Paper 430-2012

Transforming Variables for Normality and Linearity – When, How, Why and Why Not's

Steven M. LaLonde, Rochester Institute of Technology, Rochester, NY

ABSTRACT

Power transformations are often suggested as a means to "normalize" univariate data which may be skewed left or right, or as a way to "straighten out" a bivariate curvilinear relationship in a regression model.

This talk will focus on identifying when transformations are appropriate and how to choose the proper transformations using SAS® and new features of the ODS.

There is also a discussion of why, or why not, you may choose the "optimal" transformation identified by SAS.

INTRODUCTION (WHEN AND WHY)

Transformations of variables have been recommended as a solution for asymmetry and for non-linearity for decades. A search of the literature reveals dozens of paper in the last fifty years related to these types of transformations.

The most common transformations are power transformations, and the most common of power transformations are Box-Cox power transformations. Power transformations (Cleveland, 1993) follow the form of:

$$f(x)=\left\{\begin{matrix} \mathrm{x}^{\lambda} & \lambda\neq 0 \\ \ln(\mathrm{x}) & \lambda=0 \end{matrix}\right\},$$

whereas the so-called “Box-Cox” family of power transformations (Box & Cox, 1964) were first described as:

$$f(x)=\left\{\begin{matrix} \dfrac{\mathrm{x}^{\lambda}-1}{\lambda} & \lambda\neq 0 \\ \ln(\mathrm{x}) & \lambda=0 \end{matrix}\right\}$$

In both cases the values of x must be positive in order for the function to be defined everywhere. Furthermore, for the functions to produce reasonable results, x’s must be greater than one. If x is between zero and one, and is raised to a power, then very different things happen to the distribution as opposed to when x is greater than one (i.e. squaring a number between zero and one reduces the value of that number, while squaring a number greater than one increases it’s value).

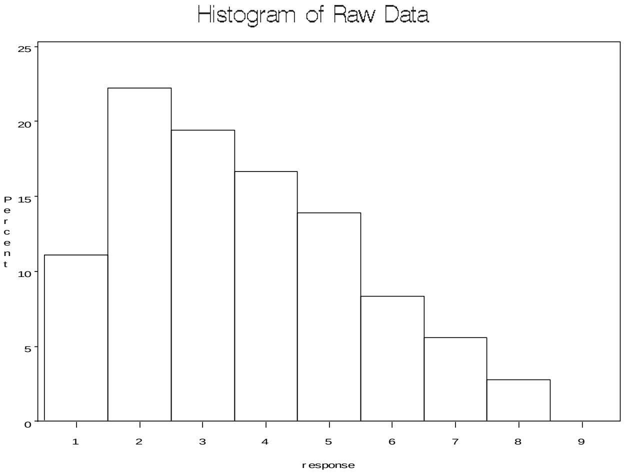
PROC TRANSREG has incorporated mean shift of the variable and a change in spread of the transformed variable in a more general form of the transform:

$$f(x)=\left\{\begin{matrix} \dfrac{(x+c)^{\lambda}-1}{\lambda g} & \lambda\neq 0 \\ \ln(x+c)/g & \lambda=0 \end{matrix}\right\}$$

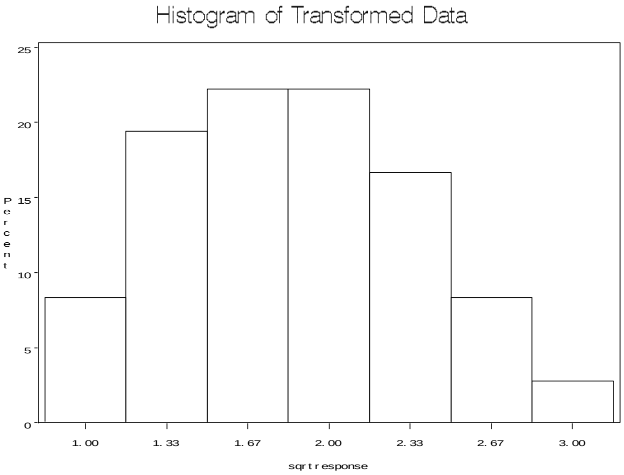
Here c is used to shift the values of x so that it is always greater than zero. The parameter g is used to change the spread of the final variable. By default, c = 0 and g = 1.

The univariate objective is generally to create a transformed variable that is more “normally” distributed. For example, consider the data in Figure 1, which is clearly skewed to the right. A simple power transformation of this variable with λ < 1 will “shrink” the larger values more than the smaller values, resulting in a distribution that is more nearly symmetric, and therefore closer to a normal distribution.

FIGURE 1:

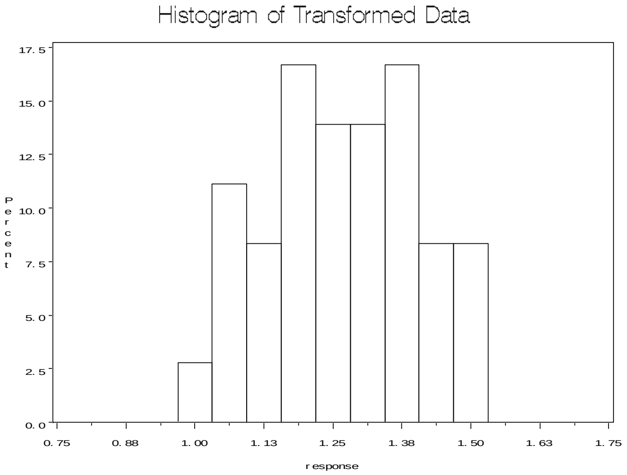


In Figure 2 below, the distribution of the square root of the original variable is plotted. It is clear that this simple transformation has resulted in a variable that is more nearly normal. Statistical tests available in PROC UNIVARIATE or PROC CAPABILITY support this observation.



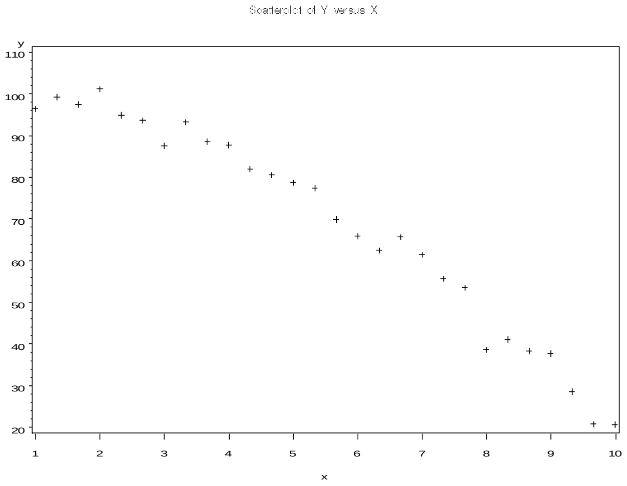
However, if the “optimal” transformation is made, based on the likelihood function described by Box and Cox (1964), then the value of λ is around 0.2. The plot of the original variable raised to the power of 0.2 is shown in Figure 3.

FIGURE 3:



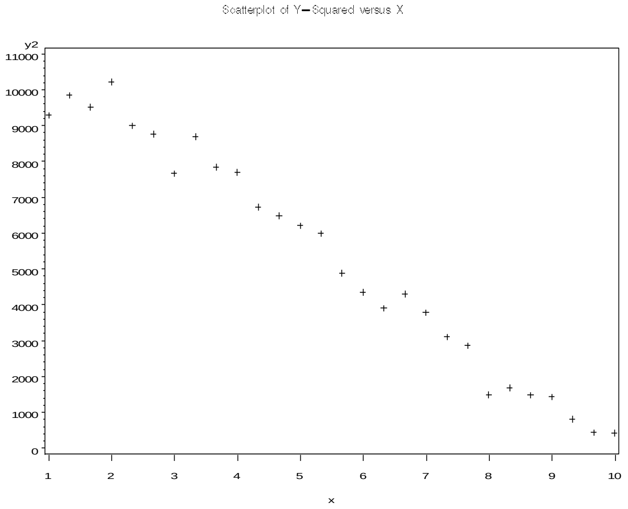
When this idea is applied to the regression problem, the objective becomes to create a transformed dependent or independent variable such that the relationship between the dependent and independent variable(s) is more nearly “linear”. Figure 4 shows a curvilinear relationship between two variables.

FIGURE 4:



In this case, simple squaring the value of the response variable yields the plot in Figure 5. This relationship is much more linear than the original plot. In this case, the “optimal” transformation is also λ = 2.0.

FIGURE 5:



Transformation can also be applied in the context of regression, or general linear models, to “simplify” the model. That is, transforming the dependent, or independent, variable in a regression model can often reduce the complexity of the model required to fit the data. This simplicity is often seen as reducing the degree of the polynomial required to fit a “curve”, as was illustrated in the previous example, or, more subtlety, eliminating the need for interactions between variables a designed experiment.

SAMPLE DATA

For the remainder of the paper the examples are based on car data that will be familiar to many. PROC CONTENTS reveals the following variables:

Alphabetic List of Variables and Attributes

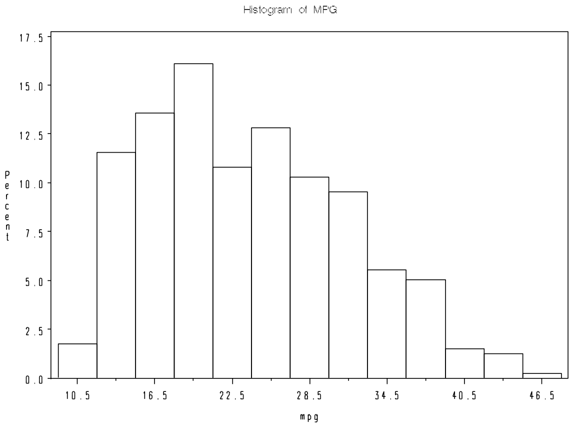
#	Variable	Type	Len	Label
8	acceleration	Num	8	acceleration
4	cylinders	Num	8	cylinders
5	displacement	Num	8	displacement
6	horsepower	Num	8	horsepower
1	make	Char	30	
2	model	Char	30	
9	model_year	Num	8	model_year
3	mpg	Num	8	mpg
10	origin	Num	8	origin
7	weight	Num	8	weight

The response variable in this dataset is mpg, or miles per gallon. Most of the other numeric variables will be considered as potential predictors of mileage. The only variable that requires special explanation is the origin variable, which contains an integer that identifies whether the car originated in the United States, Europe, or Japan. Since that variable does not represent an interval or ratio scale variable, it will not be used in our examples.

USING SAS TO TRANSFORM FOR NORMALITY (HOW)

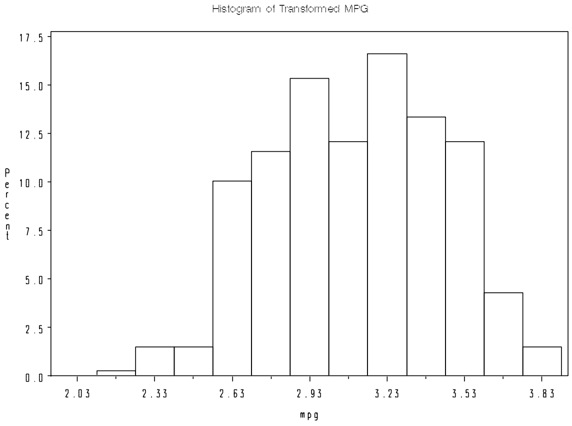
A histogram of the original response variable, mpg, created with PROC CAPABILITY, is shown in Figure 6. It is clear from this histogram that a transformation of mpg with λ < 1 is likely to produce a distribution that is more symmetric.

FIGURE 6:



The solution for the univariate Box Cox transform was presented by Dimakos (SUGI 22, Paper 95) as a IML macro. The macro (%bctrans) searches for the optimal value of λ, transforms the data, and tests the transformed data for the assumption of normality. I was able to get this macro to run in SAS, Version 9.1.3 with only a couple changes. Dimakos's macro, with the minor changes, is included in APPENDIX A. The optimal value for λ identified depends somewhat on what options are chosen with the macro. For instance, if a very fine grid search on λ is done, the optimal value is λ = 0.2. However, less fine grid search results in λ = 0, which corresponds to the log transform. The histogram of the log transformed variable is shown in Figure 7.

FIGURE 7:



It turns out that SAS, PROC IML, is not included in the SAS Learning Edition, which is what many of my students are using, so I had a need to modify the macro to work without PROC IML.

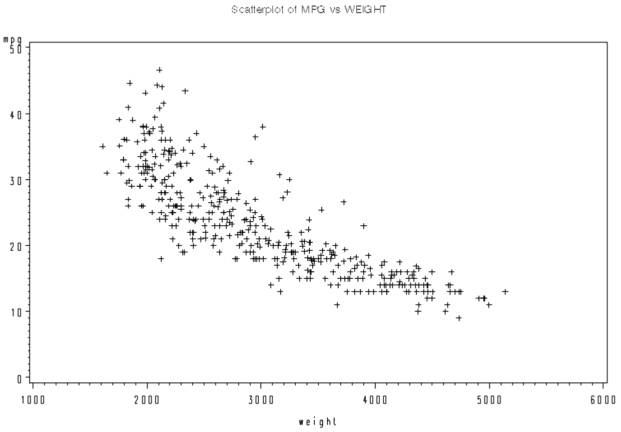
I also wanted to separate out the testing of the fit to normality, so that users with access to SAS/QC® would be able to use PROC CAPABILITY, rather than PROC UNIVARIATE, in order to get better histograms with normal curves overlaid in high resolution graphics. For those interested, I've included that macro (bctrans2) in Appendix B. The revised macro only takes three arguments, the entering SAS dataset, the exiting SAS dataset, and the variable to be "optimally" transformed.

USING SAS TO TRANSFORM FOR LINEARITY (HOW)

SAS has implemented the Box Cox transformation for regression in PROC TRANSREG. In this procedure the optimal λ is chosen, the data is transformed, and the regression model is fit. In this implementation, the transformation is limited to the dependent variable in the model.

In the cars data, suppose that we want to fit a simple linear regression where mpg is the dependent variable, and weight is the independent variable. Figure 8 shows the bivariate scatter plot of mpg versus weight.

FIGURE 8:

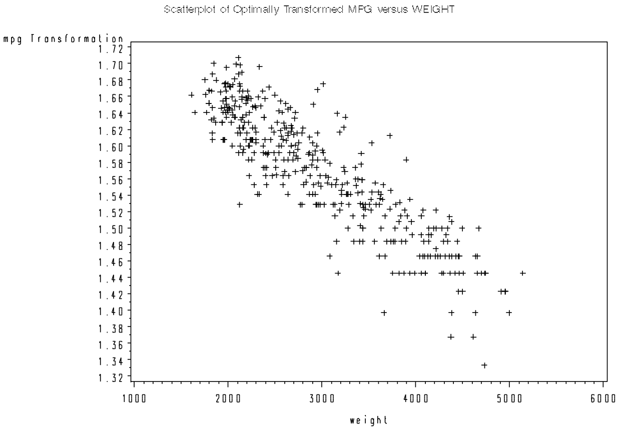


There is clearly a curvilinear relationship between the two variables, and we could choose to add a weight*weight 2nd degree polynomial term into the model which would undoubtedly improve the fit. However, if we want to approach this problem by seeking a transformation of mpg that will result in a "more linear" relationship, we would execute code similar to this:

```
proc transreg details;
model boxcox(mpg/convenient
   lambda=-3 to 3 by .125)=identity(weight);
output out=two;
run;
```

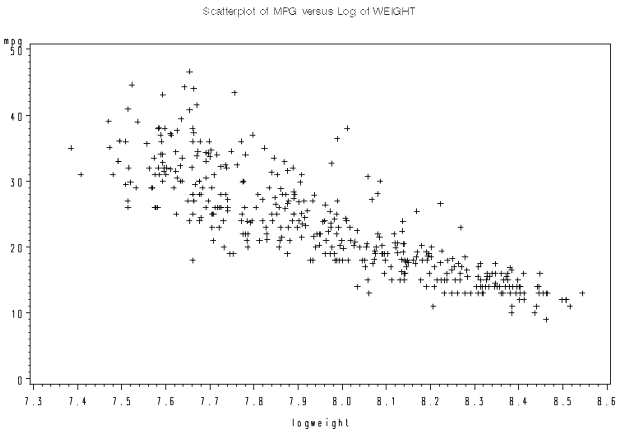
The resulting scatter plot of the relationship between the transformed mpg variable and weight is shown in Figure 9. The actual transformation chosen was λ = -0.5, which corresponds to the reciprocal square root transformation.

FIGURE 9:



This procedure assumes that one transformation of the response variable is will "fix" the model. In this case, taking a log transformation of the weight variable will yield similar results as shown in Figure 10.

FIGURE 10:



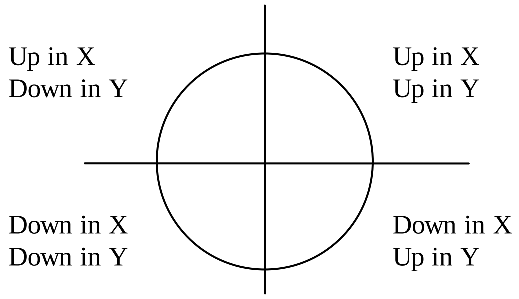
In multiple regressions, the best solution may require transforming more than one independent variable, and that each independent variable may actually require a different transformation.

An improvement to this procedure would be to allow for independent variables to be optimally, and independently, transformed. Of course, this may lead to many relatively equivalent solutions, where going "up" in the power of the response variable is the same as going "down" in the power of the independent variable. This may be why more extensive statistical, or algorithmic solutions, have not been implemented. There may be no substitution of really knowing your data.

KNOWING YOUR DATA (HOW)

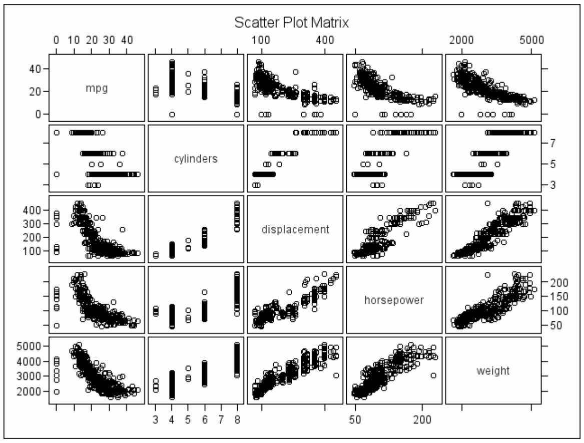
As we have see, if the bivariate relationship between two variables is shown on a scatter plot, then transformation required for linearity becomes more apparent. While "optimal" statistical solutions are useful, it is always useful to understand why a solution is optimal, and what other solutions might be considered. Figure 11 shows how the shape of the curve between the dependent and independent variable can be "straightened" by taking power transformations of either variable.

FIGURE 11:



Some of the new features in ODS GRAPHICS are particularity useful in understanding the relationships between continuous variables. For example, the MATRIX plot in PROC CORR can be used to quickly study the relationship between many variables in one display (Figure 12).

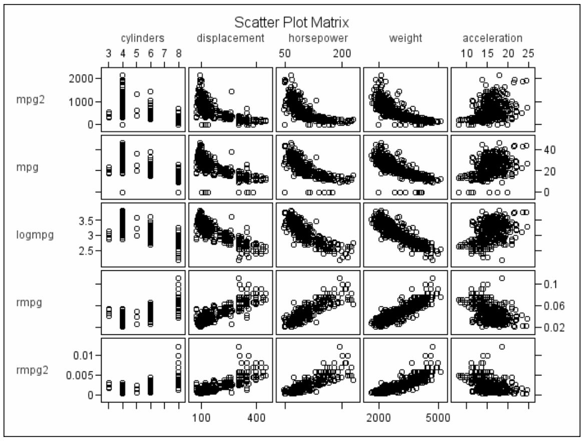
FIGURE 12:



We can see very quickly from this scatter plot matrix not only that the same curvilinear relationship exists between mpg and at least three of the independent variables, but that those three independent variables are very highly correlated (and linearly related). It is apparent from this matrix, that one transformation of the dependent variable may well solve the regression "problem". The alternative would be to transform multiple independent variables, or add numerous polynomial terms for the independent variables. Even still, there will be a significant issue with colinearity among the predictors, leading to inflated standard errors for the regression weights…

In this case, it would make good sense to make use of PROC TRANSREG to find the "optimal" transform of mpg for this model. However, making use of a few data transformations in a DATA STEP, followed by another MATRIX plot, can be used to understand the effect of various power transformations on the dependent variable (see Figure 13).

FIGURE 13:



It is apparent from this plot that the reciprocal of mpg will work quite well for all the independent variables in the model.

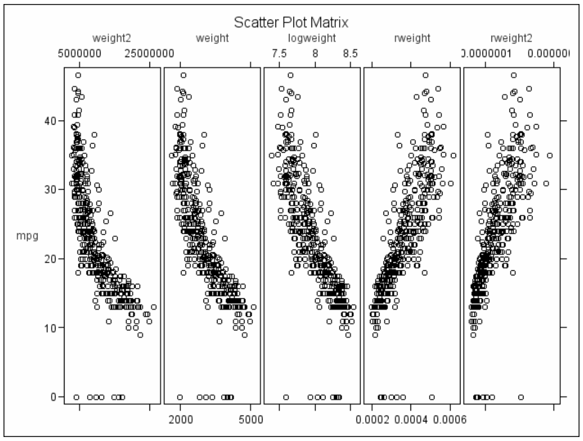
The SAS code to produce these two plots is:

```
proc corr plots=(matrix);
var mpg cylinders displacement horsepower
weight acceleration;
run;

proc corr plots=(matrix);
with mpg2 mpg logmpg rmpg rmpg2;
var cylinders displacement horsepower
    weight acceleration;
run;
```

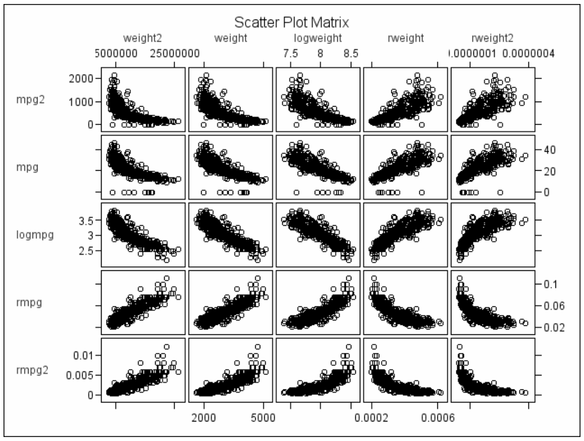
Applying those transformations in the data step to each of the independent variable in a DATA STEP, followed by a MATRIX plot, can be used to understand the effect of various power transformations on each of the independent variables. For example, transformations of the weight variable are shown in Figure 14.

FIGURE 14:



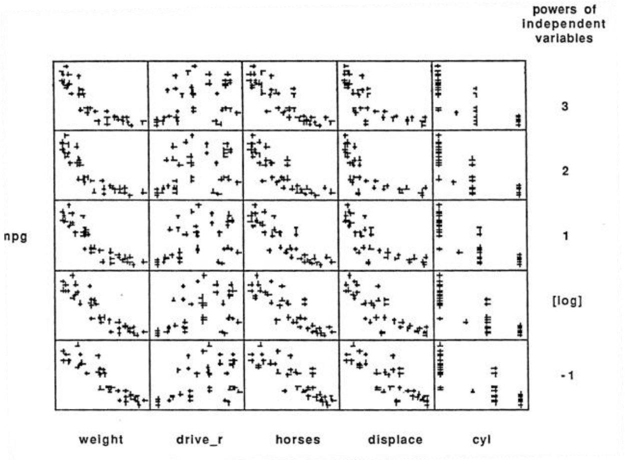
Of course, you could consider transformations of one response variable, along with transformation of one predictor variable in a matrix scatterplot. This plot is shown in Figure 15.

FIGURE 15:



A plot that cannot be obtained with current software would be one that plotted the dependent variable against each of the independent variables, where the row of the matrix was determined by the λ in the power transformation. In this matrix you could see the "optimal" choice of transformation for each independent variable. An example of what the plot might look like is shown in Figure 16. Note that only a subset of the data was used for this plot, and that the independent variables are plotted in a different order.

FIGURE 16:



I've programmed this plot using native PROC IML graphing capabilities, but the program is ready for general distribution.

SHOULD WE TRANSFORM VARIABLES AT ALL? (WHY NOT)

DOWNSIDE OF TRANSFORMATION:

Transformations, in many eyes, complicate the analysis. The complication comes in the form of explaining to the non-statistician why you are modeling the square root of their favorite variable rather than the variable in its unadulterated form. It is easier to "live with" invalid models with strange looking residuals that nobody cares about then it is to explain the complexities of the analysis. That being said, some transformations are easier to explain in some contexts than others. You might prefer the reciprocal transformation to the log transformation in the analysis of car mileage. It is easier to interpret gallons per mile than log of miles per gallon, even though the fit of the model may be similar.

ALTERNATIVE TO TRANSFORMATION: As mentioned before, transformations often reduce the complexity of the model by reducing the need for non-linear modeling, or by reducing the need for specific parameters to be estimated such as interactions or polynomial curvature. Many times such a "complex" model may be preferred over variable transformation. In some cases, a theoretical argument can be made for such models, in which the elimination of the complexity is not only undesirable, but serves to obscure the relationship between the observation and theory.

CONCLUSIONS

Transformations have their place in modern data analysis and modeling. Tools in SAS can be used in fitting and evaluating power transformations for normality and linearity. Suggestions for improving these tools have been provided.

REFERENCES:

"Power Transformations Using SAS/IML ® Software", by Ioannis C. Dimakos, SUGI 22, Paper 95.

"An Analysis of Transformations", by Box, G.E.P. & Cox, D.R. *Journal of the Royal Statistical Society, Series B*, 1964, 26, 211-252.

Visualizing Data, by Cleveland, William, S., 1993, AT&T, Hobart Press, Summit, NJ.

SAS/STAT® 9.1 User's Guide, SAS Institute, Cary, North Carolina.

ACKNOWLEDGMENTS

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Steven M. LaLonde, Ph.D.
John D. Hromi Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive
Rochester, New York 14623-5604

Work Phone: (585) 475-5854
Fax: (585) 475-5959
Email: smleqa@rit.edu

* *

APPENDIX A:

```
%macro bctrans(data=,out=,var=,r=,min=,max=,step=);

proc iml;
 use &data;
 read all var {&var} into x;
 n=nrow(x); one=j(n,1,1); mat=j(&r,2,0);
 lnx=log(x); sumlog=sum(lnx);
 start; i=0;
 do lam=&min to &max by &step;
    i=i+1;
    lambda=round(lam,.01);
    if lambda = 0 then xl=log(x);
    else xl=((x##lambda) - one)/lambda;
    mean=xl[:];
    d=xl-mean;
    ss=ssq(d)/n;
    l=-.5*n*log(ss)+((lambda-1)*sumlog);
    mat[i,1] = lambda;
    mat[i,2] = l;
 end;
 finish; run;
 print "Lambdas and their l(lambda) values",
 mat[format=8.3];
 create lambdas from mat;
 append from mat;
 quit;

data lambdas;
set lambdas;
 rename col1=lambda col2=l; run;

proc plot data=lambdas nolegend;
 plot l*lambda;
 title 'lambda vs. l(lambda) values'; run; quit;

proc sort data=lambdas;
 by descending l; run;

data &out;
set lambdas;
 if _n_>1 then delete; run;

proc print data=&out;
 title 'Highest lambda and l(lambda) value'; run;

proc iml;
 use &data;
 read all var {&var} into old;
 use &out;
 read all var {lambda l} into power;
 lambda=power[1,1];
 if lambda=0 then new=log(old);
 else new=old##lambda;
 create final from new;
 append from new; quit;

data final;
set final;
 rename col1=&var; run;

proc univariate normal plot data=final;
title 'Normality Assessment for';
title2 'Power-Transformed Variable'; run;
%mend bctrans;
```

APPENDIX B:

```
%macro bctrans2(data=_last_, out=_out_, var=y);

data keeplog;
 loglike = .; l = .; output; stop; run;

%do lbig = -20 %to 20;
data _new;
 set &data nobs=obs;
 n = obs;
 l = &lbig./10;
 x = &var;
 if l ne 0 then xt = ((x**l)-1)/l;
 else xt = log(x);
run;

proc means mean noprint;
 var xt; output out=_mean1 mean=mxt;

data _gmean1;
 set _mean1; call symput('gmean',mxt);
run;

data _new2;
 set _new; retain mxt;
 if _n_ = 1 then set _mean1;
 term1 = (xt - mxt)**2;
 term2 = log(x);
run;

proc means sum noprint;
 var term1 term2; output out=_sum1 sum=sterm1 sterm2;
run;

data loglike;
 set _sum1; set _new(obs=1);
 loglike = -1*(n/2)*log((1/n)*sterm1)+(l-1)*sterm2;
run;

data keeplog;
 set keeplog loglike; keep loglike l;
run;
%end;

proc means max data=keeplog idmin noprint;
 var loglike;
 output out=_max max=maxloglike;

data keeplogmax;
 set keeplog;
 keep l maxloglike;
 retain maxloglike;
 if _n_ = 1 then set _max;
 if loglike = maxloglike then output;
run;

proc print data=keeplog;
title 'Log Likelihoods and Values of Lambda';

proc print data=keeplogmax;
title 'Optimal Value of Lambda';

data _null_;
 set keeplogmax;
 if _n_ = 1 then call symput('l',l);
 stop;
run;

data &out;
 set &data;
 if &l ne 0 then t_&var. = ((&var.**&l)-1)/&l;
 else t_&var. = log(&var.);
run;
%mend bctrans2;
```