BIOSTATISTICS VHM 882 COURSE
WINTER 2005, ATLANTIC VETERINARY COLLEGE, PEI
HENRIK STRYHN

Index of Lecture 7–L

— in addition to overheads 8–9 from 6–L
and 280–81, 291–305 from Davis (2002)

LIBRARY (N)LME FOR S-PLUS/R

- authored by Pinheiro & Bates and documented in their book,
- implemented as standard in both S-Plus and R,
- arguably the most comprehensive software for linear mixed models:
  - ∗ most flexible model-building available (plus ability to add models),
  - ∗ wide range of analytic features, incl. confidence intervals and residuals[1],
  - ∗ able to handle moderately sized data structures (but not programmed specifically for multilevel models),
  - ∗ new points for statistical inference:
    - recommendation against LR tests for fixed effects,
    - simulation to access LR tests for variance param's,
- still under development in the new `lme4` package:
  - ∗ programmed more efficiently (using so-called S4 classes), but should give same results as `(n)lme` library,
  - ∗ multilevel structure built into code (to make analysis of larger models feasible),
  - ∗ current version not available for Windows.

---

[1] The residuals are not identical to those of MLwiN; it remains unclear which are preferable.

NOTES ON THE S PROGRAMMING LANGUAGE

Objected-oriented programming environment:
all components! (data, methods, results, etc.) are *objects* to which pre-defined sets of *functions* apply.

Objects for linear mixed models:

- data of class `groupedData`,
  - data values in a usual spreadsheet layout,
  - model formula reflecting the data structure,
  - pre-defined (and "clever") methods for summarizing and displaying data,
- results of fitting a linear mixed model[2],
  — many *functions* work on such objects, e.g.

name	function
`print`	brief information about the fit
`summary`	more detailed information about the fit
`anova`	likelihood-ratio or Wald F-tests
`intervals`	confidence intervals on model parameters
`residuals`	residuals for different levels of grouping
`plot`	diagnostic plots
`predict`	predictions for different levels of grouping
`update`	update the current fit for model change

- methods/functions are themselves objects and contain a `help()` function.

---

[2] The class of the object is either `lme` or `gls` depending on the function used.

LINEAR MIXED MODEL SPECIFICATION

Analysis of a linear mixed model is requested by a *function* of the form,[3]

lme (fixed, data, random, correlation, weights, subset,
method, na.action, control, contrasts)

where

- fixed = model formula for fixed part of model,
- random = model formula for random part of model,
- correlation = object for correlation structure,
- weights = object for variance heterogeneity structure,
- method = "REML" or "ML",
- na.action = function determining handling of missing values,
- data = name of data object.

Notes:

- many entries have default values and can be omitted (e.g., corr=NULL, weights=NULL, method="REML"),
- the returned object should be given a name, using the assignment operator "<-", e.g.,

```
testres <- lme(y~x, random=~1|subject, data=test)
```

---

[3] Models without random effects are fit by the `gls` function with a similar syntax.

CORRELATION STRUCTURES

Structures based on position (within a series):

- classical structures: compound symmetry, ar ∼ autoregressive (first and higher order), arma ∼ autoregressive moving average (first and higher order), unstructured,
- constructed by a function of the type (for arma),

  corARMA(value, form, p, q, fixed),

  where

  - value = initial value(s) — recommended for complex structures (estimation can be sensitive to these),
  - form = specification of "subjects" and order,
  - p,q = autoregressive and moving average orders,
- arma parameters difficult to interpret, except that for $p = 1$ the ar-parameter equals usual the ρ.

Structures based on (non-equidistant or spatial) distances:

- defined in terms of semivariogram γ defined as $\gamma(s) = 1 - h(s)$, where $h(s)$ is the correlation between two observations distance s apart,
- autocorrelation ∼ exponential: $\gamma(s) = 1 - \exp(-s/\rho)$,[4]
- other structures: Gaussian, linear, etc.

---

[4] In this parametrization, the `ar(1)` autocorrelation for a series with observations d units apart equals $\exp(-d/\rho)$.

R CODE FOR DENTAL STUDY

```
library(nlme)
data(Orthodont)
summary(Orthodont)
formula(Orthodont)
print(Orthodont)

# model 2: unstructured, age (centered at 8 yrs) as linear effect
Orth.unstr2 <-
gls(distance~I(age-8)*Sex,corr=corSymm(form=~1|Subject),
    weights=varIdent(form=~1|age),data=Orthodont,method="ML")
summary(Orth.unstr2)

# model 4a: unstructured, homogeneous variances
Orth.unstr.hom <- update(Orth.unstr2,weight=NULL)
summary(Orth.unstr.hom)

# model 4b: ARMA(1,1)
Orth.arma <- update(Orth.unstr.hom,correlation=corARMA(c(0.1,0.1),
                    form=~1|Subject,p=1,q=1))
summary(Orth.arma)
corMatrix(Initialize(corARMA(c(0.9703622,-0.7162636),
                     form=~1|Subject,p=1,q=1),data=Orthodont))[1]

# model 5: autoregressive (1)
Orth.ar1 <- update(Orth.unstr.hom,correlation=corAR1(form=~1|Subject))
summary(Orth.ar1)
anova(Orth.unstr2,Orth.unstr.hom,Orth.arma,Orth.ar1)

# model 6: random effects model with random slopes for age
Orth.ranslp <- lme(distance~I(age-1)*Sex,random=~age|Subject,
                   data=Orthodont,method="ML")
summary(Orth.ranslp)
```

```
# model 7: random intercept model
Orth.ran <- update(Orth.ranslp,random=~1|Subject)
summary(Orth.ran)
anova(Orth.ranslp,Orth.ran)

# model 7: compound symmetry
Orth.compsym <- update(Orth.unstr.hom,
                       corr=corCompSymm(form=~1|Subject))
summary(Orth.compsym)
anova(Orth.unstr.hom,Orth.arma,Orth.compsym)
plot(Orth.compsym,resid(.,type="n")~age|Sex)

# model 9: compound symmetry + sex variance heterogeneity
Orth.compsym.sex <- update(Orth.compsym,weights=varIdent(form=~1|Sex))
summary(Orth.compsym.sex)

# model 10: random intercept + sex variance heterogeneity
Orth.ran.sex <- update(Orth.ran,weights=varIdent(form=~1|Sex))
summary(Orth.ran.sex)

# model 11: random slopes + sex variance heterogeneity
Orth.ranslp.sex <- update(Orth.ranslp,weights=varIdent(form=~1|Sex))
summary(Orth.ranslp.sex)

# model 12: random intercept + sex co/variance heterogeneity
boy <- as.numeric(Orthodont$Sex=="Male")
girl <- as.numeric(Orthodont$Sex=="Female")
Orth.ran.sex2 <- update(Orth.ran.sex,random=pdDiag(~boy+girl-1))
summary(Orth.ran.sex2)
anova(Orth.ranslp.sex,Orth.ranslp,Orth.ran.sex2,Orth.ran.sex,
      Orth.compsym.sex,Orth.compsym,test=F)
```

R Code for Growth of Rats Study

```
library(nlme)
data(BodyWeight)
formula(BodyWeight)

# random slopes model
BW.ran <- lme(weight~Time*Diet,data=BodyWeight,random=~1|Rat)
summary(BW.ran)

# random slopes model
BW.ranslp <- update(BW.ran,random=~Time|Rat)
summary(BW.ranslp)
anova(BW.ranslp,BW.ran)
intervals(BW.ranslp)
plot(BW.ranslp,resid(.,type="p")~fitted(.),abline=0)

# added heteroscedasticity of power type
BW.ranslp.pow <- update(BW.ranslp,weights=varPower())
summary(BW.ranslp.pow)
plot(BW.ranslp.pow, resid(.,type="p")~fitted(.), abline=0)
anova(BW.ranslp,BW.ranslp.pow)
# autocorrelation function (but unequal spacing)
ACF(BW.ranslp.pow)
plot(ACF(BW.ranslp.pow,maxLag=10),alpha=0.01)
# semivariogram more appropriate
Variogram(BW.ranslp.pow,form=~Time)
plot(Variogram(BW.ranslp.pow,form=~Time,maxDist=42))

# added correlation of exponential type
BW.ranslp.pow.exp <- update(BW.ranslp.pow,corr=corExp(form=~Time))
summary(BW.ranslp.pow.exp)
plot(Variogram(BW.ranslp.pow.exp,form=~Time,maxDist=42,
               resType="n",robust=T))
anova(BW.ranslp.pow,BW.ranslp.pow.exp)
```