

Notas de clase – Taller de Stata¹

Clase 1 – Introducción, importación y descripción de datos

Contenido

1. Interfaz
2. Archivos
3. Identificar el formato de la base de datos
 - 3.1. Revisar extensión
 - 3.2. Type
4. Importar datos de otros formatos a *Stata*
 - 4.1. Archivo de Excel
 - 4.2. Archivos de texto
 - 4.3. Archivos de otros programas
 - 4.4. Manualmente
5. Compatibilidad entre versiones de *Stata*
6. Comandos para bases de datos
 - 6.1. Comprimir bases de datos
 - 6.2. Ordenar bases de datos

[Estas notas se diseñaron con base en la versión 16 de *Stata*, conforme el software cambie las notas deben ser actualizadas].

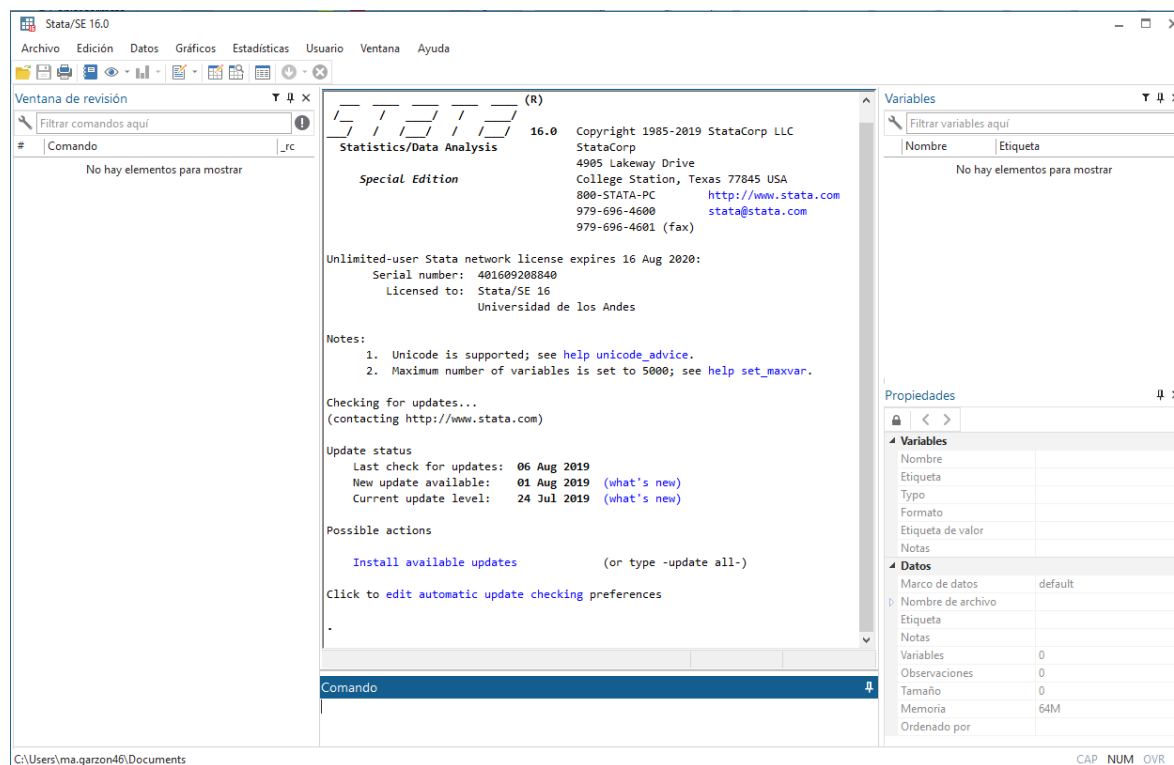
¹ Estas notas están basadas en la guía desarrollada previamente para este curso por Rodrigo Azuero Melo, Nicolás de Roux, Luis Roberto Martínez, Román Andrés Zárate y Santiago Gómez Echeverry.

Stata es un software estadístico empleado frecuentemente para gestionar la información de bases de datos, realizar simulaciones, hacer análisis estadísticos y diseñar gráficas. Es ampliamente usado en las áreas de economía, ciencias políticas, sociología, y epidemiología entre otras. Estas notas de clase presentarán cómo usar los comandos y funciones básicas de *Stata* para la investigación en economía.

1. Interfaz

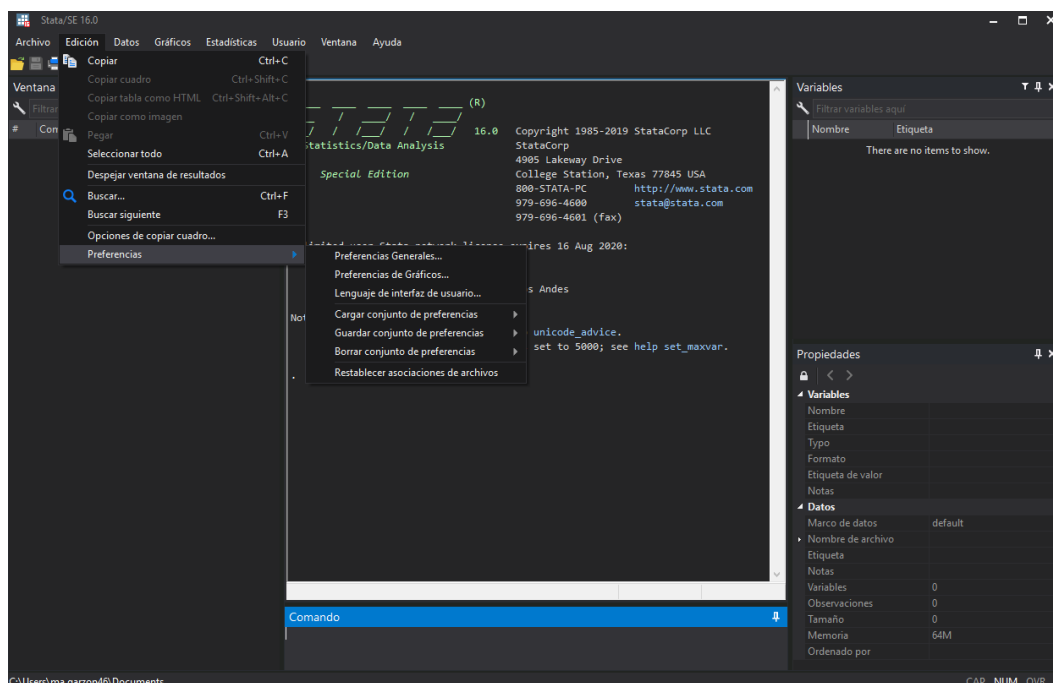
La interfaz de usuario que maneja *Stata* por defecto está compuesta por un menú principal, una barra de herramientas y cinco diferentes ventanas: (i) comandos, (ii) resultados, (iii) variables, (iv) propiedades y (v) revisión. Adicionalmente, hay tres ventanas que se usan para llevar el registro del trabajo, (vi) datos, el (vii) *do-file* y el (viii) visor que se emplea para ver los *log-files*. A continuación, se explicará en detalle la funcionalidad de cada una de estas ventanas.

Figura 1: Interfaz de usuario de *Stata* 16



En el menú Edición, el submenú Preferencias y la opción Preferencias Generales se puede cambiar el tema de la interfaz, como se muestra en la Figura 2:

Figura 2: Tema oscuro en *Stata 16*



1.1. Ventana de comandos

La ventana de comandos es aquella que se encuentra en la parte inferior central de la interfaz de usuario. En esta el usuario puede darle órdenes directas a *Stata* según su conveniencia simplemente escribiendo el comando con su sintaxis particular y presionando la tecla *enter*. Adicionalmente, varios de los comandos que vienen incluidos en el software tienen abreviaciones que pueden ser usadas en esta ventana y facilitan el proceso de programación² (e.g. *h* es la abreviación del comando *help*). En estas notas se hará referencia constantemente a los comandos por su abreviación.

Si bien cada uno de los comandos puede variar en su sintaxis, existe una estructura que todos tienen en común: primero se digita la orden que solicitada al software, lo que va después de una coma (,) es considerado como una opción. Vale la pena mencionar que varios comandos tienen opciones por defecto, que pueden ser cambiadas si se especifica.

² Estas abreviaciones pueden ser consultadas en el *help-file* de cada uno de los comandos al ejecutar *help* y el nombre del comando.

Taller de Stata
Clase 1 - Introducción, importación y descripción de datos
Miguel Garzón Ramirez, Cristhian Acosta Pardo
Facultad de Economía, Universidad de los Andes

1.2. Ventana de resultados y *log-file*

En esta ventana aparecen todas las entradas y salidas del proceso de programación. Entiéndase esto como las órdenes que se le dan a *Stata* a través de los comandos (entradas) y los textos, tablas o resultados que presenta el programa ante estas órdenes (salidas). El *log-file* puede ser visto como una bitácora, en la cual se registra todo lo que se imprime en la ventana de resultados. Este archivo puede ser bastante útil cuando se tienen bastantes líneas de programación y estos arrojan bastante información en forma de tablas o textos.

Finalmente, cuando se quiera correr un comando sin que las salidas aparezcan en la ventana de resultados se puede escribir previo al comando que se desea utilizar, en la misma línea, el comando quietly.

1.3. Ventanas de variables y de propiedades

Esta ventana muestra todas las variables que contiene la base de datos que se encuentra abierta en el momento, junto con sus etiquetas³. Solo si hay una base de datos abierta se podrá ver alguna variable. Los usuarios pueden presionar dos veces el nombre de las variables para incluirlas en la ventana de comandos.

4

La ventana de propiedades está dividida en dos partes. La primera muestra las propiedades de una variable cuando esta es seleccionada en la ventana de variables. Por propiedades de las variables se entiende su: (i) nombre, (ii) etiqueta, (iii) tipo, (iv) formato, (v) etiquetas de valor, y (vi) notas. La segunda parte de la ventana muestra las propiedades de la base de datos, lo que incluye: (i) nombre del archivo, (ii) etiqueta, (iii) notas, (iv) número de variables, (v) número de observaciones, (vi) tamaño de la base, (vii) memoria RAM⁴ que ocupa, y (viii) variable por la que se encuentra ordenada la base.

1.4. Ventana de revisión

Al interior de esta ventana se pueden encontrar todos los comandos que se hayan empleado hasta el momento. Esta ventana funciona como un historial de programación. Sin embargo, en los casos en los que se corran los comandos desde un *do-file* estos no aparecerán en el historial.

³ Son anotaciones descriptivas de las variables, son utilizadas en títulos de tablas y gráficas

⁴ *Random Access Memory*, es la memoria de trabajo de las computadoras (y en plural el mejor álbum de Daft Punk según los autores de estas notas).

En vez de esto el historial mostrará el comando do y la ubicación y nombre del archivo *do-file* que se está corriendo. Si se presiona una vez uno de los comandos del historial este aparecerá en la ventana de comandos, si se presiona dos veces este se ejecutará (y por tanto aparecerá en la ventana de resultados de nuevo).

1.5. Ventana de datos

Sobre la barra de herramientas de *Stata* hay dos iconos con los cuales se puede acceder a la base de datos que está cargada. Estos dos iconos son aquellos que tienen una cuadrícula. El icono con la cuadrícula y una lupa superpuesta a esta abre el visor de datos. En esta ventana se pueden ver cada una de las observaciones para cada una de las variables en la base de datos. Sin embargo, en esta ventana no se pueden modificar los datos. El icono con la cuadrícula y el lápiz superpuesto a esta abre el editor de datos. En esta ventana se puede ver y modificar el contenido de la base de datos. También se puede abrir esta ventana en las dos modalidades ya mencionadas con los comandos browse y edit, respectivamente.

Es recomendable modificar la base de datos únicamente a través de comandos ejecutados a través del *do-file*. Si existe alguna situación en la cual se deba emplear el editor de datos, hágalo con la mayor rigurosidad y meticulosidad con el fin de evitar posibles errores modificando la base de datos de forma indeseable e irreversible.

1.6. *Do-file*

El *do-file* es el archivo en el que se pueden registrar todas las ordenes que se le van a dar al software. Es el equivalente a lo que en varios lenguajes de programación se conoce como *script*. Para escribir un nuevo *do-file* o escribir uno que ya esté escrito debe buscar en la barra de herramientas el icono con un papel y lápiz, el cual arroja el mensaje “Abrir nuevo editor de archivos Do”. Para correr los comandos de un *do-file* se seleccionan los comandos y luego se presiona la opción de “Ejecutar” al final de la barra de herramientas del *do-file*. Esta acción también se puede hacer de manera más rápida presionando la combinación de letras Ctrl + D⁵.

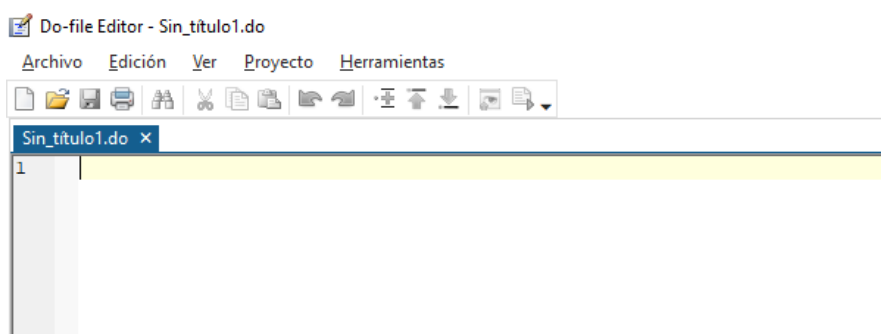
Normalmente los archivos en los que se escribe una serie de comandos u órdenes pueden llegar a ser muy complejos y muy extensos. Dado esto los programadores suelen emplear comentarios para saber qué es lo que se está haciendo con el código. En *Stata* se puede escribir comentarios

⁵ Este método abreviado aplica para el sistema operativo Windows. Para el caso de Mac la abreviación es Cmd + Shift + D
Taller de Stata

empleando un asterisco (*) o un doble *slash* o barra diagonal (/). Adicionalmente pueden emplearse las barras diagonales y el asterisco para escribir comentarios en varias líneas (i.e. todo lo que vaya entre el símbolo “/*” y el símbolo “*/” será un comentario para *Stata*⁶).

Los *do-files* son fundamentales para la adecuada programación en *Stata* ya que permiten mantener la trazabilidad de lo que se quiere hacer y pueden ser fácilmente compartidos entre usuarios. Todas las clases emplearemos un *do-file* guía y es en este tipo de archivo que se pedirán todos los trabajos a lo largo del semestre.

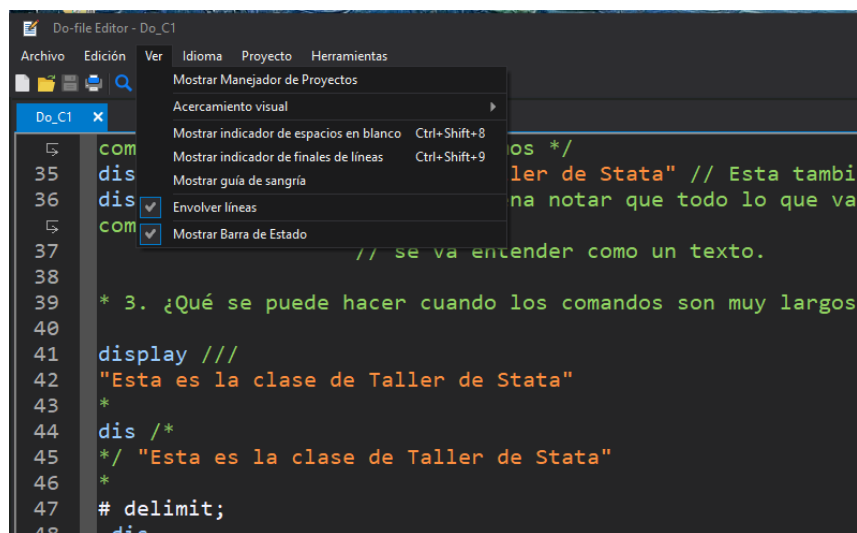
Figura 3: Editor de *do-files*



6

Para ver una línea completa de código active la opción “Envolver líneas” en el menú ver, como se muestra en la Figura 4.

Figura 4: Opciones de visualización del *do-file*



⁶ Esto solo puede emplearse mediante *do-files*. No tiene sentido incluirlo en la ventana de comandos directamente.
Taller de Stata

2. Archivos

Stata permite manejar diferentes tipos de archivos. Esto facilita el trabajo de programación y el intercambio de información. Los archivos compatibles con el software suelen visualizarse en diferentes ventanas dependiendo de su tipo. La [Tabla 1](#) muestra los diferentes tipos de archivos junto con su terminación específica.

Tabla 1: Tipos de archivos

Archivo	Tipo de archivo
Base de datos	.dta
<i>Do-file</i>	.do
<i>Log-file</i>	.log
Archivo de ayuda	.hlp
Archivos de extensión	.ado ⁷

Todos los archivos compatibles con *Stata* pueden ser abiertos directamente desde el explorador de *Windows*, esto abre una nueva sesión cada vez. Los archivos también pueden abrirse empleando las opciones del menú principal o los menús de las ventanas correspondientes a cada tipo de archivo, esto permite abrir cada archivo en la misma sesión de *Stata*.

7

3. Identificar el formato de la base de datos

En todas las sesiones de *Stata* es necesario fijar el directorio de trabajo. En otras palabras, introducir la ubicación de la carpeta donde se encuentran la base de datos. Para esto, se usa el comando `cd`, así:

Ej. 1 \rightarrow `cd "C:\Users\Uniandes\Facultad de Economía\Taller de Stata"`

Una vez se fija el directorio es posible *identificar* el formato de la base de datos, por ejemplo, al revisar la extensión del archivo desde *Stata*. Es importante conocer el formato del archivo debido a que el comando de importación de datos cambia conforme lo hace la extensión del archivo. La carpeta Mis Documentos es el directorio de trabajo predefinido en *Stata*. Se puede ver el directorio de trabajo actual en la barra inferior de la ventana principal. Puede preguntarle a *Stata* sobre el directorio de trabajo actual con el comando `"cd"`.

⁷ Este tipo de archivos se estudiará con material complementario al finalizar el curso.
Taller de Stata
Clase 1 - Introducción, importación y descripción de datos
Miguel Garzón Ramírez, Cristhian Acosta Pardo
Facultad de Economía, Universidad de los Andes

3.1. Revisar extensión

La extensión del archivo señala el tipo de formato. Por ejemplo, las bases de datos en formato de *Stata* tienen la extensión (.dta).

En *Stata* es posible conocer la extensión de varios archivos alojados en la carpeta de trabajo. Para esto se emplea el comando dir que muestra los archivos en el directorio junto con su extensión.

3.2. Type

El comando type permite visualizar el contenido de un archivo de datos en la ventana de salida de *Stata*. En la sintaxis del comando es importante especificar la extensión del archivo⁸. Este comando es particularmente útil para diferenciar bases de datos en archivos de texto. La sintaxis del comando es:

Ej. 2 \rightarrow type NBI_2011.csv

4. Importar datos de otros formatos a *Stata*

Los datos originales pueden encontrarse en diferentes tipos de formato. De ello depende la forma que usamos para importarlos a *Stata*. A continuación, listamos los tipos de archivos y los describimos brevemente. Existen dos grandes grupos de tipos de archivo: ASCII (csv, tab, raw, txt) o binario (*Excel*, *XML*, *SAS*, *SPSS*, bdf, shp). Los diferentes tipos de archivos se pueden importar mediante la barra de herramientas, pero se recomienda hacer uso del *do-file*.

8

4.1. Archivo de Excel

Un archivo de Excel puede tener extensión .xls (1997-2003) y .xlsx (2007-2016).

- Usando la barra de herramientas

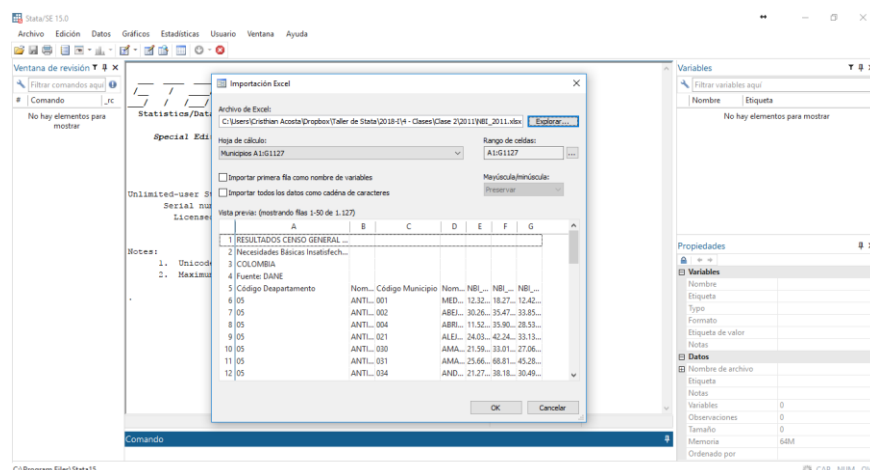
Puede importar el archivo de Excel “NBI_2011.xlsx” con la barra de herramientas, para esto:

1. Clic en *Archivo*.
2. Clic en *Importar*
3. Seleccione la opción *Hoja de Cálculo Excel (*.xls, *.xlsx)*

⁸ Debe conocer el formato del archivo para poder visualizar su contenido en *Stata*.

4. En la ventana abierta *Importación Excel* seleccione la opción de explorar. Debe buscar la carpeta donde está almacenado el archivo de *Excel*.
5. Seleccione el archivo Excel y, luego, clic en *Abrir*. Automáticamente aparecerá la ventana *Importación Excel* con la información del archivo, específicamente, *Hoja de cálculo* y el *Rango de celdas* (Figura 1).

Figura 5: Importación archivo Excel



6. Las primeras cuatro (4) filas de la vista previa corresponden al encabezado de información de los datos. Para no importar este texto se ajusta el rango de celdas a partir de la fila 5 en *Rango de celdas*.
 7. Seleccione la opción *Importar primera fila como nombre de variables*.
 8. Para finalizar la importación, clic en ok.
- Usando la ventana de comandos o el *do-file*⁹:

El comando `import excel` permite importar archivos *Excel* a *Stata*. Si desea conocer información del archivo sobre las hojas de cálculo y las dimensiones de las celdas, previo a la importación, la opción `describe` permite hacerlo:

Ej. 3 → `import excel "NBI_2011", describe`

⁹ Al realizar la importación desde la barra de herramientas, en la ventana de resultados aparece la programación de la importación. Esta programación en el *do-file* genera exactamente el mismo resultado. En este ejemplo, las opciones del comando de `import Excel` fueron *sheet*, *cellrange*, *firstrow* y *clear*.

Sino se especifica la extensión, por defecto, *Stata* intenta abrir el archivo .xls. En otro caso se procede a abrir el de extensión .xlsx. La opción *clear* cierra la base de datos previa y la reemplaza por la importada en la memoria de *Stata*.

Ej. 4 → import excel “NBI_2011”, clear

Las opciones más comunes del comando import excel son:

Ej. 5 → import excel “NBI_2011.xlsx”, sheet(“Municipios”) cellrange(A5:G1127) firstrow clear

Donde *sheet()* es el nombre de la hoja de cálculo, *cellrange()* es el rango de celdas de la hoja de cálculo, *firstrow* corresponde a la opción de importar la primera fila como nombre de variables.

Por último, si quiere que el nombre de las variables se importe en letra minúscula incluya, como opción, *case(lower)* y si quiere que esté en letra mayúscula añada *case(upper)*¹⁰.

4.2. Archivos de texto

Los archivos de texto pueden ser separado por comas (.csv), por espacios (.txt) y de formato fijo (.raw).

10

4.2.1. Separados por comas (.csv - comma-separated values)

Es un archivo de texto donde los valores de las variables están separados entre sí por comas. Los valores de una misma variable no necesariamente están uno sobre el otro entre filas.

Excel puede abrir este tipo de archivos. Para realizar la importación a *Stata* es necesario usar el comando import delimited. Por defecto, sin especificar la extensión se importa el archivo separado por comas (.csv).

Ej. 6 → import delimited “NBI_2011”, clear

Esto también es igual a:

Ej. 7 → import delimited “NBI_2011”, delimiter(comma) clear

¹⁰ Por defecto, *Stata* importa el nombre de las variables como se encuentran en el archivo Excel. Para usar esta opción es necesario añadir *firstrow*.

4.2.2. Separados por espacio (.txt)

Excel puede abrir este tipo de archivos. Para realizar la importación a *Stata* es necesario usar el comando import delimited.

Ej. 8 → `import delimited "NBI_2011.txt", delimiter(tab) clear`

4.3. Manualmente

El comando input permite introducir datos directamente a *Stata* desde la ventana de comandos. La sintaxis es:

1. `input var1 var2 ... varm`
2. `x11 x21 ... xm1`
3. ...
4. `x1n x2n ... xmn`
5. `end`

Donde m es el número de variables dentro de la base de datos y n el número de observaciones.

5. Compatibilidad entre versiones de *Stata*

11

Debido a los cambios en el software de *Stata*, las versiones anteriores a *Stata* 14 no pueden abrir las bases de datos creadas con las nuevas versiones. Para abrir la base de datos, un usuario que utiliza *Stata* 13 necesita guardar la base con el comando saveold.¹¹

Ej. 9 → `saveold "SABER11_20142_old"`

Si requiero la base en *Stata* 12 incluyo la opción *version*.

Ej. 10 → `saveold "SABER11_20142_old", version(12)`

Sin embargo, si olvido guardar la base (solo *Stata* 13) en un viejo formato podría abrirla en *Stata* 12 con el comando use13. Para esto, primero debo descargar el comando con `ssc install use13`.

Una vez descargado el paquete puede abrir la base con el comando use13.

Ej. 11 → `use13 "SABER11_20142_old", replace`

¹¹ Note que la versión por defecto es *Stata* 13.

6. Comandos para bases de datos

6.1. Comprimir bases de datos

Comprimir datos permite reducir la cantidad de memoria usada por la base de datos en la sesión de *Stata*. Para esto, es necesario usar el comando compress. En particular, el proceso de comprimir datos consiste en cambiar en un nuevo formato las variables que requieran menor memoria (bytes).

6.2. Ordenar bases de datos (order, sort)

Existen varios comandos que permiten ordenar la base de datos. Está el comando order, que se mencionó en las notas de la clase 2 y en el cual no se entrará en detalle acá dada la redundancia. Adicionalmente están los comandos sort y gsort. Ambos comandos permiten ordenar las observaciones de la base de datos de acuerdo con un conjunto de variables que el investigador le señale a *Stata*. Por ejemplo, si se quiere organizar las observaciones de la base “Ingreso_u.dta”, que contiene las notas de un curso particular y otros datos de los individuos como si estudiaron en colegio mixto o su género, el comando que se debe ejecutar es el siguiente:

Ej. 12 \rightarrow sort *n_def*

12

Ya con esto las observaciones de la base están organizadas por nota definitiva del curso en orden ascendente. Supongamos ahora que se quiere organizar las observaciones de la base de manera descendente, de manera que los primeros individuos que aparezcan sean quienes tuvieron mejores notas en el curso. En este caso, no sería de mucha ayuda el comando sort. Para poder ordenar la base de manera tanto ascendente como descendente se requiere el comando gsort.

Ej. 13 \rightarrow gsort - *n_def*

La sintaxis general del comando gsort es con un signo negativo (-) anterior a la variable que se desee ordenar de forma descendente y con un signo positivo (+) anterior a la variable que se quiera esté ordenada de forma ascendente.

Aunque parezca trivial, es importante hacer énfasis en que los comandos sort y gsort cambian el orden de las observaciones en la base de datos. La que antes era la observación 1 o la que era la observación 7 dejará de serlo una vez ejecutado alguno de estos comandos. Dado esto, es importante ser cuidadoso cuando se empleen los condicionales en observaciones particulares.