Random slopes

Mixed Models 2

Daniela Palleschi

2024-01-12

Table of contents

Learning Objectives

Today we will learn…

- how to fit a random-intercepts and slopes model
- how to inspect and interpret random slopes

Resources

- this lecture covers
  - Chapter 14 'Mixed Models 1: Conceptual Introduction' (until Section 14.8; Winter, 2019)
  - Winter (2014) (from page 16)
  - Sections 8.4 onward in Sonderegger (2023)
  - Blog post "Plotting partial pooling in mixed-effects models" from Tristin Mahr (2017)
- we will be using the data from Biondo et al. (2022)

Set-up

```r
# suppress scientific notation
options(scipen=999)
```

```r
library(broman)
# function to format p-values
format_pval <- function(pval){
    dplyr::case_when(
        pval < .001 ~ "< .001",
        pval < .01 ~ "< .01",
        pval < .05 ~ "< .05",
        TRUE ~ broman::myround(pval, 3)
    )
}
```

Load packages

```r
# load libraries
pacman::p_load(
               tidyverse,
               here,
               broom,
               janitor,
               ggeffects,
               sjPlot,
               # new packages for mixed models:
               lme4,
               lmerTest,
               broom.mixed,
               lattice)
```

```r
lmer <- lmerTest::lmer
```

Load data

- data from Biondo et al. (2022)

```r
df_biondo <-
  read_csv(here("data", "Biondo.Soilemezidi.Mancini_dataset_ET.csv"),
           locale = locale(encoding = "Latin1") ## for special characters in Spanish
           ) |>
  clean_names() |>
  mutate(gramm = ifelse(gramm == "0", "ungramm", "gramm")) |>
  mutate_if(is.character,as_factor) |> # all character variables as factors
  droplevels() |>
  filter(adv_type == "Deic")
```

0.1 Set contrasts

```r
contrasts(df_biondo$verb_t) <- c(-0.5,+0.5)
contrasts(df_biondo$gramm) <- c(-0.5,+0.5)
```

```r
contrasts(df_biondo$verb_t)
```

```
       [,1]
Past   -0.5
Future  0.5
```

```r
contrasts(df_biondo$gramm)
```

```
        [,1]
gramm   -0.5
ungramm  0.5
```

1 Review

1.1 Fixed-effects only models

- do not include any grouping factors
  - can be dangerously unconservative if violating independence assumption

1.1.1 Fixed-effects only equation

$$fp_i = \beta_0 + \beta_{verb\_t} x_i + \beta_{gramm} x_i + e_i \tag{1}$$

- Equation 1 shows the equation for such a model using first-pass reading times as a function of verb tense (`verb_t`) and grammaticality (`gramm`)
  - where i represents an observation (i = 1:N)
  - β_0 = intercept value
  - $\beta_{verb\_t} x$ = tense slope multiplied by the corresponding level (+/- 0.5)
  - $\beta_{gramm} x$ = grammaticality slope multiplied by the corresponding level (+/- 0.5)
  - e_i = residual error for this observation

1.2 Random intercepts only models

- random intercepts: varying intercepts per e.g., participant
  - intercept = mean when your predictor is centred (continuous) or sum contrast coded (categorical)
- explains some additional variance (i.e., should reduce our residual error)

1.2.1 Random intercepts model equation

- Equation 2 includes two additional terms:
  - $\alpha_{j[i]}$ = random intercept (α) for some grouping factor j
    * e.g., participants, where $i = 1 : 60$
  - $\alpha_{k[i]}$ = random intercept (α) for some grouping factor k
    * e.g., items, where $i = 1 : 96$

$$fp_i = \beta_0 + \alpha_{j[i]} + \alpha_{k[i]} + \beta_{verb\_t} x_i + \beta_{gramm} x_i + e_i \tag{2}$$

- $\alpha_{j[16]}$ = random intercept for participant 16
- $\alpha_{j[7]}$ = random intercept for item 1

1.3

Missing values and subsetting conditions

N.B., because we subsetted the data to include only `adv_type == "Deic"`, each participant did not contribute 96 data points to our current dataset, but 64.
So, our overall N observations should be `64*60`, minus however many missing observations we have + so i in fp_i has a value of 1:3840, minus missing values.
We can use the `nobs()` function to find out the number of observations in a model. For example our random-intercepts only model from the last class had 3795 observations, meaning we had 3840 - 3795 = 45 missing observations. This amounts to 1.17% or trials, which is fine (something around 5% of trials is not out of the ordinary).

```r
nobs(fit_fp_1)
```

```
[1] 3795
```

Why do we have missing values? This can depend on a lot of things, such as incorrect attention-check responses (not relevant for this data), measurement error, or pre-processing steps (likely the cause for this data, which is eye-tracking during reading).

1.3.1 Interpreting random effects

- how can we interpret this output from a model, without knowing anything else about the model?

```
Groups   Name        Std.Dev.
Word     (Intercept)  38.201
Subject  (Intercept)  91.004
Residual             127.258
```

1.3.2 Formulating a model

- can you formulate a model based on this output in the `lmer()` syntax?
  - let's call the depedent variable `rt` for reaction time

```
Groups   Name        Std.Dev.
Word     (Intercept)  38.201
Subject  (Intercept)  91.004
Residual             127.258
```

```
                  Estimate Std. Error   t value
(Intercept)       619.6160  20.761039 29.845135
NativeLanguage1   106.2954  40.622552  2.616659
freq_c            -29.4397   4.168753 -7.061990
```

1.3.3 Interpreting random effects

```
 Groups   Name        Std.Dev.
 Word     (Intercept)  38.201
 Subject  (Intercept)  91.004
 Residual             127.258
```

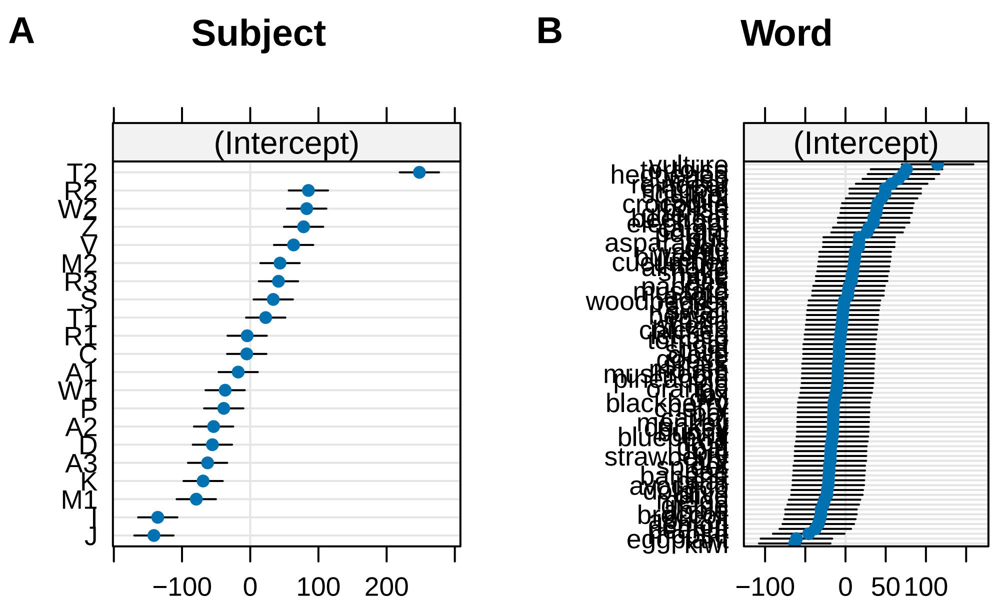

Figure 1: lattice::dotplot(ranef(model))

2 Random slopes

- random slopes: varying slopes
  - allows for different magnitude/sign of effects per e.g., participant
- recall that our model still produces by-participant and -item slopes
  - but they don't *vary*

```r
fixef(fit_fp_1)
```

```
 (Intercept)        verb_t1         gramm1 verb_t1:gramm1 
  5.95640363     0.06189237     0.00321152    -0.01431578 
```

```r
coef(fit_fp_1)$item |>
  rownames_to_column(var = "item") |>
  head()
```

```
  item (Intercept)    verb_t1     gramm1 verb_t1:gramm1
1    1    6.022184 0.06189237 0.00321152    -0.01431578
2    2    5.761268 0.06189237 0.00321152    -0.01431578
3    3    5.854873 0.06189237 0.00321152    -0.01431578
4    4    6.056862 0.06189237 0.00321152    -0.01431578
5    5    6.138213 0.06189237 0.00321152    -0.01431578
6    6    6.331058 0.06189237 0.00321152    -0.01431578
```

2.1 A short history of varying slopes

> A lot of people construct random intercept-only models but conceptually, it makes hella sense to include random slopes most of the time. After all, you can almost always expect that people differ with how they react to an experimental manipulation!

— Winter (2014), p. 17

- after Baayen et al. (2008), linguists who adopted mixed models typically used random-intercepts only models
  - but these have been shown time and again to drastically inflate Type I error rate (false positive) (e.g., Barr et al., 2013)
  - Barr et al. (2013) began the credo "keep it maximal", meaning include all random slopes justified by your design and existing theories
  - let's focus on adding just one varying slope for now

2.2 Random intercepts and slopes equation

- Equation 3 gives an example of a model with by-participant varying slopes for grammaticality

$$fp_i = \beta_0 + \alpha_{j[i]} + \alpha_{k[i]} + \beta_{verb\_t} x_i + (\beta_{gramm} + \gamma_{j[i]}) x_i + e_i \tag{3}$$

- we've changed $\beta_{grammt} x_i$ to $(\beta_{grammt} + \gamma_j[i]) x_i$
  - where $\gamma_{j[i]}$ is our by-participant varying slope for `gramm` for participant i
- imagine observation 163 comes from participant (j) 6, item (k) 38, which is a Future-grammatical condition
  - how could we plug these into the equation?

2.2.1 Visualising varying intercepts and slopes

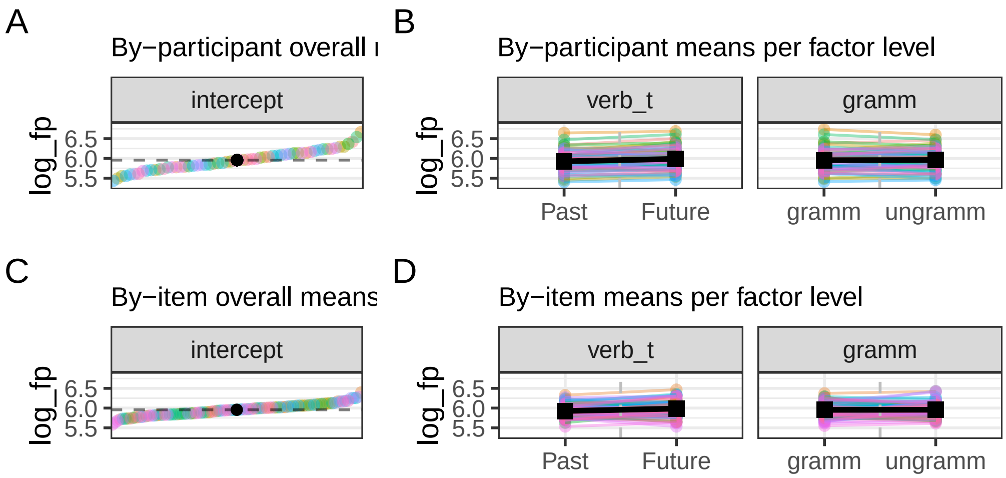

Figure 2: Mean effects by-participant overall (A) and per condition (B) with population-level effects in black, with the same plots by-item (C and D)

2.3 Comparing participant and item effects

- we've already noted that there's more variation between participants in the overall first-pass reading times
  - some tend to have higher, others lower, reading times
  - this is taken into consideration with the by-participant and -item varying intercepts
  - and we saw in our random effects parameters that the standard deviation for participant intercepts was larger
- today we will focus on varying slopes
  - there seems to be comparable inter-group slope variation in both participants and items

3 Random intercepts and slopes model

- random intercepts = taking group-level variance in effect direction/magnitude into account
  - i.e., some participants might have a stronger effect, weaker effect, or effect in the opposite direction compared to the population-level

3.1 Disclaimer!!!

> ⚠ Model building
>
> Today we are *exploring* the random effects of our model by adding and subtracting random slopes to 'see what happens'. You typically would NOT do this!
> Generally, you would start with a pre-defined random effects structure justified by your experimental design and theory (your "maximal" model (Barr et al., 2013)). We will get into model selection in the next (and last) session. Today we will be adding and removing varying slopes willy-nilly, which can amount to p-hacking, data dredging, or HARKing (Hypohtesisng After the Results are Known).

3.2 Random intercept-only model

- recall our random intercept-only model

```
fit_fp_1 <-
  lmer(log(fp) ~ verb_t*gramm +
         (1 |sj) +
         (1|item),
       data = df_biondo,
       subset = roi == 4)
```

- and inspect the random effects parameters

```
# an alternative to VarCorr(fit_fp_1):
summary(fit_fp_1)$varcor
```

```
Groups   Name        Std.Dev.
item     (Intercept) 0.13929
sj       (Intercept) 0.25795
Residual             0.40111
```

- what does this tell us?

3.3 Adding a slope

- let's look at by-item varying slopes for tense to start

```
fit_fp_item <-
  lmerTest::lmer(log(fp) ~ verb_t*gramm +
         (1 |sj) +
         (1 + verb_t|item),
       data = df_biondo,
       subset = roi == 4)
```

- we've just added `+ gramm` to `(1|sj)`
  - this reads as "*fit varying intercepts (`1`) per participant (`|sj`)...*"
  - "*...and by-item varying intercpets (`1`) and tense slopes (`+ verb_t`) per item (`|item`)*
    "

3.4 summary()

```
summary(fit_fp_item)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: log(fp) ~ verb_t * gramm + (1 | sj) + (1 + verb_t | item)
   Data: df_biondo
 Subset: roi == 4

REML criterion at convergence: 4216.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1758 -0.6096 -0.0227  0.6060  4.0568

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 item     (Intercept) 0.019424 0.13937
          verb_t1     0.002513 0.05012  0.54
 sj       (Intercept) 0.066414 0.25771
```

```
 Residual              0.160252 0.40032
Number of obs: 3795, groups:  item, 96; sj, 60

Fixed effects:
                  Estimate  Std. Error          df t value             Pr(>|t|)
(Intercept)       5.956384    0.036763   79.249351 162.023 < 0.0000000000000002
verb_t1           0.061733    0.013970   93.398427   4.419            0.0000267
gramm1            0.003298    0.012999 3544.431926   0.254                 0.80
verb_t1:gramm1   -0.014380    0.025998 3544.742546  -0.553                 0.58

(Intercept)    ***
verb_t1        ***
gramm1
verb_t1:gramm1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) vrb_t1 gramm1
verb_t1      0.077
gramm1       0.000 -0.002
vrb_t1:grm1  0.000  0.002  0.000
```

3.5 Fixed effects

Random intercept only

```
round(
  summary(fit_fp_1)$coefficients,
  5)
```

```
               Estimate Std. Error         df   t value Pr(>|t|)
(Intercept)     5.95640    0.03679   79.20081 161.90252  0.00000
verb_t1         0.06189    0.01303 3637.13315   4.75172  0.00000
gramm1          0.00321    0.01302 3637.18338   0.24657  0.80526
verb_t1:gramm1 -0.01432    0.02605 3637.10235  -0.54956  0.58265
```

Random intercept and slope

```
round(
  summary(fit_fp_item)$coefficients,
```

```r
    5)
```

```
                Estimate Std. Error         df   t value Pr(>|t|)
(Intercept)      5.95638    0.03676   79.24935 162.02259  0.00000
verb_t1          0.06173    0.01397   93.39843   4.41890  0.00003
gramm1           0.00330    0.01300 3544.43193   0.25367  0.79977
verb_t1:gramm1  -0.01438    0.02600 3544.74255  -0.55312  0.58021
```

- the uncertainty around the effect of tense in `fit_fp_item` has changed
  - slightly larger standard error
  - much fewer degrees of freedom
  - slightly smaller t-value value
  - slightly larger larger p-value

3.6 Random effects

```r
summary(fit_fp_1)$varcor # or VarCorr(fit_fp_1)
```

```
 Groups   Name        Std.Dev.
 item     (Intercept) 0.13929
 sj       (Intercept) 0.25795
 Residual             0.40111
```

```r
summary(fit_fp_item)$varcor
```

```
 Groups   Name        Std.Dev. Corr
 item     (Intercept) 0.139371
          verb_t1     0.050125 0.542
 sj       (Intercept) 0.257710
 Residual             0.400315
```

- variance components are qualitatively unchanged
- residual error is slightly lower
- but we have a new row under the `item` group: `verb_t1`
  - we see the standard deviation of by-participant varying slopes by tense (`0.05`)
  - and we see a new columns: `Corr`

3.6.1 Correlation paramater

- this is now what we call a variance-covariance matrix
  - but we only have one correlation term, that of by-item intercepts with by-item tense slopes
  - their correlation is `0.54`
    * this is a positive correlation, meaning the *higher* a participant's intercept (overall first-pass reading times), the stronger the effect of tense

3.6.2 Plotting

- to make life simple, let's use `lattice::dotplot()`: what do these plots tell us?

```
fig_item <- lattice::dotplot(ranef(fit_fp_item))$item
fig_sj <- lattice::dotplot(ranef(fit_fp_item))$sj

cowplot::plot_grid(fig_item, fig_sj, rel_widths = c(2,1), labels = c("A", "B"))
```

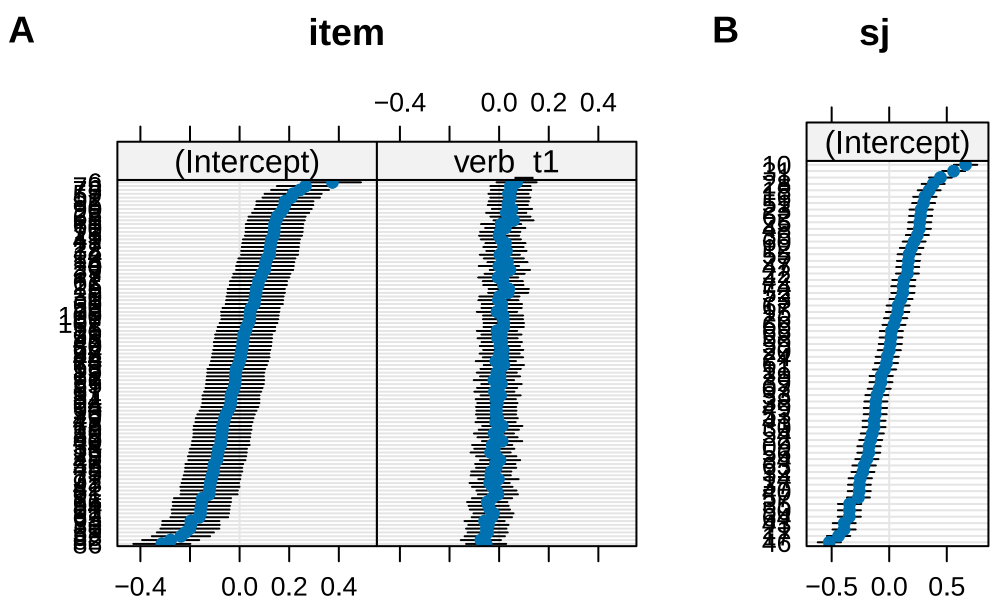

Figure 3: By-item varying intercepts and slopes (A), by-participant varying intercepts (B)

3.7 Correlation parameter

- we can plot this relationship by extracting the intercept and slope values with `coef()`

- or `ranef` to get their deviances from the population-level intercept/slope

```
coef(fit_fp_item)$item |>
  rownames_to_column(var = "item") |>
  rename(intercept = `(Intercept)`) |>
  # head()
  ggplot() +
  aes(x = verb_t1, y = intercept) +
  geom_point() +
  labs(
    title = "Correlation of slopes and intercepts"
  )
```

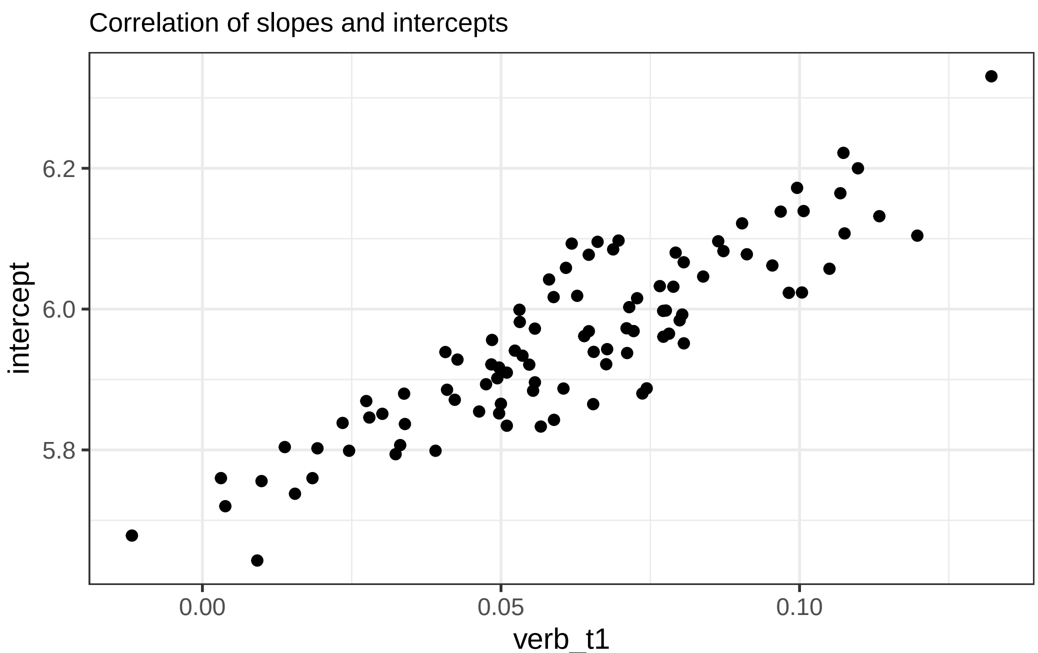

Figure 4: Correlation of by-participant gramm1 slopes (x-axis) and intercepts (y-axis)

- participants with higher intercepts had a stronger effect of grammaticality
  - with most participants estimated to have a positive effect

3.8 Model comparison

- does including by-participant slopes for adverb type improve our model fit?

```r
anova(fit_fp_1, fit_fp_item)
```

```
Data: df_biondo
Subset: roi == 4
Models:
fit_fp_1: log(fp) ~ verb_t * gramm + (1 | sj) + (1 | item)
fit_fp_item: log(fp) ~ verb_t * gramm + (1 | sj) + (1 + verb_t | item)
            npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
fit_fp_1       7 4210.3 4254.0 -2098.2   4196.3
fit_fp_item    9 4210.4 4266.5 -2096.2   4192.4 3.9796  2     0.1367
```

- not really
  - log likelihood is slightly higher ("smaller" negative number) for `fit_fp_item`
  - but $p > 0.05$
- recall from our plots that there seemed to be by-participant variance in the slopes
  - what if we add by-participant slopes?

4 Adding another slope

- here we've added `+ verb_t` to `(1|sj)`

```r
fit_fp_sj_item <-
  lmerTest::lmer(log(fp) ~ verb_t*gramm +
          (1 + verb_t|sj) +
          (1 + verb_t|item),
        data = df_biondo,
        subset = roi == 4)
```

```
boundary (singular) fit: see help('isSingular')
```

- and we get a message about singular fit

4.1 Singular fit

- `boundary (singular) fit: see help('isSingular')`
  - follow this advice: run `help('isSingular')` in the Console and see what you find

- you should *never* ignore such messages, nor report models with singular fit or convergence warnings!
  - let's explore the model to see what went wrong

4.2 Fixed effects

```
round(
  summary(fit_fp_item)$coefficients,
  5)
```

```
                Estimate Std. Error         df   t value Pr(>|t|)
(Intercept)      5.95638    0.03676   79.24935 162.02259  0.00000
verb_t1          0.06173    0.01397   93.39843   4.41890  0.00003
gramm1           0.00330    0.01300 3544.43193   0.25367  0.79977
verb_t1:gramm1 -0.01438    0.02600 3544.74255  -0.55312  0.58021
```

```
round(
  summary(fit_fp_sj_item)$coefficients,
  5)
```

```
                Estimate Std. Error         df   t value Pr(>|t|)
(Intercept)      5.95641    0.03676   79.17933 162.05485  0.00000
verb_t1          0.06173    0.01415   91.56777   4.36365  0.00003
gramm1           0.00329    0.01300 3544.45970   0.25349  0.79990
verb_t1:gramm1 -0.01434    0.02599 3544.77121  -0.55150  0.58133
```

- we see again that the effect of tense is slightly changed, with an increase in the uncertainty around the effect

4.3 Random effects

```
summary(fit_fp_item)$varcor # or VarCorr(fit_fp_1)
```

```
 Groups   Name        Std.Dev. Corr
 item     (Intercept) 0.139371
          verb_t1     0.050125 0.542
 sj       (Intercept) 0.257710
 Residual             0.400315
```

```r
summary(fit_fp_sj_item)$varcor
```

```
 Groups   Name        Std.Dev. Corr
 item     (Intercept) 0.139147
          verb_t1     0.050010 0.549
 sj       (Intercept) 0.257726
          verb_t1     0.017519 1.000
 Residual             0.400240
```

- we see that by-participant tense has a comparatively smaller variance than the other terms
  - and the correlation with by-item intercepts is 1
  - this is a red flat: 1 or -1 correlation terms are an indication of convergence failure

4.3.1 Plotting

- in Figure 5 we see by-participant varying tense slopes
  - but the confidence intervals are tiny and hard to see
  - and they constantly increase
  - this is because of the erroneous perfect correlation between them an the intercepts

```r
fig_item <- lattice::dotplot(ranef(fit_fp_sj_item))$item
fig_sj <- lattice::dotplot(ranef(fit_fp_sj_item))$sj

cowplot::plot_grid(fig_item, fig_sj, labels = c("A", "B"))
```

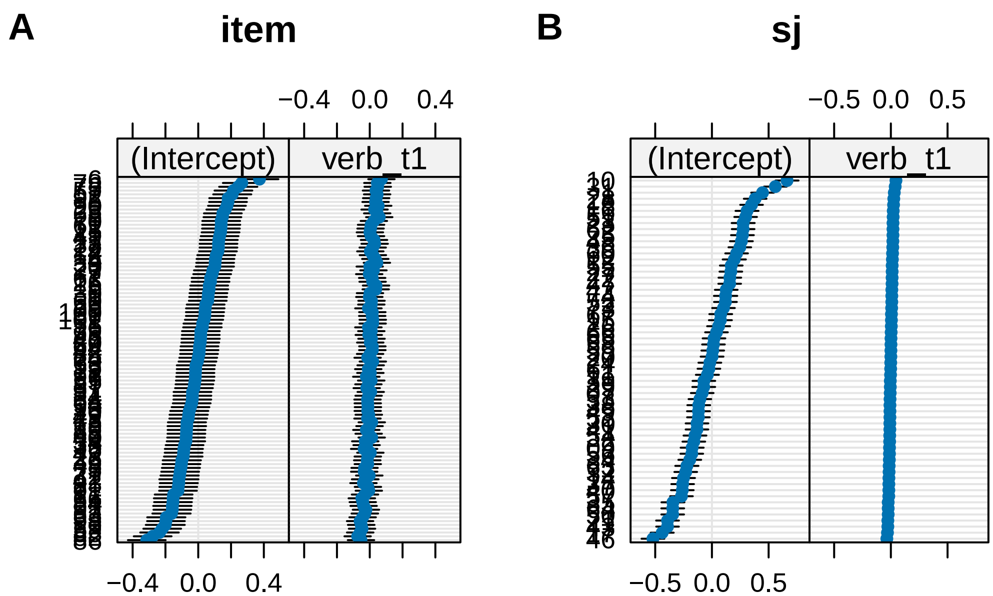

Figure 5: By-item (A) and by-participant (B) varying intercepts and slopes

4.4 Convergence warnings

- convergence warnings should not be ignored
  - they are a sign that a reliable line fit could not be found (this is an oversimplification…)
  - there can be many reasons for this:
    * impossible random effects structure (e.g., adding slopes that don't make sense)
    * sparse data
    * overfitting
- these are topics that we can address next week when discussion model selection

4.5 Dealing with convergence issues

- getting a convergence warning is an invitation to explore your random effects
  - a first step is to remove terms that are giving you Correlation terms +/-1
- so for now we would stick with `fit_fp_item`

```
# extract formula
formula(fit_fp_item)
```

```
log(fp) ~ verb_t * gramm + (1 | sj) + (1 + verb_t | item)
```

5 Reporting your model

- an example for this particular model:

  A linear-mixed model was fit to log-transformed first-pass reading times at the verb region with grammaticality, tense, and their interaction as fixed effects, and by-participant intercepts and by-item varying intercepts and tense slopes. Tense and grammaticality were sum contrast coded (past and grammatical = -0.5, future and ungrammatical = 0.5).

- however, we've made a grave misstep in coming to our final model
  - we did not start with a "maximal" model
- we'll talk about model selection and reduction next

Learning objectives

Today we learned…

- how to fit a random-intercepts and slopes model
- how to inspect and interpret random slopes

Important terms

Term	Definition	Equation/Code
linear mixed (effects) model	NA	NA

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. https://doi.org/10.1016/j.jml.2007.12.005

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, *68*(3), 255–278. https://doi.org/10.1016/j.jml.2012.11.001

Biondo, N., Soilemezidi, M., & Mancini, S. (2022). Yesterday is history, tomorrow is a mystery: An eye-tracking investigation of the processing of past and future time reference during sentence reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *48*(7), 1001–1018. https://doi.org/10.1037/xlm0001053

Sonderegger, M. (2023). *Regression Modeling for Linguistic Data.*

Winter, B. (2014). *A very basic tutorial for performing linear mixed effects analyses (Tutorial 2).*

Winter, B. (2019). Statistics for Linguists: An Introduction Using R. In *Statistics for Linguists: An Introduction Using R*. Routledge. https://doi.org/10.4324/9781315165547