Running head: VAMs in R

Fitting Value-Added Models in R

Harold C. Doran

Computer and Statistical Sciences Center

American Institutes for Research

J.R. Lockwood

The RAND Corporation

January 4, 2005

Corresponding Author:

Harold C. Doran

American Institutes for Research

1000 Thomas Jefferson Street, NW

Washington, DC 20007-3835

hdoran@air.org

This research was supported by Grant B7361 from the Carnegie Corporation of New York, by the National Science Foundation under Grant No. 99-86612, and by RAND. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Carnegie Corporation, the National Science Foundation or RAND.

Fitting Value-Added Models in R

Abstract

Value-added models of student achievement have received widespread attention in light of the current test-based accountability movement. These models use longitudinal growth modeling techniques to identify effective schools or teachers based upon the results of changes in student achievement test scores. Given their increasing popularity, this article demonstrates how to perform the data analysis necessary to fit a general value-added model using the **nlme** package available for the R statistics environment. We demonstrate techniques for inspecting the data prior to fitting the model, walk a practitioner through a sample analysis, and discuss general extensions commonly found across the literature that may be incorporated to enhance the basic model presented, including the estimation of multiple outcomes and teacher effects.

Keywords: hierarchical linear model; longitudinal data analysis; mixed-effects models; nlme; value-added model; accountability

Fitting Value-Added Models in R

Introduction

Test-based accountability systems have motivated researchers and policymakers to explore methods for analyzing student achievement data to evaluate the effectiveness of public schools. Most commonly, school effectiveness decisions have hinged upon levels and changes over time in aggregate achievement measures for successive cohorts of different students and ranks of schools based on such measures (Meyer, 1997).

However, a basic truism of learning implies that an *individual* student, not a student *group*, has increased in knowledge and skills during a specific period of time. As such, analytical methods concerned with student learning should reasonably reflect this basic principle and consider individual students as the unit of analysis with their growth trajectories employed as outcomes. Thus, when multiple waves of test score data are available, longitudinal analyses of student achievement are more likely to support inferences regarding school and teacher effects than cross-sectional methods of analysis (Willett, 1989).

Consequently, value-added models (VAM) of student achievement have arisen as methods for measuring educational impacts on student progress. Value-added models are a class of longitudinal growth techniques that attempt to identify the unique contributions of schools or teachers to student learning after controlling for preexisting differences among students (Ballou, Sanders, & Wright, 2004). Examples of VAMs in educational research include methods for evaluating school-effects (Raudenbush & Bryk, 1986) and the analysis of teacher, or residual classroom effects (McCaffrey, Lockwood, Koretz, Louis, & Hamilton, 2004; Rowan, Correnti, & Miller, 2002; Sanders, Saxton, & Horn, 1997). Increased enthusiasm for these models has resulted from the methodological and computational advances in the class of statistical models variously known as hierarchical linear models (Raudenbush & Bryk, 2002),

mixed-effects models (Pinheiro & Bates, 2000), and multilevel models (Goldstein, 1995) as well as the perceived benefit that the statistical mechanisms can adequately identify the causal impact of educators aside from non-school factors.

VAM analyses require longitudinal student-level test score data, and students must be linked over time to higher-level units such as schools and teachers. This permits researchers to explore questions such as “What proportion of the observed variance can be attributed to a school or teacher, how effective is an individual school or teacher at producing gains”, and, when data are available, “what characteristics or instructional practices are associated with effective schools or teachers?” Contingent upon the inferences desired and the structure and availability of the data, VAMs may range from simple univariate models, such as the gain score and covariate adjustment models, to more computationally demanding cross-classified models, where students change teachers and/or schools over time (Ballou et al., 2004; Lockwood, Doran, & McCaffrey, 2003; McCaffrey et al., 2004)

In Spring 2004, a special edition of this journal presented a range of articles devoted solely to VAMs. With the exception of Tekwe et al (2004) , which among other things presented implementation details of VAMs in SAS, the articles focused primarily on conceptual issues such as the development of the statistical model, including covariates as controls, the multidimensionality of test score scales, and the reasonableness of causal inferences. Given the growing demand to implement longitudinal analyses to support educational accountability systems, this article furthers the practical aspects of VAM by demonstrating how to implement certain VAMs with the **nlme** package available for the R statistics environment. Because this article assumes a didactic flavor, we also demonstrate preliminary diagnostic strategies provided in R to appropriately examine the data prior to specifying and fitting the growth model. Additionally, we discuss general extensions commonly found across the literature that may be used to enhance the basic VAM presented,

such as multivariate outcomes and methods for assessing teacher effects.

Throughout this article, font that takes `this format` represents commands, objects, and output in R. The R commands in the text follow "$>$", which is the R command prompt. If the command line cannot fit compactly onto one line of text, we use the (+) sign to indicate that same command line has been extended on a second line of text. However, the (+) sign is not part of the syntax and should be eliminated when actually implementing the code.

R Background

R is a language and environment for statistical computing and graphics (R Development Core Team, 2004). It is an open-source implementation of the S language and environment developed at Bell Laboratories by John Chambers and colleagues. It is a powerful and comprehensive statistical package capable of fitting complex models and producing a wide array of graphical displays. The base software and its contributed packages can be downloaded for free from the Comprehensive R Archive Network located at `http://lib.stat.cmu.edu/R/CRAN/`. Contributed packages, such as **nlme**, are extensions to the base program that are used to fit custom statistical models, such as mixed statistical models. R can be used on all modern operating systems such as Unix, Linux, Windows, and Macintosh.

Like SAS, R provides a well-developed programming language and a self-contained environment to produce diagnostic displays of the data and perform a wide range of statistical analyses, including models of the form considered here. The programming language makes R highly extensible and can be used to support the challenges faced by a practitioner when the desired estimation is not a part of the built in functionality. For example, Lockwood, Doran & McCaffrey (2003) show how to extend the capacity of **nlme** to fit models with crossed random effects using the programming tools available in R.

In R, results of calculations, including model estimation, can be stored in permanent

objects that can be manipulated subsequently and saved for later use in other R sessions. For example, one can fit a linear model and store the results of that operation in a variable, which includes among other things estimated coefficients, standard errors and p-values. It is then possible to reference or further manipulate those quantities later in the R session, regardless of what other operations have been performed in the interim. It is also possible to save the object and then load it for use in separate R sessions that might occur at different times or on different machines.

Unlike multilevel modeling programs with a graphical user interface (GUI), analyses in R are performed using a command prompt rather than point-and-click menus. In some respects, researchers familiar with a GUI may find this transition cumbersome. However, the command-driven environment also presents specific advantages over GUIs. In particular, repetition of analyses is trivial, as the code used can be simply stored and retrieved for subsequent use. Additionally, researchers familiar with LaTeX can combine R code within a TeX document to automatically generate reports and documents via the `Sweave` package. Lastly, it has been argued that command-driven environments require the user to more deeply understand and consider the statistical models used (Fox & Andersen, 2004). As we later demonstrate, R code is specified to closely resemble the statistical model specification.

While the array of specialized multilevel programs in wide-scale use present powerful options to users, R can be distinguished from others as it is designed to be a complete statistical environment. Techniques that may be required to fit multilevel models, such as data manipulation, missing data imputation, and exploratory data analysis can be easily performed entirely within the environment. Its open-source nature encourages the community of R users to share functions and contribute to its development, ensuring that built-in functionality stays in step with statistical procedures used across research disciplines.

One of the more convenient attributes of R is the rich environment for diagnostic visual

displays to assess model assumptions. The emphasis is such that most of the commonly used diagnostic plots can be simply called using the `plot` function. For example, using the `plot` command for a model object previously fit using least squares automatically generates five of the most common diagnostic displays for data examination. This emphasis is carried into the estimation of mixed linear models, providing the user with displays that can be used to assess the assumptions commonly made with respect to the variance components, within-group residuals, and the random effects.

R's facilities for linear mixed effects models, provided by the **nlme** package, are particularly relevant to this article. Jose Pinheiro and Douglas Bates originally developed the **nlme** package to fit both linear and non-linear mixed models, with a focus on nested random effects. Though technical support for the software does not exist, there is an active listserv dedicated to supporting R users and a book describing **nlme** and mixed models including estimation techniques and data applications (Pinheiro & Bates, 2000).

Because R and the **nlme** package are flexible, as are other environments for multilevel modeling, it is difficult to accurately characterize classes of models that can and cannot be fit with the packages. Creative programming often can be used to extend the power of other multilevel programs beyond their optimal design. For example, Goldstein (1995) shows how to structure a data matrix such that models with crossed random effects can be fit even when the computational algorithms are not optimized for such situations. Kurdek, du Toit & du Toit (2004) demonstrate how to model more complex covariance structures by utilizing dummy variables. For these reasons, we avoid making claims about the functionality of one program and its benefits over another. The web sites `http://www.ats.ucla.edu/stat/` and `http://multilevel.ioe.ac.uk/softrev/index.html` provide valuable resources for comparing and contrasting R with other multilevel software packages.

Data Requirements and Notation

Fitting a value-added model requires a minimum of two test scores per student that can be linked to higher level units such as individual teachers or schools over time. Most commonly, VAMs are applied to standardized test score data, though they may be applied to state developed standards-referenced or other criterion-referenced tests when they too have been properly constructed and scaled. Throughout this article, we assume that test scores represent a continuous, developmental scale such that gain scores (i.e., current score minus past score) represent increased academic achievement along a well-established, vertically equated scale (Peterson, Kolen, & Hoover, 1989).

Because hierarchical linear models, mixed models, and multilevel models have been shown to take the same general form (Ferron, 1997), we use the terms interchangeably and write the model in combined form, $\boldsymbol{Y_i} = \boldsymbol{X_i}\boldsymbol{\beta} + \boldsymbol{Z_i}\boldsymbol{\Theta_i} + \boldsymbol{\epsilon_i}$ (Pinheiro & Bates, 2000), where $\boldsymbol{Y_i}$ is an n_i-dimensional response vector of test score data for a single subject (e.g., reading or math), $\boldsymbol{X_i}$ is an n_i x p design matrix, β is the p-dimensional vector of fixed effects, $\boldsymbol{Z_i}$ is the n_i x q design matrix for the q-dimensional vector of random effects $\boldsymbol{\Theta}_i$, and $\boldsymbol{\epsilon_i}$ is the n_i-dimensional within-group error term. We refer to the fixed effects, $\boldsymbol{X_i}\boldsymbol{\beta}$, as the structural portion of the model and the random effects, $\boldsymbol{Z_i}\boldsymbol{\Theta_i} + \boldsymbol{\epsilon_i}$, as the stochastic portion of the model.

The general form of the data considered within this article is repeated observations nested within students nested within schools. As such, "level 1" refers to the vector of response variables, "level 2" refers to the next higher unit, students, and "level 3" refers to schools. The dataset used in this article is a subset of the US Sustaining Effects Study consisting of 1,721 students nested within 60 schools, which is freely distributed with the HLM software package (Bryk, Raudenbush, & Congdon, 1996) as three individual datafiles. The individual files, `EG1, EG2,` and `EG3` contain the Level 1, Level 2, and Level 3 information, respectively. The response variable is `math`, a test score in the IRT scale score metric; `year` is a time-varying covariate measured at up to six intervals taking values of -2.5, -1.5, -0.5, 0.5, 1.5, and 2.5;

`female` is a dummy code for gender (female=1); and `size` is the number of students enrolled in the school [1]. The variables `childid` and `schoolid` are the level 2 and level 3 identification variables, respectively.

Fitting the longitudinal models discussed in this paper requires that the data be in a format commonly referred to as the "long" or "person-period" structure. In this format, repeated observations on the same student are stored in consecutive, temporally ordered rows as demonstrated in Table 1.

Table 1: First two students in the person.period

schoolid	childid	year	grade	math	female	size
2020	273026452	.5	2	1.146	1	380
2020	273026452	1.5	3	1.134	1	380
2020	273026452	2.5	4	2.3	1	380
2020	273030991	-1.5	0	-1.303	1	380
2020	273030991	-.5	1	.439	1	380
2020	273030991	.5	2	2.43	1	380
2020	273030991	1.5	3	2.25	1	380
2020	273030991	2.5	4	3.87	1	380
...						

Creating the Full Data Matrix

In many programs specialized to fit only mixed models such as HLM, individual data files are organized and examined in a separate software package. Each individual file is then imported into the specialized program and joined using identification or grouping variables. Because R requires the full data matrix, we demonstrate how individual files organized in other software programs can be joined in R to create the single data matrix required using the `merge` function.

To begin, all three data files (`EG1, EG2, EG3`) must reside in the R workspace. These particular ASCII files would be imported into the R workspace via:

```
EG1 <- read.table("/path/to/your/file.txt", header=TRUE, sep="  ",
+ na.strings="NA", strip.white=TRUE)
```

[1] Other covariates are included in the actual dataset, however, for illustrative purposes, we include only two.

Because `SCHOOLID` is used only to merge the `EG2` and `EG3` files, it is a necessary first step to remove this variable from the `EG1` file.

```
> EG1$SCHOOLID<-NULL
> tmp1<- merge(EG1, EG2, by="CHILDID", all.x=TRUE)
> egsingle<- merge(tmp1, EG3, by="SCHOOLID", all.x=TRUE)
```

In Step 2, `EG1` is merged with `EG2` into a new object called `tmp1` using `CHILDID`. In Step 3, `tmp1` is merged with `EG3` using `SCHOOLID`, resulting in the full data matrix required, `egsingle`. Because `EG1` was temporally ordered to begin with, the data matrix is in the person-period format. The `reshape` command can be used when data must be restructured from person-level to the person-period format required.

Before proceeding with the analysis, it is convenient to convert all variable names in the dataset to lowercase. Second, it is necessary to convert all nominal and identification variables into factors and to load the **nlme** package. These steps are accomplished as follows:

```
> names(egsingle)<-tolower(names(egsingle))
> egsingle$female<-factor(egsingle$female)
> egsingle$childid<-factor(egsingle$childid)
> egsingle$schoolid<-factor(egsingle$schoolid)
> library(nlme)
```

Preliminary Diagnostics

With the full data matrix now created, we proceed with two preliminary diagnostic strategies commonly used to examine the data prior to fitting a growth model. First, it is a useful strategy to examine the shape of the individual growth trajectories to assess whether a linear, quadratic, or higher order polynomial should be fit. This can be accomplished by fitting separate OLS regressions for each student in the `egsingle` dataset with the observed data plotted around the regression line. Although the data are likely to violate OLS assumptions (e.g., independent errors), it provides an adequate exploratory tool to assess functional form (J.

Singer & Willet, 2003). Because the dataset contains 1,721 students, we select a random sample of 50 students in the following manner:

```
> egsingle<-groupedData(math~year|schoolid/childid, data=egsingle)
> samp <- sample(levels(egsingle$childid), 50)
> level2.subgroup <- subset(egsingle, childid %in% samp)
```

The first command creates a groupedData object, an object that contains the data and defines the nesting structure given the grouping variables. Steps 2 and 3 randomly sample 50 students from `egsingle` and stores their information in a new object called `level2.subgroup`. We use the `lmList()` command to fit a separate OLS regression line for each student using the following code:

```
> level2<-lmList(math~year|childid, data=level2.subgroup)
> plot(augPred(level2))
```

The first command creates a new object, `level2`, and specifies `math` as the response variable and `year` as the only fixed effect in addition to an intercept, which is included in the model by default. In R, the tilde `(~)` is read "is modeled as". In other words, `math` is modeled as a function of `year`, with an intercept included by default. The portion of the code following the vertical bar, `|childid`, specifies that OLS regressions should be fit at the student level. The segment `data=level2.subgroup` tells R to use the dataframe consisting of the 50 randomly sampled students rather the full `egsingle` dataframe.

The second command produces Figure 1, an OLS regression line for each of the 50 randomly sampled students. The `augPred` command augments the OLS lines with the observed data points on each measurement occasion. Assessment of the functional form based upon these visual displays suggests that a straight line will adequately represent the shape of the population growth trajectories.

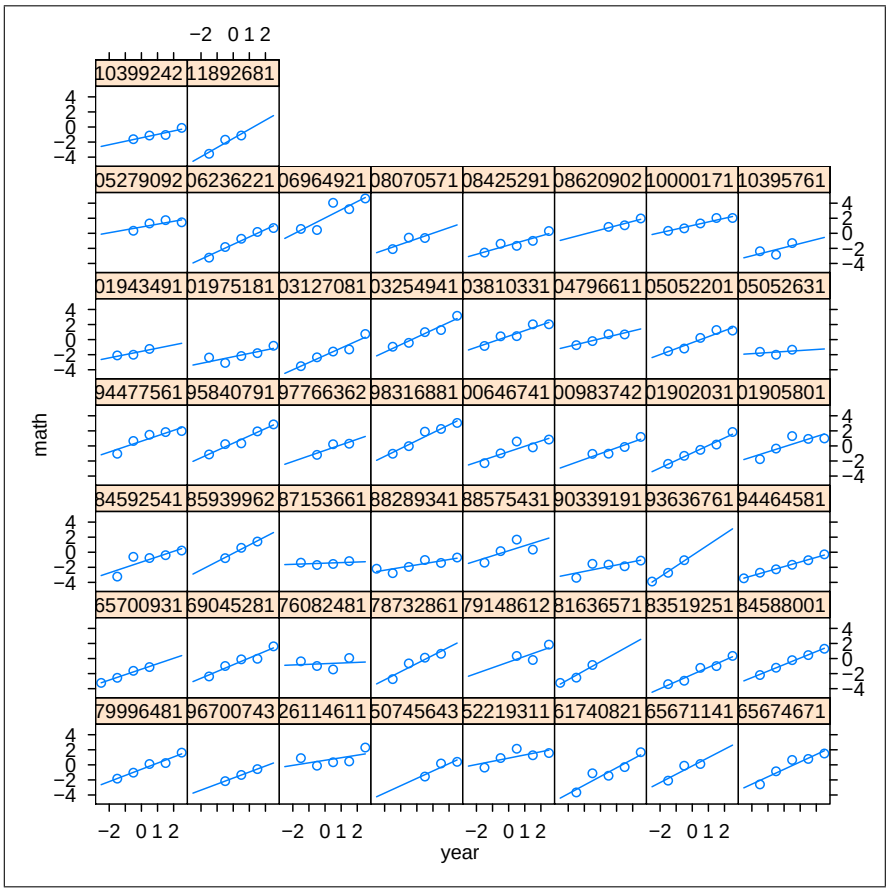

Figure 1: OLS regression lines, along with observed data, for 50 randomly sampled students

Examining Intercepts and Slopes

A second necessary step when building mixed linear models is to assess which parameters will be allowed to vary randomly at the different levels of nesting. It is often reasonable to assume that students start in different locations on the ability scale and grow at different rates, and that these unique student-level parameters could depend on the school, suggesting that the intercepts and slopes may be allowed to vary randomly. This theoretical assumption is formally examined in multilevel software packages, such as SAS Proc Mixed and HLM, after fitting an unconditional growth model and examining the p-value associated with the chi-square statistic for each intercept and slope variance component. Statistically

significant variance components are used to determine whether the researcher should retain or "fix" random intercepts and slopes at the each level of the hierarchy.

In **nlme**, chi-square statistics for the variance components are not produced. Instead, intercept and slope variability may be informally examined *a priori*, relying upon visual displays, and standard empirical tests are available via the `anova()` command, explored in detail in a later section. Producing the displays used to examine variability are created using a combination of the `lmList()` and `intervals()` commands.

```
> level3<-lmList(math~year|schoolid, data=egsingle)
> plot(intervals(level3))
```

The first command produces an OLS line for each of the 60 schools in the dataset. The second command produces Figure 2, a visual display of the point estimate of the OLS intercept and slope for each of the 60 schools surrounded by a 95 percent confidence interval.

The large number of confidence intervals for the school-level intercepts that do not overlap, combined with the fact that many of the intervals support substantially different values, suggests that random intercepts are appropriate at the school level. Although there is less variability evident among the school-level slopes, there is still a relatively large number of non-overlapping intervals, suggesting that random school-level slopes may be appropriate as well.

We now repeat this process at Level 2, the student level. Recall that we have already sampled 50 students and have fit individual OLS lines. Therefore, we examine these 50 intercepts and slopes using the following code:

```
> plot(intervals(level2))
```

Consistent with the school-level plots, the large number of non-overlapping confidence intervals in Figure (3) suggests sufficient heterogeneity among student-level intercepts to

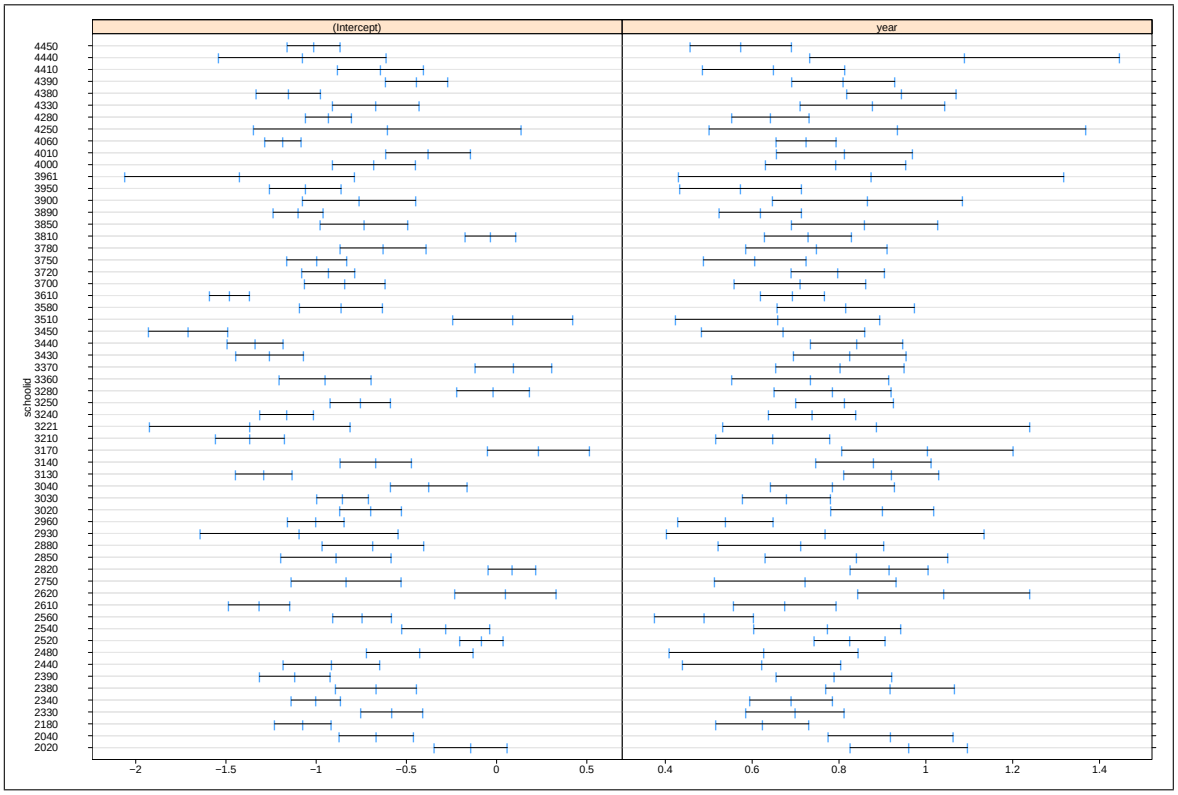

Figure 2: School-level OLS intercepts and slopes, along with 95% confidence intervals

consider random intercepts. Again, there is less variability across student-level slopes than the intercepts, but random intercepts appear to be worth consideration.

The preliminary empirical evidence from the visual displays, combined with the substantive theoretical evidence, provides sufficient support to proceed with random intercepts and slopes at each level of nesting. However, in situations where the visual displays indicate homogeneous intercepts and slopes, the analysts may choose to simplify the parameterization of the fitted model. In a later section, we demonstrate how alternative specifications may be fit and tested against a baseline model.

Fitting the Unconditional Value-Added Model

With the two preliminary diagnostics complete, we proceed to fit the unconditional VAM. This model takes the following form:

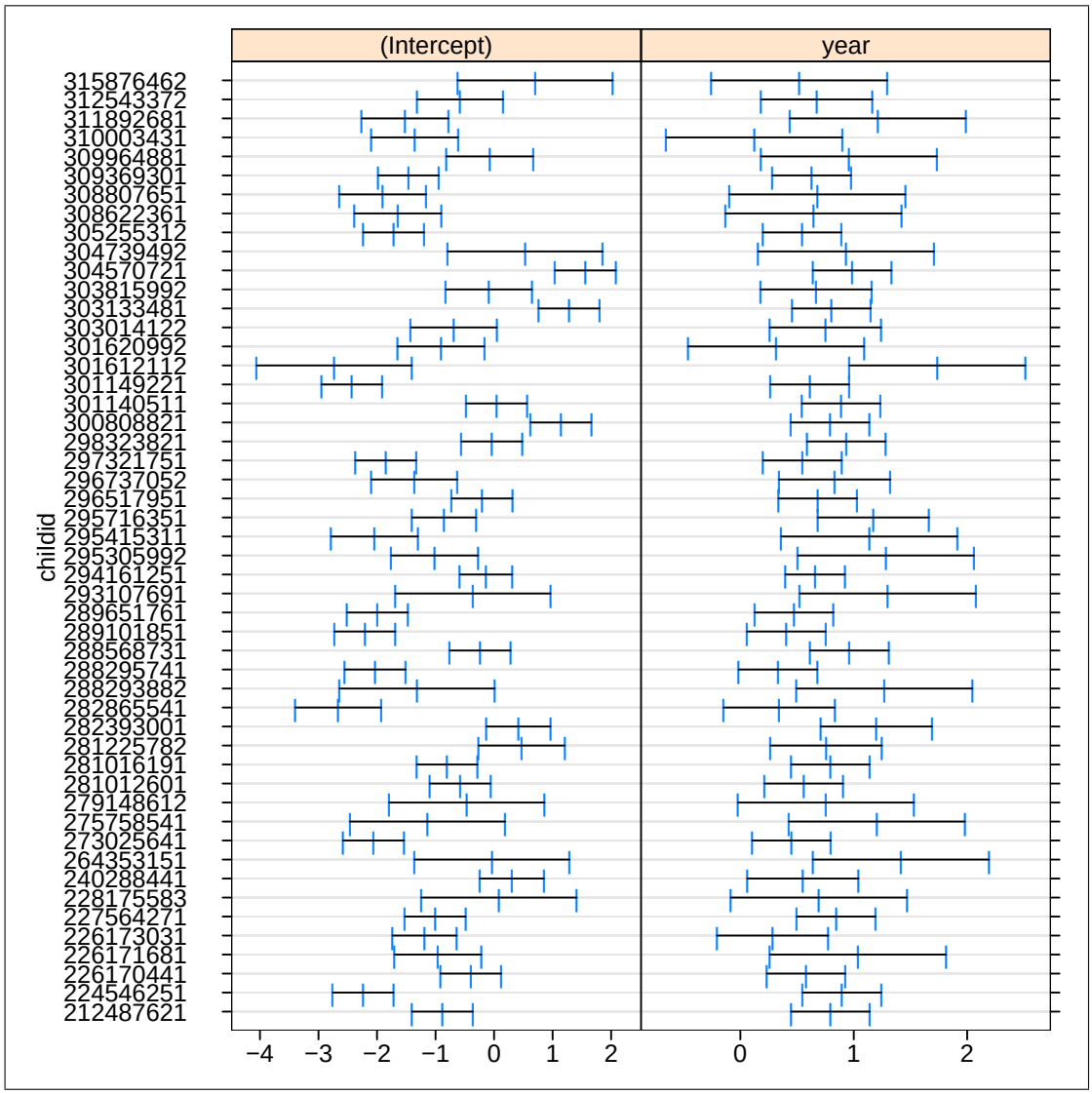

Figure 3: Student-level OLS intercepts and slopes, along with 95% confidence intervals

$$Y_{ti} = [\beta_0 + \beta_1(year)] + [\theta_{0j(i)} + \theta_{1j(i)}(year) + \delta_{0i} + \delta_{1i}(year) + \epsilon_{ti}] \quad (1)$$

where t indexes time $(t = 0, \ldots, T)$, i indexes students, and j indexes schools. The notation "$j(i)$" is used to mean the index j for the school attended by student i. The within-group errors are assumed to be independent and identically distributed, $\epsilon_{ti} \sim \mathcal{N}(0, \sigma^2)$ with an unstructured 2 x 2 covariance matrix for the school and student random effects $\boldsymbol{\theta_j}$=$(\theta_{0j}, \theta_{1j}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$,

and $\boldsymbol{\delta_i}$=$(\delta_{0i}, \delta_{1i}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})$. Consistent with Singer (1998), we separate the fixed effects and random effects using brackets.

This is the classic random coefficient growth model with intercepts and slopes as outcomes. As indicated by Equation (1) the model contains an intercept (β_0) and a main effect for `year`, (β_1), as well as random intercepts and slopes at each level of nesting.

This model is fit and stored in the object `unconditional.lme` using the following code at the R command prompt:

```
> unconditional.lme<-lme(math~year, random=~year|schoolid/childid,
data=egsingle)
```

The structural portion of the model is specified in the first segment of the code, `math~ year`, and the stochastic portion of the model is specified following the random statement, `random=~ year`. The nesting structure is equivalent to fitting random effects for the schools and random effects for the students within schools. Therefore, this is specified using the grouping variables as `|schoolid/childid` (Lockwood et al., 2003; Pinheiro & Bates, 2000). The segment of the command, `data=egsingle`, simply tells R which dataset to use.

With the growth model now fit, we proceed to examine the model output using `summary()`. The command `summary(unconditional.lme)` produces the output:

```
Linear mixed-effects model fit by REML
 Data: egsingle
       AIC      BIC   logLik
  16354.74 16416.71 -8168.37

Random effects:
 Formula: ~year | schoolid
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 0.4107722 (Intr)
year        0.1061257 0.398

Formula: ~year | childid %in% schoolid
Structure: General positive-definite, Log-Cholesky parametrization
```

```
            StdDev    Corr
(Intercept) 0.8003193 (Intr)
year        0.1062362 0.55
Residual    0.5489763

Fixed effects: math ~ year
                 Value  Std.Error   DF   t-value p-value
 Intercept) -0.7791464 0.05832736 5508 -13.35816  <.0001
year         0.7631248 0.01539913 5508  49.55635  <.0001
 Correlation:
     (Intr)
year 0.356

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.20524032 -0.56405150 -0.03713502  0.53080683  5.25903901

Number of Observations: 7230
Number of Groups:
             schoolid childid %in% schoolid
                   60                  1721
```

Interpreting Model Output

Goodness of Fit Statistics

The model output first indicates that it was fit using Restricted Maximum Likelihood (REML). However, full maximum likelihood estimation is possible when `method=''ML''` is included in the `lme` call. Typical goodness of fit statistics including Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and the value of the log-likelihood (logLik) at convergence are produced. In a later section, we demonstrate how to invoke the likelihood ratio test (LRT) where appropriate.

Variance Components

The next section of the output describes the variability of the intercepts and slopes at the school and student levels. First note that variance components are reported as standard

deviations. Because we have fit a three level model, there are three levels of random effects: the school level random effects (`~ year | schoolid`), the student level random effects (`~ year |childid %in% schoolid`), as well as the within-group residual, (`Residual`).

The school level random effects are first presented. Note that the school level intercept standard deviation (ψ_{00}) is .41, approximately four times larger than the school level standard deviation for the slope (ψ_{11}), consistent with the visual display in Figure 1. The correlation between the school level random effects (ψ_{01}) is .40.

The variance components associated with the student level random effects are also consistent with the visual display in Figure 2. Note that the intercept standard deviation (ω_{00}) is approximately .80, eight times larger than the standard deviation for the slope, (ω_{11}). The correlation between student level random effects is .55, another indication of a positive relationship between status and growth. Last, the within-group residual standard deviation is approximately .55.

Using the `intervals()` command, we can further examine the point estimates of the square roots of the variance components surrounded by approximate 95 percent confidence intervals in the following manner:

```
> intervals(unconditional.lme, which="var")

Approximate 95% confidence intervals

 Random Effects:
  Level: schoolid
                           lower      est.     upper
sd((Intercept))       0.33173226 0.4107722 0.5086445
sd(year)              0.08440695 0.1061257 0.1334328
cor((Intercept),year) 0.10241163 0.3979439 0.6289247
  Level: childid
                           lower      est.     upper
sd((Intercept))       0.76989095 0.8003193 0.8319504
sd(year)              0.08995348 0.1062362 0.1254664
cor((Intercept),year) 0.41013193 0.5500230 0.6646282
```

```
 Within-group standard error:
    lower      est.     upper
0.5373058 0.5489763 0.5609003
```

The small intervals suggest relatively precise point estimates of the intercept and slope variance components at the school and student levels. The school level correlation, however, has a rather wide range, but is bounded away from zero. Other commands that may be used to examine the variance components of a fitted model are `getVarCov`, `VarCorr` and `extract.lme.cov`. These functions extract features of the estimated variance-covariance structure of the random effects at different levels.

Fixed Effects

The next section of the output presents the fixed effects; in this case, the intercept and the main effect for `year`. The hypothesis test is the familiar t-test, which is the restricted maximum likelihood point estimate divided by its estimated asymptotic standard error. Given the coding scheme used for `year` (i.e., the centering), the intercept, -.779, represents the average status of a student halfway through Grade 3. The main effect for `year`, .76, represents the average growth rate over the duration of the study. The fixed portion of the output also shows the correlation between the estimated parameters, which is .356.

Descriptive Information

The model output also includes a five-point summary of the within-group residuals as well as the number of unique observations at each nested level. For example, this dataset contains 1,721 students nested within 60 schools with a total of 7,230 observations.

Adding Covariates as Fixed Effects

After fitting a suitable unconditional growth model, one may choose to add covariates to explain variability among the slopes and intercepts. This is easily accomplished by extending

the linear model presented in Equation (1). Suppose we would like to include `female` as an additional main effect to account for variability among student-level intercepts, a school-level variable, `size`, to account for variability among the school level intercepts, and a `size` by `year` interaction to account for variability among school level slopes. The conditional model would be specified as:

$$\begin{aligned} Y_{ti} &= [\beta_0 + \beta_1(year) + \beta_2(female) + \beta_3(size) + \beta_4(size)(year)] \\ &\quad + [\theta_{0j(i)} + \theta_{1j(i)}(year) + \delta_{0i} + \delta_{1i}(year) + \epsilon_{ti}] \end{aligned} \tag{2}$$

The code used to fit the model is very similar to the code previously used to fit the unconditional growth model. However, the covariates are now included as additional fixed effects in the first segment of the command line:

```
> covariate.lme<-lme(fixed=math~year*size+female,
+ random=~year|schoolid/childid, data=egsingle)
```

The star (*) operator in the fixed segment fits main effects for `year` and `size` as well as a cross-level interaction between `size` and `year`. When the semicolon operator (:) is used, R fits only an interaction between these variables. Fitting the conditional model specified in Equation (2) produces the following output:

```
> summary(covariate.lme)

Linear mixed-effects model fit by REML
 Data: egsingle.new
       AIC      BIC    logLik
  16394.79 16477.42 -8185.397

Random effects:
 Formula: ~year | schoolid
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 0.4021677 (Intr)
```

```
year        0.1056796 0.36

 Formula: ~year | childid %in% schoolid
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 0.8005652 (Intr)
year        0.1062595 0.55
Residual    0.5489249

Fixed effects: math ~ year * size + female
                 Value  Std.Error   DF   t-value p-value
(Intercept) -0.5687563 0.13494630 5507 -4.214686  <.0001
year         0.8127350 0.03627537 5507 22.404598  <.0001
size        -0.0003040 0.00018264   58 -1.664258  0.1015
female      -0.0195409 0.04102749 1660 -0.476288  0.6339
year:size   -0.0000746 0.00004957 5507 -1.505084  0.1324
 Correlation:
          (Intr) year   size   female
year       0.321
size      -0.892 -0.292
female    -0.142  0.000 -0.011
year:size -0.289 -0.906  0.321 -0.001

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.20133699 -0.56341683 -0.03567821  0.52987453  5.25330762

Number of Observations: 7230
Number of Groups:
             schoolid childid %in% schoolid
                   60                  1721
```

Note that the variance components for all random effects including the within-group residual are virtually identical to the unconditional growth model. The two additional main effects, `female`, `size`, and the `year:size` cross-level interaction are all non-significant as indicated by the sizes of their p-values.

Covariance Structures of Random Effects

In both of the models previously fit we assumed $\epsilon_{ti} \sim \mathcal{N}(0, \sigma^2)$ and a general 2 x 2

covariance structure for the school and student random effects, $\boldsymbol{\theta_j} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ and $\boldsymbol{\delta_i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega})$. The **nlme** package includes a host of graphical displays to examine the distributional assumptions of the within-group errors and the random effects, as well as modeling facilities for fitting more complex structures and methods for making empirical comparisons of alternative models. We illustrate some of these techniques by exploring less restrictive covariance structures for both the residuals and random effects.

First, it is possible that the residual variance is not constant over time, contrary to the assumptions of the previous models. Richer models for the residual variance structure are specified via the `weights` argument of `lme`. These models are implemented by the so-called "varFunc" classes of **nlme**; we focus on `varIdent` which allows different residual variance for each level of a stratifying variable. The following code fits a revised model where the residual variance is allowed to vary by `year`:

```
> hetVar.lme<-update(unconditional.lme, weights=varIdent(form=~1|year))
```

The `update()` command revises a previously stored model object without having to completely reenter code. The summary output for this revised model would mirror the output of the previously fitted models with one exception. The `weights` command adds a small additional section of output labeled `Parameter Estimates` describing the ratio of residual variances to the within-group error for each measurement occasion.

```
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | year
 Parameter estimates:
      0.5       1.5       2.5      -1.5      -0.5      -2.5
1.0000000 0.7333049 0.6409035 1.0061606 0.9172631 0.9599863
```

These numbers represent the factors by which the estimated residual standard deviation must be multiplied to recover the time-specific standard deviations. In the output above, the residual standard deviation at measurement occasion two (`1.5`) is estimated to be 73% as

large as that for occasion one (`0.5`). The standard deviation at occasion one is equal to the `Residual` in the summary output.

Empirically comparing `unconditional.lme` and `hetVar.lme` will help to determine which specification is a better representation of the true variance structure. Because both models were fit using REML and they only differ in their stochastic portion, it is appropriate to invoke the likelihood ratio test (LRT) to make the comparison. This is accomplished using the `anova()` command as follows:

```
> anova(unconditional.lme,hetVar.lme)
```

```
                  Model df      AIC      BIC    logLik   Test  L.Ratio p-value
unconditional.lme     1  9 16354.74 16416.71 -8168.370
hetVar.lme            2 14 16178.76 16275.16 -8075.382 1 vs 2 185.9767  <.0001
```

The larger, (i.e., closer to zero) loglikelihood, and statistically significant LRT suggests that `hetVar.lme` is a much better fit than `unconditional.lme`.

In addition to heteroskedasticity it is possible that there is additional serial correlation in the residuals from the same student even conditional on the random effects currently in the model. Richer models for the correlation of the residuals are specified via the `correlation` argument of `lme`. To empirically examine the independence assumption we update `hetVar.lme` to allow for the residuals to be correlated over time using a continuous AR(1) structure.

```
> CAR1model.lme<-update(hetVar.lme,
+ correlation=corCAR1(form=~year|schoolid/childid))
```

Judgement between fitted models is again made using the LRT as follows:

```
> anova(CAR1model.lme, hetVar.lme)
```

```
              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
CAR1model.lme     1 15 16110.77 16214.05 -8040.383
hetVar.lme        2 14 16178.76 16275.16 -8075.382 1 vs 2 69.99645  <.0001
```

The statistically significant LRT suggests that `CAR1model.lme` is a much better fit than `hetVar.lme`. Therefore we retain the model with the heteroscedastic autoregressive structure.

It is also possible to explore the assumptions of the random effects at different levels of nesting using the tools provided by the `pdMat` class. Though we do not proceed with the full diagnostics here, the code needed to revise the school level random effects with a diagonal covariance matrix is provided. This simplifying assumption reduces the number of variance-covariance parameters for the school random effects to be estimated from three (i.e., two variances and one covariance in the general structure) to two, where the covariance is assumed to be zero. Using similar empirical techniques (e.g., LRT), one may test whether this constraint results in a significantly better fit.

```
> update(CAR1model.lme, random=list(schoolid=pdDiag(~year),
+ childid=pdSymm(~year)))
```

In the code above, the `list()` command is used to specify that the the random effects differ across the hierarchy. Here, `list()` specifies a diagonal structure for the covariance of the school level random effects and retains the general covariance structure at the child level.

Obtaining Model Coefficients

Because VAMs are designed to simultaneously estimate growth rates for cases at different levels of nesting, it is of substantive interest to extract the model coefficients at different levels as well as the random effects. For example, in a value-added study where individual student growth is the object of inference, one may want to know the estimated true growth rate of individual i (Doran, 2004). Given the parameterization of Equation (1), this is computed as the sum of the main effect for `year` and the random effects at the school and student level:

$$growth_i = \beta_1 + \theta_{1j(i)} + \delta_{1i} \tag{3}$$

These values are easily obtained using the `coef()` command as follows:

```
>coef(CAR1model.lme, level=2)[1:5,]
```

This will produce a data matrix with the intercept and slope for the first five students in the dataset.

```
                        (Intercept)     year
2020       273026452     0.01929576 0.9726919
2020       273030991     0.86198171 1.0449975
2020       273059461     0.39708544 1.0052197
2020       278058841     1.01292465 1.0521309
2020       292017571     1.02811317 1.0557499
```

The data indicate that student `273026452` in school `2020` has a math score of .019 halfway through Grade 3 and is growing at an average annual rate of .97. If `level=1` in the command line, we obtain the typical yearly growth for school j (i.e., $\beta_1 + \theta_{1j}$).

To extract the random effects for the school level intercept (θ_{0j}) and slope (θ_{1j}) only we use the `ranef()` command:

```
>ranef(CAR1model.lme, level=1)
             (Intercept)        year
2020           0.5835523  0.18819638
2040           0.1002116  0.10835088
2180          -0.2408244 -0.11637441
2330           0.1783028 -0.07375527
2340          -0.2171441 -0.03866450
```

The random effects are generally of substantial interest in a value-added study as they reflect the "school effect", or the unique deviation from the typical growth trajectory. That is, the "effect" of attending school j is to adjust the growth rate from β_1 to $(\beta_1 + \theta_{1j})$. For example, the average growth rate for students in school `2020` is (.76 + .19)=.95 scale score units, where .76 is the value of the main effect for `year` (β_1) from `CAR1model.lme`.

Extensions to the Previously Fitted Models

Fixing Random Effects

It may be the case that a more parsimonious model may be found via modifications to the random effects at different levels of nesting. For example, one may consider constraining the student level random slopes to be equal, but allow for the school random slopes to vary. In such cases the model would be represented as:

$$Y_{ti} = [\beta_0 + \beta_1(year)] + [\theta_{0j(i)} + \theta_{1j(i)}(year) + \delta_{0i} + \epsilon_{ti}] \quad (4)$$

To fix the student-level random slopes in the model `CAR1model.lme`, the following code is used:

```
CAR2model.lme<-update(CAR1model.lme,random=list(schoolid=~year,childid=~1))
```

As before, the `list()` command is used to specify a different structure for the variance components at the school and student level. We again use the `anova()` command to compare models:

```
> anova(CAR2model.lme,CAR1model.lme)
              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
CAR2model.lme     1 13 16251.57 16341.08 -8112.783
CAR1model.lme     2 15 16110.77 16214.05 -8040.383 1 vs 2 144.7996  <.0001
```

The statistically significant LRT indicates that `CAR1model.lme` is a better fit than the alternative.

Multivariate Outcomes

It is often the case in education that more than one response variable is collected at each time period. As such, one may be interested in modeling the correlation across time within student as well as the correlation within time across subjects (MacCallum & Kim, 2000; Thum, 1997). Exploring the interrelationship between correlated response variables provides numerous well-known advantages and may improve efficiency of fixed effects estimates.

To demonstrate how this would be accomplished in **nlme**, we construct a new dataframe in Table 2, showing that the data must be structured somewhat differently than the model with a single outcome.

Table 2: Sample multivariate dataset for two students

childid	score	read	math	read.time	math.time
1	Y_{r11}	1	0	-1	0
1	Y_{r21}	1	0	0	0
1	Y_{r31}	1	0	1	0
1	Y_{m11}	0	1	0	-1
1	Y_{m21}	0	1	0	0
1	Y_{m31}	0	1	0	1
2	Y_{r12}	1	0	-1	0
2	Y_{r22}	1	0	0	0
2	Y_{r32}	1	0	1	0
2	Y_{m12}	0	1	0	-1
2	Y_{m22}	0	1	0	0
2	Y_{m32}	0	1	0	1
			...		

As can be seen, the reading (Y_{rti}) and math scores (Y_{mti}) are stacked in a column vector with a dummy code for reading and math flagging each respective variable. A variable for $time$ has also been created for each measure separately. The trick is to structure the outcome variables as if only one variable were to be analyzed and create a set of dummy codes to "flag" the outcomes as necessary.

The multivariate growth model for student i in school j at time t on subject s can be parameterized as:

$$Y_{sti} = [\mu_{0s} + \beta_{0s}(time)] + [\theta_{0sj(i)} + \theta_{1sj(i)}(time) + \delta_{0si} + \delta_{1si}(time) + \epsilon_{sti}] \tag{5}$$

Fitting Equation (5) for two outcome variables (reading and math), requires the estimation of four fixed effects and three covariance matrices. Specifically, the model includes an intercept and slope for each of the two variables (μ_r, μ_m, β_r, β_m), a four-dimensional vector of random effects at the school level (θ_{0rj}, θ_{1rj}, θ_{0mj}, θ_{1mj}) with covariance matrix $\mathbf{\Psi}$, a

four-dimensional vector of random effects at the student level (δ_{0ri}, δ_{1ri}, δ_{0mi}, δ_{1mi}) with covariance matrix $\mathbf{\Omega}$, and a covariance matrix for the within-group residuals, $\mathbf{V}$.

In this model, proper specification of the residuals and random effects takes on a rather significant level of complexity. For example, the covariance matrices $\mathbf{\Psi}$, $\mathbf{\Omega}$, and $\mathbf{V}$ might be specified as diagonal, block-diagonal, or unstructured. Choosing a covariance structure that allows the random effects to covary within and across outcome variables provides information that cannot otherwise be obtained by fitting separate models for each outcome. For example, the correlation across reading and math slopes describes the relationship between rates of linear change across different subjects. Additionally, the within-group covariance matrix $\mathbf{V}$ may be structured such that it depends on both time and subject.

For purposes of presentation, we proceed with a simple structure for the within-group residuals and the random effects. However, the prior sections illustrate the various `varFunc`, `corStruct`, and `pdMat` classes available in **nlme**. Using the techniques previously described may be used to model a more complex covariance structure.

Fitting the model requires that the overall intercept be removed `(-1)` in the structural and stochastic components. This code fits the model specified in Equation (5), including the separate residual variances per outcome variable using the `weights` argument and includes an unstructured covariance matrix to the school and student random effects.

```
> multoutcome.lme<-lme(score~-1+math+read+math.time+read.time,
+ random=~-1+math+read+math.time+read.time|schoolid/childid,
+ data=sample.dat, weights=varIdent(form=~1|math))
```

As a result, the main effect for `math` and for `read` can be directly interpreted as the subject-specific mean outcomes. Additionally, a unique growth slope is constructed for the two variables, `read.time` and `math.time`.

Models with Crossed Random Effects

In all of the models previously presented, it was assumed that students were properly nested within only one higher level unit. However, it is often the case in longitudinal data analysis that students change classrooms and/or schools over time. This results in a more complex data structure referred to as cross-classified, and models for these data may require crossed random effects to properly characterize the covariance structure. The two most widely known VAMs that explicitly model this source of variation include the Tennessee Value Added Assessment System (TVAAS) (Sanders et al., 1997) and the Raudenbush and Bryk cross-classified model (2002). In both cases, yearly teacher effects are allowed to cumulate undiminished over time such that the current score for student i reflects the sum of the current teacher and all prior teachers experienced by this student.

Table 3 illustrates a sample data structure where three students cross teachers over time. Note that all students began with Teacher A, but moved into different classrooms over time.

Table 3: Conceptual Crossing Structure

Student	Teacher A	Teacher B	Teacher C	Teacher D	Teacher E
Student 1	Y_1	Y_2		Y_3	
Student 2	Y_1		Y_2	Y_3	
Student 3	Y_1	Y_2			Y_3

Consequently, constructing the data matrix for these three students would take the form shown in Table 4. Because the teacher effects are allowed to cumulate, the binary variable constructed to indicate whether the student had a teacher or not is flagged for all previous teachers even as students progress through the school over time.

It is beyond the scope of this article to fully discuss the complexity of these models; the article by Lockwood et al (2003) provides a detailed account of how to fit these models in **nlme**. It is important to note that that cross-classified random effects models are notoriously computationally challenging, and the current facilities of the **nlme** package can fit such models to only modestly sized data sets.

Table 4: Sample Crossing Data Structure

student	score	time	A	B	C	D	E
1	Y_1	0	1	0	0	0	0
1	Y_2	1	1	1	0	0	0
1	Y_3	2	1	1	0	1	0
2	Y_1	0	1	0	0	0	0
2	Y_2	1	1	0	1	0	0
2	Y_3	2	1	0	1	1	0
3	Y_1	0	1	0	0	0	0
3	Y_2	1	1	1	0	0	0
3	Y_3	2	1	1	0	0	1

Miscellaneous Common Tools

Fitting mixed models via the **nlme** package permits for significant flexibility within the modeling functions. For example, two of the most common strategies when fitting mixed models involves extending the linear model to include higher order polynomials and centering the time variable. In **nlme** it is not necessary to create additional variables, rather, these modifications can be easily resolved via simple changes to the command line.

In situations where assessment of the functional form suggests parabolic curvature, one may choose to add a quadratic term to the model. The following simple extension allows for the linear term to be random, but the quadratic term to be fixed:

```
> lme(math~year+I(year^2), random=~year|schoolid/childid,
+ data=egsingle)
```

It is also common to modify the centering of the time variable to reduce collinearity among variables or to enhance interpretation. For example, assume we would like to change the way `year` is centered in the `egsingle` dataset such that it reflects the first observation (`-2.5`) for all students. This would be accomplished as:

```
> lme(math~I(year+2.5), random=~I(year+2.5)|schoolid/childid,
+ data=egsingle)
```

Last, one may want to include additional sources of variation and extend the model to include additional levels of nesting. For example, if multiple observations were collected on

individual students during the course of one school year, and teacher IDs were provided, it would be possible to construct a four-level hierarchical model via the following simple modification to the nesting structure of the random segment:

```
> lme(math~year, random=~year|schoolid/teacherid/childid,
+ data=data)
```

Conclusions

The **nlme** package is a powerful extension to the base statistical package that can be used to fit all of the common value-added models that have appeared in the literature. Additional extensions to this basic model can be accomplished via simple generalizations to the `lme` code presented in the previous sections.

Though **nlme** is powerful, one may find that very large data sets may fail to converge, especially when the size of the $\boldsymbol{Z}$ matrix is extremely large, such as when fitting models with crossed random effects. However, recent computational advances to **lme4**, an R package for linear mixed models, presents researchers with the ability to fit models with partially crossed grouping factors to data sets with 10^5 or more observations by capitalizing on sparse matrix algorithms (Bates, 2004; Bates & Debroy, 2004). These improvements should enhance the viability of R for applications of VAM to real datasets.

*

References

Ballou, D., Sanders, W., & Wright, P. (2004). Controlling for student background in value-added assessment of teachers. *Journal of Educational and Behavioral Statistics, 29*(1), 37-65.

Bates, D. (2004). *Sparse matrix representations of linear mixed models* (Tech. Rep.). http://www.stat.wisc.edu/ bates/reports/MixedEffects.pdf: University of Wisconsin-Madison.

Bates, D., & Debroy, S. (2004). Linear mixed models and penalized least squares. *Journal of Multivariate Analysis, 91*(1), 1-17.

Bryk, A., Raudenbush, S., & Congdon, R. (1996). *HLM: Hierarchical linear and nonlinear modeling with the HLM/2L and HLM/3L programs.* Chicago, IL: Scientific Software International, Inc.

Doran, H. C. (2004, July). *Value-added models and adequate yearly progress: Combining growth and adequacy in a standards-based environment.* Paper presented at the Annual CCSSO Large-Scale Assessment Conference.

Ferron, J. (1997). Moving between hierarchical modeling notations. *Journal of Educational and Behavioral Statistics, 22*(1), 119-23.

Fox, J., & Andersen, R. (2004, June). *Using the R statistical computing environment to teach social statistics courses.* http://socserv.socsci.mcmaster.ca/jfox/Teaching-with-R.pdf.

Goldstein, H. (1995). *Multilevel statistical models.* London: Oxford University Press.

Kurdek, L. A., du Toit, S., & du Toit, M. (2004). *Multilevel modeling of repeated measurements utilizing dummy variables: examples from growth-curve analyses with individuals and spouses from married couples* (Tech. Rep.). http://www.ssicentral.com/other/kurdek.htm: Wright University.

Lockwood, J., Doran, H. C., & McCaffrey, D. F. (2003). Using R for estimating longitudinal student achievement models. *The Newsletter of the R Project, 3*(3), 17-23.

MacCallum, R., & Kim, C. (2000). Modeling multivariate change. In T. Little, K. Schnabel, & J. Baumert (Eds.), *Modeling longitudinal and multilevel data.* Mahwah, NJ: Lawrence Erlbaum.

McCaffrey, D. F., Lockwood, J., Koretz, D., Louis, T., & Hamilton, L. (2004). Models for value-added modelling of teacher effects. *Journal of Educational and Behavioral Statistics*, *29*(1), 67-101.

Meyer, R. H. (1997). Value-added indicators of school performance: A primer. *Economics of education review*, *16*(3), 283-301.

Peterson, N. S., Kolen, M. J., & Hoover, H. (1989). Scaling, norming, and equating. In R. L. Linn (Ed.), *Educational measurement* (Third ed., p. 221-262). New York, NY: Macmillan.

Pinheiro, J., & Bates, D. (2000). *Mixed-effects models in S and S-Plus*. New York, NY: Springer.

R Development Core Team. (2004). *R: A language and environment for statistical computing.* Vienna, Austria. (ISBN 3-900051-00-3)

Raudenbush, S., & Bryk, A. (1986). A hierarchical model for studying school effects. *Sociology of Education*, *59*, 1-17.

Raudenbush, S., & Bryk, A. (2002). *Hierarchical linear models: Applications and data analysis methods* (Second ed.). Newbury Park, CA: Sage.

Rowan, B., Correnti, R., & Miller, R. J. (2002). What large-scale, survey research tells us about teacher effects on student achievement: Insights from the prospects study of elementary schools. *Teachers College Record*, *104*(8), 1525-1567.

Sanders, W. L., Saxton, A., & Horn, S. (1997). The Tennessee value-added assessment system: A quantitative, outcomes-based approach to educational assessment. In J. Millman (Ed.), *Grading teachers, grading schools: Is student achievement a valid evaluation measure?* Thousand Oaks, CA: Corwin Press.

Singer, J., & Willet, J. (2003). *Applied longitudinal data analysis: Modeling change and event occurrence*. New York, NY: Oxford University Press.

Singer, J. D. (1998). Using SAS PROC MIXED to fit multilevel models, hierarchical models, and individual growth models. *Journal of Educational and Behavioral Statistics, 24*(4), 323-355.

Tekwe, C. D., Carter, R. L., Ma, C.-X., Algina, J., Lucas, M., Roth, J., et al. (2004). An empirical comparison of statistical models for value-added assessment of school performance. *Journal of Educational and Behavioral Statistics*, *29*(1), 11-36.

Thum, Y. M. (1997). Hierarchical linear models for multivariate outcomes. *Journal of Educational and Behavioral Statistics*, *22*(1), 77-108.

Willett, J. B. (1989). Questions and answers in the measurement of change. In E. Rothkopf (Ed.), *Review of research in education* (Vol. 15, p. 345-422). Washington, DC: American Educational Research Association.