Data simulation for linear mixed-effects models

Nora Wickelmaier

January 9, 2023

Example: Crossed random effects

- This example will show how to include subjects and items as crossed, independent, random effects, as opposed to hierarchical or multilevel models in which random effects are assumed to be nested
- The data are taken from Baayen et al. (2008)
- Assume an example data set with three participants s1, s2 and s3 who each saw three items w1, w2, w3 in a priming lexical decision task under both short and long SOA conditions
- Let's say the data were generated by the following model

$$y_{ij} = \beta_0 + \beta_1 SOA_j + \omega_{0j} + \upsilon_{0i} + \upsilon_{1i} SOA_j + \varepsilon_{ij}$$

with $\boldsymbol{\upsilon} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_\upsilon = \begin{pmatrix} \sigma^2_{\upsilon_0} & \sigma_{\upsilon_0\upsilon_1} \\ \sigma_{\upsilon_0\upsilon_1} & \sigma^2_{\upsilon_1} \end{pmatrix}\right)$, $\omega_{0j} \sim N(0, \sigma^2_\omega)$, $\varepsilon_{ij} \sim N(0, \sigma^2_\varepsilon)$, all i.i.d.

True values

- We assume the following true parameters for a data simulation

Parameter	Model
β_0	522.11
β_1	-18.89
σ_ω	21.10
σ_{υ_0}	23.89
σ_{υ_1}	9.00
$\rho_{\upsilon_0\upsilon_1}$	-1.00
σ_ε	9.90

$$y_{ij} = \beta_0 + \beta_1 SOA_j + \omega_{0j} + \upsilon_{0i} + \upsilon_{1i} SOA + \varepsilon_{ij}$$

with $\boldsymbol{\upsilon} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_\upsilon = \begin{pmatrix} \sigma^2_{\upsilon_0} & \sigma_{\upsilon_0\upsilon_1} \\ \sigma_{\upsilon_0\upsilon_1} & \sigma^2_{\upsilon_1} \end{pmatrix}\right)$, $\omega_{0j} \sim N(0, \sigma^2_\omega)$, $\varepsilon_{ij} \sim N(0, \sigma^2_\varepsilon)$

Example data set

With random intercepts for subject and item, and random slopes for subject

Subj	Item	SOA	RT	Fixed		Random			Res
				Int	SOA	ItemInt	SubInt	SubSOA	
s1	w1	Long	466	522.2	0	−28.3	−26.2	0	−2.0
s1	w2	Long	520	522.2	0	14.2	−26.2	0	9.8
s1	w3	Long	502	522.2	0	14.1	−26.2	0	−8.2
s1	w1	Short	475	522.2	−19	−28.3	−26.2	11	15.4
s1	w2	Short	494	522.2	−19	14.2	−26.2	11	−8.4
s1	w3	Short	490	522.2	−19	14.1	−26.2	11	−11.9
s2	w1	Long	516	522.2	0	−28.3	29.7	0	−7.4
s2	w2	Long	566	522.2	0	14.2	29.7	0	0.1
s2	w3	Long	577	522.2	0	14.1	29.7	0	11.5
s2	w1	Short	491	522.2	−19	−28.3	29.7	−12.5	−1.5
s2	w2	Short	544	522.2	−19	14.2	29.7	−12.5	8.9
s2	w3	Short	526	522.2	−19	14.1	29.7	−12.5	−8.2
s3	w1	Long	484	522.2	0	−28.3	−3.5	0	−6.3
s3	w2	Long	529	522.2	0	14.2	−3.5	0	−3.5
s3	w3	Long	539	522.2	0	14.1	−3.5	0	6.0
s3	w1	Short	470	522.2	−19	−28.3	−3.5	1.5	−2.9
s3	w2	Short	511	522.2	−19	14.2	−3.5	1.5	−4.6
s3	w3	Short	528	522.2	−19	14.1	−3.5	1.5	13.2
						$\sigma^2_{\omega_0}$	$\sigma^2_{\upsilon_0}$	$\sigma^2_{\upsilon_1}$	σ^2_{ε}
							$\sigma_{\upsilon_0\upsilon_1}$		

Fixed effects

```
datsim <- expand.grid(subject = factor(c("s1", "s2", "s3")),
                      item = factor(c("w1", "w2", "w3")),
                      soa = factor(c("long", "short")))
datsim <- datsim[order(datsim$subject), ]

# Model matrix in dummy coding
model.matrix(~ soa, datsim)

beta0  <- 522.11
beta1  <- -18.89
b0 <- rep(beta0, 18)
b1 <- rep(rep(c(0, beta1), each = 3), 3)
cbind(b0, b1)
```

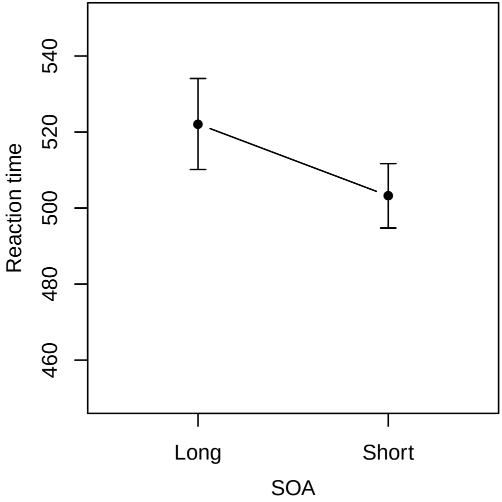
Random effects

```r
sw  <- 21.1
sy0 <- 23.89; sy1 <- 9; ry <- -1
se  <- 9.9

w  <- rep(rnorm(3, mean = 0, sd = sw), 6)
e  <- rnorm(18, mean = 0, sd = se)
# Draw from bivariate normal distribution
sig <- matrix(c(sy0^2, ry*sy0*sy1, ry*sy0*sy1, sy1^2), 2, 2)
y01 <- mvtnorm::rmvnorm(3, mean = c(0, 0), sigma = sig)
y0 <- rep(y01[,1], each = 6)
y1 <- rep(c(0, y01[1,2],
            0, y01[2,2],
            0, y01[3,2]), each = 3)
cbind(w, y0, y1, e)
```

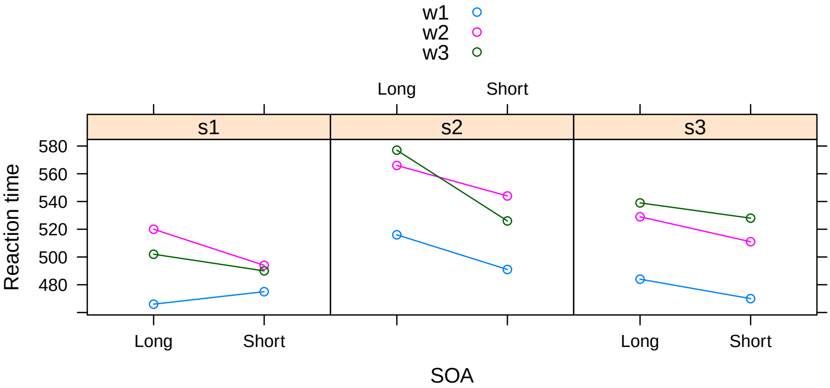
Simulate data

```r
datsim$rt <- b0 + b1 + w + y0 + y1 + e

# Fit model
library(lme4)

lme1 <- lmer(rt ~ soa + (1 | item) + (soa | subject), datsim, REML=F)
summary(lme1)
confint(lme1)

# btw
?pvalues
?convergence
```

Visualization of data



Visualization of data



Comparison of sample and model estimates

For this example, we are able to compare the "true" values to the parameter estimates

Parameter	Sample	Model
$\hat{\beta}_0$	522.2	522.11
$\hat{\beta}_1$	-19.00	-18.89
$\hat{\sigma}_\omega$	20.59	21.10
$\hat{\sigma}_{\upsilon_0}$	23.62	23.89
$\hat{\sigma}_{\upsilon_1}$	9.76	9.00
$\hat{\rho}_{\upsilon_0 \upsilon_1}$	-0.71	-1.00
$\hat{\sigma}_\varepsilon$	8.55	9.90

$$y_{ij} = \beta_0 + \beta_1 SOA_j + \omega_{0j} + \upsilon_{0i} + \upsilon_{1i} SOA_j + \varepsilon_{ij}$$

with $\boldsymbol{\upsilon} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_\upsilon = \begin{pmatrix} \sigma^2_{\upsilon_0} & \sigma_{\upsilon_0 \upsilon_1} \\ \sigma_{\upsilon_0 \upsilon_1} & \sigma^2_{\upsilon_1} \end{pmatrix}\right)$, $\omega_{0j} \sim N(0, \sigma^2_\omega)$, $\varepsilon_{ij} \sim N(0, \sigma^2_\varepsilon)$

Linear mixed-effects model

- The linear mixed-effects model has the general form

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \boldsymbol{\upsilon}_i + \boldsymbol{\varepsilon}_i$$

with fixed effects $\boldsymbol{\beta}$, random effects $\boldsymbol{\upsilon}_i$, and the design matrices $\mathbf{X}_i$ and $\mathbf{Z}_i$ and the assumptions

$$\boldsymbol{\upsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\upsilon) \text{ i.i.d.}, \qquad \boldsymbol{\varepsilon}_i \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_{n_i}) \text{ i.i.d.}$$

Linear mixed-effects model

$$
\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} z_{10} & z_{11} & \dots & z_{1q} & \dots \\ z_{20} & z_{21} & \dots & z_{2q} & \dots \\ z_{30} & z_{31} & \dots & z_{3q} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{N0} & z_{N1} & \dots & z_{Nq} & \dots \end{pmatrix} \cdot \begin{pmatrix} \upsilon_{10} \\ \vdots \\ \upsilon_{1q} \\ \upsilon_{20} \\ \vdots \\ \upsilon_{Nq} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_N \end{pmatrix}
$$

Simulate data using model matrices

```
X <- model.matrix( ~ soa, datsim)
Z <- model.matrix( ~ 0 + item + subject + subject:soa, datsim,
  contrasts.arg =
    list(subject = contrasts(datsim$subject, contrasts = FALSE)))

# Fixed effects
beta  <- c(beta0, beta1)

# Random effects
theta <- c(w = unique(w),
           y0 = y01[,1],
           y1 = y01[,2])

datsim$rt2 <- X %*% beta + Z %*% theta + e
```

Exercise

- Change the data simulation from the previous slides for $N = 30$ subjects instead of only 3.
- Download the script `simulation_baayen.R` and adjust it accordingly.
- You can choose if you want to use model matrices or create the vectors "manually."

Two-way repeated measures ANOVA

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \pi_k + (\pi\alpha)_{ik} + (\pi\beta)_{jk} + \varepsilon_{ijk}$$
$$i = 1, \dots, p; j = 1, \dots, q; k = 1, \dots, n$$

with

$$\begin{aligned} \pi_k &\sim N(0, \sigma^2_\pi) \\ (\pi\alpha)_{ik} &\sim N(0, \sigma^2_{\pi\alpha}) \\ (\pi\beta)_{jk} &\sim N(0, \sigma^2_{\pi\beta}) \\ \varepsilon_{ijk} &\sim N(0, \sigma^2_\varepsilon) \end{aligned}$$

all random effects independent

From model to data

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \pi_k + (\pi\alpha)_{ik} + (\pi\beta)_{jk} + \varepsilon_{ijk}$$

subj	μ	α	β	$(\alpha\beta)$	π	$(\pi\alpha)$	$(\pi\beta)$	ε
1	500	10	20	−30	0.82	3.72	−8.61	−15.20
1	500	10	−20	30	0.82	3.72	−0.64	25.85
1	500	−10	20	30	0.82	4.98	−8.61	−12.13
1	500	−10	−20	−30	0.82	4.98	−0.64	−3.02
⋮								
30	500	10	20	−30	7.94	3.72	−8.61	−4.14
30	500	10	−20	30	7.94	3.72	−0.64	−5.85
30	500	−10	20	30	7.94	4.98	−8.61	−5.63
30	500	−10	−20	−30	7.94	4.98	−0.64	28.02

$\sigma_\pi = 10$ $\sigma_{\pi\alpha} = 7$

$\sigma_{\pi\beta} = 8$ $\sigma_\varepsilon = 15$

From model to data

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \pi_k + (\pi\alpha)_{ik} + (\pi\beta)_{jk} + \varepsilon_{ijk}$$

subj		y_{ijk}
1	$500 + 10 + 20 - 30 + 0.82 + 3.72 - 8.61 - 15.20$	520.73
1	$500 + 10 - 20 + 30 + 0.82 + 3.72 - 0.64 + 25.85$	499.75
1	$500 - 10 + 20 + 30 + 0.82 + 4.98 - 8.61 - 12.13$	455.06
1	$500 - 10 - 20 - 30 + 0.82 + 4.98 - 0.64 - 3.02$	522.14
$\vdots$		$\vdots$
30	$500 + 10 + 20 - 30 + 7.94 + 3.72 - 8.61 - 4.14$	538.91
30	$500 + 10 - 20 + 30 + 7.94 + 3.72 - 0.64 - 5.85$	475.17
30	$500 - 10 + 20 + 30 + 7.94 + 4.98 - 8.61 - 5.63$	468.68
30	$500 - 10 - 20 - 30 + 7.94 + 4.98 - 0.64 + 28.02$	560.30

Matrix notation

Effect coding

$$
\begin{pmatrix} y_{111} \\ \vdots \\ y_{ijk} \\ \vdots \\ y_{22n} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -1 & 1 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & -1 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} \mu \\ \alpha_2 \\ \beta_2 \\ (\alpha\beta)_{22} \end{pmatrix} + \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_n \\ (\pi\alpha)_{11} \\ \vdots \\ (\pi\alpha)_{2n} \\ (\pi\beta)_{11} \\ \vdots \\ (\pi\beta)_{2n} \end{pmatrix} + \begin{pmatrix} e_{111} \\ \vdots \\ e_{ijk} \\ \vdots \\ e_{22n} \end{pmatrix}
$$

From model to data

```r
# Set effect coding
options(contrasts = c("contr.sum", "contr.poly"))

n <- 30

dat <- expand.grid(A = factor(c("a1", "a2")),
                   B = factor(c("b1", "b2")),
                   subj = factor(1:n))

# Fixed effects (in ms), effect coding
beta <- c(mu = 500, a2 = -10, b2 = -20, ab22 = -30)

# Model matrix
X <- model.matrix(~ A * B, dat)
```

From model to data

```
# Variance components (SD in ms)
sp  <- 10
spa <-  7
spb <-  8
se  <- 15

# Random effects
u <- c(p = rnorm(n, sd = sp),
       pa = rnorm(2 * n, sd = spa),
       pb = rnorm(2 * n, sd = spb))

Z <- model.matrix(~ 0 + subj + subj:A + subj:B, dat,
  contrasts.arg = lapply(dat, contrasts, contrasts = FALSE))
```

From model to data

```r
# Calculate dependent variable
dat$RT <- X %*% beta + Z %*% u + rnorm(2*2*n, sd=se)

# Look at simulated data
with(dat, interaction.plot(A, B, RT, type = "b", pch = c(21, 16),
  ylim = c(400, 600)))
```

References

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008).Mixed-effects modeling with crossed random effects for subjects and items. *Journal of memory and language*, *59*(4), 390–412.

Wickelmaier, F. (2022).Simulating the power of statistical tests: A collection of R examples. *ArXiv*. https://arxiv.org/abs/2110.09836