

# Stata Workshop 2: Data Management

Chang Y. Chung

September 11, 2014

# Resources

- Prof. Rodríguez <http://data.princeton.edu/stata/>
- Mitchell(2010) *Data Management Using Stata*  
<http://www.stata-press.com/books/data-management-using-stata/>
- UCLA ATS <http://www.ats.ucla.edu/stat/stata/>
- Stata Support Main <http://stata.com/support/>
- Stata Resources <http://stata.com/links>
- Stata help has links to manual pdf files or using browser:  
<http://www.stata-press.com/manuals/documentation-set/>
- This presentation and do files are available at github:  
<http://github.com/Chang-Y-Chung/dm>

# Topics

- Dataset
- Describe / List
- Tabulate / Summarize
- Generate / Replace
- Import from / Export to Excel file
- Append / Merge
- Infile (Free format / Using a dictionary)
- Date / Time \*
- By-Group Processing / Egen \*

Last two are for self-study. Stata do files are provided



# Setup

- example do file (dm.do) and other files are in the zip file attachment (dm.zip) sent to you this afternoon

```
// cd to where you put this file (dm.do)  
cd z:\dm
```

```
// check which directory I am in  
pwd
```

# Display

```
clear all
display 1 + 2

display ln(0.3 / (1 - 0.3))
display logit(0.3)

// it can display strings as well
display "hello, world?"

// some system values
display c(current_date)
```

# Stata Dataset

```
// dataset is an array of observations (rows) on variables (columns)  
clear all  
  
// describe the current stata dataset in memory ("master" dataset)  
describe  
  
// create some observations -- still no variables  
set obs 5  
  
// create a variable, x, with all the values equal to 1  
generate x = 1  
  
// create another variable, y, with the built-in obs number, _n  
generate y = _n  
  
// save the master data into a file on the harddrive  
save mydata.dta, replace
```

# Use and List a Dataset

```
use mydata, clear // load the data into main memory
list
```

```
## . use mydata, clear // load the data into main memory
## . list
##      +-----+
##      |  x   y  |
##      |-----|
##  1.  |  1   1  |
##  2.  |  1   2  |
##  3.  |  1   3  |
##  4.  |  1   4  |
##  5.  |  1   5  |
##      +-----+
```

# Replace

```
use mydata, clear

replace x = 2

// replace is often used with either "in" or "if"
replace x = 3 in 1/3
replace y = 9 if y == 5

// you can refer to other variables in the condition as well
replace x = -99 if y < 3

// suppose that -99 in x and 9 in y are missing values
replace x = . if x == -99
replace y = . if y == 9

save mydata2, replace
```



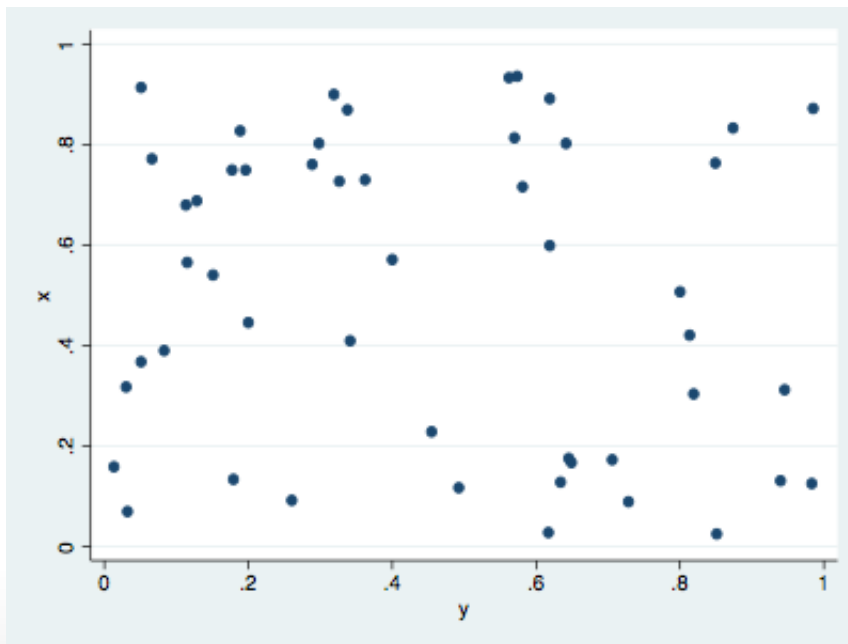
# See mydata2.dta

```
use mydata2, clear  
list in 1/5
```

```
## . use mydata2, clear  
## . list in 1/5  
##      +-----+  
##      | x   y |  
##      |-----|  
##  1. | .   1 |  
##  2. | .   2 |  
##  3. | 3   3 |  
##  4. | 2   4 |  
##  5. | 2   . |  
##      +-----+
```

# Random Data

```
clear all
set obs 50
set seed 12345
generate x = runiform()
generate y = runiform()
twoway scatter x y
graph export random.png, width(400) height(300) replace
```



# Any questions so far?

# Missing Values

```
use mydata2, clear  
list
```

```
## . use mydata2, clear  
## . list  
##      +-----+  
##      | x   y |  
##      |-----|  
##  1. | .   1 |  
##  2. | .   2 |  
##  3. | 3   3 |  
##  4. | 2   4 |  
##  5. | 2   . |  
##      +-----+
```

# Dichotomizing y around 2.5

```
use mydata2, clear

// this may *not* be correct
generate high_y = 0
replace high_y = 1 if 2.5 < y

// correct way
generate high_y2 = 0 if !missing(y)
replace high_y2 = 1 if 2.5 < y & !missing(y)

save mydata3, replace
```

# mydata3.dta

```
use mydata3, clear  
list y high_y high_y2
```

```
## . use mydata3, clear  
## . list y high_y high_y2  
##      +-----+  
##      | y   high_y   high_y2 |  
##      +-----+  
##  1. | 1       0       0 |  
##  2. | 2       0       0 |  
##  3. | 3       1       1 |  
##  4. | 4       1       1 |  
##  5. | .       1       . |  
##      +-----+
```

# Save

```
// create and save
clear all
input id str10 name yob
1 "Amy" 1990
2 "Bill" 1991
3 "Cathy" 1989
end
rename yob year_of_birth
save birth, replace
```

# Use

```
use birth, clear
assert _N == 3
list, abbreviate(15)
```

```
## . use birth, clear
## . assert _N == 3
## . list, abbreviate(15)
##      +-----+
##      | id      name    year_of_birth |
##      +-----+
##  1. |   1      Amy      1990 |
##  2. |   2      Bill      1991 |
##  3. |   3     Cathy      1989 |
##      +-----+
```



# Labels

```
use birth.dta, clear
generate gender = 1 if name == "Amy" | name == "Cathy"
replace gender = 2 if name == "Bill"
tabulate gender
save birth2, replace
```

```
## . use birth.dta, clear
## . generate gender = 1 if name == "Amy" | name == "Cathy"
## (1 missing value generated)
## . replace gender = 2 if name == "Bill"
## (1 real change made)
## . tabulate gender
##      gender |      Freq.      Percent      Cum.
## -----+-----
##           1 |           2       66.67      66.67
##           2 |           1       33.33     100.00
## -----+-----
##       Total |           3      100.00
## . save birth2, replace
## (note: file birth2.dta not found)
## file birth2.dta saved
```

# Labeling Values Takes Two Steps:

```
use birth2, clear

// 1. create the value label itself. we use the same name
label define gender 1 "girl" 2 "boy"

// 2. attach the value label to a variable
label values gender gender

save birth3, replace
```

# Check

```
use birth3, clear
```

```
tabulate gender
```

```
## . use birth3, clear
## . tabulate gender
##      gender |      Freq.      Percent      Cum.
## -----+-----
##      girl  |          2       66.67       66.67
##      boy   |          1       33.33      100.00
## -----+-----
##      Total |          3      100.00
```

# Labeling a Variable

```
use birth3, clear
```

```
// labeling a variable is simpler
```

```
label var gender "Gender of the respondent"
```

```
describe gender
```

```
## . use birth3, clear
```

```
## . // labeling a variable is simpler
```

```
## . label var gender "Gender of the respondent"
```

```
## . describe gender
```

```
##           storage   display   value
```

```
## variable name   type     format   label      variable label
```

```
## -----
```

```
## gender           float    %9.0g    gender    Gender of the respondent
```

# Some Variables from Auto.dta

```
sysuse auto, clear
describe make price mpg foreign
```

```
## . sysuse auto, clear
## (1978 Automobile Data)
## . describe make price mpg foreign
##
```

## variable name	storage type	display format	value label	variable label
## make	str18	%-18s		Make and Model
## price	int	%8.0gc		Price
## mpg	int	%8.0g		Mileage (mpg)
## foreign	byte	%8.0g	origin	Car type

```
## -----
```

# Tabulate

## *With Value Label*

```
sysuse auto, clear  
tabulate foreign
```

```
## . sysuse auto, clear  
## (1978 Automobile Data)  
## . tabulate foreign  
##      Car type |      Freq.      Percent      Cum.  
## -----+-----  
##      Domestic |          52       70.27      70.27  
##      Foreign  |          22       29.73     100.00  
## -----+-----  
##           Total |          74      100.00
```

# Tabulate

## *Without Value Label*

```
sysuse auto, clear
tabulate foreign, nolabel
```

```
## . sysuse auto, clear
## (1978 Automobile Data)
## . tabulate foreign, nolabel
##      Car type |      Freq.      Percent      Cum.
## -----+-----
##           0 |          52       70.27      70.27
##           1 |          22       29.73     100.00
## -----+-----
##       Total |          74     100.00
```

# Summarize

```
sysuse auto, clear  
summarize price mpg
```

```
## . sysuse auto, clear  
## (1978 Automobile Data)  
## . summarize price mpg  
##      Variable |      Obs      Mean   Std. Dev.      Min      Max  
## -----+-----  
##      price |      74  6165.257   2949.496     3291   15906  
##      mpg   |      74   21.2973    5.785503      12      41
```



# Other Useful Commands

```
sysuse auto, clear
```

```
describe make mpg price
```

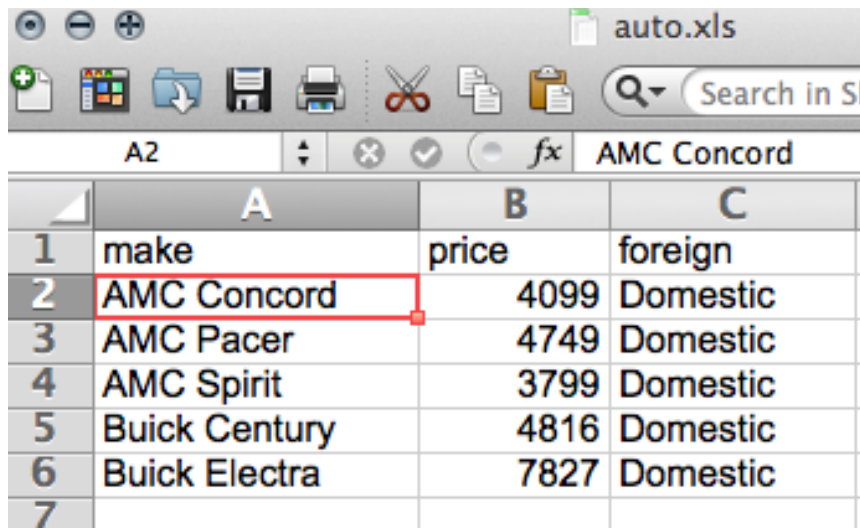
```
inspect make mpg price
```

```
codebook make mpg price
```

# Export to Excel

```
sysuse auto, clear
keep make price foreign
keep in 1/5

export excel using auto.xls, replace first(var)
!start auto.xls // windows
// !open auto.xls // mac
```



	A	B	C
1	make	price	foreign
2	AMC Concord	4099	Domestic
3	AMC Pacer	4749	Domestic
4	AMC Spirit	3799	Domestic
5	Buick Century	4816	Domestic
6	Buick Electra	7827	Domestic
7			

# Import from Excel

```
import excel using auto.xls, clear firstrow  
describe
```

```
## . import excel using auto.xls, clear firstrow
```

```
## . describe
```

```
## Contains data
```

```
##   obs:           5
```

```
##  vars:           3
```

```
##  size:          115
```

```
## -----
```

```
##           storage  display  value
```

```
## variable name  type    format  label    variable label
```

```
## -----
```

```
## make          str13   %13s                make
```

```
## price         int     %10.0g             price
```

```
## foreign       str8    %9s                foreign
```

```
## -----
```

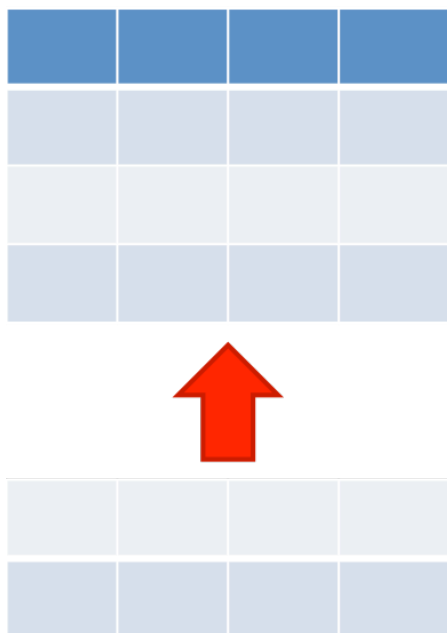
```
## Sorted by:
```

```
##      Note:  dataset has changed since last saved
```

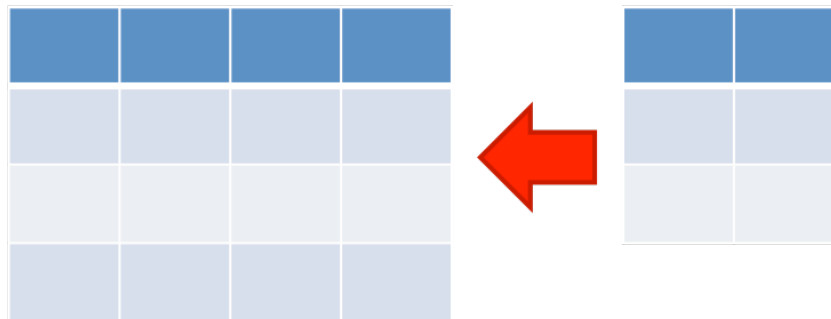
# Break for Q & A

# Combining Datasets

append

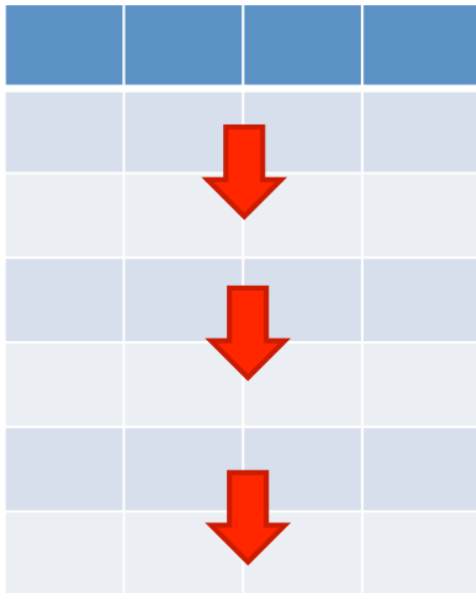


merge

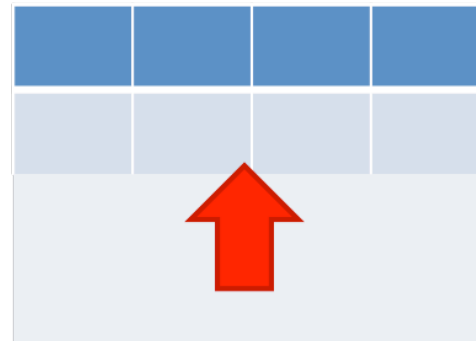


# Combining Datasets

expand



collapse / contract



# Combining Datasets


reshape long





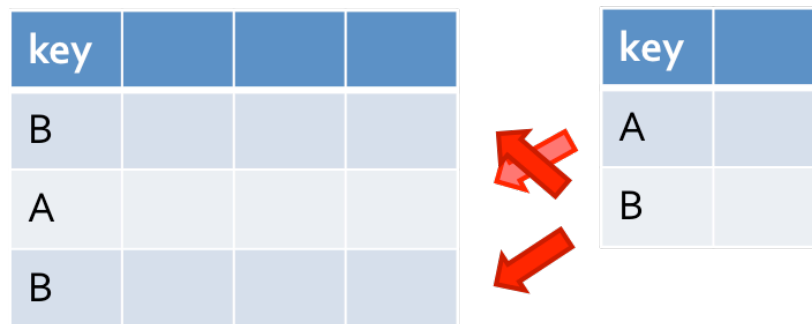

reshape wide

# Combining Datasets

cross



joinby





# Append Example

## Creating odd.dta

```
use http://www.stata-press.com/data/r13/odd1.dta, clear
keep in 1/3
list
save odd.dta, replace
```

```
## . use http://www.stata-press.com/data/r13/odd1.dta, clear
## (First five odd numbers)
## . keep in 1/3
## (2 observations deleted)
## . list
##      +-----+
##      | odd   number |
##      |-----|
##  1.  |    1         1 |
##  2.  |    3         2 |
##  3.  |    5         3 |
##      +-----+
## . save odd.dta, replace
## (note: file odd.dta not found)
## file odd.dta saved
```



# Append Example

## Creating even.dta

```
clear all
input number even odd
4 10 .
5 12 .
end
list
save even.dta, replace
```

```
## . use even.dta, clear
## . list
##      +-----+
##      | number  even  odd  |
##      +-----+
##  1.  |      4     10     .  |
##  2.  |      5     12     .  |
##      +-----+
```

# Append Example

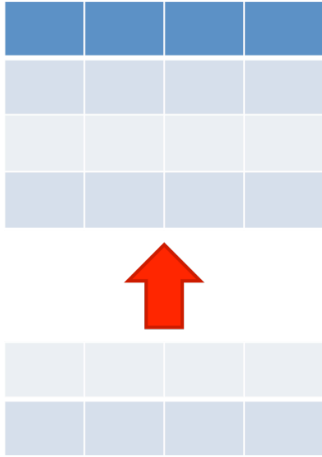
Put odd and even Together

```
use odd.dta, clear
append using even.dta, generate(obsFrom)
list
```

```
## . use odd.dta, clear
## (First five odd numbers)
## . append using even.dta, generate(obsFrom)
## . list
```

```
##      +-----+
##      | odd   number  obsFrom  even |
##      |-----|
##  1. |    1         1         0     . |
##  2. |    3         2         0     . |
##  3. |    5         3         0     . |
##  4. |    .         4         1    10 |
##  5. |    .         5         1    12 |
##      +-----+
```

# Append Pointers



- Syntax: `append using filename [, options]`
- Appends a dataset stored on disk (the *using* dataset) to the end of the dataset in memory (the *master* dataset)
- New master dataset will have more observations than before
- Variables are matched by *name* (not by variable order)
- Non-matched variables on the using side will be included

# Merge Example

```
use age, clear // master
merge 1:1 id using weight, report
save ageWeight, replace
```

Master

id	age
1	22
2	56
5	17

+

Using

id	wgt
1	130
2	180
4	110

=

merge 1:1 id using "using file name"

id	age	wgt	_merge
1	22	130	3
2	56	180	3
5	17	.	1
4	.	110	2

# How Stata Merges

- Manual says (*Stata Data-Management Reference Manual* [D] Release 13, p. 465):

The formal definition for merge behavior is the following: Start with the first observation of the master. Find the corresponding observation in the using data, if there is one. Record the matched or unmatched result. Proceed to the next observation in the master dataset. When you finish working through the master dataset, work through unused observations from the using data. By default, unmatched observations are kept in the merged data, whether they come from the master dataset or the using dataset.

- See also Bill Gould's two-part blog entry on "Merging data" at:
  - Part 1: Merges gone bad <http://tinyurl.com/jvtloka>
  - Part 2: Multiple-key merges <http://tinyurl.com/krhs7xn>

# One-to-One Match Merge Pointers

Master			Using			merge 1:1 id using "using file name"			
id	age		id	wgt		id	age	wgt	_merge
1	22	+	1	130	=	1	22	130	3
2	56		2	180		2	56	180	3
5	17		4	110		5	17	.	1
						4	.	110	2

- Syntax: `merge 1:1 varlist using filename`
- Joins corresponding observations from master and using datasets, matching on the key variable(s).
- Master data are *inviolable*, i.e., if there already exists a variable in master, the values are not replaced.
- By default, merge creates a new variable, `_merge`, which indicates:
  - 1 (master) this obs from master dataset only
  - 2 (using) this obs from using dataset only
  - 3 (match) this obs from both master and using datasets

# Break Time

- Any questions, so far?



# Inputting Raw Data

- Stata stores data in a proprietary format, i.e., the `.dta` file
- Once data are stored in a `.dta` file, we can quickly load the data into memory by the `use` command
- If data are given in other formats, we have to input / read / import them into stata first
- One common such format is known as a raw data file, which stata assumes to have a file extension of `.raw`

# infile Example

```
infile str14 country setting effort change using test.raw, clear  
list in 1/3
```

```
## . infile str14 country setting effort change using test.raw, clear  
## (20 observations read)  
## . list in 1/3  
##      +-----+  
##      | country  setting  effort  change |  
##      +-----+  
##  1. | Bolivia      46      0      1 |  
##  2. |  Brazil      74      0     10 |  
##  3. |   Chile      89     16     29 |  
##      +-----+
```

Bolivia	46	0	1
Brazil	74	0	10
Chile	89	16	29
Colombia	77	16	25
CostaRica	84	21	29
Cuba	89	15	40
DominicanRep	68	14	21
Ecuador	70	6	0
ElSalvador	60	13	13
Guatemala	55	9	4
Haiti	35	3	0
Honduras	51	7	7
Jamaica	87	23	21
Mexico	83	4	9
Nicaragua	68	0	7
Panama	84	19	22
Paraguay	74	3	6
Peru	73	0	2
TrinidadTobago	84	15	29
Venezuela	91	7	11

## Free format raw data

- values are delimited by a space, tab, or comma
- string value is quoted if embeds spaces or commas
- if one observation per line, then consider using `insheet` instead

# Fixed Column Format

- `test.raw` can also be read as a fixed column format, since the values of each variable appear in the fixed columns, for example:
  - country names are always in columns 4 to 17
  - settings values are always in columns 23 and 24
- This information can be stored in a separate *dictionary file*:

- `test.dct`

```
dictionary using test.raw {  
    _column(4)  str14 country  %14s "country name"  
    _column(23) int    settings %2.0f "settings"  
    _column(31) int    effort   %2.0f "effort"  
    _column(40) int    change   %2.0f "change"  
}
```

- Using the dictionary file, the data can be read into stata like so:

```
infile using test.dct, clear
```

# Import / Export Pointers

- Stat/Transfer can import / export data to and from various formats
- But don't blindly trust any piece of software that *translates* data from one system / package / application to another
- Be careful and double-check everything
- Ask help

# Thanks a lot!

- Any questions?