 2008, divise par $n - 2$: en descriptif, $\sigma_y \sqrt{1 - r^2} = 2004$.

Le graphique final

```
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)
```

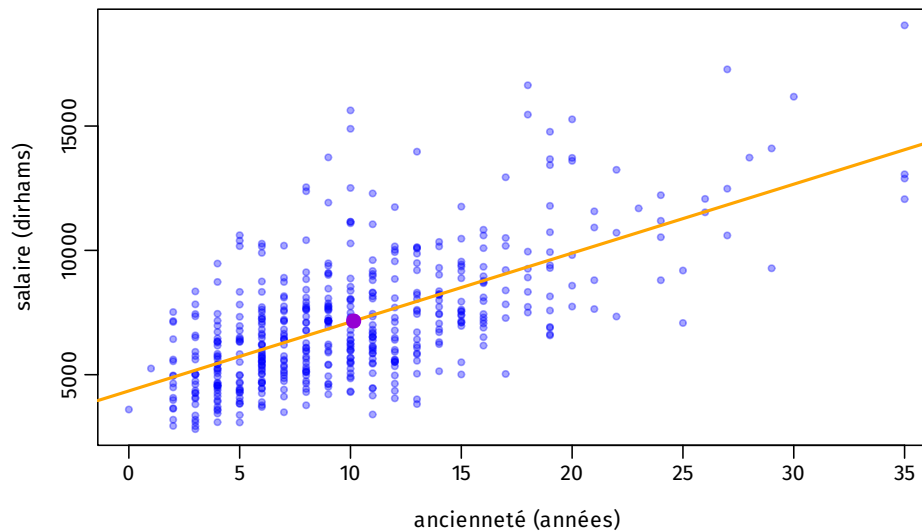
Le contrôle visuel

Le point moyen doit être **sur** la droite. S'il ne l'est pas, le calcul est faux — c'est la propriété du chapitre 13, et elle sert ici de test.

rgb(0, 0, 1, 0.35)

Du bleu à 35 % d'opacité : les points superposés apparaissent plus foncés. Avec 500 salariés et des anciennetés entières, c'est ce qui rend la **densité** du nuage visible.

Le nuage et sa droite



Le script complet

Une quinzaine de lignes

```
sal <- read.csv("salaires.csv", stringsAsFactors = TRUE)
n <- nrow(sal); x <- sal$salaire; z <- sal$anciennete
sdd <- function(v) sqrt(mean((v - mean(v))^2))
gini <- function(v) { v <- sort(v); k <- seq_along(v)
  2 * sum(k * v) / (length(v) * sum(v)) - (length(v) + 1) / length(v) }

c(moyenne = mean(x), mediane = median(x), ecart_type = sdd(x),
  CV = sdd(x) / mean(x), gini = gini(x))
quantile(x, c(.10, .25, .50, .75, .90))
c(asym = moments::skewness(x), aplat = moments::kurtosis(x))
table(sal$diplome, sal$region)
tapply(x, sal$diplome, 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 indicateurs univariés, le tableau croisé, les conditionnelles, la corrélation et la droite. Ce qui a demandé treize chapitres tient en une quinzaine de lignes — et c'est bien ainsi. **Le semestre n'a jamais servi à apprendre à calculer, mais à savoir ce qu'on calcule et comment le lire.**

Un script se relit et se rejoue

Un calcul fait au tableur se perd dans les cellules ; un script se **relit**, se **corrige** et se **rejoue** sur les données de l'année suivante en une seconde. C'est la vraie raison de passer par le code.

S'entraîner

Exercice 1 : n ou n - 1 sur dix salariés

Les dix salariés des chapitres 12 et 13

```
x <- c(2, 3, 4, 5, 6, 8, 10, 12, 15, 20) # années
y <- c(4.5, 5, 6.5, 5.5, 7.5, 6, 8.5, 7.5, 9, 11) # milliers
var(x) # 33.38889
cov(x, y) # 10.77778
cor(x, y) # 0.9275
```

1. Retrouver la variance descriptive $V(x)$ et la covariance descriptive.
2. Calculer la pente $\hat{b}$ de deux façons : avec les sorties de R, puis avec les valeurs descriptives. Conclure.
3. Pourquoi `cor()` ne demande-t-elle aucune correction ?

Exercice 1 : solution

Le facteur $(n - 1)/n = 0,9$

1. $V(x) = 0,9 \times 33,389 = 30,05$, soit $\sigma_x = 5,48$ ans ; $cov = 0,9 \times 10,778 = 9,70$: les valeurs du chapitre 12.
2. Avec R : $\hat{b} = 10,778/33,389 = 0,323$. En descriptif : $9,70/30,05 = 0,323$. **La même pente** : le facteur 0,9 se simplifie.
3. $r = cov/(\sigma_x \sigma_y)$ est aussi un rapport : le facteur se simplifie. Seuls les **niveaux** (variance, écart-type, covariance) dépendent de la convention ; les rapports (r , $\hat{b}$, CV , η^2) n'en dépendent pas, à condition d'employer la **même** convention en haut et en bas.

Exercice 2 : la région explique-t-elle le salaire ?

Une sortie de R à exploiter

```
> tapply(x, sal$region, mean)
      Autres      Casablanca      Rabat Tanger-Tetouan
6272.346      7887.445      7504.314      6805.833
> table(sal$region)
      Autres      Casablanca      Rabat Tanger-Tetouan
      81          137          102          180
```

1. Calculer la variance **inter-régions** (moyenne générale 7 158,26).
2. En déduire η^2 , sachant que $V = 6 750 122$. Comparer au diplôme (0,258).
3. Écrire en R une fonction `eta2(y, g)` qui fasse ce calcul.
4. Casablanca compte 23,4 % de Bac+5, « Autres » 13,6 %. Qu'en conclure ?

Exercice 2 : solution

Variance inter et rapport de corrélation

1. Les écarts à la moyenne générale valent $-885,9$; $729,2$; $346,1$; $-352,4$, d'où

$$81 \times 885,9^2 + 137 \times 729,2^2 + 102 \times 346,1^2 + 180 \times 352,4^2 \approx 170,99 \text{ millions}$$

$$V_{\text{inter}} \approx 170,99 \times 10^6 / 500 \approx 341\,980$$

2. $\eta^2 = 341\,980 / 6\,750\,122 = 0,051$: la région explique 5 % de la dispersion, cinq fois moins que le diplôme.

Exercice 2 : le code et l'interprétation

Le code et l'effet de structure

3. Le rapport de la variance inter à la variance totale :

```
eta2 <- function(y, g) {  
  m <- mean(y)  
  inter <- sum(tapply(y, g, function(v) length(v) * (mean(v) - m)^2))  
  inter / sum((y - m)^2)  
}  
eta2(x, sal$region)      # 0.0507
```

4. Une partie de l'avance de Casablanca vient de sa **composition** : plus de diplômés. C'est l'effet de structure du chapitre 11 ; comparer les régions à **diplôme égal** dirait si un écart subsiste.

Exercice 3 : cinq erreurs dans un script

Chaque ligne s'exécute sans message d'erreur

```
cl <- cut(sal$salaire, breaks = bornes)           # (a)  
hist(sal$salaire, breaks = bornes)               # (b)  
sigma <- sd(sal$salaire)                         # (c)  
prof <- prop.table(table(sal$diplome, sal$region)) # (d)  
vj <- tapply(sal$salaire, sal$diplome, var)       # (e)  
intra <- sum(nj * vj) / n
```

Pour chaque ligne : trouver l'erreur, la corriger, et dire ce qu'elle change sur nos données. Pour (d), on voulait les **profils-lignes**.

Exercice 3 : solution

(a), (b), (c)

- (a) Classes $[a; b]$: ajouter `right = FALSE`. Le salarié à 6 000 dh passe de la 3^e à la 2^e classe (157/166 devient 158/165).
- (b) Histogramme en **effectifs** : ajouter `freq = FALSE`. R affiche un avertissement, mais trace quand même le graphique faux.
- (c) Écart-type corrigé, 2 600,70 : écrire `sd(...) * sqrt((n-1)/n)`, ou `sdd()`, pour

2 598,10.

(d), (e)

- (d) Fréquences **conjointes** : le tableau entier somme à 1. Écrire `prop.table(..., 1)` pour que chaque ligne somme à 1.
- (e) Variances en $n_j - 1$: la variance intra devient 5 043 703 au lieu de 5 010 426, et `intra + inter` dépasse la variance totale de 33 277. Le contrôle de la décomposition détecte l'erreur.

Ce qu'il faut retenir

Le résumé du chapitre

Sept 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()`, `sd()`, `cov()` divisent par $n - 1$: multiplier par $(n - 1)/n$, ou écrire sa fonction.
5. `prop.table(T, 1)` et `prop.table(T, 2)` : profils-lignes et profils-colonnes.
6. La syntaxe `y ~ x` est la clé de R : `boxplot`, `aggregate`, `lm` l'emploient toutes.
7. Dans `summary(lm())`, seuls `Estimate` et `Multiple R-squared` relèvent de ce cours.

La suite

Le chapitre 15 refait exactement la même chaîne en Python, avec `pandas` et `matplotlib` — et retrouve les mêmes nombres.