

Introducción a gretl

Economía Aplicada

Departamento de Economía
Universidad Carlos III de Madrid

Outline

- 1 ¿Qué es gretl?
- 2 Aspectos básicos de gretl
- 3 Importar Datos
- 4 Guardando un fichero gretl
- 5 Ejecutando el script
- 6 Primeros Ejercicios
- 7 Presentación resumida
- 8 Comandos sobre bases de datos
- 9 Comandos sobre variables
- 10 Gráficos

¿Qué es gretl?

- gretl es un acrónimo de Gnu Regression Econometrics and Time-series Library
- es un programa de aplicación de técnicas econométricas gratuito
- tiene una Graphical User Interface (gui) sencilla
- permite abrir archivos en diferentes formatos
- exporta los resultados en diferentes formatos
- muy importante para nosotros: admite scripts (guiones: secuencias de órdenes guardadas en un fichero de texto)

¿Cómo puedo obtener gretl?

- está instalado en las aulas informáticas de la Carlos III
- puede descargarse en <http://gretl.sourceforge.net> para instalarlo en tu PC
- corre en Windows, Mac, Linux

¿Cómo puedo trabajar con gretl? (1/2)

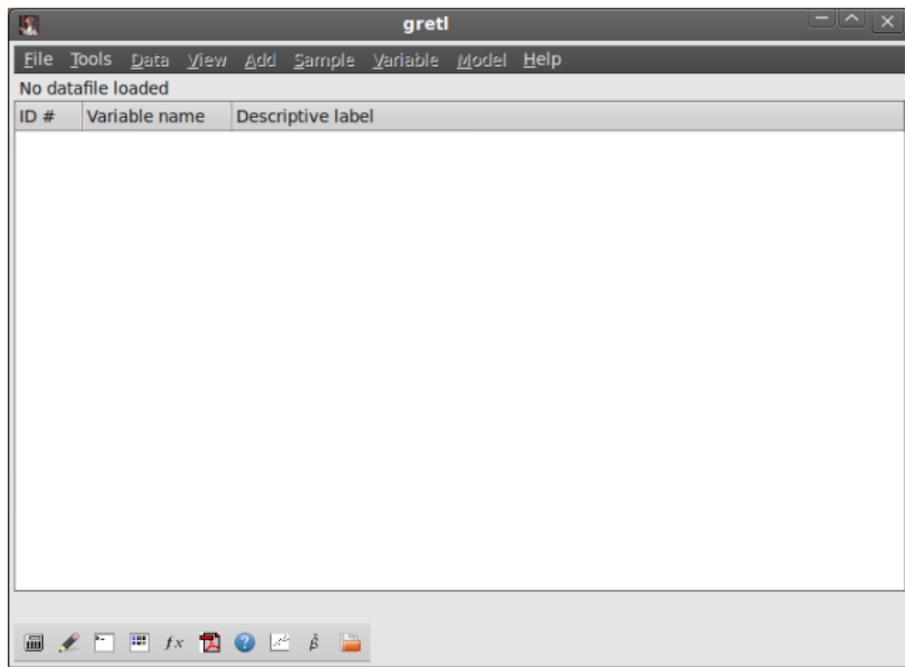
- para los principiantes, lo más fácil es con la gui
- también se puede trabajar con la “consola”: desde el signo (?) se pueden enviar órdenes a gretl. Se puede abrir desde el menu Tools o desde el icono gretl console en la barra de herramientas.
- la forma más eficiente es usando *scripts*:
 - 1 creamos un fichero script, escribimos órdenes para gretl – una orden por línea–, guardamos el fichero
 - 2 ejecutamos el fichero
 - 3 comprobamos el resultado
 - 4 si necesitamos hacer cambios en el script, los hacemos, guardamos la nueva versión y repetimos desde el paso 2

¿Cómo puedo trabajar con gretl? (2/2)

sabes cómo trabajar con la gui pero quieres aprender a hacer scripts...

- (casi) todas las acciones que hagas con la gui se guardan en un fichero llamado `session.inp`
- gretl tiene muchas bases de datos incluidas y más de 70 scripts para que practiques
- el manual da muy buenos consejos y tiene varios capítulos con soluciones de programación
- en este curso aprenderás muchos scripts que funcionan
- hoy vamos a usar la gui para crear un script que abre un fichero excel, lo modifica y lo guarda en un fichero con formato gretl

Ventana principal (1/2)

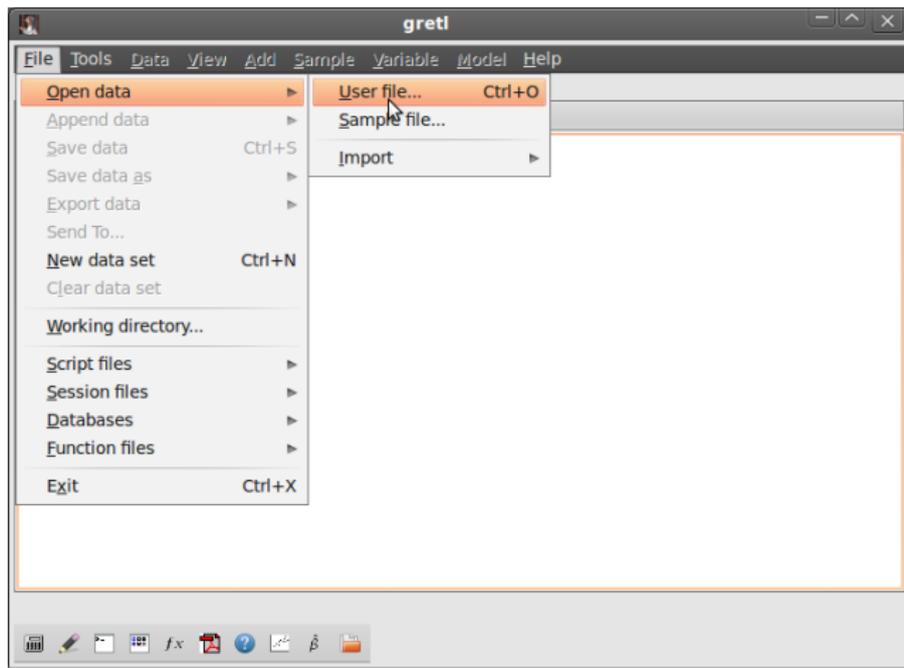


Ventana principal (2/2)

- en la parte superior de la ventana se encuentra la barra del menú. Desde allí se pueden importar datos, manipularlos, analizarlos, y controlar los resultados.
- en la parte inferior de la pantalla se encuentra la barra de herramientas de gretl. Entre otras cosas, se puede:
 - abrir la calculadora del sistema operativo
 - crear un nuevo script
 - acceder a la consola
 - abrir el manual en pdf

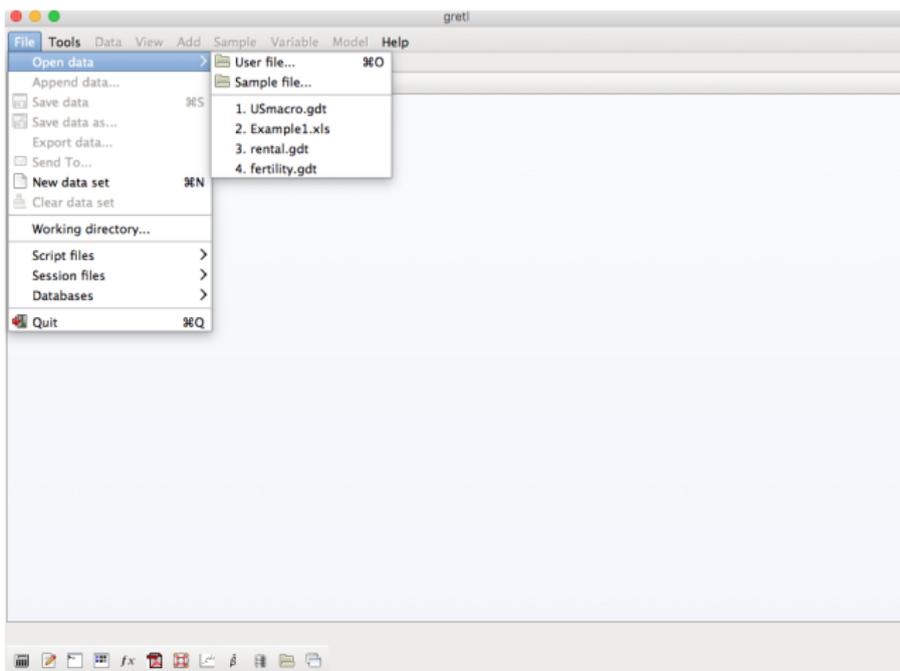
Abrir un fichero de datos de gretl (.gdt)

File/Open data/Sample file, File/Open data/User file



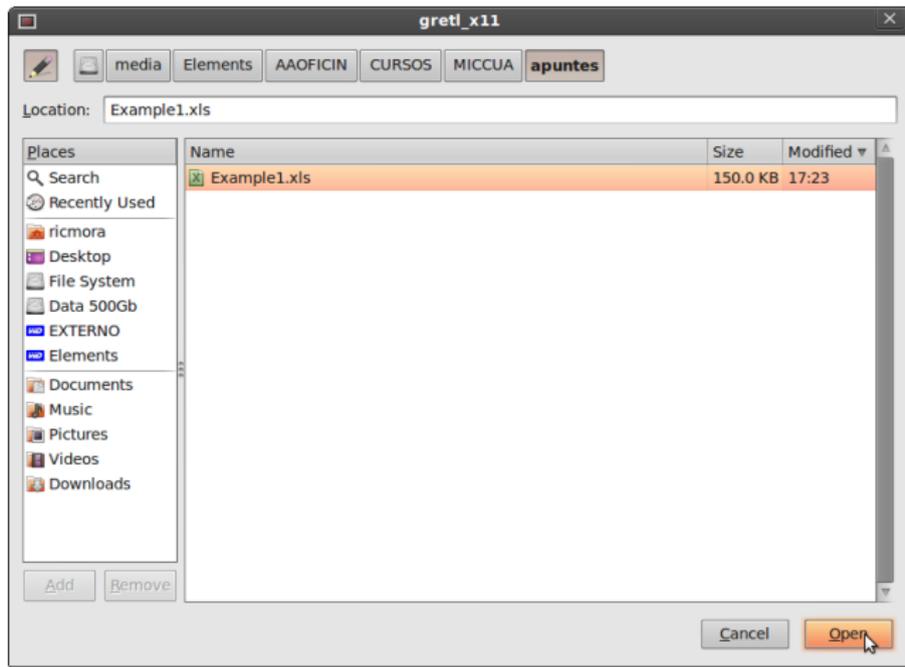
Importar desde un fichero Excel

File/Open data/User file



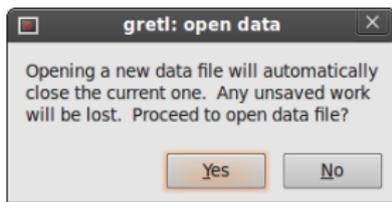
Importando Example1.xls

elige el fichero



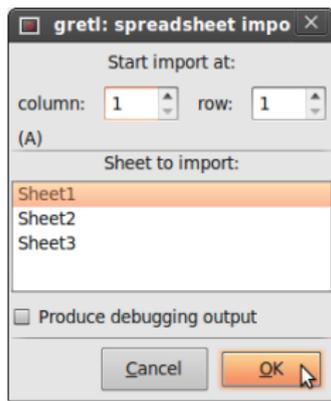
Importando Example1.xls

puede aparecer este aviso cuando usas la gui



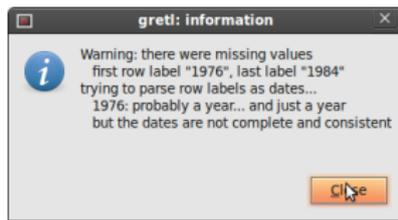
Importando Example1.xls

puedes importar cualquiera de las tres hojas



Importando Example1.xls

gret1 después da algo de información (que hoy pasamos por alto)



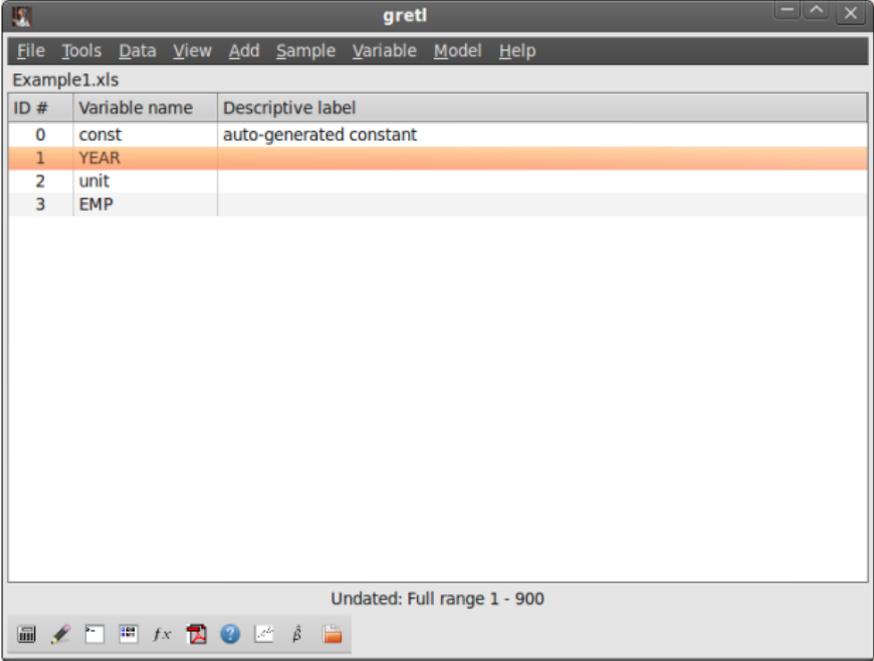
Importando Example1.xls

puedes decirle a gret1 si lo que importas es una serie temporal, una sección cruzada, un panel....



Importando Example1.xls

ya tenemos los datos cargados en gretl



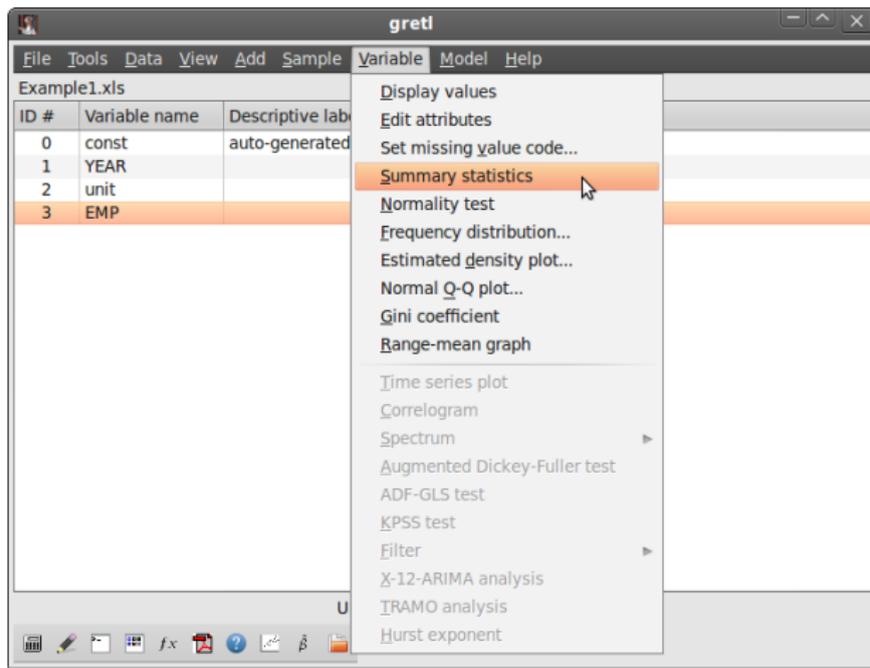
The screenshot shows the gretl software window with the following table of loaded variables:

ID #	Variable name	Descriptive label
0	const	auto-generated constant
1	YEAR	
2	unit	
3	EMP	

At the bottom of the window, it displays "Undated: Full range 1 - 900".

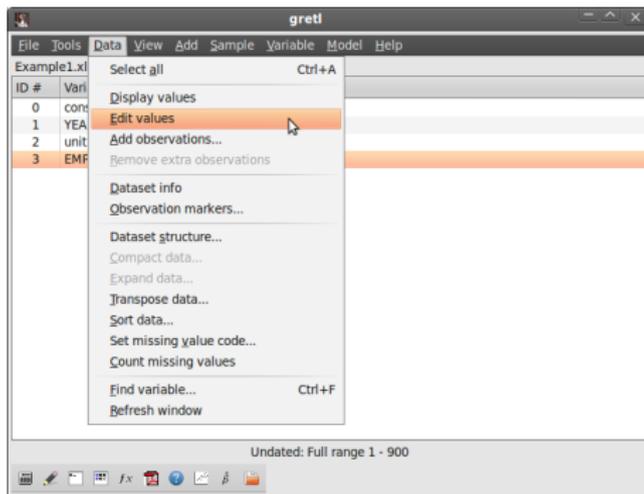
Describiendo una variable del fichero

Primero selecciona la variable, después: Variable/Summary statistics



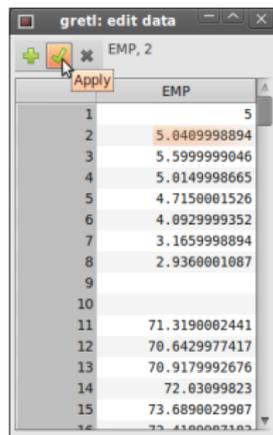
Editando una variable

Después de seleccionar la variable: Data/Edit Values



Editando una variable

ponemos "5" en la primera observación



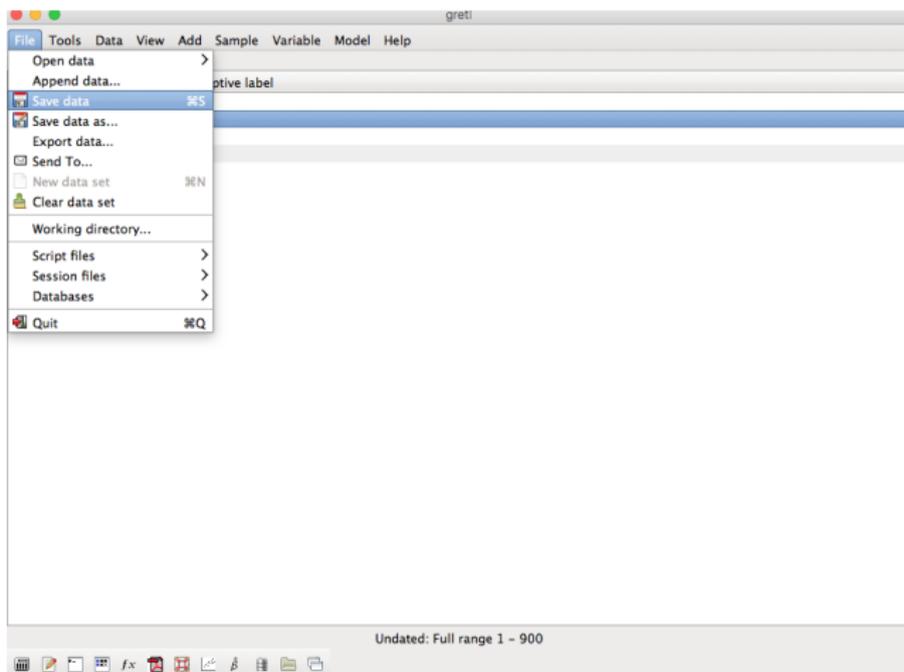
The screenshot shows the 'gret: edit data' window for the variable 'EMP, 2'. The window title is 'gret: edit data'. The variable name 'EMP, 2' is displayed at the top. Below the title bar, there are three icons: a green plus sign, a green checkmark, and a red X. The main area of the window is a table with the following data:

	EMP
1	5
2	5.0409998894
3	5.5999999046
4	5.0149998665
5	4.7150001526
6	4.0929999352
7	3.1659998894
8	2.9360001087
9	
10	
11	71.3190002441
12	70.6429977417
13	70.9179992676
14	72.03099823
15	73.6890029907
16	73.410003103

(para comprobar que lo hemos hecho bien, doble-click en EMP en la ventana principal)

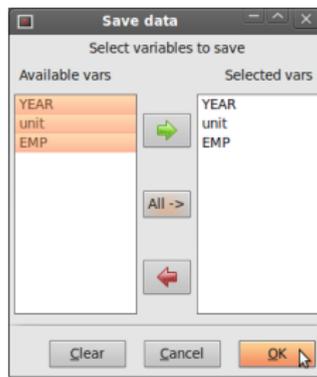
Guardando los datos como un nuevo fichero gret1

File > Save Data [as]



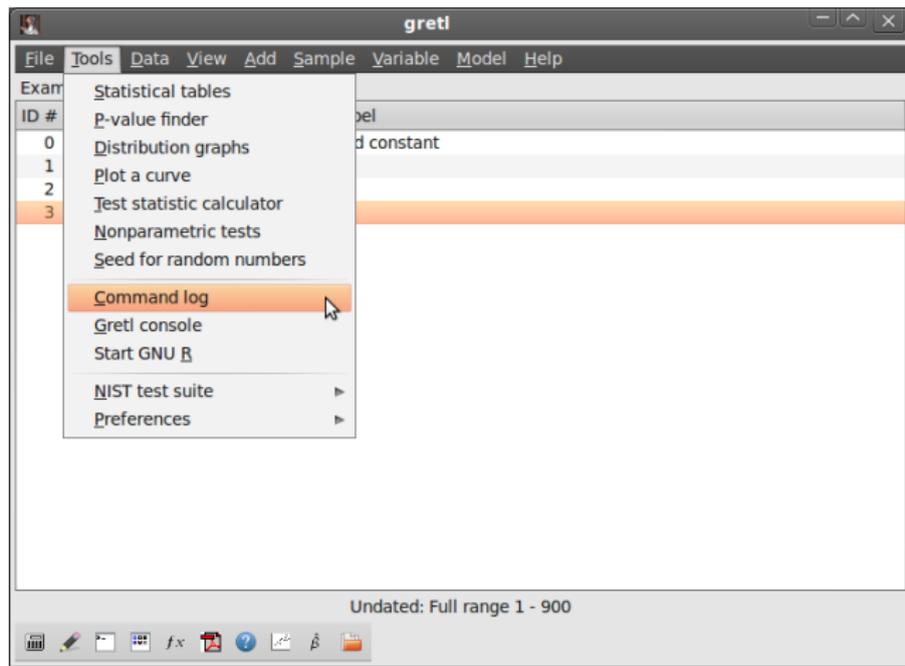
Guardando los datos como un nuevo fichero gret1

puedes seleccionar un subconjunto de las variables



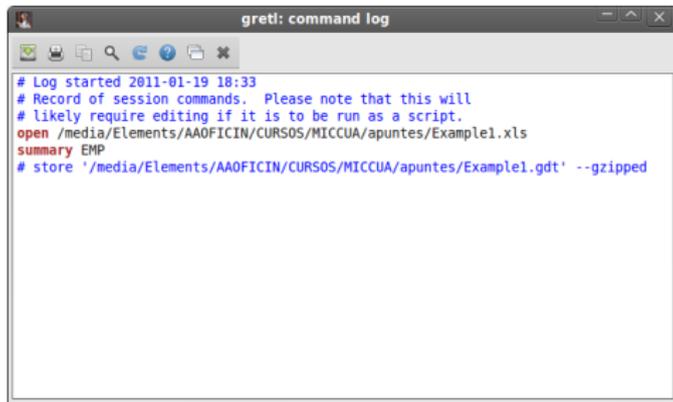
Mirando el script 1/2

Tools > Command log



Mirando el script 2/2

Notad que todas las líneas que parten con # son “comentarios”



```
# Log started 2011-01-19 18:33
# Record of session commands. Please note that this will
# likely require editing if it is to be run as a script.
open /media/Elements/AA0FICIN/CURSOS/MICCUA/apuntes/Example1.xls
summary EMP
# store '/media/Elements/AA0FICIN/CURSOS/MICCUA/apuntes/Example1.gdt' --gzipped
```

Cambiando el script

- Puedes abrir un nuevo script: usando File/Script Files/New script o usando el segundo botón de la barra de herramientas, o puedes guardar el command log como script (por defecto utiliza la extensión .inp).
- escribe las órdenes que quieres ejecutar usando una línea por orden
- las órdenes que no quieres ejecutar las precedes utilizando #
- para guardar el fichero, utiliza el botón de guardar
- para ejecutar el programa, clickea sobre el botón de ejecución

Más información sobre los scripts

- si tienes una orden más larga que una línea, utiliza (`\`) como orden de continuación
- utilizar scripts (y la terminal) presupone el uso correcto del lenguaje de `gretl`
- `gretl` distingue entre mayúsculas y minúsculas: `x` es diferente a `X`
- puedes encontrar todos los comandos de `gretl` en el botón correspondiente en la barra de herramientas
- en la terminal, puedes escribir `help`

Ejercicio 1

El archivo `food.gdt` contiene dos variables: x e y para una muestra de hogares. La variable y refleja los gastos semanales en alimentos y x mide al renta semanal en múltiplos de \$100.

- 1 Abrir `food.gdt`.
- 2 Analizar la información en la pantalla: tamaño de la muestra, nombres de las variables.
- 3 Usando la gui, poner las dos variables en logaritmos.
- 4 Usando la gui, resumir y y x para todos los hogares de la muestra.
- 5 Usando la gui, resumir y para los hogares con renta inferior a la media.
- 6 Escribir todos los comandos en un fichero script y guardarlo.
- 7 Abrir el script y ejecutarlo.
- 8 Guardar el nuevo fichero de datos con otro nombre.

Ejercicio 2

El archivo BWGHT.csv contiene información sobre el peso al nacer de una muestra de bebés, su sexo y raza, el ingreso de su familia, la educación de sus padres y la cantidad de cigarrillos consumidos diariamente por la madre durante el embarazo (*cigs*).

Escribir todos los comandos necesarios en un script.

- 1 Importar el archivo BWGHT.csv
- 2 Computar el peso medio al nacer de las niñas de la muestra.
- 3 Computar la proporción de madres que no fumaron durante el embarazo.
- 4 Computar el peso medio al nacer de los bebés de madres que no fumaron durante el embarazo.
- 5 Computar el peso medio al nacer de los bebés de madres que sí fumaron durante el embarazo.
- 6 Guardar la nueva base de datos en formato de gretl.
- 7 Cerrar la sesión, abrir el script y correr todos los comandos a la vez.
Funciona!

Comandos básicos para manejo de datos (1/2)

Comandos sobre bases de datos

- `open`: abre un archivo reemplazando el que está abierto
- `smp1`: define el rango de la muestra
- `dataset`: ordena/traspone/compacta/expande... los datos
- `setobs`: declara la estructura de los datos (sección cruzada, series de tiempo, panel)
- `append`: agrega el contenido del archivo a la base que tenemos abierta
- `store`: guarda los datos en un archivo

Comandos básicos para manejo de datos (2/2)

Comandos sobre variables

- `genr`: crea una nueva variable
- `delete`: elimina variables
- `setinfo`: fija atributos de una variable
- `rename`: renombra variables
- `summary`: muestra resumen de las variables
- `freq`: muestra la distribución de frecuencia de las variables
- `print`: muestra los valores de las variables

```
open dataname --www --sheet="name"
--coloffset=# --rowoffset=#
```

- abre una base de datos reemplazando la existente
- --www abre una base de datos del servidor de gretl
- con hojas de cálculo, puedes seleccionar la hoja, la primera columna y la primera fila de los datos
- la primera fila debe contener los nombres de las variables. En caso contrario, el programa automáticamente nombrará a las variables: v1, v2, etc.

```
open "C:\there\mydata.xls" --sheet="mysheet" --coloffset=3
--rowoffset=2
```

- abre la hoja mysheet del archivo de Excel C:\there\mydata.xls
- lee los datos empezando en la cuarta columna y la tercera fila

```
smpl (#start #end | condition --restrict | # --random |  
full) --replace --balanced
```

- *condition* --restrict: restringe la muestra a las observaciones que satisfacen *condition*
- # --random: # casos son seleccionados aleatoriamente
- full: restaura el rango completo de datos
- las restricciones a la muestra son acumulativas: --replace desactiva todas las restricciones previas

Ejemplos (usando Example2.xls)

- smpl YEAR!=1976 --restrict
- smpl EMP > 3 --restrict --replace
- smpl 50 --random

dataset (addobs # | transpose | sortby *varname* | resample # | clear)

- addobs: agrega observaciones al final de la base de datos
- transpose: traspone la base de datos activa
- sortby: ordena por la variable especificada (dsortby: orden descendiente); se puede proveer una lista de variables
- resample: muestreo aleatorio (Construye una nueva base de datos usando muestreo aleatorio con reemplazo de las filas de la base de datos activa. Los datos originales pueden recuperarse usando *smpl full*)
- clear: cierra la base de datos activa

Ejemplos

- dataset sortby EMP
- dataset resample 500
- dataset clear

setobs #freq #start (--cross-section | --time-series |
 --stacked-cross-section | --stacked-time-series)

- #freq representa la frecuencia en datos de serie de tiempos
- en un panel, #freq representa las unidades en un conjunto de secciones cruzadas o los períodos en un conjunto de series de tiempo
- para secciones cruzadas, #freq=1
- #start=1:1 para paneles y 1 para secciones cruzadas
- en series de tiempo, #start es la fecha de inicio

Ejemplos

- setobs 1 1 --cross-section
- setobs 9 1:1 --stacked-time-series

setobs *unitvar timevar* `--panel-vars`

- impone una interpretación de panel
- ordena los datos por valores ascendientes de *unitvar*

Ejemplo

- setobs *unit YEAR* `--panel-vars`

append *newdata* --time-series

- abre un archivo y lo agrega al contenido de la base abierta
- Primer caso: observaciones adicionales para las variables existentes (los nombres deben coincidir)
- Segundo caso: nuevas variables (número de observaciones compatible)
- Tercer caso: agrega una serie de tiempo en un panel

Primer caso

- open C:\there\Example2.xls --sheet="first100"
- append C:\there\Example2.xls --sheet="moreunits"
- agrega la hoja *moreunits* de C:\there\Example2.xls al final de la base.

Segundo caso

- append C:\there\Example2.xls --sheet="wages"
- agrega la hoja *wages* de C:\there\Example2.xls: nueva variable (WAGE)

append

Tercer caso

- Tenemos un panel y queremos agregar una variable en forma de serie de tiempo. Por ejemplo, queremos agregar los datos de precios al consumo de modo de deflactar las variables nominales de la base de datos.
- abrimos el archivo: `C:\there\Example2.xls --sheet="first100"`
- imponemos que sea un panel: `setobs unit YEAR -panel-vars`
- `append C:\there\Example2.xls --sheet="cpi"`

```
store datafile [varlist] --gzipped --overwrite
```

- por defecto los datos se guardan en formato gretl
- también se exportan a csv (variables separadas por comas) (usando --csv) y muchos otros formatos

```
store C:\there\mydata.gdt
```

- guarda los datos en C:\there\mydata.gdt

[genr] *newvar = formula*

- una fórmula es una función de variables de la base
- el rango sobre el que se aplica la fórmula depende de la muestra activa
- operadores aritméticos: ^, *, /, +, -
- operadores binarios: ! (negación), && (y), || (o), >, <, =, >=, <=, !=
- en Help/Function Reference hay funciones ya escritas (como *abs* o *sqrt*)

Ejemplos

- `genr y = 3 + 2 * x1 + 5 * x2`
- `D1976 = (YEAR = 1976)`
- `genr avgy = mean(y)`

```
delete [ varlist ] --db
```

- elimina las variables listadas
- si no se especifica ninguna *varlist* elimina la última variable de la base (la que tiene un número más alto)
- cuidado! desde la consola no pide confirmación para eliminar una variable.

```
setinfo varname -d "thislabel" -n "thisname" --discrete  
--continuous
```

- `-d "thislabel"`: *thislabel* pasa a ser la etiqueta que describe a la variable
- `-n "thisname"`: *thisname* se utiliza en lugar del nombre de la variable en los gráficos.
- `--discrete`: marca a la variable como discreta (por defecto las variables son continuas)

Ejemplos

- `setinfo bwght -d "Peso al nacer" -n "Peso"`
- `setinfo cigs --discrete`

rename *varname newname*

- cambia el nombre de la variable
- los nombres deben tener un máximo de 15 caracteres
- deben comenzar con una letra
- pueden contener solamente letras, números y el carácter de subrayado.

Ejemplo en `bwght.gdt`

- `rename bwght peso_nacer`

summary [*varlist*] --simple --by=*byvar*

- muestra un resumen de estadísticos para las variables en *varlist*
- si *varlist* se omite, muestra el resumen para todas las variables
- --simple: solamente muestra la media, los valores mínimo y máximo y el desvío estándar
- --by=*byvar*: se muestran los estadísticos para las sub-muestras definidas por los valores de la variable *byvar* (que debe ser discreta)

Ejemplo en bwght.gdt

- summary bwght
- summary bwght --simple
- summary bwght --simple --by=male
- summary bwght --simple --by=parity

print [*varlist*] `--byobs` `--no-dates`

- muestra los valores de las variables en *varlist*
- si no se especifica ninguna variable, muestra todas
- `--byobs`: los datos se muestran por observación
- se pueden imprimir frases o comentarios
- se puede restringir la muestra

Ejemplos

- `print bwght male --byobs`
- `print "Aquí va un comentario"`
- `print bwght ; male --byobs` : muestra el peso al nacer por observaciones para los varones

Breve comentario sobre gráficos

- `gnuplot yvars xvars`: grafica y en función de x
 - desde el script por defecto guarda el gráfico como un archivo `.plt`
 - `--output=filename`, permite elegir el nombre y el tipo de archivo
 - `--output=display`, muestra el gráfico en lugar de guardarlo
- `scatters yvar ; xvarlist`: gráficos de dos variables (varios a la vez)
- usando gui Variable/Frequency Distribution: histogramas