Introducción a Stata II

Programación para el análisis de datos

Departamento de Ciencias Sociales, UCU - Martín Opertti

Limpiar datos

Sort

- A veces queremos ordenar los datos según una variable, para eso utilizamos el comando `sort` que ordena las observaciones en orden ascendente (de menor a mayor).
- Los casos perdidos son tratados como infinitos por lo que quedan al final.
- La sintáxis es `sort varlist`
- El argumentos `stable` sirve para mantener el orden anterior dentro de cada subgrupo nuevo
- Con `gsort` podemos ordenar de forma descendente

```
sort country_name

sort region, stable

sort region country_name

gsort - country_name
```

Orden de variables

- El comando `order` sirve para cambiar el orden las columnas en la base de datos.
- La sintaxis es `order varlist [, options]`. Algunas de las opciones o argumentos son `first`, `last`, `before(varname)`, `after(varname)`.

```
order country_name region, first

order country_code, after(country_name)
```

Filtrar observaciones

Para filtrar observaciones podemos utilizar dos comandos: `drop` y `keep`. Estos comandos van seguidos de un condicional, `drop if *exp*`.

```
keep if edad == 20

drop if edad > 80
```

`drop` y `keep` también nos sirven para seleccionar o quitar variables (columnas). Su uso para este caso es: `drop *varlist*`

```
keep edad educacion

drop ingreso genero
```

Operadores y expresiones

Operadores aritméticos	Operadores relacionales	Operadores lógicos
+ sumar	& y	== igual a
- restar	| o	!= no igual a
* multplicar	! no	< menor a
/ dividir		> mayor a
^ elevar a una potencia		<= menor o igual a
		>= mayor o igual a

Operadores y expresiones

```
keep if tasa_desempleo > 5 & tasa_desempleo <= 11

keep if country_name == "Uruguay" | country_name == "Argentina"

keep if region != "North America"

drop if tasa_desempleo <= 10
```

Ejercicio

Filtrar la base para quedarse solamente con las observaciones que corresponden a países de América del Norte y Uruguay

Variables

Crear nueva variable (expresiones genéricas)

Para crear una nueva variable en Stata se puede utilizar el comando `generate` o su abreviación `gen`.

`generate` crea una nueva variable a partir de una expresión, y tiene la siguiente sintáxis: `generate newvar = exp`.

```
gen poblacion_total_m = pob_total/1000000

gen area_no_selvatica = area_total - area_selvatica

gen area_selvatica_per = (area_selvatica * 100) / area_total

gen co2 = co2_pc * pob_total
```

Funciones

Función	Definición
abs(x)	valor absoluto de x
exp(x)	función exponencial de x
log(x)	logaritmo natural de x
round(x)	redondea al entero más cercano de x
sqrt(x)	raíz cuadrada de x

Ejercicio

1. Crear una nueva variable de `pib_pc` *llamada* `pib_pc_2` *a partir de otras variables en la base. Chequear que coincidan los valores de* `pib_pc` *y* `pib_pc_2`

2. Redondear al número entero más cercano la variable co2_pc

Crear nueva variable (expresiones específicas)

Para crear una variable a partir de expresiones específicas sobre valores de otra variable (ej. una variable que indique si el país es miembro del Mercosur o no), utilizamos nuevamente `gen` pero en conjunto con `replace`.

De esta forma, la creación de la nueva variable se divide en al menos dos líneas. `replace` cambia el contenido de una variable ya existente (que en estos casos generamos previamente utilizando `gen`). La sintáxis abreviada de `replace` es: `replace varname = exp`.

```
gen mercosur = 0 // Primero creo variable con el valor 0 para todos
replace mercosur = 1 if country_name == "Argentina"
replace mercosur = 1 if country_name == "Brasil"
replace mercosur = 1 if country_name == "Paraguay"
replace mercosur = 1 if country_name == "Uruguay"
```

Crear nueva variable (expresiones específicas)

```
gen pob_m3 = 0
replace pob_m3 = 1 if pob_total >= 3000000
replace pob_m3 = . if missing(pob_total)


gen td_rec = 0
replace td_rec =  1 if tasa_desempleo < 10
replace td_rec =  2 if tasa_desempleo >= 10 & tasa_desempleo < 15
replace td_rec =  3 if tasa_desempleo >= 15
replace td_rec = . if missing(tasa_desempleo)

gen crec_ele = 0
replace crec_ele = 1 if pob_crecimiento > 2 & acceso_electricidad < 90
replace crec_ele = . if missing(pob_crecimiento) | missing(acceso_electricidad)
```

Ejercicio

Crear una variable que:

- tome el valor 2 si el gasto en educación y en salud es mayor a 6%

- tome el valor 1 si el gasto en una de estas dos variables es mayor al 6% pero no en la otra

- tome el valor 0 si el gasto en ambas variables es menor a 6%

- sea missing si tenemos valores perdidos en alguna de estas dos variables.

Crear nueva variable (expresiones específicas)

En recodificaciones complejas tener cuidado con el **orden**. Estas líneas de código producen resultados muy diferentes! Vamos de general a particular, es decir, si tenemos dos condiciones y una incluye a la otra, debemos ir por la más general primero (tener más de 6% de gasto en educación o salud) y luego por la particular (tener más de 6% de gasto en ambas).

Para nuestro ejercicio este código es el correcto:

```
gen var_ej = 0
replace var_ej = 1 if gasto_salud > 6 | gasto_educacion > 6 # general
replace var_ej = 2 if gasto_salud > 6 & gasto_educacion > 6 # particular
replace var_ej = . if missing(gasto_salud) | missing(gasto_educacion)
```

Este es incorrecto:

```
gen var_ej = 0
replace var_ej = 2 if gasto_salud > 6 & gasto_educacion > 6 # particular
replace var_ej = 1 if gasto_salud > 6 | gasto_educacion > 6 # general
replace var_ej = . if missing(gasto_salud) | missing(gasto_educacion)
```

Crear nueva variable con egen

- `egen` sirve para crear nuevas variables con funciones propias.
- Algunas de las funciones de egen son:
  - `rowmean` crea la media de las variables especificadas
  - `cut` recodifica la variable con los intervalos especificados con `at`
  - `rank` ranking de valores
- La sintaxis de `egen` es: `egen newvar = funcion(argumentos)`

```
egen promedio_gasto = rowmean(gasto_salud gasto_educacion gasto_militar)
egen td_rec_2 = cut(tasa_desempleo), at(0, 10, 15) label
egen indice_de_gini_rec = rank(indice_de_gini)
```

Ejercicio

Explicar las operaciones realizadas en las siguientes dos líneas de código

```
gen country_rec = word(country_name, 1)
replace country_rec = usubinstr(country_rec , ",", "", .)
```

Ejercicio

*Encontrar y utilizar una función del comando egen que nos indique el total de un grupo de variables para cada observación. Aplicar obteniendo el total de las variables que comienzan con gasto_*

Recodificar variables

- Con `recode` es posible recodificar los valores de una variable numérica.
- La sintaxis es: `recode varlist (rule)`
- Con `/` podemos establecer intervalos, ej `0/5 = 1` implica que todos los valores de 0 a 5 corresponderán a 1.

```
recode region (5 = 3) (6 = 1)
recode region (5 = 3) (6 = 1), gen(region_rec)

recode tasa_desempleo  (0/5 = 1) (5/10 = 2) (10/15 = 3) (15/max = 4), gen(td_rec)
recode tasa_desempleo (0/5 = 1 Baja) (5/10 = 2 Media) (10/15 = 3 Alta) (15/max = 4 Muy_alta), gen(td_rec_2)
```

Gráficos

Gráficos

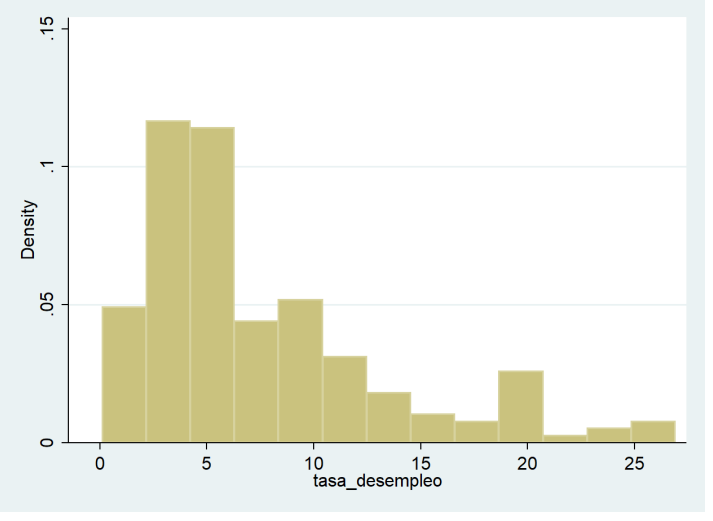
- Stata tiene comandos para realizar distintas visualizaciones de datos.
- Stata cheat sheet de visualizaciones
- El comando `graph` es la base de las visualizaciones en Stata. Alguno de los tipos que permite son:

Comando	Visualización
graph twoway	Gráficos de dispersión, de línea, etc
graph bar	Gráficos de barras
graph box	Gráficos de cajas
graph pie	Gráficos de tortas

- El comando `hist` sirve para crear histogramas, muy útiles para explorar una base de datos.

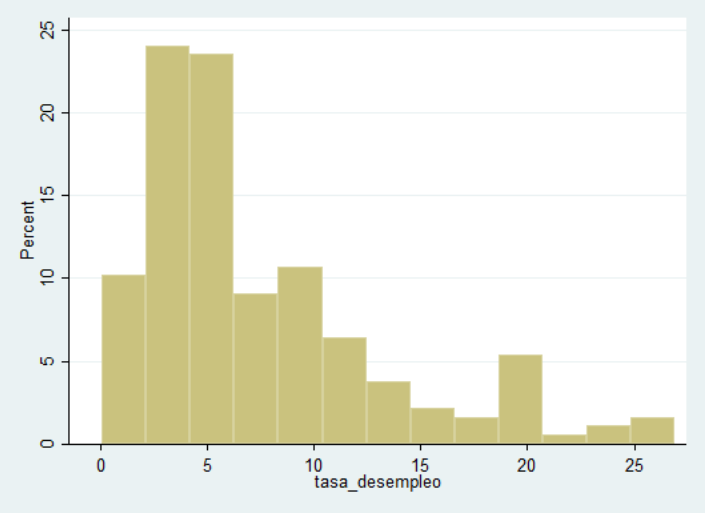
Histogramas

```
# Histograma simple
hist tasa_desempleo
```



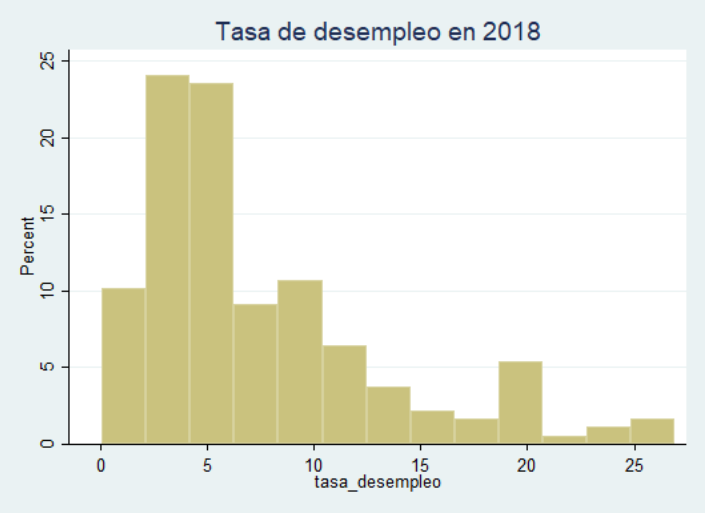
Histogramas

```
# Graficar porcentajes en lugar de proporción
hist tasa_desempleo, percent
```



Histogramas

```
# Agregar título
hist tasa_desempleo, percent title("Tasa de desempleo en 2018")
```



Histogramas

```
hist tasa_desempleo, percent by(region)
```

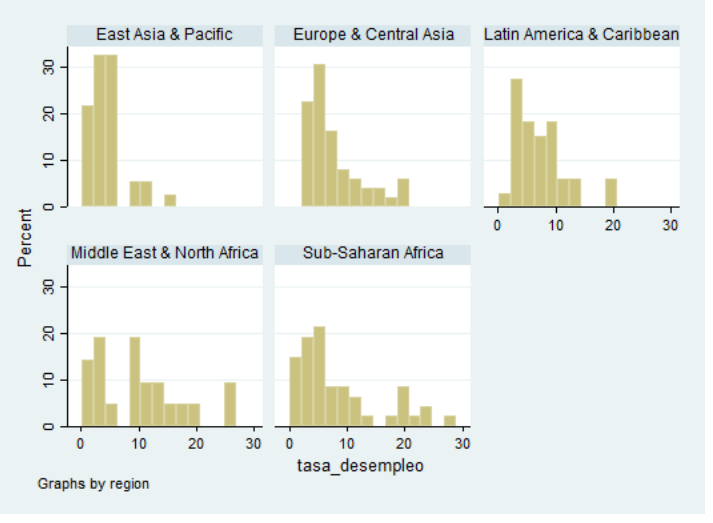


Gráfico de dispersión

Gráfico de dispersión simple:

```
graph twoway scatter pib_pc co2_pc
```

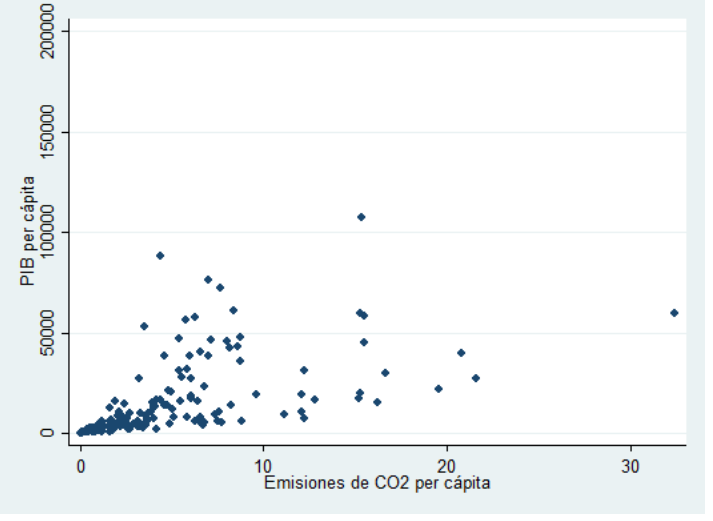


Gráfico de dispersión

No es necesario incluir `graph twoway`

```
scatter pib_pc co2_pc
```

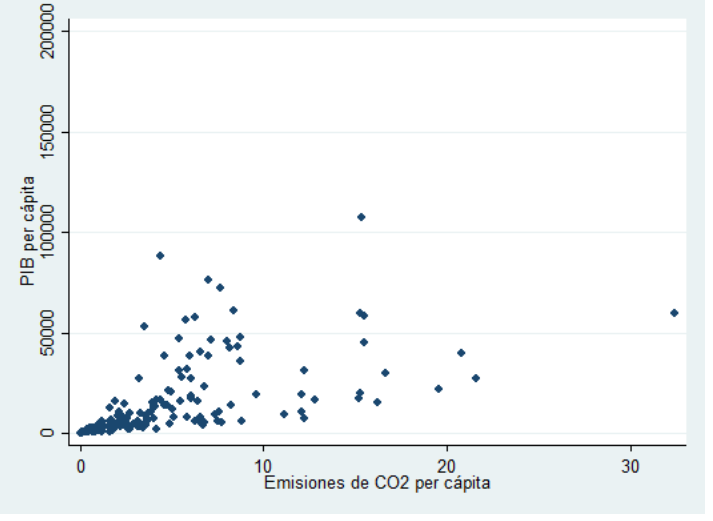


Gráfico de dispersión

Agregar etiquetas de variables para los ejes y título:

```
label variable pib_pc "PIB per cápita"
label variable co2_pc "Emisiones de CO2 per cápita"
scatter pib_pc co2_pc, ///
 title("Relación entre PIB per cápita y CO2 per cápita")
```

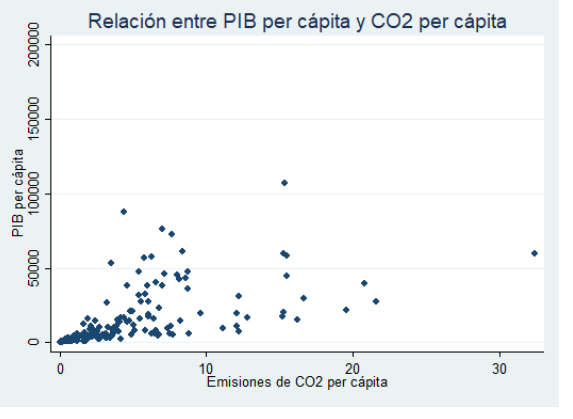


Gráfico de dispersión

Agregar etiqueta para puntos:

```
scatter pib_pc co2_pc, ///
  title("Relación entre PIB per cápita y CO2 per cápita") ///
 mlabel(country_code)
```

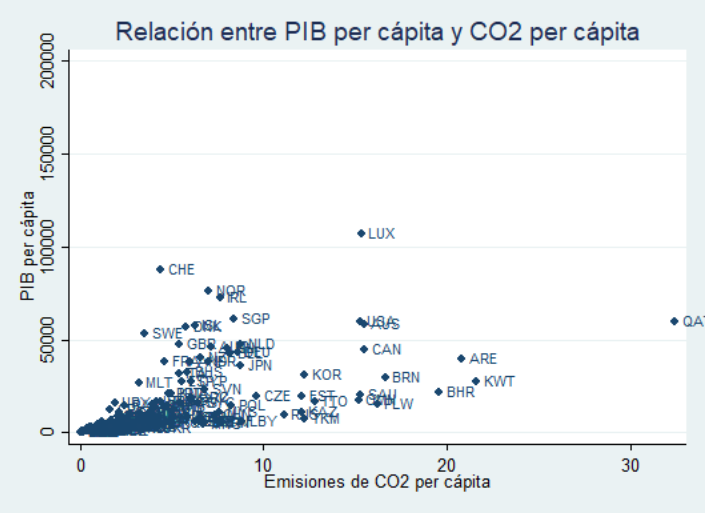


Gráfico de barras

Media de variable númerica según categoría de otra variable:

```
graph bar pib_pc, over(region) ///
 title("Media de PIB per cápita según región")
```

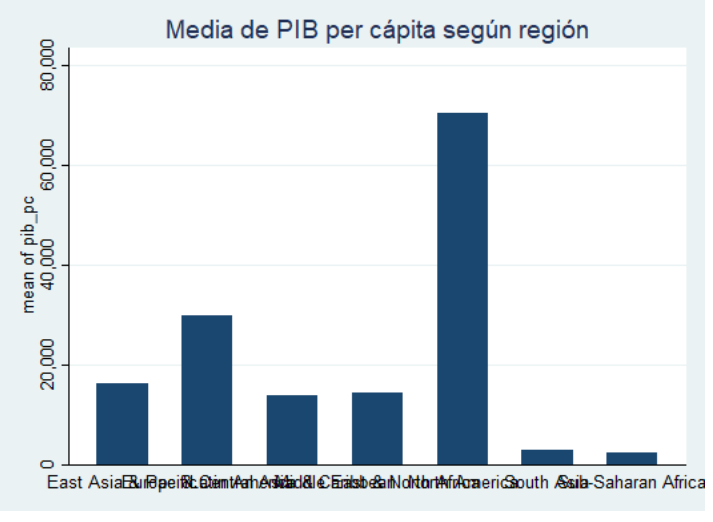


Gráfico de barras

Barras horizontales para mejorar la visualización:

```
graph hbar pib_pc, over(region) ///
 title("Media de PIB per cápita según región")
```

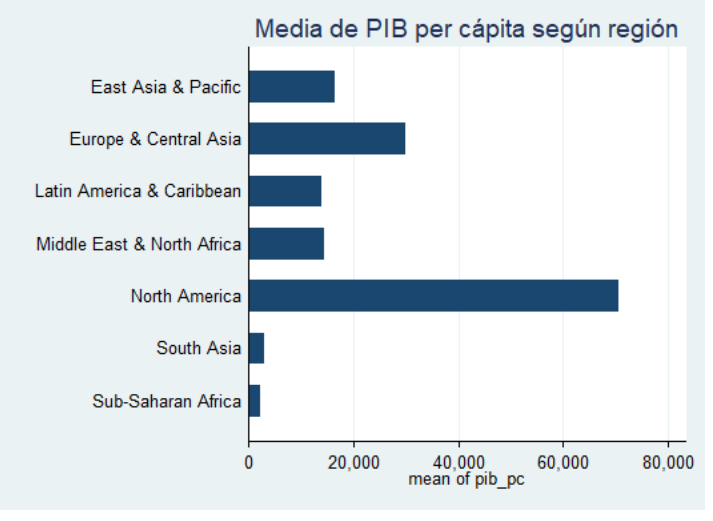
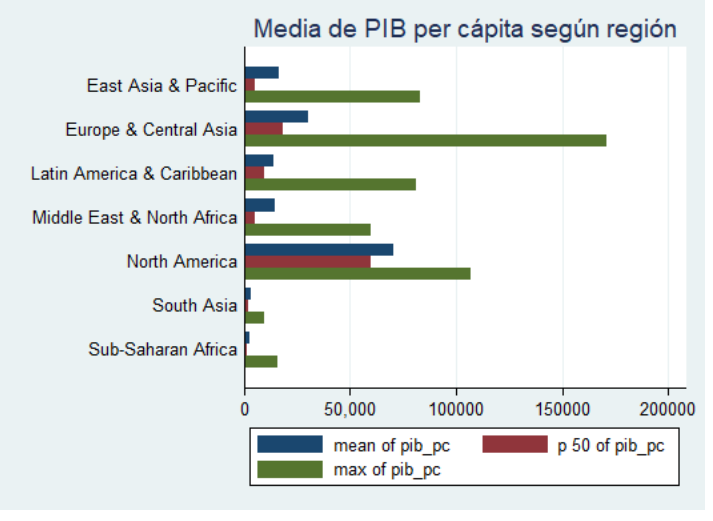


Gráfico de barras

Agregar medias, medianas y máxima:

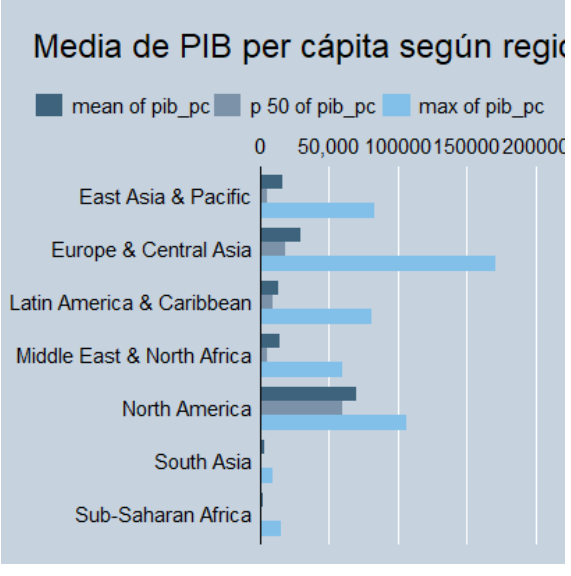
```
graph hbar (mean) pib_pc (median) pib_pc (max) pib_pc, over(region) ///
 title("Media de PIB per cápita según región")
```



Temas

- Con `scheme` (dentro de las opciones de `graph`) o `set scheme` por cada sesión es posible establecer temas.

```
graph hbar (mean) pib_pc (median) pib_pc (max) pib_pc, over(region) ///
 title("Media de PIB per cápita según región") scheme(economist)
```



Guardar gráficos

Para guardar un gráfico generado en Stata se puede utilizar tanto `graph save` como `graph export`.

La sintaxis es: `graph export newfilename.suffix`.

Ver otros formatos con `help graph export`

```
# Exportar como gph (formato Stata)
graph save ejemplo

# Exportar como png
graph export ejemplo.png

# Exportar como pdf
graph export ejemplo.pdf
```

Ejercicio

Graficar la relación entre gasto en salud y gasto en educación de la forma que te parezca más conveniente y con la mayor cantidad de información posible

Estadística inferencial

t-test

- Las pruebas t (t-test) independientes están diseñadas para comparar medias de la misma variable entre dos grupos.
- Recodifiquemos la variable de ingreso para que tenga dos categorías, bajos y altos.

```
recode income_group (1/2=1 Bajos) (3/4=2 Altos), gen(income_group_2)

ttest co2_pc, by(income_group_2)
```

```
Two-sample t test with equal variances

   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   Bajos |      81    1.040939    .1471727    1.324555    .748056    1.333822
   Altos |     109    6.523266    .4957057     5.17532    5.540691    7.505841
---------+--------------------------------------------------------------------
combined |     190    4.186064     .351219     4.84122    3.493251    4.878876
---------+--------------------------------------------------------------------
    diff |           -5.482327    .5892231               -6.644666   -4.319989
------------------------------------------------------------------------------
    diff = mean(Bajos) - mean(Altos)                              t =  -9.3043
Ho: diff = 0                                     degrees of freedom =      188

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 1.0000

.
end of do-file
```

Modelos de regresión

- Con el comando `regress` o `reg` se pueden estimar regresiones lineales.
- La sintaxis del comando es: `regress depvar [indepvars] [if] [in] [weight] [, options]`
- Las variables categóricas

```
reg co2_pc pib_pc i.region
```

```
. reg co2_pc pib_pc i.region
```

Source	SS	df	MS	Number of obs	=	186
				F(7, 178)	=	26.06
Model	2218.99941	7	316.999916	Prob > F	=	0.0000
Residual	2165.44352	178	12.165413	R-squared	=	0.5061
				Adj R-squared	=	0.4867
Total	4384.44293	185	23.6996915	Root MSE	=	3.4879

co2_pc	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
pib_pc	.000132	.0000165	7.99	0.000	.0000994	.0001646
region						
Europe & Central Asia	-.594886	.8458044	-0.70	0.483	-2.26398	1.074208
Latin America & Caribbean	-1.5898	.8954389	-1.78	0.078	-3.356842	.1772419
Middle East & North Africa	3.611685	1.014624	3.56	0.000	1.609446	5.613923
North America	5.244553	2.633704	1.99	0.048	.0472516	10.44186
South Asia	-2.293607	1.401805	-1.64	0.104	-5.059901	.4726876
Sub-Saharan Africa	-2.537074	.8438911	-3.01	0.003	-4.202392	-.871755
_cons	3.211574	.6797802	4.72	0.000	1.870109	4.553039