

# Taller de reforzamiento y nivelación de Econometría y aplicaciones con Stata

## Sesión 1: Introducción a Stata

César Mora Ruiz

QLAB - PUCP

Setiembre de 2024

# Temas a abordar

El propósito de esta sesión es brindar las herramientas necesarias para llevar a cabo análisis de datos:

- Presentación de STATA como software para análisis de datos e investigación
- Navegando entre las funciones
- El uso de Do-files
- Importación de datos en formatos dta, excel y csv
- Exploración de datos
- Visualización de datos
- Gestión de datos

# ¿Qué es STATA?

- Es un paquete muy amigable, pero poderoso, para realizar análisis de datos con fuerte potencial para:
  - Análisis estadístico
  - Manipulación y gestión de datos
  - Visualización de datos al detalle
- STATA nos ofrece muchas herramientas que implementan métodos analíticos y econométricos de uso estándar, así como métodos nuevos y avanzados que se van incorporando como parte de los nuevos lanzamientos del software.

# STATA - Ventajas

- La sintaxis de los comandos es muy sencilla, corta e intuitiva
- Dicha sintaxis es muy consistente cuando se utilizan diversos comandos al mismo tiempo, por lo que es más sencilla de comprender y aprender
- Un paquete muy competitivo ya que cuenta con diversidad de métodos
- Amplia documentación existente a nivel de libros y en la web
- Cuenta con componentes ideales para realizar análisis estadístico, econométrico y de encuestas de diseño complejo.

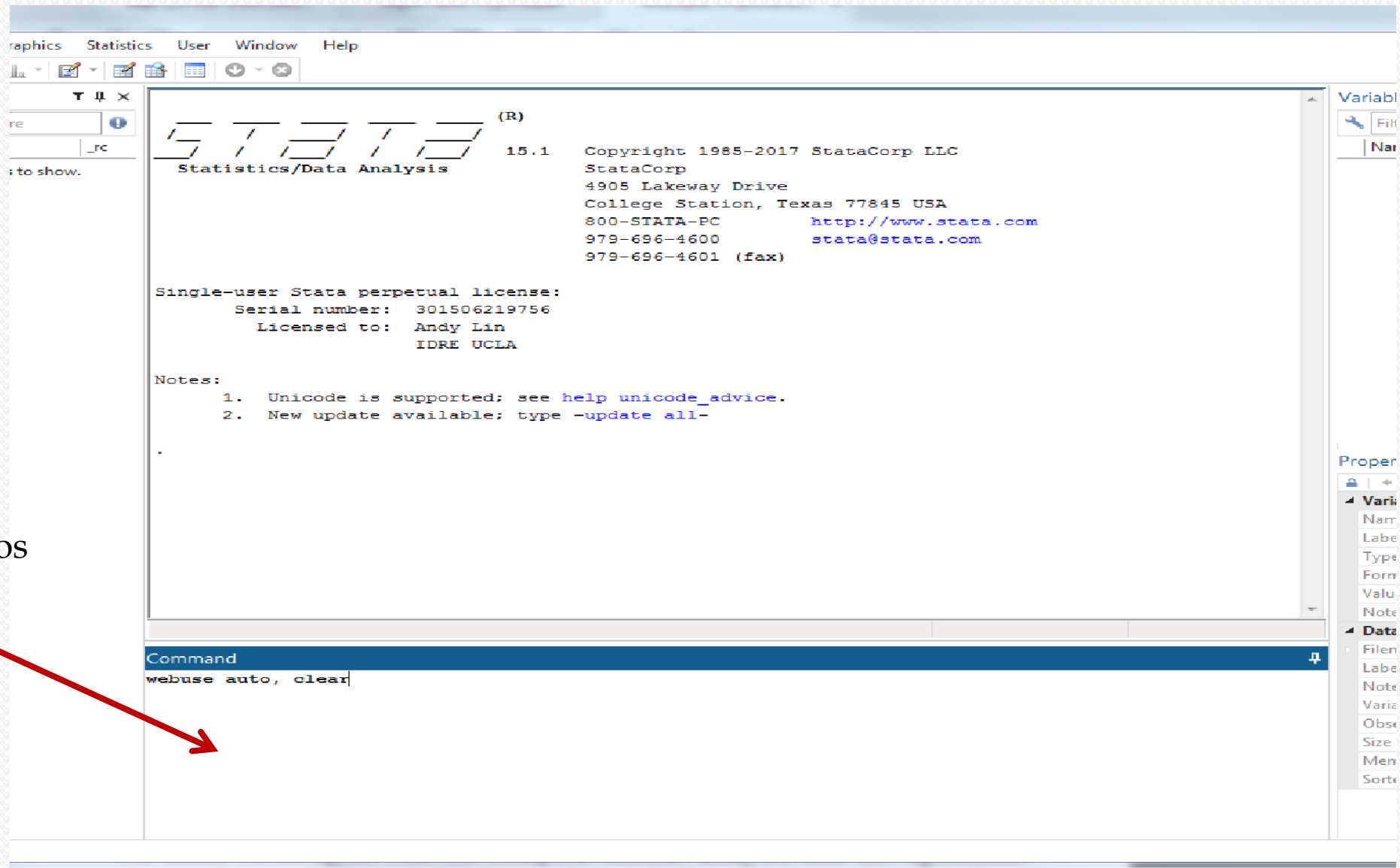
# STATA - Desventajas

- Solo puede cargar una base de datos en cada sesión
- Para trabajar con más bases de datos al mismo tiempo es necesario abrir otras sesiones de Stata
- Menor número de funcionalidades escritas por los mismos usuarios

# Navegando entre las funciones de STATA

# Ventana de comandos

- En esta ventana escribimos directamente los comandos .
- Escribe: `webuse auto, clear` , y fíjate qué sucede



The screenshot shows the Stata software interface. The main window displays the Stata startup screen, which includes the Stata logo, version 15.1, and copyright information. Below this, it shows the single-user perpetual license details, including the serial number 301506219756 and the user Andy Lin from IDRE UCLA. There are also notes about Unicode support and a new update available. At the bottom of the window, there is a command line where the command `webuse auto, clear` has been entered. A red arrow points from the text in the list to the command line.

```
STATA (R) 15.1
Statistics/Data Analysis

Copyright 1985-2017 StataCorp LLC
StataCorp
4905 Lakeway Drive
College Station, Texas 77845 USA
800-STATA-PC      http://www.stata.com
979-696-4600      stata@stata.com
979-696-4601 (fax)

Single-user Stata perpetual license:
  Serial number: 301506219756
  Licensed to:  Andy Lin
                IDRE UCLA

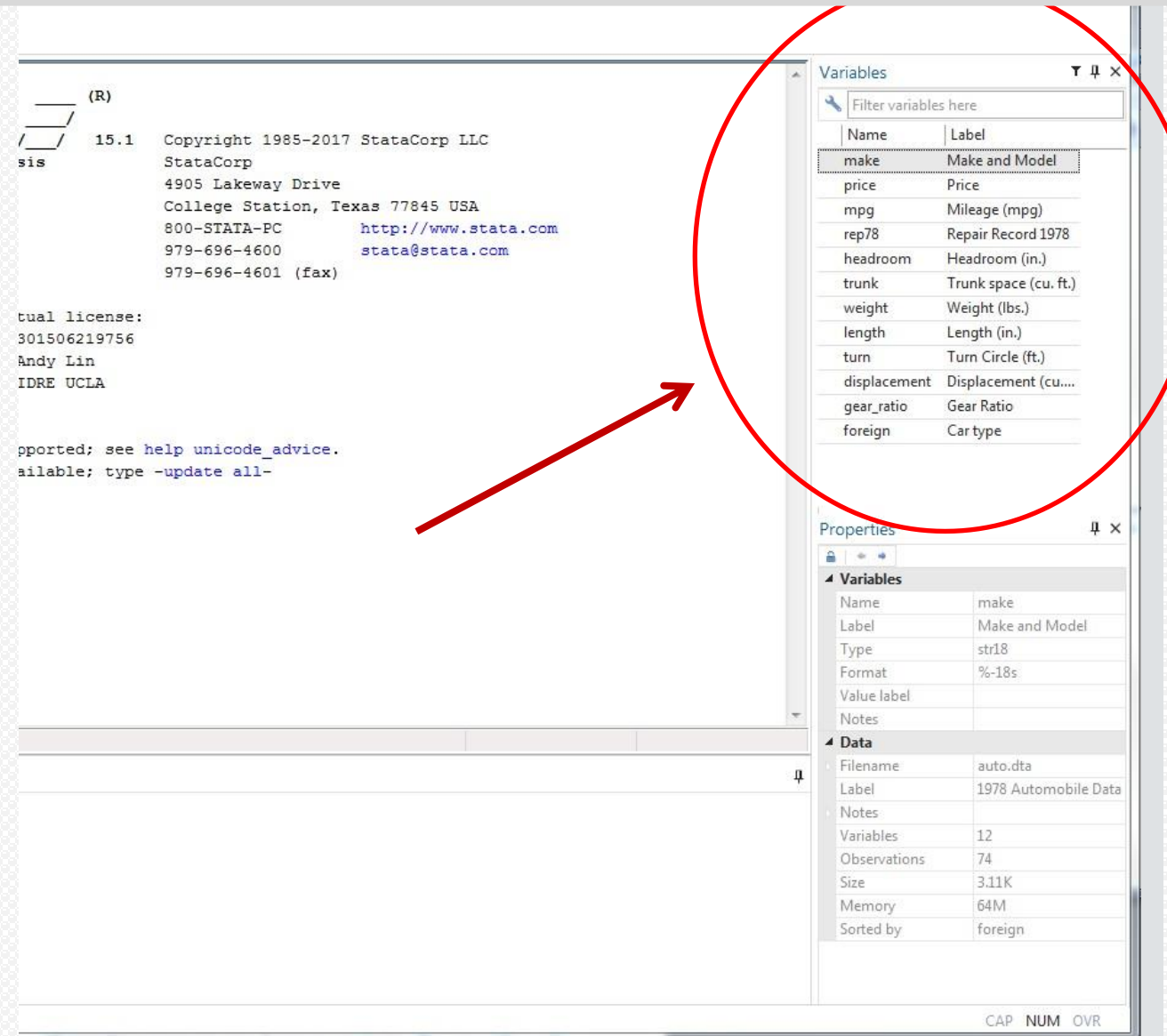
Notes:
  1. Unicode is supported; see help unicode_advice.
  2. New update available; type -update all-

.

Command
webuse auto, clear
```

# Ventana de variables

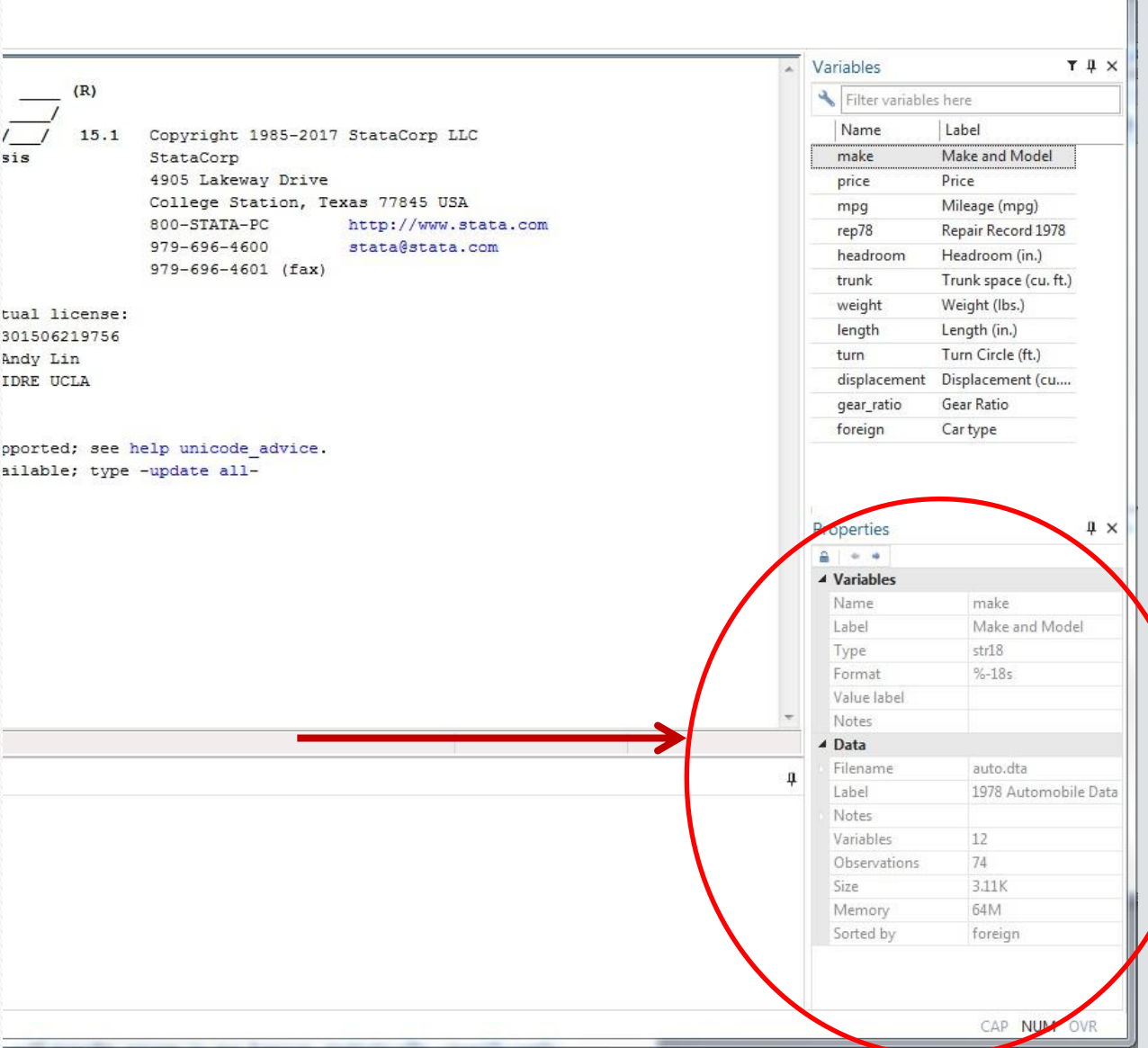
- Cuando tenemos datos cargados, esta ventana nos muestra el listado de variables que contiene la base con sus respectivas etiquetas.
- Al seleccionar una variable, aparecerá su información asociada en la **ventana de propiedades**.
- Doble click en la variable causa que esta aparezca en la ventana de comandos





# Ventana de propiedades

- Brinda información sobre cada una de las variables (cuando son seleccionadas de la lista), así como de toda la base de datos.
- Describe las características de la base de datos cargada



The screenshot shows the Stata Properties window with two tabs: 'Variables' and 'Properties'. The 'Variables' tab is active, displaying a list of variables with their names and labels. A red circle highlights the 'Properties' tab, which contains detailed information about the selected variable 'make' and the dataset 'auto.dta'.

**Variables**

Name	Label
make	Make and Model
price	Price
mpg	Mileage (mpg)
rep78	Repair Record 1978
headroom	Headroom (in.)
trunk	Trunk space (cu. ft.)
weight	Weight (lbs.)
length	Length (in.)
turn	Turn Circle (ft.)
displacement	Displacement (cu....
gear_ratio	Gear Ratio
foreign	Car type

**Properties**

**Variables**

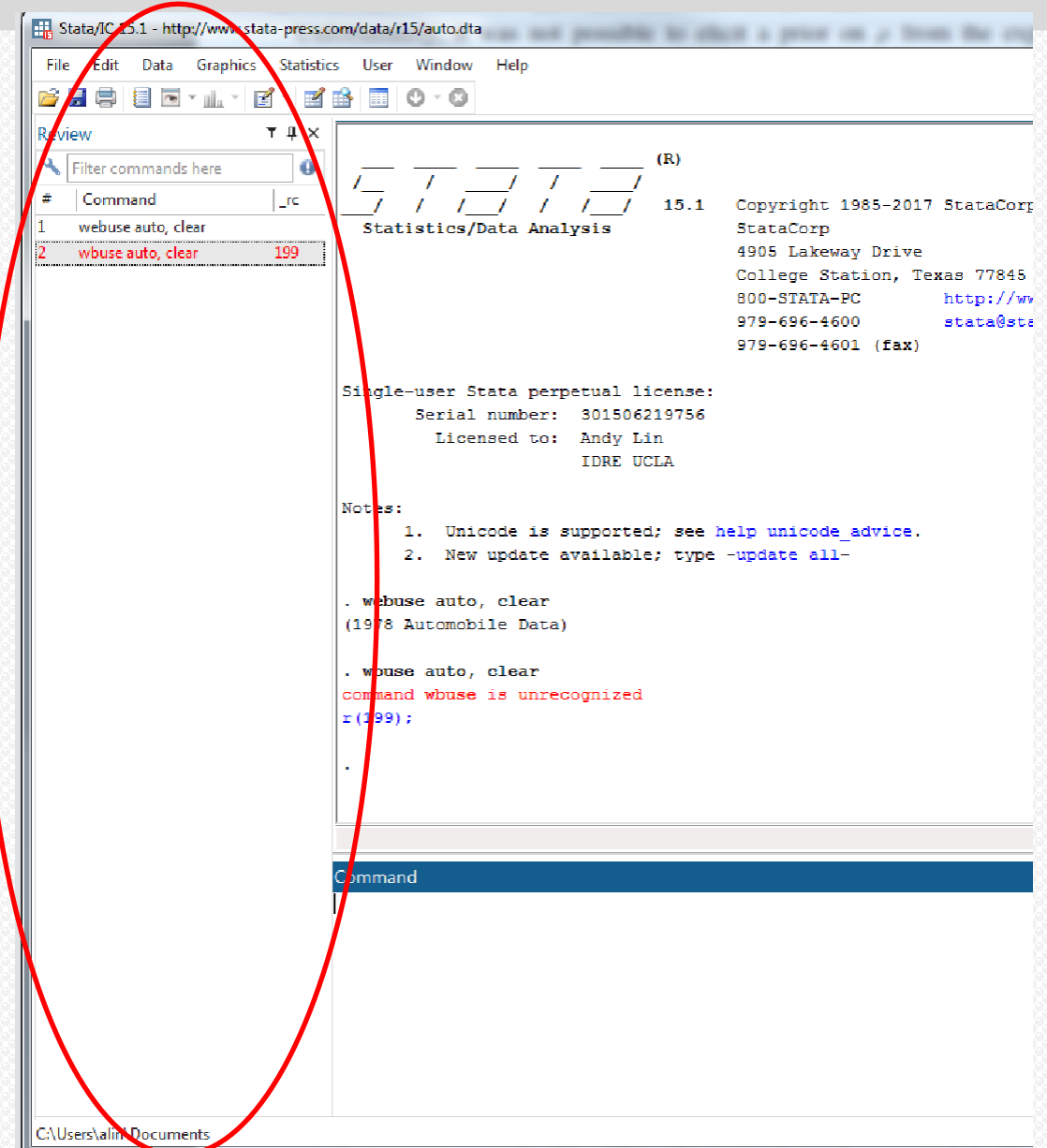
Name	Label
make	Make and Model
Type	str18
Format	%-18s
Value label	
Notes	

**Data**

Filename	Label
auto.dta	1978 Automobile Data
Notes	
Variables	12
Observations	74
Size	3.11K
Memory	64M
Sorted by	foreign

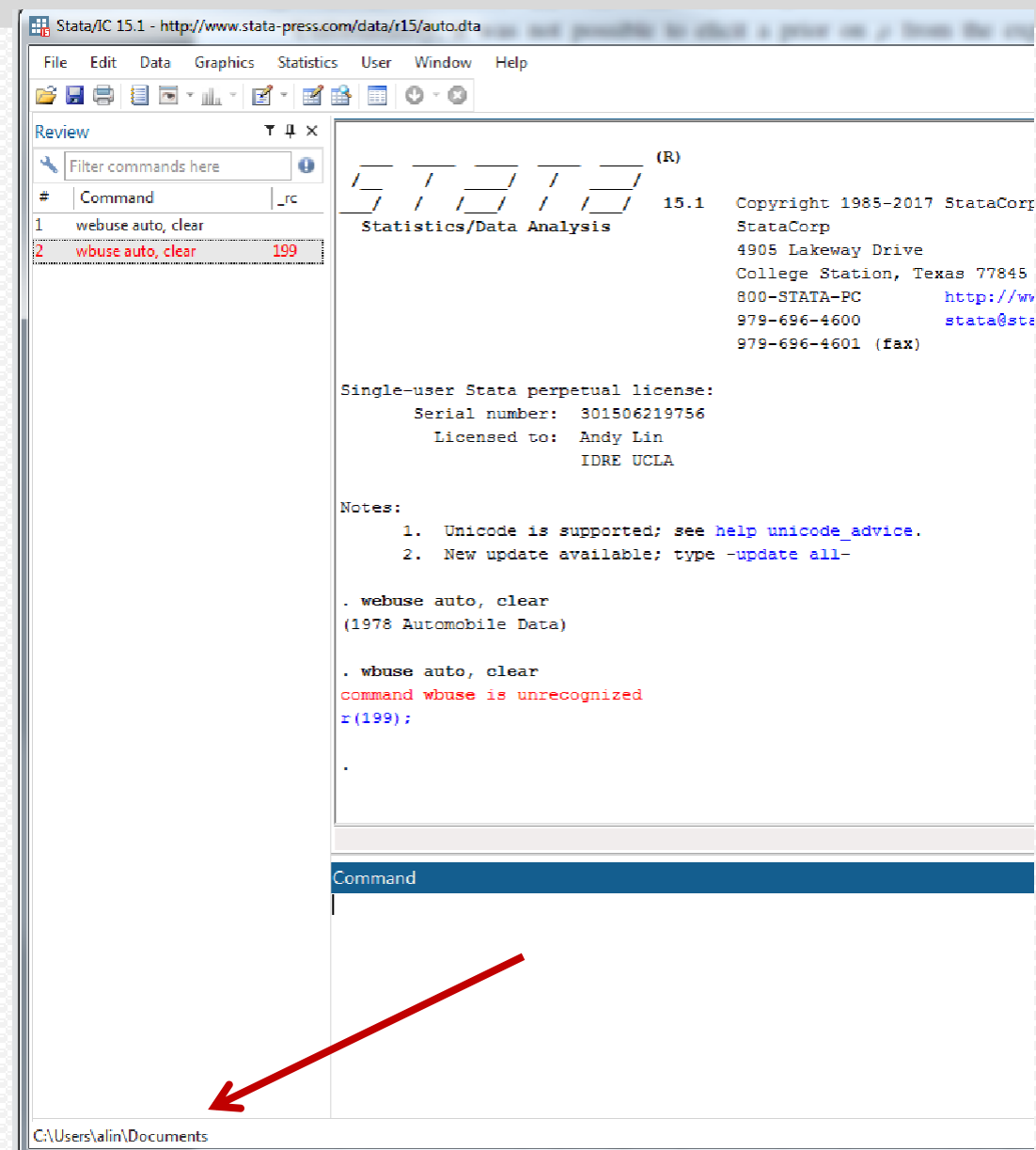
# Ventana de revisión

- Esta ventana registra todas las líneas de comando ejecutadas desde la ventana de comandos, así como aquellas que son ejecutadas desde un “Do-file” (concepto a revisar más adelante)
- Los errores serán marcados en color rojo
- Doble click en la línea de comando para que aparezca nuevamente en la ventana de comando
- Si presiono la tecla “Repag” del teclado, aparecerá el último comando ejecutado



# Directorio de trabajo

- El directorio de trabajo se muestra en la parte inferior izquierda de la ventana principal
- Los archivos serán cargados y guardados a partir de este directorio, a menos que se le especifique otro al programa
- Escribe “cd” (abreviatura de “current directory”) en la barra de comandos y observa el resultado
- Usando este comando se puede cambiar el directorio actual por otro



The screenshot shows the Stata/IC 15.1 interface. The top menu bar includes File, Edit, Data, Graphics, Statistics, User, Window, and Help. Below the menu bar is a toolbar with various icons. The main window is divided into two panes. The left pane, titled 'Review', contains a list of commands with a filter box at the top. The right pane displays the Stata startup screen, which includes the Stata logo, version information (15.1), copyright notice (1985-2017 StataCorp), and contact information for StataCorp. It also shows the single-user perpetual license details, including the serial number (301506219756) and the user (Andy Lin, IDRE UCLA). Below the license information, there are notes about Unicode support and a new update available. The command window at the bottom shows the command '. webuse auto, clear' being entered, and the output '(1978 Automobile Data)'. A red arrow points to the status bar at the bottom left, which displays the current directory path: 'C:\Users\alin\Documents'.

```
Stata/IC 15.1 - http://www.stata-press.com/data/r15/auto.dta
File Edit Data Graphics Statistics User Window Help
Filter commands here
# Command _rc
1 webuse auto, clear
2 wbuse auto, clear 199

(R)
Statistics/Data Analysis 15.1 Copyright 1985-2017 StataCorp
StataCorp
4905 Lakeway Drive
College Station, Texas 77845
800-STATA-PC http://www
979-696-4600 stata@stata
979-696-4601 (fax)

Single-user Stata perpetual license:
Serial number: 301506219756
Licensed to: Andy Lin
IDRE UCLA

Notes:
1. Unicode is supported; see help unicode_advice.
2. New update available; type -update all-

. webuse auto, clear
(1978 Automobile Data)

. wbuse auto, clear
command wbuse is unrecognized
r(199);

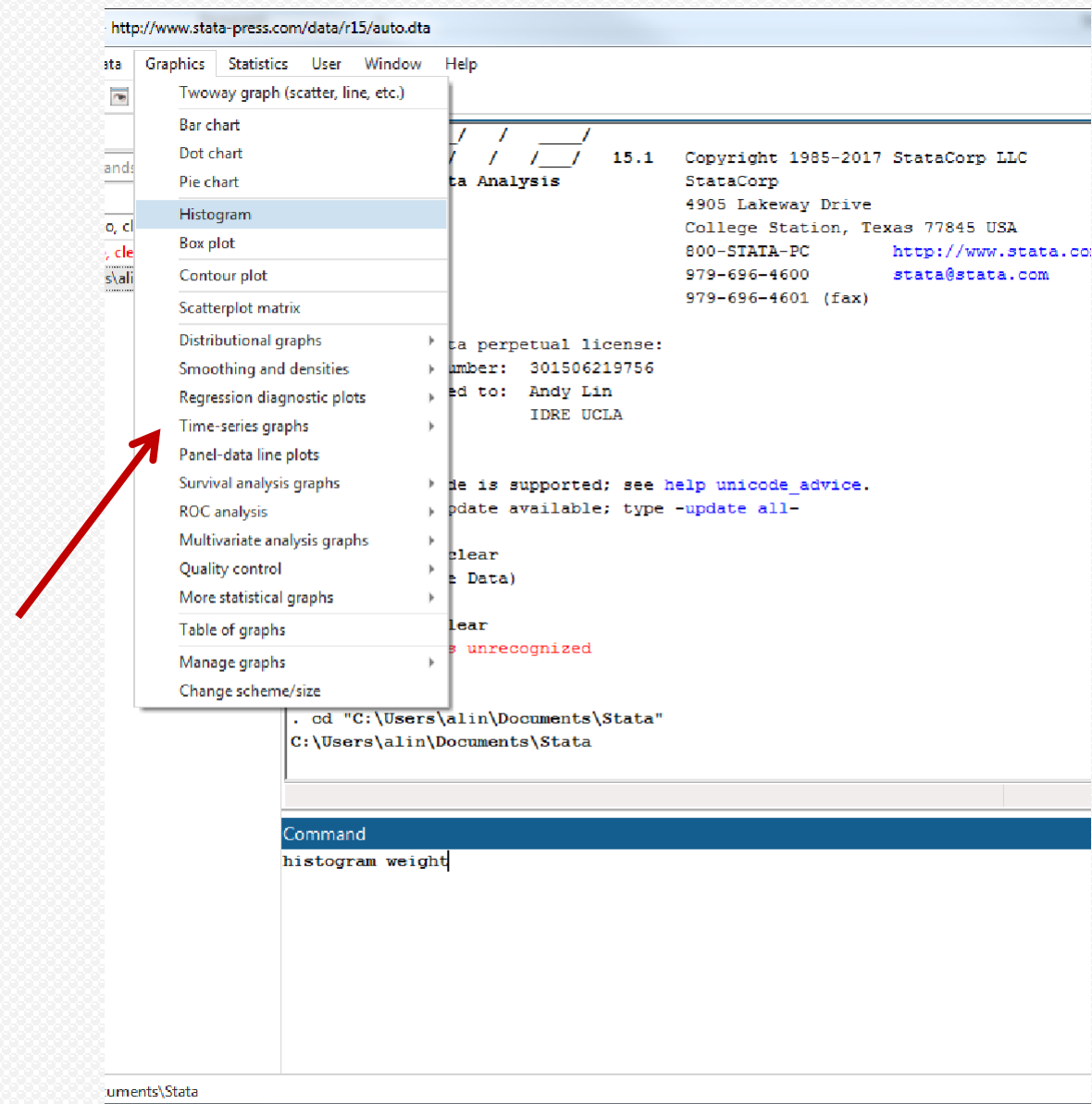
.

Command

C:\Users\alin\Documents
```

# Trabajando con Menú

- STATA también puede ser utilizado a través de ventanas y menús para ejecutar funciones.
- No obstante, la gran mayoría de usuarios prefiere ejecutar las funciones a través de líneas de comando o de Do-files (scripts de comandos)



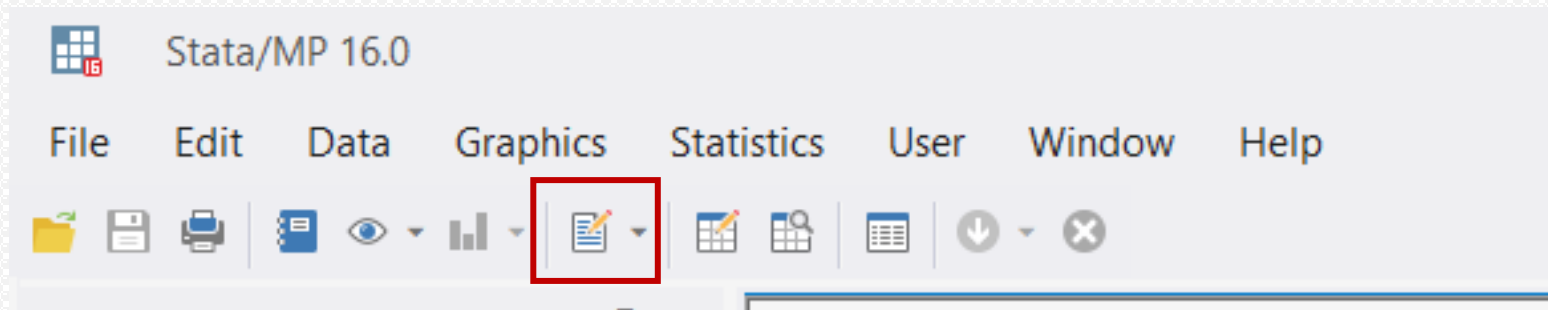
# El uso de Do-files

# Los Do-files

- Los Do-files son archivos de texto que almacenan una serie de comandos para ser reutilizados en el futuro, sin necesidad de tener que volver a escribirlos en la ventana de comandos.
- Brindan la ventaja de que son reproducibles, fáciles de ajustar y cambiar de acuerdo a nuestras necesidades
- Es altamente recomendable usar Do-files en vez de escribir solamente en la barra de comandos.
- Su extensión de archivo es **.do**

## Abriendo el editor de Do-files:

- Escribir el comando **doedit** en la barra de comandos, o hacer click en el icono de un lápiz y papel en la ventana principal de Stata:



# Los Do-files

## Colores de sintaxis:

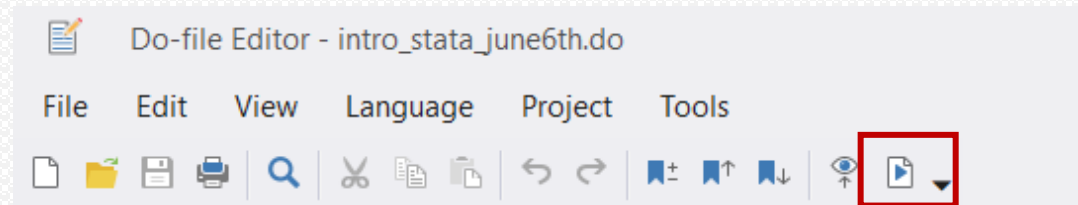
- Comandos de Stata: azul
- Comentarios: verde, y deben ser precedidos por \*
- Palabras en comillas (como nombres de archivos, valores strings) son de color rojo oscuro
- A partir de Stata 16 se presenta la función de autocompletar comandos.

```
Taller2023_Sesion_1_Introducci... X
36
37 /*-----
38 -- Importación de bases de datos --
39 -----*/
40
41 * Cargando base en formato Stata desde el disco duro:
42 use "hsfemale", clear
43
44 * Limpiando la memoria del programa (cuando ya se tiene cargada una base de datos)
45 clear
46
47 * Cargando base en formato Stata desde la web:
48 use https://stats.idre.ucla.edu/stat/data/hs0,clear
49
50 * Guardamos la base datos en nuestro disco duro
51 save "nueva_base"
52
53 * Si queremos reemplazarla (sobreescribirla), simplemente usamos la opción "replace"
54 save "nueva_base",replace
55
56
57 * Cargando un excel:
58 import excel using "hs0_excel.xlsx",sheet("Hoja1") firstrow clear
59
60 * Cargando un csv:
61 import delimited using "hs0.csv", clear
62
63 ** Guardandolo como base de Stata:
64 save "hs0_stata"
```

# Los Do-files

## Correr Do-files:

- Seleccionar la línea (o líneas) que se desea correr, y presionar las teclas CTRL+D (SHIFT+CMD+D en Mac), o el ícono de ejecución en la ventana del editor de Do-files:



## Añadir comentarios:

- Los comentarios no son ejecutados, y para insertarlos deben ser precedidos por un asterisco (\*)
- Si se quiere añadir un comentario o texto en varias líneas, el texto completo debe estar encerrado entre `/* ... */`
- Cuando el texto aparezca en verde, entonces Stata no los ejecutará como comando.

```
/* Comentarios como este pueden ser escritos  
en varias líneas  
sin necesidad de colocar asteriscos en cada línea */
```



# Importación de bases de datos

# Importación de datos

## Archivos en formato dta

- El formato de bases de datos en Stata es **.dta**
- A diferencia de otros softwares estadísticos, Stata solo puede tener abierta una base de datos en simultáneo, aunque se pueden añadir nuevas bases a la cargada sin necesidad de abrirlas.

## El comando use:

- Puede abrir archivos almacenados en el disco duro, pero también disponibles en Internet (a través de su respectiva dirección web)

# Importación de datos

## El comando save:

- Guarda las bases de datos trabajadas en el formato dta, el cual es el más eficiente para trabajar en Stata.
- Si se quiere sobrescribir una base existente, y que ha sido modificada, usar la opción **replace**
- Tener en cuenta que en la línea de comandos, la extensión **dta** para guardar los archivos puede ser omitida, ya que Stata la entiende de manera predeterminada.
- Los archivos se guardan en el “path” de la sesión (¡identifícalo!)

```
* Guardamos la base datos en nuestro disco duro
save "nueva_base"

* Si queremos reemplazarla (sobrescribirla), simplemente usamos la opción "replace"
save "nueva_base",replace
```

# Importación de datos

## Limpieza de memoria:

- Ya que Stata solo puede cargar una base de datos al mismo tiempo, es necesario limpiar la memoria cada vez que se requiera cargar una nueva base.
- Para ello, se debe usar el comando **clear** que remueve la base en uso de la memoria.

# Importación de datos

## Importando archivos de Excel:

- Para importar archivos de formato Excel, se utiliza el comando **import excel using**
- Es necesario indicar el “path” del archivo Excel (y en general de cualquier otro formato) para abrir aquellas bases que no se encuentran en el actual directorio de trabajo.
- La opción **sheet()**, permite indicar si queremos trabajar con una hoja particular del libro de Excel.
- Además, usar la opción **firstrow** para ordenar a Stata que considere a la primera fila de la base de datos como los nombres de las columnas.

# Importación de datos

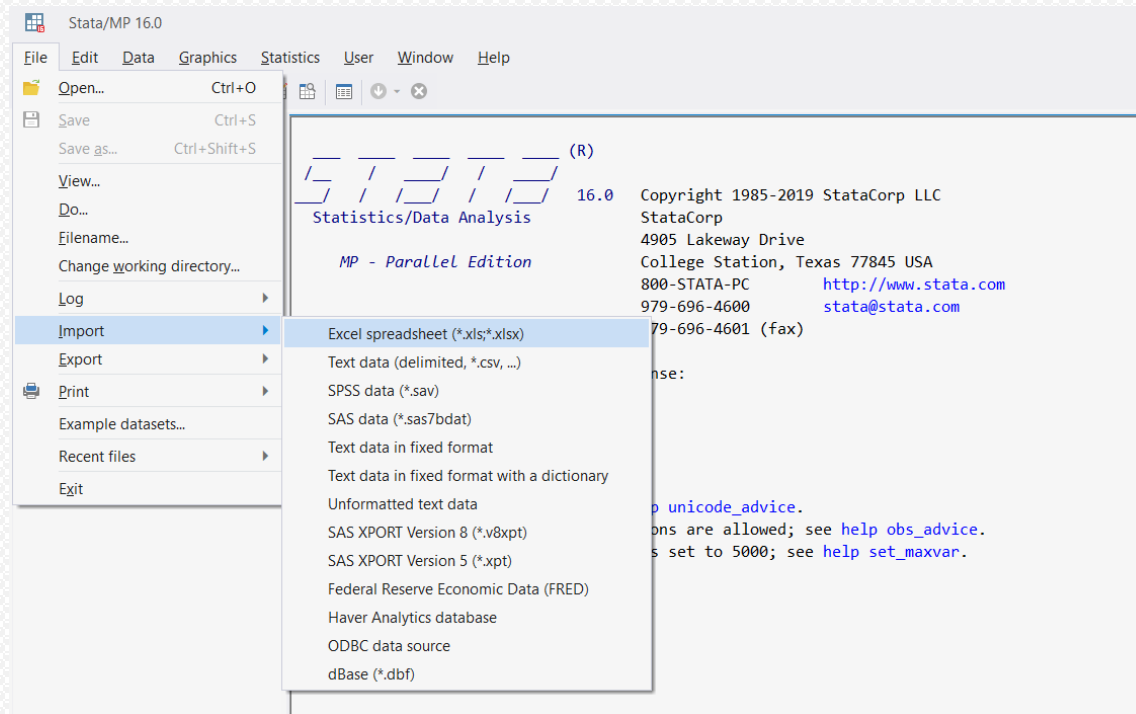
## Importando archivos .csv

- Archivos Csv (comma-separated values) pueden ser abiertos usando el comando **import delimited using**
- La sintaxis es muy semejante a la usada para abrir archivos de Excel, pero en esta ocasión no se especifica una hoja de trabajo, porque los archivos .csv no las tienen.

# Importación de datos

## Usando el menú para importar archivos Excel o Csv

- Menús pueden ser útiles para cargar estos archivos, especialmente si la ruta del path es muy larga.
- Seguir la siguiente ruta de botones de menú:
  - File → Import → Excel spreadsheet / Text data (delimited, \*.csv...)

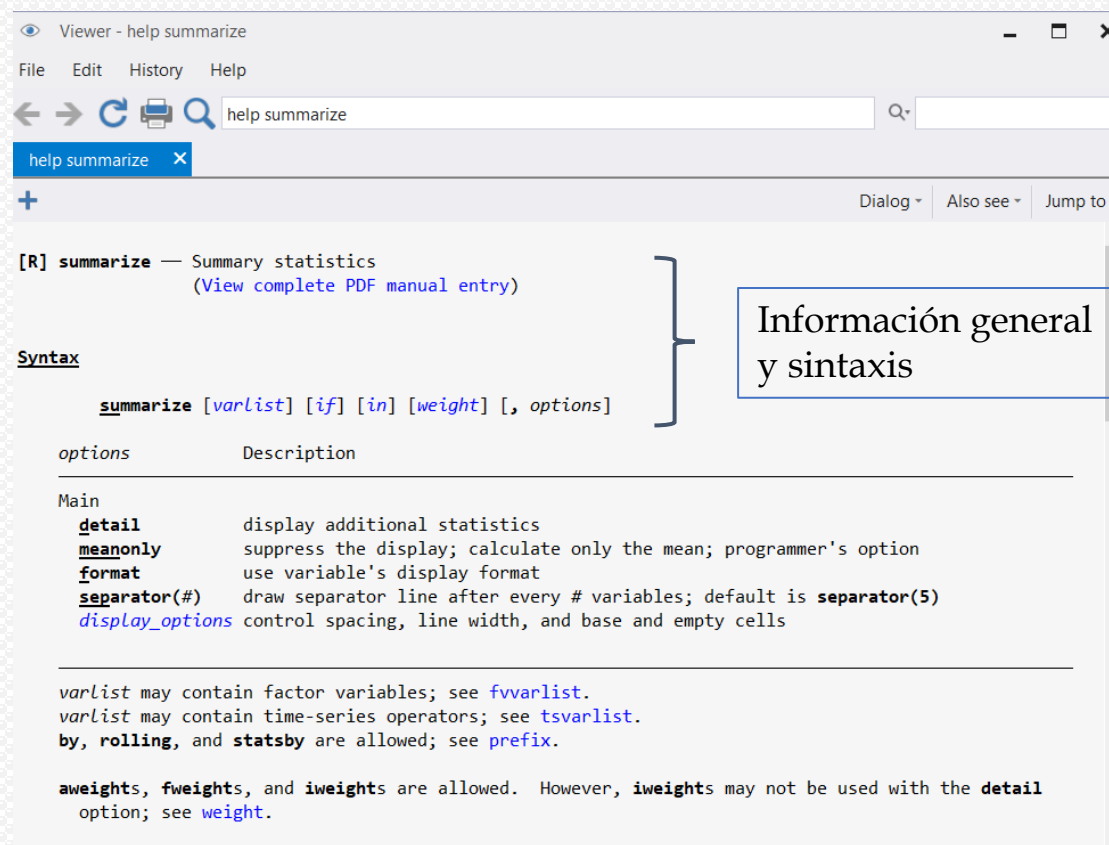


# Help files



# Help files

- En la barra de comandos precede un nombre de comando por la palabra **help** y aparecerán recursos que brindan información sobre el uso de dicho comando y sus opciones, incluyendo ejemplos.
- Prueba: `help summarize`



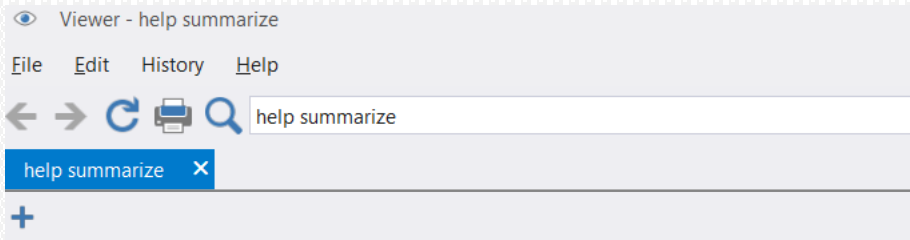
Información general  
y sintaxis

Opciones y detalles  
adicionales

# Help files

## Ejemplos:

- Los help files también brindan ejemplos aplicados con bases destinadas para aprendizaje
- En algunas ocasiones, incluso existen tutoriales en video para el comando requerido.



### Examples

```
. sysuse auto
. summarize
. summarize mpg weight
. summarize mpg weight if foreign
. summarize mpg weight if foreign, detail
. summarize i.rep78
```

Ejemplos aplicados sobre el comando y sus opciones

### Video example

[Descriptive statistics in Stata](#)

Video tutorial disponible

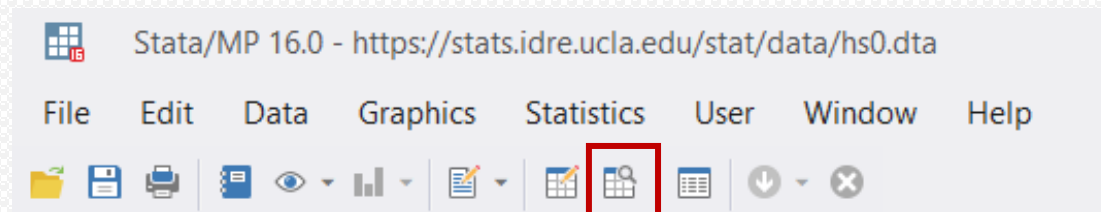
Se puede obtener más videotutoriales en el canal oficial de StataCorp en Youtube:

<https://www.youtube.com/user/statacorp>

# Visualizando la información

# Visualizar la información de la base de datos

- Una vez que los datos han sido cargados, pueden ser visualizados como una tabla u hoja de cálculo usando el comando **browse**, o usando el correspondiente icono.



Data Editor (Browse) - [hs0.dta]

File Edit View Data Tools

gender[1] 1

	gender	id	race	ses	schtyp	prgtype	read	write	math	science	socst
1	1	70	white	low	1	general	57	52	41	47	57
2	2	121	white	middle	1	vocati	68	59	53	63	61
3	1	86	white	high	1	general	44	33	54	58	31
4	1	141	white	high	1	vocati	63	44	47	53	56
5	1	172	white	middle	1	academic	47	52	57	53	61
6	1	113	white	middle	1	academic	44	52	51	63	61
7	1	50	african-amer	middle	1	general	50	59	42	53	61
8	1	11	hispanic	middle	1	academic	34	46	45	39	36
9	1	84	white	middle	1	general	63	57	54	.	51

- Columnas negras: datos numéricos
- Columnas rojas: datos tipo "string" (texto)
- Columnas azules: datos numéricos pero etiquetados.

# Visualizar la información de la base de datos

- El comando **list** nos permite visualizar los datos en la pantalla principal de Stata
- Es recomendable especificar el nombre de las variables de interés luego del comando para que la pantalla no se llene de mucha información (ver ejemplo)

```
. list read write
```

	read	write
1.	57	52
2.	68	59
3.	44	33
4.	63	44
5.	47	52
6.	44	52
7.	50	59
8.	34	46
9.	63	57
10.	57	55

# Seleccionando información

# Seleccionando información

## Opción in:

- La opción **in** selecciona la fila de observaciones que queremos visualizar, indicando la primera y última observación de interés (ver ejemplo)
- También se puede usar la opción en negativo, y la especificación “L”, particularmente cuando queremos visualizar la información de las últimas filas.
- **Ejercicio:** visualiza las últimas 10 observaciones para tres variables de tu elección.

```
. list read write in 10/15
```

	read	write
10.	57	55
11.	60	46
12.	57	65
13.	73	60
14.	54	63
15.	45	57

```
. list read write in -3/L
```

	read	write
198.	57	41
199.	55	62
200.	63	65

# Seleccionando información

## Opción if:

- La opción **if** permite seleccionar un subconjunto de información de acuerdo a una serie de condiciones.
- Esta opción se especifica luego del comando, pero antes de la coma.

## Veamos más opciones de comandos relacionales:

- ✓ Igual a: ==
- ✓ Mayor que: >
- ✓ Mayor o igual que: >=
- ✓ Menor que: <
- ✓ Menor o igual que: <=
- ✓ No: !
- ✓ No es igual que: !=
- ✓ Y: &
- ✓ O: |

```
. list read write if write>65
```

	read	write
33.	65	67
73.	73	67
118.	73	67
160.	66	67
177.	65	67
183.	52	67
185.	50	67

```
browse gender ses read if gender==2 & read>70
```

Data Editor (Browse) - [hs0.dta]

File Edit View Data Tools

gender[97] 2

	gender	ses	read
97	2	middle	71
110	2	high	71
115	2	high	73
118	2	high	73
136	2	high	76



# Ejercicio aplicado

Use la base de datos de ejemplo con información socioeconómica de estudiantes y su rendimiento académico llamada **hs0**, la cual puede abrir con el siguiente comando: use <https://stats.idre.ucla.edu/stat/data/hs0>

- Explore la base de datos y determina cuántas observaciones tiene y cuántas variables
- ¿Cuántas variables son etiquetadas, y cuántas son de tipo string (texto)
- Haz un **browse** de la información de estudiantes con puntajes de “read” y “math” mayores a 65 por separado y en simultáneo.
- Muestra en la pantalla principal de Stata a las observaciones entre las filas 20 y 35 para las variables “read” y “math”

# Explorando mis datos

# Explorar datos

- Es muy recomendable realizar una exploración de los datos antes de iniciar el análisis, y así asegurarnos de corregir posibles errores.
- Este análisis exploratorio generalmente debe considerar la revisión de las características de las unidades de observación, así como la distribución de las variables clave.

## El commando `codebook`:

- Brinda un resumen de las variables, incluyendo: número de valores únicos y missings, rangos, quintiles, media, desviación estándar para el caso de variables numéricas.
- Si las variables son de tipo “string” nos brindará información sobre frecuencias

```
. codebook read prgtype
```

read

reading score

```

      type:  numeric (float)
      range:  [28,76]
unique values: 30
      mean:   52.23
      std. dev: 10.2529
      units:  1
      missing.: 0/200

percentiles:      10%      25%      50%      75%      90%
                  39       44       50       60       67
```

prgtype

(unlabeled)

```

      type:  string (str8)
unique values: 3
      missing "": 0/200

tabulation:  Freq.  Value
              105  "academic"
              45   "general"
              50   "vocati"
```

# Explorar datos

## Resumen estadístico con summarize:

- Información sobre el número de observaciones no missing, media, desviación estándar, mínimo y máximo

```
. summarize read math
```

Variable	Obs	Mean	Std. Dev.	Min	Max
read	200	52.23	10.25294	28	76
math	200	52.645	9.368448	33	75

```
. summarize math,detail
```

math score				
Percentiles		Smallest		
1%	36	33		
5%	39	35		
10%	40	37	Obs	200
25%	45	38	Sum of Wgt.	200
50%	52		Mean	52.645
		Largest	Std. Dev.	9.368448
75%	59	72	Variance	87.76781
90%	65.5	73	Skewness	.2844115
95%	70.5	75	Kurtosis	2.337319
99%	74	75		

## Summarize con opción detail:

- Agrega más detalles como valores en percentiles, valores más altos y bajos, varianza, skewness y kurtosis

# Explorar datos

## Tabulaciones con `tabulate (tab)`:

- Muestra información sobre frecuencias de valores.
- Particularmente útil para variables categóricas.
- En el caso de variables etiquetadas, dichas etiquetas se remueven con la opción `no label`

```
. tab ses
```

ses	Freq.	Percent	Cum.
low	47	23.50	23.50
middle	95	47.50	71.00
high	58	29.00	100.00
Total	200	100.00	

```
. tab ses,no label
```

ses	Freq.	Percent	Cum.
1	47	23.50	23.50
2	95	47.50	71.00
3	58	29.00	100.00
Total	200	100.00	

## Tabulaciones cruzadas:

- Muestra frecuencias en tablas cruzadas. Solo hace falta indicar dos variables luego de `tabulate`
- Se puede incorporar información porcentual de filas y columnas usando las opciones `row` y `col` respectivamente.

```
. tab prgtype ses
```

prgtype	ses			Total
	low	middle	high	
academic	19	44	42	105
general	16	20	9	45
vocati	12	31	7	50
Total	47	95	58	200

`tab prgtype ses, row`

Key				
frequency				
row percentage				
prgtype	low	ses middle	high	Total
academic	19 18.10	44 41.90	42 40.00	105 100.00
general	16 35.56	20 44.44	9 20.00	45 100.00
vocati	12 24.00	31 62.00	7 14.00	50 100.00
Total	47 23.50	95 47.50	58 29.00	200 100.00

# Ejercicio aplicado

Usando la base de datos del ejercicio anterior (hs0) explore:

- ¿Cuántas persona de raza (race) white hay en la base de datos?
- ¿Cuál es la etiqueta para el sector socioeconómico (ses) high?
- ¿Cuántos estudiantes de raza blanca llevan el programa (prgtype) general?
- Encuentre el promedio de notas de “math” para los estudiantes de raza (race) blanca (white) y de estado socioeconómico (ses) alto (high)
- Encuentre el promedio de notas de “read” para los estudiantes que llevan el programa (prgtype) académico (academic)

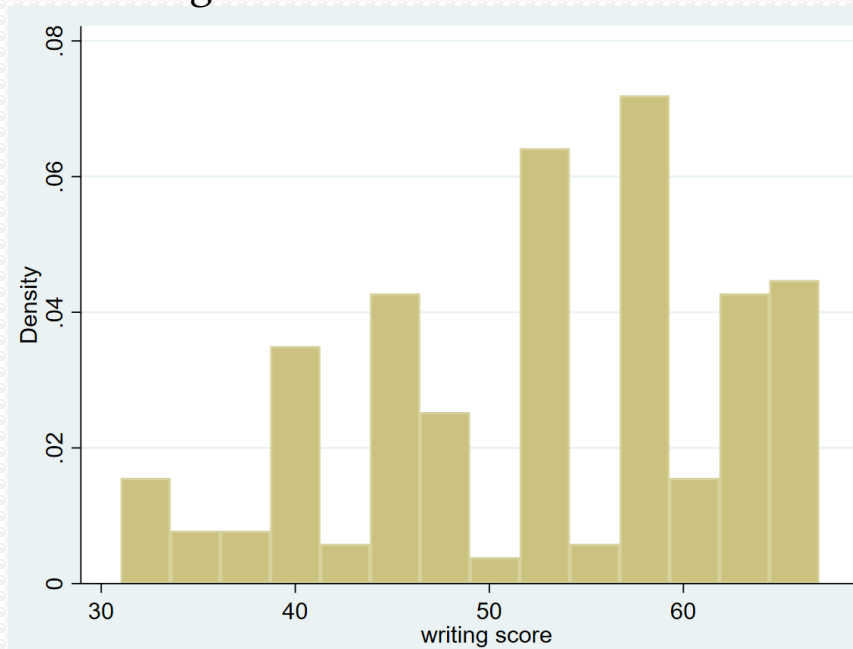
# Visualización de datos

# Visualización de datos

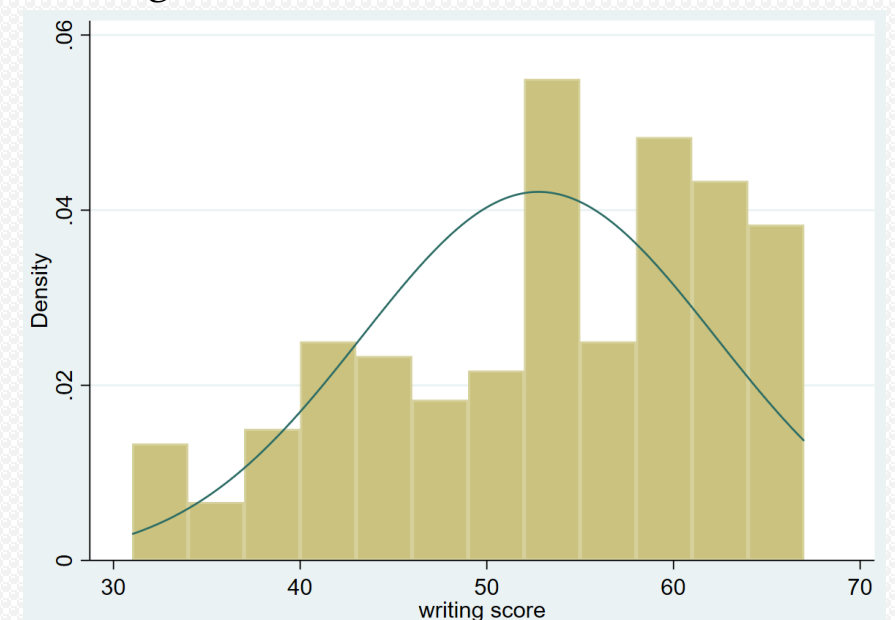
## Histogramas:

- Gráfico de la distribución de variables de acuerdo a intervalos de valores.
- Se puede colocar la opción **normal** para incorporar un gráfico comparativo de distribución normal sobre nuestro histograma.
- De igual manera se puede especificar la opción **width()** para indicar el ancho de las barras del histograma

*histogram write*



*histogram write, normal width(7)*



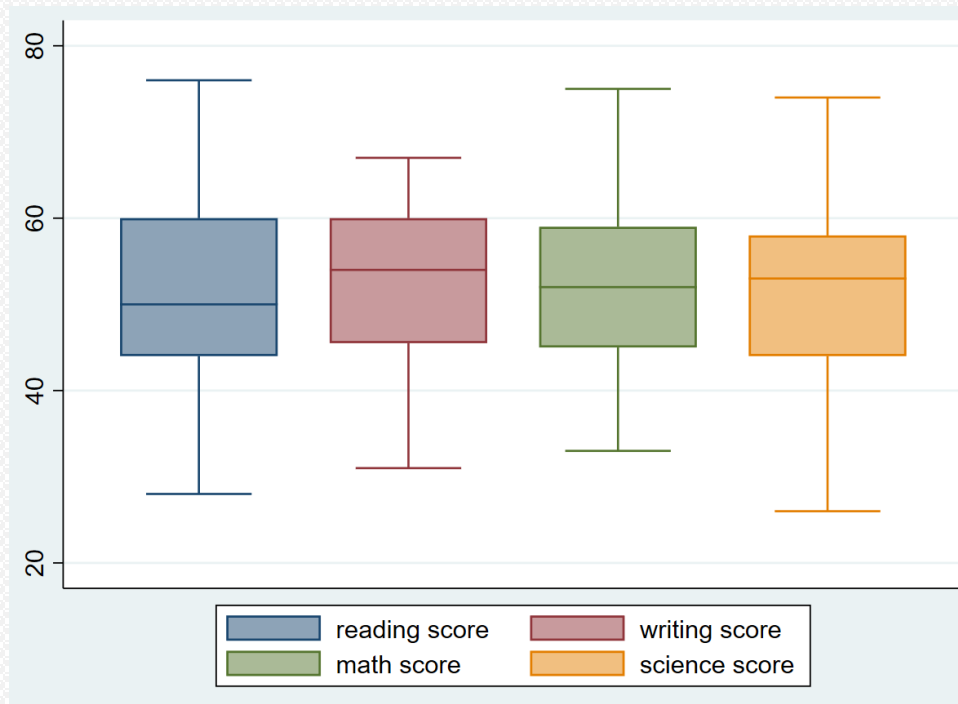


# Visualización de datos

## Boxplots:

- Muestran la distribución de variables continuas, y facilita la comparación entre categorías.
- Muestra el valor promedio, los valores intercuartiles, así como los “outliers”
- Se puede graficar boxplots para diferentes variables en un mismo panel

*graph box read write math science*



*graph box read write math science, horizontal*

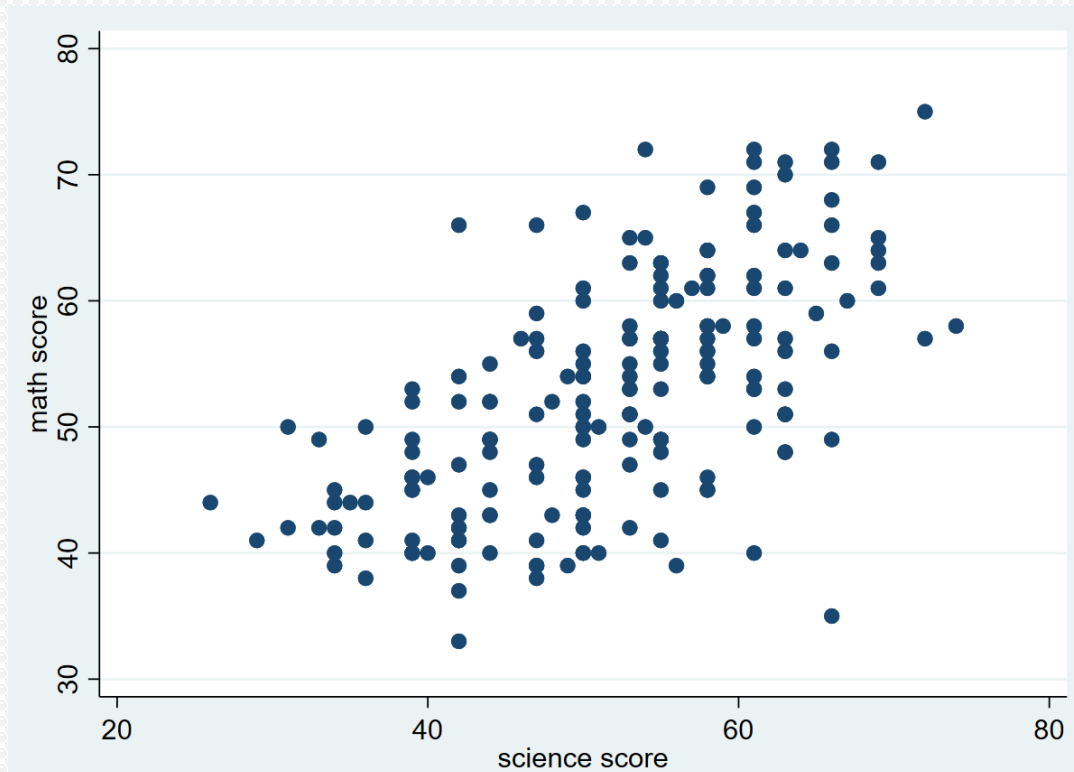


# Visualización de datos

## Scatterplots:

- Explora la relación entre dos variables continuas
- En la sintaxis, primero se coloca la variable que irá en el eje y, y luego la que irá en el eje x

*scatter math science*

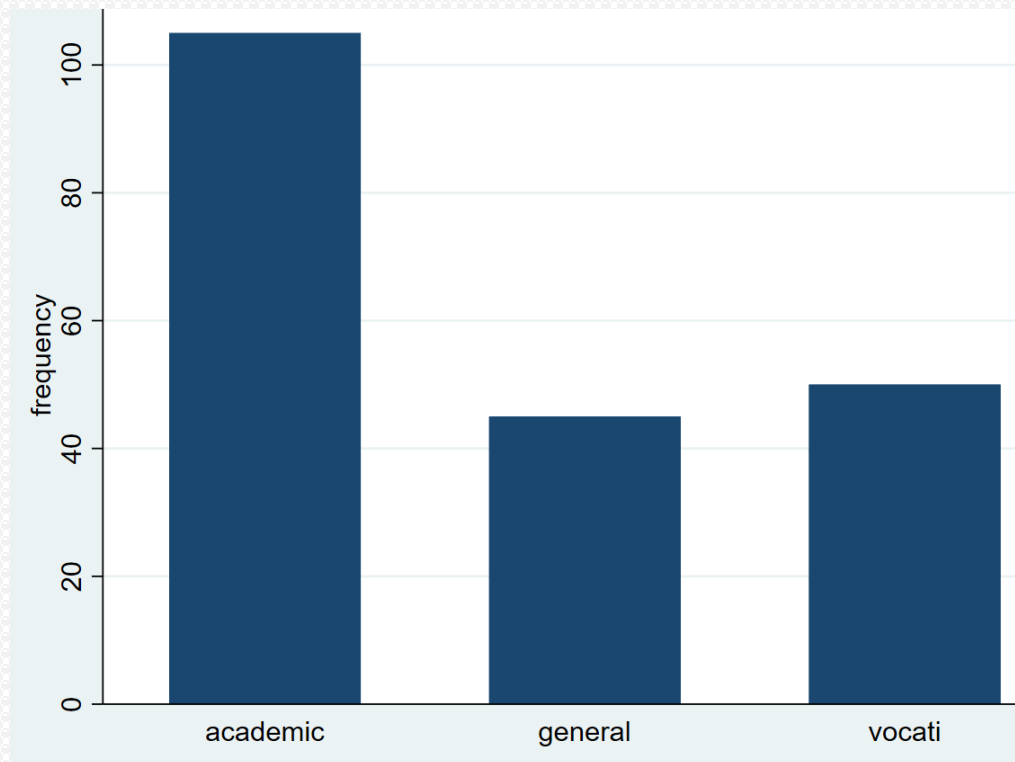


# Visualización de datos

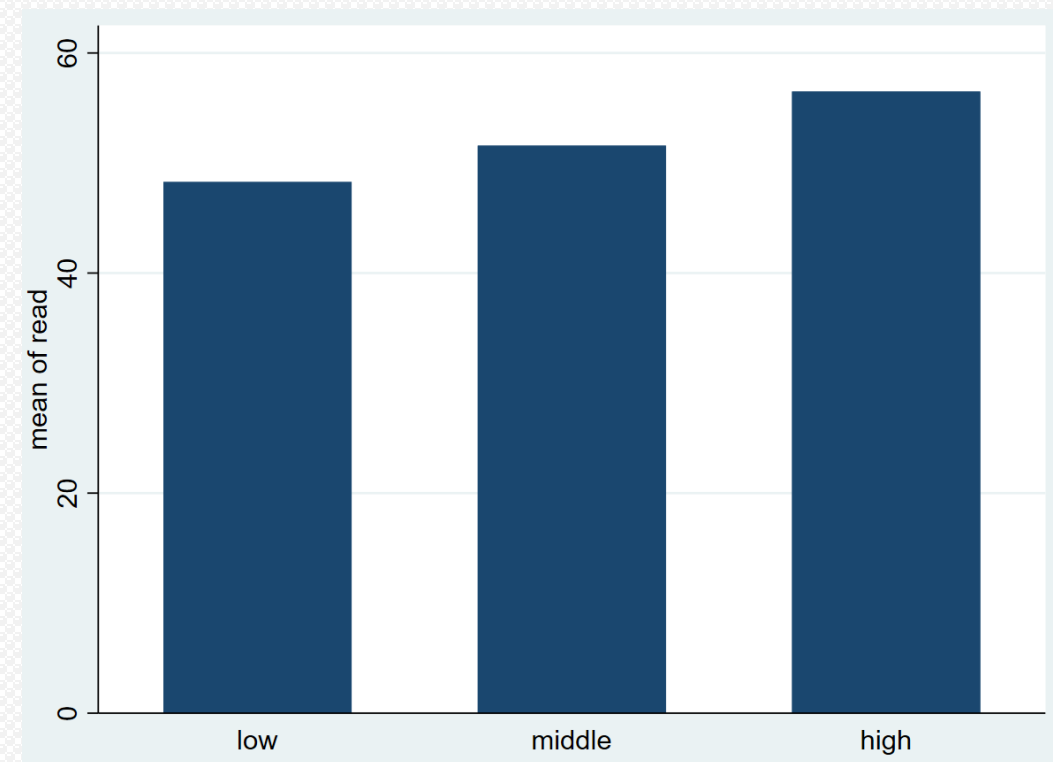
## Gráficos de barras:

- Usados para explorar frecuencias de una o más variables, de acuerdo a diferentes categorías
- Las opciones más utilizadas son la especificación del indicador, así como el señalamiento de las categorías (over)

*graph bar (count), over(prgtype)*



*graph bar (mean) read, over(ses)*



# Visualización de datos

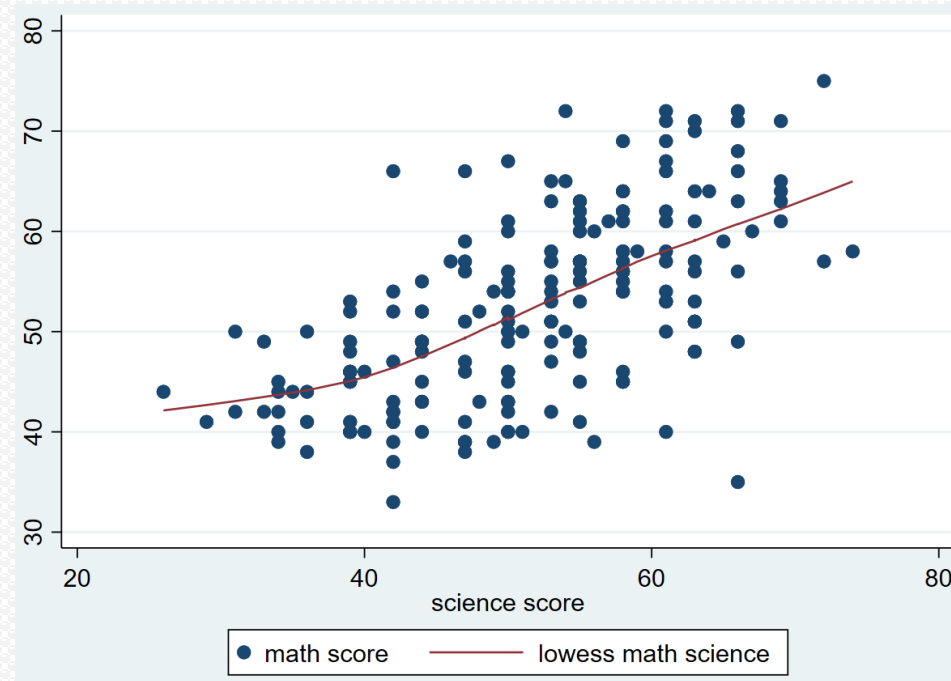
## Gráficos superpuestos con “`twoway`”:

- El comando **`twoway`** de Stata permite superponer dos o más ploteos en un solo lienzo
- Cada ploteo debe involucrar un par de variables (una para cada eje)

La sintaxis general es:

```
twoway (ploteo1 vary varx) (ploteo2 vary varx) (ploteo3 vary varx)...
```

*twoway (scatter math science) (lowess math science)*



# Ejercicio aplicado

Usando la base de datos del ejercicio anterior (hs0):

- Realice un gráfico de scatter y determine la relación entre dos notas de su elección (read, write, math, etc.)
- Para el gráfico anterior, cambie los marcadores de scatter por triángulos en vez de círculos (usar help para saber cómo hacerlo)
- Realice un gráfico de scatter entre math y science diferenciando por las tres categorías de “ses”, usando el comando **twoway**.

# Gestión de datos

# Gestión de datos

## Creación de variables:

- A menudo es necesario crear nuevas variables para obtener la información que necesitamos.
- Usamos el comando **generate (gen)** para realizar transformaciones y operaciones entre variables.
- Por lo general si combinamos dos o más variables en las que existen valores missing, entonces la operación aplicada no arrojará resultado para dichas observaciones.

```
. generate total = math + science + socst  
(5 missing values generated)
```

```
. summarize total math science socst
```

Variable	Obs	Mean	Std. Dev.	Min	Max
total	195	156.4564	24.63553	96	213
math	200	52.645	9.368448	33	75
science	195	51.66154	9.866026	26	74
socst	200	52.405	10.73579	26	71

# Gestión de datos

## Los missing values:

- Los missing values de una variable numérica son representados por “.”
- En el caso de variables string, los missing values son “” (comillas vacías)
- Los valores missing son obviados de las operaciones y los análisis numéricos de manera automática

```
. list math science socst if science == .
```

	math	science	socst
9.	54	.	51
18.	60	.	56
37.	75	.	66
55.	73	.	66
76.	43	.	31

```
. li read write total if missing(total)
```

	read	write	total
9.	63	57	.
18.	57	57	.
37.	68	54	.
55.	73	62	.
76.	47	40	.



# Gestión de datos

## Reemplazando valores:

- Usar el comando **replace** para reemplazar los valores de alguna variable.
- Por lo general se debe especificar una condición para hacer el reemplazo (if)

```
. replace total=80 if total==.  
(5 real changes made)
```

```
. sum total
```

Variable	Obs	Mean	Std. Dev.	Min	Max
total	200	154.545	27.10834	80	213

# Gestión de datos

## Creación de variables extendida

- Con el comando **egen** se puede crear variables haciendo uso de un abanico más extenso de funciones incluyendo cálculos estadísticos, estandarizaciones, entre otras de una manera más rápida y eficiente.
- Además, si alguna variable tiene valor missing, no es tomada en cuenta para hacer el cálculo requerido.

```
. egen meantest = rowmean(read math science socst)

.
. summarize meantest read math science socst
```

Variable	Obs	Mean	Std. Dev.	Min	Max
meantest	200	52.28042	8.400239	32.5	70.66666
read	200	52.23	10.25294	28	76
math	200	52.645	9.368448	33	75
science	195	51.66154	9.866026	26	74
socst	200	52.405	10.73579	26	71

```
. summarize zread
```

Variable	Obs	Mean	Std. Dev.	Min	Max
zread	200	-1.84e-09	1	-2.363225	2.31836

# Gestión de datos

## Renombrar variables:

- Usa el comando **rename**
- Sintaxis: `rename nombre_actual nombre_nuevo`

## Recodificar variables:

- Cambia el valor de una variable por otro especificado
- Sintaxis: `recode (valor actual=valor nuevo) (valor actual=valor nuevo) ...`

```
. recode female (1=0)(2=1)  
(female: 200 changes made)
```

```
.  
. tab female
```

female	Freq.	Percent	Cum.
0	91	45.50	45.50
1	109	54.50	100.00
Total	200	100.00	

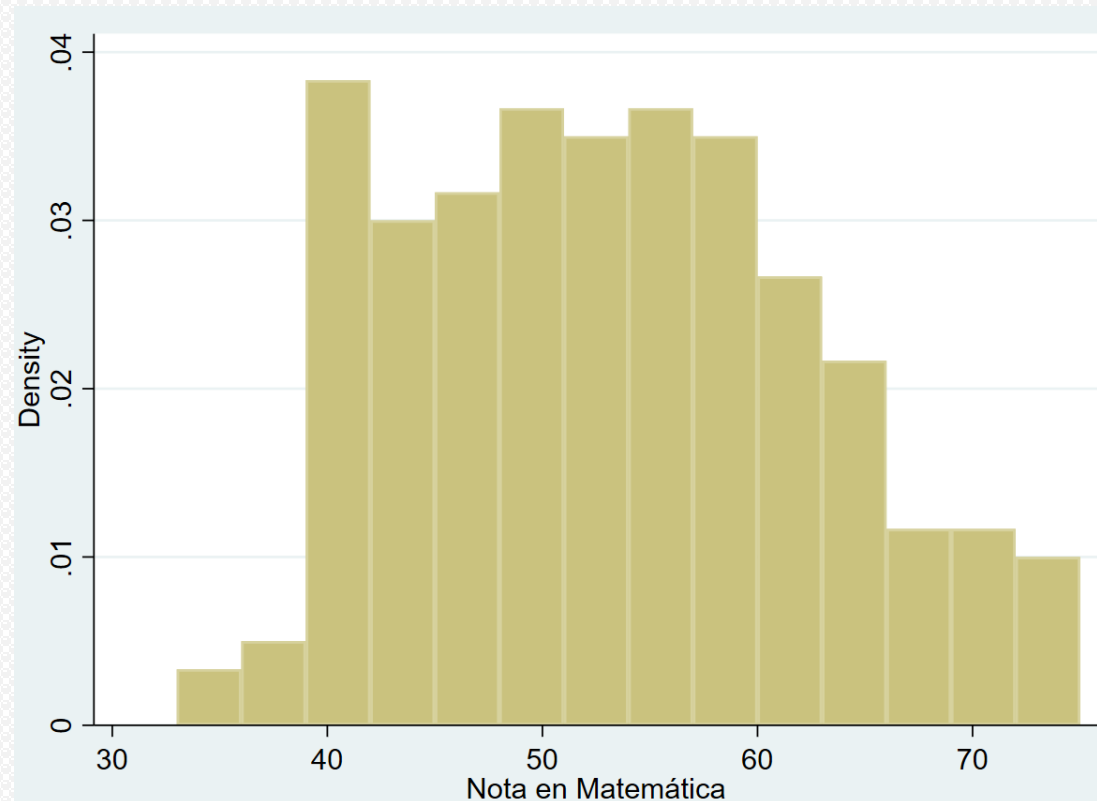
# Gestión de datos

## Etiquetado de variables:

- Los nombres cortos de las variables pueden brindarnos más eficiencia, pero a veces son poco informativos.
- Para estos casos se puede utilizar el etiquetado de variables mediante el comando **label variable**
- Sintaxis: *label variable “la etiqueta que deseo”*

```
label variable math "Nota en Matemática"
```

```
histogram math  
bin=14, start=33, width=3)
```



# Gestión de datos

## Etiquetado de valores:

- Las etiquetas de valores brindan información sobre los valores numéricos de una variable, especialmente cuando estos representan una categoría.
- Se usa el comando **label define** para crear la variable, y el comando **label values** para aplicar las etiquetas

- Sintaxis completa:

*label define nombre\_etiqueta # etiqueta1 # etiqueta2 ...*

*label values variable nombre\_etiqueta*

```
. label define tipo_escuela 1 "Pública" 2 "Privada"
```

```
. label values schtyp tipo_escuela
```

```
. tab schtyp
```

schtyp	Freq.	Percent	Cum.
Pública	168	84.00	84.00
Privada	32	16.00	100.00
Total	200	100.00	

```
. tab schtyp, nolabel
```

schtyp	Freq.	Percent	Cum.
1	168	84.00	84.00
2	32	16.00	100.00
Total	200	100.00	

# Gestión de datos

## Codificar variables string a numéricas:

- El comando **encode** convierte una variable string a una numérica etiquetada
- Las etiquetas son asignadas por orden alfabético

Sintaxis: *encode variable, gen(nueva\_variable)*

```
encode prgtype, gen(prog)
```

```
browse prog prgtype
```

	prgtype	prog	
1	general	general	
2	vocati	vocati	
3	general	general	
4	vocati	vocati	
5	academic	academic	

```
. tab prog
```

prog	Freq.	Percent	Cum.
academic	105	52.50	52.50
general	45	22.50	75.00
vocati	50	25.00	100.00
Total	200	100.00	

```
. tab prog, nolabel
```

prog	Freq.	Percent	Cum.
1	105	52.50	52.50
2	45	22.50	75.00
3	50	25.00	100.00
Total	200	100.00	

# Ejercicio aplicado

Usando la base de datos del ejercicio anterior (hs0):

- Usando los comandos **generate** y **replace** cree la variable “math\_high”, la cual es una dummy que toma el valor 1 si “math” está por encima de 60 y el valor 0 en otro caso.
- Etiquete esta nueva variable creando una etiqueta llamada “etiqueta\_math” que coloque la leyenda “Alto” al valor 1, y la leyenda “bajo” al valor 0.
- Aplique la “etiqueta\_math” a la variable “math\_high” y explore esta última variable usando **browse** y **tabulate**.

# Operaciones con bases de datos



# Operaciones con bases de datos

- Es recomendable guardar una copia de nuestra base de datos antes de hacerle grandes cambios.

## El comando keep:

- Como su nombre lo indica, conserva información
- Conservar variables: *keep var1 var2 ...*
- Conservar observaciones a través de condiciones: *keep if read>30 & math>20*

## El comando drop:

- Como su nombre lo indica, elimina información
- Eliminar variables: *drop var1 var2 ...*
- Eliminar observaciones a través de condiciones: *drop if math <10 | science==.*

# Operaciones con bases de datos

## Ordenando información con sort y gsort:

- El comando sort permite ordenar ascendentemente la información de la base de datos considerando una o más variables.
- Sintaxis: `sort var1 var2 ...`

Datos sin ordenar:

```
. li id read math in 1/5
```

	id	read	math
1.	70	57	41
2.	121	68	53
3.	86	44	54
4.	141	63	47
5.	172	47	57

Datos ordenados:

```
. sort id read math
```

```
. li id read math in 1/5
```

	id	read	math
1.	1	34	40
2.	2	39	33
3.	3	63	48
4.	4	44	41
5.	5	47	43

# Operaciones con bases de datos

## Ordenando información con sort y gsort:

- El comando **gsort** es más versátil, puesto que permite ordenar la información de manera ascendente o descendente (anteponiendo el signo - para este último caso).
- Sintaxis: `gsort (+/-) var1 var2 ..`

```
gsort -id -read -math
```

```
li id read math in 1/5
```

	id	read	math
1.	200	68	75
2.	199	52	50
3.	198	47	51
4.	197	50	50
5.	196	44	49

# Ejercicio aplicado

Usando la base de datos del ejercicio anterior (hs0):

- Vuelva a cargar la base de datos desde la web
- Obtenga una submuestra de la base de datos que considere solo aquellos estudiantes que tengan un puntaje de “write” mayor o igual a 60. Puede usar los comandos **keep** o **drop**
- Conserve solamente las variables “id” y “write”, y guarde esta nueva base de datos con el nombre “write\_alto”.
- Revise cuántas observaciones tiene esta base
- Vuelva a cargar la base de datos desde la web, pero en esta ocasión solo conserve las observaciones en las que el puntaje de “write” sea menor a 60.
- Conserve solo las variables “id” y “write” y guarde esta base con el nombre “write\_bajo”
- Revise cuántas observaciones tiene esta base
- Vuelva a cargar la base de datos desde la web, y elimine la variable “write”
- Guarde esta nueva base de datos con el nombre “sin\_write”.

# Combinando bases de datos

# Combinando bases de datos

## Apilando bases: el comando append:

- El comando append combina bases de datos apilando una encima de otra, obteniendo así una nueva base con más observaciones (filas).
- Se suele usar principalmente cuando tenemos bases de datos con las mismas variables, pero se encuentran en bases separadas por diversas razones como por ejemplo: alumnos distribuidos por secciones, rondas de encuestas por años, bases de datos independientes por región, etc.
- Sintaxis: *append using nombre\_base*
- Ejercicio rápido: abra la base “write\_alto” generada en el ejercicio anterior, y únala con la base “write\_bajo”. Contabilice el número de observaciones de la base final.

var1	var2	var3



var1	var2	var3

# Combinando bases de datos

## Uniando bases: el comando merge

- El comando merge combina bases de datos que contienen unidades en común, pero variables diferentes.
- Las bases solo pueden ser unidas si contienen una o más variables que identifican a las unidades en común (por ejemplo números de DNI, RUC, códigos de ciudades, nombres de países, etc.)
- El resultado de merge es una base de datos que tiene un mayor número de variables para un mismo conjunto de unidades
- En términos de Stata, la base cargada en la memoria es llamada “master”, mientras que la que va a ser unida se le llama “using”

*Base master*

id	var1	var2
111		
112		
113		



*Base using*

id	var3	var4	var5	var6
111				
112				
113				

# Combinando bases de datos

## Uniendo bases: el comando merge

- No obstante, existen diferentes escenarios para unir bases de datos usando merge, lo cual variará un poco la sintaxis a usar:

### Merge 1 a 1:

- Cuando cada observación de la base “master” tiene solo un empate en la base “using”
- Sintaxis: *merge 1:1 idvar using base\_using*

*Base master*

id	var1	var2
111		
112		
113		

*Base using*

id	var3	var4	var5	var6
111				
112				
113				

=

*Resultado*

id	var1	var2	var3	var4	var5	var6
111						
112						
113						



# Combinando bases de datos

## Uniendo bases: el comando `merge`

### Merge 1 a varios:

- Cuando cada observación de la base “master” tiene varios empates en la base “using”
- Por ejemplo cuando las características de un hogar se asocian a cada uno de sus miembros
- Sintaxis: `merge 1:m idvar using base_using`

Base master		Base using			Resultado			
hogar_id	electricidad	hogar_id	id	edad				
01	Sí	01	1	25				
01	Sí	01	2	30				
02	No	02	1	45				
02	No	02	2	50				
02	No	02	3	18				
		hogar_id	id	edad	hogar_id	id	electricidad	edad
01	Sí	01	1	25	01	1	Sí	25
01	Sí	01	2	30	01	2	Sí	30
02	No	02	1	45	02	1	No	45
02	No	02	2	50	02	2	No	50
02	No	02	3	18	02	3	No	18

# Combinando bases de datos

## Uniando bases: el comando `merge`

### Merge varios a 1:

- Cuando varias observaciones de la base “master” tiene solo un empate en la base “using”
- Por ejemplo cuando las características de varios individuos son asignadas a un solo hogar
- Sintaxis: `merge m:1 idvar using base_using`

*Base master*

hogar_id	id	edad
01	1	25
01	2	30
02	1	45
02	2	50
02	3	18

*Base using*

hogar_id	electricidad
01	Sí
02	No

**=**

*Resultado*

hogar_id	id	electricidad	edad
01	1	Sí	25
01	2	Sí	30
02	1	No	45
02	2	No	50
02	3	No	18

# Combinando bases de datos

## Uniendo bases: el comando merge

- Una vez aplicado el merge Stata nos indica el resultado del emparejamiento a través de la variable “\_merge” creada automáticamente, la cual toma los siguientes valores:
  - 1: indica que la observación solo se encontraba en la base “master”
  - 2: indica que la observación solo se encontraba en la base “using”
  - 3: indica que la observación se encontraba en ambas bases (hubo empate)

## Ejercicio rápido:

- Realiza el merge entre la base obtenida en el ejercicio previo de append, con la base “sin\_write” guardada previamente.
- ¿Cuántas variables tiene la base resultante?
- Revisa la distribución de la variable “\_merge” usando el comando tabulate