Mixed-effects models

Advanced statistical methods and models in experimental design

Bartosz Maćkiewicz

lexdec data set

lexdec data set provides visual lexical decision latencies elicited from 21 subjects for a set of 79 words: 44 nouns for animals, and 35 nouns for plants (fruits and vegetables). An experimental design in which we have multiple subjects responding to multiple items is referred to as a repeated measures design. For each word (item), we have 21 repeated measures (one measure from each subject). At the same time, we have 79 repeated measures for each subject (one for each item).

```r
library(lme4)
library(languageR)
head(lexdec,3)[c("Subject", "RT", "Word", "Trial",
                 "NativeLanguage", "Frequency")]
```

```
##   Subject       RT   Word Trial NativeLanguage Frequency
## 1      A1 6.340359    owl    23        English  4.859812
## 2      A1 6.308098   mole    27        English  4.605170
## 3      A1 6.349139 cherry    29        English  4.997212
```

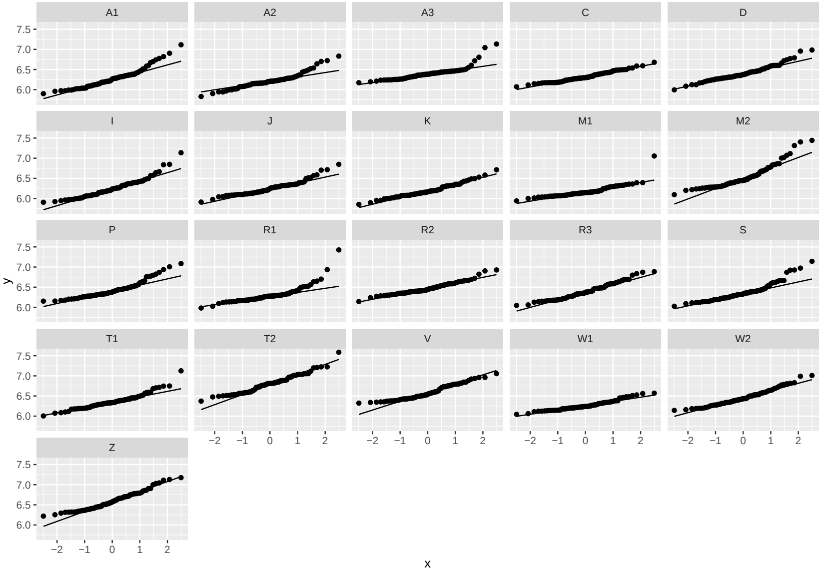
Fixed-effects factors, random-effects factors, and covariates

- Subject and item are random-effects factors;
- fixed-effects factors that are of interest include:
  - whether the subject was a native speaker of English
  - whether the word referred to an animal or a plant
  - lexical covariates such as frequency and length

The reaction times in `lexdec` are already logarithmically transformed. Nevertheless, it makes sense to inspect the distribution of the reaction times before beginning with fitting a model to the data.

Inspecting the data

```
ggplot(data = lexdec, aes(sample = RT)) + stat_qq() +
  stat_qq_line() + facet_wrap(vars(Subject))
```



Cleaning the data

- Removal of errors such as very short (e.g. $< 200ms$) and very long times
- Statistical outliers: do we need to remove observations to make the distribution more "normal"?
- Remove extreme values? Two or three SDs? But: unnecessary data trimming!

Cleaning the data

```
# RTs are logartihmically transformed
# 7 corresponds to roughly 1100ms (exp(7)) and does work fo
# Alernatively we can go through the data for each particip
lexdec2 <- lexdec %>% filter(RT < 7)
nrow(lexdec) - nrow(lexdec2)
```

```
## [1] 41
```

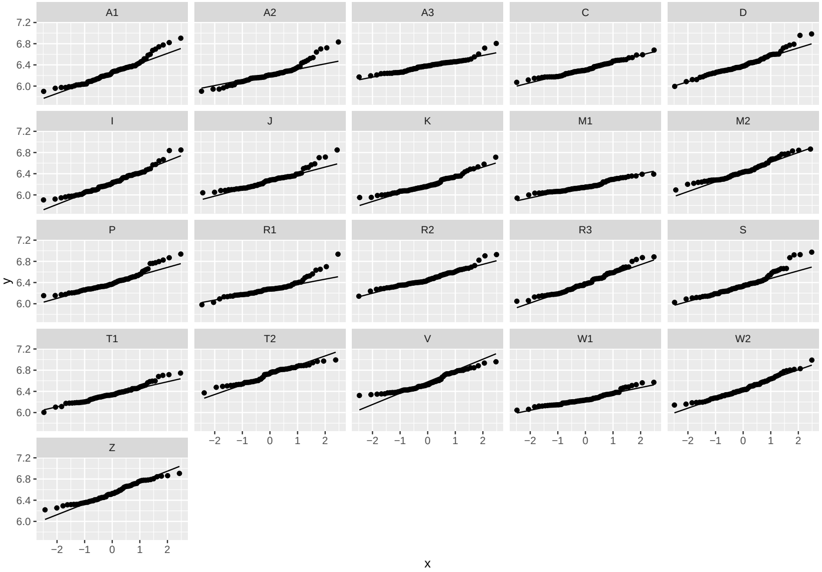
```
(nrow(lexdec) - nrow(lexdec2))/ nrow(lexdec)
```

```
## [1] 0.02471368
```

```
lexdec3 <- lexdec2 %>% filter(Correct == "correct")
```

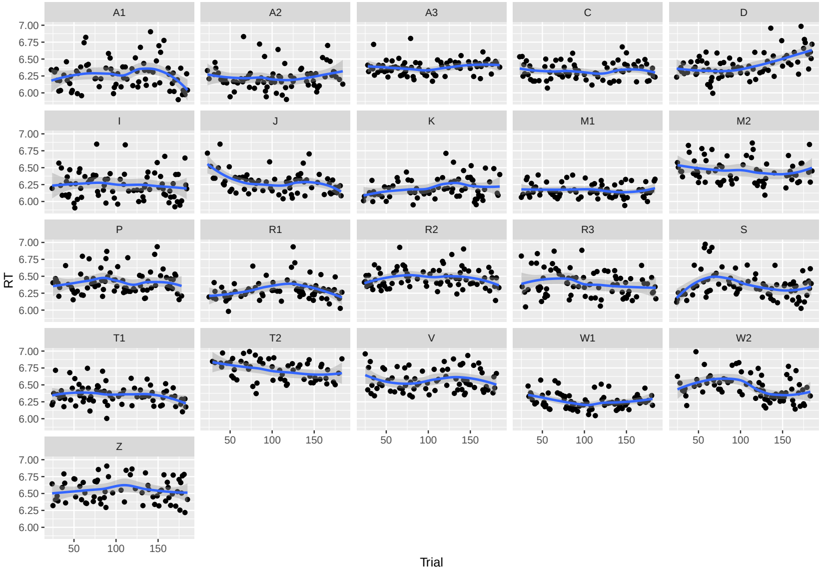
Inspecting the data: after clean-up

```
ggplot(data = lexdec3, aes(sample = RT)) + stat_qq() +
  stat_qq_line() + facet_wrap(vars(Subject))
```



Inspecting the data: familiarization fatigue?

```
ggplot(data = lexdec3, aes(x = Trial, y = RT)) + geom_point() +
  geom_smooth() + facet_wrap(vars(Subject))
```



Fitting the first model: familiarization fatigue?

```
fit_fam <- lmer(RT ~ Trial + (1|Subject) - (1|Word),
                data = lexdec3)
```

Structure of random and fixed effects:

- `Trial` - fixed factor; reaction times are modeled as depending on `Trial`
- `Subject` - random factor (sampled from all possible subjects)
  - intuitively, subjects will differ in terms of their base reaction speed (quick and slow responders); we need to make by-subject adjustments.
- `Word` - random factor (sampled from all possible words)
  - intuitively, words would differ in terms of their difficulty; more difficult words elicit longer response latiencies; we need to make by-word adjustments

Inspecting the model: random effects

The table with random effects provides information on three random effects, listed under the heading `Groups`: `Word`, `Subject`, and `Residual`. Residual stands for the residual error, the unexplained variance.

```
summary(fit_fam)
```

Inspecting the model: adjustments

The actual adjustments for specific subjects and specific words to the intercept can be extracted from the model with the `ranef()` function

```
head(ranef(fit_fam)$Subject, 3)
```

```
##     (Intercept)
## A1 -0.103186676
## A2 -0.141076806
## A3  0.005093286
```

```
head(ranef(fit_fam)$Word, 3)
```

```
##         (Intercept)
## almond  0.007609426
## ant    -0.040926537
## apple  -0.104050562
```

Inspecting the model: fixed effects

The part of the summary dealing with the fixed effects is already familiar from the summaries for objects created by the `lm()` function for models with fixed effects only.

The table lists the coefficients of the fixed effects, in this case the coefficient for the intercept and for the slope of `Trial`, and their associated standard errors and *t*-values.

```
summary(fit_fam)
```

Inspecting the model: fitted values

Let's reconstruct how the model arrived at the fitted reaction time of 6.272 for subject `A1` to item *owl* at trial 23.

```
lexdec3[1, c("Subject", "RT", "Word", "Trial")]
```

```
##   Subject       RT Word Trial
## 1      A1 6.340359  owl    23
```

```
fitted(fit_fam)[1:4]
```

```
##        1        2        3        4
## 6.272059 6.318508 6.245524 6.254167
```

```
6.3939620 + # grand intercept
  ranef(fit_fam)$Subject["A1",] + # intercept for Subject "A1"
  ranef(fit_fam)$Word["owl", ] + # intercept for Word "owl"
  -0.0001835  * 23 # slope of Trial * Trial
```

```
## [1] 6.272059
```

Inspecting the model: statistical testing

For some time it is not perfectly clear how to calculate the appropriate degrees of freedom for statistical tests on fixed effects. That's why by defailt `lmer` objects do not print them.

`lmerTest` provides a wrapper to `lmer` function that runs the tests using Satterthwaite's degrees of freedom method.

```
library(lmerTest)
fit_fam <- lmer(RT ~
                  Trial +
                  (1|Subject) +
                  (1|Word),
                data = lexdec3)
#summary(fit_fam)
```

Fitting the second model: random slopes

It is somewhat surprising that the effect of Trial does seem to reach significance, even if only at the 5% level.

What we have seen is that some subjects show an effect, sometimes in opposite directions, but also that many subjects have no clear effect at all.

In terms of model building, what we would like to do is to allow the slope of the effect of `Trial` to vary across subjects. In other words, what we need here are by-subject random slopes for Trial

We build these into the model by expanding the expression for the subject random-effects structure:

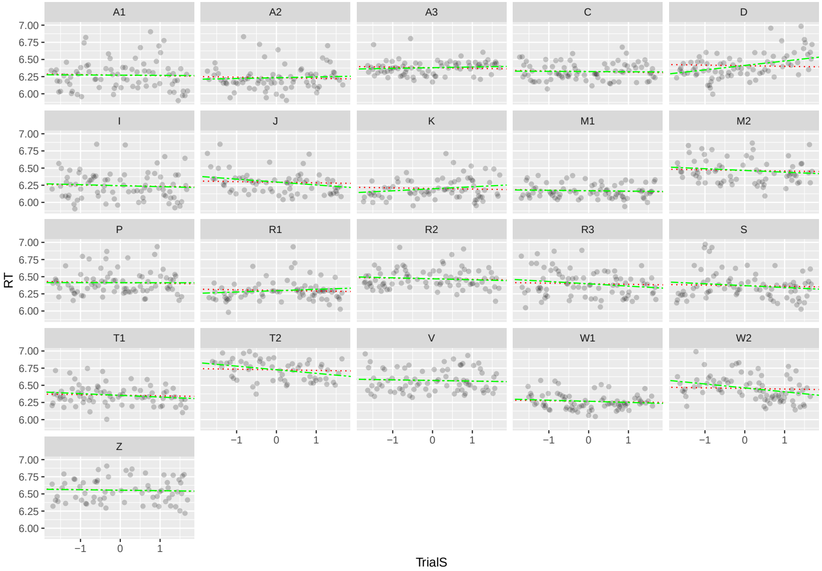
```
lexdec3$TrialS <- scale(lexdec3$Trial) # rescale variable
fit_fam_slopes <- lmer(RT ~ TrialS +
                         (1 +TrialS|Subject) +
                         (1|Word),
                       data = lexdec3)
#summary(fit_fam_slopes)
```

Inspecting the second model: comparing random and fixed slopes

```
fit_fam_slopes <- lmer(RT ~ TrialS + (1 +TrialS|Subject) - (1|Word),
                       data = lexdec3)
fit_fam <- lmer(RT ~ TrialS + (1|Subject) - (1|Word),
                data = lexdec3)
coefs_fam <- coef(fit_fam)$Subject
coefs_fam$Subject <- rownames(coefs_fam)
colnames(coefs_fam) <- c("InterceptRI", "SlopeRI", "Subject")
coefs_fam_slopes <- coef(fit_fam_slopes)$Subject
coefs_fam_slopes$Subject <- rownames(coefs_fam_slopes)
colnames(coefs_fam_slopes) <- c("InterceptRIS", "SlopeRIS", "Subject")
lexdec3 <- left_join(lexdec3, coefs_fam, by = "Subject")
lexdec3 <- left_join(lexdec3, coefs_fam_slopes, by = "Subject")

ggplot(data = lexdec3, aes(x = TrialS, y = RT)) +
  geom_point(alpha = 0.2) +
  geom_abline(aes(intercept = InterceptRI, slope = SlopeRI),
              linetype = 3, colour = "red") +
  geom_abline(aes(intercept = InterceptRIS, slope = SlopeRIS),
              linetype = 6, colour = "green") +
  facet_wrap(vars(Subject))
```

Inspecting the second model: comparing random and fixed slopes



Fitting the third model: differences between native and non-native speakers

Does the experiment also reveal differences between native and non-native speakers of English? The data frame lexdec3 contains a column labeled `NativeLanguage` for this fixed-effects factor, with levels `English` and `Other`:

```
fit_nat <- lmer(RT ~
                  NativeLanguage + # fixed
                  TrialS + # fixed
                  (1 +TrialS|Subject) + # random
                  (1|Word), # random
                data = lexdec3)
#summary(fit_nat)
```

Differences between native and non-native speakers

- There indeed appears to be support for the possibility that the non-native speakers are the slower responders. (main effect of the `NativeLanguage`)
  - Since the reference level for `NativeLanguage` is `English`, we note that non-native speakers of English had significantly longer response latencies.
- Since native speakers have more experience with their language, the frequency effect might be stronger for native speakers, leading to greater facilitation. (main effect of the `Frequency` as well as an interaction between `Frequency` and `NativeLanguage`)
- There indeed appears to be support for the possibility that the non-native speakers are the slower responders. Since native speakers have more experience with their language, the frequency effect might be stronger for native speakers, leading to greater facilitation. We test this hypothesis by including Frequency as a predictor, together with an interaction of NativeLanguage by Frequency

Fitting the fourth model: differences between native and non-native speakers

```
fit_nat_freq <- lmer(RT ~
                       NativeLanguage * Frequency + # fixed
                       TrialS + # fixed
                       (1 +TrialS|Subject) + # random
                       (1|Word), # random
                     data = lexdec3)
#summary(fit_nat_freq)
```

Fitting the fifth model: differences between native and non-native speakers

Possibly, we are led astray by a confound with word length — more frequent words tend to be shorter, and non-native readers might find shorter words easier to read compared to native readers. When we add a Length by Native-Language interaction to the model, inspection of the summary shows that the `Frequency` by `NativeLanguage` interaction is no longer significant, in contrast to the interaction of `NativeLanguage` by Length:

```
fit_nat_freq_len <- lmer(RT ~
                           NativeLanguage * Frequency + # fixed
                           NativeLanguage * Length + # fixed
                           TrialS + # fixed
                           (1 +TrialS|Subject) + # random
                           (1|Word), # random
                         data = lexdec3)
#summary(fit_nat_freq_len)
```

Fitting the sixth model: differences between native and non-native speakers

We therefore take the spurious NativeLanguage:Frequency interaction out of the model.

```
fit_nat_freq_lenB <- lmer(RT ~
                            Frequency + # fixed
                            NativeLanguage * Length + # fixed
                            TrialS + # fixed
                            (1 +TrialS|Subject) + # random
                            (1|Word), # random
                          data = lexdec3)
#summary(fit_nat_freq_lenB)
```

Note that the Length by NativeLanguage interaction makes sense. For native readers, there is no effect of Length, while non-native readers require more time to respond to longer words.

Inspecting the sixth model: correlation between random effects

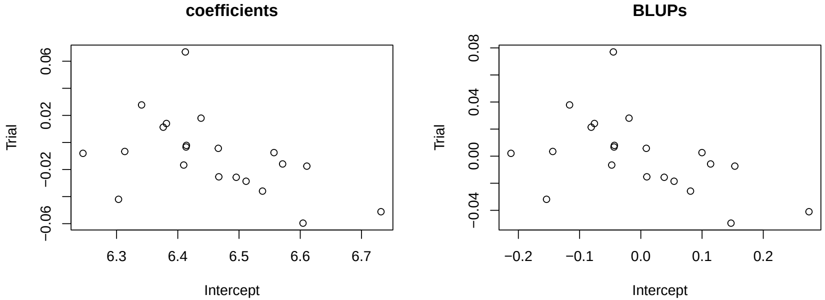
Thus far, we have examined only the table of coefficients. Let's redress our neglect of the table of random effects.

In addition to the usual standard deviations listed in the fourth column, the final column of the random effects table lists a correlation. This correlation concerns the by-subject random intercepts and the by-subject random slopes for `Trial`. Since we have random slopes and random intercepts that are paired by subject, it is possible that the vectors of random slopes and random intercepts are correlated.

The way in which we specified the random-effects structure for Subject, `(1 + Trial | Subject)`, explicitly instructed `lmer()` to allow for this possibility by including a special parameter for this correlation of the BLUPS (*best linear unbiased prediction*) for the intercept and the BLUPS for `Trial`.

Inspecting the sixth model: correlation between random effects

```
par(mfcol = c(1,2))
plot(coef(fit_nat_freq_lenB)$Subject$`(Intercept)`,
     coef(fit_nat_freq_lenB)$Subject$`TrialS`,
     main = "coefficients", xlab = "Intercept", ylab = "Trial")
plot(ranef(fit_nat_freq_lenB)$Subject$`(Intercept)`,
     ranef(fit_nat_freq_lenB)$Subject$`TrialS`,
     main = "BLUPs", xlab = "Intercept", ylab = "Trial")
```



Inspecting the sixth model: correlation between random effects

In this scatterplot, each data point represents a subject. Subjects with a large negative adjustment for the intercept are fast responders, subjects with a large positive adjustment are slow responders. Fast responders have positive adjustments for `Trial`, while slow responders have negative adjustments for Trial.

Since the estimated fixed-effects coefficient for `Trial` equals a mere -0.0002, the fastest responders appear to slow down in the course of the experiment, whereas the slowest responders speed up.

Inspecting the sixth model: comparing models

The total number of parameters in `fit_nat_freq_lenB` is 12: we have 7 fixed effects coefficients (including the intercept), and 5 random-effects parameters.

The question that arises at this point is whether all these random-effects parameters are justified. The significance of parameters for random effects is assessed by means of likelihood ratio tests, which are carried out by the `anova()` function when supplied with two mixed-effects models that have the same fixed-effects structure but different numbers of random-effects parameters.

For instance, we can evaluate the significance of the two by-subject random effects for `Subject` by fitting a simpler model with only a by-subject random intercept that we then compare with the full model:

Inspecting the sixth model: comparing models

The likelihood ratio test takes the log likelihood (logLik, an important measure of goodness of fit) for the smaller model with 9 parameters (Df) and compares it with the log likelihood for the larger model with 11 parameters.

```
fit_nat_freq_lenB1 <- lmer(RT ~
                             Frequency + # fixed
                             NativeLanguage * Length + # fixed
                             TrialS + # fixed
                             (1|Subject) + # random
                             (1|Word), # random
                           data = lexdec3)
# anova(fit_nat_freq_lenB, fit_nat_freq_lenB1)
```

As the associated probability is small, the additional parameters in the more complex model are justified.