

Économétrie

Séance 2 — Probabilités, hypothèses et propriétés des MCO

Licence 3 — Économie

Anderson VIL

vladerinson@gmail.com

THEMA

CY Cergy Paris Université

Année universitaire 2026–2027

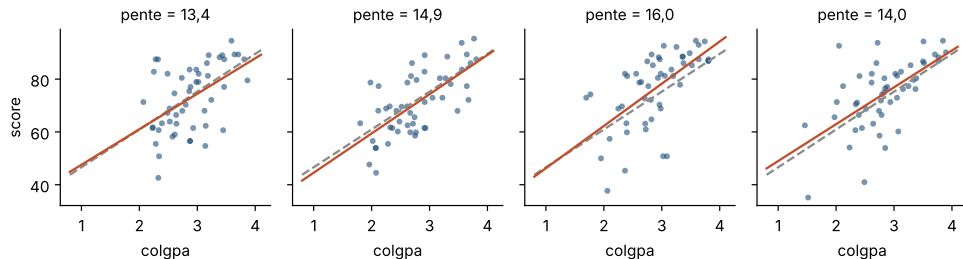
Au programme

1. D'un échantillon à l'autre
2. Rappels de probabilités
3. Les hypothèses du modèle de régression simple
4. L'absence de biais des MCO
5. La variance des estimateurs des MCO
6. Estimer σ^2 et l'erreur-type

La séance 1 a montré comment calculer $\hat{\beta}_0$ et $\hat{\beta}_1$. Celle-ci se demande si $\hat{\beta}_1$ est proche de β_1 , et à quelles conditions.

1. D'un échantillon à l'autre

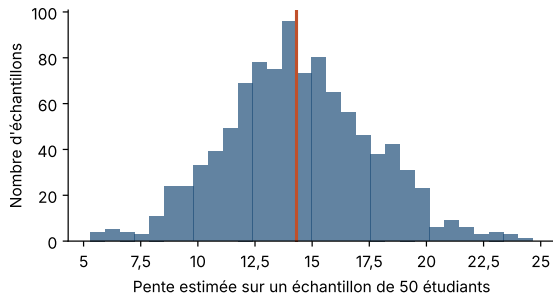
Un autre échantillon, une autre droite



On traite les 856 étudiants de ECONMATH comme la population. On y tire au hasard, avec remise¹, quatre échantillons de 50 étudiants. Chacun donne sa droite des MCO (en orange). La droite des 856, en pointillé, a pour pente 14,32.

Les quatre pentes valent 13,4, 14,9, 16,0 et 14,0. Aucune n'est « la bonne ». Toutes sont proches de 14,3.

La distribution d'échantillonnage



1 000 échantillons de 50 étudiants. Moyenne des pentes 14,38, écart-type 3,19.

Avant le tirage, la pente qu'on obtiendra est inconnue. C'est une *variable aléatoire*.

Un *estimateur* (*estimator*) est une règle de calcul, comme la formule des MCO. Appliquée à un échantillon précis, elle donne une *estimation* (*estimate*), un nombre.

La loi de l'estimateur sur tous les échantillons possibles est sa **distribution d'échantillonnage** (*sampling distribution*).

Trois questions pour la séance

- ▶ Où la distribution de $\hat{\beta}_1$ est-elle centrée ? Sur β_1 , ou ailleurs ?
- ▶ Quelle est sa dispersion ? Qu'est-ce qui la réduit ?
- ▶ Comment mesurer cette dispersion quand on ne dispose que d'un échantillon ?

Les réponses dépendent de la façon dont les données ont été produites. Pour le dire, il faut le langage des probabilités, puis des hypothèses sur le modèle.

2. Rappels de probabilités

Variable aléatoire

L'expérience. Tirer au hasard un étudiant parmi les 856. Avant le tirage, sa note est inconnue.

Une *variable aléatoire* (*random variable*) est une grandeur numérique qui dépend du résultat d'une expérience aléatoire. On la note par une majuscule, Y , et une valeur observée par une minuscule, y .

Sa *loi* (*distribution*) donne ses valeurs possibles et leurs probabilités.

X et Y sont *discrètes*, à nombre fini de valeurs. Une variable *continue* prend chaque valeur d'un intervalle avec une probabilité nulle².

Deux exemples

- ▶ $X = 1$ si l'étudiant tiré travaille 15 heures ou plus par semaine, 0 sinon (*variable de Bernoulli*)

$$P(X = 1) = \frac{241}{856} = 0,2815$$

- ▶ Y sa note finale. Chacune des 856 notes a la probabilité $1/856$

Espérance

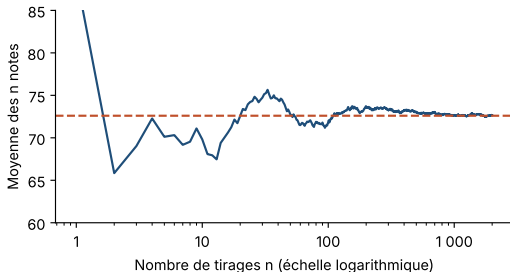
Moyenne des valeurs possibles, pondérée par leurs probabilités

$$E(X) = \sum_j x_j P(X = x_j).$$

$$E(X) = 0 \times 0,7185 + 1 \times 0,2815 = 0,2815.$$

$E(Y) = \sum_i \frac{1}{856} y_i = 72,60$, la moyenne des 856 notes.

$\mu = E(Y)$ est la moyenne de la population. La moyenne empirique $\bar{y}$ s'en rapproche quand n grandit (*loi des grands nombres*³).



Moyenne des n premières notes tirées au hasard. 69,81 après 10 tirages, 71,85 après 100, 72,66 après 2 000.

► Code

Calculer avec l'espérance

Pour des constantes $a, b, c, a_1, \dots, a_n$

$$E(c) = c \quad (\text{E.1})$$

$$E(aX + b) = aE(X) + b \quad (\text{E.2})$$

$$E\left(\sum_{i=1}^n a_i X_i\right) = \sum_{i=1}^n a_i E(X_i) \quad (\text{E.3})$$

Exemple. La note sur 20 vaut 0,2 Y , donc
 $E(0,2 Y) = 0,2 \times 72,60 = 14,52$. Et
 $E(Y - \mu) = \mu - \mu = 0$.

Piège. L'espérance ne passe pas à travers une fonction non linéaire. $E(Y^2) = 5\,450,1$ alors que $[E(Y)]^2 = 5\,270,7$.

$$\text{Var}(X) = E[(X - \mu)^2] = E(X^2) - \mu^2, \quad \text{sd}(X) = \sqrt{\text{Var}(X)}$$

La seconde écriture s'obtient en développant le carré, puis avec E.2 et E.3.

Changer d'échelle

- L'écart quadratique moyen à l'espérance, comme s_x^2 l'est à $\bar{x}$

$$\text{Var}(aX + b) = a^2 \text{Var}(X)$$

- Bernoulli, $X^2 = X$, donc
 $\text{Var}(X) = p - p^2 = 0,2023$

$$\text{car } aX + b - E(aX + b) = a(X - \mu).$$

- Notes, $\text{Var}(Y) = 179,37$ et $\text{sd}(Y) = 13,39$

Ici on divise par $N = 856$, car toute la population est connue, μ comprise. Le diviseur $n - 1$ de s_y^2 sert quand μ doit être estimée (séance 1).

$$\text{Cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] = E(XY) - \mu_X \mu_Y, \quad \text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\text{sd}(X) \text{sd}(Y)}$$

- ▶ Ce sont les grandeurs de la population dont s_{xy} et r_{xy} sont les versions empiriques. Mêmes propriétés. Le signe se lit sur les quadrants, $\text{Corr} \in [-1, 1]$, la corrélation ne dépend pas des unités
- ▶ Travail salarié et note. $\text{Cov}(X, Y) = -0,967$ et $\text{Corr}(X, Y) = -0,16$. Ceux qui travaillent 15 heures ou plus ont en moyenne des notes plus basses

Indépendance et covariance nulle

X et Y sont *indépendantes* quand, pour toutes valeurs x et y ,

$$P(X = x, Y = y) = P(X = x) P(Y = y).$$

Connaître X ne change rien aux probabilités de Y .

Indépendantes $\Rightarrow$ covariance nulle, car alors

$$E(XY) = \sum_x \sum_y xy P(X = x)P(Y = y) = E(X)E(Y).$$

La réciproque est fausse. X vaut -1 , 0 ou 1 avec probabilité $1/3$, et $Y = X^2$.

► $E(X) = 0$ et $E(XY) = E(X^3) = 0$,
donc $\text{Cov}(X, Y) = 0$

► Pourtant Y dépend de X .
 $P(X = 0, Y = 1) = 0$, mais
 $P(X = 0) P(Y = 1) = \frac{2}{9}$

C'est le nuage en U de la séance 1, dans la population.

La variance d'une somme

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{Cov}(X, Y)$$

On développe $[(X - \mu_X) + (Y - \mu_Y)]^2$ et on prend l'espérance. Si les X_i sont deux à deux non corrélées, en particulier indépendantes, ces termes disparaissent

$$\text{Var}\left(\sum_{i=1}^n a_i X_i\right) = \sum_{i=1}^n a_i^2 \text{Var}(X_i).$$

La moyenne. $X_1, \dots, X_n$ indépendantes de variance σ^2 , et $a_i = 1/n$,

$$\text{Var}(\bar{X}) = n \times \frac{\sigma^2}{n^2} = \frac{\sigma^2}{n}.$$

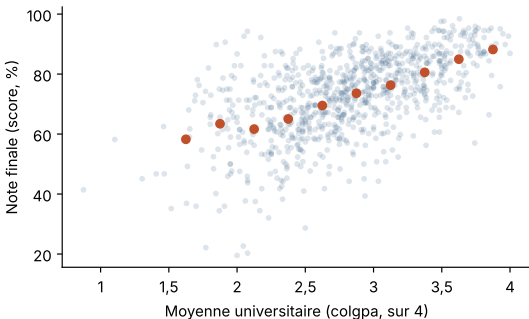
Notes. Pour 50 notes tirées au hasard, $\text{sd}(\bar{Y}) = 13,39/\sqrt{50} = 1,89$. Simulé, 1,90.

Espérance conditionnelle

$E(Y | X = x)$ est l'espérance de Y parmi les unités pour lesquelles $X = x$. C'est une fonction de x .

Travail salarié. $E(Y | X = 0) = 73,95$ et $E(Y | X = 1) = 69,17$.

Moyenne universitaire. En orange, la note moyenne par classe de `colgpa` de largeur 0,25. Elle croît presque en ligne droite.



Quand $E(y | x) = \beta_0 + \beta_1 x$, on parle de **droite de régression de la population** (*population regression function*). Elle décrit la note *moyenne* à chaque niveau de x , pas la note de chacun.

[► Code](#)

Calculer avec l'espérance conditionnelle

CE.2 Une fonction de X se comporte comme une constante.

$$E[a(X)Y + b(X) \mid X] = a(X)E(Y \mid X) + b(X)$$

CE.3 Si X et Y sont indépendantes, $E(Y \mid X) = E(Y)$

CE.4 **Espérances itérées**⁴. $E[E(Y \mid X)] = E(Y)$

Exemple de CE.4. On pondère chaque moyenne conditionnelle par la part de son groupe

$$0,7185 \times 73,95 + 0,2815 \times 69,17 = 72,60 = E(Y).$$

La moyenne simple des deux, 71,56, serait fausse, car les deux groupes n'ont pas la même taille.

De l'espérance conditionnelle à la covariance

Résultat

Si $E(u | x) = 0$, alors $E(u) = 0$ et $\text{Cov}(x, u) = 0$. Plus généralement, toute fonction de x est non corrélée à u .

Preuve. Par les espérances itérées, $E(u) = E[E(u | x)] = E(0) = 0$. Puis, avec CE.2,

$$\text{Cov}(x, u) = E(xu) - E(x)E(u) = E[E(xu | x)] = E[x E(u | x)] = 0.$$

On remplace x par $g(x)$ pour une fonction quelconque.

La réciproque est fausse. Si x vaut $-1, 0, 1$ avec probabilité $1/3$ et $u = x^2 - 2/3$, alors $\text{Cov}(x, u) = 0$ mais $E(u | x) = x^2 - 2/3$ dépend de x .

$E(u | x) = 0$ est donc plus exigeante que $\text{Cov}(x, u) = 0$. C'est elle que nous supposons.

$$\text{Var}(Y \mid X = x) = E[(Y - E(Y \mid x))^2 \mid X = x]$$

la dispersion de Y parmi les unités telles que $X = x$.

Elle peut varier avec x , indépendamment de $E(Y \mid x)$. Dans ECONMATH, la note moyenne augmente avec `colgpa` tandis que la dispersion diminue.

<code>colgpa</code>	Moyenne	Écart-type
0,88 à 2,45	63,05	14,58
2,45 à 2,81	69,07	10,08
2,81 à 3,21	75,53	10,49
3,21 à 4,00	82,78	8,81

Note finale par quart de `colgpa`, 214 étudiants environ par quart.

3. Les hypothèses du modèle de régression simple

Pourquoi des hypothèses ?

La formule des MCO donne toujours un nombre. Dire qu'il est proche de β_1 suppose de savoir comment les données ont été produites. Cinq hypothèses le décrivent.

	Nom	Énoncé
RLS.1	Linéarité dans les paramètres	$y = \beta_0 + \beta_1 x + u$ dans la population
RLS.2	Échantillonnage aléatoire	les (x_i, y_i) sont tirés indépendamment
RLS.3	Variation de x	les x_i ne sont pas tous égaux
RLS.4	Espérance conditionnelle nulle	$E(u \mid x) = 0$
RLS.5	Homoscédasticité	$\text{Var}(u \mid x) = \sigma^2$

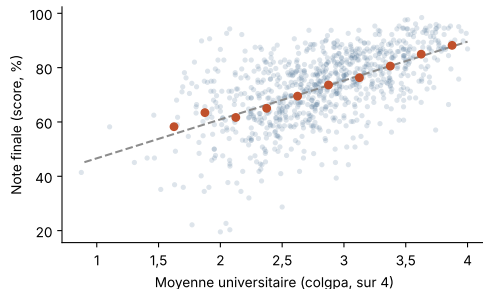
RLS pour *régression linéaire simple*.

RLS.1, linéarité dans les paramètres

Ce qu'elle dit. Dans la population, $y = \beta_0 + \beta_1 x + u$, ce qui définit β_1 . y et x peuvent être des transformations (logarithme, carré), vues plus tard.

Si elle échoue. Si l'effet de x change avec son niveau, la droite ne le résume qu'en moyenne.

L'examiner. Comparer les moyennes de y par classe de x (en orange) à la droite des MCO (en pointillé). Au-delà de $\text{colgpa} = 2$, l'écart reste sous 1,3 point. En dessous (55 étudiants), la droite sous-estime.



RLS.2, échantillonnage aléatoire

Ce qu'elle dit. Les (x_i, y_i) sont tirés indépendamment dans la même population. Chacun vérifie $y_i = \beta_0 + \beta_1 x_i + u_i$, et les u_i sont indépendants entre eux.

À quoi elle sert. Passer de la population à l'échantillon. u_i ne dépend pas des x_j des autres observations, et $\text{Cov}(u_i, u_j) = 0$.

Si elle échoue. Séries temporelles, où une année ressemble à la précédente. Échantillon sélectionné, par exemple les seuls étudiants présents à l'examen.

L'examiner. Les données ne le disent pas. Il faut savoir comment elles ont été recueillies. ECONMATH réunit les étudiants d'un cours d'une université. La population est celle des étudiants de ce cours.

L'échantillonnage aléatoire dit comment on *choisit les unités observées*.

RLS.3, variation de x dans l'échantillon

Ce qu'elle dit. Les x_i de l'échantillon ne sont pas tous égaux, soit $\sum_i (x_i - \bar{x})^2 > 0$.

À quoi elle sert. Sans elle, la formule des MCO divise par zéro. C'est la seule hypothèse nécessaire pour *calculer* $\hat{\beta}_0$ et $\hat{\beta}_1$.

Si elle échoue. Tous les points sont sur une verticale. Les données ne disent rien de la pente (séance 1).

L'examiner. Un écart-type de x strictement positif suffit. Dans ECONMATH, $s_x = 0,537$ et `colgpa` prend 612 valeurs différentes.

On note désormais

$$SCT_x = \sum_{i=1}^n (x_i - \bar{x})^2,$$

la variation totale de x dans l'échantillon. RLS.3 s'écrit $SCT_x > 0$.

RLS.4, espérance conditionnelle nulle

RLS.4

Pour toute valeur de x , $E(u \mid x) = 0$.

- ▶ Deux énoncés en un. $E(u \mid x) = E(u)$, la moyenne des inobservés ne varie pas avec x (*indépendance en moyenne*). Et $E(u) = 0$, une normalisation permise par la constante⁵
- ▶ Avec RLS.1 et CE.2, $E(y \mid x) = \beta_0 + \beta_1 x$. La droite de la population est l'espérance conditionnelle
- ▶ Avec RLS.2, $E(u_i \mid x_1, \dots, x_n) = E(u_i \mid x_i) = 0$

Dans le fil rouge. u contient la motivation, le niveau en mathématiques, le travail salarié. Leur moyenne est-elle la même quand `colgpa` vaut 2 et quand il vaut 3 ? C'est douteux.

β_1 est l'effet de x , les facteurs de u restant fixes. Or les données comparent des unités qui diffèrent par x et par u .

Sous RLS.4, u a la même moyenne à chaque niveau de x , donc

$$E(y \mid x + 1) - E(y \mid x) = \beta_1.$$

Comparer des moyennes, c'est comparer à autres facteurs égaux **en moyenne**.

Le tirage au sort rend x indépendant de u , donc $E(u \mid x) = E(u) = 0$ par CE.3. Sans lui, RLS.4 est une hypothèse à défendre.

Avec x binaire, traité ou non,

$$E(y \mid 1) - E(y \mid 0) = \beta_1 + \underbrace{E(u \mid 1) - E(u \mid 0)}_{\text{écart de sélection}}.$$

C'est la décomposition de la séance 1. RLS.4 annule l'écart de sélection.

RLS.4, comment l'examiner

Pas avec les résidus. $\sum_i x_i \hat{u}_i = 0$ dans tout échantillon, par construction. Les résidus sont non corrélés à x , que RLS.4 soit vraie ou fausse.

En comparant ce qui est observé. Des groupes différents sur l'observé le sont sans doute aussi sur l'inobservé.

ECONMATH	< 15 h	≥ 15 h	JTRAIN2	Non formés	Formés
colgpa	2,86	2,70	Années d'études	10,09	10,35
Mère diplômée	0,66	0,54	Âge	25,05	25,82
Score ACT maths	23,15	23,37	Revenu 1975 (k\$)	1,27	1,53

À gauche, RLS.4 est douteuse. À droite, le tirage au sort la garantit, et les écarts restants sont dus au hasard du tirage.

En raisonnant. Lister ce que contient u , et se demander si cela varie avec x .

RLS.5, homoscélasticité

Ce qu'elle dit. $\text{Var}(u \mid x) = \sigma^2$ pour tout x .

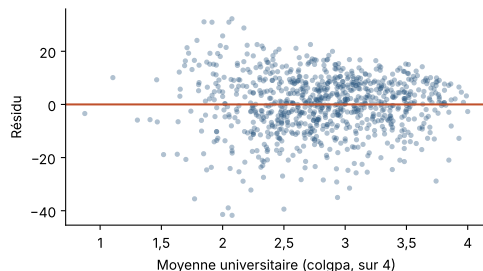
Comme $\beta_0 + \beta_1 x$ est fixe à x donné,

$\text{Var}(y \mid x) = \sigma^2$ aussi.

À quoi elle sert. À une formule simple pour la variance de $\hat{\beta}_1$. Elle ne joue **aucun rôle** dans l'absence de biais.

Si elle échoue (*hétéroscélasticité*). Les estimateurs restent sans biais, mais la formule usuelle de leur variance est fausse.

L'examiner. Tracer les résidus contre x . Leur écart-type par quart de `colgpa` vaut 14,3, 9,9, 10,5 et 8,4. RLS.5 est douteuse.



Résidus de score sur `colgpa`, plus dispersés pour les faibles moyennes. Des tests formels viendront plus tard.

Quel résultat dépend de quelle hypothèse

	RLS.1	RLS.2	RLS.3	RLS.4	RLS.5
Calculer $\hat{\beta}_0$ et $\hat{\beta}_1$			•		
Absence de biais de $\hat{\beta}_0$ et $\hat{\beta}_1$	•	•	•	•	
Variance σ^2/SCT_x	•	•	•	•	•
$\hat{\sigma}^2$ sans biais, erreurs-types	•	•	•	•	•
Lecture <i>ceteris paribus</i> de β_1	•			•	

RLS.4 est la plus importante et la plus fragile avec des données observationnelles. RLS.5 ne touche que la précision.

Pause

4. L'absence de biais des MCO

L'estimateur, c'est la vérité plus du bruit

Par la seconde identité utile de la séance 1, $\sum_i (x_i - \bar{x})(y_i - \bar{y}) = \sum_i (x_i - \bar{x}) y_i$, donc

$$\hat{\beta}_1 = \sum_{i=1}^n w_i y_i, \quad w_i = \frac{x_i - \bar{x}}{\text{SCT}_x}, \quad \sum_i w_i = 0, \quad \sum_i w_i x_i = 1, \quad \sum_i w_i^2 = \frac{1}{\text{SCT}_x}.$$

On remplace y_i par $\beta_0 + \beta_1 x_i + u_i$ (RLS.1 et RLS.2)

$$\hat{\beta}_1 = \beta_0 \underbrace{\sum_i w_i}_{=0} + \beta_1 \underbrace{\sum_i w_i x_i}_{=1} + \sum_i w_i u_i \quad \Rightarrow \quad \hat{\beta}_1 = \beta_1 + \sum_{i=1}^n w_i u_i.$$

L'écart $\hat{\beta}_1 - \beta_1$ est une somme pondérée des erreurs. Les poids ne dépendent que des x_i , et les observations éloignées de $\bar{x}$ pèsent le plus.

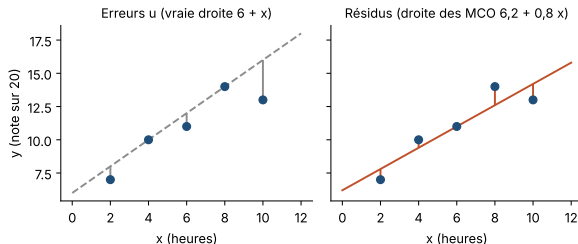
Propriétés des poids. $\sum_i (x_i - \bar{x}) = 0$, $\sum_i (x_i - \bar{x}) x_i = \text{SCT}_x$ (première identité utile),
 $\sum_i (x_i - \bar{x})^2 = \text{SCT}_x$.

Retour à l'exemple à la main

Supposons connue la vraie droite,
 $y = 6 + x$. Les erreurs sont
 $u_i = y_i - 6 - x_i$.

x_i	y_i	u_i	w_i
2	7	-1	-0,10
4	10	0	-0,05
6	11	-1	0
8	14	0	0,05
10	13	-3	0,10

$$\hat{\beta}_1 = 1 + (0,1 - 0,3) = 0,8$$



On retrouve la pente de la séance 1. L'erreur de l'étudiant 5, -3, avec le plus grand poids, tire la pente vers le bas. Avec d'autres erreurs, (1; -1; 2; 0; 1), on obtiendrait 1,05.

Théorème (absence de biais)

Sous RLS.1 à RLS.4, $E(\hat{\beta}_0) = \beta_0$ et $E(\hat{\beta}_1) = \beta_1$.

L'intuition. $\hat{\beta}_1 - \beta_1 = \sum_i w_i u_i$. Les poids sont fixés par les x_i . RLS.4 dit que, quelle que soit la valeur de x_i , l'erreur u_i est nulle en moyenne. Les erreurs ne poussent la pente ni vers le haut ni vers le bas, en moyenne sur les échantillons.

Ce que le théorème ne dit pas. Que $\hat{\beta}_1 = \beta_1$ dans notre échantillon. L'exemple à la main donne 0,8 pour une vraie pente de 1. Sans biais signifie « juste en moyenne sur tous les échantillons possibles ».

Ce dont il n'a pas besoin. RLS.5, ni la loi normale, ni un grand échantillon.

Démonstration pour $\hat{\beta}_1$

On raisonne à $x_1, \dots, x_n$ donnés⁶. Les poids w_i sont alors des constantes.

$$E(\hat{\beta}_1 \mid x_1, \dots, x_n) = \beta_1 + \sum_{i=1}^n w_i E(u_i \mid x_1, \dots, x_n) \quad (\text{CE.2})$$

$$= \beta_1 + \sum_{i=1}^n w_i E(u_i \mid x_i) \quad (\text{RLS.2})$$

$$= \beta_1 + \sum_{i=1}^n w_i \times 0 = \beta_1 \quad (\text{RLS.4})$$

RLS.3 garantit que w_i est défini ($\text{SCT}_x > 0$). Le résultat vaut pour toutes les valeurs possibles des x_i . Par les espérances itérées,

$$E(\hat{\beta}_1) = E[E(\hat{\beta}_1 \mid x_1, \dots, x_n)] = E(\beta_1) = \beta_1.$$

Démonstration pour $\hat{\beta}_0$

On moyenne $y_i = \beta_0 + \beta_1 x_i + u_i$ sur les n observations, $\bar{y} = \beta_0 + \beta_1 \bar{x} + \bar{u}$. Donc

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = \beta_0 + \bar{u} - (\hat{\beta}_1 - \beta_1) \bar{x}.$$

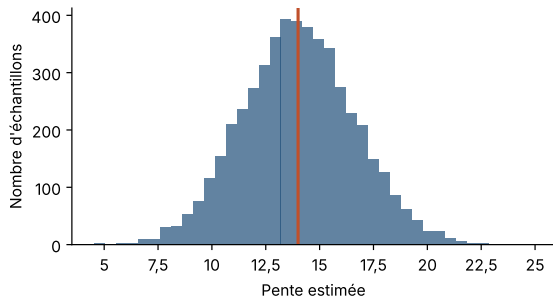
À $x_1, \dots, x_n$ donnés, $\bar{x}$ est une constante, $E(\bar{u} \mid x_1, \dots, x_n) = \frac{1}{n} \sum_i E(u_i \mid x_i) = 0$ et $E(\hat{\beta}_1 - \beta_1 \mid x_1, \dots, x_n) = 0$. Donc

$$E(\hat{\beta}_0 \mid x_1, \dots, x_n) = \beta_0 + 0 - 0 \times \bar{x} = \beta_0,$$

puis $E(\hat{\beta}_0) = \beta_0$ par les espérances itérées.

Exemple à la main. $\hat{\beta}_0 = 6 + (-1) - (0,8 - 1) \times 6 = 6,2$, la constante de la séance 1.

Voir l'absence de biais



Simulation de Monte-Carlo (*Monte Carlo simulation*). On fixe la vérité

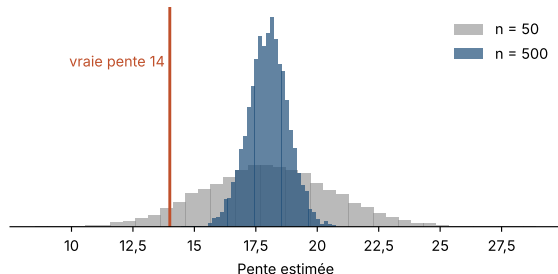
$$y = 32 + 14x + u, \quad u \sim \mathcal{N}(0; 11^2),$$

avec pour x la moyenne universitaire des 50 premiers étudiants. On tire 5 000 fois de nouvelles erreurs et on estime la pente.

Moyenne des 5 000 pentes, 14,01.
Vraie pente, 14 (en orange).

Valeurs proches de celles des 856 étudiants (pente 14,3, $\hat{\sigma} = 11,0$). $\mathcal{N}(0; 11^2)$ est la loi normale d'espérance 0 et d'écart-type 11, qui sert ici seulement à tirer les erreurs.

Quand RLS.4 échoue



Même simulation, mais les étudiants à forte moyenne sont aussi plus motivés, et la motivation est dans u

$$E(u | x) = 4(x - 2,8).$$

Alors $E(\hat{\beta}_1 | x) = 14 + 4 \sum_i w_i (x_i - 2,8) = 14 + 4 = 18$.

Avec 50 ou 500 observations, les pentes se centrent sur 18, pas sur 14.

Un échantillon plus grand rend l'estimation plus précise, **autour d'une mauvaise valeur**. Le biais ne disparaît pas avec n .

5. La variance des estimateurs des MCO

Théorème (variances d'échantillonnage)

Sous RLS.1 à RLS.5, à $x_1, \dots, x_n$ donnés,

$$\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{\text{SCT}_x}, \quad \text{Var}(\hat{\beta}_0) = \frac{\sigma^2 \frac{1}{n} \sum_{i=1}^n x_i^2}{\text{SCT}_x}.$$

Preuve pour $\hat{\beta}_1$. Les w_i sont des constantes et β_1 aussi. Les u_i sont indépendants (RLS.2), de variance σ^2 (RLS.5). Par la variance d'une somme,

$$\text{Var}(\hat{\beta}_1) = \text{Var}\left(\sum_i w_i u_i\right) = \sum_i w_i^2 \text{Var}(u_i) = \sigma^2 \sum_i w_i^2 = \frac{\sigma^2}{\text{SCT}_x}.$$

La preuve pour $\hat{\beta}_0$ est en annexe⁷. On note $\text{sd}(\hat{\beta}_1) = \sigma/\sqrt{\text{SCT}_x}$ l'écart-type de $\hat{\beta}_1$.

Trois leviers de précision

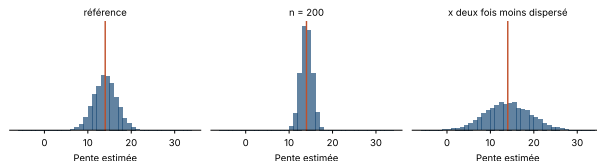
Comme $SCT_x = (n - 1) s_x^2$,

$$\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{(n - 1) s_x^2}.$$

La pente est plus précise quand

- ▶ les erreurs sont moins dispersées (σ petit)
- ▶ l'échantillon est plus grand (n)
- ▶ x est plus dispersé (s_x)

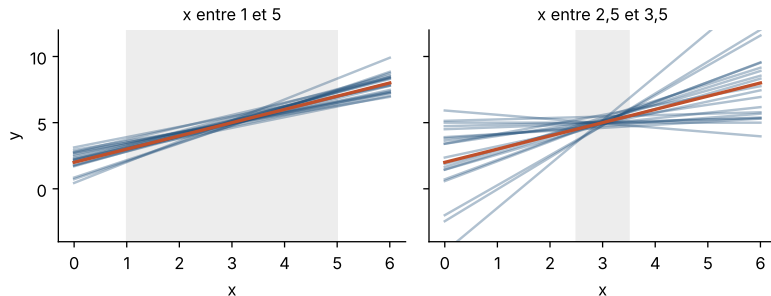
Plan de simulation	$sd(\hat{\beta}_1)$	
	théorique	simulé
Référence ($n = 50, \sigma = 11$)	2,61	2,62
$n = 200$	1,30	1,29
x deux fois moins dispersé	5,21	5,22
$\sigma = 5,5$	1,30	1,30



Quatre fois plus d'observations divisent l'écart-type par 2.

▶ Code

Pourquoi la dispersion de x aide



Vingt échantillons de 20 points autour de la vraie droite $y = 2 + x$ (en orange), avec les mêmes erreurs à gauche et à droite. Quand les x sont resserrés (zone grisée), une petite erreur fait pivoter la droite. Quand ils sont étalés, la droite est tenue à ses deux bouts. Écart-type théorique de la pente, 0,18 à gauche, 0,74 à droite.

► Code

Quand RLS.5 échoue

Même vérité, $y = 32 + 14x + u$, mais
l'écart-type de u varie avec x

$$\text{sd}(u \mid x) = 3 + 14|x - 2,8|.$$

La variance moyenne de u vaut 121, comme
dans la référence. RLS.4 tient toujours.

La preuve de la variance donne en général
 $\text{Var}(\hat{\beta}_1) = \sum_i w_i^2 \text{Var}(u_i \mid x_i)$, qui ne se
simplifie plus.

La pente reste sans biais, mais la formule usuelle sous-estime sa dispersion de plus d'un quart. D'autres erreurs-types, dites robustes, corrigent ce défaut. Elles seront vues avec l'inférence.

5 000 échantillons	
Moyenne des pentes	14,00
Écart-type simulé	3,58
Vrai écart-type	3,56
Formule usuelle $\sigma/\sqrt{\text{SCT}_x}$	2,61

6. Estimer σ^2 et l'erreur-type

Estimer la variance de l'erreur

La formule $\text{Var}(\hat{\beta}_1) = \sigma^2 / \text{SCT}_x$ contient $\sigma^2 = E(u^2)$, inconnue.

- ▶ Avec les erreurs, on calculerait $\frac{1}{n} \sum_i u_i^2$. Elles ne sont pas observées
- ▶ On les remplace par les résidus, $\hat{u}_i = u_i - (\hat{\beta}_0 - \beta_0) - (\hat{\beta}_1 - \beta_1) x_i$
- ▶ SCR/n sous-estime σ^2 . Les résidus vérifient deux contraintes, $\sum_i \hat{u}_i = 0$ et $\sum_i x_i \hat{u}_i = 0$.
Connaissant $n - 2$ résidus, on en déduit les deux autres. Il reste $n - 2$ *degrés de liberté*

Théorème (estimateur sans biais de σ^2)

Sous RLS.1 à RLS.5, $\hat{\sigma}^2 = \frac{\text{SCR}}{n-2} = \frac{1}{n-2} \sum_{i=1}^n \hat{u}_i^2$ vérifie $E(\hat{\sigma}^2) = \sigma^2$.

Démonstration en annexe⁸. C'est la même logique que le diviseur $n - 1$ de s_x^2 , avec deux paramètres estimés au lieu d'un.

Voir le rôle de $n - 2$

Simulation de référence, $\sigma^2 = 121$, $n = 50$, 5 000 échantillons.

	Moyenne sur 5 000
$SCR/(n - 2)$	120,85
SCR/n	116,01

SCR/n se trompe en moyenne de $121 \times 2/50 \approx 4,8$. Avec $n - 2$, l'erreur disparaît en moyenne.

L'écart compte surtout pour les petits échantillons. Avec les 856 étudiants, il est de 0,2 %.

$\hat{\sigma} = \sqrt{\hat{\sigma}^2}$ est l'*erreur-type de la régression* (*standard error of the regression*, SER). Elle estime l'écart-type des facteurs inobservés. Stata l'appelle *Root MSE*.

Erreur-type (*standard error*)

$$se(\hat{\beta}_1) = \frac{\hat{\sigma}}{\sqrt{SCT_x}}, \quad se(\hat{\beta}_0) = \hat{\sigma} \sqrt{\frac{\frac{1}{n} \sum_i x_i^2}{SCT_x}}$$

- ▶ On remplace σ par $\hat{\sigma}$ dans l'écart-type. $sd(\hat{\beta}_1)$ est un nombre inconnu, $se(\hat{\beta}_1)$ une estimation calculée sur l'échantillon, qui change d'un échantillon à l'autre
- ▶ Elle mesure la précision de $\hat{\beta}_1$ avec un seul échantillon. C'est la réponse à la troisième question du début
- ▶ Dans la simulation de référence, la moyenne des 5 000 erreurs-types vaut 2,59, proche de l'écart-type des pentes, 2,62

Elle repose sur RLS.5. Sous hétéroscédasticité, elle se trompe comme la formule usuelle.

D'où vient le (0,70) de la séance 1

	score
colgpa	14,32*** (0,70)
Constante	32,31*** (2,00)
<i>n</i>	856
<i>R</i> ²	0,329
$\hat{\sigma}$	10,98

MCO. Erreurs-types entre parenthèses.

*** significatif à 1 %, ** à 5 %, * à 10 %.

► $\hat{\sigma}^2 = 103\,036,6/854 = 120,65$, d'où
 $\hat{\sigma} = 10,98$

► $SCT_x = 246,43$, donc

$$se(\hat{\beta}_1) = \frac{10,98}{\sqrt{246,43}} = 0,70$$

► De même $se(\hat{\beta}_0) = 2,00$, les valeurs de la table

Ces erreurs-types supposent RLS.5, que la slide « **RLS.5, homoscélasticité** » rend douteuse. Elles seront révisées avec les erreurs-types robustes.

L'essentiel de la séance

- ▶ $\hat{\beta}_1$ est une variable aléatoire. Sa loi sur les échantillons possibles est sa distribution d'échantillonnage
- ▶ $E(u | x) = 0$ implique $E(u) = 0$ et $\text{Cov}(x, u) = 0$. Espérances itérées, $E[E(Y | X)] = E(Y)$
- ▶ RLS.1 à RLS.4 donnent $E(\hat{\beta}_1) = \beta_1$, car $\hat{\beta}_1 = \beta_1 + \sum_i w_i u_i$. RLS.4 est l'hypothèse clé, celle du *ceteris paribus* en moyenne. Si elle échoue, le biais ne diminue pas avec n
- ▶ Avec RLS.5, $\text{Var}(\hat{\beta}_1) = \sigma^2 / \text{SCT}_x$. Précision accrue avec n et la dispersion de x
- ▶ $\hat{\sigma}^2 = \text{SCR} / (n - 2)$ est sans biais et $\text{se}(\hat{\beta}_1) = \hat{\sigma} / \sqrt{\text{SCT}_x}$

- ▶ La régression multiple. Tenir compte de plusieurs facteurs observés à la fois, comme `colgpa` et le milieu familial pour mesurer l'effet du travail salarié
- ▶ Lecture *ceteris paribus* des coefficients et biais de variable omise
- ▶ Les hypothèses RLM.1 à RLM.5, qui généralisent RLS.1 à RLS.5

Pour préparer, Wooldridge, sections 3-1 et 3-2.

Travaux dirigés

Série d'exercices de la séance 2

10 questions de quiz et 5 exercices

(support distinct)

Annexes

Explications

Annexe 1. Tirer avec ou sans remise

Avec remise. Chaque tirage choisit un étudiant parmi les 856, quels que soient les tirages précédents. Un même étudiant peut sortir deux fois. Les tirages sont indépendants et suivent tous la même loi, celle de la population. C'est exactement un échantillon aléatoire au sens de RLS.2.

Sans remise. Un étudiant déjà tiré ne peut plus l'être. Les tirages ne sont plus tout à fait indépendants.

Exemple. Sachant que le premier tirage a donné l'étudiant A, la probabilité que le second donne encore A vaut $1/856$ avec remise et 0 sans remise. La différence est minime tant que l'échantillon est petit devant la population.

Les enquêtes réelles tirent sans remise dans de très grandes populations. On les traite comme des échantillons aléatoires. Nous tirons avec remise pour que RLS.2 soit exactement satisfaite.

Annexe 2. Variables aléatoires continues

Une variable continue prend chaque valeur avec une probabilité nulle. On décrit sa loi par une *densité f (probability density function)*, positive, d'aire totale 1. Les probabilités sont des aires

$$P(a \leq X \leq b) = \int_a^b f(x) dx.$$

L'espérance remplace la somme par une intégrale, $E(X) = \int x f(x) dx$. La variance garde sa définition. Les règles E.1 à E.3 et CE.2 à CE.4 sont inchangées. Nous n'aurons aucune intégrale à calculer.

Exemple. X uniforme sur $[0 ; 4]$, de densité $f(x) = 1/4$ sur cet intervalle.

- ▶ $P(2 \leq X \leq 3) = 1 \times \frac{1}{4} = 0,25$, l'aire d'un rectangle
- ▶ $P(X = 3) = 0$, l'aire d'un segment
- ▶ $E(X) = 2$, le centre de symétrie

La loi normale $\mathcal{N}(\mu ; \sigma^2)$, en cloche, d'espérance μ et de variance σ^2 , est continue. Les simulations l'utilisent pour tirer les erreurs.

Annexe 3. La loi des grands nombres

Énoncé. Si $X_1, \dots, X_n$ sont indépendantes, de même loi, d'espérance μ , alors pour tout $\varepsilon > 0$, $P(|\bar{X}_n - \mu| > \varepsilon)$ tend vers 0 quand n tend vers l'infini (Wooldridge, 2020, section C-3a).

Pourquoi, quand la variance σ^2 existe. Deux résultats suffisent.

- ▶ $\text{Var}(\bar{X}_n) = \sigma^2/n$ (slide « La variance d'une somme »)
- ▶ L'inégalité de Tchebychev, $P(|Z - E(Z)| > \varepsilon) \leq \text{Var}(Z)/\varepsilon^2$. En effet, $(Z - E(Z))^2 \geq \varepsilon^2$ chaque fois que $|Z - E(Z)| > \varepsilon$, donc $\text{Var}(Z) \geq \varepsilon^2 P(|Z - E(Z)| > \varepsilon)$

Ensemble, $P(|\bar{X}_n - \mu| > \varepsilon) \leq \sigma^2/(n\varepsilon^2)$, qui tend vers 0.

Exemple. Notes, $\sigma^2 = 179,37$. Pour $n = 2\,000$ et $\varepsilon = 1$, la borne vaut $179,37/2\,000 \approx 0,09$. La moyenne de 2 000 notes tirées au hasard a moins de 9 % de chances de s'écarter de plus d'un point de 72,60.

Annexe 4. La loi des espérances itérées

Démonstration, cas discret. $E(Y | X)$ est une fonction de X . Son espérance pondère chaque valeur $E(Y | X = x)$ par $P(X = x)$

$$E[E(Y | X)] = \sum_x P(X = x) \sum_y y P(Y = y | X = x)$$

$$= \sum_x \sum_y y P(X = x, Y = y)$$

$$= \sum_y y P(Y = y) = E(Y)$$

$$P(Y = y | X = x) = \frac{P(X=x, Y=y)}{P(X=x)}$$

en sommant d'abord sur x .

Lecture. La moyenne générale est la moyenne des moyennes de groupe, pondérée par la taille des groupes.

Exemple. $P(X = 0) = 615/856$ et $P(X = 1) = 241/856$, donc

$$\frac{615}{856} \times 73,95 + \frac{241}{856} \times 69,17 = 72,60.$$

Annexe 5. Pourquoi $E(u) = 0$ ne coûte rien

Supposons $E(u) = \alpha_0 \neq 0$. On réécrit le modèle

$$y = (\beta_0 + \alpha_0) + \beta_1 x + (u - \alpha_0).$$

La nouvelle erreur $u - \alpha_0$ a une espérance nulle par E.2. La pente β_1 n'a pas changé. Seule la constante absorbe α_0 .

Exemple. Si la motivation, comprise dans u , ajoute en moyenne 3 points à chacun, le modèle $\text{score} = 32 + 14 \text{ colgpa} + u$ avec $E(u) = 3$ s'écrit aussi $\text{score} = 35 + 14 \text{ colgpa} + v$ avec $E(v) = 0$. La pente, qui seule nous intéresse, est la même.

Le contenu de RLS.4 est donc $E(u | x) = E(u)$. La moyenne des facteurs inobservés ne doit pas varier avec x

Annexe 6. Pourquoi raisonner à $x_1, \dots, x_n$ donnés

L'idée. On calcule d'abord l'espérance en gardant fixes les x_i de l'échantillon, comme si seules les erreurs étaient tirées à nouveau. Les poids w_i sont alors des constantes, et E.3 s'applique.

Pourquoi c'est légitime. Le calcul vaut pour *toutes* les valeurs possibles des x_i . Les espérances itérées donnent ensuite le résultat sans conditionnement. On ne suppose pas que x est fixé par l'expérimentateur. C'est un outil de calcul.

Exemple. La simulation de la slide « Voir l'absence de biais » fait exactement cela. Les 50 valeurs de `colgpa` restent les mêmes, seules les erreurs changent d'un échantillon à l'autre.

Le piège. Imaginer des x fixés par un expérimentateur fait oublier que, dans les données observationnelles, x et u sont tirés ensemble. C'est RLS.4 qui garantit que les u_i ont une moyenne nulle quelle que soit la valeur de x_i

Annexe 7. La variance de $\hat{\beta}_0$

On part de $\hat{\beta}_0 - \beta_0 = \bar{u} - (\hat{\beta}_1 - \beta_1) \bar{x}$, à $x_1, \dots, x_n$ donnés.

- ▶ $\text{Var}(\bar{u}) = \sigma^2/n$, variance d'une moyenne de n erreurs indépendantes de variance σ^2
- ▶ $\text{Var}(\hat{\beta}_1) = \sigma^2/\text{SCT}_x$
- ▶ $\text{Cov}(\bar{u}, \hat{\beta}_1 - \beta_1) = \text{Cov}(\frac{1}{n} \sum_i u_i, \sum_j w_j u_j) = \frac{1}{n} \sum_i w_i \sigma^2 = 0$, car $\text{Cov}(u_i, u_j) = 0$ si $i \neq j$ et $\sum_i w_i = 0$

Donc

$$\text{Var}(\hat{\beta}_0) = \frac{\sigma^2}{n} + \bar{x}^2 \frac{\sigma^2}{\text{SCT}_x} = \sigma^2 \frac{\text{SCT}_x + n\bar{x}^2}{n\text{SCT}_x} = \frac{\sigma^2 \frac{1}{n} \sum_i x_i^2}{\text{SCT}_x},$$

car $\text{SCT}_x + n\bar{x}^2 = \sum_i x_i^2$ (première identité utile de la séance 1).

Exemple. ECONMATH, $\frac{1}{n} \sum_i x_i^2 = 2,815^2 + 246,43/856 = 8,21$, d'où
 $\text{se}(\hat{\beta}_0) = \sqrt{120,65 \times 8,21/246,43} = 2,00$.

Annexe 8. $\hat{\sigma}^2$ est sans biais

Les résidus. En moyennant $\hat{u}_i = u_i - (\hat{\beta}_0 - \beta_0) - (\hat{\beta}_1 - \beta_1)x_i$ sur i , avec $\sum_i \hat{u}_i = 0$, puis en soustrayant,

$$\hat{u}_i = (u_i - \bar{u}) - (\hat{\beta}_1 - \beta_1)(x_i - \bar{x}).$$

Les carrés. Le double produit vaut $-2(\hat{\beta}_1 - \beta_1)^2 \text{SCT}_x$, car $\sum_i (x_i - \bar{x})(u_i - \bar{u}) = \text{SCT}_x(\hat{\beta}_1 - \beta_1)$. Donc

$$\text{SCR} = \sum_i (u_i - \bar{u})^2 - (\hat{\beta}_1 - \beta_1)^2 \text{SCT}_x.$$

Les espérances, à $x_1, \dots, x_n$ donnés.

- ▶ $E[\sum_i (u_i - \bar{u})^2] = (n-1)\sigma^2$, comme pour le diviseur $n-1$ (séance 1, polycopié, annexe A)
- ▶ $E[(\hat{\beta}_1 - \beta_1)^2] \text{SCT}_x = \text{Var}(\hat{\beta}_1) \text{SCT}_x = \sigma^2$

Donc $E(\text{SCR}) = (n-2)\sigma^2$ et $E(\hat{\sigma}^2) = \sigma^2$.

Annexes

Code Python, R et Stata

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
econ = wd.data("econmath") # les 856 étudiants, vus comme la population
x, y = econ["colgpa"].to_numpy(), econ["score"].to_numpy()
pente = lambda a, b: np.cov(a, b)[0, 1] / np.var(a, ddof=1) # pente des MCO
rng = np.random.default_rng(2026) # graine : mêmes tirages à chaque exécution
ech = rng.integers(0, 856, size=(4, 50)) # 4 échantillons de 50 étudiants, avec remise
print("pente sur les 856 :", round(pente(x, y), 2))
print("pentés des 4 échantillons :", np.round([pente(x[e], y[e]) for e in ech], 2))
g = np.linspace(0.8, 4.1, 50)
fig, axs = plt.subplots(1, 4, figsize=(9, 2.6), sharey=True)
[a.scatter(x[e], y[e], s=8, color="#1F4E79", alpha=0.6) for a, e in zip(axs, ech)]
droite = lambda e: y[e].mean() + pente(x[e], y[e]) * (g - x[e].mean()) # droite des MCO
[a.plot(g, droite(np.arange(856)), color="#8C8C8C", ls="--") for a in axs] # les 856
[a.plot(g, droite(e), color="#C0502B") for a, e in zip(axs, ech)]
[a.set_title(f"pente = {pente(x[e], y[e]):.1f}".replace(".", ","), fontsize=10)
 for a, e in zip(axs, ech)]
[a.set_xlabel("colgpa") for a in axs]; axs[0].set_ylabel("score")
[a.spines[["top", "right"]].set_visible(False) for a in axs]
fig.tight_layout(); fig.savefig("fig_echantillons.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath") # les 856 étudiants, vus comme la population
x <- econmath$colgpa; y <- econmath$score
pente <- function(a, b) cov(a, b) / var(a) # pente des MCO
set.seed(2026) # graine : mêmes tirages à chaque exécution
ech <- replicate(4, sample.int(856, 50, replace = TRUE)) # 4 échantillons de 50, avec remise
round(pente(x, y), 2) # pente sur les 856
round(apply(ech, 2, function(e) pente(x[e], y[e])), 2) # pentes des 4 échantillons
par(mfrow = c(1, 4), mar = c(4, 4, 2, 1))
for (k in 1:4) {
  e <- ech[, k]
  plot(x[e], y[e], pch = 16, cex = 0.6, col = "#1F4E79", xlim = c(0.8, 4.1), ylim = c(20, 100),
       xlab = "colgpa", ylab = "score", main = paste("pente =", round(pente(x[e], y[e]), 1)))
  abline(lm(y ~ x), col = "#8C8C8C", lty = 2) # droite des 856
  abline(lm(y[e] ~ x[e]), col = "#C0502B") # droite de l'échantillon
}
```

```
use econmath.dta, clear           // les 856 étudiants, vus comme la population
quietly regress score colgpa
local a = _b[_cons]
local b = _b[colgpa]
display "pente sur les 856 : " %6.2f `b'
set seed 2026                     // graine : mêmes tirages à chaque exécution
forvalues k = 1/4 {
    preserve
    bsample 50                     // 50 étudiants tirés avec remise
    quietly regress score colgpa
    display "échantillon `k' : pente = " %6.2f _b[colgpa]
    twoway (scatter score colgpa, mcolor("31 78 121") msize(small)) ///
        (function y = `a' + `b'*x, range(0.8 4.1) lcolor(gs9) lpattern(dash)) ///
        (lfit score colgpa, lcolor("192 80 43")), legend(off) ///
        title("pente = `: display %4.1f _b[colgpa]'"') name(g`k', replace)
    restore
}
graph combine g1 g2 g3 g4, rows(1) ycommon
graph export "fig_echantillons_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
econ = wd.data("econmath")
x, y = econ["colgpa"].to_numpy(), econ["score"].to_numpy()
pente = lambda a, b: np.cov(a, b)[0, 1] / np.var(a, ddof=1) # pente des MCO
rng = np.random.default_rng(2026)
ech = rng.integers(0, 856, size=(1000, 50)) # 1 000 échantillons de 50, avec remise
b1 = np.array([pente(x[e], y[e]) for e in ech])
print("moyenne et écart-type des 1 000 pentes :", round(b1.mean(), 2), round(b1.std(ddof=1), 2))
print("plus petite et plus grande pente :", round(b1.min(), 2), round(b1.max(), 2))
fig, ax = plt.subplots(figsize=(5.2, 2.9))
ax.hist(b1, bins=30, color="#1F4E79", alpha=0.7)
ax.axvline(pente(x, y), color="#C0502B", lw=2) # pente calculée sur les 856
ax.set_xlabel("Pente estimée sur un échantillon de 50 étudiants")
ax.set_ylabel("Nombre d'échantillons")
ax.spines[["top", "right"].set_visible(False)
ax.xaxis.set_major_formatter(lambda v, p: f"{v:g}".replace(".", ","))
fig.tight_layout(); fig.savefig("fig_distrib_echant.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x <- econmath$colgpa; y <- econmath$score
pente <- function(a, b) cov(a, b) / var(a)      # pente des MCO
set.seed(2026)
b1 <- replicate(1000, {e <- sample.int(856, 50, replace = TRUE); pente(x[e], y[e])})
round(c(moyenne = mean(b1), ecart_type = sd(b1)), 2)
round(range(b1), 2)                            # plus petite et plus grande pente
hist(b1, breaks = 30, col = "#1F4E79", border = "white",
     xlab = "Pente estimée sur un échantillon de 50 étudiants",
     ylab = "Nombre d'échantillons", main = "")
abline(v = pente(x, y), col = "#C0502B", lwd = 2)  # pente calculée sur les 856
```

```
use econmath.dta, clear
quietly regress score colgpa
local b = _b[colgpa] // pente calculée sur les 856
set seed 2026
tempname sim
postfile `sim' b1 using pentes_ech, replace
forvalues r = 1/1000 {
    preserve
    quietly bsample 50 // un échantillon de 50, avec remise
    quietly regress score colgpa
    post `sim' (_b[colgpa])
    restore
}
postclose `sim'
use pentes_ech, clear
summarize b1 // moyenne, écart-type, minimum, maximum
histogram b1, bin(30) frequency fcolor("31 78 121") lcolor(white) ///
    xline(`b', lcolor("192 80 43") lwidth(thick)) ///
    xtitle("Pente estimée sur un échantillon de 50 étudiants") ytitle("Nombre d'échantillons")
graph export "fig_distrib_echant_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"])    # installe si absent
import wooldridge as wd
econ = wd.data("econmath")                                     # population de 856 étudiants
X = (econ["work"] >= 15).astype(int)                           # X = 1 si 15 heures ou plus de travail salarié
print("loi de X :", X.value_counts().sort_index().tolist(), "étudiants sur", len(X))
print("P(X = 1) :", round(X.mean(), 4), "| P(X = 0) :", round(1 - X.mean(), 4))
print("chaque note a la probabilité 1/856 :", round(1 / 856, 5))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")                                # population de 856 étudiants
X <- as.integer(econmath$work >= 15)             # X = 1 si 15 heures ou plus de travail salarié
table(X)                                         # loi de X, en effectifs
round(c(P_X1 = mean(X), P_X0 = 1 - mean(X)), 4)
round(1 / 856, 5)                               # probabilité de chaque note
```

```
use econmath.dta, clear           // population de 856 étudiants
generate X = (work >= 15)         // X = 1 si 15 heures ou plus de travail salarié
tabulate X                        // loi de X, effectifs et proportions
display "probabilité de chaque note : " %7.5f 1/856
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
econ = wd.data("econmath")
y = econ["score"].to_numpy()
rng = np.random.default_rng(2026)
tirages = rng.choice(y, size=2000) # 2 000 étudiants tirés au hasard, avec remise
moy = np.cumsum(tirages) / np.arange(1, 2001) # moyenne des n premiers tirages
print("moyenne après 10, 100, 2 000 tirages :", moy[[9, 99, 1999]].round(2))
print("E(Y) :", round(y.mean(), 2))
fig, ax = plt.subplots(figsize=(5.2, 2.9))
ax.plot(np.arange(1, 2001), moy, color="#1F4E79")
ax.axhline(y.mean(), color="#C0502B", ls="--")
ax.set_xscale("log"); ax.set_ylim(60, 85)
ax.set_xticks([1, 10, 100, 1000], ["1", "10", "100", "1 000"])
ax.set_xlabel("Nombre de tirages n (échelle logarithmique)"); ax.set_ylabel("Moyenne des n notes")
ax.spines[["top", "right"]].set_visible(False)
fig.tight_layout(); fig.savefig("fig_lgn.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
y <- econmath$score
set.seed(2026)
tirages <- sample(y, 2000, replace = TRUE)      # 2 000 étudiants tirés au hasard, avec remise
moy <- cumsum(tirages) / (1:2000)               # moyenne des n premiers tirages
round(moy[c(10, 100, 2000)], 2); round(mean(y), 2)
plot(1:2000, moy, type = "l", log = "x", col = "#1F4E79", ylim = c(60, 85),
     xlab = "Nombre de tirages n (échelle logarithmique)", ylab = "Moyenne des n notes")
abline(h = mean(y), col = "#C0502B", lty = 2)
```

```
use econmath.dta, clear
keep score
quietly summarize score
local EY = r(mean)
expand 3 // 2 568 lignes, les 856 premières sont les données
set seed 2026
generate i = runiformint(1, 856) in 1/2000 // 2 000 étudiants tirés au hasard, avec remise
generate tirage = score[i] in 1/2000
keep in 1/2000
generate n = _n
generate moy = sum(tirage) / n // moyenne des n premiers tirages
list n moy if inlist(n, 10, 100, 2000), clean
display "E(Y) : " %6.2f `EY'
twoway line moy n, xscale(log) xlabel(1 10 100 1000) lcolor("31 78 121") ///
    yline(`EY', lcolor("192 80 43") lpattern(dash)) yscale(range(60 85)) ///
    xtitle("Nombre de tirages n (échelle logarithmique)") ytitle("Moyenne des n notes")
graph export "fig_lgn_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
econ = wd.data("econmath")
Y = econ["score"] # Y = note finale, chaque note a la probabilité 1/856
mu = Y.mean() # E(Y), moyenne de la population
print("E(Y) :", round(mu, 2))
print("E(0,2 Y) :", round((0.2 * Y).mean(), 2), "| 0,2 E(Y) :", round(0.2 * mu, 2))
print("E(Y - mu) :", round((Y - mu).mean(), 10) + 0)
print("E(Y^2) :", round((Y**2).mean(), 1), "| [E(Y)]^2 :", round(mu**2, 1))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
Y <- econmath$score                                # Y = note finale, chaque note a la probabilité 1/856
mu <- mean(Y)                                       # E(Y), moyenne de la population
round(mu, 2)
round(c(mean(0.2 * Y), 0.2 * mu), 2)               #  $E(0,2 Y) = 0,2 E(Y)$ 
round(mean(Y - mu), 10)                           #  $E(Y - \mu) = 0$ 
round(c(EY2 = mean(Y^2), EY_carre = mu^2), 1)
```

```
use econmath.dta, clear
quietly summarize score           // chaque note a la probabilité 1/856
local mu = r(mean)               // E(Y), moyenne de la population
display "E(Y) : " %6.2f `mu'
generate Y20 = 0.2*score         // note sur 20
generate ecart = score - `mu'
generate Y2 = score^2
tabstat Y20 ecart Y2, statistics(mean) format(%9.2f) // E(0,2 Y), E(Y - mu), E(Y^2)
display "0,2 E(Y) : " %6.2f 0.2*`mu' " | [E(Y)]^2 : " %7.1f `mu'^2
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
econ = wd.data("econmath")
X = (econ["work"] >= 15).astype(int); Y = econ["score"]
p = X.mean()
print("Var(X) = p(1 - p) :", round(p * (1 - p), 4), "| E(X^2) - p^2 :", round((X**2).mean() - p**2, 4))
print("Var(Y) :", round(Y.var(ddof=0), 2), " (division par N = 856) | écart-type :", round(Y.std(ddof=0), 2))
print("E(Y^2) - [E(Y)]^2 :", round((Y**2).mean() - Y.mean()**2, 2))
print("Var(0,2 Y) = 0,04 Var(Y) :", round((0.2 * Y).var(ddof=0), 3), round(0.04 * Y.var(ddof=0), 3))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
X <- as.integer(econmath$work >= 15); Y <- econmath$score
p <- mean(X)
round(c(p * (1 - p), mean(X^2) - p^2), 4)      #  $Var(X) = p(1 - p)$ 
VarY <- mean((Y - mean(Y))^2)                  # division par N = 856
round(c(VarY = VarY, ecart_type = sqrt(VarY), autre = mean(Y^2) - mean(Y)^2), 2)
round(c(mean((0.2 * Y - mean(0.2 * Y))^2), 0.04 * VarY), 3)  #  $Var(0,2 Y) = 0,04 Var(Y)$ 
```

```
use econmath.dta, clear
generate X = (work >= 15)
quietly summarize X
display "Var(X) = p(1 - p) : " %6.4f r(mean)*(1 - r(mean))
quietly summarize score
local v = r(Var)*(r(N) - 1)/r(N)           // division par N = 856
display "Var(Y) : " %7.2f `v' " | écart-type : " %6.2f sqrt(`v')
display "Var(0,2 Y) = 0,04 Var(Y) : " %6.3f 0.04*`v'
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
econ = wd.data("econmath")
X = (econ["work"] >= 15).astype(int); Y = econ["score"]
cov = (X * Y).mean() - X.mean() * Y.mean() #  $Cov(X, Y) = E(XY) - E(X)E(Y)$ 
print("Cov(X, Y) :", round(cov, 3))
print("autre écriture,  $E[(X - E X)(Y - E Y)]$  :", round(((X - X.mean()) * (Y - Y.mean())).mean(), 3))
print("Corr(X, Y) :", round(cov / (X.std(ddof=0) * Y.std(ddof=0)), 3))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
X <- as.integer(econmath$work >= 15); Y <- econmath$score
covXY <- mean(X * Y) - mean(X) * mean(Y)          #  $Cov(X, Y) = E(XY) - E(X)E(Y)$ 
round(c(covXY, mean((X - mean(X)) * (Y - mean(Y)))), 3)
sdX <- sqrt(mean((X - mean(X))^2)); sdY <- sqrt(mean((Y - mean(Y))^2))
round(covXY / (sdX * sdY), 3)                     #  $Corr(X, Y)$ 
```

```
use econmath.dta, clear
generate X = (work >= 15)
generate XY = X*score
quietly summarize X
local EX = r(mean)
local sdX = sqrt(r(Var)*(r(N) - 1)/r(N))    // écart-type de la population (division par N)
quietly summarize score
local EY = r(mean)
local sdY = sqrt(r(Var)*(r(N) - 1)/r(N))
quietly summarize XY
display "Cov(X, Y) = E(XY) - E(X)E(Y) : " %7.3f r(mean) - `EX'*`EY'
display "Corr(X, Y) : " %6.3f (r(mean) - `EX'*`EY')/(`sdX'*`sdY')
```

Code Python, covariance nulle sans indépendance

[Python](#)[R](#)[Stata](#)

```
import numpy as np
x = np.array([-1, 0, 1]); prob = np.array([1, 1, 1]) / 3  # X prend -1, 0, 1 avec probabilité 1/3
y = x**2  # Y = X^2, entièrement déterminée par X
EX, EY, EXY = (prob * x).sum(), (prob * y).sum(), (prob * x * y).sum()
print("E(X), E(Y), E(XY) :", EX, round(EY, 4), EXY, "| Cov(X, Y) :", EXY - EX * EY)
print("P(X = 0 et Y = 1) :", 0, "| P(X = 0) P(Y = 1) :", round(prob[1] * (prob[0] + prob[2]), 4))
```

[◀ Retour](#)

Code R, covariance nulle sans indépendance

[Python](#)**R**[Stata](#)

```
x <- c(-1, 0, 1); prob <- c(1, 1, 1) / 3      # X prend -1, 0, 1 avec probabilité 1/3
y <- x^2                                       # Y = X^2, entièrement déterminée par X
EX <- sum(prob * x); EY <- sum(prob * y); EXY <- sum(prob * x * y)
c(EX = EX, EY = round(EY, 4), EXY = EXY, Cov = EXY - EX * EY)
c(P_X0_Y1 = 0, produit = round(prob[2] * (prob[1] + prob[3]), 4))
```

[◀ Retour](#)

```
clear
input x prob
-1 0.3333333333
0 0.3333333333
1 0.3333333333
end
generate y = x^2 // Y = X^2, entièrement déterminée par X
generate px = prob*x
generate py = prob*y
generate pxy = prob*x*y
tabstat px py pxy, statistics(sum) // E(X), E(Y), E(XY) : Cov = E(XY) - E(X)E(Y) = 0
display "P(X = 0 et Y = 1) = 0 | P(X = 0) P(Y = 1) = " %6.4f (1/3)*(2/3)
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
econ = wd.data("econmath")
y = econ["score"].to_numpy()
sigma = y.std(ddof=0) # écart-type de la note dans la population
rng = np.random.default_rng(2026)
moyennes = rng.choice(y, size=(5000, 50)).mean(axis=1) # 5 000 moyennes de 50 notes
print("écart-type théorique de la moyenne, sigma / racine(50) :", round(sigma / np.sqrt(50), 3))
print("écart-type des 5 000 moyennes simulées :", round(moyennes.std(ddof=1), 3))
print("moyenne des 5 000 moyennes :", round(moyennes.mean(), 2), "| E(Y) :", round(y.mean(), 2))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
y <- econmath$score
sigma <- sqrt(mean((y - mean(y))^2))          # écart-type de la note dans la population
set.seed(2026)
moyennes <- replicate(5000, mean(sample(y, 50, replace = TRUE))) # 5 000 moyennes de 50 notes
round(c(theorique = sigma / sqrt(50), simule = sd(moyennes)), 3)
round(c(mean(moyennes), mean(y)), 2)
```

```
use econmath.dta, clear
quietly summarize score
display "écart-type théorique de la moyenne : " %6.3f sqrt(r(Var)*(r(N) - 1)/r(N))/sqrt(50)
set seed 2026
tempname sim
postfile `sim' moy using moyennes, replace
forvalues r = 1/5000 {
    preserve
    quietly bsample 50                // 50 notes tirées avec remise
    quietly summarize score
    post `sim' (r(mean))
    restore
}
postclose `sim'
use moyennes, clear
summarize moy                        // moyenne et écart-type des 5 000 moyennes
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
econ = wd.data("econmath")
X = (econ["work"] >= 15).astype(int); Y = econ["score"]
m = Y.groupby(X).mean(); p = X.mean() #  $E(Y | X = 0)$ ,  $E(Y | X = 1)$ ,  $P(X = 1)$ 
print("E(Y|X=0), E(Y|X=1) :", m.round(2).tolist(), "| P(X=1) :", round(p, 4))
classe = (econ["colgpa"] // 0.25) * 0.25 + 0.125 # classes de colgpa de largeur 0,25
t = econ.groupby(classe)["score"].agg(["mean", "size"]); t = t[t["size"] >= 10]
print(t.round(2))
fig, ax = plt.subplots(figsize=(5.2, 3.2))
ax.scatter(econ["colgpa"], Y, s=8, color="#1F4E79", alpha=0.15)
ax.scatter(t.index, t["mean"], color="#C0502B", s=30, zorder=3) # moyennes par classe
ax.set_xlabel("Moyenne universitaire (colgpa, sur 4)"); ax.set_ylabel("Note finale (score, %)")
ax.spines[["top", "right"]].set_visible(False)
ax.xaxis.set_major_formatter(lambda v, p: f"{v:g}".replace(".", ","))
fig.tight_layout(); fig.savefig("fig_esp_cond.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
X <- as.integer(econmath$work >= 15); Y <- econmath$score
m <- tapply(Y, X, mean); p <- mean(X)           #  $E(Y \mid X = 0)$ ,  $E(Y \mid X = 1)$ ,  $P(X = 1)$ 
round(m, 2); round(p, 4)
classe <- floor(econmath$colgpa / 0.25) * 0.25 + 0.125  # classes de colgpa de largeur 0,25
t <- aggregate(score ~ classe, data = cbind(econmath, classe), FUN = mean)
t$n <- as.vector(table(classe)); t <- t[t$n >= 10, ]
round(t, 2)
plot(econmath$colgpa, Y, pch = 16, cex = 0.5, col = adjustcolor("#1F4E79", 0.15),
      xlab = "Moyenne universitaire (colgpa, sur 4)", ylab = "Note finale (score, %)")
points(t$classe, t$score, pch = 16, col = "#C0502B")    # moyennes par classe
```

```
use econmath.dta, clear
generate X = (work >= 15)
tabstat score, by(X) statistics(mean n)    //  $E(Y | X = 0)$  et  $E(Y | X = 1)$ 
generate classe = floor(colgpa/0.25)*0.25 + 0.125    // classes de colgpa de largeur 0,25
egen moy = mean(score), by(classe)
egen nb = count(score), by(classe)
tabstat score if nb >= 10, by(classe) statistics(mean n)
twoway (scatter score colgpa, mcolor("31 78 121%15") msize(vsmall)) ///
      (scatter moy classe if nb >= 10, mcolor("192 80 43")), legend(off) ///
      xtitle("Moyenne universitaire (colgpa, sur 4)") ytitle("Note finale (score, %)")
graph export "fig_esp_cond_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"])      # installe si absent
import wooldridge as wd
econ = wd.data("econmath")
X = (econ["work"] >= 15).astype(int); Y = econ["score"]
m = Y.groupby(X).mean(); p = X.mean()          #  $E(Y | X = 0)$ ,  $E(Y | X = 1)$ ,  $P(X = 1)$ 
print("(1 - p) E(Y|X=0) + p E(Y|X=1) :", round((1 - p) * m[0] + p * m[1], 2))
print("E(Y) :", round(Y.mean(), 2), "| moyenne simple des deux :", round(m.mean(), 2))
print("Cov(X, Y) = p(1 - p)[E(Y|X=1) - E(Y|X=0)] :", round(p * (1 - p) * (m[1] - m[0]), 3))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
X <- as.integer(econmath$work >= 15); Y <- econmath$score
m <- tapply(Y, X, mean); p <- mean(X)           #  $E(Y | X = 0)$ ,  $E(Y | X = 1)$ ,  $P(X = 1)$ 
round(c(iterees = (1 - p) * m[[1]] + p * m[[2]], EY = mean(Y), simple = mean(m)), 2)
round(p * (1 - p) * (m[[2]] - m[[1]]), 3)      #  $Cov(X, Y) = p(1 - p)[E(Y|X=1) - E(Y|X=0)]$ 
```

```
use econmath.dta, clear
generate X = (work >= 15)
quietly summarize X
local p = r(mean)
quietly summarize score if X == 0
local m0 = r(mean)
quietly summarize score if X == 1
local m1 = r(mean)
display "(1 - p) E(Y|X=0) + p E(Y|X=1) : " %6.2f (1 - `p')*`m0' + `p'*`m1'
display "moyenne simple des deux : " %6.2f (`m0' + `m1')/2
summarize score
display "Cov(X, Y) = p(1 - p)[E(Y|X=1) - E(Y|X=0)] : " %7.3f `p'*(1 - `p')*(`m1' - `m0')
```

```
import numpy as np
x = np.array([-1, 0, 1]); prob = np.array([1, 1, 1]) / 3  # x prend -1, 0, 1 avec probabilité 1/3
u = x**2 - 2 / 3  # u dépend de x
print("E(u) :", round((prob * u).sum(), 10), "| Cov(x, u) = E(xu) - E(x)E(u) :", round((prob * x * u).sum(), 10))
print("E(u | x) pour x = -1, 0, 1 :", u.round(4))
```

Code R, covariance nulle et espérance conditionnelle

[Python](#)**R**[Stata](#)

```
x <- c(-1, 0, 1); prob <- c(1, 1, 1) / 3      # x prend -1, 0, 1 avec probabilité 1/3
u <- x^2 - 2 / 3                               # u dépend de x
round(c(Eu = sum(prob * u), Cov = sum(prob * x * u)), 10)
round(u, 4)                                   # E(u | x) pour x = -1, 0, 1
```

[◀ Retour](#)

```
clear
input x prob
-1 0.3333333333
0 0.3333333333
1 0.3333333333
end
generate u = x^2 - 2/3           // u dépend de x
generate pu = prob*u
generate pxu = prob*x*u
tabstat pu pxu, statistics(sum) //  $E(u) = 0$  et  $E(xu) = \text{Cov}(x, u) = 0$ 
list x u, clean                 //  $E(u | x)$  pour chaque valeur de x
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import pandas as pd
econ = wd.data("econmath")
quart = pd.qcut(econ["colgpa"], 4, labels=["Q1", "Q2", "Q3", "Q4"]) # quarts de colgpa
t = econ.groupby(quart, observed=True)["score"].agg(["mean", "std", "size"])
t["std"] = econ.groupby(quart, observed=True)["score"].std(ddof=0) # écart-type de la population
print(t.round(2))
print("bornes des quarts :", pd.qcut(econ["colgpa"], 4).cat.categories.tolist())
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
bornes <- quantile(econmath$colgpa, c(0, 0.25, 0.5, 0.75, 1))
quart <- cut(econmath$colgpa, bornes, include.lowest = TRUE, labels = c("Q1", "Q2", "Q3", "Q4"))
et_pop <- function(v) sqrt(mean((v - mean(v))^2))      # écart-type de la population
round(cbind(moyenne = tapply(econmath$score, quart, mean),
            ecart_type = tapply(econmath$score, quart, et_pop), n = table(quart)), 2)
bornes                                     # bornes des quarts
```

```
use econmath.dta, clear
xtile quart = colgpa, nquantiles(4)      // quarts de colgpa
egen m = mean(score), by(quart)
generate d2 = (score - m)^2
egen v = mean(d2), by(quart)             // variance de la population, division par N
generate et = sqrt(v)
tabstat score et, by(quart) statistics(mean n) // moyenne, écart-type et effectif par quart
_pctile colgpa, nquantiles(4)
display "bornes des quarts : " r(r1) " " r(r2) " " r(r3)
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge", "statsmodels"])
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
econ = wd.data("econmath")
b = smf.ols("score ~ colgpa", data=econ).fit().params           # droite des MCO
classe = (econ["colgpa"] // 0.25) * 0.25 + 0.125
t = econ.groupby(classe)["score"].agg(["mean", "size"]); t = t[t["size"] >= 10]
print(t.assign(droite=b["Intercept"] + b["colgpa"] * t.index).round(2))  # moyennes et droite
fig, ax = plt.subplots(figsize=(5.2, 3.2))
ax.scatter(econ["colgpa"], econ["score"], s=8, color="#1F4E79", alpha=0.15)
ax.scatter(t.index, t["mean"], color="#C0502B", s=30, zorder=3)      # moyennes par classe
ax.plot([0.9, 4], [b["Intercept"] + b["colgpa"] * v for v in (0.9, 4)], color="#8C8C8C", ls="--")
ax.set_xlabel("Moyenne universitaire (colgpa, sur 4)"); ax.set_ylabel("Note finale (score, %)")
ax.spines[["top", "right"]].set_visible(False)
ax.xaxis.set_major_formatter(lambda v, p: f"{v:g}".replace(".", ","))
fig.tight_layout(); fig.savefig("fig_rls1.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
m1 <- lm(score ~ colgpa, data = econmath)      # droite des MCO
classe <- floor(econmath$colgpa / 0.25) * 0.25 + 0.125
t <- aggregate(score ~ classe, data = cbind(econmath, classe), FUN = mean)
t$n <- as.vector(table(classe)); t <- t[t$n >= 10, ]
t$droite <- coef(m1)[1] + coef(m1)[2] * t$classe
round(t, 2)                                  # moyennes par classe et droite
plot(econmath$colgpa, econmath$score, pch = 16, cex = 0.5, col = adjustcolor("#1F4E79", 0.15),
      xlab = "Moyenne universitaire (colgpa, sur 4)", ylab = "Note finale (score, %)")
points(t$classe, t$score, pch = 16, col = "#C0502B")      # moyennes par classe
abline(m1, col = "#8C8C8C", lty = 2)
```

```
use econmath.dta, clear
quietly regress score colgpa           // droite des MCO
generate classe = floor(colgpa/0.25)*0.25 + 0.125
egen moy = mean(score), by(classe)
egen nb = count(score), by(classe)
generate droite = _b[_cons] + _b[colgpa]*classe
tabstat moy droite if nb >= 10, by(classe) statistics(mean) format(%9.2f)
twoway (scatter score colgpa, mcolor("31 78 121%15") msize(vsmall)) ///
      (scatter moy classe if nb >= 10, mcolor("192 80 43")) ///
      (lfit score colgpa, lcolor(gs9) lpattern(dash)), legend(off) ///
      xtitle("Moyenne universitaire (colgpa, sur 4)") ytitle("Note finale (score, %)")
graph export "fig_rls1_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
                "--disable-pip-version-check", "wooldridge"])    # installe si absent
import wooldridge as wd
econ = wd.data("econmath")
print(econ["colgpa"].describe().round(3))    # écart-type positif : RLS.3 est satisfaite
print("nombre de valeurs différentes :", econ["colgpa"].nunique())
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
summary(econmath$colgpa); round(sd(econmath$colgpa), 3)  # écart-type positif : RLS.3
length(unique(econmath$colgpa))                        # nombre de valeurs différentes
```

```
use econmath.dta, clear
summarize colgpa, detail           // écart-type positif : RLS.3 est satisfaite
quietly tabulate colgpa
display "nombre de valeurs différentes : " r(r)
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
econ = wd.data("econmath"); jt = wd.data("jtrain2")
econ["w15"] = (econ["work"] >= 15).astype(int)
print(econ.groupby("w15")[["colgpa", "mothcoll", "actmth"]].mean().round(2))
print(jt.groupby("train")[["educ", "age", "re75"]].mean().round(2)) # re75 : revenu de 1975
```

Code R, comparer les groupes

[Python](#)**R**[Stata](#)

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath"); data("jtrain2")
econmath$w15 <- as.integer(econmath$work >= 15)
moy <- function(v) tapply(v, econmath$w15, mean, na.rm = TRUE) # na.rm : actmth a 42 manquants
round(sapply(econmath[c("colgpa", "mothcoll", "actmth")], moy), 2)
round(aggregate(cbind(educ, age, re75) ~ train, data = jtrain2, FUN = mean), 2) # re75 : 1975
```

[◀ Retour](#)

Code Stata, comparer les groupes

[Python](#)[R](#)[Stata](#)

```
use econmath.dta, clear
generate w15 = (work >= 15)
tabstat colgpa mothcoll actmth, by(w15) statistics(mean) format(%9.2f)
use jtrain2.dta, clear
tabstat educ age re75, by(train) statistics(mean) format(%9.2f)    // re75 : revenu de 1975
```

[◀ Retour](#)

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge", "statsmodels"])
import wooldridge as wd
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
econ = wd.data("econmath")
res = smf.ols("score ~ colgpa", data=econ).fit().resid      # résidus des MCO
quart = pd.qcut(econ["colgpa"], 4, labels=["Q1", "Q2", "Q3", "Q4"])
print("écart-type des résidus par quart de colgpa :")
print(res.groupby(quart, observed=True).std().round(2))
fig, ax = plt.subplots(figsize=(5.2, 3))
ax.scatter(econ["colgpa"], res, s=8, color="#1F4E79", alpha=0.35)
ax.axhline(0, color="#C0502B")
ax.set_xlabel("Moyenne universitaire (colgpa, sur 4)"); ax.set_ylabel("Résidu")
ax.spines[["top", "right"]].set_visible(False)
ax.xaxis.set_major_formatter(lambda v, p: f"{v:g}".replace(".", ","))
fig.tight_layout(); fig.savefig("fig_rls5.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
res <- resid(lm(score ~ colgpa, data = econmath))      # résidus des MCO
bornes <- quantile(econmath$colgpa, c(0, 0.25, 0.5, 0.75, 1))
quart <- cut(econmath$colgpa, bornes, include.lowest = TRUE, labels = c("Q1", "Q2", "Q3", "Q4"))
round(tapply(res, quart, sd), 2)                     # écart-type des résidus par quart de colgpa
plot(econmath$colgpa, res, pch = 16, cex = 0.5, col = adjustcolor("#1F4E79", 0.35),
     xlab = "Moyenne universitaire (colgpa, sur 4)", ylab = "Résidu")
abline(h = 0, col = "#C0502B")
```

```
use econmath.dta, clear
quietly regress score colgpa
predict res, residuals           // résidus des MCO
xtile quart = colgpa, nquantiles(4)
tabstat res, by(quart) statistics(sd) format(%9.2f) // écart-type des résidus par quart
twoway scatter res colgpa, mcolor("31 78 121%35") msize(vsmall) ///
    yline(0, lcolor("192 80 43")) xtitle("Moyenne universitaire (colgpa, sur 4)") ytitle("Résidu")
graph export "fig_rls5_stata.pdf", replace
```

```
import numpy as np
import matplotlib.pyplot as plt
x = np.array([2, 4, 6, 8, 10]); y = np.array([7, 10, 11, 14, 13])
u = y - (6 + 1 * x) # erreurs, si la vraie droite est 6 + x
w = (x - x.mean()) / ((x - x.mean())**2).sum() # poids w_i
print("erreurs :", u, "| poids :", w)
b1 = 1 + (w * u).sum() # b1 = vraie pente + somme w_i u_i
print("b1 :", round(b1, 4))
print("avec d'autres erreurs (1, -1, 2, 0, 1), b1 :", 1 + (w * np.array([1, -1, 2, 0, 1])).sum())
g = np.linspace(0, 12, 50)
fig, axs = plt.subplots(1, 2, figsize=(6.4, 2.9), sharey=True)
axs[0].plot(g, 6 + g, color="#8C8C8C", ls="--"); axs[0].vlines(x, y, 6 + x, color="#8C8C8C")
axs[1].plot(g, 6.2 + 0.8 * g, color="#C0502B"); axs[1].vlines(x, y, 6.2 + 0.8 * x, color="#C0502B")
axs[0].set_title("Erreurs u (vraie droite 6 + x)", fontsize=10)
axs[1].set_title("Résidus (droite des MCO 6,2 + 0,8 x)", fontsize=10)
[a.scatter(x, y, color="#1F4E79", zorder=3) for a in axs]
[a.set_xlabel("x (heures)") for a in axs]; axs[0].set_ylabel("y (note sur 20)")
[a.set_xticks(range(0, 13, 2)) for a in axs]
[a.spines[["top", "right"]].set_visible(False) for a in axs]
fig.tight_layout(); fig.savefig("fig_jouet_vraie.pdf"); plt.show()
```

```
x <- c(2, 4, 6, 8, 10); y <- c(7, 10, 11, 14, 13)
u <- y - (6 + 1 * x)                                # erreurs, si la vraie droite est 6 + x
w <- (x - mean(x)) / sum((x - mean(x))^2)           # poids w_i
u; w
b1 <- 1 + sum(w * u)                                # b1 = vraie pente + somme w_i u_i
b1
1 + sum(w * c(1, -1, 2, 0, 1))                     # avec d'autres erreurs
par(mfrow = c(1, 2))
plot(x, y, pch = 16, col = "#1F4E79", xlim = c(0, 12), ylim = c(6, 18), main = "Erreurs u")
abline(a = 6, b = 1, col = "#8C8C8C", lty = 2); segments(x, y, x, 6 + x, col = "#8C8C8C")
plot(x, y, pch = 16, col = "#1F4E79", xlim = c(0, 12), ylim = c(6, 18), main = "Résidus")
abline(a = 6.2, b = 0.8, col = "#C0502B"); segments(x, y, x, 6.2 + 0.8 * x, col = "#C0502B")
```

```
clear
input x y u2
 2  7  1
 4 10 -1
 6 11  2
 8 14  0
10 13  1
end
generate u = y - (6 + 1*x)           // erreurs, si la vraie droite est 6 + x
generate w = (x - 6)/40              // poids w_i : moyenne de x 6, SCT_x 40
generate wu = w*u
generate wu2 = w*wu                 // u2 : d'autres erreurs
list x y u w, clean
tabstat wu wu2, statistics(sum)      // b1 = 1 + somme : 0,8 puis 1,05
generate t = 6 + x                  // vraie droite
twoway (scatter y x, mcolor("31 78 121")) (function y = 6 + x, range(0 12) lcolor(gs9) lpattern(dash)) ///
      (rspike y t x, lcolor(gs9)), legend(off) title("Erreurs u") name(A, replace)
generate f = 6.2 + 0.8*x             // droite des MCO
twoway (scatter y x, mcolor("31 78 121")) (function y = 6.2 + 0.8*x, range(0 12) lcolor("192 80 43")) ///
      (rspike y f x, lcolor("192 80 43")), legend(off) title("Résidus") name(B, replace)
graph combine A B, rows(1)
graph export "fig_jouet_vraie_stata.pdf", replace
```

```
import numpy as np
x = np.array([2, 4, 6, 8, 10]); y = np.array([7, 10, 11, 14, 13])
u = y - (6 + 1 * x) # erreurs, si la vraie droite est 6 + x
b1 = 1 + ((x - x.mean()) * u).sum() / ((x - x.mean())**2).sum()
print("moyenne des erreurs :", u.mean(), "| b1 - 1 :", round(b1 - 1, 4))
print("b0 = 6 + moyenne(u) - (b1 - 1) x moyen :", round(6 + u.mean() - (b1 - 1) * x.mean(), 4))
```

Code R, la constante de l'exemple à la main

[Python](#)**R**[Stata](#)

```
x <- c(2, 4, 6, 8, 10); y <- c(7, 10, 11, 14, 13)
u <- y - (6 + 1 * x)                                # erreurs, si la vraie droite est 6 + x
b1 <- 1 + sum((x - mean(x)) * u) / sum((x - mean(x))^2)
c(moyenne_u = mean(u), b1_moins_1 = b1 - 1)
6 + mean(u) - (b1 - 1) * mean(x)                   # b0 = 6 + moyenne(u) - (b1 - 1) x moyen
```

[◀ Retour](#)

```
clear
input x y
 2  7
 4 10
 6 11
 8 14
10 13
end
generate u = y - (6 + 1*x)           // erreurs, si la vraie droite est 6 + x
generate wu = (x - 6)/40*u
quietly summarize wu
local b1 = 1 + r(sum)
quietly summarize u
display "moyenne des erreurs : " r(mean) " | b1 - 1 : " `b1' - 1
display "b0 = 6 + moyenne(u) - (b1 - 1) x moyen : " 6 + r(mean) - (`b1' - 1)*6
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
x = wd.data("econmath")["colgpa"].head(50).to_numpy() # x fixés : 50 premiers étudiants
w = (x - x.mean()) / ((x - x.mean())**2).sum() # poids  $w_i$ , pente = somme  $w_i y_i$ 
rng = np.random.default_rng(2026)
y = 32 + 14 * x + rng.normal(0, 11, size=(5000, 50)) # 5 000 échantillons, vraie pente 14
b1 = y @ w # pente des MCO de chaque échantillon
print("moyenne des pentes :", round(b1.mean(), 3), "| écart-type :", round(b1.std(ddof=1), 3))
print("écart-type théorique 11 / racine(SCT_x) :", round(11 / np.sqrt(((x - x.mean())**2).sum()), 3))
fig, ax = plt.subplots(figsize=(5.2, 2.9))
ax.hist(b1, bins=40, color="#1F4E79", alpha=0.7)
ax.axvline(14, color="#C0502B", lw=2) # vraie pente
ax.set_xlabel("Pente estimée"); ax.set_ylabel("Nombre d'échantillons")
ax.spines[["top", "right"]].set_visible(False)
ax.xaxis.set_major_formatter(lambda v, p: f"{v:g}".replace(".", ","))
fig.tight_layout(); fig.savefig("fig_mc_sansbiais.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x <- econmath$colgpa[1:50] # x fixés : 50 premiers étudiants
w <- (x - mean(x)) / sum((x - mean(x))^2) # poids  $w_i$ , pente = somme  $w_i y_i$ 
set.seed(2026)
u <- matrix(rnorm(5000 * 50, 0, 11), nrow = 5000) # une ligne par échantillon
y <- matrix(32 + 14 * x, 5000, 50, byrow = TRUE) + u # vraie pente 14
b1 <- as.vector(y %*% w) # pente des MCO de chaque échantillon
round(c(moyenne = mean(b1), ecart_type = sd(b1)), 3)
round(11 / sqrt(sum((x - mean(x))^2)), 3) # écart-type théorique
hist(b1, breaks = 40, col = "#1F4E79", border = "white",
     xlab = "Pente estimée", ylab = "Nombre d'échantillons", main = "")
abline(v = 14, col = "#C0502B", lwd = 2) # vraie pente
```

```
use econmath.dta, clear
keep in 1/50 // x fixés : 50 premiers étudiants
keep colgpa
quietly summarize colgpa
display "écart-type théorique 11 / racine(SCT_x) : " %6.3f 11/sqrt(r(Var)*(r(N) - 1))
set seed 2026
generate y = .
tempname sim
postfile `sim' b1 using mc1, replace
forvalues r = 1/5000 {
    quietly replace y = 32 + 14*colgpa + rnormal(0, 11) // vraie pente 14
    quietly regress y colgpa
    post `sim' (_b[colgpa])
}
postclose `sim'
use mc1, clear
summarize b1 // moyenne et écart-type des 5 000 pentes
histogram b1, bin(40) frequency fcolor("31 78 121") lcolor(white) ///
    xline(14, lcolor("192 80 43") lwidth(thick)) xtitle("Pente estimée") ytitle("Nombre d'échantillons")
graph export "fig_mc_sansbiais_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
x50 = wd.data("econmath")["colgpa"].head(50).to_numpy()
x500 = np.tile(x50, 10) # 500 observations, même dispersion de x
rng = np.random.default_rng(2026)
poids = lambda x: (x - x.mean()) / ((x - x.mean())**2).sum()
tirer = lambda x: 32 + 14 * x + 4 * (x - 2.8) + rng.normal(0, 11, size=(5000, len(x)))
b50, b500 = tirer(x50) @ poids(x50), tirer(x500) @ poids(x500) #  $E(u | x) = 4(x - 2.8)$ 
print("n = 50 : moyenne", round(b50.mean(), 3), " écart-type", round(b50.std(ddof=1), 3))
print("n = 500 : moyenne", round(b500.mean(), 3), " écart-type", round(b500.std(ddof=1), 3))
fig, ax = plt.subplots(figsize=(5.6, 2.9))
ax.hist(b50, bins=40, color="#8C8C8C", alpha=0.6, density=True, label="n = 50")
ax.hist(b500, bins=40, color="#1F4E79", alpha=0.7, density=True, label="n = 500")
ax.axvline(14, color="#C0502B", lw=2); ax.text(13.8, 0.4, "vraie pente 14", color="#C0502B", ha="right")
ax.set_xlabel("Pente estimée"); ax.set_yticks([]); ax.legend(frameon=False)
ax.spines[["top", "right", "left"]].set_visible(False)
ax.xaxis.set_major_formatter(lambda v, p: f"{v:g}".replace(".", ","))
fig.tight_layout(); fig.savefig("fig_mc_biais.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x50 <- econmath$colgpa[1:50]
x500 <- rep(x50, 10) # 500 observations, même dispersion de x
set.seed(2026)
pentes <- function(x) {
  w <- (x - mean(x)) / sum((x - mean(x))^2)
  e <- matrix(rnorm(5000 * length(x), 0, 11), nrow = 5000)
  y <- sweep(e, 2, 32 + 14 * x + 4 * (x - 2.8), "+") #  $E(u | x) = 4(x - 2,8)$ 
  as.vector(y %*% w)
}
b50 <- pentes(x50); b500 <- pentes(x500)
round(rbind(n50 = c(mean(b50), sd(b50)), n500 = c(mean(b500), sd(b500))), 3)
hist(b500, breaks = 40, freq = FALSE, col = adjustcolor("#1F4E79", 0.7), border = "white",
     xlim = c(8, 28), xlab = "Pente estimée", main = "")
hist(b50, breaks = 40, freq = FALSE, col = adjustcolor("#8C8C8C", 0.5), border = "white", add = TRUE)
abline(v = 14, col = "#C0502B", lwd = 2) # vraie pente
```

```
use econmath.dta, clear
keep in 1/50
keep colgpa
expand 10                                // 500 observations, même dispersion de x
generate premiers = (_n <= 50)           // les 50 premières lignes : les x d'origine
set seed 2026
generate y = .
tempname sim
postfile `sim' b50 b500 using mc2, replace
forvalues r = 1/5000 {
    quietly replace y = 32 + 14*colgpa + 4*(colgpa - 2.8) + rnormal(0, 11)    //  $E(u | x) = \frac{4}{5} (x - 2,8)$ 
    quietly regress y colgpa if premiers
    local a = _b[colgpa]
    quietly regress y colgpa
    post `sim' (`a') (_b[colgpa])
}
postclose `sim'
use mc2, clear
summarize b50 b500
twoway (histogram b50, bin(40) color(gs12)) (histogram b500, bin(40) color("31 78 121%70")), ///
    xline(14, lcolor("192 80 43") lwidth(thick)) legend(order(1 "n = 50" 2 "n = 500")) xtitle("Pente estimée")
graph export "fig_mc_biais_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
import matplotlib.pyplot as plt
x50 = wd.data("econmath")["colgpa"].head(50).to_numpy()
plans = {"référence": (x50, 11), "n = 200": (np.tile(x50, 4), 11),
         "x deux fois moins dispersé": (x50.mean() + 0.5 * (x50 - x50.mean()), 11), "sigma = 5,5": (x50, 5.5)}
rng = np.random.default_rng(2026)
sct = lambda x: ((x - x.mean())**2).sum()
simul = lambda x, s: (32 + 14 * x + rng.normal(0, s, size=(5000, len(x)))) @ ((x - x.mean()) / sct(x))
b = {k: simul(x, s) for k, (x, s) in plans.items()}
[print(f"{k:27s} théorique {s / np.sqrt(sct(x)):.3f} | simulé {b[k].std(ddof=1):.3f}")
 for k, (x, s) in plans.items())
fig, axs = plt.subplots(1, 3, figsize=(9, 2.6), sharex=True, sharey=True)
[a.hist(b[k], bins=40, range=(-6, 34), color="#1F4E79", alpha=0.7) for a, k in zip(axs, list(plans)[:3])]
[a.axvline(14, color="#C0502B", lw=1.5) for a in axs]
[a.set_title(k, fontsize=10) for a, k in zip(axs, list(plans)[:3])]
[a.set_xlabel("Pente estimée") for a in axs]; [a.set_yticks([]) for a in axs]
[a.spines[["top", "right", "left"]].set_visible(False) for a in axs]
fig.tight_layout(); fig.savefig("fig_mc_variance.pdf"); plt.show()
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x50 <- econmath$colgpa[1:50]
plans <- list(reference = list(x50, 11), n200 = list(rep(x50, 4), 11),
              x_resserre = list(mean(x50) + 0.5 * (x50 - mean(x50)), 11), sigma5.5 = list(x50, 5.5))
set.seed(2026)
simul <- function(x, s) {
  w <- (x - mean(x)) / sum((x - mean(x))^2)
  as.vector(sweep(matrix(rnorm(5000 * length(x), 0, s), nrow = 5000), 2, 32 + 14 * x, "+") %*% w)
}
b <- lapply(plans, function(p) simul(p[[1]], p[[2]]))
theo <- sapply(plans, function(p) p[[2]] / sqrt(sum((p[[1]] - mean(p[[1]]))^2)))
round(rbind(theorique = theo, simule = sapply(b, sd)), 3)
par(mfrow = c(1, 3))
for (k in 1:3) {
  hist(b[[k]], breaks = 40, xlim = c(-6, 34), col = "#1F4E79", border = "white",
       main = names(plans)[k], xlab = "Pente estimée")
  abline(v = 14, col = "#C0502B", lwd = 1.5)
}
```

```
use econmath.dta, clear
keep in 1/50
quietly summarize colgpa
generate xr = r(mean) + 0.5*(colgpa - r(mean))    // x deux fois moins dispersé
expand 4                                          // 200 observations : chaque x répété 4 fois
generate premiers = (_n <= 50)                  // les 50 premières lignes : les x d'origine
set seed 2026
generate y = .
postfile sim ref n200 resserre sigma55 using mc3, replace
forvalues r = 1/5000 {
    quietly replace y = 32 + 14*colgpa + rnormal(0, 11)
    quietly regress y colgpa if premiers // référence, n = 50
    local a = _b[colgpa]
    quietly regress y colgpa              // n = 200
    local b = _b[colgpa]
    quietly replace y = 32 + 14*xr + rnormal(0, 11)
    quietly regress y xr if premiers      // x deux fois moins dispersé
    local c = _b[xr]
    quietly replace y = 32 + 14*colgpa + rnormal(0, 5.5)
    quietly regress y colgpa if premiers // sigma = 5,5
    post sim (`a') (`b') (`c') (_b[colgpa])
}
postclose sim
use mc3, clear
tabstat ref n200 resserre sigma55, statistics(mean sd) format(%9.3f)    // théoriques 2,606 1,303 5,213 1,303
```

```
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.default_rng(2026)
xl = np.linspace(1, 5, 20); xe = 3 + 0.25 * (xl - 3)    # x dispersés, puis x resserrés
e = rng.normal(0, 1, size=(20, 20))                  # 20 échantillons d'erreurs, les mêmes à gauche et à droite
g = np.linspace(0, 6, 20)
droite = lambda x, u: np.polyval(np.polyfit(x, 2 + x + u, 1), g)    # droite des MCO
fig, axs = plt.subplots(1, 2, figsize=(7, 2.8), sharey=True)
[axs[0].plot(g, droite(xl, u), color="#1F4E79", alpha=0.35) for u in e]
[axs[1].plot(g, droite(xe, u), color="#1F4E79", alpha=0.35) for u in e]
[a.plot(g, 2 + g, color="#C0502B", lw=2) for a in axs]    # vraie droite 2 + x
axs[0].set_title("x entre 1 et 5", fontsize=10); axs[1].set_title("x entre 2,5 et 3,5", fontsize=10)
[a.axvspan(x.min(), x.max(), color="#8C8C8C", alpha=0.15) for a, x in zip(axs, [xl, xe])]
[a.set_xlabel("x") for a in axs]; axs[0].set_ylabel("y"); axs[0].set_ylim(-4, 12)
[a.spines[["top", "right"]].set_visible(False) for a in axs]
fig.tight_layout(); fig.savefig("fig_eventail.pdf"); plt.show()
print("écarts-types théoriques :", round(1 / np.sqrt(((xl - 3)**2).sum()), 3),
      round(1 / np.sqrt(((xe - 3)**2).sum()), 3))
```

```
set.seed(2026)
xl <- seq(1, 5, length.out = 20); xe <- 3 + 0.25 * (xl - 3)  # x dispersés, puis x resserrés
e <- matrix(rnorm(20 * 20), nrow = 20)  # 20 échantillons d'erreurs, les mêmes à gauche et à droite
par(mfrow = c(1, 2))
for (x in list(xl, xe)) {
  plot(NA, xlim = c(0, 6), ylim = c(-4, 12), xlab = "x", ylab = "y",
       main = paste("x entre", min(x), "et", max(x)))
  rect(min(x), -5, max(x), 13, col = adjustcolor("#8C8C8C", 0.15), border = NA)
  for (r in 1:20) abline(lm(2 + x + e[r, ] ~ x), col = adjustcolor("#1F4E79", 0.35))
  abline(a = 2, b = 1, col = "#C0502B", lwd = 2)  # vraie droite 2 + x
}
round(c(1 / sqrt(sum((xl - 3)^2)), 1 / sqrt(sum((xe - 3)^2))), 3)  # écarts-types théoriques
```

```
clear
set obs 20
generate xl = 1 + 4*(_n - 1)/19           // x dispersés entre 1 et 5
generate xe = 3 + 0.25*(xl - 3)          // x resserrés entre 2,5 et 3,5
set seed 2026
generate e = .
generate y = .
local gl ""
local ge ""
forvalues r = 1/20 {
    quietly replace e = rnormal(0, 1)    // mêmes erreurs à gauche et à droite
    quietly replace y = 2 + xl + e
    quietly regress y xl
    local gl `gl' (function y = `_b[_cons]' + `_b[xl]'*x, range(0 6) lcolor(navy%35))
    quietly replace y = 2 + xe + e
    quietly regress y xe
    local ge `ge' (function y = `_b[_cons]' + `_b[xe]'*x, range(0 6) lcolor(navy%35))
}
twoway `gl' (function y = 2 + x, range(0 6) lcolor(orange) lwidth(thick)), ///
    legend(off) yscale(range(-4 12)) title("x entre 1 et 5") name(L, replace)
twoway `ge' (function y = 2 + x, range(0 6) lcolor(orange) lwidth(thick)), ///
    legend(off) yscale(range(-4 12)) title("x entre 2,5 et 3,5") name(E, replace)
graph combine L E, rows(1) ycommon
graph export "fig_eventail_stata.pdf", replace
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
x = wd.data("econmath")["colgpa"].head(50).to_numpy()
w = (x - x.mean()) / ((x - x.mean())**2).sum()
s = 3 + 14 * np.abs(x - 2.8) # écart-type de u, variable selon x
print("moyenne des variances de u :", round((s**2).mean(), 1))
rng = np.random.default_rng(2026)
b1 = (32 + 14 * x + rng.normal(0, 1, size=(5000, 50)) * s) @ w
print("moyenne des pentes :", round(b1.mean(), 3), "| écart-type simulé :", round(b1.std(ddof=1), 3))
print("vrai écart-type, racine(somme w_i^2 s_i^2) :", round(np.sqrt((w**2 * s**2).sum()), 3))
sct = ((x - x.mean())**2).sum()
print("formule usuelle, racine(moyenne des s_i^2 / SCT_x) :", round(np.sqrt((s**2).mean() / sct), 3))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x <- econmath$colgpa[1:50]
w <- (x - mean(x)) / sum((x - mean(x))^2)
s <- 3 + 14 * abs(x - 2.8)           # écart-type de u, variable selon x
round(mean(s^2), 1)                 # moyenne des variances de u
set.seed(2026)
u <- sweep(matrix(rnorm(5000 * 50), nrow = 5000), 2, s, "*")
b1 <- as.vector(sweep(u, 2, 32 + 14 * x, "+") %*% w)
round(c(moyenne = mean(b1), ecart_type_simule = sd(b1)), 3)
round(c(vrai = sqrt(sum(w^2 * s^2)), formule_usuelle = sqrt(mean(s^2) / sum((x - mean(x))^2))), 3)
```

```
use econmath.dta, clear
keep in 1/50
keep colgpa
generate s = 3 + 14*abs(colgpa - 2.8)    // écart-type de u, variable selon x
quietly summarize colgpa
local sct = r(Var)*(r(N) - 1)
generate w = (colgpa - r(mean))/`sct'    // poids w_i
generate s2 = s^2
generate w2s2 = w^2*s2
quietly summarize s2
local ms2 = r(mean)
display "moyenne des variances de u : " %6.1f `ms2'
quietly summarize w2s2
display "vrai écart-type : " %6.3f sqrt(r(sum)) " | formule usuelle : " %6.3f sqrt(`ms2'/'sct')
set seed 2026
generate y = .
postfile sim b1 using mc4, replace
forvalues r = 1/5000 {
    quietly replace y = 32 + 14*colgpa + s*rnormal(0, 1)
    quietly regress y colgpa
    post sim (_b[colgpa])
}
postclose sim
use mc4, clear
summarize b1                            // moyenne et écart-type simulé
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"])    # installe si absent
import wooldridge as wd
import numpy as np
x = wd.data("econmath")["colgpa"].head(50).to_numpy()
sct = ((x - x.mean())**2).sum(); w = (x - x.mean()) / sct
rng = np.random.default_rng(2026)
y = 32 + 14 * x + rng.normal(0, 11, size=(5000, 50))
b1 = y @ w; b0 = y.mean(axis=1) - b1 * x.mean()
scr = ((y - b0[:, None] - b1[:, None] * x)**2).sum(axis=1)    # SCR de chaque échantillon
print("moyenne de SCR / (n - 2) :", round((scr / 48).mean(), 2))
print("moyenne de SCR / n :", round((scr / 50).mean(), 2))
print("vraie variance sigma^2 :", 121, "| 121 x 48 / 50 :", 121 * 48 / 50)
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x <- econmath$colgpa[1:50]
sct <- sum((x - mean(x))^2); w <- (x - mean(x)) / sct
set.seed(2026)
y <- matrix(32 + 14 * x, 5000, 50, byrow = TRUE) + matrix(rnorm(5000 * 50, 0, 11), nrow = 5000)
b1 <- as.vector(y %*% w); b0 <- rowMeans(y) - b1 * mean(x)
scr <- rowSums((y - b0 - outer(b1, x))^2)      # SCR de chaque échantillon
round(c(SCR_sur_n_moins_2 = mean(scr / 48), SCR_sur_n = mean(scr / 50)), 2)
c(sigma2 = 121, biais_attendu = 121 * 48 / 50)
```

```
use econmath.dta, clear
keep in 1/50
keep colgpa
set seed 2026
generate y = .
tempname sim
postfile `sim' scr using mc5, replace
forvalues r = 1/5000 {
    quietly replace y = 32 + 14*colgpa + rnormal(0, 11)
    quietly regress y colgpa
    post `sim' (e(rss))           // SCR de l'échantillon
}
postclose `sim'
use mc5, clear
generate s2_n2 = scr/48           // SCR / (n - 2)
generate s2_n = scr/50           // SCR / n
tabstat s2_n2 s2_n, statistics(mean) format(%9.2f)
display "vraie variance 121 | 121 x 48 / 50 = " 121*48/50
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge"]) # installe si absent
import wooldridge as wd
import numpy as np
x = wd.data("econmath")["colgpa"].head(50).to_numpy()
sct = ((x - x.mean())**2).sum(); w = (x - x.mean()) / sct
rng = np.random.default_rng(2026)
y = 32 + 14 * x + rng.normal(0, 11, size=(5000, 50))
b1 = y @ w; b0 = y.mean(axis=1) - b1 * x.mean()
scr = ((y - b0[:, None] - b1[:, None] * x)**2).sum(axis=1) # SCR de chaque échantillon
se = np.sqrt(scr / 48 / sct) # erreur-type de la pente, un par échantillon
print("moyenne des erreurs-types :", round(se.mean(), 3), "| écart-type des pentes :", round(b1.std(ddof=1), 3))
print("écart-type théorique 11 / racine(SCT_x) :", round(11 / np.sqrt(sct), 3))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
x <- econmath$colgpa[1:50]
sct <- sum((x - mean(x))^2); w <- (x - mean(x)) / sct
set.seed(2026)
y <- matrix(32 + 14 * x, 5000, 50, byrow = TRUE) + matrix(rnorm(5000 * 50, 0, 11), nrow = 5000)
b1 <- as.vector(y %*% w); b0 <- rowMeans(y) - b1 * mean(x)
scr <- rowSums((y - b0 - outer(b1, x))^2)      # SCR de chaque échantillon
se <- sqrt(scr / 48 / sct)                    # erreur-type de la pente, un par échantillon
round(c(moyenne_se = mean(se), ecart_type_pentes = sd(b1)), 3)
round(11 / sqrt(sct), 3)                     # écart-type théorique
```

```
use econmath.dta, clear
keep in 1/50
keep colgpa
quietly summarize colgpa
display "écart-type théorique 11 / racine(SCT_x) : " %6.3f 11/sqrt(r(Var)*(r(N) - 1))
set seed 2026
generate y = .
tempname sim
postfile `sim' b1 se using mc6, replace
forvalues r = 1/5000 {
    quietly replace y = 32 + 14*colgpa + rnormal(0, 11)
    quietly regress y colgpa
    post `sim' (_b[colgpa]) (_se[colgpa])    // pente et erreur-type
}
postclose `sim'
use mc6, clear
tabstat se b1, statistics(mean sd) format(%9.3f)    // moyenne des se, écart-type des pentes
```

```
import subprocess, sys
subprocess.run([sys.executable, "-m", "pip", "install", "-q",
               "--disable-pip-version-check", "wooldridge", "statsmodels"])
import wooldridge as wd
import numpy as np
import statsmodels.formula.api as smf
econ = wd.data("econmath")
m1 = smf.ols("score ~ colgpa", data=econ).fit()
n = int(m1.nobs); x = econ["colgpa"]; sct = ((x - x.mean())**2).sum()
s2 = m1.ssr / (n - 2) # sigma chapeau au carré = SCR / (n - 2)
print("SCR :", round(m1.ssr, 1), "| sigma chapeau :", round(np.sqrt(s2), 3), "| SCT_x :", round(sct, 2))
print("se(b1) = sigma chapeau / racine(SCT_x) :", f"{np.sqrt(s2 / sct):.4f}")
print("moyenne des x au carré :", round((x**2).mean(), 3))
print("se(b0) :", f"{np.sqrt(s2 * (x**2).sum() / (n * sct)):.4f}")
print("logiciel :", m1.bse.round(3).tolist(), "| sigma chapeau :", round(np.sqrt(m1.scale), 3))
```

```
if (!requireNamespace("wooldridge", quietly = TRUE)) install.packages("wooldridge")
library(wooldridge)
data("econmath")
m1 <- lm(score ~ colgpa, data = econmath)
n <- nobs(m1); x <- econmath$colgpa; sct <- sum((x - mean(x))^2)
s2 <- sum(resid(m1)^2) / (n - 2) # sigma chapeau au carré = SCR / (n - 2)
round(c(SCR = sum(resid(m1)^2), sigma_chapeau = sqrt(s2), SCT_x = sct), 3)
round(mean(x^2), 3) # moyenne des x au carré
round(c(se_b1 = sqrt(s2 / sct), se_b0 = sqrt(s2 * sum(x^2) / (n * sct))), 4)
round(summary(m1)$coefficients[, 2], 4); round(summary(m1)$sigma, 3) # logiciel
```

```
use econmath.dta, clear
regress score colgpa
local s2 = e(rss)/(e(N) - 2)           // sigma chapeau au carré = SCR / (n - 2)
quietly summarize colgpa
local sct = r(Var)*(r(N) - 1)
generate x2 = colgpa^2
quietly summarize x2
display "moyenne des x au carré : " %6.3f r(mean)
display "SCR : " %9.1f e(rss) " | sigma chapeau : " %6.3f sqrt(`s2') " | SCT_x : " %7.2f `sct'
display "se(b1) : " %6.4f sqrt(`s2'/`sct') " | se(b0) : " %6.4f sqrt(`s2'*r(sum)/(e(N)*`sct'))
display "Root MSE de Stata (sigma chapeau) : " %6.3f e(rmse)
```