

## COMANDOS DE STATA REVISADOS EN ESTA CLASE

### Generar, renombrar y recodificar variables:

<generate *newvarname* = *exp* [*in exp*] [*if exp*]> crea una nueva variable  
<replace *varname* = *exp* [*in exp*] [*if exp*]> reemplaza los valores de una variable existente  
<label *variable varname* "*label*"> crea una etiqueta para la variable  
<label define *labelname labelvalue labelvalue...*> define etiquetas para valores específicos  
<label values *varnamelabel name*> aplica las etiquetas de los valores a una variable específica  
<rename *oldvar newvar*> renombrar una variable  
<recode *varname rules*, generate(*newvarname*)> recodifica una variable (opción gen crea una nueva variable)

### Archivos .do:

Los archivos .do contienen un conjunto de programas que automatizan los procedimientos.  
<doedit *filename.do*> crea o edita un archivo .do (se puede ocupar cualquier procesador de texto para guardar un archivo con extensión .do)  
<do *filename.do*> ejecuta un .do file (también se puede hacer desde el menú "File", seleccionando la opción "Do")

### Unir, ordenar base de datos:

<merge 1:1 *varlist* using *filename.dta* [, options]> junta dos bases de datos  
<append using *filename.dta* [, options]> junta dos bases de datos  
<reshape> convierte el formato de la base (ancho o largo)

### Otros comandos útiles:

<tabstat> despliega tabla con estadísticas descriptivas  
<bysort> repite un comando al interior de subgrupos  
<egen> forma especial para generar nuevas variables  
<local> genera programas macro que se mantienen vigentes hasta cerrar la sesión de Stata  
<foreach> ejecuta un loop sobre distintos ítems  
<regress> regresión lineal

### Buenas prácticas:

Estructura de carpetas  
Programa donde datos originales están intocados

## DEMOSTRACIÓN PRÁCTICA

```
. *volver a abrir la base de datos original  
. use "C:\vs_chile_2005_v9.dta", clear
```

```
. *crear nueva variable grupo de edad
```

```
. gen ageg=.
(1000 missing values generated)
```

```
. replace ageg=1 if age<30
(264 real changes made)
```

```
. replace ageg=2 if age>=30 & age<60
(544 real changes made)
```

```
. replace ageg=3 if age>=60 & age<.
(192 real changes made)
```

```
. tab ageg,m
```

ageg	Freq.	Percent	Cum.
1	264	26.40	26.40
2	544	54.40	80.80
3	192	19.20	100.00
Total	1,000	100.00	

```
. *etiquetar la variable
```

```
. label var ageg "Grupo etario"
```

```
. *etiquetar los valores en dos pasos: primero crear la etiqueta y luego
adjudicársela a una o más variables
```

```
. label def ageg_1 1 "Joven" 2 "Adulto" 3 "Adulto mayor"
```

```
. label val ageg ageg_1
```

```
. des age ageg
```

variable name	storage type	display format	value label	variable label
age	byte	%10.0g		
ageg	float	%12.0g	ageg_1	Grupo etario

```
. tab ageg,m
```

Grupo etario	Freq.	Percent	Cum.
Joven	264	26.40	26.40
Adulto	544	54.40	80.80
Adulto mayor	192	19.20	100.00
Total	1,000	100.00	

```
. tab ageg, nol
```

Grupo etario	Freq.	Percent	Cum.
1	264	26.40	26.40
2	544	54.40	80.80
3	192	19.20	100.00
Total	1,000	100.00	

```
. codebook ageg
```

```
ageg
```

```
Grupo etario
```

```
-----  
-----  
-----  
type: numeric (float)  
label: ageg_1  
  
range: [1,3] units: 1  
unique values: 3 missing .: 0/1000  
  
tabulation: Freq. Numeric Label  
                  264      1 Joven  
                  544      2 Adulto  
                  192      3 Adulto mayor
```

```
. *cambiarle el nombre a una variable
```

```
. rename ageg grupoe
```

```
. *generar una variable dicotómica indicando con educación media o superior
```

```
. codebook educ
```

```
educ
```

```
RECODE of educf (RECODE of x025 (highest educational level attained))
```

```
-----  
-----  
-----  
type: numeric (byte)  
label: educ_1
```

```

                range:  [1,4]
unique values:  4
                units:  1
                missing.: 1/1000

```

```

tabulation:  Freq.  Numeric  Label
              149      1      No formal education or
              282      2      Incomplete primary school
              388      3      Less than high school
              180      4      High school
               1      .      More than high school

```

```

. gen educhm1=0

. replace educhm1=1 if educ==3 | educ==4
(568 real changes made)

. replace educhm1=. if educ==.
(1 real change made, 1 to missing)

. *otra forma de generar la misma variable

. gen educhm2=(educ==3 | educ==4) if educ<.
(1 missing value generated)

. tab educhm2,m

```

educhm2	Freq.	Percent	Cum.
0	431	43.10	43.10
1	568	56.80	99.90
.	1	0.10	100.00
-----+			
Total	1,000	100.00	

```

. replace educhm2=. if mi(educ)
(0 real changes made)

. *generar la edad al cuadrado

. gen age2=age*age

. *generar la raíz cuadrada de age2

. gen age2sqrt=sqrt(age2)

. sum age*

```

Variable	Obs	Mean	Std. Dev.	Min	Max
age	1000	42.931	16.97694	18	85
age2	1000	2130.999	1610.601	324	7225
age2sqrt	1000	42.931	16.97694	18	85

```

. *recodificar las categorías de educación

```

```
. recode educ (1/2=1) (3=2) (4=3)
(educ: 850 changes made)

. *recodificar las categorías de educación y generar una nueva variable

. recode educ (1/2=1) (3=2) (4=3), gen(educr)
(850 differences between educ and educr)

. tab educr
```

```
RECODE of |
educ |
(RECODE of |
educf |
(RECODE of |
x025 |
(highest |
educational |
level attai |
Freq. Percent Cum.
-----+-----
1 | 431 43.14 43.14
2 | 388 38.84 81.98
3 | 180 18.02 100.00
-----+-----
Total | 999 100.00
```

```
. *recodificar la edad para que tenga como techo 80+

. sum age
```

```
Variable | Obs Mean Std. Dev. Min Max
-----+-----
age | 1000 42.931 16.97694 18 85
```

```
. recode age 80/max=80
(age: 17 changes made)
```

```
. sum age
```

```
Variable | Obs Mean Std. Dev. Min Max
-----+-----
age | 1000 42.876 16.84953 18 80
```

```
. *guardar una nueva versión de la base de datos. NUNCA GUARDARLA CON EL
MISMO NOMBRE DE LA BASE ORIGINAL E IDEALMENTE GUARDARLA CON UN
NUEVO NOMBRE EN UNA CARPETA DISTINTA (EJ: DATAF VERSUS WORKF).
```

```
. cd
C:\
```

```
. save "C:\workf\vs_chile_2005_v02.dta", replace
file C:\workf\vs_chile_2005_v02.dta saved
```

. \*Es altamente recomendable guardar todos los comandos que uno ha ejecutado en un archivo .do. Esto permite replicar todo lo que uno ha hecho. Lo más importante es que permite corregir los errores fácilmente sin tener que hacer todo nuevamente desde cero. Una posibilidad es cortar y pegar los comandos que realizamos interactivamente desde la ventana "Review" a un procesador de texto simple. Otra posibilidad es escribir todo en un editor de texto simple y después ejecutarlo en Stata. Es bastante conveniente crear y administrar los archivos .do en Crimson Editor ([www.crimsoneditor.com/](http://www.crimsoneditor.com/)) o Text Pad ([www.textpad.com](http://www.textpad.com))

. doedit.do

. do clase2.do

. \*borrar todo en la memoria de Stata

. clear all

. \*salir de Stata

. exit

. \*Descargar la base de datos WDI\_class.dta ([https://dl.dropboxusercontent.com/u/9077607/WDI\\_class\\_s9.dta](https://dl.dropboxusercontent.com/u/9077607/WDI_class_s9.dta)) y VS\_class.dta ([https://dl.dropboxusercontent.com/u/9077607/VS\\_class.dta](https://dl.dropboxusercontent.com/u/9077607/VS_class.dta))  
> . WDI\_class.dta es un extracto de los World Development Indicators del Banco Mundial y VS\_class.dta es un extracto del World and European Values Surveys.

. use "C:\VS\_class.dta", clear

. \*Ambas bases de datos contienen datos de tipo panel para países. En palabras simples, para cada país existen varias observaciones a lo largo del tiempo. También tienen la misma estructura, de modo que las filas corresponden a países\*año.

. tab namea

Cum.	namea	Freq.	Percent
---	Albania	3	1.08
1.08	Algeria	1	0.36
1.44	Andorra	1	0.36
1.81	Argentina	5	1.81
3.61	Armenia	2	0.72
4.33			

```

5.42      Australia |      3      1.08
6.50      Austria  |      3      1.08
7.22      Azerbaijan|      2      0.72
7.94      Bangladesh|      2      0.72
9.03      Belarus  |      3      1.08
10.47     Belgium  |      4      1.44
11.55     Bosnia and Herzegovina |      3      1.08
12.64     Brazil   |      3      1.08
14.44     Bulgaria |      5      1.81
14.80     Burkina Faso |      1      0.36
16.25     Canada  |      4      1.44
17.69     Chile   |      4      1.44
19.13     China   |      4      1.44
20.22     Colombia |      3      1.08
21.30     Croatia |      3      1.08
22.02     Cyprus  |      2      0.72
23.10     Czech Republic |      3      1.08
24.55     Denmark |      4      1.44
24.91     Dominican Republic |      1      0.36
25.63     Egypt   |      2      0.72
25.99     El Salvador |      1      0.36
27.08     Estonia |      3      1.08
--Break--
r(1);

. sum year

      Variable |      Obs      Mean      Std. Dev.      Min      Max
-----+-----
      year |      277    1999.134    7.740365    1981    2009

. browse

```

. \*Con el comando isid podemos ver si existe un identificador único para cada fila en la base de datos

```
. isid namea
variable namea does not uniquely identify the observations
r(459);
```

```
. isid namea year
```

```
. isid idwb year
```

. \*Sin embargo, en las encuestas de valores hay menos países y los años no son exactamente los mismos.

```
. use "C:\WDI_class_s9.dta", clear
```

```
. tab countryname
```

Cum.	Country name	Freq.	Percent
---			
1.05	Albania	11	1.05
2.11	Algeria	11	1.05
3.16	Andorra	11	1.05
4.21	Argentina	11	1.05
5.26	Armenia	11	1.05
6.32	Australia	11	1.05
7.37	Austria	11	1.05
8.42	Azerbaijan	11	1.05
9.47	Bangladesh	11	1.05
10.53	Belarus	11	1.05
11.58	Belgium	11	1.05
12.63	Bosnia and Herzegovina	11	1.05
13.68	Brazil	11	1.05
14.74	Bulgaria	11	1.05
15.79	Burkina Faso	11	1.05
16.84	Canada	11	1.05



```

17.89          Chile |          11          1.05
18.95          China |          11          1.05
20.00    Colombia |          11          1.05
21.05          Croatia |          11          1.05
22.11          Cyprus |          11          1.05
23.16    Czech Republic |          11          1.05
24.21          Denmark |          11          1.05
25.26    Dominican Republic |          11          1.05
26.32    Egypt, Arab Rep. |          11          1.05
27.37    El Salvador |          11          1.05
28.42          Estonia |          11          1.05

```

```
--Break--
r(1);
```

```
. sum year
```

Variable	Obs	Mean	Std. Dev.	Min	Max
year	1045	2005	3.163792	2000	2010

```
. browse
```

```
. isid countryname year
```

```
. isid idwb year
```

```
. *Como la estructura de la base es la misma, podemos juntrlas de inmediato. Esto no sería posible si tuvieran una estructura distinta. Por ejemplo, podría ser que en la VS cada fila corresponde a un país, de modo que tendríamos que cambiar el formato de la base de datos WDI ocupando el comando reshape.
```

```
. help reshape
```

```
. *El comando reshape requiere especificar tanto el número identificador de la unidad de análisis de mayor nivel en la base (i) como aquella que distingue a las observaciones repetidas a su interior (j). En este caso i son países y j son años, pero podrían ser cualquier otro tipo de unidades ordenadas de forma jerárquica: personas y años, países y personas, universidades y estudiantes.
```

```
. reshape wide adfer femlg gdpcoi hxpcoi hxpugd hxpugv hxputo hxtogd life toilet water mortl povp fseat expugd expugv prur unelf imfcud, i(cid)
```

```
j(year)
```

```
(note: j = 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010)
```

```

Data                                long  ->  wide
-----
----
Number of obs.                      1045  ->    95
Number of variables                   31   ->   220
j variable (11 values)               year  -> (dropped)
xij variables:
    adfer                            adfer -> adfer2000 adfer2001 ...
adfer2010
    femlg                             femlg -> femlg2000 femlg2001 ...
femlg2010
    gdpcoi                            gdpcoi -> gdpcoi2000 gdpcoi2001 ...
gdpcoi2010
    hxpcoi                            hxpcoi -> hxpcoi2000 hxpcoi2001 ...
hxpcoi2010
    hxpugd                             hxpugd -> hxpugd2000 hxpugd2001 ...
hxpugd2010
    hxpugv                             hxpugv -> hxpugv2000 hxpugv2001 ...
hxpugv2010
    hxputo                             hxputo -> hxputo2000 hxputo2001 ...
hxputo2010
    hxtogd                             hxtogd -> hxtogd2000 hxtogd2001 ...
hxtogd2010
    life                               life -> life2000 life2001 ...
life2010
    toilet                             toilet -> toilet2000 toilet2001 ...
toilet2010
    water                              water -> water2000 water2001 ...
water2010
    mortl                             mortl -> mortl2000 mortl2001 ...
mortl2010
    povp                               povp -> povp2000 povp2001 ...
povp2010
    fseat                              fseat -> fseat2000 fseat2001 ...
fseat2010
    expugd                             expugd -> expugd2000 expugd2001 ...
expugd2010
    expugv                             expugv -> expugv2000 expugv2001 ...
expugv2010
    prur                               prur -> prur2000 prur2001 ...
prur2010
    unelf                              unelf -> unelf2000 unelf2001 ...
unelf2010
    imfcud                             imfcud -> imfcud2000 imfcud2001 ...
imfcud2010
-----

```

----

. browse

. \*Esta es solamente una forma distinta de organizar la base de datos, ya que contiene la misma información que antes. De hecho es posible volver a la forma original.

```
. reshape long adfer expugd expugv femlg fseat gdpcoi hxpcoi hxpugd
hxpugv hxputo hxtogd imfcud life morti povp prur toilet unelf water,
i(cid) j(year)
(note: j = 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010)
```

```
Data                                wide  ->  long
-----
----
Number of obs.                      95   ->  1045
Number of variables                  220  ->   31
j variable (11 values)              ->  year
xij variables:
    adfer2000 adfer2001 ... adfer2010 ->  adfer
    expugd2000 expugd2001 ... expugd2010 ->  expugd
    expugv2000 expugv2001 ... expugv2010 ->  expugv
    femlg2000 femlg2001 ... femlg2010   ->  femlg
    fseat2000 fseat2001 ... fseat2010   ->  fseat
    gdpcoi2000 gdpcoi2001 ... gdpcoi2010 ->  gdpcoi
    hxpcoi2000 hxpcoi2001 ... hxpcoi2010 ->  hxpcoi
    hxpugd2000 hxpugd2001 ... hxpugd2010 ->  hxpugd
    hxpugv2000 hxpugv2001 ... hxpugv2010 ->  hxpugv
    hxputo2000 hxputo2001 ... hxputo2010 ->  hxputo
    hxtogd2000 hxtogd2001 ... hxtogd2010 ->  hxtogd
    imfcud2000 imfcud2001 ... imfcud2010 ->  imfcud
    life2000 life2001 ... life2010      ->  life
    morti2000 morti2001 ... morti2010   ->  morti
    povp2000 povp2001 ... povp2010     ->  povp
    prur2000 prur2001 ... prur2010     ->  prur
    toilet2000 toilet2001 ... toilet2010 ->  toilet
    unelf2000 unelf2001 ... unelf2010   ->  unelf
    water2000 water2001 ... water2010   ->  water
-----
----
```

```
. order cid idwb countryname year
```

. \*Para juntar dos bases de datos lo más fácil es cuando tienen la misma estructura y cantidad de observaciones (igual cantidad de países y de años), de modo tal que el match sea perfecto. Lamentablemente esto rara vez sucede y es necesario cambiar la estructura de las bases de datos o bien hacer un match imperfecto. En cualquier caso siempre necesitamos un identificador único de las filas que sea común a ambas bases de datos y nos permita juntarlas.

```
. merge 1:1 idwb year using "C:\VS_class.dta"
```

```
Result                                # of obs.
-----
not matched                            1,050
    from master                        909  (_merge==1)
    from using                          141  (_merge==2)

matched                                136  (_merge==3)
-----
```

. \*Es importante elegir si queremos conservar todas las observaciones o solamente un grupo de ellas. En este caso solamente quiero agregar variables a los datos WDI.

. drop if \_merge==2  
(141 observations deleted)

. \*Cambiar la forma de la base de datos puede ser útil no solamente para juntar dos bases de datos, sino también para realizar algunos procedimientos específicos. Por ejemplo podríamos querer calcular el GDP per cápita promedio de cada país durante el periodo observado.

. sum year gdpcoi

Variable	Obs	Mean	Std. Dev.	Min	Max
year	1045	2005	3.163792	2000	2010
gdpcoi	1007	15.56187	13.66008	.5166825	74.02146

. bysort countryname : sum gdpcoi

-> countryname = Albania

Variable	Obs	Mean	Std. Dev.	Min	Max
gdpcoi	11	6.075775	1.149676	4.461047	7.791755

-> countryname = Algeria

Variable	Obs	Mean	Std. Dev.	Min	Max
gdpcoi	11	6.685025	.5124311	5.854095	7.259959

-> countryname = Andorra

Variable	Obs	Mean	Std. Dev.	Min	Max
gdpcoi	0				

-> countryname = Argentina

Variable	Obs	Mean	Std. Dev.	Min	Max
----------	-----	------	-----------	-----	-----

```
-----+-----
      gdpcoi |          7      10.05939      1.007383      8.596048      11.65782
-----+-----
```

```
--Break--
-> countryname = Armenia
r(1);
```

. \*El comando egen permite generar variables utilizando funciones o argumentos más complejos.

```
. bysort countryname : egen gdpupro=mean(gdpcoi)
(33 missing values generated)
```

```
. list idwb gdpcoi gdpupro, sepby(idwb)
```

```
-----+-----
      | idwb      gdpcoi      gdpupro |
-----+-----
  1. | ALB      4.4610466      6.075775 |
  2. | ALB      4.8007214      6.075775 |
  3. | ALB      7.2728438      6.075775 |
  4. | ALB       6.732368      6.075775 |
  5. | ALB      5.9979041      6.075775 |
  6. | ALB      4.9739813      6.075775 |
  7. | ALB      7.5257768      6.075775 |
  8. | ALB      5.2967925      6.075775 |
  9. | ALB      7.7917555      6.075775 |
 10. | ALB      6.3305938      6.075775 |
 11. | ALB      5.649745      6.075775 |
-----+-----
 12. | DZA      5.9258286      6.685025 |
 13. | DZA      5.854095      6.685025 |
 14. | DZA      6.4609728      6.685025 |
 15. | DZA      7.1385629      6.685025 |
 16. | DZA      6.7039302      6.685025 |
 17. | DZA      6.9689262      6.685025 |
 18. | DZA      7.0996367      6.685025 |
 19. | DZA      7.0573631      6.685025 |
 20. | DZA      7.2599595      6.685025 |
 21. | DZA      6.1238139      6.685025 |
 22. | DZA      6.9421879      6.685025 |
-----+-----
 23. | ADO              .              . |
 24. | ADO              .              . |
-----+-----
```

```
--Break--
r(1);
```

. \*Con esta nueva variable podemos clasificar a los países como de ingresos bajos, medios o altos

```
. sum gdppro
```

Variable	Obs	Mean	Std. Dev.	Min	Max
gdppro	1012	15.53367	13.53721	.6562819	67.11758

```
. recode gdppro min/9.9=1 10/19.9=2 20/max=3, gen(ning)
(1012 differences between gdppro and ning)
```

```
. label var ning "Ingreso p/c promedio"
```

```
. label def ning_1 1 "Bajo" 2 "Medio" 3 "Alto"
```

```
. label val ning ning_1
```

```
. tab ning,m
```

Ingreso p/c promedio	Freq.	Percent	Cum.
Bajo	484	46.32	46.32
Medio	165	15.79	62.11
Alto	363	34.74	96.84
.	33	3.16	100.00
Total	1,045	100.00	

```
. *Siempre es conveniente chequear el resultado de la recodificación
```

```
. tabstat gdppro, by(ning) s(min max mean n)
```

```
Summary for variables: gdppro
by categories of: ning (Ingreso p/c promedio)
```

ning	min	max	mean	N
Bajo	.6562819	9.713863	4.598043	484
Medio	10.04001	16.57089	12.8887	165
Alto	20.09322	67.11758	31.31676	363
Total	.6562819	67.11758	15.53367	1012

```
. tab countryname if ning==3
```

Cum.	Country name	Freq.	Percent
3.03	Australia	11	3.03
6.06	Austria	11	3.03
9.09	Belgium	11	3.03

---

12.12	Canada	11	3.03
15.15	Cyprus	11	3.03
18.18	Czech Republic	11	3.03
21.21	Denmark	11	3.03
24.24	Finland	11	3.03
27.27	France	11	3.03
30.30	Germany	11	3.03
33.33	Greece	11	3.03
36.36	Hong Kong SAR, China	11	3.03
39.39	Iceland	11	3.03
42.42	Ireland	11	3.03
45.45	Israel	11	3.03
48.48	Italy	11	3.03
51.52	Japan	11	3.03
54.55	Korea, Rep.	11	3.03
57.58	Luxembourg	11	3.03
60.61	Malta	11	3.03
63.64	Netherlands	11	3.03
66.67	New Zealand	11	3.03
69.70	Norway	11	3.03
72.73	Portugal	11	3.03
75.76	Saudi Arabia	11	3.03
78.79	Singapore	11	3.03
81.82	Slovenia	11	3.03
84.85	Spain	11	3.03
87.88	Sweden	11	3.03
90.91	Switzerland	11	3.03

93.94	Trinidad and Tobago	11	3.03
96.97	United Kingdom	11	3.03
100.00	United States	11	3.03
-----+			
---			
	Total	363	100.00

. \*Para hacer operaciones repetitivas existen comandos muy que ahorran mucho tiempo

```
. foreach X in expugd expugv hxpugd hxpugv {
  2.     gen `X'2 = `X'/100
  3.     }
(316 missing values generated)
(397 missing values generated)
(31 missing values generated)
(31 missing values generated)
```

```
. sum expugd* expugv* hxpugd* hxpugv*, sep(2)
```

Variable	Obs	Mean	Std. Dev.	Min	Max
expugd	729	4.698092	1.452675	1.34536	9.50976
expugd2	729	.0469809	.0145267	.0134536	.0950976
-----+					
expugv	648	14.84989	4.555848	6.20165	30.96989
expugv2	648	.1484989	.0455585	.0620165	.3096989
-----+					
hxpugd	1014	4.247176	2.144949	.0317866	10.26867
hxpugd2	1014	.0424718	.0214495	.0003179	.1026867
-----+					
hxpugv	1014	12.19458	4.234022	0	23.30972
hxpugv2	1014	.1219458	.0423402	0	.2330972

. \*Otro ejemplo de procedimiento repetitivo

```
. local i = 2000
```

```
. while `i'<=2010 {
  2.     egen life`i' = mean(life) if year==`i'
  3.     local ++i
  4.     }
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
(950 missing values generated)
```



```
. sum life????, sep(0)
```

Variable	Obs	Mean	Std. Dev.	Min	Max
life2000	95	70.5359	0	70.5359	70.5359
life2001	95	70.83074	0	70.83074	70.83074
life2002	95	71.02898	0	71.02898	71.02898
life2003	95	71.27489	0	71.27489	71.27489
life2004	95	71.6395	0	71.6395	71.6395
life2005	95	71.88087	0	71.88087	71.88087
life2006	95	72.24376	0	72.24376	72.24376
life2007	95	72.53052	0	72.53052	72.53052
life2008	95	72.82108	0	72.82108	72.82108
life2009	95	73.17147	0	73.17147	73.17147
life2010	95	73.49966	0	73.49966	73.49966

```
. *Y lo último, explorar relaciones multivariadas
```

```
. sum year
```

Variable	Obs	Mean	Std. Dev.	Min	Max
year	1045	2005	3.163792	2000	2010

```
. keep if year==2010  
(950 observations deleted)
```

```
. regress life povp
```

Source	SS	df	MS	Number of obs =
Model	841.079574	1	841.079574	F( 1, 24) =
Residual	364.266859	24	15.1777858	Prob > F =
Total	1205.34643	25	48.2138573	R-squared =
				Adj R-squared =
				Root MSE =

life	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
povp	-.1914512	.0257184	-7.44	0.000	-.2445313 - .138371
_cons	74.91684	.9624947	77.84	0.000	72.93035 76.90334

```
. regress life povp water toilet morti
```

Source	SS	df	MS	Number of obs =
Model	1025.21527	4	256.303817	F( 4, 19) =
Residual	136.313463	19	7.17439278	Prob > F =
Total	1161.52873	23	50.5012491	R-squared =

0.0000  
0.8826  
0.8579  
2.6785

Adj R-squared =  
Root MSE =

life	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
povp	.0211769	.0506493	0.42	0.681	-.0848333 .1271871
water	.0904133	.1234148	0.73	0.473	-.1678969 .3487234
toilet	-.0222986	.0678281	-0.33	0.746	-.1642643 .1196672
morti	-.2985206	.0913709	-3.27	0.004	-.489762 .1072792
_cons	70.60739	13.56075	5.21	0.000	42.22441 98.99036

### TAREA PARA LA SIGUIENTE CLASE

- Replicar todo lo que hemos hecho en clases.
- Pegar las secciones relevantes del .log file (eliminando errores)
- Enviarle al ayudante al menos una hora antes de la siguiente clase.