

Stata 与 L^AT_EX 的完美结合

连玉君
中山大学 岭南学院
arlionn@163.com

2007. 08

目录

| | | |
|----------|-------------------------|-----------|
| 1 | stata.sty 宏包简介 | 2 |
| 1.1 | 安装 stata.sty 宏包 | 2 |
| 1.2 | 引用 Stata 手册 | 3 |
| 1.3 | Stata 语法 | 3 |
| 1.4 | 列示 Stata 源码 | 5 |
| 1.5 | 列示存储结果 | 7 |
| 1.6 | 范例和说明 | 8 |
| 1.7 | 特殊字符 | 8 |
| 2 | 列示表格 | 9 |
| 2.1 | 列示变量 — listtex 命令 | 10 |
| 2.2 | 列示相关矩阵 — corrtex 命令 | 10 |
| 2.3 | 列联表 — latab 和 tabout 命令 | 11 |
| 2.4 | 基本统计量 — latabstat 命令 | 11 |
| 2.5 | 列示估计结果 — esttab 命令 | 12 |
| 3 | 插入图形 | 17 |
| | 附录: Stata 与 Word 的结合 | 18 |

众所周知，采用 \LaTeX 输出的 pdf 文档比 Word 文档要美观，而在计量经济学领域，Stata 软件也因其开放的运行环境、高效的运算速度受到了越来越多的青睐。那么如何将这两个强大的工具结合起来，让 Stata 的输出结果完美的呈现在 \LaTeX 中呢？目前，已经有一系列 Stata 命令可以将 Stata 输出结果转换为 \LaTeX 代码，使二者有机地结合在一起。在这份短小的说明书中，我们精选了一些此类命令加以介绍，但愿能让 Stata 使用者们更通畅地走进 \LaTeX 的大门。

随后内容的结构安排如下：第 1 节介绍 Stata Journal 杂志社提供的 `statapress.cls` 类文档和相关的宏包的使用方法；第 2 节介绍一组将各种 Stata 表格转换为 \LaTeX 代码的命令；第 3 节介绍图形的插入；最后，在附录中，我们将简单介绍 Stata 与 Word 结合的几个命令。

1 stata.sty 宏包简介


`stata.sty` 是一个 \LaTeX 宏包，其中包含了一系列能够在 \LaTeX 文档中呈现 Stata 估计结果和语法格式原貌的命令 (macros) 和环境 (environments)。该宏包最初是专门为有意向 Stata Journal 投稿的作者设置的，因此利用该宏包输出的 pdf 文档完全可以达到正式出版物的质量。

1.1 安装 stata.sty 宏包

正式安装前，你只需要作如下准备：

- 确保你的电脑能够连接到互联网；
- Stata 7 或以上版本；
- 已安装 MiKTeX 软件以便生成 \LaTeX 文档。

安装的具体步骤如下：

- 在 Stata 命令窗口中输入 “`findit sjlatex`”，在弹出窗口中点击安装 **sjlatex**。需要特别强调的是，最好把上述文件安装在 LaTeX 安装目录下专门存放宏包的目录下。这会使接下来的安装变得非常简单，同时也便于让 Stata Journal 风格与 \LaTeX 文档正确连接。笔者便将其安装在如下目录中 “`C:\CTeX\texmf\tex\latex`”。

- 在 Stata 命令窗口中输入：

```
sjlatex install using C:\CTeX\texmf\tex\latex\sjlatex
```



- 更新 \LaTeX 。依次点击 MiKTeX 主菜单中的 **Accessories** → **MiKTeX** → **MiKTeX Options**，在弹出窗口中顺次点击 **Refresh Now** 和 **Update Now** 按钮。完成这两个步骤后，点击 **Apply** 按钮。

至此，我们就完成了安装程序，你只需要在 $\text{L}^{\text{T}}\text{E}^{\text{X}}$ 文档的导言区中加入 `\usepackage{stata}`，就可以在 $\text{L}^{\text{T}}\text{E}^{\text{X}}$ 文档中应用 *Stata Journal* 风格所需的命令了。

1.2 引用 Stata 手册

引用 Stata 手册所需的命令列示于表 1。

表 1: Stata 手册引用命令一览

| Example | Result |
|--|-------------------------------------|
| <code>\dref{merge}</code> | [D] merge |
| <code>\gref{graph}</code> | [G] graph |
| <code>\grefi{line_options}</code> | [G] <i>line options</i> |
| <code>\iref{data types}</code> | [I] data types |
| <code>\mreff{intro}</code> | [M-0] intro |
| <code>\mrefa{ado}</code> | [M-1] ado |
| <code>\mrefb{declarations}</code> | [M-2] declarations |
| <code>\mrefc{mata clear}</code> | [M-3] mata clear |
| <code>\mrefd{matrix}</code> | [M-4] matrix |
| <code>\mrefe{st_view(\\$, \\$)}</code> | [M-5] st.view() |
| <code>\mrefg{glossary}</code> | [M-6] glossary |
| <code>\mvref{cluster}</code> | [MV] cluster |
| <code>\pref{syntax}</code> | [P] syntax |
| <code>\rref{regress}</code> | [R] regress |
| <code>\stref{streg}</code> | [ST] streg |
| <code>\svyref{svy:~tabulate oneway}</code> | [SVY] svy: tabulate oneway |
| <code>\tsref{arima}</code> | [TS] arima |
| <code>\uref{1 Read this---it will help}</code> | [U] 1 Read this—it will help |
| <code>\xtref{xtreg}</code> | [XT] xtreg |

1.3 Stata 语法

下面是 `regress` 命令的语法格式：

```
regress depvar [indepvars] [if] [in] [weight] [, noconstant hascons tsscons
vce(vcetype) level(#) beta eform(string) noheader plus depname(varname) msel]
```

这可以通过如下代码生成：

```
\begin{stsyntax}
\dunderbar{reg}ress {\it depvar}
\optindepvars\
```

```

\optif\
\optin\
\optweight\
\optional{,
\underbar{noc}onstant
\underbar{h}ascons
tsscons
vce({\it vcetype\})
\underbar{l}evel(\num)
\underbar{b}eta
\underbar{ef}orm(\ststring)
\underbar{noh}ader
plus
\dunderbar{dep}name(\varname)
msel}
\end{stsyntax}

```

每个命令的语法格式都需要在一个单独的 `stsyntax` 环境中定义。表 2 列示了 `stata.sty` 宏包提供的用于定义语法格式的各种命令。

表 2: Stata 语法定义命令一览

| Macro | Result | Macro | Result |
|------------------------------|---------------------|-----------------------------|-------------------------|
| <code>\LB</code> | [| <code>\RB</code> |] |
| <code>\varname</code> | <i>varname</i> | <code>\optvarname</code> | [<i>varname</i>] |
| <code>\varlist</code> | <i>varlist</i> | <code>\optvarlist</code> | [<i>varlist</i>] |
| <code>\newvarname</code> | <i>newvarname</i> | <code>\optnewvarname</code> | [<i>newvarname</i>] |
| <code>\newvarlist</code> | <i>newvarlist</i> | <code>\optnewvarlist</code> | [<i>newvarlist</i>] |
| <code>\ifexp</code> | <i>if</i> | <code>\optif</code> | [<i>if</i>] |
| <code>\inrange</code> | <i>in</i> | <code>\optin</code> | [<i>in</i>] |
| <code>\eqexp</code> | <i>=exp</i> | <code>\opteqexp</code> | [<i>=exp</i>] |
| <code>\byvarlist</code> | by <i>varlist</i> : | <code>\optby</code> | [by <i>varlist</i> :] |
| <code>\optional{text}</code> | [<i>text</i>] | <code>\optweight</code> | [<i>weight</i>] |
| <code>\num</code> | # | <code>\optindepvars</code> | [<i>indepvars</i>] |
| <code>\ststring</code> | <i>string</i> | <code>\opttype</code> | [<i>type</i>] |

`\underbar` 和 `\dunderbar` 命令的差异可以从下例中看出:

- `{\tt \underbar{reg}ress}` generates regress
- `{\tt \dunderbar{reg}ress}` generates regress

在描述一个新命令时，我们可以采用 `\hangpara` 和 `\morehang` 命令生成类似于 Stata 手册中的段落样式。例如，

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **23.5 Specifying the width of confidence intervals**.

便是由如下代码生成的，

```
\hangpara
{\tt level(\num)} specifies the confidence level, as a percentage,
for confidence intervals. The default is {\tt level(95)} or as set by {\tt
set level}; see \uref{23.5 Specifying the width of confidence intervals}.
```

1.4 列示 Stata 源码

在撰写 Stata 讲义或向 *Stata Journal* 投稿时，通常要在 pdf 文件中呈现 Stata 命令和估计结果的原貌，这可以采用 `stlog` 环境来实现。例如，要呈现如下 Stata 源码，

```
sysuse auto, clear
sum price
list price if foreign == 1
reg price mpg weight length
```

我们只需要上述源码包含在 `stlog` 环境中即可：

```
\begin{stlog}
sysuse auto, clear
sum price
list price if foreign == 1
reg price mpg weight length
\end{stlog}
```

更多的情况下，我们还需要列示估计结果，如

```
. sysuse auto, clear
(1978 Automobile Data)

. regress mpg weight length
```

| Source | SS | df | MS | | | |
|----------|------------|----|------------|-----------------|--------|--|
| Model | 1616.08062 | 2 | 808.040312 | Number of obs = | 74 | |
| Residual | 827.378835 | 71 | 11.653223 | F(2, 71) = | 69.34 | |
| Total | 2443.45946 | 73 | 33.4720474 | Prob > F = | 0.0000 | |
| | | | | R-squared = | 0.6614 | |
| | | | | Adj R-squared = | 0.6519 | |
| | | | | Root MSE = | 3.4137 | |

| mpg | Coef. | Std. Err. | t | P> t | [95% Conf. Interval] | |
|--------|-----------|-----------|-------|-------|----------------------|-----------|
| weight | -.0038515 | .001586 | -2.43 | 0.018 | -.0070138 | -.0006891 |
| length | -.0795935 | .0553577 | -1.44 | 0.155 | -.1899736 | .0307867 |
| _cons | 47.88487 | 6.08787 | 7.87 | 0.000 | 35.746 | 60.02374 |

这需要两个步骤:

- (1) 在 **Stata** 中使用 `sjlog` 命令, 将 `log` 文件转化成 **TeX** 格式, 并输出相应的 **TeX** 文件。如, 运行下面这个 `output.do` 文件, 就会生成一个名为 `output1.log.tex` 的 **TeX** 文件, 并自动存放到 “`D:\CTeX\chps\Report\Stlog`” 目录下。

```
* output.do
version 9.0
cd D:\CTeX\chps\Report\Stlog
sjlog using output1, replace
sysuse auto, clear
regress mpg weight length
sjlog close, replace nolog
```

这里, `cd` 命令指定了 **Stata** 的运行路径, 使输出的 `output1` 文件自动存储到这个文件夹下, 以便随后插入 **LaTeX** 文档中。通常, 运行 `output.do` 会产生三个文件: `.smcl` 文件、`.log` 文件和 `.log.tex` 文件。我们需要的是最后一个文件。由于我们在 `sjlog close` 命令行加入了选项 `nolog`, 所以在本例中并不会产生 `.log` 文件。

- (2) 在 **LaTeX** 中使用 `stlog` 环境和 `input` 命令插入第一步生成的 `output1.log.tex` 文件,

```
\begin{stlog}
\input{output1.log.tex}\nullskip
\end{stlog}
```

这里, `nullskip` 命令用于调整 `log` 文本与上下文的间距。

在某些情况下, 我们需要略去部分估计结果的呈现, 或者让某些内容呈现在下一页中。下面的代码可以帮助我们理解其处理方式:

```
\begin{stlog}
. sysuse auto
(1978 Automobile Data)
{\smallskip}
. regress mpg weight
{\smallskip}
\oom
{\smallskip}
\cnp
\end{stlog}
```

这里, `\oom` 是一个暂元 (macro), 里面存放了一条简短的信息, 告诉读者随后的估计结果将被省略。暂元 `\cnp` 里存放的信息则告诉读者目前尚未显示完的结果将在下一页继续显示。

```
. sysuse auto
(1978 Automobile Data)

. regress mpg weight

(output omitted)
```

(Continued on next page)

除了使用上面介绍的 `stata.sty` 宏包提供的 `stlog` 环境和 `sjlog` 命令外，对于已经存在的 `.log` 文件，我们还可以采用 `dotex` 命令 (需要下载) 把它转化为 \TeX 格式的文件，并以 *Stata Journal* 风格呈现在 \LaTeX 文档中。而对于已经存在 `.txt`、`.doc` 等多种类型的文档，我们可以采用 `maketex` 命令把它转换成 \LaTeX 文件。¹

此外，在 `stlog` 环境字体的默认大小为 8-point。要改变字体有两个选择：一是采用 `\fontsize{#}{#}\selectfont` 更改字体的大小；二是采用 `[auto]` 选项将字体设定为当前字体大小，如 `\begin[auto]{stlog}`。

1.5 列示存储结果

在介绍一个新的命令时，往往需要呈现其存储结果，这可以通过 `stresults` 环境来实现。输出结果中包含四列：一、三两列为 **Stata** 存储结果的标示符号 (如, `r(N)`, `e(cmd)`)，二、四两列则是相应标示符号的简短说明。整体而言，存储结果又分为四种类型：单值 (**Scalars**)、暂元 (**Macros**)、矩阵 (**Matrices**) 和 函数 (**Functions**)，类型标示可以采用 `\stresultsgroup` 命令生成。作为实例，我们在这里列示了 `regress` 命令存储在 `e()` 中的结果：

Scalars

| | | | |
|----------------------|-----------------------------|-------------------------|-------------------------------|
| <code>e(N)</code> | number of observations | <code>e(F)</code> | <i>F</i> statistic |
| <code>e(mss)</code> | model sum of squares | <code>e(rmse)</code> | root mean squared error |
| <code>e(df_m)</code> | model degrees of freedom | <code>e(ll_r)</code> | log likelihood |
| <code>e(rss)</code> | residual sum of squares | <code>e(ll_r0)</code> | log likelihood, constant-only |
| <code>e(df_r)</code> | residual degrees of freedom | | model |
| <code>e(r2)</code> | <i>R</i> -squared | <code>e(N_clust)</code> | number of clusters |

Macros

| | | | |
|------------------------|-------------------------------------|--------------------------|--------------------------------------|
| <code>e(cmd)</code> | <code>regress</code> | <code>e(wexp)</code> | weight expression |
| <code>e(depvar)</code> | name of dependent variable | <code>e(clustvar)</code> | name of cluster variable |
| <code>e(model)</code> | <code>ols</code> or <code>iv</code> | <code>e(vcetype)</code> | covariance estimation method |
| <code>e(wtype)</code> | weight type | <code>e(predict)</code> | program used to implement predict |

Matrices

| | | | |
|-------------------|--------------------|-------------------|---|
| <code>e(b)</code> | coefficient vector | <code>e(V)</code> | variance-covariance matrix of the estimators |
|-------------------|--------------------|-------------------|---|

Functions

| | |
|------------------------|-------------------------|
| <code>e(sample)</code> | marks estimation sample |
|------------------------|-------------------------|

输出上述结果的代码如下：

```
\begin{stresults}
\stresultsgroup{Scalars} \
\stcmd{e(N)} & number of observations & \stcmd{e(F)} & $\scriptstyle F$ statistic\
\stcmd{e(mss)} & model sum of squares & \stcmd{e(rmse)} & root mean squared error \
\stcmd{e(df_m)} & model degrees of freedom & \stcmd{e(ll_r)} & log likelihood \
(... ..)
\end{stresults}
```

¹经测试，该命令实际上只是在上述文件的首尾进行简单的处理，使之能在 \LaTeX 中执行，因此输出效果欠佳。

1.6 范例和说明

使用 `stexample` 和 `sttech` 环境可以输出类似于 Stata 手册风格的“范例”和“说明”，例如，

▷ Example

这是一个简单的 Stata 范例，可以使用 `stexample` 环境产生。

◀

□ 范例

这是一个中文版的 Stata 范例，可以使用自己定义的 `stexamplec` 环境产生。具体做法为：参考 `stata.sty` 宏包中第 226-239 行代码，定义一个名为 `stexamplec` 的新环境，将“Example”修改为“范例”即可。

◀

▷ Example

`stexample` 环境的缩进量可以通过 `\setlength{\stexamplehskip}{#pt}` 命令来设定。本例的缩进效果便是通过将 # 设置为 0 实现的。

◀

□ Technical note

`sttech` 环境的缩进量默认值为 `-7.00749pt`，我们同样也可以采用 `\setlength` 命令进行修改，方法同上。

□

1.7 特殊字符

表 3 列示了一些常用特殊字符命令，除了 `\stcaret` 和 `\sttilde` 外，其它命令对应的字符均为固定宽度。

(Continued on next page)

表 3: 特殊字符

| Macro | Result | Macro | Result |
|---------------------------|----------------|--------------------------|----------------|
| <code>\stbackslash</code> | <code>\</code> | <code>\stforslash</code> | <code>/</code> |
| <code>\stcaret</code> | <code>^</code> | <code>\sttilde</code> | <code>~</code> |
| <code>\caret</code> | <code>^</code> | <code>\tytilde</code> | <code>~</code> |
| <code>\lbr</code> | <code>{</code> | <code>\rbr</code> | <code>}</code> |

2 列示表格

对于 **Stata** 而言，执行指令后输出的表格主要有两种，一种是统计分析结果的汇总，如 `sum`、`tabulate` 等指令；另一种是估计结果的汇总，如 `regress`、`est table` 等指令。本节主要介绍前一种表格的输出方式，后者则可以采用 `outtable` 命令加以处理，这里不再详加介绍。

鉴于表格的多样性，**Stata** 使用者们针对不同的表格类型，编写了一系列的指令将 **Stata** 输出的表格自动转换为 \LaTeX 格式。这些命令与 **Stata** 中的命令通常都有很好的对应关系，如 `sutex` 命令将 `summarize` 命令的输出结果转换为 \LaTeX 代码，`listtex` 命令主要处理 `list` 命令的输出结果，而 `estout` 命令则主要处理与回归结果有关的命令（如，`regress`、`est table`）。表 4 列示了一些常用命令，多数都不是 **Stata** 自带的，需要使用 `findit` 命令进行搜索和下载。这里仅介绍几个典型命令的使用方法，至于其它命令的使用详情可以参考相应的帮助文件。

表 4: **Stata** \rightarrow \LaTeX 表格相关命令一览

| 命令名称 | 简单描述 |
|------------------------------------|--|
| <code>corrtext^{AA}</code> | 将 <code>corr</code> 、 <code>pwcorr</code> 的输出结果转换为 \LaTeX 代码 |
| <code>estout^{AAA}</code> | 将 <code>reg</code> 、 <code>est table</code> 等命令的输出结果转换为 \LaTeX 代码 |
| <code>latabstat</code> | 将 <code>tabstat</code> 命令的输出结果转换为 \LaTeX 代码 |
| <code>listtex</code> | 将 <code>list</code> 命令的输出结果转换为 \LaTeX 代码 |
| <code>outtable</code> | 将 Stata 矩阵转换为 \LaTeX 代码 |
| <code>sutex</code> | 将 <code>summarize</code> 命令的输出结果转换为 \LaTeX 代码 |
| <code>tabout^{AAA}</code> | 将 <code>tabulate</code> 、 <code>tabstat</code> 、 <code>table</code> 命令的输出结果转换为 \LaTeX 代码。 ² |

¹ AAA 高度推荐，AA 中度推荐

² 由于 `tabout` 命令使用了 `mata` 语句，所以必须在 **Stata** 9.2 以上版本中才能执行。

2.1 列示变量 — listtex 命令

采用 listtex 命令可以把 Stata 命令 list 列示在荧屏上的结果转换成 L^AT_EX 代码。

```
#delimit;  
listtex make price mpg foreign in 1/6 using listex1, type  
rstyle(tabular) head("\begin{tabular}{rrrr}"  
    "\textit{Make}&\textit{Price}&\textit{Mpg}&\textit{Foreign}\\")  
foot("\end{tabular}");  
#delimit cr
```

执行上面的指令后, Stata 的结果窗口将显示如下 L^AT_EX 代码:

```
\begin{tabular}{rrrr}  
\textit{Make} & \textit{Price}&\textit{Mpg}&\textit{Foreign}\\  
Merc. Zephyr&3,291&20&Domestic\\  
Chev. Chevette&3,299&29&Domestic\\  
Chev. Monza&3,667&24&Domestic\\  
Toyota Corolla&3,748&31&Foreign\\  
Subaru&3,798&35&Foreign\\  
AMC Spirit&3,799&22&Domestic\\  
\end{tabular}
```

将这些代码粘贴入 L^AT_EX 文档中, 便可输出如下列表:

| <i>Make</i> | <i>Price</i> | <i>Mpg</i> | <i>Foreign</i> |
|----------------|--------------|------------|----------------|
| Merc. Zephyr | 3,291 | 20 | Domestic |
| Chev. Chevette | 3,299 | 29 | Domestic |
| Chev. Monza | 3,667 | 24 | Domestic |
| Toyota Corolla | 3,748 | 31 | Foreign |
| Subaru | 3,798 | 35 | Foreign |
| AMC Spirit | 3,799 | 22 | Domestic |

2.2 列示相关矩阵 — corrtex 命令

在论文中经常要列示主要变量的相关系数矩阵。corrtex 命令将 Stata 中的 corr (pwcorr) 命令输出的相关系数矩阵转换为 L^AT_EX 代码。相对于此前的 corrtab 以及 mkcorr 等功能相似的命令, 该命令增加了几个新的选项, 更符合 L^AT_EX 表格的特征。为了输出表 5, 我们先在 Stata 中执行如下命令:

```
corrtex price mpg weight length gear_ratio, file(corrtex1) ///  
    title(使用 \texttt{corrtex} 生成的表格) key(tab:corrtex1) sig replace
```

然后使用 input 命令将上面生成的 corrtex1.tex 文件包含在 L^AT_EX 文档中, 即可生成表 5。

```
\input{Stlog/corrtex1.tex}
```



```
latabstat valist [if] [in] [weight] [, statistics(statname [...]) by(varname) nototal
    missing nosep columns(variables|statistics) longstub labelwidth(#)
    varwidth(#) format([%format]) casewise save tf(string) replace append
    tx(string) caption(string) clabel(string) hwidth(#) ]
```

要得到表 6，可以先在 Stata 中执行如下指令：

```
latabstat price weight len, s(mean sd med min max) ///
    cap(使用 \stcmd{latabstat} 命令生成的表格) clabel(tab:latabstat1) ///
    columns(s) f(%9.2fc) hw(16) replace ///
    tf(D:\CTeX\chps\Report\Stlog\latabstat1)
```

继而将上面生成的 `latabstat.tex` 文件包含在 \LaTeX 文档中，即

```
\input{Stlog/latabstat1.tex}
```

我们需要强调两个问题：一是该命令将表格标题的字体设定为粗体 (`\textbf{}`)，不希望该设定的读者可以将 `latabstat.ado` 文件中的 `\textbf{'caption'}` 语句全部替换为 `\text{'caption'}`；二是该命令把表格的注释自动设定为 `''Source:$S_FN''`，其中，`$S_FN` 为 Stata 中的一个全局暂元 (macro)，其内容是当前数据的存储路径。对于大多数人来说，这个设定用处不大，并且在每次编译 \MiKTeX 时都会因为存储路径中包含 “\” 符号而报告错误。因此，建议把 `latabstat.ado` 文件中的 `$S_FN` 全部清除掉。表 6 便是通过对 `latabstat.ado` 文件进行上述两项修改后得到的。

表 6: 使用 `latabstat` 命令生成的表格

| variable | mean | sd | p50 | min | max | N |
|-----------------|-------------|-----------|------------|------------|------------|----------|
| price | 6,165.26 | 2,949.50 | 5,006.50 | 3,291.00 | 15,906.00 | 74.00 |
| weight | 3,019.46 | 777.19 | 3,190.00 | 1,760.00 | 4,840.00 | 74.00 |
| length | 187.93 | 22.27 | 192.50 | 142.00 | 233.00 | 74.00 |

Source:

与 `latabstat` 命令功能相似的还有 `sutex` 命令，它能够把 Stata 命令 `summerize` 输出的结果转换为 \LaTeX 代码。该命令的使用方法与 `latabstat` 非常相似，由于它能实现的功能后者都能实现，所以这里就不再详加介绍了。

2.5 列示估计结果 — `esttab` 命令

在论文中呈现的最重要的表格可能就是估计结果了，使用 Ben Jann 编写的 `esttab` 命令可以把所有回归分析命令，如 `regress`, `xtreg` 等的结果转换为 \LaTeX 代码。³事实上，把多个

³由于该命令使用了 `mata` 语言，所以必须在 Stata 9.2 以上版本中才能执行。

模型的估计结果呈现在一张表格中才是 `estout` 命令的最大优势。该命令可视为 `Stata` 中 `esttable` 命令的 \LaTeX 版孪生兄弟，也是 `Ben Jann` 此前编写的 `estout` 命令的升级版。

假设我们采用此前的 `auto.dta` 数据估计了三个模型，分别为 `model1`、`model2` 和 `model3`，命令如下：

```

qui gen mpg2 = mpg^2
qui reg price mpg length weight
est store model1
qui reg price mpg mpg2 length weight
est store model2
qui reg price mpg mpg2 length weight foreign
est store model3

```

接下来，我们采用 `esttab` 命令将这些结果转换为 \LaTeX 代码，`Stata` 命令如下：

```
esttab model1 model2 model3 using esttab1.tex, replace
```

采用 `input` 命令把生成的 `esttab1.tex` 文件包含在 \LaTeX 文档中，即可生成下表：

| | (1) | (2) | (3) |
|----------|--------------------|----------------------|---------------------|
| | price | price | price |
| mpg | -86.79 (-1.03) | -974.5** (-2.68) | -313.4 (-0.90) |
| length | -104.9* (-2.64) | -100.5* (-2.62) | -91.85** (-2.73) |
| weight | 4.365*** (3.74) | 3.385** (2.84) | 5.300*** (4.72) |
| mpg2 | | 16.54* (2.51) | 5.500 (0.88) |
| foreign | | | 3313.8*** (4.67) |
| ._cons | 14542.4* (2.47) | 27534.4*** (3.58) | 10435.7 (1.36) |
| <i>N</i> | 74 | 74 | 74 |

t statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

虽然这个表格已近非常接近我们在论文中所要呈现的面貌了，但 `esttab` 命令还提供了许多选项使我们可以进一步美化之。比如，我们可以进一步设定表格的标题、标签、模型的标题、参数估计值和标准误的格式、星号的含义，以及变量的标签等。下面是一个更完整的设定，输出效果见表 7。

```
#delimit ;
esttab model* using esttab2.tex, replace
    title("使用 \stcmd{esttab} 命令生成的表格"\label{tab:esttab2})
    mtitle("模型1" "模型2" "模型3")
    b(%6.3f) t(%6.3f) star(* 0.1 ** 0.05 *** 0.01) ar2
    coeqlabels(mpg2 "mpg2" _cons Constant);
#delimit cr
```

表 7: 使用 `esttab` 命令生成的表格

| | 模型1 | 模型2 | 模型3 |
|----------------------------|------------------------|-------------------------|------------------------|
| mpg | -86.789 (-1.034) | -974.470*** (-2.684) | -313.395 (-0.900) |
| length | -104.868** (-2.640) | -100.493** (-2.621) | -91.851*** (-2.729) |
| weight | 4.365*** (3.739) | 3.385*** (2.841) | 5.300*** (4.724) |
| mpg ² | | 16.543** (2.508) | 5.500 (0.881) |
| foreign | | | 3313.808*** (4.672) |
| Constant | 1.5e+04** (2.469) | 2.8e+04*** (3.582) | 1.0e+04 (1.361) |
| <i>N</i> | 74 | 74 | 74 |
| adj. <i>R</i> ² | 0.330 | 0.377 | 0.521 |

t statistics in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

对于需要同时在 \LaTeX 文档和 Word 文档呈现估计结果的读者而言，⁴只需对上述结果稍作修改便可生成 `-tab-` 分栏的结果，从而可以很方便地粘贴到 Excel 表格中进行处理。详见附录。事实上，`estout` 命令还可以把 Stata 的估计结果转换成 HTML (html) 格式，这给制作网页教程的读者提供了极大的方便。

⁴有些杂志社接受 \LaTeX 格式的论文初稿，而另一些杂志社则可能要求提供 Word 格式的文稿。

Ben Jann 还编写了另外两个命令：`eststo` 和 `estadd`。前者类似于 Stata 中的 `est store` 命令，后者则可以把自己计算的一些统计量存储到已经存在的模型中。在使用 `estout` 命令的同时配合使用这两个命令，可以使输出结果更加灵活。

与 `esttab` 功能相似的命令还有 `estout`、`outtable` 等命令，但所提供的功能和使用方便程度都远不及 `esttab` 命令。

对于使用 Stata 8 的读者而言，可以采用 `estout` 命令输出类似于 `esttab` 的效果，只是设定稍微复杂一些。下面是一个相对完整的实例，输出效果见表 8。

```
#delimit ;
estout model* using estout1.tex, style(tex) replace
title("使用 \stcmd{estout} 命令生成的表格")
prehead(\begin{table}[htbp]\caption{{\sc @title}\label{tab:estout1}}
\centering\medskip
\begin{tabular}{l*{@M}{r}}\hline)
posthead("\hline")
prefoot("\hline")
varlabels(_cons "Cons") legend
stats(N r2_a F,fmt(%6.0f %6.3f %6.2f) labels("N" "$\bar{R}^2$" "F 值"))
starlevels(* 0.1 ** 0.05 *** 0.01)
cells(b(star fmt(3)) t(par fmt(2)))
postfoot(\hline\end{tabular}\
{\parbox{0.@width\textwidth}{\small 注: ***, **, * 分别表示在 1%, 5%, 10% 水平
上显著。}})
\end{table)) ;
#delimit cr
```

我们注意到，相对于 `esttab` 命令，`estout` 命令需要我们自己设定表格的 L^AT_EX 表头 (`prehead`、`prefoot`) 和表尾 (`posthead`、`postfoot`)。类似于 `esttab` 命令，在输出 L^AT_EX 格式的同时，`estout` 命令也可以输出 Excel (`tab`) 格式和 HTML (`html`) 的结果，这可以通过 [`style`] 选项来实现。

(Continued on next page)

表 8: 使用 ESTOUT 命令生成的表格

| | model1 | model2 | model3 |
|-------------|-----------------------|------------------------|-----------------------|
| | b/t | b/t | b/t |
| mpg | -86.789 (-1.03) | -974.470*** (-2.68) | -313.395 (-0.90) |
| length | -104.868** (-2.64) | -100.493** (-2.62) | -91.851*** (-2.73) |
| weight | 4.365*** (3.74) | 3.385*** (2.84) | 5.300*** (4.72) |
| mpg2 | | 16.543** (2.51) | 5.500 (0.88) |
| foreign | | | 3313.808*** (4.67) |
| Cons | 14542.434** (2.47) | 27534.426*** (3.58) | 10435.700 (1.36) |
| N | 74 | 74 | 74 |
| \bar{R}^2 | 0.330 | 0.377 | 0.521 |
| F 值 | 12.98 | 12.04 | 16.91 |

注: ***, **, * 分别表示在 1%, 5%, 10% 水平上显著。

3 插入图形

采用 `graph2tex` 命令 (需要下载) 可以帮助我们快速地将 **Stata** 图形插入 **LaTeX** 文档中。该命令首先把 **Stata** 图形转换为 `.eps` 格式, 并同时生成 **LaTeX** 代码。执行如下指令后

```
cd D:\CTeX\Mytex
scatter price mpg
graph2tex, epsfile(scatter) caption(散点图) label(scatter) ht(3)
cd D:\stata9\ado\personal\Stlog
```

将在 **Stata** 结果窗口生成如下代码:

```
\begin{figure}[h]
\begin{centering}
\includegraphics[height=3in]{scatter}
\caption{散点图}
\label{fig:scatter}
\end{centering}
\end{figure}
```

将上述代码复制、粘贴入 **LaTeX** 文档即可生成图 3-1。

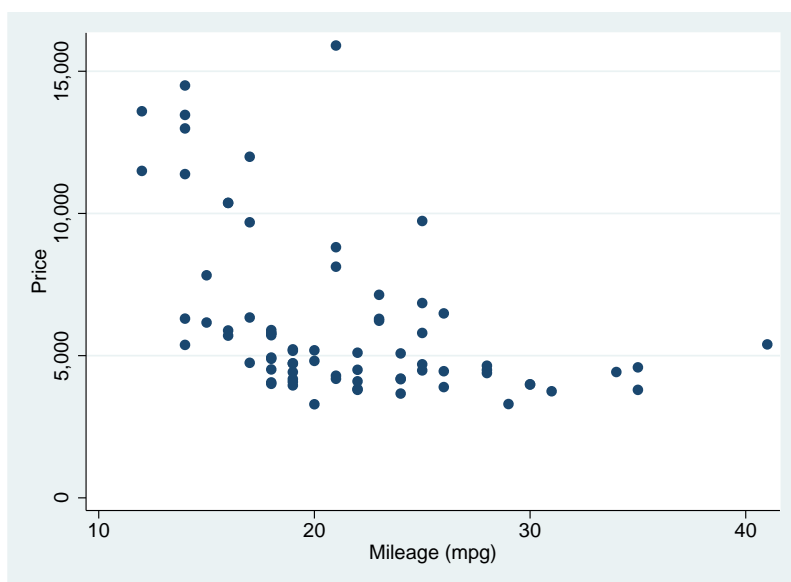


图 3-1: 散点图

细心的读者可能注意到, 在上面的 **Stata** 代码中, 我们两次使用了 `cd` 命令, 这非常重要。第一条 `cd` 命令保证随后执行 `graph2tex` 命令后生成的图形 `scatter.eps` 图形会自动存储到 **LaTeX** 的运行目录 `\Mytex` 下; 而最后一条 `cd` 指令则让 **Stata** 回到 `graph2tex` 命令执行之前的目录 `\Stlog` 下。由于 **Stata** 和 **MiKTeX** 分别在两个目录下运行, 所以上述细微的设定就显得非常必要, 否则在运行 **LaTeX** 过程中会返回如下错误:

```
!LaTeX Error: File `scatter' not found.
```

附录：Stata 与 Word 的结合

本附录简单介绍几个有效结合 Stata 和 Word 的命令。

估计结果的呈现 — esttab 命令

由 Ben Jann's 编写的 esttab 命令在 Stata 提供的 estimate table 命令基础上进行了许多改进，使我们可以非常快捷地把多个模型的估计结果呈现在一张表中，几乎不做任何修改就可以放到论文中。由于 esttab 命令输出的结果是以 -tab- 分栏的，所以可以很方便地从 Stata 的结果窗口中以表格方式 (Copy as table) 复制到 Excel 软体中，手动添加几条横线后即可插入 Word 文档。

下面是一个简单的实例。假设第 2.5 小节中估计的三个模型 (model11-model13) 都已存储在 Stata 中。执行如下指令：

```
esttab model*, ar2
```

Stata 荧屏上便会呈现出如下标准的期刊论文表格：

```
. esttab model*, ar2
```

| | (1) | (2) | (3) |
|-----------|--------------------|----------------------|---------------------|
| | price | price | price |
| mpg | -86.79 (-1.03) | -974.5** (-2.68) | -313.4 (-0.90) |
| length | -104.9* (-2.64) | -100.5* (-2.62) | -91.85** (-2.73) |
| weight | 4.365*** (3.74) | 3.385** (2.84) | 5.300*** (4.72) |
| mpg2 | | 16.54* (2.51) | 5.500 (0.88) |
| foreign | | | 3313.8*** (4.67) |
| _cons | 14542.4* (2.47) | 27534.4*** (3.58) | 10435.7 (1.36) |
| N | 74 | 74 | 74 |
| adj. R-sq | 0.330 | 0.377 | 0.521 |

t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001

有些期刊需要汇报参数估计的标准误 (s.e.)，而非 t 值，另外一些情况下，我们会分别用 *** 和 * 表示 1% 和 10% 显著水准，而 ** 则表示 5% 显著水准。即使将这两种要求都结合起来，命令也非常简单：

```
. esttab model*, ar2 se brackets star(* 0.1 ** 0.05 *** 0.01)
```

| | (1) | (2) | (3) |
|-----------|-----------------------|------------------------|----------------------|
| | price | price | price |
| mpg | -86.79 [83.94] | -974.5*** [363.1] | -313.4 [348.2] |
| length | -104.9** [39.72] | -100.5** [38.34] | -91.85*** [33.65] |
| weight | 4.365*** [1.167] | 3.385*** [1.192] | 5.300*** [1.122] |
| mpg2 | | 16.54** [6.596] | 5.500 [6.245] |
| foreign | | | 3313.8*** [709.3] |
| _cons | 14542.4** [5890.6] | 27534.4*** [7687.3] | 10435.7 [7667.3] |
| N | 74 | 74 | 74 |
| adj. R-sq | 0.330 | 0.377 | 0.521 |

Standard errors in brackets

* p<0.1, ** p<0.05, *** p<0.01

当然，esttab 命令的功能还远不止此，我们还可以列示出置信区间、AIC 值等统计量、对表格的标题、模型的标题进行设定，等等。如果进一步与 Ben Jann's 编写的另外两个命令 (eststo、estadd) 结合起来使用，输出效果将更加灵活，有兴趣的读者可以看这几个命令的帮助文件。下面是一个标准的论文格式的表格：

```
. #delimit ;
delimiter now ;
. esttab model*, replace
> title("表 10: 使用 esttab 命令生成的表格")
> mtitle("模型1" "模型2" "模型3") nonum
> b(%6.3f) t(%6.3f) star(* 0.1 ** 0.05 *** 0.01) ar2
> coeqlabels(mpg2 "mpg-square" _cons Constant);
```

(Continued on next page)

表 10: 使用 `esttab` 命令生成的表格

| | 模型1 | 模型2 | 模型3 |
|-----------|------------------------|-------------------------|------------------------|
| mpg | -86.789 (-1.034) | -974.470*** (-2.684) | -313.395 (-0.900) |
| length | -104.868** (-2.640) | -100.493** (-2.621) | -91.851*** (-2.729) |
| weight | 4.365*** (3.739) | 3.385*** (2.841) | 5.300*** (4.724) |
| mpg-sqare | | 16.543** (2.508) | 5.500 (0.881) |
| foreign | | | 3313.808*** (4.672) |
| Constant | 1.5e+04** (2.469) | 2.8e+04*** (3.582) | 1.0e+04 (1.361) |
| N | 74 | 74 | 74 |
| adj. R-sq | 0.330 | 0.377 | 0.521 |

t statistics in parentheses

* p<0.1, ** p<0.05, *** p<0.01

. #delimit cr

delimiter now cr