lme for SAS PROC MIXED Users

Douglas M. Bates
Department of Statistics
University of Wisconsin – Madison *

José C. Pinheiro
Bell Laboratories
Lucent Technologies

1 Introduction

The `lme` function from the `nlme` library for S-PLUS or the `lme` library for R is used to fit linear mixed-effects models. It is similar in scope to the SAS procedure `PROC MIXED` described in Littell, Milliken, Stroup and Wolfinger (1996).

A file on the SAS Institute web site (http://www.sas.com) contains all the data sets in the book and all the SAS programs used in Littell et al. (1996). We have converted the data sets from the tabular representation used for SAS to the `groupedData` objects used by `lme`. To help users familiar with `SAS PROC MIXED` get up to speed with `lme` more quickly, we provide transcripts of some `lme` analyses paralleling the `SAS PROC MIXED` analyses in Littell et al. (1996).

In this paper we highlight some of the similarities and differences of `lme` analysis and `SAS PROC MIXED` analysis.

2 Similarities between lme and SAS PROC MIXED

Both `SAS PROC MIXED` and `lme` can fit linear mixed-effects models expressed in the Laird-Ware formulation. For a single level of grouping Laird and Ware (1982) write the n_i-dimensional response vector $\boldsymbol{y}_i$ for the ith experimental unit as

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, M \\ \boldsymbol{b}_i &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I}) \end{aligned} \tag{1}$$

where $\boldsymbol{\beta}$ is the p-dimensional vector of *fixed effects*, $\boldsymbol{b}_i$ is the q-dimensional vector of *random effects*, $\boldsymbol{X}_i$ (of size $n_i \times p$) and $\boldsymbol{Z}_i$ (of size $n_i \times q$) are known fixed-effects and random-effects regressor matrices, and $\boldsymbol{\epsilon}_i$ is the n_i-dimensional *within-group error* vector with a spherical Gaussian distribution. The assumption $\mathrm{Var}(\boldsymbol{\epsilon}_i) = \sigma^2\boldsymbol{I}$ can be relaxed using additional arguments in the model fitting.

The basic specification of the model requires a linear model expression for the fixed effects and a linear model expression for the random effects. In `SAS PROC MIXED` the

*This research was supported by the National Science Foundation through grant DMS-9704349.

fixed-effects part is specified in the `model` statement and the random-effects part in the `random` statement. In `lme` the arguments are called `fixed` and `random`.

Both `SAS PROC MIXED` and `lme` allow a mixed-effects model to be fit by maximum likelihood (`method = ml` in SAS) or by maximum residual likelihood, sometimes also called restricted maximum likelihood or REML. This is the default criterion in `lme` and `SAS PROC MIXED`. To get ML estimates in `lme`, set the optional argument `method="ML"`.

3 Important differences

The output from `PROC MIXED` typically includes values of the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). These are used to compare different models fit to the same data. The output of the `summary` function applied to the object created by `lme` also produces values of AIC and BIC but the definitions used in `PROC MIXED` and in `lme` are different. In `lme` the definitions are such that "smaller is better". In `PROC MIXED` the definitions are such that "bigger is better".

When models are fit by REML, the values of AIC, BIC and the log-likelihood can only be compared between models with exactly the same fixed-effects structure. When models are fit by maximum likelihood these criteria can be compared between any models fit to the same data. That is, these quality-of-fit criteria can be used to evaluate different fixed-effects specifications or different random-effects specifications or different specifications of both fixed effects and random effects.

The optimization algorithm in `lme` uses an unrestricted parameterization for the random effects variance-covariance components (Pinheiro and Bates, 1996), which enforces positive-definiteness of the estimated variance-covariance matrix for the random effects. Confidence intervals on the variance-covariance components are obtained in an unrestricted scale and then transformed back to the original scale. The resulting confidence intervals are always contained in the parameter space.

4 Data manipulation

Both `PROC MIXED` and `lme` work with data in a tabular form with one row per observation. There are, however, important differences in the internal representations of variables in the data.

In SAS a qualitative factor can be stored either as numerical values or alphanumeric labels. When a factor stored as numerical values is used in `PROC MIXED` it is listed in the `class` statement to indicate that it is a factor. In S-PLUS this information is stored with the data itself by converting the variable to a factor when it is first stored. If the factor represents an ordered set of levels, it should be converted to an `ordered` factor.

For example the SAS code

```
data animal;
 input trait animal y;
 datalines;
1 1 6
```

```
1 2 8
1 3 7
2 1 9
2 2 5
2 3 .
;
```

would require that the `trait` and `animal` variables be specified in a class statement in any model that is fit.

In S-PLUS these data could be read from a file, say `animal.dat`, and converted to factors by

```
S> animal <- read.table( "animal.dat", header = TRUE )
S> class( animal )
[1] "data.frame"
S> animal$trait <- as.factor( animal$trait )
S> animal$animal <- as.factor( animal$animal )
```

In general it is a good idea to check the types of variables in a data frame before working with it. One way of doing this is to apply the function data.class to each variable in turn using the `sapply` function.

```
S> sapply( animal, data.class )
    trait    animal         y
 "factor"  "factor" "numeric"
```

To make specification of models in `lme` easier and to make graphic presentations more informative, we recommend converting from a `data.frame` object to a `groupedData` object. This class of objects contains a formula specifying the response, the primary covariate (if there is one) and the grouping factor or factors. The data sets from Littell et al. (1996) have been converted to `groupedData` objects in this directory.

4.1 Unique levels of factors

Designs with nested grouping factors are indicated differently in the two languages. An example of such an experimental design is the semiconductor experiment described in section 2.2 of Littell et al. (1996) where twelve wafers are assigned to four experimental treatments with three wafers per treatment. The levels for the wafer factor are `1`, `2`, and `3` but the wafer factor is only meaningful within the same level of the treatment factor, `et`. There is nothing associating wafer `1` of the third treatment group with wafer `1` of the first treatment group.

In SAS this nesting of factors is denoted by `wafer(et)`. In S-PLUS the nesting is written with `ET/Wafer` and read "wafer within ET". If both levels of nested factors are to be associated with random effects then this is all you need to know. You would use an expression with a `"/"` in the grouping factor part of the formula for the `groupedData` object. Then the random effects could be specified as

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

or, equivalently

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

In this case, however, there would not usually be any random effects associated with the "experimental treatment" or `ET` factor. The only random effects are at the `Wafer` level. It is necessary to create a factor that will have unique levels for each `Wafer` within each level of `ET`. One way to do this is to assign

```
S> Semiconductor$Grp <-
+  getGroups( Semiconductor, form = ~ ET / Wafer, level = 2 )
S> levels( Semiconductor$Grp )  # check on the distinct levels
 [1] "1/1" "1/2" "1/3" "2/1" "2/2" "2/3" "3/1" "3/2" "3/3" "4/1"
[11] "4/2" "4/3"
```

after which we could specify `random = 1 | Grp`.

4.2 General approach

As a general approach to importing data into S-PLUS for mixed-effects analysis you should:

- Create a `data.frame` with one row per observation and one column per variable.
- Use `ordered` or `as.ordered` to explicitly convert any ordered factors to class `ordered`.
- Use `ordered` or `as.ordered` to explicitly convert any ordered factors to class `ordered`.
- If necessary, use `getGroups` to create a factor with unique levels from inner nested factors.
- Specify the formula for the response, the primary covariate and the grouping structure to create a `groupedData` object from the data frame. Labels and units for the response and the primary covariate can also be specified at this time as can `outer` and `inner` factor expressions.
- Plot the data. Plot it several ways. The use of Trellis graphics is closely integrated with the `nlme` library. The Trellis plots can provide invaluable insight into the structure of the data. Use them.

5 Contrasts

When comparing estimates produced by `SAS PROC MIXED` and by `lme` one must be careful to consider the contrasts that are used to define the effects of factors. In SAS a model with an intercept and a qualitative factor is defined in terms of the intercept and the indicator variables for all but the last level of the factor. The default behaviour in S-PLUS is to use the Helmert contrasts for the factor. On a balanced factor these provide a set of orthogonal contrasts. In R the default is the "treatment" contrasts which are

almost the same as the SAS parameterization except that they drop the indicator of the first level, not the last level.

When in doubt, check which contrasts are being used with the contrasts function.

To make comparisons easier, you may find it worthwhile to declare

```
S> options(contrasts = c(factor = "contr.SAS",
+                        ordered = "contr.poly"))
```

at the beginning of your session.

6 An example: Average Daily Gain

These data, described in Appendix 4 of Littell et al. (1996), refer to an experiment in which steers were fed four different diets, corresponding to a base ration and three levels of a medicated feed additive added to the base ration, over a period of 160 days. The objective of the study was to determine the optimal level of feed additive to maximize the average daily gain (`adg`). A total of 32 steers were used in the experiment. They were housed in barns, which held four steers each. The initial weights of the steers (`IniWt`) were measured to serve as potential covariates to explain ADG.

The data are represented in S-PLUS as the groupedData object `AvgDailyGain`.

```
> AvgDailyGain[1:5,]
Grouped Data: adg ~ Trt | Block
   Id Block Treatment  adg InitWt Trt
 1  1     1         0 1.03    338   0
 2  2     1        10 1.54    477  10
 3  3     1        20 1.82    444  20
 4  4     1        30 1.86    370  30
 5  5     2         0 1.31    403   0
> plot(AvgDailyGain)                                    # Figure 1
```

An initial model proposed for these data in Littell et al. (1996) uses the initial weight as a covariate and assigns different intercepts and slopes for each diet, treated as a factor. A single random intercept is used to account for the *barn effect*. The model for the average daily gain corresponding to diet i in barn j, y_{ij}, is represented as

$$y_{ij} = \alpha_i + \beta_i x_{ij} + b_j + \epsilon_{ij}, \tag{2}$$

where x_{ij} is the initial weight, α_i and β_i are the diet intercept and slope, b_j are the random intercepts assumed to be independently distributed as $\mathcal{N}(0, \sigma_b^2)$, and ϵ_{ij} are the within-group error, assumed independently distributed as $\mathcal{N}(0, \sigma^2)$, and independent of the random effects.

The SAS commands used to fit such model are

```
proc mixed;
  class Treatment Block;
  model adg = Treatment Treatment*InitWt / noint solution;
  random blk;
```

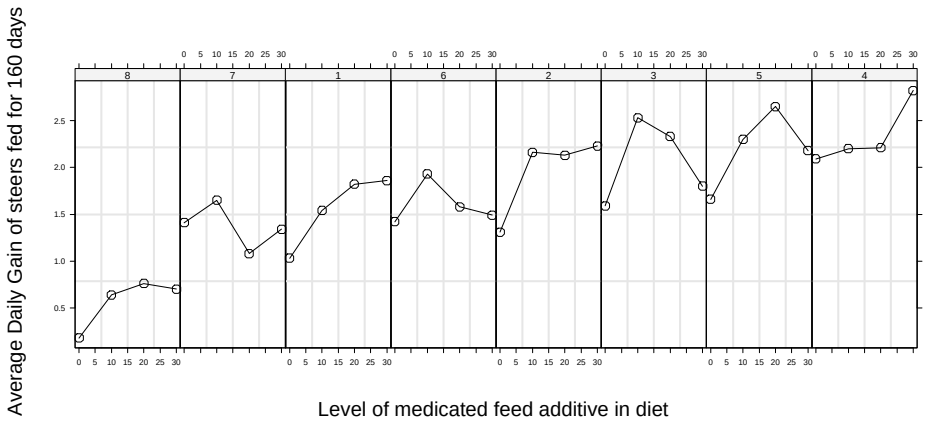

Figure 1: Average daily gain of 32 steers fed four different diets over 160 days. Panels correspond to barns where four steers were kept, each steer being fed a different diet.

The corresponding commands in S-PLUS are

```
> options(contrasts = c("contr.SAS", "contr.poly"))
> fm1ADG <- lme(adg ~ Treatment - 1 + InitWt:Treatment ,
+               data = AvgDailyGain, random = ~ 1 | Block)
> summary(fm1ADG)
Linear mixed-effects model fit by REML
 Data: AvgDailyGain
     AIC    BIC  logLik
  85.327 97.107 -32.663

Random effects:
 Formula:  ~ 1 | Block
        (Intercept) Residual
StdDev:     0.50923  0.22233

Fixed effects: adg ~ Treatment - 1 + Treatment:InitWt
                   Value Std.Error DF t-value p-value
       Treatment0 0.4391   0.71109 17  0.6176  0.5451
      Treatment10 1.4261   0.63755 17  2.2369  0.0390
      Treatment20 0.4796   0.54889 17  0.8738  0.3944
      Treatment30 0.2001   0.77520 17  0.2581  0.7994
 Treatment0InitWt 0.0023   0.00175 17  1.3127  0.2067
Treatment10InitWt 0.0011   0.00148 17  0.7298  0.4755
Treatment20InitWt 0.0034   0.00129 17  2.6152  0.0181
Treatment30InitWt 0.0044   0.00208 17  2.1368  0.0474


 Correlation:
                  Trtmn0 Trtmn1 Trtm20 Trtm30 Trt0IW Tr10IW Tr20IW
```

```
      Treatment10  0.039
      Treatment20  0.080  0.334
      Treatment30  0.011  0.097  0.043
 Treatment0InitWt -0.961  0.034  0.003  0.050
Treatment10InitWt  0.034 -0.951 -0.253 -0.033 -0.036
Treatment20InitWt  0.003 -0.258 -0.934  0.036 -0.004  0.271
Treatment30InitWt  0.050 -0.032  0.035 -0.967 -0.052  0.034 -0.037

Standardized Within-Group Residuals:
    Min       Q1       Med      Q3    Max
 -1.829 -0.44914 -0.030235 0.44739 1.5988

Number of Observations: 32
Number of Groups: 8
```

These results match closely the ones in Littell et al. (1996, §5.3, p. 178).

Next, we reparameterize model (2 as

$$y_{ij} = \alpha_4 + (\alpha_i - \alpha_4) + \beta_4 + (\beta_i - \beta_4)x_{ij} + b_j + \epsilon_{ij},$$

to test for differences in slope. This model is fit in SAS using

```
proc mixed;
  class Treatment Block;
  model adg = Treatment InitWt Treatment*InitWt / solution;
  random blk;
```

and in S-PLUS

```
> fm2ADG <- update(fm1ADG, adg ~ Treatment * InitWt)
> summary(fm2ADG)
Linear mixed-effects model fit by REML
 Data: AvgDailyGain
     AIC    BIC  logLik
  85.327 97.107 -32.663

Random effects:
 Formula:  ~ 1 | Block
        (Intercept) Residual
StdDev:     0.50923  0.22233

Fixed effects: adg ~ Treatment + InitWt + Treatment:InitWt
                    Value Std.Error DF t-value p-value
      (Intercept)  0.2001    0.7752 17  0.2581  0.7994
       Treatment0  0.2390    1.0464 17  0.2284  0.8220
      Treatment10  1.2260    0.9548 17  1.2841  0.2163
      Treatment20  0.2795    0.9305 17  0.3004  0.7675
           InitWt  0.0044    0.0021 17  2.1368  0.0474
 Treatment0InitWt -0.0022    0.0028 17 -0.7732  0.4500
Treatment10InitWt -0.0034    0.0025 17 -1.3381  0.1985
Treatment20InitWt -0.0011    0.0025 17 -0.4351  0.6690
. . .
```

The results are again nearly identical to the ones obtained with SAS. To test for differences in slope, we use the anova method in S-PLUS

```
> anova(fm2ADG, type = "m")
                 numDF denDF F-value p-value
     (Intercept)     1    17  0.0666  0.7994
       Treatment     3    17  0.8706  0.4755
          InitWt     1    17  4.5660  0.0474
Treatment:InitWt     3    17  0.9312  0.4471
```

There is no significant evidence that the slopes change with diet. We update the fit to a model with a common slope in S-PLUS using

```
> AvgDailyGain$Treatment <- ordered(AvgDailyGain$Treatment)
> fm3ADG <- update(fm2ADG, adg ~ Treatment + InitWt)
> summary(fm3ADG)
Linear mixed-effects model fit by REML
 Data: AvgDailyGain
     AIC    BIC  logLik
  51.724 60.794 -18.862

Random effects:
 Formula:  ~ 1 | Block
        (Intercept) Residual
StdDev:     0.49076  0.22379

Fixed effects: adg ~ Treatment + InitWt
              Value Std.Error DF t-value p-value
(Intercept)  0.62392   0.37024 20  1.6851  0.1075
Treatment.L  0.36597   0.08142 20  4.4948  0.0002
Treatment.Q -0.19769   0.08245 20 -2.3978  0.0264
Treatment.C  0.13657   0.07912 20  1.7261  0.0997
     InitWt  0.00278   0.00083 20  3.3356  0.0033
. . .
```

There is evidence of linear and quadratic effects of level of medicated feed additive, but no significant evidence of a cubic effect.

References

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

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

Pinheiro, J. C. and Bates, D. M. (1996). Unconstrained parameterizations for variance-covariance matrices, *Statistics and Computing* **6**: 289–296.