



Copyright © 2015 Banco Interamericano de Desarrollo. Esta obra está bajo una licencia Creative Commons IGO 3.0 Reconocimiento-No Comercial-Sin Obra Derivada (CC-IGO BY-NC-ND 3.0 IGO) (<http://creativecommons.org/licenses/by-nc-nd/3.0/igo/legalcode>) y puede ser reproducida para cualquier uso no-comercial otorgando crédito al BID. No se permiten obras derivadas.

Cualquier disputa relacionada con el uso de las obras del BID que no pueda resolverse amistosamente se someterá a arbitraje de conformidad con las reglas de la CNUDMI. El uso del nombre del BID para cualquier fin que no sea para la atribución y el uso del logotipo del BID, estará sujeta a un acuerdo de licencia por separado y no está autorizado como parte de esta licencia CC-IGO.

Notar que el enlace URL incluye términos y condicionales adicionales de esta licencia.

Las opiniones expresadas en esta publicación son de los autores y no necesariamente reflejan el punto de vista del Banco Interamericano de Desarrollo, de su Directorio Ejecutivo ni de los países que representa.

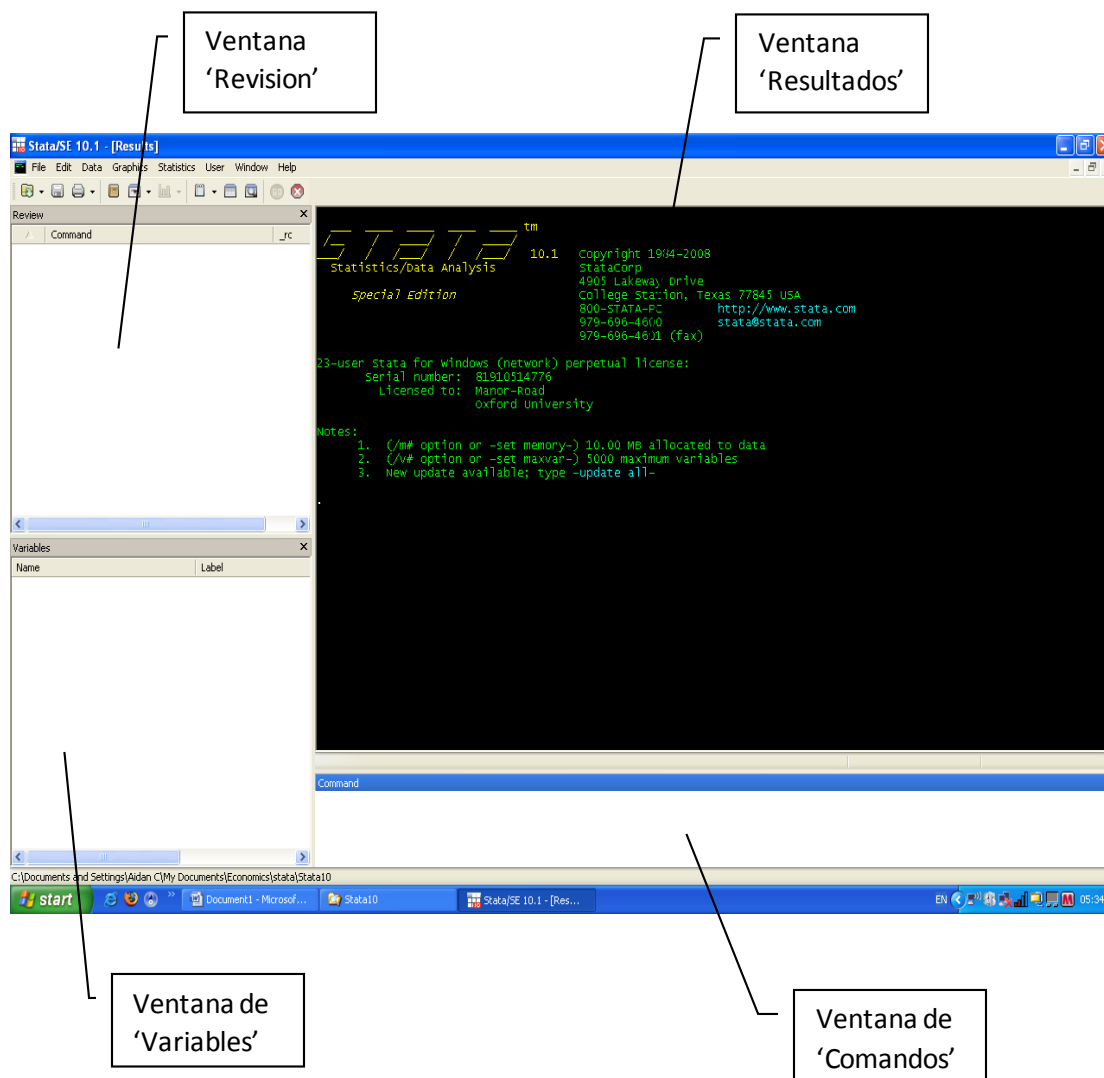
## Empezando a trabajar con STATA

STATA es un paquete estadístico muy utilizado por los economistas y cuenta con características que lo hacen especialmente adecuado para realizar el análisis que requiere una evaluación de impacto. Este documento pretende servir de guía de iniciación a todos aquellos que quieran familiarizarse con las características básicas y con el diseño del programa. Este documento le ayudara a navegar por los elementos necesarios para realizar un análisis, incluyendo:

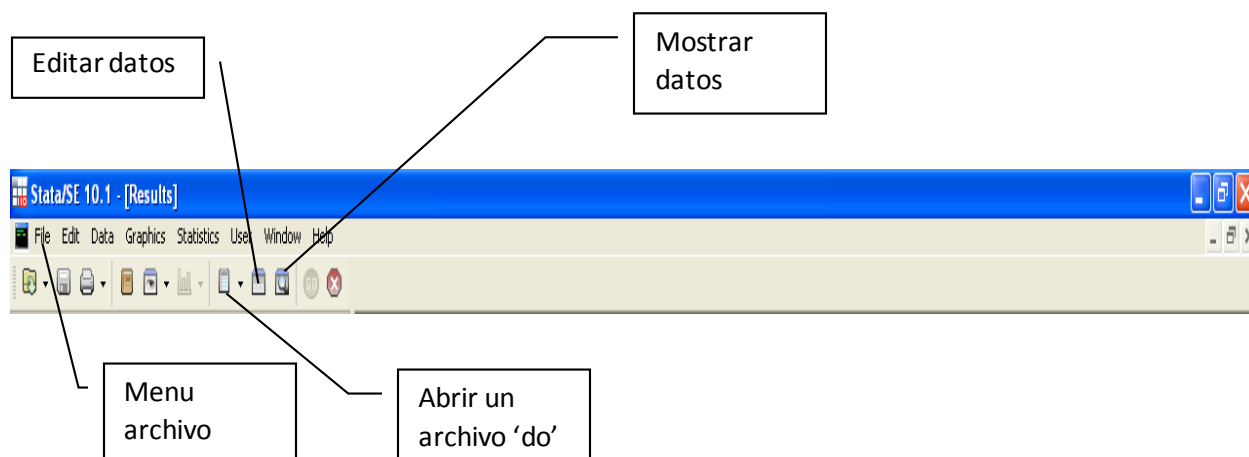
1. Utilizar el comando de ayuda 'Help'.
2. Como cargar bases de datos.
3. Abrir y trabajar con archivos 'log' y 'do'.
4. Los comandos básicos para realizar el análisis de datos.

Buena suerte!

### Orientación:



- La ventana de variables muestra las variables de la base de datos. Cuando se carga una base de datos en STATA encontraras una lista con el nombre de las variables y una breve descripción en esta ventana.
- La ventana de comandos permite escribir instrucciones para que STATA las realice en la base de datos.
- La ventana de resultados muestra los resultados obtenidos a las instrucciones que le damos a STATA.
- La ventana de revisión guarda un histórico de todos los comandos o instrucciones utilizados en cada sesión.



- El icono para mostrar los datos te permite ver la base de datos que se ha cargado en STATA, pero no permite hacer ningún cambio a los datos.
- El icono para editar datos permite ver la base de datos y además permite realizar cambios en los datos.
- El botón para abrir un archivo 'Do' abre un archivo de texto que sirve para escribir instrucciones a STATA. Es una manera alternativa de dar instrucciones a STATA a la ventana de comandos. La ventaja de escribir los comandos en un archivo 'Do' es que puedes escribirlos todos de una vez y puedes guardarlos para revisarlos o modificarlos en el futuro. Como veremos estas son unas propiedades muy útiles cuando se tiene que analizar una base de datos.

## Consiguiendo Ayuda:

Hay varias maneras de conseguir ayuda cuando no se conoce un comando o cuando no se sabe que comando es más conveniente utilizar en STATA. El propio programa tiene un directorio de ayuda que te permite conocer que operaciones hace un comando y como utilizarlo. Para esto tienes que escribir:

**help comando**

Por ejemplo, si quieres saber para qué sirve el comando *summarize* tendría que escribir:

**help summarize**

Por otra parte, si lo que te ocurre es que sabes que es lo que quieres hacer, pero no conoces el nombre del comando que tienes que utilizar, puedes utilizar la opción 'help' del icono de 'archivo' y luego seleccionar 'search' ('buscar' en español).

Ahora, intenta buscar un comando que te ayude a describir una base de datos utilizando la opción 'search';. ¿Qué resultados obtienes?

## Abriendo un archivo 'Do':

Como hemos dicho, los archivos 'Do' cuentan con unas propiedades muy útiles cuando estamos realizando análisis con STATA, así que será la herramienta en la que nos basaremos para dar instrucciones a STATA. Más concretamente es donde escribiremos nuestro programa, es decir, la sucesión de comandos que nos permitirán hacer el análisis de la base de datos. De esta manera guardaremos toda la información sobre cómo hemos modificado la base de datos, y como hemos realizado el análisis y podremos repetir el análisis en el futuro y modificarlo si es preciso.

Para abrir un archivo 'Do' se puede utilizar el icono indicado anteriormente: Abrir un archivo 'Do'. Una vez abierto puedes empezar a escribir las instrucciones o comandos que vas a utilizar. Uno de los primeros comandos es el que le indica que base de datos debe utilizar y cargarla en el programa. Hay distintas maneras de cargar una base de datos en el programa. En este caso utilizaremos el comando 'use'. Para utilizar este comando, se debe haber guardado previamente la base de datos en formato 'dta' en el ordenador. En el ejemplo que acompaña este documento de iniciación la base de datos se llama 'evaluation.dta' y ya tiene el formato correspondiente (lo sabemos porque tiene la extensión 'dta'). Así que tenemos que decirle a STATA en qué carpeta se encuentra la base de datos y el nombre de la misma. Por ejemplo:

**use 'C:\Documents and Settings\DoH\Desktop\STATA workshop\evaluation.dta'**

Si los datos se encuentran en formato Excel, también se pueden cargar en STATA, guardándolos previamente con la extensión 'csv' y utilizando el comando 'insheet'. Utilizando la ayuda de STATA puedes ver cómo funciona. Alternativamente se pueden transformar los datos de 'csv' a 'dta' utilizando un programa que se llama 'Stat-transfer'.

Cuando vamos a estar utilizando una o varias bases de datos a lo largo de un programa nos puede convenir utilizar unos comandos de memoria temporal llamados 'local' o 'global' que tiene STATA. Estos van a recordar la carpeta en que se encuentra la base de datos y así no tenemos que escribirlo cada vez que queramos cargar o guardar la base de datos.

En nuestro ejemplo podemos usar el comando 'global' de la siguiente manera:

**global data = 'C:\Documents and Settings\DoH\Desktop\STATA workshop\evaluation\_dta'**

Ahora para acceder a la carpeta donde se encuentran nuestros datos solo tendremos que escribir '\$data' ('data' es como hemos llamado a este 'atajo' y tenemos que poner el signo '\$' delante para indicar que nos estamos refiriendo a este 'atajo'). Por ejemplo, ahora, para cargar los datos podríamos substituir el comando que utilizamos anteriormente por:

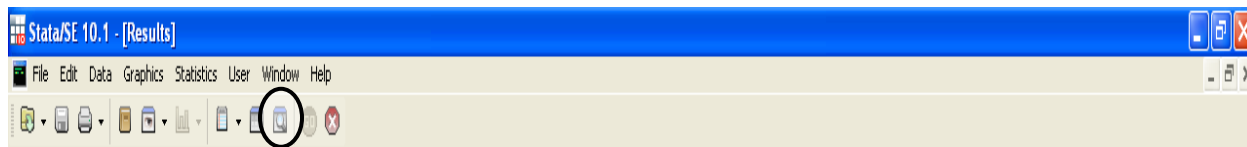
**use '\$data\evaluation.dta'**

Una vez hemos cargado la base de datos las variables con sus nombres correspondientes deberían aparecer en la ventana de variables en el lado inferior izquierdo de la pantalla.

## Explorando los datos:

Ahora que hemos cargado los datos lo primero que queremos hacer es tener una panorámica general de los mismos.

Los nombres de las variables aparecerán en la ventana de 'variables' como hemos dicho, pero si queremos acceder a los datos reales debemos utilizar el icono de 'Mostrar datos' (como se muestra abajo) o darle a STATA la instrucción 'browse'. Si lo haces veras que abre una nueva ventana con el valor de tus datos que se llama 'data browser'.



Esta ventana te sirve para observar los valores reales de tus datos, pero no te permitirá manipularlos. Para cambiar los valores numéricos de algunos de los datos o cambiar el nombre de las variables, puedes utilizar el botón 'editar datos' ('data editor', ver abajo) o escribir el comando 'edit'.



Ahora aparecerá la base de datos y podemos cambiar el valor de las variables. Hay otras maneras de cambiar el valor o el nombre de las variables utilizando comandos de STATA, y muchas veces esta es la manera más recomendable de hacerlo porque queda constancia escrita del cambio en el programa. Para ello se utilizan comandos como 'rename' o 'replace'.

Fíjate que mientras tienes la ventana de ‘ver’ o ‘editar’ datos abierta, no podrás utilizar la ventana de comandos. Necesitas cerrar la vista de la base de datos para poder utilizarla.

Esta es una manera de mirar a los datos directamente, pero como normalmente se utilizan bases de datos con cientos y miles de observaciones, generalmente resulta útil observar los estadísticos descriptivos.

Primero, nos gustaría obtener la información básica de todas las variables incluidas en la base de datos. Para ello utilizaremos el comando ‘**describe**’.

Si queremos obtener los estadísticos descriptivos principales de la base de datos (no simplemente los nombres) podemos utilizar el comando ‘**summarize**’. Teclea estos dos comandos en la ventana de comandos y observa la distinta información que te ofrecen.

Al escribir ‘**summarize**’ veras que tienes información sobre el número de observaciones, la media, la desviación típica y el valor mínimo y máximo de cada variable.

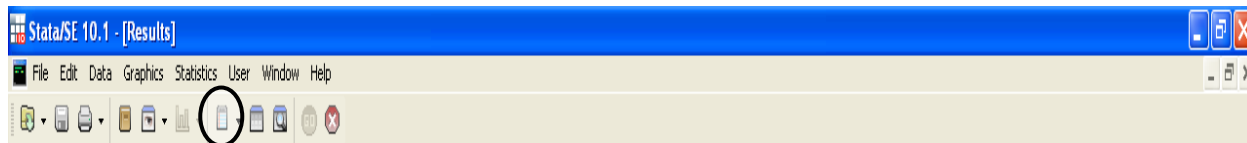
Si queremos centrarnos en un conjunto particular de variables podemos simplemente escribir el nombre de las variables que queremos que nos muestre a continuación. Por ejemplo:

**summarize ophe round**

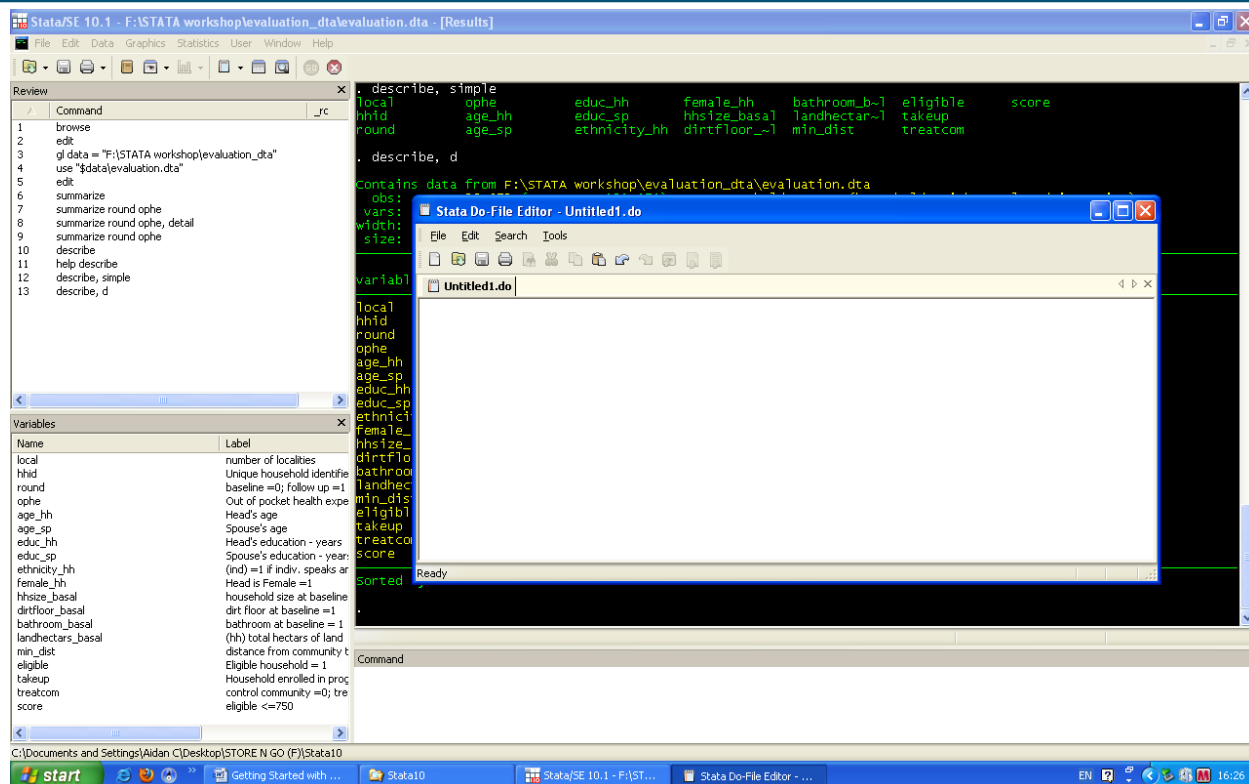
## Usando un archivo ‘Do’:

Como hemos dicho anteriormente, en lugar de escribir los comandos uno a uno en la ventana de comandos, alternativamente, podemos utilizar un archivo ‘Do’. Este es un documento de texto donde podremos escribir todos los comandos que queramos hacer y pedirle a STATA que los realice de una sola vez. De esta manera tendremos un registro de todas las operaciones que estamos haciendo y y hacemos un error (como por ejemplo eliminar accidentalmente una variable) podemos volver atrás, y corregirlo, y no perder información relevante.

Para abrir un archivo ‘Do’ podemos utilizar el icono que se encuentra en la esquina superior izquierda de la pantalla ‘New Do-File Editor’.



Esto abrirá una pantalla en blanco en la que podremos escribir nuestro programa.



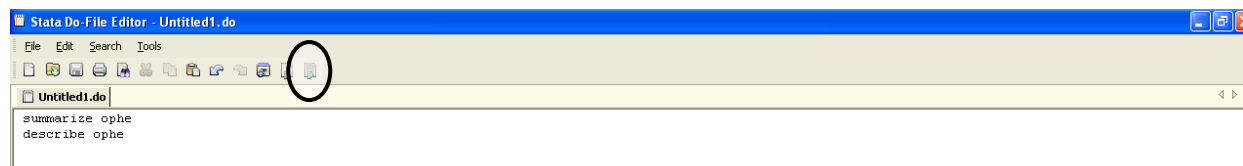
Deberá aparecer algo como en la ilustración de arriba. Cada línea que escribamos en esta hoja STATA lo interpretará como una instrucción diferente.

Para empezar a utilizarla puedes escribir en ella:

**summarize ophe**

**describe ophe**

A continuación, para indicar a STATA que quieres que realice las instrucciones que hemos escrito, haz click en el icono 'do' que aparece en la esquina superior izquierda de la pantalla.



Mira la ventana de resultados de STATA. ¿Qué ha ocurrido?

Deberías ver los mismos resultados que obtuvimos anteriormente cuando estábamos utilizando la ventana de comandos y encontrar la información de 'describe' y 'summarize' de la variable 'ophe'.

A veces quieres acompañar las instrucciones que le das a STATA de cierto texto o explicación. Si lo escribes directamente, STATA no reconocerá los comandos y te dará un error. Por eso es bueno que

conozcas el siguiente comando ‘\*’. Todo el texto o las instrucciones que escribas después de un asterisco, STATA lo ignorará. Por ejemplo ahora puedes escribir:

**\*Ahora voy a describir la variable ‘ophe’**

**describe ophe**

**summarize ophe**

¿Qué ocurre en la ventana de resultados?

Prueba a poner ahora el asterisco antes de las instrucciones de summarize y describe:

**\*describe ophe**

**\*summarize ophe**

¿Qué ocurre ahora en la ventana de resultados?

## Guardando los resultados en un archive ‘Log’:

Hasta ahora hemos estado observando los resultados en la ventana de resultados de STATA, pero algunas ocasiones queremos guardar todos los resultados del análisis en un documento aparte que podamos guardar, revisar, utilizar en algún documento o enviar a algún compañero.

Hay varias maneras de guardar los resultados de STATA en distintos archivos. En este caso vamos a ver una manera en que todo el análisis que realicemos, tanto instrucciones como resultados, quedará guardado en una hoja de texto. Para esto utilizaremos el comando ‘log’.

Para esto lo primero que hacemos es darle una ubicación en alguna carpeta al documento que queremos generar. Por ejemplo una ubicación podría ser la carpeta en la que teníamos los datos. En mi ejemplo:

**‘C:\Documents and Settings\DoH\Desktop\STATA workshop\evaluation\_dta’**

Una vez decidida solo tenemos que escribir el comando, seguido de la ubicación de la carpeta y de un nombre que le queramos dar al documento, por ejemplo ‘miprimerlog’ seguido de la extensión ‘txt’:

**log using ‘C:\Documents and Settings\DoH\Desktop\STATA workshop\evaluation\_dta\miprimerlog.txt’**

Recuerda que anteriormente creamos una memoria ‘global’ para dirigirnos a esa carpeta. Eso nos va a permitir volver a utilizarla y en vez de escribir toda la dirección de la carpeta de nuevo, podemos simplemente escribir:

**log using ‘\$data\miprimerlog.txt’**



Esto abrirá automáticamente un archivo de texto en la carpeta indicada en la que se recogerán todos los resultados e instrucciones que se escriban a continuación. Cuando se termine el programa se le debe indicar a STATA que debe cerrar este archivo mediante el comando:

**log close**

Hay un par de cosas más que te conviene saber sobre este comando para evitar que STATA te reporte problemas.

Antes de utilizar 'log using' escribe la instrucción '**capture log close**'. Esto cerrará automáticamente los archivos 'log' y evitara que STATA te reporte un error cuando el programa no corra hasta el final, y por tanto no se cierre el archivo log. Además, cuando corremos el programa varias veces, queremos que el contenido archivo de texto log se reemplace por los nuevos resultados cada vez que lo corremos. Para ello hay que añadir '**replace text**' al final del archivo.

En conclusión, deberás escribir en tu programa:

**capture log close**

**log using '\$data\myfirstlog.txt', replace t**

**\*Summary for the variable on out of pocket expenditures (ophe)**

**summarize ophe**

**describe ophe**

**log close**

¿Qué ha ocurrido?

Aparentemente nada nuevo, pero si miramos en la carpeta donde le hemos indicado a STATA que genere la hoja de resultados, tendremos que encontrar un documento de texto llamado 'miprimerlog.txt'. Ábrelo para ver qué hay escrito.

## Más comandos de STATA:

Ahora que ya tenemos las herramientas básicas para empezar a trabajar en STATA, vamos a aprender algunos comandos que nos resultaran útiles para analizar la base de datos. Recuerda que siempre puedes revisar cualquier comando de STATA utilizando la ayuda y escribiendo '**help comando**' para obtener información detallada de cada uno de ellos.

### MACROS

**global label1 '...':** ya hemos visto uno de los usos del comando 'global' para memorizar la dirección de las carpetas. Aquí vemos otro uso en el que le podemos pedir que nos recuerde con una 'etiqueta' el nombre de un grupo de variables. En este caso asignara a la etiqueta que hemos llamado 'label 1' los valores que se encuentren entre las comas.

**global label1 variable1 variable2 variable3...:** Por ejemplo, ahora STATA asignara las variables que hemos llamado 'variable1 variable2 variable3' a la etiqueta llamada 'label1'. Si le damos esta instrucción a STATA, en el futuro solo necesitaremos escribir el nombre de la etiqueta (precedido del signo '\$') para utilizar ese grupo de variables. Por ejemplo, si queremos ver los estadísticos descriptivos podemos escribir: **summarize variable1 variable2 variable3** o alternativamente **summarize \$label1**.

### DESCRIPCIÓN DE LA BASE DE DATOS

**summarize variable1 variable2:** te da los estadísticos descriptivos básicos de las variables indicadas.

**browse:** abre una tabla para que veas directamente las variables.

**edit:** abre una tabla en la que puedes manipular las variables .

**describe:** te da el nombre y la descripción de las variables de la base de datos.

**tabulate variable1 variable2:** te muestra una tabla de frecuencias con las variables indicadas (mínimo una y máximo dos).

### MODIFICACIÓN DE LOS DATOS

**keep/drop ...:** estos comandos sirven para guardar ('keep') o eliminar('drop') tanto variables como observaciones, según se le especifique. Más adelante veremos algunos ejemplos.

**If(si), &(y), |(o), ==(igual a), > (mayor que), < (menor que):** estos operadores sirven para imponer condiciones y suelen utilizarse en combinación de otros comandos. Por ejemplo, si queremos trabajar solo con un tipo de observaciones podemos combinar los dos comandos que acabamos de aprender y quedarnos en nuestra base de datos solo con los valores que nos interesan. Por ejemplo, en nuestra base de datos existen dos rondas representadas en la variable 'round' que toman los valores 0 y 1. Si solo nos interesase trabajar con las observaciones de la ronda '1' podemos escribir:

**keep if round == 1**

Esto eliminara todas las observaciones de nuestra base de datos de las rondas distintas a la 1 (en nuestro caso eliminara todas las observaciones de la ronda 0). Alternativamente podríamos haber hecho:

**drop if round ==0**

y hubieramos obtenido el mismo resultado.

Podemos querer ser más específicos y, por ejemplo, solo guardar observaciones que sean menor que un cierto valor. En nuestra base de datos tenemos un índice de pobreza representado en la variable 'score'. Si quisiéramos trabajar con las observaciones de la ronda 1 que sean menores que cierto valor del índice, podríamos escribir:

**keep if round == 1 & score < 600**

Alternativamente, si queremos guardar todas las observaciones tanto de la ronda 1 que sean menores de un cierto valor utilizaríamos el comando:

**keep if round == 1 | score < 600**

**+, -, \*, /:** Estos operadores matemáticos básicos también funcionan en STATA para sumar, restar, multiplicar o dividir.

**generate variableX:** este comando crea una nueva variable llamada 'variableX'. Esta puede ser una combinación de otras variables en la base de datos. Por ejemplo podemos generar una variable que sea la multiplicación de una variable Y y una variable Z:

**generate variableX = variableY\*variableZ**

**replace variableX = ...:** También podemos reemplazar las variables que ya existen por nuevas variables. Por ejemplo si queremos reemplazar todas las observaciones de la variable X por la suma de otras dos variables podemos escribir: **replace variableX=variableY+variableZ**

### REGRESIONES

**regress variable1 variable2 variable3:** este comando estima una regresión lineal donde la primera variable (en este caso variable1) es la variable dependiente y las que le siguen (variable2 variable3) son los regresores.

**probit variable1 variable2 variable3..:** esto estima una regression 'probit' donde la primera variable (en este caso variable1) es la variable dependiente y las que le siguen (variable2 variable3) son los regresores.

**predict newvariable:** Este comando guarda los resultados de las predicciones probit anteriores en una nueva variable (que en este ejemplo hemos llamado 'newvariable').

### GRÁFICOS

**scatter variable1 variable2:** Produce un mapa de puntos de las variables que se indiquen a continuación (en este ejemplo variable1 y variable2).

**twoway:** este commando permite hacer gráficos con varias capas.

**kdensity variable1:** calcula and y hace un grafico con la densidad 'kernel' de la variable. Es decir, que la distribución estimada (por ejemplo, una campana de Gauss para la altura de los niños) se calcula y se hace un grafico.

**ttest variable1, by (variable2):** este comando nos permite hacer un test de medias y saber si la diferencia es significativamente distinta de 0. Concretamente comparara las medias de la variable1 basándose en los dos grupos definidos en la variable2. Resulta muy útil al hacer evaluación de impacto. Por ejemplo, podemos comparar las alturas de un conjunto de niños divididos en dos subgrupos, aquellos que tomaron pastillas de crecimiento y aquellos que no. Si tomar las pastillas de crecimiento

fuera el tratamiento que queremos evaluar podríamos poner **ttest height, by (treatment)**. Esto nos diría si existen diferencias en las alturas de los dos grupos, y en caso de que exista si son significativas.