Week 10: Random slopes; nested vs crossed designs

Linear mixed-effects models (building the random-effects structure)

Bartosz Maćkiewicz

Recap (Week 9)

Random intercept model for repeated measures:

$$Y_{ij} = (\beta_0 + b_{0j}) + \beta_1 X_{ij} + e_{ij}$$

`lme4` formula:

```
lmer(Y ~ X + (1 | subject), data = df)
```

schools dataset

The example used for this class is fictional data where the interval scaled outcome variable “Extroversion” (**extro**) is predicted by fixed effects for the interval scaled predictor “Openness to new experiences” (**open**), the interval scaled predictor “Agreeableness (**agree**)”, the interval scaled predictor “Social engagement” (**social**), and the nominal scaled predictor “Class” (**class**); as well as the random (nested) effect of Class within School (**school**).

The data contains 1200 cases evenly distributed among 24 nested groups (4 classes within 6 schools).

```
library(lme4)
library(lmerTest)
schools <- read.csv("schools.csv")
head(schools, 4)
##   X id    extro     open    agree    social class school
## 1 1  1 63.69356 43.43306 38.02668  75.05811     d     IV
## 2 2  2 69.48244 46.86979 31.48957  98.12560     a     VI
## 3 3  3 79.74006 32.27013 40.20866 116.33897     d     VI
```

Structure of effects

We know that “Openness”, “Agreeableness” and “Social engagement” will be modelled as fixed factors.

How to model the random effects od School and Class?

Nested vs Crossed Random Effects (Harrison et al. 2018)

A common issue that causes confusion is this issue of specifying random effects as either **'crossed'** or **'nested'**.

In reality, the way you specify your random effects will be determined by your experimental or sampling design.

A simple example can illustrate the difference. Imagine a researcher was interested in understanding the factors affecting the clutch mass of a passerine bird. They have a study population spread across five separate woodlands, each containing 30 nest boxes. Every week during breeding they measure the foraging rate of females at feeders, and measure their subsequent clutch mass. Some females have multiple clutches in a season and contribute multiple data points. Here, female ID is said to be nested within woodland: each woodland contains multiple females unique to that woodland (that never move among woodlands).

Nested vs Crossed Random Effects (Harrison et al. 2018)

The nested random effect controls for the fact that

(i) clutches from the same female are not independent, and

(ii) females from the same woodland may have clutch masses more similar to one another than to females from other woodlands

We can write the appropriate model model as:

```
Clutch Mass ~ Foraging Rate + (1|Woodland/Female ID)
```

Nested vs Crossed Random Effects (Harrison et al. 2018)

Now imagine that this is a long-term study, and the researcher returns every year for five years to continue with measurements. Here it is appropriate fit year as a crossed random effect because every woodland appears multiple times in every year of the dataset, and females that survive from one year to the next will also appear in multiple years.

```
Clutch Mass ~ Foraging Rate + (1|Woodland/Female ID)+ (1|Year)
```

Nested vs Crossed Random Effects (Harrison et al. 2018)

Understanding whether your experimental/sampling design calls for nested or crossed random effects is not always straightforward, but it can help to visualise experimental design by drawing it, or tabulating your observations by these grouping factors (e.g. with the `table` command in R) to identify how your data are distributed.

Always ensure that their levels of random effect grouping variables are uniquely labelled. For example, females are labelled $1 - n$ in each woodland, the model will try and pool variance for all females with the same code. Giving all females a unique code makes the nested structure of the data is implicit, and a model specified as

```
`Clutch Mass ~ Foraging Rate + (1| Woodland) + (1|FemaleID)
```

would be identical to the model above.

Nested vs Crossed Random Effects (Schielzeth & Nakagawa 2012)

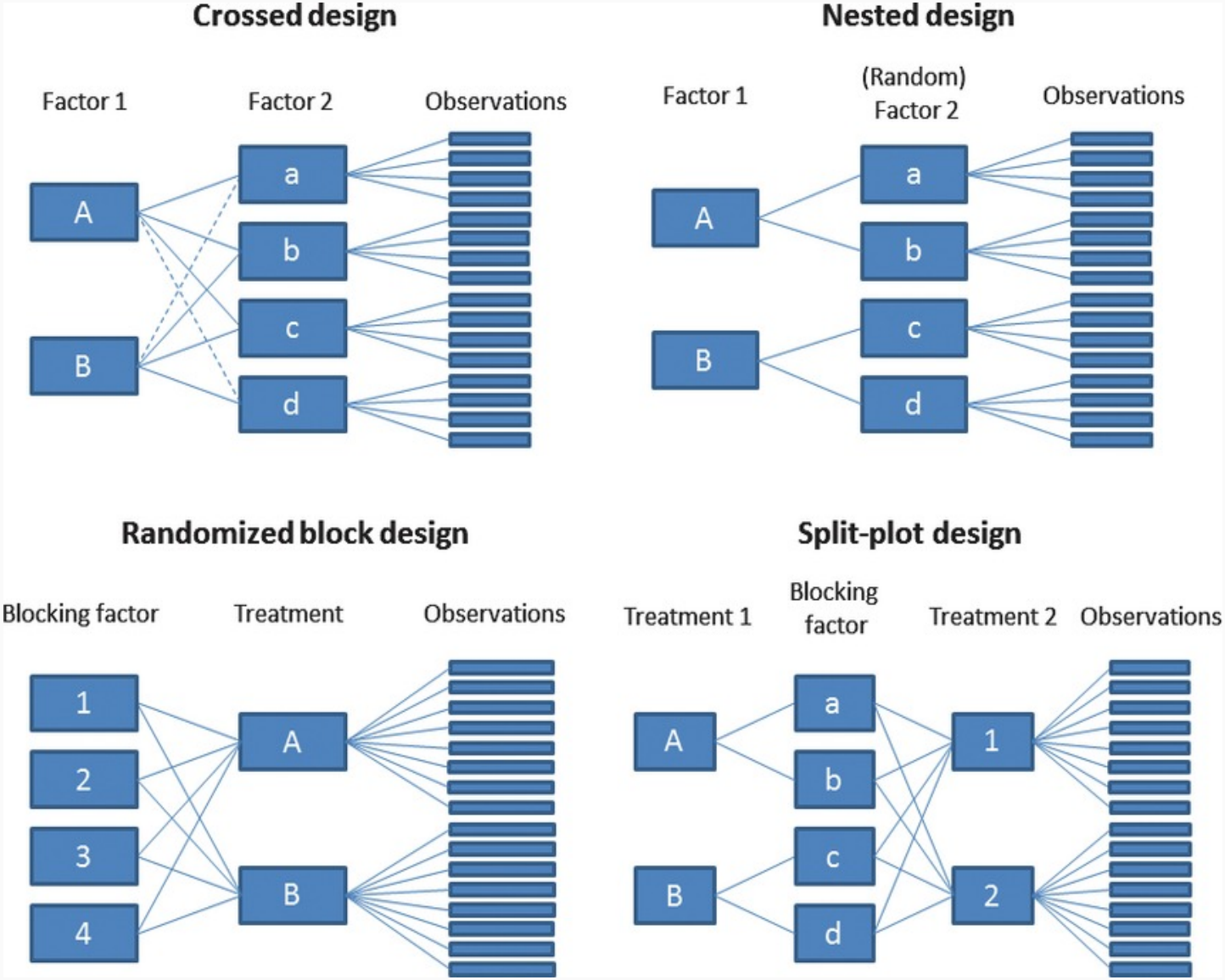


Figure 1: Common experimental designs

Crossed Random Effects

```
fit_crossed <- lmer(formula = extro ~ # Extroversion
                      open + # fixed effect of Openness
                      agree + # fixed effect of Agreeableness
                      social + # fixed effect of Social engagement
                      (1|school) + # random intercept for each school
                      (1|class), # random intercept for each class
                    data = schools)
summary(fit_crossed)
```

Is this model appropriate given the structure of the data?

Nested Random Effect

```
fit_nested <- lmer(formula = extro ~ # Extroversion
                     open + # fixed effect of Openness
                     agree + # fixed effect of Agreeableness
                     social + # fixed effect of Social engagement
                     (1|school/class),
                     # random intercept for each class WITHIN the school
                   data = schools)
summary(fit_nested)
```

This is a shorthand for:

```
fit_nested_short <- lmer(formula = extro ~
                           open +
                           agree +
                           social +
                           (1|school) + # random intercept for each school
                           (1|class:school),
                           # and for each class WITHIN school
                   data = schools)
summary(fit_nested_short)
```

Modeling the data using LME models: this is really the end

This is the end of the nested vs crossed demonstration using the `schools` dataset. Next, we turn to random slopes and why they matter for inference.

Random slopes? (Harrison et al. 2018)

Often, researchers fit random intercepts to control for non-independence among measurements of a statistical group (e.g. birds within a woodland), but force variables to have a common slope across all experimental units. However, there is growing evidence that researchers should be fitting random slopes as standard practice in (G)LMMs. Random slope models allow the coefficient of a predictor to vary based on clustering/non-independence in the data.

In our bird example above, we might fit a random slope for the effect of foraging rate on clutch mass given each individual bird ID. That is, the magnitude of the effect foraging rate on resultant clutch mass differs among birds.

Random slopes? (Harrison et al. 2018)

Schielzeth & Forstmeier (2009) found that including random slopes controls Type I error rate (yields more accurate *p* values), but also gives more power to detect among individual variation. Barr et al. (2013) suggest that researchers should fit the **maximal random effects structure possible** for the data. That is, if there are four predictors under consideration, all four should be allowed to have random slopes. However, we believe this is unrealistic because random slope models require large numbers of data to estimate variances and covariances accurately (Bates et al., 2015a).

`lexdec` data set

The `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). This is a repeated-measures design: each word is seen by multiple subjects, and each subject responds to multiple words.

This is why both subjects and words enter the model as random-effects grouping factors.

```
head(
  lexdec3[c(
    "Subject", "RT", "Word", "Trial",
    "NativeLanguage", "Frequency"
  )],
  2
)
```

```
  Subject       RT Word Trial NativeLanguage Frequency
1      A1 6.340359  owl    23        English  4.859812
2      A1 6.308098 mole    27        English  4.605170
```

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

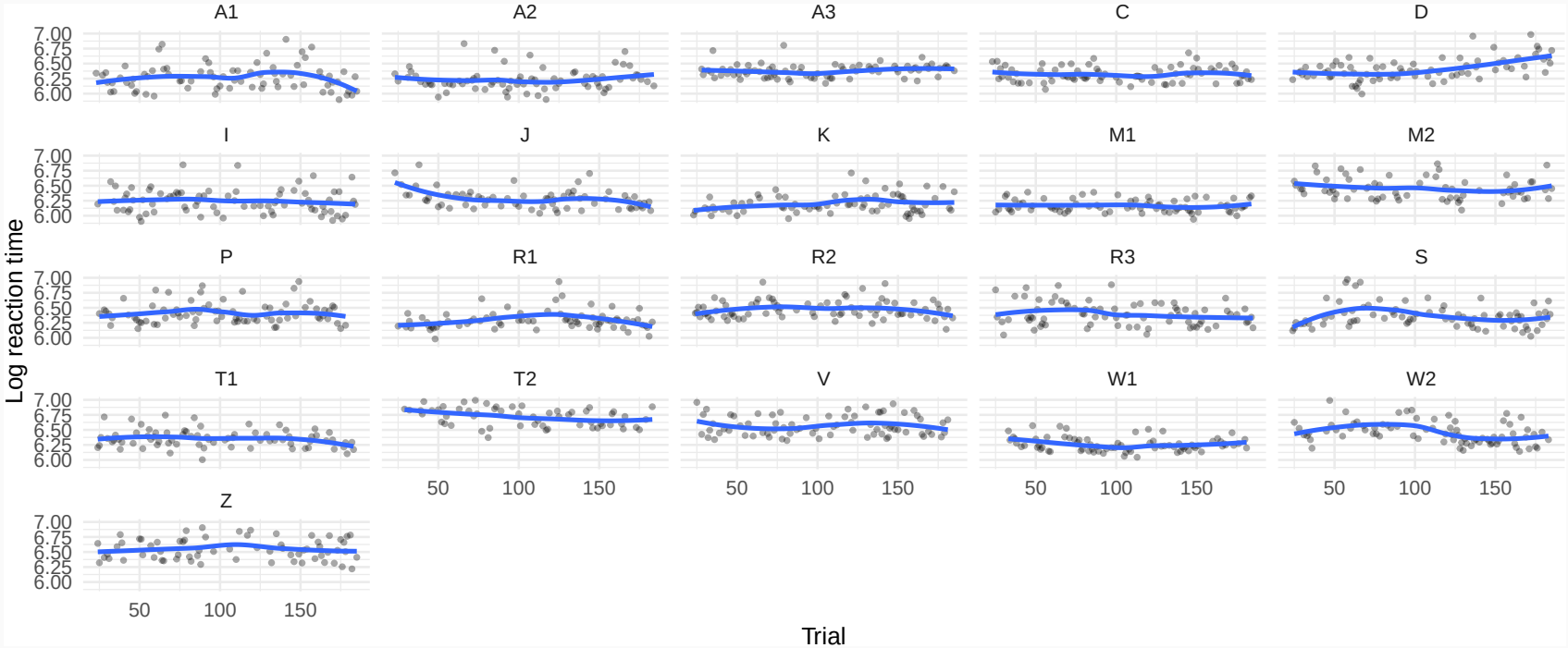
Subject and item are random-effects factors: subjects are sampled from a broader population of possible readers, and words are sampled from a broader population of possible words.

Fixed-effects factors and covariates of interest include whether the subject was a native speaker of English, whether the word referred to an animal or a plant, and lexical covariates such as frequency and length.

The reaction times in `lexdec` are already logarithmically transformed. We also remove very long reaction times and incorrect responses before modeling.

Inspecting the data: familiarization fatigue?

```
ggplot(lexdec3, aes(x = Trial, y = RT)) +
  geom_point(alpha = 0.35, size = 0.8) +
  geom_smooth(se = FALSE) +
  facet_wrap(vars(Subject)) +
  labs(x = "Trial", y = "Log reaction time")
```



Fitting the first model: familiarization fatigue?

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

The fixed effect of `TrialS` asks whether reaction times change over the course of the experiment. The random intercept for `Subject` allows different subjects to have different baseline reaction times. The random intercept for `Word` allows different words to have different baseline difficulty.

Inspecting the first model: random and fixed effects

The table with random effects provides information on the grouping factors and the residual error. The fixed-effects table is familiar from models fitted with `lm()`: it lists coefficients, standard errors, test statistics, and *p*-values.

```
VarCorr(fit_fam)
```

```
 Groups   Name        Std.Dev.
 Word     (Intercept) 0.068249
 Subject  (Intercept) 0.136485
 Residual             0.150214
```

```
coef(summary(fit_fam))
```

```
               Estimate  Std. Error         df    t value     Pr(>|t|)
(Intercept)  6.374684591 0.030995361   22.58008 205.665764 1.862059e-38
TrialS      -0.008687377 0.003879656 1476.33459  -2.239213 2.529049e-02
```

Inspecting the first 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 first model: fitted values

The fitted value for one observation combines the population-level fixed effects with the subject-specific and word-specific random-effect adjustments. For the first row:

```
row1 <- lexdec3[1, ]
subj1 <- as.character(row1$Subject)
word1 <- as.character(row1$Word)

manual_fit <-
  fixef(fit_fam)[["(Intercept)"]] +
  fixef(fit_fam)[["TrialS"]] * row1$TrialS +
  ranef(fit_fam)$Subject[subj1, 1] +
  ranef(fit_fam)$Word[word1, 1]

round(c(manual = manual_fit, model = fitted(fit_fam)[1]), 4)
```

```
 manual model.1
 6.2721  6.2721
```

Inspecting the first model: statistical testing

For some time it was not perfectly clear how to calculate the appropriate degrees of freedom for statistical tests on fixed effects. That is why `lmer` objects do not print the same fixed-effect significance tests by default.

The package `lmerTest` provides a wrapper to `lmer()` that runs the tests using Satterthwaite's degrees of freedom method. In this deck, `lmerTest` is loaded before the models are fitted, so `summary()`, `coef(summary())`, and `anova()` give the familiar tests for fixed effects.

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.

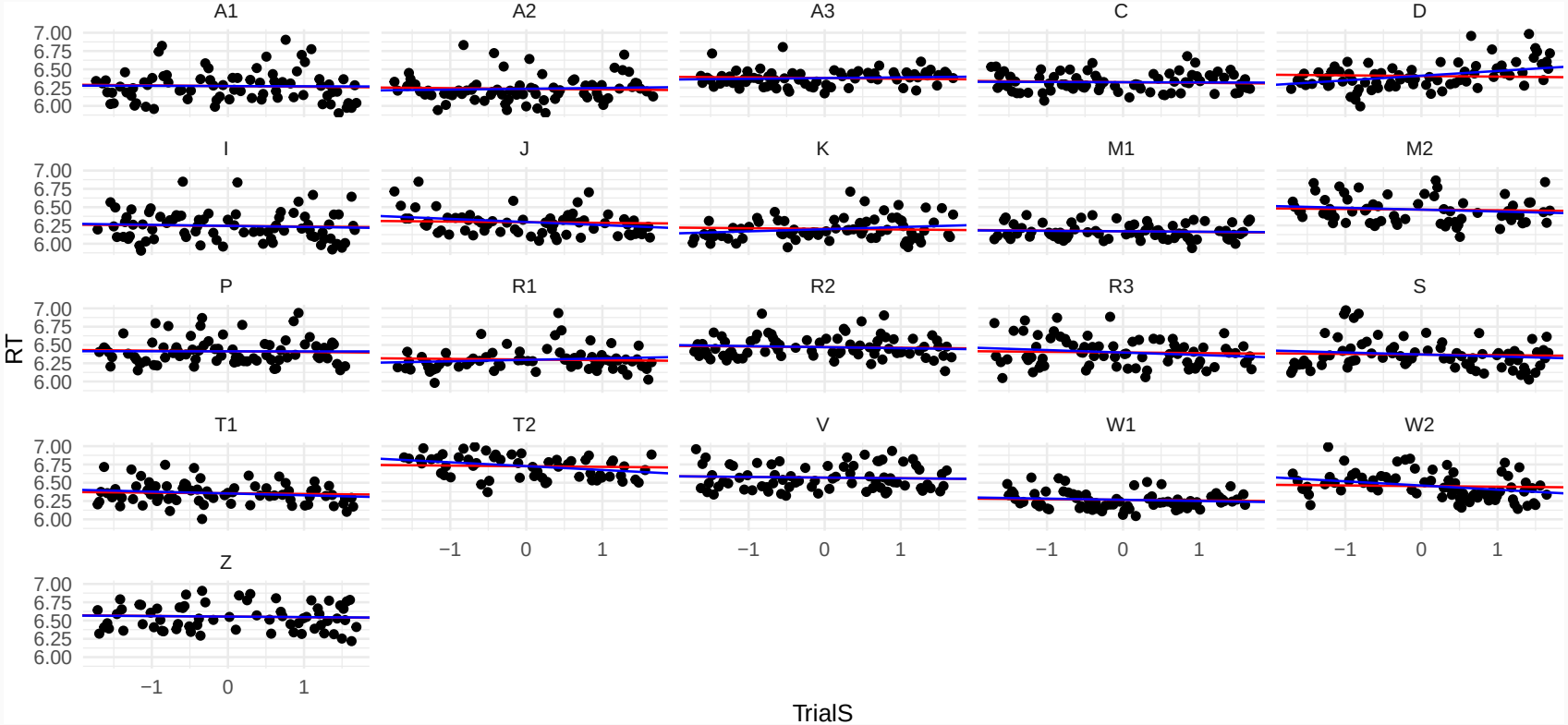
What we have seen is that some subjects show an effect, sometimes in opposite directions, but also that many subjects have no clear effect at all.

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:

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

Inspecting the second model: comparing random and fixed slopes



Red lines show the model with random intercepts only. Blue lines show the model with by-subject random slopes for `Trial`.

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 + TrialS +
    (1 + TrialS | Subject) +
    (1 | Word),
  data = lexdec3
)
```

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. 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.
- 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 + TrialS +
    (1 + TrialS | Subject) +
    (1 | Word),
  data = lexdec3
)
```

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 +
    NativeLanguage * Length +
    TrialS +
    (1 + TrialS | Subject) +
    (1 | Word),
  data = lexdec3
)
```

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 +
    NativeLanguage * Length +
    TrialS +
    (1 + TrialS | Subject) +
    (1 | Word),
  data = lexdec3
)
```

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

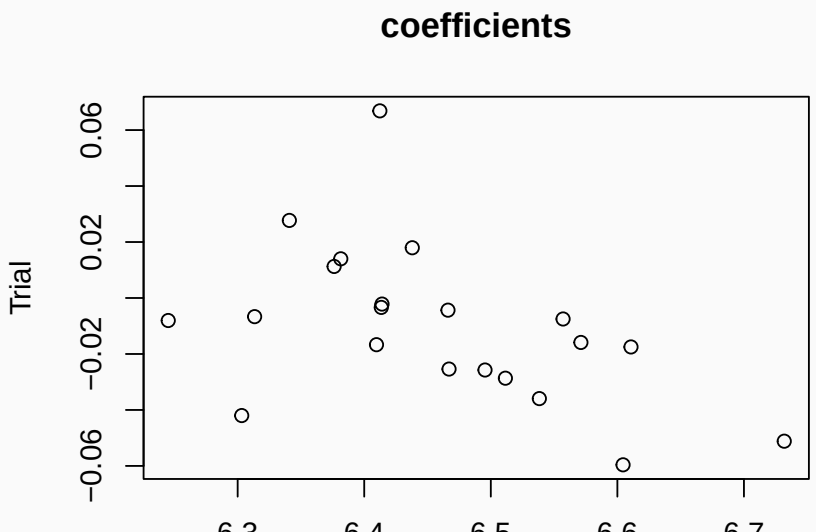
Thus far, we have examined only the table of coefficients. Let's redress our neglect of the table of random effects.

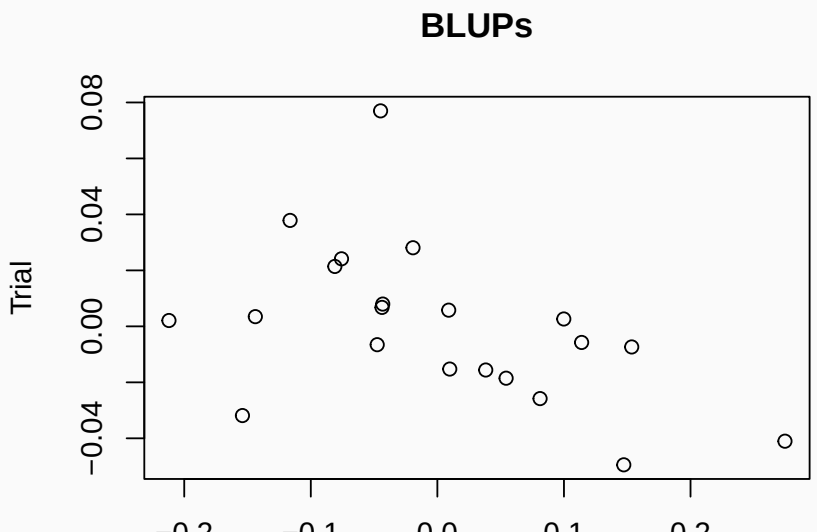
In addition to the usual standard deviations listed in the fourth column, the final column of the random effects table lists a correlation. This correlation concerns the by-subject random intercepts and the by-subject random slopes for `Trial`. Since we have random slopes and random intercepts that are 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 +
    NativeLanguage * Length +
    TrialS +
    (1 | Subject) +
    (1 | Word),
  data = lexdec3
)

anova(fit_nat_freq_lenB1, fit_nat_freq_lenB)
```

```
                  npar    Chisq Df   Pr(>Chisq)
fit_nat_freq_lenB1    9       NA NA           NA
fit_nat_freq_lenB    11 37.67125  2 6.603764e-09
```

Model complexity?

Guidelines for the ideal ratio of data points (n) to estimated parameters (k) vary widely (see Forstmeier & Schielzeth, 2011). Crawley (2013) suggests a minimum n/k of 3, though we argue this is very low and that an n/k of 10 is more conservative. A 'simple' model containing a three-way interaction between continuous predictors, all that interaction's daughter terms, and a single random intercept needs to estimate eight parameters, so requires a dataset of a minimum n of 80 using this rule.

Fit and performance?

The strength of the Nakagawa & Schielzeth (2013) method for GLMMs is that it returns two complementary R^2 values: the marginal R^2 encompassing variance explained by only the fixed effects, and the conditional R^2 comprising variance explained by both fixed and random effects i.e. the variance explained by the whole model (Nakagawa & Schielzeth, 2013). Ideally, both should be reported in publications as they provide different information; which one is more 'useful' may depend on the rationale for specifying random effects in the first instance.

Here is how you can compute both types of R^2 using R:

```
library(MuMIn)
r.squaredGLMM(fit_nested)
```

Takeaways

- Random slopes let effects vary by participant/item/class.
- Decide nested vs crossed from the design (tables/diagrams help).
- Ensure grouping-factor IDs are uniquely labeled.
- Model complexity is a statistical and practical issue: the data must contain enough information to estimate the parameters we ask for.