Supplement for Planting the seed for sound change: evidence from real-time MRI of velum kinematics in German

Contents

```r
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_knit$set(root.dir = here::here())
library(tidyverse)
library(patchwork)
```

```r
library(brms)
library(tidybayes)
library(extraDistr)
library(HDInterval)

core.num <- parallel::detectCores()
options(mc.cores = core.num)

my.seed <- 123
set.seed(my.seed)
```

1 Prepare data

```r
matdat <- read.csv("./rtMRI-velum/velum_data.csv", header = T)

# get rid of items with gesture onsets that begin *after* the vowel offset (only 5/7152 total items)
matdat <- matdat[(matdat$velumopening_gesture_on - matdat$Vokal_off) < 0,]

# separate coda contexts by alveolar voiced stop vs. alveolar voiceless stop
voiceless <- c("nt__", "nt_@", "nt_6", "nt_a")
voiced    <- c("nd_@", "nd_6", "nd_a")

matdat$voicing <- c()
matdat$voicing[matdat$post %in% voiceless]  <- "voiceless"
matdat$voicing[matdat$post %in% voiced]     <- "voiced"

# only include alveolar nasal items preceding a voiced or voiceless stop consonant
coda <- c(voiceless, voiced)

# only include neutrally stressed utterances
stresses <- c("N")

# subset the data
subdat <- matdat[matdat$post %in% coda & matdat$stress %in% stresses, ]
```

2 Velum gesture duration

Since no specific information on expectations for velum gesture duration can be found in the literature, we followed more general expectations for speech segment durations, coded as very weakly informative priors. The BRM for velum gesture duration was built using a log-normal distribution, since speech segment duration has been shown to be log-normally distributed (Rosen 2005; Ratnikova 2017; Gahl and Baayen 2019). The following distributions were used as weakly informative priors (on the log-odds scale): for the intercept of duration (corresponding to ND), a normal distribution with mean = 0 and standard deviation = 3 ($Normal(0,3)$); for the effect of voicing (when NT), $Normal(0,1)$. These roughly correspond to a belief that the intercept of velum gesture duration (when the context is ND) is between 0 and 400 ms ($e^{3\times2} = 403$), and the duration changes (increases or decreases) by a factor of 1 to 7.4 in the ND context ($e^{1\times2} = 7.4$), at 95% confidence. For the model standard deviation and the random intercept standard deviation we used a half-Cauchy distribution with location 0 and scale 0.1 ($HalfCauchy(0, 0.1)$); this corresponds to a 95% HDI = [0, 2.55] in log-odds, i.e. a factor change of 1 to 12.75 ($e^2.55 = 12.75$).

```r
inverseCDF(c(0.025, 0.975), phcauchy, sigma = 0.1)
```

```
## [1] 0.003930135 2.545175934
```

For the correlation between random effects, we used an $LKJ(2)$ distribution, as recommended by Vasishth et al. (2018). The same prior specification was used for the models of velum gesture onset time and velum gesture offset time.

NOTE: Warnings about $\hat{R}$ and sample size are false warnings, and are not signs of problems with fitting and they can safely be ignored. See https://github.com/paul-buerkner/brms/issues/864.

```r
# create the dependent variable
subdat$DV <- subdat$velumopening_gesture_dur*1000

# full model
dur <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 1), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/dur",
  save_all_pars = TRUE
)
```

```r
dur
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##     Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 6000; warmup = 3000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.12      0.02     0.09     0.15 1.00
## sd(voicingvoiceless)              0.08      0.01     0.06     0.12 1.00
## cor(Intercept,voicingvoiceless)  -0.29      0.18    -0.60     0.09 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     5422     7471
## sd(voicingvoiceless)              5663     8113
## cor(Intercept,voicingvoiceless)   6487     8256
```

3

```
##
## ~word (Number of levels: 30)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.07      0.01     0.04     0.10 1.00
## sd(voicingvoiceless)               0.03      0.02     0.00     0.07 1.00
## cor(Intercept,voicingvoiceless)   -0.14      0.37    -0.78     0.64 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      5700     6103
## sd(voicingvoiceless)               1766     4041
## cor(Intercept,voicingvoiceless)   11075     8197
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           5.71      0.03     5.66     5.76 1.00     4706     7006
## voicingvoiceless   -0.11      0.02    -0.15    -0.07 1.00     7244     7945
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.14      0.00     0.14     0.15 1.00    17280     9078
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
# reduced/null model
dur_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/dur_null",
  save_all_pars = TRUE
)
```

```r
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(dur, dur_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
```
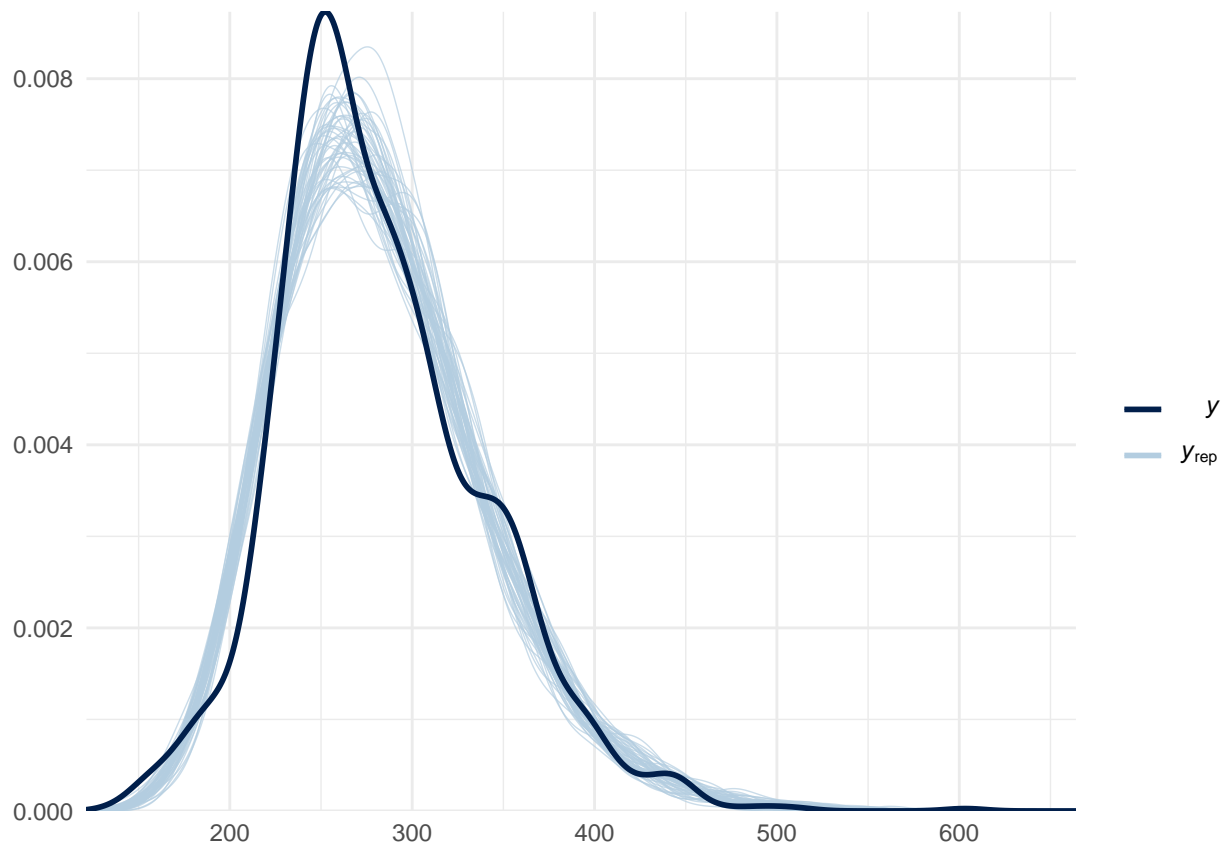
```
## Iteration: 7
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
```

```
## Estimated Bayes factor in favor of dur over dur_null: 533.96567
```

```
# calculate the marginal posteriors of the full model
dur_post <- brms::posterior_samples(dur, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept) * exp(b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```

2.1   Checks

```
pp_check(dur, nsamples = 50) + theme_minimal()
```
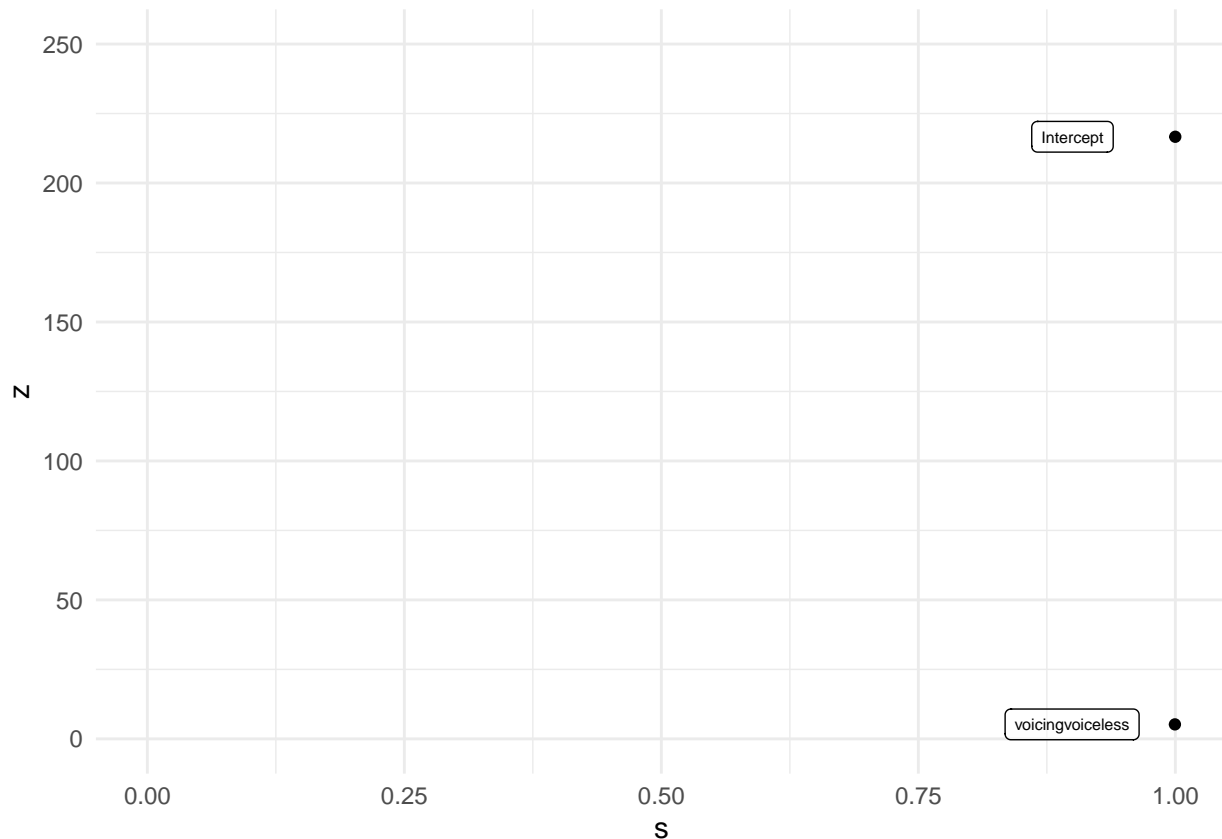

For the sensitivity analysis we followed the method introduced in Betancourt (2018). In the following plot, s (posterior shrinkage) indicates the contribution of the data to the posterior (values closer to 1 are better). z (posterior z-score) indicates how close the inference is to the data generating process (lower values are better). An ideal fit should have $s = 1$ and $z = 0$. Posteriors with high s and high z indicate overfit (the data dominate the posterior), while posteriors with low s and low z indicate that the model is poorly

identified (the data is not contributing to the model and the priors dominate).

```r
dur_fixed <- fixef(dur) %>% as_tibble(rownames = "term")

dur_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```


3 Velum gesture onset

See velum gesture duration for prior specification.

```r
# create the dependent variable
subdat$DV <- -(subdat$velumopening_gesture_on - subdat$Vokal_off)*1000

# full model
```

```r
onset <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 1), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/onset",
  save_all_pars = TRUE
)
```

```
onset
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.17      0.02     0.13     0.22 1.00
## sd(voicingvoiceless)               0.03      0.02     0.00     0.09 1.00
## cor(Intercept,voicingvoiceless)   -0.21      0.39    -0.84     0.65 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      1992     3448
## sd(voicingvoiceless)               1339     2749
## cor(Intercept,voicingvoiceless)    6310     5205
##
## ~word (Number of levels: 30)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.23      0.04     0.17     0.32 1.00
## sd(voicingvoiceless)               0.05      0.03     0.00     0.13 1.00
## cor(Intercept,voicingvoiceless)   -0.39      0.37    -0.93     0.49 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      2193     3821
## sd(voicingvoiceless)               1333     1861
## cor(Intercept,voicingvoiceless)    4879     4686
##
## Population-Level Effects:
##                  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## Intercept              4.63      0.06      4.52      4.74 1.00      1427      2639
## voicingvoiceless       0.12      0.03      0.05      0.18 1.00      4824      4270
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.31      0.01      0.29      0.32 1.00      9877      5453
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
# reduced/null model
onset_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/onset_null",
  save_all_pars = TRUE
)
```

```r
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(onset, onset_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
```

```
## Estimated Bayes factor in favor of onset over onset_null: 5.94046
```

```r
# calculate the marginal posteriors of the full model
onset_post <- brms::posterior_samples(onset, pars="b_") %>%
  dplyr::mutate(
```
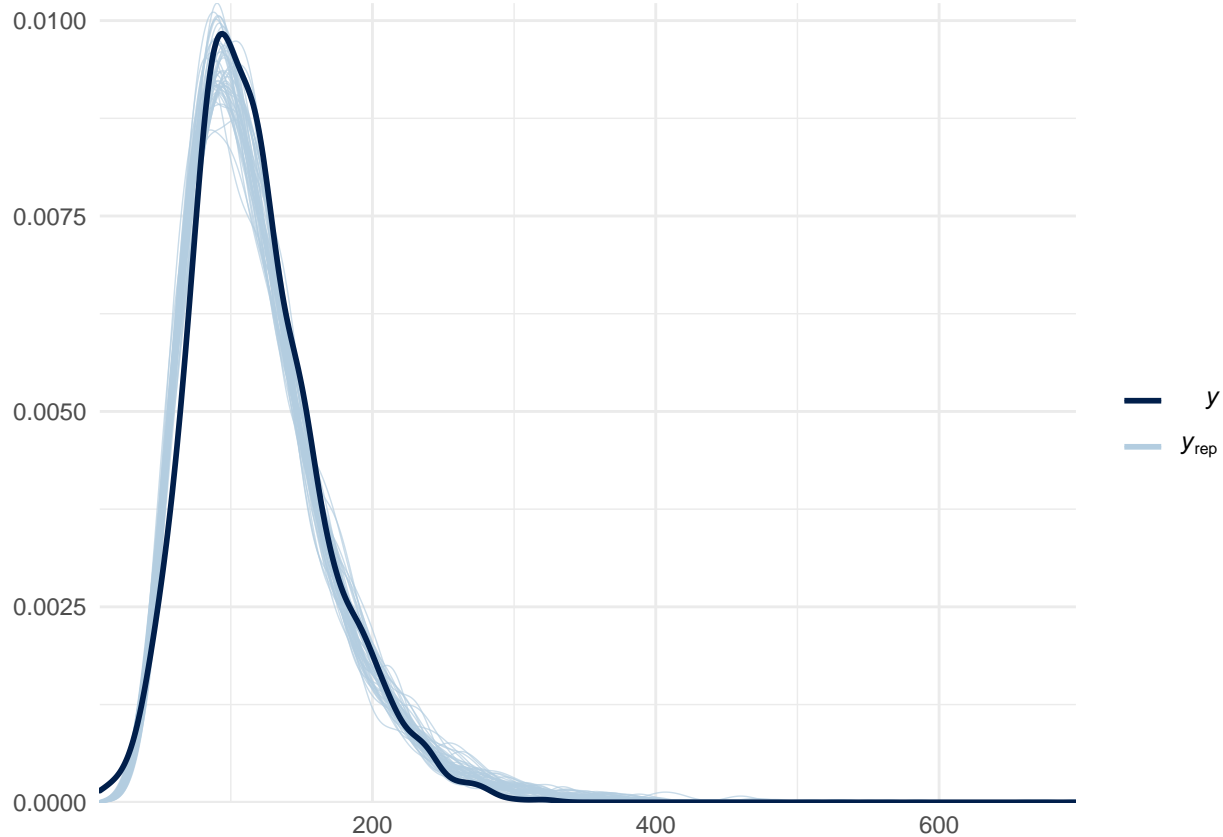
```
    nd = exp(b_Intercept),
    nt = exp(b_Intercept) * exp(b_voicingvoiceless)
) %>%
dplyr::select(nd, nt) %>%
tidyr::gather(context, DV)
```

3.1 Checks

```
pp_check(onset, nsamples = 50) + theme_minimal()
```
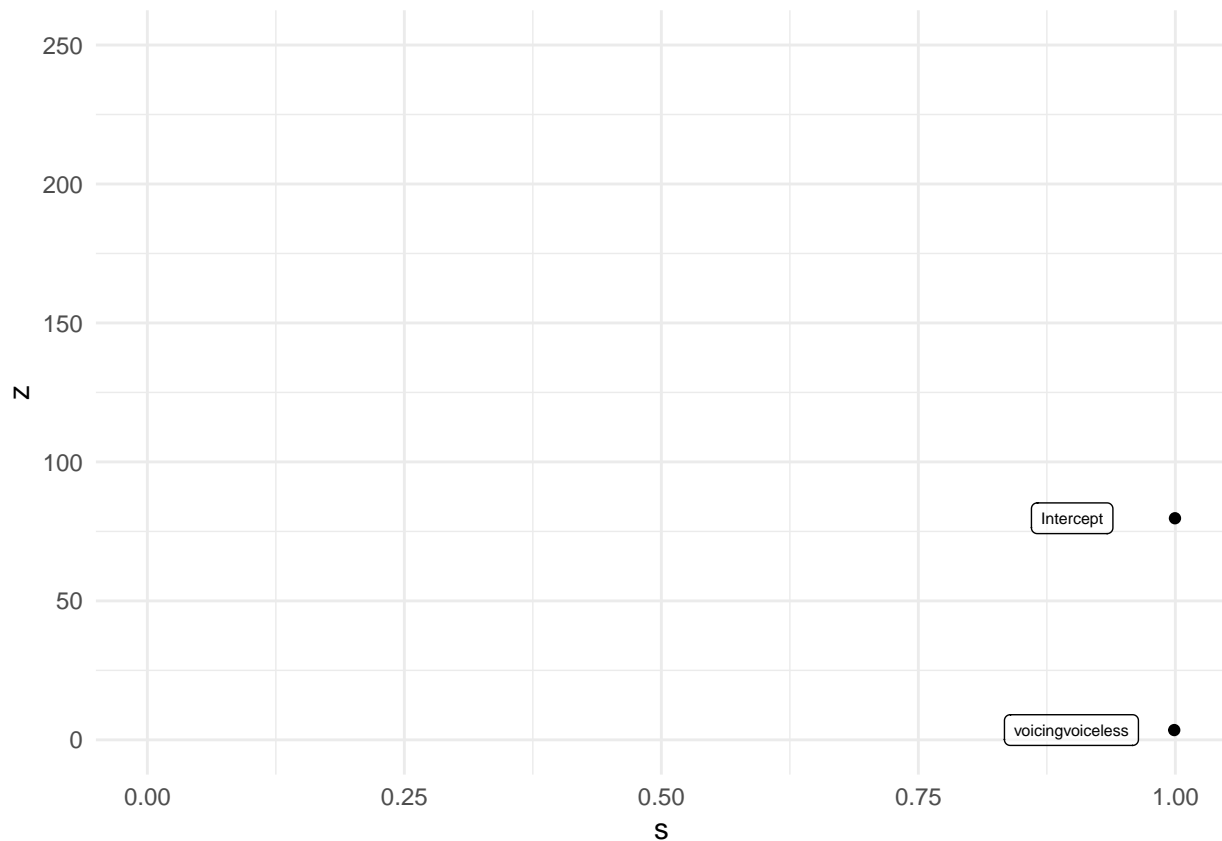


```
onset_fixed <- fixef(onset) %>% as_tibble(rownames = "term")

onset_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

4 Velum gesture peak (timing)

The BRM for the time point of the velum gesture peak was built using a Gaussian distribution; unlike the other measures of timing, the time point of the velum gesture peak is not expected to follow a one-sided (positive support) distribution, since the peak can potentially occur before or after the vowel offset. The following distributions were used as weakly informative priors (on the milliseconds scale): for the intercept (corresponding to ND), $Normal(0, 200)$; for the effect of voicing (when NT), $Normal(0, 100)$. These correspond to a belief that the intercept is between -400 and 400 ms from the vowel offset, and that the time changes by -200 to 200 ms in the NT context, at 95% confidence. For the model standard deviation and the random intercept standard deviation we used $HalfCauchy(0, 5)$, which corresponds to a 95% HDI = [0, 127] ms.

```
inverseCDF(c(0.025, 0.975), phcauchy, sigma = 5)
```

```
## [1]    0.1964676 127.2584987
```

For the correlation between random effects, an $LKJ(2)$ distribution.

```
# create the dependent variable
subdat$DV <- (subdat$velumopening_maxcon_on - subdat$Vokal_off)*1000

# full model
gest.max <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
```

```
    family = gaussian(),
    prior = c(prior(normal(0, 200), class = Intercept),
              prior(normal(0, 100), class = b, coef = voicingvoiceless),
              prior(cauchy(0, 5), class = sd),
              prior(cauchy(0, 5), class = sigma),
              prior(lkj(2), class = cor)),
    seed = my.seed,
    iter = 6000,
    warmup = 3000,
    chains = 4,
    cores = core.num,
    control = list(adapt_delta = 0.99,
                   max_treedepth = 20),
    file = "./rtMRI-velum/models/gest_max",
    save_all_pars = TRUE
)
```

```
gest.max
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 6000; warmup = 3000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      15.97      2.12    12.34    20.60 1.00
## sd(voicingvoiceless)               13.25      1.95     9.83    17.45 1.00
## cor(Intercept,voicingvoiceless)   -0.62      0.12    -0.81    -0.36 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                       3775     6145
## sd(voicingvoiceless)                3350     5750
## cor(Intercept,voicingvoiceless)     4495     6496
##
## ~word (Number of levels: 30)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      15.45      2.63    10.84    21.10 1.00
## sd(voicingvoiceless)                3.65      2.21     0.24     8.66 1.00
## cor(Intercept,voicingvoiceless)   -0.20      0.39    -0.86     0.60 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                       4210     5937
## sd(voicingvoiceless)                1965     3280
## cor(Intercept,voicingvoiceless)    11008     7531
##
## Population-Level Effects:
##                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept            45.86      4.38    37.12    54.38 1.00     2478     4656
## voicingvoiceless    -25.69      3.11   -31.86   -19.62 1.00     4152     6324
##
## Family Specific Parameters:
##         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sigma       18.52       0.37     17.82     19.28 1.00     17060        9155
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
# reduced/null model
gest.max_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = gaussian(),
  prior = c(prior(normal(0, 200), class = Intercept),
            prior(cauchy(0, 5), class = sd),
            prior(cauchy(0, 5), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/gest_max_null",
  save_all_pars = TRUE
)
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(gest.max, gest.max_null)

## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8

## Estimated Bayes factor in favor of gest.max over gest.max_null: 1873906.34322
# calculate the marginal posteriors of the full model
gest.max_post <- brms::posterior_samples(gest.max, pars="b_") %>%
  dplyr::mutate(
    nd = b_Intercept,
    nt = b_Intercept + b_voicingvoiceless
  ) %>%
```
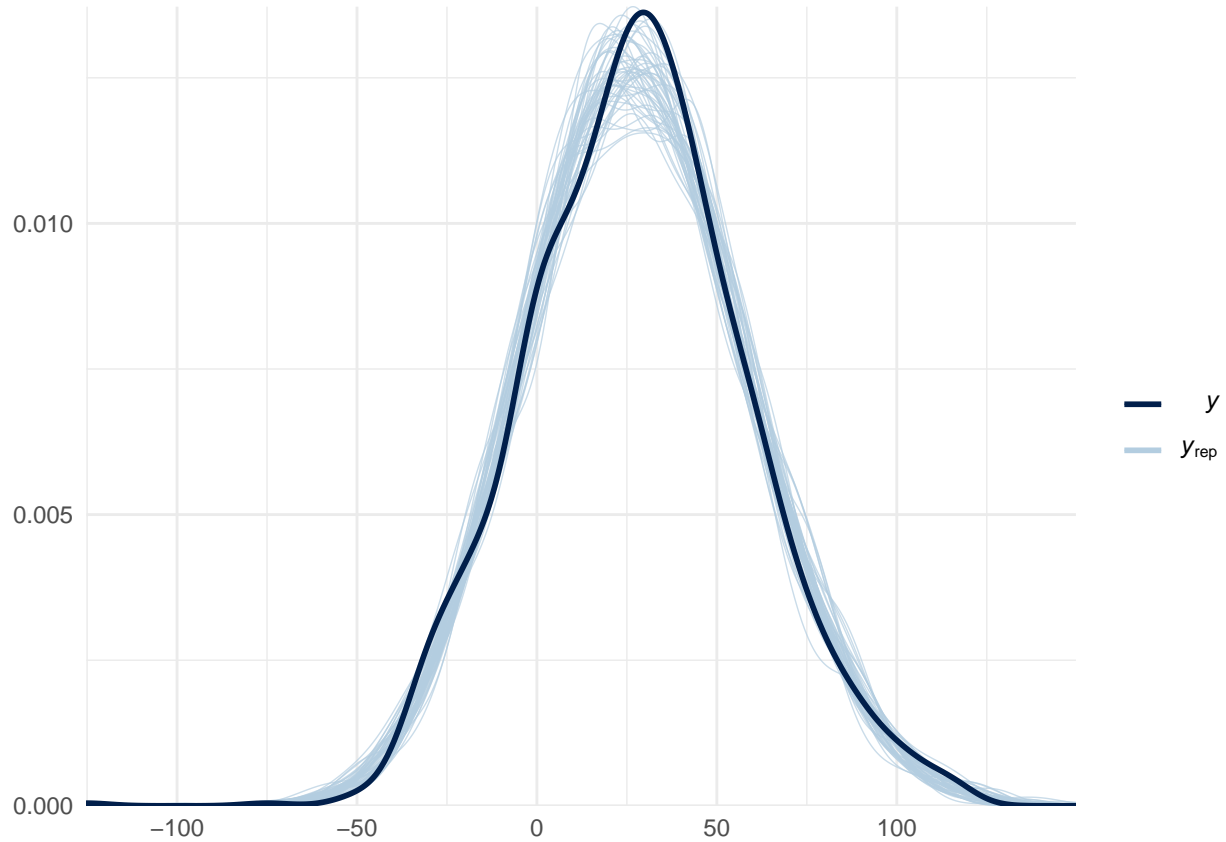
12

```
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
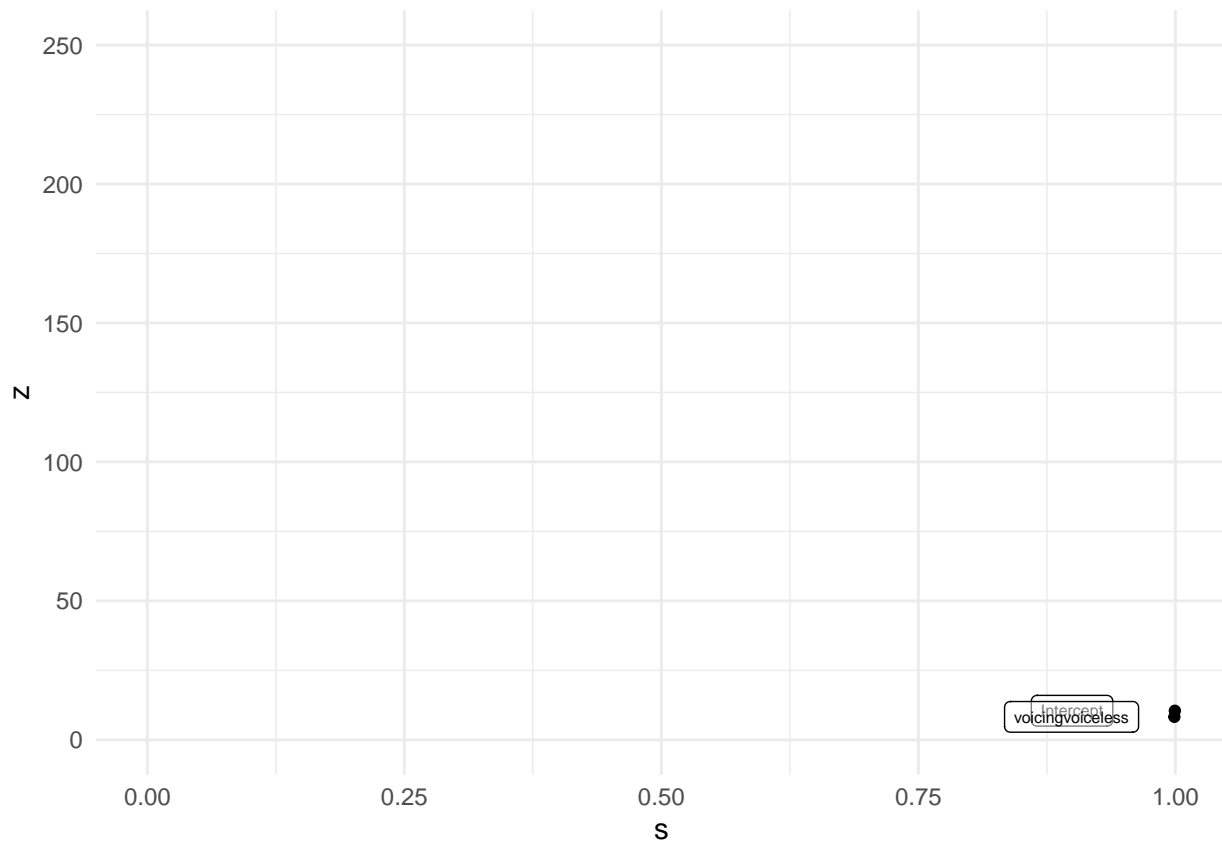
4.1 Checks

```
pp_check(gest.max, nsamples = 50) + theme_minimal()
```



```
gest.max_fixed <- fixef(gest.max) %>% as_tibble(rownames = "term")

gest.max_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(200, 100),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

5 Velum gesture offset

See velum gesture duration for prior specification.

```r
# create the dependent variable
subdat$DV <- (subdat$velumopening_gesture_off - subdat$Vokal_off)*1000

# full model
offset <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 1), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
```

```
  file = "./rtMRI-velum/models/offset",
  save_all_pars = TRUE
)
```

```
offset
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 6000; warmup = 3000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.15      0.02     0.12     0.20 1.00
## sd(voicingvoiceless)               0.11      0.02     0.08     0.15 1.00
## cor(Intercept,voicingvoiceless)   -0.42      0.15    -0.68    -0.09 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      4598     6415
## sd(voicingvoiceless)               4772     7862
## cor(Intercept,voicingvoiceless)    6016     7747
##
## ~word (Number of levels: 30)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.05      0.02     0.02     0.09 1.00
## sd(voicingvoiceless)               0.07      0.02     0.03     0.12 1.00
## cor(Intercept,voicingvoiceless)   -0.15      0.34    -0.76     0.53 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      2182     4699
## sd(voicingvoiceless)               1409     2940
## cor(Intercept,voicingvoiceless)    1627     2582
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           5.28      0.03     5.21     5.34 1.00     2568     5089
## voicingvoiceless   -0.28      0.03    -0.34    -0.22 1.00     2990     5502
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.16      0.00     0.15     0.17 1.00    17295     8738
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
offset_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
```

```r
          prior(cauchy(0, 0.1), class = sd),
          prior(cauchy(0, 0.1), class = sigma),
          prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/offset_null",
  save_all_pars = TRUE
)
```

```r
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(offset, offset_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
```
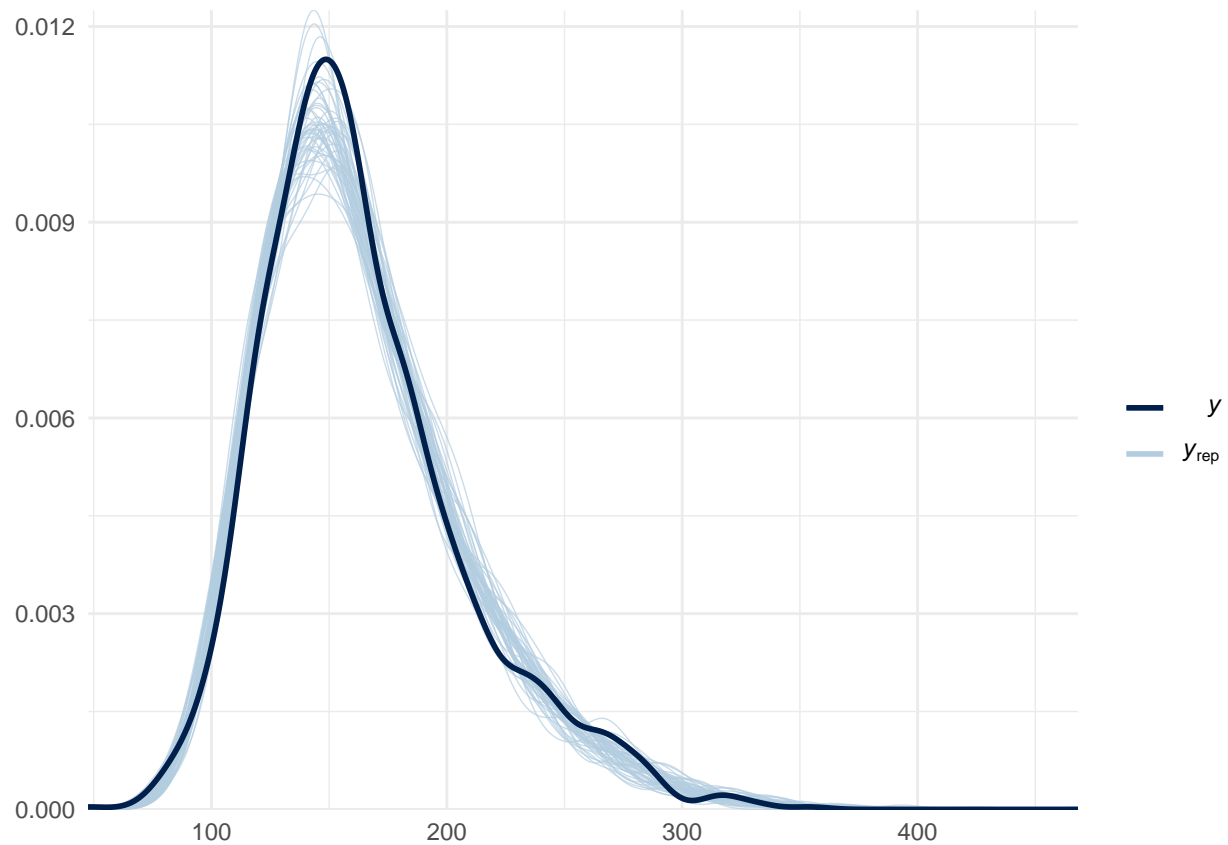
```
## Estimated Bayes factor in favor of offset over offset_null: 142262071.18830
```

```r
# calculate the marginal posteriors of the full model
offset_post <- brms::posterior_samples(offset, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept) * exp(b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
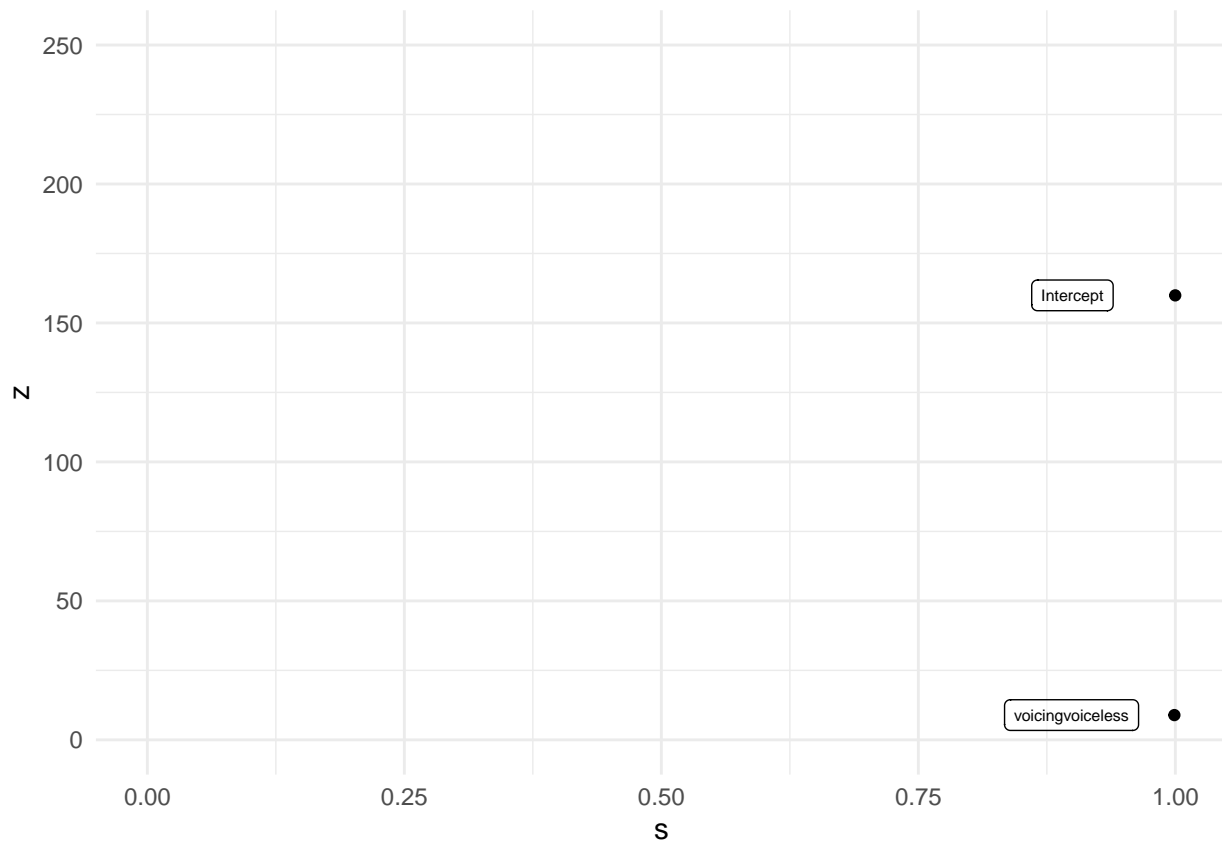
5.1 Checks

```r
pp_check(offset, nsamples = 50) + theme_minimal()
```

```r
offset_fixed <- fixef(offset) %>% as_tibble(rownames = "term")

offset_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

6 Velum gesture peak (magnitude)

The BRM for the velum gesture magnitude was built using a Beta distribution, since the magnitude values are on a 0-1 scale. The following practically flat priors were used: $Normal(0, 10)$ for the intercept; $Normal(0, 5)$ for voicing and the random effects standard deviations; the brms default prior for the ϕ parameter of the beta distribution ($gamma(0.01, 0.01)$); and $LKJ(2)$ prior for the random effects correlation.

```
# create the dependent variable
subdat$DV <- subdat$velum2US_velumopening_maxcon_onset

# full model
gest.max.mag <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = Beta(),
  prior = c(prior(normal(0, 10), class = Intercept),
            prior(normal(0, 5), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 5), class = sd),
            prior(gamma(0.01, 0.01), class = phi),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
```

```
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/gest_max_mag",
  save_all_pars = TRUE
)
```

```
gest.max.mag
```

```
##  Family: beta
##   Links: mu = logit; phi = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                       0.48      0.07     0.37     0.63 1.00
## sd(voicingvoiceless)                0.24      0.05     0.16     0.34 1.00
## cor(Intercept,voicingvoiceless)    -0.49      0.16    -0.75    -0.14 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                       2397     3605
## sd(voicingvoiceless)                3254     4705
## cor(Intercept,voicingvoiceless)     5273     5990
##
## ~word (Number of levels: 30)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                       0.20      0.05     0.12     0.31 1.00
## sd(voicingvoiceless)                0.18      0.06     0.08     0.31 1.00
## cor(Intercept,voicingvoiceless)     0.43      0.30    -0.25     0.90 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                       4047     4985
## sd(voicingvoiceless)                1900     2453
## cor(Intercept,voicingvoiceless)     2661     3856
##
## Population-Level Effects:
##                  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept            0.93      0.10     0.74     1.12 1.00     1743     2848
## voicingvoiceless    -0.32      0.07    -0.46    -0.19 1.00     3901     5110
##
## Family Specific Parameters:
##     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## phi    24.69      0.96    22.86    26.61 1.00    12290     6516
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
gest.max.mag_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
```

```
  data = subdat,
  family = Beta(),
  prior = c(prior(normal(0, 10), class = Intercept),
            prior(cauchy(0, 5), class = sd),
            prior(gamma(0.01, 0.01), class = phi),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/gest_max_mag_null",
  save_all_pars = TRUE
)
```

```
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(gest.max.mag, gest.max.mag_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
## Iteration: 11
## Iteration: 12
```
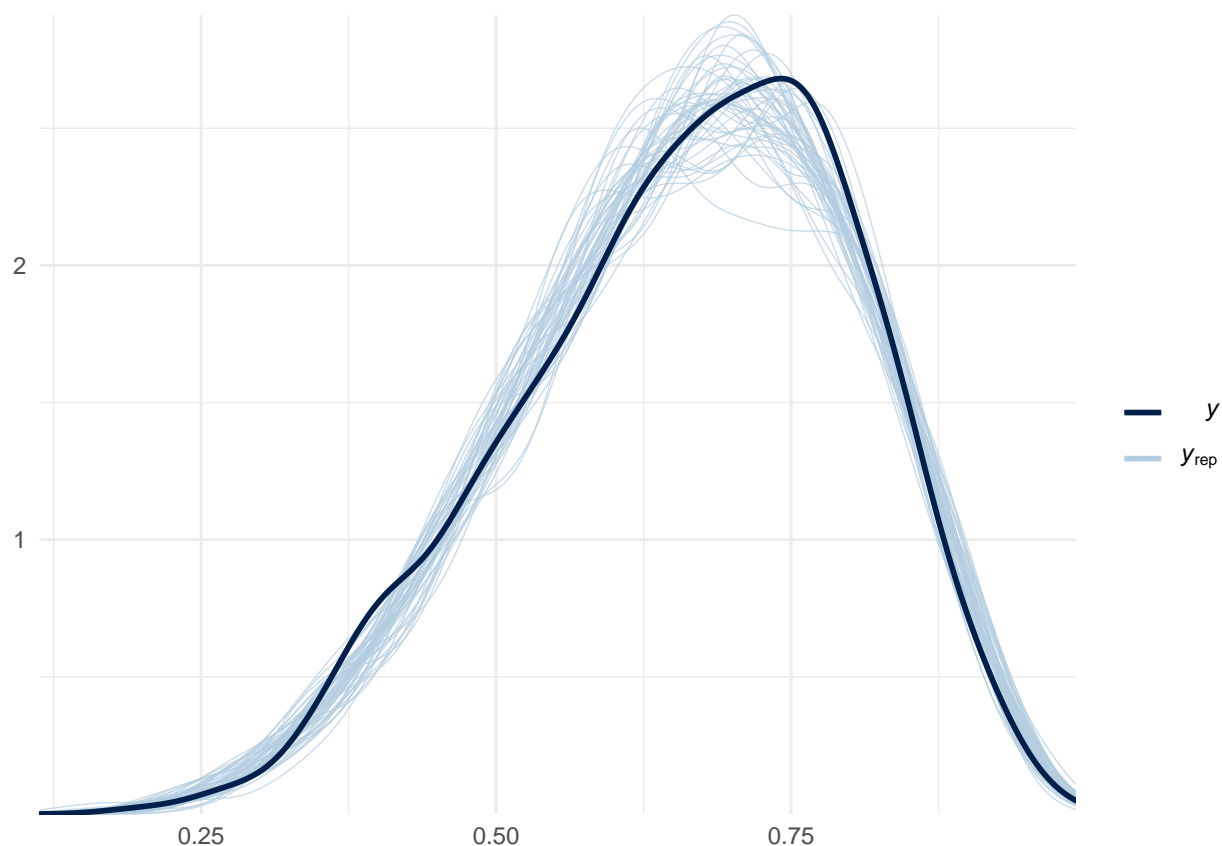
```
## Estimated Bayes factor in favor of gest.max.mag over gest.max.mag_null: 94.25061
```

```
# calculate the marginal posteriors of the full model
gest.max.mag_post <- brms::posterior_samples(gest.max.mag, pars="b_") %>%
  dplyr::mutate(
    nd = plogis(b_Intercept),
    nt = plogis(b_Intercept + b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
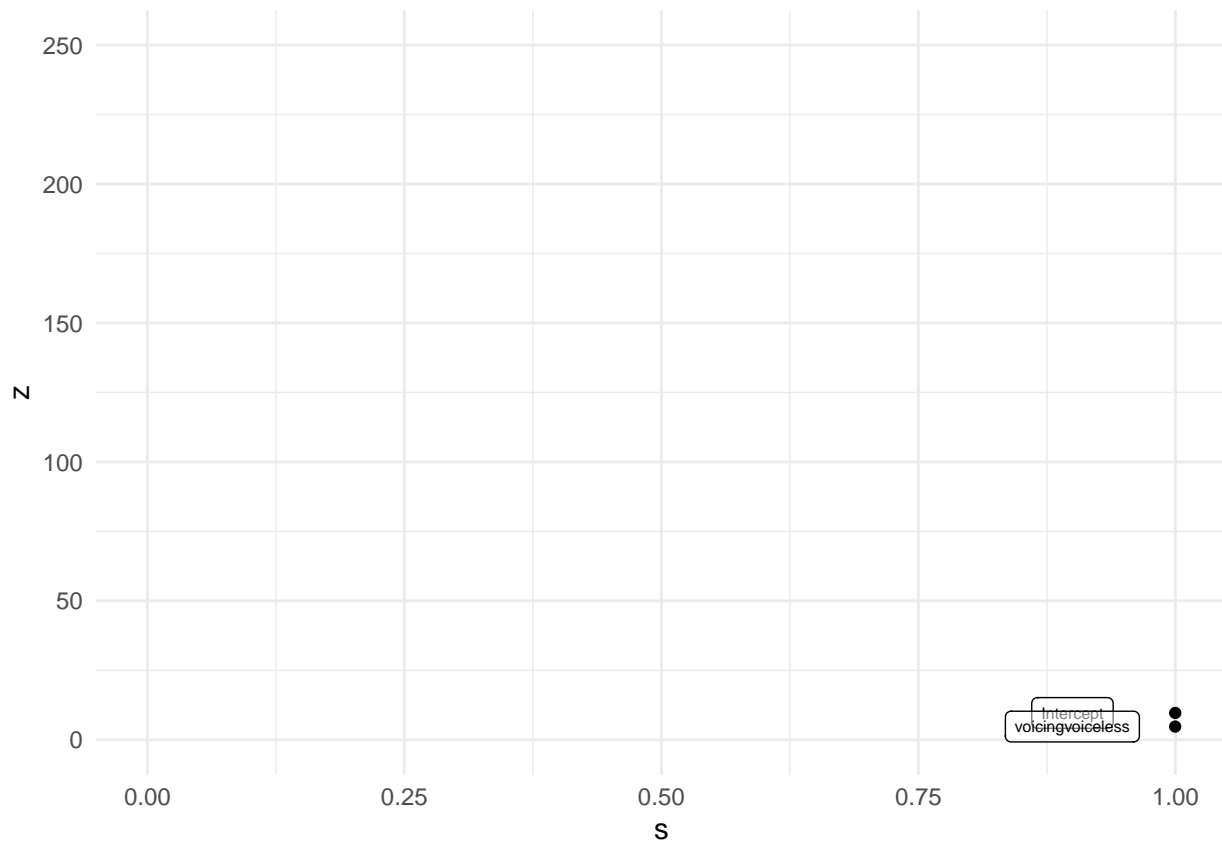
6.1 Checks

```r
pp_check(gest.max.mag, nsamples = 50) + theme_minimal()
```



```r
gest.max.mag_fixed <- fixef(gest.max.mag) %>% as_tibble(rownames = "term")

gest.max.mag_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(10, 5),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

7 Gesture offset stiffness

```r
# create the dependent variable
subdat$DV <- subdat$stiff.ons

# full model
stiff.ons <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 1), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/stiff_ons",
```

```
  save_all_pars = TRUE
)
```

```
stiff.ons
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##     Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.14      0.02     0.10     0.19 1.00
## sd(voicingvoiceless)               0.08      0.02     0.03     0.12 1.00
## cor(Intercept,voicingvoiceless)    0.23      0.28    -0.29     0.78 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      3426     5207
## sd(voicingvoiceless)               1563     2123
## cor(Intercept,voicingvoiceless)    2706     2951
##
## ~word (Number of levels: 30)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.09      0.02     0.06     0.13 1.00
## sd(voicingvoiceless)               0.02      0.02     0.00     0.06 1.00
## cor(Intercept,voicingvoiceless)    0.06      0.43    -0.78     0.82 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      2506     3897
## sd(voicingvoiceless)               1814     2244
## cor(Intercept,voicingvoiceless)    8293     5221
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           2.60      0.03     2.53     2.66 1.00     2241     3866
## voicingvoiceless    0.10      0.02     0.05     0.15 1.00     5914     6228
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.25      0.01     0.24     0.26 1.00     8951     6078
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
stiff.ons_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(cauchy(0, 0.1), class = sd),
```

```
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/stiff_ons_null",
  save_all_pars = TRUE
)
```

```
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(stiff.ons, stiff.ons_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
```
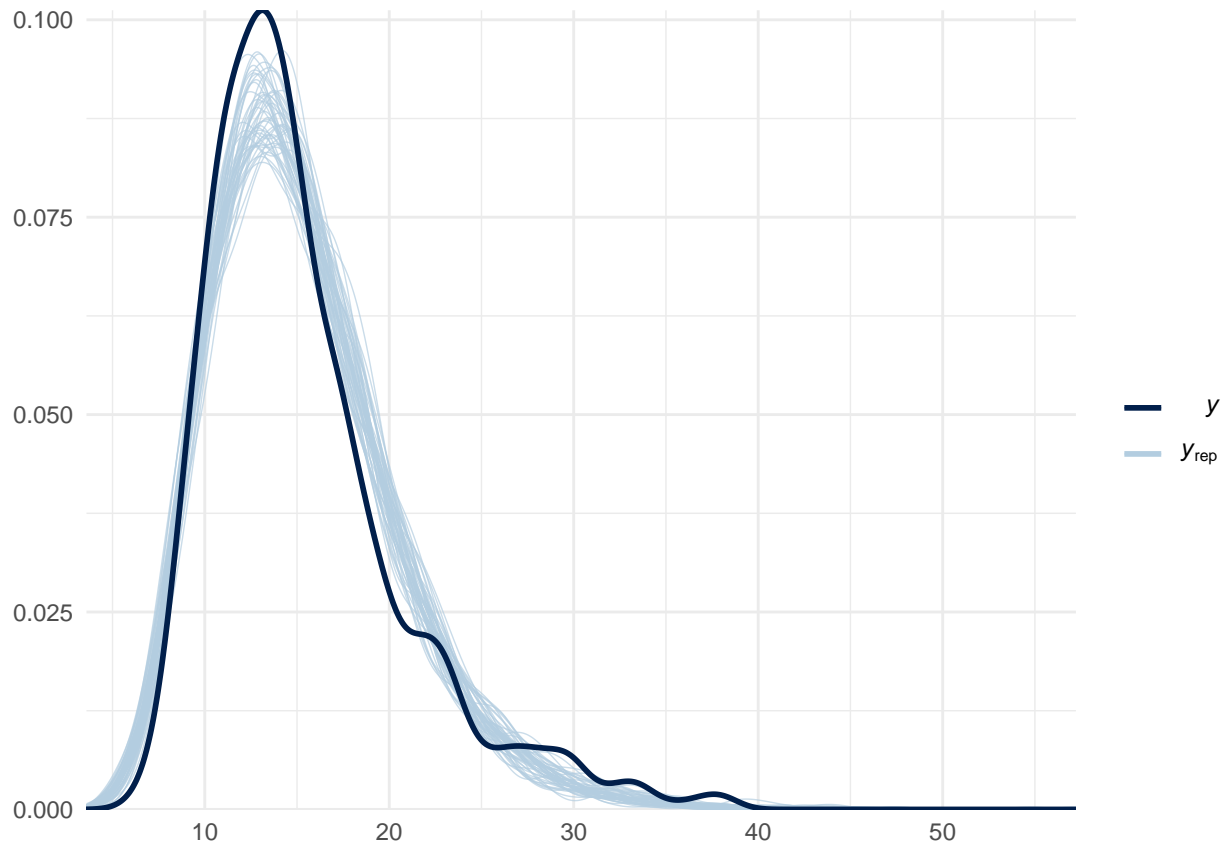
```
## Estimated Bayes factor in favor of stiff.ons over stiff.ons_null: 42.67598
```

```
# calculate the marginal posteriors of the full model
stiff.ons_post <- brms::posterior_samples(stiff.ons, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept + b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
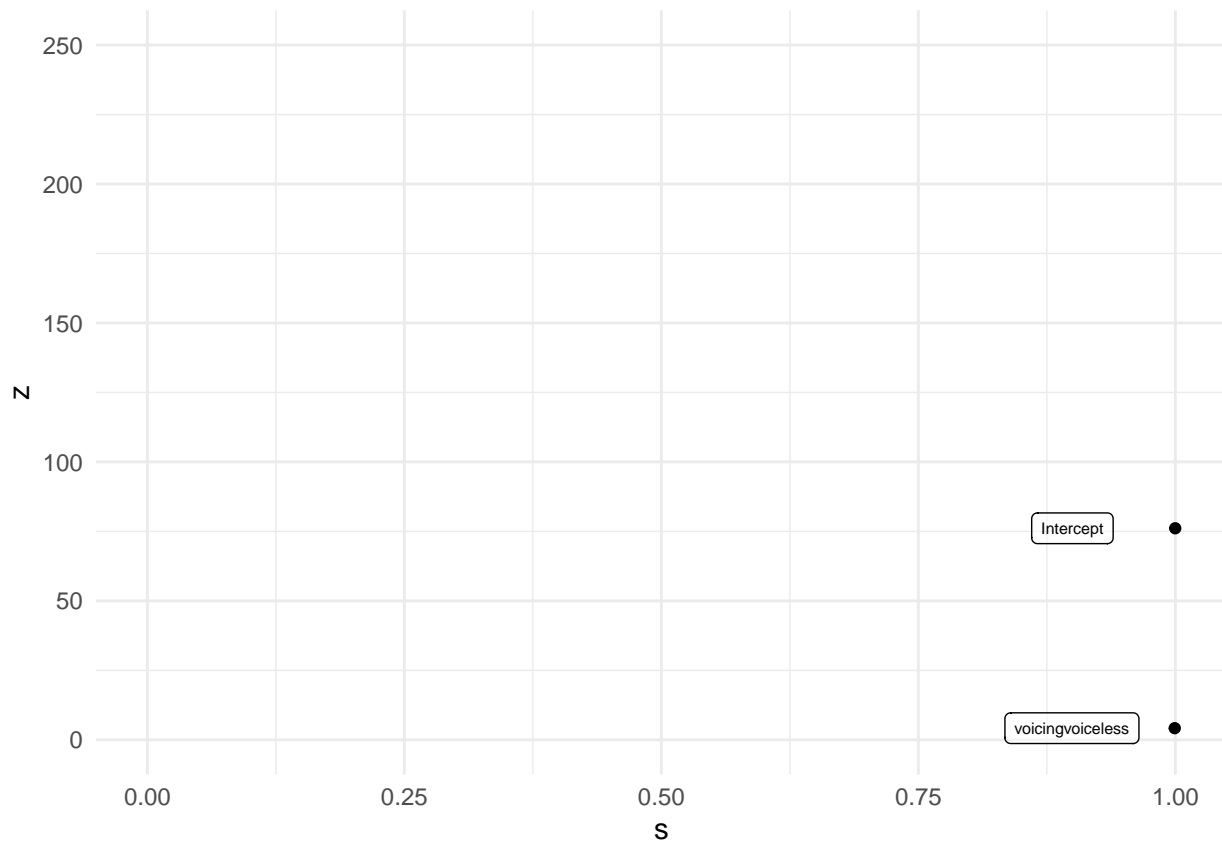
7.1 Checks

```
pp_check(stiff.ons, nsamples = 50) + theme_minimal()
```

```r
stiff.ons_fixed <- fixef(stiff.ons) %>% as_tibble(rownames = "term")

stiff.ons_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

8 Gesture offset stiffness

```r
# create the dependent variable
subdat$DV <- subdat$stiff.off

# full model
stiff.off <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 1), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 0.1), class = sd),
            prior(cauchy(0, 0.1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/stiff_off",
```

```
  save_all_pars = TRUE
)
```

```
stiff.off
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##         total post-warmup samples = 8000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.13      0.02     0.10     0.18 1.00
## sd(voicingvoiceless)              0.11      0.02     0.07     0.16 1.00
## cor(Intercept,voicingvoiceless)  -0.42      0.18    -0.72    -0.02 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     2791     4539
## sd(voicingvoiceless)              1883     3827
## cor(Intercept,voicingvoiceless)   2403     4139
##
## ~word (Number of levels: 30)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.05      0.02     0.02     0.08 1.00
## sd(voicingvoiceless)              0.04      0.02     0.00     0.08 1.00
## cor(Intercept,voicingvoiceless)  -0.15      0.40    -0.79     0.70 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     2676     2385
## sd(voicingvoiceless)              1003     2259
## cor(Intercept,voicingvoiceless)   3141     4584
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           2.64      0.03     2.59     2.70 1.00     2501     4241
## voicingvoiceless    0.14      0.03     0.09     0.20 1.00     2854     4454
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.24      0.00     0.23     0.25 1.00     9522     6161
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
stiff.off_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(cauchy(0, 0.1), class = sd),
```

```r
          prior(cauchy(0, 0.1), class = sigma),
          prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/stiff_off_null",
  save_all_pars = TRUE
)
```

```r
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(stiff.off, stiff.off_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
```
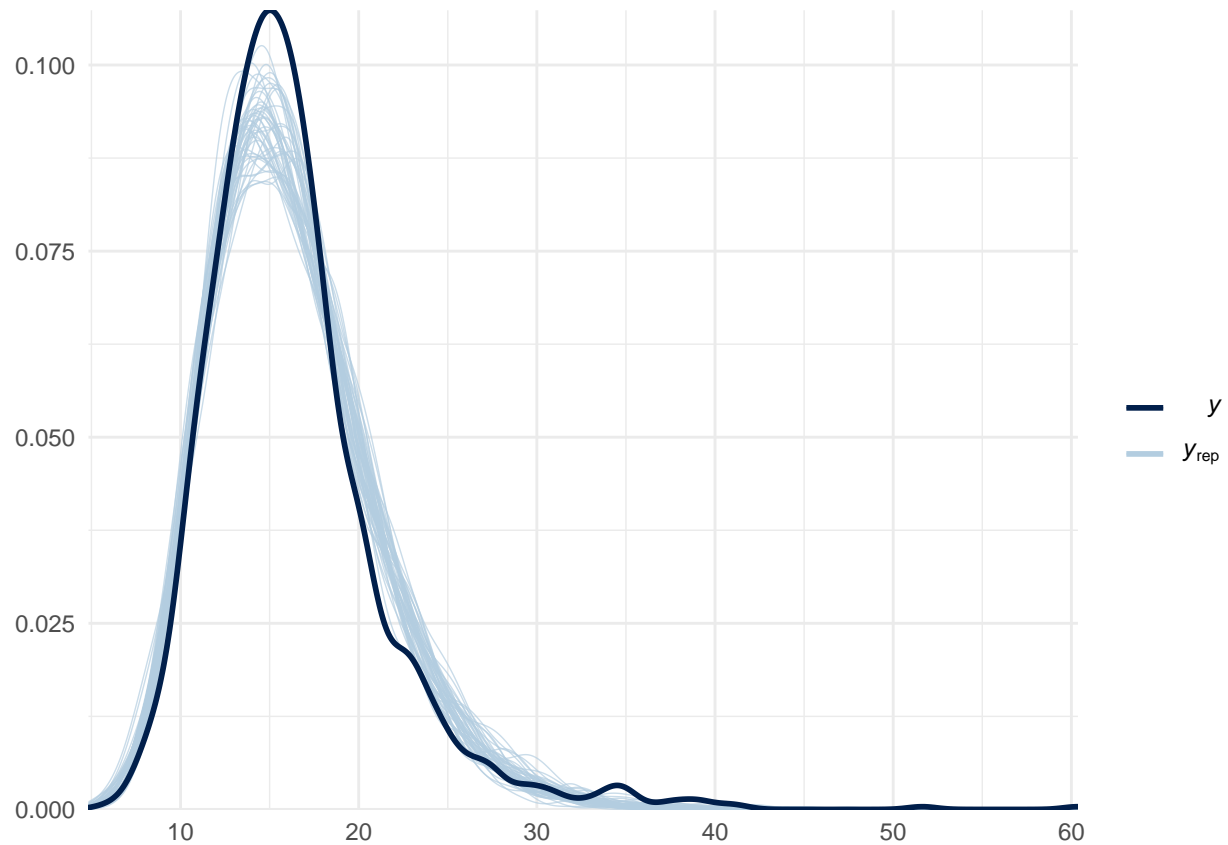
```
## Estimated Bayes factor in favor of stiff.off over stiff.off_null: 1851.88558
```

```r
# calculate the marginal posteriors of the full model
stiff.off_post <- brms::posterior_samples(stiff.off, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept) * exp(b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
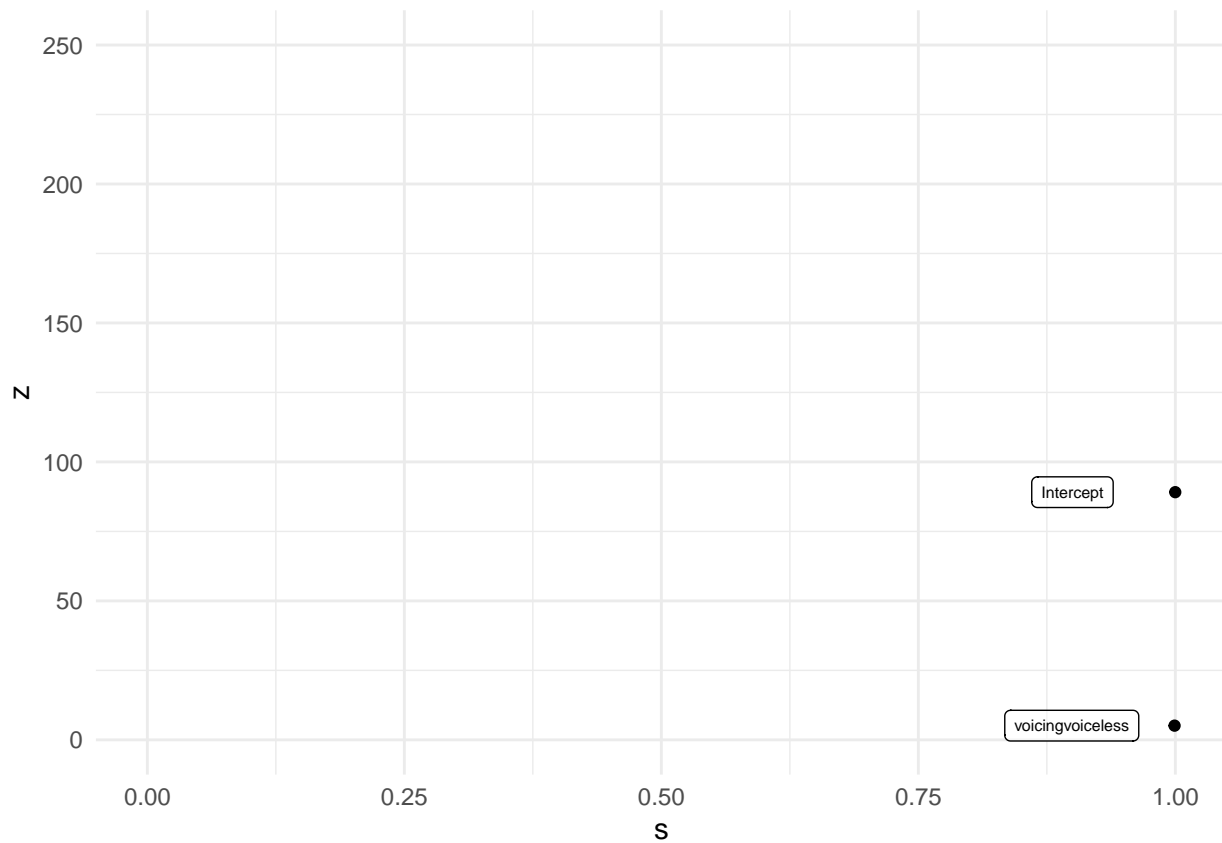
8.1 Checks

```r
pp_check(stiff.off, nsamples = 50) + theme_minimal()
```

```r
stiff.off_fixed <- fixef(stiff.off) %>% as_tibble(rownames = "term")

stiff.off_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

9 Kurtosis

```r
# create the dependent variable
subdat$DV <- subdat$kurtosis

# full model
kurt <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 3), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 1), class = sd),
            prior(cauchy(0, 1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/kurtosis",
```

```
  save_all_pars = TRUE
)
```

```
kurt
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.07      0.01     0.04     0.09 1.00
## sd(voicingvoiceless)              0.02      0.02     0.00     0.06 1.00
## cor(Intercept,voicingvoiceless)  -0.11      0.42    -0.81     0.74 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     3293     4759
## sd(voicingvoiceless)              1573     3551
## cor(Intercept,voicingvoiceless)   6016     5689
##
## ~word (Number of levels: 30)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.01      0.01     0.00     0.03 1.00
## sd(voicingvoiceless)              0.02      0.01     0.00     0.04 1.00
## cor(Intercept,voicingvoiceless)  -0.10      0.45    -0.86     0.77 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     4187     3691
## sd(voicingvoiceless)              2245     3083
## cor(Intercept,voicingvoiceless)   4650     5020
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           1.09      0.02     1.06     1.13 1.00     4024     5737
## voicingvoiceless    0.02      0.01    -0.01     0.04 1.00     9237     5809
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.21      0.00     0.20     0.22 1.00    11547     5630
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
kurt_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(cauchy(0, 1), class = sd),
```

```
          prior(cauchy(0, 1), class = sigma),
          prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/kurtosis_null",
  save_all_pars = TRUE
)
```

```
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(kurt, kurt_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
```
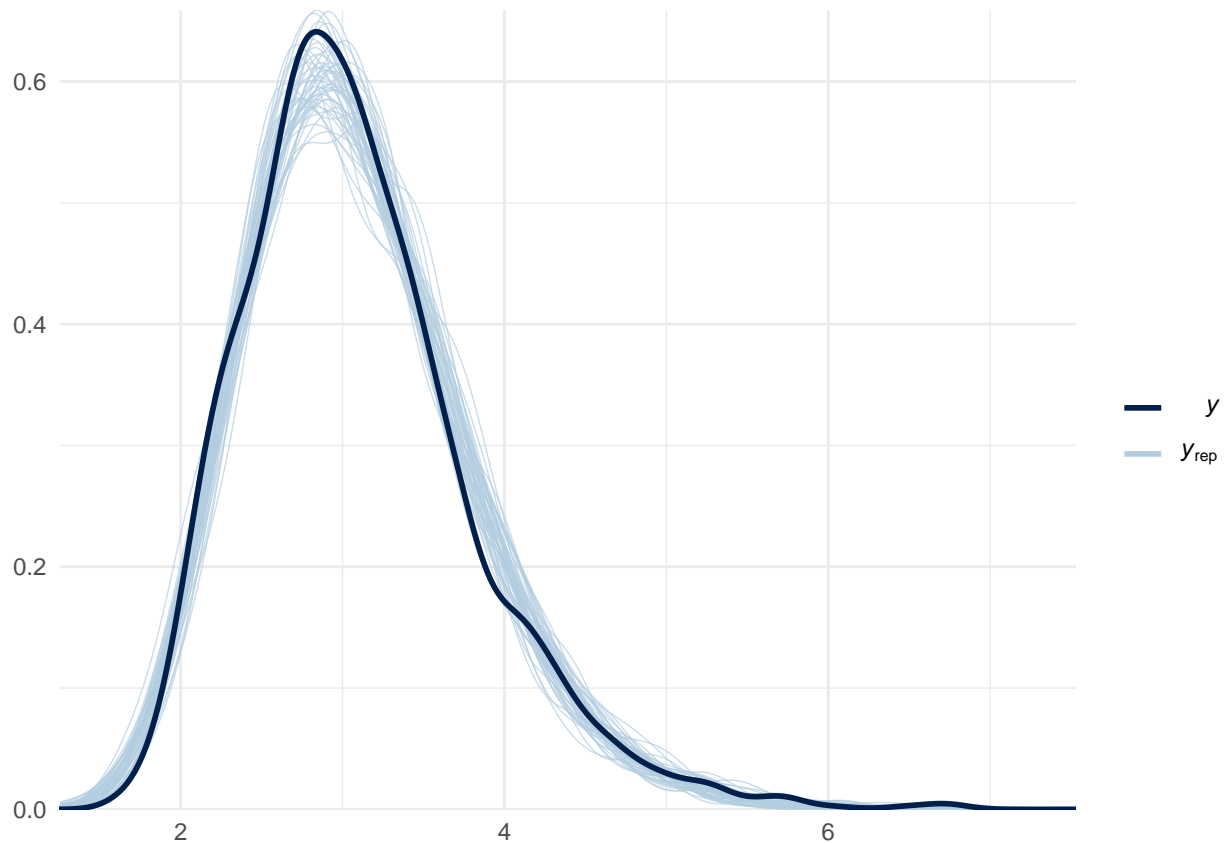
```
## Estimated Bayes factor in favor of kurt over kurt_null: 0.00866
```

```
# calculate the marginal posteriors of the full model
kurt_post <- brms::posterior_samples(kurt, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept + b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
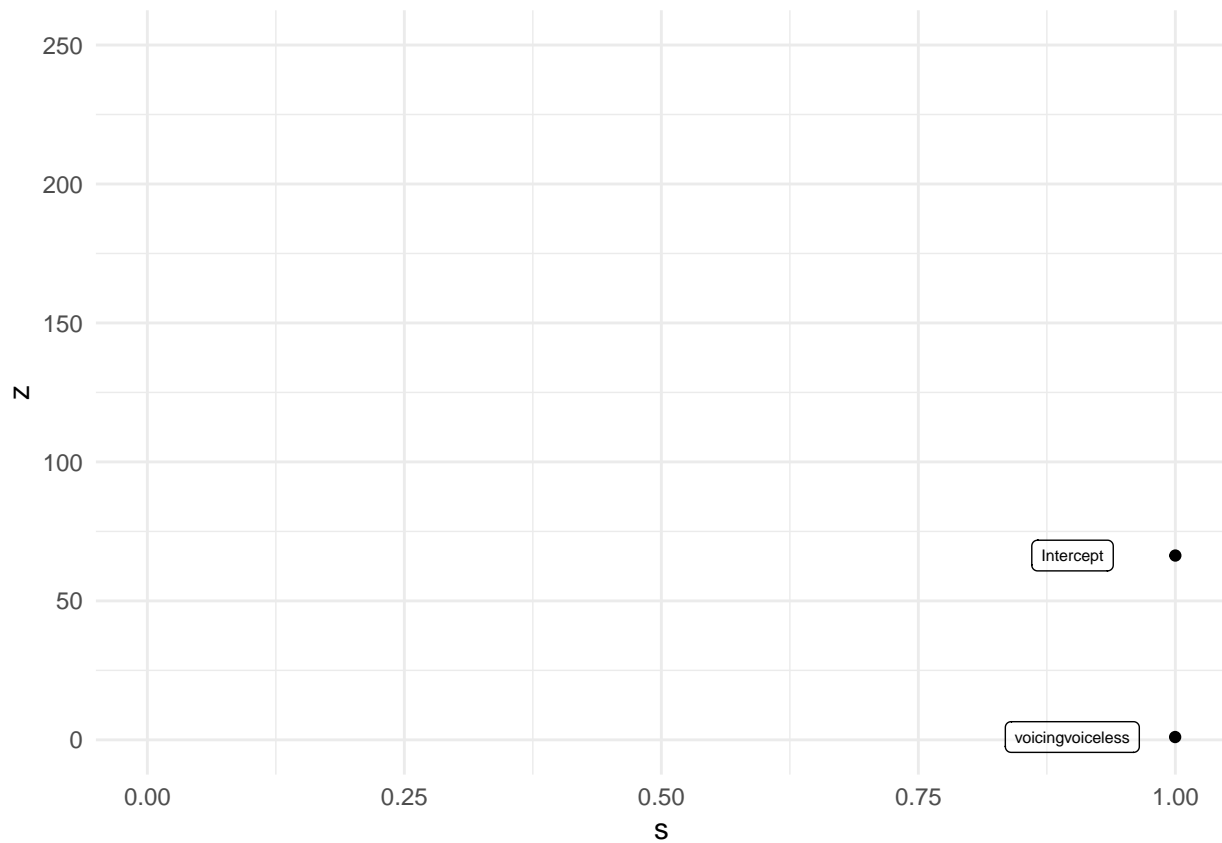
9.1 Checks

```
pp_check(kurt, nsamples = 50) + theme_minimal()
```

```r
kurt_fixed <- fixef(kurt) %>% as_tibble(rownames = "term")

kurt_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

10 Crest factor

```r
# create the dependent variable
subdat$DV <- subdat$crest.fact

# full model
crest.fact <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(normal(0, 3), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 1), class = sd),
            prior(cauchy(0, 1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/crest_factor",
```

```
  save_all_pars = TRUE
)
```

```
crest.fact
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##         total post-warmup samples = 8000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.01      0.00     0.01     0.02 1.00
## sd(voicingvoiceless)              0.02      0.01     0.00     0.03 1.00
## cor(Intercept,voicingvoiceless)   0.19      0.35    -0.46     0.83 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     3049     2851
## sd(voicingvoiceless)              1062     1068
## cor(Intercept,voicingvoiceless)   2071     2999
##
## ~word (Number of levels: 30)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                     0.01      0.00     0.00     0.01 1.00
## sd(voicingvoiceless)              0.01      0.00     0.00     0.02 1.00
## cor(Intercept,voicingvoiceless)  -0.14      0.45    -0.86     0.74 1.00
##                               Bulk_ESS Tail_ESS
## sd(Intercept)                     2098     2992
## sd(voicingvoiceless)              1135     1868
## cor(Intercept,voicingvoiceless)   2218     3504
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           0.61      0.00     0.60     0.62 1.00     5853     6461
## voicingvoiceless    0.01      0.01    -0.00     0.02 1.00     5979     6372
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.06      0.00     0.06     0.07 1.00    10135     6396
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
crest.fact_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 3), class = Intercept),
            prior(cauchy(0, 1), class = sd),
```

```
          prior(cauchy(0, 1), class = sigma),
          prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 4000,
  warmup = 2000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/crest_factor_null",
  save_all_pars = TRUE
)
```

```
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(crest.fact, crest.fact_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
## Iteration: 9
## Iteration: 10
## Iteration: 11
```
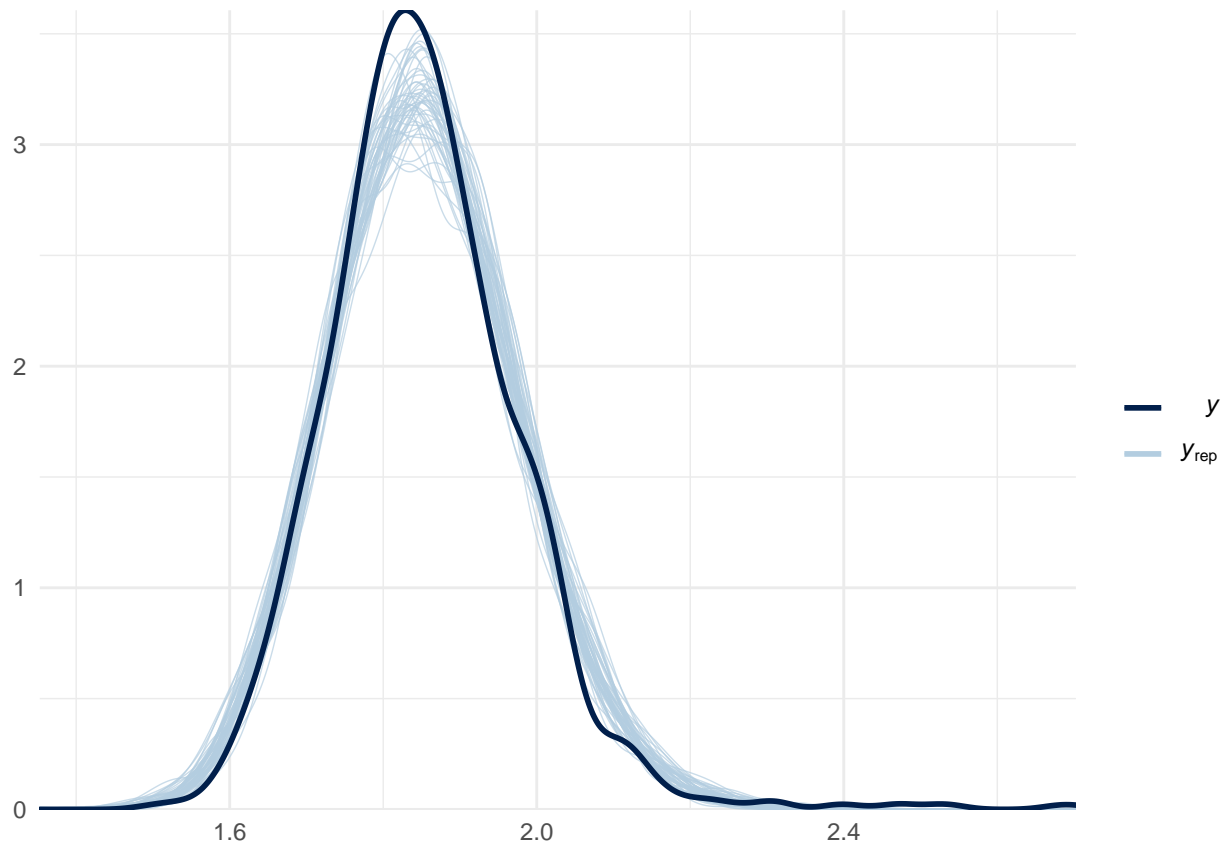
```
## Estimated Bayes factor in favor of crest.fact over crest.fact_null: 0.00593
```

```
# calculate the marginal posteriors of the full model
crest.fact_post <- brms::posterior_samples(crest.fact, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept + b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
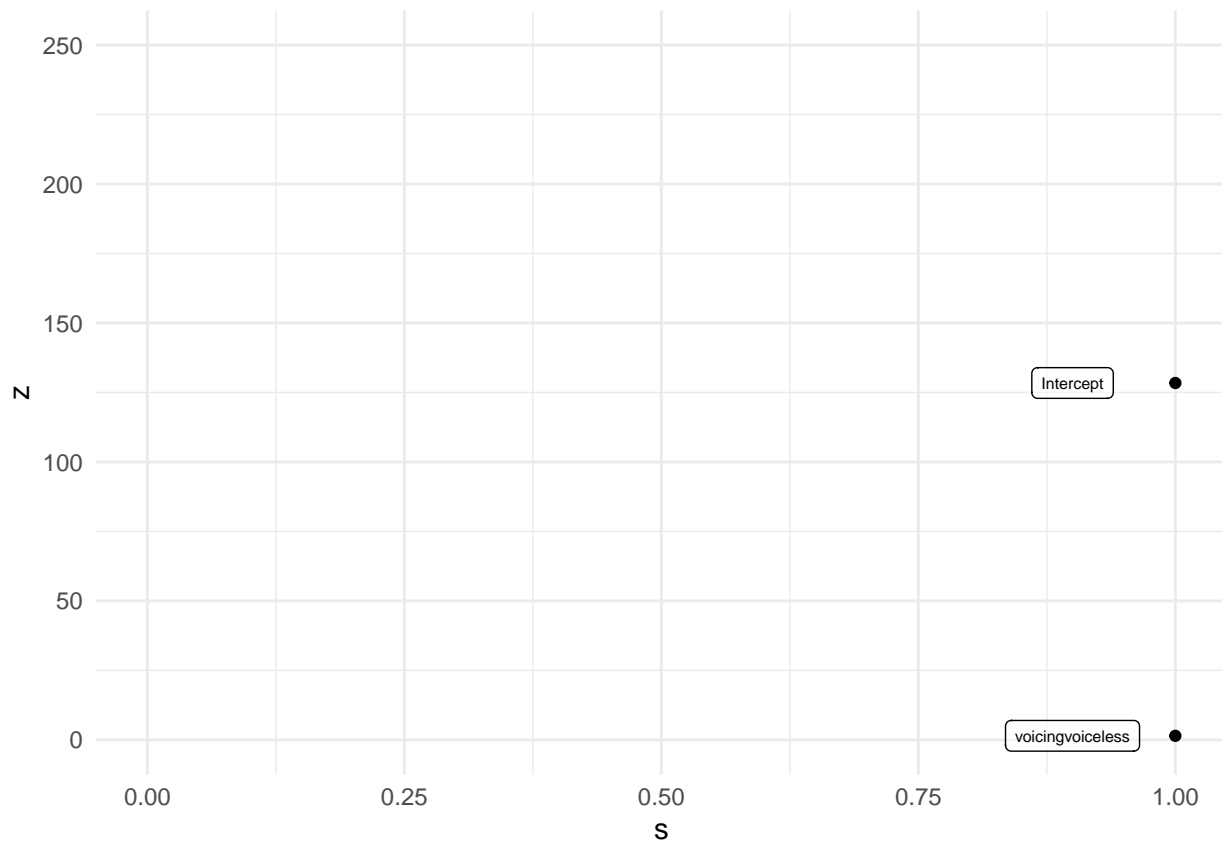
10.1 Checks

```
pp_check(crest.fact, nsamples = 50) + theme_minimal()
```

```r
crest.fact_fixed <- fixef(crest.fact) %>% as_tibble(rownames = "term")

crest.fact_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

11 Integral of velum movement in vowel (area under the curve)

```r
# create the dependent variable
subdat$DV <- subdat$vowel.integ

# full model
integ <- brms::brm(
  DV ~ voicing +
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 5), class = Intercept),
            prior(normal(0, 5), class = b, coef = voicingvoiceless),
            prior(cauchy(0, 1), class = sd),
            prior(cauchy(0, 1), class = sigma),
            prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/vowel_integ",
```

```
    save_all_pars = TRUE
)

integ
```

```
##  Family: lognormal
##   Links: mu = identity; sigma = identity
## Formula: DV ~ voicing + (1 + voicing | speaker) + (1 + voicing | word)
##    Data: subdat (Number of observations: 1365)
## Samples: 4 chains, each with iter = 6000; warmup = 3000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~speaker (Number of levels: 35)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.24      0.03     0.19     0.32 1.00
## sd(voicingvoiceless)               0.04      0.03     0.00     0.10 1.00
## cor(Intercept,voicingvoiceless)   -0.19      0.36    -0.78     0.60 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      2339     5156
## sd(voicingvoiceless)               1658     4637
## cor(Intercept,voicingvoiceless)   12192     6739
##
## ~word (Number of levels: 30)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      0.39      0.06     0.30     0.52 1.00
## sd(voicingvoiceless)               0.03      0.02     0.00     0.09 1.00
## cor(Intercept,voicingvoiceless)    0.09      0.44    -0.75     0.84 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      2527     4516
## sd(voicingvoiceless)               4188     5185
## cor(Intercept,voicingvoiceless)   16365     8396
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           3.73      0.09     3.56     3.89 1.00     1249     2419
## voicingvoiceless    0.09      0.03     0.03     0.14 1.00    12328     8717
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.29      0.01     0.28     0.30 1.00    20011     7987
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# reduced/null model
integ_null <- brms::brm(
  DV ~
    (1 + voicing | speaker) +
    (1 + voicing | word),
  data = subdat,
  family = lognormal(),
  prior = c(prior(normal(0, 5), class = Intercept),
            prior(cauchy(0, 1), class = sd),
```

```
          prior(cauchy(0, 1), class = sigma),
          prior(lkj(2), class = cor)),
  seed = my.seed,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = core.num,
  control = list(adapt_delta = 0.99,
                 max_treedepth = 20),
  file = "./rtMRI-velum/models/vowel_integ_null",
  save_all_pars = TRUE
)
```

```
# calculate the Bayes factor of the difference between the full and null models
brms::bayes_factor(integ, integ_null)
```

```
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 1
## Iteration: 2
## Iteration: 3
## Iteration: 4
## Iteration: 5
## Iteration: 6
## Iteration: 7
## Iteration: 8
```
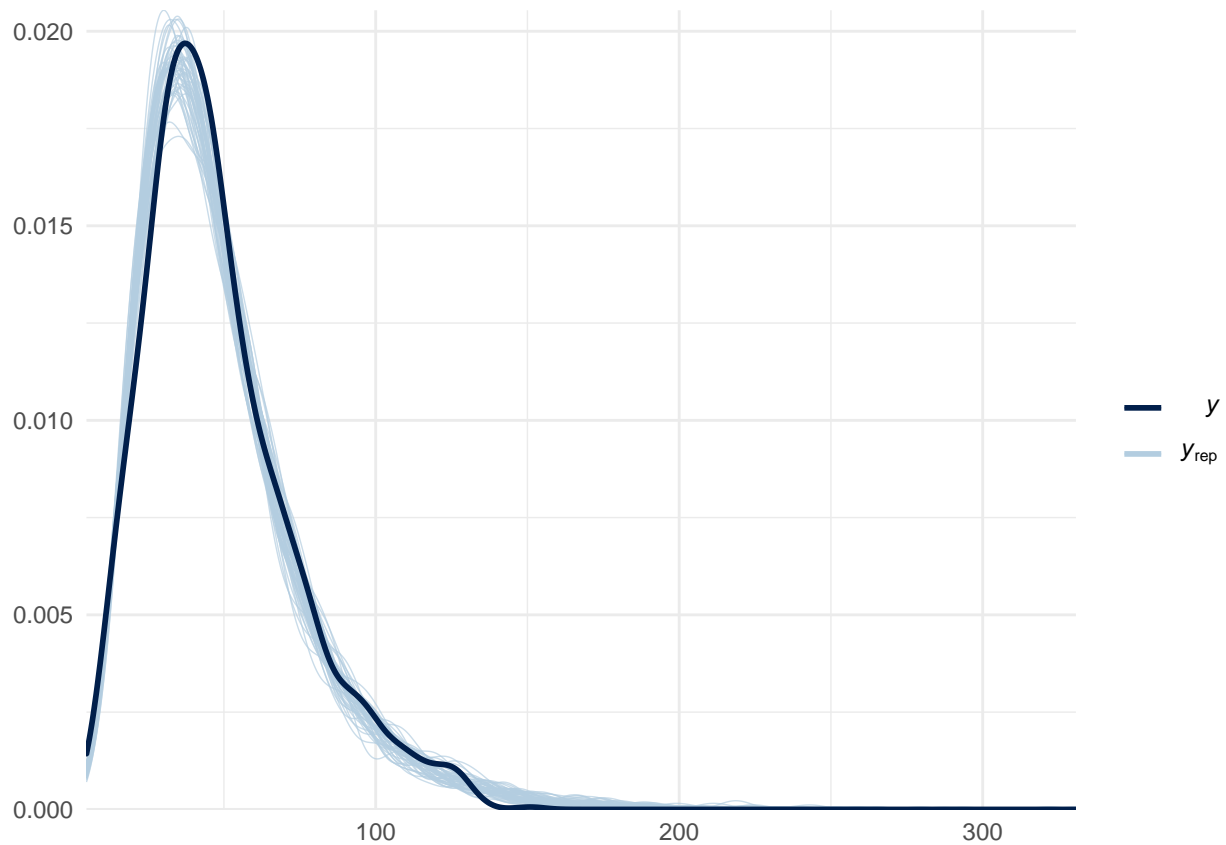
```
## Estimated Bayes factor in favor of integ over integ_null: 0.60027
```

```
# calculate the marginal posteriors of the full model
integ_post <- brms::posterior_samples(integ, pars="b_") %>%
  dplyr::mutate(
    nd = exp(b_Intercept),
    nt = exp(b_Intercept + b_voicingvoiceless)
  ) %>%
  dplyr::select(nd, nt) %>%
  tidyr::gather(context, DV)
```
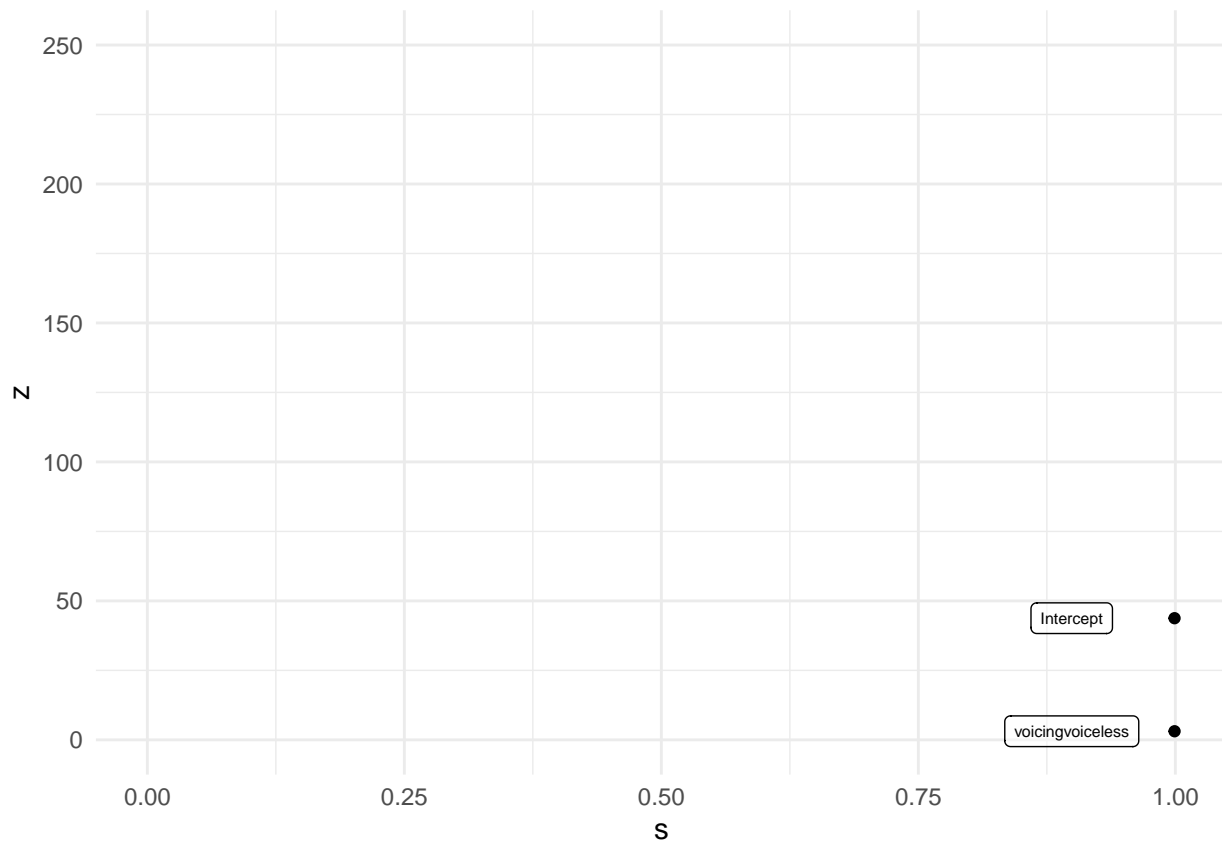
11.1 Checks

```
pp_check(integ, nsamples = 50) + theme_minimal()
```

```r
integ_fixed <- fixef(integ) %>% as_tibble(rownames = "term")

integ_fixed %>%
  mutate(
    theta = c(0, 0),
    sigma_prior = c(3, 1),
    # it's called here std.error but is the standard deviation
    z = abs((Estimate - theta) / Est.Error),
    s = 1 - (Est.Error^2 / sigma_prior^2)
  ) %>%
  ggplot(aes(s, z, label = term)) +
  geom_point() +
  geom_label(nudge_x = -0.1, size = 2, alpha = 0.5) +
  xlim(0, 1) + ylim(0, 250) +
  theme_minimal()
```

12 Plotting posteriors

```r
# colors to be used for plotting (suitable for B/W printing)
my.cols <- c("#2c7fb8","#7fcdbb")
```

12.1 Figure 5: duration, onset, peak (timing), offset

```r
# gather posteriors
fig5_post <- bind_rows(
  dur_post %>%
    rename(`duration` = DV)
  %>% pivot_longer(`duration`, names_to = "outcome", values_to = "estimate"),
  onset_post %>%
    rename(`onset` = DV) %>%
    pivot_longer(`onset`, names_to = "outcome", values_to = "estimate"),
  gest.max_post %>%
    rename(`peak` = DV) %>%
    pivot_longer(`peak`, names_to = "outcome", values_to = "estimate"),
  offset_post %>%
    rename(`offset` = DV) %>%
    pivot_longer(`offset`, names_to = "outcome", values_to = "estimate")
) %>%
  mutate(outcome = factor(outcome, levels = c("offset","peak","onset","duration")))
```

```
# flip sign of velum gesture onset posteriors for interpretability
fig5_post$estimate[fig5_post$outcome=="onset"] <- -fig5_post$estimate[fig5_post$outcome=="onset"]

# make separate plots
p1 <- fig5_post %>%
  filter(outcome == "duration") %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  scale_x_continuous(breaks=seq(0,500,5)) +
  coord_cartesian(ylim = c(1.3, 1.5), xlim = c(250,327)) +
  scale_fill_manual(values = my.cols) +
  labs(x = "Duration (ms)", y = element_blank()) + theme_bw() +
  theme(legend.position = "none",
        axis.text=element_text(size=12),
        axis.title=element_text(size=13))
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```

```
p2 <- fig5_post %>%
  filter(outcome %in% c("onset","peak","offset")) %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  geom_vline(xintercept = 0, lty=2) +
  scale_x_continuous(breaks=seq(-200,300,20)) +
  coord_cartesian(xlim = c(-125,205)) +
  scale_fill_manual(values=my.cols) +
  labs(x = "Time (ms) relative to acoustic vowel offset",
       y = element_blank(), fill = "Context") + theme_bw() +
  theme(axis.text=element_text(size=12),axis.title=element_text(size=13),
        panel.grid.major.y=element_blank())
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```

```
# save coposite plot
pdf(file="./rtMRI-velum/plots/time_plots.pdf",width=9.5,height=5,
    onefile=T,pointsize=14)
(p1 + p2) + patchwork::plot_layout(ncol = 1, guides = "collect") +
  theme(legend.position = "right")
dev.off()
```

```
## pdf
##   2
```

12.2 Figure 6: peak (magnitude)

```
# gather posteriors
fig6_post <- bind_rows(
  gest.max.mag_post %>%
    rename(`peak    \nmagnitude` = DV) %>%
    pivot_longer(`peak     \nmagnitude`, names_to = "outcome",
```

43

```
                    values_to = "estimate")
)

# make plot
p1 <- fig6_post %>%
  filter(outcome == "peak     \nmagnitude") %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  scale_x_continuous(breaks=seq(0,1,0.025)) +
  coord_cartesian(xlim = c(0.58,0.775), ylim = c(1.4,1.4)) +
  scale_fill_manual(values=my.cols) +
  labs(x = "Velum opening magnitude (speaker-scaled)", y = element_blank(),
       fill = "Context") + theme_bw() +
  theme(axis.text=element_text(size=12),axis.title=element_text(size=13))
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```

```
# save plot
pdf(file="./rtMRI-velum/plots/magnitude_plot.pdf",width=9,height=3,onefile=T,
    pointsize=14)
p1
dev.off()
```

```
## pdf
##   2
```

12.3   Figure 7: opening stiffness, closing stiffness

```
# gather posteriors
fig7_post <- bind_rows(
  stiff.ons_post %>% rename(`opening\nstiffness` = DV) %>%
    pivot_longer(`opening\nstiffness`, names_to = "outcome", values_to = "estimate"),
  stiff.off_post %>% rename(`closing \nstiffness` = DV) %>%
    pivot_longer(`closing \nstiffness`, names_to = "outcome", values_to = "estimate")
)

# make composite plot on same scale
p1 <- fig7_post %>%
  filter(outcome %in% c("opening\nstiffness","closing \nstiffness")) %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  scale_x_continuous(breaks=seq(0,30,0.5)) +
  coord_cartesian(xlim = c(12,17.5), ylim = c(1.4,2.3)) +
  scale_fill_manual(values=my.cols) +
  labs(x = "Stiffness", y = element_blank(), fill = "Context") + theme_bw() +
  theme(axis.text=element_text(size=12),axis.title=element_text(size=13),
        panel.grid.major.y=element_blank())
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```

```r
# save plot
pdf(file="./rtMRI-velum/plots/stiffness_plots.pdf",width=9,height=4,onefile=T,
    pointsize=14)
p1
dev.off()
```

```
## pdf
##   2
```

12.4   Figure 9: kurtosis, crest factor

```r
# gather posteriors
fig9_post <- bind_rows(
  kurt_post %>% rename(`kurtosis` = DV) %>%
    pivot_longer(`kurtosis`, names_to = "outcome", values_to = "estimate"),
  crest.fact_post %>% rename(`crest \nfactor` = DV) %>%
    pivot_longer(`crest \nfactor`, names_to = "outcome", values_to = "estimate")
)

# make separate plots
p1 <- fig9_post %>%
  filter(outcome == "kurtosis") %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  scale_x_continuous(breaks=seq(0,5,0.05)) +
  coord_cartesian(xlim = c(2.81,3.18), ylim = c(1.4,1.4)) +
  scale_fill_manual(values=my.cols) +
  labs(x = "Kurtosis", y = element_blank(), fill = "Context") +
  theme_bw() +
  theme(legend.position = "none",axis.text=element_text(size=12),
        axis.title=element_text(size=13),panel.grid.major.y=element_blank())
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```

```r
p2 <- fig9_post %>%
  filter(outcome == "crest \nfactor") %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  scale_x_continuous(breaks=seq(0,3,0.01)) +
  coord_cartesian(xlim = c(1.805,1.885), ylim = c(1.4,1.4)) +
  scale_fill_manual(values=my.cols) +
  labs(x = "Crest factor (ratio of peak to average displacement)",
       y = element_blank(), fill = "Context") + theme_bw() +
  theme(legend.position = "none",axis.text=element_text(size=12),
        axis.title=element_text(size=13),panel.grid.major.y=element_blank())
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```

```r
# save composite plot
pdf(file="./rtMRI-velum/plots/peakedness_plots.pdf",width=9,height=5,
```

```
    onefile=T,pointsize=14)
(p1 + p2) + patchwork::plot_layout(ncol = 1, guides = "collect") +
  theme(legend.position = "right")
dev.off()
```

```
## pdf
##   2
```

12.5 Figure 10: velum displacement integral

```
# gather posteriors
fig10_post <- bind_rows(
  integ_post %>% rename(`gesture\nintegral` = DV) %>%
    pivot_longer(`gesture\nintegral`, names_to = "outcome",
                 values_to = "estimate")
)

# make plot
p1 <- fig10_post %>%
  filter(outcome == "gesture\nintegral") %>%
  ggplot(aes(estimate, outcome, group = context, fill = context)) +
  geom_halfeyeh(slab_color="black", slab_alpha=0.5) +
  scale_x_continuous(breaks=seq(0,100,2)) +
  coord_cartesian(xlim = c(31,64.5), ylim = c(1.4,1.4)) +
  scale_fill_manual(values=my.cols) +
  labs(x = "Velum displacement integral: time (ms) X magnitude (normalized)",
       y = element_blank(), fill = "Context") + theme_bw() +
  theme(axis.text=element_text(size=12),axis.title=element_text(size=13))
```

```
## Warning: 'geom_halfeyeh' is deprecated.
## Use 'stat_halfeye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```
```
# save plot
pdf(file="./rtMRI-velum/plots/integral_plot.pdf",width=9,height=3,onefile=T,
    pointsize=14)
p1
dev.off()
```

```
## pdf
##   2
```

References

Betancourt, Michael. 2018. "Calibrating Model-Based Inferences and Decisions." arXiv preprint arXiv:1803.08393.

Gahl, Susanne, and R. Harald Baayen. 2019. "Twenty-Eight Years of Vowels: Tracking Phonetic Variation Through Young to Middle Age Adulthood." *Journal of Phonetics* 74: 42–54.

Ratnikova, E. I. 2017. "Towards a Log-Normal Model of Phonation Units Lengths Distribution in the Oral Utterances." *International Research Journal* 3 (57): 46–49.

Rosen, Kristin M. 2005. "Analysis of Speech Segment Duration with the Lognormal Distribution: A Basis for Unification and Comparison." *Journal of Phonetics* 33 (4): 411–26.

Vasishth, Shravan, M. Beckman, B. Nicenboim, Fangfang Li, and Eun Jong Kong. 2018. "Bayesian Data Analysis in the Phonetic Sciences: A Tutorial Introduction." *Journal of Phonetics* 71: 147–61.