Yeo-Johnson Power Transformations

Sanford Weisberg

Department of Applied Statistics, University of Minnesota, St. Paul, MN 55108-6042.

Supported by National Science Foundation Grant DUE 97-52887.

October 26, 2001

Abstract

This paper describes an **Arc** add-in for using the Yeo-Johnson power transformations in place of the Box-Cox power transformations in various places in **Arc**.

1 Introduction

Transformations play a central role in regression analysis (Cook and Weisberg, 1999). Often, one chooses a transformation from a parametric family of transformations. The family that is used most often is the *Box-Cox power family*, defined by

$$y^{(\lambda)} = \begin{cases} (y^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(y) & \text{if } \lambda = 0 \end{cases} \tag{1}$$

where y is a list of n *strictly positive* numbers. The Box-Cox family is useful because it is equivalent to the family of power transformation, so the parameter λ is easily understood, and it includes the important special cases of untransformed, inverse, logarithmic, and square and cube root. The Box-Cox family is used in many places in **Arc**, in particular for choosing response transformations and for transforming a set of predictors toward multivariate normality.

Several attempts to define transformation families variables y that include negative values have been suggested. One possiblity is to consider transformations of the form $(y + \gamma)^\lambda$, where γ is sufficiently large to insure that $y + \gamma$ is strictly positive. In principle, (γ, λ) could be estimated simultaneously, although in practice estimates of γ are highly variable. Alternatively, other families of transformations such as the *folded power family* (see Cook and Weisberg, 1999, p. 330) have been proposed, but are rarely used because the resulting transformations have poor properties.

Yeo and Johnson (2000) have proposed an new family of distributions that can be used without restrictions on y that have many of the good properties of the Box-Cox power family. These transformations are defined by:

$$\psi(\lambda, y) = \begin{cases} ((y+1)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{2-\lambda} - 1)]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases} \tag{2}$$

If y is strictly positive, then the Yeo-Johnson transformation is the same as the Box-Cox power transformation of $(y + 1)$. If y is strictly negative, then the Yeo-Johnson transformation is the Box-Cox power transformation of $(-y + 1)$, but with power $2 - \lambda$. With both negative and positive values, the transformation is a mixture of these two, so different powers are used for positive and negative values. In this latter case, interpretation of the transformation parameter is difficult, as it has a different meaning for $y \geq 0$ and for $y < 0$. Figure 1 shows the Box-Cox transformation and and Yeo-Johnson transformation for the values of $\lambda = -1, 0, .5$ that are the most common values of the Box-Cox transformations. For positive values, the two transformations differ in their behavior with values close to zero, with the Box-Cox transformations providing a much larger change for small values than does the Yeo-Johnson transformations.

Although interpretation of the Yeo-Johnson transformation parameter is difficult, this family can be useful in procedures for selecting a transformation for linearity or normality.

2 Getting the code

Download the file from the Add-ons page at http://www.stat.umn.edu/arc. Place it in the Extras directory in your Arc directory. If you are using Unix, either place this file in /usr/local/lib/Arc/Extras, or in an Extras directory where you start **Arc**. The file will be loaded automatically whenever you start **Arc**.

3 Using Yeo-Johnson transformations

3.1 Transforming the response

Select “Choose response transformation” from the regression menu as usual, and then select the “Yeo-Johnson” family in the dialog.

3.2 Saving Yeo-Johnson transformations as variables

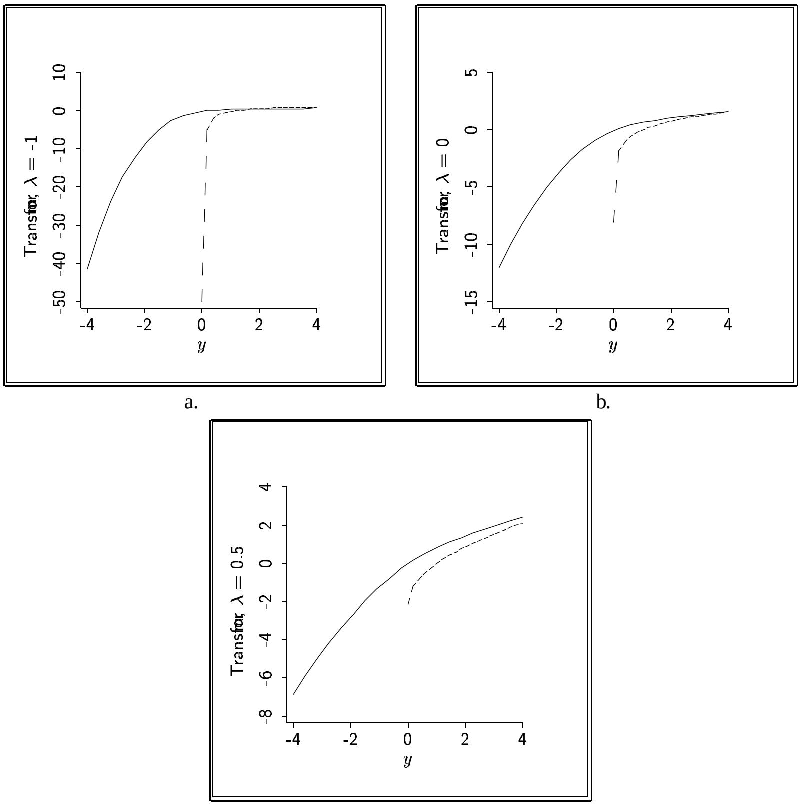
Use the “Transformations. . . ” item as usual, and select Yeo-Johnson power from the list in the dialog. Setting the constant c will give the Yeo-Johnson transformation of $y + c$.

3.3 Transformation slidebars

On scatterplot matrices, you can toggle between using the Box-Cox family and the Yeo-Johnson family using an item in the “Transformations” plot control. You can’t mix the two, and must use all of one type or the other. On scatterplots and boxplots, you cannot change the transformation family using a plot control, but the following typed command will do the trick. If the name of a graph is, say, tt plot5, type

```
> (send plot5 :transformation-family 'yj-power)
```

Use the argument `box-cox` for the Box-Cox family. Alternatively, you can change the default transformation family, as described in the next section.



a.

b.

c.

Figure 1: Comparison of Box-Cox and Yeo-Johnson power transformations for $\lambda = -1, 0.5$. The Box-Cox transformations (and simple power transformations) behave very differently for values of y closase to zero than do the Yeo-Johnson transformations.

3.4 Default family

A setting in the Settings menu for `*transformation-default-family*` can be changed to `yj-power` to make the Yeo-Johnson family the default transformation family in all graphs.

3.5 The function

The Yeo-Johnson family (actually a normalized version of it to have Jacobean equal to one) is computed by the function `yj-power`, shown in Table 1.

Table 1: The lisp function for computing the Yeo-Johnson transformation. The argument `included` is `t` for cases to be used in computing and `nil` otherwise. The function `geometric-mean` computes the geometric mean of the included cases in its argument, and `find-obs` finds the indices of non-missing cases.

```
(defun yj-power (y p &key included (normalize t))
"Function args: (data power &key included (normalize t))
This function returns the normalized Yeo-Johnson transformation,
suggested by In-Kwon Yeo and Richard A. Johnson (2000).  A new family
of power transformations to improve normality or symmetry, Biometrika,
87, 954-959."
  (let* ((lam (if (< (abs p) 1.e-6) 0 p))
         (obs (find-obs y))
         (gm (geometric-mean (ˆ (+ 1 (abs (select y obs)))
                                (if-else (< (select y obs) 0) -1 1))
                       (if included (which (select included obs)))))
         (transform (mapcar #’(lambda (x)
            (cond ((and (>= x 0) (/= lam 0))
                   (/ (- (ˆ (+ x 1) lam) 1) lam))
                  ((and (>= x 0) (=  lam 0))
                   (log (+ 1 x)))
                  ((and (<  x 0) (/= lam 2))
                   (- (/ (- (ˆ (+ (- x) 1) (- 2 lam)) 1) (- 2 lam))))
                  (t (- (log (+ (- x) 1)))))) (select y obs)))
         (z y))
    (setf (select z obs) transform)
    (if normalize (/ z (ˆ gm (- lam 1))) z)))
```

4 Bug Reports

Please send bug reports to sandy@stat.umn.edu.

5 References

Cook, R. D. and Weisberg, S. (1999). *Applied Regression Including Computing and Graphics*, New York: Wiley.
Yeo, In-Kwon and Johnson, Richard (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87, 954-959.