

DigiSem
Wir beschaffen und
digitalisieren



b
UNIVERSITÄT
BERN

Universitätsbibliothek Bern

Dieses Dokument steht Ihnen online zur Verfügung
dank DigiSem, einer Dienstleistung der
Universitätsbibliothek Bern.

Kontakt: Gabriela Scherrer
Koordinatorin digitale Semesterapparate
E-Mail digisem@ub.unibe.ch, Telefon 031 631 93 26

José C. Pinheiro
Douglas M. Bates

Mixed-Effects Models in S and S-PLUS

With 172 Illustrations



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

Series Editors:

J. Chambers
Bell Labs, Lucent
Technologies
600 Mountain Ave.
Murray Hill, NJ 07974
USA

W. Eddy
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
USA

W. Härdle
Institut für Statistik und
Ökonometrie
Humboldt-Universität zu Berlin
Spandauer Str. 1
D-10178 Berlin
Germany

S. Sheather
Australian Graduate School
of Management
University of New South
Wales
Sydney NSW 2052
Australia

L. Tierney
School of Statistics
University of Minnesota
Vincent Hall
Minneapolis, MN 55455
USA

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.

© 2000 Springer Verlag New York, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, LLC, 175 Fifth Avenue, New York, NY 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden. The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Printed in the United States of America. (HAM)

9 8 7 6 5 SPIN 10995662

Springer Verlag is a part of *Springer Science+Business Media*

springeronline.com

Fitting Linear Mixed-Effects Models

As seen in Chapter 1, mixed-effects models provide a flexible and powerful tool for analyzing balanced and unbalanced grouped data. These models have gained popularity over the last decade, in part because of the development of reliable and efficient software for fitting and analyzing them. The linear and nonlinear mixed-effects (`nlme`) library in S is an example of such software. We describe the `lme` function from that library in this chapter, as well the methods for displaying and comparing fitted models created by this function.

The first section gives a brief review of the standard linear modeling facilities in S to introduce the general style of the S modeling functions, classes, and methods that are used with the `nlme` library. The `lmeList` function, used to obtain separate `lme` fits according to the levels of a grouping variable, is described and illustrated through examples.

The next section describes the linear mixed-effects modeling capabilities in S. The S modeling function `lme` is described, together with its associated methods. Its use is illustrated through examples, including single- and multilevel grouped data.

After a model has been fit to the data, it is important to examine whether the underlying assumptions appear to be violated. Graphical methods and numerical summaries for assessing the validity of the assumptions in a linear mixed-effects model are described in §4.3.

An “inside-out” model building approach is adopted here, starting with individual fits by group, using plots of the individual coefficients to decide on the random-effects structure, and finally fitting a mixed-effects model

to the complete data. We make extensive use of examples to introduce and illustrate the available functions and methods.

In this chapter we will restrict our attention to models in which the within-group errors are independent and have equal variance. Models with more complex within-group covariance structures, such as the heterogeneous $AR(1)$ structure, will be explored in detail in Chapter 5.

4.1 Fitting Linear Models in S with `lm` and `lmList`

S offers a variety of functions and methods for fitting and manipulating linear models. The two main modeling functions are `lm`, for linear regression models, and `aov`, for analysis of variance models. These two functions have similar syntax and generate similar fitted objects. We concentrate here on `lm`. A typical call to `lm` is of the form

```
lm(formula, data)
```

where `formula` specifies the linear model to be fitted and `data` gives a data frame in which the variables in `formula` are to be evaluated. Several other arguments to `lm` are available and are described in detail in Chambers and Hastie (1992, Chapter 4) and also in Venables and Ripley (1999, Chapter 6).

The formula language used in the `formula` argument gives `lm` great flexibility in specifying linear models. The formulas use a version of the syntax defined by Wilkinson and Rogers (1973), which translates a linear model like $y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \epsilon$ into the S expression

```
y ~ 1 + x1 + x2
```

The `~` is read “is modeled as.” The expression on the left-hand side of the `~` specifies the response to be modeled. The expression on the right-hand side describes the covariates used and the ways they will be combined to form the model. The expression does not include the coefficients (the β ’s). They are implicit.

The constant term `1` is included by default and does not need to be given explicitly in the model. If a model without an intercept term is desired, a `-1` must be included in the formula. The covariates can be factors, ordered factors, continuous variables, matrices, or data frames. Any function of a covariate that returns one of these types of covariate can also be used on the right-hand side of the formula. Function calls are also allowed on the left-hand side of the formula. For example,

```
log(y) ~ exp(x1) + cos(x2)
```

is a valid `lm` formula. An interaction between two covariates is denoted by the `:` operator. Nesting of a covariate within a factor is denoted by the

`%in%` operator. More detailed references on the formula language include Chambers and Hastie (1992, Chapter 2) and Venables and Ripley (1999, §6.2).

The `lm` function operates in a style common to most modeling functions in S, in particular `lme` and `nlme`. A call to `lm` returns a fitted object of class `lm` to which several generic functions can be applied. These can display the results of the fit (`print` and `summary`), produce diagnostic plots (`plot`), return predictions (`predict`), extract components (`fitted`, `residuals`, and `coef`), update the original model (`update`), or compare different fitted objects (`anova`).

To illustrate some of these capabilities, we revisit the orthodontic growth curve data of §1.4. Suppose that we initially ignore the grouping structure in the data and fit a single linear regression model of `distance` on `age` to the data from all the subjects. The corresponding call to `lm` is

```
> fm1Orth.lm <- lm( distance ~ age, Orthodont )
```

A brief description of the results is provided by the `print` method which is called implicitly when the fitted object is to be displayed.

```
> fm1Orth.lm          # equivalent to print( fm1Orth.lm )
Call:
lm(formula = distance~age, data = Orthodont)

Coefficients:
(Intercept)      age
16.761        0.66019

Degrees of freedom: 108 total; 106 residual
Residual standard error: 2.5372
```

Diagnostic plots for assessing the quality of the fit are obtained using the `method` for the `plot` generic function

```
> par( mfrow=c(3,2) )          # arrange 6 separate plots on a page
> plot( fm1Orth.lm )          # Figure 4.1
```

There is considerable variability remaining after the fit, as shown in the residual plots. This is not surprising, as we know that the simple linear regression model does not represent the structure of the data well. Apparently there are some observations with unusual influence on the fit, especially observations 39 and 104. Furthermore, the normal probability plot of the residuals suggests that the error distribution has heavier tails than expected from normally distributed variates.

Suppose that we now want to test for possible differences in intercept or in slope between boys and girls. We can use the following model,

$$\text{distance} = \beta_0 + \beta_1 \text{Sex} + \beta_2 \text{age} + \beta_3 \text{age} \times \text{Sex} + \epsilon, \quad (4.1)$$

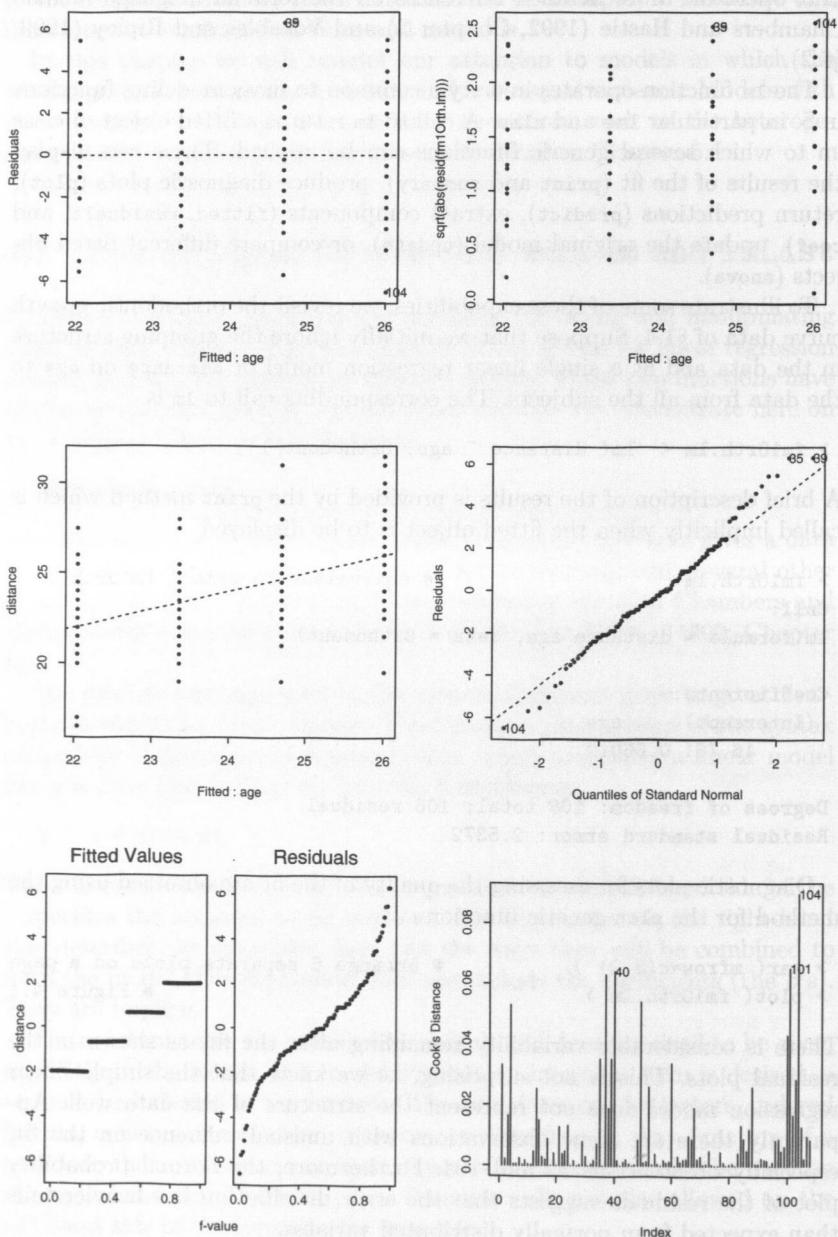


FIGURE 4.1. Diagnostic plots for the simple linear regression model fit of the orthodontic growth curve data.

with Sex representing a binary variable taking values -1 for boys and 1 for girls. The parameters β_1 and β_3 represent, respectively, the intercept and slope gender effects. We can fit this model in S with another call to lm or by using update on the previous fitted model and redefining the formula.

```
> fm20rth.lm <- update( fm10rth.lm, formula = distance ~ Sex*age )
```

The expression `Sex*age` in a linear model formula crosses the `Sex` and `age` factors. This means it generates the main effects for these factors and their interaction. It is equivalent to `Sex + age + Sex:age`.

The `summary` method displays the results in more detail. In particular, it provides information about the marginal significance of the parameter estimates.

```
> summary( fm20rth.lm )
Call: lm(formula = distance ~ Sex + age + Sex:age, data = Orthodont)
Residuals:
    Min      1Q Median      3Q     Max 
-5.62 -1.32 -0.168  1.33  5.25 

Coefficients:
```

	Value	Std. Error	t value	Pr(> t)
(Intercept)	16.857	1.109	15.194	0.000
Sex	0.516	1.109	0.465	0.643
age	0.632	0.099	6.394	0.000
Sex:age	-0.152	0.099	-1.542	0.126

Residual standard error: 2.26 on 104 degrees of freedom

Multiple R-Squared: 0.423

F-statistic: 25.4 on 3 and 104 d.o.f., the p-value is 2.11e-12

Correlation of Coefficients:

	(Intercept)	Sex	age
Sex	0.185		
age	-0.980	-0.181	
Sex:age	-0.181	-0.980	0.185

The *p*-values for the `Sex` and `Sex:age` coefficients suggest that there is no gender effect on the orthodontic measurement growth. Because the *t*-test is only measuring the marginal significance of each term in the model, we should proceed with caution and delete one term at a time from the model. Deleting first the least significant term, `Sex`, we get:

```
> fm30rth.lm <- update( fm20rth.lm, formula = . ~ . - Sex )
> summary( fm30rth.lm )
...
```

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	16.761	1.086	15.432	0.000

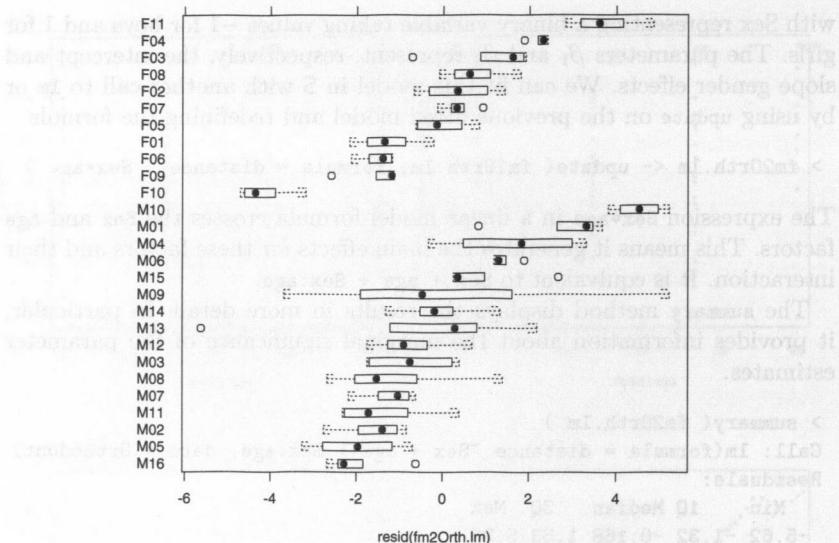


FIGURE 4.2. Residual plots corresponding to the `fm2Orth.lm` fitted object, by subject.

age	0.640	0.097	6.613	0.000
Sex:age	-0.107	0.020	-5.474	0.000

By convention, the `.~.` expression represents the `formula` in the object being updated and the `-` operator is used to delete terms from the model.

The `Sex:age` coefficient now becomes very significant, indicating that the growth patterns are different for boys and girls. Because the `lm` fit is not adequate for these data, we will postpone further discussion of these model-building issues until the linear mixed-effects model has been described.

The grouped nature of these data, with repeated measures on each subject at four different years, violates the basic assumption of independence that underlies the statistical methods used in `lm`. Boxplots of the `fm2Orth.lm` residuals by subject show this.

```
> bwplot( getGroups(Orthodont)~resid(fm2Orth.lm) ) # Figure 4.2
```

The most important feature observed in Figure 4.2 is that residuals corresponding to the same subject tend to have the same sign. This indicates the need for a “subject effect” in the model, which is precisely the motivation for linear mixed-effects models.

4.1.1 Separate `lm` Fits per Group: the `lmList` Function

The first step in the model-building process for a linear mixed-effects model, after the functional form of the model has been decided, is choosing which parameters in the model, if any, should have a random-effect component included to account for between-group variation. The `lmList` function and the methods associated with it are useful for this.

A typical call to `lmList` is

```
lmList( formula, data )
```

where the right-hand side of the `formula` consists of two parts separated by the `|` operator. The first part specifies the linear model to be fitted to each subset of `data`; the second part specifies the grouping factor. Any linear formula allowed in `lm` can also be used as a model formula with `lmList`. The `data` argument gives the data frame in which to find the variables used in `formula`.

Continuing with the analysis of the orthodontic data, we see from a Trellis plot of these data (Figure 1.11, page 31) that a simple linear regression model of `distance` as a function of `age` may be suitable. We fit this by

```
> fm10orth.lis <- lmList( distance ~ age | Subject, Orthodont )
```

If `data` is a `groupedData` object (see Chapter 3), the grouping variable can be omitted from `formula`, being extracted from the group formula in `data`.

```
> getGroupsFormula( Orthodont )
~ Subject
```

so an alternative call to `lmList` to obtain the same fitted object is

```
> fm10orth.lis <- lmList( distance ~ age, Orthodont )
```

Because the `lmList` function is a *generic* function (Chambers and Hastie, 1992, Appendix A) with different methods for arguments of different classes, this same fit can be specified in an even simpler way. If the first argument to `lmList` is a `groupedData` object, the display formula for this object is used to create a default model formula and to extract the grouping variable expression. Because we are using the same grouping, response, and covariate in our `lmList` fit as in the display formula

```
> formula( Orthodont )
distance ~ age | Subject
```

we can obtain the same fitted model object with the simpler call

```
> fm10orth.lis <- lmList( Orthodont )
```

Objects returned by `lmList` are of class `lmList`, for which several display and plot methods are available. Table 4.1 lists some of the most important methods for class `lmList`. We illustrate the use of some of these methods below.

The `print` method displays minimal information about the fitted object.

TABLE 4.1. Main `lmeList` methods.

<code>augPred</code>	predictions augmented with observed values
<code>coef</code>	coefficients from individual <code>lm</code> fits
<code>fitted</code>	fitted values from individual <code>lm</code> fits
<code>fixef</code>	average of individual <code>lm</code> coefficients
<code>intervals</code>	confidence intervals on coefficients
<code>lme</code>	linear mixed-effects model from <code>lmeList</code> fit
<code>logLik</code>	sum of individual <code>lm</code> log-likelihoods
<code>pairs</code>	scatter-plot matrix of coefficients or random effects
<code>plot</code>	diagnostic Trellis plots
<code>predict</code>	predictions for individual <code>lm</code> fits
<code>print</code>	brief information about the <code>lm</code> fits
<code>qqnorm</code>	normal probability plots
<code>ranef</code>	deviations of coefficients from average
<code>resid</code>	residuals from individual <code>lm</code> fits
<code>summary</code>	more detailed information about <code>lm</code> fits
<code>update</code>	update the individual <code>lm</code> fits

```
> fm1Orth.lis
Call:
Model: distance ~ age | Subject
Data: Orthodont

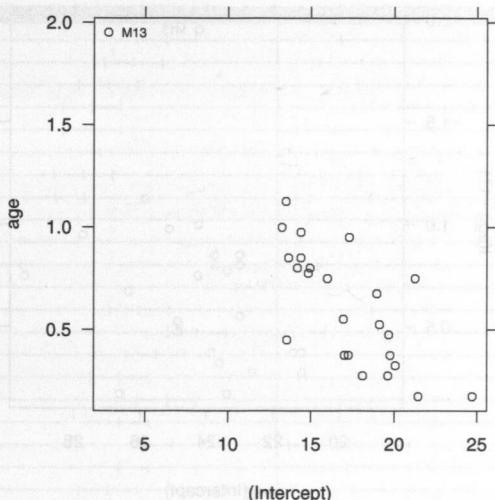
Coefficients:
(Intercept)    age
M16          16.95 0.550
...
F11          18.95 0.675

Degrees of freedom: 108 total; 54 residual
Residual standard error: 1.31
```

The *residual standard error* given in the output is the pooled estimate of the standard error calculated from the individual `lm` fits by group. More detailed output can be obtained using the `summary` method.

```
> summary( fm1Orth.lis )
Call:
Model: distance ~ age | Subject
Data: Orthodont

Coefficients:
(Intercept)
Value Std. Error t value Pr(>|t|)
M16 16.95     3.2882 5.15484 3.6952e-06
```

FIGURE 4.3. Pairs plot for `fm10orth.lis`.

```
M05 13.65      3.2882 4.15124 1.1817e-04
...
F11 0.675     0.29293 2.30428 2.5081e-02
```

Residual standard error: 1.31 on 54 degrees of freedom

Diagnostic plots can be obtained using the `plot` method, as described in §4.3.

The main purpose of the preliminary analysis provided by `lmList` is to give an indication of what random-effects structure to use in a linear mixed-effects model. We must decide which random effects to include in a model for the data, and what covariance structure these random effects should have. The `pairs` method provides one view of the random-effects covariance structure.

```
> pairs( fm10orth.lis, id = 0.01, adj = -0.5 ) # Figure 4.3
```

The `id` argument is used to identify outliers—points outside the estimated probability contour at level $1-\text{id}/2$ will be marked in the plot. We see that subject M13 has an unusually low intercept, compensated by a large slope. There appears to be a negative correlation between the intercept and slope estimates. Those with experience analyzing regression models may already have guessed why this pattern occurs. It is because all the data were collected between age 8 and age 14, but the intercept represents the measurement at age 0. This causes a high negative correlation (-0.98) between estimates of the slopes and the intercepts. We can remove this correlation by centering the data. In this case, we fit `distance` as a linear function of

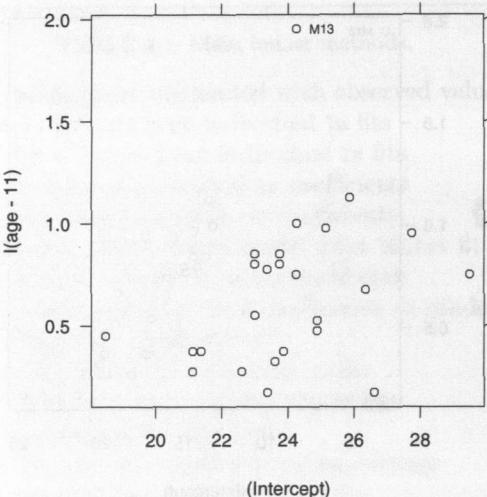


FIGURE 4.4. Pairs plot for `fm20orth.lis` with ages centered at 11 years.

`age - 11`. The two quantities being estimated then are the distance at 11 years of age and the slope, or growth rate. We fit this revised model with

```
> fm20orth.lis <- update( fm10orth.lis, distance ~ I(age-11) )
```

The corresponding pairs plot (Figure 4.4) does not suggest any correlation between the intercept estimates and the slope estimates. It is clear that the orthodontic distance for subject M13 has grown at an unusually fast rate, but his orthodontic distance at age 11 was about average. Both intercept and slope estimates seem to vary with individual, but to see how significantly they vary among subjects we need to consider the precision of the `lmeList` estimates. This can be evaluated with the `intervals` method.

```
> intervals( fm20orth.lis )
, , (Intercept)
    lower   est.   upper
M16 21.687 23.000 24.313
M05 21.687 23.000 24.313
M02 22.062 23.375 24.688
...
F04 23.562 24.875 26.188
F11 25.062 26.375 27.688
, , I(age - 11)
    lower   est.   upper
M16 -0.037297 0.550 1.1373
M05 0.262703 0.850 1.4373
```

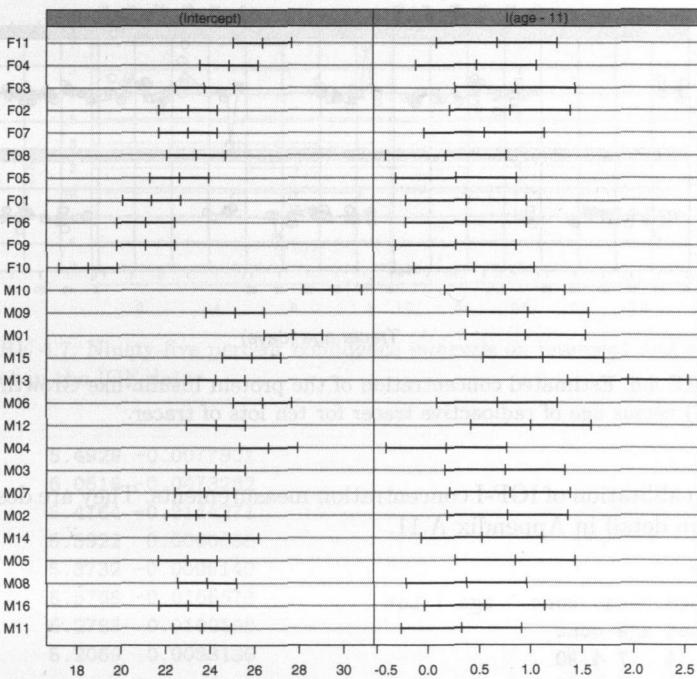


FIGURE 4.5. Ninety-five percent confidence intervals on intercept and slope for each subject in the orthodontic distance growth data.

```
M02 0.187703 0.775 1.3623
...
F04 -0.112297 0.475 1.0623
F11 0.087703 0.675 1.2623
```

As often happens, displaying the intervals as a table of numbers is not very informative. We find it much more effective to plot these intervals using

```
> plot( intervals(fm2Orth.lis) ) # Figure 4.5
```

The individual confidence intervals in Figure 4.5 give a clear indication that a random effect is needed to account for subject-to-subject variability in the intercept. Except for subject M13, all confidence intervals for the slope overlap, so perhaps this parameter can be regarded as a fixed effect in the mixed-effects model. We will explore these questions in §4.2.1, while describing the `lme` function.

To further illustrate the capabilities of `lmList`, we consider data on radioimmunoassays of the protein Insulin-like Growth Factor (IGF-I) presented in Davidian and Giltinan (1995, §3.2.1, p. 65). The data are from quality control radioimmunoassays for ten different lots of a radioactive tracer used

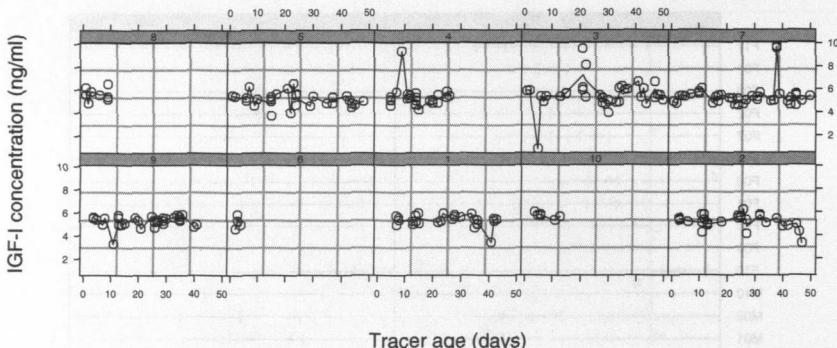


FIGURE 4.6. Estimated concentration of the protein Insulin-like Growth Factor (IGF-I) versus age of radioactive tracer for ten lots of tracer.

in the calibration of IGF-I concentration measurements. They are described in more detail in Appendix A.11.

```
> IGF
Grouped Data: conc ~ age | Lot
  Lot age conc
    1   1   7 4.90
    2   1   7 5.68
    .
    236 10  11 5.30
    237 10  13 5.63
```

This data set, displayed in Figure 4.6, is an example of unbalanced, repeated measures data. We do not consider these data to be longitudinal because different tracer samples are used at each radioimmunoassay. This reduces the potential for serial correlation in the responses.

The primary purpose of the IGF-I experiment was to investigate possible trends in control values with tracer age, which would indicate tracer decay within the usual storage period. We can investigate this by testing if the slope of a simple linear regression model is significantly different from zero. We must account for both the within-lot and the between-lot variability when fitting the model and testing for the significance of the slope. A linear mixed-effects model will do this, but first we investigate the sources of variation in the data by fitting separate regression lines to each lot. As the fixed-effects formula coincides with the display formula in `IGF`, we can use the simple form of the call to `lmList`

```
> fm1IGF.lis <- lmList( IGF )
> coef( fm1IGF.lis )
  (Intercept)      age
  9       5.0986  0.0057276
  6       4.6300  0.1700000
```

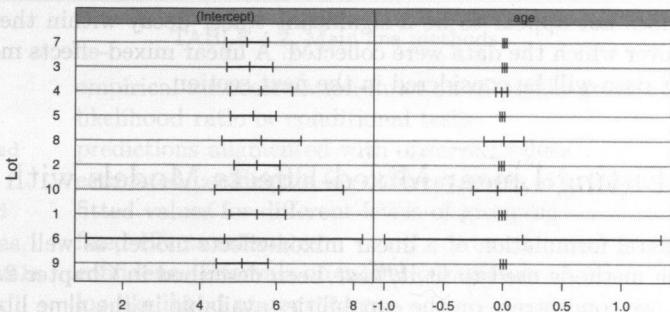


FIGURE 4.7. Ninety-five percent confidence intervals on intercept and slope for each lot in the IGF data.

```

1      5.4929 -0.0077901
10     6.0516 -0.0473282
2      5.4764 -0.0144271
8      5.5922  0.0060638
5      5.3732 -0.0095140
4      5.5768 -0.0166578
3      5.2788  0.0100830
7      5.2069  0.0093136

```

A quick look at the individual coefficient estimates indicates that Lot 6 is unusual. It has a low intercept compensated by a high slope. Examination of Figure 4.6 shows that this lot has only four observations and that these are at nearby tracer ages. The coefficients from the individual fit to this lot are unreliable. We will return to this issue in §4.2.1.

The plot of the individual 95% confidence intervals, shown in Figure 4.7, provides some insight about lot-to-lot variation in the parameter estimates.

```
> plot( intervals(fm1IGF.lis) ) # Figure 4.7
```

Because of the imbalance in the data, these confidence intervals have very different lengths. There is little indication of lot-to-lot variation in either the intercept or the slope estimates, since all confidence intervals overlap. A fixed-effects model seems adequate to represent the IGF data.

```

> fm1IGF.lm <- lm( conc ~ age, data = IGF )
> summary( fm1IGF.lm )
...
Coefficients:
            Value Std. Error t value Pr(>|t|)
(Intercept) 5.351   0.104    51.584  0.000
age        -0.001   0.004    -0.170  0.865

```

Residual standard error: 0.833 on 235 degrees of freedom

There does not appear to be a significant tracer decay within the 50-day period over which the data were collected. A linear mixed-effects model for the IGF data will be considered in the next section.

4.2 Fitting Linear Mixed-Effects Models with `lme`

The general formulation of a linear mixed-effects model, as well as the estimation methods used to fit it, have been described in Chapter 2. In this section, we concentrate on the capabilities available in the `nlme` library for fitting such models. We initially consider `lme` fits for single-level grouped data with general covariance structures for the random effects. Fitting models with patterned covariance structures for the random effects is described in §4.2.2. In §4.2.3, we describe how to fit multilevel models with `lme`.

4.2.1 Fitting Single-Level Models

We use the `lme` function to fit linear mixed-effects models by maximum likelihood or by restricted maximum likelihood (the default). Several optional arguments can be used with this function, but a typical call is

```
lme( fixed, data, random )
```

The first argument is a two-sided linear formula specifying the fixed effects in the model. The third argument is typically given as a one-sided linear formula, specifying the random effects and the grouping structure in the model. For the orthodontic data, with the ages centered at 11 years, these formulas are:

```
fixed = distance ~ I(age-11), random = ~ I(age-11) | Subject
```

Note that the response is specified only in the `fixed` formula. If the `random` formula is omitted, its default value is taken as the right-hand side of the `fixed` formula. This describes a model in which every fixed effect has an associated random effect. To use this default, `data` must be a `groupedData` object, so the formula for the grouping structure can be obtained from the display formula.

The argument `data` specifies a data frame in which the variables named in `fixed` and `random` can be evaluated. When `data` inherits from class `groupedData`, the expression defining the grouping structure can be omitted in `random`.

A simple call to `lme` to fit the orthodontic data model is

```
> fm10Orth.lme <- lme( distance ~ I(age-11), data = Orthodont,
+                         random = ~ I(age-11) | Subject )
```

or, because `Orthodont` is a `groupedData` object and, by default, the random effects have the same form as the fixed effects

TABLE 4.2. Main `lme` methods.

<code>ACF</code>	empirical autocorrelation function of within-group residuals
<code>anova</code>	likelihood ratio or conditional tests
<code>augPred</code>	predictions augmented with observed values
<code>coef</code>	estimated coefficients for different levels of grouping
<code>fitted</code>	fitted values for different levels of grouping
<code>fixef</code>	fixed-effects estimates
<code>intervals</code>	confidence intervals on model parameters
<code>logLik</code>	log-likelihood at convergence
<code>pairs</code>	scatter-plot matrix of coefficients or random effects
<code>plot</code>	diagnostic Trellis plots
<code>predict</code>	predictions for different levels of grouping
<code>print</code>	brief information about the fit
<code>qqnorm</code>	normal probability plots
<code>ranef</code>	random-effects estimates
<code>resid</code>	residuals for different levels of grouping
<code>summary</code>	more detailed information about the fit
<code>update</code>	update the <code>lme</code> fit
<code>Variogram</code>	semivariogram of within-group residuals

```
> fm1Orth.lme <- lme( distance ~ I(age-11), data = Orthodont )
```

Because the `lme` function is generic, the model can be described in several different ways. For example, there is an `lme` method for `lmList` objects. When an `lmList` object, such as `fm2Orth.lis` in §4.1.1, is given as the first argument to `lme`, it provides default values for `fixed`, `random`, and `data`. We can create the same fitted model with the simple call

```
> fm1Orth.lme <- lme( fm2Orth.lis )
```

One advantage of this method is that initial estimates for the parameters in the profiled (restricted-)likelihood of the mixed-effects model are automatically calculated from the `lmList` object.

The fitted object is of the `lme` class, for which several methods are available to display, plot, update, and further explore the estimation results. Table 4.2 lists the most important methods for class `lme`. We illustrate the use of these methods through the examples in the next sections.

Orthodontic Growth Curve

As for all the classes of objects representing fitted models, the `print` method for the `lme` class returns a brief description of the estimation results. It prints the estimates of the standard deviations and the correlations of the

random effects, the within-group standard error, and the fixed effects. For the `fm1Orth.lme` object it gives

```
> fm1Orth.lme
Linear mixed-effects model fit by REML
Data: Orthodont
Log-restricted-likelihood: -221.32
Fixed: distance ~ I(age - 11)
(Intercept) I(age - 11)
24.023     0.66019

Random effects:
Formula: ~ I(age - 11) | Subject
Structure: General positive-definite
   StdDev   Corr
(Intercept) 2.13433 (Inter
I(age - 11) 0.22643 0.503
Residual    1.31004

Number of Observations: 108
Number of Groups: 27
```

One of the questions of interest for the orthodontic growth data is whether boys and girls have different growth patterns. We can assess this by fitting the model

```
> fm2Orth.lme <- update(fm1Orth.lme, fixed = distance~Sex*I(age-11))
```

Note that `lmList` cannot be used to test for gender differences in the orthodontic growth data, as it estimates individual coefficients for each subject. In general, we will not be able to use `lmList` to test for differences due to factors that are invariant with respect to the groups.

Some more detailed output is supplied by `summary`.

```
> summary(fm2Orth.lme)
Linear mixed-effects model fit by REML
Data: Orthodont
      AIC      BIC      logLik
451.35 472.51 -217.68

Random effects:
Formula: ~ I(age - 11) | Subject
Structure: General positive-definite
   StdDev   Corr
(Intercept) 1.83033 (Inter
I(age - 11) 0.18035 0.206
Residual    1.31004

Fixed effects: distance ~ Sex + I(age - 11) + Sex:I(age - 11)
```

```

Value Std.Error DF t-value p-value
(Intercept) 23.808 0.38071 79 62.537 <.0001
Sex -1.161 0.38071 25 -3.048 0.0054
I(age - 11) 0.632 0.06737 79 9.381 <.0001
Sex:I(age - 11) -0.152 0.06737 79 -2.262 0.0264

Correlation:
      (Intrc Sex I(-11))
Sex 0.185
I(age - 11) 0.102 0.019
Sex:I(age - 11) 0.019 0.102 0.185

Standardized Within-Group Residuals:
    Min      Q1      Med      Q3      Max
-3.1681 -0.38594 0.0071041 0.44515 3.8495

```

Number of Observations: 108

Number of Groups: 27

The small *p*-values associated with Sex and Sex:I(age-11) in the *summary* output indicate that boys and girls have significantly different orthodontic growth patterns.

The *fitted* method is used to extract the fitted values from the *lme* object, using the methodology described in §1.4.2. By default, the *within-group* fitted values, that is, the fitted values corresponding to the individual coefficient estimates, are produced. Population fitted values, based on the fixed-effects estimates alone, are obtained setting the *level* argument to 0 (zero). Both types of fitted values can be simultaneously obtained with

```

> fitted( fm2Orth.lme, level = 0:1 )
  fixed Subject
 1 22.616 24.846
 2 24.184 26.576
 3 25.753 28.307
. .

```

Residuals are extracted with the *resid* method, which also takes a *level* argument.

```

> resid( fm2Orth.lme, level = 1 )
  M01      M01      M01      M01      M02      M02      M02
  1.1543 -1.5765 0.69274 0.96198 0.22522 -0.29641 -1.318 0.66034
. .
  F10      F10      F10      F10      F11      F11      F11
 -1.2233 0.44296 -0.39073 -0.72443 0.28277 -0.37929 1.4587
  F11
  0.29661
attr(, "label"):
[1] "Residuals (mm)"

```

By default, the raw, or *response* residuals, given by the observed responses minus the fitted values, are calculated. Standardized, or *Pearson* residuals,

corresponding to the raw residuals divided by the estimated within-group standard deviation, are obtained using

```
> resid( fm20orth.lme, level = 1, type = "pearson" )
   M01      M01      M01      M02      M02      M02
0.88111 -1.2034 0.5288 0.73431 0.17192 -0.22626 -1.0061
.
.
   F09      F10      F10      F10      F10      F11      F11
-0.76369 -0.93382 0.33813 -0.29826 -0.55298 0.21585 -0.28952
   F11      F11
1.1135 0.22641
attr(, "label"):
[1] "Standardized residuals"
```

Partial matching of arguments is used throughout the *nlme* library, so `type = "p"` would suffice in this case.

Predicted values are obtained with the `predict` method. For example, to predict the orthodontic distance for boy M11 and girl F03 at ages 16, 17 and 18, we first define a data frame with the relevant information

```
> newOrth <- data.frame( Subject = rep(c("M11", "F03"), c(3, 3)),
+                           Sex = rep(c("Male", "Female"), c(3, 3)),
+                           age = rep(16:18, 2) )
```

and then use

```
> predict( fm20orth.lme, newdata = newOrth )
   M11      M11      M11      F03      F03      F03
26.968 27.612 28.256 26.614 27.207 27.8
attr(, "label"):
[1] "Predicted values (mm)"
```

By default, the within-group predictions are produced. To obtain both population and within-group predictions we use

```
> predict( fm20orth.lme, newdata = newOrth, level = 0:1 )
  Subject predict.fixed predict.Subject
1      M11      28.891      26.968
2      M11      29.675      27.612
3      M11      30.459      28.256
4      F03      25.045      26.614
5      F03      25.525      27.207
6      F03      26.005      27.800
```

The `predict.fixed` column gives the population predictions, while `predict.Subject` gives the within-group predictions. We see that M11 is below the boys' average, while F03 is above the girls' average.

The `fm20orth.lme` object corresponds to a restricted-maximum likelihood fit, which tends to produce more conservative estimates of the variance components. A maximum likelihood fit is obtained with

```

> fm2Orth.lmeM <- update( fm2Orth.lme, method = "ML" )
> summary( fm2Orth.lmeM )
Linear mixed-effects model fit by maximum likelihood
Data: Orthodont
      AIC      BIC logLik
443.81 465.26 -213.9

Random effects:
Formula: ~ I(age - 11) | Subject
Structure: General positive-definite
          StdDev   Corr
(Intercept) 1.75219 (Inter
I(age - 11) 0.15414 0.234
Residual    1.31004

Fixed effects: distance ~ Sex + I(age - 11) + Sex:I(age - 11)
                Value Std.Error DF t-value p-value
(Intercept) 23.808  0.37332 79 63.775 <.0001
Sex         -1.161  0.37332 25 -3.109  0.0046
I(age - 11)  0.632  0.06606 79  9.567 <.0001
Sex:I(age - 11) -0.152  0.06606 79 -2.307  0.0237
...

```

As expected, the ML estimates of the random-effects standard deviations are smaller than the corresponding REML estimates. The estimated within-group residual standard deviations are identical, which generally need not occur. In general, the fixed-effects estimates obtained using ML and REML will be similar, though not identical, as in this example. Inferences regarding the fixed effects are essentially the same for the two estimation methods, in this case.

It is instructive, at this point, to compare the individual coefficient estimates obtained with `lmList` to those obtained with `lme`. The function `compareFits` can be used for this. The resulting object has a `plot` method that displays these coefficients side-by-side.

```

> compOrth <-
+   compareFits( coef(fm2Orth.lis), coef(fm1Orth.lme) )
> compOrth
, , (Intercept)
  coef(fm2Orth.lis)  coef(fm1Orth.lme)
M16           23.000      23.078
M05           23.000      23.128
M02           23.375      23.455
...
F04           24.875      24.764
F11           26.375      26.156

, , I(age - 11)
M16            0.550      0.59133

```

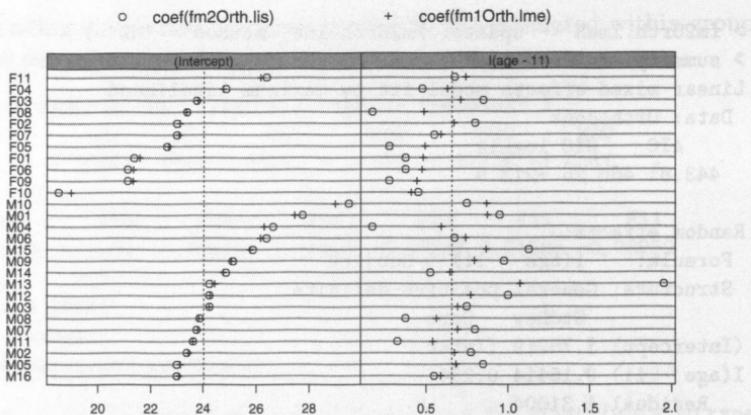


FIGURE 4.8. Individual estimates from an `lmList` fit and from an `lme` fit of the orthodontic distance growth data

```
M05           0.850          0.68579
M02           0.775          0.67469
...
M13           1.950          1.07385
...
F04           0.475          0.63032
F11           0.675          0.74338
```

```
> plot( compOrth, mark = fixef(fm1Orth.lme) ) # Figure 4.8
```

The `mark` argument to the `plot` method indicates points in the horizontal axis where dashed vertical lines should be drawn.

The plots in Figure 4.8 indicate that the individual estimates from the `lme` fit tend to be “pulled toward” the fixed-effects estimates, when compared to the `lmList` estimates. This is typical of linear mixed-effects estimation: the individual coefficient estimates from the `lme` fit represent a compromise between the coefficients from the individual fits corresponding to the `lmList` fit and the fixed-effects estimates, associated with the population averages. For this reason, these estimates are often called *shrinkage estimates*, in the sense that they shrink the individual estimates toward the population average.

The shrinkage toward the fixed effects is particularly noticeable for the slope estimate of subject M13. As pointed out in §4.2, this subject has an outlying orthodontic growth pattern, which leads to an abnormally high estimated slope in the `lm` fit. The pooling of subjects in the `lme` estimation gives a certain amount of robustness to individual outlying behavior. This feature is better illustrated by the comparison of the predicted values from the two fits, which is obtained with the `comparePred` function.

```
> plot( comparePred(fm2Orth.lis, fm1Orth.lme, length.out = 2),
```

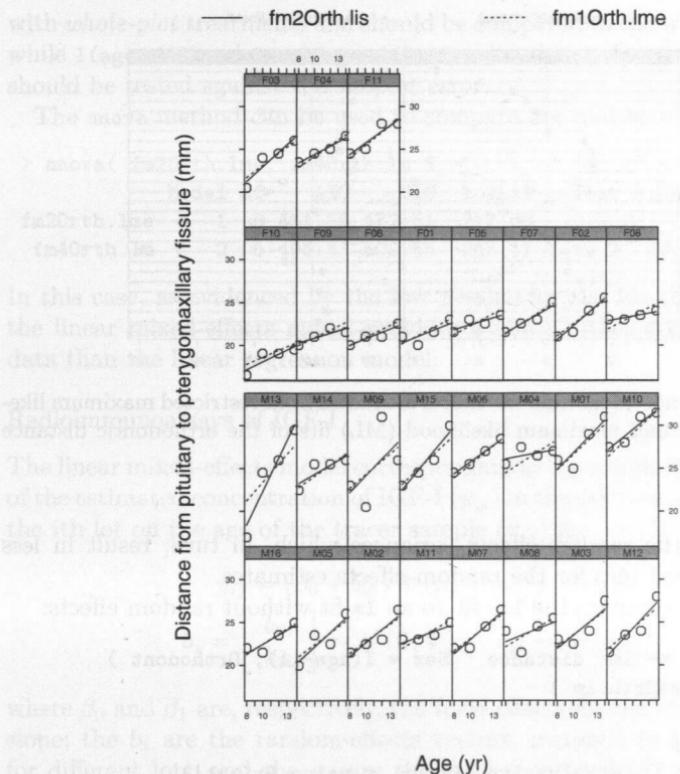


FIGURE 4.9. Individual predicted values from separate `lm` fits and from an `lme` fit of the orthodontic distance growth data

```
+ layout = c(8,4), between = list(y = c(0, 0.5)) ) # Figure 4.9
```

The `length.out` argument specifies the number of predictions for each fitted object. In this case, because the model is a straight line, only two points are needed. The plot of the individual predictions for the `lmList` and `lme` fits, shown in Figure 4.9, clearly indicates the greater sensitivity of the individual `lm` fits to extreme observations.

It is also interesting to compare the `fm2Orth.lme` and the `fmOrth.lmeM` objects, corresponding, respectively, to REML and ML fits of the same model. We compare the estimated random effects for each fit with the `compareFits` function.

```
> plot( compareFits(ranef(fm2Orth.lme), ranef(fm2Orth.lmeM)),
+       mark = c(0, 0) ) # Figure 4.10
```

The ML random-effects estimates tend to be closer to zero than the REML estimates, especially the slope random effects. This will usually occur in mixed-effects models, because REML estimation generally produces larger

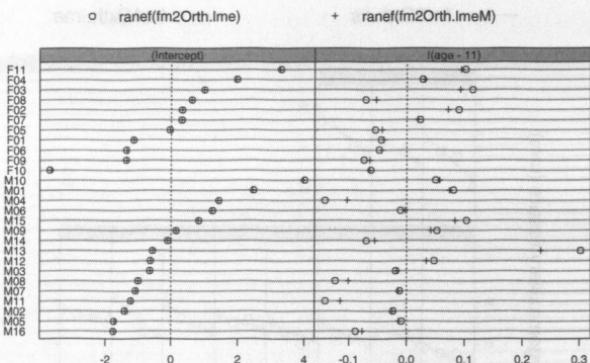


FIGURE 4.10. Individual random-effects estimates from restricted maximum likelihood (REML) and maximum likelihood (ML) fits of the orthodontic distance growth data

estimates for the random-effects variances which, in turn, result in less shrinkage toward zero for the random-effects estimates.

We can also compare the `lme` fit to an `lm` fit without random effects:

```
> fm40rth.lm <- lm( distance ~ Sex * I(age-11), Orthodont )
> summary( fm40rth.lm )
...
Coefficients:
            Value Std. Error t value Pr(>|t|)
(Intercept) 23.808    0.221   107.731  0.000
Sex         -1.161    0.221    -5.251  0.000
I(age - 11)  0.632    0.099     6.394  0.000
Sex:I(age - 11) -0.152    0.099    -1.542  0.126
Residual standard error: 2.26 on 104 degrees of freedom
...
```

The pointwise estimates of the fixed effects are almost identical, but their standard errors are quite different. The `lm` fit has smaller standard errors for the `(Intercept)` and `Sex` fixed effects and larger standard errors for the fixed effects involving `age`. This is because the model used in `lm` ignores the group structure of the data and incorrectly combines the between-group and the within-group variation in the residual standard error. Fixed effects that are associated with invariant factors (factors that do not vary within groups) are actually estimated with less precision than suggested by the `lm` output, because the contribution of the between-group variation to their standard error is larger than that included in the `lm` residual standard error. Conversely, the precision of the fixed effects related to variables that vary within group are less affected by the between-group variation. In the terminology of split-plot experiments, `(Intercept)` and `Sex` are associated

with *whole-plot* treatments and should be compared to the whole-plot error, while *I(age-11)* and *Sex:I(age-11)* are related to *subplot* treatments and should be tested against the subplot error.

The *anova* method can be used to compare *lme* and *lm* objects.

```
> anova( fm20rth.lme, fm40rth.lm )
    Model df   AIC   BIC logLik  Test L.Ratio p-value
fm20rth.lme     1  8 451.35 472.51 -217.68
fm40rth.lm      2  5 496.33 509.55 -243.17 1 vs 2  50.977 <.0001
```

In this case, as evidenced by the low *p*-value for the likelihood ratio test, the linear mixed-effects model provides a much better description of the data than the linear regression model.

Radioimmunoassays of IGF-I

The linear mixed-effects model corresponding to the simple linear regression of the estimated concentration of IGF-I (y_{ij}) in the j th tracer sample within the i th lot on the age of the tracer sample (x_{ij}) is

$$\begin{aligned} y_{ij} &= (\beta_0 + b_{0i}) + (\beta_1 + b_{1i}) x_{ij} + \epsilon_{ij}, \\ b_i &= \begin{bmatrix} b_{0i} \\ b_{1i} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \Psi), \quad \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2), \end{aligned} \quad (4.2)$$

where β_0 and β_1 are, respectively, the fixed effects for the intercept and the slope; the b_i are the random-effects vectors, assumed to be independent for different lots; and the ϵ_{ij} are the independent, identically distributed within-group errors, assumed to be independent of the random effects.

We fit the linear mixed-effects model (4.2) with

```
> fm1IGF.lme <- lme( fm1IGF.lis )
> fm1IGF.lme
Linear mixed-effects model fit by REML
  Data: IGF
  Log-restricted-likelihood: -297.18
  Fixed: conc ~ age
  (Intercept)      age
        5.375 -0.0025337
```

```
Random effects:
Formula: ~ age | Lot
Structure: General positive-definite
          StdDev   Corr
(Intercept) 0.0823594 (Inter
                  age 0.0080862 -1
Residual    0.8206310
```

```
Number of Observations: 237
Number of Groups: 10
```

The fixed-effects estimates are similar to the ones obtained with the single `lm` fit of §4.1.1. The within-group standard errors are also similar in the two fits, which suggests that not much is gained by incorporating random effects into the model. The estimated correlation between the random effects ($\simeq -1$) gives a clear indication that the estimated random-effects covariance matrix is ill-conditioned, suggesting that the model may be overparameterized. The confidence intervals for the standard deviations and correlation coefficient reinforce the indication of overparameterization.

```
> intervals( fm1IGF.lme )
Approximate 95% confidence intervals

Fixed effects:
            lower      est.      upper
(Intercept) 5.163178 5.3749606 5.5867427
age         -0.012471 -0.0025337 0.0074039

Random Effects:
Level: Lot
            lower      est.      upper
sd((Intercept)) 0.0011710 0.0823594 5.792715
sd(age)        0.0013177 0.0080862 0.049623
cor((Intercept),age) -1.0000000 -0.9999640 1.000000

Within-group standard error:
            lower      est.      upper
0.7212 0.82063 0.93377
```

The 95% confidence interval for the correlation coefficient covers all possible values for this parameter. There is also evidence of large variability in the estimates of the `(Intercept)` and `age` standard deviations. These issues are explored in more detail in §4.3.

The primary question of interest in the IGF-I study is whether the tracer decays with age. We can investigate it with the `summary` method.

```
> summary( fm1IGF.lme )
...
Fixed effects: conc ~ age
            Value Std.Error DF t-value p-value
(Intercept) 5.3750  0.10748 226  50.011 <.0001
age        -0.0025  0.00504 226   -0.502  0.6159
...
```

As with the `lm` results of §4.1.1, there is no significant evidence of tracer decay with age, for the 50-day period in which the observations were collected. Note that the standard errors for the estimates are very similar to the ones in the `lm` fit.

The plots of the coefficient estimates corresponding to `fm1IGF.lis` and `fm1IGF.lme` are shown in Figure 4.11. Once again we observe a “shrinkage

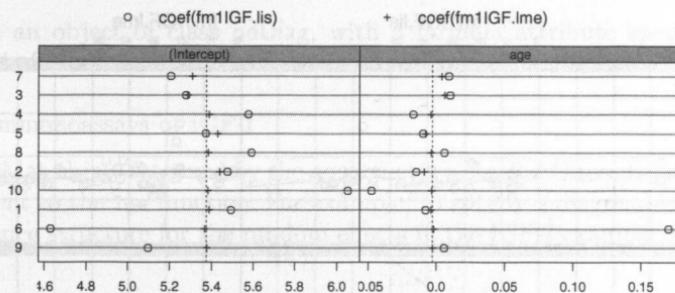


FIGURE 4.11. Individual estimates from separate `lm` fits and from an `lme` fit of the IGF-I radioimmunoassays data.

toward the population mean" pattern for the individual `lme` coefficients. The IGF-I data contain several outlying observations and the dramatic shrinkage in the coefficient estimates observed for some of the lots reflects the greater robustness of the `lme` fit. This is better illustrated by comparing the individual predictions under each fit, as presented in Figure 4.12. The differences in the predicted values for the two fits are particularly dramatic for lots 6 and 10, both of which have observations only over a very limited time range. For lot 6 a single low observation at one of the earliest times causes a dramatic change in the estimate of both the slope and intercept when this lot is fit by itself. When it is combined with the other lots in a mixed-effects model the effect of this single observation is diminished. Also notice that the outlying observations for lots 4, 3, and 7 have very little effect on the parameter estimates because in each of these lots there are several other observations at times both above and below the times of the aberrant observations.

4.2.2 Patterned Variance-Covariance Matrices for the Random Effects: The `pdMat` Classes

The models considered in §4.2.1 do not assume any special form for the random-effects variance-covariance matrix Ψ . In many practical applications, however, we will wish to restrict Ψ to special forms of variance-covariance matrices that are parameterized by fewer parameters. For example, we may be willing to assume that the random effects are independent, in which case Ψ would be diagonal, or that, in addition to being independent, they have the same variance, in which case Ψ would be a multiple of the identity matrix.

The `nlme` library provides several classes of positive-definite matrices, the `pdMat` classes, that are used to specify patterned variance-covariance matrices for the random effects. Table 4.3 lists the standard `pdMat` classes included in the `nlme` library. The default class of positive-definite matrix

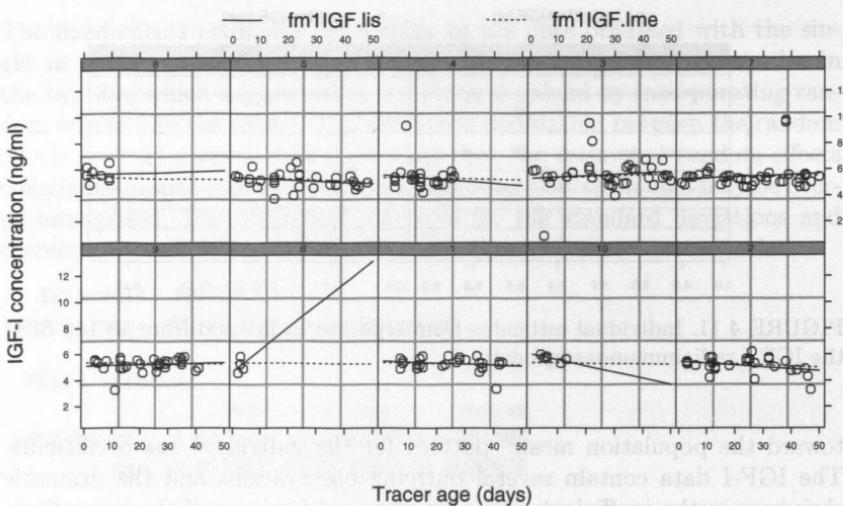


FIGURE 4.12. Individual predicted values from separate `lm` fits and from an `lme` fit of the IGF-I radioimmunoassays data.

TABLE 4.3. Standard `pdMat` classes.

<code>pdBlocked</code>	block-diagonal
<code>pdCompSymm</code>	compound-symmetry structure
<code>pdDiag</code>	diagonal
<code>pdIdent</code>	multiple of an identity
<code>pdSymm</code>	general positive-definite matrix

for the random effects in the `nlme` library is `pdSymm`, corresponding to a general symmetric positive-definite matrix.

A function that creates an object of a given class is called a *constructor* for that class. The `pdMat` constructors have the same name as their corresponding classes so, for example, the constructor for the `pdDiag` class is also called `pdDiag`.

Because initial values for Ψ can be derived internally in the `lme` function, the `pdMat` constructors are typically used only to specify a `pdMat` class and a formula for the random-effects model. For example,

```
> pd1 <- pdDiag( ~ age )
> pd1
Uninitialized positive definite matrix structure of class pdDiag
> formula( pd1 )
~ age
```

creates an object of class `pdDiag`, with a formula attribute specifying a random-effects model, but with no initial value assigned to the matrix.

Radioimmunoassays of IGF-I

The `pdMat` object returned by the constructor is passed through the `random` argument to the `lme` function. For example, to specify a diagonal variance-covariance structure for the random effects in the IGF-I example of §4.2.1, we use

```
> fm2IGF.lme <- update( fm1IGF.lme, random = pdDiag(~age) )
> fm2IGF.lme
Linear mixed-effects model fit by REML
  Data: IGF
  Log-restricted-likelihood: -297.4
  Fixed: conc ~ age
    (Intercept)      age
      5.369 -0.0019301

Random effects:
  Formula: ~ age | Lot
  Structure: Diagonal
    (Intercept)      age Residual
  StdDev:  0.00031074 0.0053722  0.8218
  . . .
```

With the exception of the standard deviation for the `(Intercept)` random effect, all estimates are similar to the ones in `fm1IGF.lme`. We can compare the two fits with

```
> anova( fm1IGF.lme, fm2IGF.lme )
   Model df   AIC   BIC logLik  Test L.Ratio p-value
fm1IGF.lme     1 6 606.37 627.12 -297.18
fm2IGF.lme     2 5 604.80 622.10 -297.40 1 vs 2 0.43436 0.5099
```

The large *p*-value for the likelihood ratio test and the smaller AIC and BIC values for the simpler model `fm2IGF.lme` indicate that it should be preferred.

Because `IGF` is a `groupedData` object, the grouping structure does not need to be given explicitly in `random`. In cases when both the grouping structure and a `pdMat` class are to be declared in `random`, we use a *named list*, with the name specifying the grouping factor.

```
> update( fm1IGF.lme, random = list(Lot = pdDiag(~ age)) )
```

The `value` argument to the constructor is used to assign a value to the positive-definite matrix. In this case, the random-effects formula needs to be specified through the `form` argument.

```
> pd2 <- pdDiag( value = diag(2), form = ~ age )
```

```
> pd2
Positive definite matrix structure of class pdDiag representing
 [,1] [,2]
[1,]    1    0
[2,]    0    1
> formula( pd2 )
~ age
```

This can be used to provide initial values for the scaled variance–covariance matrix of the random effects, $D = \Psi/\sigma^2$ in the `lme` call.

```
> lme( conc ~ age, IGF, pdDiag(diag(2), ~age) )
```

Split-Plot Experiment on Varieties of Oats

We now revisit the `Oats` example of §1.6 and describe alternative ways of analyzing the split-plot data using `pdMat` classes. The final mixed-effects model resulting from the analysis presented in that section is

$$\begin{aligned} y_{ijk} &= \beta_0 + \beta_1 N_k + b_i + b_{i,j} + \epsilon_{ijk}, \quad i = 1, \dots, 6, \\ &\quad j = 1, \dots, 3, \quad k = 1, \dots, 4, \\ \epsilon_{ijk} &\sim \mathcal{N}(0, \sigma^2), \quad b_i \sim \mathcal{N}(0, \sigma_1^2), \quad b_{i,j} \sim \mathcal{N}(0, \sigma_2^2), \end{aligned} \quad (4.3)$$

where i indexes the *Blocks*, j indexes the *Varieties*, and k indexes the *Nitrogen* concentrations N_k . The intercept is represented by β_0 , the Nitrogen slope by β_1 and the yield by y_{ijk} . The b_i denote the Block random effects, the $b_{i,j}$ denote the Variety within Block random effects, and the ϵ_{ijk} denote the within-group errors. This is an example of a two-level mixed-effects model, with the $b_{i,j}$ random effects *nested* within the b_i random effects.

The multilevel model capabilities of `lme` were used in §1.6 to fit (4.3). We recommend fitting the model this way, as it uses efficient computational algorithms designed specifically for this type of model. Nevertheless, to further illustrate the use of the `pdMat` classes, we consider equivalent single-level representations of the same model.

By defining

$$\begin{aligned} \mathbf{y}_i &= \begin{bmatrix} y_{i11} \\ y_{i12} \\ \vdots \\ y_{i34} \end{bmatrix}, \quad \boldsymbol{\epsilon}_i = \begin{bmatrix} \epsilon_{i11} \\ \epsilon_{i12} \\ \vdots \\ \epsilon_{i34} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \quad \mathbf{b}_i^* = \begin{bmatrix} b_i + b_{i,1} \\ b_i + b_{i,2} \\ b_i + b_{i,3} \end{bmatrix}, \\ \mathbf{X}_i &= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & N_1 \\ 1 & N_2 \\ 1 & N_3 \\ 1 & N_4 \end{bmatrix}, \quad \mathbf{Z}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \end{aligned}$$

with \otimes denoting the Kronecker product, we can rewrite (4.3) as the single-level model

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i^* + \boldsymbol{\epsilon}_i, \quad \mathbf{b}_i^* \sim \mathcal{N}(\mathbf{0}, \Psi^*), \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \quad (4.4)$$

where

$$\Psi^* = \begin{bmatrix} \sigma_1^2 + \sigma_2^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma_2^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma_2^2 \end{bmatrix}.$$

The Ψ^* matrix has a compound symmetry structure, represented in `nlme` by the `pdCompSymm` class. We fit (4.4) with

```
> fm40atsB <- lme( yield ~ nitro, data = Oats,
+                      random =list(Block = pdCompSymm(~ Variety - 1)))
> summary(fm40atsB)
Linear mixed-effects model fit by REML
Data: Oats
AIC      BIC    logLik
603.04  614.28 -296.52

Random effects:
Formula: ~ Variety - 1 | Block
Structure: Compound Symmetry
          StdDev   Corr
VarietyGolden Rain 18.208
VarietyMarvellous 18.208 0.635
VarietyVictory     18.208 0.635 0.635
Residual           12.867

Fixed effects: yield ~ nitro
              Value Std.Error DF t-value p-value
(Intercept) 81.872    6.9453 65 11.788 <.0001
nitro       73.667    6.7815 65 10.863 <.0001

Correlation:
  (Intr)
nitro -0.293

Standardized Within-Group Residuals:
    Min      Q1      Med      Q3     Max
-1.7438 -0.66475 0.017103 0.54299 1.803
```

Comparing this with the output of `summary(fm40ats)` in §1.6, we see that, except for the random-effects variance–covariance components, the results are nearly identical.

Verifying the equivalence of the random-effects variance–covariance components requires some extra work. Note that the variance in the compound symmetric matrix Ψ^* is $\sigma_{b^*}^2 = \sigma_1^2 + \sigma_2^2$ and the correlation is

$\rho = \sigma_1^2 / (\sigma_1^2 + \sigma_2^2)$. Therefore, $\sigma_1^2 = \rho\sigma_{b^*}^2$ and $\sigma_2^2 = \sigma_{b^*}^2 - \sigma_1^2$. We can then derive the REML estimates of σ_1 and σ_2 from `fm40atsB`

$$\hat{\sigma}_1 = \sqrt{0.63471} \times 18.208 = 14.506, \quad \hat{\sigma}_2 = \sqrt{18.208^2 - 14.506^2} = 11.005,$$

verifying that they are identical to the estimates corresponding to `fm40ats` in §1.6. Because the REML estimate of ρ in the `summary(fm40atsB)` output was displayed with only three decimal places, we used

```
> corMatrix( fm40atsB$modelStruct$reStruct$Block )[1,2]
[1] 0.63471
```

to obtain the more accurate $\hat{\rho}$ used to obtain $\hat{\sigma}_1$ and $\hat{\sigma}_2$.

Yet another representation of (4.3) as a single-level model is obtained by defining

$$\mathbf{b}_i^{**} = \begin{bmatrix} b_i \\ b_{i,1} \\ b_{i,2} \\ b_{i,3} \end{bmatrix}, \quad \mathbf{W}_i = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$\Psi^{**} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_2^2 & 0 \\ 0 & 0 & 0 & \sigma_2^2 \end{bmatrix}$$

and writing

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{W}_i \mathbf{b}_i^{**} + \boldsymbol{\epsilon}_i, \quad \mathbf{b}_i^{**} \sim \mathcal{N}(\mathbf{0}, \Psi^{**}), \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}). \quad (4.5)$$

The parameters in (4.3) are equivalent to those in (4.5).

The Ψ^{**} matrix is structured as a block diagonal matrix, with blocks σ_1^2 and $\sigma_2^2 \mathbf{I}$. The `pdBlocked` class is used to represent block diagonal matrices in the `nlme` library. It takes as an argument a list of `pdMat` objects, specifying the different blocks in the order they appear in the main diagonal of the corresponding block-diagonal matrix. In the case of Ψ^{**} , both blocks are expressed as multiples of the identity matrix, represented by the `pdIdent` class in `nlme`. We can then fit (4.5) with

```
> fm40atsC <- lme( yield ~ nitro, data = Oats,
+   random=list(Block=pdBlocked(list(pdIdent(~ 1),
+                                pdIdent(~ Variety-1)))))
```

`fm40atsC`

```
> summary( fm40atsC )
Linear mixed-effects model fit by REML
Data: Oats
    AIC      BIC    logLik
 603.04  614.28 -296.52
```

Random effects:

```
Composite Structure: Blocked

Block 1: (Intercept)
Formula: ~ 1 | Block

(Intercept)
StdDev: 14.505

Block 2: VarietyGolden Rain, VarietyMarvellous, VarietyVictory
Formula: ~ Variety - 1 | Block
Structure: Multiple of an Identity
    VarietyGolden Rain VarietyMarvellous VarietyVictory
StdDev:           11.005          11.005          11.005
    Residual
StdDev: 12.867

Fixed effects: yield ~ nitro
      Value Std.Error DF t-value p-value
(Intercept) 81.872    6.9451 65 11.788 <.0001
      nitro 73.667    6.7815 65 10.863 <.0001
Correlation:
  (Intr)
nitro -0.293
  . . .
```

Comparing this to the output of `summary(fm40ats)` in §1.6 we verify that, as expected, the two fits are nearly identical.

Cell Culture Bioassay with Crossed Random Effects

Data grouped according to *crossed* classification factors induce a variance-covariance structure for the observations which can be flexibly represented by mixed-effects models with crossed random effects. These models can also be fit with the `lme` function. However, unlike in the case of nested random effects, the underlying estimation algorithm is not optimized to take full advantage of the sparse structure of design matrices for crossed random effects.

The crossed random-effects structure is represented in `lme` by a combination of `pdBlocked` and `pdIdent` objects. We illustrate its use with an example of a cell culture plate bioassay conducted at Searle, Inc. The data, courtesy of Rich Wolfe and David Lansky, come from a bioassay run on a cell culture plate with two blocks of 30 wells each. The wells in each block are labeled according to six rows and five columns, corresponding to a crossed classification. Within each block, six different samples are randomly assigned to rows and five serial dilutions are randomly assigned to columns. The response variable is the logarithm of the optical density measure on a well. The cells are treated with a compound that they metabolize to produce the stain. Only live cells can make the stain, so the optical density is a

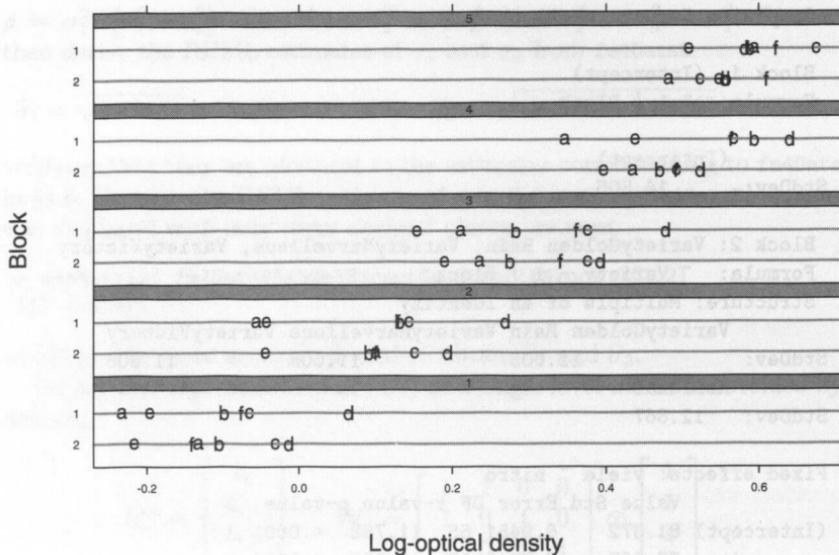


FIGURE 4.13. Log-optical density measured on the central 60 wells of a cell culture plate. The wells are divided into two blocks with six rows and five columns, with samples being assigned to rows and dilutions to columns. Panels in the plot correspond to the serial dilutions and symbols refer to the samples.

measure of the number of cells that are alive and healthy. These data are described in detail in Appendix A.2 and are included in the `nlme` library as the `groupedData` object `Assay`.

The plot of the log-optical densities, displayed in Figure 4.13, indicates that the response increases with dilution and is generally lower for treatments `a` and `e`. There does not appear to be any interactions between sample and dilution.

A full factorial model is used to represent the fixed effects and three random effects are used to account for block, row, and column effects, with the last two random effects nested within block, but crossed with each other. The corresponding mixed-effects model for the log-optical density y_{ijk} in the j th row, k th column of the i th block, for $i = 1, \dots, 2$, $j = 1, \dots, 6$, $k = 1, \dots, 5$, is

$$y_{ijk} = \mu + \alpha_j + \beta_k + \gamma_{jk} + b_i + r_{ij} + c_{ik} + \epsilon_{ijk}, \\ b_i \sim \mathcal{N}(0, \sigma_1^2), \quad r_{ij} \sim \mathcal{N}(0, \sigma_2^2), \quad c_{ik} \sim \mathcal{N}(0, \sigma_3^2), \quad \epsilon_{ijk} \sim \mathcal{N}(0, \sigma^2). \quad (4.6)$$

The fixed effects in (4.6) are μ , the grand mean, α_j and β_k , the *sample* and *dilution* main effects, and γ_{jk} , the sample-dilution interaction. To ensure identifiability of the fixed effects, it is conventioned that $\alpha_1 = \beta_1 = \gamma_{1k} =$

$\gamma_{j1} = 0$, for $j = 1, \dots, 6$, $k = 1, \dots, 5$. The random effects in (4.6) are b_i , the *block* random effect, r_{ij} , the *row within block* random effect, and c_{ik} , the *column within block* random effect. All random effects are assumed independent of each other and independent of the within-groups errors ϵ_{ijk} .

The crossed random-effects structure in (4.6) can alternatively be represented as the random-effects structure corresponding to a single-level model, with *block* as the single grouping variable, and random-effects vector $\mathbf{b}_i = (b_i, r_{i1}, \dots, r_{i6}, c_{i1}, \dots, c_{i5})^T$, $i = 1, \dots, 2$, with

$$\text{Var}(\mathbf{b}_i) = \begin{bmatrix} \sigma_1^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_3^2 \mathbf{I} \end{bmatrix}.$$

That is, \mathbf{b}_i has a block diagonal variance-covariance matrix, with diagonal blocks given by multiples of the identity matrix. This type of variance-covariance structure is represented in S by a pdBlocked object with pdIdent elements. We fit the linear mixed-effects model (4.6) with lme as

```
> ## establishing the desired parameterization for contrasts
> options( contrasts = c("contr.treatment", "contr.poly") )
> fm1Assay <- lme( logDens ~ sample * dilut, Assay,
+   random = pdBlocked(list(pdIdent(~ 1), pdIdent(~ sample - 1),
+   pdIdent(~ dilut - 1))) )
> fm1Assay
Linear mixed-effects model fit by REML
Data: Assay
Log-restricted-likelihood: 38.536
Fixed: logDens ~ sample * dilut
(Intercept) sampleb samplec sampled samplee samplef dilut2
-0.18279 0.080753 0.13398 0.2077 -0.023672 0.073569 0.20443
dilut3 dilut4 dilut5 samplebdilut2 samplecdilut2
0.40586 0.57319 0.72064 0.0089389 -0.0084953
sampleddilut2 sampleddilut2 samplefdilut2 samplebdilut3
0.0010793 -0.041918 0.019352 -0.025066
samplecdilut3 sampleddilut3 sampleedilut3 samplefdilut3
0.018645 0.0039886 -0.027713 0.054316
samplebdilut4 samplecdilut4 sampleddilut4 sampleedilut4
0.060789 0.0052598 -0.016486 0.049799
samplefdilut4 samplebdilut5 samplecdilut5 sampleddilut5
0.063372 -0.045762 -0.072598 -0.17776
sampleddilut5 samplefdilut5
0.013611 0.0040234

Random effects:
Composite Structure: Blocked
Block 1: (Intercept)
Formula: ~ 1 | Block
```

```

(Intercept)
StdDev: 0.0098084

Block 2: samplea, sampleb, samplec, sampled, samplee, samplef
Formula: ~ sample - 1 | Block
Structure: Multiple of an Identity
      samplea sampleb samplec sampled samplee samplef
StdDev: 0.025289 0.025289 0.025289 0.025289 0.025289 0.025289

Block 3: dilut1, dilut2, dilut3, dilut4, dilut5
Formula: ~ dilut - 1 | Block
Structure: Multiple of an Identity
      dilut1   dilut2   dilut3   dilut4   dilut5
StdDev: 0.0091252 0.0091252 0.0091252 0.0091252 0.0091252
      Residual
StdDev: 0.041566

Number of Observations: 60
Number of Groups: 2

```

The REML estimates of the standard deviation components in this example are $\hat{\sigma}_1 = 0.0098$, $\hat{\sigma}_2 = 0.0253$, $\hat{\sigma}_3 = 0.0091$, and $\hat{\sigma} = 0.0416$.

The primary question of interest for this experiment is whether there are significant differences among the fixed effects, which we investigate with

```

> anova( fm1Assay )
      numDF denDF F-value p-value
(Intercept)     1     29  538.03 <.0001
      sample       5     29   11.21 <.0001
      dilut        4     29  420.80 <.0001
sample:dilut    20     29    1.61  0.1193

```

As suggested by Figure 4.13, there are significant differences among samples and among dilutions, but no significant interaction between the two factors.

The small estimated standard deviations in the `fm1Assay` fit suggest that some, or perhaps all, of the random effects can be eliminated from (4.6). However, because our purpose here is just to illustrate the use of `lme` with crossed random effects, we do not pursue the analysis of the `Assay` data any further.

New `pdMat` classes, representing user-defined positive-definite matrix structures, can be added to the set of standard classes in Table 4.3 and used with the `lme` and `nlme` functions. For this, one must specify a constructor function, generally with the same name as the class, and, at a minimum, methods for the functions `pdConstruct`, `pdMatrix`, and `coef`. The `pdDiag` constructor and methods can serve as templates for these.

4.2.3 Fitting Multilevel Models

Linear mixed-effects models with nested grouping factors, generally called multilevel models (Goldstein, 1995) or hierarchical linear models (Bryk and Raudenbush, 1992), can be fitted with the `lme` function, just like single-level models. The only difference is in the specification of the `random` argument, which must provide information about the nested grouping structure and the random-effects model at each level of grouping. We describe the multi-level model capabilities in `lme` through the analyses of two examples from Integrated Circuiting (IC) manufacturing.

Thickness of Oxide Coating on a Semiconductor

Littell et al. (1996) describe data from a passive data collection study in the IC industry in which the thickness of the oxide coating layer was measured on three randomly selected *sites* in each of three *wafers* from each of eight *lots* randomly selected from the population of lots. There are two nested grouping levels in this example: *lot* and *wafer within lot*. The objective of the study was to estimate the variance components associated with the different levels of nesting and the within-group error, to evaluate assignable causes of variability in the oxide deposition process. These data are also described in Appendix A.20 and are included in the `groupedData` object `Oxide` in the `nlme` library.

The plot of the data, shown in Figure 4.14, suggests that the lot-to-lot variability of the oxide layer thickness is greater than the wafer-to-wafer variability within a lot, which, in turn, is greater than the site-to-site variation within a wafer.

A multilevel model to describe the oxide thickness y_{ijk} measured on the k th site of the j th wafer within the i th lot is

$$\begin{aligned} y_{ijk} &= \mu + b_i + b_{i,j} + \epsilon_{ijk}, \quad i = 1, \dots, 8, \quad j, k = 1, 2, 3, \\ b_i &\sim \mathcal{N}(0, \sigma_1^2), \quad b_{i,j} \sim \mathcal{N}(0, \sigma_2^2), \quad \epsilon_{ijk} \sim \mathcal{N}(0, \sigma^2), \end{aligned} \quad (4.7)$$

where the *lot* random effects b_i are assumed to be independent for different i , the *wafer within lot* random effects $b_{i,j}$ are assumed to be independent for different i and j and to be independent of the b_i , and the within-group errors ϵ_{ijk} are assumed to be independent for different i , j , and k and to be independent of the random effects.

The most general form of the argument `random` when `lme` is used to fit a multilevel model is as a *named list* where the names define the grouping factors and the formulas describe the random-effects models at each level. The order of nesting is taken to be the order of the elements in the list, with the outermost level appearing first. In the case of (4.7) we write

```
random = list( Lot = ~ 1, Wafer = ~ 1 )
```

When the random-effects formulas are the same for all levels of grouping, we can replace the named list by a one-sided formula with the common

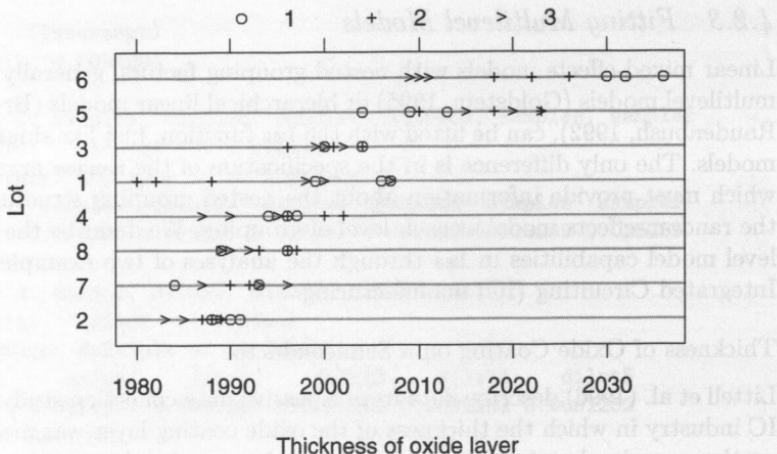


FIGURE 4.14. Thickness of oxide layer measured on different sites of wafers selected from a sample of manufacturing lots. Symbols denote different wafers within the same lot.

random-effects formula and an expression defining the grouping structure separated by a `|` operator.

```
random = ~ 1 | Lot/Wafer
```

Because `Oxide` contains this grouping structure in its display formula

```
> formula( Oxide )
Thickness ~ 1 | Lot/Wafer
```

the grouping structure expression can be omitted from `random`. In fact, because, by default, `random` is equal to the right-hand side of the `fixed` formula, `~1` in this case, we can omit `random` all together from the `lme` call and fit the model with

```
> fm1Oxide <- lme( Thickness ~ 1, Oxide )
> fm1Oxide
Linear mixed-effects model fit by REML
Data: Oxide
Log-restricted-likelihood: -227.01
Fixed: Thickness ~ 1
(Intercept)
2000.2

Random effects:
Formula: ~ 1 | Lot
(Intercept)
StdDev: 11.398
```

```
Formula: ~ 1 | Wafer %in% Lot
          (Intercept) Residual
StdDev:      5.9888   3.5453
```

Number of Observations: 72

Number of Groups:

Lot	Wafer	%in%	Lot
8			24

The REML estimates of the variance components are the square of the standard deviations in the `fm1Oxide` output: $\hat{\sigma}_1^2 = 11.398^2 = 129.91$, $\hat{\sigma}_2^2 = 5.9888^2 = 35.866$, and $\hat{\sigma}^2 = 3.5453^2 = 12.569$. We can assess the variability in these estimates with the `intervals` method.

```
> intervals( fm1Oxide, which = "var-cov" )
```

Approximate 95% confidence intervals

Random Effects:

Level: Lot

	lower	est.	upper
sd((Intercept))	5.0277	11.398	25.838

Level: Wafer

	lower	est.	upper
sd((Intercept))	3.4615	5.9888	10.361

Within-group standard error:

	lower	est.	upper
	2.6719	3.5453	4.7044

All intervals are bounded well away from zero, indicating that the two random effects should be kept in (4.7). We can test, for example, if the *wafer within lot* random effect can be eliminated from the model with

```
> fm2Oxide <- update( fm1Oxide, random = ~ 1 | Lot)
> anova( fm1Oxide, fm2Oxide )
Model df     AIC     BIC logLik  Test L.Ratio p-value
fm1Oxide    1  4 462.02 471.07 -227.01
fm2Oxide    2  3 497.13 503.92 -245.57 1 vs 2     37.11  <.0001
```

The very high value of the likelihood ratio test statistic confirms the significance of that term in the model.

As with single-level fits, estimated BLUPs of the individual coefficients are obtained using `coef`, but, because of the multiple grouping levels, a `level` argument is used to specify the desired grouping level. For example, to get the estimated average oxide layer thicknesses by lot, we use

```
> coef( fm1Oxide, level = 1 )
          (Intercept)
1        1996.7
2        1988.9
```

```

3      2001.0
4      1995.7
5      2013.6
6      2019.6
7      1992.0
8      1993.8

```

while the estimated average thicknesses per wafer are obtained with

```

> coef( fm1Oxide, level = 2 )          # default when level is omitted
   (Intercept)
1/1    2003.2
1/2    1984.7
1/3    2001.1
...
8/2    1995.2
8/3    1990.7

```

The `level` argument is used similarly with the methods `fitted`, `predict`, `ranef`, and `resid`, with the difference that multiple levels can be simultaneously specified. For example, to get the estimated random effects at both grouping levels we use

```

> ranef( fm1Oxide, level = 1:2 )
Level: Lot
   (Intercept)
1     -3.46347
2    -11.22164
...
8    -6.38538

Level: Wafer %in% Lot
   (Intercept)
1/1    6.545993
1/2   -11.958939
...
8/3   -3.074863

```

These methods are further illustrated in §4.3, when we describe tools for assessing the adequacy of fitted models.

Manufacturing of Analog MOS Circuits

The `Wafer` data, introduced in §3.3.4 and shown in Figure 4.15, provide another example of multilevel data in IC manufacturing and are used here to illustrate the capabilities in `lme` when covariates are used in a multilevel model. As described in Appendix A.30, these data come from an experiment conducted at the Microelectronics Division of Lucent Technologies to study different sources of variability in the manufacturing of analog MOS circuits. The intensity of current (in mA) at 0.8, 1.2, 1.6, 2.0, and 2.4

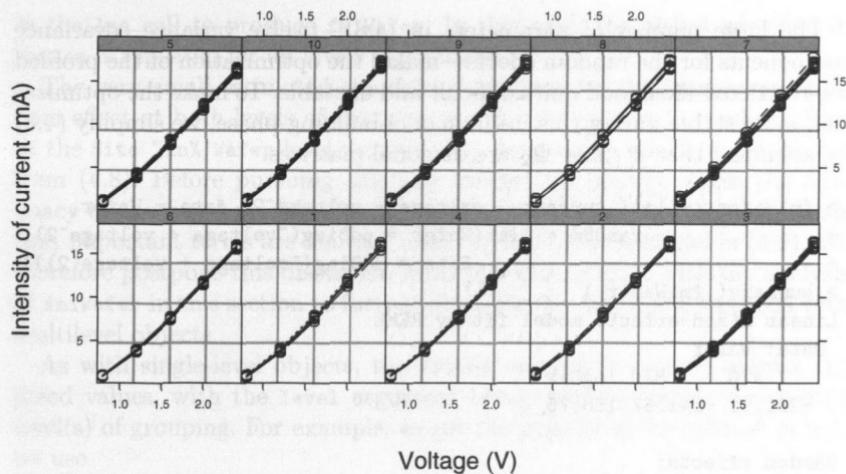


FIGURE 4.15. Current versus voltage curves for each site, by wafer.

V was measured on $80\mu\text{m} \times 0.6\mu\text{m}$ n -channel devices. Measurements were made on eight sites in each of ten wafers selected from the same lot. Two levels of nesting are present in these data: *wafer* and *site within wafer*. The main objective of the experiment was to construct an empirical model for simulating the behavior of similar circuits.

From Figure 4.15, it appears that current can be modeled as a quadratic function of voltage. We initially consider a *full* mixed-effects model, with all terms having random effects at both the *wafer* and the *site within wafer* levels. The corresponding multilevel model for the intensities of current y_{ijk} at the k th level of voltage v_k in the j th site within the i th wafer is expressed, for $i = 1, \dots, 10$, $j = 1, \dots, 8$, and $k = 1, \dots, 5$ as

$$\begin{aligned}
 y_{ijk} &= (\beta_0 + b_{0i} + b_{0i,j}) + (\beta_1 + b_{1i} + b_{1i,j}) v_k + (\beta_2 + b_{2i} + b_{2i,j}) v_k^2 + \epsilon_{ijk}, \\
 \mathbf{b}_i &= \begin{bmatrix} b_{0i} \\ b_{1i} \\ b_{2i} \end{bmatrix} \sim \mathcal{N}(0, \Psi_1), \quad \mathbf{b}_{i,j} = \begin{bmatrix} b_{0i,j} \\ b_{1i,j} \\ b_{2i,j} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \Psi_2), \\
 \epsilon_{ijk} &\sim \mathcal{N}(0, \sigma^2).
 \end{aligned} \tag{4.8}$$

The parameters β_0 , β_1 , and β_2 are the fixed effects in the quadratic model, \mathbf{b}_i is the *wafer-level* random-effects vector, $\mathbf{b}_{i,j}$ is the *site within wafer-level* random-effects vector, and ϵ_{ijk} is the within-group error. As usual, the \mathbf{b}_i are assumed to be independent for different i , the $\mathbf{b}_{i,j}$ are assumed to be independent for different i, j and independent of the \mathbf{b}_i , and the ϵ_{ijk} are assumed to be independent for different i, j, k and independent of the random effects.

The large number of parameters in (4.8)—twelve variance-covariance components for the random effects—makes the optimization of the profiled log-restricted-likelihood quite difficult and unstable. To make the optimization more stable during this initial model-building phase, we simplify (4.8) by assuming that Ψ_1 and Ψ_2 are diagonal matrices.

```
> fm1Wafer <- lme( current ~ voltage + voltage^2, data = Wafer,
+                     random = list(Wafer = pdDiag(~voltage + voltage^2),
+                                         Site = pdDiag(~voltage + voltage^2)))
> summary(fm1Wafer)
Linear mixed-effects model fit by REML
Data: Wafer
      AIC      BIC logLik
-281.51 -241.67 150.75

Random effects:
Formula: ~voltage + voltage^2 | Wafer
Structure: Diagonal
            (Intercept) voltage I(voltage^2)
StdDev:  0.00047025 0.18717    0.025002

Formula: ~voltage + voltage^2 | Site %in% Wafer
Structure: Diagonal
            (Intercept) voltage I(voltage^2) Residual
StdDev:  0.00038085 0.13579   1.5202e-05 0.11539

Fixed effects: current ~ voltage + voltage^2
                Value Std.Error DF t-value p-value
(Intercept) -4.4612  0.051282 318 -86.992 <.0001
voltage      5.9034  0.092700 318  63.683 <.0001
I(voltage^2)  1.1704  0.022955 318  50.987 <.0001

Correlation:
          (Intr) voltage
voltage -0.735
I(voltage^2) 0.884 -0.698

Standardized Within-Group Residuals:
      Min        Q1        Med        Q3       Max
-1.8967 -0.53534  0.024858  0.79853  1.7777

Number of Observations: 400
Number of Groups:
Wafer Site %in% Wafer
     10           80

Because Wafer is a groupedData object and the random-effects model is identical for both levels of grouping, we could have used

random = pdDiag(~voltage + voltage^2)
```

in the `lme` call to produce `fm1Wafer`. In this case, the object specified in `random` is repeated for all levels of grouping.

The very small estimated standard deviations for the `(Intercept)` random effect at both levels of grouping and for the `voltage^2` random effect at the `Site %in% Wafer` level suggest that these terms could be eliminated from (4.8). Before pursuing this any further, we should assess the adequacy of the fitted model. This is considered in detail in §4.3 and reveals that important terms are omitted from the fixed-effects model in (4.8). We therefore postpone this discussion until §4.3 and proceed with the analysis of `fm1Wafer` in this section to further illustrate the use of `lme` methods with multilevel objects.

As with single-level objects, the `fitted` method is used to extract the fitted values, with the `level` argument being used to specify the desired level(s) of grouping. For example, to get the population level fitted values, we use

```
> fitted( fm1Wafer, level = 0 )
   1     1     1     1     1     1     1     1     1
  1.0106 4.3083 7.9805 12.027 16.448 1.0106 4.3083 7.9805 12.027
  .
  .
  10    10    10    10
 4.3083 7.9805 12.027 16.448
attr(, "label"):
[1] "Fitted values (mA)"
```

Similarly, residuals are extracted using the `resid` method. For example, the `Wafer` and `Site %in% Wafer` residuals are obtained with

```
> resid( fm1Wafer, level = 1:2 )
      Wafer       Site
  1  0.0615008  0.0680629
  2 -0.1898559 -0.1800129
  .
  .
  399  0.0051645  0.1187074
  400 -0.2076543 -0.0714028
```

The `predict` method is used to obtain predictions for new observations. For example, to obtain the predicted currents at 1.0 V, 1.5 V, 3.0 V and 3.5 V for Wafer 1, we first construct a data frame with the relevant information

```
> newWafer <-
+   data.frame( Wafer = rep(1, 4), voltage = c(1, 1.5, 3, 3.5) )
```

and then use

```
> predict( fm1Wafer, newWafer, level = 0:1 )
      Wafer predict.fixed predict.Wafer
  1     1        2.6126      2.4014
```

2	1	7.0273	6.7207
3	1	23.7826	23.2314
4	1	30.5381	29.9192

Note that, because no predictions were desired at the `Site %in% Wafer` level, `Site` did not need to be specified in `newWafer`. If we are interested in getting predictions for a specific site, say 3, within Wafer 1, we can use

```
> newWafer2 <- data.frame( Wafer = rep(1, 4), Site = rep(3, 4),
+                           voltage = c(1, 1.5, 3, 3.5) )
> predict( fm1Wafer, newWafer2, level = 0:2 )
   Wafer Site predict.fixed predict.Wafer predict.Site
1      1 1/3       2.6126     2.4014    2.4319
2      1 1/3       7.0273     6.7207    6.7666
3      1 1/3      23.7826    23.2314   23.3231
4      1 1/3      30.5381    29.9192   30.0261
```

These methods will be used extensively in the next section, when we describe methods for assessing the adequacy of the fitted models.

4.3 Examining a Fitted Model

Before making inferences about a fitted mixed-effects model, we should check whether the underlying distributional assumptions appear valid for the data. There are two basic distributional assumptions for the mixed-effects models considered in this chapter:

Assumption 1 - the within-group errors are independent and identically normally distributed, with mean zero and variance σ^2 , and they are independent of the random effects.

Assumption 2 - the random effects are normally distributed, with mean zero and covariance matrix Ψ (not depending on the group) and are independent for different groups;

The `nlme` library provides several methods for assessing the validity of these assumptions. The most useful of these methods are based on plots of the residuals, the fitted values, and the estimated random effects. The validity of the distributional assumptions may also be formally assessed using hypothesis tests. However, only rarely do the conclusions of a hypothesis test about some assumption in the model contradict the information displayed in a diagnostic plot.

4.3.1 Assessing Assumptions on the Within-Group Error

Because the assumptions above break down into several smaller assumptions, several diagnostic plots are needed to properly assess their validity. We start by considering Assumption 1 on the within-group error term.

Dependencies among the within-group errors are usually modeled with correlation structures, which will be discussed in detail in §5.3, where methods for assessing the assumption of independence among the within-group errors will also be described. In this section, we concentrate on methods for assessing the assumption that the within-group errors are normally distributed, are centered at zero, and have constant variance.

The primary quantities used to assess the adequacy of Assumption 1 are the *within-group residuals*, defined as the difference between the observed response and the within-group fitted value. Conditional on the random-effects variance-covariance components, the within-group residuals are the BLUPs of the within-group errors. In practice, the within-group residuals are only estimated BLUPs, as the random-effects variance-covariance components need to be replaced with their estimates. Nevertheless, they generally provide good surrogates for the within-group errors and can be used to qualitatively assess the validity of Assumption 1. Other quantities used for assessing Assumption 1 graphically include the within-group fitted values, the observed values, and any covariates of interest.

The `plot` method for the `lme` class is the primary tool for obtaining diagnostic plots for Assumption 1. It takes several optional arguments, but a typical call is

```
plot( object, formula )
```

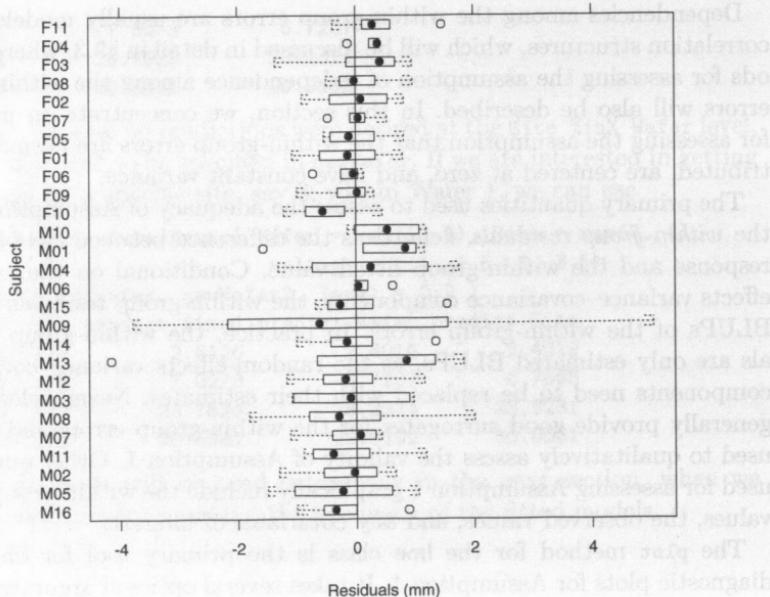
where `object` is an `lme` object and `formula` is a formula object describing the components to be used in the plot. The general expression for `formula` is `y~x|g` where `y` and `x` define, respectively, the vertical and horizontal axes for the plot and `g` is an optional factor (or a set of factors separated by `*` operators) defining the panels of a Trellis display of `y~x`. Any variables, or functions of variables, which can be evaluated in the data frame used to fit `object`, are allowed for `y`, `x`, and `g`, provided the resulting plot is a valid `Trellis` plot. The symbol `"."` is reserved to represent the fitted `object` itself in the `formula` definition. For example, `resid(.)` can be used in `formula` to represent `resid(object)`. We illustrate the capabilities of the `plot` method through the analysis of the examples described in earlier sections of this chapter. Details about the arguments to the `lmList` and `lme` plot methods are given in the help files in Appendix B.

Orthodontic Growth Curve

The first residual plot we consider is the boxplot of residuals by group. For the `fm20rth.lme` object of §4.2.1 we use

```
> plot( fm20rth.lme, Subject~resid( ), abline = 0 ) # Figure 4.16
```

The argument `abline = 0` indicates that a vertical line at zero should be added to the plot. This plot is useful for verifying that the errors are centered at zero (i.e., $E[\epsilon] = 0$), have constant variance across groups

FIGURE 4.16. Boxplots of the residuals for `fm20rth.lme` by subject.

($\text{Var}(\epsilon_{ij}) = \sigma^2$), and are independent of the group levels. Figure 4.16 indicates that the residuals are centered at zero, but that the variability changes with group. Because there are only four observations per subject, we cannot rely too much on the individual boxplots for inference about the within-group variances. We observe an outlying observation for subject M13 and large residuals for subject M09. A pattern suggested by the individual boxplots is that there is more variability among boys (the lower 16 boxplots) than among girls (the upper 11 boxplots). We can get a better feeling for this pattern by examining the plot of the standardized residuals versus fitted values by gender, shown in Figure 4.17

```
> plot( fm20rth.lme, resid(., type = "p") ~ fitted(.) | Sex,
+       id = 0.05, adj = -0.3 ) # Figure 4.17
```

The `type = "p"` argument to the `resid` method specifies that the standardized residuals should be used. The `id` argument specifies a critical value for identifying observations in the plot (standardized residuals greater than the $1-\text{id}/2$ standard normal quantile in absolute value are identified in plot). By default, the group labels are used to identify the observations. The argument `adj` controls the position of the identifying labels. It is clear from Figure 4.17 that the variability in the orthodontic distance measurements is greater among boys than among girls. Within each gender the variability seems to be constant. The outlying observations for subjects M09 and M13 are evident.

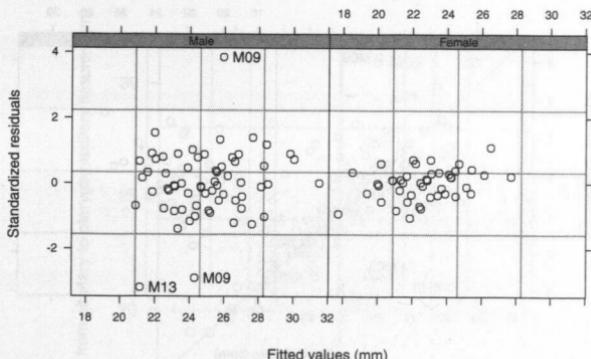


FIGURE 4.17. Scatter plots of standardized residuals versus fitted values for fm20orth.lme by gender.

A more general model to represent the orthodontic growth data allows different variances by gender for the within-group error. The `lme` function allows the modeling of heteroscedasticity of the within-error group via a `weights` argument. This topic will be covered in detail in §5.2, but, for now, it suffices to know that the `varIdent` variance function structure allows different variances for each level of a factor and can be used to fit the heteroscedastic model for the orthodontic growth data as

```
> fm30orth.lme <-  
+   update( fm20orth.lme, weights = varIdent(form = ~ 1 | Sex) )  
> fm30orth.lme  
Linear mixed-effects model fit by REML  
Data: Orthodont  
Log-restricted-likelihood: -207.15  
Fixed: distance ~ Sex + I(age - 11) + Sex:I(age - 11)  
(Intercept)      Sex I(age - 11) Sex:I(age - 11)  
23.808 -1.1605     0.63196    -0.15241
```

Random effects:

```
Formula: ~ I(age - 11) | Subject  
Structure: General positive-definite  
          StdDev   Corr  
(Intercept) 1.85498 (Inter  
I(age - 11) 0.15652 0.394  
Residual    1.62959
```

Variance function:

```
Structure: Different standard deviations per stratum  
Formula: ~ 1 | Sex  
Parameter estimates:  
Male  Female  
1 0.40885
```

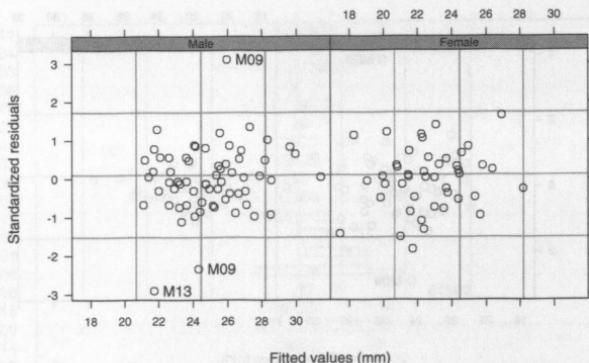


FIGURE 4.18. Scatter plots of standardized residuals versus fitted values for the heteroscedastic fit of fm30orth.lme by gender.

Number of Observations: 108

Number of Groups: 27

The parameters for varIdent give the ratio of the stratum standard errors to the within-group standard error. To allow identifiability of the parameters, the within-group standard error is equal to the first stratum standard error. For the orthodontic data, the standard error for the girls is about 41% of that for the boys. The remaining estimates are very similar to the ones in the homoscedastic fit fm20orth.lme. We can assess the adequacy of the heteroscedastic fit by re-examining plots of the standardized residuals versus the fitted values by gender, shown in Figure 4.18. The standardized residuals in each gender now have about the same variability. We can still identify the outlying observations, corresponding to subjects M09 and M13. Overall, the standardized residuals are small, suggesting that the linear mixed-effects model was successful in explaining the orthodontic growth curves. This is better seen by looking at a plot of the observed responses versus the within-group fitted values.

```
> plot( fm30Orth.lme, distance ~ fitted(.),
+       id = 0.05, adj = -0.3 ) # Figure 4.19
```

The fm30orth.lme fitted values are in close agreement with the observed orthodontic distances, except for the three extreme observations on subjects M09 and M13.

The need for an heteroscedastic model for the orthodontic growth data can be formally tested with the anova method.

```
> anova( fm20orth.lme, fm30orth.lme )
      Model df     AIC     BIC   logLik  Test L.Ratio p-value
fm20orth.lme     1  8 451.35 472.51 -217.68
fm30orth.lme     2  9 432.30 456.09 -207.15 1 vs 2   21.059 <.0001
```

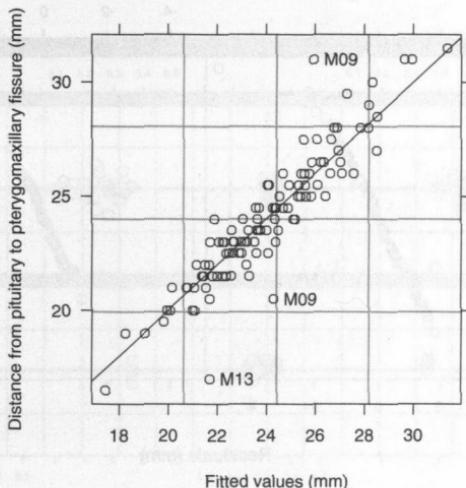


FIGURE 4.19. Observed versus fitted values plot for `fm30rth.lme`.

The very small p -value of the likelihood ratio statistic confirms that the heteroscedastic model explains the data significantly better than the homoscedastic model.

The assumption of normality for the within-group errors can be assessed with the normal probability plot of the residuals, produced by the `qqnorm` method. A typical call to `qqnorm` is of the form

```
qqnorm( object, formula )
```

where `object` is an `lme` object and `formula` is a one-sided formula object of the form `"x|g"`. As in the `plot` method, the symbol `".."` represents the fitted `lme` object. The `x` term in `formula` can be either the residuals (`resid(.)`), or the random effects (`ranef(.)`) associated with the fit. In this section, we consider only the case where `x` defines a vector of residuals. The random-effects case is considered in §4.3.2. The `g` term in `formula` defines an optional factor (or set of factors joined by `*`) determining the panels for a Trellis display.

For example, to obtain the normal plots of the residuals corresponding to `fm30rth.lme` by gender, we use

```
> qqnorm( fm30rth.lme, ~resid(.) | Sex ) # Figure 4.20
```

Once again, we observe the three outlying points, but for the rest of the observations the normality assumption seems plausible.

Radioimmunoassays of IGF-I

We initially consider the plot of the standardized residuals versus fitted values by Lot for the `fm2IGF.lme` object, obtained with

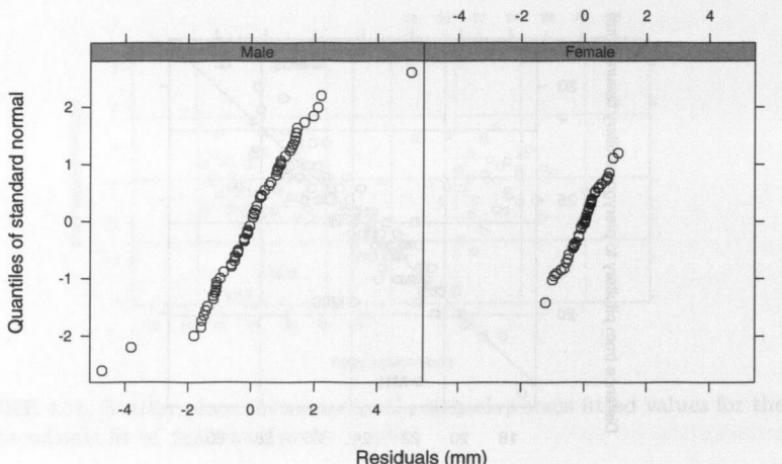


FIGURE 4.20. Normal plot of residuals for the fm30rth.lme lme fit.

```
> plot( fm2IGF.lme, resid(., type = "p") ~ fitted(.) | Lot,
+       layout = c(5,2) ) # Figure 4.21
```

The residuals are centered around zero and seem to have similar variability across lots. There are some outliers in the data, most noticeably for Lots 3 and 7.

We assess the normality of the within-group errors with a `qqnorm` plot of the residuals.

```
> qqnorm( fm2IGF.lme, ~ resid(.),
+           id = 0.05, adj = -0.75 ) # Figure 4.22
```

The normal plot in Figure 4.22 suggests that the distribution of the within-group errors has heavier tails than expected under normality, but is also symmetric around zero. Perhaps a mixture of normal distributions or a *t*-distribution with a moderate number of degrees of freedom would model the distribution of the within-group error more adequately. However, as the heavier tails seem to be distributed symmetrically, the estimates of the fixed effects should not change substantially under either a mixture model or a *t*-model. The heavier tails tend to inflate the estimate of the within-group standard error under the Gaussian model, leading to more conservative tests for the fixed effects, but, because the *p*-value for the hypothesis that the decay of tracer activity with age is zero is quite high (0.673), the main conclusion should remain unchanged under either a mixture or a *t*-model.

Thickness of Oxide Coating on a Semiconductor

The plot of the within-group standardized residuals (`level = 2` in this case) versus the within-group fitted values is the default display produced by the `plot` method. Therefore,

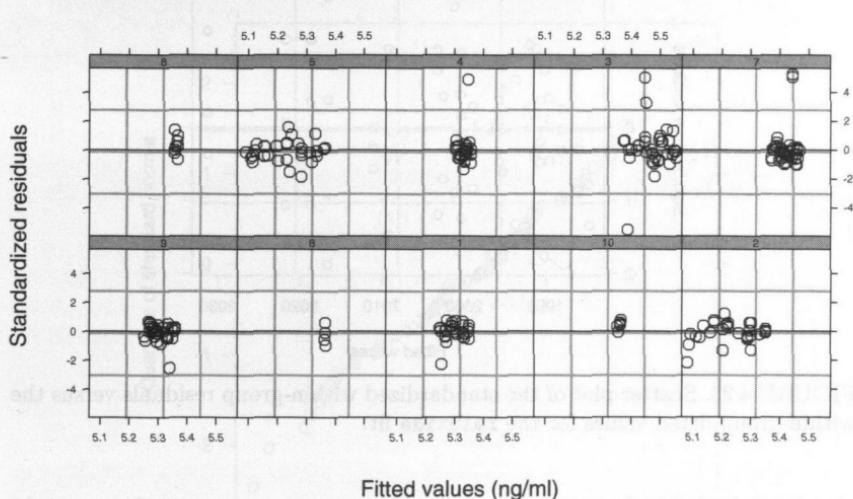


FIGURE 4.21. Scatter plots of standardized residuals versus fitted values for the fm2IGF.lme fit, by lot.

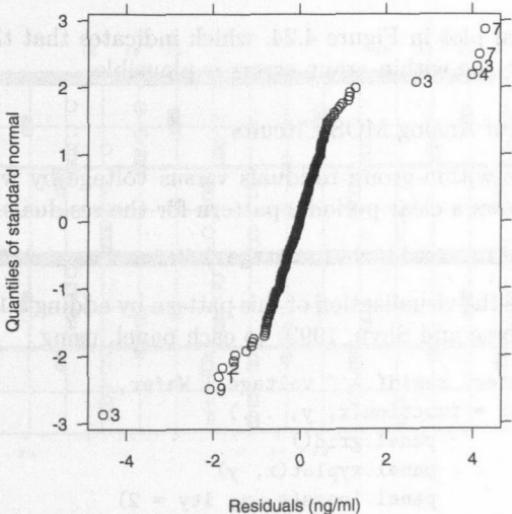


FIGURE 4.22. Normal plot of residuals for the fm2IGF.lme fit.

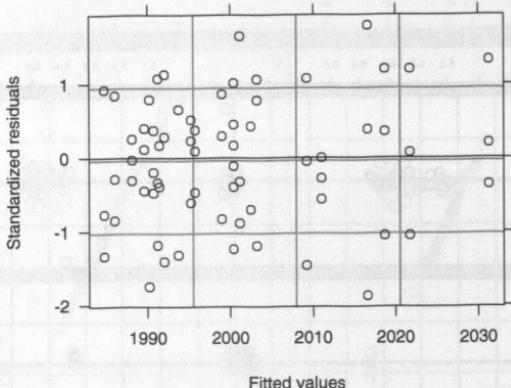


FIGURE 4.23. Scatter plot of the standardized within-group residuals versus the within-group fitted values for the `fm10xide` fit.

```
> plot( fm10xide )
```

Figure 4.23

results in the plot shown in Figure 4.23, which does not indicate any departures from the within-group errors assumptions: the residuals are symmetrically distributed around zero, with approximately constant variance.

By default, the `qqnorm` method produces a normal plot of the within-group standardized residuals. Hence,

```
> qqnorm( fm10xide )
```

Figure 4.24

gives the normal plot in Figure 4.24, which indicates that the assumption of normality for the within-group errors is plausible.

Manufacturing of Analog MOS Circuits

The plot of the within-group residuals versus voltage by wafer, shown in Figure 4.25, shows a clear periodic pattern for the residuals.

```
> plot( fm1Wafer, resid(.) ~ voltage | Wafer )
```

Figure 4.25

We can enhance the visualization of this pattern by adding a `loess` smoother (Cleveland, Grosse and Shyu, 1992) to each panel, using

```
> plot( fm1Wafer, resid(.) ~ voltage | Wafer,
+       panel = function(x, y, ...) {
+         panel.grid()
+         panel.xyplot(x, y)
+         panel.loess(x, y, lty = 2)
+         panel.abline(0, 0)
+       } )
```

Figure 4.26

The `panel` argument to the `plot` method overwrites the default panel function, allowing customized displays.

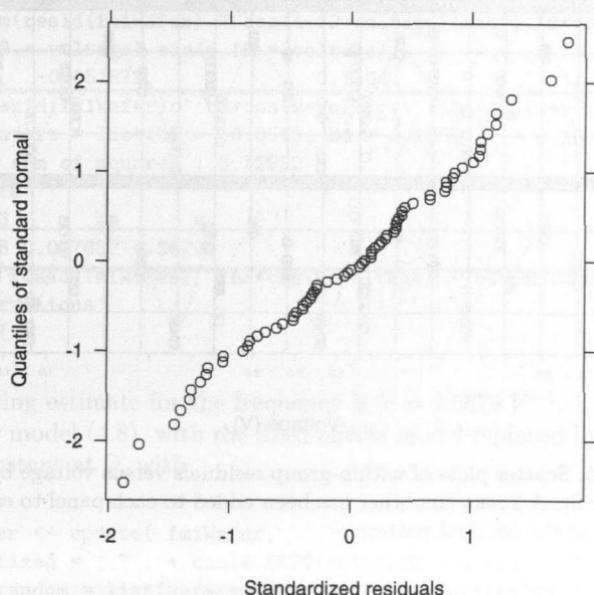


FIGURE 4.24. Normal plot of within-group standardized residuals for the **fm1Oxide** fit.

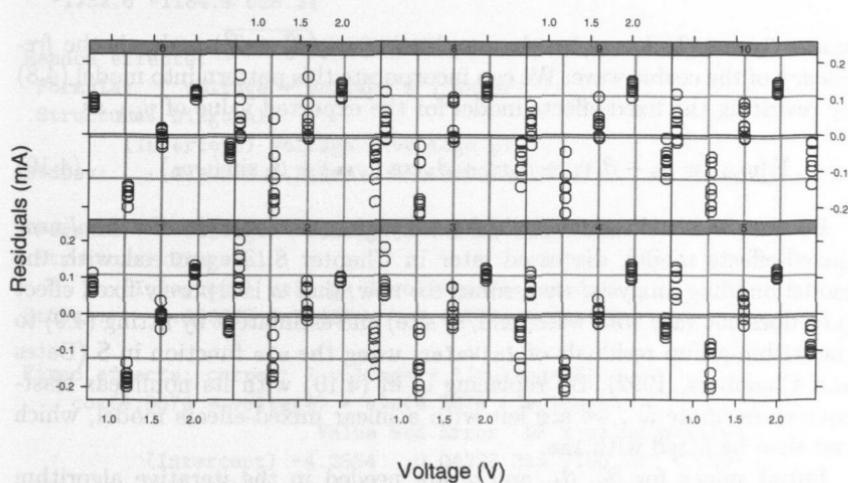


FIGURE 4.25. Scatter plots of within-group residuals versus voltage by wafer for the **fm1Wafer** fit.

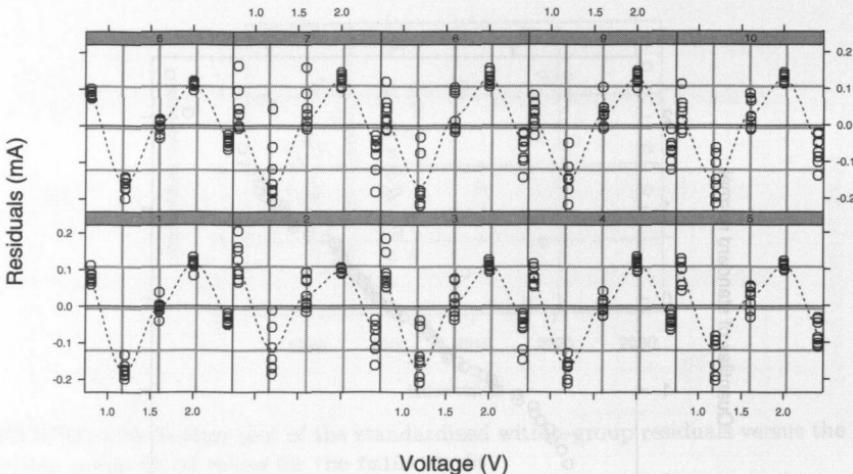


FIGURE 4.26. Scatter plots of within-group residuals versus voltage by wafer for the `fm1Wafer` fit. A loess smoother has been added to each panel to enhance the visualization of the residual pattern.

The same periodic pattern appears in all panels of Figure 4.26, with a period T of approximately 1.5 V. Noting that the residuals are centered around zero, this periodic pattern can be represented by the cosine wave

$$\beta_3 \cos(\omega v) + \beta_4 \sin(\omega v), \quad (4.9)$$

where β_3 and β_4 determine the *amplitude* ($= \sqrt{\beta_3^2 + \beta_4^2}$) and ω is the *frequency* of the cosine wave. We can incorporate this pattern into model (4.8) by rewriting the fixed-effects model for the expected value of y_{ijk} as

$$E[y_{ijk}] = \beta_0 + \beta_1 v_k + \beta_2 v_k^2 + \beta_3 \cos(\omega v_k) + \beta_4 \sin(\omega v_k). \quad (4.10)$$

Because ω is unknown, (4.10) formally gives an example of a *nonlinear* mixed-effects model, discussed later in Chapter 8. To proceed with the model building analysis, we assume, for now, that ω is a purely fixed effect (i.e., does not vary with wafer and/or site) and estimate it by fitting (4.9) to the within-group residuals of `fm1Wafer`, using the `nls` function in S (Bates and Chambers, 1992). By replacing ω in (4.10) with its nonlinear least-squares estimate $\hat{\omega}$, we are left with a linear mixed-effects model, which can then be fitted with `lme`.

Initial values for β_3 , β_4 , and ω are needed in the iterative algorithm used in `nls`. The frequency ω and the period T are related by the formula $\omega = 2\pi/T$, giving the initial estimate $\omega_0 = 2\pi/1.5 = 4.19 V^{-1}$ for ω . For a fixed $\omega = \omega_0$, (4.9) is a linear function of β_3 and β_4 and we can use `lm` to obtain initial estimates for these parameters.

```

> attach(Wafer) # making variables in Wafer available
> coef(lm(resid(fm1Wafer) ~ cos(4.19*voltage)+sin(4.19*voltage)-1))
  cos(4.19 * voltage) sin(4.19 * voltage)
-0.051872          0.1304
> nls(resid(fm1Wafer) ~ b3*cos(w*voltage) + b4*sin(w*voltage),
+      start = list(b3 = -0.0519, b4 = 0.1304, w = 4.19) )
Residual sum of squares : 0.72932
parameters:
  b3      b4      w
-0.11173 0.077682 4.5679
formula: resid(fm1Wafer) ~ b3*cos(w*voltage) + b4*sin(w*voltage)
400 observations
> detach()

```

The resulting estimate for the frequency is $\hat{w} = 4.5679 V^{-1}$.

We refit model (4.8), with the fixed-effects model replaced by (4.10) and ω held constant at $\hat{\omega}$, with

```

value by wafer
> fm2Wafer <- update(fm1Wafer,
+   fixed = . ~ . + cos(4.5679*voltage) + sin(4.5679*voltage),
+   random = list(Wafer=pdDiag(~voltage+voltage^2),
+                 Site=pdDiag(~voltage+voltage^2)) )
> summary(fm2Wafer)
Linear mixed-effects model fit by REML
Data: Wafer
AIC      BIC logLik
-1232.6 -1184.9 628.31

Random effects:
Formula: ~ voltage + voltage^2 | Wafer
Structure: Diagonal
  (Intercept) voltage I(voltage^2)
StdDev:     0.12888 0.34865  0.049074

Formula: ~ voltage + voltage^2 | Site %in% Wafer
Structure: Diagonal
  (Intercept) voltage I(voltage^2) Residual
StdDev:     0.039675 0.23437    0.047541 0.011325

Fixed effects: current ~ voltage + I(voltage^2) +
  cos(4.5679 * voltage) + sin(4.5679 * voltage)
            Value Std.Error DF t-value p-value
  (Intercept) -4.2554    0.04223 316 -100.76 <.0001
  voltage      5.6224    0.11416 316   49.25 <.0001
  I(voltage^2)  1.2585    0.01696 316   74.21 <.0001
  cos(4.5679 * voltage) -0.0956    0.00112 316  -85.05 <.0001
  sin(4.5679 * voltage)  0.1043    0.00150 316   69.42 <.0001
  ...

```

The `.~.` in the `fixed` formula is an abbreviated form for the fixed-effects formula in the original `lme` object, `fm1Wafer`. This convention is also available in other S modeling functions, such as `lm` and `aov`. The `random` argument was included in the call to `update` to prevent the estimated random-effects parameters from `fm1Wafer` to be used as initial estimates (these give bad initial estimates in this case and may lead to convergence problems).

The very high t -values for the sine and cosine terms in the `summary` output indicate a significant increase in the quality of the fit when these terms are included in the fixed-effects model. The estimated standard deviations for the random effects are quite different from the ones in `fm1Wafer` and now suggest that there is significant *wafer-to-wafer* and *site-to-site* variation in all random effects in the model. The estimated within-group standard deviation for `fm2Wafer` is about ten times smaller than that of `fm1Wafer`, giving further evidence of the greater adequacy of (4.10). We assess the variability in the estimates with

```
> intervals(fm2Wafer)
Approximate 95% confidence intervals

Fixed effects:
            lower      est.      upper
(Intercept) -4.338485 -4.255388 -4.172292
              voltage  5.397744  5.622357  5.846969
              I(voltage^2) 1.225147  1.258512  1.291878
cos(4.5679 * voltage) -0.097768 -0.095557 -0.093347
sin(4.5679 * voltage)  0.101388  0.104345  0.107303

Random Effects:
  Level: Wafer
            lower      est.      upper
sd((Intercept)) 0.080154 0.128884 0.207240
sd(voltage)     0.213234 0.348651 0.570065
sd(I(voltage^2)) 0.029014 0.049074 0.083002
  Level: Site
            lower      est.      upper
sd((Intercept)) 0.02195 0.039675 0.071712
sd(voltage)     0.19087 0.234373 0.287790
sd(I(voltage^2)) 0.03829 0.047541 0.059028

Within-group standard error:
            lower      est.      upper
0.0092741 0.011325 0.01383
```

The plots of the within-group residuals versus voltage, by wafer, shown in Figure 4.27, indicate that there is still some periodicity left in some of the wafers, suggesting that random effects may be needed to accommodate the variation in ω . We postpone this discussion until Chapter 8, when nonlinear mixed-effects models are described. Note that the absolute values of the

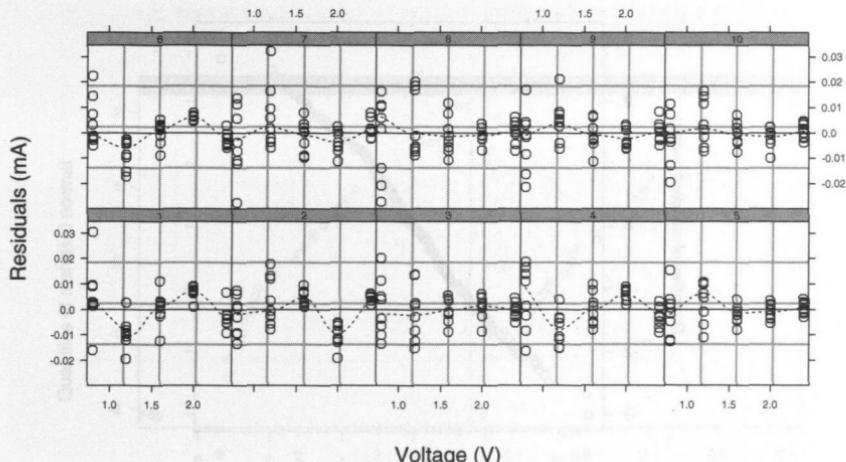


FIGURE 4.27. Scatter plots of within-group residuals versus within-group fitted values by wafer for the `fm2Wafer` fit. A loess smoother has been added to each panel to enhance the visualization of the residual pattern.

within-groups residuals in Figure 4.27 are about an order of magnitude smaller than the ones in Figure 4.26.

The normal plot of the within-group residuals for `fm2Wafer`, obtained with

```
> qqnorm( fm2Wafer ) # Figure 4.28
```

and shown in Figure 4.28, does not indicate any violations from the assumption of normality for the within-group errors.

4.3.2 Assessing Assumptions on the Random Effects

In this section, we describe diagnostic methods for assessing Assumption 2, on the distribution of the random effects. The `ranef` method is used to extract the estimated BLUPs of the random effects from `lme` objects. These are the primary quantities for assessing the distributional assumptions about the random effects.

Two types of diagnostic plots will be used to investigate departures from Assumption 2:

- `qqnorm`—normal plot of estimated random effects for checking marginal normality and identifying outliers;
- `pairs`—scatter plot matrix of the estimated random effects for identifying outliers and checking the assumption of homogeneity of the random effects covariance matrix;

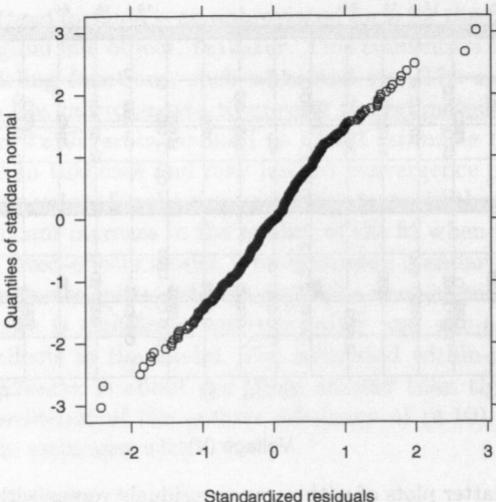


FIGURE 4.28. Normal probability plot of the within-group standardized residuals for the `fm2Wafer` fit.

We illustrate the use of these display methods by continuing the analysis of the examples in §4.3.1.

Orthodontic Growth Curve

We first consider the homoscedastic fitted object `fm20rth.lme` and investigate the marginal normality of the corresponding random effects using the normal plots in Figure 4.29.

```
> qqnorm( fm20rth.lme, ~ranef(.),
+           id = 0.10, cex = 0.7 )
```

Figure 4.29

As in the residual plots of §4.3.1, the `id` argument specifies a critical value for identifying points in the plot (standardized random-effects estimates greater than the $1 - \text{id}/2$ standard normal quantile in absolute value are identified). By default, the group labels are used to identify the points. The assumption of normality seems reasonable for both random effects, though there is some asymmetry in the distribution of the `(Intercept)` random effects. A few outliers appear to be present in both random effects: F10, F11, and M10 for the intercept and M13 for the slope.

To investigate the homogeneity of the random-effects distribution for boys and girls, we use the `pairs` method to obtain scatter plots of the random-effects estimates by gender, as shown in Figure 4.30.

```
> pairs( fm20rth.lme, ~ranef(.) | Sex,
+         id = ~ Subject == "M13", adj = -0.3 )
```

Figure 4.30

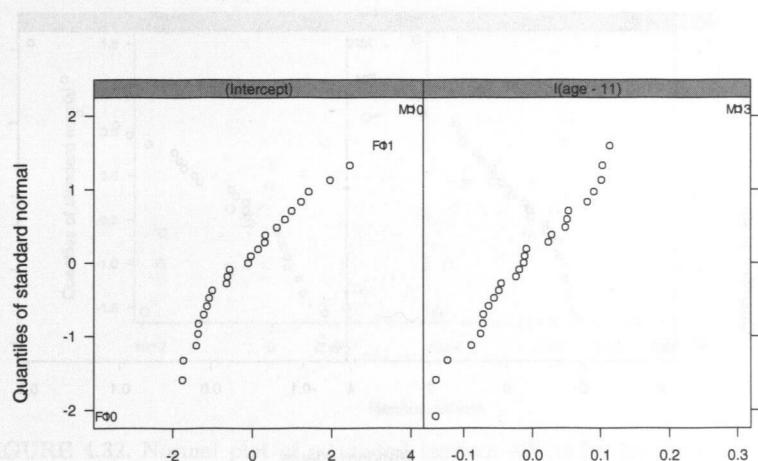


FIGURE 4.29. Normal plot of estimated random effects for the homoscedastic `fm20rth.lme` fit.

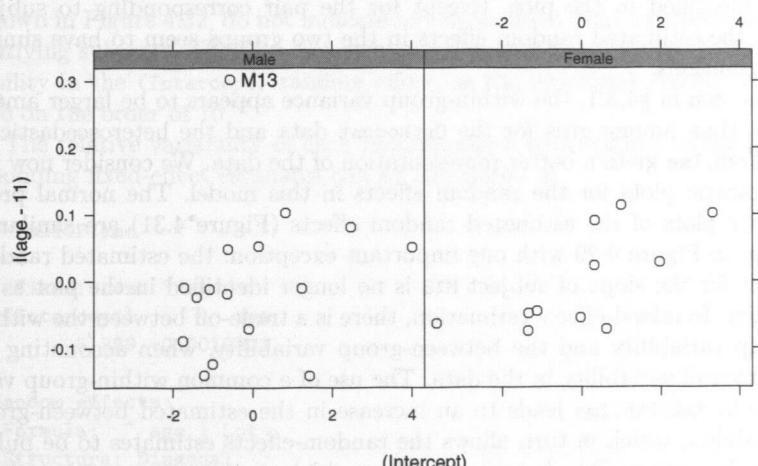


FIGURE 4.30. Scatter plot of estimated random effects for the homoscedastic `fm20rth.lme` fit.

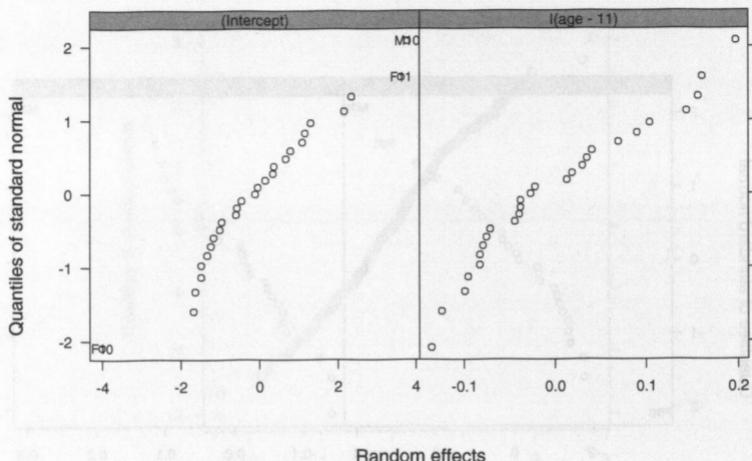


FIGURE 4.31. Normal plot of estimated random effects for the heteroscedastic `fm30rth.lme` lme fit.

The `id` argument is given as a formula object, which is used to identify subject M13 in the plot. Alternatively, it can be given as a numeric value such that points outside the normal distribution contour of level $1-\text{id}/2$ are identified in the plot. Except for the pair corresponding to subject M13, the estimated random effects in the two groups seem to have similar distributions.

As seen in §4.3.1, the within-group variance appears to be larger among boys than among girls for the `Orthodont` data and the heteroscedastic fit `fm30rth.lme` gives a better representation of the data. We consider now the diagnostic plots for the random effects in this model. The normal probability plots of the estimated random effects (Figure 4.31) are similar to those in Figure 4.29 with one important exception: the estimated random effect for the slope of subject M13 is no longer identified in the plot as an outlier. In mixed-effects estimation, there is a trade-off between the within-group variability and the between-group variability, when accounting for the overall variability in the data. The use of a common within-group variance in `fm20rth.lme` leads to an increase in the estimated between-group variability, which in turn allows the random-effects estimates to be pulled away by outliers. The heteroscedastic model in `fm30rth.lme` accommodates the impact of the boys' outlying observations in the within-group variances estimation and reduces the estimated between-group variability, thus increasing the degree of *shrinkage* in the random-effects estimates. In this case, the use of different within-group variances by gender adds a certain degree of robustness to the `lme` fit.

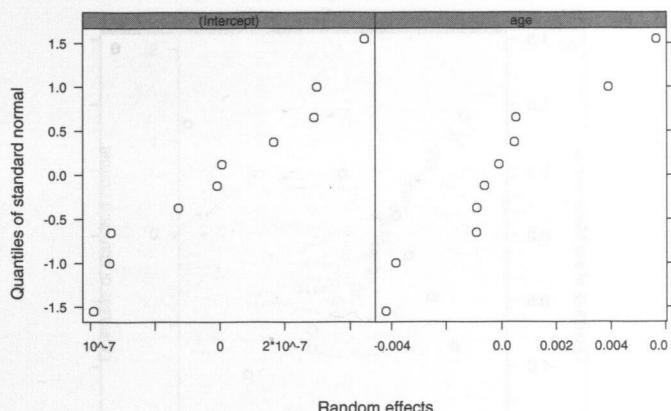


FIGURE 4.32. Normal plot of estimated random effects for the homoscedastic `fm2IGF.lme` fit.

The pairs plots by gender for `Orthodont.lme3`, not included here, do not suggest any departures from the assumption of homogeneity of the random-effects distribution.

Radioimmunoassays of IGF-I

The normal plots of the estimated random effects for the `fm2IGF.lme` fit, shown in Figure 4.32, do not indicate any departures from normality or any outlying subjects. They do, however, suggest that there is very little variability in the (Intercept) random effect, as the estimated random effects are on the order of 10^{-7} .

The relative variability of each random effect with respect to the corresponding fixed-effect estimates can be calculated as

```
> fm2IGF.lme
.
.
.
Fixed: conc ~ age
(Intercept)      age
               5.369 -0.0019301

Random effects:
Formula: ~ age | Lot
Structure: Diagonal
(Intercept)      age Residual
StdDev:  0.00031074 0.0053722  0.8218
.
.
> c( 0.00031074, 0.0053722 )/abs( fixef(fm2IGF.lme) )
(Intercept)      age
5.7876e-05 2.7833
```

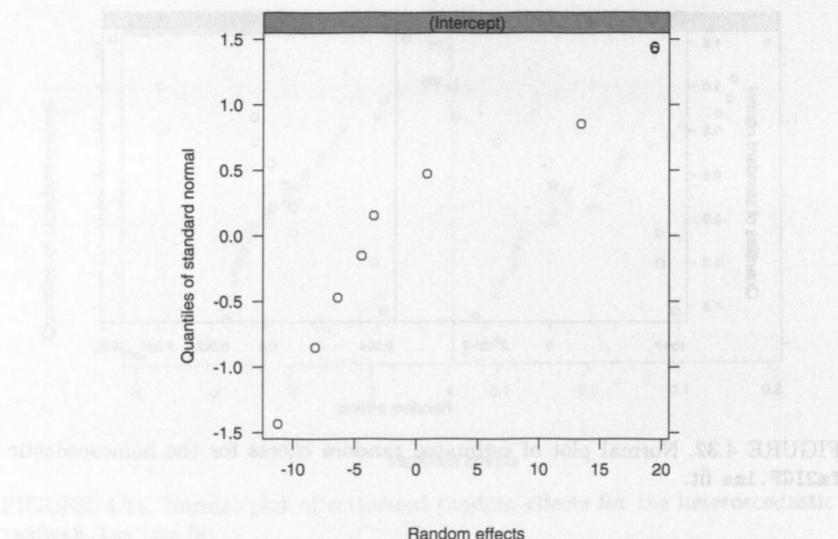


FIGURE 4.33. Normal plot of estimated Lot random effects for the `fm10oxide` fit.

The relative variability of the (Intercept) random effect is only 0.006%, while the relative variability of the age random effect is about 278.3%, suggesting that the former could be dropped from the model.

```
> fm3IGF.lme <- update( fm2IGF.lme, random = ~ age - 1 )
> anova( fm2IGF.lme, fm3IGF.lme )
    Model df   AIC   BIC logLik   Test   L.Ratio p-value
fm2IGF.lme     1   5 604.8 622.10 -297.4
fm3IGF.lme     2   4 602.8 616.64 -297.4 1 vs 2 1.0881e-05  0.9974
```

The high *p*-value for the likelihood ratio test indicates that the random effect for the intercept does not contribute significantly to the fit of the IGF data.

The normal plot of the estimated random effects for the reduced model `fm3IGF.lme`, not included here, does not suggest any violations of the normality assumption and does not show any outlying values.

Thickness of Oxide Coating on a Semiconductor

Normal probability plots of the estimated random effects must be examined at each level of grouping when assessing the adequacy of a multilevel model fit. The normal plot of the estimated Lot random effects for the `fm10oxide` fit, shown in Figure 4.33, is obtained with

```
> qqnorm( fm10oxide, ~ ranef(., level = 1), id=0.10 ) # Figure 4.33
```

The `level` argument to the `ranef` method is required in this case, as, by default, `ranef` returns a list with the estimated random effects at each level

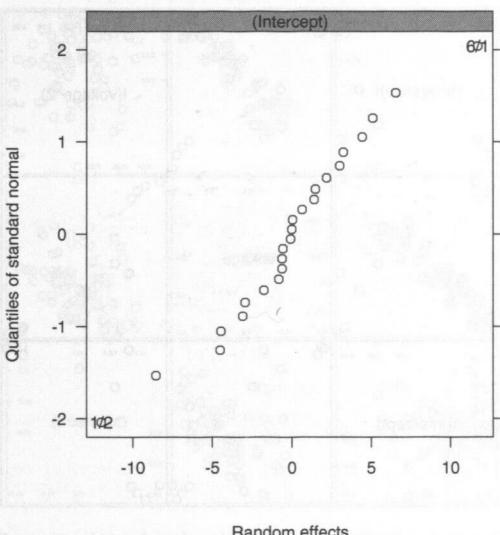


FIGURE 4.34. Normal plot of estimated Wafer %in% Lot random effects for the `fm1Oxide` fit.

of grouping, which will cause an error in `qqnorm`. Because there are only eight random effects at the Lot level, it is difficult to identify any patterns in Figure 4.33. Lot 6 is indicated as a potential outlier, which is consistent with the plot of the data in Figure 4.14, where this lot is shown to have the thickest oxide layers.

The normal plot of the Wafer %in% Lot random effects, shown in Figure 4.34, and obtained with

```
> qqnorm( fm1Oxide, ~ranef(., level = 2), id=0.10 ) # Figure 4.34
```

does not indicate any departures from normality. There is some mild evidence that Wafers 1/2 and 6/1 may be outliers.

Manufacturing of Analog MOS Circuits

The pairs plot of the estimated Wafer random effects for the `fm2Wafer` fit, shown in Figure 4.35, suggests that the random effects at that level are correlated.

The `fm2Wafer` fit uses a diagonal structure for the variance-covariance matrix of the Wafer random effects, which is equivalent to assuming that these random effects are independent. We test the independence assumption by fitting the model with a general positive-definite structure for the variance-covariance matrix of the Wafer random effects and comparing it to the `fm2Wafer` fit using the `anova` method.

```
> fm3Wafer <- update( fm2Wafer,
```

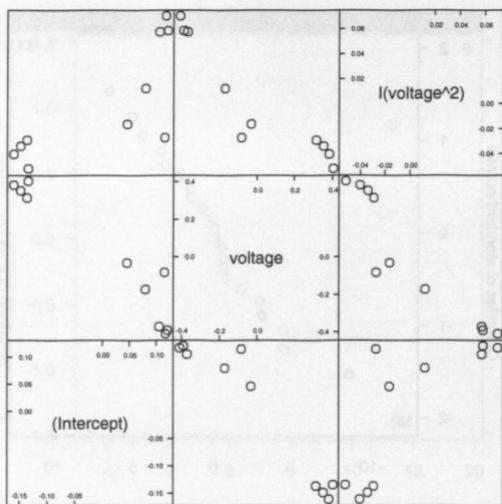


FIGURE 4.35. Scatter plot of estimated Wafer random effects for the fm2Wafer fit.

```

+           random = list(Wafer = ~voltage+voltage^2,
+                               Site = pdDiag(~voltage+voltage^2)))

> fm3Wafer
...
Random effects:
Formula: ~ voltage + voltage^2 | Wafer
Structure: General positive-definite
          StdDev   Corr
(Intercept) 0.131622 (Intr) voltag
          voltage 0.359244 -0.967
I(voltage^2) 0.051323  0.822 -0.940

Formula: ~ voltage + voltage^2 | Site %in% Wafer
Structure: Diagonal
          (Intercept) voltage I(voltage^2) Residual
StdDev:    0.033511 0.21831     0.045125 0.011832
          .
> anova( fm2Wafer, fm3Wafer )
      Model df      AIC      BIC logLik   Test L.Ratio p-value
fm2Wafer     1 12 -1232.6 -1184.9 628.31
fm3Wafer     2 15 -1267.0 -1207.3 648.50 1 vs 2  40.378 <.0001

```

There is a very significant increase in the log-restricted-likelihood, as evidenced by the large value for the likelihood ratio test, indicating that the more general model represented by fm3Wafer gives a better fit.

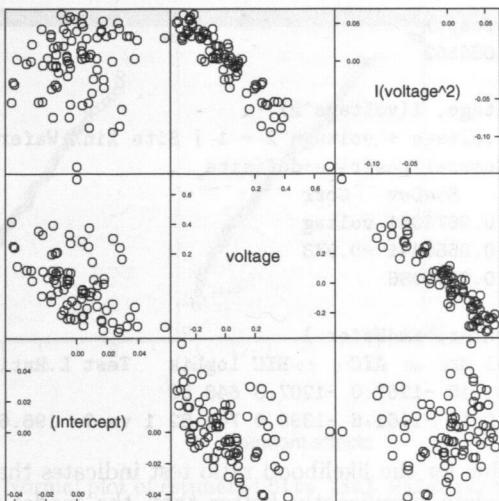


FIGURE 4.36. Scatter plot of estimated random effects at the Site %in% Wafer level for the fm3Wafer fit.

The pairs plot for the estimated Site %in% Wafer random effects corresponding to fm3Wafer in Figure 4.36 indicate that there is a strong negative correlation between the voltage and voltage² random effects, but no substantial correlation between either of these random effects and the (Intercept) random effects. A block-diagonal matrix can be used to represent such covariance structure, with the (Intercept) random effect corresponding to one block and the voltage and voltage² random effects corresponding to another block.

```
> fm4Wafer <- update(fm3Wafer,
+   random = list(Wafer = ~ voltage + voltage^2,
+                 Site = pdBlocked(list(~1, ~voltage+voltage^2 - 1))))
> fm4Wafer
...
Random effects:
Formula: ~ voltage + voltage^2 | Wafer
Structure: General positive-definite
          StdDev   Corr
(Intercept) 0.131807 (Intr) voltag
           voltage 0.354746 -0.967
I(voltage^2) 0.049957  0.814 -0.935

Composite Structure: Blocked

Block 1: (Intercept)
Formula: ~ 1 | Site %in% Wafer
```

```

  (Intercept)
StdDev: 0.066562

Block 2: voltage, I(voltage^2)
Formula: ~ voltage + voltage^2 - 1 | Site %in% Wafer
Structure: General positive-definite
      StdDev   Corr
voltage 0.2674061 voltag
I(voltage^2) 0.0556441 -0.973
Residual 0.0091086
.

> anova( fm3Wafer, fm4Wafer )
    Model df     AIC     BIC logLik  Test L.Ratio p-value
fm3Wafer     1 15 -1267.0 -1207.3 648.50
fm4Wafer     2 16 -1461.6 -1398.0 746.82 1 vs 2 196.65 <.0001

```

The small *p*-value for the likelihood ratio test indicates that the `fmWafer4` model fits the data significantly better than the model represented by `fm3Wafer`. This will be the final model consider in this chapter for the `Wafer` data. Later in §8 we revisit this example, fitting a nonlinear mixed-effects model to the data.

The normal plot of the estimated `Site %in% Wafer` random effects corresponding to `fm4Wafer`, shown in Figure 4.37, does not suggest any significant departure from the assumption of normality for these random effects. There is some moderate evidence that Sites 1/6, 7/3 and 8/1 may be outliers.

```

> qqnorm( fm4Wafer, ~ranef(., level = 2), id = 0.05,
+           cex = 0.7, layout = c(3, 1) ) # Figure 4.37

```

4.4 Chapter Summary

This chapter describes the capabilities available in the `nlme` library for fitting and analyzing linear mixed-effects models with uncorrelated, homoscedastic within-group errors. The `lme` function, for fitting linear mixed-effects models, is described in detail and its various capabilities and associated methods are illustrated through the analyses of several real data examples, covering single-level models, multilevel nested models, and models with crossed random effects.

The model-building approach developed in this chapter follows an “inside-out” strategy, using individual `lm` fits, obtained with the `lmeList` function, to construct more sophisticated linear mixed-effects models. A rich, integrated suite of diagnostic plots to assess model assumptions is described and illustrated through examples.

The class of mixed-effects models which can be fit with `lme` is greatly extended by the availability of patterned random-effects variance-covariance

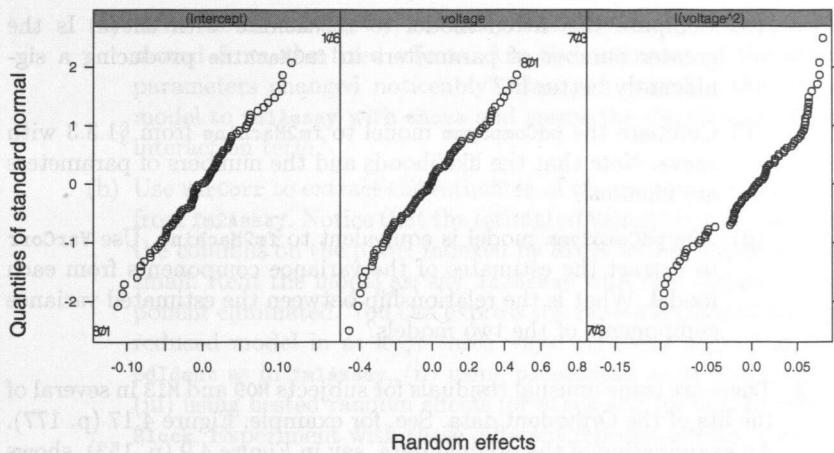


FIGURE 4.37. Normal plot of estimated Site %in% Wafer random effects for the `fm4Wafer` fit.

structures. These are implemented in S through `pdMat` classes, which can be extended with user defined classes.

The linear mixed-effects model considered in this chapter is extended in two different ways later in the book. In Chapter 5, the assumption of uncorrelated, homoscedastic within-group errors is relaxed, and variance functions and correlation structures are introduced to model heteroscedasticity and within-group dependence. The assumption of linearity for $E[\mathbf{y}_i|\mathbf{b}_i]$ is relaxed in Chapter 8, when nonlinear mixed-effects models are described.

Exercises

1. In §1.3.3 (p. 27) we fit the model

```
> fm3Machine <- update(fm1Machine, random = ~Machine-1|Worker)
```

The estimated correlations of the components of the random-effects vectors \mathbf{b}_i in `fm3Machine`, 0.803, 0.623, and 0.771, are similar in magnitude. The estimated standard deviations of these components, 4.08, 8.63, and 4.39 are comparable to each other although not as close as the estimated correlations. Together, these suggest a compound symmetry structure for Ψ as described in §4.2.2.

- (a) Fit a model like that of `fm3Machine`, but with a compound symmetry structure. This is similar to the model `fm40atsB` in §4.2.2. Examine numerical summaries and residual plots for this model.

- (b) Compare this fitted model to `fm3Machine` with `anova`. Is the greater number of parameters in `fm3Machine` producing a significantly better fit?
 - (c) Compare the `pdCompSymm` model to `fm2Machine` from §1.3.3 with `anova`. Note that the likelihoods and the numbers of parameters are identical.
 - (d) The `pdCompSymm` model is equivalent to `fm2Machine`. Use `VarCorr` to extract the estimates of the variance components from each model. What is the relationship between the estimated variance components of the two models?
2. There are some unusual residuals for subjects `M09` and `M13` in several of the fits of the `Orthodont` data. See, for example, Figure 4.17 (p. 177). An examination of the original data, say in Figure 4.9 (p. 153), shows some suspicious observations on these subjects. The observation at age 8 for subject `M13` is unusually low. The four observations for subject `M09` decrease, then increase dramatically, then decrease again. The two observations at ages 10 and 12 appear to be incorrect.
- (a) Repeat the stages of fitting a linear mixed-effects models to the `Orthodont` data with the suspicious observations for subjects `M13` and `M09` removed. Do you arrive at different conclusions regarding the models?
 - (b) There are other subjects like `M09` in the `Orthodont` data for whom the observed `distance` decreases with increasing age. Because it is unlikely that this measurement on the same child would get smaller over time, these are probably misrecorded data or unusually large measurement errors.
 - i. Find the number of pairs of measurements that represent decreases within a subject. You can begin with `gapply(Orthodont, "distance", diff)` which returns a list of successive differences of the `distance` by `Subject`.
 - ii. What should be done with these aberrant data points?
 - iii. If you handle the suspect data by deleting some observations, repeat as much of this chapter's analysis of `Orthodont` as is feasible with the modified data. (Note that if you decide to delete some of the suspect data, the resulting data will be unbalanced. Is it still possible to fit mixed-effects models to such unbalanced data?)
3. When analyzing the `Assay` data (Figure 4.13, p. 164), we noted that the interaction term in the fixed effects for the fitted model `fm1Assay` is not significant.

- (a) Refit the model as, say, `fm2Assay` with the interaction term removed from the fixed effects. Are the estimates of the other parameters changed noticeably? Can you compare this fitted model to `fm1Assay` with `anova` and assess the significance of the interaction term?
- (b) Use `VarCorr` to extract the estimates of the variance components from `fm2Assay`. Notice that the estimated variance component for the columns on the plate, indexed by `dilut` within `Block`, is very small. Refit the model as, say, `fm3Assay` with this variance component eliminated. You can express the `random` argument for this reduced model in at least three ways: (i) using `pdBlocked` and `pdIdent` as in `fm1Assay`, (ii) using `pdCompSymm` as in `fm40atsB`, or (iii) using nested random effects for `Block` and for `sample` within `Block`. Experiment with these different representations. Demonstrate that that the fitted models for these three representations are indeed equivalent. For which of the three representations does `lme` converge most easily? Compare one of these fitted models to `fm2Assay` using `anova`. Is the variance component for the columns significant?
- (c) Extract the estimates of the variance components from the fitted model `fm3Assay` and examine the confidence intervals on the standard deviations of the random effects. Do the variance components appear to be significantly greater than zero? Fit a model, say `fm4Assay`, with a single variance component for the `Block`. Compare it to `fm3Assay` with `anova`.
- (d) Can the random effects be eliminated entirely? Fit a model, say `fm5Assay`, using just the fixed-effects terms from `fm4Assay`. Because this model does not have any random-effects terms, you will need to use `lm` or `gls` to fit it.
- (e) Compare models `fm2Assay`, `fm3Assay`, `fm4Assay`, and `fm5Assay` with `anova`. Which model provides the simplest adequate representation for these data?
- (f) Notice that the `dilut` factor represents serial dilutions that will be equally spaced on the logarithm scale. Does converting `dilut` to an ordered factor, so the contrasts used for this factor will be polynomial contrasts, suggest further simplifications of the model?