Hierarchical-Model Fitting and Diagnostics

A Plan B Paper

submitted to the department of biostatistics

of the University of Minnesota

by

Dr. Peiming Ma

In partial fulfillment of the requirements

for the degree of

Master of Science

Dr. James S. Hodges, advisor

August, 1997

Acknowledgments

I am grateful to Dr. James S. Hodges for his valuable guidance in this project and for the financial support from his research grant. I also wish to thank Dr. J. Williams Thomas for the advice during the two years of my school work. My sincere thanks go to Dr. DeAnn Lazovich, Dr. J. Williams Thomas, and Dr. James S. Hodges for giving suggestions to improve this work and serving in my plan B committee.

It was the loss of my mother from Amyotrophic Lateral Sclerosis (Lou Gehrig's disease) two years ago that had me start the program in biostatistics. The memory of her has been an important support in completing the program.

Abstract

We describe certain computer procedures in S+ that were developed to perform Bayesian computations for hierarchical models by the Gibbs sampler method and to perform diagnostics: added-variable plots, collinearity measures, case influence, residuals, and a transformation diagnostic. The procedures are then applied to two studies: one on nicotinic receptors in neural tissues in rats, and the other on stress in an artificial human mandible.

The study is based on Hodges' theoretical framework in [**H**].

KEY WORDS: hierarchical models; Gibbs sampler; diagnostics.

Contents

1 Introduction

Hierarchical models have been used widely in understanding and solving statistical problems because, first of all, the advancement of both computer technology and statistical computing has made it more practical and, secondly, data often arise naturally with several levels of hierarchy. For example, suppose a clinician wants to know patients' blood concentrations of a certain drug to understand the drug's bioavailability, assuming the drug is administered steadily during the study so the inflow of the drug is viewed as constant. He or she measures

the blood concentration n times for each of the k patients. Let y_{ij} be the concentration at the j-th time from the i-th patient. One way of modeling this is to use the following equations for two different levels:

$$\begin{cases} y_{ij} &= \theta_i + \epsilon_{ij} \\ \theta_i &= \mu + \delta_i \end{cases} \qquad i = 1, \ldots, k; \quad j = 1, \ldots, n \tag{1}$$

where we suppose the error terms ϵ_{ij} and δ_i are independent and normally distributed with some unknown variances σ^2 and τ^2 respectively (i.e., $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$ and $\delta_i \overset{iid}{\sim} N(0, \tau^2)$). The parameter θ_i can be interpreted as the average amount of the agent for the i-th patient and μ the average amount of the agent among all patients. In a Bayesian setting a prior for μ is generally added:

$$\mu = M + \xi, \tag{2}$$

where M is the prior mean of μ and $\xi \sim N(0, s^2)$ so s^2 is the prior variance; M and s^2 are assumed known. The equations (1) and (2) then comprise a simple hierarchical model of three levels. See [**BR**] for a very readable introduction to hierarchical models.

Once a hierarchical model is established as reasonable, just as for any modeling problem, the next tasks are i) model fitting, or estimating all parameters in the model, ii) diagnostics, or applying procedures to determine if the estimated model provides a good fit to the data, and iii) interpretation, or extracting information from the data through the proposed model. However, the same tasks for hierarchical models are usually more involved than for simpler models, for reasons such as the nonlinearity of estimates in the observations, arising from multiple unknown variances.

Hodges [**H**] proposed a new method to study hierarchical models. His idea is to reformulate the model as an ordinary linear model with heteroscedastic errors by rewriting the modeling equations of the higher levels in a special way. The following describes Hodges' method using the above example. After rewriting the equations in (1) and (2) as

$$\begin{cases} y_{ij} = \theta_i + \epsilon_{ij} & \\ 0 = -\theta_i + \mu + \delta_i & i = 1, \ldots, k \\ M = \mu - \xi & j = 1, \ldots, n \end{cases} \tag{3}$$

the above equations can be expressed in matrix notation as $Y = X\Theta + E$:

$$\left(\begin{array}{c} y \\ \hline 0_k \\ \hline M \end{array}\right) = \left(\begin{array}{ccc|c} 1_n & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1_n & 0 \\ \hline & -I & & 1_k \\ \hline 0 & \cdots & 0 & 1 \end{array}\right) \left(\begin{array}{c} \theta_1 \\ \vdots \\ \theta_k \\ \hline \mu \end{array}\right) + \left(\begin{array}{c} \epsilon \\ \hline \delta \\ \hline -\xi \end{array}\right) \tag{4}$$

where y is a vector with components $y_{ij}, i = 1, \ldots, k; j = 1, \ldots, n$, properly ordered, 1_m is a column of vector of 1's of length m for any given positive integer m, and I is the identity matrix of order k. Each 0 in the matrix has appropriate yet obvious size and shape. The vector E of the error terms has a diagonal covariance structure, namely, the entries of the covariance matrix E are all zeros except for the diagonal which consists of nk copies of σ^2, k copies of τ^2, and a single s^2.

It turns out that many hierarchical models, e.g., variance component models or random

coefficient regression models defined in [**BR**], can be written as $Y = X\Theta + E$ with X, Y in more general forms:

$$X = \left(\begin{array}{c|ccccc} X_1 & 0 & 0 & \cdots & 0 & 0 \\ \hline H_1 & H_2 & 0 & \cdots & 0 & 0 \\ 0 & H_3 & H_4 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & H_{r-1} & H_r \\ \hline G_1 & G_2 & G_3 & \cdots & G_{s-1} & G_s \end{array}\right) \text{ and } Y = \left(\begin{array}{c} y \\ \hline 0 \\ \vdots \\ 0 \\ \hline M \end{array}\right) \tag{5}$$

where the H_i's specify model equations for parameters of levels greater than one and G_i's and M specify prior means for parameters in Θ that are not modeled as functions of higher-level parameters. For convenience, the rows of y and X_1 will often be referred to as *data cases* and, similarly, the rows of H_i's and the rows of G_i's will be called *constraint cases* and *prior cases*, respectively.

The advantage of this matrix expression is that the theory of finite-dimensional vector spaces can be applied, so the geometric properties of such linear models become more apparent. This offers insight that might be useful for model-fitting and diagnostics.

With Hodges' method as the foundation, we have developed computing tools for hierarchical model-fitting and diagnostics. In Section 2, we describe a model-fitting method that uses the Gibbs sampler; and some diagnostic procedures which are generalized from the S+ code that Hodges used in his paper. The complete S+ code of all functions can be found in Appendix A. Then in Section 3 we consider two data sets, both of which seem to call for

models with hierarchical structure. After fitting the data with the Gibbs sampler, in Section 4 we apply the diagnostic tools and interpret the results that the model has to offer for the data. There the functionality of the S+ code for diagnostic procedures will also be exhibited.

The purpose of this work is twofold. In addition to creating some general functions to fit hierarchical models and to do diagnostics, we shall try to explore certain issues in model-fitting and diagnostics as well as to compare some of the differences between different parameterizations for the same model or different models for the same data set.

2 Computer Procedures for Hierarchical Models

In this section, we first describe briefly the process of using the Gibbs sampler to obtain approximate solutions to the hierarchical model equations. Then we present in detail the S+ functions for model fitting and diagnostics.

2.1 Model Fitting

One method for fitting a linear model of the form $Y = X\Theta + E$ with X and Y in (5), as a Bayesian analysis problem, is to use the Gibbs sampler, which relies on a Markov Chain process. Consider $Y = X\Theta + E$ where the error term E is normally distributed with mean vector 0 and covariance matrix Σ, unknown but assumed to be diagonal. (We restrict the covariance matrices for the error terms to be diagonal because it is adequate for the models that we are currently interested in and the computation is much simpler. The class of models $Y = X\Theta + E$ with X and Y in (5) is more general in that the covariance matrix is usually block-diagonal.)

To simplify notation, define $\Gamma = \Sigma^{-1}$, the diagonal matrix of error precisions. It can be easily shown that for given X and Y the posterior distribution of Θ, conditional on Γ, is normal with mean $(X'\Gamma X)^{-1}X'\Gamma Y$ and covariance $(X'\Gamma X)^{-1}$. And if the distinct diagonal elements of Γ, namely the error precisions, are all assumed to have Gamma prior distributions then, conditional on Θ, the posterior distributions of the error precisions are still Gammas with easily computable parameters. Denoting by Π the appropriate conditional density, we can generate a Gibbs sequence of random vectors:

$$\Theta_1 \sim \Pi(\Theta|\Gamma_0), \quad \Gamma_1 \sim \Pi(\Gamma|\Theta_1), \quad \ldots, \quad \Theta_k \sim \Pi(\Theta|\Gamma_{k-1}), \quad \Gamma_k \sim \Pi(\Gamma|\Theta_k) \tag{6}$$

where the starting value for Γ_0 is specified and each draw thereafter is made successively from a distribution with parameters computed from the previous term. The theory of Markov chains guarantees that as $k \to \infty$ the quantities $\frac{1}{k}\sum_{m=1}^{k}\Gamma_m$ and $\frac{1}{k}\sum_{m=1}^{k}\Theta_m$ converge to $E(\Gamma|Y)$ and $E(\Theta|Y)$ respectively, thus yielding estimates for the posterior means of Γ and Θ when k is sufficiently large. However, better estimates can be obtained by Rao-Blackwellization (see [**CG**]):

$$E(\Gamma|Y) \approx \frac{1}{k}\sum_{m} E(\Gamma|Y,\Theta_m) \quad \text{and} \quad E(\Theta|Y) \approx \frac{1}{k}\sum_{m} E(\Theta|Y,\Gamma_m), \tag{7}$$

when k is large. The posterior variances of Γ and Θ can be approximated analogously by

$$\begin{aligned} \mathrm{Var}\,(\Gamma|Y) &\approx \frac{1}{k}\sum_{m} E(\Gamma^2|Y,\Theta_m) - \left(\frac{1}{k}\sum_{m} E(\Gamma|Y,\Theta_m)\right)^2, \\ \mathrm{Var}\,(\Theta|Y) &\approx \frac{1}{k}\sum_{m} E(\Theta^2|Y,\Gamma_m) - \left(\frac{1}{k}\sum_{m} E(\Theta|Y,\Gamma_m)\right)^2. \end{aligned} \tag{8}$$

2.1.1 Gibbs Sampler

We wrote the S+ function *gibbs* to create a Gibbs sequence as in (6) from which various model quantities, including the approximate posterior means and variances of Γ and Θ in (7) and (8), can be obtained.

The *gibbs* function takes as input arguments X, Y, *rows, cols, num.iter, opt, ptab.len, pta, ptb, prior.var, g.diag, burnin*:

- X and Y are matrices constructed by the user from the outcome variable and the covariates as in (5).

- *rows* is a vector of integers used to identify the different levels of the model. Given X as in (5), the elements of *rows* describe the number of rows of the submatrices $X_1, H_1, H_3, \ldots, H_{r-1}$. For example, the first component in *rows* is also the number of data cases observed and the sum of the remaining integers in *rows* is the number of constraint cases. Finally if n is the row dimension of X, then the difference between n and the sum of the integers in *rows* gives the number of prior cases.

- *cols* is an integer vector defined similarly as *rows*. It gives the number of columns of $X_1, H_2, H_4, \ldots, H_r$ in X. (For the example of the rats study with both parameterizations in Section 3.1, $rows = (48, 16)$ and $cols = (22, 2)$; and for the first model in the jaw study in Section 3.2, $rows = (55, 106)$ and $cols = (55, 52)$.)

- *num.iter* is the number of Gibbs iterations that the user wishes to use in computing posterior quantities. The default is 1000.

- *opt* is either 0 (default) or 1. It is specified by the user. If *opt* is 1 then the Gibbs

iterates are saved as part of the output; otherwise, they are not.

- *pta*, *ptb*, and *ptab.len* are vectors of the same length. Assume all the error precisions are of the form $t_i\eta_i, i = 1, \ldots, n$, where t_i are known constants and η_i are unknown. Assume also that η_i's are divided into r groups in such a way that the first n_1 of the η_i's equal h_1 and the next n_2 of the η_i's equal h_2, etc., for a list of distinct random variables $\{h_1, \ldots, h_r\}$. And, *a priori*, the h_j's are random variables having Gamma distributions with shape parameter a_j and scale parameter b_j, therefore, the mean and variance of the prior Gamma distribution are a_j/b_j and a_j/b_j^2. Then *pta* and *ptb* are r-vectors of the parameters, i.e., a_j's and b_j's respectively and *ptab.len* is an r-vector of n_j's. (This implies that X must be set up in a particular way so that the lengths of the error variances for the rows of X correspond to *ptab.len*.) The default values of *pta* and *ptb* are vectors of 1.1 and 0.1 with the same lengths as *rows*, respectively, and for *ptab.len* the default is the *rows* vector. (For the rats study in Section 3.1, $r = 2$, $pta = (1, 1.1)$, $ptb = (0, 0.1)$, and $ptab.len = (48, 16)$; for the first model in the jaw study in Section 3.2, $r = 4$, $pta = (1.1, 1.1, 1.1, 1.1)$, $ptb = (0.1, 0.1, 0.1, 0.1)$, and $ptab.len = (55, 5, 50, 51)$).

- *g.diag* is an n-vector supplied by the user whose components are the t_i's described above. The t_i's of the default *g.diag* are all equal to 1. (For the parameterizations of the jaw study in Section 3.2, *g.diag* will not take the default value.)

- *prior.var* is a vector of variances for prior cases that the user supplies. It can be specified as a single number instead of a vector, in which case *gibbs* will use that number as the variances for all the prior cases, if more than one. The default is

1,000,000, corresponding to a "nearly" flat, low-information prior.

- *burnin* is the number of burn-ins, or initial iterations which the user chooses to discard. The default value is 100. The total number of runs in *gibbs* is the sum of *num.iter* and *burnin*.

The *gibbs* function outputs a list of objects. Besides copying X, Y, *rows, cols, num.iter, ptab.len, g.diag* from the input arguments, it also produces *mean.theta, var.theta, ehe, vhe, mean.sig2, var.sig2, Ghe* and if *opt* is 1 then the Gibbs iterates for Θ and the error precisions h_j are saved in *Gibbs.iter*.

- *mean.theta* and *var.theta* give the estimated posterior means and variances of the parameters Θ.

- *ehe* and *vhe* produce the estimated posterior means and variances of all the error precisions h_j.

- *mean.sig2* and *var.sig2* are the estimated posterior means and variances of the error variances, $1/h_j$.

- *Ghe* is a vector which contains the square root of the estimated posterior means of the error precisions, $E(t_i\eta_i)$.

- *Gibbs.iter* is obtained if *opt* is 1. *Gibbs.iter* is a list of two matrices *thit* and *hit* which are the Gibbs draws of Θ and h_j's. The i-th row of the matrices are the saved i-th Gibbs draws of Θ and h_j's after the *burnin* iterations. The column dimensions of *thit* and *hit* are equal to the length of Θ and r, the number of error precisions h_j, respectively.

2.1.2 Convergence Plots

It is usually a difficult question to ascertain whether the Gibbs sampler has converged and, perhaps, more difficult to determine the rate of convergence if it does. One way of getting some sense for both questions is to make the so-called time plots for the random draws of parameters in *gibbs*. In such plots, the horizontal axis is the iteration number and the vertical axis is the value of the draws.

The function *conv* takes the model output from *gibbs* with *opt* equal to 1 and vectors *theta.idx* and *error.idx*. The two vectors, given by the user, are respectively the indices of the model parameters Θ and error precisions h_i for which the user wants to make convergence plots. The function then sketches six time plots per screen for the given indices of the Gibbs iterates and asks the user's choice to continue for the next screen of six plots if there are more.

2.2 Diagnostics

We wrote five functions to do some diagnostics that are popular for ordinary linear models. They are added-variable plots, collinearity measures, case influence, residuals, and a transformation diagnostic. One may find their description in many standard textbooks of statistical modeling (for example, in [**C**] and [**A**]). However, for hierarchical models the usual diagnostic procedures often need to be modified before they can be meaningfully applied. The S+ functions given below implement the modified diagnostics described in Section 4 of [**H**] in detail.

2.2.1 Added-Variable Plots

An added-variable plot is used to test whether a variable not currently in the model helps to explain additional variation or is correlated with the residuals from the current model.

The function *addvar* makes added-variable plots. It takes as input the *gibbs* output object, say, *model* and an external variable in the level of either data cases or constraint cases: *addv1, addv2*. The user can give a variable in one of the two levels; or if the user specifies both *addv1* and *addv2*, they will be combined, consistent with (5) to produce a single added variable. *addvar* also takes an optional argument *plotit*: if *plotit* is true (T) then it will give a scatter plot of the scaled residuals versus the new scaled residuals obtained after the external variable is regressed on X using the posterior mean of the diagonal of $\Gamma^{1/2}$ as the scale weights. The rationale for this added-variable plot is given in Section 4.1 of Hodges [**H**]. The default of *plotit* is false (F).

The user can also select the plotting characters by supplying the argument *labs* which is a vector of characters of length the same as the number of rows of X. The default here is "d" for data cases, "c" for constraint cases, and "p" for prior cases. The function produces two vectors and three real numbers: the scaled residuals of the external variable *avB* and the scaled residuals *scal.res* from the original regression, the *slope* from the least squares fit of the two variables, the usual standard error *std.err* of the slope, and finally the p-value *p.value* for the slope (see [**W**]).

2.2.2 Collinearity

Collinearity in the design matrix could be a problem in fitting hierarchical models, as it could cause slow convergence of the Gibbs sequence and imprecision of parameter estimates. One way of detecting collinearity is to find the condition number of the matrix $\Gamma^{1/2}X$, i.e., to find the ratio of the greatest and smallest eigenvalues of the matrix (Section 4.3 of [**H**]).

The function *colli* takes the model output produced by *gibbs* and computes condition numbers. The user has the option of only computing the condition number at the posterior mean of Γ if *opt* is assigned 0. However, if collinearity is of concern and if a study of the convergence of the Gibbs sequence is intended, then *opt* may be set to 1 to get condition numbers at every Gibbs iteration. In the latter case, the call to *gibbs* also needs to take the argument *opt* as 1 so the Gibbs iterates are saved by *gibbs*.

2.2.3 Case Influence

To study the influence that certain extreme cases have on the fitted model, one would usually refit the model after deleting such cases and compare the new model fit with the original one. In the case of model fitting by Gibbs samplers, the amount of computing work involved in this task is formidable. One alternative described in Section 4.4 of [**H**] is the linear approximation method. This method computes the approximate change in the posterior means of the parameters resulting when the i-th case is deleted. It does so by fixing the unknown precisions at their posterior means (from the full dataset) and using case-deletion formulae from linear regression. Then the change in each parameter is divided by its *posterior standard deviation* from the full data set (see Section 4.4 in [**H**] for the definition of the term)

to give the relative change from deleting the i-th case.

The function *influ1* takes the model as input and computes a matrix spC whose (i,j)-entry is the approximation of the relative change of the j-th coordinate of Θ from deleting case i. The cases include constraint and prior cases as well as data cases. It also gives $max.spC$ which is the maximum of the absolute values of the relative changes in spC and entries of the coordinates *entries*, at which the maximum absolute values occur.

2.2.4 Residuals

The diagnostic procedure used the most is undoubtedly the residual plot. The function *resd* takes as its arguments the model output from *gibbs* and a plotting option *plotit* and produces output vectors *yhat* of fitted values (computed using the posterior mean of Θ), *stud.res* of studentized residuals for all data cases, *slope*—the slope from the least squares fit of the two variables *stud.res* and *yhat*, and its standard error *std.err* of *slope*. The user may choose to specify the logical variable *plotit* as false, or, true (which is the default). If true then *resd* produces two QQ-plots of studentized residuals, one for the data cases and one for the constraint cases side by side, and prompts the user's carriage return to give a residual plot of *stud.res* versus *yhat* for all data cases (see Section 4.5 of [**H**]).

2.2.5 Transformations

Transformation diagnostics let the the user study the possibility that, after transforming the response from y to y^λ, the model would better explain the variation of the transformed response (see Section 4.2 of [**H**]). The number λ can be estimated roughly and checked visually using an added-variable plot for the constructed variable $\hat{y}\ln\hat{y} - \hat{y} + 1$, where the

slope of the added-variable plot is $1-\lambda$. This is called the Andrews transformation plot (see [**A**] and [**C**]).

Based on this idea, the function *transf* takes the model output from *gibbs* and does an added-variable plot for the variable $(\hat{y}+C)\ln(\hat{y}+C)-(\hat{y}+C)+1$. The constant C is user-given and necessary when the fitted values are negative and/or the transformation of y to $(y+C)^{\lambda}$ is actually of interest. The default value of C is 0. The output of *transf* consists of the Andrews added variable *avtx*, the scaled residuals *scal.res*, the added constant C, the estimated exponent *lambda* for λ, and the standard error *std.err* of λ which comes from the least squares fit to the added-variable plot. The user again may specify *plotit* as true (the default value is false); then *transf* presents a scatter plot of the scaled residuals versus the Andrews added variable for transformation with plotting characters "d" for data cases, "c" for constraint cases, and "p" for prior cases.

3 Discussion of Two Studies and Data Sets

We discuss two studies in this section by describing their data sets and presenting the questions that the investigators wished to answer. We shall see that the data are suitable to be modeled with hierarchical structures. We then set up the models by constructing the model equations. The actual data sets can be found in Appendix B.

3.1 Rats Study

Nicotine acts in the human body in various ways. Its action on neural tissue is of particular interest to neuroscience researchers. A nicotine molecule acts on neural tissue by binding

with specific molecules, called receptors, on the surface of a neural cell. Nicotine molecules can bind to more than one type of receptor. If neural cells are regularly bathed with nicotine *in vivo*, then the cell surfaces tend to express more nicotine receptors. However, they will not necessarily express equal or proportional numbers of different kinds of receptors.

This study was intended to see whether two different kinds of nicotine receptors in rat neural tissue—types A and B—were expressed in equal numbers after the rats were repeatedly injected with nicotine molecules in a saline vehicle. Eight rats were treated by nicotine, while eight other rats were injected with the saline vehicle only, as a control. When the rats were sacrificed, two kinds of neural tissue (cortical and adrenal) were harvested from all eight rats in each group, while two other kinds of neural tissue (hippocampus and thalamus) were taken from only four rats in each group. Each neural sample was split into two pieces and assayed to count nicotine receptors. Different radiolabeling molecules were used for the two pieces: one piece from each pair was assayed using tritiated epibatidine, while the other was assayed using tritiated cytisine. Epibatidine binds to both types of nicotine receptors, while cytisine only binds to type A, so the difference between the two counts estimates the number of type B receptors.

The investigators were interested specifically in whether type B receptors were expressed in response to nicotine, and whether the number of such receptors (per unit of neural tissue) differed across the four kinds of neural tissue. Thus the obvious analysis was an ANOVA with rats as subjects, nicotine vs. control as a between-subject factor, and tissue type as a within-subject factor. The analysis was complicated by imbalance: two tissue types were assayed on eight rats per group, while two other tissue types were assayed on only four rats

per group. Originally, two analyses were done: one including all eight rats per treatment group but only two tissues (cortical and adrenal), and a second including all four tissues but only four rats per treatment group.

In this paper, we consolidate these analyses into a single analysis that uses all the data. The diagnostics also allow us to address a question about the investigator's method: the rats were always treated and assayed in the same order (rats number 1 through number 8 in the control group, followed by rats number 1 through number 8 in the nicotine group), and we were concerned that there was an order effect.

Let y_{ijk} be the difference of epibatidine and cytisine receptors for rat i, treatment j, and tissue k, where

i indexes rats within treatment group, $i = 1, 2, \ldots, 8$;

j indexes treatment, $j = 1$ (control) and 2 (treatment); and

k indexes tissue, $k = 1, 2, 3, 4$ (in the order cortex, adrenal glands, thalamus, hippocampus).

The within-subject level has three independent parameters for the tissue types and three for the interaction of treatment and tissue. Call them $\beta_1, \beta_2, \beta_3$ and $\zeta_1, \zeta_2, \zeta_3$ respectively. Then we model the data by

$$y_{ijk} = \theta_{ij} + x_1^k\beta_1 + x_2^k\beta_2 + x_3^k\beta_3 + x_1^{jk}\zeta_1 + x_2^{jk}\zeta_2 + x_3^{jk}\zeta_3 + \epsilon_{ijk}$$

where θ_{ij} are the rat-specific intercept parameters and $\epsilon_{ijk} \overset{iid}{\sim} N(0, \sigma^2)$. Let α represent the

treatment effect. Then the between-subjects equation is:

$$\theta_{ij} = \mu + z_j\alpha + \delta_{ij}$$

where $\delta_{ij} \sim N(0, \tau^2)$. The x's and z_j in the above two equations are explained below.

We use two different parameterizations, i.e., two definitions of the x's, z, β's, ζ's, and α. The first one is an *indicator parameterization*. In this model the covariates are defined as indicator variables. So for example $x_1^k = 1$ if the tissue is adrenal ($k = 2$) and 0 otherwise. And x_1^{jk} is an indicator variable describing tissue-by-treatment interactions; so $x_1^{jk} = 1$ if $k = j = 2$ and 0 otherwise. And $z^j = 0$ if $j = 1$ and 1 if $j = 2$. The complete definition of covariates for x's is shown below:

k	x_1^k	x_2^k	x_3^k	x_1^{1k}	x_2^{1k}	x_3^{1k}	x_1^{2k}	x_2^{2k}	x_3^{2k}
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	0
3	0	1	0	0	0	0	0	1	0
4	0	0	1	0	0	0	0	0	1

With these definitions all the parameters in the model can be intuitively interpreted. For example, α is the average difference in the number of type B receptors between the nicotine group and the control group, θ_{ij} measures the average number of type B receptors for cortical tissue in rat i with treatment j, the parameter β_1 is the average difference between adrenal and cortical tissues in control rats, etc.

The model can be made to have less collinearity by using an *orthogonal parameterization*. Since the data are not balanced, a complete orthogonalization would obscure the meanings

of the parameters, so we settle just for a partial orthogonal parameterization. The covariates x's are defined as follows:

k	x_1^k	x_2^k	x_3^k	x_1^{1k}	x_2^{1k}	x_3^{1k}	x_1^{2k}	x_2^{2k}	x_3^{2k}
1	1	0	1	1	0	1	−1	0	−1
2	−1	0	1	−1	0	1	1	0	−1
3	0	1	−1	0	1	−1	0	−1	1
4	0	−1	−1	0	−1	−1	0	1	1

and $z^j = 1$ if $j = 1$ and -1 if $j = 2$. Because of the data imbalance, the columns corresponding to x_3^k and x_1^{jk} are not orthogonal to each other. Some parameters, nevertheless, still represent certain quantities of interest in the study: β_1 is the negative of one half times the difference between cortical and adrenal tissue types and similarly α represents one half times the average change between the control group and treatment group.

The models are fitted with Gamma priors for $1/\sigma^2$ and $1/\tau^2$ having parameters equal to 1.1 and 0.1, respectively. And 1000 iterations and 100 burn-ins are chosen. We present the results of the model-fitting and diagnostics in Section 4.1.

3.2 Jaw Study

Computer technology now allows the construction of a "virtual dental patient", a three-dimensional digital simulation of many aspects of an individual's oral anatomy. One specific aspect captured by the simulation is the structure of an individual's mandible (jawbone), the teeth, and soft tissues attached to it. Such a simulation allows study of the stress in the structure resulting from various biting forces. In general practice, a simulated jaw would permit a dentist to examine the effects of orthodontic or other types of treatments in

simulation and thus reduce the amount of guessing in treating patients.

Before the virtual dental patient is released into general use, it must be evaluated extensively to ensure that it is, indeed, a faithful representation. A recent study applied stress to a human mandible and measured the resulting strains at several points on the mandible's surface. These measurements will be compared to analogous strains computed from the virtual dental patient.

As an initial study, the experimenters used only a single mandible. Since they had no prior data on which to base sample-size calculations, a mock data set, constructed using a plastic mandible manufactured to resemble a human one in shape and gross elasticity, was analyzed by a statistician to get some sense of the variability of strains (the outcome measure) at several displacements in a precisely specified distortion. Three strain gauges were attached to the plastic mandible; each gauge measures the strain in two perpendicular directions on the surface of the mandible. The data set in Appendix B records an outcome measure which is the maximum principal (the strain in one of the two directions) from one gauge, in "microstrain" units. The jaw was flexed 5 times and the process of each time is called a repetition. We analyze this measurement as a hierarchical model from five distinct applications of the stress, with each application covering a range of displacements.

In the five repetitions there are 55 strains measured at 52 different displacements—three of the displacements had two measurements. Let $y_j, j = 1, \ldots, 55$, be the measurements of the strains (ordered as in the data set). Define two functions $I(j)$ and $K(j)$ as follows: if the observation y_j is in the i-th repetition ($1 \leq i \leq 5$) and the k-th displacement $d_k, k = 1, \ldots, 52$, then $I(j) = i$ and $K(j) = k$ (for example, $I(1) = 1$ and $K(1) = 20$ since the first observation

is in the first repetition and at $d_{20} = -123.6$, the 20-th displacement). The table below gives an abridged layout of the data where the top row indexes the displacements and the left-most column indexes the repetition.

	1	2	3	4	5	6	...	45	46	47	48	49	50	51	52
1							...	y_9				y_{10}			
2						y_{11}	...								
3					y_{24}		...		y_{33}	y_{34}					
4			y_{35}				...				y_{43}		y_{44}		y_{45}
5	y_{46}	y_{47}		y_{48}			...	y_{54}						y_{55}	

The 55 observations $y_j, j = 1, \ldots, 55$ are labeled in the table from left to right then top to bottom. So the first observation y_1 has coordinates $i = 1, k = 20$ and y_{46} at the lower left corner has coordinates $i = 5, k = 1$.

Each repetition can be thought of as yielding a smooth curve of strain as a function of displacement, with each curve being a draw from a distribution of similar curves. Here is a heuristic description of the model:

1. Each strain is a draw with error from its repetition's smooth function of displacement,

2. Each repetition's smooth function is linked to a mean smooth function as follows: at the smallest absolute value of the displacement for which that repetition was observed, the repetition's smooth function equals the mean smooth function plus an error.

3. Each repetition's smooth function changes between each pair of adjacent displacements by an amount "similar to" corresponding changes in the mean smooth function.

With these hypotheses, we may now establish the hierarchical model. Define $\theta_{i,k}$ to be the value of the strain function at the k-th displacement in the i-th repetition. Then the first assumption above leads to the following model equations:

$$y_j = \theta_{I(j),K(j)} + \epsilon_j, \quad \epsilon_j \overset{iid}{\sim} N(0, \sigma^2), \qquad j = 1, \ldots, 55.$$

Now let μ_k be the value of the mean function at the k-th displacement. Five equations result from the second assumption:

$$\begin{cases} \theta_{1,33} = \mu_{33} + \nu_1, & \theta_{2,34} = \mu_{34} + \nu_2, \\ \theta_{3,32} = \mu_{32} + \nu_3, & \theta_{4,30} = \mu_{30} + \nu_4, \\ \theta_{5,31} = \mu_{31} + \nu_5, \end{cases} \tag{9}$$

where $\nu_i \overset{iid}{\sim} N(0, \eta^2)$. For each fixed i, let j and $j-1$ be such that $I(j) = I(j-1) = i$, the third assumption yields fifty equations:

$$\theta_{i,K(j)} - \theta_{i,K(j-1)} = \mu_{K(j)} - \mu_{K(j-1)} + \delta_j, \quad \delta_j \overset{iid}{\sim} N(0, (d_{K(j)} - d_{K(j-1)})\tau^2). \tag{10}$$

And finally from the smoothness of the mean function another set of model equations follow:

$$\mu_{k+1} = \mu_k + \xi_k, \quad \xi_k \overset{iid}{\sim} N(0, (d_{k+1} - d_k)\phi^2), \qquad k = 1, ..., 52. \tag{11}$$

Again, for the precisions $1/\sigma^2, 1/\eta^2, 1/\tau^2$, and $1/\phi^2$ we shall initially use Gamma priors with shape parameter 1.1 and scale parameter 0.1 to fit a hierarchical model by the Gibbs sampler

method, using 1000 iterations and 100 burn-ins. Then we propose a modified model and fit the data with different priors. The interpretation and diagnostic results are in Section 4.2.

4 Results from Diagnostics

4.1 Rats Study—Two Parameterizations

Both the indicator and orthogonal parameterizations were fitted with quick convergence. We show the convergence plots for θ_{32}, β_3, ζ_1, α, and the two error precisions $1/\sigma^2, 1/\tau^2$ for the indicator parameterization:

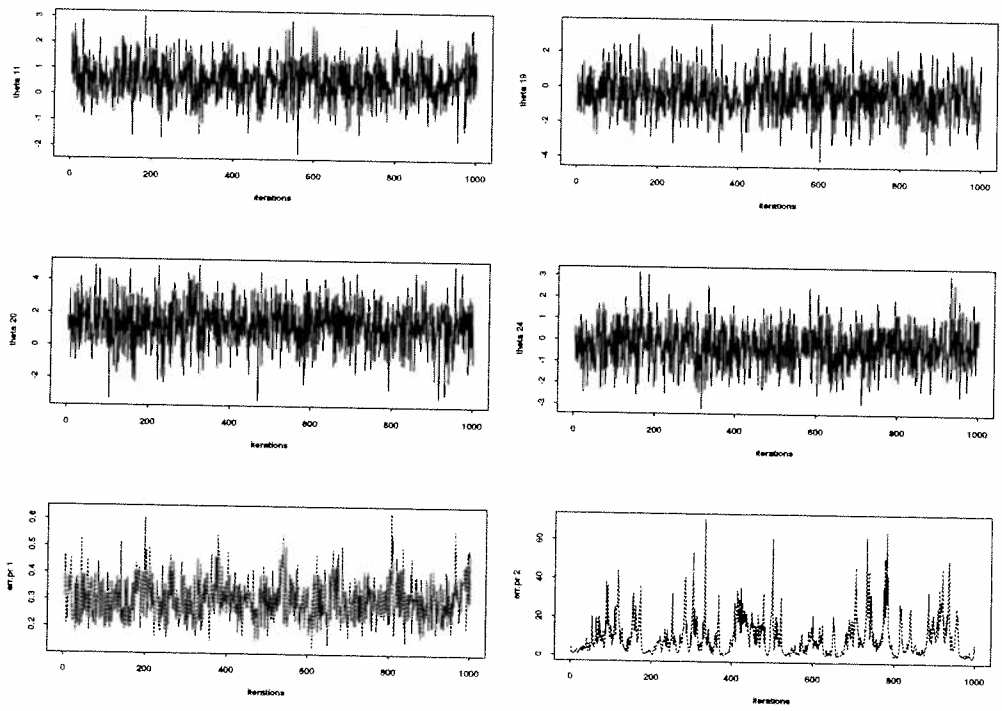

Figure 1: Convergence plots for θ_{32}, β_3, ζ_1, α, and $h_1 = 1/\sigma^2, h_2 = 1/\tau^2$ in the indicator parameterization (plots ordered first from left to right then top to bottom).

One possible concern here is the long upper tail on $1/\tau^2$, visible in the convergence plot.

This suggests that a Gibbs run longer than 1000 iterations may be necessary to reduce variation in the estimates of posterior quantities. We did a run of length 6000 and got essentially the same results reported here.

The scaled design matrix $\Gamma^{1/2}X$ for the indicator parameterization has condition number 14.7, greater than the condition number 5.1 for the orthogonal parameterization, as we expected. Two factors cause the condition number for the indicator parameterization to be greater than the condition number for the orthogonal parameterization. First, the original design matrix X is less "orthogonal" so the eigenvalues are more extreme than for the orthogonal parameterization. Another reason is that the posterior means of error precisions $1/\sigma^2$ and $1/\tau^2$, which form the diagonal elements of Γ, have a bigger range for the indicator parameterization. However, collinearity does not seem to be a problem for the indicator parameterization.

The influence diagnostic shows that few cases are influential. The maxima of the relative change of posterior standard deviations are -2.5 and -2.3 respectively for the indicator and the orthogonal parameterizations. Consistent for both models, it points to the 61-st constraint case. The deletion of the case has the biggest influence on the parameter θ_{52} (the intercept term of the data-case equation for rat number 5 in nicotine treatment group. It is no surprise after we identified the 61-st constraint-case equation as $\theta_{52} = \mu + \alpha + \delta_{52}, \delta_{52} \sim N(0, \tau^2)$ for the indicator parameterization, or $\theta_{52} = \mu - \alpha + \delta_{52}, \delta_{52} \sim N(0, \tau^2)$ for the orthogonal parameterization. Also, the rat number 5 in the nicotine group has the lowest cortical measurement and the next to the lowest adrenal measurement; therefore, the model-fitting has shrunk θ_{52} the most and the deletion of the constraint-case equation would affect

θ_{52} the most.

The residual plot for the indicator parameterization is in Figure 2. The points in the

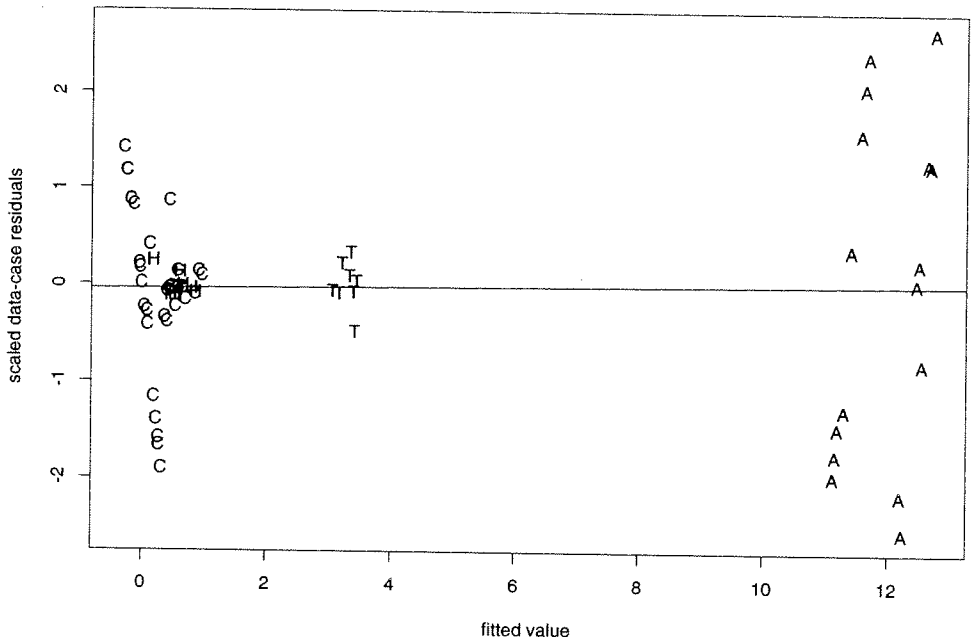

Figure 2: Residual plot for the indicator parameterization. A: adrenal, C: cortical, T: thalamus, H: hippocampus

plot form clusters. The 16 data cases on the right side of Figure 2 have fitted values greater than 10. They come from adrenal tissue of which the measurements for cytisine are all 0 and for epibatidine are relatively large in the data. Their differences as response values are large and so is the variation of the differences. The middle cluster of 8 points come from thalamus tissue. And the cortical and hippocampus tissue form the rest of the data-case points.

The plot shows an increasing spread toward the right, suggesting that a variance stabilizing transformation of the observed may be needed.

The normal error assumption seems to be questionable for the indicator parameterization (see Figure 3). The QQ-plots show that the data-case distribution has long fat tails and an

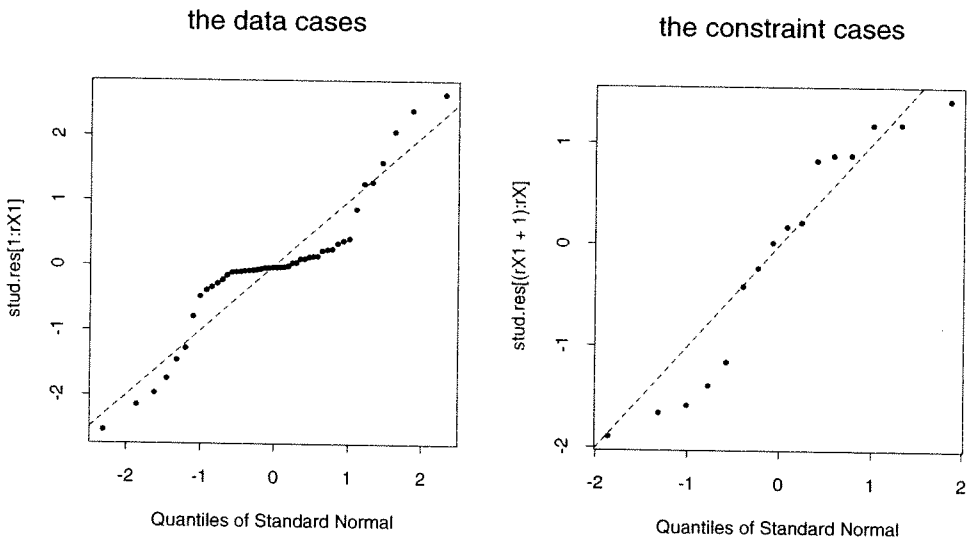

Figure 3: QQ-plots for the indicator parameterization.

elevated center, resembling perhaps some t-distributions with fairly small degrees of freedom and that the constraint cases look more normally distributed.

For the orthogonal parameterization, the residual plot and the QQ-plots are very similar to those for the indicator parameterization.

Of particular interest to us is the added-variable plot. From the data description, it seems that the rat sequence number may play an adverse role in the randomization of this experiment since the investigator has always examined the rats in the same order. To study this, we define external variables which take the values -3.5, -2.5, -1.5, -0.5, 0.5, 1.5, 2.5, and 3.5 for the sequence of the eight rats in the group that we consider and 0 for the eight rats in the other group. With this sequence number as the external variable defined for either control or treatment group, the added-variable plots are drawn (see for example, the

added-variable plot with regard to the control rat sequence in Figure 4 for the orthogonal parameterization). We find that for the orthogonal parameterization the slopes from ordinary

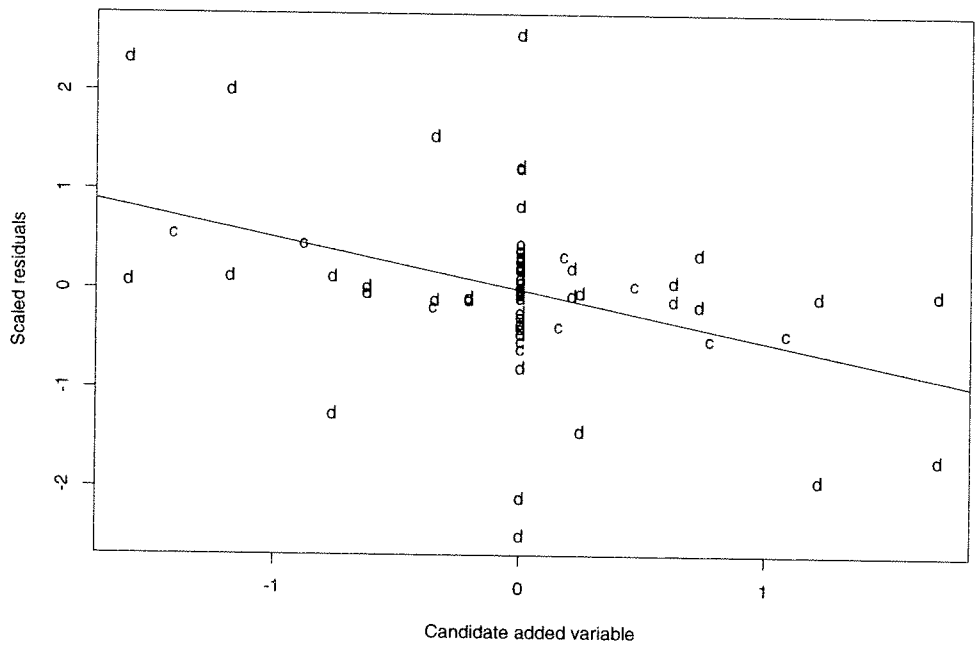

Figure 4: Added variable plot; control rat sequence.

least squares fit are -0.30 and -0.52 with standard errors 0.17 and 0.16 respectively for the sequences of treatment and control rats as an external variable.

For the indicator parameterization, the added-variable plots give almost identical answers: the slope for the treatment group is -0.30 with standard error 0.16. And for the control group the slope is -0.52 with standard error 0.15. Just as in orthogonal parameterization, the added-variable plots show stronger evidence against randomization in control rats. So there seems to be a trend in both groups where later rats tend to have lower counts of type B nicotine receptors. Working in such a fixed sequence, the investigator might have introduced bias into the experiment.

The parameter estimates from both parameterizations are given below:

Parameters	β_1	β_2	β_3	ζ_1	ζ_2	ζ_3	α
Parameterization	Indicator						
Posterior Mean	10.71	2.48	−0.39	1.39	0.29	0.61	−0.27
Posterior Std. Dev.	0.93	1.17	1.17	1.32	1.63	1.63	0.97
Parameterization	Orthogonal						
Posterior Mean	−5.70	1.36	2.22	0.35	0.08	−0.06	−0.15
Posterior Std. Dev.	0.34	0.47	0.31	0.34	0.47	0.30	0.33

Besides the concern for the normality assumption of data-case and for certain bias issue in the experiment, we are fairly assured of the validity of the model from the above diagnostics and from the robustness of the normal tests. From the table, we conclude that not enough evidence is present to believe that the type B receptors expressed in response to nicotine differ from control among different types of neural tissue since the interaction terms are not significant. However, the number of type B receptors is different among different tissues. For example, the average difference between the adrenal and cortical types of tissue is $\beta_1 = 10.71$ (note in the orthogonal parameterization model the corresponding parameter is half of 10.71 and differ by a sign). There does not appear to be a treatment effect since α is not significantly different from 0. The conclusions are partially supported by the residual plot too: the differences of variations between the neural tissues are obvious.

4.2 Jaw Study—Two Models

Despite the complicated model design, the jaw problem fits well into the framework of the model $Y = X\Theta + E$ with X and Y in (5). The big dimensions of 161 rows and 107 columns of the design matrix X make the Gibbs sampler *gibbs* slow to run for 1000 iterations after discarding 100 iterations. However the convergence in $\theta_{i,k}$'s and μ_k's appears to be immediate, as suggested by the two top plots in Figure 5. As in the rats example,

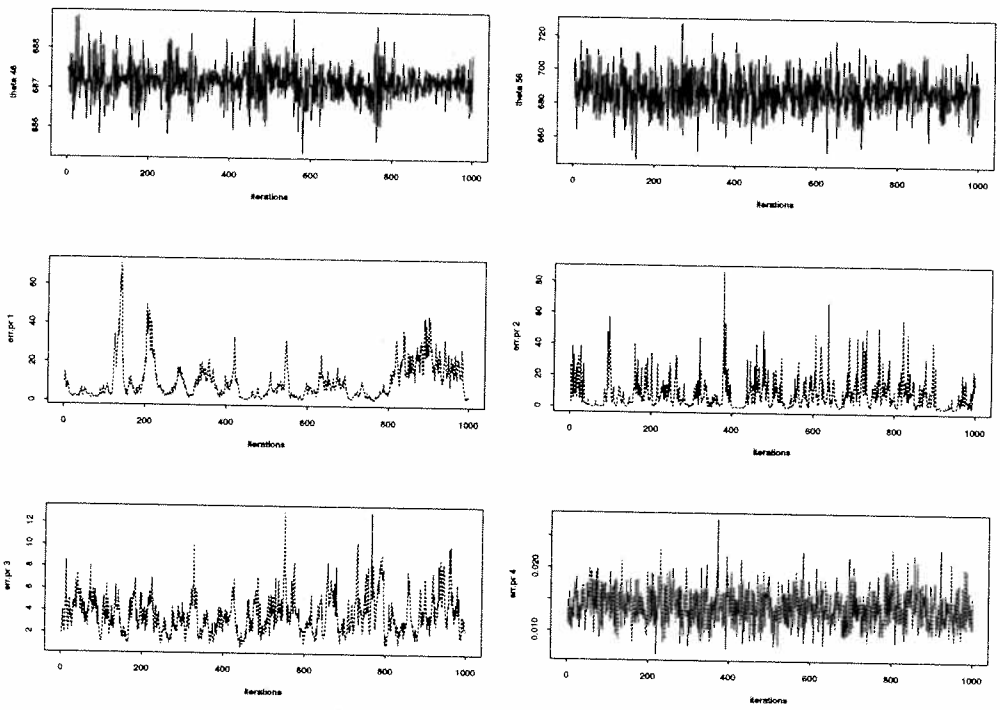

Figure 5: Convergence Plots for θ_{46}, μ_1, $1/\sigma^2$, $1/\eta^2$, $1/\tau^2$, and $1/\phi^2$.

some of the precisions have long upper tails, especially $1/\sigma^2$, $1/\eta^2$, and to a lesser extent, $1/\tau^2$. Again, a longer run might be desirable.

The error precision $1/\phi^2$ has a distribution close to zero while the other precisions are notably larger. The prior and the estimated posterior mean, variance, and the standard deviation of the error precisions are given in the following table:

	Prior	Posterior			
	$1/\sigma^2, 1/\eta^2, 1/\tau^2, 1/\phi^2$	$1/\sigma^2$	$1/\eta^2$	$1/\tau^2$	$1/\phi^2$
Mean	11.0	10.84	9.30	3.76	0.01
Var.	110.0	102.82	105.94	3.39	0.00
Std. Dev.	10.5	10.14	10.29	1.84	0.00

The table indicates that the two error precisions $1/\sigma^2, 1/\eta^2$ gain little information from the data set since the means and variances of their distributions change only slightly from prior to posterior. The error precision $1/\eta^2$ does not change much since there are only 5 constraint cases—the equations in 9—relating to it. And given the small posteriors of error precisions $1/\tau^2$ and $1/\phi^2$ the model does not have much information with which to change $1/\sigma^2$.

The small mean and variance of the error precision $1/\phi^2$ implies that the underlying mean strain as a function of the displacement is rugged. See Figure 6, where the estimated mean function is plotted against displacements along with the five repetition's draws. There we see that the mean function is very rough and all strain functions are close to the data, which implies that the model has done little smoothing to the data. The model, closely fitting the data, does not bind the mean principal strains tightly between different displacements, i.e., the precision $1/\phi^2$ is small so the μ_k's need not be near each other. One possible reason for this is that the 55 data cases are too few in comparison with 107 parameters to estimate.

The scaled design matrix of the model has a condition number of 256. The big condition number is somewhat expected. For all iterations, the condition numbers are between 48 and 384. So collinearity is a big problem, which may cause imprecision of some of the parameter estimates, especially those in the second level. This may be the reason why the standard

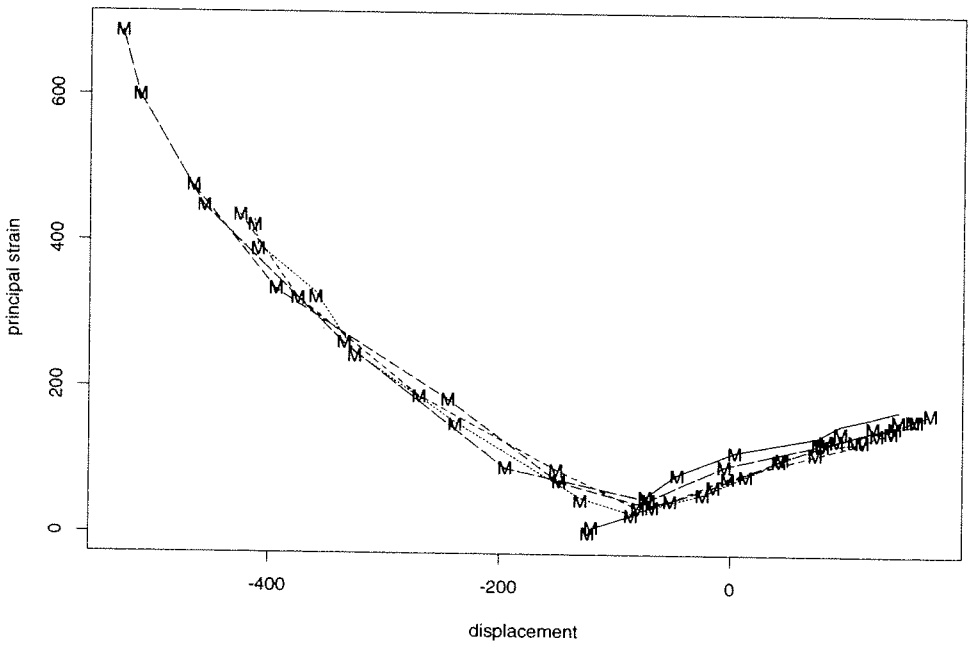

Figure 6: Mean principal strain as a function of displacement with plotting character "M" at the function values; the five repetition's draws are plotted as curves

deviations of the most estimated μ's vary between 3 to 11, except for the middle μ_{30}–μ_{35} which tie to the $\theta_{i,k}$ by the second-level equations in (9). The standard deviations of μ_{30}–μ_{35} range only from 1.07 to 1.57.

The data suggest that the strain measurements in the first repetition may be different from the ones in other repetitions. We considered an added-variable plot with a constant shift added to the first repetition as an external variable. This is equivalent to changing the first equation in (9) to:

$$\theta_{1,33} = c + \mu_{33} + \nu_1 \tag{12}$$

with an unknown constant c as the external variable. The ordinary least squares fit to the added-variable plot (Figure 7) shows a slope of 32.2 with standard error 20.9; therefore, the

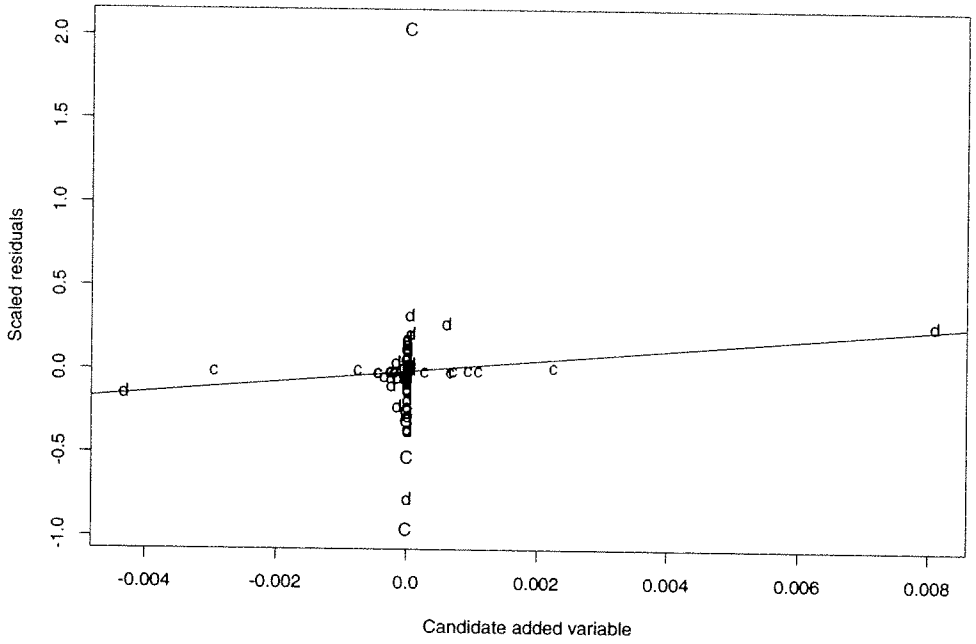

Figure 7: Added-variable plot for adding a constant shift in Rep 1.

first repetition probably is not shifted relative to the other repetitions. The capital letter C in the plot represents the 5 constraint cases from (9) and all the values of the added variable corresponding to the other constraint cases are close to zero except the ones relating to μ_{32} and μ_{33} in (11). The outlier with the greatest scaled residual 2.0 is the constraint case 56; it does not have too much effect on the slope 32.2 since it is outlying in the vertical direction. Other influential points in the plot are all related to parameters that appear in the first repetition. The points having the maximum and minimum values of the candidate added variable correspond to the two equations involving μ_{33} in the 51 equations of (11). After deleting the point with the maximum value of the candidate added variable, the least squares fit to the rest gives a slope of 25.7 and standard error of 34.8. So the apparent increasing trend depends on the presence of the outliers. All the data cases are blended into the middle

cloud of the plot and are not distinguishable.

The influence diagnostic shows that eliminating the constraint case 56 or, equivalently, the first equation in (9) has the greatest effect on parameters by the linear influence estimation. In fact, its deletion from the model would cause the posterior mean of μ_{33} to decrease by 136 posterior standard deviations or 214 micro-strain units and estimates of most other parameters of μ (except $\mu_{30}, \mu_{31}, \mu_{32}, \mu_{34}, \mu_{35}$) decrease by 2 to 29 posterior standard deviations. The influence for a few of these cases is perhaps exaggerated since the linear influence approximation may overstate the effect when the change from a case deletion is large (see Hodges [**H**], Section 4.4). This is, however, consistent with our visual observation of the data and from the first equation in (9) about μ_{33}. Since the measurements in the first repetition have relatively large displacements and take relatively large values of strain, deleting the equation would pull most of the μ's toward the negative direction.

Another big influence is from the observation 687.2 at the smallest displacement of -528 corresponding to the data case 46. Deleting it would decrease the posterior mean of the parameter θ_{46} by about 86 posterior standard deviations or about 38 micro-strain units.

The residual plot for data cases points to the 46-th observed value as a moderate outlier. In Figure 8 the outlier is marked by a capitalized D. The residual plot for the data cases is typical of what a good least squares fit may give. The slope 0.0004 (with standard error 0.004) is not significant. This indicates that there is not much shrinkage in this fit (see Hodges Section 4.5).

The largest data-case studentized residual is only 1.19 in contrast to the biggest constraint-case studentized residual of 40.9 (constraint case 56) corresponding to the first equation in

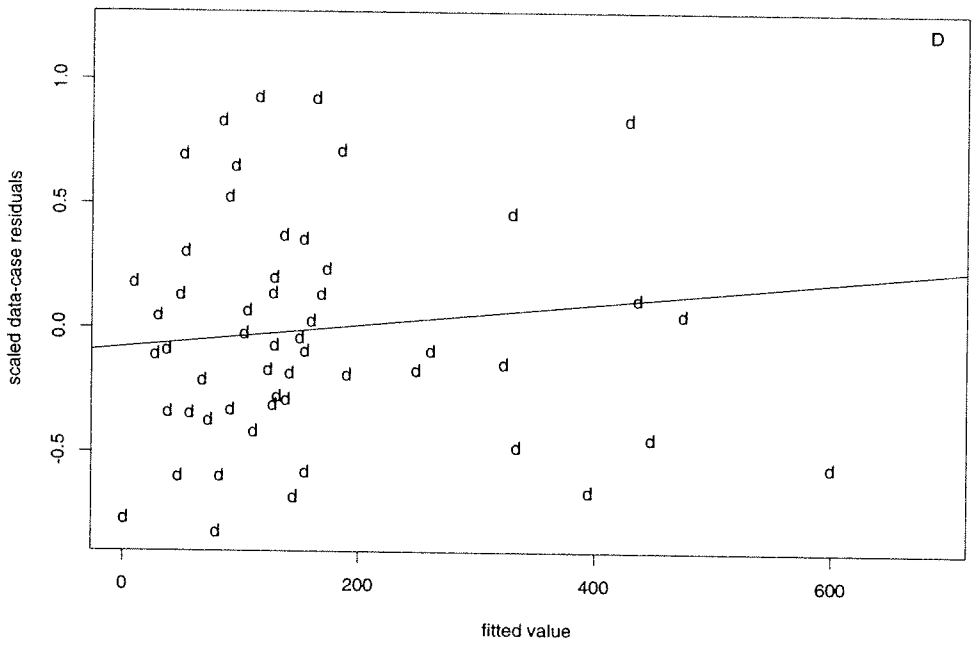

Figure 8: Residual plot for the first parameterization in the jaw study; d: data; c: constraints.

(9) for constraint cases. This shows that the model has a good fit to the data cases and that the equations in (9) do not have much influence on the model. The model-fitting process chooses to ignore them, therefore making them outliers.

The normality assumption seems reasonable for the data cases (see Figure 9), although the variance of the data-case studentized residuals seem to be too small. The same outlier (constraint case 56) appears in the constraint-case QQ-plot (the upper-right corner of the plot) and other four outliers near the lower left corner correspond to the rest of the five equations in (9). Apart from these 5 outliers, the constraint-case distribution is reasonably normal.

The transformation diagnostic suggests that a transformation of y_j to y_j^{λ} with $\lambda = 0.8$ (standard error 0.005) is helpful in modeling the data (see Figure 10). Notice that all the

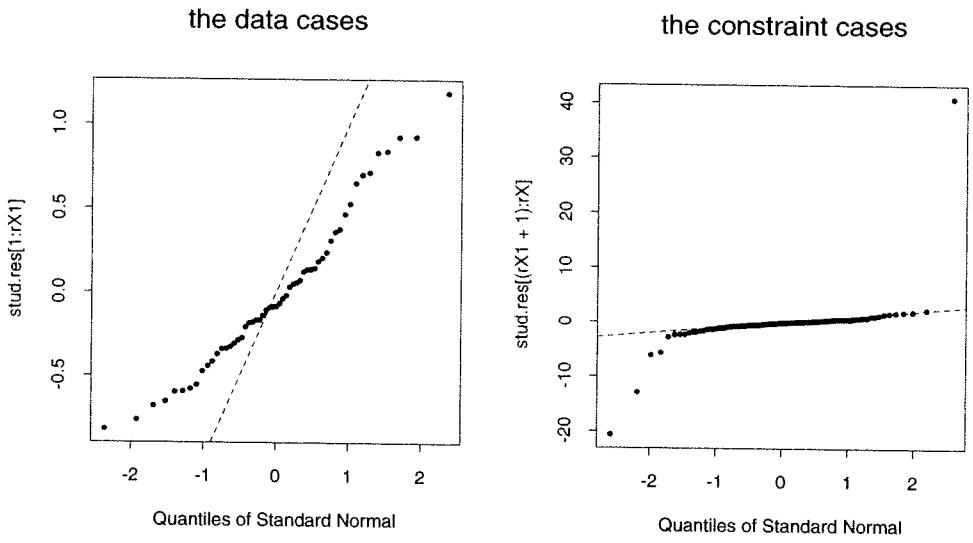

Figure 9: QQ plots for data cases and constraint cases.

data cases in the plot form a small pile in the middle, which suggests that the transformation reduces mostly the constraint-case variation.

Overall, the model seems to fit the data quite well in the level corresponding to data cases but does a poor job for constraint cases. It is clearly seen from Figure 6 that the mean function for the repetitions μ_j is not smoothed much by the model-fitting process. One reason is that the equations in (11) used to smooth the mean function tend to model the mean function either close to a constant or else as a fairly rugged function. To relax the restrictions by the equations in (11) we may replace the difference equations of the first degree in (11) by the difference equations of the second degree

$$\mu_{k+1} - \mu_k = \mu_k - \mu_{k-1} + \xi_k, \qquad \xi_k \sim N(0, (d_{k+1} - d_{k-1})\phi^2), \quad k = 2, \ldots, 51. \tag{13}$$

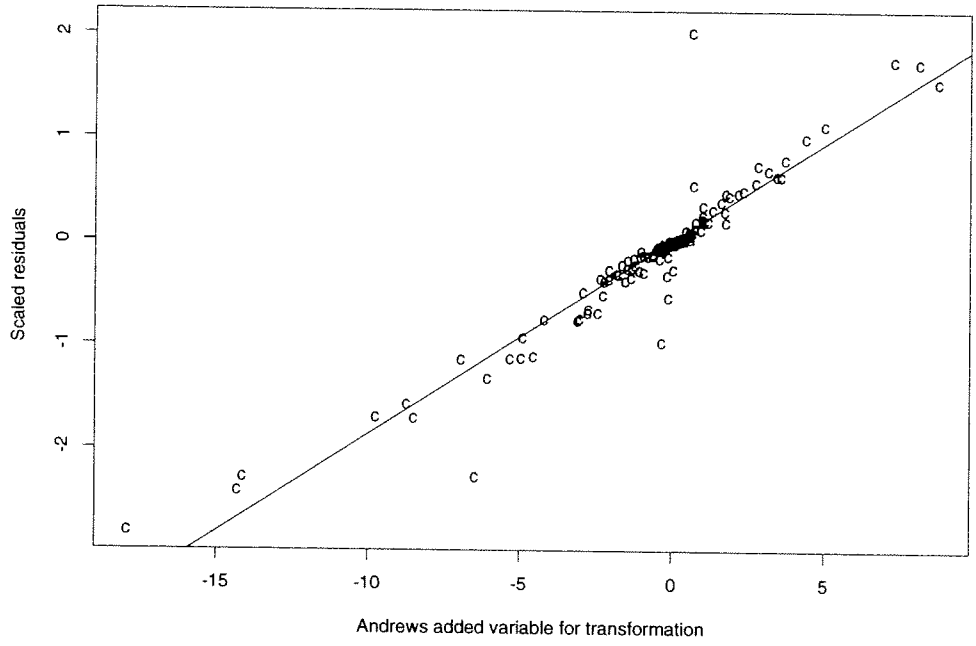

Figure 10: Transformation plot for the first model of the jaw study; $\lambda = .8$ and std.err. is .005

Namely,

$$\mu_{k+1} - 2\mu_k + \mu_{k-1} = \xi_k, \qquad \xi_k \sim N(0, (d_{k+1} - d_{k-1})\phi^2), \quad k = 2, \ldots, 51. \tag{14}$$

The equations in (14) penalize deviations from linearity, instead of deviations from constancy.

We set up the design matrix X as before, which now has one less row than for the first model. For the new analysis we chose a different Gamma prior for the error precision $1/\phi^2$. The new prior distribution of $1/\phi^2$ has shape parameter 10 and scale parameter 1, so the mean and the variance are both 10. By letting the prior variance of the error precision $1/\phi^2$ be smaller, the posterior distribution of $1/\phi^2$ is less influenced by the data. Therefore, the error precision $1/\phi^2$ should not be too small, as in the first model. Thus we hope that the

mean curve of μ is smoother than in the original fit. We then ran the Gibbs function with a) 1100 iterations discarding the first 100 iterations; and b) 3000 iterations discarding the first 1000 iterations.

The convergence plots for the *gibbs* function seem to have unusual patterns. The posterior distributions for some of the parameters and the error precisions appear to have some quite complicated yet interesting behaviors (Figure 11). The convergence for $1/\phi^2$ is evident. After

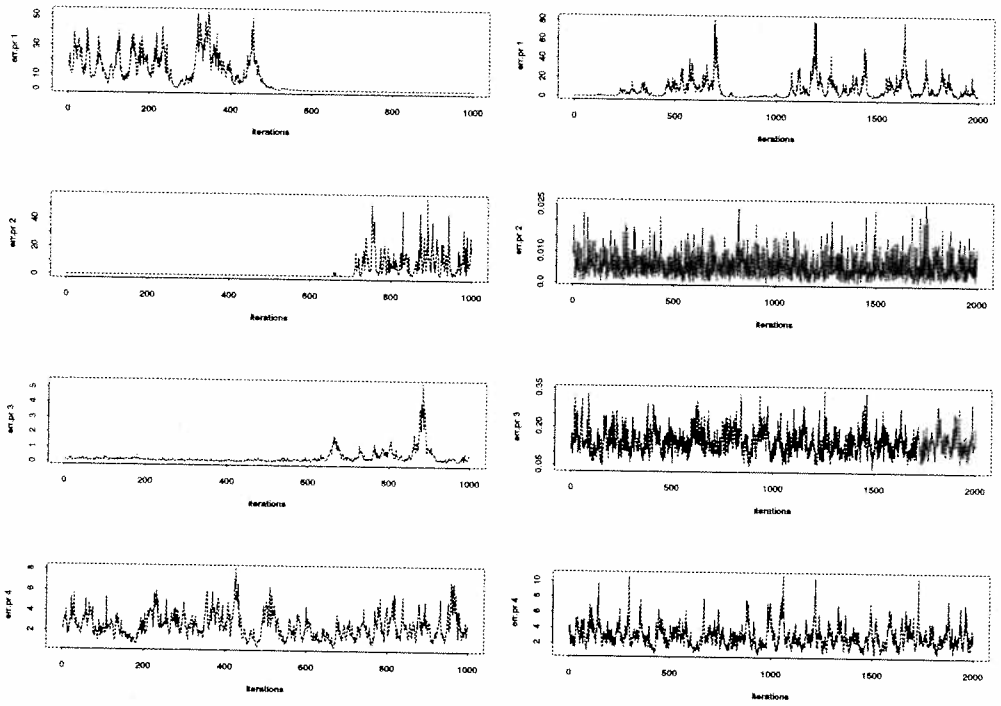

Figure 11: Convergence plots for $1/\sigma^2, 1/\eta^2, 1/\tau^2, 1/\phi^2$ from runs a) (left) and b) (right).

some 600-700 initial draws that are close to zero, the error precisions of $1/\eta^2$ and $1/\tau^2$ start to oscillate but the runs in b), with 3000 draws and the first 1000 of them discarded, show a uniform convergence to somewhere near zero. The draws for $1/\sigma^2$ also have such behavior of pattern-changing. The posterior distribution of μ-parameters all seem to have limits. The 55 θ-parameters have their draws close to their convergent means with fits of sudden

oscillations. One possible reason for such strange behavior is that the posterior distributions are quite flat which makes the Markov chain unstable. Another 4000 iterations show that the draws for $1/\sigma^2$ have occasional excursions from the apparent centre of the distribution. Other error precisions have basically benign convergence behavior. More runs are needed to assure if the sudden oscillations for $1/\sigma^2$ are persistent. If not, then the initial loitering in other neighbourhoods may come from the fact that the posterior distribution of $1/\sigma^2$ is multimodal or has a long tail.

For the sake of brevity, next we study the diagnostics of this problem, although many the convergence issues above need to be further explored. We shall concentrate on the model results from run b) as it is likely to be more accurate.

The table below gives the prior and the posterior mean, variance, and standard deviation of all error precisions after fitting by the Gibbs sampler. We have also included the results from the first model.

		Prior		Posterior			
		$1/\sigma^2$ $1/\eta^2$ $1/\tau^2$	$1/\phi^2$	$1/\sigma^2$	$1/\eta^2$	$1/\tau^2$	$1/\phi^2$
Second	Mean	11.0	10.0	8.28	0.01	0.15	3.03
Model	Var.	110	10.0	106.47	0.00	0.00	1.87
	Std. Dev.	10.5	3.16	10.32	0.00	0.05	1.37
First	Mean	11.0		10.84	9.30	3.76	0.01
Model	Var.	110.0		102.82	105.94	3.39	0.00
	Std. Dev.	10.5		10.14	10.29	1.84	0.00

From the table we see that the posterior mean of the error precision $1/\phi^2$ is bigger: 3.03,

compared to 0.01 in the first model. Hence the μ's among different displacements are more correlated; the mean curve of μ is smoother. On the other hand, since $1/\sigma^2$ has a smaller posterior mean we see that the model at the first level does not fit the data as closely as the previous model does. There is still not much information gained for the posterior distribution of $1/\sigma^2$ since the posterior mean and variance are close to the prior mean and variance, as in the first model. We also note that the posterior mean of $1/\eta^2$ has decreased close to 0. This implies that the new model does not require that each repetition's curve be anchored close to the mean curve μ at the smallest displacement. The smaller posterior mean for $1/\tau^2$ in this model implies less similarity between the mean curve of μ and the five draws of θ compared to the first model.

The condition number of the scaled design matrix is 356, bigger than the condition number in the previous model. The range of condition numbers at all 2000 iterations is between 89 and 996, probably indicating the difficult problem of convergence.

With a constant shift in the first repetition as an external variable, the added-variable plot (not shown) gives a slope from an ordinary least squares fit of 33 (standard error 5) which suggests that a constant shift in the first repetition helps to explain the variation of the data. However, the added-variable plot shows that the constraint-case 56 (corresponding to the first equation in (9)) as an outlier. So the significance of slope may be due to a few outliers. The case-influence diagnostic selects data case 48 at $d_4 = -465.8$ in repetition 5 as the most influential. The deletion of the case would cause the parameter θ_{48} to increase by 99 posterior standard deviation or 136 micro-strain units.

The residual plot (Figure 12) presents the data case 46 at the smallest displacement

$d_1 = -528.0$ in repetition 5 as an outlier.

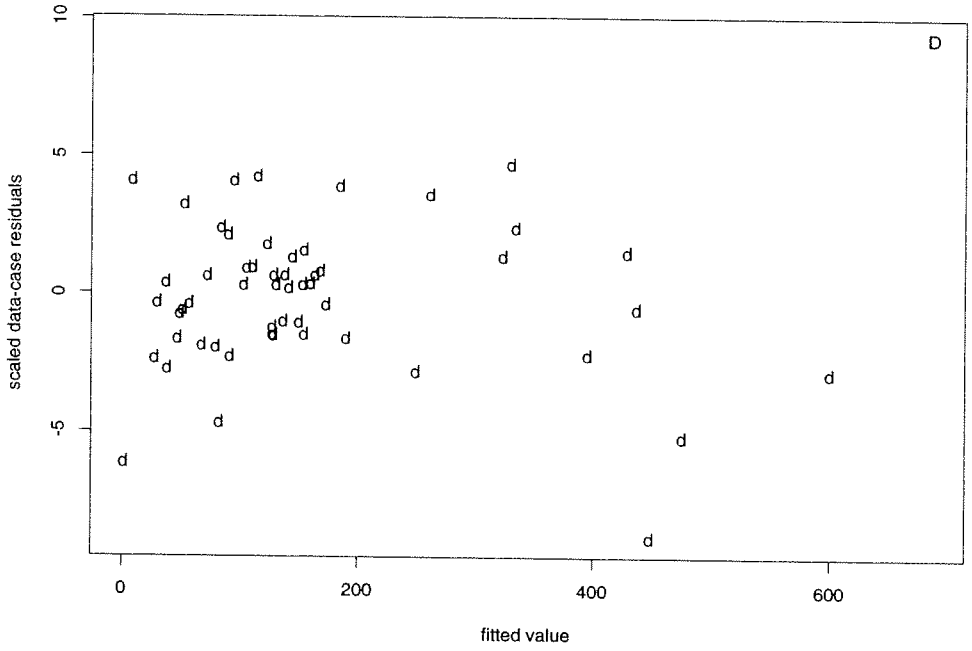

Figure 12: Residual plot. D: case 46; d: data.

Compared with Figure 9, we see that the normality assumption of the studentized residuals seems more reasonable for the constraint cases than in the first model (see Figure 13), although the plot shows a long lower tail. For the data cases, the distribution of the residuals still seems normal, but now its variance is larger.

The transformation diagnostic again gives $\lambda = 0.8$ (standard error 0.005, see Figure 14).

Finally, the estimated mean functions (Figure 15) from the modified model appear smoother than in the previous model (Figure 6).

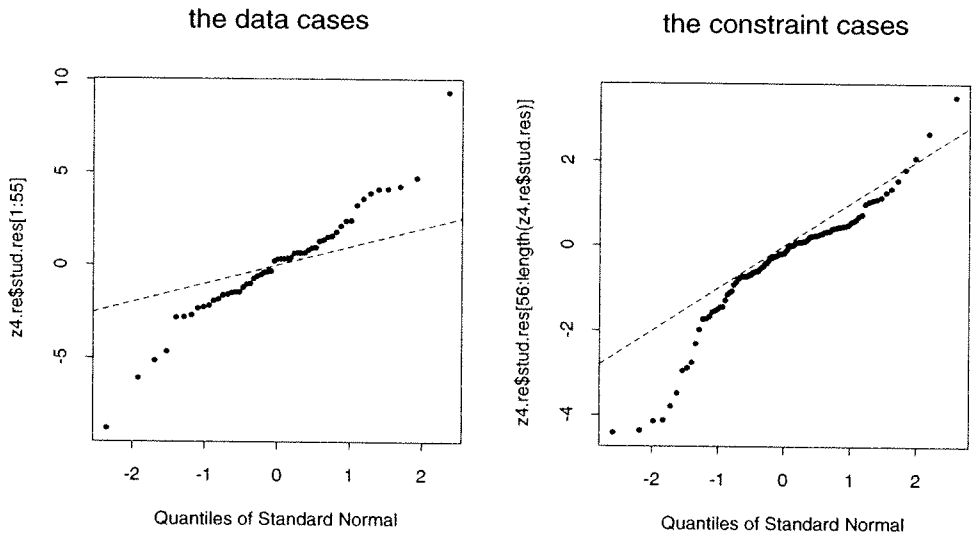

Figure 13: QQ plots for data cases and constraint cases.

5 Future Work

Despite the advance of computer technology, hierarchical model fitting, as described in this article, remains a difficult problem. One reason is that the set-up of the model usually involves matrices of huge sizes, as seen in the examples given, which renders it computationally intensive. It is often a difficult question of balancing, for example in writing the *gibbs* function, between computing speed and generality. To achieve better efficiency, future computing work should include rewriting the *gibbs* function as object code using a lower-level computer language such as C++ or Fortran. There is also much maintenance work to be done for all the functions, such as adding the argument matching features so they become more user-friendly. After such improvements so the model-fitting and diagnostic programs become more versatile, properties of hierarchical modeling with the Gibbs sampler can be

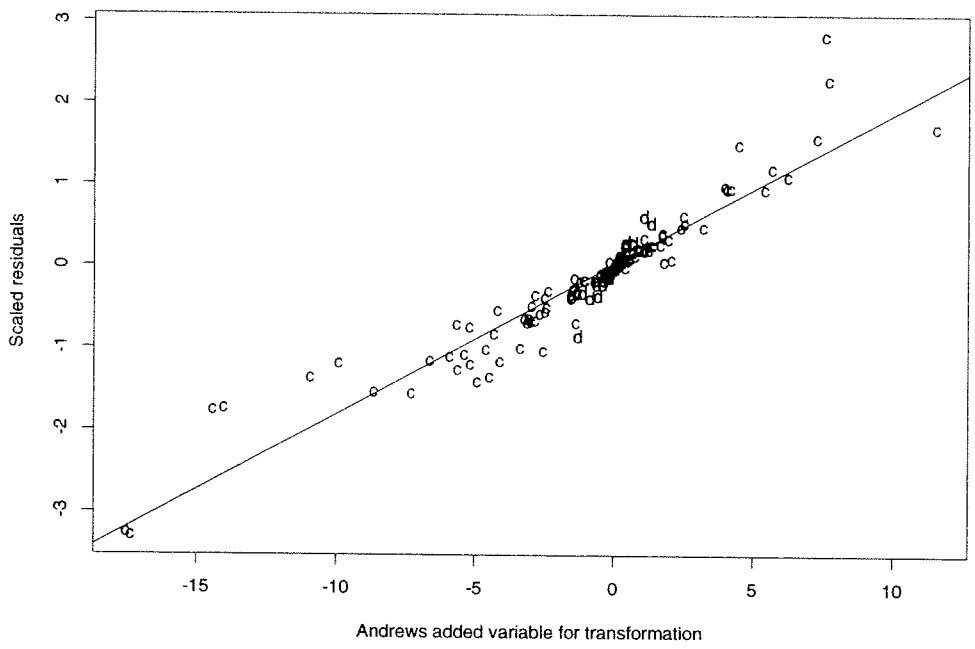

Figure 14: Transformation plot for the jaw study; The second model.

further studied and it is then possible to generalize the functions and to test more extensively by working with data sets of various structures. In particular, another model specification function is needed that takes S+ model language formulae as inputs and then constructs X, Y and other inputs to *gibbs*, and finally calls *gibbs*. All of this seems to be promising future work.

In the direction of hierarchical model-fitting, much exploration is needed regarding the models that are designed. In the jaw data, we think that with more repetitive data at the same displacements, the model can be better assessed. In all the analyses, we have used only a few prior distributions of error precisions. It is interesting to see how the model-fitting would be affected by the change of the prior information; much more work needs to be done before this is understood.

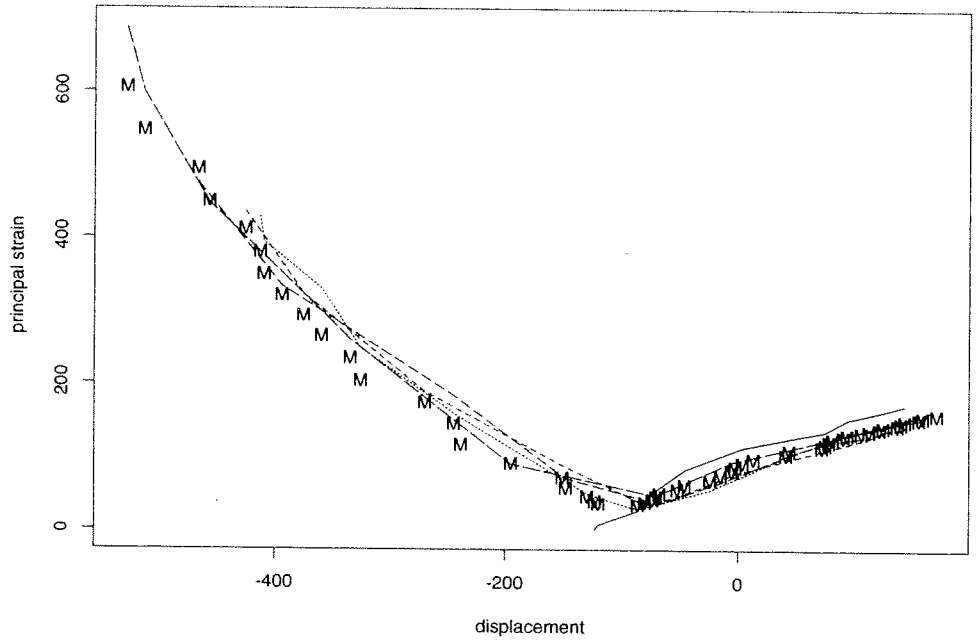

Figure 15: Mean principal strain as a function of displacement with plotting character "M"; Repetition's draws are plotted as curves for the second model.

As an example of the remaining model design questions, consider the second assumption on Page 21 which stipulates that, at only the smallest absolute value of displacements, the value of each repetition's smooth function is the mean function value plus an error. This and the third assumption there would imply that at all other displacements, the values of each repetition's smooth function is also the mean function value plus an error which has bigger variation. For instance, at the displacement $d_{20} = -123.6$, we easily conclude from equations in (9) and (10) that

$$\begin{aligned} \theta_{1,33} &= \mu_{33} + \nu_1, \qquad \nu_1 \sim N(0, \eta^2) \\ \theta_{1,33} - \theta_{1,20} &= \mu_{33} - \mu_{20} + \delta' \end{aligned}$$

where $\delta' \sim N(0, (d_{33} - d_{20})\tau^2)$. Therefore,

$$\theta_{1,20} = \mu_{20} + \nu_1 - \delta' = \mu_{20} + \nu', \qquad \nu' \sim N(0, \eta^2 + (d_{33} - d_{20})\tau^2) \tag{15}$$

Perhaps it is also reasonable to require that each repetition's function value relate to the mean function value at any displacement in exactly the same fashion. Namely, the five equations in (9) are replaced by

$$\theta_{I(j),K(j)} = \mu_{K(j)} + \nu_j, \qquad \nu_j \sim N(0, \eta^2). \tag{16}$$

Then we might expect to see that each repetition's function approximate the mean function more uniformly.

6 Appendix A: S+ programs

Included here are the S+ programs of the seven functions: *gibbs, conv, addvar, colli, influ1, resd*, and *transf*.

- *gibbs*:

```
gibbs_ function(X, Y, rows, cols, num.iter = 1000,
    opt = 0, ptab.len = rows,
    pta = rep(1.1, length(rows)), ptb = rep(.1,length(rows)),
    prior.var  = 1000000, g.diag = rep(1.0,nrow(X)),
    burnin = 100)
{
    if (opt != 0 && opt != 1 ) stop("\nopt can only be 0, 1")
    num.err_ length(ptab.len)
    if (num.err != length(pta) || num.err != length(ptb) )
        stop("\npta, ptb, and ptab.len have different lengths")
    rX_ dim(X)[1]; cX_ dim(X)[2]
    pri.var.len_ length(prior.var)
```

```
c.err.len_ sum(ptab.len)
if (pri.var.len != 1 && (pri.var.len + c.err.len) != rX)
    stop("\ntoo many or too few error terms")
ran.err_ rgamma(num.err,pta)/(ptb+10^(-8))
err_ rep(1/prior.var, rX-c.err.len)
for (i in num.err:1) err _ c(rep(ran.err[i],ptab.len[i]),err)
gamm_ g.diag * err
ahit_ ptab.len/2+pta
hit_ lhit_ rep(NA,num.err)
i _ 1
while (i <= burnin) {
    sv   _ svd(t(X * gamm) %*% X)
    Vhit _ sv$u %*% (t(sv$v) * (1/sv$d))
    eThit_ Vhit %*% (t(X) %*% (gamm * Y))
    Thit _ sv$u %*% ((t(sv$v) * sv$d^(-.5))
        %*% as.matrix(rnorm(cX))) + eThit
    resd _ g.diag * (Y - X %*% Thit)^2
    ix1_ 0
    for (j in 1:num.err) {
        ix2_ ptab.len[j] + ix1
        lhit[j]_ ptb[j] + sum(resd[(ix1+1):ix2])/2
    ix2_ ix1
    }
hit_ rgamma(num.err,ahit)/lhit
err_ rep(1/prior.var, rX-c.err.len)
for (j in num.err:1) err_ c(rep(hit[j],ptab.len[j]),err)
gamm_ g.diag * err
i _ i+1
}
lhit_ matrix(NA,num.iter,num.err)
eThit_ vThit_ matrix(NA,num.iter,cX)
if (opt==1) {
    Thit _ matrix(NA,num.iter,cX)
    hit_ matrix(NA,num.iter,num.err)
i_ 1
while (i <= num.iter) {
    sv_ svd(t(X * gamm) %*% X)
    Vhit_ sv$u %*% (t(sv$v) * (1/sv$d))
    vThit[i,]_ diag(Vhit)
    eThit[i,]_ Vhit %*% (t(X) %*% (gamm * Y))
    Thit[i,]_ sv$u %*% ((sv$d^(-.5) * t(sv$v))
                %*% as.matrix(rnorm(cX))) + eThit[i,]
    resd_ g.diag * (Y - X %*% Thit[i,])^2
    ix1_ 0
    for (j in 1:num.err) {
        ix2_ ptab.len[j] + ix1
        lhit[i,j]_ ptb[j] + sum(resd[(ix1+1):ix2])/2
        ix1_ ix2
        }
```

```
    hit[i,] _ rgamma(num.err,ahit)/lhit[i,]
    err_ rep(1/prior.var, rX-c.err.len)
    for (j in num.err:1) err_ c(rep(hit[i,j],ptab.len[j]),err)
    gamm_ g.diag * err
    i _ i+1
    }
eth _ apply(eThit,2,mean)
vth _ apply(vThit,2,mean) + apply(eThit,2,var)
ehe _ apply(ahit/t(lhit),1,mean)
vhe _ apply(ahit/t(lhit)^2,1,mean) + apply(ahit/t(lhit),1,var)
esig _ apply(t(lhit)/(ahit-1),1,mean)
vsig _ apply(t(lhit)^2/((ahit-1)^2*(ahit-2)),1,mean) +
apply(t(lhit)/(ahit-1),1,var)
z_ list(
    Y = Y, X = X, num.iter = num.iter,
    rows = rows, cols = cols, ptab.len = ptab.len,
    mean.theta = eth, var.theta = vth,
    ehe = ehe, vhe = vhe,
    mean.sig2 = esig, var.sig2= vsig,
    Ghe = gamm^.5, g.diag = g.diag,
    Gibbs.iter=list(thit=Thit,hit=hit)
    )
z
}
else {
i _ 1
while (i <= num.iter) {
    sv_ svd(t(X * gamm) %*% X)
    Vhit_ sv$u %*% (t(sv$v) * (1/sv$d))
    vThit[i,]_ diag(Vhit)
    eThit[i,]_ Vhit %*% (t(X) %*% (gamm * Y))
    Thit_ sv$u %*% ((t(sv$v) * (sv$d^(-.5)))
        %*% matrix(rnorm(cX),byrow=T)) + eThit[i,]
    resd_ Y - X %*% as.matrix(Thit)
    resd_ g.diag * resd^2
    ix1_ 0
    for (j in 1:num.err) {
        ix2_ ptab.len[j] + ix1
        lhit[i,j]_ ptb[j] + sum(resd[(ix1+1):ix2])/2
        ix1_ ix2
    }
    hit_ rgamma(num.err,ahit)/lhit[i,]
    err_ rep(1/prior.var, rX-c.err.len)
    for (j in num.err:1) err_ c(rep(hit[j],ptab.len[j]),err)
    gamm_ g.diag * err
    i _ i+1
}
eThit_ eThit[1:num.iter,]
vThit_ vThit[1:num.iter,]
```

```
    lhit _ lhit[1:num.iter,]
    eth _ apply(eThit,2,mean)
    vth _ apply(vThit,2,mean) + apply(eThit,2,var)
    ehe _ apply(ahit/t(lhit),1,mean)
    vhe _ apply(ahit/t(lhit)^2,1,mean) + apply(ahit/t(lhit),1,var)
    esig _ apply(t(lhit)/(ahit-1),1,mean)
    vsig _ apply(t(lhit)^2/((ahit-1)^2*(ahit-2)),1,mean) +
    apply(t(lhit)/(ahit-1),1,var)
    z_ list(Y = Y, X = X, num.iter = num.iter,
        rows = rows, cols = cols, ptab.len = ptab.len,
        mean.theta = eth, var.theta = vth,
        ehe = ehe, vhe = vhe,
        mean.sig2 = esig, var.sig2 = vsig,
        Ghe = gamm^.5, g.diag = g.diag,)
    z
    }
}
```

- *conv*:

```
conv_ function(model, theta.idx = 0, error.idx = 0)
{
    if (is.null(model$Gibbs.iter))
        stop("\nmodel input not from gibbs(...) with opt=1")
    Gibbs_ model$Gibbs.iter
    iterations_ c(1:length(model$Gibbs.iter$thit[,1]))
    ix1_ theta.idx[theta.idx > 0 & theta.idx <= length(Gibbs$thit[1,])]
    ix2_ error.idx[error.idx > 0 & error.idx <= length(Gibbs$hit[1,])]
    if ( length(ix1) != 0 || length(ix2) !=0) {
        par(mfrow=c(3,2))
        for (i in 1:(length(ix1) + length(ix2))) {
            if (i != 1 && (i-1) %% 6 == 0) {
                cat("continue plotting (y/n) ?\t")
                ans_ scan(what = character(),n=1)
                if (ans != "Y" && ans != "y" &&
                    ans != "Yes" && ans != "yes") stop()
                par(mfrow=c(3,2),ask = F)
            }
            if (i <= length(ix1)) {
                plot(iterations,Gibbs$thit[,ix1[i]],xlab="iterations",
                        ylab=paste("theta", ix1[i]),type="l",lty=1)
            }
            else {
                plot(iterations,Gibbs$hit[,ix2[i-length(ix1)]],
                    xlab="iterations",
                    ylab=paste("err.pr", ix2[i-length(ix1)]),type="n")
                points(iterations,Gibbs$hit[,ix2[i-length(ix1)]],
                                                    type="l",lty=2)
```

```
            }
        }
    }
}
```

- *addvar*:

```
addvar_ function(model, addv1 = rep(0,model$rows[1]),
                    addv2 = rep(0,sum(model$ptab.len) - model$rows[1]),
                    plotit = F,
                    labs = c(rep("d",model$rows[1]),
                     rep("c",sum(model$ptab.len)-model$rows[1]),
                     rep("p",dim(model$X)[1]-sum(model$ptab.len))) )

{
    rX1_ model$rows[1]; X _ model$X; rX_ dim(X)[1]
    if (length(addv1) > rX1 || length(c(addv1,addv2)) > rX)
        stop("\nadded variable doesn't have the right length.")
    Y _ model$Y
    G _ model$Ghe * model$g.diag
    E _ Y - X %*% as.matrix(model$mean.theta)
    sE_ as.vector(G * E)
    temp_ svd(X * G)
    V _ temp$u %*% t(temp$u)
    addv2_ c(rep(0,rX1),addv2,rep(0,rX-length(addv2)-rX1))
    B_ as.matrix(c(addv1,rep(0,rX-rX1)))+ as.matrix(addv2)
    sB _  B * G
    avB _ as.vector((diag(rX) - V) %*% sB)
    av.ls_ lsfit(avB,sE)
    if (plotit) {
        plot(avB, sE, xlab="Candidate added variable",
                ylab="Scaled residuals", type="n")
        if (all(addv1==0) && all(addv2==0))
            warning("added variable has all zero values")
        text(avB, sE, labs, cex = 1)
        abline(av.ls)
    }
    av.lsp_ ls.print(av.ls,print.it=F)
    z_ list(avB = as.vector(avB), sE = as.vector(sE),
            slope = av.lsp$coef.table[2,1],
            std.err = av.lsp$coef.table[2,2],
            p.value = av.lsp$coef.table[2,4],)
    z
}
```

- *colli*:

```
colli_ function(model,opt = 0,
                v.iter = c(1:length(model$Gibbs.iter$hit[,1])))
{
    if (opt!=0 && opt!=1 )
        stop("\nopt can only be 0, 1")
    g.diag_ model$g.diag
    if (opt==0) {
        svdX _ svd(model$X * sqrt(model$Ghe * g.diag))
        cond_ ifelse(svdX$d[length(svdX$d)],
        svdX$d[1]/svdX$d[length(svdX$d)], "infinity")
    return(cond)
    }
    else {
        if (is.null(model$Gibbs.iter))
            stop("\nmodel input not from gibbs(...) with opt=1")
        gibbs.len_ length(model$Gibbs.iter$hit[,1])
        v.iter_ v.iter[v.iter<=gibbs.len]
        cond.len_ min(length(v.iter),gibbs.len)
        cond_ rep(NA,cond.len)
        d.c.len_ dim(model$X)[v.iter[1]]
        c.err.len_ sum(model$ptab.len)
        for (i in 1:length(v.iter)) {
            err_ NA
            for(j in 1:length(model$ptab.len))
                err <- c(err, rep(model$Gibbs.iter$hit[v.iter[i],j],
                                  model$ptab.len[j]))
            err _ err[!is.na(err)]
            G_ sqrt(g.diag[1:length(err)] * err)
            if (length(G) < length(g.diag))  G_
                    c(G, model$Ghe[(length(G)+1):length(g.diag)])
            svdX _ svd(model$X * G)
            cond[i]_ ifelse(svdX$d[length(svdX$d)],
                    svdX$d[1]/svdX$d[length(svdX$d)], "infinity")
        }
        v.c_ list(v.iter = v.iter, cond = cond)
        v.c
    }
}
```

- *influ1*:

```
influ1_ function(model) {
    G_ model$Ghe
    sE _ G * (model$Y - model$X %*% as.matrix(model$mean.theta))
    svX_ svd(model$X * G)
#H_ svX$v %*% (t(svX$v)*(1/svX$d^2))
    V_ svX$u %*% t(svX$u)
    if ( (sum(diag(V)) <= 0) > 0)
```

```
        stop("\nnot all diagonals of V are greater than 0")
    spC _ - ( svX$v * sqrt(c(1/model$var.theta)) ) %*%
            ( t(svX$u * (as.vector(sE)/(1-diag(V))))*(1/svX$d))
    maxabs_ max(abs(spC))
    rr_ dim(spC)[1];  nn_ dim(spC)[2]
    idx_ (1:(rr*nn))*(abs(spC)==maxabs)
    idx1_ idx2_ idx_ idx[idx>0]
    max.spC_ rep(NA,length(idx))
    for (i in 1:length(idx)) {
        idx1[i]_ idx[i] %% rr
        idx2[i]_ 1 + as.integer(idx[i]/rr)
        max.spC[i]_ round(spC[idx1[i],idx2[i]],3)
    }
    ifu_ list(spC=t(spC),entries=cbind(rows=idx2,cols=idx1),
                    max.spC = max.spC)
    ifu
}
```

- *resd*:

```
resd_ function(model,plotit=T,
               labs=rep("d",model$rows[1])    )
{
    X_ model$X; G_ model$Ghe
    rX1_ model$rows[1]
    yhat_ as.vector(X %*% model$mean.theta)
    rX _ dim(X)[1]
    E_ model$Y - yhat
    svX_ svd(X * G)
    V  _ svX$u %*% t(svX$u)
    sE _ G * E
    stud.res_ sE/sqrt(diag(diag(rX)-V))
    stud.res1_ stud.res[1:rX1];  stud.res2_ stud.res[(rX1+1):rX]
    if (plotit) {
        par(mfrow=c(1,2),pty="s")
        qqnorm(stud.res[1:rX1], main="the data cases")
        abline(1,lty=3)
        qqnorm(stud.res[(rX1+1):sum(model$ptab.len)],
                            main="the constraint cases")
        abline(1,lty=3)
        par(mfrow=c(1,1),pty="m",ask=T)
        plot(yhat[1:rX1],stud.res1, xlab="fitted value",
        ylab="scaled data-case residual",type="n")
        text(yhat,stud.res, labs, cex = 1)
        fit.sry_ lsfit(yhat,stud.res)
        abline(fit.sry)
    }
    z_ list(yhat=as.vector(yhat),stud.res=stud.res,
```

```
            slope=1-fit.sry$coef[2],
            std.err=ls.diag(fit.sry)$std.err[2])
    z
}
```

- *transf*:

```
transf_ function(model, C=0, plotit=F,
                labs=c(rep("d",model$rows[1]),
                       rep("c",sum(model$ptab.len) -
                                   model$rows[1]),
                       rep("p",dim(model$X)[1] -
                                   sum(model$ptab.len)))
                )
{
    X_ model$X
    rX1_ model$rows[1]
    rX1.1_ sum(model$ptab.len)
    rX_  dim(X)[1]
    e_ as.matrix(model$mean.theta)
    yhat_ X %*% e + C
    E_ model$Y - yhat - C
    sE_ as.vector(E * model$Ghe)
    svX_ svd(model$X * model$Ghe)
    V_ svX$u %*% t(svX$u)
    yhat_ yhat[1:rX1]
    if (any(yhat <= 0)) stop("\nnegative fitted values, choose C>0?")
    G_ as.matrix(c(yhat*log(yhat) - yhat + 1,
                           rep(0, rX - rX1))) * model$Ghe
    avtx_ as.vector((diag(rX) - V) %*% G)
    fit.avtx_ lsfit(avtx,sE)
    if(plotit) {
        plot(avtx, sE,
            xlab = "Andrews added variable for transformation",
            ylab = "Scaled residuals", type="n")
        text(avtx,sE,labs, cex = 1)
        abline(fit.avtx)
    }
    z_ list(avtx = avtx, scal.res = sE, C=C,
                lambda = 1 - fit.avtx$coef[2],
                std.err = ls.diag(fit.avtx)$std.err[2])
    z
}
```

7 Appendix B: Data Sets

7.1 Rats Data

The data set below is from 16 rats in the two treatment groups. The first column ("Epi") is the count of epibatidine receptors (per unit neural tissue); the second column ("Cyt") is the count of cytisine receptors; the third column ("Diff") is the difference of epibatidine and cytisine. This should be positive, but it is not surprising that a few are negative, because the epibatidine and cytisine receptors were assayed on different pieces of tissue (and the assay has non-trivial error). The fourth column ("Tx") is the treatment group—C for control and N for nicotine. The fifth column ("Tiss") is the tissue type: CORtical, ADRenal, THAlamus, and HIPpocampus. The last column ("Rat") gives the number of the rat from which the tissue was drawn.

Epi	Cyt	Diff	Tx	Tiss	Rat
3.951	2.82	1.131	C	COR	1
3.585	2.43	1.155	C	COR	2
4.77	3.96	0.81	C	COR	3
4.491	3.8	0.691	C	COR	4
4.783	4.35	0.433	C	COR	5
4.455	4.02	0.435	C	COR	6
4.245	3.95	0.295	C	COR	7
4.284	3.91	0.374	C	COR	8
4.779	2.88	1.899	N	COR	1
4.993	4.4	0.593	N	COR	2
5.905	5.37	0.535	N	COR	3
5.562	5.81	-0.248	N	COR	4
4.689	5.07	-0.381	N	COR	5
5.583	5.43	0.153	N	COR	6
5.285	4.47	0.815	N	COR	7
4.642	4.85	-0.208	N	COR	8

Epi	Cyt	Diff	Tx	Tiss	Rat
15.69	0	15.69	C	ADR	1
15.09	0	15.09	C	ADR	2
9.15	0	9.15	C	ADR	3
14.23	0	14.23	C	ADR	4
8.75	0	8.75	C	ADR	5
12.04	0	12.04	C	ADR	6
7.84	0	7.84	C	ADR	7
8.24	0	8.24	C	ADR	8
11.21	0	11.21	N	ADR	1
17.18	0	17.18	N	ADR	2
14.8	0	14.8	N	ADR	3
12.9	0	12.9	N	ADR	4
8.59	0	8.59	N	ADR	5
14.78	0	14.78	N	ADR	6
7.98	0	7.98	N	ADR	7
12.52	0	12.52	N	ADR	8
9.914	6.411	3.503	C	THA	1
9.783	6.503	3.28	C	THA	2
8.971	6.001	2.97	C	THA	3
10.059	6.58	3.479	C	THA	4
10.339	6.774	3.565	N	THA	1
9.929	7.257	2.672	N	THA	2
11.553	7.676	3.877	N	THA	3
10.887	7.851	3.036	N	THA	4
2.024	1.498	0.526	C	HIP	1
2.089	1.731	0.358	C	HIP	2
2.234	1.661	0.573	C	HIP	3
1.823	1.537	0.286	C	HIP	4
2.723	2.083	0.64	N	HIP	1
2.906	2.169	0.737	N	HIP	2
3.231	2.492	0.739	N	HIP	3
3.119	2.298	0.821	N	HIP	4

7.2 Jaw Data

The data set below contains these measurements: the first column is the bend repetition (Rep), taking values 1 through 5; the second column is the displacements; and the third is the measurement of principal strains. The displacements are not all unique: there are 55 measurements but only 52 distinct displacements.

Rep	Disp	Princ
1	-123.60	0.40
1	-120.50	8.50
1	-86.30	26.70
1	-73.80	52.20
1	-45.80	83.50
1	3.90	114.20
1	75.50	135.90
1	94.20	152.50
1	122.20	163.10
1	143.90	172.10
2	-412.90	428.10
2	-409.80	394.20
2	-360.00	329.50
2	-325.80	248.50
2	-238.70	153.80
2	-129.80	50.50
2	-86.30	29.30
2	-24.10	55.90
2	13.30	81.50
2	44.40	105.10
2	78.60	122.50
2	106.60	127.90
2	112.80	126.50
3	-425.40	435.90
3	-375.60	322.70

Rep	Disp	Princ
3	-269.80	189.40
3	-151.60	89.50
3	-80.10	36.50
3	-52.10	46.40
3	-2.30	78.60
3	72.40	110.20
3	112.80	127.60
3	125.30	137.70
3	137.70	140.70
4	-465.80	474.40
4	-394.20	333.40
4	-244.90	184.40
4	-148.50	71.70
4	-67.60	37.50
4	-14.70	66.60
4	41.30	102.50
4	91.00	130.20
4	140.80	149.30
4	156.40	159.20
4	171.90	167.60
5	-528.00	687.20
5	-512.50	599.10
5	-456.50	447.30
5	-335.10	260.60
5	-195.20	90.70
5	-70.70	48.10
5	-5.40	94.50
5	81.70	126.70
5	122.20	143.80
5	159.50	153.60

References

[**A**] Atkinson, A.C. (1985). Plots, Transformations, and Regression. Oxford: Clarendon Press.

[**BR**] Bryk, A.S. and Raudenbush, S.W. (1992). *Hierarchical Linear Models: Applications and Data Analysis Methods.* Newbury Park, California: SAGE Publications

[**CG**] Casella, G. and George, E.I. (1992). Explaining the Gibbs Sampler. *American Statistician*, 46:167-174

[**C**] Cook, R.D. and Weisberg S. (1982). *Residuals and Influence in Regression.* New York: Chapman and Hall

[**H**] Hodges, J.S. (1997). Some Algebra and Geometry for Hierarchical Models, Applied to Diagnostics. *J. Royal Statistical Society, Series B*, to appear

[**W**] Weisberg S. (1985). *Applied Linear Regression,* 2nd edition. New York: Wiley