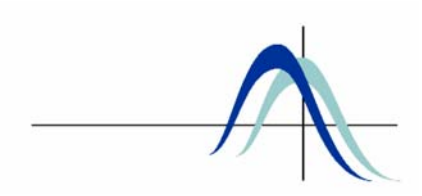
CENTRO DE
CIENCIAS HUMANAS
Y SOCIALES



Curso de introducción al paquete Stata

(versión 9.1 SE)

José Manuel Rojo Abuín
Unidad de Análisis Estadístico
Centro de Ciencias Humanas y Sociales
Consejo Superior de Investigaciones Científicas (CCHS, CSIC)
Madrid

Madrid, 14 y 15 de Febrero de 2008

Contenido

I. INTRODUCCIÓN

El objetivo de este manual es familiarizar al lector con el paquete Stata. La versión utilizada en la elaboración de este manual es Stata 9.1 SE para la familia Windows NT, es decir Windows professional y Windows XP, en sus múltiples versiones.

Stata es una aplicación desarrollada para realizar análisis estadísticos sobre muestras aleatorias de poblaciones. Hay quien señala una fuerte especialización en problemas asociados con la econometría.

Si bien tiene una interface gráfica de usuario (GUI), en la práctica es mucho más cómodo utilizar el potente a la vez que sencillo lenguaje de programación que incluye. Este lenguaje de programación requiere un cierto esfuerzo inicial hasta que el usuario empiece a acostumbrarse con los comandos; también suministra un sistema de ayuda realmente detallado y completo.

Nota:

En general, bajo el programa Stata trabajaremos mediante lenguaje de programación, y conviene recordar que es *case-sensitive*, es decir, **diferencia entre letras mayúsculas y minúsculas**

Genero ¡= genero

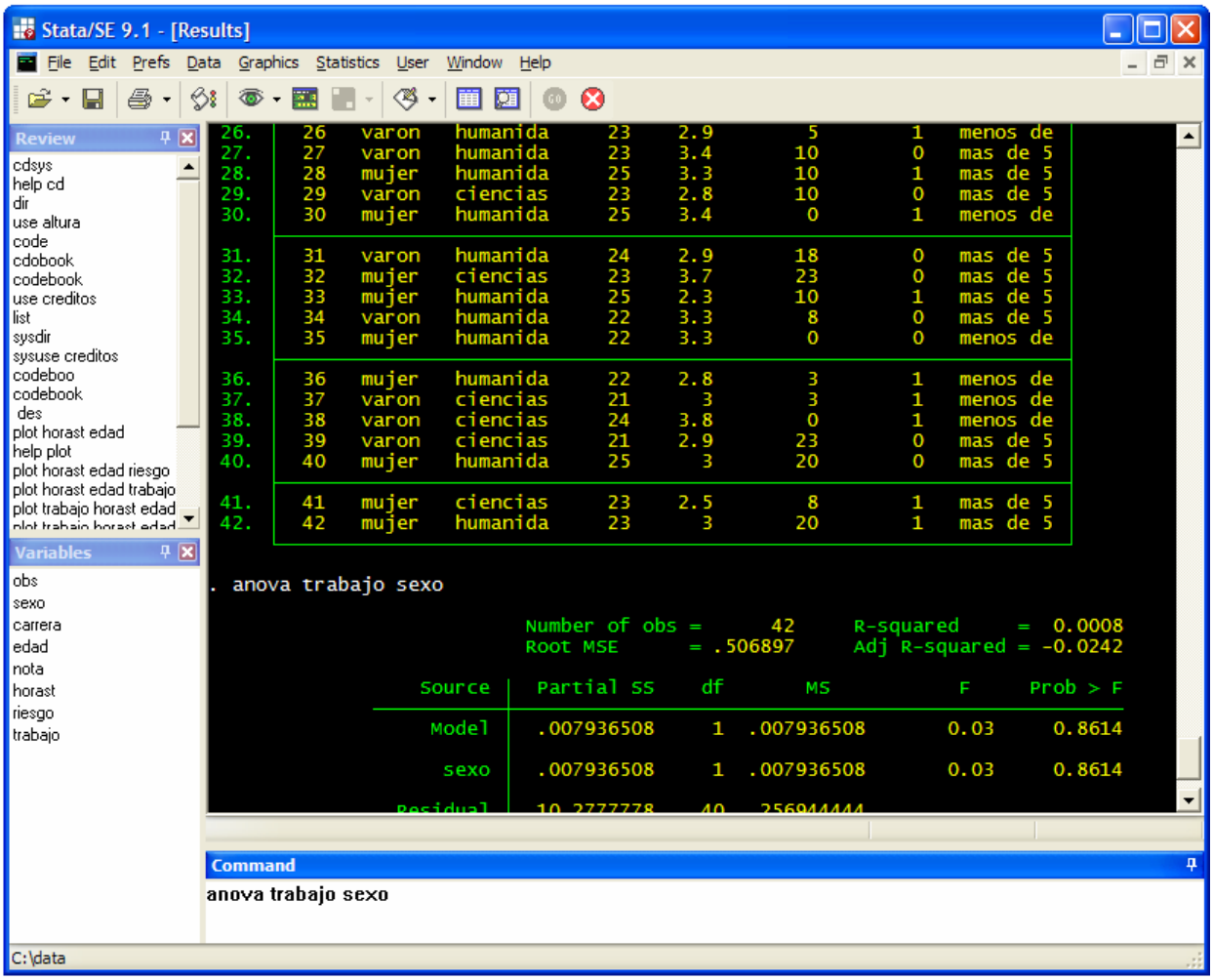
En cuanto a la capacidad de manejar grandes volúmenes de información, a diferencia de otras aplicaciones como SPSS y SAS, Stata necesita hacer una copia la base de datos que vamos a analizar en la memoria RAM de nuestro computador, por tanto, la memoria disponible en nuestro computador deberá de estar acorde con el tamaño de las bases de datos que vamos a utilizar.

Existe una limitación en cuanto al **número máximo de variables**; en la versión 9.1 SE el número máximo de variables contenidas en la base de datos está en torno a las 35.000.

II. EL AMBIENTE DE TRABAJO DE STATA

La interface de Stata consiste en un entorno de trabajo que facilita la interacción con la aplicación:

El entorno de trabajo tiene el siguiente aspecto:



Este entorno puede ser configurado (hasta cierto punto) para adaptarlo a nuestras necesidades.

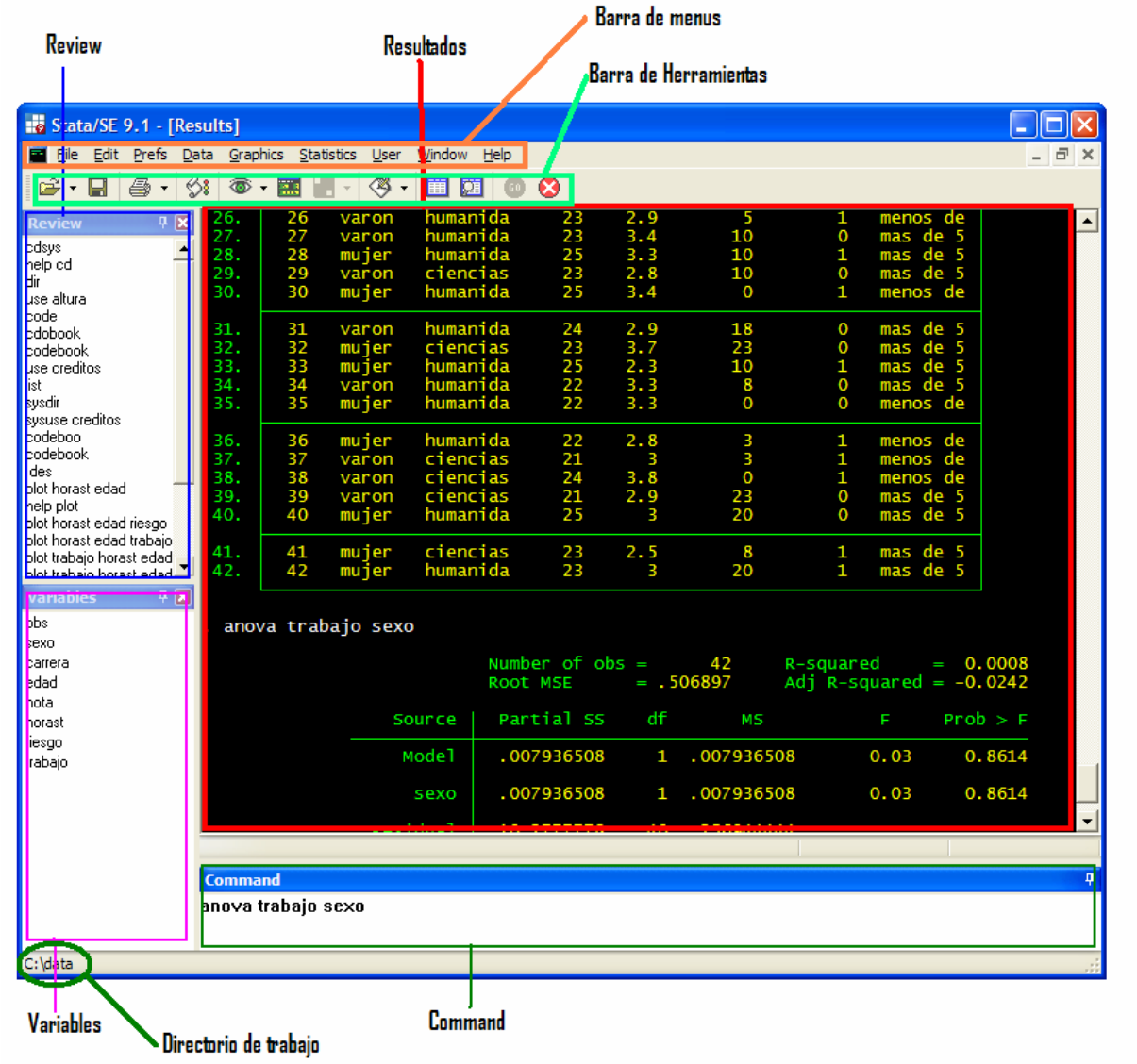
Descripción del ambiente de trabajo

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
Variables
Command
Result

La ubicación de estas ventanas es la siguiente:



Descripción de las ventanas

Command	En esta ventana se escriben las órdenes que se dan a la aplicación
Result	Aparecen los resultados de las órdenes ejecutadas
Variable	Listado de las variables contenidas en la base de datos cargada en la aplicación; si no tenemos ninguna, esta ventana aparece vacía
Review	Lista completa de los comandos ejecutados desde que se inició la aplicación

Nota:

En la esquina inferior izquierda de la pantalla aparece el directorio de trabajo; para cambiar de directorio basta con escribir el comando cd seguido del nuevo directorio, exactamente igual a como se trabajaba en la consola de MS Windows.

Ejemplo:

cd d:\datos\enuesta

Descripción de la barra de herramientas

	Abrir ficheros de datos en formato Stata
	Guardar el actual fichero de datos
	Imprimir resultados, gráficos y órdenes
	Iniciar o cerrar un archivo para guardar resultados
	Abrir el visor de ayuda
	Restaurar la ventana de resultados a primer plano
	Restaurar la ventana de gráficos de alta resolución a primer plano
	Creaf un nuevo fichero de comandos (equivalente a los ficheros de sintaxis de SPSS)
	Invocar al editor de datos, se pueden modificar datos
	Invocar al visualizador de datos, no podemos modificar datos
	Continuar con la ejecución de comandos
	Detener la ejecución de la tarea que está realizando

Ayuda del sistema

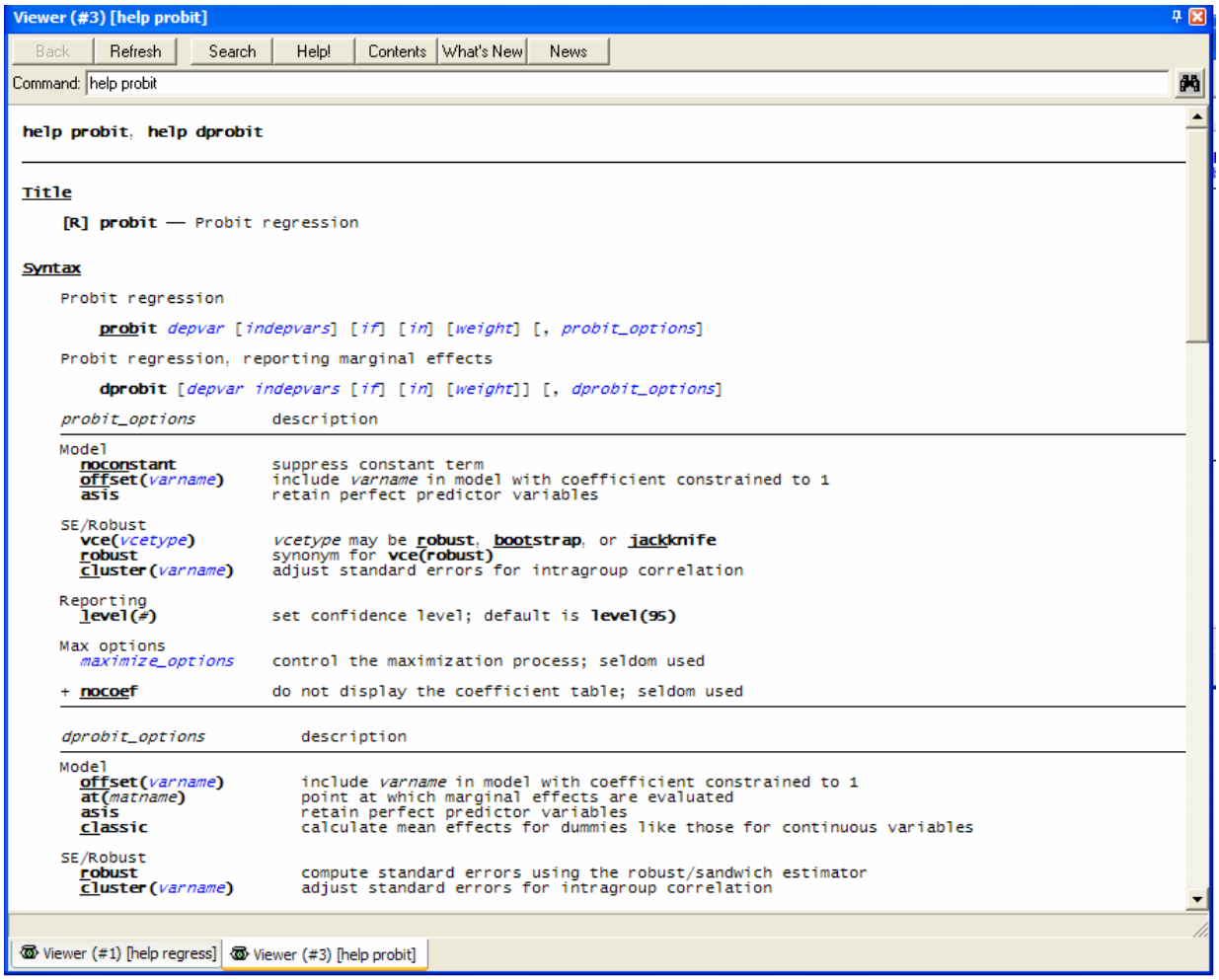
La aplicación Stata posee un sistema de ayuda realmente excelente (personalmente, es el mejor que conozco).

Para solicitar ayuda sobre un tema, por ejemplo regresión, sólo tenemos que escribir la orden **help** seguida de la palabra clave. Automáticamente se abre el visor de ayuda mostrando un completísimo informe, incluso con ejemplos y temas relacionados.

Ejemplo

help probit

Resultado (mostrado sólo parcialmente):



Sintaxis de las órdenes de Stata

La aplicación Stata posee un lenguaje de programación avanzado que respeta unas normas de sintaxis, al igual que otros lenguajes de programación como PASCAL o C++; quienes estén familiarizados con estos lenguajes les resultará muy sencillo asimilar el lenguaje de Stata.

Cualquier orden en Stata (con muy pocas excepciones) posee la siguiente sintaxis:

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

Pero la mayoría de las veces vamos a utilizar la siguiente forma, mucho más simplificada:

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

[…]	todo lo que aparece entre 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 el rango de observaciones que deseamos analizar
weight	sirve para indicar una variable de ponderación
options	son las opciones especificas del comando que estemos utilizando

Ejemplos:

table genero edad Realiza una tabla de contingencia del género y la edad

table genero edad if altura > 1.70 Realiza una tabla de contingencia del género y la edad sólo para personas de más de 1,70 de altura

table genero edad if altura > 1.70 in 1/100 Realiza una tabla de contingencia del género y la edad solo para personas de más de 1,70 de altura utilizando únicamente los 100 primeros casos de la base de datos

table genero edad if altura > 1.70 in 1/100 [weight = pondera] , chi En este comando, además, se indica que los datos van ponderados por la variable **pondera** y se solicita que realice el test Chi cuadrado

III. GESTION DE BASES DE DATOS

Los ficheros de datos en Stata se denominan **dataset**.

Un dataset es una tabla, donde las columnas representan variables y las filas observaciones o casos.

sexo	carrera	edad	nota	horast	riesgo
mujer	ciencias	25	4	5	0
varon	humanida	28	3.3	5	1
mujer	humanida	25	3.3	0	1
mujer	humanida	24	2.2	20	0
varon	humanida	23	2.9	5	1
varon	humanida	23	3.4	13	0
mujer	humanida	25	3.3	10	1
varon	ciencias	23	2.8	10	0
mujer	humanida	25	3.4	0	1
varon	humanida	24	2.9	18	0
mujer	ciencias	23	3.7	20	0
mujer	humanida	25	2.3	10	1
varon	humanida	22	3.3	5	0
mujer	humanida	22	3.3	0	1
mujer	humanida	22	2.8	0	1
varon	ciencias	21	3	3	1
varon	ciencias	24	3.8	3	1
varon	ciencias	21	2.9	23	0
mujer	humanida	25	3	20	0
mujer	ciencias	23	2.5	5	0
mujer	humanida	23	3	20	0
mujer	ciencias	25	4	8	0
varon	humanida	28	3.3	8	1
mujer	humanida	25	3.3	3	1
mujer	humanida	24	2.2	23	0
varon	humanida	23	2.9	5	1
varon	humanida	23	3.4	10	0
mujer	humanida	25	3.3	10	1
varon	ciencias	23	2.8	10	0
mujer	humanida	25	3.4	0	1

La variable del sistema **_N** indica el número total de observaciones contenidas en el dataset.

```
display _N
```

Para conocer la naturaleza de los datos contenidos en el dataset se utiliza la instrucción **describe**

Ejemplo:

```
. des

Contains data from C:\data\creditos.dta
  obs:            42
 vars:             8
 size:           630 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
obs             byte   %8.0g
sexo            byte   %8.0g       sexo       sexo
carrera         byte   %8.0g       carrera    tipo de carrera cursada
edad            byte   %8.0g                  edad en años
nota            float  %9.0g                  nota sobre 5.
horast          byte   %8.0g                  horas trabajadas a la semana
riesgo          str1   %1s                    evaluacion del riesgo
trabajo         byte   %8.0g       trabajo
-------------------------------------------------------------------------------
Sorted by:
```

Tipos de datos

La columna denominada **storage type** indica el formato de almacenamiento, es decir, el número de bytes y, por tanto, la precisión de la variable.

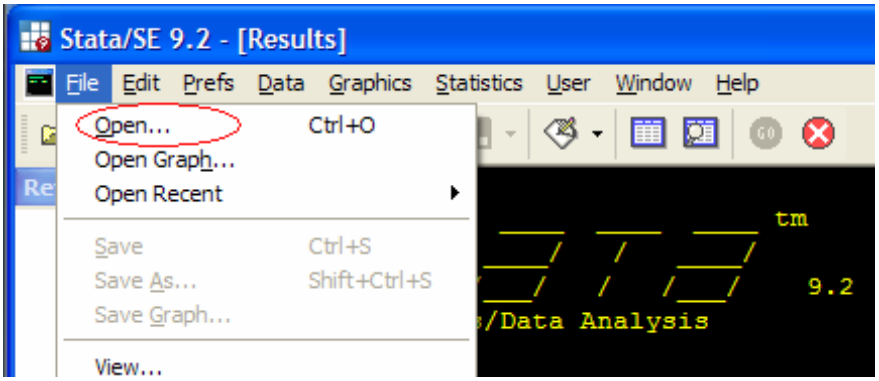
Tipo	menor valor	mayor valor	valor mas cercano de cero	bytes
byte	-127	100	+/-1 1	1
int	-32,767	32,74	+/-1	2
long	-2,147,483,647	2,147,483,620	+/-1	4
float	-1.70141173319*10^38	1.70141173319*10^36	+/-10^-36	4
double	-8.9884656743*10^307	8.9884656743*10^307	+/-10^-323	8

Precision		
	float	3.795x10^-8
	double	1.414x10^-16

Cargar un fichero de datos en formato Stata

Para cargar un fichero de datos en formato Stata (extensión ***.dta***) tenemos dos opciones: bien realizar esta operación desde la barra de menús, o bien mediante una instrucción escrita en la ventana command.

a) Desde la barra de menús



Si ya tenemos un fichero en memoria, el sistema nos advertirá de ello, permitiéndonos continuar o abortar la operación.

b) Instrucción en la ventana Command

use “nombre de fichero”

Si ya tenemos un fichero de datos en memoria, la aplicación no permitirá cargarlo en memoria pues borraría el anterior. Si este es nuestro caso, deberemos de utilizar la opción **clear** para desalojar de la memoria el anterior fichero de datos:

use “nombre de fichero”, clear

La sintaxis de este comando es bastante flexible, permitiendo controlar qué parte del fichero se desea cargar.

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

Introducir datos por teclado

Si ejecutamos la orden **edit** invocamos el editor de datos desde el cual podemos ir introduciendo los datos tal como haríamos en una hoja de cálculo Excel. También podemos realizar un **copy-past** aunque es poco recomendable.

Cargar los datos desde un fichero de datos en formato ASCII

El comando para realizar la lectura de ficheros ASCII es **infile**

La sintaxis (muy simplificada) de este comando es:

infile varlist using "nombre de fichero" [if] [in], options

Un ejemplo sencillo: supongamos que deseamos leer el siguiente fichero en formato ASCII con la siguiente estructura (sin los nombres de variables):

creditos.dat - Bloc de notas

Archivo Edición Formato Ver Ayuda

obs	sexo	carrera	edad	nota	horast	riesgo	trabajo
1	2	1	25	4	5	0	1
2	1	2	28	3.3	5	1	1
3	2	2	25	3.3	0	1	1
4	2	2	24	2.2	20	0	2
5	1	2	23	2.9	5	1	1
6	1	2	23	3.4	13	0	2
7	2	2	25	3.3	10	1	2
8	1	1	23	2.8	10	0	2
9	2	2	25	3.4	0	1	1
10	1	2	24	2.9	18	0	2
11	2	1	23	3.7	20	0	2
12	2	2	25	2.3	10	1	2
13	1	2	22	3.3	5	0	1
14	2	2	22	3.3	0	1	1
15	2	2	22	2.8	0	1	1
16	1	1	21	3	3	1	1

Escribiríamos:

```
infile obs sexo carrera edad nota horast riesgo trabajo using "C:\creditos.dat" , clear
```

Nota

Está disponible el programa **Stat/Transfer** para realizar la traducción de bases de datos entre distintos formatos, como por ejemplo SPSS, SAS, Stata, MS-Excel, …, etc.

Guardar ficheros de datos

En principio, Stata no guarda las modificaciones realizadas en la base de datos; por ello, si deseamos guardar el fichero de datos con las posibles modificaciones que hayamos realizado, entonces utilizaremos el comando **save**

La sintaxis de este comando es:

```
save [nombre de fichero] [, opciones]
```

Ejemplo:

```
save creditos, replace
```

Con la opción **replace** grabamos encima del fichero que ya existía.

Para ordenar el **dataset** en función a una serie de variables se utiliza el comando **sort**

```
sort by iden
sort by hogar iden
```

Unir conjunto de datos

Es muy común en el trabajo diario combinar varias bases de datos. En este manual vamos a mostrar dos operaciones básicas: añadir variables y añadir casos. Las instrucciones asociadas a estas operaciones son **merge** y **append**.

La instrucción Merge

Merge se utiliza para añadir variables, pero no observaciones. Los ficheros de datos deben de tener una variable de identificación y, además, deben de estar ordenados por dicha variable.

Ejemplo:

Data1

obs	sexo	carrera	edad	nota
1	mujer	ciencias	25	4
2	varon	humanida	28	3.3
3	mujer	humanida	25	3.3
4	mujer	humanida	24	2.2
5	varon	humanida	23	2.9
6	varon	humanida	23	3.4
7	mujer	humanida	25	3.3
8	varon	ciencias	23	2.8
9	mujer	humanida	25	3.4
10	varon	humanida	24	2.9
11	mujer	ciencias	23	3.7
12	mujer	humanida	25	2.3
13	varon	humanida	22	3.3
14	mujer	humanida	22	3.3
15	mujer	humanida	22	2.8
16	varon	ciencias	21	3
17	varon	ciencias	24	3.8
18	varon	ciencias	21	2.9
19	mujer	humanida	25	3
20	mujer	ciencias	23	2.5
21	mujer	humanida	23	3
22	mujer	ciencias	25	4
23	varon	humanida	28	3.3
24	mujer	humanida	25	3.3
25	mujer	humanida	24	2.2
26	varon	humanida	23	2.9
27	varon	humanida	23	3.4
28	mujer	humanida	25	3.3
29	varon	ciencias	23	2.8
30	mujer	humanida	25	3.4

Data2

obs	nota	horast	riesgo	trabajo
1	4	5	0	menos de
2	3.3	5	1	menos de
3	3.3	0	1	menos de
4	2.2	20	0	mas de 5
5	2.9	5	1	menos de
6	3.4	13	0	mas de 5
7	3.3	10	1	mas de 5
8	2.8	10	0	mas de 5
9	3.4	0	1	menos de
10	2.9	18	0	mas de 5
11	3.7	20	0	mas de 5
12	2.3	10	1	mas de 5
13	3.3	5	0	menos de
14	3.3	0	1	menos de
15	2.8	0	1	menos de
16	3	3	1	menos de
17	3.8	3	1	menos de
18	2.9	23	0	mas de 5
19	3	20	0	mas de 5
20	2.5	5	0	menos de
21	3	20	0	mas de 5
22	4	8	0	mas de 5
23	3.3	8	1	mas de 5
24	3.3	3	1	menos de
25	2.2	23	0	mas de 5
26	2.9	5	1	menos de
27	3.4	10	0	mas de 5
28	3.3	10	1	mas de 5
29	2.8	10	0	mas de 5
30	3.4	0	1	menos de

La variable de identificación es **obs** y los dos **dataset** están ordenados de forma ascendente.

Ejecutamos la orden

```
use data1
merge obs using data2
```

y obtenemos:

obs	nota	horast	riesgo	trabajo	sexo	carrera	edad
1	4	5	0	menos de	mujer	ciencias	25
2	3.3	5	1	menos de	varon	humanida	28
3	3.3	0	1	menos de	mujer	humanida	25
4	2.2	20	0	mas de 5	mujer	humanida	24
5	2.9	5	1	menos de	varon	humanida	23
6	3.4	13	0	mas de 5	varon	humanida	23
7	3.3	10	1	mas de 5	mujer	humanida	25
8	2.8	10	0	mas de 5	varon	ciencias	23
9	3.4	0	1	menos de	mujer	humanida	25
10	2.9	18	0	mas de 5	varon	humanida	24
11	3.7	20	0	mas de 5	mujer	ciencias	23
12	2.3	10	1	mas de 5	mujer	humanida	25
13	3.3	5	0	menos de	varon	humanida	22
14	3.3	0	1	menos de	mujer	humanida	22
15	2.8	0	1	menos de	mujer	humanida	22

La instrucción Append

Se utiliza para añadir observaciones. La sintaxis de este comando es mucho más sencilla.

Dataset 1

obs	sexo	edad
2	varon	28
5	varon	23
6	varon	23
8	varon	23
10	varon	24
13	varon	22
16	varon	21
17	varon	24
18	varon	21
23	varon	28
26	varon	23
27	varon	23
29	varon	23
31	varon	24
34	varon	22
37	varon	21
38	varon	24
39	varon	21

Dataset 2

obs	sexo	edad
1	mujer	25
3	mujer	25
4	mujer	24
7	mujer	25
9	mujer	25
11	mujer	23
12	mujer	25
14	mujer	22
15	mujer	22
19	mujer	25
20	mujer	23
21	mujer	23
22	mujer	25
24	mujer	25
25	mujer	24
28	mujer	25
30	mujer	25
32	mujer	23
33	mujer	25
35	mujer	22
36	mujer	22
40	mujer	25
41	mujer	23
42	mujer	23

Así, para combinar estas dos bases de datos, utilizamos los siguientes comandos:

```
use data1
append using data2
```

Si deseamos construir una nueva base de datos que condense la información original, podemos utilizar el comando **collapse**

Ejemplo

Supongamos que tenemos una base de datos de hogares y en cada hogar se ha realizado un muestreo de personas. Deseamos construir una base de datos donde la unidad muestral sea el hogar, con una variable **edad** que represente la edad media de las personas que viven en dicho hogar.

pa_s	relig	alfabet	tasa_nat	tasa_mor
Acerbaján	Musulma.	98	23	7
Afganistán	Musulma.	29	53	22
Alemania	Protest.	99	11	11
Arabia Saudí	Musulma.	62	38	6
Argentina	Católica	95	20	9
Armenia	Ortodoxa	98	23	6
Australia	Protest.	100	15	8
Austria	Católica	99	12	11
Bahrein	Musulma.	77	29	4
Bangladesh	Musulma.	35	35	11
Barbados	Protest.	99	16	8.4
Bélgica	Católica	99	12	11
Bielorusia	Ortodoxa	99	13	11
Bolivia	Católica	78	34	9
Bosnia	Musulma.	86	14	6.39
Botswana	Tribal	72	32	8
Brasil	Católica	81	21	9
Bulgaria	Ortodoxa	93	13	12
Burkina Faso	Animista	18	47	18
Burundi	Católica	50	44	21

```
collapse (mean) alfabet  tasa_nat tasa_mor , by (relig)
```

relig	alfabet	tasa_nat	tasa_mor
	76.00	34.00	8.00
Animista	39.50	44.25	15.25
Budista	83.14	22.09	8.08
Católica	84.15	23.83	9.61
Indú	52.00	29.00	10.00
Judía	92.00	21.00	7.00
Musulma.	60.00	35.63	8.38
Ortodoxa	96.75	14.25	10.25
Protest.	92.75	18.00	10.90
Taoista	83.00	18.50	6.50
Tribal	72.00	32.00	8.00

Resumen

En los capítulos previos hemos visto el funcionamiento básico de la aplicación, introduciendo de forma progresiva los principios más elementales de la importación y gestión de las bases de datos.

IV. CREACIÓN Y MODIFICACIÓN DE VARIABLES

Una vez cargada la base de datos que deseamos analizar es muy común modificarla, creando nuevas variables o bien transformado las existentes.

Básicamente existen tres formas de crear y modificar las variables contenidas en la base de datos:

a) Generar nuevas variables en función a expresiones matemáticas ya existentes.

b) Reemplazar algunos o todos los valores de una variable en función a una regla.

c) Agrupar los valores en intervalos prefijados, es decir recodificar.

a) Generar nuevas variables en función de expresiones matemáticas ya existentes

Para crear nuevas variables en función de expresiones numéricas se utiliza el comando **generate**

La sintaxis de generate es:

generate [tipo] nueva_variable [: Etiqueta] = exp [if] [in]

Ejemplo

Generate precio_pesetas = precio*166.

Nota
Si la variable a crear ya existe, el sistema dará un mensaje de error, pues este comando no permite cambiar o alterar los valores de una variable ya existente. Si lo que deseamos es alterar los valores, entonces deberemos utilizar el comando **replace** , que tiene la misma sintaxis que **generate**

El número de funciones que podemos utilizar con el comando **generate** es realmente amplio; a modo ilustrativo presentamos las más usuales:

Funciones aritméticas
+
-
*
/
^

Funciones matemáticas
abs(x)
acos(x) arcocoseno de x; -1 < x < 1
asin(x) arcoseno de x; -1 < x < 1
atan(x) arcotangente de x
ceil(x) retorna el entero mas pequeño mayor que x; n-1<x<=n
comb(n,k) numero de combinaciones posibles de n elementos tomados de k en k
cos(x) coseno de x
exp(x) exponencial de x, la function inversea es ln(x)
int(x) retorna el enetero de truncar x; int(1.2) = 1, and int(-1.2) = -1
ln(x) logaritmo en base e
log10(x) logaritmo en base 10
logit(x) logit de x, logit(x) = ln(x/(1-x))
max(x1,x2,...,xn) retorna el maximo de x1, x2, ..., xn
min(x1,x2,...,xn) retorna el minimo de x1, x2, ..., xn
mod(x,y) retorna el modulo de x respecto de y, mod(x,y) = x - y*int(x/y)
sin(x) seno de x
sqrt(x) raiz cuadrada de x
sum(x) suma acumulada de x
tan(x) tangente de x

Nota

Todas las funciones trigonométricas están en radianes.

b) El comando egen

El comando **egen** es una extensión del comando **generate** . El comando **egen** genera variables en función de valores de otras variables; por ejemplo:

egen sdEdad= sd(edad) , by sexo

Genera una variable que contiene la desviación estándar de la edad en cada género.

Es importante saber que la aplicación de este comando desordena el fichero de datos.

c) Recodificación de variables

Para recodificar variables, tanto continuas como discretas, se puede utilizar el comando **recode** .

recode variable (regla) [(regla) ...] [, generate(nueva variable)]

Algunos ejemplos

recode edad 0/25 =1 25/50 = 2 50/ max = 3, gen(edad_agrupada)
recode x (1 2 3 =1) (4 5 6=2), gen(n_x)

Generación de retardos y diferencias

Cuando se está trabajando con series temporales es habitual tener que utilizar el operador de retardos.
La forma de generar retardos de orden k es la siguiente:

gen temperatura_k=temperatura [_n-k]

Nota

Es indispensable que el dataset esté correctamente ordenado

y para generar diferencias de orden k:

gen dtempk=*temp-temp[_n-k]*

V. ESTADÍSTICOS DESCRIPTIVOS UNIVARIANTES

En general, todo análisis estadístico, por complejo que sea, empieza por un completo análisis descriptivo. A continuación presentamos los comandos más utilizados.

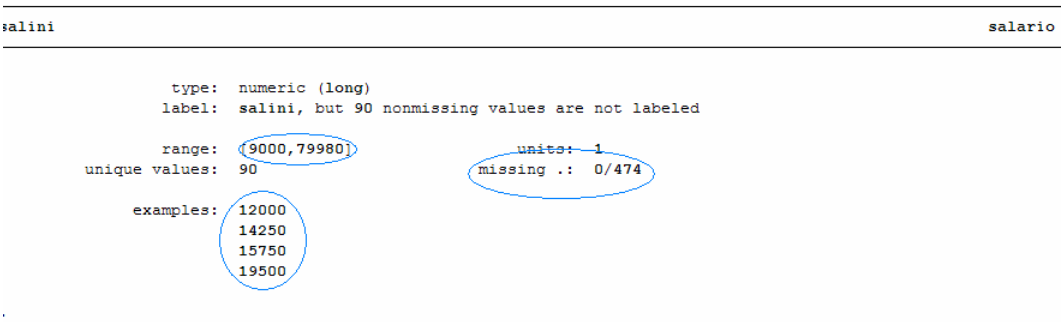
1) Para crear el libro resumen de las variables está el comando **codebook** .

codebook lista de variables

Si omitimos la lista de variables entonces realiza un informe de todas las variables contenidas en el dataset.

Ejemplo:

```
codebook  salini
```



2) Para solicitar estadísticos de resumen como la media, varianza, …, etc., se utiliza el comando **sumaries** .

Ejemplo:

```
sumamarize  salini  salario  educ
```

Variable	Obs	Mean	Std. Dev.	Min	Max
salini	474	17016.09	7870.638	9000	79980
salario	474	34419.57	17075.66	15750	135000
educ	474	13.49156	2.884846	8	21

3) Para construir tablas de estadísticos resumen se puede utilizar el comando **tabstat** .

```
tabstat salario salini , stat (min median mean  max n cv )
```

stats	salario	salini
min	15750	9000
p50	28875	15000
mean	34419.57	17016.09
max	135000	79980
N	474	474
cv	.4961033	.462541

En la última fila figura el coeficiente de variación.

4) Para solicitar tablas de frecuencias se utiliza el comando **tabulate**:

```
tabulate clima
```

```
. tabulate  clima
```

clima predominante	Freq.	Percent	Cum.
desierto	7	6.54	6.54
árido / desierto	5	4.67	11.21
árido	6	5.61	16.82
4	5	4.67	21.50
tropical	32	29.91	51.40
mediterráneo	10	9.35	60.75
marítimo	4	3.74	64.49
templado	34	31.78	96.26
ártico / temp.	4	3.74	100.00
Total	107	100.00	

También con este comando podemos solicitar tablas de doble entrada:

```
tabulate   clima  region ,   chi
```

clima predominante	región económica ocde	europa or	asia / pa	África	oriente m	américa l	Total
desierto	0	0	0	6,667	117,500	0	124,167
árido / desierto	0	0	0	43,900	93,961	0	137,861
árido	17,800	0	20,500	0	8,000	115,450	161,750
tropical	0	0	565,200	209,359	0	275,256	1,049,815
mediterráneo	58,100	10,400	1,037,100	58,400	9,120	5,200	1,178,320
marítimo	10,500	8,100	0	0	0	0	18,600
templado	565,387	158,500	1,401,400	14,400	67,600	51,100	2,258,387
ártico / temp.	43,000	149,200	0	0	0	0	192,200
Total	694,787	326,200	3,024,200	332,726	296,181	447,006	5,121,100

Pearson chi2(35) = 7.4e+06 Pr = 0.000

5) Con el comando **table** podemos crear tablas de estadísticos resumen controlando el contenido de cada casilla; por ejemplo, podemos calcular una tabla de doble entrada y situar en cada celda el estadístico solicitado.

```
table sexo minoría ,  cont (  mean salario) col row format (%9.1f)
```

```
. table sexo minoría ,   cont (  mean salario) col row format (%9.1f)
```

sexo	clasificación de minorías no	sí	Total
h	44475.4	32246.1	41441.8
m	26706.8	23062.5	26031.9
Total	36023.3	28713.9	34419.6

6) Para calcular el coeficiente de correlación de Pearson se utiliza el comando **correlate**:

```
correlate  espvidaf  espvidaf alfabet  pib_cap  calorías)
```

	espvidaf	espvidaf	alfabet	pib_cap	calorías
espvidaf	1.0000				
espvidaf	1.0000	1.0000			
alfabet	0.8693	0.8693	1.0000		
pib_cap	0.6759	0.6759	0.6274	1.0000	
calorías	0.7757	0.7757	0.6816	0.7597	1.0000

Combinando los comandos y ajustando los formatos podemos crear tablas de resumen realmente atractivas:

Ejemplo:

```
table region , cont (mean pib_cap sd pic_cap count pib_cap) format (%8,2f)
```

```
. table región ,cont( mean pib_cap sd pib_cap count pib_cap) format (%8.2f)
```

región económica	mean(pib_cap)	sd(pib_cap)	N(pib_cap)
ocde	16610.86	3725.97	21
europa oriental	5159.79	1708.70	14
asia / pacífico	4263.00	6291.05	17
África	998.68	1178.26	19
oriente medio	4957.41	4057.45	17
américa latina	1997.67	1482.12	21

VI. MODELOS DE REGRESIÓN POR MÍNIMOS CUADRADOS

Introducción

A diferencia de otras aplicaciones, en Stata los modelos de regresión se ejecutan en dos fases claramente diferenciadas:

- Estimación de los parámetros del modelo.
- Diagnosis del modelo estimado.

El comando para realizar una estimación de los parámetros de un modelo de regresión lineal es:

```
regress depvar [indepvars] [if] [in] [weight] [, options]
```

Por ejemplo:
reg altura edad peso

Un ejemplo un poco más elaborado:
reg altura edad peso , beta noconstant

Diagnósticos sobre el modelo de regresión lineal simple

Con los parámetros del modelo estimados es habitual realizar una serie de diagnósticos para contrastar si se cumplen determinadas hipótesis.

Contraste de homocedasticidad	estat hettest
Contraste de multicolinealidad	estat vif Nota: VIF(Xk) = $1 - R^2(Xk, X1...Xp)$
Contraste RSET	estat ovtest
Para guardar las estimaciones en una variable	predict nombre de variable

Ejemplo:

```
regres  alfabet alfabfem  espvidaf espvidam, beta
estat hettest
estat vif
predict alfabet_pronos
```

VII. REGRESIÓN LOGISTICA

Introducción

Las fases de un análisis de regresión logística son las siguientes:

- Estimación de los parámetros.
- Estudio de la capacidad de predicción del modelo.
- Interpretación de los parámetros.
- Generación de gráficos auxiliares.

Estimación de los parámetros

El comando para realizar una estimación de los parámetros de un modelo de regresión logística es:

```
logit depvar [indepvars] [if] [in] [weight] [, options]
```

Por ejemplo:

logit voto genero ingresos estudios

Las opciones más importantes son:

- **or**: muestra los ODDS Ratio.
- **noconstat**: suprime la constante del modelo.

Ejemplo:

```
Logit voto genero ingresos estudios, or
```

Este comando puede ser combinado con el procedimiento **stepwise** para conseguir el mejor conjunto de variables regresoras:

stepwise, pr(0.1): logit voto genero ingresos estudios

Estudio de la capacidad de predicción del modelo

Para estudiar la capacidad de predicción del modelo comparamos las estimaciones ofrecidas por el modelo con los datos reales mediante el comando:

estat class

Así mismo podemos guardar en variables las estimaciones mediante el comando **predict** para, posteriormente, realizar contrastes personalizados:

predict yhat, p	Guarda en la variable yhat la probabilidad estimada
predict lfor, xb	Guarda en la variable lfor el logit

Estudio de los parámetros

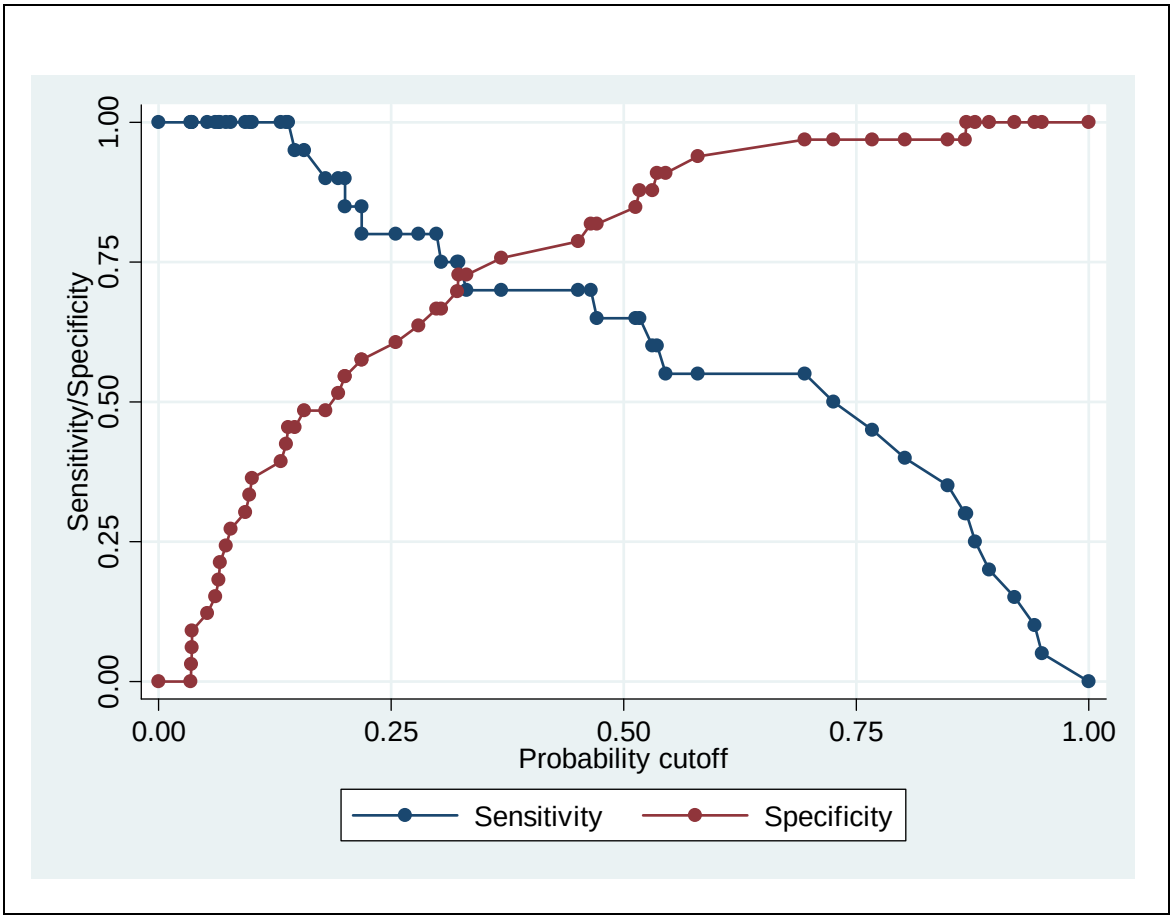
El comando **mfx** muestra las derivadas parciales evaluadas en el centro de gravedad de la distribución o bien en puntos concretos.

mfx	Muestra las derivadas parciales evaluadas en el centro de gravedad de la distribución
mfx , at(sex=1, income=2000)	Muestra las derivadas parciales calculadas en el punto indicado

Generación de gráficos auxiliares

Las curvas de sensibilidad y especificidad nos ayudaraán a calibrar el modelo.Las podemos calcular mediante los siguientes comandos:

lsens
Lroc



VIII. LISTA DE COMANDOS

Comandos generales

clear	Eliminar el fichero de datos actual
display	Mostrar valores
cd	Cambiar el directorio de trabajo
exit	Salir de la aplicación
help	Ayuda sobre el tema solicitado
cd	Cambiar de directorio de trabajo
save	Guardar el actual conjunto de datos
use	Cargar un conjunto de datos en formato Stata
set memory	Reservar una cantidad de memoria concreta para el trabajo
dir	Mostrar el contenido del directorio de trabajo

Conjunto de datos y variables

collapse	Cambiar la unidad muestral
encode	Recodificación automática
describe	Describir las variables del conjunto de datos
destring	Convertir una variable cadena a numérica
drop	Eliminar variables y observaciones
by varlist	Analizar por grupos
encode	Recodificar automáticamente cadenas a números
format	Definir un formato
generate	Crear nueva variable
infile	Leer datos en formato ASCII
input	Introducir datos por teclado
label	Añadir / cambiar etiquetas
list	Listar casos
merge	Combinar ficheros
move	Cambiar la secuencia de variables en el actual dataset
log using	Resultados a fichero
log close	Cerrar el fichero de resultados
order	Reordenar la secuencia de variables
recode	Recodificar variable
rename	Renombrar variable
replace	Cambiar el contenido de variable
set	Ajustar parámetros opcionales
sort /gsort	Ordenar el dataset

Gráficos

hbar	Histogramas
scatter	Diagramas de dispersión

Estadísticos descriptivos

codebook	Resumen de variables
hist	Histograma
summarize	Estadísticos muestrales
table	Tablas de doble entrada
tabulate	Tablas de 1 y 2 dimensiones con estadísticos resumen

Análisis estadísticos habituales

anova	Análisis de varianza
correlate	Correlación
oneway	Análisis de varianza
ranksum	Test de Wilcoxon
tabulate	crosstables (incl. tests de homogeneidad, gamma, exact r*c test)
table	Tablas de estadísticos resumen
ttest	Contraste de medias

Modelos de regresión

clogit	Regresión logística condicional
logit	Regresión logística
logistic	Regresión logística
Poisson	Regresión de Poisson
predict	prediction + indicator of fit
regress	Regresión lineal
Variable dummy: xi: comando …i . variable	