The Stata Journal (*yyyy*) *vv*, Number *ii*, pp.“ 1“endash21

Inference for imputed latent classes using multiple imputation

Stas Kolenikov
NORC
Columbia, Missouri, USA
kolenikov-stas@norc.org

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, postlca_class_predpute, latent class analysis, multiple imputation

1 Latent class analysis

Latent class analysis

Running example

```
. webuse gsem_lca1
. gsem (accident play insurance stock <- ), logit lclass(C 2)
```

Researchers are often interested in describing the latent classes or using these classes in analysis as predictors or as moderators. The official [SEM] **gsem postestimation** commands provide limited possibilities, namely reporting of the means of the dependent variables by class via `estat lcmean`. For nearly all meaningful applications of LCA, this is insufficient.

One possible approach is to predict the modal class for each observation, and use it in subsequent downstream analyses treating that as fixed:

The program distributed with the current package, `postlca_class_predpute`, provides a pathway for the appropriate statistical inference that would account for uncertainty in class prediction. This is achieved through the mechanics of multiple imputation (van Buuren 2018). The name is supposed to convey that

1. it is supposed to be run after LCA as a post-estimation command;
2. it predicts / imputes the latent classes.

2 The new command

Imputation of latent classes, a `gsem` postestimation command:


```
\begin{stsyntax}
    postlca\_class\_predpute,
    lcimpute(\varname)
    addm(\num)
    \optional{ seed(\num) }
\end{stsyntax}
```

`lcimpute(`*varname*`)` specifies the name of the latent class variable to be imputed. This option is required.

`addm(`#`)` specifies the number of imputations to be created. This option is required.

`seed(`#`)` specifies the random number seed.

3 Examples

3.1 Stata manual data set example

The LCA capabilities of Stata are exemplified in [SEM] **Example 50g**:

```
. frame change default
. cap frame gsem_lca1: clear
. cap frame drop gsem_lca1
. frame create gsem_lca1
. frame change gsem_lca1
.
. webuse gsem_lca1.dta, clear
(Latent class analysis)
. describe
Contains data from https://www.stata-press.com/data/r18/gsem_lca1.dta
 Observations:           216                  Latent class analysis
    Variables:             4                  17 Jan 2023 12:52
                                              (_dta has notes)
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
accident        byte    %9.0g                 Would testify against friend in accident case
play            byte    %9.0g                 Would give negative review of friend´s play
insurance       byte    %9.0g                 Would disclose health concerns to friend´s
                                                insurance company
stock           byte    %9.0g                 Would keep company secret from friend
-------------------------------------------------------------------------------
Sorted by: accident  play  insurance  stock
. gsem (accident play insurance stock <-), logit lclass(C 2)
  (output omitted)
Generalized structural equation model                     Number of obs = 216
Log likelihood = -504.46767
------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
1.C          |  (base outcome)
-------------+
```

```
2.C
       _cons |  -.9482041   .2886333    -3.29   0.001    -1.513915   -.3824933
```

```
Class:    1
Response: accident
Family:   Bernoulli
Link:     Logit
Response: play
Family:   Bernoulli
Link:     Logit
Response: insurance
Family:   Bernoulli
Link:     Logit
Response: stock
Family:   Bernoulli
Link:     Logit
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
accident						
_cons	.9128742	.1974695	4.62	0.000	.5258411	1.299907
play						
_cons	-.7099072	.2249096	-3.16	0.002	-1.150722	-.2690926
insurance						
_cons	-.6014307	.2123096	-2.83	0.005	-1.01755	-.1853115
stock						
_cons	-1.880142	.3337665	-5.63	0.000	-2.534312	-1.225972

```
Class:    2
```
(*output omitted*)

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
accident						
_cons	4.983017	3.745987	1.33	0.183	-2.358982	12.32502
play						
_cons	2.747366	1.165853	2.36	0.018	.4623372	5.032395
insurance						
_cons	2.534582	.9644841	2.63	0.009	.6442279	4.424936
stock						
_cons	1.203416	.5361735	2.24	0.025	.1525356	2.254297

One of the official post-estimation commands available after `gsem, lclass()` is the computation of the class-specific means of the outcome variables:

```
. set rmsg on
r; t=0.00 13:42:16
. estat lcprob
Latent class marginal probabilities                     Number of obs = 216
```

```
------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      [95% conf. interval]
-------------+----------------------------------------------------------------
           C |
           1 |   .7207539   .0580926      .5944743    .8196407
           2 |   .2792461   .0580926      .1803593    .4055257
------------------------------------------------------------------------------
r; t=1.21 13:42:17

. estat lcmean

Latent class marginal means                             Number of obs = 216

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      [95% conf. interval]
-------------+----------------------------------------------------------------
1            |
    accident |   .7135879   .0403588      .6285126    .7858194
        play |   .3296193   .0496984      .2403573    .4331299
   insurance |   .3540164   .0485528      .2655049    .4538042
       stock |   .1323726   .0383331      .0734875    .2268872
-------------+----------------------------------------------------------------
2            |
    accident |   .9931933   .0253243      .0863544    .9999956
        play |   .9397644   .0659957      .6135685    .9935191
   insurance |   .9265309   .0656538      .6557086    .9881667
       stock |    .769132   .0952072      .5380601    .9050206
------------------------------------------------------------------------------
r; t=5.26 13:42:23

. set rmsg off
```

The mutiple imputation version of this estimation task could look as follows:

```
. set rmsg on
r; t=0.00 13:42:23

. postlca_class_predpute, lcimpute(lclass) addm(10) seed(12345)
(216 missing values generated)
(10 imputations added; M = 10)
r; t=0.05 13:42:23

. mi estimate : prop lclass

Multiple-imputation estimates     Imputations     =         10
Proportion estimation             Number of obs   =        216
                                  Average RVI     =     0.4594
                                  Largest FMI     =     0.3319
                                  Complete DF     =        215
DF adjustment:   Small sample     DF:     min     =      55.99
                                          avg     =      55.99
Within VCE type:     Analytic             max     =      55.99

------------------------------------------------------------------------------
             |                                          Normal
             | Proportion   Std. err.      [95% conf. interval]
-------------+----------------------------------------------------------------
      lclass |
           1 |   .7236111   .0367281      .6500355    .7971867
           2 |   .2763889   .0367281      .2028133    .3499645
------------------------------------------------------------------------------
r; t=0.65 13:42:23
```

```
. mi estimate : mean accident, over(lclass)
Multiple-imputation estimates     Imputations     =         10
Mean estimation                   Number of obs   =        216
                                  Average RVI     =     0.3882
                                  Largest FMI     =     0.4485
                                  Complete DF     =        215
DF adjustment:   Small sample     DF:     min     =      35.59
                                          avg     =     116.62
Within VCE type:     Analytic             max     =     197.64
------------------------------------------------------------------------------
                             |       Mean   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
c.accident@lclass            |
                           1 |   .7144964   .0369935      .6415438    .7874491
                           2 |   .9934973   .0135709      .9659633    1.021031
------------------------------------------------------------------------------
Note: Numbers of observations in e(_N) vary among imputations.
r; t=0.72 13:42:24
. set rmsg off
```

The name of the latent class variable (here, `lclass`) and the number of imputations are required. The seed is optional, but of course is strongly recommended for reproducibility of the results, as the underlying data are randomly simulated. The multiple imputation version is notably faster.

3.2 NHANES complex survey data

In many important and realistic applications of LCA, including the case that necessitated the development of this package, the data come from complex survey designs that require setting the data up for the appropriate survey-design adjusted analyses. See [SVY] **svyset**, [MI] **mi svyset**, and Kolenikov and Pitblado (2014).

As one of many diagnostic outputs of MI, the increase in variances / standard errors due to imputations serves as an indication of how much of a problem would treating the singly imputed (e.g. modal probability) latent classes would have been

4 User's guide to sj.sty

The *Stata Journal* is produced using `statapress.cls` and `sj.sty`, a LaTeX 2$_\varepsilon$ 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*.

4.1 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}
\sjSetDOI{!!}
\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). The short author list is only the author initial(s) and last name, and the long author list is the author initial(s) and last name, author affiliation(s), and city and state or country (spelled out with accents applied as necessary). An email address should be included for, at least, the corresponding author. `\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. `\sjSetDOI{!!}` is filled in by Stata Press with a DOI. `\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.

4.2 The abstract

The abstract is generated using the `abstract` environment. The abstract states the purpose of the article and area of research. Abstracts must be able to stand alone from the full-text article. For this reason, fully cite references rather than merely supplying the author and date. Also, avoid introduction of acronyms in the abstract. 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). The first keyword will be the article tag (assigned by Stata Press); other keywords for indexing purposes should be added by the author(s). Community-contributed command names should be listed after the article tag. Plural terms and multiple concepts should be avoided.

4.3 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.

4.4 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: Akaike (1973), Ben-Akiva and Lerman (1985), Dyke and Patterson (1952), Greene (2003), Kendall and Stuart (1979), Hilbe (1993a), Hilbe (1994), Hilbe (1993b), Maddala (1983), and Goossens, Mittelbach, and Samarin (1994). They are generated by using the `\citet` and `\citet*` commands from the `natbib` package. Here we test `\citeb` and `\citebetal`: Akaike [1973], Ben-Akiva and Lerman [1985], Dyke and Patterson [1952], Greene [2003], Kendall and Stuart [1979], Hilbe [1993a], Hilbe [1994], Hilbe [1993b], Maddala [1983], and Goossens, Mittelbach, and Samarin [1994]. 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 Goossens, Mittelbach, and Samarin (1994).

4.5 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}
```

5 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.

5.1 Citing the Stata manuals

The macros for generating references to the Stata manuals are given in table 1.

Table 1: Stata manual references

Example	Result
`\bayesref{bayes}`	[BAYES] **bayes**
`\cmref{cmchoiceset}`	[CM] **cmchoiceset**
`\dref{Data types}`	[D] **Data types**
`\dsgeref{dsge}`	[DSGE] **dsge**
`\ermref{eregress}`	[ERM] **eregress**
`\fnref{Statistical functions}`	[FN] **Statistical functions**
`\fmmref{fmm:~betareg}`	[FMM] **fmm: betareg**
`\grefa{Graph Editor}`	[G-1] **Graph Editor**
`\grefb{graph}`	[G-2] **graph**
`\grefci{line\_options}`	[G-3] ***line_options***
`\grefdi{connectstyle}`	[G-4] ***connectstyle***
`\gsref{6~Using the Data Editor}`	[GS] **6 Using the Data Editor**
`\irtref{irt}`	[IRT] **irt**
`\lassoref{Lasso intro}`	[LASSO] **Lasso intro**
`\metaref{meta}`	[META] **meta**
`\meref{me}`	[ME] **me**
`\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**
`\pssrefa{Intro}`	[PSS-1] **Intro**
`\pssrefb{power}`	[PSS-2] **power**
`\pssrefc{ciwidth}`	[PSS-3] **ciwidth**
`\pssrefd{Unbalanced designs}`	[PSS-4] **Unbalanced designs**
`\pssrefe{Glossary}`	[PSS-5] **Glossary**
`\pssref{Subject and author index}`	[PSS] **Subject and author index**
`\rptref{Dynamic documents intro}`	[RPT] **Dynamic documents intro**
`\rref{regress}`	[R] **regress**
`\spref{Intro}`	[SP] **Intro**
`\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**

5.2 Stata syntax

Here is an example syntax display:

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

This syntax is generated by

```
\begin{stsyntax}
\dunderbar{reg}ress
    \depvar\
    \optindepvars\
    \optif\
    \optin\
    \optweight\
    \optional{,
    \underbar{nocons}tant
    \underbar{h}ascons
    tsscons
    vce({\it vcetype\/})
    \underbar{l}evel(\num)
    \underbar{b}eta
    \underbar{ef}orm(\ststring)
    \dunderbar{dep}name(\varname)
    {\it display\_options}
    \underbar{nohe}ader
    \underbar{notab}le
    plus
    \underbar{ms}e1
    \underbar{coefl}egend}
\end{stsyntax}
```

Each command should be formatted using a separate `stsyntax` environment. Table 2 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`	*newvar*	`\byvarlist`	`by` *varlist*`:`
`\optnewvarname`	[*newvar*]	`\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] **20.8 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{20.8~Specifying the width of confidence intervals}.
```

5.3 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). Results should be reproducible. Begin examples by loading the data. Code should be written to respect a linesize of 80 characters. 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

```
. sjlog 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 Knuth (1986).

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.

```
. sjlog type output.do
```

5.4 About tables

Tables should be created using the standard LaTeX methods. See Lamport (1994) for a discussion and examples. Tables should be included in the main text rather than at the end of the document. Tables should be called out in the text prior to appearance.

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
```

Tables with notes

Table 3 shows the order and format to use for notes to tables.

Table 3: Industrial clusters

China		United States	
Core of cluster	Size (in # of units)	Core of cluster	Size (in # of units)
Construction	28[a]	Public administration and defense; compulsory social security	30[b]
Food, beverages, and tobacco	3	Food, beverages, and tobacco	2
Textiles and textile products	2	Chemicals and chemical products	1
Chemicals and chemical products	1	Basic metals and fabricated metal	1
Transport equipment	1	Transport equipment	1
$L_a = 0.602$***		$L_a = 0.567$	
$L_w = 0.828$**		$L_w = 0.837$	
$L_m = 0.335$*		$L_m = 0.287$	
$K^* = 5$		$K^* = 5$	
$K = 35$		$K = 35$	

SOURCE: Pew Research Center.
NOTE: U.S. industrial clusters based on U.S. input–output flows of goods expressed in millions of dollars between 35 ISIC industries from the WIOD data. The minimum number of clusters `k()` was set equal to five. The algorithm returns L_a, L_w, and L_m, which refer to the average of the internal relative flows, the population-weighted average of the internal relative flows, and the minimum of the internal relative flows, respectively. K^* and K refer to the number of defined regional clusters and the number of distinct starting units, respectively.
[a] This note pertains only to row 1 column 2.
[b] This note pertains only to row 1 column 4.
*** denotes $p < 0.01$; ** denotes $p < 0.05$; * denotes $p < 0.1$.

Order of notes should be

1. source notes
2. notes applying to the whole table
3. notes applying to specific parts of the table

4. notes on significance levels

Special notes:

- Use `\centering` because the `center` environment adds unnecessary vertical spacing.
- Place the `\begin{threeparttable}` line above the caption.

Tables should be included in the main text rather than at the end of the document. Tables should be called out in the text prior to appearance.

5.5 Encapsulated PostScript (EPS)

You can include figures by using either `\includegraphics` or `\epsfig`.

```
\begin{figure}[h!]
\begin{center}
\includegraphics{eps/output1.eps}
\end{center}
\caption{Scatterplot with simple linear regression line}
\label{fig}
\end{figure}

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

Figure 1 is included using `\epsfig` from the `epsfig` package.

The graph was generated by running `output.do`, the do-file given in section 5.3. The `epsfig` package is described in Goossens, Mittelbach, and Samarin (1994).

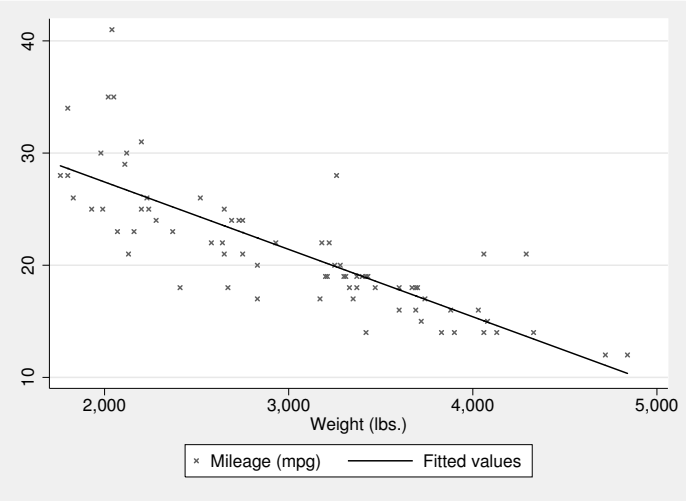

Figure 1: Scatterplot with simple linear regression line

EPS is the preferred format for graphs and line art. Figures should be included in the main text rather than at the end of the document and should be called out in the text prior to appearance. If your article is written in Word, you should submit your figures as separate EPS files. Rasterized-based files of at least 300 dpi (dots per inch) are acceptable. Avoid using bitmaps for figures and graphs, because even if images are outputted at 300 dpi, bitmaps can increase the size of the resulting file for printing. (However, bitmaps will be allowed for photographs, which are used in, for example, the *Stata Journal* Editors' prize announcement.) Images should be submitted in black and white (grayscale). We recommend that graphs created in Stata use the `sj` scheme.

5.6 Stored results

The `stresults` environment provides a table to describe the stored 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` stored 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

Alternatively, you can use the `stresults2` environment to create a two column table. This format works better if your descriptions are long.

5.7 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).

❑

5.8 Special characters

Table 4 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 4: Special characters

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

5.9 Equations and formulas

In (1), $\overline{x}$ was generated using `\stbar{x}`. Here `\stbar` is equivalent to the TEX macro `\overline`.

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

In (2), $\widehat{\beta}$ was generated using `\sthat{\beta}`. Here `\sthat` is equivalent to the TEX macro `\widehat`.

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

Formulas should be defined and follow a concise style. Different disciplines adhere to different notation styles; however, if the notation cannot be clearly interpreted, you may be asked to make changes. The bolding and font selection guidelines are the following:

- Matrices are capitalized and bolded; for instance, $\mathbf{\Pi} + \mathbf{\Theta} + \mathbf{\Phi} - \mathbf{B}$.
- Vectors are lowercased and bolded; for instance, $\boldsymbol{\pi} + \boldsymbol{\theta} + \boldsymbol{\phi} - \mathbf{b}$.
- Scalars are lowercased and nonbolded; for instance, $r_2 + c_1 - c_2$.

Sentence punctuation should not be used in formulas set off from the text.

Formulas in line with the text should use the solidus (/) instead of a horizontal line for fractional terms.

Nesting of grouping is square brackets, curly braces, and then parentheses, or [{()}].

Only those equations explicitly referred to in the text should be assigned an equation number.

6 References

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, ed. B. N. Petrov and F. Csaki, 267–281. Budapest, Hungary: Akademiai Kiado.

Ben-Akiva, M., and S. R. Lerman. 1985. *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA: MIT Press.

Dyke, G. V., and H. D. Patterson. 1952. Analysis of factorial arrangements when the data are proportions. *Biometrics* 8: 1–12.

Goossens, M., F. Mittelbach, and A. Samarin. 1994. *The LaTeX Companion*. Reading, MA: Addison–Wesley.

Greene, W. H. 2003. *Econometric Analysis*. 5th ed. Upper Saddle River, NJ: Prentice Hall.

Hilbe, J. 1993a. sg16: Generalized linear models. *Stata Technical Bulletin* 11: 20–28. Reprinted in *Stata Technical Bulletin Reprints*. Vol. 2, pp. 149–159. College Station, TX: Stata Press.

———. 1993b. Log Negative Binomial Regression as a Generalized Linear Model. *Graduate College Committee on Statistics* (Technical Report 26).

———. 1994. Generalized linear models. *American Statistician* 48: 255–265.

Kendall, M., and A. Stuart. 1979. *The Advanced Theory of Statistics*. Vol. 2. 4th ed. London: Griffin.

Knuth, D. E. 1986. *The TEX book*. Reading, MA: Addison–Wesley.

Kolenikov, S., and J. Pitblado. 2014. Analysis of complex health survey data. In *Handbook of Health Survey Methods*, ed. T. P. Johnson, chap. 29. Hoboken, NJ: Wiley.

Lamport, L. 1994. *LATEX: A Document Preparation System*. 2nd ed. Reading, MA: Addison–Wesley.

Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press.

van Buuren, S. 2018. *Flexible Imputation of Missing Data*. 2nd ed. Chapman & Hall/CRC.

About the authors

Stas Kolenikov is Principal Statistician at NORC who has been using Stata and writing Stata programs for about 25 years. He had worked on economic welfare and inequality, spatiotemporal environmental statistics, mixture models, missing data, multiple imputation, structural equations with latent variables, resampling methods, complex sampling designs, survey weights, Bayesian mixed models, combining probability and non-probability samples, latent class analysis, and likely some other stuff, too.