

# Guía del usuario de Gretl



Gnu Regression, Econometrics and Time-series

Allin Cottrell  
Department of Economics  
Wake Forest university

Riccardo "Jack" Lucchetti  
Dipartimento di Economia  
Università di Ancona

November, 2005

Copyright © 2001-2005 Allin Cottrell

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

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

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

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Características principales . . . . .	1
1.2. Agradecimientos . . . . .	1
1.3. Instalación de los programas . . . . .	2
<b>2. Puesta en marcha</b>	<b>4</b>
2.1. Ejecutar una regresión . . . . .	4
2.2. Resultados de la estimación . . . . .	6
2.3. Menús de la ventana principal . . . . .	7
2.4. La barra de herramientas de gretl . . . . .	10
<b>3. Modos de trabajo</b>	<b>11</b>
3.1. Lotes de instrucciones . . . . .	11
3.2. El concepto de sesión . . . . .	12
<b>4. Fichero de datos</b>	<b>14</b>
4.1. Formato propio . . . . .	14
4.2. Otros formatos de archivos de datos . . . . .	14
4.3. Bases de datos binarias . . . . .	14
4.4. Crear un archivo de datos desde cero . . . . .	15
4.5. Datos ausentes . . . . .	17
<b>5. Funciones especiales en <i>gretl</i></b>	<b>18</b>
5.1. Introducción . . . . .	18
5.2. Filtros de series temporales . . . . .	18
5.3. Remuestreo y <i>bootstrap</i> . . . . .	19
5.4. Valores ausentes . . . . .	20
5.5. Recuperar variables internas . . . . .	21
<b>6. Datos de Panel</b>	<b>23</b>
6.1. Estructura de Panel . . . . .	23
6.2. Variables ficticias . . . . .	24
6.3. Uso de valores retardados con datos de panel . . . . .	24
6.4. Estimación combinada . . . . .	25
6.5. Ilustración: La Tabla Mundial de Penn . . . . .	26
<b>7. Gráficos y Diagramas</b>	<b>27</b>
7.1. Gráficos Gnuplot . . . . .	27
7.2. Gráficos de caja . . . . .	27

Índice general	II
<b>8. Construcción de iteraciones</b>	<b>29</b>
8.1. Simulaciones Monte Carlo . . . . .	29
8.2. Mínimos cuadrados iterativos . . . . .	30
8.3. Bucle con índice . . . . .	30
<b>9. Cointegración y Modelos de Corrección del Error</b>	<b>32</b>
9.1. El contraste de cointegración de Johansen . . . . .	32
<b>10. Solución de problemas en gretl</b>	<b>34</b>
10.1. Informes de errores . . . . .	34
10.2. Programas auxiliares . . . . .	34
<b>11. El interfaz de línea de instrucciones</b>	<b>35</b>
11.1. Gretl en la consola . . . . .	35
11.2. Diferencias con ESL . . . . .	35
<b>A. Comentarios sobre los archivos de datos</b>	<b>37</b>
A.1. Formato nativo básico . . . . .	37
A.2. Formato tradicional ESL . . . . .	37
A.3. Bases de datos binarias: detalles . . . . .	38
<b>B. Comentarios técnicos</b>	<b>40</b>
<b>C. Precisión numérica</b>	<b>41</b>
<b>D. Análisis econométrico avanzado con software libre</b>	<b>43</b>
<b>E. Lista de URLs</b>	<b>44</b>
<b>Bibliografía</b>	<b>45</b>

## Capítulo 1

# Introducción

### 1.1. Características principales

Gretl es un paquete de cálculo econométrico, e incluye una biblioteca 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 a los manuales de Econometría de Ramu Ramanathan, Jeffrey Woolridge y James Stock y Mark Watson 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 click 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, pero también está disponible para MS Windows y Mac OS X y debería 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.

**Sofisticación** Gretl ofrece una completa gama de estimadores de mínimos cuadrados, incluyendo *mínimos cuadrados en dos etapas* y mínimos cuadrados no lineales. También ofrece varios estimadores de máxima verosimilitud específicos (por ejemplo, logit, probit, tobit) y desde la versión 1.5.0, un estimador de máxima verosimilitud general. El programa es capaz de estimar sistemas de ecuaciones simultáneas, GARCH, ARMA, VAR y modelos de corrección de error vectoriales.

**Precisión** Gretl ha sido contrastado con los conjuntos de datos de referencia de NIST, obteniendo muy buenos resultados. 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, Francés, Italiano, Polaco o Alemán dependiendo del idioma nativo del ordenador.

### 1.2. Agradecimientos

El código base de Gretl procedía originariamente del programa ESL (“Econometrics Software Library”), escrito por el profesor Ramu Ramanathan, de la Universidad de California, San Diego. Estamos muy en deuda con el Profesor Ramu Ramanathan por dejar este código disponible bajo la Licencia Pública General GNU y por hacer de guía en el desarrollo de Gretl.

También estamos agradecidos a los autores de varios manuales de Econometría por su permiso para incluir en la distribución de Gretl varios conjuntos de datos asociados a sus textos. Esta lista actualmente incluye a William Greene, autor de *Econometric Analysis*; Jeffrey Woolridge (*Introductory Econometrics: A Modern Approach*); James Stock and Mark Watson (*Introduction*

to *Econometrics*); Damodar Gujarati (*Basic Econometrics*); y Rusell Davidson y James MacKinnon (*Econometric Theory and Methods*).

La estimación GARCH en Gretl se basa en el código depositado en el archivo del *Journal of Applied Econometrics* por los profesores Fiorentini, Calzolari y Panattoni y el código para generar valores p para los contrastes de Dickey-Fuller se debe a James MacKinnon. Agradecemos a estos autores su permiso para utilizar sus trabajos.

Con respecto a la internacionalización de Gretl, queremos dar las gracias a Ignacio Díaz-Emparanza, Michel Robitaille, Cristian Rigamonti, Tadeusz y Pawel Kufel, y Markus Hahn, quienes prepararon las traducciones al Español, Francés, Italiano, Polaco y Alemán respectivamente.

Gretl se ha beneficiado mucho del trabajo realizado por numerosos programadores de software de código abierto y libre: para más detalles véase el Apéndice B. Nuestro agradecimiento también a 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.

Muchos usuarios de Gretl han aportado sugerencias útiles e informes de error. A este respecto hemos de dar las gracias en particular a Ignacio Díaz-Emparanza, Tadeusz Kufel, Pawel Kufel, Alan Isaac, Cri Rigamonti y Dirk Eddelbuettel, quien además mantiene el paquete Gretl para Debian GNU/Linux.

### 1.3. Instalación de los programas

#### Linux

En la plataforma Linux<sup>1</sup> tenemos la opción de compilar el código Gretl o usar un paquete ya precompilado. 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). 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](http://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-1.1.5`).
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 necesario no se encuentre en el sistema, y por lo tanto falle todo el proceso de instalación, consúltese el Apéndice B.

---

<sup>1</sup>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.

Gretl ofrece soporte para 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`.

### **MS Windows**

La versión para MS Windows se distribuye en un archivo autoejecutable. La instalación consiste simplemente en bajar `Gretl_install.exe` y ejecutar este programa. Durante el proceso de instalación se le preguntará por un lugar donde instalar el paquete (por defecto se instala 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 una 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 emergente: 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

### 2.1. 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 11 y el *Guía de instrucciones de Gretl*.

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.<sup>1</sup>

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.

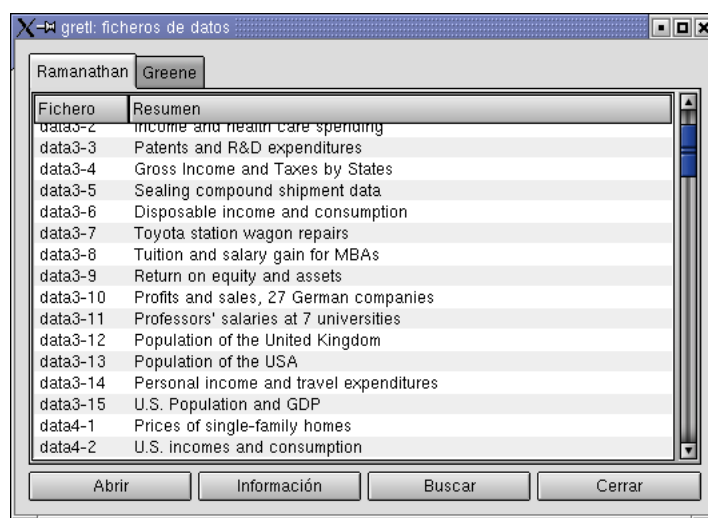


Figura 2.1: Ventana de archivo de datos para ejercicios prácticos

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`.

<sup>1</sup>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 `gretlw32.exe`. En sistemas Linux también se instala un lote de instrucciones de envoltura (wrapper script) llamado `gretl`.

En ventanas gretl 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.

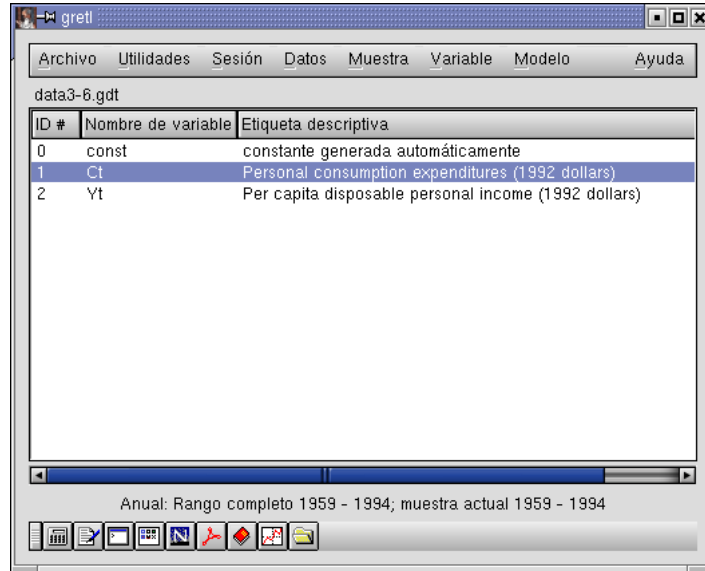


Figura 2.2: Ventana principal, con un archivo de datos para ejercicios abierto

Bien, ¿cuál es el siguiente paso? gretl intenta que las diferentes opciones del menú sean bastante explícitas. Primero, hagamos una breve visita al menú Modelo; en Sección 2.3 se realiza un breve recorrido a lo largo de los menús de la ventana principal.

El menú Modelo de gretl 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.

## 2.2. 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).

gretl: modelo 1

Archivo Editar Contrastes Gráficos Datos del Modelo LaTeX

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
CCV	41940,3	RICE	42085,9		

Cerrar

Figura 2.4: Ventana de resultados del modelo

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.<sup>2</sup>

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

<sup>2</sup>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.

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.

### 2.3. 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.
- Revisar bases de datos: Véanse 4.3.
- Crear conjunto de datos: Inicia la hoja de cálculo incorporada en el programa, para añadir datos de forma manual. Véase 4.4.
- 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`.
  - **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`.
  - **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.
- **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 [3.2](#).
  - **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 el capítulo 6.
    - **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` 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 4.5).
    - **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.
  - **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 fórmulas, consúltese el término `genr` en la ayuda *en línea* o véase la instrucción `genr`. Un ejemplo sencillo sería:

$$\text{foo} = \text{x1} * \text{x2}$$

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, consúltese el *Guía de instrucciones de Gretl* 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))

## 2.4. 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 Sección 2.3).
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)).

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

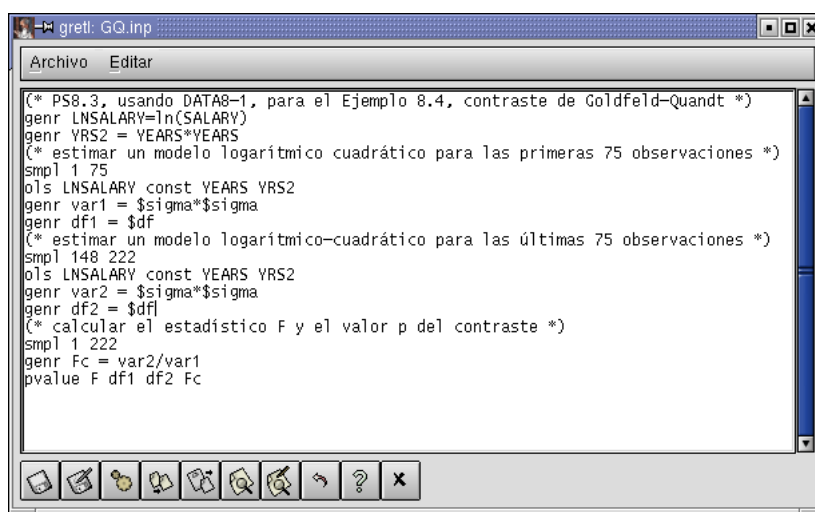
# Modos de trabajo

### 3.1. 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 *Guía de instrucciones de Gretl*.

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. Teclar cualquier instrucción que sea necesaria para poner en marcha las transformaciones de los datos (véase la instrucción `genr` en el *Guía de instrucciones de Gretl*). 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 visitar 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 el GUI es actualizado basándose en las instrucciones ejecutadas desde la consola, haciendo posible el trabajo en cualquier entorno.

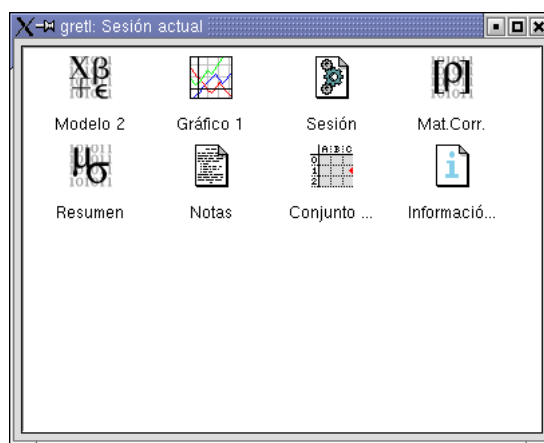
### 3.2. El concepto de sesión

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.

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.



**Figura 3.2:** Vista de iconos: a los iconos por defecto, se ha añadido un modelo y un gráfico

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.

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

### 4.1. 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](#).

### 4.2. 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 Sección [4.4](#).
- 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 Sección [4.4](#).
- 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).

### 4.3. 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, consúltese 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.

#### 4.4. 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 está 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 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). Enton-

ces 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 `bcih.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 RATS,<sup>1</sup> 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 apéndice A.

### 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): 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.

---

<sup>1</sup>Ver [www.estima.com](http://www.estima.com)

#### 4.5. 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.<sup>2</sup>

---

<sup>2</sup>`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

# Funciones especiales en `genr`

### 5.1. Introducción

La instrucción `genr` proporciona una manera flexible de definir nuevas variables. Está documentada en la *Guía de instrucciones de Gretl*. Este capítulo ofrece una explicación más extensa de algunas funciones especiales que están disponibles por medio de `genr` y algunas de las características más peculiares de la instrucción.

### 5.2. Filtros de series temporales

Una de las funciones especializadas de `genr` es el filtrado de series temporales. Actualmente están disponibles dos filtros, el de Hodrick-Prescott y el filtro pasabanda de Baxter-King. Se accede a ellos usando `hpfilt()` y `bkfilt()` respectivamente. En cada caso la función tiene un argumento: el nombre de la variable a procesar.

#### Filtro de Hodrick-Prescott

Una serie temporal  $y_t$  puede descomponerse en un componente de tendencia o de crecimiento  $g_t$  y un componente cíclico  $c_t$ .

$$y_t = g_t + c_t, \quad t = 1, 2, \dots, T$$

El filtro de Hodrick-Prescott efectúa dicha descomposición mediante la minimización de:

$$\sum_{t=1}^T (y_t - g_t)^2 + \lambda \sum_{t=2}^{T-1} ((g_{t+1} - g_t) - (g_t - g_{t-1}))^2.$$

El primer término de arriba es la suma del componente cíclico  $c_t = y_t - g_t$  al cuadrado. El segundo término es la suma de cuadrados de la segunda diferencia del componente de tendencia multiplicada por  $\lambda$ . Este segundo término penaliza las variaciones en la tasa de crecimiento del componente de tendencia: cuanto mayor sea el valor de  $\lambda$ , mayor es la penalización y así, más suave el componente de tendencia.

Nótese que la función `hpfilt` en `gretl` produce el componente cíclico,  $c_t$ , de la serie original. Si se desea obtener la tendencia suavizada, se puede restar el ciclo a la serie original

```
genr ct = hpfilt(yt)
genr gt = yt - ct
```

Hodrick y Prescott (1997) sugieren que, para datos trimestrales, es razonable un valor de  $\lambda = 1600$ . El valor por defecto en `gretl` es 100 veces el cuadrado de la periodicidad de los datos (lo cual, obviamente, da 1600 para datos trimestrales). Ese valor puede ajustarse usando la instrucción `set` con un valor para el parámetro `hp_lambda`. Por ejemplo, `set hp_lambda 1200`.

#### Filtro de Baxter y King

Considérese la representación espectral de una serie temporal  $y_t$ :

$$y_t = \int_{-\pi}^{\pi} e^{i\omega t} dZ(\omega)$$

Para extraer el componente de  $y_t$  de frecuencia entre  $\underline{\omega}$  y  $\overline{\omega}$  se podría aplicar un filtro *pasabanda*:

$$c_t^* = \int_{-\pi}^{\pi} F^*(\omega) e^{i\omega t} dZ(\omega)$$

donde  $F^*(\omega) = 1$  para  $\underline{\omega} < |\omega| < \overline{\omega}$  y 0 en el resto. Esto implicaría, en el dominio del tiempo, aplicar a la serie un filtro con un número infinito de coeficientes, lo cual no es algo deseable. El filtro pasabanda de Baxter y King aplica a  $y_t$  un polinomio finito en el operador de retardos  $A(L)$ :

$$c_t = A(L)y_t$$

donde  $A(L)$  se define como

$$A(L) = \sum_{i=-k}^k a_i L^i$$

Los coeficientes  $a_i$  se eligen de manera que  $F(\omega) = A(e^{i\omega})A(e^{-i\omega})$  es la mejor aproximación a  $F^*(\omega)$  para un  $k$  dado. Claramente, cuanto mayor sea  $k$  mejor será la aproximación, pero como hay que descartar  $2k$  observaciones, hay que buscar un valor de compromiso para  $k$ . Por otra parte, el filtro tiene también otras características teóricas interesantes, entre ellas la propiedad de que  $A(1) = 0$ , así que, una serie con una sólo raíz unitaria se transforma en estacionaria mediante la aplicación del filtro.

En la práctica, normalmente se usa el filtro con datos mensuales o trimestrales para extraer el componente de “ciclo de negocios”, es decir, el componente cíclico de periodo entre 6 y 36 trimestres. Los valores más usuales que se eligen para  $k$  son 8 ó 12 (puede ser mayor para series mensuales). Los valores por defecto para las cotas de frecuencia son 8 y 32, y el valor por defecto para el orden de aproximación,  $k$ , es 8. Estos valores se pueden ajustar mediante la instrucción `set`. El parámetro para elegir los límites de la frecuencia es `bkbp_limits` y el parámetro para  $k$  es `bkbp_k`. Así, por ejemplo, si se estuvieran utilizando datos mensuales y se desea ajustar las cotas de frecuencia a 18 y 96 y  $k$  a 24, se ejecutaría

```
set bkbp_limits 18 96
set bkbp_k 24
```

Esos valores permanecerían en uso para llamadas posteriores a la función `bkfilter` hasta que se cambien de nuevo mediante la instrucción `set`.

### 5.3. Remuestreo y *bootstrap*

Otra función especial de `genr` es el remuestreo con reemplazamiento de una serie. Dada una serie de datos original  $x$ , la instrucción

```
genr xr = resample(x)
```

crea una nueva serie en la que cada uno de sus elementos se obtiene aleatoriamente desde los elementos de  $x$ . Si la serie original tiene 100 observaciones, cada elemento de  $x$  puede ser seleccionado con probabilidad  $1/100$  en cada extracción. Así el efecto es “mezclar” los elementos de  $x$ , con la peculiaridad de que cada uno de ellos puede aparecer más de una vez, o ninguna, en  $xr$ .

El uso primario de esta función es la construcción de intervalos de confianza *bootstrap* o valores  $p$ . He aquí un ejemplo simple. Supongamos que estimamos una regresión simple de  $y$  sobre  $x$  por medio de MCO y encontramos que el estadístico  $t$  del coeficiente de pendiente se muestra igual a 2,5 con 40 grados de libertad. El valor  $p$  a dos colas para la hipótesis nula de que el parámetro de pendiente es igual a cero es entonces 0,0166 en la distribución  $t(40)$ . Sin embargo, dependiendo del contexto en que nos encontremos, podemos dudar de si el cociente entre el coeficiente estimado y la desviación típica realmente sigue la distribución  $t(40)$ . En ese caso, podríamos obtener un valor  $p$  “*bootstrap*” como se muestra en el Ejemplo 5.1

Bajo la hipótesis nula de que la pendiente con respecto a  $x$  es cero,  $y$  es simplemente igual a su media más un término de error. Se calcula la media de  $y$  y se simulan nuevos datos de

$y$  mediante la media de  $y$  más una muestra obtenida por remuestreo de los residuos de la estimación MCO inicial y se vuelve a estimar el modelo con las observaciones simuladas de  $y$ . Se repite este procedimiento un gran número de veces y se cuenta el número de veces que el valor absoluto del estadístico  $t$  es mayor que 2,5; la proporción correspondiente a estos casos es nuestro valor  $p$  bootstrap. Una buena referencia sobre contrastes basados en simulación y bootstrap es Davidson y MacKinnon (2004, capítulo 4).

#### Ejemplo 5.1: Cálculo de valor $p$ bootstrap

```

ols y 0 x
# se guardan los residuos
genr ui = $uhat
scalar ybar = mean(y)
# número de replicaciones para el bootstrap
scalar replcs = 10000
scalar tcount = 0
series ysim = 0
loop replcs --quiet
  # generar la y simulada mediante remuestreo
  ysim = ybar + resample(ui)
  ols ysim 0 x
  scalar tsim = abs(coeff(x) / stderr(x))
  tcount += (tsim > 2.5)
endloop
printf "proporcion de casos en que |t| > 2.5 = %g\n", \
  tcount / replcs

```

#### 5.4. Valores ausentes

Hay cuatro funciones especiales disponibles para el manejo de valores ausentes. La función booleana `missing()` toma como único argumento el nombre de una variable; devuelve una serie con valor 1 para cada observación en que la serie tiene un valor ausente y un valor de 0 en los demás casos (es decir, si la variable dada tiene un valor válido en esa observación). La función `ok()` es la complementaria de `missing`; es sólo un atajo de `!missing` (donde `!` es el operador booleano NO). Por ejemplo, se pueden contar los valores ausentes de la variable `x` usando

```
genr nmiss_x = sum(missing(x))
```

La función `zeromiss()`, que, de nuevo, toma el nombre de una serie como único argumento, devuelve una serie en la que a todos los valores cero se les asigna el código de valor ausente. Esto debe usarse con precaución — no son los mismos valores ausentes que ceros — pero puede ser útil en algunos contextos. Por ejemplo, es posible determinar cuál es la primera observación válida de una variable `x` utilizando

```
genr time
genr x0 = min(zeromiss(time * ok(x)))
```

La función `misszero()` realiza lo contrario de `zeromiss`, es decir, convierte todos los valores ausentes en ceros.

Puede ser conveniente comentar la propagación de los valores ausentes dentro de las fórmulas construidas mediante `genr`. La regla general es que, en las operaciones aritméticas en las que intervienen dos variables, si alguna de las variables tiene un valor ausente en la observación  $t$ , entonces la serie resultante también tendrá un valor ausente en  $t$ . La única excepción a esta regla es la multiplicación por cero: el cero multiplicado por un valor ausente produce un cero (ya que esto es matemáticamente válido independientemente de cuál sea el valor desconocido).

### 5.5. Recuperar variables internas

La instrucción `genr` proporciona un método para recuperar varios valores definidos internamente que calcula el programa al estimar los modelos o al contrastar hipótesis. Las variables que pueden ser recuperadas mediante este procedimiento están listadas en la *Guía de instrucciones de Gretl*; aquí sólo detallaremos un poco más el uso de las variables especiales `$test` y `$pvalue`.

Esas variables contienen, respectivamente, el valor del último estadístico de contraste calculado utilizando una instrucción de contraste específica y el valor `p` para ese estadístico de contraste. Si no se ha ejecutado ninguna instrucción de contraste en el momento en que se referencian esas variables, producirán el código de valor ausente. Las “instrucciones de contraste específicas” que funcionan de esta manera son las siguientes: `add` (contraste conjunto para la significación de variables añadidas a un modelo); `adf` (contraste de Dickey-Fuller aumentado, ver más abajo); `arch` (contraste de ARCH); `chow` (contraste de Chow de cambio estructural); `coeffsum` (contraste de la suma de los coeficientes especificados); `cusum` (el estadístico  $t$  de Harvey-Collier); `kpss` (contraste de estacionariedad KPSS, el valor `p` no está disponible); `lmtest` (ver más abajo); `meantest` (contraste de diferencia de medias); `omit` (contraste conjunto de significatividad de las variables que se han omitido en un modelo); `reset` (Contraste RESET de Ramsey); `restrict` (restricciones lineales en general); `runs` (contraste de rachas, de aleatoriedad); `testuhat` (contraste de normalidad de los residuos); y `vartest` (contraste de diferencia de varianzas). En la mayoría de los casos se guardan tanto el valor del estadístico (`$test`) como el valor `p` (`$pvalue`); la excepción es el contraste KPSS, para el que actualmente no está disponible el valor `p`.

Es importante advertir respecto a este mecanismo que las variables internas `$test` y `$pvalue` se sobrescriben cada vez que se ejecuta alguno de los contrastes listados anteriormente. Si Vd. desea referirse a esos valores, debe hacerlo en el punto correcto dentro de la secuencia de instrucciones de `gretl`.

También hay que mencionar que algunas de estas instrucciones de contraste generan, por defecto, más de un estadístico de contraste y valor `p`; en estos casos sólo se guardan los últimos valores. Para controlar adecuadamente los valores que se recuperan por medio de `$test` y `$pvalue` se debería formular la instrucción de contraste de tal forma que el resultado no sea ambiguo. Este comentario se refiere en particular a las instrucciones `adf` y `lmtest`.

- Por defecto, la instrucción `adf` genera tres variantes del contraste de Dickey-Fuller: una basada en una regresión que incluye una constante, otra usando una constante y una tendencia lineal y otra utilizando una constante y una tendencia cuadrática. Cuando Vd. desea referirse a `$test` o `$pvalue` en relación a esta instrucción, Vd. puede controlar la variante que se graba mediante el uso de las opciones `--nc`, `--c`, `--ct` o `--ctt` con `adf`.
- Por defecto, la instrucción `lmtest` (que debe ejecutarse después de una regresión MCO) desarrolla varios contrastes de diagnóstico sobre la regresión en cuestión. Para controlar cuál se graba en `$test` y `$pvalue` Vd. debería limitar el contraste usando una de las opciones `--logs`, `--autocorr`, `--squares` o `--white`.

Como ayuda, al trabajar con valores recuperados mediante `$test` y `pvalue` se escribe el tipo de contraste correspondiente en la etiqueta descriptiva de la variable generada. Vd. puede leer la etiqueta de la variable mediante la instrucción `label` (con sólo un argumento, el nombre de la variable), para comprobar que ha recuperado el valor correcto. Como ilustración de esto véase la siguiente sesión interactiva.

```
? adf 4 x1 --c
```

```
Contrastes aumentados de Dickey-Fuller, orden 4, para x1:
tamaño muestral 59
hipótesis nula de raíz unitaria: a = 1
```

```
contraste con constante
modelo: (1 - L)y = b0 + (a-1)*y(-1) + ... + e
```

valor estimado de  $(a - 1)$ : -0,216889  
Estadístico de contraste:  $t = -1,83491$   
valor p asintótico 0,3638

Valores p basados en el artículo de MacKinnon (JAE, 1996)

```
? genr pv = $pvalue  
Generada escalar pv (ID 13) = 0,363844  
? label pv  
pv=valor p de Dickey-Fuller (scalar)
```

## Capítulo 6

# Datos de Panel

### 6.1. 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.<sup>1</sup>

- 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 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 *Guía de*

<sup>1</sup>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.

*instrucciones de Gretl*) o la opción del menú GUI “Muestra, Seleccionar frecuencia, observación inicial..”).

## 6.2. 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 paneldum`). 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 paneldum -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 de 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 lo demás casos, se puede hacer lo siguiente

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

## 6.3. 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  $x_1$ , 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  $x_1_1$ , si no más bien la última observación de  $x_1$  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 6.2 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 *Guía de instrucciones de Gretl*).

#### Ejemplo 6.1: 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 dum --dummy
# crear de nuevo la estructura de observaciones, para 56 unidades
setobs 29 1.01 56.29
```

### 6.4. 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.

### 6.5. Ilustración: La Tabla Mundial de Penn

La Tabla Mundial de Penn (Penn World Table) (dirección [pwt.econ.upenn.edu](http://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](http://gretl.sourceforge.net/gretl_data.html) (se puede descargar gratuitamente, aunque no está incluido en el paquete principal de gretl).

El Ejemplo 6.2 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 6.2: 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 7

# Gráficos y Diagramas

### 7.1. 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](http://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](http://www.gnuplot.info) de `gnuplot` o ir a [www.gnuplot.info](http://www.gnuplot.info).

También consúltese la entrada `gnuplot` más adelante en el *Guía de instrucciones de Gretl*— y las instrucciones `graph` y `plot` para gráficos de ASCII “rápidos, es decir, estilo borrador”.

### 7.2. 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

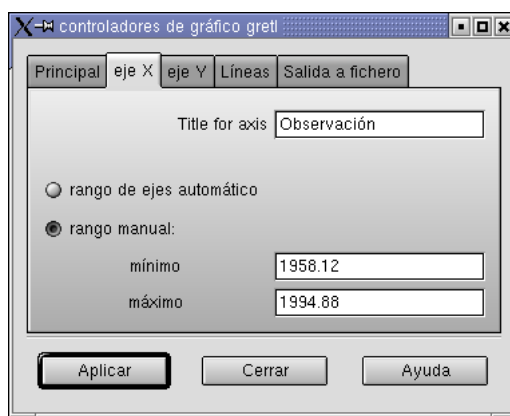


Figura 7.1: Controlador gnuplot de gretl

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 8

# Construcción de iteraciones

### 8.1. 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”, 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 8.1 se muestra un sencillo ejemplo de simulación de Monte Carlo.

#### Ejemplo 8.1: 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 *Guía de instrucciones de Gretl* para información sobre cómo generar series pseudo-aleatorias repetibles.

## 8.2. 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 un 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 8.2 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...".

## 8.3. 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 8.3 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 8.2:** 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

```

**Ejemplo 8.3:** Ejemplo de un bucle indexado

```

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

```

## Capítulo 9

# Cointegración y Modelos de Corrección del Error

### 9.1. El contraste de cointegración de Johansen

El contraste de cointegración de Johansen requiere tener en cuenta la hipótesis que está uno dispuesto a establecer sobre los componentes deterministas, lo que deriva en los famosos “cinco casos.” Para ilustrar de forma completa y exhaustiva los cinco casos se requiere una gran cantidad de álgebra matricial, pero para comprender esta cuestión de manera intuitiva se puede utilizar un ejemplo sencillo. Sea  $x_t$  una serie que se comporta como sigue

$$x_t = m + x_{t-1} + \varepsilon_t$$

donde  $m$  es un número real y  $\varepsilon_t$  es un ruido blanco. Es fácil demostrar que  $x_t$  sigue un paseo aleatorio que fluctúa alrededor de una tendencia determinista con pendiente igual a  $m$ . En el caso especial  $m = 0$ , la tendencia determinista desaparece y  $x_t$  es simplemente un paseo aleatorio.

Considera ahora otro proceso  $y_t$ , definido por

$$y_t = k + x_t + u_t$$

donde, de nuevo,  $k$  es un número real y  $u_t$  sigue un proceso de ruido blanco. Dado que  $u_t$  es estacionario por definición,  $x_t$  e  $y_t$  están cointegrados: esto es, su diferencia

$$z_t = y_t - x_t = k + u_t$$

es un proceso estacionario. Para  $k = 0$ ,  $z_t$  es un simple ruido blanco de media cero, mientras que si  $k \neq 0$  el proceso  $z_t$  es un ruido blanco con media distinta de cero.

Las dos ecuaciones anteriores se pueden representar conjuntamente como un sistema VAR(1).

$$\begin{bmatrix} y_t \\ x_t \end{bmatrix} = \begin{bmatrix} k + m \\ m \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \begin{bmatrix} u_t + \varepsilon_t \\ \varepsilon_t \end{bmatrix}$$

o en forma de un Modelo de Corrección de Error (MCE)

$$\begin{aligned} \begin{bmatrix} \Delta y_t \\ \Delta x_t \end{bmatrix} &= \begin{bmatrix} k + m \\ m \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \begin{bmatrix} u_t + \varepsilon_t \\ \varepsilon_t \end{bmatrix} = \\ &= \begin{bmatrix} k + m \\ m \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \begin{bmatrix} u_t + \varepsilon_t \\ \varepsilon_t \end{bmatrix} = \\ &= \mu_0 + \alpha \beta' \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \eta_t = \mu_0 + \alpha z_{t-1} + \eta_t, \end{aligned}$$

donde  $\beta$  es el vector de cointegración y  $\alpha$  es el vector de “pesos” o de “ajuste” hacia el equilibrio. Podemos considerar ahora tres posibles casos:

1.  $m \neq 0$ : En este caso  $x_t$  presenta una tendencia lineal determinista, tal y como hemos comentado anteriormente; por lo tanto  $y_t$  también la presenta, ya que en media se mantiene a una distancia  $k$  de la serie  $x_t$ . El vector  $\mu_0$  no está restringido. Este es el caso por defecto de la instrucción `vecm` de `gretl`.

2.  $m = 0$  y  $k \neq 0$ : En este caso,  $x_t$  no presenta una tendencia lineal determinista y en consecuencia  $y_t$  tampoco. Sin embargo, la distancia media entre  $y_t$  y  $x_t$  no es cero. El vector  $\mu_0$  viene dado por

$$\mu_0 = \begin{bmatrix} k \\ 0 \end{bmatrix}$$

el cual no es cero por lo que el MCE anterior tiene un término constante. Sin embargo, el vector de parámetros asociados a la constante está sujeto a la restricción de que su segundo elemento tiene que ser 0. En términos más generales,  $\mu_0$  es proporcional al vector  $\alpha$ . Notese que el MCE también se puede escribir de la siguiente forma:

$$\begin{bmatrix} \Delta y_t \\ \Delta x_t \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & -k \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \\ 1 \end{bmatrix} + \begin{bmatrix} u_t + \varepsilon_t \\ \varepsilon_t \end{bmatrix}$$

donde el intercepto está dentro del vector de cointegración. Este caso se conoce con el nombre de “constante restringida”; se puede elegir en el comando de gretl `vecm` utilizando la etiqueta opcional `--rc`

3.  $m = 0$  y  $k = 0$ : Este caso es el más restrictivo: claramente, ni  $x_t$  ni  $y_t$  presentan tendencias deterministas, y la distancia media entre las dos series es cero. El vector  $\mu_0$  también es 0, lo cual explica porqué este caso se denomina “sin constante.” Este último caso se especifica utilizando la etiqueta opcional `--nc` con la instrucción `vecm`.

En general en la práctica, la elección entre estas tres posibilidades se basa en una mezcla entre observación empírica y razonamiento económico. Si las variables de estudio parecen mostrar una tendencia lineal entonces no se debería imponer ninguna restricción sobre el intercepto. En otro caso, nos podemos preguntar si tiene sentido especificar una relación de cointegración que incluya un intercepto distinto de cero. Un ejemplo donde esto sí es razonable es en la relación entre dos tipos de interés: en general, estas variables no presentan tendencia determinista, pero el VAR puede aún tener un intercepto porque la diferencia entre los dos (el “spread o diferencial de los tipos de interés”) puede ser estacionario alrededor de una media distinta de cero (por ejemplo porque hay una prima de riesgo o de liquidez). El ejemplo anterior se puede generalizar en tres direcciones:

1. Si el VAR es de orden mayor que uno, el algebra se complica más pero se obtienen las mismas conclusiones.
2. Si las series consideradas en el análisis del VAR son más de dos, el rango de cointegración  $r$  puede ser mayor que 1. En ese caso,  $\alpha$  es una matriz con  $r$  columnas, y el caso de la constante restringida implicaría la restricción de que  $\mu_0$  debería de ser una combinación lineal de las columnas de  $\alpha$ .
3. Si en el modelo se incluye una tendencia lineal  $t$ , la parte determinista del VAR es en este caso  $\mu_0 + \mu_1 t$ . El razonamiento es prácticamente el mismo que antes excepto que ahora la atención se centra en  $\mu_1$  más que en  $\mu_0$ . El caso anterior de “la constante restringida” que se ha discutido antes sería ahora el caso de una “tendencia restringida”, tal que la o las relaciones de cointegración incluyen una tendencia pero las primeras diferencias de las variables en cuestión no. En el caso de una tendencia no restringida, esta aparece tanto en las relaciones de cointegración como en las primeras diferencias de las variables, lo cual corresponde con la presencia de una tendencia cuadrática en las series en niveles. Estos dos casos se especifican con las etiquetas opcionales `--crt` y `--ct`, respectivamente, utilizando el comando `vecm`.

## Capítulo 10

# Solución de problemas en gretl

### 10.1. 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.)

### 10.2. 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 `gretl\w32.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.<sup>1</sup>

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.

	<i>Gráficos</i>	<i>Composición</i>	<i>GNU R</i>
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, consúltese la ayuda *en línea* o la documentación para su sistema operativo (shell).

---

<sup>1</sup>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

### 11.1. 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 el *Guía de instrucciones de Gretl*).

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.<sup>1</sup>

Las más comunes son :

<i>Combinaciones de teclas</i>	<i>Efecto</i>
<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.

### 11.2. 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 `tsls`, o como marcador para una primera o última observación sin cambios en la instrucción `smpl`. 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.

<sup>1</sup>En realidad, las combinaciones dadas a continuación son sólo las que hay por defecto; pueden ser personalizadas. Véase el [manual de readline](#).

- 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
```

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

### A.1. 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 Ud. 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.

### A.2. 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 6 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.
*)
l dinero l renta 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 .lbl, 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.

### A.3. 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](http://info-zip.org). La dirección de Internet de Libxmls es [xmlsoft.org](http://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](http://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](http://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 esta 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](http://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](http://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 “Lib-gretl 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 `GRETLM_P_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 Baoticchi 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 Baoticchi 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 (se puede utilizar `genr`) y a continuación pedir su presentación mediante la orden `print -t` (véase el *Guía de instrucciones de Gretl*). 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](http://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](http://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.

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

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

**Estima (RATS)** <http://www.estima.com/>

**Página web del escritorio Gnome** <http://www.gnome.org/>

**Biblioteca de Precisión Múltiple de GNU** <http://swox.com/gmp/>

**Página web de GNU Octave** <http://www.octave.org/>

**Página web de GNU R** <http://www.r-project.org/>

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

**Página web de Gnuplot** <http://www.gnuplot.info/>

**Manual en línea de Gnuplot** <http://ricardo.ecn.wfu.edu/gnuplot.html>

**Página de datos de Gretl** [http://gretl.sourceforge.net/gretl\\_data.html](http://gretl.sourceforge.net/gretl_data.html)

**Página web de Gretl** <http://gretl.sourceforge.net/>

**Página web de GTK+** <http://www.gtk.org/>

**Port GTK+ para win32** <http://user.sgic.fi/~tml/gimp/win32/>

**Página web de Gtkextra** <http://gtkextra.sourceforge.net/>

**Página web de InfoZip** <http://www.info-zip.org/pub/infozip/zlib/>

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

**Página web de Mingw (gcc para win32)** <http://www.mingw.org/>

**Minpack** <http://www.netlib.org/minpack/>

**Tabla Mundial de Penn (Penn World Table)** <http://pwt.econ.upenn.edu/>

**Página web de GNU Readline** <http://cnswww.cns.cwru.edu/~chet/readline/rltop.html>

**Manual de GNU Readline** <http://cnswww.cns.cwru.edu/~chet/readline/readline.html>

**Página web de Xmlsoft** <http://xmlsoft.org/>

## Bibliografía

- Akaike, H. (1974) "A New Look at the Statistical Model Identification", *IEEE Transactions on Automatic Control*, AC-19, pp. 716-23.
- Baiocchi, G. and Distaso, W. (2003) "GRET: Econometric software for the GNU generation", *Journal of Applied Econometrics*, 18, pp. 105-10.
- Baxter, M. and King, R. G. (1995) "Measuring Business Cycles: Approximate Band-Pass Filters for Economic Time Series", National Bureau of Economic Research, Working Paper No. 5022.
- Belsley, D., Kuh, E. and Welsch, R. (1980) *Regression Diagnostics*, New York: Wiley.
- Berndt, E., Hall, B., Hall, R. and Hausman, J. (1974) "Estimation and Inference in Nonlinear Structural Models", *Annals of Economic and Social Measurement*, 3/4, pp. 653-65.
- 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.
- Davidson, R. and MacKinnon, J. G. (1993) *Estimation and Inference in Econometrics*, New York: Oxford University Press.
- Davidson, R. and MacKinnon, J. G. (2004) *Econometric Theory and Methods*, New York: Oxford University Press.
- Doornik, J. A. and Hansen, H. (1994) "An Omnibus Test for Univariate and Multivariate Normality", working paper, Nuffield College, Oxford.
- Doornik, J. A. (1998) "Approximations to the Asymptotic Distribution of Cointegration Tests", *Journal of Economic Surveys*, 12, pp. 573-93. Reprinted with corrections in M. McAleer and L. Oxley (1999) *Practical Issues in Cointegration Analysis*, Oxford: Blackwell.
- Fiorentini, G., Calzolari, G. and Panattoni, L. (1996) "Analytic Derivatives and the Computation of GARCH Estimates", *Journal of Applied Econometrics*, 11, pp. 399-417.
- Greene, William H. (2000) *Econometric Analysis*, 4th edition, Upper Saddle River, NJ: Prentice-Hall.
- Gujarati, Damodar N. (2003) *Basic Econometrics*, 4th edition, Boston, MA: McGraw-Hill.
- Hamilton, James D. (1994) *Time Series Analysis*, Princeton, NJ: Princeton University Press.
- Johansen, Søren (1995) *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*, Oxford: Oxford University Press.
- Kiviet, J. F. (1986) "On the Rigour of Some Misspecification Tests for Modelling Dynamic Relationships", *Review of Economic Studies*, 53, pp. 241-261.
- Kwiatkowski, D., Phillips, P.C.B., Schmidt, P. and Shin, Y. (1992) "Testing the Null of Stationarity Against the Alternative of a Unit Root: How Sure Are We That Economic Time Series Have a Unit Root?", *Journal of Econometrics*, 54, pp. 159-178.
- Locke, C. (1976) "A Test for the Composite Hypothesis that a Population has a Gamma Distribution", *Communications in Statistics — Theory and Methods*, A5(4), pp. 351-364.
- MacKinnon, J. G. (1996) "Numerical Distribution Functions for Unit Root and Cointegration Tests", *Journal of Applied Econometrics*, 11, pp. 601-618.
- 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.
- Maddala, G. S. (1992) *Introduction to Econometrics*, 2nd edition, Englewood Cliffs, NJ: Prentice-Hall.
- Matsumoto, M. and Nishimura, T. (1998) "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", *ACM Transactions on Modeling and Computer Simulation*, 8: 1.

- Neter, J. Wasserman, W. and Kutner, M. H. (1990) *Applied Linear Statistical Models*, 3rd edition, Boston, MA: Irwin.
- 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.
- Shapiro, S. and Chen, L. (2001) "Composite Tests for the Gamma Distribution", *Journal of Quality Technology*, 33, pp. 47-59.
- Silverman, B. W. (1986) *Density Estimation for Statistics and Data Analysis*, London: Chapman and Hall.
- Stock, James H. and Watson, Mark W. (2003) *Introduction to Econometrics*, Boston, MA: Addison-Wesley.
- Wooldridge, Jeffrey M. (2002) *Introductory Econometrics, A Modern Approach*, 2nd edition, Mason, Ohio: South-Western.