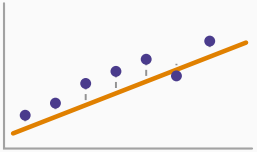
Introduction à l'économétrie

Chapitre 22 : Mise en œuvre sur R et Python



Jaouad Madkour

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

Plan du chapitre

Lire une sortie

Neuf nombres, et vous savez tous les lire

	Estimate	Std. Error	t value	Pr(> |t|)
(Intercept)	3320,42	158,85	20,90	< 2e-16
ancienneté	270,27	11,72	23,06	< 2e-16
diplomeBac+2	1099,57	158,59	6,93	1,3e-11
diplomeBac+5	3336,99	184,25	18,11	< 2e-16
Residual standard error: 1561 on 496 degrees of freedom				
Multiple R-squared: 0,6418 Adjusted: 0,6396				
F-statistic: 296,2 on 3 and 496 DF, p-value: < 2,2e-16				

Neuf nombres, et vous savez tous les lire

Estimate — les $\widehat{\beta}_j$, chapitre 9.

Std. Error — $\widehat{\sigma}(\widehat{\beta}_j)$, chapitre 6.

t value — le rapport des deux, chapitre 6.

Pr — la p-value, chapitre 15 de l'inférence.

Residual standard error — $\widehat{\sigma}_\varepsilon$, l'écart-type des résidus.

degrees of freedom — $n - k - 1$, un degré perdu par paramètre estimé, chapitre 4 de l'inférence.

R-squared — SCE/SCT, chapitre 5.

F-statistic — le test global, chapitre 10.

Ce que la sortie ne dit pas

Rien sur l'hétéroscédasticité, rien sur les variables

C'est l'aboutissement de quatre cours.

La moyenne et la variance viennent de la descriptive ; la loi de Student et le théorème central limite, des probabilités ; l'erreur-type, le test et la p-value, de l'inférence ; la décomposition et le F, de l'analyse de la variance. **L'économétrie n'a rien ajouté d'autre qu'une question économique.**

Sur R

Estimer

```r
sal <- read.csv("salaires.csv", stringsAsFactors = TRUE)
sal$diplome <- relevel(sal$diplome, ref = "Bac")   # fixer la reference

m1 <- lm(salaire ~ anciennete, data = sal)
m2 <- lm(salaire ~ anciennete + diplome, data = sal)
m4 <- lm(salaire ~ anciennete + diplome + sexe + region, data = sal)

summary(m2)
confint(m2)                          # 247.65   293.19   pour l'anciennete
anova(m1, m2)                        # Fisher partiel : F = 164.0
```

Une ligne qui évite bien des contresens

`relevel()` fixe la modalité de référence. Sans elle, R prend la première par ordre alphabétique — ici « Bac », par chance.

Sur d'autres données, l'ordre alphabétique donne une référence absurde. **Vérifiez toujours laquelle a été retenue.**

Diagnostiquer

```r
par(mfrow = c(2, 2)) ; plot(m2)    # les quatre graphiques d'un coup

library(lmtest) ; library(car) ; library(sandwich)

bptest(m2)                         # Breusch-Pagan : BP = 57.60, p = 1.9e-12
vif(m4)                            # tous inferieurs a 2.1
shapiro.test(residuals(m2))        # normalite rejetee
cooks.distance(m2) > 4/nrow(sal)   # les points a examiner
```

La commande la plus rentable du chapitre

`plot(m2)` produit les quatre graphiques du chapitre 13 en une instruction : résidus contre valeurs ajustées, quantile-quantile, échelle-position, résidus contre levier.

Quatre secondes, et l'essentiel du diagnostic est fait.

Corriger

```r
# Ecarts-types robustes a l'heteroscedasticite (chapitre 15)
coeftest(m2, vcov = vcovHC(m2, type = "HC1"))

#                Estimate  Std. Error  t value
# (Intercept)     3320.42     144.37    23.00
# anciennete       270.27      13.00    20.78
# diplomeBac+2    1099.57     138.43     7.94
# diplomeBac+5    3336.99     226.28    14.75
```

Les coefficients n'ont pas bougé

Comparez à la sortie de `summary(m2)` : mêmes estimations au centime près, écarts-types différents.

C'est exactement ce que la théorie annonçait — et c'est la vérification la plus convaincante du chapitre 15.

L'enquête de ménages

```r
men <- read.csv("menages.csv")
mk  <- lm(consommation ~ revenu, data = men)
mk2 <- lm(consommation ~ revenu + taille + milieu, data = men)

summary(mk)              # b1 = 0.6960, t = 47.3, R2 = 0.8490

# Tester l'hypothese de Keynes : beta1 = 1
linearHypothesis(mk, "revenu = 1")     # F = 426.9 , soit t = -20.66
```

L'enquête de ménages (suite)

```
# Ecarts-types de grappe, au niveau du quartier (chapitre 16)
coeftest(mk2, vcov = vcovCL(mk2, cluster = ~ quartier))
#                Estimate  Std. Error  t value
# revenu           0.7045      0.0232    30.35
# taille          70.6689     12.2912     5.75
# milieuUrbain   -99.4397    101.9173    -0.98
```

Deux lignes concluent deux chapitres

`linearHypothesis` teste n'importe quelle contrainte linéaire — ici la restriction que la théorie de Keynes impose. **La démarche des quatre étapes s'achève sur une commande.**

`vcovCL` corrige le regroupement. Comparez la ligne `milieuUrbain` à la sortie ordinaire : l'écart-type passe de $55{,}69$ à $101{,}92$, parce que cette variable est **constante dans le quartier**.

Sur Python

Les mêmes opérations

```
import pandas as pd, statsmodels.formula.api as smf
import statsmodels.stats.api as sms

sal = pd.read_csv("salaires.csv")
m2 = smf.ols("salaire ~ anciennete + C(diplome, Treatment('Bac'))",
             data=sal).fit()
print(m2.summary())
m2.conf_int()
```

Les mêmes opérations (suite)

```
# Heteroscedasticite
sms.het_breuschpagan(m2.resid, m2.model.exog)     # (57.60, 1.9e-12, ...)
m2r = m2.get_robustcov_results(cov_type="HC1")    # ecarts-types robustes
```

La correspondance des commandes

R	Python
`lm`	`smf.ols(...).fit()`
`relevel`	`C(var, Treatment('ref'))`
`anova(m1, m2)`	`m2.compare_f_test(m1)`
`vcovHC`	`get_robustcov_results(cov_type="HC1")`
`vcovCL`	`cov_type="cluster", cov_kwds={...}`
`vif`	`variance_inflation_factor`

La régression finale de Mincer

```
import numpy as np
sal["lsal"] = np.log(sal.salaire)

mincer = smf.ols("lsal ~ anciennete + C(diplome, Treatment('Bac'))"
                 " + C(region, Treatment('Autres'))", data=sal).fit()

# anciennete        0.0345   ->  +3.5 %
# Bac+5             0.4353   ->  +54.5 %   (exp(0.4353) - 1)
# R-squared         0.654
100 * (np.exp(mincer.params) - 1)      # les effets exacts en pourcentage
```

La dernière ligne mérite d'être écrite à chaque fois

Elle convertit tous les coefficients en pourcentages **exacts**, sans l'approximation 100β.

Sur le Bac+5, la différence est de onze points : $43{,}5$ contre $54{,}5$. **C'est une erreur d'interprétation très répandue, et une ligne de code la supprime.**

Ce que le logiciel ne fait pas

Trois choses, et ce sont les trois qui comptent

Il ne choisit pas les variables

La spécification vient de la théorie économique. Aucune procédure automatique ne remplace la question : *quelles variables devraient figurer dans ce modèle, et lesquelles manquent ?*

Il ne signale pas les variables omises, ni l'endogénéité

La sortie reste parfaitement propre. Le R^2 est honorable, les t écrasants, les résidus impeccables — **et le coefficient peut être faux**.

C'est le chapitre 17 et le chapitre 20 réunis, et c'est ce qui distingue un économètre d'un utilisateur de logiciel.

Ce qu’il faut retenir

Le résumé du chapitre

Six points

1. `lm` et `smf.ols` estiment ; tout le reste de la sortie a été calculé à la main.
2. `relevel` ou `Treatment` fixent la **référence** des indicatrices : à vérifier systématiquement.
3. `plot(m)` donne les **quatre graphiques** de diagnostic en une instruction.
4. `vcovHC` et `vcovCL` corrigent les écarts-types sans toucher aux coefficients.
5. `linearHypothesis` teste une contrainte **économique** — c'est l'étape 4 de la démarche.
6. Le logiciel ne choisit pas les variables et ne signale jamais l'**endogénéité**.

La fin du parcours

Quatre cours, un seul fichier de cinq cents salariés. On l'a décrit, on en a fait un modèle de hasard, on en a tiré des échantillons pour estimer et tester, et l'on a fini par expliquer le salaire par l'ancienneté, le diplôme et la région.

Aucun de ces quatre cours n'a introduit d'outil que le précédent n'annonçait pas. **C'est une seule discipline, enseignée en quatre fois** — et ce que vous savez maintenant lire, un tableau de régression, en est l'aboutissement.