Statistics 196A, Education 401D **Spring 2021**

Multilevel Modeling Using R

Spring 2021 Remote Asynchronous Instruction

Instructor. David Rogosa (usually)Sequoia 224, rag{AT}stanford{DOT}edu
Course web page: http://rogosateaching.com/stat196//

Course Welcome and Logistics (first day stuff, posted in February, call it Week0)
Lecture slides, week 0 (pdf) **Audio companion, week 0**
For recreation of in-classroom experience, linked below are youtube versions of the music I play
before starting lecture and after lecture concludes. Some may wish to reverse that ordering.

To see full course materials from Spring 2020 go here

From explorecourses

```
  STATS 196A (EDUC 401D): Multilevel Modeling Using R
Multilevel data analysis examples using R.
Topics include: two-level nested data, growth curve modeling, generalized linear models for counts and categorical data,
nonlinear models, three-level analyses. For more information, see course website: http://rogosateaching.com/stat196/
Terms: Spr | Units: 1 | Grading: Satisfactory/Unsatisfactory  Remote: Asyncrhonous
Instructors: Rogosa, D. (PI)
STATS 196A | 1 units | Class # 21356 | Section 01 | Grading: Satisfactory/No Credit | WKS |


 Course Schedule, Remote Asynchronous
Four (2hr) classes (formerly located in  Sequoia 200)
 Weeks 1 - 4.
    a. Introduction: Basic analyses for two-level nested data, normal models  (UK Exam data)
    b. Additional two-level (normal) models: experimental designs (Dyestuff), longitudinal data (growth curves, sleepstudy),
        observational data (High School and Beyond)
    c. Generalized linear mixed models for counts and categorical outcomes
    d. Three-level analyses (nested data and longitudinal data)
    e. Specialized applications (as time permits):  regression diagnostics, power calculations and design, ecological inference, survival analysis,
        nonlinear functional forms, mediation analysis, propensity scores and matching, imputation, item response theory
 Week 5. Substitute for live "Student presentations of multilevel data analyses" by online submission of materials
```

For the 1-unit enrollment in this short course, students are expected to engage in (i.e. consume) the four presentation class sessions.
For the fifth session, formerly each student made a short (~10 min) presentation of a relevant data analysis they have conducted.
In Spring 2021, as it was in Spring 2020, students instead submit the small project online (such as using rpubs or google drive). Refer to Week0 audio.

Core Sources
Many of the example presented in this short course are described in Examples from Multilevel Software Comparative Reviews Douglas Bates.
Code version of MlmSoftRev R-package containing mlmRev data examples. Bates talk on mlmRev U Bristol documentation
Additional examples in core package lme4
Another set of examples: lmer for SAS PROC MIXED Users Douglas Bates Department of Statistics University of Wisconsin Madison
Data sets from SAS System for Mixed Models
Overviews and additional examples from Doug Bates:
lme4: Mixed-effects modeling with R February 17, 2010 Springer (book chapters).
A merged updated version of Bates book lme4: Mixed-effects modeling with R May 2020
R Journal intro Fitting linear mixed models in R Using the lme4 package Douglas Bates (pp.27-30)
Collection of all Doug Bates lme4 talks
lme4 vignette: Douglas Bates Martin Machler Ben Bolker. Fitting linear mixed-effects models using lme4, *Journal of Statistical Software*
Technical topics: Mixed models in R using the lme4 package Part 4: Theory of linear mixed models
HSB and growth curve examples in John Fox lme tutorial
Another nice lmer exposition with life sciences examples: Mixed-effects models, Remko Duursma, Jeff Powell Hawkesbury Institute for the Environment, Western Sydney University. September 2016. HIE Datasets

Current version of R is R version 4.0.4 released on 2021-02-15. For references and software: The R Project for Statistical Computing
Berkeley mirror is no longer, choose a mirror from the main R page (first link, I use TN).

a. Respiratory clinical trial from HSAUR. lmList does logistic, introducing glmer lmList, glmer for respiration data (placebo group)
b. Contraception (Bangladesh) use from Bates review Rogosa R-session glmer model slide
c. Test scores (pass/fail outcome) from Ch 8, Multilevel Modeling Using R. Rogosa R-session

2. Count outcome, GLMM poisson models.
a. Count data: Contagious bovine pleuropneumonia, data(cbpp) in lme4.
Rogosa R-session herd plots
b. Factorial design, Count outcome. From HIE Sydney. EucFACE ground cover data Rogosa glmer session.
c. Another count data example from mlmRev package,
data(Mmmec): Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.
Rogosa glmer session more Rogosa Mmmec session

`glmer.nb` Fits a generalized linear mixed-effects model (GLMM) for the negative binomial family, building on glmer, and initializing via theta.ml from MASS.

3. hglm -- a different package for fitting hierarchical generalized linear models.
R Journal December 2010. manual vignette

Addendum. A little more on Overdispersion in Generalized Linear Mixed Models
pdf slides companion audio Rogosa R-session
Resources: Basic Overdispersion Overview: Overdispersion, and how to deal with it in R
Overdispersion in Github GLMM Faq (Ben Bolker)
R-package `RVAideMemoire` function `overdisp.glmer` man page 1 2

Week 4 April 28 2021

Lecture slides, week 4 (pdf)
Audio companion, week 4
parta partb

1. Three-level (and above) lmer examples.
Measured outcomes
a. Achieve data from *Multilevel Modeling Using R* book. Rogosa R-session
b. example from mlmRev package, data(Chem97): Scores on A-level Chemistry in 1997. Rogosa R-session
Count outcome.
c. data(Mmmec): Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure. Rogosa 3-level session

2. Missing Data and Imputation Methods for Multilevel Data Analysis
Vignette: Analyzing Imputed Data with Multilevel Models and merTools
Rogosa R-session for vignette
Missing data wide-form imputation: `mice` multiple regression example, nhanes data in package `mice` R-session using mice package
Missing data background. Multiple Imputation. Nhanes data example (mice primer) in van Buuren S and Groothuis-Oudshoorn K (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67.
See also Flexible Imputation of Missing Data. Stef van Buuren Chapman and Hall/CRC 2012. Chapter 9, Longitudinal Data Sec 3.8 Multilevel data. He is the originator of `mice` book extras
R resources. Multivariate Analysis Task View, *Missing data* section, esp packages `mice`
New package: `hmi`: hierarchical multiple imputation, vignette

Addendum. Study Design (power) for Mixed Models
Most activity and resources are for longitudinal designs (c.f Week2, topic 2A): How often and How many subjects?
In my Stat222 course longitudinal experimental design is taken up in Week 5, Lecture topic 4.
Audio for the topic starts at partc that week, with the lecture content for design starting at pdf page 93.
Resources for that Stat222, Week5 content Power Calculations for Longitudinal Group Comparsions.
R-package `longpower` Vignettes found by "browseVignettes(package = "longpower")" . Functions in MBESS package--`ss.power.pcm`.
R-package: powerlmm
Background pubs: Power for linear models of longitudinal data with applications to Alzheimer's Disease Phase II study design Michael C. Donohue, Steven D. Edland, Anthony C. Gamst
Sample Size Planning for Longitudinal Models: Accuracy in Parameter Estimation for Polynomial Change Parameters Ken Kelley Notre Dame Joseph R. Rausch *Psychological Methods* 2011
Additional Resources: basic R analogues, `power.t.test` `power.anova.test`
more `powerlmm`
intro: Introducing 'powerlmm' an R package for power calculations for longitudinal multilevel models
vignette: Power Analysis for Two-level Longitudinal Models with Missing Data
Simulation approaches to mixed-model power calculations:
Power for Multilevel Analysis Power Analysis in R for Multilevel Models Simulation approaches for mixed-model power calculations

`data` an optional data frame containing the variables named in `formula`. By default the variables are taken from the environment from which `lmer` is called. While `data` is optional, the package authors *strongly* recommend its use, especially when later applying methods such as `update` and `drop1` to the fitted model (*such methods are not guaranteed to work properly if* `data` *is omitted*). If `data` is omitted, variables will be taken from the environment of `formula` (if specified as a formula) or from the parent frame (if specified as a character vector).

`family` a GLM family, see `glm` and `family`.

`control` a list (of correct class, resulting from `lmerControl()` or `glmerControl()` respectively) containing control parameters, including the nonlinear optimizer to be used and parameters to be passed through to the nonlinear optimizer, see the `*lmerControl` documentation for details.

`start` a named list of starting values for the parameters in the model, or a numeric vector. A numeric `start` argument will be used as the starting value of `theta`. If `start` is a list, the `theta` element (a numeric vector) is used as the starting value for the first optimization step (default=1 for diagonal elements and 0 for off-diagonal elements of the lower Cholesky factor); the fitted value of `theta` from the first step, plus `start[["fixef"]]`, are used as starting values for the second optimization step. If `start` has both `fixef` and `theta` elements, the first optimization step is skipped. For more details or finer control of optimization, see `modular`.

`verbose` integer scalar. If `> 0` verbose output is generated during the optimization of the parameter estimates. If `> 1` verbose output is generated during the individual penalized iteratively reweighted least squares (PIRLS) steps.

`nAGQ` integer scalar - the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log-likelihood. Defaults to 1, corresponding to the Laplace approximation. Values greater than 1 produce greater accuracy in the evaluation of the log-likelihood at the expense of speed. A value of zero uses a faster but less exact form of parameter estimation for GLMMs by optimizing the random effects and the fixed-effects coefficients in the penalized iteratively reweighted least squares step. (See Details.)

`subset` an optional expression indicating the subset of the rows of `data` that should be used in the fit. This can be a logical vector, or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.

`weights` an optional vector of 'prior weights' to be used in the fitting process. Should be `NULL` or a numeric vector.

`na.action` a function that indicates what should happen when the data contain `NA`s. The default action (`na.omit`, inherited from the 'factory fresh' value of `getOption("na.action")`) strips any observations with any missing values in any variables.

`offset` this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of cases. One or more `offset` terms can be included in the formula instead or as well, and if more than one is specified their sum is used. See `model.offset`.

`contrasts` an optional list. See the `contrasts.arg` of `model.matrix.default`.

mustart	optional starting values on the scale of the conditional mean, as in glm; see there for details.
etastart	optional starting values on the scale of the unbounded predictor as in glm; see there for details.
devFunOnly	logical - return only the deviance evaluation function. Note that because the deviance function operates on variables stored in its environment, it may not return *exactly* the same values on subsequent calls (but the results should always be within machine tolerance).
...	other potential arguments. A method argument was used in earlier versions of the package. Its functionality has been replaced by the nAGQ argument.

Details

Fit a generalized linear mixed model, which incorporates both fixed-effects parameters and random effects in a linear predictor, via maximum likelihood. The linear predictor is related to the conditional mean of the response through the inverse link function defined in the GLM family.

The expression for the likelihood of a mixed-effects model is an integral over the random effects space. For a linear mixed-effects model (LMM), as fit by lmer, this integral can be evaluated exactly. For a GLMM the integral must be approximated. The most reliable approximation for GLMMs is adaptive Gauss-Hermite quadrature, at present implemented only for models with a single scalar random effect. The nAGQ argument controls the number of nodes in the quadrature formula. A model with a single, scalar random-effects term could reasonably use up to 25 quadrature points per scalar integral.

Value

An object of class merMod (more specifically, an object of *subclass* glmerMod) for which many methods are available (e.g. methods(class="merMod"))

See Also

lmer (for details on formulas and parameterization); glm for Generalized Linear Models (*without* random effects). nlmer for nonlinear mixed-effects models.

glmer.nb to fit negative binomial GLMMs.

Examples

```
## generalized linear mixed model
library(lattice)
xyplot(incidence/size ~ period|herd, cbpp, type=c('g','p','l'),
       layout=c(3,5), index.cond = function(x,y)max(y))
(gm1 <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
              data = cbpp, family = binomial))
## using nAGQ=0 only gets close to the optimum
(gm1a <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
               cbpp, binomial, nAGQ = 0))
## using nAGQ = 9 provides a better evaluation of the deviance
## Currently the internal calculations use the sum of deviance residuals,
## which is not directly comparable with the nAGQ=0 or nAGQ=1 result.
```

Mmmec *Malignant melanoma deaths in Europe*

Description

Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.

Usage

```
data(Mmmec)
```

Format

A data frame with 354 observations on the following 6 variables.

nation a factor with levels `Belgium`, `W.Germany`, `Denmark`, `France`, `UK`, `Italy`, `Ireland`, `Luxembourg`, and `Netherlands`

region Region ID - a factor.

county County ID - a factor.

deaths Number of male deaths due to MM during 1971–1980

expected Number of expected deaths.

uvb Centered measure of the UVB dose reaching the earth's surface in each county.

Source

http://multilevel.ioe.ac.uk/softrev/mmmec.html

References

Langford, I.H., Bentham, G. and McDonald, A. 1998: Multilevel modelling of geographically aggregated health data: a case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41-58.

Examples

```
str(Mmmec)
summary(Mmmec)
(fm1 <- glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)))
```

<- Back to Examples from Multilevel Modelling Software Review (/cran/mlmRev/)

Malignant melanoma deaths in Europe

Description

Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.

Usage

```
data(Mmmec)
```

Format

A data frame with 354 observations on the following 6 variables.

nation
a factor with levels `Belgium`, `W.Germany`, `Denmark`, `France`, `UK`, `Italy`, `Ireland`, `Luxembourg`, and `Netherlands`

region
Region ID - a factor.

county
County ID - a factor.

deaths
Number of male deaths due to MM during 1971–1980

expected
Number of expected deaths.

uvb
Centered measure of the UVB dose reaching the earth's surface in each county.

Source

http://multilevel.ioe.ac.uk/softrev/mmmec.html (http://multilevel.ioe.ac.uk/softrev/mmmec.html)

References

```
45 70 17 29 33 39 48 55 59 65 66 49 72 44
 7  7  8  8  8  8  8  8  9  9  9 12 12 13

> # two level analysis: counties within region (ignoring country for now)
> fm1 = glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base "expected"
> summary(fm1)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 | region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2256.4   2268.0  -1125.2   2250.4      351

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6593 -0.7340 -0.0528  0.6122  4.2276

Random effects:
 Groups Name        Variance Std.Dev.
 region (Intercept) 0.1697   0.4119
Number of obs: 354, groups:  region, 78

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.13859    0.04934  -2.809 0.004971 **
uvb         -0.03443    0.01002  -3.435 0.000593 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb 0.047

> exp(fixef(fm1)) # so here mult effect of uvb is to reduce mel?
(Intercept)         uvb
  0.8705813   0.9661564

> exp(confint(fm1))
Computing profile confidence intervals ...
                2.5 %    97.5 %
.sig01      1.4146433 1.6428082
(Intercept) 0.7888942 0.9597464
uvb         0.9477699 0.9861602

# base (unconditional) model without uvb
> fm0 = glmer(deaths ~ 1 + (1|region), Mmmec, poisson, offset = log(expected))
> summary(fm0)
```

```
> fivenum(deaths - expected)
[1] -64.379  -7.300  -1.647   3.488 130.103
```

MORE ...

```
R version 3.3.3 (2017-03-06) -- "Another Canoe"
  ######  more two-level analyses, melanoma data
> library(lme4)
Loading required package: Matrix
>  library(mlmRev)

>  data(Mmmec)
>  head(Mmmec)
   nation region county deaths expected     uvb
1 Belgium      1      1     79  51.2220 -2.9057
2 Belgium      2      2     80  79.9560 -3.2075
3 Belgium      2      3     51  46.5169 -2.8038
4 Belgium      2      4     43  55.0530 -3.0069
5 Belgium      2      5     89  67.7580 -3.0069
6 Belgium      2      6     19  35.9760 -3.4175
>  str(Mmmec)
'data.frame':   354 obs. of  6 variables:
 $ nation  : Factor w/ 9 levels "Belgium","W.Germany",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 78 levels "1","2","3","4",..: 1 2 2 2 2 2 3 3 3 3 ...
 $ county  : Factor w/ 354 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ deaths  : int  79 80 51 43 89 19 19 15 33 9 ...
 $ expected: num  51.2 80 46.5 55.1 67.8 ...
 $ uvb     : num  -2.91 -3.21 -2.8 -3.01 -3.01 ...
>  # two level analysis: counties within region (ignoring country for now)
>  fm1 = glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base "expect

## fm1 in week 3; try in fm1a allowing death uvb gradient to vary by region
>  fm1a = glmer(deaths ~ uvb + (1+ uvb|region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base "
> summary(fm1a)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 + uvb | region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2257.4   2276.7  -1123.7   2247.4      349

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6536 -0.7579 -0.0382  0.6025  4.2340

Random effects:
 Groups Name        Variance  Std.Dev. Corr
```

```
 region (Intercept) 0.1558795 0.39482
        uvb         0.0002171 0.01473  -1.00
Number of obs: 354, groups:  region, 78

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.149304   0.048105  -3.104  0.00191 **
uvb         -0.037124   0.008658  -4.288 1.81e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb -0.253

> anova(fm1,fm1a)
Data: Mmmec
Models:
fm1: deaths ~ uvb + (1 | region)
fm1a: deaths ~ uvb + (1 + uvb | region)
     Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
fm1   3 2256.4 2268.0 -1125.2   2250.4
fm1a  5 2257.4 2276.7 -1123.7   2247.4 3.026      2     0.2202
```

```
  #### Melanoma, 3-level analyses   county/region/nation
R version 3.3.3 (2017-03-06) -- "Another Canoe"
> library(lme4)
>  library(mlmRev)
>  data(Mmmec)
>  head(Mmmec)
   nation region county deaths expected     uvb
1 Belgium      1      1     79  51.2220 -2.9057
2 Belgium      2      2     80  79.9560 -3.2075
3 Belgium      2      3     51  46.5169 -2.8038
4 Belgium      2      4     43  55.0530 -3.0069
5 Belgium      2      5     89  67.7580 -3.0069
6 Belgium      2      6     19  35.9760 -3.4175
>  str(Mmmec)
'data.frame':   354 obs. of  6 variables:
 $ nation  : Factor w/ 9 levels "Belgium","W.Germany",..: 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 78 levels "1","2","3","4",..: 1 2 2 2 2 2 3 3 3 3 ...
 $ county  : Factor w/ 354 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ deaths  : int  79 80 51 43 89 19 19 15 33 9 ...
 $ expected: num  51.2 80 46.5 55.1 67.8 ...
 $ uvb     : num  -2.91 -3.21 -2.8 -3.01 -3.01 ...

> # 3-level county within region with country
>  fm3 = glmer(deaths ~ uvb + (1|nation/region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base
> summary(fm3)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 | nation/region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2198.7   2214.2  -1095.3   2190.7      350

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9440 -0.7788 -0.0071  0.6277  3.9102

Random effects:
 Groups        Name        Variance Std.Dev.
 region:nation (Intercept) 0.04829  0.2198
 nation        (Intercept) 0.13708  0.3702
Number of obs: 354, groups:  region:nation, 78; nation, 9
```

```
Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.06398    0.13351  -0.479   0.6318
uvb         -0.02822    0.01139  -2.478   0.0132 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb 0.185
> exp(fixef(fm3))
(Intercept)         uvb
  0.9380231   0.9721792

> exp(confint(fm3, oldNames = FALSE))
Computing profile confidence intervals ...
                                    2.5 %    97.5 %
sd_(Intercept)|region:nation 1.1933152 1.3171300
sd_(Intercept)|nation        1.2591286 1.9432869
(Intercept)                  0.7018999 1.2560813
uvb                          0.9510651 0.9949174

## allow death ~ uvb gradient to vary
>  fm4 = glmer(deaths ~ uvb + (1+ uvb|nation/region), Mmmec, poisson, offset = log(expected)) # multiplicative model with
> summary(fm4)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 + uvb | nation/region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2186.8   2217.8  -1085.4   2170.8      346

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1506 -0.7342 -0.0476  0.6647  3.6854

Random effects:
 Groups        Name        Variance  Std.Dev. Corr
 region:nation (Intercept) 0.0222085 0.14903
               uvb         0.0005432 0.02331  -0.34
 nation        (Intercept) 0.1453019 0.38118
               uvb         0.0053806 0.07335  0.21
```

```
Number of obs: 354, groups:  region:nation, 78; nation, 9

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.107310   0.165058   0.650    0.516
uvb         0.005121   0.035695   0.144    0.886

Correlation of Fixed Effects:
    (Intr)
uvb 0.520
# here uvb is no longer protective, that's likley good
# fm4 better than fm3
> anova(fm3, fm4)
Data: Mmmec
Models:
fm3: deaths ~ uvb + (1 | nation/region)
fm4: deaths ~ uvb + (1 + uvb | nation/region)
    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
fm3  4 2198.7 2214.2 -1095.3   2190.7
fm4  8 2186.8 2217.8 -1085.4   2170.8 19.847      4  0.0005353 ***
---

> confint(fm4, oldNames = FALSE)
Computing profile confidence intervals ...
                                        2.5 %     97.5 %
sd_(Intercept)|region:nation       0.09051795 0.22565023
cor_uvb.(Intercept)|region:nation -0.92239656 1.00000000
sd_uvb|region:nation               0.00000000 0.04188866
sd_(Intercept)|nation              0.22343491 0.75278860
cor_uvb.(Intercept)|nation        -0.70602555 0.82608717
sd_uvb|nation                      0.03069115 0.17838145
(Intercept)                       -0.25293629 0.47694641
uvb                               -0.07470335 0.09638890
Warning messages:
1: In FUN(X[[i]], ...) :
  non-monotonic profile for cor_uvb.(Intercept)|nation
2: In confint.thpr(pp, level = level, zeta = zeta) :
  bad spline fit for cor_uvb.(Intercept)|nation: falling back to linear interpolation


## look at Bates way of writing 3-level model instead of nation/region in fm3, fm4
>  fm3_bates = glmer(deaths ~ uvb + (1|nation:region) + (1|nation), Mmmec, poisson, offset = log(expected)) # multiplicati
> summary(fm3_bates)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
```

```
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 | nation:region) + (1 | nation)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2198.7   2214.2  -1095.3   2190.7      350

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9440 -0.7788 -0.0071  0.6277  3.9102

Random effects:
 Groups        Name        Variance Std.Dev.
 nation:region (Intercept) 0.04829  0.2198
 nation        (Intercept) 0.13708  0.3702
Number of obs: 354, groups:  nation:region, 78; nation, 9

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.06398    0.13351  -0.479   0.6318
uvb         -0.02822    0.01139  -2.478   0.0132 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb 0.185

> summary(fm3)  ## see that results are the same with fm3_bates
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 | nation/region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2198.7   2214.2  -1095.3   2190.7      350

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9440 -0.7788 -0.0071  0.6277  3.9102

Random effects:
```

```
 Groups        Name        Variance Std.Dev.
 region:nation (Intercept) 0.04829  0.2198
 nation        (Intercept) 0.13708  0.3702
Number of obs: 354, groups:  region:nation, 78; nation, 9

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.06398    0.13351  -0.479   0.6318
uvb         -0.02822    0.01139  -2.478   0.0132 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb 0.185
>
```

3 Three-level Normal Models

These results are from the 1997 A-level Chemistry exam. The `school` is nested in `lea` (local education authority) and has unique levels for each of the 2410 schools. It is a good practice to make the nesting explicit by specifying the grouping factors as the 'outer' factor, `lea` in this case, and the interaction of the outer and inner factors, `lea:school` or `school:lea` in this case. This will ensure unique levels for each `school` within `lea` combination.

To fit the model `mC2` we increase the number of EM iterations from its default of 20 to 40. Without this change the current version of the `optim` function in R will declare convergence to an incorrect optimum. By increasing the number of EM iterations we are able to get closer to the optimum before calling `optim` and converge to the correct value. The optim function will be patched so this change will not be needed in future versions of R.

Data from the 1997 A-level Chemistry exam are available as `Chem97`.

```
> str(Chem97)
'data.frame':        31022 obs. of  8 variables:
 $ lea      : Factor w/ 131 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ school   : Factor w/ 2410 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ student  : Factor w/ 31022 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ score    : num  4 10 10 10 8 10 6 8 4 10 ...
 $ gender   : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 2 2 ...
 $ age      : num  3 -3 -4 -2 -1 4 1 4 3 0 ...
 $ gcsescore: num  6.62 7.62 7.25 7.50 6.44 ...
 $ gcsecnt  : num  0.339 1.339 0.964 1.214 0.158 ...

> system.time(mC1 <- lmer(score ~ 1 + (1 | lea:school) + (1 | lea),
+     Chem97), gc = TRUE)
[1] 2.74 0.12 2.86 0.00 0.00

> summary(mC1)
Linear mixed-effects model fit by REML
Formula: score ~ 1 + (1 | lea:school) + (1 | lea)
   Data: Chem97
      AIC      BIC   logLik MLdeviance REMLdeviance
 157881.8 157915.2 -78936.9   157869.9     157873.8
Random effects:
 Groups     Name        Variance Std.Dev.
 lea:school (Intercept) 2.74878  1.6579
 lea        (Intercept) 0.15351  0.3918
 Residual               8.51607  2.9182
# of obs: 31022, groups: lea:school, 2410; lea, 131

Fixed effects:
             Estimate Std. Error    DF t value  Pr(>|t|)
(Intercept) 5.3190e+00 5.8110e-02 31021  91.533 < 2.2e-16

> system.time(mC2 <- lmer(score ~ gcsecnt + (1 | school) + (1 |
+     lea), Chem97, control = list(niterEM = 40)), gc = TRUE)
```

Scores on 1997 A-level Chemistry exam

Scores on the 1997 A-level Chemistry examination in Britain. Students are grouped into schools within local education authorities. Some demographic and pre-test information is also provided.

```
> str(Chem97)
'data.frame':        31022 obs. of  8 variables:
 $ lea       : Factor w/ 131 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ school    : Factor w/ 2410 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ student   : Factor w/ 31022 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ score     : num  4 10 10 10 8 10 6 8 4 10 ...
 $ gender    : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 2 2 ...
 $ age       : num  3 -3 -4 -2 -1 4 1 4 3 0 ...
 $ gcsescore: num  6.62 7.62 7.25 7.50 6.44 ...
 $ gcsecnt   : num  0.339 1.339 0.964 1.214 0.158 ...
```

Chem97 grouping factors

```
> summary(Chem97[, c("lea", "school", "gender")])
      lea             school       gender
 118    :  969   698    :  188   M:17262
 116    :  931   1408   :  126   F:13760
 119    :  916   431    :  118
 109    :  802   416    :  111
 113    :  791   1215   :   99
 129    :  762   908    :   94
 (Other):25851   (Other):30286
```

Models fit to these data have $n = 31022$ and $k = 1$ or $k = 2$. When $k = 2$, $m_1 = 2410$ and $m_2 = 131$.

Chemistry (Chem 97)

students / schools / lea

31022 2410 131

gender score

17262 13760

M F

---

variance component models

model 1 school w/in lea

lea/school notation

score ~ 1 + (1|lea/school)

schools nested w/in lea

equivalent

score ~ 1 + (1|lea:school) + (1|lea)

2 level pieces

m1a ignores lea

var school 2.884

mean 5.319

m1b ignores schools

var between .51

within 10.74

---

add prior ~~gse~~

gescnt

cuts down variance

3 Three-level Normal Models

These results are from the 1997 A-level Chemistry exam. The `school` is nested in `lea` (local education authority) and has unique levels for each of the 2410 schools. It is a good practice to make the nesting explicit by specifying the grouping factors as the 'outer' factor, `lea` in this case, and the interaction of the outer and inner factors, `lea:school` or `school:lea` in this case. This will ensure unique levels for each `school` within `lea` combination.

To fit the model `mC2` we increase the number of EM iterations from its default of 20 to 40. Without this change the current version of the `optim` function in R will declare convergence to an incorrect optimum. By increasing the number of EM iterations we are able to get closer to the optimum before calling `optim` and converge to the correct value. The optim function will be patched so this change will not be needed in future versions of R.

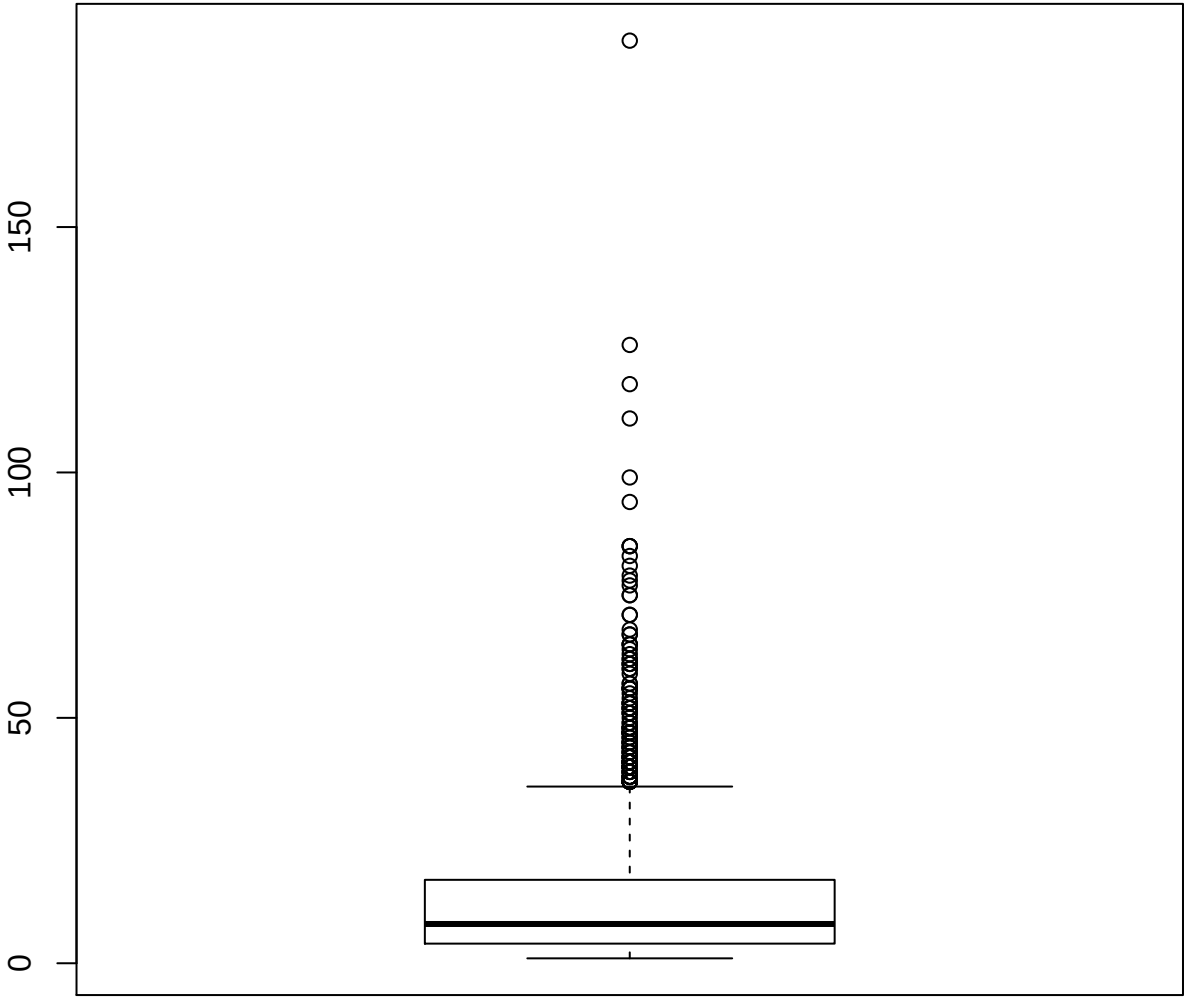
Data from the 1997 A-level Chemistry exam are available as `Chem97`.

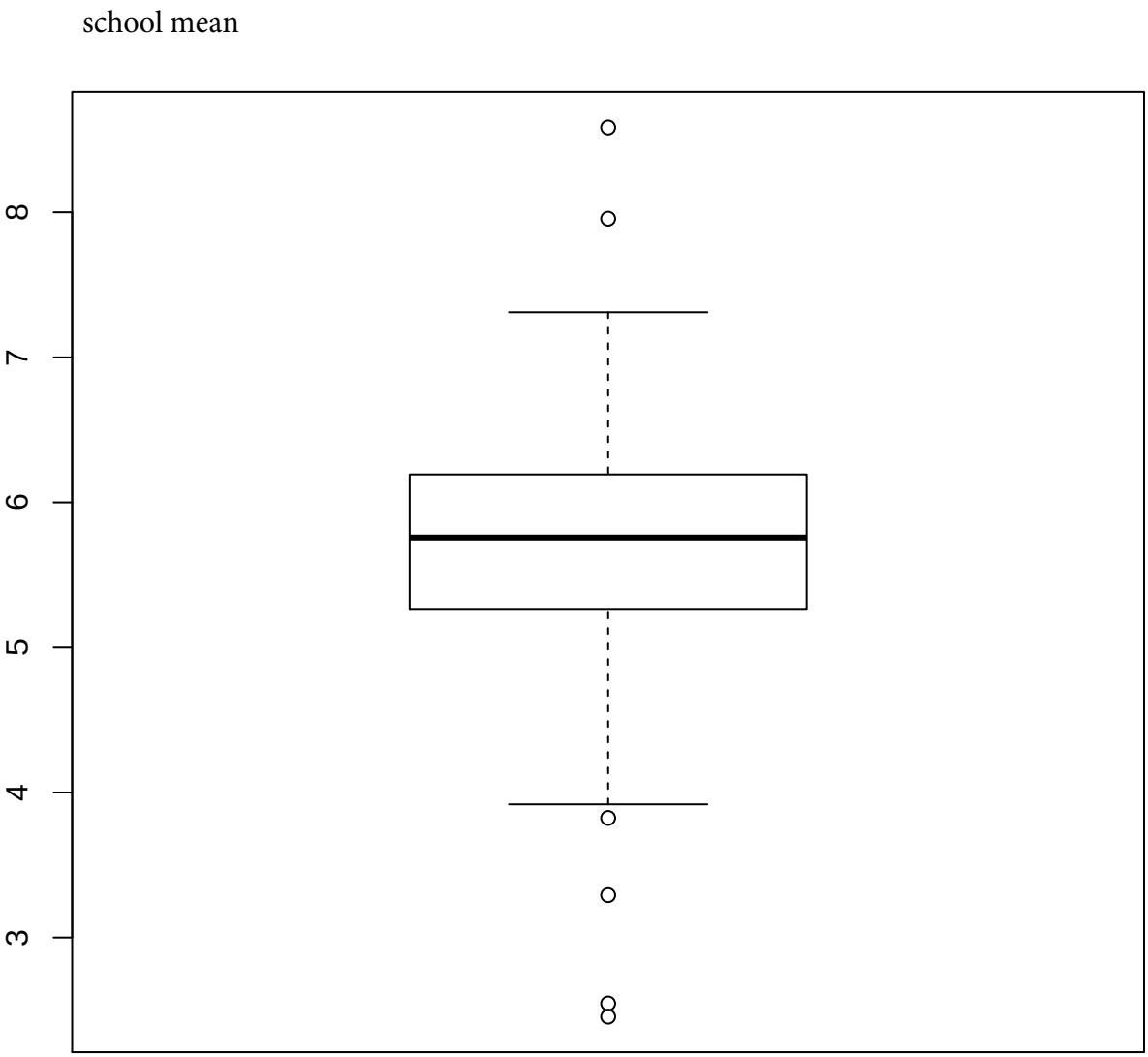
```
> str(Chem97)
'data.frame':        31022 obs. of  8 variables:
 $ lea      : Factor w/ 131 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ school   : Factor w/ 2410 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ student  : Factor w/ 31022 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ score    : num  4 10 10 10 8 10 6 8 4 10 ...
 $ gender   : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 2 2 ...
 $ age      : num  3 -3 -4 -2 -1 4 1 4 3 0 ...
 $ gcsescore: num  6.62 7.62 7.25 7.50 6.44 ...
 $ gcsecnt  : num  0.339 1.339 0.964 1.214 0.158 ...
```

```
> system.time(mC1 <- lmer(score ~ 1 + (1 | lea:school) + (1 | lea),
+     Chem97), gc = TRUE)
[1] 2.74 0.12 2.86 0.00 0.00

> summary(mC1)
Linear mixed-effects model fit by REML
Formula: score ~ 1 + (1 | lea:school) + (1 | lea)
   Data: Chem97
      AIC      BIC   logLik MLdeviance REMLdeviance
 157881.8 157915.2 -78936.9   157869.9     157873.8
Random effects:
 Groups     Name        Variance Std.Dev.
 lea:school (Intercept) 2.74878  1.6579
 lea        (Intercept) 0.15351  0.3918
 Residual               8.51607  2.9182
# of obs: 31022, groups: lea:school, 2410; lea, 131

Fixed effects:
             Estimate Std. Error    DF t value  Pr(>|t|)
(Intercept) 5.3190e+00 5.8110e-02 31021  91.533 < 2.2e-16

> system.time(mC2 <- lmer(score ~ gcsecnt + (1 | school) + (1 |
+     lea), Chem97, control = list(niterEM = 40)), gc = TRUE)
```

```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"

> library(lme4)
Loading required package: Matrix
Loading required package: Rcpp
> library(mlmRev)
Warning message:
package 'mlmRev' was built under R version 3.1.3
> data(Chem97)
> str(Chem97)
'data.frame':   31022 obs. of  8 variables:
 $ lea      : Factor w/ 131 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ school   : Factor w/ 2410 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ student  : Factor w/ 31022 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ score    : num  4 10 10 10 8 10 6 8 4 10 ...
 $ gender   : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 2 2 ...
 $ age      : num  3 -3 -4 -2 -1 4 1 4 3 0 ...
 $ gcsescore: num  6.62 7.62 7.25 7.5 6.44 ...
 $ gcsecnt  : num  0.339 1.339 0.964 1.214 0.158 ...
> ## Description
> ## Scores on the 1997 A-level Chemistry examination in Britain. Students are grouped into schools within local edu
> ## In addition some demographic and pre-test information is provided.
> ## Usage
> ## data(Chem97)
> ## Format
> ## A data frame with 31022 observations on the following 8 variables.
> ## lea    Local Education Authority - a factor
> ## school School identifier - a factor
> ## student    Student identifier - a factor
> ## score    Point score on A-level Chemistry in 1997
> ## gender    Student's gender
> ## age    Age in month, centred at 222 months or 18.5 years
> ## gcsescore    Average GCSE score of individual.
> ## gcsecnt    Average GCSE score of individual, centered at mean.
> ## Details This data set is relatively large with 31,022 individuals in 2,280 schools.
>

> attach(Chem97)
> dim(Chem97)
[1] 31022     8
> head(Chem97)
  lea school student score gender age gcsescore   gcsecnt
1   1      1       1     4      F   3     6.625 0.3393157
2   1      1       2    10      F  -3     7.625 1.3393157
3   1      1       3    10      F  -4     7.250 0.9643157
4   1      1       4    10      F  -2     7.500 1.2143157
5   1      1       5     8      F  -1     6.444 0.1583157
6   1      1       6    10      F   4     7.750 1.4643157

> length(unique(school))
[1] 2410
> length(unique(lea))
[1] 131

> fivenum(tapply(gcsescore, school, length))
  10 1391 2348 1649  698
   1    4    8   17  188
> boxplot(tapply(gcsescore, school, length)) # schoolsize boxplot

>  fivenum(tapply(score, school, mean))
       25       337        30       309      2401
 0.000000  3.818182  5.422619  6.727273 10.000000
>  fivenum(tapply(score, lea, mean))
       4       18      111       82        1
2.454545 5.260468 5.758242 6.192977 8.585366

> boxplot(tapply(score, school, mean)) # school mean boxplot
> boxplot(tapply(score, lea, mean))    # lea mean boxplot

> fivenum(tapply(score, school, var))  # distrib of within-school vars
       25       733      1943      1196      1422
 0.000000  4.952381  8.000000 11.000000 50.000000
> fivenum(tapply(score, lea, var))  # distrib of within-LEA vars
       1        46        47        20        41
```

schoolsize boxplot




school mean
8
7
6
5
4
3

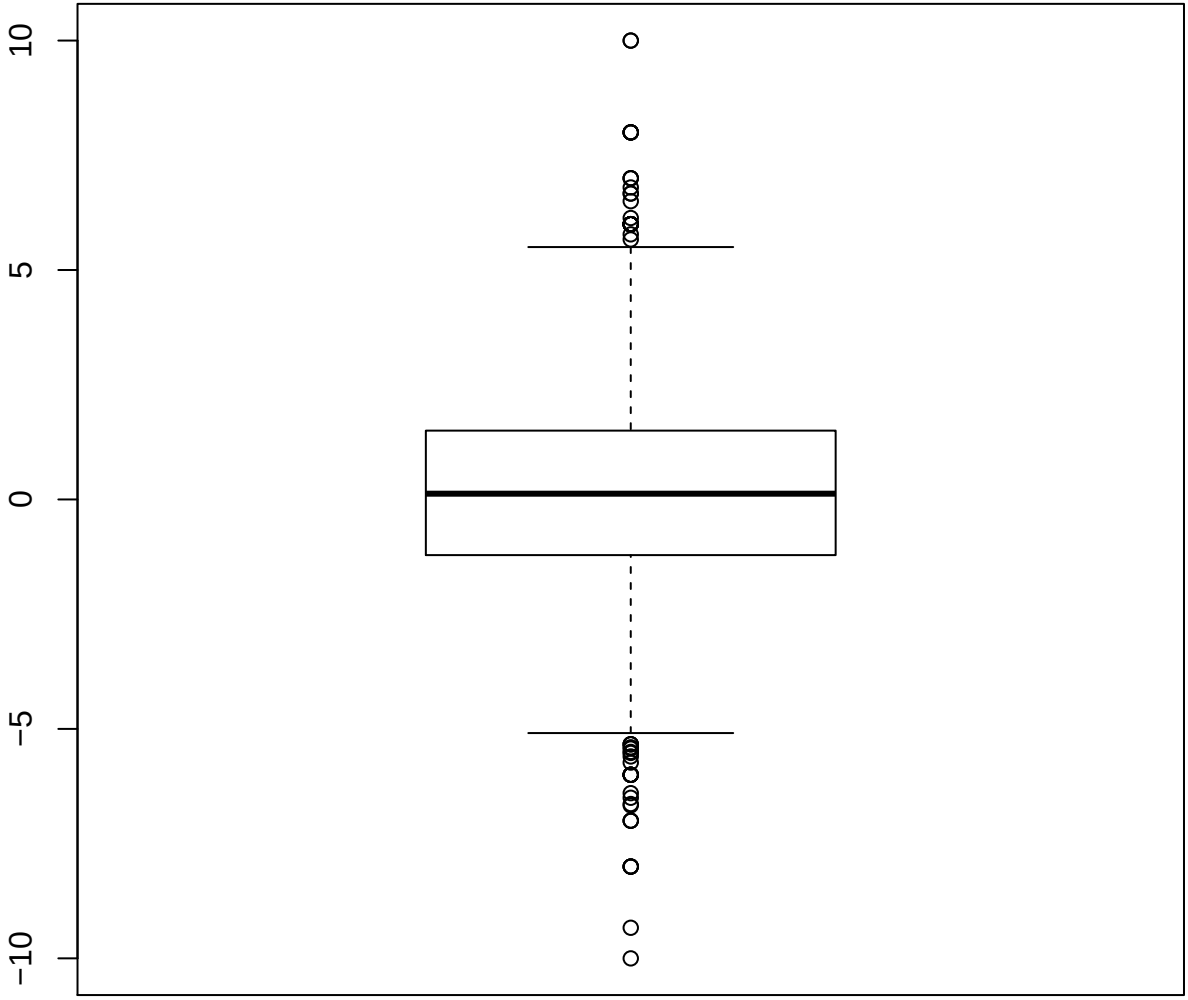
```
 5.048780  9.952842 10.759780 11.593020 14.849673

> var((tapply(score, lea, mean))) # var of LEA means
[1] 0.7581529
> var((tapply(score, school, mean))) # var of School means
[1] 5.02233


# Bates first model, variance components
> m1 = lmer(score ~ 1 + (1|lea/school), data = Chem97)
> summary(m1)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ 1 + (1 | lea/school)
   Data: Chem97

REML criterion at convergence: 157873.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9972 -0.7366  0.1083  0.7586  2.6580

Random effects:
 Groups     Name        Variance Std.Dev.
 school:lea (Intercept) 2.7487   1.6579
 lea        (Intercept) 0.1535   0.3918
 Residual               8.5161   2.9182
Number of obs: 31022, groups:  school:lea, 2410; lea, 131

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.31898    0.05811   91.54

#In mlmRev Bates does m1 this equivalent way
> mC1 <- lmer(score ~ 1+(1|lea:school) + (1|lea), Chem97)
> summary(mC1)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ 1 + (1 | lea:school) + (1 | lea)
   Data: Chem97

REML criterion at convergence: 157873.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9972 -0.7366  0.1083  0.7586  2.6580

Random effects:
 Groups     Name        Variance Std.Dev.
 lea:school (Intercept) 2.7487   1.6579
 lea        (Intercept) 0.1535   0.3918
 Residual               8.5161   2.9182
Number of obs: 31022, groups:  lea:school, 2410; lea, 131

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.31898    0.05811   91.54

## look at 2-level pieces of this
> m1a = lmer(score ~ 1 + (1|school), data = Chem97)
> m1b = lmer(score ~ 1 + (1|lea), data = Chem97)
> summary(m1a)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ 1 + (1 | school)
   Data: Chem97

REML criterion at convergence: 157894.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.9975 -0.7406  0.1077  0.7608  2.6160

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept) 2.884    1.698
 Residual             8.517    2.918
```

```
Number of obs: 31022, groups:  school, 2410

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.32967    0.04148   128.5


> summary(m1b)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ 1 + (1 | lea)
   Data: Chem97

REML criterion at convergence: 161967

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.29748 -0.69397  0.08704  0.77878  1.90128

Random effects:
 Groups   Name        Variance Std.Dev.
 lea      (Intercept)  0.5086  0.7132
 Residual             10.7371  3.2768
Number of obs: 31022, groups:  lea, 131

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.68424    0.06729   84.48

# Bates second model, two equivalent models
> mC2 <- lmer(score ~ gcsecnt + (1|school) + (1|lea), Chem97)
> mC2a <- lmer(score ~ gcsecnt + (1|lea/school) , Chem97)
> summary(mC2)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ gcsecnt + (1 | school) + (1 | lea)
   Data: Chem97

REML criterion at convergence: 141697

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1866 -0.6353  0.0759  0.6677  6.6927

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept) 1.16620  1.0799
 lea      (Intercept) 0.01477  0.1215
 Residual             5.15420  2.2703
Number of obs: 31022, groups:  school, 2410; lea, 131

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.63545    0.03123   180.4
gcsecnt      2.47256    0.01690   146.3

Correlation of Fixed Effects:
        (Intr)
gcsecnt 0.058
> summary(mC2a)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ gcsecnt + (1 | lea/school)
   Data: Chem97

REML criterion at convergence: 141697

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1866 -0.6353  0.0759  0.6677  6.6927

Random effects:
 Groups     Name        Variance Std.Dev.
 school:lea (Intercept) 1.16620  1.0799
 lea        (Intercept) 0.01477  0.1215
 Residual               5.15420  2.2703
Number of obs: 31022, groups:  school:lea, 2410; lea, 131
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.63545    0.03123   180.4
gcsecnt      2.47256    0.01690   146.3

Correlation of Fixed Effects:
        (Intr)
gcsecnt 0.058


## try a formulation for gender diffs, females rule even in Chem
> tapply(score, gender,mean)
       M        F
5.759124 5.879651
> sexList <- lmList(score ~ gender|school , Chem97)
> sexC = coef(sexList)
> fivenum(sexC[,2] )
[1] -10.0000000  -1.2125000   0.1255411   1.5000000  10.0000000
> length(sexList)
[1] 2410
> boxplot(sexC[,2]) #boxplot of F - M

> mCsex <- lmer(score ~ gender + (gender|lea/school) , Chem97)
> summary(mCsex)
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ gender + (gender | lea/school)
   Data: Chem97

REML criterion at convergence: 157817.3

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0083 -0.7358  0.1076  0.7520  2.6732

Random effects:
 Groups      Name        Variance Std.Dev. Corr
 school:lea (Intercept) 2.62025  1.6187
            genderF     0.47798  0.6914   -0.03
 lea        (Intercept) 0.15747  0.3968
            genderF     0.02681  0.1637   -0.17
 Residual               8.42735  2.9030
Number of obs: 31022, groups:  school:lea, 2410; lea, 131

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.23501    0.06117   85.58
genderF      0.17314    0.04785    3.62

Correlation of Fixed Effects:
        (Intr)
genderF -0.324

> confint(mCsex)
Computing profile confidence intervals ...
                  2.5 %    97.5 %
.sig01       1.53791969 1.7033900
.sig02      -0.18766852 0.1619631
.sig03       0.51427801 0.8515856
.sig04       0.00000000       Inf
.sig05      -1.00000000 1.0000000
.sig06       0.00000000 0.3365437
.sigma       2.87888736 2.9274811
(Intercept)  5.11272330 5.3567584
genderF      0.07363619 0.2684750
```


)

F - M; school means



```
> confint(mCsex, method = "boot", nsim = 1000, boot.type = "perc")
Computing bootstrap confidence intervals ...
                                              2.5 %     97.5 %
sd_(Intercept)|school:lea                1.542173940 1.6995366
cor_genderF.(Intercept)|school:lea -0.173526409 0.1292961
sd_genderF|school:lea                    0.532795821 0.8199634
sd_(Intercept)|lea                       0.249956868 0.5319830
cor_genderF.(Intercept)|lea             -1.000000000 1.0000000
sd_genderF|lea                           0.009096021 0.3062512
sigma                                    2.879502762 2.9283745
(Intercept)                              5.117125886 5.3495786
genderF                                  0.075526250 0.2635595
```

The additional piece of Equation (2.19) is $\gamma_{h1}z_j$, which represents the slope for (γ_{h1}), and value of the average vocabulary score for the school (z_j). In other words, the mean school performance is related directly to the coefficient linking the individual vocabulary score to the individual reading score. For our specific example, we can combine Equations (2.18) and (2.19) to yield a single equation for the two-level MLM.

$$y_{ij} = \gamma_{00} + \gamma_{10}x_{ij} + \gamma_{01}z_j + \gamma_{1001}x_{ij}z_j + U_{0j} + U_{1j}x_{ij} + \varepsilon_{ij} \quad (2.20)$$

Each of these model terms has been defined previously in this chapter: γ_{00} is the intercept or grand mean for the model, γ_{10} is the fixed effect of variable x (vocabulary) on the outcome, U_{0j} represents the random variation for the intercept across groups, and U_{1j} represents the random variation for the slope across groups.

The additional pieces of Equation (2.13) are γ_{01} and γ_{11}. The γ_{01} represents the fixed effect of Level 2 variable z (average vocabulary) on the outcome and γ_{11} represents the slope for and value of the average vocabulary score for the school. The new term in Equation (2.20) is the cross-level interaction $\gamma_{1001}x_{ij}z_j$. As the name implies, the cross-level interaction is simply an interaction of Level 1 and Level 2 predictors. In this context, it represents the interaction between an individual's vocabulary score and the mean vocabulary score for his or her school. The coefficient for this interaction term, γ_{1001}, assesses the extent to which the relationship between a student's vocabulary score is moderated by the mean for the school attended. A large significant value for this coefficient would indicate that the relationship between an individual's vocabulary test score and overall reading achievement is dependent on the level of vocabulary achievement at his or her school.

2.8 Overview of Three-Level MLMs

It is entirely possible to utilize three or more levels of data structures with MLMs. We should note, however, that four-level and larger models are rare in practice. For our reading achievement data in which the second level was school, a possible third level might be the district in which the school is located. In that case, we would have multiple equations to consider when expressing the relationship between vocabulary and reading achievement scores, starting at the individual level:

$$y_{ijk} = \beta_{0jk} + \beta_{1jk}x_{ijk} + \varepsilon_{ijk} \quad (2.21)$$

The subscript k represents the Level 3 cluster to which the individual belongs.

Before formulating the rest of the model, we must evaluate whether the slopes and intercepts are random at both Levels 2 and 3 or only at Level 1, for example. This decision should always be based on the theory surrounding the research questions, what is expected in the population, and what is revealed in the empirical data. We will proceed with the remainder of this discussion under the assumption that the Level 1 intercepts and slopes are random for both Levels 2 and 3 in order to provide a complete description of the most complex model possible when three levels of data structure are present. When the Level 1 coefficients are not random at both levels, the terms in the following models for which this randomness is not present would simply be removed. We will address this issue more specifically in Chapter 4 when we discuss the fitting of three-level models using R. The Level 2 and Level 3 contributions to the MLM described in Equation (2.13) appear below.

$$\text{Level 2: } \beta_{0jk} = \gamma_{00k} + U_{0jk}$$

$$\beta_{1jk} = \gamma_{10k} + U_{1jk}$$

$$\text{Level 3: } \gamma_{00k} = \delta_{000} + V_{00k}$$

$$\gamma_{10k} = \delta_{100} + V_{10k} \tag{2.22}$$

We can then use simple substitution to obtain the expression for the Level 1 intercept and slope in terms of both Level 2 and Level 3 parameters.

$$\beta_{0jk} = \delta_{000} + V_{00k} + U_{0jk}$$

$$\beta_{1jk} = \delta_{100} + V_{10k} + U_{1jk} \tag{2.23}$$

In turn, these terms may be substituted into Equation (2.15) to provide the full three-level MLM.

$$y_{ijk} = \delta_{000} + V_{00k} + U_{0jk} + (\delta_{100} + V_{10k} + U_{1jk})x_{ijk} + \varepsilon_{ijk} \tag{2.24}$$

There is an implicit assumption in this expression of Equation (2.24) that there are no cross-level interactions, although they certainly may be modeled across all three levels or for any pair of levels. Equation (2.24) expresses individuals' scores on the reading achievement test as a function of random and fixed components from the school they attend, the district in which the school is located, and their own vocabulary test scores and random variations associated only with them. Although not included in Equation (2.24), it is also possible to include variables at both Levels 2 and 3, similar to what we described for the two-level model structure.

4

Models of Three and More Levels

Chapters 2 and 3 introduced the multilevel modeling framework and demonstrated the use of the `nlme` and `lme4` R packages in fitting two-level models. In Chapter 4, we will expand upon this basic two-level framework by fitting models with additional levels of data structure. As described in Chapter 2, it is conceivable for a Level 1 unit such as student to be nested in higher level units such as classroom. Thus, in keeping with our examples, we may assume that at least a portion of a student's performance on a reading test is due to the classroom in which he or she learns. Each classroom may have a unique learning context that may contribute to student performance, for example, the quality of the teacher, the presence of disruptive students, and time of day when students are in the class, among others. Furthermore, as we saw in the earlier chapters, the impacts of fixed effects on a dependent variable can vary among Level 2 units, resulting in a random slope model.

We will see that it is possible to estimate models with three or more levels of a nested structure using R and learn that the R commands for defining and fitting these models are very similar to those used in the two-level case. Within the `nlme` and `lme4` packages, the same function calls that we used for two-level models can be used to define models with three or more levels:

```
lme(fixed, data, random, correlation, weights, subset, method,
    na.action, control, contrasts = NULL, keep.data = TRUE)

lmer(formula, data, family = NULL, REML = TRUE,
     control = list(), start = NULL, verbose = FALSE,
     doFit = TRUE, subset, weights, na.action, offset,
     contrasts = NULL, model = TRUE, x = TRUE,...)
```

In this chapter, we will continue working with the data described in Chapter 3. The examples in that chapter included two levels of data structures (students within schools and associated predictors of reading achievement at each level). We will now add a third level of structure, the classroom, which is nested within schools. In this context, *nested* simply means that students within a classroom all attend the same school. Thus, students are nested within classrooms that in turn are nested within schools.

```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"

> setwd("D:\\drr15\\ed401D\\week4\\")
> library(lme4)
Loading required package: Matrix
Loading required package: Rcpp
> library(lattice)
> ach = read.csv("D:\\drr15\\ed401D\\week4\\Achieve.csv") #downloaded from book website
> str(ach)
'data.frame':   10320 obs. of  89 variables:
 $ row     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ region  : int  2 2 2 2 2 2 2 2 2 2 ...
 $ corp    : int  940 940 940 940 940 940 940 940 940 940 ...
 $ school  : int  767 767 767 767 767 767 767 767 767 767 ...
 $ class   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ gender  : int  2 2 2 2 1 1 2 1 2 ...
 $ age     : int  104 106 112 109 107 108 101 103 104 110 ...
 $ race    : int  5 5 5 5 5 5 5 5 5 5 ...
 $ gread   : num  3.5 1.2 2.1 1.6 3.7 2.4 3.1 2.9 4.9 5 ...
 $ gevocab : num  3.1 2.8 1.7 2.1 2.4 2.4 3.3 3.2 9.3 5.4 ...
 $ greadcm: num  3.2 2 1.9 1.9 2.8 2.3 3.1 3 7.2 5.2 ...
 $ gelang  : num  2.3 1.8 2.5 1.9 2.1 2 3.7 2.8 4.4 4.4 ...
 $ gelangmc: num  4.3 1.7 2.2 1.6 1.7 2.9 1.7 2.7 6.6 4.5 ...
 $ gelangcm: num  3 1.6 2.4 1.6 1.9 2.3 2.7 2.7 5.3 4.4 ...
 $ gemath  : num  3.1 3.2 3.6 3.3 3.2 5.5 3 3.4 4.4 6.2 ...
 $ gemathcp: num  2.7 4.3 3.2 3.2 2.9 4.1 3.1 4 5.5 3.8 ...
 $ gemathcm: num  2.9 3.8 3.4 3.2 3 4.7 3.1 3.8 5.1 4.7 ...
 $ getotal : num  3 2.7 2.7 2.3 2.7 3.4 3 3.3 6.3 4.9 ...
 $ npanverb: int  16 39 37 45 50 48 16 34 94 86 ...
 $ npamem  : int  56 79 41 56 71 89 29 31 38 66 ...
 $ npaverb : int  46 37 34 23 24 50 41 36 33 96 ...
 $ npatotal: int  26 47 35 39 45 60 21 32 75 92 ...
 $ csi     : int  90 99 94 95 98 104 87 93 111 123 ...
 $ multi   : int  2 2 2 2 2 2 2 2 2 2 ...
 $ clenroll: int  16 16 16 16 16 16 16 16 16 16 ...
 $ classize: int  1 1 1 1 1 1 1 1 1 1 ...
 $ ptratio : num  16 16 16 16 16 16 16 16 16 16 ...
 $ ptia    : int  2 2 2 2 2 2 2 2 2 2 ...
 $ locale  : int  7 7 7 7 7 7 7 7 7 7 ...
 $ chapter1: int  1 1 1 1 1 1 1 1 1 1 ...
 $ ses     : num  80.4 80.4 80.4 80.4 80.4 80.4 80.4 80.4 80.4 80.4 ...
 $ context : int  929 929 929 929 929 929 929 929 929 929 ...
 $ calender: int  1 1 1 1 1 1 1 1 1 1 ...
 $ senroll : int  463 463 463 463 463 463 463 463 463 463 ...
 $ sattend : num  95.6 95.6 95.6 95.6 95.6 95.6 95.6 95.6 95.6 95.6 ...
 $ white1  : num  99.8 99.8 99.8 99.8 99.8 99.8 99.8 99.8 99.8 99.8 ...
 $ black1  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ hispanc1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ asian1  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ aindian1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ multi1  : num  0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 ...
 $ total1  : num  0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 ...
 $ noteach : num  28.5 28.5 28.5 28.5 28.5 28.5 28.5 28.5 28.5 28.5 ...
 $ avgage1 : num  46.2 46.2 46.2 46.2 46.2 46.2 46.2 46.2 46.2 46.2 ...
```

```
 $ avgexp1 : num  20.6 20.6 20.6 20.6 20.6 20.6 20.6 20.6 20.6 20.6 ...
 $ avgsal1 : int  43517 43517 43517 43517 43517 43517 43517 43517 43517 43517 ...
 $ spert   : num  16.3 16.3 16.3 16.3 16.3 16.3 16.3 16.3 16.3 16.3 ...
 $ thrdclss: int  4 4 4 4 4 4 4 4 4 4 ...
 $ thrdstud: int  60 60 60 60 60 60 60 60 60 60 ...
 $ passla1 : int  63 63 63 63 63 63 63 63 63 63 ...
 $ passmth1: int  74 74 74 74 74 74 74 74 74 74 ...
 $ passbth1: int  52 52 52 52 52 52 52 52 52 52 ...
 $ tmnnce1 : num  60.5 60.5 60.5 60.5 60.5 60.5 60.5 60.5 60.5 60.5 ...
 $ rmdnce1 : num  57.6 57.6 57.6 57.6 57.6 57.6 57.6 57.6 57.6 57.6 ...
 $ lamdnce1: num  55 55 55 55 55 55 55 55 55 55 ...
 $ mmdnce1 : num  61 61 61 61 61 61 61 61 61 61 ...
 $ tmdnce1 : num  59.7 59.7 59.7 59.7 59.7 59.7 59.7 59.7 59.7 59.7 ...
 $ avgcsi1 : int  99 99 99 99 99 99 99 99 99 99 ...
 $ geog    : int  2 2 2 2 2 2 2 2 2 2 ...
 $ totepp  : int  5956 5956 5956 5956 5956 5956 5956 5956 5956 5956 ...
 $ cenroll : int  3115 3115 3115 3115 3115 3115 3115 3115 3115 3115 ...
 $ cattend : num  95.9 95.9 95.9 95.9 95.9 95.9 95.9 95.9 95.9 95.9 ...
 $ freelnch: num  16 16 16 16 16 16 16 16 16 16 ...
 $ lep     : num  0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 ...
 $ speced  : num  14.4 14.4 14.4 14.4 14.4 14.4 14.4 14.4 14.4 14.4 ...
 $ minority: num  1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 ...
 $ white2  : int  3077 3077 3077 3077 3077 3077 3077 3077 3077 3077 ...
 $ black2  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hispanc2: int  12 12 12 12 12 12 12 12 12 12 ...
 $ asian2  : int  3 3 3 3 3 3 3 3 3 3 ...
 $ aindian2: int  1 1 1 1 1 1 1 1 1 1 ...
 $ multi2  : int  21 21 21 21 21 21 21 21 21 21 ...
 $ total2  : int  38 38 38 38 38 38 38 38 38 38 ...
 $ thrdadm : int  208 208 208 208 208 208 208 208 208 208 ...
 $ thrdtech: num  11 11 11 11 11 11 11 11 11 11 ...
 $ avgage2 : num  45.2 45.2 45.2 45.2 45.2 45.2 45.2 45.2 45.2 45.2 ...
 $ avgexp2 : num  18.4 18.4 18.4 18.4 18.4 18.4 18.4 18.4 18.4 18.4 ...
 $ avgsal2 : int  41096 41096 41096 41096 41096 41096 41096 41096 41096 41096 ...
 $ thrdaide: num  3 3 3 3 3 3 3 3 3 3 ...
 $ passla2 : int  67 67 67 67 67 67 67 67 67 67 ...
 $ passmth2: int  70 70 70 70 70 70 70 70 70 70 ...
 $ passbth2: int  52 52 52 52 52 52 52 52 52 52 ...
 $ tmnnce2 : num  59.4 59.4 59.4 59.4 59.4 59.4 59.4 59.4 59.4 59.4 ...
 $ rmdnce2 : num  57.8 57.8 57.8 57.8 57.8 57.8 57.8 57.8 57.8 57.8 ...
 $ lamdnce2: num  59.8 59.8 59.8 59.8 59.8 59.8 59.8 59.8 59.8 59.8 ...
 $ mmdnce2 : num  60.6 60.6 60.6 60.6 60.6 60.6 60.6 60.6 60.6 60.6 ...
 $ tmdnce2 : num  60.8 60.8 60.8 60.8 60.8 60.8 60.8 60.8 60.8 60.8 ...
 $ rmediate: int  18 18 18 18 18 18 18 18 18 18 ...

> summary(ach[, c("geread", "gevocab", "cenroll", "clenroll", "school", "class", "corp")])
     geread          gevocab          cenroll         clenroll        school         class           corp
 Min.   : 0.000   Min.   : 0.000   Min.   :  110   Min.   :12.00   Min.   : 767   Min.   :1.000   Min.   : 940
 1st Qu.: 2.800   1st Qu.: 2.900   1st Qu.: 1936   1st Qu.:19.00   1st Qu.:2478   1st Qu.:1.000   1st Qu.:3025
 Median : 3.800   Median : 3.800   Median : 3993   Median :21.00   Median :4677   Median :2.000   Median :4670
 Mean   : 4.341   Mean   : 4.494   Mean   : 7867   Mean   :21.37   Mean   :4632   Mean   :2.545   Mean   :4687
 3rd Qu.: 4.900   3rd Qu.: 5.200   3rd Qu.:11428   3rd Qu.:23.00   3rd Qu.:6393   3rd Qu.:3.000   3rd Qu.:5930
 Max.   :12.000   Max.   :11.200   Max.   :41359   Max.   :31.00   Max.   :9173   Max.   :8.000   Max.   :8435

> length(unique(corp))
[1] 59
```

```
> length(unique(school))
[1] 160
> length(unique(class)) # unique only within school, Bates fix Exam data
[1] 8
> fivenum(tapply(geread, class, length))
    8      7      5      3      1
 28.0  149.0 1021.5 2534.0 2883.0
> fivenum(tapply(geread, school, length))
2201  1949  1773  2751  2478
11.0  45.0  63.0  81.5 162.0
```

```
> summary(ach)
      row             id            region          corp          school         class          gender           age             race
 Min.   :    1   Min.   :    1   Min.   :1.000   Min.   : 940   Min.   : 767   Min.   :1.000   Min.   :1.000   Min.   : 82.0   Min.   :1.000
 1st Qu.: 2810   1st Qu.: 2791   1st Qu.:3.000   1st Qu.:3025   1st Qu.:2478   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:104.0   1st Qu.:5.000
 Median : 5530   Median : 5422   Median :4.000   Median :4670   Median :4677   Median :2.000   Median :2.000   Median :107.0   Median :5.000
 Mean   : 5490   Mean   : 5447   Mean   :4.814   Mean   :4687   Mean   :4632   Mean   :2.545   Mean   :1.502   Mean   :107.5   Mean   :4.688
 3rd Qu.: 8192   3rd Qu.: 8093   3rd Qu.:7.000   3rd Qu.:5930   3rd Qu.:6393   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:111.0   3rd Qu.:5.000
 Max.   :10927   Max.   :10953   Max.   :9.000   Max.   :8435   Max.   :9173   Max.   :8.000   Max.   :2.000   Max.   :135.0   Max.   :6.000
     geread          gevocab         gereadcm         gelang         gelangmc        gelangcm         gemath          gemathcp
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.200   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   :0.000
 1st Qu.: 2.800   1st Qu.: 2.900   1st Qu.: 2.900   1st Qu.: 2.80   1st Qu.: 2.900   1st Qu.: 2.900   1st Qu.: 3.200   1st Qu.:2.900
 Median : 3.800   Median : 3.800   Median : 3.800   Median : 3.70   Median : 4.200   Median : 4.000   Median : 4.000   Median :3.200
 Mean   : 4.341   Mean   : 4.494   Mean   : 4.378   Mean   : 4.93   Mean   : 4.763   Mean   : 4.744   Mean   : 4.776   Mean   :3.501
 3rd Qu.: 4.900   3rd Qu.: 5.200   3rd Qu.: 5.000   3rd Qu.: 5.90   3rd Qu.: 6.000   3rd Qu.: 5.900   3rd Qu.: 5.800   3rd Qu.:4.000
 Max.   :12.000   Max.   :11.200   Max.   :12.500   Max.   :12.00   Max.   :11.400   Max.   :12.500   Max.   :11.000   Max.   :6.500
    gemathcm        getotal         npanverb         npamem         npaverb         npatotal          csi             multi          clenroll
 Min.   :0.200   Min.   :0.200   Min.   : 1.00   Min.   : 1.00   Min.   : 1.00   Min.   : 1.00   Min.   : 58.0   Min.   :1.000   Min.   :12.00
 1st Qu.:3.100   1st Qu.:3.100   1st Qu.:34.00   1st Qu.:32.00   1st Qu.:32.00   1st Qu.:36.00   1st Qu.: 94.0   1st Qu.:2.000   1st Qu.:19.00
 Median :3.600   Median :3.900   Median :58.00   Median :58.00   Median :55.00   Median :58.00   Median :103.0   Median :2.000   Median :21.00
 Mean   :3.992   Mean   :4.248   Mean   :56.29   Mean   :55.87   Mean   :54.16   Mean   :56.99   Mean   :103.9   Mean   :1.976   Mean   :21.37
 3rd Qu.:4.500   3rd Qu.:5.000   3rd Qu.:80.00   3rd Qu.:84.00   3rd Qu.:77.00   3rd Qu.:80.00   3rd Qu.:114.0   3rd Qu.:2.000   3rd Qu.:23.00
 Max.   :7.800   Max.   :8.900   Max.   :99.00   Max.   :99.00   Max.   :99.00   Max.   :99.00   Max.   :141.0   Max.   :2.000   Max.   :31.00
    classize        ptratio          ptia          locale         chapter1          ses              context          calender     senroll
 Min.   :1.000   Min.   :11.28   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :  0.00   Min.   :  12.0   Min.   :1   Min.   :115.0
 1st Qu.:2.000   1st Qu.:18.05   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:1.000   1st Qu.: 66.30   1st Qu.: 314.0   1st Qu.:1   1st Qu.:438.0
 Median :2.000   Median :19.74   Median :2.000   Median :6.000   Median :1.000   Median : 81.70   Median : 649.0   Median :1   Median :519.0
 Mean   :2.421   Mean   :19.72   Mean   :1.653   Mean   :4.624   Mean   :1.311   Mean   : 72.85   Mean   : 639.5   Mean   :1   Mean   :533.4
 3rd Qu.:3.000   3rd Qu.:21.00   3rd Qu.:2.000   3rd Qu.:7.000   3rd Qu.:2.000   3rd Qu.: 87.80   3rd Qu.: 901.0   3rd Qu.:1   3rd Qu.:644.0
 Max.   :4.000   Max.   :31.00   Max.   :3.000   Max.   :7.000   Max.   :2.000   Max.   :100.00   Max.   :1312.0   Max.   :1   Max.   :916.0
    sattend          white1          black1          hispanc1         asian1          aindian1         multi1          total1
 Min.   :90.80   Min.   :  0.00   Min.   : 0.000   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000   Min.   : 0.0   Min.   :  0.00
 1st Qu.:95.80   1st Qu.: 82.10   1st Qu.: 0.000   1st Qu.: 0.100   1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.4   1st Qu.:  1.70
 Median :96.50   Median : 96.00   Median : 0.500   Median : 0.800   Median : 0.2000   Median : 0.000   Median : 1.4   Median :  4.00
 Mean   :96.24   Mean   : 83.94   Mean   : 9.167   Mean   : 4.037   Mean   : 0.7883   Mean   : 0.243   Mean   : 1.8   Mean   : 15.95
 3rd Qu.:96.90   3rd Qu.: 98.30   3rd Qu.: 9.000   3rd Qu.: 1.900   3rd Qu.: 0.8000   3rd Qu.: 0.200   3rd Qu.: 2.8   3rd Qu.: 17.90
 Max.   :99.00   Max.   :100.00   Max.   :95.900   Max.   :75.200   Max.   :18.8000   Max.   :16.000   Max.   :15.8   Max.   :100.00
    noteach         avgage1         avgexp1         avgsal1          spert          thrdclss        thrdstud         passla1         passmth1
 Min.   : 8.50   Min.   :35.30   Min.   : 5.90   Min.   :34907   Min.   :11.60   Min.   :1.000   Min.   : 11.00   Min.   :25.00   Min.   :34.00
 1st Qu.:24.40   1st Qu.:41.20   1st Qu.:13.30   1st Qu.:39839   1st Qu.:17.10   1st Qu.:3.000   1st Qu.: 60.00   1st Qu.:63.00   1st Qu.:66.00
 Median :29.70   Median :43.45   Median :15.60   Median :41511   Median :18.40   Median :4.000   Median : 75.00   Median :70.00   Median :74.00
 Mean   :29.34   Mean   :43.03   Mean   :15.38   Mean   :41762   Mean   :18.24   Mean   :4.137   Mean   : 78.68   Mean   :69.34   Mean   :73.32
 3rd Qu.:34.30   3rd Qu.:44.90   3rd Qu.:17.40   3rd Qu.:43812   3rd Qu.:19.60   3rd Qu.:5.000   3rd Qu.: 95.00   3rd Qu.:78.00   3rd Qu.:83.00
 Max.   :47.80   Max.   :49.70   Max.   :22.80   Max.   :51605   Max.   :23.10   Max.   :8.000   Max.   :166.00   Max.   :97.00   Max.   :99.00
    passbth1        tmnnce1         rmdnce1         lamdnce1        mmdnce1         tmdnce1         avgcsi1          geog            totepp
```

```
 Min.   :20.00   Min.   :44.20   Min.   :  39.30   Min.   :42.00   Min.   :45.30   Min.   :29.50   Min.   : 88.0   Min.   :1.000   Min.   : 5441
 1st Qu.:53.00   1st Qu.:56.70   1st Qu.:  53.50   1st Qu.:55.00   1st Qu.:54.00   1st Qu.:56.00   1st Qu.:100.0   1st Qu.:1.000   1st Qu.: 6541
 Median :61.00   Median :61.20   Median :  57.40   Median :59.00   Median :61.60   Median :60.30   Median :103.0   Median :2.000   Median : 7298
 Mean   :60.48   Mean   :60.81   Mean   :  97.79   Mean   :59.83   Mean   :62.53   Mean   :60.43   Mean   :102.5   Mean   :2.327   Mean   : 7270
 3rd Qu.:70.00   3rd Qu.:65.00   3rd Qu.:  60.70   3rd Qu.:64.80   3rd Qu.:69.50   3rd Qu.:64.80   3rd Qu.:106.0   3rd Qu.:3.000   3rd Qu.: 7872
 Max.   :95.00   Max.   :81.30   Max.   :7603.00   Max.   :81.50   Max.   :96.70   Max.   :83.50   Max.   :116.0   Max.   :4.000   Max.   :10164
    cenroll         cattend         freelnch          lep             speced          minority        white2          black2          hispanc2
 Min.   :  110   Min.   :93.20   Min.   : 1.00   Min.   : 0.000   Min.   : 9.70   Min.   : 0.50   Min.   :  192   Min.   :    0   Min.   :   0.0
 1st Qu.: 1936   1st Qu.:95.20   1st Qu.:10.00   1st Qu.: 0.100   1st Qu.:13.50   1st Qu.: 2.00   1st Qu.: 1497   1st Qu.:    2   1st Qu.:  11.0
 Median : 3993   Median :95.80   Median :16.00   Median : 0.300   Median :15.30   Median : 4.80   Median : 3675   Median :   24   Median :  86.0
 Mean   : 7867   Mean   :95.68   Mean   :20.21   Mean   : 2.175   Mean   :16.01   Mean   :15.44   Mean   : 5597   Mean   : 1708   Mean   : 316.5
 3rd Qu.:11428   3rd Qu.:96.30   3rd Qu.:25.00   3rd Qu.: 1.100   3rd Qu.:18.10   3rd Qu.:17.40   3rd Qu.: 9320   3rd Qu.: 2213   3rd Qu.: 149.0
 Max.   :41359   Max.   :97.10   Max.   :71.00   Max.   :30.000   Max.   :24.60   Max.   :96.30   Max.   :19009   Max.   :24196   Max.   :3234.0
     asian2          aindian2         multi2          total2          thrdadm          thrdtech         avgage2         avgexp2         avgsal2
 Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   :    2   Min.   :  16.0   Min.   :  1.00   Min.   :38.80   Min.   : 9.20   Min.   :35538
 1st Qu.:  3.00   1st Qu.: 1.00   1st Qu.: 11.0   1st Qu.:   36   1st Qu.: 155.0   1st Qu.:  7.40   1st Qu.:41.70   1st Qu.:14.50   1st Qu.:39553
 Median : 19.00   Median : 5.00   Median : 34.0   Median :  170   Median : 305.0   Median : 15.39   Median :43.00   Median :15.60   Median :42204
 Mean   : 95.47   Mean   :13.02   Mean   :145.1   Mean   : 2278   Mean   : 647.3   Mean   : 30.61   Mean   :42.72   Mean   :15.16   Mean   :41629
 3rd Qu.:160.00   3rd Qu.:24.00   3rd Qu.:269.0   3rd Qu.: 3754   3rd Qu.: 895.0   3rd Qu.: 39.81   3rd Qu.:43.70   3rd Qu.:16.20   3rd Qu.:43007
 Max.   :551.00   Max.   :52.00   Max.   :793.0   Max.   :26796   Max.   :3782.0   Max.   :189.16   Max.   :46.80   Max.   :18.90   Max.   :49064
    thrdaide        passla2         passmth2        passbth2        tmnnce2         rmdnce2         lamdnce2        mmdnce2         tmdnce2
 Min.   : 0.00   Min.   :44.00   Min.   :52.00   Min.   :32.00   Min.   :47.70   Min.   :45.50   Min.   :46.30   Min.   :47.70   Min.   :30.00
 1st Qu.: 0.00   1st Qu.:66.00   1st Qu.:69.00   1st Qu.:55.00   1st Qu.:58.30   1st Qu.:55.30   1st Qu.:56.30   1st Qu.:56.60   1st Qu.:56.20
 Median : 3.00   Median :71.00   Median :73.00   Median :63.00   Median :60.20   Median :57.70   Median :60.40   Median :63.20   Median :62.20
 Mean   : 5.98   Mean   :69.81   Mean   :72.97   Mean   :60.62   Mean   :59.76   Mean   :57.13   Mean   :60.12   Mean   :62.34   Mean   :59.98
 3rd Qu.: 9.50   3rd Qu.:76.00   3rd Qu.:80.00   3rd Qu.:67.00   3rd Qu.:62.50   3rd Qu.:59.60   3rd Qu.:63.60   3rd Qu.:66.90   3rd Qu.:63.60
 Max.   :22.00   Max.   :92.00   Max.   :96.00   Max.   :89.00   Max.   :73.00   Max.   :70.30   Max.   :76.90   Max.   :84.50   Max.   :77.60
    rmediate
 Min.   : 7.00
 1st Qu.:16.00
 Median :18.00
 Mean   :20.05
 3rd Qu.:23.00
 Max.   :40.00
```

```
> mod4.7 =  lmer(geread~1+(1|school/class), data = ach)
> summary(mod4.7)
Linear mixed model fit by REML ['lmerMod']
Formula: geread ~ 1 + (1 | school/class)
   Data: ach

REML criterion at convergence: 46146

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3052 -0.6290 -0.2094  0.3049  3.8673

Random effects:
 Groups       Name        Variance Std.Dev.
 class:school (Intercept) 0.2727   0.5222
 school       (Intercept) 0.3118   0.5584
 Residual                 4.8470   2.2016
Number of obs: 10320, groups:  class:school, 568; school, 160
```

4.2 `lme4` for Three and More Levels

As we will see below, defining and fitting three-level models using `lme4` is very similar in most ways to doing so with `lme` and is also closely aligned with fitting two-level models with `lme4`. In this section, we will demonstrate the syntax and output for `lme4` using the examples described above with `nlme`. To fit the null model including only the random intercept, classroom nested in school, and school with reading as the dependent variable, we would use the following syntax. Note that as with `lme`, we specify the nested data structure by `school/class`, which is denoted as a random effect by its inclusion in the parentheses.

```
Model4.7 <- lmer(gread~1+(1|school/class), data = Achieve)
```

To examine the resulting output of our analysis, we will use the `summary` command.

```
summary(Model4.7)
Linear mixed model fit by REML
Formula: gread ~ 1 + (1 | school/class)
   Data: Achieve
   AIC   BIC logLik deviance REMLdev
 46154 46183 -23073    46142   46146
Random effects:
 Groups        Name        Variance Std.Dev.
 class:school  (Intercept)  0.27265  0.52216
 school        (Intercept)  0.31181  0.55840
 Residual                   4.84700  2.20159
Number of obs: 10320, groups: class:school, 568; school, 160
Fixed effects:
             Estimate    Std. Error     t value
(Intercept)     4.30806       0.05499       78.34
```

With the exception of rounding errors, these results are essentially identical to those obtained using `lme`. Specifically, the variance associated with class nested in school is 0.273, while that associated with school is 0.312, and residual variance is 4.847. We can also obtain confidence intervals for the random effects in the model using the MCMC approach discussed in Chapter 3.

```
Model4.7.pvals<-pvals.fnc(Model4.7, nsim = 10000, withMCMC = TRUE)
Model4.7.pvals$random
       Groups        Name   Std.Dev. MCMCmedian  MCMCmean HPD95lower HPD95upper
1 class:school (Intercept)    0.5222     0.4574    0.4572     0.3926     0.5235
2       school (Intercept)    0.5584     0.5399    0.5416     0.4550     0.6310
3     Residual                2.2016     2.2094    2.2095     2.1786     2.2406
```

Because the confidence intervals for each term exclude 0, we can conclude from these results that each of the terms included in the model was related to

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.30806    0.05499   78.34
> confint(mod4.7)
Computing profile confidence intervals ...
                2.5 %    97.5 %
.sig01      0.4516596 0.5958275
.sig02      0.4658891 0.6561674
.sigma      2.1710519 2.2328459
(Intercept) 4.1997929 4.4160128



> Model4.8 <- lmer( geread~gevocab+clenroll+cenroll+(1|school/
+ class), data = ach)
Warning message:
Some predictor variables are on very different scales: consider rescaling
> summary(Model4.8)
Linear mixed model fit by REML ['lmerMod']
Formula: geread ~ gevocab + clenroll + cenroll + (1 | school/class)
   Data: ach

REML criterion at convergence: 43130.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2212 -0.5673 -0.2079  0.3184  4.4736

Random effects:
 Groups       Name        Variance Std.Dev.
 class:school (Intercept) 0.09047  0.3008
 school       (Intercept) 0.07652  0.2766
 Residual                 3.69790  1.9230
Number of obs: 10320, groups:  class:school, 568; school, 160

Fixed effects:
             Estimate Std. Error t value
(Intercept)  1.675e+00  2.081e-01    8.05
gevocab      5.076e-01  8.427e-03   60.23
clenroll     1.899e-02  9.559e-03    1.99
cenroll     -3.721e-06  3.642e-06   -1.02

Correlation of Fixed Effects:
         (Intr) gevocb clnrll
gevocab  -0.124
clenroll -0.961 -0.062
cenroll  -0.134  0.025 -0.007
> anova(mod4.7, Model4.8)
refitting model(s) with ML (instead of REML)
Data: ach
Models:
mod4.7: geread ~ 1 + (1 | school/class)
Model4.8: geread ~ gevocab + clenroll + cenroll + (1 | school/class)
         Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
mod4.7    4 46150 46179 -23071    46142
Model4.8  7 43101 43152 -21544    43087 3054.6      3  < 2.2e-16 ***
```

the outcome variable. In other words, differences were noted in reading scores across the classrooms within schools and across the schools themselves.

Now that we see how to fit three-level models using `lmer`, we can fit a more complex model including the predictor variables of student vocabulary (`gevocab`), the size of the student's class (`clenroll`), and the size of the student's school (`cenroll`) as fixed effects. We continue to fit the three-level model with class nested in school as before. The syntax for fitting this model in `lmer` and obtaining the resultant output is

```
Model4.8 <- lmer(geread~gevocab+clenroll+cenroll+(1|school/
                 class), data = Achieve)

summary(Model4.8)
Linear mixed model fit by REML
Formula: geread ~ gevocab + clenroll + cenroll + (1 | school/
          class)
   Data: Achieve
   AIC   BIC logLik deviance REMLdev
 43145 43196 -21565    43087   43131
Random effects:
 Groups               Name     Variance    Std.Dev.
 class:school   (Intercept)    0.090473     0.30079
 school         (Intercept)    0.076518     0.27662
 Residual                      3.697895     1.92299
Number of obs: 10320, groups: class:school, 568; school, 160

Fixed effects:
                Estimate    Std. Error    t value
(Intercept)    1.675e+00     2.081e-01       8.05
gevocab        5.076e-01     8.426e-03      60.23
clenroll       1.898e-02     9.558e-03       1.99
cenroll       -3.721e-06     3.641e-06      -1.02

Correlation of Fixed Effects:
           (Intr)    gevocb   clnrll
gevocab    -0.124
clenroll   -0.961    -0.062
cenroll    -0.134     0.025   -0.007
```

When interpreting these results, we first want to consider whether this more complex model fits the data better than the simpler null model that does not include the three fixed predictors. The AIC and BIC values for Model 4.8 are 43145 and 43196, respectively. They are lower than those for the null model (Model 4.7)—46154 and 46183. As we noted previously, lower values of these information indices indicate better fit, thereby leading us to the conclusion that the model including the fixed effects provides superior fit.

We will now examine the parameter estimates for the three fixed effects. We see that vocabulary and class size are both positively related to reading

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> Model4.10 <- lmer( geread~1+(1|corp/school/class), data = ach) # a four level version
> summary(Model4.10)
Linear mixed model fit by REML ['lmerMod']
Formula: geread ~ 1 + (1 | corp/school/class)
   Data: ach

REML criterion at convergence: 46103.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.2995 -0.6305 -0.2131  0.3029  3.9448

Random effects:
 Groups               Name        Variance Std.Dev.
 class:(school:corp) (Intercept) 0.27540  0.5248
 school:corp         (Intercept) 0.08748  0.2958
 corp                (Intercept) 0.17726  0.4210
 Residual                        4.84699  2.2016
Number of obs: 10320, groups:  class:(school:corp), 568; school:corp, 160; corp, 59

Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.32583    0.07198    60.1
>
```

almost matches mod 4.7

to estimate this model and obtain the output appear below. We see that the four-level nested structure is simply an amplification of the three-level structure in which the higher levels appear first in the list, separated by slashes (/).

```
Model4.10 <- lmer(geread~1+(1|corp/school/class), data =
                  Achieve)
summary(Model4.10)

Linear mixed model fit by REML
Formula: geread ~ 1 + (1 | corp/school/class)
   Data: Achieve
   AIC   BIC logLik deviance REMLdev
 46113 46149 -23052    46100   46103
Random effects:
 Groups                      Name    Variance    Std.Dev.
 class:(school:corp)  (Intercept)    0.275399    0.52478
 school:corp          (Intercept)    0.087452    0.29572
 corp                 (Intercept)    0.177256    0.42102
 Residual                            4.846993    2.20159
Number of obs: 10320, groups: class:(school:corp), 568;
 school:corp, 160; corp, 59

Fixed effects:
              Estimate   Std. Error    t value
(Intercept)    4.32583      0.07196      60.11
```

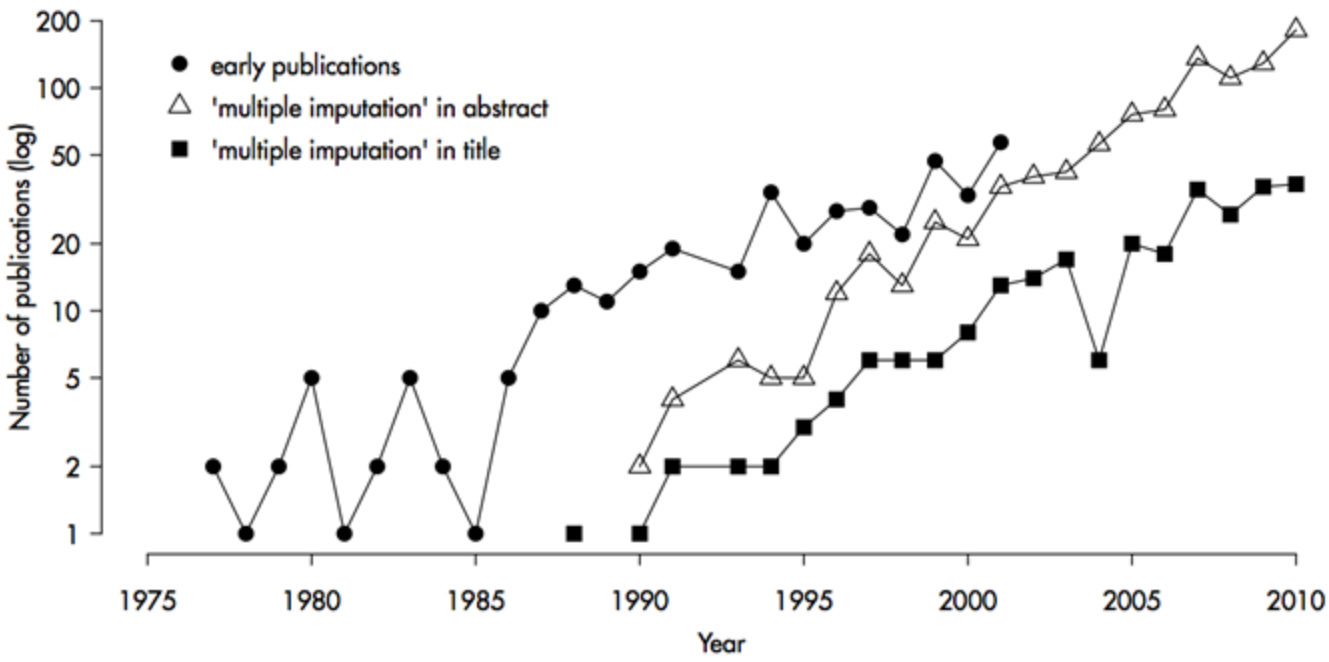
Based on these results, we conclude that classroom nested within school within corporation accounts for the largest share of score variance, followed by corporation, and finally school nested within corporation. The MCMC hypothesis random effects confidence intervals for this model appear below. The fact that not one includes 0 indicates that at each level of the data, there were between-cluster differences in average reading performance.

```
Model4.10.pvals<-pvals.fnc(Model4.10, nsim = 10000, withMCMC = TRUE)
Model4.10.pvals$random
              Groups       Name  Std.Dev. MCMCmedian MCMCmean HPD95lower HPD95upper
1 class:(school:corp) (Intercept)   0.5248     0.4606   0.4605     0.3943     0.5281
2        school:corp (Intercept)   0.2957     0.2999   0.2984     0.1895     0.4054
3               corp (Intercept)   0.4210     0.4203   0.4235     0.3147     0.5327
4           Residual              2.2016     2.2086   2.2086     2.1791     2.2405
```

Using `lmer`, it is possible to estimate a random slopes model in which the coefficient linking a fixed effect to the outcome variable is allowed to vary by level of the random effect. In the case of a three-level data structure, we can fit a random slopes model such that the coefficient is allowed to vary for both random effects simultaneously. In the current example, this would mean allowing the coefficient for a fixed effect (e.g., gender) to vary by classroom nested in school and by school. The R command sequence for fitting a model with a random intercept, and a random coefficient for gender, using `lmer` appears below.

www.multiple-imputation.com

Home
MI
MICE
FIMD
Software
Contact



Number of publications (log) on multiple imputation during the period 1977–2010 according to three counting methods.

The figure contains three time series with counts of the number of publications on multiple imputation during the period 1977–2010. The search was done in Scopus on the July 11, 2011. Counts were made in three ways.

1. The right most series corresponds to the number of publications per year that featured the search term `multiple imputation' in the title. These are often methodological articles in which new adaptations are being developed.
2. The series in the middle is the number of publication that featured `multiple imputation' in the title, abstract or key words on the same search data. This set includes a growing group of papers that contain applications.
3. The left most series is the number of publications in a collection of early publications. This collection covers essentially everything related to multiple imputation since its inception in 1977 up to the year 2001. This group also includes chapters in books, dissertations, conference proceedings, technical reports, and so on.

Note that the vertical axis is set in the logarithm. Perhaps the most interesting series is the middle series number of applications is growing at an exponential rate.

Rubin on Stef

Foreword

I'm delighted to see this new book on multiple imputation by Stef van Buuren for several reasons. First, to me at least, having another book devoted to multiple imputation marks the maturing of the topic after an admittedly somewhat shaky initiation. Stef is certainly correct when he states in Section 2.1.2: "The idea to create multiple versions must have seemed outrageous at that time [late 1970s]. Drawing imputations from a distribution, instead of estimating the 'best' value, was a severe breach with everything that had been done before." I remember how this idea of multiple imputation was even ridiculed by some more traditional statisticians, sometimes for just being "silly" and sometimes for being hopelessly inefficient with respect to storage demands and outrageously expensive with respect to computational requirements.

Some others of us foresaw what was happening to both (a) computational storage (I just acquired a 64 GB flash drive the size of a small finger for under $60, whereas only a couple of decades ago I paid over $2500 for a 120 KB hard-drive larger than a shoe box weighing about 10 kilos), and (b) computational speed and flexibility. To develop statistical methods for the future while being bound by computational limitations of the past was clearly inapposite. Multiple imputation's early survival was clearly due to the insight of a younger generation of statisticians, including many colleagues and former students, who realized future possibilities.

A second reason for my delight at the publication of this book is more personal and concerns the maturing of the author, Stef van Buuren. As he mentions, we first met through Jan van Rijckevorsel at TNO. Stef was a young and enthusiastic researcher there, who knew little about the kind of statistics that I felt was essential for making progress on the topic of dealing with missing data. But consider the progress over the decades starting with his earlier work on MICE! Stef has matured into an independent researcher making important and original contributions to the continued development of multiple imputation.

This book represents a "no nonsense" straightforward approach to the application of multiple imputation. I particularly like Stef's use of graphical displays, which are badly needed in practice to supplement the more theoretical discussions of the general validity of multiple imputation methods. As I have said elsewhere, and as implied by much of what is written by Stef, "It's not that multiple imputation is so good; it's really that other methods for addressing missing data are so bad." It's great to have Stef's book on mul-

© 2012 by Taylor & Francis Group, LLC

```
> ### fimd1.R
> ### R code from Van Buuren, S. (2012).
> ###           Flexible Imputation of Missing Data.
> ###           CRC/Chapman & Hall, Boca Raton, FL.
> ### (c) 2012 Stef van Buuren, www.multiple-imputation.com
> ### Version 1, 21mar2012
> ### Tested with Mac OS X 10.7.3, R2.14-2, mice 2.12
>
> ### Chapter 1         Introduction
> ### Section 1.1 The problem of missing data
>
> ### calculate the mean of three numbers
> y <- c(1, 2, 4)

> mean(y)
[1] 2.333333

> ### calculate the mean of three numbers, where one is missing
> y <- c(1, 2, NA)

> mean(y)
[1] NA

> ### repeat, but with any missing data removed
> mean(y, na.rm=TRUE)
[1] 1.5

> ### store your current options (safety measure)
> youroptions <- options()

> ### demo regression that generates an error message using default na.fail
> options(na.action = na.fail)

> ### uncomment to see the error
> ### lm(Ozone ~ Wind, data=airquality)
>
> ### remove incomplete observations by 'na.action = na.omit'
> fit <- lm(Ozone ~ Wind, data = airquality, na.action = na.omit)

> coef(fit)
(Intercept)        Wind
  96.872895   -5.550923

> ### set automatic na.action = na.omit
> options(na.action = na.omit)

> ### find out how many rows were deleted
> deleted <- na.action(fit)

> naprint(deleted)
[1] "37 observations deleted due to missingness"

> ### more incomplete rows if we add Solar.R as predictor
> fit2 <- lm(Ozone ~ Wind+Solar.R, data=airquality)

> naprint(na.action(fit2))
[1] "42 observations deleted due to missingness"

> ### restore your original options, but set na.omit
> options(youroptions)

> options(na.action=na.omit)
```

cor {stats} R Documentation

Correlation, Variance and Covariance (Matrices)

Description

`var`, `cov` and `cor` compute the variance of `x` and the covariance or correlation of `x` and `y` if these are vectors. If `x` and `y` are matrices then the covariances (or correlations) between the columns of `x` and the columns of `y` are computed.

`cov2cor` scales a covariance matrix into the corresponding correlation matrix *efficiently*.

Usage

```
var(x, y = NULL, na.rm = FALSE, use)

cov(x, y = NULL, use = "everything",
    method = c("pearson", "kendall", "spearman"))

cor(x, y = NULL, use = "everything",
    method = c("pearson", "kendall", "spearman"))

cov2cor(V)
```

Arguments

`x`	a numeric vector, matrix or data frame.
`y`	`NULL` (default) or a vector, matrix or data frame with compatible dimensions to `x`. The default is equivalent to `y = x` (but more efficient).
`na.rm`	logical. Should missing values be removed?
`use`	an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings `"everything"`, `"all.obs"`, `"complete.obs"`, `"na.or.complete"`, or `"pairwise.complete.obs"`.
`method`	a character string indicating which correlation coefficient (or covariance) is to be computed. One of `"pearson"` (default), `"kendall"`, or `"spearman"`, can be abbreviated.
`V`	symmetric numeric matrix, usually positive definite such as a covariance matrix.

Details

For `cov` and `cor` one must *either* give a matrix or data frame for `x` *or* give both `x` and `y`.

The inputs must be numeric (as determined by `is.numeric`: logical values are also allowed for historical compatibility): the `"kendall"` and `"spearman"` methods make sense for ordered inputs but `xtfrm` can be used to find a suitable prior transformation to numbers.

`var` is just another interface to `cov`, where `na.rm` is used to determine the default for `use` when that is unspecified. If `na.rm` is `TRUE` then the complete observations (rows) are used (`use = "na.or.complete"`) to compute the variance. Otherwise, by default `use = "everything"`.

If `use` is `"everything"`, NAs will propagate conceptually, i.e., a resulting value will be `NA` whenever one of its contributing observations is `NA`.
If `use` is `"all.obs"`, then the presence of missing observations will produce an error. If `use` is `"complete.obs"` then missing values are handled by casewise deletion (and if there are no complete cases, that gives an error).
`"na.or.complete"` is the same unless there are no complete cases, that gives `NA`. Finally, if `use` has the value `"pairwise.complete.obs"` then the correlation or covariance between each pair of variables is computed using all complete pairs of observations on those variables. This can result in covariance or correlation matrices which are not positive semi-definite, as well as `NA` entries if there are no complete pairs for that pair of variables. For `cov` and `var`, `"pairwise.complete.obs"` only works with the `"pearson"` method. Note that (the equivalent of) `var(double(0), use = *)` gives `NA` for `use = "everything"` and `"na.or.complete"`, and gives an error in the other cases.

The denominator *n - 1* is used which gives an unbiased estimator of the (co)variance for i.i.d. observations. These functions return `NA` when there is only one observation (whereas S-PLUS has been returning `NaN`), and fail if `x` has length zero.

For `cor()`, if `method` is `"kendall"` or `"spearman"`, Kendall's *tau* or Spearman's *rho* statistic is used to estimate a rank-based

measure of association. These are more robust and have been recommended if the data do not necessarily come from a bivariate normal distribution.
For `cov()`, a non-Pearson method is unusual but available for the sake of completeness. Note that `"spearman"` basically computes `cor(R(x), R(y))` (or `cov(., .)`) where `R(u) := rank(u, na.last = "keep")`. In the case of missing values, the ranks are calculated depending on the value of `use`, either based on complete observations, or based on pairwise completeness with reranking for each pair.

Scaling a covariance matrix into a correlation one can be achieved in many ways, mathematically most appealing by multiplication with a diagonal matrix from left and right, or more efficiently by using `sweep(.., FUN = "/")` twice. The `cov2cor` function is even a bit more efficient, and provided mostly for didactical reasons.

Value

For `r <- cor(*, use = "all.obs")`, it is now guaranteed that `all(r <= 1)`.

Note

Some people have noted that the code for Kendall's tau is slow for very large datasets (many more than 1000 cases). It rarely makes sense to do such a computation, but see function `cor.fk` in package **pcaPP**.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

`cor.test` for confidence intervals (and tests).

`cov.wt` for *weighted* covariance computation.

`sd` for standard deviation (vectors).

Examples

```
var(1:10)  # 9.166667

var(1:5, 1:5) # 2.5

## Two simple vectors
cor(1:10, 2:11) # == 1

## Correlation Matrix of Multivariate sample:
(Cl <- cor(longley))
## Graphical Correlation Matrix:
symnum(Cl) # highly correlated

## Spearman's rho  and  Kendall's tau
symnum(clS <- cor(longley, method = "spearman"))
symnum(clK <- cor(longley, method = "kendall"))
## How much do they differ?
i <- lower.tri(Cl)
cor(cbind(P = Cl[i], S = clS[i], K = clK[i]))


## cov2cor() scales a covariance matrix by its diagonal
##           to become the correlation matrix.
cov2cor # see the function definition {and learn ..}
stopifnot(all.equal(Cl, cov2cor(cov(longley))),
          all.equal(cor(longley, method = "kendall"),
            cov2cor(cov(longley, method = "kendall"))))

##--- Missing value treatment:

C1 <- cov(swiss)
range(eigen(C1, only.values = TRUE)$values) # 6.19        1921

## swM := "swiss" with  3 "missing"s :
swM <- swiss
```

```
> ssdat = read.table("D:\\drr13\\ed493\\ss423", header = T)
> dim(ssdat)
[1] 137  12
> summary(ssdat)
     grade           gender       schoolgrades        Age
 Min.   :6.000   Min.   :1.000   Min.   :1.000   Min.   :11.00
 1st Qu.:6.000   1st Qu.:1.000   1st Qu.:4.000   1st Qu.:12.00
 Median :7.000   Median :2.000   Median :6.000   Median :13.00
 Mean   :6.993   Mean   :1.526   Mean   :5.762   Mean   :12.92
 3rd Qu.:8.000   3rd Qu.:2.000   3rd Qu.:8.000   3rd Qu.:14.00
 Max.   :8.000   Max.   :2.000   Max.   :9.000   Max.   :17.00
                                 NA's   :7
   SpecialEd            SES             belong         hopeless
 Min.   :0.00000   Min.   :1.000   Min.   :32.00   Min.   :17.00
 1st Qu.:0.00000   1st Qu.:3.000   1st Qu.:52.00   1st Qu.:18.00
 Median :0.00000   Median :3.000   Median :61.00   Median :20.00
 Mean   :0.07353   Mean   :3.161   Mean   :61.77   Mean   :20.46
 3rd Qu.:0.00000   3rd Qu.:3.000   3rd Qu.:72.00   3rd Qu.:23.00
 Max.   :1.00000   Max.   :5.000   Max.   :89.00   Max.   :30.00
 NA's   :1
    depress          master           idea           behavior
 Min.   :10.00   Min.   : 7.00   Min.   : 7.000   Min.   :4.000
 1st Qu.:14.00   1st Qu.:40.00   1st Qu.: 7.000   1st Qu.:4.000
 Median :20.00   Median :54.00   Median : 7.000   Median :4.000
 Mean   :19.68   Mean   :50.79   Mean   : 7.613   Mean   :4.467
 3rd Qu.:25.00   3rd Qu.:61.00   3rd Qu.: 8.000   3rd Qu.:5.000
 Max.   :37.00   Max.   :80.00   Max.   :14.000   Max.   :8.000
 NA's   :1
> ssdat$depress
  [1] 14 19 23 16 17 15 29 29 11 15 10 22 15 12 20 16 19 11 26 20 14 13 13 23 24 15 29
 [31] 14 28 23 22 24 22 20 17 29 25 21 27 17 27 10 28 28 14 13 13 17 13 16 25 28 21 12
 [61] 10 23 32 27 18 18 25 13 12 27 20 18 28 33 16 26 12 15 26 10 27 27 11 18 23 25 19
 [91] 22 18 15 22 25 20 21 20 12 27 16 37 25 13 27 12 15 18 30 20 26 12 24 23 25 16 15
[121] 33 12 19 22 15 22 12 22 13 10 23 10 12 27 10 15 24
> attach(ssdat)
> cor(ssdat, use = "pairwise.complete.obs")
                    grade      gender schoolgrades          Age   SpecialEd          SE
grade         1.000000000  0.12007161   0.10324966  0.734537653  0.03288994  0.00210796
gender        0.120071606  1.00000000  -0.07693380  0.111785222 -0.01660040  0.08280056
schoolgrades  0.103249658 -0.07693380   1.00000000 -0.036878790 -0.12149464  0.08078070
Age           0.734537653  0.11178522  -0.03687879  1.000000000  0.17137161 -0.04921041
SpecialEd     0.032889941 -0.01660040  -0.12149464  0.171371614  1.00000000 -0.02857142
SES           0.002107961  0.08280056   0.08078070 -0.049210411 -0.02857143  1.00000000
belong        0.165002057  0.13290907   0.32461877  0.020568292 -0.12735146  0.31256444
hopeless     -0.043621530 -0.11000899  -0.26012870  0.033950951  0.04260438 -0.35331780
depress       0.041356609 -0.14628074  -0.23805195  0.114296820  0.04913473 -0.25545168
master        0.178163488  0.26860484   0.25281286  0.063919356  0.03180739  0.34131229
idea         -0.180486806 -0.18685063  -0.06102518 -0.005341527  0.08341562 -0.12990632
behavior     -0.267367122 -0.17665021  -0.14875906 -0.083403825 -0.01597250 -0.11311648
                  belong    hopeless     depress      master         idea    behavior
grade         0.16500206 -0.04362153  0.04135661  0.17816349 -0.180486806 -0.26736712
gender        0.13290907 -0.11000899 -0.14628074  0.26860484 -0.186850627 -0.17665021
schoolgrades  0.32461877 -0.26012870 -0.23805195  0.25281286 -0.061025178 -0.14875906
Age           0.02056829  0.03395095  0.11429682  0.06391936 -0.005341527 -0.08340382
SpecialEd    -0.12735146  0.04260438  0.04913473  0.03180739  0.083415617 -0.01597250
SES           0.31256444 -0.35331781 -0.25545169  0.34131230 -0.129906324 -0.11311649
belong        1.00000000 -0.53386517 -0.50984362  0.59237433 -0.249387198 -0.16690321
hopeless     -0.53386517  1.00000000  0.50487406 -0.62739248  0.221453622  0.20342006
depress      -0.50984362  0.50487406  1.00000000 -0.46763766  0.292176215  0.27719693
master        0.59237433 -0.62739248 -0.46763766  1.00000000 -0.187823985 -0.23182823
```

```
idea         -0.24938720  0.22145362  0.29217622 -0.18782399  1.000000000  0.72197708
behavior     -0.16690321  0.20342006  0.27719693 -0.23182823  0.721977076  1.00000000

> reg1 = lm(depress ~ belong)
> summary(reg1)

Call:
lm(formula = depress ~ belong)

Residuals:
    Min      1Q  Median      3Q     Max
-14.791  -3.491   0.109   3.566  16.323

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.59112    2.36527  15.047  < 2e-16 ***
belong      -0.25715    0.03748  -6.861  2.3e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.386 on 134 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.2599,     Adjusted R-squared: 0.2544
F-statistic: 47.07 on 1 and 134 DF,  p-value: 2.303e-10

> reg2 = lm(schoolgrades ~ belong)
> reg3 = lm(depress ~ schoolgrades + belong)
> summary(reg3)

Call:
lm(formula = depress ~ schoolgrades + belong)

Residuals:
     Min       1Q   Median       3Q      Max
-15.3857  -3.4017  -0.1492   3.6832  15.7179

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.99046    2.56219  14.047  < 2e-16 ***
schoolgrades -0.26774    0.25091  -1.067    0.288
belong       -0.23974    0.04172  -5.747 6.38e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.472 on 127 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared: 0.2514,     Adjusted R-squared: 0.2396
F-statistic: 21.32 on 2 and 127 DF,  p-value: 1.039e-08


> summary(reg2)

Call:
lm(formula = schoolgrades ~ belong)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2527 -1.2272  0.2203  1.5314  3.5029

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.42056    0.87688   2.760 0.006621 **
belong       0.05397    0.01390   3.883 0.000165 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.928 on 128 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared: 0.1054,     Adjusted R-squared: 0.09839
F-statistic: 15.08 on 1 and 128 DF,  p-value: 0.0001645
```

Y_i outcomes
X_i covariates
r_i elements 1 not missing
0 missing

Schafer talk
also LDA ch 17

STAT 222
week 9

Classification of DOM's

Distribution of Missingness (Rubin mechanism)

Based on Rubin (1976), Little and Rubin (1987), and Little (1995)

- **Missing completely at random (MCAR):** DOM does not depend on covariates or outcomes

LDA 17.2.1

complete cases a random sample from population

$$P(r_i|x_i, y_i, \phi) = P(r_i|\phi)$$

- **Covariate-dependent (CD) missingness:** DOM may possibly depend on covariates but not outcomes

sometimes MAR

$$P(r_i|x_i, y_i, \phi) = P(r_i|x_i, \phi)$$

- **Missing at random (MAR):** DOM may depend on covariates and observed outcomes

LDA 17.2.2

analogy: random assignment vs regression discontinuity

lmer random effects models

$$P(r_i|x_i, y_i, \phi) = P(r_i|x_i, y_{i(obs)}, \phi)$$

Note that MCAR $\subset$ CD $\subset$ MAR.

conditional on observed values

- **Missing not at random (MNAR):** Any violation of MAR; DOM still depends on $y_{i(mis)}$ even after any dependence on x_i and $y_{i(obs)}$ has been accounted for

LDA 17.2.3
joint model 17.3

missingness depends on unmeasured

sensitivity analysis
LDA ch. 22

TABLE 3.

Summary of Available Options for Handling Missing Data

Action	Pros	Cons	Recommendation
Careful deliberation about why data are missing	Helps determine correct model	Not always clear	Always do this
Last observation carried forward	Easy to do	Unrealistic assumption, overestimates precision	Never do this
Mean imputation	Easy to do, preserves mean	Does not preserve relationships in data. Overestimates precision.	Never do this
Complete-case analysis	Easy to do	Biased estimates unless data are MCAR. Loss of information.	Never do this.
End-point analysis	Missing values no longer an issue	Ignores information, ignores time	Never do this.
Single imputation	Reduces bias	Overestimates precision.	Use multiple imputation.
Mixed-effects regression model	Makes use of all available information	Can be complicated to fit. Assumes MAR.	Often a good choice
Multiple imputation	Allows one to incorporate auxiliary variables into imputation model	Requires expertise. Additional steps for analyzing data.	Do it if MAR assumption is likely to be satisfied
Nonignorable models	Explore the effect of different missing data assumptions	Not clear what is correct model. Can be complicated.	Worth doing, especially as a sensitivity analysis
Fit several different statistical models that make different assumptions regarding why data are missing	Sensitivity of inferences to different assumptions regarding the missing data mechanism.	Additional work. May complicate the overall picture.	Worth doing.

Lavori, part B

the study should be included in the imputa-

MI, irrespective of the inclusion of dosage

to adequately handle missing data may

References

White, I. R., Royston, P. (2009). Imputing missing covariate values for the Cox model. *Statistics in Medicine*, *28*(15), 1982-1998.

van Buuren, S. (2012). *Flexible Imputation of Missing Data.* Boca Raton, FL: Chapman \& Hall/CRC Press.

Examples

```
leuk$status <- 1  ## no censoring occurs in leuk data (MASS)
ch <- nelsonaalen(leuk, time, status)
plot(x = leuk$time, y = ch, ylab="Cumulative hazard", xlab="Time")

### See example on http://www.engineeredsoftware.com/lmar/pe_cum_hazard_function.htm
time <- c(43, 67, 92, 94, 149, rep(149,7))
status <- c(rep(1,5),rep(0,7))
eng <- data.frame(time, status)
ch <- nelsonaalen(eng, time, status)
plot(x = time, y = ch, ylab="Cumulative hazard", xlab="Time")
```

---

nhanes *NHANES example - all variables numerical*

Description

A small data set with non-monotone missing values.

Usage

```
data(nhanes)
```

Format

A data frame with 25 observations on the following 4 variables.

`age` Age group (1=20-39, 2=40-59, 3=60+)

`bmi` Body mass index (kg/m**2)

`hyp` Hypertensive (1=no,2=yes)

`chl` Total serum cholesterol (mg/dL)

Details

A small data set with all numerical variables. The data set `nhanes2` is the same data set, but with `age` and `hyp` treated as factors.

Journal of Statistical Software

December 2011, Volume 45, Issue 3. http://www.jstatsoft.org/

mice: Multivariate Imputation by Chained Equations in R

Stef van Buuren
TNO

Karin Groothuis-Oudshoorn
University of Twente

Abstract

The R package **mice** imputes incomplete multivariate data by chained equations. The software **mice** 1.0 appeared in the year 2000 as an S-PLUS library, and in 2001 as an R package. **mice** 1.0 introduced predictor selection, passive imputation and automatic pooling. This article documents **mice** 2.9, which extends the functionality of **mice** 1.0 in several ways. In **mice** 2.9, the analysis of imputed data is made completely general, whereas the range of models under which pooling works is substantially extended. **mice** 2.9 adds new functionality for imputing multilevel data, automatic predictor selection, data handling, post-processing imputed values, specialized pooling routines, model selection tools, and diagnostic graphs. Imputation of categorical data is improved in order to bypass problems caused by perfect prediction. Special attention is paid to transformations, sum scores, indices and interactions using passive imputation, and to the proper setup of the predictor matrix. **mice** 2.9 can be downloaded from the Comprehensive R Archive Network. This article provides a hands-on, stepwise approach to solve applied incomplete data problems.

Keywords: MICE, multiple imputation, chained equations, fully conditional specification, Gibbs sampler, predictor selection, passive imputation, R.

1. Introduction

Multiple imputation (Rubin 1987, 1996) is the method of choice for complex incomplete data problems. Missing data that occur in more than one variable presents a special challenge. Two general approaches for imputing multivariate data have emerged: joint modeling (JM) and fully conditional specification (FCS), also known as multivariate imputation by chained equations (MICE). Schafer (1997) developed various JM techniques for imputation under the multivariate normal, the log-linear, and the general location model. JM involves specifying a multivariate distribution for the missing data, and drawing imputation from their conditional

(1997, p. 237). The data contains four variables: `age` (age group), `bmi` (body mass index), `hyp` (hypertension status) and `chl` (cholesterol level). The data are stored as a `data frame`. Missing values are represented as `NA`.

```
R> nhanes
```

```
   age  bmi hyp chl
1    1   NA  NA  NA
2    2 22.7   1 187
3    1   NA   1 187
4    3   NA  NA  NA
5    1 20.4   1 113
     ...
```

Inspecting the missing data

The number of the missing values can be counted and visualized as follows:

```
R> md.pattern(nhanes)
```

```
   age hyp bmi chl
13   1   1   1   1  0
 1   1   1   0   1  1
 3   1   1   1   0  1
 1   1   0   0   1  2
 7   1   0   0   0  3
     0   8   9  10 27
```

There are 13 (out of 25) rows that are complete. There is one row for which only `bmi` is missing, and there are seven rows for which only `age` is known. The total number of missing values is equal to $(7 \times 3) + (1 \times 2) + (3 \times 1) + (1 \times 1) = 27$. Most missing values (10) occur in `chl`.

Another way to study the pattern involves calculating the number of observations per patterns for all pairs of variables. A pair of variables can have exactly four missingness patterns: both variables are observed (pattern `rr`), the first variable is observed and the second variable is missing (pattern `rm`), the first variable is missing and the second variable is observed (pattern `mr`), and both are missing (pattern `mm`). We can use the `md.pairs()` function to calculate the frequency in each pattern for all variable pairs as

```
R> p <- md.pairs(nhanes)
R> p
```

```
$rr
    age bmi hyp chl
age  25  16  17  15
bmi  16  16  16  13
```

```
 # refer to van Buuren S and Groothuis-Oudshoorn K (2011). mice: Multivariate Imputation by Chaine
Equations in R. Journal of Statistical Software, 45(3), 1-67. http://www.jstatsoft.org/v45/i03/
R version 2.14.1 (2011-12-22)
> install.packages("mice") # imputation package
> install.packages("VIM") # additional graphics
>library(mice)
> library(VIM)
> data(nhanes)
> head(nhanes)
  age  bmi hyp chl
1   1   NA  NA  NA
2   2 22.7   1 187
3   1   NA   1 187
4   3   NA  NA  NA
5   1 20.4   1 113
6   3   NA  NA 184
> md.pattern(nhanes) #age complete
   age hyp bmi chl
13   1   1   1   1  0
 1   1   1   0   1  1
 3   1   1   1   0  1
 1   1   0   0   1  2
 7   1   0   0   0  3
     0   8   9  10 27
                          #number of observations per patterns for all pairs of variables
                          > md.pairs(nhanes)      $mr
                          $rr                         age bmi hyp chl
                              age bmi hyp chl     age   0   0   0   0
                          age  25  16  17  15      bmi   9   0   1   2
                          bmi  16  16  16  13      hyp   8   0   0   1
                          hyp  17  16  17  14      chl  10   3   3   0
                          chl  15  13  14  15
                                                  $mm
                          $rm                         age bmi hyp chl
                               age bmi hyp chl    age   0   0   0   0
                          age    0   9   8  10     bmi   0   9   8   7
                          bmi    0   0   0   3     hyp   0   8   8   7
                          hyp    0   1   0   3     chl   0   7   7  10
                          chl    0   2   1   0
                          #multiply imputed data set is stored in the object impdrr of class mids
> impdrr = mice(nhanes, seed =23109) # m=5 imps is default
 iter imp variable
  1   1  bmi  hyp  chl
  1   2  bmi  hyp  chl
  1   3  bmi  hyp  chl
  .....................
  5   3  bmi  hyp  chl
  5   4  bmi  hyp  chl
  5   5  bmi  hyp  chl  #default method, numerical data, predictive mean matching (pmm)
#The complete() function extracts the five imputed data sets from the imp object as a
                                                long (row-stacked) matrix with 125 records
> stripplot(impdrr, pch = 20, cex = 1.2)

#The fit object has class mira and contains the results of 5 complete-data analyses
> fitdrr = with(impdrr, lm(chl ~ age + bmi))
# pool separate results
> round(summary(pool(fitdrr)), 2) # match Stef JSS results
                est    se     t   df Pr(>|t|)   lo 95  hi 95 nmis  fmi lambda
(Intercept) -34.16 76.07 -0.45 6.81     0.67 -215.05 146.73   NA 0.57   0.47
age          34.33 14.86  2.31 4.04     0.08   -6.76  75.42    0 0.75   0.65
bmi           6.21  2.21  2.81 8.80     0.02    1.20  11.23    9 0.48   0.37


> impdrr2 = mice(nhanes, seed =23009) #try again or set m=50, set.seed
> fitdrr2 = with(impdrr2, lm(chl ~ age + bmi))
> round(summary(pool(fitdrr2)), 2) #this is not closer to JSS, diff seed
                est    se     t    df Pr(>|t|)   lo 95  hi 95 nmis  fmi lambda
(Intercept) -32.13 59.67 -0.54 14.04     0.60 -160.08  95.82   NA 0.29   0.20
age          35.76 10.50  3.41  8.52     0.01   11.80  59.72    0 0.49   0.38
bmi           6.12  1.87  3.28 14.86     0.01    2.14  10.11    9 0.27   0.17
>
```

MICE (imputation) ex
from sec 2.4 JSS paper

STAT 222
Week 9

```
R version 2.14.1 (2011-12-22)
> install.packages("mice") # imputation package
> install.packages("VIM") # additional graphics
>library(mice)
> library(VIM)
> data(nhanes)
> head(nhanes)
  age  bmi hyp chl
1   1   NA  NA  NA
2   2 22.7   1 187
3   1   NA   1 187
4   3   NA  NA  NA
5   1 20.4   1 113
6   3   NA  NA 184
> md.pattern(nhanes) #age complete
   age hyp bmi chl
13   1   1   1   1  0
 1   1   1   0   1  1
 3   1   1   1   0  1
 1   1   0   0   1  2
 7   1   0   0   0  3
     0   8   9  10 27
```

#number of observations per patterns for all pairs of vari

```
> md.pairs(nhanes)
$rr
    age bmi hyp chl
age  25  16  17  15
bmi  16  16  16  13
hyp  17  16  17  14
chl  15  13  14  15

$rm
    age bmi hyp chl
age   0   9   8  10
bmi   0   0   0   3
hyp   0   1   0   3
chl   0   2   1   0

$mr
    age bmi hyp chl
age   0   0   0   0
bmi   9   0   1   2
hyp   8   0   0   1
chl  10   3   3   0

$mm
    age bmi hyp chl
age   0   0   0   0
bmi   0   9   8   7
hyp   0   8   8   7
chl   0   7   7  10
```

#multiply imputed data set is stored in the object impdrr of class mids

```
> impdrr = mice(nhanes, seed =23109) # m=5 imps is default
 iter imp variable
  1   1  bmi  hyp  chl
  1   2  bmi  hyp  chl
  1   3  bmi  hyp  chl
  .....................
  5   3  bmi  hyp  chl
  5   4  bmi  hyp  chl
  5   5  bmi  hyp  chl  #default method, numerical data, predictive mean matching (pmm)
#The complete() function extracts the five imputed data sets from the imp object as a
                                    long (row-stacked) matrix with 125 records
> stripplot(impdrr, pch = 20, cex = 1.2)

#The fit object has class mira and contains the results of 5 complete-data analyses
> fitdrr = with(impdrr, lm(chl ~ age + bmi))
# pool separate results
> round(summary(pool(fitdrr)), 2) # doesn't match Stef JSS results all that well
               est    se     t   df Pr(>|t|)   lo 95  hi 95 nmis  fmi lambda
(Intercept) -34.16 76.07 -0.45 6.81     0.67 -215.05 146.73   NA 0.57   0.47
age          34.33 14.86  2.31 4.04     0.08   -6.76  75.42    0 0.75   0.65
bmi           6.21  2.21  2.81 8.80     0.02    1.20  11.23    9 0.48   0.37


> impdrr2 = mice(nhanes, seed =23009) #try again or set m=50, set.seed
> fitdrr2 = with(impdrr2, lm(chl ~ age + bmi))
> round(summary(pool(fitdrr2)), 2) #this is closer to JSS
               est    se     t    df Pr(>|t|)   lo 95  hi 95 nmis  fmi lambda
(Intercept) -32.13 59.67 -0.54 14.04     0.60 -160.08  95.82   NA 0.29   0.20
age          35.76 10.50  3.41  8.52     0.01   11.80  59.72    0 0.49   0.38
bmi           6.12  1.87  3.28 14.86     0.01    2.14  10.11    9 0.27   0.17
>
```

lmer, missing data

```
> ncM = read.table("D:\\drr17\\stat222\\week10\\ncLong_dataM", header = T)
> head(ncM)
      ID time   Y   Z
1 705810    1 380 120
2 705810    2 377 120
3 705810    3 460 120
4 705810    4 472 120
5 705810    5 495 120
6 705810    6 566 120
> # ID 2,3,4, Z made missing
> ncM$timeInt = ncM$time - 1
> summary(ncM)
       ID                time             Y                Z             timeInt
 Min.   :  705810   Min.   :1.00   Min.   :270.0   Min.   : 64.0   Min.   :0.00
 1st Qu.:  847813   1st Qu.:2.75   1st Qu.:395.0   1st Qu.: 97.0   1st Qu.:1.75
 Median : 1046817   Median :4.50   Median :464.0   Median :106.0   Median :3.50
 Mean   : 1461655   Mean   :4.50   Mean   :469.9   Mean   :106.1   Mean   :3.50
 3rd Qu.: 1290819   3rd Qu.:6.25   3rd Qu.:540.0   3rd Qu.:115.0   3rd Qu.:5.25
 Max.   :11090821   Max.   :8.00   Max.   :762.0   Max.   :145.0   Max.   :7.00
                                                   NA's   :24
> ncCon2 = lmer(Y ~  Z*timeInt + ( 1 + timeInt  | ID), data = ncM) # incl Z in slope L2
> summary(ncCon2)
Linear mixed model fit by REML ['lmerMod']
Formula: Y ~ Z * timeInt + (1 + timeInt | ID)
   Data: ncM

REML criterion at convergence: 20256.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.5700 -0.6066 -0.0356  0.5935  3.1695

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 ID       (Intercept) 194.47   13.945
          timeInt      24.96    4.996   0.38
 Residual             401.45   20.036
Number of obs: 2192, groups:  ID, 274

Fixed effects:
             Estimate Std. Error t value
(Intercept) 253.32204    8.84339  28.645
Z             0.83725    0.08261  10.135
timeInt       0.82916    2.73189   0.304
Z:timeInt     0.33587    0.02552  13.162

Correlation of Fixed Effects:
          (Intr) Z      timInt
Z         -0.992
timeInt   -0.065  0.065
Z:timeInt  0.065 -0.065 -0.992
> ## full data week 2 Number of obs: 2216, groups:  ID, 277, 3 ID's deleted from this analys
>
```

```
## see also merTools vignette
R version 3.5.3 (2019-03-11) -- "Great Truth"


> library(merTools)
Loading required package: arm
Loading required package: MASS
Loading required package: Matrix
Loading required package: lme4

arm (Version 1.10-1, built: 2018-4-12)

Working directory is C:/Users/rag/Documents

Loading required package: dplyr

Attaching package: ‘dplyr’

The following object is masked from ‘package:MASS’:

    select

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

> data(hsb)
> str(hsb)
'data.frame':   7185 obs. of  8 variables:
 $ schid   : chr  "1224" "1224" "1224" "1224" ...
 $ minority: num  0 0 0 0 0 0 0 0 0 0 ...
 $ female  : num  1 1 0 0 0 0 1 0 1 0 ...
 $ ses     : num  -1.528 -0.588 -0.528 -0.668 -0.158 ...
 $ mathach : num  5.88 19.71 20.35 8.78 17.9 ...
 $ size    : num  842 842 842 842 842 842 842 842 842 842 ...
 $ schtype : num  0 0 0 0 0 0 0 0 0 0 ...
 $ meanses : num  -0.428 -0.428 -0.428 -0.428 -0.428 -0.428 -0.428 -0.428 -0.428 -0.428 ...
> table(hsb$schtype)

   0    1
3642 3543

> length(unique(hsb$schid))
[1] 160
> library(lme4)
> install.packages("Amelia")

> library(Amelia)
```

```
Loading required package: Rcpp
##
## Amelia II: Multiple Imputation
## (Version 1.7.5, built: 2018-05-07)
## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## Refer to http://gking.harvard.edu/amelia/ for more information
##
> # use vignette function to Create a function to randomly assign NA values
>
> add_NA <- function(x, prob){
+   z <- rbinom(length(x), 1, prob = prob)
+   x[z==1] <- NA
+   return(x)
+ }
> hsb$minority <- add_NA(hsb$minority, prob = 0.05)
> table(is.na(hsb$minority))

FALSE  TRUE
 6800   385
>
>
> hsb$female <- add_NA(hsb$female, prob = 0.05)
> table(is.na(hsb$female))

FALSE  TRUE
 6824   361
>
>
> hsb$ses <- add_NA(hsb$ses, prob = 0.05)
> table(is.na(hsb$ses))

FALSE  TRUE
 6838   347
>
>
> hsb$size <- add_NA(hsb$size, prob = 0.05)
> table(is.na(hsb$size))

FALSE  TRUE
 6828   357
> # about 5% missing on 3 vars, see what lme does
> test_lm = lm(mathach ~ female + minority + ses, data = hsb)
> summary(test_lm)

Call:
lm(formula = mathach ~ female + minority + ses, data = hsb)

Residuals:
     Min       1Q   Median       3Q      Max
-19.6257  -4.5566   0.2428   4.8951  17.4746

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.2447     0.1264 112.738   <2e-16 ***
female       -1.3188     0.1600  -8.244   <2e-16 ***
minority     -2.8286     0.1852 -15.271   <2e-16 ***
ses           2.6775     0.1063  25.178   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.247 on 6141 degrees of freedom
  (1040 observations deleted due to missingness)
Multiple R-squared:  0.1705,    Adjusted R-squared:  0.1701
F-statistic: 420.9 on 3 and 6141 DF,  p-value: < 2.2e-16

> test_lm = lm(mathach ~ female + minority + ses + size, data = hsb)
> summary(test_lm)

Call:
lm(formula = mathach ~ female + minority + ses + size, data = hsb)

Residuals:
     Min       1Q   Median       3Q      Max
-19.6979  -4.5335   0.1934   4.8868  17.4024

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.3499100  0.1934485  74.179  < 2e-16 ***
female      -1.2897764  0.1636363  -7.882 3.82e-15 ***
minority    -2.7724618  0.1888099 -14.684  < 2e-16 ***
ses          2.6851962  0.1086022  24.725  < 2e-16 ***
size        -0.0001348  0.0001358  -0.992    0.321
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.228 on 5844 degrees of freedom
  (1336 observations deleted due to missingness)
Multiple R-squared:  0.1704,    Adjusted R-squared:  0.1698
F-statistic: 300.1 on 4 and 5844 DF,  p-value: < 2.2e-16

> # any row with a single NA is deleted
> # now use Amelia, similar to mice
> # Declare the variables to include in the imputation data
> varIndex <- names(hsb)
> # Declare ID variables to be excluded from imputation
> IDS <- c("schid", "meanses")
> # Imputate
> #
> impute.out <- amelia(hsb[, varIndex], idvars = IDS,
+                      noms = c("minority", "female"),
+                      m = 5)
-- Imputation 1 --

  1  2  3
```

```
-- Imputation 2 --

  1  2  3

-- Imputation 3 --

  1  2  3  4

-- Imputation 4 --

  1  2  3

-- Imputation 5 --

  1  2  3

> ##noms a vector of numbers or names indicating columns in the data that are nominal variables.
> ##ords a vector of numbers or names indicating columns in the data that should be treated as ordinal variables
> summary(impute.out)

Amelia output with 5 imputed datasets.
Return code:  1
Message:  Normal EM convergence.

Chain Lengths:
--------------
Imputation 1:  3
Imputation 2:  3
Imputation 3:  4
Imputation 4:  3
Imputation 5:  3

Rows after Listwise Deletion:  5849
Rows after Imputation:  7185
Patterns of missingness in the data:  14

Fraction Missing for original variables:
-----------------------------------------

         Fraction Missing
schid          0.00000000
minority       0.05358386
female         0.05024356
ses            0.04829506
mathach        0.00000000
size           0.04968685
schtype        0.00000000
meanses        0.00000000

> # use their mixed model
> fmla <- "mathach ~ minority + female + ses + meanses + (1 + ses|schid)"
```

```
> # run lmer on the missingness data
> mod <- lmer(fmla, data = hsb)
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
   Data: hsb

REML criterion at convergence: 39611.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2497 -0.7078  0.0385  0.7540  2.9604

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schid    (Intercept)  2.3284  1.5259
          ses          0.1787  0.4227   -1.00
 Residual             35.6998  5.9749
Number of obs: 6145, groups:  schid, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  14.0532     0.1801  78.009
minority     -2.7850     0.2181 -12.770
female       -1.1556     0.1714  -6.741
ses           1.9190     0.1215  15.790
meanses       3.0231     0.3633   8.321

Correlation of Fixed Effects:
         (Intr) minrty female ses
minority -0.334
female   -0.496  0.005
ses      -0.255  0.144  0.042
meanses  -0.118  0.136  0.025 -0.267
> # now using multiple imputation
> modList <- lmerModList(fmla, data = impute.out$imputations)
Warning message:
In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model failed to converge with max|grad| = 0.00206328 (tol = 0.002, component 1)

> summary(modList)
Linear mixed model fit by REML ['lmerMod']
Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
   Data: d

REML criterion at convergence: 46353.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2250 -0.7188  0.0348  0.7653  2.9399

Random effects:
```

```
 Groups   Name        Variance Std.Dev. Corr
 schid    (Intercept)  2.2738  1.5079
          ses          0.2964  0.5444   -0.59
 Residual             35.9046  5.9920
Number of obs: 7185, groups:  schid, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  13.9814     0.1720  81.286
minority     -2.4441     0.1983 -12.328
female       -1.2614     0.1581  -7.977
ses           1.9413     0.1160  16.739
meanses       3.1613     0.3548   8.910

Correlation of Fixed Effects:
         (Intr) minrty female ses
minority -0.319
female   -0.481  0.005
ses      -0.215  0.140  0.033
meanses  -0.099  0.127  0.026 -0.240
Linear mixed model fit by REML ['lmerMod']
Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
   Data: d

REML criterion at convergence: 46348.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2158 -0.7209  0.0313  0.7595  2.9309

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schid    (Intercept)  2.3064  1.5187
          ses          0.3834  0.6192   -0.53
 Residual             35.8367  5.9864
Number of obs: 7185, groups:  schid, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  13.9905     0.1724  81.169
minority     -2.5638     0.1965 -13.050
female       -1.2180     0.1584  -7.688
ses           1.9141     0.1189  16.099
meanses       3.2129     0.3572   8.995

Correlation of Fixed Effects:
         (Intr) minrty female ses
minority -0.315
female   -0.477  0.004
ses      -0.215  0.131  0.042
meanses  -0.096  0.122  0.023 -0.239
Linear mixed model fit by REML ['lmerMod']
```

```
Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
   Data: d

REML criterion at convergence: 46330.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2379 -0.7168  0.0350  0.7609  2.9242

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schid    (Intercept)  2.2741  1.508
          ses          0.2632  0.513    -0.69
 Residual             35.8081  5.984
Number of obs: 7185, groups:  schid, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  14.0238     0.1717  81.692
minority     -2.6521     0.1958 -13.543
female       -1.2024     0.1580  -7.608
ses           1.9390     0.1148  16.888
meanses       3.0852     0.3533   8.734

Correlation of Fixed Effects:
         (Intr) minrty female ses
minority -0.319
female   -0.480  0.007
ses      -0.228  0.128  0.039
meanses  -0.103  0.122  0.023 -0.248
convergence code: 0
Model failed to converge with max|grad| = 0.00206328 (tol = 0.002, component 1)

Linear mixed model fit by REML ['lmerMod']
Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
   Data: d

REML criterion at convergence: 46318.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2380 -0.7143  0.0363  0.7578  2.9547

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schid    (Intercept)  2.2934  1.5144
          ses          0.3153  0.5616   -0.63
 Residual             35.7195  5.9766
Number of obs: 7185, groups:  schid, 160

Fixed effects:
            Estimate Std. Error t value
```

```
(Intercept)  14.0013     0.1719  81.469
minority     -2.6519     0.1965 -13.494
female       -1.1782     0.1578  -7.467
ses           1.9740     0.1169  16.893
meanses       3.0962     0.3548   8.727

Correlation of Fixed Effects:
         (Intr) minrty female ses
minority -0.315
female   -0.477  0.001
ses      -0.227  0.134  0.038
meanses  -0.100  0.124  0.020 -0.243
Linear mixed model fit by REML ['lmerMod']
Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
   Data: d

REML criterion at convergence: 46338.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2504 -0.7178  0.0354  0.7536  2.9358

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 schid    (Intercept)  2.3133  1.5210
          ses          0.2901  0.5386   -0.67
 Residual             35.8303  5.9858
Number of obs: 7185, groups:  schid, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  13.9598     0.1725  80.940
minority     -2.5941     0.1972 -13.153
female       -1.1254     0.1580  -7.124
ses           1.9475     0.1157  16.829
meanses       3.1386     0.3545   8.854

Correlation of Fixed Effects:
         (Intr) minrty female ses
minority -0.315
female   -0.478  0.000
ses      -0.237  0.137  0.040
meanses  -0.101  0.120  0.022 -0.241
> fixef(mod) # model with dropped missing
(Intercept)    minority      female         ses     meanses
  14.053228   -2.784994   -1.155596    1.918987    3.023132
>
> fixef(modList)
(Intercept)    minority      female         ses     meanses
  13.991342   -2.581206   -1.197090    1.943176    3.138835
> # additional features in vignette
>
```

Analyzing Imputed Data with Multilevel Models and merTools

Jared Knowles

2018-06-04

Introduction

Multilevel models are valuable in a wide array of problem areas that involve non-experimental, or observational data. In many of these cases the data on individual observations may be incomplete. In these situations, the analyst may turn to one of many methods for filling in missing data depending on the specific problem at hand, disciplinary norms, and prior research.

One of the most common cases is to use multiple imputation. Multiple imputation involves fitting a model to the data and estimating the missing values for observations. For details on multiple imputation, and a discussion of some of the main implementations in R, look at the documentation and vignettes for the `mice` and `Amelia` packages.

The key difficulty multiple imputation creates for users of multilevel models is that the result of multiple imputation is K replicated datasets corresponding to different estimated values for the missing data in the original dataset.

For the purposes of this vignette, I will describe how to use one flavor of multiple imputation and the function in `merTools` to obtain estimates from a multilevel model in the presence of missing and multiply imputed data.

Missing Data and its Discontents

To demonstrate this workflow, we will use the `hsb` dataset in the `merTools` package which includes data on the math achievement of a wide sample of students nested within schools. The data has no missingness, so first we will simulate some missing data.

```
data(hsb)

# Create a function to randomly assign NA values

add_NA <- function(x, prob){
  z <- rbinom(length(x), 1, prob = prob)
  x[z==1] <- NA
  return(x)
}

hsb$minority <- add_NA(hsb$minority, prob = 0.05)
table(is.na(hsb$minority))
#>
```

```
#> FALSE  TRUE
#>  6836   349

hsb$female <- add_NA(hsb$female, prob = 0.05)
table(is.na(hsb$female))
#>
#> FALSE  TRUE
#>  6830   355

hsb$ses <- add_NA(hsb$ses, prob = 0.05)
table(is.na(hsb$ses))
#>
#> FALSE  TRUE
#>  6821   364

hsb$size <- add_NA(hsb$size, prob = 0.05)
table(is.na(hsb$size))
#>
#> FALSE  TRUE
#>  6848   337


# Load imputation library
library(Amelia)
# Declare the variables to include in the imputation data
varIndex <- names(hsb)
# Declare ID variables to be excluded from imputation
IDS <- c("schid", "meanses")
# Imputate
impute.out <- amelia(hsb[, varIndex], idvars = IDS,
                         noms = c("minority", "female"),
                         m = 5)
#> -- Imputation 1 --
#>
#>   1  2  3
#>
#> -- Imputation 2 --
#>
#>   1  2  3
#>
#> -- Imputation 3 --
#>
#>   1  2  3
#>
#> -- Imputation 4 --
#>
#>   1  2  3
#>
#> -- Imputation 5 --
#>
```

```
#>   1  2  3
summary(impute.out)
#>
#> Amelia output with 5 imputed datasets.
#> Return code:  1
#> Message:  Normal EM convergence.
#>
#> Chain Lengths:
#> --------------
#> Imputation 1:  3
#> Imputation 2:  3
#> Imputation 3:  3
#> Imputation 4:  3
#> Imputation 5:  3
#>
#> Rows after Listwise Deletion:  5865
#> Rows after Imputation:  7185
#> Patterns of missingness in the data:  13
#>
#> Fraction Missing for original variables:
#> -----------------------------------------
#>
#>          Fraction Missing
#> schid          0.00000000
#> minority       0.04857342
#> female         0.04940849
#> ses            0.05066110
#> mathach        0.00000000
#> size           0.04690327
#> schtype        0.00000000
#> meanses        0.00000000
```

Fitting and Summarizing a Model List

Fitting a model is very similar

```
fmla <- "mathach ~ minority + female + ses + meanses + (1 + ses|schid)"
mod <- lmer(fmla, data = hsb)
modList <- lmerModList(fmla, data = impute.out$imputations)
```

The resulting object `modList` is a list of `merMod` objects the same length as the number of imputation datasets. This object is assigned the class of `merModList` and `merTools` provides some convenience functions for reporting the results of this object.

Using this, we can directly compare the model fit with missing data excluded to the aggregate from the imputed models:

```
fixef(mod) # model with dropped missing
```

```
#> (Intercept)    minority      female         ses     meanses
#>   14.042814   -2.701367   -1.182849    1.899925    2.992339
fixef(modList)
#> (Intercept)    minority      female         ses     meanses
#>   13.966791   -2.507946   -1.191014    1.919880    3.228822


VarCorr(mod) # model with dropped missing
#>  Groups   Name        Std.Dev. Corr
#>  schid    (Intercept) 1.53617
#>           ses         0.64942  -0.566
#>  Residual             6.00409
VarCorr(modList) # aggregate of imputed models
#> $stddev
#> $stddev$schid
#> (Intercept)         ses
#>    1.520475    0.659457
#>
#>
#> $correlation
#> $correlation$schid
#>             (Intercept)        ses
#> (Intercept)   1.0000000 -0.5147035
#> ses          -0.5147035  1.0000000
```

If you want to inspect the individual models, or you do not like taking the mean across the imputation replications, you can take the `merModList` apart easily:

```
lapply(modList, fixef)
#> $imp1
#> (Intercept)    minority      female         ses     meanses
#>   14.000272   -2.681289   -1.174556    1.889865    3.185328
#>
#> $imp2
#> (Intercept)    minority      female         ses     meanses
#>   13.976422   -2.474407   -1.216113    1.926781    3.204234
#>
#> $imp3
#> (Intercept)    minority      female         ses     meanses
#>   13.969942   -2.458477   -1.213627    1.940791    3.232341
#>
#> $imp4
#> (Intercept)    minority      female         ses     meanses
#>   13.938151   -2.391582   -1.184232    1.907684    3.301779
#>
#> $imp5
#> (Intercept)    minority      female         ses     meanses
#>   13.949169   -2.533975   -1.166541    1.934279    3.220429
```

And, you can always operate on any single element of the list:

```
fixef(modList[[1]])
#> (Intercept)    minority      female         ses     meanses
#>   14.000272   -2.681289   -1.174556    1.889865    3.185328
fixef(modList[[2]])
#> (Intercept)    minority      female         ses     meanses
#>   13.976422   -2.474407   -1.216113    1.926781    3.204234
```

Output of a Model List

```
print(modList)
#> [1] "Linear mixed model fit by REML"
#> Model family:
#> lmer(formula = mathach ~ minority + female + ses + meanses +
#>     (1 + ses | schid), data = d)
#>
#> Fixed Effects:
#>             estimate std.error statistic         df
#> (Intercept)   13.967     0.174    80.318 378818.246
#> female        -1.191     0.159    -7.479 416213.147
#> meanses        3.229     0.360     8.966 144227.982
#> minority      -2.508     0.210   -11.924   1342.313
#> ses            1.920     0.121    15.891 334071.727
#>
#> Random Effects:
#>
#> Error Term Standard Deviations by Level:
#>
#> schid
#> (Intercept)         ses
#>       1.520       0.659
#>
#>
#> Error Term Correlations:
#>
#> schid
#>             (Intercept) ses
#> (Intercept)   1.000      -0.515
#> ses          -0.515       1.000
#>
#>
#> Residual Error = 5.988
#>
#> ---Groups
#> number of obs: 7185, groups: schid, 160
#>
#> Model Fit Stats
#> AIC = 46374.6
```

```
#> Residual standard deviation = 5.988
```

```
summary(modList)
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
#>    Data: d
#>
#> REML criterion at convergence: 46343.5
#>
#> Scaled residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.2140 -0.7239  0.0350  0.7628  2.9151
#>
#> Random effects:
#>  Groups   Name        Variance Std.Dev. Corr
#>  schid    (Intercept)  2.3529  1.5339
#>           ses          0.4588  0.6773   -0.47
#>  Residual             35.7671  5.9806
#> Number of obs: 7185, groups:  schid, 160
#>
#> Fixed effects:
#>             Estimate Std. Error t value
#> (Intercept)  14.0003     0.1741  80.420
#> minority     -2.6813     0.2000 -13.408
#> female       -1.1746     0.1590  -7.389
#> ses           1.8899     0.1210  15.620
#> meanses       3.1853     0.3612   8.819
#>
#> Correlation of Fixed Effects:
#>          (Intr) minrty female ses
#> minority -0.318
#> female   -0.481  0.013
#> ses      -0.210  0.144  0.041
#> meanses  -0.094  0.124  0.024 -0.231
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
#>    Data: d
#>
#> REML criterion at convergence: 46364
#>
#> Scaled residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.2124 -0.7202  0.0367  0.7565  2.9167
#>
#> Random effects:
#>  Groups   Name        Variance Std.Dev. Corr
#>  schid    (Intercept)  2.3179  1.5225
#>           ses          0.3793  0.6159   -0.60
#>  Residual             35.9233  5.9936
```

```
#> Number of obs: 7185, groups:  schid, 160
#>
#> Fixed effects:
#>             Estimate Std. Error t value
#> (Intercept)  13.9764     0.1735  80.542
#> minority     -2.4744     0.1989 -12.443
#> female       -1.2161     0.1589  -7.652
#> ses           1.9268     0.1191  16.176
#> meanses       3.2042     0.3567   8.984
#>
#> Correlation of Fixed Effects:
#>          (Intr) minrty female ses
#> minority -0.319
#> female   -0.485  0.015
#> ses      -0.236  0.136  0.047
#> meanses  -0.101  0.123  0.021 -0.239
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
#>    Data: d
#>
#> REML criterion at convergence: 46346.2
#>
#> Scaled residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.2167 -0.7297  0.0396  0.7633  2.9230
#>
#> Random effects:
#>  Groups   Name        Variance Std.Dev. Corr
#>  schid    (Intercept)  2.2662  1.5054
#>           ses          0.4036  0.6353   -0.55
#>  Residual             35.8290  5.9857
#> Number of obs: 7185, groups:  schid, 160
#>
#> Fixed effects:
#>             Estimate Std. Error t value
#> (Intercept)  13.9699     0.1722  81.129
#> minority     -2.4585     0.1975 -12.445
#> female       -1.2136     0.1584  -7.661
#> ses           1.9408     0.1193  16.267
#> meanses       3.2323     0.3549   9.109
#>
#> Correlation of Fixed Effects:
#>          (Intr) minrty female ses
#> minority -0.318
#> female   -0.485  0.012
#> ses      -0.228  0.146  0.048
#> meanses  -0.096  0.116  0.021 -0.238
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
#>    Data: d
```

```
#>
#> REML criterion at convergence: 46379.7
#>
#> Scaled residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.2050 -0.7232  0.0327  0.7602  2.9187
#>
#> Random effects:
#>  Groups   Name        Variance Std.Dev. Corr
#>  schid    (Intercept)  2.2941  1.5146
#>           ses          0.4589  0.6774   -0.51
#>  Residual             35.9703  5.9975
#> Number of obs: 7185, groups:  schid, 160
#>
#> Fixed effects:
#>             Estimate Std. Error t value
#> (Intercept)  13.9382     0.1732  80.464
#> minority     -2.3916     0.1983 -12.059
#> female       -1.1842     0.1588  -7.459
#> ses           1.9077     0.1212  15.743
#> meanses       3.3018     0.3576   9.233
#>
#> Correlation of Fixed Effects:
#>          (Intr) minrty female ses
#> minority -0.321
#> female   -0.483  0.012
#> ses      -0.216  0.138  0.038
#> meanses  -0.098  0.122  0.025 -0.233
#> Linear mixed model fit by REML ['lmerMod']
#> Formula: mathach ~ minority + female + ses + meanses + (1 + ses | schid)
#>    Data: d
#>
#> REML criterion at convergence: 46349.7
#>
#> Scaled residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.1958 -0.7228  0.0332  0.7635  2.9254
#>
#> Random effects:
#>  Groups   Name        Variance Std.Dev. Corr
#>  schid    (Intercept)  2.3287  1.5260
#>           ses          0.4781  0.6914   -0.44
#>  Residual             35.7935  5.9828
#> Number of obs: 7185, groups:  schid, 160
#>
#> Fixed effects:
#>             Estimate Std. Error t value
#> (Intercept)  13.9492     0.1736  80.350
#> minority     -2.5340     0.1995 -12.703
#> female       -1.1665     0.1587  -7.348
```

```
#> ses           1.9343     0.1214  15.933
#> meanses       3.2204     0.3607   8.927
#>
#> Correlation of Fixed Effects:
#>          (Intr) minrty female ses
#> minority -0.319
#> female   -0.482  0.017
#> ses      -0.201  0.134  0.047
#> meanses  -0.091  0.126  0.023 -0.231


fastdisp(modList)
#> lmer(formula = mathach ~ minority + female + ses + meanses +
#>     (1 + ses | schid), data = d)
#>             estimate std.error
#> (Intercept)    13.97      0.17
#> female         -1.19      0.16
#> meanses         3.23      0.36
#> minority       -2.51      0.21
#> ses             1.92      0.12
#>
#> Error terms:
#>  Groups   Name        Std.Dev. Corr
#>  schid    (Intercept) 1.52
#>           ses         0.66     -0.47
#>  Residual             5.99
#> ---
#> number of obs: 7185, groups: schid, 160
#> AIC = 46374.6
```

The standard errors reported for the model list include a correction, Rubin's correction (see documentation), which adjusts for the within and between imputation set variance as well.

Specific Model Information Summaries

```
modelRandEffStats(modList)
#>                        term    group   estimate   std.error
#> 1 cor_(Intercept).ses.schid    schid -0.5147035 0.065851800
#> 2      sd_(Intercept).schid    schid  1.5204747 0.010929919
#> 3   sd_Observation.Residual Residual  5.9880388 0.007245653
#> 4              sd_ses.schid    schid  0.6594570 0.032203227
modelFixedEff(modList)
#>          term  estimate std.error  statistic         df
#> 1 (Intercept) 13.966791 0.1738931  80.318271 378818.246
#> 2      female -1.191014 0.1592574  -7.478547 416213.147
#> 3     meanses  3.228822 0.3601076   8.966271 144227.982
#> 4    minority -2.507946 0.2103196 -11.924455   1342.313
#> 5         ses  1.919880 0.1208161  15.890936 334071.727
```

```
VarCorr(modList)
#> $stddev
#> $stddev$schid
#> (Intercept)         ses
#>    1.520475    0.659457
#>
#>
#> $correlation
#> $correlation$schid
#>             (Intercept)        ses
#> (Intercept)   1.0000000 -0.5147035
#> ses          -0.5147035  1.0000000
```

Diagnostics of List Components

```
modelInfo(mod)
#>   n.obs n.lvls      AIC   sigma
#> 1  6161      1 39813.18 6.00409
```

Let's apply this to our model list.

```
lapply(modList, modelInfo)
#> $imp1
#>   n.obs n.lvls      AIC    sigma
#> 1  7185      1 46361.49 5.980563
#>
#> $imp2
#>   n.obs n.lvls      AIC    sigma
#> 1  7185      1 46381.96 5.993608
#>
#> $imp3
#>   n.obs n.lvls      AIC    sigma
#> 1  7185      1 46364.19 5.985732
#>
#> $imp4
#>   n.obs n.lvls      AIC    sigma
#> 1  7185      1 46397.73 5.997524
#>
#> $imp5
#>   n.obs n.lvls      AIC    sigma
#> 1  7185      1 46367.72 5.982767
```

Cautions and Notes

Often it is desirable to include aggregate values in the level two or level three part of the model such as level 1 SES and level 2 mean SES for the group. In cases where there is missingness in either the level 1

SES values, or in the level 2 mean SES values, caution and careful thought need to be given to how to proceed with the imputation routine.

2. *Special and Specialized Multi-level Topics*:

A. Nonlinear Models. `nlmer` in lme4 Bates presentation Orange trees, logistic, in Stat222

B. Extensions of lme4 modeling: `npmlreg` Nonparametric Maximum Likelihood (NPML) estimation; `hglm` Hierarchical Generalized Linear Models,
Package `RLRsim` Title Exact (Restricted) Likelihood Ratio Tests for Mixed and Additive Models An Introduction to merTools

C. Multilevel Survival Analysis.

Mixed effects (Frailty) survival models. Package `coxme` Terry Therneau July 2014: Cox proportional hazards models containing Gaussian random effects, also known as frailty models coxme manual. Maintainer Terry Therneau. Mixed Effects Cox Models Terry Therneau Mayo Clinic June 15, 2015. Additional materials. frailtyHL: A Package for Fitting Frailty Models with H-likelihood

Stat222 coxme handout plain text version

D. Mediation Analysis Package `multilevel` Package `mediation` package vignette

E. Propensity Score and Matching Methods. Package `matchMulti` Title Optimal Multilevel Matching using a Network Algorithm Package `multilevelPSA` Title Multilevel Propensity Score Analysis

F. Ecological Inference R-package `eiPack`: R x C Ecological Inference and Higher-Dimension Data Management. R News Oct 2007

G. Multiple Imputation. Package `mitml` Title Tools for Multiple Imputation in Multilevel Modeling

H. Power calculations and Design;

Package pamm Title Power Analysis for Random Effects in Mixed Models
Package simr Title Power Analysis for Generalised Linear Mixed Models by Simulation.
longpower power and sample size for linear models of longitudinal data
Package MultiRR Title Bias, Precision, and Power for Multi-Level Random Regression

Mixed models in R using the lme4 package
Part 8: Nonlinear mixed models

Douglas Bates

University of Wisconsin - Madison
and R Development Core Team
<Douglas.Bates@R-project.org>

University of Lausanne
July 3, 2009

A simple example - logistic model of growth curves

- The `Orange` data set are measurements of the growth of a sample of five orange trees in a location in California.
- The response is the circumference of the tree at a particular height from the ground (often converted to "diameter at breast height").
- The covariates are `age` (days) and `Tree` (balanced).
- A data plot indicates that the growth patterns are similar but the eventual heights vary.
- One possible growth model is the *logistic* growth model

$$f(t, A, t_0, s) = \frac{A}{1 + e^{-(t-t_0)/s}}$$

which can be seen to be related to the inverse logit link function.

Orange *Growth of orange trees*

Description

The Orange data frame has 35 rows and 3 columns of records of the growth of orange trees.

Usage

```
Orange
```

Format

This data frame contains the following columns:

Tree a factor indicating the tree on which the measurement is made.

age a numeric vector giving the age of the tree (days since 1968/12/31)

circumference a numeric vector of trunk circumferences (mm). This is probably "circumference at breast height", a standard measurement in forestry.

Source

Draper, N. R. and Smith, H. (1998), *Applied Regression Analysis (3rd ed)*, Wiley (exercise 24.N).

Pinheiro, J. C. and Bates, D. M. (2000) *Mixed-effects Models in S and S-PLUS*, Springer.

Examples

```
xyplot(circumference ~ age, Orange, groups = Tree, type = c("g", "b"),
       auto.key = list(space = "right", lines = TRUE), aspect = "xy",
       xlab = "Age (days since 1968/12/31)", ylab = "Circumference (mm)")
## Not run:
m1 <- nlmer(circumference ~ SSlogis(age, Asym, xmid, scal) ~ Asym|Tree,
            Orange, verbose = TRUE,
            start = c(Asym = 190, xmid = 730, scal = 350))
.Call("mer_optimize", m1, 1L, 1L, PACKAGE = "lme4")
print(m1)
ranef(m1)

## End(Not run)
```

Using nlmer

- The nonlinear mixed-effects model is fit with the `nlmer` function in the `lme4` package.
- The formula argument for `nlmer` is in three parts: the response, the nonlinear model function depending on covariates and a set of nonlinear model (nm) parameters, and the mixed-effects formula.
- There is no longer a concept of an intercept or a `1` term in the mixed-effects model. All terms in the mixed-effects formula incorporate names of nm parameters.
- The default term for the fixed-effects is a separate "intercept" parameter for each nm parameter.
- At present, the nonlinear model must provide derivatives, in addition to the expected response. The `deriv` function can be used to create such a function from an expression.
- The starting values for the fixed effects must also be given. It is safest to phrase these as a named vector.

SSlogis {stats} R Documentation

Self-Starting Nls Logistic Model

Description

This `selfStart` model evaluates the logistic function and its gradient. It has an `initial` attribute that creates initial estimates of the parameters `Asym`, `xmid`, and `scal`.

Usage

```
SSlogis(input, Asym, xmid, scal)
```

Arguments

`input`	a numeric vector of values at which to evaluate the model.
`Asym`	a numeric parameter representing the asymptote.
`xmid`	a numeric parameter representing the `x` value at the inflection point of the curve. The value of `SSlogis` will be `Asym/2` at `xmid`.
`scal`	a numeric scale parameter on the `input` axis.

Value

a numeric vector of the same length as `input`. It is the value of the expression `Asym/(1+exp((xmid-input)/scal))`. If all of the arguments `Asym`, `xmid`, and `scal` are names of objects the gradient matrix with respect to these names is attached as an attribute named `gradient`.

Author(s)

José Pinheiro and Douglas Bates

See Also

`nls`, `selfStart`

Examples

```
Chick.1 <- ChickWeight[ChickWeight$Chick == 1, ]
SSlogis(Chick.1$Time, 368, 14, 6)  # response only
Asym <- 368; xmid <- 14; scal <- 6
SSlogis(Chick.1$Time, Asym, xmid, scal) # response and gradient
getInitial(weight ~ SSlogis(Time, Asym, xmid, scal), data = Chick.1)
## Initial values are in fact the converged values
fm1 <- nls(weight ~ SSlogis(Time, Asym, xmid, scal), data = Chick.1)
summary(fm1)
```

8.9 Balanced longitudinal data - Random coefficients and cubic smoothing splines

This section illustrates the use of random coefficients and cubic smoothing splines for the analysis of balanced longitudinal data.

The implementation of cubic smoothing splines in asreml() is based on the mixed model formulation of Verbyla et al. [1999]. More recently the methodology has been extended so that the user can specify knot points; in the original approach the knot points were taken to be the ordered set of unique values of the explanatory variable. The specification of knot points is particularly useful if the number of unique values in the explanatory variable is large, or if units are measured at different times.

These data were originally reported by Draper and Smith [1998, ex24N, p559] and have recently been reanalysed by Pinheiro and Bates [2000, p338]. The data are trunk circumferences (in millimetres) of each of 5 trees taken at 7 times (Figure 8.12). All trees were measured at the same time so that the data are balanced. The aim of the study is unclear, though both previous analyses involved modelling the overall *growth* curve, accounting for the obvious variation in both level and shape between trees.

Pinheiro and Bates [2000] used a nonlinear mixed effects modelling approach, in which they modelled the growth curves by a three parameter logistic function of age:

$$y = \frac{\phi_1}{1 + \exp\left[-(x - \phi_2)/\phi_3\right]} \tag{8.13}$$

where y is the trunk circumference, x is the tree age in days since December 31 1968, ϕ_1 is the asymptotic height, ϕ_2 is the inflection point or the time at which the tree reaches $0.5\phi_1$, ϕ_3 is the time elapsed between trees reaching half and about 3/4 of ϕ_1.

The data frame `orange` contains:

```
> orange <- asreml.read.table("orange.csv", header=T, sep=",")
> names(orange)
[1] "Tree" "x" "circ" "Season"
```

where `Tree` is a factor with 5 levels, `x` is tree age in days since 31 December 1968, `circ` is the trunk circumference and `Season` is a factor with two levels, `Spring` and `Autumn`. The factor `Season` was included after noting that tree age spans several years and if converted to day of year, measurements were taken in either April/May (`Spring`) or September/October (`Autumn`).

Initially we restrict the dataset to tree 1 to demonstrate fitting cubic splines in asreml(). The model includes the intercept and linear regression of trunk circumference on x and an additional random term `spl(x)` which includes a random term with a special design matrix with $7 - 2 = 5$ columns which relate to the vector, $\boldsymbol{\delta}$ whose elements $\delta_i, i = 2, \ldots, 6$ are the second differentials of the cubic spline at the knot points. The second differentials of a natural cubic spline are zero at the first and last knot points [Green and Silverman, 1994].

```
> orange.asr <- asreml(circ ~ x, random = ~ spl(x),
+ splinepoints = list(x = c(118,484,664,1004,1231,1372,1582)),
+ data = orange, subset = Tree==1)
```

In this example the spline knot points are specifically given in the `splinepoints` argument. These extra points have no effect in this case as they are the seven ages existing in the data file. In this instance the analysis would be the same if the `splinepoints` argument was omitted.

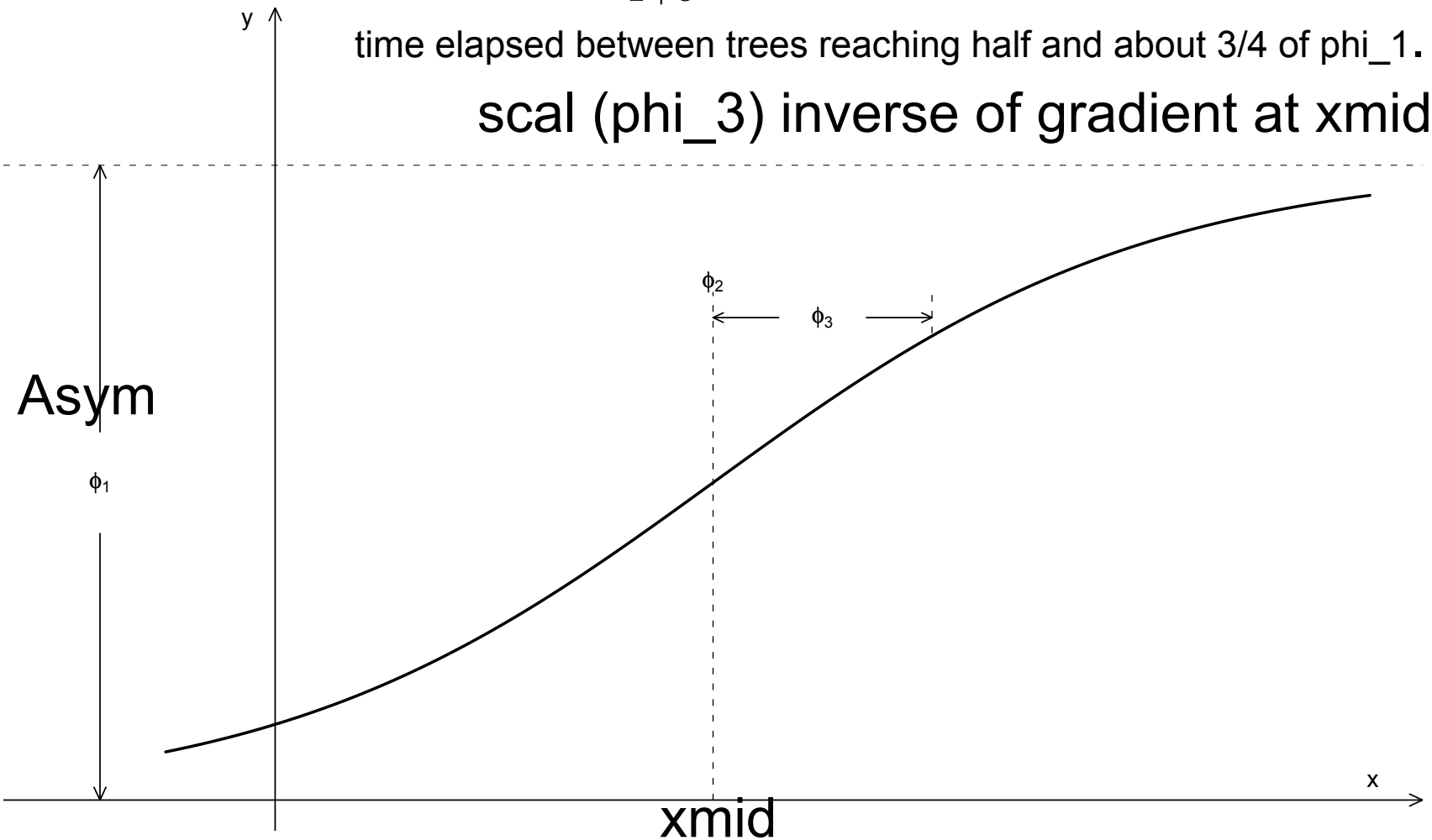
```
> summary(orange.asr)$varcomp

                gamma component std.error   z.ratio constraint
spl(x)     0.07876884  3.954159  9.950608 0.3973786   Positive
R!variance 1.00000000 50.199529 37.886791 1.3249876   Positive
```

The logistic growth model, SSlogis

$$y = \frac{\phi_1}{1+e^{-(x-\phi_2)/\phi_3}}$$

time elapsed between trees reaching half and about 3/4 of phi_1.

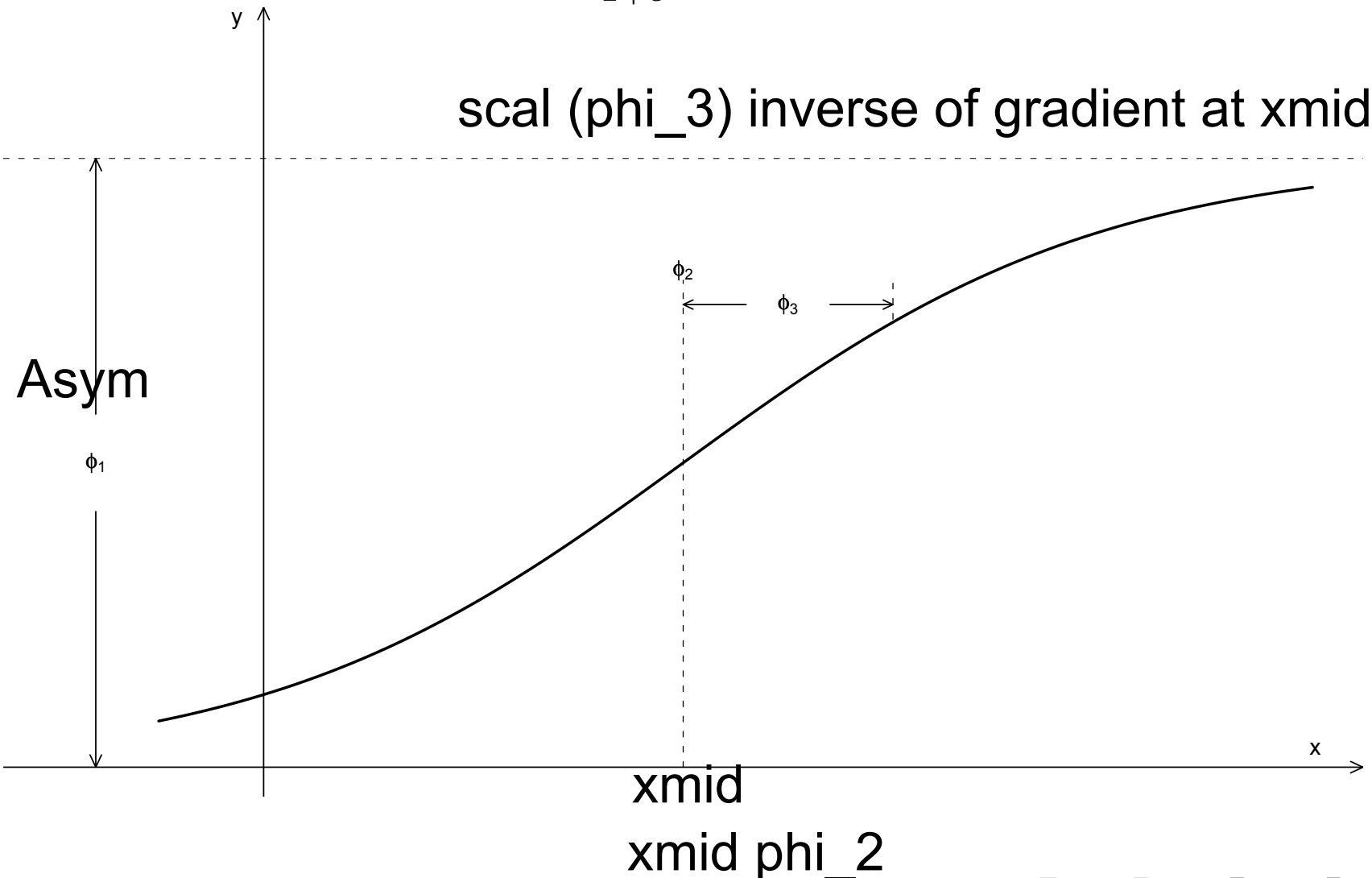
scal (phi_3) inverse of gradient at xmid



xmid phi_2

The logistic growth model, SSlogis

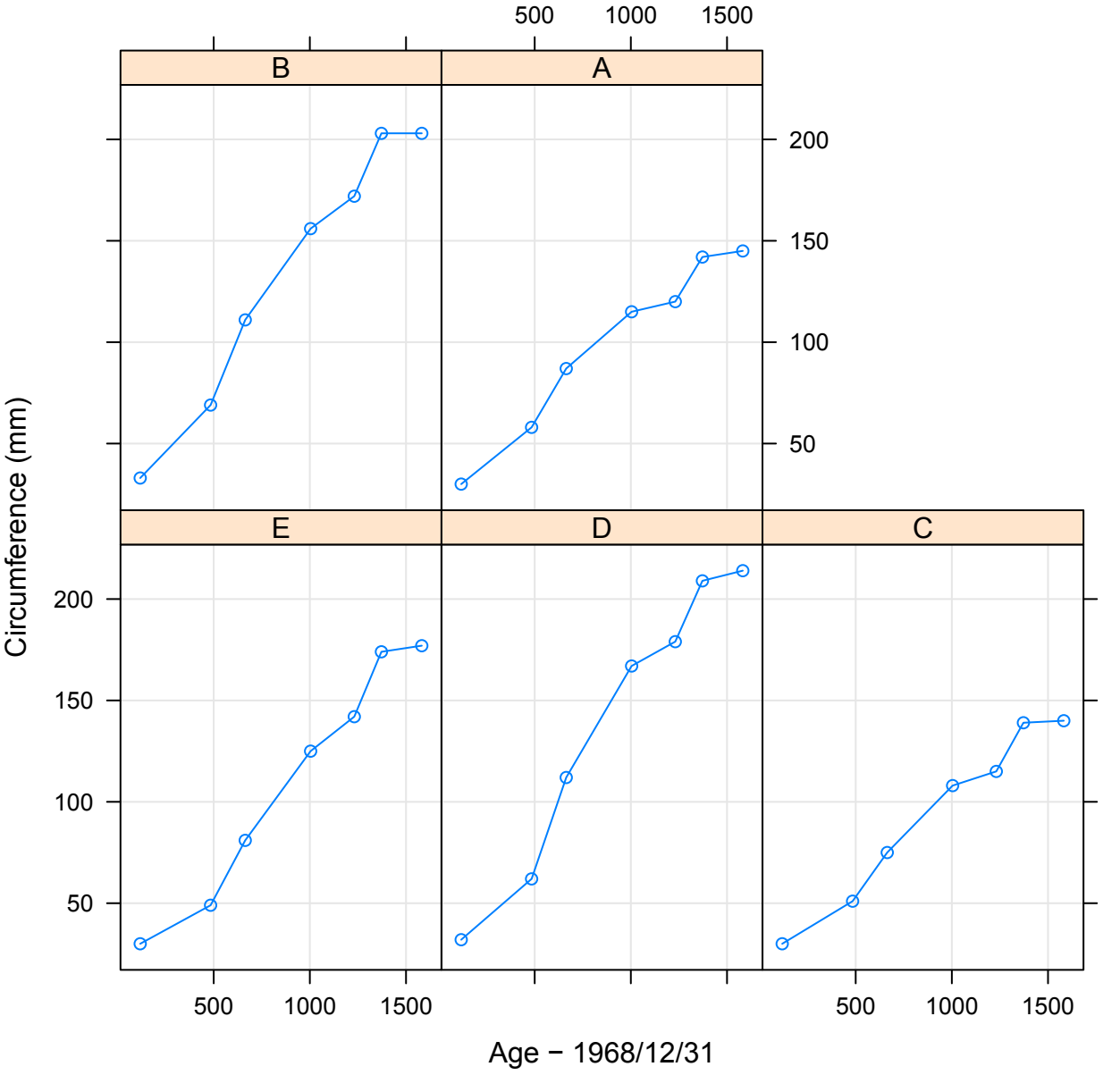
$$y = \frac{\phi_1}{1+e^{-(x-\phi_2)/\phi_3}}$$

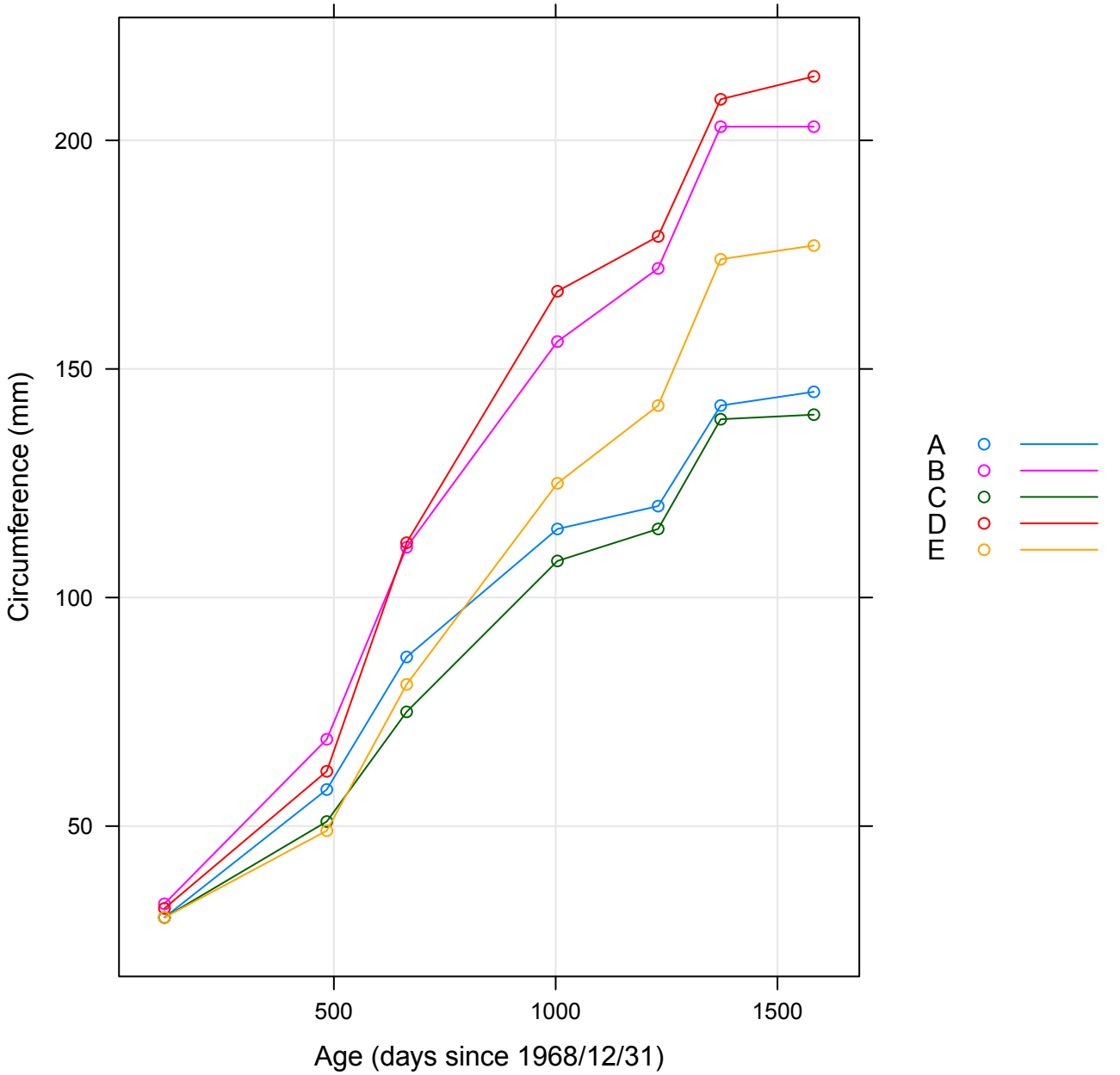


```
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

> library(MEMSS)
Loading required package: lme4
Loading required package: Matrix
Loading required package: lattice

Attaching package: 'Matrix'

The following object(s) are masked from 'package:base':

    det


Attaching package: 'lme4'

The following object(s) are masked from 'package:stats':

    AIC, BIC

Warning message:
package 'MEMSS' was built under R version 2.14.2
> head(Orange) #get data
  Tree  age circumference
1    A  118            30
2    A  484            58
3    A  664            87
4    A 1004           115
5    A 1231           120
6    A 1372           142


>   xyplot(circumference ~ age | Tree, Orange, type = c("g","b"),
+   index = function(x,y) coef(lm(y ~ x))[1],
+   xlab = "Age - 1968/12/31",
+   ylab = "Circumference (mm)", aspect = "xy")
> # connect-the-dots, frame-by-frame


> # connect-the-dots, all 5 trees
>  xyplot(circumference ~ age, Orange, groups = Tree, type = c("g", "b"),
+  auto.key = list(space = "right", lines = TRUE), aspect = "xy",
+  xlab = "Age (days since 1968/12/31)", ylab = "Circumference (mm)")

> ?SSlogis
starting httpd help server ... done

> # random asymptote model, provide starting values
>  m1 <- nlmer(circumference ~ SSlogis(age, Asym, xmid, scal) ~ Asym|Tree,
+  Orange, start = c(Asym = 190, xmid = 730, scal = 350))
>  print(m1)
Nonlinear mixed model fit by the Laplace approximation
Formula: circumference ~ SSlogis(age, Asym, xmid, scal) ~ Asym | Tree
   Data: Orange
  AIC  BIC logLik deviance
 1901 1908 -945.3     1891
Random effects:
 Groups   Name Variance  Std.Dev.
 Tree     Asym 53985.687 232.348
 Residual         52.868   7.271
Number of obs: 35, groups: Tree, 5

Fixed effects:
     Estimate Std. Error t value
Asym   192.04     104.09   1.845
xmid   727.89      31.97  22.771
scal   347.97      24.42  14.252

Correlation of Fixed Effects:
     Asym  xmid
xmid 0.053
```

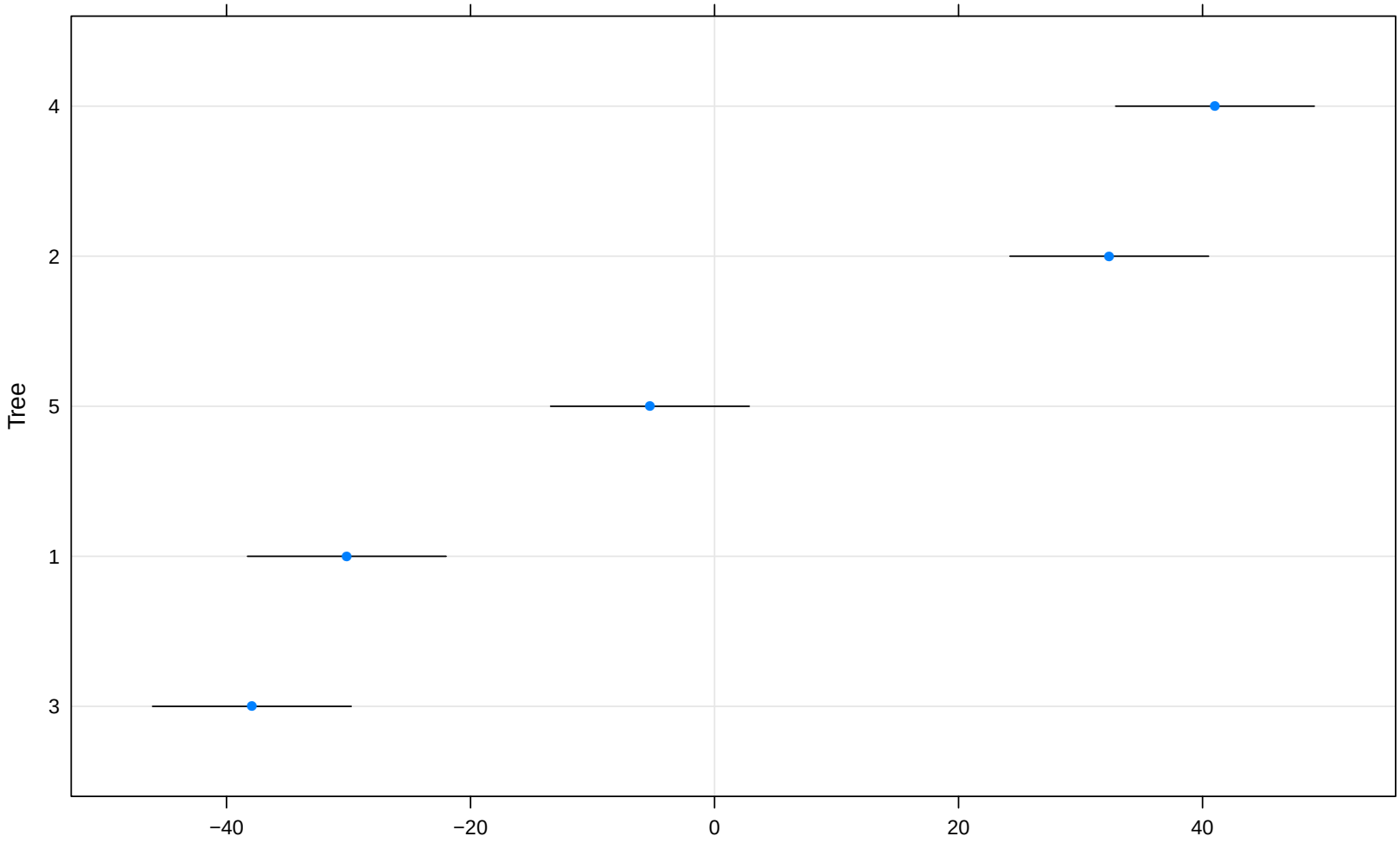
Model fit for orange tree data

```
> print(nm1 <- nlmer(circumference ~ SSlogis(age,
+     Asym, xmid, scal) ~ Asym | Tree, Orange, start = c(Asym =
+     xmid = 770, scal = 120)), corr = FALSE)
Nonlinear mixed model fit by the Laplace approximation
Formula: circumference ~ SSlogis(age, Asym, xmid, scal) ~ Asym | Tree
   Data: Orange
  AIC  BIC logLik deviance
 1901 1908 -945.3     1891
Random effects:
 Groups   Name Variance  Std.Dev.
 Tree     Asym 53985.368 232.348
 Residual         52.868   7.271
Number of obs: 35, groups: Tree, 5
Fixed effects:
     Estimate Std. Error t value
Asym   192.04     104.09   1.845
xmid   727.89      31.97  22.771
scal   347.97      24.42  14.252
```

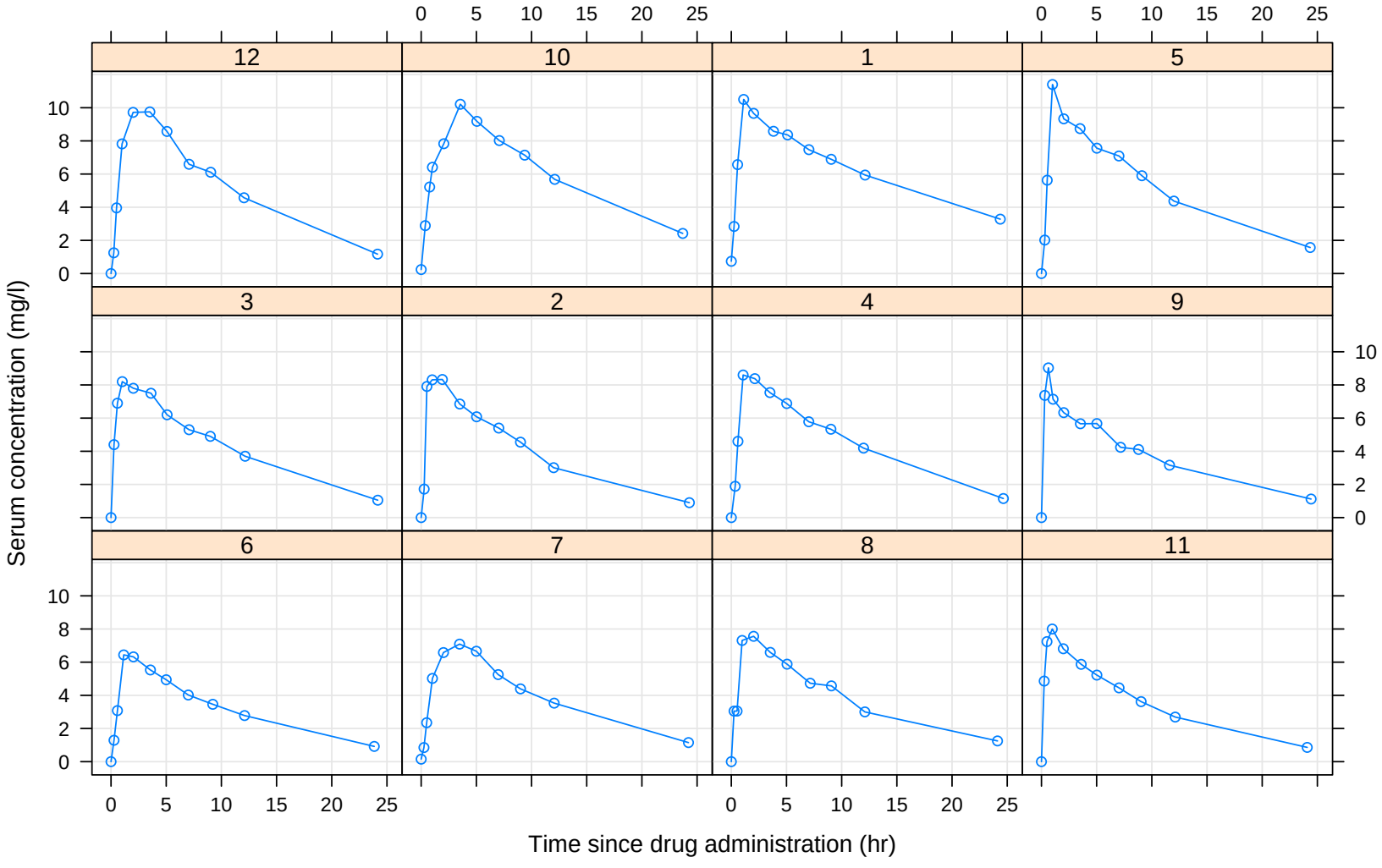

B
A
E
D
C
500
1000
1500
50
100
150
200
Circumference (mm)
Age − 1968/12/31


Circumference (mm)
200
150
100
50
500
1000
1500
Age (days since 1968/12/31)
A
B
C
D
E

Random effects for trees



Theophylline pharmacokinetics



SSfol {stats} R Documentation

Self-Starting Nls First-order Compartment Model

Description

This `selfStart` model evaluates the first-order compartment function and its gradient. It has an `initial` attribute that creates initial estimates of the parameters `lKe`, `lKa`, and `lCl`.

Usage

```
SSfol(Dose, input, lKe, lKa, lCl)
```

Arguments

`Dose`
: a numeric value representing the initial dose.

`input`
: a numeric vector at which to evaluate the model.

`lKe`
: a numeric parameter representing the natural logarithm of the elimination rate constant.

`lKa`
: a numeric parameter representing the natural logarithm of the absorption rate constant.

`lCl`
: a numeric parameter representing the natural logarithm of the clearance.

Value

a numeric vector of the same length as `input`, which is the value of the expression

```
Dose * exp(lKe+lKa-lCl) * (exp(-exp(lKe)*input) - exp(-exp(lKa)*input))
    / (exp(lKa) - exp(lKe))
```

If all of the arguments `lKe`, `lKa`, and `lCl` are names of objects, the gradient matrix with respect to these names is attached as an attribute named `gradient`.

Author(s)

José Pinheiro and Douglas Bates

See Also

`nls`, `selfStart`

Examples

```
Theoph.1 <- Theoph[ Theoph$Subject == 1, ]
SSfol(Theoph.1$Dose, Theoph.1$Time, -2.5, 0.5, -3) # response only
lKe <- -2.5; lKa <- 0.5; lCl <- -3
SSfol(Theoph.1$Dose, Theoph.1$Time, lKe, lKa, lCl) # response and gradient
getInitial(conc ~ SSfol(Dose, Time, lKe, lKa, lCl), data = Theoph.1)
## Initial values are in fact the converged values
fm1 <- nls(conc ~ SSfol(Dose, Time, lKe, lKa, lCl), data = Theoph.1)
summary(fm1)
```

Initial fit of first-order model

```
Nonlinear mixed model fit by the Laplace approximation
Formula: conc ~ SSfol(Dose, Time, lKe, lKa, lCl) ~ (lKe + lKa + lCl |
   Data: Theoph
   AIC   BIC logLik deviance
 152.1 181.0 -66.07    132.1
Random effects:
 Groups   Name Variance Std.Dev. Corr
 Subject  lKe  0.000000 0.00000
          lKa  0.227357 0.47682     NaN
          lCl  0.015722 0.12539     NaN -0.012
 Residual      0.591717 0.76923
Number of obs: 132, groups: Subject, 12
Fixed effects:
    Estimate Std. Error t value
lKe -2.47519    0.05641  -43.88
lKa  0.47414    0.15288    3.10
lCl -3.23550    0.05235  -61.80
```

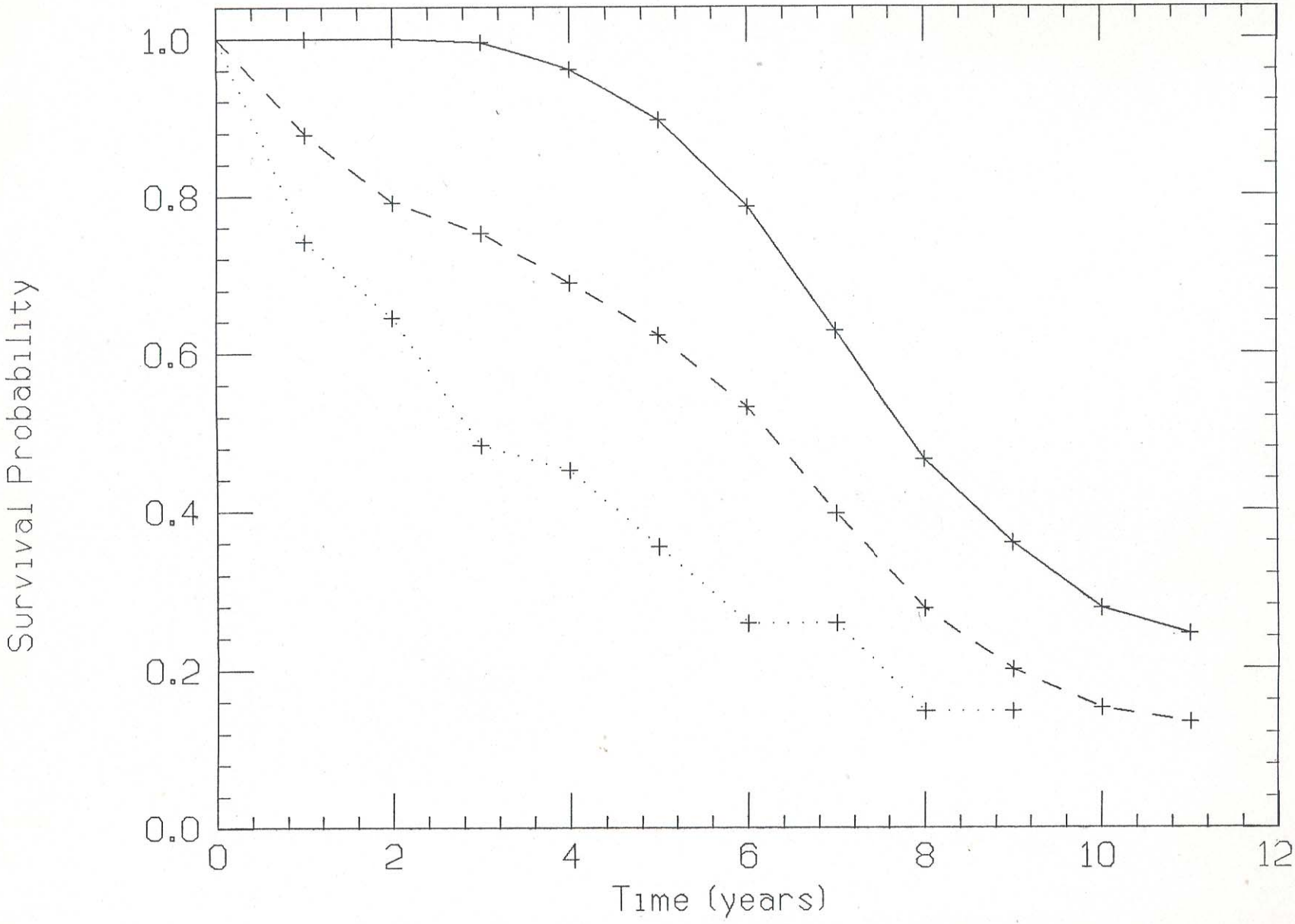


Figure 4. Survival curves of time-to-Ph.D. for all students (solid line), time-to-leaving-Stanford for non-minority students (dashed line) and time-to-leaving-Stanford for minority students (dotted line) matriculating in a Ph.D. program in Group D.

STAT 222

Cox Regression, Proportional Hazards

week 7

survival $S(t) = 1 - F(t)$

hazard $h(t) = f(t)/S(t)$; integrated hazard $H(t) = \int_0^t h(u)\,du$ (cum)

$$h(t) = \frac{-d\log(S(t))}{dt} \Rightarrow S(t) = \exp(-H(t))$$

hazard ↑ survival ↓

Modelling $h(t)$: baseline hazard

$$h_i(t) = h_0(t)\exp(\underset{\sim}{\beta}\underset{\sim}{x})$$

lm vs lmer

multiple regression?

$$\log(h_i(t)) = \alpha(t) + \beta_1 x_1 + \beta_2 x_2 + \dots$$

for unit increase in x_2 . . . .

for $x = 0$ $h_i(t) = h_0(t)$

coxph in survival

coxreg in eha

Heroin IML exs

cum hazard

$$H_i(t) = H_0(t)\exp(\underset{\sim}{\beta}\underset{\sim}{x})$$

$$\log(H_i(t)) = \log(H_0(t)) + \underset{\sim}{\beta}\underset{\sim}{x}$$

$$S_i(t) = S_0(t)^{\exp(\underset{\sim}{\beta}\underset{\sim}{x})}$$

---

exp(β) is multiplicative effect on hazard

[redacted]. The main assumption of the model is that the hazard of any individual i is a time-constant multiple of the hazard function of any other individual j, the factor being $\exp(\beta^T(\mathbf{x}_i - \mathbf{x}_j))$, the *hazard ratio* or incidence rate ratio. This property is called the *proportional hazards* assumption. The exponentiated regression parameters can therefore be interpreted as hazard ratios when the corresponding explanatory variable increases by one unit if all other covariates remain constant.

The parameters β are estimated by maximizing the *partial log like-*

Mixed Effects Cox Models

coxme vignette

Terry Therneau
Mayo Clinic

June 15, 2015

1 Introduction

Like many other projects, this software library started with a data set and a problem. From this came statistical ideas for a solution, followed by some initial programming — which more than anything else helped to define the *real* computational and statistical issues — and then a more ambitious programming solution. The problem turned out to be harder than I thought; the first release-worthy code took over 3 years in gestation and resulted in the *kinship* library. This in turn led to application to a larger set of problems, and a complete re-design of the underlying code. It was then split into separate libraries: coxme contains the central code for mixed effects Cox and linear models, kinship2 contains code for the construction and manipulation of pedigree data, and bdsmatrix contains underlying routines for block-diagonal sparse matrices.

2 Model

The coxme function fits the model

$$\lambda(t) = \lambda_0(t) e^{X\beta + Zb}$$
$$b \sim G(0, \Sigma(\theta))$$

where λ_0 is an unspecified baseline hazard function, X and Z are the design matrices for the fixed and random effects, respectively, β is the vector of fixed-effects coefficients and b is the vector of random effects coefficients. The random effects distribution G is modeled as Gaussian with mean zero and a variance matrix Σ, which in turn depends a vector of parameters θ.

dependent censoring. The MIICD package implements Pan's (2000) multiple imputation approach to the Fine and Gray model for interval censored data.

- ***Recurrent event data:*** `coxph` from the survival package can be used to analyse recurrent event data. The `cph` function of the rms package fits the Anderson-Gill model for recurrent events, model that can also be fitted with the frailtypack package. The latter also permits to fit joint frailty models for joint modelling of recurrent events and a terminal event. The survrec package proposes implementations of several models for recurrent events data, such as the Peña-Strawderman-Hollander, Wang-Chang estimators, and MLE estimation under a Gamma Frailty model. The condGEE package implements the conditional GEE for recurrent event gap times. The TestSurvRec package implements weighted logrank type tests for recurrent events.

Relative Survival

- The relsurv package proposes several functions to deal with relative survival data. For example, `rs.surv` computes a relative survival curve. `rs.add` fits an additive model and `rsmul` fits the Cox model of Andersen et al. for relative survival, while `rstrans` fits a Cox model in transformed time.
- The timereg package permits to fit relative survival models like the proportional excess and additive excess models.
- The JPSurv package implements methods for population-based survival analysis, like the proportional hazard relative survival model and the join point relative survival model.
- The survexp.fr package computes relative survival, absolute excess risk and standardized mortality ratio based on French death rates.
- The MRsurv package permits to fit multiplicative regression models for relative survival.
- The ROCt package implements time-dependent ROC curves and extensions to relative survival.

Random Effect Models

- ***Frailties:*** Frailty terms can be added in `coxph` and `survreg` functions in package survival. A mixed-effects Cox model is implemented in the coxme package. The `two.stage` function in the timereg package fits the Clayton-Oakes-Glidden model. The parfm package fits fully parametric frailty models via maximisation of the marginal likelihood. The frailtypack package fits proportional hazards models with a shared Gamma frailty to right-censored and/or left-truncated data using a penalised likelihood on the hazard function. The package also fits additive and nested frailty models that can be used for, e.g., meta-analysis and for hierarchically clustered data (with 2 levels of clustering), respectively. A proportional hazards model with mixed effects can be fitted using the phmm package. The lmec package fits a linear mixed-effects model for left-censored data. The Cox model using h-likelihood estimation for the frailty terms can be fitted using the frailtyHL package. The tlmec package implements a linear mixed effects model for censored data with Student-t or normal distributions.
- ***Joint modelling of time-to-event and longitudinal data:*** The joineR package allows the analysis of repeated measurements and time-to-event data via joint random effects models.

Multivariate Survival

Multivariate survival refers to the analysis of unit, e.g., the survival of twins or a family. To analyse such data, we can estimate the joint distribution of the survival times

- ***Joint modelling:*** Both Icens and MLEcens can estimate bivariate survival data subject to interval censoring.
- The mets package implements various statistical models for multivariate event history data, e.g., multivariate cumulative incidence models, bivariate random effects probit models, Clayton-Oakes model.
- The MST package constructs trees for multivariate survival data using marginal and frailty models.

Bayesian Models

- The bayesSurv package proposes an implementation of a bivariate AFT model.
- The package BMA computes a Bayesian model averaging for Cox proportional hazards models.
- The `DPsurvint` function in DPpackage fits a Bayesian semi-parametric AFT model. `LDDPsurvival` in the same package fits a Linear Dependent Dirichlet Process Mixture of survival models.
- `NMixMCMC` in mixAK performs an MCMC estimation of normal mixtures for censored data.
- A MCMC for Gaussian linear regression with left-, right- or interval-censored data can be fitted using the `MCMCtobit` in MCMCpack.
- The BayHaz package estimates the hazard function from censored data in a Bayesian framework.
- The `weibullregpost` function in LearnBayes computes the log posterior density for a Weibull proportional-odds regression model.
- The MCMCglmm fits generalised linear mixed models using MCMC to right-, left- and interval censored data.
- The BaSTA package aims at drawing inference on age-specific mortality from capture-recapture/recovery data when some or all records have missing information on times of birth and death. Covariates can also be included in the model.
- The JMbayes package performs joint modelling of longitudinal and time-to-event data under a bayesian approach.

Weinke

Chapter 3

Univariate frailty models

The structure, properties, and applicability of survival models to practical problems depend on the nature of time-to-event data, ancillary information about influential factors and processes, and the aims of the studies. This chapter focuses on the analysis of univariate data, i.e. single-spell data on unrelated individuals.

Ordinary survival models deal with the simplest case of independent and identically distributed data. This is based on the assumption that the study population is homogeneous. But it is a basic observation of medical statistics that individuals differ greatly. So do the effects of a drug, or the influence of various explanatory variables. This heterogeneity is often referred to as variability and it is generally recognized as one of the most important sources of variability in medical and biological applications. The subject of this chapter is heterogeneity in survival analysis. This heterogeneity may be difficult to assess, but it is nevertheless of great importance. In recent decades, a large amount of papers on 'frailty models' have appeared. The key idea of these models is that individuals have different frailties, and that the most frail will die earlier than the lesser frail. Consequently, systematic selection of robust individuals takes place, which distorts what is observed. When mortality rates are estimated, one may be interested in how they change over time or age. Quite often, they rise at the beginning of the observation period, reach a maximum and then decline (unimodal intensity) or level off. This, for example, is typical for death rates of cancer patients, meaning that the longer the patient lives beyond a certain time, the better his or her chances of survival are. It is likely that unimodal intensities are often the result of selection and that they do not reflect an underlying development on the individual level. The population intensity starts to decline simply because the high-risk individuals have already died. The intensity of a given individual might well continue to increase.

If covariates are known, they can be included in the analysis, for example by using the

The univariate frailty model

The simplest random effects model for clustered survival data is the univariate frailty model

$$\lambda_{ij}(t) = \lambda_0(t) \exp\{\boldsymbol{\beta}' \mathbf{Z}_{ij} + b_i\}.$$

Here i indicates the cluster, and j the individual from cluster i ($i = 1, ..., n,\ j = 1, ...n_i$), and b_i is the 'random cluster effect'. This is sometimes called a hierarchical model.

The assumption of the **random effects** model is that $b_1, ..., b_n$ are i.i.d. random variables, from a distribution known up to a finite number of parameters. For example, $b_i \sim N(0, \sigma^2)$. This can substantially reduce the number of parameters compared to fixed effects model.

Such modeling is most suitable when there are many clusters and each of relative small size.

In general b_i induces correlation among the individuals in the same cluster i.

Package 'coxme'

June 15, 2015

Title Mixed Effects Cox Models

Priority optional

Version 2.2-5

Depends survival (>= 2.36.14), methods, bdsmatrix(>= 1.3)

Imports nlme, Matrix (>= 1.0)

Suggests mvtnorm, kinship2

LinkingTo bdsmatrix

LazyData Yes

LazyLoad Yes

Description Cox proportional hazards models containing Gaussian random effects, also known as frailty models.

License LGPL-2

URL http://r-forge.r-project.org

NeedsCompilation yes

Author Terry M. Therneau [aut, cre]

Maintainer Terry M. Therneau <therneau@mayo.edu>

Repository CRAN

Date/Publication 2015-06-15 17:08:23

R topics documented:

anova.coxme 2
coxme 3
coxme.control 5
coxme.object 6
coxmeFull 8
coxmeMlist 9
eortc 10
expand.nested 11
fixed.effects 12

coxme *Fit a mixed effects Cox model*

Description

Fit a Cox model containing mixed (random and fixed) effects. Assume a Gaussian distribution for the random effects.

Usage

```
coxme(formula, data, weights, subset, na.action, init, control,
ties = c("efron", "breslow"),
varlist, vfixed, vinit,
sparse = c(50, 0.02),  x = FALSE, y = TRUE,
refine.n = 0, random, fixed, variance, ...)
```

Arguments

formula	a two-sided formula with a survival object as the left hand side of a ~ operator and the fixed and random effects on the right.
data	an optional data frame containing the variables named in the formula.
subset, weights, na.action	further model specifications arguments as in lm; see there for details.
init	optional initial values for the fixed effects.
control	optional list of control options. See coxme.control for details.

We can integrate out the random effects to create the integrated partial likelihood

$$IPL(\beta,\theta) = \frac{1}{(2\pi)^{q/2}|\Sigma(\theta)|^{1/2}} \int PPL(\beta,b) e^{-b'\Sigma^{-1}(\theta)b/2}\, db$$

where q is the length of b, i.e., the number of random effects. Ripatti and Palmgren [3] showed that the IPL can be treated as a likelihood, just as integrated full likelihoods can. An ML estimate is obtained by joint maximization over β and θ. We can compare a nested set of random or fixed effects Cox models using chi-square tests.

The REML estimate is obtained by also integrating out the fixed effects

$$REML(\theta) = \frac{1}{(2\pi)^{q/2}|\Sigma(\theta)|^{1/2}} \int PPL(\beta,b) e^{-b'\Sigma^{-1}(\theta)b/2}\, db\, d\beta$$

The REML estimate is obtained by maximization of this quantity over θ. As in linear models, the REML estimate cannot be used for testing fixed effects, due to the lack of a proper prior distribution for β. For instance, assume that `age` were a candidate variable. The REML value for a fit with age in days versus one with age in years will differ by the constant log(365.25); this has no impact on the estimate of θ, but makes the comparison of two models, one with and one without age, completely arbitrary.

In linear models the REML estimate is often preferred, since practical experience has shown it to be more reliable; the ML estimates are often too large. By analogy, some have recommended using a REML estimate for the mixed effects Cox model. However, either theoretical and practical evidence for its superiority is sparse. The coxme function currently only provides an ML estimate.

For a very simple fit such as

```
> fit1 <- coxme(Surv(endage, cancer) ~ parity + (1| famid))
```

found below there will be one random intercept per group (family), and Z will be the usual design matrix for a one-way anova, i.e., the same design matrix as would be used by a linear model `lm(y ~ factor(famid)-1)`. In this case $Z_{ij} = 1$ iff subject i is a member of family j. The variance of the random effects in this case is $\Sigma = \theta I$.

Note that if there are k groups the vector of random effects coefficients b will have k elements. Because of the shrinkage provided by the random effect they will satisfy $\sum b_k = 0$, rather than having one of their members set to zero. With respect to the random effects the `contrasts` option has no relevance.

3 Simple Models

For many random effects models the random effects can be specified very simply in the model formula. As an example consider the `eortc` data set, which is included with the package for illustration. This is a simple simulated example, based on the results of a breast cancer trial undertaken by the European Organization for Research and Treatment of Cancer. There are 37 enrolling centers with enrollments ranging from 21 to 247 subjects. We start by fitting a simple model with a random intercept per center.

- The total number of observations and the total number of events (deaths) in the data set.
- The computational effort, summarized as the number of iterations for the optim routine and the underlying Newton-Raphson iterations used.
- The log partial likelihood for a model with no covariates or random effects, the fitted partial likelihood, and the value with the random effects integrated out. We will normally be interested in the null and integrated values. (The log values are printed, but labeled as PL and IPL for brevity).
- Likelihood ratio tests based on the integrated and penalized views of the model, along with penalized values. The AIC penalizes by twice the effective degrees of freedom, and the BIC by $\log(d)$ times the effective degrees of freedom, where d is the number of events.
- A summary of the fixed effects
- A summary of the variances of the random effects

After integrating out the random effects, the log partial likelihood for the mixed effects model is 236.1. As a test of the random effects, we would normally compare this to `efit1`, the fit with no random effects which has a log partial likelihood of 105.7 The estimated standard deviation between centers of .33 is fairly substantial (more on this below). The difference is > 100 on one degree of freedom, which is highly significant.

Another way to approach the fit is as a penalized model. The coefficients are viewed as the solution to a penalized problem $\log(PPL(\beta, b)) - \log(PL(\beta, b)) - p(\theta)$ where the penalty function $p(\theta) = b'\Sigma^{-1}(\theta)b/2$. In this case we use the ordinary PL for assessment, but with an effective degrees of freedom which lies somewhere between the total number of coefficients (β, b) of 38 and the degrees of freedom for the fixed effects of 1. For this particular problem $\Sigma(\theta) = \theta I$, a multiple of the identity matrix and our penalty is $p(\theta) = -1/(2\theta)\sum j = 137b_j^2$. As the penalty goes to zero the fit will be equivalent to treating factor(center) as a fixed effect with degrees of freedom equal to 38, as the penalty becomes larger the effective degrees of freedom will decrease. In this case a test for the random effects would be based on (319.7 - 105.7) on 27.7 degrees of freedom. We do not normally use this test, but the adjusted degrees of freedom is useful in summarizing the fit.

The major components of the model can be extracted via the following functions:

	Method		
	lme/lmer	coxme < 2.2	coxme ≥ 2.2
fixed effects coefficients	fixef	fixef	fixef
random effects coefficients	ranef		ranef
fixed effects variance matrix	vcov		vcov
random effects parameters (θ)	VarCorr	ranef	VarCorr

Note that for version 2.2 and greater these follow the same pattern as the linear mixed effects models in R; prior to version 2.2 the ranef method was incorrectly assigned.

One feature of the mixed effects Cox model is that the standard deviation of the random effect is directly interpretable. The random effects b_j for each center j are in the risk score, a value of .33 for instance (one standard deviation above the mean) corresponds to a relative risk of $\exp(.33) = 1.39$, an almost 40% higher risk of death for subjects at that center. (In

`eortc` *Simulated data set based on an EORTC trial*

Description

This is a simulated surival data set for investigating random center effects. To make it realistic, the number of centers and their sizes is based on an EORTC cancer trial.

Usage

```
data(eortc)
```

Format

A data frame with 2323 observations on the following 4 variables.

`y` survival time

`uncens` 0=alive, 1=dead

`center` enrolling center, a number from 1 to 37

`trt` treatment arm, 0 or 1

Details

This is used in the test suite for the code.

Source

PhD thesis work of Jose Cortinas Abrahantes

References

Cortinas Abrahantes, Jose; Burzykowski, Tomasz (2002), A version of the EM algorithm for proportional hazards models with random effects , Published in: Lecture Notes of the ICB Seminars. p. 15-20

Examples

```
data(eortc)
coxme(Surv(y, uncens) ~ trt + (trt| center) + strata(center), eortc)
```

STAT 222 Week 9

Survival Analysis
mixed effects
(aka "frailty")

```
> install.packages("coxme")
> library(coxme)
Loading required package: survival splines bdsmatrix nlme Matrix lattice
> data(eortc)
results of a breast cancer trial undertaken by the European Organization for Research and Treatment of Cancer.
There are 37 enrolling centers with enrollments ranging from 21 to 247 subjects. We start by fitting a simple
model with a random intercept per center.
> dim(eortc)
[1] 2323    4
> head(eortc)
          y uncens center trt
2  506.1603      1      1   1
3  294.3800      1      1   1
4  383.9152      1      1   0
5 2441.8338      0      1   0
6 2442.2923      0      1   0
7  312.3571      1      1   1
> attach(eortc)
> table(uncens, center)
      center
uncens  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
     0 15 38 11 53 42 14 53  9 12 10 18 156 13 34 12 10 81  7 10  6  9 16  8 14 24
     1 48 48 23 51 74 21 67 14 44 20 24 91 26 18 22 16 102 27 11 29 21 19 33 39 67
      center
uncens 26 27 28 29 30 31 32 33 34 35 36 37
     0 10 14 13 45 16  7  6 20  7 36  5  6
     1 27 52 48 110 27 25 19 52 45 49 37 17
> table(uncens) #uncens 0=alive, 1=dead
uncens
   0    1
 860 1463
>  efit1 <- coxph(Surv(y, uncens) ~ trt, eortc)
Random effects are specified in the formula by a parenthesised expression which contains a
vertical bar separating effects on the left from grouping variables on the right. In the case above
we read it as an intercept (effect) per center (group).
>  efit2 <- coxme(Surv(y, uncens) ~ trt + (1|center), eortc)
> fixed.effects(efit2)
      trt
0.7086127
> print(efit2)
Cox mixed-effects model fit by maximum likelihood
  Data: eortc
  events, n = 1463, 2323
  Iterations= 9 49
                    NULL Integrated    Fitted
Log-likelihood -10638.71  -10520.65 -10478.84

                   Chisq    df p    AIC    BIC
Integrated loglik 236.11  2.00 0 232.11 221.53
 Penalized loglik 319.74 28.69 0 262.37 110.67

Model:  Surv(y, uncens) ~ trt + (1 | center)
Fixed coefficients
        coef exp(coef)   se(coef)     z p
trt 0.7086127  2.031171 0.06424398 11.03 0

Random effects
 Group  Variable  Std Dev   Variance
 center Intercept 0.3292140 0.1083818
> summary(efit1)
Call:
coxph(formula = Surv(y, uncens) ~ trt, data = eortc)
  n= 2323, number of events= 1463
       coef exp(coef) se(coef)     z Pr(>|z|)
trt 0.61826   1.85570  0.06343 9.747   <2e-16 ***
---
    exp(coef) exp(-coef) lower .95 upper .95
trt     1.856     0.5389     1.639     2.101

Concordance= 0.563  (se = 0.006 )
Rsquare= 0.044   (max possible= 1 )
Likelihood ratio test= 105.7  on 1 df,   p=0
Wald test            = 95.01  on 1 df,   p=0
Score (logrank) test = 98.04  on 1 df,   p=0
```

efit2

centers have diff efficacy

basic

coxme model

treatment effect

coxme

mult effect

coxme 2.03 vs coxph 1.86

big variability over centers

range on reverse

coxph

Cox regression

compare models (back)
236 vs 105

```
> install.packages("coxme")
> library(coxme)
Loading required package: survival splines bdsmatrix nlme Matrix lattice
> data(eortc)
results of a breast cancer trial undertaken by the European Organization for Research and Treatment of Cancer.
There are 37 enrolling centers with enrollments ranging from 21 to 247 subjects. We start by fitting a simple
model with a random intercept per center.
> dim(eortc)
[1] 2323    4
> head(eortc)
          y uncens center trt
2  506.1603      1      1   1
3  294.3800      1      1   1
4  383.9152      1      1   0
5 2441.8338      0      1   0
6 2442.2923      0      1   0
7  312.3571      1      1   1
> attach(eortc)
> table(uncens, center)
      center
uncens   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25
     0  15  38  11  53  42  14  53   9  12  10  18 156  13  34  12  10  81   7  10   6   9  16   8  14  24
     1  48  48  23  51  74  21  67  14  44  20  24  91  26  18  22  16 102  27  11  29  21  19  33  39  67
      center
uncens  26  27  28  29  30  31  32  33  34  35  36  37
     0  10  14  13  45  16   7   6  20   7  36   5   6
     1  27  52  48 110  27  25  19  52  45  49  37  17
> table(uncens) #uncens 0=alive, 1=dead
uncens
   0    1
 860 1463
>  efit1 <- coxph(Surv(y, uncens) ~ trt, eortc)
Random effects are specified in the formula by a parenthesised expression which contains a vertical
bar separating effects on the left from grouping variables on the right.  we read it as an
intercept (effect) per center (group).
>  efit2 <- coxme(Surv(y, uncens) ~ trt + (1|center), eortc)
> fixed.effects(efit2)
      trt
0.7086127
> print(efit2)
Cox mixed-effects model fit by maximum likelihood
  Data: eortc
  events, n = 1463, 2323
  Iterations= 9 49
                    NULL Integrated    Fitted
Log-likelihood -10638.71  -10520.65 -10478.84

                   Chisq    df p    AIC    BIC
Integrated loglik 236.11  2.00 0 232.11 221.53
 Penalized loglik 319.74 28.69 0 262.37 110.67

Model:  Surv(y, uncens) ~ trt + (1 | center)
Fixed coefficients
         coef exp(coef)   se(coef)     z p
trt 0.7086127  2.031171 0.06424398 11.03 0

Random effects
 Group  Variable  Std Dev   Variance
 center Intercept 0.3292140 0.1083818
> summary(efit1)
Call:
coxph(formula = Surv(y, uncens) ~ trt, data = eortc)
  n= 2323, number of events= 1463
       coef exp(coef) se(coef)     z Pr(>|z|)
trt 0.61826   1.85570  0.06343 9.747   <2e-16 ***
---
    exp(coef) exp(-coef) lower .95 upper .95
trt     1.856     0.5389     1.639     2.101

Concordance= 0.563  (se = 0.006 )
Rsquare= 0.044   (max possible= 1 )
Likelihood ratio test= 105.7  on 1 df,   p=0
Wald test            = 95.01  on 1 df,   p=0
Score (logrank) test = 98.04  on 1 df,   p=0
```

After integrating out the random effects, the log partial likelihood for the mixed effects model is 236.1. As a test of the random effects, we would normally compare this to efit1, the fit with no random effects which has a log partial likelihood of 105.7 The estimated standard deviation between centers of .33 is fairly substantial (more on this below). The difference is > 100 on one degree of freedom, which is highly significant

compare coxme coxph

The random effects bj for each center j are in the risk score,
a value of .33 for instance (one standard deviation above the mean) corresponds to a relative risk of exp(.33) = 1.39, an almost 40% higher risk of death for subjects at that center

```
> stem(exp(ranef(efit2)[[1]]))
  The decimal point is 1 digit(s) to the left of the |
   4 | 44
   6 | 42447
   8 | 679015556888
  10 | 12458966
  12 | 45670
  14 | 159
  16 | 27
> max(exp(ranef(efit2)[[1]]))    [1] 1.768246
```

variability over centers

To look at random treatment effects within center we can add a nested effect

```
> efit3 <- coxme(Surv(y, uncens) ~ trt + (1 | center/trt), eortc)
> efit3
Cox mixed-effects model fit by maximum likelihood
  Data: eortc
  events, n = 1463, 2323
  Iterations= 10 54
                    NULL Integrated    Fitted
Log-likelihood -10638.71  -10517.57 -10464.38

                   Chisq    df p    AIC    BIC
Integrated loglik 242.28  3.00 0 236.28 220.42
 Penalized loglik 348.67 39.26 0 270.16  62.56

Model:  Surv(y, uncens) ~ trt + (1 | center/trt)
Fixed coefficients
         coef exp(coef)   se(coef)    z p
trt 0.7420388  2.100213 0.08270483 8.97 0

Random effects
 Group      Variable    Std Dev    Variance
 center/trt (Intercept) 0.20451052 0.04182455
 center     (Intercept) 0.26273062 0.06902738
```

coxme extended "3-level" model

added term

This shows a further improvement in fit, but by much smaller amount.

```
> anova(efit2, efit3)
Analysis of Deviance Table
 Cox model: response is  Surv(y, uncens)
 Model 1: ~trt + (1 | center)
 Model 2: ~trt + (1 | center/trt)  # a '3-level' model
  loglik  Chisq Df P(>|Chi|)
1 -10521
2 -10518 6.1727  1    0.01297 *
---
```

compare coxme models

```
> #Also in vignette 4 The Minnesota Breast Cancer Family Study
Source: Mixed Effects Cox Models Terry Therneau Mayo Clinic May 15, 2012
```

Further examples of coxme (compared with frailtyHL) in
frailtyHL: A Package for Fitting Frailty Models with H-likelihood
by Il Do Ha, Maengseok Noh and Youngjo Lee
The R Journal Vol. 4/2, December 2012

shown in lecture

After integrating out the random effects, the log partial likelihood for the mixed effects model is 236.1. As a test of the random effects, we would normally compare this to efit1, the fit with no random effects which has a log partial likelihood of 105.7 The estimated standard deviation between centers of .33 is fairly substantial (more on this below). The difference is > 100 on one degree of freedom, which is highly significant

---

The random effects bj for each center j are in the risk score,
a value of .33 for instance (one standard deviation above the mean) corresponds to a relative risk of exp(.33) = 1.39, an almost 40% higher risk of death for subjects at that center

```
> stem(exp(ranef(efit2)[[1]]))
  The decimal point is 1 digit(s) to the left of the |
   4 | 44
   6 | 42447
   8 | 679015556888
  10 | 12458966
  12 | 45670
  14 | 159
  16 | 27
> max(exp(ranef(efit2)[[1]]))    [1] 1.768246
```

To look at random treatment effects within center we can add a nested effect

```
> efit3 <- coxme(Surv(y, uncens) ~ trt + (1 | center/trt), eortc)
> efit3
Cox mixed-effects model fit by maximum likelihood
  Data: eortc
  events, n = 1463, 2323
  Iterations= 10 54
                    NULL Integrated    Fitted
Log-likelihood -10638.71  -10517.57 -10464.38

                   Chisq    df p    AIC    BIC
Integrated loglik 242.28  3.00 0 236.28 220.42
 Penalized loglik 348.67 39.26 0 270.16  62.56

Model:  Surv(y, uncens) ~ trt + (1 | center/trt)
Fixed coefficients
         coef exp(coef)   se(coef)    z p
trt 0.7420388  2.100213 0.08270483 8.97 0

Random effects
 Group      Variable    Std Dev    Variance
 center/trt (Intercept) 0.20451052 0.04182455
 center     (Intercept) 0.26273062 0.06902738
```

This shows a further improvement in fit, but by much smaller amount.

```
> anova(efit2, efit3)
Analysis of Deviance Table
 Cox model: response is  Surv(y, uncens)
 Model 1: ~trt + (1 | center)
 Model 2: ~trt + (1 | center/trt)  # a '3-level' model
  loglik  Chisq Df P(>|Chi|)
1 -10521
2 -10518 6.1727  1   0.01297 *
---


> #Also in vignette 4 The Minnesota Breast Cancer Family Study
```

Source: Mixed Effects Cox Models Terry Therneau Mayo Clinic May 15, 2012

`rats` *Rat treatment data from Mantel et al*

Description

Rat treatment data from Mantel et al. Three rats were chosen from each of 100 litters, one of which was treated with a drug, and then all followed for tumor incidence.

Usage

```
rats
```

Format

litter:	litter number from 1 to 100
rx:	treatment,(1=drug, 0=control)
time:	time to tumor or last follow-up
status:	event status, 1=tumor and 0=censored
sex:	male or female

Note

The subset of females (odd numbered litters) was used Lee et al.

Source

N. R. Bohidar and J. L. Ciminera. Mantel-Haenszel analyses of litter-matched time to response data, with modifications for recovery of interlitter information. Cancer Research, 37:3863-3868, 1977. 5

References

E. W. Lee, L. J. Wei, and D. Amato, Cox-type regression analysis for large number of small groups of correlated failure time observations, in "Survival Analysis, State of the Art", Kluwer, 1992.