Getting Started in Data Analysis using Stata

(v. 6.1)

Oscar Torres-Reyna
otorres@princeton.edu

December 2007

http://www.princeton.edu/~otorres/

Stata Tutorial Topics

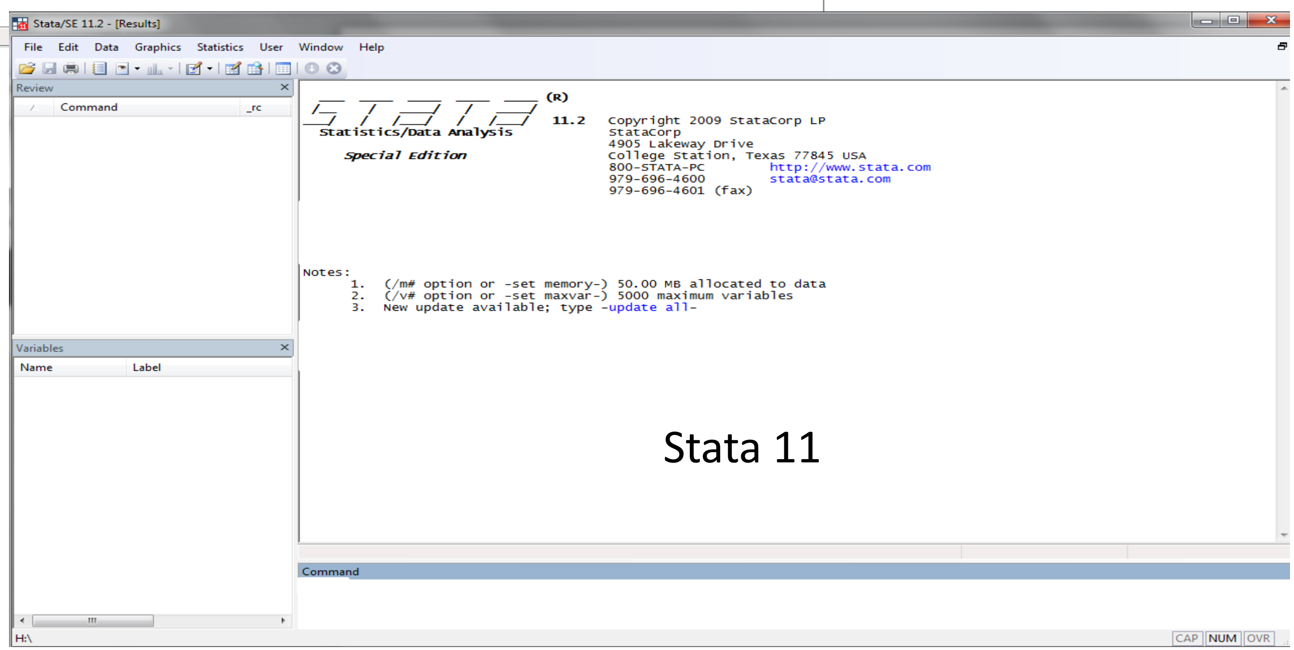
- ❑ What is Stata?
- ❑ Stata screen and general description
- ❑ First steps:
  - ✓ Setting the working directory (`pwd` and `cd` ….)
  - ✓ Log file (`log using` …)
  - ✓ Memory allocation (`set mem` …)
  - ✓ Do-files (`doedit`)
  - ✓ Opening/saving a Stata datafile
  - ✓ Quick way of finding variables
  - ✓ Subsetting (using conditional "if")
  - ✓ Stata color coding system
- ❑ From SPSS/SAS to Stata
- ❑ Example of a dataset in Excel
- ❑ From *Excel* to *Stata* (**copy-and-paste**, ***.csv**)
- ❑ **Describe** and **summarize**
- ❑ **Rename**
- ❑ **Variable labels**
- ❑ Adding **value labels**
- ❑ Creating new variables (**generate**)
- ❑ Creating new variables from other variables (**generate**)
- ❑ Recoding variables (**recode**)
- ❑ Recoding variables using **egen**
- ❑ Changing values (**replace**)
- ❑ Indexing (using _n and _N)
  - ✓ Creating ids and ids by categories
  - ✓ Lags and forward values
  - ✓ Countdown and specific values
- ❑ Sorting (ascending and descending order)
- ❑ Deleting variables (**drop**)
- ❑ Dropping cases (**drop if**)
- ❑ Extracting characters from regular expressions
- ❑ **Merge**
- ❑ **Append**
- ❑ Merging fuzzy text (**reclink**)
- ❑ Frequently used Stata commands
- ❑ **Exploring data:**
  - ✓ Frequencies (**tab, table**)
  - ✓ Crosstabulations (with test for associations)
  - ✓ Descriptive statistics (**tabstat**)
- ❑ Examples of frequencies and crosstabulations
- ❑ Three way crosstabs
- ❑ Three way crosstabs (with average of a fourth variable)
- ❑ Creating dummies
- ❑ Graphs
  - ✓ Scatterplot
  - ✓ Histograms
  - ✓ Catplot (for categorical data)
  - ✓ Bars (graphing mean values)
- ❑ **Data preparation/descriptive statistics**
- ❑ **Linear Regression**
- ❑ **Panel data (fixed/random effects)**
- ❑ **Multilevel Analysis**
- ❑ **Time Series**

What is Stata?

- It is a multi-purpose statistical package to help you explore, summarize and analyze datasets. It is widely used in social science research.
- A dataset is a collection of several pieces of information called variables (usually arranged by columns). A variable can have one or several values (information for one or several cases).

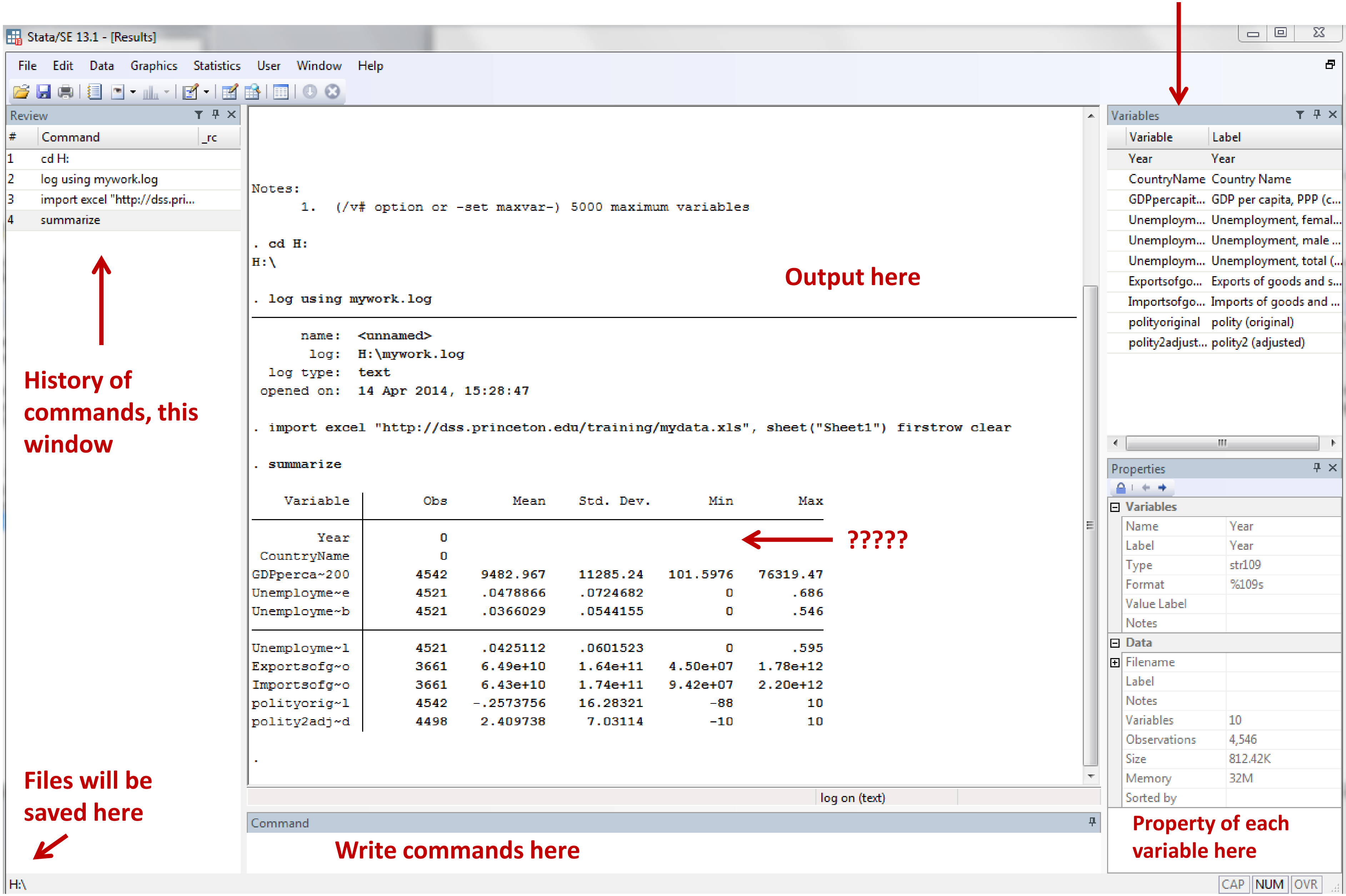
Features	SPSS	SAS	Stata	JMP (SAS)	R	Python (Pandas)
Learning curve	Gradual	Pretty steep	Gradual	Gradual	Pretty steep	Steep
User interface	Point-and-click	Programming	Programming/ point-and-click	Point-and-click	Programming	Programming
Data manipulation	Strong	Very strong	Strong	Strong	Very strong	Strong
Data analysis	Very strong	Very strong	Very strong	Strong	Very strong	Strong
Graphics	Good	Good	Very good	Very good	Excellent	Good
Cost	Expensive (perpetual, cost only with new version). Student disc.	Expensive (yearly renewal) Free student version, 2014	Affordable (perpetual, cost only with new version). Student disc.	Expensive (yearly renewal) Student disc.	Open source (free)	Open source (free)
Released	1968	1972	1985	1989	1995	2008

Stata's previous screens



Stata 12/13+ screen

Variables in dataset here



First steps: Working directory

To see your working directory, type

pwd

```
. pwd
h:\statadata
```

To change the working directory to avoid typing the whole path when calling or saving files, type:

cd c:\mydata

```
. cd c:\mydata
c:\mydata
```

Use quotes if the new directory has blank spaces, for example

cd “h:\stata and data”

```
. cd "h:\stata and data"
h:\stata and data
```

First steps: log file

Create a ***log file***, sort of Stata's built-in tape recorder and where you can: 1) retrieve the output of your work and 2) keep a record of your work.

In the command line type:

```
log using mylog.log
```

This will create the file '`mylog.log`' in your working directory. You can read it using any word processor (notepad, word, etc.).

To close a log file type:

```
log close
```

To add more output to an existing log file add the option `append`, type:

```
log using mylog.log, append
```

To replace a log file add the option `replace`, type:

```
log using mylog.log, replace
```

Note that the option `replace` will delete the contents of the previous version of the log.

First steps: memory allocation

Stata 12+ will automatically allocate the necessary memory to open a file. It is recommended to use Stata 64-bit for files bigger than 1 g.

If you get the error message "`no room to add more observations…`", (usually in older Stata versions, 11 or older) then you need to manually set the memory higher. You can type, for example

```
set mem 700m
```

Or something higher.

If the problem is in variable allocation (default is 5,000 variables), you increase it by typing, for example:

```
set maxvar 10000
```

To check the initial parameters type

```
query memory
```

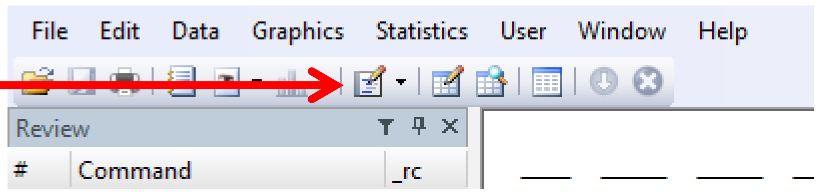
First steps: do-file

Do-files are ASCII files that contain of Stata commands to run specific procedures. It is highly recommended to use do-files to store your commands so do you not have to type them again should you need to re-do your work.

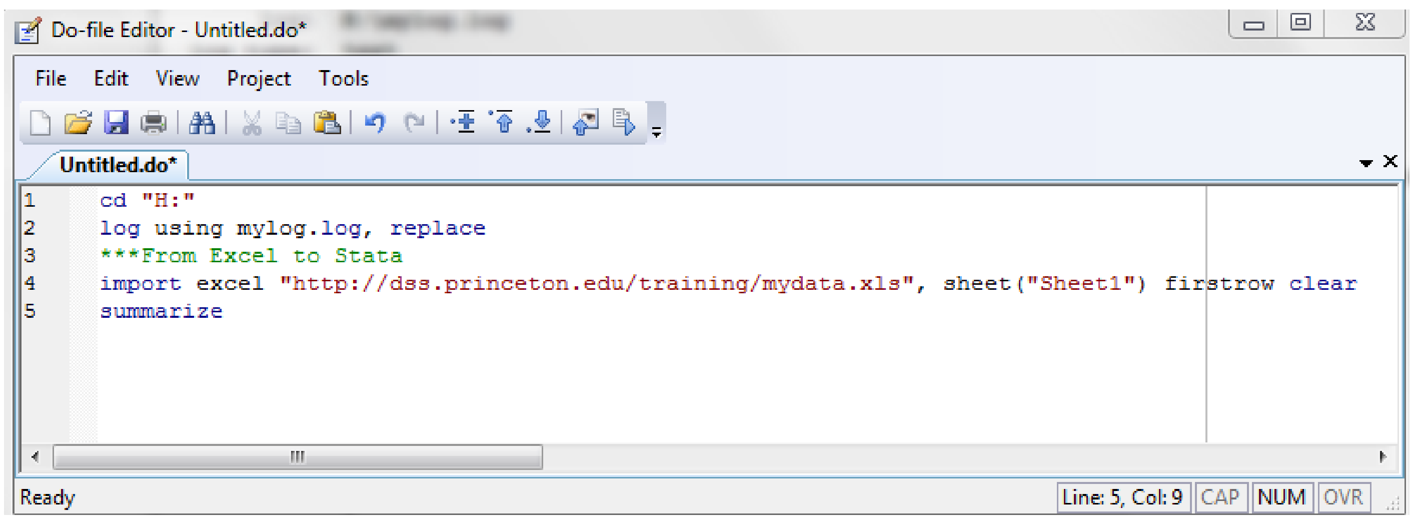
You can use any word processor and save the file in ASCII format, or you can use Stata's 'do-file editor' with the advantage that you can run the commands from there. Either , in the command window type:

```
doedit
```

Or, click on the icon here:



You can write the commands, to run them select the line(s), and click on the last icon in the do-file window



Check the following site for more info on do-files: http://www.princeton.edu/~otorres/Stata/

First steps: Opening/saving Stata files (*.dta)

To open files already in Stata with extension *.dta, run Stata and you can either:

- Go to file->open in the menu, or
- Type use *"c:\mydata\mydatafile.dta"*

If your working directory is already set to c:\mydata, just type

use *mydatafile*

To save a data file from Stata go to file – save as or just type:

save, replace

If the dataset is new or just imported from other format go to file –> save as or just type:

save mydatafile /*Pick a name for your file*/

For ASCII data please see https://www.princeton.edu/~otorres/DataPrep101.pdf

PU/DSS/OTR

First steps: Quick way of finding variables (`lookfor`)

You can use the command lookfor to find variables in a dataset, for example you want to see which variables refer to education, type:

lookfor educ

```
. lookfor educ

                  storage  display     value
variable name    type   format      label      variable label
-------------------------------------------------------------------------------
educ             byte   %10.0g                 Education of R.
```

`lookfor` will look for the keyword ‘educ’ in the variable name and labels. You will need to be creative with your keyword searches to find the variables you need.

It always recommended to use the codebook that comes with the dataset to have a better idea of where things are.

First steps: Subsetting using conditional ‘if’

Sometimes you may want to get frequencies, crosstabs or run a model just for a particular group (lets say just for females or people younger than certain age). You can do this by using the conditional ‘if’, for example:

```
/*Frequencies of var1 when gender = 1*/
tab var1 if gender==1, column row

/*Frequencies of var1 when gender = 1 and age < 33*/
tab var1 if gender==1 & age<33, column row

/*Frequencies of var1 when gender = 1 and marital status = single*/
tab var1 if gender==1 & marital==2 | marital==3 | marital==4, column row

/*You can do the same with crosstabs: tab var1 var2 … */

/*Regression when gender = 1 and age < 33*/
regress y x1 x2 if gender==1 & age<33, robust

/*Scatterplots when gender = 1 and age < 33*/
scater var1 var2 if gender==1 & age<33
```

“`if`” goes at the end of the command BUT before the comma that separates the options from the command.

First steps: Stata color-coded system

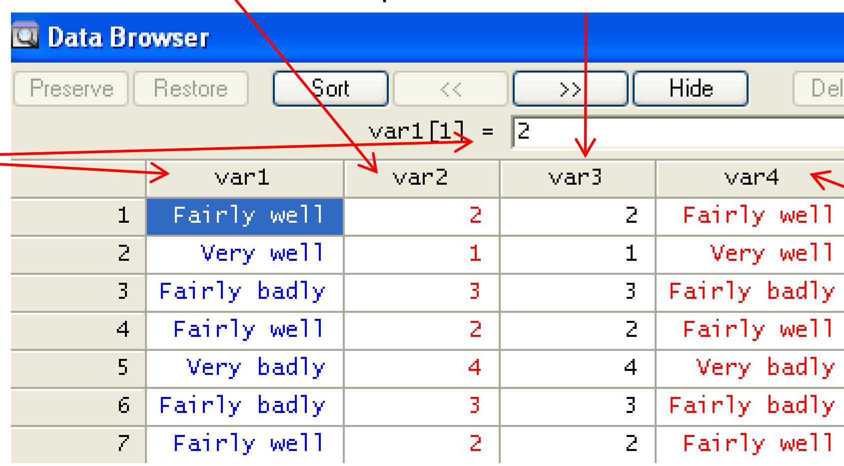
An important step is to make sure variables are in their expected format.

Stata has a color-coded system for each type. Black is for numbers, red is for text or string and blue is for labeled variables.

Var2 is a string variable even though you see numbers. You can't do any statistical procedure with this variable other than simple frequencies

Var3 is a numeric You can do any statistical procedure with this variable

For var1 a value 2 has the label "Fairly well". It is still a numeric variable



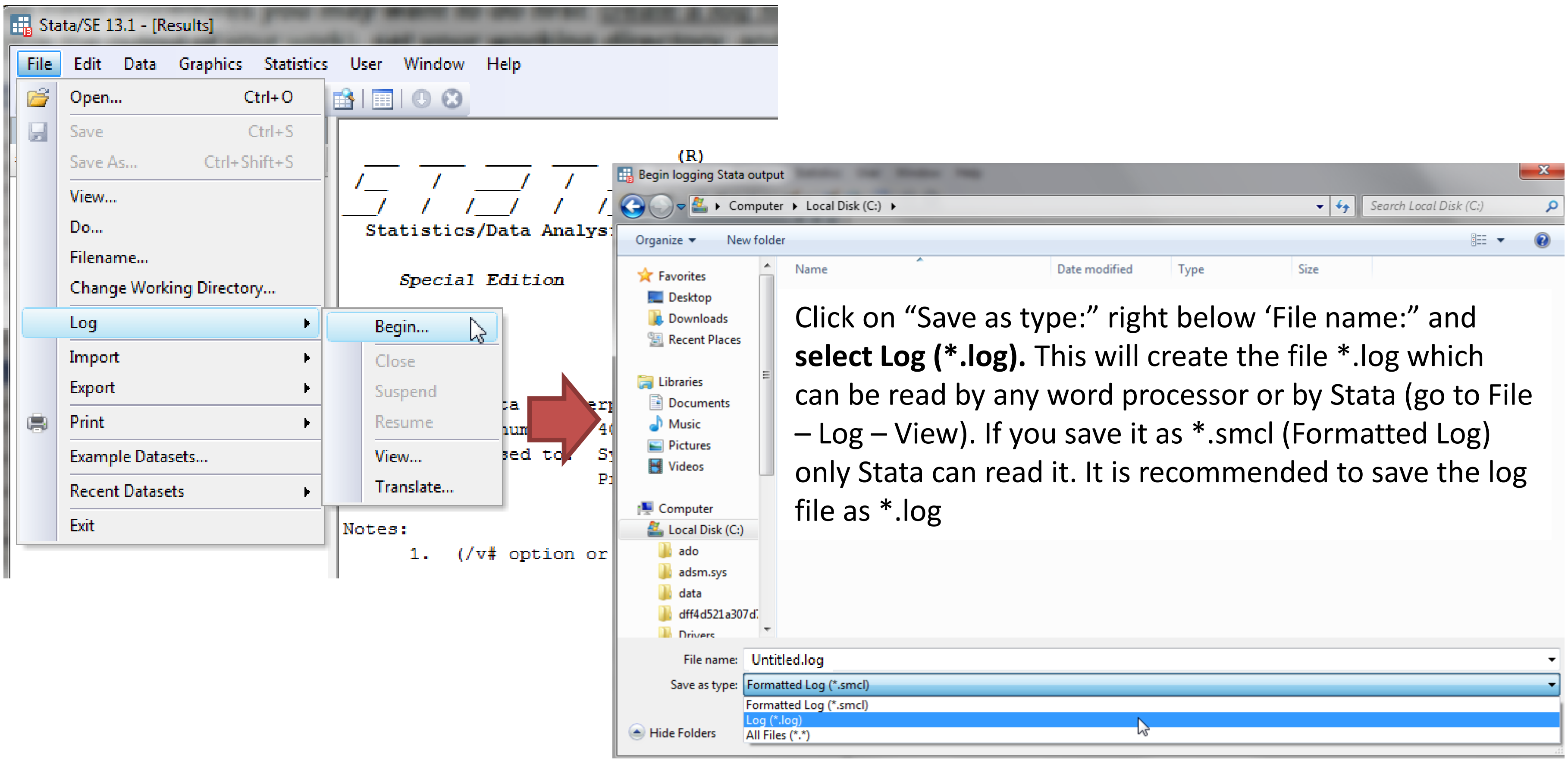
	var1	var2	var3	var4
1	Fairly well	2	2	Fairly well
2	Very well	1	1	Very well
3	Fairly badly	3	3	Fairly badly
4	Fairly well	2	2	Fairly well
5	Very badly	4	4	Very badly
6	Fairly badly	3	3	Fairly badly
7	Fairly well	2	2	Fairly well

Var4 is clearly a string variable. You can do frequencies and crosstabulations with this but not statistical procedures.

OTR

First steps: starting the log file using the menu

Log files help you to keep a record of your work, and lets you extract output. When using extension *.log any word processor can open the file.



Click on “Save as type:” right below ‘File name:” and **select Log (*.log).** This will create the file *.log which can be read by any word processor or by Stata (go to File – Log – View). If you save it as *.smcl (Formatted Log) only Stata can read it. It is recommended to save the log file as *.log

From SPSS/SAS to Stata

Stata 16+ can import SPSS and SAS data directly.

In the menu go to File --> Import

Example of a dataset in Excel.

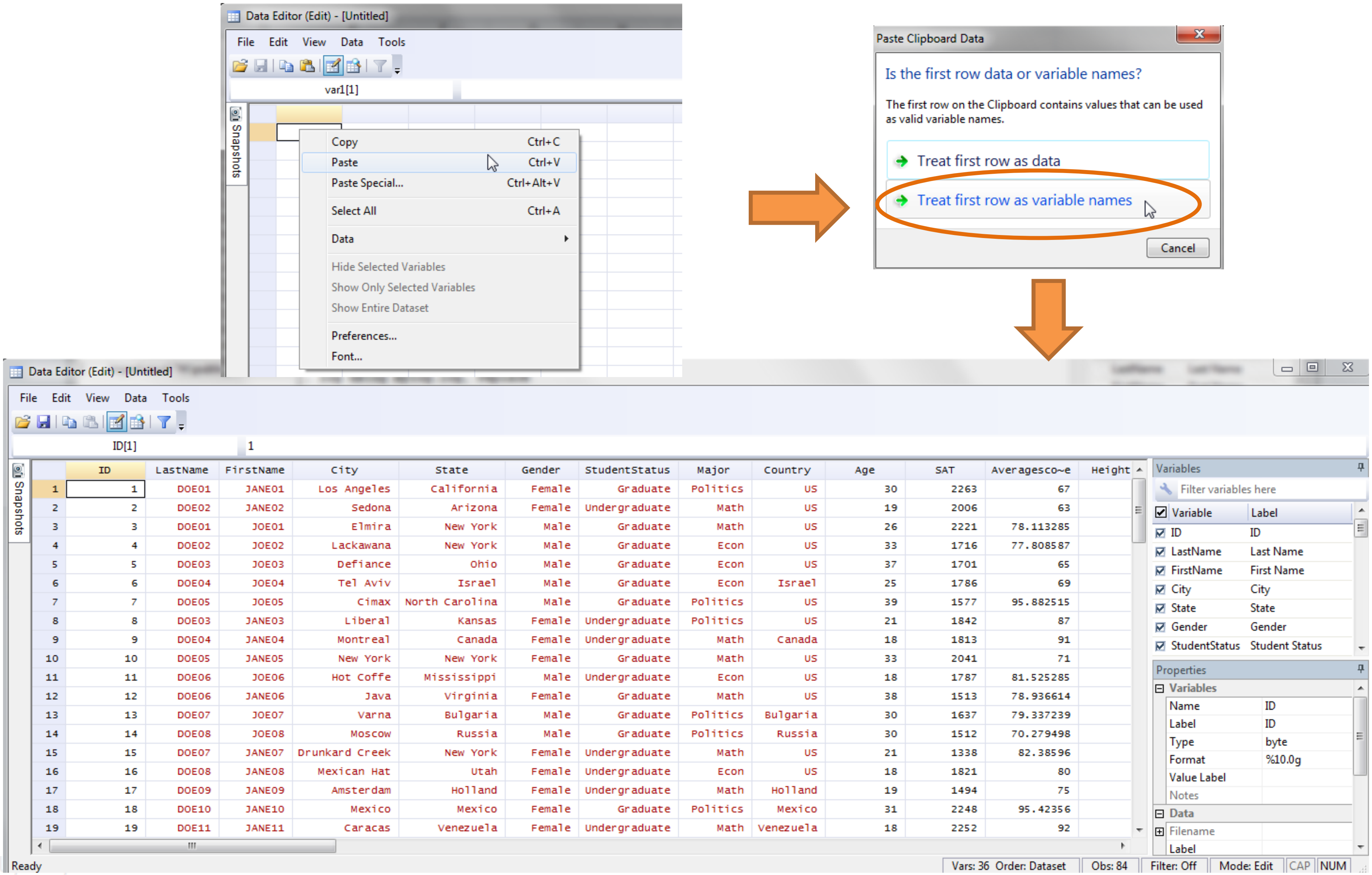
Variables are arranged by columns and cases by rows. Each variable has more than one value

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	Last Name	First Name	City	State	Gender	Student Status	Major	Country	Age	SAT	Average score (grade)	Height (in)	Newspaper readership (times/wk)
2	1	DOE01	JANE01	Los Angeles	California	Female	Graduate	Politics	US	30	2263	67	61	5
3	2	DOE02	JANE02	Sedona	Arizona	Female	Undergraduate	Math	US	19	2006	63	64	7
4	3	DOE16	JOE16	Elmira	New York	Male	Graduate	Math	US	26	2221	78	73	6
5	4	DOE17	JOE17	Lackawana	New York	Male	Graduate	Econ	US	33	1716	78	68	3
6	5	DOE18	JOE18	Defiance	Ohio	Male	Graduate	Econ	US	37	1701	65	71	6
7	6	DOE19	JOE19	Tel Aviv	Israel	Male	Graduate	Econ	Israel	25	1786	69	67	5
8	7	DOE20	JOE20	Cimax	North Carolina	Male	Graduate	Politics	US	39	1577	96	70	5
9	8	DOE03	JANE03	Liberal	Kansas	Female	Undergraduate	Politics	US	21	1842	87	62	5
10	9	DOE04	JANE04	Montreal	Canada	Female	Undergraduate	Math	Canada	18	1813	91	62	6
11	10	DOE05	JANE05	New York	New York	Female	Graduate	Math	US	33	2041	71	66	5
12	11	DOE21	JOE21	Hot Coffe	Mississippi	Male	Undergraduate	Econ	US	18	1787	82	67	3
13	12	DOE06	JANE06	Java	Virginia	Female	Graduate	Math	US	38	1513	79	59	5
14	13	DOE22	JOE22	Varna	Bulgaria	Male	Graduate	Politics	Bulgaria	30	1637	79	63	4
15	14	DOE23	JOE23	Moscow	Russia	Male	Graduate	Politics	Russia	30	1512	70	75	6
16	15	DOE07	JANE07	Drunkard Creek	New York	Female	Undergraduate	Math	US	21	1338	82	64	5
17	16	DOE08	JANE08	Mexican Hat	Utah	Female	Undergraduate	Econ	US	18	1821	80	63	3
18	17	DOE09	JANE09	Amsterdam	Holland	Female	Undergraduate	Math	Holland	19	1494	75	60	3
19	18	DOE10	JANE10	Mexico	Mexico	Female	Graduate	Politics	Mexico	31	2248	95	59	4
20	19	DOE11	JANE11	Caracas	Venezuela	Female	Undergraduate	Math	Venezuela	18	2252	92	68	5
21	20	DOE24	JOE24	San Juan	Puerto Rico	Male	Graduate	Politics	US	33	1923	95	63	7
22	21	DOE12	JANE12	Remote	Oregon	Female	Undergraduate	Econ	US	19	1727	67	62	7
23	22	DOE25	JOE25	New York	New York	Male	Undergraduate	Econ	US	21	1872	82	73	4
24	23	DOE13	JANE13	The X	Massachusetts	Female	Graduate	Politics	US	25	1767	89	68	6
25	24	DOE14	JANE14	Beijing	China	Female	Undergraduate	Math	China	18	1643	79	65	6
26	25	DOE26	JOE26	Stockholm	Sweden	Male	Undergraduate	Politics	Sweden	19	1919	88	64	4
27	26	DOE27	JOE27	Embarrass	Minnesota	Male	Graduate	Econ	US	28	1434	96	71	4
28	27	DOE28	JOE28	Intercourse	Pennsylvania	Male	Undergraduate	Math	US	20	2119	88	71	5
29	28	DOE15	JANE15	Loco	Oklahoma	Female	Undergraduate	Econ	US	20	2309	64	68	6
30	29	DOE29	JOE29	Buenos Aires	Argentina	Male	Graduate	Politics	Argentina	30	2279	85	72	3
31	30	DOE30	JOE30	Acme	Louisiana	Male	Undergraduate	Econ	US	19	1907	79	74	3

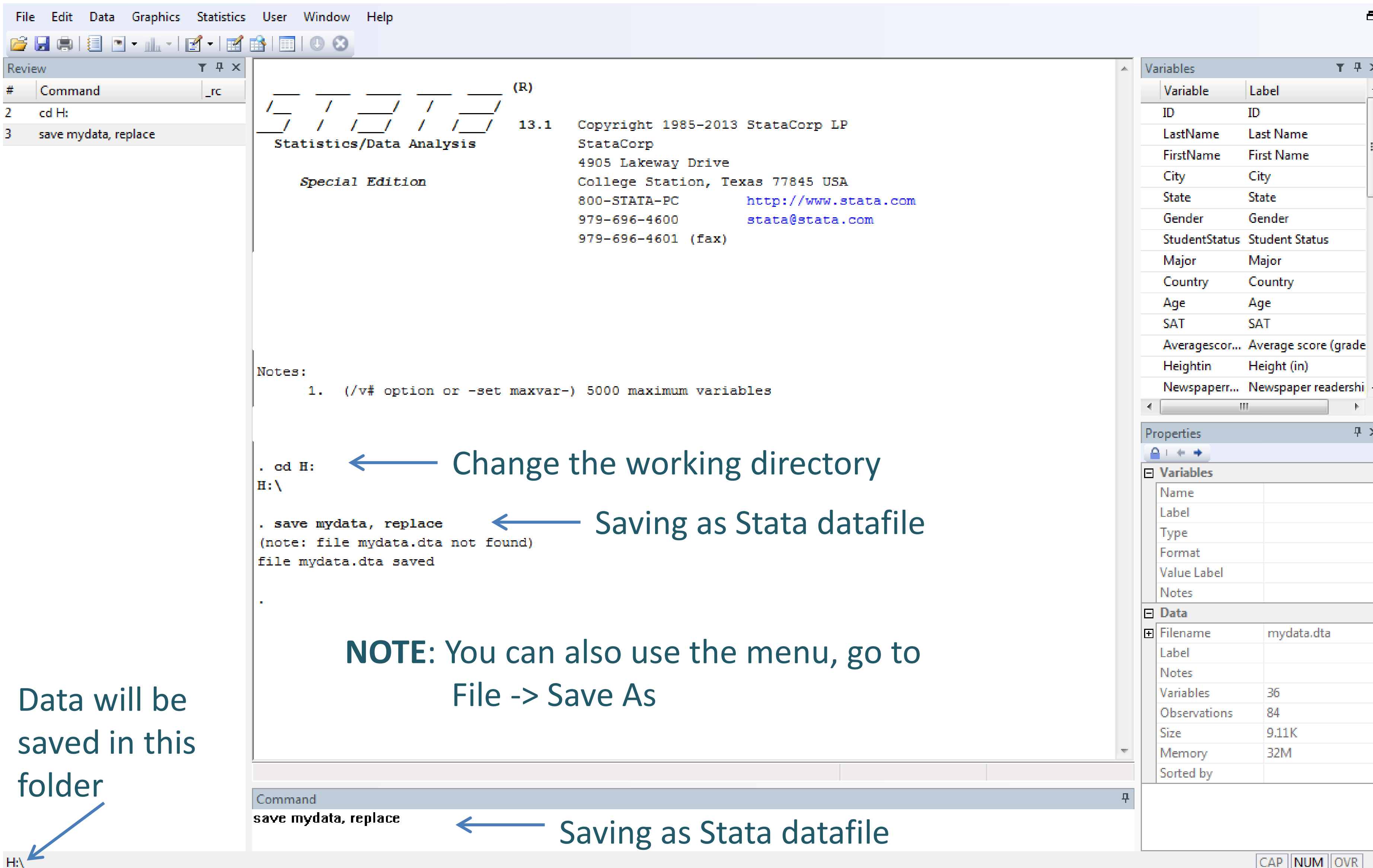
Path to the file: http://www.princeton.edu/~otorres/Stata/Students.xls

From Excel to Stata using copy-and-paste

In Excel, **select and copy** the data you want. Then, in Stata type `edit` in the command line to open the data editor. Point the cursor to the first cell, then right-click, select '`Paste`'.



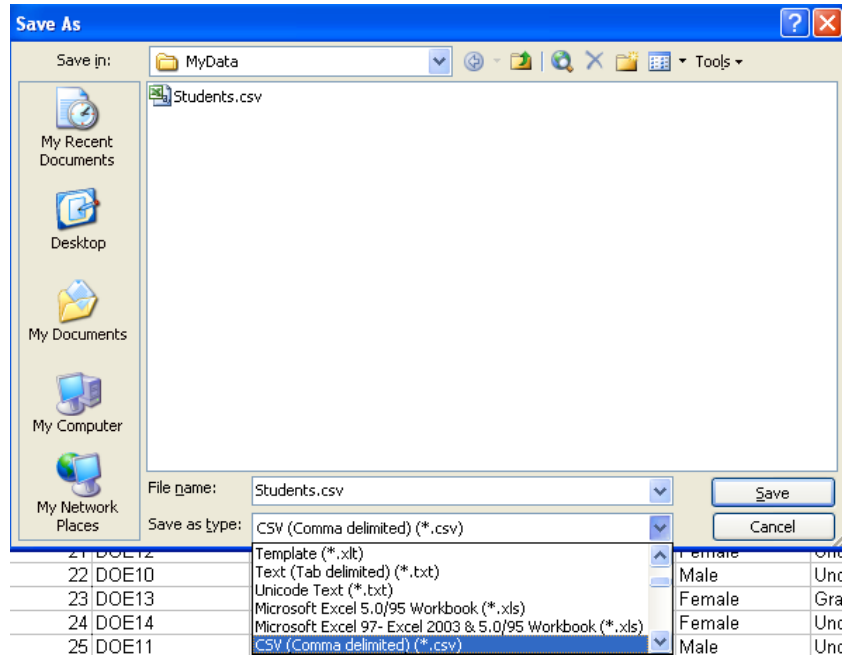
Saving data as Stata file



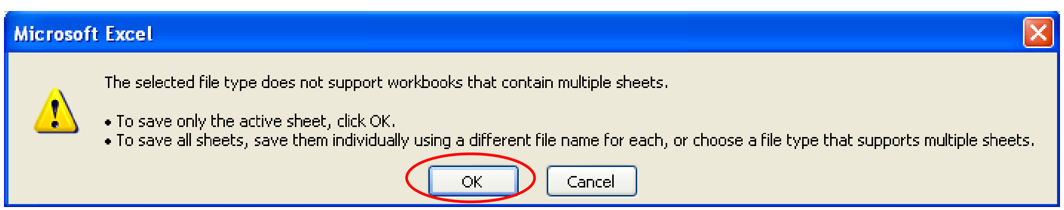
Excel to Stata (using `insheet`) step 1

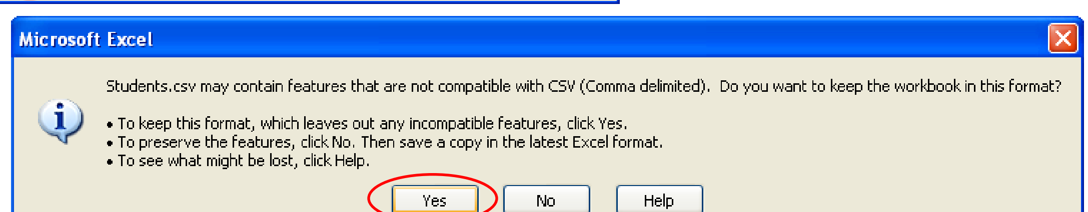
Another way to bring excel data into Stata is by saving the Excel file as ***.csv** (comma-separated values) and import it in Stata using the `insheet` command.

In **Excel** go to File->Save as and save the Excel file as *.csv:



You may get the following messages, click OK and YES…

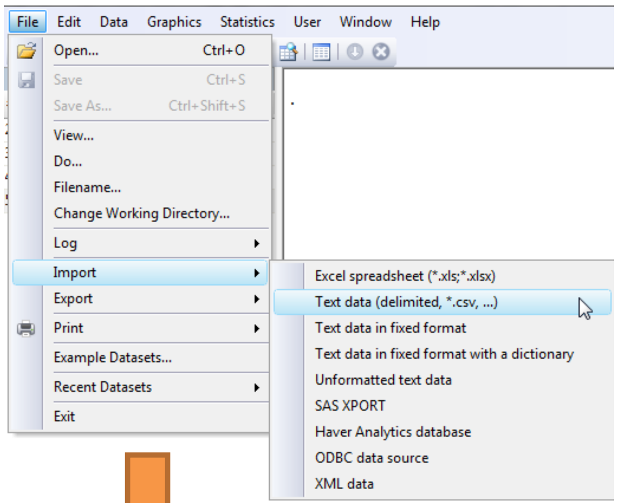


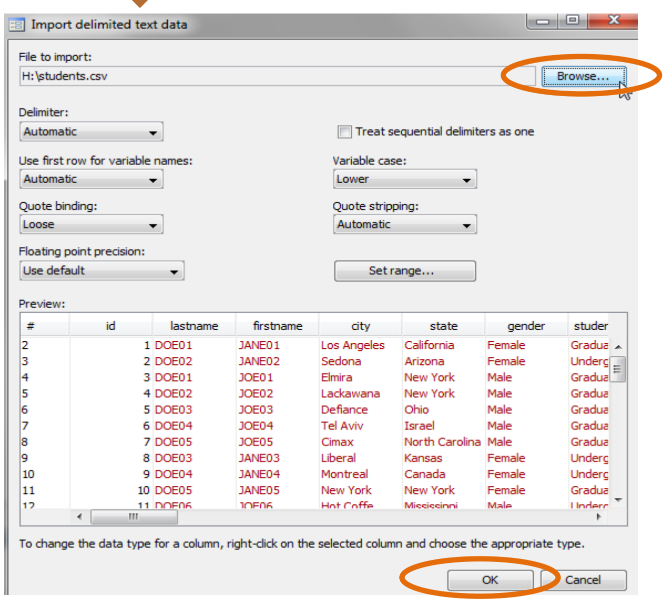


Go to the next page…

Excel to Stata (insheet using *.csv, - step 2)

From *.csv using the menu

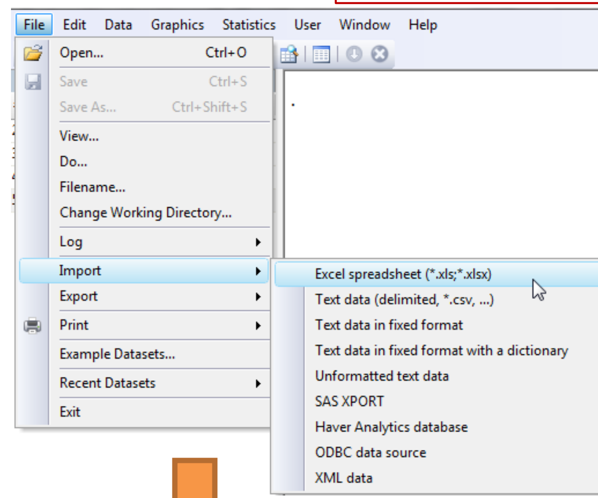


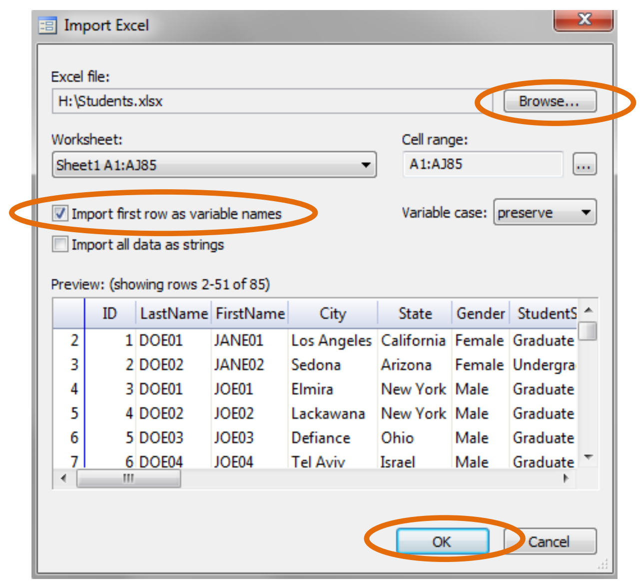


import delimited "H:\students.csv", clear

insheet using "H:\students.csv", clear

From *.xls(x) using the menu





import excel "H:\Students.xlsx", sheet("Sheet1") firstrow clear

To get a general description of the dataset and the format for each variable type
`describe`

```
. describe

Contains data from http://dss.princeton.edu/training/students.dta
  obs:            30
 vars:            14                          29 Sep 2009 17:12
 size:         2,580 (99.9% of memory free)
-----------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-----------------------------------------------------------------
id              byte   %8.0g                  ID
lastname        str5   %9s                    Last Name
firstname       str6   %9s                    First Name
city            str14  %14s                   City
state           str14  %14s                   State
gender          str6   %9s                    Gender
studentstatus   str13  %13s                   Student Status
major           str8   %9s                    Major
country         str9   %9s                    Country
age             byte   %8.0g                  Age
sat             int    %8.0g                  SAT
averagescoreg~e byte   %8.0g                  Average score (grade)
heightin        byte   %8.0g                  Height (in)
newspaperread~k byte   %8.0g                  Newspaper readership
-----------------------------------------------------------------
```

Type `help describe` for more information…

Command: summarize

Type summarize to get some basic descriptive statistics.

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
id	30	15.5	8.803408	1	30
lastname	0				
firstname	0				
city	0				
state	0				
gender	0				
studentsta~s	0				
major	0				
country	0				
age	30	25.2	6.870226	18	39
sat	30	1848.9	275.1122	1338	2309
averagesco~e	30	80.36667	10.11139	63	96
heightin	30	66.43333	4.658573	59	75
newspaperr~k	30	4.866667	1.279368	3	7

Zeros indicate string variables

Use 'min' and 'max' values to check for a valid range in each variable. For example, 'age' should have the expected values ('don't know' or 'no answer' are usually coded as 99 or 999)

Type help summarize for more information…

Exploring data: frequencies

Frequency refers to the number of times a value is repeated. Frequencies are used to analyze categorical data. The tables below are *frequency tables*, values are in ascending order. In Stata use the command **`tab`** ***`varname`*****.**

variable

```
. tab major
```

Major	Freq.	Percent	Cum.
Econ	10	33.33	33.33
Math	10	33.33	66.67
Politics	10	33.33	100.00
Total	30	100.00	

'Freq.' provides a raw count of each value. In this case 10 students for each major.

'Percent' gives the relative frequency for each value. For example, 33.33% of the students in this group are econ majors.

'Cum.' is the cumulative frequency in ascending order of the values. For example, 66.67% of the students are econ or math majors.

variable

```
. tab readnews
```

Newspaper readership (times/wk)	Freq.	Percent	Cum.
3	6	20.00	20.00
4	5	16.67	36.67
5	9	30.00	66.67
6	7	23.33	90.00
7	3	10.00	100.00
Total	30	100.00	

'Freq.' Here 6 students read the newspaper 3 days a week, 9 students read it 5 days a week.

'Percent'. Those who read the newspaper 3 days a week represent 20% of the sample, 30% of the students in the sample read the newspaper 5 days a week.

'Cum.' 66.67% of the students read the newspaper 3 to 5 days a week.

Type `help tab` for more details.

Exploring data: frequencies and descriptive statistics (using `table`)

Command `table` produces frequencies and descriptive statistics per category. For more info and a list of all statistics type `help table`. Or see the link below

https://www.stata.com/manuals/rtable.pdf

Exploring data: crosstabs

Also known as *contingency tables*, crosstabs help you to analyze the relationship between two or more categorical variables. Below is a crosstab between the variable 'ecostatu' and 'gender'. We use the command **`tab`** ***`var1 var2`***

Options 'column', 'row' gives you the column and row percentages.

var1 var2

```
. tab ecostatu gender, column row
```

Key
frequency *row percentage* *column percentage*

Status of Nat'l Eco	Gender of Respondent Male	Female	Total
Very well	90 60.40 14.33	59 39.60 7.92	149 100.00 10.85
Fairly well	337 50.30 53.66	333 49.70 44.70	670 100.00 48.80
Fairly badly	139 39.94 22.13	209 60.06 28.05	348 100.00 25.35
Very badly	57 29.84 9.08	134 70.16 17.99	191 100.00 13.91
Not sure	2 16.67 0.32	10 83.33 1.34	12 100.00 0.87
Refused	3 100.00 0.48	0 0.00 0.00	3 100.00 0.22
Total	628 45.74 100.00	745 54.26 100.00	1,373 100.00 100.00

The first value in a cell tells you the number of observations for each xtab. In this case, 90 respondents are 'male' and said that the economy is doing 'very well', 59 are 'female' and believe the economy is doing 'very well'

The second value in a cell gives you row percentages for the first variable in the xtab. Out of those who think the economy is doing 'very well', 60.40% are males and 39.60% are females.

The third value in a cell gives you column percentages for the second variable in the xtab. Among males, 14.33% think the economy is doing 'very well' while 7.92% of females have the same opinion.

NOTE: You can use `tab1` for multiple frequencies or `tab2` to run all possible crosstabs combinations. Type help tab for further details.

Exploring data: crosstabs (a closer look)

You can use crosstabs to compare responses among categories in relation to aggregate responses. In the table below we can see how opinions for males and females diverge from the national average.

```
tab ecostatu gender, column row
```

Key
frequency *row percentage* *column percentage*

Status of Nat'l Eco	Gender of Respondent Male	Female	Total
Very well	90 60.40 14.33	59 39.60 7.92	149 100.00 10.85
Fairly well	337 50.30 53.66	333 49.70 44.70	670 100.00 48.80
Fairly badly	139 39.94 22.13	209 60.06 28.05	348 100.00 25.35
Very badly	57 29.84 9.08	134 70.16 17.99	191 100.00 13.91
Not sure	2 16.67 0.32	10 83.33 1.34	12 100.00 0.87
Refused	3 100.00 0.48	0 0.00 0.00	3 100.00 0.22
Total	628 45.74 100.00	745 54.26 100.00	1,373 100.00 100.00

As a rule-of-thumb, a margin of error of ±4 percentage points can be used to indicate a significant difference (some use ±3).

For example, rounding up the percentages, 11% (10.85) answer 'very well' at the national level. With the margin of error, this gives a range roughly between 7% and 15%, anything beyond this range could be considered significantly different (remember this is just an approximation). It does not appear to be a significant bias between males and females for this answer.

In the 'fairly well' category we have 49%, with range between 45% and 53%. The response for males is 54% and for females 45%. We could say here that males tend to be a bit more optimistic on the economy and females tend to be a bit less optimistic.

If we aggregate responses, we could get a better picture. In the table below 68% of males believe the economy is doing well (comparing to 60% at the national level, while 46% of females thing the economy is bad (comparing to 39% aggregate). Males seem to be more optimistic than females.

RECODE of ecostatu (Status of Nat'l Eco)	Gender of Respondent Male	Female	Total
Well	427 52.14 67.99	392 47.86 52.62	819 100.00 59.65
Bad	196 36.36 31.21	343 63.64 46.04	539 100.00 39.26
Not sure/ref	5 33.33 0.80	10 66.67 1.34	15 100.00 1.09
Total	628 45.74 100.00	745 54.26 100.00	1,373 100.00 100.00

```
recode ecostatu (1 2 = 1 "Well") (3 4 = 2 "Bad") (5 6=3 "Not sure/ref"), gen(ecostatu1) label(eco)
```

Exploring data: crosstabs (test for associations)

To see whether there is a relationship between two variables you can choose a number of tests. Some apply to nominal variables some others to ordinal. I am running all of them here for presentation purposes.

```
tab ecostatu1 gender, column row nokey chi2 lrchi2 V exact gamma taub
```

Likelihood-ratio χ^2(chi-square)

Goodman & Kruskal's γ (gamma)

X^2(chi-square)

Cramer's *V*

Kendall's τ_b (tau-b)

Fisher's exact test

```
. tab ecostatu1 gender, column row nokey chi2 lrchi2 V exact gamma taub

Enumerating sample-space combinations:
stage 3:  enumerations = 1
stage 2:  enumerations = 16
stage 1:  enumerations = 0
```

RECODE of ecostatu (Status of Nat'l Eco)	Gender of Respondent Male	Female	Total
Well	427 52.14 67.99	392 47.86 52.62	819 100.00 59.65
Bad	196 36.36 31.21	343 63.64 46.04	539 100.00 39.26
Not sure/ref	5 33.33 0.80	10 66.67 1.34	15 100.00 1.09
Total	628 45.74 100.00	745 54.26 100.00	1,373 100.00 100.00

```
          Pearson chi2(2) =  33.5266   Pr = 0.000
 likelihood-ratio chi2(2) =  33.8162   Pr = 0.000
               Cramér's V =   0.1563
                    gamma =   0.3095  ASE = 0.050
          Kendall's tau-b =   0.1553  ASE = 0.026
           Fisher's exact =                 0.000
```

- For *nominal* data use chi2, lrchi2, V
- For *ordinal* data use gamma and taub
- Use `exact` instead of `chi2` when frequencies are less than 5 across the table.

X^2(chi-square) tests for relationships between variables. The null hypothesis (Ho) is that there is no relationship. To reject this we need a Pr < 0.05 (at 95% confidence). Here both chi2 are significant. Therefore we conclude that there is some relationship between perceptions of the economy and gender. `lrchi2` reads the same way.

Cramer's *V* is a measure of association between two nominal variables. It goes from 0 to 1 where 1 indicates strong association (for rXc tables). In 2x2 tables, the range is -1 to 1. Here the V is 0.15, which shows a small association.

Gamma and taub are measures of association between two ordinal variables (both have to be in the same direction, i.e. negative to positive, low to high). Both go from -1 to 1. Negative shows inverse relationship, closer to 1 a strong relationship. Gamma is recommended when there are lots of ties in the data. Taub is recommended for square tables.

Fisher's exact test is used when there are very few cases in the cells (usually less than 5). It tests the relationship between two variables. The null is that variables are independent. Here we reject the null and conclude that there is some kind of relationship between variables

Exploring data: descriptive statistics

For continuous data use descriptive statistics. These statistics are a collection of measurements of: *location* and *variability*. Location tells you the central value the variable (the mean is the most common measure of this) . Variability refers to the spread of the data from the center value (i.e. variance, standard deviation). Statistics is basically the study of what causes such variability. We use the command `tabstat` to get these stats.

```
tabstat age sat score heightin readnews, s(mean median sd var count range min max)
```

```
. tabstat  age sat score heightin readnews, s(mean median sd var count range min max)
```

stats	age	sat	score	heightin	readnews
mean	25.2	1848.9	80.36667	66.43333	4.866667
p50	23	1817	79.5	66.5	5
sd	6.870226	275.1122	10.11139	4.658573	1.279368
variance	47.2	75686.71	102.2402	21.7023	1.636782
N	30	30	30	30	30
range	21	971	33	16	4
min	18	1338	63	59	3
max	39	2309	96	75	7

Type `help tabstat` for a complete list of descriptive statistics

- The *mean* is the sum of the observations divided by the total number of observations.
- The *median* (p50 in the table above) is the number in the middle . To get the median you have to order the data from lowest to highest. If the number of cases is odd the median is the single value, for an even number of cases the median is the average of the two numbers in the middle.
- The *standard deviation* is the squared root of the variance. Indicates how close the data is to the mean. Assuming a normal distribution, 68% of the values are within 1 sd from the mean, 95% within 2 sd and 99% within 3 sd
- The *variance* measures the dispersion of the data from the mean. It is the simple mean of the squared distance from the mean.
- *Count* (N in the table) refers to the number of observations per variable.
- *Range* is a measure of dispersion. It is the difference between the largest and smallest value, max – min.
- *Min* is the lowest value in the variable.
- *Max* is the largest value in the variable.

Exploring data: descriptive statistics

You could also estimate descriptive statistics by subgroups (i.e. gender, age, etc.)

```
tabstat age sat score heightin readnews, s(mean median sd var count range min max) by(gender)
```

```
. tabstat  age sat score heightin readnews, s(mean median sd var count range min max) by(gender)

Summary statistics: mean, p50, sd, variance, N, range, min, max
  by categories of: gender (Gender)
```

gender	age	sat	score	heightin	readnews
Female	23.2	1871.8	78.73333	63.4	5.2
	20	1821	79	63	5
	6.581359	307.587	10.66012	3.112188	1.207122
	43.31429	94609.74	113.6381	9.685714	1.457143
	15	15	15	15	15
	20	971	32	9	4
	18	1338	63	59	3
	38	2309	95	68	7
Male	27.2	1826	82	69.46667	4.533333
	28	1787	82	71	4
	6.773899	247.0752	9.613978	3.943651	1.302013
	45.88571	61046.14	92.42857	15.55238	1.695238
	15	15	15	15	15
	21	845	31	12	4
	18	1434	65	63	3
	39	2279	96	75	7
Total	25.2	1848.9	80.36667	66.43333	4.866667
	23	1817	79.5	66.5	5
	6.870226	275.1122	10.11139	4.658573	1.279368
	47.2	75686.71	102.2402	21.7023	1.636782
	30	30	30	30	30
	21	971	33	16	4
	18	1338	63	59	3
	39	2309	96	75	7

Type `help tabstat` for more options.

Examples of frequencies and crosstabulations

Frequencies (`tab` command)

```
. tab gender
```

Gender	Freq.	Percent	Cum.
Female	15	50.00	50.00
Male	15	50.00	100.00
Total	30	100.00	

In this sample we have 15 females and 15 males. Each represents 50% of the total cases.

Crosstabulations (`tab` with two variables)

```
. tab gender studentstatus, column row
```

Key
frequency
row percentage
column percentage

Gender	Student Status Graduate	Student Status Undergrad	Total
Female	5 33.33 33.33	10 66.67 66.67	15 100.00 50.00
Male	10 66.67 66.67	5 33.33 33.33	15 100.00 50.00
Total	15 50.00 100.00	15 50.00 100.00	30 100.00 100.00

```
. tab gender major, sum(sat)
```

Means, Standard Deviations and Frequencies of SAT

Gender	Major Econ	Major Math	Major Politics	Total
Female	1952.3333 312.43773 3	1762.5 317.99326 8	2030 262.25052 4	1871.8 307.58697 15
Male	1743.2857 155.6146 7	2170 72.124892 2	1807.8333 288.99994 6	1826 247.07518 15
Total	1806 219.16559 10	1844 329.76928 10	1896.7 287.20687 10	1848.9 275.11218 30

Average SAT scores by gender and major. Notice, 'sat' variable is a continuous variable. The first cell reads the average SAT score for a female whose major is econ is 1952.3333 with a standard deviation 312.43, there are only 3 females with a major in econ.

Three way crosstabs

bysort *var3*: tab *var1 var2*, colum row

bysort *studentstatus*: tab *gender major*, colum row

```
. bysort  studentstatus: tab gender major, column row

-> studentstatus = Graduate
```

Key
frequency *row percentage* *column percentage*

Gender	Major Econon	Math	Politics	Total
Female	0 0.00 0.00	2 40.00 66.67	3 60.00 37.50	5 100.00 33.33
Male	4 40.00 100.00	1 10.00 33.33	5 50.00 62.50	10 100.00 66.67
Total	4 26.67 100.00	3 20.00 100.00	8 53.33 100.00	15 100.00 100.00

```
-> studentstatus = Undergraduate
```

Key
frequency *row percentage* *column percentage*

Gender	Econ	Major Math	Politics	Total
Female	3 30.00 50.00	6 60.00 85.71	1 10.00 50.00	10 100.00 66.67
Male	3 60.00 50.00	1 20.00 14.29	1 20.00 50.00	5 100.00 33.33
Total	6 40.00 100.00	7 46.67 100.00	2 13.33 100.00	15 100.00 100.00

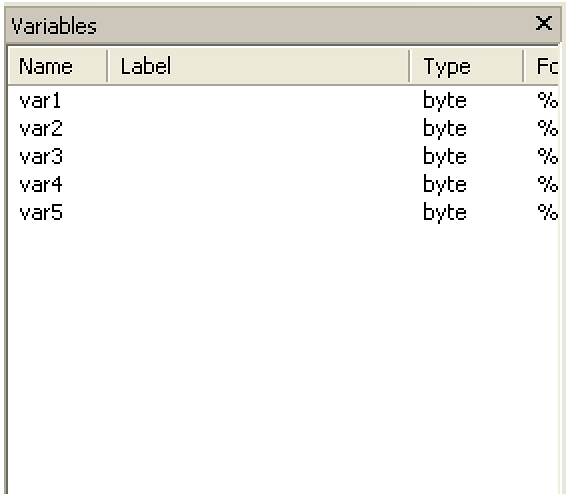
Three way crosstabs with summary statistics of a fourth variable

Average SAT scores by gender and major for graduate and undergraduate students. The third cell reads: The average SAT score of a female graduate student whose major is politics is 2092.6667 with a standard deviation of 2.82.13, there are 3 graduate female students with a major in politics.

```
. bysort  studentstatus: tab gender major, sum(sat)

-----------------------------------------------------------------
-> studentstatus = Graduate

            Means, Standard Deviations and Frequencies of SAT

           |              Major
    Gender |      Econ       Math   Politics |     Total
-----------+---------------------------------+----------
    Female |         .       1777  2092.6667 |    1966.4
           |         .  373.35238  282.13531 | 323.32924
           |         0          2          3 |         5
-----------+---------------------------------+----------
      Male |   1659.25       2221     1785.6 |    1778.6
           | 154.66819          0  317.32286 |  284.3086
           |         4          1          5 |        10
-----------+---------------------------------+----------
     Total |   1659.25       1925    1900.75 |    1841.2
           | 154.66819  367.97826   324.8669 | 300.38219
           |         4          3          8 |        15

-----------------------------------------------------------------
-> studentstatus = Undergraduate

            Means, Standard Deviations and Frequencies of SAT

           |              Major
    Gender |      Econ       Math   Politics |     Total
-----------+---------------------------------+----------
    Female | 1952.3333  1757.6667       1842 |    1824.5
           | 312.43773  337.01197          0 | 305.36872
           |         3          6          1 |        10
-----------+---------------------------------+----------
      Male | 1855.3333       2119       1919 |    1920.8
           | 61.711695          0          0 | 122.23011
           |         3          1          1 |         5
-----------+---------------------------------+----------
     Total | 1903.8333  1809.2857     1880.5 |    1856.6
           | 208.30979  336.59952  54.447222 | 257.72682
           |         6          7          2 |        15
```

Renaming variables, type:

Before



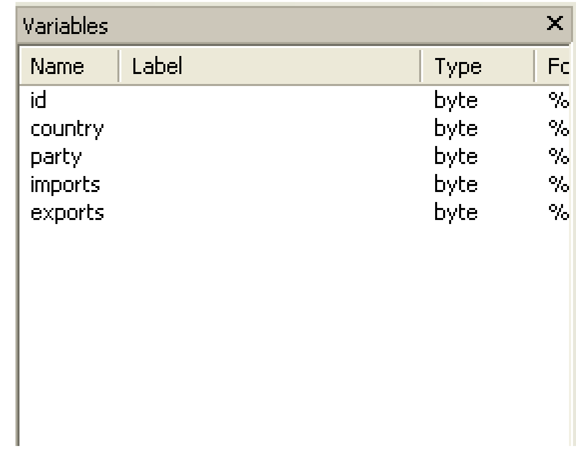
Variables

Name	Label	Type	Fc
var1		byte	%
var2		byte	%
var3		byte	%
var4		byte	%
var5		byte	%

rename [old name] [new name]

```
rename var1 id
rename var2 country
rename var3 party
rename var4 imports
rename var5 exports
```

After

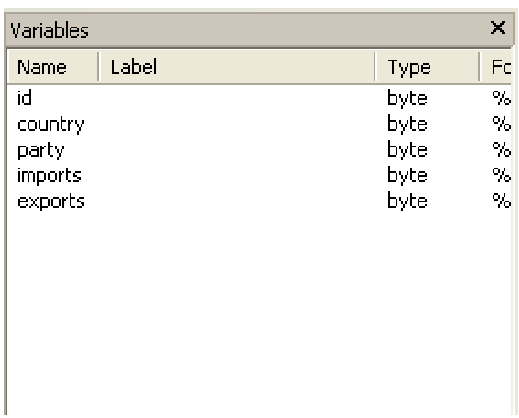


Variables

Name	Label	Type	Fc
id		byte	%
country		byte	%
party		byte	%
imports		byte	%
exports		byte	%

Adding/changing variable labels, type:

Before



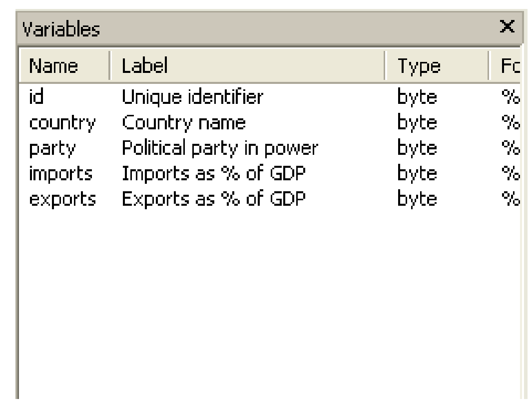
Variables

Name	Label	Type	Fc
id		byte	%
country		byte	%
party		byte	%
imports		byte	%
exports		byte	%

label variable [var name] “Text”

```
label variable id "Unique identifier"
label variable country "Country name"
label variable party "Political party in power"
label variable imports "Imports as % of GDP"
label variable exports "Exports as % of GDP"
```

After



Variables

Name	Label	Type	Fc
id	Unique identifier	byte	%
country	Country name	byte	%
party	Political party in power	byte	%
imports	Imports as % of GDP	byte	%
exports	Exports as % of GDP	byte	%

Assigning value labels

Adding labels to each category in a variable is a two step process in Stata.

Step 1: You need to create the labels using `label define`, type:

```
label define label1 1 “Agree” 2 “Disagree” 3 “Do not know”
```

Setp 2: Assign that label to a variable with those categories using `label values`:

```
label values var1 label1
```

If another variable has the same corresponding categories you can use the same label, type

```
label values var2 label1
```

Verify by running frequencies for var1 and var2 (using `tab`)

If you type `labelbook` it will list all the labels in the datafile.

NOTE: Defining labels is not the same as creating variables

To generate a new variable use the command `generate` (gen for short), type
`generate [newvar] = [expression]`

```
generate score2 = score/100
generate readnews2 = readnews*4
```

… results for the first five students…

score	height	readnews	score2	readnews2
67	61	5	.67	20
63	64	7	.63	28
78	73	6	.78	24
78	68	3	.78	12
65	71	6	.65	24

You can use `generate` to create constant variables. For example:

```
generate x = 5
generate y = 4*15
generate z = y/x
```

… results for the first five students…

x	y	z
5	60	12
5	60	12
5	60	12
5	60	12
5	60	12

You can also use `generate` with string variables. For example:

```
generate fullname = last + ", " + first
label variable fullname "Student full name"
browse id fullname last first
```

… results for the first five students…

id	fullname	last	first
1	DOE01, JANE01	DOE01	JANE01
2	DOE02, JANE02	DOE02	JANE02
3	DOE01, JOE01	DOE01	JOE01
4	DOE02, JOE02	DOE02	JOE02
5	DOE03, JOE03	DOE03	JOE03

To generate a new variable as a conditional from other variables type:

```
generate newvar=(var1==1 & var2==1)
generate newvar=(var1==1 & var2<26)
```

NOTE: & = and, | = or

```
. gen fem_grad=(gender==1 & status==1)

. tab fem_grad
```

fem_grad	Freq.	Percent	Cum.
0	25	83.33	83.33
1	5	16.67	100.00
Total	30	100.00	

```
. tab gender status
```

Gender	Student Status Graduate	Undergrad	Total
Female	5	10	15
Male	10	5	15
Total	15	15	30

```
. gen fem_less25=(gender==1 & age<26)

. tab fem_less25
```

fem_less25	Freq.	Percent	Cum.
0	19	63.33	63.33
1	11	36.67	100.00
Total	30	100.00	

```
. tab age gender
```

Age	Gender Female	Male	Total
18	4	1	5
19	3	2	5
20	1	1	2
21	2	1	3
25	1	1	2
26	0	1	1
28	0	1	1
30	1	3	4
31	1	0	1
33	1	2	3
37	0	1	1
38	1	0	1
39	0	1	1
Total	15	15	30

Recoding variables

1.- Recoding 'age' into three groups.

```
. tab age

        Age |      Freq.     Percent        Cum.
------------+-----------------------------------
         18 |          5       16.67       16.67
         19 |          5       16.67       33.33
         20 |          2        6.67       40.00
         21 |          3       10.00       50.00
         25 |          2        6.67       56.67
         26 |          1        3.33       60.00
         28 |          1        3.33       63.33
         30 |          4       13.33       76.67
         31 |          1        3.33       80.00
         33 |          3       10.00       90.00
         37 |          1        3.33       93.33
         38 |          1        3.33       96.67
         39 |          1        3.33      100.00
------------+-----------------------------------
      Total |         30      100.00
```

2.- Use `recode` command, type

Type `help recode` for more details

```
recode age (18 19 = 1 "18 to 19") ///
           (20/29 = 2 "20 to 29") ///
           (30/39 = 3 "30 to 39") (else=.), generate(agegroups) label(agegroups)
```

3.- The new variable is called '`agegroups`':

```
. tab agegroups

  RECODE of |
  age (Age) |      Freq.     Percent        Cum.
------------+-----------------------------------
   18 to 19 |         10       33.33       33.33
   20 to 29 |          9       30.00       63.33
   30 to 39 |         11       36.67      100.00
------------+-----------------------------------
      Total |         30      100.00
```

Recoding variables using egen

You can recode variables using the command `egen` and options `cut/group`.

```
egen newvariable = cut (oldvariable), at (break1, break2, break3, etc.)
```

Notice that the breaks show ranges. Below we type four breaks. The first starts at 18 and ends before 20, the second starts at 20 and ends before 30, the third starts at 30 and ends before 40.

```
. egen agegroups2=cut(age), at(18, 20, 30, 40)

. tab agegroups2
```

agegroups2	Freq.	Percent	Cum.
18	10	33.33	33.33
20	9	30.00	63.33
30	11	36.67	100.00
Total	30	100.00	

You could also use the option `group,` which specifies groups with equal frequency (you have to add value labels:

```
egen newvariable = cut (oldvariable), group(# of groups)
```

```
. egen agegroups3=cut(age), group(3)

. tab agegroups3
```

agegroups3	Freq.	Percent	Cum.
0	10	33.33	33.33
1	9	30.00	63.33
2	11	36.67	100.00
Total	30	100.00	

For more details and options type `help egen`

Changing variable values (using `replace`)

Before

```
. tab read

  Newspaper |
 readership |
 (times/wk) |      Freq.     Percent        Cum.
------------+-----------------------------------
          3 |          6       20.00       20.00
          4 |          5       16.67       36.67
          5 |          9       30.00       66.67
          6 |          7       23.33       90.00
          7 |          3       10.00      100.00
------------+-----------------------------------
      Total |         30      100.00
```

```
replace read = . if read>5
```

After

```
. tab read, missing

  Newspaper |
 readership |
 (times/wk) |      Freq.     Percent        Cum.
------------+-----------------------------------
          3 |          6       20.00       20.00
          4 |          5       16.67       36.67
          5 |          9       30.00       66.67
          . |         10       33.33      100.00
------------+-----------------------------------
      Total |         30      100.00
```

Before

```
. tab read

  Newspaper |
 readership |
 (times/wk) |      Freq.     Percent        Cum.
------------+-----------------------------------
          3 |          6       20.00       20.00
          4 |          5       16.67       36.67
          5 |          9       30.00       66.67
          6 |          7       23.33       90.00
          7 |          3       10.00      100.00
------------+-----------------------------------
      Total |         30      100.00
```

```
replace read = . if inc==7
```

After

```
. tab read, missing

  Newspaper |
 readership |
 (times/wk) |      Freq.     Percent        Cum.
------------+-----------------------------------
          3 |          6       20.00       20.00
          4 |          5       16.67       36.67
          5 |          9       30.00       66.67
          6 |          7       23.33       90.00
          . |          3       10.00      100.00
------------+-----------------------------------
      Total |         30      100.00
```

Before

```
. tab gender

     Gender |      Freq.     Percent        Cum.
------------+-----------------------------------
     Female |         15       50.00       50.00
       Male |         15       50.00      100.00
------------+-----------------------------------
      Total |         30      100.00
```

After

```
. tab gender

     Gender |      Freq.     Percent        Cum.
------------+-----------------------------------
          F |         15       50.00       50.00
          M |         15       50.00      100.00
------------+-----------------------------------
      Total |         30      100.00
```

```
replace gender = "F" if gender == "Female"
replace gender = "M" if gender == "Male"
```

You can also do:

```
replace var1=# if var2==#
```

Extracting characters from regular expressions

To remove strings from var1 use the following command

```
gen var2=regexr(var1,"[.\}\)\*a-zA-Z]+","")

destring var2, replace
```

```
. list var1 var2
```

	var1	var2
1.	123A33	12333
2.	2144F	2144
3.	2312A	2312
4.	3567754G	3567754
5.	35457S	35457
6.	34234N	34234
7.	234212*	234212
8.	23146}	23146
9.	31231)	31231
10.	AFN.345	345
11.	NYSE.12	12

To extract strings from a combination of strings and numbers

```
gen var2=regexr(var1,"[.0-9]+","")
```

```
. list var1 var2
```

	var1	var2
1.	AFM.123	AFM
2.	ADGT.2345	ADGT
3.	ACDET.1234564	ACDET
4.	CDFGEEGY.596544	CDFGEEGY
5.	ACGETYF.1235	ACGETYF

More info see: http://www.ats.ucla.edu/stat/stata/faq/regex.htm

Indexing: creating ids

Using **_n**, you can create a unique identifier for each case in your data, type

Check the results in the data editor, 'idall' is equal to 'id'

```
. generate idall = _n

. move idall id

. label variable idall "General student ID"
```

	idall	id
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

Using **_N** you can also create a variable with the total number of cases in your dataset:

Check the results in the data editor:

```
. generate total = _N

. move total idall

. label variable total "Number of students in the sample"
```

	total	idall	id
1	30	1	1
2	30	2	2
3	30	3	3
4	30	4	4
5	30	5	5

Indexing: creating ids by categories

Check the results in the data editor:

	major	idmajor
1	Econ	1
2	Econ	2
3	Econ	3
4	Econ	4
5	Econ	5
6	Econ	6
7	Econ	7
8	Econ	8
9	Econ	9
10	Econ	10
11	Math	1
12	Math	2
13	Math	3
14	Math	4
15	Math	5
16	Math	6
17	Math	7
18	Math	8
19	Math	9
20	Math	10
21	Politics	1
22	Politics	2
23	Politics	3
24	Politics	4
25	Politics	5
26	Politics	6
27	Politics	7
28	Politics	8
29	Politics	9
30	Politics	10

We can create ids by categories. For example by `major`.

```
. sort major

. by major: gen idmajor = _n

. browse major idmajor
```

First we have to `sort` the data by the variable on which we are basing the id (`major` in this case).

Then we use the command by to tell Stata that we are using `major` as the base variable (notice the colon).

Then we use `browse` to check the two variables.

----- You can create lagged values with _**n** .

```
gen lag1_year=year[_n-1]
gen lag2_year=year[_n-2]
```

	year	lag1_year	lag2_year
1	1980	.	.
2	1981	1980	.
3	1982	1981	1980
4	1983	1982	1981
5	1984	1983	1982
6	1985	1984	1983
7	1986	1985	1984

A more advance alternative to create lags uses the "L" operand within a time series setting (`tsset` command must be specified first):

```
tsset year
        time variable:  year, 1980 to 2009
                delta:  1 unit
```

```
gen l1_year=L1.year
gen l2_year=L2.year
```

	year	l1_year	l2_year
1	1980	.	.
2	1981	1980	.
3	1982	1981	1980
4	1983	1982	1981
5	1984	1983	1982
6	1985	1984	1983
7	1986	1985	1984

----- You can create forward values with _**n**:

```
gen for1_year=year[_n+1]
gen for2_year=year[_n+2]
```

	year	for1_year	for2_year
1	1980	1981	1982
2	1981	1982	1983
3	1982	1983	1984
4	1983	1984	1985
5	1984	1985	1986
6	1985	1986	1987
7	1986	1987	1988

You can also use the "F" operand (with `tsset`)

```
gen f1_year=F1.year
gen f2_year=F2.year
```

	year	f1_year	f2_year
1	1980	1981	1982
2	1981	1982	1983
3	1982	1983	1984
4	1983	1984	1985
5	1984	1985	1986
6	1985	1986	1987
7	1986	1987	1988

NOTE: Notice the square brackets

For times series see: https://www.princeton.edu/~otorres/TS101.pdf

Indexing: countdown and specific values

Combining _**n** and _**N** you can create a countdown variable.

```
. generate reverseid = id[_N - _n+1]

. order id reverseid
```

Check the results in the data editor:

	id	reverseid
1	1	30
2	2	29
3	3	28
4	4	27
5	5	26
6	6	25
7	7	24

You can create a variable based on one value of another variable. For example, create a variable with the highest SAT value in the sample.

```
. sort sat

. generate highestSAT = sat[_N]

. browse sat highestSAT
```

Check the results in the data editor:

	sat	highestSAT
1	1338	2309
2	1434	2309
3	1494	2309
4	1512	2309
5	1513	2309
25	2221	2309
26	2248	2309
27	2252	2309
28	2263	2309
29	2279	2309
30	2309	2309

NOTE: You could get the same result without sorting by using egen and the `max` function

```
. egen highestSAT1 = max(sat)
```

Sorting

Before

	last	first	city
1	DOE01	JANE01	Los Angeles
2	DOE02	JANE02	Sedona
3	DOE01	JOE01	Elmira
4	DOE02	JOE02	Lackawana
5	DOE03	JOE03	Defiance
6	DOE04	JOE04	Tel Aviv
7	DOE05	JOE05	Cimax

```
sort var1 var2 …
```

```
. sort city

. browse  last first city
```

After

	last	first	city
1	DOE15	JOE15	Acme
2	DOE09	JANE09	Amsterdam
3	DOE14	JANE14	Beijing
4	DOE14	JOE14	Buenos Aires
5	DOE11	JANE11	Caracas
6	DOE05	JOE05	Cimax
7	DOE03	JOE03	Defiance

`gsort` is another command to sort data. The difference between `gsort` and `sort` is that with `gsort` you can sort in ascending or descending order, while with `sort` you can sort only in ascending order. Use +/- to indicate whether you want to sort in ascending/descending order. Here are some examples:

```
. gsort -id

. browse  id last first city
```

	id	last	first	city
1	30	DOE15	JOE15	Acme
2	29	DOE14	JOE14	Buenos Aires
3	28	DOE15	JANE15	Loco
4	27	DOE13	JOE13	Intercourse
5	26	DOE12	JOE12	Embarrass
6	25	DOE11	JOE11	Stockholm
7	24	DOE14	JANE14	Beijing

```
. gsort +major -sat

. browse  id last first major sat
```

	id	last	first	major	sat
1	28	DOE15	JANE15	Econ	2309
2	30	DOE15	JOE15	Econ	1907
3	22	DOE10	JOE10	Econ	1872
4	16	DOE08	JANE08	Econ	1821
5	11	DOE06	JOE06	Econ	1787
6	6	DOE04	JOE04	Econ	1786
7	21	DOE12	JANE12	Econ	1727
8	4	DOE02	JOE02	Econ	1716
9	5	DOE03	JOE03	Econ	1701
10	26	DOE12	JOE12	Econ	1434
11	19	DOE11	JANE11	Math	2252
12	3	DOE01	JOE01	Math	2221
13	27	DOE13	JOE13	Math	2119
14	10	DOE05	JANE05	Math	2041
15	2	DOE02	JANE02	Math	2006
16	9	DOE04	JANE04	Math	1813
17	24	DOE14	JANE14	Math	1643
18	12	DOE06	JANE06	Math	1513
19	17	DOE09	JANE09	Math	1494
20	15	DOE07	JANE07	Math	1338
21	29	DOE14	JOE14	Politics	2279
22	1	DOE01	JANE01	Politics	2263
23	18	DOE10	JANE10	Politics	2248
24	20	DOE09	JOE09	Politics	1923
25	25	DOE11	JOE11	Politics	1919
26	8	DOE03	JANE03	Politics	1842
27	23	DOE13	JANE13	Politics	1767
28	13	DOE07	JOE07	Politics	1637
29	7	DOE05	JOE05	Politics	1577
30	14	DOE08	JOE08	Politics	1512

Deleting variables

Use `drop` to delete variables and `keep` to keep them

Before

Variables

Name	Label
id	Student ID
reverseid	
months	General student ID
for_months	
lag_months	
total	Number of students in the sample
fullname	Student full name
last	Student last name
first	Student first name
city	City
state	State
gender	Gender
status	Status: grad or undergad
major	Major
country	Country
age	Age
sat	SAT
score	Average score (grade)
height	Height (in)
readnews	Newspaper read / week
score2	Score in decimals
readnews2	Monthly readership
x	
y	
z	
agegroups	Age by groups
agegroups2	
agegroups3	
highestSAT	
highestSAT1	
idmajor	
lag_months1	
for_months1	

```
. drop  reverseid for_months lag_months x y z agegroups2 agegroups3
. drop highestSAT highestSAT1 idmajor lag_months1 for_months1
```

Or

```
. keep  id months total-readnews2 agegroups
```

After

Variables

Name	Label
id	Student ID
months	General student ID
total	Number of students in the sample
fullname	Student full name
last	Student last name
first	Student first name
city	City
state	State
gender	Gender
status	Status: grad or undergad
major	Major
country	Country
age	Age
sat	SAT
score	Average score (grade)
height	Height (in)
readnews	Newspaper read / week
score2	Score in decimals
readnews2	Monthly readership
agegroups	Age by groups

Notice the dash between ‘total’ and ‘readnews2’, you can use this format to indicate a list so you do not have to type in the name of all the variables

Deleting cases (selectively)

You can drop cases selectively using the conditional "`if`", for example

```
drop if var1==1  /*This will drop observations (rows)
                 where gender =1*/
drop if age>40 /*This will drop observation where
               age>40*/
```

Alternatively, you can keep options you want

```
keep if var1==1
keep if age<40
keep if country==7 | country==13
keep if state==“New York” | state==“New Jersey”
```

| = "or", & = "and"

For more details type `help keep` or `help drop`.

Merge/Append

Please check this document:

https://www.princeton.edu/~otorres/Merge101.pdf

Merging fuzzy text (reclink)

RECLINK - Matching fuzzy text. Reclink stands for ‘record linkage’. It is a program written by Michael Blasnik to merge imperfect string variables. For example

Data1	Data2
Princeton University	Princeton U

Reclink helps you to merge the two databases by using a matching algorithm for these types of variables. Since it is a user created program, you may need to install it by typing `ssc install reclink.` Once installed you can type `help reclink` for details

As in merge, the merging variables must have the same name: state, university, city, name, etc. Both the master and the using files should have an id variable identifying each observation.

Note: <u>the name of ids must be different</u>, for example id1 (id master) and id2 (id using). Sort both files by the matching (merging) variables. The basic sytax is:

```
reclink var1 var2 var3 … using myusingdata, gen(myscore) idm(id1) idu(id2)
```

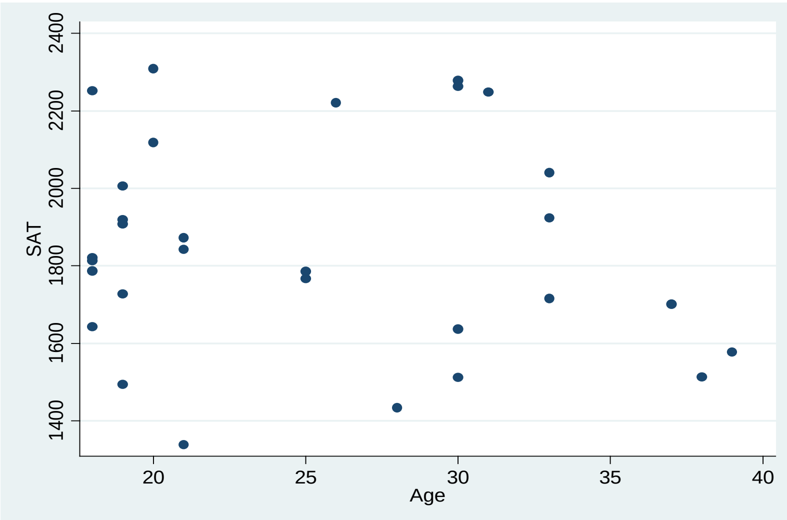
The variable `myscore` indicates the strength of the match; a perfect match will have a score of 1. Description (from reclink help pages):

```
“reclink uses record linkage methods to match observations between two datasets where no perfect key fields exist --
   essentially a fuzzy merge.  reclink allows for user-defined matching and non-matching weights for each variable and
   employs a bigram string comparator to assess imperfect string matches.

   The master and using datasets must each have a variable that uniquely identifies observations.  Two new variables are
   created, one to hold the matching score (scaled 0-1) and one for the merge variable.  In addition, all of the
   matching variables from the using dataset are brought into the master dataset (with newly prefixed names) to allow
   for manual review of matches.”
```

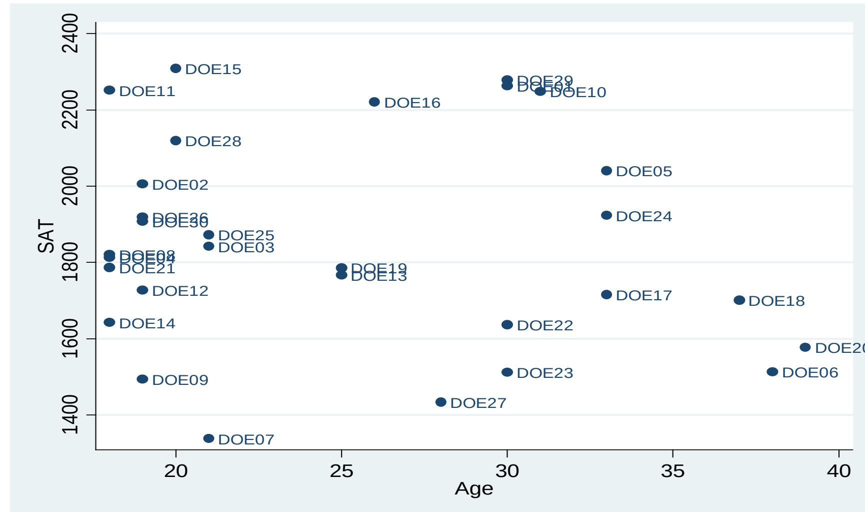
Graphs: scatterplot

Scatterplots are good to explore possible relationships or patterns between variables and to identify outliers. Use the command `scatter` (sometimes adding `twoway` is useful when adding more graphs). The format is `scatter y x`. Below we check the relationship between SAT scores and age. For more details type `help scatter` .

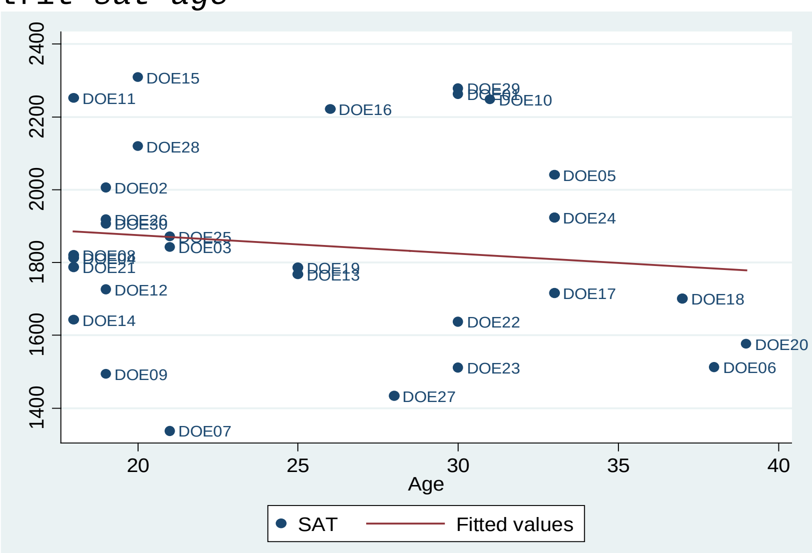
`twoway scatter` *`sat age`*



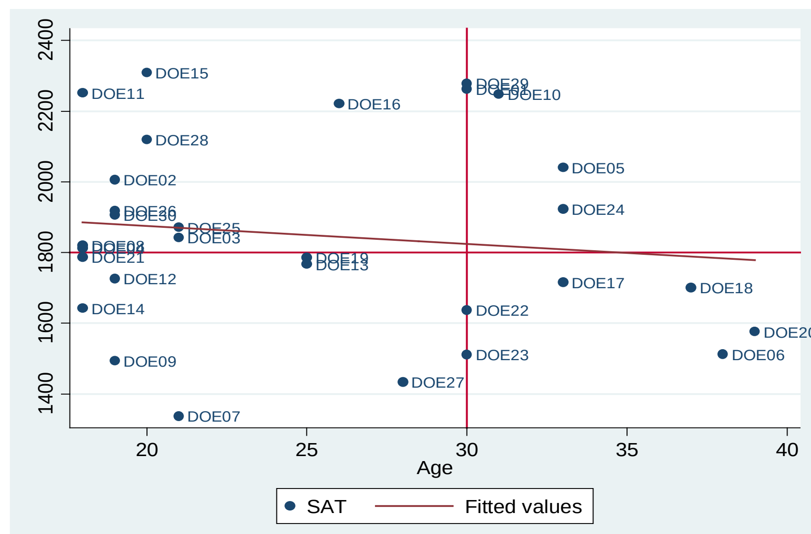
`twoway scatter` *`sat age`*`, mlabel(`*`last`*`)`



`twoway scatter` *`sat age`*`, mlabel(`*`last`*`) ||`
`lfit` *`sat age`*



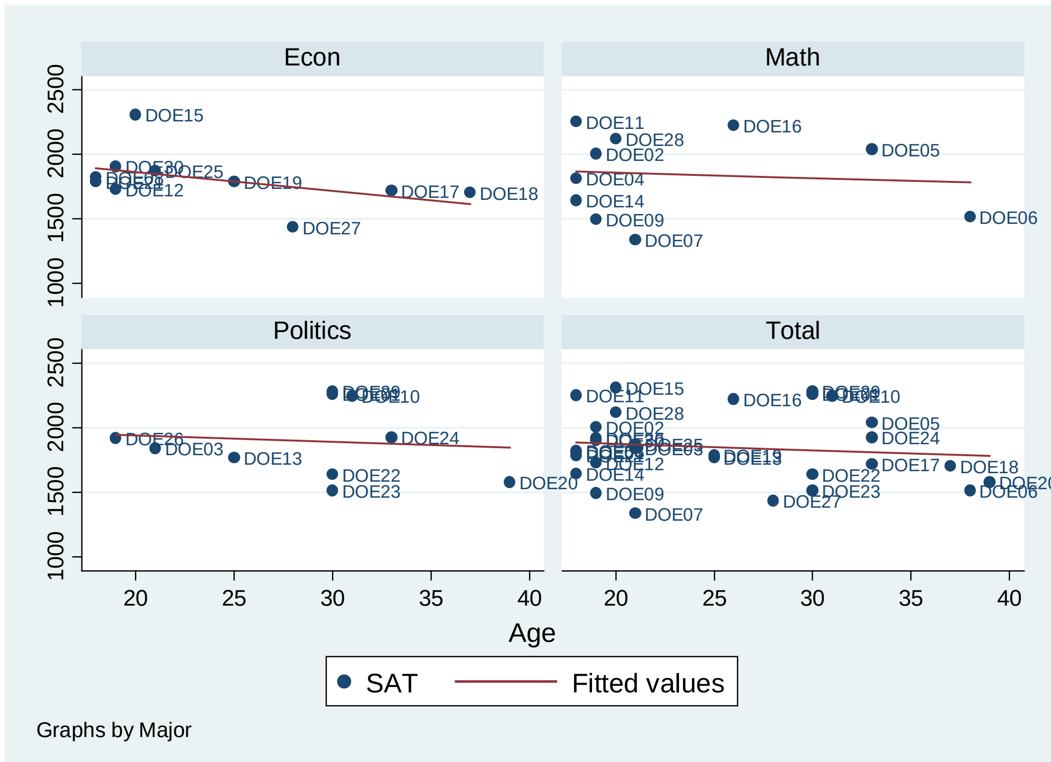
`twoway scatter` *`sat age`*`, mlabel(`*`last`*`) ||`
`lfit` *`sat age`*`, yline(30) xline(1800)`



Graphs: scatterplot

By categories

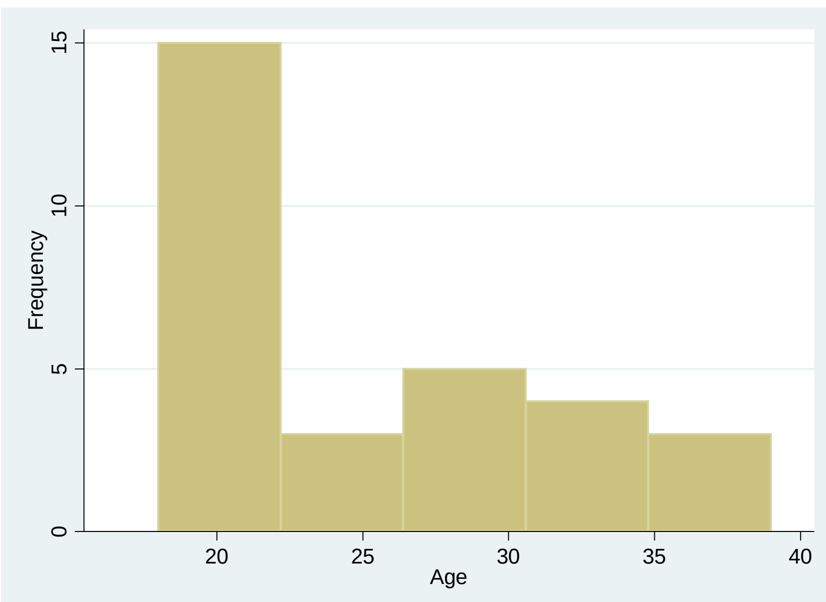
```
twoway scatter sat age, mlabel(last) by(major, total)
```



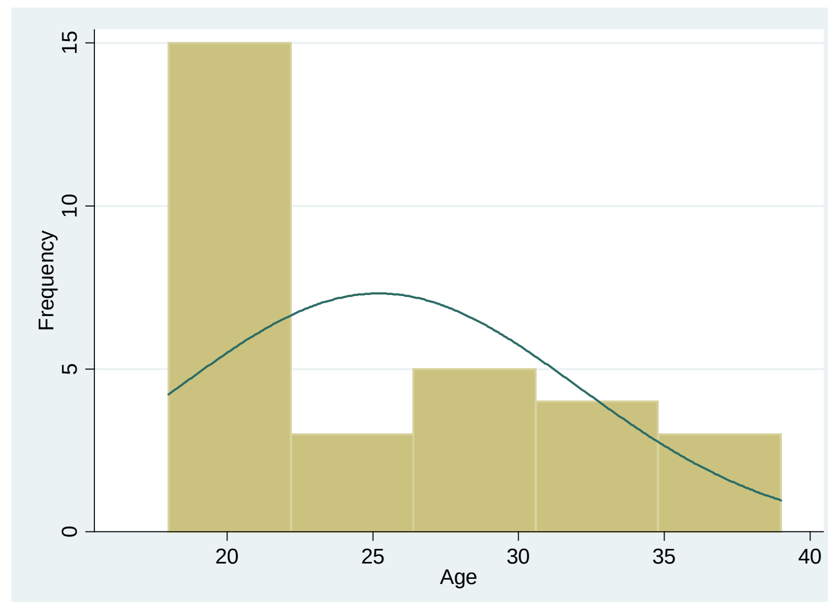
Go to http://www.princeton.edu/~otorres/Stata/ for additional tips

Histograms are another good way to visually explore data, especially to check for a normal distribution. Type `help histogram` for details.

`histogram` *`age`*`, frequency`



`histogram` *`age`*`, frequency normal`



Graphs: `catplot`

To graph categorical data use `catplot`. Since it is a user defined program you have to install it typing: `ssc install catplot`

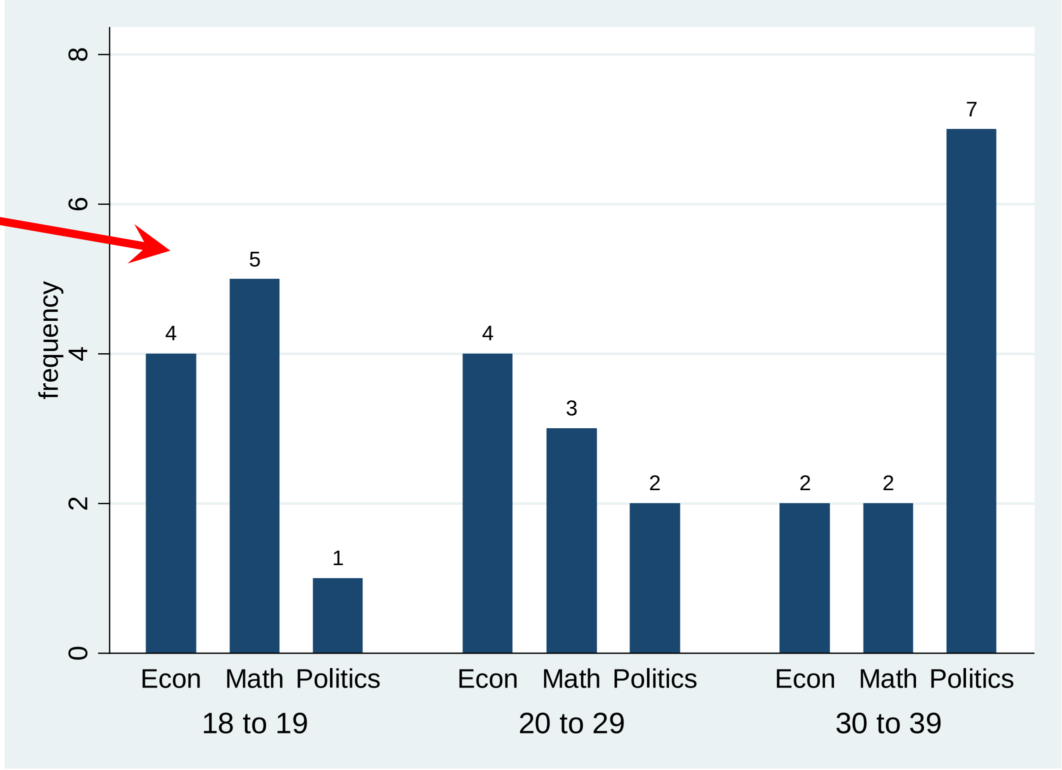
`tab` *`agegroups major`*`, col row cell`

`catplot bar` *`major agegroups`*`, blabel(bar)`

```
. tab agegroups major, col row cell
```

Key
frequency *row percentage* *column percentage* *cell percentage*

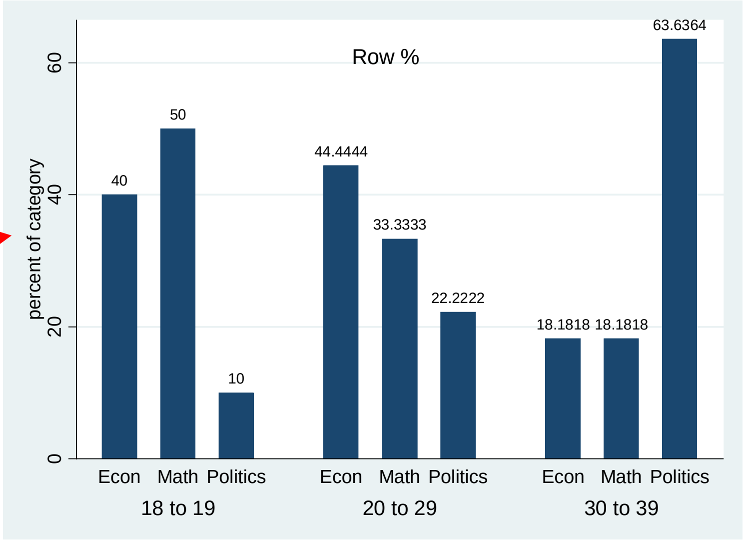
RECODE of age (Age)	Major Econ	Math	Politics	Total
18 to 19	4 40.00 40.00 13.33	5 50.00 50.00 16.67	1 10.00 10.00 3.33	10 100.00 33.33 33.33
20 to 29	4 44.44 40.00 13.33	3 33.33 30.00 10.00	2 22.22 20.00 6.67	9 100.00 30.00 30.00
30 to 39	2 18.18 20.00 6.67	2 18.18 20.00 6.67	7 63.64 70.00 23.33	11 100.00 36.67 36.67
Total	10 33.33 100.00 33.33	10 33.33 100.00 33.33	10 33.33 100.00 33.33	30 100.00 100.00 100.00



Note: Numbers correspond to the frequencies in the table.

Graphs: catplot

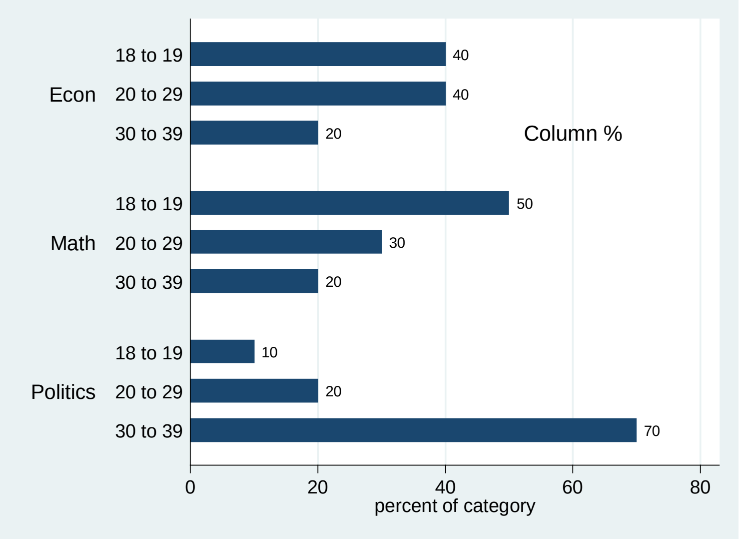
```
catplot  bar major agegroups, percent(agegroups) blabel(bar)
```




```
. tab  agegroups major, col row
```

Key
frequency   *row percentage*   *column percentage*

RECODE of age (Age)	Major Econ	Math	Politics	Total
18 to 19	4 40.00 40.00	5 50.00 50.00	1 10.00 10.00	10 100.00 33.33
20 to 29	4 44.44 40.00	3 33.33 30.00	2 22.22 20.00	9 100.00 30.00
30 to 39	2 18.18 20.00	2 18.18 20.00	7 63.64 70.00	11 100.00 36.67
Total	10 33.33 100.00	10 33.33 100.00	10 33.33 100.00	30 100.00 100.00




```
catplot  hbar agegroups major, percent(major) blabel(bar)
```

Graphs: catplot

```
catplot  hbar major agegroups, blabel(bar) by(gender)
```

```
. bysort gender: tab  agegroups major, col nokey
```

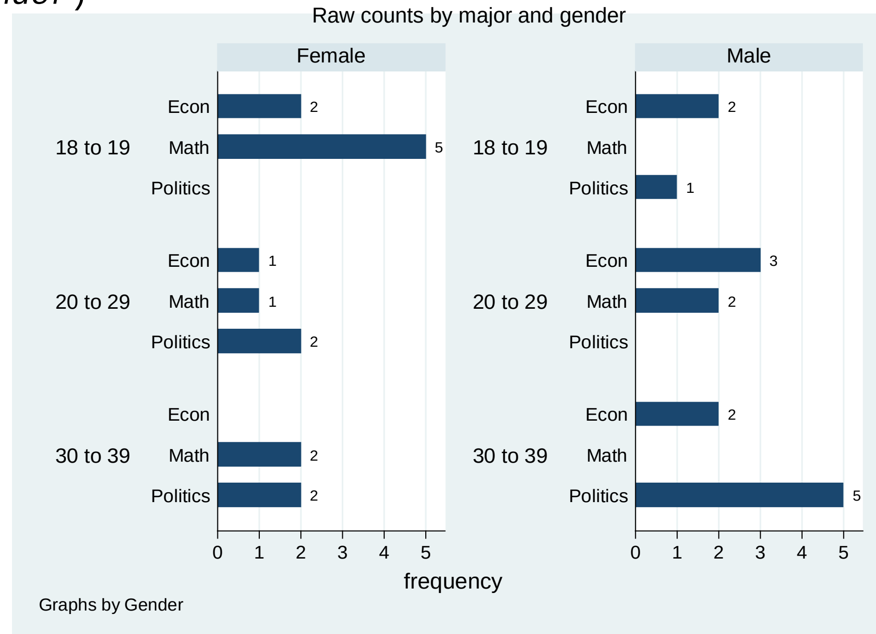
-> gender = Female

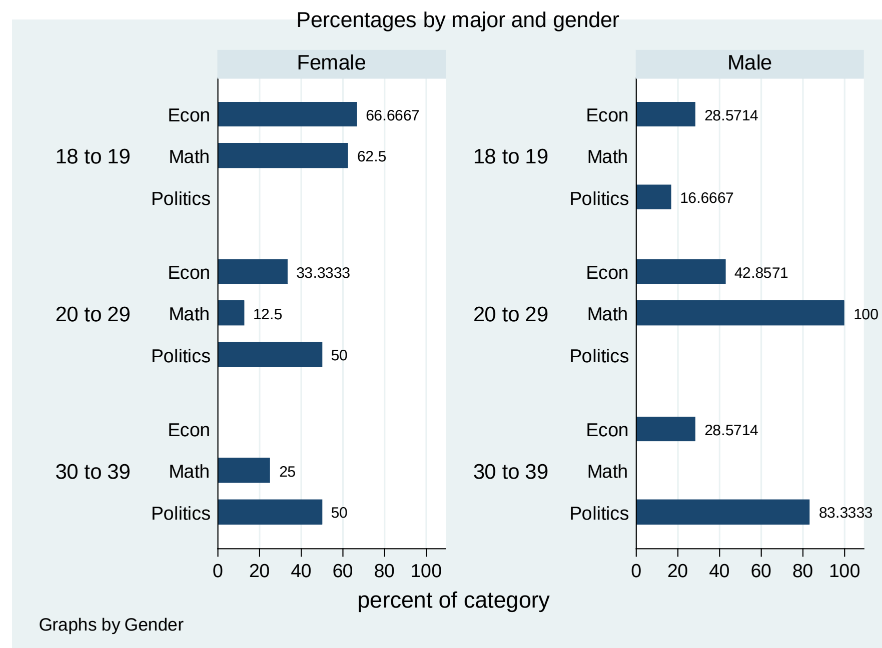
RECODE of age (Age)	Major Econ	Math	Politics	Total
18 to 19	2 66.67	5 62.50	0 0.00	7 46.67
20 to 29	1 33.33	1 12.50	2 50.00	4 26.67
30 to 39	0 0.00	2 25.00	2 50.00	4 26.67
Total	3 100.00	8 100.00	4 100.00	15 100.00

-> gender = Male

RECODE of age (Age)	Major Econ	Math	Politics	Total
18 to 19	2 28.57	0 0.00	1 16.67	3 20.00
20 to 29	3 42.86	2 100.00	0 0.00	5 33.33
30 to 39	2 28.57	0 0.00	5 83.33	7 46.67
Total	7 100.00	2 100.00	6 100.00	15 100.00

```
catplot  hbar major agegroups, percent(major gender)  blabel(bar) by(gender)
```

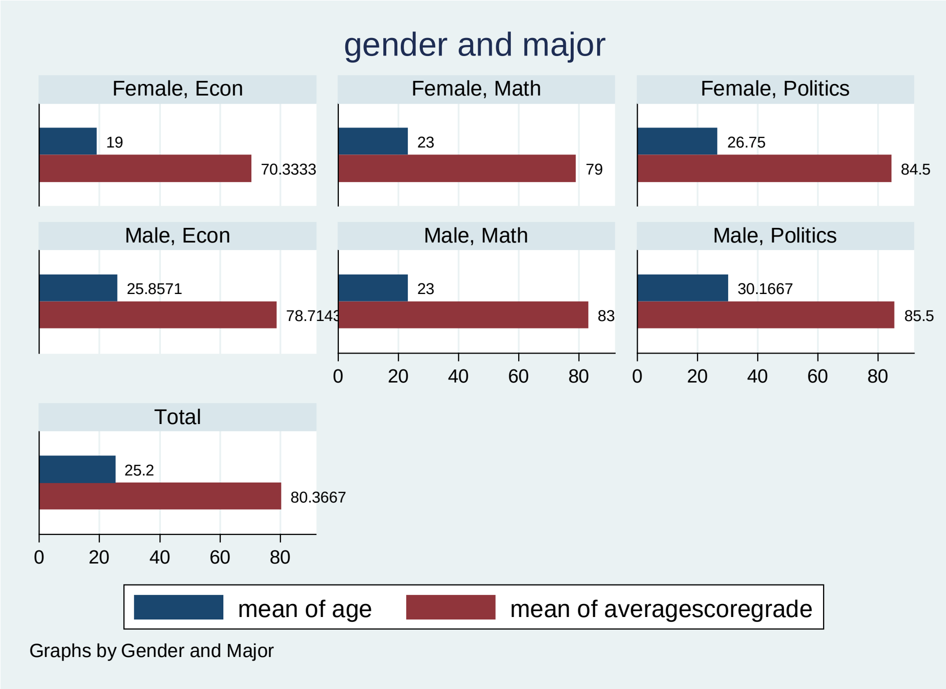




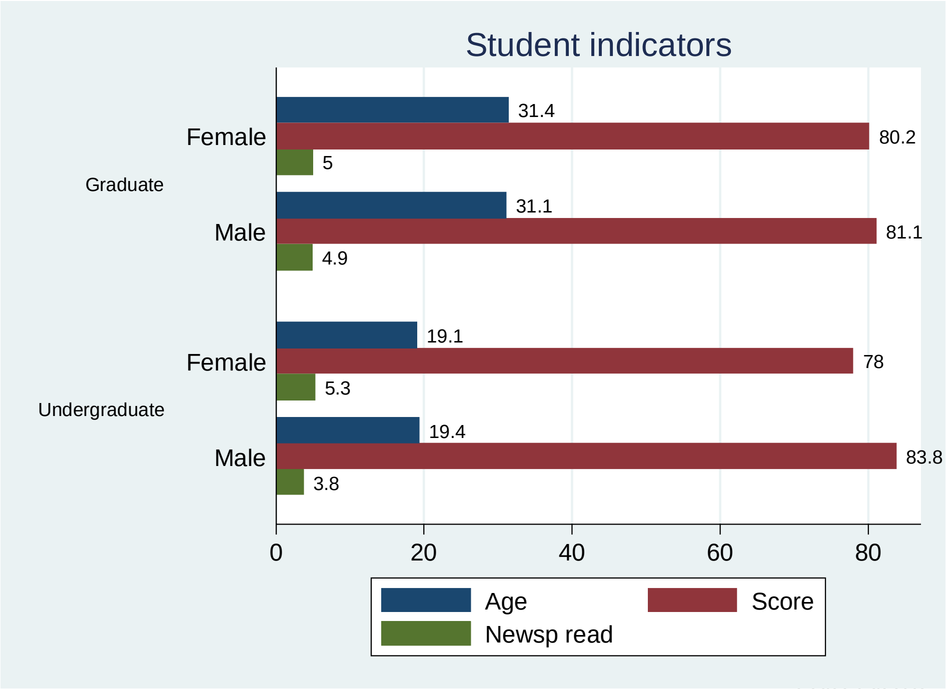
Graphs: means

Stata can also help to visually present summaries of data. If you do not want to type you can go to ‘graphics’ in the menu.

```
graph hbar (mean) age (mean) averagescoregrade,
blabel(bar) by(, title(gender and major)) by(gender
major, total)
```




```
graph hbar (mean) age  averagescoregrade
newspaperreadershiptimeswk, over(gender)
over(studentstatus, label(labsize(small))) blabel(bar)
title(Student indicators) legend(label(1 "Age")
label(2 "Score") label(3 "Newsp read"))
```



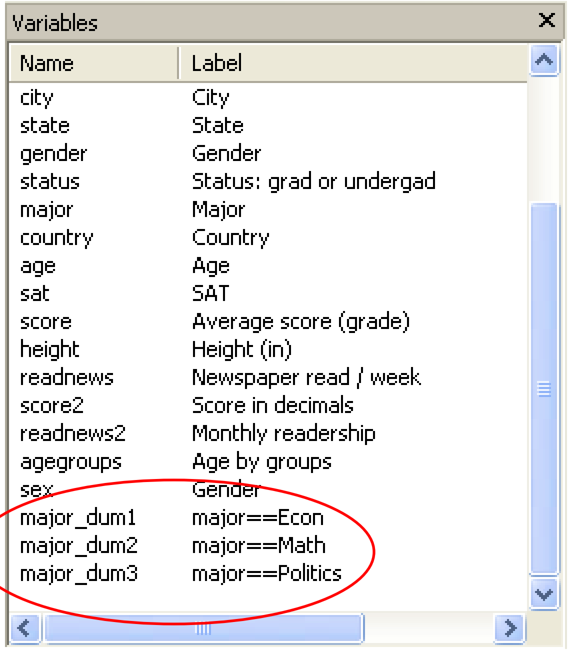
Creating dummies

You can create dummy variables by either using `recode` or using a combination of `tab/gen` commands:

`tab major, generate(major_dum)`

```
. tab major, generate(major_dum)

      Major |      Freq.     Percent        Cum.
------------+-----------------------------------
       Econ |         10       33.33       33.33
       Math |         10       33.33       66.67
   Politics |         10       33.33      100.00
------------+-----------------------------------
      Total |         30      100.00
```

Check the 'variables' window, at the end you will see three new variables. Using `tab1` (for multiple frequencies) you can check that they are all 0 and 1 values



Name	Label
city	City
state	State
gender	Gender
status	Status: grad or undergad
major	Major
country	Country
age	Age
sat	SAT
score	Average score (grade)
height	Height (in)
readnews	Newspaper read / week
score2	Score in decimals
readnews2	Monthly readership
agegroups	Age by groups
sex	Gender
major_dum1	major==Econ
major_dum2	major==Math
major_dum3	major==Politics

```
. tab1  major_dum1 major_dum2 major_dum3

-> tabulation of major_dum1

major==Econ |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         20       66.67       66.67
          1 |         10       33.33      100.00
------------+-----------------------------------
      Total |         30      100.00

-> tabulation of major_dum2

major==Math |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         20       66.67       66.67
          1 |         10       33.33      100.00
------------+-----------------------------------
      Total |         30      100.00

-> tabulation of major_dum3

major==Poli |
       tics |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         20       66.67       66.67
          1 |         10       33.33      100.00
------------+-----------------------------------
      Total |         30      100.00
```

Creating dummies (cont.)

Here is another example:

`tab agregroups, generate(agegroups_dum)`

```
. tab agegroups, generate(agegroups_dum)

  RECODE of |
  age (Age) |      Freq.     Percent        Cum.
------------+-----------------------------------
   18 to 19 |         10       33.33       33.33
   20 to 29 |          9       30.00       63.33
   30 to 39 |         11       36.67      100.00
------------+-----------------------------------
      Total |         30      100.00
```

Check the 'variables' window, at the end you will see three new variables. Using `tab1` (for multiple frequencies) you can check that they are all 0 and 1 values

Variables

Name	Label
status	Status: grad or undergad
major	Major
country	Country
age	Age
sat	SAT
score	Average score (grade)
height	Height (in)
readnews	Newspaper read / week
score2	Score in decimals
readnews2	Monthly readership
agegroups	Age by groups
sex	Gender
major_dum1	major==Econ
major_dum2	major==Math
major_dum3	major==Politics
agegrups_dum1	agegroups==18 to 19
agegrups_dum2	agegroups==20 to 29
agegrups_dum3	agegroups==30 to 39

```
. tab1  agegroups_dum1 agegroups_dum2 agegroups_dum3

-> tabulation of agegroups_dum1

 agegroups== |
    18 to 19 |      Freq.     Percent        Cum.
-------------+-----------------------------------
           0 |         20       66.67       66.67
           1 |         10       33.33      100.00
-------------+-----------------------------------
       Total |         30      100.00

-> tabulation of agegroups_dum2

 agegroups== |
    20 to 29 |      Freq.     Percent        Cum.
-------------+-----------------------------------
           0 |         21       70.00       70.00
           1 |          9       30.00      100.00
-------------+-----------------------------------
       Total |         30      100.00

-> tabulation of agegroups_dum3

 agegroups== |
    30 to 39 |      Freq.     Percent        Cum.
-------------+-----------------------------------
           0 |         19       63.33       63.33
           1 |         11       36.67      100.00
-------------+-----------------------------------
       Total |         30      100.00
```

Frequently used Stata commands

Type help [command name] in the windows command for details

Category	Stata commands
Getting on-line help	**help** **search**
Operating-system interface	**pwd** **cd** **sysdir** **mkdir** **dir / ls** **erase** **copy** **type**
Using and saving data from disk	**use** **clear** **save** **append** **merge** **compress**
Inputting data into Stata	**input** **edit** **infile** **infix** **insheet**
The Internet and Updating Stata	**update** **net** **ado** **news**
Basic data reporting	**describe** **codebook** **inspect** **list** **browse** **count** **assert** **summarize** **Table (tab)** **tabulate**
Data manipulation	**generate** **replace** **egen** **recode** **rename** **drop** **keep** **sort** **encode** **decode** **order** **by** **reshape**
Formatting	**format** **label**
Keeping track of your work	**log** **notes**
Convenience	**display**

Source: http://www.ats.ucla.edu/stat/stata/notes2/commands.htm

Useful links / Recommended books

- ESS https://economics.princeton.edu/undergraduate-program/ess/#
- UCLA Resources to learn and use STATA http://www.ats.ucla.edu/stat/stata/
- *Introduction to Stata* (PDF), Christopher F. Baum, Boston College, USA. “A 67-page description of Stata, its key features and benefits, and other useful information.” http://fmwww.bc.edu/GStat/docs/StataIntro.pdf
- STATA FAQ website http://stata.com/support/faqs/

Books

- *Introduction to econometrics* / James H. Stock, Mark W. Watson. 2nd ed., Boston: Pearson Addison Wesley, 2007.
- *Data analysis using regression and multilevel/hierarchical models* / Andrew Gelman, Jennifer Hill. Cambridge ; New York : Cambridge University Press, 2007.
- *Econometric analysis* / William H. Greene. 6th ed., Upper Saddle River, N.J. : Prentice Hall, 2008.
- *Designing Social Inquiry: Scientific Inference in Qualitative Research* / *Gary King, Robert O. Keohane, Sidney Verba, Princeton University Press*, 1994.
- *Unifying Political Methodology: The Likelihood Theory of Statistical Inference* / Gary King, Cambridge University Press, 1989
- *Statistical Analysis: an interdisciplinary introduction to univariate & multivariate methods /* Sam Kachigan, New York : Radius Press, c1986
- *Statistics with Stata (updated for version 9) /* Lawrence Hamilton, Thomson Books/Cole, 2006