

Programa de Especialización en Econometría Aplicada

Economía - Microeconometría II
UNI, Seminarios

Edinson Tolentino
Semestre 2022

Clase 1, 15 de mayo de 2022

1. STATA

Los primeros comandos del programa stata mas usados son: **merge**, **append**, **collapse**.

Respecto al comando **merge**, se tiene el siguiente detalle: este comando sirve para unir las variables que se encuentran en la base de datos abierta (master dataset) con las que se encuentran en otra base de datos (using dataset). Se unen de forma horizontal

1. Uno a Uno

`merge 1:1 varlist using filename , options`

2. Varios a Uno

`merge m:1 varlist using filename , options`

3. Uno a Varios

`merge 1:m varlist using filename , options`

4. Varios a Varios

`merge m:m varlist using filename , options`

data1.dta

	Nombres	Micro	Macro	Econom
1	Luis	10	13	15
2	Carlos	12	15	6
3	Jose	15	8	12
4	Manuel	11	9	5

data2.dta

	Nombres	Mate	Historia	Finan
1	Luis	17	13	13
2	Carlos	13	16	11
3	Jose	18	17	15
4	Manuel	10	11	8

```
use data1.dta, clear  
br  
  
use data2.dta, clear  
br  
merge m:m Nombres using "data1.dta"
```

Al unir las bases, se crea una variable llamada **merge** que se define de la siguiente forma:

	Nombres	Mate	Historia	Finan	Micro	Macro	Econom	_merge
1	Carlos	13	16	11	12	15	6	matched (3)
2	Jose	18	17	15	15	8	12	matched (3)
3	Luis	17	13	13	10	13	15	matched (3)
4	Manuel	10	11	8	11	9	5	matched (3)

Result	# of obs.
not matched	0
matched	4 (_merge==3)

- `merge== 1` si las observaciones contienen valores solo para las series de la base principal (master)
- `merge== 2` si las observaciones contienen valores solo para las series de la base secundaria (using)
- `merge== 3` si las observaciones contienen valores solo para las series de ambas bases (master y using)

```

tab _merge      // Tabular la variable _merge

tab _m          // Tabular la variable _merge

drop if _m==2    // Borrar registros si _merge==2

drop if _m==1    // Borrar registros si _merge==1

drop _m         // Eliminar la variable _merge

```

1. Encuesta Nacional de Hogares (ENAH)

La ENAH se realiza mediante un muestreo probabilístico en el cual se reconoce explícitamente la probabilidad que cada individuo tiene de ser seleccionado dado que se conocen las características de la población (marco muestral).

Dado que para construirla se parte de un marco censal (población), se conoce a priori la probabilidad de seleccionar una determinada unidad de estudio. Para lograr representatividad estadística se muestrean diversos estratos (tamaños de centros poblados), dentro de ellos se muestran conglomerados (grupos de 80 viviendas aprox.) y dentro de estos se seleccionan un conjunto de viviendas (6 en zonas urbanas y 8 en zonas rurales).

La única forma de identificar a un hogar es conocer su número de conglomerado, la vivienda y el hogar. Además, si se quiere identificar a un individuo dentro del hogar deberá añadirse el código de la persona a los datos anteriores. Todas estas variables en conjunto constituyen lo que se conoce como *llave de identificación*.

Es importante identificar que algunos módulos o capítulos de la encuesta son respondidos para todo el hogar (por ejemplo características de la vivienda) y otros para cada individuo (por ejemplo educación), pero por suerte, es posible hacer un *merge* entre ambas alternativas.

Muestreo Probabilístico

Los factores de expansión nos permiten pasar de la muestra al conjunto de la población. Ellos son necesarios también para tener en cuenta las diferentes tasas de muestreo según dominios geográficos, etc.

La mayoría de comandos de Stata pueden ponderar datos. Stata proporciona cuatro tipos de ponderaciones y la que más se emplea es la asociada a los factores de expansión, que se indica con la siguiente opción:

pweight [*weight – type = varname*]

pweight permite identificar los pesos que indican la inversa de la probabilidad de que la observación sea incluida debido al diseño del muestreo. Cuando no sea posible utilizarlo podemos optar por especificar **iweight (iw)**

2. Procesamiento de los módulos de ENAHO

En la siguiente sección se buscará trabajar con los módulos de la ENAHO, en específico: módulos 100 y sumaria. El objetivo es poder construir una base de datos a través de la sintaxis del *dofile*, que me permite construir una base de datos con variables predeterminadas.

La idea principal de la siguiente sección es poder construir una base de datos para poder analizar los determinantes (factores) que pueden aumentar o disminuir la probabilidad de un hogar en ser caracterizado como pobre (definición de pobreza desde el ingreso).

Variables	Descripción
rpobre	==1, si el hogar es pobre
r6	Ingreso mensual (Soles)
logr6	Logaritmo ingreso mensual (Soles)
rsexo	==1, mujer
r1r a	edad del jefe de hogar
r2r b	Tipo de educación (dummies)
ry	gasto mensual per-capita (Soles)
rbrecha	brecha de pobreza (%)
rseveri	severidad de pobreza (%)
rluz	==1, hogar con acceso a electricidad
ragua	==1, hogar con acceso a agua

La sintaxis de *dofile* en la línea 6, permitirá conocer el nombre del sistema de la laptop, la cual podrá ser usada para direccionar las carpetas con la que se van a trabajar.

La sintaxis de *dofile* en la línea 8-11, permite que los comandos en su interior no sean mostrados en la ventana de resultados de Stata. La línea 9 y 10, se utiliza el comando **global (nombre corto glo)** para poder nombrar un objeto macros de nombre **path** y **main**. El detalle de las macros son rutas de carpetas donde se van a trabajar o donde se tienen los archivos.

La sintaxis 13 y 14, detalla dos macros de nombre **clean** y **codigos** respectivamente vinculadas a dos carpetas de nombre *clean* y *codigos*.

Listing 1: *Dofile.do*

```
1
2 clear all
3 set more off
4
5 //Set pathways
6 di in yellow "`c(username)'"
7
8 qui if "`c(username)'"=="edinson" {
9     glo path "D:\Dropbox\BASES\ENAHO" // ET
10    glo main "D:\Dropbox\Docencia\UNI\L1\Aplicacion"
11 }
12
13 glo clean "${main}/Clean"
14 glo codigos "${main}/Codigos" // Codigos de dofiles variables
```

Módulo 500 (Empleo e Ingresos): la siguiente sintaxis, detalla la construcción de nuestra primera base de datos de empleo. En particular, se observa entre las líneas 10 a 14 el llamado de *dofiles*, los cuales en su interior presentan comandos para la construcción de variables adicionales. Por ejemplo, **41.- rsexo.do** detalla una sintaxis para construir una variable de nombre *rsexo* (la cual contiene información de crear una variable de mujer y hombres respectivamente).

La sintaxis en la línea 17, muestra que solo se trabajará o procesará con las observaciones de personas como jefe de hogar. La línea 18 y 19 realizará la variable logarítmica de salarios. La línea 20 (permite limpiar la información previa en la ventana de resultados de stata). Asimismo, la línea 21, permite quedarse con las variables código hogar, lnr6, r6, etc.

La línea 23, permite guardar la información a través de la ruta *clean*, donde se nombra la data procesada a través de empleo 2021 con el formato , **Empleo 2021.dta**.

Listing 2: Dofile.do

```

1  *Paso 1 :
2  *Carga de la base de datos
3  *-----
4      *Trabajando modulo de empleo
5      *-----
6      u "${path}//2021//enaho01a-2021-500.dta", clear
7
8      *Codigo de hogar
9      egen codigo.hogar=concat(conglome vivienda hogar)
10     do "${codigos}/5.- r1r.a.do"
11     do "${codigos}/9c.- r3.do"
12     do "${codigos}/15.- r6.do"
13     do "${codigos}/41.- rsexo.do"
14     do "${codigos}/8.- r2r.b.do"
15
16     *Solo jefe de hogar
17     keep if p203==1
18     g lnr6=ln(r6)
19     label var lnr6 "log-salarios (princ + secun)"
20     cls
21     keep codigo.hogar lnr6 r6 r1r.a rsexo r2r.b
22     d
23     saveold "${clean}//Empleo.2021.dta",replace

```

Modulo 100 (Vivienda): en la siguiente sintaxis se construye la base de datos de activos de la vivienda, tales como acceso a luz, agua, entre otros.

Listing 3: Dofile.do

```

1
2      *Trabajando modulo de vivienda
3      *-----
4      u "${path}//2021//enaho01-2021-100.dta", clear
5      *Codigo de hogar
6      egen codigo.hogar=concat(conglome vivienda hogar)
7      do "${codigos}/r2.-ractivos.do"
8
9      cls
10     keep codigo.hogar ragua rluz
11     d
12     saveold "${clean}//Activos.2021.dta",replace

```

La sintaxis de nombre **r2.-ractivos.do**, donde se construye las variables de acceso a electricidad por parte de las viviendas, acceso al agua. En el caso de la variable **rluz** se construye una variable dummy (discreta) para el cual se tiene el valor de 1 para los casos donde la vivienda tiene acceso a electricidad y 0 para otros casos. Sobre la imagen se puede observar como se estructura la pregunta en la ENAHO para poder extraer información de los casos (observaciones) donde la vivienda tiene acceso a electricidad (etiquetado con el valor de 1).

Listing 4: Dofile.do

```

1
2 ** Electricidad
3 gen rluz = (p1121==1)
4 replace rluz = . if p1121==.
5 lab var rluz "Acceso a electricidad"
6 lab define rluz 0 "Sin electricidad" 1 "Electricidad"
7 lab values rluz rluz

```

112. ¿CUÁL ES EL TIPO DE ALUMBRADO QUE TIENE SU HOGAR:
(Acepte una o más alternativas)

Electricidad?	1	} PASE A 113
Petróleo/gas (lámpara)?	3	
Vela?	4	
Generador?	5	
Otro?	6	
(Especifique)		
NO UTILIZA	7	

Modulo Sumaria (Medición de Pobreza): en la siguiente sintaxis se construye la base de datos de variables como la pobreza, donde se detalla **r0.- rpobre.do** que contiene las variables que se procesan para poder generar las variables mas relevantes para analizar la pobreza. Por ejemplo, como se observa en la sintaxis en detalle , se calcula la pobreza (medición unidimensional desde el ingreso) para casos como el calculo de un nuevo ponderar por hogar **rpond**, gasto mensual percapital **ry**, variable discreta de la pobreza **rpobre** (casos igual a la unidad si el hogar es pobre y 0 para otros casos).

Listing 5: Dofile.do

```

1
2 *Trabajando modulo de vivienda
3 *-----
4 u "${path}///2021//enaho01-2021-100.dta", clear
5 *Codigo de hogar
6 egen codigo.hogar=concat(conglome vivienda hogar)
7 do "${codigos}/r2.-ractivos.do"
8
9 cls
10 keep codigo.hogar ragua rluz
11 d
12 saveold "${clean}///Activos.2021.dta", replace

```

Listing 6: Dofile.do

```

1 *****
2 * Calculo de pobreza
3 * Brecha
4 * Severidad
5 *-----
6
7 g rpond=factor07*mieperho
8 tab pobreza [iw=rpond]
9 g ry=gashog2d/ mieperho/ 12
10
11 g rpobre=(pobreza<=2)
12 label var rpobre "Incidencia pobreza"
13 label define rpobre 1 "Pobre" 0 "No Pobre"
14 label values rpobre rpobre
15 *** Calculo de brecha de pobreza
16
17 gen rbrecha=(linea-ry)/linea if rpobre==1
18 replace rbrecha=0 if rpobre==0
19 label variable rbrecha "Brecha de pobreza (%)"

```

```

20
21 *** Calculo de Severidad de pobreza total
22
23     gen rseveri=rbrecha^2
24     label variable rseveri "Severidad de la pobreza"

```

Union de bases de datos: la siguiente sintaxis permitira unir las bases previamente procesadas a través del comando **merge** que permitira que se pueda unir dado la variable clave como **codigo hogar** que se repite entre cada base construida. La linea 12 y 14 permitira que se pueda borrar las bases previamente procesadas.

Listing 7: Dofile.do

```

1
2     *Union de bases de datos en una sola
3     cls
4     u "${clean}//Pobreza.2021.dta",clear
5     merge 1:1 codigo.hogar using "${clean}//Activos.2021.dta", keep(match ) nogen
6     merge 1:1 codigo.hogar using "${clean}//Empleo.2021.dta", keep(match ) nogen
7
8     order codigo.hogar rpobre r6 lnr6 rsexo rlr_a r2r_b
9     d
10    saveold "${clean}/BD.Eleccion.2021.dta",replace
11
12    erase "${clean}//Empleo.2021.dta"
13    erase "${clean}//Activos.2021.dta"
14    erase "${clean}//Pobreza.2021.dta"

```