Software Design for Nonlinear Mixed Effects

D. M. Bates

Department of Statistics
University of Wisconsin
Madison, WI 53706-1685

J. C. Pinheiro

Bell Labs
600 Mountain Ave
Murray Hill, NJ 07974-0636

Abstract

Software design for population pharmacokinetic analysis presents many challenging problems. The data can range from relatively small data sets with simple structure to comparatively large data sets collected in routine clinical settings. These larger collections often have a complicated structure which makes graphics or tabular presentation of the results of an analysis challenging. Current approaches to modeling such data, especially with nonlinear mixed-effects models, present difficult computational problems. Since the statistical approaches to such modeling are an active area of research, we want the software used for the modeling to be as flexible as possible.

Thus we have a need to deal with complicated, large data sets and with difficult models for which computational efficiency is important while still providing flexibility and ease of use. Of course, good graphical displays are expected as well.

We have found that an object-oriented approach provided by environments such as S-PLUS is extremely valuable to us in designing such software. We describe some of the problems we faced and our solutions to them including groupedData objects, selfStart nonlinear regression models, nlsList objects and classes of model "add-on's". The use of our nonlinear mixed-effects software is illustrated through the analysis of a "real data" example.

1 Introduction

Software design for population pharmacokinetic analysis and, more generally, for longitudinal data analysis presents many challenging problems. Recent developments in graphical display of data with the trellis approach (Cleveland, 1994; Becker, Cleveland and Shyu, 1996) and in object-oriented languages for data analysis with S version 4 (Chambers, 1993), offer new and efficient solutions to many of these problems.

We describe some of the problems we faced while designing software for mixed-effects models and our solutions to them using the trellis approach and object-orientation, including groupedData objects, selfStart nonlinear regression models, nlsList and nlme objects, and classes of model "add-on's".

In §2 we describe the different capabilities of our nlme software for mixed-effects models through the analysis of a typical longitudinal data set. Our conclusions are given in §3.

2 An example of nonlinear mixed effects

Longitudinal data, such as pharmacokinetics data, can range from small data sets with simple structure to comparatively large data sets with complicated structures. There are several characteristics in common to longitudinal data sets, though. They consist of measurements of a continuous response at several levels of a continuous covariate, usually time or dose. Further, these measurements are grouped according to a factor. Additional covariates may be present. Some of these vary within a group (*inner* covariates) and some do not (*outer* covariates).

An example of longitudinal data with these characteristics, which will be used throughout this section, is given by the Soybean data described in Davidian and Giltinan (1995, § 1.1.3, p. 7) as "Data from an experiment to compare growth patterns of two genotypes of soybeans: Plant Introduction #416937 (P), an experimental strain, and Forrest (F), a commercial variety." The response variable for these data is the leaf weight (in grams), the primary covariate is time (in days after planting), and the grouping factor is plot number. Additionally, two outer factors are included in the data: the variety used in the plot and the year of planting.

The choice of a data structure for longitudinal data will affect the ease and flexibility with which we can display the data and fit models to the data. We have chosen to create a new class of object (Chambers and Hastie, 1992, Appendix A), which we have called groupedData, to represent this type of data in S. The groupedData class stores the data as a data.frame (i.e. a rectangular array), but also packs additional information about the the different variables in the data.

For example, reading the Soybean data from a file and converting it to a groupedData object would be accomplished by

```
> Soybean <- groupedData(weight ~ time | Plot,
+    data = read.table("Soybean", header = T),
+    outer = ~ Year * Variety,
+    labels = list(x = "Time since planting",
+                  y = "Leaf weight/plant"),
+    units = list(x = "(days)", y = "(g)"))
> Soybean
Grouped Data: weight ~ time | Plot
      Plot Variety Year time    weight
  1 1988F1       F 1988   14  0.106000
  2 1988F1       F 1988   21  0.261000
. . .
411 1990P8       P 1990   64 16.411700
412 1990P8       P 1990   79 16.946700
```

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

The plot method for the groupedData class gives a trellis display of the data.

```
> plot(Soybean, layout = c(8, 6), # Figure 1
+      between = list(y = c(0, 0.5)))
>   # 'layout' and 'between' are trellis arguments
>   # used to enhance the layout of the panels
```

We can see a definite curvature to each of these growth curves, leading us to consider models such as the three-parameter logistic regression model

$$\theta_1 / \left\{1 + \exp\left[\left(\theta_2 - t\right)/\theta_3\right]\right\} \qquad (1)$$

The groupedData plot method allows an optional argument outer which can be given a logical value or a formula. A logical value of TRUE indicates that the outer formula stored with the data should be used in the plot. For example

```
> plot(Soybean, outer = T)   # Figure 2
```

helps visualizing the differences in leaf weight between the varieties and among the years of planting. A more detailed description of the groupedClass class and its related methods can be found in Bates and Pinheiro (1997).

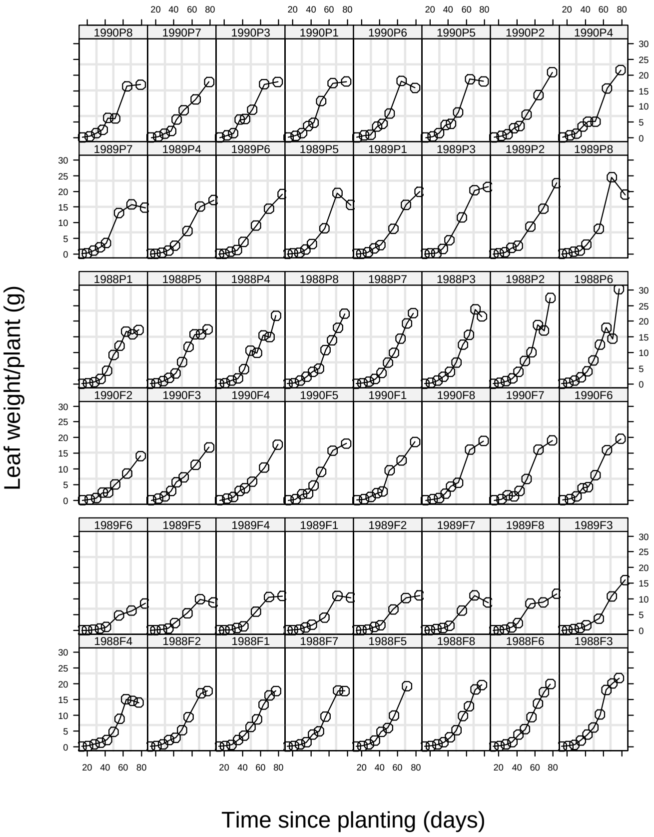

Figure 1: Average leaf weight per plant versus time since planting for plots of soybeans. The plots are from two different years and represent two different genotypes of soybeans.

The approach we have taken in nlme is to provide a variety of tools for building linear or nonlinear mixed-effects models. The final model can relate fixed effects to covariates and can include sophisticated models for random effects, for serial correlation in the within-group noise term, or for the variance of the within-group noise term. We expect, however, that the user will build the model interactively from preliminary fits rather than trying to fit an "omnibus" model from the beginning.

Of course, this iterative model building must start from somewhere. We usually consider that "somewhere" to be individual fits of a nonlinear model like the logistic regression model (1) for each level of the grouping factor or, when an inner factor is present, for each level of the inner factor within the grouping factor. Because we can determine the response, the primary covariate, and the grouping factor from the formula stored with the data, it is straightforward to set up the necessary nonlinear regression calls. There is, however, one part of the non-

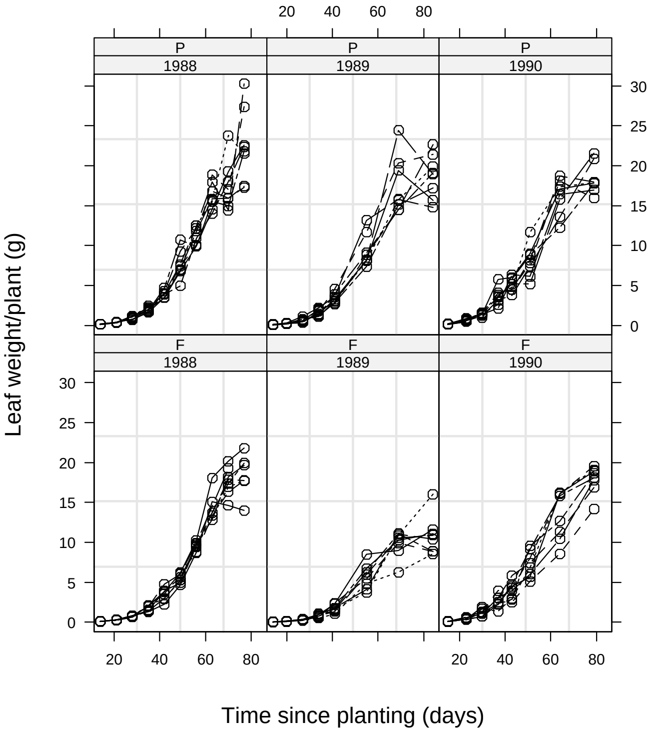

Figure 2: Average leaf weight per plant versus time since planting for plots of soybeans. The plots are grouped by each combination of three different planting years and two different genotypes of soybeans.

linear regression that is always problematic; determining starting values for the parameter estimates.

When the groups of data are similar, a single set of starting values can be applied to all the groups. However, when there are marked differences between the groups, such as the Soybean data, a single set of starting values may not be adequate. An experienced analyst will often be able to find suitable starting values for each group quickly by applying a few common "tricks". For example, starting values for the logistic regression model can be obtained by:

1. Form an initial estimate $\theta_1^{(0)}$ of the asymptote θ_1 from a plot of the data.
2. Regress $\log\left(\theta_1^{(0)}/y - 1\right)$ on t. The estimated intercept is $\theta_2^{(0)}$ and the estimated slope is $-1/\theta_3^{(0)}$.
3. Use an algorithm for partially linear models (Bates and Chambers, 1992, §10.2.5) to refine estimates of θ_1, θ_2, and θ_3.

These steps form an algorithm for calculating starting values in a robust way. In this case, the "starting values" would also be the final parameter estimates but that isn't required.

Based upon examples like this we created a class of nonlinear regression model functions that have an auxillary function to calculate the starting values. We call these *self-starting* nonlinear regression models. The corresponding class is selfStart. Venables and Ripley (1996) describe the method we used in nlme 2.1 for communicating data to the function that calculates the starting values.

Self-starting regression models may be used in conjunction with the nlsList function to generate individual fits per group for a groupedData object. The formula descriptor of the groupedData object provides the necessary information about the roles of the variables, so preliminary fits for the Soybean data can be obtained simply with

```
> Soybean.lis <- nlsList(ll, Soybean)
> coef(Soybean.lis)
             Asym      xmid      scal
1988F4  15.152181  52.83510  5.177737
1988F2  19.746046  56.57598  8.407161
. . .
1988P2  36.644323  66.54816 11.912636
1988P6 163.038801 104.87821 17.924085
1989P7  15.471780  46.34263  5.393812
. . .
1989P8         NA        NA        NA
. . .
1990P4  26.129500  61.19955 10.971970
```

where ll is a self-starting function for the logistic model. The coefficients Asym, xmid, and scal correspond respectively to θ_1, θ_2, and θ_3 in the logistic model (1). Convergence was not attained for one plot with unusual growth pattern (1989P8) and extreme values for the estimates were obtained for one plot with little information on leaf weight growth (1988P6).

These individual fits provide useful preliminary information for building the nonlinear mixed-effects model to be used later. For example, to decide which parameters in the logistic model should be allowed to vary with Plot, we can use the intervals method

```
> plot(intervals(Soybean.lis), omitGroups = "1988P6")
>   # plot 1988P6 omitted because of
>   # its outlying values
```

The last call produces a trellis plot of confidence intervals on the parameter estimates as shown in Figure 3

All three parameters appear to vary with plot and a *full* nonlinear mixed-effects model, with all terms having fixed and random components seems adequate. The separate nonlinear regression fits can be used to describe this full nonlinear mixed-effects model and implicitly to provide starting estimates for the parameters

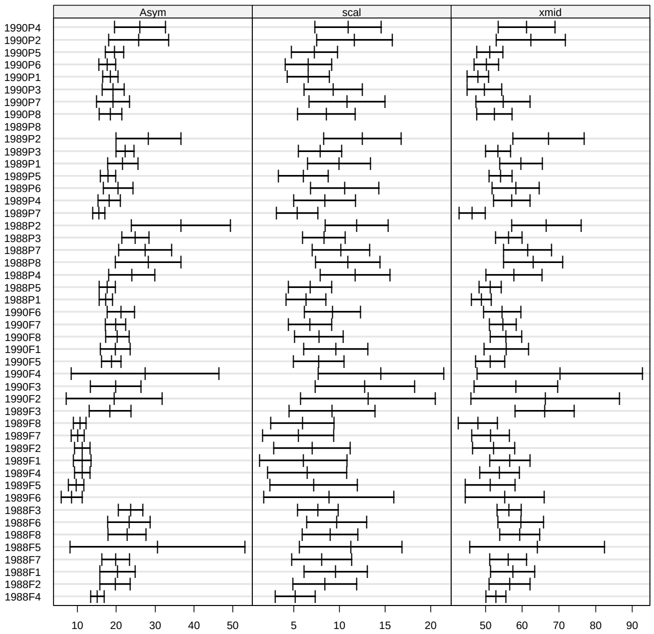

Figure 3: 95% confidence intervals on the parameter estimates of the logistic regression model fit separately to each plot in the Soybean data

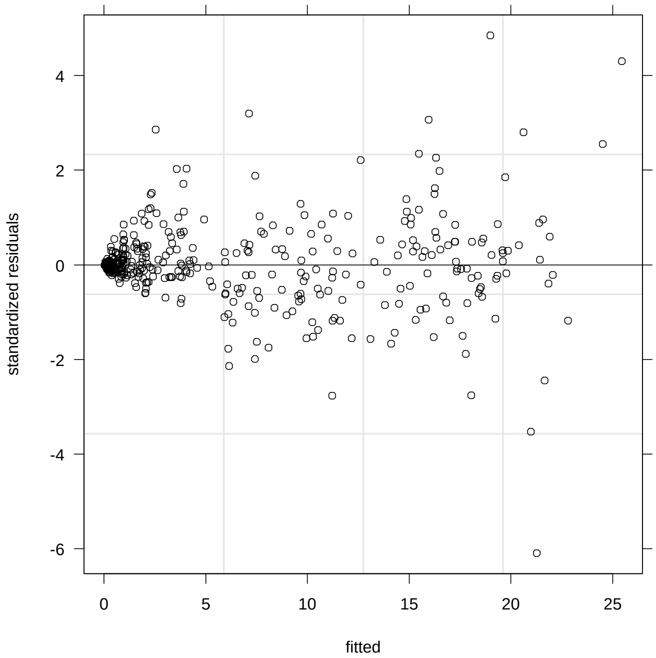

Figure 4: Standardized residuals versus fitted values in the nlme fit of the Soybean data.

```
> Soybean.nlme <- nlme(Soybean.lis)
```

The fitted object Soybean.nlme inherits from class nlme, and several methods are available for displaying, plotting, and updating it. We start by assessing the quality of the fit with the plot method.

```
> plot(Soybean.nlme)    # Figure 4
```

There is a clear indication that the dispersion increases with the value of the response. We can include that information in the nlme model using a variance function structure. In the fit below, we choose to use a variance function proportional to some (estimated) power of the response variable.

```
> Soybean.nlme2 <- update(Soybean.nlme,
+            varFStruct = varFPower())
```

The two fits may be compared with the anova method.

```
> anova(Soybean.nlme, Soybean.nlme2)
              Model df    AIC    BIC  logLik
 Soybean.nlme     1 10 1499.7 1539.9 -739.84
Soybean.nlme2     2 11  737.0  781.3 -357.52
                  Test Lik.Ratio P-value
 Soybean.nlme
Soybean.nlme2  1 vs. 2   764.64       0
```

The heterocedastic model provides a much better represention of the data. We can assess the adequacy of the *power* variance function structure using the plot method again.

```
> plot(Soybean.nlme2)   # Figure 5
```

The power variance function seems to model the within-plot heterocedasticity correctly.

It is interesting at this point, to compare the individual estimated coefficients of the nlme fit to those of the individual nonlinear regresion fits corresponding to Soybean.lis. The compareFits method can be used for this purpose.

```
>        # Figure 6
> plot(compareFits(Soybean.lis, Soybean.nlme2),
+      omitGroups = "1988P6", layout = c(3,1))
```

The coefficients corresponding to the separate fits vary much more than the nlme coefficients. This is typical of mixed-effects models fits. The separate fits are only concerned with the individual data, while the nlme fit takes into account the data as a whole, penalizing large deviations of the random effects from their population mean.

The primary question of interest for the Soybean data is the possible relation between the growth pattern of the soybean plants and the experimental factors Variety and Year. The random.effects method, with its companion plot method, provides as a useful tool for exploring the relationship between the individual parameters estimates and the experimental factors.

```
>            # Figure 7
```

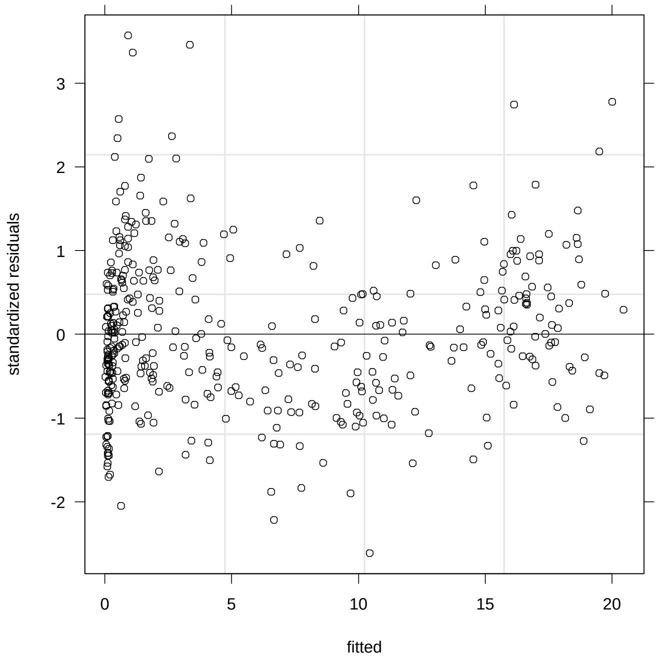

Figure 5: Standardized residuals versus fitted values in the nlme fit of the Soybean data, with heterocedastic error.

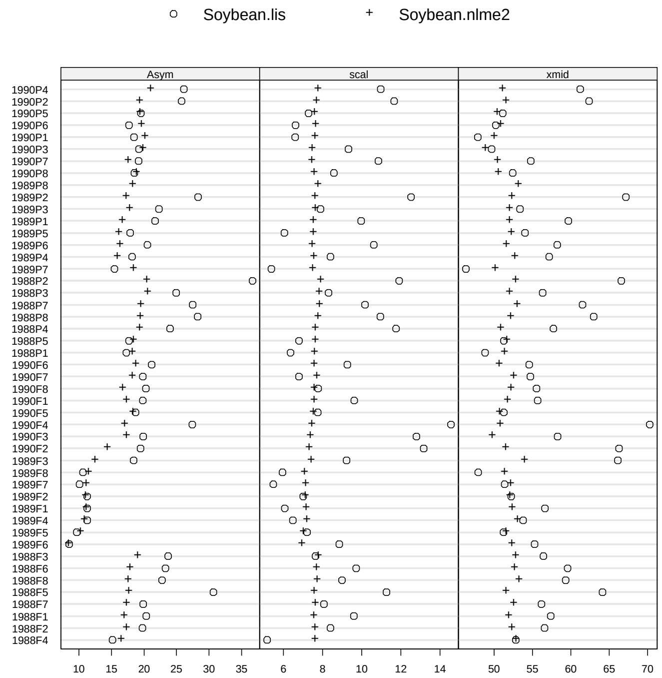

Figure 6: Individual estimated coefficients for nlme fit and separate non-linear regression fits.

```
> plot(random.effects(Soybean.nlme2,aug.frame = T),
+      outer = ~ Year * Variety, layout = c(3,1))
```

All three parameters seem to vary with year and variety. There appears that the asymptote (θ_1) and the scale (θ_3) are larger for P soybeans than for F soybeans and this difference is more pronounced in 1989. The time at which half of the asymptotic leaf weight is attained (θ_2) appears to be smaller among P soybeans than among F soybeans.

Linear modeling of parameters with respect to covariates can be done in nlme using the fixed argument. For example, we can model the dependence of all three parameters on Year with

```
> Soybean.nlme3 <- update(Soybean.nlme2,
+   fixed = list(Asym ~ Year, xmid ~ Year,
+                scal ~ Year))
> anova(Soybean.nlme2, Soybean.nlme3)
               Model df    AIC    BIC  logLik
 Soybean.nlme2     1 11 737.03 781.26 -357.52
 Soybean.nlme3     2 17 683.98 752.34 -324.99
                  Test Lik.Ratio    P-value
 Soybean.nlme2
 Soybean.nlme3 1 vs. 2    65.053 4.2069e-12
```

As suggested by Figure 7, Year has a very significant effect on the growth pattern of the soybean plants.

The estimated standard deviation for the scal random effect in the Soybean.nlme3 fit was only 0.04, corresponding to an estimated coefficient of variation of 0.5%. This suggests that θ_3 could be treated as a purely fixed effect and in fact, when we refit the model droping the scal random effect, we get a p-value of 0.99 for the corresponding likelihood ratio test. This happens quite frequently in model building for mixed-effects models: when covariates are brought into the model some of the random effects are no longer needed. In these cases, most, or all, of the between-group parameter variation is explained by the covariates included in the model.

Proceeding sequentially in the model building process by testing for the inclusion of covariates and the elimination of random effects, we end up with the following fitted object, with just one random effect for θ_1.

```
> summary(Soybean.nlme4)
Call:  Model: weight ~ ll(time, Asym, xmid, scal)
  Fixed: list(Asym ~ Year * Variety,
              xmid ~ Year + Variety, scal ~ Year)
 Random: list(Asym ~ .)
 Groups:  ~ Plot
   Data: Soybean
. . .
Fixed Effects Estimate(s):
                          Value Std.Error   z-value
    Asym.(Intercept) 19.4337563 0.9386001 20.705044
```

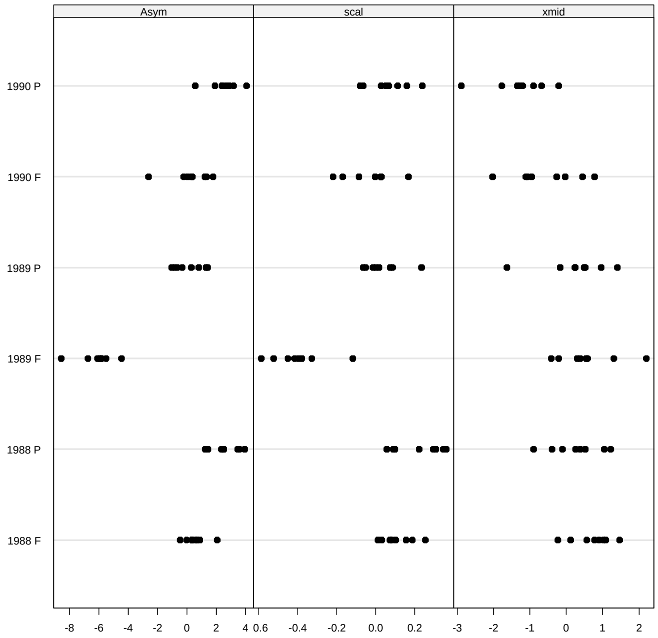

Figure 7: Random effects by Year and Variety in the nlme fit of the Soybean data.

```
         Asym.Year1989 -8.8421762 1.0551015 -8.380403
         Asym.Year1990 -3.7077747 1.1583028 -3.201041
          Asym.Variety  1.6232548 1.0217624  1.588681
Asym.Year1989Variety  5.5705385 1.1522732  4.834390
Asym.Year1990Variety  0.1473987 1.1570482  0.127392
     xmid.(Intercept) 54.8147656 0.7427505 73.799707
         xmid.Year1989 -2.2383957 0.9562405 -2.340829
         xmid.Year1990 -4.9711801 0.9587453 -5.185089
          xmid.Variety -1.2971909 0.4077588 -3.181270
     scal.(Intercept)  8.0633659 0.1448889 55.652079
         scal.Year1989 -0.8952299 0.1980641 -4.519901
         scal.Year1990 -0.6733403 0.2087876 -3.225001
. . .
```

The sequential improvements in goodness-of-fit during the model building process may be summarized with the anova method.

```
> anova(Soybean.nlme1, Soybean.nlme2, Soybean.nlme3,
+        Soybean.nlme4, test = F)
                Model df    AIC    BIC  logLik
  Soybean.nlme1     1 10 1499.7 1539.9 -739.84
  Soybean.nlme2     2 11  737.0  781.3 -357.52
  Soybean.nlme3     3 17  684.0  752.3 -324.99
  Soybean.nlme4     4 16  616.3  680.6 -292.15
```

A final model building issue that needs to be taken into account, especially with longitudinal data such as the Soybean data, is the possible presence of within-group serial correlation. The estimated autocorrelations and partial autocorrelations for the within-group residuals of the Soybean.nlme4 fitted object (not included here) give some indication that an MA(3) serial correlation structure would be adequate. Nevertheless, because of the relatively small number of observations per group (between 8 and 10), we decided not to include such a structure in the final model.

3 Conclusions

Acknowledgements

This research was supported by the National Science Foundation under grant number DMS-9309101 and by Lucent Technologies.

References

Bates, D. M. and Chambers, J. M. (1992). *Nonlinear Models*, in Chambers and Hastie (1992), chapter 10, pp. 421–454.

Bates, D. M. and Pinheiro, J. C. (1997). Software design for longitudinal data, *in* T. G. Gregoire (ed.), *Modelling Longitudinal and Spatially Correlated Data: Methods, Applications and Future Directions*, Springer-Verlag, New York. (Currently available as ftp://franz.stat.wisc.edu/pub/NLME/Nantucket.ps).

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. (1993). Classes and methods in S. II: Future directions, *Computatonal Statistics* **8**(3): 185–196.

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

Cleveland, W. S. (1994). *Visualizing Data*, Hobart Press.

Davidian, M. and Giltinan, D. M. (1995). *Nonlinear Models for Repeated Measurement Data*, 1st edn, Chapman & Hall, London.

Venables, W. N. and Ripley, B. D. (1996). Complements to Modern Applied Statistics with S-Plus. Supplement to the current edition of MASS covering new features.
URL: *http://www.stats.ox.ac.uk/pub/MASS/Compl.ps.gz*