Lección 6

Marcos Bujosa

5 de octubre de 2023

Índice

1. Varianza de los estimadores

Guión: EjPvivienda3.inp

Observe la matriz $\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$, del ejemplo del "precio de las viviendas".

$$\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1} = \begin{bmatrix} 9{,}1293e-01 & -4{,}4036e-04 \\ -4{,}4036e-04 & 2{,}3044e-07 \end{bmatrix};$$

¿Qué estimación cree que es más fiable, la de la pendiente o la de la constante?

- Calcule dicha matriz $\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$ con Gretl.

```
open data3-1
matrix X      = {const, sqft}
matrix invXTX = invpd(X'X)
print invXTX
```

- Realice la regresión de los precios sobre la superficie y añada el ajuste a la tabla de modelos (la opción `--vcv` muestra la matriz de varianzas y covarianzas de los estimadores ($\mathfrak{s}^2\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$) que Gretl estima con la muestra empleada en el ajuste `ols`). Compruebe que si multiplica la matriz `invXTX` por la cuasivarianza de los errores obtiene el mismo resultado.

```
ols price const sqft --vcv
modeltab add                              # incluimos el modelo a la tabla de modelos
matrix cv2invXTX = $ess/$df * invXTX
print cv2invXTX
```

- Repita el punto anterior pero con las siguientes modificaciones:

  1. con todos los datos excepto los de la última vivienda

     ```
     smpl 1 13                            # quitamos la primera observacion
     ols price const sqft --vcv
     modeltab add                         # incluimos el modelo a la tabla de modelos
     smpl full                            # recuperamos toda la muestra
     ```

  2. con todos los datos excepto los de las últimas dos viviendas

     ```
     smpl 2 14                            # quitamos la ultima observacion
     ols price const sqft --vcv
     modeltab add                         # incluimos el modelo a la tabla d
     smpl full                            # recuperamos toda la muestra
     ```

  3. con todos los datos excepto los de la primera y la última viviendas

     ```
     smpl 2 13                            # quitamos la primera y la ultima observaciones
     ols price const sqft --vcv
     modeltab add                         # incluimos el modelo a la tabla de modelos
     modeltab show                        # MOSTRAMOS TODOS LOS MODELOS a la vez
     ```

- ¿Confirman los resultados de estas regresiones su respuesta a la primera pregunta?

1.1. Código completo de la práctica `EjPvivienda3.inp`

```
open data3-1
matrix X      = {const, sqft}
matrix invXTX = invpd(X'X)
print invXTX

ols price const sqft --vcv
modeltab add                              # incluimos el modelo a la tabla de modelos
matrix cv2invXTX = $ess/$df * invXTX
print cv2invXTX

smpl 1 13                                 # quitamos la primera observacion
ols price const sqft --vcv
modeltab add                              # incluimos el modelo a la tabla de modelos
smpl full                                 # recuperamos toda la muestra

smpl 2 14                                 # quitamos la ultima observacion
ols price const sqft --vcv
modeltab add                              # incluimos el modelo a la tabla d
smpl full                                 # recuperamos toda la muestra

smpl 2 13                                 # quitamos la primera y la ultima observaciones
ols price const sqft --vcv
modeltab add                              # incluimos el modelo a la tabla de modelos
modeltab show                             # MOSTRAMOS TODOS LOS MODELOS a la vez
```

2. Un experimento de Montecarlo

Guión: samplinghouses0.inp

Cargue los datos del ejemplo de los precios de casas unifamiliares `data3-1.gdt`. Estime el modelo visto en clase. Guárdelo como icono. También debe guardar como icono el diagrama de dispersión entre `sqrt` y `price`.

```
open data3-1
Modelo0  <- ols price 0 sqft
Scatter0 <- gnuplot price sqft --output=display
```

Vamos a simular este modelo generando nuevos datos por Montecarlo.

- Genere una serie con la parte sistemática del modelo (empleando valores parecidos a los estimados):

  ```
  series x  = sqft
  series y1 = 52 + 0.14*x
  ```

- Genere una serie de perturbaciones con distribución normal, con esperanza nula y varianza 39 un valor parecido al obtenido con los datos originales

  ```
  series u1 = randgen(N, 0, 39)
  ```

- Genere una nueva serie de precios sumando a la parte sistemática las perturbaciones generadas en el paso anterior

  ```
  # Datos de precios simulados
  series y1 = ys + u1
  ```

- Con los nuevos datos de precios simulados ajuste el modelo de regresión de clase. ¿Se parecen los resultados? Grafique la nube de puntos `x,y` ¿Observa diferencias respecto al diagrama de dispersión original?

  ```
  Modelo1  <- ols y1 0 x
  Scatter1 <- gnuplot y1 x --output=display
  ```

- Repita los pasos pasos anteriores con nuevas simulaciones y observe los cambios.

  ```
  series u2 = randgen(N, 0, 39)
  series y2 = ys + u2
  Modelo2  <- ols y2 0 x
  Scatter2 <- gnuplot y2 x --output=display
  ```

2.1. Código completo de la práctica samplinghouses0.inp

```
open data3-1
Modelo0  <- ols price 0 sqft
Scatter0 <- gnuplot price sqft --output=display

series x  = sqft
series y1 = 52 + 0.14*x

series u1 = randgen(N, 0, 39)

# Datos de precios simulados
series y1 = ys + u1

Modelo1  <- ols y1 0 x
Scatter1 <- gnuplot y1 x --output=display

series u2 = randgen(N, 0, 39)
series y2 = ys + u2
Modelo2  <- ols y2 0 x
Scatter2 <- gnuplot y2 x --output=display
```

3. Repitiendo el experimento de Montecarlo muchas veces

Guión: samplinghouses1.inp

Vamos a simular el modelo de la práctica anterior 10000 veces para ver hasta qué punto estamos replicando los resultados originales.

Cargue los datos del ejemplo de los precios de casas unifamiliares `data3-1.gdt` y estime el modelo visto en clase. Guárdelo como icono para poder consultar los resultados más tarde.

```
open data3-1.gdt
Modelo <- ols price 0 sqft
```

- Como antes, genere una nueva serie con la parte sistemática del modelo empleando valores de los parámetros parecidos a los estimados.

  ```
  x = sqft
  series ys = 52 + 0.14*x
  ```

- Defina un escalar `s` con el valor aproximado de la desviación típica de los residuos del modelo original.

  ```
  scalar s  = 39
  ```

- Especifique un bucle para realizar 10000 iteraciones y que almacene los coeficientes estimados (`--progressive`) pero sin mostrar los resultados (`--quiet`) de cada iteración. *Lea antes la documentación sobre* `loops`.

  Dentro del bucle indicaremos una serie de operaciones y órdenes que se describen más abajo

  ```
  loop 10000 --progressive --quiet
     <<Simulamos el regresando y ajustamos por MCO>>
     <<Almacenamos los parámetros estimados>>
     <<Mostramos los resultados>>
     <<Guardamos los resultados en el disco>>
  endloop
  ```

1. ... dentro del bucle introduzca las instrucciones para simular en cada iteración un nuevo vector de precios (sumando unas perturbaciones con media cero y desviación típica `s`. Y realice la correspondiente regresión.

   ```
   y = ys + randgen(N, 0, s)
   ols y const x
   ```

2. Almacene los valores estimados de los betas correspondientes a la constante, la pendiente y la cuasivarianza varianza de los residuos

   ```
   scalar b1   = $coeff(const)
   scalar b2   = $coeff(x)
   scalar cs2  = $ess/$df                      # cuasivarianza de los errores
   ```

3. Indique que Gretl muestre el resumen estadístico de los parámetros estimados en las 10000 iteraciones.

   ```
   print b1 b2 cs2
   ```

4. Guarde los parámetros estimados en el disco

   ```
   store "@workdir\coef.gdt" b1 b2 cs2
   ```

(*Puede almacenar dentro del bucle otros estadísticos (varianza estimada, coeficiente de determinación, etc.) para observar el comportamiento de los valores obtenidos en este experimento de Montecarlo.*)

- Para analizar en detalle los valores obtenidos y almacenados en el fichero `coef.gdt`, Gretl debe abrir y leer dicho fichero. Lo podemos indicar en este mismo guión, pero Gretl abrirá una nueva sesión y perderemos lo calculado con el bucle y que no haya sido guardado en el fichero (no quiere hacerlo así, entonces tendrá que abrir una segunda sesión de Gretl y cargar a ahí los datos del fichero `coef.gdt`).

  ```
  open "@workdir\coef.gdt"
  ```

- Observe los valores máximos y mínimos estimados, y compárelos con los valores indicados en la simulación `b1=52`, `b2=0.14` y `s2=39^2=1521` (verá que en algunos casos lo estimado dista mucho del verdadero valor de los parámetros). Observe el histograma de los valores obtenidos para los parámetros

  ```
  summary   --simple
  freq b1   --normal --silent --plot="display"
  freq b2   --normal --silent --plot="display"
  freq cs2  --normal --silent --plot="display"
  ```

- Ejecute el guión y coteje los resultados de los estadísticos descriptivos de los betas estimados con los parámetros estimados en el modelo original (del de los datos originales visto en clase).

3.1. Código completo de la práctica `samplinghouses1.inp`

```
open data3-1.gdt
Modelo <- ols price 0 sqft

x = sqft
series ys = 52 + 0.14*x

scalar s  = 39

loop 10000 --progressive --quiet
   y = ys + randgen(N, 0, s)
   ols y const x
   scalar b1   = $coeff(const)
   scalar b2   = $coeff(x)
   scalar cs2  = $ess/$df                    # cuasivarianza de los errores
   print b1 b2 cs2
   store "@workdir\coef.gdt" b1 b2 cs2
endloop

open "@workdir\coef.gdt"

summary   --simple
freq b1   --normal --silent --plot="display"
freq b2   --normal --silent --plot="display"
freq cs2  --normal --silent --plot="display"
```

4. Repitiendo el experimento de Montecarlo muchas veces (Matriz de covarianzas)

Complete el experimento de Montecarlo de la Práctica 3 de más arriba con el siguiente añadido:

- Antes del bucle obtenga la matriz $\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$ y defina tres escalares `m11`, `m12` y `m22` correspondientes a los elementos (1,1), (1,2) y (2,2) de la matriz.

```
open data3-1.gdt
x = sqft
series ys = 52 + 0.14*x
scalar s  = 39

matrix X   = {const, sqft}
matrix invXTX = invpd(X'X) # inversa de X'X
scalar m11 = invXTX[1,1]
scalar m21 = invXTX[2,1]
scalar m22 = invXTX[2,2]
```

- Dentro del bucle incluya dichos escalares `m11`, `m12` y `m22` en la lista de parámetros a guardar

```
loop 10000 --progressive --quiet
   y = ys + randgen(N, 0, s)
   ols y const x
   scalar b1   = $coeff(const)
   scalar b2   = $coeff(x)
   scalar cs2  = $ess/$df                      # cuasivarianza de los errores
   print b1 b2 cs2
   store "@workdir\coef.gdt" b1 b2 cs2 m11 m21 m22
endloop
open "@workdir\coef.gdt"
summary   --simple  b1 b2 cs2
freq b1   --normal --silent --plot="display"
freq b2   --normal --silent --plot="display"
freq cs2  --normal --silent --plot="display"
```

- Finalmente, genere una matriz `S` de varianzas y covarianzas muestrales de las estimaciones de los betas obtenidos en las 10000 iteraciones. Compárela con el promedio las matrices $\mathfrak{s}^2\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$ cuyos elementos (1,1), (1,2) y (2,2) son los escalares `m11`, `m12` y `m22` (que guardamos en el punto anterior) multiplicados por la media de las cuasivarianzas estimadas `cs2`.

```
matrix s2invXTXhat = {var(b1), cov(b1,b2); cov(b2,b1), var(b2)}
matrix s2invXTX    = mean(cs2)*{m11[1],m21[1];m21[1],m22[1]}
print s2invXTX s2invXTXhat
```

- En promedio ¿es $\widehat{\mathfrak{s}^2}\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$ un buen estimador de las varianzas y covarianzas de los parámetros beta estimados en la simulación?

4.1. Código completo de la práctica samplinghouses2.inp

```
open data3-1.gdt
x = sqft
series ys = 52 + 0.14*x
scalar s  = 39

matrix X   = {const, sqft}
matrix invXTX = invpd(X'X) # inversa de X'X
scalar m11 = invXTX[1,1]
scalar m21 = invXTX[2,1]
scalar m22 = invXTX[2,2]

loop 10000 --progressive --quiet
   y = ys + randgen(N, 0, s)
   ols y const x
   scalar b1   = $coeff(const)
   scalar b2   = $coeff(x)
   scalar cs2  = $ess/$df                    # cuasivarianza de los errores
   print b1 b2 cs2
   store "@workdir\coef.gdt" b1 b2 cs2 m11 m21 m22
endloop
open "@workdir\coef.gdt"
summary   --simple  b1 b2 cs2
freq b1   --normal --silent --plot="display"
freq b2   --normal --silent --plot="display"
freq cs2  --normal --silent --plot="display"

matrix s2invXTXhat = {var(b1), cov(b1,b2); cov(b2,b1), var(b2)}
matrix s2invXTX    = mean(cs2)*{m11[1],m21[1];m21[1],m22[1]}
print s2invXTX s2invXTXhat
```

5. Repitiendo el experimento de Montecarlo muchas veces (perturbaciones con distribuciones no Gaussianas)

Complete el experimento de Montecarlo de la práctica anterior pero generando perturbaciones con distribución no normal (aunque con esperanza nula). Para ello

- Consulte la documentación sobre la función `randgen`.
- Repita los experimentos del ejercicio anterior pero generando perturbaciones con distribuciones distintas de la normal. Por ejemplo pruebe con
  - Distribución uniforme: `series U = randgen(u, -5, 5)`
  - Distribución beta (centrada): `series U = randgen(beta, 0.5, 0.5) - 0.5`
  - Distribución chi cuadrado (centrada): `series U = randgen(X, 3) - 3`
- Observe los histogramas y distribuciones de frecuencia así como los contrastes de normalidad. ¿Cambia mucho la distribución de los estimadores de los betas? ¿Y la del estimador de la varianza σ^2 de las perturbaciones?
- En promedio ¿es $\widehat{s^2}\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}$ un buen estimador de las varianzas y covarianzas de los parámetros beta estimados en la simulación incluso cuando las preturbaciones tienen distribución muy distinta a la gausiana?

5.1. Código completo de la práctica `samplinghouses3.inp`

```
open data3-1.gdt
x = sqft
series ys = 52 + 0.14*x
scalar s  = 39

matrix X   = {const, sqft}
matrix invXTX = invpd(X'X) # inversa de X'X
scalar m11 = invXTX[1,1]
scalar m21 = invXTX[2,1]
scalar m22 = invXTX[2,2]

loop 10000 --progressive --quiet
   series U = randgen(u, -5,  5)              # Descomente el que corresponda
   #series U = randgen(beta, 0.5, 0.5) - 0.5   # Descomente el que corresponda
   #series U = randgen(X, 3) - 3               # Descomente el que corresponda
   y = ys + U
   ols y const x
   scalar b1   = $coeff(const)
   scalar b2   = $coeff(x)
   scalar cs2  = $ess/$df                      # cuasivarianza de los errores
   print b1 b2 cs2
   store "@workdir\coef.gdt" b1 b2 cs2 m11 m21 m22
endloop
open "@workdir\coef.gdt"
summary   --simple  b1 b2 cs2
freq b1   --normal --silent --plot="display"
freq b2   --normal --silent --plot="display"
freq cs2 --normal --silent --plot="display"

matrix s2invXTXhat = {var(b1), cov(b1,b2); cov(b2,b1), var(b2)}
matrix s2invXTX    = mean(cs2)*{m11[1],m21[1];m21[1],m22[1]}
print s2invXTX s2invXTXhat
```