A tutorial of Bayesian beta regressions with brms in R

Stefano Coretta[a,1,*], Paul Bürkner

[a] *University of Edinburgh, Linguistics and English Language, 3 George Sq, Edinburgh, EH8 9AD*

Abstract

This is the abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum augue turpis, dictum non malesuada a, volutpat eget velit. Nam placerat turpis purus, eu tristique ex tincidunt et. Mauris sed augue eget turpis ultrices tincidunt. Sed et mi in leo porta egestas. Aliquam non laoreet velit. Nunc quis ex vitae eros aliquet auctor nec ac libero. Duis laoreet sapien eu mi luctus, in bibendum leo molestie. Sed hendrerit diam diam, ac dapibus nisl volutpat vitae. Aliquam bibendum varius libero, eu efficitur justo rutrum at. Sed at tempus elit.

Keywords: keyword1, keyword2

1. Introduction

Phonetic research often involves numeric continuous outcome variables, like durations, frequencies, loudness and ratios. Another commonly employed type of outcome variable are proportions: for example, proportion of voicing during closure, vocal folds contact quotient, gesture amplitude, nasalance. Moreover, virtually any measure can be MIN-MAX normalised, a procedure which transforms values so that they are in the range 0–1.

Regression models are very common, but there is a tendency of using Gaussian distribution families (i.e. probability distributions for the outcome variable) for anything that is numeric. A possible reason is that the base R function for fitting regression models, `lm()`, and the lme4 function used to fit regression models with varying terms, `lmer()`, both fit Gaussian regressions by default and the user does not have to specify the distribution family. This tacit defaulting to Gaussian models is also reflected in teaching practices, where test and models using the Gaussian distribution are the first to be taught, due to their relative simplicity and the fact that other models are generalisations of Gaussian models.

However, proportion are naturally not Gaussian, since they are limited between 0 and 1. The theoretical distribution that generates values with this characteristics is the beta distribution. Thus, regression models with proportions as the outcome variable should be fitted using a beta distribution as the distribution family. This tutorial introduces researchers to beta regression models in R using the package brms. Familiarity with regression modelling in R with a package like lme4 is assumed, but no prior knowledge of Bayesian statistics is necessary.

2. The beta distribution

…

*Corresponding author

Email addresses: `s.coretta@ed.ac.uk` (Stefano Coretta), `paul.buerkner@gmail.com` (Paul Bürkner)

[1] This is the first author footnote.

3. Case study 1: voicing within consonant closure

For the first case study, we will model the proportion of voicing within consonant closure. The measurements come from a data set of audio and electroglottographic (EGG) recordings of 19 speakers of Northwestern Italian. The participants read frame sentences which included target words of the form /CVCo/, where /C/ was either /k, t, p/ in all permutations and /V/ was either /i, e, a, , u/ (two resulting words, /peto/ and /kako/ were excluded because they are profanities), for a total of 43 target words. There were 4 different frame sentence, with a total of 172 trials per participant (3,268 grand total). The actual observation count is 2,419, after removing speech errors, EGG measurement errors, and cases in which voicing ceased before the closure onset/after the closure offset of the second /C/.

The proportion of voicing during the closure of the second /C/ was calculated as the proportion of contiguous voicing duration after closure onset to total duration of closure. The following code chunk attaches the tidyverse packages (for reading and wrangling data) and loads the `ita_egg` tibble (data frame). The tibble is filtered so as to remove voicing proportions (`voi_clo_prop`) that are smaller than 0 and greater than 1. The variables `vowel` and `c2` are converted to factors to specify the order of the levels.

```
# attach tidyverse and set light theme for plots
library(tidyverse)
theme_set(theme_light())

# load tibble
load("data/coretta2018/ita_egg.rda")

# filter and mutate data
ita_egg <- ita_egg |>
  filter(voi_clo_prop > 0, voi_clo_prop < 1) |>
  mutate(
    vowel = factor(vowel, levels = c("i", "e", "a", "o", "u")),
    c2 = factor(c2, levels = c("k", "t", "p"))
  )
```

Here is what the tibble looks like.

```
ita_egg
```

```
# A tibble: 2,419 x 53
   speaker ipu    stimulus    sentence_ons sentence_off word_ons word_off v1_ons
   <chr>   <chr>  <chr>              <dbl>        <dbl>    <dbl>    <dbl>  <dbl>
 1 it01    ipu_1  Ripete 'po~         13.2         14.9     13.7     14.1   13.9
 2 it01    ipu_2  Ripete 'to~         16.9         18.6     17.4     17.9   17.5
 3 it01    ipu_3  Ripete 'pa~         20.2         21.9     20.7     21.1   20.8
 4 it01    ipu_4  Sentivo 't~         23.5         25.1     24.0     24.4   24.1
 5 it01    ipu_5  Sentivo 't~         26.3         27.8     26.8     27.2   26.9
 6 it01    ipu_6  Scrivete '~         29.2         30.9     29.7     30.1   29.8
 7 it01    ipu_7  Sentivo 'c~         32.1         33.6     32.6     33.1   32.8
 8 it01    ipu_8  Scrivete '~         35.0         36.6     35.5     35.9   35.6
 9 it01    ipu_9  Ha detto '~         41.9         43.5     42.3     42.7   42.4
10 it01    ipu_10 Sentivo 'p~         47.4         48.9     47.9     48.4   48.1
# i 2,409 more rows
# i 45 more variables: c2_ons <dbl>, v2_ons <dbl>, voice_ons <dbl>,
```

```
#   voice_off <dbl>, c1_rel <dbl>, c2_rel <dbl>, stimulus_id <dbl>,
#   sentence <chr>, word <chr>, c1 <chr>, vowel <fct>, c2 <fct>,
#   backness <chr>, height <fct>, c1_place <fct>, c2_place <fct>,
#   v1_duration <dbl>, c2_clos_duration <dbl>, rel_voff <dbl>,
#   sent_duration <dbl>, speech_rate <dbl>, speech_rate_c <dbl>, ...
```

Figure 1 shows the raw voicing duration proportion values split by vowel /i, e, a, , u/ and second consonant /k, t, p/ in the /CVCo/ target words. The plot suggests that, on average, the voicing proportion is slightly lower with /k/ than with /t, p/. Moreover, there is greater variability between vowels in /t, p/ than in /k/. We will use a beta regression model to assess these patterns. ["expectations" XXX]

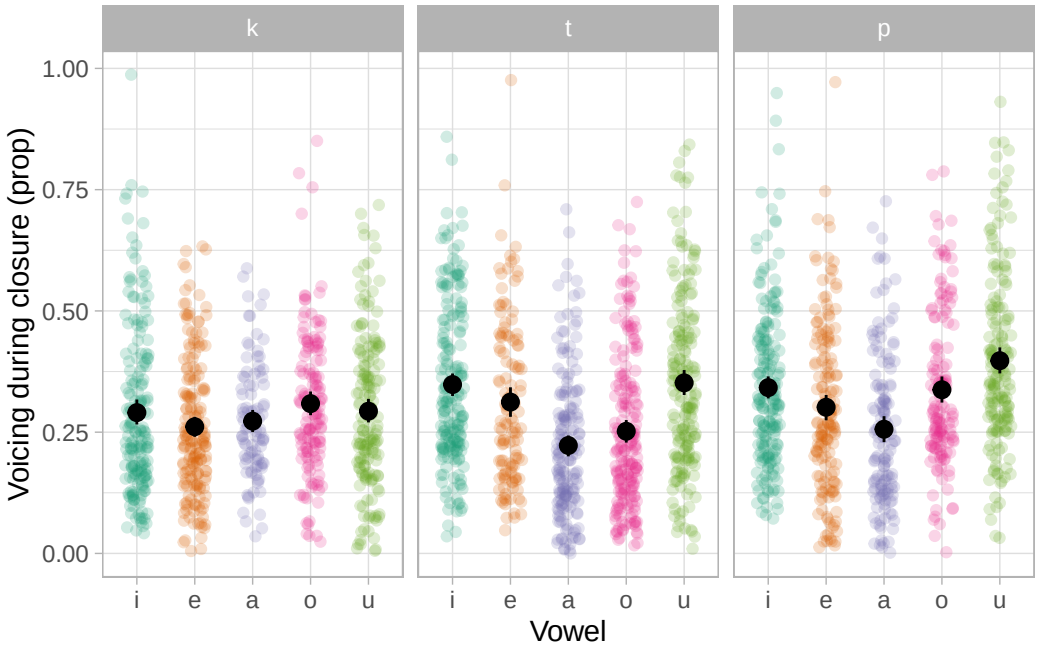


Figure 1: Proportion of voicing during the closure of the second consonant in /CVCo/ words by vowel and the second consonant.

We will use brms to fit Bayesian beta regressions. [XXX why Bayesian]. The model has voicing proportion as the outcome variable and the following terms: an interaction between vowel (/i, e, a, , u/) and second consonant C2 (/k, t, p/), centred speech rate (number of syllables per second); as varying (aka random) terms, by-speaker varying coefficients for the vowel/consonant interaction and for centred speech rate.[2] The categorical predictors vowel and C2 are coded using indexing rather than traditional R contrasts: in R, this corresponds to suppressing the model's intercept with the `0 +` syntax; using indexing instead of contrasts makes it easier to specify priors. For pedagogical simplicity, the model will use the default priors, but note that in real data analyses contexts, priors should be specified by the user. I refer the readers to XXX.

```
# attach brms
library(brms)

# fit the model
# Takes 3 minutes on MacBook Pro 2020, M1
```

[2]Footnote about Gelman's terminology for random effects.

```r
voi_prop_bm <- brm(
  # model formula
  voi_clo_prop ~
    # constant terms
    0 + vowel:c2 + speech_rate_c +
    # varying terms
    (0 + vowel:c2 + speech_rate_c | speaker),
  # uses the beta family for the outcome
  family = Beta,
  data = ita_egg,
  cores = 4,
  seed = 3749,
  file = "data/cache/voi_prop_bm"
)
```

The `summary()` function prints the full model summary. For conciseness, we will use the `fixef()` function which prints the regression coefficients. [EXPLAIN EXPECTED PREDICTIONS XXX]. The full summary with an explanation of each part can be found in XXX. For each coefficient in the model, `fixef()` prints the name of the coefficient, the mean estimate, the estimate error and the lower and upper limits of a Bayesian Credible interval (CrI). Here, we print an 80% CrI. There is nothing special about 95% CrI within Bayesian inference and instead experts recommend to check and report a variety of CrIs. Bayesian CrIs indicate that at a certain probability levels the "true" estimate lies within that interval: so, for example, a 90% CrI [A, B] indicates that there is a 90% probability that the "true" estimate is between A and B. Different probability levels correspond to different levels of confidence: the higher the probability the higher the confidence (always conditional on data and model). Obtaining CrIs at different probability levels allows researchers to make more fine-grained inferential statements than the frequentist significance dichotomy affords. For simplicity of exposition, we will use 80% CrIs in this case study but we strongly recommend researchers to always obtain CrIs at different levels of probability and base their inferences on all and not one in particular. To reiterate, in Bayesian inference, an 80% CrI indicates the range of values within which the true estimate falls at 80% probability or confidence. We round all values to the nearest 2 digits for clarity.

```r
round(
  fixef(voi_prop_bm, prob = c(0.1, 0.9)),
  digits = 2
)
```

```
             Estimate Est.Error   Q10   Q90
speech_rate_c     0.08      0.06  0.01  0.15
voweli:c2k       -0.91      0.14 -1.08 -0.74
vowele:c2k       -1.08      0.11 -1.22 -0.94
vowela:c2k       -0.99      0.12 -1.14 -0.84
vowelo:c2k       -0.79      0.14 -0.96 -0.62
vowelu:c2k       -1.00      0.16 -1.20 -0.80
voweli:c2t       -0.66      0.11 -0.79 -0.53
vowele:c2t       -0.84      0.14 -1.02 -0.66
vowela:c2t       -1.43      0.13 -1.60 -1.26
vowelo:c2t       -1.15      0.13 -1.31 -0.99
vowelu:c2t       -0.68      0.12 -0.83 -0.54
voweli:c2p       -0.68      0.11 -0.81 -0.54
vowele:c2p       -0.88      0.15 -1.07 -0.68
```

```
vowela:c2p       -1.14      0.13 -1.31 -0.98
vowelo:c2p       -0.66      0.11 -0.80 -0.53
vowelu:c2p       -0.44      0.12 -0.59 -0.28
```

The coefficients of a beta regression are estimated on the log-odds scale, as in Bernoulli/binomial (aka logistic) regressions. From the summary, we see that speech rate (number of syllables per second) has a positive effect on voicing proportion: the 80% CrI is between 0.01 and 0.15 log-odds [$\beta = 0.08$, SD $= 0.06$]. Log-odds can be converted to odd-ratios by exponentiating the value: 0.01-0.15 log odds correspond to an odd-ratio of 1.01 to 1.16, or as percentages, to an increase of voicing of 1 to 16% for every increase of one syllable per second. Since this is an 80% CrI, we can be 80% confident that the true effect of speech rate is between 1-16% increase of voicing proportion, conditional on the data and model. Note that transforming measures this way is appropriate *only* with quantile-based measures (like CrIs) but not with moments like the mean and standard deviation: to correctly get mean and SDs in the transformed scale, you must first extract the posterior draws (see below), convert them and then take moments such as mean and SD (for a more detailed explanation, see XXX). In the avoidance of doubt, we will always transform the drawn values first and then take summary measures.

Turning now to the coefficients for vowel and C2, given the indexing approach of coding these variables the model summary and the output of `fixef()` reports the *predictions* in log-odds for each combination of vowel and C2, rather than differences between levels. The CrIs of the vowel/C2 coefficients span all negative log-odds values: these correspond to proportions that are lower than 0.5 (which is 0 in log-odds). This matches the general trends in the raw data, which we plotted in Figure 1.

Next, we will plot the predicted proportions of each vowel/C2 combination at mean speech rate (i.e. centred speech rate = 0) and then calculate the average pair-wise difference in voicing proportion between /k, t, p/. Finally, we will assess whether there is greater between-vowel variability in /t, p/ relative to /k/.

Before being able to plot the predictions, it's important to get familiar with the so-called posterior draws. [Bayesian MCMC XXX]. Posterior draws can be conveniently obtained with the `as_draws_df()`. For the moment, we will extract only the draws of the constant regression coefficients (model variables starting with `b_`). To check which coefficients are available in a model, use `get_variables()` from the tidybayes package. `as_draws_df()` returns a tibble where each column contains the drawn values of a coefficient. The probability distribution of the drawn values of each coefficient is the posterior probability distribution of that coefficient. Note that, due to the indexing coding of vowel and C2, all coefficient except `b_speech_rate_c` are *predicted log-odds* for each vowel/C2 combination (the drawn values for `b_speech_rate_c` are drawn *differences* in log-odds for each unit increase of speech rate). The drawn values are in log-odds, but we can convert them to proportions with `plogis()` (we will do this when plotting below).

```
# extract only coefficient variables starting with "b_"
voi_prop_bm_draws <- as_draws_df(voi_prop_bm, variable = "^b_", regex = TRUE)
voi_prop_bm_draws
```

```
# A draws_df: 1000 iterations, 4 chains, and 16 variables
   b_speech_rate_c b_voweli:c2k b_vowele:c2k b_vowela:c2k b_vowelo:c2k
1            0.159        -0.72        -1.05        -1.12        -1.02
2            0.035        -0.75        -1.39        -1.06        -0.89
3            0.126        -0.85        -0.98        -1.10        -0.64
4            0.064        -0.70        -1.07        -1.05        -0.83
5            0.032        -0.77        -1.18        -0.94        -0.70
6            0.073        -0.85        -1.23        -1.06        -0.77
7            0.101        -0.95        -1.07        -1.14        -0.86
8            0.074        -1.02        -1.07        -1.17        -0.83
```

```
9            0.087        -1.04        -1.07        -1.26        -0.79
10           0.092        -1.20        -1.04        -0.92        -0.64
   b_vowelu:c2k b_voweli:c2t b_vowele:c2t
1         -1.05        -0.62        -0.80
2         -1.04        -0.67        -0.99
3         -0.91        -0.66        -0.80
4         -1.07        -0.53        -0.87
5         -1.06        -0.63        -0.87
6         -1.00        -0.69        -0.73
7         -1.22        -0.74        -1.08
8         -1.22        -0.73        -1.04
9         -1.14        -0.66        -0.91
10        -0.84        -0.63        -0.61
# ... with 3990 more draws, and 8 more variables
# ... hidden reserved variables {'.chain', '.iteration', '.draw'}
```

We can now wrangle this tibble and plot the posterior distributions for each vowel/C2 combination.

```
voi_prop_bm_draws_long <- voi_prop_bm_draws |>
  # drop b_speech_rate_c before pivoting
  select(-b_speech_rate_c) |>
  # pivot vowel:c2 columns
  pivot_longer(`b_voweli:c2k`:`b_vowelu:c2p`, names_to = "coeff") |>
  # separate "coeff" labels into type ("b"), vowel and c2
  separate(coeff, into = c("type", "vowel", "c2"))
```

```
Warning: Dropping 'draws_df' class as required metadata was removed.
```

```
voi_prop_bm_draws_long
```

```
# A tibble: 60,000 x 7
   .chain .iteration .draw type  vowel  c2     value
    <int>      <int> <int> <chr> <chr>  <chr>  <dbl>
 1      1          1     1 b     voweli c2k   -0.721
 2      1          1     1 b     vowele c2k   -1.05
 3      1          1     1 b     vowela c2k   -1.12
 4      1          1     1 b     vowelo c2k   -1.02
 5      1          1     1 b     vowelu c2k   -1.05
 6      1          1     1 b     voweli c2t   -0.625
 7      1          1     1 b     vowele c2t   -0.795
 8      1          1     1 b     vowela c2t   -1.40
 9      1          1     1 b     vowelo c2t   -1.10
10      1          1     1 b     vowelu c2t   -0.668
# i 59,990 more rows
```

For plotting, we can use ggplot2 statistics layers from the ggdist package. `stat_halfeye()` plots the density of the posterior probability (in grey), its median (point) and CrIs (lines). Let's use 60 and 80% CrIs and transform the log-odds values to proportions with `plogis()`.

```
# attach ggdist package
library(ggdist)
```

```
Attaching package: 'ggdist'

The following objects are masked from 'package:brms':

    dstudent_t, pstudent_t, qstudent_t, rstudent_t
```

```
voi_prop_bm_draws_long |>
  ggplot(aes(plogis(value), vowel)) +
  stat_halfeye(.width = c(0.6, 0.8)) +
  facet_grid(rows = vars(c2))
```

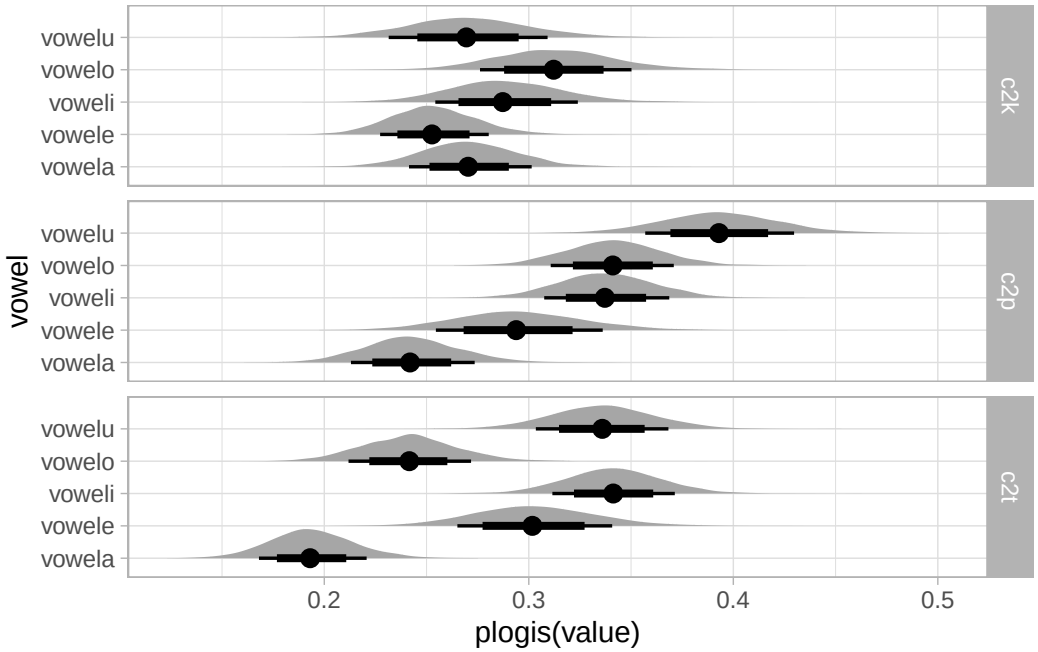


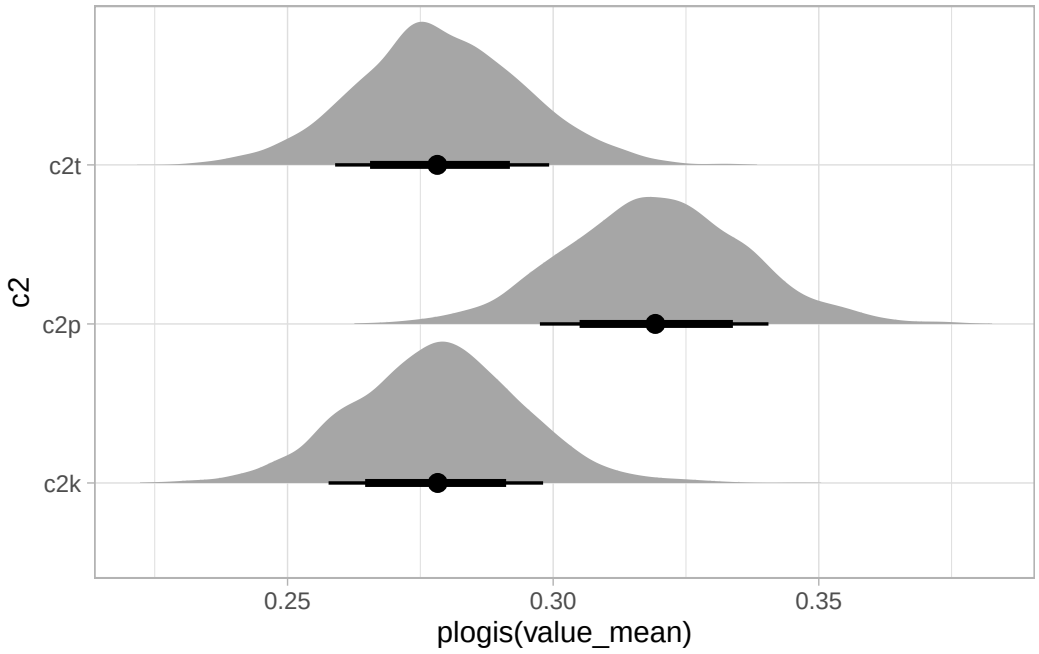
Figure 2

What if we want to plot the average predicted voicing proportion for the three consonants /k, t, p/? One approach is to take the mean across vowels within each consonant for each posterior draw, and the posterior distribution of the resulting list of values is the predicted posterior distribution of voicing proportion for each consonant, assuming an "average" vowel.

```
voi_prop_bm_draws_long_c2 <- voi_prop_bm_draws_long |>
  # grouping by .draw and c2 ensures that averaging applies only within draw and c2
  group_by(.draw, c2) |>
  # calculate the mean value within draw/c2
  summarise(
```

```
    value_mean = mean(value), .groups = "drop"
  )

voi_prop_bm_draws_long_c2 |>
  ggplot(aes(plogis(value_mean), c2)) +
  stat_halfeye(.width = c(0.6, 0.8))
```



Based on the posterior distributions of the mean voicing proportion by consonant, /p/ has a somewhat higher voicing proportion than /k/ and /t/. The real question is: how much higher? We can quantify this by taking the difference of the drawn values for /p/ and those for /t, k/. Since we want to compare /t, k/ to /p/, we should first average the draws of /t, k/ and then take the difference of the averaged draws and the draws of /p/.

```
voi_prop_bm_diff <- voi_prop_bm_draws_long_c2 |>
  # pivot data to create one column per consonant with the mean drawn values,
  # with one draw per raw
  pivot_wider(names_from = c2, values_from = value_mean) |>
  mutate(
    # calculate the mean of /k/ and /t/, for each draw
    c2tk = mean(c(c2k, c2t)),
    # calculate the difference of /p/ and /t, k/
    c2p_tk_diff = c2p - c2tk
  )

voi_prop_bm_diff
```

```
# A tibble: 4,000 x 6
   .draw    c2k    c2p    c2t   c2tk c2p_tk_diff
   <int>  <dbl>  <dbl>  <dbl>  <dbl>       <dbl>
```

```
 1     1 -0.992 -0.745 -0.917 -0.953      0.208
 2     2 -1.02  -0.790 -1.03  -0.953      0.163
 3     3 -0.895 -0.687 -0.890 -0.953      0.266
 4     4 -0.944 -0.814 -0.910 -0.953      0.140
 5     5 -0.928 -0.791 -0.920 -0.953      0.162
 6     6 -0.983 -0.728 -0.926 -0.953      0.225
 7     7 -1.05  -0.873 -1.08  -0.953      0.0802
 8     8 -1.06  -0.875 -1.07  -0.953      0.0782
 9     9 -1.06  -0.974 -1.05  -0.953     -0.0206
10    10 -0.929 -0.829 -0.953 -0.953      0.125
# i 3,990 more rows
```

```
voi_prop_bm_diff |>
  ggplot(aes(c2p_tk_diff)) +
  stat_halfeye(.width = c(0.6, 0.8, 0.9)) +
  geom_vline(xintercept = 0)
```

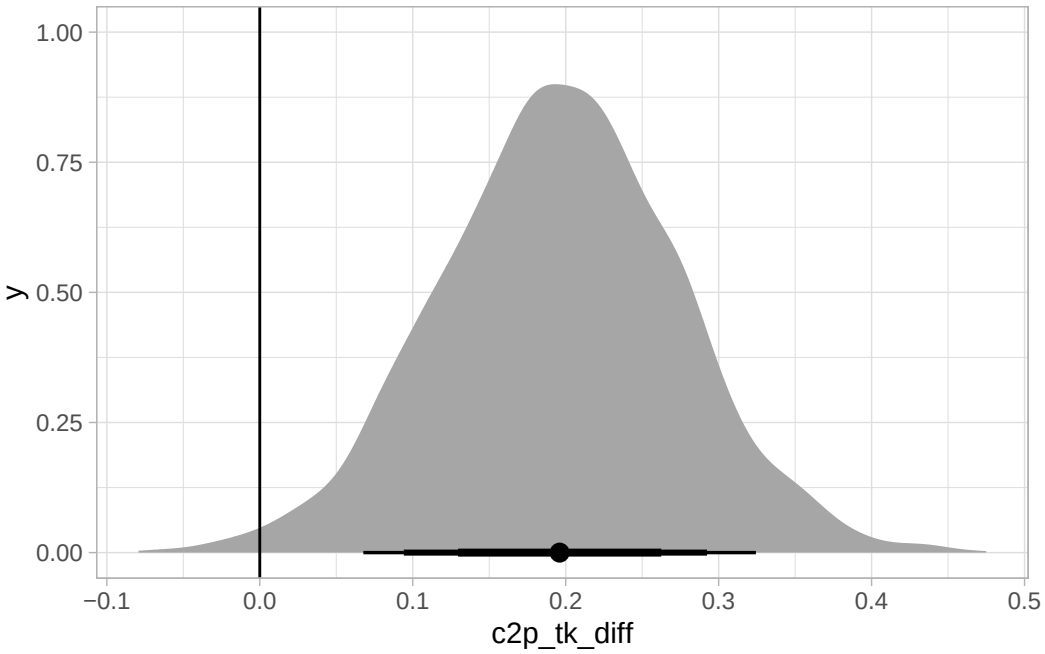


Figure 3

Once we have the posterior difference, we can obtain CrIs of the difference using `quantile2()` from the posterior package. Beware that the values of the difference are in log-odds! We can convert these into odd-ratios with `exp()`. odd-ratios indicate the ratio of the difference between A and B, so that 1 means no difference, values greater than 1 indicate an increase in A relative to B and values lower than 1 indicate a decrease in A relative to B. odd-ratios are useful when looking at differences that are in log-odds because while the relative magnitude of the difference in proportion between two groups is the same independent of the baseline proportion, the *absolute* magnitude of the difference depends on the baseline value. For example, an odd-ratio difference of 1.25 would correspond to a proportion increase of 13 percentage points if the baseline proportion is 0.62 but it would correspond to a proportion increase of 17 percentage points if the baseline proportion is 0.73. Of course, in real research contexts it is still usefull to think about absolute

magnitudes and their relevance from a conceptual and methodological perspective. In this tutorial we just focus on odd-ratios for simplicity.

```
library(posterior)
```

```
This is posterior version 1.6.0


Attaching package: 'posterior'

The following objects are masked from 'package:stats':

    mad, sd, var

The following objects are masked from 'package:base':

    %in%, match
```

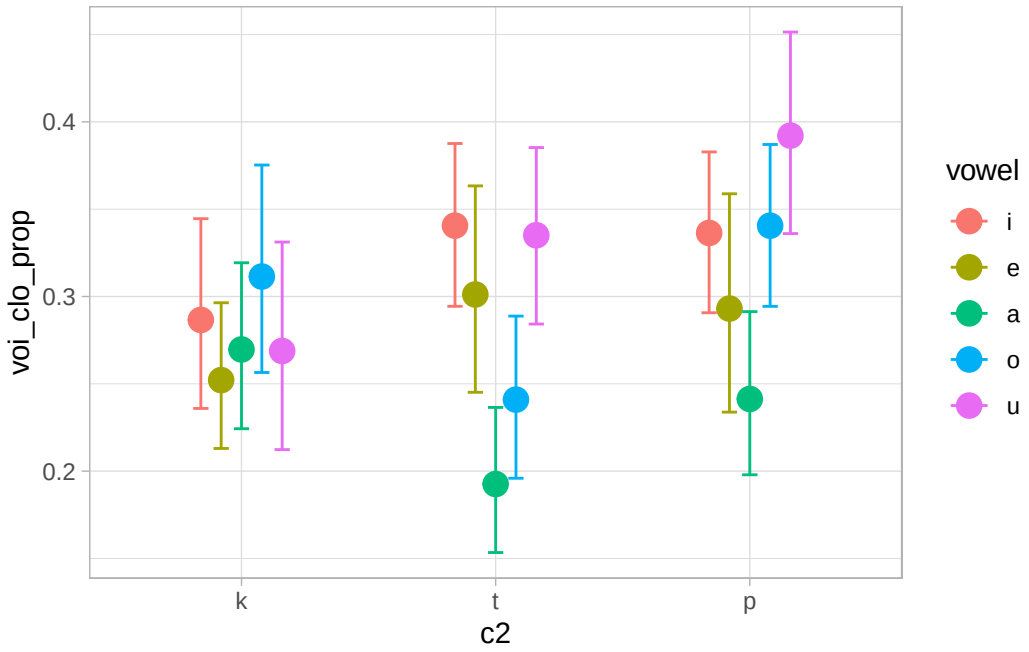
```
voi_prop_bm_diff |>
  mutate(c2p_tk_diff_ratio = exp(c2p_tk_diff)) |>
  reframe(
    # 90% CrI
    q90 = quantile2(c2p_tk_diff_ratio, probs = c(0.05, 0.95)),
    # 80% CrI
    q80 = quantile2(c2p_tk_diff_ratio, probs = c(0.1, 0.9)),
    # 60% CrI
    q60 = quantile2(c2p_tk_diff_ratio, probs = c(0.2, 0.8)),
  ) |>
  # round to 2 digits
  mutate(across(everything(), ~round(.x, 2)))
```

```
# A tibble: 2 x 3
    q90   q80   q60
  <dbl> <dbl> <dbl>
1  1.07  1.1   1.14
2  1.38  1.34  1.3
```

So, based on the model and data, there is a 90% probability that the voicing proportion in /p/ is 1.07-1.38 times longer (or 7-38% increase) than in /t, k/. At 80% confidence, the change ratio is 1.10-1.34 (or 10-34% increase) while at 60% confidence is 1.14-1.30 (14-30% increase). In other words we can be quite confident that the voicing proportion in /p/ is longer than in /t, k/ and that the increase is less than 35%.

brms comes with a convenient function, `conditional_effects()`, to plot posterior means and CrIs based on predictors in the model. In the following example, we plot the predicted proportion of voicing by consonant and vowel.

```
conditional_effects(voi_prop_bm, "c2:vowel")
```



Finally, the package marginaleffects [XXX] has two other convenience functions that return CrIs of comparisons across predictor levels (`avg_comparisons()`) and CrIs of posterior predictions across predictor levels (`avg_predictions`). [XXX]

```
library(marginaleffects)

avg_comparisons(voi_prop_bm, variables = list(c2 = "pairwise"), conf_level = 0.8, type = "link")
```

```
          Contrast Estimate   10.0 % 90.0 %
 mean(t) - mean(k)   0.0348 -0.00418 0.0738
 mean(p) - mean(k)   0.2176  0.17837 0.2558
 mean(p) - mean(t)   0.1829  0.14474 0.2192

Term: c2
Type:  link
Columns: term, contrast, estimate, conf.low, conf.high, predicted_lo, predicted_hi, predicted, tmp_idx
```

```
avg_predictions(voi_prop_bm, variables = "vowel", conf_level = 0.8)
```

```
 vowel Estimate 10.0 % 90.0 %
     i    0.331  0.325  0.338
     e    0.300  0.293  0.307
     a    0.245  0.238  0.252
     o    0.305  0.298  0.312
     u    0.348  0.341  0.354

Type:  response
Columns: vowel, estimate, conf.low, conf.high
```

4. Case study 2: coarticulatory vowel nasalisation

For the second case study we will use data from Carignan et al. (2021). The study looked at properties of nasality in German VNC sequences. Here, we will focus on the effect of C voicing (voiceless /t/ vs voiced /d/) on the proportion of nasalisation within the vowel in the VNC sequence. Previous work on coarticulatory nasalisation in English has suggested that vowels followed by an NC sequence where C is voiceless (NT) should show earlier coarticulatory nasalisation than vowels followed by an NC sequence where C is voiced (ND, see review in Carignan et al., 2021). This pattern has been suggested to be driven by the articulatory and acoustic incompatibility of voicelessness and nasalisation, by which the velum opening gesture of the nasal consonant is pushed away (i.e. earlier) when the consonant following the nasal is voiceless. Everything else being equal, a greater proportion of vowel nasalisation (from the perspective of time) should be found in vowels followed by NT than in vowels followed by ND.

We will model the proportion of coarticulatory nasalisation in the German short vowels /i, e, a, o, u/ when followed by /nt/ or /nd/, using a Bayesian beta regression model. The proportion was calculated as the proportion of the nasal interval to the duration of the vowel. The nasal interval was defined thus: the interval between the time of peak velocity of velum opening to the offset of the vowel. We will use the results of the regression model to answer the following questions:

1. Is the nasalisation proportion, on average across vowels, greater in voiceless NC sequences?
2. Is there individual speaker variation?

First, let's read and plot the data.

```
nasal <- read_csv("data/carignan2021/nasal.csv")
nasal
```

```
# A tibble: 705 x 6
   speaker label               vowel NC    voicing   nas_prop
   <chr>   <chr>               <chr> <chr> <chr>        <dbl>
 1 S03     b__U_nt_@___N_B17/s u     nt    voiceless  0.367
 2 S03     b__a_nd_@___N_B19/s a     nd    voiced     0.195
 3 S03     b__a_nt_@___N_B15/s a     nt    voiceless  0.279
 4 S03     f__I_nt_@___N_B05/s i     nt    voiceless  0.764
 5 S03     l__I_nd_@___N_B06/s i     nd    voiced     0.00529
 6 S03     p__E_nt_____N_B09/s e     nt    voiceless  0.335
 7 S03     r__a_nt_@___N_B06/s a     nt    voiceless  0.243
 8 S03     v__I_nd_@___N_B07/s i     nd    voiced     0.0248
 9 S03     v__I_nt_6___N_B15/s i     nt    voiceless  0.135
10 S03     z__E_nd_@___N_B17/s e     nd    voiced     0.538
# i 695 more rows
```

- `speaker` indicates the speaker ID.
- `label` is the word label as given in the original data.
- `vowel` is the target vowel in the VNC sequence.
- `NC` is the NC sequence.
- `voicing` indicates the voicing of C.
- `nas_prop` is the proportion of coarticulatory nasalisation of the vowel.

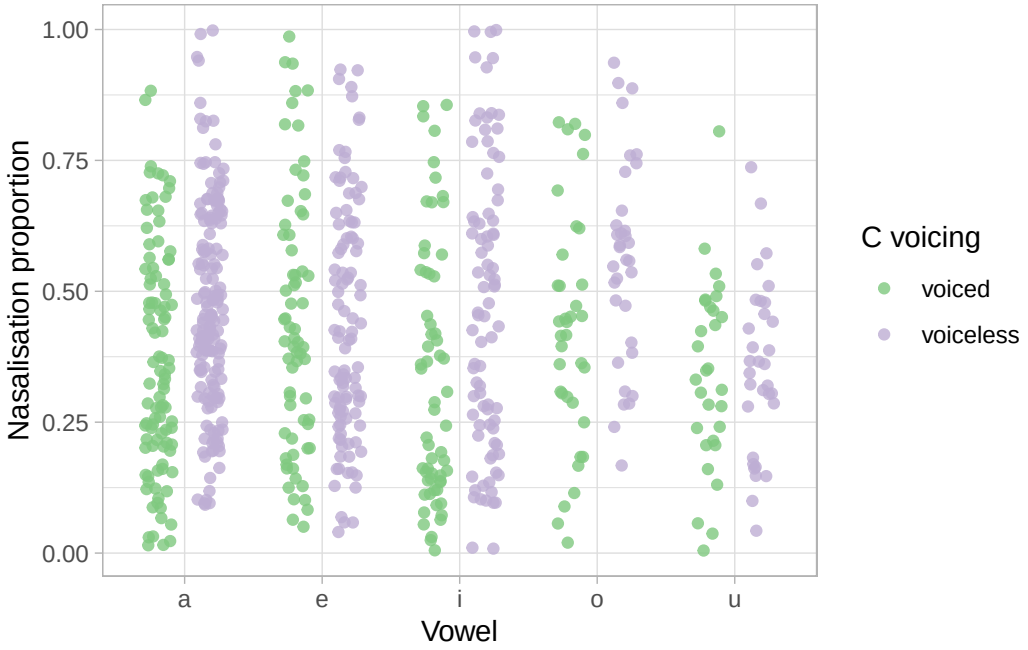


Figure 4: Proportion of coarticulatory nasalisation during the vowel in VNC sequences in German, depending on C voicing.

Figure 4 shows the proportion of coarticulatory nasalisation in vowels followed by /nd/ (voiced) vs /nt/ (voiceless) sequences, for the short vowels /i, e, a, o, u/. We can see a pattern of higher nasalisation proportion in vowels followed by /nt/, at least in the vowels /a, i, o/. For /e, u/, the distribution of nasalisation proportion seems to be similar between the voiced and voiceless contexts.

Now onto modelling with a beta regression. Note that a full appropriate model would include further predictors (both constant and varying), but for simplicity here we include only the following predictors: voicing (voiced /nd/ vs voiceless /nt/), vowel (/i, e, a, o, u/), including an interaction between them. As varying terms, we include a varying intercept by speaker and a by-speaker varying slope for voicing and vowel in interaction. As with the model from the first case study, voicing and vowel are coded using indexing, by suppressing the intercept with `0 +`. Here's the code of the model.

```
nas_prop_bm <- brm(
  nas_prop ~ 0 + voicing:vowel + (0 + voicing:vowel | speaker),
  data = nasal,
  family = Beta,
  cores = 4,
  seed = 3749,
  file = "data/cache/nas_prop_bm"
)
```

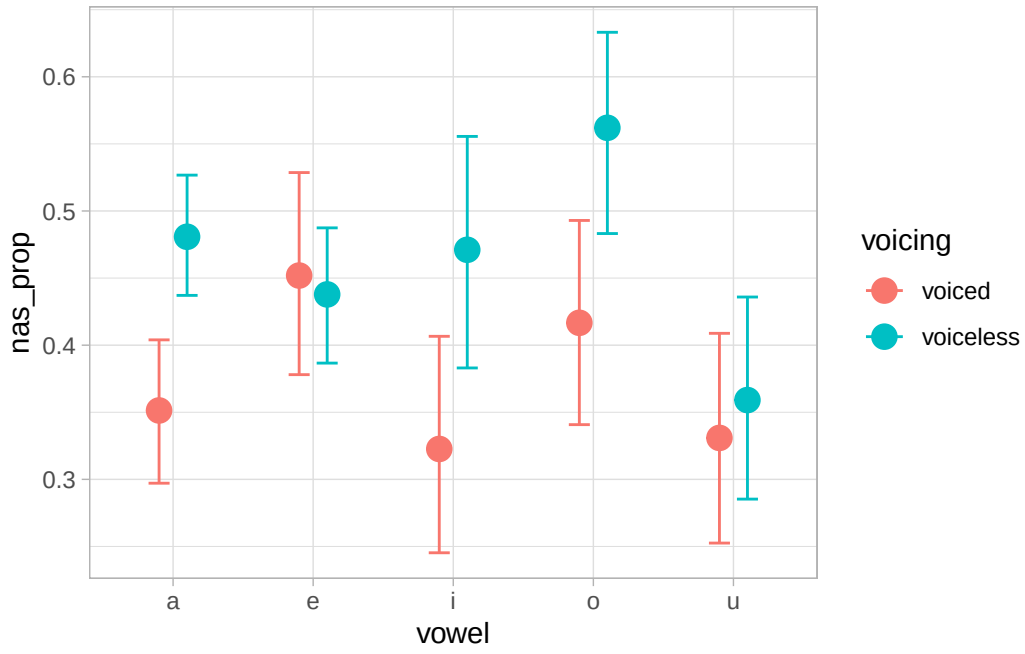
Let's inspect the output of `fixef()`.

```
round(
  fixef(nas_prop_bm, prob = c(0.1, 0.9)),
  digits = 2
)
```

```
                        Estimate Est.Error   Q10   Q90
voicingvoiced:vowela       -0.61      0.12 -0.77 -0.46
voicingvoiceless:vowela    -0.08      0.09 -0.19  0.04
voicingvoiced:vowele       -0.19      0.16 -0.38  0.01
voicingvoiceless:vowele    -0.25      0.10 -0.38 -0.12
voicingvoiced:voweli       -0.74      0.19 -0.98 -0.51
voicingvoiceless:voweli    -0.12      0.18 -0.35  0.10
voicingvoiced:vowelo       -0.34      0.16 -0.54 -0.14
voicingvoiceless:vowelo     0.25      0.15  0.05  0.44
voicingvoiced:vowelu       -0.71      0.18 -0.94 -0.49
voicingvoiceless:vowelu    -0.58      0.16 -0.78 -0.38
```

Negative log-odds indicate a proportion that is smaller than 50%, while positive log-odds a proportion that is greater than 50%. Generally, the expected log-odd predictions are negative, indicating an overall tendency for the nasalisation to take less than 50% of the duration of the vowel. Moreover, the predictions are higher for voiceless NC sequences than for voiced NC sequences, indicating a greater proportion of nasalisation in the former. However there is vowel-specific variation, and there doesn't seem to be much of a difference in nasalisation proportion in /e/ and /u/. Plotting the expected predictions with `conditional_effects()` should make this clearer.

```
conditional_effects(nas_prop_bm, "vowel:voicing")
```



Now that we fitted the model we can use the draws to answer the two research questions (repeated from above):

1. Is the nasalisation proportion, on average across vowels, greater in voiceless NC sequences?
2. Is there individual speaker variation?

To answer question 1, we can calculate the average difference in nasalisation proportion by first calculating the average nasalisation across all vowels for voiced and voiceless sequences and then take the difference of those, similarly to what we have done in the Case Study 1 above.

```r
# extract only coefficient variables starting with "b_"
nas_prop_bm_draws <- as_draws_df(nas_prop_bm, variable = "^b_", regex = TRUE)

nas_prop_bm_draws_long <- nas_prop_bm_draws |>
  # pivot vowel:c2 columns
  pivot_longer(`b_voicingvoiced:vowela`:`b_voicingvoiceless:vowelu`, names_to = "coeff") |>
  # separate "coeff" labels into type ("b"), vowel and c2
  separate(coeff, into = c("type", "voicing", "vowel"))
```

```
Warning: Dropping 'draws_df' class as required metadata was removed.
```

```r
nas_prop_bm_draws_long
```

```
# A tibble: 40,000 x 7
   .chain .iteration .draw type  voicing          vowel    value
    <int>      <int> <int> <chr> <chr>            <chr>    <dbl>
 1      1          1     1 b     voicingvoiced    vowela -0.463
 2      1          1     1 b     voicingvoiceless vowela -0.0944
 3      1          1     1 b     voicingvoiced    vowele -0.155
 4      1          1     1 b     voicingvoiceless vowele -0.208
 5      1          1     1 b     voicingvoiced    voweli -0.481
 6      1          1     1 b     voicingvoiceless voweli -0.0529
 7      1          1     1 b     voicingvoiced    vowelo -0.310
 8      1          1     1 b     voicingvoiceless vowelo  0.321
 9      1          1     1 b     voicingvoiced    vowelu -0.620
10      1          1     1 b     voicingvoiceless vowelu -0.368
# i 39,990 more rows
```

Now let's calculate the mean nasalisation proportion within each draw by voicing, and plot the resulting posterior distributions. Note that, as discussed for Case study 1, when working wit log-odds it is important to first do all necessary calculations in log-odds, here calculate the mean log-odds across vowels, and *then* transform the calculated estimands to proportions/probabilities.

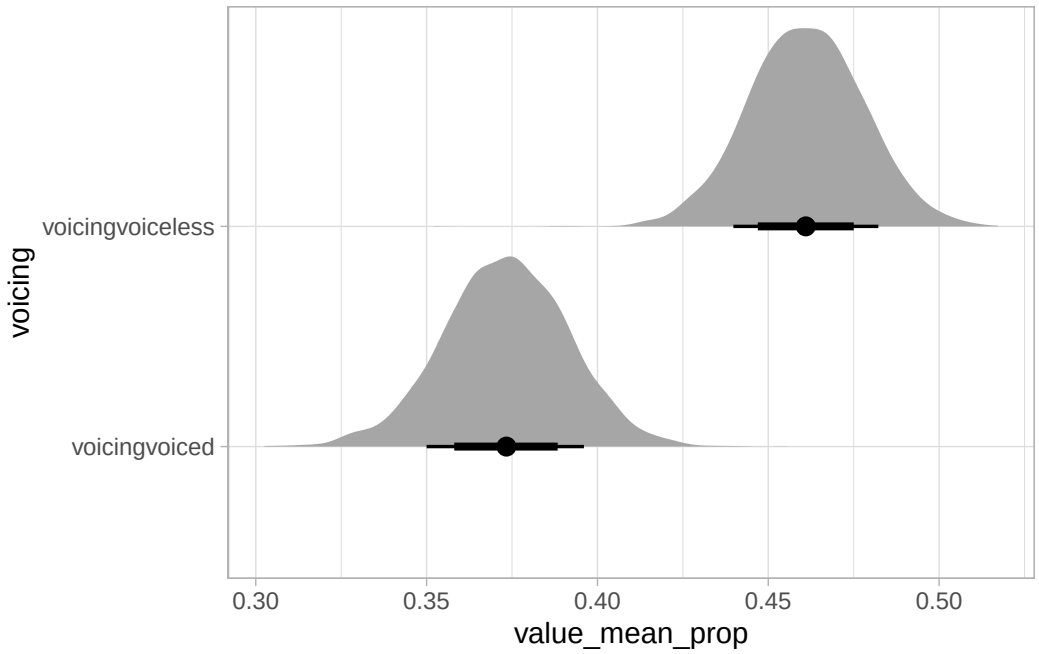
```r
nas_prop_bm_draws_long_voicing <- nas_prop_bm_draws_long |>
  # grouping by .draw and voicing ensures that averaging applies only within draw and voicing
  group_by(.draw, voicing) |>
  summarise(
    # calculate the mean value within draw/voicing in log-odds
    value_mean = mean(value),
    # we can now transform log-odds to proportion with plogis()
    value_mean_prop = plogis(value_mean),
    .groups = "drop"
  )

nas_prop_bm_draws_long_voicing
```

```
# A tibble: 8,000 x 4
   .draw voicing          value_mean value_mean_prop
   <int> <chr>                 <dbl>           <dbl>
```

```
 1     1 voicingvoiced       -0.406           0.400
 2     1 voicingvoiceless    -0.0804          0.480
 3     2 voicingvoiced       -0.391           0.404
 4     2 voicingvoiceless    -0.123           0.469
 5     3 voicingvoiced       -0.444           0.391
 6     3 voicingvoiceless    -0.132           0.467
 7     4 voicingvoiced       -0.439           0.392
 8     4 voicingvoiceless    -0.0982          0.475
 9     5 voicingvoiced       -0.383           0.405
10     5 voicingvoiceless    -0.266           0.434
# i 7,990 more rows
```

```
nas_prop_bm_draws_long_voicing |>
  ggplot(aes(value_mean_prop, voicing)) +
  stat_halfeye(.width = c(0.6, 0.8))
```



The plot suggests an overall greater nasalisation proportion in voiceless NC sequences. Let's quantify how greater as we did in Case Study 1. We will use odd-ratios in this context as well, i.e. we will convert log-odds to odd-ratios using the `exp()` (exponential) function (and as before we first calculate the difference and then exponentiate the resulting values, after which we can take summary measures, like means and quantile-based measures such as CrIs).

```
nas_prop_bm_diff <- nas_prop_bm_draws_long_voicing |>
  # pivot data to create one column per voicing with the mean drawn values,
  # with one draw per raw. we need to drop the value_mean_prop col
  select(-value_mean_prop) |>
  pivot_wider(names_from = voicing, values_from = value_mean) |>
  mutate(
    # calculate the difference of voiceless and voiced in log-odds
```

```
    voicing_diff = voicingvoiceless - voicingvoiced,
    # now transform with exp() to get the ratio difference
    voicing_diff_ratio = exp(voicing_diff)
  )
nas_prop_bm_diff
```

```
# A tibble: 4,000 x 5
   .draw voicingvoiced voicingvoiceless voicing_diff voicing_diff_ratio
   <int>         <dbl>            <dbl>        <dbl>               <dbl>
 1     1        -0.406          -0.0804        0.325                1.38
 2     2        -0.391           -0.123        0.268                1.31
 3     3        -0.444           -0.132        0.312                1.37
 4     4        -0.439          -0.0982        0.340                1.41
 5     5        -0.383           -0.266        0.117                1.12
 6     6        -0.541          -0.0903        0.450                1.57
 7     7        -0.512           -0.111        0.401                1.49
 8     8        -0.493           -0.152        0.341                1.41
 9     9        -0.527           -0.175        0.352                1.42
10    10        -0.491           -0.257        0.234                1.26
# i 3,990 more rows
```

```
nas_prop_bm_diff |>
  reframe(
    # 90% CrI
    q90 = quantile2(voicing_diff_ratio, probs = c(0.05, 0.95)),
    # 80% CrI
    q80 = quantile2(voicing_diff_ratio, probs = c(0.1, 0.9)),
    # 60% CrI
    q60 = quantile2(voicing_diff_ratio, probs = c(0.2, 0.8)),
  ) |>
  mutate(across(everything(), ~round(.x, 2)))
```

```
# A tibble: 2 x 3
    q90   q80   q60
  <dbl> <dbl> <dbl>
1  1.23  1.27  1.33
2  1.69  1.63  1.56
```

The CrIs of the ratio difference in nasalisation proportion in voiceless vs voiced NC sequences suggest a robust increase of nasalisation in the voiceless NC sequences, with a 90% probability that the increase is between 23% and 69% of the proportion in voiced NC sequences.

Moving onto question 2: is there individual speaker variation? [TAMMINGA XXX] For this, we will use the `spread_draws()` function from tidybayes [XXX] to extract the draws of the varying terms (in brms these are the coefficients that start with `r_`). There is quite a few steps of processing to get from the raw draws to the estimand we need: while we have commented the following code, we encourage readers to test each line sequentialy and inspect the intermediate output to fully understand the process. We assume that readers are familiar enough with models with varying terms (aka random effects, mixed-effects models). What readers should note is that to obtain the expected predictions of nasalisation proportion for each speaker, the constant terms and the varying terms should be added (since the varying terms indicate the deviation of each speaker from the overall estimate).

```r
library(tidybayes)

nas_prop_r <- nas_prop_bm |>
  # extract draws from model, only `r_speaker` varying terms
  spread_draws(r_speaker[speaker,voicingvowel]) |>
  # separate the column voicingvowel to two columns
  separate(voicingvowel, c("voicing", "vowel")) |>
  # join the draws with the `b_` terms
  left_join(y = nas_prop_bm_draws_long) |>
  # get the expected log-odd value of each speaker, in each draw
  # this is the sum of the `value` from the b_ terms and the value from the
  # r_speaker term.
  mutate(r_speaker_value = value + r_speaker) |>
  # group the data for summarise
  group_by(.draw, speaker, voicing) |>
  # get mean expected log-odds by draw, speaker and voicing (averaging across vowel)
  summarise(r_speaker_value_mean = mean(r_speaker_value)) |>
  # make the data wider: two columns, one for voiced and one for voiceless
  pivot_wider(names_from = voicing, values_from = r_speaker_value_mean) |>
  # finally, calculate the difference in expected log-odds of voiceless and voiced
  mutate(voicing_diff = voicingvoiceless - voicingvoiced)

nas_prop_r
```

```
# A tibble: 140,000 x 5
# Groups:   .draw, speaker [140,000]
   .draw speaker voicingvoiced voicingvoiceless voicing_diff
   <int> <chr>           <dbl>            <dbl>        <dbl>
 1     1 S03          -1.42             0.0676      1.49
 2     1 S04           0.209            0.185      -0.0238
 3     1 S05          -0.284           -0.0546      0.229
 4     1 S06           0.00669         -0.324      -0.331
 5     1 S07          -0.446           -0.147       0.299
 6     1 S08          -0.610           -0.127       0.483
 7     1 S09          -0.473            0.0497      0.523
 8     1 S10          -0.556           -0.311       0.245
 9     1 S11           0.201           -0.426      -0.628
10     1 S12          -0.0812          -0.0856     -0.00439
# i 139,990 more rows
```

Figure 5 plots the posterior distributions of the expected log-odd difference of coarticulatory nasalisation in voiceless vs voiced NC sequences (x-axis), for each speaker in the data (y-axis), as predicted by the model. The red solid vertical line indicates the constant (overall) expected log-odd difference based on the (constant) `b_` terms. The black dashed vertical line marks log-difference 0 (i.e., no difference in proportion of nasalisation between voiceless and voiced NC).

```r
nas_prop_r |>
  ggplot(aes(voicing_diff, reorder(speaker, voicing_diff))) +
  stat_halfeye() +
  geom_vline(xintercept = mean(nas_prop_bm_diff$voicing_diff), colour = "red") +
```

```
  geom_vline(xintercept = 0, linetype = "dashed") +
  lims(x = c(-1, 1.5))
```

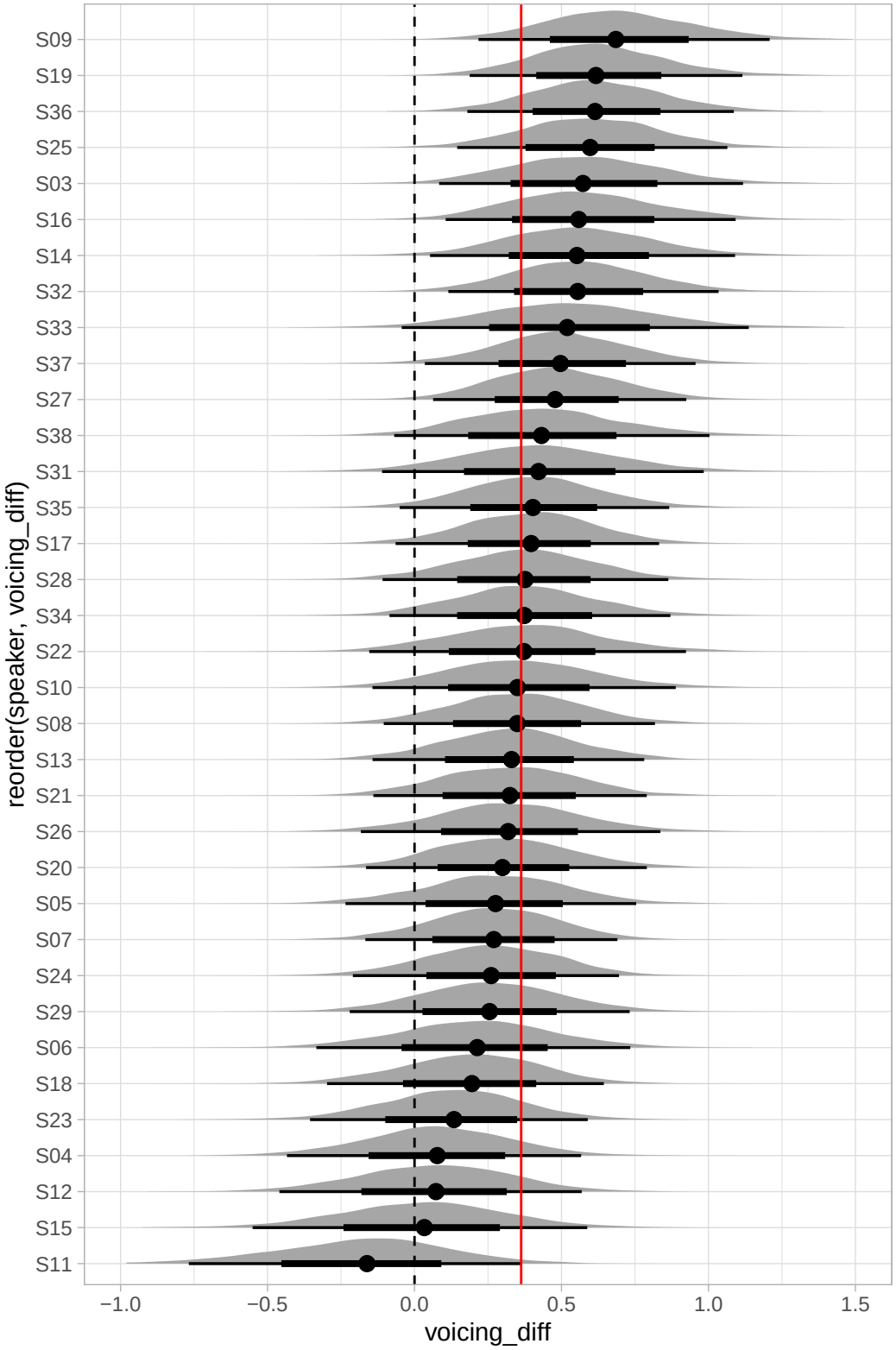

S09
S19
S36
S25
S03
S16
S14
S32
S33
S37
S27
S38
S31
S35
S17
S28
S34
S22
S10
S08
S13
S21
S26
S20
S05
S07
S24
S29
S06
S18
S23
S04
S12
S15
S11
reorder(speaker, voicing_diff)
−1.0
−0.5
0.0
0.5
1.0
1.5
voicing_diff

Figure 5

There is a lot of uncertainty within and between speakers: while the distributions of most speakers are located in the positive range, some expected distributions (see last 5 speakers at the bottom of figure) do substantially span both negative and positive values. In other words, while most speakers are more likely to have a larger nasalisation proportion in voiceless NC sequences, a few might in fact have the opposite pattern. Even among those speakers that do have a more robust positive difference, there is a lot of uncertainty as to the magnitude of the difference.

References

Carignan, C., Coretta, S., Frahm, J., Harrington, J., Hoole, P., Joseph, A., Kunay, E., Voit, D., 2021. Planting the seed for sound change: Evidence from real-time MRI of velum kinematics in German. Language 97, 333–364. doi:10.1353/lan.2021.0020.