Fitting mixed-effects models in R (version 1.5.1)

1 A brief introduction to R

1.1 Background

R is a system for statistical computation and graphics developed initially by Ross Ihaka and Robert Gentleman at the Department of Statistics of the University of Auckland in Auckland, New Zealand Ihaka and Gentleman (1996). It can be regarded as an Open--Source implementation of the S language, which in turn underlies the S-Plus software.

R offers facilities for data manipulation, calculation and graphical display either through build-in functions or add-on packages contributed by users.

It is distributed freely under the GNU General Public License (www.gnu.org/copyleft/gpl.html) and can be used for commercial purposes. The latest version and documentation can be obtained via CRAN, the Comprehensive R Archive Network that can be found at the URL http://cran.r-project.org.

1.2 Operating System

R is available for Unix/Linux (most platforms/distributions including i386-freebsd, i386-linux-gnu, i386-sun-solaris, powerpc-linux-gnu, powerpc-apple-darwin, mips-sgi-irix, alpha-linux-gnu, alpha-dec-osf4, rs6000-ibm-aix, hppa-hp-hpux, sparc-linux-gnu, and sparc-sun-solaris), Windows (95, 98, ME, NT4, 2000 and XP) and MacOS (8.6 to 9.1 or X natively).

1.3 Data input/output functionality

R can read from and write to ASCII and text files (*.txt, *.dat), spreadsheet-like data (e.g. *.csv for input in Excel), read from fixed-width-format files using command-line functions like `scan`, `read.table`, `write.table`. Also, the contributed package `foreign` available on CRAN provides facilities for importing data from other statistical packages namely Minitab, Octave, S-Plus, SAS, SPSS, Stata and exporting to Stata.

Since all data have to reside in memory, there are certain limitations on the size of data sets that R can handle (not more than a few tens of megabytes). On the other hand, R has facilities for communicating with Database Management Systems (DBMS). In particular, the contributed packages `RPgSQL`, `RODBC`, `RMySQL` also available on CRAN provide interfaces to most common DBMSs.

R offers great graphical features. Mathematical symbols. Mathematical symbols can be easily added to a plot and the users have control over almost every aspect of the final output. Graphs can be saved as Postscript, Encapsulated Postscript or PDF files and incorporated into another document. Commands can be input interactively at the command-line or sourced from a previously saved file using the command `source` (any text editor can be used for this purpose and the file should be saved with extension *.R). The command history can be loaded, saved and displayed at any time using the build-in functions `loadhistory`, `savehistory` and `history` respectively.

1.4 Interface features

When R is launched, the user is presented with a console window for input/output and command line options (Windows and Macintosh). Under Unix-like systems, various editors can be used to add extra capabilities (e.g. Emacs Speaks Statistics or ESS). The GUI under Windows and Macintosh offers limited features and, in general, the only way of using R is through the command-line input/output. Loaded data and intermediate results are kept in a working directory that can be specified by the used. At the end of a session, the user can choose whether or not to save its contents.

R operates on named data structures like vectors, matrices and n-dimensional arrays, which can be created using assignment statements. Data manipulation is via build-in or add-on functions on these named structures; in particular, R offers operators for calculation on matrices and arrays and a simple programming language which included conditionals, loops and user defined functions.

In addition to basic tools for summarising data, a non-exhaustive list of models statistical models that can be fit using R includes linear models, ANOVA, generalized linear models, survival analysis, non-linear least-squares and maximum likelihood models and linear and non-linear mixed-effects models through the functions `lme` and `nlme` respectively in the user-contributed package `nlme`.

2 Packages for fitting mixed-effects models

2.1 Features and syntax

The package `nlme` by JC Pinheiro and DM Bates (2000} provides methods for fitting linear (function `lme`) and non-linear (function `nlme`) mixed-effects models assuming that both the random effects and the errors follow Gaussian distributions. Table 1 shows details of models that can be fitted using the package.

Once the data have been loaded in the working directory, a *grouped* version is constructed that captures the clustering structure. This is done using the `groupedData` function, which in the simple case of two-level data with `group` as the clustering factor uses the following syntax

groupedData(response~covariate(s) | group, data, *options*). (1)

A three level structure can be specified as `group1/group2` where `group2` is a grouping factor nested in `group1`, and so forth.

Having specified the grouping structure, there are then methods for plotting and summarizing the data by the grouping factor(s).

Fitting a linear mixed-effects model involves using the `lme` function on a grouped data object; by default, this includes the random effects implied by the structure in~(1) i.e. random terms associated with the intercept (which is always included and can be excluded using -1) and the covariate(s). Alternatively, random effects can be specified within the call to `lme`. For example, a simple two-level model with random intercept and slope across categories in `group` could be written as

lme(fixed=respons~covariate(s), random=~covariate(s) | group, *options*) (2)

The user can also specify the form for the variance-covariance matrix for the random effects choosing from a block-diagonal, compound-symmetry structure, diagonal, multiple of an identity and general positive-definite matrix or create their own.

The optional argument `correlation` allows specification of the within-group correlation structure choosing from 11 standard forms or by constructing one.

Similarly, the optional argument `weights` specifies the within-group heteroscedasticity structure with available forms as follows: exponential of a variance covariate, power of a variance covariate, constant plus power of a variance covariate, constant variance(s) (used to allow different variances according to the levels of a classification factor), fixed weights (determined by a variance covariate) or a combination of variance functions.

To fit a modified model, the function `update` can be used.

The syntax of the function `nlme` for non-linear mixed-effects models is similar and will be omitted.

Various packages are available for fitting generalized linear mixed models (GLMM) in R. The package GLMMGibbs available on CRAN uses a Bayesian approach and is limited to binomial and Poisson families with canonical link function.

Libraries accompanying Lindsay's book `Models for repeated measurements' (Lindsey,1999) are available at the URL http://alpha.luc.ac.be/~lucp0753/rcode.html and include functions for fitting GLMM.

A third option is the function `glmmPQL` available in the recommended library MASS (in the package bundle VR on CRAN). It uses the same syntax as in~(2) and parameter estimation is based on penalized quasi-likelihood (PQL) (Breslow,1993}. In this review, for non-Guassian outcomes, the latter function will be used.

2.2 Tools for statistical inference and model diagnostics

When using the function `lme`, models with different random effects specification can be compared using likelihood ratio tests or by simulation-based parametric bootstrap evaluations. The significance of fixed-effects terms is assessed by standard linear regression tests including *t*-test for individual coefficients or *F*-tests for complicated terms or linear combinations of coefficients.

The function `summary.lme`, which takes as argument the output of a call to `lme`, gives additional information on the model fit including the Akaike and Bayesian Information Criterion (not valid with `glmmPQL`} as the latter uses PQL).

Approximate confidence intervals for the fixed-effects and the variance-covariance parameters are produced by the command `intervals.lme`. Diagnostic plots include box plots of the residuals by level-1 units, scatter plots of standardized residuals versus fitted values possibly by the values of some categorical variable, observed values versus fitted values and normal plot of residuals.

Finally, predicted values at any level can be obtained using the function `predict.lme`.

A summary of available tools for inference and model diagnostics is given in Table 2.

3 Model specifications -- Basic models

3.1 Two-level Normal models

Using standard MLwiN notation, the collected subscripts *ij* indicate variables measured on the i^{th} level 1 unit clustered in the j^{th} level-2 unit. The data set `exam' in the MLwiN manual is used for illustration. The response variable consists of exam scores obtained by each student at age 16 normalised to have an approximate standard normal distribution (`normexam`). Covariates include:

x_{1ij} : school intake variable, standardised London reading test (`standlrt`);

x_{2ij} : gender of students with boys as the reference group (`gender` as factor);

x_{3j} : school gender for mixed school against girls' school (`schgend` as factor);

x_{4j} : school gender for boys' school against girls' school (`schgend` as factor).

Five models have been considered differing in both their fixed and random parts.

Results are shown in Table 3. Notice that convergence times refer to R version 1.5.1 for Mac OS X 10.1.5 running on PowerPC G4 700Mhz with 512MB of RAM.

A variance component model with covariates `standlrt, gender` and `schgend` can be written as

$$y_{ij} = \beta_{0j} + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3j} + \beta_4 x_{4j} + e_{0ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

where $u_{0j} \sim N(0, \sigma^2_{u_0})$ and $e_{0ij} \sim N(0, \sigma^2_{e_0})$. Having defined the variables `gender` and `schgend` as categorical variables and set the desired contrasts with the command

```
options(contrasts=c(factor="contr.treatment",ordered="contr.poly"))
```

the model is fitted using the expression

```
lme(normexam~standlrt+gender+schgend,random=~1|school,data=tutorial).
```

By default, REML estimates are given; maximum likelihood estimates can be obtained setting the option `method=`ML'` in the call to `lme`.

The variance component model with `standlrt*gender` interaction term

$$y_{ij} = \beta_{0j} + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3j} + \beta_4 x_{4j} + \beta_5 (x_{1ij} \times x_{2ij}) + e_{0ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

is fitted using the syntax

```
lme(normexam~standlrt*gender+schgend,random=-1 |school,data=tutorial)
```

or, equivalently

```
lme(normexam~standlrt+gender+standlrt:gender+schgend,random=-1|school,data=tutorial).
```

Random slopes on `standlrt` can be specified by modifying the formula for the random part as

```
lme(normexam~standlrt*gender+schgend,random=~standlrt |school,data=tutorial)
```

where the intercept term for school is implicit.

Level 1 variance functions can be specified with the `weights` option. Thus the model with random slopes on `standlrt` and variances that differ by gender of students

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3j} + \beta_4 x_{4j} + \beta_5 (x_{1ij} \times x_{2ij}) + e_{ij}$$

$$\beta_{0j} = \beta_0 + u_{0j}$$

$$\beta_{1j} = \beta_1 + u_{1j}$$

$$e_{ij} = \alpha_0 + \alpha_1 x_{2ij}$$

is fitted using the syntax

```
lme(normexam~standlrt*gender+schgend,random=~1 | school,
          weights=var | dent(form=~1 | gender),data=tutorial).
```

Many other build-in variance functions are available. For instance, a level 1 variance model

$$\delta_1 + | x_{1ij} |^{\delta_2}$$

corresponding to the variance function $(\delta_1 + | x_{1ij} |^{\delta_2})^2$ i.e. a constant plus a power of the absolute value of the variance covariate `standlrt` is fitted as

```
weights=varConstPower(form=~standlrt).
```

Notice that, for the `exam' dataset, $\delta_1$ was held fixed at 1 using the optional argument `fixed` of `varConstPower` implying that the level-1 variance is equal to $\sigma^2$ for values of `standlrt` close to zero and increases as a power of `|standlrt|` as `standlrt` increases in magnitude away from zero. In all cases, `lme` objects should be created using myfit< - *call to lme*. Then `summary.lme`(*myfit*), `plot.lme`(*myfit*) and `intervals.lme`(*myfit*) provide additional information on the model fit, diagnostic plots and approximate confidence intervals for all parameters, respectively. Each feature of the fitted model can be modified using the `update.lme`.

3.2 Three-level Normal models

The data used for illustration refer to A-level Chemistry point scores obtained by 31022 student with outer clustering represented by 2280 schools within 131 Local Education Authorities (LEA) in the example data set `chem97'. The only covariate is `gcsecnt`, the average GCSE score of students centered around its mean value. Two models have been considered and results are shown in Table~4. The syntax used in `lme` differs from the two-levels case in the expression used to specify the random part. Thus, for instance, the model with random terms on classes and schools

$$y_{ijk} = \beta_0 + v_{0k} + u_{0jk} + e_{0ijk}$$

using the syntax

```
lme(y~1, random=~1|LEA/school,data=chem97)
```

and one including a fixed effect for `gcsecnt` has `lme` syntax

```
lme(y~gcsecnt, random=~1|LEA/school,data=chem97).
```

Parameter estimates for both models are shown in Table 4.

3.3 Two-level models for binary data

Data come from the 1988 Bangladesh Fertility survey and consists of a subsample of 1934 women grouped in 60 districts in data set `mmmec'. The response variable is contraceptive use status at the time of survey (binary outcome `use` with 1 indicating women who used contraception and 0 otherwise). Covariates of interest were type of region of residence (variable `urban` with 1=urban, 0=rural), age centred around mean (`agecnt`) and number of living children (`livch` as factor with `none' as the baseline category).

Two-level models with logit or probit link functions have been fitted using the function `glmmPQL` with syntax

```
glmmPQL(use~urban+agecnt+livch,random=~1 | district,
        family=binomial(link=logit(or probit)),data=mmmec).
```

Parameter estimates are shown in Table 5.

3.4 Growth models for repeated measures data

Data consist in repeated measurements of height taken on 26 boys at 9 different time points in the example data set `oxboys'. The results from the fit of two different models are shown in Table 5. In particular, the polynomial growth curve

$$ht_{ij} = \beta_{0j} + \beta_{1j} t_{ij} + \beta_{2j} t_{ij}^2 + \beta_3 t_{ij}^3 + \beta_4 t_{ij}^4 + e_{ij}$$

$$\beta_{hj} = \beta_h + u_{hj}, \quad h=0,1,2$$

$$u_{hj} \sim MVN(0,\Omega_2), \quad e_{ij} \sim N(0,\sigma_e^2)$$

where t is the age of boys centred at mean, is fitted using the expression

```
lme(ht~age+age2+age3+age4,random=~age+age2 | id,data=oxboys).
```

The second model adds sine and cosine functions of π `*season/6`, `sinseas` and `cosseas` respectively, in the fixed part of the model where season is the season in decimal year and an autoregressive correlation structure of order 1 for the level 1 residuals

$$ht_{ij} = \beta_{0j} + \beta_{1j} t_{ij} + \beta_{2j} t_{ij}^2 + \beta_3 t_{ij}^3 + \beta_4 t_{ij}^4 + \beta_5 \sin_{ij} + \beta_6 \cos_{ij} + e_{ij}$$

$$\beta_{hj} = \beta_h + u_{hj}, \quad h=0,1,2$$

$$u_{hj} \sim MVN(0,\Omega_2), \; e_{ij} = \rho e_{(i-1)j} + \delta_{ij}$$

The syntax in `nlme` is

```
lme(ht~age+age2+age3+age4+sinseas+cosseas,
    random=~age+age2 | boy,corr=corAR1(),data=oxboys).
```

3.5 Cross-classification model

The data are on 3435 children who attended 148 primary schools and 19 secondary schools in Fife, Scotland in the example data set `XC'. The response variable consists of exam attainment at the age sixteen; children are cross-classified by the secondary school and the primary school they attended (indexed by *j* and *k* respectively). The only covariate considered is gender coded as 0 for boy and 1 for girl.

The model can be written as

$$y_{i(jk)} = \beta_0 + \beta_{1ij} x_{1ij} + u_j + u_k + e_{i(jk)}$$

$$u_j \sim N(0, \sigma^2_{u_j}),\ u_k \sim N(0, \sigma^2_{u_k}),\ e_{i(jk)} \sim N(0, \sigma^2_e)$$

In `lme`, crossed random-effects structures are represented and fitted as random-effects structures corresponding to a two-level model with a block-diagonal variance-covariance matrix with blocks corresponding to the cross-classifying effects (primary and secondary school in our case). First, a `groupedData` version of the data set is defined using the expression

```
XCgroupedData <-groupedData(attain~sex | cons,data=XC)
```

where `cons` is a 3435 $\times 1$ column vector of ones. The call to `lme` is then

```
lme(attain~sex,random=pdBlocked(list(pdIdent(~pid-1),pdIdent(~sid-1))),data=XCgroupedData)
```

where in the expression above, `pid` and `sid` contain identifying codes for primary and secondary schools respectively.

Parameter estimates for this model are reported in Table 7.

3.6 Multivariate Normal response model

Multivariate mixed-effects models can be fitted using ad-hoc manipulation of the response vectors and design matrix, namely by stacking the former (within each cluster unit) and expanding the latter accordingly with extra columns of dummy covariates flagging each element of the original multivariate response.

The data used for illustrative purposes consist of GCSE exam scores on a science subject obtained by 1905 students from 73 schools in England. A bivariate outcome was considered consisting of written paper and course work scores. Gender of student was the only covariate included (0= boy, 1= girl).

The model can be written as

$$y_{1jk} = \beta_0 + \beta_1 x_{jk} + v_{1k} + u_{1jk}$$

$$y_{2jk} = \alpha_0 + \alpha_1 x_{jk} + v_{2k} + u_{2jk}$$

$$\begin{pmatrix} v_{1k} \\ v_{2k} \end{pmatrix} \sim MVN(0, \Omega_v):\ \Omega_v = \begin{pmatrix} \sigma^2_{v_1} & \sigma_{v_{12}} \\ \sigma_{v_{21}} & \sigma^2_{v_2} \end{pmatrix}$$

$$\begin{pmatrix} u_{1jk} \\ u_{2jk} \end{pmatrix} \sim MVN(0, \Omega_u):\ \Omega_u = \begin{pmatrix} \sigma^2_{u_1} & \sigma_{u_{12}} \\ \sigma_{u_{21}} & \sigma^2_{u_2} \end{pmatrix};$$

a possible way of specifying this model in `lme` is as follows

```
lme(score~-1+wtn+cwk+I(wtn*gender)+I(cwk*gender),random=~-1+wtn+cwk|school,
weights=varIdent(form=~1| wtn),corr=corCompSymm(form=~1 | school/student),data=gcseex)
```

where `gcseex` is the expanded data set obtained by stacking the two responses for each student, the term -1 in the fixed and random parts of the model is necessary to remove the overall intercept term fitted by default, the fixed effects `wtn` and `cwk` represent the population average scores for boys and `I(wtn*gender)` and `I(csw*gender)` are appropriate contrast terms for girls.

Results are shown in Table 8.

3.7 Documentation

Documentation for R includes an online help for most of the functions, available also as one reference manual for on-line reading in HTML and PDF formats, five manuals that came with the installation (An Introduction to R, Writing R extensions, R data import/export, The R language definition, R installation and administration). Furthermore, there are many books available for S/Splus, which can be used in conjunction with R. In particular WN Venables and BD Ripley provide an `R Complements' to their book *Modern Applied Statistics with S-PLUS*, Third Edition, Springer, 1999 at the URL http://www.stats.ox.ac.uk/pub/MASS3/Sprog/ (the fourth edition released in August 2002 covers R as well). The library MASS accompanying the book comes with the current version of R on CRAN. At a more advanced level, the book *S Programming*, Springer, 2000 by the same authors provides a guide to writing software using the S language with R-specific variants.

Finally, there is a very active mailing list called R-help (details on how to subscribe are available on the R-project website).

4 Conclusions

R is a fast open--source clone of S-Plus that works on multiple computer platforms offering excellent data handling and graphical display features. It does however have a GUI with very limited capabilities and therefore requires some basic knowledge of command line input/output to get started.

Mixed effect models for Gaussian outcome variables can be fitted in R using the package `nlme` freely available on CRAN and included in the current full distribution of the software (1.5.1). Models with more than two levels of nesting, cross-classified random effects as well as multivariate normal models can be fitted (the latter with ad-hoc manipulation of the response vectors and design matrix). The package is comprehensively documented in the book `Linear and nonlinear mixed-effects models in S and S-Plus' by Pinheiro and Bates (2000).

Generalized linear mixed models can be fitted using the function `glmmPQL` in the library `MASS` of the package bundle `VR` also available on CRAN. The syntax used for model specification is as in `lme` and parameter estimates are based on PQL. The function is documented in the fourth edition of the book `Modern Applied Statistics with S' by Venables and Ripley (2002).

R is freely distributed under the GNU General Public License and available at the URL http://www.r-project.org.

References

Breslow, N.E. and Clayton, D.G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, **88,** 9-25.

Ihaka, R. and Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5**, 299-314.

Lindsey, J.K. (1999). *Models for Repeated Measurements*. Oxford University Press.

Pinheiro, J. and Bates, D.M. (2000). *Linear and nonlinear mixed-effects models in S and S-Plus*. New York: Springer-Verlag.

Venables, W.N. and Ripley, B.D. (2002). *Modern Applied Statistics with S*. New York: Springer-Verlag.

Table 1: Random effects models that can and cannot be fitted in R using functions **lme**, **nlme** and **glmmPQL**.

Data/model type	Estimation algorithm	Max number of levels in data	Covariates	Random slopes	Weighting	Fitting variance function	Estimates available
Normal response, Repeated Measures, Cross-classified, Multivariate Normal (function **lme**)	Expectation-Maximisation (EM) iterations followed by Newton-Raphson.	Arbitrary	Yes	Yes	No	Yes (built-in and used-defined).	Maximum Likelihood or Restricted Maximum Likelihood.
Binary/Binomial, Poisson, Gamma, Inverse Gaussian (function **glmmPQL** in library MASS)	PQL coupled with call to estimation algorithm of **lme**	Arbitrary	Yes	Yes	Yes	No	PQL estimates.
Negative Binomial	—	—	—	—	—	—	—
Nominal multinomial	—	—	—	—	—	—	—
Ordered multinomial	—	—	—	—	—	—	—
Multiple membership	—	—	—	—	—	—	—
Survival data	—	—	—	—	—	—	—
Time series	—	—	—	—	—	—	—
Multivariate mixed responses	—	—	—	—	—	—	—
Nonlinear (function **nlme**)	Alternates between a penalized nonlinear least squares step and a linear-mixed-effects step (approximates the marginal likelihood of the response by that of a linear-mixed-effects model).	Arbitrary	Yes	Yes	No	Yes (built-in and used-defined).	Maximum Likelihood or Restricted Maximum Likelihood.
Structural Equation model	—	—	—	—	—	—	—

Table 2: Tools for inference and diagnostics.

Models that can be fitted	Tests for overall goodness of fit	Inference on fixed effects	Inference on random effects	Diagnostics	Other specific features
Function **lme**: Normal response, Repeated Measures, Cross-classified, Multivariate Normal	Likelihood ratio tests for nested models, Akaike and Bayesian Information criterions.	*t*-tests for individual coefficients and *F*-tests for more complicated terms or linear combinations of coefficients; confidence intervals based on approximate distributions of ML or REML estimates.	Likelihood ratio tests or simulation-based parametric bootstrap; approximate confidence intervals for variance-covariance parameters.	Fitted values; Residuals; Predicted values; Normal probability plots; Semivariogram of within-group residuals; Various other diagnostic plots to assess assumptions for random effects.	Patterned variance-covariate matrices for the random effects and various correlation structures can be specified; Custering can be accounted for by specifying correlation structures and variance functions in the **gls** function (many built-in structures are available and the user can define his own); Function lmList fits separate linear regression models to data according to the levels of a grouping factor.
Function **nlme**: Nonlinear mixed-effects models	Likelihood ratio tests for nested models, Akaike and Bayesian Information criterions.	*t*-tests for individual coefficients and *F*-tests for more complicated terms or linear combinations of coefficients; confidence intervals based on approximate distributions of ML or REML estimates.	Likelihood ratio tests and confidence intervals for variance-covariance parameters.	Fitted values; Residuals; Predicted values; Normal probability plots; Semivariogram of within-group residuals.	Patterned variance-covariate matrices for the random effects and various correlation structures can be specified; Custering can be accounted for by specifying correlation structures and variance functions in the **gnls** function (many built-in structures are available and the user can define his own); Function nlsList fits separate nonlinear regression models to data according to the levels of a grouping factor.
Function **glmmPQL** (library MASS): Generalized linear mixed models		*t*-tests for individual coefficients; confidence intervals based on approximate distributions.	Approximate confidence intervals for variance-covariance parameters.	Fitted values; Residuals; Predicted values.	Weights can be attached to level-one units. Optional correlation structures can be specified.

Table 3: Parameter estimates for two-level Normal models (using **lme**).

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik
Variance component with covariates 'standlrl', 'gender' and 'schgend' (0=girls', 1=mixed, 2=boys')	*Fixed*			2	9347.67
	Intercept	β_0	-0.009(-0.162,0.143)		
	standlrt	β_1	0.560(0.535,0.584)		
	gender	β_2	0.167(0.100,0.234)		
	Schgend (mixed vs girls')	β_3	-0.159(-0.338,0.020)		
	Schgend (boys' vs girls')	β_4	0.019(-0.233,0.270)		
	Random				
	school	σ_{0u}	0.293(0.239,0.359)		
	Residual	σ_e	0.750(0.734,0.767)		
Variance component with 'standlrt'*'gender' interaction	*Fixed*			2	9353.20
	Intercept	β_0	-0.009(-0.162,0.143)		
	standlrt	β_1	0.562(0.526,0.598)		
	gender	β_2	0.167(0.100,0.234)		
	Schgend (mixed vs girls')	β_3	-0.159(-0.337,0.020)		
	Schgend (boys' vs girls')	β_4	0.019(-0.233,0.271)		
	Standlrt:gender	β_5	-0.005(-0.053,0.043)		
	Random				
	school	σ_{u0}	0.293(0.239,0.360)		
	Residual	σ_{e0}	0.750(0.734,0.767)		
Random slopes on standlrt	*Fixed*			4	9308.24
	Intercept	β_0	-0.012(-0.157,0.133)		
	standlrt	β_1	0.550(0.500,0.601)		
	gender	β_2	0.168(0.102,0.235)		
	Schgend (mixed vs girls')	β_3	-0.178(-0.342,-0.014)		
	Schgend (boys' vs girls')	β_4	0.000(-0.233,0.232)		
	Standlrt:gender	β_5	0.007(-0.051,0.065)		
	Random				
	School,standlrt	σ_{u0}	0.289(0.236,0.355)		
		σ_{u1}	0.123(0.091,0.167)		
		ρ_{u10}	0.574(0.234,0.789)		
	Residual	σ_{e0}	0.742(0.726,0.758)		

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik
Level 1 variances by 'gender'	*Fixed*			30	9302.21
	Intercept	β_0	-0.012(-0.157,0.133)		
	standlrt	β_1	0.550(0.499,0.601)		
	gender	β_2	0.169(0.102,0.235)		
	Schgend (mixed vs girls')	β_3	-0.178(-0.341,-0.015)		
	Schgend (boys' vs girls')	β_4	0.000(-0.234,0.232)		
	Standlrt:gender	β_5	0.007(-0.051,0.065)		
	Random				
	School,standlrt	σ_{u0}	0.289(0.236,0.355)		
		σ_{u1}	0.124(0.092,0.168)		
		ρ_{u10}	0.576(0.238,0.790)		
	Residual	σ_{e0}	0.725(0.704,0.746)		
		σ_{e2}	0.767(0.732,0.803)		
Level 1 variance as function of 'standlrt'	*Fixed*			35	9308.24
	Intercept	β_0	-0.012(-0.157,0.134)		
	standlrt	β_1	0.550(0.500,0.601)		
	gender	β_2	0.168(0.102,0.235)		
	Schgend (mixed vs girls')	β_3	-0.178(-0.342,-0.014)		
	Schgend (boys' vs girls')	β_4	0.000(-0.233,0.232)		
	Standlrt:gender	β_5	0.007(-0.051,0.065)		
	Random				
	School,standlrt	σ_{u0}	0.289(0.236,0.355)		
		σ_{u1}	0.123(0.091,0.167)		
		ρ_{u10}	0.576(0.240,0.789)		
	Residual	σ	0.371(0.362,0.381)		
		δ_2	0.001(-0.037,0.039)		

Table 4: Parameter estimates for three-level Normal models (using **lme**)

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik
Variance component without covariates	*Fixed*			25	157873.8
	Intercept	β_0	5.319(5.205,5.433)		
	Random				
	School (level 2)	σ_{u0}	1.658(1.589,1.729)		
	LEA (level 3)	σ_{v0}	0.394(0.279,0.555)		
	Residual	σ_{e0}	2.918(2.894,2.942)		
Variance component with covariate 'gcsescore'	*Fixed*			25	141696.9
	Intercept	β_0	5.635(5.574,5.696)		
	gcsescore	β_1	2.472(2.439,2.506)		
	Random				
	School (level 2)	σ_{u0}	1.080(1.035,1.127)		
	LEA (level 3)	σ_{v0}	0.120(0.049,0.294)		
	Residual	σ_{e0}	2.270(2.252,2.289)		

Table 5: Parameter estimates for two-level models for binary data (using **glmmPQL**)

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik (PQL)
Random intercept across districts (logit link)	*Fixed*			5	8488.624
	Intercept	β_0	-1.660(-1.941,-1.380)		
	Urban	β_1	0.719(0.491,0.948)		
	Age	β_2	-0.026(-0.041,-0.011)		
	1 liv. children vs 0	β_3	1.092(0.790,1.394)		
	2 liv. children vs 0	β_4	1.354(1.021,1.688)		
	>3 liv. children vs 0	β_5	1.324(0.982,1.666)		
	Random				
	District (level 2)	σ_{u0}	0.457(0.330,0.632)		
	Residual	σ_{e0}	0.984(0.953,1.016)		
Random slope on 'urban' across districts (logit link)	*Fixed*			29	8519.092
	Intercept	β_0	-1.666(-1.967,-1.365)		
	Urban	β_1	0.791(0.470,1.113)		
	Age	β_2	-0.026(-0.041,-0.011)		
	1 liv. children vs 0	β_3	1.099(0.796,1.401)		
	2 liv. children vs 0	β_4	1.334(1.000,1.668)		
	>3 liv. children vs 0	β_5	1.323(0.979,1.666)		
	Random				
	Rural	σ_{u0}	0.608(0.448,0.825)		
	Urban	σ_{u1}	0.797(0.509,1.250)		
		ρ_{u10}	-0.793(-0.931,-0.456)		
	Residual	σ_{e0}	0.976(0.945,1.008)		
Random intercept across districts (probit link)	*Fixed*			6	6554.516
	Intercept	β_0	-1.021(-1.187,-0.854)		
	Urban	β_1	0.445(0.306,0.585)		
	Age	β_2	-0.016(-0.025,-0.007)		
	1 liv. children vs 0	β_3	0.665(0.483,0.848)		
	2 liv. children vs 0	β_4	0.829(0.627,1.030)		
	>3 liv. children vs 0	β_5	0.809(0.603,1.014)		

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik (PQL)
	Random				
	District (level 2)	σ_{u0}	0.280(0.204,0.386)		
	Residual	σ_{e0}	0.985(0.954,1.017)		
Random slope on 'urban' across districts (probit link)	*Fixed*			25	6576.054
	Intercept	β_0	-1.029(-1.209,-0.850)		
	Urban	β_1	0.494(0.296,0.692)		
	Age	β_2	-0.016(-0.025,-0.007)		
	1 liv. children vs 0	β_3	0.674(0.491,0.856)		
	2 liv. children vs 0	β_4	0.821(0.619,1.023)		
	>3 liv. children vs 0	β_5	0.815(0.609,1.022)		
	Random				
	Rural	σ_{u0}	0.373(0.276,0.506)		
	Urban	σ_{u1}	0.493(0.315,0.771)		
		ρ_{u10}	-0.797(-0.932,-0.464)		
	Residual	σ_{e0}	0.977(0.946,1.009)		

Table 6: Parameter estimates for repeated measures ‘growth’ models (using **lme**).

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik
Polynomial growth model (up to quartic term) with random coefficients (up to quadratic) without time series	*Fixed*			10	629.82
	Intercept	β_0	148.975(145.879,152,070)		
	age	β_1	6.166,5.461,6.870)		
	age^2	β_2	1.090(0.395,1.785)		
	age^3	β_3	0.467(0.145,0.790)		
	age^4	β_4	-0.340(-0.936,0.255)		
	Random				
	Intercept, age, age^2	σ_{u0}	8.000(6.062,10.558)		
		σ_{u1}	1.695(1.277,2.248)		
		σ_{u2}	0.8125(0.571,1.157)		
		ρ_{u10}	0.613(0.305,0.804)		
		ρ_{u20}	0.218(-0.220,0.583)		
		ρ_{u21}	0.660(0.289,0.859)		
	Residual	σ_e	0.470(0.420,0.525)		
The model above plus sine and cosine functions in the fixed part and autocorrelation structure for level 1 residuals of order 1	*Fixed*			12	623.554
	Intercept	β_0	148.873(145.778,151.967)		
	age	β_1	6.174(5.461,6.886)		
	age^2	β_2	2.061(1.148(2.973)		
	age^3	β_3	0.405(0.051,0.760)		
	age^4	β_4	-1.432(-2.310,-0.555)		
	sin(pi*season/6)	β_5	0.015(-0.088,0.118)		
	cos(pi*season/6)	β_6	-0.223(-0.356,-0.089)		
	Random				
	Intercept, age, age^2	σ_{u0}	7.993(6.053,10.553)		
		σ_{u1}	1.664(1.250,2.222)		
		σ_{u2}	0.764(0.506,1.155)		
		ρ_{u10}	0.618(0.307,0.810)		
		ρ_{u20}	0.261(-0.212,0.635)		
		ρ_{u21}	0.702(0.265,0.900)		
	Residual	ρ	0.239(-0.074,0.509)		
		σ_δ	0.507(0.407,0.630)		

Table 7: Parameter estimates for cross-classified models (using **lme**).

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik
Attainment scores of students cross-classified by primary and secondary school with covariate 'gender' (0=boy, 1=girl)	*Fixed*			198	17127.91
	Intercept	β_0	5.255(4.893,5.616)		
	gender	β_1	0.498(0.306,0.691)		
	Random				
	Primary school	σ_{uj}	1.053(0.934,1.187)		
	Secondary school	σ_{uk}	0.608(0.386,0.958)		
	Residual	σ_e	2.838(2.770,2.907)		

Table 8: Parameter estimates for the multivariate Normal response model (using **lme**).

Model	Parameters		Estimates (95% CI)	Seconds to convergence	-2*loglik
Bivariate model for written paper and course work scores by gender of student	*Fixed*			68	26794.58
	Intercept wtn (boy)	β_0	49.010(47.171,50.849)		
	Intercept csw (boy)	α_0	69.621(67.308,71.934)		
	Gender wtn (girl)	β_1	-2.491(-3.590,-1.392)		
	Gender csw (girl)	α_1	6.757(5.442,8.072)		
	Random				
	School wtn	σ_{v1}	6.883(5.650,8.385)		
	School csw	σ_{v2}	8.743(7.218,10.590)		
	School corr wtn,csw	ρ_{v12}	0.421(0.182,0.614)		
	Residual				
	wtn	σ_{u1}	11.158(10.784,11.544)		
	csw	σ_{u2}	13.423(12.862,14.007)		
	Corr wtn,csw	ρ_{u12}	0.486(0.446,0.523)		