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

underlined parts are shortcuts – use "capture" or "cap"

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

packages contain extra commands that expand Stata's toolkit

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

frequently used commands are highlighted in yellow

import excel "yourSpreadsheet.xlsx", /*
*/ **sheet(**"Sheet1"**) cellrange(**A2:H11**) firstrow**

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):

[**by** varlist1**:**] *command* [varlist2] [=*exp*] [*if exp*] [**in** range] [weight] [**using** filename] [,*options*]

[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 *varlists*?	column 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*

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

In this example, we want a ***detailed*** summary with stats like kurtosis, plus mean and median

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

== tests if something is equal
= assigns a value to a variable

&	and	**==**	equal	**<**	less than
! or **~**	not	**!=** or **~=**	not equal	**<=**	less than or equal to
|	or			**>**	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
count if price > 5000
number of rows (observations) can be combined with logic

ds, has(type string)
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

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 & !**missing**(price) **clist** ... (compact form)
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

assert price!=.
verify truth of claim

levelsof rep78
display the unique values for rep78

Change Data Types

Stata has 6 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:

1 → "1" "1" "foreign"

gen foreignString **= string(**foreign**)**
tostring foreign, **gen(**foreignString**)**
decode foreign , **gen(**foreignString**)**

"1" "1" "foreign" → 1

gen foreignNumeric **= real(**foreignString**)**
destring foreignString, **gen(**foreignNumeric**)**
encode foreignString, **gen(**foreignNumeric**)**

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

formats numbers for all data; displays stats

table foreign, **contents(**mean price sd price**)** f(%9.2fc) **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 **gen byte** lowPr = price < 4000
create a new variable. Useful also for creating binary variables based on a condition (**generate byte**)

generate id = **_n** **bysort** rep78**: gen** repairIdx = **_n**
_n creates a running index of observations in a group

generate totRows = **_N** **bysort** rep78**: gen** repairTot = **_N**
_N creates a running count of the total observations per group

pctile mpgQuartile = 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

Tim Essam (tessam@usaid.gov) • Laura Hughes (lhughes@usaid.gov)
follow us @StataRGIS and @flaneuseks

inspired by RStudio's awesome Cheat Sheets (rstudio.com/resources/cheatsheets)
geocenter.github.io/StataTraining
updated July 2019