Limburgs Universitair Centrum
Diepenbeek

Nonlinear Modeling Applications in Experimental Pharmacology

January 1997

Project presented to obtain a
Master Degree in Biostatistics by
Luc Wouters

Janssen Research Foundation, Turnhoutseweg 30, B2340 Beerse

Acknowledgements

Completion of a project necessarily involves the direct and indirect contribution of many people.

First of all, I would like to thank Prof. Dr. Marcel Borgers for his personal encouragement and support through the duration of the studies and the completion of this project.

Many thanks to Ludo Gypen for the numerous statistical discussions, the exchange of theoretical knowledge and practical experience and above all, for reviewing part of the text.

Special thanks is also due to Jeff Lubin for his linguistic assistance.

I would also like to acknowledge the teaching staff of the postgraduate programme for their motivating courses.

I am also indebted to Dr. Marcel Janssen and Dr. Staf Van Reet, for giving their support and for recognizing the important role of continuous education in a research organisation.

During this project it was essential to collect and analyze actual research data. Therefore, my special thanks to the colleagues that gave permission to use their data. Interesting data sets were brought forward by many people, especially Walter De Ridder and Karel Van Ammel were most helpful.

In these days of world-wide electronic communication, help can come from many places. Newsgroups are becoming as essential as a library. Special thanks to those persons that almost instantly replied to my cries for help, in particular: Douglas Bates, José Pinheiro, Brian Ripley, William Venables, and Frank Vonesh.

This may be the proper moment to express my gratitude to two very special persons: Dr. Paul Janssen, founder of the Janssen Research Foundation, who has always been a source of inspiration for those people who were fortunate to work with him; and Dr. Paul Lewi who, many years ago, introduced me to the fascinating world of multivariate statistics.

Contents

1.
Introduction

1.1 Nonlinear models in experimental pharmacology

In the search for new and better drugs, pharmaceutical research centers employ large-scale screening programs in which, on a yearly basis, thousands of new molecules are tested for a variety of biological activities. Once a promising new compound has been selected, it is necessary to further typify and quantify its action. Therefore, additional studies are conducted and parameters characterizing the activity of the drug are estimated from the experimental data.

In an initial stage, the effect of the compound is studied in simplified biological systems (*in vitro* systems such as cell cultures, isolated tissues, etc.), where a continuous response of the system is measured against increasing concentrations of the drug. In more advanced stages, experiments that involve testing the drug in living animals are carried out. In both cases, it is sometimes possible to explicitly model the response as a function of the concentration or dose. Such quantitative experiments, whose interest lies in estimating and comparing *potencies* of drugs have historically been termed *biological assays* or *bioassays* (Finney, 1964).

Another approach to the drug-discovery process focuses on the magnitude of the effect of a single dose of the drug. However, in such studies, it is not always the measured response that is of direct interest, since this may evolve over time, such as the growth of an organism. It is then necessary to model the response as a function of time and estimate the effect of the drug on the parameters of the model.

In both paradigms, the mathematical expressions that relate the response to the regressor variable are usually nonlinear in the parameters. Nonlinear models can be based on theoretical considerations or be used empirically, to build some known nonlinear behavior in a model. Theory-derived or *mechanistic* models have the advantage of not just yielding a description of the data, but the estimated parameters also have a physical interpretation.

1.2 A theory-based model: the dose-response relation

Biological assays, as receptor-occupancy studies, provide a good example for the derivation of a theory-based nonlinear model. In these experiments[1], rats are treated by subcutaneous injection of a drug at different dosages. The animals are sacrificed, the brains removed and the percentage receptor-occupancy is measured by autoradiography. The simplest model to describe the binding of a drug to a receptor is based on the law of mass action (Tallarida and Jacob, 1979). If A denotes the drug and R its receptor, then the combination of p molecules of the drug with one molecule of the receptor is expressed by the reversible chemical reaction:

$$pA + R \underset{k_2}{\overset{k_1}{\rightleftarrows}} A_p R$$

where A_pR is the drug-receptor complex, and k_1 and k_2 are the rate constants of association and dissociation.

According to the law of mass action, at equilibrium the product of the masses on the right side of the equation, divided by that on the left side is equal to some constant, the association constant of the drug-receptor complex:

$$K = \frac{k_1}{k_2} = \frac{[A_p R]}{[A]^p [R]} \tag{1.1}$$

Let x denote the concentration of the drug and y the fraction of receptors bound as A_pR. The remaining fraction of receptors left unbound is $1 - y$ and equation (1.1) can be reformulated as:

$$K = \frac{y}{x^p (1-y)} \tag{1.2}$$

Solving (1.2) for y and assuming an additive stochastic disturbance, yields the following relationship for the receptor-occupancy y_i of the i^{th} animal treated with dose x_i:

$$y_i = \frac{K x_i^p}{1 + K x_i^p} + \varepsilon_i \tag{1.3}$$

Usually the concentrations or dosages that are studied are administered as a geometric progression. Applying (1.3) would then give excessive weight to the results obtained at the higher concentrations. For geometric series of dosages, a more convenient solution to (1.2) is obtained by taking logarithms[2]:

$$\log \frac{y}{1-y} = \log K + p \log x \tag{1.4}$$

[1] A more complete description of these experiments is given in Schotte, A. (1993).

[2] The notation log refers to the natural logarithm base e.

Solving (1.4) for y, yields as an alternative to (1.3), the logistic model:

$$y_i = \frac{1}{1+\exp(-\log K - p\log x_i)} + \varepsilon_i \qquad (1.5)$$

The parameters *K* and *p* of models (1.3) and (1.5) have a physical meaning in the sense that *K* is the dissociation constant and *p* is the number of drug molecules that bind to a single receptor molecule.

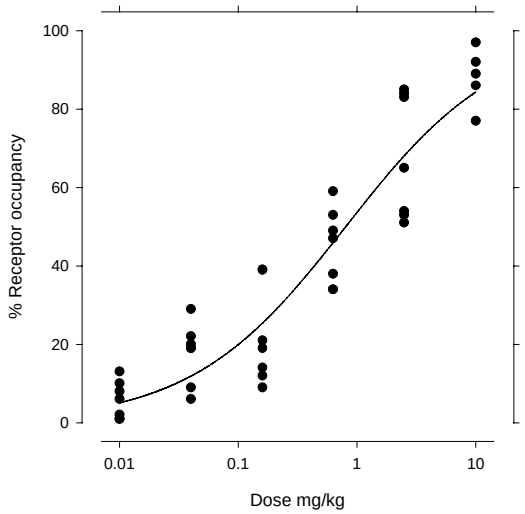

Figure 1.1 Data from a receptor-occupancy experiment and fitted logistic model (1.6)

Note that for a receptor-occupancy of 50 % (*y = 0.5)*, (1.4) equals zero and the logarithm of the corresponding concentration x_{50} is equal to –(log *K*)/*p*. Subtracting log x_{50} from log x_i in (1.5), yields another parametrization of the model:

$$y_i = \frac{1}{1+\exp(p(\log x_{50} - \log x_i))} + \varepsilon_i \qquad (1.6)$$

This expression describes a sigmoidally shaped curve for response versus the logarithm of the concentration. The parameter x_{50} is the inflection point of the curve and is used in pharmacology as a measure for the potency of a drug, often referred to as EC_{50}, ED_{50}, or IC_{50}-value (effective concentration or dose, or inhibitory concentration for 50 % effect or inhibition) of a compound. The application of the logistic model (1.6) to bioassays and dose-response studies in general dates back to Emmens (1940). Equations (1.3), (1.5), and (1.6) constitute models that are nonlinear in the parameters and cannot be transformed to linear models without modifying the error structure.

Figure 1.1 shows data from an actual ex vivo receptor-occupancy experiment (cf. Appendix A), together with the least squares fit of model (1.6). Although there is a lot of variability among the individual responses, the data are well described by the theoretical model.

1.3 Repeated measurements and longitudinal observations

In the receptor-occupancy study, only one observation was made for each animal. However, in many pharmacological experiments, repeated observations on the same individual are possible. Growth studies and gastric emptying experiments necessarily involve the collection of longitudinal data on the same subject. Repeated measures also arise in many *in vitro* experiments where observations are made in the same experimental unit (Petri-dish, batch, run) and consequently correlated data are obtained.

The combination of an underlying nonlinear model with repeated measures requires fairly new statistical modeling methods. Nonlinear mixed effects models, also called hierarchical nonlinear models, were developed for dealing with this type of analysis. However, their usage has mostly been restricted to the area of pharmacokinetic analysis and only very recently have textbooks describing their theory been published (Davidian and Giltinan, 1995) or are in press (Vonesh and Chinchilli, 1996).

1.4 Objectives

The objective of this project was to investigate the applicability of nonlinear models and, in particular, nonlinear mixed effects models in different areas of pharmacological research at the Janssen Research Foundation, with special emphasis on the dose response relationship. This included a study of the underlying theory of nonlinear models and nonlinear mixed effects models, an evaluation and comparison of the available software and the application of the methods to different data sets.

1.5 Structure of the report

The remaining part of this report consists of separate chapters on nonlinear regression, and the handling of heteroscedasticity and repeated measurements in the context of nonlinear modeling. Each chapter starts with a short review of the underlying theory, followed by a section on the implementation in the statistical packages SAS 6.11 and S-Plus 3.3. At the end of the chapter actual research data are analyzed. Figures and tables supporting major ideas or findings are kept in the main part of the report, while intermediate results and computer outputs are contained in separate appendices.

2.
Nonlinear models

In contrast to linear models, for which there is a vast number of textbooks available, there is a paucity of books on nonlinear regression. Bates and Watts (1988) and Seber and Wild (1989) present both the theoretical and practical aspects of nonlinear estimation. Ratkowsky (1990) gives practical advice on how to best parametrize some commonly used nonlinear models.

2.1 Nonlinear least squares fitting

The general form of a univariate nonlinear regression model with additive error can be written as:

$$y_i = f(\mathbf{x}_i, \beta) + \varepsilon_i \tag{2.1}$$

where y_i represents the continuous response for the i^{th} case of a total of n observations, f some function (expectation function) that is nonlinear in the parameters, $\mathbf{x}_i$ a vector of regressor variables, β the set of p parameters and ε_i the error term. The error terms are assumed to be independent, and normally distributed with zero mean and constant variance σ^2.

Model (2.1) is called *nonlinear* if at least one of the derivatives of the expectation function f with respect to the parameters depends on at least one of the parameters. In some cases it is possible to transform the nonlinear model into a linear model. However, the effect of such transformations on the error term has also to be taken into account. Furthermore, the parameters of the linearized model are often not as interesting, or as important, as the original parameters.

The least squares estimate $\hat{\beta}$ of β minimizes the error sums of squares:

$$S(\beta) = \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i, \beta) \right)^2 \tag{2.2}$$

Since for model (2.1) at least one of the normal equations involves nonlinear functions of the β, closed-form solutions to (2.2) are not always possible. Hence, iterative procedures are necessary to find an approximate solution. Seber and Wild (1989) list a large variety of iterative strategies that have been developed for finding nonlinear least squares estimates. Five classical techniques available in statistical software (e.g. SAS) are: the (modified) Gauss-Newton method, the method of steepest-descent (also called *gradient* method), a compromise between these two methods known as the Levenberg-Marquardt algorithm, the Newton-Raphson method which uses the second derivatives as well, and the derivative-free method DUD ("Doesn't Use Derivatives"). More technical details on these algorithms are given in

the books by Draper and Smith (1981), Bates and Watts (1988), Seber and Wild (1989), and Glantz and Slinker (1990). Since the Gauss-Newton method has the most widespread use and is related to asymptotic inference, the derivation of this technique will be given in full detail.

The idea behind all iterative procedures is to improve an initial guess β^0 for β and to keep improving the estimates until there is no change. The Gauss-Newton method is based on a linearization of the expectation function $f(\mathbf{x}_i, \beta)$ using a first order Taylor series about β^0:

$$f(\mathbf{x}_i, \beta) \approx f(\mathbf{x}_i, \beta^0) + \sum_{k=1}^{p} \left. \frac{\partial f(\mathbf{x}_i, \beta_k)}{\partial \beta_k} \right|_{\beta_k = \beta_k^0} (\beta_k - \beta_k^0) \tag{2.3}$$

Let $f_i(\beta) = f(\mathbf{x}_i, \beta)$, $\mathbf{f}(\beta) = \left(f_1(\beta), f_2(\beta), \ldots, f_n(\beta) \right)^{\mathrm{T}}$, $\mathbf{r}(\beta) = \mathbf{y} - \mathbf{f}(\beta)$ the residual vector, and let $\mathbf{V}(\beta^0)$ denote the matrix of partial derivatives with respect to β and evaluated at β^0:

$$\mathbf{V}(\beta^0) = \left. \frac{\partial f_i(\beta)}{\partial \beta_k} \right|_{\beta_k = \beta_k^0}$$

Then (2.3) can be written as $\mathbf{f}(\beta) \approx \mathbf{f}(\beta^0) + \mathbf{V}(\beta^0)(\beta - \beta^0)$ (2.4)

Substituting (2.4) into the definition of the residual vector yields:

$$\begin{aligned} \mathbf{r}(\beta) &\approx \mathbf{y} - \mathbf{f}(\beta^0) - \mathbf{V}(\beta^0)(\beta - \beta^0) \\ &\approx \mathbf{r}(\beta^0) - \mathbf{V}(\beta^0)(\beta - \beta^0) \end{aligned} \tag{2.5}$$

Rewriting the sums of squares as $S(\beta) = \mathbf{r}^{\mathrm{T}}(\beta)\, \mathbf{r}(\beta)$, which using (2.5) is approximated by:

$$S(\beta) \approx \left(\mathbf{r}(\beta^0) - \mathbf{V}(\beta^0)(\beta - \beta^0) \right)^{\mathrm{T}} \left(\mathbf{r}(\beta^0) - \mathbf{V}(\beta^0)(\beta - \beta^0) \right) \tag{2.6}$$

$$\approx \mathbf{r}^{\mathrm{T}}(\beta^0)\mathbf{r}(\beta^0) - 2\mathbf{r}^{\mathrm{T}}(\beta^0)\mathbf{V}(\beta^0)(\beta - \beta^0) + (\beta - \beta^0)^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}(\beta^0)\mathbf{V}(\beta^0)(\beta - \beta^0) \tag{2.7}$$

Vector differentiation shows that the right-hand side of (2.7) is minimized with respect to β when $(\beta - \beta^0) = \left(\mathbf{V}^{\mathrm{T}}(\beta^0)\mathbf{V}(\beta^0) \right)^{-1} \mathbf{V}^{\mathrm{T}}(\beta^0)\mathbf{r}(\beta^0) = \delta^0$

This suggests that a better approximation to β is obtained by the new approximation:

$$\beta^1 = \beta^0 + \delta^0 \tag{2.8}$$

Full iteration until convergence will yield the least squares estimate $\hat{\beta}$. Note that (2.6) is in fact a linear regression problem with $\mathbf{V}(\beta^0)$ as the matrix of regressors and $\delta = \beta^1 - \beta^0$ the parameters to be estimated.

The Gauss-Newton method will always converge, provided the sample size *n* is large enough and the initial values are close enough to the true values. When the sample size is small, or the initial guess is poor, the algorithm may converge very slowly, or even diverge. The *modified* Gauss-Newton method[1] protects against divergence by halving the increments in

[1] This modification is due to Hartley (1961).

the parameter estimates if the residual sums of squares increases in any step. Hence, a step factor λ is introduced and (2.8) becomes $\beta^1 = \beta^0 + \lambda\delta^0$.

Which of the five available algorithms performs best depends on the form of the estimation function, the design of the experiment, and the sample size. In general, the steepest descent method is not recommended, since it may converge very slowly. It can be of use when only a poor guess of the starting values is possible. The Levenberg-Marquardt method is recommended when the parameter estimates are highly correlated or the objective function S(β) is not well approximated by a quadratic. The derivative-free DUD method is of interest when it is difficult to obtain the partial derivatives analytically.

Cucchiara, *et al*. (1989) used Monte-Carlo simulations to investigate the performance of the five techniques for estimating the parameters of a bi-exponential model. They found that the method of steepest descent failed to converge regularly. The Levenberg-Marquardt procedure produced estimates that agreed best with the true parameters. Although of the remaining four methods, this method failed to converge the most often, the other methods had a tendency to claim convergence at the expense of poor estimation.

All procedures require a good first guess of the parameters to find the global minimum in the residual sum of squares. Starting values may be obtained from prior knowledge of the situation, inspection of the data, grid search, or trial and error. An overview of techniques for obtaining starting values is given by Bates and Watts (1988).

For small to moderate sample sizes, problems of non-convergence or convergence to a local-minimum may still be encountered. Other problem areas include identifiability of the parameters and ill-conditioning due to correlation among the estimates. In all of these cases, a reparametrization of the model, centering and scaling of the data, or use of a grid of starting values must be considered.

2.2 Inference in nonlinear regression

Inference in nonlinear regression is based on the same linearization as was used in the Gauss-Newton method. In the neighborhood of β*, the true value of β, applying (2.6) with β^0 replaced by β* reduces the nonlinear problem to a linear model. Then, under certain regularity conditions, $\hat{\beta}$ and $s^2 = S(\hat{\beta}) / (n - p)$ are consistent estimates of β* and σ^2 respectively. With further regularity conditions, the $\hat{\beta}$ are also asymptotically normally distributed with mean **β*** and variance-covariance matrix $\sigma^2 (\mathbf{V}^{\mathrm{T}}\mathbf{V})^{-1}$, which is estimated by:

$$\mathbf{s}^2(\hat{\beta}) = s^2 (\hat{\mathbf{V}}^{\mathrm{T}}\hat{\mathbf{V}})^{-1} \tag{2.9}$$

where $\hat{\mathbf{V}}$ is the matrix of partial derivatives of $f(\mathbf{x}_i, \beta)$ with respect to β, evaluated at $\hat{\beta}$.

Note that in nonlinear regression $\hat{\mathbf{V}}$ plays the same role as the **X**-matrix in linear regression.

When the ε_i are normally distributed, then it is straight forward to show that $\hat{\beta}$ is also the maximum likelihood estimator.

Given the distributional assumptions about the error terms of model (2.1), then in analogy to linear regression, an approximate 100(1 – α) % joint (Wald-type) confidence region for β is given by the ellipsoid:

$$(\hat{\beta}-\beta)^{T}(\hat{\mathbf{V}}^{T}\hat{\mathbf{V}})^{-1}(\hat{\beta}-\beta) \leq ps^{2}F(p,n-p,1-\alpha) \tag{2.10}$$

while an approximate 100(1 – α) % confidence interval for an individual parameter β_k is obtained by:

$$\hat{\beta}_k \pm t(1-\alpha/2,n-p)s(\hat{\beta}_k) \tag{2.11}$$

where $s^2(\hat{\beta}_k)$ is the kth diagonal element of the covariance matrix defined in (2.9).

The above Wald-inference depends on the validity of the linear approximation to the model about $\hat{\beta}$. Generally, the smaller the sample size, the greater the extent of nonlinearity. Moreover, when sample size is moderate or small, the least squares estimators are not necessarily unbiased, normally distributed, minimum variance estimators. As a consequence, the linear approximation is often inadequate and the Wald-type intervals can be very inaccurate and even misleading.

An alternative to the Wald-type confidence regions is provided by likelihood-based confidence regions. These regions are called *exact*, as they are not based on a linear approximation as is the case for (2.10) and (2.11). However, the confidence levels or coverage probabilities of such regions are generally unknown, though approximate levels can be obtained from asymptotic theory. A likelihood-based 100(1 – α) % *exact* joint confidence region is defined as the set of all values of β, such that:

$$S(\beta) \leq S(\hat{\beta})\left(1+\frac{p}{n-p}F(p,n-p,1-\alpha)\right) \tag{2.12}$$

For individual parameters, profile likelihood intervals can be constructed, which are related to the graphical assessment of curvature.

Graphical procedures and a numerical measure for assessing the degree of nonlinearity were developed by Bates and Watts (1988). One graphical method consists of plotting for each parameter θ_k the profile *t* function:

$$\tau(\beta_k) = sign(\beta_k-\hat{\beta}_k)\sqrt{\tilde{S}(\beta_k)-S(\hat{\beta})}\,/\,s \tag{2.13}$$

where $\tilde{S}(\beta_k)$ is the profile sums of squares function, i.e. the residual sums of squares minimized with respect to all other *p – 1* parameters. Plots of $\tau(\beta_k)$ versus β_k reveal how nonlinear the estimation situation is. For a linear model this plot is a straight line, for a nonlinear

model it is curved and the amount of curvature yields information about the nonlinearity of the model. In addition to profile *t* plots, plotting for each parameter pair the 2-dimensional projection of the joint confidence regions provided by both (2.10) and (2.12) can also be informative. The profile *t* function (2.13) also allows $100(1 - \alpha)$ % profile-likelihood intervals to be set up for individual parameters, as the set of all β_k for which:

$$t(n-p,\alpha/2) \leq \tau(\beta_k) \leq t(n-p,1-\alpha/2) \tag{2.14}$$

Bates and Watts (1988) also measure the degree of curvature by two numerical summary statistics that are based on the second derivatives of the expectation function. They define intrinsic curvature, i.e. curvature of the solution locus itself, and parameter-effects curvature. Intrinsic curvature measures the amount of curvature of the solution locus in sample space, where the locus represents all possible solutions to the estimation problem. For most models intrinsic curvature is small. Parameter effects curvature is a measure of the lack of parallelism and the inequality in the spacing of parameter lines on the solution locus at the least squares solution. As a rule of thumb, curvature values multiplied by the square root of *F(p, n-p, 0.95)* should not exceed 0.3. Seber and Wild (1989) mention cases where it has been demonstrated that these measures are not entirely reliable.

When there are indications of parameter effects curvature, a reparametrization of the expectation function must be considered. Ratkowsky (1990) has termed the phrase *close-to-linear* for certain nonlinear model-data combinations where the likelihood-based regions and the Wald-type regions coincide, even when the sample size is small. He also provides reparametrizations for several commonly used nonlinear models.

2.3 Diagnostics

In addition to the measures of curvature discussed earlier, nonlinear modeling also requires assessing the fit of the model and the appropriateness of the assumptions. In nonlinear regression, it is sometimes possible to converge to parameter estimates that are obviously wrong. Therefore, a first diagnostic is to check whether the parameter estimates make sense and convergence occurred smoothly. If parameter estimates do not make sense, or convergence did not occur smoothly, a different set of starting values should be considered. A second check consists of the correlation matrix of parameter estimates. Correlations of 0.99 or higher in absolute value are considered to indicate serious collinearity among the parameter estimates. Simplifying the model or transforming variables and parameters can be of use to overcome the collinearity problem.

Graphical tools include a plot of the observed responses versus the predicted values and, as in linear regression, various plots of the residuals. However, in nonlinear regression, residuals do not have the nice properties as in linear regression. When there is substantial intrinsic curvature, the residuals will have non-zero means and different variances. A plot of the residuals versus the fitted values will then tend to slope downwards (Seber and Wild, 1989). In

practice, when curvature is not too extreme, most of the diagnostics and residual analyses for linear least squares apply, in particular studentized residuals and Cook's Distance. (Cook and Weissberg, 1982).

Studentized residuals r_i^* are computed from the original residuals and the nonlinear hat matrix, defined as:

$$\mathbf{H} = \hat{\mathbf{V}}(\hat{\mathbf{V}}^{\mathrm{T}}\hat{\mathbf{V}})^{-1}\hat{\mathbf{V}}^{\mathrm{T}}$$

$$r_i^* = r_i / s\sqrt{1-h_{ii}}$$

with h_{ii} being the diagonal of **H**. Cook's distance measure D_i is computed analogously, using the expression:

$$D_i = \frac{r_i^2}{ps^2}\frac{h_{ii}}{(1-h_{ii})^2}$$

When there are repeated observations at the same level of the regressor, it is also possible to carry out a formal test for lack of fit, in the same manner as for linear regression (Draper and Smith, 1981; Neter, Wasserman, and Kutner, 1990).

2.4 Software implementation

In S-Plus, nonlinear regression problems can be fitted with the function *nls*. This function contains an implementation of the modified Gauss-Newton algorithm and can be used with or without explicitly specifying the partial derivatives. In the latter case, the derivatives are approximated numerically[1]. The matrix of partial derivatives can be generated with the function *deriv*. Alternatively, second derivatives can also be supplied and are then taken into account. Second derivatives can be generated with the function *deriv3*, which is contained in the *mass*-library[2] of Venables and Ripley (1994).

Chambers and Hastie (1993), and Venables and Ripley (1994) give detailed instructions on how to fit nonlinear functions with *nls* and how to construct profile *t* plots with the *profile* function. Venables and Ripley (1994) also show how to compute the measures for curvature using the functions *deriv3* and *rms.curv* from the *mass*-library. The construction of likelihood intervals requires some programming effort, as does the computation of studentized residuals and Cook's D. For the computation of profile likelihood intervals, Venables (personal communication) has implemented a computer-efficient approximation by spline interpolation on the results of the *profile* function. Confidence regions can be plotted with the function *ellipse* from the *ellipse*-library.

---

[1] Note that using numerical derivatives in the Gauss-Newton procedure is not the same as the DUD-algorithm that is implemented in SAS.

[2] The *mass* and other S-Plus libraries mentioned (*ellipse*, *nlme*), are available from *StatLib*, located at Carnegie-Mellon University. *StatLib* is a system for electronic distribution of statistical software, datasets, etc. and can be accessed by http://lib.stat.cmu.edu/.

In SAS, all five iterative strategies are available in *PROC NLIN* (SAS Institute, 1989). SAS only provides for Wald confidence intervals and no diagnostics for curvature are computed. However, Price and Shafii (1992), and O'Brien and Wang (1996) give details on how to construct profile pair sketches, profile *t* plots and profile likelihood intervals. A major drawback in using SAS for nonlinear regression analysis is that, apart from the DUD method, the user has to supply the partial derivatives with respect to the parameters, which can be a tedious undertaking for complex models.

Sections A.3 and A.4 of Appendix A contain the analysis of the receptor-occupancy data in S-Plus and SAS. The S-Plus analysis was performed with the numerical and analytical derivatives. In SAS, all five available iteration strategies were used. A comparison of the different packages and the different algorithms shows that, as long as starting values are reasonable and the problem is well-defined, the results are comparable. A more detailed discussion of the analysis is given in the next section. The remaining model assessment and analyses were carried out in S-Plus, as it was less tedious than SAS for obtaining additional results.

2.5 Applications

2.5.1 Receptor-occupancy study

Since the investigator is interested in an estimate of the potency of the drug, model (1.6) is used. Unlike linear models, it is not necessary to perform the logarithmic transformation of the dosage outside the model, since nonlinear models can incorporate any mathematical function. It is also tempting to estimate the value of x_{50} (ED_{50}) directly. However, this would allow the iterative procedure to try fitting negative values for the parameter x_{50}, which would cause the algorithm to run into difficulties. Furthermore, it is a well known fact that values for equi-effective doses such as ED_{50} are log-normally distributed (Fleming, *et al.*, 1972). Therefore, the parameter $\xi = \log x_{50}$ is estimated instead. This will also constrain x_{50} to positive values greater than zero.

Starting values for the parameter estimates are determined using the linearizing transformation of (1.4). Figure A.1 in Appendix A, shows a plot of ($\log y/(100 - y)$) versus log dose. Notice that the relation is almost linear but the logit-transformation has seriously disturbed the error structure. While for the original data (Fig. 1.1), the error was independent of the response, now there is more spread at the low and high doses. Linear regression (Section A.2) of $\log y/(100 - y)$ on log dose yields as estimates for slope and intercept -0.26 and 0.73 respectively, from which -0.26/0.73 = -0.36 is obtained as starting value for ξ.

The analysis in S-Plus (Section A.3) starts with a nonlinear regression in which analytical derivatives are automatically generated using the *deriv* function. This produces a new S-function (*der*), which subsequently is used as input to the nonlinear fitting function *nls*. For

the starting values, the model yields an error sums of squares of 4308. The first iteration is already a marked improvement, since the error sums of squares drops to 4119. The procedure takes two more steps to converge, yielding final estimates for the parameters ξ and p of -0.221 and 0.668 respectively. The correlation coefficient between the two parameter estimates, of -0.121, indicates that collinearity can be neglected.

Using numerical derivatives in the present application requires the same number of iterations as above. However, this is not always the case. Usually, omitting the gradient will require more iterations, since the algorithm is forced to use a less precise numerical approximation. In the present application, the model is close-to-linear and the initial values of the parameters are quite near the final estimates. The resulting parameter estimates and standard errors are exactly the same as when supplying analytical derivatives. The function *deriv3* from the *mass*-library also allows the generation of second derivatives. Supplying these to the *nls* function yields the same results as before.

All five iterative strategies that are available in SAS were carried out on the data (Section A.4). For the gradient-based methods, SAS requires derivatives to be generated by the user and supplied to the *PROC NLIN* programming step as "*der.xxx"* expressions. The Gauss-Newton procedure takes four iterations to converge. The final estimates are almost equal (up to 4 decimal places) to those obtained in S-Plus. The steepest-descent or gradient method converges more slowly than the other methods. This is to be expected, since it is known that the method of steepest descent has difficulty converging once it gets close to the final values. The results obtained for the five iterative strategies are almost equal to one another.

An assessment of the fit of the model is carried out only in S-Plus (Section A.5). Plotting the observed versus the predicted values (Fig. A.2), shows that there is no evidence for lack of fit or unequal error variances. As mentioned earlier, the residuals do not necessarily sum to zero and have 0.77 as mean value. A function for computation of studentized residuals is also given in A.5. Plots of the studentized residuals show that no outlying observations are present (outliers defined as absolute value of studentized residual ≥ 4). Considering tail areas of 0.05 on each side of the *t* distribution to be extreme (Neter, Wasserman and Kutner, 1990), results in one extreme value having a response of 51 % at a dose of 2.5 mg/kg. Cook's distance measure for this observation is 0.103, which corresponds to the 11^{th} percentile of the F distribution with 2 and 38 degrees of freedom. Using a percentile value of 50 percent as a criterion (Neter, Wasserman, and Kutner 1990), this indicates that this observation has little influence. The normal probability plot of the studentized residuals (Fig. A.3) shows that there are no major deviations from normality. The correlation coefficient between the quantiles and their expected value under normality is well above the critical value (Neter, Wasserman and Kutner, 1990) at $\alpha = 0.1$. A plot of the studentized residuals versus the predicted values (Fig. A.4) confirms the earlier findings of homoscedasticity and absence of lack of fit.

Since there are replications at each dose level, a formal test on lack of fit can also be carried out. The error sums of squares and corresponding degrees of freedom are extracted from the nonlinear fit. Next, an analysis of variance with dose levels as factor is carried out, to obtain the pure error sums of squares. Subtracting the pure error component from the error sums of squares obtained earlier, yields the lack-of-fit sums of squares. The corresponding degrees of freedom are obtained in the same manner. Comparing the lack-of-fit component with the pure error component yields an F-value of 0.947 with a corresponding p-value of 0.449. Hence, the formal test also gives no indication for lack of fit.

Assessment of curvature is summarized in Section A.6. The profile *t* plot (Fig. A.5) shows that for this model-data combination, the linear approximation can safely be used for the parameter $\xi = \log x_{50}$. For the slope *p*, there is a small deviation from the linear approximation. Numerical measures for curvature are computed using the methods supplied by Venables and Ripley (1994). Both measures are below the 0.3 criterion, which confirms Ratkowsky (1990), who states that the logistic equation is a close-to-linear model. The joint 95 % confidence regions for *p* and ξ are plotted with the aid of the *ellipse* function and shown in Figure A.6. The confidence regions are very similar to one another, the major discrepancy between the two approaches being found for the slope *p* when ξ is equal to its least squares estimate. Table 2.1 summarizes both types of confidence intervals for the individual parameters.

Table 2.1 Estimated parameter value, lower and upper 95 % confidence intervals

Parameter	Estimated Value	Wald C.I.	Profile likelihood C.I.
p	0.668	0.538, 0.798	0.548, 0.819
ξ	-0.221	-0.512, 0.070	-0.508, 0.072
$x_{50} = exp(\xi)$	*0.802*	*0.600, 1.073*	*0.602, 1.074*

2.5.2 Inhibition of growth-rate of micro-organisms

To assess the environmental impact of the introduction of certain new pharmaceuticals, their effect on the growth-rate of unicellular green algae is investigated. Appendix B contains the data and analysis of such an experiment in which the growth-rate was determined at different concentrations (in mg/l) of a drug. At each concentration three independent replicates were carried out. The individual data are plotted in Figure 2.1. According to the guidelines (Organisation for Economic Co-operation and Development, 1984), the investigator must report the concentration corresponding to a 50 % reduction in growth rate.

As in the receptor-occupancy study, the concentration-response curve has a sigmoidal shape (Fig. 2.1), but now the data are not bounded by 0 and 100 %. An extension of model (1.6) to allow for finite lower and upper asymptotes is given by the four-parameter logistic model:

$$y_i = y_{\min} + \frac{y_{\max} - y_{\min}}{1 + \exp\big(\beta(\xi - \log x_i)\big)} + \varepsilon_i \qquad (2.15)$$

where y_{min} and y_{max} respectively stand for the lower and upper asymptote of the curve, β is the slope, standardized for data between 0 and 1, and $\xi = \log x_{50}$ is the potency parameter. Model (2.16) is described by Ratkowsky as having good statistical properties and its application to dose-response studies and bioassay has been elaborated by Vølund (1978). The data and their detailed analysis are given in Appendix B.

Starting values for y_{min} and y_{max} are obtained by simply taking the minimum and maximum of the data. Subtracting the minimum from the data and dividing by the range yields a standardized variable in the interval [0, 1]. Figure B.1 shows that, for the central portion of the data, there is a linear relation between the logits of the standardized variable and the logarithm of the concentration. Linear regression on this part of the data yields starting values for ξ and β.

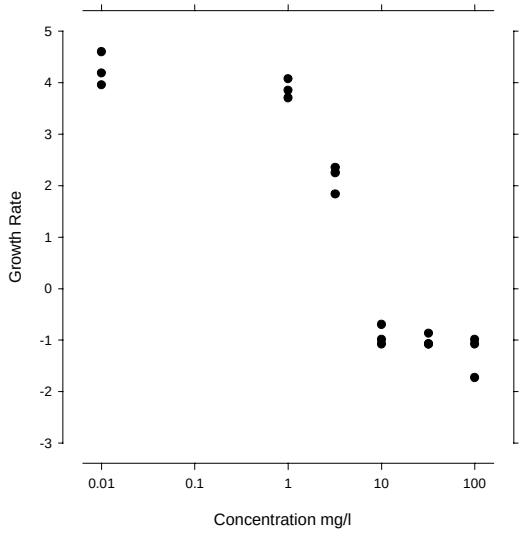

Figure 2.1 Scatterplot of individual growth rates versus concentration

The Gauss-Newton method (Appendix B, Section B.3) converges after six steps, to yield final estimates and asymptotic standard errors of the four parameters. The correlation matrix of the parameter estimates shows no indication for severe multicollinearity.

Assessment of the goodness of fit (Section B.4) shows that model (2.15) adequately describes the data. There are two extreme values, corresponding to the minimum and maximum of the data (observation 2 and 17). While the influence of the maximum can be neglected

completely (Cook's D of 0.232, i.e. 9th percentile of the F-distribution with 4 and 14 degrees of freedom), the extent of the influence of the minimum is substantially larger (Cook's D of 0.770, 44th percentile of the F-distribution), but still not large enough to be of particular concern. A comparison of the fit with and without observation 17 shows that this observation mainly influences the estimates for the lower asymptote (one standard error difference) and the slope, while the estimates for the upper asymptote and potency parameter are hardly affected. A formal test on lack of fit (p = 0.272) confirms that the model adequately describes the data. From the normal probability plot and the correlation coefficient between the studentized residuals and their expected value under normality, it follows that the distribution of the studentized residuals does not deviate significantly from normality.

While the traditional model assessment did not indicate particular problems with the data-model combination, the picture is completely different when one looks at the assessment of curvature (Section B.5). The first indication that there is something wrong is the failure of the *profile* function when the default values are used for setting up the profiling region. Restriction of the region using the *alphamax*= argument does allow the profile *t* plots to be constructed. The graphical assessment (Fig. B.5) shows that the Wald-type confidence intervals (Section B.6) can safely be applied for y_{min} and y_{max}, the parameters describing the asymptotes. The profile *t* plot for ξ shows a strange irregularity, but the Wald confidence interval still provides a reasonable approximation. It is apparent from the profile *t* plot for the slope parameter β that this parameter is poorly determined and that the 95 % likelihood interval will have no lower bound. The numerical measures of curvature yield a value of 0.15 for intrinsic curvature, which is acceptable according to the 0.3 criterion, but the parameter effect curvature yields the immense value of 1.0. The effect of curvature is even more pronounced when one looks at the two-dimensional projections of the 95 % confidence regions (Fig. B.6). The likelihood contours of the plots involving β are not at all elliptical and presumably extend to -∞.

Table 2.2 Estimated values and 95 % confidence intervals for the four parameters of model (2.15)

Parameter	Estimated Value	Wald C.I.	Profile likelihood C.I.
y_{min}	-1.16	-1.42, -0.923	-1.42, -0.923
y_{max}	4.12	3.86, 4.39	3.86, 4.41
β	-2.87	-4.08, -1.66	-∞, -2.01
ξ	1.34	1.21, 1.46	1.20, 1.46
$x_{50} = exp(\xi)$	*3.80*	*3.36, 4.30*	*3.32, 4.33*

Wald-type confidence intervals are computed in Section B.6. For the determination of the profile likelihood confidence intervals (Section B.7), the profiling region of the function *Conf.int* had to be extended (alphamax=0.05/8 instead of 0.05/4). Even then, the function

still failed to produce an upper limit for y_{min}. Further extension of the profiling range by trial and error produced an upper limit for y_{min} of -0.923 (results not shown). The results of the two inference methods are summarized in Table 2.2. Comparison of the profile likelihood intervals with the Wald intervals shows that for y_{max} and $\xi = \log x_{50}$ the results are comparable, as was suggested by the profile t plots. However, for the slope parameter β, the Wald intervals are completely misleading. A final estimate of the potency of the drug can be reported to the investigator as 3.8 with a 95 % confidence interval of 3.3- 4.3.

The reason for the extreme nonlinear behavior for the parameter β of this model, described by Ratkowsky (1990) as having good estimation properties and being close-to-linear, is the extremely steep slope that occurs in the center of the data and that is determined by only three different values for the dose. In fact, the data are behaving more like a step function than like a logistic model. This is made clearer if one considers model (2.15) when the slope approaches -∞. In this case, for concentrations less than x_{50}, (2.15) always yields the upper asymptote, while for concentrations greater than x_{50}, the lower asymptote is obtained. This corresponds rather well with the data for which responses that are markedly different from the asymptotes are obtained only at one concentration. Hence, reparametrizations of the model will not solve the problem that is present in the data. If the investigator wants a more reliable estimate of the slope parameter, he should consider designing a new experiment, with more concentrations located between 1 and 10 mg/l. A remarkable finding is that, although this is an almost pathological data set, the linearization intervals (i.e. Wald intervals) are still applicable, at least approximately, for the parameters related to the asymptotes and the potency.

2.6 Summary

The theory of nonlinear regression is well developed (Seber and Wild, 1990). Commercial statistical packages such as SAS and S-Plus provide a number of methods for fitting nonlinear regression models. For good model-data set combinations, confidence intervals for the parameters constructed using the linear approximation (Wald-type intervals) correspond well with profile likelihood intervals. However, for ill-conditioned model-data set combinations, conventional Wald inference can be seriously misleading for parameters sensitive to curvature. It is essential that the effect of curvature is explored when these intervals are reported. Alternatively, profile likelihood intervals that do not depend upon a linear approximation can be reported. However, the latter can be asymmetric and do not necessarily have the nominal coverage probability. Moreover, cases exist where the profile likelihood intervals are open and it is impossible to report the limits. In these cases, computational difficulties can arise, making it also difficult to compute the profile.

3.
Heterogeneity of variance

In some applications, the assumptions of model (2.1) about equality of variance and independence of residuals are violated. Carroll and Ruppert (1988) describe two strategies for handling these situations. The first method, which can be applied to inequality of variance as well as to correlated residuals, is generalized least squares. A second technique, called *transform both sides*, is useful when there is skewness and nonconstant variance in the data. As the method of generalized least squares is the most versatile, it will be discussed in more detail. The development here is restricted to the problem of heteroscedasticity, since correlated data will commonly be associated with the repeated measures setting of the next chapter.

3.1 Weighted least squares

Consider the case where the responses y_i are averages of w_i independent replicates, with all replicates having common variance σ^2. Then, the y_i have nonconstant variance inversely proportional to the w_i:

$$\mathrm{Var}(y_i) = \sigma^2 / w_i$$

Model (2.1) can now be rewritten as:

$$y_i = f(\mathbf{x}_i, \beta) + \frac{\sigma}{\sqrt{w_i}}\varepsilon_i \tag{3.1}$$

where σ is the scale parameter, and the $\varepsilon_i \sim N(0,1)$.

If we redefine the response as $y_i^* = y_i\sqrt{w_i}$ and the expectation function as:

$f^*(w_i, \mathbf{x}_i, \beta) = f^*(\mathbf{x}_i, \beta)\sqrt{w_i}$, then the redefined response y^* will have constant variance σ^2 with means given by the new expectation function f^*. Fitting the redefined response to the redefined expectation function with the iterative methods discussed earlier, is equivalent to minimizing with respect to θ the expression:

$$S(\theta) = \sum_{i=1}^{n} w_i\left(y_i - f(\mathbf{x}_i, \beta)\right)^2$$

Under the assumptions of this new model, for a large enough sample size, the weighted least squares estimator $\hat{\beta}_{WLS}$ may be shown to be less variable than the ordinary least squares estimator $\hat{\beta}$. An estimator for the variance σ^2 is given by the weighted mean squared error:

$$s^2_{WLS} = \frac{1}{n-p}\sum_{i=1}^{n} w_i\left(y_i - f(\mathbf{x}_i, \hat{\beta}_{WLS})\right)^2 \tag{3.2}$$

For large *n* and with **W** the (diagonal) matrix of weights w_i, the estimator $\hat{\beta}_{WLS}$ is multivariate normally distributed with mean β^*, the true value of β, and approximate variance-covariance matrix:

$$\mathbf{s}^2(\hat{\beta}_{WLS}) = s^2_{WLS}(\hat{\mathbf{V}}^{\mathrm{T}}\mathbf{W}\hat{\mathbf{V}})^{-1} \tag{3.3}$$

Consequently, the inferential procedures that were developed for ordinary least squares also apply to weighted least squares. Standardized residuals are obtained from (3.1) as:

$$r_i^* = \sqrt{w_i}\left(y_i - f(\mathbf{x}_i, \hat{\beta}_{WLS})\right) / \sigma \tag{3.4}$$

3.2 Generalized least squares

In practice, the true weights are rarely known a priori and have to be estimated from the data. Using estimated weights in the above procedure leads to the method of generalized least squares. Estimation of the weights is done by considering the variance of the response to change systematically and smoothly with the level of the predictors or the response[1]. Hence, variance estimation is regarded as a regression problem. The nonlinear regression model of (2.1) is then generalized to a model allowing for heteroscedasticity by specification of a variance function *g*, which can depend on the mean response μ_i, on some constant z_i, or on a parameter vector θ of length *q*. Formally, the model is written as:

$$y_i = f(\mathbf{x}_i, \beta) + \sigma g(\mu_i, z_i, \theta)\varepsilon_i \tag{3.5}$$

where σ is the scale parameter, $g(\mu_i, z_i, \theta)$ is the variance function, and $\varepsilon_i \sim N(0,1)$.

For this model, we have:

$$\mathrm{E}(y_i) = \mu_i = f(\mathbf{x}_i, \beta) \qquad \mathrm{Var}(y_i) = \sigma^2 g^2(\mu_i, z_i, \theta)$$

The specification of the variance function $g(\mu_i, z_i, \theta)$ allows the model to be extremely versatile. For instance, in many applications the variance depends on the mean response as a power of the mean, or $g(\mu_i, z_i, \theta) = \mu_i^{\theta}$ which, for $\theta = 1$ represents a constant coefficient of variation model (variance proportional to the square of the mean).

When the complete functional form of the variance model, as well as the parameter θ**,** is known, an iterative procedure is used to simultaneously estimate the weights and the regression parameter β. The following scheme is applied:

1. Estimate β by a preliminary estimator $\hat{\beta}_*$, usually the ordinary least square estimator.
1. Form the estimated weights: $\hat{w}_i = 1/g^2(\mu_i, z_i, \theta)$
1. Using the weights from step 2, re-estimate β by weighted least squares, yielding $\hat{\beta}_w$
1. Update the preliminary estimator by setting $\hat{\beta}_* = \hat{\beta}_w$ and update the weights in step 2

[1] When genuine repeats are present, one could consider estimating the weights by calculating the variance at each value of the regressor. However, this naïve approach is unreliable if the number of replicates at each value of the regressor is small.

1. Repeat steps 3 and 4 c - 1 more times

The final estimate is denoted by $\hat{\beta}_{GLS}$. Full iteration of the algorithm is a version of iteratively reweighted least squares and is usually performed with alternative computational methods (Mc Cullagh and Nelder, 1989). Carroll and Ruppert (1988) recommend using at least two cycles of generalized least squares, to eliminate the effect of the initial unweighted least squares estimate.

Generalized least squares can be applied without making any distributional assumptions. The estimator $\hat{\beta}_{GLS}$ has the same distributional properties as the weighted least squares estimator $\hat{\beta}_{WLS}$ where the known weights w_i in (3.2) - (3.4) are now replaced with the estimated weights $\hat{w}_i = 1/g^2(\mu_i, z_i, \theta)$. Application of likelihood-based inference proceeds in the same manner as for ordinary least squares with the estimated weights based on the full model.

3.3 Choice of variance function

A first consideration in the choice of a functional model for the variance is based on the physical process underlying the data. For example, consider an experiment in which the variance of the response increases with the mean, but due to imprecision there is also a minimum level of variability. An appropriate model for the variance would be:

$$g(\mu_i, z_i, \theta) = (\theta_1 + \mu_i^{\theta_2})^{1/2}$$

In addition, graphical techniques can be used to determine a functional form for the heterogeneity of variance. Carroll and Ruppert (1988) argue for plotting transformations (identity, logarithm, 2/3 power, etc.) of absolute studentized residuals of an unweighted least squares fit against predicted values. Computing Spearman rank correlation coefficients and smoothing the scatter plots can be of help in interpreting the graphical displays.

3.4 Variance function estimation

Although graphical methods can be of assistance for determining the structural form of the variance model, they do not provide an efficient method for estimating the parameters of the model. Carroll and Ruppert (1988) give an overview of available methods for estimation of the parameter θ. The most popular methods, being pseudo-likelihood and its bias-corrected version restricted maximum likelihood, are based on transformations of the squared residuals from a preliminary fit.

Assume that the data originate from a normal distribution, then the log-likelihood of model (3.5) is by definition (constant omitted):

$$l(\beta, \theta, \sigma) = -n \log \sigma - \sum_{i=1}^{n} \log g(f(x_i, \beta), z_i, \theta) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \frac{[y_i - f(x_i, \beta)]^2}{g^2(f(x_i, \beta), z_i, \theta)} \qquad (3.6)$$

Substituting the preliminary estimator $\hat{\beta}_*$ into (3.6) for β leads to the pseudo-likelihood estimator for θ, i.e. maximize in σ and θ (3.6), given $\beta = \hat{\beta}_*$. Differentiation with respect to σ and θ shows that maximizing (3.6) is similar to a weighted least squares regression of the squared residuals $\left(y_i - f(\mathbf{x}_i, \hat{\beta}_*)\right)^2$ on the regression function $\sigma^2 g^2(f(\mathbf{x}_i, \hat{\beta}_*), z_i, \theta)$ with weights $g^{-4}(f(\mathbf{x}_i, \hat{\beta}_*), z_i, \theta)$. Thus, pseudo-likelihood estimation does not depend on the distribution of the data. Davidian and Haaland (1990) follow an alternative reasoning to place the problem of variance function estimation in the context of weighted least squares regression.

Estimating β in the above procedure involves a loss of degrees of freedom, which is not accounted for by the method of pseudo-likelihood. Hence, the estimate for θ is biased, with the amount of bias depending on the ratio *p/n.* Restricted maximum likelihood is based upon Bayesian ideas and corrects for the loss of degrees of freedom due to estimating β.

Outlying observations can severely affect the estimates. Therefore, Davidian and Haaland (1990), and Davidian and Giltinan (1995) propose, as an alternative, to replace squared residuals by absolute deviations and variances with mean values of absolute deviations divided by the variance function.

3.5 Software implementation

Weighted nonlinear regression in SAS is carried out by specifying the variable containing the weights with the *WEIGHT* statement in *PROC NLIN*. To carry out a weighted nonlinear regression in S-Plus, one has to implement the specific problem into an S-Plus function whose result is the square root of the objective function. More details and an example are given by Bates and Chambers (1993).

In some cases, nonlinear regression methods can be used to implement generalized least squares estimation. Following Carroll and Ruppert (1988), and Giltinan and Ruppert (1989), it is possible to solve explicitly the pseudo-likelihood criterion (3.6) for σ^2, yielding:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} \frac{\left[y_i - f(x_i, \hat{\beta}_*)\right]^2}{g^2(f(x_i, \hat{\beta}_*), z_i, \theta)}$$

Substituting this value back into (3.6) and after simplification, the pseudo-likelihood criterion is:

$$-\left[\frac{n}{2} + \frac{n}{2}\log\frac{1}{n}\sum_{i=1}^{n}\frac{\left[y_i - f(x_i, \hat{\beta}_*)\right]^2}{g^2(f(x_i, \hat{\beta}_*), z_i, \theta)} + \sum_{i=1}^{n}\log g(f(x_i, \hat{\beta}_*), z_i, \theta)\right] \quad (3.7)$$

Notice that $\sum_{i=1}^{n} \log g(f(x_i, \hat{\beta}_*), z_i, \theta) = n \log \tilde{g}(\theta)$, where $\tilde{g}(\theta)$ denotes the geometric mean of $g(f(x_i, \hat{\beta}_*), z_i, \theta)$. Putting the geometric mean $\tilde{g}(\theta)$ into (3.7) yields:

$$-\left[\frac{n}{2} + \frac{n}{2} \log \frac{1}{n} \sum_{i=1}^{n} \tilde{g}^2(\theta) \frac{\left[y_i - f(x_i, \hat{\beta}_*)\right]^2}{g^2(f(x_i, \hat{\beta}_*), z_i, \theta)}\right] \tag{3.8}$$

Maximizing (3.8) is equivalent to minimizing the quantity:

$$\sum_{i=1}^{n} \left[\frac{\tilde{g}(\theta)\left(y_i - f(x_i, \hat{\beta}_*)\right)}{g(f(x_i, \hat{\beta}_*), z_i, \theta)}\right]^2$$

Hence, estimating θ may be accomplished by regressing a dummy variable, which is identically zero for all *i* on the regression function:

$$\frac{\tilde{g}(\theta)\left(y_i - f(x_i, \hat{\beta}_*)\right)}{g(f(x_i, \hat{\beta}_*), z_i, \theta)} \tag{3.9}$$

When the variance function is of the type power of the mean, or exponential, the geometric mean can be written with the parameter θ outside the product, i.e.:

$$\tilde{g}(\theta) = \left(\prod_{i=1}^{n} v(f(x_i, \hat{\beta}_*), z_i)\right)^{\theta/n}$$

where *v* is a known function of the predicted values and *z*. Giltinan and Ruppert (1989) give details on how to implement generalized least squares estimation for the power of the mean variance model using *PROC NLIN* of SAS. However, they do not provide for more than one iteration in their SAS code. As shown in Appendix C, the program is easily extended to more iterations using the SAS-macro language. An S-Plus implementation of their approach is also given in Appendix C.

Transforming both sides is an alternative to weighted least squares and is related to the Box-Cox transform. Giltinan and Ruppert (1989) also describe how to implement this technique in SAS.

3.6 Application: metrazol-induced seizure threshold in rats

Infusion of rats with the drug metrazol (pentylenetetrazole) causes clonic seizures, which can be suppressed by therapy. The dosage in mg/kg of metrazol needed to induce tremor and clonic seizure is commonly known as the metrazol or pentylenetetrazole seizure threshold. Drugs that cause the seizure threshold to rise, are of potential use as anticonvulsant therapy in epileptic patients. Figure 3.1 summarizes the results from a study in which the rise in seizure threshold is measured following administration of different doses of the conventional antiepileptic drug diazepam combined with a fixed dose (10 mg/kg) of the experimental anticonvulsant drug loreclezole or its solvent. The purpose of the experiment is to investigate

how additional treatment with 10 mg/kg of loreclezole influences the dose response curve of diazepam. Figure 3.1 suggests the use of an extension of the four-parameter logistic model of (2.15), with expectation function:

$$f(\mathbf{x},\beta) = \left(\beta_0 + \beta_4 x_1\right) + \frac{\beta_1 + \beta_5 x_1}{1 + \exp\left((\beta_2 + \beta_6 x_1)(\beta_3 + \beta_7 x_1 - \log x_2)\right)} \qquad (3.10)$$

where x_1 is an indicator variable equal to 1 if the observation belongs to the loreclezole group and 0 otherwise; x_2 is the dose administered; β_0 is the lower asymptote; β_1 denotes the range (y_{max} - y_{min}) of the estimated response; β_2 and β_3 are respectively the slope and potency parameter; and β_4, β_5, β_6, and β_7 denote the effect of loreclezole treatment on respectively the lower asymptote, the range, the slope, and the potency (horizontal position) of the dose response curve of diazepam. For this problem, it makes mores sense to model the lower asymptote and the range (y_{max} - y_{min}) than to use both asymptotes as in (2.15). Indeed, effects on the lower asymptote will probably be reflected on the upper asymptote, unless the latter has some fixed upper bound. Model (3.10) will be referred to as the full or saturated model.

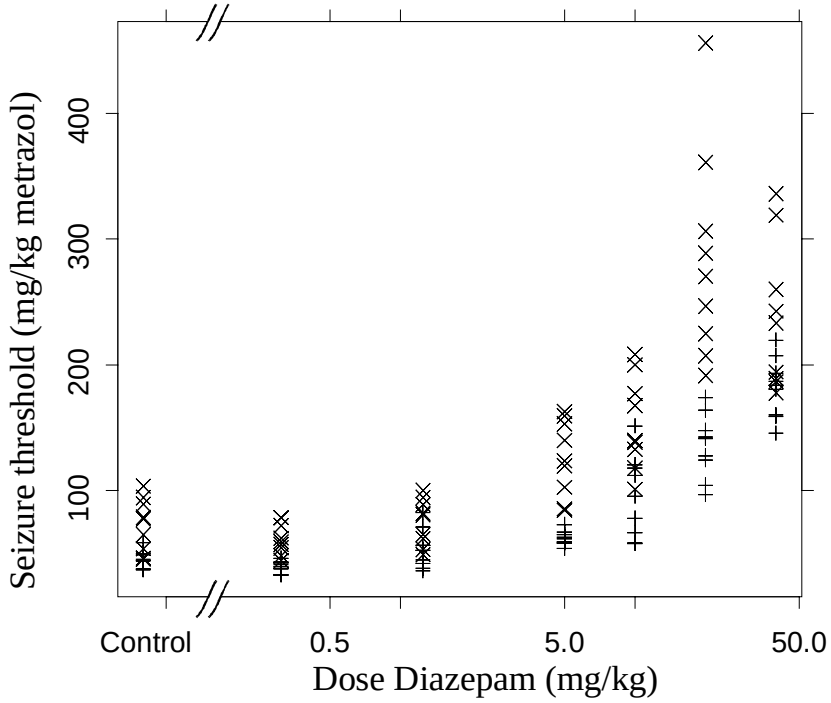

Figure 3.1 Dose response of diazepam in combination with loreclezole 10 mg/kg (×) or its solvent (+)

The individual data (Table C.1) and their analysis are shown in Appendix C. From Figure 3.1, it is apparent that there is serious heterogeneity of variance in the data. Another problem is the presence of animals that did not receive diazepam (i.e. zero-dose controls). Obviously, this zero-dose control cannot be positioned at -∞ where the logarithm of 0 would put it. On the other hand, omitting the zero-dose control values would mean losing valuable information. Different approaches are possible for positioning zero-dose controls on the log-dose axis. Considering the logarithms of the doses as an arithmetic series, it makes sense to keep the values equidistant to one another and to put the zero-dose equal to the lowest dose di-

vided by the constant ratio between doses. Hence, animals that did not receive diazepam would be considered as though they were treated with 0.08 mg/kg.

Starting values for the four parameters describing the sigmoidal shape are obtained from the control experiments using the same approach as in the previous chapter (Section C.2). The minimum and maximum of the data provide estimates for the lower asymptote β_0 (30) and range β_1 (190). Subtracting the lower asymptote and divisiding by the range (set at 200) yields a standardized (0-1) variable, which after logit transformation can be used in a linear regression to yield initial values for the slope (0.6) and potency parameter (3.0). The parameters β_4, β_5, β_6, and β_7 describing the displacement of the basic dose-response curve due to loreclezole are initially set to 0.

Plotting the observed versus the predicted values from an unweighted nonlinear least squares fit (Section C.3, Fig. C.2), shows that the model underestimates the response at the two lowest dose levels. Assessment of heterogeneity of variance is done using the studentized residuals from the unweighted nonlinear least squares fit. The Spearman rank-correlation between the absolute value of the studentized residuals and the predicted values is 0.58 with a significance level of less than 0.001. This indicates a strong heterogeneity of variance. Figure C.3 displays the logarithm of the absolute value of the studentized residuals versus the logarithm of the predicted values, together with a *lowess* smoothing. The plot shows a pronounced, almost linear, trend suggesting the variance should be modeled as a power of the mean. Linear regression yields a value of 1.16 for the slope, which suggests a constant coefficient of variation model. Figure C.4 shows that the fitted regression line is a reasonable estimate for the variance function.

The implementation of generalized least squares with pseudo-likelihood estimation of the variance parameter for the variance modeled as a power of the mean is shown in Section C.4, together with a sample run. The effect of the number of iterations on the parameter estimates is summarized in Table C.2. Convergence is obtained after about 5 cycles of the generalized least squares algorithm. For most parameters, the generalized least squares estimates differ substantially from the unweighted fit. The final pseudo-likelihood estimate for θ, the parameter of the variance function, is 1.132. The SAS implementation of generalized least squares is given in Section C.5. Convergence was less rapid in SAS using the DUD method for nonlinear least squares than in S-Plus. A sample run using 100 iterations shows that, at convergence, the results are comparable to those obtained using S-Plus.

Assessment in S-Plus of the generalized least squares fit after 5 iterations is carried out in Section C.6. A plot of observed values versus the fitted values (Fig. C.5) shows a marked improvement with regard to the lack of fit. Computation of the studentized residuals is changed to accommodate for the weights. The Spearman rank-correlation between the absolute value of the studentized residuals and the predicted values is 0.04 with a significance

level of 0.677. Plots of the residuals versus the fitted values (Fig. C.6), and of the logarithm of the absolute value of the studentized residuals versus the logarithm of the predicted values together with *lowess* smoothing (Fig. C.7), do not reveal the presence of a particular pattern. However, the normal probability plot of the studentized residuals (Fig. C.8) shows a considerable deviation from normality, with the distribution of the residuals being positively skewed (Fig. C.9). The correlation coefficient between the quantiles of the studentized residuals and the expected value under normality is 0.977, which is below the critical value of 0.982 for n = 100 and $\alpha = 0.01$. Thus, modeling heterogeneity of variance solved only one side of the problem.

An alternative strategy is to consider transformations of both the response and estimation function. Taking logarithms of both is a plausible transformation. Indeed, seizure threshold is measured as the dosage of metrazol needed to produce a certain physiological reaction, and previous work (Fleming, *et al*., 1972) has shown that such dosages are log-normally distributed. Furthermore, the constant coefficient of variation model is consistent with a log-normal distribution and implies a multiplicative rather than an additive error term.

The results of fitting the logarithm of the response versus the logarithm of the estimation function are shown in Section C.7. The estimated parameter values are close to the generalized least squares estimates. Assessment of the model is carried out in Section C.8. The fit is excellent (Fig. C.10) and there is no specific pattern in plots of the residuals (Fig. C.11 and C.12). In contrast to the results obtained earlier, the residuals are now normally distributed (Fig. C.13) with a correlation coefficient between the quantiles and their expected value under normality of 0.996, well above the critical value at $\alpha = 0.10$.

It is obvious that model (3.10) is over-parametrized. A more parsimonious model can be obtained with a backward elimination process, based on the extra sums of squares principle. Table C.3 in Section C.9 summarizes the results of backward elimination from model (3.10) (with logarithmic transformation). The final model, summarized in Table 3.1, does not contain the parameters β_5 and β_6 and provides an adequate fit to the data (Section C.10, Fig. C.14). Numeric measures of curvature yield a value of 0.214 for intrinsic and 1.14 for parameter effects curvature. The latter value is far above the threshold of 0.3. Construction of profile *t* plots (Fig. C.15) shows that, while curvature is a major problem for the parameters β_1 and β_3, Wald inference can still be applied to the remaining parameters.

For the reduced model, it is interesting to compare the results obtained using logarithmic transformation with those from the generalized least squares approach. Profile likelihood confidence intervals for the generalized least squares estimates can only be obtained after refitting the model using the estimated weights (Section C.11). Table 3.2 summarizes parameter estimates and confidence intervals of the generalized squares fit with pseudo-likelihood estimation of the variance function. Parameter estimates of both fitting strategies

are comparable to one another. Confidence intervals from the generalized least squares fit are wider than those from the logarithmic fit. This makes sense, since for normally distributed data, maximum likelihood estimation (which is equivalent to ordinary least squares) is superior.

The reduced model allows the effect of loreclezole to be described as an upward shift of the diazepam dose response curve of 20.7 mg/kg metrazol, combined with a displacement to the left of 0.41 times the potency of diazepam (Fig. 3.2). This combined effect indicates that, apart from its own effect on seizure threshold, loreclezole also enhances the action of diazepam. Indeed, only an upward shift of the complete dose-response curve would not involve a change in potency, since the latter is defined as the dosage needed to yield a response of half the total range.

Table 3.1 Estimated values and 95 % confidence intervals of the reduced model with logarithmic transformation

Parameter	Estimated Value	Wald C.I.	Profile likelihood C.I.
β_0	43.4	39.0, 47.9	39.0, 48.0
β_1	218	144, 291	167, 336
β_2	1.44	0.96, 1.93	1.04, 1.99
β_3	3.23	2.66, 3.80	2.79, 4.00
β_4	20.7	12.8, 28.6	12.9, 28.7
β_7	-0.882	-1.20, -0.57	-1.20, -0.58
exp(β_3)	*25.3*	*14.4, 44.6*	*16.3, 54.6*
exp(β_7)	*0.41*	*0.30, 0.56*	*0.30,0.56*

Table 3.2 Estimated values and 95 % confidence intervals of the reduced model fitted by generalized least squares

Parameter	Estimated Value	Wald C.I.	Profile likelihood C.I.
β_0	44.1	39.8, 48.4	39.6, 48.3
β_1	230	138, 322	168, 384
β_2	1.40	0.91, 1.90	0.99, 1.99
β_3	3.27	2.61, 3.93	2.76, 4.19
β_4	21.9	13.9, 29.9	13.9, 29.9
β_7	-0.885	-1.22, -0.55	-1.23, -0.56
exp(β_3)	*26.3*	*13.6, 51.0*	*15.7, 66.2*
exp(β_7)	*0.41*	*0.29, 0.58*	*0.29,0.57*

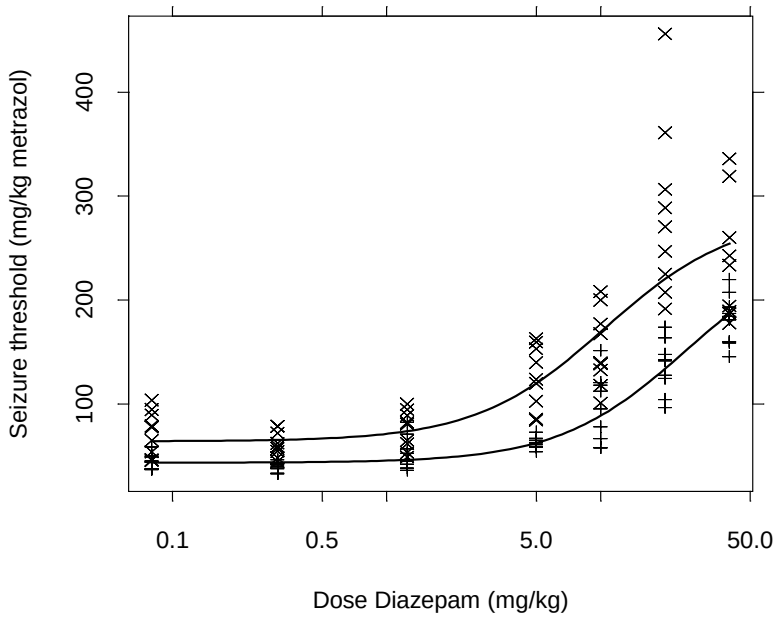

Figure 3.2 Dose response of diazepam in combination with loreclezole 10 mg/kg (×, upper curve) or its solvent (+, lower curve).

3.7 Summary

Generalized least squares is a useful technique when there is heteroscedasticity in the data. The pseudo-likelihood version is easily implemented when the variance function is a simple power of the mean model, and can be applied when the number of parameters is small relative to the number of replicates. However, heteroscedasticity is often accompanied by non-normality and sometimes a simple transformation of both the response and estimation function can be more appropriate, especially when there is a priori information on the distribution of the response.

4.
Repeated Measurements

In the experimental situations discussed so far, only one observation was made for each individual[1]. However, situations very often arise in which repeated observations are made in the same individual, allowing differential effects to be studied directly in the same animal, tissue, or cell culture. In such experiments, it is essential to recognize two kinds of variability: random variation within a given individual (intra-individual variability) and random variation among individuals (inter-individual variability). Special techniques are needed to characterize these two sources of variability and to use them properly for making reliable inference. Nonlinear mixed effects modeling provides a statistical framework for parametric analysis of nonlinear models with repeated measurements. An introduction to the theory of nonlinear mixed models is given by Davidian and Giltinan (1993). In a more recent work, the same authors (Davidian and Giltinan, 1995) provide an overview and thorough discussion of various techniques for the analysis of nonlinear models with repeated measurements.

4.1 Nonlinear mixed effects model

Let y_{ij} denote the j^{th} response, $j = 1,\ldots,n_i$ for the i^{th} individual $i = 1, \ldots,m$ and $\mathbf{x}_{ij}$ the vector of covariates (regressors). Assume that the relationship between y_{ij} and $\mathbf{x}_{ij}$ is modeled by a nonlinear function $f(\mathbf{x},\beta)$. The form of f is common to all individuals, but the parameter β is allowed to vary between the individuals. A hierarchical two-staged model is used to account for the different sources of variability. Corresponding to these two stages, there are also two types of inference possible: population and individual inference. The terms population parameters, population means, or fixed effects refer to population inference, while random effects refer to deviations of the individual parameters from their population mean. In pharmacological applications population inference is usually the major objective. However, there are problem areas, such as calibration in bioassays, for which individual inference is the primary target.

4.1.1 Stage 1: Intra-individual variation

In the first stage, the mean and covariance structure for a given individual are specified. The mean response for individual i depends on the parameter vector β_i, specific for that individual, as $\mathrm{E}(y_{ij}|\beta_i) = f(x_{ij},\beta_i)$. The response-vector $\mathbf{y}_i = \left[y_{i1},\ldots,y_{in_i}\right]^{\mathrm{T}}$ for the i^{th} individual is assumed to follow the model:

$$\mathbf{y}_i = \mathbf{f}_i(\beta_i) + \mathbf{e}_i \tag{4.1}$$

[1] The term *individual* is used to identify a given entity on which repeated observations are made.

with $\mathbf{f}_i(\beta_i) = \left[f(x_{i1},\beta_i),\ldots, f(x_{in_i},\beta_i)\right]^{\mathrm{T}}$; and $\mathbf{e}_i = \left[\varepsilon_{i1},\ldots,\varepsilon_{in_i}\right]^{\mathrm{T}}$ where ε_{ij} is the random error reflecting uncertainty in the response, given the i^{th} individual and $\mathrm{E}(\mathbf{e}_i|\beta_i) = \mathbf{0}$.

Model (4.1) describes the systematic and random variation associated with measurements on the i^{th} individual. It is reasonable to assume that the pattern of intra-individual variation is the same across individuals. A very general way to model the common intra-individual covariance structure is then provided by:

$$\begin{aligned}\mathrm{Cov}(\mathbf{e}_i|\beta_i) &= \sigma^2\mathbf{G}_i^{1/2}(\beta_i,\theta)\Gamma_i(\alpha)\mathbf{G}_i^{1/2}(\beta_i,\theta)\\ &= \mathbf{R}_i(\beta_i,\xi), \quad \xi = (\sigma,\theta^{\mathrm{T}},\alpha^{\mathrm{T}})^{\mathrm{T}}\end{aligned}$$

The $(n_i \times n_i)$ diagonal matrix $\mathbf{G}_i^{1/2}(\beta_i,\theta)$ is used to model intra-individual variance and is analogous to the function $g(\mu_i,z_i,\theta)$ of the previous chapter:

$$\mathbf{G}_i(\beta_i,\theta) = \mathrm{diag}\left[g^2(\mu_{i1},z_{i1},\theta_i),\ldots,g^2(\mu_{in_i},z_{in_i},\theta_i)\right]$$

with the variance now dependent on the regression parameter β_i through the mean response. The $(n_i \times n_i)$ symmetric matrix $\Gamma_i(\alpha)$ represents the correlations among the observations of the i^{th} individual. When repeated observations are made over time, the matrix $\Gamma_i(\alpha)$ may reflect the serial correlation that is present. For example, the simplest case, when the n_i observations are equally spaced in time, is the autoregressive model of order 1, which corresponds to:

$$\Gamma_i(\alpha) = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n_i-1} \\ & 1 & \alpha & \cdots & \alpha^{n_i-2} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \alpha \\ & & & & 1 \end{bmatrix}$$

The model for intra-individual variation is completed by specifying the distributional assumptions for the error term $\mathbf{e}_i$:

$$\mathbf{e}_i|\beta_i \sim N\left(\mathbf{0},\mathbf{R}_i(\beta_i,\xi)\right)$$

which, in terms of the random effects $\mathbf{b}_i = \beta_i - \beta$, is equivalent to:

$$\mathbf{e}_i|\mathbf{b}_i \sim N\left(\mathbf{0},\mathbf{R}_i(\beta_i,\xi)\right) \tag{4.2}$$

4.1.2 Stage 2: Inter-individual variation

In the second stage, variation among individuals is characterized by specifying a model for the regression parameters β_i in (4.1). A general model for the $(p \times 1)$ vector of individual regression parameters β_i is given by:

$$\beta_i = \mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i) \tag{4.3}$$

where **d** is a *p*-dimensional vector-valued function, $\mathbf{a}_i$ an (*a* x 1) vector of covariate values specific for individual *i*, β an $(r \times 1)$ vector of fixed parameters, and $\mathbf{b}_i$ a $(k \times 1)$ vector of

random effects associated with the i^{th} individual. In addition, the random effects $\mathbf{b}_i$ are considered to be normally distributed with covariance matrix **D**:

$$\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$$

where **D** is a $k \times k$ covariance matrix.

A special case of (4.3), often used in practice, is given by the linear form:

$$\beta_i = \mathbf{A}_i \beta + \mathbf{b}_i \tag{4.4}$$

where $\mathbf{A}_i$ is a ($p \times$ r) design matrix corresponding to the systematic (fixed) part of the model.

A more general, but still linear specification of (4.4) is given by:

$$\beta_i = \mathbf{A}_i \beta + \mathbf{B}_i \mathbf{b}_i \tag{4.5}$$

where the ($p \times k$) design matrix $\mathbf{B}_i$ is associated with the random effects $\mathbf{b}_i$. The expression in (4.5) allows some of the β_i to be fixed and others to be random.

4.2 Strategies for parameter estimation

In the nonlinear mixed effects model, as described above, the random component $\mathbf{b}_i$ appears in the model through the nonlinear function *f*. Hence, the effect of inter-individual variation on the response is complicated and it is not possible to write down a distribution for the $\mathbf{y}_i$. Thus, standard techniques such as maximum likelihood may be difficult to implement. Therefore, methods for estimation and inference are based on approximations that allow a distribution to be specified for $\mathbf{y}_i$. Historically, the techniques found their origin and major applications in the area of pharmacokinetics.

The methods available for estimating the parameters of the nonlinear mixed effects model fall into two broad classes. The first approach is based on the ability to obtain an estimate for the parameter β_i for each individual. Based on these individual estimates, the remaining parameters σ, α, θ, and **D** are estimated. For this approach to be successful, sufficient data must be available for each individual and the nonlinear least squares procedure must converge for all individuals. This class of methods is referred to as two-stage methods since they involve two stages: initial estimation of the β_i and subsequent estimation of the other parameters. The second class of methods is based on a linearization of the nonlinear mixed effects model by a Taylor series in the random effects $\mathbf{b}_i$. This set of techniques is of particular importance when some individuals lack sufficient data for individual fitting.

4.2.1 Two-stage methods

4.2.1.1 Construction of individual estimates

The method of generalized least squares is extended to allow for a general covariance matrix:

$$\mathbf{R}_i(\beta_i, \xi) = \sigma^2 \mathbf{S}_i(\beta_i, \gamma), \quad \xi = (\sigma, \theta^T, \alpha^T)^T, \quad \gamma = (\theta^T, \alpha^T)^T$$

The general least squares algorithm is then changed as follows:

1. In *m* separate unweighted regressions obtain preliminary estimates $\hat{\beta}_i^{(p)}$ for each individual $i=1,\ldots,m$
2. Residuals from these preliminary fits are used to estimate σ and γ and estimated weight matrices are formed by:

$$\hat{\mathbf{W}}_i = \mathbf{S}_i^{-1}(\hat{\beta}_i^{(p)}, \hat{\gamma})$$

3. Using the estimated weight matrices, re-estimate the β_i by *m* separate minimizations in β_i of the objective functions:

$$\left(\mathbf{y}_i - \mathbf{f}_i(\beta_i)\right)^{\mathrm{T}} \hat{\mathbf{W}}_i \left(\mathbf{y}_i - \mathbf{f}_i(\beta_i)\right)$$

yielding a new set of estimators.

4. Update the preliminary estimators with the new estimators and return to step 2.
5. Iterate at least once.

The final estimates are the individual generalized least squares estimates $\hat{\beta}_i^*$.

Estimation of the covariance parameters in step 2 can be carried out by maximization in σ and γ of the pseudo-likelihood criterion:

$$-\sum_{i=1}^{m} \log\left|\sigma^2 \mathbf{S}_i(\hat{\beta}_i^{(p)}, \gamma)\right| + \left(\mathbf{y}_i - \mathbf{f}_i(\hat{\beta}_i^{(p)})\right)^{\mathrm{T}} \mathbf{S}_i^{-1}(\hat{\beta}_i^{(p)}, \gamma)\left(\mathbf{y}_i - \mathbf{f}_i(\hat{\beta}_i^{(p)})\right) \Big/ \sigma^2$$

Alternatively, this objective function could be replaced by the restricted maximum likelihood criterion.

4.2.1.2 Estimation of population parameters

Once individual parameter estimates have been obtained, there are several possibilities for constructing estimates of β and **D**. One method, referred to as the *standard two-stage method*, considers the generalized least squares estimates $\hat{\beta}_i^*$ as if they were the true values for the parameters β_i. Standard two-stage estimates are then constructed by simply computing the sample mean and covariance of the $\hat{\beta}_i^*$. The major drawback of this naïve approach is that the uncertainty in estimating the β_i is not taken into account. Davidian and Giltinan (1995) show that the standard two-stage estimator for the covariance matrix **D** is upwardly biased and do not recommend this procedure. An alternative is provided by applying asymptotic theory for nonlinear regression. Then, under the simplified model of (4.4), maximum likelihood estimates can be obtained by using the iterative EM algorithm. Davidian and Giltinan (1993, 1995) provide full details on the derivation and necessary calculations of this *global two-stage method*.

4.2.2 Linearization methods

When individual data are sparse or do not yield convergence in all cases, it is impossible to apply the above two-stage procedure. An alternative methodology is provided by the class of linearization methods. The methods are based on a linearization of the nonlinear mixed effects model by a Taylor series expansion in the random effects $\mathbf{b}_i$.

4.2.2.1 First-order linearization

Assuming the nonlinear mixed effect as described earlier, with the assumption of normality of both random components, we have:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{f}_i(\beta_i) + \mathbf{e}_i, \qquad \mathbf{e}_i|\mathbf{b}_i \sim N\left(\mathbf{0}, \mathbf{R}_i(\beta_i, \xi)\right) \\ \beta_i &= \mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i), \qquad \mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D}) \end{aligned}$$

With $\mathbf{R}_i^{1/2}(\beta_i, \xi)$ the Cholesky decomposition of $\mathbf{R}_i(\beta_i, \xi)$, it is possible to write:

$$\mathbf{e}_i = \mathbf{R}_i^{1/2}(\beta_i, \xi)\varepsilon_i$$

where $\varepsilon_i \sim N\left(\mathbf{0}, \mathbf{I}_{n_i}\right)$.

Writing out β_i as the nonlinear function $\beta_i = \mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i)$, the first stage of the model may be rewritten as:

$$\mathbf{y}_i = \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i)\right) + \mathbf{R}_i^{1/2}\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i), \xi\right)\varepsilon_i \tag{4.6}$$

A Taylor series expansion of (4.6) in $\mathbf{b}_i$, about the mean value E($\mathbf{b}_i$) = $\mathbf{0}$, with the first two terms in the expansion for $\mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i)\right)$ and only the leading term of the expansion of $\mathbf{R}_i^{1/2}\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i), \xi\right)\varepsilon_i$ retained[1], yields:

$$\mathbf{y}_i \approx \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{0})\right) + \mathbf{F}_i(\beta, \mathbf{0})\Delta_{bi}(\beta, \mathbf{0})\mathbf{b}_i + \mathbf{R}_i^{1/2}\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{0}), \xi\right)\varepsilon_i \tag{4.7}$$

where $\mathbf{F}_i(\beta, \mathbf{0})$ is the $(n_i \times p)$ matrix of partial derivatives of $\mathbf{f}_i(\beta_i)$ with respect to β_i evaluated at $\beta_i = \mathbf{d}(\mathbf{a}_i, \beta, \mathbf{0})$, and $\Delta_{bi}(\beta, \mathbf{0})$ the $(p \times k)$ matrix of derivatives of $\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i)$ with respect to $\mathbf{b}_i$ and evaluated at $\mathbf{b}_i = \mathbf{0}$.

With $\mathbf{Z}_i(\beta, \mathbf{0}) = \mathbf{F}_i(\beta, \mathbf{0})\Delta_{bi}(\beta, \mathbf{0})$ and $\mathbf{e}_i^* = \mathbf{R}_i^{1/2}\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{0}), \xi\right)\varepsilon_i$, (4.7) can be written as:

$$\mathbf{y}_i \approx \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{0})\right) + \mathbf{Z}_i(\beta, \mathbf{0})\mathbf{b}_i + \mathbf{e}_i^* \tag{4.8}$$

The latter approximation has a form similar to the linear mixed effects model (4.5), with the random effects $\mathbf{b}_i$ and within-individual errors $\mathbf{e}_i^*$ entering the model in the same linear and additive manner. Approximation (4.8) allows the marginal mean and covariance of $\mathbf{y}_i$ to be specified directly:

$$\begin{aligned} \mathrm{E}(\mathbf{y}_i) &\approx \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{0})\right) \\ \mathrm{Cov}(\mathbf{y}_i) &\approx \mathbf{R}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i), \xi\right) + \mathbf{Z}_i(\beta, \mathbf{0})\mathbf{D}\mathbf{Z}_i^{\mathrm{T}}(\beta, \mathbf{0}) \\ &\equiv \mathbf{V}_i(\beta, \mathbf{0}, \omega) \end{aligned} \tag{4.9}$$

where ω is a vector containing the collection of the intra-individual covariance parameter ξ and the distinct elements of the inter-individual covariance matrix $\mathbf{D}$. When the $\mathbf{b}_i$ and $\mathbf{e}_i^*$ are

[1] Retaining two terms for the mean function and only the leading term for the intra-individual error component is motivated because misspecification of first-moments is more serious than that of second moments (Davidian and Giltinan, 1995).

assumed to be normally distributed, the marginal distribution of $\mathbf{y}_i$ is approximately normally distributed with first and second moments as given above in (4.9).

Two general strategies for inference based on the first-order linearization have been developed. The first inference method, known as the *first-order method* and proposed by Beal and Sheiner (1982), uses maximum likelihood estimation under the assumption that the approximations in (4.7) and (4.8) are exact and the $\mathbf{b}_i$ and $\mathbf{e}_i^*$ are normally distributed. However, as Davidian and Giltinan (1995) argue, this method can be sensitive to non-normality of the response and misspecification of the individual covariance structure.

An alternative class of methods is based on the assumption that the mean and covariance specification in (4.9) are exact instead of approximate and uses an extension of generalized least squares. Here, several strategies are possible. With $\mathbf{W}_i(\beta,\mathbf{0},\omega) = \mathbf{V}_i^{-1}(\beta,\mathbf{0},\omega)$, one procedure, proposed by Davidian and Giltinan (1995), works as follows:

1. Estimate β by a preliminary estimator $\hat{\beta}^{(p)}$ using ordinary least squares on the pooled data.
2. Using the residuals from this preliminary fit, estimate the covariance parameters ω by $\hat{\omega}$ and form the *m* estimated weight matrices $\mathbf{W}_i(\hat{\beta}_i^{(p)},\mathbf{0},\hat{\omega}))$
3. Using the estimated weight matrices, re-estimate β, by minimizing in β:

$$\sum_{i=1}^{m}\left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{0})\right)\right]^{\mathrm{T}}\mathbf{W}_i(\hat{\beta}^{(p)},\mathbf{0},\hat{\omega})\left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{0})\right)\right]$$

yielding a new set of estimators.

4. Update the preliminary estimators with the new estimators and return to step 2.
5. Iterate at least once.

Estimation of the covariance parameters ω can be accomplished by maximization in ω of the pseudo-likelihood criterion:

$$PL(\hat{\beta}^{(p)},\omega) =$$
$$-\sum_{i=1}^{m}\log\left|\mathbf{V}_i(\hat{\beta}^{(p)},\mathbf{0},\omega)\right| + \left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i,\hat{\beta}^{(p)},\mathbf{0})\right)\right]^{\mathrm{T}}\mathbf{V}_i^{-1}(\hat{\beta}^{(p)},\mathbf{0},\omega)\left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i,\hat{\beta}^{(p)},\mathbf{0})\right)\right]$$

or alternatively, the restricted maximum likelihood criterion:

$$REML(\hat{\beta}^{(p)},\omega) = PL(\hat{\beta}^{(p)},\omega) - \sum_{i=1}^{m}\log\left|\mathbf{X}_i^{\mathrm{T}}(\hat{\beta}^{(p)},\mathbf{0})\mathbf{V}_i^{-1}(\hat{\beta}^{(p)},\mathbf{0},\omega)\mathbf{X}_i(\hat{\beta}^{(p)},\mathbf{0})\right|$$

where $\mathbf{X}_i(\beta,\mathbf{0}) = \mathbf{F}_i(\beta,\mathbf{0})\Delta_{\beta i}(\beta,\mathbf{0})$ and $\Delta_{\beta i}(\beta,\mathbf{0})$ is the $(p \times r)$ matrix of derivatives of $\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i)$ with respect to β, evaluated at $\mathbf{b}_i = \mathbf{0}$.

Pseudo-likelihood or restricted maximum-likelihood estimation of the covariance parameters does not depend on normality of the marginal distribution of the $\mathbf{y}_i$.

A variation on the above generalized least squares algorithm was proposed by Vonesh and Carter (1992). The main difference between the two methods is in the estimation of the covariance parameter. Vonesh and Carter base their estimate of the covariance parameter on estimates of the random effects $\mathbf{b}_i$. Estimation of ξ and $\mathbf{D}$, the two components of ω, proceeds in two separate stages. For a general intra-individual covariance matrix, step 2 of the generalized least squares algorithm consists of another iterative procedure. Detailed descriptions of the algorithm are given in Davidian and Giltinan (1993, 1995) and Vonesh and Carter (1992). An advantage of the Vonesh and Carter procedure is that no starting values for the covariance matrix $\mathbf{D}$ are required.

Without assumptions on the normality of the marginal distribution of the $\mathbf{y}_i$, for m large, the generalized least squares estimator of the parameters $\hat{\beta}_{GLS}$ is asymptotically normal with mean β and covariance matrix

$$\Sigma_{GLS} = \left(\sum_{i=1}^{m} \mathbf{X}_i^{T}(\beta,\mathbf{0})\mathbf{V}_i^{-1}(\beta,\mathbf{0},\omega)\mathbf{X}_i(\beta,\mathbf{0}) \right)^{-1} \tag{4.10}$$

Under additional regularity assumptions, estimates obtained from the Vonesh-Carter method are also asymptotically normal with the same covariance matrix. An estimate of Σ_{GLS} is obtained by evaluating (4.10) at the final estimates of β and ω.

4.2.2.2 Conditional first-order linearization

Using the linearization of (4.8) implies that the marginal mean as given in (4.9) does not reflect intra-individual variation. When the intra-individual variation is large, the first-order linearization may be a poor approximation leading to biased and imprecise estimation. Lindstrom and Bates (1990) propose using a refinement of the linearization (4.8) to obtain a better approximation. Instead of a Taylor expansion about the expectation of the random effects $\mathbf{b}_i = \mathbf{0}$ as in (4.7), they consider linearization about some value $\mathbf{b}_i^*$ that is closer to $\mathbf{b}_i$ than its expected value $\mathbf{0}$. Using a Taylor expansion analogous to (4.7), yields:

$$\mathbf{y}_i \approx \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i^*)\right) + \mathbf{F}_i(\beta,\mathbf{b}_i^*)\Delta_{bi}(\beta,\mathbf{b}_i^*)(\mathbf{b}_i - \mathbf{b}_i^*) + \mathbf{R}_i^{1/2}\left(\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i^*),\xi\right)\varepsilon_i \tag{4.11}$$

where $\mathbf{F}_i(\beta,\mathbf{b}_i^*)$ is the $(n_i \times p)$ matrix of partial derivatives of $\mathbf{f}_i(\beta_i)$ with respect to β_i evaluated at $\beta_i = \mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i^*)$, and $\Delta_{bi}(\beta,\mathbf{b}_i^*)$ is the $(p \times k)$ matrix of derivatives of $\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i)$ with respect to $\mathbf{b}_i$ and evaluated at $\mathbf{b}_i = \mathbf{b}_i^*$.

With $\mathbf{Z}_i(\beta,\mathbf{b}_i) = \mathbf{F}_i(\beta,\mathbf{b}_i)\Delta_{bi}(\beta,\mathbf{b}_i)$ and $\mathbf{e}_i^* = \mathbf{R}_i^{1/2}\left(\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i^*),\xi\right)\varepsilon_i$, (4.11) can be written as:

$$\mathbf{y}_i \approx \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i^*)\right) - \mathbf{Z}_i(\beta,\mathbf{b}_i^*)\mathbf{b}_i^* + \mathbf{Z}_i(\beta,\mathbf{b}_i^*)\mathbf{b}_i + \mathbf{e}_i^* \tag{4.12}$$

It follows that for $\mathbf{b}_i$ close to $\mathbf{b}_i^*$, the approximate marginal mean and covariance of y is:

$$
\begin{aligned}
\mathrm{E}(\mathbf{y}_i) &\approx \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i^*)\right) - \mathbf{Z}_i(\beta, \mathbf{b}_i^*)\mathbf{b}_i^* \\
\mathrm{Cov}(\mathbf{y}_i) &\approx \mathbf{R}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i^*), \xi\right) + \mathbf{Z}_i(\beta, \mathbf{b}_i^*)\mathbf{D}\mathbf{Z}_i^{\mathrm{T}}(\beta, \mathbf{b}_i^*) \\
&\equiv \mathbf{V}_i(\beta, \mathbf{b}_i^*, \omega)
\end{aligned}
\tag{4.13}
$$

with ω the same as above.

The fundamental idea of Lindstrom and Bates (1990) is to obtain a suitable estimate of $\mathbf{b}_i$, insert it as the value for $\mathbf{b}_i^*$ in (4.13), and, treating the estimate as fixed, use generalized least squares to estimate β and ω. In their original proposal, they make some restrictions about the model. They assume that the inter-individual regression function **d** has the linear form as in (4.5), and that the intra-individual covariance matrix does not depend on β_i (or equivalently $\mathbf{b}_i$) but depends on *i* only through its dimension. Davidian and Giltinan (1995) relax these restrictions to allow for a general inter-individual regression function and a general intra-individual covariance matrix. In analogy to the linear mixed effects model, values of β and $\mathbf{b}_i$ are estimated that jointly maximize the marginal likelihood:

$$
-\sum_{i=1}^{m} \log|\mathbf{D}| + \mathbf{b}_i^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{b}_i + \log\left|\mathbf{R}_i(\xi)\right| + \left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i)\right)\right]^{\mathrm{T}} \mathbf{R}_i^{-1}(\xi)\left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \mathbf{b}_i)\right)\right]
\tag{4.14}
$$

where **D** and ξ are assumed to be known. Estimation of the parameters proceeds in two-steps:

1. Maximize β in and $\mathbf{b}_i$ (4.14), with **D** and ξ replaced by their current estimates $\hat{\mathbf{D}}$ and $\hat{\xi}$, yielding estimates $\hat{\beta}_0$ and $\hat{\mathbf{b}}_i$.
2. Estimate β, **D** and ξ as the values $\hat{\beta}$, $\hat{\mathbf{D}}$ and $\hat{\xi}$ that maximize the approximate marginal normal log-likelihood:

$$
\begin{aligned}
L_{LB}(\beta, \omega) &= -\sum_{i=1}^{m} \log\left|\mathbf{V}_i(\hat{\beta}_0, \hat{\mathbf{b}}_i, \omega)\right| \\
&+ \left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \hat{\mathbf{b}}_i)\right) + \mathbf{Z}_i(\hat{\beta}_0, \hat{\mathbf{b}}_i)\hat{\mathbf{b}}_i\right]^{\mathrm{T}} \mathbf{V}_i^{-1}(\hat{\beta}_0, \hat{\mathbf{b}}_i, \omega)\left[\mathbf{y}_i - \mathbf{f}_i\left(\mathbf{d}(\mathbf{a}_i, \beta, \hat{\mathbf{b}}_i)\right) + \mathbf{Z}_i(\hat{\beta}_0, \hat{\mathbf{b}}_i)\hat{\mathbf{b}}_i\right]
\end{aligned}
$$

with ω as in (4.9). Alternatively, a restricted maximum likelihood criterion may be maximized:

$$
REML_{LB}(\beta, \omega) = L_{LB}(\beta, \omega) - \sum_{i=1}^{m} \log\left|\mathbf{X}_i^{\mathrm{T}}(\hat{\beta}_0, \mathbf{b}_i)\mathbf{V}_i^{-1}(\hat{\beta}_0, \mathbf{b}_i, \omega)\mathbf{X}_i(\hat{\beta}_0, \mathbf{b}_i)\right|
$$

where $\mathbf{X}_i(\beta,\hat{\mathbf{b}}_i) = \mathbf{F}_i(\beta,\hat{\mathbf{b}}_i)\Delta_{\beta i}(\beta,\hat{\mathbf{b}}_i)$ and $\Delta_{\beta i}(\beta,\hat{\mathbf{b}}_i)$ is the $(p \times r)$ matrix of derivatives of $\mathbf{d}(\mathbf{a}_i,\beta,\mathbf{b}_i)$ with respect to β, evaluated at $\mathbf{b}_i = \hat{\mathbf{b}}_i$.

The algorithm is iterated until convergence. Standard errors for $\hat{\beta}$ are obtained from the Hessian of the likelihood function after the last iteration step.

Davidian and Giltinan (1995) show that this two-step procedure is essentially equivalent to a generalized least squares approach, where the approximate marginal log-likelihood is replaced by a pseudo-likelihood.

4.3 Software implementation

Nonlinear mixed effects modeling inherits the computational difficulties of both nonlinear modeling and linear mixed effects models. This is the major reason why applications of these techniques have been restricted to the rather specialized field of population pharmacokinetics. However, implementations of the linearization algorithms in S-Plus and SAS have recently become available.

Standard implementations of the global two-stage procedure in S-Plus or SAS are not available, but it is easily implemented for models with a simple structure of the variance function and without intra-individual correlation.

In S-Plus, the *nlme* library, developed by Pinheiro and Bates (1995) implements the Lindstrom and Bates procedure with both approximate marginal likelihood and restricted maximum likelihood. The latest edition (version 2.0) does allow for a general intra-individual covariance matrix. This collection of functions requires SPlus version 3.3 or higher and is available from *StatLib*.

Vonesh (1995) has written the SAS-macro *MIXNLIN*, which implements the Vonesh and Carter method as well as the procedure of Lindstrom and Bates. The macro makes use of the SAS matrix language IML and allows general functions of the random effects. The *MIXNLIN* macro was obtained from the author on simple request. Another SAS-macro *NLINMIX*, based on *PROC MIXED*, was developed by R. Wolfinger and is described in Littell, *et al.* (1996). In addition to the Vonesh-Carter and Lindstrom-Bates methods, the macro also implements a generalized estimating equations approach. The *NLINMIX* macro uses different algorithms than *MIXNLIN* and requires SAS version 6.11 or higher. *NLINMIX* can be obtained from SAS Institute World Wide Web (*http:://www.sas.com/*).

4.4 Applications

4.4.1 Inhibition of thromboxane A_2 formation in blood samples

Thromboxane A_2 (TXA_2) is synthesized by blood platelets and is an extremely potent inducer of platelet aggregation and vasoconstriction. Drugs that inhibit the formation of TXA_2 have potential use in thrombolytic therapy. The experimental compound ridogrel is a potent inhibitor of thromboxane A_2 synthetase, an enzyme involved in the formation of TXA_2. Since, formation of TXA_2 depends on intracellular calcium, addition of the calcium ionophore calcimycin (A23187) to blood induces platelets to synthesize TXA_2. TXA_2 itself is very unstable and undergoes a nonenzymic hydrolysis to Thromboxane B_2 (TXB_2), which is inactive and stable. Hence, the plasma concentration of TXB_2 is used as an index for the formation of TXA_2.

In a recent study, Beetens, *et al.* (1996) investigated the effect of ridogrel on calcimycin-induced TXA_2 formation in whole blood. Blood was drawn from healthy volunteers, free of medication for at least two weeks. Blood samples were randomly divided into different aliquots, to which 25 µM of the agonist calcimycin together with a known amount of ridogrel ($0 - 10^{-5}$ M), was added. After 1 hour of incubation, test tubes were cooled on ice and plasma was obtained by centrifugation. Subsequently, plasma concentration of TXB_2 was assessed by Enzyme-Immuno-Assay (EIA). The purpose of the experiment was to provide an estimate of the dose of ridogrel that yielded a 50 % reduction in the formation of TXB_2.

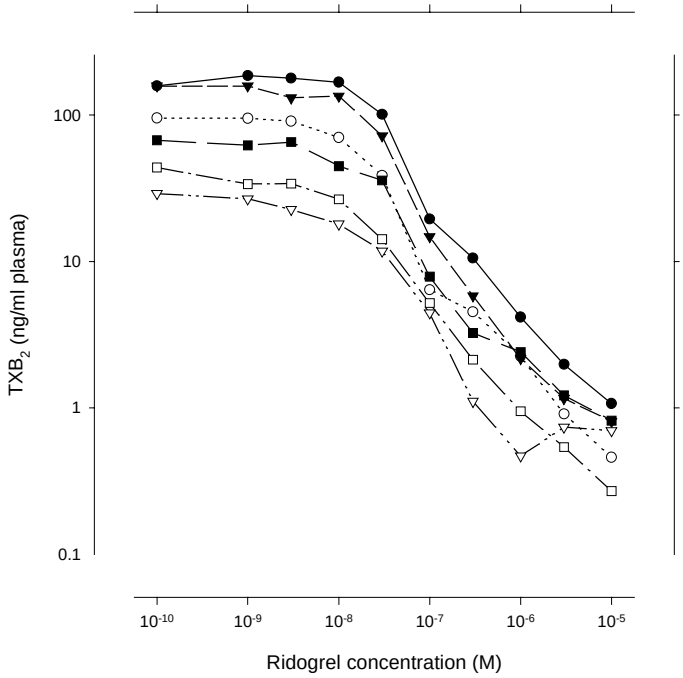

Figure 4.1 Effect of ridogrel on formation of TXB_2 in blood samples from six volunteers

The individual data and details on the analysis of this study are contained in Appendix D. Figure 4.1 depicts the individual concentration response profiles. Control measurements, obtained in the absence of the drug, are considered to be made at a concentration of 10^{-10} M. The data (Table D.1) consist of repeated measurements on blood samples from volunteers. These data are balanced, since for each volunteer all concentrations are tested. Two sources of variability can be recognized: within blood samples and between blood samples. However, these are not longitudinal data, since different aliquots are randomly assigned to different concentrations.

Figure 4.1 suggests fitting the four parameter logistic model with logarithms of both the estimation function and the response, i.e.:

$$\log y_{ij} = \log\left[\left(\beta_0 + b_{i0}\right) + \frac{\beta_1 + b_{i1}}{1 + \exp\left((\beta_2 + b_{i2})(\beta_3 + b_{i3} - \log x_{ij})\right)}\right] + e_{ij} \qquad (4.15)$$

where y_{ij} is the j^{th} TXB_2 measurement for the i^{th} volunteer, x_{ij} is the j^{th} concentratrion of the drug for the i^{th} volunteer, $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T$ is a vector of fixed population effects and $\mathbf{b}_i=(b_{i0}, b_{i1}, b_{i2}, bi_{i3})^T$ is a vector of individual random effects with $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$. Initially, all four parameters will be assumed to have both a fixed and random component as in (4.15). However, Figure 4.1 already suggests that only the vertical asymptotes are different among subjects. Therefore, more parsimonious models with fewer random effects will also be considered. Figure 4.1 is also useful in estimating starting values β^0 for the fixed parameter, yielding $\beta^0 = (1, 100, -0.14, -16)^T$.

Analysis in S-Plus (Section D.2) starts with defining a function for the partial derivatives of (4.15), with respect to the fixed parameters. Improved estimates of the initial values are then obtained by fitting a model without random effects to the pooled data. Parameter estimates and standard errors of the pooled data approach are given in Table 4.1.

Traditionally, a naïve approach, corresponding to the standard two-stage method of 4.2.1.2., has always been applied to this kind of data, i.e. the population parameters are estimated by simple means of the individual nonlinear least squares parameter estimates, and standard errors of the population parameter estimates are calculated as standard errors of the means. Section D.3 shows the computation of the individual estimates, their means and standard errors. Parameter estimates for the individual blood samples and estimates of the population means are summarized in Table 4.1, together with the pooled data approach. The main variation among the different samples is found for β_1, the parameter related to the range of the response. For this parameter, the two approaches also yield considerably different results (78.7 versus 99.8). This difference is trivial, in the sense that the pooled data estimate of the vertical upper asymptote of model (4.15) is in principle the geometric mean (since we take logarithms of both sides), while the standard two-stage approach estimate is, by definition, the arithmetic mean of the individual estimates. Note that the simple two-stage approach rec-

ognizes the two sources of variability and yields a substantially improved estimate of the parameter of interest β_3.

An attempt was made to make a crude implementation of the global two-stage method (Section D.4) for this specific problem. Since it was not possible to verify the output, the results, obtained for 200 iterations of the algorithm, must be interpreted with extreme care and will not be discussed further.

Table 4.1 Individual parameter estimates and population means for model (4.15), standard two-stage and pooled data approach.

Volunteer	β_0	β_1	β_2	β_3
1	1.002	193.2	-1.114	-17.675
2	0.365	109.7	-0.964	-18.365
3	0.833	161.0	-1.280	-17.718
4	0.601	24.1	-1.648	-17.279
5	0.918	68.6	-1.116	-17.789
6	0.209	42.0	-0.935	-18.151
Population mean	0.655	99.8	-1.176	-17.829
SE	0.130	27.5	0.107	0.156
Pooled data fitting	0.662	78.7	-1.139	-17.827
SE	0.200	14.64	0.191	0.481

Table 4.2 Individual parameter estimates and population means for model (4.15), Lindstrom-Bates approach (nlme)

Volunteer	β_0	β_1	β_2	β_3
1	0.969	196.2	-1.124	-17.729
2	0.475	97.0	-1.049	-18.012
3	0.824	148.5	-1.187	-17.741
4	0.549	26.5	-1.210	-17.772
5	0.780	70.4	-1.198	-17.663
6	0.273	42.9	-0.998	-18.121
Population Mean	0.645	96.9	-1.127	-17.840
SE	0.122	25.2	0.069	0.166

The *nlme* suite of S-functions, developed by Pinheiro and Bates (1995) was used to apply the Lindstrom-Bates approach to the data. (cf. Section D.5). With regard to the covariance structure, it was assumed that responses within individuals are uncorrelated and the covariance matrix of the random effects is of the unstructured type. It was extremely difficult to obtain convergence for this data-model combination, which is presumably caused by over-parametrization of the model. After much trial-and-error, a combination of starting values for the fixed effects was found for which convergence was obtained. The results are summarized in Table 4.2. The individual estimates, provided by the Lindstrom-Bates approach, have an empirical Bayes interpretation as they are shrunk towards the population mean. Comparing

Table 4.1 with 4.2, shows that the Lindstrom-Bates method yields comparable but more efficient estimates than the standard two-stage approach (smaller standard errors for all parameter estimates, apart from β_3). The *nlme* library also provides graphical tools for assessment of the fit. The diagnostic plots of the standardized residuals shown in Figure D.1 suggest that the model provides a good fit to the data.

The problem of convergence encountered in fitting model (4.15) with the *nlme* package motivates the search for a more parsimonious model, i.e. a models with fewer random effects than (4.15). Nested nonlinear mixed effects models can be compared with the likelihood ratio statistic ($2 \times$ difference in log-likelihoods), which is approximately chi-square distributed, with as degrees freedom the difference in the number of parameters between both models. As is shown in Section D.6, the *nlme* package allows nested models to be compared with the function *anova.* Table D.2 contains the results of a backward elimination of the random components of (4.15). The resulting final model contains only random effects for the parameters β_0 and β_1:

$$\log y_{ij} = \log\left[(\beta_0 + b_{i0}) + \frac{\beta_1 + b_{i1}}{1 + \exp\left(\beta_2(\beta_3 - \log x_{ij})\right)}\right] + e_{ij} \qquad (4.16)$$

This makes also sense from a physiological point of view: β_0 and β_1 determine the range of TXB_2 levels specific for a given individual, while the parameters β_2 and β_3 are specific for the drug. Unbiased population parameter estimates (Table 4.4) of the constrained model are obtained by using restricted maximum likelihood as objective function (Section D.7). Graphical assessment (Fig. D.2) of the fit of model (4.16) shows that, taking β_2 and β_3 in (4.15) as fixed, still provides an adequate fit to the data.

Table 4.3 Population means (SE) for parameters of model (4.16), comparison between different implementations (nlme and NLINMIX) and linearization methods.

Parameter	*nlme*	*NLINMIX*	*NLINMIX*
	Lindstrom-Bates	Lindstrom-Bates	first-order linearization
β_0	0.650 (0.123)	0.650 (0.123)	0.662 (0.129)
β_1	98.0 (28.9)	98.0 (28.9)	78.7 (25.7)
β_2	-1.126 (0.059)	-1.126 (0.059)	-1.139 (0.063)
β_3	-17.827 (0.157)	-17.827 (0.157)	-17.827 (0.159)

Alternative fitting strategies are provided by the SAS-macro *NLINMIX* (Section D.8). Attempts to fit model (4.16) with the Lindstrom-Bates algorithm yielded convergence problems, that could be solved by skipping the initial nonlinear fit (setting *options=skipnlin* in the macro specifications). The algorithm also provides for a first-order linearization algorithm which is close to the Vonesh-Carter approach. Table 4.3 shows that the two implementations of the Lindstrom-Bates algorithm yielded the same results. Estimates of β_0, β_2, and β_3, ob-

tained with the first-order linearization method are also comparable with the Lindstrom-Bates method, but for β_1, the two approaches differ markedly.

According to the specifications given by the author (Vonesh, 1995), the SAS-macro *MIXNLIN* allows the fitting of nonlinear mixed models of a very general type. However, error messages are cryptic and, even for the simple case as is discussed here, it is very difficult to supply the correct code to the macro. Moreover, for the code in Section D.9, although seemingly valid output was obtained, the SAS log showed an error message, which makes the results questionable.

The apparent discrepancy between the two linearization methods, with regard to estimating the parameter β_1, has the same explanation as was given earlier when the pooled data and standard two-stage method were compared. The first-order method is based on a preliminary estimater $\hat{\beta}^{(p)}$ using ordinary least squares on the pooled data, while the Lindstrom-Bates approach is based on individual estimates. The problem of having substantially different estimates for β_1 with different statistical methods has largely to do with the parametrization of (4.15) and (4.16). A better reparametrization of (4.15) that constrains the lower and upper vertical asymptotes to positive values is provided by:

$$\log y_{ij} = \log\left[\exp\left(\beta_0 + b_{i0}\right) + \frac{\exp\left(\beta_1 + b_{i1}\right)}{1 + \exp\left((\beta_2 + b_{i2})(\beta_3 + b_{i3} - \log x_{ij})\right)} \right] + e_{ij} \qquad (4.17)$$

Results for fitting this model with the pooled data, standard two-stage, and Lindstrom-Bates approach are summarized in Table 4.4. Now, all three fitting strategies yield virtually the same parameter estimates.

Table 4.4 Population means (SE) for parameters of model (4.17), comparison of pooled data, standard two-stage, and Lindstrom-Bates approach.

Parameter	Pooled data	Standard two-stage	Lindstrom-Bates
β_0	-0.412 (0.303)	-0.558 (0.252)	-0.463 (0.220)
β_1	4.365 (0.186)	4.365 (0.329)	4.362 (0.294)
β_2	-1.139 (0.191)	-1.176 (0.107)	-1.162 (0.073)
β_3	-17.827 (0.481)	-17.830 (0.156)	-17.814 (0.168)

Table 4.5 Population means (SE) for parameters of model (4.17) with β_2 and β_3 fixed, comparison of Lindstrom-Bates and first-order linearization approach.

Parameter	Lindstrom-Bates	First-order linearization
β_0	-0.450 (0.196)	-0.412 (0.196)
β_1	4.365 (0.327)	4.365 (0.327)
β_2	-1.136 (0.062)	-1.139 (0.063)
β_3	-17.834 (0.160)	-17.827 (0.160)

Backward elimination of random effects from (4.17) yielded the same results as obtained earlier, with random effects present only for β_0 and β_1. Fitting this final model with the Lindstrom-Bates method and restricted maximum likelihood as objective function (Section D. 10), provided a good fit to the data (Fig. D.3). Restricted maximum likelihood parameter estimates obtained with the Lindstrom-Bates and first-order linearization approach (Section D.11) are summarized in Table 4.5. Comparison of the results shows that both linearization methods now yield consistent estimates.

Finally, the potency of the drug is reported as 1.80×10^{-8} M. To set up a 95 % confidence interval, one could make use of the large sample approximation to obtain an approximate interval of :

$$\exp(-17.834 \pm 0.160 \times 1.96) = \left[1.31 \times 10^{-8},\ 2.46 \times 10^{-8}\right]$$

The use of the large sample approximation for setting up confidence intervals may be questionable when $m = 6$. Furthermore, the approximation depends highly on the accuracy of the linearization. Pinheiro (personal communication), suggests to use the above normal approximation only for m being large, and for small m to use a series of likelihood ratio tests to obtain profile likelihood intervals. This still is a large sample approximation, but can be motivated by $N = \sum_{i=1}^{m} n_i$ being large. Conducting a series of likelihood ratio tests (Section D.12) yielded as 95 % confidence interval:

$$\exp\left(\left[-18.2285,\ -17.5187\right]\right) = \left[1.21 \times 10^{-8},\ 2.46 \times 10^{-8}\right]$$

It is evident from this application that little is gained by making use of rather highly sophisticated statistical techniques as linearization methods. However, it should be stressed that these data represent almost the ideal case of unbalanced data without autocorrelation. When some of the blood samples are lost, due to contamination or other random mechanisms, inference made with the standard two-stage approach will become invalid and the Lindstrom-Bates approach, or alternatively when data are very sparse the Vonesh-Carter method, will be the most appropriate way of analysis.

4.4.2 Gastric emptying of solid test meals in dogs

Dogs are surgically implanted with a duodenal cannula which allows gastroduodenal efflux to be measured directly. After a recovery period of at least two weeks, the cannula is opened, duodenal juice and food remnants and removed and the duodenum is cleansed with water. After a stabilization period, during which certain experimental conditions are set, a test meal is fed. Thereafter, effluent from the duodenal cannula is collected for 5 hours in 15 min. por-

tions that were dried and weighed. The experiment is repeated, with a one week interval, under different experimental conditions.

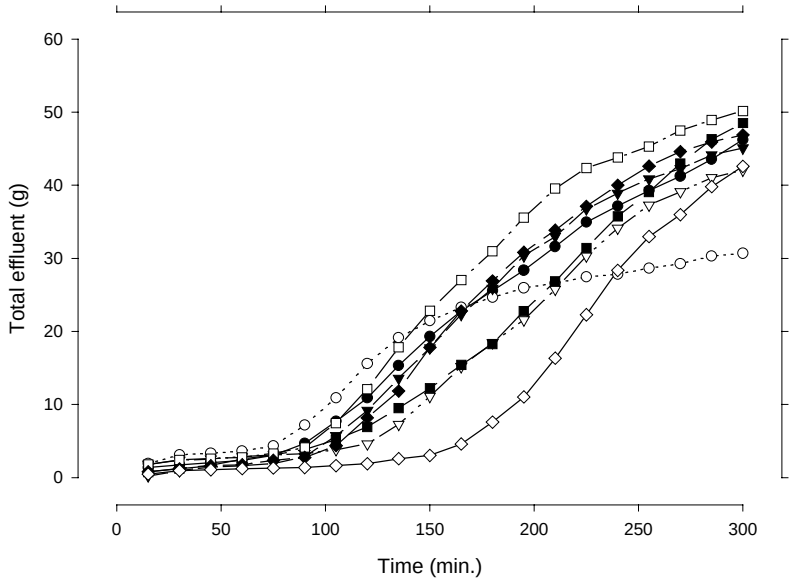

Figure 4.2 Individual gastric emptying profiles after administration of solvent

The response variable is defined as the cumulative dry weight of the effluent portions at each 15 min. period. The individual data of a typical experiment are listed in Appendix E, Table E.1. The individual gastric emptying profiles are depicted in Figure 4.2. It is the purpose of the investigator to estimate individual as well as population parameters of the emptying pattern for a given experimental condition.

For the modeling of gastric emptying data Elashoff, *et al.* (1982) proposed a modification of a Weibull-type function (Ratkowsky, 1990), which they referred to as the *power exponential*. This model describes the fraction of a meal remaining stomach as:

$$y_t = 2^{-\left(t/t_{1/2}\right)^{\beta}} \tag{4.18}$$

where t is the elapsed time after the start of the experiment, $t_{½}$ is the time at which 50 % of the meal has emptied and β is a shape parameter. The power exponential is a very flexible model, which has the simple exponential as a special case ($\beta = 1$), but allows also to fit emptying patterns in which slow emptying is followed by a rapid phase ($\beta > 1$). A value of $\beta < 1$ describes a pattern with very rapid initial emptying, followed by a slower emptying phase. For the present data (4.18) has to be slightly modified. Indeed, instead of the fraction remaining in the stomach, the amount emptied is measured and the total amount emptied at $t = \infty$ is actually unknown. Furthermore, the initial measurements at 15 min. show some baseline emptying, which presumably consists of duodenal juices, that is not accounted for by (4.18). Therefore, the following adjusted model is proposed:

$$y_i = \beta_0 + \beta_1\left(1 - 2^{-\left(x_i/\beta_2\right)^{\beta_3}}\right) \tag{4.19}$$

where β_0 is the baseline emptying, β_1 the total amount emptied at t = ∞ minus the baseline emptying, β_2 is the time at which 50 % of the meal has emptied and β_3 is the shape parameter.

Before fitting the estimation function, it is essential to consider whether the current parametrization is optimal with regard to both its physical interpretation and statistical properties. As it is parametrized in (4.19), the model allows the response and the time to 50 % emptying to attain negative values, which is physically impossible. A reparametization of (4.19) that constrains the response variable as well as the time to 50 % emptying to positive values, is given by:

$$y_i = \exp(\beta_0) + \exp(\beta_1)\left(1 - 2^{-\left(\frac{x_i}{\exp(\beta_2)}\right)^{\beta_3}}\right) \qquad (4.20)$$

To investigate the estimation properties of a model, Ratkowsky (1990) suggests the use of Monte Carlo simulations. Section E.2 shows the implementation of this idea. Using the parameter estimates of the first dog, as if it were the true values, 2000 data sets of each 20 observations were generated by allowing the error term to change randomly according to a normal distribution. The least squares estimates were obtained and their distribution investigated by constructing normal quantile plots. Figure E.1 shows that the parametrization of (4.19) has excellent estimation properties for the parameters β_0 and β_3, while there is a minor deviation from normality for β_1 and β_2. As is shown in Figure E.2 fitting the parametrization of (4.20) yields a substantial deviation from normality for β_0, while the asymptotic distribution of the remaining parameters is practically unchanged. Since β_0 is in fact a nuisance factor and not of direct interest to the investigator, parametrization (4.20) is chosen as estimation function. Putting (4.20) into the framework of nonlinear mixed effects yields as saturated mixed effects model:

$$y_{ij} = \exp(\beta_0 + b_{0i}) + \exp(\beta_1 + b_{1i})\left(1 - 2^{-\left(\frac{x_{ij}}{\exp(\beta_2 + b_{2i})}\right)^{\beta_3 + b_{3i}}}\right) + e_{ij} \qquad (4.21)$$

The process that generated the data will almost surely introduce a substantial amount of serial correlation. Furthermore, the possibility that response variation changes systematically with the level of the response should also be considered.

A first guess of the initial values is obtained from Figure 4.2, while more refined estimates are obtained from fitting the model to the pooled data (Section E.3). Conventionally the data were analyzed using a standard two-stage approach (Section E.4) without allowance for serial correlation.

Section E.5 shows the results of fitting model (4.21) with the *nlme* library, assuming independence between successive measurements and homogeneneity of variance. In the graphical assessment of the fit (Fig. E.3), the plot of the standardized residuals versus the fitted values shows a cyclic pattern, suggesting the existence of serial correlation. The plot of the logarithm of the absolute value of the intra-individual residuals versus the logarithm of the predicted values shows that there is no evidence for the response variation to be related to the level of the response. Plotting for each dog separately the individual data together with the fitted curve (Fig. E.4) shows that, although in general the fit is very good, the onset of the upward rise is over-estimated in six dogs.

The cumulative nature of the response, makes it plausible that a serial correlation is present. Serial correlation for different lags can be assessed by constructing the correlogram or variogram based on the differences with the sample mean at each time point. The variogram (Diggle, 1990) in Figure E.4, which is based on log-transformed responses, gives evidence for a serial correlation and suggests a Gaussian correlation model of the form:

$$\rho(u) = \exp(-\alpha u^2)$$

According to the documentation, the *nlme* library provides a very flexible way for modeling the intra-individual correlation matrix $\Gamma_i(\alpha)$. Apart from a set of pre-defined correlation structures (identity, compound symmetry, AR(1), AR(2), MA(1), MA(2), ARMA (1,1)), the user can define his own set of functions to handle less common correlation structures. However, the documentation is rather obscure in how to actually implement these functions. Furthermore, the fact that the experiment consists of only eight dogs makes the fitting of complex structures as the unstructured correlation type impossible. Moreover, even for some relatively simple covariance models convergence problems were encountered. Therefore, it was decided to compare the models that could be fitted without extra programming effort using Akaike's information criterion (AIC). Four covariance structures were compared: independence (AIC = 533), compound symmetry (AIC = 535), MA(1) (AIC = 450), AR(1) (AIC = 415), Gaussian (AIC = 432) ,and ARMA(1,1) (AIC =386). Although, the ARMA(1,1) model performed best, it must be stressed that there is no reason to assume that this is the correct model for the intra-individual correlation. However, it is used as a working model, simply because it is considered to be closer to reality than assuming independence among the observations.

Next, redundant random effects were eliminated from model (4.21). The backward elimination procedure, summarized in Table E.2 (Section E.7), encountered the same convergence problems as mentioned earlier. The first step of the procedure yielded only a solution for removal of random effects from β_0. The resulting p-value of 0.994 indicates that for β_0 the random effects are redundant. Subsequent removal of the remaining random effects, with β_0 kept fixed, yielded highly significant contributions for all random components.

The final model is refitted using restricted maximum likelihood as objective function (Section E.8). The Wald confidence intervals for this model suggest to remove also the removal of the fixed parameter β_0. However, a likelihood ratio test, using models fitted with approximate likelihood as objective function, yielded a p-value of 0.027 for this parameter. Graphical assessment (Fig. E.6) shows that much of the cyclic pattern has disappeared from the plot of the standardized residuals versus the fitted values. The plot of the logarithm of the absolute values of the residuals versus the logarithm of the predictive values shows some pattern, which could be modeled by a second degree polynomial. However, the dependency of the variance on the predicted value was not important enough to consider extra modeling.

Figure E.7 depicts for each dog the individual data and the plotted curve according to the *nlme* estimates of the individual parameters. Apart from the second dog, that clearly is an outlier, the fitted curves provide a good description of the experimental data.

Table 4.2 summarizes the results of the standard two-stage and Lindstrom-Bates approach. It is evident from the table that the two strategies yield very similar results with regard to both the individual and population parameters.

Table 4.2 Individual parameter estimates and population means for model (4.15). Comparison between standard two-stage and Lindstrom-Bates approach

Dog	Standard two-stage				Lindstrom-Bates			
	β_0	β_1	β_2	β_3	β_0	β_1	β_2	β_3
1	-0.538	3.861	5.174	2.555	-0.101	3.886	5.192	2.590
2	0.526	3.310	4.822	2.804	-0.101	3.350	4.801	3.019
3	-2.088	3.803	5.129	3.143	-0.101	3.795	5.139	3.144
4	0.742	3.710	5.282	3.981	-0.101	3.748	5.272	3.865
5	0.341	4.022	5.385	3.160	-0.101	3.970	5.350	3.357
6	0.224	3.868	5.084	3.189	-0.101	3.896	5.098	2.969
7	-0.803	3.833	5.151	3.350	-0.101	3.836	5.159	3.244
8	0.012	3.726	5.417	6.034	-0.101	3.788	5.425	5.133
Population mean	-0.198	3.767	5.181	3.527	-0.101	3.784	5.180	3.415
SE	0.327	0.074	0.067	0.387	0.450	0.071	0.069	0.313

4.4.3 Extracellular acetylcholine levels in the striatum of rats.

Rats were anesthetized and a probe holder was implanted in the striatum of the brain. One week after surgery a microdialysis probe was inserted into the probe holder. A microdyalysis probe allows certain molecules to be exchanged across a membrane. The probe was perfused at a constant flow rate and each hour the concentration of the neurotransmitter acetylcholine was determined in fractions of the perfusate. After collection of the first perfusate, an experimental acetylcholine inhibitor was injected intraperitoneally.

Two dosages (2.5 and 5 mg/kg) of the experimental drug were randomly allocated to sixteen rats. The purpose of the experiment was to determine whether the response curve of the acetylcholine level was different between the two treatment groups.

The individual data are listed in Appendix F, Table F.1. In both treatment groups there are missing values at the two last observation times. It was argued by the investigators that these missing data were due to contamination of the samples and occurred completely at random. As is shown in Figure 4.3, there is substantial variation present in the data, making it rather a challenge to model them.

It was decided to model the data using an exponentially damped polynomial model of the form:

$$\beta_0 + \beta_1 x^3 e^{-\beta_2 x}$$

This estimation function is presented by Hand and Crowder (1996) for the fitting of blood glucose levels after alcohol intake. The parameter β_0 represents the baseline level of acetylcholine, β_1 is related to the initial increase and β_2 to the exponential decay. The model is modified to allow for the two treatment groups, the large range of response levels, and the presence of random effects:

$$\log y_{ij} = \log\left[\exp(\beta_0 + b_{0i}) + \exp(\beta_1 + b_{1i} + \gamma_1 g_i) x_{ij}^3 \exp\left(-(\beta_1 + b_{1i} + \gamma_1 g_i) x_{ij}\right)\right] + e_{ij} \tag{4.22}$$

where y_{ij} is the concentration of acetylcholine in nmol/l, x_{ij} is the time after administration, and g_i an indicator (0, 1) for the type of treatment. A test on the difference in response curves between the two treatment groups is carried out by comparing (4.22) with the reduced model:

$$\log y_{ij} = \log\left[\exp(\beta_0 + b_{0i}) + \exp(\beta_1 + b_{1i}) x_{ij}^3 \exp\left(-(\beta_1 + b_{1i}) x_{ij}\right)\right] + e_{ij} \tag{4.23}$$

Analysis is started by computing the partial derivatives of (4.23) and carrying out a fit on the pooled data to obtain more refined initial values (Section F.2). The presence of missing values makes the standard two-stage approach questionable. Individual parameter estimates, their mean value and standard error are shown in Section F.3.

Model 4.23 is fitted using *nlme* in Section F.4. Graphical assessment (Fig. F.1) of the fit shows that, apart from the second experiment where some rather large residuals are found at the lowest levels of the response, the model provides a good fit to the data.

A first impression of the effect of the different treatments is obtained by constructing plots of the random effects by treatment group (Section F.5). From Figure F.2 it is evident that the 5 mg/kg group has higher values for the parameter β_1. Formal testing is carried out in Section F.6. First, model (4.22), with allowance for treatment effect on β_1 as well as on β_2, is fitted.

Testing the joint effect of treatment on these parameters is carried out by a likelihood ratio test (Section F.7) with the former fit, i.e. model (4.23). The resulting p-value of 0.005 is a strong indication that the response curves differ from one another.

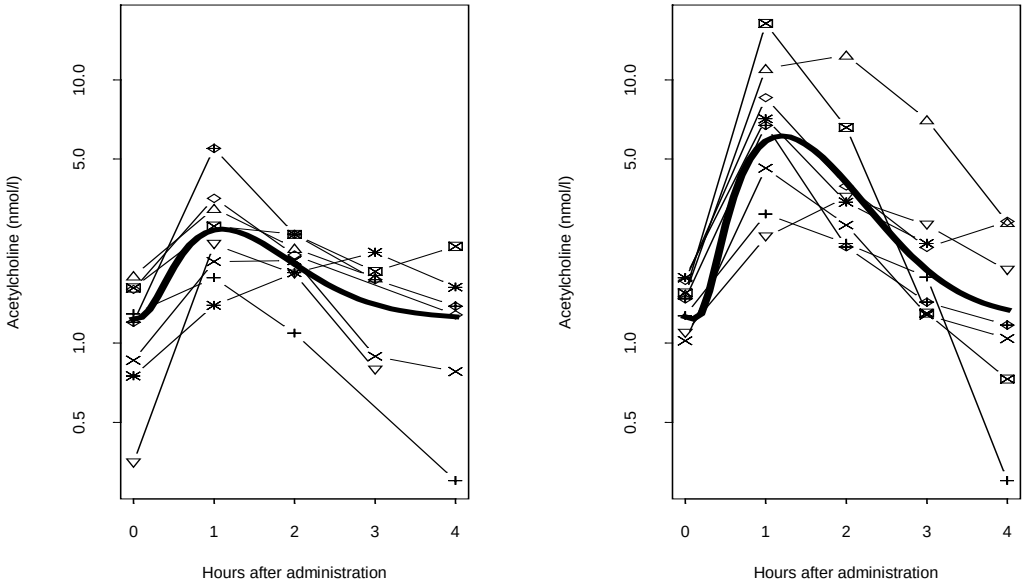

Figure 4.3 Individual levels of acetylcholine and fitted curves for the two treatment groups: left panel 2.5 mg/kg, right panel 5 mg/kg.

Taking model (4.22), the next question is whether random effects are still needed. Likelihood ratio tests of reduced models with model (4.22) yield a p-value of 0.402 and 0.398 for the presence of random effects in β_1 and β_2 respectively (Section F.8). Testing for the importance of random effects in β_2, when β_1 is considered fixed, yields a p-value of 0.074. Although, the latter result is somewhat borderline, it was decided to remove random effects from both parameters.

Next, the test on treatment difference is reduced to likelihood ratio tests on the significance of the parameters β_1 and β_2 (Section F.9). For β_1 the result is borderline significant with a p-value of 0.049, while for β_2 no difference between the treatment groups is detected ($p = 0.787$). Section F.10 shows a fit to the model with random effects only present in the baseline (β_0) and a treatment effect only for the initial rise (β_0). The graphical assessment of the model (Fig. F.3) shows that this rather simple model provides a surprisingly good description of the data. The fitted population model is also depicted in Figure 4.5 as a continuous curve.

4.5 Summary

Repeated measures in nonlinear models can be handled using individual fits or linearization methods. In some instances, i.e. no missing data present and small intra-individual variation, individual fits can provide a simple and valid solution. However, when there are missing data, linearization methods are more appropriate. The *nlme* implementation of the Lind-

strom-Bates algorithm is the most appealing. However, convergence problems occur frequently when using this method. An alternative method, that is implemented in SAS, is the Vonesh-Carter first-order linearization.

5.
Concluding remarks

The main purpose of this project was to study some new developments in nonlinear regression techniques and to look at their applicability in experimental pharmacology. This covered items that are somewhat scattered in the literature, i.e. importance of parametrization, use of profile likelihood intervals, heterogeneity of variance, and repeated measurements. As is shown in many applications these techniques can be quite powerful tools for answering specific research questions. However, several items were necessarily not covered. For instance, robust estimation is discussed in some part by Seber and Wild (1989), and Carroll and Ruppert (1988). These techniques are especially promising in laboratory systems, where parameter estimation is carried out by laboratory personnel with only a limited statistical background. Experimental design issues, such as the construction of D-optimal designs, are also of particular importance to investigators. Another interesting issue, worth further study, is the sequential determination of parameters such as IC_{50}-values.

Use of resampling techniques like bootstrapping is often advocated and has not been discussed in this project. My experience with these techniques in nonlinear regression, is that they are often confronted with convergence problems, especially in those cases where their application is most interesting. In the repeated measures context, it is not entirely clear how these techniques can be applied. Computability can also be an issue in this area.

Convergence is also the main problem in repeated measures when the problem is less well-conditioned. It can take hours to find the right parametrization - initial value combination, and having found a working combination for some saturated model, does not automatically mean that it will work for nested models as well.

When comparing different software implementations, S-Plus turned out to be the most flexible. However, the documentation is sometimes very poor, even when supplemented with textbooks like Venables and Ripley (1994), and Chambers and Hastie (1993) . For instance, no real description of the *profile* function could be found. The same remark applies even more to added software as the *nlme* library.

6.
References

Ashton, D., Fransen, J.F., Clincke, G., Lewi, P., Woestenborghs, R., Wouters, L. (1992). Acute pharmacodynamic and pharmacokinetic interactions between loreclezole and reference anti-epileptic drugs in the rat. *Unpublished report*. Janssen Research Foundation, Beerse.

Bates, D.M., Watts, D.G.. (1988). *Nonlinear Regression Analysis and its Applications*. Wiley, New York.

Bates, D.M., Chambers, J.M. (1993). Nonlinear Models. Chapter 10 of Chambers, J.M., Hastie, T.J. (eds) *Statistical Models in S*. Chapman & Hall, New York.

Beal, S.L., Sheiner, L.B. (1982). Estimating population kinetics. *CRC Critical Reviews in Biomedical Engineering 8,* 195-222.

Beetens, J., Beerens, D., Van de Wiele, R., Van Ammel, K., De Clerck, F. (1996). Effects of ridogrel (R068070) on the formation of eicosanoids in human whole blood. *Preclinical research report*. Janssen Research Foundation, Beerse.

Carroll, R.J., Ruppert, D. (1988). *Transformation and Weighting in Regression.* Chapman & Hall, New York.

Cucchiara, A.J., Deal, R.B., Kenny, S.J., Costiloe, J.P. (1989). Nonlinear regression analysis of a biexponential model using various methods of the NLIN procedure of SAS$^{(R)}$ software. *SAS*® *Users Group International. Proceedings of the Fourteenth Annual Conference.* 1324-1329. SAS Institute Inc., Cary, North Carolina.

Cook, R.D., Weisberg, S. (1982). *Residuals and influence in Regression.* Chapman & Hall, New York.

Davidian, M., Giltinan, D.M. (1993). Analysis of repeated measurement data using the nonlinear mixed effects model. *Chemometrics and Intelligent Laboratory Systems 20,* 1-24.

Davidian, M., Giltinan, D.M. (1995). *Nonlinear Models for Repeated Measurement Data*. Chapman & Hall, London.

Davidian, M., Haaland, P.D. (1990). Regression and calibration with nonconstant error variance. *Chemometrics and Intelligent Laboratory Systems 9,* 231-248.

Diggle, P. (1990). *Time Series: a biostatistical introduction.* Oxford University Press, Oxford.

Draper, N.R., Smith, H. (1981). *Applied Regression Analysis* (2nd edn). Wiley, New York.

Elashoff, J.D., Reedy, J., Meyer, J.H. (1982). Analysis of gastric emptying data. *Gastroenterology 83,* 1306-1312.

Emmens, C.W. (1940). The dose/response relation for certain principles of the pituitary gland, and of the serum and urine of pregnancy. *Journal of Endocrinology 2,* 194-225.

Finney, D.J. (1964). *Statistical Method in Biological Assay* (2nd edn). Griffin, London.

Fleming, W.W., Westfall, D.P., De La Lande, I.S., Jellett, L.B. (1972). Log-normal distribution of equieffective doses of norepinephrine and acetylcholine in several tissues. *The Journal of Pharmacology and Experimental Therapeutics 181,* 339-345.

Giltinan, D.M., Ruppert, D. (1989). Fitting heteroscedastic regression models to individual pharmacokinetic data using standard statistical software. *Journal of Phamacokinetics and Biopharmaceutics 17,* 601-634.

Glantz, S.A., Slinker, B.K. (1990). *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, New York.

Hand, D., Crowder, M. (1996). *Practical Longitudinal Data Analysis.* Chapman & Hall, London.

Hartley, H.O. (1961). The modified Gauss-Newton method for the fitting of non-linear regression functions by least squares. *Technometrics, 3,* 269-280.

Lindstrom, M.J., Bates, D.M. (1990). Nonlinear mixed effects models for repeated measures data. *Biometrics 46,* 673-687.

Littell, R.C., Milliken, G.A., Stroup, W.W., Wolfinger, R.D. (1996). *SAS® System for Mixed Models.* SAS Institute Inc., Cary, North Carolina.

McCullagh, P., Nelder, J.A. (1989). *Generalized Linear Models* (2nd edn). Chapman & Hall, London.

Neter, J., Wasserman, W., Kutner, M.H. *Applied Linear Statistical Models* (3rd edn). Irwin, Homewood.

O'Brien, T.E., Wang, Y. (1996). Using SAS software to assess and adjust for nonlinearity in nonlinear regression models. *SAS® Users Group International. Proceedings of the Twenty-First Annual Conference.* 1274-1283. SAS Institute Inc., Cary, North Carolina.

Organisation for Economic Co-operation and Development (1984). Guideline No. 201 Alga, Growth Inhibition Test. *OECD guidelines for testing chemicals*. OECD, Paris.

Pinheiro, J.C., Bates, D.M. (1995). *lme and nlme. Mixed Effects Methods and Classes for S and Splus*. Version 1.2. University of Wisconsin, Madison.

Price, W.J., Shafii, B. (1992). Graphical aids for summarizing inferential results of nonlinear regression. *SAS® Users Group International. Proceedings of the Seventeenth Annual Conference.* 1329-1334. SAS Institute Inc., Cary, North Carolina.

Ratkowsky, D.A. (1990). *Handbook of Nonlinear Regression Models.* Dekker, New York.

SAS Institute (1989). *SAS/STAT® User's Guide, Version 6,* (4th edn), Volume 2. SAS Institute, Cary, North Carolina.

Schotte, A., Janssen, P.F.M., Megens, A.A.H.P., Leysen, J.E. (1993). Occupancy of central neurotransmitter receptors by risperidone, clozapine and haloperidol, measured ex vivo by quantitative autoradiography. *Brain Research 631,* 191-202.

Seber, G.A.F., Wild, C.J. (1989). *Nonlinear Regression*. Wiley, New York..

Tallarida, R.J., Jacob, L.S.(1979). *The Dose-Response Relation in Pharmacology*. Springer, New York.

Venables, W.N., Ripley, B.D.(1994). *Modern Applied Statistics with S-Plus*. Springer, New York.

Vølund, A. (1978). Application of the four-parameter logistic model to bioassay: comparison with slope ratio and parallel line models. *Biometrics 34*, 357-365.

Vonesh, E.F. (1995). MIXNLIN: A SAS® procedure for nonlinear mixed-effects models. Version 3.1. Technical report number TR92M-0300B. Applied Statistics Center, Baxter Healthcare Corporation, Round Lake, Illinois.

Vonesh, E.F., Carter, R.L. (1992). Mixed-effects nonlinear regression for unbalanced repeated measures. *Biometrics 48*, 1-17.

Vonesh, E., Chinchilli, V. (1996). *Linear and Nonlinear Models for the Analysis of Repeated Measurements.* Dekker, New York.

Appendix A
Receptor-occupancy study

A.1 Data (source: A. Schotte, unpublished results)

Table A.1 Percentage receptor-occupancy as determined by autoradiography

Dose mg/kg					
0.01	0.04	0.16	0.63	2.5	10
13	22	39	59	85	97
1	29	39	53	84	92
10	19	14	49	83	77
8	6	21	38	65	89
1	19	9	34	53	86
6	9	19	47	51	.
2	20	12	34	54	.

- Parameter coding of model (1.6): p = p (slope parameter), $\xi = \log x_{50}$ = lx50

A.2 Starting Values

Plotting Log(y/(1-y) versus log dose shows the linear relation as in 1.4 (Fig. A.1). Linear regression can be applied to (1.4) to find starting values of the 2-parameter logistic equation (1.6).

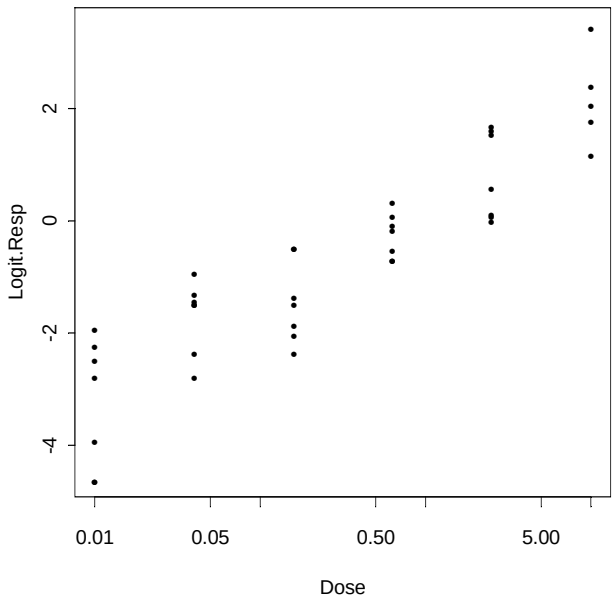

Figure A.1 Plot of log(y/(100-y)) versus log dose, the linear relation is used to obtain starting values

```
> Logit.Resp<-log(Respons/(100-Respons))
> plot(Dose,Logit.Resp,log="x")
> fit.lin<-lm(Logit.Resp~LDose)
> fit.lin$coefficient
 (Intercept)     LDose
   0.2606395 0.7304701
> -fit.lin$coefficient[1]/fit.lin$coefficient[2]
 (Intercept)
  -0.3568106
```

A.3 S-Plus implementation

- Gauss-Newton using analytical derivatives:

    1. Building a function containing the model and the first derivative

```
> der<-deriv(Respons~100/(1+exp(p*(lx50-log(x)))),c("lx50","p"),function(lx50,p,x)
+       NULL)
> der
function(lx50, p, x)
{
        .expr2 <- lx50 - (log(x))
        .expr4 <- exp((p * .expr2))
        .expr5 <- 1 + .expr4
        .expr9 <- .expr5^2
        .value <- 100/.expr5
        .grad <- array(0, c(length(.value), 2), list(NULL, c("lx50", "p")))
        .grad[, "lx50"] <-  - ((100 * (.expr4 * p))/.expr9)
        .grad[, "p"] <-  - ((100 * (.expr4 * .expr2))/.expr9)
        attr(.value, "gradient") <- .grad
        .value
}
```

    2. Fitting the model:

```
> fit<-nls(Respons~der(lx50,p,Dose),data=Receptor.data,start=
+          list(lx50=-0.36,p=0.73),trace=T)
4308.2 : -0.36 0.73
4119.25 : -0.219224 0.663316
4118.67 : -0.221245 0.667886
4118.67 : -0.220632 0.668104
> summary(fit)

Formula: Respons ~ der2(lx50, p, Dose)

Parameters:
         Value Std. Error   t value
lx50 -0.220632  0.1437070  -1.53529
   p  0.668104  0.0642933  10.39150

Residual standard error: 10.4109 on 38 degrees of freedom
Correlation of Parameter Estimates:
    lx50
p -0.121
```

- Gauss-Newton using numerical derivatives:

```
>fit2<-nls(Respons~100/(1+exp(p*(lx50-log(Dose)))),data=Receptor.data,start=
+         list(lx50=-0.36,p=0.73),trace=T)
4308.2 : -0.36 0.73
4119.25 : -0.219224 0.663316
4118.67 : -0.221245 0.667886
4118.67 : -0.220632 0.668104
> summary(fit2)

Formula: Respons ~ 100/(1 + exp(p * (lx50 - log(Dose))))

Parameters:
         Value Std. Error   t value
lx50 -0.220632  0.1437070  -1.53529
   p  0.668104  0.0642933  10.39150

Residual standard error: 10.4109 on 38 degrees of freedom

Correlation of Parameter Estimates:
    lx50
p -0.121
```

- Generating and supplying second derivatives with the *MASS* library:

```
> library(mass,first=T)
> der2<-deriv3(~100/(1+exp(p*(lx50-log(x)))),c("lx50","p"), function(lx50,p,x) NULL)
> der2
function(lx50, p, x)
{
  .expr2 <- lx50 - (log(x))
  .expr4 <- exp((p * .expr2))
  .expr5 <- 1 + .expr4
  .expr7 <- .expr4 * p
  .expr8 <- 100 * .expr7
  .expr9 <- .expr5^2
  .expr18 <- .expr9^2
  .expr22 <- .expr4 * .expr2
  .expr28 <- 2 * (.expr22 * .expr5)
  .expr32 <-  -(((100*((.expr22*p) + .expr4))/.expr9) - ((.expr8*.expr28)/.expr18))
  .expr33 <- 100 * .expr22
  .value <- 100/.expr5
  .grad <- array(0, c(length(.value), 2), list(NULL, c("lx50", "p")))
  .hess <- array(0, c(length(.value), 2, 2), list(NULL, c("lx50", "p"), c("lx50","p")))
  .grad[, "lx50"] <-  - (.expr8/.expr9)
  .grad[, "p"] <-  - (.expr33/.expr9)
  .hess[, "lx50", "lx50"] <- -(((100*(.expr7*p))/.expr9)-((.expr8*(2*(.expr7*.expr5)))
                               /.expr18))
  .hess[, "p", "lx50"] <- .expr32
  .hess[, "lx50", "p"] <- .expr32
  .hess[, "p", "p"] <- -(((100*(.expr22*.expr2))/.expr9)-((.expr33 * .expr28)/.expr18))
  attr(.value, "gradient") <- .grad
  attr(.value, "hessian") <- .hess
  .value
}
```

```
> fit3<-nls(Respons~der2(lx50,p,Dose),data=Receptor.data,start= list(lx50=-0.35,
+       p=0.73), trace=T)
4293.33 : -0.35 0.73
4119.14 : -0.218807 0.663819
4118.67 : -0.221186 0.66793
4118.67 : -0.220628 0.668108
> summary(fit3)

Formula: Respons ~ der2(lx50, p, Dose)

Parameters:
         Value Std. Error   t value
lx50 -0.220628  0.1437070  -1.53526
   p  0.668108  0.0642937  10.39150

Residual standard error: 10.4109 on 38 degrees of freedom

Correlation of Parameter Estimates:
    lx50
p -0.121
```

A.4 SAS PROC NLIN implementation

- Setting up the partial derivatives:

With $u = \log(x)$ and $\xi = \log x_{50}$ the partial derivatives of (1.6) with respect to the parameters p and ξ are:

$$\frac{\partial y}{\partial p} = -\frac{(\xi - u)\exp\big(p(\xi - u)\big)}{\big[1+\exp\big(p(\xi - u)\big)\big]^2} \text{ and } \frac{\partial y}{\partial \xi} = -\frac{p\exp\big(p(\xi - u)\big)}{\big[1+\exp\big(p(\xi - u)\big)\big]^2}$$

For the Newton-Raphson procedure in SAS, the second derivatives are also required:

$$\frac{\partial^2 y}{\partial p^2} = \frac{\exp\big(p(\xi - u)\big)\big[\exp\big(p(\xi - u)\big) - 1\big](\xi - u)^2}{\big[1+\exp\big(p(\xi - u)\big)\big]^3}$$

$$\frac{\partial^2 y}{\partial \xi^2} = \frac{\exp\big(p(\xi - u)\big)\big[\exp\big(p(\xi - u)\big) - 1\big]p^2}{\big[1+\exp\big(p(\xi - u)\big)\big]^3}$$

$$\frac{\partial^2 y}{\partial p \partial \xi} = \frac{\exp\big(p(\xi - u)\big)\big[(p\xi - pu - 1)\exp\big(p(\xi - u)\big) + pu - p\xi - 1\big]}{\big[1+\exp\big(p(\xi - u)\big)\big]^3}$$

- Gauss-Newton method

```
Title 'Gauss-Newton';
proc nlin data=receptor method=gauss;
 parms Lx50=-0.36 p=0.73;
 x=Lx50-log(dose);
 v=exp(p*x);
 z=(1+v)**2;
 model respons=100/(1+v);
 der.p=-100*x*v/z;
 der.lx50=-100*p*v/z;
run;
                                  Gauss-Newton

Non-Linear Least Squares Iterative Phase   Dependent Variable RESPONS   Method:
Gauss-Newton
                     Iter           LX50              P Sum of Squares
                        0      -0.360000       0.730000    4308.201154
                        1      -0.219224       0.663316    4119.252329
                        2      -0.221245       0.667886    4118.671803
                        3      -0.220632       0.668104    4118.667883
                        4      -0.220613       0.668134    4118.667853
NOTE: Convergence criterion met.

          Non-Linear Least Squares Summary Statistics     Dependent Variable RESPONS

               Source                DF Sum of Squares     Mean Square

               Regression             2   90581.332147    45290.666074
               Residual              38    4118.667853      108.385996
               Uncorrected Total     40   94700.000000

               (Corrected Total)     39   34792.400000

               Parameter     Estimate    Asymptotic            Asymptotic 95 %
                                         Std. Error         Confidence Interval
                                                            Lower         Upper
                   LX50  -.2206128566 0.14370339491 -.51152368260 0.07029796936
                      P  0.6681343248 0.06429652510 0.53797348037 0.79829516919


                              Asymptotic Correlation Matrix

                         Corr                  LX50                 P
                         ffffffffffffffffffffffffffffffffffffffffffff
                             LX50                 1      -0.121338794
                                P      -0.121338794                 1
```

- Steepest-Descent or Gradient method

```
Title 'Steepest Descent';
proc nlin data=receptor method=gradient;
 parms Lx50=-0.36 p=0.73;
 x=Lx50-log(dose);
 v=exp(p*x); z=(1+v)**2;
 model respons=100/(1+v);
 der.p=-100*x*v/z;
 der.lx50=-100*p*v/z;
run;
                                Steepest Descent
Non-Linear Least Squares Iterative Phase     Dependent Variable RESPONS
Method:
Gradient
                     Iter           LX50              P Sum of Squares
                        0      -0.360000       0.730000    4308.201154
                        1      -0.297539       0.621724    4213.139520
                        2      -0.278804       0.681350    4140.369074
                        3      -0.256209       0.659763    4127.593882
                        4      -0.236320       0.683867    4125.704018
                        5      -0.230020       0.653259    4125.003273
                        6      -0.225244       0.684142    4124.901263
                        7      -0.223993       0.652917    4124.730918
                        8      -0.222843       0.668500    4118.696736
                        9      -0.221199       0.667446    4118.682179
                       10      -0.221002       0.668403    4118.670292
                       11      -0.220860       0.667935    4118.669274
                       12      -0.220793       0.668170    4118.668053
                       13      -0.220694       0.668098    4118.667933
                       14      -0.220668       0.668154    4118.667877
                       15      -0.220649       0.668130    4118.667863
NOTE: Convergence criterion met.
          Non-Linear Least Squares Summary Statistics     Dependent Variable RESPONS
               Source                DF Sum of Squares     Mean Square
               Regression             2   90581.332137    45290.666069
               Residual              38    4118.667863      108.385996
               Uncorrected Total     40   94700.000000
               (Corrected Total)     39   34792.400000
               Parameter     Estimate    Asymptotic            Asymptotic 95 %
                                         Std. Error         Confidence Interval
                                                            Lower         Upper
                   LX50  -.2206489420 0.14370375332 -.51156049360 0.07026260951
                      P  0.6681298842 0.06429586494 0.53797037620 0.79828939215
                               Asymptotic Correlation Matrix
                          Corr                 LX50                P
                          ffffffffffffffffffffffffffffffffffffffffff
                              LX50                1      -0.121335794
                                 P      -0.121335794                1
```

- Levenberg-Marquardt method

```
Title 'Marquardt';
proc nlin data=receptor method=Marquardt;
 parms Lx50=-0.36 p=0.73;
 x=Lx50-log(dose);
 v=exp(p*x);
 z=(1+v)**2;
 model respons=100/(1+v);
 der.p=-100*x*v/z;
 der.lx50=-100*p*v/z;
run;
                                  Marquardt

Non-Linear Least Squares Iterative Phase     Dependent Variable RESPONS
Method:
Marquardt
                     Iter           LX50              P Sum of Squares
                        0      -0.360000       0.730000    4308.201154
                        1      -0.220734       0.664042    4119.089440
                        2      -0.221125       0.667884    4118.671045
                        3      -0.220634       0.668107    4118.667879
                        4      -0.220612       0.668134    4118.667853
NOTE: Convergence criterion met.

          Non-Linear Least Squares Summary Statistics     Dependent Variable RESPONS

               Source                DF Sum of Squares     Mean Square

               Regression             2   90581.332147    45290.666074
               Residual              38    4118.667853      108.385996
               Uncorrected Total     40   94700.000000

               (Corrected Total)     39   34792.400000


               Parameter     Estimate    Asymptotic             Asymptotic 95 %
                                         Std. Error          Confidence Interval
                                                             Lower         Upper
                   LX50  -.2206124119 0.14370337915 -.51152320596 0.07029838220
                      P  0.6681344644 0.06429654213 0.53797358555 0.79829534332

                              Asymptotic Correlation Matrix

                         Corr                  LX50                 P
                         ffffffffffffffffffffffffffffffffffffffffffff
                             LX50                 1      -0.121338798
                                P      -0.121338798                 1
```

- DUD method

```
Title "Doesn't Use Derivatives (DUD)";
proc nlin data=receptor method=DUD;
 parms Lx50=-0.36 p=0.73;
 x=Lx50-log(dose); v=exp(p*x);
 model respons=100/(1+v);
run;
                          Doesn't Use Derivatives (DUD)
        Non-Linear Least Squares DUD Initialization     Dependent Variable RESPONS
                      DUD            LX50               P Sum of Squares
                       -3      -0.360000       0.730000    4308.201154
                       -2      -0.396000       0.730000    4371.275680
                       -1      -0.360000       0.803000    4591.815817

   Non-Linear Least Squares Iterative Phase     Dependent Variable RESPONS
Method: DUD
                     Iter           LX50               P Sum of Squares
                        0      -0.360000       0.730000    4308.201154
                        1      -0.219965       0.661299    4119.842459
                        2      -0.212011       0.669630    4119.137174
                        3      -0.216850       0.667233    4118.759028
                        4      -0.219005       0.667638    4118.686710
                        5      -0.220630       0.668323    4118.668711
                        6      -0.220805       0.668071    4118.668184
                        7      -0.220648       0.668146    4118.667863
                        8      -0.220642       0.668145    4118.667860
NOTE: Convergence criterion met.
          Non-Linear Least Squares Summary Statistics     Dependent Variable RESPONS
               Source                DF Sum of Squares     Mean Square

               Regression             2   90581.332140    45290.666070
               Residual              38    4118.667860      108.385996
               Uncorrected Total     40   94700.000000

               (Corrected Total)     39   34792.400000

               Parameter     Estimate    Asymptotic             Asymptotic 95 %
                                         Std. Error          Confidence Interval
                                                             Lower         Upper
                   LX50  -.2206422504 0.14370358105 -.51155345318 0.07026895243
                      P  0.6681449983 0.06429313198 0.53799102291 0.79829897375

                                Asymptotic Correlation Matrix
                           Corr                  LX50                 P
                           ffffffffffffffffffffffffffffffffffffffffffff
                               LX50                 1      -0.121318388
                                  P      -0.121318388
```

- Newton-Raphson (using second derivatives).

```
Title 'Newton-Raphson';
proc nlin data=receptor method=Newton;
 parms Lx50=-0.36 p=0.73;
 u=log(dose);
 x=Lx50-u; v=exp(p*x);
 z=(1+v)**2;
 z3=(1+v)**3;
 model respons=100/(1+v);
 der.p=-100*x*v/z;
 der.lx50=-100*p*v/z;
 der.p.p=100*x**2*v*(v-1)/z3;
 der.lx50.lx50=100*p**2*v*(v-1)/z3;
 der.p.lx50=100*v*((p*lx50-u*p-1)*v+u*p-p*lx50-1)/z3;
run;
                                Newton-Raphson
 Non-Linear Least Squares Iterative Phase     Dependent Variable RESPONS
Method: Newton
                     Iter           LX50              P Sum of Squares
                        0      -0.360000       0.730000    4308.201154
                        1      -0.232223       0.662451    4120.293002
                        2      -0.220547       0.668049    4118.668060
                        3      -0.220609       0.668137    4118.667853
                        4      -0.220609       0.668137    4118.667853
NOTE: Convergence criterion met.

          Non-Linear Least Squares Summary Statistics     Dependent Variable RESPONS

               Source                DF Sum of Squares     Mean Square

               Regression             2   90581.332147    45290.666074
               Residual              38    4118.667853      108.385996
               Uncorrected Total     40   94700.000000

               (Corrected Total)     39   34792.400000

               Parameter     Estimate    Asymptotic             Asymptotic 95 %
                                         Std. Error          Confidence Interval
                                                             Lower         Upper
                   LX50  -.2206090381 0.14256080716 -.50920682795 0.06798875174
                      P  0.6681368651 0.06607712608 0.53437140087 0.80190232925

                               Asymptotic Correlation Matrix
                          Corr                  LX50                 P
                          ffffffffffffffffffffffffffffffffffffffffffff
                              LX50                 1      -0.060005666
                                 P      -0.060005666                 1
```

A.5 Assessment of fit

- Plot of observed versus predicted values

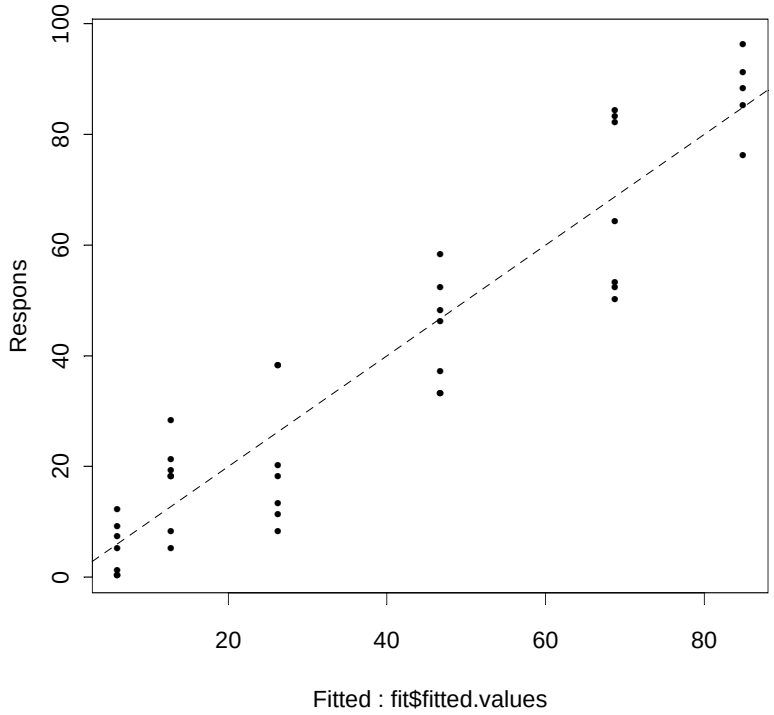

Figure A.2 Plot of observed versus predicted values

- Analysis of residuals

    - Notice that in nonlinear regression the residuals can have mean value ≠ 0.

```
> res<-fit$residuals
> mean(res)
[1] 0.771735
```

    - Computation of studentized residuals: one extreme observation (# 33)

```
> nls.studres
function(grad, nls.fit)
{
       v <- attributes(grad)$gradient
       h <- diag(v %*% solve(t(v) %*% v) %*% t(v))
       s <- summary(nls.fit)$sigma
       r <- nls.fit$residuals/(s * sqrt(1 - h))
       r
}
> pars<-fit$parameters
> resid<-nls.studres(der(pars["lx50"],pars["p"],Receptor.data$Dose),fit)
> Receptor.data[abs(stresid)>qt(0.95,38),]
   Dose Respons     LDose
33  2.5      51 0.9162907
> qqnorm(resid)
```

- Normal quantile plot of studentized residuals: normality seems to be a plausible assumption

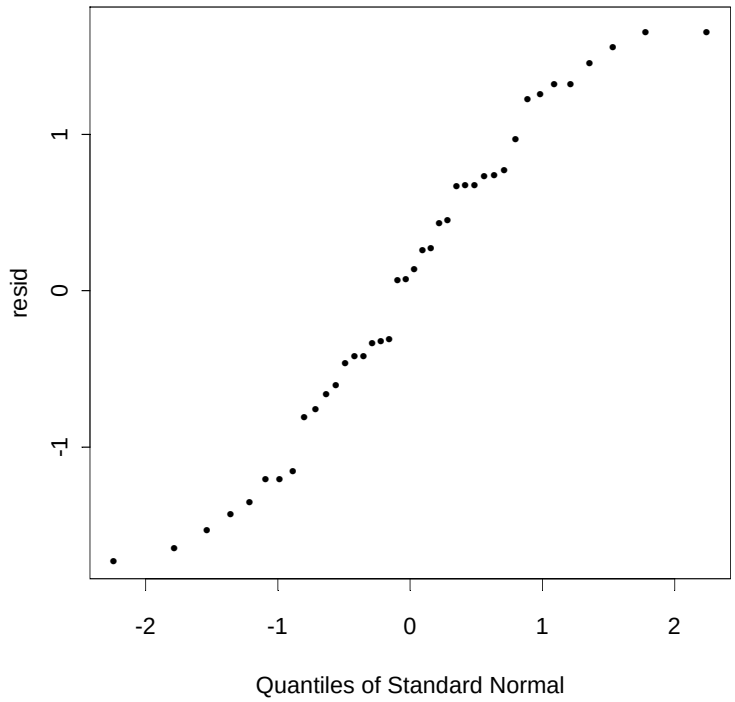

Figure A.3 Normal quantile plot of studentized residuals

- Correlation between quantiles and expected value under normality (critical value $\alpha = 0.1$, n = 40: 0.977 (Neter, Wasserman, and Kutner, 1990).

```
> nqcor
function(x)
{
# Correlation between quantiles and expected value under normality
       xs <- sort(x)
       n <- length(x)
       xn <- qnorm(((1:n) - 0.375)/(n + 0.25))
       c(n, cor(xs, xn))
}
> nqcor(fita.stres)
[1] 40.0000000  0.9838331
```

- Plot of studentized residuals versus predicted values

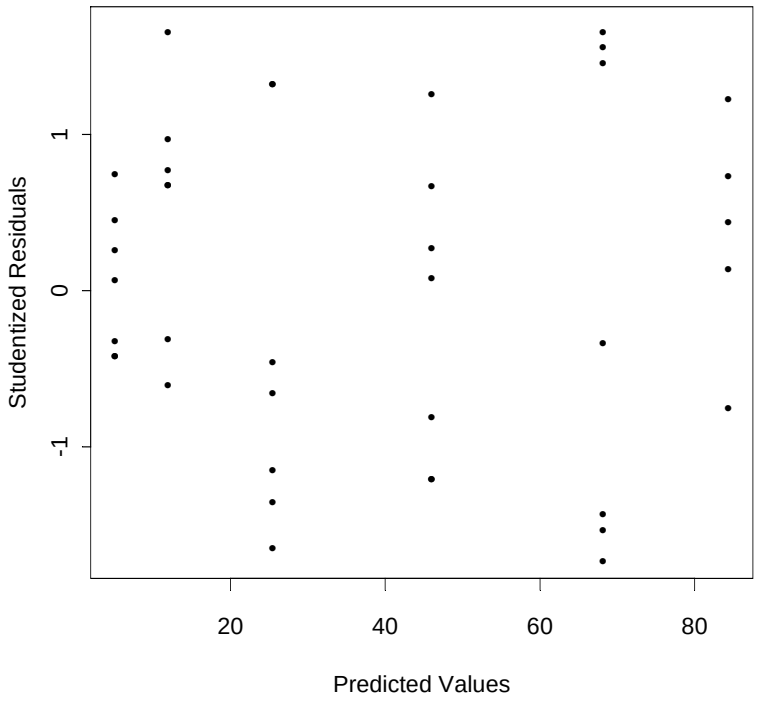

Figure A.4 Studentized residuals versus predicted values

- Influence of observation 33, Cook's D

```
> nls.CookD
function(grad, nls.fit)
{
       v <- attributes(grad)$gradient
       h <- diag(v %*% solve(t(v) %*% v) %*% t(v))
       s <- summary(nls.fit)$sigma
       l <- h/((1 - h)^2)
       D <- (l * (nls.fit$residuals^2))/(length(nls.fit$parameters) * s^2)
       D
}
> CooksD<-nls.CookD(der(pars["lx50"],pars["p"],Receptor.data$Dose),fit)
> CooksD[33]
[1] 0.1030288
> 100*pf(CooksD[33],2,38)
[1] 10.46598
```

- Formal test on lack of fit

```
> sse<-sum(fit$residuals^2)
> df.sse<-summary(fit)$df[2]
> attach(Receptor.data)
> pure.error<-aov(Respons~as.factor(Dose))
> sspe<-sum(pure.error$residuals^2)
> df.sspe<-pure.error$df.residual
c(sse,df.sse)
[1] 4118.668   38.000
> c(sspe,df.sspe)
[1] 3705.657   34.000
> sslf<-sse-sspe
> df.lf<-df.sse-df.sspe
> c(sslf,df.lf)
[1] 413.0107   4.0000
> mslf<-sslf/df.lf
> mspe<-sspe/df.sspe
> mspe
[1] 108.9899
> mslf
[1] 103.2527
> f<-mslf/mspe
> f
[1] 0.94736
> 1-pf(f,df.lf,df.sspe)
[1] 0.4487107
```

A.6 Assessment of curvature

- Graphical assessment of parameter curvature:

```
> par(mfrow=c(1,2),pty="s")
> plot(profile(fit),ask=F)
```

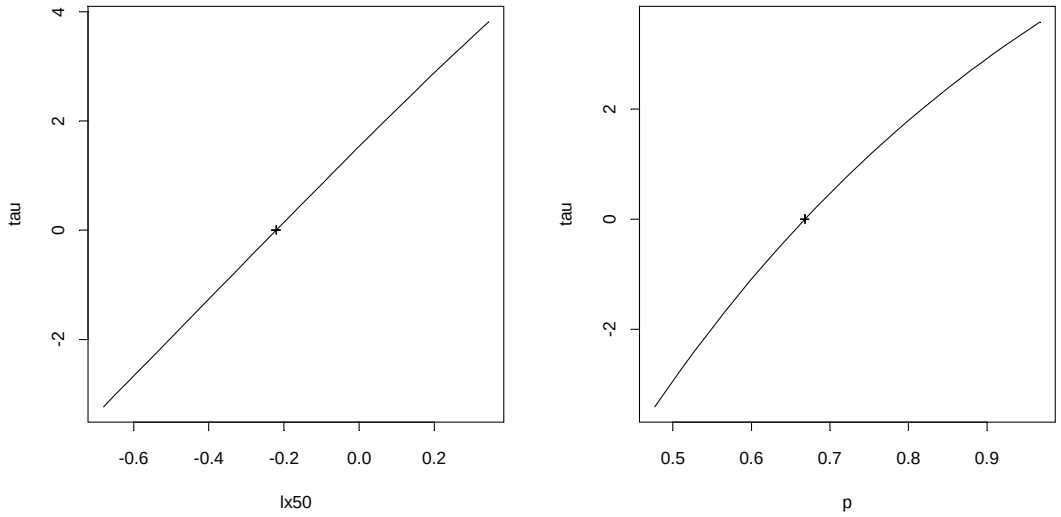

Figure A.5 Profile *t* plot of $\xi = \log x_{50}$ and p

- Numerical measures of intrinsic and parameter curvature:

```
> library(mass,first=T)
> enlist
function(vec)
{
        x <- as.list(vec)
        names(x) <- names(vec)
        x
}
> plist<-enlist(coef(fit))
> der2<-deriv3(Respons~100/(1+exp(p*(lx50-log(x)))),c("lx50","p"),function(lx50,p,x)
+ NULL)
> fit.he<-nls(Respons~der2(lx50,p,Dose),start=plist,data=Receptor.data)
> rms.curv(fit.he)
Parameter effects: c^theta x sqrt(F) = 0.225
        Intrinsic: c^iota  x sqrt(F) = 0.1282
```

- Wald and profile joint confidence regions (using the *ellipse*) library

```
> library(ellipse,first=T)
> nls.confreg
function(thefit, which = c(1, 2), alphamax = 0.05)
{
        ell1 <- ellipse(thefit, which = which, level = 0.95)
        ell2 <- ellipse(profile(thefit, alphamax = alphamax), which = which, level =
0.95)
        xrange <- quantile(c(ell1[, 1], ell2[, 1]), probs = c(0, 1))
        yrange <- quantile(c(ell1[, 2], ell2[, 2]), probs = c(0, 1))
        plot(ell2, type = "l", pty = "s", xlim = xrange, ylim = yrange)
        lines(ell1, lty = 2)
        points(thefit$parameter[which[1]], thefit$parameter[which[2]], pch = 3)
}
```

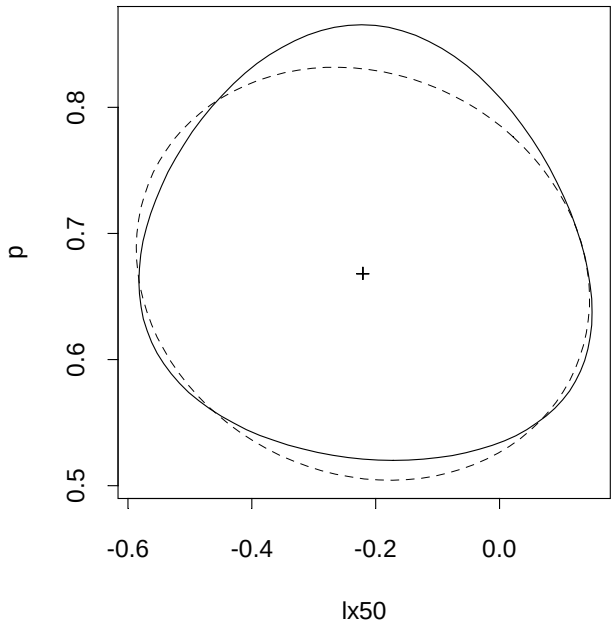

Figure A.6 Joint 95% Wald (broken line) and likelihood (solid line) confidence regions.

A.7 Profile likelihood confidence intervals

```
> Conf.int
function(object, parm, level = 0.95)
{
# Author: W. Venables
        if(is.character(parm))
                parm <- match(parm, names(coef(object)), nomatch = 0)
        pro <- profile(object, which = parm, alphamax = (1 - level)/4[[parm]]
        x <- pro[, "par.vals"][, parm]
        y <- pro$tau
        n <- length(object$fitted.values) - length(object$parameters)
        cutoff <- qt(1 - (1 - level)/2, n)
        if(max(y) < cutoff & min(y) >  - cutoff)
                stop("profiling did not extend far enough to give a complete interval")
        sp <- spline(x, y)
        approx(sp$y, sp$x, xout = c( - cutoff, cutoff))$y
}
> Conf.int(fit,"lx50")
[1] -0.50805146  0.07189714
> Conf.int(fit,"p")
[1] 0.5475247 0.8194141
> exp(Conf.int(fit,"lx50"))
[1] 0.6016668 1.0745448
```

A.8 Wald confidence intervals in S

```
> est<-summary(fit)$parameters
> tval<-qt(1 - 0.5 * 0.05, summary(fit)$df[2])
> conf<-cbind((est[,1]-est[,2]*tval),est[,1]+est[,2]*tval)
> conf
           [,1]       [,2]
lx50 -0.5115527 0.07028779
   p  0.5379492 0.79825903
> exp(conf[1,])
[1] 0.5995639 1.0728169
```

Appendix B
Inhibition of growth rate of micro-organisms

B.1 Data (source: D. Weytjens)

Table B.1 Growth rate of micro-organisms at different concentrations of drug

Concentration (mg/l)					
0.01	1	3.2	10	32	100
3.9546	4.0735	2.3507	-0.9894	-1.075	-1.0750
4.5965	3.8513	2.2504	-1.075	-0.8687	-1.7289
4.1846	3.7039	1.8368	-0.6974	-1.075	-0.9894

- Parameter coding of model (2.15): y_{min} = y0, y_{max} = y1, ξ = log x_{50} = lx50, β = b

B.2 Starting Values

Starting values for y_{min} and y_{max} are obtained by taking the minimum and maximum of the data. Next, the response is standardized to a range 0 - 1 and transformed to logits. From a plot of the logits versus the logarithm of the dose, it follows that only the central portion of the data can be approximated by a line. Linear regression yields -1.6465 as an estimate of the slope and 2.0909 as estimate for the intercept. Hence, the starting value for β is -1.65 and for ξ = 2.0909/1.6465 = 1.27.

```
> attach(Micro.organisms)
> min(Respons)
[1] -1.7289
> max(Respons)
[1] 4.5965 >
y<-(Respons-min(Respons)+0.1)/(0.2+max(Respons)-min(Respons))
y2<-log(y/(1-y))
plot(Dose,y2,log="x",xlab="Dose",ylab="logit(standardized growth rate)")
> summary(lm(y2[ii]~log(Dose[ii])))
Call: lm(formula = y2[ii] ~ log(Dose[ii]))
Residuals:
     Min      1Q  Median     3Q    Max
 -0.3728 -0.2126 0.07253 0.1577 0.4018

Coefficients:
                Value Std. Error  t value Pr(>|t|)
  (Intercept)   2.0909   0.1615    12.9483   0.0000
log(Dose[ii])  -1.6465   0.1084   -15.1862   0.0000

Residual standard error: 0.3058 on 7 degrees of freedom
Multiple R-Squared: 0.9705
F-statistic: 230.6 on 1 and 7 degrees of freedom, the p-value is 1.292e-006

Correlation of Coefficients:
             (Intercept)
log(Dose[ii]) -0.7757
> 2.0909/1.6456
[1] 1.2706
```

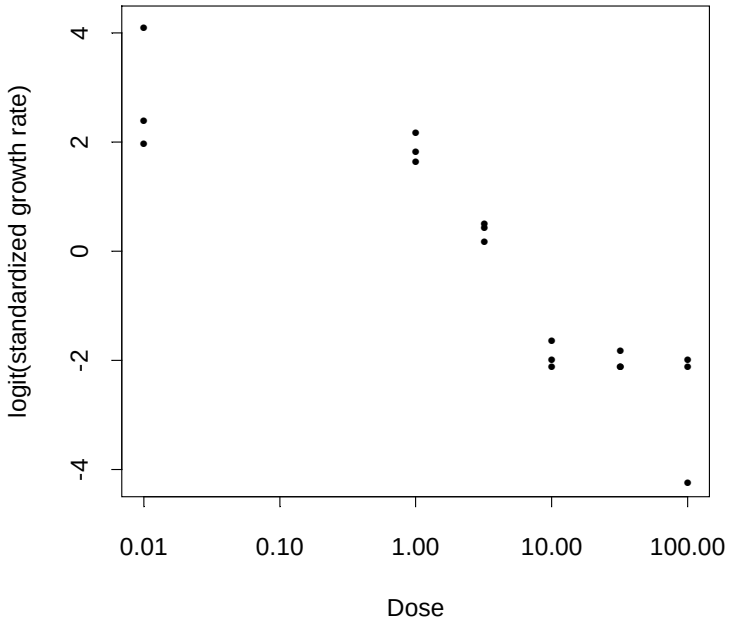

Figure B.1 Plot of logit of standardized growth rate versus dose, the central portion can be used to determine starting values.

B.3 Nonlinear regression

- Parameter coding: y_{min} = y0, y_{max} = y1, ξ = lx50, β = b

- Gradient

```
> der4l<-deriv(~y0+(y1-y0)/(1+exp(b*(lx50-log(x)))),c("y0","y1","b","lx50"),
+ function(y0,y1,b,lx50,x) NULL)
> der4l
function(y0, y1, b, lx50, x)
{
        .expr1 <- y1 - y0
        .expr3 <- lx50 - (log(x))
        .expr5 <- exp((b * .expr3))
        .expr6 <- 1 + .expr5
        .expr9 <- 1/.expr6
        .expr13 <- .expr6^2
        .value <- y0 + (.expr1/.expr6)
        .grad <- array(0, c(length(.value), 4), list(NULL, c("y0","y1","b","lx50")))
        .grad[, "y0"] <- 1 - .expr9
        .grad[, "y1"] <- .expr9
        .grad[, "b"] <-  - ((.expr1 * (.expr5 * .expr3))/.expr13)
        .grad[, "lx50"] <-  - ((.expr1 * (.expr5 * b))/.expr13)
        attr(.value, "gradient") <- .grad
        .value
}
```

- Gauss-Newton method

```
> fit4l<-nls(Respons~der4l(y0,y1,b,lx50,Dose),Micro.organisms,
+ start=list(y0=-1.7,y1=4.6,b=-1.6,lx50=1.3),trace=T)
3.08918 : -1.7 4.6 -1.6 1.3
1.40155 : -1.15282 4.15135 -2.16409 1.33079
1.09434 : -1.17481 4.13764 -2.61591 1.33939
1.07323 : -1.17086 4.1298 -2.80806 1.33583
1.07228 : -1.16898 4.12592 -2.85643 1.33578
1.07225 : -1.16836 4.12488 -2.86566 1.3358
1.07225 : -1.16824 4.12467 -2.86729 1.33581
> summary(fit4l)

Formula: Respons ~ der4l(y0, y1, b, lx50, Dose)

Parameters:
       Value Std. Error   t value
  y0 -1.16824  0.1158790 -10.08160
  y1  4.12467  0.1248770  33.02980
   b -2.86729  0.5653110  -5.07205
lx50  1.33581  0.0577111  23.14650

Residual standard error: 0.276747 on 14 degrees of freedom

Correlation of Parameter Estimates:
         y0     y1      b
  y1 -0.147
   b -0.536  0.342
lx50 -0.486 -0.222  0.448
```

B.4 Assessment of fit

- Plot of observed versus predicted values

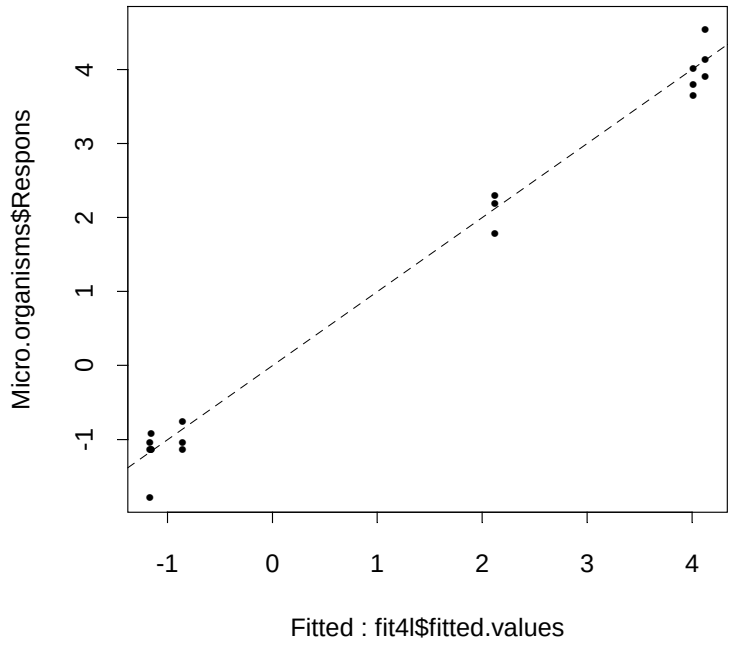

Figure B.2 Plot of observed versus predicted values

- Analysis of studentized residuals

```
> pars<-fit4l$parameters
>  stresid<-
nls.studres(der4l(pars["lx"],pars["b"],pars["y0"],pars["y1"],Micro.organisms$Dose),
+ fit4l)
> stresid[abs(stresid)>qt(0.95,14)]
[1]  1.909224 -2.482780
> Micro.organisms[abs(stresid)>qt(0.95,14),]
     Dose Respons
 2   0.01  4.5965
17 100.00 -1.7289
> qqnorm(stresid)
> nqcor(stresid)
[1] 18.0000000  0.9751829 # Critical value n = 20, alpha = 0.1: 0.960
```

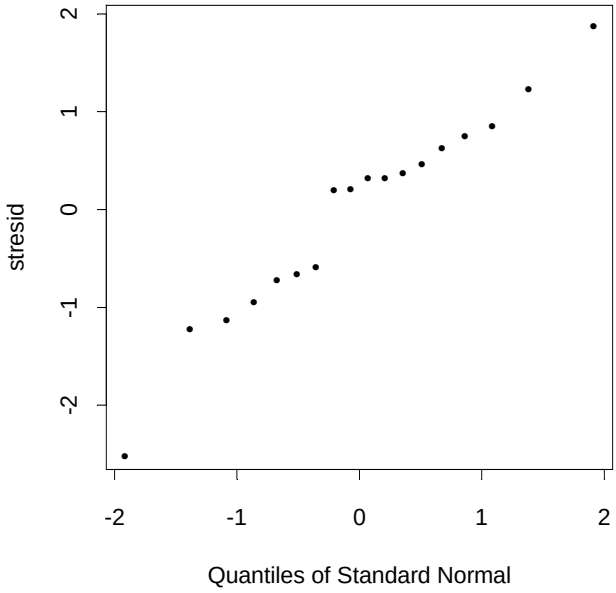

Figure B.3 Normal quantile plot of studentized residuals

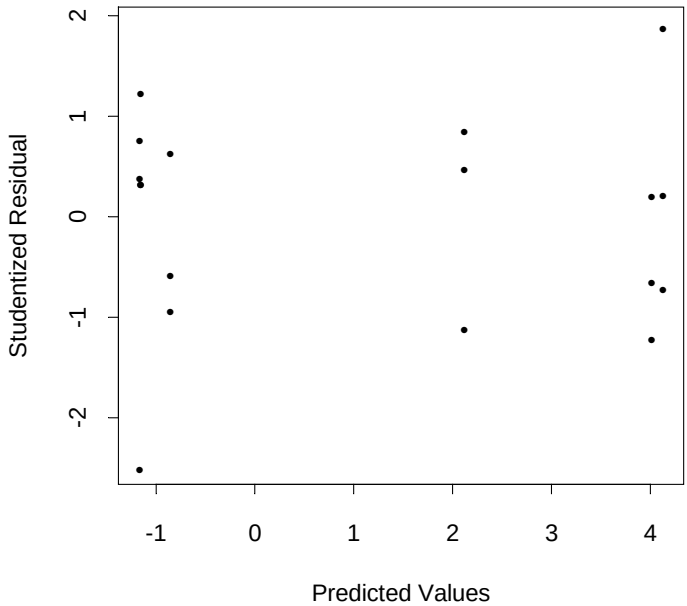

Figure B.4 Studentized residuals versus predicted values

No real outlying observations present (absolute value of studentized residual > 4). Two observations can be considered as extreme, namely the minimum and maximum of the data.

- Cook's D: influence of extreme values:

```
> CooksD<-nls.CookD(der4l(pars["lx"],pars["b"],pars["y0"],pars["y1"],
+                   Micro.organisms$Dose),fit4l)
> CooksD[c(2,17)]
[1] 0.2315114 0.7697355
> 100*pf(CooksD[c(2,17)],4,14)
[1]  8.394481 43.746376
```

- Comparison of models with possible influential case (# 17) omitted

Original fit:

```
> summary(fit4l)$parameters
Value Std. Error    t value
  y0 -1.168237 0.11587865 -10.081553
  y1  4.124673 0.12487742  33.029777
   b -2.867285 0.56531068  -5.072052
lx50  1.335813 0.05771112  23.146546
```

Observation 17 omitted:

```
> summary(fit4l2)$parameters
         Value Std. Error    t value
  y0 -1.051851 0.10417604 -10.096859
  y1  4.101381 0.10223425  40.117486
   b -3.270274 0.75983832  -4.303908
lx50  1.309017 0.04943874  26.477559
```

- Formal test on lack of fit

```
> sse<-sum(fit4l$residual^2)
> df.sse<-summary(fit4l)$df[2]
> attach(Micro.organisms)
> pure.error<-aov(Respons~as.factor(Dose))
> sspe<-sum(pure.error$residual^2)
> df.sspe<-pure.error$df.residual
> c(sse,df.sse)
[1]  1.072246 14.000000
> c(sspe,df.sspe)
[1]  0.8631949 12.0000000
> sslf<-sse-sspe
> df.lf<-df.sse-df.sspe
> c(sslf,df.lf)
[1] 0.209051 2.000000
> mslf<-sslf/df.lf
> mspe<-sspe/df.sspe
> mspe
[1] 0.0719329
> mslf
[1] 0.1045255
> f<-mslf/mspe
> f
[1] 1.453097
> 1-pf(f,df.lf,df.sspe)
[1] 0.2721992
```

B.5 Assessment of curvature

- Graphical assessment of parameter curvature:

```
> par(mfrow=c(2,2),pty="s")
> plot(profile(fit4l),ask=F)
Error in profile(fit4l): 6 NAs found in gradient
Dumped
Error was while calling subroutine "do_nls"
> plot(profile(fit4l,alphamax=0.1),ask=F)
```

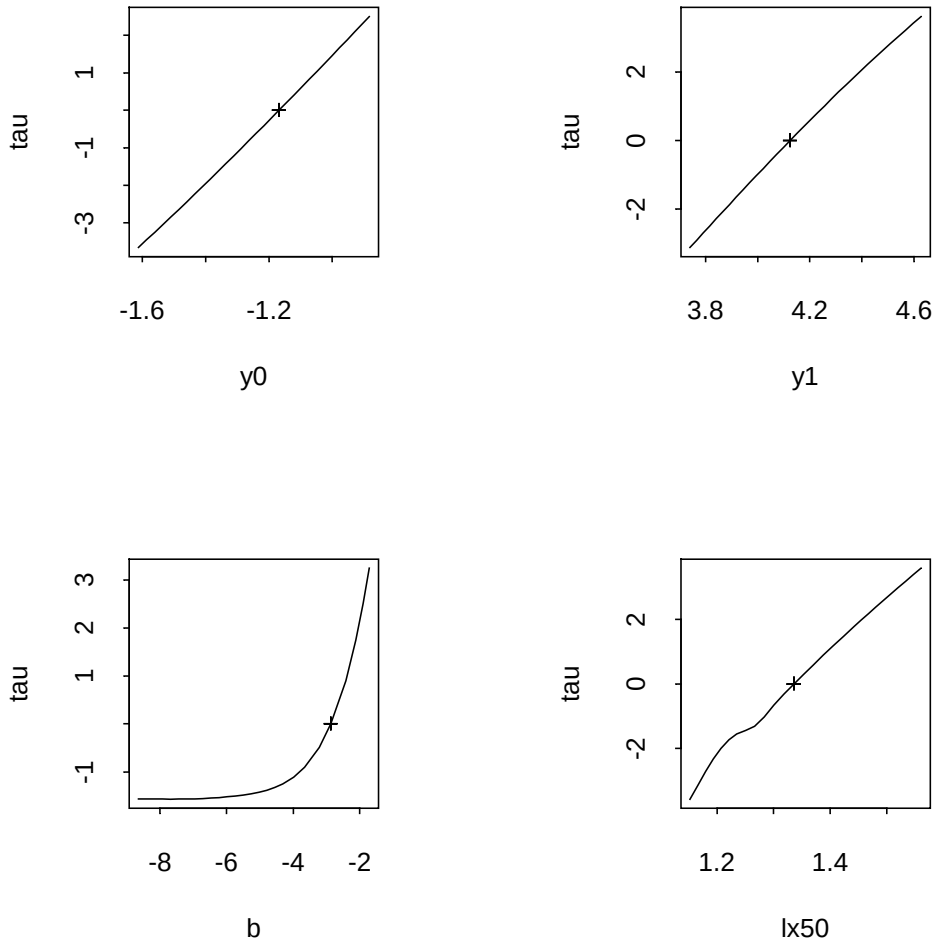

Figure B.5 Profile t plots of the estimates for y_{min} (y0), y_{max} (y1), β (b), and ξ (lx50)

- Numerical measures of intrinsic and parameter curvature:

```
>  library(mass,first=T)
>  der4l2<-deriv3(~y0+(y1-y0)/(1+exp(b*(lx50-log(x)))),c("y0","y1","b","lx50"),
+ function(y0,y1,b,lx50,x) NULL)
>  fit4l2<-nls(Respons~der4l2(y0,y1,b,lx50,Dose),Micro.organisms,
+ start=list(y0=-1.7,y1=4.6,b=-1.6,lx50=1.3))
> rms.curv(fit4l2)
Parameter effects: c^theta x sqrt(F) = 1.0081
        Intrinsic: c^iota  x sqrt(F) = 0.1507
```

- 2-Dimensional projections of joint confidence regions (Wald & likelihood)

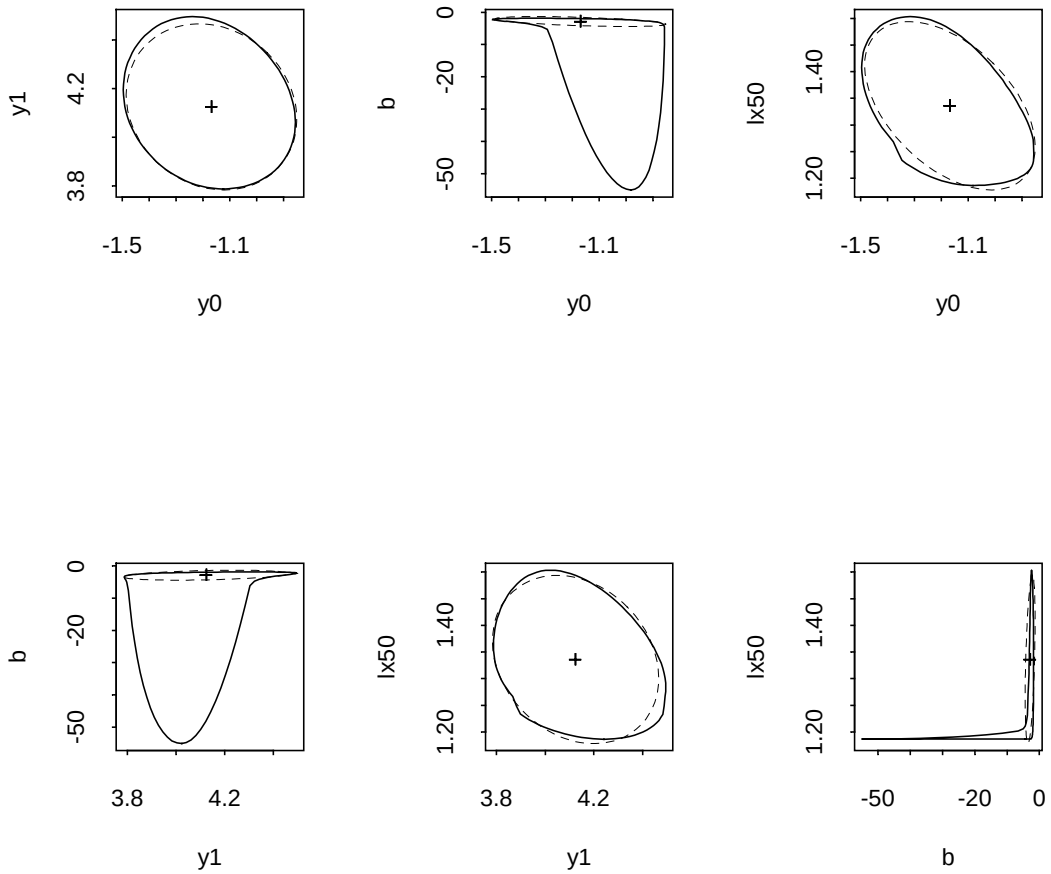

Figure B.6 2-Dimensional projections of joint 95 % Wald (broken line) and likelihood (solid line) confidence regions

B.6 Wald-type confidence intervals

```
> tval<-qt(1-0.5*0.05,summary(fit4l)$df[2])
> est<-summary(fit4l)$parameters
> conf<-cbind((est[,1]-est[,2]*tval),(est[,1]+est[,2]*tval) )
> conf
          [,1]       [,2]
  y0 -1.416772 -0.9197018
  y1  3.856838  4.3925089
   b -4.079756 -1.6548146
lx50  1.212035  1.4595911
```

B.7 Profile likelihood confidence intervals

```
> Conf.int(fit4l,"y0")
[1] -1.423528        NA
> Conf.int(fit4l,"y1")
[1] 3.858230 4.411603
> Conf.int(fit4l,"lx50")
[1] 1.198723 1.464992
> Conf.int(fit4l,"b")

[1]        NA -2.014537
```

Appendix C
Metrazol-induced seizure threshold in rats

C.1 Data (source: Ashton, et al. 1992)

Table C.1 Effect of diazepam alone, or in combination with loreclezole (10 mg/kg) on seizure threshold (mg/kg metrazole)

Dose diazepam (mg/kg p.o.)						
0	0.31	1.25	5	10	20	40
A. Control						
43.9	32.7	82.4	61.6	118	147.6	158.8
36.8	40	52.6	63	120.6	124.5	145.6
37.8	33.3	42.1	66.9	151.4	163.8	186.9
37.3	38.2	38.2	59.3	66.7	173.9	184.2
50	46.2	56.7	65.1	57.7	96.6	180.8
45.4	37.8	52.6	58.4	95.3	104.3	207.4
49.1	42.9	71.1	54.1	77.9	141.4	219.7
58.6	41.7	36	58.3	112.1	127.8	193.3
44.5	43.3	45	72.8	58.1	142.5	160.1
B. Loreclezole 10 mg/kg p.o.						
53.9	48.2	53.2	159.5	132.8	456.1	260
64.9	57.1	88.9	123.6	208.2	191.5	177.8
77.5	59.3	80.7	140	118	246.8	336
46.3	44.5	49	85.6	138.6	361.2	186.7
78.7	78.4	94.6	119.7	200	270.5	189.2
89.2	62.2	100	102.7	139.7	306.4	194.3
103.6	71.9	82.1	153	177.1	288.8	233.4
94.7	78.5	61.8	163	167.6	224.9	242.7
45.7	54.4	65	84.3	101	207.4	319.1

- Zero-dose displacement:

```
Metrazol.data$Dose[Metrazol.data$Dose==0]<-0.08
```

C.2 Starting Values

- Lower asymptote and range from the data
- Standardizing response to 0-1 scale and computing logit transformation

```
> Control<-Metrazol.data[Metrazol.data$Treat==0,]
> quantile(Control$Respons,probs=c(0,0.025,0.975,1))
   0.0%   2.5%   97.5% 100.0%
   32.7 34.785 199.645  219.7
> y<-(Control$Respons-30)/200
> y2<-logit(y)
> plot(Control$Dose,y2,log="x")
```

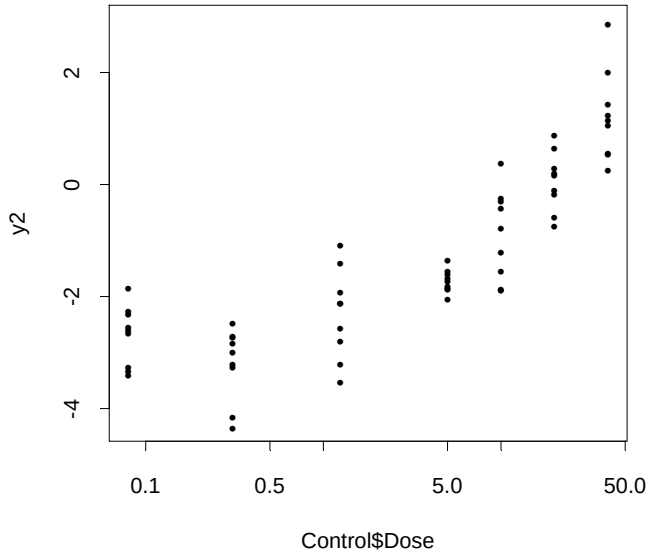

Figure C.1 Plot of logit of standardized response versus dose,
the relation is close to linear and is used for the initial values of slope and potency.

- Linear regression of logit standardized response versus logarithm of dose

```
> x<-log(Control$Dose)
> summary(lm(y2~x))
Call: lm(formula = y2 ~ x)
Residuals:
    Min      1Q   Median     3Q   Max
 -1.676 -0.6592 0.002549 0.5882 2.521

Coefficients:
               Value Std. Error  t value Pr(>|t|)
(Intercept)  -1.9417   0.1244   -15.6032   0.0000
          x   0.6326   0.0532    11.8873   0.0000

Residual standard error: 0.8893 on 61 degrees of freedom
Multiple R-Squared: 0.6985
F-statistic: 141.3 on 1 and 61 degrees of freedom, the p-value is 0
Correlation of Coefficients:
  (Intercept)
x -0.4352
> 1.9417/0.6326
[1] 3.069396
```

C.3 Unweighted Nonlinear Regression

```
> der.c0<-deriv(~
+       b0 + b4 * Treat + (b1 + b5 * Treat)/(1 + exp((b2 + b6 * Treat)
+       * (b3 + b7*Treat -  log(Dose)))),
+       c("b0","b1","b2","b3","b4","b5","b6","b7"),
+       function(b0,b1,b2,b3,b4,b5,b6,b7,Dose,Treat) NULL)
> fit.c0<-nls(Respons~der.c0(b0,b1,b2,b3,b4,b5,b6,b7,Dose,Treat),Metrazol.data,
+       start=list(b0=30,b1=190,b2=0.6,b3=3,b4=0,b5=0,b6=0,b7=0),trace=F)
> summary(fit.c0)
Formula: Respons ~ der.c0(b0, b1, b2, b3, b4, b5, b6, b7, Dose, Treat)
Parameters:
         Value Std. Error    t value
b0   43.814900   8.182360  5.3548000
b1   195.012000 106.520000  1.8307600
b2    1.417910   0.771693  1.8374000
b3    3.066200   0.847489  3.6179900
b4   29.363400  10.915700  2.6900300
b5   -2.276990 107.846000 -0.0211133
b6    1.295100   1.083250  1.1955700
b7   -0.838695   0.855061 -0.9808590
Residual standard error: 36.9351 on 118 degrees of freedom
Correlation of Parameter Estimates:
       b0     b1     b2     b3     b4     b5     b6
b1 -0.440
b2  0.513 -0.934
b3 -0.296  0.979 -0.908
b4 -0.750  0.329 -0.384  0.222
b5  0.434 -0.988  0.923 -0.967 -0.382
b6 -0.365  0.666 -0.712  0.647  0.424 -0.733
b7  0.294 -0.970  0.900 -0.991 -0.199  0.967 -0.670
```

- Plot of observed versus predicted values

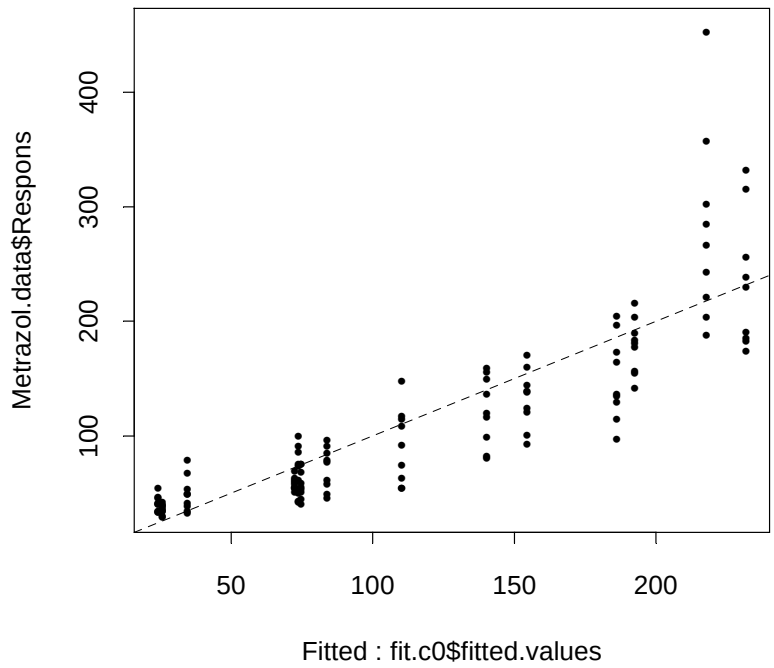

Figure C.2 Observed versus predicted response from an unweighted fit.

- Spearman correlation and plot of logarithm of absolute value of residuals versus logarithm of predicted values

```
> pars<-fit.c0$parameters
> fit.c0.stres<-nls.studres(der.c0(pars[1],pars[2],pars[3],pars[4],pars[5],
+      pars[6],pars[7],pars[8],Metrazol.data$Dose,Metrazol.data$Treat),fit.c0)
> cor.test(fit.c0$fitted.value,abs(fit.c0.stres),method="spearman")
        Spearman's rank correlation
data:  fit.c0$fitted.value and abs(fit.c0.stres)
normal-z = 6.5194, p-value = 0
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
 0.5831143
> x<-log(fit.c0$fitted.values)
> y<-log(abs(fit.c0.stres))
> quantile(y,probs=c(0.01,0.99))
       1%      99%
 -4.054001 1.104495
> plot(x[y>-4],y[y>-4],xlab="Logarithm of Predicted Value",ylab="Logarithm of Absolute
Studentized Residual")
> lines(lowess(x[y>-4],y[y>-4]) )
```

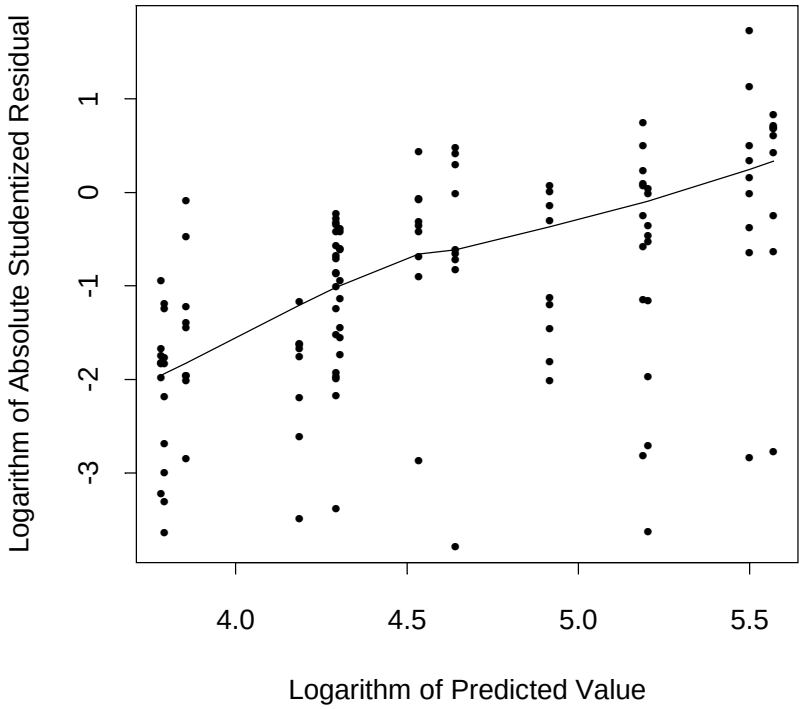

Figure C.3 Scatterplot with lowess smoothing of logarithms of absolute values of studentized residuals versus logarithms of predicted values from an unweighted least squares fit. Smallest three values of the absolute residuals deleted.

- Linear regression on logarithms of absolute values of studentized residuals versus logarithms of predicted values

```
> lm(y~x)
Call:
lm(formula = y ~ x)
Coefficients:
 (Intercept)        x
   -6.356534 1.155136
Degrees of freedom: 126 total; 124 residual
Residual standard error: 1.167237
```

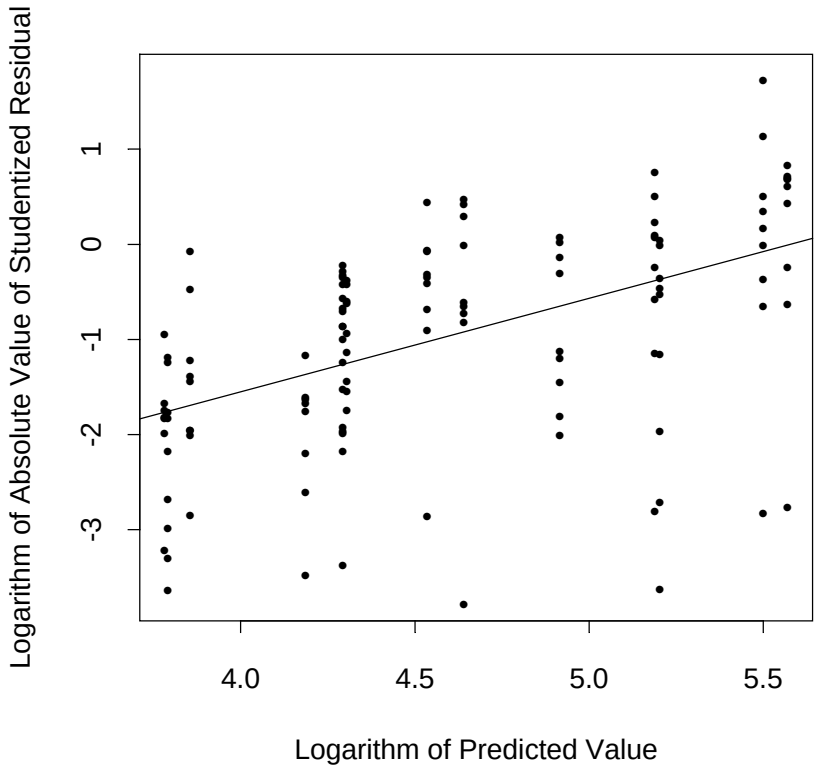

Figure C.4 Scatterplot with line of best fit of logarithms of absolute values of studentized residuals versus logarithms of predicted values from an unweighted least squares fit. Smallest three values of the absolute residuals deleted.

C.4 Implementing generalized least squares with pseudo-likelihood estimation of the power of the mean variance function

- General function for GLS with pseudo-likelihood estimation

```
> nls.genls
function(fmodel, data, start, Respons, nriter = 1)
{
# Generalized least squares for variance being a power of the mean
   theDat <- cbind(data, 1)   # initial weights = 1
   v <- (dimnames(theDat)[[2]])
   v[dim(theDat)[2]] <- "w"
   dimnames(theDat)[[2]] <- v
   fiti <- nls(fmodel, theDat, start = start, trace = T) # fit model of form (Y-Pred)
   fiti$fitted.values <- fiti$fitted.values + theDat[,Respons]
# Iterate
for(iter in 1:nriter) {
       gmean <- exp(mean(log(fiti$fitted.values)))  # Geometric mean of fitted values
       vfe <- ms( ~ (fiti$residuals * (gmean/fiti$fitted.values)^theta)^2, start =
              list(theta = 0), trace = T)   # Variance function estimation
   # Refit model using estimated weights
       theDat$w <- (fiti$fitted.values)^(-2 * vfe$parameters) # Estimated weights
       fiti <- nls(fmodel, theDat, start = start, trace = T) # fit
       fiti$fitted.values <- fiti$fitted.values/sqrt(theDat$w) + theDat[,Respons]
       fiti$residuals <- fiti$residuals/sqrt(theDat$w)
   }
   fiti$variance.par <- vfe$parameters
   fiti$GLSweights <- theDat$w
   fiti$residuals <- fiti$residuals * sqrt(fiti$GLSweights)
   fiti fiti
}
```

- Definition of estimation function of the problem and sample run.

```
> logistic8p
function(resp, treat, Dose, b0, b1, b2, b3, b4, b5, b6, b7, w)
{
       Pred <- b0 + b4 * treat + (b1 + b5 * treat)/(1 + exp((b2 + b6 * treat) * (b3 +
+      b7 * treat - log(Dose))))
       (resp - Pred)*sqrt(w)
}
 > fit.c1<-nls.genls(~logistic8p(Respons,Treat,Dose,b0,b1,b2,b3,b4,b5,b6,b7,w),
+      Metrazol.data,start=list(b0=43.8,b1=195,b2=1.42,b3=3.1,b4=29.4,
+      b5=-2.28,b6=1.30,b7=-0.84),"Respons",nriter=5)
```

```
> summary(fit.c5)

Formula:  ~ logistic8p(Respons, Treat, Dose, b0, b1, b2, b3, b4, b5, b6, b7, w)

Parameters:
         Value Std. Error    t value
b0   43.130700   2.587190  16.670900
b1  241.462000 154.960000   1.558220
b2    1.222640   0.397207   3.078090
b3    3.413250   1.025400   3.328710
b4   24.516900   4.622620   5.303670
b5  -26.630600 160.327000  -0.166101
b6    0.516366   0.662590   0.779314
b7   -1.152480   1.067280  -1.079830

Residual standard error: 0.136789 on 118 degrees of freedom

Correlation of Parameter Estimates:
       b0     b1     b2     b3     b4     b5     b6
b1 -0.469
b2  0.601 -0.909
b3 -0.463  0.991 -0.938
b4 -0.560  0.262 -0.336  0.259
b5  0.453 -0.967  0.878 -0.958 -0.334
b6 -0.360  0.545 -0.599  0.562  0.522 -0.687
b7  0.445 -0.952  0.901 -0.961 -0.299  0.985 -0.715
```

Table C.2 Effect of number of iterations of generalized least squares on parameter estimates and their standard errors (italic).

Iterations	Parameter								
	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	θ
0*	43.82	195.0	1.418	3.066	29.36	-2.277	1.295	-0.839	-
	8.18	*106.5*	*0.772*	*0.847*	*10.92*	*107.85*	*1.083*	*0.855*	
1	43.19	240.4	1.227	3.408	24.28	-27.547	0.468	-1.161	1.110
	2.63	*151.9*	*0.398*	*1.012*	*4.93*	*157.30*	*0.648*	*1.055*	
2	43.13	241.9	1.222	3.416	24.54	-27.225	0.525	-1.157	1.137
	2.58	*155.9*	*0.397*	*1.029*	*4.61*	*161.16*	*0.666*	*1.071*	
3	43.13	241.4	1.223	3.413	24.51	-26.466	0.514	-1.151	1.131
	2.59	*154.8*	*0.397*	*1.025*	*4.62*	*160.20*	*0.662*	*1.067*	
5	43.13	241.5	1.223	3.413	24.51	-26.504	0.515	-1.152	1.132
	2.59	*155.0*	*0.397*	*1.025*	*4.62*	*160.35*	*0.662*	*1.067*	
10	43.13	241.5	1.223	3.413	24.51	-26.504	0.515	-1.152	1.132
	2.59	*155.0*	*0.397*	*1.025*	*4.62*	*160.36*	*0.662*	*1.067*	
100	43.13	241.5	1.223	3.413	24.51	-26.504	0.515	-1.152	1.132
	2.59	*155.0*	*0.397*	*1.025*	*4.62*	*160.36*	*0.662*	*1.067*	

*: unweighted nonlinear least squares

C.5 SAS Implementation

```
data one;
  set metrazol;
  Y=respons;
%macro gls(nriter=2);
Title1 'Metrazol Data: Generalized Least Squares Analysis';
Title2 'Variance modelled as a power of the mean';
Title3 "Pseudo MLE of the variance power parameter, &Nriter iterations";
***           Unweighted Regression Stage                      ***;
Proc Nlin;
  parms b0=44 b1=195 b2=1.4 b3=3.1 b4=29 b5=-2.3 b6=1.3 b7=-0.84;
  f=b0+b4*treat+(b1+b5*treat)/(1+exp((b2+b6*treat)*(b3+b7*treat-log(dose))));
  model Y=f;
  output out=two r=resid p=pred;
run;
%DO I=1 %TO &NRITER;
   data three;set two;lpred=log(pred);run;
   *** Compute and output the geometric mean of the fitted values ***;
   Proc Means noprint;
    var lpred;
    output out=four mean=mlog;
   data five;merge three four;
    retain x 0;
    if _n_=1 then x=mlog;
    if mlog=. then mlog=x;
   *** Estimate the variance parameter by nonlinear least squares ***;
   data six;
     set five;
     Dummy=0;
     Taudot=exp(mlog);
   Proc nlin noprint;
    parms gamma=1;
    bounds gamma>-1.5, gamma<1.5;
    g=(Resid)*((Taudot/Pred)**gamma);
    model dummy=G;
    output out=seven parms=gamma;
   *** Stage 3 : Weighted estimation of the regression parameters ***;
   data eight;
     merge six seven;
     v=pred**(2*gamma);
     wt=1/v;
     drop gamma pred resid;
   proc nlin noprint;
     parms b0=44 b1=195 b2=1.4 b3=3.1 b4=29 b5=-2.3 b6=1.3 b7=-0.84;
     f=b0+b4*treat+(b1+b5*treat)/(1+exp((b2+b6*treat)*(b3+b7*treat-log(dose))));
     model y=f;
     _weight_=wt;
     output out=two r=resid p=pred;
   run;
%end;
   proc nlin data=eight;
     parms b0=44 b1=195 b2=1.4 b3=3.1 b4=29 b5=-2.3 b6=1.3 b7=-0.84;
     f=b0+b4*treat+(b1+b5*treat)/(1+exp((b2+b6*treat)*(b3+b7*treat-log(dose))));
     model y=f;
     _weight_=wt;
   run;
%mend;
%gls(nriter=100);
```

Metrazol Data: Generalized Least Squares Analysis
Variance modelled as a power of the mean
Pseudo MLE of the variance power parameter, 100 iterations

Non-Linear Least Squares DUD Initialization Dependent Variable Y

DUD	B0	B1	B2	B3	B4	B5	Weighted SS
	B6	B7					
-9	44.000000	195.000000	1.400000	3.100000	29.000000	-2.300000	2.378294
	1.300000	-0.840000					
-8	48.400000	195.000000	1.400000	3.100000	29.000000	-2.300000	2.570512
	1.300000	-0.840000					
-7	44.000000	214.500000	1.400000	3.100000	29.000000	-2.300000	2.411122
	1.300000	-0.840000					
-6	44.000000	195.000000	1.540000	3.100000	29.000000	-2.300000	2.415806
	1.300000	-0.840000					
-5	44.000000	195.000000	1.400000	3.410000	29.000000	-2.300000	2.846973
	1.300000	-0.840000					
-4	44.000000	195.000000	1.400000	3.100000	31.900000	-2.300000	2.415135
	1.300000	-0.840000					
-3	44.000000	195.000000	1.400000	3.100000	29.000000	-2.530000	2.378555
	1.300000	-0.840000					
-2	44.000000	195.000000	1.400000	3.100000	29.000000	-2.300000	2.392622
	1.430000	-0.840000					
…							
31	43.116751	241.752183	1.221443	3.415110	24.531578	-27.205468	2.276871
	0.520118	-1.156717					
32	43.123696	241.491557	1.222009	3.413453	24.523700	-26.653141	2.276869
	0.516826	-1.152575					
33	43.130241	241.448573	1.222645	3.412854	24.520114	-26.630849	2.276869
	0.515905	-1.152053					
34	43.131127	241.438931	1.222676	3.412897	24.516260	-26.589260	2.276869
	0.515609	-1.151983					
35	43.132888	241.138339	1.223309	3.411203	24.514015	-26.323847	2.276869
	0.515217	-1.150318					
36	43.134305	241.090329	1.223444	3.410878	24.512343	-26.298594	2.276869
	0.515445	-1.150171					
37	43.134255	241.175945	1.223358	3.411343	24.511927	-26.357054	2.276869
	0.515703	-1.150465					
38	43.135422	240.899077	1.223980	3.409481	24.514657	-26.064061	2.276869
	0.515309	-1.148614					
39	43.135499	240.912498	1.223938	3.409580	24.514342	-26.076639	2.276869
	0.515329	-1.148703					

NOTE: Convergence criterion met.

Non-Linear Least Squares Summary Statistics Dependent Variable Y

Source	DF	Weighted SS	Weighted MS
Regression	8	39.387601910	4.923450239
Residual	118	2.276868615	0.019295497
Uncorrected Total	126	41.664470524	
(Corrected Total)	125	10.079028956	

Parameter	Estimate	Asymptotic Std. Error	Asymptotic 95 % Confidence Interval Lower	Upper
B0	43.1354993	2.59431445	37.99801132	48.27298720
B1	240.9124979	154.17928292	-64.40677711	546.23177301
B2	1.2239381	0.39763612	0.43650432	2.01137185
B3	3.4095797	1.02194605	1.38583291	5.43332648
B4	24.5143424	4.62866638	15.34825321	33.68043166
B5	-26.0766392	159.42276290	-341.77951106	289.62623262
B6	0.5153286	0.66333902	-0.79827323	1.82893042
B7	-1.1487025	1.06312696	-3.25399935	0.95659428

Asymptotic Correlation Matrix

Corr	B0	B1	B2	B3	B4	B5	B6	B7
B0	1	-0.46939025	0.599689963	-0.46316333	-0.55986065	0.453864604	-0.35935602	0.445174779
B1	-0.46939025	1	-0.90883553	0.990667896	0.261747655	-0.96645992	0.545285935	-0.95188084
B2	0.599689963	-0.90883553	1	-0.93792723	-0.33654393	0.878427783	-0.60032575	0.901110162
B3	-0.46316333	0.990667896	-0.93792723	1	0.258510153	-0.95737302	0.562863996	-0.9607451
B4	-0.55986065	0.261747655	-0.33654393	0.258510153	1	-0.33330544	0.522556211	-0.29737004
B5	0.453864604	-0.96645992	0.878427783	-0.95737302	-0.33330544	1	-0.68743347	0.984516687
B6	-0.35935602	0.545285935	-0.60032575	0.562863996	0.522556211	-0.68743347	1	-0.71478186
B7	0.445174779	-0.95188084	0.901110162	-0.9607451	-0.29737004	0.984516687	-0.71478186	1

C.6 Assessment of generalized least squares fit

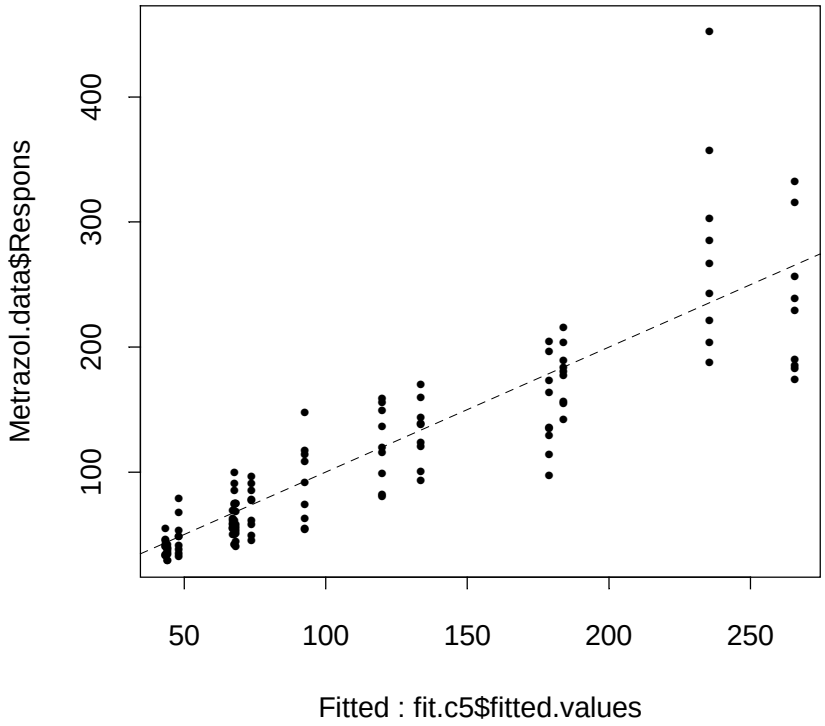

Figure C.5 Observed response versus values predicted from a generalized least squares fit.

```
> nls.studres
function(grad, nls.fit)
{
        v <- attributes(grad)$gradient * sqrt(nls.fit$GLSweights)
        h <- diag(v %*% solve(t(v) %*% v) %*% t(v))
        r <- (nls.fit$residuals)/(summary(nls.fit)$sigma * sqrt(1 - h))
        r
}
```

```
> pars<-fit.c5$parameters
>
> fit.c5.stres<-
nls.studres(der.c0(pars[1],pars[2],pars[3],pars[4],pars[5],pars[6],pars[7],pars[8],
+       Metrazol.data$Dose,Metrazol.data$Treat),fit.c5)
>
> cor.test(fit.c5$fitted.value,abs(fit.c5.stres),method="spearman")

         Spearman's rank correlation

data:  fit.c5$fitted.value and abs(fit.c5.stres)
normal-z = 0.416, p-value = 0.6774
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
 0.0372143
```

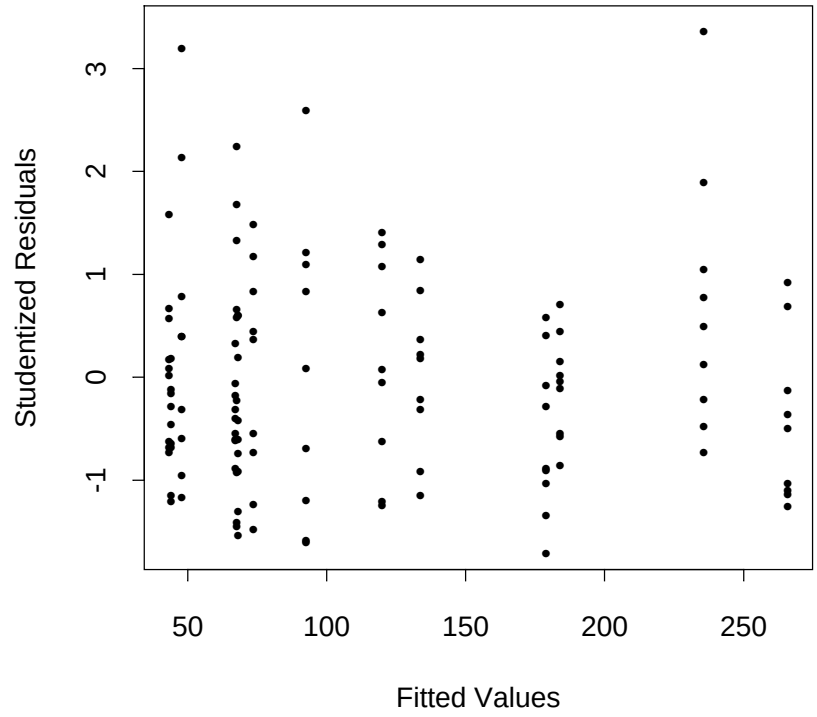

Figure C.6 Studentized residuals versus fitted values

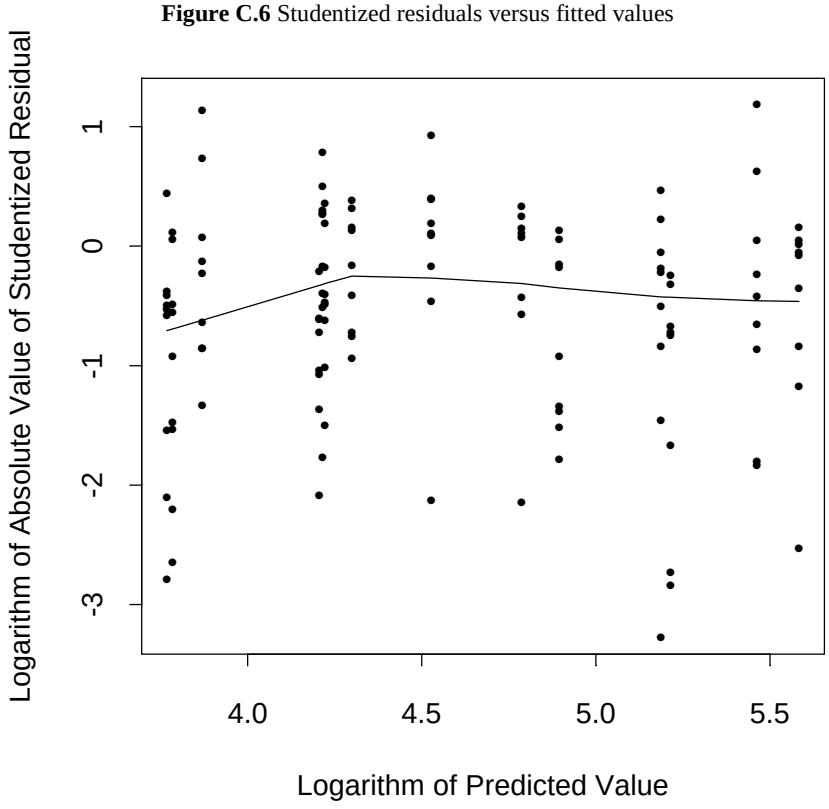

Figure C.7 Scatterplot with lowess smoothing of logarithms of absolute values of studentized residuals versus logarithms of predicted values from a generlaized least squares fit (smallest values of the absolute residuals deleted).

```
> nqcor(fit.c5.stres)
[1] 126.0000000   0.9773788
```

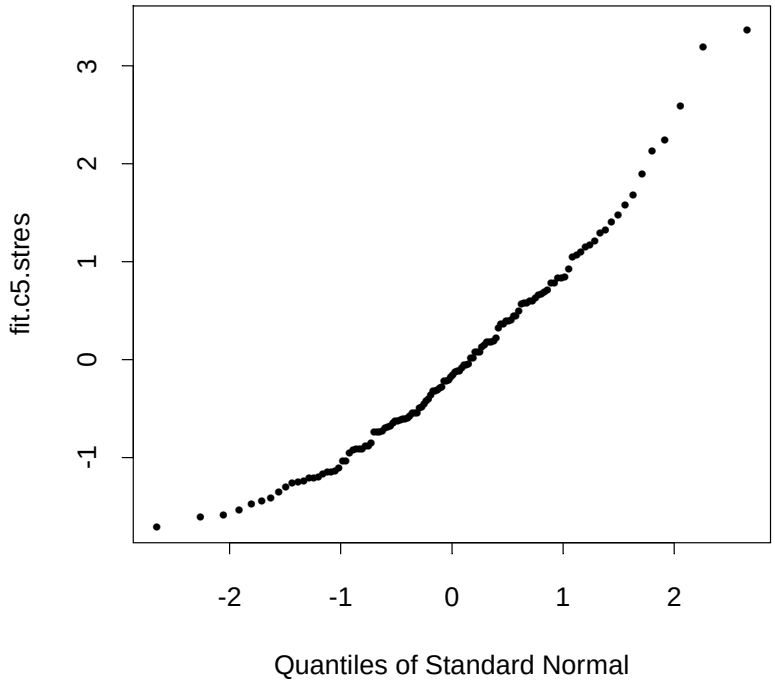

Figure C.8 Normal probability plot of studentized residuals from generalized least squares fit.

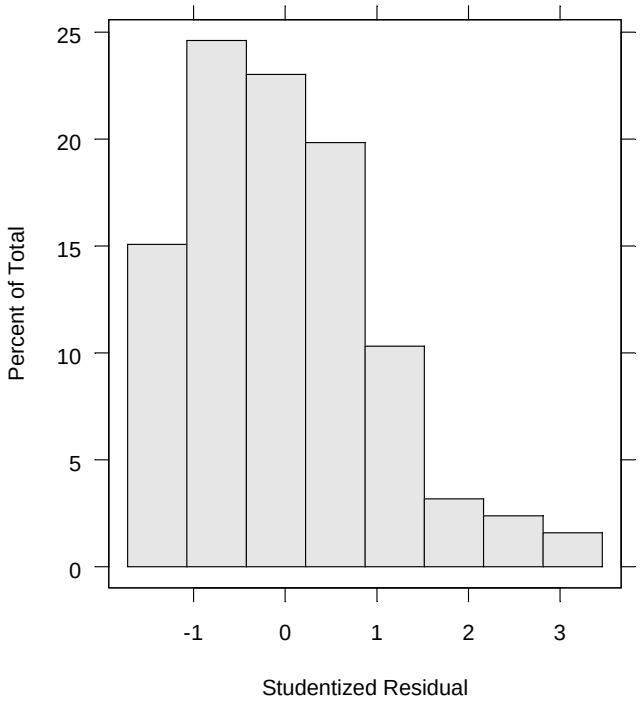

Figure C.9 Histogram of studentized residuals from generalized least squares fit.

C.7 Log-Transform of response and estimation function

```
> der.c0l<-deriv(
+       ~log(b0 + b4 * Treat + (b1 + b5 * Treat)/(1 + exp((b2 + b6 * Treat)
+       * (b3 + b7*Treat -  log(Dose))))),
+       c("b0","b1","b2","b3","b4","b5","b6","b7"),
+       function(b0,b1,b2,b3,b4,b5,b6,b7,Dose,Treat) NULL)


>  fit.c0l<-
nls(log(Respons)~der.c0l(b0,b1,b2,b3,b4,b5,b6,b7,Dose,Treat),Metrazol.data,
+       start=list(b0=43,b1=241,b2=1.22,b3=3.41,b4=24.5,b5=-26.5,b6=0.51,b7=-
1.15),trace=T)
7.06812 : 43 241 1.22 3.41 24.5 -26.5 0.51 -1.15
6.96626 : 42.5489 241.297 1.24915 3.43505 22.9437 -35.5729 0.476289 -1.176
6.96613 : 42.6004 236.106 1.26469 3.39943 22.9271 -30.3362 0.462622 -1.14141
6.96613 : 42.6116 235.737 1.26693 3.39649 22.9125 -29.8904 0.459214 -1.13786
> summary(fit.c0l)

Formula: log(Respons) ~ der.c0l(b0, b1, b2, b3, b4, b5, b6, b7, Dose, Treat)

Parameters:
         Value Std. Error    t value
b0   42.611600   2.633620  16.179800
b1  235.737000 140.801000   1.674260
b2    1.266930   0.405907   3.121230
b3    3.396490   0.936507   3.626770
b4   22.912500   4.585670   4.996550
b5  -29.890400 145.164000  -0.205908
b6    0.459214   0.640777   0.716651
b7   -1.137860   0.974050  -1.168170

Residual standard error: 0.242971 on 118 degrees of freedom

Correlation of Parameter Estimates:
       b0     b1     b2     b3     b4     b5     b6
b1 -0.448
b2  0.573 -0.913
b3 -0.434  0.991 -0.938
b4 -0.574  0.257 -0.329  0.250
b5  0.435 -0.970  0.886 -0.961 -0.328
b6 -0.363  0.578 -0.633  0.594  0.511 -0.711
b7  0.418 -0.953  0.902 -0.961 -0.284  0.985 -0.737
```

C.8 Assessment of double log-model

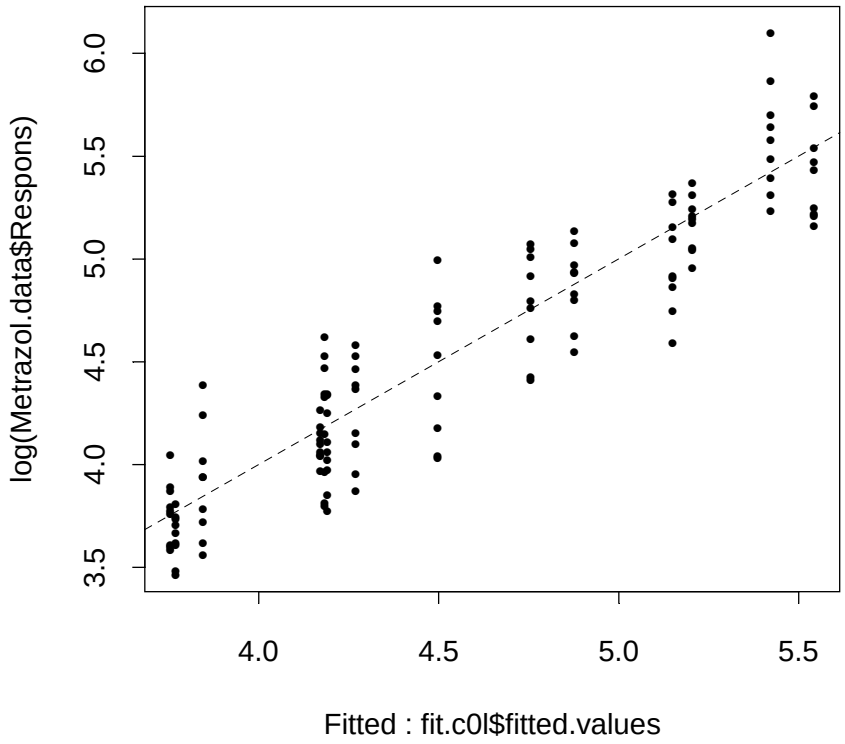

Figure C.10 Observed response versus values predicted from a model with logarithms of both response and estimation function.

```
> pars<-fit.c0l$parameters
> fit.c0l$GLSweights<-1
>  fit.c0l.stres<-
nls.studres(der.c0l(pars[1],pars[2],pars[3],pars[4],pars[5],pars[6],pars[7],pars[8],
+       Metrazol.data$Dose,Metrazol.data$Treat),fit.c0l)
> cor.test(fit.c0l$fitted.value,abs(fit.c0l.stres),method="spearman")

        Spearman's rank correlation

data:  fit.c0l$fitted.value and abs(fit.c0l.stres)
normal-z = 1.5223, p-value = 0.1279
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
 0.1361638
>  x<-log(fit.c0l$fitted.values)
>  y<-log(abs(fit.c0l.stres))
>  quantile(y,probs=c(0.01,0.99))
        1%       99%
 -3.863763 0.8580842
>  plot(x[y>-4],y[y>-4],xlab="Logarithm of Predicted Value",ylab="Logarithm of
Absolute Studentized Residual")
> lines(lowess(x,y))
```

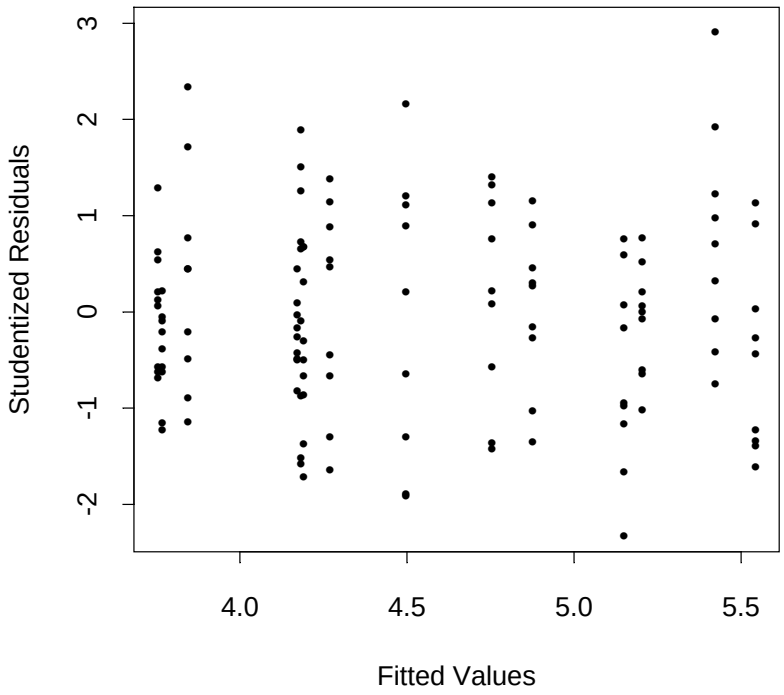

Figure C.11 Studentized residuals versus fitted values

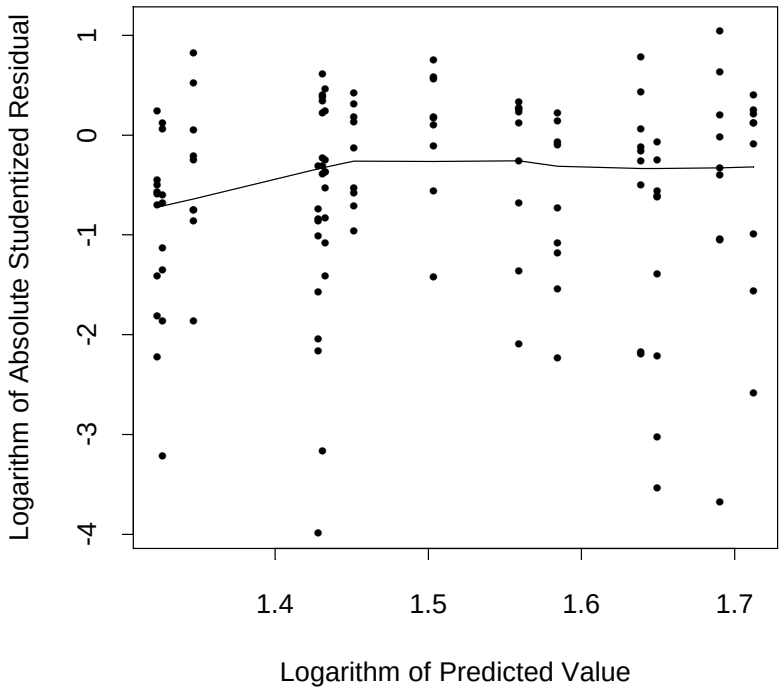

Figure C.12 Scatterplot with lowess smoothing of logarithms of absolute values of studentized residuals versus logarithms of predicted values from a fit using logarithms of both response and estimation function

```
> nqcor(fit.c0l.stres)
[1] 126.0000000   0.9963734
```

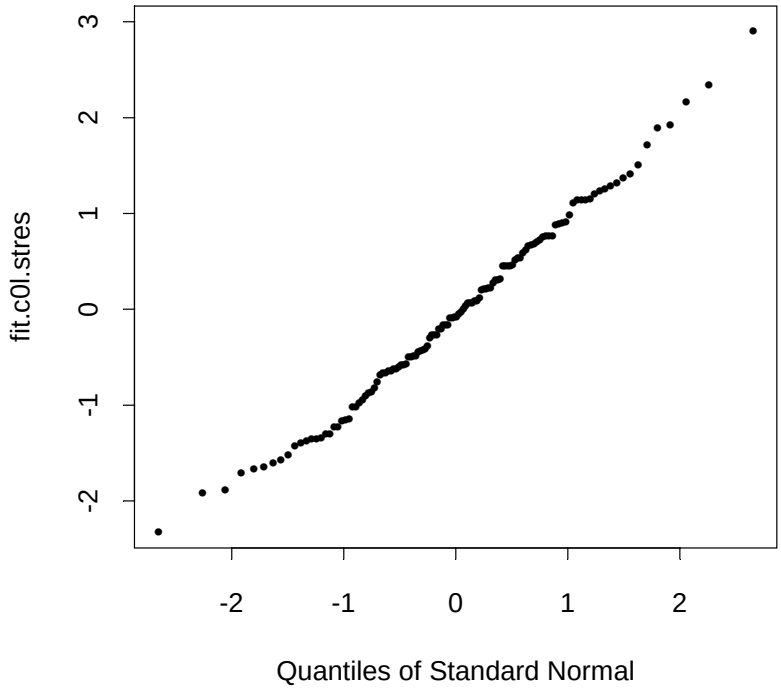

Figure C.13 Normal probability plot of studentized residuals.

C.9 Backward elimination of parameters from model (3.10)

Table C.3 Summary of backward elimination of parameters from model (3.10)

Step	Parameter	MSE	F	Prob > F
0	Model: all parameters MSE = 0.242971 with 118 d.f.			
	β_4	0.260199	9.44	0.003
	β_5	0.242004	0.53	0.470
	β_6	0.242518	0.78	0.380
	β_7	0.245500	2.24	0.137
1	Model: β_5 removed MSE = 0.242004 with 119 d.f.			
	β_4	0.261341	10.59	0.001
	β_6	0.242025	1.01	0.317
	β_7	0.253563	6.73	0.011
2	Model: β_6 removed MSE = 0.242025 with 120 d.f.			
	β_4	0.268324	14.15	<0.001
	β_7	0.271796	15.88	<0.001

C.10 Assessment of reduced model

```
> summary(fit.b5.b6)

Formula: log(Respons) ~ log(b0 + b4 * Treat + b1/(1 + exp(b2 * (b3 + b7 * Treat -
log(Dose)))))

Parameters:
        Value Std. Error   t value
b0   43.44170   2.265110  19.17860
b1  217.54800  37.292500   5.83355
b2    1.44230   0.244708   5.89397
b3    3.23096   0.285875  11.30200
b4   20.66980   3.988670   5.18212
b7   -0.88156   0.158636  -5.55712

Residual standard error: 0.242025 on 120 degrees of freedom

Correlation of Parameter Estimates:
         b0       b1       b2       b3       b4
b1 -0.26500
b2  0.39200 -0.80500
b3 -0.15100  0.93000 -0.84800
b4 -0.43500 -0.09550  0.10900 -0.18100
b7 -0.20100  0.00866  0.12100 -0.22100  0.51400
> exp(3.23096)
[1] 25.30394
> exp(-0.88156)
 [1] 0.4141364
```

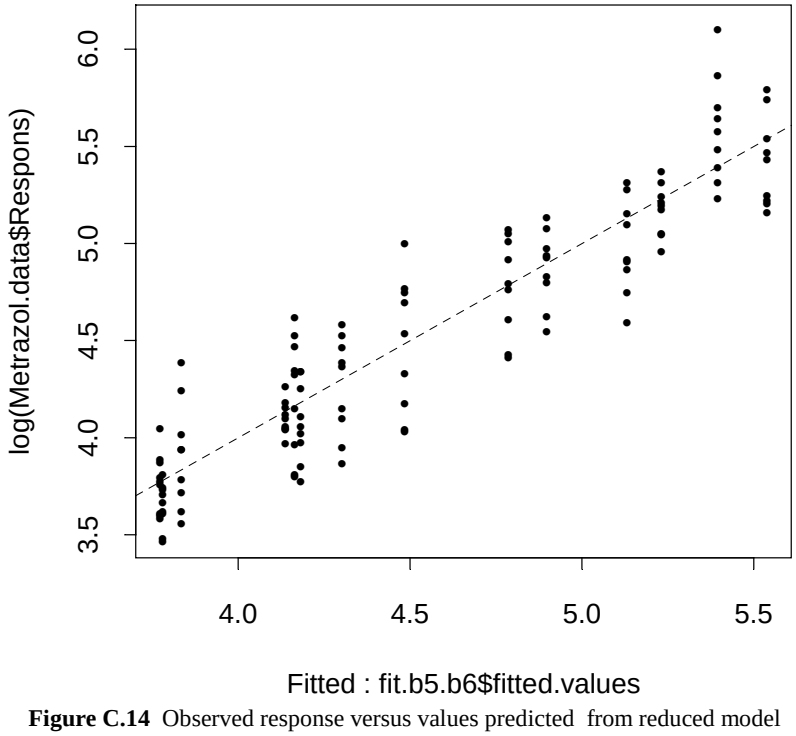

Figure C.14 Observed response versus values predicted from reduced model

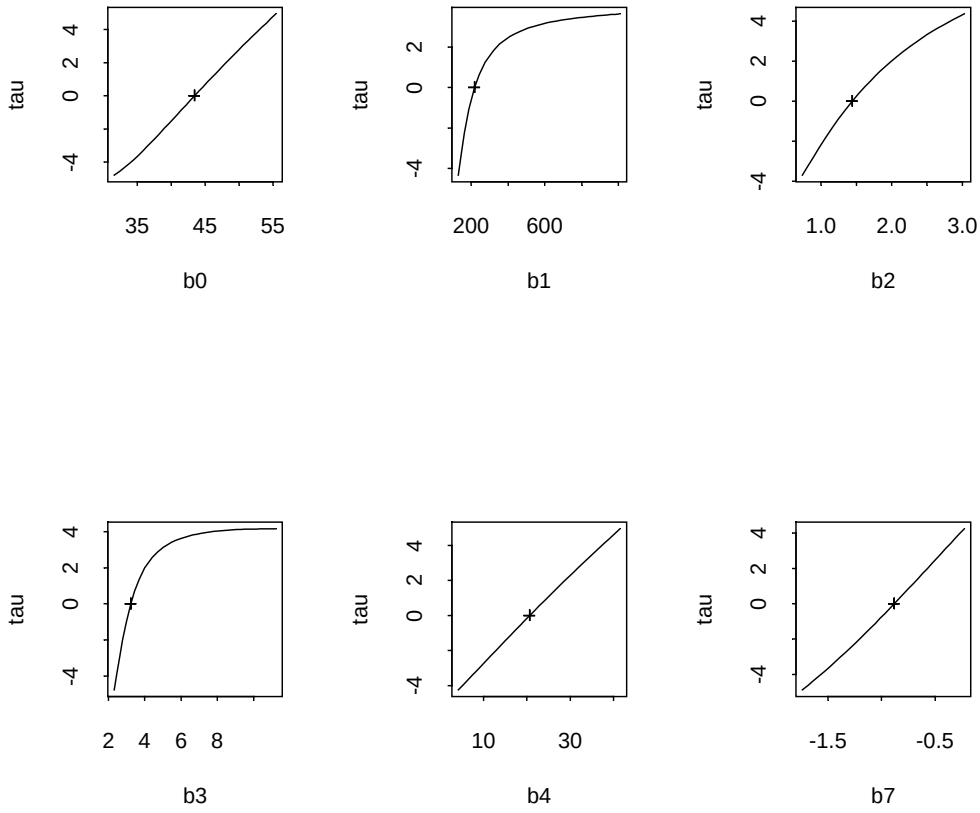

Figure C.15 Profile *t* plots for final model

```
> plist<-enlist(coef(fit.b5.b6))
> der.c2<-
+   deriv3(~log(b0 + b4 * Treat + b1/(1 + exp(b2 * (b3 + b7 * Treat - log(Dose))))),
+  ,c("b0","b1","b2","b3","b4","b7")
+  ,function(b0,b1,b2,b3,b4,b7,Treat,Dose) NULL)
fit.c.he<-nls(log(Respons)~der.c2(b0,b1,b2,b3,b4,b7,Treat,Dose),
+      start=plist,data=Metrazol.data)
> rms.curv(fit.c.he)
Parameter effects: c^theta x sqrt(F) = 1.1444
        Intrinsic: c^iota  x sqrt(F) = 0.2141
```

- Profile likelihood confidence intervals

```
> Conf.int(fit.b5.b6,"b0")
[1] 38.96894 48.04859
> Conf.int(fit.b5.b6,"b1")
[1] 167.4105 336.4079
> Conf.int(fit.b5.b6,"b2")
[1] 1.036254 1.990565
> Conf.int(fit.b5.b6,"b3")
[1] 2.789632 3.999388
> Conf.int(fit.b5.b6,"b4")
[1] 12.90833 28.70821
> Conf.int(fit.b5.b6,"b7")
[1] -1.2036338 -0.5759102
```

- Wald-type confidence intervals

```
> tval<-qt(1-0.5*0.05,summary(fit.b5.b6)$df[2])
> est<-summary(fit.b5.b6)$parameters
> conf<-cbind(est[,1]-est[,2]*tval,est[,1]+est[,2]*tval)
> conf
           [,1]         [,2]
b0   38.9569148   47.9264333
b1  143.7109444  291.3840653
b2    0.9577959    1.9268055
b3    2.6649465    3.7969729
b4   12.7724950   28.5670866
b7   -1.1956484   -0.5674712
```

C.11 Fitting the reduced model with generalized least squares

```
> fit.fingls<-nls.genls(~logistic6p(Respons,Treat,Dose,b0,b1,b2,b3,b4,b7,w),
+              Metrazol.data,start=list(b0=43.8,b1=195,b2=1.42,b3=3.1,b4=29.4,
+              b7=-0.84),"Respons",nriter=5)
> summary(fit.fingls)

Formula:  ~ logistic6p(Respons, Treat, Dose, b0, b1, b2, b3, b4, b7, w)

Parameters:
         Value Std. Error   t value
b0   44.107000   2.199000  20.05770
b1  229.825000  46.394900   4.95366
b2    1.403510   0.250191   5.60976
b3    3.269260   0.334762   9.76593
b4   21.925200   4.039490   5.42771
b7   -0.885422   0.171179  -5.17250

Residual standard error: 0.133706 on 120 degrees of freedom

Correlation of Parameter Estimates:
        b0      b1      b2      b3      b4
b1 -0.2770
b2  0.4130 -0.8030
b3 -0.1950  0.9390 -0.8630
b4 -0.3980 -0.0864  0.1150 -0.1630
b7 -0.1730  0.0209  0.1220 -0.1900  0.4990
> fit.fingls$variance.par
    theta
 1.138252
> exp(3.26926)
[1] 26.29188
> exp(-0.885422)
[1] 0.41254
```

- Wald confidence intervals

```
> tval<-qt(1-0.5*0.05,summary(fit.fingls)$df[2])
>  est<-summary(fit.fingls)$parameters
> conf<-cbind(est[,1]-est[,2]*tval,est[,1]+est[,2]*tval)
> conf
           [,1]        [,2]
b0   39.7531084   48.460850
b1  137.9658966  321.683384
b2    0.9081502    1.898872
b3    2.6064578    3.932069
b4   13.9272772   29.923088
b7   -1.2243446   -0.546500
```

- Profile likelihood confidence intervals

```
> Metrazol.wdata<-Metrazol.data
> Metrazol.wdata$w<-fit.fingls$GLSweights
> fit.fingls2<-nls(Respons*sqrt(w)~
+              (b0+b4*Treat+b1/(1+exp(b2*(b3+b7*Treatlog(Dose)))))*sqrt(w),
+              data=Metrazol.wdata,
+              start=list(b0=44,b1=230,b2=1.4,b3=3.3,b4=22,b7=-0.9))
>  Conf.int(fit.fingls2,"b0")
[1] 39.57887 48.34845
>  Conf.int(fit.fingls2,"b1")
[1] 167.5209 384.4258
>  Conf.int(fit.fingls2,"b2")
[1] 0.9914119 1.9853532
>  Conf.int(fit.fingls2,"b3")
[1] 2.756266 4.191927
>  Conf.int(fit.fingls2,"b4")
[1] 13.85825 29.86708
>  Conf.int(fit.fingls2,"b7")
[1] -1.2302504 -0.5559937
> exp( Conf.int(fit.fingls2,"b3"))
[1] 15.74095 66.15017
> exp( Conf.int(fit.fingls2,"b7"))
[1] 0.2922194 0.5735021
```

Appendix D
Inhibition of thromboxane A_2 formation in blood samples

D.1 Data (source: J. Beetens, et al. ,1996).

Table D.1 Effect of ridogrel on the formation of TXB_2 (ng/ml) after stimulation with calcimycin (25 µM) in blood samples of volunteers.

	Ridogrel concentration (M)									
Volunteer	0	10^{-9}	3×10^{-9}	10^{-8}	3×10^{-8}	10^{-7}	3×10^{-7}	10^{-6}	3×10^{-6}	10^{-5}
1	158.3	186.1	179.1	167.7	101.3	19.48	10.56	4.17	1.99	1.07
2	95.46	95.12	90.91	70.29	38.56	6.42	4.53	2.25	0.91	0.46
3	158.1	158.0	131.1	134.6	72.28	14.81	5.82	2.17	1.16	0.81
4	29.1	26.75	22.72	18.06	11.80	4.46	1.11	0.47	0.74	0.70
5	67.36	62.16	65.53	44.81	35.79	7.88	3.26	2.41	1.22	0.82
6	43.88	33.85	34.08	26.64	14.18	5.21	2.14	0.95	0.54	0.27
Mean	92.0	93.7	87.2	77.0	45.7	9.7	4.6	2.1	1.1	0.7
SD	56.0	65.9	59.7	61.1	34.9	6.1	3.4	1.3	0.5	0.3

D.2 Derivatives and OLS fit on pooled data

```
>  txb2.fun<-deriv(~log(b0+b1/(1+exp(b2*(b3-log(Conc))))),
+       c("b0","b1","b2","b3"),function(b0,b1,b2,b3,Conc) NULL)
> txb2.fun
function(b0, b1, b2, b3, Conc)
{
        .expr2 <- b3 - (log(Conc))
        .expr4 <- exp((b2 * .expr2))
        .expr5 <- 1 + .expr4
        .expr7 <- b0 + (b1/.expr5)
        .expr14 <- .expr5^2
        .value <- log(.expr7)
        .grad <- array(0, c(length(.value), 4), list(NULL, c("b0", "b1", "b2", "b3")))
        .grad[, "b0"] <- 1/.expr7
        .grad[, "b1"] <- (1/.expr5)/.expr7
        .grad[, "b2"] <-  - (((b1 * (.expr4 * .expr2))/.expr14)/.expr7)
        .grad[, "b3"] <-  - (((b1 * (.expr4 * b2))/.expr14)/.expr7)
        attr(.value, "gradient") <- .grad
        .value
}
```

```
> fit.pool<-nls(log(Respons)~txb2.fun(b0,b1,b2,b3,Conc),data=txb2.data,
+       start=list(b0=0,b1=100,b2=-0.14,b3=-16),trace=T)
325.946 : 0 100 -0.14 -16
325.288 : 9.30733 82.9205 -0.168594 -16.3132
323.876 : 18.5956 64.8966 -0.216831 -16.5572
318.884 : 24.9345 50.8603 -0.29381 -16.6513
304.148 : 26.3343 44.2918 -0.408784 -16.636
269.345 : 22.0726 46.2 -0.562327 -16.6383
199.122 : 12.8596 55.5962 -0.727188 -16.7935
87.7449 : 3.32444 67.6864 -0.862272 -17.1817
30.0002 : 0.618198 75.0934 -0.985697 -17.7075
27.2921 : 0.660118 80.0147 -1.08721 -17.9475
27.2329 : 0.659403 78.2985 -1.13776 -17.8142
27.2312 : 0.66082 78.7241 -1.13733 -17.8306
27.2312 : 0.661882 78.6587 -1.13881 -17.8272
> summary(fit.pool)

Formula: log(Respons) ~ txb2.fun(b0, b1, b2, b3, Conc)

Parameters:
        Value Std. Error   t value
b0   0.661882   0.200621   3.29917
b1  78.658700  14.635600   5.37448
b2  -1.138810   0.190851  -5.96699
b3 -17.827200   0.481173 -37.04960

Residual standard error: 0.697331 on 56 degrees of freedom

Correlation of Parameter Estimates:
       b0     b1     b2
b1 -0.218
b2 -0.600  0.464
b3  0.353 -0.747 -0.829
>
```

D.3 Naïve standard two-stage approach (means of individual estimates)

- Individual estimates

```
>  fit.indiv<-nls(log(Respons)~
+   log(b0[Volnr]+b1[Volnr]/(1+exp(b2[Volnr]*(b3[Volnr]-log(Conc)))))
+   ,data=txb2.data,start=
+   list(b0=rep(1,6),b1=c(158,95,158,29,67,44),b2=rep(-1.4,6),b3=rep(-17.8,6)))
> summary(fit.indiv)

Formula: log(Respons) ~ log(b0[Volnr] + b1[Volnr]/(1 + exp(b2[Volnr] * (b3[Volnr] -
log(Conc)))))

Parameters:
          Value Std. Error   t value
b01    1.002240  0.3153640   3.17805
b02    0.365048  0.1535180   2.37788
b03    0.832865  0.1881520   4.42656
b04    0.601420  0.0903884   6.65373
b05    0.917765  0.2092250   4.38649
b06    0.208765  0.0904550   2.30794
b11  193.201000 27.9537000   6.91149
b12  109.710000 18.5226000   5.92301
b13  161.040000 22.4445000   7.17505
b14   24.103900  3.1421000   7.67126
b15   68.565300 10.4129000   6.58468
b16   41.968500  7.0237600   5.97522
b21   -1.114040  0.1363620  -8.16969
b22   -0.963685  0.1098580  -8.77212
b23   -1.280260  0.1506200  -8.49993
b24   -1.647610  0.3094780  -5.32385
b25   -1.115970  0.1700120  -6.56404
b26   -0.934646  0.1199180  -7.79407
b31  -17.675500  0.3774480 -46.82890
b32  -18.364700  0.4521570 -40.61570
b33  -17.718400  0.3263690 -54.28960
b34  -17.278500  0.2997660 -57.64010
b35  -17.789100  0.4095010 -43.44100
b36  -18.150700  0.4771330 -38.04130

Residual standard error: 0.227882 on 36 degrees of freedom
```

- Means, standard errors

```
> fit.indiv.coef<-matrix(coef(fit.indiv),nrow=6,ncol=4)
> fit.indiv.coef
          [,1]      [,2]       [,3]      [,4]
[1,] 1.0022408 193.20131 -1.1140373 -17.67547
[2,] 0.3650475 109.70970 -0.9636852 -18.36465
[3,] 0.8328651 161.04025 -1.2802609 -17.71844
[4,] 0.6014203  24.10390 -1.6476134 -17.27854
[5,] 0.9177652  68.56535 -1.1159682 -17.78912
[6,] 0.2087648  41.96851 -0.9346461 -18.15074
> apply(fit.indiv.coef,2,mean)
[1]   0.6546839  99.7648374  -1.1760352 -17.8294926
> sqrt(apply(fit.indiv.coef,2,var)/6)
[1]  0.1300418 27.4604884  0.1070922  0.1560800
```

D.4 Global two-stage method using EM-algorithm

```
> glob.two.stage
function(indivfit, nriter = 2, npar = 4, m = 6)
{
# Global two-stage method for parameter estimation using EM-algorithm
        betas <- matrix(coef(indivfit), nrow = m, ncol = npar)
        ii <- c(0, m, 2 * m, 3 * m)   # change this if npar <> 4
        cv <- summary(indivfit)$cov.unscaled
        betac <- betas
        beta2 <- betac
        EBeta <- apply(betas, 2, mean)
        Dc <- cov.wt(betas)$cov
        for(ms in 1:nriter) {
                D1 <- 0
                iDc <- solve(Dc)
                iSigma <- 0
                for(i in 1:m) {
                        icv <- solve(cv[ii + 1, ii + 1])
                        p1 <- solve(icv + iDc)
                        D1 <- D1 + p1
                        p2 <- (icv %*% betas[i,  ] + iDc %*% EBeta)
                        betac[i,  ] <- p1 %*% p2
                        iSigma <- iSigma + solve(cv[ii + 1, ii + 1] + Dc)
                }
                W <- solve(m * iDc)%*%iDc
                for(i in 1:m) {
                        beta2[i,  ] <- W %*% betac[i,  ]
                }
                Ebeta <- apply(beta2, 2, sum)
                Ebetam <- matrix(Ebeta, nrow = m, ncol = npar, byrow = T)
                cp <- (t(betac - Ebetam)) %*% (betac - Ebetam)
                Dc <- (D1 + cp)/m
        }
        Sigma <- solve(iSigma)
        list(betac, Ebeta, Sigma)
}
gg2<-glob.two.stage(fit.indiv,nriter=200)
> gg2
[[1]]:
          [,1]      [,2]      [,3]      [,4]
[1,] 0.7192977 121.39546 -1.136932 -17.84893
[2,] 0.6488133  99.08588 -1.177140 -17.83094
[3,] 0.6785195 108.20860 -1.162569 -17.83587
[4,] 0.5923676  79.87084 -1.216676 -17.80681
[5,] 0.6569446 100.81679 -1.174048 -17.83056
[6,] 0.6308223  93.15691 -1.187224 -17.82536
[[2]]:
[1]    0.6544608  100.4224150   -1.1757648  -17.8297434
[[3]]:
            b01          b11         b21         b31
b01  0.32155200    -5.881263 -0.08913448   0.1510829
b11 -5.88126303  2624.753599  5.72819235 -25.2139068
b21 -0.08913448     5.728192  0.06042538  -0.1387243
b31  0.15108294   -25.213907 -0.13872434   0.4591572
# Standard errors
> sqrt(diag(gg2[[4]]/6))
[1]  0.2314995 20.9155190  0.1003539  0.2766337
```

D.5 Lindstrom-Bates method (NLME-package)

- Approximate likelihood (pseudo-likelihood):

```
> txb2.fit0<-
nlme(object=log(Respons)~txb2.fun(b0,b1,b2,b3,Conc),fixed=list(b0~.,b1~.,b2~.,b3~.),
+   random=list(b0~.,b1~.,b2~.,b3~.),cluster=~Volnr,data=txb2.data,
+   start=list(fixed=c(1,70,-1.6,-17.4)))
> summary(txb2.fit0)
Call:
  Model: log(Respons) ~ txb2.fun(b0, b1, b2, b3, Conc)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
 Random: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
Cluster:  ~ Volnr
   Data: txb2.data
Estimation Method: ML
Convergence at iteration: 12
Approximate Loglikelihood: -12.28775
AIC: 54.5755
BIC: 85.99067

Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
       b0       b1        b2        b3
 0.2607517 60.06326 0.1093486 0.2085867
 Correlation of Random Effects
            b0          b1          b2
b1  0.67424858
b2 -0.66987193 -0.03420313
b3  0.88248127  0.26348922 -0.91247338
 Cluster Residual Variance: 0.04789817
Fixed Effects Estimate(s):
          Value Approx. Std.Error  z ratio(C)
b0   0.6450992        0.12222585    5.277928
b1  96.9215248       25.17463594    3.849967
b2  -1.1274409        0.06916772  -16.300102
b3 -17.8396747        0.16587704 -107.547585
 Conditional Correlation(s) of Fixed Effects Estimates
              b0            b1            b2
b1  0.5592912374
b2 -0.5734640080  0.0399294178
b3  0.5001017117  0.0004114797 -0.8252438148
Random Effects (Conditional Modes):
           b0          b1           b2          b3
1  0.32379864  99.3167388  0.003383139  0.11061266
2 -0.16983751   0.1156571  0.078344215 -0.17232642
3  0.17840324  51.5342620 -0.059067242  0.09908762
4 -0.09568079 -70.4351783 -0.082076770  0.06729817
5  0.13536549 -26.5399460 -0.070357120  0.17708376
6 -0.37204908 -53.9915335  0.129773777 -0.28175578
Standardized Population-Average Residuals:
       Min         Q1        Med        Q3      Max
 -2.924744 -0.3792375 0.05231151 0.6884802 2.242268
Number of Observations: 60
Number of Clusters: 6
```

- Assessment of fit

```
> plot(txb2.fit0,option="s")
```

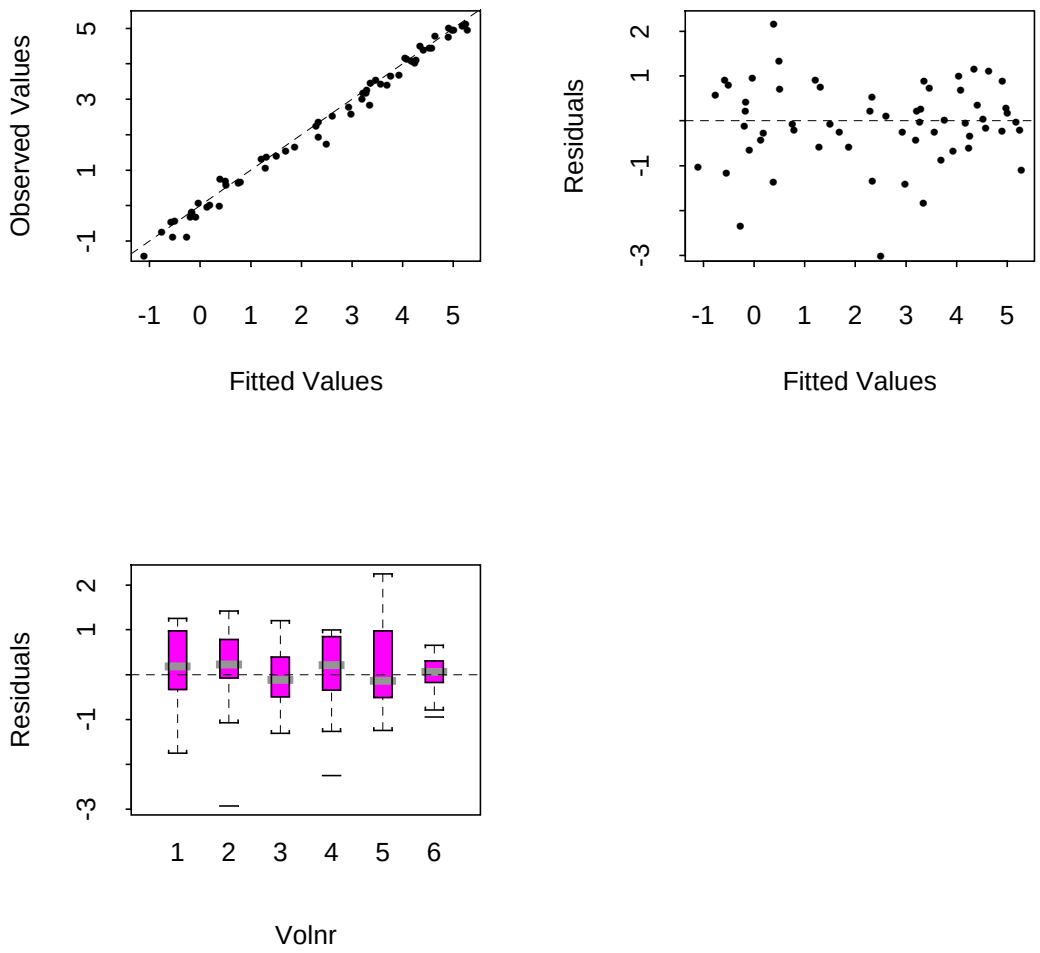

Figure D.1 Plots of standardized residuals and fitted values of the model with random effects for all parameters.

D.6 Backward elimination of random effects

```
> anova(txb2.fit.b0f,txb2.fit0)
Response: log(Respons)
txb2.fit.b0f
 fixed: b0, b1, b2, b3
 random: b1, b2, b3
 block: list(1:3)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
txb2.fit0
 fixed: b0, b1, b2, b3
 random: b0, b1, b2, b3
 block: list(1:4)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
            Model Df    AIC    BIC  Loglik    Test Lik.Ratio  P value
txb2.fit.b0f     1 11 55.207 78.245 -16.603
   txb2.fit0     2 15 54.575 85.991 -12.288 1 vs. 2    8.6314 0.071004
```

Table D.2 Summary of backward elimination of random effect parameters from model (4.15)

Step	Parameter as fixed	Log likelihood	Likelihood Ratio	P-value
0	Model: all parameters random, log-likelihood = -12.29, # parameters reference model = 15, # parameters reduced model = 11			
	β_0	-16.60	8.63	0.071
	β_1	-38.19	51.8	<0.001
	β_2	-13.15	1.73	0.785
	β_3	-13.07	1.56	0.817
1	Model: random effects for β_3 removed, log-likelihood = -13.07 # parameters reference model = 11, # parameters reduced model = 8			
	β_0	-16.68	7.22	0.065
	β_1	-61.44	96.7	<0.001
	β_2	-13.33	0.53	0.911
2	Model: random effects for β_2 removed, log-likelihood = -13.33 # parameters reference model = 8, # parameters reduced model = 6			
	β_0	-18.31	14.15	0.007
	β_1	-60.99	95.32	<0.001

D.7 Fit of reduced model with only random effects for β_0 and β_1

- Approximate likelihood

```
> summary(txb2.fit.b3.b2f)
Call:
  Model: log(Respons) ~ txb2.fun(b0, b1, b2, b3, Conc)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
 Random: list(b0 ~ ., b1 ~ .)
Cluster:  ~ Volnr
   Data: txb2.data

Estimation Method: ML
Convergence at iteration: 4
Approximate Loglikelihood: -13.33367
AIC: 42.66734
BIC: 59.4221

Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
       b0      b1
 0.2276653 62.6181
 Correlation of Random Effects
         b0
b1 0.7982952

 Cluster Residual Variance: 0.0529698

Fixed Effects Estimate(s):
          Value Approx. Std.Error  z ratio(C)
b0   0.6481998        0.11207793    5.783474
b1  97.6012493       26.25600585    3.717292
b2  -1.1261566        0.05764447  -19.536249
b3 -17.8223682        0.15333701 -116.230048

 Conditional Correlation(s) of Fixed Effects Estimates
            b0          b1          b2
b1  0.63527831
b2 -0.30658885  0.08117341
b3  0.15558194 -0.15330456 -0.81510665

Random Effects (Conditional Modes):
           b0          b1
1  0.35282390  106.189909
2 -0.09294753   -4.930827
3  0.08966928   51.978181
4 -0.14962588  -71.211937
5  0.09855408  -24.218375
6 -0.29847385  -57.806951


Standardized Population-Average Residuals:
       Min         Q1        Med        Q3      Max
 -2.864292 -0.3868895 0.09251997 0.5650977 1.936793

Number of Observations: 60
Number of Clusters: 6
```

- Restricted maximum likelihood

```
> txb2.fit.b3.b2fR<-
nlme(object=log(Respons)~txb2.fun(b0,b1,b2,b3,Conc),est.method="RML",
+    fixed=list(b0~.,b1~.,b2~.,b3~.),random=list(b0~.,b1~.),cluster=~Volnr,
+    data=txb2.data,start=list(fixed=c(1,70,-1.6,-17.4)))
> summary(txb2.fit.b3.b2fR)
Call:
  Model: log(Respons) ~ txb2.fun(b0, b1, b2, b3, Conc)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
 Random: list(b0 ~ ., b1 ~ .)
Cluster:  ~ Volnr
   Data: txb2.data

Estimation Method: RML
Convergence at iteration: 4
Approximate Restricted Loglikelihood: -14.11703
Restricted AIC: 44.23407
Restricted BIC: 60.98882
Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
       b0       b1
 0.2562426 69.04356
 Correlation of Random Effects
         b0
b1 0.772191

 Cluster Residual Variance: 0.05504093

Fixed Effects Estimate(s):
         Value Approx. Std.Error  z ratio(C)
b0   0.650324        0.12337235    5.271230
b1  97.959738       28.85659358    3.394709
b2  -1.125769        0.05906085  -19.061166
b3 -17.826531        0.15710793 -113.466782

 Conditional Correlation(s) of Fixed Effects Estimates
            b0          b1          b2
b1  0.62916036
b2 -0.29442982  0.07779173
b3  0.15330197 -0.14464248 -0.81676578

Random Effects (Conditional Modes):
           b0          b1
1  0.36024934  106.855902
2 -0.10444780   -4.834059
3  0.07692954   52.556356
4 -0.14701731  -71.585079
5  0.12155261  -25.019379
6 -0.30726638  -57.973741
Standardized Population-Average Residuals:
       Min         Q1        Med        Q3      Max
 -2.812325 -0.3743573 0.06016352 0.5986531 1.850926

Number of Observations: 60
Number of Clusters: 6
```

- Assessment of fit of reduced model with β_2 and β_3 fixed REML fitting

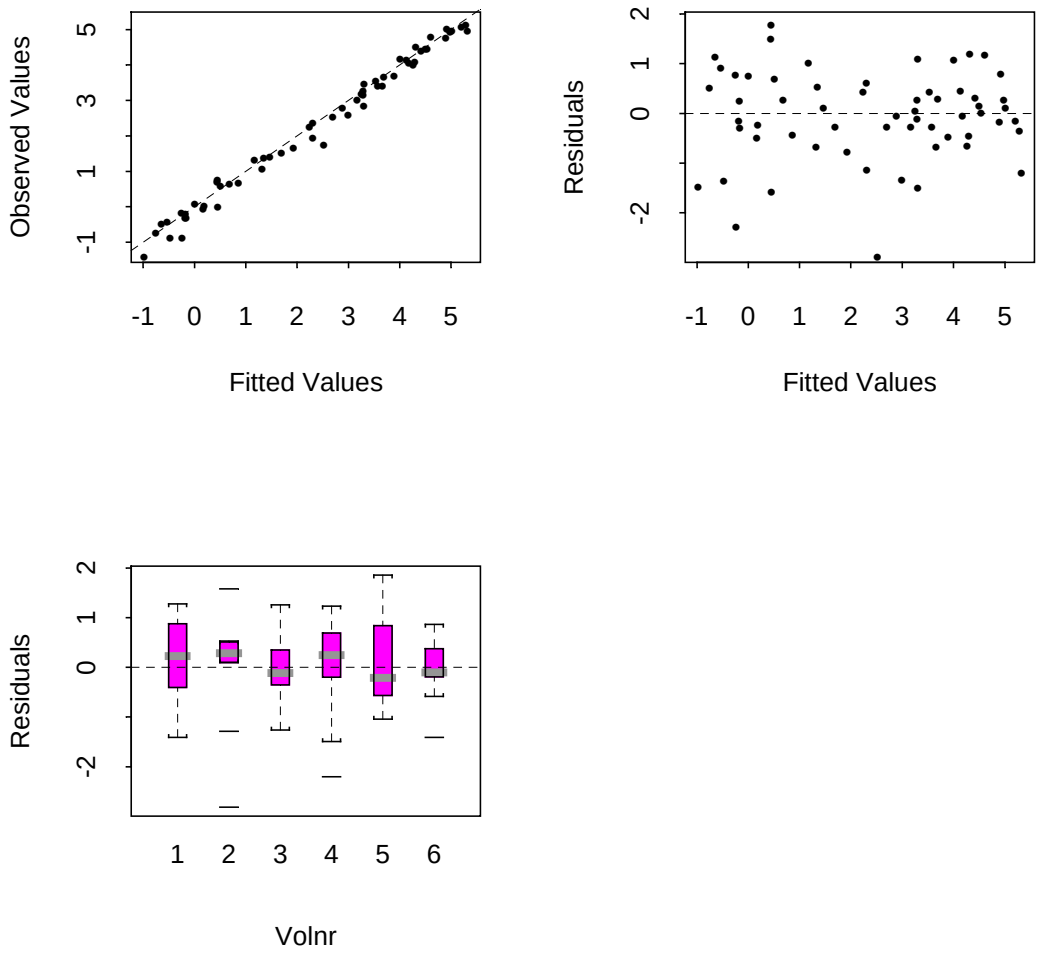

Figure D.2 Plots of standardized residuals and fitted values from REML-fit of the model with β_2 and β_3 kept fixed

D.8 SAS implementation using the NLINMIX macro

- Lindstrom-Bates conditional linearization with REML.

```
%nlinmix(data=mydat.txb,response=lresp,subject=volnr,options=skipnlin,
         model=%str(
         e2=b3-lconc;
         e4=exp((b2)*e2);
         e5=1+e4;
         e7=(b0+u0)+(b1+u1)/e5;
         e14=e5**2;
         pred=log(e7);
         ),
         derivs=%str(
         d_b0=1/e7;
         d_b1=(1/e5)/e7;
         d_b2=-((((b1+u1)*(e4*e2))/e14)/e7);
         d_b3=-((((b1+u1)*(e4*(b2)))/e14)/e7);
         d_u0=d_b0;
         d_u1=d_b1;
         ),
         parms=%str(b0=1 b1=70 b2=-1.6 b3=-17.4),
         random=u0 u1,
         type=un,
         expand=eblup,
         method=REML
);
```

```
                              The MIXED Procedure

                            Class Level Information

                          Class     Levels  Values

                          _SUBJECT       6  1 2 3 4 5 6


                                  Parameter Search

        COL1      COL2      COL3      COL4  Variance   REML_LL  -2REML_LL  Objective

      0.0656   13.6547  4767.271    0.0550    0.0550  -14.1171    28.2341   -74.6870


                           REML Estimation Iteration History

                  Iteration  Evaluations     Objective     Criterion

                          1            1  -74.68698307    0.00000000

                              Convergence criteria met.


                          Covariance Parameter Estimates (REML)

Cov Parm              Ratio      Estimate     Std Error       Z  Pr > |Z| Alpha   Lower  Upper

D_U0 UN(1,1)     1.19224550    0.06562178    0.05617948    1.17   0.2428  0.05  -0.0445     0.1757
     UN(2,1)   248.08455781   13.65469606   11.33791239    1.20   0.2285  0.05  -8.5672    35.8766
     UN(2,2)   86613.890601  4767.2711325  3170.5544835    1.50   0.1327  0.05 -1446.90 10981.44
Residual         1.00000000    0.05504049    0.01146128    4.80   0.0001  0.05   0.0326     0.0775
```

Model Fitting Information for _RESID

Description	Value
Observations	60.0000
Variance Estimate	0.0550
Standard Deviation Estimate	0.2346
REML Log Likelihood	-14.1171
Akaike's Information Criterion	-18.1171
Schwarz's Bayesian Criterion	-22.1678
-2 REML Log Likelihood	28.2341
PARMS Model LRT Chi-Square	0.0000
PARMS Model LRT DF	3.0000
PARMS Model LRT P-Value	1.0000

Solution for Fixed Effects

Parameter	Estimate	Std Error	DDF	T	Pr > |T|	Alpha	Lower	Upper
D_B0	0.65027492	0.12334633	46	5.27	0.0001	0.05	0.4020	0.8986
D_B1	97.96174153	28.85740063	46	3.39	0.0014	0.05	39.8748	156.0487
D_B2	-1.12574279	0.05905857	46	-19.06	0.0001	0.05	-1.2446	-1.0069
D_B3	-17.82658079	0.15711176	46	-113.5	0.0001	0.05	-18.1428	-17.5103

Solution for Random Effects

Parameter	Subject	Estimate	SE Pred	DDF	T	Pr > |T|
D_U0	_SUBJECT 1	0.36014572	0.15720346	46	2.29	0.0266
D_U1	_SUBJECT 1	106.85956671	31.90030921	46	3.35	0.0016
D_U0	_SUBJECT 2	-0.10443013	0.13511980	46	-0.77	0.4436
D_U1	_SUBJECT 2	-4.83445622	29.21980861	46	-0.17	0.8693
D_U0	_SUBJECT 3	0.07689601	0.14576188	46	0.53	0.6004
D_U1	_SUBJECT 3	52.55830881	30.35561064	46	1.73	0.0901
D_U0	_SUBJECT 4	-0.14696504	0.12885898	46	-1.14	0.2600
D_U1	_SUBJECT 4	-71.58689569	28.75324545	46	-2.49	0.0165
D_U0	_SUBJECT 5	0.12156952	0.14137579	46	0.86	0.3943
D_U1	_SUBJECT 5	-25.02104598	28.99425036	46	-0.86	0.3926
D_U0	_SUBJECT 6	-0.30721609	0.12505840	46	-2.46	0.0179
D_U1	_SUBJECT 6	-57.97547763	28.75271534	46	-2.02	0.0496

Tests of Fixed Effects

Source	NDF	DDF	Type III F	Pr > F
D_B0	1	46	27.79	0.0001
D_B1	1	46	11.52	0.0014
D_B2	1	46	363.34	0.0001
D_B3	1	46	12874.15	0.0001

- First-order linearization (Vonesh-Carter like)

```
%nlinmix(data=mydat.txb,response=lresp,subject=volnr,
         model=%str(
         e2=b3-lconc;
         e4=exp((b2)*e2);
         e5=1+e4;
         e7=(b0+u0)+(b1+u1)/e5;
         e14=e5**2;
         pred=log(e7);
         ),
         derivs=%str(
         d_b0=1/e7;
         d_b1=(1/e5)/e7;
         d_b2=-((((b1+u1)*(e4*e2))/e14)/e7);
         d_b3=-((((b1+u1)*(e4*(b2)))/e14)/e7);
         d_u0=d_b0;
         d_u1=d_b1;
         ),
         parms=%str(b0=1 b1=70 b2=-1.6 b3=-17.4),
         random=u0 u1,
         type=un,
         expand=zero,
         method=REML
);
run;
```

```
                                The MIXED Procedure

                              Class Level Information

                            Class     Levels  Values

                            _SUBJECT       6  1 2 3 4 5 6

                                   Parameter Search

        COL1      COL2      COL3      COL4  Variance   REML_LL  -2REML_LL  Objective

      0.0745   11.5785  3838.163    0.0535    0.0535  -12.9488    25.8976   -77.0235

                             REML Estimation Iteration History

                   Iteration  Evaluations     Objective     Criterion

                             1            1  -77.02346944    0.00000000

                                 Convergence criteria met.


                          Covariance Parameter Estimates (REML)

Cov Parm              Ratio      Estimate     Std Error       Z  Pr > |Z|  Alpha    Lower    Upper

D_U0 UN(1,1)     1.39274056    0.07445613    0.05680472    1.31    0.1899   0.05  -0.0369   0.1858
     UN(2,1)   216.58205912   11.57851040    9.77868446    1.18    0.2364   0.05  -7.5874  30.7444
     UN(2,2)   71794.834525  3838.1629651  2460.0030796    1.56    0.1187   0.05 -983.354 8659.678
Residual         1.00000000    0.05346015    0.01114721    4.80    0.0001   0.05   0.0316   0.0753
```

Model Fitting Information for _RESID

Description	Value
Observations	60.0000
Variance Estimate	0.0535
Standard Deviation Estimate	0.2312
REML Log Likelihood	-12.9488
Akaike's Information Criterion	-16.9488
Schwarz's Bayesian Criterion	-20.9995
-2 REML Log Likelihood	25.8976
PARMS Model LRT Chi-Square	0.0000
PARMS Model LRT DF	3.0000
PARMS Model LRT P-Value	1.0000

Solution for Fixed Effects

Parameter	Estimate	Std Error	DDF	T	Pr > |T|	Alpha	Lower	Upper
D_B0	0.66212524	0.12974759	46	5.10	0.0001	0.05	0.4010	0.9233
D_B1	78.66162212	25.75351148	46	3.05	0.0037	0.05	26.8225	130.5008
D_B2	-1.13894721	0.06329164	46	-18.00	0.0001	0.05	-1.2663	-1.0115
D_B3	-17.82727673	0.15952650	46	-111.8	0.0001	0.05	-18.1484	-17.5062

Solution for Random Effects

Parameter	Subject	Estimate	SE Pred	DDF	T	Pr > |T|
D_U0	_SUBJECT 1	0.34440600	0.14623540	46	2.36	0.0228
D_U1	_SUBJECT 1	76.83186401	26.07828894	46	2.95	0.0050
D_U0	_SUBJECT 2	-0.08545545	0.14623540	46	-0.58	0.5618
D_U1	_SUBJECT 2	13.41299057	26.07828894	46	0.51	0.6095
D_U0	_SUBJECT 3	0.07052931	0.14623540	46	0.48	0.6319
D_U1	_SUBJECT 3	53.65805001	26.07828894	46	2.06	0.0453
D_U0	_SUBJECT 4	-0.12035771	0.14623540	46	-0.82	0.4147
D_U1	_SUBJECT 4	-84.26838821	26.07828894	46	-3.23	0.0023
D_U0	_SUBJECT 5	0.17037875	0.14623540	46	1.17	0.2500
D_U1	_SUBJECT 5	-6.29086403	26.07828894	46	-0.24	0.8104
D_U0	_SUBJECT 6	-0.37950090	0.14623540	46	-2.60	0.0126
D_U1	_SUBJECT 6	-53.34365236	26.07828894	46	-2.05	0.0465

Tests of Fixed Effects

Source	NDF	DDF	Type III F	Pr > F
D_B0	1	46	26.04	0.0001
D_B1	1	46	9.33	0.0037
D_B2	1	46	323.83	0.0001
D_B3	1	46	12488.33	0.0001

D.9 SAS implementation using the MIXNLIN macro (Vonesh, 1995)

- Vonesh-Carter first-order linearization

```
%mixnlin(imllib=rmlib.storemod);
procm mixnlin data=mydat.txb method=reml expand=average outpred=results;
subject volnr;
vars 'lconc';
parms b1='1' b2='100' b3='-1.1' b4='-17.8';
design all='_class_';
model 'lresp=log(user_4)';
random 'b1 b2';
deriv der_b1='1/user_4'
      der_b2='(1/user_3)/user_4'
      der_b3='-(((b2*user_2*user_1)/user_5)/user_4)'
      der_b4='-(((b2*user_2*b3)/user_5)/user_4)';
function user_1='b4-lconc'
         user_2='exp(b3*user_1)'
         user_3= '1+user_2'
         user_4='b1+b2/user_3'
         user_5='user_3##2';
execute mixnlin;
run;
```

```
                            MACRO PROCEDURE MIXNLIN

                        VARIABLE AND CLASS LEVEL INFORMATION
    VARIABLE:  DEPENDENT  INDEPENDENT  PARAMETERS  EFFECT      LEGEND  RANDOM EFFECTS

               LRESP      LCONC        B1          _CLASS_     B1.1    B1
                                       B2          _CLASS_     B2.1    B2
                                       B3          _CLASS_     B3.1
                                       B4          _CLASS_     B4.1

                                  MIXNLIN OPTIONS
                      METHOD OF ESTIMATION:            REML
                      EXPAND ABOUT:                    AVERAGE
                      RANDOM-EFFECT ESTIMATION:        EB
                      STD. ERROR ESTIMATES:            ASYMPTOTIC
                      ALTERNATIVE LARGE SAMPLE F-TEST: MODIFIED

                            REPEATED MEASURES MULTIVARIATE
                        NON-LINEAR LEAST SQUARES ITERATIVE PHASE

                               STAGE 1 ESTIMATION: OLS
              CYCLE         ITER     HALFSTEP         B1.1         B2.1         B3.1
    :          B4.1  RESIDUAL SS

                 0            0            0            1          100         -1.1
    :         -17.8 33.804478099

                 1            1            1 0.5844564657  76.09017768 -1.136690733
    : -17.83211345 27.461027148

                 1            2            1 0.6533461168 78.739466398 -1.134175689
    : -17.83430887 27.231984484

                 1            3            1 0.6608667655 78.673645335 -1.137935427
    : -17.82826454 27.231173492

                 1            4            1 0.6618708663 78.665050953 -1.138732464
    : -17.82752003  27.23115324
                           NOTE: CONVERGENCE CRITERION MET.
```

MODEL FITTING INFORMATION

DESCRIPTION	VALUE
TOTAL OBSERVATIONS	60
AVERAGE MODEL R-SQUARE:	0.871908
AVERAGE MODEL CONCORDANCE CORRELATION:	0.931886
CONDITIONAL MODEL R-SQUARE:	0.871908
CONDITIONAL MODEL CONCORDANCE CORRELATION:	0.931886
VARIANCE-COVARIANCE CONCORDANCE CORRELATION:	0.60252
APPROX. CHI-SQUARE FOR H0: COVARIANCE STRUCTURE IS CORRECT	50.8083
DEGREES OF FREEDOM	10
PR > CHI-SQUARE	0.0001
AKAIKE'S INFORMATION CRITERION (AIC)	-65.5351
SCHWARTZ'S BAYESIAN CRITERION (SBC)	-70.5985
-2*LOG LIKELIHOOD	121.0702

FINAL PARAMETER ESTIMATES

STAGE 1 ESTIMATION: OLS

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE
REGRESSION	4	543.653386014025	135.913346503506
RESIDUAL	56	27.2311532397542	0.48627059356704
UNCORRECTED TOTAL	60	570.88453925378	.

PARAMETER	EFFECT	ESTIMATE	ASYMPTOTIC STD. ERROR	T FOR H0: PARAMETER=0	PR > |T|	LOWER 95% LIMIT	UPPER 95% LIMIT
B1	_CLASS_	0.66187087	0.20062864	3.30	0.0017	0.25996341	1.06377832
B2	_CLASS_	78.665051	14.637675	5.37	0.0001	49.3422643	107.987838
B3	_CLASS_	-1.1387325	0.19083354	-5.97	0.0001	-1.521018	-0.7564469
B4	_CLASS_	-17.82752	0.48120906	-37.05	0.0001	-18.791498	-16.863542

D.10 REML fit of final model (4.17) with β_2 and β_3 fixed

```
>  txb2.fit2.b3.b2.fR<-
nlme(object=log(Respons)~txb2.fun2(b0,b1,b2,b3,Conc),est.method="RML",
+     fixed=list(b0~.,b1~.,b2~.,b3~.),random=list(b0~.,b1~.),cluster=~Volnr,
+     data=txb2.data,start=list(fixed=c(0,4.365,-1.13903,-17.827100)))
> summary(txb2.fit2.b3.b2.fR)
Call:
  Model: log(Respons) ~ txb2.fun2(b0, b1, b2, b3, Conc)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
 Random: list(b0 ~ ., b1 ~ .)
Cluster:  ~ Volnr
   Data: txb2.data
Estimation Method: RML
Convergence at iteration: 3
Approximate Restricted Loglikelihood: -17.31241
Restricted AIC: 50.62483
Restricted BIC: 67.37958

Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
       b0       b1
 0.4067271 0.786155
 Correlation of Random Effects
          b0
b1 0.7237369

 Cluster Residual Variance: 0.05499859

Fixed Effects Estimate(s):
          Value Approx. Std.Error  z ratio(C)
b0  -0.4495009         0.1959642   -2.293791
b1   4.3654430         0.3269741   13.351035
b2  -1.1358771         0.0619399  -18.338374
b3 -17.8343423         0.1604168 -111.175052

 Conditional Correlation(s) of Fixed Effects Estimates
            b0          b1          b2
b1  0.58025601
b2 -0.31710328  0.08774867
b3  0.18382262 -0.14264827 -0.82558191

Random Effects (Conditional Modes):
          b0          b1
1  0.4978969  0.98271157
2 -0.1120963  0.17257726
3  0.1274678  0.67060301
4 -0.2521369 -1.06613950
5  0.2743018 -0.08132983
6 -0.5354333 -0.67842250


Standardized Population-Average Residuals:
       Min         Q1        Med        Q3      Max
 -2.745423 -0.3741319 0.02918201 0.5556255 1.782257

Number of Observations: 60
Number of Clusters: 6
```

- Assesment of fit

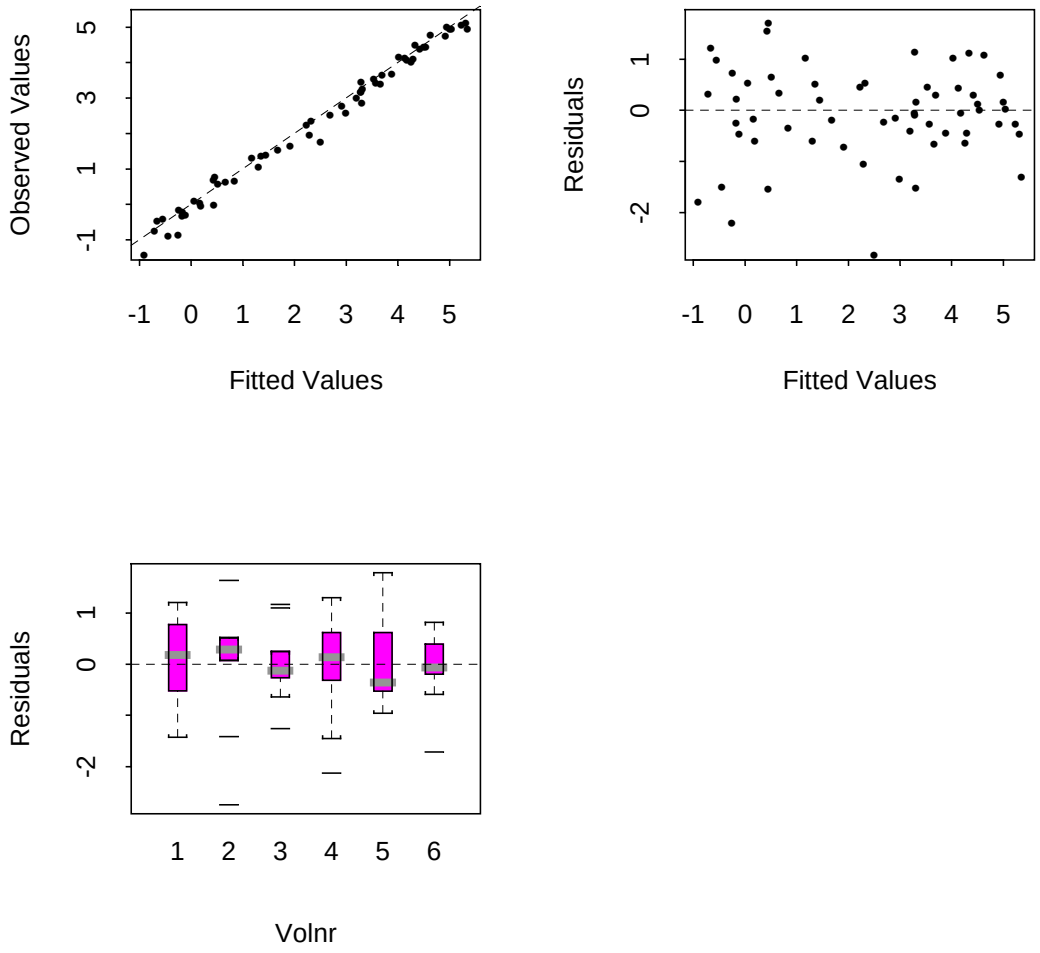

Figure D.3 Plots of standardized residuals and fitted values from REML-fit of model (4.17) with β_2 and β_3 kept fixed

D.11 First-order linearization fit (REML) of model (4.17) with β_2 and β_3 fixed

```
%nlinmix(data=mydat.txb,response=lresp,subject=volnr,
         model=%str(
         e1=exp(b0+u0);
         e2=exp(b1+u1);
         e4=b3-lconc;
         e6=exp(b2*e4);
         e7=1+e6;
         e8=e2/e7;
         e9=e1+e8;
         e15=e7**2;
         pred=log(e9);
         ),
         derivs=%str(
         d_b0=e1/e9;
         d_b1=e8/e9;
         d_b2=-(((e2*(e6*e4))/e15)/e9);
         d_b3=-(((e2*(e6*b2))/e15)/e9);
         d_u0=d_b0;
         d_u1=d_b1;
         ),
         parms=%str(b0=0 b1=4.6 b2=-1.6 b3=-17.4),
         random=u0 u1,
         type=un,
         expand=zero,
         method=REML
```

```
                              The MIXED Procedure

                           Class Level Information

                          Class     Levels  Values

                          _SUBJECT       6  1 2 3 4 5 6

                                 Parameter Search

        COL1      COL2      COL3      COL4  Variance   REML_LL  -2REML_LL  Objective

      0.1698    0.2223    0.6203    0.0535    0.0535  -16.9017    33.8034   -69.1178


                         REML Estimation Iteration History

                 Iteration  Evaluations     Objective     Criterion

                         1            1  -69.11775975    0.00000000

                             Convergence criteria met.


                        Covariance Parameter Estimates (REML)

Cov Parm              Ratio      Estimate     Std Error       Z  Pr > |Z|  Alpha     Lower   Upper

D_U0 UN(1,1)     3.17680489    0.16983254    0.12957029    1.31    0.1899   0.05   -0.0841  0.4238
     UN(2,1)     4.15834990    0.22230610    0.18774946    1.18    0.2364   0.05   -0.1457  0.5903
     UN(2,2)    11.60291412    0.62029379    0.39756624    1.56    0.1187   0.05   -0.1589  1.3995
Residual         1.00000000    0.05346017    0.01114722    4.80    0.0001   0.05    0.0316  0.0753
```

Model Fitting Information for _RESID

Description	Value
Observations	60.0000
Variance Estimate	0.0535
Standard Deviation Estimate	0.2312
REML Log Likelihood	-16.9017
Akaike's Information Criterion	-20.9017
Schwarz's Bayesian Criterion	-24.9524
-2 REML Log Likelihood	33.8034
PARMS Model LRT Chi-Square	0.0000
PARMS Model LRT DF	3.0000
PARMS Model LRT P-Value	1.0000

Solution for Fixed Effects

Parameter	Estimate	Std Error	DDF	T	Pr > |T|	Alpha	Lower	Upper
D_B0	-0.41230055	0.19595640	46	-2.10	0.0409	0.05	-0.8067	-0.0179
D_B1	4.36515539	0.32739608	46	13.33	0.0001	0.05	3.7061	5.0242
D_B2	-1.13894721	0.06329165	46	-18.00	0.0001	0.05	-1.2663	-1.0115
D_B3	-17.82727673	0.15952653	46	-111.8	0.0001	0.05	-18.1484	-17.5062

Solution for Random Effects

Parameter	Subject	Estimate	SE Pred	DDF	T	Pr > |T|
D_U0	_SUBJECT 1	0.52015256	0.22085768	46	2.36	0.0228
D_U1	_SUBJECT 1	0.97673884	0.33152487	46	2.95	0.0050
D_U0	_SUBJECT 2	-0.12906204	0.22085768	46	-0.58	0.5618
D_U1	_SUBJECT 2	0.17051496	0.33152487	46	0.51	0.6095
D_U0	_SUBJECT 3	0.10652007	0.22085768	46	0.48	0.6319
D_U1	_SUBJECT 3	0.68213748	0.33152487	46	2.06	0.0453
D_U0	_SUBJECT 4	-0.18177558	0.22085768	46	-0.82	0.4147
D_U1	_SUBJECT 4	-1.07127684	0.33152487	46	-3.23	0.0023
D_U0	_SUBJECT 5	0.25732062	0.22085768	46	1.17	0.2500
D_U1	_SUBJECT 5	-0.07997362	0.33152487	46	-0.24	0.8104
D_U0	_SUBJECT 6	-0.57315563	0.22085768	46	-2.60	0.0126
D_U1	_SUBJECT 6	-0.67814082	0.33152487	46	-2.05	0.0465

Tests of Fixed Effects

Source	NDF	DDF	Type III F	Pr > F
D_B0	1	46	4.43	0.0409
D_B1	1	46	177.77	0.0001
D_B2	1	46	323.83	0.0001
D_B3	1	46	12488.32	0.0001

D.12 Profile likelihood confidence intervals

```
> likrat.ci
function(begin, end)
{
        step <- (end - begin)/10
        if(begin > end)
                step <-  - step
        m0 <- nlme(object = log(Respons) ~ txb2.fun2(b0, b1, b2, b3, Conc),
        +               est.method = "ML", fixed = list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .),
        +                random = list(b0 ~ ., b1 ~ .), cluster =  ~ Volnr, data =
txb2.data,
        +                start = list(fixed = c(-0.4480476, 4.365339, -1.136248, -
17.83356)))
        x <- begin
        p <- 1
        while(abs(x - end) > 0 & abs(p - 0.025) > 0.0001) {
                assign("x", x, frame = 1)
                m1 <- nlme(object = log(Respons) ~ txb2.funb(b0, b1, b2, x, Conc),
                +            est.method = "ML", fixed = list(b0 ~ ., b1 ~ ., b2 ~.),
                +           random = list(b0 ~ ., b1 ~ .), cluster =  ~ Volnr, data =
txb2.data,
                +            start = list(fixed = c(-0.4480476, 4.365339, -1.136248)))
                p <- (anova(m0, m1)[[14]])[2]
                cat("x,p ", x, p, fill = T)
                x <- x + step
        }
        if(abs(p - 0.025) > 0.0001)
                r <- x - step
        else r <- NA
        r
}
```

```
> likrat.ci(-17.52,-17.51)
x,p  -17.52 0.025577204930869
x,p  -17.519 0.0251321080189638
x,p  -17.518 0.0246937945650225

> likrat.ci(-18.25,-18.2)
x,p  -18.25 0.0178960122017355
x,p  -18.245 0.0199238509097972
x,p  -18.24 0.0213477192297413
x,p  -18.235 0.0228755357041914
x,p  -18.23 0.024491389451069
x,p  -18.225 0.0262171231530761
```

Appendix E
Gastric emptying in dogs

E.1 Data (source: W. De Ridder, unpublished data).

Table E.1 Gastric emptying in dogs as measured by weight (g) of total duodenal efflux.

Time	Dog							
(min)	1	2	3	4	5	6	7	8
15	1.36	1.91	0.19	1.81	0.75	1.79	0.80	0.48
30	1.74	3.10	0.93	2.36	1.23	2.41	1.19	0.95
45	2.04	3.31	1.41	2.53	1.69	2.59	1.60	1.07
60	2.34	3.62	1.61	2.85	2.51	2.76	1.74	1.21
75	3.06	4.32	1.90	3.08	3.08	3.30	2.36	1.29
90	4.68	7.17	2.83	3.27	3.84	4.11	2.75	1.38
105	7.70	10.92	5.72	3.80	5.17	7.47	4.36	1.63
120	10.90	15.59	9.18	4.59	6.93	12.10	8.19	1.89
135	15.33	19.12	13.58	7.27	9.50	17.82	11.83	2.58
150	19.30	21.46	17.73	11.10	12.21	22.85	17.81	3.02
165	22.81	23.28	22.32	15.18	15.42	27.04	22.80	4.59
180	25.55	24.66	25.88	18.44	18.26	30.97	26.90	7.59
195	28.37	25.95	30.33	21.58	22.77	35.56	30.84	11.04
210	31.61	26.60	32.98	25.75	26.84	39.55	33.85	16.33
225	34.94	27.46	36.77	30.32	31.39	42.38	37.12	22.28
240	37.16	27.81	38.91	34.13	35.76	43.81	40.00	28.36
255	39.26	28.65	40.87	37.30	39.10	45.32	42.63	32.98
270	41.24	29.25	42.25	39.11	43.00	47.51	44.62	35.97
285	43.57	30.32	44.16	41.04	46.30	48.94	45.93	39.81
300	46.22	30.67	45.10	42.07	48.52	50.19	46.90	42.59

E.2 Evaluation of alternative parametrizations

```
> mc.gempt
function(b0, b1, b2, b3, t, nrsim = 100)
{
        pars <- matrix(nrow = nrsim, ncol = 4)
        pars2 <- pars
        for(i in 1:nrsim) {
                y <- b0 + b1 * (1 - 2^( - (t/b2)^b3)) + rnorm(length(t), 0, 0.8)
                y <- (y > 0) * y
                dat <- as.data.frame(cbind(t, y))
                assign("dat", dat, frame = 1)
                fiti <- nls(y ~ GEmpt.fun0(b0, b1, b2, b3, t), dat, start = list(b0 =
b0, b1 = b1, b2 = b2, b3 = b3))
                pars[i,  ] <- fiti$parameters
                fiti <- nls(y ~ GEmpt.fun2(b0, b1, b2, b3, t), dat, start = list(b0 =
log(b0), b1 = log(b1), b2 = log(b2), b3 = b3))
                pars2[i,  ] <- fiti$parameters
        }
        list(pars, pars2)
}
```

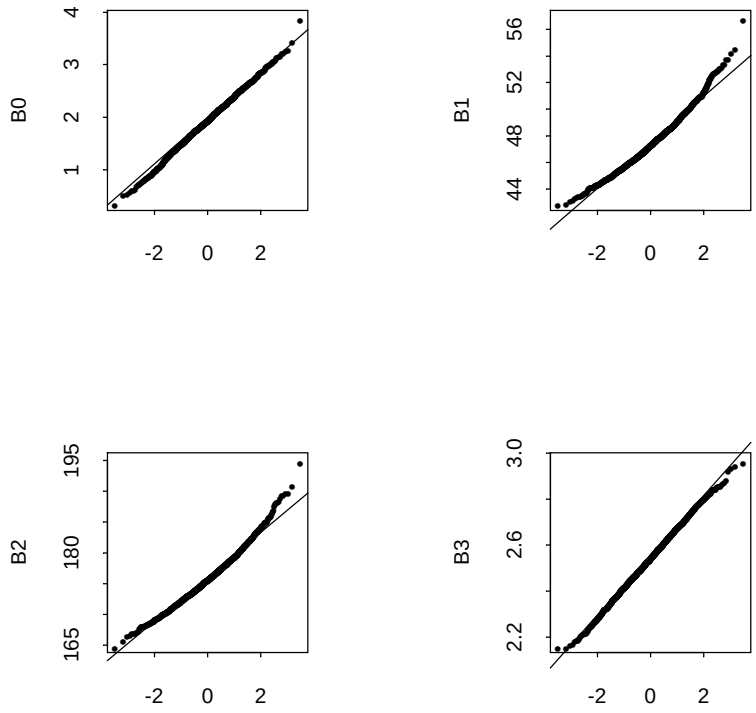

Figure E.1 Normal quantile plot of 2000 Monte Carlo simulations of parameter estimates for (4.19)

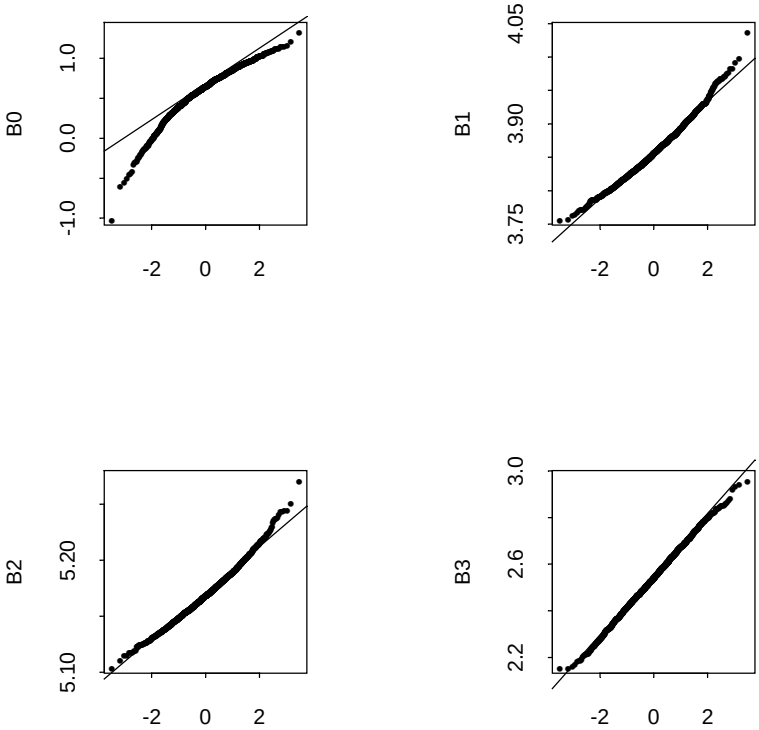

Figure E.2 Normal quantile plot of 2000 Monte Carlo simulations of parameter estimates obtained for reparametrizating of (4.19)

E.3 Derivatives and OLS fit on pooled data

```
> GEmpt.fun2<-deriv(~exp(b0)+exp(b1)*(1-2^(-(t/exp(b2))^b3)),
+              c("b0","b1","b2","b3"),function(b0,b1,b2,b3,t) NULL)
> GEmpt.fun2
function(b0, b1, b2, b3, t)
{
        .expr1 <- exp(b0)
        .expr2 <- exp(b1)
        .expr3 <- exp(b2)
        .expr4 <- t/.expr3
        .expr5 <- .expr4^b3
        .expr7 <- 2^( - .expr5)
        .expr9 <- .expr2 * (1 - .expr7)
        .value <- .expr1 + .expr9
        .grad <- array(0, c(length(.value), 4), list(NULL, c("b0", "b1", "b2", "b3")))
        .grad[, "b0"] <- .expr1
        .grad[, "b1"] <- .expr9
        .grad[, "b2"] <-  - (.expr2 * (.expr7 * (0.693147180559945 * ((.expr4^(b3 - 1))
+               * (b3 * ((t * .expr3)/(.expr3^2)))))))
        .grad[, "b3"] <- .expr2 * (.expr7 * (0.693147180559945 * (.expr5 *
+               (log(.expr4)))))
        attr(.value, "gradient") <- .grad
        .value
}
>
```

```
> Pool.fit<-nls(Volume~GEmpt.fun2(b0,b1,b2,b3,Time),Gast.Empt1,
+              list(b0=0,b1=log(50),b2=log(160),b3=3.5))
> summary(Pool.fit)
Formula: Volume ~ GEmpt.fun2(b0, b1, b2, b3, Time)
Parameters:
       Value Std. Error     t value
b0 0.0647676  0.8453850   0.0766131
b1 3.8157400  0.0765889  49.8211000
b2 5.2276300  0.0392012 133.3540000
b3 2.9284400  0.3221010   9.0916800

Residual standard error: 4.82859 on 156 degrees of freedom

Correlation of Parameter Estimates:
        b0      b1      b2
b1 -0.5830
b2 -0.0132  0.7740
b3  0.6440 -0.8700 -0.5290
```

E.4 Standard two-stage approach

- Individual estimates, mean value, standard error

```
>  Indiv.fit<-nls(Volume~exp(b0[Dog])+exp(b1[Dog])*(1-2^((Time/exp(b2[Dog]))^b3[Dog]))
+       ,data=Gast.Empt1,start=list(b0=rep(0,8),b1=log(c(46,31,45,42,49,50,47,43)),
+       b2=rep(log(183),8),b3=rep(3.5,8)))
> indiv.coef<-matrix(coef(Indiv.fit),nrow=8,ncol=4)
> indiv.coef
            [,1]     [,2]     [,3]     [,4]
[1,] -0.53825270 3.861027 5.173514 2.554933
[2,]  0.52640612 3.310081 4.821874 2.803860
[3,] -2.08774069 3.803150 5.129202 3.143290
[4,]  0.74167468 3.709930 5.282319 3.981411
[5,]  0.34101467 4.021617 5.385187 3.159615
[6,]  0.22437981 3.867700 5.084490 3.188566
[7,] -0.80279923 3.833006 5.150733 3.349876
[8,]  0.01160388 3.725897 5.416734 6.033824
> apply(indiv.coef,2,mean)
[1] -0.1979642  3.7665510  5.1805066  3.5269218
>  sqrt(apply(indiv.coef,2,var)/8)
[1] 0.32659777 0.07358122 0.06674408 0.38698962
```

E.5 Nonlinear mixed effects model with independence and homogeneity of variance

```
>   fit0<-nlme(object=Volume~GEmpt.fun2(b0,b1,b2,b3,Time),
+       random=list(b0~.,b1~.,b2~.,b3~.),
+       fixed=list(b0~.,b1~.,b2~.,b3~.),
+       data=Gast.Empt1,cluster=~Dog,
+       start=list(fixed=c(-0.1979642,3.7665510,5.1805066,3.5269218)))
> summary(fit0)
Call:
  Model: Volume ~ GEmpt.fun2(b0, b1, b2, b3, Time)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
 Random: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
Cluster:  ~ Dog
   Data: Gast.Empt1
Estimation Method: ML
Convergence at iteration: 10
Approximate Loglikelihood: -251.6222
AIC: 533.2445
BIC: 579.3721
Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
       b0        b1        b2        b3
 0.710714 0.1919922 0.1766827 0.9920573
 Correlation of Random Effects
             b0           b1           b2
b1 -0.432380416
b2  0.168845696  0.663763581
b3  0.183966014  0.001812201  0.644309756

 Cluster Residual Variance: 0.6744823
Fixed Effects Estimate(s):
         Value Approx. Std.Error  z ratio(C)
b0 -0.1987368        0.27864791  -0.7132185
b1  3.7661029        0.06867498  54.8395198
b2  5.1780814        0.06270830  82.5740988
b3  3.5060726        0.35587062   9.8520992
 Conditional Correlation(s) of Fixed Effects Estimates
            b0           b1           b2
b1 -0.41992139
b2  0.15708515  0.66187342
b3  0.20663101 -0.01770493  0.62866229
Random Effects (Conditional Modes):
          b0           b1           b2          b3
1  0.1929964  0.06746953 -0.01102627 -0.8429983
2  0.6302894 -0.44697198 -0.35682541 -0.7483886
3 -0.3301114  0.02112396 -0.04538346 -0.2631612
4  0.9166330 -0.05549394  0.10314347  0.4780128
5  0.3745796  0.24314357  0.19621287 -0.3413142
6 -1.4282272  0.13392774 -0.10095864 -0.5191396
7 -0.5848854  0.06730675 -0.02668753 -0.1636267
8  0.2287256 -0.03050563  0.24152496  2.4006158
Standardized Population-Average Residuals:
       Min         Q1        Med        Q3      Max
 -2.227605 -0.6375675 0.09583168 0.5568108 2.416533
Number of Observations: 160
Number of Clusters: 8
```

```
> par(mfrow=c(2,2))
> plot(fit0,option="s")
> x<-log(Gast.Empt1$Volume-residuals(fit0)$cluster)
> y<-log(abs(residuals(fit0)$cluster))
> plot(x[y>-4],y[y>-4],xlab="Logarithm of Predicted Value",ylab="Logarithm of Absolute
Residual")
> lines(lowess(x[y>-4],y[y>-4]))
```

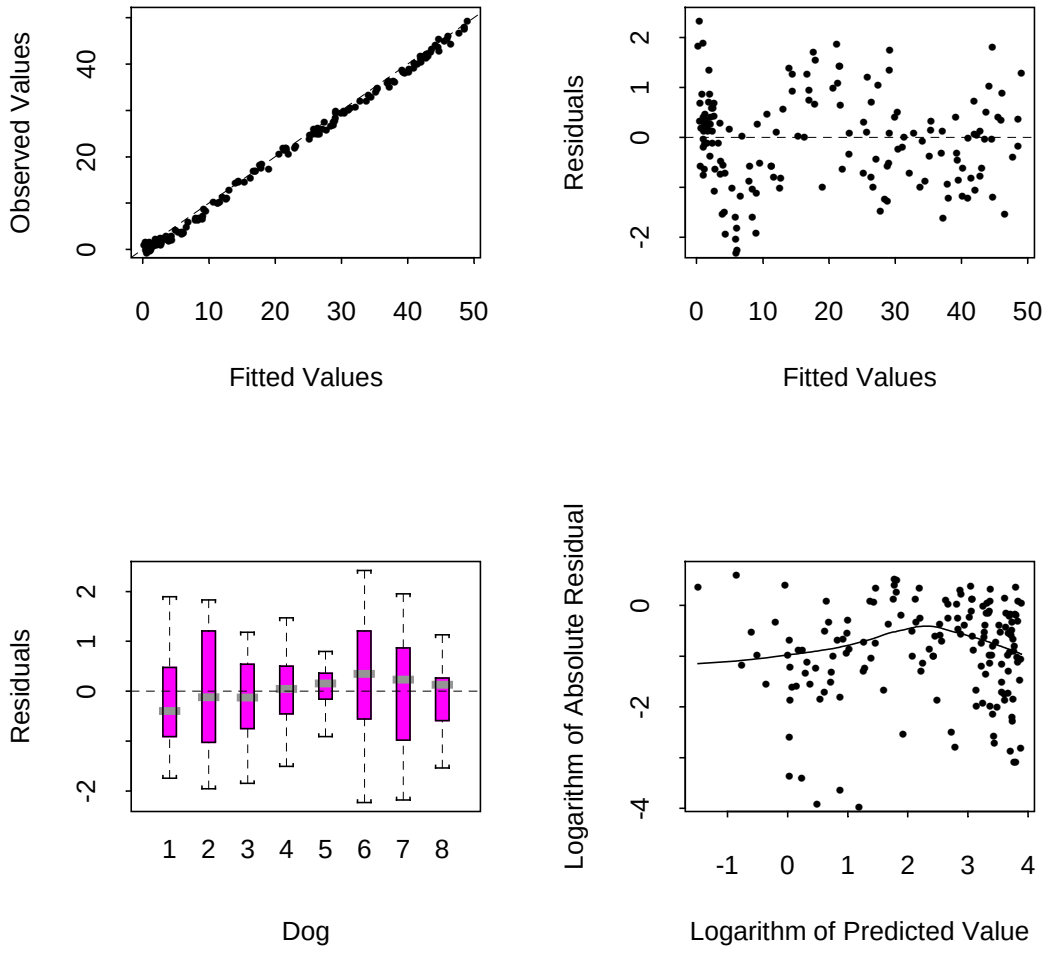

Figure E.3 Graphical assessment of fit assuming independence of successive measurements and homogeneity of variance. Notice the cyclic pattern in the plot of the standardized residuals versus the fitted values. There is no evidence of variance being dependent on the level of the response.

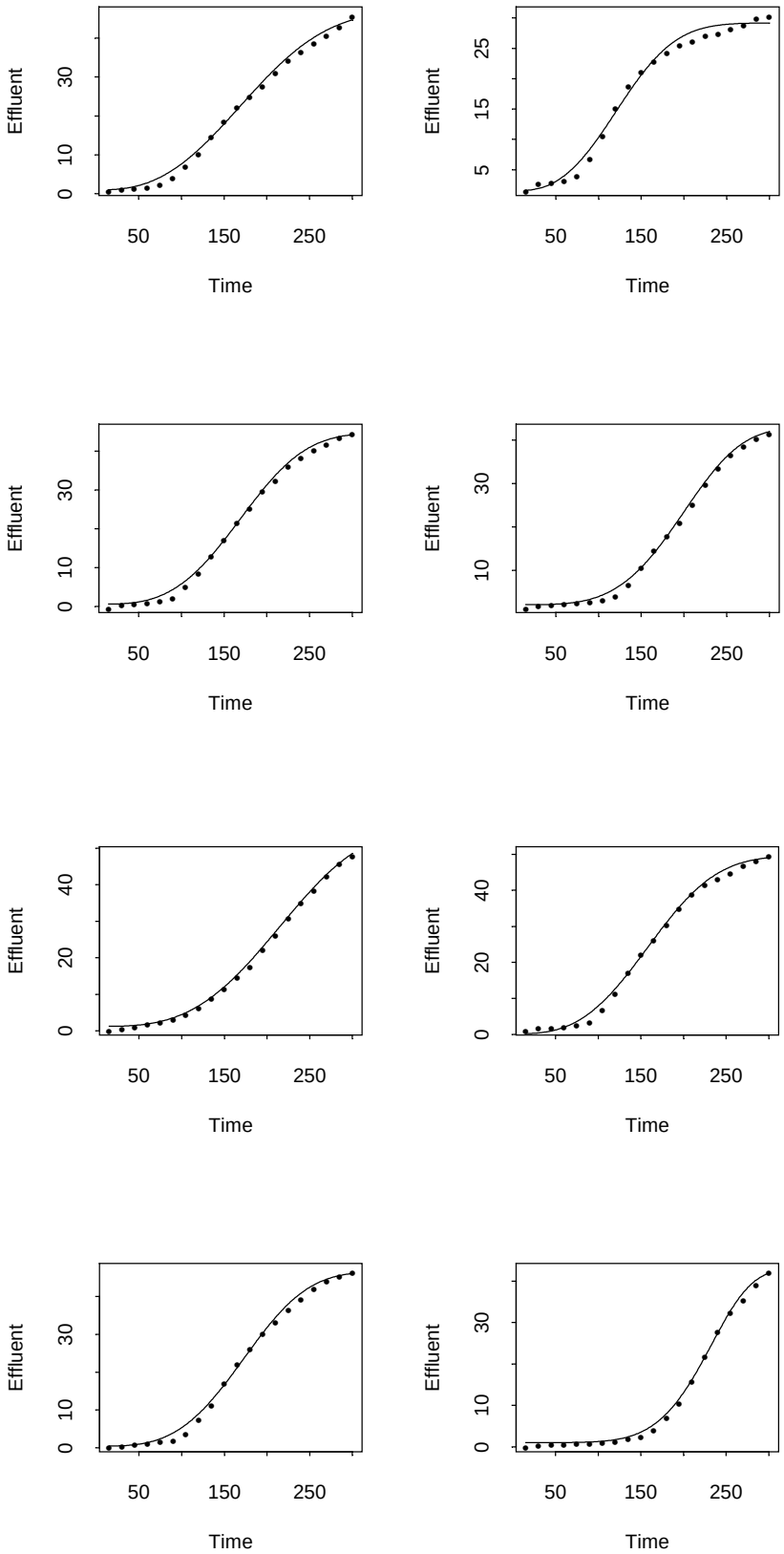

Figure E.4 Individual data and fitted curves for the eight dogs.
Notice that the model over-estimates the initial upward rise in six dogs.

E.6 Intra-individual covariance structure (Variogram)

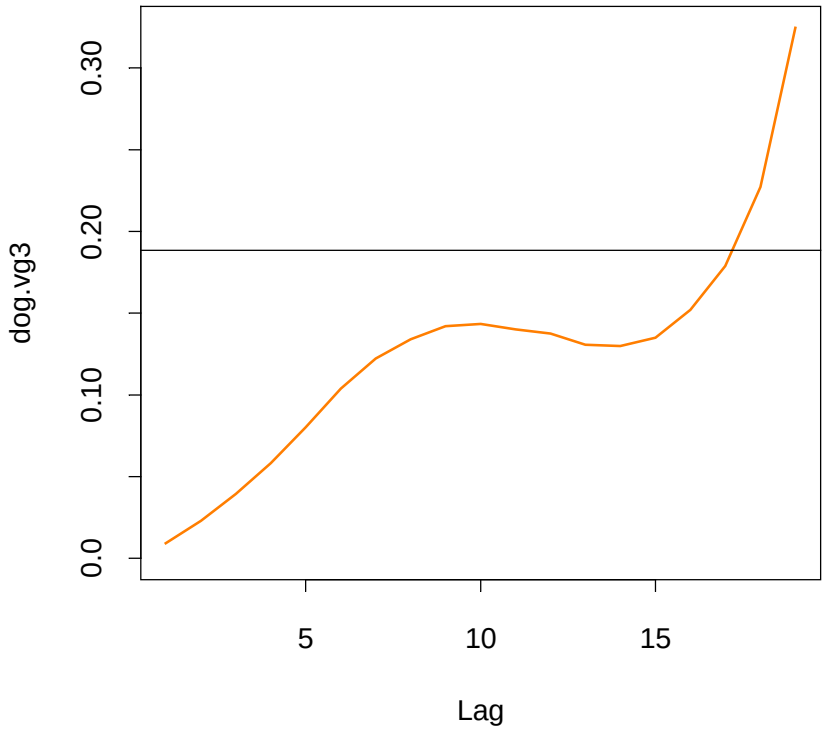

Figure E.5 Sample variogram obtained from differences with mean value of log-transformed data .

E.7 Backward elimination of random effects

Table E.2 Summary of backward elimination of random effect from the parameters. Missing values indicate convergence problems.

Step	Parameter as fixed	Log likelihood	Likelihood Ratio	P-value
0	Model: all parameters random, log-likelihood = -175.88, # parameters reference model = 17, # parameters reduced model = 13			
	β_0	-175.99	0.23	0.994
	β_1	-	-	-
	β_2	-	-	-
	β_3	-	-	-
1	Model: random effects for β_0 removed, log-likelihood = -175.99 # parameters reference model = 13, # parameters reduced model = 10			
	β_1	-186.44	20.90	<0.001
	β_2	-208.79	65.58	<0.001
	β_3	-185.36	18.74	<0.001

E.8 ML fit of model with β_0 fixed and ARMA(1,1) type of intra-individual covariance

```
>    fit.arma.b0R<-nlme(object=Volume~GEmpt.fun2(b0,b1,b2,b3,Time),
+       fixed=list(b0~.,b1~.,b2~.,b3~.),
+       random=list(b1~.,b2~.,b3~.),
+       data=Gast.Empt1,cluster=~Dog,serial.structure="arma11",
+       est.method="RML",start=list(fixed=c(0,3.5,5.227,2.7616)))
> summary(fit.arma.b0R)
Call:
  Model: Volume ~ GEmpt.fun2(b0, b1, b2, b3, Time)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ ., b3 ~ .)
 Random: list(b1 ~ ., b2 ~ ., b3 ~ .)
Cluster:  ~ Dog
   Data: Gast.Empt1
Estimation Method: RML
Convergence at iteration: 9
Approximate Restricted Loglikelihood: -180.7663
Restricted AIC: 387.5326
Restricted BIC: 427.5099
Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
       b1        b2        b3
 0.193391 0.1920987 0.8252286
 Correlation of Random Effects
            b1          b2
b2  0.73320512
b3 -0.01870585  0.58668007
 Cluster Residual Variance: 1.616544
 Serial Correlation Structure: arma11
 Serial Correlation Parameter(s):  0.7912779 -0.4117012
Fixed Effects Estimate(s):
         Value Approx. Std.Error  z ratio(C)
b0 -0.1010529        0.45032720  -0.2243988
b1  3.7837026        0.07092815  53.3455713
b2  5.1795935        0.06881800  75.2651033
b3  3.4152418        0.31289047  10.9151353
 Conditional Correlation(s) of Fixed Effects Estimates
            b0          b1          b2
b1 -0.15696075
b2  0.03655882  0.71204788
b3  0.13906003 -0.07579889  0.53973224
Random Effects (Conditional Modes):
            b1          b2          b3
1  0.101989696  0.01290395 -0.82534452
2 -0.433893984 -0.37891231 -0.39634300
3  0.011344545 -0.04062550 -0.27086630
4 -0.035846930  0.09194222  0.44936436
5  0.186203725  0.17086274 -0.05805926
6  0.112763943 -0.08151057 -0.44601481
7  0.052729765 -0.02041924 -0.17091325
8  0.004709239  0.24575871  1.71817679
Standardized Population-Average Residuals:
       Min         Q1         Med        Q3      Max
 -1.872997 -0.5381373 -0.06249976 0.4600694 1.501183
Number of Observations: 160
Number of Clusters: 8
```

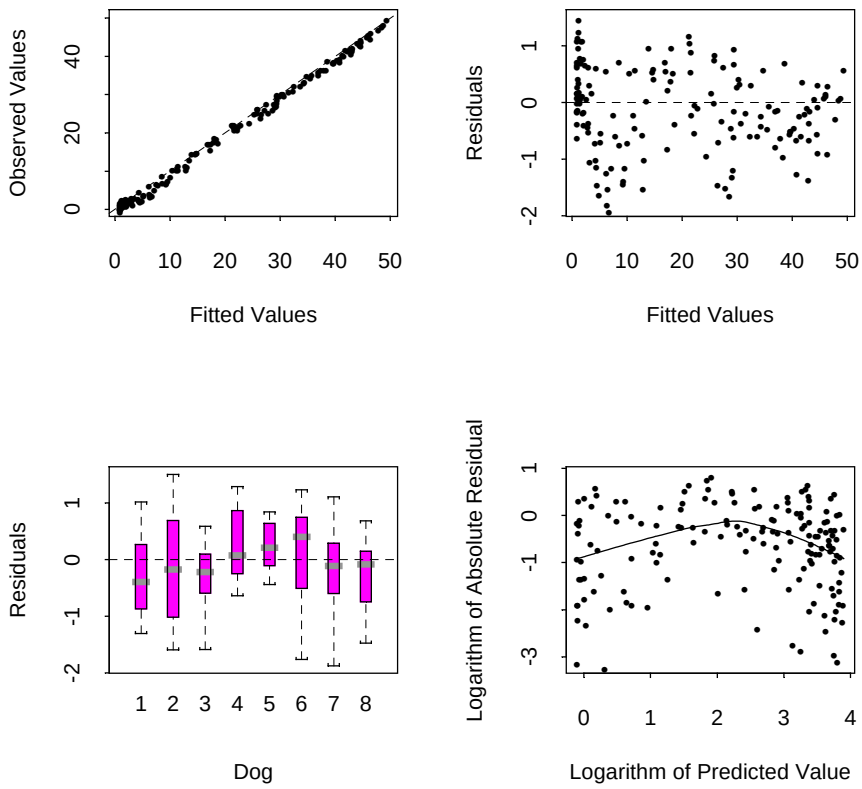

Figure E.6 Graphical assessment of REML fit with serial correlation of type ARMA(1,1) and β_0 kept fixed

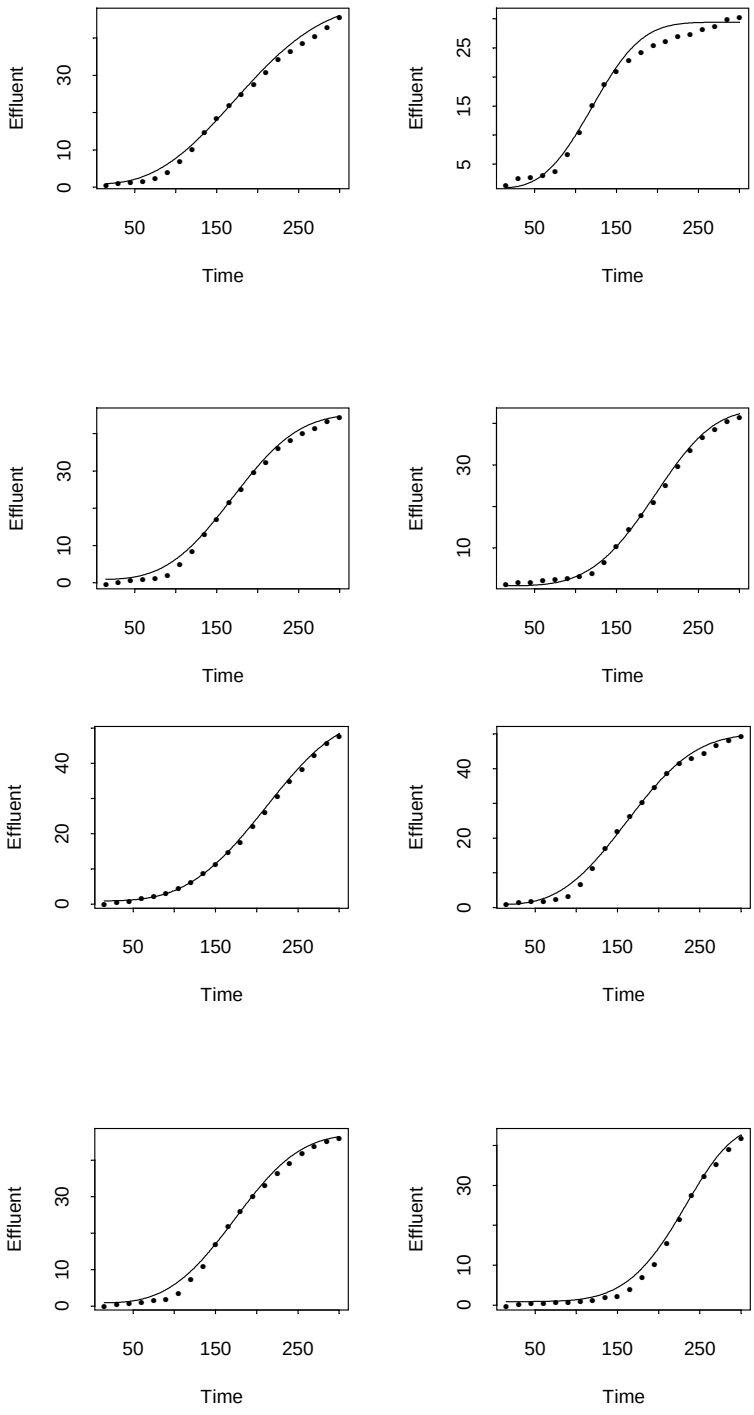

Figure E.7 Individual data and fitted curves for the eight dogs using the REML-based *nlme* individual parameter estimates (blup).

Appendix F
Acetylcholine levels in striatum of rats

F.1 Data (source: D. Scheller, JRF Neuss).

Table F.1 Acetylcholine concentration in striatum perfusate (nmol/l)

Exp. Nr.	Dose	Time following treatment (hrs.)				
		0	1	2	3	4
1	2.50	1.80	3.25	2.29	1.77	.
2	2.50	1.29	1.77	1.09	.	0.30
3	2.50	0.86	2.04	2.06	0.89	0.78
4	2.50	1.60	3.55	2.16	.	1.28
5	2.50	0.35	2.37	1.83	0.79	.
6	2.50	1.62	2.78	2.59	1.87	2.33
7	2.50	0.75	1.39	1.85	2.21	1.63
8	2.50	1.20	5.49	2.59	1.74	1.38
9	5.00	1.53	11.05	12.42	7.05	2.87
10	5.00	1.27	3.09	2.39	1.78	0.30
11	5.00	1.02	4.62	2.81	1.28	1.04
12	5.00	1.73	8.56	3.95	2.31	2.90
13	5.00	1.09	2.53	3.58	2.81	1.89
14	5.00	1.55	16.40	6.59	1.30	0.73
15	5.00	1.77	7.12	3.43	2.39	.
16	5.00	1.47	6.72	2.32	1.43	1.17

F.2 Partial derivatives and OLS fit on pooled data

```
> ACH.fun0<-deriv(~log(exp(b0)+exp(b1)*(x^3)*exp(-
b2*x)),c("b0","b1","b2"),function(b0,b1,b2,x) NULL)
> ACH.fun0
function(b0, b1, b2, x)
{
        .expr1 <- exp(b0)
        .expr4 <- (exp(b1)) * (x^3)
        .expr7 <- exp((( - b2) * x))
        .expr8 <- .expr4 * .expr7
        .expr9 <- .expr1 + .expr8
        .value <- log(.expr9)
        .grad <- array(0, c(length(.value), 3), list(NULL, c("b0", "b1", "b2")))
        .grad[, "b0"] <- .expr1/.expr9
        .grad[, "b1"] <- .expr8/.expr9
        .grad[, "b2"] <-  - ((.expr4 * (.expr7 * x))/.expr9)
        attr(.value, "gradient") <- .grad
        .value
}
```

```
> ACH.pool<-nls(LConc~ACH.fun0(b0,b1,b2,Time),
+           data=ACH.data,start=list(b0=1,b1=3,b2=3))
> summary(ACH.pool)

Formula: LConc ~ ACH.fun0(b0, b1, b2, Time)

Parameters:
      Value Std. Error t value
b0 0.169771   0.119193 1.42433
b1 3.577350   0.406796 8.79396
b2 2.542550   0.262062 9.70207

Residual standard error: 0.592029 on 72 degrees of freedom

Correlation of Parameter Estimates:
      b0    b1
b1 0.222
b2 0.437 0.906
```

F.3 Individual estimates by separate nonlinear OLS regressions

- Treatment group 1:

```
>  c1
             b0        b1       b2
[1,]  0.5636692 3.5732498 3.166427
[2,] -0.4630436 3.3185609 3.153389
[3,] -0.2670011 2.8301343 2.469077
[4,]  0.3572488 3.9279169 3.171621
[5,] -1.0515274 3.1144795 2.406194
[6,]  0.6107461 2.2994328 2.365882
[7,] -0.2692313 0.6031902 1.216879
[8,]  0.2815980 4.5021207 3.106686
> apply(c1,2,mean)
          b0       b1       b2
 -0.02969267 3.021136 2.632019
> sqrt(apply(c2,2,var)/8)
        b0        b1        b2
 0.1335471 0.4645636 0.2666482
```

- Treatment group 2:

```
> c2
               b0       b1       b2
[1,]  0.400822318 4.278133 1.987474
[2,] -0.372162466 3.462131 2.499577
[3,] -0.004179438 4.146978 2.841907
[4,]  0.784828582 5.213930 3.364601
[5,]  0.079407016 1.971250 1.577645
[6,]  0.011594553 6.223226 3.399419
[7,]  0.644204706 4.802558 3.173785
[8,]  0.278841727 5.446294 3.760609
> apply(c2,2,mean)
        b0       b1       b2
 0.2279196 4.443062 2.825627
> sqrt(apply(c1,2,var)/8)
        b0        b1        b2
 0.2050479 0.4187859 0.2399738
```

F.4 Nonlinear mixed effects fit of reduced model (4.23)

```
> ACH.fit0<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.,b1~.,b2~.),
+       fixed=list(b0~.,b1~.,b2~.),data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.16,3.57,2.5)))
> summary(ACH.fit0)
Call:
  Model: LConc ~ ACH.fun0(b0, b1, b2, Time)
  Fixed: list(b0 ~ ., b1 ~ ., b2 ~ .)
 Random: list(b0 ~ ., b1 ~ ., b2 ~ .)
Cluster:  ~ Exp
   Data: ACH.data
Estimation Method: ML
Convergence at iteration: 6
Approximate Loglikelihood: -52.12831
AIC: 124.2566
BIC: 147.4315
Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
        b0      b1        b2
 0.3553003 1.14027 0.5325539
 Correlation of Random Effects
          b0        b1
b1 0.4293514
b2 0.1593343 0.8290816
Cluster Residual Variance: 0.1133782
Fixed Effects Estimate(s):
         Value Approx. Std.Error z ratio(C)
b0 0.07260254         0.1128671   0.643257
b1 3.52495921         0.3560379   9.900516
b2 2.47791987         0.1864199  13.292140
 Conditional Correlation(s) of Fixed Effects Estimates
          b0        b1
b1 0.3273953
b2 0.2516640 0.8496148
Random Effects (Conditional Modes):
            b0          b1           b2
 1  0.15155392 -0.23018932 -0.003634371
 2 -0.48081391 -0.72807283  0.134374128
 3 -0.30534015 -0.62991771 -0.029965645
 4  0.10199758 -0.06809415  0.094512028
 5 -0.60023191 -0.50228241  0.015382351
 6  0.22575164 -0.62391511 -0.253511092
 7 -0.27737850 -2.01853227 -0.895084065
 8  0.10643881  0.44822665  0.211285247
 9  0.41627860  0.64216763 -0.446762265
10 -0.28432145 -0.15331597  0.036008311
11 -0.05727494  0.33771024  0.200805495
12  0.47396146  0.95077386  0.259002159
13  0.02428575 -1.07267739 -0.667681425
14  0.08333825  2.06602097  0.680441386
15  0.31714654  0.71548238  0.200966899
16  0.10460831  0.86661543  0.463860859
Standardized Population-Average Residuals:
       Min         Q1        Med        Q3      Max
 -3.226984 -0.4125277 0.00974174 0.4283754 1.968579
Number of Observations: 75
Number of Clusters: 16
```

```
> plot (ACH.fit0,option="s")
```

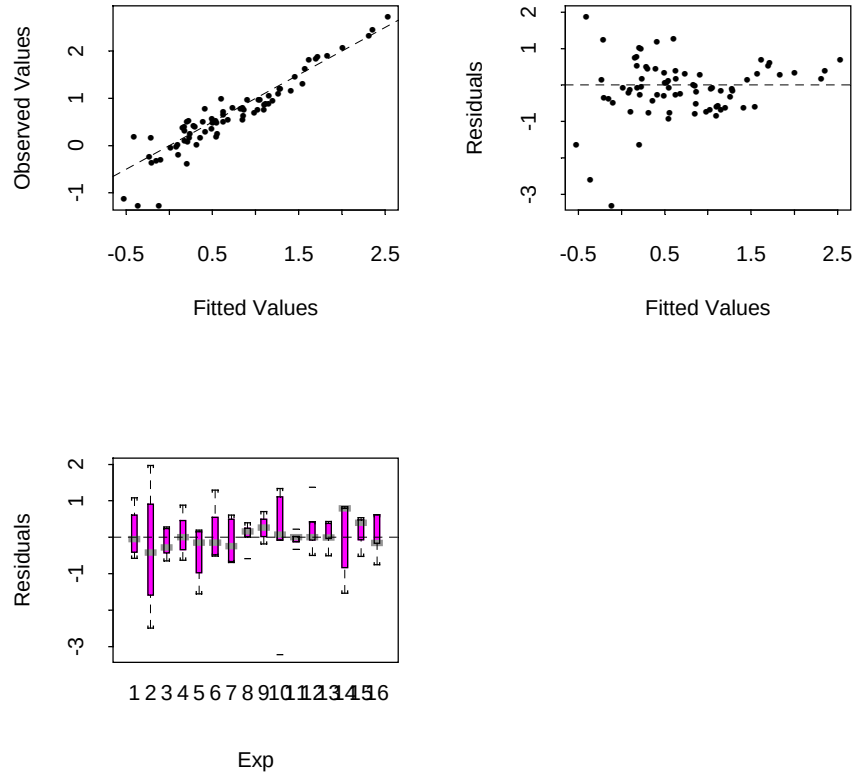

Figure F.1 Graphical assessment of model fit. Note that there are some large outliers present when the predicted value is near the lower detection limit. Experiment 2 does not correspond well to the model and yields high residuals. For the remaining experiments the fit is rather good.

F.5 Graphical exploration for the effect of treatment

```
> ACH.random<-data.frame(ACH.fit0$coef$random)
 >ACH.random$trt<-as.factor(rep(c(2.5,5),c(8,8)))
```

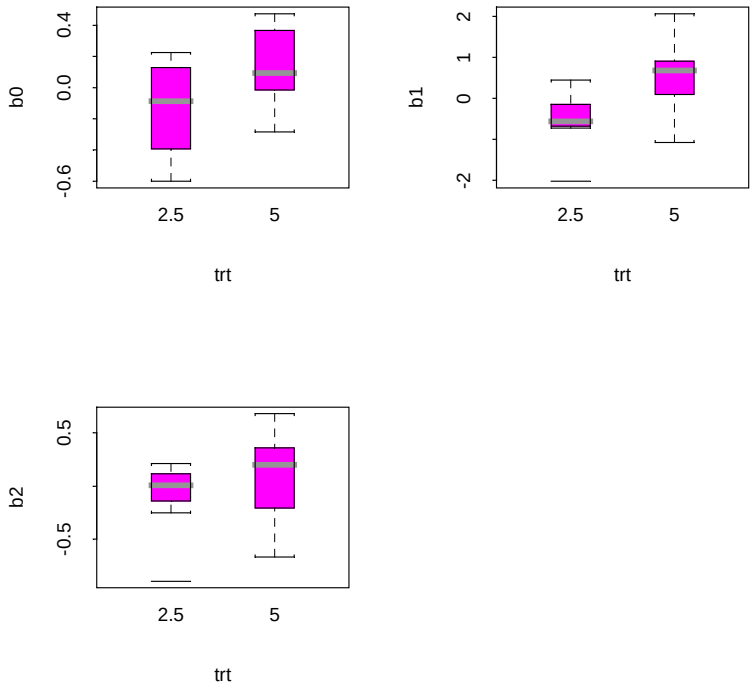

Figure F.2 Graphical assessment of the effect of treatment on the random effects.

F.6 Fit of model (4.22) with effect of treatment on β_1 and β_2

```
> ACH.fit1<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.,b1~.,b2~.),
+       fixed=list(b0~.,b1~Treat,b2~Treat),
+       data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.08,3.2,4.2,2.56,2.56))
+ )
> summary(ACH.fit1)
Call:
  Model: LConc ~ ACH.fun0(b0, b1, b2, Time)
  Fixed: list(b0 ~ ., b1 ~ Treat, b2 ~ Treat)
 Random: list(b0 ~ ., b1 ~ ., b2 ~ .)
Cluster:  ~ Exp
   Data: ACH.data

Estimation Method: ML
Convergence at iteration: 5
Approximate Loglikelihood: -46.89268
AIC: 117.7854
BIC: 145.5952

Variance/Covariance Components Estimate(s):
  Structure: matrixlog
  Standard Deviation(s) of Random Effect(s)
        b0 b1.(Intercept) b2.(Intercept)
 0.3679864      0.9642021      0.5623634
```

```
 Correlation of Random Effects
                     b0 b1.(Intercept)
b1.(Intercept) 0.3678020
b2.(Intercept) 0.2546479 0.8970910

 Cluster Residual Variance: 0.1059516

Fixed Effects Estimate(s):
                     Value Approx. Std.Error z ratio(C)
            b0 0.08159618         0.1134538  0.7192018
b1.(Intercept) 2.95253538         0.4901093  6.0242390
      b1.Treat 1.20079054         0.6298968  1.9063290
b2.(Intercept) 2.48310293         0.3004105  8.2656993
      b2.Treat 0.08648559         0.3795088  0.2278882

 Conditional Correlation(s) of Fixed Effects Estimates
                       b0 b1.(Intercept)    b1.Treat b2.(Intercept)
b1.(Intercept)  0.17671622
      b1.Treat  0.00872442 -0.75223859
b2.(Intercept)  0.20791186  0.88906521    -0.66136106
      b2.Treat -0.02491684 -0.67908329     0.88605432 -0.76253991

Random Effects (Conditional Modes):
            b0 b1.(Intercept) b2.(Intercept)
 1  0.27480240     0.25782375     0.13108542
 2 -0.44996844    -0.17106637     0.16353428
 3 -0.26340752    -0.16551276    -0.01448100
 4  0.19249890     0.40471930     0.21255324
 5 -0.59774705    -0.10468857    -0.03640137
 6  0.34952197    -0.02587488    -0.04311987
 7 -0.27325654    -1.58110601    -0.94175624
 8  0.20525301     0.84438932     0.31000506
 9  0.28957808    -0.10896006    -0.60609834
10 -0.37951199    -0.51110703    -0.03220404
11 -0.11836516    -0.06937520     0.13948559
12  0.47682980     0.52734711     0.27215865
13 -0.09913825    -1.44952875    -0.74176811
14  0.02505979     1.40476088     0.57020855
15  0.28958802     0.26911246     0.16966402
16  0.07826297     0.47906680     0.44713415


Standardized Population-Average Residuals:
       Min        Q1         Med        Q3      Max
 -3.155419 -0.338797 0.003962358 0.4727568 1.914011

Number of Observations: 75
Number of Clusters: 16
```

F.7 Joint test for treatment effect on parameters β_1 and β_2

```
> anova(ACH.fit1,ACH.fit0)
Response: LConc
ACH.fit1
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0, b1.(Intercept), b2.(Intercept)
 block: list(1:3)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
ACH.fit0
 fixed: b0, b1, b2
 random: b0, b1, b2
 block: list(1:3)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
         Model Df    AIC    BIC  Loglik    Test Lik.Ratio   P value
ACH.fit1     1 12 117.79 145.60 -46.893
ACH.fit0     2 10 124.26 147.43 -52.128 1 vs. 2    10.471 0.0053235
```

F.8 β_1 and β_2 fixed instead of random

- β_1 fixed

```
> ACH.fit1.b1f<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.,b2~.),
+       fixed=list(b0~.,b1~Treat,b2~Treat),
+       data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.08,3.2,4.2,2.56,2.56))
+ )
> anova(ACH.fit1.b1f,ACH.fit1)
Response: LConc
ACH.fit1.b1f
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0, b2.(Intercept)
 block: list(1:2)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
ACH.fit1
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0, b1.(Intercept), b2.(Intercept)
 block: list(1:3)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
             Model Df    AIC    BIC  Loglik    Test Lik.Ratio P value
ACH.fit1.b1f     1  9 114.72 135.58 -48.362
    ACH.fit1     2 12 117.79 145.60 -46.893 1 vs. 2    2.9382 0.40125
```

- β_2 fixed

```
> ACH.fit1.b2f<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.,b1~.),
+       fixed=list(b0~.,b1~Treat,b2~Treat),
+       data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.08,3.2,4.2,2.56,2.56))
+ )
> anova(ACH.fit1.b2f,ACH.fit1)
Response: LConc
ACH.fit1.b2f
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0, b1.(Intercept)
 block: list(1:2)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
ACH.fit1
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0, b1.(Intercept), b2.(Intercept)
 block: list(1:3)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
             Model Df    AIC   BIC  Loglik    Test Lik.Ratio P value
ACH.fit1.b2f     1  9 114.74 135.6 -48.371
    ACH.fit1     2 12 117.79 145.6 -46.893 1 vs. 2    2.9572 0.39826
```

- β_1 and β_2 both fixed versus β_2 random

```
> ACH.fit2<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.),
+       fixed=list(b0~.,b1~Treat,b2~Treat),
+       data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.08,3.2,4.2,2.56,2.56))
+ )
> anova(ACH.fit2,ACH.fit1.b1f)
Response: LConc
ACH.fit2
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0
 block: list(1:1)
 covariance structure: identity
 serial correlation structure: identity
 variance function: identity
ACH.fit1.b1f
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0, b2.(Intercept)
 block: list(1:2)
 covariance structure: unstructured
 serial correlation structure: identity
 variance function: identity
             Model Df    AIC    BIC  Loglik    Test Lik.Ratio  P value
    ACH.fit2     1  7 115.92 132.14 -50.961
ACH.fit1.b1f     2  9 114.72 135.58 -48.362 1 vs. 2    5.1987 0.074323
```

F.9 Test for treatment effect on β_1 and β_2, with random effects only for β_0

- Test for β_1:

```
> ACH.fit.b1<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.),
+       fixed=list(b0~.,b1~.,b2~Treat),
+       data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.08,3.2,2.56,2.56))
+ )
> anova(ACH.fit2,ACH.fit.b1)
Response: LConc
ACH.fit2
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0
 block: list(1:1)
 covariance structure: identity
 serial correlation structure: identity
 variance function: identity
ACH.fit.b1
 fixed: b0, b1, b2.(Intercept), b2.Treat
 random: b0
 block: list(1:1)
 covariance structure: identity
 serial correlation structure: identity
 variance function: identity
           Model Df    AIC    BIC  Loglik    Test Lik.Ratio P value
  ACH.fit2     1  7 115.92 132.14 -50.961
ACH.fit.b1     2  6 117.78 131.68 -52.888 1 vs. 2    3.8536 0.04964
```

- Test for β_2:

```
> ACH.fit.b2<-nlme(object=LConc~ACH.fun0(b0,b1,b2,Time),random=list(b0~.),
+       fixed=list(b0~.,b1~Treat,b2~.),
+       data=ACH.data,cluster=~Exp,
+       start=list(fixed=c(0.08,3.2,4.2,2.56))
+ )
> anova(ACH.fit2,ACH.fit.b2)
Response: LConc
ACH.fit2
 fixed: b0, b1.(Intercept), b1.Treat, b2.(Intercept), b2.Treat
 random: b0
 block: list(1:1)
 covariance structure: identity
 serial correlation structure: identity
 variance function: identity
ACH.fit.b2
 fixed: b0, b1.(Intercept), b1.Treat, b2
 random: b0
 block: list(1:1)
 covariance structure: identity
 serial correlation structure: identity
 variance function: identity
           Model Df    AIC    BIC  Loglik    Test Lik.Ratio P value
  ACH.fit2     1  7 115.92 132.14 -50.961
ACH.fit.b2     2  6 114.00 127.90 -50.998 1 vs. 2  0.072954 0.78708
```

F.10 Final model with treatment effect for β_1 and random effect for β_0

```
> summary(ACH.fit.b2)
Call:
  Model: LConc ~ ACH.fun0(b0, b1, b2, Time)
  Fixed: list(b0 ~ ., b1 ~ Treat, b2 ~ .)
 Random: list(b0 ~ .)
Cluster:  ~ Exp
   Data: ACH.data
Estimation Method: ML
Convergence at iteration: 4
Approximate Loglikelihood: -50.99761
AIC: 113.9952
BIC: 127.9001
Variance/Covariance Components Estimate(s):
  Structure: identity
  Standard Deviation(s) of Random Effect(s)
       b0
 0.4078341
 Cluster Residual Variance: 0.1747638

Fixed Effects Estimate(s):
                   Value Approx. Std.Error z ratio(C)
           b0 0.2097957         0.1292384   1.623323
b1.(Intercept) 3.1011070         0.3457005   8.970503
      b1.Treat 1.1981843         0.2762773   4.336889
           b2 2.7256382         0.2034873  13.394638

 Conditional Correlation(s) of Fixed Effects Estimates
                        b0 b1.(Intercept)    b1.Treat
b1.(Intercept)  0.05043104
      b1.Treat  0.11777574 -0.45685785
           b2  0.24982091  0.73505833     0.14456634

Random Effects (Conditional Modes):
           b0
 1  0.21993142
 2 -0.52346678
 3 -0.30898006
 4  0.12670761
 5 -0.57875095
 6  0.29423706
 7 -0.04340082
 8  0.17570625
 9  0.74520498
10 -0.45228819
11 -0.21855034
12  0.37502094
13  0.09016027
14 -0.04764009
15  0.20896586
16 -0.06285715
Standardized Population-Average Residuals:
       Min         Q1         Med        Q3      Max
 -2.550847 -0.4540633 -0.01007154 0.5351417 2.405168
Number of Observations: 75
Number of Clusters: 16
```

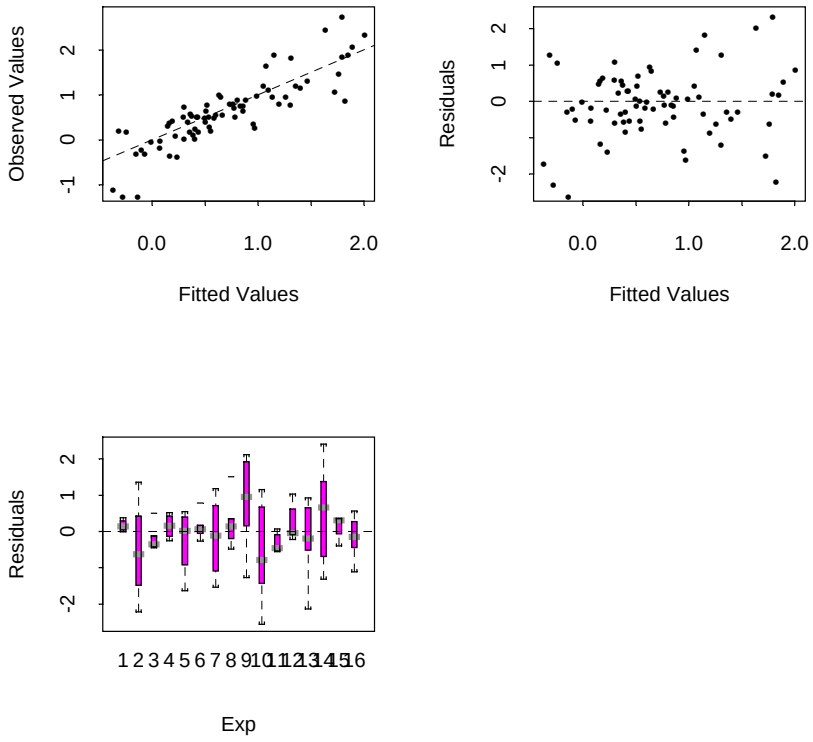

Figure F.3 Graphical assessment of final model.