The Stata Journal (2004)
4, Number 1, pp. 95–96

Stata tip 6: Inserting awkward characters in the plot

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

Did you know about the function `char()`? `char(`n`)` returns the character corresponding to ASCII code n for $1 \leq n \leq 255$. There are several numbering schemes for so-called ASCII characters. Stata uses the ANSI scheme; a web search for "ANSI character set" will produce tables showing available characters. This may sound like an arcane programmer's tool, but it offers a way to use awkward text characters—either those not available through your keyboard or those otherwise problematic in Stata. A key proviso, however, is that you must have such characters available in the font that you intend to use. Fonts available tend to vary not only with platform but even down to what is installed on your own system. Some good fonts for graphics, in particular, are Arial and Times New Roman.

Let us see how this works by considering the problem of inserting awkward characters in your Stata graphs, say as part of some plot or axis title. Some examples of possibly useful characters are

`char(133)`	ellipsis	…
`char(134)`	dagger	†
`char(135)`	double dagger	‡
`char(169)`	copyright	©
`char(176)`	degree symbol	°
`char(177)`	plus or minus	±
`char(178)`	superscript 2	^{2}
`char(179)`	superscript 3	^{3}
`char(181)`	micro symbol	μ
`char(188)`	one-fourth	$\frac{1}{4}$
`char(189)`	one-half	$\frac{1}{2}$
`char(190)`	three-fourths	$\frac{3}{4}$
`char(215)`	multiply	×

There are many others that might be useful to you, including a large selection of accented letters of the alphabet. You can use such characters indirectly or directly. The indirect way is to place such characters in a local macro and then to refer to that macro within the same program or do-file.

For example, I use data on river discharge, for which the standard units are cubic meters per second. I can get the cube power in an axis title like this:

```
. local cube = char(179)
. scatter whatever, xtitle("discharge, m`cube'/s")
```


Or, I have used Hanning, a binomial filter of length 3:

```
. local half = char(189)
. local quarter = char(188)
. twoway connected whatever,
>        title("Smoothing with weights `quarter':`half':`quarter'")
```

The direct way is to get a macro evaluation on the fly. You can write `` `=char(176)' `` and, in one step, get the degree symbol (for temperatures or compass bearings). This feature was introduced in Stata 7 but not documented until Stata 8. See [P] **macro**.

`char()` has many other uses besides graphs. Suppose that a string variable contains fields separated by tabs. For example, `insheet` leaves tabs unchanged. Knowing that a tab is `char(9)`, we can

```
. split data, p(`=char(9)') destring
```

Note that `p(char(9))` would not work. The argument to the `parse()` option is taken literally, but the function is evaluated on the fly as part of macro substitution.

Note also that the SMCL directive {`c #`} may be used for some but not all of these purposes. See [P] **smcl**. Thus,

```
. scatter whatever, xtitle("discharge, m{c 179}/s")
```

would work, but using a SMCL directive would not work as desired with `split`.