

Data Processing with Stata

Cheat Sheet

For more info, see Stata's reference manual (stata.com)

Useful shortcuts

- F2** — keyboard buttons describe data
- Ctrl + 9** open a new do-file
- Ctrl + 8** open the data editor
- Ctrl + D** highlight text in do-file, then ctrl + d executes it in the command line
- clear** delete data in memory

AT COMMAND PROMPT

- PgUp** **PgDn** scroll through previous commands
- Tab** autocompletes variable name after typing part
- cls** clear the console (where results are displayed)

Set up

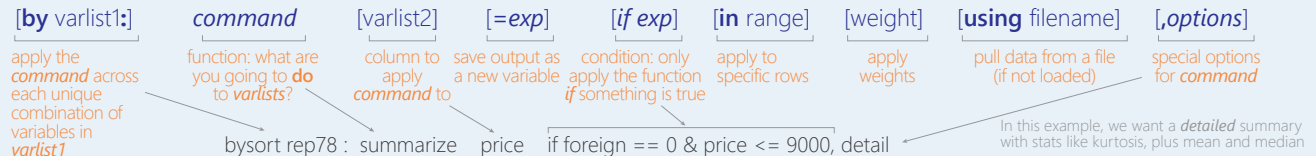
- pwd** print current (working) directory
- cd "C:\Program Files\Stata16"** change working directory
- dir** display filenames in working directory
- dir *.dta** List all Stata data in working directory
- capture log close** close the log on any existing do-files
- log using "myDoFile.txt", replace** create a new log file to record your work and results
- search mdesc** find the package mdesc to install
- ssc install mdesc** install the package mdesc; needs to be done once

Import data

- sysuse auto, clear** load system data (auto data)
- use "yourStataFile.dta", clear** load a dataset from the current directory
- import excel "yourSpreadsheet.xlsx", /* sheet("Sheet1") cellrange(A2:H11) firstrow** frequently used commands are highlighted in yellow
- import delimited "yourFile.csv", /* rowrange(2:11) colrange(1:8) varnames(2)**
- import sas "yourSASfile.sas7bdat", bcat("value labels file")**
- import spss "yourSPSSfile.sav"** see help import for more options
- webuse set "https://github.com/GeoCenter/StataTraining/raw/master/Day2/Data"**
- webuse "wb_indicators_long"** set web-based directory and load data from the web

Basic syntax

All Stata commands have the same format (syntax):



To find out more about any command-like what options it takes-type **help command**

Basic data operations

Arithmetic

- +** add (numbers) combine (strings)
- subtract
- *** multiply
- /** divide
- ^** raise to a power

Logic

- &** and
- !** or **~** not
- |** or
- ==** tests if something is equal
- =** assigns a value to a variable
- <** less than
- <=** less than or equal to
- >** greater than
- >=** greater or equal to

if foreign != 1 & price >= 10000

make	foreign	price
Chevy Colt	0	3,984
Buick Riviera	0	10,372
Honda Civic	1	4,499
Volvo 260	1	11,995

if foreign != 1 | price >= 10000

make	foreign	price
Chevy Colt	0	3,984
Buick Riviera	0	10,372
Honda Civic	1	4,499
Volvo 260	1	11,995

Explore data

VIEW DATA ORGANIZATION

- describe make price** display variable type, format, and any value/variable labels
- count** number of rows (observations) can be combined with logic
- count if price > 5000**
- ds, has(type string)** search for variable types, variable name, or variable label
- lookfor "in."** search for variable types, variable name, or variable label
- isid mpg** check if mpg uniquely identifies the data

SEE DATA DISTRIBUTION

- codebook make price** overview of variable type, stats, number of missing/unique values
- summarize make price mpg** print summary statistics (mean, stdev, min, max) for variables
- inspect mpg** show histogram of data and number of missing or zero observations
- histogram mpg, frequency** plot a histogram of the distribution of a variable

BROWSE OBSERVATIONS WITHIN THE DATA

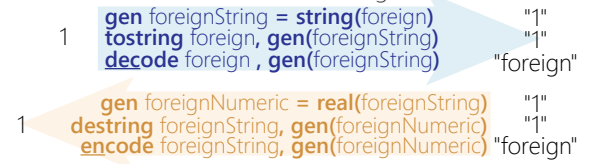
- browse** or **Ctrl + 8** open the data editor
- list make price if price > 10000 & !missing(price)** list the make and price for observations with price > \$10,000
- display price[4]** display the 4th observation in price; only works on single values
- gsort price mpg (ascending) gsort -price -mpg (descending)** sort in order, first by price then miles per gallon
- duplicates report** finds all duplicate values in each variable
- levelsof rep78** display the unique values for rep78
- assert price!=.** verify truth of claim

Change data types

Stata has six data types, and data can also be missing:

no data **true/false** **words** **numbers**
missing **byte** **string** **int** **long** **float** **double**

To convert between numbers & strings:



recast double mpg generic way to convert between types

Summarize data

- tabulate rep78, mi gen(repairRecord)** one-way table: number of rows with each value of rep78
- tabulate rep78 foreign, mi** two-way table: cross-tabulate number of observations for each combination of rep78 and foreign
- bysort rep78: tabulate foreign** for each value of rep78, apply the command tabulate foreign
- tabstat price weight mpg, by(foreign) stat(mean sd n)** create compact table of summary statistics

- table foreign, statistic(mean price) nformat(%9.2f)** create a flexible table of summary statistics
- collapse (mean) price (max) mpg, by(foreign)** replaces data calculate mean price & max mpg by car type (foreign)

Create new variables

- generate mpgSq = mpg^2** create a new variable. Useful also for creating binary variables based on a condition (**generate byte**)
- generate id = _n** creates a running index of observations in a group
- generate totRows = _N** creates a running count of the total observations per group
- pctile mpg Quartile = mpg, nq(4)** create quartiles of the mpg data
- egen meanPrice = mean(price), by(foreign)** calculate mean price for each group in foreign