Chapter 8

Lin 205C
Santiago Barreda

Chapter Precap

- Prior predictive checks and the importance of these for model building.
- How to specify more specific prior probabilities for individual model parameters.
- We introduce heteroscedastic models, that is, models with error terms that vary from observation to observation.
- We present a ‘simple’ model that includes only variation in the error term across two conditions.
- We present a ‘complex’ model that features listener-dependent error terms fit using shrinkage, in addition to the equivalent of listener ‘random effects’ for the error term.
- Finally, we discuss building identifiable models, models supported by the available data, collinearity, linear dependence, and saturated models.

More About Priors

- I many cases (in linguistics) the prior will have little to no practical effect.

- Use domain knowledge to get priors in the ballpark: 12 lightyears or 12 nanometers?

- Sometimes this is easy (human height), sometimes this is not so easy (the effect of lexical frequency on log reaction times).

Prior Predictive Checks

- Create fake data sampling only from the prior (ignore the likelihood i.e. the data).

```
# Fit the model yourself
priors = c(brms::set_prior("student_t(3,156, 1000)", class = "Intercept"),
           brms::set_prior("student_t(3,0, 1000)", class = "b"),
           brms::set_prior("student_t(3,0, 1000)", class = "sd"),
           brms::set_prior("lkj_corr_cholesky (1000)", class = "cor"),
           brms::set_prior("student_t(3,0, 1000)", class = "sigma"))
```

```
prior_uninformative =
  brms::brm (height ~ A + G + A:G + (A + G + A:G|L) + (1|S),
             sample_prior="only", data = exp_data, chains = 4, cores = 4,
             warmup = 1000, iter = 5000, thin = 4, prior = priors)
```

Prior Predictive Checks

Uninformative Priors

```
priors = c(brms::set_prior("student_t(3,156, 1000)", class = "Intercept"),
           brms::set_prior("student_t(3,0, 1000)", class = "b"),
           brms::set_prior("student_t(3,0, 1000)", class = "sd"),
           brms::set_prior("lkj_corr_cholesky (1000)", class = "cor"),
           brms::set_prior("student_t(3,0, 1000)", class = "sigma"))
```

Mildly informative Priors

```
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 (12)", class = "cor"),
           brms::set_prior("student_t(3,0, 12)", class = "sigma"))
```

Conservative Priors

```
priors = c(brms::set_prior("student_t(3,156, 6)", class = "Intercept"),
           brms::set_prior("student_t(3,0, 6)", class = "b"),
           brms::set_prior("student_t(3,0, 6)", class = "sd"),
           brms::set_prior("lkj_corr_cholesky (2)", class = "cor"),
           brms::set_prior("student_t(3,0, 6)", class = "sigma"))
```

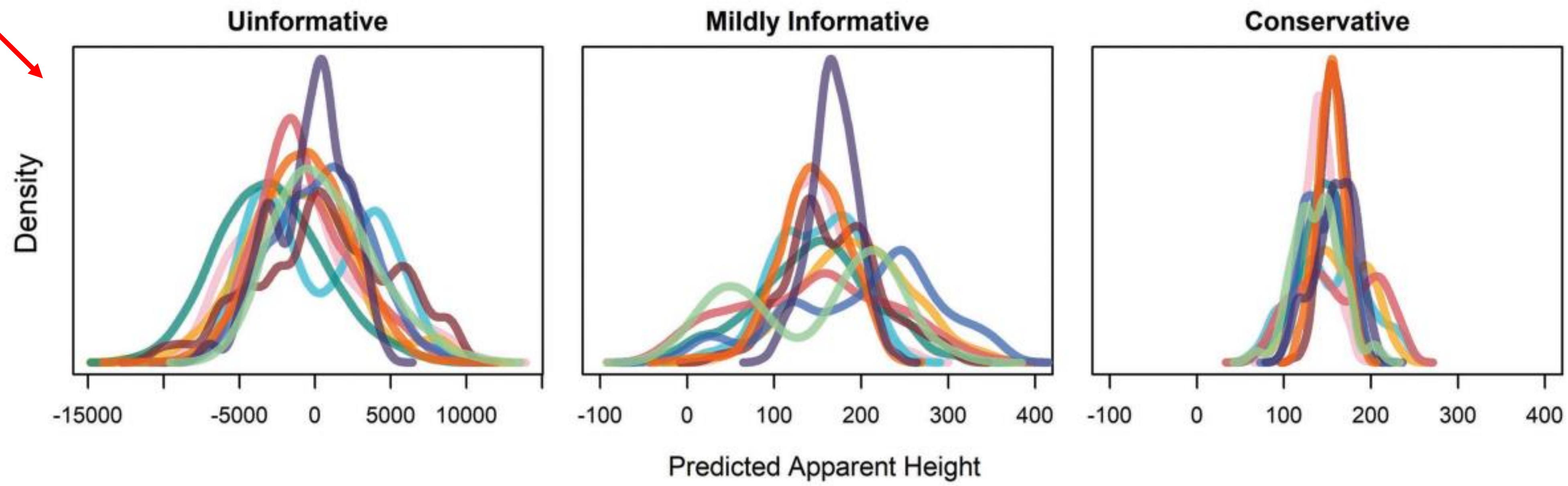
Prior Predictive Checks

```
pp_uninformative = predict (prior_uninformative, summary = FALSE)
```

```
hist (pp_uninformative[1,])
```

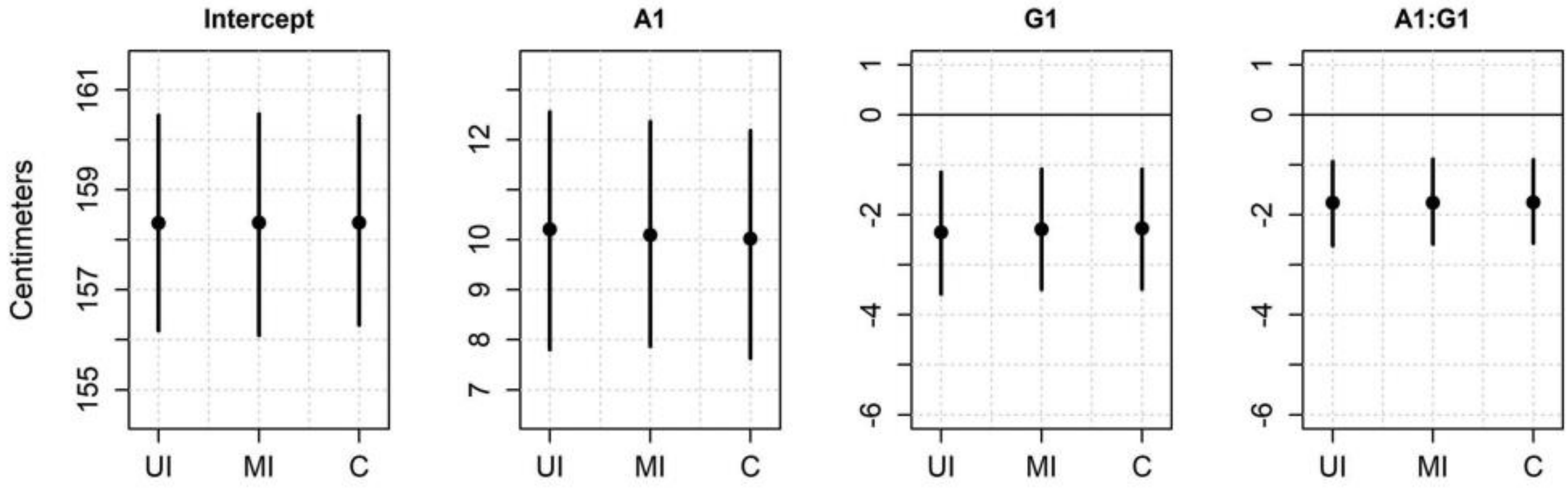
```
bmmb::p_check (pp_uninformative)
```

NOT compared to data. Compared to domain knowledge.



No Effect on Results!

UI: Uninformative MI: Mildly informative C: Conservative



More Specific Priors

```
# we omit empty columns to let the output fit on the page
bmmb::prior_summary(model_mildly_informative)[,-c(5:9)]
##                  prior     class      coef group  source
##      student_t(3,0, 12)         b                   user
##      student_t(3,0, 12)         b        A1      default
##      student_t(3,0, 12)         b     A1:G1      default
##      student_t(3,0, 12)         b        G1      default
##    student_t(3,156, 12) Intercept                   user
##   lkj_corr_cholesky (2)         L                   user
##   lkj_corr_cholesky (2)         L                L default
##      student_t(3,0, 12)        sd                   user
##      student_t(3,0, 12)        sd                L default
##      student_t(3,0, 12)        sd        A1      L default
##      student_t(3,0, 12)        sd     A1:G1      L default
##      student_t(3,0, 12)        sd        G1      L default
##      student_t(3,0, 12)        sd Intercept      L default
##      student_t(3,0, 12)        sd                S default
##      student_t(3,0, 12)        sd Intercept      S default
##      student_t(3,0, 12)     sigma                   user
```

More Specific Priors

```r
priors =
  c(set_prior("student_t(3,156, 6)", class = "Intercept"),
    set_prior("student_t(3,0, 6)", class = "b"),
    set_prior("student_t(3,0, 10)", class = "b", coef = "A1"),
    set_prior("student_t(3,0, 3)", class = "b", coef = "G1"),
    set_prior("student_t(3,0, 10)", class = "sd"),
    set_prior("student_t(3,0, 5)", class = "sd", coef = "A1", group="L"),
    set_prior("student_t(3,0, 1.5)", class = "sd", coef = "G1", group="L"),
    set_prior("lkj_corr_cholesky (2)", class = "cor"),
    set_prior("student_t(3,0, 6)", class = "sigma"))
```

```r
bmmb::prior_summary(prior_informative)[,-c(5:8)]
##                  prior     class      coef group
##      student_t(3,0, 6)         b
##     student_t(3,0, 10)         b        A1
##     student_t(3,0, 10)         b     A1:G1
##      student_t(3,0, 3)         b        G1
##    student_t(3,156, 6) Intercept
##  lkj_corr_cholesky (2)         L
##  lkj_corr_cholesky (2)         L               L
##     student_t(3,0, 10)        sd
##     student_t(3,0, 10)        sd               L
##      student_t(3,0, 5)        sd        A1     L
##      student_t(3,0, 5)        sd     A1:G1     L
##    student_t(3,0, 1.5)        sd        G1     L
##    student_t(3,0, 1.5)        sd Intercept     L
##    student_t(3,0, 1.5)        sd               S
##    student_t(3,0, 1.5)        sd Intercept     S
##      student_t(3,0, 6)     sigma
```

Data and Research Questions

```
library (brms)
library (bmmb)
data (exp_data)
options (contrasts = c('contr.sum','contr.sum'))
```

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

(Q1) Does our error standard deviation vary as a function of apparent speaker age?
(Q2) Does our error standard deviation vary as a function of the listener?

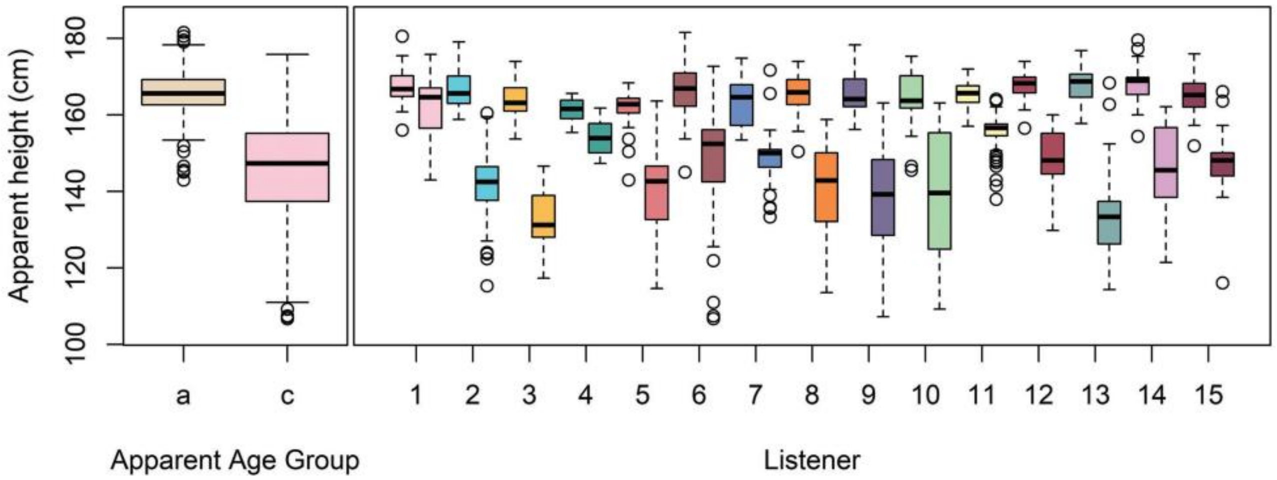
Homoskedasticity

- Most regression models only predict variation in means.

- A single error variance is assumed for all data (i.e., σ doesn't get a subscript).

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

$$\mu_{[i]} = x_1 + x_2 + \ldots + x_p$$



Heteroskedasticity

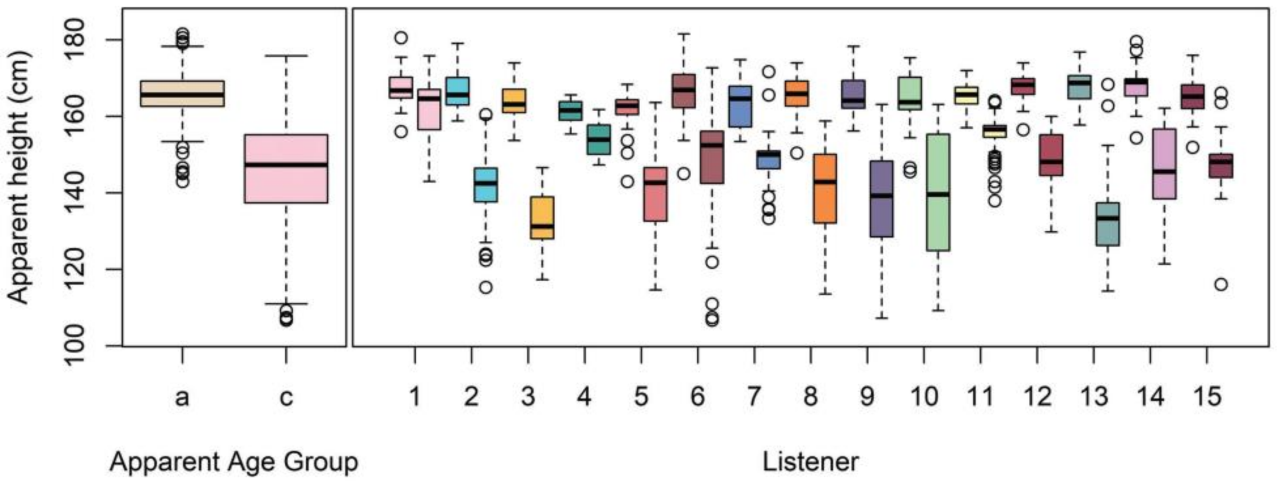
- With Bayesian models it is trivial to model variation in error variances as well.

- Different error variances can exist for different data (i.e., σ does get a subscript).

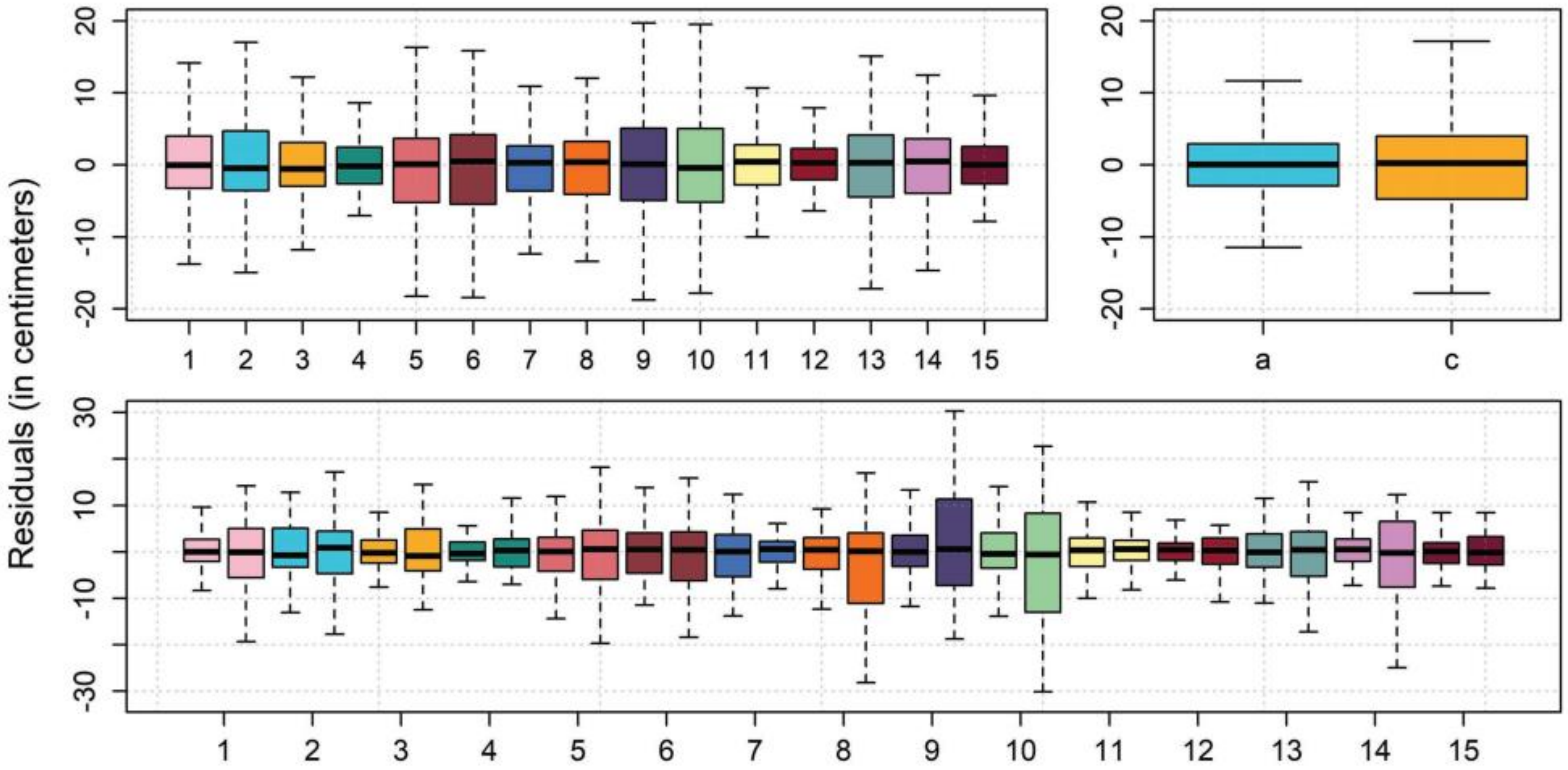
$$\text{height}_{[i]} \sim \mathrm{N}\left(\mu_{[i]}, \sigma_{[i]}\right)$$

$$\mu_{[i]} = x_1 + x_2 + \ldots + x_p$$

$$\sigma_{[i]} = x_{\sigma 1} + x_{\sigma 2} + \ldots + x_{\sigma p}$$



Heteroskedasticity



Description of Our Model

Chapter 6

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

New component,
similar to chapter 5.

```
sigma ~ A
```

Together in one model

```
model_formula = brms::bf(height ~ A*G + (A*G|L) + (1|S),
                         sigma ~ A + (A|L))
```

Description of Our Model

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

$$\log\left(\sigma_{[i]}\right) = \text{Intercept}_\sigma + A_\sigma$$

$$\text{Priors:}$$
$$S_{[\bullet]} \sim \text{t}(3, 0, \sigma_S)$$
$$\begin{bmatrix} L_{[\bullet]} \\ A:L_{[\bullet]} \\ G:L_{[\bullet]} \\ A:G:L_{[\bullet]} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \Sigma\right)$$

$$\text{Intercept} \sim \text{t}(3, 156, 12)$$
$$A, G, A:G \sim \text{t}(3, 0, 12)$$
$$\sigma_L, \sigma_{A:L}, \sigma_{G:L}, \sigma_{A:G:L}, \sigma_S \sim \text{t}(3, 0, 12)$$
$$\nu \sim \text{gamma}(2, 0.1)$$
$$R \sim \text{LKJCorr}(2)$$
$$\text{Intercept}_\sigma \sim \text{N}(0, 1.5)$$
$$A_\sigma \sim \text{N}(0, 1.5)$$

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

$$\text{Intercept}_\sigma \sim \text{t}(3, 0, 1.5)$$
$$A_\sigma \sim \text{t}(3, 0, 1.5)$$

Fitting the Model

```
model_formula = brms::bf(height ~ A*G + (A*G|L) + (1|S),
                         sigma ~ A)

priors =
  c(set_prior("student_t(3, 156, 12)", class = "Intercept"),
    set_prior("student_t(3, 0, 12)", class = "b"),
    set_prior("student_t(3, 0, 12)", class = "sd"),
    set_prior("gamma(2, 0.1)", class = "nu"),
    set_prior("normal(0, 1.5)", class = "Intercept", dpar = "sigma"),
    set_prior("normal(0, 1.5)", class = "b", dpar = "sigma"),
    set_prior("lkj_corr_cholesky (2)", class = "cor"))

prior_A_sigma =
  brms::brm (model_formula, data = exp_data, chains = 4, cores = 4,
             warmup = 1000, iter = 3500, thin = 2, family="student",
             prior = priors, sample_prior = "only")
```

Inspecting the Fixed Effects

```
fixef (model_A_sigma)
##                   Estimate Est.Error    Q2.5    Q97.5
## Intercept          158.383   1.10922 156.193 160.6196
## sigma_Intercept      1.724   0.03265   1.659   1.7881
## A1                  11.271   1.19021   8.915  13.6901
## G1                  -3.067   0.59091  -4.271  -1.9084
## A1:G1               -1.528   0.43464  -2.403  -0.6654
## sigma_A1            -0.247   0.02199  -0.290  -0.2036
```

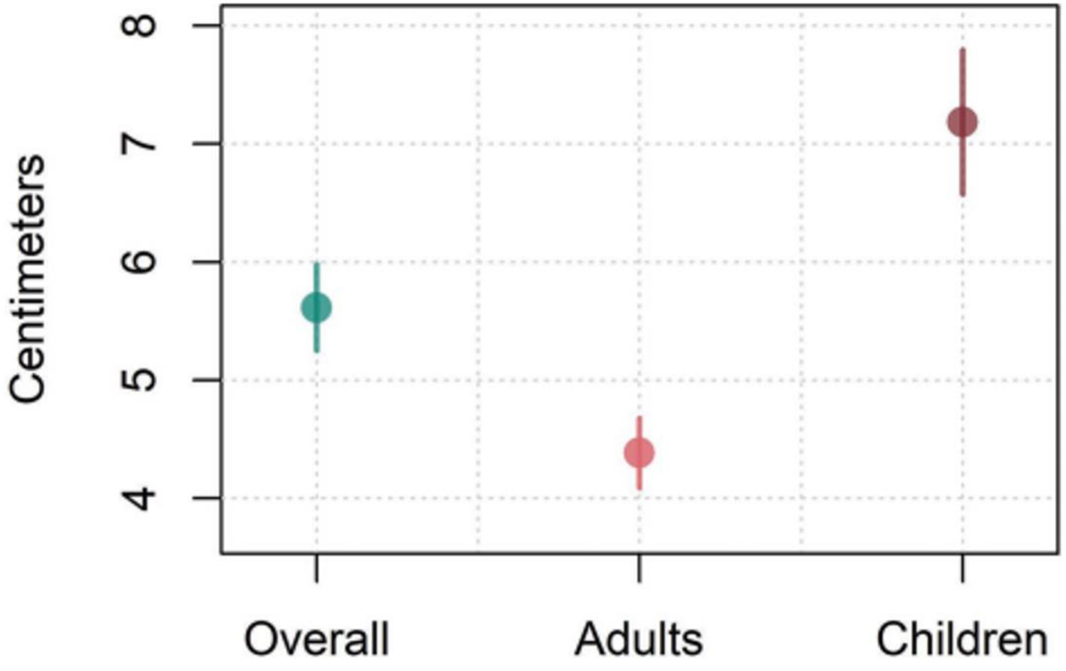
In log units. Must exponentiate values to get original units, i.e., σ = exp (sigma_Intercept)

Recovering predicted Sigmas

You <u>MUST</u> combine parameters <u>before</u> exponentiating. Otherwise, the results will be <u>totally wrong</u>.

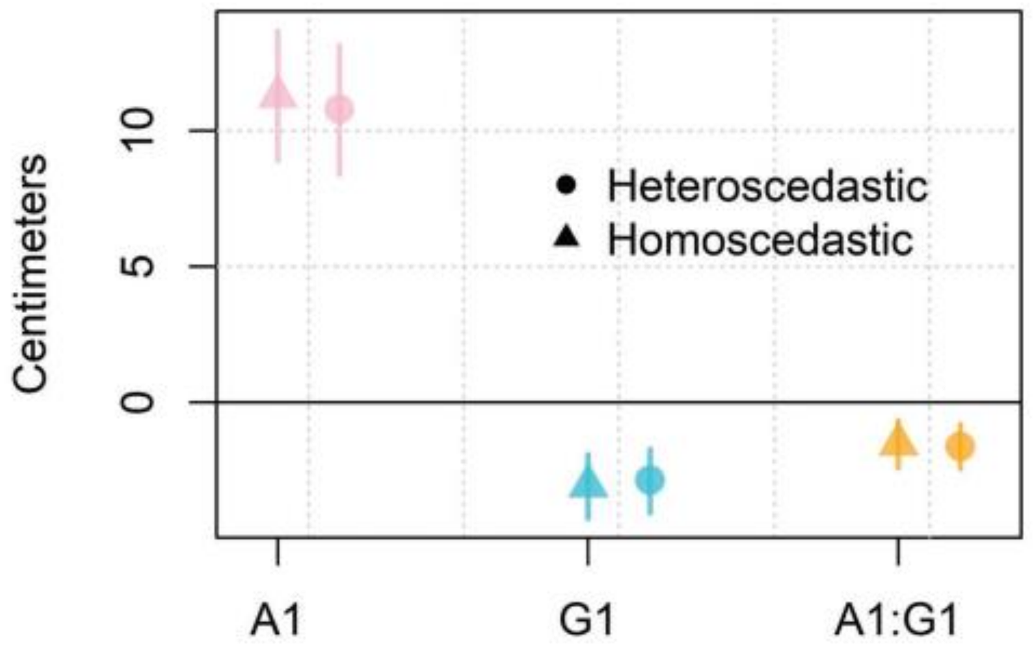
```
sigmas = short_hypothesis(
  model_A_sigma,
  c("exp(sigma_Intercept) = 0",                    # overall sigma
    "exp(sigma_Intercept + sigma_A1) = 0",          # adult sigma
    "exp(sigma_Intercept - sigma_A1) = 0"))         # child sigma
```

```
sigmas[,-5]
##     Estimate Est.Error  Q2.5 Q97.5
## H1     5.609    0.1830 5.253 5.978
## H2     4.382    0.1526 4.085 4.680
## H3     7.184    0.3115 6.573 7.793
```



Model Comparison

```
model_interaction = add_criterion(model_interaction,"loo")
model_A_sigma = add_criterion(model_A_sigma,"loo")
loo_compare (model_interaction, model_A_sigma)
##                   elpd_diff se_diff
## model_A_sigma        0.0       0.0
## model_interaction -53.7      12.8
```



A ‘Complex’ Model

- Our last model had only a ‘fixed effect’ for apparent age. `sigma ~ A`

- If we were predicting the mean, we would want to include random effects for listener.

- We can include these in our prediction of sigma as well. `sigma ~ A + (A|L)`

Our Model Formula

- When we have ‘the same’ random effects across formulas, we need to let our model know this.

```
model_formula = brms::bf(height ~ A*G + (A*G|L) + (1|S),
                         sigma ~ A + (A|L))
```

- We do this by putting the same indicator between pipes (|inhere|) before a random effect.

```
model_formula = brms::bf(height ~ A*G + (A*G|x|L) + (1|y|S),
                         sigma ~ A + (A|x|L) + (1|y|S))
```

- Our model will estimate the correlation of random effects across predicted variables!

Model Description

$$\text{height}_{[i]} \sim \mathrm{N}\left(\mu_{[i]}, \sigma_{[i]}\right)$$

$$\mu_{[i]} = \text{Intercept} + A + G + A:G +$$

$$L_{[\mathsf{L}[i]]} + A:L_{[\mathsf{L}[i]]} + G:L_{[\mathsf{L}[i]]} + A:G:L_{[\mathsf{L}[i]]} + S_{[\mathsf{S}[i]]}$$

$$\log\left(\sigma_{[i]}\right) = \text{Intercept}_{\sigma} + A_{\sigma} + A_{\sigma}:L_{\sigma[\mathsf{L}[i]]} + L_{\sigma[\mathsf{L}[i]]}$$

Priors:

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

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

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

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

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

$$\text{Intercept}_{\sigma} \sim \mathrm{N}(0, 1.5)$$

$$\mathrm{A}_{\sigma} \sim \mathrm{N}(0, 1.5)$$

$$\sigma_{L_{\sigma}}, \sigma_{A_{\sigma}:L_{\sigma}} \sim \mathrm{N}(0, 1.5)$$

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

Fitting the Model

```
# Fit the model yourself
model_formula = brms::bf(height ~ A*G + (A*G|x|L) + (1|S),
                         sigma ~ A + (A|x|L))

priors =
  c(set_prior("student_t(3, 156, 12)", class = "Intercept"),
    set_prior("student_t(3, 0, 12)", class = "b"),
    set_prior("student_t(3, 0, 12)", class = "sd"),
    set_prior("gamma(2, 0.1)", class = "nu"),
    set_prior("normal(0, 1.5)", class = "Intercept", dpar = "sigma"),
    set_prior("normal(0, 1.5)", class = "b", dpar = "sigma"),
    set_prior("normal(0, 1.5)", class = "sd", dpar = "sigma"),
    set_prior("lkj_corr_cholesky (2)", class = "cor"))

model_A_L_sigma =
  brms::brm (model_formula, data = exp_data, chains = 4, cores = 4,
             warmup = 1000, iter = 3500, thin = 2, family="student",
             prior = priors)
```

Inspecting the Results

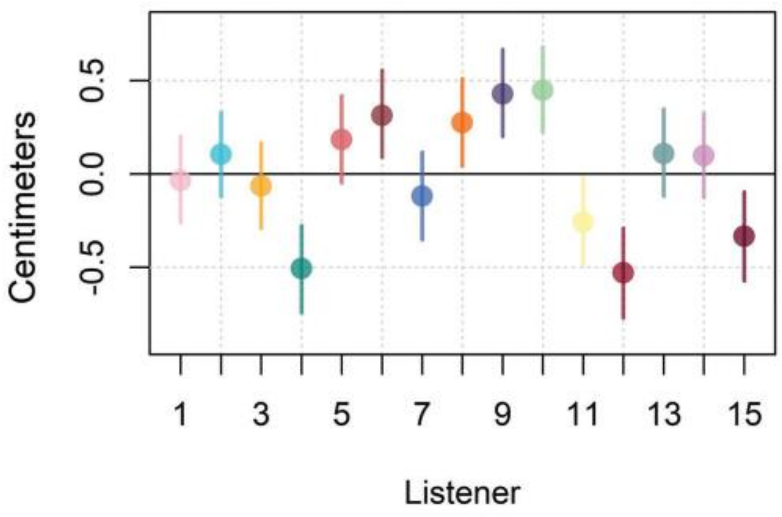
```
bmmb::short_summary (model_A_L_sigma)
## ...
## sd(sigma_Intercept)                 0.36       0.08      0.24      0.54
## sd(sigma_A1)                        0.17       0.04      0.10      0.27
## ...
## cor(A1,sigma_A1)                   -0.29       0.22     -0.67      0.16
## cor(G1,sigma_A1)                    0.22       0.24     -0.27      0.65
## cor(A1:G1,sigma_A1)                 0.12       0.24     -0.35      0.58
## cor(sigma_Intercept,sigma_A1)      -0.44       0.21     -0.79      0.05
## ...
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI
## Intercept         158.29      1.10   156.15   160.51
## sigma_Intercept     1.74      0.10     1.55     1.93
## A1                 11.28      1.18     8.98    13.64
## G1                 -2.92      0.57    -4.06    -1.80
## A1:G1              -1.63      0.42    -2.47    -0.77
## sigma_A1           -0.23      0.05    -0.33    -0.14
##
## Family Specific Parameters:
##    Estimate Est.Error l-95% CI u-95% CI
## nu     8.07      1.54     5.71    11.72
```

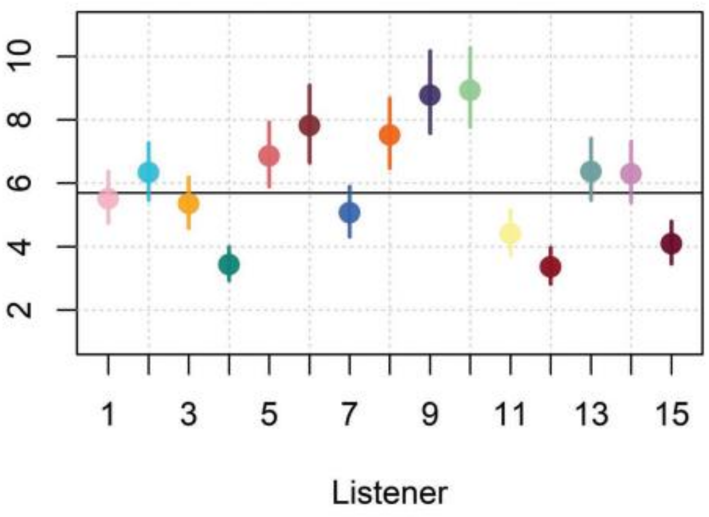
Listener-dependent Error Terms

```
# listener random effects for sigma
log_sigmas_centered = short_hypothesis(model_A_L_sigma, "sigma_Intercept=0",
                              scope = "ranef",group="L")

# the sum of the sigma intercept and the listener sigma random effect
log_sigmas = short_hypothesis(model_A_L_sigma, "sigma_Intercept=0",
                              scope = "coef",group="L")

# the exponent of the sum of the sigma intercept
# and the listener sigma random effect
sigmas = short_hypothesis(model_A_L_sigma, "exp(sigma_Intercept)=0",
                              scope = "coef",group="L")
```





Model Comparison

- Our new model is <u>much</u> better.

```
model_A_L_sigma = add_criterion(model_A_L_sigma, "loo")
```

```
loo_compare (model_interaction, model_A_sigma, model_A_L_sigma)
##                   elpd_diff se_diff
## model_A_L_sigma       0.0       0.0
## model_A_sigma      -122.1      15.6
## model_interaction -175.8      20.8
```

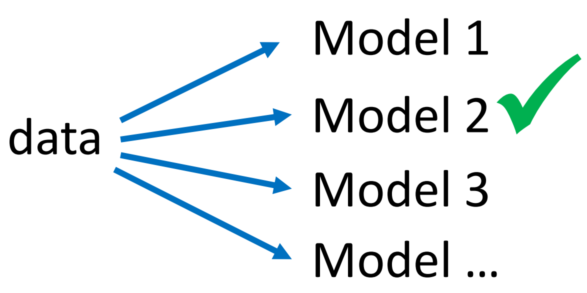
Answering our Research Questions

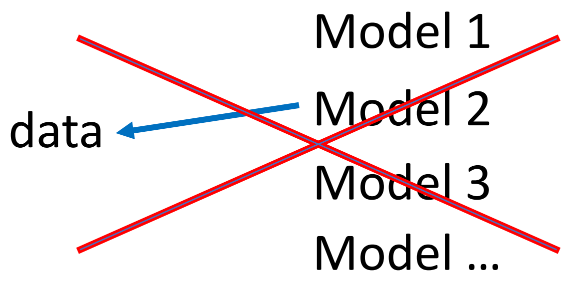
(Q1) Does our error standard deviation vary as a function of apparent speaker age?
(Q2) Does our error standard deviation vary as a function of the listener?

- Yes, and yes.
- Some tougher questions are: Which is the ‘better/rea’ model? Which model should we report?

Underdetermination

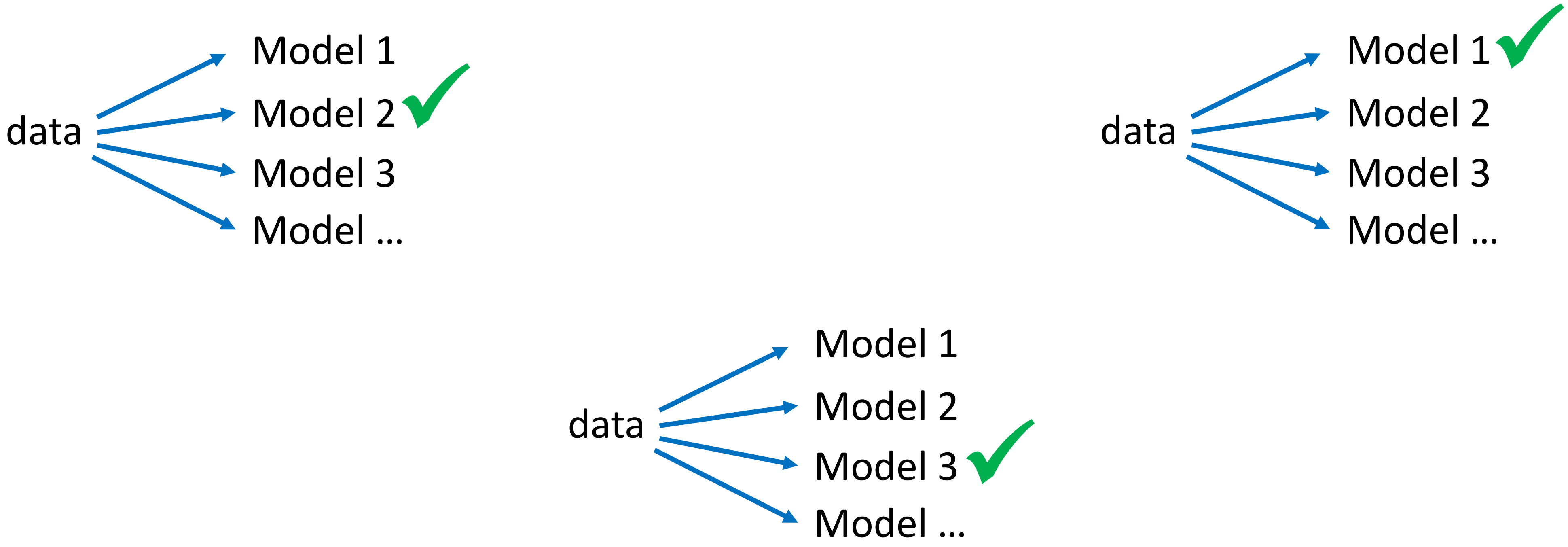
- Given some observations, there are always multiple different interpretations/models.





Researcher Degrees of Freedom

- The decisions a researcher makes during design or analysis that affect the model used and conclusions reached.



Which Model To Report?

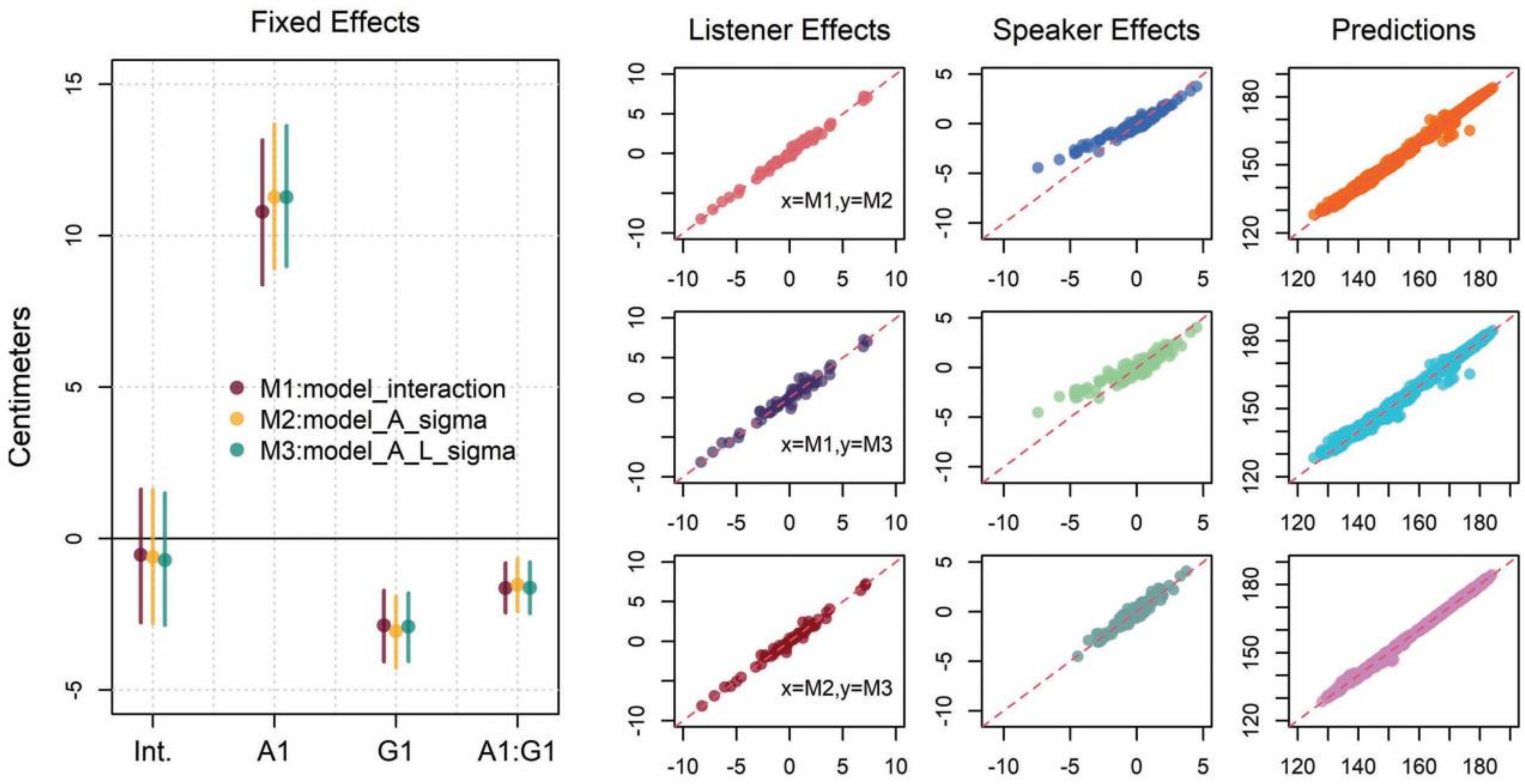
- Our models represent different *information*.
- When considering which to report, we can consider two things:

1. Which parameters/information are we interested in?
2. How much do these differ across models?

    2a. If they differ, why?

Which Model To Report?

- Fixed effects are about the same.
- Speaker effects differ in heteroskedastic models.
- Do you care about the variation in the errors?



Should You fit the Model?

- As we progress, you *can* fit more and more complicated models.
- But *should* you?
- Two things to worry about:
  - Identifiability: the ability to estimate unique, independent values for all your parameters.
  - Support: Having enough data to realistically estimate all your parameters.

Collinearity

- A set of predictor vectors is linearly independent when there is no vector of non-zero numbers that can be used to combine our predictors such that they <u>always</u> equal zero.

$$0 = x_1 \cdot a_1 + x_2 \cdot a_2 + \ldots + x_n \cdot a_n$$

If this is possible the x predictors are not linearly dependent.

Collinearity

- You <u>cannot</u> fit a model using height in meters and height in centimeters, and estimate both effects independently.

```
exp_data$vtl_m = exp_data$vtl / 100
```

```
exp_data$vtl + exp_data$vtl_m * -100
```

- These predictors are linearly dependent!

$$x_1 \cdot a_1 + x_2 \cdot a_2 = 0$$

$$\mathrm{vtl} \cdot 1 + \mathrm{vtl_m} \cdot -100 = 0$$

Collinearity

```
model_bad_1 =
  brms::brm (height ~ vtl_m + vtl, data = exp_data, chains = 4, cores = 4,
       warmup = 1000, iter = 3500, thin = 2,
       prior = c(brms::set_prior("normal(176, 50)", class = "Intercept"),
                 brms::set_prior("normal(0, 15)", class = "sigma")))
```

```
## Population-Level Effects:
##            Estimate Est.Error   l-95% CI  u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     45.69      2.37      41.17     50.47 1.00     4031     4425
## vtl_m       1383.91 246800.93 -531293.80 343951.54 2.12        5       13
## vtl           -5.28   2468.01   -3430.88   5321.56 2.12        5       13
```

Semi-Collinearity

- Linear dependence is binary.

- Correlation is a continuous measure of linear dependence (basically).

```
cor (exp_data$vtl_m, exp_data$vtl)
## [1] 1
```

- What is we make our predictors almost linearly dependent?

```
set.seed(1)
exp_data$vtl_m_noise = exp_data$vtl_m +
  rnorm (length(exp_data$vtl_m),0,sd(exp_data$vtl_m)/10)

cor (exp_data$vtl, exp_data$vtl_m_noise)
## [1] 0.9946
```

Semi-Collinearity: Not that Bad

```
model_bad_2 =
  brms::brm (height ~ vtl_m_noise + vtl, data = exp_data, chains = 4,
           cores = 4,warmup = 1000, iter = 3500, thin = 2, prior = priors)
```

```
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      45.75      2.36    41.16    50.35 1.00     4523     4087
## vtl_m_noise  -201.86    175.11  -544.11   139.03 1.00     3504     3697
## vtl            10.57      1.76     7.14    14.01 1.00     3439     3866
```

Semi-Collinearity: Could be Better

```
model_good =
  brms::brm (height ~ vtl, data = exp_data, chains = 4, cores = 4,
        warmup = 1000, iter = 3500, thin = 2,
        prior = c(brms::set_prior("normal(176, 50)", class = "Intercept"),
                  brms::set_prior("normal(0, 15)", class = "sigma")))
```

```
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     45.72      2.38    41.04    50.40 1.00     5052     4679
## vtl            8.55      0.18     8.21     8.90 1.00     5072     4831
```

```
model_bad_2 =
  brms::brm (height ~ vtl_m_noise + vtl, data = exp_data, chains = 4,
          cores = 4,warmup = 1000, iter = 3500, thin = 2, prior = priors)
```

```
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept       45.75      2.36    41.16    50.35 1.00     4523     4087
## vtl_m_noise   -201.86    175.11  -544.11   139.03 1.00     3504     3697
## vtl             10.57      1.76     7.14    14.01 1.00     3439     3866
```

Predictable Values for Predictors

- You can't fit models where the value of one categorical predictors can be guessed based on the vlues of other predictors.

- This is again a problem of linear dependence.

- This is why we can't estimate both levels of a two-group factor.

```
x = cbind (intercept=rep(1,4), x1=rep(c(1,0),2),
x2=rep(c(0,1),2))
x
##      intercept x1 x2
## [1,]         1  1  0
## [2,]         1  0  1
## [3,]         1  1  0
## [4,]         1  0  1
```

```
x[,1] + x[,2]*(-1) + x[,3]*(-1)
## [1] 0 0 0 0
```

Predictable Values for Predictors

- And also why we can't estimate all levels of a four level factor.

```
x = cbind (intercept=rep(1,4), C1=c(1,0,0,0), C2=c(0,1,0,0),
           C3=c(0,0,1,0),C4=c(0,0,0,1))
x
##      intercept C1 C2 C3 C4
## [1,]         1  1  0  0  0
## [2,]         1  0  1  0  0
## [3,]         1  0  0  1  0
## [4,]         1  0  0  0  1
```

```
x[,1] + x[,2]*(-1) + x[,3]*(-1) + x[,4]*(-1) + x[,5]*(-1)
## [1] 0 0 0 0
```

Predictable Values for Predictors

- And also why we can't include group, age, *and* gender.

```
x = cbind (intercept=rep(1,4), C1=c(1,0,0,0), C2=c(0,1,0,0),
           C3=c(0,0,1,0),A1=c(0,0,1,1), G1=c(0,1,0,1),A1G1=c(1,0,0,1))
```

```
x
##      intercept C1 C2 C3 A1 G1 A1G1
## [1,]         1  1  0  0  0  0    1
## [2,]         1  0  1  0  0  1    0
## [3,]         1  0  0  1  1  0    0
## [4,]         1  0  0  0  1  1    1
```

```
x[,1]*1 + x[,2]*(-1) + x[,3]*(-1) + x[,5]*(-1)
## [1] 0 0 0 0
```

```
x[,1]*1 + x[,3]*(-1) + x[,4]*(-1) + x[,7]*(-1)
## [1] 0 0 0 0
```

Predictable Values for Predictors

```
model_bad_3 =
  brms::brm (height ~ C + A*G, data = exp_data, chains = 4, cores = 4,
       warmup = 1000, iter = 3500, thin = 2,
       prior = c(brms::set_prior("normal(176, 15)", class = "Intercept"),
                 brms::set_prior("normal(0, 15)", class = "sigma")))
```

```
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept   157.98      0.20   157.57   158.35 1.06       61       78
## C1         1066.31   2604.72 -4431.79  5158.68 2.07        5       18
## C2          496.91   1751.32 -2543.19  3708.83 1.87        6       13
## C3           74.39   2086.76 -4113.31  2801.74 1.99        5       21
## A1          793.68   1363.24 -2383.78  3019.96 1.90        6       12
## G1          567.63   2171.69 -4107.90  3364.23 2.04        5       18
## A1:G1       283.89   1112.60 -2295.30  2141.73 2.26        5       18
```

This is not ideal.

Saturated Models

- Saturated models have one parameter for every observation.
- Without shrinkage this means that there is no random variation in the model, i.e., the error cannot be estimated.

```
exp_data$S = factor (exp_data$S)
exp_data$L = factor (exp_data$L)

model_bad_4  =
  brms::brm (height ~ S*L, data = exp_data, chains = 4, cores = 4,
       warmup = 1000, iter = 3500, thin = 2,
       prior = c(brms::set_prior("normal(176, 15)", class = "Intercept"),
                 brms::set_prior("normal(0, 15)", class = "sigma")))
```

This model crashes my session of R.

Nearly-Saturated Models

- A model can be ‘nearly’-saturated.

- What if you have 2 observations per speaker per listener. You <u>can</u> fit this model (probably), but <u>should</u> you?

- Follow up: What is your n?

- Think of your n for each individual parameter rather than overall.

Exercises: Week 7

Use the data in ‘exp_ex’ to do one of the following. You may also use your own data to answer a related question. In any case, describe the model, present and explain the results. More requirements:

- You must include a model at least this complex: y ~ A*B + (A*B|L), meaning two factors and an interaction.
- Fit a new model, not like in the book and not like for week 6.
- Include at least two non-book figures.
- Recreate the predicted group means and report them.
- Report and interpret at least one simple effect.