Chapter 6

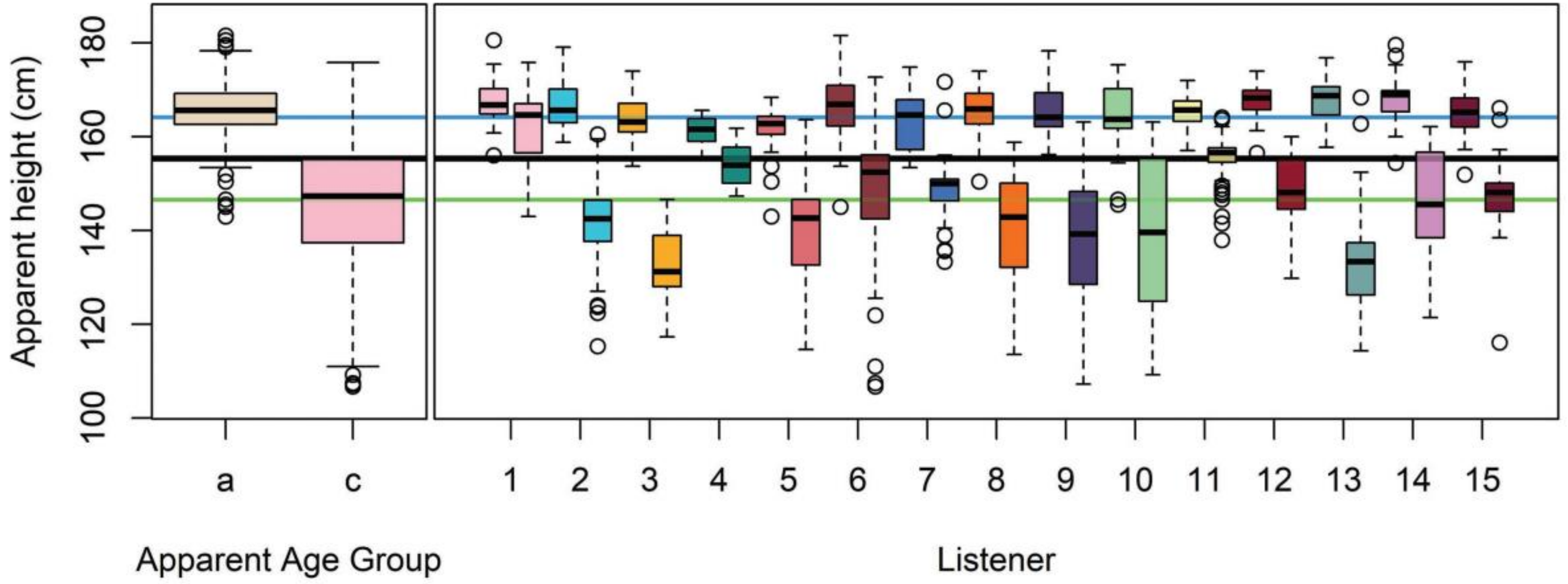
Lin 205C
Santiago Barreda

Chapter Precap

- Conditional effects, and marginal (average) 'main' effects.
- Crossing and nesting, and the importance of crossing for the estimation of interactions.
- Models with multiple group-dependent parameters ('random effects') and the interpretation of these as interactions.
- Introduce the multivariate normal distribution, in addition to the concepts of multidimensionality and correlation.
- The use of multivariate normal distributions in estimating random effects.
- Finally, an introduction and explanation of model comparison and selection.

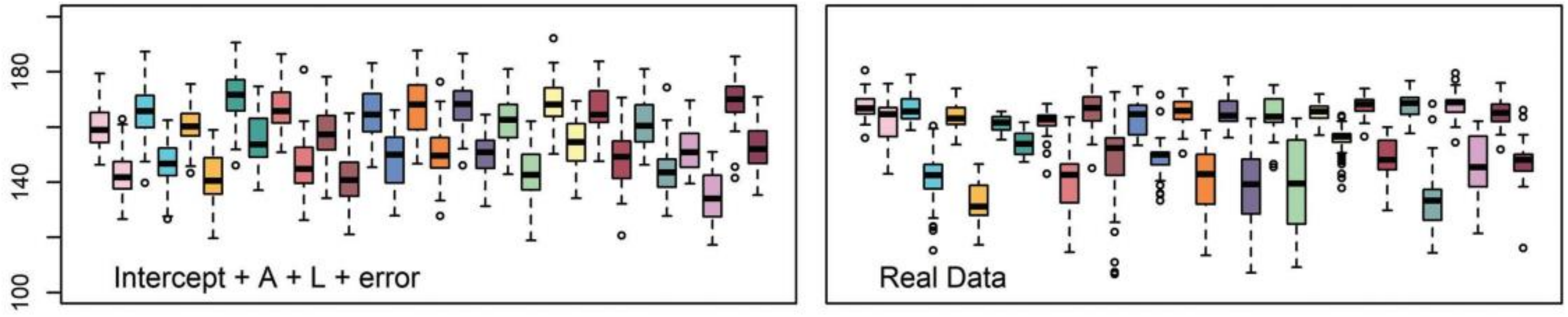
Listener Variation in the Age Effect

```
height ~ A + (1|L) + (1|S)
```



Listener Variation in the Age Effect

```
height ~ A + (1|L) + (1|S)
```



Data

```
# load packages and data
library (bmmb)
library (brms)
data (exp_data)

# exclude actual men and apparent men
notmen = exp_data[exp_data$C_v!='m' & exp_data$C!='m',]
```

The relevant variables in our data frame are:

- `L`: An integer from 1 to 15 indicating which *listener* responded to the trial.
- `height`: A floating-point number representing the *height* (in centimeters) reported for the speaker on each trial.
- `S`: An integer from 1 to 139 indicating which *speaker* produced the trial stimulus.
- `A`: The *apparent age* of the speaker indicated by the listener, `a` (adult) or `c` (child).

Research Questions

(Q1) How tall do speakers perceived as adult females sound?

(Q2) How tall do speakers perceived as children sound?

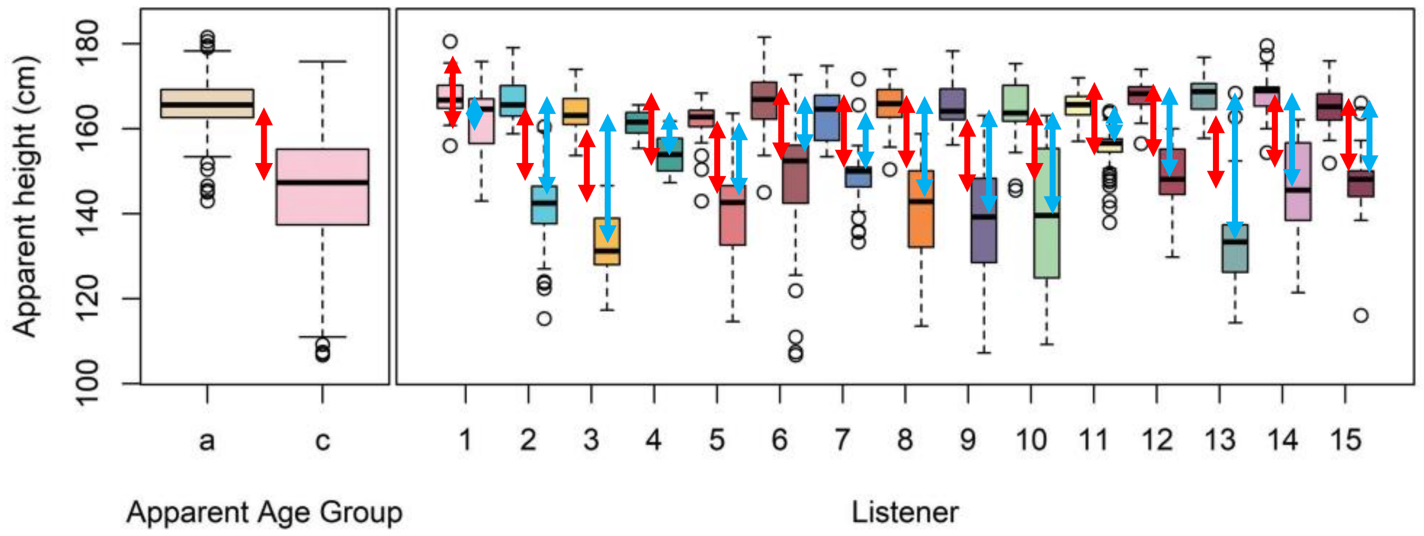
(Q3) What is the difference in apparent height associated with the perception of adultness?

Main and Interaction Effects

- Main effects are average, overall effects (e.g., P(A)).

- Interactions reflect the effect of some predictor, *conditional* on the value of another (e.g., P(A|L)).

- In many cases the marginal effects will not equal the effects of that predictor conditional on other variables (e.g., P(A)≠P(A|L)).



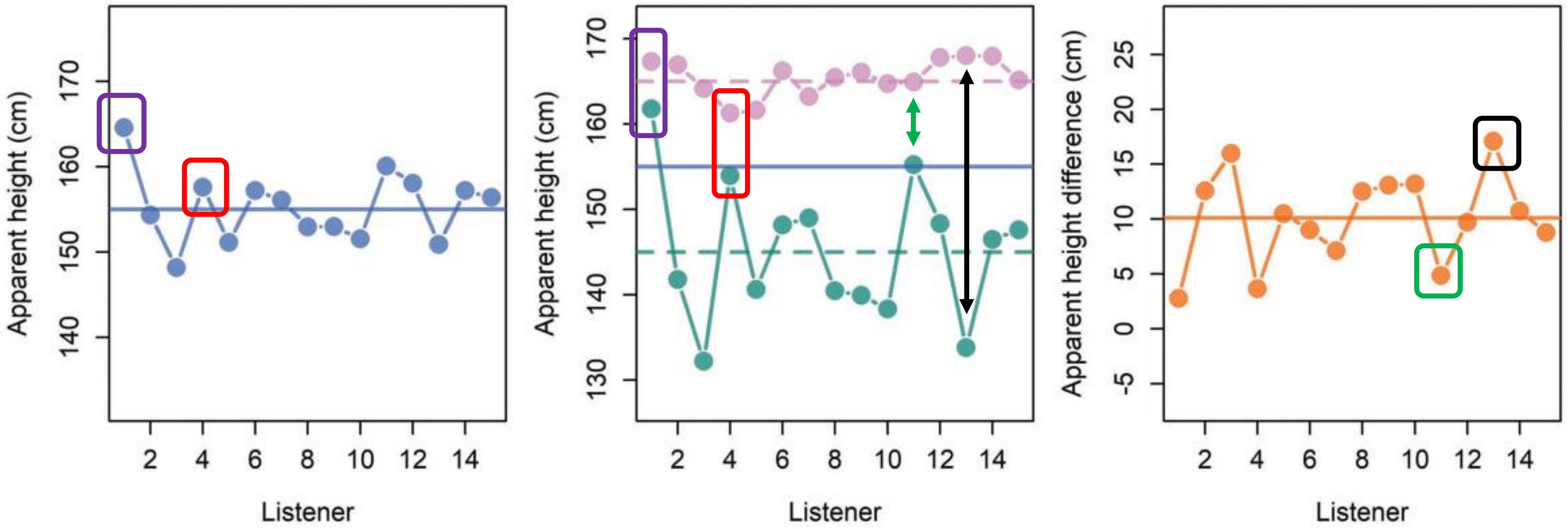
Main and Interaction Effects

- Main effects are denoted a name or letter, e.g. A, S, L.

- Interactions are denoted using a ‘:’, which can be read as ‘given’. So the following terms can be read like:

- ‘L’: the effect of listener.
- ‘A’: the effect of apparent age.
- ‘A:L’: the effect of apparent age given the level of listener.
- ‘L:A’: the effect of listeners given apparent age.

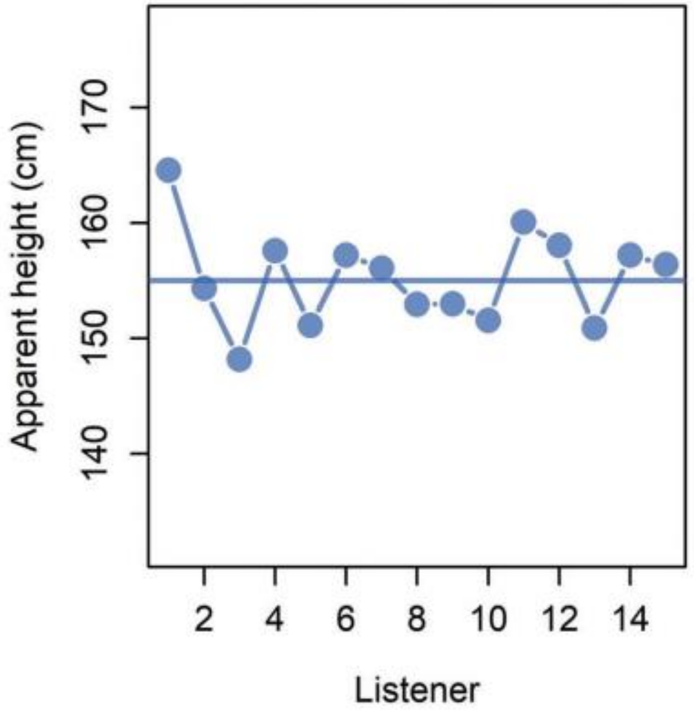
Interaction Effects

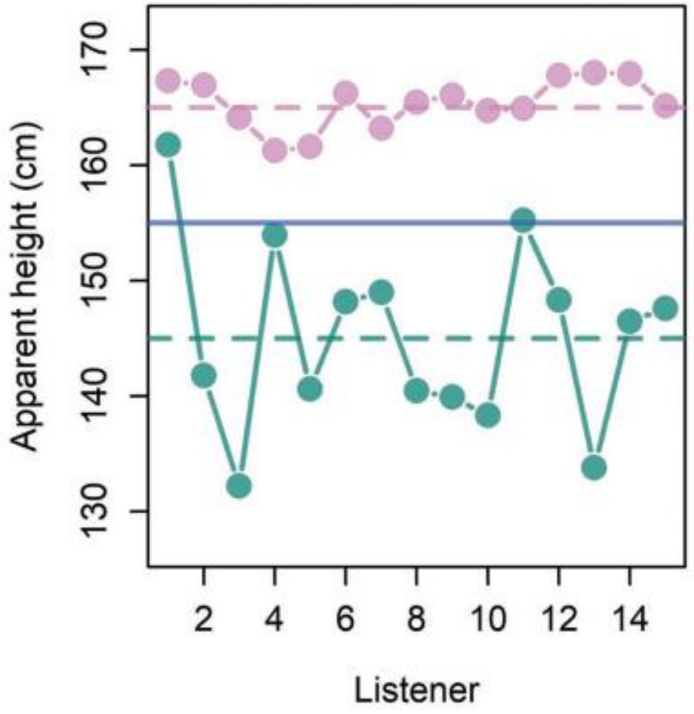


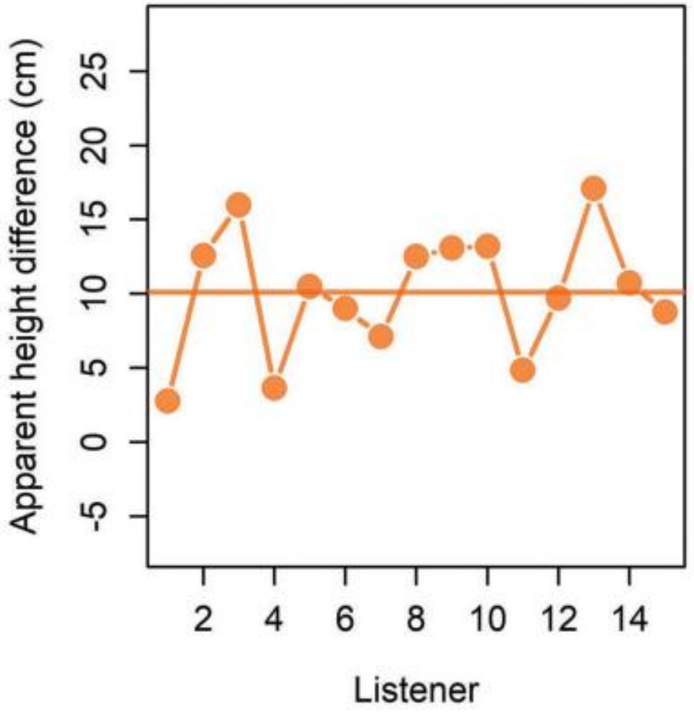
Interaction Effects

Where can you see?

- ‘L’: the effect of listener.
- ‘A’: the effect of apparent age.
- ‘A:L’: the effect of apparent age given the level of listener.
- ‘L:A’: the effect of listeners given apparent age.







Description of our Model

Last chapter $\rightarrow$

```
height ~ 1 + A + (1|L) + (1|S)
```

Let’s pretend it just had fixed effects $\rightarrow$

```
height ~ 1 + A + L + S
```

A:L tells our model to estimate listener-dependent age effects. $\rightarrow$

```
height ~ 1 + A + L + A:L + S
```

A*L = A + L + A:L $\rightarrow$

```
height ~ 1 + A * L + S
```

Description of our Model

‘fixed’ effects interactions.

```
height ~ 1 + A * L + S
height ~ 1 + A + L + A:L + S
```

‘random’ effects interactions.

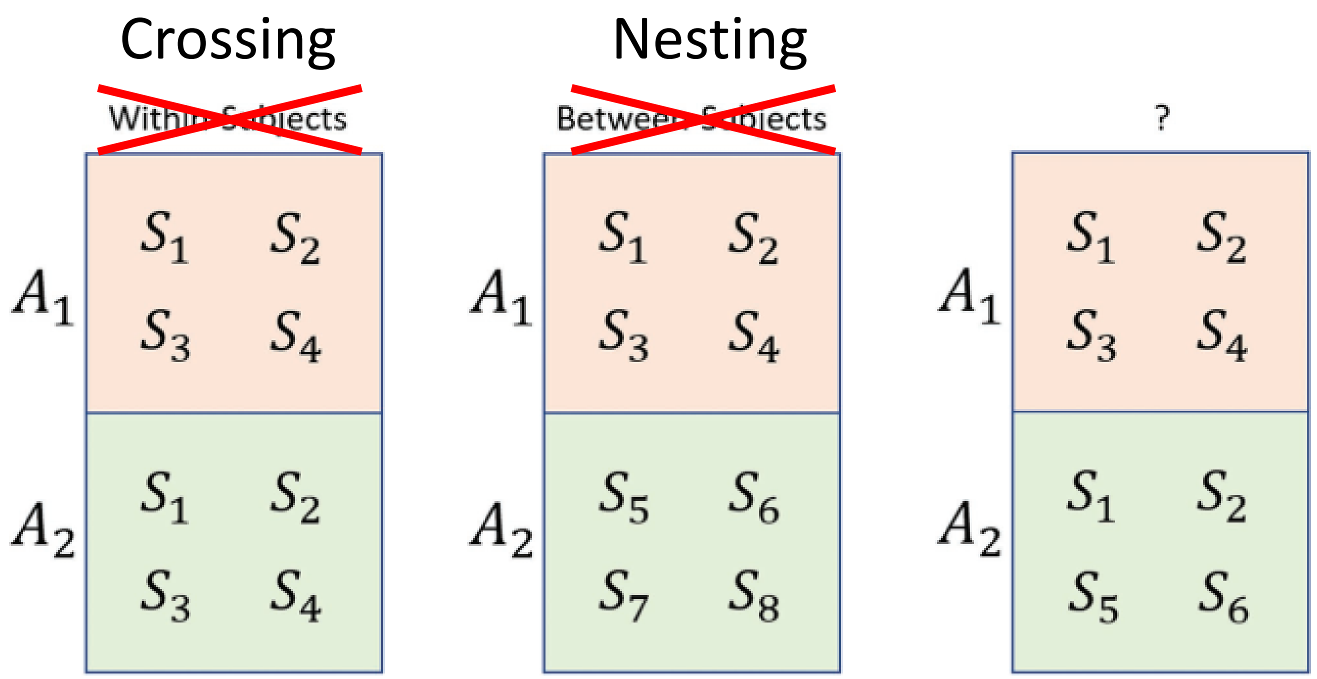
```
height ~ 1 + A + (1 + A|L) + (1|S)
```

Equivalent parameters

```
height ~ 1 + A +  L +  A:L      +  S
height ~ 1 + A + (1 +   A   |L) + (1 |S)
```

Crossing and Nesting

- Crossed: All factors of one factor appear at all levels of the other.
- Nested: Some levels of a factor only appear at some levels of the other.



Crossing and Nesting: Example

- If listeners each have one native language…….
- You can estimate the effect of native language on height judgments.

```
height ~ NatLang + (1|L)
```

- You cannot estimate the native language by listener interaction for most people.

```
height ~ NatLang + (NatLang|L)
height ~ NatLang + L + NatLang:L
```

Comparing Models

```
height ~ 1 + A + (1 + A|L) + (1|S)
```

$$\text{height}_{[i]} \sim \text{t}\left(\nu, \mu_{[i]}, \sigma\right)$$
$$\mu_{[i]} = \text{Intercept} + A + L_{\left[\text{L}_{[i]}\right]} + A:L_{\left[\text{L}_{[i]}\right]} + S_{\left[\text{S}_{[i]}\right]}$$

Priors :

$$L_{[\bullet]} \sim \text{N}\left(0, \sigma_L\right)$$
$$A:L_{[\bullet]} \sim \text{N}\left(0, \sigma_{A:L}\right)$$
$$S_{[\bullet]} \sim \text{N}\left(0, \sigma_S\right)$$

$$\text{Intercept} \sim \text{t}(3,156,12)$$
$$A \sim \text{t}(3,0,12)$$
$$\sigma_L, \sigma_S, \sigma_{A:L} \sim \text{t}(3,0,12)$$
$$\nu \sim \text{gamma}(2,0.1)$$
$$\sigma \sim \text{t}(3,0,12)$$

```
height ~ 1 + A + L + A:L + S
```

$$\text{height}_{[i]} \sim \text{t}\left(\nu, \mu_{[i]}, \sigma\right)$$
$$\mu_{[i]} = \text{Intercept} + A + L_{\left[\text{L}_{[i]}\right]} + A:L_{\left[\text{L}_{[i]}\right]} + S_{\left[\text{S}_{[i]}\right]}$$

Priors :

$$L_{[\bullet]} \sim \text{N}(0,12)$$
$$A:L_{[\bullet]} \sim \text{N}(0,12)$$
$$S_{[\bullet]} \sim \text{N}(0,12)$$

$$\text{Intercept} \sim \text{t}(3,156,12)$$
$$A \sim \text{t}(3,0,12)$$
$$\nu \sim \text{gamma}(2,0.1)$$
$$\sigma \sim \text{t}(3,0,12)$$

Correlation

$$\hat{r} = \frac{\hat{\sigma}_{xy}}{\hat{\sigma}_x \hat{\sigma}_y}$$

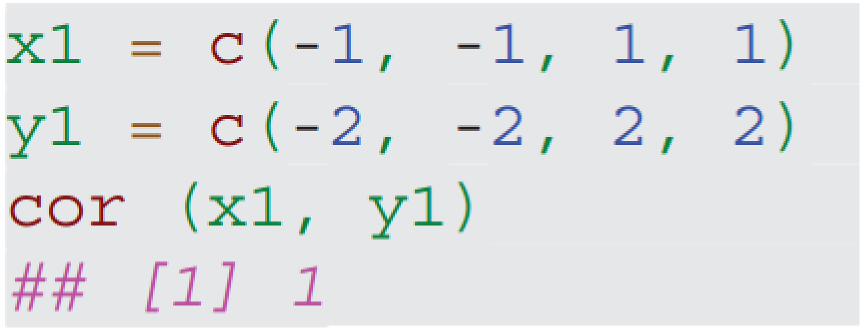
Pearson's correlation coefficient

$$\hat{r} = \frac{\Sigma\left(x_{[i]} - \bar{x}\right)\left(y_{[i]} - \bar{y}\right)}{(n-1) \cdot \hat{\sigma}_x \hat{\sigma}_y}$$

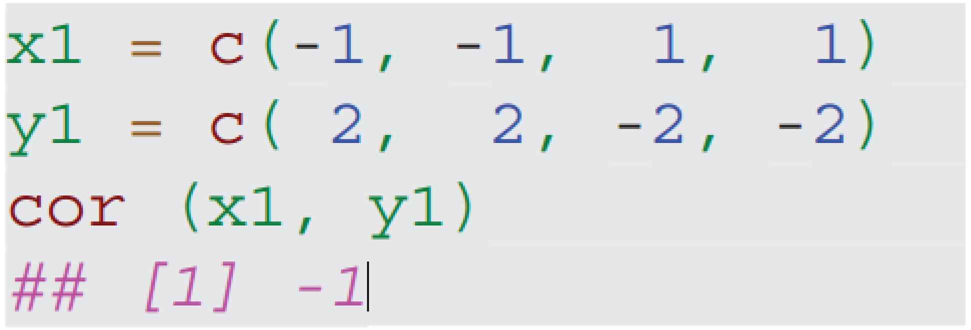
Covariance

$$\hat{\sigma}_{xy} = \frac{\Sigma\left(x_{[i]} - \bar{x}\right)\left(y_{[i]} - \bar{y}\right)}{(n-1)}$$

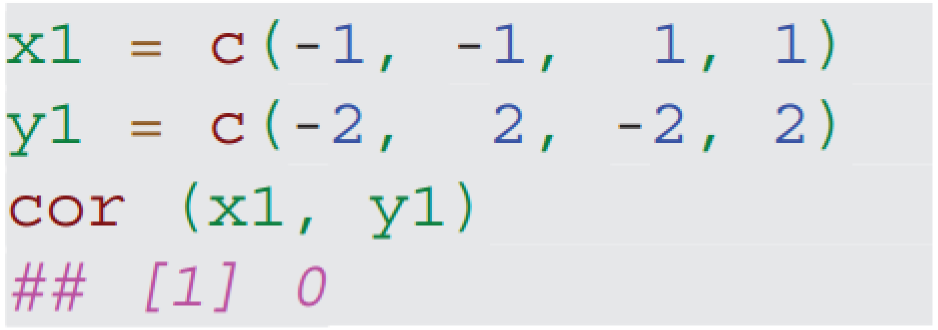
Correlation



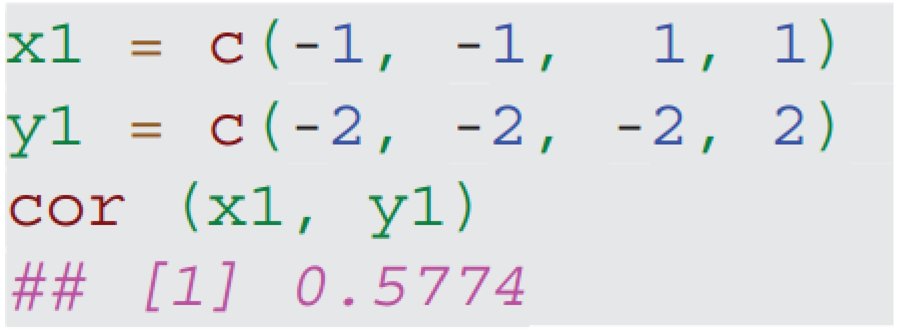
```
x1 = c(-1, -1, 1, 1)
y1 = c(-2, -2, 2, 2)
cor (x1, y1)
## [1] 1
```



```
x1 = c(-1, -1,  1,  1)
y1 = c( 2,  2, -2, -2)
cor (x1, y1)
## [1] -1
```

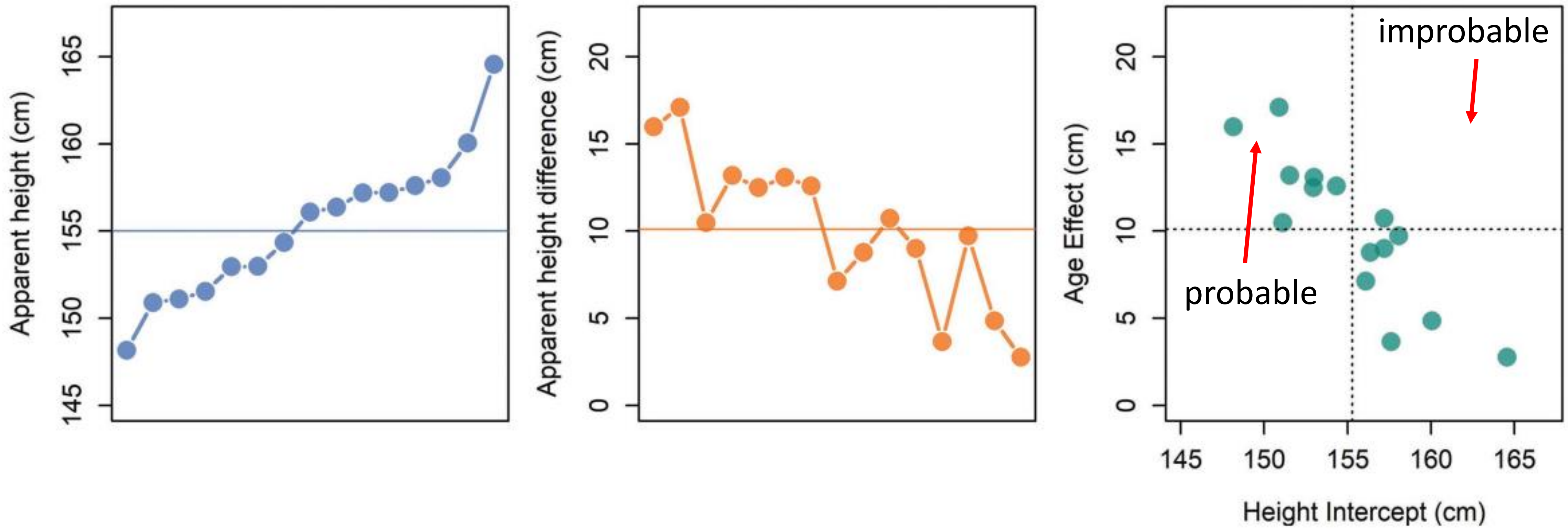


```
x1 = c(-1, -1,  1, 1)
y1 = c(-2,  2, -2, 2)
cor (x1, y1)
## [1] 0
```

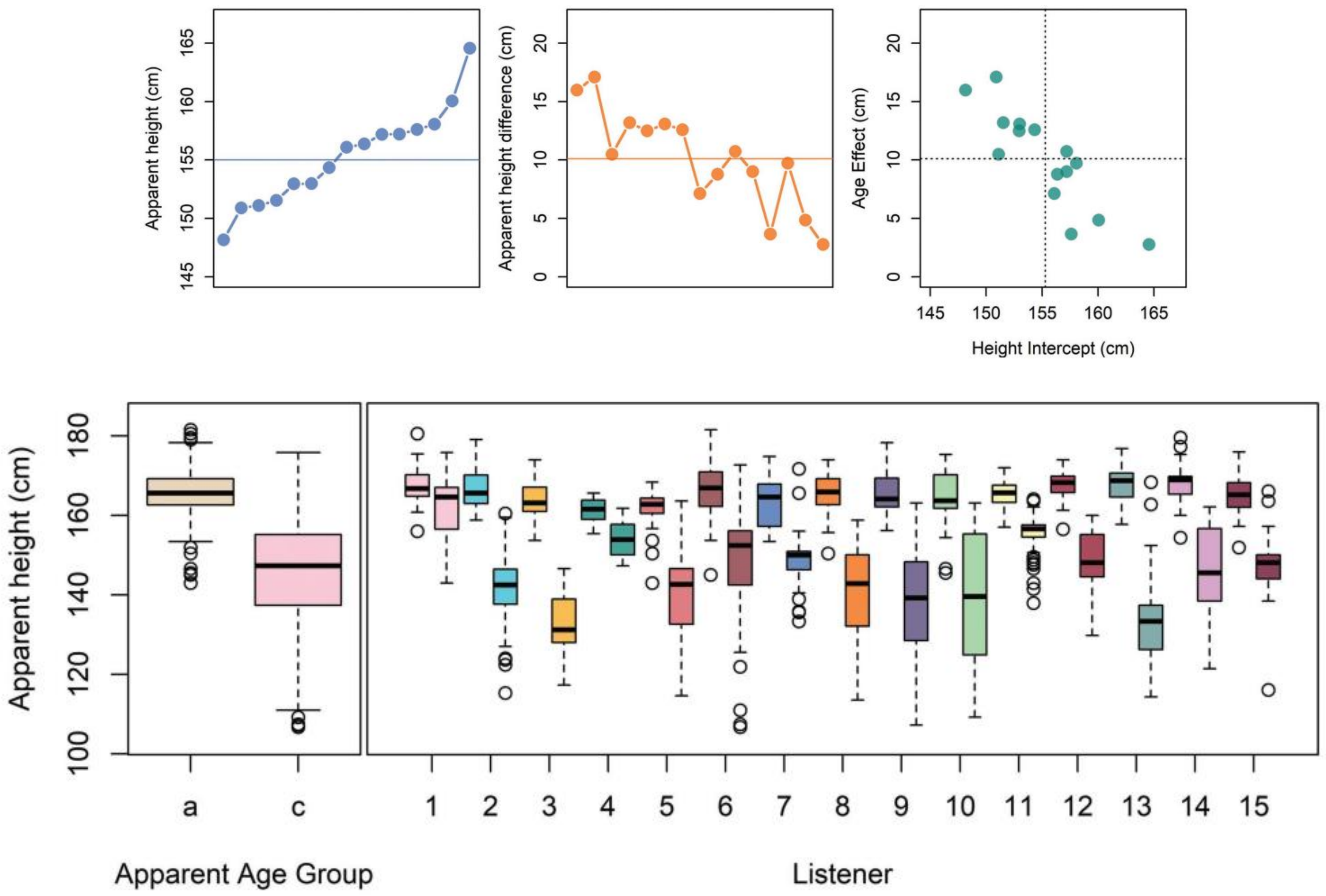


```
x1 = c(-1, -1,  1, 1)
y1 = c(-2, -2, -2, 2)
cor (x1, y1)
## [1] 0.5774
```

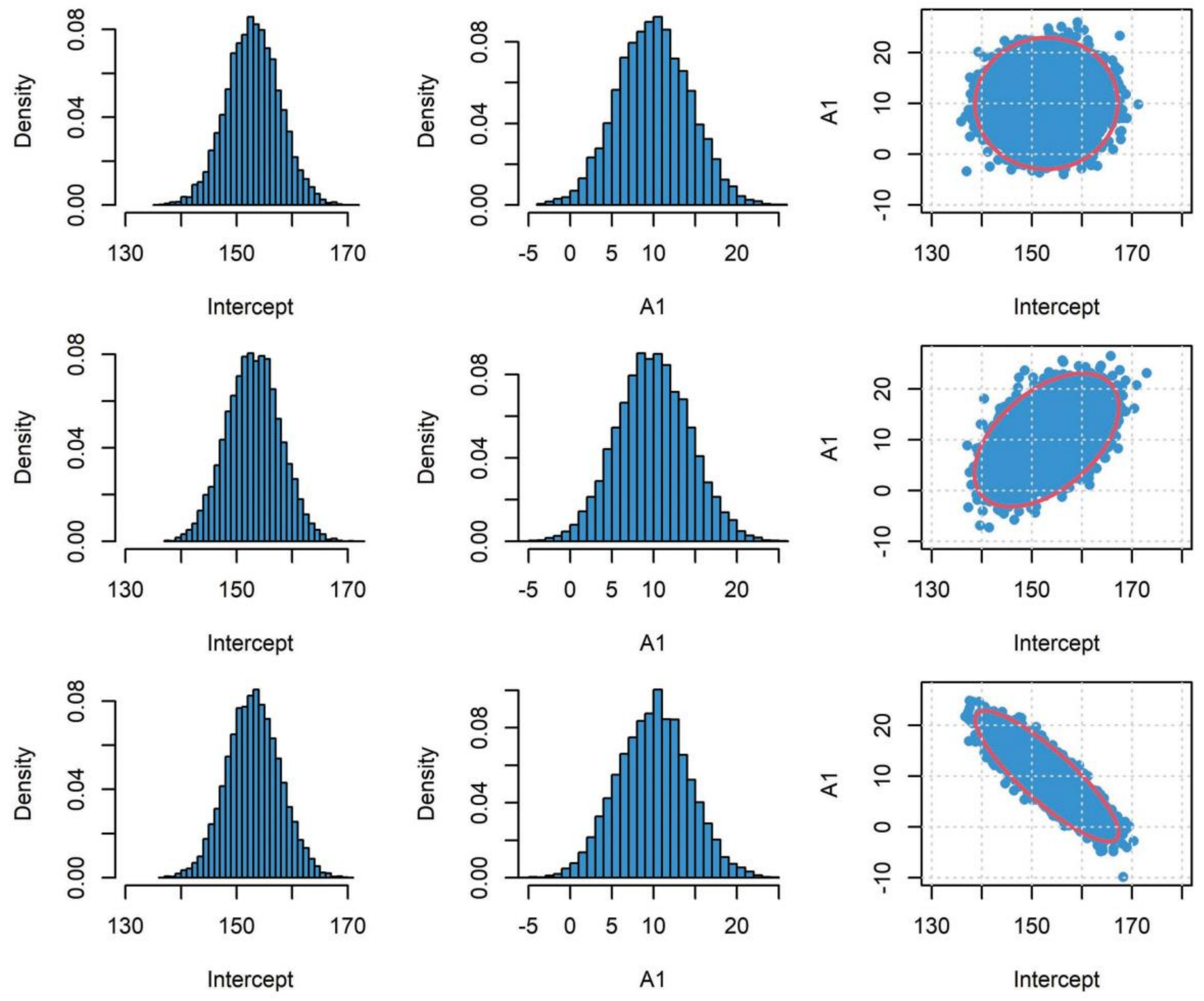
Correlated Random Effects



Correlated Random Effects



Multivariate Normal Distribution



$$\Sigma = \begin{bmatrix} \sigma_L^2 & \sigma_{L,A:L} \\ \sigma_{A:L,L} & \sigma_{A:L}^2 \end{bmatrix}$$

Priors for the Covariance Matrix

1 Specify priors for the standard deviation of each dimension. $\rightarrow$ $\sigma_L, \sigma_S, \sigma_{A:L} \sim \mathrm{t}(3,0,12)$

2 Specify a prior for the correlation matrix for the multivariate normal used for the random parameters. $\rightarrow$ $R \sim \mathrm{LKJCorr}(\eta)$

brms will draw correlation matrices

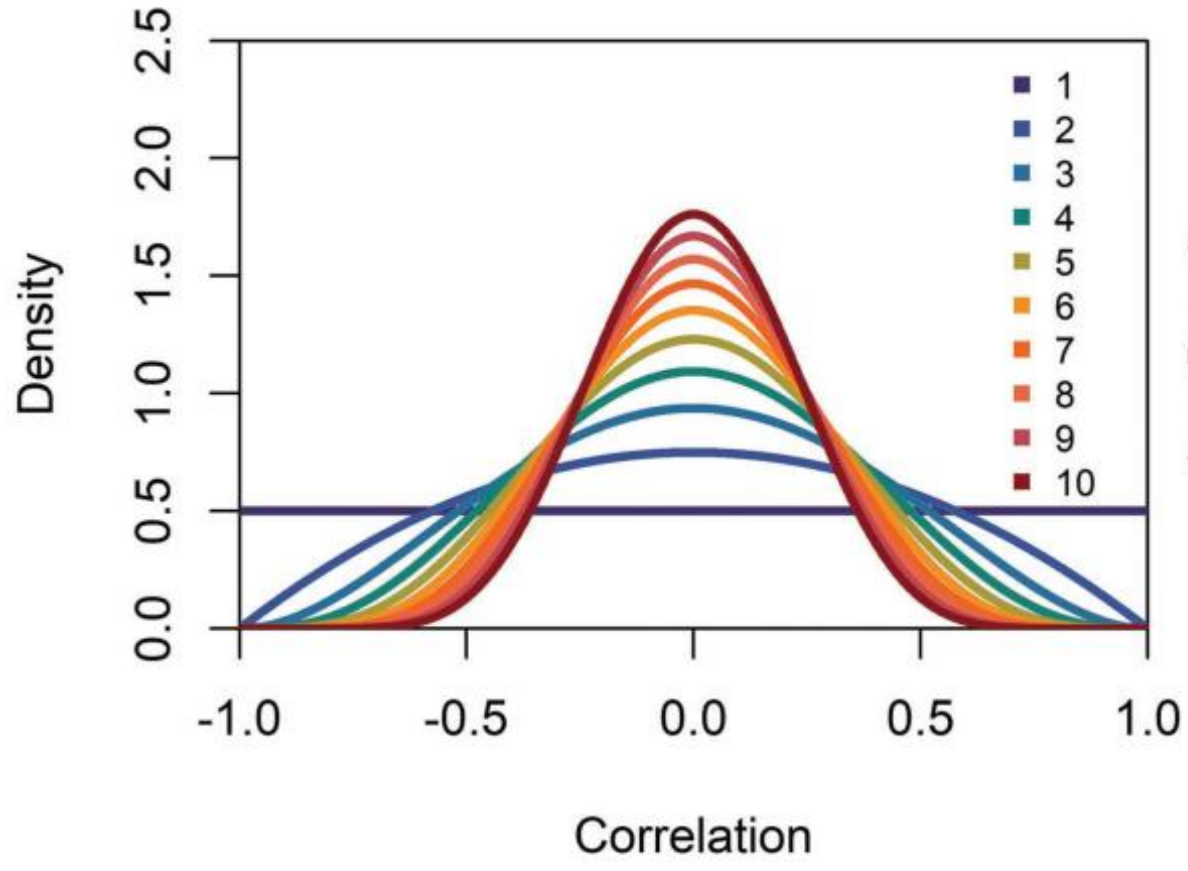
$$R = \begin{bmatrix} 1 & r_{L,A:L} \\ r_{A:L,L} & 1 \end{bmatrix}$$

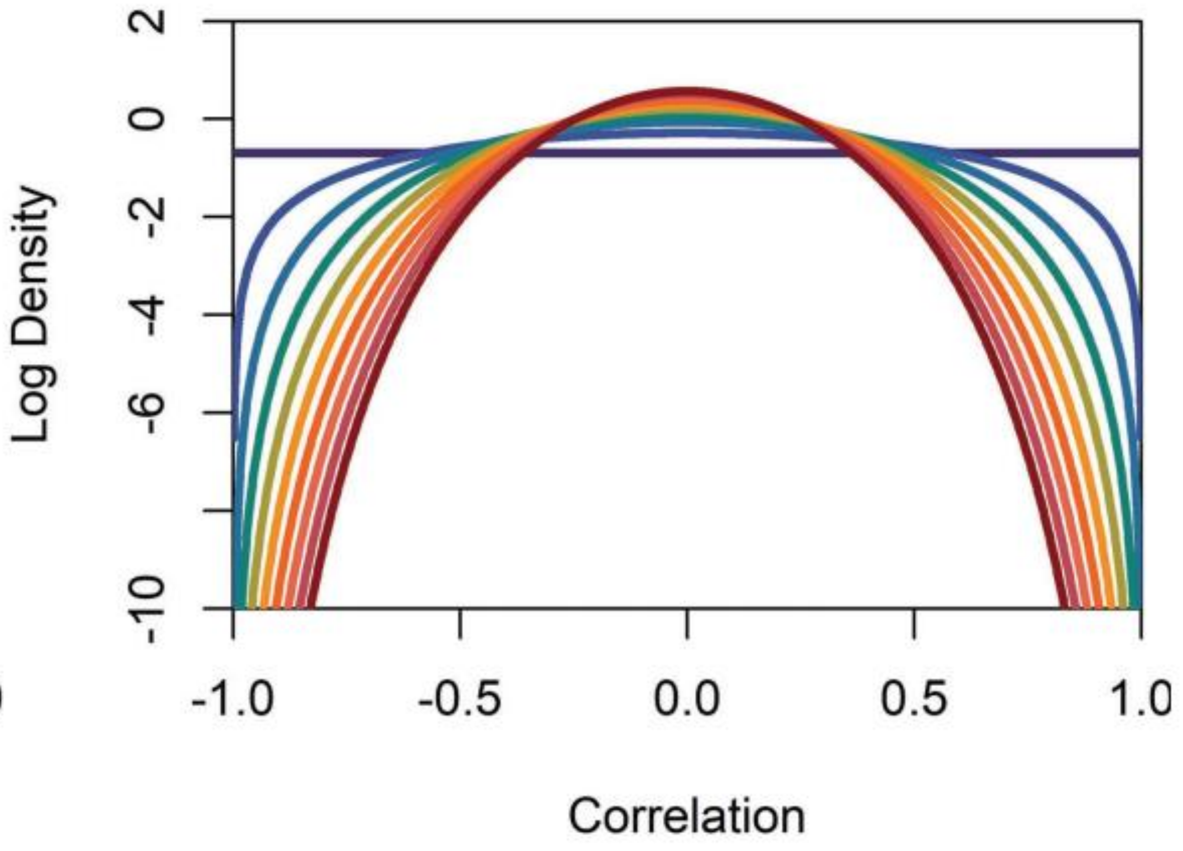
And ‘build’ the covariance matrix for you

$$\Sigma = \begin{bmatrix} \sigma_L^2 & \sigma_{L,A:L} \\ \sigma_{A:L,L} & \sigma_{A:L}^2 \end{bmatrix} = \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix} \cdot R \cdot \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix}$$

Lewandowski-Kurowicka-Joe (LKJ) Distribution

$$R \sim \mathrm{LKJCorr}(\eta)$$





Model Description

$$\text{height}_{[i]} \sim \text{t}\left(\nu, \mu_{[i]}, \sigma\right)$$
$$\mu_{[i]} = \text{Intercept} + A + L_{\left[\mathsf{L}_{[i]}\right]} + A:L_{\left[\mathsf{L}_{[i]}\right]} + S_{\left[\mathsf{S}_{[i]}\right]}$$

We are modeling apparent height as coming from a t distribution with unknown nu (ν), mean (μ), and scale (σ) parameters. The expected value for any given trial (μ) is modeled as the sum of an intercept, an effect for apparent age (A), a listener effect (L), a listener dependent effect for apparent age ($A:L$), and a speaker effect (S). The speaker effects were drawn from a univariate normal distribution with a standard deviation (σ_S) estimated from the data. The two listener effects were drawn from a bivariate normal distribution with standard deviations ($\sigma_L, \sigma_{A:L}$) and a correlation matrix (R) that was estimated from the data. The remainder of the 'fixed' effects and the bivariate correlation were given prior distributions appropriate for their expected range of values.

Priors:

$$S_{[\bullet]} \sim \text{N}\left(0, \sigma_S\right)$$
$$\begin{bmatrix} L_{[\bullet]} \\ A:L_{[\bullet]} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma\right)$$

$$\text{Intercept} \sim \text{t}(3,156,12)$$
$$A \sim \text{t}(3,0,12)$$
$$\sigma, \sigma_L, \sigma_{A:L}, \sigma_S \sim \text{t}(3,0,12)$$
$$\nu \sim \text{gamma}(2,0.1)$$
$$R \sim \text{LKJCorr}(2)$$

$$\Sigma = \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix} \cdot R \cdot \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix}$$

Comparing Models

$$\text{height}_{[i]} \sim \text{t}\left(\nu, \mu_{[i]}, \sigma\right)$$
$$\mu_{[i]} = \text{Intercept} + A + L_{\left[\mathsf{L}[i]\right]} + A:L_{\left[\mathsf{L}[i]\right]} + S_{\left[\mathsf{S}[i]\right]}$$

Priors :

$$L_{[\bullet]} \sim \text{N}(0, \sigma_L)$$
$$A:L_{[\bullet]} \sim \text{N}(0, \sigma_{A:L})$$
$$S_{[\bullet]} \sim \text{N}(0, \sigma_S)$$

$$\text{Intercept} \sim \text{t}(3, 156, 12)$$
$$A \sim \text{t}(3, 0, 12)$$
$$\sigma_L, \sigma_S, \sigma_{A:L} \sim \text{t}(3, 0, 12)$$
$$\nu \sim \text{gamma}(2, 0.1)$$
$$\sigma \sim \text{t}(3, 0, 12)$$

$$\text{height}_{[i]} \sim \text{t}\left(\nu, \mu_{[i]}, \sigma\right)$$
$$\mu_{[i]} = \text{Intercept} + A + L_{\left[\mathsf{L}[i]\right]} + A:L_{\left[\mathsf{L}[i]\right]} + S_{\left[\mathsf{S}[i]\right]}$$

Priors :

$$S_{[\bullet]} \sim \text{N}(0, \sigma_S)$$
$$\begin{bmatrix} L_{[\bullet]} \\ A:L_{[\bullet]} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma\right)$$

$$\text{Intercept} \sim \text{t}(3, 156, 12)$$
$$A \sim \text{t}(3, 0, 12)$$
$$\sigma, \sigma_L, \sigma_{A:L}, \sigma_S \sim \text{t}(3, 0, 12)$$
$$\nu \sim \text{gamma}(2, 0.1)$$
$$R \sim \text{LKJCorr}(2)$$

$$\Sigma = \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix} \cdot R \cdot \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix}$$

Fitting the Model

```
# Fit the model yourself
priors = c(brms::set_prior("student_t(3,156, 12)", class = "Intercept"),
     brms::set_prior("student_t(3,0, 12)", class = "b"),
     brms::set_prior("student_t(3,0, 12)", class = "sd"),
     brms::set_prior("lkj_corr_cholesky (2)", class = "cor"),
     brms::set_prior("gamma(2, 0.1)", class = "nu"),
     brms::set_prior("student_t(3,0, 12)", class = "sigma"))

model_re_t =
  brms::brm (height ~ A + (A|L) + (1|S), data = notmen, chains = 4,
             cores = 4, warmup = 1000, iter = 5000, thin = 4,
             prior = priors, family = "student")
```

$$\text{height}_{[i]} \sim \text{t}\left(\nu, \mu_{[i]}, \sigma\right)$$

$$\mu_{[i]} = \text{Intercept} + A + L_{\left[\mathsf{L}[i]\right]} + A : L_{\left[\mathsf{L}[i]\right]} + S_{\left[\mathsf{S}[i]\right]}$$

Priors :

$$S_{[\bullet]} \sim \text{N}(0, \sigma_S)$$

$$\begin{bmatrix} L_{[\bullet]} \\ A : L_{[\bullet]} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma\right)$$

$$\text{Intercept} \sim \text{t}(3,156,12)$$

$$A \sim \text{t}(3,0,12)$$

$$\sigma, \sigma_L, \sigma_{A:L}, \sigma_S \sim \text{t}(3,0,12)$$

$$\nu \sim \text{gamma}(2,0.1)$$

$$\rightarrow R \sim \text{LKJCorr}(2)$$

$$\Sigma = \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix} \cdot R \cdot \begin{bmatrix} \sigma_L & 0 \\ 0 & \sigma_{A:L} \end{bmatrix}$$

More stuff in the L Random Effect

- Now we get more than one listener random effects sd.
- We also get an estimate of the correlation between predictors.

```
## Group-Level Effects:
## ~L (Number of levels: 15)
##                      Estimate Est.Error l-95% CI u-95% CI
## sd(Intercept)            4.34      0.84     3.05     6.29
## sd(A1)                   4.26      0.85     2.96     6.25
## cor(Intercept,A1)       -0.80      0.12    -0.95    -0.51
```

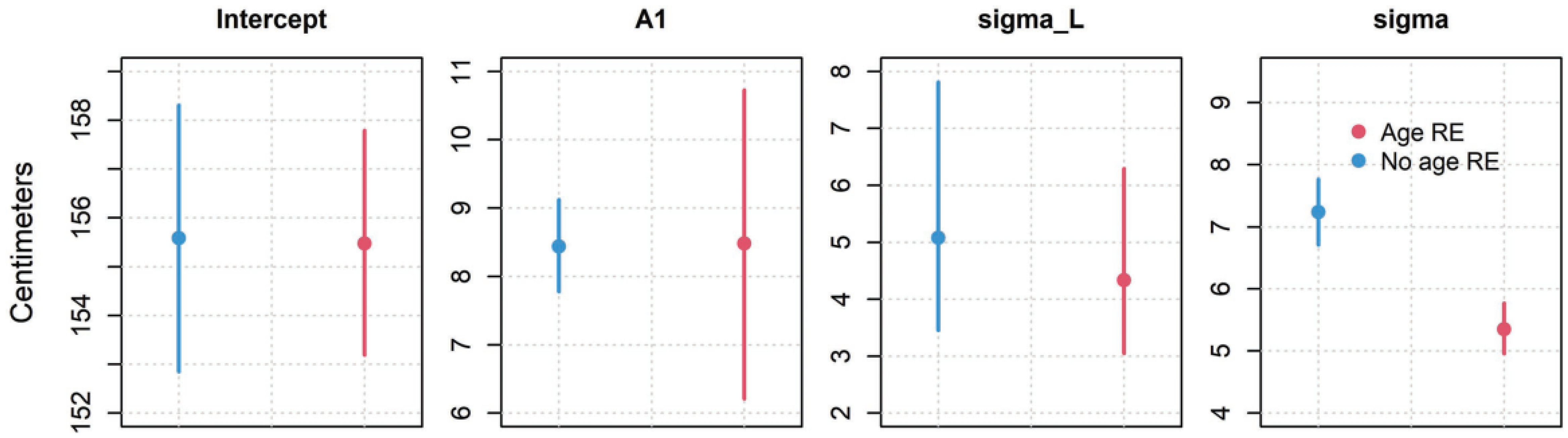
Helper Functions

```
bmmb::get_sds (model_re_t)
##         Estimate Est.Error  Q2.5 Q97.5 group
## Int.L      4.338    0.8356 3.050 6.289     L
## A1         4.258    0.8463 2.964 6.248     L
## Int.S      3.057    0.3594 2.398 3.808     S
## sigma      5.354    0.2057 4.958 5.770 sigma
```

```
# specify that we want the correlations for L
bmmb::getcorrs (model_re_t, factor="L")
##           Estimate Est.Error    Q2.5   Q97.5
## A1, Int.   -0.8018    0.1151 -0.9491 -0.5087
```

Consequences of Random Slopes

```
height ~ A + (A|L) + (1|S)
height ~ A + (1|L) + (1|S)
```



Model Comparison

- You can conceivably fit several different models to any data set.
- How do you know which one you should use?
- A particular problem is that more complicated models always fit data at least as well as less-complicated models.

Bias Variance Tradeoff

- If you sell clothing, you can sell clothes that fit everyone very well but then you need a lot of different models.
  - Excellent fit but many different models = low bias high variance.

- You can also just sell a small number of models that fit everyone pretty well but almost no one perfectly.
  - Worse fit but few different models = high bias low variance.

In and out of sample data

- In sample data: The data you have, e.g., y
- Out-of-sample data: The data you don't have, but that is just like the data you do have, e.g., $\tilde{y}$
- In general, you only care about y because of what it tells you about $\tilde{y}$.
- Overfitting: When your model is so good at predicting y that it is bad at predicting $\tilde{y}$.

In and out of sample prediction

- Log (pointwise) predictive density, lpd: the log density of your data given your model.

$$\widehat{\text{lpd}} = \sum_{i=1}^{N} \log\left(p\left(y_{[i]} \mid \theta\right)\right)$$

- Expected log (pointwise) predictive density, elpd: the expected value of the log density of the data you don’t have, given your model.

$$\text{elpd} = \sum_{i=1}^{N} \mathbb{E}\left(\log\left(p\left(\tilde{y}_{[i]} \mid \theta\right)\right)\right)$$

Simulating Prediction

```
n = 50          # how many observations
iter = 1000     # how many simulations

# these will hold the model log likelihoods for each iteration
lpd_hat = matrix (0, iter, 3)
elpd_hat = matrix (0, iter, 3)

set.seed(1)
for (i in 1:iter){
  # create 3 random predictors
  x1 = sample (c(-1,1), n, replace=TRUE)
  x2 = sample (c(-1,1), n, replace=TRUE)
  x3 = sample (c(-1,1), n, replace=TRUE)

  # generate the observed (in sample) data with an
  # underlying process that only uses the x1 predictor
  y = 1 + x1 + rnorm (n, 0, 1)
  # use the same process to simulate some "out-of-sample" data
  y_tilde = 1 + x1 + rnorm (n, 0, 1)

  for (j in 1:3){
    # fit three models, the first using the real underlying model
    if (j==1) mod = lm (y ~ 1+x1)
    # the next two include random useless predictors
    if (j==2) mod = lm (y ~ 1+x1 + x2)
    if (j==3) mod = lm (y ~ 1+x1 + x2 + x3)
```

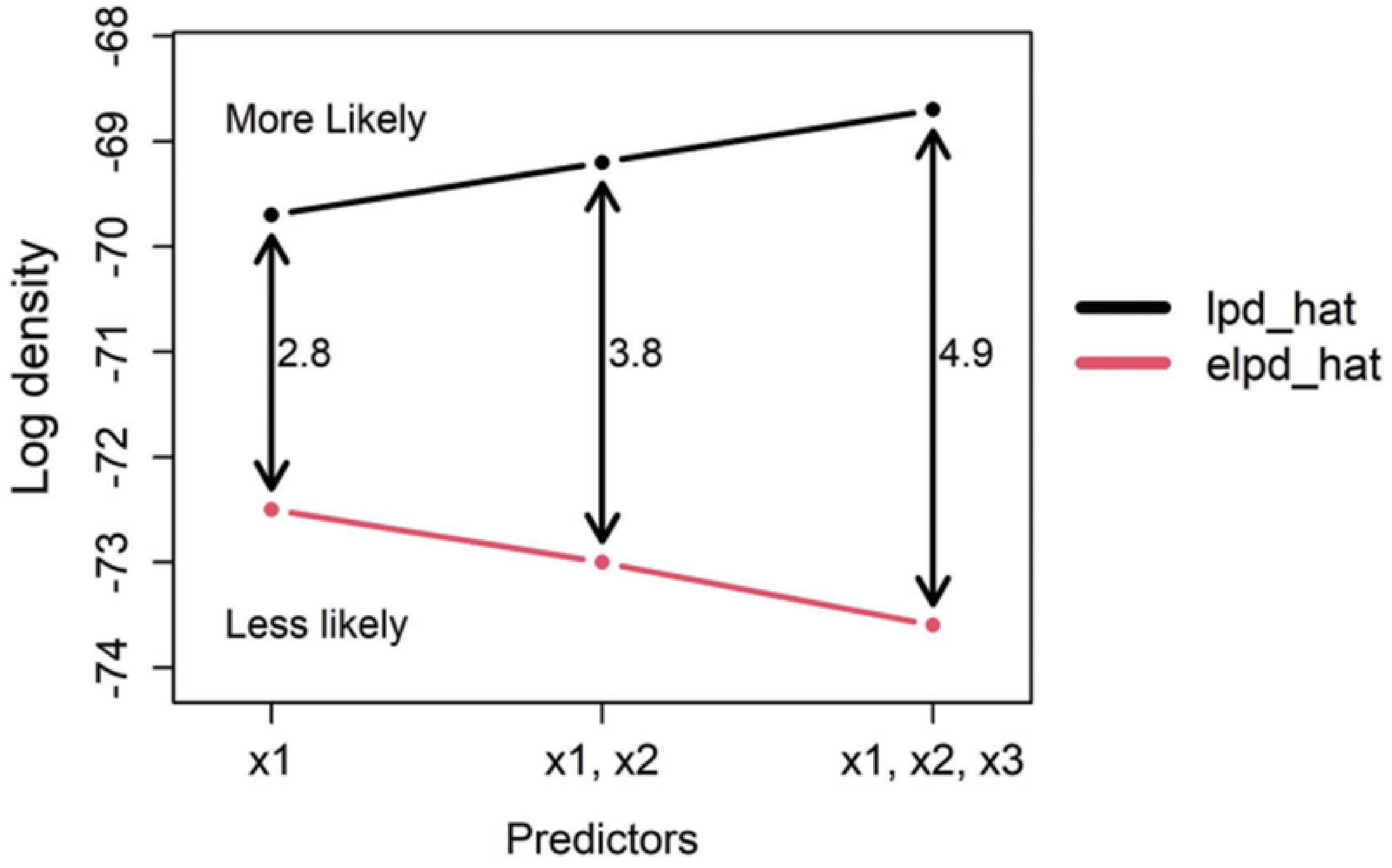
```
    # find the predicted value (mu) for each data point
    mu = mod$fitted.values
    # and the estimated sigma parameter
    sigma = summary(mod)$sigma

    # equivalent to equation 6.10
    lpd_hat[i,j] = sum (dnorm (y, mu, sigma, log = TRUE))
    # equivalent to equation 6.11
    elpd_hat[i,j] = sum (dnorm (y_tilde, mu, sigma, log = TRUE))
  }
}
```

Simulating Prediction

$$\widehat{\text{lpd}} = \sum_{i=1}^{N} \log\Big(\text{N}\big(y_{[i]} \mid \mu, \sigma\big)\Big)$$

$$\widehat{\text{elpd}} = \sum_{i=1}^{N} \log\Big(\text{N}\big(\tilde{y}_{[i]} \mid \mu, \sigma\big)\Big)$$



Penalizing Prediction

- More complex models always improve lpd.

$$\widehat{\text{lpd}} = \sum_{i=1}^{N} \log\left(p\left(y_{[i]} \mid \theta\right)\right)$$

- We can estimate elpd by *penalizing* lpd according to model complexity.

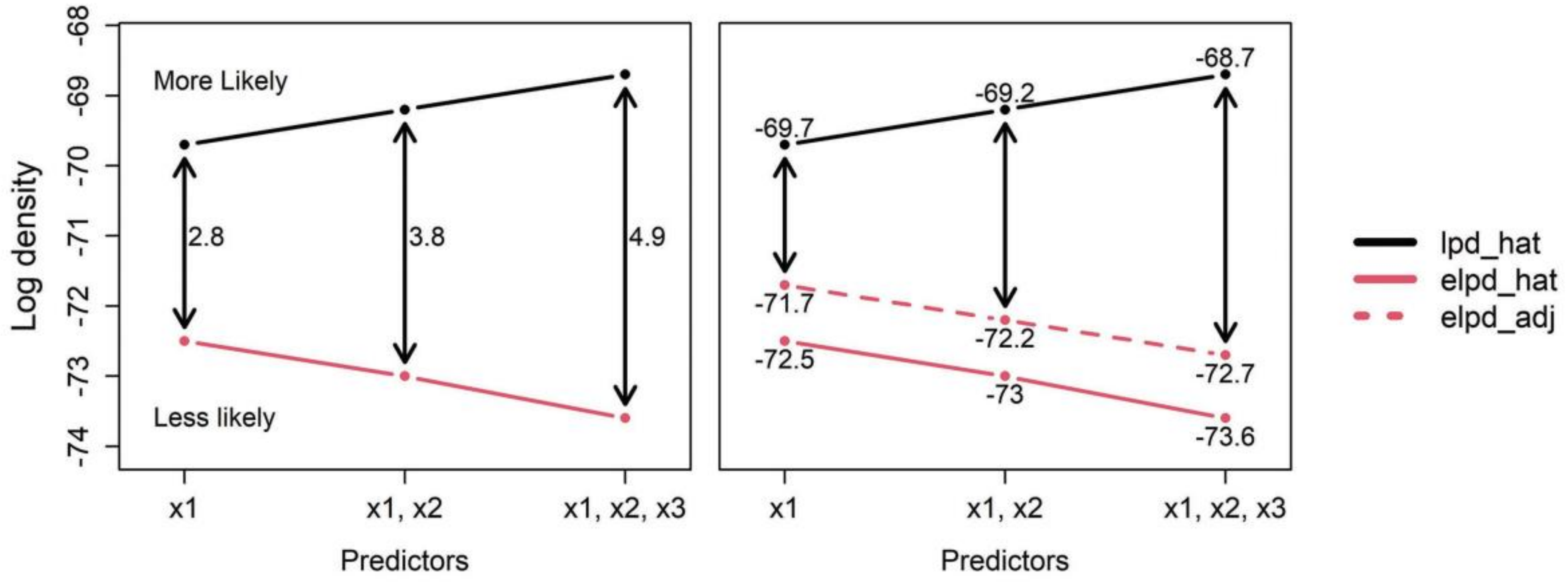
$$\text{elpd} = \sum_{i=1}^{N} \mathbb{E}\left(\log\left(p\left(\tilde{y}_{[i]} \mid \theta\right)\right)\right)$$

- We can call the penalty *p*.

$$\Rightarrow \quad \widehat{\text{elpd}} = \widehat{\text{lpd}} - \text{p}$$

Simulating Prediction

$$\widehat{\text{lpd}} = \sum_{i=1}^{N} \log\left(\text{N}\left(y_{[i]} \mid \mu, \sigma\right)\right) \qquad \widehat{\text{elpd}} = \sum_{i=1}^{N} \log\left(\text{N}\left(\tilde{y}_{[i]} \mid \mu, \sigma\right)\right) \qquad \widehat{\text{elpd}} = \widehat{\text{lpd}} - \text{p}$$



Cross Validation

- You usually can’t have out-of-sample data.
- You can replicate out-of-sample data by:
    1. Withholding some of your data when fitting your model.
    2. Predicting the withheld data using your model.
- This is called *cross-validation*.

Cross Validation

- The data you use for your model is your training data. Your withheld data is your testing data.
- There are different ways to cross-validate:
  - Holdout: Make one partition for testing and training.
  - K-fold: Split data into k parts. Each of the k parts gets its turn being the holdout.
  - Leave-one-out: For data with n points, each point gets its turn being the holdout.

Leave-one-out (LOO)

- We won’t actually fit n models, we can estimate out-of-sample prediction using LOO with the loo package in R.

- This will tell us about the ‘best’ model controlling for model complexity.

- This approach is extremely robust and can compare models that differ in many ways.

Leave-one-out (LOO)

$$\widehat{\mathrm{elpd}}_{LOO} \approx \sum_{i=1}^{n} \log\left(p\left(y_{[i]} \mid \theta_{y_{[-i]}}\right)\right)$$

$$\widehat{\mathrm{elpd}}_{LOO} = \widehat{\mathrm{lpd}} - \mathrm{p}_{\mathrm{LOO}}$$

$$\mathrm{p}_{\mathrm{LOO}} = \widehat{\mathrm{lpd}} - \widehat{\mathrm{elpd}}_{LOO}$$

Adding the LOO Criterion

```
model_sum_coding =
  brms::add_criterion (model_sum_coding, criterion="loo")
```

```
model_sum_coding$criteria$loo
##
## Computed from 5000 by 1401 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -5042.7 32.5
## p_loo        82.2  3.8
## looic     10085.4 65.1
## ------
## Monte Carlo SE of elpd_loo is 0.2.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

elpd estimate

effective number of parameters

AIC clone (-2* elpd)

Effective Number of Parameters

- In very simple models, the p term is related to the number of independent model parameters.

- The number of ‘independent’ parameters is not so obvious for multilevel models.

- The amount of flexibility can be thought of as the ‘effective number of parameters’.

Effective Number of Parameters

```
# Actual and effective number of parameters
# for simplest model
ncol(bmmb::get_samples(model_sum_coding))-2
## [1] 114
sum(model_loo_info[,'p_loo'])
## [1] 82.17

# for t model
ncol(bmmb::get_samples(model_sum_coding_t))-2
## [1] 115
sum(model_t_loo_info[,'p_loo'])
## [1] 85.23

# for 'random effects' model
ncol(bmmb::get_samples(model_re_t))-2
## [1] 132
sum(model_re_t_lo_info[,'p_loo'])
## [1] 109.1
```

Model Comparison

```
brms::loo_compare (model_sum_coding,
                   model_sum_coding_t,
                   model_re_t, criterion = "loo")
##                    elpd_diff se_diff
## model_re_t            0.0       0.0
## model_sum_coding_t -204.4      19.2
## model_sum_coding   -224.1      19.6
```

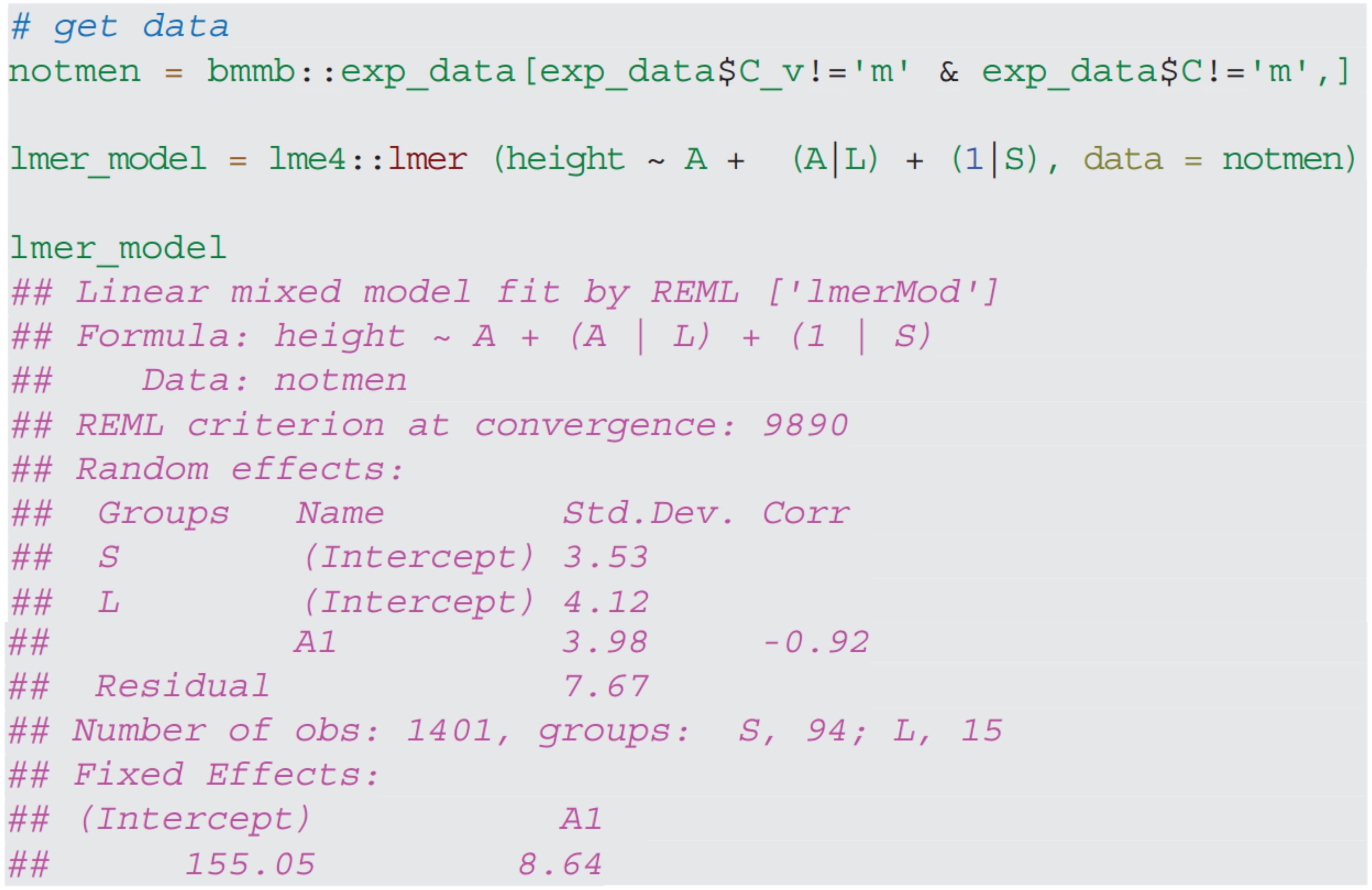
```
brms::loo_compare (model_sum_coding, model_sum_coding_t,
criterion = "loo")
##                    elpd_diff se_diff
## model_sum_coding_t   0.0       0.0
## model_sum_coding   -19.6       7.6
```

Adding a Useless Parameter

```
brms::fixef(model_re_t_tooBig)
##            Estimate Est.Error     Q2.5    Q97.5
## Intercept  155.4454    1.1583 153.1882 157.7513
## A1           8.5267    1.1226   6.2757  10.7094
## useless      0.3362    0.2522  -0.1688   0.8357
```

```
brms::loo_compare (model_re_t, model_re_t_tooBig)
##                   elpd_diff se_diff
## model_re_t_tooBig  0.0       0.0
## model_re_t        -0.7       2.8
```

Traditionalist’s Corner: lmer



```
# get data
notmen = bmmb::exp_data[exp_data$C_v!='m' & exp_data$C!='m',]

lmer_model = lme4::lmer (height ~ A +  (A|L) + (1|S), data = notmen)

lmer_model
## Linear mixed model fit by REML ['lmerMod']
## Formula: height ~ A + (A | L) + (1 | S)
##    Data: notmen
## REML criterion at convergence: 9890
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  S        (Intercept) 3.53
##  L        (Intercept) 4.12
##           A1          3.98     -0.92
##  Residual             7.67
## Number of obs: 1401, groups:  S, 94; L, 15
## Fixed Effects:
## (Intercept)          A1
##      155.05        8.64
```

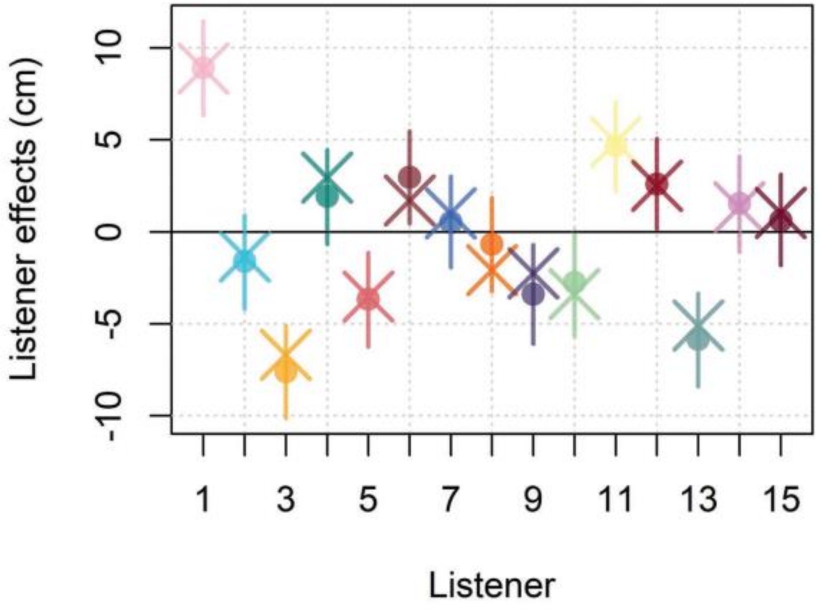
Traditionalist's Corner: lmer

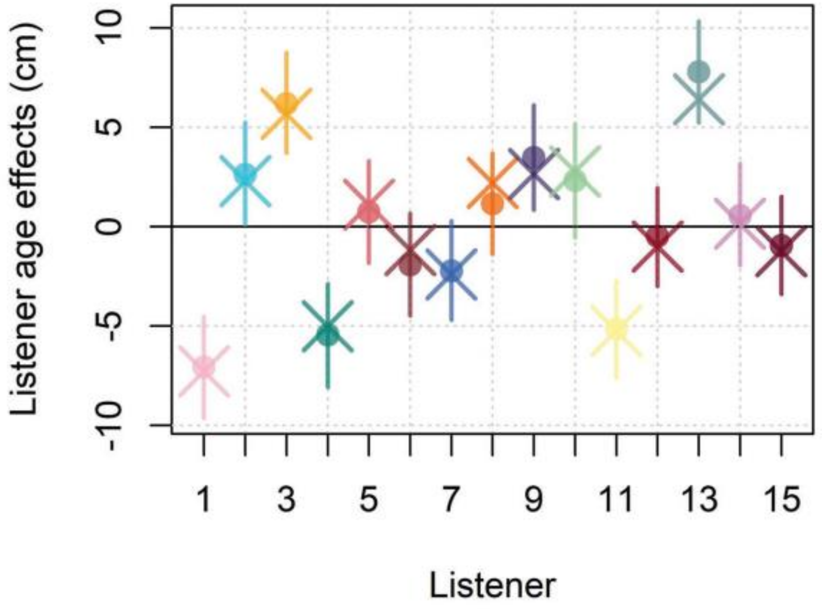
lme4

```
## Random effects:
##  Groups    Name        Std.Dev. Corr
##  S         (Intercept) 3.53
##  L         (Intercept) 4.12
##            A1          3.98     -0.92
```

brms

```
## Group-Level Effects:
## ~L (Number of levels: 15)
##                     Estimate Est.Error l-95% CI u-95% CI
## sd(Intercept)           4.34      0.84     3.05     6.29
## sd(A1)                  4.26      0.85     2.96     6.25
## cor(Intercept,A1)      -0.80      0.12    -0.95    -0.51
```


Listener effects (cm)
Listener


Listener age effects (cm)
Listener