Advanced mixed-models workshop: Session 1

Dale Barr

University of Glasgow

Bremen March 2015

Tentative schedule

Today

	Start	End	Activity
1	09:00	10:30	Review / Overview
2	11:00	12:30	Dataset 1 (one factor)
3	13:30	15:00	Dataset 2 (multifactor)
4	15:30	17:00	Slack time / Q&A / BYOD*

Tomorrow

	Start	End	Activity
1	09:00	10:30	Dataset 3 (GLMM)
2	11:00	12:30	Dataset 4 (multifactor GLMM)
3	13:00	15:00	Slack time / Q&A / BYOD*

* Bring Your Own Data

Repository for this workshop

If you have git installed, use:

```
git clone https://github.com/dalejbarr/bremen.git
```

or download full archive from:

https://github.com/dalejbarr/bremen/archive/master.zip

General information on LMEMs

- Baayen (2008), *Analyzing Linguistic Data*
- Baayen, Davidson, & Bates (2008), *JML*
- Barr, Levy, Scheepers, Tily (2013), *JML*
- Barr (2013), *Frontiers in Psychology* (interactions)
- Bates et al. `http://arxiv.org/pdf/1406.5823.pdf` (technical)
- Bolker et al. (2009), *Trends in Ecology & Evolution*
- Westfall, Kenny, and Judd (2014), *JEP: General* (power)
- see also `r-lang` and `r-sig-mixed-models` mailing lists
- `r-sig-mixed-models` FAQ `http://glmm.wikidot.com/faq`
- add-on packages `afex`, `pbkrcomp`, `lmerTest`

Simulated data

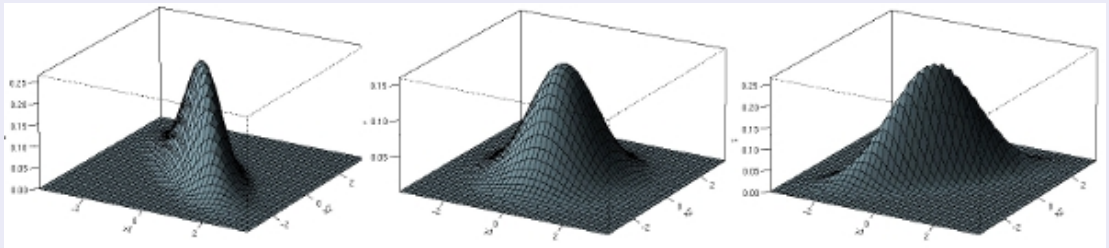
- single-factor within subject / between items
- IV: type of word, DV = lexical decision times

$$Y_{si} = \beta_0 + S_{0s} + I_{0i} + (\beta_1 + S_{1s})X_i + e_{si}$$

Subjects

$$< S_{0i}, S_{1i} > \sim N(< 0, 0 >, \Sigma)$$

$$\Sigma = \begin{pmatrix} {\tau_{00}}^2 & \rho_I \tau_{00} \tau_{11} \\ \rho_I \tau_{00} \tau_{11} & {\tau_{11}}^2 \end{pmatrix}$$



Items

$$I_{0i} \sim N(0, \omega_{00}^2)$$

Define the data structures

```
library("MASS") # needed for mvrnorm

set.seed(11709)

nsubj <- 100
nitem <- 50 # must be an even number

##########################################
## create the data structures
subj <- data.frame(subject_id = 1:nsubj)

item <- data.frame(item_id = 1:nitem,
                   cond = rep(1:2, each = nitem / 2))

trial <- expand.grid(subject_id = 1:nsubj,
                     item_id = 1:nitem)
```

Define parameters for data generation

```
##########################################
## define parameters for data generation
## first, fixed effects
mu <- 800 # grand mean
eff <- 80 # 80 ms difference
effc <- c(-.5, .5) # deviation codes
res_sd <- 200 # residual (standard deviation)
iri <- 80 # by-item random intercept sd
sri <- 100 # by-subject random intercept sd
srs <- 40 # by-subject random slope sd
rcor <- .2 # correlation between intercept and slope
```

By-item random effects

```
## define item random effects variance
## and sample items
item$iri <- rnorm(nitem, mean = 0, sd = iri)

## view the expected mean for each item
## for a typical subject (random effs = 0)
head(cbind(mu, eff[item$cond], item$iri,
           mu + eff[item$cond] + item$iri), 3)
```

```
      mu
[1,] 800 80  14.94782 894.9478
[2,] 800 80 -86.30801 793.6920
[3,] 800 80 -12.78345 867.2165
```

By-subject random effects

```r
## define subject random effects variance
## variance co-variance matrix
svcov <- matrix(c(sri^2,
                  rcor * sri * srs,
                  rcor * sri * srs,
                  srs^2), nrow = 2)

## sample subjects
srfx <- mvrnorm(nsubj, mu = c(0, 0), Sigma = svcov)

subj$sri <- srfx[, 1]
subj$srs <- srfx[, 2]

head(subj, 3)
```

```
  subject_id        sri         srs
1          1 -80.025967  -0.7625934
2          2  44.612596  54.5130100
3          3   8.744992 -20.4295562
```

Pull it all together

```
dat <- merge(merge(subj, trial), item)
dat <- dat[order(dat$subject_id, dat$item_id), ] # sort
dat$err <- rnorm(nrow(dat), sd = res_sd) # trial-level noise
dat$Y <- with(dat,
    mu + sri + iri + (eff + srs) * effc[cond] + err)

head(dat)
```

```
    item_id subject_id       sri        srs cond       iri        err        Y
1         1          1 -80.02597 -0.7625934    1  14.94782  382.34441 1077.6476
173       2          1 -80.02597 -0.7625934    1 -86.30801  283.44878  877.4961
235       3          1 -80.02597 -0.7625934    1 -12.78345   30.35586  697.9277
390       4          1 -80.02597 -0.7625934    1 -13.91040 -282.01806  384.4269
414       5          1 -80.02597 -0.7625934    1  55.61871 -238.73081  497.2432
513       6          1 -80.02597 -0.7625934    1 -45.92916   73.42391  707.8501
```

Decomposition matrix

$$Y_{si} = \beta_0 + S_{0s} + I_{0i} + (\beta_1 + S_{1s})X_i + e_{si}$$

```
Attaching package: 'dplyr'

The following object is masked from 'package:MASS':

    select

The following object is masked from 'package:stats':

    filter

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
 Source: local data frame [16 x 11]

   sid iid c          Y  mu        sri       iri eff         srs    x
1    1   1 1 1077.6476 800 -80.025967  14.94782  80  -0.7625934 -0.5
2    1   2 1  877.4961 800 -80.025967 -86.30801  80  -0.7625934 -0.5
3    1  26 2  637.6155 800 -80.025967 -65.44482  80  -0.7625934  0.5
4    1  27 2  808.4316 800 -80.025967 171.89799  80  -0.7625934  0.5
5    2   1 1  533.2496 800  44.612596  14.94782  80  54.5130100 -0.5
6    2   2 1  930.9572 800  44.612596 -86.30801  80  54.5130100 -0.5
```

Fitting the model

```
library("lme4")
dat$c <- dat$cond - mean(dat$cond)
mod <- lmer(Y ~ c + (1 + c | subject_id) + (1 | item_id),
            dat, REML = FALSE)
```

```
Loading required package: Matrix
Loading required package: Rcpp
```

Viewing results

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: Y ~ c + (1 + c | subject_id) + (1 | item_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 67635.0  67680.6 -33810.5  67621.0     4993

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8927 -0.6637 -0.0136  0.6749  3.7872

Random effects:
 Groups     Name        Variance Std.Dev. Corr
 subject_id (Intercept)  9467     97.30
            c            1254     35.42   0.32
 item_id    (Intercept)  7767     88.13
 Residual                40119    200.30
Number of obs: 5000, groups:  subject_id, 100; item_id, 50

Fixed effects:
            Estimate Std. Error t value
(Intercept)   793.29      16.06   49.38
c             112.10      25.81    4.34
```