

Clase Auxiliar N°2

Economía de la Educación

Lunes, 25 de Agosto, 2008.

En esta clase se hará una breve introducción al uso del SW Estadístico STATA, el cual posee la ventaja de tener incorporados varios métodos econométricos de análisis de datos (Corrección de Heckman, Matching, etc.).

Se usará la base de datos MROZ.DTA para ejemplificar el uso de STATA

Fuente: T.A. Mroz (1987), “The Sensitivity of an Empirical Model of Married Women’s Hours of Work to Economic and Statistical Assumptions,” *Econometrica* 55, 765-799.

# Observaciones: 753	
VARIABLES	
1. inlf	=1 if in labor force, 1975
2. hours	hours worked, 1975
3. kidslt6	# kids < 6 years
4. kidsge6	# kids 6-18
5. age	woman's age in yrs
6. educ	years of schooling
7. wage	estimated wage from earns., hours
8. repwage	reported wage at interview in 1976
9. hushrs	hours worked by husband, 1975
10. husage	husband's age
11. huseduc	husband's years of schooling
12. huswage	husband's hourly wage, 1975
13. faminc	family income, 1975
14. mtr	fed. marginal tax rate facing woman
15. motheduc	mother's years of schooling
16. fatheduc	father's years of schooling
17. unem	unem. rate in county of resid.
18. city	=1 if live in SMSA
19. exper	actual labor mkt exper
20. nwifeinc	(faminc - wage*hours)/1000
21. lwage	log(wage)
22. expersq	exper^2

DESCRIPCIÓN DEL AMBIENTE DE TRABAJO EN STATA

El entorno de trabajo, o ventana principal, de STATA se subdivide en cuatro ventanas básicas, además de la barra de menús y la barra de herramientas.

Las **ventanas básicas** reciben los siguientes nombres:

Review	Listado de los comandos ejecutados desde que se inició la aplicación
Variables	Listado de las variables contenidas en la base de datos cargada.
Command	En esta ventana se escriben las instrucciones que se le dan a la aplicación
Result	Aparecen los resultados de las órdenes ejecutadas.

GESTION DE BASES DE DATOS

Las bases de datos en STATA se denominan **dataset**.

Para cargar un dataset en formato STATA (extensión .dta) se tienen 2 opciones:

- desde el menú FILE → OPEN
- mediante una instrucción desde la ventana **Command**:

use “nombre de fichero”

Nota: Si ya tenemos un dataset en memoria, la aplicación no permitirá cargar uno nuevo, ya que borraría el anterior. Si este es el caso, se debe usar la opción **clear** para desalojar de la memoria el anterior dataset:

use “nombre de fichero”, clear

El comando use es bastante flexible, permitiendo controlar qué parte de la base de datos se desea cargar.

use [varlist] [if] [in] using filename [, clear nolabel]

Para conocer el contenido del dataset se utiliza el comando (instrucción) **describe**

Bases de datos en otros formatos

▪ ASCII (.txt):

El comando para realizar la lectura de este tipo de archivos es **infile**:

infile varlist using filename [if] [in] [, options]
--

```
infile  inlf hours kidslt6 kidsge6 age educ wage repwage  hushrs husage huseduc  huswage faminc  
mtr motheduc fatheduc unem city exper nwifeinc lwage expersq  using "D:\.....\mroz.txt"
```

Nota: El archivo de datos debe contener sólo las realizaciones de las variables (sin el nombre); si la primera fila corresponde al nombre de las variables, debe borrarse.

▪ Otros:

Está disponible el programa STAT/TRANSFER para realizar la traducción de datos entre distintos formatos, como por ejemplo SPSS, SAS, STATA, MS-Excel, etc.

SINTAXIS DE COMANDOS STATA

[prefix :] command [varlist] [=exp] [if] [in] [weight] [using filename] [, options]
--

command [varlist] [if] [in] [weight] [, options]

[...]	Todo lo que aparece en corchetes es opcional
if	Seguida de una expresión lógica indica que sólo los datos que verifiquen dicha condición serán incluidos en el análisis
in	Sirve para indicar un rango de observaciones a analizar (p.e.: in 1/100)
weight	Sirve para indicar una variable de ponderación
options	Son las opciones específicas del comando utilizado

Los comandos se escriben en la ventana **Command**. Para ejecutar, presionar **ENTER**.

Ejemplos de Comandos

- Para solicitar estadísticos de resumen (media, desviación estándar, máximos y mínimos) se usa el comando **summarize**:

```
summarize wage educ age exper  
help summarize
```

- Para construir tablas de estadísticos también se puede usar **tabstat**

```
tabstat wage educ age exper, stat (min, max, mean, median)  
tabstat wage educ age exper, stat (min, max, mean, median) by(inlf)
```

- Para tablas de frecuencias se usa el comando **tabulate**

```
tabulate educ
```

- Con el comando **tabulate** también se pueden crear tablas de contingencia

```
tabulate city inlf, chi
```

- Comando **table**, para controlar el contenido de cada casilla.

```
table educ, cont (mean wage) row format (%9.1f)
```

```
help table
```

- Gráficos

```
hist wage  
hist lwage, freq normal
```

```
graph bar wage, over(educ)
```

```
graph box age, over(inlf)
```

- Diferencia de medias

```
ttest age, by(inlf)
```

- Generación/reemplazo de variables

```
gen exper_cuad=exper^2  
summarize exper_cuad expersq
```

```
gen wage_log=log(wage)  
summarize lwage wage_log
```

```
gen educ_sup=0  
replace educ_sup=1 if educ>12  
tabulate educ educ_sup  
ttest wage, by(educ_sup)  
graph box wage, over(educ_sup)  
tabulate educ_sup inlf, chi
```

- Etiquetas (labels) para valores de categorías

```
label define labeds 0 NO 1 SI  
label values educ_sup labeds  
tabulate educ_sup  
codebook educ_sup
```

- Regresión lineal

```
reg lwage educ exper expersq
```

- Modelo Probit

```
probit inlf nwifeinc educ exper expersq age kidslt6 kidsge6
```

- Corrección de Heckman

```
heckman lwage educ exper expersq, sel(inlf = nwifeinc educ exper expersq  
age kidslt6 kidsge6) twostep
```

```
heckman lwage educ exper expersq, sel(inlf = nwifeinc educ exper expersq  
age kidslt6 kidsge6) robust
```

Para guardar los resultados obtenidos

El comando **log** crea un archivo permanente en disco con los contenidos de la ventana Results de los comandos ejecutados a través de la ventana Commands.

```
log using mylog.log
lista de comandos
log close
```

El commando crea un archive llamado mylog.log con los resultados de la lista de comandos que se ejecuten entre las instrucciones **log using mylog.log** y **log close**.

DO-FILES

Qué es un do-file:

Un do-file es un archivo de texto que contiene comandos de STATA.

Por qué usar do-files:

- Da la opción de corregir y re-ejecutar comandos.
- Provee la documentación sobre cómo se llegaron a los resultados.

Para crear un do-file:

Abrir el editor “Do file editor” (WINDOW → DO FILE EDITOR)

Escribir los comandos de STATA que se desean ejecutar.

Insertar comentarios en los do-files:

Se pueden agregar comentarios comenzando una línea con un asterisco *, STATA ignora todo lo que aparece en esa línea.

Ejecución del do-file creado:

Press the Do button (number two top right).

Ejemplo de do-file:

```
log using filename, replace
* Este do-file es un ejemplo
use data, clear
describe
... otros comandos....
log close
```