SOFTWARE FOR MULTILEVEL ANALYSIS

JAN DE LEEUW AND ITA G.G. KREFT

CONTENTS

1. INTRODUCTION

In this paper we review some of the more important software programs and packages that can are designed for, or can be used for, multilevel analysis. These programs differ in many respects. Some are part of major statistics packages such as SAS or BMDP. Others are written in the macro language of a major package. And some are stand-alone special purpose programs that can do nothing but multilevel analysis.

We have been involved in a number of these comparisons before. The first one [Kreft et al., 1990], comparing HLM, ML3, VARCL, BMDP5-V, and GENMOD, was published in Kreft et al. [1994]. The second comparison [van der Leeden et al., 1991], comparing HLM, ML3, and BMDP5-V on repeated measures data, was published in van der Leeden et al. [1996]. We give both the reference to the internal report version and to the published version, because the unpublished version is usually has much more material. Giving both references also shows the unfortunate time-interval between the two, which is especially annoying in the case of software reviews. The reviews were summarized briefly in our book [Kreft and de Leeuw, 1998, Section 1.6].

Since our last publication on the subject, there have been many major changes. The program `GENMOD`, which was never easy to obtain, has more or less completely disappeared. `VARCL`, which was one of the leading contenders in the early nineties, is not actively supported or developed any more, which means it rapidly lost ground. `BMDP`, as a company, went out of business, which had serious consequences for its software products. Programs such as `HLM`, written originally for `DOS`, were upgraded for Windows. `ML3` transformed to `MLn`, which transformed to `MLwiN`. And so on. Obviously, it is time for an update.

In this update, we will change our focus somewhat. We will indicate what the programs can do, where you can get them, for how much money, on what systems they run, and how easy it is to use them. We do not emphasize computation speed, because this is hard to define, and not very relevant in most applications anyway. Computing time is usually infinitely small compared to the time needed to collect and clean the data.

1.1. **Programs, Macros, and Packages.** Throughout this review we use a number of classifications of the software we review. Of course the boundaries between the categories are somewhat fuzzy.

Research versus Production: Some software is written as a research tool, to analyze the examples in a research paper, or to analyze the data sets in a research project. Other software is written as a commercial product, or at least it is clearly intended for general use. Many programs start out in the research phase, and are subsequently promoted to the production phase. Many programs are still somewhere in between.

Stand-alone versus Module: There are multilevel programs that stand on their own, i.e. they are executables and they do not require other software (except the operating system, of course) to be present. Other programs are modules of an existing package. Usually they require the rest of the package to be present as well.

Program versus Macro: For programs written on top of an existing package, there are still two possibilities. Either the program is a module, existing in object code within the package, or the program is a macro, written in the scripting or extension language of the package. Macros are handled by an interpreter that is part of the package.

Another distinction we make is between minor and major specialized programs. This is somewhat subjective, but the general idea is that a research program, usually with a rather primitive interface, is a minor specialized program. As soon as the authors start thinking about user friendliness, making a nice graphical user interface, adding options that people have asked for, and going commercial, the minor program will become a major one. It will become bigger, have more possibilities, look better, and cost more.

1.2. **Omissions.** There are some programs that can be used to perform multilevel analysis, but that we do not discuss in detail. We just mention these programs briefly here, and we give the URL[1] in case readers want to know more.

`TERRACE`: This is multilevel research software, written by James Hilden-Minton for his Ph.D. thesis. It is an add-on to the `XLISP-STAT` package, and it can be found (with manual) at

`www.stat.ucla.edu/consult/paid/nels/papers`.

`NLME`: The `NLME` software comprises a set of `S-plus` functions, methods, and classes for the analysis of both linear and nonlinear mixed-effects models. It extends the linear and nonlinear modeling facilities available in release 3 of `S-plus`. Written by José C. Pinheiro and Douglas M. Bates. It is available for Unix and Windows platforms from

`franz.stat.wisc.edu/pub/NLME`.

`BUGS`: `BUGS` is a piece of computer software which permits the analysis of complex statistical models using Markov Chain Monte Carlo methods. The emphasis is on the Monte Carlo method, and a great variety of multilevel models can be analyzed as well. See

`www.mrc-bsu.cam.ac.uk/bugs/mainpage.html`

for further details.

`Oswald`: `Oswald`, developed by the Statistics Group at the University of Lancaster, is a suite of `S-plus` functions for analyzing longitudinal data.

`www.maths.lancs.ac.uk/Software/Oswald/`.

It includes mixed effects models, and many other possible options.

2. Major Specialized Programs

There are two major specialized programs for performing multilevel analysis, one from the UK and one from the US.

[1] Throughout, we omit the `http://` part of the URL's.

2.1. MLwiN.

The ML series of programs has a complicated history. The series was erected on top of the NANOSTAT program by Michael Healy. NANOSTAT is a general purpose statistics program. The multilevel extensions started with ML2 in 1988, ML3 was introduced in 1990, and the final DOS program in the series was MLn, published in 1995. In 1998 they were all superseded by MLwiN. One can think of MLwiN as a separate program, but also as a graphical user interface on top of MLn. Throughout the project most of the programming was done by Jon Rasbash, but clearly the program is the result of a team effort.

MLwiN contains the NANOSTAT package, so it can do a fair amount of data manipulation and general purpose statistics. This is all meant to assist in the multilevel analysis, and thus we still think of MLwiN as a specialized program.

In our report [Kreft et al., 1990] we looked at ML2, in the published version [Kreft et al., 1994] at ML3. In a subsequent comparison, the internal report version [van der Leeden et al., 1991] looked at ML3, the published version [van der Leeden et al., 1996] at MLn.

We shall only discuss MLwiN in this paper, since it includes all previous programs as special cases, and since most people seem to feel that putting a Windows interface on top of a DOS program is a step ahead.

2.1.1. *Availability.* The Multilevel Project has three mirror homepages at

```
www.ioe.ac.uk/multilevel/
www.medent.umontreal.ca/multilevel/
www.edfac.unimelb.edu.au/multilevel/
```

The websites have information about the software, but also about the project and its activities. The MLwiN program has its own homepage at

```
www.ioe.ac.uk/mlwin/
```

The price of a single copy given on the MLwiN website is $ 900 (£ 500), with a 40% educational discount.

2.1.2. *Documentation.* There is a large amount of documentation available, most of it is can be obtained from the multilevel project. Again, the various versions of the program lead to a somewhat bewildering variety of documents. Basically, there are three types of documents available for most versions of the program. The first document is the user's guide. For MLwiN this is Goldstein et al. [1998]. There is another type of user's guide, which is more introductory. For MLn this is Woodhouse [1995]. The second is the advanced manual, that discusses macros to fit more complicated models. This is Yang et al. [1998]. Thirdly, there is the command reference. Since the command language for MLwiN is actually MLn, this is Rasbash and Woodhouse [1995].

In a sense, the most important document in the Multilevel Project is the book by Harvey Goldstein. Not surprisingly, the book also exists in two very different editions. It developed with the project. The first edition [1987] covers basic multilevel analysis, with emphasis on applications in education. The second edition [1995] is much more statistical, covers a large variety of extensions of the basic model and a much broader range of applications.

2.1.3. *Model.* The basic model[2], in the case of two levels and p predictors, is

$$\underline{y}_{ij} = \underline{\beta}_{0ij} x_{0ij} + \underline{\beta}_{1ij} x_{1ij} + \ldots + \underline{\beta}_{pij} x_{pij}, \tag{1a}$$

where

$$\underline{\beta}_{sij} = \beta_s + \underline{u}_{sj} + \underline{e}_{sij}. \tag{1b}$$

It is easy to see how this generalizes to more than two levels. Observe that usually we have $x_{0ij} = 1$ for all i, j, i.e. the zero term in the regression corresponds with the intercept. Also observe that each regression coefficient has a fixed part and a random part, where the random part has random components for both levels.

2.1.4. *Interface.* As indicated above, `MLwiN` provides (or maybe we should say "is") a Windows interface on top of `MLn`.

It is possible to use `MLwiN` very much like `MLn`, because there is a command window in which `MLn` commands can be entered. This window also tracks the command history. In fact, menu commands are translated to `MLn` commands in the command window. There are many `MLn` and `NANOSTAT` commands which *cannot* be entered from the menus.

An interesting feature of the program is the *equations window*. This is a specialized equations editor, in which the model can be defined in equation form by clicking and pasting. Obviously, there is no way to do this in `DOS`, so this defines is a clear distinction between `MLn` and `MLwiN`. On the other hand, it is unclear if this is actually superior to written output that the user herself has to read translate into a formula. It seems that if users know what they are doing, they can make the translation very easily themselves. If they don't know what they are doing, then they should not be using the program in the first place.

2.1.5. *Algorithm.* `MLwiN` uses the `IGLS` or the `RIGLS` algorithms first described, respectively, by Goldstein [1986] and Goldstein [1989].

The algorithms are block-relation algorithms. There are two blocks of parameters, the fixed regression coefficients and the variance/covariance

[2] As in our book, we use the convention that random variables are underlined.

components. The algorithm fixes the variance components at some initial value and maximizes the likelihood over the fixed coefficients. This is just a Generalized Least Squares problem. Then it fixes the coefficients at their current values and maximizes the likelihood over the variance components, by solving another, more complicated, Generalized Least Squares problem. The two optimizations are alternated until convergence. A concise description appears in Goldstein and Rasbash [1992].

It is not entirely clear from the documentation what happens in boundary cases when dispersion matrices become singular or even indefinite.

2.2. **Extensions.** By “extensions” we mean various options and additions that do not really belong to the core of the program, but that the authors have added because of user demand, competitive pressure, or their research program. Some extensions are obviously more useful for the general public than others, but most of them are at least interesting enough to be mentioned.

In `MLwiN` there are two “levels” of extensions. The first are features which are part of the program core. They can be handled by using the menus, or the `MLn` language, but they are features that will not often be needed. The second level consists of true extensions, written in the macro language provided by the package, and these are add-ons that one may or may not load.

The first class of extension are discussed in the user’s guide [Goldstein et al., 1998]. This already covers a substantial number of procedures. Hierarchical generalized linear models with binomial or Poisson outcomes can be fitted. Monte Carlo Markov Chain methods are available to optimize complicated likelihoods or compute complicated posterior distributions. Parametric bootstrap methods are used for bias correction and for standard error computation.

These extensions may be exciting to some, but they do take `MLwiN` several steps in the direction of research software. The user has to take very many things for granted, and has to hope that the default values of the many parameters and tuning constants work in her case. It seems a bit too demanding to ask the casual user to choose between the MQL and PQL methods for quasi-likelihood estimation, or to choose an appropriate burn-in period for her MCMC procedure. It is true that with these extensions models can be fitted that could not be fitted by older versions of `MLn`. But for most people it becomes impossible to understand what is actually going on inside the program, and to explain why certain choices and not others were made.

The macro-based extensions of `MLwiN` allow for even more flexibility. Macros are available for fitting multi-category models, survival models, time-series models, and nonlinear models. While these extensions are undoubtedly useful, again the reservations in the previous paragraph apply. For the average MLwiN user, the instructions in the manual are voodoo. There are references to the statistical literature, of course, but these references are in many cases too technical to be of much use.

This, of course, is a well-known dilemma. If applied researchers, with often quite limited technical expertise, want to fit very complicated models with very complicated algorithms, then the documentation and the implementation should handle this very carefully. Both in Goldstein [1995] and in the various manuals one often gets the idea that instead of carefully guiding the user through complicated territory, she is invited to climb on the roller coaster, close her eyes, and enjoy the ride.

2.3. HLM.

HLM's the history is similar to MLwiN's. There first was a HLM for two-level models, then one for three-level models, then one which could also do generalized linear regression, and finally a Windows version. This last version, version 4 of HLM/2L and HLM/3L, is the one we review here. Of course HLM/2L does two-level analysis and HLM/3L three-level analysis. The programming was done by Richard Congdon, the HLM team also includes Stephen Raudenbush and Tony Bryk.

2.3.1. *Availability.* The software is available from Scientific Software International in Chicago, where HLM has its homepage

www.ssicentral.com/hlm/mainhlm.htm.

The price is $ 400 for the DOS version and $ 430 for the Windows version.

2.3.2. *Documentation.* The HLM documentation consists of a user's manual [Bryk et al., 1996] and of the book by Bryk and Raudenbush [1991]. In a sense, the book is independent of the software, but since you get the book when you buy the software, and since the HLM program is used throughout the book, the two are really intimately related (even closer than MLwiN and Goldstein [1995]). The choice of name HLM can lead to unfortunate confusion of the model, the technique, and the package [de Leeuw and Kreft, 1995].

The Windows interface to HLM is only documented in the on-line help, it seems.

2.3.3. *Model.* The basic HLM model[3], for two-level data, is

$$\underline{y}_{ij} = \underline{\beta}_{0j}x_{0ij} + \underline{\beta}_{1j}x_{1ij} + \ldots + \underline{\beta}_{pj}x_{pij} + \underline{e}_{ij}, \tag{2a}$$

[3] We have changed the HLM notation to make it conform more to MLwiN notation

where

$$\underline{\beta}_{sj} = \gamma_{s0} z_{j0} + \gamma_{s1} z_{j1} + \ldots + \gamma_{sh} z_{jh} + \underline{u}_{sj}. \tag{2b}$$

To make the comparison with MLwiN somewhat easier, we rewrite this as

$$\underline{y}_{ij} = \gamma_{00} x_{0ij} z_{j0} + \ldots + \gamma_{ph} x_{pij} z_{jh} + \underline{u}_{0j} x_{0ij} + \ldots + \underline{u}_{pj} x_{pij} + \underline{e}_{ij}. \tag{3}$$

There are two obvious differences between the programs.

1. In MLwiN we can have random coefficients $\underline{\beta}_{sij}$ on the first level. This is not possible in HLM.
2. In HLM the emphasis is on the cross-level interactions in the fixed part, which are products of a first-level and a second-level predictor. It is possible to have such variables in MLwiN, but they are not as central.

2.3.4. *Interface.* Similar to what we saw in MLn, the newer versions of HLM now have a Windows interface. In addition, they have the interactive (question-and-answer) and batchfile interfaces from the older versions. In Bryk et al. [1996] the authors remark that most PC users will prefer the Windows interface, but from the rest of the book it is pretty clear that they themselves do not.

The Windows version of HLM also comes with an equation editor, similar to the one in MLwiN.

2.3.5. *Algorithm.* By default, HLM/2L uses REML estimation, while HLM/3L uses FIML. Nevertheless, both programs can actually do both forms of estimation. Older versions of HLM relied on the EM algorithm, which can sometimes be hopelessly slow. It is now possible to speed up convergence by switching to Fisher scoring.

2.3.6. *Extensions.* There are a number of interesting extension in HLM. First, there is the V-known option (where the variance components are supposed to be known) which can be used in meta-analysis. Second, there are hierarchical generalized linear models, fitted by using penalized quasi-likelihood or generalized estimation equations. In particular, Poisson, Bernoulli, and binomial models can be fitted. Third, a form of plausible value analysis using multiple imputation is available for two-level models. Although the scope of HLM is obviously much smaller than that of MLwiN, this restriction of generality makes the program easier to use. Generally, the documentation is more user-oriented, the number of choices the user can make is more limited, and often the authors of the program have already made many of the choices.

3. Minor Specialized Program

3.1. VARCL.

The program VARCL started out in the mid-eighties as one of the major contenders. It was Longford's research software, used in the path-breaking paper by Aitkin and Longford [1986] and in his book [Longford, 1993]. Since 1990, Longford has moved to using S-plus for research software development, and nothing has happened with VARCL. It is still a DOS program.

3.1.1. *Availability.* The program is sold by ProGAMMA in Groningen, Netherlands, but it is in a very remote section of their catalog:

indy1.gamma.rug.nl/sibweb/catalog/Catalo10.htm#I12

The price is $ 350 ($ 250 educational).

3.1.2. *Documentation.* The distribution comes with a 100-page user manual [Longford, 1990], with an additional 100 pages of example runs. The manual follows the interactive interface closely, but 20 pages are used for explaining some of the technical background.

3.1.3. *Model.* There are two different versions of VARCL available. The first one is VARCL3, which handles three-level models, the second is VARCL9, which handles random intercept models with up to nine levels. The model in VARCL3 is

$$\underline{y}_{ijh} = \underline{\beta}_{0jh} x_{0ijh} + \underline{\beta}_{1jh} x_{1ijh} + \ldots + \underline{\beta}_{pjh} x_{pijh} + \underline{e}_{ijh}, \tag{4a}$$

where

$$\underline{\beta}_{sjh} = \beta_s + \underline{u}_{sh} + \underline{u}_{sjh} \tag{4b}$$

Again, we use the notation of MLwiN, and we assume that $x_{0ijh} = 1$ for all i, j, h.

In VARCL9 the model is (for four levels, as an example)

$$\underline{y}_{ijhg} = \underline{\alpha}_{jhg} + \beta_1 x_{1ijhg} + \ldots + \beta_p x_{pijhg} + \underline{e}_{ijgh}, \tag{5a}$$

where

$$\underline{\alpha}_{jgh} = \alpha + \underline{u}_g + \underline{u}_{hg} + \underline{u}_{jhg}. \tag{5b}$$

Here we have singled out the intercept more explicitly, by using α for the corresponding regression coefficient.

We see that VARCL does not have the emphasis on cross-level interactions (or on two-level specification) that HLM has. It also does not have the possibility of random coefficients on the first level, which we have in MLwiN.

3.1.4. *Interface.* `VARCL` has a command-line interface. The programs asks a large number of questions, and it uses the information provided by the user to build up a setup file that describes the analysis. It first builds up information about the maximal model, which is the largest model[4] that can be fitted in a session. Then additional questions are used to exclude variables from the maximal model to define the model to be fitted. Constraints on the parameters can be defined, by setting them to fixed values. The model is shown, and if the user likes it, the program computes the estimates. Then the interface asks if additional models will be fitted, and if so, which variables are to be removed and added, and which parameters have to be constrained or freed.

If you know from experience which questions the program will ask, and which answers you are going to give, then you can put the answers in a batch file and take standard input from that file.

3.1.5. *Algorithm.* The program uses the scoring method on the reduced form. It is based on the fact that within-group means, variances and covariances are jointly sufficient for the model parameters. Thus if we compute the within-group statistics for the maximal model, we can forget about the original data. Since these within-group statistics can be computed in the input loop, the number of observations that `VARCL` can handle is infinitely large, as long as they come in a finite number of groups. This is different from `MLwiN`, which keeps the data in core for the whole session. It is similar to `HLM`, which uses the within-group statistics to compute separate regressions for each group (if the groups are big enough). Both `VARCL` and `HLM` use within group statistics to compute initial estimates of the variance and covariance components.

In our experience, documented in our previous review papers, the scoring algorithms is both fast and reliable.

3.1.6. *Extensions.* `VARCL` was also the first multilevel program that could deal with hierarchical generalized linear models. Early on in the interactive questions and answers session the user can choose the error model to be either normal, binomial, Poisson, or gamma. Again quasi-likelihood methods are used to fit these models, using the procedures designed by Longford [1988] for this purpose.

3.2. `MIXFOO`. The `MIXFOO` program is special, because it does not exist. It is the generic name for a whole series of multilevel programs. At the `MIXFOO` website we find

> The statistical; research presented in this homepage is based
> on the collaborative effort of Donald Hedeker and Robert D.

[4] in terms of the number of variables

> Gibbons of the University of Illinois at Chicago. The computer programs were written by Don Hedeker with interfaces written by Dave Patterson (Discerning Systems, Inc.).

The names of all programs in the series start with `MIX`. There are `DOS` programs `MIXOR`, `MIXREG`, `MIXGSUR`, `MIXNO`, and `MIXPREG`, while `MIXOR` and `MIXREG` also exist with Windows interfaces. The website also has a `SAS-IML` macro that fit a random intercept version of `MIXREG`, and `SPSS-MATRIX` macros for random intercept versions of both `MIXOR` and `MIXREG`. All this is freely available from the website.

We indicate briefly what these programs do. `MIXREG` fits the linear multilevel model, but it allows for various forms of autocorrelation between the first-level disturbances. `MIXOR` adds ordinal multicategory outcomes. `MIXGSUR` does grouped-time survival data, `MIXNO` does nominal multicategory data, and `MIXPREG` does multilevel Poisson regression.

3.2.1. *Availability.* Software and manuals can be downloaded from

`www.uic.edu/˜hedeker/mix.html`

All programs, macros, and manuals are free. There are PowerMac and Sun/Solaris version of the software at

`www.stat.ucla.edu/˜deleeuw/mixfoo`.

It must be emphasized that this is quite unique, all other programs (except `VARCL` and `MLA`) only exist for `DOS` or Windows.

3.2.2. *Documentation.* The two core programs, `MIXOR` and `MIXREG`, are described in Hedeker and Gibbons [1996a] and Hedeker and Gibbons [1996b]. These are published versions of the manuals, the manuals themselves are included in the software distribution. The theory is described in great detail in Hedeker [1989] and for `MIXOR` in the article by Hedeker and Gibbons [1994]. `MIXGSUR`, `MIXNO`, and `MIXPREG` also have a manuals on the website. The theory of `MIXGSUR` is described in the technical report by by Hedeker et al. [1996]. More generally, there is a list of both theoretical and applied articles using `MIXFOO` at

`www.uic.edu/˜hedeker/works.html`.

3.2.3. *Model.* For `MIXREG` the model is

$$\underline{y}_{ij} = \alpha_0 w_{oij} + \ldots + \alpha_p w_{pij} + \underline{\beta}_{0j} x_{0ij} + \ldots + \underline{\beta}_{qj} x_{qij} + \underline{e}_{ij}, \tag{6a}$$

with

$$\underline{\beta}_{sj} = \beta_s + \underline{u}_{sj}. \tag{6b}$$

This is a straightforward two-level random coefficient model, very similar to what we have in `VARCL`. The unique aspect is that we do not assume that

$$\mathbf{C}\left(\underline{e}_{ij},\underline{e}_{kj}\right)=\delta^{ik}\sigma^{2}, \tag{6c}$$

but we allow for a much more general parametric first level error covariance structure. More specifically, the $\underline{e}_{ij}$ can have an autoregressive AR(1), moving average MA(1), or autoregressive moving average ARMA(1,1) covariance structure, a general stationary autocorrelation structure, and even a special nonstationary one.

For `MIXOR` the model is the same, except that we do not observe the $\underline{y}_{ij}$, we observe a multi-category version generated by a threshold model. Thus there are unknown cut-off points (assumed to be the same for all variables). If $\underline{y}_{ij}$ is below the first cut-off we observe a "1", if it is between the first and second cut-off we observe a "2", and so on. Thus the model is basically the same as before, but there are "missing data" because we only know that $\underline{y}_{ij}$ is in a particular interval, but we don't know where it is in that interval. More precisely the model is

$$\underline{\eta}_{ij}=\alpha_0 w_{oij}+\ldots+\alpha_p w_{pij}+\underline{\beta}_{0j}x_{0ij}+\ldots+\underline{\beta}_{qj}x_{qij}+\underline{e}_{ij}, \tag{7a}$$

where $\underline{\eta}_{ij}$ is the unobserved continuous response, and

$$\underline{y}_{ij}=\begin{cases}1, & \text{if } \kappa_0\leq\underline{\eta}_{ij}<\kappa_1,\\ 2, & \text{if } \kappa_1\leq\underline{\eta}_{ij}<\kappa_2,\\ \ldots & \\ r, & \text{if } \kappa_{r-1}\leq\underline{\eta}_{ij}<\kappa_r,\end{cases} \tag{7b}$$

where $\kappa_0=-\infty$ and $\kappa_1=+\infty$.

3.2.4. *Interface.* All programs exist as batch versions, using command files. `MIXOR` and `MIXREG` also have interactive `DOS` version, in which the command file is constructed from menu commands issued by the user. Finally, `MIXREG` and `MIXOR` are available as Windows programs.

3.2.5. *Algorithm.* `MIXREG` uses a combination of the `EM` and the scoring algorithm, in much the same way as, for instance, `HLM`. For `MIXOR` there are additional complications, because multidimensional integrals must be evaluated to compute the likelihood and its derivatives. `MIXOR` approximates these integrals by using Gauss-Hermite quadrature.

3.2.6. *Extensions.* If we consider `MIXREG` and `MIXOR` to be the basic components, then `MIXGSUR`, `MIXNO`, and `MIXPREG` are extensions. But of course drawing the line in this way is rather arbitrary. It is better to think of the whole set of programs as a modular alternative to programs such as

HLM and MLwiN, which cover about the same amount of territory in a single program.

3.3. MLA. MLA is a batch program, running under DOS. It is written by Frank Busing, Rien van der Leeden, and Eric Meijer of the University of Leiden, The Netherlands. It differs from similar program because it has extensive simulation possibilities built in (notably, the bootstrap and the jackknife), and it has various ordinary least squares estimation methods as options.

3.3.1. *Availability.* Software and manual can be downloaded from

```
www.fsw.leidenuniv.nl/www/w3_ment/
    medewerkers/BUSING/MLA.HTM
```

The program is free.

3.3.2. *Documentation.* The software distribution contains a 70-page Postscript manual [Busing et al., 1994]. It is a bit wordy, because it tries to spell out all details, especially the technical ones. The ultimate example is the 20-page technical appendix A, which gives in painstaking detail the derivations of the formulas for the likelihood function, its first and second derivatives, and their expectations. It illustrates the effect TEX has on the mind of an individual who has just escaped from the dungeons of WYSIWYG. The user's guide portion of the manual is quite clear, however.

The authors have informed us that the manual is out of date, because many options have been added to MLA in the meantime. The program can now make histograms (of bootstrap results, for instance) and scatterplots (of residuals) and there are different bootstrap-based confidence intervals. Also, permutation tests for testing intra-class correlation are available.

3.3.3. *Model.* The model is the same as in (2a) and (2b).

3.3.4. *Interface.* MLA requires the user to create a parameter file, and then the program is started from the DOS command line as

```
mla < inputfile > outputfile
```

Not much of an interface, but it does the job.

3.3.5. *Algorithm.* MLA uses the BFGS algorithm, which is a general purpose quasi-Newton optimization algorithm, to maximize either FIML or REML. Alternatively, it can also use EM. In order to make sure that the level-two dispersion matrix is positive semi-definite, two different parametrizations are available that ensure this.

3.3.6. *Extensions.* MLA is somewhat unremarkable as a multilevel program (although the parametrizations of the dispersion matrix and the use of BFGS are unique) But it is remarkable because it supports a wide variety of simulation analyses. This tends to suggest it is research software, but for practitioners the confidence information will also be useful.

Here is a list of the unique features taken from the website.

- different kinds of simulations (bootstrap, jackknife and permutation)
- different methods of bootstrap simulation (cases, parametric and error)
- different types of residual estimation (raw and shrunken)
- different cases resampling schemes (level-1, level-2 and both)
- balanced resampling schemes (balanced bootstrap)
- linking of residual levels
- distribution plots (histograms) for parameters, standard errors and t-value

4. Modules in a Major Package

4.1. BMDP5-V. BMDP was the first major statistical package. It has always had an excellent reputation, because its approximately 40 programs were developed in close cooperation with excellent statisticians. But BMDP has a stormy recent past. The company was bought by SPSS in 1996. It is difficult to find BMDP on the main SPSS web site at http://www.spss.com. After some search, it turns out that many of the BMDP products have been declared "dead" by SPSS, and the only product still sold seems to be BMDP Classic for DOS. See

www.spss.com/software/science/BMDP.

Some additional research suggests that SPSS plans to integrate BMDP in its SYSTAT product. In corporate-speak: "I would like to welcome BMDP customers to the SYSTAT family."

In Europe the situation is different. One can buy BMDP from Statistical Solutions in Cork, and their web page clearly indicates they consider it to be one of their major products.

4.1.1. *Availability.* BMDP can be ordered from

www.statsol.ie/bmdp.html

The price is $ 895 ($ 695 Academic), but of course this is for the whole package (all 40+ routines, of which BMDP5-V is only a single one).

4.1.2. *Documentation.* The theory behind BMDP5-V is explained in Jennrich and Schluchter [1986]. Very useful information about both the methodology and the program is in Schluchter [1988]. There is also a chapter on 5V in the BMDP user's manual [Dixon, 1992] and the BMDP user's digest [Software, 1992]. And, last but not least, there is a detailed how-to-use book with many examples [Dixon and Merdian, 1992] (with a 5.25 floppy in the back !).

4.1.3. *Model.* `BMDP5-V` is described in the documentation as a program for unbalanced repeated measures models with structured covariance matrices. The means that the emphasis is not on general hierarchies, but specifically on repeated measures, i.e. on repetitions nested within subjects.

To maintain the same notation as we have used before, we rewrite the model as

$$\underline{y}_{ij} = x_{ij1}\beta_1 + \ldots + x_{ijp}\beta_p + \underline{e}_{ij}, \tag{8}$$

where index j now stands for subjects, and index i stands for the repeated measures (nested within subjects). Thus it seems there are no random coefficients.

`BMDP5-V` allows a large number of possible choices for the within-subject covariance matrices Σ_j which have elements

$$\sigma_{jik} = \mathbf{E}\,(\underline{e}_{ij}\underline{e}_{kj}). \tag{9}$$

We briefly describe the available options, in matrix notation,

$$\Sigma_j = \sigma^2 I, \tag{10a}$$

$$\Sigma_j = \omega^2 ee' + \sigma^2 I, \tag{10b}$$

$$\Sigma_j = Z_i \Omega Z_i' + \sigma^2 I, \tag{10c}$$

$$\Sigma_j = \Lambda\Lambda' + \Psi, \tag{10d}$$

$$\Sigma_j = \theta_1 G_1 + \ldots + \theta_r G_r, \tag{10e}$$

$$\Sigma_j = \Sigma. \tag{10f}$$

Moreover there are two options that can best be defined in elementwise notation

$$\sigma_{jik} = \sigma^2 \rho^{|i-k|}, \tag{10g}$$

$$\sigma_{jik} = \theta_r, \text{ where } r = |i-k| + 1 \tag{10h}$$

Clearly, this covers many of our well-known friends from earlier sections.

Model (10a) defines independent observations (within subjects, there is always between subject independence), while model (10b) (sometimes described as *compound symmetry*) is in fact the random intercept model. Model (10c) is the random-coefficient model, which allows for random slopes as well. Thus random coefficients, which seemed to be missing in the `BMDP5-V` specification, enter again through the back door. Model (10d) says that within-subject covariance matrices have a factor analysis structure, while (10e) describes a general linear covariance structure, and (10f) allows for any matrix (except that it must be the same over subjects). The final two models allow for autoregressive structures, similar to the ones we have seen in MIXREG. BMDP5-V also allows you to define your very own structure, provided you write a FORTRAN program to compute the structure and its derivatives. Really enterprising users could fit a repeated measures model in which the within-subject covariance structures satisfy a LISREL model, for instance.

4.1.4. *Interface.* BMDP is a very modular package. Each of the 44 programs can basically be used in a stand-alone way. This makes it quite different from SAS, which is much more integrated. Modularity produces a great deal of efficiency. On the other hand, BMDP is very, very DOS. It comes with BMDP/DYNAMIC, a somewhat optimistically named data and program managing module. BMDP/DYNAMIC has the familiar DOS type pseudo-menus and pseudo-windows. It is fairly lightweight in terms of overhead.

4.1.5. *Algorithm.* There is a variety of algorithms available. If you choose the FIML criterion, you can use Fisher scoring, Newton-Raphson, or generalized EM. In case of REML, generalized EM and quasi-scoring are available.

4.1.6. *Extensions.* Some of the options for the within-subject covariance matrix are rather esoteric and could be considered extensions. Certainly options for which you have to write the code yourself qualify as such.

4.2. PROC MIXED. SAS has many modules (or PROC's) to fit linear models and generalized linear models. Until fairly recently, however, there was nothing that could compete with BMDP5-V. But now there is, and with a vengeance. PROC MIXED is the most general program of the ones we discuss, it is also embedded in the totality of SAS(or whatever parts of SAS you happen to have installed).

4.2.1. *Availability.* You can obviously order the package from SASInstitute. Start at

```
www.sas.com/service/techsup/faq
   /stat_proc/mixeproc.html
```

Because SAS is such a huge and complicated product, it does not make much sense to give a price. Many users will simply add PROC MIXED to their PC or mainframe SAS setup. Many people probably already have it available without actually knowing this.

4.2.2. *Documentation.* The documentation for PROC MIXED is very extensive. It consist of a chapter in SAS [1992]. This is written in the familiar format of the SAS manuals. It has almost 100 pages, setup like a user's guide. There is not much theory, and not many examples, but the instructions and options are explained in detail.

There is also an excellent and extensive book by Littell et al. [1996]. It has chapters on randomized blocks, repeated measures, multilevel models, analysis of covariance, spatial models, heteroscedastic models, and best linear unbiased prediction. There are also chapters on generalized linear mixed models and nonlinear mixed models, where the SAS-IML macros GLIMMIX and NLINMIX are explained. The book contains many examples, worked out in detail.

For people who have started out with a program such as HLM, which has a much simpler interface and a much smaller menu of models, the use of PROC MIXED is explained expertly by Singer [1998].

In terms of documentation, PROC MIXED is the clear winner in our comparison. In most other repects, it is not.

4.2.3. *Model.* The model in PROC MIXED is very similar to the model in BMDP5-V. It may even be true, historically, that BMDP5-V inspired PROC MIXED. The main difference, however, is that the PROC MIXED specification does not have levels or repeated measures. They start from the general mixed model, which is

$$\underline{y}_i = \beta_1 x_{i1} + \ldots + \beta_r x_{ir} + + \underline{u}_1 z_{i1} + \ldots + \underline{u}_s z_{is} + \underline{e}_i. \tag{11}$$

The random effects $\underline{u}$ have a covariance matrix Ω and the errors $\underline{e}$ have a covariance Δ. As in BMDP5-V , we can specify additional parametric structure in these covariance matrices. PROC MIXED allows for uncorrelated, compound symmetry, unstructured, autoregressive, and spatial (or a block-diagonal version of these – which is how the levels come in again). There is no factor analytic or general linear covariance structure.

It is interesting that HLM and MLA are true multilevel models, in the sense that they specify their regression model at multiple levels. MIXFOO and MLwiN have multiple levels, but only a single regression equation. The same is true for BMDP5-V, where the levels are defined by repeated measures. In PROC MIXED the levels have disappeared, and they have to be introduced by suitably arranging the input an parameter files. This is precisely the reason why a paper such as Singer [1998] is necessary for some groups of users.

4.2.4. *Interface.* The interface to PROC MIXED is the familiar SAS interface, or more precisely any one of the familiar SAS interfaces. As in the case of BMDP, all these interfaces are rather desperate attempts to get away from the mainframe or DOS heritage. Most of the windowing is used to construct batch files with options and instructions, which are then submitted to the old familair SAS engine in the background.

4.2.5. *Algorithm.* PROC MIXED can use both REML and FIML estimation, and it maximizes the likelihood by a combination of Fisher scoring and Newton-Raphson. The default is to use only Newton-Raphson, but optionally one can start with some Fisher scoring iterations.

4.2.6. *Extensions.* The two major extensions of PROC MIXED are both written in the macro language SAS-IML, by Russ Wolfinger. They are discussed in detail in Littell et al. [1996, Ch. 11 and 12].

Generalized linear mixed models can be fitted with the GLIMMIX macro. This incorporates binomial models with logit and probit links and Poisson (count) models with the log link. The macros uses a joint quasi-likelihood method. Nonlinear mixed models can be fitted with the NLINMIX macro. This macro actually has three options, depending on whether first-order or second-order or marginal approximations are used. This allows one to fit a really large range of models such as nonlinear growth curve models.

It is probably obvious that using such macros cannot possibly be very efficient. Using SAS to fit your multilevel models is already quite a stretch, because you have to carry along the enormous overhead of the SAS system. A class of students, each using SAS on a single CPU, will show you what this means. On top of that, the use of macros and the SAS-IML interpreter, adds another layer of inefficiency. It's a little bit like using a big truck to pick up some groceries at your local supermarket.

There is a trade-off, of course. Many people have SAS installed and are quite familiar with its interface and macro language. If they need to fit a multilevel model once in a while, it probably does not make sense for them to learn a new language or to buy a new software package. If you are already driving around in a truck, you might as well pick up your groceries.

5. Conclusions

We have already commented throughout the text on some of the differences. There are multilevel, random coefficient, and mixed linear model programs. There are package modules and stand alone programs. There are research and production programs. One obvious characteristic they have in common is that they are almost all DOS, trying to become Windows.

We do not really want to get too deeply into the discussion of the relative advantages of putting a GUI on top of a DOS program. Clearly, there are some definite advantages. Graphics output becomes much more attractive and more flexible, and the equations window of MLn is a useful addition, which would be impossible under DOS. There are also some disadvantages, which are the same as for other programs with the same history such as SAS, SPSS, and BMDP. We are forced to leave the safe domain of objective and impartial software comparison here, in order to voice an opinion. The GUI of all these programs is ugly as sin. This is partly due to the inherent ugliness of the Windows environment, but also to the fact that the interface is an afterthought. Its only job is to prepare parameter and data files, that are then passed to the DOS programs running in the background. One does not need a GUI to prepare parameter files, in fact this is not the natural way to prepare them at all. If you are a command line program at heart, then you should acknowledge that, and you should not pretend that a layer of Windows make-up can change this.

A command line interface which builds up the parameter file by asking questions, which is the interface for VARCL and which was also the interface for older versions of HLM, has some advantages over the Windows interfaces. It is obviously more structured, and the user does not have to search around in the menus for the appropriate commands. Strange and impossible combinations of options are more easily avoided. It is more fool-proof. On the other hand, it rapidly becomes very boring to go through the same sequence of questions every time, and more experienced users will prefer to type the answers ahead of time into a command file and use the program in batch mode (which makes the "interface" similar to the batch versions of MIXFOO or MLA).

If we summarize our findings, there is little doubt that MLwiN is the most comprehensive program for multilevel analysis is. Of course, it is not exactly fair to compare MLwiN with the whole SAS or even BMDP system. Those are more comprehensive, and they can also be used to perform multilevel analysis. HLM obviously has also developed into a mature and excellent product. Its basic design is more modest than MLwiN, it lacks the macro language, the graphics, and the NANOSTAT statistics and data handling. MLA is somewhat limited, but it can do many useful things, some of which are impossible or very hard to do in the other packages.

Probably for most academic users the price of the packages is not really a problem. The fact that MLA is free, and that MLwiN costs money does not seem to be a major problem. We must say, however, that the suit of MIXFOO programs by Hedeker is really a bargain. One can do ordinary multilevel analysis, multilevel survival analysis, categorical and ordinal outcomes, Poisson multilevel analysis, and all this in a uniform and modular way. The programs are fast, have little overhead, are versatile, well documented, and completely free. Moreover binaries are available for PowerMacs and for Sun Solaris machines. For most of the MIXFOO programs one needs to write batch setup files, but as we have indicated above, that is not really a major disadvantage. Market forces have conspired to make us believe that it is, and that graphical interfaces are the be-all and end-all of software, but most quality statistical and numerical software does not have and does not need such interfaces. Editing an existing text file is usually faster than wading through not very intuitive menus.

References

M. A. Aitkin and N. T. Longford. Statistical modeling issues in school effectiveness studies. *Journal of the Royal Statistical Society A*, 149: 1–43, 1986.

A. S. Bryk and S. W. Raudenbush. *Hierachical Linear Models for Social and Behavioral Research: Applications and Data Analysis Methods*. Sage Publications, Newbury Park, CA, 1991.

A. S. Bryk, S. W. Raudenbush, and Jr. R. T. Congdon. *HLM. Hierarchical Linear and Nonlinear Modeling with the HLM/2L and HLM/3L Programs*, 1996.

F.M.T.A. Busing, E. Meier, and R. van der Leeden. `MLA` software for multilevel analysis of data with two levels. Technical report, Department of Psyhcometrics and Research Methodology, University of Leiden, 1994.

J. de Leeuw and I. G. G. Kreft. Questioning multilevel models. *Journal of Educational and Behavioral Statistics*, 20:171–190, 1995.

W. J. Dixon and K. Merdian. *ANOVA and Regression with* `BMDP5-V`. Dixon Statistical Associates, Los Angeles, 1992.

W.J. Dixon. `BMDP` *Statistical Software Manual*. University of California Press, 1992.

H. Goldstein. Multilevel mixed linear model analysis using iterative generalized least squares. *Biometrika*, 73:43–56, 1986.

H. Goldstein. *Multilevel Models in Educational and Social Research*. Griffin, London, GB, 1987.

H. Goldstein. Restricted (unbiased) iterative generalized least squares estimation. *Biometrika*, 76:622–623, 1989.

H. Goldstein. *Multilevel Statistical Models*. Edward Arnold, London, GB, 1995.

H. Goldstein and J. Rasbash. Efficient computational procedures for the estimation of parameters in multilevel models based on iterative generalized least squares. *Computational Statistics and Data Analysis*, 13: 63–71, 1992.

H. Goldstein, J. Rasbash, I. Plewis, D. Draper, W. Browne, M. Yang, G. Woodhouse, and M. Healy. A user's guide to `MLwiN`. Technical report, Institute of Education, University of London, 1998.

D. Hedeker. *Random regression models with autocorrelated errors*. PhD thesis, University of Chicago, 1989.

D. Hedeker and R. D. Gibbons. A random effects ordinal regression model for multilevel analysis. *Biometrics*, 50:933–944, 1994.

D. Hedeker and R.D. Gibbons. MIXOR. a computer program for mixed-effects ordinal probit and logistic regression analysis. *Computer Methods and Programs in Biomedicine*, 49:157–176, 1996a.

D. Hedeker and R.D. Gibbons. MIXREG. a computer program for mixed-effects regression analysis with autocorrelated errors. *Computer Methods and Programs in Biomedicine*, 49:229–252, 1996b.

D. Hedeker, O. Siddiqui, and F.B. Hu. Random-effects regression analysis of correlated grouped-time survival data. Technical report, University of Illinois, Chicago, 1996.

R. Jennrich and M. Schluchter. Unbalanced repeated measures models with structured covariance matrices. *Biometrics*, 42:805–820, 1986.

I. G. G. Kreft and J. de Leeuw. *Introducing Multilevel Modelling*. Sage publications, London, Thousand Oaks, New Delhi, 1998.

I. G. G. Kreft, J. de Leeuw, and K.-S. Kim. Comparing four different statistical packages for hierarchical linear regression: GENMOD, HLM, ML2, VARCL. Preprint 50, UCLA Statistics, Los Angeles, CA, 1990.

I.G.G. Kreft, J. de Leeuw, and R. van der Leeden. Review of five multilevel analysis programs: BMDP-5V, GENMOD, HLM, ML3, VARCL. *American Statistician*, 48:324–335, 1994.

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

N. T. Longford. VARCL. software for variance component analysis of data with nested random effects (maximum likelihood). Technical report, Educational Testing Service, Princeton, NJ, 1990.

N.T. Longford. A quasi-likelihood adaptation for variance component analysis. In *Poceedings of the Section on Statistical Computing of the ASA*, 1988.

N.T. Longford. *Random Coefficient Models*. Oxford University Press, Oxford, GB, 1993.

J. Rasbash and Geoff Woodhouse. MLn Command Reference. Technical report, Institute of Education, University of London, 1995.

SAS. SAS/STAT software: Changes and enhancements. Technical report, SAS Institute, 1992.

M. Schluchter. BMDP5-V – Unbalanced Repeated Measures Models with Structured Covariance Matrices. Technical Report 86, BMDP Statistical Software, Los Angeles, CA, 1988.

J. Singer. Notes on using PROC MIXED. *Journal of Educational and Behavioral Statistics*, page ??, 1998.

BMDP Statistical Software. BMDP *User's Digest*. University of California Press, 1992.

R. van der Leeden, K. Vrijburg, and J. de Leeuw. A review of two different approaches for the analysis of growth data using longitudinal mixed linear models: Comparing hierarchical linear regression (ML3, HLM) and repeated measures design with structured covariance matrices (BMDP-5V). Preprint 98, UCLA Statistics, Los Angeles, 1991.

R. van der Leeden, K. Vrijburg, and J. de Leeuw. A review of two different approaches for the analysis of growth data using longitudinal mixed linear models. *Computational Statistics and Data Analysis*, 21:583–605, 1996.

G. Woodhouse. A guide to `MLn` for new users. Technical report, Multilevel Models Project, Instutute of Education, University of London, 1995.

M. Yang, J. Rasbash, and H. Goldstein. `MLwiN` macros for advanced multilevel modelling. Technical report, Institute of Education, University of London, 1998.

JAN DE LEEUW, UNIVERSITY OF CALIFORNIA AT LOS ANGELES
E-mail address: `deleeuw@stat.ucla.edu`

ITA G.G. KREFT, CALIFORNIA STATE UNIVERSITY LOS ANGELES
E-mail address: `kreft@stat.ucla.edu`