

Manual de Gretl



Gnu Regression, Econometrics and Time-series Library

Allin Cottrell
Departamento de Economía
Wake Forest University
Noviembre, 2002 (gretl 1.0.1)
(Traducción julio, 2003)

Manual de Gretl: Gnu Regression, Econometrics and Time-series Library
por Allin Cottrell

Copyright © 2001-2002 Allin Cottrell

Copyright © de la traducción española: 2003 Parmeeta Bhogal - 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.

Tabla de contenidos

1. Introducción	1
Características principales	1
Agradecimientos	1
Instalación de los programas	2
Linux	2
MS Windows	2
Actualizaciones	3
2. Puesta en marcha	4
Ejecutar una regresión	4
Resultados de la estimación	6
Menús de la ventana principal	7
La barra de herramientas de gretl	11
3. Modos de trabajo	12
Lotes de instrucciones	12
El concepto de “sesión”	13
4. Fichero de datos	16
Formato propio	16
Otros formatos de archivos de datos	16
Bases de datos binarias	16
Acceso en línea a bases de datos	16
Bases de datos RATS 4	16
Crear un archivo de datos desde cero	17
Utilización de una hoja de cálculo diferente	17
Utilizar la hoja de cálculo incorporada	17
Seleccionar desde un base de datos	18
Crear un archivo de datos independiente	18
Nota adicional	18
Datos ausentes	19
5. Datos de Panel	20
Estructura de Panel	20
Variables ficticias	20
Uso de valores retardados con datos de panel	21
Estimación combinada	22
Ilustración: La Tabla Mundial de Penn	23
6. Gráficos y Diagramas	24
Gráficos Gnuplot	24
Gráficos de caja	25
7. Construcción de iteraciones	26
Simulaciones Monte Carlo	26
Mínimos cuadrados iterativos	27
Bucle con índice	28
8. Opciones, argumentos y directorios	29
gretl	29
gretlcli	30
Directorios	31
MS Windows	31
9. Guía de Instrucciones	33
Introducción	33
Instrucciones de gretl	33
add	33
addto	33
adf	34
ar	34
arch	34

chow	35
coint	35
corc	35
corr	35
corrgm	36
criteria	36
cusum	36
delete	36
diff	36
endloop	37
eqnprint	37
fcast	37
fcasterr	37
fit	37
freq	37
genr	38
gnuplot	40
graph	41
hausman	41
hccm	41
help	41
hilu	42
hsk	42
if	42
import	42
info	43
labels	43
lad	43
lags	43
ldiff	43
lmtest	43
logit	43
logs	44
loop	44
meantest	44
mpols	45
multiply	45
noecho	45
nulldata	45
ols	45
omit	46
omitfrom	46
open	46
pergm	46
plot	47
pooled	47
print	47
probit	47
pvalue	48
quit	48
reset	48
rhodiff	48
rmplot	48
run	49
runs	49
scatters	49
seed	50
setobs	50
setmiss	50

shell.....	50
sim.....	50
simpl.....	51
spearman.....	51
square.....	52
store.....	52
summary.....	52
tabprint.....	52
testuhat.....	53
tsls.....	53
var.....	53
varlist.....	53
vartest.....	53
wls.....	53
Estimadores y contrastes: resumen.....	54
10. Solución de problemas en gretl	56
Informes de errores	56
Programas auxiliares	56
11. El interfaz de línea de instrucciones	57
Gretl en la consola	57
Diferencias con ESL.....	57
A. Comentarios sobre los archivos de datos.....	59
Formato nativo básico.....	59
Formato tradicional ESL.....	59
Bases de datos binarias: detalles.....	60
B. Comentarios técnicos.....	62
C. Precisión numérica.....	63
D. Análisis econométrico avanzado con software libre.....	64
E. Lista de URLs	65
Bibliografía	67

Lista de tablas

8-1. Directorios por defecto.....	31
9-1. Ejemplos de uso de la instrucción <code>genr</code>	39
9-2. Estimadores.....	54
9-3. Contrastes para los modelos.....	54

Tabla de figuras

2-1. Ventana de archivo de datos para ejercicios prácticos.....	4
2-2. Ventana principal, con un archivo de datos para ejercicios abierto.....	5
2-3. Cuadro de diálogo de especificación del modelo.....	5
2-4. Ventana de resultados del modelo.....	6
3-1. Ventana de lotes de instrucciones, editando un archivo de instrucciones	12
3-2. Vista de iconos: a los iconos por defecto, se ha añadido un modelo y un gráfico	13
6-1. Controlador <code>gnuplot</code> de <code>gretl</code>	24

Tabla de ejemplos

5-1. Retardos con datos de panel	22
5-2. Uso de la tabla mundial de Penn	23
7-1. Una simulación sencilla de Monte Carlo	26
7-2. Función de consumo no lineal	27
7-3. Ejemplo de un bucle indexado	28

Capítulo 1. Introducción

Características principales

Gretl es un paquete de cálculo econométrico, e incluye una librería compartida, un programa cliente de línea de instrucciones y un interfaz gráfico para el usuario (GUI).

Fácil de usar

Gretl ofrece un interfaz intuitivo para el usuario; es muy fácil de instalar y de usar para realizar análisis econométrico. El paquete está muy ligado al libro *Introductory Econometrics* de Ramanathan, lo cual permite ofrecer muchos archivos de datos para ejercicios prácticos así como ficheros con lotes de instrucciones (scripts). Estos son fácilmente accesibles y vienen acompañados de muchos comentarios.

Flexibilidad

Gretl ofrece la posibilidad de elegir entre un enorme rango de opciones; desde apuntar y hacer clic en modo interactivo hasta la utilización de procesos por lotes, resultando fácil además hacer diferentes combinaciones entre las mismas.

Multiplataforma

La plataforma nativa de gretl es Linux, aunque hay una versión disponible para MS Windows. Fue compilada en AIX y puede funcionar en cualquier sistema tipo unix que contenga los archivos de biblioteca básicos adecuados (véase el [Apéndice B](#)).

Código abierto

El código fuente de gretl se encuentra a la total disposición de cualquiera que esté interesado en revisar, completar o extender el programa. El autor agradece cualquier información sobre errores o fallos del programa.

Sofisticación

Gretl ofrece una completa gama de estimadores de mínimos cuadrados, incluyendo *mínimos cuadrados en dos etapas*. También ofrece estimación logit y probit (binomial), y cuenta con una estructura de bucle para realizar simulaciones de Monte Carlo o estimaciones de mínimo cuadrados iterativas para modelos no lineales. Aunque no incluye todos los estimadores y contrastes que un econométra profesional puede necesitar, el programa permite exportar datos a los formatos (GNU R) y (GNU Octave) para hacer otros tipos de análisis (véase el [Apéndice D](#)).

Precisión

Gretl funciona bien con los conjuntos de datos de referencia de NIST. Véase el [Apéndice C](#).

Uso de Internet

Gretl tiene acceso a un servidor en Wake Forest University desde el cual puede descargar algunas bases de datos. La versión para MS Windows incorpora un programa de actualización que detecta la disponibilidad de nuevas versiones de gretl y ofrece la opción de actualización de manera automática.

Internacional

Gretl puede producir sus resultados en inglés, español o francés, dependiendo del idioma nativo del ordenador.

Agradecimientos

En primer lugar estoy en deuda con el Profesor Ramu Ramanathan de la Universidad de California, San Diego, quien hace algunos años tuvo la bondad de poner a mi disposición el código fuente de su programa ESL (“Econometrics Software Library”), el cual traduje a

Linux, y con quien he colaborado para extender y actualizar el programa. Para el proyecto gretl, he realizado muchos cambios en el código original de ESL. He añadido nuevas funcionalidades econométricas y el interfaz gráfico es completamente nuevo. Por favor, téngase en cuenta que el Profesor Ramanathan no es responsable de ningún error que gretl pueda tener en su código.

Estoy agradecido a William Greene, autor del libro *Econometric Analysis*, por su permiso para incluir en la distribución de gretl algunos de los datos analizados en su libro, y a Jeffrey Wooldridge por su ayuda en la preparación de una versión gretl de los conjuntos de datos incluidos en su libro *Introductory Econometrics: A Modern Approach*.

Con respecto a la internacionalización de gretl, quiero dar la gracias a Ignacio Díaz-Empananza y Michel Robitaille, encargados de preparar las traducciones al Español y Francés respectivamente.

Me he beneficiado mucho del trabajo realizado por numerosos programadores de software de código abierto: para más detalles véase el [Apéndice B](#) de este manual. Mis agradecimientos van también para Richard Stallman de la Free Software Foundation, por su apoyo al software libre en general y, en particular, por su decisión de “adoptar” gretl como un programa GNU.

Instalación de los programas

Linux

En la plataforma Linux¹ tenemos la opción de compilar el código gretl o usar un paquete ya construido a propósito. Hay paquetes precompilados disponibles en el formato rpm (aconsejable para Red Hat Linux y sistemas relacionados con este) y también en el formato deb (Debian GNU/Linux). Estoy agradecido a Dirk Eddelbüttel por la construcción de este último. En el caso de que el usuario prefiriera construir uno por sí mismo, (o si el sistema unix no dispone de paquetes ya construidos), deberá seguir los siguientes pasos.

1. Descargar la última distribución de gretl desde gretl.sourceforge.net.
2. Descomprimir (unzip+untar) el paquete. En un sistema que disponga de utilidades GNU, la instrucción sería `tar -xvfz gretl-N.tar.gz` (reemplazar N con el número de versión específico del archivo que hemos bajado en el paso 1).
3. Ir al directorio de gretl creado en el paso 2 (e.g. `gretl-1.0`).
4. Entonces, la rutina básica de compilación es

```
./configure
make
make check
make install
```

Sin embargo, antes de empezar, es conveniente leer el archivo `INSTALL`, y/o hacer `./configure --help` para ver las opciones disponibles. Una de las opciones que el usuario puede querer ajustar es `--prefix`. La instalación se hace en `/usr/local` por defecto, pero es posible cambiarlo. Por ejemplo, `./configure --prefix=/usr` pondrá todo bajo el directorio `/usr`. En el caso de que algún archivo de biblioteca no se encuentre en el sistema, y por lo tanto falle todo el proceso de instalación, consúltese el [Apéndice B](#) de este manual.

Desde la versión 0.97 gretl es compatible con el escritorio de gnome. Para utilizarlo, se compila el programa en la forma descrita anteriormente. Si se desean suprimir los rasgos específicos de gnome, es preciso pasar la opción `--without-gnome` directamente a `configure`.

1. La terminología nos origina aquí un pequeño problema, aunque en este manual usaremos “Linux” para referirnos al sistema operativo GNU/Linux. Lo que aquí se dice acerca de Linux es, en su mayor parte, válido también para otros sistemas similares a Unix, aunque puede resultar necesaria alguna que otra modificación local.

MS Windows

La versión para MS Windows se extrae y se ejecuta de forma automática. La instalación consiste simplemente en bajar `gretl_install.exe` y ejecutar este programa. Es posible especificar el lugar donde se instalará el paquete (por defecto en `c:\userdata\gretl`).

Actualizaciones

En el caso de que el ordenador este conectado a Internet, `gretl` cuenta con la posibilidad de conectarse a su sitio web, en la Universidad de Wake Forest, al iniciar el programa para ver si hay alguna nueva versión disponible. Si la hay, se abre una ventana informando al usuario de este hecho. Si queremos activar esta opción, hay que marcar la opción “Informar sobre actualizaciones de `gretl`” en el menú “Archivo, Preferencias, General ” de `gretl`.

La versión para MS Windows adelanta un paso más: da la opción de actualización automática si se desea. Para seleccionar esta opción, síganse las instrucciones en la ventana desplegable: cerrar `gretl` y a continuación ejecutar el programa llamado “`gretl updater`” (que se encuentra, junto con el programa principal de `gretl`, bajo Programas en el menú de inicio de Windows). Una vez terminado el proceso de actualización, se puede reiniciar `gretl`.

Capítulo 2. Puesta en marcha

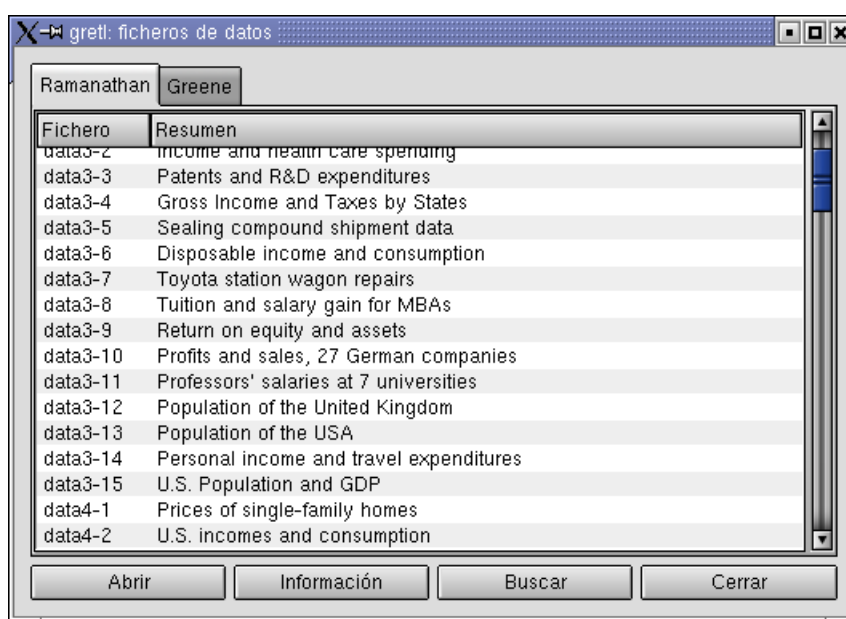
Ejecutar una regresión

Esta parte introductoria está enfocada principalmente hacia el programa de cliente gráfico; para más detalles del programa de línea de instrucciones, `gretlcli`, consultar el [Capítulo 9](#) y el [Capítulo 11](#).

Aunque es posible abrir `gretl` tecleando el nombre de un archivo de datos como argumento, por el momento no vamos a probar este modo; simplemente ejecutamos el programa.¹ Aparecerá una ventana principal (que puede tener alguna información sobre los datos pero que al principio está vacía) y varios menús, algunos de los cuales están desactivados al principio.

¿Qué podemos hacer a partir de este punto? Se puede consultar los archivos (o bases) de datos que vienen con el programa, abrir un archivo de datos, crear un nuevo archivo de datos, leer los apartados de la ayuda, o abrir algún lote de instrucciones. Por ahora vamos a revisar los archivos de datos que acompañan al programa. Desde el menú Archivo elegir “Abrir datos, archivo de muestra, Ramanathan...”. Se debería abrir una segunda ventana con una lista de archivos de datos que vienen con el paquete (véase la [Figura 2-1](#)). Los archivos están numerados siguiendo los capítulos del libro de Ramanathan (2002), el cual también contiene los puntos relacionados con el análisis de estos datos. Aún en el caso de que no dispongamos de este texto, los datos en sí son muy útiles para realizar ejercicios prácticos.

Figura 2-1. Ventana de archivo de datos para ejercicios prácticos



Si seleccionamos una fila en esta ventana y pulsamos “Información”, se abre el archivo de cabecera (header file) de los datos en cuestión, que contiene información sobre el origen y las definiciones de las variables. Para abrir algún archivo que nos interese, pulsar “Abrir”, o simplemente hacer doble clic sobre el nombre del archivo. Por el momento, abriremos el archivo `data3-6`.

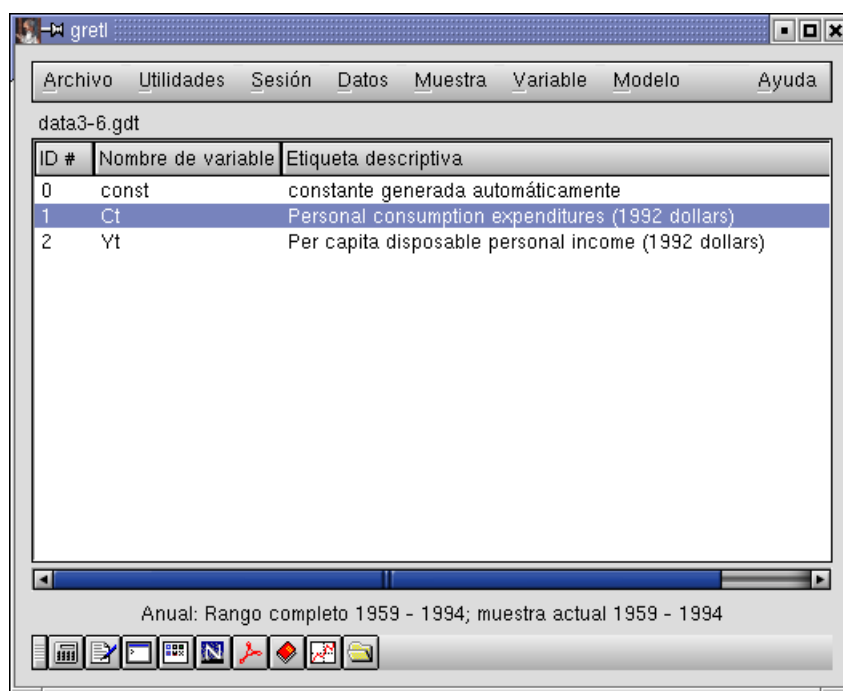
1. Por comodidad este manual se refiere al programa de cliente gráfico simplemente como `gretl`. Nótese, sin embargo, que el nombre específico del programa cambia según la plataforma del ordenador. En Linux se llama `gretl_x11` mientras que en MS Windows se llama `gretl_w32.exe`. En sistemas Linux también se instala un lote de instrucciones de envoltura (wrapper script) llamado `gretl` - véase también el [Capítulo 8](#).



En ventanas gret1 con listas, un doble clic sobre una línea lanza una acción asociada por defecto a esta entrada de la lista: por ejemplo, mostrar valores de una serie de datos, abrir un archivo.

Este archivo contiene datos pertenecientes a un “problema” clásico de econometría: la función de consumo. Ahora, la ventana de datos debería mostrar el nombre del archivo de datos actual, el rango total de los datos y el rango de la muestra, así como los nombres de las variables junto con sus notas descriptivas - véase la [Figura 2-2](#).

Figura 2-2. Ventana principal, con un archivo de datos para ejercicios abierto



Bien, ¿cuál es el siguiente paso? gret1 intenta que las diferentes opciones del menú sean bastante explícitas. Primero, hagamos una breve visita al menú Modelo; en [la sección de nombre Menús de la ventana principal](#) se realiza un breve recorrido a lo largo de los menús de la ventana principal.

El menú Modelo de gret1 ofrece numerosos métodos de estimación econométrica. El más sencillo y estándar es el de Mínimos Cuadrados Ordinarios (MCO). Si se selecciona MCO, aparece un cuadro de diálogo en el que hay que *especificar el modelo* - véase la [Figura 2-3](#).

Figura 2-3. Cuadro de diálogo de especificación del modelo

Para seleccionar la variable dependiente, resaltamos la variable deseada en la lista que aparece a la izquierda y pulsamos el botón “Elegir” que apunta hacia la caja de la variable dependiente. Si marcamos la casilla “Selección por defecto”, esta variable será preseleccionada como dependiente la siguiente vez que abramos el cuadro de diálogo de especificación de modelo. Un atajo: un doble clic sobre una variable dentro de la lista de la izquierda, hace que ésta sea la variable dependiente por defecto. Para seleccionar las variables independientes, primero las resaltamos en la lista de la izquierda, y a continuación pulsamos el botón “Añadir” (o el botón derecho del ratón con el cursor sobre la variable seleccionada). Para seleccionar varias variables en la lista, arrastramos el ratón sobre ellas; para seleccionar diferentes variables que no son contiguas, apretamos la tecla **Ctrl** y manteniéndola así, hacemos clic sobre las variables deseadas.

Para ejecutar una regresión con el consumo como variable dependiente, y la renta como la independiente, hacemos clic sobre *Ct* para ponerla en el espacio de “Variable dependiente”, y luego añadimos *Yt* a la lista de variables Independientes.

Resultados de la estimación

Una vez que hayamos especificado un modelo, aparecerá una ventana mostrando los resultados de la regresión. La información que presenta es bastante comprensible y está escrita en un formato estándar. (Figura 2-4).

Figura 2-4. Ventana de resultados del modelo

Modelo 1: estimaciones MCO
utilizando las 36 observaciones 1959-1994
Variable dependiente: Ct

	VARIABLE	COEFICIENTE	DESV.TÍP.	ESTAD.T	2Prob(t > t)
0)	const	-384,105	151,330	-2,538	0,015892 **
2)	Yt	0,932738	0,0106966	87,199	< 0,00001 ***

Media de la var. dependiente = 12490,9
D.T. de la var. dependiente = 2940,03
Suma de cuadrados de los residuos = 1,34675e+06
Desviación típica de los residuos = 199,023
R-cuadrado no corregido = 0,995548
R-cuadrado corregido = 0,995417
Estadístico F (1, 34) = 7603,7 (valor p < 0,00001)
Estadístico de Durbin-Watson = 0,513696
Coef. de autocorr. de primer orden = 0,768301

ESTADÍSTICOS DE SELECCIÓN DE MODELOS

SGMASQ	39610,3	AIC	41806,1	FPE	41810,9
HQ	43109,7	SCHWARZ	45650,5	SHIBATA	41566,4
GCV	41940,3	RICE	42085,9		

Cerrar

La ventana de resultados contiene menús que nos permiten inspeccionar o hacer gráficos de los valores ajustados y de los residuos, y además podemos ejecutar varios programas de diagnósticos sobre el modelo.

Para la mayoría de los modelos, también existe la opción de reimprimir los resultados de la regresión en formato LaTeX. Podemos imprimir los resultados en formato tabular (similar a lo que hay en la ventana output, pero bien escritos) o como una ecuación, en una página. Para cada una de estas opciones podemos elegir entre una vista preliminar de la composición final, o guardar los resultados en un archivo para su posterior incorporación en un documento LaTeX. La vista preliminar requiere que tengamos un sistema TeX funcionando en nuestro ordenador.

Para exportar los resultados de gretl a un procesador de textos, se puede copiar y pegar desde una ventana de resultados, utilizando su menú Editar, al programa deseado. Muchas (pero no todas) las ventanas de gretl ofrecen la opción de copiar como RTF ("Rich Text Format" de Microsoft) o como documento LaTeX. Si se quiere pegar el contenido en un procesador de texto, la opción RTF quizá sea la mejor porque conserva el formato tabular de los resultados.² Alternativamente, podemos guardar los resultados en un archivo de texto y luego importar este archivo desde el programa que estemos utilizando. Al terminar una sesión de gretl, tenemos la opción de guardar todos los resultados de la sesión en un único archivo.

Téngase en cuenta que tanto en el escritorio gnome como bajo MS Windows, el menú Archivo incluye una instrucción para enviar los resultados directamente a una impresora.



Al pegar o exportar los resultados de gretl como texto a un procesador de textos, conviene seleccionar una fuente monoespaciada o estilo máquina de escribir (e.g. Courier) para conservar el formato tabular del output. Selecciónese un tamaño pequeño de fuente (un Courier de 10 puntos será suficiente) para evitar que las líneas de resultados se partan incorrectamente.

2. Téngase en cuenta que al copiar como RTF en MS Windows, Windows sólo permitirá que se pegue en aquellas aplicaciones que "entiendan" el formato RTF. Por lo tanto, se puede pegar en MS Word, pero no en notepad.

Menús de la ventana principal

Si se lee la barra del menú de la ventana principal, empezando por la izquierda hacia la derecha, se encuentran los siguientes menús: Archivo, Utilidades, Sesión, Datos, Muestra, Variable, Modelo y Ayuda.



§ Menú Archivo

- Abrir datos: Abre un archivo de datos propio de gretl o importa desde otro formato. Véase el [Capítulo 4](#).
- Guardar datos: Guarda el archivo de datos propio de gretl abierto en este momento.
- Guardar datos como: Escribe el conjunto de datos en formato propio, con las opciones de utilizar gzip para comprimir los datos o guardar los datos en formato binario. Véase el [Capítulo 4](#).
- Exportar datos: Escribe el conjunto de datos en formato CSV (comma-separated values — valores separados por comas), o en los formatos de GNU R o GNU Octave. Véase el [Capítulo 4](#) así como el [Apéndice D](#).
- Cerrar conjunto de datos: Cierra el conjunto de datos con que se está trabajando en la memoria. Generalmente no tenemos que hacer esto (ya que al abrir un archivo de datos nuevo, se cierra de manera automática el viejo) pero es de utilidad en algunas ocasiones (véase [la sección de nombre Crear un archivo de datos desde cero en Capítulo 4](#)).
- Revisar bases de datos: Véanse [la sección de nombre Bases de datos binarias en Capítulo 4](#) y [la sección de nombre Crear un archivo de datos desde cero en Capítulo 4](#).
- Crear conjunto de datos: Inicia la hoja de cálculo incorporada en el programa, para añadir datos de forma manual. Véase [la sección de nombre Crear un archivo de datos desde cero en Capítulo 4](#).
- Guardar el último gráfico: Guarda el gráfico más reciente.
- Ver historial de instrucciones: Abre una ventana donde puede verse el historial de todas las instrucciones ejecutadas hasta el momento.
- Abrir archivo de instrucciones: Abre un lote de instrucciones de gretl, bien uno ya creado por el usuario, o bien uno de los archivos de ejercicios prácticos que acompañan al programa. Si se desea crear un lote de instrucciones desde cero, úsese la siguiente opción: Nuevo archivo de instrucciones.
- Preferencias: Especifica la ubicación de varios archivos a los que gretl necesita acceder. Selecciona la fuente en que gretl mostrará los textos generados. Activa o desactiva el “modo experto”. (Si se activa este modo, se suprimen varios avisos.) Activa o desactiva los mensajes de gretl sobre la disponibilidad de las nuevas versiones del programa. Configura o quita/pone la barra de herramientas en la ventana principal.
- Salir: Salir del programa. Si no se ha activado el modo experto, se mostrará un aviso para que se guarde cualquier trabajo no guardado.

§ Menú de Utilidades

- Tablas estadísticas: Consultar los valores críticos de las distribuciones más frecuentes (normal o gaussiana, t , chi-cuadrado, F y Durbin-Watson).
- Buscador de valores p: Abre una ventana que nos permite consultar valores p de distribuciones gaussianas, t , chi-cuadrado, F o gamma. Véase también la instrucción `pvalue` en el [Capítulo 9](#).

- Calculadora de estadísticos de contraste: Calcula los estadísticos de contraste y valores p para una amplia gama de contrastes de hipótesis comunes (media poblacional, varianzas y proporciones; diferencias de medias, de varianzas y de proporciones). Los estadísticos muestrales relevantes han de estar disponibles con anterioridad para su incorporación en el cuadro de diálogo. Para algunos contrastes sencillos, que toman como entrada series de datos en vez de estadísticos muestrales previamente calculados, véase “Diferencia de medias” y “Diferencia de varianzas” en el menú de Datos.
- Consola Gretl: Abre una ventana tipo “consola” dentro de la cual podemos escribir instrucciones (en lugar de apuntar y hacer clic), del mismo modo que en el programa de línea de instrucciones, `gretlcli`. Véase el [Capítulo 9](#).
- Iniciar GNU R: Inicia una sesión de R (si está instalado en el sistema), y carga una copia del conjunto de datos que esté abierto en `gretl`. Véase el [Apéndice D](#).

§ Menú de Sesión

- Vista de iconos: Abre una ventana que muestra la sesión actual de `gretl` en forma de un conjunto de iconos. Para más detalles véase [la sección de nombre El concepto de “sesión” en Capítulo 3](#).
- Añadir el último gráfico: Captura el gráfico más reciente en forma de icono de sesión, para poder acceder a él y manipularlo.
- Abrir: Abre un archivo de sesión previamente guardado.
- Guardar: Guarda la sesión actual en un archivo.
- Guardar como: Guarda la sesión actual en el archivo deseado.

§ Menú de Datos

- Mostrar valores: Muestra una ventana con una simple vista (no editable) de los valores de las variables (todas o una parte seleccionada).
- Editar valores: Muestra una ventana con una hoja de cálculo en la cual podemos introducir cambios, añadir nuevas variables, y extender el número de observaciones. (La matriz de datos tiene que ser rectangular, con un número idéntico de observaciones para cada serie.)
- Gráficos: Aquí se puede elegir entre trazar un gráfico de series temporales, un diagrama de dispersión X-Y regular, un gráfico X-Y de impulsos (barras verticales), un gráfico X-Y “con separación de factores” (es decir, con los puntos en diferentes colores dependiendo del valor de una variable ficticia) y gráficos de caja (boxplots). Muestra un cuadro de diálogo donde se pueden especificar las variables a trazar. La manera más sencilla de rellenarlo es refiriéndose a las variables con sus números de identificación (se encuentran en la primera columna de la izquierda en la ventana principal de datos). Por lo tanto, si hemos elegido la opción de diagrama de dispersión, poniendo “2 3” como los valores de los datos, se dibuja la variable 2 (aquí, el consumo) contra la variable 3 (renta). La última variable referenciada será situada en el eje x . El programa utiliza `gnuplot` para hacer los gráficos (excepto en la opción gráficos de caja).
- Gráficos bivariantes múltiples: Muestra una colección de (hasta un máximo de seis) diagramas, bien como una variable en el eje y contra varias variables en el eje x , o como varias variables en el eje y contra uno en el eje x . Puede ser útil para realizar un análisis preliminar de los datos.
- Leer información, Editar información: “Leer información” simplemente muestra la información de cabecera del archivo actual; “Editar información” permite cambiar esta información (en Linux, si tenemos los permisos para hacerlo).
- Estadísticos principales: Muestra un conjunto de estadísticos descriptivos bastante completo respecto de todas las variables incluidas en el conjunto de datos.

- Matriz de correlación: Muestra los coeficientes de correlación de cada par de variables incluidas en el conjunto de datos.
- Diferencia de medias: calcula el estadístico t para la hipótesis nula de que las medias poblacionales sean iguales para dos variables seleccionadas y muestra su valor p .
- Diferencia de varianzas: Calcula el estadístico F para la hipótesis nula de que las varianzas poblacionales sean iguales para dos variables seleccionadas y muestra su valor p .
- Añadir variables: Ofrece un submenú con las transformaciones de variables más típicas (logaritmos, retardos, cuadrados, etc.) las cuales pueden añadirse al conjunto de datos. También se da la opción de añadir variables aleatorias, y (para datos de series temporales) añadir variables ficticias estacionales (por ejemplo, variables ficticias trimestrales para datos trimestrales). Incluye una opción para poner una semilla en el generador de números pseudoaleatorios del programa.
- Actualizar ventana: A veces las instrucciones de `gretl` generan variables nuevas. La “actualización” asegura que el listado de variables visibles en la ventana principal de datos esté sincronizado con el estado interno del programa.

§ Menú de muestra

- Establecer rango: Establecer un punto de partida y/o terminación diferente para la muestra actual, dentro del rango de los datos disponibles.
- Restaurar el rango completo: *idem*.
- Establecer frecuencia, observación inicial: Impone una interpretación particular de los datos en términos de frecuencia y observación inicial. Está pensado principalmente para datos de panel; véase [Capítulo 5](#).
- Definir a partir de v. ficticia: Dada una variable ficticia (indicador) con valor 0 o 1, se eliminan todas las observaciones de la muestra actual donde la variable ficticia tenga el valor 0.
- Restringir a partir de criterio: Similar al anterior excepto que no necesitamos una variable predefinida: escribimos una expresión booleana (por ejemplo `sqft > 1400`) y la muestra se restringe a las observaciones que cumplen este requisito. Véase la ayuda para `genr` en [Capítulo 9](#) para detalles sobre los operadores booleanos que se pueden utilizar.
- Quitar todas las obs. con valores perdidos: Elimina de la muestra actual todas las observaciones para las cuales hay, por lo menos, una variable que tiene un valor perdido (véase [la sección de nombre Datos ausentes en Capítulo 4](#)).
- Contar valores perdidos: Emite un informe sobre las observaciones en las que faltan valores de los datos. Puede ser útil para examinar conjuntos de datos de panel, en los cuales suele faltar algún que otro valor.
- Añadir marcadores de caja: Pregunta por el nombre de un archivo de texto que contenga marcadores de caja (etiquetas asociadas a observaciones individuales) y añade esta información al conjunto de datos. Véase el [Capítulo 4](#).
- Reestructurar panel: Abre un cuadro de diálogo que nos permite determinar la interpretación de un conjunto de datos panel, bien como series temporales apiladas o como muestras transversales apiladas (véase [Capítulo 5](#)).

§ Menú de variable: La mayoría de las opciones que encontramos aquí actúan sobre las variables una por una. Podemos elegir la variable “activa” seleccionándola y pulsando sobre ella en la ventana principal de datos. La mayoría de las opciones no necesitan explicación. Nótese que es posible renombrar una variable y editar su descripción. También es posible “Definir una nueva variable” mediante una fórmula (por ejemplo, con una función de alguna/s variable/s ya existentes). Para conocer la sintaxis de estas fór-

mulas, consúltase el término `genr` en la ayuda *en línea* o véase la instrucción `genr` en el [Capítulo 9](#). Un ejemplo sencillo sería:

```
foo = x1 * x2
```

esto crearía una nueva variable `foo` como el producto de las variables `x1` y `x2` ya existentes. En estas fórmulas hay que referenciar las variables por su nombre, y no por su número.

- § Menú de modelo: Ya comentado en el [Capítulo 2](#). Para más detalles sobre los estimadores ofrecidos en este menú, consúltase [la sección de nombre *Estimadores y contrastes: resumen en Capítulo 9*](#), el [Capítulo 9](#) a continuación, y/o la ayuda *en línea* sobre “estimación”.
- § Menú de ayuda: Incluye detalles sobre la sintaxis requerida en los cuadros de diálogo. (Nota del traductor: el fichero de ayuda en Español suele actualizarse con retraso, por eso, a partir de la versión gretl-1.0.9 se ha añadido en la ventana de ayuda una nueva opción de menú que muestra la ayuda en Inglés, normalmente más actualizada y que mostrará también la ayuda sobre las nuevas instrucciones añadidas en la última versión))

La barra de herramientas de gretl

La barra de herramientas está situada en la esquina inferior-izquierda de la ventana principal.



Los iconos tienen las siguientes funciones, empezando de izquierda a derecha:

1. Lanzar calculadora. Es una función cómoda si queremos un rápido acceso a una calculadora mientras trabajamos en gretl. El programa utilizado por defecto es `calc.exe` en MS Windows, o `xcalc` en el sistema de ventanas X. Es posible cambiar el programa desde el menú “Archivo, Preferencias, General”, y submenú “Barra de herramientas”.
2. Lanzar editor o procesador de textos. Por defecto `winword.exe` en MS Windows, y `emacs` en X. Se puede configurar de la misma manera que la calculadora.
3. Abrir consola gretl. Un acceso rápido a la entrada del menú “Consola gretl” (ver más arriba [la sección de nombre *Menús de la ventana principal*](#)).
4. Vista de iconos de sesión.
5. Abrir el sitio web de gretl en nuestro navegador de Internet. Esto sólo funcionará en el caso de que el ordenador esté conectado a Internet y el navegador esté configurado de manera correcta.
6. Abrir la versión actual de este manual en formato PDF. Requiere una conexión con Internet; además el navegador ha de tener la capacidad de procesar archivos PDF.
7. Abrir la ayuda sobre la sintaxis de lotes de instrucciones (es decir, un listado detallando todas las instrucciones disponibles).
8. Abrir un cuadro de diálogo para definir un gráfico.
9. Grabar el último gráfico mostrado, para modificarlo. (Este botón no se muestra en Linux, donde se puede grabar el gráfico mediante su menú desplegable)
10. Abrir una ventana mostrando los conjuntos de datos asociados al libro *Introductory Econometrics* de Ramanathan (alternativamente puede configurarse este botón para que muestre los conjuntos de datos de Wooldridge (2002)- véase el [Capítulo 8](#)).

Para desactivar la barra de herramientas, ir a al menú “Archivo, preferencias, general”, y desmarcar la casilla de verificación “Mostrar barra de herramientas gretl”.

Capítulo 3. Modos de trabajo

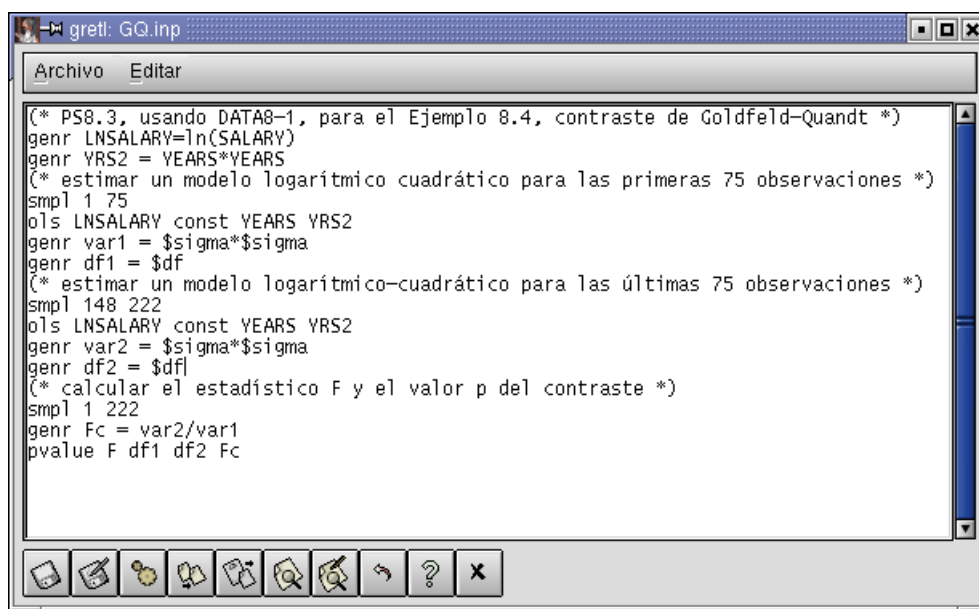
Lotes de instrucciones

A medida que vayamos ejecutando instrucciones en gretl, utilizando el GUI y rellenando cuadros de diálogo, estas instrucciones quedan registradas en forma de un “lote”. Podemos modificar o reejecutar estos lotes de instrucciones mediante gretl o el cliente de línea de instrucciones, gretlcli.

Para visualizar el estado actual del lote durante una sesión gretl, se selecciona “Ver historial de instrucciones” dentro del menú Archivo. Este archivo contiene el historial, llamado `session.inp`, y se sobrescribe cada vez que empezamos una nueva sesión. Para conservarlo, hay que guardarlo con otro nombre. Será más fácil encontrar los archivos de lotes utilizando el selector de archivos GUI, si se nombran con la extensión “.inp”.

Para abrir un lote instrucciones escrito de manera independiente, úsese la opción “Archivo, Abrir archivo de instrucciones, archivo de usuario” en el menú; para crear un lote desde cero, se utiliza la opción “Archivo, Nuevo archivo de instrucciones”. En ambos casos se abrirá una ventana para escribir las instrucciones (véase la [Figura 3-1](#)).

Figura 3-1. Ventana de lotes de instrucciones, editando un archivo de instrucciones



La barra de herramientas en la esquina inferior-izquierda de la ventana principal ofrece las siguientes opciones (de izquierda a derecha): (1) Guardar el archivo; (2) Guardar como; (3) Ejecutar las instrucciones en el archivo; (4) Copiar el texto seleccionado; (5) Pegar el texto seleccionado; (6) Buscar y reemplazar; (7) Deshacer la última acción de Pegar o Reemplazar; (8) Ayuda (si colocamos el cursor sobre una palabra de la instrucción y pulsamos el signo de interrogación, se muestra la ayuda sobre esta instrucción); (9) Cerrar la ventana.

Estas funciones (y además la opción de Imprimir) se encuentran también en los menús “Archivo” y “Editar” en la parte superior de la ventana de instrucciones.

Al pulsar el icono Ejecutar o al elegir la opción del menú “Archivo, Ejecutar” los resultados se dirigen a una sola ventana, desde donde es posible editarlos, guardarlos o copiarlos al portapapeles.

Para conocer más a fondo las posibilidades de los lotes de instrucciones, consúltese la Ayuda sobre “Sintaxis de guión de instrucciones” de gretl, o alternativamente, iniciar el programa de línea de instrucciones gretlcli y consultar su ayuda. Una tercera posibilidad

es leer el [Capítulo 9](#) de este manual. Además, el paquete `gretl` incluye más de 70 lotes de “ejercicios”. La mayoría de ellos corresponden al libro de Ramanathan (2002), pero independientemente de ello también pueden utilizarse como una introducción al modo de escribir lotes de instrucciones en `gretl` así como a varios temas de teoría econométrica. Se puede acceder a los ejercicios desde “Archivo, Abrir archivo de instrucciones, Archivo de prácticas”. Allí hay un listado de los ficheros junto con una breve descripción de los puntos que estos ilustran y los datos que utilizan. Abra cualquier archivo y ejecútelo (“Archivo, Ejecutar” en la ventana de lotes de instrucciones resultante) para ver los resultados.

Nótese que es posible separar las instrucciones largas en un lote de instrucciones en dos o más líneas utilizando la barra inversa como carácter de continuidad.

Es posible, si así lo desea el usuario, usar los controles GUI y los lotes de modo simultáneo, según cual sea el método más apropiado en cada momento. Hay dos sugerencias a continuación.

§ Abrir un archivo de datos en el GUI. Explorar los datos, generar los gráficos, ejecutar las regresiones, hacer los contrastes. Luego abrir el historial de instrucciones, quitar cualquier instrucción redundante y guardarlo con un nombre diferente. Ejecutar el lote de instrucciones para generar un único archivo que contendrá un historial conciso de nuestro trabajo.

§ Empezar creando un nuevo archivo de lotes. Teclear cualquier instrucción que sea necesaria para poner en marcha las transformaciones de los datos (véase la instrucción `genr` en el [Capítulo 9](#) más adelante). Normalmente, este tipo de tareas son más fáciles de realizar si se han pensado las instrucciones con antelación y no pulsando sobre la marcha. Luego guardar y ejecutar el lote: la ventana de datos del GUI se actualizará en consecuencia. Ahora podemos manipular los datos mediante el GUI. Para revisitar los datos más tarde, abrir y ejecutar en primer lugar el lote “preliminar”.

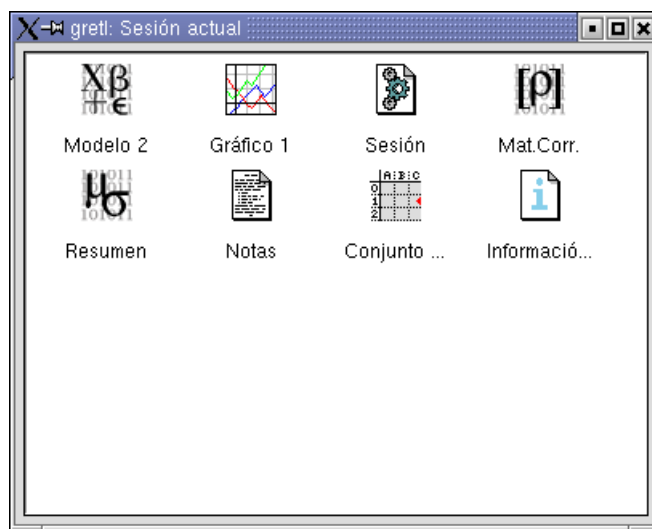
Hay otra opción para hacer más cómoda la tarea. En el menú Archivo de `gretl` se encuentra la opción “Consola `gretl`”. Esta abre una ventana donde se pueden teclear las instrucciones y ejecutarlas una por una (con la tecla Retorno) de manera interactiva. Esencialmente, es el mismo modo de operación que `gretlcli`, excepto que (a) el GUI es actualizado basándose en las instrucciones ejecutadas desde la consola, haciendo posible el trabajo en cualquier entorno, y que (b) por ahora, la tarea de simulaciones de Monte Carlo de `gretl` (véase la [sección de nombre Simulaciones Monte Carlo en Capítulo 7](#)) no está disponible en este modo.

El concepto de “sesión”

`Gretl` ofrece la idea de una “sesión” como un modo de seguimiento de los trabajos realizados y la posibilidad de volver a ellos más tarde. Se encuentra en un estado experimental (y en la actualidad hay más posibilidades de encontrar errores aquí que en el resto del programa); el autor está interesado en conocer las opiniones de los usuarios a este respecto.

La idea inicial es ofrecer un pequeño espacio con iconos, que contenga varios objetos pertenecientes a la sesión de trabajo actual (véase la [Figura 3-2](#)). Después, se pueden añadir objetos (representados por iconos) a este espacio. Si se guarda la sesión, estos objetos añadidos estarán disponibles cuando se vuelva a abrir la sesión.

Figura 3-2. Vista de iconos: a los iconos por defecto, se ha añadido un modelo y un gráfico



Al iniciar gretl, abriendo un conjunto de datos, y seleccionando “Vista de iconos” del menú Session, se visualizan los iconos básicos que aparecen por defecto: estos son una manera rápida de acceder al lote de instrucciones (“Sesión”), información sobre los datos (si la hay), la matriz de correlación y estadísticas principales. Todos estos se activan mediante un doble clic sobre el icono deseado. El icono de “Conjunto de datos” es un poco más complejo: al hacer doble clic se abre una hoja de cálculo con los datos incorporados, pero también se despliega un menú con otras opciones si se pulsa el botón derecho del ratón.



En muchas ventanas de gretl, al hacer clic con el botón derecho del ratón se despliega un menú con una lista de tareas habituales.

Es posible añadir dos tipos de objetos a la ventana de vista de iconos: modelos y gráficos. Para añadir un modelo, primero hay que estimarlo utilizando el menú de Modelo. Luego se pulsa la opción Archivo en la ventana de modelo y se selecciona “Guardar a sesión como icono...” o “Guardar como icono y cerrar”. Se puede atajar a la segunda opción pulsando la tecla S sobre la ventana de modelo.

Para añadir un gráfico, primero hay que crearlo (en el menú Datos, “Gráficos”, o mediante una de las otras instrucciones de gretl para generar gráficos), luego se selecciona “Añadir último gráfico” desde el menú Sesión o se pulsa el ratón sobre el botón con el dibujo de una pequeña cámara fotográfica en la barra de herramientas (véase [la sección de nombre La barra de herramientas de gretl en Capítulo 2](#)).

Una vez añadido el modelo o gráfico, su icono aparecerá en la ventana de vista de iconos. Haciendo doble clic sobre el icono se vuelve a mostrar el objeto, mientras que pulsando el botón derecho del ratón se despliega un menú que permite mostrar o eliminar el objeto. Este menú desplegable también ofrece la posibilidad de editar los gráficos.

Si se crean modelos o gráficos y tenemos la intención de volver a utilizarlos más adelante, entonces conviene seleccionar “Guardar como” del menú Sesión antes de salir de gretl, dando un nombre a la sesión. Para volver a abrir la sesión,

§ Iniciar gretl y volver a abrir el archivo de la sesión desde la opción “Abrir” en el menú Sesión, o

§ Desde una línea de instrucciones, teclear `gretl -r archivo de sesión`, donde *archivo de sesión* es el nombre bajo el cual se había guardado la sesión.

Capítulo 4. Fichero de datos

Formato propio

Gretl tiene su propio formato para los archivos de datos. La mayoría de los usuarios probablemente no querrán escribir o leer estos archivos desde fuera de gretl, pero ocasionalmente esto puede ser útil. Los detalles completos sobre los formatos de los archivos están en el [Apéndice A](#).

Otros formatos de archivos de datos

Gretl puede leer varios otros formatos de datos.

§ Ficheros CSV (comma-separated values — valores separados por comas). Estos archivos pueden importarse mediante la opción del menú “Archivo, Abrir datos, Importar CSV...” de gretl, o mediante la instrucción de consola `import`. Para conocer lo que gretl espera encontrar en un archivo CSV, véase [la sección de nombre Crear un archivo de datos desde cero](#).

§ Libros de trabajo en formatos MS Excel o Gnumeric. Estos archivos también pueden importarse utilizando la opción “Archivo, Abrir datos, Importar” del menú. Los requisitos para este tipo de archivos se detallan en [la sección de nombre Crear un archivo de datos desde cero](#).

§ Datos en formato BOX1. Hay una gran cantidad de datos micro disponibles (gratis) en este formato mediante el servicio de extracción de datos del Censo de los Estados Unidos. Los datos BOX1 pueden importarse con la opción del menú “Archivo, Abrir datos, importar BOX” o la orden de consola `import -o`.

Al importar datos desde los formatos CSV o BOX, gretl abre una ventana “de diagnóstico”, informando sobre su progreso al leer los datos. Si hay algún problema con datos mal formateados, los mensajes de esta ventana ayudarán a resolver el problema.

Para comodidad de quienes desean llevar a cabo análisis de datos más complejos, gretl cuenta con la opción de escribir los datos en los formatos nativos de GNU R y GNU Octave (véase el [Apéndice D](#)). En el cliente del GUI se accede a esta opción a través del menú “Archivo”; en el cliente de línea de instrucciones se utiliza la instrucción `store` con la marca `-r` (R) o `-m` (Octave).

Bases de datos binarias

Para trabajar con grandes cantidades de datos se ha dotado a gretl de una rutina para manejar bases de datos. Una *base de datos*, al contrario que un *archivo de datos*, no puede ser leída directamente al espacio de trabajo del programa. Una base de datos puede contener series con una gran variedad de frecuencias y rangos en los datos. Normalmente, se abre la base de datos y se selecciona la serie a importar al conjunto de datos con el que estamos trabajando. Luego, pueden guardarse estas series en el formato propio de archivos de datos. Es posible acceder a las bases de datos mediante la opción “Archivo, Revisar bases de datos)” en el menú de gretl.

Para más detalles sobre el formato de las bases de datos de gretl, véase el [Apéndice A](#).

Acceso en línea a bases de datos

Desde la versión 0.40, gretl puede acceder a bases de datos a través de internet. Hay varias bases de datos disponibles desde la Wake Forest University. El ordenador debe estar conectado a internet para que funcione esta opción. Por favor, consúltase el apartado sobre “Bases de datos en línea” en el menú de Ayuda de gretl.

Bases de datos RATS 4

Gracias a Thomas Doan de *Estima*, quien me facilitó las especificaciones del formato utilizado por las bases de datos en RATS 4 (Regression Analysis of Time Series), gretl puede trabajar con este tipo de bases de datos. Bien, en realidad con una parte de ellos: sólo se trabaja con bases de datos de series temporales que contengan series mensuales y trimestrales. Mi universidad tiene la base de datos de RATS G7 que contiene los datos de las siete economías más grandes de la OCDE y gretl puede leerlos sin problemas.



Visite la página de datos de gretl para más detalles y actualizaciones de los datos disponibles.

Crear un archivo de datos desde cero

Hay cuatro maneras de hacer esto: (1) Usando nuestra hoja de cálculo favorita para crear el archivo de datos, guardándola en formato CSV (comma-separated values – valores separados por comas) si es necesario, y usando finalmente la opción “Importar” de gretl. (2) Usando la hoja de cálculo incorporada en el propio gretl. (3) Seleccionando las series de datos en una base de datos adecuada. (4) Usando nuestro editor de textos habitual u otras herramientas de software para crear el archivo de datos de manera independiente.

A continuación, hay algunos comentarios y detalles sobre estos métodos.

Utilización de una hoja de cálculo diferente

Esta opción puede ser buena si se siente más cómodo con alguna hoja de cálculo en particular. Si se elige esta opción, hay que tener cuidado en especificar la apariencia final de la hoja de cálculo. gretl espera un archivo que tenga (a) nombres válidos de las variables en la primera fila y (b) un bloque rectangular de datos debajo de ésta. Un nombre válido para una variable contiene 8 caracteres como máximo; empieza con una letra; y no esta compuesto por nada excepto letras, números y el carácter de subrayado, `_`. Opcionalmente, la primera columna puede contener etiquetas de fechas (máximo 8 caracteres). Una columna de este tipo debe estar encabezada por “obs” o “date”, o en su defecto, tener vacía la primera celda de la primera fila. Debe de haber *exactamente una* fila sin datos encabezando el archivo.

El formato descrito arriba debe de ser respetado de manera exacta en el caso de importaciones desde CSV. En el caso de importaciones desde Excel o Gnumeric, hay un poco más de libertad: se puede seleccionar la fila y columna, desde donde hay que empezar la importación y así evitar cualquier fila o columna “extra”. Se aplican las normas descritas anteriormente dentro del área seleccionada para importar.

Si se utiliza una hoja de cálculo para preparar los datos, es fácil hacer varias transformaciones de los datos “originales” (hacer sumas, porcentajes, o lo que sea): sin embargo, nótese que pueden hacerse estas cosas fácilmente - quizá con mayor facilidad - en gretl, utilizando las herramientas disponibles en el menú “Datos, Añadir variables” y/o “Variable, definir nueva variable”.

Si así se desea, es posible establecer un conjunto de datos en gretl pieza a pieza, incrementando los datos mediante importaciones desde otras fuentes. Se puede llevar a cabo esto mediante las opciones del menú “Fichero, Añadir datos”. gretl comprobará que los datos nuevos son compatibles con el conjunto de datos ya existente, y, si todo le parece bien, fusionará los datos. De esta manera es posible añadir nuevas variables, con tal de que la frecuencia de datos, la primera observación y la longitud de la serie sean iguales al conjunto de datos ya existente. O también se pueden añadir nuevas observaciones a las series de datos que ya tenemos; en este caso los nombres de las variables tienen que ser iguales. Nótese que por defecto, (es decir, si se elige “Abrir datos” en vez de “Añadir datos”), abrir un nuevo archivo de datos significa el cierre del que esté abierto actualmente.

Utilizar la hoja de cálculo incorporada

En el menú “Archivo, Crear conjunto de datos” de `gretl`, puede elegirse el tipo de conjunto de datos que se desea establecer (por ejemplo, series temporales trimestrales, mensuales). Entonces el programa pregunta por las fechas de inicio y fin (o el número de observaciones) y el nombre de la primera variable a añadir al conjunto de datos. Después de dar esta información, se abre una sencilla hoja de cálculo donde pueden introducirse los valores de los datos. Dentro de la ventana de la hoja de cálculo, pulsando el botón derecho del ratón se invoca un menú desplegable que brinda la opción de añadir una nueva variable (columna), añadir una observación (añadir una fila al final de la hoja), o introducir una observación en el punto deseado (mover e insertar una nueva fila en blanco).

Una vez que se hayan introducido los datos en la hoja de cálculo, pueden ser importados al espacio de trabajo de `gretl`, utilizando el botón “Aplicar los cambios ” de la hoja de cálculo.

Nótese que la hoja de cálculo de `gretl` es bastante básica y no tiene soporte para funciones o fórmulas. Las transformaciones de los datos se hacen a través de los menús de “Datos” o de “Variable” en la ventana principal de `gretl`.

Seleccionar desde un base de datos

Otra alternativa consiste en organizar el archivo de datos seleccionando variables desde una base de datos. `Gretl` viene con una base de datos de series temporales macroeconómicas de EEUU, y como se ha dicho anteriormente, el programa puede leer bases de datos de RATS 4.

Se comienza con la opción de menú “Archivo, Revisar bases de datos ” de `gretl`. Esta contiene tres posibilidades: “nativa `gretl`”, “RATS 4” y “en servidor” (en servidor de bases de datos). El archivo `bc1h.bin` se encuentra en el selector de archivos que se abre al elegir la opción “nativa `gretl`” - este archivo viene con el paquete del programa.

Si no se han comprado o generado datos en formato RATS¹, no hay nada en “RATS 4” Si tiene datos en formato RATS, vaya al cuadro de dialogo “Archivo, Preferencias, General ” de `gretl`, seleccione la ficha Bases de Datos, y rellene el camino correcto hacia los archivos RATS.

Si el ordenador esta conectado a Internet, es posible encontrar varias bases de datos (en la Wake Forest University) en la opción “en servidor”. Pueden consultarse en modo remoto, y también existe la opción de instalarlos en el ordenador. La ventana inicial de las bases de datos remotos tiene una opción para cada archivo que muestra si está ya instalado localmente (y si es así, también muestra si esta versión está actualizada con la versión en Wake Forest).

Suponiendo que se ha podido abrir una base de datos, es posible importar las series seleccionadas al espacio de trabajo de `gretl`, mediante la opción de menú “Importar” en la ventana de bases de datos (o vía el menú desplegable que aparece al pulsar el botón derecho del ratón).

Crear un archivo de datos independiente

Es posible crear un archivo de datos usando un editor de textos o herramienta de software como `awk`, `sed` o `perl`. Esto puede ser una buena elección si se dispone de una gran cantidad de datos en formato legible para ordenadores. Por supuesto hay que familiarizarse con los formatos de datos de `gretl` (formato XML o “tradicional”) tal como se describe en el [Capítulo 4](#).

Nota adicional

`Gretl` no tiene ningún problema al compactar series de datos de unas frecuencias relativamente altas (por ejemplo mensuales) a una frecuencia más baja (por ejemplo trimestrales):

1. Ver www.estima.com

esto se hace sacando las medias correspondientes. Pero no tiene ningún método para convertir datos de frecuencias más bajas a las más altas. Por lo tanto, si se quieren importar series de diferentes frecuencias a una base de datos de gretl, *tenemos que empezar importando la serie de más baja frecuencia que queramos usar*. Esto iniciará el archivo de datos en gretl a la frecuencia más baja, pudiéndose importar los datos de frecuencias más altas después (se compactarán automáticamente). Si se empieza con series de frecuencias altas no es posible importar ninguna serie de frecuencia más baja que la original.

Datos ausentes

Los datos ausentes se representan como -999. En un archivo de datos en formato nativo deberían de estar representados del mismo modo. Al importar datos CSV, gretl acepta cualquiera de estas tres representaciones para valores ausentes: -999, la cadena NA, o una celda vacía. Por supuesto, las celdas vacías tienen que estar delimitadas correctamente, por ejemplo 120.6, ,5.38: donde se supone que no existe el segundo valor.

Gretl hace lo siguiente en cuanto al manejo de valores ausentes a la hora de hacer análisis estadísticos:

- § Al calcular estadísticos descriptivos (medias, desviaciones estándar, etc.) con la instrucción `summary`, simplemente se omiten los valores ausentes, y se ajusta el tamaño de la muestra.
- § Al ejecutar regresiones, gretl ajusta primero el principio y el final del rango de la muestra, truncando la muestra si es necesario. Es corriente que falten datos al principio de una serie temporal por la inclusión de retardos, primeras diferencias, etc; y no es inusual que falten valores al final de una serie debido a diferentes actualizaciones de las mismas y quizá también, debido a la inclusión de avances.
- § En caso de que gretl detecte que haya datos ausentes “dentro” de una muestra (posiblemente truncada), da un mensaje de error y se niega a hacer estimaciones.

Los valores ausentes dentro de un conjunto de datos representan un problema. En un conjunto de datos transversales, puede que sea posible mover estas observaciones hacia el principio o hacia el final del archivo, pero esto es obviamente imposible con datos de series temporales. Para quienes sepan lo que están haciendo (!), existe la opción `misszero` en la instrucción `genr`. Haciendo

```
genr foo = misszero(bar)
```

se crea una serie `foo` que es idéntica a `bar`, con la excepción de que cualquier valor -999 aparece como cero. Entonces es posible utilizar variables ficticias (construidas de forma adecuada) para, de hecho, dejar fuera de la regresión los valores ausentes y retener la muestra que los rodea.²

2. `genr` también ofrece la opción inversa a `misszero`, a saber `zeromiss`, la cual reemplaza los ceros en una serie dada con el código de observación ausente.

Capítulo 5. Datos de Panel

Estructura de Panel

Los datos de panel (una muestra combinada de datos de series temporales y de sección cruzada) requieren un cuidado especial. He aquí algunas observaciones a tener en cuenta.

Considérese un conjunto de datos consistente en observaciones de n unidades de sección cruzada (países, provincias, personas, etc.) durante T periodos. Supongamos que cada observación contiene los valores de m variables de interés. El conjunto de datos está formado entonces por mnT valores.

Los datos deben de ordenarse “por observación”: cada fila representa una observación; cada columna contiene los valores de una variable en particular. La matriz de datos tiene entonces nT filas y m columnas. Esto deja abierta la cuestión de cómo ordenar las filas. Existen dos posibilidades.¹

§ Filas agrupadas por *unidad*. Piénsese en la matriz de datos como si estuviera compuesta de n bloques, cada uno con T filas. El primer bloque de T filas contiene las observaciones de la unidad 1 de la muestra para cada uno de los periodos; el siguiente bloque contiene las observaciones de la unidad 2 para todos los periodos; y así sucesivamente. De hecho, la matriz de datos es un conjunto de datos de series temporales apilados verticalmente.

§ Filas agrupadas por *periodo*. Piénsese en la matriz de datos como si estuviera compuesta por T bloques, cada uno con n filas. La primera de las n filas contiene las observaciones de cada unidad muestral en el periodo 1; el siguiente bloque contiene las observaciones de todas las unidades en el periodo 2; y así sucesivamente. La matriz de datos es un conjunto de datos de muestras de sección cruzada, apiladas verticalmente.

Puede utilizarse el esquema que resulte más conveniente. El primero es quizá más fácil de mantener ordenado. Si se utiliza el segundo, hay que asegurarse de que las unidades de sección cruzada aparezcan en el mismo orden en cada uno de los bloques de datos de cada periodo.

En cualquiera de los dos casos se puede utilizar el campo frecuencia en la línea *observaciones* del archivo de cabecera de datos (véase el [Capítulo 4](#)) para que el asunto resulte un poco más sencillo.

§ *Agrupados por unidades*: Establecer la frecuencia igual a T . Supongamos que hay observaciones sobre 20 unidades durante 5 periodos de tiempo. En este caso, la línea de observaciones más apropiada es la siguiente: 5 1.1 20.5 (léase: frecuencia 5, empezando con la observación de la unidad 1, en el periodo 1, y finalizando con la observación de la unidad 20, periodo 5). Entonces, por ejemplo, la observación de la unidad 2 en el periodo 5 puede ser referenciada como 2.5, y la correspondiente a la unidad 13 en periodo 1 como 13.1.

§ *Agrupado por periodos*: Establecer la frecuencia igual a n . En este caso, si hay observaciones sobre 20 unidades en cada uno de los 5 periodos, la línea de observaciones debería ser: 20 1.01 5.20 (léase: frecuencia 20, empezando con la observación del periodo 1, unidad 01, y finalizando con la observación del periodo 5, unidad 20). Así, nos referiremos a la observación de la unidad 2, periodo 5 como 5.02.

Si se construye un conjunto de datos de panel utilizando un programa de hoja de cálculo para después importar los datos a gretl, puede ser que el programa no reconozca, al principio, la clase especial de los datos. Esto se puede arreglar mediante la instrucción `setobs` (véase el [Capítulo 9](#)) o la opción del menú GUI “Muestra, Seleccionar frecuencia, observación inicial...”.

1. Si no queremos diferenciar de manera conceptual o estadística entre variaciones muestrales y temporales, podemos ordenar las filas de modo arbitrario, pero esto es probablemente un derroche de datos.

Variables ficticias

En un estudio de panel puede que se desee construir variables ficticias de uno o ambos tipos descritos a continuación: (a) variables ficticias como identificadores de las unidades muestrales, y (b) variables ficticias como identificadores de los periodos de tiempo. El primer método puede utilizarse para permitir que el intercepto de la regresión sea diferente en diferentes unidades, y el segundo para permitir lo mismo en diferentes periodos.

Hay dos opciones especiales para crear estas variables ficticias. Se encuentran dentro del menú “Datos, Añadir variables” en el GUI, o en la instrucción `genr` en el modo lote de instrucciones, o `gretlcli`.

1. “variables ficticias periódicas” (lote de instrucciones: `genr dummy`). Esta instrucción normalmente se utiliza para crear variables ficticias periódicas hasta la frecuencia de datos en los estudios de series temporales (por ejemplo un conjunto de variables ficticias trimestrales para ser utilizado en corrección estacional). No obstante, también funciona con datos de panel. Nótese que la interpretación de las variables ficticias creadas mediante esta instrucción difiere dependiendo de si las filas de datos están agrupadas por unidad o por periodo. Si están agrupadas según *unidades* (frecuencia T) las variables resultantes son *variables ficticias periódicas* y habrá un número T de ellas. Por ejemplo, `dummy_2` tendrá el valor 1 en cada fila de datos correspondiente a una observación del periodo 2, o 0 en caso contrario. Si están agrupadas según *periodos* (frecuencia n) entonces se generaran n *variables ficticias unitarias*: `dummy_2` tendrá el valor 1 en cada fila de datos asociada con la unidad muestral 2, o 0 en caso contrario.
2. “Variables ficticias de panel” (en modo consola `genr panelum`). Esta instrucción crea todas las variables ficticias, de cada unidad y periodo, de golpe. Se supone que por defecto, las filas de datos están agrupadas por unidades. Las variables ficticias de cada unidad se denominan `du_1`, `du_2` y así sucesivamente, mientras que las variables ficticias periódicas se llaman `dt_1`, `dt_2`, etc. Es incorrecto utilizar la u (por unidad) y la t (por tiempo) en estos nombres si las filas de datos están agrupadas por periodos: su utilización correcta en este contexto se hace mediante `genr panelum -o` (sólo en modo lote de instrucciones).

Si el conjunto de datos de panel contiene el año `YEAR` como una de las variables, es posible crear un periodo ficticio para escoger algún año en particular como en este ejemplo `genr dum = (YEAR=1960)`. También es posible crear variables ficticias periódicas utilizando el operador de módulo, `%`. Por ejemplo, para crear una variable ficticia con valor 1 para la primera observación y cada treinta observaciones y 0 en los demás casos, se puede hacer lo siguiente

```
genr index
genr dum = ((index-1)%30) = 0
```

Uso de valores retardados con datos de panel

Si los periodos de tiempo están divididos en intervalos regulares, quizá queramos usar los valores retardados de las variables en una regresión de panel. En este caso es preferible agrupar las filas de datos por *unidades* (series temporales apiladas).

Supongamos que creamos un retardo de la variable `x1`, utilizando `genr x1_1 = x1(-1)`. Los valores de esta variable serán en general correctos, pero en los límites de los bloques de datos de cada unidad son “utilizables”: el valor “previo” no es realmente el primer retardo de `x1_1`, si no más bien la última observación de `x1` para la unidad muestral previa. `Gretl` marca estos valores como ausentes.

Si hay que incluir un retardo de este tipo en una regresión, hay que asegurarse de que la primera observación de cada bloque o unidad no esté incluida. Un modo de hacer esto es mediante Mínimos Cuadrados Ponderados (`wls`) utilizando una variable ficticia apropiada como ponderación. Esta variable ficticia (vamos a denominarla `lagdum`) debe tener el valor 0 para las observaciones a descartar, y 1 en el caso contrario. Es decir, es complementaria a

una variable para el periodo 1. De este modo, si hemos utilizado la instrucción `genr dummy` podemos teclear `genr lagdum = 1 - dummy_1`. En caso de que hubiéramos utilizado `genr paneldum` ahora tendríamos que teclear `genr lagdum = 1 - dt_1`. De cualquier manera, la siguiente instrucción sería

```
wls lagdum y const x1_1 ...
```

para obtener una regresión combinada utilizando el primer retardo de `x1`, descartando todas las observaciones del periodo 1.

Otra opción es utilizar `smp1` con la marca `-o` y una variable ficticia apropiada. El [Ejemplo 5-2](#) muestra unas instrucciones de ejemplo, suponiendo que cada bloque de datos de cada unidad contiene 30 observaciones y queremos descartar la primera fila de cada uno. Podemos entonces ejecutar las regresiones sobre el conjunto de datos restringido sin tener que usar la instrucción `wls`. Si se desea reutilizar el conjunto de datos restringido, podemos guardarlo mediante la instrucción `store` (véase el [Capítulo 9](#) más abajo).

Ejemplo 5-1. Retardos con datos de panel

```
(* crear la variable índice *)
genr index
(* crear dum = 0 para cada 30 observaciones *)
genr dum = ((index-1)%30) > 0
(* establecer la muestra por medio de esa variable ficticia *)
smp1 -o dum
(* crear de nuevo la estructura de observaciones, para 56 unidades *)
setobs 29 1.01 56.29
```

Estimación combinada

Llegados a este punto, podemos revelar que hay una instrucción de estimación con el propósito especial de ser utilizado con datos de panel, la opción “MCO combinados” en el menú **Modelo**. Esta instrucción sólo está disponible cuando se reconoce el conjunto de datos como un panel. Para aprovechar esta opción, es preciso especificar un modelo que no contenga ninguna variable ficticia para representar unidades de sección cruzada. La rutina presenta estimaciones sencillas de MCO combinadas, que tratan de igual manera las variaciones de sección cruzada y de series temporales. Este modelo puede que sea el apropiado o no. En el menú **Contrastes** en la ventana de modelo, se encuentra una opción llamada “Diagnósticos de panel”, la cual plantea el contraste de MCO combinados contra las principales alternativas, es decir, los modelos de efectos fijos o de efectos aleatorios.

El modelo de efectos fijos añade una variable ficticia a todas menos una de las unidades de sección cruzada, permitiendo que varíe el intercepto de la regresión en cada unidad. Se presenta un contraste F para la significación conjunta de estas variables ficticias: si el valor p para este contraste es pequeño, entonces se rechaza la hipótesis nula (de que un simple modelo combinado es adecuado) en favor de un modelo de efectos fijos.

Por otro lado, el modelo de efectos aleatorios descompone la varianza residual en dos partes, una parte específica a la unidad de sección cruzada o “grupo” y la otra específica a una observación en particular. (Este estimador sólo puede calcularse cuando el panel es lo suficientemente “amplio”, es decir, cuando el número de unidades de sección cruzada en el conjunto de datos excede el número de parámetros a estimar.) El contraste LM de Breusch-Pagan comprueba la hipótesis nula (una vez más, de que el estimador de MCO combinados es adecuado) contra la alternativa de efectos aleatorios.

Cabe dentro de lo posible que el modelo MCO combinados sea rechazado contra las dos alternativas de efectos fijos y aleatorios. Entonces la pregunta es, ¿cómo podemos valorar los méritos relativos de los estimadores alternativos? El contraste de Hausman (también incluido en el informe, siempre que el modelo de efectos aleatorios se pueda estimar) intenta resolver este problema. El estimador de efectos aleatorios es más eficiente que el estimador de efectos fijos, siempre y cuando el error específico a la unidad o grupo no esté correlacionado con las variables independientes; si no es así, el estimador de efectos

aleatorios es inconsistente, en cuyo caso es preferible el estimador de efectos fijos. La hipótesis nula para el contraste de Hausman dice que el error específico al grupo no está tan correlacionado (y por lo tanto es preferible el modelo de efectos aleatorios). Por lo tanto, un valor p pequeño para este contraste supone rechazar el modelo de efectos aleatorios en favor del modelo de efectos fijos.

Para una discusión más rigurosa sobre este tema, véase Greene (2000), capítulo 14.

Ilustración: La Tabla Mundial de Penn

La Tabla Mundial de Penn (Penn World Table) (dirección pwt.econ.upenn.edu) es un excelente conjunto de datos macroeconómicos de panel, que incluye datos sobre 152 países entre los años 1950-1992. Los datos están disponibles en formato gretl; véase el sitio web de datos de gretl http://gretl.sourceforge.net/gretl_data.html (se puede descargar gratuitamente, aunque no está incluido en el paquete principal de gretl).

El [Ejemplo 5-2](#) de abajo abre `pwt56_60_89.gdt`, un conjunto parcial de la pwt que contiene datos sobre 120 países, entre los años 1960-89, para 20 variables, sin que haya ninguna observación ausente (el conjunto de datos completo, que también está incluido en el paquete pwt para gretl, contiene muchas observaciones con valores ausentes). El total de crecimiento del PIB real, entre 1960-89, se calcula para cada país y se regresa contra el nivel real del PIB en 1960, para ver si hay indicios de “convergencia” (es decir, crecimiento más rápido en los países que empezaron con el nivel más bajo).

Ejemplo 5-2. Uso de la tabla mundial de Penn

```
open pwt56_60_89.gdt
(*
para 1989 (última observación), el retardo 29 da 1960,
la primera observación
*)
genr gdp60 = RGDPL(-29)
(*
encontrar el crecimiento total del PNB total
durante 30 años
*)
genr gdpgro = (RGDPL - gdp60)/gdp60
(* restringir la muestra a la sección cruzada de año 1989 *)
smpl -r YEAR=1989
(*
¿Hay convergencia? ¿los países con una base menor,
crecieron mas rápido?
*)
ols gdpgro const gdp60
(* resultado: ¡No! Intentar la relación inversa *)
genr gdp60inv = 1/gdp60
ols gdpgro const gdp60inv
(* No otra vez. ¿Intentar prescindir de Africa? *)
genr afdum = (CCODE = 1)
genr afslope = afdum * gdp60
ols gdpgro const afdum gdp60 afslope
```

Capítulo 6. Gráficos y Diagramas

Gráficos Gnuplot

Los gráficos se generan llamando a un programa que es independiente de Gretl, `gnuplot`. Gnuplot es un programa muy completo para la realización de gráficos con múltiples opciones. Está disponible en www.gnuplot.info (no obstante téngase en cuenta que una copia de gnuplot se incluye con la versión MS Windows de gretl). Mediante un interfaz gráfico, gretl da acceso directo a sólo una pequeña parte de las opciones de gnuplot, pero intenta elegir valores razonables; también permite controlar todos los detalles del gráfico si así se desea.

Los detalles para manejar los gráficos en gretl varían ligeramente entre Linux y MS Windows.

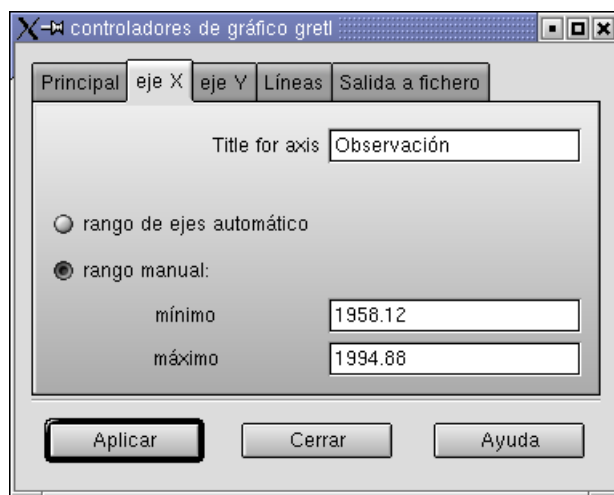
En Linux, pulsando con el ratón sobre el gráfico se muestra un menú desplegable, que nos da la opción de guardar el gráfico en un archivo (bien como archivo postscript encapsulado o bien en formato PNG) o de guardarlo en la sesión actual “como un icono”. En la mayoría de los gráficos también tenemos la opción de usar el zoom para inspeccionar las áreas que nos interesen. Si gretl se construye con soporte gnome este menú también ofrece la posibilidad de imprimir el gráfico.

En MS Windows, al hacer clic en la esquina superior-izquierda en la ventana del gráfico, se abre un menú desplegable de gnuplot que nos permite elegir entre varias opciones (incluyendo copiar el gráfico al portapapeles de Windows e imprimirlo). Este menú no incluye la opción de guardar el gráfico como un icono de sesión de gretl - para hacer esto, hay que cerrar la ventana del gráfico, y después, desde el menú de Sesión, elegir “Añadir último gráfico”. También se puede pulsar sobre el icono (en forma de una pequeña maquina de fotos) para hacer lo mismo.

Una vez que se guarda el gráfico como un icono de sesión, se pueden realizar más cambios a la medida deseada. Para ello, hay que abrir la ventana de icono de sesión, pulsar el botón derecho del ratón sobre el nuevo icono del gráfico y elegir entre “Editar utilizando GUI” o “Editar instrucciones de gráfico”. La opción “Editar utilizando GUI” despliega un controlador gráfico de gnuplot que permite afinar varios aspectos del gráfico. La opción “Editar instrucciones de gráfico” abre una ventana del editor que contiene el archivo de las instrucciones de gnuplot para generar el gráfico: este proporciona control total sobre los detalles del gráfico - en el caso de que tengamos conocimientos de gnuplot. Para más información, ver el manual en línea de gnuplot o ir a www.gnuplot.info.

También consúltese la entrada `gnuplot` más adelante en el [Capítulo 9](#) - y las instrucciones `graph` y `plot` para gráficos de ASCII “rápidos, es decir, estilo borrador”.

Figura 6-1. Controlador gnuplot de gretl



Gráficos de caja

Los gráficos de caja (boxplots) no son generados por gnuplot, sino mediante gretl.

Estos gráficos (llamados "boxplots" por Tukey y Chambers) muestran la distribución de una variable. La caja central abarca el 50 por ciento de los datos centrales, es decir, está flanqueada por el primer y el tercer cuartiles. Las "patillas" se extienden hasta los valores mínimos y máximos. Una línea atraviesa la caja por la mediana.

En el caso de los gráficos de caja recortados (notched boxplots), el corte muestra los límites de confianza de aproximadamente el 90 por ciento. Esto se obtiene mediante el método bootstrap, lo cual puede tardar un rato si la serie es muy larga.

Pulsando con el ratón en la ventana de los gráficos de caja (boxplots) se despliega un menú, que nos permite guardar los gráficos como postscript encapsulado (EPS), o como un archivo postscript de página completa. En el sistema de ventanas X también existe la posibilidad de guardar la ventana como un archivo XPM; y en MS Windows puede copiarse al portapapeles como un mapa de bits. El menú también da la opción de abrir una ventana de resumen, la cual muestra cinco valores de resumen (mínimo, primer cuartil, mediana, tercer cuartil y máximo), además de un intervalo de confianza para la mediana en el caso de que esté seleccionada la opción de "gráficos de caja recortados".

Algunos detalles de los gráficos de caja de gretl pueden controlarse mediante un archivo de texto llamado `.boxplotrc`, el programa busca este archivo en los siguientes sitios; primero, en la carpeta de trabajo actual, segundo, en la carpeta del usuario (correspondiente a la variable de entorno HOME), y por último, en la carpeta del usuario de gretl (esta última se muestra en el menú "Archivo, Preferencias, General" y puede ser cambiada). Las opciones que podemos seleccionar son las siguientes: primero, la fuente a utilizar en el archivo postscript resultante (el nombre genérico de la fuente tiene que ser válido para postscript; por defecto es Helvética); segundo, el tamaño de la fuente en puntos (también para la salida postscript, por defecto es 12); tercero, el mínimo y el máximo para el rango del eje, y la anchura y altura del gráfico en pixels (por defecto, 560 x 448); cuarto, si hay que imprimir los valores numéricos de los cuartiles y la media (por defecto, no imprimir), y por último, si los valores extremos o outliers (puntos más allá de 1,5 veces el rango recorrido intercuartílico desde la caja central) deben de indicarse por separado (por defecto, no). Veamos un ejemplo a continuación:

```
font = Times-Roman fontsize = 16 max = 4.0 min = 0 width = 400
height = 448
numbers = %3.2f
outliers = true
```

En la penúltima línea, el valor asociado con `numbers` es una cadena en formato "printf" como en el lenguaje de programación C; si se especifica, este valor controla la impresión de la mediana y los cuartiles próximos al gráfico de caja. Si no hay entrada de `numbers` estos valores no se imprimen. En el ejemplo se imprimirán los valores hasta 3 dígitos, con la precisión de 2 dígitos a partir de la coma.

No es necesario especificar todas las opciones y el orden tampoco importa. Las líneas que no siguen el esquema de "clave = valor" serán ignoradas, y también las líneas que empiecen con el símbolo #.

Después de cada variable especificada en la instrucción del gráfico de caja, podemos añadir una expresión booleana entre paréntesis para delimitar la muestra de la variable en cuestión. Hay que insertar un espacio entre el nombre o el número de la variable y la expresión. Supongamos que tenemos datos de salarios de hombres y mujeres y tenemos la variable ficticia GÉNERO con el valor 1 para hombres y 0 para mujeres. En este caso, podemos dibujar gráficos de caja comparativos con la siguiente línea en el cuadro de diálogo:

```
salario (GÉNERO=1) salario (GÉNERO=0)
```

Capítulo 7. Construcción de iteraciones

Simulaciones Monte Carlo

Gretl ofrece un soporte (limitado) para realizar simulaciones de Monte Carlo. Para hacer este tipo de trabajos se puede utilizar o bien el programa GUI en modo “lote de instrucciones” (véase [la sección de nombre *Lotes de instrucciones* en Capítulo 3](#) arriba), o bien el cliente de línea de instrucciones. La instrucción `loop` (bucle) abre el programa en un modo especial, en el cual se acepta la repetición de una instrucción el número de veces indicado. Sólo 7 instrucciones pueden ser utilizadas dentro de dicho bucle: `genr`, `ols`, `print`, `sim`, `smp1`, `store` y `summary`. `genr` y `ols` permiten hacer bastantes operaciones. Tecleando la instrucción `endloop` se sale de este modo: esto hace que se ejecute el lote de instrucciones. Los bucles no pueden estar anidados.

La instrucción `ols` produce un resultado especial dentro del modo bucle; no se muestran los resultados de cada regresión individual y, en su lugar, el programa muestra (a) el valor medio de cada coeficiente estimado en cada una de las iteraciones, (b) la desviación típica de estos coeficientes estimados, (c) el valor medio de la desviación típica estimada de cada coeficiente, y (d) la desviación típica de las desviaciones típicas estimadas. Todos estos resultados cobran sentido solamente si se introduce algún efecto aleatorio.

La instrucción `print` también se comporta de manera diferente en el modo bucle. Muestra la media y la desviación típica de la variable a lo largo de todas las repeticiones del bucle. Está pensada para ser utilizada con variables que tengan un sólo valor en cada iteración, por ejemplo la suma de cuadrados de los errores de una regresión.

La instrucción `store` (a utilizar una única vez en cada bucle) escribe los valores de las variables especificadas en cada una de las iteraciones, en el archivo indicado por el usuario. Por lo tanto, mantiene un historial completo de las variables. Es posible leer y analizar este archivo de datos dentro del programa.

En el [Ejemplo 7-1](#) se muestra un sencillo ejemplo de simulación de Monte Carlo.

Ejemplo 7-1. Una simulación sencilla de Monte Carlo

```
(* Crear un conjunto de datos vacío de tamaño 50 *)
nulldata 50
genr x = uniform()
(* abrir un bucle que se repite 100 veces *)
loop 100
  genr u = normal()
  (* construir la variable dependiente *)
  genr y = 10*x + 20*u
  (* ejecutar una regresión MCO *)
  ols y const x
  (* recuperar el R-cuadrado de la regresión *)
  genr r2 = $rsq
  (* hacer que se muestren las estadísticas sobre los R-cuadrados calculados *)
  print r2
  (* recuperar las estimaciones de los coeficientes individuales *)
  genr a = coeff(const)
  genr b = coeff(x)
  (* y guardarlas en un fichero *)
  store foo.gdt a b
endloop
```

Este bucle mostrará los estadísticos de resumen correspondientes a las estimaciones de ‘a’ y ‘b’ en cada una de las 100 iteraciones, así como los valores R^2 de las 100 regresiones. Después de ejecutar el bucle, el archivo `foo.gdt`, que contiene las estimaciones de los coeficientes individuales de todas las veces que ha sido ejecutado el bucle, puede abrirse desde `gretl` para examinar la distribución de frecuencias de las estimaciones con más detalle. Por favor, téngase en cuenta que aunque se permiten líneas de comentarios dentro de una replicación (como muestra el ejemplo), estos no pueden exceder una línea.

La instrucción `nulldata` es útil para realizar trabajos de Monte Carlo. En lugar de abrir unos datos “reales”, `nulldata 50` (por ejemplo) abre un conjunto de datos vacío, con solamente una constante, y un tamaño muestral igual a 50. Después pueden añadirse variables mediante la instrucción `genr`.

Véase la instrucción `seed` en el [Capítulo 9](#) para información sobre cómo generar series pseudo-aleatorias repetibles.

Mínimos cuadrados iterativos

Se ha diseñado una segunda forma de estructurar las replicaciones, principalmente para calcular mínimos cuadrados iterativos. Greene (2000, c. 11) muestra cómo puede usarse este método para estimar modelos no lineales.

Para comenzar este tipo de replicaciones, hay que especificar una *condición* en lugar de un número incondicional de veces a iterar. Esta condición debe tener la forma de la palabra clave `while` (mientras) seguida por una desigualdad: el término de la izquierda debe ser el nombre de la variable ya definida; el término de la derecha puede ser una constante numérica o el nombre de otra variable predefinida. Por ejemplo,

```
loop while essdiff > .00001
```

Las instrucciones serán ejecutadas dentro del bucle (es decir, hasta que se encuentre `endloop`) mientras se cumpla la condición que estamos evaluando.

El programa supone que si se especifica una replicación tipo “número de veces”, probablemente estemos realizando un análisis de Monte Carlo, y, por lo tanto, no nos interesan los resultados de cada iteración individual, sino más bien los momentos de ciertas variables en el conjunto de las iteraciones. Por otra parte, si se especifica una replicación de tipo “while” probablemente se esté haciendo algo como mínimos cuadrados iterativos y por lo tanto nos gustaría visualizar el resultado final - y también, quizá, los valores de alguna(s) variable(s) (por ejemplo, la suma de cuadrados de los residuos) de cada iteración dentro del bucle. El comportamiento de las instrucciones `print` y `ols` se acomoda a estas suposiciones. En una bucle “while”, `print` se comporta como siempre; así que se imprimen la(s) variable(s) especificada(s) en cada iteración. La instrucción `ols` imprime los resultados de la estimación final.

El [Ejemplo 7-2](#) utiliza un bucle “while” para reproducir la estimación de una función de consumo no lineal de la forma $C = \alpha + \beta Y^\gamma + \epsilon$ como en Greene (2000, Ejemplo 11.3). Este lote de instrucciones se incluye en la distribución de `gretl` bajo el nombre de `greene11_3.inp`; se puede encontrar en `gretl` bajo la opción del menú “Archivo, Abrir archivo de instrucciones, archivo de ejercicios, Greene...”.

Ejemplo 7-2. Función de consumo no lineal

```
open greene11_3.gdt
(* ejecutar MCO inicialmente *)
ols C 0 Y
genr essbak = $ess
genr essdiff = 1
genr b0 = coeff(Y)
genr gamma0 = 1
(* formar las variables linealizadas *)
genr C0 = C + gamma0 * b0 * Y^gamma0 * log(Y)
genr x1 = Y^gamma0
genr x2 = b0 * Y^gamma0 * log(Y)
(* iterar los MCO hasta que la suma de cuadrados converja *)
loop while essdiff > .00001
  ols C0 0 x1 x2 -o
  genr b0 = coeff(x1)
  genr gamma0 = coeff(x2)
  genr C0 = C + gamma0 * b0 * Y^gamma0 * log(Y)
  genr x1 = Y^gamma0 genr x2 = b0 * Y^gamma0 * log(Y)
  genr ess = $ess genr
```

```
    essdiff = abs(ess - essbak)/essbak
    genr essbak = ess
endloop
(* mostrar las estimaciones de los parámetros usando sus "propios nombres" *)
genr alpha = coeff(0)
genr beta = coeff(x1)
genr gamma = coeff(x2)
print alpha beta gamma
```

Bucle con índice

El tercer modo de construir un bucle en gretl es hacer un bucle con índice, utilizando la variable interna *i*. Es necesario especificar los valores inicial y final para *i*, que aumenta en cada iteración del bucle. La sintaxis es la siguiente: `loop i=1..20`. El [Ejemplo 7-3](#) muestra el uso de esta construcción. Considérese el caso de un conjunto de datos de panel, consistente en observaciones sobre varios hospitales entre los años 1991-2000. Restringimos la muestra para cada uno de estos años sucesivamente y obtenemos los estadísticos de resumen de sección cruzada para las variables 1 a 4.

Ejemplo 7-3. Ejemplo de un bucle indexado

```
open hospitals.gdt
loop for i=1991..2000
    smpl -r (year=i)
    summary 1 2 3 4
endloop
```

Capítulo 8. Opciones, argumentos y directorios

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 [Capítulo 4](#) arriba). El programa intenta detectar el formato del archivo y procede en consecuencia. Véase también [la sección de nombre Directorios](#) 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 [la sección de nombre Directorios](#) 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 [la sección de nombre Bases de datos binarias en Capítulo 4](#)) 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 (véase el [Apéndice D](#)).

§ 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.

§ 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

1. En 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`.

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 [Capítulo 4](#)). 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.

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 [Capítulo 4](#)). El programa intentará detectar el formato del archivo y procederá en consecuencia. Véase también [la sección de nombre Directorios](#) 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 [la sección de nombre Directorios](#) 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 [la sección de nombre Directorios](#) 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. Por compatibilidad hacia atrás con el programa original ESL de Ramanathan se ofrece otro mecanismo (ESL no tenía la instrucción `open`). Una línea de la siguiente forma:

```
(* ! mi_archivo.gdt *)
```

hará que (a) `gretlcli` cargue `mi_archivo.gdt`, pero será (b) ignorado como un comentario por el programa ESL original. Nótese la especificación con cuidado: hay exactamente un espacio entre la marca de comentario inicial, `*`, y el símbolo `!`; hay justo un espacio entre el `!` y el nombre del archivo de datos.

Un truco más permite a `gretl` y `gretlcli` obtener información sobre los “archivos de ejercicios prácticos” de ESL incluidos con el paquete `gretl`. Un archivo de prácticas típico empieza así:

(* PS4.1, usando el archivo de datos DATA4-1, para reproducir la Tabla 4.2 *)

El algoritmo utilizado es el siguiente: si una línea de entrada empieza con la marca de comentario, se busca la cadena DATA (en mayúsculas). Si se encuentra, se extrae la cadena desde la D hasta el siguiente espacio o coma, se cambia a minúsculas y se trata como si fuera el nombre del archivo de datos que queremos abrir.

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 (véase [la sección de nombre `gretl`](#) y [la sección de nombre `gretlcli`](#)), 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 8-1](#) 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 8-1](#) para los valores dados por defecto.
5. En el caso de archivos de datos, la búsqueda procede entonces hacia todos los subdirectorios por debajo del directorio principal de datos.

Tabla 8-1. Directorios por defecto

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

Nota: PREFIX denota el directorio base elegido en el momento en que fue instalado `gretl`.

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_CLASSES_ROOT` (el directorio principal de `gretl` y la instrucción para invocar `gnuplot`) y `HKEY_CURRENT_USER` (las demás variables configurables).

Capítulo 9. Guía de Instrucciones

Introducción

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*, *h1u*, *corc*, *ar*, *arch*, *hsk*, *ts1s*, *w1s*, *hccm*, *add* y *omit*.

Las referencias a secciones y capítulos en lo que sigue se refieren al libro de Ramanathan (2002).

Instrucciones de gretl

add

(añadir)

Argumento: *lista_var* [-o]

Ejemplos: `add 5 7 9`

`add xx yy zz -o`

Debe de invocarse después de la instrucción de estimación. Las variables en *lista_var* se añaden al modelo que había previamente y se estima el nuevo modelo. Si se añade más de una variable, se mostrará el estadístico *F* para las variables añadidas (sólo para el método MCO) junto con su valor *p*. Un valor *p* de menos de 0.05 significa que los coeficientes son conjuntamente significativos al nivel del 5 por ciento. Podemos recuperar algunas variables internas a través de la instrucción *genr*, invocando *genr* directamente después de esta instrucción. La marca -o hace que se muestre la matriz de varianzas y covarianzas.

addto

(añadir a)

Argumentos: *ID_de_modelo lista_var*

Ejemplo: `addto 2 5 7 9`

Funciona como la instrucción *add*, con la diferencia de que hay que indicar un modelo previo (utilizando su número de identificación, que se muestra 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 con números 5, 7 y 9 al modelo 2.

adf

(Dickey-Fuller)

Argumentos: *orden nombre_var*

Ejemplo: `adf 2 x1`

Calcula los estadísticos para dos contrastes de Dickey-Fuller. En ambos casos, la hipótesis nula es que la variable en cuestión presenta una raíz unitaria. El primero es un contraste t basado en el modelo

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

La hipótesis nula es que $g = 0$. El segundo contraste (aumentado) se hace estimando una regresión sin restricciones (con una constante, tendencia, el primer retardo de la variable, y un número igual a *orden* de retardos de la primera diferencia como los regresores), y una versión restringida (omitiendo la tendencia 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 en el modelo sin restricciones, y los subíndices u y r significan los modelos sin restricciones y restringido respectivamente. Nótese que los valores críticos para estos estadísticos no son los habituales; se muestra un rango para el valor p , cuando puede determinarse.

ar

Argumentos: *retardos ; vardep varindeps [-o]*

Ejemplo: `ar 1 3 4 ; y 0 x1 x2 x3`

Calcula las estimaciones de los parámetros utilizando el procedimiento iterativo generalizado de Cochrane-Orcutt (véase Sección 9.5 del libro de Ramanathan). Las iteraciones se dan por finalizadas cuando las sucesivas sumas de cuadrados de los errores no difieren en más de un 0.005 por ciento, o después de 20 iteraciones. *retardos* es una lista de retardos de los residuos, y termina con un punto y coma. En el ejemplo de arriba, el término de error se especifica como

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

vardep es la variable dependiente y *varindeps* es la lista de las variables independientes. Poner la marca *-o* hace que se muestre la matriz de covarianzas de los coeficientes de la regresión. Los residuos de la regresión transformada se guardan con el nombre de *uhat*, y se pueden recuperar mediante *genr*. Otras variables internas pueden recuperarse utilizando la instrucción *genr*, invocando *genr* después de esta instrucción.

arch

Argumentos: *orden vardep varindep [-o]*

Ejemplo: `arch 4 y 0 x1 x2 x3`

Esta instrucción contrasta la posibilidad de ARCH (Autoregressive Conditional Heteroskedasticity - Heterocedasticidad Condicional Autorregresiva) en el modelo, del orden de retardos especificado en *orden*, que tiene que ser un número entero. Si el estadístico de contraste LM tiene un valor p menor que 0.10, entonces también se lleva a cabo la esti-

mación ARCH. Si la varianza estimada de alguna observación en la regresión auxiliar no es positiva, entonces se utiliza el cuadrado del residuo correspondiente en su lugar. En este caso, se hace la estimación mínimo-cuadrática ponderada sobre el modelo original. La marca -o muestra la matriz de covarianzas de los coeficientes.

chow

Argumento: *observación*

Ejemplo: `chow 25`
`chow 1988.1`

Tiene que ejecutarse después de una regresión MCO. Crea una variable ficticia que es igual a 1 desde el punto de corte especificado por *observación* 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 la no restringida y la original como la restringida. Este estadístico es apropiado para hacer un contraste de la hipótesis nula de que no existe cambio estructural en el punto de ruptura especificado.

coint

Argumentos: *orden vardep varindeps*

Ejemplos: `coint 2 y x`
`coint 4 y x1 x2`

Lleva a cabo los contrastes de Dickey-Fuller Aumentados de la hipótesis nula de que cada una de las variables incluidas en la lista tiene una raíz unitaria, utilizando el orden de retardos especificado. Se estima la regresión cointegrante y se hace un contraste ADF sobre los residuos que resultan de esta regresión. También se proporciona el estadístico de Durbin-Watson para la regresión cointegrante. Téngase en cuenta que ninguno de estos estadísticos de contraste pueden compararse con las tablas estadísticas habituales.

corc

Argumentos: *vardep varindeps [-o]*

Ejemplos: `corc 1 0 2 4 6 7`
`corc -o 1 0 2 4 6 7`
`corc y 0 x1 x2 x3`
`corc -o y 0 x1 x2 x3`

Calcula las estimaciones de los parámetros utilizando el procedimiento iterativo de Cochrane-Orcutt (véase Sección 9.4 en el libro de Ramanathan) con *depvar* como variable dependiente y *varindeps* como lista de variables independientes. Las iteraciones se dan por terminadas cuando estimaciones sucesivas de los coeficientes de autocorrelación muestran diferencias menores que 0.001, o después de 20 iteraciones. Si se indica la marca -o, se muestra la matriz de covarianzas de los coeficientes de la regresión. Los residuos de esta regresión transformada se guardan con el nombre de *uhat*. Se pueden recuperar varias variables internas utilizando la instrucción *genr*, siempre que invoquemos *genr* inmediatamente después de esta instrucción.

corr

Argumento: *[lista_var]*

Ejemplos: `corr 1 3 5`

`corr y x1 x2 x3`

Muestra los coeficientes de correlación de las variables de la lista *lista_var*, o de todas las variables en el conjunto de datos si no se especifica *lista_var*.

corrgm

Argumentos: `variable [maxret]`

Muestra los valores de la función de autocorrelación de la *variable* especificada (bien por su nombre o por su número). Véase Ramanathan, Sección 11.7. Calcula, por lo 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 muestran las correlaciones parciales: estas son el resultado neto después de descontar los efectos de los retardos intermedios. La instrucción también dibuja el gráfico del correlograma y proporciona el estadístico Q de Ljung-Box para contrastar la hipótesis nula de que la serie es “ruido blanco”. Este estadístico se distribuye asintóticamente como una chi-cuadrado con los grados de libertad iguales al número de retardos utilizados.

Si se aporta el valor *maxret* en forma de un número entero (opcional), la longitud del correlograma se limita, como máximo, a este número de retardos, de otro modo, la longitud se determinará de manera automática.

criteria

Argumentos: `scr T k`

Ejemplo: `criteria 23.45 45 8`

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

cusum

Debe ejecutarse después de la estimación de un modelo mediante MCO. Realiza el contraste CUSUM de estabilidad de los parámetros. Se obtiene una serie de errores de predicción un-paso-adelante (tipificados) mediante la ejecución de una serie de regresiones: la primera regresión utiliza las primeras k observaciones 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 del modelo original). Se realiza el gráfico y se muestra la suma acumulada de los errores de predicción tipificados. La hipótesis nula de estabilidad de los parámetros se rechaza al nivel de significación del 5 por ciento si la suma acumulada sale fuera de la banda de confianza del 95 por ciento.

Además, esta instrucción muestra el estadístico t de Harvey-Collier para comprobar la hipótesis nula de estabilidad de parámetros. Para más detalles, consultar el capítulo 7 del libro *Econometric Analysis* de Greene.

delete

(borrar)

Elimina la última variable (de número más alto) en el conjunto de datos actual. *Utilizar con cuidado*: no se pide ninguna confirmación. Puede ser útil para quitar variables ficticias temporales. No hay posibilidad de eliminar otra variable que no sea la última.

diff

Argumento: *lista_var*

Se obtiene la primera diferencia de cada variable de la lista *lista_var* y el resultado se guarda en una nueva variable con el prefijo *d_*. De este modo *diff x* y crea las variables nuevas $d_x = x(t) - x(t-1)$ y $d_y = y(t) - y(t-1)$.

endloop

(fin de bucle)

Da por finalizadas las iteraciones de un bucle. Véase *loop*.

eqnprint

Debe ejecutarse despues de la estimación de un modelo mediante MCO. Se guardan las estimaciones en formato de ecuación LaTeX, en un archivo cuyo nombre tiene la estructura *equation_N.tex*, donde *N* es el número de modelos estimados hasta el momento en la sesión actual. Dicho archivo puede incorporarse en un documento LaTeX. Para información adicional, véase *tabprint*.

fcast

Argumentos: *[primobs ultobs] nuevo_nombre_var*

Ejemplos: *fcast 1997.1 1999.4 f1*

fcast f2

Debe de ejecutarse despues de una instrucción de estimación. Las predicciones se generan para el rango especificado (o el rango más grande posible si no hemos puesto los argumentos *primobs* y *ultobs*) y los valores se guardan como *nuevo_nombre_var*, que así pueden mostrarse o representarse en un gráfico. Las variables del lado derecho de la ecuación son las del modelo original. No hay posibilidad de sustituirlas por otras variables. Si se ha especificado un término de error autorregresivo (en *hi lu*, *corc*, y *ar*) la predicción es condicional un-paso-adelante, e incorpora el proceso de error.

fcasterr

Argumentos: *primobs ultobs [-o]*

Después de estimar un modelo mediante MCO que incluya una constante y al menos una variable independiente (estas restricciones pueden relajarse en un algún punto), se puede utilizar esta instrucción para mostrar los valores ajustados sobre el rango de observación especificado, junto con las desviaciones típicas estimadas de estas predicciones y los intervalos de 95 por ciento de confianza. Además, si se utiliza la opción *-o*, los resultados también se mostrarán en un gráfico *gnuplot*. Se utiliza el método de regresión aumentada de Salkever (1976) para generar los errores de predicción estándar.

fit

(ajuste)

Esta orden *fit* (que debe seguir a una instrucción de estimación) es un atajo a la instrucción *fcast*. Esta instrucción genera los valores ajustados, para la muestra actual, basándose en la última regresión y los guarda en una serie denominada "autofit". En el caso de los modelos de series temporales, *fit* también muestra un gráfico *gnuplot* de los valores actuales y estimados de la variable dependiente a lo largo del tiempo.

freq

Argumento: *freq nombre_var (o numero_var)*

Muestra la distribución de frecuencias de *nombre_var* o *numero_var* junto con un contraste chi-cuadrado de normalidad. En el modo interactivo, se muestra un gráfico gnuplot de la distribución.

genr

Argumento: *nueva_var = formula*

Crea variables nuevas, normalmente mediante transformaciones de variables ya existentes. Para más información sobre los posibles atajos, véase, *diff*, *logs*, *lags*, *ldiff*, *multiply* y *square*.

El programa admite los siguientes *operadores aritméticos*, en orden de precedencia: \wedge (exponenciación); $*$, $/$ y $\%$ (módulo o resto); $+$ y $-$.

Los *operadores Booleanos* son (de nuevo en orden de precedencia): $!$ (negación), $\&$ (Y lógica), $|$ (O lógica), $>$, $<$, $=$, $>=$ (mayor o igual a), $<=$ (menor o igual a) y \neq (no igual a). Es posible utilizar los operadores Booleanos para definir variables ficticias: por ejemplo, $(x > 10)$ produce 1 si $x > 10$, y 0 en los demás casos. Las *funciones* permitidas pertenecen a los siguientes grupos:

§ Funciones matemáticas típicas: *abs*, *cos*, *exp*, *int* (parte entera del número), *ln* (logaritmo natural: *log* es un sinónimo), *sin* (seno), *sqrt* (raíz cuadrada)

§ Funciones estadísticas: *max* (valor máximo de una serie), *min* (mínimo), *mean* (media aritmética), *median* (mediana), *var* (varianza), *sd* (desviación típica), *sum* (suma), *cov* (covarianza), *corr* (coeficiente de correlación), *pvalue* (valor p).

§ Funciones de series temporales: *lag* (retardo), *lead* (avance), *diff* (primera diferencia), *ldiff* (diferencia logarítmica, o primera diferencia de los logaritmos naturales).

§ Otros: *cum* (acumulación), *sort* (ordenar), *uniform* (distribución uniforme), *normal* (distribución normal), *misszero* (reemplazar el código de observación ausente en una serie con ceros), *zeromiss* (la operación inversa de *misszero*).

Todas las funciones citadas arriba a excepción de *cov*, *corr*, *pvalue*, *uniform* y *normal* toman como único argumento, o bien el nombre de la variable (nótese que, en una instrucción *genr*, no es posible referirse a las variables por su número de identificación) o bien una expresión compuesta que se evalúa como una variable (por ejemplo, $\ln((x1+x2)/2)$). Tanto *cov* como *corr* requieren dos argumentos, y dan respectivamente la covarianza y el coeficiente de correlación de las dos variables nombradas. La función *pvalue* toma los mismos argumentos que la instrucción *pvalue* (véase abajo), pero en este contexto hay que poner comas entre los argumentos. *uniform()* y *normal()*, que no toman argumentos, devuelven series pseudoaleatorias extraídas respectivamente de las distribuciones uniforme (0-100) [Nota del traductor: (0-1) a partir de gretl-1.0.9] y normal reducida (véase también la instrucción *seed*). Las series uniformes se generan utilizando la función de biblioteca *C rand()*; para las series normales se emplea el método de Box y Muller (1958).

Además de los operadores y funciones comentados aquí, hay algunos usos especiales de *genr*:

§ *genr time* crea una variable de tendencia temporal (1,2,3,...) denominada *time*.

genr; index hace lo mismo excepto que la variable se denomina *index*.

§ `genr dummy` crea variables ficticias hasta la periodicidad de los datos. Por ejemplo, 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 otros trimestres, y así sucesivamente.

§ `genr paneldum` crea un conjunto de variables ficticias especiales para usar con datos de panel - véase el [Capítulo 5](#) arriba.

§ Es posible recuperar varias de las variables internas definidas mientras se ejecuta una regresión, utilizando `genr` como se describe a continuación:

<code>\$ess</code>	suma de cuadrados de los residuos
<code>\$rsq</code>	R -cuadrado no ajustado
<code>\$T</code>	número de observaciones utilizadas en el modelo
<code>\$df</code>	grados de libertad
<code>\$trsq</code>	TR -cuadrado (tamaño muestral multiplicado por R -cuadrado)
<code>\$sigma</code>	desviación típica de los residuos
<code>\$lnl</code>	log de la verosimilitud (en modelos logit y probit)
<code>coeff(var)</code>	coeficiente estimado para la variable <i>var</i>
<code>stderr(var)</code>	desviación típica estimada para la variable <i>var</i>
<code>rho(i)</code>	coeficiente autoregresivo de orden i -ésimo de los residuos
<code>vcv(xi,xj)</code>	covarianza entre los coeficientes de las variables x_i y x_j

Nota: En el programa cliente de línea de instrucciones, las instrucciones `genr` que recuperan datos relacionados con los modelos siempre se refieren al modelo estimado más recientemente. Esto mismo ocurre en el cliente GUI, si se utiliza `genr` en la “consola de gretl” o se proporciona una formula a través de la opción “Definir nueva variable” en el menú Variable en la ventana principal. Sin embargo, con el GUI existe la opción de recuperar los datos de cualquier modelo que actualmente esté mostrándose en la ventana (independientemente de que sea el modelo más reciente o no). Esto se consigue desde el menú “Datos del modelo” en la ventana del modelo.

Hay, además, otras tres variables “internas” disponibles: `$nobs` contiene el número de observaciones del rango de muestra actual (nótese que este puede o no ser equivalente al valor de `$T`, el cual se refiere al número de observaciones utilizadas en la estimación del modelo más reciente). La variable `t` sirve como un índice de las observaciones. Así por ejemplo, `genr dum = (t=15)` generará una variable ficticia que tiene el valor 1 para la observación número 15 y 0 en el resto. La variable `$pd` contiene la frecuencia o la periodicidad de los datos (por ejemplo, 4 para datos trimestrales).

La [Tabla 9-1](#) ofrece varios ejemplos de los diferentes usos de `genr` con comentarios de explicación; a continuación hay un par de sugerencias a tener en cuenta sobre el uso de variables ficticias:

§ Supongamos que `x` se codifica con los valores 1, 2, ó 3 y `Vd` desea tres variables ficticias, `d1 = 1` si `x = 1`, 0 si no, `d2 = 1` si `x = 2`, y así sucesivamente. Para crear estas variables, utilice las siguientes instrucciones:

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

§ Para obtener la serie $z = \max(x, y)$ haga

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

Tabla 9-1. Ejemplos de uso de la instrucción **genr**

Instrucción	Comentario
<code>genr y = x1^3</code>	x1 elevado al cubo
<code>genr y = ln((x1+x2)/x3)</code>	
<code>genr z = x>y</code>	asigna a $z(t)$ el valor 1 si $x(t) > y(t)$ y 0 en caso contrario
<code>genr y = x(-2)</code>	x retardada 2 periodos
<code>genr y = x(2)</code>	x adelantada 2 periodos
<code>genr y = diff(x)</code>	$y(t) = x(t) - x(t-1)$
<code>genr y = ldiff(x)</code>	$y(t) = \log x(t) - \log x(t-1)$, la tasa de crecimiento instantánea de x
<code>genr y = sort(x)</code>	ordena x en orden creciente y lo guarda en y
<code>genr y = -sort(-x)</code>	ordena x en orden decreciente
<code>genr y = int(x)</code>	trunca x y guarda su valor entero en y
<code>genr y = abs(x)</code>	guarda los valores absolutos de x
<code>genr y = sum(x)</code>	suma los valores de x excluyendo los valores perdidos -999
<code>genr y = cum(x)</code>	acumula: $y_t = \sum_{\tau=1}^t x_\tau$
<code>genr aa = \$ess</code>	define aa como la suma de cuadrados de los residuos de la última regresión
<code>genr x = coeff(sqft)</code>	guarda en x el coeficiente estimado de la variable <i>sqft</i> obtenido en el último modelo
<code>genr rho4 = rho(4)</code>	guarda en rho4 el coeficiente autorregresivo de cuarto orden obtenido del último modelo (supone un modelo ar)
<code>genr cvx1x2 = vcv(x1, x2)</code>	guarda la covarianza de los coeficientes estimados de las vars x1 y x2 en el último modelo
<code>genr foo = uniform()/100</code>	variable pseudoaleatoria uniforme, de rango 0 a 1
<code>genr bar = 3 * normal()</code>	variable pseudoaleatoria normal, $\mu = 0$, $\sigma = 3$

gnuplot

Argumentos: `vars_y vars_x [-o | -m]`
`gnuplot -z var_y var_x ficticia`

En el primer caso, se representan las variables *vars_y* con respecto a las variables *vars_x*. Con la opción `-o`, el gráfico utilizará líneas; con la opción `-m`, el gráfico utilizará impulsos (líneas verticales); en los demás casos, se trazará con puntos.

En el segundo caso, se representa *var_y* con respecto a *var_x* con los puntos en diferentes colores dependiendo de si el valor de la variable *ficticia* es 1 ó 0.

Para realizar un gráfico de series temporales, se puede pedir `gnuplot vars_y time`. Si no existe previamente ninguna variable llamada *time*, entonces ésta se generará de forma automática. Se crearán variables ficticias especiales para representar datos mensuales y trimestrales.

En modo interactivo, el resultado se pasa a `gnuplot` para que lo muestre en pantalla. En

el modo de proceso por lotes, se escribe un archivo de gráfico llamado `gpttmp01.plt`. (Se crean archivos similares con el uso posterior de `gnuplot`, y se incrementa el número en el nombre del archivo para reflejar esto.) Se pueden volver a generar los gráficos más adelante utilizando la instrucción `gnuplot gpttmp.plt`. (En MS Windows, invocar `wgnuplot` y abrir el archivo `gpttmp01.plt`.) Si se quieren controlar los detalles del gráfico, se puede editar el archivo `.plt`.

graph

Argumentos: `var1 var2 [-o]`
`graph var1 var2 var3`

Gráficos ASCII. En los dos primeros ejemplos, se representa la variable `var1` (que puede darse mediante un nombre o un número) contra la variable `var2` (eje x) utilizando símbolos ASCII. La opción `-o` trazará el gráfico con 40 filas y 60 columnas, sin ella el gráfico será de 20 por 60 (para ver los resultados en pantalla). En el tercer ejemplo, ambas variables `var1` y `var2` se representarán (sobre el eje y) contra `var3`. Esto es especialmente útil para representar los valores observados y predichos contra el tiempo. Véase también la instrucción `gnuplot`.

hausman

Este contraste sólo está disponible después de haber estimado un modelo mediante la instrucción `pooled` (véanse también `panel` y `setobs`). Se contrasta el modelo combinado simple contra las principales alternativas, es decir, contra los modelos de efectos fijos o de efectos aleatorios.

En el modelo de efectos fijos se añade una variable ficticia para todas menos una de las unidades de sección cruzada, permitiendo de esta manera que el intercepto de la regresión varíe a través de las unidades. Se presenta un contraste F de significación conjunta de estas variables ficticias. En el modelo de efectos aleatorios se descompone la varianza residual en dos partes, una parte específica de la unidad de sección cruzada y la otra específica de la observación particular. (Sólo se puede calcular este estimador si el número de las unidades de sección cruzada en el conjunto de datos excede el número de parámetros a estimar). El estadístico de LM de Breusch-Pagan sirve para contrastar la hipótesis nula (de que el estimador MCO de datos combinados es adecuado) contra la alternativa del modelo de efectos aleatorios.

El modelo de MCO combinados puede ser rechazado contra ambas alternativas, es decir, la de efectos fijos y la de efectos aleatorios. Si el error específico de grupo o de unidad no está correlacionado 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 resulta ser inconsistente, y es preferible el estimador de efectos fijos. La hipótesis nula para el contraste de Hausman indica que el error específico al grupo no está correlacionado (y por lo tanto es preferible el modelo de efectos aleatorios). Un valor p bajo para este contraste lleva a rechazar el modelo de efectos aleatorios en favor del modelo de efectos fijos.

hccm

Argumentos: `vardep varindeps [-o]`

Presenta las estimaciones de MCO junto con las desviaciones típicas de los coeficientes obtenidas por medio de una estimación consistente ante heterocedasticidad de la matriz de varianzas y covarianzas. Utiliza para ello el método “jackknife” de MacKinnon y White (1985) (véase Ramanathan, Sección 8.3). Si ponemos la marca `-o` se muestra la matriz de covarianzas de los coeficientes.

help

Proporciona una lista de las instrucciones disponibles en gretl. `help` instrucción describe la *instrucción* (por ejemplo `help smpl`). Se puede teclear `man` en lugar de `help` si así se desea.

hilu

Argumentos: `vardep varindeps [-o]`

Ejemplos: `hilu 1 0 2 4 6 7`

`hilu -o y 0 x1 x2 x3`

Calcula las estimaciones de un modelo utilizando el procedimiento de red de búsqueda de Hildreth-Lu (se hace el ajuste fino con el procedimiento Cochrane-Orcutt) siendo *vardep* la variable dependiente y *varindeps* una lista con las variables independientes separadas por espacios. Se representa graficamente la suma de cuadrados de los residuos del modelo transformado contra los valores de rho desde -0.99 hasta 0.99. Si se utiliza la opción `-o`, también se mostrará la matriz de varianzas y covarianzas de los coeficientes de las regresiones. Los residuos de esta regresión transformada se guardan bajo el nombre de `uhat`.

hsk

Argumentos: `vardep varindeps [-o]`

Calcula las estimaciones corregidas de heterocedasticidad (véase Ramanathan, cap. 8) y sus estadísticos asociados. Se ajusta una regresión auxiliar para el logaritmo de los cuadrados de los residuos (utilizando los cuadrados de las variables independientes pero no sus productos cruzados), a partir de la cual se obtienen los estimadores de mínimos cuadrados ponderados del modelo inicial. Si se usa la opción `-o`, se mostrará también la matriz de varianzas y covarianzas estimada de los coeficientes de la regresión. Es posible recuperar varias de las variables internas mediante la instrucción `genr`, para ello ha de invocarse `genr` inmediatamente después de esta instrucción.

if

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

```
if condición
    instrucciones
else
    instrucciones
endif
```

condición tiene que ser una expresión Booleana; para ver su sintaxis, véase [la sección de nombre *genr*](#). El bloque `else` es opcional; los bloques `if ... endif` pueden estar anidados.

import

Argumento: `nombre_de_archivo [-o]`

Sin la opción `-o`, importa datos desde un archivo con formato CSV (comma-separated values - valores separados por comas), como por ejemplo los que se pueden escribir fácilmente desde cualquier programa de hoja de cálculo. Este archivo debe de tener los nom-

bres de las variables en la primera línea, y una matriz de datos rectangular en las líneas restantes. Las variables deben estar alineadas “por observación” (una columna por cada variable; cada fila representa una observación). Véase el [Capítulo 4](#) para más detalles.

Con la opción `-o`, lee archivos de datos en formato BOX1, como los que se obtienen utilizando el servicio de extracción de datos de la Oficina del Censo de los Estados Unidos (US Bureau of the Census).

info

`info` muestra cualquier información contenida en el archivo de cabecera que corresponda al archivo de datos actual. (Esta información debe estar limitada entre `(*` y `*)`, estando situados estos marcadores en líneas separadas.)

labels

Muestra las etiquetas informativas de cualquiera de las variables generadas utilizando `genr`, así como de cualquier etiqueta añadida al conjunto de datos mediante el GUI.

lad

Argumentos: `vardep varindeps`

Calcula una regresión que minimiza la suma de las desviaciones absolutas entre los valores observados y ajustados de la variable dependiente. Las estimaciones de los coeficientes se obtienen utilizando el algoritmo simplex de Barrodale-Roberts; si la solución obtenida no es única aparece un mensaje de aviso. Las desviaciones típicas se obtienen mediante el procedimiento “bootstrap” con 500 extracciones.

lags

Argumento: `lista_var`

Crea variables nuevas que son los valores retardados de cada una de las variables en “`lista_var`”. El número de variables retardadas que se crean es igual a la periodicidad. Por ejemplo, si la periodicidad fuera 4 (datos trimestrales), la instrucción `lags x` y creará `x_1 = x(t-1)`, `x_2 = x(t-2)`, `x_3 = x(t-3)` y `x_4 = x(t-4)`; y de forma similar para `y`. Estas variables deben referenciarse de forma exacta, esto es, con el carácter de subrayado.

ldiff

Argumento: `lista_var`

Se obtiene la primera diferencia del logaritmo natural de cada variable de “`lista_var`” y el resultado se guarda en una variable nueva con el prefijo `ld_`. Por lo tanto `ldiff x` y crea las nuevas variables $ld_x = \ln(x_t) - \ln(x_{t-1})$ y $ld_y = \ln(y_t) - \ln(y_{t-1})$.

lntest

Debe utilizarse justo después de una instrucción `ols`. Calcula el estadístico de contraste del multiplicador de Lagrange (LM) (y los valores `p` asociados) para los contrastes de no linealidad y de heterocedasticidad (contraste de White) o, si se utiliza la opción `-o`, el estadístico de contraste LMF para correlación serial hasta la periodicidad (véase Kiviet, 1986). También se muestran los coeficientes de las regresiones auxiliares correspondientes. Véanse los Capítulos 7, 8, y 9 del libro de Ramanathan para más detalles. Solamente se utilizan los cuadrados de las variables independientes, y no sus productos cruzados. (No se obtienen los estadísticos de contraste LM si al crearse internamente los cuadrados de las variables independientes se aprecia multicolinealidad exacta.)

logit

Argumentos: *vardep varindeps*

Regresión logit binomial. La variable dependiente ha de ser una variable binaria. Se calculan las estimaciones máximo-verosímiles de los coeficientes para las *varindeps* mediante el método EM (Expectation-Maximization method) (véase Ruud, 2000, ch. 27). Como el modelo no es lineal, las pendientes dependen de los valores de las variables independientes: las pendientes mencionadas se evalúan en las medias de estas variables. El estadístico de contraste chi-cuadrado contrasta la hipótesis nula de que todos los coeficientes son cero a excepción de la constante.

Si queremos utilizar logit para llevar a cabo un análisis de proporciones (donde la variable dependiente es la proporción de casos con alguna característica en particular, en cada observación, en lugar de una variable 1 o 0 indicando si la característica esta presente o ausente) no debemos utilizar la instrucción `logit`, sino construir la variable logit (por ejemplo, mediante `genr lgt_p = log[p/(1 - p)]`) y utilizar esta como la variable dependiente en una regresión MCO. Véase Ramanathan, cap. 12.

logs

Argumento: *lista_var*

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

loop

(bucle)

Uso: `loop número_de_veces`
`loop while condición`
`loop for i= principio.. final`

Ejemplos: `loop 1000`
`loop while difscr > .00001`
`loop for i=1991..2000`

Esta instrucción da acceso a un modo especial, en el que el programa acepta instrucciones a repetir, bien un número especificado de veces, o bien mientras se satisfaga alguna condición, o bien para valores sucesivos de la variable índice (interna) *i*. Dentro de un bucle, sólo se pueden utilizar siete instrucciones: `genr`, `ols`, `print`, `sim`, `smpl`, `store` y `summary` (`store` no puede utilizarse en un bucle de “while”). Se sale del modo de introducción de órdenes de bucle con la instrucción `endloop`: en este punto, se ejecutarán las instrucciones acumuladas hasta entonces. Los bucles contruidos mediante “loop” no pueden estar anidados. Para conocer más detalles, véase el [Capítulo 7](#).

meantest

(contraste de medias)

Argumentos: *var1 var2 [-o]*

Calcula el estadístico *t* para contrastar la hipótesis nula de que las medias poblacionales son iguales, para las variables *var1* y *var2* y muestra su valor *p*. Sin la opción `-o`, el estadístico se calcula partiendo del supuesto de que las varianzas son iguales para las dos variables; con la opción `-o`, se supone que las varianzas son distintas. (La opción sólo es relevante si hay un número diferente de observaciones no perdidas para las dos variables.)

mpols

(MCO de precisión múltiple)

Argumentos: *vardep varindeps*Ejemplos: `mpols 1 0 2 4 6 7`
`mpols y 0 x1 x2 x3`

Calcula los estimadores de mínimo cuadrados ordinarios (MCO) con *vardep* como variable dependiente y *varindeps* como lista de variables independientes, utilizando aritmética de punto flotante y precisión múltiple. Las variables pueden especificarse por su nombre o su número; utilice el número cero para indicar un término constante. Esta instrucción sólo está disponible si gretl ha sido compilado con soporte GMP, la biblioteca GNU de precisión múltiple.

Para estimar un ajuste polinomial, utilizando aritmética de precisión múltiple para generar las potencias requeridas de la variable independiente, se utiliza, por ejemplo, `mpols y 0 x ; 2 3 4`. Esto realiza una regresión de *y* sobre *x*, *x* al cuadrado, *x* al cubo y *x* elevado a la cuarta potencia. En otras palabras, los números a la derecha del punto y coma (que han de ser números enteros y positivos), especifican las potencias de *x* a utilizar. Si se especifica más de una variable independiente, la última variable antes del punto y coma es la que será elevada a las potencias que se indican.

multiplyArgumentos: *x sufijo vars*Ejemplos: `multiply invpop pc 3 4 5 6`
`multiply 1000 big x1 x2 x3`

las variables de la lista *vars* (referenciadas por nombre o número) se multiplican por *x*, que puede ser un valor numérico o el nombre de una variable previamente definida. Los productos se nombran con el *sufijo* especificado (máximo 3 caracteres). Si hace falta, se truncan primero los nombres de las variables originales. Por ejemplo, si se desea crear las versiones per capita de ciertas variables y existe ya la variable *pob* (población), una serie de instrucciones apropiadas sería: `genr invpop = 1/pob; multiply invpob pc renta gasto` lo cual creará *rentapc* como el producto de *renta* y *invpob*, y *gastopc* como *gasto* multiplicado por *invpob*.

noecho

Suprime el eco normal al introducir las instrucciones y comentarios (excepto los introducidos utilizando la instrucción `print`) a la hora de ejecutar un lote de instrucciones.

nulldataArgumento: *tamaño_serie*Ejemplo: `nulldata 100`

Establece un conjunto de datos “vacío”, que contiene sólo una constante, con periodicidad 1 y el número de observaciones especificado. Esto puede usarse para hacer simulaciones: algunas de las instrucciones `genr` (por ejemplo, `genr uniform()`, `genr normal()`, `genr time`) generarán datos ficticios desde cero para rellenar el conjunto de datos. Este orden puede resultar útil al utilizarla conjuntamente con `loop`. Véase también la instrucción `seed`.

ols

(Mínimos cuadrados ordinarios - MCO)

Argumentos: *vardep varindeps* [-o]

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

Calcula las estimaciones de mínimos cuadrados ordinarios (MCO) con *vardep* como variable dependiente y *varindeps* como lista de variables independientes. La opción -o hace que se muestre la matriz de covarianzas de los coeficientes de regresión. Las variables pueden especificarse bien por su nombre, o por su número; se utiliza el número cero para indicar un término constante. El programa también muestra el valor p para los estadísticos *t* (a dos colas) y *F*. Un valor p menor que 0,01 indica significación al nivel del 1 por ciento y se marca con ***. ** indica significatividad a niveles entre el 1 y el 5 por ciento y * indica significatividad a niveles entre el 5 y el 10 por ciento. También se presentan algunos estadísticos de selección de modelos (descritos en el libro de Ramanathan, Sección 4.3). Varias de las variables internas pueden recuperarse mediante la instrucción *genr*, siempre que *genr* se invoque justo después de la instrucción *ols*.

omit

(excluir)

Argumento: *lista_var* [-o]

Ejemplo: `omit 5 7 9`

Debe ejecutarse después de una instrucción de estimación. Las variables en *lista_var* se excluyen del modelo anterior y se estimará el nuevo modelo. Si se omite más de una variable, se muestra el estadístico *F* de Wald junto con su valor p (sólo para el método MCO). Un valor p menor que 0,05 indica que los coeficientes son conjuntamente significativos al nivel del 5 por ciento. Algunas variables internas pueden recuperarse mediante la instrucción *genr*, siempre que *genr* se invoque justo después de esta instrucción. Si se indica la opción -o se mostrará la matriz de covarianzas de los coeficientes.

omitfrom

Argumentos: *ID_de_modelo lista_var*

Ejemplo: `omitfrom 2 5 7 9`

Funciona de modo similar a la instrucción *omit*, excepto que hay que especificar un modelo previo (mediante su número de identificación, que aparece al comienzo de los resultados del modelo), para tomarlo como base para excluir las variables. En el ejemplo de arriba se omiten las variables con números 5, 7 y 9 del Modelo 2.

open

(abrir)

Argumento: *archivo_de_datos*

Abre un archivo de datos. Si ya hay un archivo de datos abierto, este será reemplazado por el nuevo. El programa intentará detectar el formato del archivo nuevo (propio, CSV o BOX1) y lo tratará como corresponda.

pergm

Argumento: *nombre_de_variable* [-o]

Calcula y muestra (y en modo interactivo, representa) el espectro de la variable especifica-

da. Sin la opción `-o`, se obtiene el periodograma muestral; con la opción `-o`, se utiliza una ventana de retardos de Bartlett de tamaño $2\sqrt{T}$ (donde T es el tamaño de la muestra) para estimar el espectro (véase el capítulo 18 del libro de Greene, *Econometric Analysis*). Al mostrar el periodograma muestral, también se proporciona un contraste t de integración fraccional de la serie (“de memoria larga”): la hipótesis nula es que el orden de integración de la serie es cero.

plot

(gráfico en modo texto, de baja calidad)

Ejemplos: `plot x1`
 `plot x1 x2`
 `plot -o x1 x2`

Representa los valores de los datos para las variables especificadas, para el rango de observaciones actualmente en efecto utilizando símbolos ASCII. Cada línea se refiere a una observación y los valores se representan horizontalmente. Con la opción `-o`, `x1` y `x2` se representan en la misma escala, si no, la escala de cada una se ajusta por separado de manera apropiada. La opción `-o` sólo debe emplearse si las variables tienen aproximadamente sus valores dentro del mismo rango (por ejemplo, la variable dependiente observada y su estimación). Véase también la orden `gnuplot`.

pooled

(MCO combinados)

Argumentos: `vardep varindeps [-o]`

Estima un modelo mediante MCO (véase `ols` para detalles sobre su sintaxis), y lo marca como un modelo combinado o de panel. Por lo tanto, hace que la opción de contraste `hausman` se muestre disponible.

print

(mostrar, presentar en la pantalla)

Argumento: `[lista_var] [-o | -t]` ó `[cadena]`

Ejemplos: `print`
 `print x y`
 `print 1 2 3 -o`

Si se ha indicado una `lista_var`, muestra en la pantalla los valores de las variables especificadas; si no se ha indicado ninguna lista, se muestran los valores de todas las variables que hay en el archivo de datos actual. Con la opción `-o`, los datos se muestran por observaciones, y si no, por variables. Con la opción `-t`, los datos se muestran con 10 valores significativos.

Si el argumento de `print` es una cadena de texto literal (que tiene que empezar con comillas dobles, `"`), se imprime la cadena tal cual.

probit

Argumentos: `vardep varindeps`

Regresión probit. La variable dependiente ha de ser una variable binaria. Se obtienen los estimadores de máxima verosimilitud de los coeficientes de las `varindeps` mediante míni-

mos cuadrados iterativos (el método EM o expectativa-maximización). Como el modelo es no lineal, las pendientes dependen de los valores de las variables independientes: las pendientes se evalúan en las medias de estas variables. El estadístico de contraste chi-cuadrado contrasta la hipótesis nula de que todos los coeficientes, excepto el término constante, son cero.

El probit para el análisis proporcional no está implementado en gretl actualmente.

pvalue

(valor p)

Uso:

```
pvalue 1 valor_x (distribución normal)
pvalue 2 gl valor_x (distribución t)
pvalue 3 gl valor_x (distribución chi-cuadrado)
pvalue 4 gln gld valor_x (distribución F)
pvalue 5 media varianza valor_x (distribución Gamma)
```

Calcula el área a la derecha del *valor_x* en la distribución especificada. *gl* son los grados de libertad, *gln* son los grados de libertad del numerador, *gld* son los grados de libertad del denominador. En vez de los dígitos se puede utilizar z, t, X, F y G para las distribuciones normal, *t*, chi-cuadrado, distribución, *F*, y gamma respectivamente.

quit

(salir)

Salir de gretl, ofrece la posibilidad de guardar los resultados de la sesión antes de cerrar (*q* es un atajo; *x* sale sin preguntar si se deben guardar los resultados).

reset

Debe utilizarse inmediatamente después de la estimación de un modelo mediante MCO. Realiza el contraste de especificación de modelos (no linealidad) RESET de Ramsey, añadiendo el cuadrado y el cubo de los valores ajustados a la regresión, y calculando el estadístico *F* para la hipótesis nula de que los parámetros de los dos términos añadidos son cero.

rhodiff

Argumentos: *lista_rho ; lista_var*

Ejemplos: *rhodiff .65 ; 2 3 4*

```
rhodiff r1 r2 ; x1 x2 x3
```

Crea las transformaciones rho-diferenciadas de las variables en *lista_var* (identificadas por su nombre o por su número) y las añade al conjunto de datos, utilizando el sufijo # para las nuevas variables. Dada la variable *v1* en *lista_var*, y las entradas *r1* y *r2* en *lista_rho*, se crea *v1#* = $v1(t) - r1 \cdot v1(t-1) - r2 \cdot v1(t-2)$. Las entradas de *lista_rho* pueden darse como valores numéricos, o mediante los nombres de variables previamente definidas.

rmplot

(gráfico rango-media)

Argumento: *nombre_de_variable*

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 un tamaño arbitrario k . La primera submuestra se forma mediante $y(1),...,y(k)$, la segunda, $y(k+1), ..., y(2k)$, y así sucesivamente. Para cada submuestra, calculamos la media de la muestra y su rango (= máximo-mínimo), y construimos un gráfico con las medias sobre el eje horizontal, y los rangos en el vertical. De esta forma cada submuestra se representa por un punto en el plano. Si la varianza de la serie es constante, esperaríamos que el rango de las submuestras fuera independiente de la media de las submuestras; por tanto, si vemos que los puntos se aproximan a una recta con pendiente positiva, esto sugiere que la varianza de la serie se incrementa con su media; y si los puntos se aproximan a una recta con pendiente negativa, esto parece indicar que la varianza está decreciendo con su media.

Aparte del gráfico, `gretl` muestra las medias y los rangos de cada submuestra, el coeficiente de la pendiente de la regresión MCO de los rangos sobre las medias y el valor p de la hipótesis nula de que esta pendiente es cero. Si el coeficiente de la pendiente es significativo al nivel de significación del 10 por ciento, entonces el gráfico también muestra la recta de regresión estimada de los rangos sobre las medias.

run

(ejecutar)

Argumento: *archivo_de_entrada*

Ejecuta las instrucciones del *archivo_de_entrada* (archivo de instrucciones) y devuelve el control al cursor interactivo.

runs

Argumento: *nombre_de_variable*

Realiza el "contraste de rachas" (no paramétrico) sobre aleatoriedad de la variable especificada. Si se quiere hacer un contraste para la aleatoriedad de las desviaciones respecto a la mediana de una variable llamada $x1$ con una mediana distinta de cero, se puede hacer de la siguiente forma:

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

scatters

(gráficos bivariantes)

Argumentos: *vary ; lista_varx*
lista_vary ; varx

Ejemplos: `scatters 1 ; 2 3 4 5`
`scatters 1 2 3 4 5 6 ; time`

Representa (mediante `gnuplot`) gráficos bivariantes (scatterplots) por parejas de variables *vary* con respecto a todas las variables en *lista_varx*, o de todas las variables en *lista_vary* contra *varx*. El primer ejemplo sitúa la variable 1 en el eje y y se dibujan cuatro gráficos, el primero con la variable dos en el eje x , el segundo con la variable 3 en el eje x , y así sucesivamente. El segundo ejemplo traza cada una de las variables, desde 1 hasta 6,

a lo largo del tiempo. Un paso útil en el análisis exploratorio de datos consiste en inspeccionar un conjunto de gráficos de este tipo. No se admiten más de 6 gráficos; cualquier variable extra en la lista será ignorada.

seed

(semilla)

Argumento: *integer*

Establece la semilla para el generador de números pseudo-aleatorios para la distribución uniforme [*uniform()*] y normal (*normal()*) (véase la instrucción *genr*). Al iniciar el programa se establece la semilla por defecto utilizando el reloj del sistema. Si se desea obtener secuencias de números pseudo-aleatorios repetibles, hay que establecer la semilla de forma manual.

setobs

Argumentos: *setobs periodicidad primobs*

Ejemplos: *setobs 4 1990.1*
setobs 12 1978.03
setobs 20 1.01

Obliga al programa a interpretar el conjunto de datos actual como una serie temporal o de panel, cuando los datos han sido leídos inicialmente como una serie simple sin fechas. La *periodicidad* tiene que ser un número entero; *primobs* es una cadena que representa la fecha o identificación de panel de la primera observación. Véase también el [Capítulo 5](#).

setmiss

Argumentos: *valor [lista_var]*

Ejemplos: *setmiss -1*
setmiss 100 x2

Hace que el programa interprete un valor numérico especificado (el primer parámetro de la instrucción) como código de “valor perdido” cuando se importan datos. Si este valor es el único parámetro, como en el primer ejemplo de arriba, esta interpretación también se aplicará a todas las series en el conjunto de datos. Si el *valor* va seguido de una lista de variables, definidas por su nombre o por su número, la interpretación se limitará a la(s) variable(s) especificadas. Por lo tanto, en el segundo ejemplo, el valor 100 será interpretado como un código para definir “valor perdido”, pero sólo para la variable *x2*.

shell

Usage: *! shellcommand*

Un *!* al comienzo de una línea de instrucción se interpreta como un escape al sistema operativo (al “shell” del usuario). De este modo, se pueden ejecutar instrucciones del shell desde dentro de *gretl* (no disponible en la versión para MS Windows).

sim

Argumentos: *primobs ultobs y a0 a1 a2*

Ejemplos: *sim 1979.2 1983.1 y 0 0.9* crea $y(t) = 0.9 \cdot y(t-1)$
sim 15 25 y 10 0.8 x crea $y(t) = 10 + 0.8 \cdot y(t-1) + x(t) \cdot y(t-2)$

Simula valores para y para los periodos desde *primobs* hasta *ultobs*. La variable y tiene que haber sido definida con anterioridad con los valores iniciales apropiados. La fórmula empleada es $y(t) = a_0(t) + a_1(t)*y(t-1) + a_2(t)*y(t-2) + \dots$. Los términos $a_i(t)$ pueden ser constantes numéricas o nombres de variables definidas previamente.

En el siguiente ejemplo se simula un proceso autorregresivo de orden uno con coeficiente 0.8 $y(t) = 0.8*y(t-1) + u(t)$ para $t = 2, \dots, 100$ siendo $u(t)$ normal (0,1):

```
genr y = const*0
genr coeff = const*0.8
genr u = normal()
sim 2 100 y u coeff
```

smp1

(muestra)

Argumentos: *primobs ultobs*
smp1 -o var_ficticia
smp1 -o
smp1 -r condición

Cambia el rango de la muestra. En el primer modo, *primobs* y *ultobs* tienen que ser consistentes con la periodicidad de los datos. En el segundo modo, *var_ficticia* tiene que ser una variable indicadora con valores 0 o 1 en cada observación; la muestra se restringirá a las observaciones cuyo valor sea 1. El tercer modo, *smp1 -o*, excluye todas las observaciones para las cuales los valores de una o más variables estén “perdidos”. El cuarto modo (*-r*) restringe la muestra a las observaciones que cumplan con la condición (Booleana) impuesta.

<i>smp1 3 10</i>	datos con periodicidad 1
<i>smp1 1950 1990</i>	datos anuales con periodicidad 1
<i>smp1 1960.2 1982.4</i>	datos trimestrales
<i>smp1 1960.04 1985.10</i>	datos mensuales
<i>smp1 1960.2 ;</i>	mantiene <i>ultobs</i> sin cambios
<i>smp1 ; 1984.3</i>	mantiene <i>primobs</i> sin cambios
<i>smp1 -o dum1</i>	crea una muestra de las observaciones donde <i>dum1=1</i>
<i>smp1 -r renta > 30000</i>	restringe la muestra a los casos en los que la variable <i>renta</i> tenga un valor mayor que 30000.

Conviene tomar nota sobre las formas *-o* y *-r* del *smp1*: Cualquier información “estructural” en el fichero de cabecera de los datos (referente a la naturaleza de los datos, es decir, que sean series temporales o de panel) se pierde al ejecutar esta orden. Se puede reimponer de nuevo la estructura con la instrucción *setobs*.

spearman

Argumentos: x y [*-o*]

Calcula el coeficiente de correlación por rangos de Spearman para las dos variables, x e y . No es preciso previamente ordenar las variables y asignar los rangos de forma manual; la instrucción se encarga de hacer esto. Con la opción *-o*, se muestran los datos originales junto con los ordenados.

La ordenación automática se hace de mayor a menor (es decir, al dato mayor se le asigna rango 1). Si *Vd* necesita invertir ese orden, puede crear una nueva variable cuyos valores sean los de la variable original cambiados de signo. Por ejemplo:

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

square

(cuadrados)

Argumento: `lista_var [-o]`

Genera nuevas variables correspondientes a los cuadrados y a los productos cruzados de las variables en `lista_var` (-o creará los productos cruzados). Por ejemplo, `square x y` generará `sq_x = x` cuadrado, `sq_y = y` cuadrado y `x_y = x` por `y`. Si hay alguna variable en particular que es ficticia, esta no será elevada al cuadrado ya que el resultado sería la misma variable.

store

(guardar)

Argumento: `nombre_fichero [lista_var] [flag]`

`nombre_fichero` es el nombre del archivo donde se guardarán los valores.

Si falta `lista_var`, se guardan los valores de todas las series de datos en el conjunto de datos actual. Nótese que las variables escalares no se guardan de manera automática: si deseamos guardar las escalares, tenemos que incluirlas explícitamente en `lista_var`.

Por defecto, se guardarán en el formato XML nativo de `gretl`. Hay seis *opciones* validas (excluyentes entre sí):

- z El formato por defecto, pero comprimido con gzip.
- o los datos se guardan en formato binario utilizando precisión doble.
- s los datos se guardan en formato binario utilizando precisión simple.
- c los datos se guardan en formato CSV (comma-separated values — valores separados por comas). Los programas de hojas de cálculo pueden leer este formato directamente.
- r los datos se guardan en formato GNU R.
- m los datos se guardan en formato GNU Octave.
- t los datos se guardan en formato ESL, “tradicional” de `gretl` con un archivo de datos ASCII y otro fichero separado de cabecera.

summary

Argumento: `[lista_var]`

Muestra los estadísticos principales de las variables incluidas en `lista_var` o, si se omite `lista_var`, de todas las variables incluidas en el conjunto de datos. Los resultados que se proporcionan son la media, la desviación típica (sd), el coeficiente de variación (= dt/media), la mediana, el mínimo, el máximo, el coeficiente de asimetría y el exceso de curtosis.

tabprint

Debe de ejecutarse despues de la estimación de un modelo mediante MCO. Guarda el modelo estimado en forma de “entorno tabular” de LaTeX, en un archivo con el nombre `model_N.tex`, donde N es el número de modelos estimados hasta el momento en la sesión

actual. Este archivo puede incorporarse a un documento LaTeX. Véase también `eqnprint`.

testuhat

Debe seguir a una instrucción de estimación de modelos. Muestra la distribución de frecuencias de los residuos del modelo junto con el contraste chi-cuadrado de normalidad.

tsls

(Mínimos cuadrados en dos etapas)

Argumentos: `vardep lista_var1 ; lista_var2 [-o opcional]`

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

Calcula las estimaciones de los parámetros por el método de mínimos cuadrados en dos etapas (MC2E). *vardep* es la variable dependiente, *lista_var1* es la lista de las variables independientes (incluyendo las variables endógenas) en la ecuación estructural para las cuales se necesitan las estimaciones de MC2E. *lista_var2* es la lista conjunta de las variables predeterminadas y exógenas en todas las ecuaciones. Si *lista_var2* no es, por lo menos, tan larga como *lista_var1*, el modelo no está identificado. La opción `-o` muestra la matriz de covarianzas de los coeficientes. En el ejemplo de arriba, las *ys* son las variables endógenas y las *xs* son las variables exógenas y predeterminadas. Algunas variables internas pueden recuperarse mediante la instrucción `genr`, siempre que `genr` se invoque justo después de esta instrucción.

var

Argumentos: `orden vardep varindep`

Ejemplos: `var 4 x1 const time x2 x3`
`var 3 1 0 2 3 4`

Calcula (vía MCO) las estimaciones de una autorregresión vectorial. El primer argumento especifica el orden de los retardos, luego sigue la especificación de la primera ecuación, de igual forma que en la instrucción `ols`. Hay que tener cuidado de no incluir los retardos entre los elementos de la lista *varindep* - estos se añaden de forma automática. Se ejecutará una regresión para cada variable de la lista, excluyendo la constante, la tendencia temporal y cualquier variable ficticia. Los resultados de cada ecuación incluyen los contrastes *F* para restricciones cero en todos los retardos de cada una de las variables, y un contraste *F* para el retardo máximo.

varlist

Muestra una lista de las variables disponibles en la actualidad. `list` y `ls` son sinónimos.

vartest

Argumentos: `var1 var2`

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

wls

Argumentos: `varpesos vardep varindeps [-o opcional]`

Se obtienen las estimaciones de mínimos cuadrados ponderados siendo *varpesos* la variable de ponderaciones, *vardep* la variable dependiente y *varindeps* la lista de variables

independientes. Más concretamente, se ejecuta una regresión MCO de *varpesos* * *vardep* contra *t* * *varindeps*. Si *varpesos* es una variable ficticia, esto equivale a la eliminación de todas las observaciones que tengan valor cero para *varpesos*. Con la opción *-o* se proporciona la matriz de covarianzas de los coeficientes. Mediante la instrucción *genr* pueden recuperarse algunas variables internas, siempre que *genr* sea invocada inmediatamente después de la instrucción *wls*.

Estimadores y contrastes: resumen

La [Tabla 9-2](#) muestra los estimadores disponibles bajo el menú de “Modelo” en la ventana principal de gretl. La instrucción de consola correspondiente (en caso de que haya una) se muestra entre paréntesis. Para más detalles consúltese el apartado sobre la instrucción en el [Capítulo 9](#).

Tabla 9-2. Estimadores

Estimador	Comentario
Mínimos Cuadrados Ordinarios	El estimador MCO básico
Mínimos Cuadrados Ponderados	Heterocedasticidad, exclusión de observaciones seleccionadas
HCCM	MCO con matriz de covarianzas corregida de heterocedasticidad
Corregida de heterocedasticidad	Mínimos Cuadrados Ponderados basados en la varianza del error de predicción
Cochrane-Orcutt	Autocorrelación de primer orden
Hildreth-Lu	Autocorrelación de primer orden
Estimación autorregresiva	Autocorrelación de orden superior (Cochrane-Orcutt generalizado)
Autorregresión vectorial (VAR)	Sistemas de ecuaciones de series temporales
Contraste de cointegración	Relaciones de largo plazo entre series
Mínimos cuadrados en dos etapas	Ecuaciones simultáneas
Logit	Variable dependiente binaria (distribución logística)
Probit	Variable dependiente binaria (distribución normal)
Mínima desviación absoluta	Alternativa a Mínimos Cuadrados
Correlación por rangos	Correlación con datos ordinales
MCO combinados	Estimación MCO para datos combinados de sección cruzada y de series temporales
MCO con precisión múltiple	Estimación MCO usando aritmética de precisión múltiple

La [Tabla 9-3](#) muestra los contrastes disponibles en el menú de contrastes en una ventana de modelo, después de la estimación.

Tabla 9-3. Contrastes para los modelos

contraste	Instrucción correspondiente
Omitir variables (contraste F si es MCO)	<code>omit</code>
Añadir variables (contraste F si es MCO)	<code>add</code>
No linealidad (cuadrados)	<code>lmtest</code>
No linealidad (logaritmos)	<code>lmtest</code>
No linealidad (el contraste RESET de Ramsey)	<code>reset</code>
Heterocedasticidad (contraste de White)	<code>lmtest</code>
Autocorrelación hasta la frecuencia de los datos	<code>lmtest -o</code>
Chow (cambio estructural)	<code>chow</code>
CUSUM (estabilidad de los parámetros)	<code>cusum</code>
ARCH (heterocedasticidad condicional)	<code>arch</code>
Normalidad de los residuos	<code>testuhat</code>
Diagnósticos de panel	<code>hausman</code>

Capítulo 10. Solución de problemas en gretl

Informes de errores

En el camino hacia una versión “estable” de gretl (versión 1.0), la colaboración de los usuarios resulta muy útil, y por lo tanto, agradezco cualquier información sobre posibles errores encontrados en el programa. Creo que será difícil que haya errores en los cálculos llevados a cabo por gretl (aunque con esto no quiero garantizar nada). Pero es posible que haya errores en el comportamiento del interfaz gráfico. En este contexto, por favor, recuérdese que la utilidad de los informes de errores aumenta en función de detalles concretos: ¿qué ocurrió *exactamente*, bajo qué condiciones, y en qué sistema operativo? ¿Si había un mensaje de error, qué decía exactamente? (Aunque no es necesario incluir los números de direcciones de memoria mostrados en los informe de errores de MS Windows - no significarán nada para mí.)

Programas auxiliares

Como se ha mencionado, gretl llama a otros programas para realizar ciertas tareas (gnuplot para los gráficos, LaTeX para un formateo de alta calidad de los resultados de las regresiones, GNU R). Si algo no va bien con estas conexiones externas, la tarea de producir una ventana de mensaje con información sobre los errores no siempre resulta fácil. Si falla alguna de estas conexiones cuando se accede desde el interfaz gráfico de gretl, es posible que encontremos más información si recomenzamos gretl desde el cursor de instrucciones (por ejemplo, desde un xterm en el sistema de ventanas X, o una “ventana del MSDOS” en MS Windows, tecleando gretlw32.exe), en vez de hacerlo desde el menú del escritorio o un icono. Puede que se muestren mensajes adicionales de errores en la ventana del terminal.

Además, ténganse en cuenta que para la mayoría de las llamadas externas, gretl supone que los programas en cuestión están disponibles en nuestro “directorio” - es decir, pueden invocarse simplemente mediante el nombre del programa, sin la necesidad de tener que teclear la dirección entera.¹ Así, si falla alguno de los programas disponibles, es conveniente experimentar poniendo el nombre del programa en la línea de instrucciones, como se muestra a continuación.

Sistema	Gráficos	Composición	GNU R
Sistema de ventanas X	gnuplot	latex, xdvi	R
MS Windows	wgnuplot.exe	latex, windvi	RGui.exe

Si el programa no consigue empezar desde el cursor, entonces el problema no está relacionado con gretl, sino que el directorio del programa no está en nuestra senda de directorios, o que el programa no está instalado (correctamente). Para más detalles sobre cómo modificar la ubicación de directorios, consúltase la ayuda *en línea* o la documentación para su sistema operativo (shell).

1. La excepción a esta regla es la invocación de gnuplot en MS Windows, donde consta la dirección completa del programa.

Capítulo 11. El interfaz de línea de instrucciones

Gretl en la consola

El paquete `gretl` incluye el programa de línea de instrucciones `gretlcli`. Este es, esencialmente, una versión actualizada del programa ESL de Ramu Ramanathan. En Linux, puede ejecutarse desde la consola, o en un `xterm` (o similar). En MS Windows, puede ejecutarse desde una “ventana de MSDOS”. `Gretlcli` cuenta con su propio archivo de ayuda, al que puede accederse tecleando la palabra “help” desde el cursor. Es posible ejecutarlo en un proceso por lotes, y enviar los resultados directamente a un archivo (véase [la sección de nombre `gretlcli` en Capítulo 8](#) arriba).

Si `gretlcli` está conectado a la biblioteca “readline” (esto ocurre automáticamente en el caso de la versión MS Windows; véase también el [Apéndice B](#)), la línea de instrucciones es recuperable y editable, y ofrece terminación de las instrucciones. Podemos utilizar las flechas de Arriba y Abajo para pasar a las instrucciones que ya han sido tecleados previamente. En cualquier línea de instrucciones, podemos utilizar las flechas para movernos, junto con las combinaciones de teclas para editar de Emacs.¹ Las más comunes son :

Combinaciones de teclas	Efecto
<code>Ctrl-a</code>	ir al inicio de la línea
<code>Ctrl-e</code>	ir al final de la línea
<code>Ctrl-d</code>	eliminar el carácter de la derecha

donde “`Ctrl-a`” significa pulsar la tecla “a” a la vez que la tecla “`Ctrl`”. Así, si queremos cambiar algo en el inicio de una instrucción, *no* es preciso ir hacia atrás por toda la línea, borrándola con nuestro paso. Simplemente saltamos al inicio y añadimos o eliminamos caracteres.

Al teclear las primeras letras de una instrucción, pulsando la tecla `Tab` el programa `readline` intentará completar el nombre de la instrucción. Si es una combinación única, saldrá la instrucción automáticamente. Si hay más de una posibilidad, pulsando `Tab` otra vez, se mostrará una lista.

Diferencias con ESL

Los lotes o guiones de instrucciones desarrollados para el programa ESL original de Ramanathan deberían de ser utilizables en `gretlcli` con pocos o ningún cambio: en lo único que hay que tener cuidado es en las instrucciones multilineales y la instrucción `freq`.

§ ESL utiliza el punto y coma como terminación de muchas de las instrucciones. Esto ha sido eliminado en `gretlcli`. El punto y coma simplemente se ignora, excepto en unos casos especiales donde tiene un significado determinante: como separador de dos listas en las instrucciones `ar` y `ts1s`, o como marcador para una primera o última observación sin cambios en la instrucción `smp1`. En ESL, el punto y coma da la posibilidad de partir una instrucción larga de más de una línea; en `gretlcli` esto se hace con un `\` situado al final de la línea que va a continuarse.

§ Con `freq`, actualmente no se pueden especificar rangos definidos por el usuario como en ESL. Un contraste chi-cuadrado para normalidad ha sido añadido a los resultados de esta instrucción.

Nótese que se ha simplificado la sintaxis de línea de instrucciones para un proceso por lotes. En ESL se tecleaba, por ejemplo

```
esl -b datafile < inputfile > outputfile
```

1. En realidad, las combinaciones dadas a continuación son sólo las que hay por defecto; pueden ser personalizadas. Véase el manual de `readline` .

mientras que en `gretlcli` hay que teclear:

```
gretlcli -b inputfile > outputfile
```

El archivo de entrada se trata como un argumento del programa; debe de especificar el archivo de datos que hay que usar de forma interna, utilizando la sintaxis `open fichero_de_datos` o el comentario especial `(* ! datafile *)`

Apéndice A. Comentarios sobre los archivos de datos

Formato nativo básico

En el formato nativo básico de gretl, un conjunto de datos se guarda en formato XML (extensible mark-up language). Los archivos de datos corresponden a la simple DTD (document type definition) contenida en `gretldata.dtd`, que viene con la distribución de gretl y se instala en el directorio de datos del sistema (por ejemplo `/usr/share/gretl/data` en Linux.) Los archivos de datos pueden ser de texto simple o comprimidos con gzip. Contienen los valores de los datos, además de información adicional tal como el nombre y la descripción de las variables, la frecuencia de los datos, etc.

La mayoría de los usuarios no tendrán necesidad de leer o escribir archivos de este tipo, excepto mediante el propio gretl, pero si Vd. tiene la necesidad de manipularlos utilizando otras herramientas de software, tendrá que examinar el DTD y también mirar algunos de los archivos de datos de prácticas que acompañan al programa: `data4-1.gdt` expone un ejemplo sencillo; `data4-10.gdt` es un ejemplo en el que se incluyen etiquetas para las observaciones.

Formato tradicional ESL

Por compatibilidad hacia atrás, gretl también puede manejar archivos de datos en el formato “tradicional” heredado del programa ESL de Ramanathan. En este formato (que era el formato por defecto de gretl en versiones previas a la 0.98) un conjunto de datos se representa mediante dos archivos. Uno contiene los datos en sí, y el otro contiene información sobre cómo leerlos. En otras palabras:

1. *Los datos en sí*: Una matriz rectangular de números separados por espacios en blanco. Cada columna representa una variable, cada fila una observación sobre cada una de las variables (estilo hoja de cálculo). Las columnas de datos pueden estar separadas por espacios o tabuladores. El nombre del archivo debe de llevar el sufijo `.gdt`. Por defecto, el archivo de datos es de tipo ASCII (texto simple). Opcionalmente, puede estar comprimido con gzip para ahorrar espacio en el disco. Se pueden insertar comentarios en un archivo de datos: si una línea empieza por la marca de almohadilla (`#`), se ignora la línea entera. Esto es consistente con los archivos de datos de gnuplot y octave.
2. *Cabecera*: El archivo de datos tiene que venir acompañado por un archivo de cabecera que tiene el mismo nombre raíz que el archivo de datos más el sufijo `.hdr`. Este archivo contiene lo siguiente (en orden):
 - (Opcional) *comentarios* sobre los datos, iniciados con la cadena de texto (`*` y finalizados con la cadena de cierre `*`), cada uno de estos marcadores tiene que tener su propia línea separada.
 - (Obligatoria) lista de *nombres de las variables* del archivo de datos, separadas por espacios en blanco. Los nombres tienen un límite de 8 caracteres, tienen que empezar por una letra, y están restringidos a los caracteres alfanuméricos más el carácter de subrayado (`_`). La lista puede ser de más de una línea; se termina con un punto y coma, `;`.
 - (Obligatoria) línea de *observaciones* con la forma `1 1 85`. El primer elemento nos da la frecuencia de los datos (1 para los datos sin fechas o anuales, 4 para los trimestrales, 12 para los mensuales). Los elementos segundo y tercero dan las observaciones de inicio y final. Generalmente, estas serán el 1 y el número de observaciones respectivamente, para los datos sin fecha. Para las series temporales, se pueden utilizar fechas en la forma `1959.1` (trimestral, 1 dígito después del punto) o `1967.03` (mensual, dos dígitos después del punto). Véase el [Capítulo 5](#) para el uso especial de esta línea en el caso de datos panel.
 - La palabra clave `BYOBS`.

Aquí hay un ejemplo de un archivo de cabecera de datos escrito correctamente.

```
(*
DATA9-6:
Datos sobre log(dinero), log(renta) y tipo de interés de los EEUU.
Fuente: Stock and Watson (1993) Econometrica
(datos sin alisar) El periodo es 1900-1989 (datos anuales).
Datos recogidos por Graham Elliott.
*)
ldinero lrenta tipoint ;
1 1900 1989 BYOBS
```

El archivo de datos correspondiente contiene tres columnas de datos, cada una con 90 entradas.

Hay tres características del formato “tradicional” de datos que hay que tener en cuenta.

1. Si la palabra clave BYOBS se reemplaza por BYVAR y va seguida por la palabra clave BINARY, esto indica que el archivo de datos correspondiente está en formato binario. Se puede escribir este tipo de archivos desde `gretlcli` utilizando la instrucción `store` con la marca `-s` (precisión sencilla) o la marca `-o` (precisión doble).
2. Si BYOBS es seguido por la palabra clave MARKERS, `gretl` espera un archivo de datos en el cual la *primera columna* contiene cadenas (de 8 caracteres como máximo) que se utilizan para la identificación de las observaciones. Esto puede resultar útil en datos de sección cruzada donde las unidades de las observaciones son identificables: países, provincias, ciudades, etc. También puede ser de utilidad para datos de series temporales irregulares, como por ejemplo, los precios diarios de los valores en la bolsa, donde hay unos días de cierre - en este caso las observaciones pueden señalarse con una fecha tipo 10/01/98. (Recuérdese el límite de 8 caracteres.) Nótese que las opciones BINARY y MARKERS son mutuamente exclusivas. Otra consideración a tener en cuenta es que los “marcadores” no son una variable: la columna de marcadores no tiene una entrada correspondiente en la lista de los nombres de las variables en el archivo de cabecera.
3. Si se encuentra otro archivo con el mismo nombre raíz que el archivo de datos y de cabecera, pero con el sufijo `.tbl`, este se lee para rellenar las etiquetas descriptivas para las series de datos. El formato de un archivo de etiquetas es sencillo: cada línea contiene el nombre de una variable (como en el archivo de cabecera), seguido por uno o más espacios, seguido por la etiqueta descriptiva. Un ejemplo sería: precio Índice de precios de automóviles nuevos, año base 1982

Para guardar los datos en formato tradicional, se utiliza la instrucción `store`, con la opción `-t` bien en el programa de línea de instrucciones, o bien en la ventana de consola del cliente GUI.

Bases de datos binarias: detalles

Una base de datos `gretl` contiene dos partes: un archivo índice ASCII (el nombre de archivo lleva el sufijo `.idx`), donde se guarda la información sobre la serie y un archivo binario (sufijo `.bin`) que contiene los datos en sí. Dos ejemplos de posibles formatos de entradas en el archivo `idx` son:

```
GOM910 índice compuesto por 11 indicadores adelantados (1987=100)
M 1948.01 - 1995.11 n = 575
currbal Balanza de pagos: Balanza por cuenta corriente; SA
Q 1960.1 - 1999.4 n = 160
```

El primer campo contiene el nombre de la serie. El segundo es una descripción de la serie (máximo de 128 caracteres). En la segunda línea el primer campo es un código de frecuencias: M para mensual, Q para trimestral, A para anual, B diario laborable (diario con cinco días a la semana) y D para diario (siete días a la semana). No se admite ninguna otra frecuencia de momento. Luego viene la fecha de inicio (N.B. con dos dígitos después del

punto para datos mensuales, uno para datos trimestrales, ninguno para datos anuales), un espacio, un guión, otro espacio, la fecha de finalización, la cadena "n = " y el número de observaciones en números enteros. En el caso de datos diarios, las fechas de inicio y finalización deben de venir en la forma YYYY/MM/DD (año/mes/día/). Este formato tiene que respetarse de manera estricta.

Opcionalmente, la primera línea del archivo de índice puede contener un comentario corto (hasta 64 caracteres) sobre el origen y naturaleza de los datos, iniciado por una marca de almohadilla. Por ejemplo:

```
# Federal Reserve Board (interest rates)
```

El archivo de base de datos binarios correspondiente contiene los valores de los datos, representados como "flotantes", es decir, números de punto flotante de precisión simple, que ocupan, típicamente, cuatro bytes cada uno. Los números se empaquetan "por variable", así que los primeros n números son las observaciones de la variable 1, los siguientes m son las observaciones de la variable 2, y así sucesivamente.

Apéndice B. Comentarios técnicos

Gretl está escrito en el lenguaje de programación C. Se ha intentado, en la medida de lo posible, cumplir con el estándar ISO/ANSI C (C89), aunque el interfaz gráfico para el usuario y algunos otros componentes hacen uso de extensiones específicas de ciertas plataformas.

Gretl sigue desarrollándose en Linux. Es posible recompilar y ejecutar la librería compartida y el cliente de línea de instrucciones en cualquier plataforma que, (a) apoye ISO/ANSI C y (b) tenga instalada las librerías `zlib` (compresión) y `libxml` (manipulación XML). La página web de `zlib` se encuentra en info-zip.org. La dirección de Internet de Libxmls es xmlsoft.org. Si se encuentra la biblioteca `readline` de GNU en el sistema anfitrión, ésta será utilizada por `gretcli`, aportando así una línea de instrucciones editable muy mejorada. Consultar la página web de `readline`.

El programa de cliente gráfico debería de poder recompilarse y ejecutarse en cualquier sistema que, además de los requisitos detallados arriba, ofrezca la versión de GTK 1.2.3 o superior (véase gtk.org). En el momento de escribir este manual, hay dos variantes de las librerías GTK: la serie 1.2 y la serie 2.0, distribuida en el verano de 2002. Estas dos variantes son incompatibles entre sí. Gretl puede compilarse con cualquiera de las dos - el paquete de código fuente incluye dos subdirectorios, `gui` para GTK 1.2 y `gui2` para GTK 2.0. Si está disponible, recomendamos el uso de GTK 2.0, ya que ofrece muchas mejoras sobre GTK 1.2.

Gretl llama a `gnuplot` para hacer los gráficos. Se puede encontrar `gnuplot` en gnuplot.info. En el momento de escribir esto, la versión oficial más reciente es la 3.7.1 (es de Noviembre de 1999). Si acceder a repositorios de CVS y recompilar códigos fuente no les supone ningún problema, recomendamos la instalación de `gnuplot` desde el CVS actual. El autor de este manual actualmente está utilizando `gnuplot` versión 3.8i obtenida desde el CVS.

Algunas opciones de `gretl` utilizan la librería `gtkextra` de Adrian Feguin. `gtkextra` se encuentra en gtkextra.sourceforge.net.

Se encuentra disponible una versión binaria de `gretl` para el sistema operativo Windows de Microsoft (versión de 32 bits, es decir, Windows 95 o superior). Esta versión fue compilada en Linux a través de `mingw` (el compilador C de GNU, `gcc`, trasvasado para su uso con `win32`) y vinculada con la librería C de Microsoft, `msvcrt.dll`. Emplea el “port” de GTK 2.0 de Tor Lillqvist para `win32`. El programa instalador de Windows (gratis, de código abierto) es cortesía de Jordan Russell (jrsoftware.org).

Tengo la esperanza de que `gretl` interese lo suficiente a algunos usuarios con habilidades en códigos y programación, como para mejorarlo y extenderlo. La documentación del `libgretl` API está lejos de ser completa, pero Vd. puede encontrar algunos detalles si sigue la conexión “`Libgretl` API docs” en la página web de `gretl`.

Apéndice C. Precisión numérica

Gretl siempre utiliza aritmética de doble precisión - excepto en el plugin de precisión múltiple invocado con la opción del menú “Modelo/MCO de alta precisión” que representa los valores de punto flotante empleando un número de bits determinados por la variable de entorno GRETL_MP_BITS (valor por defecto 256). Las ecuaciones normales de Mínimos Cuadrados se resuelven mediante la descomposición de Choleski, lo cual es suficientemente preciso para la mayoría de los propósitos. El programa ha sido probado de manera bastante exhaustiva con los conjuntos de datos de referencia estadística aportados por el NIST (el Instituto Nacional de Estándares y Tecnología de los Estados Unidos) y puede encontrarse un informe completo de los resultados en la página web de gretl (pinchar en “Numerical accuracy (Precisión numérica)” y seguir la conexión).

En Octubre de 2002, tuve un útil intercambio de ideas con Giovanni Baioicchi y Walter Disasto, quienes estaban escribiendo una reseña de gretl para el *Journal of Applied Econometrics*, y con James McKinnon, el editor de evaluaciones de software de dicha revista. Quiero expresar mis agradecimientos a Baioicchi y Disasto por sus exhaustivas comprobaciones del programa, las cuales dieron lugar a las siguientes modificaciones.

1. Los evaluadores indicaron que existía un error en el “buscador de valores p” de gretl, por el cual el programa mostraba el complemento de la probabilidad correcta para valores negativos de z. Esto fue solucionado en la versión 0.998 del programa (2002, Julio 9).
2. También señalaron que el buscador de valores p producía resultados incorrectos para valores extremos de x (por ejemplo, valores alrededor de 8 a 10 en la distribución *t* con 100 grados de libertad). Esto fue también corregido en gretl versión 0.998, por medio de un cambio a un código de distribución de probabilidad más preciso.
3. Los evaluadores notaron un defecto en la presentación de los coeficientes de regresión en gretl, por el cual algunos coeficientes podían ser impresos con un número de cifras significativas inaceptablemente pequeño. Esto fue corregido en la versión 0.999 (2002, Agosto 25): ahora todos los estadísticos asociados con una regresión se imprimen con 6 cifras significativas.
4. Por medio de las comprobaciones realizadas por los evaluadores se supo además que la precisión numérica de gretl en MS Windows era menor que en Linux, en la cual había realizado yo mis pruebas. Por ejemplo, en los datos Longley - un bien conocido conjunto de datos “mal comportados” frecuentemente utilizados en la comprobación de programas econométricos - la versión Windows de gretl recogía coeficientes erróneos a partir del 7º dígito mientras que los mismos coeficientes eran correctos en Linux. Esta anomalía fue corregida en la versión 1.0pre3 de gretl (2002, Octubre 10).

La versión actual del código fuente de gretl contiene un subdirectorio `tests`, con una batería de pruebas basadas en los conjuntos de datos del NIST. Esto se invoca mediante `make check` en el directorio raíz. Se recibe un aviso si la precisión numérica cae por debajo de lo estándar. Por favor, consúltese el archivo `README` en el directorio `tests` para más detalles.

La batería de contrastes del NIST se distribuye asimismo con la versión MS Windows de gretl. Pueden ejecutarse los contrastes invocando el programa `nisttest.exe`.

Como ya se ha mencionado, todos los estadísticos de regresión se imprimen con 6 cifras significativas en la actual versión de gretl (excepto cuando se usa el plugin de precisión múltiple, en cuyo caso los resultados se ofrecen con 12 cifras). Si se desea examinar un valor con más precisión, se puede guardar primero (véase [la sección de nombre *genr* en Capítulo 9](#)) y a continuación pedir su presentación mediante la orden `print -t` (véase el [Capítulo 9](#)). De esta manera se mostrará el valor con hasta 10 dígitos.

Apéndice D. Análisis econométrico avanzado con software libre

Como se ha mencionado en el texto principal, gretl ofrece una selección bastante amplia de estimadores de Mínimos Cuadrados, así como algunos estimadores adicionales como logit y probit (binomial) y de Mínima Desviación Absoluta. Sin embargo, para los usuarios avanzados, las rutinas estadísticas de gretl pueden ser algo limitadas.

Sin duda, algunos de estos usuarios preferirán escribir sus propios programas de rutinas estadísticas en algún lenguaje básico de programación como C, C++ o Fortran. Otra opción será utilizar otro lenguaje de alto nivel que ofrezca la posibilidad de manipular matrices y que tenga algunas rutinas estadísticas propias, o también paquetes adicionales que puedan acoplarse al programa. Si esta última opción les resulta atractiva y están interesados en software de código abierto gratuito, les recomiendo estos dos programas: GNU R (r-project.org) o GNU Octave. Estos dos programas son muy similares a los programas comerciales S y Matlab respectivamente.

Otra opción, ya mencionada, es que gretl ofrece la posibilidad de exportar datos tanto en formato Octave como en R. En el caso de Octave, el conjunto de datos gretl se guarda del siguiente modo: la primera variable dispuesta para exportar se trata como la variable dependiente y se guarda como un vector, *y*, mientras que las restantes variables se guardan como una matriz, *X*. Una vez cargado en Octave, podemos desagregar esta matriz *X* si lo deseamos. Consúltese el manual de Octave para los detalles. En cuanto a R, el archivo de datos exportado conserva la estructura de series temporales que tenga en gretl. Las series se guardan como estructuras individuales. Los datos deben importarse mediante la instrucción `source()` en R.

Entre estos dos programas, R es quizá de más interés inmediato para los econométricos ya que ofrece más en términos de rutinas estadísticas (por ejemplo, modelos lineales generalizados, estimación máximo-verosímil, métodos de series temporales). Por eso, gretl tiene una opción que transfiere los datos de manera rápida y cómoda a R. En el menú “Sesión” de gretl se encuentra la opción “Iniciar GNU R”. Esta opción escribe una versión R del conjunto de datos de gretl (`Rdata.tmp`, en el directorio del usuario de gretl) y lo envía a una sesión nueva de R. A continuación, se describen algunos detalles a tener en cuenta.

En primer lugar, los datos se pasan a R escribiendo una versión provisional de `.Rprofile` en el directorio de trabajo actual. (Si existe un archivo de este tipo, R lo indica al iniciar.) En el caso de que ya exista un archivo `.Rprofile` personal, el archivo original se copiará provisionalmente en `.Rprofile.gretltmp`, y será devuelto a su sitio al salir de gretl. (Si alguien encuentra una manera mejor de hacer esto, por favor que me lo comunique.)

En segundo lugar, el modo particular de invocar a R depende de la variable interna de gretl `Rcommand`, cuyo valor puede establecerse en el menú “Archivo, Preferencias”. En MS Windows, la instrucción por defecto es `RGui.exe`. En X, es `R --gui=gnome` si a la hora de compilar se detecta una instalación del escritorio Gnome (gnome.org), o `xterm -e R` si no se encuentra Gnome. Nótese, por favor, que en esta instrucción se procesarán, como máximo, tres elementos separados por espacios; cualquier elemento adicional será ignorado.

Apéndice E. Lista de URLs

A continuación, hay una lista de las direcciones completas (URL) de los sitios web citados en el manual, según orden de apariencia.

Página web de GNU R

<http://www.r-project.org/>

Página web de GNU Octave

<http://www.octave.org/>

Página web de Gretl

<http://gretl.sourceforge.net/>

Oficina del Censo, Servicio de Obtención de Datos (Census Bureau, Data Extraction Service)

<http://www.census.gov/ftp/pub/DES/www/welcome.html>

Página de datos de Gretl

http://ricardo.ecn.wfu.edu/gretl/gretl_data.html

Estima (RATS)

<http://www.estima.com/>

Tabla Mundial de Penn (Penn World Table)

<http://pwt.econ.upenn.edu/>

Página web de Gnuplot

<http://www.gnuplot.info/>

Manual en línea de Gnuplot

<http://ricardo.ecn.wfu.edu/gnuplot.html>

Manual de GNU Readline

<http://cnswww.cns.cwru.edu/~chet/readline/readline.html>

Biblioteca de Precisión Múltiple de GNU

<http://swox.com/gmp/>

Página web de InfoZip

<http://www.info-zip.org/pub/infozip/zlib/>

Página web de Xmlsoft

<http://xmlsoft.org/>

Página web de GNU Readline

<http://cnswww.cns.cwru.edu/~chet/readline/r1top.html>

Página web de GTK+

<http://www.gtk.org/>

Página web de Gtkextra

<http://gtkextra.sourceforge.net/>

Página web de Mingw (gcc para win32)

<http://www.mingw.org/>

Port GTK+ para win32

<http://user.sgi.com/~tml/gimp/win32/>

JRSoftware

<http://www.jrsoftware.org/>

Página web del escritorio Gnome

<http://www.gnome.org/>

Manual de GNU R

<http://cran.r-project.org/doc/manuals/R-intro.pdf>

Bibliografía

- Box, G. E. P. and Muller, M. E. (1958) "A Note on the Generation of Random Normal Deviates", *Annals of Mathematical Statistics*, 29, pp. 610-11.
- Greene, William H. (2000) *Econometric Analysis*, 4th edition, Upper Saddle River, NJ: Prentice-Hall.
- Kiviet, J. F. (1986) "On the Rigour of Some Misspecification Tests for Modelling Dynamic Relationships", *Review of Economic Studies*, 53, pp. 241-261.
- MacKinnon, J. G. and White, H. (1985) "Some Heteroskedasticity-Consistent Covariance Matrix Estimators with Improved Finite Sample Properties", *Journal of Econometrics*, 29, pp. 305-25.
- R Core Development Team (2000) *An Introduction to R*, version 1.1.1,
- Ramanathan, Ramu (2002) *Introductory Econometrics with Applications*, 5th edition, Fort Worth: Harcourt.
- Ruud, Paul A. (2000) *An Introduction to Classical Econometric Theory*, New York and Oxford: Oxford University Press.
- Salkever, D. (1976) "The Use of Dummy Variables to Compute Predictions, Prediction Errors, and Confidence Intervals", *Journal of Econometrics*, 4, pp. 393-7.
- Wooldridge, Jeffrey M. (2002) *Introductory Econometrics, A Modern Approach*, 2nd edition, Mason, Ohio: South-Western.