8-1-1996

Double-power transformations to analyze data

Erika Sato

DOUBLE-POWER TRANSFORMATIONS TO ANALYZE MIXTURE DATA

by

Erika M. Sato

A Thesis Submitted

in

Partial Fulfillment

of the

Requirements for the Degree of

MASTER OF SCIENCE

in

Applied and Mathematical Statistics

Approved by:

Prof. J. O. Voelkel
(Thesis Advisor)

Prof. Daniel R. Lawrence

Prof. Hubert N. Wood

Prof. Donald D. Baker
(Department Head)

GRADUATE STATISTICS
COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
AUGUST, 1996

authorize or **deny** permission to reproduce an RIT thesis.

1. Title of thesis DOUBLE-POWER TRANSFORMATIONS TO ANALYZE MIXTURE DATA

I ERIKA M. SATO hereby **grant permission** to the Wallace Memorial Library of RIT to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date AUGUST 2, 1996

2. Title of thesis _____

_____

I _____ **prefer to be contacted** each time a request for reproduction is made. I can be reached at the following address.

_____

_____

Date _____

3. Title of Thesis _____

_____

I _____ hereby **deny** permission to the Wallace Memorial Library of RIT to reproduce my thesis in whole or in part.

Date _____

rev. 1/89

Abstract

Power transformations are commonly used in order to fit simpler and/or more appropriate models to data. These transformations are well-known and well-documented for cases where the predictor variables are not linearly constrained, unlike mixture experiments. In the case of mixture designs, however, for which linear constraints do exist, several linear models proposed in recent literature fall into a power transformation family; this suggests that similar transformations might be useful for mixture experiments, as well. The log-likelihood function for λ and γ, transformations on the response and predictor variables, was derived for the mixture case where the predictor variables are linearly constrained and was maximized using a specially-written SAS program. To test the effectiveness of this procedure, simulations were done for two different designs and for four different combinations of λ and γ. It was found that the 95% confidence region about $\hat{\lambda}$ and $\hat{\gamma}$ captured the true values of λ and γ approximately 90% of the time, regardless of the nature of the design or of the transformation. This procedure appeared to be able to discriminate between the different transformations on the response better than on the predictor variables, particularly when the correct transformation was the log-transformation (i.e., when $\gamma = 0$). This could be due in part to the fact that the ranges of the predictors chosen was simply not large enough given the amount of replication used.

TABLE OF CONTENTS

Introduction

Power Transformations of Variables

Transformations are commonly used in linear regression to better ensure that the assumption of constant variance is met. There are well-known transformations on the response variable to account for the situation where the variance is a function of the mean. Transformations on the predictors and/or response variable are also commonly used in order to simplify the model expressing the relationship between the predictors and the response, and, in some cases, could also lead to a model that sheds more light on their true relationship. For example, a straight-line fit of ln(Y) versus X might be simpler and perhaps more enlightening than a quadratic fit of Y versus X to the same data.

A useful family of transformations is the power transformation family. Tukey[1] suggested

$$Y_i = \begin{cases} y_i^{\lambda}, & \gamma \neq 0 \\ \ln(y_i), & \gamma = 0, \end{cases}$$

while Box and Cox[2] used the equivalent

$$Y_i = \begin{cases} \dfrac{\left(y_i^{\gamma} - 1\right)}{\gamma}, & \gamma \neq 0 \\ \ln(y_i), & \gamma = 0 \end{cases}$$

to transform the response variable. Box and Tidwell[3] suggested a transformation of each predictor variable of the form

$$X_j = \begin{cases} x_j^{\gamma_j}, & \gamma_j \neq 0 \\ \ln(x_j), & \gamma_j = 0. \end{cases}$$

(It is assumed that the reader has not only a basic understanding of power transformations but also some experience with the design and analysis of mixture experiments.)

Some Classes of Mixture Models

These transformations are all well-known and thoroughly discussed for cases where the predictor variables are not linearly constrained. In the case of mixture experiments, this is not the case. In a mixture experiment, the *X*'s commonly represent volume or weight fractions of the components that make up the mixture. These X's are always non-negative and are constrained in such a way that their sum is equal to 1. To increase the fraction of one component, then, the fraction of one or more of the remaining components must decrease accordingly. The inclusion of this constraint adds additional complexity to the mixture situation that will be addressed in this paper.

A simple linear polynomial model which could be used to represent a q-component mixture system is

$$Y_i = \beta_\circ + \sum_{j=1}^{q} \beta_j x_{ij} + \varepsilon_{ij}, \qquad i = 1, 2, \ldots, n,$$

where n represents the number of observations. Because of the linear constraints implicit in a mixture system, the **X'X** matrix for this model will be singular unless a constraint is applied. Of the many possible constraints that could be used, Scheffé[4] chose the constraint $\beta_\circ = 0$, leading to the Scheffé polynomial model

$$Y_i = \sum_{j=1}^{q} \beta_j x_{ij} + \varepsilon_{ij} \ .$$

Aitchison and Bacon-Shone[5] proposed log-contrast models, which offer advantages over polynomial models in testing hypotheses of so-called "inactivity" and "additivity" of one or more components. These models have the form

$$Y_i = \beta_\circ + \sum_{j=1}^{q-1} \beta_j \ln(x_{ij} / x_{iq}) + \varepsilon_{ij} \ ,$$

which can be rewritten as

$$Y_i = \beta_\circ + \sum_{j=1}^{q} \beta_j \ln x_{ij} + \varepsilon_{ij} \ ,$$

where $\beta_q = -\sum_{j=1}^{q-1} \beta_j$. This, then, results in the constraint $\sum_{j=1}^{q} \beta_j = 0$.

Draper and St. John[6] and Cornell[7] suggested the inclusion of one or more inverse terms, x^{-1}, into the Scheffé model in cases involving a steep cliff in the response as the amount of a component goes to zero. One could, however, also envision a "pure" inverse model which contains only inverse terms, i.e.,

$$Y_i = \beta_\circ + \sum_{j=1}^{q} \beta_j x_{ij}^{-1} + \varepsilon_{ij} \ .$$

This is the third class of models that has been introduced to this point. To summarize, the three are as follows:

Model		Constraint
Linear Polynomial Model:	$Y_i = \beta_\circ + \sum_{j=1}^{q} \beta_j x_{ij} + \varepsilon_{ij}$	many possible e.g., $\beta_\circ = 0$ (Scheffé) e.g., $\sum_{j=1}^{q} \beta_j = 0$
Log-contrast:	$Y_i = \beta_\circ + \sum_{j=1}^{q} \beta_j \ln x_{ij} + \varepsilon_{ij}$	$\sum_{j=1}^{q} \beta_j = 0$
Pure Inverse:	$Y_i = \beta_\circ + \sum_{j=1}^{q} \beta_j x_{ij}^{-1} + \varepsilon_{ij}$	?

For a pure inverse model, a constraint is not mathematically required in order to solve the regression equations, but because it is still a mixture system, a constraint should be considered here, as well. The log-contrast model uses the constraint $\sum_{j=1}^{q} \beta_j = 0$, and in the case of the Scheffé model, the constraint is $\beta_\circ = 0$. For a simple polynomial model, another constraint, such as the one used with the log-contrast model, could also be used.

In order to maintain internal consistency, the constraint $\sum_{j=1}^{q} \beta_j = 0$ is used for all models considered in this thesis.

Transformation of the Predictor and Response Variables

The three models discussed above are members of a power transformation family, which suggests that similar transformations might be useful for mixture experiments, as well. The three "classes" of mixture models can be obtained by a transformation of the form

$$X_i^{(\gamma)} = \begin{cases} \dfrac{x_i^{\gamma} - 1}{\gamma}, & \gamma \neq 0, \\ \ln(x_i), & \gamma = 0, \end{cases} \qquad i = 1, 2, \ldots, q.$$

A linear polynomial model would be obtained when $\gamma = 1$, a log-contrast model when $\gamma = 0$, and the pure inverse model when $\gamma = -1$. An algorithm for determining γ, then, would allow the data to better dictate which model would best describe the system. The form of the above transformation, as suggested by Box and Cox[2], was chosen so that it would be continuous at $\gamma = 0$. While it is possible to envision a different transformation parameter for each component, such as in Box and Tidwell's[3] proposal, we address only the case where the same parameter, γ, is assumed for all the components.

To broaden the class of models, we propose, in addition to transformations on the predictor variables, power transformations on the response variable of the same form, namely,

$$Y^{(\lambda)} = \begin{cases} \dfrac{y^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\ \ln y, & \lambda = 0. \end{cases}$$

Thus, we have the model

$$Y_i^{(\lambda)} = \beta_{\circ} + \sum_j \beta_j x_{ij}^{(\gamma)} + \varepsilon_{ij}, \qquad \sum_{j=1}^{q} \beta_j = 0,$$

where $Y^{(\lambda)}$ and $x^{(\gamma)}$ represent the transformed variables. The choice of λ and γ, then, could help us better define the most appropriate model for our data.

Estimation of Parameters, λ and γ

Assume that:

1. Y_i, $i = 1,2, ..., n$, are independent random variables.
2. The predictor variables, $x_{1i}, ..., x_{qi}$, $i = 1,2,..., n$, are fixed.
3. There exists a λ and γ such that
    a. The $Y_i^{(\lambda)}$, $i = 1,2, ..., n$, are normally distributed with constant variance σ^2.
    b. $E[\boldsymbol{Y}^{(\lambda)}] = \boldsymbol{X}^{(\gamma)}\beta$, for some vector β.

The maximum-likelihood estimators for λ and γ can be obtained via a procedure analogous to that used by Box and Cox[2]. For a given λ and γ, the joint log-likelihood, within a constant determined by n, the number of observations in the experiment, is

$$\ln L(\lambda,\gamma) = -\frac{1}{2} n \ln\left(\frac{SS_{resid}}{n}\right) + (\lambda - 1)\sum_{i=1}^{n} \ln y_i$$

where SS_{resid} is simply the residual sum of squares from regressing the transformed response, $y^{(\lambda)}$, on the transformed predictors, $x^{(\gamma)}$, using ordinary least squares regression. This derivation is shown in Appendix A.

This function is then maximized with respect to λ and γ to obtain $\hat{\lambda}$ and $\hat{\gamma}$. For this research, a grid-search algorithm was used to find the approximate $\hat{\lambda}$ and $\hat{\gamma}$ that maximize the log-likelihood function for a given data set. A program was written to calculate the log-likelihood for every combination of λ and γ on the grid, and the maximum was then selected from the calculated log-likelihoods. This approximately maximized the log-likelihood function with respect to λ and γ, as well as with respect to σ and β. The program, which was written using SAS code, appears in Appendix C.

An approximate asymptotic 100(1 - α)% confidence region for λ and γ can be constructed from

$$\ln L_{\max}(\hat{\lambda},\hat{\gamma}) - \ln L_{\max}(\lambda,\gamma) < \frac{1}{2}\chi^2_{\upsilon}(\alpha)$$

(see Box and Cox[2]), where L_{max} is the joint likelihood function $L(\beta,\sigma,\lambda,\gamma)$, with maximum likelihood estimates substituted in for all parameters that are not explicitly stated in the function. For example, $L_{max}(\lambda,\gamma) = L(\hat{\beta},\hat{\sigma},\lambda,\gamma)$ and $L_{max}(\lambda) = L(\hat{\beta},\hat{\sigma},\lambda,\hat{\gamma})$. For a joint confidence region, $\nu = 2$ (based on the two estimated parameters λ and γ). For $\alpha = 0.05$, with a critical χ^2_2-value of 5.991, then, all values of λ and γ which satisfy the inequality $\ln L_{\max}(\hat{\lambda},\hat{\gamma}) - \ln L_{\max}(\lambda,\gamma) < 3$ define an asymptotic 95% confidence region about λ and γ. Similarly, a univariate 95% confidence interval for λ, for example, could be constructed from $\ln L_{\max}(\hat{\lambda}) - \ln L_{\max}(\lambda) < 1.92$ ($\nu = 1$, χ^2_1-value of 3.84).

Simulations

Simulations were done for eight different cases. Data were generated for each combination of $\lambda = 0, 1$ and $\gamma = 0, 1$, with two different designs. In each case, random error was introduced to produce, on average, an R^2_{adj} of 0.95 with the correct λ and γ (see Appendix B). The program then simulated 1000 occurrences of each combination of the following: $\lambda = 0, 1$, $\gamma = 0, 1$, and $R^2_{adj} = 0.95$ for the two different designs. For each case, the grid for λ and γ covered a range of ± 1 about the "true" values of λ and γ, in steps of 0.1.

To cover designs which might be commonly used in practice, a 3-component design and a 6-component design were chosen, both with only lower bounds on the component fractions (see Tables 1 and 2). Lower bounds of 0.1 were selected for each component in both designs, so therefore, the ranges of all the components in the 3-component case run from 0.1 to 0.8, and in the 6-component case from 0.1 to 0.5. Since only lower bounds were used for each component, the space covered still represented a simplex. The 3-component design was completely replicated with points at the vertices, edge midpoints and overall centroid (14 points), while the 6-component design had replication only at the vertices and overall centroid (30 points).

To compare the effectiveness of this procedure among the eight cases, the coefficients for each model were selected to produce responses with the same magnitude and spread for each case. In order to exhibit the nonlinearity of the log-transformation, the data were generated by selecting the coefficients in such a way that the expected value of the response at the points run in the design varied between 1 and 5. The choice of the coefficients used was subject to the additional constraint that $\sum_{i=1}^{q} \beta_i = 0$. Stipulating that the coefficients on the X's followed $2x_1 = 2x_2 = -x_3$ for the 3-component model and $x_1 = x_2 = x_3 = -x_4 = -x_5 = -x_6$ for the 6-component model accommodated this constraint. The models used for the 3-component design were:

$$y = 3.6667 + 1.9047 * x_1 + 1.9047 * x_2 - 3.8095 * x_3$$

$$\ln y = 1.0732 + 0.766 * x_1 + 0.766 * x_2 - 1.532 * x_3$$

$$y = 3.3211 + 0.5581 * \ln x_1 + 0.5581 * \ln x_2 - 1.1162 * \ln x_3$$

$$\ln y = 0.9340 + 0.2246 * \ln x_1 + 0.2246 * \ln x_2 - 0.4491 * \ln x_3,$$

and for the 6-component design:

$$y = 3.000 + 5.000 * x_1 + 5.000 * x_2 + 5.000 * x_3 - 5.000 * x_4 - 5.000 * x_5 - 5.000 * x_6$$

$$\ln y = 0.805 + 2.012 * x_1 + 2.012 * x_2 + 2.012 * x_3 - 2.012 * x_4 - 2.012 * x_5 - 2.012 * x_6$$

$$y = 3.000 + 0.910 * \ln x_1 + 0.910 * \ln x_2 + 0.910 * \ln x_3 - 0.910 * \ln x_4$$
$$- 0.910 * \ln x_5 - 0.910 * \ln x_6$$

$$\ln y = 0.805 + 0.366 * \ln x_1 + 0.366 * \ln x_2 + 0.366 * \ln x_3 - 0.366 * \ln x_4$$
$$- 0.366 * \ln x_5 - 0.366 * \ln x_6.$$

Contour plots of relative log-likelihood versus λ and γ were generated for some of the simulations. These contours represent deviations from the maximum log-likelihood, which was set to zero. The "cutoff" of 3 units, representing a 95% confidence region about $(\hat{\lambda}, \hat{\gamma})$, is represented by a solid line on the contour plots (see an example in Figure 1). For each simulation, then, it is possible to check whether the "true" value of λ and γ were covered by that 95% confidence region by checking whether

$\ln L_{\max}(\hat{\lambda},\hat{\gamma}) - \ln L_{\max}(\lambda_{true},\gamma_{true}) < 3$. $\hat{\lambda}$ and $\hat{\gamma}$ are the maximum likelihood estimates obtained for that given data set, and λ_{true} and γ_{true} are the "true" values which were used to generate the data. This is equivalent to doing an asymptotic hypothesis test on $(\hat{\lambda},\hat{\gamma})$. The frequency with which the "true" value of λ and γ was covered by the 95% confidence region is tabulated in Table 3 for all eight cases. Histograms of the maximum likelihood estimates of λ and γ are plotted for each case in Figure 2, while Table 4 shows the sample mean and standard deviation of the maximum-likelihood estimates. It should be noted that the standard deviations calculated here are not adjusted for the boundary problems encountered in several of the simulations, so the actual standard deviations are very likely larger than what has been reported for those cases.

Results of Simulations

The 95% confidence region about $\hat{\lambda}$ and $\hat{\gamma}$ captured the "true" λ and γ approximately 90% of the time (see Table 3), regardless of the nature of the model or of the design. However, the width of the histograms shown in Figure 2 are tighter for λ than γ (see also Table 4), indicating that this procedure is able to discriminate between the transformations

on the response, λ, better than on the predictors, γ. In particular, the procedure has trouble identifying which transformation to use on the predictor variables when the correct transformation is $\gamma = 0$.

Contour plots of the log-likelihood function with respect to λ and γ, examples of which are shown in Figures 1 and 3, indicate that the log-likelihood changes faster with changes in λ than in γ. This could be due to the way that λ and γ enter into the log-likelihood function. γ influences the log-likelihood only by affecting the predictor variables used in the regression; λ not only affects the response variables used in the regression, but also enters directly into the log-likelihood function. Scatterplots of $\hat{\lambda}$ versus $\hat{\gamma}$ for the 3-component models (see Figure 4) indicate that there is a correlation between $\hat{\lambda}$ and $\hat{\gamma}$ which is largely absent in the 6-component case (see Figure 5). Why this occurs is not known. Correlation coefficients are tabulated in Table 4.

Even though the range of the predictors selected for the simulations was maximized (subject to the restriction of $x_i \geq 0.1$, for $i = 1,2, \ldots, q$, and the same range for all the components), it is possible that the range of the predictors was still too small to readily pick up the log-transformation without more replication. In cases where the number of replicates was increased or the noise in the data was reduced (smaller σ, higher R^2), the width of the histograms was reduced. The extra number of points may also be a

contributing factor to the better behavior in the 6-component case. This might suggest that, in general, the value of γ should be selected by the experimenter according to preference or prior knowledge, rather than have the value be determined by the data.

Summary and Conclusions

The linear polynomial model, the log-contrast model and the pure inverse model are part of a power family. An extension of the Box-Cox transformations was made to include the predictor variables, as well as the response variables, in a mixture situation where the X's are linearly constrained. The choices of λ and γ in the model

$$Y_i^{(\lambda)} = \beta_o + \sum_j \beta_j x_{ij}^{(\gamma)} + \varepsilon_{ij}, \qquad \sum_{j=1}^{q} \beta_j = 0,$$

where $Y^{(\lambda)}$ and $x^{(\gamma)}$ represent the transformed variables, can help us better define the most appropriate model to fit our data.

A specially written grid-search algorithm located the maximum-likelihood estimates for the transformations on the response, λ, and the predictors, γ. Simulations with different models and designs show that the "true" λ and γ are covered by a 95% confidence interval about $\hat{\lambda}$ and $\hat{\gamma}$ approximately 90% of the time. However, the algorithm seems to be able to locate $\hat{\lambda}$ more readily than $\hat{\gamma}$, especially when the true transformation on the predictor

variables is the log-transformation (i.e., when $\gamma = 0$). Decreasing the noise in the data and/or using many replicates improves the discrimination of this method for the different transformations.

Future Work

Extensions of this investigation to include second-order models or individual transformations on each component would produce a procedure with additional practical application. It might also be useful to investigate the small-sample properties of $\hat{\lambda}$ and $\hat{\gamma}$ and to study the covariance structure of the parameter estimates. Future research suggested by the results of the simulations includes an investigation of the factors affecting detection of the log-transformation. The influence of the design (how many replicates and which points are chosen) and the effect of the model coefficients resulting from the design and the response range could be better understood. In addition, extensions to include hypothesis testing of the estimates or to OC-curves for sample designs would be helpful if discriminating between the models on a mechanistic level was of interest.

References

1. Tukey, J. W. (1957), "On the Comparative Anatomy of Transformations"; *Annals of Mathematical Statistics*, 28, pp. 602-632.

2. Box, G. E. P. and Cox, D. R. (1964), "An Analysis of Transformations"; *Journal of the Royal Statistical Society, Series B*, 26, pp. 211-243.

3. Box, G. E. P. and Tidwell, P. W. (1962), "Transformation of the Independent Variables"; *Technometrics* 4, pp. 531-550.

4. Scheffé, H. (1958), "Experiments with Mixtures"; *Journal of the Royal Statistical Society, Series B* 20, pp. 340-360.

5. Aitchison, J. and Bacon-Shone, J. (1984), "Log Contrast Models for Experiments with Mixtures"; *Biometrika* 71, pp. 323-330.

6. Draper, N. R. and St. John, R. C. (1977), "A Mixtures Model with Inverse Terms"; *Technometrics* 19, pp. 37-46.

7. Cornell, J. A. (1990), *Experiments with Mixtures*, 2nd. ed.; John Wiley & Sons, New York, NY.

Table 1. Replicated 3-Component Mixture Design

Points	x1	x2	x3	replicates
Extreme Vertices				
1	0.1	0.1	0.8	2
2	0.1	0.8	0.1	2
3	0.8	0.1	0.1	2
Edge Midpoints				
4	0.1	0.45	0.45	2
5	0.45	0.1	0.45	2
6	0.45	0.45	0.1	2
Overall Centroid				
7	0.33	0.33	0.34	2

Table 2. Replicated 6-Component Mixture Design

Points	x1	x2	x3	x4	x5	x6	replicates
Extreme Vertices							
1	0.1	0.1	0.1	0.1	0.1	0.5	2
2	0.1	0.1	0.1	0.1	0.5	0.1	2
3	0.1	0.1	0.1	0.5	0.1	0.1	2
4	0.1	0.1	0.5	0.1	0.1	0.1	2
5	0.1	0.5	0.1	0.1	0.1	0.1	2
6	0.5	0.1	0.1	0.1	0.1	0.1	
Edge Midpoints							
7	0.1	0.1	0.1	0.1	0.3	0.3	1
8	0.1	0.1	0.1	0.3	0.1	0.3	1
9	0.1	0.1	0.1	0.3	0.3	0.1	1
10	0.1	0.1	0.3	0.1	0.1	0.3	1
11	0.1	0.1	0.3	0.1	0.3	0.1	1
12	0.1	0.1	0.3	0.3	0.1	0.1	1
13	0.1	0.3	0.1	0.1	0.1	0.3	1
14	0.1	0.3	0.1	0.1	0.3	0.1	1
15	0.1	0.3	0.1	0.3	0.1	0.1	1
16	0.1	0.3	0.3	0.1	0.1	0.1	1
17	0.3	0.1	0.1	0.1	0.1	0.3	1
18	0.3	0.1	0.1	0.1	0.3	0.1	1
19	0.3	0.1	0.1	0.3	0.1	0.1	1
20	0.3	0.1	0.3	0.1	0.1	0.1	1
21	0.3	0.3	0.1	0.1	0.1	0.1	1
Overall Centroid							
22	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	3

Table 3. Percentage of times the true λ and γ were covered by a 95% CI around the calculated maximum for a univariate CI about $\hat{\lambda}$ and $\hat{\gamma}$ and a joint confidence region about $\hat{\lambda}$ and $\hat{\gamma}$.

		$\hat{\lambda}$	$\hat{\gamma}$	$(\hat{\lambda},\hat{\gamma})$
3-Component Model				
	$\lambda = 1$ $\gamma = 1$	89.1	87.6	85.3
	$\lambda = 1$ $\gamma = 0$	90.4	90.2	87.5
	$\lambda = 0$ $\gamma = 1$	91.0	88.6	88.7
	$\lambda = 0$ $\gamma = 0$	89.4	88.6	86.3
6-Component Model				
	$\lambda = 1$ $\gamma = 1$	91.1	89.7	89.6
	$\lambda = 1$ $\gamma = 0$	90.5	89.8	87.1
	$\lambda = 0$ $\gamma = 1$	91.3	89.5	89.0
	$\lambda = 0$ $\gamma = 0$	92.8	89.9	90.2

Table 4. Mean (deviated from true value) and Standard Deviation of $\hat{\lambda}$ and $\hat{\gamma}$ obtained from each case; correlation coefficient for $\hat{\lambda}$ and $\hat{\gamma}$.

		$\hat{\lambda}$	$\hat{\gamma}$	$\rho_{\hat{\lambda},\hat{\gamma}}$
3-Component Model				
	$\lambda = 1$ $\gamma = 1$	-0.02 ± 0.26	-0.02 ± 0.41	-0.506
	$\lambda = 1$ $\gamma = 0$	-0.05 ± 0.32	-0.08 ± 0.66	-0.578
	$\lambda = 0$ $\gamma = 1$	+0.01 ± 0.22	+0.02 ± .39	-0.508
	$\lambda = 0$ $\gamma = 0$	-0.001 ± 0.30	-0.13 ± 0.66	-0.566
6-Component Model				
	$\lambda = 1$ $\gamma = 1$	-0.015 ± .14	-0.007 ± 0.23	0.003
	$\lambda = 1$ $\gamma = 0$	-0.018 ± 0.17	-0.016 ± 0.36	0.068
	$\lambda = 0$ $\gamma = 1$	-0.002 ± 0.13	-.001 ± 0.23	-0.046
	$\lambda = 0$ $\gamma = 0$	0.011 ± 0.15	-0.018 ± 0.36	0.015

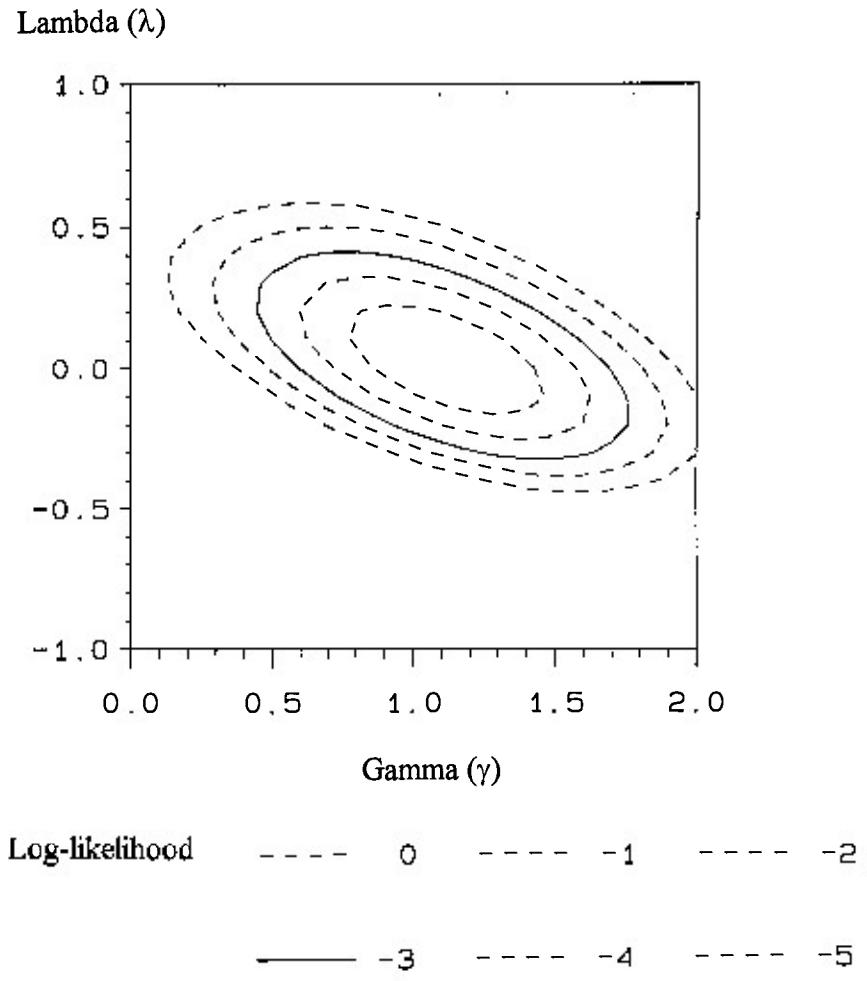

Figure 1. Example of contour plots of relative log-likelihood vs. λ and γ.

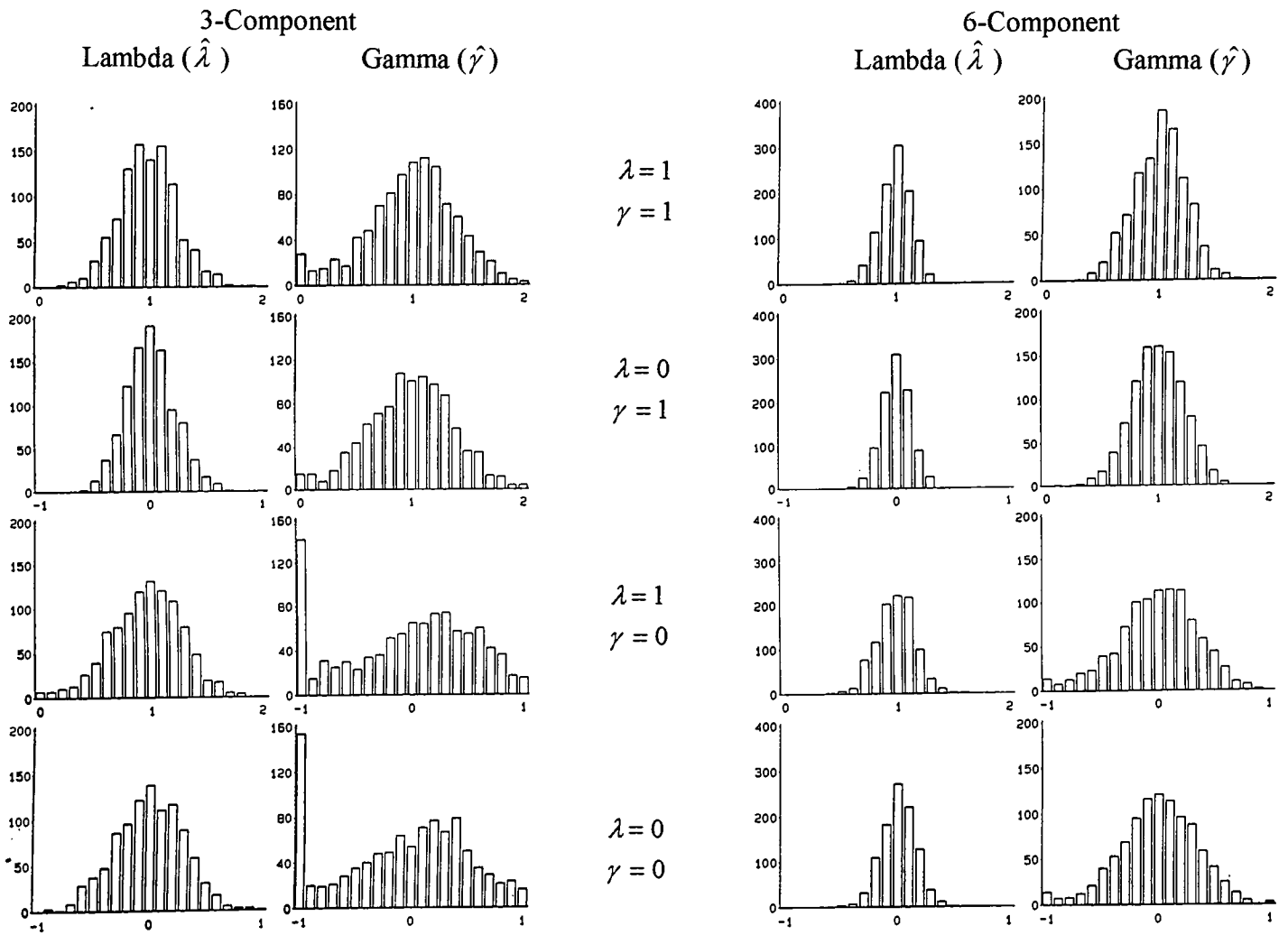

Figure 2. Histograms of $\hat{\lambda}$ and $\hat{\gamma}$ for simulations using the various models.

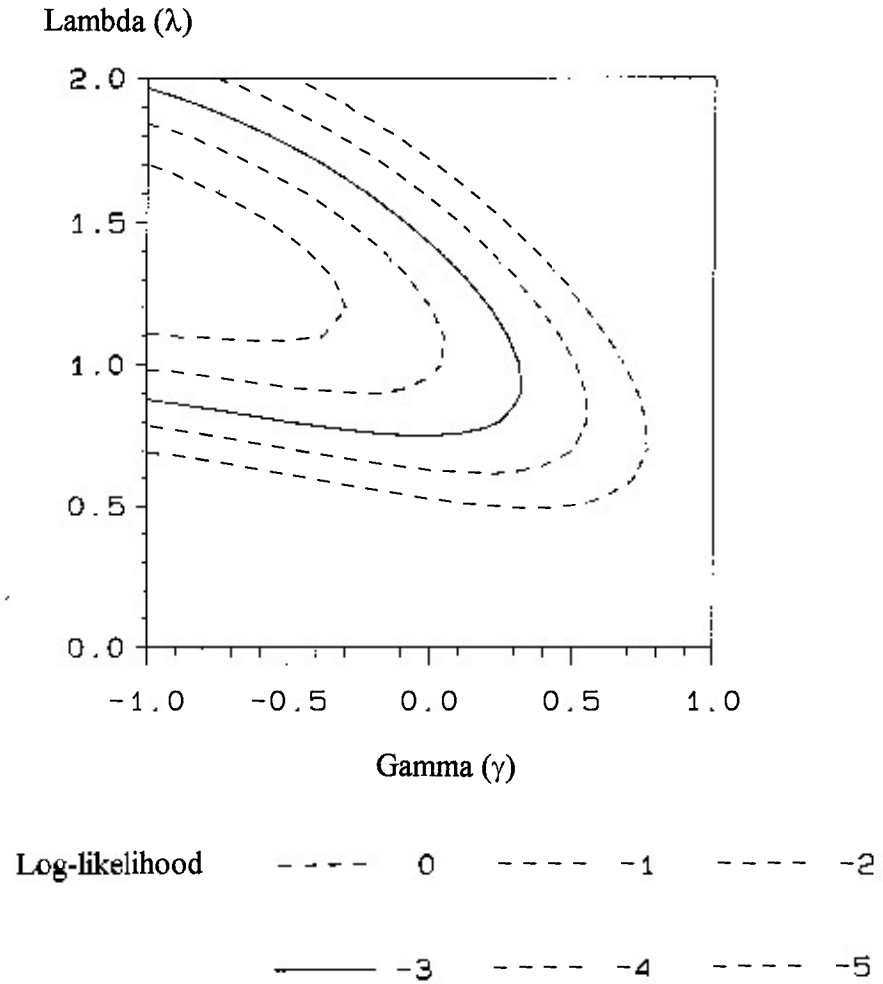

Figure 3. Contour plot of relative loglikelihood vs. λ and γ.

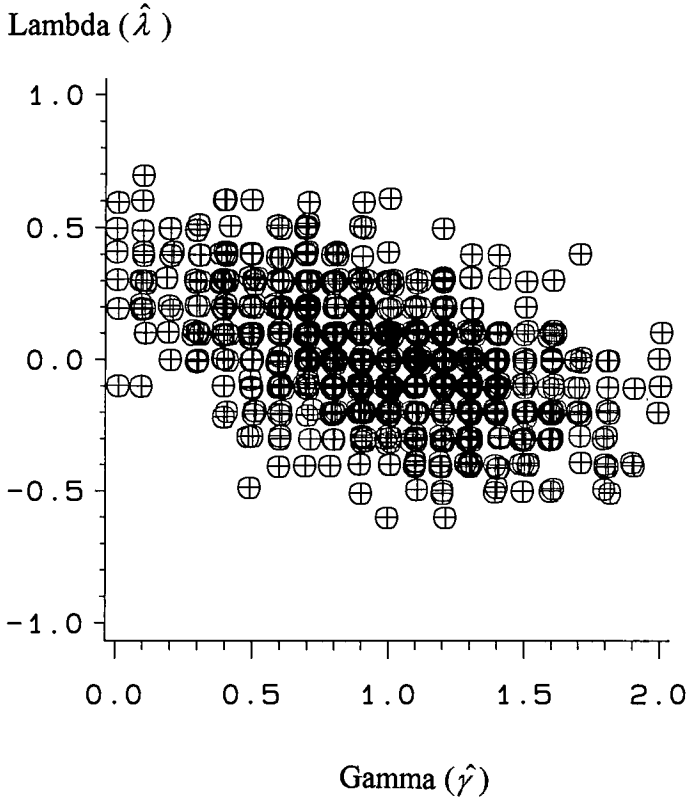

Figure 4. Scatterplot of $\hat{\lambda}$ and $\hat{\gamma}$ for the 1000 simulations where the true value is $\lambda = 0$ and $\gamma = 1$ for the 3-component model.

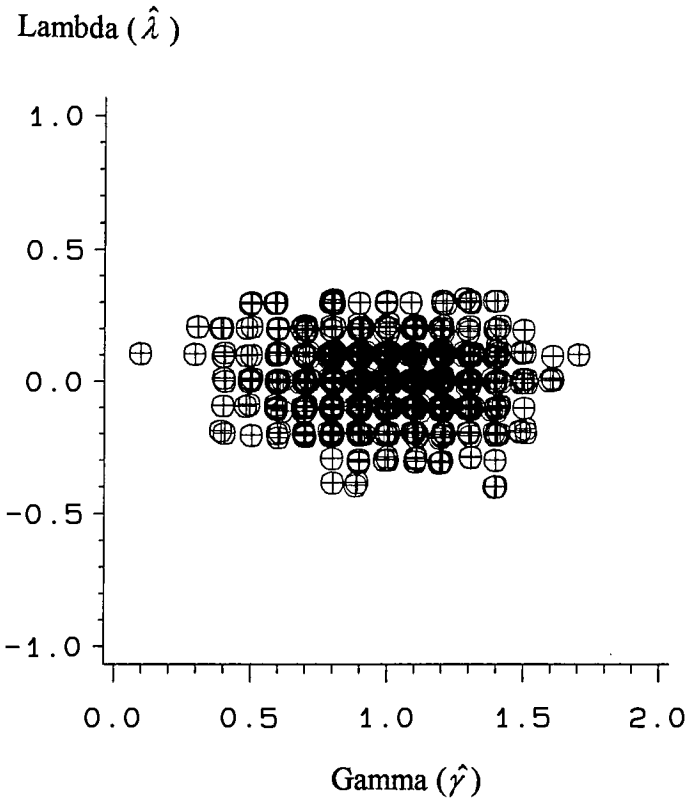

Figure 5. Scatterplot of $\hat{\lambda}$ and $\hat{\gamma}$ for the 1000 simulations where the true value is $\lambda = 0$ and $\gamma = 1$ for the 6-component model.

Appendix A. Log-likelihood Function for the Transformed Variables

Using the assumptions outlined in the "Estimation" section of this paper and applying the transformation given by g(y) = $f(y^{(\lambda)}) \cdot \left| \frac{dy^{(\lambda)}}{dy} \right|$, for a given λ and γ, the likelihood function in terms of the original observations is

$$L(\beta, \gamma, \lambda, \gamma) = \prod_{i=1}^{n} f(y_i^{(\lambda)}) \cdot \left| \frac{dy_i^{(\lambda)}}{dy_i} \right|$$

or

$$L(\beta, \sigma, \lambda, \gamma) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{1/2} \cdot \sigma_{(\lambda,\gamma)}} \cdot \exp\left[- \frac{\left(y_i^{(\lambda)} - \boldsymbol{x}_{ij}^{(\gamma)} \beta_{j(\lambda,\gamma)} \right)' \left(y_i^{(\lambda)} - \boldsymbol{x}_{ij}^{(\gamma)} \beta_{j(\lambda,\gamma)} \right)}{2\sigma_{(\lambda,\gamma)}^2} \right] \cdot \left| \frac{dy_i^{(\lambda)}}{dy_i} \right|.$$

For a given λ and γ, the likelihood function is equivalent to an ordinary least squares regression. Therefore, maximizing first with respect to β and substituting its maximum likelihood estimate back into $L(\beta, \gamma, \lambda, \gamma)$, we have

$$L_{\max}(\sigma, \lambda, \gamma) = \frac{1}{(2\pi)^{n/2} \cdot \sigma_{(\lambda\gamma)}^{n}} \cdot \exp\left[- \frac{\sum (y_i^{(\lambda)} - \hat{y}_i^{(\lambda)})^2}{2\sigma_{(\lambda\gamma)}^2} \right] \cdot \prod_{i=1}^{n} \left| \frac{dy_i^{(\lambda)}}{dy_i} \right|,$$

where $\hat{y}_i^{(\lambda)} = x_{ij}^{(\gamma)} \hat{\beta}_{j(\lambda\gamma)}$, $\hat{\beta}_{j(\lambda\gamma)}$ is the least squares estimate of $\beta_{j(\lambda\gamma)}$, and $\prod_{i=1}^{n} \left| \frac{dy^{(\lambda)}}{dy_i} \right|$ reduces to $\prod_{i=1}^{n} \left| y_i^{\lambda-1} \right|$. Next, maximizing with respect to σ and similarly substituting its maximum likelihood estimate into $L_{\max}(\sigma, \lambda, \gamma)$, leads to

$$L_{\max}(\lambda,\gamma) = \frac{1}{(2\pi)^{n/2} \cdot \hat{\sigma}_{(\lambda\gamma)}^{n}} \cdot \exp\left[-\frac{n}{2}\right] \cdot \prod_{i=1}^{n} \left| y_i^{(\lambda-1)} \right| ,$$

where $\hat{\sigma}_{(\lambda\gamma)}^2 = \dfrac{\sum (y_i^{(\lambda)} - \hat{y}_i^{(\lambda)})^2}{n}$, also obtained from the least squares regression. The corresponding log-likelihood is

$$\ln L_{\max}(\lambda,\gamma) = C - \frac{1}{2} n \ln \hat{\sigma}_{(\lambda\gamma)}^2 + (\lambda - 1) \sum_i \ln y_i ,$$

where $C = -\dfrac{n}{2} - \dfrac{n}{2} \ln(2\pi)$. For a given experiment, C is a constant which depends only on n. Therefore, the log-likelihood function is, within a constant, equal to

$$\ln L_{\max}(\lambda,\gamma) = -\frac{1}{2} n \ln \hat{\sigma}_{(\lambda\gamma)}^2 + (\lambda - 1) \sum_i \ln y_i .$$

Appendix B. Derivation of σ to Obtain a Given R^2_{adj}

The σ required to obtain a final R^2_{adj} of 0.95 was derived as follows. Let

$R^2_{adj} = 1 - \frac{SSE/(n-p)}{SST_{corr}/(n-1)}$, where SSE = error sum of squares, SST_{corr} = corrected total sum of squares, n = number of observations and p = number of parameters in a least squares linear regression. If the matrix $\mathbf{H} = \mathbf{X(X'X)^{-1}X'}$, then $\hat{\mathbf{Y}} = \mathbf{X\hat{b}} = \mathbf{X(X'X)^{-1}X'Y} = \mathbf{HY}$, and $SSE = \mathbf{Y'(I-H)Y}$. Letting $\mathbf{X_o} = \mathbf{1_{nx1}}$ and $\mathbf{H_o} = \mathbf{X_o(X_o'X_o)^{-1}X_o'}$, $SST_{corr} = \mathbf{Y'(I-H_o)Y}$. This, then, means that $R^2_{adj} = 1 - \frac{\mathbf{Y'(I-H)Y}/(n-p)}{\mathbf{Y'(I-H_o)Y}/(n-1)}$.

To get the expected value, the expectations of the numerator and denominator were taken separately. This means we are actually estimating $E^*(R^2_{adj}) = \frac{E(numerator)}{E(denominator)}$ rather than $E(R^2_{adj}) = E\left[\frac{numerator}{denominator}\right]$. While these expectations are not equivalent, inspection of the data produced showed $E^*(R^2_{adj})$ to be a fairly good approximation of $E(R^2_{adj})$. If we assume that R^2_{adj} comes from a least squares regression, $E(\mathbf{Y}) = \mathbf{Xb} = \mu$ and $V(\mathbf{Y}) = \sigma^2\mathbf{I}$. Since $E(\mathbf{x'Ax}) = tr(\mathbf{A}) + \mu'\mathbf{A}\mu$ for any quadratic form, the expected value of the numerator is σ^2 and the expected value of the denominator is $\sigma^2 + \frac{\mathbf{b'X'(I-H_o)Xb}}{(n-1)}$.

Thus, the approximate expected value of R^2_{adj} is

$E^*(R^2_{adj}) = 1 - \dfrac{\sigma^2}{\sigma^2 + [\mathbf{b}'\mathbf{X}'(\mathbf{I} - \mathbf{H}_0)\mathbf{X}\mathbf{b}]/(n-1)}$, from which the required σ^2 can then be derived.

Appendix C. SAS Code

SAS program to find the "appropriate" transformation of the response and predictor variables using a grid-search algorithm. No missing data is permitted and, at present, this program only allows for one response variable.

The data for the simulations was generated using a macro called "model" with the following calls:

&data: ASCII file containing the design points.
&nobs: number of observations.
&r: desired R^2_{adj} in fraction form.

The following variables are also used:

b: vector of coefficients.
ho: $\mathbf{X_o(X'X)^{-1}X'_o}$
In: nobs x nobs identity matrix
sigma: required sigma

The $\hat{\lambda}$ and $\hat{\gamma}$ are generated using a macro called "loop" with the following calls:

&data: SAS data set of raw data
&resp: response variables.
&pred: predictor variables.
&lamrange: range of λ-values.
&gamrange: range of γ-values.
&nobs: number of observations.
&reallam: "true" value of lambda.
&realgam: "true" value of gamma.

Other variables used:

tresp: transformed response.
npred: number of predictor variables.
pred: predictor variables.
tpred: transformed predictor variables.
slnr: sum of the log(response).
ss: residual sum of squares for least squares regression for each pair of λ and γ.
r: residual from least squares regression for each pair of λ and γ.
lnlke: log-likelihood function for each λ and γ.
maxln: maximum log-likelihood value.
newln: relative log-likelihood value.

```
%macro loop(data=, resp=, pred=,  lamrange=, gamrange=, nobs=,
    reallam=, realgam=, datout=);

    /* NAME RESPONSE VARIABLE */
        %let tresp = t&resp;

    /* BREAK APART PREDICTOR VARIABLES */
        %let npred=0; %let temp=begin; %let i=1;
        %do %until(%quote(&temp)= or &i>1000);
            %let temp=%scan(&pred, &i, %str( ));
            %if %quote(&temp) ne %then %do;
                %let npred=%eval(&npred+1);
                %let pred&npred=&temp;
                %end;
            %let i=%eval(&i+1);
            %end;
        %do i=1 %to &npred;
            %put pred&i=&&pred&i;
            %let tpred&i = t&&pred&i;
            %end;

    /* CALCULATE THE SUM OF THE LN(RESP) */
        proc sort data=&data; by iter;
        data temp sum(keep= slnr iter);
            set &data; by iter;
            if first.iter then slnr=0;
            slnr + log(&resp);
            if last.iter then output sum;
            run;

    /* CALCULATE THE TRANSFORMED RESPONSE AND PREDICTOR FOR */
    /* THE RANGE OF LAMBDA AND GAMMA                        */
        data new(keep = lambda gamma &tresp iter
                %do i=1 %to &npred; &&tpred&i %end;);
            set &data;
            do lambda = &lamrange;
                if lambda gt -1e-10 and lambda lt 1e-10
                    then &tresp = log(&resp);
                    else &tresp = (&resp**lambda-1)/lambda;
                do gamma = &gamrange;
```

```
            if gamma gt -1e-10 and gamma lt 1e-10 then do;
                %do j = 1 %to &npred;
                    &&tpred&j = log(&&pred&j);
                    %end;
                end;
            else do;
                %do j = 1 %to &npred;
                    &&tpred&j = (&&pred&j**gamma-1)/gamma;
                    %end;
                end;
            output;
            end;
        end;
    run;

/* DO LEAST SQUARES REGRESSION FOR EACH PAIR OF LAMBDA AND */
 /* GAMMA AND CALCULATE THE SUM OF SQUARES RESIDUAL          */
    proc sort data=new; by iter lambda gamma;
    proc reg noprint data=new;
        by iter lambda gamma;
        model &tresp = %do i=1 %to &npred; &&tpred&i  %end;;
            restrict %do i=1 %to &npred; %if &i=&npred %then %str(&&tpred&i);
            %else %str(&&tpred&i +) ; %end; = 0;
            output out=resid(keep=iter lambda gamma r)
            r= r;

    proc datasets library=work nolist;
        delete new;
        run;

    proc means noprint;
        by iter lambda gamma;
        var r;
        output out = ssresid(keep=iter lambda gamma ss)
            uss= ss;
        run;

/* CALCULATE THE LOG-LIKELIHOOD FUNCTION FOR EACH PAIR OF */
/* LAMBDA AND GAMMA                                       */
    data lnlike;
        merge sum ssresid;
        by iter;
```

```
        lnlke=-0.5*&nobs*log(ss/&nobs) + (lambda-1)*slnr;
        run;

/* CALCULATE RELATIVE LOG-LIKELIHOOD */
    proc means max noprint;
        by iter;
        id lambda gamma;
        var lnlke;
        output out=maxln(keep=iter lambda gamma maxln)  max=maxln;
        run;

    data maxlike;
        merge maxln lnlike;
        by iter;
        newln = lnlke - maxln;
        run;

/* CHECK WHETHER "TRUE" λ FALLS WITHIN 95% CI ABOUT λ̂ */
    proc sort data=maxlike; by iter lambda; run;
    data histlam1;
        set maxlike;
        if lambda < &reallam+1e-10 and lambda > &reallam-1e-10;
        run;

    proc means data=histlam1 max noprint;
        by iter;
        var lnlke;
        output out=lam(keep=iter lnl)
            max=lnl;
        run;

    data histlam2;
        merge histlam1 lam;
        by iter;
        run;

    data histolam(keep=iter lamci);
        set histlam2;
        if abs(maxln-lnl)<1.92 then lamci = 'y';
            else lamci = 'n';
        run;
```

```
/* CHECK WHETHER "TRUE" γ FALLS WITHIN 95% CI ABOUT γ̂ */
    proc sort data=maxlike; by iter gamma; run;
    data histgam1;
        set maxlike;
        if gamma  < &realgam+1e-10 and gamma  > &realgam-1e-10;
        run;

    proc means data=histgam1 max noprint;
        by iter;
        var lnlke;
        output out=gam(keep=iter lnl)
            max=lnl;
        run;

    data histgam2;
        merge histgam1 gam;
        by iter;
        run;

    data histogam(keep=iter gamci);
        set histgam2;
        if abs(maxln-lnl)<1.92 then gamci = 'y';
        else gamci = 'n';
        run;

/* FIND WHETHER THE "TRUE" λ AND γ FALL WITHIN A 95% JOINT */
/* CONFIDENCE REGION ABOUT λ̂ AND γ̂                        */
    data temp3;
        set maxlike;
        if gamma  < &realgam+1e-10 and gamma  > &realgam-1e-10
            and lambda < &reallam+1e-10 and lambda > &reallam-1e-10;
        run;

    data joint(keep=iter jointci);
        set temp3;
        if abs(maxln-lnlke)<3.0 then jointci = 'y';
            else jointci = 'n';
        run;

    data combine1;
        merge histolam histogam;
        by iter;
```

```
        run;

    data combine2;
        merge combine1 joint;
        by iter;
        run;

    data combine3;
        merge maxlike combine2;
        by iter;
        run;

/* KEEP ONLY LAMBDA AND GAMMA FOR MAXIMUM LIKELIHOOD */
/* VALUE                                              */
    data &datout;
        set combine3;
        if maxln  ne lnlke  then delete;
        run;

%mend loop;


%macro model (data=, nobs=, r=);
/* READ IN DATA */
    data dat;
        infile &data        ;
        input x0 x1 x2;
        run;

/* CALCULATE REQUIRED σ */
    proc iml;
        use dat;
        read all into x;
            b = {1.9048, 1.9048,-3.8096};
            xo = j(&nobs,1,1);
            ho = xo*inv(xo`*xo)*xo`;
            In = I(&nobs);
            sigma = sqrt((1-&r)/&r*(b`*x`*(In-ho)*x*b)/(&nobs-1));
        create temp var{sigma};
            append;
        quit;
```

```
   /* GENERATE DATA FOR SIMULATIONS */
      data model1.dat (keep = iter x1 x2 x3 y );
         if _n_=1 then set temp; set dat;
         x3 = 1 - x1 - x2;
         do iter = 1 to 50  ;
            y= 3.6667 + 1.9048*x1 + 1.9048*x2 - 3.8096*x3 + sigma*rannor(0);
            output model1.dat;
            end;
         run;

      proc sort data=model1.dat; by iter; run;

%mend model;


%model(data='thesis data1 a', nobs=14, r=.95)

%loop(data=model1.dat, resp = y, pred = x1 x2 x3  ,
    lamrange = 0  to 2.0 by .1 , gamrange= 0   to 2.0   by .1,
    nobs=14, reallam=1.0, realgam=1.0, datout=compile1.dat );

endsas;
```