CUSE：构建自己的 Stata 数据集仓库

程振兴

版本：*2.0.0*

更新：*2019* 年 *6* 月 *3* 日

摘　　要

几乎每个 Stata 的使用者都知道 sysuse、webuse 这些命令。它们能够非常方便的调用一些示例数据集。本文通过仿造 Mitchell (2012) 一书中提供的 vguse 命令的结构构建了自己的 Stata 数据集仓库。使用数据集仓库可以非常方便的存放和调用自己的一些示例和常用数据集，我将自己的数据仓库命名为 cuse。

1　导论

程序的使用演示经常需要一些示例数据集，存放在本地的示例数据集不便于分享，存放在服务器上的示例数据集虽然方便分享，但是使用的时候需要加上网址。例如我想介绍 Stata 的回归命令 regress （[R] **regress**）。我使用数据集 grilic_small.dta 进行演示，这个数据集的网址链接为：https://github.com/czxa/cuse/raw/master/g/grilic.dta。

regress 的用法示例：

```
use https://github.com/czxa/cuse/raw/master/g/grilic.dta, clear
regress lw s

*>       Source |       SS           df       MS      Number of obs   =       758
*> -------------+----------------------------------   F(1, 756)       =    255.70
*>        Model |  35.2039946         1  35.2039946   Prob > F        =    0.0000
*>     Residual |  104.082155       756  .137674809   R-squared       =    0.2527
*> -------------+----------------------------------   Adj R-squared   =    0.2518
*>        Total |   139.28615       757  .183997556   Root MSE        =    .37105
*> ------------------------------------------------------------------------------
*>           lw |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
*> -------------+----------------------------------------------------------------
*>            s |   .0966245   .0060425    15.99   0.000     .0847624    .1084866
*>        _cons |   4.391486   .0821136    53.48   0.000     4.230288    4.552684
*> ------------------------------------------------------------------------------
```

而如果 grilic_small.dta 是个系统数据集，就能使用 sysuse 命令调用了，而 grilic_small.dta 并不是一个系统数据集，所以使用 sysuse 调用会失败：

```
sysuse grilic_small, clear
*> file "grilic_small.dta" not found
*> r(601);
```

这里就体现出构建一个自己的数据集仓库的必要性了。于是就有了 cuse 命令。

2 cuse 命令包的结构

为了便于数据集的管理，我们首先需要建立名称分别为 a, b, c, ..., z 和 0 的文件夹，这些文件夹可以用来盛放不同字母开头的数据集。

然后我们需要编写一个 cuse 命令，代码如下：

```
*! 功能1: 调用本地仓库数据集
*! 功能2: 调用远端仓库数据集
*! 功能3: 将数据集存入系统文件夹中
cap prog drop cuse
prog define cuse
  version 14.0
  if `"`0'"' == "" {
      error 198
  }
  local 0 `"using `0'"'
  syntax using/ [, Clear Web Savetosystem]
  if "`web'" != ""{
      local url "https://github.com/czxa/cuse/raw/master"
  }
  else{
      local url "~/Documents/cuse"
  }
  local prefix = substr("`using'", 1, 1)
  use `"`url'/`prefix'/`using'"', `clear'
  if "`savetosystem'" != ""{
      local syspath "`c(sysdir_plus)'"
      save "`syspath'`prefix'/`using'", replace
  }
end
```

我为 cuse 命令设置了一个 web 选项，该选项的功能是对数据集的位置进行判断，因为该仓库文件夹在我的电脑上存放的位置是 ~/Documents/，同时该仓库托管在 GitHub 上，我的 GitHub 主页的链接为：https://github.com/czxa 我自己平时使用数据集的时候可以直接从本地读取数据，这样会比较快，然后把代码分享给别人的时候在所有的 cuse 语句后面添加 web 选项。也就是说：

```
cuse grilic_small, clear
* 等价于
use ~/Documents/cuse/g/grilic_small, clear

* 而
cuse grilic_small, clear web
* 等价于
use https://github.com/czxa/cuse/raw/master/g/grilic_small, clear
```

除此之外，我还设置了 savetosystem 选项，使用该选项可以把数据集保存成系统数据集，这样就可以通过 sysuse 命令使用了：

```
cuse grilic_small, clear web savetosystem
*> (note: file /Users/czx/Library/Application Support/Stata/ado/plus/g/grilic_small.
    dta not found)
*> file /Users/czx/Library/Application Support/Stata/ado/plus/g/grilic_small.dta
    saved

sysuse grilic_small, clear
```

为了记录 cuse 仓库里到底有那些数据集，我又写了个 cuselist 命令，这个命令通过下载一个临时的 cuselist_temp.ado 命令执行，命令仓库的作者在仓库中添加新数据之后可以通过修改 cuselist_temp.ado 文件进行记录。这样，使用者无需更新 cuse 命令，每次运行 cuselist 命令得到的数据集列表就是最新的。

```
*! 显示数据库中的所有数据集
cap prog drop cuselist
prog define cuselist
  copy "https://github.com/czxa/cuse/raw/master/cuselist_temp.ado" cuselist_temp.ado
      , replace
  cuselist_temp
end
```

cuselist_temp.ado 的代码目前为：

```
*! 显示数据库中的所有数据集
cap prog drop cuselist_temp
prog define cuselist_temp
  di "【0】"
  di "------------------------------------------------------------------------"
  di "1. 000001.dta: 平安银行历史股票数据"
  di "【a】"
  di "------------------------------------------------------------------------"
  di "1. amricancellmapdata.dta: 美国蜂窝地图各个省份的位置坐标"
  di "【c】"
  di "------------------------------------------------------------------------"
```

```
    di "1. cellmapdata.dta: 中国蜂窝地图各个省份的位置坐标"
    ......(此处省略了一些代码)
    di "2. tourism.dta: 旅游事业发展情况"
    di "-------------------------------------------------------------------------"
    di "【书籍数据集】"
    di "注意! 如果你想调用的数据集的名字里含大写字母, 你需要把它的首字母调成小写才能调用!"
    di "1. 《计量经济学及Stata应用》——陈强著"
    di "2. 《高级计量经济学及Stata应用》——陈强著"
    di "3. 《An Introduction to Stata Programming, Second Edition》——Christopher F. Baum著"
  end
```

最后再添加一些辅助文件，就搭建好了一个数据集仓库。

3 安装

我把该命令存放在了 GitHub 上。Stata 提供了一种安装外部命令的基础命令：net install，你可以在 Stata 的命令输出窗口输入下面的命令安装 cuse 命令：

```
net install cuse, from("https://www.czxa.top/cuse")
```

另外推荐使用 E. F. Haghish 开发的 github 命令安装：

首先你需要安装 github 命令：

```
net install github, from("https://haghish.github.io/github/")
```

然后就可以安装这个命令了：

```
github install czxa/cuse, replace
```

4 用法

```
cuse ["]filename["] [, clear web savetosystem ]
```

filename：是你想要调用的数据集名称，例如上面的 grilic_small。

1. *clear*：可以简写为 c。使用该选项时会先清空已有数据集再读入。
2. *web*：可以简写 w。使用该选项时表示从 GitHub 上读取数据。
3. *savetosystem*：可以简写 s。使用该选项时表示读入数据集后把该数据集存放在系统文件夹里。

5 用法示例

从本地文件夹读取 ctbc2.dta 数据集（只有自己能使用）：

```
cuse ctbc2, clear
*> (2002年-2018年中债国债到期收益率)
```

从 GitHub 上读取 ctbc2.dta 数据集：

```
cuse ctbc2, clear web
*> (2002年-2018年中债国债到期收益率)
```

查看当前数据集仓库中的所有可用数据：

```
 cuselist
 *> 【0】
 *> ------------------------------------------------------------------------
 *> 1. 000001.dta: 平安银行历史股票数据
 *> 【a】
 *> ------------------------------------------------------------------------
 *> 1. amricancellmapdata.dta: 美国蜂窝地图各个省份的位置坐标
 *> 【c】
 *> ------------------------------------------------------------------------
 *> 1. cellmapdata.dta: 中国蜂窝地图各个省份的位置坐标
 *> 1. countycode.dta: 中国各省市区县编号(即身份证前六位号码)
 ......(此处省略了一些代码结果)
 *> 【书籍数据集】
 *> 注意! 如果你想调用的数据集的名字里含大写字母, 你需要把它的首字母调成小写才能调用!
 *> 1. 《计量经济学及Stata应用》——陈强著
 *> 2. 《高级计量经济学及Stata应用》——陈强著
 *> 3. 《An Introduction to Stata Programming, Second Edition》——Christopher F.
    Baum著
```

参考文献

HAGHISH E F. github: a module for building, searching, installing, managing, and mining stata packages from github[EB/OL]. https://github.com/haghish/github.

MITCHELL M N, 2012. A visual guide to stata graphics[M/OL]. 3rd edition ed. Stata Press. https://www.stata-press.com/data/vgsg3.html.