The Stata Journal (2009)
9, Number 3, pp. 478–496

Speaking Stata: Creating and varying box plots

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

Abstract. Box plots have been a standard statistical graph since John W. Tukey and his colleagues and students publicized them energetically in the 1970s. In Stata, `graph box` and `graph hbox` are commands available to draw box plots, but sometimes neither is sufficiently flexible for drawing some variations on standard box plot designs. This column explains how to use `egen` to calculate the statistical ingredients needed for box plots and `twoway` to re-create the plots themselves. That then allows variations such as adding means, connecting medians, or showing all data points beyond certain quantiles.

Keywords: gr0039, box plots, dispersion diagrams, distributions, egen, graphics, percentile, quantile, range bars, twoway

1 Box plots

1.1 Origins

Box plots were so named by John W. Tukey and were publicized energetically within statistics by him, his colleagues, and his students from the 1970s on (e.g., Tukey [1972, 1977]; Velleman and Hoaglin [1981]; and Hoaglin, Mosteller, and Tukey [1983]). Box plots spread beyond statistics into several quantitative sciences through their own literature (e.g., Kleiner and Graedel [1980] and Cox and Jones [1981]). The publicity was so successful that the box plot is now widely regarded as a standard statistical graph. It appears in most introductory statistical texts; indeed, the exceptions to this rule (e.g., Freedman, Pisani, and Purves [2007]) are more striking than the examples. Further, the box plot is often assumed not to need explanation beyond such texts.

Box plots had several under-appreciated precursors under different names, including range bars (Spear 1952, 1969) and dispersion diagrams in geography and climatology (e.g., Crowe [1933] and Monkhouse and Wilkinson [1971]). Despite this earlier history, my guess is that box plots would not now be nearly so popular without Tukey's reinvention and propaganda.

1.2 Purpose

Stata users wishing to see box plots can call upon `graph box` or `graph hbox`. The manual entry [G] **graph box** explains several ways of tuning that command. Mitchell (2008) gives many examples of possible results and the code to get them. This column focuses on showing what to do whenever you want some variation on the standard design that cannot be met with `graph box` or `graph hbox`. To show that, we must understand how to re-create box plots using `graph twoway`. It is very much a case of *reculer pour mieux sauter*.

1.3 Structure

Let us first remind ourselves of the structure of a box plot by using the life expectancy data shipped with Stata. We will compare life expectancy in 1998 for three groups of countries: in Europe and Central Asia, North America, and South America (figure 1). We use `graph box`. Here and subsequently we will spell out a preference for horizontal axis labels.

```
. sysuse lifeexp
. label var lexp "Life expectancy (years)"
. graph box lexp, over(region) yla(, ang(h))
```

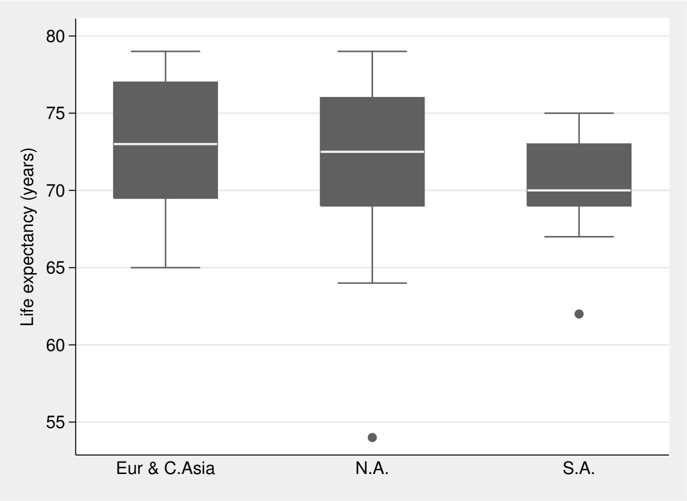

Figure 1. Box plots of life expectancy in 1998 for various countries in three regions

The main ingredient of a box plot is the eponymous box, used to indicate the lower and upper quartiles of the variable or group being plotted against a magnitude scale. The median is represented by a line subdividing the box, or, alternatively, by a point symbol. The length of the box thus represents the interquartile range (IQR). Tukey used a variety of alternative terms for both the quartiles (hinges, fourths, etc.) and their difference, the range or spread between them, but most such terms were adopted only locally or briefly and have long since faded away. It seems simpler now to revert to the classical terms of quartiles and IQR. Whatever the terminology, recall that numerous slightly different calculation rules exist for quartiles and quantiles or percentiles generally (Frigge, Hoaglin, and Iglewicz 1989; Hyndman and Fan 1996). The different rules explain some of the differences in box plots from different software, but otherwise are not of great interest. Stata's rule is set out in [R] **summarize**. Among other details, note that any practical rule must extend to data with weights assigned.

Box plots differ in what else may be shown outside the box. `graph box` and `graph hbox` by default follow what is perhaps the most common recipe (Tukey 1977):

1. Lines, often called whiskers, are drawn to span all data points within 1.5 IQR of the nearer quartile. That is, one whisker extends to include all data points within 1.5 IQR of the upper quartile and stops at the largest such value, while the other whisker extends to include all data within 1.5 IQR of the lower quartile and stops at the smallest such value. Tukey called the outer limits of the whiskers *adjacent values*. The whiskers also explain his alternative term, *box-and-whiskers plots*. Note that either whisker could be of zero length. In practice, that will occur only with very small datasets or heavily tied data.

2. Any data points beyond the whiskers are shown individually and often labeled informatively.

De Veaux, Velleman, and Bock (2008, 81) record Tukey's laconic reply when asked the reason for 1.5: 1 would be too small and 2 would be too large. Evidently, the choice of multiplier gives an informal but objective rule for outlier identification. Any choice is a compromise between revealing too much (flagging data points that are of neither statistical nor scientific concern) and revealing too little (missing data points that require thought or action). Dümbgen and Riedwyl (2007) recently discovered a clever way of justifying 1.5, but experience that it often works quite well is a more compelling basis for the rule.

This kind of box plot, and indeed most other kinds, thus conveys information about level (median); spread (interquartile range and range are both represented directly); symmetry or asymmetry about the median both within and beyond the central half of the data; and, on its own definition, possible outliers. It is thus a fairly information-rich graphical reduction of key quantiles (or of order statistics, if you prefer).

That may be the most common recipe, but many others have been entertained. McGill, Tukey, and Larsen (1978) suggested two refinements: varying the width of boxes to indicate group sizes and notching boxes to indicate approximate confidence intervals. Harris (1999, 57) even reported that some box plots are based at least in part on mean and standard deviation. It is natural to hope that different conventions are all explained clearly for the benefit of readers, but unfortunately, that is often not the case. For example, several authors in the collection edited by Chen, Härdle, and Unwin (2008) use differing varieties of box plots, but the differences are typically unexplained.

However, many variations encountered appear to be essentially cosmetic. In particular, box plots may be horizontal as well as vertical. There can be a small struggle between the convention of showing response or outcome variables increasing vertically and the desire that text labels explaining variables or groups can be spelled out fully and legibly. Whatever the reasoning, Stata users can reach for `graph hbox` if they prefer horizontal alignment. As a matter of careful and conscious design, the change between typing `box` and `hbox` is the only change that need be made. Contrary to mathematical custom, the y axis of box plots in Stata is considered to be whichever axis the response is plotted against. (`graph bar` and `graph hbar` are related in exactly the same way.)

1.4 Utility

Box plots can be very useful, particularly for comparison, especially if the number of variables or groups is nearer 20 or 200 rather than 2. But if you have just a few variables or groups, you have enough space for the greater detail of (say) histograms, dot plots, density traces, or quantile or distribution plots. And because they are reductions of the data, box plots may be uninformative about key details. They tend to perform poorly whenever data are highly skewed—which in many fields is overwhelmingly usual. Naturally, one simple answer to skewness is to transform data. If box plots of a variable are highly asymmetric, then roots or logs or reciprocals are likely to improve matters considerably.

There are deeper problems yet. What is so special about quartiles, in particular? Medians have a clear statistical role as defining midpoints on distribution functions, and they are natural and resistant summaries for (approximately) symmetric distributions. Quartiles take the median idea one step further by being medians of each half of the distribution, but beyond that, their role is much less evident. Simplicity of definition and familiarity from early teaching do not add up to a statistically natural role. In any case, if half the data lie inside the boxes, then half too lie outside the boxes, yet that half—often statistically or scientifically the more important half—is represented in a mostly generalized way within box plots.

So, other quantiles besides quartiles may well be as or more worthy of display. That argument leads ultimately to displaying all quantiles, a tactic discussed in other issues of the *Stata Journal* (Cox 2005, 2007).

With a nod of gratitude for an example given by Wainer (1990, 345), figure 2 points out one further weakness of box plots.

(*Continued on next page*)

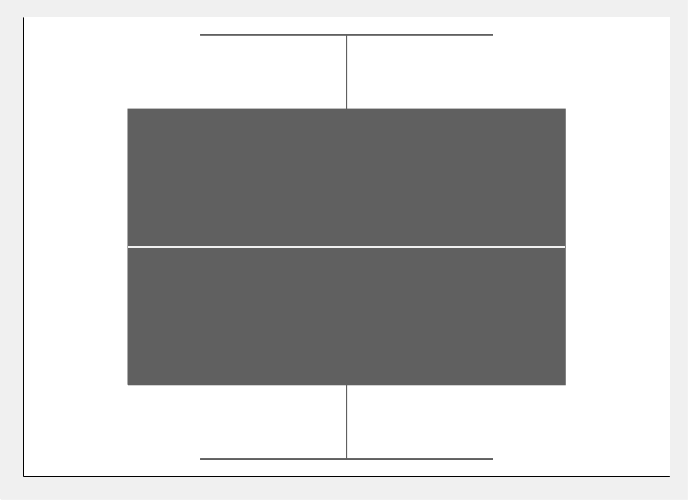

Figure 2. An innocent-looking box plot with a surprise wrapped inside

Asked what can be inferred about the distribution from this plot, even very experienced data analysts typically mutter something about a short-tailed symmetric unimodal distribution. But the box plot clearly implies that the average density in the tails is much greater than that in the middle, so the best inference should be something like a U-shaped distribution. My guess is that although respondents are all familiar with the main idea of box plots, they are being misled by the subdued representation of the tails. Guessing apart, no detailed histogram, density trace, or quantile plot would be guilty of such ambiguity. More generally, box plots inevitably gloss over bimodality or multimodality or granularity of distribution.

To reveal the small surprise, figure 2 is based on a set of quantiles from a beta distribution:

```
. generate y = invibeta(0.6, 0.6, (_n - 0.5) / _N)
```

With these parameter values, the distribution is indeed U-shaped, as the histogram in figure 3 shows more clearly.

```
. histogram y, width(0.1) start(0) horizontal yla(, ang(h))
```

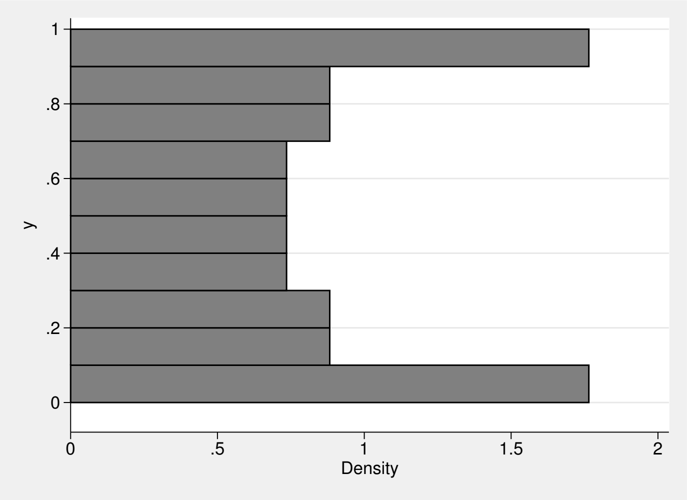

Figure 3. The distribution underlying the innocent-looking box plot: a U-shaped beta distribution

2 Using twoway to create box plots

2.1 Ingredients

To re-create a box plot from scratch given some data, we need to calculate the basic summary statistics. Here the `egen` command is your friend, particularly because its `by()` option allows recording of results for two or more groups. The `by()` option is undocumented in favor of doing things with `by` *varlist*`:`, but it is supported for those `egen` functions of concern to us here. Either way, using `by:` as prefix is exactly equivalent. See the online help or [D] **egen** for more details on that command. A tutorial discussing `egen` is available in Cox (2002).

The median and quartiles are easiest:

```
. egen median = median(lexp), by(region)
. egen upq = pctile(lexp), p(75) by(region)
. egen loq = pctile(lexp), p(25) by(region)
```

We could now get the IQR by subtraction, `upq - loq`, which would be more efficient, but we will mention that it has its own `egen` function.

```
. egen iqr = iqr(lexp), by(region)
```

In fact, we do not strictly need the IQR, as will become clear shortly, but if you like box plots, you might as well know ways of getting the IQR easily into a variable.

The upper and lower limits of the whiskers require a little more thought. Here is one way to get them. The upper limit is the largest value not greater than `upq + 1.5 * iqr`. That can be calculated in one line:

```
. egen upper = max(min(lexp, upq + 1.5 * iqr)), by(region)
```

That one line could bear some deconstruction, however. The outer `max()` is an `egen` function, as the context implies. The inner `min()` is emphatically not another `egen` function, as might be guessed: it is just the standard Stata function `min()`. Why is that allowed here? Because the syntax of `egen` allows here an arbitrary expression, indicated in the syntax diagram by *exp*. Often that expression is just one variable name, but it could be more complicated. Here the entire expression is `min(lexp, upq + 1.5 * iqr)`. The expression could have been `min(lexp, upq + 1.5 * (upq - loq))`, showing that the IQR variable is indeed redundant.

As before, the `by(region)` option ensures that maximums for the expression supplied are calculated separately for each region.

For lower limits of whiskers, we can use the same tactic, except for swapping minimum and maximum:

```
. egen lower = min(max(lexp, loq - 1.5 * iqr)), by(region)
```

We now have in hand all the ingredients we need. But one basic point needs emphasis. By construction, the values for the median, quartiles, and upper and lower limits of the whiskers are repeated for each distinct value of `region`. If instead of comparing groups we were comparing variables, then values would be repeated for each observation. Unless we do nothing further, the graphical consequence will be repeated plotting of the same information, which could be time-consuming and which leads to unnecessarily bloated graph files. It would be important to do something about that in any program with pretensions to efficiency, but for our purposes, we will set this detail aside, beyond noting that `collapse` and `egen, tag()` offer some solutions.

2.2 Assembly

Let us jump immediately to a tolerable mock-up of a box plot and then talk through all the details. Let me also stress that even Stata graph experts never write down code just like this, unless they happen to have solved the problem a few minutes earlier and have excellent memory. To get here requires much experiment and consultation of the help. Figure 4 shows the result.

```
. twoway rbar med upq region, pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rbar med loq region, pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rspike upq upper region, pstyle(p1) ||
> rspike loq lower region, pstyle(p1) ||
> rcap upper upper region, pstyle(p1) msize(*2) ||
> rcap lower lower region, pstyle(p1) msize(*2) ||
> scatter lexp region if !inrange(lexp, lower, upper), ms(Oh) mla(country)
> legend(off)
> xla(1 ˜" "Europe and" "Central Asia" "˜ 2 "North America" 3 "South America",
> noticks) yla(, ang(h)) ytitle(Life expectancy (years)) xtitle("")
```

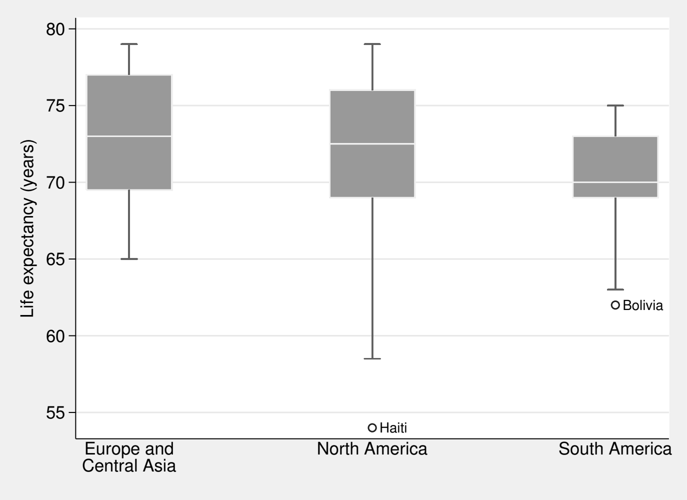

Figure 4. Box plots of life expectancy in 1998 for various countries in three regions, but constructed entirely using `twoway`

Now the commentary:

1. The details may look scary in total, but note first the strategy, which is divide and conquer. Different parts of `twoway` are enlisted to draw different parts of the graph. Similarly, divide and conquer is the strategy to understand the code. There is clearly no need to try to reproduce all the details produced by `graph box` if you prefer something different.

2. `region` is a numeric variable, so we can plot against it. Its values are 1, 2, and 3, and value labels are attached, so it is already in good condition for graphics. If you had a variable that was not in good condition, say, because it was a string variable or a numeric variable that needed tidying up, then creating a new variable with `egen, group()` with its `label` option is the best way to proceed. `encode` is an alternative for string variables.

3. `pstyle(p1)` is a simple trick to enforce general consistency of style. You can then depart from whatever results in your preferred directions.

4. The boxes are drawn with `twoway rbar`, one from the median to the upper quartile and one from the median to the lower quartile. A light outline color, `blcolor(gs15)`, is sufficient to indicate where the medians are. I chose as a matter of personal taste a lighter color for the bar fill than the default in the `sj` scheme. Light color for fill and dark color for outline are equally acceptable statistically, and perhaps preferable aesthetically. `barwidth(0.35)` reflects my personal taste: I regard the default boxes of `graph box` as a little fat. If the values of the categorical variable did not differ by 1, a quite different bar width would be needed.

5. The whiskers are drawn with `twoway rspike`, one from the lower quartile to the lower end of the whiskers, and one from the upper quartile to their upper end.

6. The whiskers are capped using `twoway rcap`. Note that there is no typo in `rcap upper upper region` or `rcap lower lower region`. The code was not `rcap upq upper region` or `rcap loq lower region` to ensure that no caps are visible interfering with the box. The marker size is twice default, but even so is much less than the default of `graph box`, to say nothing of what can be obtained using its `capsize()` option.

7. Clearly, the caps could be omitted if so desired, simply by omitting the calls to `twoway rcap`. Why does the standard box plot design include them? It seems to be an admission of weakness, namely, that the whiskers might be overlooked if the graph did not emphasize where they end.

8. Data points beyond the whiskers are shown using `scatter`. Hollow circles given by `ms(Oh)` are a personal choice as suitably prominent yet tolerating overlap well (think of the overlapping rings of the Olympic symbol). Note the simple logic: points within the range of the boxes and whiskers are `inrange(lexp, lower, upper)` and so points beyond them are the logical complement, obtained by negation, `!`. See Cox (2006) for more on `inrange()` if so desired. Putting this into words as “not in range” is a simple way of underlining what is being done.

9. Such data points are labeled using marker labels, `mla(country)`. In this case, defaults work fine. In other cases, we might want to tune marker label size or other properties, as later examples will make clear.

10. All the different `twoway` calls produce a complicated `legend`, which we just suppress. So many different variables are being portrayed, from `twoway`'s point of view, that we have to add our own y-axis title.

11. In this particular case, the value labels attached to `region` are over-abbreviated, so we step in and provide our own. I agree with the designer of `graph box` that axis ticks serve no useful purpose when distinct categories are being shown. The default `xtitle()` would be the variable name `region`, which also is dispensable here. (In other contexts, I routinely suppress variable names indicating date or year when axis labels such as 1990 or 2000 make abundantly clear what is being shown.)

2.3 Horizontal

Clearly, we need to know how to produce horizontal box plots too. Here is a first stab, with the result in figure 5:

```
. twoway rbar med upq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rbar med loq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rspike upq upper region, horiz pstyle(p1) ||
> rspike loq lower region, horiz pstyle(p1) ||
> rcap upper upper region, horiz pstyle(p1) msize(*2) ||
> rcap lower lower region, horiz pstyle(p1) msize(*2) ||
> scatter region lexp if !inrange(lexp, lower, upper), mla(country) legend(off)
> yla(1 `" "Europe and" "Central Asia" "' 2 "North America" 3 "South America",
> ang(h) noticks) xtitle(Life expectancy (years)) ytitle("")
```

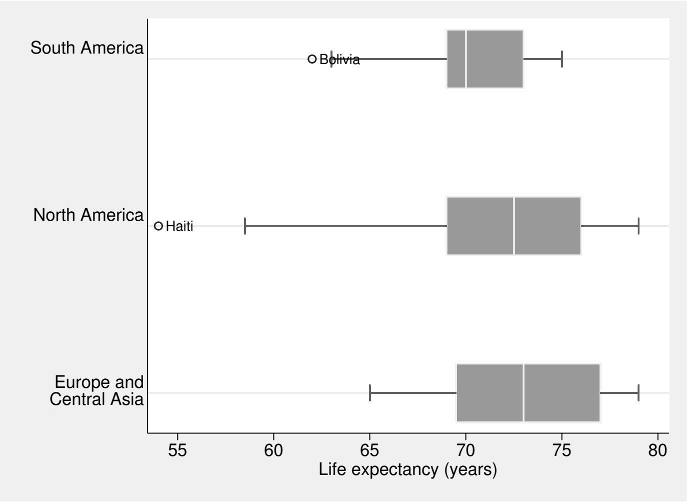

Figure 5. Horizontal box plots of life expectancy in 1998 for various countries in three regions, but constructed entirely using `twoway`

The necessary changes are to add the `horizontal` option to calls to `rbar`, `rspike`, and `rcap` and to swap *y* and *x* within the call to `scatter` (variables are swapped, `x` options become `y` options, and vice versa).

The result is most of the way to where we want to be. The marker labels would be better lifted clear of the whiskers. The *x* axis also needs to be lengthened a little to give enough space for the text `Haiti`. A little experiment shows that the extra options `xsc(r(53,.))`, `mlabpos(12)`, and `mlabgap(1.5)` give those improvements; see figure 6.

(*Continued on next page*)

```
. twoway rbar med upq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rbar med loq region, horiz pstyle(p1) blc(gs15) bfc(gs8) barw(0.35) ||
> rspike upq upper region, horiz pstyle(p1) ||
> rspike loq lower region, horiz pstyle(p1) ||
> rcap upper upper region, horiz pstyle(p1) msize(*2) ||
> rcap lower lower region, horiz pstyle(p1) msize(*2) ||
> scatter region lexp if !inrange(lexp, lower, upper), mla(country)
> mlabpos(12) mlabgap(1.5)  xsc(r(53, .)) legend(off)
> yla(1 ˆ" "Europe and" "Central Asia" "ˆ 2 "North America" 3 "South America",
> ang(h) noticks) xtitle(Life expectancy (years)) ytitle("") ;
```

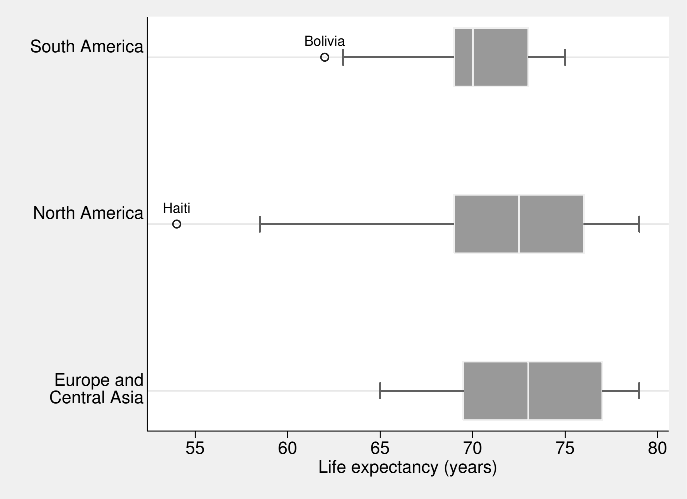

Figure 6. Horizontal box plots of life expectancy in 1998 for various countries in three regions, with improved positioning of marker labels for outliers

3 Moving beyond standard designs

Provided that you are broadly familiar with how twoway works, you should now have a sense that a small new world is open before you, in which you can add to, subtract from, or otherwise vary box plot designs exactly as you wish. If you do this repeatedly, you will want to encapsulate code for favored designs in a do-file or program. Explaining that further would take us beyond the main story, but both the User's Guide and Baum (2009) are excellent sources of advice and examples.

3.1 Adding means

One common request, on Statalist and elsewhere, is to add means to box plots. For this, you need an extra variable containing means. egen is again convenient:

```
. egen mean = mean(lexp), by(region)
```

We need to add a `scatter` call to the code above:

```
scatter region mean, ms(Dh) msize(*2) ||
```

A simple but crucial detail is plotting the means after, and therefore on top of, the boxes. Usually, although not inevitably, means will lie between the quartiles, and so their symbols would disappear under the boxes otherwise. Figure 7 shows the result.

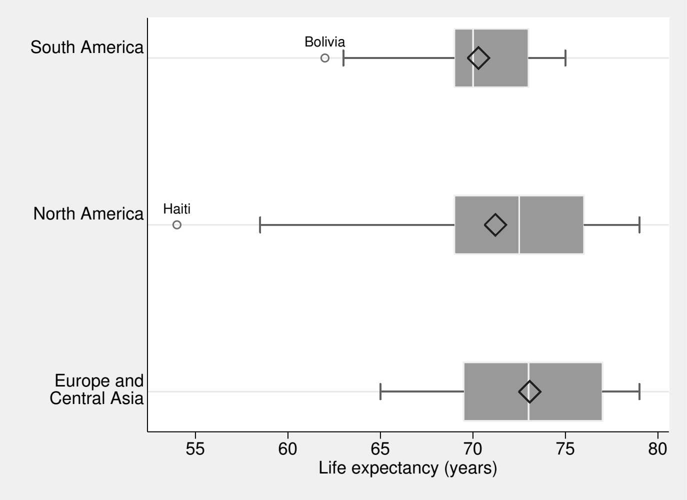

Figure 7. Horizontal box plots of life expectancy in 1998 for various countries in three regions; diamond symbols indicate means

3.2 Connecting medians

Another common request is connecting medians. One context for this could be that the box plots indicate variation within time periods. The connected medians thus would emphasize variation between time periods. This request is met as previously, by adding another `twoway` call such as

```
line median timevar, lw(*2)
```

or

```
line timevar median, lw(*2)
```

depending on whether plots are vertical or horizontal. Emphasis is added if and as desired, here by doubling line width. As before, plot connecting lines after, and so on top of, boxes.

3.3 Unequal spacing

Nothing in the `twoway` route to box plots commits you to equal spacing of box plots. Unequal spacing is perfectly possible: you just specify the positions of the box plots. Binning of responses or residuals in unequal intervals of a covariate is one large class of possible examples.

3.4 Variable width

If there were a desire for boxes of variable width, that could be met by repeated calls to `twoway rbar` with differing `barwidth()` options. `barwidth()` requires a single number as argument, and does not accept a numeric variable indicating width.

3.5 Percentile-based whiskers

Let us now imagine a different design in which whiskers are drawn out to 10% and 90% points. Cleveland (1985) showed such box plots. They have three advantages over the standard design. First, the definition of whiskers is of the same kind as the definition of boxes. Second, almost always, we see some detail in the tails. The exceptions when there is heavy tying in one or the other tail are also discernible. Third, to a very good approximation, drawing such box plots commutes with any monotonic transformation so that, for example, the box plot of a logged variable is the log of the box plot of the variable on the original scale. Some minor inaccuracy may arise in practice because quantiles may be calculated as the average of two order statistics: see the FAQ at http://www.stata.com/support/faqs/graphics/boxandlog.html for more on this thorny little detail.

Evidently, the choice of 10% and 90% is in no sense compulsory: other values may suit some purposes better.

We will also ensure that all points outside the whiskers are labeled. Because we are in complete control, we will go back to vertical, reverse box coloring and drop those whisker caps that we do not much like.

We know how to get further percentiles:

```
. egen p10 = pctile(lexp), p(10) by(region)
. egen p90 = pctile(lexp), p(90) by(region)
```

There are no new tricks needed for the graph, or so we might think; see figure 8.

```
. twoway rbar med upq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rbar med loq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rspike upq p90 region, pstyle(p1) ||
> rspike loq p10 region, pstyle(p1) ||
> scatter lexp region if !inrange(lexp, p10, p90), ms(Oh) mla(country)
> mlabgap(1.5) legend(off)
> xla(1 `" "Europe and" "Central Asia" "' 2 "North America" 3 "South America",
> noticks) yla(, ang(h)) ytitle(Life expectancy (years)) xtitle("")
```

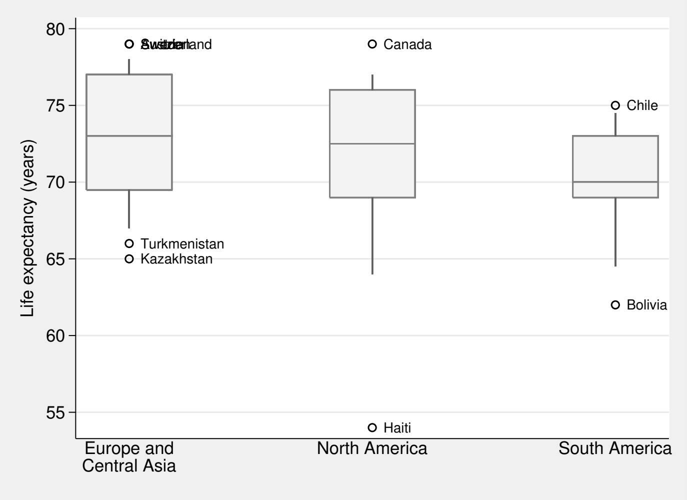

Figure 8. Box plots of life expectancy in 1998 for various countries in three regions; whiskers extend to 10% and 90% points of the distribution

A little mess of marker labels turns out to arise because Austria, Sweden, and Switzerland tie at 79 years. Some experimenting indicates that we can just rotate two of those labels away from the default position. Figure 9 is the improved graph.

```
. gen pos = cond(country == "Austria", 1, cond(country == "Sweden", 4, 3))
. twoway rbar med upq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rbar med loq region, pstyle(p1) bfc(gs15) blc(gs8) barw(0.35) ||
> rspike upq p90 region, pstyle(p1) ||
> rspike loq p10 region, pstyle(p1) ||
> scatter lexp region if !inrange(lexp, p10, p90), ms(Oh) mla(country)
> mlabgap(1.5) legend(off) mlabvpos(pos)
> xla(1 `" "Europe and" "Central Asia" "' 2 "North America" 3 "South America",
> noticks) yla(, ang(h)) ytitle(Life expectancy (years)) xtitle("")
```

(*Continued on next page*)

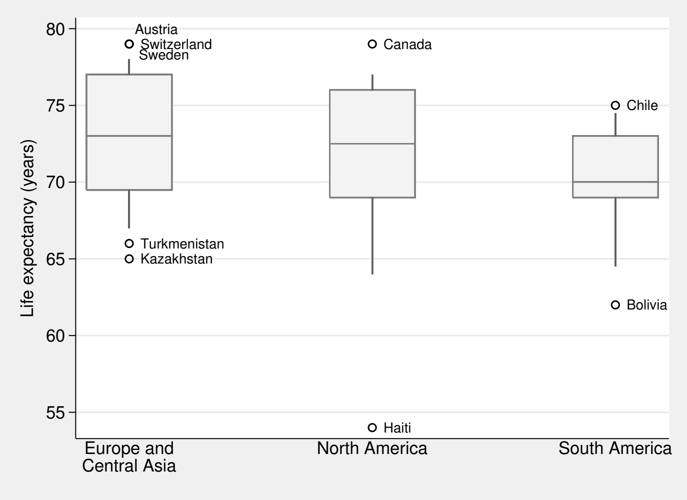

Figure 9. Box plots of life expectancy in 1998 for various countries in three regions; whiskers extend to 10% and 90% points of each distribution; marker labels for Austria and Sweden have been moved to avoid overlap

3.6 Other data structures

So far, we have considered only the case of one response variable, subdivided by groups of a categorical variable. Box plots are often needed for other data structures. We need to see that they are also within reach given a little technique.

Another dataset shipped with Stata contains temperature data for 956 cities in the United States, including variables `tempjan` and `tempjuly` indicating mean monthly temperatures for January and July. The cities are classified coarsely by `region` and more finely by `division`. We will produce box plots of the Cleveland (1985) kind for the two temperature responses `tempjan` and `tempjuly`, subdivided by division and month.

We could just superimpose box plots for `tempjan` and `tempjuly`, but a `reshape` of the data makes matters easier thereafter. `reshape` requires a unique identifier, so we put the observation number into a new variable to act as a pacifier. The identifier will play no role thereafter in our graphics. See the online help or [D] **reshape** if you need more discussion.

```
. sysuse citytemp, clear
. gen id = _n
. reshape long temp, i(id) string j(month)
```

The new string variable `month` takes on two values, `jan` and `july`. The summary statistics come from `egen`. The extra twist that we need to distinguish both `division` and `month` is easily satisfied:

```
. egen median = median(temp), by(division month)
. egen loq = pctile(temp), p(25)  by(division month)
. egen upq = pctile(temp), p(75)  by(division month)
. egen p10 = pctile(temp), p(10)  by(division month)
. egen p90 = pctile(temp), p(90)  by(division month)
```

`division` is an integer variable with values from 1 to 9 and value labels attached. To show box plots for January and July side by side, we just need a position variable in which months are offset. Cui (2007) gives further discussion of this simple trick. We still want to use the value labels of `division`, so we assign them to the new variable.

```
. gen division2 = division + cond(month == "jan", -0.2, 0.2)
. label val division2 division
```

The code is now very similar to previous examples. Figure 10 gives the result.

```
. twoway rbar median upq division2, bfc(gs15) blc(gs8) barw(0.35) ||
> rbar median loq division2, bfc(gs15) blc(gs8) barw(0.35) ||
> rspike loq p10 division2 ||
> rspike upq p90 division2 ||
> scatter temp division2 if !inrange(temp, p10, p90), ms(o) legend(off)
> xaxis(1 2) xla(1/9, valuelabel noticks grid axis(1))
> xla(1/9, valuelabel noticks axis(2)) xtitle("", axis(1)) xtitle("", axis(2))
> yaxis(1 2) yla(14(18)86, ang(h) axis(2))
> yla(14 "-10" 32 "0" 50 "10" 68 "20" 86 "30", ang(h) axis(1))
> ytitle(mean temperature ({c 176}F), axis(2))
> ytitle(mean temperature ({c 176}C), axis(1))
> ysc(titlegap(0) axis(1)) ysc(titlegap(0) axis(2))
> plotregion(lstyle(p1))
```

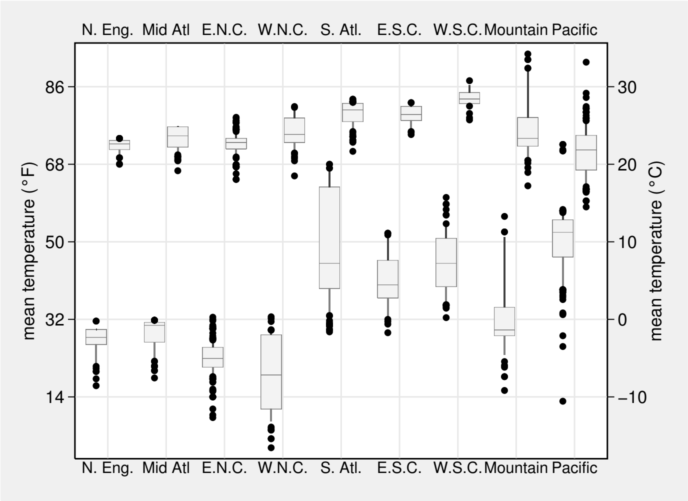

Figure 10. Box plots of mean temperatures in January (left plot in each group) and July (right plot in each group) for various places in divisions of the United States; whiskers extend to 10% and 90% points of each distribution

Here we know that each division will take up less space than in previous graphs, so we bump up the marker symbol to `ms(o)` so that they remain visible.

We also add a few `twoway` flourishes. Horizontal axis labels at the top as well as the bottom of the graph ease look-up of division labels. 0°F is of less importance than 32°F as a reference temperature. We also align equivalent temperatures on Fahrenheit and Celsius scales. Note the trick to get the degree symbol (Cox 2004).

Incidentally, when we look at the box plot to learn something about the data, we see that the upper tail of Mid-Atlantic July temperatures is curiously truncated. Inspection of the data shows that 20 places are all given a mean July temperature of 76.8°F. This is the highest temperature observed for that division but is also the 90% point, because those 20 are more than 10% of the places in the division. Thus no places are plotted as having a higher temperature than the 90% point. Hence the graph is correct in terms of the data, but the graph has also told us something new about the data, which is as it should be. People more familiar with how the U.S. Census reports temperature data may be able to throw more light on this little mystery.

3.7 Convenience and efficiency

The stress in this column has been on getting results conveniently using standard commands. It is nevertheless proper to repeat a note of caution sounded earlier. The commands used here are not the most efficient way to get box plots, nor will the graph files produced be as lean as they could be. For small or moderate datasets, you would have to strain to notice that, but otherwise you might be bitten. Industrial-strength alternatives to these commands would need to work at lower levels to optimize speed and storage, by replacing calls to `egen` with direct calls to `summarize` and by ensuring that the information defining box plot ingredients is not duplicated unnecessarily.

4 Conclusion

`graph box` and `graph hbox` are very useful commands, but they only do what they claim to do. This column has shown an alternative way to create, and then to vary, box plots, using `egen` for calculations and `twoway` for graphics. Once the problem is broken down into components, it can be solved without any programming. That then gives researchers scope for whatever variants of box plots are likely to prove interesting and useful. Cleveland's 1985 variant seems especially worthy of further consideration.

5 Errata to previous column

Marcel Zwahlen helpfully pointed out an inconsistency between an equation on p. 308 and the corresponding Mata code on p. 309 within the previous column (Cox 2009). The equation was incorrect.

$$K = \frac{N^2 - \sum_{i=1}^{I} n_i^2}{\sum_{i=1}^{I} n_i - Nq_i}$$

should have been

$$K = \frac{N^2 - \sum_{i=1}^{I} n_i^2}{\sum_{i=1}^{I} (n_i - Nq_i)^2}$$

6 References

Baum, C. F. 2009. *An Introduction to Stata Programming*. College Station, TX: Stata Press.

Chen, C., W. Härdle, and A. Unwin, ed. 2008. *Handbook of Data Visualization*. Berlin: Springer.

Cleveland, W. S. 1985. *The Elements of Graphing Data*. Monterey, CA: Wadsworth.

Cox, N. J. 2002. Speaking Stata: On getting functions to do the work. *Stata Journal* 2: 411–427.

———. 2004. Stata tip 6: Inserting awkward characters in the plot. *Stata Journal* 4: 95–96.

———. 2005. Speaking Stata: The protean quantile plot. *Stata Journal* 5: 442–460.

———. 2006. Stata tip 39: In a list or out? In a range or out? *Stata Journal* 6: 593–595.

———. 2007. Stata tip 47: Quantile–quantile plots without programming. *Stata Journal* 7: 275–279.

———. 2009. Speaking Stata: I. J. Good and quasi-Bayes smoothing of categorical frequencies. *Stata Journal* 9: 306–314.

Cox, N. J., and K. Jones. 1981. Exploratory data analysis. In *Quantitative Geography: A British View*, ed. N. Wrigley and R. J. Bennett, 135–143. London: Routledge and Kegan Paul.

Crowe, P. R. 1933. The analysis of rainfall probability: A graphical method and its application to European data. *Scottish Geographical Magazine* 49: 73–91.

Cui, J. 2007. Stata tip 42: The overlay problem: Offset for clarity. *Stata Journal* 7: 141–142.

De Veaux, R. D., P. F. Velleman, and D. E. Bock. 2008. *Stats: Data and Models*. 2nd ed. Boston, MA: Addison–Wesley.

Dümbgen, L., and H. Riedwyl. 2007. On fences and asymmetry in box-and-whiskers plots. *American Statistician* 61: 356–359.

Freedman, D., R. Pisani, and R. Purves. 2007. *Statistics.* 4th ed. New York: W. W. Norton.

Frigge, M., D. C. Hoaglin, and B. Iglewicz. 1989. Some implementations of the boxplot. *American Statistician* 43: 50–54.

Harris, R. L. 1999. *Information Graphics: A Comprehensive Illustrated Reference.* New York: Oxford University Press.

Hoaglin, D. C., F. Mosteller, and J. W. Tukey, ed. 1983. *Understanding Robust and Exploratory Data Analysis.* New York: Wiley.

Hyndman, R. J., and Y. Fan. 1996. Sample quantiles in statistical packages. *American Statistician* 50: 361–365.

Kleiner, B., and T. E. Graedel. 1980. Exploratory data analysis in the geophysical sciences. *Reviews of Geophysics* 18: 699–717.

McGill, R., J. W. Tukey, and W. A. Larsen. 1978. Variations of box plots. *American Statistician* 32: 12–16.

Mitchell, M. 2008. *A Visual Guide to Stata Graphics.* 2nd ed. College Station, TX: Stata Press.

Monkhouse, F. J., and H. R. Wilkinson. 1971. *Maps and Diagrams.* London: Methuen.

Spear, M. E. 1952. *Charting Statistics.* New York: McGraw–Hill.

———. 1969. *Practical Charting Techniques.* New York: McGraw–Hill.

Tukey, J. W. 1972. Some graphic and semigraphic displays. In *Statistical Papers in Honor of George W. Snedecor*, ed. T. A. Bancroft and S. A. Brown, 293–316. Ames, IA: Iowa State University Press.

———. 1977. *Exploratory Data Analysis.* Reading, MA: Addison–Wesley.

Velleman, P. F., and D. C. Hoaglin. 1981. *Applications, Basics, and Computing of Exploratory Data Analysis.* Boston, MA: Duxbury.

Wainer, H. 1990. Graphical visions from William Playfair to John Tukey. *Statistical Science* 5: 340–346.

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*.