The Stata Journal (2009)
9, Number 4, pp. 621–639

Speaking Stata: Paired, parallel, or profile plots for changes, correlations, and other comparisons

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

Abstract. Paired, parallel, or profile plots showing the values of two variables may be constructed readily using a combination of `graph twoway` commands. This column explores the principles and practice of such plot-making, considering both wide and long (panel or longitudinal) data structures in which such data may appear. Applications include analysis of change over time or space and indeed any kind of correlation or comparison between variables. Such plots may be extended to show numeric values and associated name information.

Keywords: gr0041, profile plot, parallel coordinates plot, parallel line plot, pair-link diagram, bumps charts, barometer charts, graphics, panel data, longitudinal data, arrows, twoway

1 Introduction

1.1 Graphical comparison of two variables

Comparison of two variables is a graphical problem that arises in many different situations.

The variables are often raw data, but need not be. They could be, for example, sets of summary statistics or quantities calculated from fitting one or more models (residuals, predicted values, figures of merit, etc.). Here the focus is on examples in which the variables, however defined, are recorded on identical (or at least comparable) scales. That restriction is not strong, as it could easily be satisfied by some kind of standardizing or ranking.

The situations could also vary. They include studies of change over time or space, correlations between variables, and other kinds of comparison. However, the graphical problem is much the same.

This column is a sequel to an earlier discussion of graphing agreement and disagreement (Cox 2004). After that column was published, Stata 9 added a set of paired-coordinate commands to `graph twoway`, which makes several pertinent graphs much easier. Cox (2005b) publicized the use of `twoway pcarrow` for graphing changes over time, but the wider possibilities still deserve attention. Although user-written commands are available, the emphasis here is on working out how to plot data yourself with `twoway` from first principles.

1.2 Pairs, parallel coordinates, and profiles

The main kind of graphs under consideration goes under several different names, often emphasizing variously the paired, parallel, or profile character of the plot. You may know yet other names used in your field. Glance ahead to figures 3, 4, 7, and 8 to get a picture of what is being talked about.

Whatever the name, such graphs have a long history. Friendly (2007) drew attention to the use by Guerry (1833) of what many now call a parallel coordinates plot for comparing relative frequency of crimes at different ages. Friendly (2008) adds further nineteenth century examples. Such plots have been the subject of many recent accounts, ranging from introductory (Robbins 2005; Few 2009) through intermediate (Wilkinson 2005; Unwin, Theus, and Hofmann 2006; Cook and Swayne 2007; Chen, Härdle, and Unwin 2008; Theus and Urbanek 2009) to more advanced (Inselberg 2009).

Interaction plots long common in looking at analyses of variance (e.g., Cox [1958]) could be considered as a variation on the main idea.

"Bumps charts" are a version of parallel coordinates plots that often appear in newspapers or on the web. Such charts originally showed changes in rank in series of rowing races held at Oxford and Cambridge and also elsewhere. Given relatively narrow rivers, boats start in single file and crews aim to overtake or "bump" boats in front and not be overtaken from behind. Tufte (1990, 111; 2006, 56) helped publicize bumps charts to a wider readership. The term is now often used beyond its sporting origins.

Another very common variant is widely known as a profile plot. For example, in behavioral research, human or animal subjects may be monitored through time or according to various tests or measures. Data for each subject are plotted as a connected line or profile. That usage is broadly consistent with others. In some Earth or environmental sciences, profiles record variation in properties such as surface altitude along paths or transects in space (e.g., Cox [1990]). Basford and Tukey (1997, 1999) made extensive use of profile plots in a major graphically-based analysis of a plant breeding trial.

Campbell and Kenny (1999) showed how such graphs, which they called pair-link diagrams, could be used in discussing regression artifacts such as regression toward the mean. Wallgren et al. (1996) used the term barometer charts, while Harris (1999) wrote of comparative graphs.

Wilkinson (2005, 314) suggested a distinction: profile plots have a common measurement scale but parallel coordinates plots do not. This distinction is puzzling (why plot at all if the scale is not common in some sense?) and, more crucially, it appears to be neither preached nor practiced widely. To muddy the waters further, further senses of profile plots can be found in the literature. Ramsey and Schafer (2002) use the term for graphs in which each individual is plotted in a separate panel so that profiles are not superimposed. On the other hand, du Toit, Steyn, and Stumpf (1986) refer to both separate and superimposed traces as profiles.

Parallel coordinates and profile plots lend themselves easily to plotting several variables simultaneously, which is a major attraction. Indeed, that is exactly where most treatments start. However, problems with just two variables are sufficiently common to merit detailed attention. If data for only two variables are being plotted, there is usually scope to elaborate the graph by adding information on (say) identifying names or the values themselves. Conversely, a common criticism of such plots (e.g., Venables and Ripley [2002]; Cox [2004]; Young, Valero-Mora, and Friendly [2006]) is that they may become busy and confusing. Naturally, the aim is to avoid, or at least to reduce, such confusion and to turn to other graph forms if they work better for some problems.

2 Treatments for anorexia

2.1 The example data

An interesting dataset that provides a suitable example comes from Hand et al. (1994, 229). They reported some data from Brian S. Everitt on weights of young girls receiving different treatments for anorexia. The weights were said to be in kg, but are clearly in pounds (lb), as McNeil (1996, 57) also commented. The weights are reported for before and after various treatments: cognitive behavioral therapy, control, and family therapy. Hand et al. also comment: "Whichever statistical technique is employed, it is instructive to look at the three scatterplots of after/before." The data are provided with the media for this issue. They are also available at http://www.stat.ucla.edu/data/hand-daly-lunn-mcconway-ostrowski/ANOREXIA.DAT.

2.2 Scatterplot

Let us start with the advice given. Figure 1 is a scatterplot.

```
. use anorexia
. scatter after before before, ms(Oh i) c(. l) lc(none gs12)
> sort(before) yla(, ang(h)) ytitle(after)
> by(treatment, row(1) note("weight, lb") legend(off))
```

(*Continued on next page*)

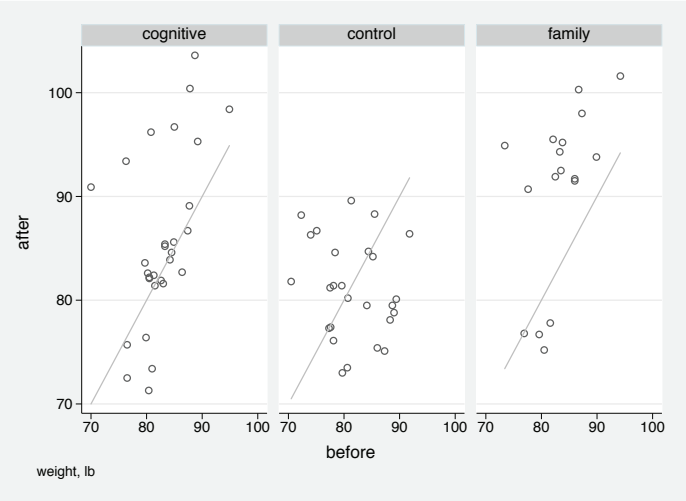

Figure 1. Weights of anorexic girls before and after various treatments. Diagonal lines mark no change in weight.

This scatterplot plots `after` against `before` and adds reference lines $y = x$ by also plotting `before` versus itself. The data are plotted separately by treatment in one row, `by(treatment, row(1))`. The reference lines are plotted with subdued color, `lc(gs12)`, while `sort(before)` ensures that they are plotted smoothly. Other options above tune cosmetic details. Alphabetical order leaves control subjects in the middle, which can be regarded as fortunate.

To get a feel for magnitudes, readers in most countries may like to know that 30 (40, 50) kg are about 66 (88, 110) pounds, spanning the range shown.

The scatterplot does show clearly the broad features of the data. The impressions are that control subjects are about equally divided between weight gainers and losers. Most subjects gained weight with family therapy, but a distinct group lost weight substantially. Weights generally improved with cognitive behavioral therapy, but there were also several exceptions.

2.3 Parallel coordinates plots

A limitation of the scatterplot is that change in weight, after − before, is encoded only indirectly, despite being the response measure of most interest. A parallel coordinates plot is a move toward more direct encoding. For such a plot, we already have the variables `before` and `after` to serve as parallel y coordinates; we just need to construct the corresponding x coordinates. Convention puts `before` to the left of `after` and convenience leads to a choice such as

```
. gen byte one = 1
. gen byte two = 2
```

Sticklers for style and efficiency will appreciate that those variables are produced as `byte`. The names are arbitrary, and we will make sure that graph readers never see them.

As a first stab, we will show paired values by spikes using `twoway pcspike`. Spikes here are straight line segments with no symbol at either end. For more about the alternatives, start with [G] **graph twoway** and look at the list of other paired-coordinate graph types. We will shortly look at `twoway pcarrow`. Figure 2 is the result of plotting changes as spikes.

```
. twoway pcspike before one after two,
> xla(1 "before" 2 "after") xtitle("") yla(, ang(h)) ytitle("weight, lb")
```

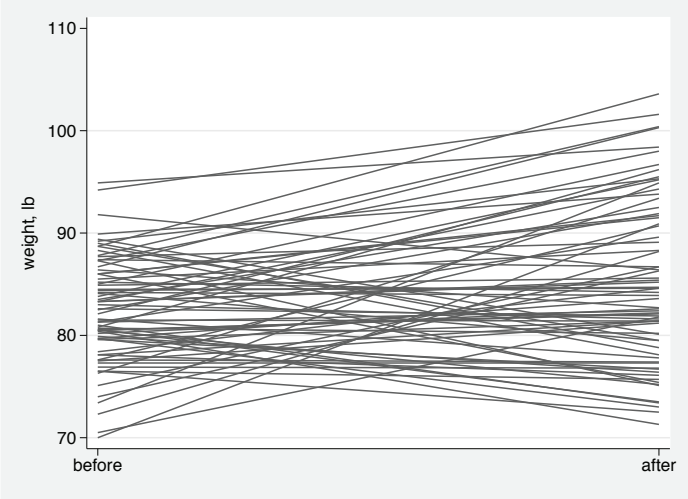

Figure 2. Rough parallel coordinates or profile plot for weights of anorexic girls before and after various treatments

The result is rather busy. Clearly, we want to move on to a graph that separates the different treatments. Figure 3 is the result.

```
. twoway pcspike before one after two,
> xla(1 "   before" 2 "after   ")) xtitle("")
> yla(, ang(h) nogrid) ytitle("weight, lb")
> by(treatment, row(1) noixtick legend(off))
```

(*Continued on next page*)

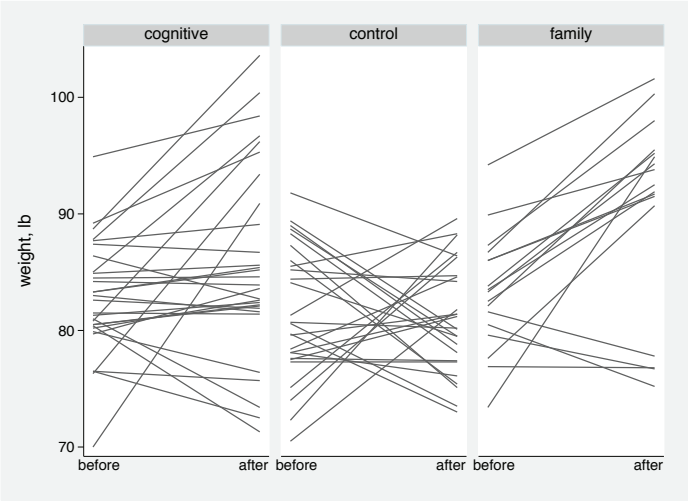

Figure 3. Improved parallel coordinates or profile plot for weights of anorexic girls before and after various treatments

When we do that, we add some small changes. The labels `before` and `after` are pushed inward with extra spaces. The associated ticks do no good, so they are suppressed by the `noixtick` suboption. The grid of horizontal lines is also a distraction given the graph style of line segments. The legend is suppressed, as in figure 1.

We can go further. A basic feature that bears a little emphasis is the contrast between gainers and losers of weight. Two simple choices are to emphasize a minority (the converse can be too loud) and to emphasize any group that is scientifically interesting or practically important. The two choices give the same answer here: stress those who lost weight.

In this column, the distinction is made by line width and grayscale color (Cox 2009). There is also freedom to vary line pattern. You may well have greater freedom yet, say, to choose bolder colors for a presentation.

```
. twoway pcspike before one after two if before <= after, lcolor(gs12) ||
> pcspike before one after two if before > after,
> lw(*1.2) lcolor(gs2) xla(1 "   before" 2 "after   ")
> xtitle("") yla(, nogrid ang(h)) ytitle("weight, lb")
> by(treatment, row(1) note("") legend(off) noixtick)
```

Figure 4 is the result.

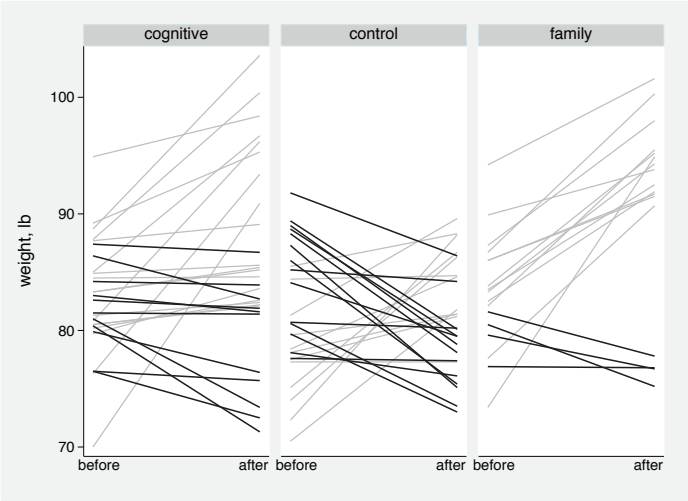

Figure 4. Further improved parallel coordinates or profile plot for weights of anorexic girls before and after various treatments. Girls who lost weight are emphasized.

An obvious but nevertheless common small error is to forget about equal values. That is, `>` and `<` usually need to include `=` on one side or the other. The exceptions are when you really do intend to look only at cases that changed one way or the other and omit unchanged cases from the graph.

Whatever you want to emphasize should be plotted second, that is, on top, so that any overwriting on the graph is in its favor. Thus the trick is to plot spikes for the losers on top of spikes for the gainers, and thicken the lines and darken the color for the losers.

2.4 Parallel line plots

Another plot form that shows after − before even more directly is the parallel line plot (McNeil 1992, 1996; Cox 2004, 2006). The version here capitalizes on the rough equality of treatment numbers, but if groups were very unequal, that could be accommodated otherwise. There are no identifiers or other variables in the published data, so sorting might as well be by weight, either before or after treatment. Any ties should be sorted tidily. A horizontal coordinate can then be the order by weight, except that subtracting the approximate mean rank will center each display.

Weight before is perhaps the more obvious choice.

```
. bysort treatment (before after) : gen order1 = _n - _N/2
```

Readers needing more information on the `by:` prefix—including the principle that under `by:`, the built-in variables `_n` and `_N` are interpreted within groups—can find a tutorial in Cox (2002).

```
. twoway pcarrow before order1 after order1, pstyle(p1)
> || scatter before order1, pstyle(p1) ms(o)
> xla(none) xtitle("") yla(, ang(h)) ytitle("weight, lb")
> by(treatment, row(1) note("") legend(off))
```

Figure 5 is the result.

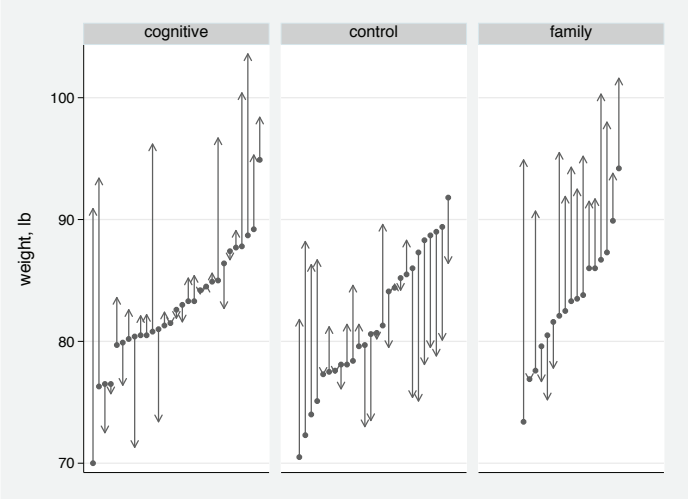

Figure 5. Parallel line plot for weights of anorexic girls before and after various treatments. Point symbols indicate weights before treatment. Arrows point in the direction of change.

We turn to arrows to show change (Cox 2005b) but add marker symbols for the weights before treatment to give slight emphasis to the distribution as a set. Thus each panel is a quantile plot (Cox 2005a), together with added vectors of change. `pstyle(p1)` is a trick to ensure consistent style for both the arrows and the scatterplot. The x axis does have a meaning as rank order, but that remains tacit.

It is also possible to adopt the opposite point of view. We may focus on the end or later result and ask: How did the individuals get to here? In education, for example, you might focus on exit grade-point average or other final achievement measure, as compared with entry data. The code is simply a variation on that just seen. Figure 6 is the result.

```
. bysort treatment (after before) : gen order2 = _n - _N/2
. twoway pcspike before order2 after order2, pstyle(p1)
> || scatter after order2, pstyle(p1) ms(o)
> xla(none) xtitle("") ytitle("weight, lb") yla(, ang(h))
> by(treatment, row(1) note("") legend(off))
```

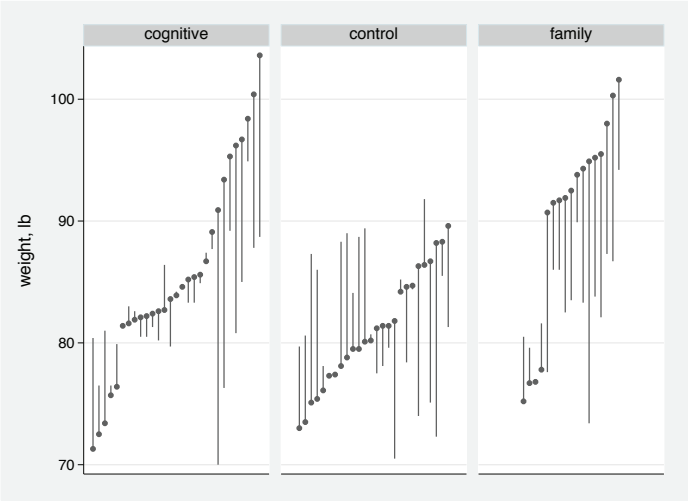

Figure 6. Parallel line plot for weights of anorexic girls before and after various treatments. Point symbols indicate weights after treatment. Spikes indicate the change from before treatment.

You can see one detailed change: spikes were used rather than arrows to avoid the graph becoming too busy around the data points. Clearly, you could have arrows if you wanted them, and you could tweak the arrowhead display to make it just subtly noticeable.

Tastes and judgments will differ, but these two last graphs are my favorites for these data. Other graphs could be shown too, say, box plots (as in McNeil [1996]) or quantile–quantile plots. However, neither of those graph forms respects the pairing in the data. It is true that figures 5 and 6 may not extend easily to series of three or more measurements. Figure 4 would be much more flexible in that regard. However, exploiting the structure of the data to a good end is entirely fair play.

3 Panel or longitudinal structure

Many readers will have been surprised that these data have not yet been described as panel or longitudinal data and treated as such. They will regard using what they recognize as a wide data structure (each girl as one observation) as perverse and prefer a long structure (each girl as two observations). Nothing in the information indicates that the times of measurements before and after were the same, or even equally spaced, so any times assigned are wholly relative and arbitrary. But that corresponds exactly to how the data have been treated so far.

3.1 Mapping to long data structure

If we back-tracked to have only the original data in memory, then there are at least two ways to map the data to long structure.

```
. keep treatment before after
```

Either way, we need an identifier to help Stata keep track, even if we have to invent it ourselves.

```
. gen id = _n
```

The first way to restructure, and probably the better known among Stata users, is to `reshape`. Stata will not regard `before` and `after` as cognate unless their names share a prefix:

```
. rename before weight1
. rename after weight2
. reshape long weight, i(id) j(time)
```

The second way would be to use `stack`:

```
. stack id before treatment id after treatment, into(id weight treatment) clear
. rename _stack time
```

Regardless of how you restructure, and of whether you declare the panel data as such using `tsset` or `xtset`, it is a good idea to sort the data now:

```
. sort id time
```

3.2 Graphs from long data structure

Turning now to graphs, first note that `xtline` is a dead end for our problem, because it does not support a `by()` option. We can easily work from first principles instead.

```
. line weight time, c(L) xla(1 "   before" 2 "after   ") xtitle("")
> yla(, ang(h)) ytitle("weight, lb")
> by(treatment, row(1) note("") noixtick)
```

produces the spitting image of figure 3. `c(L)` here is an old Stata trick: exactly the same syntax has carried over from the old graphics before Stata 8. `c(L)` says "join the data if and only if the x axis variable is increasing". The previous `sort id time` means that `time` goes 1 2 1 2 1 2 and so forth for identifiers 1 2 3 and so forth, and so joining of points takes place only within panels, as 1 increases to 2, and not between panels, as 2 decreases back to 1. Thus that one option ensures that data points are connected only for each individual girl.

How do we separate weight losers and gainers with this data structure? Consider the command

```
. by id: gen byte falling = weight[2] < weight[1]
```

What happens with this command depends delicately on the previous `sort id time`, so for clarity let us combine the two commands:

```
. bysort id (time): gen byte falling = weight[2] < weight[1]
```

The trick here is very Stataish. Some readers may smile, recognizing an old friend. The tutorial earlier mentioned (Cox 2002) rehearses the basics. The key point is that under `by:`, subscripts such as `[1]` and `[2]` are interpreted within the groups defined by `by:`. Here that means groups defined by distinct values of `id`, the panels or individual girls. The panels for this dataset are balanced with precisely two observations each, so everything is about as simple as it could be.

With this sort order, the first observation in each panel, subscripted `[1]`, is the first in `time`, and the second, `[2]`, is the second in `time`. Therefore, for both observations in each panel, the new variable created with the value for that panel of

```
weight[2] < weight[1]
```

which is true, evaluated as 1, if `weight` fell from time 1 to time 2 and false, evaluated as 0, otherwise.

Let us stay with this syntax briefly and note some further implications. Suppose first that the data were organized in panels of three or more observations. Then the expression just above would still be evaluated using the values for the first two observations in each panel and that would be done also for observations other than the first two. There is no rule that expressions evaluated must refer to any data in the current observation. If the reference was just to `weight`, then Stata would always use the value of `weight` in each observation, but explicit subscripts override that kind of reference.

Suppose now that there was only one observation in a panel. Then the reference to `weight[2]` remains legal but Stata would not find a corresponding value and would return missing, in this problem, numeric missing (`.`). The expression `weight[2] < weight[1]` would then return false unless the value of `weight[1]` was itself missing. In this example, no great harm would ensue, but in other problems you might be bitten, so watch out.

```
. bysort id (time): gen byte falling = weight[2] < weight[1] if _N == 2
```

would trap this particular problem.

Back to the graphics: With our new indicator variable, `falling`, we have the means to separate weight losers and gainers.

One technique is to superimpose graphs for the subset `if falling` and the subset `if !falling`, as used before for figure 4. Another technique is to use `separate` first, which we will show as a variation. Cox (2005c) gave another example of the use of `separate` for scatterplot. The result is a replica of figure 4.

```
. separate weight, by(falling)
. line weight? time, c(L L) lp(solid ..) lc(gs12 gs2)
> xla(1 "   before" 2 "after   ") xtitle("") yla(, ang(h)) ytitle("weight, lb")
> by(treatment, row(1) note("") legend(off) noixtick)
```

`weight?` here is a wildcard that catches the variables `weight0` and `weight1` produced by `separate`. To replicate figure 4, we need to spell out that the line patterns for the two variables are the same; alternatively, we could have different line patterns if we so wished.

However, replicating figures 5 and 6 would be easier with a restructuring back to the data structure we started with, so in one sense, we now close a loop with this example.

4 Big rivers

4.1 The example data

We turn now to a very different example in which two measured variables are on quite different scales, so that we choose to compare them using ranks, and in which we also have names that we wish to see.

Allen (1997, 136–139) gave data on 97 of the world's largest rivers. We focus on basin (catchment or watershed) area and mean discharge, the mean volume of water per unit time leaving the river basin. As an aside, note that complete databases for even the largest rivers remain elusive and that definitions and measurements of these and other quantities are highly problematic. See, for example, the often different data listings of Gleick (1993) or Shiklomanov and Rodda (2003).

The data are provided with the media for this issue.

```
. use rivers, clear
```

4.2 Paired plots with names

The initial stimulus for writing this particular column was seeing some attractive displays produced by Fry (2008), broadly similar to what we are going to see. In essence, they are modern equivalents of Guerry's displays (Guerry 1833).

For our illustration, we select the 25 largest rivers according to basin area.

```
. gsort -area
. keep in 1/25
(72 observations deleted)
```

```
. list name area discharge
```

	name	area	discha~e
1.	Amazon	6150	200000
2.	Zaire (Congo)	3700	40900
3.	Mississippi	3344	18400
4.	Nile	2715	317
5.	Parana	2600	18000
6.	Yenisei	2580	17800
7.	Ob	2500	12200
8.	Lena	2430	16200
9.	Yangtze	1940	28500
10.	Amur	1855	10300
11.	Mackenzie	1448	9830
12.	Zambezi	1400	6980
13.	Volga	1350	8400
14.	St Lawrence	1185	14300
15.	Niger	1112.7	6020
16.	Shatt al Arab	1050	1460
17.	Ganges	980	11600
18.	Yellow (Huang He)	980	1550
19.	Indus	960	7610
20.	Orinoco	945	34900
21.	Murray	910	698
22.	Chari	880	1320
23.	Yukon	855	6180
24.	Danube	815	6660
25.	Mekong	810	14900

The area of the Niger basin for some reason is recorded with an extra decimal place. We will round for display purposes.

```
. replace area = round(area)
```

For the vertical coordinates, we need ranks on area and discharge. Given our previous sort, the first is immediately accessible as the observation number. For the second, we use `egen, rank()`. Note the use of a negative sign to ensure consistent ranking so that the largest is first.

```
. gen rank1 = _n
. egen rank2 = rank(-discharge)
```

It so happens that there are no ties. `egen, rank()` has a `unique` option for that situation.

The horizontal coordinates we can set with small integers, as before:

```
. gen byte one = 1
. gen byte two = 2
```

I am going to give the rest of the code all at once and then comment once you have seen the display. This code is, not surprisingly, a cleaned-up version after various small experiments. My experience is simply that you should get a rough version up and running and then improve it step by step.

```
. gen left = 0.4
. twoway pcspike rank1 one rank2 two,
> xla(none) xsc(noline r(0.3 2.3)) xtitle("")
> ysc(r(-1 .) reverse off) yla(, nogrid)
> || scatter rank1 one, mla(area) mlabpos(9) ms(none)
> || scatter rank2 two, mla(discharge) mlabpos(3) ms(none)
> || scatter rank1 left, mla(name) mlabpos(3) ms(none)
> text(-0.5 1 "area, 000 sq.km") text(-0.5 2 "discharge, cu.m/s")
> legend(off) graphregion(color(white))
```

Figure 7 shows the result.

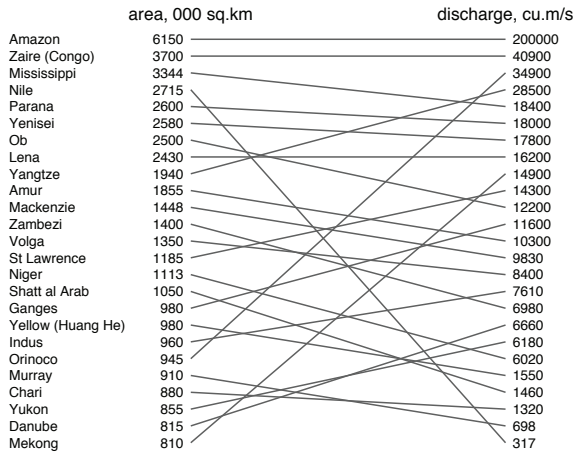

Figure 7. Paired display ranking 25 big rivers by both area and mean discharge

Comments now:

1. The spikes are drawn by `twoway pcspike`, as is now familiar.

2. The values of `area`, as just rounded slightly, are displayed to the left using a `scatter` with vertical coordinate the rank, horizontal coordinate 1, invisible marker symbol, and the values as marker labels. `mlabpos(9)` puts the labels at the 9 PM position, i.e., to the left of where the marker symbols would have been visible.

3. Similarly, the values of `discharge` are displayed to the right using a `scatter` with vertical coordinate the rank, horizontal coordinate 2, invisible marker symbol, and

the values as marker labels. `mlabpos(3)` puts the labels at the 3 PM position, i.e., to the right of where the marker symbols would have been visible.

4. The river names are shown using the same device as in 2 and 3. The horizontal position of 0.4 is the result of experiment. Clearly, longer or shorter names, or different judgments about spacing, would mean different values.

5. Column headers are added through `text()` options.

6. Given the extra material on the margins of the graph, we need to stretch axis limits using `xscale()` and `yscale()`.

7. We want rank 1 at the top and rank 25 at the bottom, so spell out `ysc(reverse)`.

8. The rest of the code consists of subtracting stuff that is unnecessary or would be a distraction:

```
xla(none) xsc(noline) xtitle("") ysc(off) yla(, nogrid) legend(off) ///
   graphregion(color(white))
```

   Bear in mind that what is shown just above is specific to the `sj` scheme. If other graph schemes had been used, the code might have been slightly different.

4.3 Other possibilities

The main point of the example is to show something of what is possible. Other possibilities now open up in turn. For example, we could highlight particular observations or groups of observations. Someone might want to emphasize that the Nile is anomalous in having high area but low discharge, a result of arid climate in its lower parts and much extraction of water for human uses, including irrigation. Figure 8 shows how this might appear. The particular change here of thickening one spike was made in the Graph Editor without needing to work out the command-based logic.

Some might want to tweak the presentation by using different justifications (left, right, centered) of the numeric values. Others might want to subdivide rising and falling groups, those with higher ranks on one variable than another. Fry (2008) gives some good examples of this style.

Now that ranks have been mentioned, we should spell out what will be intuitive: the connection between these graphs and rank correlation. At one extreme, perfect coincidence of ranks corresponds to both perfect positive rank correlation and a graph with no crossings of lines. For details on the relationship between line crossings and rank correlation, start with Fisher (1983). Except in extreme cases, anyone wanting to see a rank correlation will still find it faster to use the corresponding Stata command.

People who work with panel or longitudinal data have wrestled with the problem that graphs with many panels superimposed can become too busy. For many researchers, the datasets looked at here are just toy examples: their sample sizes are hundreds, thousands, or millions. The common affectionate reference to spaghetti plots understates the problem, because graphs can be much more entangled even than spaghetti ever gets.

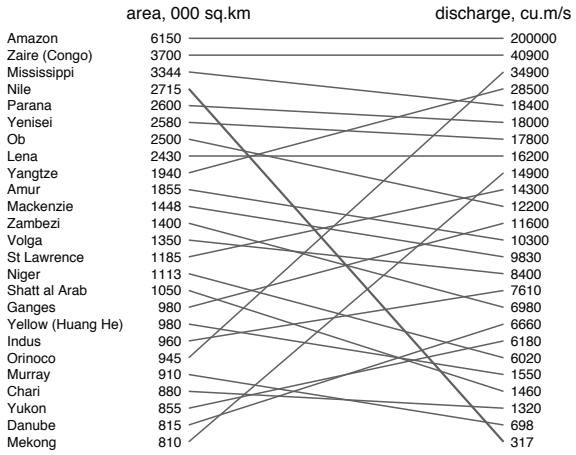

Figure 8. Paired display ranking 25 big rivers by both area and mean discharge. Note how the Nile has very low discharge considering its area.

One standard idea is to plot the mass of curves in subdued form or as unconnected data points and plot some kind of regression or smooth summary on top (e.g., Singer and Willett [2003]). This can solve the problem of not being able to see the wood for the trees, as the old English cliché has it. However, how individual trees change is also of central concern.

Diggle et al. (2002, 38–39) suggest an interesting compromise. Choose the individuals that on some characteristic (say, their median residual from the overall mean or smooth curve) lie at certain selected quantiles throughout the distribution. For concreteness, these might be the minimum, the 5th, 10th, 25th, 50th, 75th, 90th, and 95th percentiles, and the maximum. Thus if these individual curves were plotted, something of the range and style of variability would remain evident in the plot and not be submerged in a gray mess of data points. Although this idea is not explored here, its application in Stata would rest on the devices exemplified in this column.

5 Conclusion

In graphics as in the rest of statistical science, the very simplest ideas are often the best. Plotting data for two variables in parallel is a beautifully simple idea made easy to implement by Stata's paired-coordinate graphs. Only experiment indicates what works best with particular datasets, but it is important to remember that other possibilities lie beyond the scatter or time-series plot.

6 Acknowledgment

Rino Bellocco drew my attention to Campbell and Kenny (1999).

7 References

Allen, P. A. 1997. *Earth Surface Processes*. Oxford: Blackwell Science.

Basford, K. E., and J. W. Tukey. 1997. Graphical profiles as an aid to understanding plant breeding experiments. *Journal of Statistical Planning and Inference* 57: 93–107.

———. 1999. *Graphical Analysis of Multiresponse Data: Illustrated with a Plant Breeding Trial*. Boca Raton, FL: Chapman & Hall/CRC.

Campbell, D. T., and D. A. Kenny. 1999. *A Primer on Regression Artifacts*. New York: Guilford Press.

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

Cook, D., and D. F. Swayne. 2007. *Interactive and Dynamic Graphics for Data Analysis: With R and GGobi*. New York: Springer.

Cox, D. R. 1958. *Planning of Experiments*. New York: Wiley.

Cox, N. J. 1990. Hillslope profiles. In *Geomorphological Techniques*, ed. A. S. Goudie, 92–96. London: Unwin Hyman.

———. 2002. Speaking Stata: How to move step by: step. *Stata Journal* 2: 86–102.

———. 2004. Speaking Stata: Graphing agreement and disagreement. *Stata Journal* 4: 329–349.

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

———. 2005b. Stata tip 21: The arrows of outrageous fortune. *Stata Journal* 5: 282–284.

———. 2005c. Stata tip 27: Classifying data points on scatter plots. *Stata Journal* 5: 604–606.

———. 2006. Assessing agreement of measurements and predictions in geomorphology. *Geomorphology* 76: 332–346.

———. 2009. Stata tip 78: Going gray gracefully: Highlighting subsets and downplaying substrates. *Stata Journal* 9: 499–503.

Diggle, P. J., P. Heagerty, K.-Y. Liang, and S. L. Zeger. 2002. *Analysis of Longitudinal Data*. 2nd ed. Oxford: Oxford University Press.

du Toit, S. H. C., A. G. W. Steyn, and R. H. Stumpf. 1986. *Graphical Exploratory Data Analysis*. New York: Springer.

Few, S. 2009. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Oakland, CA: Analytics Press.

Fisher, N. I. 1983. Graphical methods in nonparametric statistics: a review and annotated bibliography. *International Statistical Review* 51: 25–38.

Friendly, M. 2007. A.-M. Guerry's *Moral Statistics of France*: Challenges for multivariable spatial analysis. *Statistical Science* 22: 368–399.

———. 2008. The golden age of statistical graphics. *Statistical Science* 23: 502–535.

Fry, B. 2008. *Visualizing Data: Exploring and Explaining Data with the Processing Environment*. Sebastopol, CA: O'Reilly.

Gleick, P. H., ed. 1993. *Water in Crisis: A Guide to the World's Fresh Water Resources*. New York: Oxford University Press.

Guerry, A.-M. 1833. *Essai sur la Statistique Morale de la France*. Paris: Crochard.

Hand, D. J., F. Daly, A. D. Lunn, K. J. McConway, and E. Ostrowski, ed. 1994. *A Handbook of Small Data Sets*. London: Chapman & Hall.

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

Inselberg, A. 2009. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. New York: Springer.

McNeil, D. 1992. On graphing paired data. *American Statistician* 46: 307–311.

———. 1996. *Epidemiological Research Methods*. Chichester, UK: Wiley.

Ramsey, F. L., and D. W. Schafer. 2002. *The Statistical Sleuth: A Course in Methods of Data Analysis*. 2nd ed. Pacific Grove, CA: Duxbury.

Robbins, N. M. 2005. *Creating More Effective Graphs*. Hoboken, NJ: Wiley.

Shiklomanov, I. A., and J. C. Rodda, ed. 2003. *World Water Resources at the Beginning of the 21st Century*. Cambridge: Cambridge University Press.

Singer, J. D., and J. B. Willett. 2003. *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford: Oxford University Press.

Theus, M., and S. Urbanek. 2009. *Interactive Graphics for Data Analysis: Principles and Examples*. Boca Raton, FL: Chapman & Hall/CRC.

Tufte, E. R. 1990. *Envisioning Information*. Cheshire, CT: Graphics Press.

———. 2006. *Beautiful Evidence*. Cheshire, CT: Graphics Press.

Unwin, A., M. Theus, and H. Hofmann. 2006. *Graphics of Large Datasets: Visualizing a Million*. New York: Springer.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. New York: Springer.

Wallgren, A., B. Wallgren, R. Persson, U. Jorner, and J.-A. Haaland. 1996. *Graphing Statistics and Data: Creating Better Charts*. Newbury Park, CA: Sage.

Wilkinson, L. 2005. *The Grammar of Graphics*. 2nd ed. New York: Springer.

Young, F. W., P. M. Valero-Mora, and M. Friendly. 2006. *Visual Statistics: Seeing Data with Dynamic Interactive Graphics*. Hoboken, NJ: Wiley.

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*.