



SECTION FOR CLINICAL EPIDEMIOLOGY & BIOSTATISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University

Pitfall in BIG data management

Bunyarit Sukrat, MD

CEB Special Workshop 18/11/2016

OUTLINES

- Working with oversized datasets
- Working with do-files
- Repeating similar commands
- Organizing your work

WORKING WITH OVERSIZED DATASETS

- Working memory
- Machines to run Stata
- Stata versions
- Import files
- Combine dataset

WORKING WITH OVERSIZED DATASETS

- **Working memory**

- ▶ Stata loads the entire data file into the working memory (RAM)
- ▶ Quick access to the data
- ▶ Physical limit to the size of the dataset
 - ▶ Your computer should contain **50% more memory than the size of your largest dataset**

SECTION FOR CLINICAL EPIDEMIOLOGY & BIOSTATISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University

← Settings

⚙ Home

Find a setting

System

Display

Apps & features

Default apps

Notifications & actions

Power & sleep

Storage

Offline maps

Tablet mode

Multitasking

Projecting to this PC

Apps for websites

About



Windows 10

PC name

DESKTOP-3U1MR2O

Rename PC

Organization

WORKGROUP

Connect to work or school

Edition

Windows 10 Home

Version

1607

OS Build

14393.447

Product ID

00326-10003-65453-AA345

Processor

Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz 4.01 GHz

Installed RAM

64.0 GB

System type

64-bit operating system, x64-based processor

Pen and touch

No pen or touch input is available for this display

Change product key or upgrade your edition of Windows

Read the Privacy Statement for Windows and Microsoft services

Read the Microsoft Services Agreement that applies to our services

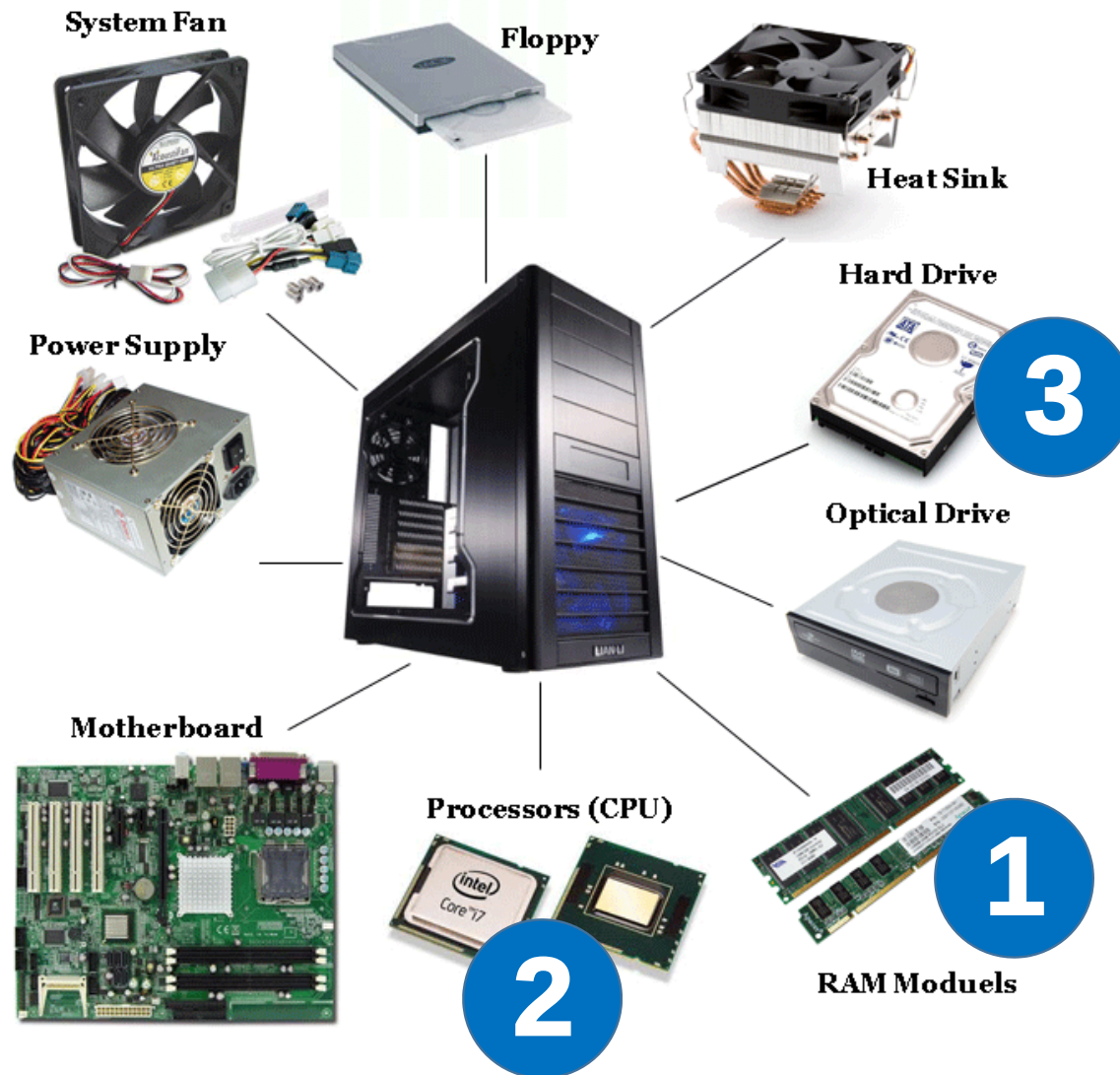
Read the Microsoft Software License Terms

WORKING WITH OVERSIZED DATASETS

- **Machines to run Stata**
- **Stata versions**

SECTION FOR CLINICAL EPIDEMIOLOGY & BIostatISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University



SECTION FOR CLINICAL EPIDEMIOLOGY & BIOSTATISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University



- 1) Clock speed (GHz)
- 2) Physical cores

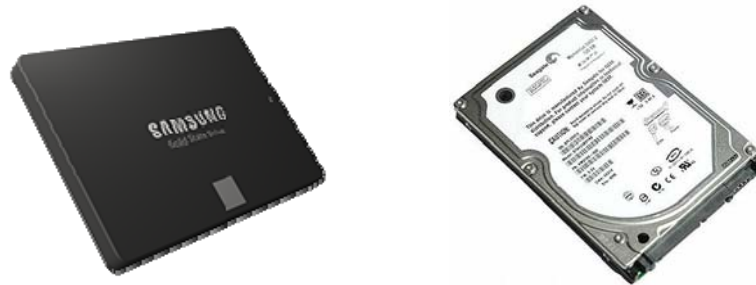
Compare features

Package	Max. no. of variables	Max. no. of right-hand variables	Max. no. of observations	64-bit version available?	Fastest: designed for parallel processing?	Platforms
Stata/MP	32,767	10,998	20 billion*	Yes	Yes	Windows, Mac, or Unix
Stata/SE	32,767	10,998	2.14 billion	Yes	No	Windows, Mac, or Unix
Stata/IC	2,047	798	2.14 billion	Yes	No	Windows, Mac, or Unix
Small Stata	99	98	1,200	Yes	No	Windows, Mac, or Unix

*The maximum number of observations is limited by the amount of available RAM on your system. Stata/MP can theoretically analyze up to 281 trillion observations, but current hardware memory limitations don't yet allow that many.

SECTION FOR CLINICAL EPIDEMIOLOGY & BIostatISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University



Category	840 SSD (500GB)	2.5" SATA HDD (500GB, 7200rpm)	Difference
Media	NAND FLASH	Magnetic Platters	
Seq. R/W Speed (MB/s)	540 / 330	60 / 160 (*140/70)	x 3~8 / 2~5
Ran R/W Speed (IOPS)	98,000 / 70,000	450/400	217 / 175
Data Access Time (ms)	0.1	10~12	x100~120
Benchmark Score (PCMark Vantage)	78,700	5,600	x 14
Power Consumption (Operation)	0.127W (active)	1.75W	x 13 ↓
Idle Power	0.046W (Idle)	0.8W	x 17 ↓
Vibration	20G (10~2000Hz)	0.5G (22~350Hz)	x 40
Shock (Operation)	1500G/0.5ms	350G/2.0ms	x 4
Reliability (MTBF*)	1.5M hours	700k hours	x 2

* HDD performance may vary by brand and model. The above data is for explanatory purposes only.

WORKING WITH OVERSIZED DATASETS

• Import files

Table 11.1. Filename extensions used by statistical packages

Extension	Origin
.dta	Stata
.odf	OpenOffice Calc
.por	SPSS (portable)
.rda, .rdata	R
.sas7bdat	SAS dataset
.sav	SPSS
.xls, .xlsx	Excel
.xpt	SAS (portable)

WORKING WITH OVERSIZED DATASETS

• Import files

- ▶ ASCII text files : **infile**, **insheet**, and **infix**
- ▶ Spreadsheet format
 - ▶ Each observation is written into a separate row
 - ▶ The columns (variables) are separated by delimiters
- ▶ **Tab-delimited** with the file extension .txt
- ▶ Comma-separated with the extension .csv



WORKING WITH OVERSIZED DATASETS

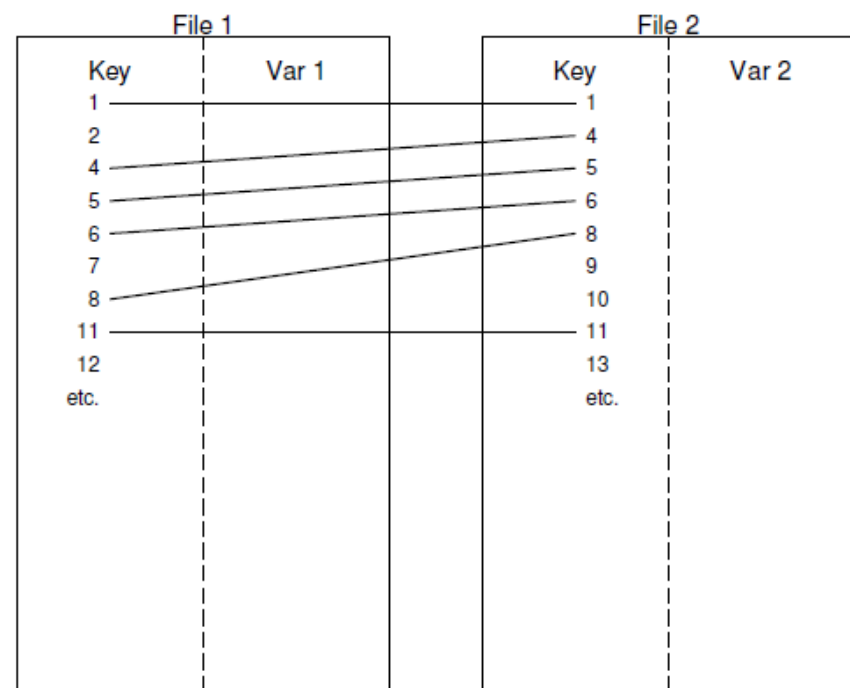
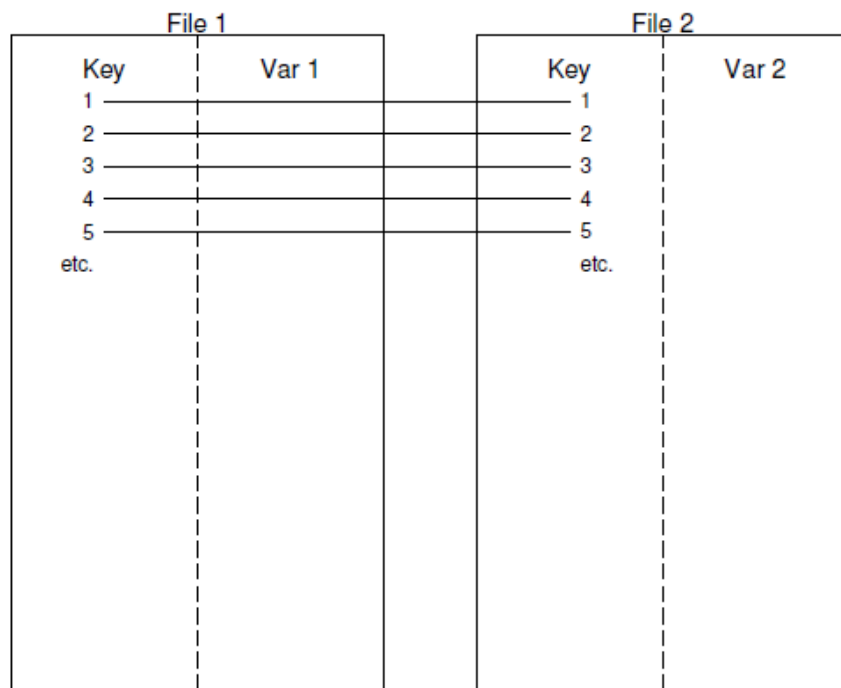
- **Import files**

- ▶ **Common pitfalls**

- ▶ Commas are used in numbers as thousands separators
 - ▶ Sometimes missing values are coded as blanks or dots
 - ▶ If characters other than commas or tabs were used to separate cells, you must use the delimiter option.
 - ▶ You cannot use **insheet** if the values of any observations span across multiple rows.

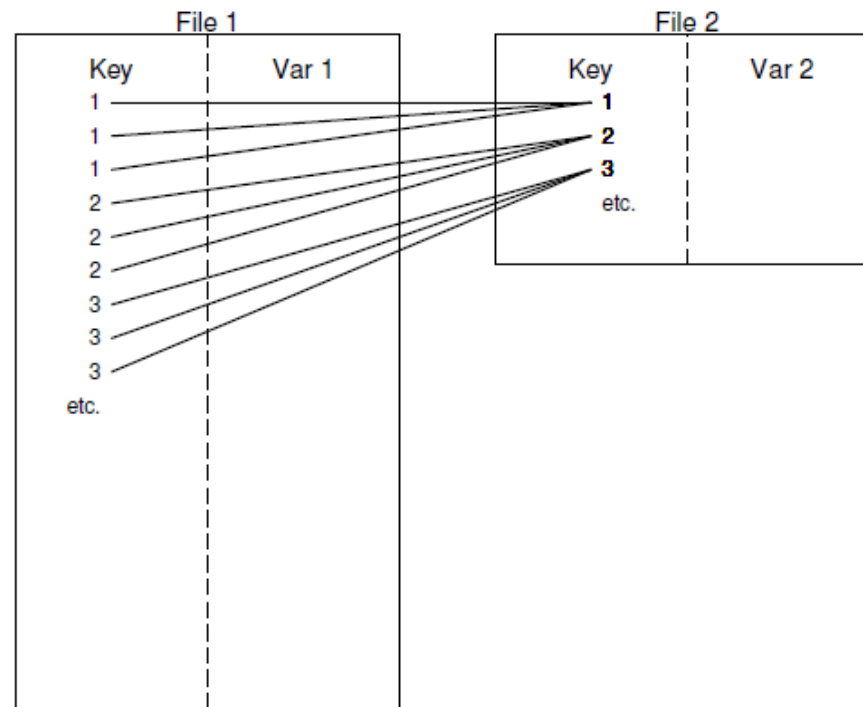
WORKING WITH OVERSIZED DATASETS

• Combine dataset merge 1:1



WORKING WITH OVERSIZED DATASETS

- Combine dataset **merge m:1**



WORKING WITH OVERSIZED DATASETS

- Combine dataset **append**

File 1

Var 1	Var 2
0	2002
1	1876
0	3000
0	2130
1	1000
etc.	etc.

File 2

Var 1	Var 2	Var 3
0	1238	7
1	1500	9
etc.	etc.	etc.

WORKING WITH OVERSIZED DATASETS

- Combine dataset

- ▶ Common pitfalls

m:m merges

m:m specifies a many-to-many merge and is a bad idea. In an m:m merge, observations are matched within equal values of the key variable(s), with the first observation being matched to the first; the second, to the second; and so on. If the master and using have an unequal number of observations within the group, then the last observation of the shorter group is used repeatedly to match with subsequent observations of the longer group. Thus m:m merges are dependent on the current sort order—something which should never happen.

Because m:m merges are such a bad idea, we are not going to show you an example. If you think that you need an m:m merge, then you probably need to work with your data so that you can use a 1:m or m:1 merge. Tips for this are given in *Troubleshooting m:m merges* below.

SECTION FOR CLINICAL EPIDEMIOLOGY & BIostatISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University

joinby

	family_id	child_id	x1	x2
1.	1025	3	11	320
2.	1025	1	12	300
3.	1025	4	10	275
4.	1026	2	13	280
5.	1027	5	15	210

	family_id	parent_id	x1	x3	child_id	x2
1.	1025	12	27	721	1	300
2.	1025	12	27	721	4	275
3.	1025	12	27	721	3	320
4.	1025	11	20	643	4	275
5.	1025	11	20	643	1	300
6.	1025	11	20	643	3	320
7.	1026	13	30	760	2	280
8.	1026	14	26	668	2	280

	family_id	parent_id	x1	x3
1.	1025	11	20	643
2.	1025	12	27	721
3.	1026	14	26	668
4.	1026	13	30	760
5.	1030	15	32	684
6.	1030	10	39	600

merge m:m

	family_id	parent_id	x1	x3	child_id	x2	_merge
1.	1025	11	20	643	3	320	matched (3)
2.	1025	12	27	721	1	300	matched (3)
3.	1026	14	26	668	2	280	matched (3)
4.	1026	13	30	760	2	280	matched (3)
5.	1030	10	39	600	.	.	master only (1)
6.	1030	15	32	684	.	.	master only (1)
7.	1025	12	27	721	4	275	matched (3)
8.	1027	.	15	.	5	210	using only (2)

WORKING WITH OVERSIZED DATASETS

• Combine dataset

▶ Common pitfalls

▶ Useful merge option

options	Description
Options	
<code>keepusing(varlist)</code>	variables to keep from using data; default is all
<code>generate(newvar)</code>	name of new variable to mark merge results; default is <code>_merge</code>
<code>nogenerate</code>	do not create <code>_merge</code> variable
<code>nolabel</code>	do not copy value-label definitions from using
<code>nonotes</code>	do not copy notes from using
<code>update</code>	update missing values of same-named variables in master with values from using
<code>replace</code>	replace all values of same-named variables in master with nonmissing values from using (requires <code>update</code>)
<code>noreport</code>	do not display match result summary table
<code>force</code>	allow string/numeric variable type mismatch without error
Results	
<code>assert(results)</code>	specify required match results
<code>keep(results)</code>	specify which match results to keep
<code>sorted</code>	do not sort; datasets already sorted

WORKING WITH DO-FILES

- Naming variables and do-files
- Designing do-files
- Comments
- Line breaks
- Crucial commands

WORKING WITH DO-FILES

- Naming variables

- ▶ *“Never change a variable unless you give it a new name”*

- ▶ Sequential naming systems

- ▶ e.g. v1 v2 v3

- ▶ Source naming systems

- ▶ e.g. q1a q1b q1c

- ▶ Mnemonic naming systems

- ▶ e.g. age ht bp kids

WORKING WITH DO-FILES

- Naming variables (**A-Z**, **a-z**, **0-9**, **_**)
 - ▶ Up to 32 characters long
 - ▶ Restrict to 14 characters for **describe** command
 - ▶ Cannot begin with number
 - ▶ The following names are not allowed

`_all`
`_b`
`byte`
`_coef`
`_cons`

`double`
`float`
`if`
`in`
`int`

`long`
`_n`
`_N`
`_pi`
`_pred`

`_rc`
`_skip`
`str#`
`using`
`with`

WORKING WITH DO-FILES

- Naming variables (**A-Z**, **a-z**, **0-9**, **_**)
 - ▶ Use simple, unambiguous names
 - ▶ Use shorter names
 - ▶ Use clear and consistent abbreviations
 - ▶ Use names that convey content
 - ▶ Be careful with capitalization

WORKING WITH DO-FILES

- Naming do-files

- ▶ The alphabetized names indicate the run order
- ▶ Use names that remind you of what is in the file
- ▶ Anticipate revising your do-files and adding new do-files.
- ▶ Choose names that are easy to type.
 - ▶ Avoid too long names with special characteristics

Project [-task] step [letter] [version] [description].do

WORKING WITH DO-FILES

- Designing do-files
 - ▶ Make do-files self-contained
 - ▶ Use version control
 - ▶ Exclude directory information
 - ▶ Make do-files easy reading
 - ▶ Comment, alignment and indentation, line break

WORKING WITH DO-FILES

- Designing do-files

- ▶ **Comments**

- ▶ * at the beginning of the line
 - ▶ // add comments within a line
 - ▶ /* */ Stata ignores everything between /* and */

WORKING WITH DO-FILES

- Designing do-files

- ▶ **Line breaks**

- ▶ Stata command can be very long.
 - ▶ But avoiding long lines makes your do-file more readable.
 - ▶ It is a good idea to restrict lines to 75–80 characters.
 - ▶ `///` command continues on the next line.
 - ▶ `#delimit` defines the character that indicates the end of a command.

WORKING WITH DO-FILES

- Designing do-files

- ▶ **Crucial commands**

- ▶ *Some commands are recommended in all do-files.*

- ▶ version 14

- ▶ set more off

- ▶ capture log close

- ▶ log using *example1*, replace

- ▶ Log close

- ▶ exit

REPEATING SIMILAR COMMANDS

- Macros
- Information return by Stata commands
- Loop
 - ▶ The **foreach** loop
 - ▶ The **forvalues** loop

REPEATING SIMILAR COMMANDS

- **Macros**
- A macro assigns a string of text or a number to an abbreviation.
- Local and global macros

REPEATING SIMILAR COMMANDS

- **Information return by Stata commands**
- `return list` (R class command)
- `ereturn list` (E class command)
- Scalars and Matrix

REPEATING SIMILAR COMMANDS

- **foreach loop**
- The foreach loop is used to repeat a specific task for each element of a list.
- Single or series of commands

```
foreach lname listtype list {  
    commands  
}
```

REPEATING SIMILAR COMMANDS

- **foreach loop**

```
foreach lname listtype list {  
    commands  
}
```

- of `varlist` for lists of variables
- of `newlist` for lists of new variables
- of `numlist` for lists of numbers
- `in` for arbitrary lists of letters, words, and numbers separated by spaces

REPEATING SIMILAR COMMANDS

- **forvalues loop**
- A shorter way to set up a foreach loop with of numlist as the *listtype*.
- Single or series of commands

```
forvalues lname = range {  
    commands  
}
```

ORGANIZING YOUR WORK

• Objectives

- ▶ Important files are not lost
- ▶ You can find your do-files for a particular result again without problems
- ▶ All steps are clearly documented
- ▶ *All analyses can be reproduced easily*

ORGANIZING YOUR WORK

• Suggestion

- ▶ Master do file
 - ▶ Creation do-files
 - ▶ **Don't** use to carry out an analysis
 - ▶ Analysis do-files.
 - ▶ **Don't** save a dataset with an analysis do-file
- ▶ Project log book

SECTION FOR CLINICAL EPIDEMIOLOGY & BIOSTATISTICS

Faculty of Medicine, Ramathibodi Hospital, Mahidol University

THANK YOU

For your kind attention...