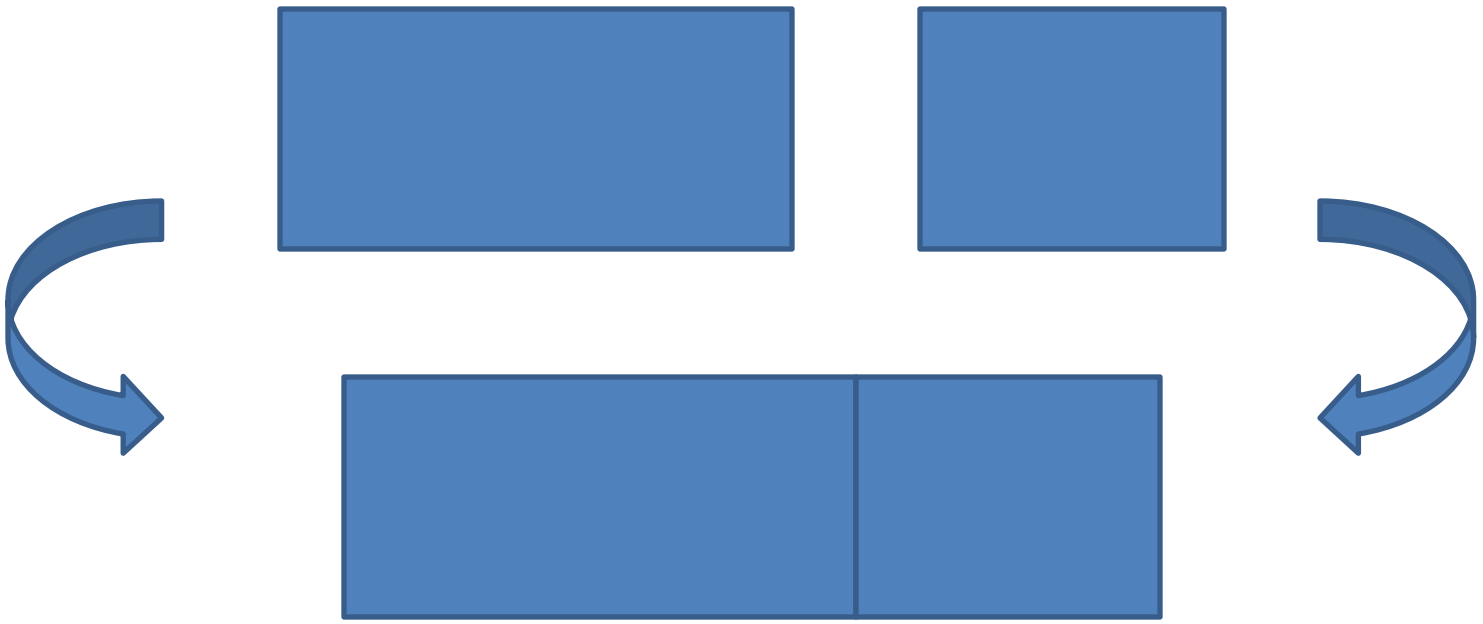
Introduction to Stata Data Management



Chang Y. Chung
Office of Population Research
Princeton University
September 2013

Online Resources

- http://data.princeton.edu/stata/, by German Rodriguez

- *Data Management Using Stata: A Practical Handbook*, by Michael Mitchell, 2010, Stata Press. http://www.stata.com/bookstore/data-management-using-stata/

- UCLA Academic Technology Services (ATS) http://www.ats.ucla.edu/stat/stata/

- Stata Corporation

  http://stata.com/support/

  http://stata.com/links/

Topics

- Display
- Stata Dataset
- Generate / Replace
- Describe / List
- Tabulate / Summarize
- Import from / Export to Excel File
- Append / Merge
- Infile (Free Format / Using a Dictionary)

Display

```
clear all
display 1 + 2

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

// displaying a string
display "hello, world?"

// displaying a system value
display c(current_date)
```

Stata Dataset

- A Stata dataset is a rectangular arrangement of values, where
  - rows are observations
  - columns are variables

```
clear all

// describe the current Stata dataset in memory ("master" dataset)
describe

// create some observations - still no variables
set obs 5

// create a variable named x, which has the
// value of 1 for all observations
generate x = 1

// create another variable y, which has the
// observation number as its value
generate y = _n

list
```

```
     +-------+
     | x   y |
     |-------|
  1. | 1   1 |
  2. | 1   2 |
  3. | 1   3 |
  4. | 1   4 |
  5. | 1   5 |
     +-------+
```

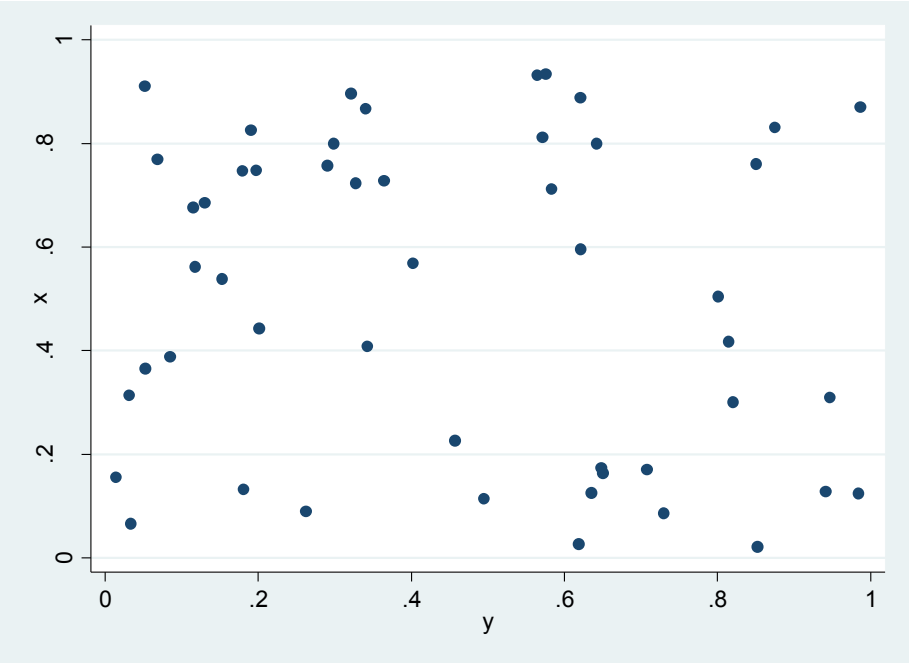
Replace

```
clear all
set obs 5
generate x = 1
generate y = _n

replace x = 2

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

// other variables can be specified in an if condition
replace x = -99 if y < 3

// change the x values of -99 to "missing"
// and change y values of 9 to "missing"
replace x = . if x == -99
replace y = . if y == 9

list
```

	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
```



Missing Values

```
clear all
input x y
. 1
. 2
3 3
2 4
2 .
end

// create new variable high_y
// that dichotomizes y around 2.5
// this is incorrect!
generate high_y = 0
replace high_y = 1 if 2.5 < y

// create high_y2 correctly
generate high_y2 = 0 if !missing(y)
replace high_y2 = 1 if 2.5 < y & !missing(y)
list
```

	x	y	high_y	high_y2
1.	.	1	0	0
2.	.	2	0	0
3.	3	3	1	1
4.	2	4	1	1
5.	2	.	1	.

Save and Use

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

// later, we can bring the dataset back into memory
// via the "use" command
clear all
use birth.dta
assert _N == 3
list
```

```
     +-----------------------+
     | id    name   year_o~h |
     |-----------------------|
  1. |  1     Amy        1990 |
  2. |  2    Bill        1991 |
  3. |  3   Cathy        1989 |
     +-----------------------+
```

Labels

```
clear all
use birth.dta

generate gender = 1 if name == "Amy" | name == "Cathy"
replace  gender = 2 if name == "Bill"
tabulate gender

// associating a variable with a value label requires two steps:
// first, create the value label
label define gender 1 "girl" 2 "boy"

// second, attach the value label to the variable
label values gender gender
tabulate gender

// we can also create a variable label
label variable gender "Gender of the respondent"
describe gender
```

variable name	storage type	display format	value label	variable label
gender	int	%8.0g	gender	Gender of the respondent

Summarize

```
clear all
sysuse auto

list make price mpg foreign in 1/5
list make price mpg foreign in -5/L
```

```
     +-------------------------------------+
     | make           price   mpg   foreign |
     |-------------------------------------|
 70. | VW Dasher      7,140    23   Foreign |
 71. | VW Diesel      5,397    41   Foreign |
 72. | VW Rabbit      4,697    25   Foreign |
 73. | VW Scirocco    6,850    25   Foreign |
 74. | Volvo 260     11,995    17   Foreign |
     +-------------------------------------+
```

```
// variable foreign has a value label
tabulate foreign
tabulate foreign, nolabel
```

```
  Car type |      Freq.     Percent        Cum.
-----------+-----------------------------------
         0 |         52       70.27       70.27
         1 |         22       29.73      100.00
-----------+-----------------------------------
     Total |         74      100.00
```

```
// continuous variables can be
// summarized nicely via the "summarize" command
summarize price
summarize price, detail
```

```
// other commands
inspect price
codebook make price
```

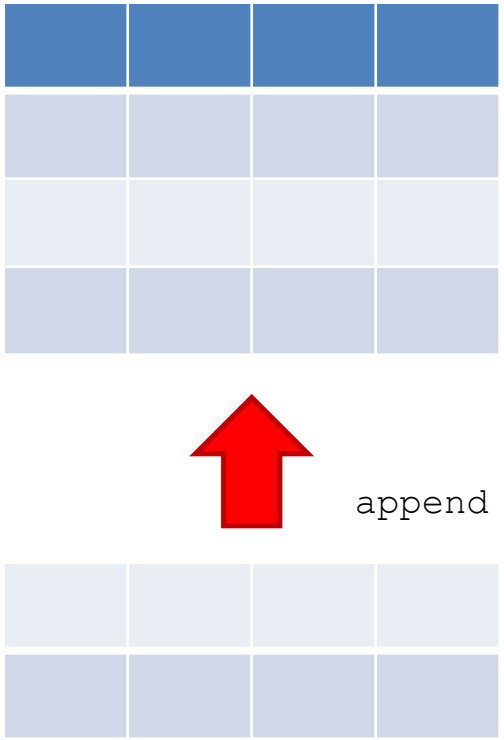
```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       price |        74    6165.257    2949.496       3291      15906
```

Excel

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

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

// bring the excel file back into memory as a Stata dataset
clear all
import excel using auto.xls, clear firstrow

describe
list
```

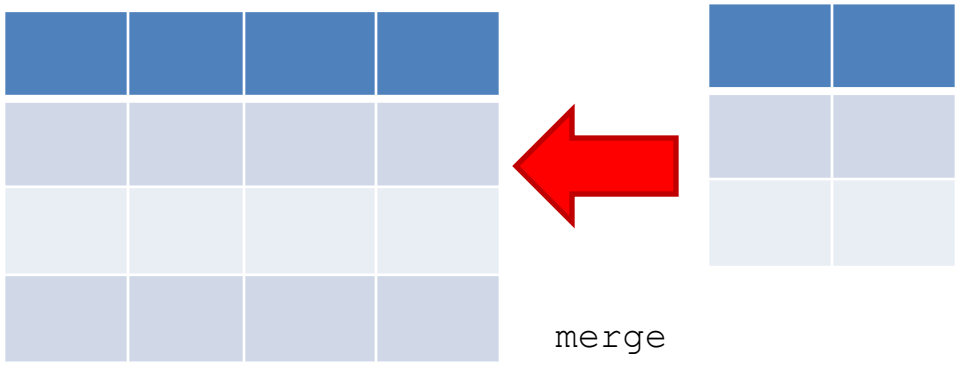
variable name	storage type	display format	value label	variable label
make	str13	%13s		make
price	int	%10.0g		price
mpg	byte	%10.0g		mpg
foreign	str8	%9s		foreign

Combining Datasets



append

Stata dataset stored on disk (the *using* dataset) is added to the end of the dataset in memory (the *master* dataset)



merge

Variables from corresponding observations determined by the key variable(s) are joined to form observations containing variables from both the master dataset and variables from the using dataset.

Append

- Stata dataset stored on disk (the using dataset) is added to the end of the dataset in memory (the master dadtaset)
- Syntax: `append using filename [, options]`
- New master dataset has more observations than before
- Variables are matched by name (not by variable order)
- When combining datasets, the master dataset usually has authority and the values in the master dataset are often *inviolable*
  - Master dataset's variable labels, value labels, and other attributes are maintained, although the storage types are automatically adjusted if necessary.
- Non-matched variables are included

Append Example

```
clear all
use http://www.stata-press.com/data/r13/odd1
append using http://www.stata-press/data/r13/even
list
```

	odd	number	even
1.	1	1	.
2.	3	2	.
3.	5	3	.
4.	7	4	.
5.	9	5	.
6.	.	6	12
7.	.	7	14
8.	.	8	16

One-to-One Match Merge

- Variables from corresponding observations determined by the key variable(s) are joined to form observations containing variables from both the dataset stored on disk (the using dataset) and variables from the dataset in memory (the master dataset)
- Syntax: `merge 1:1 varlist using filename`
- Master data are inviolable: if a variable already exists in the master dataset, its values are not replaced by values from the using dataset
- By default, merge creates a new variable, _merge, which contains numeric codes concerning the source and the contents of each observation in the new, merged dataset.

```
_merge values:
1 (master) originally appeared in master only
2 (using) originally appeared in using only
3 (match) originally appeared in both
```

One-to-One Match Merge Example

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

```
capture drop _merge
merge 1:1 id using "using file name", report
drop _merge
```

Many-to-One Match Merge Example

Master

id	region	a
1	2	26
2	1	29
3	2	22
4	3	21
5	1	24
6	5	20

Using

region	x
1	15
2	13
3	12
4	11

merge m:1 region using "using_file_name"

id	region	a	x	_merge
1	2	26	13	3
2	1	29	15	3
3	2	22	13	3
4	3	21	12	3
5	1	24	15	3
6	5	20	.	1
.	4	.	11	2

```
capture drop _merge
merge m:1 region using "using file name", report keepusing(region x)
drop _merge
```

How Stata Merges

- "Remember this formal definition. It will serve you well" (Stata 12 Manual, Data Management [D], p.438)

"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."

Working with Raw Data

- Stata stores dta in a proprietary format, i.e. the .dta file
- Once data are stored in a .dta file, it can quickly be loaded into memory via the "use" command
- Data in other formats need to be converted into Stata format
- One such other format is known as raw data, which Stata assumes is sotred in a file with a .raw extension

Data Import Commands

- For an overview of the commands that import (or conert) data into Stata format: help import
- Stata's flagship input command to read raw data is infile, which can deal with both:
  - Free Format Data:
    - Values are delimited (by a space, tab, or comma)
    - String value are quoted if they contain spaces or commas
  - Fixed format data:
    - Values are not delimited by appear in fixed columns
- For simple free format files, may use insheet command
- For simple fixed format files, may use infix command
- Both infile and infix allow using a separate *dictionary* file

Free Format

- test.raw file looks like:

```
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
```

```
infile str14 country setting effort change using test.raw, clear
```

Fixed Column Format

- test.raw can also be read as a fixed format file, since the values for each variable appear in fixed columns:
  - country names are always in columns 4-17
  - settings values are always in columns 23-24
  - effort values are always in columns 31-32
  - change values are always in columns 40-41
- Column specifications can be separately stored in a 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, data can be imported into Stata format:

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

Lessons About Importing/Exporting Data

- Stat/Transfer can import/export data to/from various formats
- But don't blindly trust any piece of software that moves data from one system/package/application to another
- It helps to know both systems/packages/applications well
- Be careful and double-check everything