The Stata Journal (2011)
11, Number 1, pp. 149–154

Stata tip 96: Cube roots

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

1 Introduction

Plotting the graph of the cube function $x^3 = y$ underlines that it is single-valued and defined for arguments everywhere on the real line. So also is the inverse or cube root function $x = y^{1/3} = \sqrt[3]{y}$. In Stata, you can see a graph of the cube function by typing, say, `twoway function x^3, range(-5 5)` (figure 1). To see a graph of its inverse, imagine exchanging the axes. Naturally, you may want to supply other arguments to the `range()` option.

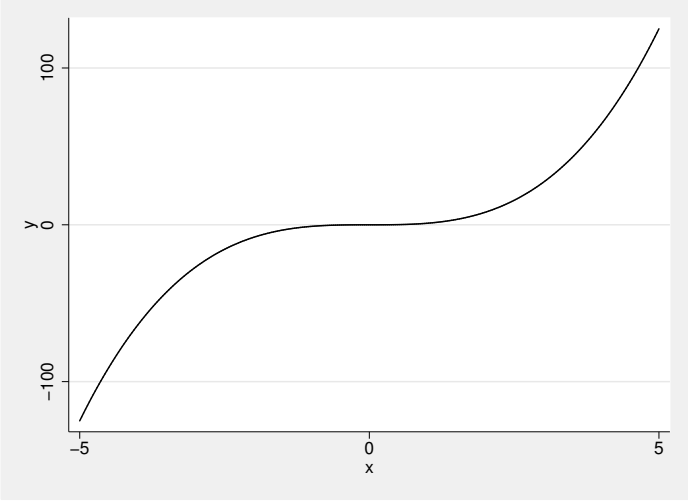

Figure 1. The function x^3 for $-5 \leq x \leq 5$

Otherwise put, for any $a \geq 0$, we can write

$$(-a)(-a)(-a) = -a^3 \qquad (a)(a)(a) = a^3$$

so that cube roots are defined for negative, zero, and positive cubes alike.

This concept might well be described as elementary mathematics. The volume of literature references, say, for your colleagues or students, could be multiplied; I will single

out Gullberg (1997) as friendly but serious and Axler (2009) as serious but friendly. Elementary or not, Stata variously does and does not seem to know about cube roots:

```
. di 8^(1/3)
2
. di -8^(1/3)
-2
. di(-8)^(1/3)
.
. set obs 1
obs was 0, now 1
. gen minus8 = -8
. gen curtminus8 = minus8^(1/3)
(1 missing value generated)
```

This tip covers a bundle of related questions: What is going on here? In particular, why does Stata return missing when there is a perfectly well-defined result? How do we get the correct answer for negative cubes? Why should we ever want to do that? Even if you never do in fact want to do that, the examples above raise details of how Stata works that you should want to understand.

Those who know about complex analysis should note that we confine ourselves to real numbers throughout.

2 Calculation of cube roots

To Stata, cube roots are not special. As is standard with mathematical and statistical software, there is a dedicated square-root function `sqrt()`; but cube roots are just powers, and so they are obtained by using the `^` operator. I always write the power for cube roots as (1/3), which ensures reproducibility of results and ensures that Stata does the best it can to yield an accurate answer. Experimenting with 8 raised to the powers 0.33, 0.333, 0.3333, and so forth will show that you would incur detectable error even with what you might think are excellent approximations. The parentheses around 1/3 are necessary to ensure that the division occurs first, before its result is used as a power.

What you understand from your study of mathematics is not necessarily knowledge shared by Stata. The examples of cube rooting -8 and 8 happen to have simple integer solutions -2 and 2, but even any appearance that Stata can work this out as you would is an illusion:

```
.  di 8^(1/3)
2
.  di %21x 8^(1/3)
+1.fffffffffffffX+000
.  di %21x 2
+1.0000000000000X+001
```

Showing here results in hexadecimal format (Cox 2006) reveals what might be suspected. No part of Stata recognizes that the answer should be an integer. The problem is being treated as one in real (not integer) arithmetic, and the appearance that 2 is the solution is a pleasant side effect of Stata's default numeric display format. Stata's answer is in fact a smidgen less than 2. What is happening underneath? I raised this question with William Gould of StataCorp and draw on his helpful comments here. He expands further on the matter within the Stata blog; see "How Stata calculates powers" at http://blog.stata.com/2011/01/20/how-stata-calculates-powers/.

The main idea here is that Stata is using logarithms to do the powering. This explains why no answer is forthcoming for negative arguments in the `generate` statement: because the logarithm is not defined for such arguments, the calculation fails at the first step and is fated to yield missings.

We still have to explain why `di -8^(1/3)` yields the correct result for the cube root of −8. That is just our good fortune together with convenient formatting. Stata's precedence rules ensure that the negation is carried out last, so this request is equivalent to `-(8^(1/3))`, a calculation that happens to have the same answer. We would not always be so lucky: for the same reason, `di -2^2` returns −4, not 4 as some might expect.

The matter is more vexed yet. Not only does 1/3 have no exact decimal representation, but also, more crucially, it has no exact binary representation. It is easy enough to trap 0 as a special case so that Stata does not fail through trying to calculate ln 0. But Stata cannot be expected to recognize 1/3 as its own true self. The same goes for other odd integer roots (powers of 1/5, 1/7, and so forth) for which the problem also appears.

To get the right result, human intervention is required to spell out what you want. There are various work-arounds. Given a variable `y`, the cube root is in Stata

```
cond(y < 0, -((-y)^(1/3)), y^(1/3))
```

or

```
sign(y) * abs(y)^(1/3)
```

The `cond()` function yields one of two results, depending on whether its first argument is nonzero (true) or zero (false). See Kantor and Cox (2005) for a tutorial if desired. The `sign()` function returns −1, 0, or 1 depending on the sign of its argument. The `abs()` function returns the absolute value or positive square root. The second of the two solutions just given is less prone to silly, small errors and extends more easily to Mata. The code in Mata for a scalar `y` is identical. For a matrix or vector, we need the generalization with elementwise operators,

```
sign(y) :* abs(y):^(1/3)
```

3 Applications of cube roots

Cube roots do not have anything like the utility, indeed the natural roles, of logarithms or square roots in data analysis, but they do have occasional uses. I will single out three reasons why.

First, the cube root of a volume is a length, so if the problem concerns volumes, dimensional analysis immediately suggests cube roots as a simplifying transformation. A case study appears in Cox (2004).

Second, the cube root is also a good transformation yielding approximately normal distributions from gamma or gamma-like distributions. See, for example, McCullagh and Nelder (1989, 288–289). Figure 2 puts normal probability plots for some raw and transformed chi-squared quantiles for 4 degrees of freedom side by side.

```
. set obs 99
. gen chisq4 = invchi2(4, _n/100)
. qnorm chisq4, name(g1)
. gen curt_chisq4 = chisq4^(1/3)
. qnorm curt_chisq4, name(g2)
. graph combine g1 g2
```

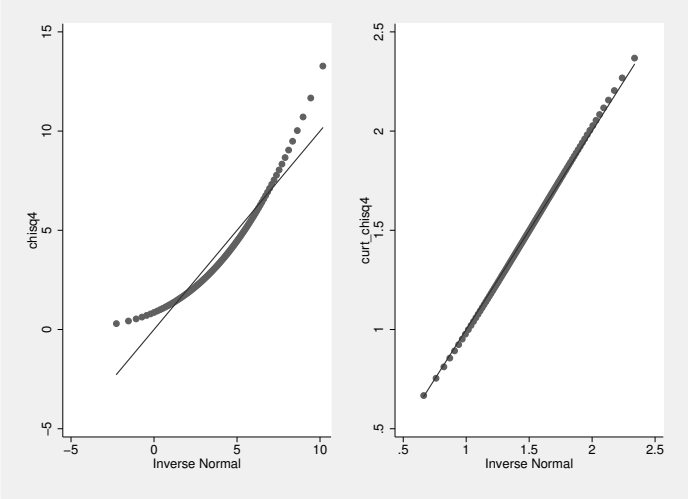

Figure 2. Ninety-nine quantiles from a chi-squared distribution with 4 degrees of freedom are distinctly nonnormal, but their cube roots are very nearly normal

The cube root does an excellent job with a distinctly nonnormal distribution. It has often been applied to precipitation data, which are characteristically right-skewed and sometimes include zeros (Cox 1992).

Third, beyond these specific uses, the cube root deserves wider attention as the simplest transformation that changes distribution shape but is easily applicable to values with varying signs. It is an odd function—an odd function f is one for which

$f(-x) = -f(x)$—but in fact, it treats data evenhandedly by preserving the sign (and in particular, mapping zeros to zeros).

There are many situations in which response variables in particular can be both positive and negative. This is common whenever the response is a balance, change, difference, or derivative. Although such variables are often skew, the most awkward property that may invite transformation is usually heavy (long or fat) tails, high kurtosis in one terminology. Zero usually has a strong substantive meaning, so that we should wish to preserve the distinction between negative, zero, and positive values. (Celsius or Fahrenheit temperatures do not really qualify here, because their zero points are statistically arbitrary, for all the importance of whether water melts or freezes.)

From a different but related point of view, there are frequent discussions in statistical (and Stata) circles of what to do when on other grounds working on logarithmic scales is indicated, but the data contain zeros (or worse, negative values). There is no obvious solution. Working with logarithms of $(x + 1)$ or more generally $(x + k)$, k being large enough to ensure that $x + k$ is always positive, variously appeals and appalls. Even its advocates have to admit to an element of fudge. It certainly does not treat negative, zero, and positive values symmetrically.

Other solutions that do precisely that include $\text{sign}(x)\ \ln(|x| + 1)$ and $\text{asinh}(x)$, although such functions may appear too complicated or esoteric for presentation to some intended audiences. As emphasized earlier, the cube root is another and simpler possibility. It seems unusual in statistical contexts to see its special properties given any mention, but see Ratkowsky (1990, 125) for one oblique exception.

One possible application of cube roots is whenever we wish to plot residuals but also to pull in the tails of large positive and negative residuals compared with the middle of the distribution around zero. See Cox (2008) for pertinent technique. In this and other graphical contexts, the main question is not whether cube roots yield approximately normal distributions, but simply whether they make data easier to visualize and to think about.

These issues extend beyond cube roots. As hinted already, higher odd integer roots (fifth, seventh, and so forth) have essentially similar properties although they seem to arise far less frequently in data analysis. Regardless of that, it is straightforward to define powering of negative and positive numbers alike so long as we treat different signs separately, most conveniently by using `sign()` and `abs()` together with the power operator. That is, the function may be defined as $-(-y)^p$ if $y < 0$, and y^p otherwise.

References

Axler, S. 2009. *Precalculus: A Prelude to Calculus*. Hoboken, NJ: Wiley.

Cox, N. J. 1992. Precipitation statistics for geomorphologists: Variations on a theme by Frank Ahnert. *Catena* 23 (Suppl.): 189–212.

———. 2004. Speaking Stata: Graphing model diagnostics. *Stata Journal* 4: 449–475.

———. 2006. Stata tip 33: Sweet sixteen: Hexadecimal formats and precision problems. *Stata Journal* 6: 282–283.

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

Gullberg, J. 1997. *Mathematics: From the Birth of Numbers*. New York: W. W. Norton.

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

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Ratkowsky, D. A. 1990. *Handbook of Nonlinear Regression Models*. New York: Marcel Dekker.