Software Design for Longitudinal Data Analysis

Douglas M. Bates
José C. Pinheiro

ABSTRACT Software for exploring and modelling longitudinal data can be made much easier to use by incorporating an object-oriented design. Current versions of S-Plus provide some object-oriented capability but experimental versions of S emphasize an even stronger commitment to object orientation. These new capabilities, combined with the development of Trellis graphics by Cleveland and Becker, caused us to reexamine the basic design of our mixed-effects modelling functions - lme and nlme. We chose to implement a groupedData class with associated constructors, modeling, and display methods. This provides powerful visualization capabilities almost automatically. In addition it dramatically simplifies the interface to the modelling code.
We describe and illustrate the use of this approach and its impact on the modeling software with special emphasis on nonlinear mixed-effects models.

1 Introduction

Just as the statistical methodology for longitudinal data can be considerably more complicated than for cross-sectional data, the design of software for the exploration and modelling of longitudinal data is more complicated. As always with statistical software tradeoffs must be made between convenience for the user, sophistication of the analysis methods, and efficiency of the code. With longitudinal data additional issues of delineating experimental units or longitudinal characteristics must be considered. When sophisticated graphical techniques for data exploration or model diagnostics are also included the design problems become even greater.

For several years we have been developing classes and methods in the S language (Chambers and Hastie, 1992) for modelling longitudinal data with linear or nonlinear mixed-effects models. The nlme library is now included with versions 3.4 and higher of S-Plus from the Data Analysis Products Division of MathSoft, Inc. Its use is described in MathSoft (1996, chapter 2) and Venables and Ripley (1996, sections 6 and 9).

Although the library is useful in its current state, we have decided to change the design of the code and put a much greater emphasis on object-orientation. The primary reasons for the change are to enhance the use of trellis graphics displays; to provide a cleaner, more intuitive user interface; and to facilitate migration of the code to version 4 of the S language (Chambers, 1993). The redesign has affected several areas of the code. Here we will focus on two important areas: the use of groupedData objects to encapsulate the data and key aspects of the structure of the data, and the use of self-starting nonlinear regression models to derive starting estimates for nonlinear regression model parameters.

In §2 we describe trellis graphics displays for longitudinal data and the motivation they provide for groupedData objects. These objects are described in §3. The selfStart class of nonlinear regression models are described in §4 and our conclusions are given in §5.

2 Trellis graphics and longitudinal data

One of the most exciting recent developments in graphical display of data is the trellis approach (Cleveland, 1994; Becker, Cleveland and Shyu, 1996) to multi-panel displays of conditional plots. The approach is particularly valuable for longitudinal data that typically represent measurements of a response over time for different experimental units. We want to examine the behaviour within these units but also allow ourselves to compare behaviour between units. Trellis displays are well suited to this.

The approach is best shown with an example. Kung (1986) presents data, shown in Figure 1, on the growth of Loblolly pine trees. A common method of representing such data would be a plot with fourteen curves on it, one for each tree. Usually this type of plot becomes cluttered and difficult to read. In a trellis display like Figure 1 the measurements from different trees are plotted in separate panels but with consistent axes to facilitate comparison between panels. The panels are labelled with a strip giving an identifier, the seed source in this case, for the tree.

There are other, more subtle aspects to this plot. The order in which the panels are plotted is determined by a characteristic of the data. In this case, the panels have been ordered according to the maximum height measured for the tree. The shortest tree is from seed source 329 and is shown in the lower left panel. The tallest tree is from seed source 305 and is shown in the rightmost panel of the top row.

We can see a definite curvature to each of these growth curves leading us to consider models such as the asymptotic regression model

$$\theta_1 + (\theta_2 - \theta_1)\exp(-\theta_3 t), \quad \theta_3 > 0 \tag{1.1}$$

with parameters representing the asymptotic height, θ_1, the height at time zero, θ_2, and the rate constant, θ_3, for the growth of an individual tree. The other notable feature of these data is the consistency of the growth pattern between trees.

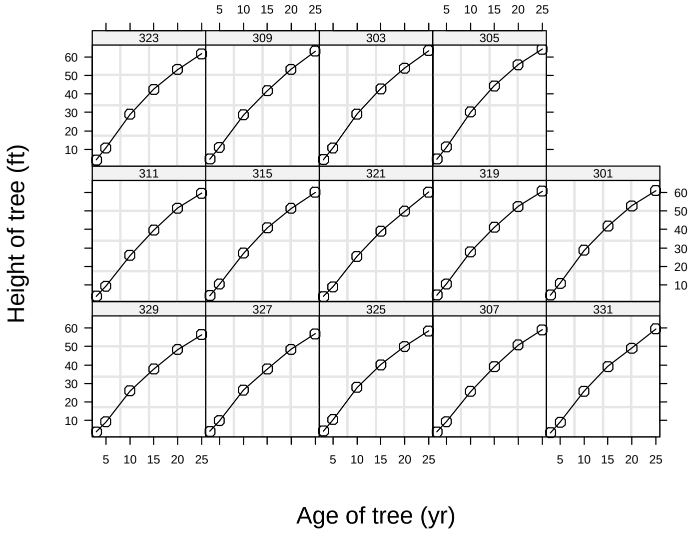

FIGURE 1. Height of Loblolly pine trees over time. Each panel is labelled with the number of the seed source for the tree.

Often in the analysis of longitudinal data we are looking for differences in the response patterns rather than consistency - especially if the differences are related to treatments applied to or characteristics of the experimental units. Trellis displays facilitate such comparisons. For example, Figure 2 shows the Soybean data described in Davidian and Giltinan (1995, § 1.1.3, p. 7). The data were collected on eight different plots each for two genotypes of soybean in each of three growing seasons. The presentation in Figure 2 allows evaluation of the aggregate behaviour for the genotype-year combinations. The layout with genotypes determining the row and year determining the column facilitates comparisons of the response from year to year. We can see, for example, that final weights for the 'P' genotype decreased more-or-less uniformly across years while the 'F' genotype had the lowest leaf weights in 1989. We may wish to consider an interaction between year and genotype when modelling these data.

The opposite arrangement, rows determined by genotype and columns determined by year (not shown), would enhance comparisons between genotypes within year. Even the arrangement in Figure 2 allows us to see that the 'P' genotype is producing heavier leaves than the 'F' genotype.

The aspect ratio of the individual panels in Figure 2 has been chosen by the "banking" method described in Cleveland (1994) to enhance comparisons of slopes.

Each of the panels in Figure 2 displays the data from eight soybean plots as overlaid lines. This makes the panels a bit too cluttered for evaluation of the behaviour of individual plots. That is best done by having one panel per

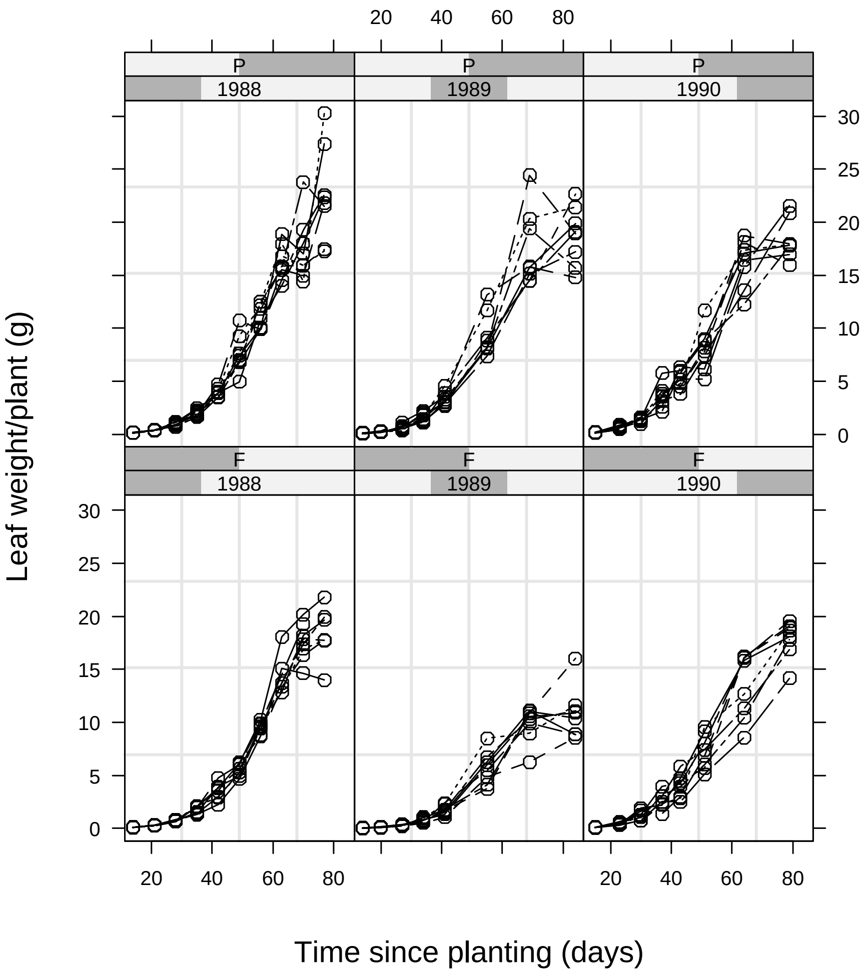

FIGURE 2. Average leaf weight per plant versus time since planting for plots of soybeans. The plots are from three different years and represent two different genotypes of soybeans.

plot and arranging the panels according to year and genotype (Figure 3). Here is not as easy as in Figure 2 to compare years or genotypes but this figure does reveal other characteristic of these data. We see there is considerably greater variability in the responses at the higher levels than at the lower levels. We may wish to consider modelling the variance of the response as an increasing function of the level of the response.

The ordering of the panels in Figure 3 preserves the levels of the Year and Variety (i.e. genotype) factors and applies increasing order of the maximum response within each combination of these factors.

In our terminology Year and Variety are *outer* factors for this experiment in that they do not vary within groups. (The "group" here is one plot for one year.) In contrast an *inner* factor can change within a group such as in cross-over trials where each subject is exposed to more than one treatment. For example, in the experiment described by Ludbrook (1994) and analysed in Venables and Ripley (1996) each of five rabbits was treated with the drug MDL 72222 and with a placebo. After treatment the rabbit was exposed to ascending doses of phenylbiguanide and its blood pressure was measured. The resulting data are shown in Figure 4. This figure strongly indicates that effect of MDL 72222 is to shift the curve to the right. Although different rabbits have different levels of the response the pattern of a shift to the right when treated with MDL 72222 is consistent across rabbits.

3 GroupedData objects

The data sets considered in the previous section have several characteristics in common. 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).

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. We have chosen throughout to represent the data as a data.frame (i.e. a rectangular array) in S where the columns represent variables and the rows represent cases.

At a minimum the data frame must contain the response, the primary covariate, such as time, and the grouping factor, such as Seed or Plot. Additional factors or continuous covariates can be present. For example

```
> names(Loblolly)  # Loblolly pine trees
[1] "height" "age"    "Seed"
> names(Soybean)   # Soybean growth
[1] "weight"   "time"   "Plot"    "Variety" "Year"
> names(PBG)        # Blood pressure vs phenylbiganide dose
[1] "deltaBP"   "dose"    "Treatment" "Rabbit"
```

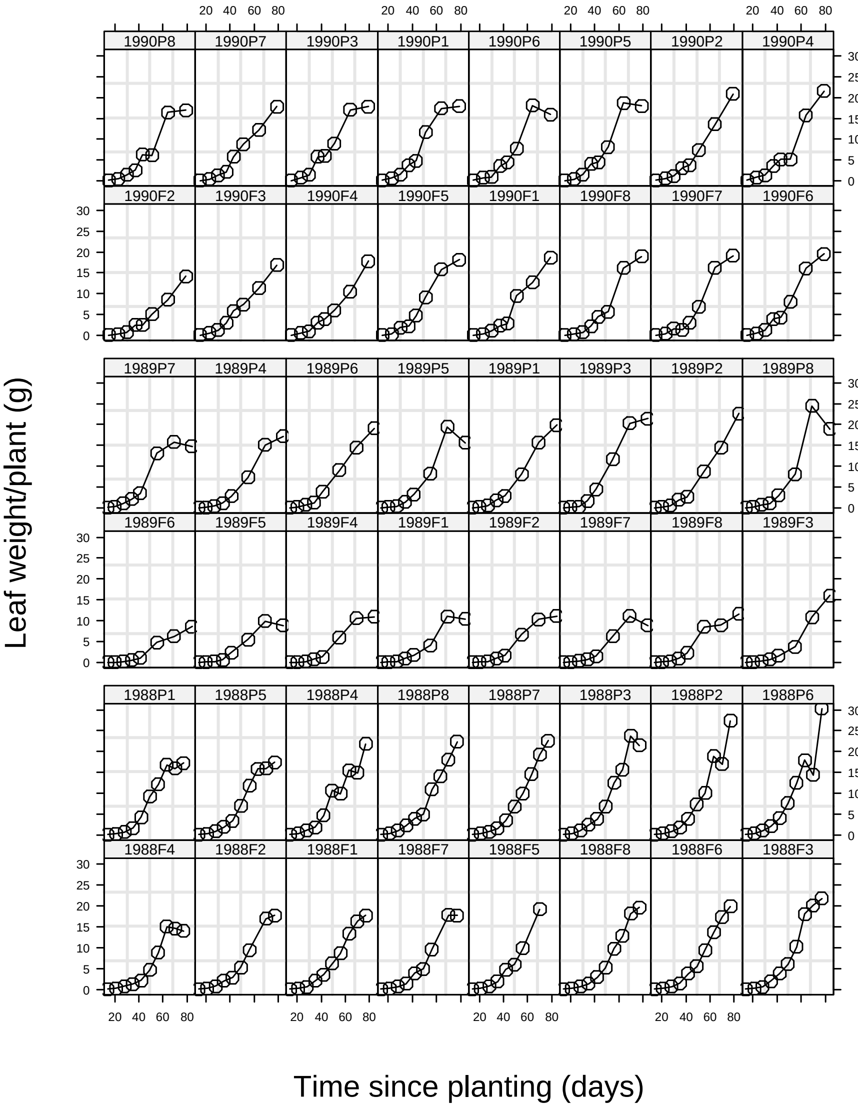

FIGURE 3. Average leaf weight per plant versus time since planting for plots of soybeans. The plots are from three different years and represent two different genotypes of soybeans. The label on a panel combines the year, the genotype, and the sequence number within that year and genotype for the experimental plot.

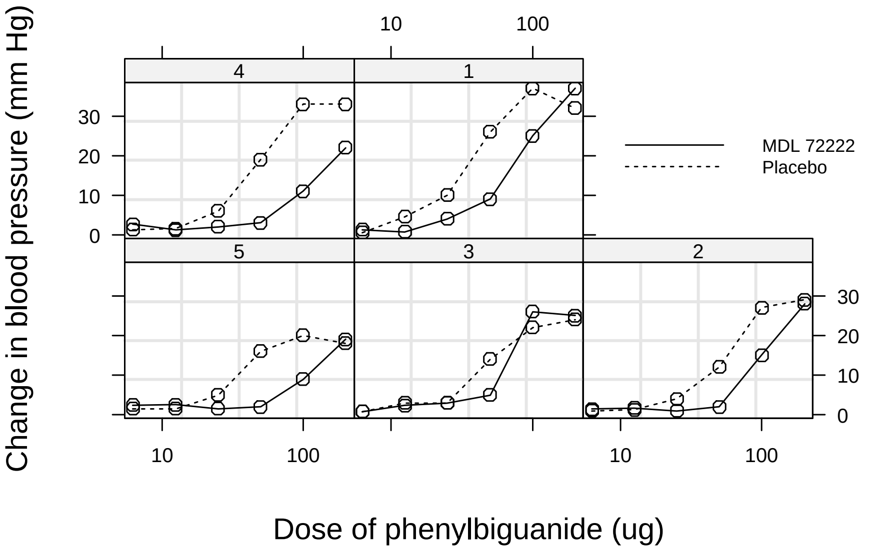

FIGURE 4. Change in blood pressure of five rabbits after ascending doses of phenylbiguanide (PBG). Each rabbit had one set of measurements made after treatment with the 5-HT_3 antagonist MDL 72222 and one set made after comparable treatment with a placebo. The dose of PBG is on a logarithmic scale.

The natural trellis plots for such data depict the response versus the primary covariate in panels determined by the grouping factor. They are described by a formula of the form

```
response ~ primary | grouping
```

We found that we were writing that type of formula so frequently that it made sense to incorporate it in the data. For example

```
> formula(Loblolly)
height ~ age | Seed
```

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 factor is converted to an ordered factor with the order determined by a summary function applied to the response split according to the groups. (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, reading the Loblolly pine tree data from a file and converting it to a groupedData object would be accomplished by

```
> Loblolly <- groupedData(height ~ age | Seed,
+    data = read.table("Loblolly", header = T),
+    labels = list(x = "Age of tree", y = "Height of tree"),
+    units = list(x = "(yr)", y = "(ft)"))
> plot(Loblolly)     # produces Figure 1
```

This call to the 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.

When outer factors are present, as for the Soybean data, they are given as a formula such as

```
outer = ~ Variety * Year
```

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 soybean plots of the same genotype in the same year are grouped together in Figure 3.

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 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(Soybean, outer = ~ Year * Variety) # produces Figure 2
> plot(Soybean, outer = T) # (not shown) Variety = rows, Year=columns
>   # Figure 3 is produced by plot with a couple of trellis arguments
>   # to enhance the layout of the panels
> plot(Soybean, layout = c(8, 6), between = list(y = c(0, 0.5)))
```

An inner factor, such as Treatment in the PBG data, is used to determine which points within a panel are joined by lines.

We can see that incorporating a display formula with the data in the groupedData object makes it much easier to produce informative trellis displays. This general approach of using a data frame augmented by a formula that defines the roles of the variables is one of the powerful ideas in trellis. An alternative structure often used for longitudinal data is to represent the data as a matrix where the elements of the matrix are the responses, the rows (or columns) are determined by the groups, and the columns (or rows) are determined by the times. This can work when the data are balanced; that is, when all groups are measured at the same set of times. Any imbalance in the data makes this representation very awkward. The representation we have chosen does not encounter any problems with imbalance or with missing data. For example, the Soybean data are not balanced. The times at which the plots were sampled were different at different years. In fact, in 1988 there were different numbers of samples drawn from the plots for different genotypes.

```
> table(Soybean$Variety, Soybean$Year)
  1988 1989 1990
F   76   64   64
P   80   64   64
```

Another advantage of using a formula to describe the roles of the variables is that this information can be used within the model-fitting software to make the specification of the model easier. Even something as difficult as getting preliminary fits for a nonlinear mixed-effects model can be made as simple as

```
> Loblolly.lis <- nlsList(asymptotic, Loblolly)
```

when a self-starting nonlinear regression model has been defined as described in the next section.

4 Self-starting nonlinear regression models

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 asymptotic regression model (1.1) 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 nonlinear regression that is always problematic; determining starting values for the parameter estimates.

When the groups of data are similar, such as the Loblolly pine trees, 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 PBG 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 asymptotic 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 $\log(|y - \theta_1^{(0)}|)$ on t. The estimated slope is $-\theta_3^{(0)}$.

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.

An enhancement of the final step is to use the logarithm of θ_3 instead of θ_3 in the iterative algorithm so as to enforce the positivity constraint in (1.1).

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. This method is awkward and will be changed in nlme 3.0 but this need only concern those who will be writing the selfStart functions. The general user simply needs to know the name of the appropriate (univariate) model. It can then be used as shown in the previous section

```
> Loblolly.lis <- nlsList(asymptotic, Loblolly)
> coef(Loblolly.lis)
         Asym        R0       lrc
329  94.12877 -8.250685 -3.217588
327  94.94058 -7.757495 -3.229325
325  89.88361 -8.759223 -3.086195
307 110.69818 -8.169516 -3.390332
331 111.00287 -8.462608 -3.397574
311 109.98436 -8.558667 -3.362500
315 101.05602 -8.443651 -3.232822
321 127.13533 -7.679281 -3.575346
319 101.08717 -8.502381 -3.214016
301  95.66688 -9.078241 -3.116381
323  95.55627 -9.665035 -3.092266
309 113.51390 -7.595622 -3.352815
303 105.71443 -8.906502 -3.222905
305  99.17167 -9.916694 -3.084837
> plot(intervals(Loblolly.lis))
```

The last call produces a trellis plot of confidence intervals on the parameter estimates as shown in Figure 5

The separate nonlinear regression fits can also be used to describe a nonlinear mixed-effects model and implicitly to provide starting estimates for the parameters

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

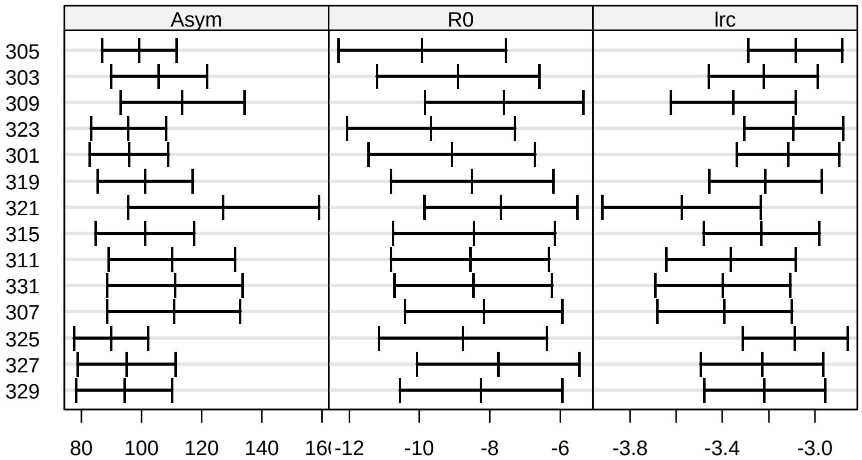

FIGURE 5. 95% confidence intervals on the parameter estimates of the asymptotic regression model fit separately to each tree in the Loblolly pine tree data

We can then fit alternate models say by reducing the number of random effects. Since R0, the response at time zero (written as θ_2 in (1.1)) seems to have the least variability between seed sources relative to the precision of the individual estimates, we eliminate it first and compare to the full model

```
> Loblolly.nlme1 <- nlme(Loblolly.nlme,
                         random = list(Asym ~ ., lrc ~ .))
> anova(Loblolly.nlme, Loblolly.nlme1)
               Model df    AIC    BIC  loglik    Test Lik.Ratio
 Loblolly.nlme     1 10 244.25 268.56 -112.13
Loblolly.nlme1     2  7 238.25 255.26 -112.12 1 vs. 2 0.0036802
               P-value
 Loblolly.nlme
Loblolly.nlme1 0.99994
```

The model building would continue from here.

5 Conclusions

Space limitations preclude showing a thorough examination of even one of these examples. Our purpose here is simply to show the power of trellis graphics for the exploration of longitudinal data and the ease with which such displays can be created. One of the keys to the success of the statistical modelling tools described in Chambers and Hastie (1992) is the concept of the data frame and the formula language. We have extended those here to a groupedData object that incorporates both the data as a frame and a formula giving important characteristics of the data; and the selfStart class of nonlinear regression model functions.

6 References

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

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.

Kung, F. H. (1986). Fitting logistic growth curve with predetermined carrying capacity, *ASA Proceedings of the Statistical Computing Section* pp. 340–343.

Ludbrook, J. (1994). Repeated measurements and multiple comparisons in cardiovascular research, *Cardiovascular Research* **28**: 303–311.

MathSoft (1996). *Version 3.4 for Unix Supplement*, Data Analysis Products, MathSoft Inc., Seattle, WA.

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*