The Stata Journal (2010)
**10**, Number 2, pp. 200–214

# Optimal power transformation via inverse response plots

Charles Lindsey
Texas A & M University
Department of Statistics
College Station, TX
lindseyc@stat.tamu.edu

Simon Sheather
Texas A & M University
Department of Statistics
College Station, TX
sheather@stat.tamu.edu

**Abstract.** We present a new Stata command, `irp`, that generates the inverse response plot (Cook and Weisberg, 1994, *Biometrika* 81: 731–737) of a response on its predictors. Using the inverse response plot, an appropriate scaled power transformation for the positive response variable can be found so that the transformed response mean is linear in the predictors. The optimal transformation is displayed in the plot, as are user-specified guesses. By using the graphical display, the user may determine whether an appropriate transformation exists as well as determine its value. We demonstrate the `irp` command using both a generated and a real example.

**Keywords:** st0188, irp, regress, scaled power transformation

## 1 Theory/motivation

When investigating the regression of the positive variable $y$ on $x_1, \ldots, x_p$, one may discover or suspect a nonlinear relationship between the response and the predictors. To overcome this difficulty, one may decide to transform $y$ or its predictors so that the transformed variables have a linear relationship via the invertible power transformation $t$. There are methods for this procedure that are already implemented in Stata (see [R] **boxcox**). These methods are useful and powerful, but they mostly rely on numeric output.

By using an inverse response plot, we provide a concise and intuitive plot that demonstrates the effectiveness of the transformation by the closeness of the transformation's curve to points in a scatterplot. This plot may significantly ease comparison between different transformations and vividly demonstrate the adequacy of the optimal transformation.

Cook and Weisberg (1994) used results in Li and Duan (1989) to develop the inverse response plot. To use an inverse response plot to estimate $t$, Cook and Weisberg showed that the following results should also hold.

 

For unknown constant matrices $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_p)$, $\Lambda_{p\times p} = (\lambda_{ij})$, $\Phi_{p\times p} = (\phi_{ij})$, and zero mean random variable $\epsilon$, where $\epsilon$ is independent of the $x_0, \ldots, x_p$,

$$t(y) = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_p x_p + \epsilon \qquad (1)$$

$$p \neq 1 \qquad E(x_i \,|\, x_j) = \lambda_{ij} + \phi_{ij} x_j, \qquad i, j = 1, \ldots, p \qquad (2)$$

$$p = 1 \qquad x_1 \quad \text{is elliptically symmetric} \qquad (3)$$

Both (2) and (3) are implied by (multivariate) normality of the predictors. Equation (2) need only apply approximately for the inverse response plot to work in practice. Equation (3) is not satisfied if the predictor is skewed. A skewed predictor is easy to check for with a kernel density plot (see [R] **kdensity**) or a Q–Q plot (see [R] **diagnostic plots**). Similarly, checking the linear relation of (2) with a matrix plot (see [G] **graph matrix**) is simple. Equation (1) is difficult to check before using the inverse response plot, because (1) depends directly on $t$, which we use the inverse response plot to estimate. If we find that (2) and (3) are met, and we obtain a satisfactory estimate of $t$ from the inverse response plot, then (1) is implied.

Supposing that (2) and (3) are met, we estimate $t$ by plotting the fitted values $\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \cdots + \widehat{\beta}_p x_p$ for the regression of $y$ on $x_1, \ldots, x_p$ on the vertical axis versus the actual response values $y$ on the horizontal axis. We fit scaled power curves to this scatterplot. The power of the curve suggests the functional form of $t$. The curve fitting is done by transforming $y$ with a scaled power transformation, and then scaling and relocating the transformed values so that they are an optimal fit for the scattered points.

Essentially, we regress $\widehat{y}$ on the transformed $y$. Then we plot a curve through the predictions of $\widehat{y}$ from the transformed $y$. This methodology of predicting $t$ was developed by Cook and Weisberg (1994) based on results of Li and Duan (1989). In particular, Cook and Weisberg show that this method works even though the fitted values are from an invalid regression model.

It will be instructive to look at a sample inverse response plot (see figure 1). Here $y$ is the cube of its predictor and error term. (We will return to this example in the next section, where all details will be given.) The first entry in the legend corresponds to the optimal choice of transformation. We have $y^{1/3} = x + \epsilon$, so this choice is quite logical. The 0 curve shows how $\log(y)$ approximates $y^{1/3}$. The 1 curve shows how poorly $y$ approximates $y^{1/3}$.

(*Continued on next page*)

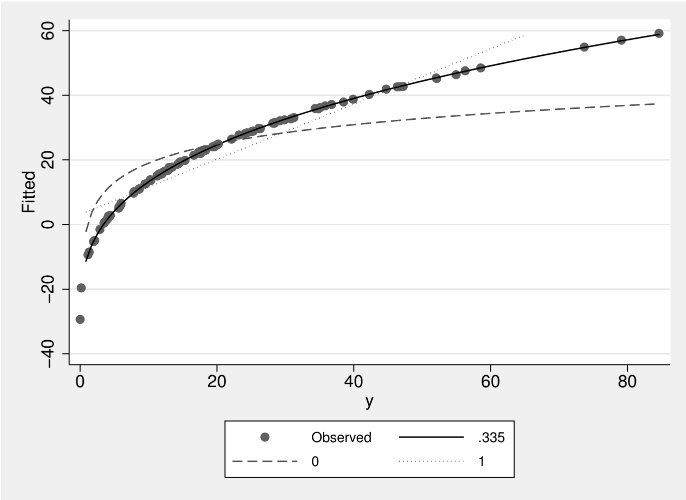


Figure 1. Generated example $y = (x + \epsilon)^3$

In this example, $t$ was an unscaled power transformation. Sometimes using such a transformation will switch the direction of the relationship between the response and predictors. For example, suppose $y$ and $x$ are directly related. When we transform $y$ to $1/y$, we make $x$ and the new $y$ inversely related.

We define a scaled power transformation as

$$\psi_s(y, \lambda) = \begin{pmatrix} \frac{y^\lambda - 1}{\lambda}, & \text{if} \lambda \neq 0 \\ \log y, & \text{if} \lambda = 0 \end{pmatrix}$$

Scaled power transformations preserve the direction of associations that the transformed variable had with other variables. So scaled power transformations will not switch or erase collinear relationships of interest. The power of the optimal scaled power transformation should in principle be identical to the power of the optimal unscaled transformation.

Finding the closest fitting scaled power curve (and thus the best estimate of $t$) is equivalent to finding the scaled power transformation that, when performed on $y$, minimizes the residual sum of squares for the regression of $\widehat{y}$ on the transformed $y$. The optimal transformation is found in `irp` via a numerical optimization on this residual sum of squares.

The power of the closest fitting scaled power curve may be quite esoteric to the user or the subject matter experts that he or she is helping. So it is often necessary to try more than one transformation, picking the curve with the best fit and most plausibility with respect to the subject matter of the data. We visually compared three transformations in our first example. The process was intuitive and simple.

We can augment a scaled power transformation by multiplying the transformed variable by the original's geometric mean, $\text{gm}(Y) = \exp\big(1/n \sum_{i=1}^{n} \log y_i\big)$ raised to the $1-\lambda$ power. This transformation would maintain the scale of the transformed variables. We denote this as a modified scaled power transformation:

$$\psi_M(y,\lambda) = \text{gm}(Y)^{1-\lambda}\psi_s(y,\lambda)$$

Box and Cox (1964) make the additional assumption that the $\psi_M(y,\lambda)$ is normal with a mean that is linearly determined by $x_1,\ldots,x_p$ for the true transformation power $\lambda$. Under Box and Cox's assumption, the optimal transformation power $\lambda$ minimizes the residual sum of squares for the regression of $\psi_M(y,\lambda)$ on $x_1,\ldots,x_p$. Using this method does allow for statistical inference on $\lambda$ through likelihood methods, but it also imposes distributional assumptions.

## 2 Use and a generated example

The `irp` command is straightforward to use. We will demonstrate this as we revisit the generated example that produced figure 1.

In this example, we generate predictor `x` distributed as $N(2.6, 0.75^2)$ and error term $\epsilon$ distributed as $N(0, 0.01^2)$. Our response variable $\texttt{y} = (\texttt{x} + e)^3$. See figure 2.

```
. set obs 100
obs was 0, now 100
. set seed 1234
. generate x = .75*invnormal(runiform()) + 2.6
. generate e = .01*invnormal(runiform())
. generate y = (x + e)^3
```

(*Continued on next page*)

```
. twoway scatter y x
```

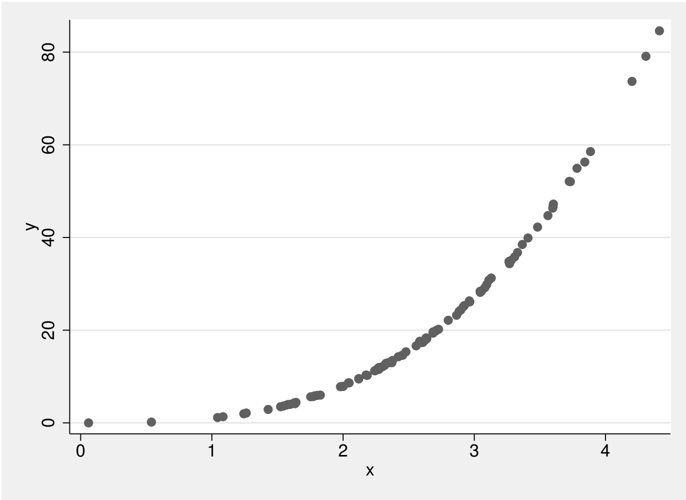


Figure 2. x versus y, $y = (x + \epsilon)^3$

Clearly, y and x lack a linear relationship. So linear regression of y on x is ill-advised. We will see if we can use an inverse response plot to transform y to linearize its relationship with x. See figure 3.

```
. summarize y
```

| Variable | Obs | Mean | Std. Dev. | Min | Max |
|---|---|---|---|---|---|
| y | 100 | 21.23332 | 17.5544 | .0000891 | 84.59297 |

```
. swilk x
```

Shapiro-Wilk W test for normal data

| Variable | Obs | W | V | z | Prob>z |
|---|---|---|---|---|---|
| x | 100 | 0.99201 | 0.660 | -0.922 | 0.82175 |

```
. qnorm x
```

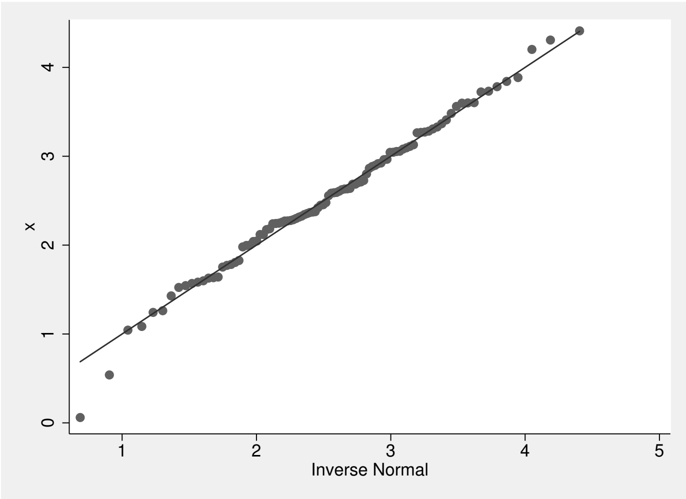


Figure 3. `x` normal quantile plot

The response `y` is positive. The `swilk` and `qnorm` commands show that `x` is consistent with a normal distribution. So we can use an inverse response plot to determine a scaled power transformation for `y` so that the new `y` is linear in `x`.

The `irp` command has the following syntax:

`irp` *depvar indepvars* [ *if* ] [ *in* ] [ `,` `optimum` `try(`*numlist*`)` `old(`*matname*`)` `generate` *twoway_options* ]

The *depvar* variable is the response variable of the inverse response plot, and *indepvars* are the predictors. A subset of the data may be specified with the optional *if* condition or *in* range. The `optimum` option instructs `irp` to find the best scaled power transformation for *depvar* by a numerical optimization. The `try()` option allows the user to specify a list of transformation powers to be examined in the plot. The `old()` option allows the user to specify a matrix containing calculations from a previous execution of `irp` that will be incorporated into the inverse response plot. The `generate` option outputs the transformed response variables for the given input powers to the data. Finally, additional graphical options are allowed to be specified in *twoway_options*.

Now let's test `irp` on our example. See figure 1 for the graphical output.

(*Continued on next page*)

```
. irp y x, opt try(0 1)
```

| Response | y |
|---|---|
| Fitted | 20.34*x + -30.58 |

| Optimal Power | .3353386 |
|---|---|

| Power | RSS( F \| R ) |
|---|---|
| .3353386 | 4.263288 |
| 0 | 8082.853 |
| 1 | 3787.411 |

```
. return list
scalars:
            r(optimum) =  .335338566748325
matrices:
            r(tranres) :  3 x 4
. matrix list r(tranres)
r(tranres)[3,4]
         Power         RSS   Intercept       Slope
r1   .33533857   4.2632882  -30.373909   6.7538795
r1           0    8082.853  -.80120822   8.5953352
r2           1   3787.4108   3.0839681   .85475809
```

The irp command provides the residual sum of squares for the regression of the fitted values on each scaled power transformation attempted on the response. Under the default display of irp, the residual sum of squares are shown under the RSS( F | R ) column. The corresponding power transformation parameters are given in the Power column. This quantitative information can help the user understand the magnitude of fit difference between transformations.

In the returned results, the slopes and intercepts for each regression of the fitted values on a transformed response are also provided. If the user wishes to compare additional transformation powers without reperforming any calculations, then the user can pass the name of a matrix with the same format as r(tranres) into the old() option. This option is only useful when the user wants to redraw an inverse response plot that took considerable computation to draw the first time. We will demonstrate the use of the old() option in our next example, in section 3.

In this example, the additional quantitative information provided by irp strongly corroborates the notion that 1/3 is an appropriate transformation power.

We will put 1/3 into the try() list and reexecute irp to compare the fit of 1/3 and the optimal transformation. See figure 4.

```
. irp y x, opt try(.3333333)
```

| Response | y |
|---|---|
| Fitted | 20.34*x + -30.58 |

| Optimal Power | .3353386 |
|---|---|

| Power | RSS( F \| R ) |
|---|---|
| .3353386 | 4.263288 |
| .3333333 | 4.36517 |

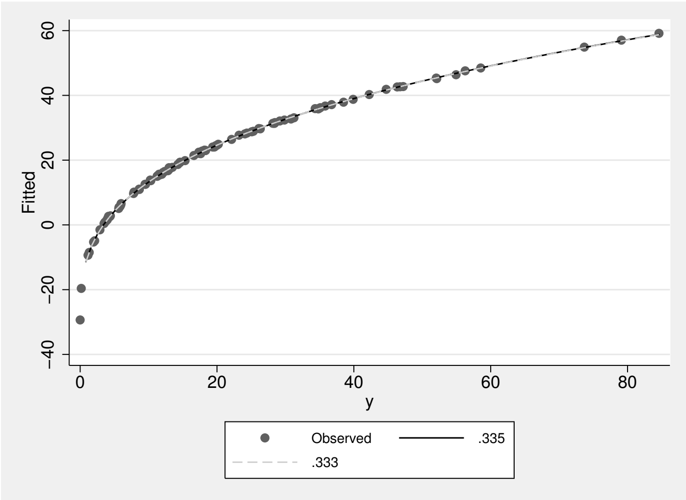


Figure 4. 1/3 versus optimum inverse response plot

The fits are nearly identical. We will transform y to $\psi_s(\mathtt{y}, 1/3)$ and recheck the linearity of y in x. See figure 5.

```
. generate trany = (y^(1/3) - 1)/(1/3)
. summarize trany
```

| Variable | Obs | Mean | Std. Dev. | Min | Max |
|---|---|---|---|---|---|
| trany | 100 | 4.641468 | 2.390982 | -2.865988 | 10.1694 |

(Continued on next page)

```
. twoway scatter trany x
```

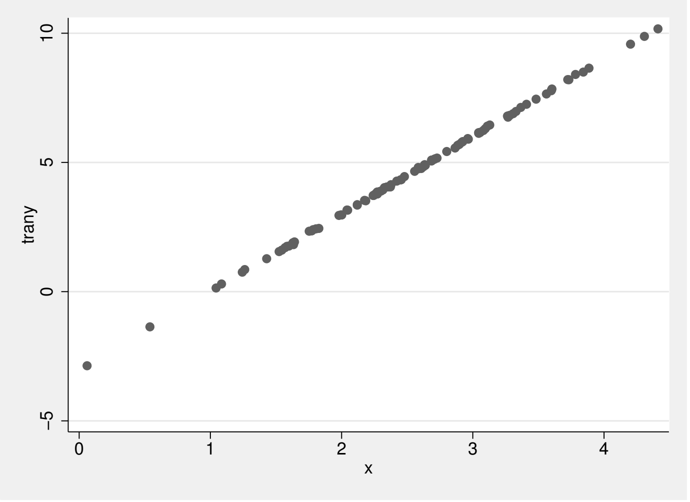


Figure 5. x versus $\psi_s(\mathtt{y}, 1/3)$, $\mathtt{y} = (\mathtt{x} + \epsilon)^3$

We could have used an unscaled power transformation in this case without disturbing the direction of association between `y` and `x`, but it was instructive to practice implementation of the scaled power transformation. We will have strong reason to use one in our next example.

# 3 Real example

Consider the UCI Machine Learning Repository (1993) dataset `auto-mpg.dta`. The data contain information on individual automobiles. Of interest is the regression of miles per gallon, `mpg`, on the car's weight, `wt`, and horsepower, `hp`. A matrix plot of the three variables reveals a problem and the path to a solution. See figure 6.

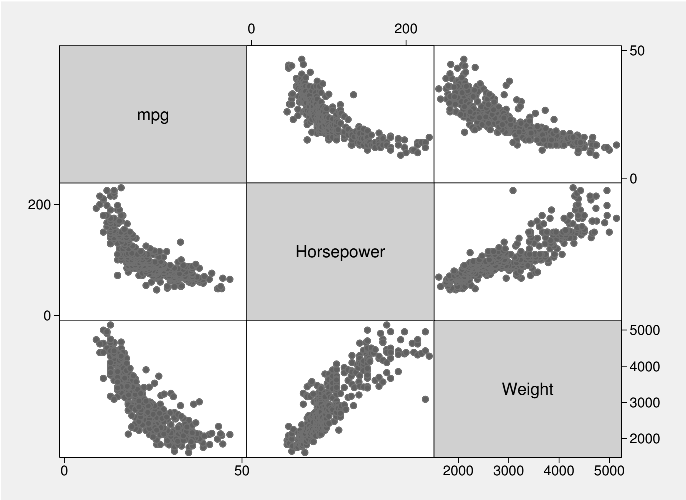


Figure 6. Auto 1993 example, matrix plot

The `mpg` variable appears to have a nonlinear relationship with both `hp` and `wt`, but the two predictors appear to have an approximately linear relationship. Therefore (2) for the use of an inverse response plot is satisfied.

We will now use `irp` to render an inverse response plot. We are going to be conservative and not guess any transformation powers. See figure 7.

```
. irp mpg hp wt, optimum
```

| | |
|---|---|
| Response | mpg |
| Fitted | -.0473*hp + -.0058*wt + 45.64 |

| | |
|---|---|
| Optimal Power | -.8776173 |

| Power | RSS( F \| R ) |
|---|---|
| -.8776173 | 3149.255 |

(*Continued on next page*)

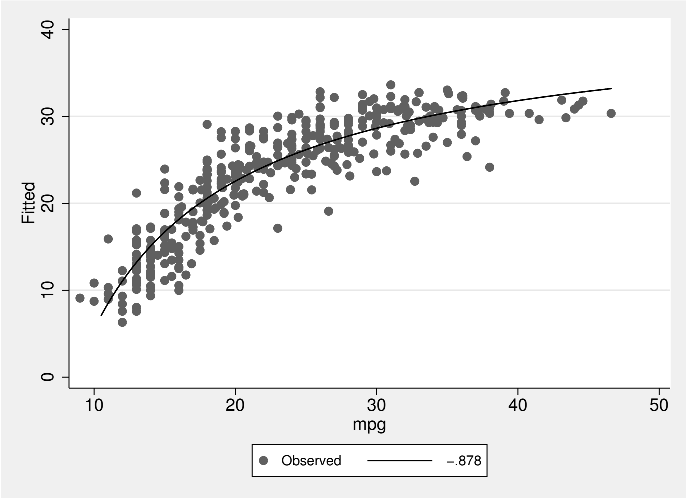


Figure 7. Auto 1993 example, inverse response plot optimum

```
. matrix b = r(tranres)
```

We stored the results of our optimum power calculation in the matrix `b`. Momentarily, we will use them again. Miles per gallon is often translated to gallons per mile via the reciprocal transformation 1/`mpg`. We will compare this transformation with our optimal transformation. See figure 8.

```
. irp mpg hp wt, try(-1) old(b) generate
```

| | |
|---|---|
| Response | mpg |
| Fitted | -.0473*hp + -.0058*wt + 45.64 |

| | |
|---|---|
| Optimal Power | Not Calculated/Re-Calculated |

| Power | RSS( F \| R ) |
|---|---|
| -1 | 3157.848 |
| -.8776173 | 3149.255 |

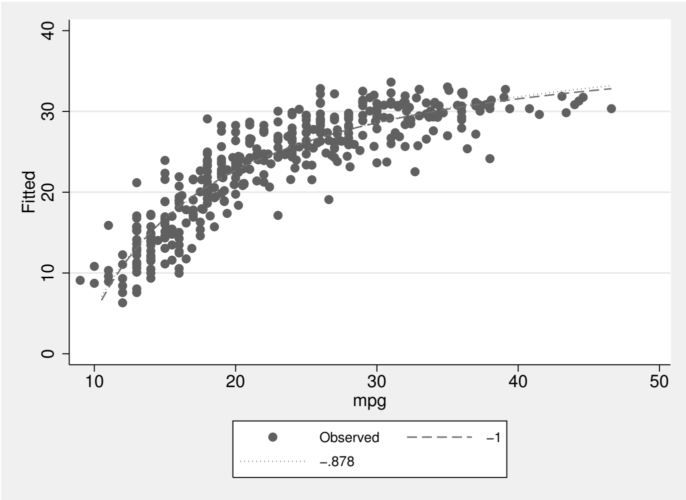


Figure 8. Auto 1993 example, inverse response plot optimum, −1

The reciprocal transformation is nearly as good as the optimal transformation. Because it makes more sense theoretically, we will use the reciprocal transformation instead of the optimal.

Note how the matrix `b` was used in the last `irp` call. We were able to save computation time by using the previously calculated results in `b`. This was not necessary; we could have executed `irp mpg hp wt, optimum try(-1)` and obtained the same results.

It may be tempting to just raise `mpg` to the −1 power and ignore whatever relationships `mpg` had before we transformed it. Under the reciprocal transformation, any direct relationships `mpg` had with other variables are now inverse relationships. Similarly, all inverse relationships `mpg` had with other variables are now direct relationships.

The following correlation matrices demonstrate what happens when we use a reciprocal transformation. The `ecc` variable is the acceleration capability of the automobile.

```
. correlate mpg hp wt acc
(obs=392)

             |      mpg       hp       wt      acc
-------------+------------------------------------
         mpg |   1.0000
          hp |  -0.7784   1.0000
          wt |  -0.8322   0.8645   1.0000
         acc |   0.4233  -0.6892  -0.4168   1.0000

. generate rmpg = 1/mpg
```

```
. correlate rmpg hp wt acc
(obs=392)

             |     rmpg       hp       wt      acc
-------------+------------------------------------
        rmpg |   1.0000
          hp |   0.8548   1.0000
          wt |   0.8851   0.8645   1.0000
         acc |  -0.4563  -0.6892  -0.4168   1.0000
```

It may be wise to maintain the direction of the variable relationships `mpg` holds so that we do not falsely take them for granted and make a mistake. We can easily maintain the direction by transforming `mpg` using the scaled power transformation $\psi_s(\texttt{mpg}, -1) = -1/\texttt{mpg} + 1$. We will generate a new variable using this formula and compare it with that produced by `irp` in the `irp1` variable.

The following correlation matrix demonstrates how the scaled power transformation maintains the directionality of `mpg` relationships and how the generated variable from `irp` perfectly matches with the scaled power transformation.

```
. describe irp1

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
irp1            float  %9.0g                  Transform Power = -1

. generate srmpg = -1/mpg + 1

. correlate mpg srmpg irp1 hp wt acc
(obs=392)

             |      mpg    srmpg     irp1       hp       wt      acc
-------------+------------------------------------------------------
         mpg |   1.0000
       srmpg |   0.9359   1.0000
        irp1 |   0.9359   1.0000   1.0000
          hp |  -0.7784  -0.8548  -0.8548   1.0000
          wt |  -0.8322  -0.8851  -0.8851   0.8645   1.0000
         acc |   0.4233   0.4563   0.4563  -0.6892  -0.4168   1.0000
```

So we will be careful and use the scaled power transformation to transform `mpg`. Now we should check that this transformation solves our nonlinearity problem. First, we will revisit our matrix plot from figure 6. See figure 9.

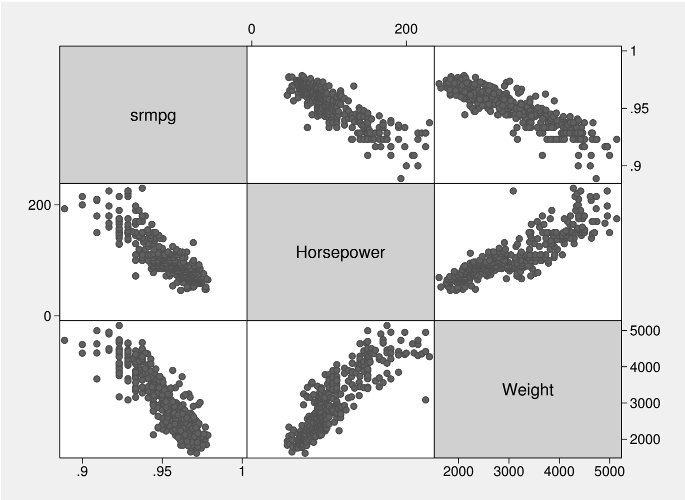


Figure 9. Auto 1993 example, matrix plot revisited

The relationships between `srmpg` (the transformed `mpg`), `hp`, and `wt` are all approximately linear. Our transformation was successful. We will conclude this example with a final graphic that demonstrates how the transformation we found via `irp` affected the fit of the regression of `mpg` on `hp` and `wt`; see figure 10. The first plot shows `mpg` versus its predicted values under initial regression, with no transformation. The second plot shows `mpg` versus its predicted values, after the transformation. Note the change in linearity.

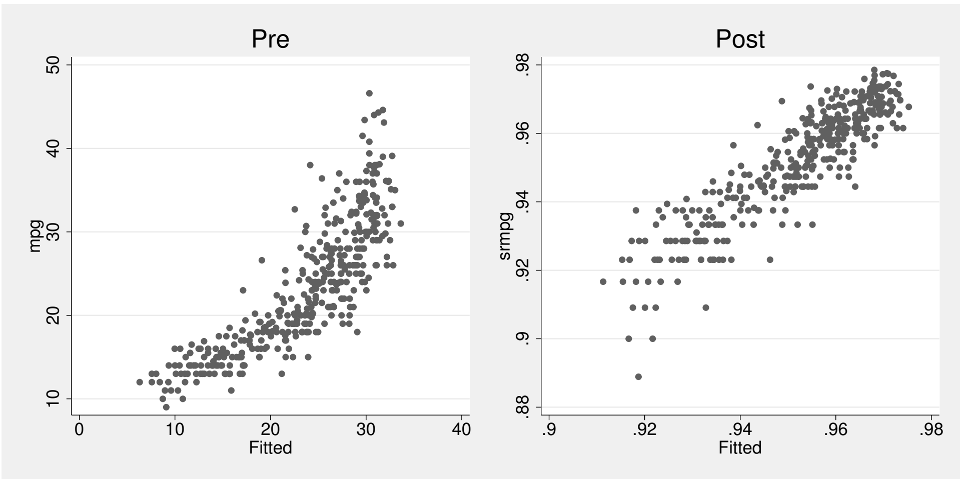


Figure 10. Auto 1993 example, fitted versus response

# 4 Conclusion

We have demonstrated the use of inverse response plots in response transformations to linearity. We used generated and real datasets. Both the theory and the practice of the method was explored.

The `irp` command was fully defined as a method for using inverse response plots in Stata. Its graphical and numeric output were demonstrated, and the process of fitting multiple inverse response plots to the same data was also shown.

# 5 References

Box, G. E. P., and D. R. Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society, Series B* 26: 211–252.

Cook, R. D., and S. Weisberg. 1994. Transforming a response variable for linearity. *Biometrika* 81: 731–737.

Li, K.-C., and N. Duan. 1989. Regression analysis under link violation. *Annals of Statistics* 17: 1009–1052.

UCI Machine Learning Repository. 1993.
http://mlearn.ics.uci.edu/databases/auto-mpg/.

### About the authors

Charles Lindsey is a PhD candidate in statistics at Texas A & M University. His research is currently focused on nonparametric methods for regression and classification. He currently works as a graduate research assistant for the Institute of Science Technology and Public Policy within the Bush School of Government and Public Service. In the summer of 2007, he worked as an intern at StataCorp. Much of the groundwork for this article was formulated there.

Simon Sheather is a professor in and head of the Department of Statistics at Texas A & M University. Simon's research interests are in the fields of flexible regression methods, and nonparametric and robust statistics. In 2001, Simon was named an honorary fellow of the American Statistical Association. Simon is currently listed on http://www.ISIHighlyCited.com among the top one-half of one percent of all mathematical scientists, in terms of citations of his published work.