



Universidad de la República
Facultad de Ciencias Sociales
DEPARTAMENTO DE ECONOMIA

Notas Docentes

Introducción a STATA

**Juan Pablo Pagano, Ianina Rossi,
Máximo Rossi y Patricia Triunfo**

Nota Docente No. 17



INTRODUCCIÓN A STATA

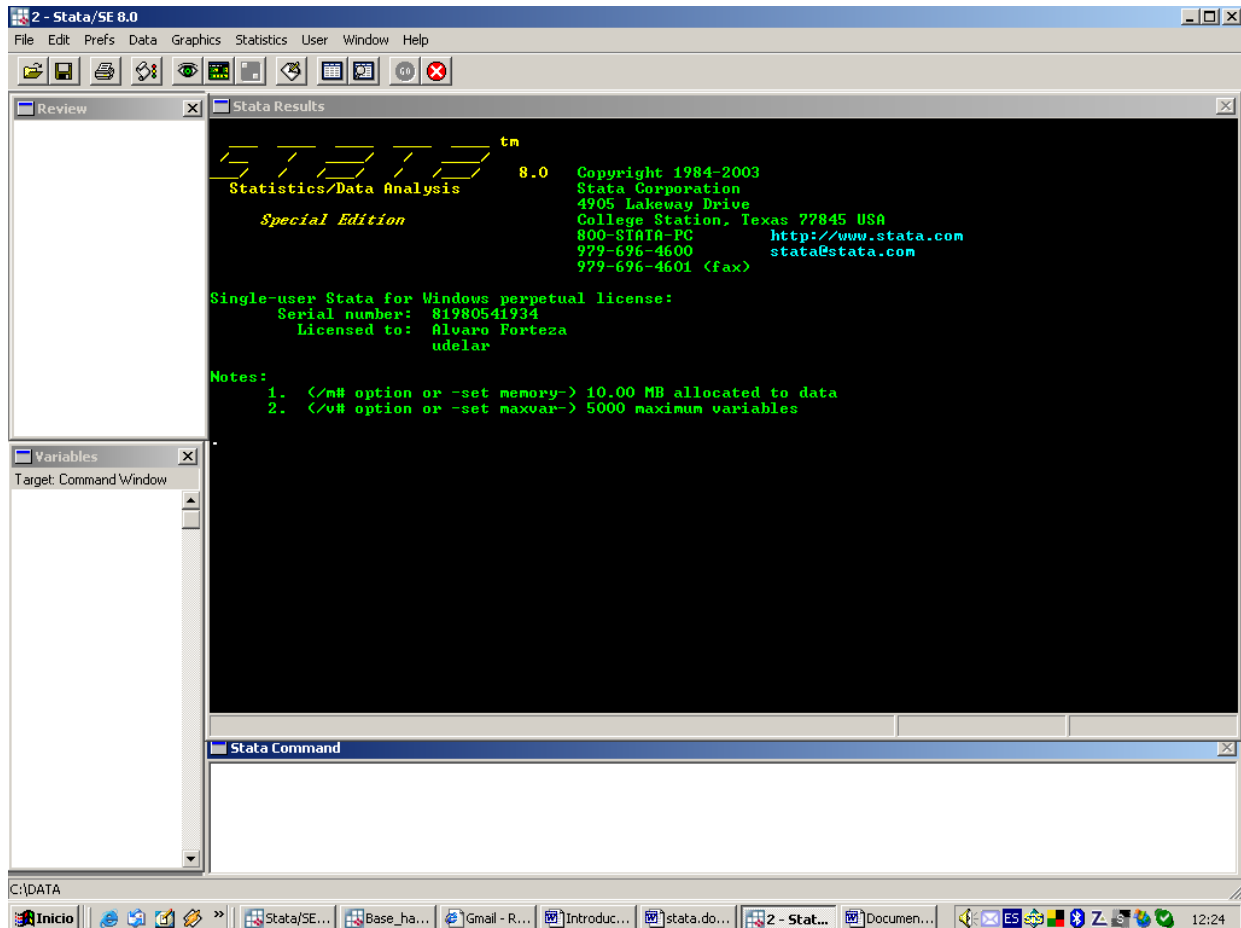
Juan Pablo Pagano*
Ianina Rossi*
Máximo Rossi*
Patricia Triunfo*

* Departamento de Economía, Facultad de Ciencias Sociales, Universidad de la República, Uruguay.

1. Introducción General¹

En Stata se pueden ejecutar los comandos interactivamente o ejecutar un conjunto de comandos a partir de un archivo **do**.

Un panorama general: pantallas



En la pantalla *Stata Results* se despliegan los resultados a medida que se van ejecutando los distintos comandos.²

La ventana *Stata Command* permite escribir comandos que se ejecutarán y se desplegarán en *Stata Results* al introducir la tecla *enter*.

¹ Todos los ejemplos de estas notas se basan en la Encuesta Continua de Hogares del INE del año 2005, la cual se encuentra disponible en red (<http://www.ine.gub.uy>).

² Para quienes utilizaban otros paquetes estadísticos como el SPSS, la ventana *Stata Results* equivale al *Output*.

La pantalla *Review* guarda el historial de los comandos ejecutados desde el inicio de la sesión en Stata. Si se quiere repetir algún comando, puede seleccionarse de allí con el *mousse* en vez de escribirlo de nuevo en el *Stata Command*.

La ventana *Variables* muestra las variables de la base de datos en la cual se esté trabajando. Las variables pueden seleccionarse de allí con el *mousse* en lugar para no tener que escribirlas en el *Stata Command*.

Tipos de archivos:

Clases de archivos Stata con sus extensiones

.dta	archivos de datos
.do	archivos de comandos
.ado	programas
.hlp	archivos de ayuda
.gph	gráficos
.dct	archivos diccionarios
.smcl	archivos log

Cargando archivos de datos:

Stata mantiene los datos en memoria. Si se ha estado trabajando en una base y se quiere trabajar sobre otra, es necesario removerla de la memoria.

clear

El comando *clear* borra todos los datos que se encontraban en memoria.

Si la base es grande, es necesario incrementar la cantidad de memoria disponible para una sesión de Stata. Esto es necesario si se sabe de antemano que la base es grande o si cuando se intenta leer la base se obtiene el siguiente mensaje:

no room to add more observations

El comando para ampliar la memoria es:

*set memory 64m*³

³ La parte subrayada del comando indica la posibilidad de ejecutar el mismo abreviando; es decir, que es lo mismo escribir *set memory* que *set mem*. Stata reconoce el comando con las primeras letras, siempre que sean únicas, es decir, que no haya dos comandos que empiecen con las mismas letras. Por eso en el *help* de cada comando aparece una parte del comando subrayada, indicando las posibilidades de abreviación. Ver numeral 2.

64m o lo que sea necesario, el límite lo da el computador que se este usando. Luego se lee la base de datos.

Con el comando *memory* se puede conocer el monto de memoria que se esta usando.

Lectura de datos:

Existen varios caminos para leer los datos en Stata

use es para leer datos que han sido guardados en formato Stata:

Distintas posibilidades:

a) si los datos están en C:\ECH\2005\

```
cd "C:\ECH\2005\"  
use p2005
```

o

```
use C:\ECH\2005\ p2005.dta
```

b) Si los datos están en un directorio por defecto

```
use p2005
```

c) Subir datos parciales de un archivo

```
use e1 e2 using p2005 // solamente selecciona las variables sexo y edad
```

```
use p2005 in 1/5 // selecciona sólo las cinco primeras observaciones, todas las variables
```

```
use e1 e2 using p2005 in 1/5 // selecciona dos variables y cinco primeras observaciones
```

Hay una serie de comandos para importar datos en distintos formatos, pero lo más útil es tener el programa *Stat-transfer* que permite convertir archivos de datos en distintos formatos.

Descripción de datos:

Una vez que se leen los datos, pueden ver las características de la base de datos y las variables que se seleccionen (si no se seleccionan variables, se muestran todas las variables de la base de datos que se encuentre actualmente en uso) con el comando *describe*.

```
describe e1 e2 // descripción general de la base de datos y de las variables sexo y edad
```

```
Contains data from C:\ECH\2005\p2005.dta
obs:      54,330
vars:      159
size:     20,591,070 (80.4% of memory free)
```

```
-----
```

variable name	storage type	display format	value label	variable label
e1	byte	%8.0g		
e2	byte	%8.0g		

```
-----
```

Sorted by:

También se pueden conocer las características de la base sin la lectura de los datos (se realiza la descripción de la base de datos y variables aun cuando me encuentre trabajando sobre otra base de datos):

describe using C:\ECH\2005\p2005.dta

El comando *list* muestra los valores de las variables seleccionadas en todas las observaciones o en un rango de observaciones:

list // muestra los valores de todas las variables en todas las observaciones

list in 1/5 // muestra los valores de todas las variables en las primeras cinco observaciones

list e1 // muestra los valores de la variable sexo en todas las observaciones

list e1 e2 in 1/5 // muestra los valores de las variables sexo y edad en las primeras cinco observaciones

```
+-----+
| e1  e2 |
+-----+
1. |  2  58 |
2. |  1  20 |
3. |  2  32 |
4. |  2  79 |
5. |  1  39 |
+-----+
```

El comando *codebook* informa sobre las variables: estadísticas descriptivas, etiquetas, detalle de los valores missing, etc. Si no se especifican variables, *codebook* despliega estas estadísticas para todas las variables de la base de datos.

codebook // despliega estadísticas para todas las variables de la base de datos

codebook e1 e2 // despliega estadísticas sólo para las variables sexo y edad

```
-----
e1
(unlabeled)
-----
```

```

type:  numeric (byte)

range:  [1,2]          units:  1
unique values:  2      missing .:  0/54330

tabulation:  Freq.  Value
              25290  1
              29040  2
```

```
-----
e2
(unlabeled)
-----
```

```

              type:  numeric (byte)

              range:  [0,99]                units:  1
unique values:  100                          missing .:  0/54330

              mean:    37.328
              std. dev: 23.6804

percentiles:      10%      25%      50%      75%      90%
                  7        17       35       56       72

```

El comando *summarize* informa las estadísticas descriptivas básicas de las variables especificadas (si no se especifican variables *summarize* reporta las estadísticas descriptivas de todas las variables de la base de datos): número de observaciones, media, desvío estándar, mínimo y máximo.

summarize // despliega estadísticas descriptivas básicas para todas las variables de la base de datos

summarize e1 e2 // despliega estadísticas descriptivas básicas para el sexo y la edad

Variable	Obs	Mean	Std. Dev.	Min	Max
e1	54330	1.534511	.4988121	1	2
e2	54330	37.32803	23.68041	0	99

La opción *detail* da información adicional:⁴

summarize e1, detail // despliega estadísticas descriptivas detalladas para la edad

```

              e1
-----
Percentiles  Smallest
 1%          1          1
 5%          1          1
10%          1          1   Obs          54330
25%          1          1   Sum of Wgt.  54330

50%          2
              Largest   Mean          1.534511
75%          2          2   Std. Dev.    .4988121
90%          2          2   Variance     .2488135
95%          2          2   Skewness    -.1383753
99%          2          2   Kurtosis    1.019148

```

El comando *tabulate* despliega la tabla de frecuencia de una o dos variables. Tiene distintas opciones tales como porcentajes por fila, columna y celda, además de Chi-cuadrado y otros estadísticos.

tabulate e1 // despliega tabla de frecuencias para la variable sexo

⁴ Las opciones de los diferentes comandos siempre van separados del propio comando por una coma (","), Las opciones también admiten abreviaciones. Para ver las opciones disponible para un comando escribir *help summarize* (o el comando en cuestión).

e1	Freq.	Percent	Cum.
1	25,290	46.55	46.55
2	29,040	53.45	100.00
Total	54,330	100.00	

tabulate e1 e4 // despliega tabla de frecuencias de doble entrada para sexo y estado conyugal

e1	e4						Total
	1	2	3	4	5	6	
1	8,850	2,497	586	480	679	12,198	25,290
2	8,887	2,507	1,525	1,007	3,437	11,677	29,040
Total	17,737	5,004	2,111	1,487	4,116	23,875	54,330

tabulate e1 e4, missing // despliega tabla de frecuencias de doble entrada para sexo y estado conyugal especificando los porcentajes de valores perdidos (*missing values*).

Sintaxis básica de los comandos

El programa es sensible a mayúsculas y minúsculas para las variables.

Los comandos deben ser escritos en minúsculas.

La estructura general de una instrucción en Stata es:

[by varlist]: comando [varlist] [=exp] [if exp] [in range] [ponderador], [opciones]

varlist:

Lista de variables a las cuales se les quiere aplicar el comando específico.

Ejemplo:

summarize e1 e2 // aquí varlist = e1 e2

Variable	Obs	Mean	Std. Dev.	Min	Max
e1	54330	1.534511	.4988121	1	2
e2	54330	37.32803	23.68041	0	99

by varlist:

Se aplica el comando específico por valores de determinada variable. Para poder aplicar la opción *by* los datos siempre deben estar ordenados por la variable de corte seleccionada. El comando utilizado para ordenar datos es *sort*.

Ejemplo:

sort e1 // ordena los datos por los valores de la variable sexo

by e1: summarize e2 // despliega las estadísticas descriptivas de la variable edad para hombres y mujeres por separado.

-> e1 = 1

Variable	Obs	Mean	Std. Dev.	Min	Max
e2	25290	35.2295	23.03871	0	98

-> e1 = 2

Variable	Obs	Mean	Std. Dev.	Min	Max
e2	29040	39.15558	24.07724	0	99

Alternativamente también podemos hacer:

by e1, sort: summarize e2

if exp:

Se aplica el comando específico si se cumple la condición especificada. Para poder definir una expresión que involucre valores de variables, se utilizan los siguientes símbolos: < (menor), <= (menor o igual), > (mayor), >= (mayor o igual), == (igual), & (y), | (o), ~= o != (diferente de).

Ejemplos:

tabulate e1 if pobpcoac==2 // estructura por sexo de los ocupados

e1	Freq.	Percent	Cum.
1	12,395	55.19	55.19
2	10,062	44.81	100.00
Total	22,457	100.00	

tabulate e4 if pobpcoac>=6 & pobpcoac<=11 // estado conyugal de los inactivos

e4	Freq.	Percent	Cum.
1	6,831	37.99	37.99
2	1,088	6.05	44.04
3	620	3.45	47.49
4	363	2.02	49.51
5	3,426	19.05	68.56
6	5,654	31.44	100.00
Total	17,982	100.00	

tabulate e1 if pobpcoac==9 / pobpcoac==10 // estructura por sexo de jubilados y pensionistas

e1	Freq.	Percent	Cum.
1	3,673	38.89	38.89
2	5,772	61.11	100.00
Total	9,445	100.00	

in range:

Se aplica el comando a un rango de las observaciones.

Ejemplo:

tabulate pobpcoac in 5/10 // tabla de frecuencias de la variable condición de actividad en las segundas cinco observaciones

pobpcoac	Freq.	Percent	Cum.
1	1	16.67	16.67
2	1	16.67	33.33
4	1	16.67	50.00
6	1	16.67	66.67
7	1	16.67	83.33
9	1	16.67	100.00
Total	6	100.00	

=exp:

Especifica el valor asignado a una variable. Generalmente se usa con los comandos *generate* y *replace*.

Ejemplo:

generate edad2=e2^2

weight:

Indica el ponderador asignado a cada observación, hay distintos ponderadores (ver *help weights*)

Ejemplo:

summarize e2 [fweight=pesoan]

Variable	Obs	Mean	Std. Dev.	Min	Max
e2	2309272	37.08543	23.67529	0	99

options:

Las opciones se indican en la sintaxis de un comando después de una coma. Las opciones también admiten abreviaciones y dependiendo del comando se tendrá un set de opciones diferentes. Para ver la sintaxis y las opciones admitidas por un comando (así como para tener información acerca de qué es lo que hace exactamente el comando) se puede utilizar la ayuda de Stata.

***help* [comando]**

Ejemplo:

help summarize

help for summarize
manual: [R] summarize

summarize

dialog:

Summary statistics

```
summarize [varlist] [weight] [if exp] [in range] [, [detail|meanonly] format separator(#) ]
```

by ... : may be used with summarize; see help by.

aweight, fweight, and iweight are allowed; see help weights. However, iweight may not be used with the detail option.

The varlist following summarize may contain time-series operators; see help varlist.

Description

summarize calculates and displays a variety of univariate summary statistics. If no varlist is specified, summary statistics are calculated for all the variables in the data.

See help tabsum for making tables of means.

See help table and tabstat for making tables of means and other statistics.

See help ci for calculating the standard error and confidence intervals of the mean.

See help stsum for calculating summary statistics with survival-time data.

See help xtsum for calculating summary statistics with cross-sectional time-series data.

See help svymean and svytab for calculating summary statistics with survey data.

Options

detail produces additional statistics including skewness, kurtosis, and the four smallest and four largest values, along with various percentiles.

meanonly is a programmer's option. meanonly is allowed only when detail is not specified. It suppresses display of the results and calculation of the variance.

format requests that the summary statistics be displayed using the display format associated with the variables rather than the default g display format; see help format.

separator(#) specifies how often separation lines should be inserted into the output; the default is separator(5), meaning that a line is drawn every 5 variables. separator(10) would draw the line every 10 variables. separator(0) suppresses the separation line altogether.

Examples

```
. summarize
. summarize mpg weight
. summarize mpg weight if foreign
. summarize mpg weight if foreign, detail
```

Also see

Manual: [R] summarize

Online: help for centile, cf, ci, codebook, compare, describe, egen, inspect, lv, means, pctlile, stsum, svymean, svytab, table, tabstat, tabsum, xtsum

Tipos de datos

Stata almacena o formatea los datos con dos posibilidades: numeric o string. Numeric almacena números (por ejemplo: precios, empleo, etc) mientras que string almacena texto (Uruguay, Montevideo, mujer, etc). String puede también almacenar números pero no se podrá realizar ninguna operación con ellos.

Los números missing se verán como (.) y missing string por dobles comillas en blanco (“”).

Creando un archivo *log* log file

Es conveniente mantener en un archivo toda la sesión de trabajo. Para crear un log file se usa las instrucciones:

capture log close // cierra el log justo al comienzo en caso de que este abierto⁵
log using ech

Stata por defecto asignará la extensión *.smcl* (Stata markup control language).

log using ech.log

Para cerrar el log file:

log close

log using ech.log, append
continúa grabando un archivo log

log using ech.log, replace
reemplaza un archivo previamente grabado

Para ver el log file:

view ech.smcl

Para convertir el log file a formato texto y poder levantarlo con cualquier procesador de palabras

translate ech.smcl to ech.txt

Para correr Stata: llame a Stata; abra un log file; acceda a los datos; corra los procedimientos; cierre el log file; imprima los resultados.

Grabando archivos;

Para grabar un archivo de datos se utiliza el comando *save*:

save ech

Si ya se dispone de ese archivo y se desea rescribirlo:

save ech, replace

⁵ *capture* es una opción antes de un comando que impide que Stata “corte” la sesión en caso de que el comando no sea válido. Por ejemplo, si uno al principio de un archivo *.do* le pide a Stata que cierre un log (*log close*) y no hay un log abierto, se desplegará un mensaje de error y el *.do* deja de correr en ese momento. Con *capture* no se despliegan mensajes de error, y por lo tanto el programa sigue corriendo.

Con el comando *replace* se pierden los datos originales.
Para hacerlo en un directorio determinado:

save a:/ech, replace

saveold graba en formato para ser leído por Stata7

Creando archivos *do*;

En vez de escribir los comandos interactivamente Stata permite coleccionar los comandos en un archivo do-file y ejecutar el do-file completo o seleccionar partes.⁶

Todos los resultados aparecerán grabados en un archivo log-file para revisar cuando se ejecute toda la sesión.

Do-files se pueden escribir en cualquier editor de texto como notepad. Stata tiene su propio editor de texto.

Los comandos se separan por *enter* o *return*.

También se pueden insertar comentarios dentro de los do-files, que no serán leídos por Stata. En este sentido, podemos insertar comentarios de las siguientes formas:
una única línea que comienza por *
texto de una o varias líneas entre /* y */
texto en una línea precedido por ///

En general tiene el siguiente formato:

```
clear
set memory 64m
versión 8.0
cd "C:|ECH|2005|"
capture log close
log using ech.log, replace
set more off
listado de diferentes comandos
log close
```

version 8.0 – Permite ejecutar los comandos en la versión correspondiente. Stata es compatible siempre con las versiones anteriores.

clear // limpia cualquier dato que se encuentre en memoria.

⁶ Para quienes utilizaban paquetes estadísticos como el SPSS, los archivos do equivalen a los archivos *Syntax*

`cd "C:\ECH\2005\"` // establece el directorio por defecto del proyecto que se esté trabajando.

`capture log close` // cierra todos los archivos log que estén abiertos si es que los hay.

`log using ech.log, replace` // señala guardar todos los resultados en un archivo log ya existente y que lo sobrescriba.

`set more off` // cuando los resultados de la corrida de un do-file superan en extensión a una página del *Stata Results*, Stata pausa el do-file para dar tiempo a que el usuario pueda revisar los resultados que va obteniendo en cada página, para continuar observando la ejecución de los comandos hay que presionar una tecla cualquiera. El comando *set more off* ordena a Stata a correr sin pausas el do-file entero, pudiendo ver el usuario el proceso paso a paso en el archivo log correspondiente. La decisión puede cambiarse en cualquier momento con el comando *set more on*.

`set memory 64m` // incrementa el uso de memoria hasta los 64 mega por parte de Stata.

`log close` // cierra el archivo log.

Es bueno poner comentarios a los comandos que se van ejecutando o a la creación de variables o cualquier otro criterio que sea importante recordar.

Stata ignora una línea que empieza con *. Esto es ideal para poner comentarios del estilo de títulos de lo que se está haciendo.

Stata ignora todo lo que esté entre /* y */. Esto es ideal para realizar comentarios de resultados que se van obteniendo o acerca de temas pendientes, etc. que habitualmente tienen una extensión mayor a una línea. También es muy útil para partir comandos largos. Si en un archivo do-file el comando es más largo que el ancho de la pantalla, el comando no se corta sino que sigue hacia la derecha y hay que moverse con el cursor para poder ver el comando completo. Se puede utilizar /* y */ para que Stata no lea los espacios en el medio y tome a dos o más líneas como si fueran una sola.

Ejemplo:

```
generate yotros_1 = ( g2_3 + g4_1_1 + g4_1_2 + g4_2_1 + g4_2_2 + g4_2_3 + g4_2_4 /*  
*/ + g4_2_5 + g4_2_6 + g5_2 + g5_3 + g5_4 + g5_5 + g5_6 + g5_7 + g5_8 + g5_9 + g5_10 /*  
*/ + g5_11)* infla05
```

Stata también ignora todo lo que escrito posteriormente de //, incluso dentro de una misma línea.

Ejemplo:

tabulate e1 if pobpcoac==9 / pobpcoac==10 // estructura por sexo de jubilados y pensionistas

e1	Freq.	Percent	Cum.
1	3,673	38.89	38.89
2	5,772	61.11	100.00
Total	9,445	100.00	

* Vemos que el 61% de los jubilados y pensionistas son mujeres y el 39% son hombres.

Preserve y restore

Preserve y restore comandos: preserve toma una foto de la base de datos y se pueden realizar modificaciones en la base. Si se quiere revertir se hace restore.

Ejecutando un archivo *do*

Ejecutando un archivo do y mostrar los resultados

do myfile.do

Ejecutando un archivo do y no mostrar los resultados

run myfile.do

Seleccionando ejecución en el menú (se selecciona la sección que se quiere ejecutar se va al menú del do-file y se selecciona tools / do selection).

Tips 1

El archivo de datos por defecto de Stata es .dta, por consiguiente no es necesario tipear la extensión.

Se pueden rescribir comandos y variables de Stata haciendo *click* en los comandos almacenados en la ventanas *Review* y *Variable Windows*.

Se pueden repetir comandos usando las teclas PageUp y PageDown.

Los comandos más usados se pueden guardar y copiar y pegar cada vez que sea necesario.

Organización de los datos

Creación o modificación de variables:

Creación de variables a partir de las que ya se disponen. Los comandos más usados son `generate` y `egen`.

Por ejemplo crear una variable que es simplemente la suma de las existentes:

`generate yotran_1 = g3_4 + g3_5 + g3_6 + g3_7 + g3_8`

`label variable yotran_1 "Ingresos por otras transferencias"` // Se etiqueta la variable

Para generar la suma acumulada del total (va sumando renglón a renglón):

`generate totyotran=sum(yotran_1)`

Veamos lo que queda:

`list yotran_1 totyotran in 1/20`

	yotran_1	totyot~n
1.	0	0
2.	0	0
3.	0	0
4.	0	0
5.	725	725
6.	0	725
7.	0	725
8.	0	725
9.	0	725
10.	0	725
11.	0	725
12.	0	725
13.	1500	2225
14.	0	2225
15.	8217	10442
16.	2435	12877
17.	0	12877
18.	0	12877
19.	0	12877
20.	0	12877

Para generar el número de una observación en la base de datos (tener en cuenta que es a partir del momento que se ejecuta el comando y no en el orden original):

`generate id=_n`

Otras expresiones:

`generate` admite todas las expresiones de (if exp.) más todos los operadores matemáticos sencillos (suma, resta, multiplicación, división, elevado a, etc.)


```

generate cidiff=abs(cc-ci) // diferencia absoluta
generate lnedad=ln(e2) // natural log
generate logedad=log10(e2) // log base 10
generate edadsq=e2^2 // edad al cuadrado
generate ten=10 // valor constante 10

```

replace total=total1 /* el comando *replace* permite cambiar una variable existente */

Rename

Cambia el nombre de las variables:

rename e1 sexo // primero se introduce el nombre viejo y luego el nuevo

Se puede cambiar una variable por instrucción.

Recode

Cambia el valor de determinada variable:

recode sexo 2=0

Importante cuando se tiene los missing como un número determinado

recode var1 -99=.

Con variables string es necesario utilizar el comando *replace*;

replace sexo="mujer" if sexo=="female"

Variables binarias o dummy:

Hay distintas opciones para crearlas:

tabulate sexo, generate(dummy)

Se crea una variable binaria para cada valor que se encuentra en sexo, poniéndoles el nombre dummy1 y dummy2 (o el nombre que se elija, pero siempre asignará valores correlativos a las variables binarias generadas).

generate dummymed=sexo==2

crea una variable binaria que es igual a 1 si sexo=2 y cero en otro caso

También se puede crear por la vía:

generate dummymed=0

replace dummymed=1 if sexo==2

En cualquiera de estos dos casos se debe tener cuidado si hay *missing values*, puesto que se estaría asignando un cero a algo que en realidad es un *missing*.

Para arreglar el problema se puede hacer

replace *dummymed*=. if *sexo*==.

Keep and drop

Permite mantener o eliminar variables u observaciones de la base de datos.

keep e1 e2 // sólo utilizará las variables sexo y edad
***keep if mes*>=3** // utilizará sólo las encuestas realizadas en verano
***drop if mes*<4**
***keep if mes*>=4 & *mes*<=9**
***drop if mes*<4 | *mes*>8**
***keep if (mes*>=1 & *mes*<=3) | (*mes*>=9 & *mes*<=12)**
***drop if e1*==.**

Eliminar las primeras 10 observaciones:

***drop if _n*<=10**

Eliminar la última observación en la base:

***drop if _n*==_N**

sort

sort ordena la base de datos por una o varias variables, dando prioridad al orden de la variable que se pone en primer lugar, segundo, etc.. Por defecto, *sort* ordena las observaciones de menor a mayor.

sort *e1*
sort *e1 e2*

gsort

gsort permite elegir cómo se debe ordenar cada variable, pudiendo cambiar entre descendente o ascendente para cada variable.

gsort +*e1* -*e2* / ordena al sexo de menor a mayor y la edad de mayor a menor

Egen un comando muy útil para crear variables:

Egen o *extended generate* se utiliza cuando se quieren generar variables que requieren operadores matemáticos complejos: suma de una variable (no renglón a renglón sino el total), máximo, mínimo, número de casos, media, mediana, etc..

Ejemplos:

egen sumvar1=sum(var1) // suma de la var1

egen meanvar1=mean (var1), by (var3) // media de la variable var1 por cada var3

egen count=count(id), by(dpto) // número de casos por variable departamento

Lista de posibles opciones para egen:

count number of non missing values

diff compares variables, 1 if different, 0 otherwise

fill fill with a pattern

group creates a group id from a list of variables

iqr interquartile range

ma moving average

max Maximum value

mean mean

median median

min minimum value

pctile percentile

rank rank

rmean

mean accross variables

sd standard deviation

std standard scores

sum sums

Tips 2

Búsqueda de Ayuda en Stata:

Comandos usados para este objetivo:

help

search

tutorial

Help: puede usarse desde la línea de comando o desde la ventana de ayuda. Al usar help el comando se debe escribir sin abreviación.

Ejemplos:

help if

help summarize

help regression

help tabulate

Search: este comando busca información en los manuales de Stata, en FAQs y en los Stata Technical Bulletins. El comando findit combina esta búsqueda y la búsqueda en la red.

Tutorial: Escribiendo tutorial se da información sobre el tutorial. Escribiendo tutorial regress (por ejemplo) da información del tutorial sobre regresión.

tutorial

tutorial regress