Lecture 3: Some Advanced Tips for Stata

Yi Chen

ShanghaiTech University

2021

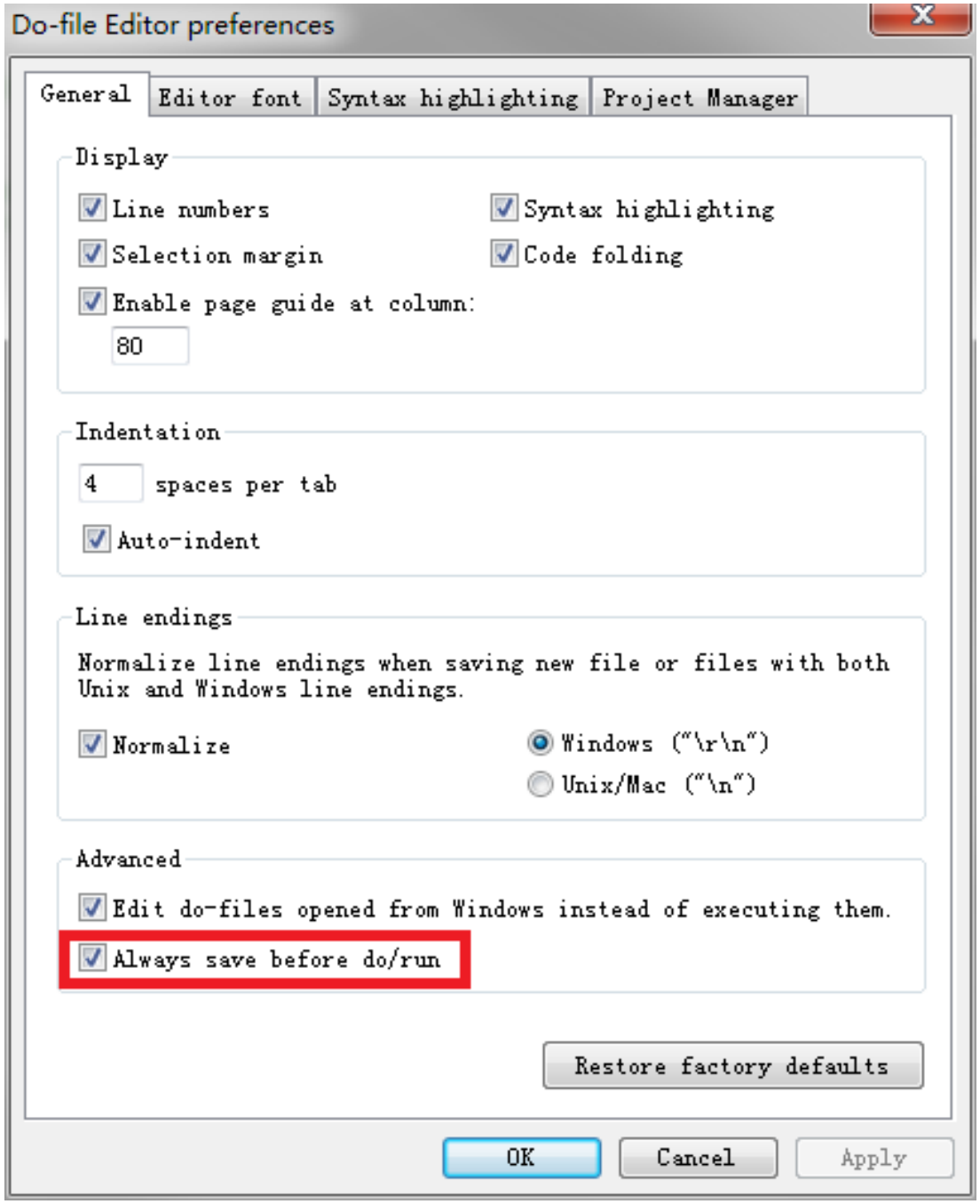
Outline

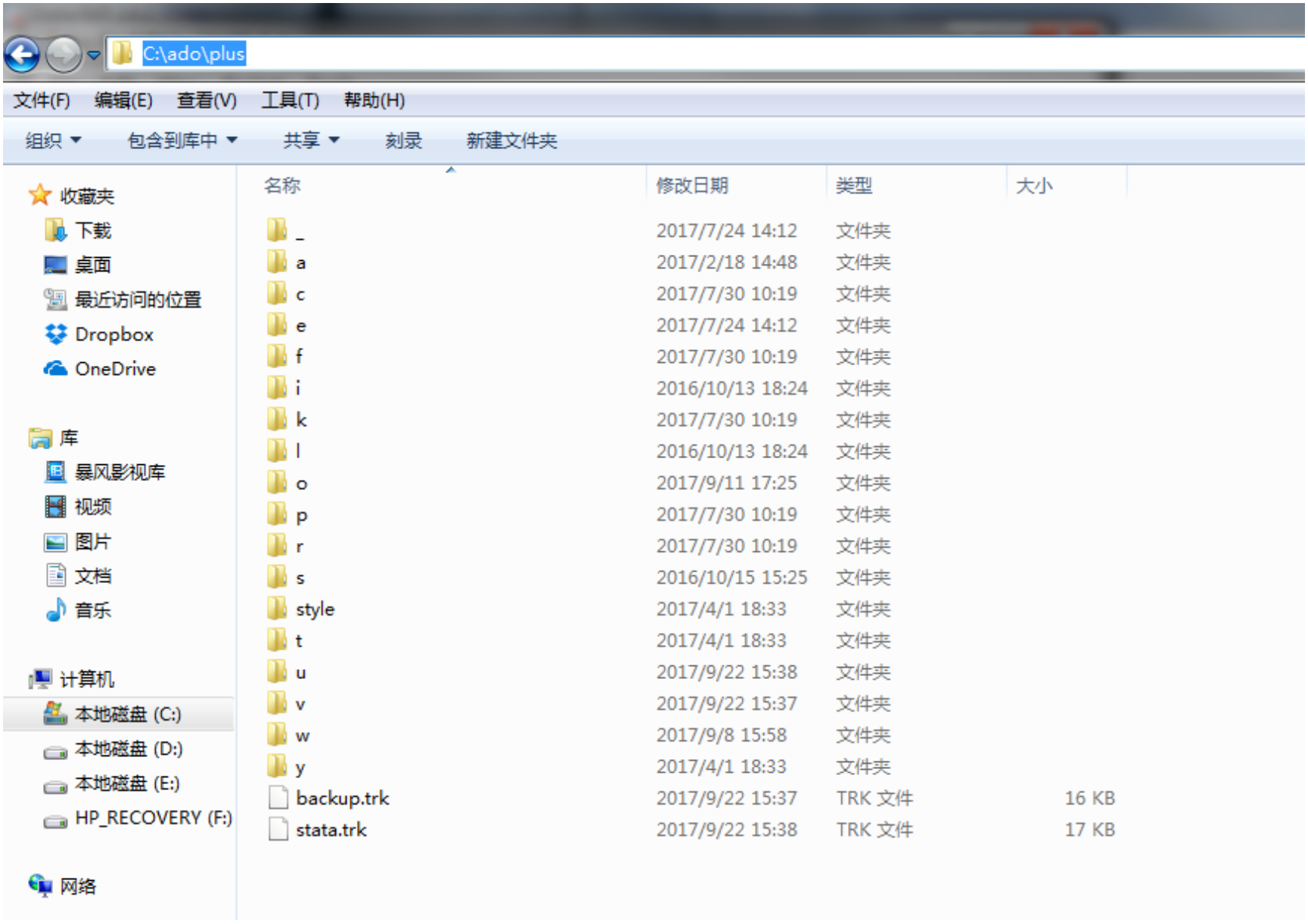
1. (Advanced) Tips for Stata
2. Nine Principles of Writing a Do-file Well
3. More Programming Details

Outline

Auto-save in Stata

- VERY IMPORTANT! Stata do-file editor does not have an auto-save feature.
  - Do-file Editor – Edit – Preference – General – "Always save before do/run"
  - That is why the cloud storage introduced in the last lecture can be helpful.
  - Note that it's still not auto-saving.
- If you are using other text editors (e.g., Sublime Text 3), also check out auto-save first!
- If you are moving from your old computer to a new computer, it is highly recommended to copy the external program. The default path is "C:\ado\plus"


Do-file Editor preferences
General
Editor font
Syntax highlighting
Project Manager
Display
Line numbers
Syntax highlighting
Selection margin
Code folding
Enable page guide at column:
80
Indentation
4
spaces per tab
Auto-indent
Line endings
Normalize line endings when saving new file or files with both Unix and Windows line endings.
Normalize
Windows ("\r\n")
Unix/Mac ("\n")
Advanced
Edit do-files opened from Windows instead of executing them.
Always save before do/run
Restore factory defaults
OK
Cancel
Apply


C:\ado\plus
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 包含到库中 共享 刻录 新建文件夹
收藏夹
下载
桌面
最近访问的位置
Dropbox
OneDrive
库
暴风影视库
视频
图片
文档
音乐
计算机
本地磁盘 (C:)
本地磁盘 (D:)
本地磁盘 (E:)
HP_RECOVERY (F:)
网络
名称 修改日期 类型 大小
_ 2017/7/24 14:12 文件夹
a 2017/2/18 14:48 文件夹
c 2017/7/30 10:19 文件夹
e 2017/7/24 14:12 文件夹
f 2017/7/30 10:19 文件夹
i 2016/10/13 18:24 文件夹
k 2017/7/30 10:19 文件夹
l 2016/10/13 18:24 文件夹
o 2017/9/11 17:25 文件夹
p 2017/7/30 10:19 文件夹
r 2017/7/30 10:19 文件夹
s 2016/10/15 15:25 文件夹
style 2017/4/1 18:33 文件夹
t 2017/4/1 18:33 文件夹
u 2017/9/22 15:38 文件夹
v 2017/9/22 15:37 文件夹
w 2017/9/8 15:58 文件夹
y 2017/4/1 18:33 文件夹
backup.trk 2017/9/22 15:37 TRK 文件 16 KB
stata.trk 2017/9/22 15:38 TRK 文件 17 KB

Using profile.do

- If you have a list of commands that you are SURE that you wish to run EVERYTIME when you run Stata, you can put the codes in a file named "profile.do" and put it in the Stata directory.

profile.do

```
1   set type double
2   set more off, permanently
3
4   sysdir set PLUS "D:\Program Files\Stata 14\ado\plus"
```

- Detour: why double precision is important?
- Art of storage: no larger, no smaller

Useful Logical Function

- `inrange(z,a,b)`
  - 1 if $a \leq z \leq b$; otherwise, 0
  - Preferable to z>=a & z<=b because:
    1. Shorter and clearer
    2. Easier to specify the alternative `!inrange(z,a,b)`
- `inlist(z,a,b,...)`
- `missing(x1,x2,...,xn)`
  - Very useful when you have multiple set of control variables.
  - Be careful! Some Stata commands use comma to separate variables, others use space.

More Flexible Usages of foreach loop

While the usage of "forvalues" is generally fixed, the usage of "foreach" can be quite flexible:

- foreach *lname* in *any_list*
- foreach *lname* of local *lmacname*
- foreach *lname* of global *gmacname*
  - Do not put `' (or $)
- foreach *lname* of varlist *varlist*
  - *varlist* can be flexible, e.g. age-grade
- foreach *lname* of numlist *numlist*
  - *numlist* can be flexible, e.g. 1 4/8 13(2)21 103

Listing Multiple Variables

- `sum age race married never_married grade`
- `sum age-grade`
  - Pay VERY attention that the order of the variables are the same.
- `sum s*`
  - * = multiple symbols
- `sum ?a?e`
  - ? = single symbol

Very powerful if combined with: foreach *lname* of varlist *varlist*

Why foreach + varlist can be useful?

```
cd004_1_        byte    %8.0g       cd004_1_    How Often Do You Have Contact with Child 1
cd004_2_        byte    %8.0g       cd004_2_    How Often Do You Have Contact with Child 2
cd004_3_        byte    %8.0g       cd004_3_    How Often Do You Have Contact with Child 3
cd004_4_        byte    %8.0g       cd004_4_    How Often Do You Have Contact with Child 4
cd004_5_        byte    %8.0g       cd004_5_    How Often Do You Have Contact with Child 5
cd004_6_        byte    %8.0g       cd004_6_    How Often Do You Have Contact with Child 6
cd004_7_        byte    %8.0g       cd004_7_    How Often Do You Have Contact with Child 7
cd004_8_        byte    %8.0g       cd004_8_    How Often Do You Have Contact with Child 8
cd004_9_        byte    %8.0g       cd004_9_    How Often Do You Have Contact with Child 9
cd004_10_       byte    %8.0g       cd004_10_
                                                How Often Do You Have Contact with Child 10
cd004_11_       byte    %8.0g       cd004_11_
                                                How Often Do You Have Contact with Child 11
cd004_12_       byte    %8.0g       cd004_12_
                                                How Often Do You Have Contact with Child 12
cd004_13_       byte    %8.0g       cd004_13_
                                                How Often Do You Have Contact with Child 13
cd004_14_       byte    %8.0g       cd004_14_
                                                How Often Do You Have Contact with Child 14
cd004_15_       byte    %8.0g       cd004_15_
                                                How Often Do You Have Contact with Child 15
cd004_16_       byte    %8.0g       cd004_16_
                                                How Often Do You Have Contact with Child 16
a001_w3s1       byte    %8.0g       a001_w3     Whom do You Live Together
```

Why foreach + varlist can be useful?

```
a001_w3s1       byte    %8.0g       a001_w3     Whom do You Live Together
a001_w3s2       byte    %8.0g       a001_w3     Whom do You Live Together
a001_w3s3       byte    %8.0g       a001_w3     Whom do You Live Together
a001_w3s4       byte    %8.0g       a001_w3     Whom do You Live Together
a001_w3s5       byte    %8.0g       a001_w3     Whom do You Live Together
a001_w3s6       byte    %8.0g       a001_w3     Whom do You Live Together
a001_w3_0s1     byte    %8.0g       a001_w3_0
                                                Which Parents
```

(Optional) Sublime Text 3 + Stata

- As previously mentioned, Stata's build-in do-file editor is essentially a text file editor.
- Stata is clearly not an expert in text editing. Therefore, you may wish to use a more professional text editor, e.g., Sublime Text 3.
- Search "Sublime Text 3" + "Stata" (Reading Material 3.1)
- Important features include:
  - External plugins (e.g., Stata Editor for Sublime Text 3)
  - Can also be used for other programs, e.g., LaTeX, Matlab, R, Python.

ST3 Examples

1. Project management
2. Stata
   1. Auto-completion command/variable
   2. Group operation
3. LaTeX
   - Auto-completion command/citation
   - Multiple selection
   - Useful tools

The Tradeoff—To Learn or Not?

- ST3 does not come at a free price
  - Lump-sum set-up cost (non-trivial)
  - Changes in user habits
- The margnial benefits are larger for more frequent/efficient users.
- Personal recommendation for beginners
  - Stick to Stata/LaTeX themselves for now
  - Know the existence of ST3 (and other similar text editors), return to it when you want to pursue efficiency/convenience.

Outline

Writing a Do-file "Well"

- One requirement for this course is to use a software "nicely." (Reading Material 3.2, 3.3)
- An extreme example:

美程序员枪击4同事 竟因代码不写注释？

2018年09月23日 20:32 新浪财经-自媒体综合 新浪财经APP

原标题：美程序员枪击4同事，竟因代码不写注释？产品汪薪水高过码农

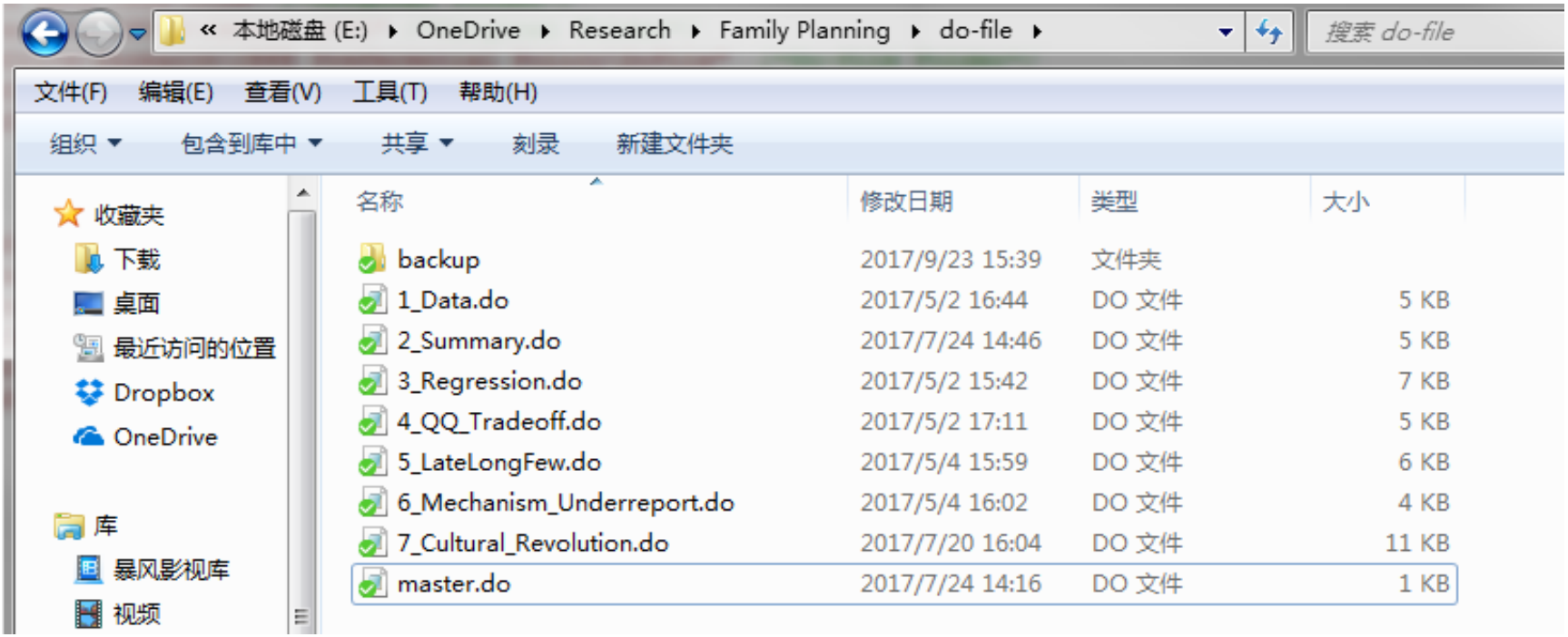
现实版"杀一名程序员祭天"在现实中上演。

9月19日，一名程序员在美国某办公楼向4名同事开枪，导致一人情况危机，两人伤情严重，一人被子弹擦伤。目前，凶手已死，身份被警方查明。

- In previous lectures, we already mentioned three principles:
  - Principle 1 Replication
  - Principle 2 Automation
  - Principle 3 Annotation

Principle 4: Organization

- Each do file should have its own purpose. Don't put everything in a huge do-file.
  - In many cases, we don't start from the beginning. e.g., codes used for cleaning the data would mostly remain untouched during the analysis.
- Because you have multiple do-files, number them to indicate the order.
- Also, organize well WITHIN each do-file. So you can easily pin down the place where you wish to modify.


« 本地磁盘 (E:) ▸ OneDrive ▸ Research ▸ Family Planning ▸ do-file ▸
搜索 do-file
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 包含到库中 共享 刻录 新建文件夹
收藏夹
下载
桌面
最近访问的位置
Dropbox
OneDrive
库
暴风影视库
视频
名称 修改日期 类型 大小
backup 2017/9/23 15:39 文件夹
1_Data.do 2017/5/2 16:44 DO 文件 5 KB
2_Summary.do 2017/7/24 14:46 DO 文件 5 KB
3_Regression.do 2017/5/2 15:42 DO 文件 7 KB
4_QQ_Tradeoff.do 2017/5/2 17:11 DO 文件 5 KB
5_LateLongFew.do 2017/5/4 15:59 DO 文件 6 KB
6_Mechanism_Underreport.do 2017/5/4 16:02 DO 文件 4 KB
7_Cultural_Revolution.do 2017/7/20 16:04 DO 文件 11 KB
master.do 2017/7/24 14:16 DO 文件 1 KB

Principle 5: Use a log File

- A log file is basically a text file that records everything in the result window.
- In almost all Stata textbooks, they encourage using log file . . . without telling you why.
  capture log close
  log using mylog1.log, text replace
- One often claimed purpose is recording.
  - But we wish to do it in a more explicit way (e.g., save graphs/tables).
- The log file is more helpful for "comparison."

```
global sysdate=c(current_date)
log using ``$path1\lecture3_$sysdate.txt'', text replace
```

Principle 6: Version Control

- Imagine what would happen if Word does not have an "undo" feature. An accidental "Enter" may ruin your life.
- Version control can be viewed as a global "undo" button: it provides a quick way to roll back changes you want to discard.
  - Recall the "Version History" feature in Dropbox.

- Another usage is for comparison purpose.
- There exist professional version control softwares.

Principle 7: Portability

- Have a "master" file. Its purpose is to make the program portable across computers.
  - Different co-authors can work on their own computer simply by changing path in the master file, without making any change to the other do-files.
- Different computers can differ in
  - Stata version
  - Path
  - External programs

master.do × Lecture 2.do × Untitled.do ×

```
1   clear all
2   version 14.1
3   set type double
4   set more off, permanently
5
6   local platform = 1 /*1 = Desktop, 2 = Laptop */
7   local install = 0
8
9   *Desktop
10  if (`platform' == 1) {
11  global path1  "E:\OneDrive\Research\CEEE Preferential Policy\Data"  /*Working Data
12  global path2A "E:\Data Sets\CEEE\CCSS" /*Original Folder*/
13  global path2B "E:\Data Sets\IPUMS\Census 2000" /*Original Folder*/
14  global path3  "E:\OneDrive\Research\CEEE Preferential Policy\DoFile"  /*Do-file Fo
15  global path4  "E:\OneDrive\Research\CEEE Preferential Policy\Output"  /*Output Fol
16  cd            "E:\OneDrive\Research\CEEE Preferential Policy\Data"
17  }
18
19  *Laptop
20  if (`platform' == 2) {
21  global path1  "D:\OneDrive\Research\CEEE Preferential Policy\Data"  /*Working Data
22  global path2A "C:\Data Sets\CEEE\CCSS" /*Original Folder*/
23  global path2B "C:\Data Sets\IPUMS\Census 2000" /*Original Folder*/
24  global path3  "D:\OneDrive\Research\CEEE Preferential Policy\DoFile"  /*Do-file Fo
25  global path4  "D:\OneDrive\Research\CEEE Preferential Policy\Output"  /*Output Fol
26  cd            "D:\OneDrive\Research\CEEE Preferential Policy\Data"
27  }
28
29  if (`install' == 1) {
30  ssc install reghdfe
31  ssc install winsor2
32  ssc install estout
33  }
34
```

Principle 8: Readability

- Use space properly
  - "gen t = hours + minutes/60 + seconds/3600" looks better than "gen t=hours+minutes / 60+seconds / 3600"
- Use the comment.
- Don't make the line too long.
- Abstraction

Break complicated algebraic calculations into pieces. Programming languages have no objection to definitions like

```
gen percap_gdp_real = ///
  (consumption + govt_expenditures + exports - imports - taxes) * ///
  10^6 / (price_index * pop_thousands * 1000)
```

or far longer ones. But a human may find it easier to parse the following:

```
gen gdp_millions_nominal = ///
  (consumption + govt_expenditures + exports - imports - taxes)
gen gdp_total_real = gdp_millions * 10^6 / price_index
gen pop_total = pop_thousands * 10^3
gen gdp_percap_real = gdp_total_real / pop_total
```

Principle 9: Efficiency (in Running Time)

- Pay attention to "slow" codes.
- Can we speed up the process?
  - e.g., regressions using large dataset such as census.
  - May consider reduce the data size by `compress` and by dropping redundant variables.
- Is it necessary to run those codes every time? (slow codes in a loop)

```
/* un-comment them if you want to re-define the drug category

****************************************
*Drug Name Matching 1: Perfect Match*
****************************************
insheet using "$path1\FDA\Application.txt",clear
keep applno chemical_type
gen NME = [chemical_type == 1]
gen new_drug = [chemical_type <= 5]
tempfile temp
save `temp',replace

insheet using "$path1\FDA\Product.txt",clear
keep applno drugname
duplicates drop
merge n:1 applno using `temp'
keep if new_drug==1
drop _merge
```

Outline

Scalar v.s. Local v.s. Global

We already know how to define and call a scalar/local/global.

Scalar v.s. Local:

- At the first glance, *scalar* seems to be a more convenient version of *local*
  - Note *scalar* can be used for both numbers and strings.
- The answer is quite surprising... We should try our best to avoid using *scalar*
  - The main problem of *scalar* is, the way of calling is exactly the same as a variable.
  - Don't be overconfident! Don't forget Stata allows for abbreviations. You could easily fall into the pitfall. (see the example in the do-file)

Local v.s. Global:

- Local is effective only during the execution of a do-file. Global is always effective unless Stata is closed.
- Global often results in unintended consequence.
- Nested global does not work so well. See the example in the do-file.

Personal Recommendations about Scalar/Local/Global

1. Use scalar only when the return value is a scalar
2. Use global only for defining global environment and storing the variable list
   - Use different names for different variable lists, e.g., `$var_regress1, $var_regress2, $var_iv`
   - If you really wish to use global, avoid repeatedly defining the same global.
3. Use local in all the other scenarios.

Tempfile

```
forvalues y = 1980/2014 {
use "$path2A\cepr_march_`y'.dta",clear
replace incp_all = incp_all - incp_int
keep hhid year wgt age incp_all

keep if age>=25&age<=100

tempfile year`y'
save `year`y'',replace
}

use `year1980',clear
forvalues y = 1981/2014 {
append using `year`y''
}
rename year YEAR
merge n:1 YEAR using "$path1\CPI_stata.dta",nogenerate
replace incp_all = incp_all*100/CPI
rename YEAR year
drop CPI
save "$path1A\CPS.dta",replace
```

Random Numbers

In Stata, there are two sets of functions related to random numbers.

- `help statistical function` calculates, PDF, CDF, and ICDF
- `help random number` generates a set of numbers that follow a certain distribution
- The second set of functions are used much more widely
  - e.g., generating a set of "placebo" shocks

Pseudorandomness

The concept of seed, `set seed`

- The machine generated random numbers are actually "pseudorandom" (read any textbook on numerical solution for details)
- From Wiki: "pseudorandom sequences typically exhibit statistical randomness while being generated by an entirely deterministic causal process."
- `set seed` in Stata is to guarantee the replicability. Everytime you run the code, you will get exactly the same sequence of "random" numbers.
  - One alternative is to store the random numbers in a separate file.

Matrix Basics

- Matrix is not the comparative advantage of Stata
  - Mainly used for creating your own program
  - Although Mata can do lots of matrix analysis, R and Matlab are more convenient alternatives.
- Usage 1: store information
  - `mkmat`, transits variables into matrix
  - `svmat`, transits matrix into variables
- Usage 2: extract information
  - Many return values are stored as matrices, such as `e(V)`. Sometime we wish to extract information from such matrices, such as variance-covariance matrix.

One Way of Presenting Matrix

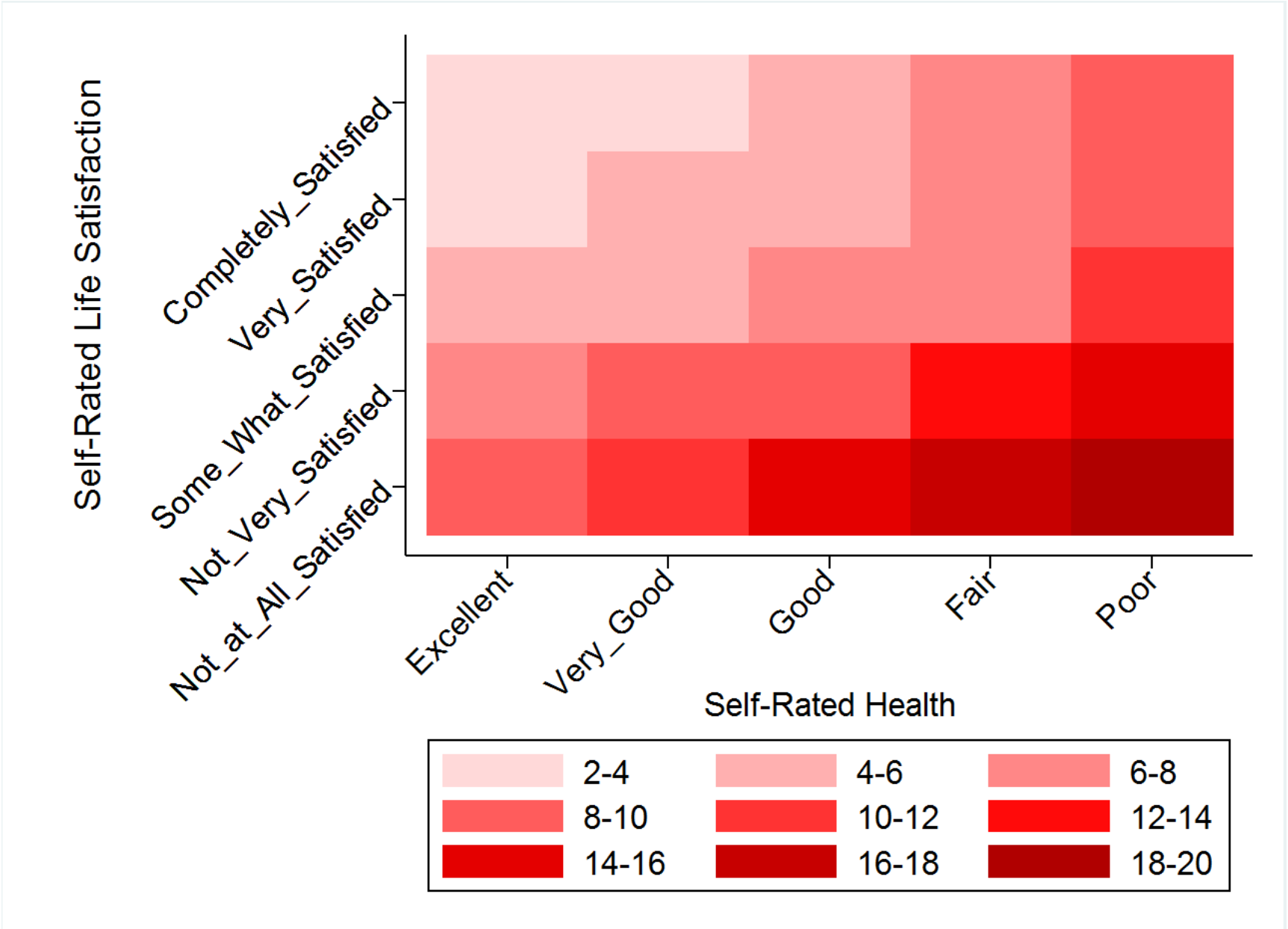


Table 1: Summary Statistics

	All	By Residence		By Gender	
		Urban	Rural	Male	Female
Basic Demographic Variables					
Age	9.86	10.17	9.75***	9.78	9.95°°
Male	0.53	0.51	0.54***		
Urban	0.26			0.25	0.27°°°
Height (cm)	129.66	132.73	128.56***	130.21	129.05°°°
Weight (kg)	30.68	32.75	29.93***	31.09	30.20°°°
Health-Indicating Variables					
Height for age z-score	-0.96	-0.73	-1.05***	-0.95	-0.98°
Weight for age z-score (age<10)	-0.28	-0.16	-0.32***	-0.26	-0.30°°
Family Background Variables					
Household income per capita[a] (yuan)	3509.64	4464.97	3166.66***	3573.15	3437.87°°°
Drinks tap water	0.62	0.88	0.53***	0.61	0.63°°
Uses a flush toilet at home	0.23	0.45	0.15***	0.23	0.23
Mother's age	36.42	36.57	36.37	36.35	36.51
Mother's height	155.59	155.93	155.46***	155.73	155.43°°°
Mother's years of education	6.46	7.89	5.96***	6.55	6.36°°°
Mother has ever smoked	0.02	0.02	0.02	0.02	0.02
Father's age	38.04	38.64	37.83***	37.92	38.19°°
Father's height	166.12	166.67	165.92***	166.13	166.12
Father's years of education	8.08	8.88	7.80***	8.12	8.03°
Father has ever smoked	0.70	0.71	0.70	0.68	0.72°°°
Medical Service Variables					
Has medical insurance	0.25	0.32	0.23***	0.26	0.25
Received preventive health service in the past four weeks	0.05	0.08	0.04***	0.05	0.05
Nutrition Intake Variables					
Daily protein intake (g)	54.21	57.50	53.00***	56.55	51.59°°°
Daily fat intake (g)	52.55	63.49	48.52***	54.41	50.46°°°
Daily calorie intake (g)	1837.96	1847.91	1834.29	1911.33	1755.86°°°
Daily carbohydrate intake (g)	286.99	261.58	296.34***	298.76	273.81°°°
Observations	17553	4638	12915	9311	8242

Notes:

Source: China Health and Nutrition Survey, 1991, 1993, 1997, 2000, 2004, 2006 and 2009. Age 0-17 if not specified. * indicate regional difference significant at 10%; ** significant at 5%; *** significant at 1%. °,°°,°°° refer to gender difference.

a. Income all inflated to 2009 CPI.

Challenge: Generate Table by Hand!

- A good practice: it involves almost all the programming elements we have learned so far.
- It is easy to generate "standard" regression tables using `estout` or `outreg`. But not all tables are standard.
- The way that using `estout` to generate nonstandard table is essentially the same as what we are going to introduce.

```
. foreach name of local rnames {
  2.     local ++i
  3.     local j 0
  4.     capture matrix drop b
  5.     capture matrix drop se
  6.     foreach model of local models {
  7.         local ++j
  8.         matrix tmp = C[`i', 2*`j'-1]
  9.         if tmp[1,1]<. {
 10.             matrix colnames tmp = `model'
 11.             matrix b = nullmat(b), tmp
 12.             matrix tmp[1,1] = C[`i', 2*`j']
 13.             matrix se = nullmat(se), tmp
 14.         }
 15.     }
 16.     ereturn post b
 17.     quietly estadd matrix se
 18.     eststo `name'
 19. }

. esttab, se mtitle noobs

------------------------------------------------------------
                      (1)             (2)             (3)
                   weight             mpg           _cons
------------------------------------------------------------
model1              2.044***                       -6.707
                   (0.377)                       (1174.4)

model2              1.747**         -49.51          1946.1
                   (0.641)         (86.16)         (3597.0)
------------------------------------------------------------
```

Another Example

```
reghdfe wage_expected_mean, absorb(发布城市 企业规模 企业类型)
scalar r1 = e(r2)

reghdfe wage_expected_mean, absorb(发布城市 企业规模 企业类型 学历要求 经验要求 管理经验)
scalar r2 = e(r2)

reghdfe wage_expected_mean, absorb(发布城市 企业规模 企业类型 学历要求 经验要求 管理经验 行业)
scalar r3 = e(r2)

reghdfe wage_expected_mean, absorb(发布城市 企业规模 企业类型 学历要求 经验要求 管理经验 行业 首要职位大类)
scalar r4 = e(r2)

reghdfe wage_expected_mean, absorb(发布城市 企业规模 企业类型 学历要求 经验要求 管理经验 行业 首要职位大类 首要职位小类)
scalar r5 = e(r2)

clear

gen square = .
set obs 5

forvalues i = 1/5 {
    replace square = r`i' in `i'
}

outsheet using "$path4\R_squared.txt", replace nonames
```

Another Example

Dependent Variable: Average Applicant's Expected Wage	
Control Variables	R-squared
City+Firm Size+Firm Type	0.1013
+ Job Requirement (Education, Experience, Management)	0.4245
+ Industry (52 categories)	0.4382
+ Broad Occupation (59 Categories)	0.4984
+ Detailed Occupation (588 Categories)	0.5942

- No matter how complicated a program is, you can generally proceed in three steps:
  - What input do I need?
  - How to organize the output?
  - Adjust the detail
- Input: sample mean (by group), T-test
- Output: each row represents one variable, each column represents one subsample
- Detail: observations, one blank line between two categories of variables