Curs bàsic d'Anàlisi de dades amb Stata

Contingut

- Sessió 1
  - **Introducció al Stata**
  - **Gestió d'arxius amb Stata**
  - **Manipulación de datos con Stata**
  - **Exercici pràctic**
- Sessió 2
  - **Estadística descriptiva**
  - **Grandaria Mostral**
  - **Exercici pràctic**
- Sessió 3
  - **Estimació i Contrast d'Hipòtesi**
  - **Correlació i Regressió**
  - **Exercici pràctic**
- Sessió 4
  - **Regressió lineal**
  - **Regressió logística**
  - **Análisi de supervivència**

Sessió 1

- Introducció al Stata
  - **Característiques generals**
  - **Menús**
  - **Ajuda**
  - **Forma de treball en Stata**
- Gestió d'arxius en Stata
  - **Entrada de dades**
  - **Obrir i desar dades**
  - **Combinar dades**
- Manipulació de dades amb Stata
  - **Definir i etiquetar variables**
  - **Transformar i recodificar variables**
  - **Crear noves variables**
  - **Control de duplicats**
- Exercici pràctic

Introducción

- Stata programa estadístico disponible para diversos sistemas operativos
- Fácil manejo de datos con mucha versatilidad para combinar y generar nuevos datos
- Numerosos tipos de análisis estadísticos sencillos y complejos con posibilidad de modificarlos y añadir nuevos métodos elaborados por los usuarios
- Muy utilizado en ambientes epidemiológicos
- Puede trabajar por menú, pero es mejor trabajar por comandos que se ejecutan al instante, dispone de una ayuda exhaustiva y completa y fácil de generar funciones o trabajar con programas que ejecuten varias ordenes a la vez

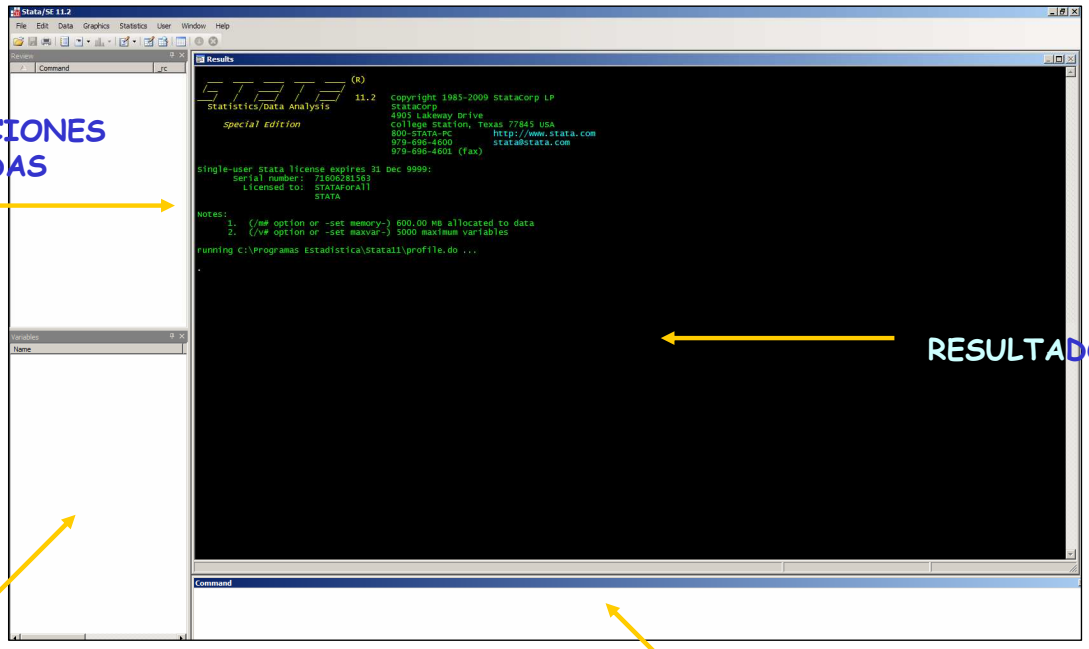
Extensiones de los ficheros de Stata

- .dta: Ficheros de datos en formato STATA
- .log: Fichero de texto con resultados
- .do: Fichero con instrucciones STATA
- .ado: Ficheros con macro/funciones de Stata
- .gph: Ficheros de gràficos

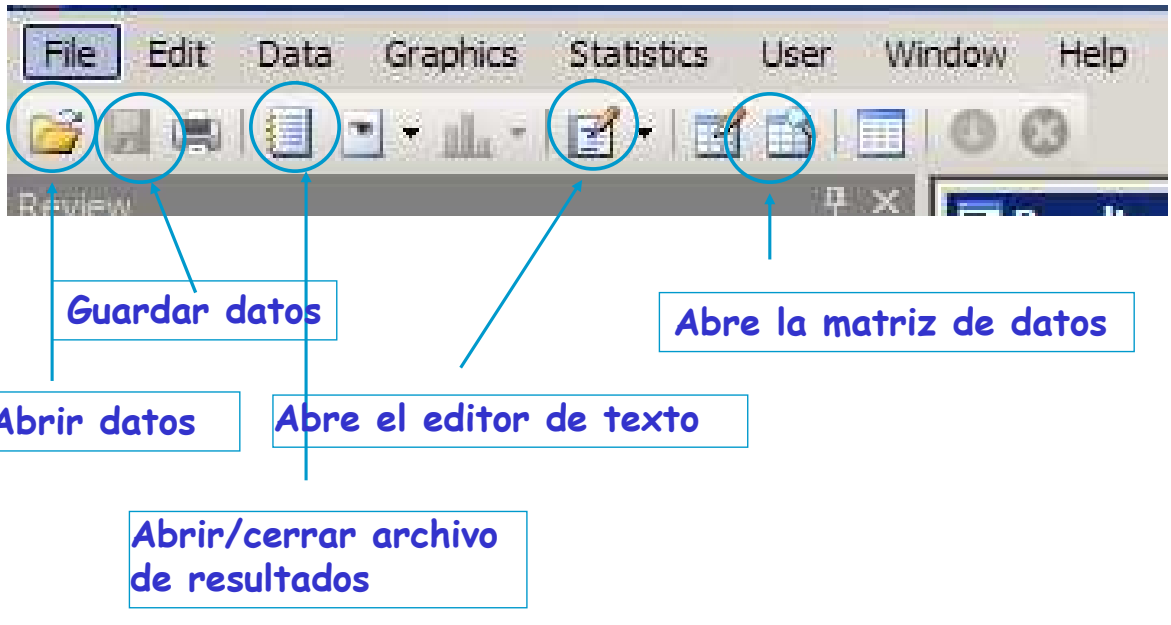
Algunas cosas que hay que saber

- Stata distingue entre mayúsculas y minúsculas. No es lo mismo var1 que Var1
- Por defecto trabaja con 1024K de memoria, muchas veces insuficiente.
- La memoria se amplia utilizando el comando
  `set memory 20m`
- El directorio por defecto es c:\data
- Los comandos pueden ser acortados a 3 primeras letras
- Se debe actualizar el Stata de vez en cuando
  `update all`

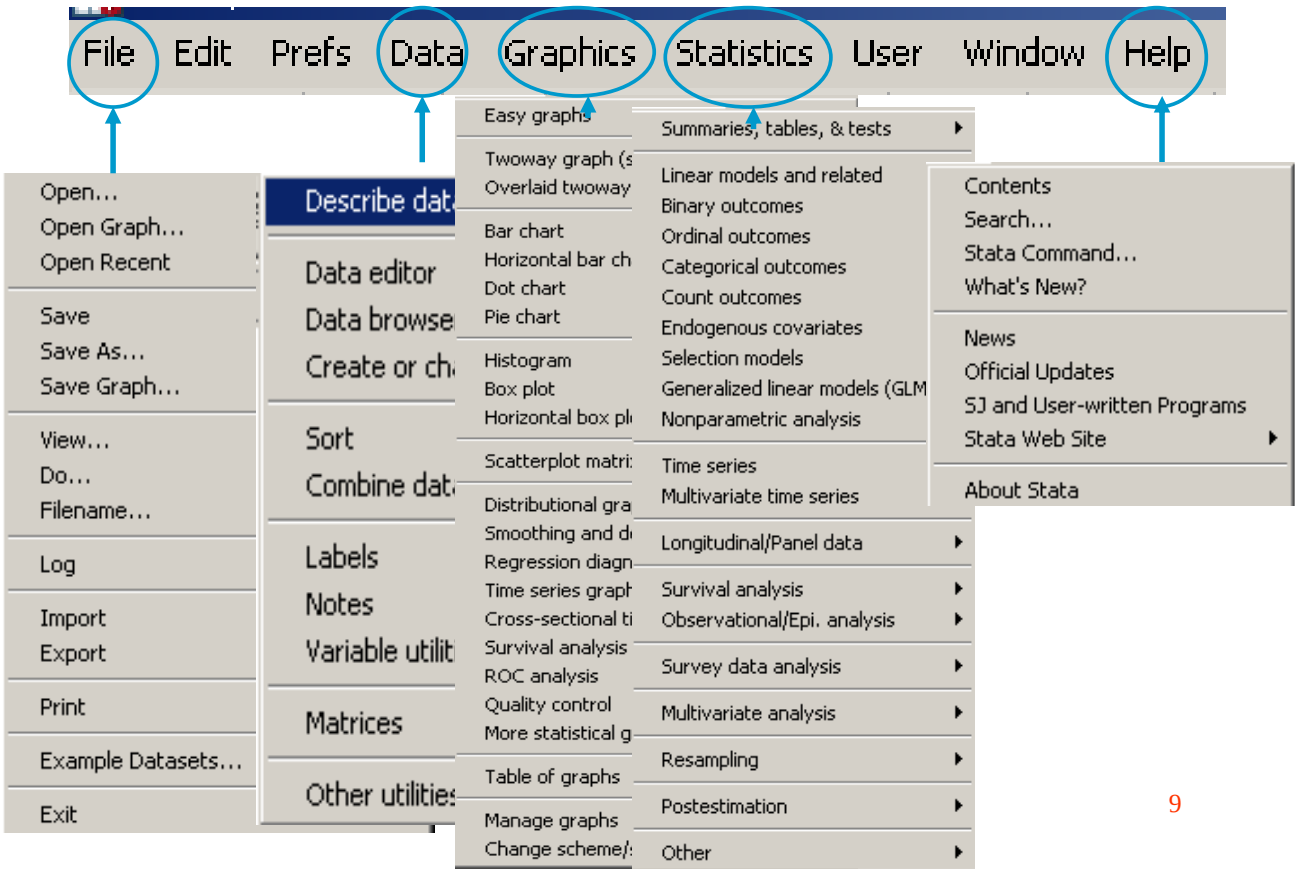
Ventanas de STATA



Barra de botones de Stata



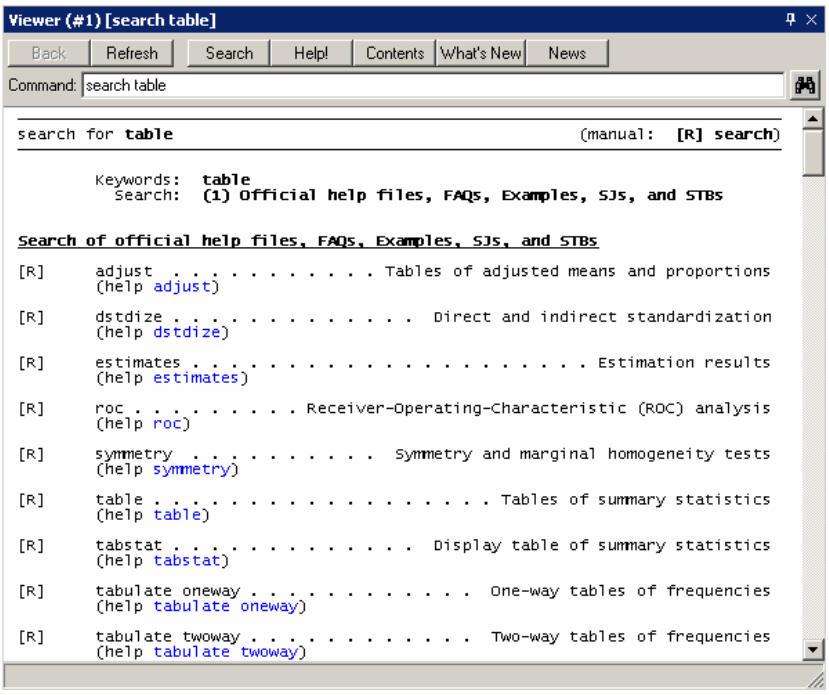
Menu de Stata



Ayuda

- Help comando

  `help table`

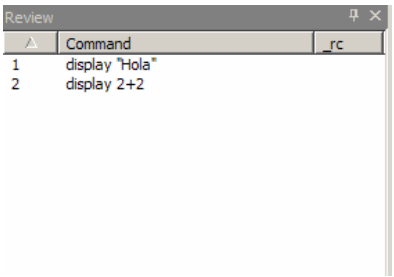


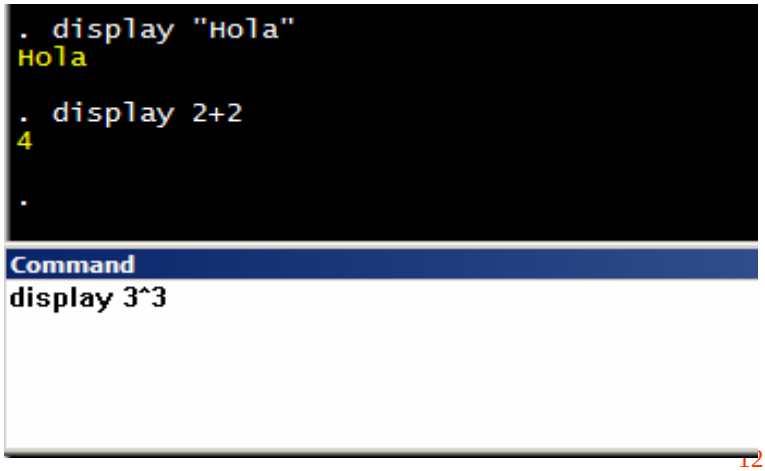
Modos de trabajar en Stata

- Escribiendo instrucciones en la línea de comandos ejecutando una a una y viendo el resultado por pantalla sin guardarlo
- Escribiendo varias instrucciones en un fichero .do y ejecutándolas en lote
- Es la forma óptima de trabajar

Escribiendo instrucciones en línea

- Se puede utilizar como una calculadora
- Los comandos ejecutados previamente se pueden recuperar utilizando la tecla RePàg o clickando sobre el en la ventana de comandos





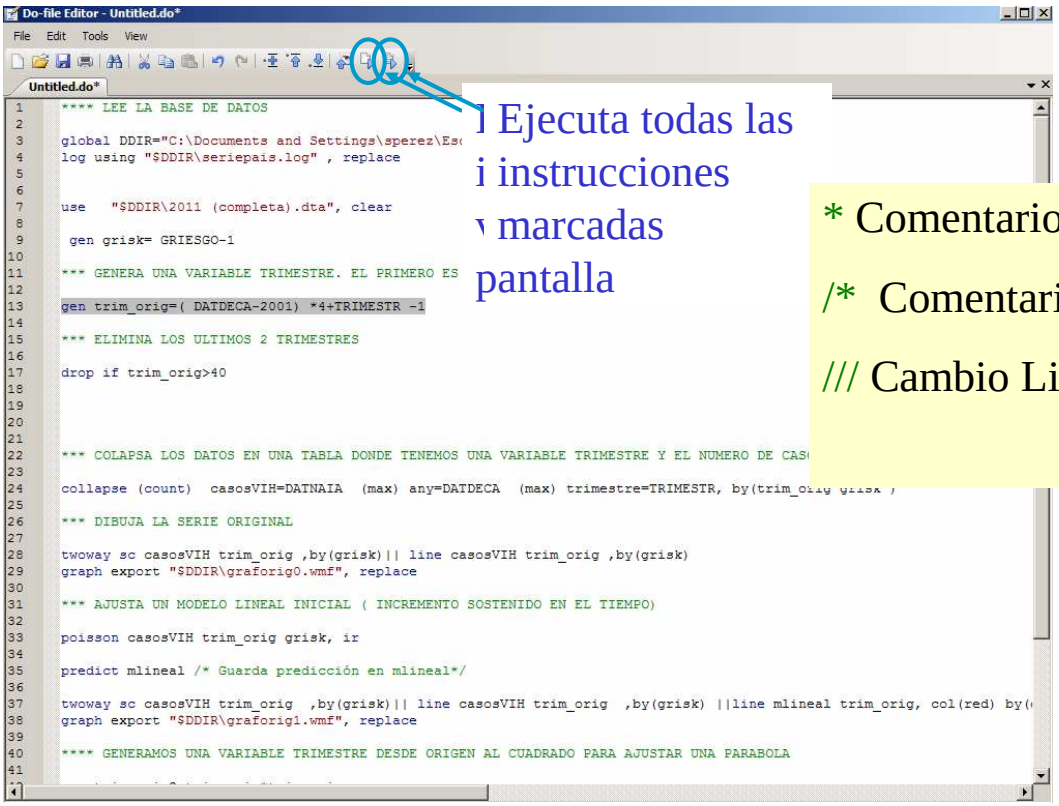
Estructura de los Comandos

comando *lista de variables*
condición(if), opciones

- **Ejemplos:**

```
tabulate  grupedad  sexo , row col
gen edad_15=edadsero-15
drop if cd4>500
xi:poisson iam i.estrés i.sexo, exp(perany)
```


Fichero Do



Guardar resultados

- Todos los resultados se pueden guardar un fichero de resultados.
- Por defecto se graban en formato .smcl y sólo se ven desde el visor
- Si se quiere ver en otro formato se debe de usar el formato texto

```
log using nombrefichero                [abre fichero .smcl]
log using nombrefichero, replace       [reemplaza fichero]
log using nombrefichero, append        [añade a  fichero]
log using nombrefichero.log  , text    [empieza fichero texto]
....
log off                          [pausa fichero texto]
log on                          [reinicia fichero texto]
log close                        [cierra el fichero de resultados]
```

Activa y desactiva fichero que guarda resultados

Gestión de bases de datos

- Se pueden introducir datos directamente con el editor de Stata
- Mejor cargar fichero transferido con Statransfer o grabarlo como .dta por otro programa (i.e. SPSS)
- Se puede cargar por el menu o con sintaxis

```
use  nomfichero, clear
```

- Se puede importar directamente desde excel o access usando ODBC
- Excel

```
odbc load, dsn("Excel Files;DBQ=C:/EST.xls")
  table("GeneralFV1$") clear datestring lower
```

- .CSV

```
insheet using  "C:/EST.csv") .clear delimiter(",")
  names
```

- Access

```
odbc load, dsn("MS Access Database;DBQ=C:/EST.mdb")
  table("tabla1") clear datestring lower
```

Gestión de bases de datos

- Para grabar ficheros se usa el menu o la sintaxis

```
save nomfichero, replace
```

- Exportar ficheros a excel

```
odbc insert, dsn("Excel Files;DBQ=C:/EST.xls")
  table("GeneralFV1$") create quoted

outsheet using "C:/EST.xls", delimiter(";") replace
```

- TRUCOS

  Definir el directorio de trabajo, para no tener que escribir cada vez la ruta

```
global Ddir "C:\GEMES\Datagemes_2011\sandoval\"
use "$DDir\Sandoval_2011.dta", clear
. . .
. . .
save "$DDir\Sandoval_2011.dta", replace
```

  Ó cambiar de directorio

```
cd C:\GEMES\Datagemes_2011\sandoval
```


Gestión de bases de datos

- La base de datos se ordena con

```
sort var1 var2…
gsort –var1 +var2…
```

- Las características de la base de datos se miran con

```
describe          [lista nombre variables y etiquetas]
codebook var1     [lista nombre, etiquetas y datos descriptivos]
```

- La tabla de datos se puede ver con

```
browse
browse var1 var2
```

- Y se puede ver y modificar con

```
edit
edit var1 var2…
```

- Los datos se listan con

```
list
list var1 var2…
```

Gestión de bases de datos

- Para borrar variables
  drop *var1 var2*
- Para borrar casos
  drop if *condición*
- Para mantener variables
  keep *var1 var2*
- Para mantener casos
  keep *var1 var2* **if** *condición*
- Repite comandos en un subconjunto de datos
  by *var1*,**sort**: **comando stata**
- TRUCO

  Genera un indicador del número de medición por paciente

  by *paciente*,**sort: gen** nvisita=_n

  Mantiene el primer caso de cada paciente

  by *paciente*,**sort: keep if** _n==1

  _n = Número de registro

  _N = Número total de casos

Gestión de bases de datos

- Para añadir casos a un fichero existente

  use *nomfile1*, **clear**
  append using *nomfile2*
  save *nomfile1+2*, **replace**

- Para añadir variables a un fichero existente

  use *nomfile1*, **clear**
  sort *variableclave*
  merge 1:1 variableclave **using** nomfile2, **sort**
  merge m:1 variableclave **using** nomfile2, **sort**
  merge 1:m variableclave **using** nomfile2, **sort**

  Añade una variable interna _merge que se codifica como sigue

1	master	La observación aparece sólo en el fichero 1(master)
2	using	La observación aparece sólo en el fichero 2 (using)
3	match	La observación aparece en los dos ficheros

  keep if _merge==3
  drop _merge
  save *nomfilevar1+2*, **replace**

Gestión de variables

- Para renombrar variables

```
rename nomvarviejo nomvarnuevo
```

- Para etiquetar la base de datos

```
label data "contenido de la base"
```

- Para etiquetar variables

```
label var var1 "etiqueta de la variable"
```

- Para etiquetar valores

```
label define nomormato valor1 "etiq1" valor2 "etiq2"
label val variable nomformato
```

- Para asignar formato a las variables

```
format varlist %fmt
```

- Truco (código para cambiar nombre variables a minúsculas)

```
unab listavar:*
foreach var of varlist `listavar' {
 cap ren `var' `=lower("`var'")'
}
```


Formato de variables

```
    %fmt         description                               example
------------------------------------------------------------------------------
Right-justified formats
    %#.#g        general numeric format                    %9.0g
    %#.#f        fixed numeric format                      %9.2f
    %#.#e        exponential numeric format                %10.7e
    %d           default numeric elapsed date format       %d
    %d...        user-specified elapsed date format        %dM/D/Y
    %#s          string format                             %15s
Right-justified, comma formats
    %#.#gc       general numeric format                    %9.0gc
    %#.#fc       fixed numeric format                      %9.2fc
Leading-zero formats
    %0#.#f       fixed numeric format                      %09.2f
    %0#s         string format                             %015s
Left-justified formats
    %-#.#g       general numeric format                    %-9.0g
    %-#.#f       fixed numeric format                      %-9.2f
    %-#.#e       exponential numeric format                %-10.7e
    %-d          default numeric elapsed date format       %-d
    %-d...       user-specified elapsed date format        %-dM/D/Y
    %-#s         string format                             %-15s
Left-justified, comma formats
    %-#.#gc      general numeric format                    %-9.0gc
    %-#.#fc      fixed numeric format                      %-9.2fc
Centered formats
    %~#s         string format (special)                   %~15s
------------------------------------------------------------------------------
```

Creación de variables

- Para generar nuevas variables *(ver **help functions**)*

```
gen nomnuevavar = expresión
gen nomnuevavar = expresión if condiciónlogica
```

- Para reemplazar valores en una variable existente

```
replace nomvar = expresión if condiciónlogica
```

- Operadores lógicos

```
Igual (==), mayor(>), mayor o igual(>=), menor(<), menor o
  igual(<=), diferente (!=)
```

- *TRUCOS*

```
gen var_sino=(var1==valor) /* Genera variable 0=no 1=si */
gen data_nac = mdy(mes,dia,any) /* crea variable fecha */
gen num_ident = _n /* 1 número identificación por caso*/
gen random= uniform() /* nº aleatorios entre 0 y 1 */
gen varnum= real(vartexto) /* convierte var texto en nº */
gen varnoblanco= trim(vartexto) /*elimina texto en blanco */
gen seletxt= substr(vartexto,pos,len)  /*selecciona texto de
  longitud len desde la posicion pos en la variable texto
  i.e. substr("12/11/2012",1,2)=12
  substr("12/11/2012",4,2)=11 */
```


Creación de variables

- Para recodificar variables existentes

```
recode variable sint1 sint2..., generate(varnueva)
```

sintaxis	Ejemplo	Significado
# = #	3 = 1	3 recodifica a 1
# # = #	2 . = 9	2 Y . recodifican a 9
#/# = #	1/5 = 4	De 1 a 5 recodifican a 4
nonmissing = #	nonmiss = 8	resto de no perdidos a 8
missing = #	miss = 9	resto de perdidos a 9
else= #	else = 9	resto a 9

- Convertir variables en números cuando el contenido es número

```
destring variable_txt, replace
```

- Generar variable número con etiquetas a partir de variable texto

```
encode var_txt, gen(var_num) label
```

Creación de variables

- Para repetir por subgrupos de datos

  by *vargrupo*: **gen** *nomnuevavar* **=** *expresión*

  by *vargrupo*: **comando análisis**

- Genera variables especiales

  egen *nomvar* **=** *función(argumentos) , opciones*

anycount*(varlist)*, ***values*** *(numlist)*	**[cuenta apariciones de valores en variables]**
cut*(varname)*, ***at****(#,#,...,#)*	**[categoria en grupos]**
group*(var1 var2 ....)*	**[combina 2 variables]**
rowmean*(varlist)*	**[calcula la media de las variables de la lista]**
rowmax(min/total)*(varlist)*	**[elige el maximo(minimo/suma)]**

- Para generar variables dummy o ficticias

  tabulate *variablecat*, **gen(***vardummy***)**

- Para definir valores perdidos

  mvdecode *var*, **mv***(valor)*

  mvdecode *_all*, **mv***(valor)*

Control de duplicados

- Para identificar casos duplicados

  duplicates report *variables* [tabla casos duplicados]

  duplicates list *variables* [lista casos duplicados]

  duplicates drop *vars*,**force** [elimina casos duplicados]

  duplicates tag *vars* ,***gen****(nvar)*[marca casos duplicados]

- Para duplicar casos

  expand *#* **if** *condición*

Girar ficheros

- Convertir columnas en filas

```
reshape long inc , i(id) j(year)
```

- Convertir filas en columnas

```
reshape wide inc , i(id) j(year )
```

(wide form)

i		..... x_ij		
id	sex	inc80	inc81	inc82
1	0	5000	5500	6000
2	1	2000	2200	3300
3	0	3000	2000	1000

(long form)

i	j		x_ij
id	year	sex	inc
1	80	0	5000
1	81	0	5500
1	82	0	6000
2	80	1	2000
2	81	1	2200
2	82	1	3300
3	80	0	3000
3	81	0	2000
3	82	0	1000

Inspeccionar datos

- Para ver que todo es correcto se puede ver la estructura de los datos

```
describe
describe var1 var2...
```

- Se puede tener una pequeña descripción que permite ver si hay datos extraños y una pequeña descripción (freqüencia valores, medias, etc.)

```
codebook
codebook var1 var2
```

- Si todo es correcto ya estamos en condiciones de empezar el análisis estadístico propiamente dicho