The Stata Journal

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 webpage

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®)
- Scopus™
- Social Sciences Citation Index®

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, Mata, NetCourse, and Stata Press are registered trademarks of StataCorp LP.

The Stata Journal (2011)
11, Number 4, pp. 627–631

Stata tip 103: Expressing confidence with gradations

Ulrich Kohler
Wissenschaftszentrum Berlin (WZB)
Berlin, Germany
kohler@wzb.eu

Stephanie Eckman
Institute for Employment Research
Nuremberg, Germany
stephanie.eckman@iab.de

Researchers often express the uncertainty associated with a parameter estimate $\widehat{\mu}$ by plotting the 95% confidence interval (CI) around the statistic. The meaning of these intervals is that in the long run, 95% of the intervals formed in this way will include the fixed but unknown parameter of interest μ. It is also true, though we often neglect to mention it, that "the unknown μ is more often captured near the center of an interval than near the lower or upper limit, or end-point, of an interval" (Cumming 2007, 90). Our technique provides a method of plotting CIs that makes this facet of their interpretation more clear. Below we describe how to use Stata to plot the entire CI function—the p-value function (Poole 1987)—with gradations, thus depicting this often forgotten fact about CIs.

Let us start by reconsidering the creation of a plot of a point estimate with its 95% CI. The first step is usually to create a dataset containing several point estimates and their respective standard errors (what Newson [2003] calls a resultsset). We demonstrate how to do this for the arithmetic mean of the body mass index (BMI) from the National Health and Nutrition Examination Study (NHANES):

```
. use http://www.stata-press.com/data/r12/nhanes2
. collapse (mean) mean=bmi (semean) se=bmi, by(female)
```

The upper and lower bounds of the 95% CIs are calculated by adding and subtracting 1.96 times the standard error to the point estimates. We generate variables holding those values,

```
. generate upper = mean + 1.96*se
. generate lower = mean - 1.96*se
```

and use a range plot overlaid with a scatterplot to show the upper and lower limits of the CI along with the point estimate (see figure 1). Such graphs are commonly used to convey uncertainty in point estimates.

 gr0050

```
. graph twoway
>   || rcap upper lower female, lstyle(p1)
>   || scatter mean female, mstyle(p1)
>   || , legend(off) xlabel(0 "Men" 1 "Women") xtitle("")
>   xscale(range(-.5 1.5)) ytitle("BMI")
```

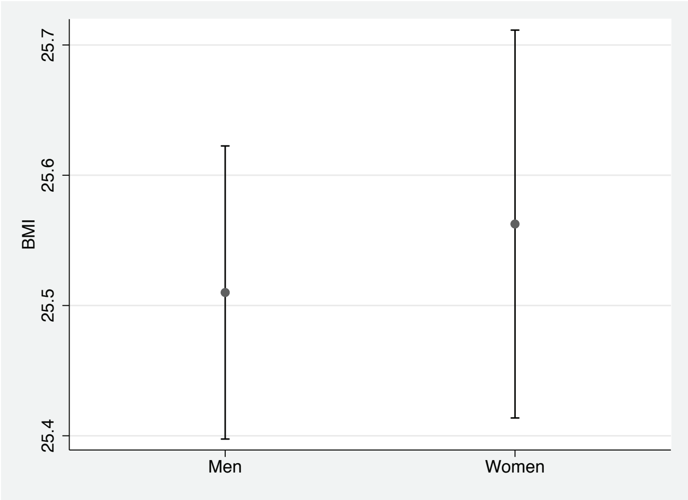

Figure 1. Average BMI of men and women with 95% CIs

The CIs shown in this graph are correct, assuming that the standard errors were calculated appropriately. However, we worry, in the spirit of Cumming (2007) and Läärä (2009, 141), that the figure gives the incorrect impression that the true value is equally likely to lie at any point in the interval. To avoid this misinterpretation, our technique plots the CIs with shading gradations to convey the sense that the true value is much more likely to be near the estimated mean than at the ends of the intervals. Our revised figure is shown in figure 2. Below we discuss the details of how we created these plots.

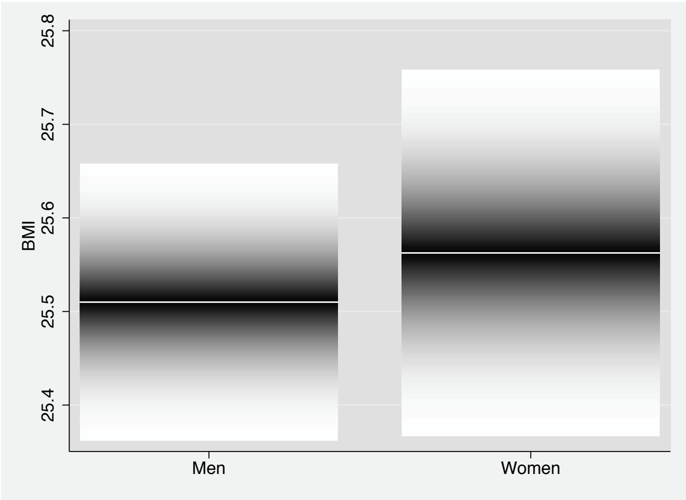

Figure 2. Average BMI of men and women with 95% CIs; the relative chance of capturing the true average is approximated by shading

In calculating the upper and lower limits of the 95% CI above, we multiplied the standard error by 1.96, the critical value of a Student's t distribution for 95% CI. We can calculate this critical value in Stata by using the `invttail()` function with the appropriate degrees of freedom:

```
. display abs(invttail(10350,(1-95/100)/2))
1.9601932
```

If we insert values other than 95 into this command, we can find the critical values for constructing intervals for any given level of confidence C. In fact, we can use a loop to form a series of pairs of variables holding the boundaries for CIs for all values of $C = 1, 2, \ldots, 99$:

```
. forv C = 1(1)99 {
  2.     gen ub`C´ = mean + abs(invttail(10350,(1-`C´/100)/2))*se
  3.     gen lb`C´ = mean - abs(invttail(10350,(1-`C´/100)/2))*se
  4. }
```

We can now plot each of these 99 pairs of CI bounds in one plot. We start with the 99% CI, which is the widest one. We then overlay it with the second widest, the 98% CI, and so on. To create the gradations in figure 2, we plot the wide intervals in a lighter shade than the narrower ones.

To accomplish these shading gradations, we use plot type `rbar` or `rarea`, which allows us to change the outlook of the bars or areas with each CI we plot, using the `fcolor()`, `fintensity()`, and `lcolor()` options. The `fcolor()` option sets the fill color of the bars; for example, `fcolor(black)` makes the bars black. The `fintensity()` option sets the intensity of the selected colors with a parameter from 0 to 100; for example, `fintensity(0)` sets a low intensity and `fintensity(100)` sets a high intensity. The `lcolor()` option sets the outline color; for example, `lcolor(black)` outlines the bars or areas with black lines. It is also possible to control the intensity of the outline color by multiplying the color with an intensity; for example, `lcolor(black*0.9)` is the same color as `fcolor(black)` with `fintensity(90)`.

Thankfully, we do not need to write the command for each of the 99 plots by hand. Instead, we can use a loop to create a local macro that holds the code for all 99 CI plots. We then create the command for the graph itself with that local macro. Here is the loop to create the local macro `rbar`, which holds the definitions for the 99 `rbar` plots. Note the use of the local `i` inside the loop to change the intensity and the line color with each plotted CI.

```
. forvalue i = 99(-1)1 {
  2.      local  rbarvar `rbarvar´
>      ||  rbar  ub`i´ lb`i´ female,
>      fcolor(black) fintensity(`=100-`i´´)
>      lcolor(black*`=(100-`i´)/100´)
>      barwidth(.8)
  3. }
```

And here is how we use the local macro `rbarvar` in a `graph twoway` command to create the graph. (The `graph` command also uses `scatteri` to add white lines to indicate the point estimates. We set the background color of the plot region to `gs14` to make the bright parts of the plot stand out.)

```
. graph twoway
>   `rbarvar´
>   || scatteri `=mean[1]´ -.4 `=mean[1]´ +.4
>   , recast(line) lcolor(white) lpattern(solid)
>   || scatteri `=mean[2]´ .6 `=mean[2]´ +1.4
>   , recast(line) lcolor(white) lpattern(solid)
>   || , legend(off) xlabel(0 "Men" 1 "Women") xtitle("")
>   plotregion(color(gs14)) ytitle("BMI")
```

The result is figure 2, shown above. The range bars spread between the limits of the 99% CI; however, the relative chance of capturing the true mean value of the BMI is approximated by shading: the darker the color, the greater the relative chance that the true mean lies in that area. Although "the vagaries of printing, and the human visual system, mean that [the plot] may not give an accurate impression of the relative chance" (Cumming 2007, 90), we feel that figure 2 does a better job of conveying the meaning of a CI than does figure 1.

This technique generalizes to many other situations. It works well for other kinds of point estimates and can also display CIs based on standard errors calculated by jackknife, bootstrap, or other methods.

One note of caution: We advise users of this technique to work with the resultsset approach (Newson 2003), that is, to construct a dataset that holds only the point estimates and their standard errors. Otherwise, the creation of 99 pairs of CI boundaries in large datasets could cause a user to run into memory issues.

References

Cumming, G. 2007. Inference by eye: Pictures of confidence intervals and thinking about levels of confidence. *Teaching Statistics* 29: 89–93.

Läärä, E. 2009. Statistics: Reasoning on uncertainty, and the insignificance of testing null. *Annales Zoologici Fennici* 46: 138–157.

Newson, R. 2003. Confidence intervals and p-values for delivery to the end user. *Stata Journal* 3: 245–269.

Poole, C. 1987. Beyond the confidence interval. *American Journal of Public Health* 77: 195–199.