

Data Processing with Stata 14.1 Cheat Sheet

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

Useful Shortcuts

F2 — keyboard buttons describe data
Ctrl + N — open a new .do file
Ctrl + B — 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 (x86)\Stata13" — change working drive
dir — display filenames in working directory
fs *.dta — List all Stata files in working directory (underlined parts are shortcuts — use "capture" or "cap")
capture log close — close the log on any existing do files
log using "myDoFile.do", replace — create a new log file to record your work and results
search mdesc — find the package mdesc to install (packages contain extra commands that expand Stata's toolkit)
ssc install mdesc — install the package mdesc; needs to be done once

Import Data

sysuse auto, clear — load system data (Auto data) — for many examples, we use the auto dataset.
use "yourStataFile.dta", clear — load a dataset from the current directory
import excel "yourSpreadsheet.xlsx", firstrow — frequently used commands are highlighted in yellow
import delimited "yourFile.csv", firstrow — import an Excel spreadsheet
import delimited "yourFile.csv", firstrow — import a .csv file
webuse set — <https://github.com/CenterForStataTraining/HowtoUseStata14>
webuse "wb_indicators_long" — set web-based directory and load data from the web

Tim Essam (tessam@usaid.gov) • Laura Hughes (lhughes@usaid.gov)
 Disclaimer: we are not affiliated with Stata. But we like it.

Basic Syntax

All Stata functions have the same format (syntax):

[by varlist1]	command	[varlist2]	[=exp]	[if exp]	[in range]	[weight]	[using filename]	[options]
apply the command across each unique combination of variables in varlist1	function: what are you going to do to varlist2?	columns to apply command to	save output as a new variable	condition: only apply the function if something is true	apply to specific rows	apply weights	pull data from a file (if not loaded)	special options for command

Example: `bysort rep78 : summarize price if foreign == 0 & price <= 9000, detail`

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

Basic Data Operations

Arithmetic	Logic	Tests if something is equal	Assigns a value to a variable
add (numbers) + combine (strings)	& and or ! not	== equal != not equal < less than <= less than or equal to > greater than >= greater or equal to	
subtract - multiply * divide / raise to a power ^			

Explore Data

VIEW DATA ORGANIZATION	SEE DATA DISTRIBUTION
describe make price — display variable type, format, and any value/variable labels	codebook make price — overview of variable type, stats, number of missing/unique values
count if price > 5000 — number of rows (observations). Can be combined with logic	summarize make price mpg — print summary statistics (mean, stdev, min, max) for variables
ds, has(type string) — search for variable types, variable name, or variable label	inspect mpg — show histogram of data, number of missing or zero observations
lookfor — search for variable types, variable name, or variable label	histogram mpg, frequency — plot a histogram of the distribution of a variable

BROWSE OBSERVATIONS WITHIN THE DATA

browse or **Ctrl + B** — Missing values are treated as the largest positive number. To exclude missing values, ask whether the value is less than "."

list make price if price > 10000 & price < . — compact form

list make price 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

levels of rep78 — display the unique values for rep78

Change Data Types

Stata has 6 data types, and data can also be missing:

no data	truncatable	words	numbers
missing	byte	string	int long float double

To convert between numbers & strings:

```

1 gen foreignString = string(foreign)
1 tostring foreign, gen(foreignString)
1 decode foreign, gen(foreignNumbers)
1 gen foreignNumbers = real(foreignString)
1 destring foreignString, gen(foreignNumbers)
1 encode foreignNumbers, gen(foreignString)
  
```

recast double mpg — generic way to convert between types

Summarize Data

include missing values — create binary variable for every rep78 value in a new variable: `repairRecord`

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 (displays stats for all data)

table foreign, contents(mean price sd price) f(%9.2f) row — 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 — `bysort rep78: gen repairid = _n` — creates a running index of observations in a group

generate totRows = _N — `bysort rep78: gen repairTot = _N` — creates a total count of 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 (see help egen for more options)

Inspired by RStudio's awesome Cheat Sheets (rstudio.com/resources/cheatsheets)

[procenter.github.io/StataTraining](https://github.com/CenterForStataTraining)

updated January 2016
 CC-BY-NC