数据清洗常用技巧

小花经济学术

彭文威

MPhil in Social Science
Division of Social Science, HKUST

温州

2017 Chinese Stata Users Group Meeting

2017 年 8 月 19-20 日

Stata Group Meeting

彭文威

目录

Table of Contents

通过引用，自动填充。

list FamilyID ID in 10

replace FamilyID = FamilyID[_n-1] if FamilyID[_n]==.

Table of Contents

分组处理：bysort: 、egen

生成一个每个家庭人数的变量：
bysort Family: gen Num = _N
bysort Family sex : gen Num_bysex = _N
生成每个家庭的总收入：
egen Num_egen = count(ID),by(FamilyID)
egen Num_egen_bysex = count(ID),by(FamilyID sex)
生成每个家庭的总收入：
egen income_total= sum(income),by(FamilyID)
生成每个家庭的个人收入最高的收入变量：
egen income_total= max(income),by(FamilyID)

Table of Contents

return list

- return list
  - creturn list
  - ereturn list
  - sreturn list

ds 命令


```
ds // 列举出所有变量名称，并存在返回值里。
ds *ID*
return list
```

levelsof

```
levelsof FamilyID // 列举出变量的所有值，并存在返回值里。
return list
levelsof FamilyID ,local(fam)
foreach family in `fam'{
.....

}
```

local 扩展函数


```
local list : dir . dirs "*" makes a list of all subdirectories of the current directory
In list might be returned "notes" "subpanel".
eg:
local folderslist: dir . dir "*" // . 为当前工作路径
dis `""`folderslist'"' // 列举当前工作路径下的所有文件夹名称
```

local 扩展函数

```
local list : dir .files "*" makes a list of all regular files in the current directory.

eg:
local fileslist: dir . files "*.dta" // . 为当前工作路径
dis `""`fileslist'"' // 列举当前工作路径下的所有 dta 文件名称
```

循环

对当前路径下所有子文件夹的所有 dta 文件进行循环操作。

```
local folderslist: dir . dir "*"
foreach folder in `folderslist' {

local fileslist: dir `".\`folder'"' files "*.dta"
foreach file in `fileslist' {
use `file',clear
.....

}
}
```

循环

对当前路径下所有子文件夹的所有 dta 文件进行循环操作。

```
local folderslist: dir . dir "*"
foreach folder in `folderslist' {

local fileslist: dir `".\`folder'"' files "*.dta"
foreach file in `fileslist' {
use `file',clear
ds
foreach var in `r(varlist)'{
.....
}

}
}
```

Table of Contents

Table of Contents

assert

假设，每个家庭的成员 ID 是以年龄大小顺序排列的，比如，FamilyID 为 1 的家庭有 4 个成员，成员 ID 分别为 1、2、3、4。则成员的年龄排序为 1>2>3>4。当出现不连续的排序时，为数据录入错误，则要舍弃。数据量大，肉眼识别易错且效率低下。

```
bysort FamilyID: gen index= _n
assert index==ID
return list
```

assert 命令将对其后的表达式进行判断，并返回多少个判断错误。

datacheck

同 assert 命令，但功能更强大。假设，每个家庭的成员 ID 是以年龄大小顺序排列的，比如，FamilyID 为 1 的家庭有 4 个成员，成员 ID 分别为 1、2、3、4。则成员的年龄排序为 1>2>3>4。当出现不连续的排序时，为数据录入错误，则要舍弃。

```
bysort FamilyID: gen index= _n
assert index==ID
return list
```

datacheck 命令将对其后的表达式进行判断，返回多少个判断错误并列举。

Thank you!