The Stata Journal

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

The Stata Journal (2005)
5, Number 4, pp. 574–593

Speaking Stata: Smoothing in various directions

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

Abstract. Identifying patterns in bivariate data on a scatterplot remains a basic statistical problem, with special flavor when both variables are on the same footing. Ideas of double, diagonal, and polar smoothing inspired by Cleveland and McGill's 1984 paper in the *Journal of the American Statistical Association* are revisited with various examples from environmental datasets. Double smoothing means smoothing both y given x and x given y. Diagonal smoothing means smoothing based on the sum and difference of y and x that treats the two variables symmetrically, possibly under standardization. Polar smoothing is based on the transformation from Cartesian to polar coordinates followed by smoothing and then reverse transformation; here the smoothing is implemented by regression on a series of sine and cosine terms. These methods thus offer exploratory tools for determining the broad structure of bivariate data.

Keywords: gr0021, exploratory data analysis, statistical graphics, bivariate data, double smoothing, doublesm, diagonal smoothing, diagsm, polar smoothing, polarsm

1 Introduction

In my last two columns (Cox 2005a,b), I discussed some personal favorite methods that in some sense or another have failed to become totally standard. This theme continues with an examination of various ideas for smoothing data on scatterplots that were prominent in a major paper in a leading journal several years ago (Cleveland and McGill 1984) but which have not been widely adopted.

Smoothing has moved to center stage in statistical science over the last few decades. Not so long ago, smoothing meant, mostly, moving averages applied to time series. Moving averages are often useful, and they are tied to deep and beautiful mathematics in the frequency domain, but even in time-series analysis they are hardly the main idea. Yet in various ways, the search for smooth structures has become the focus of a large part of data analysis. This shift, as is well known, reflects new theory, new methods, and better computing facilities. It also reflects greater emphasis on identifying structures that both grow out of and can contribute to scientific explanations.

The interest in smoothing has grown hand in hand with a wider realization that regression, interpreted suitably broadly, is the most powerful single idea in statistics. Regardless of whether you agree with that, the theme of smoothing is broad enough to stretch from simple exploratory data analysis to deeper and more formally analyzed models. Much interest in smoothing begins with a relatively simple but perpetually interesting case, bivariate data on a scatterplot. It may seem surprising that anything can be said about that case that has not been said many times before, but this column surveys just one set of neglected ideas by way of example: double smoothing, diagonal smoothing, and polar smoothing. They are not competing methods, and they may even be complementary.

A few broad comments may be added at the outset on the homespun philosophy underlying this work. Scatterplots containing just data points do most of the work in conveying the information wrapped up in the two variables being examined. At most, we are adding enhancements that highlight some of the structure in the data—or highlight a lack of such structure. The question is often how far such enhancements support, or even contradict, any interpretation based mostly on "eyeballing" the data. In practice, such interpretations are also influenced by the scientific knowledge and the personal biases of the observer and are thus subjective to some extent. How far the data show what we think they show is a permanent issue.

2 Double and diagonal smoothing

Suppose that we have two variables that have the same units or are on equivalent unit-free scales (for example, they are both probabilities). This restriction can be relaxed, as will become clear. In addition, it is supposed that both are measured with error.

Examples are

1. Two sets of measurements of some quantity using different methods. Comparison of methods is a common problem throughout science and a much discussed subject in statistics (e.g., Dunn 2004).
2. Two sets of measurements at nearby locations. Concretely, we might have precipitation or temperature data from nearby meteorological stations.
3. Observed responses and predictions from some model. This model need not have a statistical flavor.
4. Corresponding quantiles for two groups of a variable or two variables, as might be shown on a quantile–quantile plot.

In these examples, and in many others, equality of variables is a natural reference so that equality is either expected or at least a benchmark.

If we show data for these variables on a scatterplot, which variable goes on the y-axis and which on the x-axis is likely to be arbitrary. The examples are all variables that are essentially on the same footing. We might, however, want to add some smooth trace to the graph to help to identify any systematic pattern. The motivation here is exploratory or heuristic. We might prefer not to specify formally what error structure is associated with the variables, marginally or jointly.

This problem is therefore some distance from the classic smoothing problem in which one variable fluctuates erratically and the other variable (for example, time) is considered known. The problem of "errors in variables" is related but not identical. Evidently, neither smoothing y given x nor smoothing x given y would be entirely suitable. We could

1. Choose one of these, say, smoothing y given x. At best, if the relationship is strong, the other smooth will be very similar. This is in principle wrong, as it assumes that one variable is known exactly, although we might choose not to worry about that.

2. Choose and plot both. Although not often done, this could be interesting whatever was found. Either the traces are similar—so the choice is in fact not an issue—or they are different, which should provoke further thought. This practice I here call *double smoothing*. My inspiration is Cleveland and McGill (1984, 817–818).

3. Choose a smoothing method that treats y and x symmetrically. This is even less often done, but again an ingeniously simple idea from Cleveland and McGill (1984, 818) offers a way forward. They call their procedure *sum-difference smoothing*, but I propose the name *diagonal smoothing* to reflect the idea that smoothing is done diagonally, rather than parallel to one of the axes.

Whenever x and y are positively correlated, the procedure is to form the difference and sum

$$d = y - x \qquad\qquad s = y + x$$

and then smooth the difference as a function of sum in some way, calling the result $\widehat{d}$. The calculation is then reversed, producing

$$\widehat{y} = (s + \widehat{d}\,)/2 \qquad\qquad \widehat{x} = (s - \widehat{d}\,)/2$$

to plot with the data points y and x.

Whenever x and y are negatively correlated, the procedure is to form the difference and sum, as before, and then smooth the sum as a function of the difference in some way, calling the result $\widehat{s}$. The calculation is then reversed, producing

$$\widehat{y} = (d + \widehat{s})/2 \qquad\qquad \widehat{x} = (\widehat{s} - d)/2$$

to plot with the data points y and x. (The case of negative correlation is not discussed by Cleveland and McGill but is in practice much less common in problems in which this technique is most appealing.)

This method does not depend on $y - x$ being roughly constant, but it should work well whenever that is so. But calculating the difference and sum does depend on y and x sharing the same units. However, it is easy to ensure comparability by prior use of some method of standardization, followed by smoothing on standardized scales and a reversal of the standardization.

2.1 Double smoothing in Stata

Double smoothing in Stata is straightforward. In practice, a wrapper program to do it is helpful. One is published with this column as `doublesm`. Nevertheless, let us consider how to attack the problem. Suppose that `lowess` is the smoothing command of choice and our variables are y and x. The first step is to smooth y on x and keep the smoothed values:

```
lowess y x, nograph gen(ys)
```

The second step is to smooth x on y and keep the smoothed values:

```
lowess x y, nograph gen(xs)
```

Then we put together the results in a graph. The main point requiring care is the sort order for each graph component:

```
twoway line ys x, sort || ///
       line y xs, sort(y xs) || ///
       scatter y x
```

Naturally, this schematic outline omits details of how the `lowess` calls and, in turn, the graph call might be tuned using various options. But a wrinkle does deserve mention: with `lowess` as implemented in Stata, the same smooth is not guaranteed for the same (x, y) pairs. Normally this is not a great problem. If a graphical display makes nonuniqueness evident, it is just drawing attention to a genuine characteristic of the data, the occurrence of ties. However, it is worth noting that this may happen.

With `doublesm`, this strategy is implemented via a single call `doublesm` y x; smoothing using `lowess` is the default. Optionally, the running line method as implemented by Sasieni (1995); Sasieni and Royston (1998); and Sasieni, Royston, and Cox (2005) may be specified. Earnest programmers wishing to add their favorite smoothing method would find it easy to adapt `doublesm`. In general, any method of smoothing that copes with unequal spacing of values is likely to be adequate for exploration. The usual comment from both theory and experience is that the degree of smoothing is more important than the particular flavor implemented.

2.2 Diagonal smoothing in Stata

As before, the main question in a Stata implementation of diagonal smoothing is precisely how to smooth. A program `diagsm` published with this column uses [R] **lowess** by default and running line smoothing optionally.

`diagsm` by default standardizes each variable using (value − median) / MAD before smoothing. Here MAD is the median absolute deviation from the median. This should be a reasonable general choice, given the possibility of variables with skewed and/or heavy-tailed distributions. Note, however, that MAD may be 0 whenever half or more of the values are equal to the median, in which circumstance the method will not work. A real example is the repair record variable `rep78` in Stata's `auto` dataset for domestic cars.

A less robust standardization available through the `meansd` option is (value − mean)/ (standard deviation). In each case, standardization is reversed to produce smoothed values. Only by specification of the `raw` option is smoothing on the original data scales available. This may seems backwards design, given the motivation of the method, but the intent is to protect naive users from the effects of what may well be an inappropriate default.

3 Polar smoothing

One idea that several people have tried to transfer from the univariate case to the bivariate is that of the middle or central part of the data. The ideas of the median as a center and of the quartiles as defining the central half of a distribution, from an order statistic point of view, were very well established by the latter nineteenth century, most prominently in the work of Francis Galton. Naturally, the middle or central part—whether precisely or roughly one half of the data—defines as a complement the outer part, which may well be practically or scientifically even more important or interesting.

There are various possibilities for extending these ideas to two variables and no apparent reason why several should not be fruitful. One is based on the idea of convex hulls (e.g., Barnett 1976 and various papers in Barnett 1981). Others are based closely on the box plot, generalized in some fashion (e.g., Goldberg and Iglewicz 1992 and Rousseeuw, Ruts, and Tukey 1999). Here we focus on what Cleveland and McGill (1984, 819–820) called polar smoothing. They cite a 1977 personal communication from Alan M. Gross.

Given data on x and y, the main idea is to map to polar coordinates, smooth the radius with respect to its angle, and then map back again. The smoothing thus defines a closed curve including the central part of the data. Including half of the values is not guaranteed but is often approximately satisfied. Generalizing mildly from the formulation of Cleveland and McGill, one algorithm runs like this:

1. Given measures of location $m(x)$ and $m(y)$ and measures of scale (width) $w(x)$ and $w(y)$, form $x' = \{x - m(x)\}/w(x),\ y' = \{y - m(y)\}/w(y)$.
2. Convert these to normalized coordinates $d = y' - x',\ s = y' + x'$.
3. Normalize in turn to $d' = d/w(d),\ s' = s/w(s)$.
4. Convert to polar coordinates $r = \sqrt{s'^2 + d'^2},\ \theta = \arctan(d/s)$.
5. Transform r to $z = r^{2/3}$ to provide some robustness, thus pulling in the larger spikes particularly. As r is a square root, this produces a cube root: compare

the Wilson–Hilferty transform that arises particularly in treatments of gamma distributions.

6. Smooth z as a function of θ, producing $\widehat{z}$.
7. Reverse 5 by $\widehat{r} = \widehat{z}^{3/2}$.
8. Reverse 4 by $\widehat{d'} = \widehat{r} \sin\theta$, $\widehat{s}' = \widehat{r}\cos\theta$.
9. Reverse 3 to get $\widehat{d} = w(d)\ \widehat{d'}$, $\widehat{s} = w(s)\ \widehat{s}'$.
10. Reverse 2 to get $\widehat{x}' = (\widehat{s} - \widehat{d})/2$, $\widehat{y}' = (\widehat{s} + \widehat{d})/2$.
11. Reverse 1 to get $\widehat{x} = m(x) + w(x)\ \widehat{x}'$, $\widehat{y} = m(y) + w(y)\ \widehat{y}'$.

What is tunable here?

Measures of location and scale. Cleveland and McGill use median for location and MAD, namely, median absolute deviation from the median, as measure of scale. Other possibilities are mean for location and standard deviation for scale. Yet another set of possibilities is not to normalize or equivalently to set $w(x) = w(y) = 1$.

Robustness step. The cube root transform could be varied or even omitted. But without some "robustifying" step, the representation of the central part would inevitably be sensitive to outliers, which is arguably contrary to the main purpose here.

Smoother. The most interesting issue is how to smooth the radii of polar coordinates. It is crucial to ensure that the smoother wraps around the circle without jumps or kinks and so respects the fact of working with respect to polar coordinates. Cleveland and McGill copy data fore and aft of the range of 360° or 2π radians and then apply lowess, and many other smoothers could be applied in the same way.

3.1 Polar smoothing in Stata

A program `polarsm` is published with this column. I have found it easiest to smooth by means of a regression on pairs of sine and cosine terms. Thus we `regress` z on $\sin j\theta$, $\cos j\theta$ for $j = 1$ to whatever k is desired. As sine and cosine functions automatically wrap around the circle, no special trickery of copying fore and aft is needed.[1] If $k = 1$, then some $b_0 + b_1 \sin\theta + b_2 \cos\theta$ is fitted to the transformed polar coordinates. This expression has a single maximum and a single minimum half a circle away from that. The effect when back-transformed to Cartesian coordinates thus resembles fitting an ellipse that represents the central part of the data. More pairs of terms (a larger k) allow more complication, and the resulting curve can then display concave as well as convex parts. The choice of k can, and arguably should, be informal; use whatever

[1] So, the slogan is not trickery, but trigonometry! Not fudge, but Fourier!

seems to work best at revealing structure in the data. If you have too high a value of *k*, the curve tends towards an interpolation of the data, yielding a slightly spectacular but utterly useless starburst pattern. The only common-sense stipulation is to smooth groups to be compared to similar degrees.

Mentioning groups leads to one of the most interesting ways to apply polar smoothing, either to separate *y* variables or to separate groups defined by some third variable not shown on the scatterplot. `polarsm` allows either of these choices. In the latter case, groups are indicated by an `over()` option. Either way, the question then is how far the polar smooths differ or are similar.

By default, polar smooths are shown superimposed, but you may also wish to show them in separate panels. This can be done by specifying `by()`, as usual. If separate *y* variables are being shown, know that `polarsm` temporarily restructures the dataset using `stack`. The option `by(_stack)` is thus the way to produce separate panels. Any preexisting `_stack` is temporarily `drop`ped from the data. If separate groups are indicated by `over(`*groupvar*`)`, using an additional `by(`*groupvar*`)` is the way.

Although the polar smooth is, by its definition as a sum of sines and cosines, a very smooth function, it is shown by `polarsm` as a connected line. Strictly, it is shown as two connected lines, one connecting the first to the last smoothed value, and the other closing the loop by connecting the last to the first smoothed value, but they share a linestyle. The reasons for using connected lines are both negative and positive. Negatively, `twoway mspline` can show only single-valued curves, not loops. Positively, reminding you that the smooth is an interpolation between distinct values, one for each data point, might even be a small feature.

4 The doublesm, diagsm, and polarsm programs

4.1 Syntax for doublesm

`doublesm` *yvar xvar* [*if*] [*in*] [`,`
[`lowess(`*lowess_options*`)` | `running`[`(`*running_options*`)`]] `line(`*line_options*`)`
`xsmooth(`*line_options*`)` `ysmooth(`*line_options*`)` `data(`*scatter_options*`)`
[`addplot(`*plot*`)` | `plot(`*plot*`)`] *graph_options*]

4.2 Syntax for diagsm

`diagsm` *yvar xvar* [*if*] [*in*] [`,` [`raw` | `meansd`]
[`lowess(`*lowess_options*`)` | `running`[`(`*running_options*`)`]] `line(`*line_options*`)`
`data(`*scatter_options*`)` [`addplot(`*plot*`)` | `plot(`*plot*`)`] `generate(`*newyvar*
newxvar`)` *graph_options*]

4.3 Syntax for polarsm

`polarsm` *varlist* [*if*] [*in*] [, `terms(`*#*`)` `over(`*groupvar*`)`
[`colors(`*colorstyle*`)` | `colours(`*colorstyle*`)`] `smooth(`*line_options*`)`
`data(`*scatter_options*`)` `addplot(`*plot*`)` *graph_options*]

4.4 Options for doublesm

`lowess(`*lowess_options*`)` specifies options for `lowess`; see [R] **lowess**.

`running`[`(`*running_options*`)`] specifies the use of the command `running` as an alternative to `lowess`. The option `running` by itself specifies using the `running` command and its defaults. Alternatively, `running()` specified with arguments passes options to the `running` command.

`line(`*line_options*`)` specifies options for `line` (see [G] **graph twoway line**) that are to be applied to both smooth traces.

`xsmooth(`*line_options*`)` specifies options for `line` (see [G] **graph twoway line**) that are to be applied to the smooth of *xvar* given *yvar*.

`ysmooth(`*line_options*`)` specifies options for `line` (see [G] **graph twoway line**) that are to be applied to the smooth of *yvar* given *xvar*.

`data(`*scatter_options*`)` specifies options for `scatter` (see [G] **graph twoway scatter**) that are to be applied to rendering of the data points.

`addplot(`*plot*`)` (Stata 9 and later; see [G] ***addplot_option***) or `plot(`*plot*`)` (Stata 8 only; see [G] ***plot_option***) provides a way to add other plots to the generated graph.

graph_options are other options documented in [G] ***twoway_options***.

4.5 Options for diagsm

`raw` specifies no scaling or standardization so that the sum and difference are calculated directly from the data before smoothing.

`meansd` specifies scaling by (value − mean) / standard deviation before smoothing. The standardization will be reversed after smoothing.

`lowess(`*lowess_options*`)` specifies options for `lowess`; see [R] **lowess**.

`running`[`(`*running_options*`)`] specifies the use of the command `running` as an alternative to `lowess`. The option `running` by itself specifies using the `running` command and its defaults. Alternatively, `running()` specified with arguments passes options to the `running` command.

`line(`*line_options*`)` specifies options for `line` (see [G] **graph twoway line**) that are to be applied to the smooth trace.

`data(`*scatter_options*`)` specifies options for `scatter` (see [G] **graph twoway scatter**) that are to be applied to the data points.

`addplot(`*plot*`)` (Stata 9 and later; see [G] ***addplot_option***) or `plot(`*plot*`)` (Stata 8 only; see [G] ***plot_option***) provides a way to add other plots to the generated graph.

`generate(`*newyvar newxvar*`)` specifies the names of two new variables to hold the y and x coordinates of the diagonal smooth.

graph_options are other options documented in [G] ***twoway_options***.

4.6 Options for polarsm

`terms(`*#*`)` specifies the number k of Fourier terms, $\sin j\theta$, $\cos j\theta$, $j = 1, \ldots, k$ on which the radius is regressed as a function of the polar angle θ. By default, k is 1.

`over(`*groupvar*`)` specifies a variable to be used to identify different groups. `over()` is not allowed whenever three or more variables are specified.

`colors(`*colorstyle*`)` or `colours(`*colorstyle*`)` (choose just one) specifies a list of colors with which to show different polar smooths. See [G] ***colorstyle***.

`smooth(`*line_options*`)` specifies options for `line` (see [G] **graph twoway line**) controlling the representation of different polar smooths.

`data(`*scatter_options*`)` specifies options for `scatter` (see [G] **graph twoway scatter**) controlling the representation of data points for different variables or subsets of data.

`addplot(`*plot*`)` provides a way to add other plots to the generated graph; see [G] ***addplot_option***.

graph_options are other options documented in [G] ***twoway_options***. In particular, note that by default, polar smooths are shown superimposed. You may want to show them in separate panels by specifying `by()`. If separate y variables are being shown, `polarsm` temporarily restructures the dataset using `stack`; see [D] **stack**. The option `by(_stack)` is thus the way to produce separate panels. (Any preexisting `_stack` is temporarily dropped from the data.) If separate groups are specified by `over(`*groupvar*`)`, an additional `by(`*groupvar*`)` is the way to go.

5 Examples

Two substantial examples using environmental datasets will now be examined. Polar smoothing also features briefly in the engaging text by Helsel and Hirsch (1992).

5.1 Rainfall in the southern Pennines

A first example features the rainfall captured at three rain gauges over 42 years (1906–1947) in part of the Derwent catchment (drainage basin or watershed) in the southern Pennines in Britain. Rainfall means, strictly, precipitation: any fall of snow or other solid forms of water is included as rainfall equivalent. The original data as reported by Law (1953) were given in inches, which are long since obsolete as measurement units, except among historians and U.S. citizens. However, those units are retained here: for your information, 1 inch is exactly 25.4 mm.

As rainfall is sensitive to small differences in geographic location, especially but not only landsurface altitude, many rain gauges are needed to get a good idea of the total or average rainfall in an area. Conversely, rainfalls at nearby places do tend to be similar. A common hydrological practice, as rain gauges are expensive, is thus to pepper an area initially with many gauges, and then find out which pairs of gauges correlate so highly that one can be removed without much loss of information. Standard climatological folklore runs (e.g., Linacre 1992, Dingman 2002) that ratios of rainfalls at nearby rain gauges tend to be nearly constant, but naturally this needs checking. Even if it holds, the constant needs to be determined from data.

What should influence analysis is that the amounts at different gauges are variables on the same footing. It would be at best arbitrary, and at worst wrong, to regard the rainfall *here* as a response to the rainfall *there*, or indeed vice versa. It would be nearer the mark to regard both as related to some latent or hidden variable, but here we proceed otherwise, merely treating pairs of variables symmetrically.

Let us start with the idea of double smoothing, the idea of smoothing x given y, as well as that of smoothing y given x. Figures 1, 2, and 3 show the results of double smoothing pairs of three gauges, numbered 17, 23, and 60. More generally, double smoothing pairs of k variables would yield $k(k-1)/2$ plots, taking interchange of axes for granted, just as with a half of a scatterplot matrix. The Stata recipe for each of these graphs resembles this command:

```
. doublesm srain23 srain17, yla(, nogrid ang(h)) ms(oh)
> legend(order(1 "23 | 17" 2 "17 | 23") ring(0) pos(5) col(1))
> addplot(function x, range(srain23) lstyle(grid))
```

As the different gauges are highly correlated, it is possible to tuck the legend into the bottom right-hand corner, although taste or convention might lead you to another choice. A reference line $y = x$ is also added. This could be done in various ways, but putting a `twoway function` call into an `addplot()` option and requesting a `grid` linestyle is one. (In Stata 8, use a `plot()` option instead.)

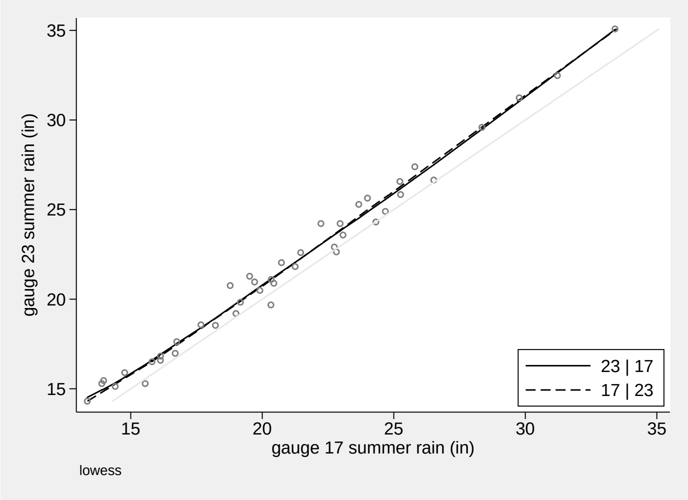

Figure 1: Double smoothing by lowess of summer rainfall at two rain gauges. The smooths of gauge 23 given 17 and of gauge 17 given 23 are almost identical and shifted slightly from the reference line $y = x$.

The graphs show part of the range of behavior that might be expected. When two variables agree quite closely, as with gauges 17 and 23 in figure 1, the two smooths are essentially the same, and both are close to the reference line of equality. The structure in this example seems more additive than multiplicative, although there is a hint of multiplicativity at higher rainfalls.

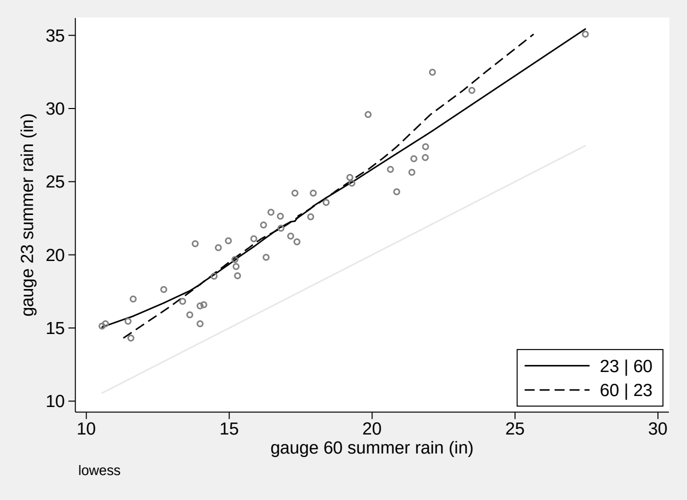

Figure 2: Double smoothing by lowess of summer rainfall at two rain gauges. The smooths of gauge 23 given 60 and of gauge 60 given 23 are similar. Both indicate a mainly multiplicative shift.

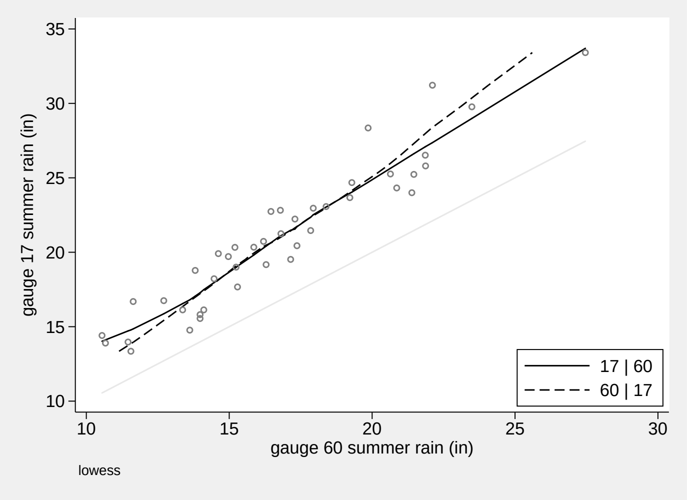

Figure 3: Double smoothing by lowess of summer rainfall at two rain gauges. The smooths of gauge 17 given 60 and of gauge 60 given 17 are similar. Again both indicate a mainly multiplicative shift.

When pairs of variables do not agree so closely, the smooths inevitably differ to some extent (figures 2 and 3). What is most common is a scissors pattern of intersecting smooths, familiar from any discussion of the two regression lines of y given x and of

x given y. In either case, however, the smooths in these two figures both support the notion of multiplicative structure. This could be explored in other ways, such as plotting the difference versus the mean or the ratio versus the geometric mean, an approach discussed, with several references, in a previous column (Cox 2004c).

When the double smooths agree, many would be happy to accept a compromise between them, and this is where diagonal smoothing is most congenial. Figure 4 gives one example. A `generate()` option is used to store the coordinates of the smooth for later use. As earlier mentioned, the default of `diagsm` is to smooth after a robust standardization, which is then reversed.

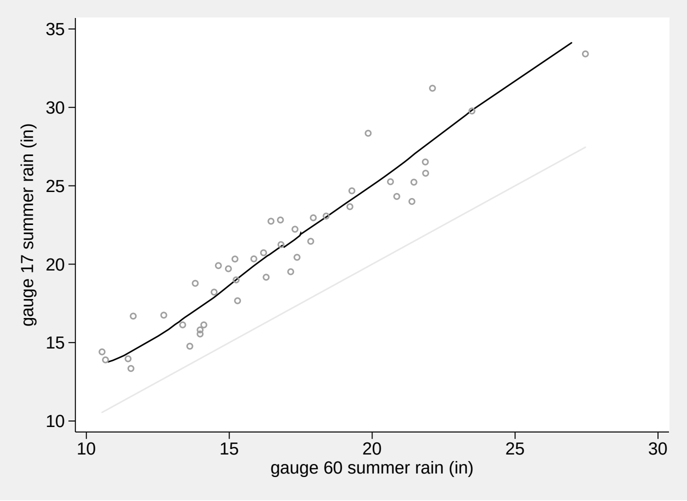

Figure 4: Diagonal smoothing by lowess of summer rainfall at two rain gauges, 17 and 60.

```
. diagsm srain17 srain60, yla(, nogrid ang(h)) ms(oh)
> legend(ring(0) pos(5) col(1)) addplot(function x, range(srain60) lstyle(grid))
> generate(y_1760 x_1760)
```

The new variables can then be added to a composite graph (figure 5).

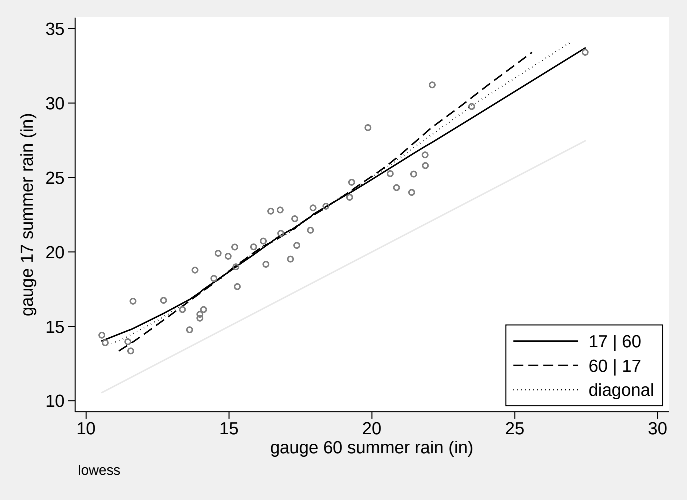

Figure 5: Double and diagonal smoothing by lowess of summer rainfall at two rain gauges, 17 and 60. The diagonal smooth provides a reasonable compromise between the double smooths.

```
. doublesm srain17 srain60, yla(, nogrid ang(h)) ms(oh)
> addplot(function x, range(srain60) lstyle(grid) ||
> line y_1760 x_1760, lp(dot) sort)
> legend(order(1 "17 | 60" 2 "60 | 17" 5 "diagonal") ring(0) pos(5) col(1))
```

The `legend(order())` call shows the benefit of insider information. Know that the fifth variable being plotted here is the diagonal smooth: the third is the data, as a set of point symbols, and the fourth is the line of equality.

This graph is, as you will appreciate, a well-behaved example, and experience will differ with worse behavior. Whenever candidate smooths agree, it matters little, either scientifically or even statistically, which one we choose, but at least in this case we considered different possibilities for analysis. Even in worse cases, differences between candidate smooths will emphasize uncertainty about how best to summarize the structure, and thus caution against overhasty interpretation.

5.2 Cirques in the English Lake District

We continue with a geomorphological example, returning to a dataset used previously in these columns (Cox 2004a, 2005a) on 158 glacial cirques from the English Lake District (Evans and Cox 1995). Glacial cirques are hollows excavated by glaciers that are open downstream, bounded upstream by the crest of a steep slope, and arcuate in plan around a more gently sloping floor. More informally, they are sometimes described as ‘armchair-shaped’. A similar dataset from Wales is used elsewhere in this issue (Cox 2005c).

Two key variables are the length and width of these cirques. Previous analyses indicate on various grounds that it is best to think on logarithmic scales, so we smooth on those scales. However, we will want graphs to be labeled in terms of the original variable labels and units, which make much more scientific sense. This is a common-enough desire in other problems for us to detail one way how to do that.

First, set up default titles and axis labels in local macros. The program `mylabels` is downloadable from SSC and was exemplified in an earlier column (Cox 2004b).

```
. local yttl "Width, max. perpendicular to median axis (m)"
. local xttl "Length of median axis, focus to crest (m)"
. mylabels 200 500 1000 2000, myscale(log10(@)) local(la)
```

Width and length are, once again, variables on the same footing. Geometrically, both are aspects of plan size and shape. Physically, neither is a forcing variable. So diagonal and double smoothing appeal here. Figure 6 was produced similarly to figure 5 by first diagonal smoothing, using its `generate()` option to save the coordinates of the smooth, and then plotting those on top of a double smooth. What is especially intriguing and a new detail is the hint of curvature at high values. Power-function models, which transform to straight lines when both variables are logged, have long been popular with variables like these, but the graph hints that here the power function may not fit quite as well as thought. Other analyses are needed to follow this up, but the smoothing has produced a new idea.

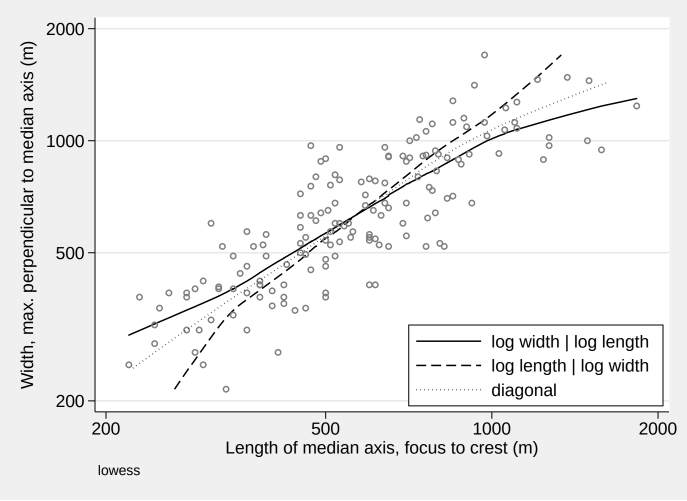

Figure 6: Double and diagonal smoothing by lowess of log cirque width and length. Smoothing was carried out on logarithmic scales. There is a hint of curvature at high values that conflicts slightly with the idea of a power function fit, which would imply a straight line fit on this graph.

Now we turn to examples of polar smoothing. In practice, start with the default in which one pair of terms (one sine and one cosine) is used to smooth polar coordinates, thus yielding an approximate ellipse (figure 7). Then add further terms until it seems that they are merely corresponding to noise or uninteresting detail (figures 8 and 9).

```
. polarsm logw logl, yla(‘la’, ang(h)) xla(‘la’) yti("‘yttl’") xti("‘xttl’") ms(oh)
. polarsm logw logl, yla(‘la’, ang(h)) xla(‘la’) yti("‘yttl’") xti("‘xttl’") ms(oh)
> terms(2)
. polarsm logw logl, yla(‘la’, ang(h)) xla(‘la’) yti("‘yttl’") xti("‘xttl’") ms(oh)
> terms(4)
```

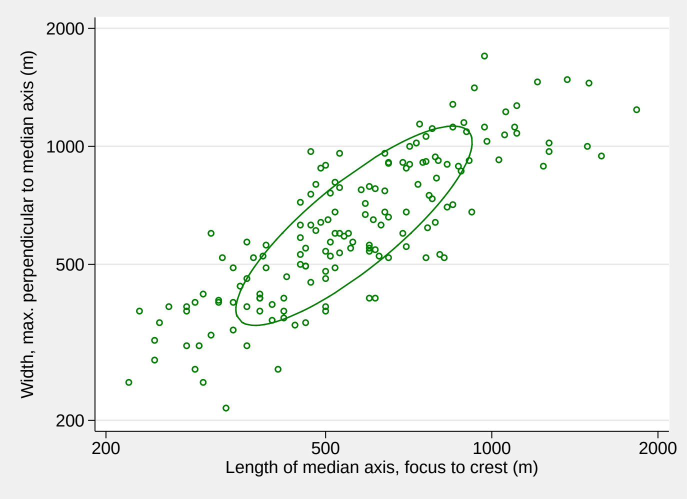

Figure 7: Polar smooth of log cirque width and length. Smoothing was carried out on logarithmic data. The smoothing is based on regressing radial coordinates on one sine term and one cosine term, producing an approximate ellipse on back-transformation.

(*Continued on next page*)

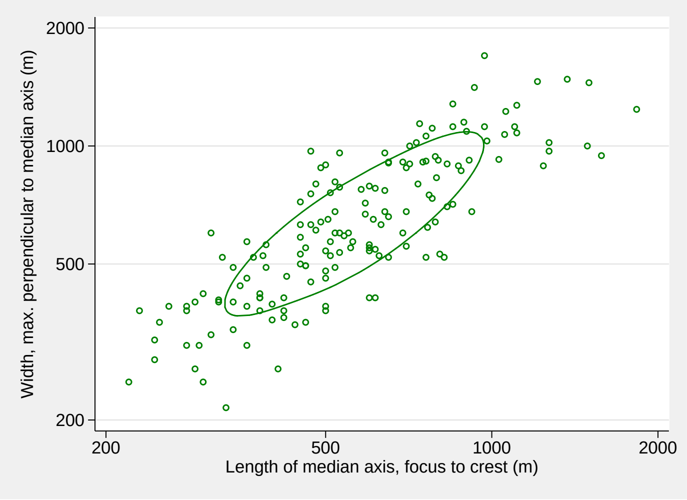

Figure 8: Polar smooth of log cirque width and length. Smoothing was carried out on logarithmic data. The smoothing is based on regressing radial coordinates on two sine terms and two cosine terms, but the result resembles figure 7.

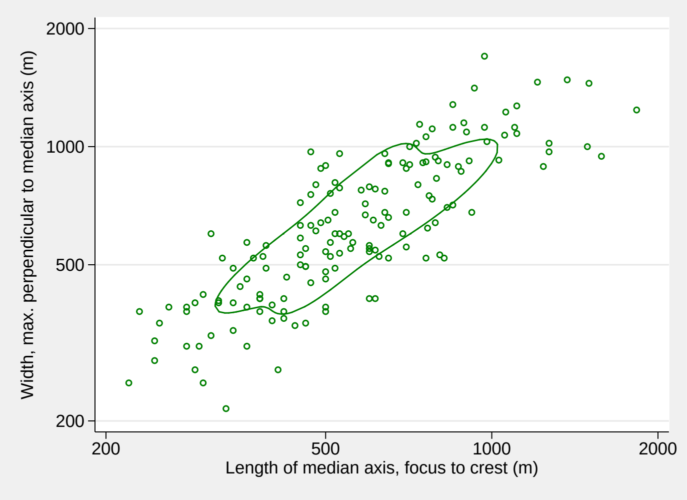

Figure 9: Polar smooth of log cirque width and length. Smoothing was carried out on logarithmic data. The smoothing is based on regressing radial coordinates on four sine terms and four cosine terms and shows some sensitivity to small details of the configuration of points.

As earlier mentioned, polar smoothing done groupwise is a way of checking whether groups behave similarly or differently. An ordered categorical variable in this dataset is the ‘grade’ of the cirque, codifying the expert’s best summary of how well developed the cirque is, meaning how far it represents evidence of strong modification by ice. The numbers in each grade are classic 11, well-defined 31, definite 44, marginal 37, and poor 35. The number of classic cirques is too small for this technique, so the first two categories are combined:

```
. gen grade2 = cond(grade <= 2, 1, grade)
. label def grade2 1 "classic, well-defined" 3 "definite" 4 "poor" 5 "marginal"
. label val grade2 grade2
. polarsm logw logl, yla(‘la’, ang(h)) xla(‘la’) yti("‘yttl’") xti("‘xttl’")
> ms(oh) terms(2) over(grade2) by(grade2, row(1) legend(off) note("") compact)
> subtitle(, ring(0) pos(11) nobexpand box fcolor(none)) xla(, ang(v))
```

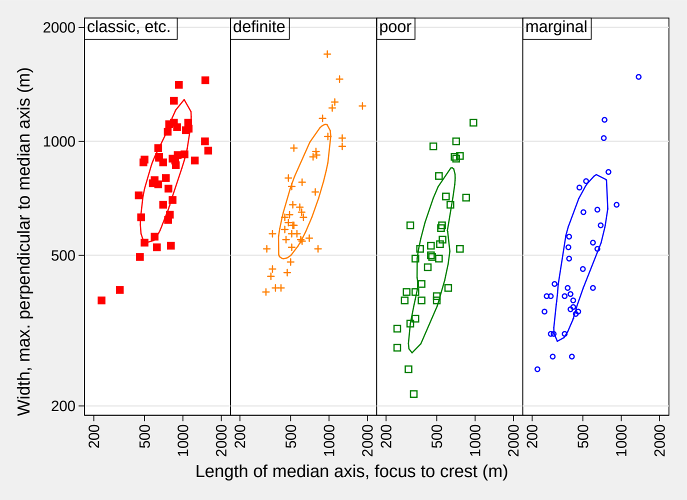

Figure 10: Polar smooth of log cirque width and length, separately by expert-determined grade. Smoothing was carried out on logarithmic data. A systematic shift over categories is evident, although results indicate that poor and marginal cirques are similar.

The polar smooths in figure 10 show evidence of a systematic change in characteristic lengths and widths but also some novel details, including the approximate similarity of the last two categories.

6 A note on quantile–quantile plots

An application of diagonal smoothing particularly that is not discussed in detail here is to quantile–quantile plots. These plots, for example, as implemented in Stata’s `qqplot`, are often plainly presented, usually with no more than a reference line. Yet adding a smooth can be useful. For this, it is helpful to have a utility generating corresponding quantiles for two variables or for two groups of one variable. `cquantile` is built for this purpose and may be downloaded using `ssc` ([D] **ssc**). A quantile–quantile plot with smoothing can then be produced with `diagsm`.

7 Conclusions

Cleveland and McGill (1984) presented an abundance of simple but valuable ideas in their wide-ranging paper, perhaps too many for them all to be absorbed. Their 'sunflowers' have seeded and borne fruit recently in Stata (see [R] **sunflower**; Dupont and Plummer, Jr., 2005). This paper has drawn attention to an array of more neglected ideas for identifying basic structure on scatterplots, together with the release of new programs for experimentation, `doublesm`, `diagsm`, and `polarsm`. They are particularly relevant to the common but rather neglected problem of smoothing when both variables are on the same footing.

8 Acknowledgments

William S. Cleveland, Marcello Pagano, and Vincent Wiggins provided encouragement. Continuing joint work with Ian S. Evans provided data and motivation.

9 References

Barnett, V. 1976. The ordering of multivariate data. *Journal of the Royal Statistical Society, Series A* 139: 318–355.

Barnett, V., ed. 1981. *Interpreting Multivariate Data*. Chichester, UK: Wiley.

Cleveland, W. S. and R. McGill. 1984. The many faces of a scatterplot. *Journal of the American Statistical Association* 79: 807–822.

Cox, N. J. 2004a. Speaking Stata: Graphing distributions. *Stata Journal* 4(1): 66–88.

———. 2004b. Speaking Stata: Graphing categorical and compositional data. *Stata Journal* 4(2): 190–215.

———. 2004c. Speaking Stata: Graphing agreement and disagreement. *Stata Journal* 4(3): 329–349.

———. 2005a. Speaking Stata: Density probability plots. *Stata Journal* 5(2): 259–273.

———. 2005b. Speaking Stata: The protean quantile plot. *Stata Journal* 5(3): 442–460.

———. 2005c. Stata tip 27: Classifying data points on scatter plots. *Stata Journal* 5(4): 604–606.

Dingman, S. L. 2002. *Physical Hydrology*. Upper Saddle River, NJ: Prentice Hall.

Dunn, G. 2004. *Statistical Evaluation of Measurement Errors: Design and Analysis of Reliability Studies*. London: Arnold.

Dupont, W. D. and W. D. Plummer, Jr. 2005. Using sunflower plots to explore bivariate relationships in dense data. *Stata Journal* 5(3): 371–384.

Evans, I. S. and N. J. Cox. 1995. The form of glacial cirques in the English Lake District, Cumbria. *Zeitschrift für Geomorphologie* 39: 175–202.

Goldberg, K. and B. Iglewicz. 1992. Bivariate extensions of the boxplot. *Technometrics* 34(3): 307–320.

Helsel, D. R. and R. M. Hirsch. 1992. *Statistical Methods in Water Resources*. Amsterdam: Elsevier. Free download from *http://pubs.usgs.gov/twri/twri4a3/*.

Law, F. 1953. The estimation of the reliable yield of a catchment by correlation of rainfall and run-off. *Journal of the Institution of Water Engineers* 7: 273–293.

Linacre, E. T. 1992. *Climate Data and Resources: A Reference and Guide*. London: Routledge.

Rousseeuw, P., I. Ruts, and J. W. Tukey. 1999. The bagplot, a bivariate boxplot. *American Statistician* 53: 382–387.

Sasieni, P. 1995. sed9: Symmetric nearest neighbor linear smoothers. *Stata Technical Bulletin* 24: 10–14. In *Stata Technical Bulletin Reprints*, vol. 4, 97–101. College Station, TX: Stata Press.

Sasieni, P. and P. Royston. 1998. sed9.1: Pointwise confidence intervals for running. *Stata Technical Bulletin* 41: 17–23. In *Stata Technical Bulletin Reprints*, vol. 7, 156–163. College Station, TX: Stata Press.

Sasieni, P., P. Royston, and N. J. Cox. 2005. sed9_2: Symmetric nearest neighbor linear smoothers. *Stata Journal* 5(2): 285.

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 co-authored fifteen commands in official Stata. He was an author of several inserts in the *Stata Technical Bulletin* and is an Editor of the *Stata Journal*.