Lección 10

Marcos Bujosa

8 de octubre de 2023

Índice

1. Relaciones lineales en las variables

Guión: POE2-4.inp

Código completo de la práctica

```
open food.gdt

#Ajuste por Minimos Cuadrados
Modelo <- ols food_exp const income

#Resumen de estadisticos descriptivos
summary food_exp income

#Representacion grafica de los datos
NubePuntos <- gnuplot food_exp income --output="display"

#Listado de datos
print food_exp income --byobs
print food_exp income

#Calculo de la elasticidad
scalar elast = $coeff(income)*mean(income)/mean(food_exp)

#Prediccion del modelo
scalar yhat  = $coeff(const) + $coeff(income)*20

#Analisis de los residuos
series ehat = $uhat
normtest --jbera ehat
normtest --all   ehat
DibujoResiduos <- gnuplot ehat income --output="display"
```

2. Precio de casas unifamiliares

Guión: RamanathanEX6-1.inp

Código completo de la práctica

```
open data4-1

help logs                                          # mas informacion sobre el comando "logs"
logs sqft bedrms baths
Lin     <- ols price 0 sqft                        # "mejor"modelo lineal
modeltab add                                  # incluimos el modelo a la tabla de modelos

ols price 0 l_sqft l_bedrms l_baths # modelo lin-log
LinLog2  <- omit --auto=0.05
modeltab add                                  # incluimos el modelo a la tabla de modelos
modeltab show                                 # MOSTRAMOS TODOS LOS MODELOS a la vez

scalar yhat1500      = Lin.$coeff(const)+Lin.$coeff(sqft)*(1500)
scalar etaLin1500    = Lin.$coeff(sqft) * 1500/yhat1500
scalar yhat2000      = Lin.$coeff(const)+Lin.$coeff(sqft)*(2000)
scalar etaLin2000    = Lin.$coeff(sqft) * 2000/yhat2000
scalar yhat2500      = Lin.$coeff(const)+Lin.$coeff(sqft)*(2500)
scalar etaLin2500    = Lin.$coeff(sqft) * 2500/yhat2500
scalar yhat1500      = LinLog2.$coeff(const)+LinLog2.$coeff(l_sqft)*log(1500)
scalar etaLinLog1500 = LinLog2.$coeff(l_sqft) / yhat1500
scalar yhat2000      = LinLog2.$coeff(const)+LinLog2.$coeff(l_sqft)*log(2000)
scalar etaLinLog2000 = LinLog2.$coeff(l_sqft) / yhat2000
scalar yhat2500      = LinLog2.$coeff(const)+LinLog2.$coeff(l_sqft)*log(2500)
scalar etaLinLog2500 = LinLog2.$coeff(l_sqft) / yhat2500

print etaLin1500    etaLin2000    etaLin2500
print etaLinLog1500 etaLinLog2000 etaLinLog2500

printf "\n\n Un incremento de un 1\% en la superticie de una casa \n \
supone un aumento del precio de unos %.4f miles de dolares \n\n", LinLog2.$coeff(l_sqft)/100
```

3. Modelo para los salarios

Guión: RamanathanEX6-5.inp

Código completo de la práctica

```
open data6-4

logs WAGE
square EDUC EXPER AGE

Modelo1 <- ols l_WAGE const EDUC EXPER AGE sq_EDUC sq_EXPER sq_AGE
Modelo2 <- omit --auto

printf "\n Un año adicional de experiencia supone un incremento salarial \n\
esperado de approx. %2.2f por ciento\n\n", $coeff(EXPER)*100

printf "\n Con más precision: un año adicional de experiencia supone un \n\
incremento salarial esperado de %2.2f por ciento\n\n", (exp($coeff(EXPER))-1)*100

series lwhat   = $yhat
scalar sigmasq = $ess/$df
series what    = exp(lwhat+(sigmasq/2))
AjusteSalarios <- gnuplot what WAGE --suppress-fitted --output="display"

printf "\n Un año adicional de educación para alguien con formación de 1 año \n\
supone un incremento salarial esperado de approx. %2.2f por ciento\n\n",
2*$coeff(sq_EDUC)*1*100

printf "\n Un año adicional de educación para alguien con formación de 7 años \n\
supone un incremento salarial esperado de approx. %2.2f por ciento\n\n",
2*$coeff(sq_EDUC)*7*100

printf "\n Con más precision: un año adicional de educación para alguien con formación \n\
de 1 año supone un incremento salarial esperado de %2.2f por ciento\n\n",
(exp(2*$coeff(sq_EDUC)*1)-1)*100

printf "\n Con más precision: un año adicional de educación para alguien con formación \n\
de 7 años supone un incremento salarial esperado de %2.2f por ciento\n\n",
(exp(2*$coeff(sq_EDUC)*7)-1)*100
```

4. Otro modelo para los salarios

Guión: RamanathanEX6-6.inp

Código completo de la práctica

```
include criteria.gfn
open data6-4

logs WAGE
square EDUC

LinLin        <- ols WAGE 0 sq_EDUC EXPER
LogLin        <- ols l_WAGE 0 sq_EDUC EXPER

series lwhat  = $yhat
scalar sgmasq = $ess/$df
series what   = exp(lwhat+(sgmasq/2))
series error  = WAGE -what
scalar ess    = sum(error*error)

# info sobre la función çriteria"
# help criteria
# calculando criterios de seleccion
criteria(LinLin.$ess, $nobs, LinLin.$ncoeff)
criteria(        ess, $nobs,        $ncoeff)

# correlacion entre salarios observados y ajustados
corr WAGE what
scalar c1          = corr(WAGE,what)
scalar LinLog_rsq  = c1*c1
scalar LinLin_rsq  = LinLin.$rsq
print  LinLin_rsq LinLog_rsq
```

5. Elasticidades en la demanda del transporte en autobús

Guión: RamanathanAp6-11.inp

Código completo de la práctica

```
open data4-4
logs BUSTRAVL FARE GASPRICE INCOME POP DENSITY LANDAREA
ols l_BUSTRAVL const l_FARE l_GASPRICE l_INCOME l_POP l_DENSITY l_LANDAREA
LogLog <- omit --auto

scalar tInc  = (abs($coeff(l_INCOME))  -1) / $stderr(l_INCOME)
scalar pInc  = 2*pvalue(t,$df,abs(tInc))

scalar tPop  = (abs($coeff(l_POP))     -1) / $stderr(l_POP)
scalar pPop  = 2*pvalue(t,$df,abs(tPop))

scalar tLand = (abs($coeff(l_LANDAREA))-1) / $stderr(l_LANDAREA)
scalar pLand = 2*pvalue(t,$df,abs(tLand))
```

6. Precio de casas unifamiliares (Modelo Lin-log)

Guión: Houses2.inp

Objetivo
- Ajustar un modelo que incorpore posibles no linealidades en el comportamiento de los precios.
- Eliminar variables no significativas de un modelo.
- Comparar el ajuste de dos modelos.
- Interpretar los coeficientes de un modelo lineal-logarítmico.

Los datos Son los datos del ejemplo de clase, junto con dos variables adicionales: número de dormitorios (`bedrms`) y cuartos de baño (`baths`).

Motivación: Probablemente la valoración en el mercado de un metro cuadrado adicional de casa no es la misma para casas grandes (pues un metro más añade poca utilidad) que en casas pequeñas.

Por ello, una forma funcional que recoja esta no linealidad en el comportamiento de los precios parece sensata.

Así pues, vamos a incorporar los regresores en logaritmos.

Para empezar primero cargue los datos de la base de datos de Gretl

```
open data4-1
```

6.1. Tareas

Actividad 1 Transforme las variables `sqft`, `bedrms` y `baths`} en logaritmos y ajuste el siguiente modelo de regresión lin-log

$$PRICE = \beta_1 + \beta_2 L\_SQFT + \beta_3 L\_BEDRMS + \beta_4 L\_BATHS + U.$$

Marque con el ratón las variables `sqft`, `bedrms` y `baths` y "pinche" en ***Añadir ->Logaritmos de las variables seleccionadas***.

Ajuste por MCO el modelo indicado.

O bien ejecute el código

```
logs sqft bedrms baths
Modelo1 <- ols price 0 l_sqft l_bedrms l_baths
```

Actividad 2 Observe que el parámetro asociado a `l_baths`} no es estadísticamente significativo.

Pruebe a omitir dicha variable para ver si el modelo mejora.

```
omit l_baths
```

Actividad 3 Observe que el parámetro asociado a `l_bedrms`, tampoco es estadísticamente significativo.

Repita con esta variable los pasos de la actividad anterior.

```
omit l_bedrms
```

Actividad 4 Habrá podido observar que aunque el parámetro asociado a `l_bedrms` no era estadísticamente significativo al 10 %, si lo era al 12 %.

Como además ninguno de los estadísticos de selección de modelos ha mejorado al omitir esta variable vamos a optar por incluirla de nuevo.

- En la ventana del último modelo estimado, "pinche" en ***Contrastes ->Añadir variables*** y seleccione `l_bedrms`.

- Compruebe que mejoran los 3 estadísticos de selección de modelos.

O bien ejecute el código

```
Modelo2 <- add l_bedrms
```

Actividad 5 En un modelo con variable dependiente en niveles, y regresor en logaritmos, la interpretación del parámetro β asociado es la siguiente:

Un incremento de un uno por ciento $(0{,}01)$ *del regresor supone un incremento del nivel del regresando de* $\beta * 0{,}01$.

Empleando el último modelo, compruebe que 10,65 pies cuadrados adicionales en la casa más pequeña tienen el mismo precio esperado que 30 pies cuadrados en la casa más grande de la muestra.

- Amplíe la muestra en dos observaciones (*Datos ->Añadir observaciones* y añada 2).
  - Para la casa número 15, replique los datos de la 14, pero con 1 % más de superficie.
  - Para la casa número 16, replique los datos de la 1, pero con 1 % más de superficie.

  Marque `l_sqft`, `l_bedrms` y `l_baths` y en *Datos ->Editar los valores* añada los siguientes valores

Cuadro 1: Logaritmo de los datos para las casas 15 y 16 (réplica de las casa 14 y 1 salvo por el incremento en un 1 % de la superficie)

obs.	l_sqft	l_bedrms	l_baths
15	8,0163179=log(3000*1.01)	8,0163179=log(4)	1,0986123=log(3)
16	6,9806804=log(1065*1.01)	1,0986123=log(3)	0,55961579=log(1.75)

O bien ejecute el código

```
dataset addobs 2

genr l_sqft[15]  =log(3000*1.01)
genr l_bedrms[15]=log(4)
genr l_baths[15] =log(3)

genr l_sqft[16]  =log(1065*1.01)
genr l_bedrms[16]=log(3)
genr l_baths[16] =log(1.75)
```

- Ajuste la muestra a las observaciones 1 a 16 (*Muestra ->Establecer rango* y establezca el rango de 1 a 16)
- Re-estime el modelo finalmente elegido
- En la ventana del modelo re-estimado observe la predicción de los precios (*Análisis ->Predicciones* y ajuste el *Dominio de predicción* para que incluya toda la muestra (observaciones 1 a 15).
- Verifique que el aumento de precio $(2984, 8$ dólares) entre la casa 1 y la 16 es el mismo que entre la casa 14 y la 15, pese a que a la casa 15 se le ha agregado el triple de superficie que a la 16.

  O bien ejecute el código

```
smpl 1 16
fcast --static predP
print predP -o

matrix P = {predP}
d1 = P[16,1]-P[1,1]
d2 = P[15,1]-P[14,1]
```

6.2. Código completo de la práctica

```
open data4-1

logs sqft bedrms baths
Modelo1 <- ols price 0 l_sqft l_bedrms l_baths

omit l_baths

omit l_bedrms

Modelo2 <- add l_bedrms

dataset addobs 2

genr l_sqft[15]  =log(3000*1.01)
genr l_bedrms[15]=log(4)
genr l_baths[15] =log(3)

genr l_sqft[16]  =log(1065*1.01)
genr l_bedrms[16]=log(3)
genr l_baths[16] =log(1.75)

smpl 1 16
fcast --static predP
print predP -o

matrix P = {predP}
d1 = P[16,1]-P[1,1]
d2 = P[15,1]-P[14,1]
```