

Sesión 16 (A) — El experimento de Montecarlo: ¿de verdad está centrado $\hat{\beta}_2$?

Marcos Bujosa

Descripción de la práctica

La lección 11 demostró dos cosas sobre $\hat{\beta}_2$: que es *insesgado* ($E(\hat{\beta}_2) = \beta_2$) y que su varianza condicional vale exactamente

$$Var[\hat{\beta}_2 | \mathbf{X}] = \frac{\sigma^2}{\sum_i (X_i - \bar{X})^2}.$$

Son dos afirmaciones sobre *todas las muestras que podríamos haber observado*, y por eso resultan difíciles de creer con una sola muestra delante: por definición, de ese conjunto de muestras posibles solo vemos una.

Esta práctica hace trampa —una trampa legítima y muy instructiva—: *fabrica* ese conjunto de muestras posibles. Inventamos un mundo artificial en el que conocemos β_1 , β_2 y σ^2 porque los hemos elegido nosotros; generamos cientos de muestras de ese mundo; estimamos $\hat{\beta}_2$ en cada una; y comparamos lo que sale con lo que la lección 11 predice. Es lo que se conoce como un *experimento de Montecarlo*.

Objetivos

1. Construir un mundo artificial con parámetros conocidos, anclado a un ajuste real sobre `hprice2`.
2. Generar *una* muestra simulada a mano y estimarla, para ver con claridad qué es lo que se va a repetir después.
3. Aprender la mecánica completa del comando `loop` de Gretl: cómo repetir un experimento, cómo guardar resultados de cada repetición y cómo analizarlos después.
4. Comprobar que la media de las estimaciones se acerca a β_2 (insesgadez) y que su varianza se acerca a la fórmula de la lección 11.
5. Ver el histograma de $\hat{\beta}_2$ y empezar a preguntarse por su *forma* —una pregunta que esta práctica deliberadamente deja sin responder.

Nota sobre el loop. Ya usamos este comando una vez, en la práctica de verificación geométrica que siguió a la lección 5, pero allí lo presentamos como una “caja negra”: servía para repetir una comprobación muchas veces, sin explicar su mecánica. Hoy toca abrir la caja.

Actividad 1 - Un mundo artificial con parámetros conocidos

Para simular necesitamos un mundo del que conozcamos la verdad. Podríamos inventar los números de la nada, pero es más instructivo tomarlos prestados de un ajuste real: así el mundo artificial se parece al que ya hemos estado estudiando.

Partimos de `hprice2` y de la regresión `price~rooms` que ya estimamos en la práctica que siguió a la lección 8. Sus resultados dejan de ser “estimaciones” para convertirse, *por decreto*, en los parámetros verdaderos de nuestro mundo inventado.

o bien teclee en línea de comandos:

Licencia: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

```
open hprice2.gdt --quiet

ols price const rooms --quiet
scalar b1_true = $coeff(const)          # beta_1 "verdadero"
scalar b2_true = $coeff(rooms)          # beta_2 "verdadero"
scalar sigma   = sqrt($ess/$df)         # sigma "verdadera" (raiz de la cuasivarianza)
scalar n       = $nobs
scalar Sxx     = sum((rooms - mean(rooms))^2)
scalar var_teo = sigma^2 / Sxx           # formula de la leccion 11
```

```
printf "n           = %d\n", n
printf "beta_1      = %.4f\n", b1_true
printf "beta_2      = %.4f\n", b2_true
printf "sigma       = %.4f\n", sigma
printf "Sxx        = %.4f\n", Sxx
printf "Var[b2 | X] = sigma^2/Sxx = %.4f\n", var_teo
printf "desv. tipica = %.4f\n", sqrt(var_teo)
```

```
n           = 506
beta_1      = -34796,2008
beta_2      = 9119,5484
sigma       = 6620,8652
Sxx         = 249,2872
Var[b2 | X] = sigma^2/Sxx = 175844,7988
desv. tipica = 419,3385
```

Una notación nueva, solo para el guion: Sxx

En el código aparece un nombre que no hemos usado en ninguna lección teórica: **Sxx**. No es un concepto nuevo, solo una abreviatura cómoda para el denominador de la fórmula de la varianza:

$$Sxx = \sum_i (X_i - \bar{X})^2.$$

Conviene reconocerla bajo sus otros dos disfraces, que sí son del curso:

- Es el *cuadrado de la norma euclídea del vector en desviaciones del regresor*, $\|\mathbf{x} - \mu_{\mathbf{x}}\mathbf{1}\|_e^2$ —exactamente la lectura geométrica que daba la nota a pie de página de la lección 11—.
- Y es n veces la varianza del regresor con el convenio del curso: $Sxx = n S^2$, con $S^2 = \frac{1}{n} \sum_i (X_i - \bar{X})^2$. Por eso la fórmula de la lección 11 se puede escribir indistintamente σ^2/Sxx o $\sigma^2/(nS^2)$.

La usamos aquí porque en un guion resulta más legible que cualquiera de las dos expresiones completas, pero no es notación del curso: fuera de estas prácticas, escriba $\sum_i (X_i - \bar{X})^2$ o nS^2 .

Tres observaciones sobre lo que acabamos de hacer

- **rooms** *no se toca*. El vector de datos del regresor se queda exactamente como está, con sus 506 valores observados. Es el $\mathbf{x}$ de la lección 11, y permanecerá fijo durante todo el experimento. Esto es lo que significa “condicionar en $\mathbf{X}$ ”; en la práctica siguiente veremos qué pasa si dejamos de hacerlo.
- σ se toma como la raíz de $SRC/(n - k)$, es decir, de la cuasivarianza de los residuos $\mathbf{s}^2$ que definimos en la lección 11. Aquí no la usamos como *estimación* de nada: la usamos como el valor *verdadero* de nuestro mundo inventado.
- **var_teo** es la predicción de la lección 11. Anótela: es el número contra el que vamos a contrastar todo lo que sigue.

Actividad 2 - Una sola muestra simulada, a mano

Antes de repetir nada mil veces, conviene hacer *una* vez, despacio, lo que después se va a repetir.

Una “muestra de nuestro mundo” se construye en dos pasos: primero la parte sistemática $\beta_1 \mathbf{1} + \beta_2 \mathbf{x}$, que es siempre la misma; después una perturbación aleatoria nueva, que es lo único que cambia de una muestra a otra. Sumadas dan un regresando simulado.

o bien teclee en línea de comandos:

```
set seed 20261030                                # para que el resultado sea reproducible

series u_sim = normal(0, sigma)                    # perturbacion nueva
series y_sim = b1_true + b2_true*rooms + u_sim      # muestra simulada

ols y_sim const rooms
printf "\nbeta_2 verdadero = %.4f\n", b2_true
printf "beta_2 estimado   = %.4f\n", $coeff(rooms)
printf "diferencia        = %.4f\n", $coeff(rooms) - b2_true
```

Modelo 1: MCO, usando las observaciones 1-506

Variable dependiente: y_sim

	coeficiente	Desv. típica	Estadístico t	valor p
const	-34761,1	2711,25	-12,82	9,11e-33 ***
rooms	9140,64	428,784	21,32	2,34e-72 ***

Media de la vble. dep.	22679,15	D.T. de la vble. dep.	9326,634
Suma de cuad. residuos	2,31e+10	D.T. de la regresión	6769,993
R-cuadrado	0,474145	R-cuadrado corregido	0,473102
F(1, 504)	454,4397	Valor p (de F)	2,34e-72
Log-verosimilitud	-5180,030	Criterio de Akaike	10364,06
Criterio de Schwarz	10372,51	Crit. de Hannan-Quinn	10367,38

```
beta_2 verdadero = 9119,5484
beta_2 estimado  = 9140,6352
diferencia       = 21,0868
```

Lo que debe observar

La estimación *no* coincide con el valor verdadero. No es un fallo: es lo que la lección 11 predice. Recuerde la identidad central de aquella lección:

$$\hat{\beta}_2 = \beta_2 \mathbf{1} + \sum_i W_i U_i.$$

El segundo sumando no se anula en una muestra cualquiera; es una variable aleatoria con media cero pero dispersión no nula. Lo que acabamos de obtener es β_2 más una realización concreta de ese término.

La pregunta interesante no es, por tanto, “¿por qué no ha salido exacto?”, sino “¿cómo de grande suele ser ese error, y se compensa en promedio?”. Y a esa pregunta no se puede responder con una sola muestra. Hay que repetir.

Actividad 3 - El comando loop: repetir el experimento muchas veces

Aquí está la novedad técnica de la sesión. El comando `loop` de Gretl repite un bloque de instrucciones un número dado de veces. Su forma general es

```
loop NUMERO_DE_REPETICIONES --progressive --quiet
... instrucciones que se repiten ...
endloop
```

Dos opciones merecen comentario:

- `--quiet` evita que Gretl imprima la salida completa de cada una de las repeticiones (imagine 1000 tablas de regresión seguidas).
- `--progressive` es la que hace el trabajo interesante: le dice a Gretl que va a acumular resultados a lo largo de las repeticiones, en lugar de olvidarlos al terminar cada una. Es lo que permite usar `store` dentro del bucle.

Y `store` es la pieza que convierte el experimento en datos analizables:

```
store "@workdir/fichero.gdt" escalar1 escalar2
```

Dentro de un bucle `--progressive`, `store` no sobrescribe el fichero en cada vuelta: va *añadiendo* una fila por repetición. Al terminar el bucle tenemos un conjunto de datos nuevo en el que **cada observación es una repetición del experimento** y cada variable es un resultado que hemos decidido guardar. Después basta con abrirlo, fuera del bucle, y tratarlo como cualquier otro conjunto de datos.

o bien teclee en línea de comandos:

```
scalar R = 2000                                # numero de replicas

set seed 20261030
loop R --progressive --quiet
  series u = normal(0, sigma)
  series ysim = b1_true + b2_true*rooms + u
  ols ysim const rooms --quiet
  scalar bihat = $coeff(const)
  scalar b2hat = $coeff(rooms)
  store "@workdir/replicas.gdt" bihat b2hat
endloop
```

```
printf "Hecho: %d replicas guardadas en replicas.gdt\n", R
```

Hecho: 2000 replicas guardadas en replicas.gdt

Merece la pena detenerse un segundo en *qué cambia y qué no* dentro del bucle. Cambia una sola cosa: la perturbación `u`, que se sortea de nuevo en cada vuelta. No cambian ni `rooms`, ni `b1_true`, ni `b2_true`, ni `sigma`. Estamos generando, literalmente, las muestras que “podrían haber salido” del mismo mundo con el mismo regresor.

Actividad 4 - ¿Se cumple la insesgadez? ¿Y la fórmula de la varian-za?

Abrimos ahora el fichero de réplicas y lo tratamos como lo que es: un conjunto de datos con R observaciones, donde cada observación es una estimación que *podríamos* haber obtenido.

Atención a un detalle práctico: al abrir un conjunto de datos nuevo, Gretl descarta los escalares que teníamos definidos. Por eso imprimimos los valores teóricos *antes* de abrir el fichero de réplicas, y los empíricos después.

o bien teclee en línea de comandos:

```
printf "=== LO QUE PREDICE LA LECCION 11 ===\n"
printf "beta_2 verdadero      = %.4f\n", b2_true
printf "Var[b2 | X] = sigma^2/Sxx = %.4f\n", var_teo
printf "desviacion tipica     = %.4f\n\n", sqrt(var_teo)
```

```
open "@workdir/replicas.gdt" --quiet
```

```
printf "=== LO QUE SALE DE %d REPLICAS SIMULADAS ===\n", $nobs
printf "media de b2hat        = %.4f\n", mean(b2hat)
printf "varianza de b2hat     = %.4f\n", sum((b2hat-mean(b2hat))^2)/$nobs
printf "desviacion tipica     = %.4f\n", sd(b2hat)
```

```
=== LO QUE PREDICE LA LECCION 11 ===
beta_2 verdadero      = 9119,5484
Var[b2 | X] = sigma^2/Sxx = 175844,7988
desviacion tipica     = 419,3385

=== LO QUE SALE DE 2000 REPLICAS SIMULADAS ===
media de b2hat        = 9115,1726
varianza de b2hat     = 173769,1343
desviacion tipica     = 416,9605
```

Lo que debe observar

Compare las dos mitades de la salida, línea por línea:

1. *Insesgadez*. La media de las estimaciones debe quedar muy cerca de β_2 . No igual: con un número finito de réplicas, la media muestral de las estimaciones es a su vez una variable aleatoria. Pero la distancia debe ser pequeña comparada con la dispersión de las propias estimaciones.
2. *Varianza*. La varianza empírica de las R estimaciones debe quedar cerca de $\sigma^2 / \sum_i (X_i - \bar{X})^2$. Esta es la comprobación más llamativa de la práctica: una fórmula obtenida con álgebra y esperanzas condicionales, sin simular nada, acierta el resultado de un experimento numérico.
3. *La insesgadez no dice nada sobre la varianza*. Fíjese en que el error típico de una estimación individual es mucho mayor que la distancia entre la media de las estimaciones y β_2 . Estar bien centrado no equivale a ser preciso: son las dos preguntas distintas que la lección 11 planteó desde su primera transparencia.

Nota sobre la reproducibilidad. La instrucción `set seed` fija la secuencia de números pseudoaleatorios, de modo que este guion produce siempre los mismos números. Si la comenta y lo ejecuta dos veces, obtendrá resultados ligeramente distintos —y ese es, en sí mismo, un buen experimento: la propia media de las réplicas tiene su dispersión.

Actividad 5 - El histograma de $\hat{\beta}_2$

Tenemos 2000 estimaciones. Veamos cómo se reparten.
o bien teclee en línea de comandos:

```
freq b2hat --normal --plot=display
```

La opción `--normal` superpone al histograma la densidad de una normal con la misma media y la misma varianza que los datos, además de contrastar formalmente si la distribución es normal.

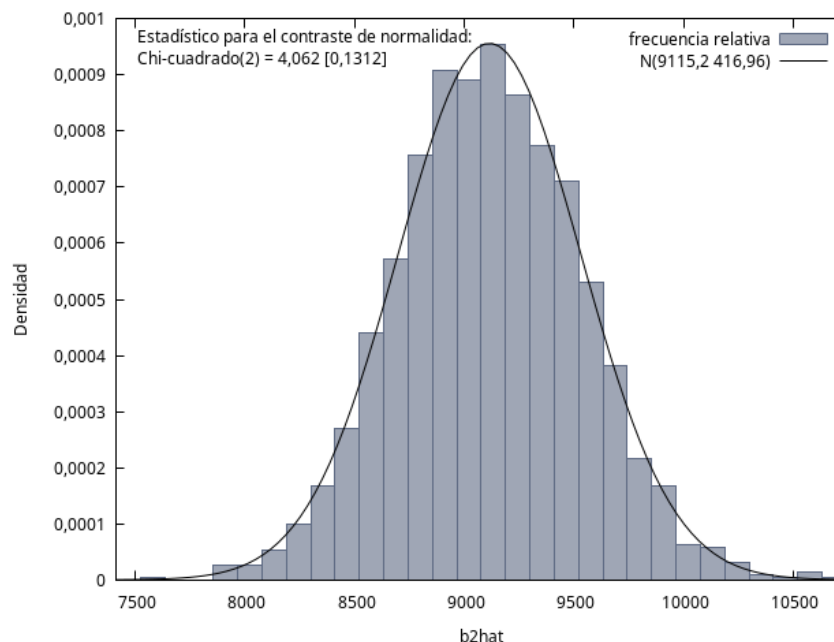


Figura 1: Distribución de $\hat{\beta}_2$ sobre 2000 muestras simuladas del mismo mundo, con una densidad normal superpuesta.

Una pregunta que esta práctica NO responde

El histograma tiene una forma muy reconocible. Y es tentador dar el paso siguiente: si conocemos la media de $\hat{\beta}_2$, su varianza, y *además* su forma, entonces podríamos calcular la probabilidad de que $\hat{\beta}_2$ caiga en tal o cual intervalo, y con eso construir intervalos de confianza y contrastes de hipótesis.

Conviene resistirse a ese paso, por dos motivos:

- El primero es de método: lo que vemos es la forma que sale *en este experimento concreto*, en el que hemos generado las perturbaciones con una distribución normal porque así lo hemos decidido nosotros. No hemos demostrado nada sobre la forma; la hemos impuesto al construir el mundo. En la práctica C veremos qué ocurre si la cambiamos.
- El segundo es de alcance: la lección 11 demostró la media y la varianza, y se detuvo ahí explícitamente. La *forma* de la distribución de $\hat{\beta}_2$ —y lo que hace falta suponer para conocerla— es el asunto de la lección 12.

Apéndice - La matriz de varianzas y covarianzas

Material complementario. Este apéndice va más allá de lo que da tiempo a hacer en clase; queda aquí para quien quiera completarlo por su cuenta, con el mismo estatus que el apéndice matricial de la práctica sobre datos simulados que siguió a la lección 10.

La última transparencia de la lección 11 enunció, *sin demostrarla*, la generalización matricial de lo que sí demostramos:

$$\text{Var}[\hat{\beta} | \mathbf{X}] = \sigma^2(\mathbf{X}^\top \mathbf{X})^{-1}.$$

Podemos verificarla numéricamente con el mismo experimento, guardando también $\hat{\beta}_1$ y comparando la matriz de varianzas y covarianzas *empírica* de las réplicas con la que predice la fórmula.

o bien teclee en línea de comandos:

```
open hprice2.gdt --quiet # recuperamos los regresores
ols price const rooms --quiet
scalar sigma = sqrt($ess/$df)
```

```
matrix X = {const, rooms}
matrix invXTX = invpd(X'X)
matrix Vteo = sigma^2 * invXTX
printf "Matriz sigma^2 (X'X)^{-1} predicha por la leccion 11:\n"
print Vteo
```

```
matrix B = {b1hat, b2hat}
matrix Vemp = mcov(B)
printf "\nMatriz de varianzas y covarianzas EMPIRICA de las replicas:\n"
print Vemp
```

Matriz $\sigma^2 (X'X)^{-1}$ predicha por la lección 11:
Vteo (2 x 2)

```
7,0306e+06 -1,1050e+06
-1,1050e+06 1,7584e+05
```

Matriz de varianzas y covarianzas EMPIRICA de las replicas:
Vemp (2 x 2)

```
6,9249e+06 -1,0905e+06
-1,0905e+06 1,7386e+05
```

Compare las dos matrices elemento a elemento. El elemento (2, 2) es la varianza de $\hat{\beta}_2$ que ya habíamos comprobado en la Actividad 4; el (1, 1) es la de $\hat{\beta}_1$; y los de fuera de la diagonal son la covarianza entre ambos coeficientes —que, como señalaba la lección 11, es negativa: si la recta se inclina más, tiene que bajar el punto de corte para seguir pasando cerca de la nube de puntos.

Preguntas de interpretación para la clase

1. En la Actividad 2 obtuvimos una estimación distinta del valor verdadero. ¿Significa eso que MCO ha fallado en esa muestra? ¿Qué habría tenido que ocurrir para que coincidieran?
2. ¿Por qué mantenemos `rooms` fijo dentro del bucle, en lugar de generarlo de nuevo en cada repetición? ¿Con qué expresión de la lección 11 tiene eso que ver?
3. La media de las 2000 estimaciones no coincide con β_2 . ¿Contradice eso la insesgadez? ¿Qué esperaríamos que ocurriese con esa diferencia si hiciéramos 200 000 réplicas en lugar de 2000?
4. Suponga que repetimos todo el experimento con un σ diez veces mayor. ¿Qué cambiaría en la media de las estimaciones? ¿Y en su dispersión?
5. Un compañero sugiere que, como el histograma de la Actividad 5 parece una campana, ya podemos calcular la probabilidad de que $\hat{\beta}_2$ se aleje más de 500 de su valor verdadero. ¿Qué objeción le haría?

Para profundizar:

- Lección 11 ([lección 11](#)): insesgadez y varianza de $\hat{\beta}_2$.
- Wooldridge, J. M. (2020). *Introductory Econometrics*, cap. 2, Teoremas 2.1 y 2.2.
- Manual de Gretl, capítulo sobre *loop constructs* (en particular, los bucles `--progressive`).

Código completo de la práctica

```
# -----
# Copyright (C) 2026 Marcos Bujosa
# Licencia: GNU General Public License v3.0 o posterior
# Este código es software libre y puede ser redistribuido y/o modificado bajo los términos de la GPL.
# Ver el archivo LICENSE del repositorio para más detalles.
# -----

# Los dos primeros comandos son necesarios para que Gretl guarde los resultados de la práctica en el
↳ directorio de trabajo
# al ejecutar lo siguiente desde un terminal (use los nombres y ruta que correspondan)
#
#   DIRECTORIO="Nombre_Directorio_trabajo" gretlcli -b ruta/nombre_fichero_de_la_practica.inp
#
# Si esto no le funciona en su sistema, comente las siguientes dos líneas y sitúese en el directorio de
↳ trabajo de gretl
# que corresponda (configure dicho directorio de trabajo desde la ventana principal de Gretl).

string directory = getenv("DIRECTORIO")
set workdir "@directory"

set verbose off

open hprice2.gdt --quiet

ols price const rooms --quiet
scalar b1_true = $coeff(const)           # beta_1 "verdadero"
scalar b2_true = $coeff(rooms)           # beta_2 "verdadero"
scalar sigma = sqrt($ess/$df)           # sigma "verdadera" (raíz de la cuasivarianza)
scalar n = $nobs
scalar Sxx = sum((rooms - mean(rooms))^2)
scalar var_teo = sigma^2 / Sxx           # formula de la leccion 11

outfile --quiet mundoArtificial.txt
  printf "n          = %d\n", n
  printf "beta_1     = %.4f\n", b1_true
  printf "beta_2     = %.4f\n", b2_true
  printf "sigma      = %.4f\n", sigma
  printf "Sxx        = %.4f\n", Sxx
  printf "Var[b2 | X] = sigma^2/Sxx = %.4f\n", var_teo
  printf "desv. tipica = %.4f\n", sqrt(var_teo)
end outfile

outfile --quiet unaReplica.txt
  set seed 20261030                      # para que el resultado sea reproducible

  series u_sim = normal(0, sigma)         # perturbacion nueva
  series y_sim = b1_true + b2_true*rooms + u_sim # muestra simulada

  ols y_sim const rooms
  printf "\nbeta_2 verdadero = %.4f\n", b2_true
  printf "beta_2 estimado  = %.4f\n", $coeff(rooms)
  printf "diferencia       = %.4f\n", $coeff(rooms) - b2_true
end outfile

scalar R = 2000                          # numero de replicas

set seed 20261030
loop R --progressive --quiet
  series u = normal(0, sigma)
  series ysim = b1_true + b2_true*rooms + u
  ols ysim const rooms --quiet
  scalar b1hat = $coeff(const)
  scalar b2hat = $coeff(rooms)
  store "@workdir/replicas.gdt" b1hat b2hat
```

```

endloop

outfile --quiet bucle.txt
  printf "Hecho: %d replicas guardadas en replicas.gdt\n", R
end outfile

outfile --quiet analisisReplicas.txt
  printf "=== LO QUE PREDICE LA LECCION 11 ===\n"
  printf "beta_2 verdadero      = %.4f\n", b2_true
  printf "Var[b2 | X] = sigma^2/Sxx = %.4f\n", var_teo
  printf "desviacion tipica      = %.4f\n\n", sqrt(var_teo)
end outfile
open "@workdir/replicas.gdt" --quiet
outfile --quiet --append analisisReplicas.txt
  printf "=== LO QUE SALE DE %d REPLICAS SIMULADAS ===\n", $nobs
  printf "media de b2hat          = %.4f\n", mean(b2hat)
  printf "varianza de b2hat         = %.4f\n", sum((b2hat-mean(b2hat))^2)/$nobs
  printf "desviacion tipica        = %.4f\n", sd(b2hat)
end outfile

freq b2hat --normal --plot="@workdir/histograma_b2.png"

open hprice2.gdt --quiet                                # recuperamos los regresores
ols price const rooms --quiet
scalar sigma = sqrt($ess/$df)

outfile --quiet matrizVCV.txt
  matrix X      = {const, rooms}
  matrix invXTX = invpd(X'X)
  matrix Vteo   = sigma^2 * invXTX
  printf "Matriz sigma^2 (X'X)^{-1} predicha por la leccion 11:\n"
  print Vteo
end outfile
open "@workdir/replicas.gdt" --quiet
outfile --quiet --append matrizVCV.txt
  matrix B      = {b1hat}~{b2hat}
  matrix Vemp   = mcov(B)
  printf "\nMatriz de varianzas y covarianzas EMPIRICA de las replicas:\n"
  print Vemp
end outfile

```

Enlace al guión: [S16-Prct-A-montecarlo.inp](#)

Respuestas

1. *¿Ha fallado MCO?* No. MCO ha hecho exactamente lo que sabe hacer: encontrar la combinación lineal de los regresores más próxima al regresando *de esa muestra*. Lo que ocurre es que el regresando de esa muestra incluye una perturbación concreta que no conocemos y que MCO no puede separar de la parte sistemática. Según la identidad de la lección 11, $\hat{\beta}_2 = \beta_2 \mathbf{1} + \sum_i W_i U_i$: para que coincidieran tendría que anularse $\sum_i W_i U_i$, es decir, la perturbación sorteada tendría que resultar ortogonal al regresor centrado. Con una perturbación generada al azar eso no ocurre nunca (salvo casualidad numérica); en la práctica C lo forzaremos a propósito para ver qué pasa.
2. *¿Por qué el regresor se queda fijo?* Porque lo que la lección 11 calculó es la varianza *condicional* $\text{Var}[\hat{\beta}_2 \mid \mathbf{X}]$: la dispersión de las estimaciones *para un conjunto dado de valores del regresor*. Mantener **rooms** fijo en todas las réplicas es, literalmente, la versión experimental de condicionar en $\mathbf{X}$: fijamos la matriz de regresores y dejamos variar solo lo que el modelo considera aleatorio, la perturbación. La práctica B explora qué ocurre si dejamos de condicionar.
3. *La media no coincide.* No lo contradice. La insesgadez es una afirmación sobre la esperanza, es decir, sobre el promedio en el límite de infinitas muestras posibles. Con R réplicas obtenemos un promedio *muestral* de estimaciones, que es él mismo una variable aleatoria con varianza aproximadamente $\text{Var}[\hat{\beta}_2 \mid \mathbf{X}]/R$. Al multiplicar R por cien, esa varianza se divide por cien y la desviación típica por diez: cabe esperar que la diferencia con β_2 se reduzca aproximadamente a la décima parte.
4. *Un σ diez veces mayor.* La media de las estimaciones seguiría centrada en β_2 : la insesgadez no depende de σ en absoluto (la demostración de la lección 11 solo usa $E[U_i \mid \mathbf{X}] = 0$). La dispersión, en cambio, se multiplicaría por diez la desviación típica —y por cien la varianza—, porque σ^2 aparece como factor en el numerador de la fórmula. Dicho de otro modo: más ruido no descentra el estimador, solo lo vuelve menos preciso.
5. *La objeción al compañero.* Que la forma de campana del histograma no la hemos *descubierto*, sino *impuesto*: hemos generado las perturbaciones con `normal(0, sigma)` porque así lo decidimos al construir el mundo artificial. Nada de lo demostrado en la lección 11 garantiza esa forma; allí se demostraron la media y la varianza, y ambas se obtuvieron sin suponer nada sobre la distribución de U . Para calcular probabilidades hace falta un supuesto adicional sobre esa distribución, y ese supuesto —con sus consecuencias— es justamente el asunto de la lección 12. La práctica C de hoy muestra, además, que con perturbaciones muy poco normales la media y la varianza siguen cumpliéndose igual de bien.