

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
Tübingen University, Germany

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 Editor

Peter A. Lachenbruch
Oregon State University

Jens Lauritsen
Odense University Hospital

Stanley Lemeshow
Ohio State University

J. Scott Long
Indiana University

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

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[®]
- Current Contents/Social and Behavioral Sciences[®]
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch[®])
- Social Sciences Citation Index[®]

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.

Speaking Stata: The statsby strategy

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

Abstract. The `statsby` command collects statistics from a command yielding r-class or e-class results across groups of observations and yields a new reduced dataset. `statsby` is commonly used to graph such data in comparisons of groups; the `subsets` and `total` options of `statsby` are particularly useful in this regard. In this article, I give examples of using this approach to produce box plots and plots of confidence intervals.

Keywords: gr0045, statsby, graphics, groups, comparisons, box plots, confidence intervals

1 Introduction

Datasets are often subdivided at one or more levels according to some kind of group structure. Statistically minded researchers are typically strongly aware of the need for, and the value of, comparisons between patients, hospitals, firms, countries, regions, sites, or whatever the framework is for collecting and organizing their data. Indeed, for many people, that kind of comparison is at the heart of what they do daily within their research.

Stata supports separate group analyses in various ways. Perhaps the most well-known and important is the `by:` construct, a subject of one of the earliest *Speaking Stata* columns (Cox 2002). This column focuses on [D] `statsby`, a command that until now has received only passing mention in *Speaking Stata* (Cox 2001, 2003). The main idea of `statsby` is simple and it is well documented. However, experience on Statalist and elsewhere indicates that many users who would benefit from `statsby` are unaware of its possibilities. The extra puff of publicity here goes beyond the manual entry in stressing its potential for graphical comparisons.

Focusing exclusively on `statsby` is not intended as a denial that there are other solutions to the same, or related, problems. The work of Newson (1999, 2000, 2003) is especially notable in this regard and goes beyond the singular purpose explored here.

2 The main idea

The main idea of `statsby` is that it offers a framework, not only for automating separate analyses for each of several groups, but also for collating the results. The effect is to relieve users of much of the tedious organizing work that would be needed otherwise. The

default mode of operation is that `statsby` overwrites the original dataset, subdivided in some way, with a reduced dataset with just one observation for each group. The `saving()` option, however, permits results to be saved on the side so that the original dataset remains in memory.

A common and essentially typical example of applying `statsby` is that a panel dataset containing one or more observations for each panel would be reduced to a dataset with precisely one observation for each panel. Those observations contain panel identifiers together with results for each panel, usually `e-` or `r-`class results from some command. Because there is no stipulation that the command called is an official command, there is scope for users to write their own programs leaving such results in their wake and thus to automate essentially any kind of calculation.

`statsby` does not support graphs directly, but the implications for graphics are immediate. Graphics for groups imply the collation of group results followed by graphing operations. Using `statsby` can reduce the problem to just the second of these two, subject as usual to minor questions of titling, labeling, and so forth.

3 Box plots for all possible subsets

I will not recapitulate the details of the manual entry, which those unfamiliar with the command can read for themselves. Rather, I will underline the value of `statsby` by showing how it makes several graphical tasks much easier.

Variants on box plots remain popular in statistical science. In an earlier column (Cox 2009), I underlined how `graph twoway` allows your own alternatives if ever the offerings of `graph box` or `graph hbox` are not quite suitable.

Let us pick up that theme and give it a new twist. The `subsets` option of `statsby` makes easy a division into all possible subsets of a dataset. That can be useful so long as you remember enough elementary combinatorics to avoid trying to produce an impracticable or impossible graph.

We will use the `sj` scheme standard for the *Stata Journal* and `auto.dta` bundled with Stata.

```
. set scheme sj
. sysuse auto
(1978 Automobile Data)
```

A small piece of foresight—benefiting from the hindsight given by earlier attempts excised from public view—is now to save a variable label that would otherwise disappear on reduction. In this example, we could also just type in the label or some other suitable text afterward. But if you try something similar yourself, particularly if you want to automate the production of several graphs, the small detail of saving text you want as a graph title may avoid some frustration.

```
. local xtitle "`': var label mpg'"
```

Our call to `statsby` spells out that we want five quantiles, the median, two quartiles, and two extremes. `summarize`, `detail` does that work. Because `summarize` is an r-class command, we need to look up the codes used, either by reverse engineering from the results of `return list` or by looking at the command help or the manual entry.

The principle with an e-class command is identical, except that we would reverse engineer from `ereturn list`. If this detail on r- and e-class results goes beyond your present familiarity, start at `help saved results` and follow the documentation pointers there if and as desired.

We are subdividing `auto.dta` by the categorical variables `foreign` and `rep78`, but with a twist given by the `subsets` option. Another useful option—in practice, probably even more useful—is to use the `total` option to add results for the whole set.

```
. statsby p50=r(p50) p25=r(p25) p75=r(p75) min=r(min) max=r(max),
> by(foreign rep78) subsets total: summarize mpg, detail
(running summarize on estimation sample)
      command: summarize mpg, detail
              p50: r(p50)
              p25: r(p25)
              p75: r(p75)
              min: r(min)
              max: r(max)
              by:  foreign rep78

Statsby subsets
-----|----- 1 -----|----- 2 -----|----- 3 -----|----- 4 -----|----- 5
.....|.....
```

Let us look at the results. To emphasize the key point: This is a reduced dataset and the original dataset is gone, although overwriting is avoidable through `saving()`. We have results for all combinations of `foreign` and `rep78` that exist in the data; for all categories of `foreign` and for all categories of `rep78`; and for all observations.

(Continued on next page)

```
. list
```

	foreign	rep78	p50	p25	p75	min	max
1.	Domestic	1	21	18	24	18	24
2.	Domestic	2	18	16.5	23	14	24
3.	Domestic	3	19	16	21	12	29
4.	Domestic	4	18	15	21	14	28
5.	Domestic	5	32	30	34	30	34
6.	Domestic	.	19	16	22	12	34
7.	Foreign	3	23	21	26	21	26
8.	Foreign	4	25	23	25	21	30
9.	Foreign	5	25	18	35	17	41
10.	Foreign	.	25	21	28	17	41
11.	.	1	21	18	24	18	24
12.	.	2	18	16.5	23	14	24
13.	.	3	19	17	21	12	29
14.	.	4	22.5	18	25	14	30
15.	.	5	30	18	35	17	41
16.	.	.	20	18	25	12	41

Plotting that data directly would produce a reasonable working graph. Largely as a matter of personal taste, I chose to reorganize and edit the data slightly to get something more attractive. First, I wanted all two-group categories together, then all one-group categories, and then all the data. The number of groups in each category is the complement of the number of missing values of the first two variables in each observation or row, so that can be calculated by counting missing values in each row and `sorting` accordingly. The `stable` option minimizes departure from the present sort order.

```
. egen order = rowmiss(foreign rep78)
. sort order, stable
```

To get group labels, I combine the value labels (where used) and the values (otherwise) with `egen's concat()` function, and I remove the periods indicating missing and any marginal spaces:

```
. egen label = concat(foreign rep78), decode p(" ")
. replace label = trim(subinstr(label, ".", "", .))
(8 real changes made)
```

The total category for results for all observations deserves due prominence:

```
. replace label = "Total" in L
(1 real change made)
```

The final detail of preparation is to use a couple of helper programs to set up one axis variable with gaps and to map the values in the `label` variable to the value labels of that axis variable:

```
. seqvar x = 1/5 7/9 11/12 14/18 20
. labmask x, values(label)
```

For more details on `seqvar` and `labmask`, see Cox (2008).

Now we assemble the box plot from ingredients produced by members of the `twoway` family. `rspike` draws spikes between each quartile and each tailward extreme. `rbar` draws boxes between the quartiles. `scatter` draws point symbols showing the medians. The result is shown in figure 1. At this point, we use the title carefully stored in a local macro before the call to `statsby`.

```
. twoway rspike min p25 x, horizontal bcolor(gs12) ||
> rspike p75 max x, horizontal bcolor(gs12) ||
> rbar p25 p75 x, horizontal barw(0.8) bcolor(gs12) ||
> scatter x p50, ms(0) yla(1/5 7/9 11/12 14/18 20, val nogrid noticks ang(h))
> legend(off) ysc(reverse) xtitle(`xtitle`)
```

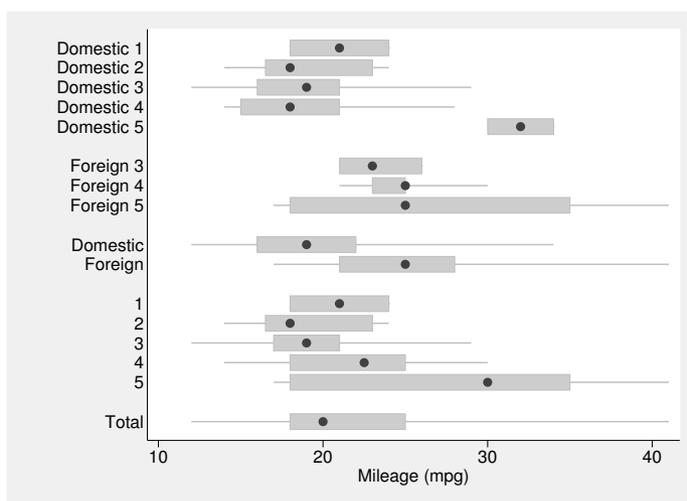


Figure 1. All subsets box plot of mileage for 78 cars by domestic or foreign origin, repair record in 1978, and combinations thereof. Spikes extend to extremes, boxes show quartiles, and circles show medians.

Beyond question, the statistical and stylistic choices here of what to show and how to show it are all arguable and variable. However, that is not the main point. Rather, you should appreciate how `statsby` with its `subsets` and `total` options made a different kind of plot much easier.

(Continued on next page)

4 Confidence-interval plots

Plotting confidence intervals for some group statistic, such as the mean, is another common application. The basic trick, which now starts to look fairly obvious, is to use a command such as `ci` (see [R] `ci`) under the aegis of `statsby` to produce a reduced dataset that is then ready for graphics.

We read in the U.S. National Longitudinal Survey data available from Stata's web site. We will look at the relationship between wage (on an adjusted logarithmic scale) and highest education grade.

```
. webuse nlswork, clear
(National Longitudinal Survey. Young Women 14-26 years of age in 1968)
```

Saving the variable label is a trick we saw before. At worst, it does no harm.

```
. local ytitle "`': var label ln_wage'"
```

`ci` is our workhorse. The default gives 95% confidence intervals, but evidently other choices may suit specific purposes. As the dataset is panel data, we need to decide how far to respect its structure. One of several possible approaches is to select one observation from each panel randomly (but reproducibly). In addition to estimates and confidence intervals, we save the sample sizes, which are a key part of the information.

```
. set seed 2803
. generate rnd = runiform()
. bysort idcode (rnd): generate byte select = _n == 1
. statsby mean=r(mean) ub=r(ub) lb=r(lb) N=r(N) if select, by(grade) clear:
> ci ln_wage
(running ci on estimation sample)
      command: ci ln_wage if select
             mean:  r(mean)
             ub:   r(ub)
             lb:   r(lb)
             N:    r(N)
             by:   grade

Statsby groups
-----|----- 1 -----|----- 2 -----|----- 3 -----|----- 4 -----|----- 5
(2 missing values generated)
.....
```

Here the grades run over all the integers 0/18, but `levelsof` (see [P] `levelsof`) simplifies capture for later graphical use of all values that occur, especially in more complicated cases.

```
. levelsof grade, local(levels)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
```

A basic plot is now at hand. Using `scatter` for the means and `rcap` for the intervals themselves is widely conventional. A delicate detail is that means on top of intervals look better than the converse. The result is shown in figure 2.

```
. twoway rcap ub lb grade || scatter mean grade, yti(`ytitle`) legend(off)
> subtitle(95% confidence intervals for mean, place(w)) xla(`levels`)
```

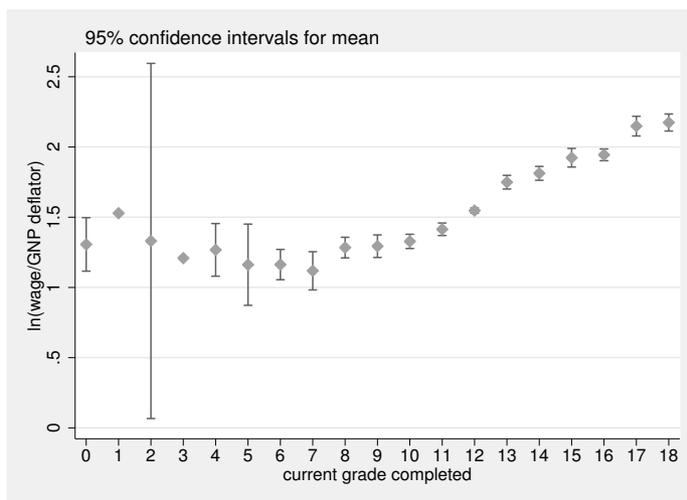


Figure 2. Graph of 95% confidence intervals of mean adjusted log wage by education grade.

The graph can be improved in various minor cosmetic ways and also by showing sample sizes. After some experimenting, the method for the latter was refined to showing sizes as marker labels on a horizontal line. Horizontal alignment of those labels would have been preferable, except that they then would run into one another. Exchanging axes so that grade is plotted vertically seems too awkward for this kind of data. In other circumstances, exchanging axes might well be a good idea. Thus the vertical alignment here is regarded as the lesser of two evils. Figure 3 shows the result.

```
. generate where = 2.7
. twoway rcap ub lb grade || scatter mean grade, yti(`ytitle`) legend(off)
> subtitle(95% confidence intervals for mean, place(w)) xla(`levels`)
> || scatter where grade, ms(none) mla(N) mlabangle(v) mlabpos(0) ysc(r(. 2.8))
> yla(0(.5)2.5, ang(h))
```

(Continued on next page)

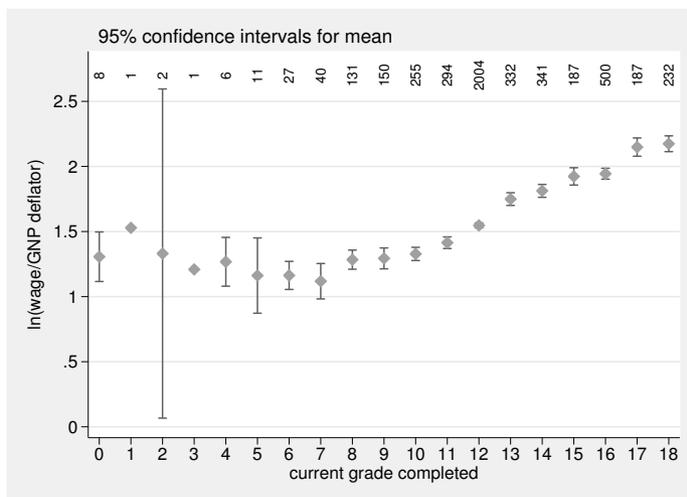


Figure 3. Graph of 95% confidence intervals of mean adjusted log wage by education grade. Text labels show sample sizes at each grade.

5 Conclusions

This column has promoted one simple idea, using `statsby` to prepare a reduced dataset for subsequent graphing. Its `subsets` and `total` options allow useful variations on the default. You might still need to do some further work to get a good graph, but the overall labor is nevertheless likely to be much reduced. The method is widely applicable in so far as any calculation can be represented by a program as yielding r-class or e-class results.

6 Acknowledgments

Vince Wiggins planted the immediate seed for this column with a single cogent remark. Martin Weiss has been an energetic proponent of `statsby` on Statalist.

7 References

- Cox, N. J. 2001. Speaking Stata: How to repeat yourself without going mad. *Stata Journal* 1: 86–97.
- . 2002. Speaking Stata: How to move step by: step. *Stata Journal* 2: 86–102.
- . 2003. Speaking Stata: Problems with tables, Part I. *Stata Journal* 3: 309–324.

- . 2008. Speaking Stata: Between tables and graphs. *Stata Journal* 8: 269–289.
- . 2009. Speaking Stata: Creating and varying box plots. *Stata Journal* 9: 478–496.
- Newson, R. 1999. dm65: A program for saving a model fit as a dataset. *Stata Technical Bulletin* 49: 2–6. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 19–23. College Station, TX: Stata Press.
- . 2000. dm65.1: Update to a program for saving a model fit as a dataset. *Stata Technical Bulletin* 58: 2. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, p. 7. College Station, TX: Stata Press.
- . 2003. Confidence intervals and p-values for delivery to the end user. *Stata Journal* 3: 245–269.

About the author

Nicholas Cox is a statistically minded geographer at Durham University. He contributes talks, postings, FAQs, and programs to the Stata user community. He has also coauthored 15 commands in official Stata. He wrote several inserts in the *Stata Technical Bulletin* and is an editor of the *Stata Journal*.