The Stata Journal (*yyyy*) *vv*, Number *ii*

Long title for title page of article insert

First author
First affiliation
City, State/Country
Email address

Second author
Second affiliation
City, State/Country
Email address

Abstract. This is an example article. You should change the `\input{}` line in `main.tex` to point to your file. If this is your first submission to the *Stata Journal*, please read the following "getting started" information.

Keywords: st0001, command name(s), keyword(s)

1 User's guide to sj.sty

The *Stata Journal* is produced using `statapress.cls` and `sj.sty`, a LaTeX 2_ε document class and package, respectively, each developed and maintained at StataCorp by the Stata Press staff. These files manage the look and feel of each article in the *Stata Journal*.

1.1 The title page

Each insert must begin with title-generating commands. For example,

```
\inserttype[st0001]{article}
\author{short author list}{%
  First author\\First affiliation\\City, State/Country\\Email address
  \and
  Second author\\Second affiliation\\City, State/Country\\Email address
}
\title[short toc title]{Long title for first page of journal insert}
\maketitle
```

Here `\inserttype` identifies the tag (for example, st0001) associated with the journal insert and the insert type (for example, article). The default `\inserttype` is "notag", possibly with a number appended. `\author` identifies the short and long versions of the list of authors (that is, J. M. Doe for the short title and John Michael Doe for the long). `\title` identifies the short (optional) and long (required) versions of the title of the journal insert. The optional argument to `\title` is used as the even-numbered page header. If the optional argument to `\title` is not given, the long title is used. The required argument to `\title` is placed in the table of contents with the short author list. Titles should not have any font changes or TeX macros in them. `\maketitle` must be the last command of this sequence; it uses the information given in the previous commands to generate the title for a new journal insert.

1.2 The abstract

The abstract is generated using the `abstract` environment. The `\keywords` are also appended to the abstract. Here is an example abstract with keywords:

```
\begin{abstract}
This is an example article.  You should change the \input{} line in
\texttt{main.tex} to point to your file.  If this is your first submission to
the {\sl Stata Journal}, please read the following ‘‘getting started’’
information.

\keywords{\inserttag, command name(s), keyword(s)}
\end{abstract}
```

`\inserttag` will be replaced automatically with the tag given in `\inserttype` (here st0001).

1.3 Sectioning

All sections are generated using the standard LaTeX sectioning commands: `\section`, `\subsection`, ...

Sections in articles are numbered. If the optional short section title is given, it will be put into bookmarks for the electronic version of the journal; otherwise, the long section title is used. Like article titles, section titles should not have any font changes or TeX macros in them.

1.4 The bib option

BibTeX is a program that formats citations and references according to a bibliographic style. The following two commands load the bibliographic style file for the *Stata Journal* (`sj.bst`) and open the database of bibliographic entries (`sj.bib`):

```
\bibliographystyle{sj}
\bibliography{sj}
```

Here are some example citations: **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, and **?**. They are generated by using the `\citet` and `\citet*` commands from the `natbib` package. Here we test `\citeb` and `\citebetal`: **?** [?], **?** [?], **?** [?], **?** [?], **?** [?], **?** [?], **?** [?], **?** [?], **?** [?], and **?** [?]. Sometimes using the `\cite` macros will result in an overfull line as shown above. The solution is to list the author names and the citation year separately, for example, `Ben-Akiva and Lerman [\citeyear{benAkivaLerman}]`.

The `bib` option of `statapress.sty` indicates that citations and references will be formatted using BibTeX and the `natbib` package. This option is the default (meaning that it need not be supplied), but there is no harm in supplying it to the `statapress` document class in the main LaTeX driver file (for example, `main.tex`).

```
\documentclass[bib]{sj}
```

If you choose not to use BibTeX, you can use the `nobib` option of `statapress.sty`.

```
\documentclass[nobib]{statapress}
```

BibTeX and bibliographic styles are described in **?**.

1.5 Author information

The *About the authors* section is generated by using the `aboutauthors` environment. There is also an `aboutauthor` environment for journal inserts by one author. For example,

```
\begin{aboutauthor}
Text giving background about the author goes in here.

\end{aboutauthor}
```

2 User's guide to stata.sty

`stata.sty` is a LaTeX package containing macros and environments to help authors produce documents containing Stata output and syntax diagrams.

2.1 Citing the Stata manuals

The macros for generating references to the Stata manuals are given in table **??**.

Table 1: Stata manual references

Example	Result
`\dref{merge}`	[D] **merge**
`\grefa{graph editor}`	[G-1] **graph editor**
`\grefb{graph}`	[G-2] **graph**
`\grefci{line\_options}`	[G-3] ***line_options***
`\grefdi{connectstyle}`	[G-4] ***connectstyle***
`\iref{data types}`	[I] **data types**
`\mreff{intro}`	[M-0] **intro**
`\mrefa{ado}`	[M-1] **ado**
`\mrefb{declarations}`	[M-2] **declarations**
`\mrefc{mata clear}`	[M-3] **mata clear**
`\mrefd{matrix}`	[M-4] **matrix**
`\mrefe{st\_view($\,$)}`	[M-5] **st_view( )**
`\mrefg{Glossary}`	[M-6] **Glossary**
`\miref{mi impute}`	[MI] **mi impute**
`\mvref{cluster}`	[MV] **cluster**
`\pref{syntax}`	[P] **syntax**
`\rref{regress}`	[R] **regress**
`\stref{streg}`	[ST] **streg**
`\svyref{svy:~tabulate oneway}`	[SVY] **svy: tabulate oneway**
`\tsref{arima}`	[TS] **arima**
`\uref{1~Read this---it will help}`	[U] **1 Read this—it will help**
`\xtref{xtreg}`	[XT] **xtreg**

2.2 Stata syntax

Here is an example syntax display:

`regress` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [`, noconstant hascons tsscons`
`vce(`*vcetype*`) level(`#`) beta eform(`*string*`) noheader plus`
`depname(`*varname*`) mse1`]

This syntax is generated by

```
\begin{stsyntax}
\dunderbar{reg}ress
    \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{nohe}ader
    plus
    \dunderbar{dep}name(\varname)
    mse1}
\end{stsyntax}
```

Each command should be formatted using a separate `stsyntax` environment. Table **??** contains an example of each syntax macro provided in `stata.sty`.

Table 2: Stata syntax elements

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

`\underbar` is a standard macro that generates underlines. The `\dunderbar` macro from `stata.sty` generates the underlines for words with descenders. For example,

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

The plain TEX macros `\it`, `\sl`, and `\tt` are also available. `\it` should be used to denote "replaceable" words, such as *varname*. `\sl` can be used for emphasis but should not be overused. `\tt` should be used to denote words that are to be typed, such as command names.

When describing the options of a new command, the `\hangpara` and `\morehang` commands provide a means to reproduce a paragraph style similar to that of the Stata reference manuals. For example,

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.7 Specifying the width of confidence intervals**.

was generated by

```
\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{20.7~Specifying the width of confidence intervals}.
```

2.3 Stata output

When submitting *Stata Journal* articles that contain Stata output, also submit a do-file and all relevant datasets that reproduce the output (do not forget to set the random-number seed when doing simulations). The following is an example of the `stlog` environment containing output from simple linear regression analysis on two variables in `auto.dta`:

```
. sysuse auto
(1978 Automobile Data)

. regress mpg weight

      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  1,    72) =  134.62
       Model |   1591.9902     1   1591.9902           Prob > F      =  0.0000
    Residual |  851.469256    72  11.8259619           R-squared     =  0.6515
-------------+------------------------------           Adj R-squared =  0.6467
       Total |  2443.45946    73  33.4720474           Root MSE      =  3.4389

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0060087   .0005179   -11.60   0.000    -.0070411   -.0049763
       _cons |   39.44028   1.614003    24.44   0.000     36.22283    42.65774
------------------------------------------------------------------------------
```

The above listing was included using

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

where `output1.log.tex` is a Stata log file converted to include TeX macros by using the `sjlog` command (more on `sjlog` shortly). `\nullskip` adjusts the spacing around the log file.

On occasion, it is convenient (maybe even necessary) to be able to omit some of the output or let it spill onto the next page. Here is a listing containing the details of the following discussion:

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

The `\oom` macro creates a short message indicating omitted output in the following example, and the `\clearpage` macro inserts a page break.

```
. sysuse auto
(1978 Automobile Data)

. regress mpg weight
```

  (*output omitted*)

The output in `output1.log.tex` was generated from the following `output.do`:

```
* output.do
set more off
capture log close

sjlog using output1, replace
sysuse auto
regress mpg weight
sjlog close, replace

sort weight
predict yhat
set scheme sj
scatter mpg yhat weight, c(. l) s(x i)
graph export output1.eps, replace

exit
```

`output.do` generates a `.smcl` file, `.log` file, and `.log.tex` file using `sjlog`. The actual file used in the above listing was generated by

```
. stlog type output.do
```

`sjlog.ado` is provided in the Stata package for `sjlatex`. `sjlog` is a Stata command that helps generate log output to be included in LATEX documents using the `stlog` environment. If you have installed the `sjlatex` package, see the help file for `sjlog` for more details. The lines that make up the table output from `regress` are generated from line-drawing macros defined in `stata.sty`; these were macros written using some font metrics defined in **?**.

By default, `stlog` sets an 8-point font for the log. Use the `auto` option to turn this behavior off, allowing you to use the current font size, or change it by using `\fontsize{#}{#}\selectfont`. The call to `stlog` with the `auto` option looks like `\begin[auto]{stlog}`.

Here is an example where we are using a 12-point font.

```
. stlog type output.do
```

2.4 About tables

Tables should be created using the standard LATEX methods. See **?** for a discussion and examples.

There are many user-written commands that produce LATEX output, including tables. Christopher F. Baum has written `outtable`, a Stata command for creating LATEX tables from Stata matrices. Ben Jann's well-known `estout` command can also produce LATEX output. To find other user-written commands that produce LATEX output, try

```
. net search latex
```

2.5 Encapsulated PostScript (EPS)

Figure **??** is included using `\epsfig` from the `epsfig` package.

```
\begin{figure}[h!]
\begin{center}
\epsfig{file=output1}
\end{center}
\caption{Scatterplot with simple linear regression line}
\label{fig}
\end{figure}
```

The graph was generated by running `output.do`, the do-file given in section **??**. The `epsfig` package is described in **?**.

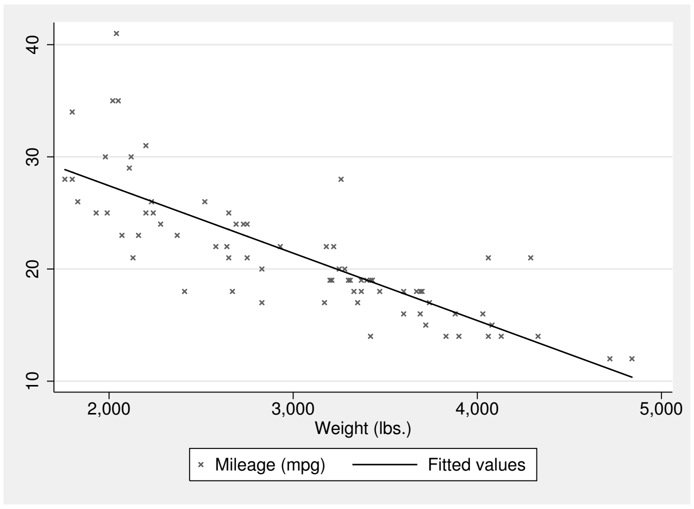

Figure 1: Scatterplot with simple linear regression line

2.6 Saved results

The `stresults` environment provides a table to describe the saved results of a Stata command. It consists of four columns: the first and third column are for Stata result identifiers (for example, `r(N)`, `e(cmd)`), and the second and fourth columns are for a brief description of the respective identifier. Each group of results is generated using the `\stresultsgroup` macro. The following is an example containing a brief description of the results that `regress` saved to `e()`:

Scalars			
`e(N)`	number of observations	`e(F)`	F statistic
`e(mss)`	model sum of squares	`e(rmse)`	root mean squared error
`e(df_m)`	model degrees of freedom	`e(ll_r)`	log likelihood
`e(rss)`	residual sum of squares	`e(ll_r0)`	log likelihood, constant-only model
`e(df_r)`	residual degrees of freedom		
`e(r2)`	R-squared	`e(N_clust)`	number of clusters
Macros			
`e(cmd)`	`regress`	`e(wexp)`	weight expression
`e(depvar)`	name of dependent variable	`e(clustvar)`	name of cluster variable
`e(model)`	`ols` or `iv`	`e(vcetype)`	title used to label Std. Err.
`e(wtype)`	weight type	`e(predict)`	program used to implement `predict`
Matrices			
`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators
Functions			
`e(sample)`	marks estimation sample		

2.7 Examples and notes

The following are environments for examples and notes similar to those given in the Stata reference manuals. They are generated using the `stexample` and `sttech` environments, respectively.

▷ **Example**

This is the default alignment for a Stata example.

◁

▷ **Example**

For this example, `\stexamplehskip` was set to `0.0pt` before beginning. This sentence is supposed to spill over to the next line, thus revealing that the first sentence was indented.

This sentence is supposed to show that new paragraphs are automatically indented (provided that `\parindent` is nonzero).

◁

❏ **Technical note**

For this note, `\sttechhskip` was set to `-13.90755pt` (the default) before beginning. This sentence is supposed to spill over to the next line, thus revealing that the first sentence was indented.

This sentence is supposed to show that new paragraphs are automatically indented (provided that `\parindent` is nonzero).

❏

2.8 Special characters

Table **??** contains macros that generate some useful characters in the typewriter (fixed width) font. The exceptions are `\stcaret` and `\sttilde`, which use the currently specified font; the strictly fixed-width versions are `\caret` and `\tytilde`, respectively.

Table 3: Special characters

Macro	Result	Macro	Result
`\stbackslash`	\	`\sttilde`	~
`\stforslash`	/	`\tytilde`	~
`\stcaret`	ˆ	`\lbr`	{
`\caret`	^	`\rbr`	}

2.9 Equations and formulas

In (**??**), $\overline{x}$ was generated using `\stbar{x}`. Here `\stbar` is equivalent to the TEX macro `\overline`.

$$E(\overline{x}) = \mu \tag{1}$$

In (**??**), $\widehat{\beta}$ was generated using `\sthat{\beta}`. Here `\sthat` is equivalent to the TEX macro `\widehat`.

$$V(\widehat{\beta}) = V\{(X'X)^{-1}X'y\} = (X'X)^{-1}X'V(y)X(X'X)^{-1} \tag{2}$$

2.10 Other miscellaneous macros and environments

The following box was created by

```
\begin{ttbox}
A special framed
box that obeys lines and spaces.
\end{ttbox}
```

```
A special framed
box that obeys lines and spaces.
```

The following box was created by

```
\ttboxWd=2.5in
\ttboxIndent=2em
\begin{ttbox}
Test that the width of the
box is \the\ttboxWd
and is indented \the\ttboxIndent
\end{ttbox}
```

```
Test that the width of the
box is 180.67499pt
and is indented 20.00003pt
```

About the authors

Some background information about the first author.

Some background information about the second author.