lme and **nlme**

Mixed-Effects Methods and Classes for S and S-PLUS

Version 3.0
October 1998

by José C. Pinheiro and Douglas M. Bates

Bell Labs, Lucent Technologies and University of Wisconsin — Madison

`nlme` 3.0: Linear and Nonlinear Mixed-Effects Methods and Classes for S and S-PLUS

Mixed-effects models are frequently used to analyze grouped data, because they flexibly model the within-group correlation often present in this type of data. Examples of grouped data include longitudinal data, repeated measures data, multilevel data, and split-plot designs.

We describe a set of S functions, classes, and methods for the analysis of linear and nonlinear mixed-effects models. These extend the modeling facilities available in release 3 of S and releases 3.4 and 4 of S-PLUS. The source code, written in S and C is available at

http://nlme.stat.wisc.edu/.

Help files for all functions and methods described here are included in the PostScript file HelpFunc.ps, which is included with the `nlme` distribution.

José C. Pinheiro
Bell Laboratories
Lucent Technologies
jcp@research.bell-labs.com

Douglas M. Bates
Department of Statistics
University of Wisconsin–Madison
bates@stat.wisc.edu

Contents

1 Introduction

Mixed-effects models are frequently used to analyze grouped data, because they model flexibly the within-group correlation often present in this type of data. Examples of grouped data include longitudinal data, repeated measures data, multilevel data, and split-plot designs. We consider only mixed-effects models for a continuous response, assumed to have a Gaussian distribution. We describe a set of S functions, classes, and methods for the analysis of linear and nonlinear mixed-effects models. These extend the modeling facilities available in release 3 of S (Chambers and Hastie, 1992) and releases 3.4 (Unix) and 4.5 (Windows) of S-PLUS. The source code, written in S and C, is available at http://nlme.stat.wisc.edu/.

The purpose of this document is to describe some of the capabilities in Version 3.0 of the `nlme` software and to give examples of their usage. A detailed description of the various functions, classes, and methods can be found in the corresponding help files, which are available on-line. The PostScript file *HelpFunc.ps*, included with the `nlme` distribution, contains printed versions of the help files.

§2 presents a new class for representing grouped data and some of the methods for this class. Functions and methods for fitting and analyzing linear mixed-effects models are described in §3. The nonlinear mixed-effects functions and methods are described in §4. §5 presents some future directions for the code development.

2 A `groupedData` class

The datasets used for fitting mixed-effects models have several characteristics in common. They consist of measurements of a continuous response at several levels of a covariate, usually `time`, `dose`, or `treatment`. Further, these measurements are grouped according to one, or several, factors. Additional covariates may be present. Some of these vary within a group (*inner* covariates) and some do not (*outer* covariates).

As a first example of grouped data, we consider the data from an orthodontic study presented in Potthoff and Roy (1964). The data, displayed in Figure 1, consist of four measurements of the distance (in millimeters) from the center of the pituitary to the pterygomaxillary fissure made at ages 8, 10, 12, and 14 years on 16 boys and 11 girls. This is an example of balanced repeated measures data, with a single level of grouping (`Subject`).

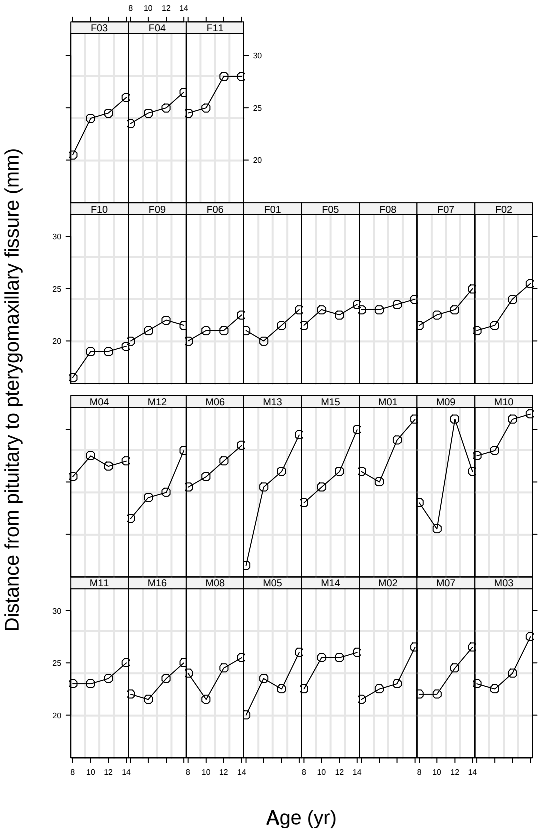

Figure 1: Orthodontic growth patterns in 16 boys(M) and 11 girls(F) between 8 and 14 years of age. Panels within each gender group are ordered by maximum response.

An example of grouped data for which the primary covariate is a categorical variable is given by the data on an experiment to compare three brands of machines used in an industrial process. These data, presented in Milliken and Johnson (1992, §23.1, p. 285), are shown in Figure 2. Six workers were randomly chosen from the employees of a factory to operate each of three machines three times. The response is an overall productivity score taking into account the number and quality of components produced.

As an example of grouped data with a nonlinear response, we consider an experiment on the cold tolerance of a C_4 grass species, *Echinochloa crus-galli*, described in

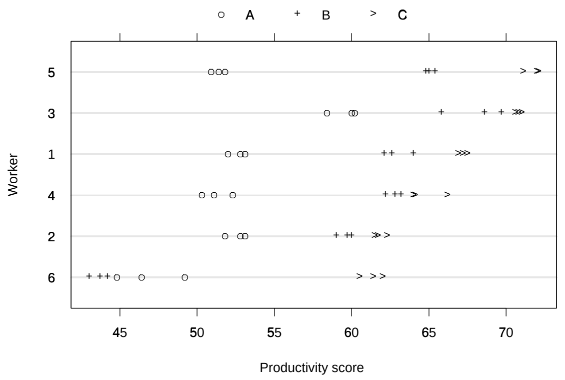

Figure 2: Productivity scores for three brands of machines. Scores take into account number and quality of components produced.

Potvin, Lechowicz and Tardif (1990). The CO_2 uptake of six plants from Québec and six plants from Mississippi was measured at several levels of ambient CO_2 concentration. Half the plants of each type were chilled overnight before the experiment was conducted. The data are shown in Figure 3.

An example of grouped data with two levels of grouping is given by a study in radiology consisting of repeated measures of mean pixel values from CT scans of the right and the left lymphnodes in the axillary region of 10 dogs over a period of 14 days after application of a contrast. The purpose of the experiment was to model the mean pixel value as a function of time, so as to estimate the time where the maximum mean pixel value was attained. The data are shown in Figure 4.

The choice of a data structure for this type of data will affect the ease and flexibility with which we can display the data and fit models to the data. A natural way to represent such data in S-PLUS is as a `data.frame` (i.e a rectangular array). For displaying and modelling grouped data, it is often useful to incorporate a formula specifying some of the roles of the variables in the `data.frame`.

At a minimum the data frame must contain the response, the primary covariate, such as `time`, and the grouping factor(s), such as `Subject`, `Plant`, or `Dog` and `Side`. Additional factors or continuous covariates can be present. For example

```
> names(Orthodont)                  # Orthodontic growth
[1] "distance" "age"      "Subject"  "Sex"
> names(Machines)
```

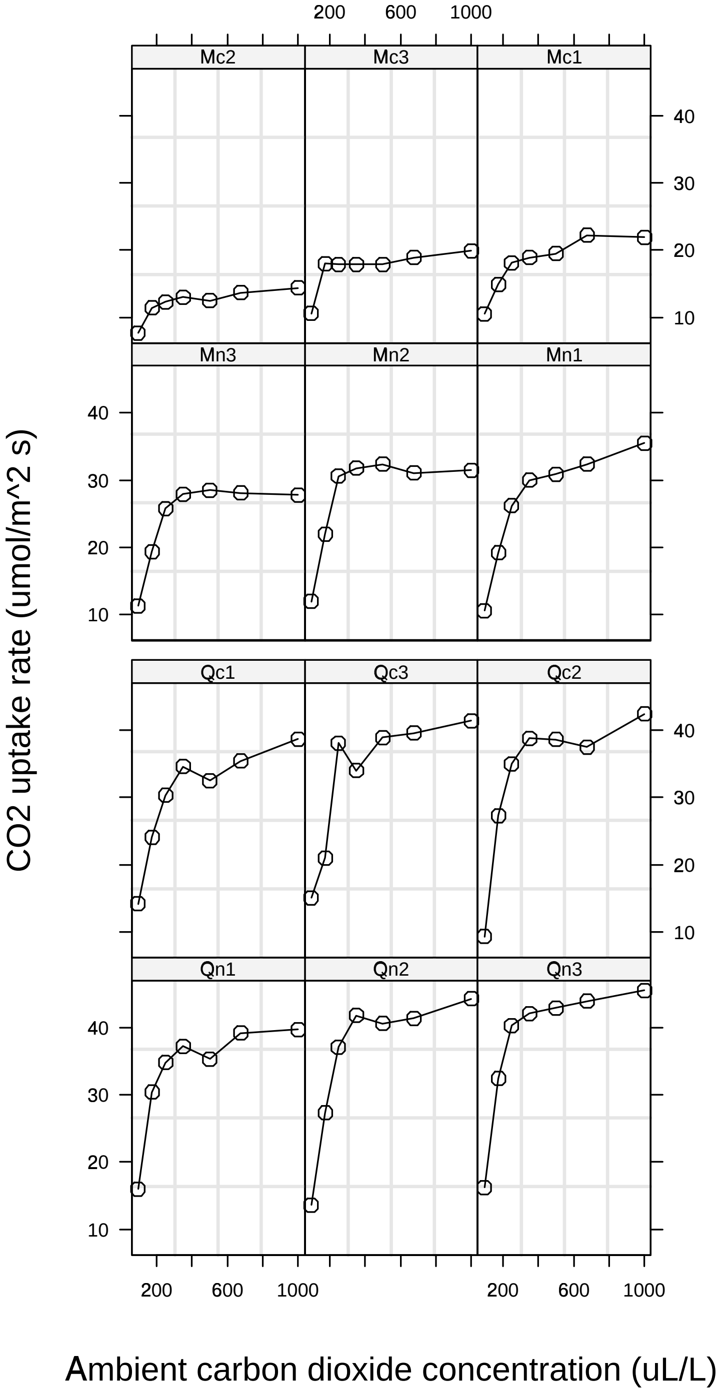

Figure 3: CO_2 uptake versus ambient CO_2 by treatment and type for *Echinochloa crus-galli* plants, six from Québec and six from Mississippi. Half the plants of each type were chilled overnight before the measurements were taken.

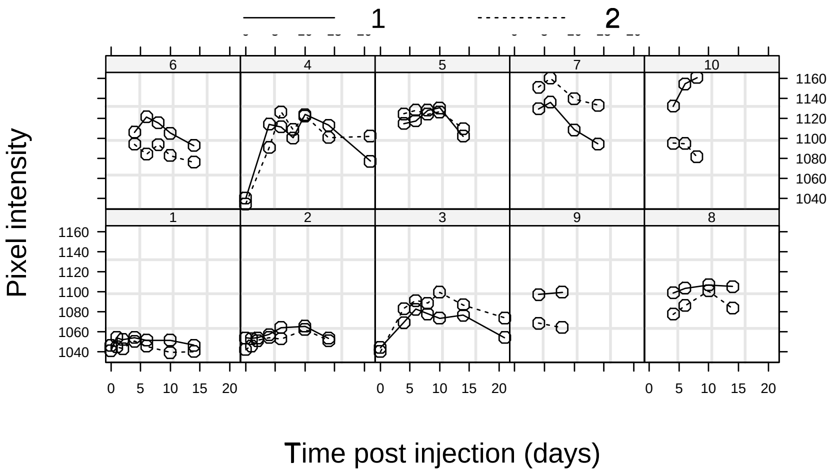

Figure 4: Mean pixel intensity of the right (*1*) and left (*2*) lymphnodes in the axillary region versus time from intravenous application of a contrast. The pixel intensities were obtained from CT scans.

```
[1] "Worker"  "Machine" "score"
> names(CO2)                          # CO2 uptake
[1] "Plant"     "Type"      "Treatment" "conc"      "uptake"
> names(Pixel)                        # Pixel intensity
[1] "Dog"   "Side"  "day"   "pixel"
```

The different roles of the variables in the data frame (response, primary covariate, and grouping factors) can be described by a formula of the form `response ~ primary | grouping1/grouping2/...` which is similar to the display formula in a Trellis plot (Becker, Cleveland and Shyu, 1996). For example

```
> formula(Orthodont)
distance ~ age | Subject
> formula(Machines)
score ~ 1 | Worker
> formula(CO2)
uptake ~ conc | Plant
> formula(Pixel)
pixel ~ day | Dog/Side
```

The most convenient way of packaging the formula with the data is to create a new class of object (Chambers and Hastie, 1992, Appendix A) which we have called `groupedData`.

The function used to create objects of a given class is called the constructor for that class. The constructor for `groupedData` takes a formula and data frame as described above. By default, the grouping factors are converted to ordered factors with the order determined by a summary function applied to the response split according to the groups, taking into account the nesting order (i.e. levels of a factor are sorted within the levels of the factors that are *outer* to it). (The default summary function is the maximum.) Additionally, labels can be given for the response and the primary covariate and their units can be specified as arbitrary strings. The reason for separating the labels and the units is to allow propagation of the units to derived quantities such as the residuals from a fitted model.

For example, creating `groupedData` objects for the examples above from `data.frame` is accomplished by

```
> Orthodont <- groupedData(distance ~ age | Subject,
+   data = Orthodont, outer = ~ Sex,
+   labels = list(x = "Age",
+    y="Distance from pituitary to pterygomaxillary fissure"),
+   units = list(x = "(yr)", y = "(mm)"))
> plot(Orthodont, layout = c(8,4),         # produces Figure 1
+   between = list(y = c(0, 0.5, 0)))
> Machines <- groupedData(score ~ Machine | Worker,
+   data = Machines,
+   labels = list(y = "Productivity score"))
> plot(Machines)                           # produces Figure 2
> CO2 <- groupedData(uptake ~ conc | Plant, data = CO2,
+   outer = ~ Treatment * Type,
+   labels = list(x = "Ambient carbon dioxide concentration",
+     y = "CO2 uptake rate"),
+   units = list(x = "(uL/L)", y = "(umol/m^2 s)"))
> plot(CO2)                                # produces Figure 3
> Pixel <- groupedData(pixel ~ day | Dog/Side,
+   data = Pixel,
+   labels =list(x="Time post injection",y="Pixel intensity"),
+   units = list(x = "(days)"))
> plot(Pixel, display = 1, inner = ~Side)  # produces Figure 4
```

The call to the constructor establishes the roles of the variables, converts the grouping factors to ordered factors so panels in plots are ordered in a natural way and stores descriptive labels for data plots and plots of derived quantities.

When outer factors are present, as in the Orthodont and CO_2 data, they are given as a formula such as `outer = ~ Sex` and `outer = ~ Treatment * Type` or, when, multiple grouping factors are present, as a list of such formulas. Inner factors are described in a similar way. When establishing the order of the levels of the grouping factor, and hence the order of panels in a plot, re-ordering is only permitted within combinations of levels for the outer factors. That is why the panels from boys and girls are grouped together in Figure 1.

The plot method for the `groupedData` class allows an optional argument `outer` which can be given a logical value or a formula. A logical value of `TRUE` (or `T`) indicates that the outer formula stored with the data should be used in the plot. The right hand side of the explicit or inferred formula replaces the grouping factor in the trellis formula. The grouping factor is then used to determine which points to join with lines. For example

```
> plot(Orthodont, outer = T)                # produces Figure 5
> plot(CO2, outer = T)                      # produces Figure 6
```

An inner factor is used to determine which points within a panel are joined by lines, such in the plot of the `Pixel` data above.

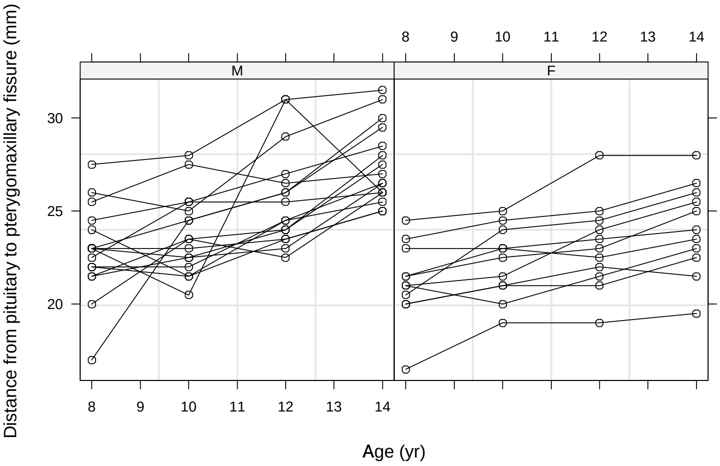

Figure 5: Orthodontic growth patterns in 16 boys(M) and 11 girls(F) between 8 and 14 years of age, with different panels per gender.

When multiple levels of grouping are present, the `plot` method allows two optional arguments `displayLevel` and `collapseLevel`, specifying, respectively, the grouping level to be used to determine the panels of the Trellis plot and the grouping level over which to collapse the data.

Another advantage of using a formula to describe the roles of the variables is that this information can be used within the model-fitting functions to make the specifica-

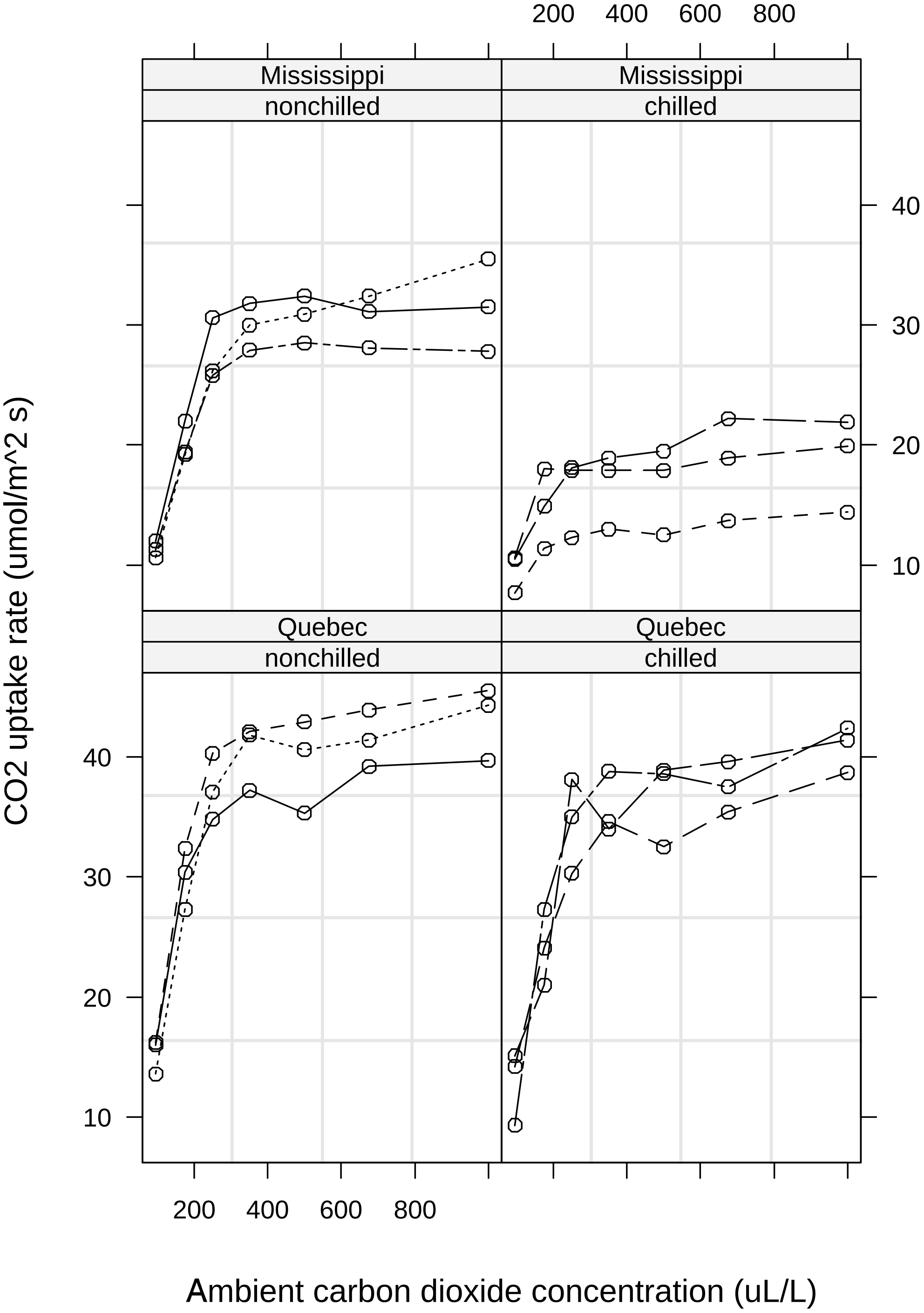

Figure 6: CO_2 uptake versus ambient CO_2 by treatment and type for *Echinochloa crus-galli* plants, six from Québec and six from Mississippi. Half the plants of each type were chilled overnight before the measurements were taken.

tion of the model easier. For example, getting preliminary simple linear regression fits by subject for the orthodontic growth example can be made as simple as

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

3 The `lme` class and related methods

The plot of the individual growth curves in Figure 1 suggests that a linear model adequately explains the orthodontic distance as a function of age, but the intercept and the slope seem to vary with the individual. The corresponding linear mixed-effects model is

$$d_{ij} = (\beta_0 + b_{i0}) + (\beta_1 + b_{i1})\,\mathrm{age}_j + \varepsilon_{ij} \tag{1}$$

where d_{ij} represents the distance for the ith individual at age j, β_0 and β_1 are the population average intercept and the population average slope, b_{i0} and b_{i1} are the effects in intercept and slope associated with the ith individual, and ε_{ij} is the within-subject error term. It is assumed that the $\boldsymbol{b}_i = (b_{i0}, b_{i1})^T$ are independent and identically distributed with a $\mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{D})$ distribution and the ε_{ij} are independent and identically distributed with a $\mathcal{N}(0, \sigma^2)$ distribution, independent of the $\boldsymbol{b}_i$.

One of the questions of interest for these data is whether the curves show significant differences between boys and girls. Model (1) can be modified as

$$d_{ij} = (\beta_{00} + \beta_{01}\mathrm{sex}_i + b_{i0}) + (\beta_{10} + \beta_{11}\mathrm{sex}_i + b_{i1})\,\mathrm{age}_j + \varepsilon_{ij} \tag{2}$$

to test for sex related differences in intercept and slope. In model (2), sex_i is an indicator variable assuming the value zero if the ith individual is a boy and one if she is a girl. β_{00} and β_{10} represent the population average intercept and slope for the boys and β_{01} and β_{11} are the changes in population average intercept and slope for girls. Differences between boys and girls can be evaluated by testing whether β_{01} and β_{11} are significantly different from zero. The remaining terms in (2) are defined as in (1).

In the `Pixel` example, a second order polynomial seems adequate to explain the evolution of pixel intensity with time since the contrast was injected. Preliminary analyses indicated that the intercept varies with dog, as well as with side within dog, and the linear term varies with dog, but not with side.

The corresponding multilevel linear mixed-effects model is

$$\mathrm{y}_{ijk} = (\beta_0 + b_{0i} + b_{0i,j}) + (\beta_1 + b_{1i})\, t_{ijk} + \beta_2 t_{ijk}^2 + \varepsilon_{ijk}, \tag{3}$$

where i refers to the dog number (1 through 10), j to the lymphnode side (1 – right, 2 – left), and k refers to time; β_0, β_1, and β_2 denote respectively the intercept, the linear term, and the quadratic term fixed effects; b_{0i} denotes the intercept random effect at the dog level, b_{0ij} denotes the intercept random effect at the side within dog level, and b_{1i} denotes the linear term random effect at the dog level; y denotes the pixel intensity, t denotes the time since contrast injection, and ε_{ijk} denotes the error term. It is assumed that the $\boldsymbol{b}_i = [b_{0i}, b_{1i}]^T$ are independent and identically distributed with common distribution $\mathcal{N}\big(\mathbf{0}, \sigma^2\boldsymbol{D}_1\big)$, the $\boldsymbol{b}_{i,j} = [b_{0i,j}]$ are independent and identically distributed with common distribution $\mathcal{N}\big(\mathbf{0}, \sigma^2\boldsymbol{D}_2\big)$ and independent of the $\boldsymbol{b}_i$, and the ε_{ijk} are independent and identically distributed with common distribution $\mathcal{N}(0, \sigma^2)$ and independent of the $\boldsymbol{b}_i$ and the $\boldsymbol{b}_{i,j}$.

3.1 The `lme` function

The `lme` function is used to fit a linear mixed-effects model, as described in Laird and Ware (1982), or a multilevel linear mixed-effects model as described, for example, in Longford (1993) or Goldstein (1995), using either maximum likelihood or restricted maximum likelihood. It produces an object of class `lme`. Several optional arguments can be used with this function, but the typical call is

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

Only the first argument is required. The arguments `fixed` and `random` are generally given as formulas as illustrated below. Any linear model formula (Chambers and Hastie, 1992, chapter 3) is allowed, giving the model formulation considerable flexibility. For the Orthodont data these formulas would be written as

```
fixed = distance ~ age, random = ~ age
```

for model (1) and

```
fixed = distance ~ age * Sex, random = ~ age
```

for model (2). Note that the response variable is given only in the formula for the `fixed` argument. By default, all terms in the fixed effects formula are assumed to have random effects.

Because `Orthodont` is a `groupedData` object, no grouping structure must be explicitly given in `random`, as it is extracted from the `groupedData` display formula. Alternatively, the grouping structure can be included in the formula as conditioning expression.

```
random = ~ age | Subject
```

When multiple levels of grouping are present, as in the pixel intensity example, `random` must be given as a list of formulas, as below.

```
fixed = pixel ~ day+day^2, random =list(Dog =~ day, Side =~ 1)
```

Note that the names of the elements in the `random` list correspond to the names of the grouping factors and are assumed to be in outermost to innermost order. A model with a single intercept is represented by `~ 1`.

The optional argument `data` specifies the data frame in which the variables used in the model are available. A simple call to `lme` to fit model (1) is

```
> Ortho.fit1 <- lme(fixed = distance ~ age, data = Orthodont,
+    random = ~ age | Subject)
```

To fit model (2) we use

```
> ## set contrasts for desired parameterization
> options(contrasts = c("contr.treatment", "contr.poly"))
> Ortho.fit2 <- update(Ortho.fit1, fixed = distance ~ age*Sex)
```

The multilevel model (3) is fit by:

```
> Pixel.fit1 <- lme(fixed = pixel ~ day + day^2, data = Pixel,
+    random = list(Dog = ~ day, Side = ~1))
```

There are several methods available for the fitted objects of class `lme`, including those for the generic functions `anova`, `print`, `summary`, and `plot`. These are illustrated in the next sections.

3.2 The `print`, `summary`, and `anova` methods

A brief description of the estimation results is returned by the `print` method. It gives estimates of the standard errors and correlations of the random effects, the within-group variance, and the fixed effects. For the `Ortho.fit1` object we get

```
> Ortho.fit1
Linear mixed-effects model fit by REML
  Data: Orthodont
  Log-restricted-likelihood: -221.32
  Fixed: distance ~ age
 (Intercept)     age
      16.761 0.66019

Random effects:
 Formula:  ~ age | Subject
 Structure: General positive-definite
             StdDev   Corr
(Intercept) 2.32704 (Inter
        age 0.22643 -0.609
   Residual 1.31004

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

A more complete description of the estimation results is returned by `summary`.

```
> summary(Ortho.fit2)
Linear mixed-effects model fit by REML
 Data: Orthodont
     AIC    BIC  logLik
  448.58 469.74 -216.29

Random effects:
 Formula:  ~ age | Subject
 Structure: General positive-definite
             StdDev   Corr
(Intercept) 2.40549 (Inter
        age 0.18034 -0.668
   Residual 1.31004

Fixed effects: distance ~ age + Sex + age:Sex
              Value Std.Error DF t-value p-value
(Intercept)  16.341     1.019 79  16.043   0.000
        age   0.784     0.086 79   9.121   0.000
        Sex   1.032     1.596 25   0.647   0.524
    age:Sex  -0.305     0.135 79  -2.262   0.026
 Correlation:
        (Intr)    age    Sex
    age -0.880
    Sex -0.638  0.562
age:Sex  0.562 -0.638 -0.880
```

```
Standardized Within-Group Residuals:
     Min       Q1      Med      Q3    Max
 -3.1681 -0.38594 0.007103 0.44516 3.8495

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

The approximate standard errors for the fixed effects are derived using the asymptotic theory described in Pinheiro (1994). The results above indicate that the measurement increases faster in boys than in girls (significant, negative `age:Sex` fixed effect), but the average intercept is common to boys and girls (non-significant `Sex` fixed effect).

Alternatively, a likelihood ratio test can be used to test the difference between the fixed effects models represented by `CO2.fit1` and `CO2.fit2`. The `anova` method provides that capability. Because the default estimation method in `lme` is restricted maximum likelihood (REML) and likelihood comparisons between REML fits with different fixed effects structures are not meaningful, we need to refit the two objects using maximum likelihood, before calling `anova`.

```
> Ortho.fit1.ML <- update(Ortho.fit1, method = "ML")
> Ortho.fit2.ML <- update(Ortho.fit2, method = "ML")
> anova(Ortho.fit1.ML, Ortho.fit2.ML)
              Model df    AIC    BIC  logLik    Test Lik.Ratio
Ortho.fit1.ML     1  6 451.21 467.30 -219.61
Ortho.fit2.ML     2  8 443.81 465.26 -213.90 1 vs. 2    11.406
                p-value
Ortho.fit1.ML
Ortho.fit2.ML 0.0033365
```

The likelihood ratio test strongly rejects the null hypothesis of no sex differences. For small sample sizes, likelihood ratio tests tend to be *too liberal* when comparing models with nested fixed effects structures and should be used with caution. We recommend using the Wald-type tests provided by the `anova` method with a single argument, as these tend to have significance levels close to nominal, even for small samples.

The same methods can be used with `lme` objects resulting from multilevel fits. To summarize the estimation results for model (3) we use

```
> summary(Pixel.fit1)
Linear mixed-effects model fit by REML
 Data: Pixel
     AIC    BIC  logLik
  841.21 861.97 -412.61
```

```
Random effects:
 Formula:  ˜ day | Dog
 Structure: General positive-definite
             StdDev   Corr
(Intercept) 28.3699 (Inter
        day  1.8437 -0.555

 Formula:  ˜ 1 | Side %in% Dog
        (Intercept) Residual
StdDev:      16.824   8.9896

Fixed effects: pixel ˜ day + dayˆ2
              Value Std.Error DF t-value p-value
(Intercept)  1073.3      10.2 80   105.5       0
        day     6.1       0.9 80     7.0       0
   I(dayˆ2)    -0.4       0.0 80   -10.8       0
 Correlation:
         (Intr)    day
     day -0.517
I(dayˆ2)  0.186 -0.668

Standardized Within-Group Residuals:
     Min       Q1      Med       Q3   Max
 -2.8291 -0.44918 0.025549 0.55722 2.752

Number of Observations: 102
Number of Groups:
 Dog Side %in% Dog
  10            20
```

3.3 The `plot` method

Diagnostic plots for assessing the quality of the fitted model are obtained using the `plot` method for class `lme`. This method takes several optional arguments, but a typical call is of the form.

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

where the first argument is the `lme` object and the second is a display formula for the Trellis plot to be produced. The fitted object can be referenced by the symbol "." in the formula argument. For example, to produce a plot of the standardized residuals versus fitted values by gender for the `Ortho.fit2` object included in Figure 7, we use.

```
> plot(Ortho.fit2,                         # produces Figure 7
+   resid(., type = "p") ˜ fitted(.) | Sex)
```

There is evidence that the variability of the orthodontic distance is greater in boys than in girls and that some possible outliers are present in the data. To assess the predictive

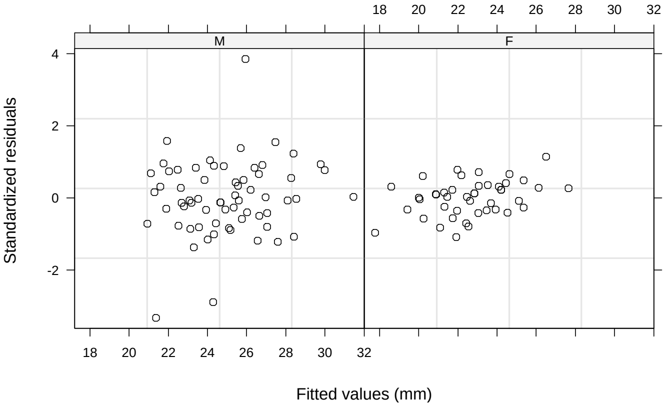

Figure 7: Standardized residuals versus fitted values by gender, for the `lme` fit of model (2).

power of the fitted model, we consider the plot of the observed versus fitted values by individual, presented in Figure 8 and obtained with.

```
> plot(Ortho.fit2,                            # produces Figure 8
+   distance ˜ fitted(.) | Subject, layout = c(8, 4),
+   between = list(y = c(0, 0.5, 0)), abline = c(0,1))
```

For most of the subjects, there is very good agreement between the observed and fitted values, indicating that the fit is adequate.

The `formula` argument to the `plot` method gives virtually unlimited flexibility for generating customized diagnostic plots. As one last example, we consider the plot of the standardized residuals (at the side within dog level) for the `Pixel.fit1` object by dog.

```
> plot(Pixel.fit1, Dog˜resid(., type="p"))# produces Figure 9
```

The residuals seem symmetrically scattered around zero, with similar variabilities, except, possibly, for dog number 4.

3.4 Other methods

Standard S methods for extracting components of fitted objects, such as `residuals`, `fitted`, and `coefficients`, can be also be used on `lme` objects. In addition, the `lme` includes the methods `fixed.effects` and `random.effects` for extracting the

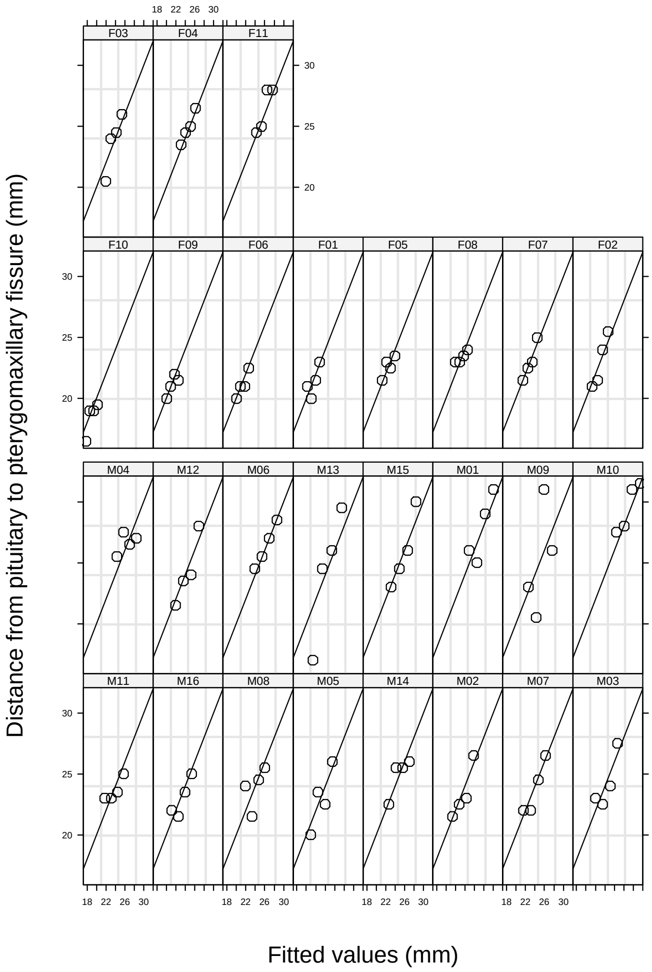

Figure 8: Observed distances versus fitted values by subject, for the `lme` fit of model (2).

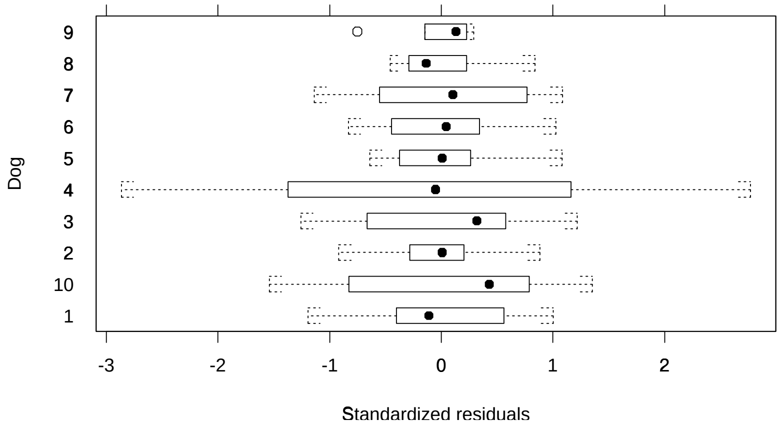

Figure 9: Standardized residuals by dog, for the `lme` fit of model (3).

fixed effects and the random effects estimates, respectively. Short names for the last two functions are `fixef` and `ranef`.

```
> coef(Ortho.fit2)
    (Intercept)     age    Sex  age:Sex
M16      15.557 0.69573 1.0321 -0.30483
M05      14.695 0.77590 1.0321 -0.30483
. . .
F04      18.002 0.81259 1.0321 -0.30483
F11      18.537 0.88586 1.0321 -0.30483
> fixef(Pixel.fit1)
 (Intercept)    day I(dayˆ2)
      1073.3 6.1296 -0.36735
> ranef(Pixel.fit1, level = 1) # random effects at Dog level
   (Intercept)       day
 1    -24.7142 -1.195371
10     19.3659 -0.099369
 2    -23.5821 -0.432431
 3    -27.0803  2.194756
 4    -16.6585  3.095973
 5     25.2998 -0.561271
 6     10.8232 -1.037000
 7     49.3539 -2.274458
 8     -7.0540  0.990255
 9     -5.7537 -0.681084
```

Predicted values are returned by the `predict` method. For example, if we are interested in predicting the average measurement for both boys and girls at ages 14, 15, and 16, as well as for subjects `M01` and `F10` at age 13, based on model (2), we should create a new data frame, say `Orthodont.new`, as follows,

```
> Orthodont.new  <-
+  data.frame(Sex = c(1, 1, 1, 0, 0, 0, 1, 0),
+   age = c(14, 15, 16, 14, 15, 16, 13, 13),
+   Subject = c(NA, NA, NA, NA, NA, NA, "M01", "F10"))
```

and then use

```
> predict(Ortho.fit2, Orthodont.new, level = c(0,1))
  Subject predict.fixed predict.Subject
1      NA        24.086              NA
2      NA        24.566              NA
3      NA        25.045              NA
4      NA        27.322              NA
5      NA        28.106              NA
6      NA        28.891              NA
7     M01        23.607          26.242
8     F10        26.537          22.738
```

to get the subject-specific and population predictions. The `level` argument is used to define the desired prediction levels, with `0` (zero) referring to the population predictions.

3.5 Positive-definite matrix structures

Different positive-definite matrix structures can be used to represent the random effects covariance matrix in `lme`. These are organized in the code as different `pdMat` classes. Table 1 lists the standard `pdMat` classes available in `lme`.

Class	Description
pdSymm	general positive-definite
pdDiag	diagonal
pdIdent	multiple of an identity
pdCompSymm	compound symmetry
pdBlocked	block diagonal

Table 1: Classes of positive-definite matrices in `lme`.

By default, the `pdSymm` class is used to represent a random effects covariance matrix. The desired `pdMat` class must be specified with the `random` argument. For example, to fit a model with independent intercept and slope random effects in model (2), one should use

```
> Ortho.fit3 <- update(Ortho.fit2, random = pdDiag(~ age))
> Ortho.fit3
Linear mixed-effects model fit by REML
  Data: Orthodont
  Log-restricted-likelihood: -216.58
  Fixed: distance ~ age + Sex + age:Sex
 (Intercept)     age    Sex  age:Sex
      16.341 0.78437 1.0321 -0.30483

Random effects:
 Formula:  ~ age | Subject
 Structure: Diagonal
        (Intercept)      age Residual
StdDev:      1.5546 0.088016   1.3655

Number of Observations: 108
Number of Groups: 27
> anova(Ortho.fit2, Ortho.fit3)
           Model df    AIC    BIC  logLik    Test Lik.Ratio
Ortho.fit2     1  8 448.58 469.74 -216.29
Ortho.fit3     2  7 447.15 465.66 -216.58 1 vs. 2   0.56928
           p-value
Ortho.fit2
Ortho.fit3 0.45054
```

Note that, because the two models have the same fixed effects structure, the likelihood ratio test based on REML is meaningful.

The grouping structured is inferred from the `groupedData` display formula. Alternatively, `random` could have been passed to the function as

```
 random = list(Subject = pdDiag(~ age))
```

As evidenced by the large p-value for the likelihood ratio test in the `anova` method output, the independence between the random effects seems plausible.

Users may define their own `pdMat` classes by specifying a *constructor* function and, at a minimum, methods for the functions `pdConstruct`, `pdMatrix` and `coef`. For examples of these functions, see the methods for classes `pdSymm` and `pdDiag`.

3.6 Correlation and variance function structures

The within-group error covariance structure can be flexibly modeled in `lme` by combining correlation structures and variance functions. Similarly to the positive-definite matrix structures described in §3.5, the different correlation and variance functions structures are organized into `corStruct` and `varFunc` classes, respectively. Tables 2 and 3 list the standard classes for each structure.

Class	Description
corAR1	AR(1)
corARMA	ARMA(p,q)
corCAR1	continuous AR(1)
corCompSymm	compound symmetry
corExp	exponential spatial correlation
corGaus	Gaussian spatial correlation
corLin	linear spatial correlation
corRation	Rational quadratic spatial correlation
corSpher	spherical spatial correlation
corSymm	general correlation matrix

Table 2: Classes of correlation structures in `lme`.

The optional argument `correlation` is used to specify a correlation structure and the optional argument `weights` is used for variance functions. By default, the within-group errors are assumed to independent and homoscedastic.

Class	Description
varExp	exponential of a variance covariate
varPower	power of a variance covariate
varConstPower	constant plus power of a variance covariate
varIdent	different variances per level of a factor
varFixed	fixed weights, determined by a variance covariate
varComb	combination of variance functions

Table 3: Classes of variance functions in `lme`.

The variance function structures are used to model heteroscedasticity in the within-group errors. For example, the residual versus fitted values plot of the residuals on Figure 7 suggests that different variances should be allowed for boys and girls. We can test that by updating the fit using the `varIdent` variance function structure.

```
> Ortho.fit4 <-
+   update(Ortho.fit3, weights = varIdent(form = ~1|Sex))
> Ortho.fit4
Linear mixed-effects model fit by REML
  Data: Orthodont
  Log-restricted-likelihood: -206.08
  Fixed: distance ~ age + Sex + age:Sex
 (Intercept)     age    Sex  age:Sex
      16.341 0.78438 1.0321 -0.30483

Random effects:
 Formula:  ~ age | Subject
 Structure: Diagonal
        (Intercept)    age Residual
StdDev:      1.4487 0.1094   1.6584

Variance function:
 Structure: Different standard deviations per stratum
 Formula:  ~ 1 | Sex
 Parameter estimates:
 Male  Female
    1 0.42537
Number of Observations: 108
Number of Groups: 27
> anova(Ortho.fit3, Ortho.fit4)
           Model df    AIC    BIC  logLik    Test Lik.Ratio
Ortho.fit3     1  7 447.15 465.66 -216.58
Ortho.fit4     2  8 428.17 449.32 -206.08 1 vs. 2    20.983
              p-value
Ortho.fit3
Ortho.fit4 4.6342e-06
```

There is strong indication that the orthodontic distance is less variable in girls than in boys. The fitted object can be referenced in the `form` argument to the `varFunc` constructors through the symbol “.”. For example, to use a variance function that is an arbitrary power of the fitted values in model (3), one can re-fit `Pixel.fit1` as below.

```
> Pixel.fit2 <-
+   update(Pixel.fit1, weights = varPower(form=~fitted(.)))
> Pixel.fit2
Linear mixed-effects model fit by REML
  Data: Pixel
  Log-restricted-likelihood: -412.46
  Fixed: pixel ~ day + day^2
 (Intercept)    day I(day^2)
      1073.3 6.1011 -0.36638

Random effects:
 Formula:  ~ day | Dog
 Structure: General positive-definite
             StdDev   Corr
(Intercept) 28.5049 (Inter
        day  1.8734 -0.567

 Formula:  ~ 1 | Side %in% Dog
        (Intercept)  Residual
StdDev:       16.66 4.217e-06

Variance function:
 Structure: Power of variance covariate
 Formula:  ~ fitted(.)
 Parameter estimates:
  power
 2.0845
Number of Observations: 102
Number of Groups:
 Dog Side %in% Dog
  10            20
> anova(Pixel.fit1, Pixel.fit2)
           Model df    AIC    BIC  logLik    Test Lik.Ratio
Pixel.fit1     1  8 841.21 861.97 -412.61
Pixel.fit2     2  9 842.92 866.28 -412.46 1 vs. 2   0.29119
           p-value
Pixel.fit1
Pixel.fit2 0.58946
```

There is no evidence of heteroscedasticity in this case, as evidenced by the large p-value of the likelihood ratio test in the `anova` output. Because the default value for `form` in `varPower` is `~fitted(.)`, it suffices to use `weights = varPower()` in this example.

The correlation structures are used to model within-group correlations, not captured by the random effects. These are generally associated with temporal or spatial dependencies. For example, we can test for the presence of an autocorrelation of lag 1 in the orthodontic growth example by updating `Ortho.fit4` as below.

```
> Ortho.fit5 <- update(Ortho.fit4, corr = corAR1())
> Ortho.fit5
Linear mixed-effects model fit by REML
  Data: Orthodont
  Log-restricted-likelihood: -206.04
  Fixed: distance ~ age + Sex + age:Sex
 (Intercept)     age    Sex age:Sex
      16.317 0.78599 1.0608 -0.3069

Random effects:
 Formula:  ~ age | Subject
 Structure: Diagonal
        (Intercept)     age Residual
StdDev:       1.451 0.11211   1.6307

Correlation Structure: AR(1)
 Parameter estimate(s):
       Phi
 -0.057025
Variance function:
 Structure: Different standard deviations per stratum
 Formula:  ~ 1 | Sex
 Parameter estimates:
 Male  Female
    1 0.42506
Number of Observations: 108
Number of Groups: 27
> anova(Ortho.fit4, Ortho.fit5)
           Model df    AIC    BIC  logLik    Test Lik.Ratio
Ortho.fit4     1  8 428.17 449.32 -206.08
Ortho.fit5     2  9 430.07 453.87 -206.04 1 vs. 2  0.094035
           p-value
Ortho.fit4
Ortho.fit5 0.75911
```

The large p-value of the likelihood ratio test indicates that the autocorrelation is not present. Note that the correlation structure is used together with the variance function, representing an heterogeneous AR(1) process (Littel, Milliken, Stroup and Wolfinger, 1996). Because the two structures are defined and constructed separately, any correlation structure can be combined with any variance function.

Users may define their own correlation and variance function classes by specifying appropriate *constructor* functions and a few method functions. For a new correlation structure, method functions must be defined for at least `corMatrix` and `coef`. For examples of these functions, see the methods for classes `corSymm` and `corAR1`. A new variance function structure requires methods for at least `coef`, `coef<-`, and `initial-`

`ize`. For examples of these functions, see the methods for class `varPower`.

4 The `nlme` class and related methods

We illustrate the use of the functions and methods for the nonlinear mixed-effects model by analyzing the `CO2` data of §2. These data come from a study of the cold tolerance of a C_4 grass species, *Echinochloa crus-galli*. A total of twelve four-week-old plants, six from Québec and six from Mississippi, were divided into two groups: control plants that were kept at 26°C and chilled plants that were subject to 14 h of chilling at 7°C. After 10 h of recovery at 20°C, CO_2 uptake rates (in $\mu mol/m^2 s$) were measured for each plant at seven concentrations of ambient CO_2 (100, 175, 250, 350, 500, 675, 1000μL/L). Each plant was subjected to the seven concentrations of CO_2 in increasing, consecutive order. The objective of the experiment was to evaluate the effect of plant type and chilling treatment on the CO_2 uptake.

The model used in Potvin et al. (1990) is

$$U_{ij} = \phi_{1i} \left\{ 1 - \exp\left[-\phi_{2i} \left(C_j - \phi_{3i} \right) \right] \right\} + \varepsilon_{ij}, \tag{4}$$

where U_{ij} denotes the CO_2 uptake rate of the ith plant at the jth CO_2 ambient concentration; ϕ_{1i}, ϕ_{2i}, and ϕ_{3i} denote respectively the asymptotic uptake rate, the uptake growth rate, and the maximum ambient CO_2 concentration at which no uptake is verified for the ith plant; C_j denotes the jth ambient CO_2 level; and the ε_{ij} are independent and identically distributed error terms with distribution $\mathcal{N}(0, \sigma^2)$.

4.1 The `nlme` function

The `nlme` function is used to fit nonlinear mixed-effects models, as defined in Lindstrom and Bates (1990), using either maximum likelihood or restricted maximum likelihood. Several optional arguments can be used with this function, but a typical call is

```
nlme(model, data, fixed, random, start)
```

The `model` argument is required and consists of a formula specifying the nonlinear model to be fitted. Any S nonlinear formula can be used, giving the function considerable flexibility. From (4) we have that for the CO_2 uptake data this argument is declared as

```
uptake ~ A * (1 - exp(-B * (conc - C)))
```

where we have used the notation $A = \phi_1$, $B = \phi_2$, and $C = \phi_3$. To enforce the rate parameter ϕ_2 to be positive, while preserving an unrestricted parametrization, we can re-parametrize the model above using $lB = log(B)$

```
uptake ~ A * (1 - exp(-exp(lB) * (conc - C)))
```

Alternatively, we can define an S function, say `CO2.func`, as

```
> CO2.func <-
+   function(conc, A, lB, C) A*(1 - exp(-exp(lB)*(conc - C)))
```

then write the `model` argument as

```
uptake ~ CO2.func(conc, A, lB, C)
```

The advantage of this latter approach is that the analytic derivatives of the model function can be passed to the `nlme` function as the `gradient` attribute of the returned value from `CO2.func` and used in the optimization algorithm. The S function `deriv` can be used to create expressions for the derivatives.

```
> CO2.func <- deriv(~ A * ( 1 - exp(-exp(lB) * ( conc - C))),
+   c("A", "lB", "C"), function(conc, A, lB, C))
```

If the value returned by the model function does not have a `gradient` attribute, numerical derivatives are used in the optimization.

The arguments `fixed` and `random` are formulas, or lists of formulas, that define the structures of the fixed and random effects in the model. The first argument is required. In these formulas a `1` on the right hand side of a formula indicates that a single parameter is associated with the effect, but any linear formula in S could be used instead. This gives considerable flexibility to the model, as time-dependent parameters can be easily incorporated (e.g. when a formula in `fixed` involves a covariate that changes with time). Usually every parameter in the model will have an associated fixed effect, but it may, or may not, have an associated random effect. Since we assumed that all random effects have mean zero, the inclusion of a random effect without a corresponding fixed effect would be unusual. Note that the `fixed` and `random` formulas could be directly incorporated in the model declaration. The approach used in `nlme` allows for more efficient calculation of derivatives.

For the CO_2 uptake data, if we want to fit a model in which all parameters are random and no covariates are included we use

```
fixed = A + lB + C ~ 1, random = A + lB + C ~ 1
```

By default, `random = fixed`, so the `random` argument can be omitted. Because `CO2` is a `groupedData` object, no grouping structure must be explicitly given in `random`, as it is extracted from the `groupedData` display formula. Alternatively, the grouping structure can be included in the formula as conditioning expression.

```
random = A + lB + C ~ 1 | Plant
```

If we want to estimate the (fixed) effects of plant type and chilling treatment on the parameters in the model we can use

```
fixed = A + lB + C ~ Type * Treatment, random = A + lB + C ~ 1
```

`Data` is an optional argument that names a data frame in which the variables in `model`, `fixed`, and `random` are found, and `start` provides a list of starting values for the iterative algorithm. Only the fixed effects starting estimates are required. The default starting estimates for the random effects are zero.

A simple call to `nlme` to fit model (4), without any covariates and with all parameters as mixed effects is

```
> CO2.fit1 <-
+  nlme(model = uptake ~ CO2.func(conc, A, lB, C),
+   fixed = A + lB + C ~ 1, data = CO2,
+   start = c(30, log(0.01), 50))
```

The initial values for the fixed effects were obtained from Potvin et al. (1990).

4.2 Methods for `nlme` objects

Objects returned by the `nlme` function are of class `nlme` which inherits from `lme`. All methods described in section 3 are also available for the `nlme` class. In fact, with the exception of the `predict` method, all methods are common to both classes. We illustrate their use here with the $\mathrm{CO_2}$ uptake data.

The `print` method provides a brief description of the estimation results. It gives estimates of the standard errors and correlations of the random effects, of the within-group variance, and of the fixed effects.

```
> CO2.fit1
Nonlinear mixed-effects model fit by maximum likelihood
  Model: uptake ~ co2.func(conc, A, lB, C)
  Data: CO2
  Log-likelihood: -201.29
  Fixed: A + lB + C ~ 1
      A      lB      C
 32.468 -4.6323 43.827

Random effects:
 Formula: list(A ~ 1, lB ~ 1, C ~ 1)
 Level: Plant
 Structure: General positive-definite
          StdDev   Corr
       A  9.5052 A      lB
      lB  0.1465 -0.129
       C 11.9562  0.883  0.125
Residual  1.7427

Number of Observations: 84
Number of Groups: 12
```

Note that there is a moderately strong correlation between the A and the C random effects and that these have small correlations with the lB random effect. The scatter plot matrix of the random effects, obtained using the `pairs` method

```
> pairs(CO2.fit1, ~ranef(.))
```

and shown in Figure 10, gives a graphical description of the random effects correlation structure.

The correlation between A and C may be due to the fact that the plant type and the chilling treatment, which were not included in the `CO2.fit1` model, are affecting A and C in the similar ways.

The `plot` method for the `ranef.lme` class can be used to explore the dependence of the individual parameters A, lB, and C in model (4) on plant type and chilling factor.

```
> plot(ranef(CO2.fit1, augFrame = T), outer = ~Treatment*Type,
+   layout = c(3,1))            # produces Figure 11
```

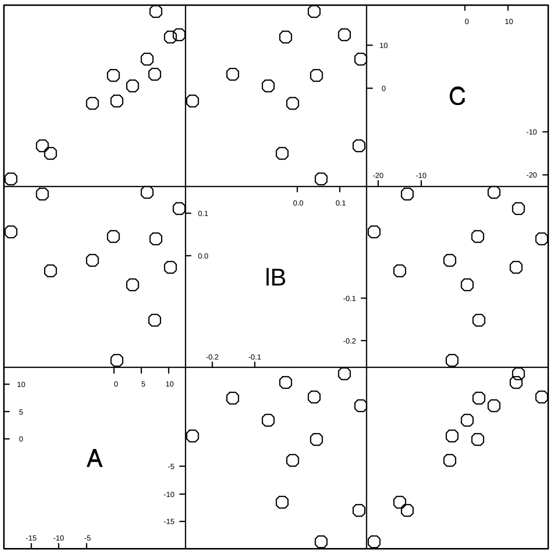

Figure 10: Scatter plot matrix of the estimated random effects in model (4).

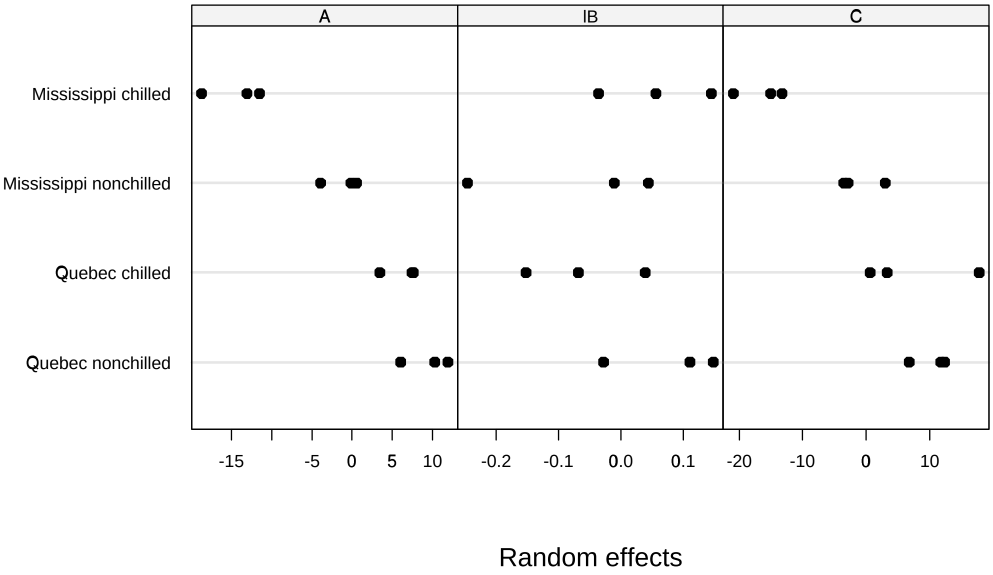

Figure 11: Estimated random effects versus plant type and chilling treatment.

These plots indicate that chilled plants tend to have smaller values of A and C, but the Mississippi plants seem to be much more affected than the Québec plants, suggesting an interaction effect between plant type and chilling treatment. There is no clear pattern of dependence between lB and the treatment factors, suggesting that this parameter is not significantly affected by either plant type or chilling treatment.

We can then update the fitted object letting the A and C fixed effects depend on the treatment factors, as below.

```
> CO2.fit2 <- update(CO2.fit1,
+   fixed = list(A+C ˜ Treatment * Type, lB ˜ 1),
+   start = c(32.55, 0, 0, 0, 41.56, 0, 0, 0, -4.6))
```

The `summary` method provides more detailed information on the new fitted object.

```
> summary(CO2.fit2)
Nonlinear mixed-effects model fit by maximum likelihood
  Model: uptake ˜ co2.func(conc, A, lB, C)
 Data: CO2
     AIC   BIC logLik
  392.41 431.3 -180.2

Random effects:
 Formula: list(A ˜ 1, lB ˜ 1, C ˜ 1)
 Level: Plant
 Structure: General positive-definite
               StdDev   Corr
A.(Intercept) 2.37058 A.(In) lB
           lB 0.14749 -0.336
C.(Intercept) 8.16451  0.356  0.761
     Residual 1.71134

Fixed effects: list(A + C ˜ Treatment * Type, lB ˜ 1)
                   Value Std.Error DF t-value p-value
   A.(Intercept)  42.249     1.498 64  28.212   0.000
     A.Treatment  -3.692     2.058 64  -1.794   0.078
          A.Type -11.078     2.065 64  -5.366   0.000
A.Treatment:Type  -9.575     2.943 64  -3.254   0.002
   C.(Intercept)  46.300     6.436 64   7.194   0.000
     C.Treatment   8.830     7.230 64   1.221   0.226
          C.Type   3.010     8.048 64   0.374   0.710
C.Treatment:Type -49.019    17.679 64  -2.773   0.007
              lB  -4.651     0.080 64 -58.069   0.000
. . .
```

The small p-values of the t-statistics associated with the `Treatment:Type` effects indicate that both factors have a significant effect on parameters A and C and their joint effect is not just the sum of the individual effects. We can investigate the joint effect of `Treatment` and `Type` on A and C using the `anova` method.

```
> anova(CO2.fit2,
+   terms = c("A.Treatment", "A.Type", "A.Treatment:Type"))
F-test for: A.Treatment, A.Type, A.Treatment:Type
  numDF denDF F-value P-value
1     3    64  51.782       0
> anova(CO2.fit2,
+   terms = c("C.Treatment", "C.Type", "C.Treatment:Type"))
F-test for: C.Treatment, C.Type, C.Treatment:Type
  numDF denDF F-value P-value
1     3    64    2.94    0.04
```

The p-values of the Wald F-tests suggest that `Treatment` and `Type` have a stronger influence on A than on C.

Diagnostic plots can be obtained using the plot method, in the exact same way as for `lme` objects. For example, plots of the standardized residuals versus fitted values broken up by `Treatment` and `Type`, shown in Figure 12, are obtained with

```
> plot(CO2.fit2,
+   resid(., type = "p") ~ fitted(.) | Treatment * Type)
```

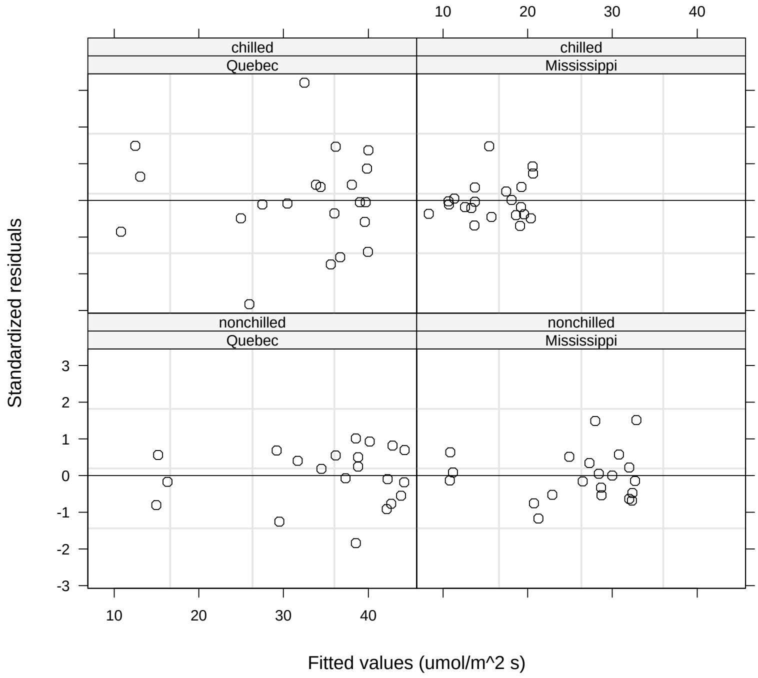

Figure 12: Standardized residuals versus fitted values for the `CO2.fit2` fit, by plant type and chilling treatment.

The plots do not indicate any departures from the assumptions in the model — no outliers seem to be present and the residuals are symmetrically scattered around the

$y = 0$ line, with constant spread for different levels of the fitted values.

Predictions are returned by the `predict` method. For example, to obtain the population predictions of the CO_2 uptake rate for Québec and Mississippi plants under chilling and no chilling, at ambient CO_2 concentrations of $75, 100, 200$, and $500\mu L/L$, we would first define

```
> CO2.new <-
+   data.frame(Type = rep(c("Quebec","Mississippi"), c(8, 8)),
+     Treatment =rep(rep(c("chilled","nonchilled"),c(4,4)),2),
+     conc = rep(c(75, 100, 200, 500), 4))
```

and then use

```
> predict(CO2.fit2, CO2.new, level = 0)
 [1]  6.7850 11.9669 23.7850 30.7508  8.3637 10.3910 15.0145
 [8] 17.7397 10.1335 16.9579 32.5219 41.6956  6.6677 13.4441
[15] 28.8986 38.0078
attr(, "label"):
[1] "Predicted values (umol/mˆ2 s)"
```

to obtain the predictions.

The `augPred` method can be used for plotting smooth fitted curves by calculating fitted values at closely spaced concentrations. Figure 13 presents the individual fitted curves for all twelve plants evaluated at 51 concentrations between 50 and 1000 $\mu L/L$, obtained with

```
> plot(augPred(CO2.fit2))
```

The `CO2.fit2` model explains the data reasonably well, as evidenced by the close agreement between its fitted values and the observed uptake rates.

4.3 `pdMat`, `corStruct`, and `varFunc` objects

All classes of positive-definite matrices, correlation structures, and variance functions described in §3.5 and §3.6 can be used with the `nlme` function, in the exact same way as with `lme`. For example, to test if the random effects in `CO2.fit2` can be assumed to be independent, we can use

```
> CO2.fit3 <- update(CO2.fit2, random = pdDiag(A+lB+C˜1))
```

```
> anova(CO2.fit2, CO2.fit3)
         Model df    AIC    BIC logLik    Test Lik.Ratio
```

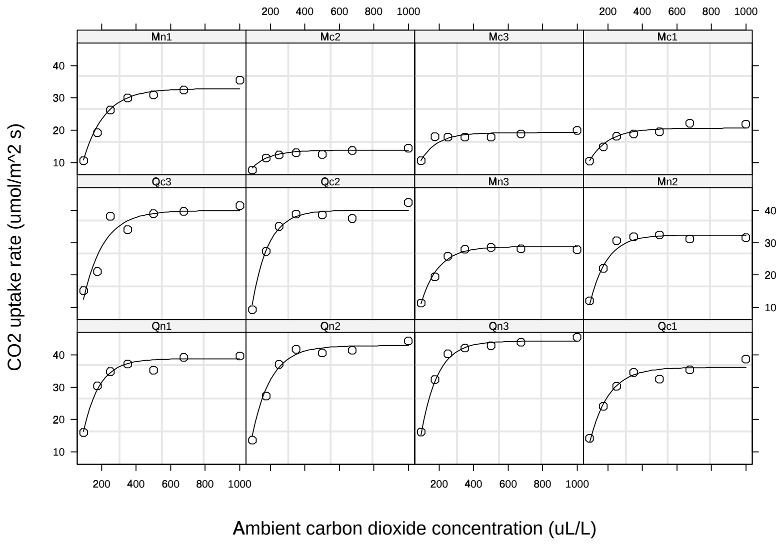

Figure 13: Individual fitted curves for the twelve plants in the CO_2 uptake data based on the `CO2.fit2` object.

```
CO2.fit2     1 16 392.41 431.30 -180.2
CO2.fit3     2 13 391.39 422.99 -182.7 1 vs. 2    4.9846
         p-value
CO2.fit2
CO2.fit3 0.17293
```

The large p-value of the likelihood ratio test suggests that the assumption of independence is reasonable.

To test for the presence of serial correlation in the within-group errors, we can use

```
> CO2.fit4 <- update(CO2.fit3, correlation = corAR1())
> anova(CO2.fit3, CO2.fit4)
         Model df    AIC    BIC  logLik    Test Lik.Ratio
CO2.fit3     1 13 391.39 422.99 -182.70
CO2.fit4     2 14 393.30 427.33 -182.65 1 vs. 2  0.092787
         p-value
CO2.fit3
CO2.fit4 0.76066
```

There does not appear to be evidence of within-group serial correlation.

Methods for extracting components from a fitted `nlme` object are also available and parallel those for `lme` objects. Some of the most commonly used are `coef`, `fitted`, `fixef`, `ranef`, and `resid`.

5 Conclusion

The functions, classes, and methods described here provide a comprehensive set of tools for analyzing linear and nonlinear mixed-effects models with an arbitrary number of nested grouping levels. As they are defined within the `S` environment, all the powerful analytical and graphical machinery present in `S` is simultaneously available. The analyses of the `Orthodont`, `Pixel` and `CO2` data illustrate some of the available features, but many other features are available.

References

Becker, R. A., Cleveland, W. S. and Shyu, M.-J. (1996). The visual design and control of trellis graphics displays, *J. of Computational and Graphical Statistics* **5**(2): 123–156.

Chambers, J. M. and Hastie, T. J. (eds) (1992). *Statistical Models in S*, Wadsworth, Belmont, CA.

Goldstein, H. (1995). *Multilevel Statistical Models*, Halstead Press, New York.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data, *Biometrics* **38**: 963–974.

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

Littel, R. C., Milliken, G. A., Stroup, W. W. and Wolfinger, R. D. (1996). *SAS Systems for Mixed Models*, SAS Institute Inc., Cary, NC.

Longford, N. T. (1993). *Random Coefficient Models*, Oxford University Press, New York.

Milliken, G. A. and Johnson, D. E. (1992). *Analysis of messy data. Volume 1: Designed experiments*, Chapman & Hall.

Pinheiro, J. C. (1994). *Topics in Mixed Effects Models*, PhD thesis, University of Wisconsin–Madison.

Potthoff, R. F. and Roy, S. N. (1964). A generalized multivariate analysis of variance model useful especially for growth curve problems, *Biometrika* **51**: 313–326.

Potvin, C., Lechowicz, M. J. and Tardif, S. (1990). The statistical analysis of ecophysiological response curves obtained from experiments involving repeated measures, *Ecology* **71**: 1389–1400.