

# THE STATA JOURNAL

## **Editor**

H. Joseph Newton  
Department of Statistics  
Texas A&M University  
College Station, Texas 77843  
979-845-8817; fax 979-845-6077  
jnewton@stata-journal.com

## **Editor**

Nicholas J. Cox  
Department of Geography  
Durham University  
South Road  
Durham City DH1 3LE UK  
n.j.cox@stata-journal.com

## **Associate Editors**

Christopher F. Baum  
Boston College

Nathaniel Beck  
New York University

Rino Bellocco  
Karolinska Institutet, Sweden, and  
University of Milano-Bicocca, Italy

Maarten L. Buis  
Vrije Universiteit, Amsterdam

A. Colin Cameron  
University of California–Davis

Mario A. Cleves  
Univ. of Arkansas for Medical Sciences

William D. Dupont  
Vanderbilt University

David Epstein  
Columbia University

Allan Gregory  
Queen's University

James Hardin  
University of South Carolina

Ben Jann  
ETH Zürich, Switzerland

Stephen Jenkins  
University of Essex

Ulrich Kohler  
WZB, Berlin

Frauke Kreuter  
University of Maryland–College Park

**Stata Press Editorial Manager**

**Stata Press Copy Editors**

Jens Lauritsen  
Odense University Hospital

Stanley Lemeshow  
Ohio State University

J. Scott Long  
Indiana University

Thomas Lumley  
University of Washington–Seattle

Roger Newson  
Imperial College, London

Austin Nichols  
Urban Institute, Washington DC

Marcello Pagano  
Harvard School of Public Health

Sophia Rabe-Hesketh  
University of California–Berkeley

J. Patrick Royston  
MRC Clinical Trials Unit, London

Philip Ryan  
University of Adelaide

Mark E. Schaffer  
Heriot-Watt University, Edinburgh

Jeroen Weesie  
Utrecht University

Nicholas J. G. Winter  
University of Virginia

Jeffrey Wooldridge  
Michigan State University

Lisa Gilmore

Jennifer Neve and Deirdre Patterson

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the web page

<http://www.stata-journal.com>

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index<sup>®</sup>
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch<sup>®</sup>)

**Copyright Statement:** The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible web sites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.

## Stata tip 79: Optional arguments to options

Nicholas J. Cox  
Department of Geography  
Durham University  
Durham, UK  
n.j.cox@durham.ac.uk

Programmers occasionally would like an option for a program to come in two flavors: a simple or default option, with no arguments, and a more complicated but more flexible alternative, with arguments. For example, the simple option might call up a graph with programmer-chosen defaults, while the complicated option might pass graph options to the graph command in question, signaling variations from those defaults.

With a simple trick, you can implement two options that appear to the user to be this single option that is either simple or complicated. Following age-old programmer jargon, let us imagine an option that can be `foobar` or `foobar(arguments)`.

Step 1: Declare to `syntax` that there are two options, say, `foobar` and `FOOBAR2()`, and the latter is precisely, say, `FOOBAR2(string)`. The outburst of uppercase letters indicates to `syntax` that the latter can be abbreviated `foobar()`. You can also indicate names that can be abbreviated more, say, `FOObar` and `FOObar2(string)`.

Step 2: Process input within your program. For example,

```
if "`foobar'`foobar2'" != ""
```

is a test of whether either option has been called. If `foobar2()` has been called, then the local macro `foobar2` will be defined and can be treated further. If the argument to `foobar2()` might itself contain quotation marks, then compound double quotes, `" "`, are in order.

Step 3: In documentation for the user, you need not mention the two options but may merely declare that the syntax is that an argument is optional, e.g.,  
`foobar [ (string) ]`.

If curious users find out by looking at the code that the option with arguments is really `foobar2()`, no harm is done. They would be partway to working out, independently of this tip, how the optional options are coded. Browsing code and borrowing tricks that you want to use yourself remains one of the best ways to grow as a programmer.