José C. Pinheiro
Douglas M. Bates

Mixed-Effects Models in S and S-PLUS

Springer

José C. Pinheiro
Department of Biostatistics
Novartis Pharmaceuticals
One Health Plaza
East Hanover, NJ 07936-1080
USA
jose.pinheiro@pharma.novartis.com

Douglas M. Bates
Department of Statistics
University of Wisconsin
Madison, WI 53706-1685
USA
bates@stat.wisc.edu

Library of Congress Cataloging-in-Publication Data
Pinheiro, José C.
Mixed-effects models in S and S-PLUS/José C. Pinheiro, Douglas M. Bates
p. cm.—(Statistics and computing)
Includes bibliographical references and index.
ISBN 0-387-98957-9 (alk. paper)
I. Bates, Douglas M. II. Title. III. Series.
QA76.73.S15P56 2000
005.13'3—dc21 99-053566

Printed on acid-free paper.

Printed in the United States of America. (HAM/BP)

9 8 7

springer.com

6

Nonlinear Mixed-Effects Models: Basic Concepts and Motivating Examples

This chapter gives an overview of the nonlinear mixed-effects (NLME) model, introducing its main concepts and ideas through the analysis of real-data examples. The emphasis is on presenting the motivation for using NLME models when analyzing grouped data, while introducing some of the key features in the nlme library for fitting and analyzing such models. This chapter serves as an *appetizer* for the material covered in the last two chapters of the book: the theoretical foundations and computational methods for NLME models described in Chapter 7 and the nonlinear modeling facilities available in the nlme library, described in detail in Chapter 8.

6.1 LME Models vs. NLME Models

The first and possibly most important question about NLME models is *why would one want to use them?* This question, of course, also applies to nonlinear regression models in general as does the answer: *interpretability, parsimony,* and *validity beyond the observed range of the data.*

When choosing a regression model to describe how a response variable varies with covariates, one always has the option of using models, such as polynomial models, that are linear in the parameters. By increasing the order of a polynomial model, one can get increasingly accurate approximations to the true, usually nonlinear, regression function, *within the observed range of the data.* These *empirical* models are based only on the observed relationship between the response and the covariates and do not include

any theoretical considerations about the underlying mechanism producing the data.

Nonlinear models, on the other hand, are often *mechanistic*, i.e., based on a model for the mechanism producing the response. As a consequence, the model parameters in a nonlinear model generally have a natural physical interpretation. Even when derived empirically, nonlinear models usually incorporate known, theoretical characteristics of the data, such as asymptotes and monotonicity, and in these cases, can be considered as *semi-mechanistic* models. A nonlinear model generally uses fewer parameters than a competitor linear model, such as a polynomial, giving a more *parsimonious* description of the data. Nonlinear models also provide more reliable predictions for the response variable outside the observed range of the data than, say, polynomial models would.

To illustrate these differences between linear and nonlinear models, let us consider a simple example in which the expected height h_t of a tree at time t follows a three-parameter *logistic* growth model.

$$h_t = \phi_1 / \{1 + \exp[-(t - \phi_2)/\phi_3]\}. \tag{6.1}$$

As described in Appendix C.7, the parameters in (6.1) have a physical interpretation: ϕ_1 is the asymptotic height; ϕ_2 is the time at which the tree reaches half of its asymptotic height; and ϕ_3 is the time elapsed between the tree reaching half and $1/(1 + e^{-1}) \simeq 3/4$ of its asymptotic height. The logistic model (6.1) is linear in one parameter, ϕ_1, but nonlinear in ϕ_2 and ϕ_3.

To make the example more concrete, suppose that $\phi_1 = 3$, $\phi_2 = 1$, and $\phi_3 = 0.2$ and that we initially want to model the tree growth for $0.4 \leq t \leq 1.6$. The logistic curve, shown as a solid line in Figure 6.1, is approximated very well in the interval $[0.4, 1.6]$ by the fifth-degree polynomial

$$h_t \simeq -2.2911 + 16.591t - 44.411t^2 + 56.822t^3 - 31.514t^4 + 6.3028t^5$$

obtained as a least-squares fit to equally spaced t values in the interval $[0.4, 1.6]$. The polynomial fit, shown as a dashed line in Figure 6.1, is virtually indistinguishable from the logistic curve within this interval.

Unlike the coefficients in the logistic model, the coefficients in the polynomial approximation do not have any physical interpretation. Also, the linear polynomial model uses twice as many parameters as the logistic model to give comparable fitted values. Finally, the polynomial approximation is unreliable outside the interval $[0.4, 1.6]$. Figure 6.2, displaying the two curves over the extended interval $[0, 2]$, shows the dramatic differences between the curves outside the original range. We would expect growth curves to follow a pattern more like the logistic model than like the polynomial model.

Nonlinear mixed-effects models extend linear mixed-effects models by allowing the regression function to depend nonlinearly on fixed and random effects. Because of its greater flexibility, an NLME model is generally more

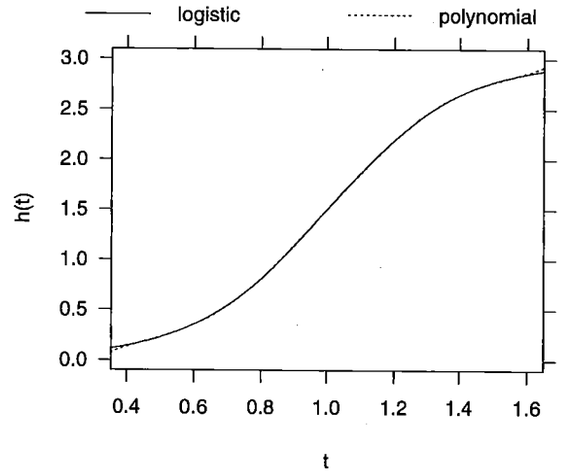

FIGURE 6.1. Logistic curve with parameters $\phi_1 = 3$, $\phi_2 = 1$, and $\phi_3 = 1.2$ and its fifth-order polynomial approximation over the interval $[0.4, 1.6]$.

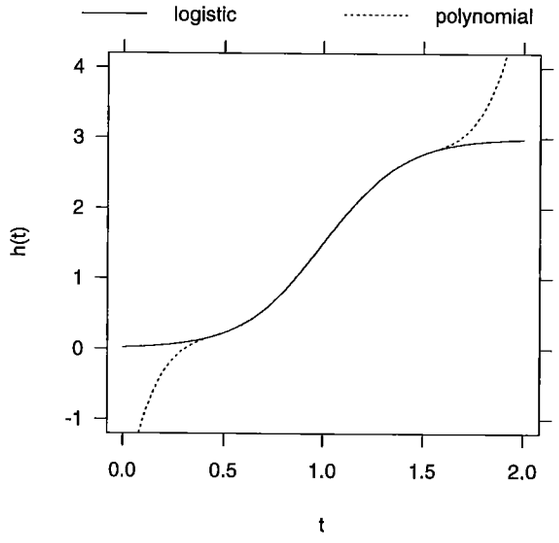

FIGURE 6.2. Logistic curve with parameters $\phi_1 = 3$, $\phi_2 = 1$, and $\phi_3 = 1.2$ and its fifth-order polynomial approximation over the interval $[0.4, 1.6]$, plotted over the interval $[0, 2]$.

interpretable and parsimonious than a competitor empirical LME model based, say, on a polynomial function. Also, the predictions obtained from an NLME model extend more reliably outside the observed range of the data.

The greater flexibility of NLME models does not come without cost, however. Because the random effects are allowed to enter the model nonlinearly, the marginal likelihood function, obtained by integrating the joint density of the response and the random effects with respect to the random effects, does not have a closed-form expression, as in the LME model. As a consequence, an approximate likelihood function needs to be used for the estimation of parameters, leading to more computationally intensive estimation algorithms and to less reliable inference results. These issues are described and discussed in detail in Chapter 7.

An important practical difference between NLME and LME models is that the former require starting estimates for the fixed-effects coefficients. Determining reasonable starting estimates for the parameters in a nonlinear model is somewhat of an art, although some general recommendations are available (Bates and Watts, 1988, §3.2). In many applications, the same nonlinear model is to be used several times with similar datasets. In these cases, it is worthwhile to program the steps used to obtain starting estimates into a function, which can then be used to produce starting estimates from many datasets. Such *self-starting* nonlinear models are described and illustrated in §8.1.2.

There are far more similarities than differences between LME and NLME models. Both models are used with grouped data and serve the same purpose: to describe a response variable as a function of covariates, taking into account the correlation among observations in the same group. Random effects are used to represent within-group dependence in both LME and NLME models, and the assumptions about the random effects and the within-group errors are identical in the two models.

The same "inside-out" model building strategy used for LME models in Part I is used here with NLME models: whenever feasible, we begin by getting separate fits of a model for each group, then examining these individual fits to see which coefficients appear to be common to all groups and which coefficients seem to vary among groups. We then proceed to fit an overall mixed-effects model to the data, using random-effects terms if they seem warranted. Diagnostics plots are then used to assess the model's assumptions and to decide on refinements of the initial model.

The S methods for displaying, plotting, comparing, and updating nlme fitted objects will look very familiar to the readers of Part I. Because of the similarities between LME and NLME models, most of the lme methods described in §4.2.1 can be used, without changes, with nlme objects. The nonlinear mixed-effects models modeling functions and methods included in the nlme library are described and illustrated in detail in Chapter 8. The examples in the next sections serve to illustrate some of the basic concepts

of NLME models and to introduce the capabilities provided by the nlme library for analyzing them.

6.2 Indomethacin Kinetics

The data for our first example come from a laboratory study on the pharmacokinetics of the drug indomethacin (Kwan, Breault, Umbenhauer, McMahon and Duggan, 1976). Six human volunteers received bolus intravenous injections of the same dose of indomethacin and had their plasma concentrations of the drug (in mcg/ml) measured 11 times between 15 minutes and 8 hours postinjection. These data are described in more detail in Appendix A.12 and are also analyzed in Davidian and Giltinan (1995, §2.1). The indomethacin data are included in the nlme library as the groupedData object `Indometh`.

The plot of the `Indometh` data, obtained with

```
> plot( Indometh )                                    # Figure 6.3
```

and displayed in Figure 6.3, reveals a familiar pattern in plots of grouped data: the concentration curves have a similar shape, but differ among individuals. As indicated in Figure 6.3, these data are balanced, as the serum concentrations were measured at the same time points for all six individuals.

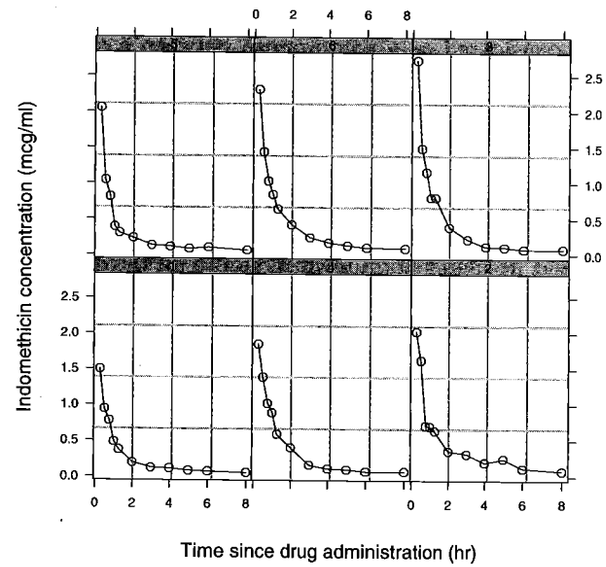

FIGURE 6.3. Concentration of indomethacin over time for six subjects following intravenous injection.

A common method of modeling pharmacokinetic data is to represent the human body as a system of compartments in which the drug is transferred according to first-order kinetics (Gibaldi and Perrier, 1982). In such a *compartment model* the concentration of the drug over time in the different compartments is determined by a linear system of differential equations, whose solution can be expressed as a linear combination of exponential terms. The mechanistic model for the indomethacin concentration is from a two-compartment model. It expresses the expected concentration $\mathrm{E}(y_t)$ at time t as a linear combination of two exponentials

$$\mathrm{E}(y_t) = \phi_1 \exp(-\phi_2 t) + \phi_3 \exp(-\phi_4 t), \quad \phi_2 > 0,\ \phi_4 > 0. \tag{6.2}$$

The biexponential model (6.2) is not *identifiable*, in the sense of having a unique vector of parameters associated with a given set of predictions, because the parameters in the two exponential terms may be exchanged without changing the predictions. Identifiability is ensured by requiring that $\phi_2 > \phi_4$ so that the first exponential term determines the initial elimination phase of the drug (the sharp decreases in the individual curves of Figure 6.3). The terminal elimination phase is primarily determined by the second exponential term.

Model (6.2) is linear in multipliers ϕ_1 and ϕ_3, but nonlinear in the rate constants ϕ_2 and ϕ_4. Because the rate constants must be positive to be physically meaningful, we reparameterize (6.2) in terms of the log-rate constants $\phi_2' = \log \phi_2$ and $\phi_4' = \log \phi_4$.

When introducing LME models in §1.1 we demonstrated that the LME model can be considered as a compromise between a single linear model that ignores the grouping in the data and a linear model that provides separate fits for each of the groups. These linear models were fit using `lm` and `lmList`, respectively. To illustrate why NLME models are useful for datasets like the indomethacin data, we will repeat that development using the nonlinear model-fitting functions `nls` and `nlsList`.

If we ignore the grouping of the concentration measurements according to individual and fit a single nonlinear model to all the data we express the indomethacin concentration y_{ij} in individual i at time t_j is

$$y_{ij} = \phi_1 \exp[-\exp(\phi_2')\, t_j] + \phi_3 \exp[-\exp(\phi_4')\, t_j] + \epsilon_{ij}, \tag{6.3}$$

where the error terms ϵ_{ij} are assumed to be independently distributed as $\mathcal{N}(0, \sigma^2)$. Nonlinear regression models with independent, identically distributed Gaussian errors, like (6.3), are fit by *nonlinear least squares* (Bates and Watts, 1988; Seber and Wild, 1989), as implemented in the S function `nls` (Bates and Chambers, 1992).

Starting values for the parameters in the biexponential model are usually obtained through the method of *peeling*, as described in Appendix C.4.1. However, we need not be concerned with obtaining initial estimates for the biexponential model parameters, as the self-starting model function

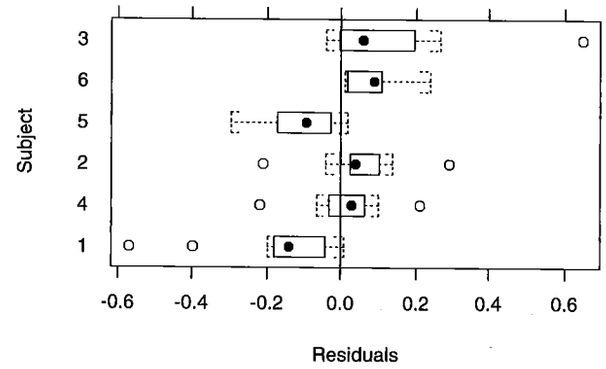

FIGURE 6.4. Boxplots of residuals by subject for `fm1Indom.nls`, a nonlinear least squares fit of a two-compartment model to the indomethacin data ignoring the subject dependence.

`SSbiexp`, described in Appendix C.4, produces them automatically from the data. A call to `nls` to fit the biexponential model (6.3) using a self-starting function is almost as simple as a call to `lm` to fit a linear model.

```
> fm1Indom.nls <- nls( conc ~ SSbiexp(time, A1, lrc1, A2, lrc2),
+   data = Indometh )
> summary(fm1Indom.nls)

Formula: conc ~ SSbiexp(time, A1, lrc1, A2, lrc2)

Parameters:
      Value Std. Error  t value
A1  2.77342    0.25323  10.9522
lrc1 0.88627    0.22222   3.9882
A2  0.60663    0.26708   2.2713
lrc2 -1.09209    0.40894  -2.6705

Residual standard error: 0.174489 on 62 degrees of freedom
. . .
> plot( fm1Indom.nls, Subject ~ resid(.), abline = 0 ) # Figure 6.4
```

The correspondence between the parameters in `fm1Indom.nls` and in (6.3) is $\phi_1 =$ `A1`, $\phi_2' =$ `lrc1`, $\phi_3 =$ `A2`, $\phi_4' =$ `lrc2`.

The boxplots of the residuals by individual in Figure 6.4 are similar to those from the `lm` fit to the `Rails` data (Figure 1.2). In Figure 6.4 the residuals tend to be mostly negative for some subjects and mostly positive for others although the pattern is not as pronounced as that in Figure 1.2. Because a single concentration curve is used for all subjects, the individual differences noticed in Figure 6.3 are incorporated in the residuals, thus inflating the residual standard error. Probably the most important drawback of using an `nls` model with grouped data is that it prevents us from

understanding the true structure of the data and from considering different sources of variability that are of interest in themselves. For example, in the indomethacin study, an important consideration in determining an adequate therapeutic regime for the drug is knowing how the concentration profiles vary among individuals.

To fit a separate biexponential model to each subject, thus allowing the individual effects to be incorporated in the parameter estimates, we express the model as

$$y_{ij} = \phi_{1i} \exp\left[-\exp\left(\phi'_{2i}\right) t_j\right] + \phi_{3i} \exp\left[-\exp\left(\phi'_{4i}\right) t_j\right] + \epsilon_{ij}, \qquad (6.4)$$

where, as before, the ϵ_{ij} are independent $\mathcal{N}(0, \sigma^2)$ errors and use the `nlsList` function

```
> fm1Indom.lis <- nlsList(conc ~ SSbiexp(time, A1, lrc1, A2, lrc2),
+   data = Indometh )
> fm1Indom.lis
Call:
  Model: conc ~ SSbiexp(time, A1, lrc1, A2, lrc2) | Subject
   Data: Indometh

Coefficients:
      A1     lrc1      A2      lrc2
1 2.0293 0.57938 0.19154 -1.78783
4 2.1979 0.24249 0.25481 -1.60153
2 2.8277 0.80143 0.49903 -1.63508
5 3.5663 1.04095 0.29170 -1.50594
6 3.0022 1.08811 0.96840 -0.87324
3 5.4677 1.74968 1.67564 -0.41226

Degrees of freedom: 66 total; 42 residual
Residual standard error: 0.07555
```

We can see that there is considerable variability in the individual parameter estimates and that the residual standard error is less than one-half that from the `nls` fit. The boxplots of the residuals by subject, shown in Figure 6.5, indicate that the individual effects have been accounted for in the fitted `nlsList` model.

The `nlsList` model is at the other extreme of the flexibility spectrum compared to the `nls` model: it uses 24 coefficients to represent the individual concentration profiles and does not take into account the obvious similarities among the individual curves, indicated in Figure 6.3. The `nlsList` model is useful when one is interested in modeling the behavior of a particular, fixed set of individuals, but it is not adequate when the observed individuals are to be treated as a sample from a population of similar individuals, which constitutes the majority of applications involving grouped data. In this case, the interest is in estimating the average behavior of an

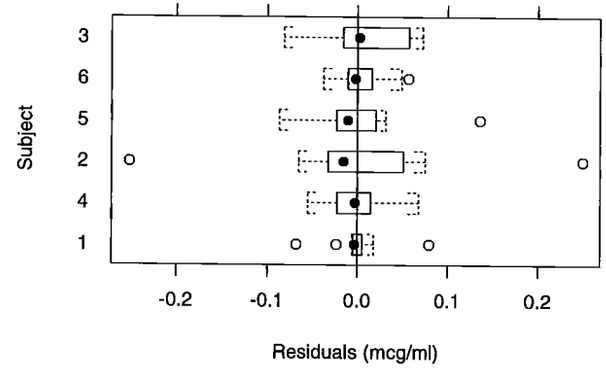

FIGURE 6.5. Boxplots of residuals by subject for `fm1Indom.lis`, a set of nonlinear regression fits of a two-compartment model to the indomethacin data where each subject's data is fit separately.

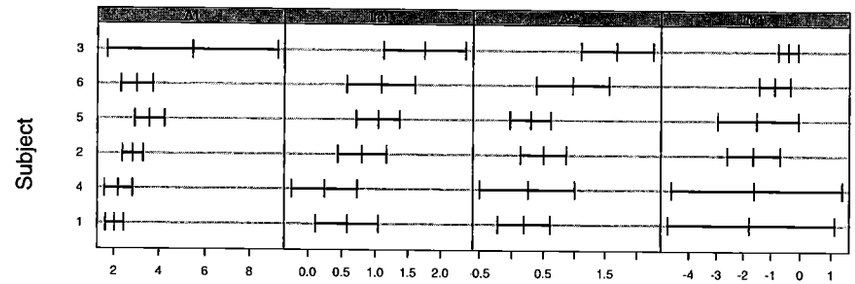

FIGURE 6.6. Ninety-five percent confidence intervals on the biexponential model parameters for each individual in the indomethacin data.

individual in the population and the variability among and within individuals, which is precisely what mixed-effects models are designed to do.

The plot of the individual confidence intervals for the coefficients in the `nlsList` model, shown in Figure 6.6, gives a better idea about their variability among subjects.

```
> plot( intervals(fm1Indom.lis) )                    # Figure 6.6
```

The terminal phase log-rate constants, ϕ_{4i}, do not seem to vary substantially among individuals, but the remaining parameters do.

Recall that in `lmList` fits to balanced data, the lengths of the confidence intervals on a parameter were the same for all the groups (see Figure 1.12, p. 33, or Figure 1.13, p. 34). This does not occur in an `nlsList` fit because the approximate standard errors used to produce the confidence intervals in a nonlinear least squares fit depend on the parameter estimates (Seber and Wild, 1989, §5.1).

To introduce the concepts of fixed and random effects in a nonlinear mixed-effects model, it is useful to re-express the model (6.4) as

$$y_{ij} = \left[\bar{\phi}_1 + (\phi_{1i} - \bar{\phi}_1)\right] \exp\left\{-\exp\left[\bar{\phi}'_2 + (\phi'_{2i} - \bar{\phi}'_2)\right] t_j\right\} + \left[\bar{\phi}_3 + (\phi_{3i} - \bar{\phi}_3)\right] \exp\left\{-\exp\left[\bar{\phi}'_4 + (\phi'_{4i} - \bar{\phi}'_4)\right] t_j\right\} + \epsilon_{ij}, \quad (6.5)$$

where $\bar{\phi}$ denotes the average of the individual parameters. The `nlsList` model treats the deviations of the individual coefficients from their mean as parameters to be estimated. Mixed-effects models, on the other hand, represent these deviations from the mean value of the coefficients (the *fixed effects*) as *random effects*, treating the individuals as a sample from a population. The nonlinear mixed-effects version of (6.5) is

$$y_{ij} = (\beta_1 + b_{1i}) \exp\left[-\exp(\beta_2 + b_{2i})\, t_j\right] + (\beta_3 + b_{3i}) \exp\left[-\exp(\beta_4 + b_{4i}))\, t_j\right] + \epsilon_{ij}. \quad (6.6)$$

The fixed effects β_1, β_2, β_3, and β_4 represent the mean values of the parameters in the population of individuals. The individual deviations are represented by the random effects b_{1i}, b_{2i}, b_{3i}, and b_{4i}, which are assumed to be distributed normally with mean $\mathbf{0}$ and variance–covariance matrix $\boldsymbol{\Psi}$. Random effects corresponding to different individuals are assumed to be independent. The within-group errors ϵ_{ij} are assumed to be independently distributed as $\mathcal{N}(0, \sigma^2)$ and to be independent of the random effects.

The nonlinear mixed-effects model (6.6) gives a compromise between the rigid `nls` model (6.3) and the overparameterized `nlsList` model (6.4). It accommodates individual variations through the random effects, but ties the different individuals together through the fixed effects and the variance–covariance matrix $\boldsymbol{\Psi}$.

A crucial step in the model-building of mixed-effects models is deciding which of the coefficients in the model need random effects to account for their between-subject variation and which can be treated as purely fixed effects. Plots of individual confidence intervals obtained from an `nlsList` fit, like the one shown in Figure 6.6, are often useful for that purpose. In the case of the indomethacin data, the individual confidence intervals suggest that the b_{4i} random effect for the terminal elimination phase log-rate constant in (6.6) is not needed.

An alternative model-building strategy is to start with a model with random effects for all parameters and then examine the fitted object to decide which, if any, of the random effects can be eliminated from the model. One problem with this approach is that, when a general positive-definite structure is assumed for the random effects variance–covariance matrix $\boldsymbol{\Psi}$, the number of parameters to estimate increases with the square of the number of random effects. In cases where the number of random effects is large relative to the number of individuals, as in the indomethacin data, it is generally recommended to use a diagonal $\boldsymbol{\Psi}$ initially, to prevent

convergence problems with an overparameterized model. We apply this approach to the indomethacin data.

```
> fm1Indom.nlme <- nlme( fm1Indom.lis,
+   random = pdDiag(A1 + lrc1 + A2 + lrc2 ~ 1) )
> fm1Indom.nlme
Nonlinear mixed-effects model fit by maximum likelihood
  Model: conc ~ SSbiexp(time, A1, lrc1, A2, lrc2)
  Data: Indometh
  Log-likelihood: 54.592
  Fixed: list(A1 ~ 1, lrc1 ~ 1, A2 ~ 1, lrc2 ~ 1)
     A1   lrc1      A2   lrc2
 2.8276 0.7733 0.46104 -1.345

Random effects:
 Formula: list(A1 ~ 1, lrc1 ~ 1, A2 ~ 1, lrc2 ~ 1)
 Level: Subject
 Structure: Diagonal
             A1    lrc1      A2       lrc2 Residual
StdDev: 0.57136 0.15811 0.11154 7.2051e-11 0.081496

Number of Observations: 66
Number of Groups: 6
```

The `nlme` function extracts the information about the model to fit, the parameters to estimate, and the starting estimates for the fixed effects from the `fm1Indom.lis` object.

The near-zero estimate for the standard deviation of the `lrc2` random effect suggests that this term could be dropped from the model. The remaining estimated standard deviations suggest that the other random effects should be kept in the model. We can test if the `lrc2` random effect can be removed from the model by updating the fit and using `anova`.

```
> fm2Indom.nlme <- update( fm1Indom.nlme,
+   random = pdDiag(A1 + lrc1 + A2 ~ 1) )
> anova( fm1Indom.nlme, fm2Indom.nlme )
             Model df      AIC      BIC logLik   Test    L.Ratio
fm1Indom.nlme     1  9 -91.185 -71.478 54.592
fm2Indom.nlme     2  8 -93.185 -75.668 54.592 1 vs 2 6.2637e-06
             p-value
fm1Indom.nlme
fm2Indom.nlme   0.998
```

The two fits give nearly identical log-likelihoods, confirming that `lrc2` can be treated as a purely fixed effect.

To further explore the variance–covariance structure of the random effects that are left in `fm2Indom.nlme`, we update the fit using a general positive-definite $\boldsymbol{\Psi}$ matrix.

```
> fm3Indom.nlme <- update( fm2Indom.nlme, random = A1+lrc1+A2 ~ 1 )
```

```
> fm3Indom.nlme
. . .
Random effects:
 Formula: list(A1 ~ 1, lrc1 ~ 1, A2 ~ 1)
 Level: Subject
 Structure: General positive-definite
            StdDev  Corr
      A1 0.690406 A1    lrc1
    lrc1 0.179030 0.932
      A2 0.153669 0.471 0.118
Residual 0.078072
. . .
```

The large correlation between the `A1` and `lrc1` random effects and the small correlation between these random effects and the `A2` random effect suggest that a block-diagonal Ψ could be used to represent the variance–covariance structure of the random effects.

```
> fm4Indom.nlme <- update( fm3Indom.nlme,
+   random = pdBlocked(list(A1 + lrc1 ~ 1, A2 ~ 1)) )
> anova( fm3Indom.nlme, fm4Indom.nlme )
              Model df      AIC      BIC logLik   Test L.Ratio
fm3Indom.nlme     1 11 -94.945 -70.859 58.473
fm4Indom.nlme     2  9 -97.064 -77.357 57.532 1 vs 2  1.8809
              p-value
fm3Indom.nlme
fm4Indom.nlme  0.3904
```

The large p-value for the likelihood ratio test and the smaller values for the AIC and BIC corroborate the block-diagonal variance–covariance structure. Allowing the `A1` and `lrc1` random effects to be correlated causes a significant improvement in the log-likelihood.

```
> anova( fm2Indom.nlme, fm4Indom.nlme )
              Model df      AIC      BIC logLik   Test L.Ratio
fm2Indom.nlme     1  8 -93.185 -75.668 54.592
fm4Indom.nlme     2  9 -97.064 -77.357 57.532 1 vs 2  5.8795
              p-value
fm2Indom.nlme
fm4Indom.nlme  0.0153
```

The plot of the standardized residuals versus the fitted values corresponding to `fm4Indom.nlme`, presented in Figure 6.7, does not indicate any departures from the NLME model assumptions, except for two possible outlying observations for Individual 2.

```
> plot( fm4Indom.nlme, id = 0.05, adj = -1 )          # Figure 6.7
```

No significant departures from the assumption of normality for the within-group errors is observed in the normal probability plot of the standardized residuals of `fm4Indom.nlme`, shown in Figure 6.8.

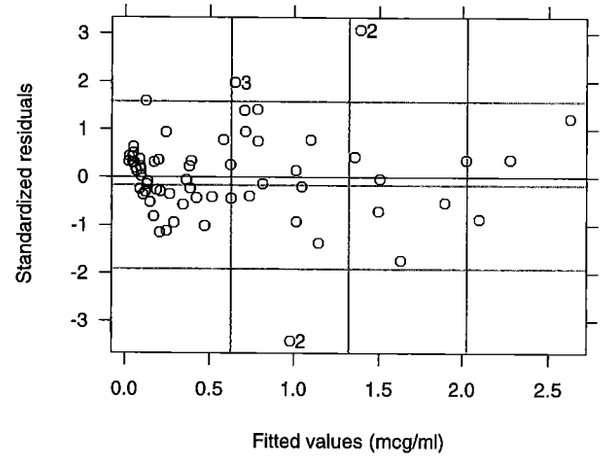

FIGURE 6.7. Scatter plot of standardized residuals versus fitted values for `fm4Indom.nlme`.

```
> qqnorm( fm4Indom.nlme )                             # Figure 6.8
```

A final assessment of the adequacy of the `fm4Indom.nlme` model is given by the plot of the augmented predictions in Figure 6.9. For comparison, and to show how individual effects are accounted for in the NLME model, both the population predictions (corresponding to random effects equal to zero) and the within-group predictions (obtained using the estimated random effects) are displayed.

```
> plot( augPred(fm4Indom.nlme, level = 0:1) )         # Figure 6.9
```

Note that the within-group predictions are in close agreement with the observed concentrations, illustrating that the NLME model can accommodate individual effects.

We conclude that `fm4Indom.nlme` provides a good representation of the concentration profiles in the indomethacin data. Its summary

```
> summary( fm4Indom.nlme )
Nonlinear mixed-effects model fit by maximum likelihood
  Model: conc ~ SSbiexp(time, A1, lrc1, A2, lrc2)
 Data: Indometh
      AIC      BIC logLik
  -97.064 -77.357 57.532

Random effects:
 Composite Structure: Blocked

 Block 1: A1, lrc1
 Formula: list(A1 ~ 1, lrc1 ~ 1)
```

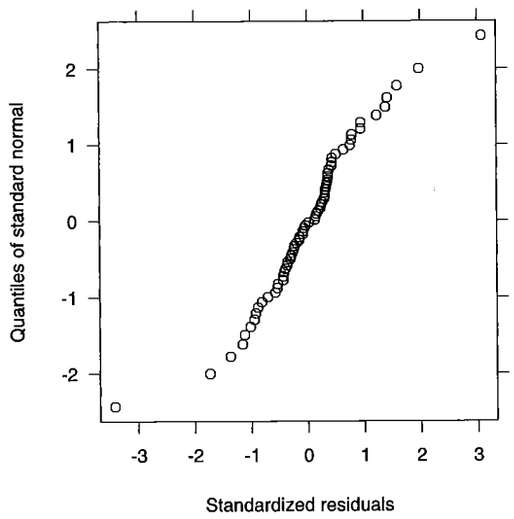

FIGURE 6.8. Normal plot of standardized residuals for the `fm4Indom.nlme` `nlme` fit.

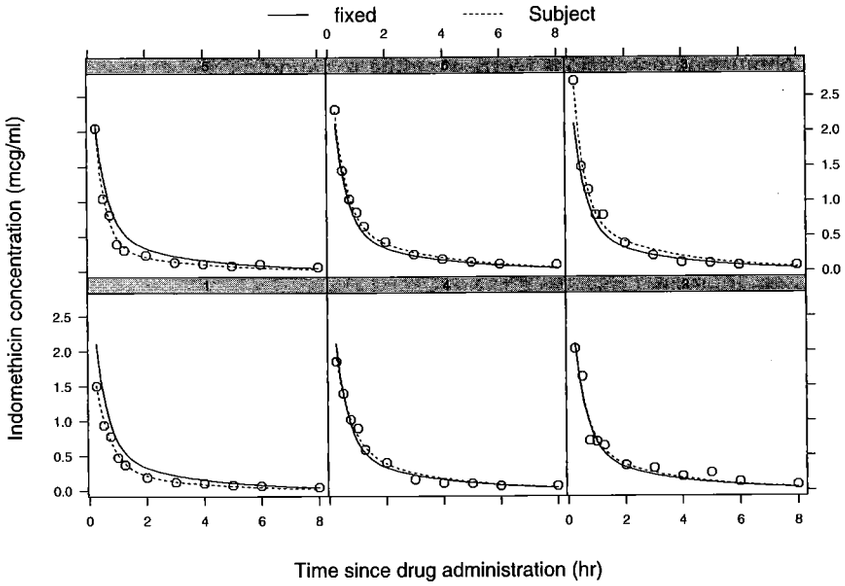

FIGURE 6.9. Population predictions (`fixed`), within-group predictions (`Subject`), and observed concentrations of indomethacin (circles) versus time since injection for `fm4Indom.nlme`.


```
 Level: Subject
 Structure: General positive-definite
      StdDev  Corr
  A1 0.69496 A1
lrc1 0.17067 0.905

 Block 2: A2
 Formula: A2 ~ 1 | Subject
            A2 Residual
StdDev: 0.18344 0.078226

Fixed effects: list(A1 ~ 1, lrc1 ~ 1, A2 ~ 1, lrc2 ~ 1)
       Value Std.Error DF t-value p-value
  A1  2.8045   0.31493 57  8.9049  <.0001
lrc1  0.8502   0.11478 57  7.4067  <.0001
  A2  0.5887   0.13321 57  4.4195  <.0001
lrc2 -1.1029   0.16954 57 -6.5054  <.0001
. . .
```

shows that the fixed-effects estimates are similar to the parameter estimates in the `nls` fit `fm1Indom.nls`. The approximate standard errors for the fixed effects are substantially different from and, except for `A1`, considerably smaller than those from the `nls` fit. The estimated within-group standard error is slightly larger than the residual standard error in the `nlsList` fit `fm1Indom.lis`.

6.3 Growth of Soybean Plants

The example discussed in this section illustrates an important area of application of NLME models: growth curve data. It also introduces the concept of using covariates to explain between-group variability in NLME models.

The soybean data, displayed in Figure 6.10, are described in Davidian and Giltinan (1995, §1.1.3, p. 7) as "Data from an experiment to compare growth patterns of two genotypes of soybeans: Plant Introduction #416937 (P), an experimental strain, and Forrest (F), a commercial variety." The average leaf weight (in grams) of six plants chosen at random from each plot was measured at approximately weekly intervals, between two and eleven weeks after planting. The experiment was carried out over three different planting years: 1988, 1989, and 1990. Eight plots were planted with each genotype in each planting year, giving a total of forty-eight plots in the study. These data are available in the nlme library as the groupedData object `Soybean` and are also described in Appendix A.27.

```
> Soybean[1:3, ]
Grouped Data: weight ~ Time | Plot
```

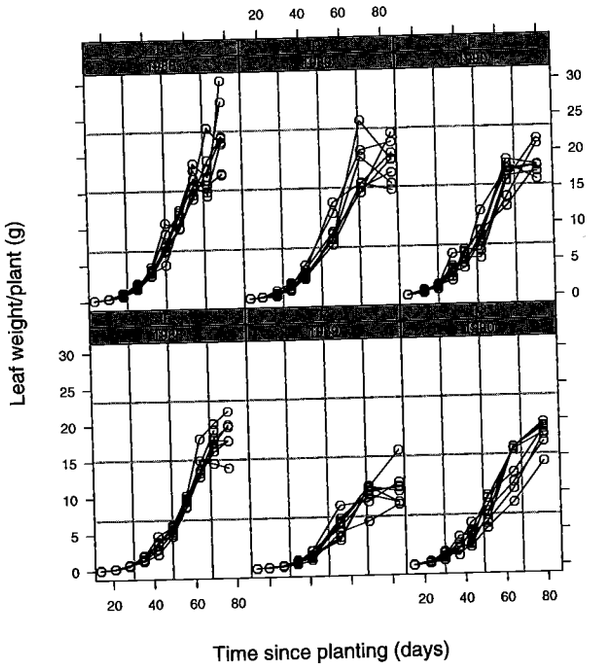

FIGURE 6.10. Average leaf weight per plant of two genotypes of soybean versus time since planting, over three different planting years. Within each year data were obtained on eight plots of each variety of soybean.

```
    Plot Variety Year Time weight
1 1988F1       F 1988   14  0.106
2 1988F1       F 1988   21  0.261
3 1988F1       F 1988   28  0.666
> plot( Soybean, outer = ~ Year * Variety )          # Figure 6.10
```

The average leaf weight per plant in each plot is measured the same number of times, but at different times, making the data unbalanced.

There is considerable variation in the growth curves among plots, but the same overall S-shaped pattern is observed for all plots. This nonlinear growth pattern is well described by the three parameter logistic model (6.1), introduced in §6.1. The self-starting function `SSlogis`, described in Appendix C.7, can be used to automatically generate starting estimates for the parameters in an `nlsList` fit.

```
> fm1Soy.lis <- nlsList( weight ~ SSlogis(Time, Asym, xmid, scal),
+   data = Soybean )
Error in nls(y ~ 1/(1 + exp((xmid - x)/scal)..: singular gradient
        matrix
Dumped
```



```
> fm1Soy.lis
Call:
  Model: weight ~ SSlogis(Time, Asym, xmid, scal) | Plot
   Data: Soybean

Coefficients:
          Asym    xmid    scal
1988F4 15.1513  52.834  5.1766
1988F2 19.7455  56.575  8.4067
. . .
1989P8     NA      NA      NA
. . .
1990P5 19.5438  51.148  7.2920
1990P2 25.7873  62.360 11.6570

Degrees of freedom: 404 total; 263 residual
Residual standard error: 1.0438
```

The error message from `nls` indicates that convergence was not attained for one of the plots, `1989P8`. The `nlsList` function is able to recover from such nonconvergence problems and carry on with subsequent `nls` fits. Missing values (`NA`) are assigned to the coefficients of the nonconverging fits. The coefficients in `fm1Soy.lis` are related to the logistic model parameters as follows: $\phi_1 =$ `Asym`, $\phi_2 =$ `xmid`, and $\phi_3 =$ `scal`

Analysis of the individual confidence intervals for `fm1Soy.lis` suggests that random effects are needed for all of the parameters in the logistic model. The corresponding nonlinear mixed-effects model for the average leaf weight per plant y_{ij} in plot i at t_{ij} days after planting is

$$\begin{aligned} y_{ij} &= \frac{\phi_{1i}}{1+\exp\left[-\left(t_{ij}-\phi_{2i}\right)/\phi_{3i}\right]} + \epsilon_{ij}, \\ \boldsymbol{\phi}_i &= \begin{bmatrix}\phi_{1i}\\ \phi_{2i}\\ \phi_{3i}\end{bmatrix} = \begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\end{bmatrix} + \begin{bmatrix}b_{1i}\\ b_{2i}\\ b_{3i}\end{bmatrix} = \boldsymbol{\beta} + \boldsymbol{b}_i, \\ \boldsymbol{b}_i &\sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Psi}\right), \quad \epsilon_{ij} \sim \mathcal{N}\left(0, \sigma^2\right). \end{aligned} \tag{6.7}$$

The fixed effects $\boldsymbol{\beta}$ represent the mean values of the parameters $\boldsymbol{\phi}_i$ in the population and the random effects $\boldsymbol{b}_i$ represent the deviations of the $\boldsymbol{\phi}_i$ from their mean values. The random effects are assumed to be independent for different plots and the within-group errors ϵ_{ij} are assumed to be independent for different i, j and to be independent of the random effects.

Because the number of plots in the soybean data, 48, is large compared to the number of random effects in (6.7), we use a general positive-definite $\boldsymbol{\Psi}$ for the initial NLME model. Because we can extract information about the model, the parameters to estimate, and the starting values for the fixed effects from the `fm1Soy.lis` object we can fit model (6.7) with the simple call

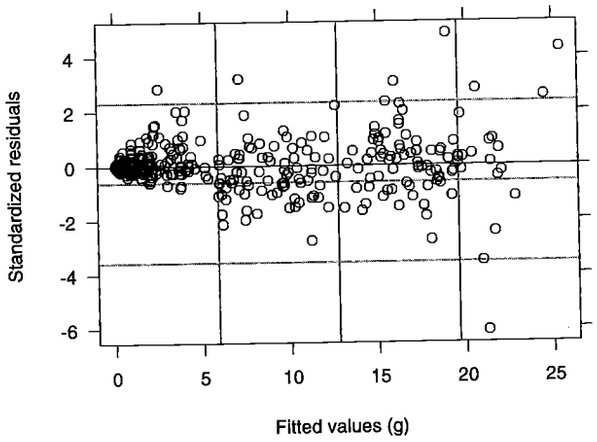

FIGURE 6.11. Scatter plot of standardized residuals versus fitted values for `fm1Soy.nlme`.

```
> fm1Soy.nlme <- nlme( fm1Soy.lis )
> fm1Soy.nlme
Nonlinear mixed-effects model fit by maximum likelihood
  Model: weight ~ SSlogis(Time, Asym, xmid, scal)
  Data: Soybean
  Log-likelihood: -739.84
  Fixed: list(Asym ~ 1, xmid ~ 1, scal ~ 1)
   Asym  xmid   scal
 19.253 55.02 8.4033

Random effects:
 Formula: list(Asym ~ 1, xmid ~ 1, scal ~ 1)
 Level: Plot
 Structure: General positive-definite
         StdDev  Corr
    Asym 5.2011 Asym  xmid
    xmid 4.1974 0.721
    scal 1.4047 0.711 0.958
Residual 1.1235

Number of Observations: 412
Number of Groups: 48
```

The plot of the standardized residuals versus the fitted values in Figure 6.11 shows a pattern of increasing variability for the within-group errors. We model the within-group heteroscedasticity using the *power* variance function, described in §5.2.1 and represented in S by the varPower class.

```
> fm2Soy.nlme <- update( fm1Soy.nlme, weights = varPower() )
```


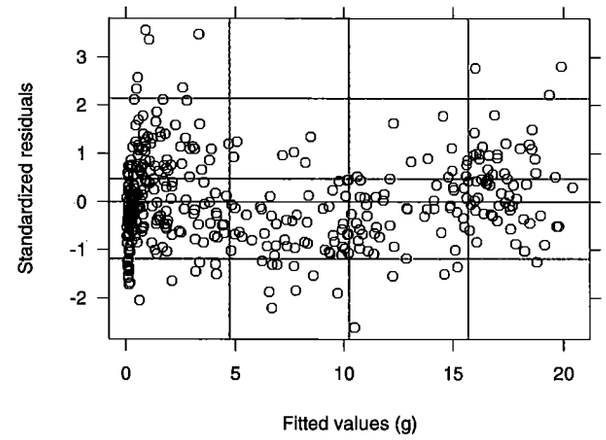

FIGURE 6.12. Standardized residuals versus fitted values in the `nlme` fit of the soybean data, with heteroscedastic error.

```
> anova( fm1Soy.nlme, fm2Soy.nlme )
            Model df    AIC    BIC  logLik   Test L.Ratio p-value
fm1Soy.nlme     1 10 1499.7 1539.9 -739.84
fm2Soy.nlme     2 11  737.3  781.6 -357.66 1 vs 2  764.35  <.0001
```

The heteroscedastic model provides a much better representation of the data. We can assess the adequacy of the *power* variance function by again plotting the standardized residuals against the fitted values as in Figure 6.12. The power variance function seems to model adequately the within-plot heteroscedasticity.

The primary question of interest for the soybean data is the possible relationship between the growth pattern of the soybean plants and the experimental factors `Variety` and `Year`. Plots of estimates of the random effects are useful for exploring the relationship between the individual growth patterns and the experimental factors.

```
> plot(ranef(fm2Soy.nlme, augFrame = T),
+      form = ~ Year * Variety, layout = c(3,1))       # Figure 6.13
```

In Figure 6.13 all three parameters seem to vary with year and variety. It appears that the asymptote (`Asym`) and the scale (`scal`) are larger for the `P` variety than for the `F` variety and that this difference is more pronounced in 1989. The time at which half of the asymptotic leaf weight is attained (`xmid`) appears to be smaller for the `P` variety than for the `F` variety.

The `fixed` argument to `nlme` allows linear modeling of parameters with respect to covariates. For example, we can model the dependence of all three parameters on `Year` with

```
> soyFix <- fixef( fm2Soy.nlme )
```

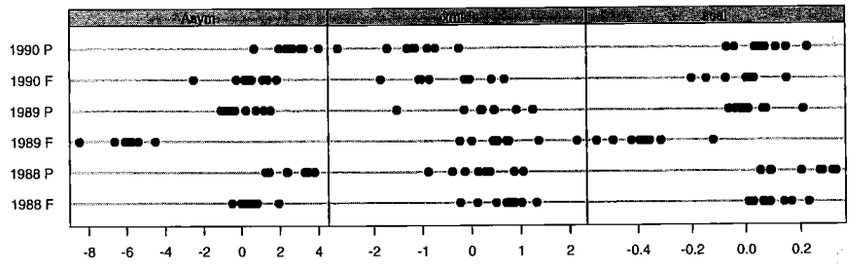

FIGURE 6.13. Estimates of the random effects by `Year` and `Variety` in the `nlme` fit of the soybean data.

```
> options( contrasts = c("contr.treatment", "contr.poly") )
> fm3Soy.nlme <- update( fm2Soy.nlme,
+   fixed = Asym + xmid + scal ~ Year,
+   start = c(soyFix[1], 0, 0, soyFix[2], 0, 0, soyFix[3], 0, 0) )
> fm3Soy.nlme
. . .
  Log-likelihood: -325.02
  Fixed: Asym + xmid + scal ~ Year
 Asym.(Intercept) Asym.Year1989 Asym.Year1990 xmid.(Intercept)
           20.222       -6.3775       -3.4995           54.118
 xmid.Year1989 xmid.Year1990 scal.(Intercept) scal.Year1989
       -2.4696       -4.8764           8.0515      -0.93374
 scal.Year1990
      -0.66884

Random effects:
 Formula: list(Asym ~ 1, xmid ~ 1, scal ~ 1)
 Level: Plot
 Structure: General positive-definite
                     StdDev   Corr
Asym.(Intercept) 2.3686896 Asy(I) xmd(I)
xmid.(Intercept) 0.5863454 -0.997
scal.(Intercept) 0.0043059 -0.590  0.652
        Residual 0.2147634

Variance function:
 Structure: Power of variance covariate
 Formula:  ~ fitted(.)
 Parameter estimates:
   power
 0.95187
. . .
```


The default parameterization for contrasts is changed to `contr.treatment` to allow easier interpretation of the fixed effects included in the model: they represent differences from the year 1988. Note that, because the fixed-effects model has changed, new starting values for the fixed effects must be provided.

We can assess the significance of `Year` for the fixed effects model using `anova` with a single argument.

```
> anova( fm3Soy.nlme )
                numDF denDF F-value p-value
Asym.(Intercept)     1   356   402.4  <.0001
       Asym.Year     2   356   105.0  <.0001
xmid.(Intercept)     1   356  9641.2  <.0001
       xmid.Year     2   356    10.2  <.0001
scal.(Intercept)     1   356  8378.5  <.0001
       scal.Year     2   356    11.7  <.0001
```

As suggested by Figure 6.13, `Year` has a very significant effect on the growth pattern of the soybean plants.

The estimated standard deviation for the `scal` random effect in the `fm3Soy.nlme` fit is only 0.004, corresponding to an estimated coefficient of variation with respect to the `scal.(Intercept)` fixed effect of only 0.05%. This suggests that `scal` can be treated as a purely fixed effect. When we do refit the model dropping the `scal` random effect, we get a p-value of 0.99 in the likelihood ratio test. It often happens that creating a better-fitting model for the fixed effects, by including their dependence on covariates, reduces the need for random-effects terms. In these cases, the between-group parameter variation is mostly being explained by the covariates included in the model.

Proceeding sequentially in the model-building process by examining plots of the estimated random effects against the experimental factors, testing for the inclusion of covariates and for the elimination of random effects, we end up with the following model, in which the only random effect is that for `Asym`.

```
> summary( fm4Soy.nlme )
. . .
     AIC    BIC  logLik
  616.32 680.66 -292.16

Random effects:
 Formula: Asym ~ 1 | Plot
        Asym.(Intercept) Residual
StdDev:           1.0349  0.21804

Variance function:
 Structure: Power of variance covariate
 Formula:  ~ fitted(.)
```

```
Parameter estimates:
 power
0.9426
Fixed effects: list(Asym ~ Year * Variety, xmid ~ Year + Variety,
                    scal ~ Year)
                     Value Std.Error  DF t-value p-value
    Asym.(Intercept)  19.434    0.9537 352  20.379  <.0001
       Asym.Year1989  -8.842    1.0719 352  -8.249  <.0001
       Asym.Year1990  -3.707    1.1768 352  -3.150  0.0018
        Asym.Variety   1.623    1.0380 352   1.564  0.1188
Asym.Year1989Variety   5.571    1.1704 352   4.760  <.0001
Asym.Year1990Variety   0.147    1.1753 352   0.125  0.9004
    xmid.(Intercept)  54.815    0.7548 352  72.622  <.0001
       xmid.Year1989  -2.238    0.9718 352  -2.303  0.0218
       xmid.Year1990  -4.970    0.9743 352  -5.101  <.0001
        xmid.Variety  -1.297    0.4144 352  -3.131  0.0019
    scal.(Intercept)   8.064    0.1472 352  54.762  <.0001
       scal.Year1989  -0.895    0.2013 352  -4.447  <.0001
       scal.Year1990  -0.673    0.2122 352  -3.172  0.0016
. . .
```

The residual plots for `fm4Soy.nlme` do not indicate any violations in the NLME model assumptions. An overall assessment of the adequacy of the model is provided by the plot of the augmented predictions in Figure 6.14, which indicates that the `fm4Soy.nlme` model describes the individual growth patterns of the soybean plots well.

6.4 Clinical Study of Phenobarbital Kinetics

The data for the last example in this chapter come from a clinical pharmacokinetic study of the drug phenobarbital, used for preventing seizures (Grasela and Donn, 1985). The study followed 59 preterm infants during the first 16 days after birth. Each infant received one or more intravenous injections of phenobarbital. As part of routine clinical monitoring, blood samples were drawn from the infants at irregular time intervals to determine the serum concentration of phenobarbital. The number of concentration measurements per individual varies between 1 and 6, with a total of 155 measurements for all 59 individuals. This is typical of clinical pharmacokinetic studies: there are relatively few observations on each of a large number of individuals. In addition to the concentration and dosing information, each infant's birth weight (in kilograms) and 5-minute Apgar score (which gives an overall indication of health of the newborn) are provided. These data have been analyzed in Grasela and Donn (1985), Boeckmann, Sheiner and Beal (1994), Davidian and Giltinan (1995, §6.6), and Littell et al. (1996, §12.5).

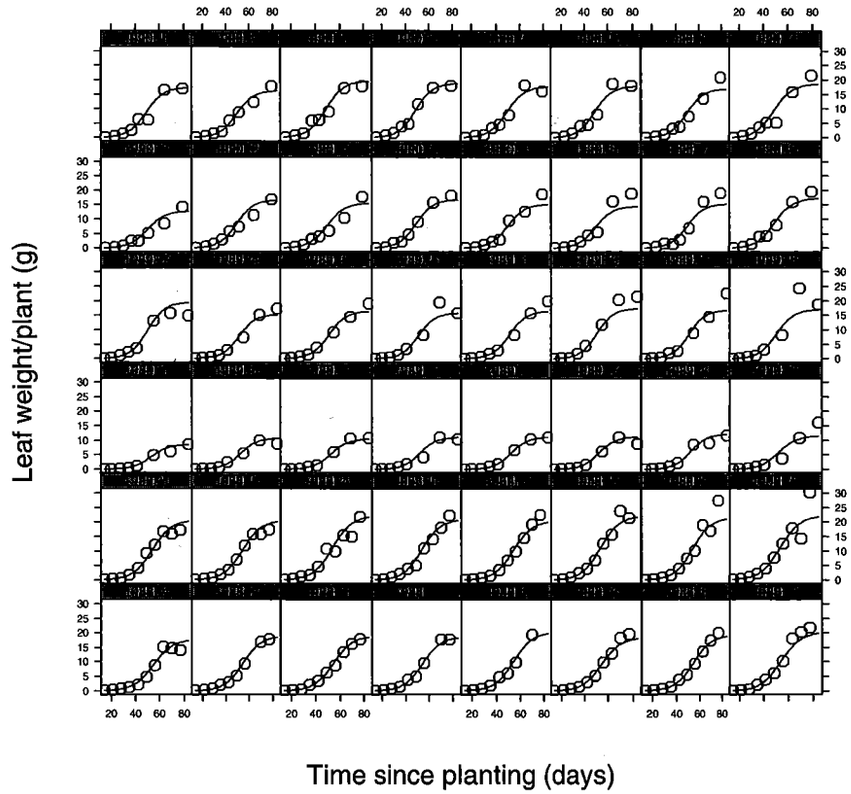

FIGURE 6.14. Average leaf weights of soybean plants versus time since planting and their within-group predictions from the model `fm4Soy.nlme`.

The phenobarbital data, displayed in Figure 6.15, are available in the nlme library as the groupedData object `Phenobarb` and are described in more detail in Appendix A.23.

The mechanistic model postulated for the phenobarbital kinetics is a *one-compartment open model with intravenous administration and first-order elimination* (Grasela and Donn, 1985) . The corresponding model for the expected phenobarbital concentration c_t at time t for an infant receiving a single dose D_d administered at time t_d is

$$c_t = \frac{D_d}{V} \exp\left[-\frac{Cl}{V}(t - t_d)\right], \quad t_d < t,$$

where V and Cl denote, respectively, the *volume of distribution* and the *clearance*. The model for an individual who has received several doses by

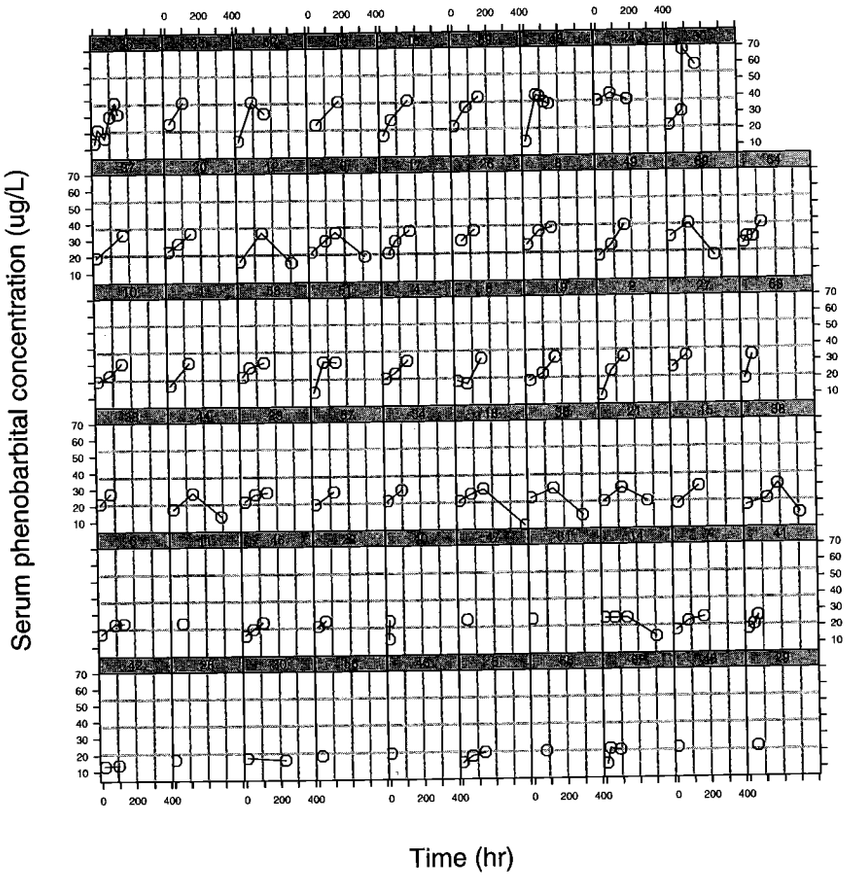

FIGURE 6.15. Serum concentrations of phenobarbital in 59 newborn infants under varying dosage regimens versus time after birth.

time t is given by the sum of the individual contributions of each dose.

$$c_t = \sum_{d:t_d<t} \frac{D_d}{V} \exp\left[-\frac{Cl}{V}(t-t_d)\right]. \tag{6.8}$$

Model (6.8) can also be expressed in recursive form (Grasela and Donn, 1985). To ensure that the estimates of V and Cl are positive, we reparameterize (6.8) using the logarithm of these parameters: $lV = \log V$ and $lCl = \log Cl$. The function `phenoModel` in the `nlme` library implements the reparameterized version of model (6.8) in S. Because `phenoModel` is *not* self-starting, initial values need to be provided for the parameters when this function is used for estimation in S.

Because of the small number of concentrations recorded for each individual, the usual model-building approach of beginning the analysis with an `nlsList` fit cannot be used with the phenobarbital data and we must go directly to an NLME fit. The nonlinear mixed-effects model corresponding to (6.8) representing the phenobarbital concentration y_{ij} measured at time t_{ij} on individual i, following intravenous injections of dose D_{id} at times t_{id}, is expressed as

$$y_{ij} = \sum_{d:t_{id}<t_{ij}} \frac{D_{id}}{\exp(lV_i)} \exp\left[-\exp(lCl_i - lV_i)(t_{ij}-t_{id})\right] + \epsilon_{ij},$$

$$\begin{bmatrix} lCl_i \\ lV_i \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} b_{1i} \\ b_{2i} \end{bmatrix} = \boldsymbol{\beta} + \boldsymbol{b}_i, \quad \boldsymbol{b}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}), \quad \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2). \tag{6.9}$$

The fixed effects, $\boldsymbol{\beta}$, represent the average log-clearance and log-volume of distribution in the infant population, and the random effects, $\boldsymbol{b}_i$, account for individual differences from the population average. As usual, the random effects are assumed to be independent for different individuals and the within-group errors ϵ_{ij} are assumed to be independent for different i, j and to be independent of the random effects.

To avoid convergence problems with the optimization algorithm used in `nlme` due to the sparsity of the phenobarbital concentrations in the data, a diagonal $\boldsymbol{\Psi}$ is assumed in model (6.9), which is then fitted with

```
> fm1Pheno.nlme <-
+   nlme( conc ~ phenoModel(Subject, time, dose, lCl, lV),
+     data = Phenobarb, fixed = lCl + lV ~ 1,
+     random = pdDiag(lCl + lV ~ 1), start = c(-5, 0),
+     na.action = na.include, naPattern = ~ !is.na(conc) )
> fm1Pheno.nlme
Nonlinear mixed-effects model fit by maximum likelihood
  Model: conc ~ phenoModel(Subject, time, dose, lCl, lV)
  Data: Phenobarb
  Log-likelihood: -505.41
  Fixed: lCl + lV ~ 1
     lCl      lV
 -5.0935 0.34259

Random effects:
 Formula: list(lCl ~ 1, lV ~ 1)
 Level: Subject
 Structure: Diagonal
            lCl      lV Residual
StdDev: 0.43989 0.45048   2.7935

Number of Observations: 155
Number of Groups: 59
```

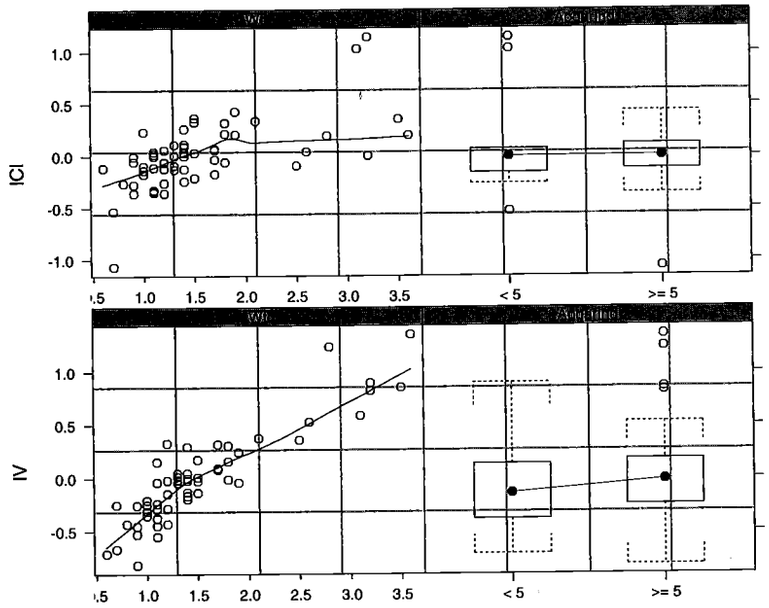

FIGURE 6.16. Estimated log-clearance and log-volume of distribution random effects from model `fm1Pheno.nlme` versus birth weight (`Wt`) and Apgar score indicator (`ApgarInd`) in the phenobarbital data. A `loess` smoother is included in the scatter plots of the continuous covariates to aid in visualizing possible trends.

Starting values for the fixed effects are obtained from Davidian and Giltinan (1995, §6.6). The `na.action` argument in the `nlme` call is used to preserve those rows with dose information. (These rows contain `NA` for the concentration.) The `naPattern` argument is used to remove these rows from the calculation of the objective function in the optimization algorithm.

One of the questions of interest for the phenobarbital data is the possible relationship between the pharmacokinetic parameters in (6.9) and the additional covariates available on the infants, birth weight and 5-minute Apgar score. For the purposes of modeling the pharmacokinetic parameters, the 5-minute Apgar score is converted to a binary indicator of whether the score is < 5 or ≥ 5, represented by the column `ApgarInd` in the `Phenobarb` data frame.

Figure 6.16 contains plots of the estimated random effects from `fm1Pheno.nlme` versus birth weight (`Wt`) and 5-minute Apgar score indicator (`ApgarInd`). It is produced by

```
> fm1Pheno.ranef <- ranef( fm1Pheno.nlme, augFrame = T )
> plot( fm1Pheno.ranef, form = lCl ~ Wt + ApgarInd )
> plot( fm1Pheno.ranef, form = lV ~ Wt + ApgarInd )
```


The plots in Figure 6.16 clearly indicate that both clearance and volume of distribution increase with birth weight. A linear model seems adequate to represent the increase in `lV` with birth weight. For birth weights less than 2.5 kg, the increase in `lCl` seems linear. Because there are few infants with birth weights greater than 2.5 kg in this data set, it is unclear whether the linear relationship between `lCl` and `Wt` extends beyond this limit, but we will assume it does. The Apgar score does not seem to have any relationship with clearance and is not included in the model for the `lCl` fixed effect. It is unclear whether the Apgar score and the volume of distribution are related, so we include `ApgarInd` in the model for `lV` to test for a possible relationship.

The updated fit with covariates included in the fixed-effects model is then obtained with

```
> options( contrasts = c("contr.treatment", "contr.poly") )
> fm2Pheno.nlme <- update( fm1Pheno.nlme,
+   fixed = list(lCl ~ Wt, lV ~ Wt + ApgarInd),
+   start = c(-5.0935, 0, 0.34259, 0, 0),
+   control = list(pnlsTol = 1e-6) )
> #pnlsTol reduced to prevent convergence problems in PNLS step
> summary( fm2Pheno.nlme )
. . .
Random effects:
 Formula: list(lCl ~ 1, lV ~ 1)
 Level: Subject
 Structure: Diagonal
        lCl.(Intercept) lV.(Intercept) Residual
StdDev:         0.21599        0.17206   2.7374

Fixed effects: list(lCl ~ Wt, lV ~ Wt + ApgarInd)
                 Value Std.Error DF t-value p-value
lCl.(Intercept) -5.9574   0.12425 92 -47.947  <.0001
         lCl.Wt  0.6197   0.07569 92   8.187  <.0001
 lV.(Intercept) -0.4744   0.07258 92  -6.537  <.0001
          lV.Wt  0.5325   0.04141 92  12.859  <.0001
    lV.ApgarInd -0.0228   0.05131 92  -0.444  0.6577
. . .
```

As expected, the fixed effects corresponding to birth weight, `lCl.Wt` and `lV.Wt`, are highly significant. The large p-value for the `lV.ApgarInd` fixed effect indicates that the volume of distribution is not related to the Apgar scores. The estimated standard deviations for the random effects are about half of the corresponding values in the `fm1Pheno.nlme` fit, indicating that a substantial part of the between-individual variability in the pharmacokinetic parameters is explained by birth weight.

The `ApgarInd` variable is dropped from the `lV` fixed effect model to give the final NLME model for the phenobarbital data considered here.

```
> fm3Pheno.nlme <- update( fm2Pheno.nlme,
+   fixed = lCl + lV ~ Wt, start = fixef(fm2Pheno.nlme)[-5] )
> fm3Pheno.nlme
Nonlinear mixed-effects model fit by maximum likelihood
  Model: conc ~ phenoModel(Subject, time, dose, lCl, lV)
  Data: Phenobarb
  Log-likelihood: -437.7
  Fixed: lCl + lV ~ Wt
 lCl.(Intercept)  lCl.Wt lV.(Intercept)   lV.Wt
         -5.9577 0.61968       -0.48452 0.53205

Random effects:
 Formula: list(lCl ~ 1, lV ~ 1)
 Level: Subject
 Structure: Diagonal
        lCl.(Intercept) lV.(Intercept) Residual
StdDev:         0.21584        0.17316   2.7326
. . .
```

The likelihood ratio tests for dropping either of the random effects in `fm3Pheno.nlme` have very significant p-values (< 0.0001), indicating that both random effects are needed in the model to account for individual effects.

A plot of the augmented predictions is not meaningful for the phenobarbital data, due to the small number of observations per individual, but we can still assess the adequacy of the fit with the plot of the observed concentrations against the within-group fitted values, produced with

```
> plot( fm3Pheno.nlme, conc ~ fitted(.), abline = c(0,1) )
```

and displayed in Figure 6.17. The good agreement between the observations and the predictions attests the adequacy of the `fm3Pheno.nlme` model.

6.5 Chapter Summary

This chapter gives an introductory overview of the nonlinear mixed-effects model, describing its basic concepts and assumptions and relating it to the linear mixed-effects model described in the first part of the book. Real-life examples from pharmacokinetics studies and an agricultural experiment are used to illustrate the use of the `nlme` function in S, and its associated methods, for fitting and analyzing NLME models.

The many similarities between NLME and LME models allow most of the lme methods defined in the first part of the book to also be used with the nlme objects introduced in this section. There are, however, important differences between the two models, and the methods used to fit them, which translate into more complex estimation algorithms and less accurate inference for NLME models.

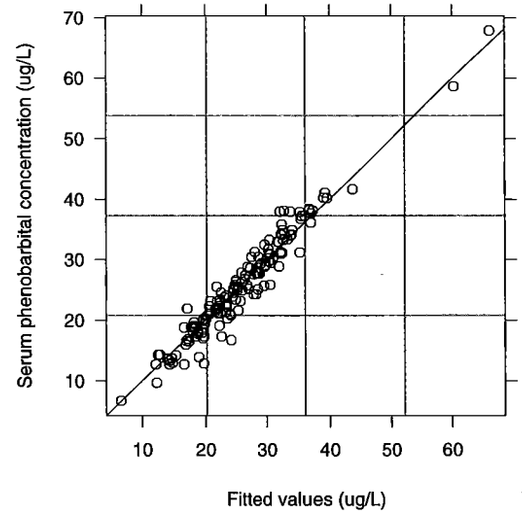

FIGURE 6.17. Observed phenobarbital concentrations versus within-group fitted values corresponding to the `fm3Pheno.nlme` model.

The purpose of this chapter is to present the motivation for using NLME models with grouped data and to set the stage for the following two chapters in the book, dealing with the theory and computational methods for NLME models (Chapter 7) and the nonlinear modeling facilities in the nlme library (Chapter 8).

Exercises

1. The `Loblolly` data described in Appendix A.13 consist of the heights of 14 Loblolly pine trees planted in the southern United States, measured at 3, 5, 10, 15, 20, and 25 years of age. An asymptotic regression model, available in nlme as the self-starting function `SSasymp` (Appendix C.1), seems adequate to explain the observed growth pattern.

   (a) Plot the data (using `plot(Loblolly)`) and verify that the same growth pattern is observed for all trees. The similarity can be emphasized by putting all the curves into a single panel using `plot(Loblolly, outer = ~1)`. What is the most noticeable difference among the curves?

   (b) Fit a separate asymptotic regression model to each tree using `nlsList` and `SSasymp`. Notice that no starting values are needed for the parameters, as they are automatically produced by `SSasymp`.

(c) Plot the confidence intervals on the individual parameters from the `nlsList` fit. Can you identify the ones that vary significantly with tree?

(d) Use the `nlsList` object to fit an NLME model with random effects for all parameters (you can use `nlme(object)`, with `object` replaced with the name of the `nlsList` object). Can you evaluate the confidence intervals on the NLME parameters (using `intervals(object)`)? The error message indicates that the maximum likelihood estimates do not correspond to a numerically stable solution (this generally occurs when the random-effects model is overparameterized).

(e) Update the previous `nlme` fit using a diagonal structure for Ψ (use `random = pdDiag(Asym+R0+lrc ~ 1)` in the call to `update`). Print the results and examine the estimated standard errors of the random effects. Which random effect can be dropped from the model?

(f) Update the `nlme` fit eliminating the random effect you identified in the previous item (continue to use a diagonal Ψ). Compare this fit to the one obtained in the previous item using `anova`. What do you conclude? Can you drop further random effects from the model?

(g) Plot the augmented predictions (using `plot(augPred(object))`) for the final model obtained in the previous item. Does the model seem adequate?

2. Davidian and Giltinan (1995, §1.1, p. 2) describe a pharmacokinetic study of the drug cefamandole in which plasma concentrations of cefamandole at 14 time points following intravenous infusion were collected on 6 healthy volunteers. These data are available in nlme as the object `Cefamandole` and are described in greater detail in Appendix A.4.

(a) Plot the cefamandole plasma concentrations versus time by subject (use `plot(Cefamandole)`. The concentration-time profiles are well described by the biexponential model (6.2) used with the `Indometh` data in §6.2.

(b) Use the self-starting function `SSbiexp` (Appendix C.4) in conjunction with `nlsList` to produce separate fits per subject. Plot the individual confidence intervals and identify the coefficients that seem to vary significantly with subject.

(c) Fit an NLME model to the data with random effects for all coefficients, using the `nlsList` object produced in the previous item. Because of the small number of subjects, convergence is not attained for a model with a general Ψ (try it). Use a diagonal structure for Ψ (`random = pdDiag(A1+lrc1+A2+lrc2~1)`) and examine the estimated standard errors for the random effects. Can any of the random effects be dropped from the model? Does this agree with your conclusions from the plot of the confidence intervals for the `nlsList` fit?

(d) Update the `nlme` fit in the previous item according to your previous conclusions for the random-effects model. Use `anova` to compare the two models. Produce the confidence intervals for the parameters in the updated model. Do you think any further random effects can be dropped from the model? If so, update the fit and compare it to the previous model using `anova`.

(e) Plot the standardized residuals versus the fitted values for the final model obtained in the previous item. Do you observe any patterns that contradict the model's assumptions? Plot the observed concentrations versus the fitted values by subject. Does the fitted model produce sensible predictions?

3. Data on the intensity of 23 large earthquakes in western North America between 1940 and 1980 were reported by Joyner and Boore (1981). The data, included in the object `Earthquake` , are described in more detail in Appendix A.8. The objective of the study was to predict the maximum horizontal acceleration (`accel`) at a given location during a large earthquake based on the magnitude of the quake (`Richter`) and its distance from the epicenter (`distance`). These data are analyzed in Davidian and Giltinan (1995, §11.4, pp. 319–326). The model proposed by Joyner and Boore (1981) can be written as

$$\log_{10}(\texttt{accel}) = \phi_1 + \phi_2 \texttt{Richter} - \log_{10}\sqrt{\texttt{distance}^2 + \exp(\phi_3)} - \phi_4\sqrt{\texttt{distance}^2 + \exp(\phi_3)}.$$

(a) Plot the data and verify that acceleration measurements are sparse or noisy for most of the quakes. No common attenuation pattern is evident from the plot.

(b) No self-starting function is available in nlme for the `Earthquake` data model. The estimates reported in Davidian and Giltinan (1995) ($\widehat{\phi}_1 = -0.764$, $\widehat{\phi}_2 = 0.218$, $\widehat{\phi}_3 = 1.845$, and $\widehat{\phi}_4 = 5.657 \times 10^{-3}$) can be used as initial estimates for an `nlsList` fit (use `start = c(phi1 = -0.764, phi2 = 0.218, phi3 = 1.845, phi4 = 0.005657)`. However, due to sparse and noisy nature of the data, convergence is not attained for any of the quakes in the `nlsList` fit (verify it).

(c) Fit an NLME model to the data with random effects for all coefficients and a diagonal Ψ, using as starting values for the

fixed effects the estimates reported in Davidian and Giltinan (1995) (listed in the previous item). Note that, even though the model did not converge for any of the individual quakes in the `nlsList` fit, it converged for the combined data in the `nlme` fit. The individual quakes "borrow strength" from each other in the `nlme` fit.

(d) Examine the estimated standard errors for the random effects in the `nlme` fit relative to the absolute value of the estimated fixed effects. Can any of the random effects be dropped from the model? If so, refit the model with fewer random effects an compare it to the previous model using `anova`. Repeat the procedure until no further random effects can be removed from the model. (You may have to use `nls` to fit a model with no random effects, which can also be compared to an `nlme` fit using `anova`, provided the `nlme` object comes first in the argument list.)

(e) Examine the plot of the standard residuals versus the fitted values and the normal plot of the standardized residuals for the final model obtained in the previous item. Are there any apparent departures from the model's assumptions? Plot the observed `log10(accel)` measurements versus the fitted values. How well does the model predict the accelerations?

(f) One of the questions of interest in this study was the possible effect of soil type (represented in `Earthquake` by the indicator variable `soil`, taking values `0` for "rock" and `1` for "soil") on acceleration. Update the final `nlme` model obtained before to include a "soil effect" for the fixed effects of coefficients with an associated random effect. For example, if `phi4` is the only coefficient with a random effect, you can use `fixed = list(phi1+phi2+phi3 ~ 1, phi4 ~ soil)`. You will also need to provide initial estimates for the fixed effects, using for example `start = c(fixef(object), 0)`, where `object` should be replaced with the name of the `nlme` object being updated. Use `summary` to test the significance of soil type.