Quantitative analysis for corpus phonetics and phonology

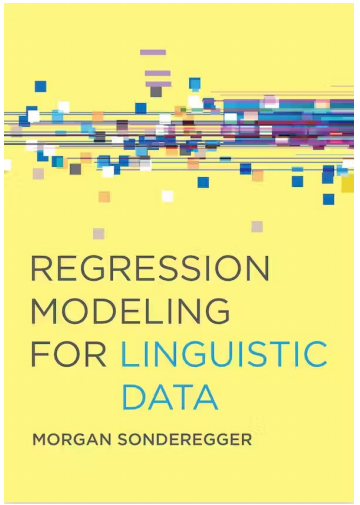
Morgan Sonderegger (McGill U.)

Unlaboratory Phonology: Corpus Approaches

July 18, 2023

Introduction

- Me:
  - Laboratory phonology, phonetics, quantitative analysis
  - Mostly corpus data, also laboratory & field



- Today: topics in quant. analysis of linguistic data
  - Focus: particularly relevant for corpus data
  - Not: a comprehensive introduction to any topic
  - For (mixed-effects) linear and logistic regression
    - Extends to more complex models (Bayesian, GAMMs)

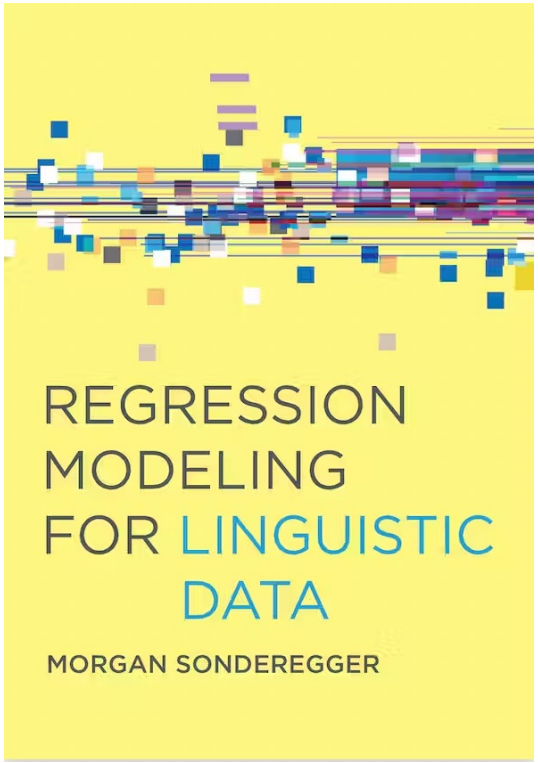
Roadmap

- Part 1: Datasets & visualization
- Part 2: Variable selection
- Part 3: Unpacking model results
- Part 4: Mixed-effects models

full treatments in *RMLD*

- 10 min break: after 1.5 hours

Resources



"RMLD", e.g.

RMLD: 5.1.2, A.2

- Much of material today (Parts 2-4): guided tour of book sections, emphasis on corpus data
- Book OSF page: preprint, datasets, code

coming down soon

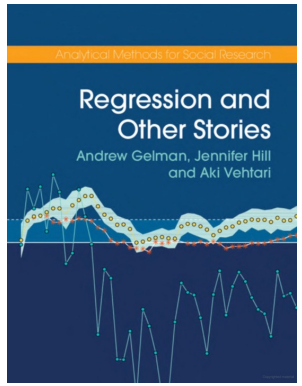
- Workshop OSF page
  - today's slides
  - underlying datasets, code

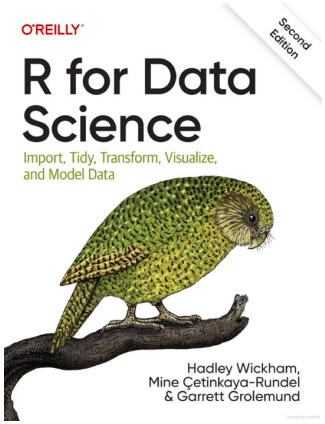
Book: https://osf.io/pnumg/

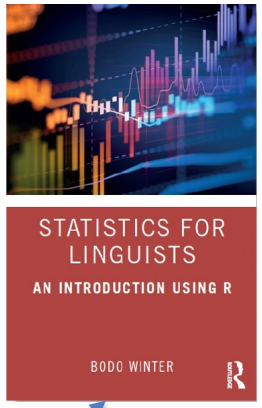
Workshop: https://osf.io/qdp8u/

Part 1: Datasets & visualization

- `vot` dataset
  - continuous *y*
  - linear regression
- `french_cdi_24` dataset
  - categorical *y*
  - logistic regression
- Visualization / model predictions
  - Here: a few points important for corpus data





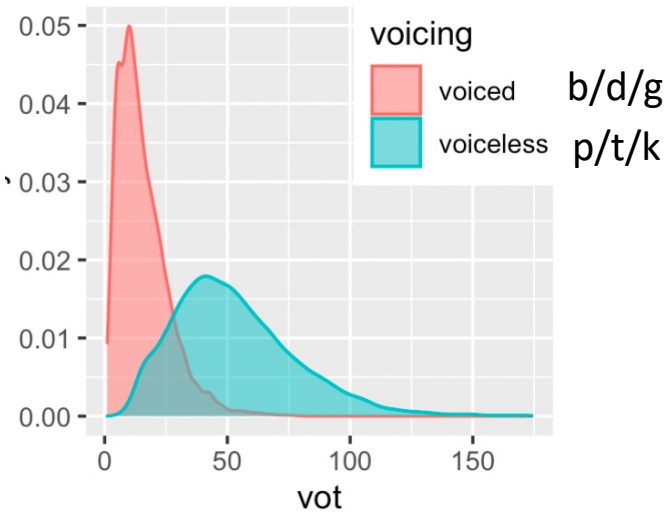


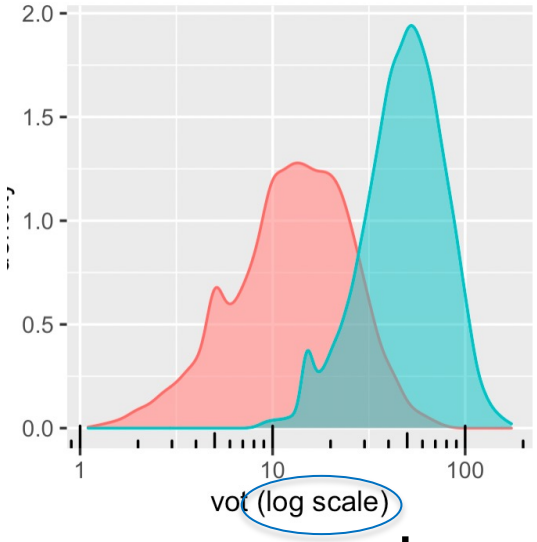
General references:

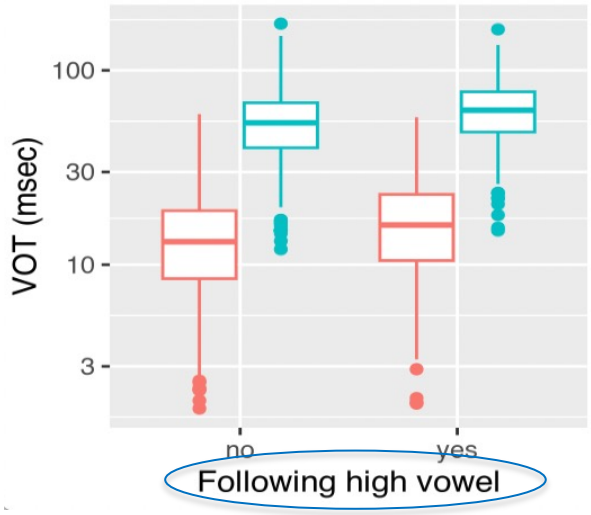
Dataset: vot

RMLD: 5.1.2, A.2

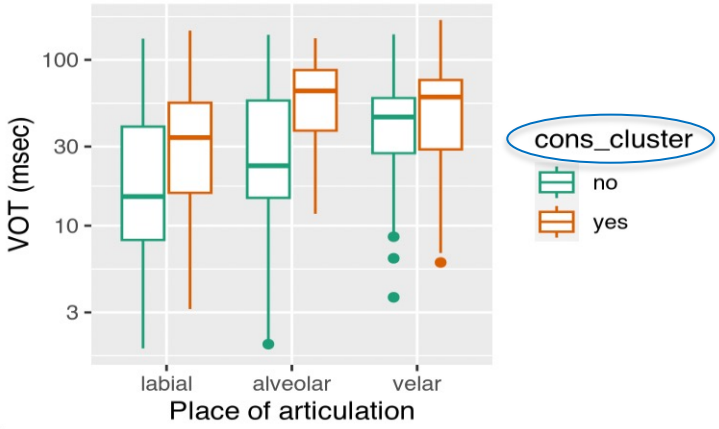
n = 25k







- British English spontaneous speech
- 20 speakers, 1752 words
- Predictors:
  - Word-level: C `voicing`, word `frequency`, `place` of articulation, ...
  - Speaker-level: `gender`
  - Observation-level: `speaking_rate`



(Sonderegger et al., 2017)

Visualizing corpus data

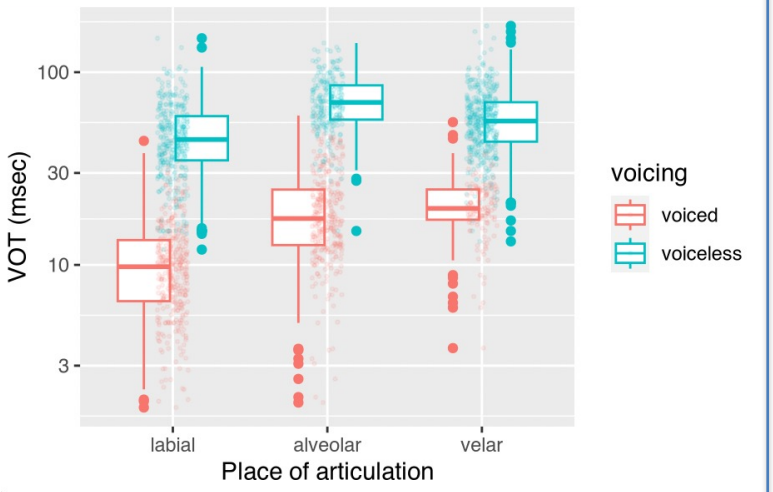
- Corpus data tends to be
  - grouped: by-speaker, by-word, etc.
  - unbalanced: different *n* per speaker, word, etc.
    - Unlike much linguistic data (from expts)
  - messy: correlated predictors ("collinearity"), …

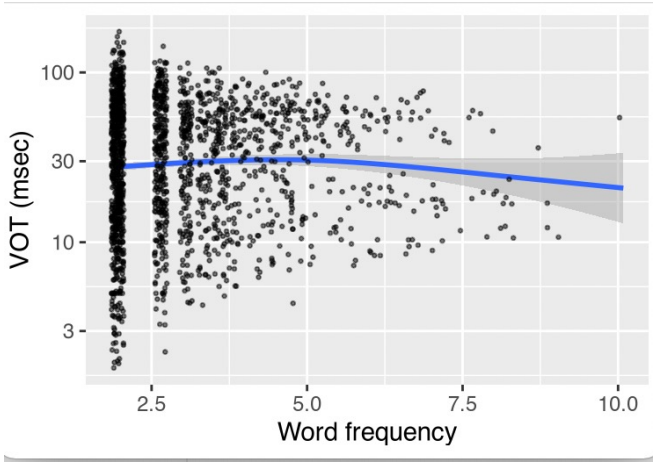
- All OK, using mixed-effects regression models
  - No assumption of balance, collinearity
  - Grouping taken into account (e.g. Gelman & Hill, 2007)

- But all affect visualization practice
  - Ex 1: How/whether to average before plotting
  - Ex 2: Empirical plots vs. model predictions

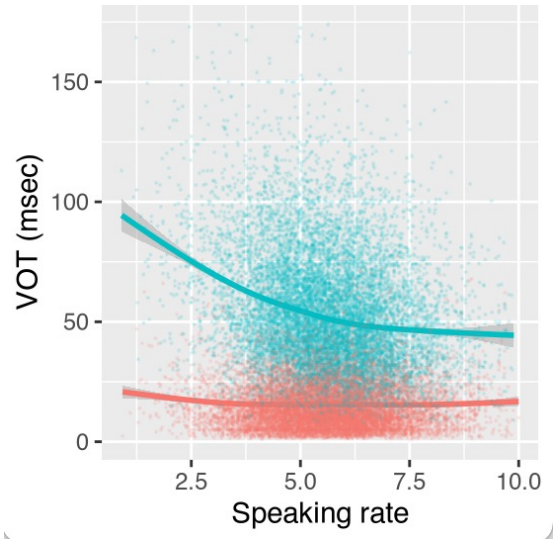
- Some empirical plots:

plots of by-word averages

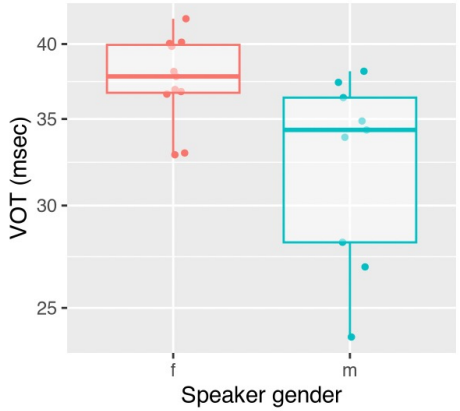




plot of observation VOTs

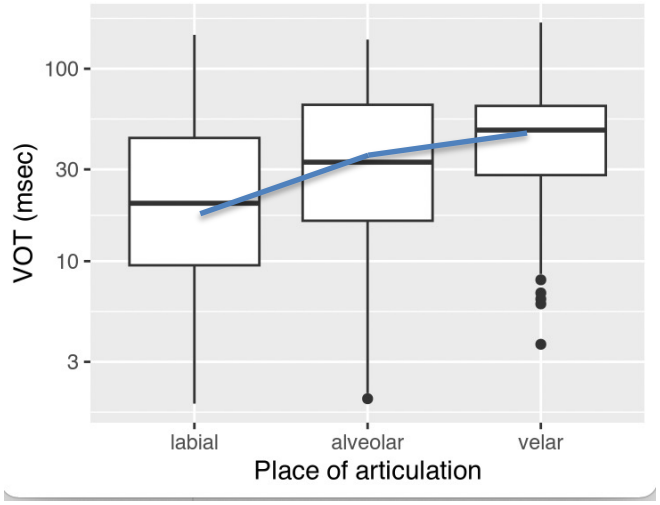


plot of by-speaker averages

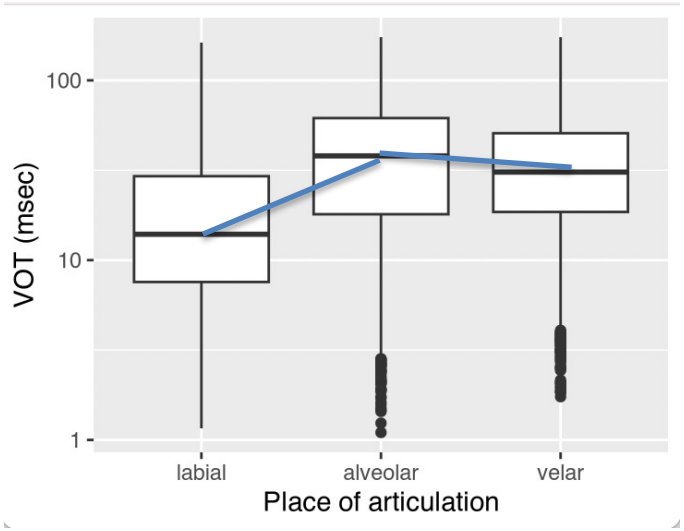


visualization should reflect the level of the predictor(s)

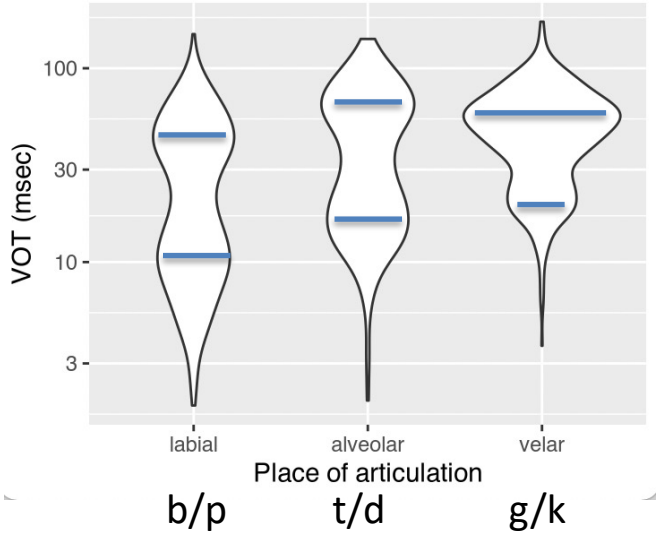
Plots of by-word average VOT (*n* = 1.7k)

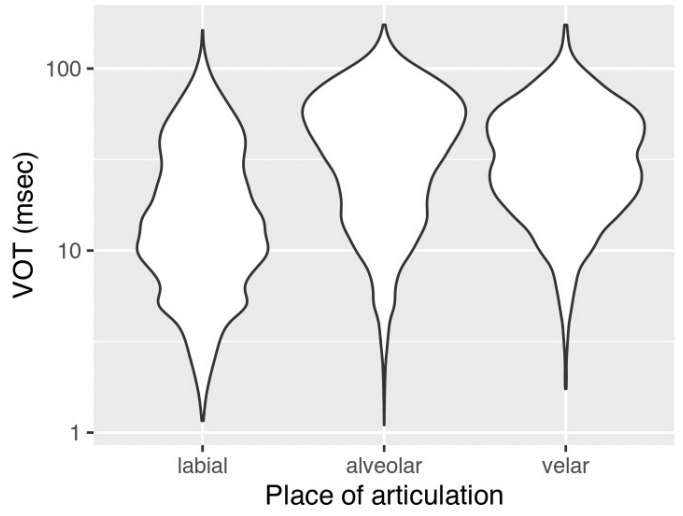


Plots of VOT (*n* = 25k)



qualitative pattern differs





Important aspect of data missed

stop voicing matters

- Another perspective:
  - Visualization answers question(s) about data
  - A decent model should be assumed for the question, including by-X random effects for X-level predictors

  - Bad: `log(VOT) ~ place`

  - Better: `log(VOT) ~ place + (1|word)`

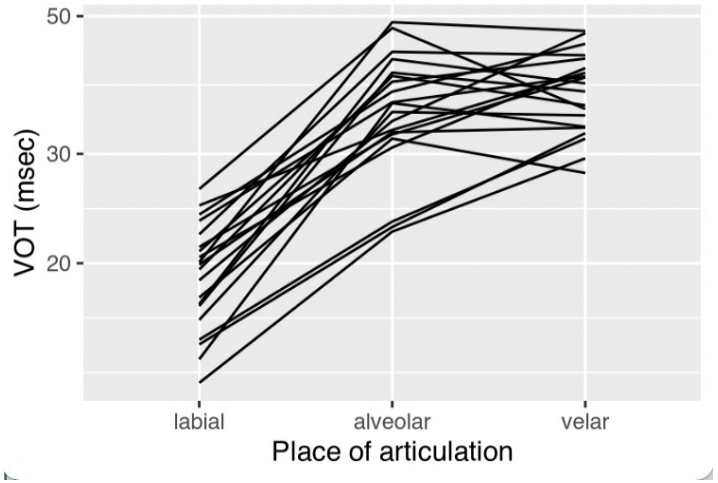
left column in prev. slide

right column in prev. slide

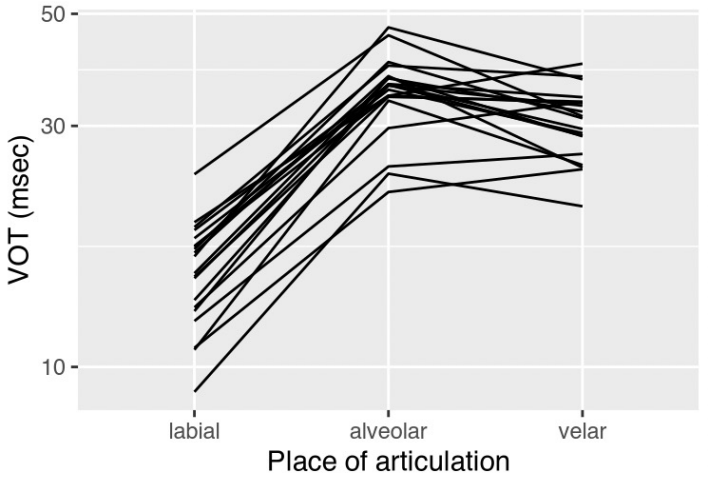
- Unbalanced data ⇒ more important
  - Otherwise plots will be dominated by frequent words, speakers, etc.

- Generalizes to more complex plots
- Ex: empirical effect of place for each speaker

Plot using speaker-word pair average VOT (*n* = 5.5k)



Plot using VOTs (*n* = 25k)



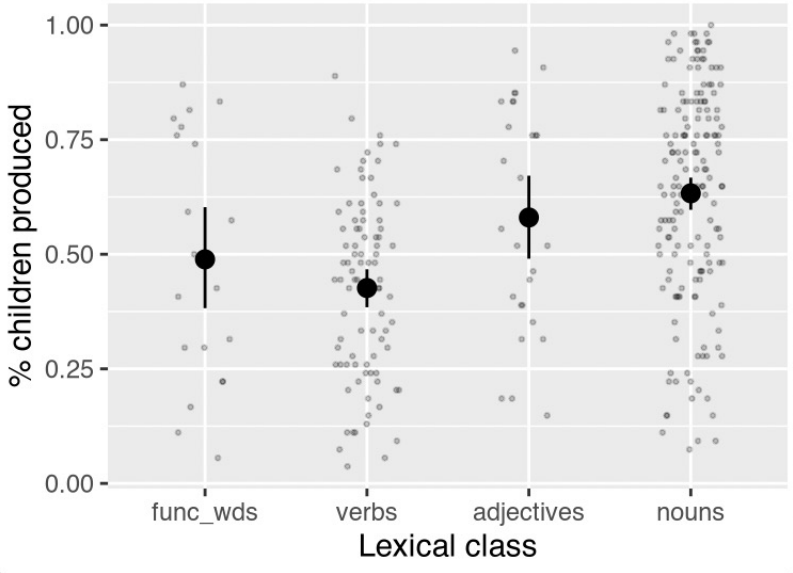
- related: *y*-axis should reflect what's being modeled
  - Here: log(VOT) ⇒ log scale
  - *y* = 0/1 ⇒ probabilities/log-odds

Data: `french_cdi_24`

- from Wordbank (wordbank.stanford.edu : Frank et al., 2021)
  - CDI questionnaire database

- Quebec French-learning children at 24 months: 45 children, 664 words
- *y*: can child produce this word? (0/1)

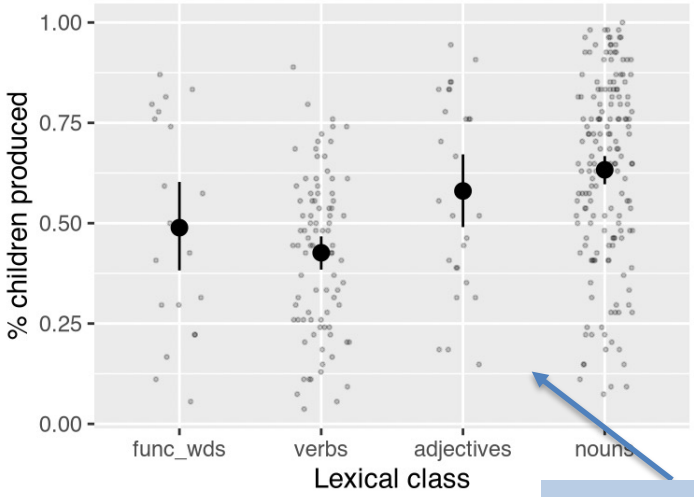
- Predictors:
  - Word-level:
    lexical class ,
    word frequency, length, ...
  - Child-level: gender…

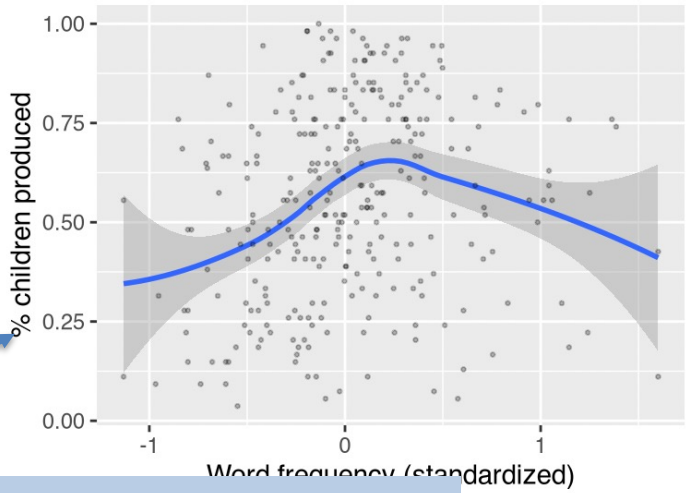


RQs: noun bias? function-word bias?

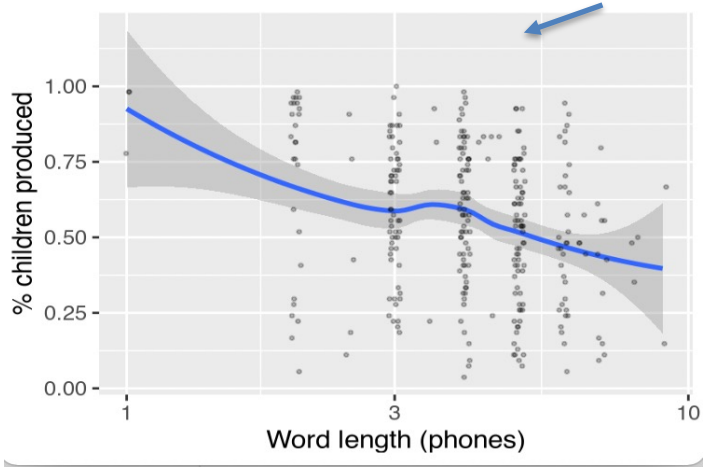
RMLD: 7.1.2, 9.7.2

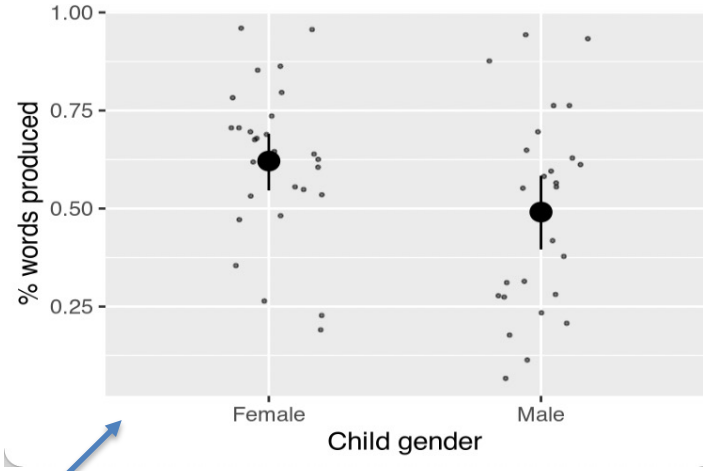
- Empirical effects:





word-level predictors: one point per word





Child-level predictor: one point per child

* Restricted to 299 words for which frequency/length/MLU defin

Model predictions

- Crucial for unpacking model results

- Predicted *y* as 1+ predictor(s) are varied.
- Other predictors can be:
  - Averaged over ("marginalized"): gives expected marginal mean
  - Held at some value

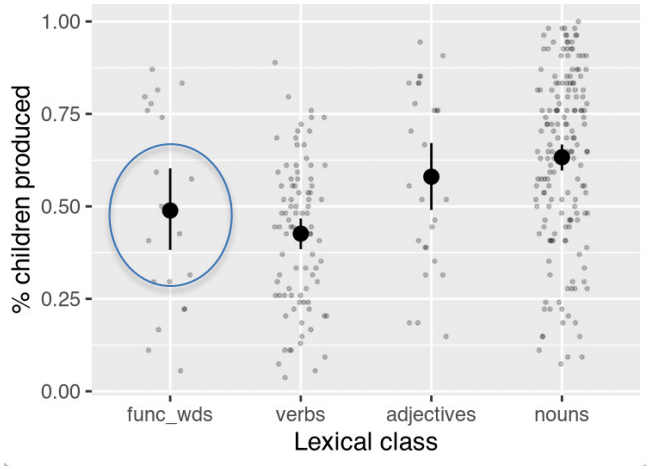
- Can make model predictions:
  - "by hand" (e.g. `predict()`)
  - Using existing packages
    (e.g. ggeffects, Effects, modelbased, marginaleffects)

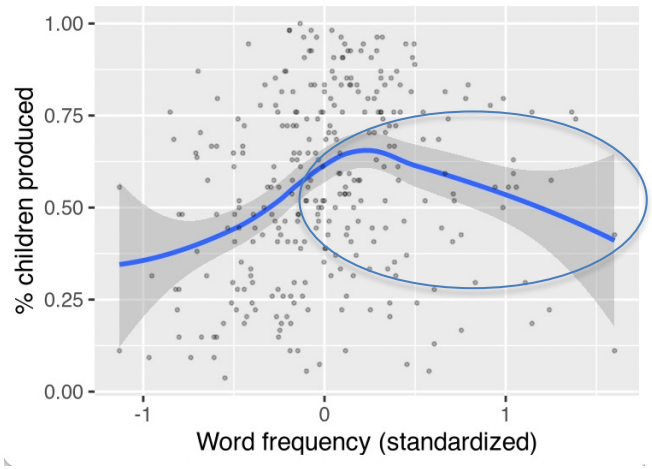
- Various choices need to be made, especially once models are more complex (mixed-effects)
  - Packages will make them for you, but beware not understanding what you're doing

Examples of both approaches: RMLD: 4.5.2, 6.7.3, 8.8, 9.5

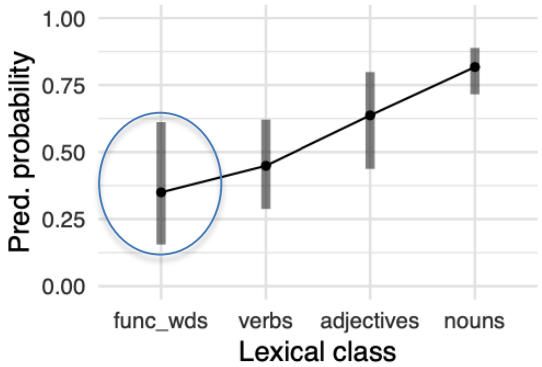
Empirical plots vs. model predictions

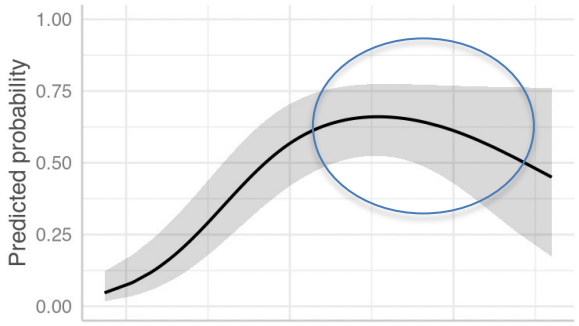
- `lexical_class`, `frequency` effects: unexpected





- predictions from a mixed-effects model:

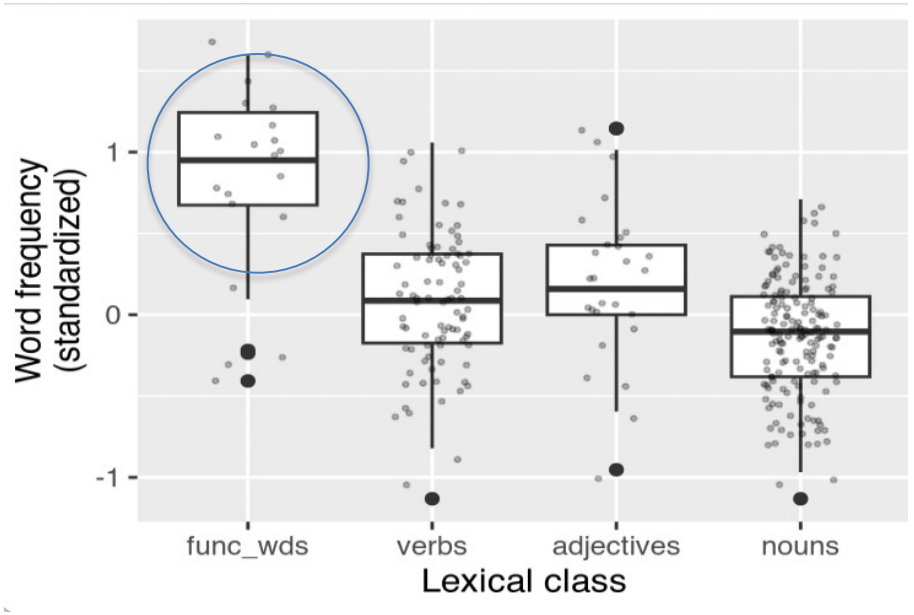




Uncertainty about high-frequency and function words

Empirical plots vs. model predictions

- Explanation: lexical class/frequency relationship
  - Captured by the model



- model predictions >> empirical plots, when the difference matters

mixed-effects model formula: on slide in part 3

Questions

Part 2: Variable selection

1. Trade-offs between different possible models — RMLD 5.8
2. Model comparison — RMLD 5.9
3. Choosing a set of predictors ("variable selection") — RMLD 5.10

Dataset: `vot_michael`

- For simplicity, take a subset of `vot` data for Part 2

  $n = 593$

  – one `speaker`

  – one row per `word`

- Lets us use just linear regression, no random effects

  – Satisfies independence assumptions

What is a regression model?

- Models relationship between:
  - Response / dependent variable: y
  - Predictors / indep. variables: $x_1, \ldots, x_k$
  - For n observations (response/predictor pairs)
- Often: linear model

$$C(y|x_1, \ldots, x_k) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k.$$

regression coefficients

RMLD: 4.2

Why are we building a regression model?

- Explanation: accurately estimate coefficients
  - Ex: `voicing` effect on VOT (in msec)

$\hat{\beta}_0 \cdot \hat{\beta}_1$

- Prediction: accurately predict response for new data
  - Ex: predict VOT in unseen data (max. R^2)

- Description: of response/predictor relationship
  - Ex: exploratory study

- Goal of analysis, as well as research questions, affects methodological choices

- Often in lang. sciences, explanation is the (unstated) goal

Trade-offs between models

- Regression analysis requires choosing between different possible models
  - "What predictors should I include?"
- Especially hard for corpus data, where there are many possible predictors
- Useful to have a high-level understanding of trade-offs in choosing one model vs. another
  - Bias/variance
  - Overfitting/underfitting

Bias/variance trade-off

- If our goal is accurate <u>coefficient estimates</u>
  - Especially effects of interest, corresponding to RQs – call this β

corpus data: **many** possible predictors, often don't know which are important

- Omit important predictors: estimate of β is biased
  - = "wrong", increased Type I error

- Add unimportant predictors: estimate of β is imprecise
  - = high standard error, low power

- Example:
  - y ~ x without controlling for 5 confounds
  - y ~ x1*x2*x3*x4, when in reality only x1 and x2 matter.

Overfitting vs. underfitting

- Closely-related tradeoff, if goal is <u>prediction</u>
  - Example: small subset of `vot` data (n = 50)
- RQ: effect of `voicing`
- Three possible plausible models:

"just make a boxplot of voicing vs. VOT"

- Model 1: `voicing` only ($k = 1$)
- Model 2: add five other plausible predictors, based on previous work ($k = 6$)
- Model 3: add all two-way interactions ($k = 16$)

"just throw everything into the model"

```
ou_mod_1 <- lm(log_vot ~ voicing, data = vot_train)
ou_mod_2 <- update(ou_mod_1, . ~ (voicing + speaking_rate +
    foll_high_vowel + cons_cluster + log_corpus_freq + stress))
ou_mod_3 <- update(ou_mod_1, . ~ (voicing + speaking_rate +
    foll_high_vowel + cons_cluster + log_corpus_freq + stress)^2)
```

Overfitting vs. underfitting

- Metrics to quantify overfitting
  - Cross-validation, data-splitting, optimism-adjusted R^2
- In this case:
  - Model 1: adj. R^2 = 0.78 – underfits
  - Model 2: adj. R^2 = 0.79 – "best"
  - Model 3: adj. R^2 = 0.63 – overfits

(Harrell 2015, 5.3.5; Baayen 2008)

- Upshot, whether our goal is estimation or prediction
  - too many/few predictors has consequences
  - what the "right predictors" are depends on the data and research questions

Heuristics for variable selection

- Before proceeding to quantitative methods: important rules of thumb for "what variables should I include?"

- Divide-by-15 rule
  - $n/15$ predictors ⇒ overfitting unlikely
  - for logistic regression, n = less common case

- Assuming goal is explanation, need terms that:
  - Directly test RQs (regardless of significance)
  - Have large effects on y (regardless of RQs)

(Harrell 2015, 4.4; Baguely 2012, 12.4.1)

Heuristics for variable selection

- Examples:
  - Don't do a linear regression with 10 predictors for 50 data points
  - If RQ involves "effect of `frequency`", don't drop `frequency` term
  - For data with 20 participants, a mixed-effects model shouldn't include 5 participant-level predictors
  - When modeling VOT, always control for speaking rate, place of articulation
    - These have v. large effects on VOT

Questions

Model comparison

- Quantitative methods for testing a set of predictors
  - Needed for variable selection

- Case 1: nested. e.g.

$$M_1 : \texttt{RTlexdec} = \beta_0 + \beta_1 \cdot \texttt{WrittenFrequency} + \varepsilon$$

$$M_2 : \texttt{RTlexdec} = \beta_0 + \beta_1 \cdot \texttt{WrittenFrequency} + \beta_2 \cdot \texttt{Familiarity} + \varepsilon$$

- Methods differ by model type
  - Lin regression: *F*-test / ANOVA - linear regression
  - Likelihood ratio rest – logistic regression
  - …

Model comparison

- Case 2: non-nested, e.g.

```
nn_m1 <- lm(RTlexdec ~ WrittenFrequency + Familiarity, english_40)
nn_m2 <- lm(RTlexdec ~ WrittenFrequency + LengthInLetters, english_40)
nn_m3 <- lm(RTlexdec ~ Familiarity + LengthInLetters, english_40)
```

- Can use information criteria

$$AIC = 2k - 2\log(L),$$

$$BIC = k\log(n) - 2\log(L),$$

$$AICc = 2k\left[1 + \frac{k+1}{n-k-1}\right] - 2\log(L)$$

All model comparison methods trade off model likelihood (L) vs. size (k)

Use instead of AIC for small sample size (e.g. n/k < 40)

Model comparison

- Different methods can give different results

```
ou_mod_1 <- lm(log_vot ~ voicing, data = vot_train)
ou_mod_2 <- update(ou_mod_1, . ~ (voicing + speaking_rate +
    foll_high_vowel + cons_cluster + log_corpus_freq + stress))
ou_mod_3 <- update(ou_mod_1, . ~ (voicing + speaking_rate +
    foll_high_vowel + cons_cluster + log_corpus_freq + stress)^2)
```

- Optimism-adjusted R^2: chose model 1
  - Using AIC, AICc : same
- Using BIC, *F*-tests: choose model 2

- Important to be aware of this, but there is no "right" method independent of context
  - I suspect for corpus data, AIC(c) often makes sense

RMLD: 5.9, Box 5.10

Variable selection

- Some right ways, many wrong ways
  - Thus: crucial to explain what you are doing and why in any writeup
- Possible methods include:
- choose model from a fixed set of 'candidate models'
  - using F test, AIC, BIC, adjusted R^2, etc.
  - Requires clear 'candidate models'. Sometimes unclear.
- Fully automatic approaches: common
  - Especially: stepwise variable selection
  - dangerous!
- Holistic approaches :
  - domain knowledge and research questions
  - model comparison

Stepwise model comparison

- Most common variable selection procedure, e.g. using R `step()` function
- "[stepwise variable selection]… violates every principle of statistical estimation and hypothesis testing" (Harrell, 2015)
  - One problem: resulting *p*-values greatly inflated
- Demonstration: RMLD 5.10.1 / today's code file
- Not recommended

Holistic approaches

- One possible holistic approach: the Gelman & Hill method

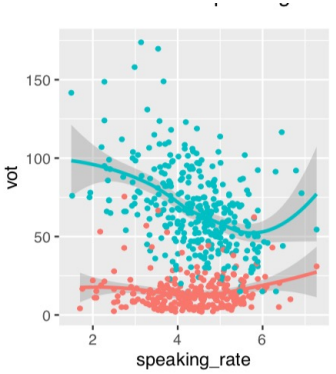
1. Include all predictors that, **for substantive reasons**, are expected to be important, e.g., they are part of the study's design or are covariates known from previous work to have large effects on the response.[13]
2. For predictors with large effects, consider interactions as well.
3. Now decide whether to exclude predictors, one at a time:
   - Not significant, coefficient has expected sign: Consider leaving in.
   - Not significant, wrong sign: Remove.
   - Significant, wrong sign: Think hard if something's wrong.
   - Significant, right sign: Keep in.

- Requires some domain knowledge ("right" vs "wrong" sign)

RMLD: 5.10.2

Example: Gelman & Hill method

- Data: `vot_michael`
- RQ: what factors modulate the `voicing` contrast?
- Effects of interest: interactions with `voicing`
- predictors: `voicing, speaking_rate, foll_high_vowel, cons_cluster, place, and log_corpus_freq`



- From previous work:
  - direction of effect on `vot` known: all predictors
  - known to modulate `vot`: just `speaking_rate`

Example: Gelman & Hill method

- Thus: first model includes
  - main effects of all predictors
  - `voicing:speaking_rate` interaction

```
gh_mod_1 <- lm(log_vot ~ voicing * speaking_rate + foll_high_vowel +
    cons_cluster + log_corpus_freq + place, data = vot_michael)
```

```
##                   term estimate std.error statistic  p.value
## 1          (Intercept)    3.375     0.018     183.6  0.0e+00
## 2              voicing    1.569     0.038      41.3 1.8e-175
## 3       speaking_rate   -0.147     0.037      -3.9  9.9e-05
## 4      foll_high_vowel    0.116     0.040       2.9  4.1e-03
## 5        cons_cluster    0.454     0.043      10.5  9.2e-24
## 6      log_corpus_freq   -0.075     0.038      -2.0  4.9e-02
## 7               place   -0.513     0.038     -13.4  5.2e-36
## 8 voicing:speaking_rate   -0.351     0.074      -4.7  2.8e-06
```

highest effect sizes => consider two-way interactions

- Expected direction of ixns:
  - `foll_high_vowel:voicing` : positive – some voiceless stops aspirated before high vowels
  - `voicing:cons_cluster` : unclear
  - etc.
- Add all ixns to model:

```
gh_mod_2 <- update(gh_mod_1, . ~ . +
    (voicing + foll_high_vowel + cons_cluster + place)^2)
```

- Then:
  - Most have expected sign: leave in
  - `voicing:cons_cluster` : significant, and expected sign unclear
    - Important for RQ, so leave in the model
  - etc.

- The final model:

```
##                      term estimate std.error statistic  p.value
## 1             (Intercept)    3.402     0.018     192.6  0.0e+00
## 2                 voicing    1.556     0.036      43.1 2.0e-183
## 3           speaking_rate   -0.160     0.035      -4.5  7.2e-06
## 4         foll_high_vowel    0.103     0.038       2.7  7.2e-03
## 5            cons_cluster    0.452     0.041      10.9  1.8e-25
## 6         log_corpus_freq   -0.078     0.036      -2.2  3.1e-02
## 7                   place   -0.488     0.036     -13.4  4.6e-36
## 8  voicing:speaking_rate   -0.343     0.070      -4.9  1.3e-06
## 9   voicing:cons_cluster   -0.304     0.083      -3.7  2.5e-04
## 10         voicing:place    0.558     0.072       7.7  5.7e-14
## 11 foll_high_vowel:place    0.162     0.078       2.1  3.7e-02
```

- RQ answer: these factors modulate voicing contrast

RMLD: 5.10.3

Other holistic approaches

- G&H is one approach integrating
  - Domain knowledge
  - Research questions
  - Quantitative model comparison

- I recommend holistic methods
  - Science is primary (RQs, domain knowledge) : whether to add/drop terms shouldn't just be done automatically
  - … but this does introduce subjectivity (Roettger, 2019)

- Drawback of G&H approach: requires extensive domain knowledge
  - Corpus phonetics/phonology data: often, "expected effects" unclear and there are many possible predictors

Other holistic approaches

- My usual practice (not in book):
  1. Terms directly related to RQs: stay in
  2. Controls / other terms: combination of

     2a: domain knowledge

     2b: model selection (e.g. *F*-test) and exploratory plots
  3. Think carefully about implications of (2) (especially 2b) for what terms in (1) say about RQs
     - Type I, II errors?
  4. Explain what we're doing and why

Example: Bang et al. (2018)

ELSEVIER

Journal of Phonetics

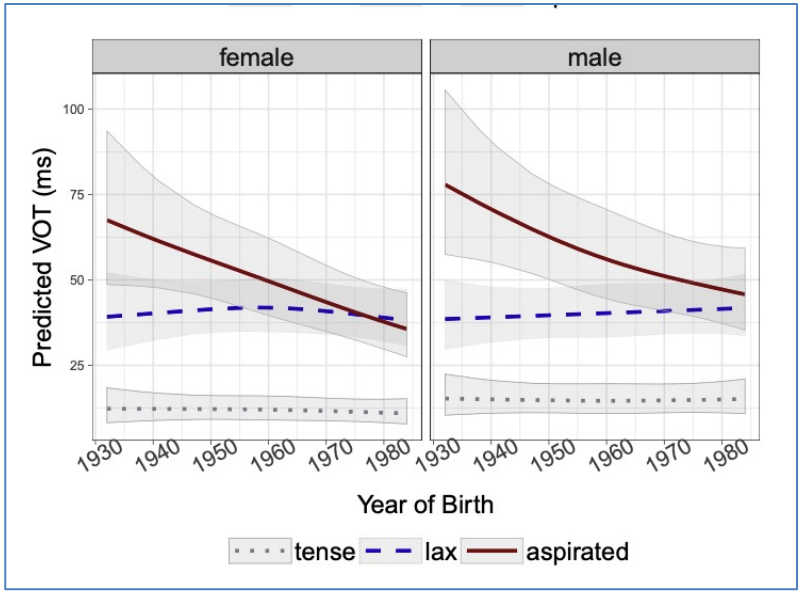
journal homepage: www.elsevier.com/locate/Phonetics

Research Article

The emergence, progress, and impact of sound change in progress in Seoul Korean: Implications for mechanisms of tonogenesis

Hye-Young Bang [a,*], Morgan Sonderegger [a,b], Yoonjung Kang [c,d], Meghan Clayards [a,b], Tae-Jin Yoon [e]

- Focus: change over time in VOT*
- Known from previous work: effect of `yob`, `gender`, `laryngeal` class
- RQs: effect of word `frequency`, following `vowel_height` in change



* also f0, not considered in this example

- The statistical model:

Fixed effect coefficient	VOT				
	Estimate	SE	df	*t*	$P(>t)$
Intercept	3.409	0.028	97.544	122.569	<0.001
YOB′	−0.034	0.012	122.56	−2.798	0.006
YOB″	0.003	0.058	115.529	0.052	0.959
LARYNGEAL1(tense vs. nontense)	1.242	0.063	78.149	19.705	<0.001
LARYNGEAL2(lax vs. aspirated)	0.221	0.056	75.249	3.909	<0.001
HEIGHT(high)	0.134	0.062	67.434	2.139	0.036
FREQUENCY	−0.11	0.05	65.925	−2.2	0.031
POSITION1(initial vs. medial)	0.1	0.035	97.702	2.848	0.005
POSITION2(short vs. longer pause)	0.001	0.018	160.767	0.066	0.948
POSITION3(medial vs. long pause)	−0.025	0.02	137.155	−1.253	0.212
RATE DEVIATION	−0.007	0.015	5591.925	−0.439	0.661
GENDER(male)	0.127	0.027	124.629	4.706	<0.001
PLACE1(labial vs. non-labial)	0.123	0.048	68.565	2.575	0.012
PLACE2(alveolar vs. velar)	0.314	0.062	68.326	5.1	<0.001
SPEAKER MEAN RATE	−0.021	0.035	120.983	−0.605	0.546
YOB′:LARYNGEAL1	−0.061	0.017	124.312	−3.561	0.001
YOB′:LARYNGEAL2	−0.118	0.013	114.175	−8.826	<0.001
YOB″:LARYNGEAL1	0.037	0.09	109.225	0.408	0.684
YOB″:LARYNGEAL2	0.173	0.069	106.083	2.505	0.014
YOB′:HEIGHT	−0.011	0.013	84.856	−0.858	0.393
YOB′:FREQ.	−0.015	0.01	71.298	−1.492	0.14
YOB″:HEIGHT	0.154	0.063	85.177	2.449	0.016
YOB″:FREQ.	−0.041	0.048	67.411	−0.854	0.396
LARYNGEAL1:HEIGHT	−0.233	0.152	65.32	−1.527	0.132
LARYNGEAL2:HEIGHT	0.326	0.103	66.717	3.157	0.002
LARYNGEAL1:FREQ.	0.193	0.114	66.345	1.686	0.096
LARYNGEAL2:FREQ.	−0.185	0.109	64.986	−1.695	0.095
LARYNGEAL1:POSITION1	−0.037	0.067	204.377	−0.553	0.581
LARYNGEAL2:POSITION1	0.158	0.092	65.725	1.709	0.092
LARYNGEAL1:POSITION2	−0.031	0.047	139.183	−0.657	0.512
LARYNGEAL2:POSITION2	0.009	0.032	5565.071	0.296	0.768
LARYNGEAL1:POSITION3	0.048	0.048	127.345	1.015	0.312
LARYNGEAL2:POSITION3	0.075	0.038	5617.112	1.989	0.047
LARYNGEAL1:RATE DEV.	−0.079	0.041	633.019	−1.922	0.055
LARYNGEAL2:RATE DEV.	−0.005	0.026	4381.899	−0.177	0.86
LARYNGEAL1:GENDER	−0.155	0.046	123.46	−3.374	0.001
LARYNGEAL2:GENDER	0.16	0.036	118.285	4.486	<0.001
YOB′:GENDER	0.029	0.019	112.736	1.521	0.131
YOB″:GENDER	0.06	0.114	110.257	0.524	0.601
YOB′:LARYNGEAL1:HEIGHT	0.068	0.031	71.561	2.151	0.035
YOB′:LARYNGEAL2:HEIGHT	0.024	0.022	63.715	1.105	0.273
YOB′:LARYNGEAL1:FREQ.	0.006	0.025	78.795	0.233	0.816
YOB′:LARYNGEAL2:FREQ.	−0.005	0.021	55.19	−0.241	0.81
YOB″:LARYNGEAL1:HEIGHT	−0.169	0.147	69.039	−1.151	0.254
YOB″:LARYNGEAL2:HEIGHT	−0.073	0.101	60.468	−0.723	0.472
YOB″:LARYNGEAL1:FREQ.	0.103	0.118	72.869	0.866	0.389
YOB″:LARYNGEAL2:FREQ.	0.023	0.094	57.789	0.249	0.804

How did we arrive at this??

FULL MODELS

Intercept
YOB′
YOB″
LARYNGEAL1(tense vs. nontense)
LARYNGEAL2(lax vs. aspirated)
HEIGHT(high)
FREQUENCY
POSITION1(initial vs. medial)
POSITION2(short vs. longer pause)
POSITION3(medial vs. long pause)
RATE DEVIATION
GENDER(male)
PLACE1(labial vs. non-labial)
PLACE2(alveolar vs. velar)
SPEAKER MEAN RATE
YOB′:LARYNGEAL1
YOB′:LARYNGEAL2
YOB″:LARYNGEAL1
YOB″:LARYNGEAL2
YOB′:HEIGHT
YOB′:FREQ.
YOB″:HEIGHT
YOB″:FREQ.

LARYNGEAL1:HEIGHT
LARYNGEAL2:HEIGHT
LARYNGEAL1:FREQ.
LARYNGEAL2:FREQ.
LARYNGEAL1:POSITION1
LARYNGEAL2:POSITION1
LARYNGEAL1:POSITION2
LARYNGEAL2:POSITION2
LARYNGEAL1:POSITION3
LARYNGEAL2:POSITION3
LARYNGEAL1:RATE DEV.
LARYNGEAL2:RATE DEV.
LARYNGEAL1:GENDER
LARYNGEAL2:GENDER
YOB′:GENDER
YOB″:GENDER
YOB′:LARYNGEAL1:HEIGHT
YOB′:LARYNGEAL2:HEIGHT
YOB′:LARYNGEAL1:FREQ.
YOB′:LARYNGEAL2:FREQ.
YOB″:LARYNGEAL1:HEIGHT
YOB″:LARYNGEAL2:HEIGHT
YOB″:LARYNGEAL1:FREQ.
YOB″:LARYNGEAL2:FREQ.

no three-way interactions unrelated to RQs considered

previous work

(e.g. Kang, 2014)

controls

(domain knowledge about Korean, VOT)

research questions

- Our approach was:
  - Fit one large model
  - Justify its structure from RQs and domain knowledge
  - Report the results

- Notes:
  - All possible interactions not included
    - Only three-way interactions motivated by RQs
    - Trying to strike a balance between including everything important for RQs and overfitting
  - Terms testing RQs are left in regardless of "significance"
    - Ex: all `frequency` interactions : $p > 0.05$
  - These kind of choices may be questioned during review

Questions

Part 3: Unpacking results

1. Multi-level factors / contrast coding RMLD 7.2
2. post-hoc tests, expected marginal means/trends RMLD 7.3
3. interactions RMLD 7.4

Examples for mixed-effects models: RMLD 9.7

Unpacking results

- Should be done with a combination of
  - Visual exposition: especially model predictions
  - Hypothesis tests

Part 1

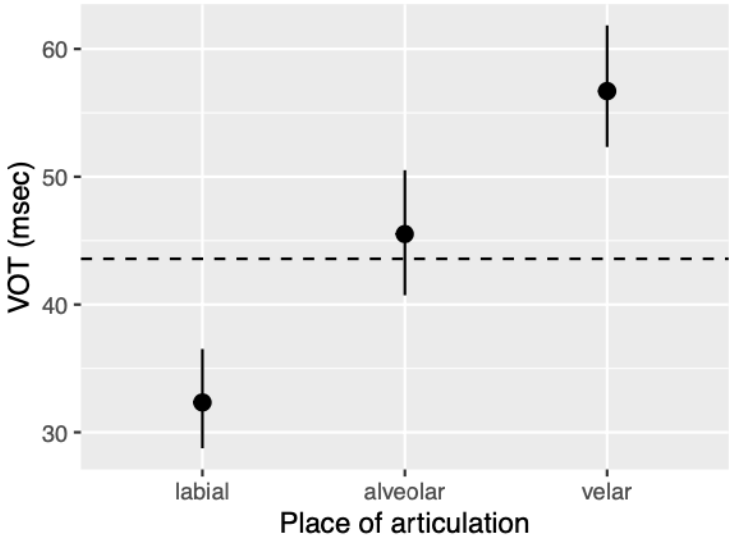
- This gets harder once we use realistic models with
  - Factors with 3+ levels
  - Interactions
  - Nonlinear effects

focus today – especially common for corpus data

RMLD 7.5, 9.7

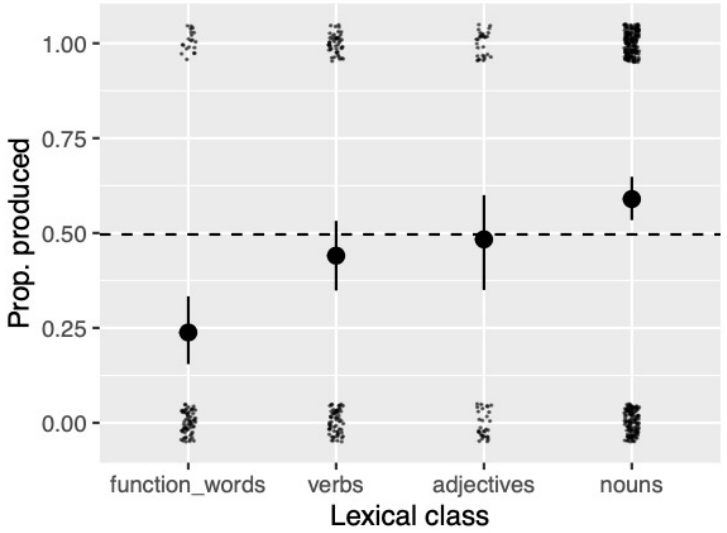
Multi-level factors

- Factor *x* with *k* = 3+ levels
- Running examples:



`vot_michael` data
factor: `place` (*k*=3)

linear regression



`french_cdi_24` data – one child
factor: `lexical_class` (*k*=4)

logistic regression

Contrast coding

- "How does *x* affect the response *y*?"
  - In terms of level 1, 2, … means: μ_1 , μ_2, …
- Code as *k-1* contrasts
  - = regression coefficients for *x*
  - (+ 1 intercept)
- Different contrast coding schemes ⇒ different interpretations of
  - Intercept
  - Coefficients

- By the pre-workshop survey, I'm not going to cover contrast coding in detail

RMLD 7.2 (also: Schad et al. ,2020)

Contrast coding: the short version

- Advice:
  1. code binary predictors as -0.5, 0.5 (not 0/1)
  2. use orthogonal contrasts
     - = independent information
  3. use centered contrasts
     - = "center all predictors" advice, for factors

- R defaults do not follow (1)-(3).
- Helmert contrasts are a good default

- Weighted contrasts and custom contrasts are nice and underused options for corpus data

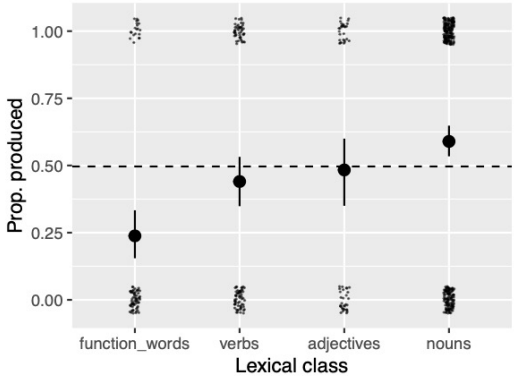
Choosing a coding scheme

- Every multi-level factor must be contrast-coded
- Choose by:
  - ~~– Whatever R does by default (treatment coding)~~
  1. Primary: theory (matching model structure to RQs)
  2. Practical considerations
     (easier model interpretation, fitting)
     - Especially important: centered

- Sensible default: Helmert contrasts
- For #1: a useful and underused option is custom contrasts

Custom contrasts

- `french_cdi_24`: RQs about `lexical_class` focus on:
  1. Noun bias: *nouns* vs. *verbs/adjectives*
  2. Function word bias: *FWs* vs. *verbs/adjectives*
- Less important
  3. *verbs* vs. *adjectives*



- contrasts for `lexical_class`
  - Intercept: 0.25, 0.25, 0.25, 0.25 (unweighted mean)
  - Contrast 1 (noun bias): 0, –0.5, –0.5, 1 (NOUNS vs. average of VERBS/ADJECTIVES)
  - Contrast 2 (function-word bias): –1, 0.5, 0.5, 0 (Verb/adj VS. FUNCTION_WORDS)
  - Contrast 3 (within-predicates): 0, –1, 1, 0 (VERBS VS. ADJECTIVES)

Custom contrasts

Contrasts for `lexical_class`:

- Intercept: 0.25, 0.25, 0.25, 0.25 (unweighted mean)
- Contrast 1 (noun bias): 0, –0.5, –0.5, 1 (NOUNS vs. average of VERBS/ADJECTIVES)
- Contrast 2 (function-word bias): –1, 0.5, 0.5, 0 (predicates vs. FUNCTION_WORDS)
- Contrast 3 (within-predicates): 0, –1, 1, 0 (VERBS VS. ADJECTIVES)

Verb/adj

Logistic regression:
`produces ~ lexical_class`

95% confidence interval:
doesn't contain 0

```
## # A tibble: 4 x 5
##   term                estimate conf.low conf.high
## 1 (Intercept)            -0.28    -0.49    -0.069
## 2 lexical_class_custom1   0.52     0.13     0.906
## 3 lexical_class_custom2   1.01     0.43     1.621
## 4 lexical_class_custom3   0.17    -0.46     0.807
```

Coefficient estimates directly address research questions

(significant "noun bias", "function word bias")

Omnibus + post-hoc tests

- "How does *x* affect the response *y*?"
  - In terms of level 1, 2, … means: μ_1 , μ_2, …
- Option 1: contrast coding
- Option 2
  - Omnibus test ("does *x* have any effect?")
  - Post-hoc tests ("which levels of *x* differ?")

Omnibus test

- Model comparison:
  1. With *x* (factor with *k* levels)
  2. Without *x*

- *F*-test (lin. reg.) or likelihood-ratio test (log. reg.)

```
anova(vot_cc_mod_1, update(vot_cc_mod_1, . ~ 1))
...
## Model 1: vot ~ place
## Model 2: vot ~ 1
##   Res.Df    RSS Df Sum of Sq    F  Pr(>F)
## 1    590 604254
## 2    592 666243 -2    -61988 30.3 3.1e-13
```

`place` significantly contributes ($p<0.001$)

- Note the difference from regression coeffs – none answers "does `place` contribute?"

```
vot_cc_mod_1 <- lm(vot ~ place, data=vot_michael)
vot_cc_mod_1 %>% tidy()
## # A tibble: 3 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       32.3      2.07     15.7  1.92e-46
## 2 placealveolar     13.2      3.19      4.13 4.17e- 5
## 3 placevelar        24.4      3.16      7.72 4.93e-14
```

- Omnibus test does not depend on contrast coding scheme
  - True for model comparison more generally

`place` coded with sum contrasts

```
anova(update(vot_cc_mod_1, . ~ place_sum),
      update(vot_cc_mod_1, . ~ 1) # intercept-only model
)
```

Result: same as previous slide (where treatment contrasts used)

Post-hoc tests

- Test differences between (combinations of) levels of *x*, after fitting the model

- Similar to contrast coding, but:
  - Test any # of differences
  - If we test more than *k* - 1 differences, adjust for multiple comparisons

- Useful to think of as:
  1. Compute estimated marginal means of *y*
  2. Test differences in EMMs

`emmeans` package

Example: pairwise comparisons

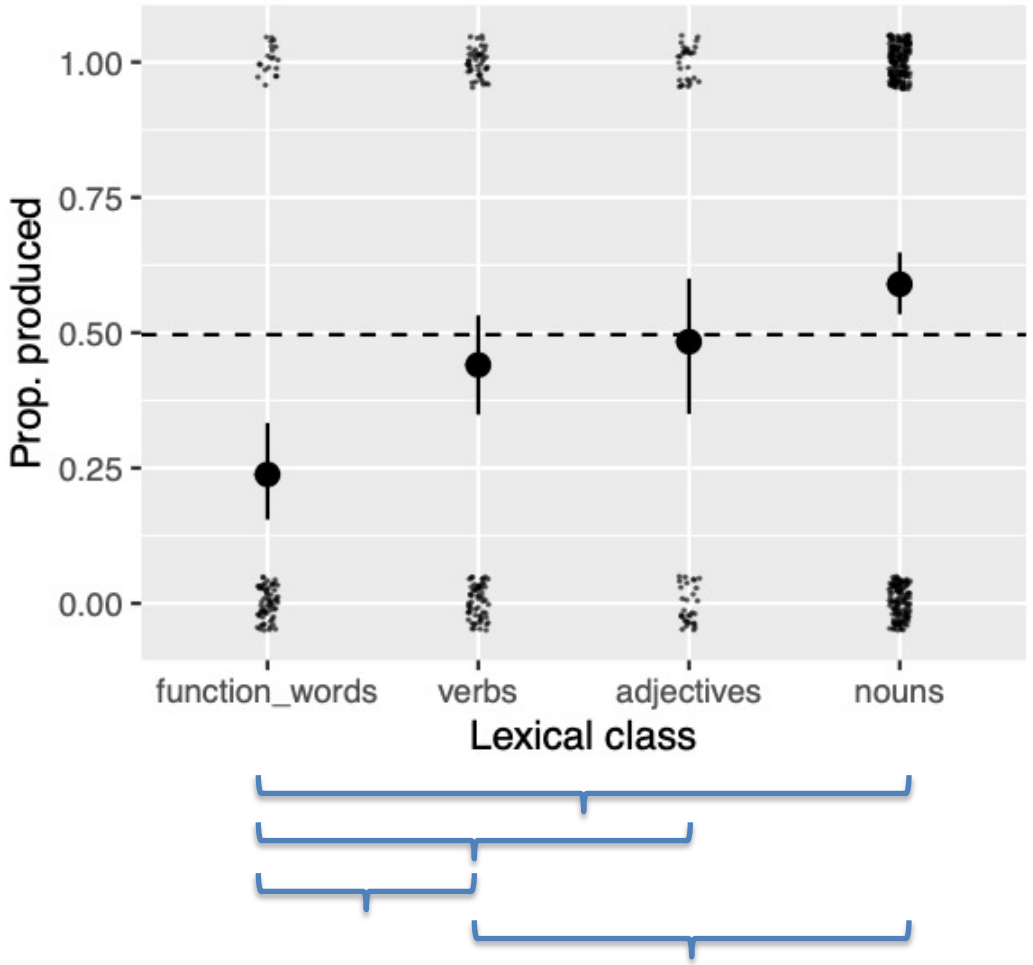
- Most common application of post-hoc tests
  - After "does *x* matter?", tests "which levels of *x* differ?"

```
emm_cdi <- emmeans(cdi_cc_mod_1, ~lexical_class)
```

```
contrast(emm_cdi, 'pairwise')
##  contrast                    estimate    SE  df z.ratio p.value
##  function_words - verbs        -0.923 0.321 Inf -2.880  0.0210
##  function_words - adjectives   -1.096 0.364 Inf -3.010  0.0140
##  function_words - nouns        -1.525 0.281 Inf -5.420  <.0001
##  verbs - adjectives            -0.173 0.322 Inf -0.540  0.9500
##  verbs - nouns                 -0.602 0.225 Inf -2.670  0.0380
##  adjectives - nouns            -0.429 0.283 Inf -1.510  0.4290
##
## Results are given on the log odds ratio (not the response) scale.
## P value adjustment: tukey method for comparing a family of 4 estimates
```

RMLD 7.3.2

Example: pairwise comparisons



FW < V, A, N

V < N

($p < 0.05$)

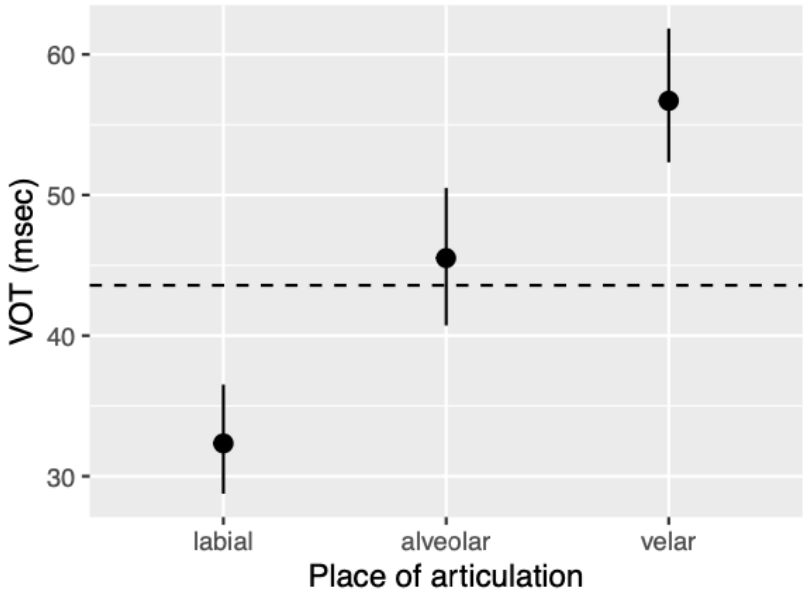
RMLD 7.3.2

- Simpler example:

```
vot_emm <- emmeans(vot_cc_mod_1, ~place)
```

```
> pairs(vot_emm)
 contrast          estimate   SE  df t.ratio p.value
 labial - alveolar    -13.2 3.19 590 -4.129  0.0001
 labial - velar       -24.4 3.16 590 -7.722  <.0001
 alveolar - velar     -11.2 3.41 590 -3.284  0.0031

P value adjustment: tukey method for comparing a family of 3 estimates
```



labial < alveolar < velar

RMLD 7.3.2

- NB: emmeans functionality works for more complex models
  - Ex: marginal effect of `place` in our final `vot_michael` model from Part I

```
gh_mod_4 <- lm(log_vot ~ voicing*speaking_rate + voicing*cons_cluster +
    voicing*place + foll_high_vowel*place + log_corpus_freq, data = vot_michael)
```

```
> vot_emm_2 <- emmeans(gh_mod_4, ~place)
```

extract expected marginal means

```
> pairs(vot_emm_2)
 contrast          estimate     SE  df t.ratio p.value
 labial - alveolar  -0.4863 0.0445 579 -10.941  <.0001
 labial - velar     -0.4691 0.0499 579  -9.406  <.0001
 alveolar - velar    0.0173 0.0520 579   0.332  0.9410

Results are averaged over the levels of: voicing, cons_cluster, foll_high_vowel
P value adjustment: tukey method for comparing a family of 3 estimates
```

pairwise comparisons

automatically marginalizes over vars which interact with `place` to give "average effect"

Questions

Post-hoc tests: more

- Post-hoc tests don't depend on coding scheme
  - To understand effect of a factor *x* , can code useful contrasts or just use post-hoc tests
- Tradeoff: post–hoc test approach has lower power

- Post-hoc tests/EMM idea: more generally useful
  - Testing custom contrasts corresp. to RQs
  - Check how trends (continuous *x*) differ
  - Unpacking interactions

Custom post-hoc tests

- Similar to coding custom contrasts to capture RQs (slide 52), but using post-hoc tests:

```
cdi_cc_mod_1 <- glm(produces ~ lexical_class, data = french_cdi_24,
  family = "binomial")
```

1. fit model

```
emm_cdi <- emmeans(cdi_cc_mod_1, ~lexical_class)
```

2. extract expected marginal mean

```
interpVecs <- list(
  nounBias = c(0, -0.5, -0.5, 1),
  fwBias = c(-1, 0.5, 0.5, 0),
  verbVAdj = c(0, -1, 1, 0)
)
```

3. define custom contrasts

- Contrast 1 (noun bias): 0, –0.5, –0.5, 1 (NOUNS vs. average of VERBS/ADJECTIVES)
- Contrast 2 (function-word bias): –1, 0.5, 0.5, 0 (predicates vs. FUNCTION_WORDS)
- Contrast 3 (within-predicates): 0, –1, 1, 0 (VERBS VS. ADJECTIVES)

```
contrast(emm_cdi, method = interpVecs) %>% confint()
##  contrast estimate   SE  df asymp.LCL asymp.UCL
##  nounBias     0.52 0.20 Inf      0.13       0.9
##  fwBias       1.01 0.30 Inf      0.42       1.6
##  verbVAdj     0.17 0.32 Inf     -0.46       0.8
##
## Results are given on the log odds ratio (not the response) scale.
## Confidence level used: 0.95
```

4. perform post-hoc tests

RMLD 7.3.3

Custom post-hoc tests

- The general idea here is very useful in practice
  1. Compute estimated marginal means of *y*
  2. Test differences in EMMs

- Contrast coding can be difficult to implement or interpret, especially for more complex models

- It is always an option to just fit your model with any coding scheme, then use post-hoc tests to assess hypotheses of interest.
  - and correcting for multiple comparisons if you examine more than k – 1 tests.

- Note: it's often fine to skip the "omnibus test" part, unless it's important for your analysis
  - My impression: this is a holdover from ANOVA methodology (?)

Post-hoc trends

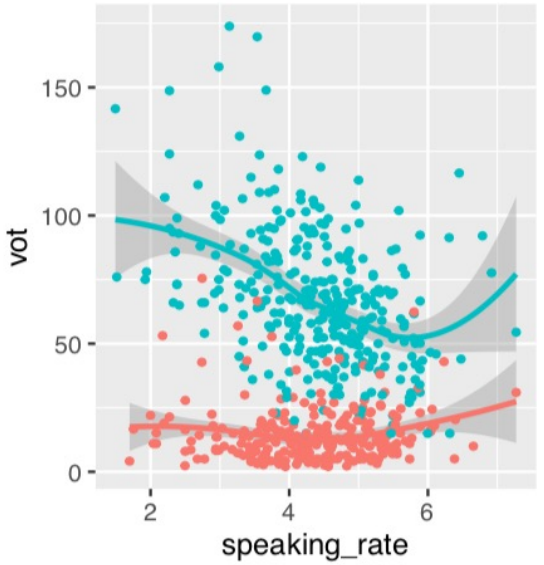
Ex: is there a `speaking_rate` effect for voiced and voiceless stops?

- What is a continuous predictor slope as a factor varied?

```
emtrends(gh_mod_4, ~voicing, var = "speaking_rate")
```

```
 voicing   speaking_rate.trend     SE  df lower.CL upper.CL
 voiced                0.00825 0.0274 579  -0.0456   0.0621
 voiceless            -0.16472 0.0247 579  -0.2132  -0.1162

Results are averaged over the levels of: cons_cluster, place, foll_high_vowel
Confidence level used: 0.95
```



The same, with p-values :

```
> emtrends(gh_mod_4, ~voicing, var = "speaking_rate") %>% summary(infer=c(FALSE, TRUE))
 voicing   speaking_rate.trend     SE  df t.ratio p.value
 voiced                0.00825 0.0274 579   0.301  0.7638
 voiceless            -0.16472 0.0247 579  -6.666  <.0001
```

This is different from "is there a significant interaction?"

Interpreting interactions with multi-level factors

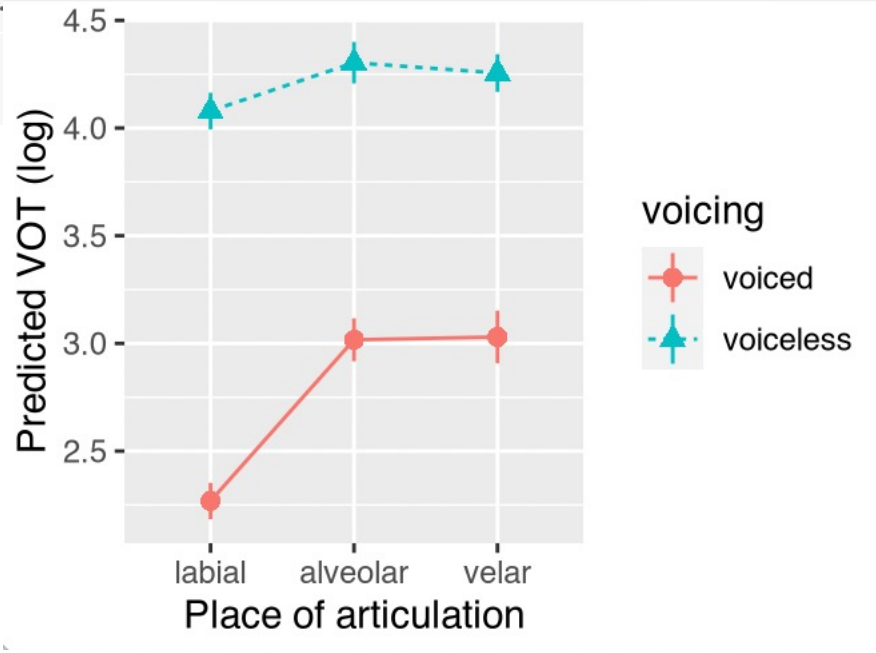
- Ex 1: two-way interaction

```
emmeans(gh_mod_4, ~ place * voicing)
##  place    voicing   emmean    SE  df lower.CL upper.CL
##  velar    voiced       3.0 0.062 579      2.9      3.2
##  alveolar voiced       3.0 0.051 579      2.9      3.1
##  labial   voiced       2.3 0.043 579      2.2      2.4
##  velar    voiceless    4.3 0.044 579      4.2      4.3
##  alveolar voiceless    4.3 0.049 579      4.2      4.4
##  labial   voiceless    4.1 0.043 579      4.0      4.2

## Results are averaged over the levels of: cons_cluster
## Confidence level used: 0.95
```

1. compute EMMs as `place & voicing` varied for `vot_michael` model

2. plot EMMs for qualitative understanding

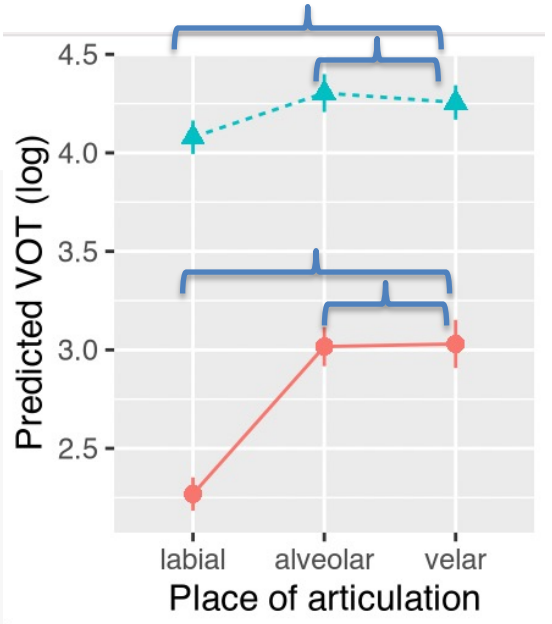


- suppose `place` is of primary interest
  - RQ is: do we always see the expected order (*labial* < *alveolar* < *velar*)?

3. Compute pairwise comparisons for place for each voicing level

```
emm <- emmeans(gh_mod_4, ~ place | voicing)
contrast(emm, "pairwise")
```

```
## voicing = voiced:
##  contrast          estimate    SE  df t.ratio p.value
##  velar - alveolar      0.01 0.078 579   0.200  0.9800
##  velar - labial        0.76 0.073 579  10.500  <.0001
##  alveolar - labial     0.75 0.060 579  12.500  <.0001
##
## voicing = voiceless:
##  contrast          estimate    SE  df t.ratio p.value
##  velar - alveolar     -0.05 0.064 579  -0.700  0.7400
##  velar - labial        0.18 0.060 579   2.900  0.0100
##  alveolar - labial     0.22 0.064 579   3.500  <.0001
##
```



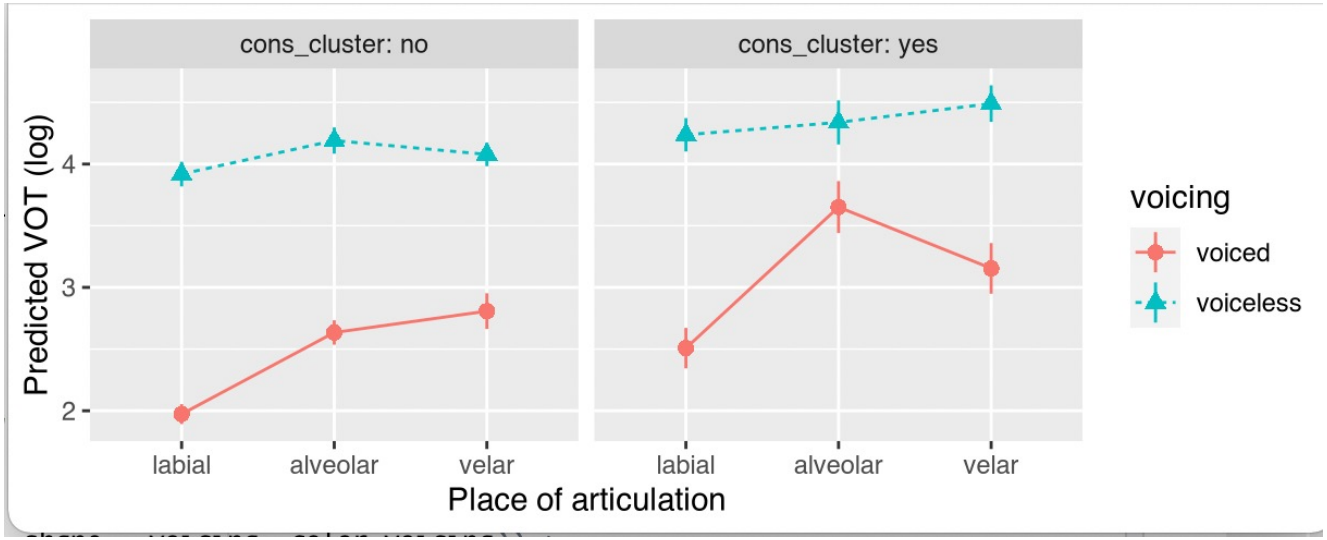
RQ answer: Regardless of voicing: labial < non-labial, but alveolar ~ velar

Interpreting interactions with multi-level factors

- Ex 2: three-way interaction
  - add a term to `vot_michael` model, just for this example

```
gh_mod_5 <- update(gh_mod_4, . ~ . +
        cons_cluster * place * voicing, data = vot_michael)
```

- Model predictions:



- Effect of `place` as `cons_cluster`, `voicing` varied:

```
gh_mod_5_emm <- emmeans(gh_mod_5, ~ place | voicing + cons_cluster)
contrast(gh_mod_5_emm, "pairwise")
## voicing = voiced, cons_cluster = no:
##  contrast          estimate    SE  df t.ratio p.value
##  velar - alveolar      0.17 0.089 575   1.900  0.1300
##  velar - labial        0.83 0.085 575   9.900  <.0001
##  alveolar - labial     0.66 0.065 575  10.100  <.0001
##
## voicing = voiceless, cons_cluster = no:
##  contrast          estimate    SE  df t.ratio p.value
##  velar - alveolar     -0.11 0.073 575  -1.500  0.2700
##  velar - labial        0.16 0.070 575   2.300  0.0600
##  alveolar - labial     0.27 0.074 575   3.700  <.0001
##
## voicing = voiced, cons_cluster = yes:
##  contrast          estimate    SE  df t.ratio p.value
##  velar - alveolar     -0.50 0.150 575  -3.300  <.0001
##  velar - labial        0.65 0.134 575   4.800  <.0001
##  alveolar - labial     1.14 0.136 575   8.400  <.0001
##
## voicing = voiceless, cons_cluster = yes:
##  contrast          estimate    SE  df t.ratio p.value
##  velar - alveolar      0.15 0.118 575   1.300  0.4000
##  velar - labial        0.25 0.102 575   2.500  0.0300
##  alveolar - labial     0.10 0.114 575   0.900  0.6500
##
## Results are averaged over the levels of: foll_high_vowel
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Qualitative summary:

labial < *velar* across word types

placement of *alveolar* inconsistent

easier than unpacking regression table!

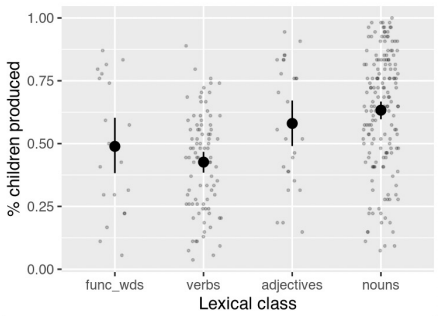
Example for mixed-effects models

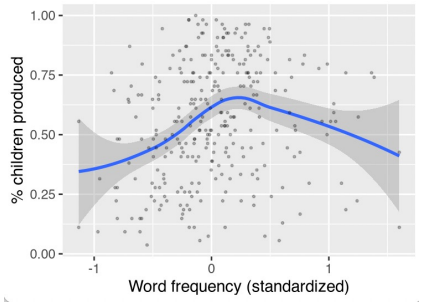
- Model of full `french_cdi_24` dataset:

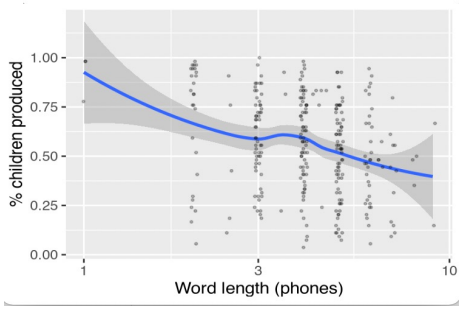
n = 16k

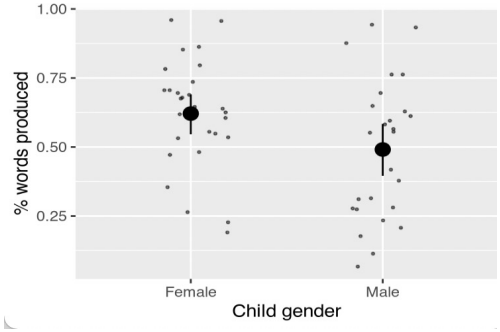
```
cdi_mod <- glmer(produces ~ lexical_class + sex + ns(freq.std, 2) +
  log_nphones.std + (1 + lexical_class | child) + (1 | definition),
data = french_cdi_24, family = "binomial",
control = glmerControl(optimizer = "bobyqa")
)
```

natural splines: good default for fitting nonlinear effects in an MEM









this model, which is not ideal (minimal random effects), takes 10-30 min to fit

good practice to not refit models, instead fit once and save/load as needed (e.g. for large corpus data)

RMLD 9.7

Post-hoc tests: MEMs

- Interpreting multi-level factors (for fixed effects) works similarly to non-MEMs:

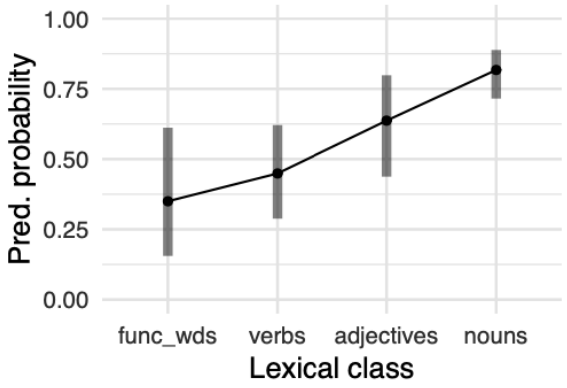
```
emm_cdi_1 <- emmeans(cdi_mod, ~lexical_class)
```

model predictions for an "average child"

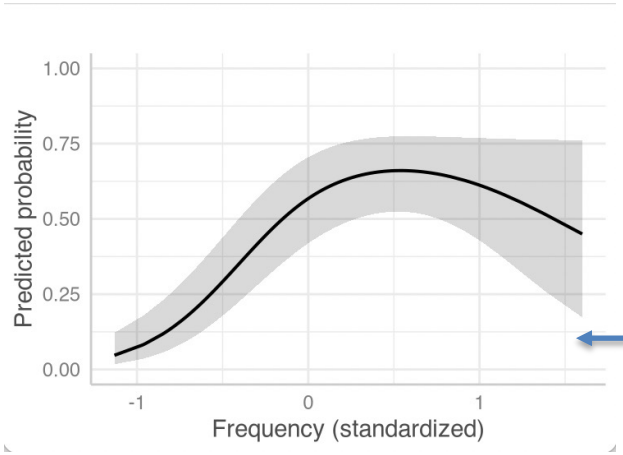
```
contrast(emm_cdi_1, method = "pairwise")
##  contrast                  estimate   SE  df z.ratio p.value
##  function_words - verbs       -0.41 0.55 Inf  -0.800  0.8700
##  function_words - adjectives  -1.18 0.56 Inf  -2.100  0.1400
##  function_words - nouns       -2.12 0.53 Inf  -4.000  <.0001
##  verbs - adjectives           -0.77 0.39 Inf  -2.000  0.2000
##  verbs - nouns                -1.70 0.27 Inf  -6.200  <.0001
##  adjectives - nouns           -0.94 0.37 Inf  -2.500  0.0600
##
## Results are averaged over the levels of: sex
## Results are given on the log odds ratio (not the response) scale.
## P value adjustment: tukey method for comparing a family of 4 estimates
```

function words, verbs < nouns

Model predictions: examples



`emmip()` from emmeans



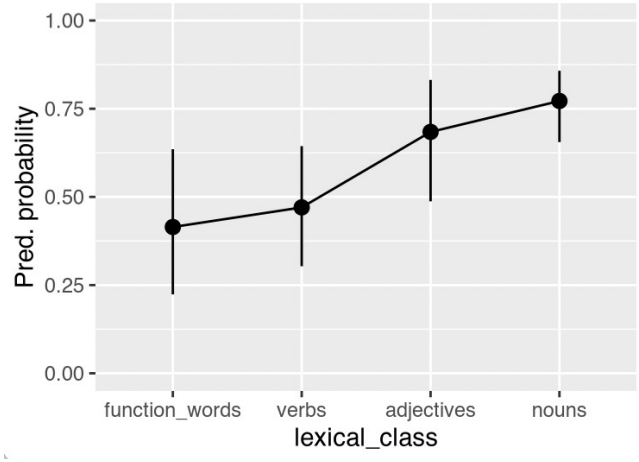
`ggemmeans()` from ggeffects

both for "average child", word length = 0 (avg)

averaging across `lexical_class`

- Predicted % children who know a function word with <u>average frequency</u> (across dataset), noun with <u>average frequency</u>, etc.

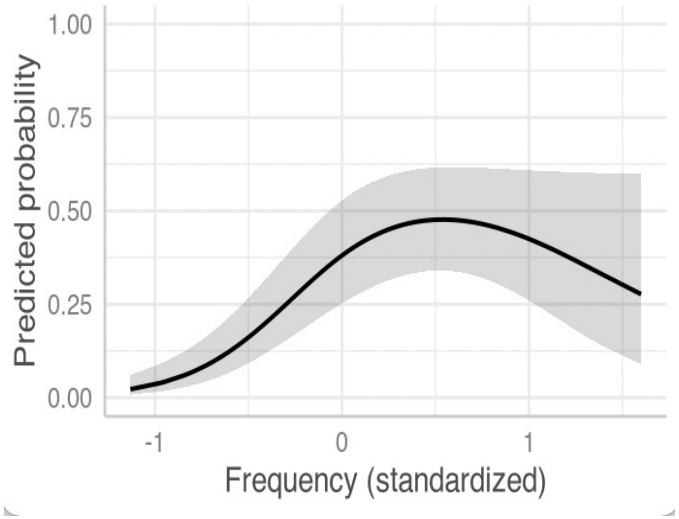
- This may not make sense given frequency ~ lexical class correlation
  - Another option: set frequency to its average for each lexical class.
- (slightly different)



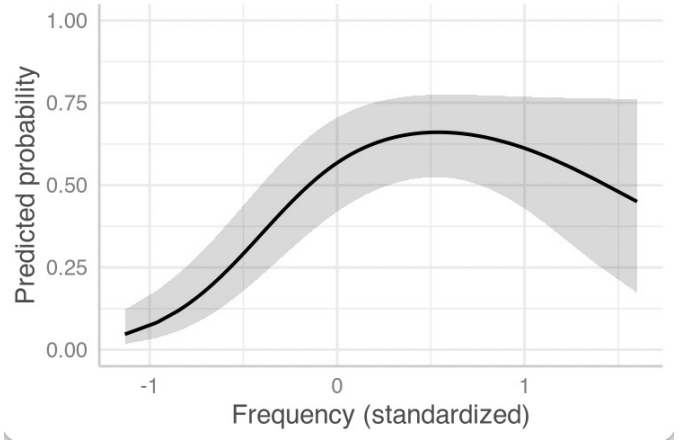
RMLD 9.7

Model predictions: examples

neither of these is “right” - depends on the context



`ggpredict()` from ggeffects



`ggemmeans()` from ggeffects

Note the difference:

```
Adjusted for:
*   lexical_class = function_words
*             sex =          Female
* log_nphones.std =           -0.03
*         MLU.std =           -0.00
*           child = 0 (population-level)
*      definition = 0 (population-level)
```

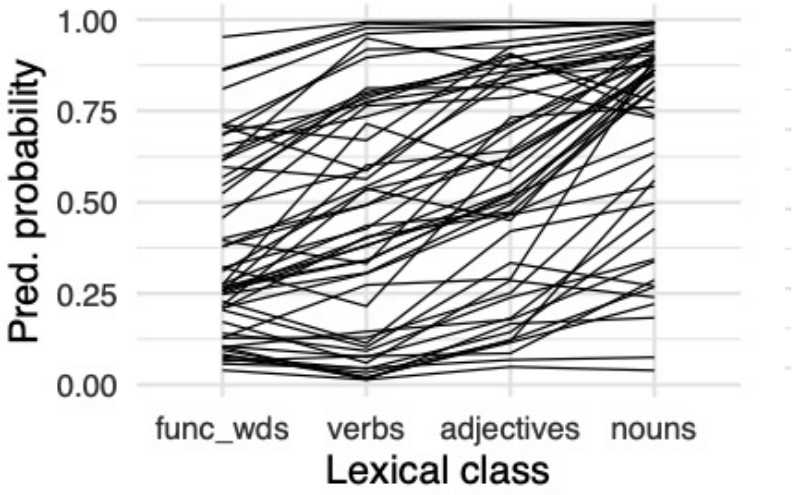
```
Adjusted for:
* log_nphones.std = -0.03
*         MLU.std = -0.00
```

Doesn’t tell you: uses emmeans, which averages over sex, lexical_class, child, definition

- Moral: automatic predictions are great, but be cautious

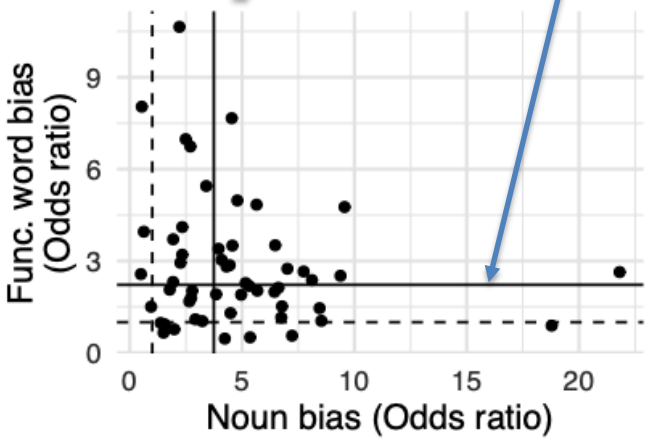
Model predictions: examples

- By-child predicted effects:



… of factor of interest

population-level effects



… of contrasts of interest (for RQs)

- Done "by hand", taking into account each child's gender

RMLD 9.7.5

Model predictions: examples

- Demo: other effects for this model
  - Time permitting

Unpacking results: last note

- Implementation of emmeans/ggeffects or similar packages for model predictions and EMMs
  - Effects, ggeffects, modelbased, marginaleffects, ….
  - Tricky at first, but excellent investment of time to learn
  - The underlying ideas are more important than the particular package used (which will change)

also: gratia, itsadug, tidygam

- All work for more complex models
  - GLMMs, GAMMs, Bayesian models…

also: tidybayes, bayesplot

Questions

Part 4: Mixed-effects models

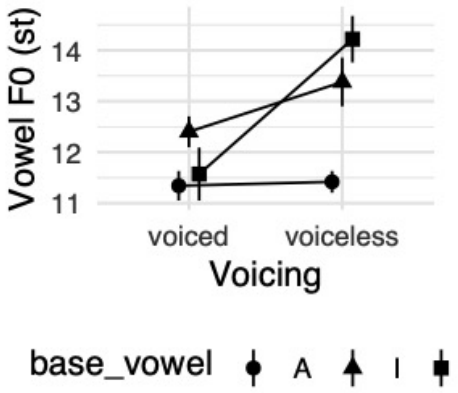
- RMLD Ch. 8-10
- Today: some aspects particularly relevant for corpus data
- Random effects: lesser-known uses, selecting RE structure
- Model-fitting issues: convergence, singularity
- Model selection: high-level
- Code: not in today's file.
  - In RMLD and associated R/Rmd files on the book's OSF website
  - `008-linear-mixed-models.R, 009-mixed-models-2.R, 010-mixed-models-3.R, ch10_appendix.R`

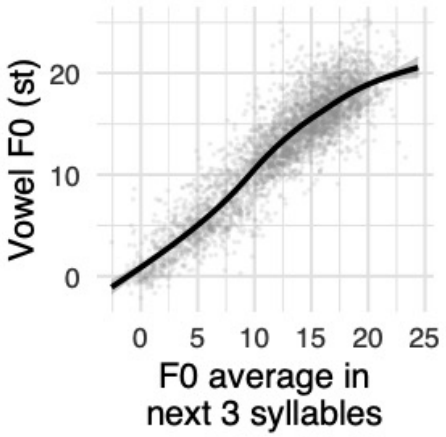
Dataset: `turkish_if0`

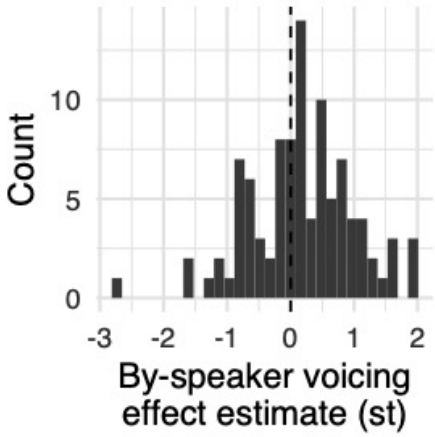
- Turkish read sentences (GlobalPhone corpus)
  - Intrinsic F0 effects
- F0 of vowels in utterance-initial CV
- Predictors:
  - Word-level: `base_vowel`, C `Voicing`, …

n = 4.3k

98 speakers, 1.3k words







primary interest, e.g "pa" - "ba" f0

Updated analysis, across 16 languages: Ting et al. (2022)

Random effects: lesser-known uses

- In lang. sciences, random effects commonly used to account for non-independence of data from units drawn from a "population"
  - By-speaker/word/item random intercepts, slopes
  - Typically "crossed": (1+…|speaker) + (1 + … | word)

- Can also use to account for other kinds of non-independence

Controlling for a nuisance factor

- A first model for turkish_if0 :

```
if0_m00 <- lmer(f0 ~ Voicing.vl * base_vowel + gender.male +
  local_f0 + (1 | word) + (1 | speaker), data = turkish_if0)
```

- word-level predictor of interest: model asks, "does Voicing affect f0, after controlling for by-word variation?"

- This may not be enough!
  - Voicing is a property of `consonant`, property of `word`

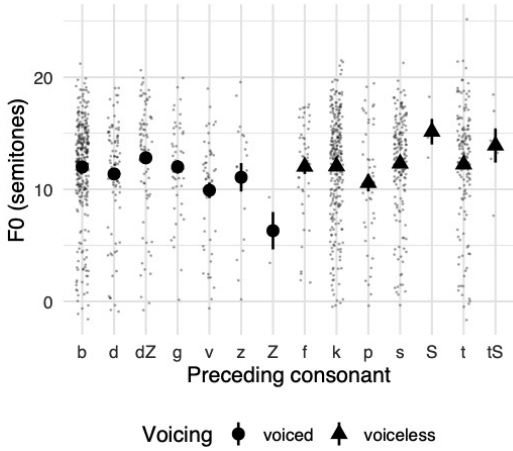


Figure 10.2
Empirical effect of preceding `consonant` on `f0` for the `turkish_if0` data.

Controlling for a nuisance factor

- `consonant` is a nuisance factor
  - many levels, we don't care about it, but non-independence needs to be accounted for

- Solution: code as a random intercept
  - OK as long as its levels are ~normally distributed

```
if0_m0 <- update(if0_m00, . ~ . + (1 | consonant))
```

- `Voicing` effect now asks: "what is the effect of this `consonant`-level predictor, controlling for `consonant` differences?

RMLD 10.2

Controlling for a nuisance factor

without (1|consonant)

```
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)           12.5031     0.1692   73.91
## Voicing.vl             0.7620     0.0931    8.18
## base_vowelAvIU         0.7645     0.0842    9.08
## base_vowelIvU         -0.0308     0.1270   -0.24
```

with (1|consonant)

```
## Fixed effects:
##                      Estimate Std. Error t value
## (Intercept)           12.4916     0.1889   66.14
## Voicing.vl             0.7583     0.1863    4.07
## base_vowelAvIU         0.7493     0.0853    8.79
## base_vowelIvU         -0.0237     0.1265   -0.19
```

more conservative ✅

- Other examples:
  - controlling for preceding and following consonant effects on vowel formants (Sonderegger et al. 2017)
  - …

RMLD 10.2

Nested random effects

- Also shows an example of hierarchical grouping structure in the data
  - `word` is nested within `consonant`

1. `(1|x) + (1|y)` (as in `if0_m0`)
2. `(1|x:y) + (1|y)`
3. `(1|y/x)`

Equivalent ways of writing grouping structure in lme4

- Examples:
  - Corpus linguistics (`text` within `register`)
  - Dialectology (`speaker` within `dialect`)
  - … still uncommon

Gries (2015), Tanner et al (2020), Sonderegger et al. (2023)

Random-effect structure: refresher

- Using a new example:
- `vot_core` : voiced stops, subset of speakers

```
core_speakers <- c("dale", "darnell", "lisa", "luke", "michael",
                   "mohamed", "rachel", "rebecca", "rex", "sara", "stuart")

vot_voiced_core <- vot %>%
  filter(syll_length==1 & speaker %in% core_speakers &
           voicing=='voiced') %>% droplevels()
```

- *y* : `vot`
- Grouping factors: `word`, `speaker`
- RQ: effect of `speaking_rate_dev`
  - observation-level

- A simple `vot_voiced_core` model:

```
vot_mod1 <- lmer(log_vot ~ speaking_rate_dev + foll_high_vowel +
                       cons_cluster + log_corpus_freq + place  +
                       gender + (1|word) + (1|speaker),
                     data = vot_voiced_core)

    ## add an uncorrelated by-speaker random slope
    vot_mod2 <- update(vot_mod1, . ~ . + (0+speaking_rate_dev|speaker))
```

- By-speaker, item random intercept
- By-speaker `speaking_rate` slope

accounts for basic grouping structure and the predictor of interest

- Random slopes are important
- This is not a "good" model -- only one random slope included for pedagogical simplicity
  - What about others?

RMLD 8.4.3.1

Possible random slopes

- By-z random slopes for predictor *x* only possible when *x* is not z-level
  - ✅ by-speaker `speaking_rate_dev`, by-word `gender` slopes
  - ❌ by-speaker `gender`, by-word `place` slopes

- For `vot_core` example, the maximal model would be:

```
(1 + speaking_rate_dev + foll_high_vowel + cons_cluster +
    log_corpus_freq + place || speaker) +
  (1 + speaking_rate_dev + gender || word)
```

RMLD 8.4.5

Random-effects model selection

- What adding a random slope does to fixed effect:
  - lower Type I error, higher *p*-value : this is good!
- But also: higher Type II error, lower power

- Of all possible random slopes, which ones to add?
  - This is the biggest practical issue in building mixed-effects models
  - … especially for corpus data

RMLD 8.4.4

Random-effects model selection

- Strategies:
  - Maximal: add as many random slopes as possible
    (Barr et al., 2013)
  - Data-driven: add slopes which significantly improve the model
    (Bates et al. ,2015; Matuschek et al. 2017)
  - 'Uncorrelated first'
    (Sonderegger 2023; Seedorf et al., 2019)
- Considered in more detail in RMLD Ch. 10

RMLD 8.4.4

Random-effects model selection

- Regardless of strategy followed: basic guidelines to fit mixed-effects models that make sense

1. Include all possible random intercepts

2. Consider all possible random slope terms for effects of theoretical interest

3. Random slope for an interaction ⇒ include random slopes for all subsets

These guidelines are not lmer-specific: hold for GAMMs, Bayesian MEMs, etc.

Correlated random effects

- Technical but important issue when fitting (g)lmer models to corpus data
  - Or any setting with many predictors

RMLD 8.7, 10.4

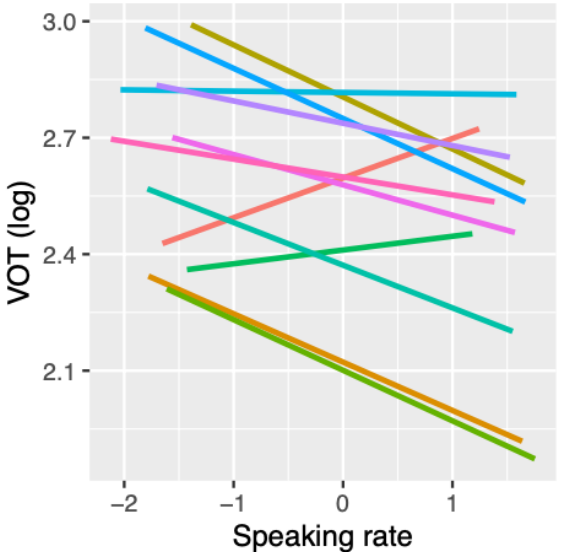
Correlated random effects

- Our model above assumed <u>uncorrelated</u> random effects `vot_mod2` :

  - `(1+speaking_rate_dev||speaker)`
  - `(1|speaker) + (0+speaking_rate_dev|speaker)`
  - `(1|speaker) + (-1+speaking_rate_dev|speaker)`

`lme4` notation: equivalent

- Maybe not realistic:
  - Average VOT unrelated to `speaking_rate` slope?



Speaker empirical effects

RMLD 8.7, 10.4

```
> vot_mod3 <- lmer(log_vot ~ speaking_rate_dev + foll_high_vowel + con
+     log_corpus_freq + place + gender + (1 | word) +
+       (1 + speaking_rate_dev | speaker), data=vot_voiced_core)
```

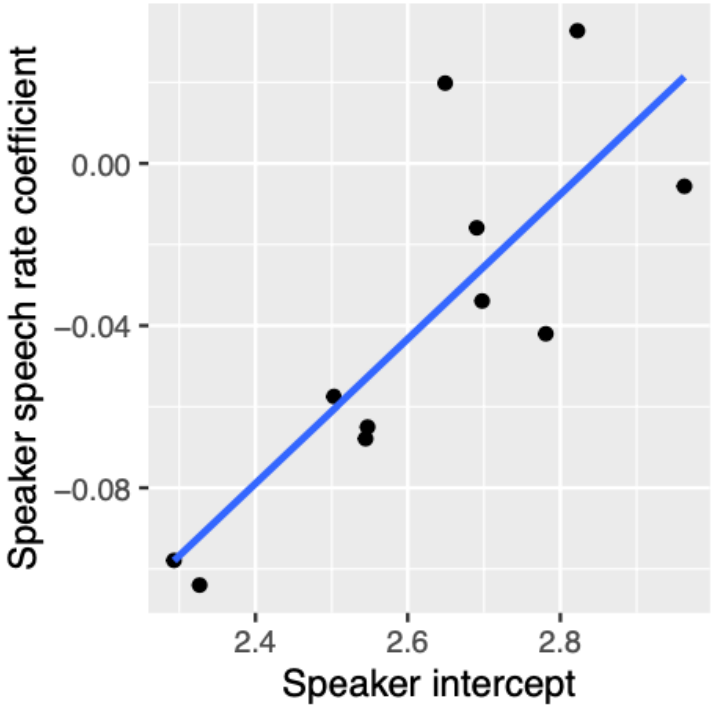
Correlated random intercept and slope: default in lme4

| = correlated, || = uncorrelated

- Random effects:

```
Random effects:
 Groups   Name                Std.Dev. Corr
 word     (Intercept)         0.13293
 speaker  (Intercept)         0.21253
          speaking_rate_dev   0.05367  0.68
 Residual                     0.56648
Number of obs: 7175, groups:  word, 272; speaker, 11
```

- Speakers with higher intercepts have higher `speaking_rate` slopes:



- Adding a random-effect correlation (`vot_mod2` vs. `vot_mod3`):
  - Random effects change
  - fixed effects very similar
  - `anova(vot_mod2, vot_mod3)`: $p<0.05$

Singular model

example using neutralization dataset, not considered today

- However, in practice models complex ranef structure often singular or doesn't coverge for linguistic data
  - Perfect correlations or zero random slopes

```
## Random effects:
##  Groups    Name        Variance Std.Dev. Corr
##  item_pair (Intercept)  76.62    8.75
##            voicing       8.34    2.89    -1.00
##  subject   (Intercept) 674.21   25.97
##            voicing      19.84    4.45     0.97
##  Residual              395.10   19.88
```

(1+voicing|item_pair) +
(1+voicing|subject)

- Typically because model is "too complex" to estimate from data
  - Much bigger problem as # random slopes increases, as in much corpus data

RMLD 8.7.2

Correlated or uncorrelated?

- Adding random slope terms: crucial
- Including RE correlation terms
  - Usually slightly improves model, without changing qualitative conclusions (fixed effects)
  - Can lead to practical issues (singular/non-convergent model)

- useful to default to models with uncorrelated random effects when choosing RE structure
  - add in correlations heuristically
- uncorrelated-first strategy
  - Important practical details: RMLD 8.7.3-4, 10.5

Correlated or uncorrelated?

- Caveats:
  - Doesn't apply if RQs involve random effects
  - Predictors must be "centered" for uncorrelated ranefs to make sense

- Singularity/convergence involving complex random effect structure is not an issue if you transition to Bayesian MEMs
  - this is an important motivation for doing so (e.g. Vasishth et al., 2018; Nicenboim et al, 2022)

RMLD 8.7.3-4, 10.5

Random slopes for factors

- To include uncorrelated random slopes for factors, need a hack:
  - Extract contrasts as numeric predictors (e.g. `place1`, `place2`)
  - Include these in random-effect structure

`(1+place|speaker)` ❌

`(1+place1+place2|speaker)` ✅

RMLD 8.7.4

Questions

Model-fitting issues

- Very common when fitting (any) complex statistical model:
  - General issues (optimization)
  - lme4-specific issues
- Basic troubleshooting knowledge is crucial
  - Especially for model selection

1. Model convergence

Whether a model converges is not a good measure of model quality.

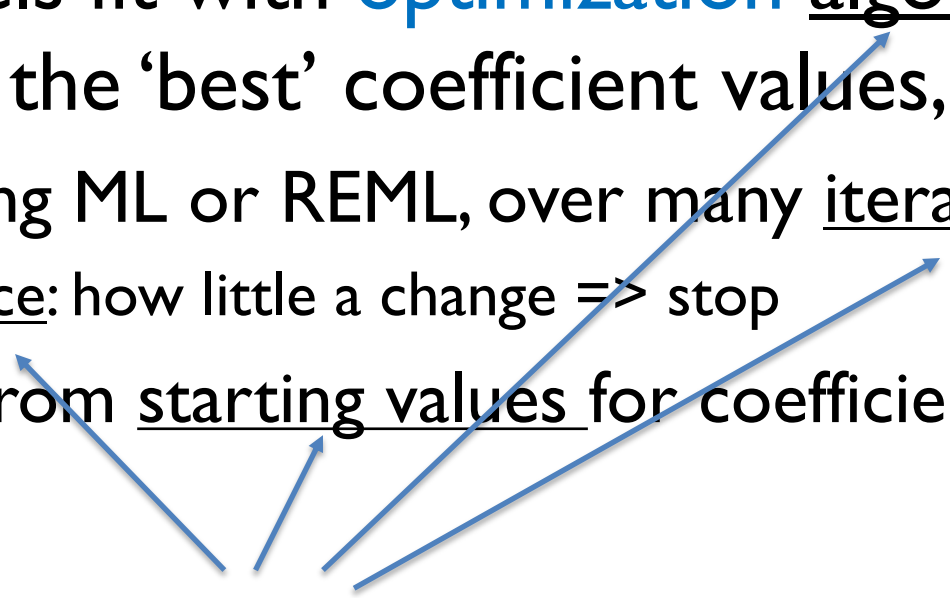
2. Singular models

RMLD 10.3-10.4

Background

- Critical predictors: directly related to RQs
- vs. control predictors

- lmer models fit with optimization algorithms, which find the 'best' coefficient values, by:
  - Maximizing ML or REML, over many iterations
    - tolerance: how little a change => stop
  - Starting from starting values for coefficients



lmer/glmer use default values: not sacred, can be changed to aid fitting

Non-convergence

- Very common!
- Opaque error messages:

```
## > vot_mod3_full <- update(vot_mod3, . ~ . + stress, data=vot_voiced)

## Warning message:
## Model failed to converge with max|grad| = 0.00359033 (tol = 0.002, component
1)
```

```
_coda +
```

```
## Warning message:
## unable to evaluate scaled gradient
## Warning message:
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

```
## Warning message:
## convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations
exceeded
## boundary (singular) fit: see ?isSingular
```

RMLD 10.3

Model convergence

- Upshot:
  - **Whether a model converges is not a good measure of model quality.**
  - Simple fixes often possible
  - Do not use convergence (alone) to choose between models

Model convergence: fixes

- Try in order:

1. Non-intrusive:
   i. Check your data and model
   ii. Standardize predictors (center, possibly scale)
   iii. Increase number of iterations
   iv. Change the optimizer
   v. Give the optimizer better start values
2. Intrusive:
   i. Remove random effects involving control predictors (must not be in interactions with critical predictors)
   ii. Selectively remove random-effect correlations: for control predictors, then correlations that are probably close to 0
   iii. Remove random intercept (leaving slope terms in)
   iv. Remove random slopes for critical predictors

most important

see `?convergence`

Adapted from Brauer & Curtin (2018)

Example: checking the model

- Spot the issue in this model of the `vot` data:

```
## > vot_mod3_bad <- lmer(log_vot ~ speaking_rate_dev +
## + foll_high_vowel + cons_cluster + log_corpus_freq + place + gender +
## + (1+speaking_rate_dev+place|speaker) + (1+place|word),
## +   data=vot_voiced_core)

## Warning message:
## unable to evaluate scaled gradient
## Warning message:
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

invalid by-word random slope

RMLD 10.3

Example: standardizing predictors

- RMLD 10.3.2.4 shows a model of `vot` data including these "maximal" random effects:

```
(1 + speaking_rate_dev | speaker) +
(1 + gender + speaking_rate_dev | word) +
(1 + speaking_rate_dev + foll_high_vowel + cons_cluster +
  log_corpus_freq + place + stress | speaker))
```

predictors not standardized

```
## Model failed to converge with max|grad| = 0.0424372 (tol = 0.002
```

- with standardized predictors: does converge

RMLD 10.3.2.4

Changing optimizer parameters

- When nothing is wrong with data/model/predictors, non-convergence often solved by:
  - Tweaking some model fitting parameters
  - Running the model longer

Summary: non-intrusive methods

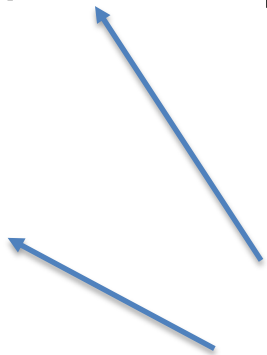
- Fixes are easy and/or non-technical
  - .. and boring
- Often work

- Otherwise, go to 'intrusive methods'
  - Pruning random-effect structure

Singular models

- A 'dimension' of the random effects has been estimated at zero
- Usually can see why: random-effect
  - variances near 0
  - correlations near -1/1
- Also possible:
  - No obvious reason

not necessarily a problem



problem / should not be in final model

What about Bayesian models?

- To the extent your problems with model fitting are just related to random-effect structure, a worthwhile option is Bayesian MEMs
  - These will fit "any" ranef structure without these issues

(e.g. Vasishth et al., 2018; McElreath, 2020; Nicenboim et al, 2022)

- Caveats:
  - Non-Bayesian MEMs are easier to work with in practice
  - Bayesian models are harder to understand, and you shouldn't use methods you don't understand.
  - Bayesian models are not a panacea! The underlying issues with data/modeling don't go away by switching to Bayesian models (RMLD Box 10.2)

What about Bayesian models?

- Nonetheless….this is what I actually do in practice for most corpus data in 2023: Bayesian MEMs, in brms or rstanarm.

- Bayesian modeling is the natural next thing to learn once you're comfortable with (G)LMMs – much more powerful and flexible.

Model selection for MEMs

- Actually choosing a model is hard – the hardest part of mixed-effects modeling

  Meteyard & Davies (2020)

  - Especially for corpus data!

- Two steering wheels:

  Snijders & Bosker (2011: 6.2)

  - Substantive (subject-matter related)
  - Statistical considerations
  - ... Both fixed and random effects (four wheels?)
- ⇒ there are no fixed rules to follow
  - … but there are principles, and heuristics

- In-depth treatment, with both lab and corpus data in mind: RMLD 10.5

Bad models

- Simple recipes can easily lead you to bad models
  - These recipes for random-effect selection follow "expert advice", without their nuance

1. Random intercepts only

Baayen, Davidson, Bates (2008)

2. Maximal random effects

Barr et al. (2013)

3. Data-driven random effects

Bates et al. (2015), Matuschek et al. (2017)

4. Uncorrelated random effects

Sonderegger et al. (2018), Sonderegger (2023)

Bad models

- Also applies to fixed-effect structure, by same logic as in Variable Selection (Part 2)

5. Throw in all possible predictors

6. Use a fully-automatic method to select predictors, not distinguishing between critical and control terms

Principles for model selection

- In order:

1. Prioritize subject matter considerations

RQs, study design, theoretical considerations, previous work, common sense

2. Distinguish between critical and control terms

Design model to estimate critical terms as accurately as possible

3. Fixed effects need appropriate error terms

usually means: random slopes

adapted from Snijders & Bosker (2011)

Principles for model selection

4. Larger models have lower power to detect individual effects

   beware “too many” random effects or control predictors

5. The model’s “hierarchical” structure should make sense

6. Be reluctant to overfit / include non-significant effects

7. Be reluctant to underfit

Model selection: practical procedure

- High level:

  - Step 1. Perform exploratory analyses, visualization, selecting possible predictors, etc.
  - Step 2. Build maximal *fixed-effects* structure.
  - Step 3. Select *random-effects* structure, using those fixed effects.
  - Step 4. Do any pruning of fixed effects.

- Concretely: many possible procedures
  - This is OK
- Most important: say what you did and why
  - Currently uncommon in published work

RMLD 10.5.3-10.5.4

Model selection: case studies

- If we have time:
  - RMLD 10.5.6
    - "Data-driven" approach: `turkish_if0` data
  - RMLD 10.5.7
    - "Uncorrelated first" approach: `vot` data

Questions

References

- Baayen, R. Harald. 2008. *Analyzing linguistic data.* Cambridge University Press.
- Baayen, R. Harald, D. J. Davidson, and D. M. Bates. 2008. "Mixed-effects modeling with crossed random effects for subjects and items." *Journal of Memory and Language* 59 (4): 390–412
- Baguley, Thomas. 2012. *Serious stats: A guide to advanced statistics for the behavioral sciences.* Macmillan.
- Hye-Young Bang, Morgan Sonderegger, Yoonjung Kang, Meghan Clayards, Tae-Jin Yoon, The emergence, progress, and impact of sound change in progress in Seoul Korean: Implications for mechanisms of tonogenesis, *Journal of Phonetics*, Volume 66, 2018, Pages 120-144,
- Brauer, Markus, and John J. Curtin. 2018. "Linear mixed-effects models and the analysis of nonindependent data: A unified framework to analyze categorical and continuous independent variables that vary within-subjects and/or within-items." *Psychological Methods* 23 (3): 389–411.
- Gelman, Andrew, and Jennifer Hill. 2007. *Data analysis using regression and multilevel/hierarchical models.* Cambridge University Press.
- Gries, Stefan Th. 2015. "The most under-used statistical method in corpus linguistics: Multi-level (and mixed-effects) models." *Corpora* 10 (1): 95–125.
- Harrell, F. E. 2015. *Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis.* 2nd ed. Springer Verlag.
- Kang, Yoonjung. Voice onset time merger and development of tonal contrast in Seoul Korean stops: A corpus study *Journal of Phonetics*, 45 (2014), pp. 76-90

References

- McElreath, Richard. 2020. *Statistical rethinking: A Bayesian course with examples in R and Stan*. 2nd ed. Chapman / Hall/CRC.
- Nicenboim, Bruno, D. J. Schad, and Shravan Vasishth. 2022. *An introduction to Bayesian data analysis for cognitive science*. https://vasishth.github.io/bayescogsci/book/.
- Roettger, Timo B. 2019. "Researcher degrees of freedom in phonetic research." *Laboratory Phonology* 10 (1): 1–27.
- Schad, Daniel J., Shravan Vasishth, Sven Hohenstein, and Reinhold Kliegl. 2020. "How to capitalize on a priori contrasts in linear (mixed) models: A tutorial." *Journal of Memory and Language* 110:104038.
- Seedorff, Michael, Jacob Oleson, and Bob McMurray. 2019. "Maybe maximal: Good enough mixed models optimize power while controlling type I error." PsyArXiv preprint: https://psyarxiv.com/xmhfr/.
- Snijders, T. A. B., and R. J. Bosker. 2011. *Multilevel analysis*. 2nd ed. Sage Publications.
- Sonderegger, Morgan, Max Bane, and Peter Graff. 2017. "The medium-term dynamics of accents on reality television." *Language* 93 (3): 598–640.
- Morgan Sonderegger, Jane Stuart-Smith, Jeff Mielke, and The SPADE Consortium. (2023) How variable are English sibilants? *Proceedings of the 20th International Congress of Phonetic Sciences*.
- Tanner, James, Morgan Sonderegger, Jane Stuart-Smith, and Josef Fruehwald. 2020. "Toward "English" phonetics: Variability in the pre-consonantal voicing effect across English dialects and speakers." *Frontiers in Artificial Intelligence* 3:38.
- Vasishth, Shravan, Bruno Nicenboim, Mary E. Beckman, Fangfang Li, and Eun Jong Kong. 2018. "Bayesian data analysis in the phonetic sciences: A tutorial introduction." *Journal of Phonetics* 71:147–161.