

Guía de Instrucciones



Gnu Regression, Econometrics and Time-series

Allin Cottrell
Department of Economics
Wake Forest university

Riccardo "Jack" Lucchetti
Dipartimento di Economia
Università Politecnica delle Marche

Copyright © 2001–2005 Allin Cottrell

Copyright de la traducción española: © 2006 Ignacio Díaz-Emparanza (Basado en la traducción del manual de 2003 de Parmeeta Bhogal e Ignacio Díaz-Emparanza)

Este documento puede ser copiado, distribuido y/o modificado bajo los términos de la *GNU Free Documentation License*, Versión 1.1 o cualquiera de las versiones posteriores publicadas por la Free Software Foundation (véase <http://www.gnu.org/licenses/fdl.html>).

Se agradece la financiación de la Universidad del País Vasco, a través del grupo de investigación 9/UPV-00038.321-13503/2001, para la traducción de este manual.

Índice general

1. Instrucciones de gretl	1
1.1. Notación	1
1.2. Instrucciones	1
add	1
addto	2
adf	2
ar	2
arch	2
arima	3
boxplot	3
chow	4
coeffsum	4
coint	4
coint2	5
corc	5
corr	5
corrgm	5
criteria	6
cusum	6
data	6
delete	7
diff	7
else	7
end	7
endif	7
endloop	7
eqnprint	7
equation	8
fcast	8
fcasterr	8
freq	8
function	8
garch	9
genr	9
gnuplot	11
graph	13

hausman	13
hccm	13
help	13
hilu	13
hsk	14
if	14
info	14
setinfo	14
labels	14
lad	15
lags	15
ldiff	15
leverage	15
lmtest	16
logistic	16
logit	16
logs	17
loop	17
meantest	17
modeltab	18
mpols	18
nls	18
nulldata	19
ols	19
omit	20
omitfrom	20
open	20
outfile	21
panel	21
pca	21
pergm	21
plot	22
print	22
printf	22
probit	23
pvalue	23
pwe	23
quit	23
rename	24
reset	24
restrict	24

rhodiff	24
rmplot	25
run	25
runs	25
scatters	25
set	26
setobs	26
setmiss	27
shell	27
simpl	27
spearman	28
square	28
store	28
summary	29
system	29
tabprint	29
testuhat	30
tobit	30
tsls	30
var	30
varlist	31
vartest	31
vif	31
wls	31
1.3. Instrucciones arreglaron por tema	31
Estimación	31
Contrastes	32
Transformaciones	32
Estadísticos	32
Conjunto de datos	32
Gráficos	32
Impresión	33
Predicción	33
Programación	33
Utilidades	33
2. Opciones, argumentos y directorios	34
2.1. gretl	34
2.2. gretlcli	35
2.3. Directorios	36
MS Windows	36

Índice general	IV
3. Reserved words	37

Capítulo 1

Instrucciones de gretl

1.1. Notación

Las instrucciones definidas en esta sección pueden ejecutarse en el programa cliente de línea de instrucciones. También pueden incluirse en un archivo o “lote de instrucciones” (script) y así ejecutarse en el GUI, o teclearse mediante el modo consola de este último. En la mayoría de los casos la sintaxis que se menciona es también aplicable para rellenar una línea en el correspondiente cuadro de diálogo del GUI (véase también la ayuda *en línea* de gretl), excepto que *no* es preciso teclear la palabra inicial de la instrucción — está implícita por el contexto. Una diferencia adicional es que no se puede insertar la marca *-o* para las instrucciones de regresión en los cuadros de diálogo del GUI: hay una opción de menú para mostrar la matriz formada por las varianzas y covarianzas de los coeficientes (que es el efecto de *-o* en las instrucciones de las regresiones).

A lo largo de este capítulo se utilizan las siguientes convenciones:

- La fuente *typewriter* se utiliza en lo que tecleamos directamente, y también para los nombres internos de las variables.
- Los términos en *cursiva* son marcadores de ubicación: es posible sustituirlos por algo más específico, por ejemplo, se puede escribir *renta* en lugar del genérico *varx*.
- [*-o*] significa que la marca *-o* es opcional: puede ser añadida o no (pero en todo caso sin los paréntesis).
- La frase “instrucción de estimación” puede significar cualquiera de las siguientes *ols*, *h1lu*, *corc*, *ar*, *arch*, *hsk*, *ts1s*, *wls*, *hccm*, *add* y *omit*.

1.2. Instrucciones

add

Argumento: *listavar*

Opciones: *--vcv* (mostrar matriz de covarianzas)

--quiet (no muestra las estimaciones del modelo aumentado)

Ejemplos: *add 5 7 9*

add xx yy zz

Debe utilizarse después de una instrucción de estimación. Las variables de *listavar* se añaden al modelo anterior y se estima el nuevo modelo. Si se añade más de una variable, se presenta el estadístico *F* para las variables añadidas (sólo para el método MCO) junto con su valor-*p*. Un valor-*p* inferior a 0.05 indica que los coeficientes son conjuntamente significativos al nivel del 5 por ciento.

Si se usa la opción *--quiet* el resultado que se muestra es sólo el estadístico para el contraste de significación conjunta de las variables añadidas, en los demás casos se presentan también las estimaciones del modelo aumentado. En este último caso, la opción *--vcv* hace que también se presente la matriz de covarianzas de los coeficientes.

Menú gráfico: Ventana de Modelo, /Contrastes/Añadir variables

addtoArgumentos: *ID-modelo listavar*Opción: *--quiet* (no mostrar las estimaciones del modelo aumentado)Ejemplo: *addto 2 5 7 9*

Funciona como la instrucción *add*, salvo que Vd. especifica un modelo previo (usando su identificador *-ID-* de modelo, que se presenta al principio de los resultados del modelo) que se toma como base para añadir las variables. En el ejemplo de arriba se añaden las variables números 5, 7 y 9 al Modelo 2.

Menú gráfico: Ventana de Modelo, /Contrastes/añadir variables

adfArgumentos: *orden nombre-var*Ejemplo: *adf 2 x1*

Calcula los estadísticos para dos contrastes de Dickey-Fuller. En cada caso la hipótesis nula es que la variable seleccionada presenta una raíz unitaria. El primero es un estadístico *t* basado en el modelo

$$(1 - L)x_t = m + gx_{t-1} + \epsilon_t$$

La hipótesis nula es que $g = 0$. En el segundo contraste (aumentado) se realiza la estimación de una regresión no retringida (cuyos regresores son una constante, una tendencia temporal, el primer retardo de la variable y *orden* retardos de la primera diferencia) y una versión restringida (quitando la tendencia temporal y el primer retardo). El estadístico de contraste es

$$F_{2,T-k} = \frac{(ESS_r - ESS_u)/2}{ESS_u/(T - k)}$$

donde T es el tamaño de la muestra, k el número de parámetros del modelo no restringido, y los subíndices u y r denotan el modelo no restringido y el restringido respectivamente. Hay que señalar que los valores críticos para estos estadísticos no son los habituales; se muestra el valor *p* cuando es posible determinarlo.

Menú gráfico: /Variable/Contraste aumentado de Dickey-Fuller

arArgumentos: *retardos ; vardep varindeps*Opción: *--vcv* (mostrar la matriz de covarianzas)Ejemplo: *ar 1 3 4 ; y 0 x1 x2 x3*

Computa las estimaciones de los parámetros usando el procedimiento iterativo generalizado de Cochrane-Orcutt (ver Sección 9.5 del libro de Ramanathan). Las iteraciones terminan cuando dos sumas de cuadrados residuales sucesivas no difieren en más del 0.005 por ciento o después de 20 iteraciones.

retardos es una lista de retardos de los residuos, que termina con un punto y coma. En el ejemplo anterior el término de error se especifica como

$$u_t = \rho_1 u_{t-1} + \rho_3 u_{t-3} + \rho_4 u_{t-4} + e_t$$

Menú gráfico: /Modelo/Serie temporal/Estimación autorregresiva

archArgumentos: *orden vardep varindeps*Ejemplo: *arch 4 y 0 x1 x2 x3*

Contrasta la existencia de ARCH (Autoregressive Conditional Heteroskedasticity - Heterocedasticidad condicional autorregresiva) del orden especificado, en el modelo. Si el estadístico de

contraste LM tiene un valor p inferior a 0.10, entonces también se realiza la estimación del modelo ARCH. Si la predicción de la varianza de cualquier observación, en la regresión auxiliar, no es positiva entonces se usa en su lugar el correspondiente residuo al cuadrado. Después se realiza una estimación por mínimos cuadrados ponderados sobre el modelo original.

Ver también [garch](#).

Menú gráfico: Ventana de Modelo, /Contrastes/ARCH

arima

Argumentos: $p\ q ; vardep\ [varindeps]$

Opciones: `--native` (Usar rutina nativa plugin (por defecto))
`--x-12-arima` (usar el programa X-12-ARIMA para realizar la estimación)
`--verbose` (mostrar detalles de las iteraciones)
`--vcv` (mostrar la matriz de covarianzas)

Ejemplos: `arima 1 2 ; y`
`arima 2 2 ; y 0 x1 x2 --verbose`

Si no se da una lista *varindeps*, estima un modelo ARMA (Autoregressive, Moving Average) univariante. Los valores enteros p y q representan los órdenes AR y MA respectivamente. Si se añade una lista *varindeps*, el modelo pasa a ser un ARMAX.

Por defecto se usa la función “nativa” ARMA de gretl; en el caso de un modelo ARMA univariante se puede usar X-12-ARIMA en su lugar (si está instalado el paquete X-12-ARIMA para gretl).

Las opciones dadas arriba pueden combinarse, pero la matriz de covarianzas no está disponible cuando se realiza la estimación mediante X-12-ARIMA.

El algoritmo ARMA nativo de gretl ha sido, en su mayor parte, creado por Riccardo “Jack” Luchetti. Utiliza un procedimiento de máxima verosimilitud condicional, implementado por medio de la estimación por mínimos cuadrados iterativos de la regresión del producto externo del gradiente (OPG, outer product of the gradient regression). Ver the *Gretl User's Guide* para entender la lógica de este procedimiento. Los coeficientes AR (y los de cualquier regresor adicional) se inicializan por medio de la estimación por MCO de un AR, y los coeficientes MA se inicializan asignándoles valor cero.

El valor del coeficiente AIC para los modelos ARMA se calcula de acuerdo con la definición usada en el programa X-12-ARIMA, es decir

$$AIC = -2L + 2k$$

donde L es la log-verosimilitud y k es el número total de parámetros estimados. La “frecuencia” que se presenta asociada a las raíces AR y MA es el valor λ que soluciona

$$z = re^{i2\pi\lambda}$$

donde z es la raíz en cuestión y r es su módulo.

Menú gráfico: /Variable/Modelo ARMA, /Modelo/Serie temporal/ARMAX

Otro acceso: Menú emergente de la ventana principal (selección simple)

boxplot

Argumento: *listavar*

Opción: `--notches` (presenta el intervalo de confianza 90 por ciento para la mediana)

Estos gráficos (debidos a Tukey y Chambers) muestran la distribución de una variable. La caja central recoge el 50 por ciento intermedio de los datos, es decir, está acotada por el primero y tercer cuartiles. Las “patillas” se extienden hasta los valores mínimo y máximo. Se dibuja una línea a lo largo de la caja en el lugar de la mediana.

En el caso de gráficos de caja recortados, el recorte muestra los límites de un intervalo de aproximadamente el 90 por ciento de confianza para la mediana. Este intervalo se obtienen por el método bootstrap.

Después de cada variable especificada en la instrucción `boxplot`, se puede añadir una expresión booleana entre paréntesis para limitar la muestra para la variable en cuestión. Debe insertarse un espacio entre el nombre o número de la variable y la expresión. Supongamos que usted tiene valores de los salarios de hombres y mujeres, y tiene una variable ficticia `SEXO` con valor 1 para los hombres y 0 para las mujeres. En este caso usted puede dibujar gráficos de caja comparativos usando la siguiente *listavar*:

```
salary (GENDER=1) salary (GENDER=0)
```

Algunos detalles de los gráficos de caja de gretl pueden controlarse por medio de un fichero (de texto plano) llamado `.boxplotrc`. Para más detalles sobre esto ver the *Gretl User's Guide*.

Menú gráfico: /Datos/Gráficos/Gráficos de caja

chow

Argumento: *obs*

Ejemplos: `chow 25`

`chow 1988:1`

Primero debe ejecutarse una regresión por MCO. Crea una variable ficticia que es igual a 1 desde el punto de corte especificado en *obs* hasta el final de la muestra y 0 en el resto. También crea los términos de interacción entre esta variable ficticia y las variables independientes originales. Se ejecuta una regresión aumentada incluyendo estos términos y se calcula un estadístico *F*, tomando la regresión aumentada como 'no restringida' y la original como 'restringida'. Este estadístico es adecuado para contrastar la hipótesis nula de que no hay cambio estructural en el punto de ruptura indicado.

Menú gráfico: Ventana de Modelo, /Contrastes/Contraste de Chow

coeffsum

Argumento: *listavar*

Ejemplo: `coeffsum xt xt_1 xr_2`

Debe ejecutarse después de una regresión. Calcula la suma de los coeficientes de las variables de la lista *listavar*. Presenta la suma junto con su desviación típica y el valor *p* para la hipótesis nula de que la suma es cero.

Hay que señalar la diferencia entre esta instrucción y `omit`, que contrasta la hipótesis nula de que los coeficientes de un subconjunto de variables independientes son *todos* iguales a cero.

Menú gráfico: Ventana de Modelo, /Contrastes/Suma de coeficientes

coint

Argumentos: *orden vardep varindeps*

Ejemplo: `coint 4 y x1 x2`

Contraste de cointegración de Engle-Granger. Esta instrucción realiza los contrastes de Dickey-Fuller aumentados de la hipótesis nula de que cada una de las variables de la lista tiene una raíz unitaria, usando el orden de retardos dado. Se estima la ecuación cointegrante y se realiza un contraste ADF sobre los residuos de esta regresión. También se presenta el estadístico de Durbin-Watson para la regresión cointegrante. Hay que señalar que ninguno de estos estadísticos de contraste puede compararse con las tablas estadísticas usuales.

Menú gráfico: /Modelo/Serie temporal/Contrates de cointegración/Engle-Granger

coint2

Argumentos: *orden vardep varindeps*

Opción: `--verbose` (mostrar los detalles de las regresiones auxiliares)

Ejemplos: `coint2 2 y x`
`coint2 4 y x1 x2 --verbose`

Realiza el contraste de la traza de Johansen para contrastar cointegración entre las variables de la lista para el orden dado. Los valores críticos se computan por medio de la aproximación gamma de J. Doornik (Doornik, 1998). Para detalles sobre este contraste ver el libro de Hamilton, *Time Series Analysis* (1994), Capítulo 20.

Se presenta aquí la siguiente tabla como guía para la interpretación de los resultados ofrecidos por el contraste, para el caso de 3 variables. H_0 denota la hipótesis nula, H_1 la hipótesis alternativa y c el número de relaciones cointegrantes.

Rango	Contraste de la traza		Contraste Lmax	
	H_0	H_1	H_0	H_1
0	$c = 0$	$c = 3$	$c = 0$	$c = 1$
1	$c = 1$	$c = 3$	$c = 1$	$c = 2$
2	$c = 2$	$c = 3$	$c = 2$	$c = 3$

Menú gráfico: /Modelo/Serie temporal/Contraste de cointegración/Johansen

corc

Argumentos: *vardep varindeps*

Opción: `--vcv` (mostrar la matriz de covarianzas)

Ejemplo: `corc 1 0 2 4 6 7`

Computa estimaciones de los parámetros usando el procedimiento iterativo de Cochrane-Orcutt (ver Sección 9.4 del libro de Ramanathan). Las iteraciones finalizan cuando dos estimaciones sucesivas del coeficiente de autocorrelación no difieren en más de 0,001 o después de 20 iteraciones.

Menú gráfico: /Modelo/Serie temporal/Cochrane-Orcutt

corr

Argumento: *[listavar]*

Ejemplo: `corr y x1 x2 x3`

Presenta los coeficientes de correlación por pares para las variables de *listavar*, o para todas las variables del conjunto de datos, si no se proporciona *listavar*.

Menú gráfico: /Datos/Matriz de correlación

Otro acceso: Menú emergente de la ventana principal (cuando se hace selección múltiple)

corrgm

Argumentos: *variable [retardo-max]*

Ejemplo: `corrgm x 12`

Presenta los valores de la función de autocorrelación para la *variable* especificada (por nombre o por número). Ver Ramanathan, Sección 11.7. Es por tanto $\rho(u_t, u_{t-s})$ donde u_t es la t -ésima observación de la variable u y s es el número de retardos.

También se presentan las autocorrelaciones parciales, siendo éstas los coeficientes una vez descontado el efecto de los demás retardos intervinientes. Esta instrucción también representa el correlograma y muestra el estadístico Q de Box-Pierce para el contraste de la hipótesis nula

de que la serie es “ruido blanco”. Este se distribuye asintóticamente como una chi-cuadrado con grados de libertad igual al número de retardos utilizados.

Si se especifica un valor para *retardo-max* entonces el tamaño del correlograma se limita como máximo a ese número de retardos, en los demás casos se determina automáticamente.

Menú gráfico: /Variable/Correlograma

Otro acceso: Menú emergente de la ventana principal (cuando se hace una selección simple)

criteria

Argumentos: *scr T k*

Ejemplo: *criteria 23.45 45 8*

Calcula los estadísticos de selección de modelos (ver Ramanathan, Sección 4.3), dados *scr* (suma de cuadrados de los residuos), el número de observaciones (*T*), y el número de coeficientes (*k*). *T*, *k*, y *scr* pueden ser valores numéricos o nombres de variables definidas previamente.

cusum

Debe realizarse después de la estimación de un modelo por MCO. Realiza el contraste CUSUM de estabilidad de los parámetros. Se obtiene una serie de errores de predicción (escalados) un período hacia adelante al ejecutar una serie de regresiones: la primera regresión utiliza las primeras *k* observaciones y se usa para generar una predicción de la variable dependiente en la observación *k + 1*; la segunda utiliza las primeras *k + 1* observaciones y genera una predicción para la observación *k + 2*, y así sucesivamente (donde *k* es el número de parámetros en el modelo original). Se muestra y se representa gráficamente la suma acumulada de los errores de predicción escalados. La hipótesis nula de estabilidad en los parámetros se rechaza al nivel de significación del 5 por ciento si la suma acumulada se sale de la banda de 95 por ciento de confianza.

También se presenta el estadístico *t* de Harvey-Collier para contrastar la hipótesis nula de estabilidad de los parámetros. Para más detalles ver Capítulo 7 del libro de Greene *Econometric Analysis*.

Menú gráfico: Ventana de Modelo, /Contrastes/CUSUM

data

Argumento: *listavar*

Lee las variables de *listavar* desde una base de datos (de gretl o RATS 4.0), que debe de haber sido abierta previamente usando la instrucción [open](#). Además, se debe de establecer una frecuencia y un rango muestral para los datos usando las instrucciones [setobs](#) y [smp1](#) antes de usar esta orden. He aquí un ejemplo completo:

```
open macrodat.rat
setobs 4 1959:1
smp1 ; 1999:4
data GDP_JP GDP_UK
```

Estas instrucciones abren una base de datos denominada *macrodat.rat*, establecen un conjunto de datos trimestral que empieza en el primer trimestre de 1959 y acaba en el cuarto trimestre de 1999, y después importan las series denominadas *GDP_JP* y *GDP_UK*.

Si las series que van a ser leídas son de una frecuencia mayor que el conjunto de datos actual, se debe especificar un método de compactado como en las siguientes líneas:

```
data (compact=average)
LHUR PUNEW
```

Los cuatro métodos de compactado disponibles son “average” (promediar: toma la media de las observaciones de frecuencia mayor), “last” (última: usa la última observación), “first” y “sum”.

Menú gráfico: /Archivo/Revisar bases de datos

delete

Argumento: [*listavar*]

Borra las variables de la lista (dadas mediante nombre o número) del conjunto de datos. *Debe usarse con precaución*: no se pide confirmación, y cualquier variable con número de ID mayor será renumerada.

Si con esta instrucción no se proporciona una lista *listavar*, se borra la última variable (la de número ID mayor) del conjunto de datos.

Menú gráfico: Menú emergente de la ventana principal (selección simple)

diff

Argumento: *listavar*

Se obtiene la primera diferencia de cada variable de la lista *listavar* y el resultado se guarda en una nueva variable cuyo nombre tiene el prefijo *d_*. Así *diff x y* crea las nuevas variables $d_x = x(t) - x(t-1)$ y $d_y = y(t) - y(t-1)$.

Menú gráfico: /Datos/Añadir variables/Primeras diferencias

else

Ver [if](#).

end

Cierra un bloque de instrucciones de algunos tipos. Por ejemplo, *end system* termina un sistema de ecuaciones (Ver [system](#)).

endif

Ver [if](#).

endloop

Marca el final de un bucle de instrucciones. Ver [loop](#).

eqnprint

Argumento: [*-f nombre-de-fichero*]

Opción: *--complete* (Crea un documento completo)

Debe ejecutarse después de la estimación de un modelo. Presenta el modelo estimado en formato de ecuación $\LaTeX$. Si se especifica un nombre de fichero usando la opción *-f* la salida va a ese fichero, en caso contrario va a una fichero cuyo nombre tiene la forma *equation_N.tex*, donde N es el número de modelos estimados hasta el momento en la sesión actual. Ver también [tabprint](#).

Si se aplica la opción *--complete*, el fichero $\LaTeX$ es un documento completo, listo para ser procesado; en caso contrario deberá incluirse dentro de un documento previamente preparado.

Menú gráfico: Ventana de Modelo, / $\LaTeX$

equationArgumentos: *vardep varindeps*Ejemplo: `equation y x1 x2 x3 const`

Sirve para especificar una ecuación dentro de un sistema de ecuaciones (ver [system](#)). La sintaxis para especificar una ecuación dentro de un sistema SUR (Ecuaciones de regresión aparentemente no relacionadas) es la misma que, por ejemplo, la de MCO (ver [ols](#)). Para una ecuación dentro de un sistema de Mínimos Cuadrados en Tres Etapas se puede (a) dar la especificación de una ecuación de tipo MCO y proporcionar una lista común de instrumentos usando la opción *instr* (ver de nuevo [system](#)), o (b) usar la misma sintaxis de definición de ecuaciones que para [tsls](#).

fcastArgumentos: `[ obs-inic obs-fin ] var-ajustada`Ejemplos: `fcast 1997:1 2001:4 f1``fcast fit2`

Debe ejecutarse después de una instrucción de estimación. Se generan predicciones para el dominio muestral especificado (o el dominio muestral mayor posible si no se han proporcionado *obs-inic* y *obs-fin*) y los valores se guardan como *var-ajustada*, que luego puede mostrarse o representarse gráficamente. Las variables del lado derecho son las del modelo original. No se pueden sustituir por otras variables. Si se especifica un proceso de error autorregresivo (para *hilo*, *corc*, y *ar*) la predicción se hace condicionada un paso adelante y tiene en cuenta el proceso del error.

Menú gráfico: Ventana de Modelo, /Datos del Modelo/Predicciones con desviaciones típicas

fcasterrArgumentos: *obs-inic obs-fin*

Después de estimar un modelo por medio de MCO usted puede usar esta instrucción para mostrar los valores ajustados para el rango de observaciones determinado, las desviaciones típicas de estos valores ajustados y los intervalos de confianza del 95 por ciento.

Las desviaciones típicas se calculan de la forma descrita en el Capítulo 6 del libro de Wooldridge *Introductory Econometrics*. Estas tienen en cuenta dos fuentes de variabilidad: la asociada al valor esperado de la variable dependiente condicionado a los valores dados de las variables independientes, y la varianza de los residuos de la regresión.

Menú gráfico: Ventana de Modelo, /Datos del modelo/Predicciones con desviaciones típicas

freqArgumento: *var*

Presenta la distribución de frecuencias de la variable *var* (que se da por nombre o número); también muestra los resultados del contraste de normalidad de Doornik-Hansen. En modo interactivo, se presenta un gráfico de la distribución.

Menú gráfico: /Variable/Distribución de frecuencias

function

Define una función.

garch

Argumentos: *p q ; vardep [varindeps]*

Opciones: `--robust` (Desviaciones típicas robustas)
`--verbose` (muestra los detalles de las iteraciones)
`--vcv` (presenta la matriz de covarianzas)

Ejemplos: `garch 1 1 ; y`
`garch 1 1 ; y 0 x1 x2 --robust`

Estima un modelo GARCH (GARCH = Generalized Autoregressive Conditional Heteroskedasticity), o un modelo univariante o, si se especifican *varindeps*, un modelo que incluye las variables exógenas dadas. Los valores enteros *p* y *q* representan los órdenes de retardos en la ecuación de la varianza condicional.

$$h_t = \sum_{i=1}^q \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^p \beta_j h_{t-j}$$

El algoritmo GARCH de gretl es básicamente el de Fiorentini, Calzolari y Panattoni (1996), y se usa con el amable permiso del profesor Fiorentini.

Con esta instrucción están disponibles varias variantes de las estimaciones de la matriz de covarianzas de los coeficientes. Por defecto se usa el Hessiano a no ser que se indique la opción `--robust`, en cuyo caso se utiliza la matriz de covarianzas QML (White). Se pueden especificar otras posibilidades (p.ej. la matriz de información, o el estimador Bollerslev-Wooldridge) usando la instrucción [set](#).

Menú gráfico: /Modelo/Serie temporal/GARCH

genr

Argumentos: *nueva-var = fórmula*

Crea variables nuevas, normalmente como transformaciones de variables ya existentes. Ver también [diff](#), [logs](#), [lags](#), [ldiff](#) y [square](#) como atajos.

Los *operadores aritméticos* soportados son, en orden de precedencia: $\wedge$ (exponenciación); $*$, $/$ y $\%$ (módulo o resto); $+$ y $-$.

Los *operadores Booleanos* disponibles son (de nuevo, en orden de precedencia): $!$ (negación), $\&\&$ (Y lógico), $||$ (O lógico), $>$, $<$, $=$, $>=$ (mayor o igual que), $<=$ (menor o igual que) y $!=$ (distinto que). Se pueden usar operadores Booleanos para construir variables ficticias: por ejemplo $(x > 10)$ devuelve 1 si $x > 10$, y 0 en otro caso.

Las *funciones* se clasifican en:

- Funciones matemáticas estándar: `abs`, `cos`, `exp`, `int` (parte entera), `ln` (logaritmo natural: `log` es un sinónimo), `sin`, `sqrt`.
- Funciones estadísticas: `max` (valor máximo de una serie), `min` (mínimo), `mean` (media aritmética), `median`, `var` (varianza) `sd` (desviación típica), `sst` (suma de desviaciones desde la media, al cuadrado), `sum`, `cov` (covarianza), `corr` (coeficiente de correlación), `pvalue`, `sort`, `cum` (acumulación, o suma secuencial), `resample` (remuestrear una serie con reemplazamiento, con el propósito de hacer bootstrap), `hpfilt` (Filtro de Hodrick-Prescott: esta función devuelve el “componente cíclico” de la serie).
- Funciones de series temporales: `diff` (primera diferencia), `ldiff` (diferencia logarítmica, o primera diferencia del logaritmo natural). Para generar retardos de una variable *x*, se usa la sintaxis `x(-N)`, donde *N* representa la longitud deseada del retardo; para generar adelantos, se usa `x(+N)`.
- Funciones de conjunto de datos: `misszero` (reemplaza los códigos de observación ausente de una serie dada con ceros); `zeromiss` (la operación inversa a `misszero`); `nobs` (devuelve el número de observaciones válidas de una serie de datos determinada), `missing` (para

cada observación, devuelve 1 si el argumento tiene una observación ausente, y 0 en caso contrario); ok (el opuesto de missing).

- Números pseudo-aleatorios: uniform, normal.

Todas las funciones de arriba excepto cov, corr, pvalue, uniform y normal toman como único argumento o el nombre de una variable (nótese que, en una fórmula genr, no es posible referirse a las variables por su número de ID) o una expresión que se evalúa en una variable (p.ej. $\ln((x1+x2)/2)$). cov y corr requieren dos argumentos, y devuelven respectivamente la covarianza y el coeficiente de correlación entre sus argumentos. La función pvalue toma los mismos argumentos que la instrucción [pvalue](#), pero en este contexto deben introducirse comas entre los argumentos. Esta función devuelve un valor p a una cola y, en el caso de las distribuciones normal y t, es para la “cola corta”. Con la normal, por ejemplo, tanto 1,96 como -1,96 darán un resultado de aproximadamente 0,025.

uniform() y normal(), que no toman argumentos, devuelven series de números pseudo-aleatorios extraídos de la distribución uniforme (0-1) y normal estándar respectivamente (ver también la instrucción [set](#), opción seed). La series de datos uniformes se generan utilizando el Mersenne Twister; Ver Matsumoto y Nishimura (1998). La implementación la proporciona glib, si está disponible, o el código C escrito por Nishimura y Matsumoto.

para series normales se usa el método de Box y Muller (1958), tomando la entrada del Mersenne Twister.

Además de los operadores y funciones ya mencionados, hay algunos usos especiales de genr:

- genr time crea una variable de tendencia temporal (1,2,3,...) llamada time. genr index hace lo mismo, excepto que ahora la variable se denomina index.
- genr dummy crea variables ficticias hasta la periodicidad de los datos. P.ej. en el caso de datos trimestrales (periodicidad 4), el programa crea dummy_1 = 1 para el primer trimestre y 0 en los demás trimestres, dummy_2 = 1 para el segundo trimestre y 0 en los demás, y así sucesivamente.
- genr paneldum crea un conjunto de variables ficticias especiales para ser usadas con un conjunto de datos de panel — ver [panel](#).

Usando genr se pueden recuperar los valores de algunas variables internas que se definen al ejecutar una regresión, de la siguiente forma:

- \$ess: suma de cuadrados de los residuos
- \$rsq: R-cuadrado no corregido
- \$T: número de observaciones usadas
- \$df: grados de libertad
- \$trsqr: TR-cuadrado (el tamaño muestral por el R-cuadrado)
- \$sigma: desviación típica de los residuos
- \$aic: Criterio de información de Akaike
- \$bic: Criterio de información de Schwarz
- \$lnl: logaritmo de la verosimilitud (donde es aplicable)
- coeff(var): coeficiente estimado para la variable var
- stderr(var): desviación típica estimada para la variable var
- rho(i): coeficiente autorregresivo de i-ésimo orden de los residuos
- vcv(x1,x2): covarianza entre los coeficientes para las variables nombradas x1 y x2

Nota: En el programa en línea de instrucciones, las instrucciones `genr` que recuperan datos relativos a un modelo siempre hacen referencia al último modelo estimado. Esto también es cierto para el programa en modo GUI, si uno usa `genr` en la “consola de gretl” o introduce una fórmula usando la opción “Definir nueva variable” del menú de Variable en la ventana principal. Sin embargo, con el GUI, usted tiene la posibilidad de recuperar datos de cualquier modelo que actualmente se esté mostrando en una ventana (sea o no el modelo más reciente). Usted puede hacer esto bajo el menú “Datos del Modelo” de la ventana del modelo.

Las series internas `uhat` e `yhat` contienen, respectivamente, los residuos y los valores ajustados de la última regresión.

También están disponibles dos variables “internas” relacionadas con el conjunto de datos actual: `$nobs` contiene el número de observaciones del rango muestral actual (que puede ser o no igual a `$T`, el número de observaciones usadas al estimar el último modelo), y `$pd` contiene la frecuencia o periodicidad de los datos (p.ej 4 para datos trimestrales).

La variable `t` sirve como índice de las observaciones. Por ejemplo `genr dum = (t=15)` generará una variable ficticia que toma valor 1 para la observación 15, y 0 en las demás. La variable `obs` es similar pero más flexible: usted puede usarla para extraer observaciones particulares por fecha o nombre. Por ejemplo, `genr d = (obs>1986:4)` o `genr d = (obs=ÇA)`. La última forma presupone que las observaciones tienen etiquetas; la etiqueta debe ponerse entre comillas.

Pueden utilizarse valores escalares de una serie en el contexto de una fórmula `genr`, usando la sintaxis `nombre-var[obs]`. El valor `obs` puede darse por número o fecha. Ejemplos: `x[5]`, `CPI[1996:01]`. Para datos diarios `data`, se debe de usar la forma `YYYY:MM:DD`, p.ej. `ibm[1970:01:23]`.

Se puede modificar una observación individual de una serie por medio de `genr`. Para hacerlo, hay que adjuntar al nombre de la variable en el lado izquierdo de la fórmula, entre corchetes, un número de observación válido o una fecha. Por ejemplo, `genr x[3] = 30` o `genr x[1950:04] = 303.7`.

Aquí hay un par de sugerencias sobre variable ficticias:

- Supongamos que `x` está codificada con los valores 1, 2, o 3 y usted quiere tres variables ficticias, `d1 = 1` si `x = 1`, 0 en otro caso, `d2 = 1` si `x = 2`, y así sucesivamente. Para crearlas, use las instrucciones:

```
genr d1 = (x=1)
genr d2 = (x=2)
genr d3 = (x=3)
```

- Para crear `z = max(x, y)` do

```
genr d = x>y
genr z = (x*d)+(y*(1-d))
```

Menú gráfico: /Variable/Definir nueva variable

Otro acceso: Menú emergente de la ventana principal

gnuplot

Argumentos: `yvars xvar [ dumvar ]`

Opciones: `--with-lines` (usar líneas, no puntos)

`--with-impulses` (usar líneas verticales)

`--suppress-fitted` (no mostrar el ajuste mínimo-cuadrático)

`--dummy` (ver abajo)

Ejemplos: `gnuplot y1 y2 x`

`gnuplot x time --with-lines`

`gnuplot wages educ gender --dummy`

Cuadro 1.1: Ejemplos de uso de la instrucción `genr`

<i>Fórmula</i>	<i>Comentario</i>
<code>y = x1^3</code>	x1 al cubo
<code>y = ln((x1+x2)/x3)</code>	
<code>z = x>y</code>	$z(t) = 1$ si $x(t) > y(t)$, en otro caso 0
<code>y = x(-2)</code>	x retardada 2 periodos
<code>y = x(2)</code>	x adelantada 2 periodos
<code>y = diff(x)</code>	$y(t) = x(t) - x(t-1)$
<code>y = ldiff(x)</code>	$y(t) = \log x(t) - \log x(t-1)$, la tasa de crecimiento instantánea de x
<code>y = sort(x)</code>	ordena x en orden creciente y lo guarda en y
<code>y = -sort(-x)</code>	ordena x en orden decreciente
<code>y = int(x)</code>	trunca x y guarda su valor entero como y
<code>y = abs(x)</code>	guarda los valores absolutos de x
<code>y = sum(x)</code>	suma los valores de x excluyendo los valores ausentes, con entradas -999
<code>y = cum(x)</code>	acumulación: $y_t = \sum_{\tau=1}^t x_{\tau}$
<code>aa = \$ess</code>	hace aa igual a la suma de cuadrados de los residuos de la última regresión
<code>x = coeff(sqft)</code>	guarda el coeficiente estimado de la variable <code>sqft</code> de la última regresión
<code>rho4 = rho(4)</code>	guarda el coeficiente autorregresivo de 4º orden del último modelo (supone un modelo ar)
<code>cvx1x2 = vcv(x1, x2)</code>	guarda la covarianza de los coeficientes estimados de las variables x1 y x2 del último modelo
<code>foo = uniform()</code>	variable pseudo-aleatoria uniforme en el rango 0-1
<code>bar = 3 * normal()</code>	variable pseudo-aleatoria normal, $\mu = 0$, $\sigma = 3$
<code>samp = !missing(x)</code>	= 1 para las observaciones en las que no hay valores ausentes de x.

Sin la opción `--dummy`, las variables *yvars* se representan contra *xvar*. Con `--dummy`, *yvar* se representa contra *xvar* con los puntos pintados de diferentes colores dependiendo de si el valor de *dumvar* es 1 o 0.

La variable `time` se comporta de manera especial: si no existe, entonces se generará automáticamente.

En modo interactivo los resultados se muestran inmediatamente. En modo batch se escribe un fichero de instrucciones `gnuplot` con un nombre que sigue el modelo `gpttmpN.plt`, empezando con `N = 01`. Los gráficos actuales se pueden generar más tarde usando `gnuplot` (bajo MS Windows, `wgnuplot`).

Hay una nueva opción disponible en esta instrucción: después de la especificación de las variables a representar y las opciones (si se elige alguna), usted puede añadir instrucciones literales de `gnuplot` para controlar la apariencia del gráfico (por ejemplo, poner el título del gráfico y/o el rango de los ejes). Estas instrucciones deberían incluirse entre paréntesis y cada instrucción `gnuplot` debe terminar con un punto y coma. Se puede usar una barra invertida (`/`) para continuar un conjunto de instrucciones `gnuplot` sobre más de una línea. He aquí un ejemplo de la sintaxis:

```
{ set title 'Mi título'; set yrange [0:1000]; }
```

Menú gráfico: `/Datos/Gráficos`

Otro acceso: Menú emergente de la ventana principal, botón de gráfico de la barra de herramientas

graphArgumentos: *yvars xvar*Opción: `--tall` (usar 40 filas)

Gráficos ASCII. Las variables *yvars* (que pueden determinarse por nombre o número) se representan con respecto a *xvar* usando símbolos ASCII. La opción `--tall` producirá un gráfico con 40 filas y 60 columnas. Sin ella, el gráfico será de 20 por 60 (para salida por pantalla). Ver también [gnuplot](#).

hausman

Este contraste sólo está disponible después de haber estimado un modelo mediante la instrucción [ols](#) (ver también [panel](#) y [setobs](#)). Contrasta el modelo simple combinado contra sus alternativas principales, el modelo de efectos fijos y el modelo de efectos aleatorios.

El modelo de efectos fijos añade una variable ficticia para todas menos una de las unidades de sección cruzada, permitiendo al intercepto de la regresión variar a través de estas unidades. Se presenta un estadístico *F* para el contraste de significación conjunta de estas variables ficticias. El modelo de efectos aleatorios descompone la varianza residual en dos partes, una parte específica de las unidades de sección cruzada y la otra específica de cada observación particular. (Este estimador sólo puede computarse si el número de unidades de sección cruzada es mayor que el número de parámetros a estimar.) El estadístico LM de Breusch-Pagan sirve para contrastar la hipótesis nula (de que el estimador MCO combinado es el adecuado) contra la alternativa de efectos aleatorios.

El modelo de MCO combinados puede ser rechazado contra ambas alternativas, el modelo de efectos fijos y el de efectos aleatorios. Si el error específico de unidad o grupo está incorrelacionado con las variables independientes, el estimador de efectos aleatorios es más eficiente que el estimador de efectos fijos; en caso contrario el estimador de efectos aleatorios sería inconsistente y sería preferible el estimador de efectos fijos. La hipótesis nula para el contraste de Hausman es que el error específico de grupo no está tan correlacionado (así que es preferible el modelo de efectos aleatorios). Un valor *p* bajo para este contraste es un síntoma en contra del modelo de efectos aleatorios y a favor del modelo de efectos fijos.

Menú gráfico: Ventana de Modelo, /Contrastes/Diagnósticos de panel

hccmArgumentos: *vardep varindeps*Opción: `--vcv` (presenta la matriz de covarianzas)

Matriz de covarianzas consistente ante heterocedasticidad: esta instrucción ejecuta una regresión en la que los coeficientes se estiman mediante un procedimiento MCO estándar, pero las desviaciones típicas de los coeficientes estimados se calculan de una manera que es robusta ante heterocedasticidad, en concreto usando el procedimiento “jackknife” de MacKinnon-White.

Menú gráfico: /Modelo/HCCM

help

Proporciona una lista de las instrucciones disponibles. `help instrucción` describe la *instrucción* (p.ej. `help smpl`). Vd. puede escribir `man` en lugar de `help` si lo desea.

Menú gráfico: /Ayuda

hiluArgumentos: *vardep varindeps*Opción: `--vcv` (presenta la matriz de covarianzas)

Calcula estimaciones de los parámetros del modelo especificado usando el procedimiento de búsqueda de Hildreth-Lu search procedure (refinado mediante el procedimiento CORC). Este

procedimiento está diseñado para corregir las estimaciones teniendo en cuenta la correlación serial del término de error. La suma de cuadrados de los residuos del modelo transformado se representa con respecto al valor de rho desde -0.99 hasta 0.99 .

Menú gráfico: /Modelo/Serie temporal/Hildreth-Lu

hsk

Argumentos: *vardep varindeps*

Opción: *--vcv* (presenta la matriz de covarianzas)

Se calcula una regresión por MCO y se guardan los residuos. El logaritmo del cuadrado de los residuos entonces pasa a ser la variable dependiente en una regresión auxiliar, en cuyo lado derecho están las variables independientes originales más sus cuadrados. Los valores ajustados en la regresión auxiliar se usan entonces para construir una serie de ponderaciones y el modelo original se reestima utilizando mínimos cuadrados ponderados. El resultado final se presenta en un informe.

La serie de ponderaciones se forma como $1/\sqrt{e^{y^*}}$, donde y^* denota los valores ajustados mediante la regresión auxiliar.

Menú gráfico: /Modelo/Corregido de Heterocedasticidad

if

Control de flujo para la ejecución de instrucciones. La sintaxis es:

```
if condición
    instrucciones1
else
    instrucciones2
endif
```

condición debe ser una expresión Booleana, para su sintaxis ver [genr](#). El bloque *else* es opcional; los bloques *if ... endif* pueden estar anidados.

info

Presenta cualquier información suplementaria que se haya guardado con el fichero de datos actual.

Menú gráfico: /Datos/Leer información

Otro acceso: Ventanas de navegación de datos

setinfo

Argumentos: *nombre-var -d descripción -n nombre a mostrar*

Ejemplo: `label x1 -d "Descripción de x1n" "Nombre en los gráficos"`

Establece la etiqueta descriptiva de la variable determinada (si se da la opción *-d*, seguida de una cadena de caracteres entre comillas) y/o el “nombre a mostrar” para la variable (si se da la opción *-n*, seguida de una cadena de caracteres entre comillas). Si una variable tiene un “nombre a mostrar”, se usará éste al generar los gráficos.

Menú gráfico: /Variable/Editar atributos

Otro acceso: Menú emergente de la ventana principal

labels

Presenta las etiquetas informativas de cualquier variable que haya sido generada usando la instrucción *genr*, y cualquier variable añadida al conjunto de datos por medio del GUI.

lad

Argumentos: *vardep varindeps*

Calcula una regresión que minimiza la suma de las desviaciones absolutas de los valores observados de la variable dependiente con respecto a los ajustados. Las estimaciones de los coeficientes se derivan usando el algoritmo simplex de Barrodale-Roberts; se muestra un aviso si la solución no es única. Las desviaciones típicas se derivan utilizando el procedimiento bootstrap con 500 extracciones.

Menú gráfico: /Modelo/Mínima desviación absoluta

lags

Argumento: *listavar*

Crea variables nuevas que son valores retardados de cada una de las variables de *listavar*. El número de variables retardadas es igual a la periodicidad. Por ejemplo, si la periodicidad es 4 (trimestral), la instrucción `lags x` crea $x_{-1} = x(t-1)$, $x_{-2} = x(t-2)$, $x_{-3} = x(t-3)$ y $x_{-4} = x(t-4)$. Lo mismo para y . Estas variables deben ser referenciadas de manera exacta, es decir, con el carácter de subrayado.

Menú gráfico: /Datos/Añadir variables/retardos de las variables seleccionadas

ldiff

Argumento: *listavar*

Calcula la primera diferencia del logaritmo natural de cada variable de *listavar* y guarda el resultado en una nueva variable con el prefijo `ld_`. Así `ldiff x` crea las nuevas variables $ld_x = \ln(x_t) - \ln(x_{t-1})$ y $ld_y = \ln(y_t) - \ln(y_{t-1})$.

Menú gráfico: /Datos/Añadir variables/diferencias de logaritmos de las variables seleccionadas

leverage

Opción: `--save` (guardar las variables)

Debe ejecutarse inmediatamente después de una instrucción MCO. Calcula el apalancamiento (h , que debe estar entre 0 y 1) para cada punto de datos de la muestra sobre la que se estimó el último modelo. Presenta el residuo (u) de cada observación junto a su apalancamiento y una medida de su influencia sobre las estimaciones, $uh/(1-h)$. Los “puntos palanca” para los cuales el valor de h es mayor que $2k/n$ (donde k es el número de parámetros estimados y n es el tamaño muestral) se marcan con un asterisco. Para más detalles sobre los conceptos de apalancamiento e influencia ver Davidson and MacKinnon (1993, Capítulo 2).

También se presentan los valores DFFITS: estos son los “residuos studentizados” (residuos predichos divididos por sus desviaciones típicas) multiplicados por $\sqrt{h/(1-h)}$. Para más detalles sobre residuos studentizados y DFFITS ver el libro de G. S. Maddala, *Introduction to Econometrics*, capítulo 12; y también Belsley, Kuh y Welsch (1980). En resumen, un “residuo predicho” es la diferencia entre el valor observado de la variable dependiente en la observación t y el valor ajustado para la observación t obtenido de una regresión en la que se omite esa observación (o se añade una variable ficticia con valor 1 sólo para la observación t); el residuo studentizado se obtiene dividiendo el residuo predicho por su desviación típica.

Si con esta instrucción se da la opción `--save`, entonces los valores del apalancamiento, influencia y DFFITS se añaden al conjunto de datos actual.

Menú gráfico: Ventana de Modelo, /Contrastes/observaciones influyentes

lmtest

Opciones: --logs (no linealidad, logaritmos)
 --autocorr (correlación serial)
 --squares (no linealidad, cuadrados)
 --white (heterocedasticidad (Contraste de White))

Debe ejecutarse justo después de una instrucción mco. Realiza alguna combinación de lo siguiente: contrastes de Multiplicador de Lagrange de no linealidad (logaritmos y cuadrados), contraste de heterocedasticidad de White, y el contraste LMF de correlación serial hasta el orden de la periodicidad (ver Kiviet, 1986). También se presentan los correspondientes coeficientes de la regresión auxiliar. Ver Ramanathan, Capítulos 7, 8, y 9 para más detalles. En el caso del contraste de White, sólo se usan los cuadrados de las variables independientes y no sus productos cruzados. En el caso del contraste de autocorrelación, si el valor p del estadístico LMF es menor que 0,05 (y el modelo no se había estimado originariamente con desviaciones típicas robustas) entonces se calculan y se presentan las desviaciones típicas robustas ante correlación serial. Para detalles sobre el cálculo de estas desviaciones típicas ver Wooldridge (2002, Capítulo 12).

Menú gráfico: Ventana de Modelo, /Contrastes

logistic

Argumentos: *vardep varindeps* [*y*max=valor]
 Opción: --vcv (mostrar la matriz de covarianzas)
 Ejemplos: *logistic y const x*
*logistic y const x y*max=50

Regresión logística: desarrolla una regresión MCO utilizando la transformación logística de la variable dependiente,

$$\log\left(\frac{y}{y^* - y}\right)$$

La variable dependiente debe ser estrictamente positiva. Si es una fracción decimal, entre 0 y 1, por defecto se usa un valor y^* (el máximo asintótico de la variable dependiente) of 1. Si la variable dependiente es un porcentaje, entre 0 y 100, por defecto y^* es 100. Si Vd. desea establecer otro valor máximo utilice el parámetro opcional *y*max=valor después de la lista de regresores. El valor proporcionado debe ser mayor que todos los valores observados de la variable dependiente.

Los valores ajustados y los residuos de la regresión se transforman automáticamente usando

$$y = \frac{y^*}{1 + e^{-x}}$$

donde x representa o un valor ajustado o un residuo de la regresión MCO usando la variable dependiente transformada. Los valores que se presentan son así comparables con los valores originales de la variable dependiente.

Nótese que si la variable dependiente es binaria, en lugar de esto se debería usar la instrucción *logit*.

Menú gráfico: /Modelo/Logístico

logit

Argumentos: *vardep varindeps*
 Opción: --vcv (mostrar la matriz de covarianzas)

Regresión binomial *logit*. La variable dependiente debería ser una variable binaria. Se obtienen las estimaciones máximo-verosímiles de los coeficientes de las variables *varindeps* por medio del método EM (Expectation-Maximization method, ver Ruud, 2000, Capítulo 27). Como el modelo no es lineal las pendientes dependen de los valores de las variables independientes: las

poendientes que se presentan se han evaluado en la media de dichas variables. La hipótesis de que todos los coeficientes, aparte de la constante, son cero se contrasta mediante el estadístico chi-cuadrado.

Si Vd. desea utilizar logit para el análisis de proporciones (donde la variable dependiente, para cada observación, es la proporción de casos que tienen una determinada característica, en lugar de un 1 o un 0 indicando si la característica está presente o no) no debería usar la instrucción `logit`, sino mejor debería construir la variable logit (p.ej. `genr lgt_p = log(p/(1 - p))`) y usarla como variable dependiente en una regresión MCO. Ver Ramanathan, Capítulo 12.

Menú gráfico: /Modelo/Logit

logs

Argumento: *listavar*

Se obtiene el logaritmo natural de cada una de las variables de *listavar* y el resultado se guarda en una nueva variable con el prefijo `l_` que es “le” y carácter de subrayado. `logs x` y crea las nuevas variables `l_x = ln(x)` y `l_y = ln(y)`.

Menú gráfico: /Datos/Añadir variables/logaritmos de las variables seleccionadas

loop

Argumento: *control*

Opción: `--progressive` (permite formas especiales de ciertas instrucciones)

Ejemplos: `loop 1000`

`loop 1000 --progressive`

`loop while essdiff >.00001`

`loop for i=1991..2000`

`loop for (r=-.99; r<=.99; r+=.01)`

El parámetro *control* debe tomar una de las cuatro formas posibles, tal y como se muestra en los ejemplos: un número de veces entero a repetir las instrucciones del bucle; “`while`” más una condición numérica; o “`for`” más un rango de valores para la variable índice interna *i*; o “`for`” más tres expresiones entre paréntesis, separadas por puntos y comas. En la última forma, la expresión del lado izquierdo inicializa una variable, la del medio establece una condición para que continúen las iteraciones, y la de la derecha determina un incremento o decremento a aplicar al comienzo de la segunda y siguientes iteraciones. (Esta es una forma restringida de la instrucción `for` del lenguaje de programación C.)

Esta instrucción abre un modo de ejecución especial en el cual el programa acepta instrucciones a ejecutar repetidamente. Dentro de un bucle sólo se pueden utilizar ciertas instrucciones: `genr`, `ols`, `print`, `printf`, `pvalue`, `sim`, `smpl`, `store`, `summary`, `if`, `else` y `endif`. Se sale del modo de introducción de instrucciones de bucle con `endloop`: en este punto se ejecutan todas las instrucciones que están en la cola del bucle.

Ver the *Gretl User's Guide* para más detalles y ejemplos. El efecto de la opción `--progressive` (que está diseñada para su uso en simulaciones de Monte Carlo) se explica allí.

meantest

Argumentos: *var1 var2*

Opción: `--unequal-vars` (suponer que las varianzas son diferentes)

Calcula el estadístico *t* para la hipótesis nula de que las medias poblacionales de las variables *var1* y *var2* son iguales, y presenta su valor *p*. Por defecto se calcula el estadístico bajo el supuesto de que las varianzas de las dos variables son iguales; con la opción `--unequal-vars` se suponen varianzas distintas. Esto implicará una diferencia en el estadístico de contraste sólo si hay un número diferente de valores no ausentes para las dos variables.

Menú gráfico: /Datos/Diferencia de medias

modeltab

Argumentos: *add* o *show* o *free*

Sirve para manipular la “tabla de modelos” de gretl. Ver the *Gretl User’s Guide* para más detalles. Las sub-instrucciones tienen los siguientes efectos: *add* añade el último modelo estimado a la tabla de modelos, si es posible; *show* presenta la tabla de modelos en una ventana; y *free* vacía la tabla.

Menú gráfico: Ventana de sesión, icono de Tabla de modelos

mpols

Argumentos: *vardep varindeps*

Computa las estimaciones MCO para el modelo especificado, usando aritmética de punto flotante con precisión múltiple. Esta instrucción sólo está disponible si gretl se compila con soporte para la biblioteca Gnu de Precisión Múltiple (GMP).

Para estimar un ajuste polinomial usando aritmética de precisión múltiple para generar las potencias necesarias de la variable independiente use por ejemplo la forma *mpols y 0 x ; 2 3 4*. Esto realiza una regresión de *y* sobre *x*, *x* cuadrado, *x* al cubo y *x* a la cuarta potencia. Es decir, los números a la derecha del punto y coma (que deben ser enteros positivos) determinan las potencias de *x* a utilizar. Si se especifica más de una variable independiente, ha de considerarse que la última antes del punto y coma es la variable que va a ser elevada a varias potencias.

Menú gráfico: /Modelo/MCO de alta precisión

nls

Argumentos: *función derivadas*

Opción: *--vcv* (mostrar la matriz de covarianzas)

Desarrolla la estimación por Mínimos Cuadrados No Lineales (NLS) utilizando una versión modificada del algoritmo de Levenberg-Marquardt. El usuario debe suministrar la especificación de una función. Los parámetros de esta función deben ser declarados antes y hay que asignarles unos valores iniciales (usando la instrucción *genr*) antes de la estimación. Opcionalmente, el usuario puede especificar las derivadas de la función de regresión con respecto a cada uno de los parámetros; si no se proporcionan las derivadas analíticas se computa una aproximación numérica al Jacobiano.

Es más sencillo mostrar lo que se necesita con un ejemplo. Lo siguiente es un lote de instrucciones completo para estimar la función de consumo no lineal definida en el libro de William Greene *Econometric Analysis* (Capítulo 11 de la 4ª edición inglesa, o Capítulo 9 de la 5ª). Los números a la izquierda de las líneas son para referencia y no son parte de las instrucciones. Hay que señalar que la opción *--vcv*, para presentar la matriz de covarianzas de las estimaciones de los parámetros se escribe junto a la instrucción final, *end nls*.

```

1  open greene11_3.gdt
2  ols C 0 Y
3  genr alfa = coeff(0)
4  genr beta = coeff(Y)
5  genr gamma = 1.0
6  nls C = alfa + beta * Y^gamma
7  deriv alfa = 1
8  deriv beta = Y^gamma
9  deriv gamma = beta * Y^gamma * log(Y)
10 end nls --vcv

```

Muchas veces es conveniente inicializar los parámetros haciendo referencia a un modelo lineal relacionado con el actual; esto se realiza aquí en las líneas 2 a 5. A los parámetros *alfa*, *beta* y *gamma* se les podría asignar cualesquiera valores iniciales (no necesariamente basados en un modelo estimado por MCO), aunque la convergencia del procedimiento de MC no lineales no está garantizada para un punto de inicio arbitrario.

Las instrucciones de MC no lineales actuales ocupan las líneas 6 a 10. En la línea 6 se invoica la instrucción `nls`: se especifica una variable dependiente seguida de un signo igual y después la especificación de una función. La sintaxis para la expresión de la derecha es la misma que para la instrucción `genr`. Las tres siguientes líneas determinan las derivadas de la función de regresión con respecto a cada uno de los parámetros. Cada línea comienza con la palabra `deriv`, da el nombre de un parámetro, un signo igual y una expresión por la que se calcula la derivada (de nuevo, la sintaxis aquí es la misma que para `genr`). Estas líneas `deriv` son opcionales, pero es conveniente proporcionarlas, si es posible. La línea 10, `end nls`, completa la instrucción e invoca la estimación.

Para más detalles sobre la estimación por MC. no lineales ver the *Gretl User's Guide*.

Menú gráfico: /Modelo/Mínimos cuadrados no lineales

nulldata

Argumento: *largura de la serie*

Ejemplo: `nulldata 500`

Forma un conjunto de datos “vacío”, que contiene sólo una constante y una variable índice, con periodicidad 1 y el número de observaciones especificado. Esto puede usarse para hacer simulaciones: algunas de las instrucciones `genr` (p.ej. `genr uniform()`, `genr normal()`) generan, desde cero, datos ficticios para rellenar el conjunto de datos. Esta instrucción puede ser útil también junto a la instrucción `loop`. Ver también la opción “seed” de la instrucción `set`.

Menú gráfico: /Archivo/Crear conjunto de datos

ols

Argumentos: *vardep varindeps*

Opciones: `--vcv` (mostrar la matriz de covarianzas)
`--robust` (desviaciones típicas robustas)
`--quiet` (suprime la presentación de los resultados)
`--no-df-corr` (suprime la corrección de grados de libertad)
`--print-final` (ver más abajo)

Ejemplos: `ols 1 0 2 4 6 7`
`ols y 0 x1 x2 x3 --vcv`
`ols y 0 x1 x2 x3 --quiet`

Calcula las estimaciones de mínimos cuadrados ordinarios (MCO) con *vardep* como variable dependiente y siendo *varindeps* la lista de variables independientes. Las variables se pueden determinar por nombre o por su número; hay que usar el número cero para el término constante.

Además de las estimaciones de los coeficientes y de las desviaciones típicas, el programa también presenta los valores p de los estadísticos *t* (a dos colas) y *F*. Un valor p inferior a 0,01 indica significatividad al nivel del 1 por ciento y se denota mediante ***. ** indica un nivel de significatividad entre el 1 y 5 por ciento y * indica significatividad entre el 5 y 10 por ciento. También se presentan algunos estadísticos de selección de modelos (se describen en Ramanathan, Sección 4.3).

Si se da la opción `--no-df-corr`, no se aplica la corrección habitual de grados de libertad al calcular la estimación de la varianza de las perturbaciones (y así tampoco en la estimación de las desviaciones típicas de los estimadores de los parámetros).

La opción `--print-final` es aplicable sólo en el contexto de un bucle (ver `loop`). Se encarga de que las regresiones se ejecuten “silenciosamente” en todas las iteraciones excepto la iteración final del bucle. Ver the *Gretl User's Guide* para más detalles.

Algunas variables que se definen internamente al ejecutar la instrucción `ols`, pueden recuperarse mediante la instrucción `genr`, teniendo en cuenta que `genr` debe invocarse inmediatamente después de la instrucción `ols`.

Mediante la instrucción [set](#) puede ajustarse la fórmula específica que se usa para generar las desviaciones típicas robustas (cuando se da la opción `--robust`).

Menú gráfico: /Modelo/Mínimos cuadrados ordinarios

Otro acceso: Botón beta con sombrero en la barra de herramientas

omit

Argumento: *listavar*

Opciones: `--vcv` (mostrar la matriz de covarianzas)
`--quiet` (no mostrar las estimaciones del modelo reducido)

Ejemplo: `omit 5 7 9`

Esta instrucción debe invocarse justo después de una instrucción de estimación. Las variables seleccionadas se omiten del modelo anterior y se estima el nuevo modelo. Si se omite más de una variable, se presentará el estadístico F de Wald para las variables omitidas y su valor p (sólo para el método MCO). Un valor p inferior a 0,05 indica que los coeficientes son conjuntamente significativos al nivel de significación del 5 por ciento.

Si se da la opción `--quiet` el resultado que se muestra es sólo el contraste de significación conjunta de las variables omitidas, en caso contrario, también se presentan las estimaciones del modelo reducido. En este último caso, la opción `--vcv` hace que también se muestre la matriz de covarianzas de los coeficientes estimados.

Menú gráfico: ventana de Modelo, /Contrastes/omitir variables

omitfrom

Argumentos: *ID-modelo listavar*

Opción: `--quiet` (no mostrar las estimaciones del modelo reducido)

Ejemplo: `omitfrom 2 5 7 9`

Funciona igual que [omit](#), excepto que aquí Vd. tiene que especificar un modelo anterior (usando su número ID identificador de modelo, que se presenta al principio de los resultados del modelo) que se toma como base para omitir las variables. En el ejemplo de arriba se omiten del Modelo 2 las variables con números 5, 7 y 9.

Menú gráfico: Ventana de Modelo, /Contrastes/omitir variables

open

Argumento: *fichero-de-datos*

Abre un fichero de datos. Si ya hay un fichero de datos abierto, se reemplaza por el que ahora se abre. El programa tratará de detectar el formato del fichero de datos (formato nativo de gretl, texto plano, CSV o BOX1).

Esta instrucción también se puede usar para abrir una base de datos (de formato gretl o RATS 4.0). En este caso, después debería ejecutarse la instrucción [data](#) para extraer de ella series determinadas.

Menú gráfico: /Archivo/Abrir datos

Otro acceso: Arrastrar un fichero de datos sobre gretl (en MS Windows o Gnome)

outfile

Argumentos: *nombre-de-fichero opción*
 Opciones: --append (añadir a un fichero)
 --close (cerrar el fichero)
 --write (sobreescribir el fichero)
 Ejemplos: outfile --write regres.txt
 outfile --close

Envía los resultados a *nombre-de-fichero*, hasta nuevo aviso. Use la opción --append para añadir los resultados a un fichero ya existente o --write para empezar un nuevo fichero (o sobreescribir uno existente). De esta forma sólo se puede abrir un fichero en cada momento.

La opción --close se usa para cerrar un fichero de resultados que estuviera abierto previamente. Los resultados se enviarán entonces a la salida por defecto.

En el primer ejemplo de arriba, se abre el fichero `regres.txt` para escritura, y en el segundo se cierra. Esta secuencia tendría sentido sólo si se ejecutaran algunas instrucciones antes de --close. Por ejemplo si se invocara una instrucción `ols`, sus resultados irían a `regres.txt` en lugar de a la pantalla.

panel

Opciones: --cross-section (secciones cruzadas apiladas)
 --time-series (series temporales apiladas)

Indica que el conjunto de datos actual debe ser interpretado como un panel (combinando secciones cruzadas y series temporales). Por defecto, o usando la opción --time-series, se considera que el conjunto de datos está en la forma de series temporales apiladas (los sucesivos bloques de datos contienen series temporales para cada unidad de sección cruzada). Con la opción --cross-section, el conjunto de datos se lee como secciones cruzadas apiladas (los sucesivos bloques de datos contienen secciones cruzadas para cada periodo temporal). Ver también [setobs](#).

Menú gráfico: /Muestra/Interpretar como panel

pca

Argumento: *listavar*
 Opciones: --save-all (guardar todos los componentes)
 --save (guardar los componentes más importantes)

Análisis de componentes principales. Presenta los valores propios de la matriz de correlación de las variables de la lista *listavar* y la proporción de la varianza conjunta explicada por cada uno de los componentes. También presenta los vectores propios correspondientes (o “ponderaciones de los componentes”).

Si se da la opción --save, los componentes con valores propios mayores que 1,0 se guardan como variables en el conjunto de datos, con nombres PC1, PC2 y así sucesivamente. Estas variables artificiales se forman como la suma de (ponderaciones de los componentes) por (X_i estandarizadas), donde X_i denota la i -ésima variable de la lista *listavar*.

Si se da la opción --save-all, se guardan todos los componentes de la manera que se ha descrito arriba.

Menú gráfico: Menú emergente de la ventana principal (selección múltiple)

pergm

Argumento: *nombre-var*
 Opción: --bartlett (usar la ventana de retardos de Bartlett)

Calcula y presenta (y si no se ejecuta en modo batch, representa gráficamente) el espectro de la variable especificada. Sin la opción `--bartlett` se ofrece el periodograma muestral; con dicha opción, se utiliza una ventana de retardos de Bartlett de longitud $2\sqrt{T}$ (donde T es el tamaño muestral) para estimar el espectro (ver Capítulo 18 del libro de Greene *Econometric Analysis*). Cuando se presenta el periodograma muestral, también se proporciona un contraste t sobre integración fraccional de la serie (“memoria larga”): la hipótesis nula es que el orden de integración es cero.

Menú gráfico: /Variable/Espectro

Otro acceso: Menú emergente de la ventana principal (selección simple)

plot

Argumento: *listavar*

Opción: `--one-scale` (forzar una sólo escala)

Representa gráficamente los valores de las variables especificadas, para el rango muestral que esté actualmente activo, utilizando símbolos ASCII. Cada línea indica una observación y los valores se representan horizontalmente. Por defecto, las variables se cambian a la escala más adecuada. Ver también [gnuplot](#).

print

Argumentos: *listavar* o *cadena-literal*

Opciones: `--byobs` (por observaciones)
`--ten` (usar 10 dígitos significativos)

Ejemplos: `print x1 x2 --byobs`
`print "Esto es una cadena de caracteres"`

Si se da una lista *listavar*, presenta los valores de las variables especificadas; si no se da ninguna lista, muestra los valores de todas las variables del conjunto de datos actual. Si se da la opción `--byobs`, los datos se presentan ‘por observación’, en caso contrario, se presentan ‘por variable’. Si se da la opción `--ten`, los datos se presentan por variable mostrando 10 dígitos significativos.

Si el argumento de la instrucción `print` es una cadena literal (que debe comenzar con comillas dobles, `"`), la cadena se presenta tal y como está. Ver también [printf](#).

Menú gráfico: /Datos/Mostrar valores

printf

Argumentos: *format args*

Presenta valores escalares bajo el control de una cadena de caracteres (proporcionando así un subconjunto de las utilidades de la instrucción `printf()` del lenguaje de programación C). Los formatos que se reconocen son `%g` y `%f`, en cada caso con los diferentes modificadores disponibles en C. Ejemplos: el formato `%.10g` presenta un valor con 10 dígitos significativos; `%12.6f` presenta un valor con 6 decimales y con una anchura de 12 caracteres.

La propia cadena de caracteres que indica el formato debe de estar incluida entre comillas. Los valores a mostrar deben seguir el formato de la cadena, separados por comas. Estos valores deberían de tener la forma de (a) nombres de variables del conjunto de datos, o (b) expresiones que sean válidas para la instrucción `genr`. En el siguiente ejemplo se muestran los valores de dos variables más el de una expresión calculada:

```
ols 1 0 2 3
genr b = coeff(2)
genr dt_b = stderr(2)
printf "b = %.8g, desviación típica %.8g, t = %.4f\n", b, dt_b, b/dt_b
```

La largura máxima de una cadena de formato es de 127 caracteres. Se reconocen las secuencias de escape `\n` (nueva línea), `\t` (tabulador), `\v` (tabulador vertical) y `\\` (backslash literal). Para presentar un signo de porcentaje literal use `%%`.

probit

Argumentos: *vardep varindeps*

Opción: `--vcv` (mostrar la matriz de covarianzas)

La variable dependiente debería ser una variable binaria. Se obtienen las estimaciones máximo-verosímiles de los coeficientes de las variables *varindeps* por medio de mínimos cuadrados iterativos (el método EM o Expectation-Maximization). Como el modelo no es lineal las pendientes dependen de los valores de las variables independientes: las pendientes que se presentan están evaluadas en las medias de dichas variables. El estadístico chi-cuadrado sirve para contrastar la hipótesis nula de que todos los coeficientes, excepto la constante, son cero.

El análisis Probit de proporciones no está aún implementado en gretl.

Menú gráfico: /Modelo/Probit

pvalue

Argumentos: *dist [params] xval*

Ejemplos: `pvalue z zscore`

`pvalue t 25 3.0`

`pvalue X 3 5.6`

`pvalue F 4 58 fval`

`pvalue G xbar varx x`

Calcula el área a la derecha de *xval* en la distribución especificada (*z* para Gaussiana, *t* para la *t* de Student, *X* para la chi-cuadrado, *F* para la *F* y *G* para la distribución gamma). Para las distribuciones *t* y chi-cuadrado deben proporcionarse los grados de libertad; para la *F* hay que proporcionar los grados de libertad del numerador y del denominador; y para la distribución gamma se necesitan la media y la varianza.

Menú gráfico: /Utilidades/buscador de valores-p

pwe

Argumentos: *vardep varindeps*

Opción: `--vcv` (mostrar la matriz de covarianzas)

Ejemplo: `pwe 1 0 2 4 6 7`

Calcula estimaciones de los parámetros utilizando el procedimiento de Prais-Winsten, un método de Mínimos cuadrados generalizados factibles que está diseñado para tener en cuenta la autocorrelación de primer orden del término de error. El procedimiento es iterativo, igual que [corc](#); la diferencia es que mientras el método de Cochrane-Orcutt desperdicia la primera observación, el de Prais-Winsten la utiliza. Para más detalles, ver por ejemplo el Capítulo 13 del libro de Greene *Econometric Analysis* (2000).

Menú gráfico: /Modelo/Serie temporal/Prais-Winsten

quit

Sale del programa, dándole a Vd. la opción de guardar las órdenes y resultados de su sesión al salir.

Menú gráfico: /Archivo/Salir

rename

Argumentos: *var-número nuevo-nombre*

Cambia el nombre de la variable con número de identificación *var-número* a *nuevo-nombre*. El número *var-número* debe estar entre el 1 y el número de variables en el conjunto de datos. El nuevo nombre debe tener como máximo 8 caracteres, debe empezar con una letra y debe estar formado sólo por letras dígitos y el carácter de subrayado (_). character.

Menú gráfico: /Variable/Editar atributos

Otro acceso: Menú emergente de la ventana principal (selección simple)

reset

Debe ejecutarse después de la estimación de un modelo vía MCO. Realiza el contraste de especificación de modelos (no linealidad) RESET de Ramsey. Para ello añade a la regresión el cuadrado y el cubo de los valores ajustados y calcula el estadístico F para la hipótesis nula de que los parámetros de las dos variables añadidas son cero.

Menú gráfico: Ventana de Modelo, /Contrastes/Contraste RESET de Ramsey

restrict

Evalúa un conjunto de restricciones sobre los parámetros del último modelo estimado. En modo "guión de instrucciones", el conjunto de restricciones debe incluirse entre "restrict" y "end restrict", pero en la caja de diálogo sobre restricciones (en el entorno gráfico) pueden omitirse esas líneas.

Cada restricción del conjunto debe expresarse como una ecuación, con una combinación lineal de los parámetros a la izquierda y un valor numérico a la derecha del signo igual. Los parámetros se referencian en la forma b_N , donde N representa la posición en la lista de regresores, comenzando en cero. Por ejemplo, b_1 denota al segundo parámetro de la regresión.

El segundo y siguientes términos b_N de una ecuación pueden ir premultiplicados por un número, utilizando $*$ para representar la multiplicación, por ejemplo $3.5*b_4$.

He aquí un ejemplo de un conjunto de restricciones:

```
restrict
b1 = 0
b2 - b3 = 0
b4 + 2*b5 = 1
end restrict
```

Las restricciones se evalúan mediante un contraste F de Wald, usando la matriz de covarianzas de los coeficientes del modelo en cuestión.

Menú gráfico: Ventana de Modelo, /Contrastes/Restricciones lineales

rhodiff

Argumentos: *listarho ; listavar*

Ejemplos: *rhodiff .65 ; 2 3 4*

rhodiff r1 r2 ; x1 x2 x3

Crea las versiones rho-diferenciadas de las variables (dadas por nombre o número) de la lista *listavar* y las añade al conjunto de datos, usando el sufijo # para las nuevas variables. Dada la variable v_1 una variable de *listavar*, y las entradas r_1 y r_2 de *listarho*, se crea

$$v_{1\#} = v_1 - r_1*v_1(-1) - r_2*v_1(-2)$$

Las entradas de *listarho* pueden darse como valores numéricos o como nombres de variables definidas previamente.

rmplot

Argumento: *nombre-var*

Gráfico Rango-media plot: esta instrucción crea un sencillo gráfico para ayudar a decidir si una serie temporal, $y(t)$, tiene varianza constante o no. Se toma la muestra completa $t=1,...,T$ y se divide en pequeñas submuestras de tamaño arbitrario k . La primera submuestra se forma con $y(1),...,y(k)$, la segunda con $y(k+1), ..., y(2k)$, y así sucesivamente. Para cada submuestra se calcula la media muestral y el rango (= el máximo menos el mínimo), y se forma un gráfico con las medias en el eje horizontal y los rangos en el vertical. Así, cada submuestra está representada por un punto en este plano. Si la varianza de la serie es constante, se espera que los rangos de las submuestras sean independientes de las medias; si vemos que los puntos se distribuyen a lo largo de una línea de creciente, esto sugiere que la varianza de la serie aumenta con la media; si los puntos siguen una línea decreciente esto indica que la varianza disminuye a medida que la media aumenta.

Además del gráfico, gretl presenta las medias y los rangos de cada submuestra, el coeficiente para la pendiente en una regresión MCO de los rangos sobre las medias y el valor p para el contraste de la hipótesis nula de que dicha pendiente es cero. Si el coeficiente de la pendiente es significativo, al nivel de significación del 10 por ciento, se muestra también en el gráfico la recta de regresión estimada de los rangos sobre las medias.

Menú gráfico: /Variable/Gráfico rango-media

run

Argumento: *inputfile*

Ejecuta las instrucciones de *inputfile* y devuelve el control a la línea de instrucciones.

Menú gráfico: Icono de 'ejecutar' en la ventana de guión de instrucciones

runs

Argumento: *nombre-var*

Realiza el contraste "rachas" (no paramétrico) de aleatoriedad de la variable especificada. Si Vd. desea contrastar la aleatoriedad de las desviaciones respecto a la mediana, para una variable denominada $x1$ con mediana distinta de cero, puede hacer lo siguiente:

```
genr signx1 = x1 - median(x1)
runs signx1
```

Menú gráfico: /Variable/Contraste de rachas

scatters

Argumentos: *yvar ; xlistavar o ylistavar ; xvar*

Ejemplos: `scatters 1 ; 2 3 4 5`

`scatters 1 2 3 4 5 6 ; 7`

Dibuja gráficos bivariantes, de la variable *yvar* con respecto a todas las variables de la lista *xlistavar*, o de todas las variables de la lista *ylistavar* con respecto a *xvar*. El primer ejemplo de arriba sitúa la variable 1 en el eje y y realiza cuatro gráficos, el primero con la variable 2 en el eje x , el segundo con la variable 3 en el eje x , y así sucesivamente. En el segundo ejemplo se representa cada una de las variables 1,..., 6 con respecto a la variable 7 en el eje x . Revisar un conjunto de gráficos como estos puede ser útil al realizar análisis exploratorio de datos. El número máximo de gráficos es seis; cualquier otra variable extra en la lista será ignorada.

Menú gráfico: /Datos/Gráficos bivariantes múltiples

set

Argumentos: *variable valor*

Establece los valores de varios parámetros del programa. Los valores dados permanecen vigentes durante todo el desarrollo de la sesión de gretl, a no ser que se cambien con una nueva llamada a la instrucción set. Los parámetros que pueden determinarse de esta forma se enumeran abajo. Hay que señalar que los parámetros de `hc_lag` y `hc_version` se usan cuando se elige la opción `--robust` en la instrucción `ols`.

- `echo`: `off` u `on` (por defecto). Suprime o activa el eco de las instrucciones en la salida de gretl.
- `qr`: `on` u `off` (por defecto). Utiliza QR en lugar de la descomposición de Cholesky al calcular las estimaciones por MCO.
- `hc_lag`: `nw1` (por defecto) o `nw2`, o un entero. Establece el retardo máximo, p , usado al calcular desviaciones típicas HAC (Heteroskedasticity and Autocorrelation Consistent) usando el método de Newey-West para datos de series temporales. `nw1` y `nw2` representan dos variantes para el cálculo automático basadas en el tamaño muestral, T : para `nw1`, $p = 0,75 \times T^{1/3}$, y para `nw2`, $p = 4 \times (T/100)^{2/9}$.
- `hc_version`: 0 (por defecto), 1, 2 o 3. Determina la variante usada al calcular desviaciones típicas consistentes ante heterocedasticidad con datos de sección cruzada. Las opciones corresponden a los valores HC0, HC1, HC2 y HC3 estudiadas por Davidson y MacKinnon en *Econometric Theory and Methods*, capítulo 5. HC0 produce lo que normalmente se llaman “desviaciones típicas de White”.
- `force_hc`: `off` (por defecto) u `on`. Por defecto, con datos de series temporales y cuando se da la opción `--robust` en `ols`, se usa el estimador HAC. Si se establece `force_hc` en “on”, esto fuerza al cálculo de la Matriz de Covarianzas Consistente ante Heterocedasticidad regular (que no tiene en cuenta la autocorrelación).
- `garch_vcv`: `unset`, `hessian`, `im` (matriz de información), `op` (matriz de productos externos), `qml` (estimador QML), `bw` (Bollerslev-Wooldridge). Especifica la variante que se usará para estimar la matriz de covarianzas de los coeficientes para modelos GARCH. Si se da `unset` (por defecto) entonces se usa el Hessiano a no ser que se dé también la opción “robust” para la instrucción `garch`, en cuyo caso se usa QML.
- `hp_lambda`: `auto` (por defecto), o un valor numérico. Determina el parámetro de suavizado para el filtro de Hodrick-Prescott (ver la función `hpfilt` bajo la instrucción `genr`). El valor por defecto es usar 100 veces el cuadrado de la periodicidad, lo cual da 100 para datos anuales, 1600 para trimestrales, y así sucesivamente.

setobs

Argumentos: *periodicidad obs-inicial*

Ejemplos: `setobs 4 1990:1`
`setobs 12 1978:03`
`setobs 20 1:01`

Fuerza al programa a interpretar el conjunto de datos actual como serie temporal o como panel, cuando los datos se han leído como series simples sin fechas. La *periodicidad* debe ser un entero; *obs-inicial* es una cadena de caracteres que representa la fecha o identificación de panel de la primera observación. Ver también [panel](#).

4 1990:1	Interpretar los datos como trimestrales, comenzando en 1990, Q1
12 1978:03	Interpretar los datos como mensuales, comenzando en marzo de 1978
20 1:01	Frecuencia 20, empezando en la observación 1.01 (datos de panel)
5 1972/01/20	Datos diarios (5 días por semana), comenzando el 20 de enero de 1972
7 2002/01/20	Datos diarios (7 días por semana), comenzando el 20 de enero de 2002

Menú gráfico: /Muestra/Establecer frecuencia, observación inicial

setmissArgumentos: *valor* [*listavar*]Ejemplos: `setmiss -1`
`setmiss 100 x2`

Hace que el programa interprete algún valor numérico concreto (el primer parámetro de la instrucción) como código para valor “ausente”, en el caso de datos importados. Si este valor es el único parámetro, como en el primer ejemplo de arriba, la interpretación se aplicará a todas las series del conjunto de datos. Si después de *valor* sigue una lista de variables, por nombre o número, la interpretación se restringe a las variables especificadas. Así, en el segundo ejemplo el valor 100 se interpreta como código para “valor ausente”, pero sólo para la variable *x2*.

Menú gráfico: /Muestra/Establecer código de 'valor perdido'

shellArgumento: *instrucción-de-shell*Ejemplos: `! ls -al`
`! notepad`

Un `!` al comienzo de una línea de instrucciones se interpreta como un escape al shell del usuario. Así se pueden ejecutar instrucciones de shell arbitrarias desde dentro de gretl.

smplVariantes: `smpl startobs endobs``smpl +i -j``smpl dummyvar --dummy``smpl condition --restrict``smpl --no-missing [varlist]``smpl n --random``smpl full`Ejemplos: `smpl 3 10``smpl 1960:2 1982:4``smpl +1 -1``smpl x > 3000 --restrict``smpl y > 3000 --restrict --replace``smpl 100 --random`

Cambia el rango muestral. El nuevo rango puede definirse de varias formas. En la primera forma alternativa (y los dos primeros ejemplos) de arriba *startobs* y *endobs* deben ser coherentes con la periodicidad de los datos. Cualquiera de ellos se puede sustituir por un punto y coma para dejar el valor sin cambios. En la segunda forma los enteros *i* y *j* (que pueden ser positivos o negativos y deberían tener signo) se toman como offsets respecto al rango muestral existente. En la tercera forma, *dummyvar* debe ser una variable indicador con valores 0 o 1 en cada observación; la muestra se restringirá a las observaciones en las que el valor es 1. La cuarta forma, usando `--restrict` restringe la muestra a las observaciones que satisfacen la condición Booleana (que se especifica de acuerdo a la sintaxis de la instrucción [genr](#)).

Con la opción `--no-missing`, si se especifica *listavar*, las observaciones se seleccionan con la condición de que todas las variables de *listavar* tengan valores válidos en esa observación; cuando no se especifica *listavar*, las observaciones se seleccionan con la condición de que *todas* las variables tengan valores validos (no ausentes).

Con la opción `--random`, se extrae del conjunto de datos el número de observaciones especificado, de manera aleatoria. Si se desea poder replicar esta selección más adelante, se debe establecer primero una semilla para el generador de números aleatorios (Véase la instrucción [set](#)).

La forma final, `smp1 full`, restablece el rango muestral completo.

Nótese que las restricciones sobre la muestra son, por defecto, acumulativas: la base para cualquier orden `smp1` es la muestra actual. Si Vd. desea que la instrucción actúe reemplazando cualquier restricción previa, puede añadir la opción `--replace` al final de la instrucción.

Puede usarse la variable interna `obs` junto con la opción `--restrict` de `smp1` para excluir observaciones particulares de la muestra. Por ejemplo,

```
smp1 obs!=4 --restrict
```

quitará sólo la cuarta observación. Si los datos se identifican mediante etiquetas,

```
smp1 obs!="USA" --restrict
```

quitará la observación con etiqueta "USA".

Con respecto a las formas `--dummy`, `--restrict` y `--no-missing` de `smp1`, hay que señalar que cualquier información "estructural" en el fichero de datos (referente a la estructura de series temporales o de panel de los datos) se pierde al ejecutar esta orden. Es posible reimponer la estructura inicial con la instrucción [setobs](#).

Véase the *Gretl User's Guide* para más detalles.

Menú gráfico: /Muestra

spearman

Argumentos: `x y`

Opción: `--verbose` (mostrar los datos ordenados)

Presenta el coeficiente de rango de correlación para las dos variables x e y . No es necesario antes ordenar y hacer el ranking de las variables, la función se encarga de ello.

El ranking automático es de mayor a menor (es decir, el mayor valor de los datos obtiene rango 1). Si Vd. necesita invertir este ranking cree una nueva variable que sea el negativo de la primera original. Por ejemplo:

```
genr altx = -x
spearman altx y
```

Menú gráfico: /Modelo/Correlación por Rangos

square

Argumento: `listavar`

Opción: `--cross` (además de los cuadrados genera los productos cruzados)

Genera variables nuevas que son los cuadrados de las variables de la lista `listavar` (más los productos cruzados, si se da la opción `--cross`). Por ejemplo, `square x y` generará `sq_x = x` cuadrado, `sq_y = y` cuadrado and (optionalmente) `x_y = x` por y . Si una determinada variable es una variable ficticia, no será elevada al cuadrado, ya que se obtendría lo mismo.

Menú gráfico: /Datos/Añadir variables/cuadrados delas variables

store

Argumentos: `fichero-de-datos [ listavar ]`

Opciones: `--csv` (usar formato CSV)

`--gnu-octave` (usar formato GNU Octave)

`--gnu-R` (usar formato GNU R)

`--traditional` (usar el formato tradicional de ESL)

`--gzipped` (aplicar la compresión mediante gzip)

`--dat` (usar el formato ASCII de PcGive)

Guarda el conjunto de datos completo o, si se suministra una lista *listavar*, el subconjunto especificado de variables del conjunto de datos actual, al fichero dado en *fichero-de-datos*.

Por defecto los datos se guardan en formato gretl “nativo”, pero las distintas opciones permiten guardarlos en varios formatos alternativos. Los datos CSV (Comma-Separated Values) pueden ser leídos por programas de hoja de cálculos y también pueden manipularse mediante un editor de textos. Los formatos de Octave, R y PcGive están diseñados para el uso con estos respectivos programas. La compresión mediante Gzip puede ser útil para grandes conjuntos de datos. Ver the *Gretl User's Guide* para detalles sobre los distintos formatos.

Hay que señalar que las variables escalares no serán guardadas automáticamente: si Vd. desea guardar escalares debe escribirlas explícitamente en la lista *listavar*.

Menú gráfico: /Archivo/Guardar datos; /Archivo/Exportar datos

summary

Argumento: [*listavar*]

Presenta los estadísticos principales de las variables de la lista *listavar*, o si se omite *listavar*, de todas las variables del conjunto de datos. Se muestran la media, desviación típica, coeficiente de variación (= $dt/media$), mediana, mínimo, máximo, coeficiente de asimetría y exceso de curtosis.

Menú gráfico: /Datos/Estadísticos principales

Otro acceso: Menú emergente de la ventana principal

system

Argumentos: *type savevars*

Ejemplos: `system type=sur`

`system type=sur save=resids`

`system type=3sls save=resids,fitted`

Da comienzo a un sistema de ecuaciones. Actualmente se soportan dos tipos de sistemas: *sur* (Seemingly Unrelated Regressions, ecuaciones de regresión aparentemente no relacionadas) y *3sls* (Three-Stage Least Squares, mínimos cuadrados en tres etapas). En el campo opcional *save=* se puede especificar si guardar los residuos (*resids*) y/o los valores ajustados (*fitted*). El sistema debe contener al menos dos ecuaciones, que se especifican por medio de la instrucción *equation*, y debe terminarse con la línea *end system*.

En el contexto de un sistema de Mínimos Cuadrados en Tres Etapas, Vd. puede proporcionar una lista de instrumentos (mediante nombre o número). Esta debería ir en una línea por separado dentro del bloque *system*, precedido de la palabra *instr*.

tabprint

Argumento: [*-f fichero*]

Opción: `--complete` (Crea un documento completo)

Debe ejecutarse después de la estimación de un modelo. Presenta el modelo estimado en forma de una tabla de $\text{\LaTeX}$. Si se especifica un nombre de fichero por medio de la opción *-f* el resultado va a ese fichero, en caso contrario va a un fichero con un nombre de la forma *model_N.tex*, donde N es el número de modelos estimados hasta ese momento en la sesión actual de gretl. Ver también [eqnprint](#).

Si se da la opción `--complete` el fichero $\text{\LaTeX}$ será un documento completo, listo para ser procesado; en caso contrario debería ser incluido en un documento ya formado.

Menú gráfico: Ventana de Modelo, / $\text{\LaTeX}$

testuhat

Debe ejecutarse después de una instrucción de estimación. Da la distribución de frecuencias de los residuos del modelo y un contraste chi-cuadrado de normalidad, basado en el procedimiento propuesto por Doornik y Hansen (1984).

Menú gráfico: Ventana de Modelo, /Contrastes/normalidad de los residuos

tobit

Argumentos: *vardep varindeps*

Opciones: --vcv (mostrar la matriz de covarianzas)

--verbose (mostrar los detalles de las iteraciones)

Estima un modelo Tobit. Este modelo puede ser adecuado cuando la variable dependiente es una variable “truncada”. Por ejemplo, se observan valores positivos y cero de las compras de bienes duraderos por parte de las familias, pero no valores negativos. Sin embargo las decisiones sobre tales compras pueden considerarse como resultados de una disposición subyacente a la compra que puede ser negativa en algunos casos. Para detalles ver el libro de Greene *Econometric Analysis*, Capítulo 20.

Menú gráfico: /Modelo/Tobit

tsls

Argumentos: *vardep varindeps ; instrumentos*

Opciones: --vcv (mostrar la matriz de covarianzas)

--robust (desviaciones típicas robustas)

Ejemplo: `tsls y1 0 y2 y3 x1 x2 ; 0 x1 x2 x3 x4 x5 x6`

Calcula las estimaciones por mínimos cuadrados en dos etapas (MC2E o VI): *vardep* es la variable dependiente, *varindeps* es la lista de variables independientes (incluyendo las variables endógenas del lado derecho) en la ecuación estructural para la cual se necesitan las estimaciones MC2E; y *instrumentos* es la lista combinada de variables exógenas y predeterminadas en todas las ecuaciones. Si la lista de *instrumentos* no es al menos tan larga como *varindeps*, el modelo no está identificado.

En el ejemplo de arriba, las *ys* son las variables endógenas y las *xs* son las exógenas y predeterminadas.

Menú gráfico: /Modelo/Mínimos cuadrados en dos etapas

var

Argumentos: *orden listavar ; detlist*

Opción: --quiet (no mostrar respuestas al impulso etc.)

Ejemplo: `var 4 x1 x2 x3 ; const time`

Construye y estima (usando MCO) un modelo autorregresivo vectorial (VAR). El primer argumento especifica el orden de retardos, después va la estructura de la primera ecuación. No hay que incluir retardos entre los elementos de *listavar* — se añadirán automáticamente. El punto y coma separa las variables estocásticas, para las que se incluirán un número *orden* de retardos, de los términos determinísticos de *detlist*, tales como la constante, la tendencia temporal y las variables ficticias estacionales.

De hecho, gretl es capaz de reconocer las variables determinísticas más comunes (constante, tendencia temporal, variables ficticias que no tengan más valores que el 0 y el 1), así que estas no es necesario situarlas después del punto y coma. Variables determinísticas más complejas (p.ej. una tendencia temporal interactuando con una variable ficticia) deben incluirse después del punto y coma.

Se ejecuta una regresión por separado para cada variable de *listavar*. La salida de cada ecuación presenta los contrastes *F* de restricciones cero de todos los retardos de cada variable; un con-

traste F de significatividad del retardo máximo; descomposiciones de la varianza de predicción y las funciones impulso-respuesta.

Las descomposiciones de la varianza y las funciones impulso-respuesta se basan en la descomposición de Cholesky de la matriz de covarianzas contemporánea, y en este contexto el orden en el que se dan las variables (estocásticas) es importante. La primera variable de la lista se supone que es “más exógena” dentro del periodo.

Menú gráfico: /Modelo/Serie temporal/Autorregresión vectorial (VAR)

varlist

Presenta un listado de las variables actualmente disponibles. `list` y `ls` son sinónimos.

vartest

Argumentos: `var1 var2`

Calcula el estadístico F para contrastar la hipótesis nula de que las varianzas poblacionales de las variables `var1` y `var2` son iguales, y muestra su valor p .

Menú gráfico: /Datos/Diferencia de varianzas

vif

Debe ejecutarse inmediatamente después de la estimación de un modelo que incluya al menos dos variables independientes. Calcula y muestra los factores de inflación de varianza (VIFs) de los regresores. El factor de inflación de varianza (VIF) del regresor j se define como

$$\frac{1}{1 - R_j^2}$$

donde R_j es el coeficiente de correlación múltiple entre el regresor j y los demás regresores. El factor tiene un valor mínimo de 1,0 cuando la variable en cuestión es ortogonal a las demás variables independientes. Neter, Wasserman, y Kutner (1990) sugieren inspeccionar el mayor VIF como una medida de diagnóstico para la multicolinealidad; un valor mayor que 10 se suele considerar que indica un grado de multicolinealidad problemático.

Menú gráfico: Model window, /Tests/collinearity

wls

Argumentos: `wtvar vardep varindeps`

Opción: `--vcv` (mostrar la matriz de covarianzas)

Calcula las estimaciones de Mínimos cuadrados ponderados usando `wtvar` como variable de ponderaciones, `vardep` como variable dependiente y `varindeps` como lista de variables independientes. En concreto, se ejecuta una regresión MCO de `wtvar * vardep` contra `wtvar * varindeps`. Si `wtvar` es una variable ficticia, esto es equivalente a eliminar todas las observaciones en las que `wtvar` tiene valor cero.

Menú gráfico: /Modelo/Mínimos cuadrados ponderados

1.3. Instrucciones arreglaron por tema

Estimación

<code>ar</code>	Estimación autorregresiva	<code>arima</code>	Modelo ARMA
<code>equation</code>		<code>garch</code>	Modelo GARCH
<code>hccm</code>	Estimación HCCM	<code>hsk</code>	Estimaciones corregidas de heterocedasticidad
<code>lad</code>	Estimación de mínima desviación absoluta	<code>logistic</code>	Regresión logística

<code>logit</code>	Regresión Logit	<code>mpols</code>	MCO de precisión Múltiple
<code>nls</code>	Mínimos cuadrados no lineales	<code>ols</code>	Mínimos Cuadrados Ordinarios
<code>probit</code>	Modelo Probit	<code>system</code>	
<code>tobit</code>	Modelo Tobit	<code>tsls</code>	Mínimos cuadrados en dos etapas
<code>var</code>	Autorregresión vectorial	<code>wls</code>	Mínimos cuadrados ponderados

Contrastes

<code>add</code>	Añadir variables	<code>addto</code>	
<code>adf</code>	Contraste aumentado de Dickey-Fuller	<code>arch</code>	Contraste ARCH
<code>chow</code>	Contraste de Chow	<code>coeffsum</code>	Suma de coeficientes
<code>coint</code>	Contraste de cointegración de Engle-Granger	<code>coint2</code>	Contraste de cointegración de Johansen
<code>cusum</code>	Contraste CUSUM	<code>hausman</code>	Contraste de Hausman (diagnósticos de panel)
<code>leverage</code>	Observations influyentes	<code>lmtest</code>	Contrastes LM
<code>meantest</code>	Diferencia de medias	<code>omit</code>	Omisión de variables
<code>omitfrom</code>		<code>reset</code>	Ramsey's RESET
<code>restrict</code>	Restricciones lineales	<code>runs</code>	Contraste de rachas
<code>testuhat</code>	Normalidad de los residuos	<code>vartest</code>	Diferencia de varianzas
<code>vif</code>	Factores de inflación de varianza		

Transformaciones

<code>diff</code>	<code>lags</code>
<code>ldiff</code>	<code>logs</code>
<code>square</code>	

Estadísticos

<code>corr</code>	<code>corrgram</code>	Correlograma
<code>freq</code>	<code>pca</code>	Análisis de componentes principales
<code>pergram</code>	<code>spearman</code>	Correlación por rangos de Spearman
<code>summary</code>		

Conjunto de datos

<code>data</code>		<code>delete</code>	
<code>genr</code>	Generar una variable nueva	<code>info</code>	
<code>setinfo</code>	Editar los atributos de una variable	<code>labels</code>	
<code>nulldata</code>	Crear un conjunto de datos vacío	<code>open</code>	
<code>panel</code>	Datos de panel	<code>rename</code>	
<code>setobs</code>	Frecuencia y observación inicial	<code>setmiss</code>	Código de valor ausente
<code>smp1</code>		<code>store</code>	Guardar datos
<code>varlist</code>			

Gráficos

<code>boxplot</code>	Gráficos de caja	<code>gnuplot</code>	
<code>graph</code>		<code>plot</code>	
<code>rmp1ot</code>	Gráfico Rango-Media	<code>scatters</code>	Gráficos bivariantes múltiples

Impresión

eqnprint
print
tabprint

outfile
printf

Predicción

fcast

Programación

else
endif
function
loop
set

end
endloop
if
run

Utilidades

criteria
modeltab La tabla de modelos
quit

help
pvalue
shell

Capítulo 2

Opciones, argumentos y directorios

2.1. gretl

`gretl` (en MS Windows, `gretlw32.exe`)¹

— Abre el programa y espera las instrucciones del usuario.

`gretl archivo-de-datos`

— Abre el programa con el archivo de datos indicado en su ventana de trabajo. El archivo de datos puede estar en formato propio de `gretl`, formato CSV, o formato BOX1 (véase el *Gretl User's Guide*). El programa intenta detectar el formato del archivo y procede en consecuencia. Véase también Sección 2.3 abajo para el comportamiento de búsqueda de directorios.

`gretl --help` (o `gretl -h`)

— Muestra un breve resumen del historial de la sesión y sale al sistema operativo.

`gretl --version` (o `gretl -v`)

— Muestra la identificación de la versión del programa y sale al sistema operativo.

`gretl --run archivo-de-instrucciones` (o `gretl -r archivo-de-instrucciones`)

— Abre el programa con una ventana mostrando el archivo de instrucciones indicado, listo para su ejecución. Véase Sección 2.3 abajo para las características de búsqueda de directorios.

`gretl --db base-de-datos` (o `gretl -d base-de-datos`)

— Abre el programa con una ventana en la que se muestra la base de datos especificada. Si los archivos de la base de datos (el archivo `.bin` y su acompañante `.idx` — véase el *Gretl User's Guide*) no están en el directorio de bases de datos por defecto del sistema, es preciso especificar la ubicación completa.

Podemos configurar varios detalles de `gretl` con el menú “Archivo, Preferencias”.

- El directorio base para los archivos compartidos de `gretl`.
- El directorio base del usuario para archivos relacionados con `gretl`.
- La instrucción para lanzar `gnuplot`.
- La instrucción para lanzar GNU R.
- La instrucción para visualizar archivos TeX DVI.
- El directorio en donde empezar a buscar las bases de datos nativas de `gretl`.
- El directorio en donde empezar a buscar las bases de datos de RATS 4.
- El número IP del servidor de bases de datos de `gretl`.
- El número IP y número de puerto del servidor proxy HTTP a utilizar para contactar con el servidor de bases de datos, en caso de que sea aplicable (si tenemos un cortafuegos).
- Los programas de calculadora y edición a lanzar desde la barra de herramientas.
- La fuente monoespacio a utilizar en los resultados de `gretl` en la pantalla.

¹En Linux, se instala un “lote de instrucciones de envoltura (wrapper)” llamado `gretl`. Este lote de instrucciones comprueba si la variable de entorno `DISPLAY` está activada; si lo está, lanza el programa GUI, `gretl_x11`, y si no, lanza el programa de línea de instrucciones, `gretlcli`.

- La fuente a utilizar en los menús y otros mensajes. (Nota: esta opción no está presente cuando gretl está compilado para el escritorio gnome, ya que en este caso la elección de fuentes se controla directamente desde gnome.)

También hay algunas casillas para señalar. Señalando la casilla “modo experto” se suprimen algunos mensajes de aviso, que se muestran en caso contrario. Marcar la casilla “Informar sobre las actualizaciones de gretl” hace que gretl intente conectar con el servidor de actualizaciones al empezar el programa. Desmarcar “Mostrar barra de herramientas de gretl” desactiva el icono del mismo. Si el idioma del sistema no es el Inglés, y el carácter de punto decimal no es el punto “.”, desactivar la casilla “Utilizar configuración local para el punto decimal” hará que gretl utilice el punto a pesar de todo.

Finalmente, hay algunas elecciones alternativas: en la pestaña “Abrir/Guardar camino” puede establecerse el directorio por defecto donde gretl tiene que buscar al abrir o guardar un archivo - bien el directorio de usuario de gretl o bien el directorio de trabajo actual. En la pestaña “Ficheros de datos” puede establecerse el sufijo a utilizar por defecto para los nombres de archivos de datos. El sufijo normal es .gdt pero, si se desea, se puede poner .dat, que era la norma en las versiones anteriores del programa. Si se pone .dat por defecto, entonces los archivos de datos serán guardados en el formato “tradicional” (véase el *Gretl User's Guide*). También es posible seleccionar la acción a asociar con el icono de carpeta en la barra de herramientas en esta misma pestaña de “Ficheros de datos”: se puede elegir entre abrir el listado de los archivos de datos asociados con el libro de texto de Ramanathan, o con el libro de Wooldridge.

Las opciones establecidas de esta manera se gestionan de modo diferente dependiendo del contexto. En MS Windows se guardan en el registro de Windows. En el escritorio de gnome se guardan en .gnome/gretl en el directorio personal del usuario. En otros casos, se guardan en un archivo llamado .gretlrc en el directorio del usuario.

2.2. gretlcli

gretlcli

— Abre el programa y espera a que el usuario introduzca órdenes.

gretlcli *archivo-de-datos*

— Abre el programa con un archivo de datos específico en su espacio de trabajo. El archivo de datos puede estar en formato nativo de gretl, formato CSV, o formato BOX1 (véase el *Gretl User's Guide*). El programa intentará detectar el formato del archivo y procederá en consecuencia. Véase también Sección 2.3 abajo, para más detalles sobre las características de búsqueda de directorios.

gretlcli --help (o gretlcli -h)

— Muestra un breve resumen de uso (lista de instrucciones).

gretlcli --version (o gretlcli -v)

— Muestra la identificación de la versión del programa.

gretlcli --pvalue (o gretlcli -p)

— Abre el programa de modo que podemos especificar de manera interactiva el valor p para varios estadísticos habituales.

gretlcli --run *archivo-de-guión* (o gretlcli -r *archivo-de-guión*)

— Ejecuta las instrucciones que hay en *archivo-de-guión* y después pasa el control a la línea de instrucciones. Véase Sección 2.3 para más detalles sobre las características de búsqueda de directorios.

gretlcli --batch *archivo-de-guión* (o gretlcli -b *archivo-de-guión*)

— Ejecuta las instrucciones que hay en *archivo-de-guión* y después sale del programa. Cuando utilice esta opción, Vd. probablemente deseará redirigir la salida del programa a un archivo. Véase Sección 2.3 para más detalles sobre las características de búsqueda de directorios.

Al usar las opciones `--run` y `--batch`, el guión de instrucciones en cuestión tiene que abrir un archivo de datos. Es posible hacer esto mediante la instrucción `open` dentro del lote.

2.3. Directorios

Cuando se indica el nombre de un archivo de datos o de un archivo de lote de instrucciones a `gretl` o a `gretlcli` en la línea de instrucciones, se busca el archivo de la siguiente manera:

1. “Como tal”. Es decir, en el directorio de trabajo actual o, si se especifica la dirección completa, en el sitio especificado.
2. En el directorio del usuario de `gretl` (véase la Tabla 2.1 para valores dados por defecto).
3. En cualquier subdirectorio por debajo del directorio de usuario de `gretl`.
4. En el caso de archivos de datos, la búsqueda continúa con el directorio principal de `gretl`. En el caso de un lote de instrucciones, la búsqueda continúa en el directorio de lotes de instrucciones del sistema. Véase la Tabla 2.1 para los valores dados por defecto. (PREFIX denota el directorio base elegido en el momento en que fue instalado `gretl`.)
5. En el caso de archivos de datos, la búsqueda procede entonces hacia todos los subdirectorios por debajo del directorio principal de datos.

Cuadro 2.1: Directorios por defecto

	<i>Linux</i>	<i>MS Windows</i>
Directorio del usuario	<code>\$HOME/gretl</code>	<code>PREFIX\gretl\user</code>
Directorio de datos del sistema	<code>PREFIX/share/gretl/data</code>	<code>PREFIX\gretl\data</code>
Directorio de lotes de instrucciones del sistema	<code>PREFIX/share/gretl/scripts</code>	<code>PREFIX\gretl\scripts</code>

Por lo tanto no es necesario especificar el directorio completo de los archivos de datos o de instrucciones a menos que se quiera prescindir del mecanismo de búsqueda automática. (Esto también es válido para `gretlcli` cuando se indica un nombre de archivo como argumento en las instrucciones `open` o `run`.)

Cuando un lote de instrucciones contiene una instrucción para abrir un archivo de datos, el orden de búsqueda del archivo de datos es como se ha descrito arriba, con la excepción de que también se busca en el directorio donde está el lote, inmediatamente después de intentar encontrar el archivo de datos “como tal”.

MS Windows

En MS Windows, la información sobre la configuración de `gretl` y `gretlcli` se guarda en el registro de Windows. Cuando `gretl` se instala por primera vez, se crea un conjunto de entradas de registro que pueden cambiarse mediante el menú “Archivo, Preferencias”. En el caso de que haya que llevar a cabo algunos ajustes de forma manual a esta información, las entradas se encuentran (utilizando el programa estándar de Windows, `regedit.exe`) dentro de `Software\gretl` en `HKEY_LOCAL_MACHINE` (el directorio principal de `gretl` y la instrucción para invocar `gnuplot`) y `HKEY_CURRENT_USER` (las demás variables configurables).

Capítulo 3

Reserved words

Reserved words, which cannot be used as the names of variables, fall into the following categories:

- Names of constants: `CONST`, `NA`, `const`, `null`, `pi`.
- Names of internal variables and data types: `matrix`, `obs`, `scalar`, `series`, `t`.
- Names of functions, as shown in Table 3.1.

Cuadro 3.1: Function names

BFGSmax	I	abs	acos	argname	asin	atan	bkfilt
boxcox	cdemean	cdf	cdiv	ceil	cholesky	cmult	cnorm
colnames	cols	corr	cos	cov	critical	cum	det
diag	diff	dnorm	dsort	dummify	eigengen	eigensym	exp
fdjac	fft	ffti	filter	firstobs	floor	fracdiff	gammafun
genpois	getenv	gini	ginv	grab	hpfilt	imaxc	imaxr
iminc	iminr	infnorm	int	inv	invcdf	invpd	islist
isnull	isscalar	isseries	isstring	lags	lastobs	ldet	ldiff
lincomb	ljungbox	ln	lngamma	log	log10	log2	lower
lrvar	makemask	max	maxc	maxr	mcorr	mcov	mean
meanc	meanr	median	mexp	min	minc	minr	missing
misszero	mlag	mnormal	mols	movavg	mread	mshape	msortby
muniform	mwrite	mxtab	nelem	nobs	normal	nullspace	obslabel
obsnum	ok	onenorm	ones	orthdev	pdf	pmax	pmean
pmin	pnobs	polroots	princomp	psd	pvalue	qform	qnorm
qrdecomp	quantile	randgen	rank	ranking	rcond	readfile	resample
round	rows	sd	sdc	sdiff	selifc	selifr	seq
sin	sort	sortby	sqr	sst	strlen	strstr	sum
sumc	sumr	svd	tan	tr	transp	trimr	uniform
unvech	upper	urcpval	values	var	varname	varnum	vec
vech	wmean	wsd	wvar	xpx	zeromiss	zeros	