Generalized confidence interval plots using commands or dialogs

Roger Newson
King's College London, London, UK
roger.newson@kcl.ac.uk
http://phs.kcl.ac.uk/rogernewson/

Presented at the 11th UK Stata Users' Group Meeting on 17 May 2005

This presentation can be downloaded from the conference website at
http://ideas.repec.org/s/boc/usug05.html

The `eclplot` package

The `eclplot` package

- The `eclplot` package (downloadable from SSC) produces plots of estimates and confidence limits.

The `eclplot` package

- The `eclplot` package (downloadable from SSC) produces plots of estimates and confidence limits.
- It inputs a Stata dataset with 1 observation per confidence interval, 3 variables containing estimates and lower and upper confidence limits, and a fourth variable, against which the confidence intervals are plotted.

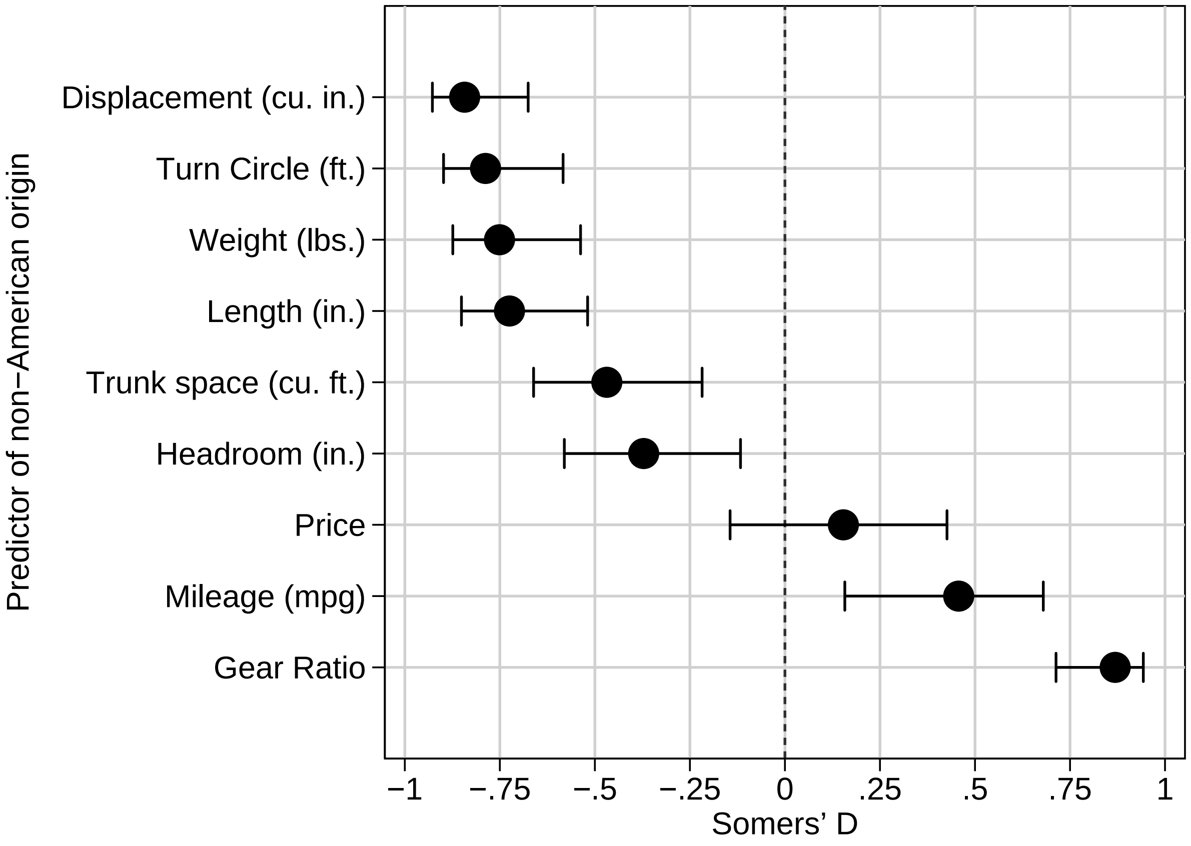
The `eclplot` package

- The `eclplot` package (downloadable from SSC) produces plots of estimates and confidence limits.
- It inputs a Stata dataset with 1 observation per confidence interval, 3 variables containing estimates and lower and upper confidence limits, and a fourth variable, against which the confidence intervals are plotted.
- It outputs a confidence interval plot, giving a choice of 56 plot types.

The `eclplot` package

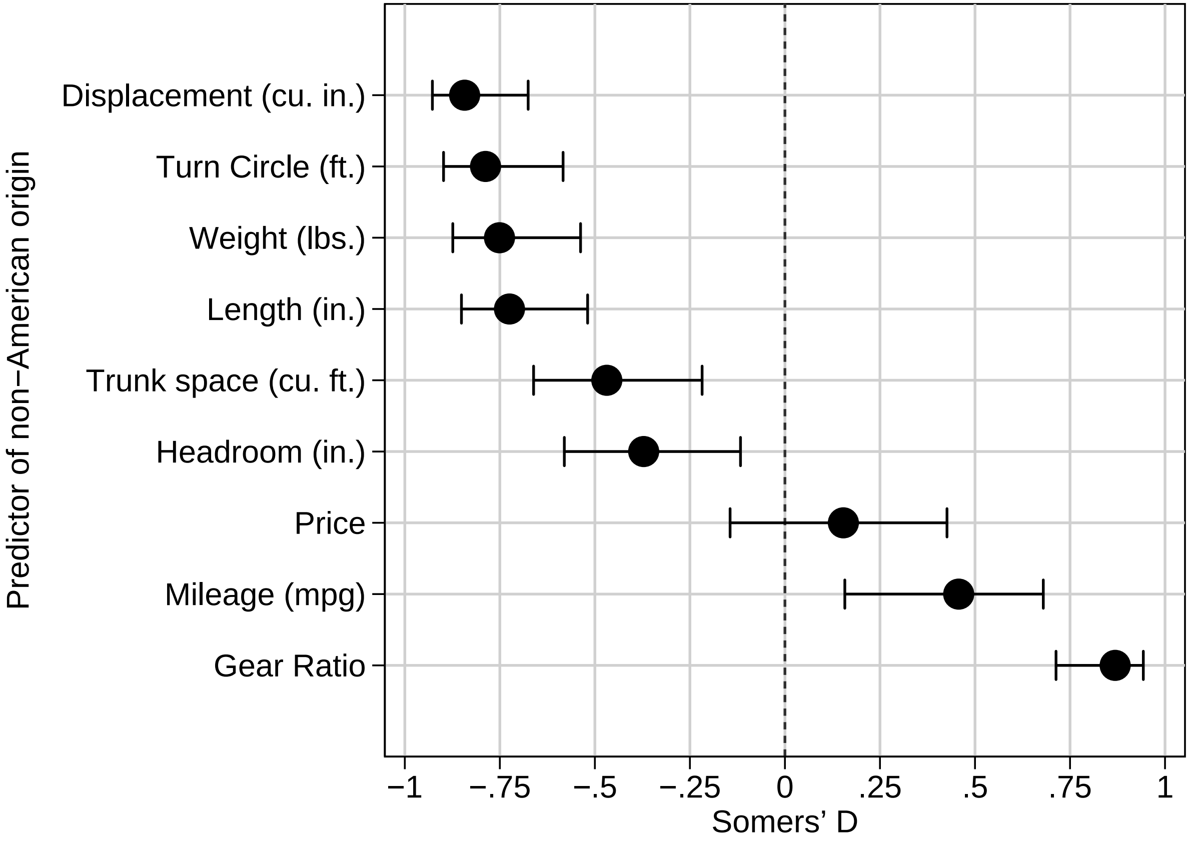
- The `eclplot` package (downloadable from SSC) produces plots of estimates and confidence limits.
- It inputs a Stata dataset with 1 observation per confidence interval, 3 variables containing estimates and lower and upper confidence limits, and a fourth variable, against which the confidence intervals are plotted.
- It outputs a confidence interval plot, giving a choice of 56 plot types.
- Some of these plot types are more useful than others, but they *all* can be either vertical or horizontal.

A horizontal confidence interval plot produced by `eclplot`



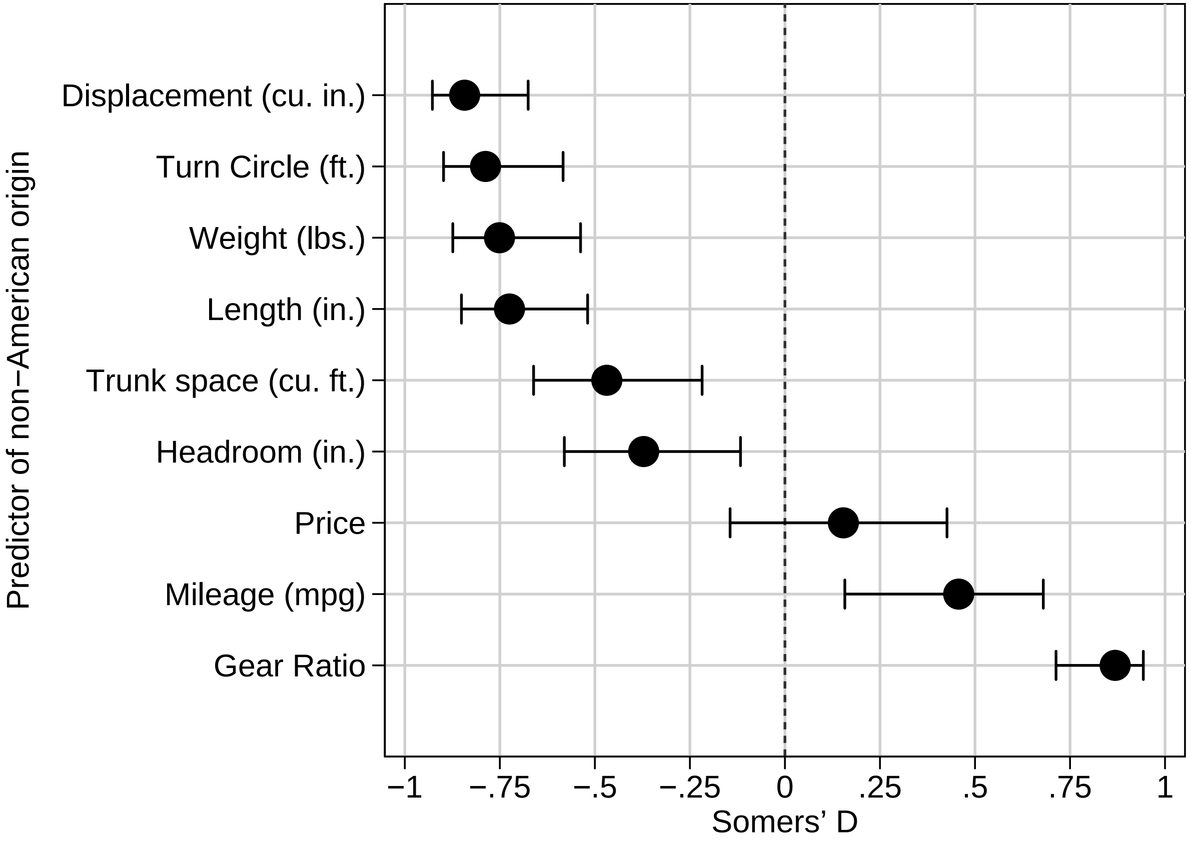
A horizontal confidence interval plot produced by `eclplot`

- The input dataset was produced from the **`auto`** data, using the SSC packages **`somersd`**, **`parmest`** and **`sencode`**.



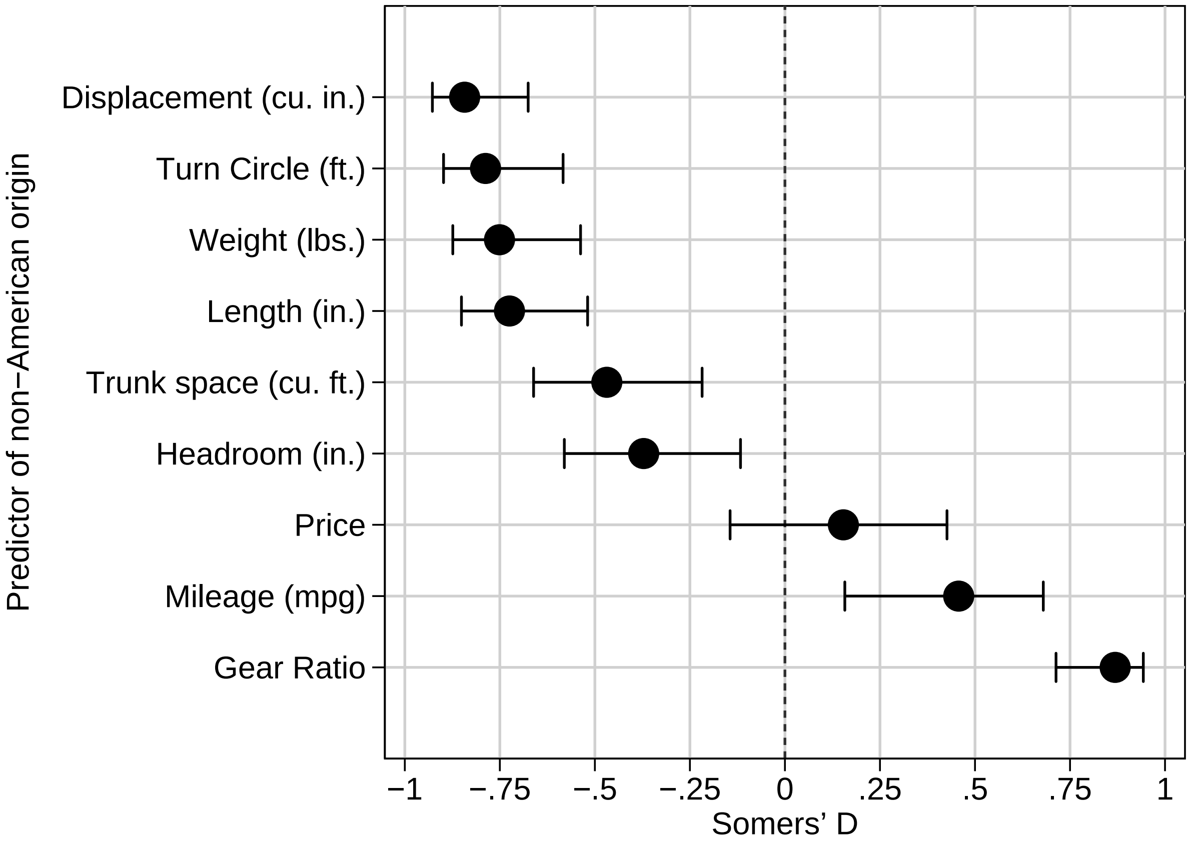
A horizontal confidence interval plot produced by `eclplot`

- The input dataset was produced from the **`auto`** data, using the SSC packages **`somersd`**, **`parmest`** and **`sencode`**.
- Somers' D measures the power of 9 quantitative car measurements to predict non-US origin.



A horizontal confidence interval plot produced by `eclplot`

- The input dataset was produced from the **`auto`** data, using the SSC packages **`somersd`**, **`parmest`** and **`sencode`**.
- Somers' D measures the power of 9 quantitative car measurements to predict non-US origin.
- It is negative for negative predictors, positive for positive predictors, and zero for non-predictors.



The input dataset: resultssets *versus* spreadsheets

The input dataset: resultssets *versus* spreadsheets

- A **resultsset** is a Stata dataset created as output by a Stata command, *usually* for input to another Stata command.

The input dataset: resultssets *versus* spreadsheets

- A **resultsset** is a Stata dataset created as output by a Stata command, *usually* for input to another Stata command.
- Resultssets with 1 observation per confidence interval can be produced by `statsby`, `postfile`, or the SSC package `parmest`.

The input dataset: resultssets *versus* spreadsheets

- A **resultsset** is a Stata dataset created as output by a Stata command, *usually* for input to another Stata command.
- Resultssets with 1 observation per confidence interval can be produced by `statsby`, `postfile`, or the SSC package `parmest`.
- Resultssets allow us to write do-files which take us *all the way* from the original data to the final plots and tables of confidence intervals.

The input dataset: resultssets *versus* spreadsheets

- A **resultsset** is a Stata dataset created as output by a Stata command, *usually* for input to another Stata command.
- Resultssets with 1 observation per confidence interval can be produced by `statsby`, `postfile`, or the SSC package `parmest`.
- Resultssets allow us to write do-files which take us *all the way* from the original data to the final plots and tables of confidence intervals.
- *However*, `eclplot` works equally well on datasets input from spreadsheets.

The input dataset: resultssets *versus* spreadsheets

- A **resultsset** is a Stata dataset created as output by a Stata command, *usually* for input to another Stata command.
- Resultssets with 1 observation per confidence interval can be produced by `statsby`, `postfile`, or the SSC package `parmest`.
- Resultssets allow us to write do-files which take us *all the way* from the original data to the final plots and tables of confidence intervals.
- *However*, `eclplot` works equally well on datasets input from spreadsheets.
- "Resultsspreadsheets" of confidence intervals can be produced using Ben Jann's SSC package `estout`.

Running `eclplot`: do-files *versus* dialogs

Running `eclplot`: do-files *versus* dialogs

- Do-files contain a documentary record of what we did (especially if we also use comments and variable labels).

Running `eclplot`: do-files *versus* dialogs

- Do-files contain a documentary record of what we did (especially if we also use comments and variable labels).
- *Unfortunately*, Stata graphics commands (even `eclplot`) are often long-winded, having many options and suboptions, with confusing names.

Running `eclplot`: do-files *versus* dialogs

- Do-files contain a documentary record of what we did (especially if we also use comments and variable labels).
- *Unfortunately*, Stata graphics commands (even `eclplot`) are often long-winded, having many options and suboptions, with confusing names.
- *Therefore*, unless you are a very fluent programmer, then you will probably find the `eclplot` dialog easier to use.

Running `eclplot`: do-files *versus* dialogs

- Do-files contain a documentary record of what we did (especially if we also use comments and variable labels).
- *Unfortunately*, Stata graphics commands (even `eclplot`) are often long-winded, having many options and suboptions, with confusing names.
- *Therefore*, unless you are a very fluent programmer, then you will probably find the `eclplot` dialog easier to use.
- *Fortunately*, the `eclplot` dialog works by assembling a full `eclplot` command, which is echoed to the `Results` window.

Running `eclplot`: do-files *versus* dialogs

- Do-files contain a documentary record of what we did (especially if we also use comments and variable labels).
- *Unfortunately*, Stata graphics commands (even `eclplot`) are often long-winded, having many options and suboptions, with confusing names.
- *Therefore*, unless you are a very fluent programmer, then you will probably find the `eclplot` dialog easier to use.
- *Fortunately*, the `eclplot` dialog works by assembling a full `eclplot` command, which is echoed to the `Results` window.
- This command can be cut and pasted to a do-file for future use.

Demonstration of `eclplot`

Demonstration of `eclplot`

- In the remainder of this presentation, we demonstrate **`eclplot`**, using example do-files and the **`eclplot`** dialog.

Demonstration of `eclplot`

- In the remainder of this presentation, we demonstrate `eclplot`, using example do-files and the `eclplot` dialog.
- The example do-files, and the demonstration notes, can be downloaded from the conference website at *http://ideas.repec.org/s/boc/usug05.html* together with these overheads.

Demonstration notes

We begin our demonstration by entering Stata and typing

```
do example1
```

The do-file `example1.do` inputs the `auto` dataset, replaces it with a resultsset of Somers' D estimates, and uses `eclplot` to create the plot we saw earlier. We can look at the resultsset using the menu sequence

```
Window->Data Editor
```

This dataset was not produced using a spreadsheet, but it easily could have been. It contains 1 observation for each of 9 quantitative predictors of non-US origin in the `auto` data, and variables giving the name of the predictor, the Somers' D estimate, the lower and upper confidence limits, the P-value, and the P-value stars.

The plot produced by `example1.do` can also be produced using the `eclplot` dialog. If we type

```
db eclplot
```

then we enter the `eclplot` dialog, which has 12 tabs. The 8 tabs on the right are standard official Stata tabs. The 4 tabs on the left are special `eclplot` tabs. The tab on the extreme left is the `Main` tab, and has 4 group boxes of controls. The first group box from the left contains the 4 most important controls, which the user *must* set. These specify 3 variables, containing the estimates, the lower confidence limits and the upper confidence limits, and a fourth variable, against which the confidence intervals are plotted. If we specify these and then `Submit`, then we see, after a system-dependent pause, that the default confidence interval plot is a vertical plot.

We wanted a horizontal confidence interval plot. To do this, we enter the `Main` tab, and reset the radio button labelled
`Confidence interval orientation:`
to `Horizontal CI plot`. We then `Submit`, and find that this produces a horizontal confidence interval plot. *However*, the Y-axis does not have the full set of labels, and the X-axis might also look better with more labels.

To change the Y-axis, we enter the `Y-axis` tab, set the `Rule:` control to specify axis ticks `1(1)9`, set `Range:` from 0 to 10, and enable `Grid` lines, and select, as a `Color:` for these grid lines, a shade of gray such as `Gray 13`. Similarly, to change the X-axis, we enter the `X-axis` tab, set the labels to `-1(0.25)1`, and specify grid lines in the same shade of gray. When we `Submit`, we find that the axes are fully labelled and have grid lines. *However*, now that we have grid lines, the estimates and confidence intervals are less prominent.

To change the look of the estimates, we enter the `Estimates` tab, and set the marker symbol size to `v.Large` and the marker color to `Black`. Similarly, to change the look of the confidence limits, we enter the `Confidence limits` tab, set the cap marker size to `v.Large`, and set the line color to `Black`. When we `Submit`, we find that the confidence intervals now stand out against the background of grid lines.

Finally, we might want to add a reference line on the X-axis at zero, the value of Somers' D for non-predictors. To do this, we enter the `Reference lines` tab, specify an X-axis reference line at 0, and set the line pattern to to `Short-dash`. When we `Submit`, we see the plot previously produced using `example1.do`.

If we look at the `Results` window, then we can now see the sequence of `eclplot` commands generated by the dialog to produce our sequence of plots. The first command was simple, but they become increasingly complicated as the plots become increasingly customized.

Plot types for estimates and confidence limits

We noted earlier that `eclplot` allows a choice of

56 plot types. If we enter the `Main` tab, then we see that the third group box from the left is labelled `Plot types`. This box contains two list box controls. One is labelled `Estimates plot type:`, and has 7 options. The other is labelled `Confidence intervals plot type:`, and has 8 options. These can be combined in any way, giving $7 \times 8 = 56$ combinations. We will only have time to see a few of them. Exactly how many depends on the length of the pause between hitting `Submit` and seeing the graph, which is highly system dependent.

The default combination (which we have already seen) is the estimates plot type `Symbols` and the confidence intervals plot type `Capped spikes`. An important non-default combination is the estimates plot type `Bars` and the confidence intervals plot type `Capped spikes`. This combination is known as a **detonator plot**. It is probably a good idea to enter the `Estimates` tab and select a bar width (such as 0.85 axis units), and a bar fill color (preferably a light one). Unlike a lot of software on the market, `eclplot` produces detonator plots with the confidence limits in the foreground and the bars in the background, so the bars do not hide the confidence limits. However, if you are determined to annoy statisticians, then this default can be reset, using a radio button in the second group box of the `Main` tab.

Superimposed plots using the `plot()` option

It is possible to add features to a confidence interval plot produced by `eclplot` by superimposing other plots. This can be done using the `plot()` option. To use this in the dialog, we must enter the very last tab on the right (labelled `Overall`), select the very last control (labelled `Additional graph options:`), and enter some of the complicated Stata graphics language that the dialog box was designed to avoid. For instance, in our current dataset, we can label the confidence intervals with P-value stars by typing

```
plot(scatter predictor max95, msymbol(none)
mlabel(stars) mlabpos(3) mlabsize(large))
```

When we `Submit`, we find that the confidence intervals are now labelled with P-value stars, one star for $P \leq 0.05$, two for $P \leq 0.01$, and three for $P \leq 0.001$.

Multiple plots using the `by()` and `supby()` options

`eclplot` can also produce multiple plots corresponding to multiple by-groups. We will demonstrate these using our second example do-file by typing `do example2`

The do-file `example2.do` inputs the `auto` data, but this time two sets of Somers' D estimates are calculated, one set for even-numbered cars and one set for odd-numbered cars. We first see the separate confidence interval plots produced by the `by()` option. The plot for even-numbered cars is on the left, and the plot for odd-numbered cars is on the right. We then see the superimposed confidence interval plots produced by the `supby()` option. This time, corresponding confidence intervals for the 2 car groups are side by side for ease of comparison, and we can see from the legend that the circles are estimates for even-numbered cars and the squares are estimates for odd-numbered cars.

Acknowledgements

Finally, I would like to thank John Moran of the Queen Elizabeth Hospital in Woodville, South Australia, for suggesting that something like the `supby()` option might be a good idea, and all at StataCorp, particularly Jean Marie Linhart, James Hassell, Derek Wagner and Vince Wiggins, for their help and advice on developing `eclplot` to where it is today.