Working with Stata

Leslie Martin

Uni Melb honours program
March 18, 2014

We will cover first steps and advanced

1. doFiles
2. Importing data
3. Inspecting and cleaning data
4. Graphs and regressions, exporting results to LaTeX

Good practice file management

Stata/

doFiles	sets of commands; one file for each set of tasks
OrigData	original data, never over-write
Data	working datasets
Graphs	graphs for your paper
Tables	summary and regression tables for your paper
Logs	record of commands and screen output

master.do

AUTOMATE everything
– it should be easy to replicate your results

Stata: Use all of the windows

- Command prompt: single lines, to try things out
- doFile: your record of commands that work
- Results window: check out raw output
- LaTeX (or Word) for readable regression output
- Variable list, properties
- Browser: inspect individual observations

In command prompt: Ctrl-R (on Mac) to pull up last line typed

SAMPLE MASTER.DO FILE

```
clear
set more off
capture log close
log using Logs/master, replace text
cd Documents/India/Stata/

do doFiles/import.do
do doFiles/clean.do
do ``doFiles/summary statistics.do''
do doFiles/regressions.do
```

DO-FILE FORMATS

Line breaks:

- 1 command per line unless `#delimit ;` revert using `#delimit cr`
- `///` to extend command to next line

Annotate your code:

- at end of a comand: `//` blah blah
- full line: `*` blah dee blah
- several lines: `/*` yah dee yah dee ya `*/`

IMPORTING DATA INTO STATA

What format is your data in?

- Stata (filename.dta):
  `use OrigData/statadata, clear`
- tab or comma -delimited (filename.csv or filename.txt):
  `insheet using OrigData/textfile.csv, [names] clear`
- other:
  `edit` (paste, close window, choose save option)

Note: always run `clear` before trying to load a new dataset or add `clear` as option to load commands as above

LOADING SUBSETS OF OBSERVATIONS

- First 100:
  `insheet using OrigData/data.txt in 1/100`
- Random sample (1%):
  `insheet if runiform()< .01 using OrigData/data.txt`
- Conditional on values:
  `insheet if store== 10 using OrigData/data.txt`

DICTIONARY FILES

```
infile using OrigData/dict.txt, using(OrigData/data.txt)
```

where `dict.txt` contains, say, the following:

```
dictionary {
long        year        %4f     "Year of survey"
str8        firmid      %8s     "Factory ID"
int         open        %1f     "Factory is open"
str24       statename   %24s    "State"
long        industry    %4f     "Industry (4-digit level of NIC-98)"
}
```

Note: make sure to leave a hard return at the end of the dictionary text file

FORMATS

Color in editor:

black	numeric	byte, int, long, float, double
red	string	
blue	label	

Common conversions:

- `destring` stringvar, [`ignore`("chars")] `gen`(numvar)
- `encode` stringvar, `gen`(numvar)
- `compress` numvar

Note: if any observation includes a string character (e.g. "N/A"), Stata will assume the insheet-ed variable is a string

Naming variables

Never (try to) give two variables the exact same name

Ideally names also unique across datasets (within project)

- case-sensitive: `ID` $\neq$ `id` $\neq$ `Id`
- without asking (!) Stata will auto-complete variable names in commands if unambiguous
- wildcards: ? single character, * multiple character
- if varnames end with number, can call range: year1998-year2006

Naming variables

Good practice: names that make sense without being too long

averageemploymentbystateandyear	X
emp7	X
labor_stateyear	√

Dummy variables: best names make value codes obvious

gender	X
female	√ (1=female, 0=male)

LABEL DATASETS, VARIABLES, AND VALUES

```
label data "Where I got this data, what it contains"

label var kWh "Household consumption in kWh/year"

label define statevals 1 "Victoria"
label define statevals 2 "New South Wales", add
label define statevals 3 "Queensland", add
label values state statevals


tab state, nolabel
browse, nolabel
```

reminder of what's assigned to what: `label list`

Missing values

A 0 is not the same as a missing value. Neither is -9

Stata represents missing values with

- Numeric variable: .
- String variable: ""

Important:

- Often first step in cleaning data is to correctly code missing data as missing
- In calculations, Stata interprets numeric missing values as infinitely large (!)
- Observations with missing values in any called variable will be dropped from regressions

Inspecting your data

`describe`	about the dataset
`codebook, [compact]`	all variables, labels, mean, min/max
`inspect` var	positive, negative, missing values
`unique` var	number unique values

EXCEL-LIKE BROWSER/EDITOR

Observations in rows, variables in columns

`browse` [varlist]	can't make changes (safer)
`edit` [varlist]	overwrite possible

Before opening the browser:

`order` var1 var2 ...	moves variables to left
`sort` var1 var2 ...	sorts (i.e. moves observations to top)
`gsort` -var1 var2 ...	allows reverse order sorting

Inspecting your data

`sum` var	statistics (numeric vars only)
`sum` var, `detail`	show percentiles
`by` var1, `sort: sum` var2	conditional statistics
`tab` var	values & frequencies
`tab` var1 var2	2x2 table of values, frequencies
`tab` var1, `sum`(var2)	stats var2 by value of var1
`table` var1, `c(sum` var2`)`	tables with statistics

Cleaning

Drop variables or observations:

`drop` uselessvar1 uselessvar2

`drop if` todrop`==1`

`keep if` todrop`~=1`

Rename variable:

`rename` oldname newname

Replace values:

`replace` todrop`=1 if missing`(hhid)

`recode` dummyvar `(1=0) (2=1) (-9=.)`

Identify and/or drop duplicate observations:

`duplicates tag` [varlist], `gen`(var)

`duplicates drop` [varlist], `force`

Creating new variables

```
gen x3 = (x==3)
gen age4 = 1 + (age>20) + (age>40) + (age>60)
tab x, generate(newvar)
by state district year, sort:  gen freq = _N
by hhold :  gen oldest = _n == _N
egen hhinc = sum(inc), by(hhold)
gen newid = string(id) + stringvar
```

EGEN CAN CREATE (ALMOST) ANYTHING

`mean`	average
`sum`	not rolling (`gen sum` produces rolling sum)
`group`	create ids for sets of variables
`min`	min value
`max`	max value
`pctile`, `p`(#)	#th percentile
`count`	number of non-missing observations

Loops

```
foreach i of numlist 1 3 6 9/12 102(2)110 {
foreach i of varlist age female state {
foreach i in 1 3 age {

forvalues i = 1/10 {
   gen var`i' = whatever
   more commands...
   }
```

COLLAPSING DATASETS (1 OF 3)

12 x 100 x 3 = 3600 observations of kWh/month electricity consumption for 100 households in 5 postcodes over 3 years

```
collapse (sum) kWh, by(household year)
```

$\Rightarrow$ 100 x 3 = 300 household-year observations of total annual household consumption

COLLAPSING DATASETS (2 OF 3)

12 x 100 x 3 = 3600 observations of kWh/month electricity consumption for 100 households in 5 postcodes over 3 years

```
collapse (mean) kWh, by(postcode year)
```

$\Rightarrow$ 5 x 3 = 15 observations of average household consumption in each postcode for each year

COLLAPSING DATASETS (3 OF 3)

12 x 100 x 3 = 3600 observations of kWh/month electricity consumption for 100 households in 5 postcodes over 3 years

```
collapse (sum) kWsum=kW (mean) kWmean=kW, by(year)
```

⇒ 3 observations of total annual consumption and average annual consumption

I usually save the dataset before collapsing and then use the saved version later, but you can equivalently type `preserve` before and `restore` after

Reshaping datasets

1000 observations (1 for each postcode)
5 variables: postcode kW2001 kW2002 kW2003 kW2004

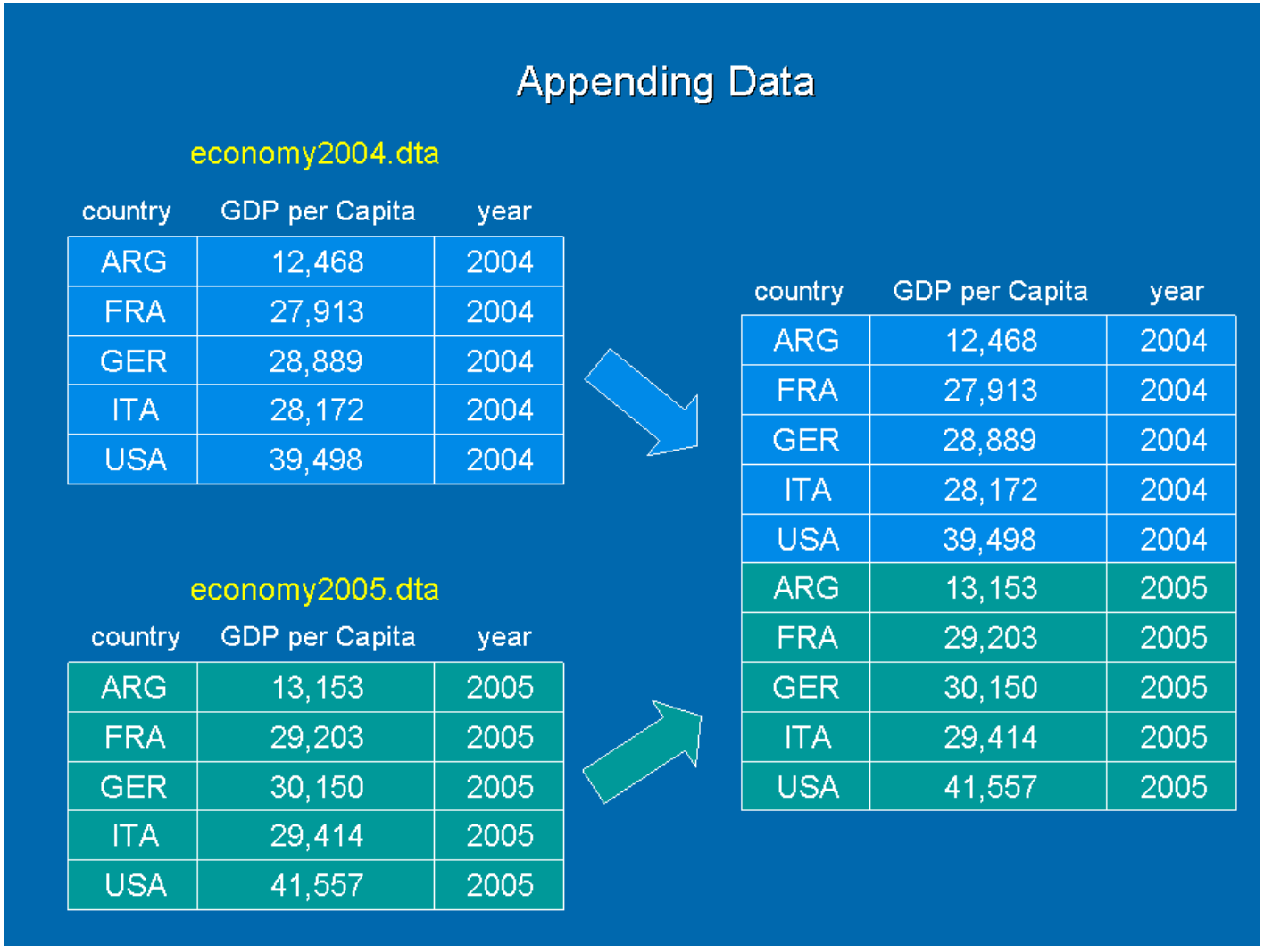
```
reshape long kW, i(postcode) j(year)
```

$\Rightarrow$ 4000 observations (4 for each postcode)
3 variables: postcode year kW
year takes on 4 values: 2001 2002 2003 2004

```
reshape wide kW, i(postcode) j(year)
```

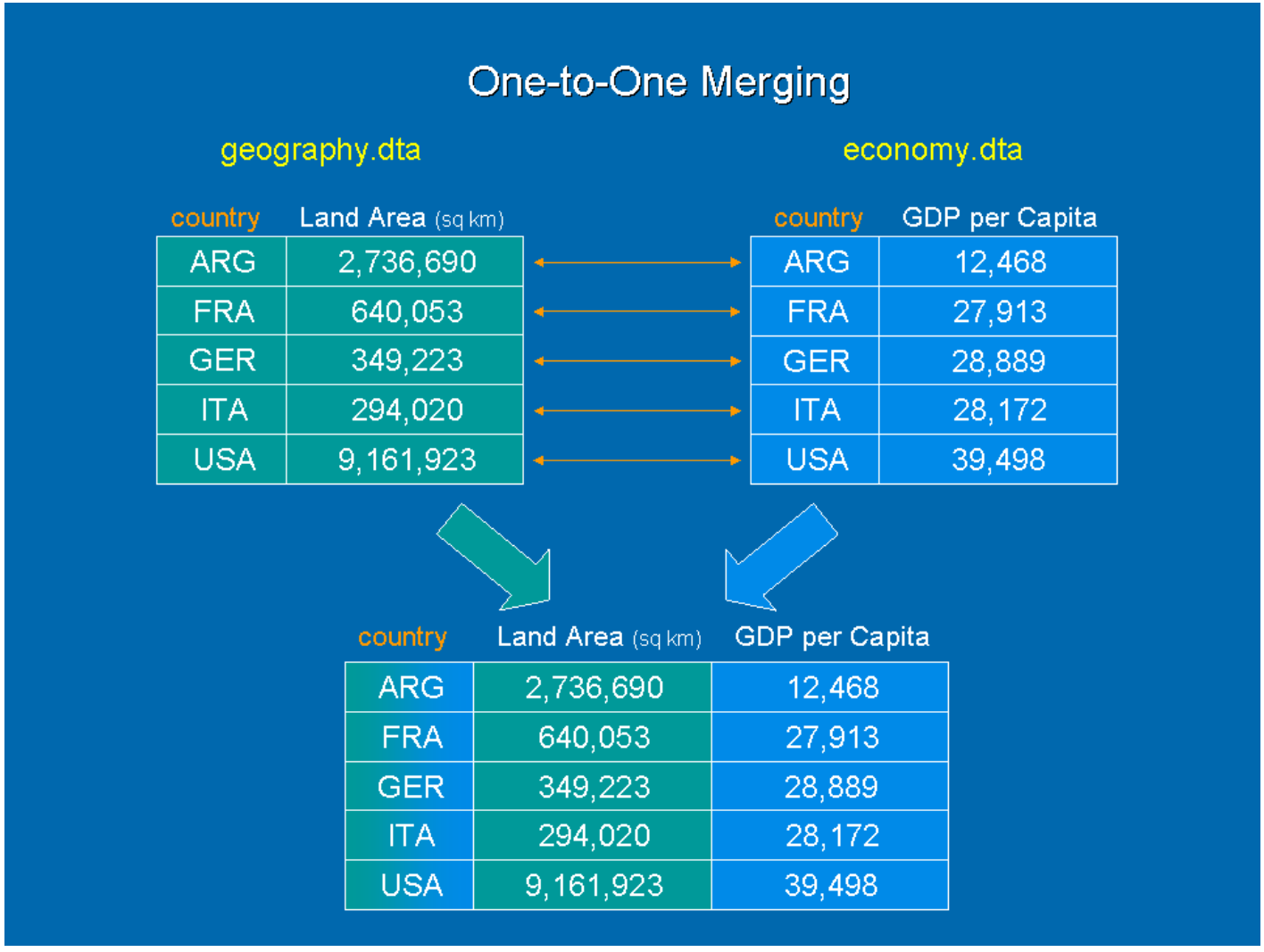
$\Rightarrow$ back to original

Appending datasets



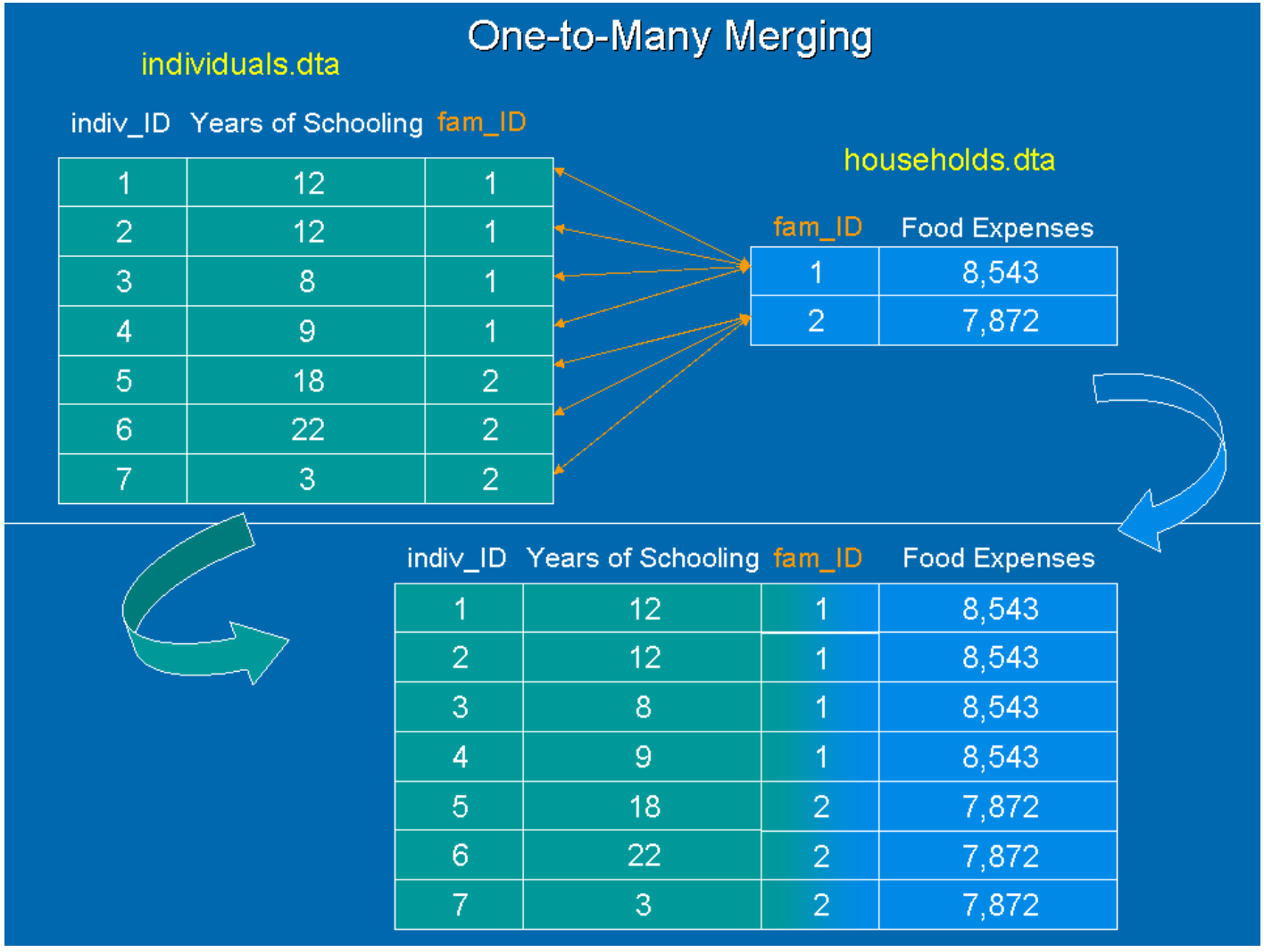
Source: http://www.paulgubbins.com

Merging: one-to-one



Source: http://www.paulgubbins.com

Merging: many-to-one



Source: http://www.paulgubbins.com

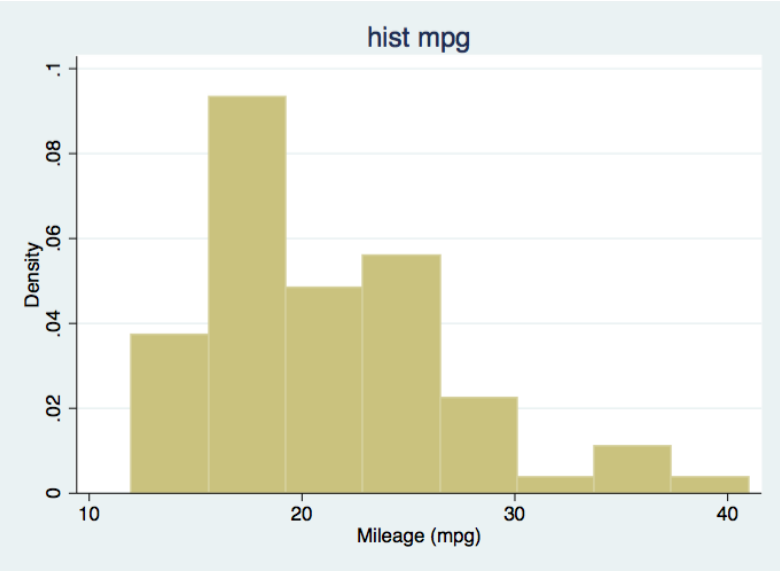
Merging datasets

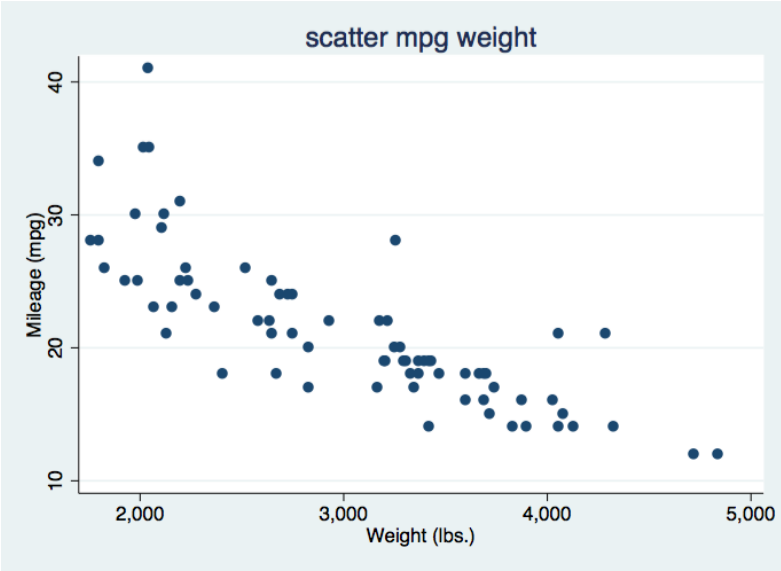
1. Open using dataset. Check for duplicates in varlist. Sort by varlist. Save.
2. Open master dataset. Sort by varlist.
3. Run either
   ```
   merge 1:1 varlist using filename
   merge m:1 varlist using filename
   ```
   Options: `keepusing(`keyvars`)` to ignore other vars in using dataset; `update` to update missing values in master dataset with non-missing values in using
4. Check merge results using `tab _merge`
5. Common to drop observations that appear only in the using dataset
   ```
   drop if _merge==2
   ```
6. Drop or rename _merge

String commands (advanced)

`strpos`(s1,s2)	search in s1 for s2, return position s2 first found
`substr`(s,n1,n2)	returns substring of s, starting at n1 for length n2
	(n2=. returns entire remaining string)
`strmatch`(s1,s2)	returns 0,1 for match, use wildcards
`regexm`(s,re)	returns 1 if expression holds, else 0
`regexs`(n)	returns subexpression from regexm
	(n=0 provides entire string)
`regexr`(s1,re,s2)	replaces first substring in s1 that matches re with s2

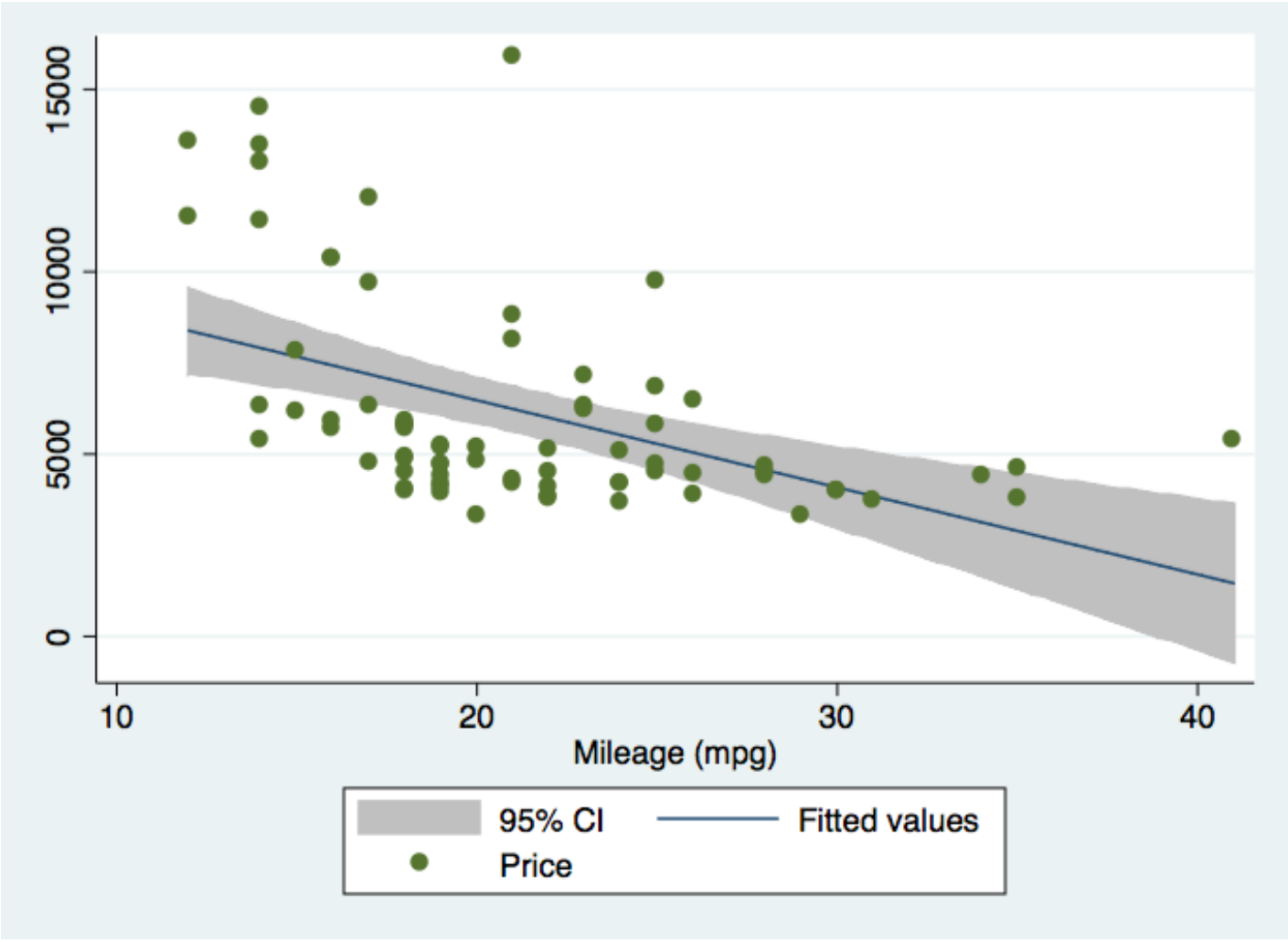
Graphs





Easiest way to learn graph commands: use windows editor to select options, then cut and paste code generated

```
twoway (lfitci price mpg) (scatter price mpg)
```



When you've got it as you want it

```
twoway scatter tfp year, connect(l) lcolor(black) msymbol(T)
mcolor(black) || scatter tfpwithin year, connect(l) lcolor(blue)
mcolor(blue) msymbol(X) || scatter tfpbetween year, connect(l)
lcolor(red) mcolor(red) msymbol(Oh) legend(label(1 "Aggregate")
label(2 "Within") label(3 "Between")) title("State-Industry
Total Factor Productivity") ytitle("TFP") xline(7) yline(0,
lcolor(gs8)) xtitle("Year")

graph export "Graphs/state industry TFP.png", as(png) replace
```

Some regressions

`reg` y x1 x2	OLS
`reg` y x1 x2, `robust`	OLS, Newey-White SE
`areg` y x1 x2, `absorb`(id) `vce(robust)`	with fixed effects (hidden)
`xi: reg` y x1 `i.`x2 `i.`x2*x1	with interactions
`ivregress 2sls` y x1 (x2 = instruments)	2SLS
`logit` y x1 x2	logit
`clogit` y x1 x2, `group`(id)	conditional logit

REGRESSIONS: PANEL DATA

`xtset` id year	set panel and time vars
`xtreg` y x1 x2, fe vce(robust)	fixed effects, control for heteroskedasticity
`xtreg` y x1 x2, fe vce(cluster state)	fixed effects, cluster SE

SOME USER-CREATED COMMANDS: .ADO FILES

`ivreg2`	extra options IV regressions
`tsls`	two-stage least squares (faster than `xtivreg`)
`spmap`	GIS facilities within Stata
`vwlowess`	non-parametric estimation
`sutex`	export summary statistics to LaTeX

```
ssc install filename
```

Exporting regression output to LaTeX

See demo

EXPORTING REGRESSION OUTPUT TO LATEX

```
local base x1 x2 x3 x4
local extra x5 x6 x7 x8

reg y `base', robust
est2vec tableC, replace vars(`base' `extra') e(r2) name(reg1)

reg y `base' `extra', robust
est2vec tableC, addto(tableC) name(reg2)

est2tex tableC, path(Tables/) replace preserve mark(stars) ///
  digits(3) fancy label collab("Base" "Extra~Controls")
```

Saving regression coefficients

```
reg y x1 x2
ereturn list
```

Save coefficients:

```
matrix bvec = get(_b)
matrix bvec = e(b)
```

Save standard errors:

```
matrix Vvec = get(VCE)
matrix Vvec = e(V)
```

Errors Stata will point out for you

- Forgetting to close active log file before opening a new one
- Confusing = and ==
- Running commands for numeric variables on string variables
- Trying to merge a second time without dropping first _merge
- Running files from the wrong directory

Mistakes Stata will let you make

- Forgetting to exclude missing values when generating new variables using $>$ or $>=$
- Adding a categorical variable as-is to a regression
- Losing significant digits when importing: import as string
- Not removing thousands-separator commas (or not checking EU comma to AU decimal conversion) before exporting from Excel
- Running a regression in logs forgetting that all observations with a value of 0 will be dropped
- Forgetting to update missing values when merging
- Running a subset of a do-file that omits local variable definitions
- Calling an old/not updated dataset in a do-file

NEED HELP?

Stata:

- `help` command
- `findit` approximate-command
- google
- do-files templates: from friends, AER papers, Deaton's 1997 "The Analysis of Household Surveys", online supplemental materials to the below books
- Baum 2006 "An Introduction to Modern Econometrics using Stata"
- Cameron & Trivedi 2009 "Microeconometrics using Stata"

Theory:

- Wooldridge 2002, 2010 "Econometric Analysis of Cross Section and Panel Data"
- Cameron & Trivedi 2005 "Microeconometrics: Methods and Applications"

Live document

Please help future classes by emailing me any important commands/mistakes/caveats that you would like added to this document

leslie.martin@unimelb.edu.au