The Stata Journal

The Stata Journal (2006)
6, Number 3, pp. 397–419

Speaking Stata: Graphs for all seasons

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

Abstract. Time series showing seasonality—marked variation with time of year—are of interest to many scientists, including climatologists, other environmental scientists, epidemiologists, and economists. The usual graphs plotting response variables against time, or even time of year, are not always the most effective at showing the fine structure of seasonality. I survey various modifications of the usual graphs and other kinds of graphs with a range of examples. Although I introduce here two new Stata commands, `cycleplot` and `sliceplot`, I emphasize exploiting standard functions, data management commands, and graph options to get the graphs desired.

Keywords: gr0025, cycleplot, sliceplot, seasonality, time series, graphics, cycle plot, rotation, state space, incidence plots, folding, repeating

1 Seasonality

Seasonality—marked variation with time of year—must have been evident to the first humans. Indeed many organisms show awareness of, or adaptations to, seasonality. It remains a matter of great interest to many scientists.

Astronomers explain seasonality in terms of the motion of the earth relative to the sun. That story is part of one of the great successes of modern science, which we owe largely to Copernicus, Kepler, and Newton. Viewed astronomically, seasonality—for example, prediction of times of sunrise or sunset—is a classic deterministic problem, but for all other sciences it has a strongly stochastic or statistical flavor. Climatologists look at variations in temperature, rainfall, and other elements around the year, but everyone knows that no two summers are identical. Seasonality of climate has many other environmental effects. Many are fairly direct, such as those on water supply or vegetation condition, but some are more subtle and even controversial, such as alleged seasonality in the incidence of earthquakes or volcanic eruptions in response to variations in overburden pressure. Epidemiologists examine seasonal variations in morbidity, mortality, and natality, an approach that goes back at least as far as the Hippocratic writing *Airs, Waters, Places* in the fifth century BCE. Economists have long monitored seasonal variations in variables such as employment, sales, and GDP, although often these are regarded as nuisances requiring seasonal adjustment.

The most common graphs for seasonal data are plots of one or more response variables versus time or time of year. This statement is surely well known, so why then this column? Negatively, such plots are often not especially effective at showing the fine structure of seasonality. Positively, their effectiveness can be improved by various tricks, and other kinds of plots can be useful too: indeed, we can borrow ideas on seasonal graphics from various fields.

I will introduce two user-written commands, `cycleplot` and `sliceplot`, but I will emphasize using some basic functions, graphics options, and data management commands.

This column is the second of a series with the general theme of circular arguments. The first column examined time of day as a circular scale (Cox 2006).

2 Related problems

Although the focus here is on seasonality, the main ideas carry over to other periodicities, such as time of day or time of week. I will not spell out that connection further, as translating code to other periodicities will typically be straightforward. Similarly, just flagging a standard point should be enough: seasonality is usually combined with variations on other time scales. The graphics to be discussed apply either to data with some seasonal variation or to a seasonally varying component of such data, calculated in some way.

Traditionally, we distinguish seasons by named divisions: in English, as winter, spring, summer, and autumn or fall. In climatology, these divisions are often made more precise as the four quarters December–February, March–May, June–August, and September–November, because surface phenomena tend to lag solar inputs enough to justify the offset of 1 month from the conventional calendar year beginning in January. In data analysis, any such divisions are usually at best conventional or convenient categories. Underlying them are periodic or circular numerical scales, such as month of year or day of year, in which the last value of any year is followed by the first value of the following year.

How far, then, should seasonal data be considered a kind of circular data? Some intriguing circular graphs have been suggested for seasonal data. For example, Tufte (2001, 72) reproduces a spiral representation of Italian postal bank deposits from 1876 to 1881. Unfortunately, reading off the structure of seasonality from such graphs is hard. I suggest that, on the whole, seasonal data are better shown using linear graphics. This conclusion follows partly because seasonal data are one kind of time series, for which a linear time axis is both customary and natural, and partly because few scientists have much experience in interpreting seasonal graphics displayed in circular formats, in contrast to their frequent familiarity with compass or map formats. Brinton (1914, 80) aired a similar view.

That said, one elementary but also fundamental idea is worth borrowing in seasonal graphics and has already been hinted at. January is an arbitrary start to the year in almost all senses but calendar convention, so rotating the seasons to start the time-of-year scale at another time may be useful. The concept is already familiar to those accustomed to thinking in financial or fiscal years.

The examples here are all for time series in the strict sense: variables counted or measured for regularly spaced times, whether intervals or points. There are also event data, times for deaths, earthquakes, riots, and so forth. Ideas for graphing the occurrence or frequency of such point process data follow readily from the ideas to be discussed here.

With its focus on graphics, this column cannot do justice to a theme that is linked but also distinct: how best to model (or smooth) time series, given the presence of seasonality. Similarly, Fourier or spectral (or frequency domain) methods also deserve more discussion. My own prejudice is that seasonality is usually obvious enough not to need discovery as a massive spike in the spectrum. Nevertheless, sometimes only spectral methods can give the full context of variability at a range of frequencies. Newton (1993) surveyed graphics for time series, discussing frequency domain displays in some detail.

3 The Bills of Mortality

Bills of Mortality were issued weekly in London from the 16th century on giving counts of deaths from various causes, collating data from the several parishes in the city. They stimulated John Graunt (1620–1674), a London draper, to write *Natural and Political Observations . . . upon the Bills of Mortality*, one of the founding documents of statistics, epidemiology, and demography. He was elected to the then-young Royal Society within weeks of the book's publication.

From the fifth (and posthumous) edition of 1676, we take data on deaths from plague in various years, noting the peaks around August and September. Figure 1 shows the annual series superimposed, and figure 2 shows them separated. Logarithmic scales seem especially appropriate for explosive phenomena such as plague.

(*Continued on next page*)

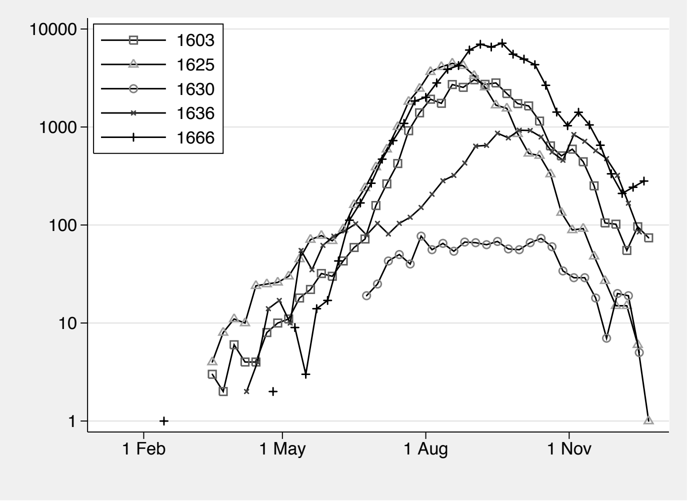

Figure 1: Plague deaths in London in various years from data reported by Graunt (1676). Note the shared tendency to peaks around August and September.

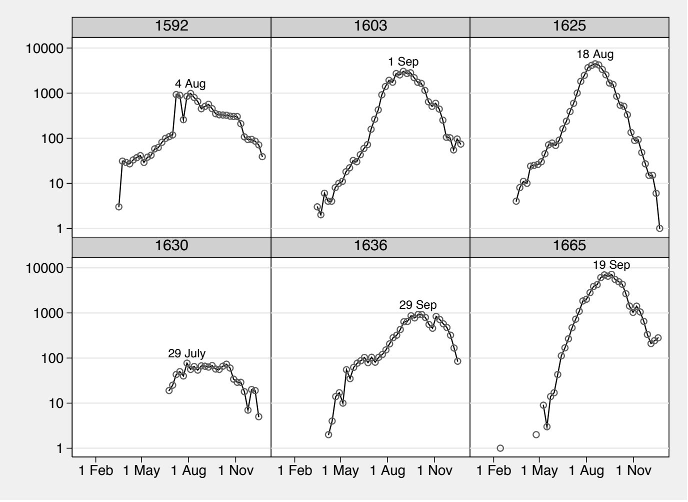

Figure 2: Plague deaths in London in various years from data reported by Graunt (1676). Added dates show weekly reports with highest numbers in each year.

In his edition of Graunt (1676), Hull gave detailed comments on the data. Implausible numerical quirks imply that the 1592 data are unreliable. Other sources indicate various small corrections and qualifications for the later years. However, none of these problems affect the main argument here.

Choosing between superimposing and juxtaposing is not always easy. Although examples clearly give complementary views of a given dataset, you may not be able to persuade reviewers or editors to include both in a publication.

4 Stata tips for plotting versus time of year

Reviewing some small but practical points for graphs of this kind may be helpful. The data may have arrived as, or been converted to, Stata date variables, but having, e.g., separate month and year variables is also helpful.

An especially useful function is `doy()` for day of year, running from 1 to 365 or 366. Note also the `egen` function `foy()` for fraction of year in the `egenmore` package on SSC (see [R] **ssc** for more on SSC).

Check out built-in sequences, such as `c(Mons)`. See the results of `creturn list`, scrolling toward the end. See also Cox (2004a).

Remember `twoway connected` as well as `line`. Although line plots are conventional in various disciplines, connected plots have the merit of showing individual data points. Marker symbol size can always be tuned to be noticeable but not obtrusive.

Use the `separate` command to separate one variable into several for easy comparison. See also Cox (2005b) for another example.

Because zeros cannot be shown as such on logarithmic scales, change zeros to missing in a copy of the data. Then prohibit connections across spells of missing values with the option `cmissing(n)`.

5 Cycle plots

5.1 Introduction

Graunt's data come for selected years. Having single or multiple time series extending over several years is more common. Figure 3 is an example from economics with monthly data. Trend, seasonality, and irregularities (attributable here mostly to strikes) are all evident. The data are for distance flown by U.K. airlines and come from Kendall and Ord (1990). Logarithmic scales again appear natural.

(*Continued on next page*)

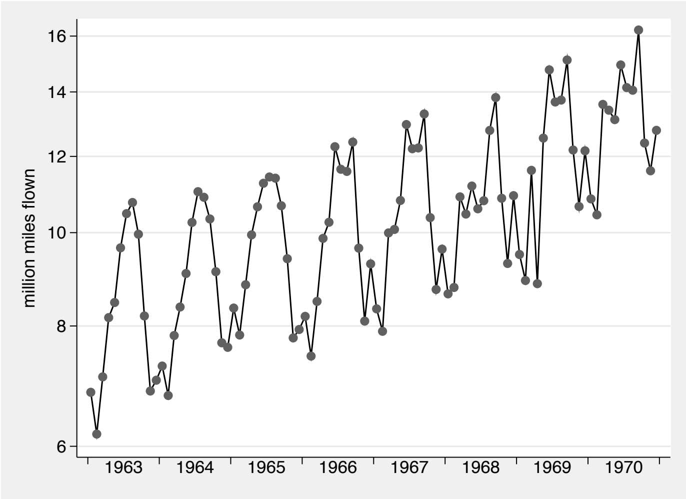

Figure 3: Distance flown by U.K. airlines—a common kind of economic time-series graph showing trend, seasonality, and irregularities.

This graph illustrates an elementary principle: the `sort` order for monthly data is naturally first by year and then by month. The idea of cycle plots is just to reverse that: sort by month, and then by year, to see the information in a different way. We could do this by using some `graph` command and an option, `by(`*monthvar*`)`, but there would be too much scaffolding. Hence I have written `cycleplot` for this purpose and formally publish it with this column.

5.2 Syntax

`cycleplot` *responsevars month year* [*if*] [*in*] [, `length(`#`)` `start(`#`)` `summary(`*egen_function*`)` `mylabels(`*labels_list*`)` *line_options*]

5.3 Options

`length(`#`)` indicates that data are for # shorter periods within each longer period. The default is 12, for months within a year.

`start(`#`)` indicates the first value of month plotted on the x-axis. The default is `start(1)`. This option may be used whenever there is some better natural start to the year than (say) January. For example, rainfall in climates with a wet season either side of December is best plotted starting in (say) July.

`summary(`*egen_function*`)` calculates a summary function to be shown for each month. The summary function may be any function acceptable to `egen` that has syntax like `egen` *newvar* `= mean(`*response*`), by(`*month*`)`. `mean()` and `median()` are the most obvious possibilities. Know that whenever summaries are plotted, the order of variables on the graph is all the response variables followed by all the corresponding summary variables.

`mylabels(`*labels_list*`)` specifies text labels to use on the time axis, instead of default labels such as 1/12. The number of labels specified should be the same as the argument of `length()`, or by default 12. Labels consisting of two or more words should be bound in `" "`. Labels including `"` should be bound in `` `" "' ``. ``mylabels(`c(Mons)')`` specifies `Jan Feb Mar` ... `Nov Dec`, and ``mylabels(`c(Months)')`` specifies `January February March` ... `November December`. Do not rotate the list to reflect a `start()` choice other than 1; this step will be done automatically.

line_options refers to options of `graph twoway line`; see [G] **graph twoway line**. `connect(L ..)` is wired in. You can use `recast()` to get a different twoway type.

5.4 Examples

Cycle plots have been discussed under other names in the literature, including cycle-subseries plot, month plot, seasonal-by-month plot, and seasonal subseries plot. For textbook treatments, see Becker, Chambers, and Wilks (1988); Cleveland (1993, 1994); or Robbins (2005). For research paper examples, see Cleveland and Devlin (1980); Cleveland and Terpenning (1982); Cleveland, Freeny, and Graedel (1983); or Cleveland et al. (1990).

Figure 4 is a default cycle plot for our example. We see the structure of seasonality much more easily, especially details such as the shift in peak from July to September.

The syntax used was

```
. cycleplot air month year,
> ylabel(6000 "6" 8000 "8" 10000 "10" 12000 "12" 14000 "14" 16000 "16", ang(h))
> ytitle(million miles flown) yscale(log)
```

(*Continued on next page*)

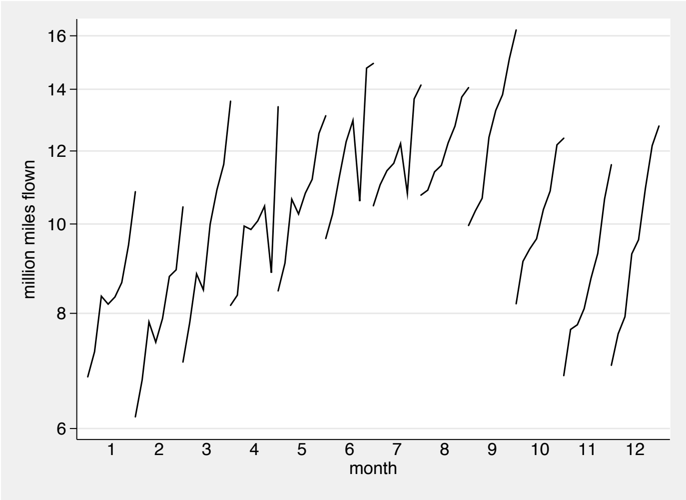

Figure 4: Distance flown by U.K. airlines. This cycle plot gives a different take on seasonality, more clearly showing timing (and shifts in timing) of peaks and troughs.

The program `cycleplot` can plot several responses and is applicable to any setup of longer periods divided into a fixed number of shorter periods. Quarterly data are thus another application. We will stick to the terms "month" and "year" as more concise, despite the imprecise terminology.

In `cycleplot`, you can rotate the time axis to start within the year. Experience indicates that splitting troughs, not peaks of the cycle, is best, although the opposite would apply if troughs were the focus of interest. Thus in studying rainfall variations, split the dry season rather than the wet season, unless the structure of the dry season is of concern.

You can also superimpose a summary for each month by naming the corresponding `egen` function, such as `mean`.

Standard graph options include `recast()`. Figure 5 shows the previous cycle plot, modified merely by adding the option `recast(connected)` and tweaking the axis labels by the option `mylabels('c(Mons)')`.

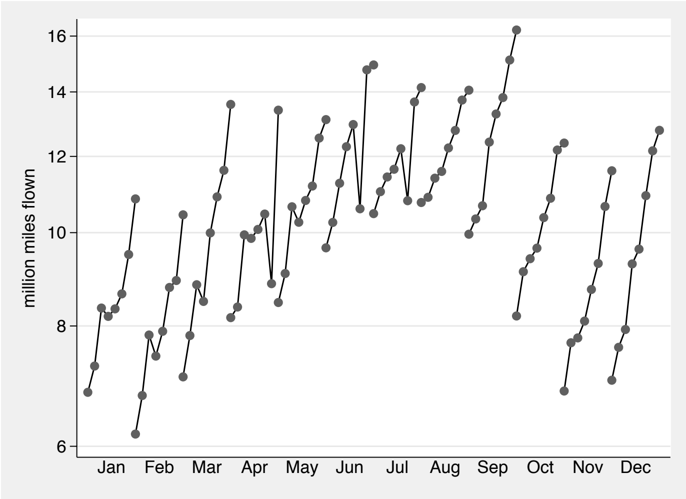

Figure 5: Distance flown by U.K. airlines. This cycle plot has been tweaked into a connected plot, and the month axis labels have been modified.

Here is another example, from medical statistics. Figure 6, using data from Diggle (1990), shows deaths in the United Kingdom from bronchitis, emphysema, and asthma. Seasonality is no surprise here, but as before a cycle plot is better than the standard time-series plot at showing the fine structure—indeed at showing basic details such as peak and trough months. A logarithmic scale makes each fluctuation up or down come out around the same height. Figure 7 shows a cycle plot, here rotated so that the winter is not cut, by using the option `start(8)`, and recast as a connected plot, by using the option `recast(connected)`.

(*Continued on next page*)

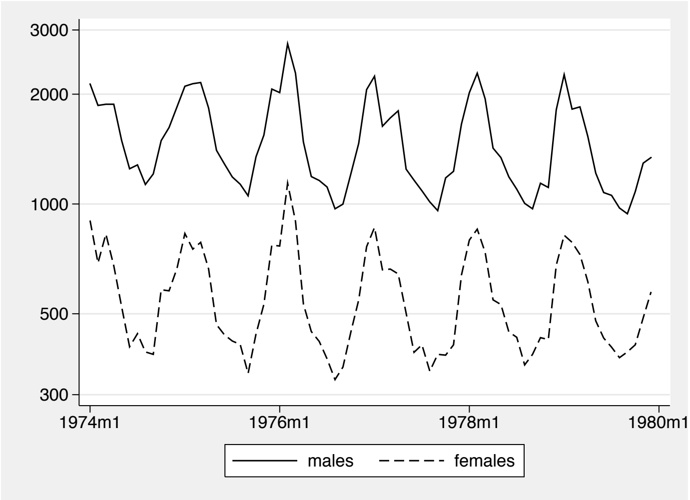

Figure 6: Deaths in the United Kingdom from bronchitis, emphysema, and asthma. Standard line plot of a strongly seasonal series.

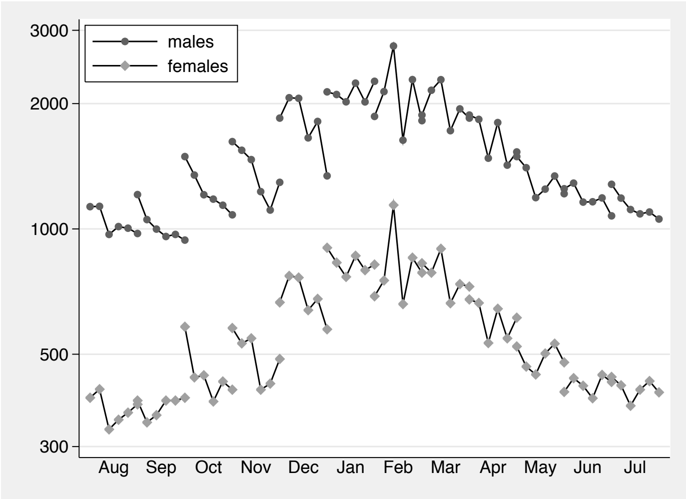

Figure 7: Deaths in the United Kingdom from bronchitis, emphysema, and asthma. This cycle plot more clearly shows the structure of seasonality.

6 Do-it-yourself rotation

`cycleplot` allows you to rotate the time-of-year axis. Few analysts will need much convincing that rotation can be a good idea. So how could you do it yourself?

Let us keep the example of monthly data and assume that a `month` variable runs from 1 (January) to 12 (December). (Separate month and year variables are useful even when you have Stata date variables.) Say that you want to start the year in month 8 (August). So months 8–12 are to be mapped to positions 1–5, and months 1–7 are to be mapped to positions 6–12.

An expression to use in generating such a new variable is

```
cond(month > 7, month - 7, month + 5)
```

as there are two cases to cover, the second part of the year that becomes the first and vice versa. See Kantor and Cox (2005) for a tutorial on `cond()`. An alternative is

```
1 + mod(month - 8,12)
```

as the remainder on dividing integers by 12 must vary from 0 to 11. I suggest that the latter method is more elegant but the former is easier to emulate.

Short of fixing axis labels, that is all that you need to know. However, you might wish to note various pertinent `egen` functions in Cox (1999, 2000) and `egenmore` from SSC.

7 Mauna Loa: Superimposing, slicing, stacking

7.1 Introduction

In 1958 the oceanographer Charles D. Keeling (1928–2005) started what is now the longest continuous series of carbon dioxide measurements on top of Mauna Loa, Hawaii. This dataset is crucial to discussions of human effects on the atmosphere. The units are ppm, parts per million (by volume). Thus 300 ppm = 0.03%.

I accessed data from http://cdiac.ornl.gov/ftp/trends/co2/maunaloa.co2 on March 22, 2006 and linearly interpolated a few small gaps in the early part of the record. Figure 8a shows a strong trend and seasonality. Given the trend, a plot against month using `connect(L)` is interesting (figure 8b). The lack of overlap here can be considered fortuitous but also fortunate. `connect(L)` connects if and only if the x-axis variable is increasing (strictly, not decreasing). `connect(l)` would be useless here, producing logical but confusing backward connections between each December (12) and the following January (1).

(*Continued on next page*)

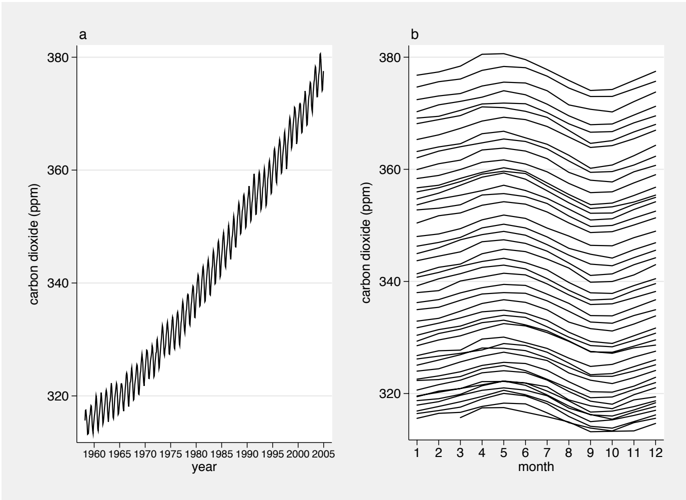

Figure 8: (a) Carbon dioxide measured at Mauna Loa shows a strong upward trend and fairly systematic seasonality. (b) Plotting against time of year gives a handle on the seasonality. By chance, no playing with offsets is needed for the annual segments.

Given such series, we should smooth or model and look at the residuals. How best to do that is a fascinating subject, and time-series experts could have a field day comparing their favorite methods, but here we just use the `lowess` default and plot the residuals from that. A superimposed line plot (figure 9a) and a standard time-series plot (figure 9b) of residuals show the family resemblance of seasonal cycles, but whether you choose spaghetti or a roller-coaster, each shows a clear pattern but also fails to suggest anything new.

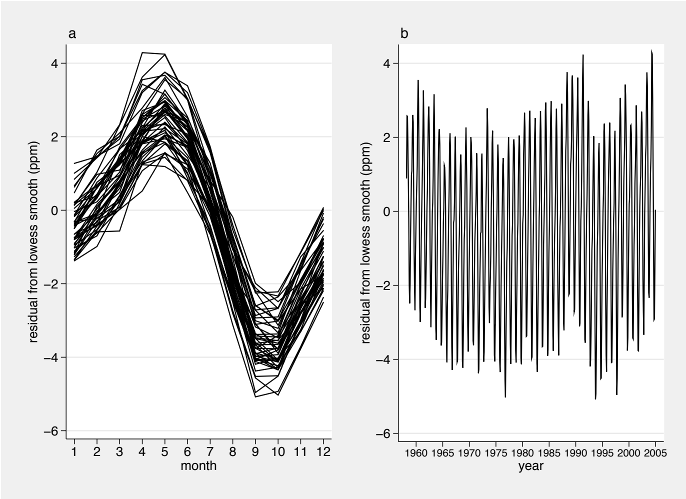

Figure 9: (a) Residuals from `lowess` default plotted against time of year. (b) Same residuals plotted as time series.

In particular, the aspect ratio of figure 9b is a problem. Standard advice (Fisher 1925; Cox 2004b) is to choose an aspect ratio such that line segments are as near 45° as possible, but here that would lead to a long graph. An alternative is to slice the series into parts, graph each part, and then stack the graphs by using `graph combine`. The details are mostly mundane but typically tedious. `sliceplot`, here published formally, is a wrapper program to automate that process.

7.2 Syntax

`sliceplot` *plottype yvarlist xvar* [*if*] [*in*] [, `at(`*numlist*`)` `unequal` `length(`#`)` `slices(`#`)` `combine(`*combine_options*`)` *twoway_options*]

7.3 Options

`at(`*numlist*`)` specifies cutpoints for the ends of each slice as values of the x-axis variable. Values outside the range of the data will be ignored with a warning.

`unequal` may be used with `at()` if you want to allow slices to have unequal scales. It specifies that unequal scales be used on slices of different length. The default is to use (approximately) the same scale. A common application is to show more interesting values at a greater magnification than others.

`length(`#`)` specifies the maximum length of each slice in units of the x-axis variable. The default is `length(100)`.

`slices(#)` specifies the number of slices.

`combine(`*combine_options*`)` specifies options of `graph combine`; see [G] **graph combine**. The defaults are `imargin(zero) cols(1)`.

twoway_options are options of `graph twoway` (see [G] **graph twoway**) controlling other features of the graph.

7.4 Examples

Figure 10 shows an example of what `sliceplot` can do.

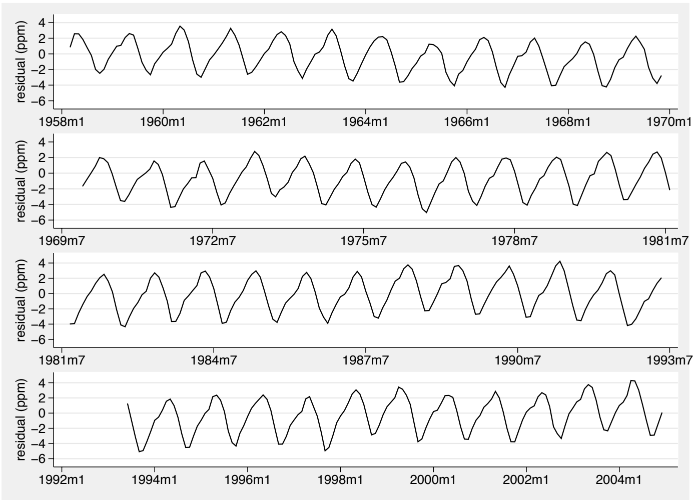

Figure 10: Residuals from `lowess` default plotted in slices to give a more congenial aspect ratio.

The command for that is

```
. sliceplot line res date, slices(4) ytitle(residual (ppm))
> ylabel(-6(2)4, angle(h)) xtitle("")
```

showing that `sliceplot` is a wrapper command that calls up a graphics command and slices the dataset by cutting the horizontal axis. You may specify both slicing options and standard `graph` options. Here we ask for just four slices, but options also exist to control slice endpoints and lengths. An analog could be written to cut the vertical axis, but I find that this aspect ratio problem occurs mostly with time series.

8 Loops in state space

One basic technique—perhaps more common in physics than in mainstream statistics—is to consider plots in some state space. Figure 11a is a basic line plot of residual versus previous residual for the Mauna Loa data. `lwidth(0)` (indeed) is a way to get thin lines. Figure 11b shows that we can identify months, which underlines the regularity of this cycle.

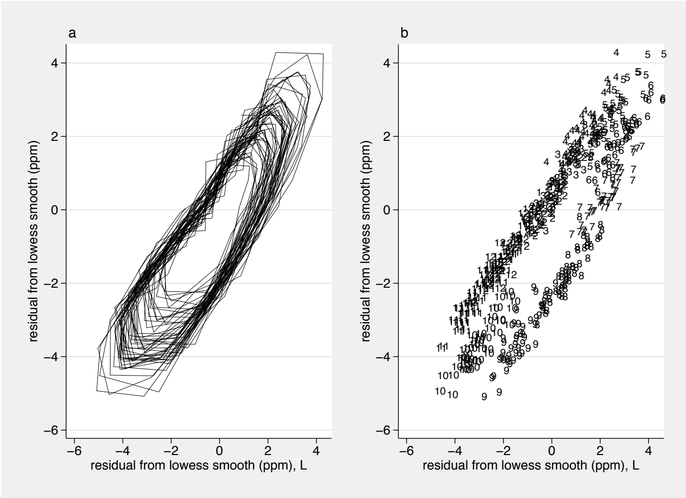

Figure 11: Residuals versus previous residuals shown using (a) a connected line and (b) month identifiers.

We can also connect with arrows by using `twoway pcarrow`. The main idea here was discussed in detail in Cox (2005a). Figure 12 gives another handle showing more of the repetitive fine structure of each seasonal cycle.

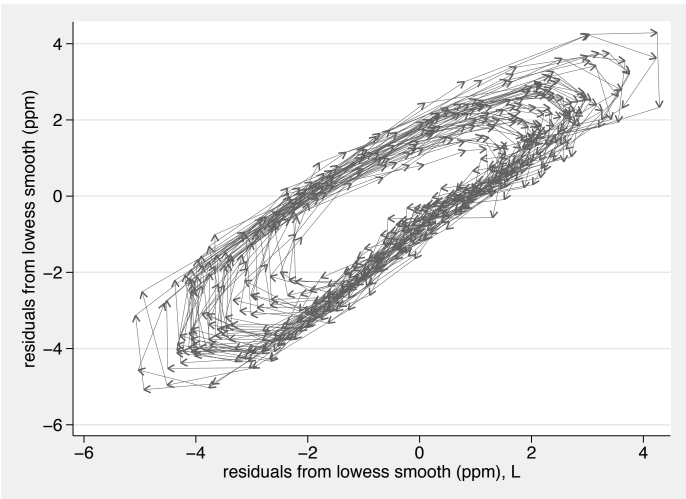

Figure 12: Residuals versus previous residuals shown using arrows.

For another application of the state space idea, let us revisit one of the staples of elementary geography, graphs of monthly means of precipitation and temperature. The usual graphs cut the year, sometimes painfully. Figures 13 and 14 give conventional graphs of the seasonal cycle for Boston, Houston, and San Francisco in the United States, using data from Pearce and Smith (1984). In the dataset, these cities are separate panels.

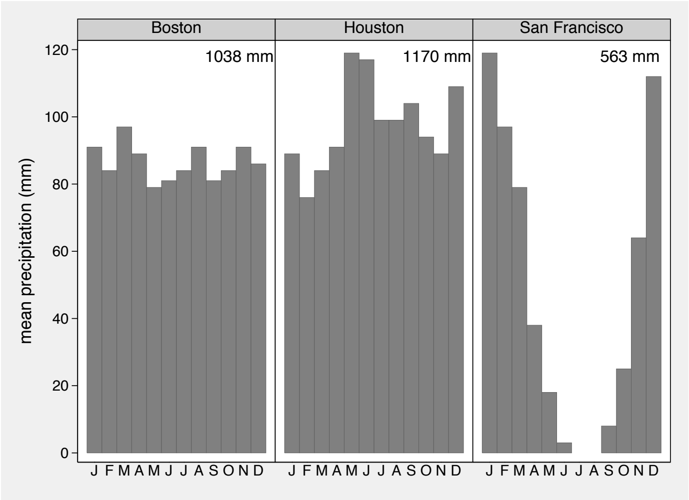

Figure 13: Annual cycle of precipitation for Boston, Houston, and San Francisco. Annual totals shown by text.

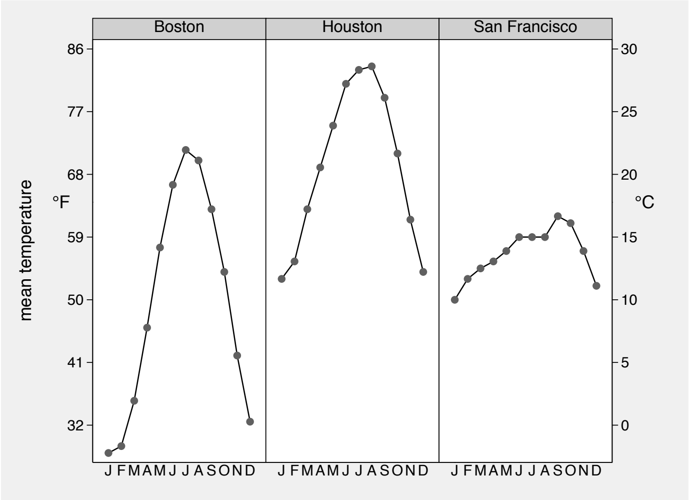

Figure 14: Annual cycle of temperature for Boston, Houston, and San Francisco.

One of various alternatives to the usual graphs is to plot the annual cycle as a loop in some two-dimensional space, say, combining precipitation and temperature. Such graphs are often called climagraphs or climographs, but there is nothing intrinsically climatic about them. It appears (Linacre 1992) that they go back to Alexandre Gustave Eiffel (1832–1923), better known for more towering achievements. For examples in a medical context, see Cliff, Haggett, and Smallman-Raynor (2004).

Figure 15a is an example in which the monthly means from January to December are connected in time order. However, December logically should also be connected to January to close the loop. Figure 15b is the result.

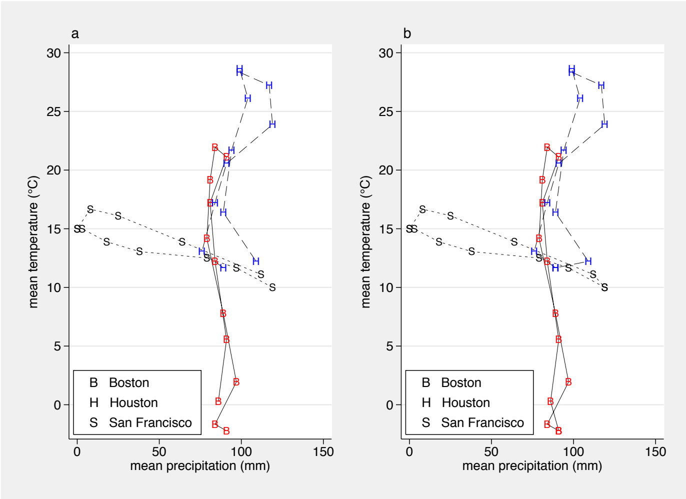

Figure 15: Annual cycle of precipitation and temperature for Boston, Houston, and San Francisco. (a) Open loop. (b) Closed loop.

How did we do that? We need to add an extra observation at the end of each panel that is a copy of the first observation. The main idea is to use `by:` and `expand`.

In more detail: The structure of the dataset is three panels and 12 months for each panel. We need to tag the first observation in each panel and then create a copy of those first observations. Knowing that `expand` adds extra observations at the end of the dataset helps. Each extra observation is assigned a value of `month` of 13, which ensures that after `sort`ing, the new observation will be in the right position.

```
. preserve
. local N = _N
. by place (month), sort: gen first = _n == 1
. expand 2 if first
. replace month = 13 if _n > `N'
. sort place month
. graph_commands
. restore
```

Here we `preserve` and then `restore` so that the original dataset is in memory after graphics. Other solutions to the problem caused by a modification of the data, which we want only for this purpose, include a `save` of the original dataset so that it can be returned to as and when desired.

9 Incidence plots

What are here called incidence plots are scatterplots of the form

`scatter year month if` *condition*

`year` and `month` are named here for concreteness. Your names naturally may differ, and your month variable may even be day of year, quarter, or some other suitable time unit. Whichever variables you choose, such an incidence plot is in essence a graphical table in which each year is a row. Logically equivalent is a scatterplot of the form

`scatter month year if` *condition*

in which each year is a column.

As we can superimpose several such plots, we can compare different years, even in a fairly long time series, with a bird's-eye view of the incidence of several different *condition*s.

The Mauna Loa data have been `tsset`, so we can use time-series operators, for example to look at changes from value to value. So after

```
. summarize D.co2, detail
```

we can show months with large positive changes (say, those in the top 10%) and months with large negative changes (say, those in the bottom 10%). The result is given in figure 16.

```
. scatter year month if D.co2 > `=r(p90)', options
> ||
> scatter year month if D.co2 < `=r(p10)', more_options
```

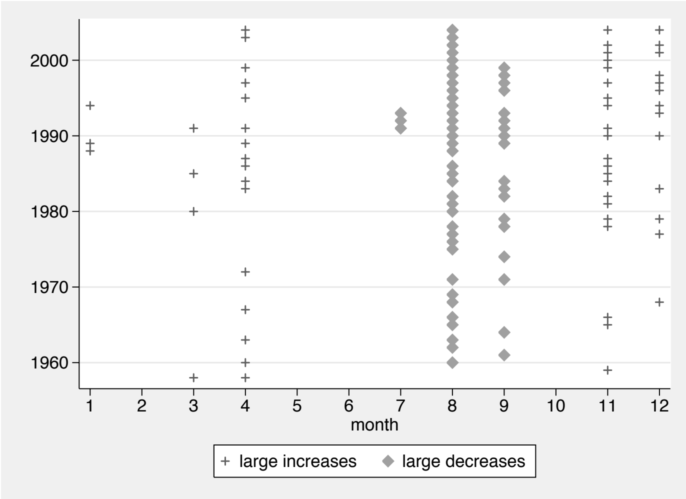

Figure 16: Incidence plot showing months of largest increases and decreases in carbon dioxide content at Mauna Loa.

Sakamoto-Momiyama (1977) makes good use of a related idea. Her disease calendars use a series of bar charts to show months of highest mortality for various diseases for different years, age groups, countries, etc. This information is within a monograph that is dense with a variety of carefully designed graphics to show seasonal variations in mortality.

10 Folding

The time-of-year axis can be folded so that the second half of the year is superimposed on the first, giving more space and a graphical handle on the asymmetry of annual cycles.

With monthly data, folding is best accomplished by the transformation `min(month, 14 - month)`, which pairs months as follows: 1 by itself, 2 and 12, 3 and 11, 4 and 10, 5 and 9, 6 and 8, and 7 by itself. Naturally a similar transformation may be used after a rotation.

Folding in this manner was used by the climatologist Victor Conrad (1876–1962). See Conrad and Pollak (1950).

11 Repeating

Values in the latter part of the year can be copied left of the start, and values in the earlier part of the year can be copied right of the end. This method reduces the effects of cutting. Mathematician and scientist Johann Heinrich Lambert (1728–1777) used repeating in this manner with seasonal data. Tufte (2001, 29) accessibly reproduces an example graph. More recently, Tukey (1972) blew a trumpet for the idea that two cycles are better than one. Two cycles are naturally not compulsory: you can copy as much or as little as desired.

The Stata code for this process is a variation on that given earlier for adding extra observations to close loops by connecting the last and first in each panel. It can be done using `expand`, often after `preserve` and before `restore`. One sequence could run like this, for two cycles:

```
. preserve
. local N = _N
. expand 2 if month <= 6
. replace month = month + 12 if _n > `N'
. local N = _N
. expand 2 if month >= 7
. replace month = month - 12 if _n > `N'
. graph_commands
. restore
```

This code gives two cycles of monthly data. First, the first 6 months are copied, and in the copies, months 1–6 are mapped to 13 to 18. Then the last 6 months are copied, and in the copies, months 7–12 are mapped to −5 to 0. The correct sort order for the graph can be obtained by an explicit `sort` or on the fly by a `sort` option of `graph`. Panel data need use of `by:`, as seen earlier.

Figure 17 reunites San Francisco's wet winter.

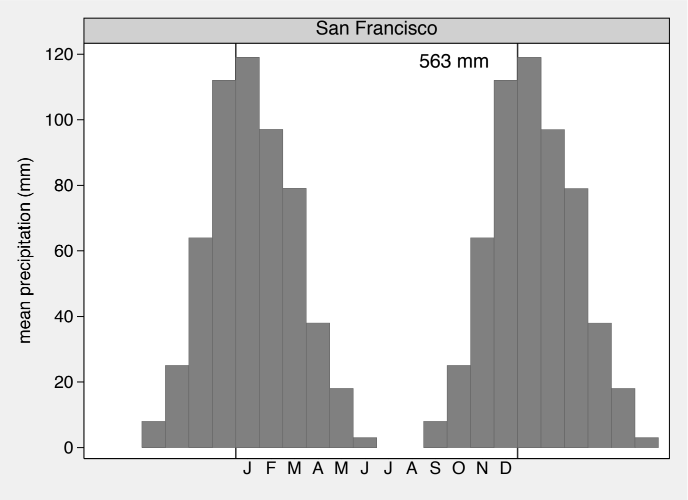

Figure 17: Annual cycle of precipitation in San Francisco. Each month is shown twice. Annual total shown by text.

12 Conclusion

For seasonal data, I give this advice on graphics.

Graphs showing the fine structure of seasonality tell us more than graphs that serve mostly to reveal its existence. The examples here are of well-understood phenomena. Can you use the method to break new ground in understanding fresh datasets?

Reordering the data into subseries (`cycleplot`) is often useful; rotate to start at an appropriate time of year for the analysis; superimpose, slice, and stack to compare years (`sliceplot`); plot loops in state space; use incidence plots; fold the time-of-year axis; and repeat values fore and aft to show up to two cycles.

Know your functions, graphics options, and data management commands. Each new program can be a curse as well as a convenience, being just one more thing to learn, remember, forget, and confuse. Once you understand the logic for rotating axes or repeating values fore and aft, the need for extra commands or extra functions to do such tasks diminishes rapidly.

13 Acknowledgments

Aurelio Tobias contributed to the development of `cycleplot`. Marcello Pagano, Austin Nichols, and Joe Newton supplied valuable references.

14 References

Becker, R. A., J. M. Chambers, and A. R. Wilks. 1988. *The New S Language: A Programming Environment for Data Analysis and Graphics*. Pacific Grove, CA: Wadsworth and Brooks/Cole.

Brinton, W. C. 1914. *Graphic Methods for Presenting Facts*. New York: Engineering Magazine Company.

Cleveland, R. B., W. S. Cleveland, J. E. McRae, and I. Terpenning. 1990. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics* 6: 3–73.

Cleveland, W. S. 1993. *Visualizing Data*. Summit, NJ: Hobart Press.

———. 1994. *The Elements of Graphing Data*. Summit, NJ: Hobart Press.

Cleveland, W. S., and S. J. Devlin. 1980. Calendar effects in monthly time series: Detection by spectrum analysis and graphical methods. *Journal of the American Statistical Association* 75: 487–496.

Cleveland, W. S., A. E. Freeny, and T. E. Graedel. 1983. The seasonal component of atmospheric CO_2: Information from new approaches to the decomposition of seasonal time series. *Journal of Geophysical Research* 88: 10934–10946.

Cleveland, W. S., and I. J. Terpenning. 1982. Graphical methods for seasonal adjustment. *Journal of the American Statistical Association* 77: 52–62.

Cliff, A. D., P. Haggett, and M. Smallman-Raynor. 2004. *World Atlas of Epidemic Diseases*. London: Arnold.

Conrad, V., and L. W. Pollak. 1950. *Methods in Climatology*. Cambridge, MA: Harvard University Press.

Cox, N. J. 1999. dm70: Extensions to generate, extended. *Stata Technical Bulletin* 50: 9–17. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 34–45. College Station, TX: Stata Press.

———. 2000. dm70.1: Extensions to generate, extended: corrections. *Stata Technical Bulletin* 57: 2. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, p. 9. College Station, TX: Stata Press.

———. 2004a. Stata tip 9: Following special sequences. *Stata Journal* 4: 223.

———. 2004b. Stata tip 12: Tuning the plot region aspect ratio. *Stata Journal* 4: 357–358.

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

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

———. 2006. Speaking Stata: Time of day. *Stata Journal* 6: 124–137.

Diggle, P. J. 1990. *Time Series: A Biostatistical Introduction*. Oxford: Oxford University Press.

Fisher, R. A. 1925. *Statistical Methods for Research Workers*. Edinburgh: Oliver & Boyd.

Graunt, J. 1676. *Natural and political observations, mentioned in a following index, and made upon the bills of mortality*. London: John Martyn. Reprinted in *The Economic Writings of Sir William Petty, Together with the Observations upon the Bills of Mortality, More Probably by Captain John Graunt*, 1899, ed. C. H. Hull. Cambridge: Cambridge University Press.

Kantor, D., and N. J. Cox. 2005. Depending on conditions: a tutorial on the `cond()` function. *Stata Journal* 5: 413–420.

Kendall, M. G., and J. K. Ord. 1990. *Time Series*. London: Arnold.

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

Newton, H. J. 1993. Graphics for time series analysis. In *Handbook of Statistics 9: Computational Statistics*, ed. C. R. Rao, 803–823. Amsterdam: North-Holland.

Pearce, E. A., and C. G. Smith. 1984. *The World Weather Guide*. London: Hutchinson.

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

Sakamoto-Momiyama, M. 1977. *Seasonality in Human Mortality: A Medico-geographical Study*. Tokyo: University of Tokyo Press.

Tufte, E. R. 2001. *The Visual Display of Quantitative Information*. 2nd ed. Cheshire, CT: Graphics Press.

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.

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 was an author of several inserts in the *Stata Technical Bulletin* and is an editor of the *Stata Journal*.