

Lección 13

Marcos Bujosa

4 de noviembre de 2023

Índice

1. Omisión de un regresor ortogonal al resto de regresores	2
2. Omisión de un regresor NO ortogonal al resto de regresores	3
3. Omisión de un regresor cuando hay multicolinealidad	4
4. Omisión de un regresor cuando hay multicolinealidad	5

1. Omisión de un regresor ortogonal al resto de regresores

Guión: `OmisionRegresorOrtogonal.inp`

Código completo de la práctica

```
nulldata 1000                                # tamaño de muestra en la simulación

loop 5000 --progressive --quiet
  series S = normal(0,10)                    # generamos datos S correspondientes a la superficie
  series D = normal(0,1)                    # generamos datos D correspondientes a la distancia
  series U = normal(0,25)                    # perturbaciones U: otros factores que afectan el precio

  /* variante para regresores verdaderamente ortogonales (media cero y covarianza cero) */
  /* series SS = normal(0,10)                # generamos datos S correspondientes a la superficie */
  /* series S = SS - mean(SS)                # con media cero */
  /* series DD = normal(0,1)                # generamos datos D correspondientes a la distancia */
  /* ols DD 0 S                             # regresion auxiliar */
  /* series D = $uhat                        # con media cero y perpendiculares a S */

  series P = 100 + 3*S - 130*D + U          # precios P como combinacion lineal de regresores + U

  ols P 0 S D                              # Regresion MCO Verdadero modelo: P = 100 + 3*S - 130*D + U
  scalar C1 = $coeff(const)
  scalar S1 = $coeff(S)
  scalar D1 = $coeff(D)
  scalar sigma1 = $sigma

  ols P 0 D                                # Regresion MCO omision S
  scalar C2 = $coeff(const)
  scalar D2 = $coeff(D)
  scalar sigma2 = $sigma

  ols P 0 S                                # Regresion MCO omision D
  scalar C3 = $coeff(const)
  scalar S3 = $coeff(S)
  scalar sigma3 = $sigma

  print C1 S1 D1 sigma1
  print C2 D2 sigma2
  print C3 S3 sigma3
endloop
```

2. Omisión de un regresor NO ortogonal al resto de regresores

Guión: [OmisionRegresorNoOrtogonal.inp](#)

Código completo de la práctica

```
nulldata 1000                                # tamaño de muestra en la simulación
loop 5000 --progressive --quiet

    series C = randgen(U, 0, 40)                # C: un factor común en S y D
    series S = randgen(U, 35, 100) + C          # datos S correspondientes a la superficie
    series D = randgen(X, 10) + C               # datos D correspondientes a la distancia
    series U = randgen(N, 0, 40)                # U: otros factores que afectan a P
    series P = 100 + 3*S - 130*D + U            # precios P como combinación lineal de regresores + U

    ols P 0 S D                                # Regresión MCO Verdadero modelo:  $P = 100 + 3*S - 130*D + U$ 
    scalar C1 = $coeff(const)
    scalar S1 = $coeff(S)
    scalar D1 = $coeff(D)
    scalar sigma1 = $sigma

    ols P 0 D                                  # Regresión MCO omisión S
    scalar C2 = $coeff(const)
    scalar D2 = $coeff(D)
    scalar sigma2 = $sigma

    ols P 0 S                                  # Regresión MCO omisión D
    scalar C3 = $coeff(const)
    scalar S3 = $coeff(S)
    scalar sigma3 = $sigma

    print C1 S1 D1 sigma1
    print C2 D2 sigma2
    print C3 S3 sigma3
endloop
```

3. Omisión de un regresor cuando hay multicolinealidad

Guión: [OmisionRegresorMulticol.inp](#)

Código completo de la práctica

```
nullldata 1000                                # tamaño de muestra en la simulacion

matrix C = {1 , 0; -10, 1}

loop 10000 --progressive --quiet
  series S = randgen(N, 10 , 5)                # primer regresor

  series X = randgen(N, -10 , 5)
  matrix Z = {S,X}
  Z *= C'                                       # note: use the transpose '
  series D = Z[,2]                             # segundo regresor

  series U = randgen(N, 0 , 40 )               # perturbaciones

  series Y = 1 + 10*S -1*D + U                 # regresando

  ols Y 0 S D                                  # Regresion MCO modelo completo
  scalar C1 = $coeff(const)
  scalar S1 = $coeff(S)
  scalar D1 = $coeff(D)
  scalar sigma1 = $sigma

  ols Y 0 D                                    # Regresion MCO omision S
  scalar C2 = $coeff(const)
  scalar D2 = $coeff(D)
  scalar sigma2 = $sigma

  ols Y 0 S                                    # Regresion MCO omision D
  scalar C3 = $coeff(const)
  scalar S3 = $coeff(S)
  scalar sigma3 = $sigma

  print C1 S1 D1 sigma1
  print C2 D2 sigma2
  print C3 S3 sigma3
endloop
```

4. Omisión de un regresor cuando hay multicolinealidad

Guión: [ErrorDeEspecificacionPorInclusion.inp](#)

Código completo de la práctica

```
nulldata 1000                                # tamaño de muestra en la simulacion

matrix C = { 1, 0, 0, 0; \
             100, 1, 0, 0; \
             -6, 0, 1, 0; \
             0, 1, 0, 1}

loop 10000 --progressive --quiet
  series S = randgen(N, 0, 1) # primer regresor

  series U = randgen(N, 0, 1) # perturbaciones

  series X = randgen(N, 0, 1)

  matrix Z = {S,X,U,U}
  Z *= C'                                     # note: use the transpose '

  series D = Z[,2]                           # segundo regresor fuertemente correlado solo con S (Variable Irrelevante)

  series E = Z[,4]                           # segundo regresor fuertemente correlado solo con U

  series F = Z[,3]/10                         # segundo regresor fuertemente correlado con S y U

  series Y = 1 - S + U                       # regresando beta1=1 y beta2=-1

  ols Y 0 S                                  # Regresion MCO modelo completo (estimadores insesgados)
  scalar C1 = $coeff(const)
  scalar S1 = $coeff(S)
  scalar sigma1 = $sigma

  ols Y 0 S D                                # Incusion regresor ortogonal a U (no a S) (Variable irrelevante)
  scalar C2 = $coeff(const)
  scalar S2 = $coeff(S)
  scalar D2 = $coeff(D)
  scalar sigma2 = $sigma

  ols Y 0 S E                                # Incusion regresor ortogonal a S (no a U)
  scalar C3 = $coeff(const)
  scalar S3 = $coeff(S)
  scalar E3 = $coeff(E)
  scalar sigma3 = $sigma

  ols Y 0 S F                                # Incusion regresor ni ortogonal a S ni a U
  scalar C4 = $coeff(const)
  scalar S4 = $coeff(S)
  scalar F4 = $coeff(F)
  scalar sigma4 = $sigma

  print C1 S1 sigma1
  print C2 S2 D2 sigma2
  print C3 S3 E3 sigma3
  print C4 S4 F4 sigma4
endloop
corr U S D E F
```