The Stata Journal (2019)
19, Number 3, pp. 741–747 DOI: 10.1177/1536867X19874264

Stata tip 132: Tiny tricks and tips on ticks

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

Vince Wiggins
StataCorp
College Station, TX
vwiggins@stata.com

1 Ticks are helpful, but how can we tune them?

Ticks on graph axes are small but ideally useful details that you may often want to tune. This column is a collection of small points on those small ticks, intended to explain what is not quite obvious—or not quite obviously useful—on reading the documentation. Naturally and necessarily, the topic has implications for axis labels too.

2 Removing ticks, or the charms of invisibility and nonexistence

Ticks classically mark steps on a continuous numerical scale, just like marks on a ruler showing lengths in centimeters and millimeters or inches and fractions of an inch. If you are as old as or even older than the authors, you drew your first graphs in school on squared paper, hoping not to make too many silly errors or smudges or splodges. You drew axes and then annotated them with ticks and labeled some if not all of the ticks. Now we have commands like `graph` to do it, but their defaults may not give you exactly what you want.

The first tip is that sometimes the ticks can be removed without pain, especially if they do not help or are misleading because the items on an axis are distinct categories, not values on a numerical scale. Thus, in comparing mileage for foreign and domestic cars using a `dotplot`, say,

```
. sysuse auto
. dotplot mpg, over(foreign) ylabel(, angle(horizontal))
```

ticks appear against "Domestic" and "Foreign" (figure 1). Tufte's term "chartjunk" (Tufte 2001) is a little strong for this case, but the ticks convey or imply nothing we do not know otherwise and so could be deleted.

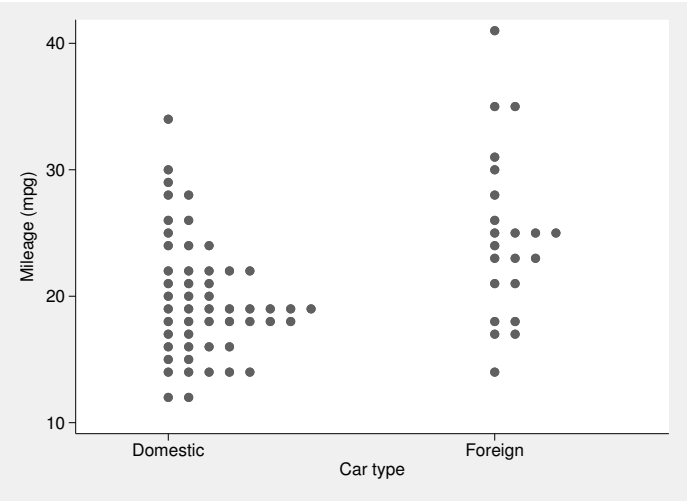

Figure 1. Dotplot of mileage by domestic and foreign cars. Do we need the ticks on the x axis?

What is going on? `dotplot` is just a wrapper for `scatter`, but `scatter` neither knows nor cares exactly what kind of numeric variable is being fed to it. Further, `dotplot` is broad-minded enough to accept numeric and string variables as arguments to `over()` without wanting to decide for you what kind of variable you have and in particular whether ticks make sense. Implicit there is the fact that `dotplot` maps your input to always produce a numeric x-axis variable inside the command.

To get rid of the ticks, you can just add the option

```
xlabel(, noticks)
```

and they disappear. But now you may complain that the labels are too close to the axis. If Stata could talk back, which the company should be working on for a future version, it would be a little sharp and say that it gave you precisely what you asked for. But there is another way. You can make the tick invisible with

```
xlabel(, tlcolor(bg))
```

because changing the color to `bg` (background) suffices to make the tick invisible. The results of `xlabel(, noticks)` and `xlabel(, tlcolor(bg))` are shown side by side in figure 2.

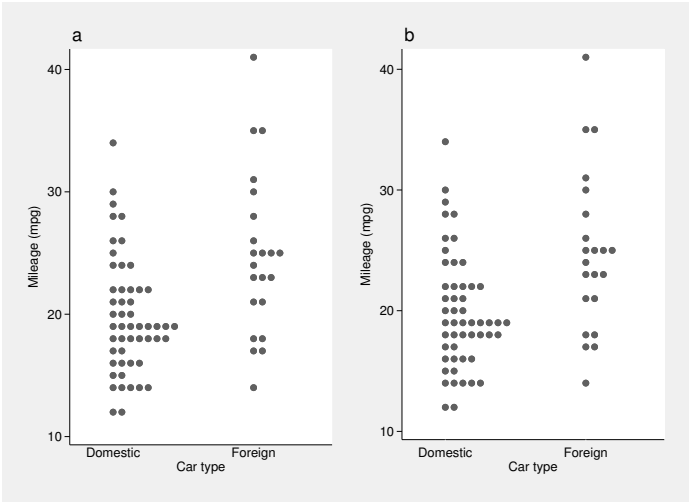

Figure 2. Two different ways of removing the ticks in figure 1. In a, the ticks really are removed. In b, the tick line color is set to the same as the background color, which leaves the label text exactly where it was.

Let us underline a monumentally noteworthy principle: if you cannot see it, that is psychologically equivalent to it not being there. The cosmological, philosophical, and theological versions of this principle are best reserved for other accounts.

But still you might complain. We have removed the ticks and kept the text labels where they were, but now without their ticks, the text labels appear displaced to the left. The next trick here is to put the text labels on a very small slant. `graph` will then flip the text. First, the code respects the sign (not sine!) of the angle—whether zero, negative, positive—in aligning text so that it is centered, or flush-left, or flush-right, given the tick position. The code also respects the angle (or, if you like, the sine) in deciding on the slant.

```
. dotplot mpg, over(foreign) ylabel(, angle(horizontal))
> xlabel(, angle(-0.001) tlcolor(bg))
```

Using -0.001 is a fudge, but if it looks as you wish, then that's what counts; you will be happy, and your readers should be too. Please look ahead to figure 3a for the result.

Obvious, but worth a little emphasis: it is best to have text in graphs aligned horizontally for ease of reading (Tufte [2001] again, if you want a reference). `graph` lets you, for example, put axis labels at, say, 45 degrees, which Stata people often exploit, but that is still a device of despair. The pejorative "giraffe graphics" has been applied to graphics that require willingness to hold the reader's head at arbitrary angles (Cox 2004).

Now again the text is where you want, but possibly again too close to the axis. We have yet further dodges. We can increase the length of our still invisible tick.

```
. dotplot mpg, over(foreign) ylabel(, angle(horizontal))
> xlabel(, angle(-0.001) tlength(*2) tlcolor(bg))
```

Figure 3 shows the two steps side by side.

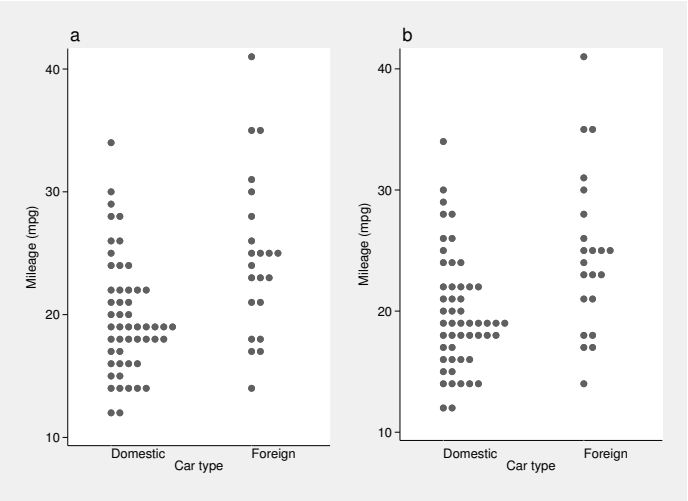

Figure 3. Two steps in aligning axis label text. In a, cunning choice of angle puts text underneath each dot pattern. In b, the invisible tick has been lengthened to push the text away from the axis.

There is an easy corollary: as tick length can be tuned, so also `tlength(0)` is a further way to remove ticks. So we now know three ways to remove ticks: the suboptions `noticks`, `tlcolor(bg)`, and `tlength(0)`. That is not too many ways to do it. Sometimes, we want to combine those suboptions, as we have already seen. Note for your future graphing: if one way does not work because of a program quirk or even a bug, try using another way. (We have not yet touched on `tlstyle()` or `tlwidth()`, with which keen readers may wish to experiment.)

3 Adding ticks where none are shown

Quite the opposite issue can arise with `graph bar` or `graph hbar`—and occasionally also with `graph box`, `graph hbox`, or `graph dot`. Underlying the design and implementation of these commands is the idea that one axis is to be thought of as categorical, so ticks on that axis are not the default and not easily added either.

A common example: people want a bar chart of something that varies with time, say, over a period of years, and seek more control over the time axis than is given or allowed

by `graph bar`. Those wanting this rarely use `graph hbar`, given strong conventions in many fields that time belongs on the horizontal axis, but it would pose the same problems.

Perhaps your years are irregularly spaced, but `graph bar` ignores the gaps. It just puts the categories it sees (years to you) in order. Or you have so many years that the labels run into each other, and you would be happy with labels every other year or every fifth year. Or you wish to combine bar charts with line charts.

Instead of giving detailed tricks for this situation, we confine ourselves to a broad strategic hint. The best way to advance in this circumstance is to retreat. You would be better off with `twoway bar`. Then your time axis (x axis) is numeric, and the rest of `twoway` is there for the asking.

4 Two kinds of ticks, and labels too

A feature often overlooked is that you can have different kinds of ticks (and labels) on each axis that allows them. The secondary labels are called minor, although that describes only their default smaller sizes. You can make the minor labels and ticks bigger, longer, or in other ways different from the default by reaching in and tuning suboptions.

A simple but often helpful use of such secondary ticks or labels is to emphasize individual crucial values: thresholds for action, major events or stages, or whatever.

We illustrate with data on successive records for the number of decimal places of the constant π known correctly at different times. The dataset used is downloadable from the resources for this issue at the *Stata Journal* website. The data were read from https://en.wikipedia.org/wiki/Chronology_of_computation_of_pi on April 18, 2019. The `notes` for the dataset carry some qualifications. The graph to come was inspired by one given by Trefethen (2011, 277).

Because this dataset will become rapidly outdated, note that at the time of writing the record number of decimal places known correctly was as reported on March 14, 2019, by Emma Haruka Iwao: 31,415,926,535,897 digits, although as yet there are no media reports of anyone memorizing them all.

A plot of these data requires (nay, demands) a logarithmic scale for number of decimal places. Whether time should be plotted differently we leave as an open question (Cox 2012). The plot here (figure 4) focuses only on records since 1800. As is evident, a crucial shift in behavior was the first use of digital computers for this purpose in 1949. The code shown for the graph should be almost self-explanatory. Given the very large numbers to be shown, the axis labels are just powers of 10. After reading the dataset, we use `clonevar` to ensure copying of the variable label but promptly `replace` with the logarithm to base 10.

```
. use pi_decplaces, clear
. clonevar log_places = places
. replace log_places = log10(places)
. twoway connected log_places year if year > 1800,
> ylabel(0(1)13, angle(horizontal))
> ms(o) xlabel(1844 1879 1914 1984 2019)
> xmlabel(1949, labsize(*2) tlength(medlarge))
> xtitle("") subtitle(powers of 10, place(w))
```

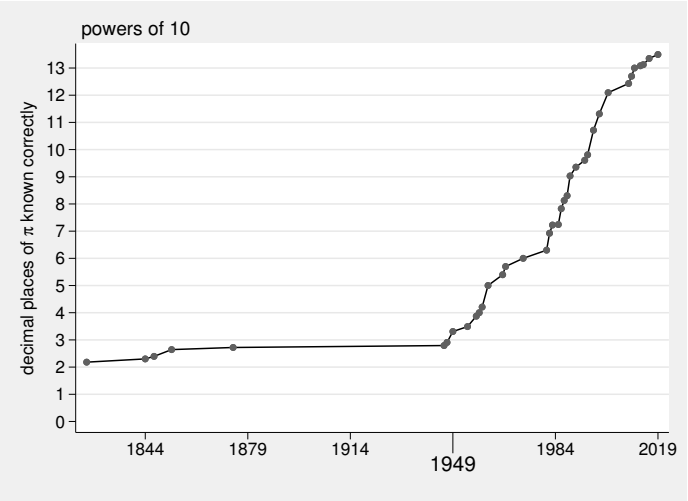

Figure 4. Although this plot is intriguing and inspiring in its own right, here it is a vehicle to show the use of secondary labels to flag a key date, 1949, as the first use of digital computers to calculate the decimal digits of π.

5 Columns and tips with further tips

We here cross-reference other discussions in previous issues of the *Stata Journal*.

Suppose your data span a relatively small number of years (or other time intervals). Hence, each interval can be labeled, and each year accounts for an appreciable fraction of the time axis. The ticks would be better placed at the end of each interval, which is the beginning of the next interval. Text labels with no ticks then belong in the middle of each interval. See Cox (2007) for the details.

Suppose you have chosen a nonlinear scale (logarithmic or other, say, square root or logistic) and wish to see "nice" numbers on your scale, which thus will not be equally spaced on a graph axis. See Cox (2018) for logarithmic scales and Cox (2008, 2012) more generally.

References

Cox, N. J. 2004. Speaking Stata: Graphing categorical and compositional data. *Stata Journal* 4: 190–215.

———. 2007. Stata tip 55: Better axis labeling for time points and time intervals. *Stata Journal* 7: 590–592.

———. 2008. Stata tip 59: Plotting on any transformed scale. *Stata Journal* 8: 142–145.

———. 2012. Speaking Stata: Transforming the time axis. *Stata Journal* 12: 332–341.

———. 2018. Speaking Stata: Logarithmic binning and labeling. *Stata Journal* 18: 262–286.

Trefethen, L. N. 2011. *Trefethen's Index Cards: Forty Years of Notes about People, Words, and Mathematics*. Singapore: World Scientific.

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