E3 rate (Bayes version)

R. Cassano-Coleman

2025-06-26

This notebook takes response rates by subject, log-transforms them, and runs a Bayesian version of a mixed effects model to replace the non-parametric ANOVA-type test that is currently reported in the paper.

```
set.seed(15000)
```

Load the data.

```
data <- read_csv("../data/response_rate_by_sub.csv")
```

```
## Rows: 380 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Musician, scramble
## dbl (3): exp_subject_id, stimulus_set, mean_response_rate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Change musician and scramble into a factor.

```
data %<>% mutate(Musician = factor(Musician, levels = c('Yes', 'No')),
                 scramble = factor(scramble, levels = c('Intact', '8B', '2B', '1B')))
```

Set Intact as reference level.

```
contrasts(data$scramble) <- contr.treatment(4)
```

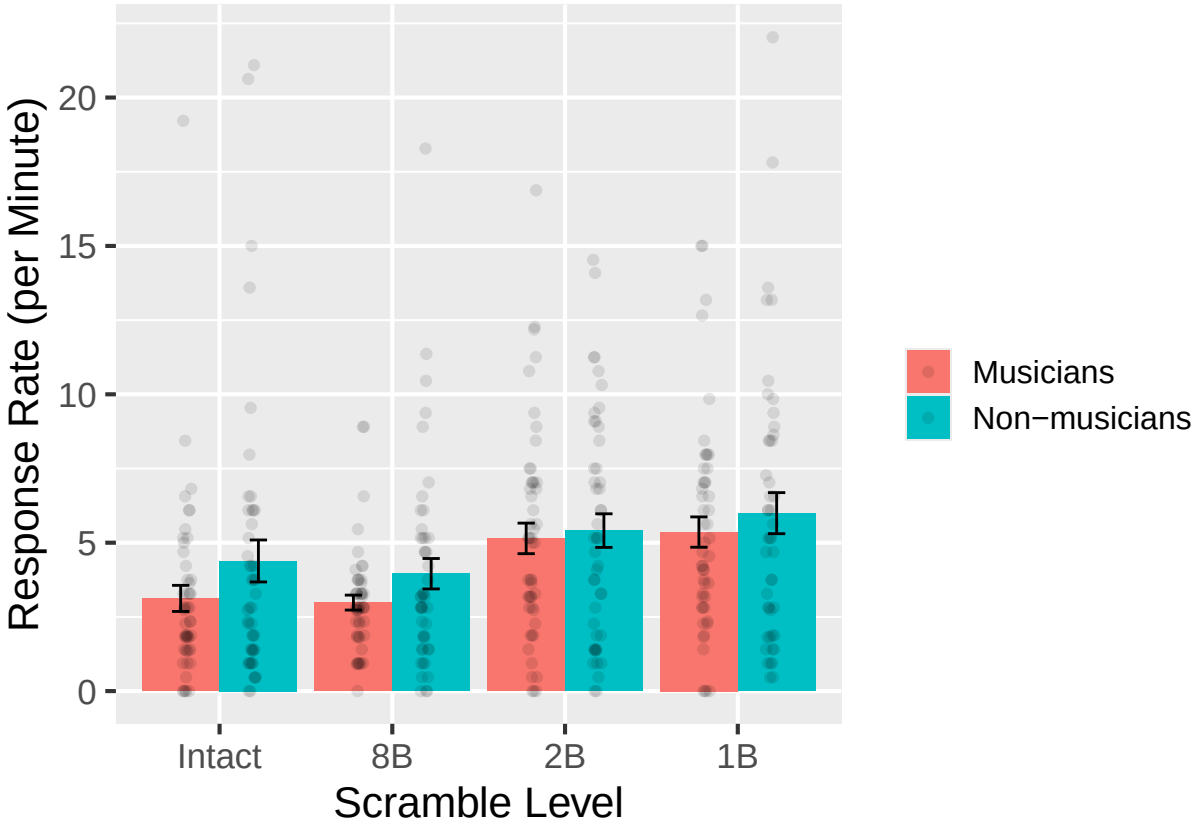
Check normality of the true response rate data. (Ignoring stimulus set.)

```
data %>%
  group_by(Musician, scramble) %>%
  shapiro_test(mean_response_rate)
```

```
## # A tibble: 8 x 5
##   Musician scramble variable           statistic            p
##   <fct>    <fct>    <chr>                  <dbl>        <dbl>
## 1 Yes      Intact   mean_response_rate     0.724 0.0000000283
## 2 Yes      8B       mean_response_rate     0.863 0.0000429
## 3 Yes      2B       mean_response_rate     0.941 0.0162
## 4 Yes      1B       mean_response_rate     0.929 0.00575
## 5 No       Intact   mean_response_rate     0.744 0.000000135
## 6 No       8B       mean_response_rate     0.839 0.0000157
## 7 No       2B       mean_response_rate     0.948 0.0401
## 8 No       1B       mean_response_rate     0.894 0.000530
```

```
data %>%
  ggplot(aes(x = scramble, y = mean_response_rate, fill = Musician)) +
  geom_bar(position = "dodge", stat = "summary", fun = mean) +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.2, stat = "summary") +
  geom_point(position = position_jitterdodge(jitter.width = 0.1), alpha = 0.1) +
  theme_gray(base_size = 16) +
  xlab('Scramble Level') +
  ylab('Response Rate (per Minute)') +
  scale_fill_discrete(name="", labels=c('Musicians', 'Non-musicians')) +
  theme(legend.text = element_text(size = 12))
```

```
## No summary function supplied, defaulting to `mean_se()`
```


Response Rate (per Minute)
20
15
10
5
0
Intact
8B
2B
1B
Scramble Level
Musicians
Non−musicians

Log-transform the rates and check for normality.

```
data %<>% mutate(log_rate = log(1 + mean_response_rate))
# add 1 so rates that are zero transform to 0 (rather than negative infinity)
```

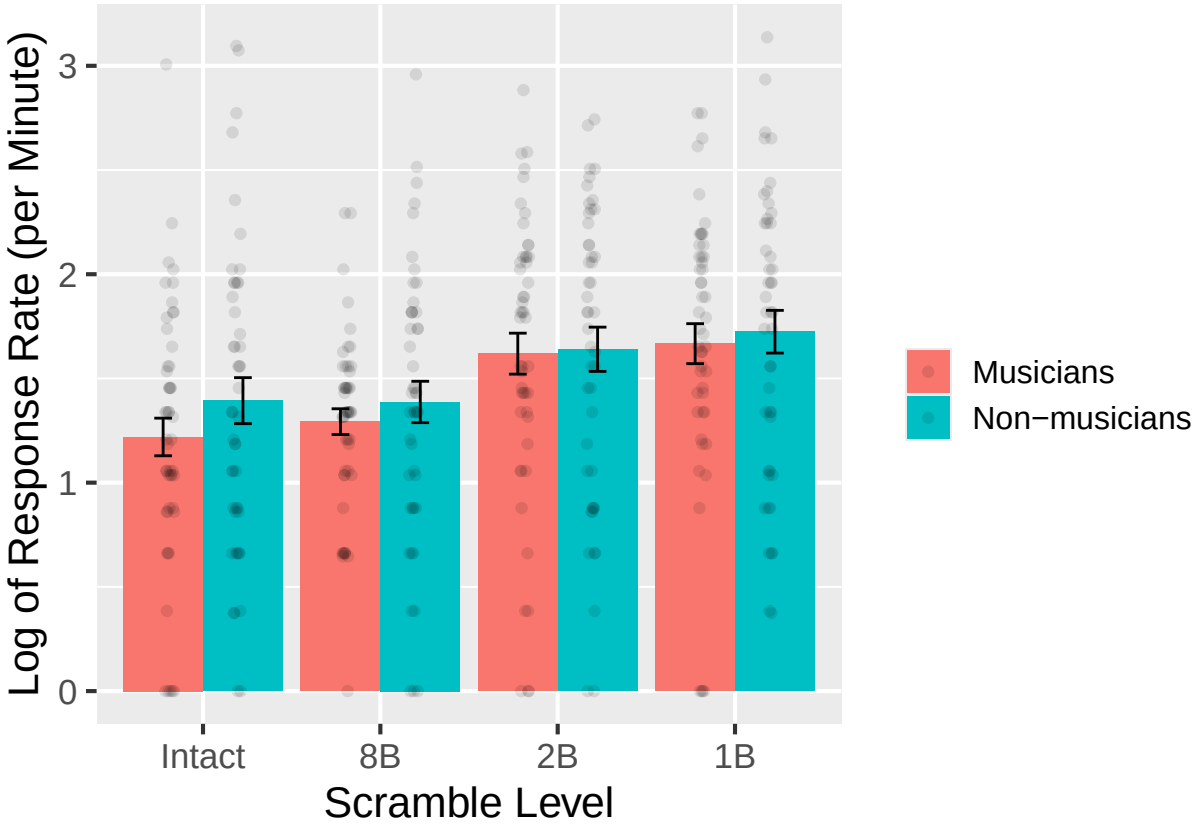
```
data %>%
  group_by(Musician, scramble) %>%
  shapiro_test(log_rate)
```

```
## # A tibble: 8 x 5
##   Musician scramble variable statistic       p
##   <fct>    <fct>    <chr>        <dbl>   <dbl>
## 1 Yes      Intact   log_rate     0.963 0.130
## 2 Yes      8B       log_rate     0.947 0.0284
## 3 Yes      2B       log_rate     0.943 0.0193
## 4 Yes      1B       log_rate     0.913 0.00153
## 5 No       Intact   log_rate     0.976 0.457
## 6 No       8B       log_rate     0.982 0.674
## 7 No       2B       log_rate     0.946 0.0319
## 8 No       1B       log_rate     0.973 0.358
```

Some of these are worse than others. Visualize:

```
data %>%
  ggplot(aes(x = scramble, y = log_rate, fill = Musician)) +
  geom_bar(position = "dodge", stat = "summary", fun = mean) +
  geom_errorbar(position = position_dodge(width = 0.9), width = 0.2, stat = "summary") +
  geom_point(position = position_jitterdodge(jitter.width = 0.1), alpha = 0.1) +
  theme_gray(base_size = 16) +
  xlab('Scramble Level') +
  ylab('Log of Response Rate (per Minute)') +
  scale_fill_discrete(name="", labels=c('Musicians', 'Non-musicians')) +
  theme(legend.text = element_text(size = 12))
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
#ggsave('rate.png', width = 7, height = 5)
```

It seems like this lack of normality is driven by the zero rates.

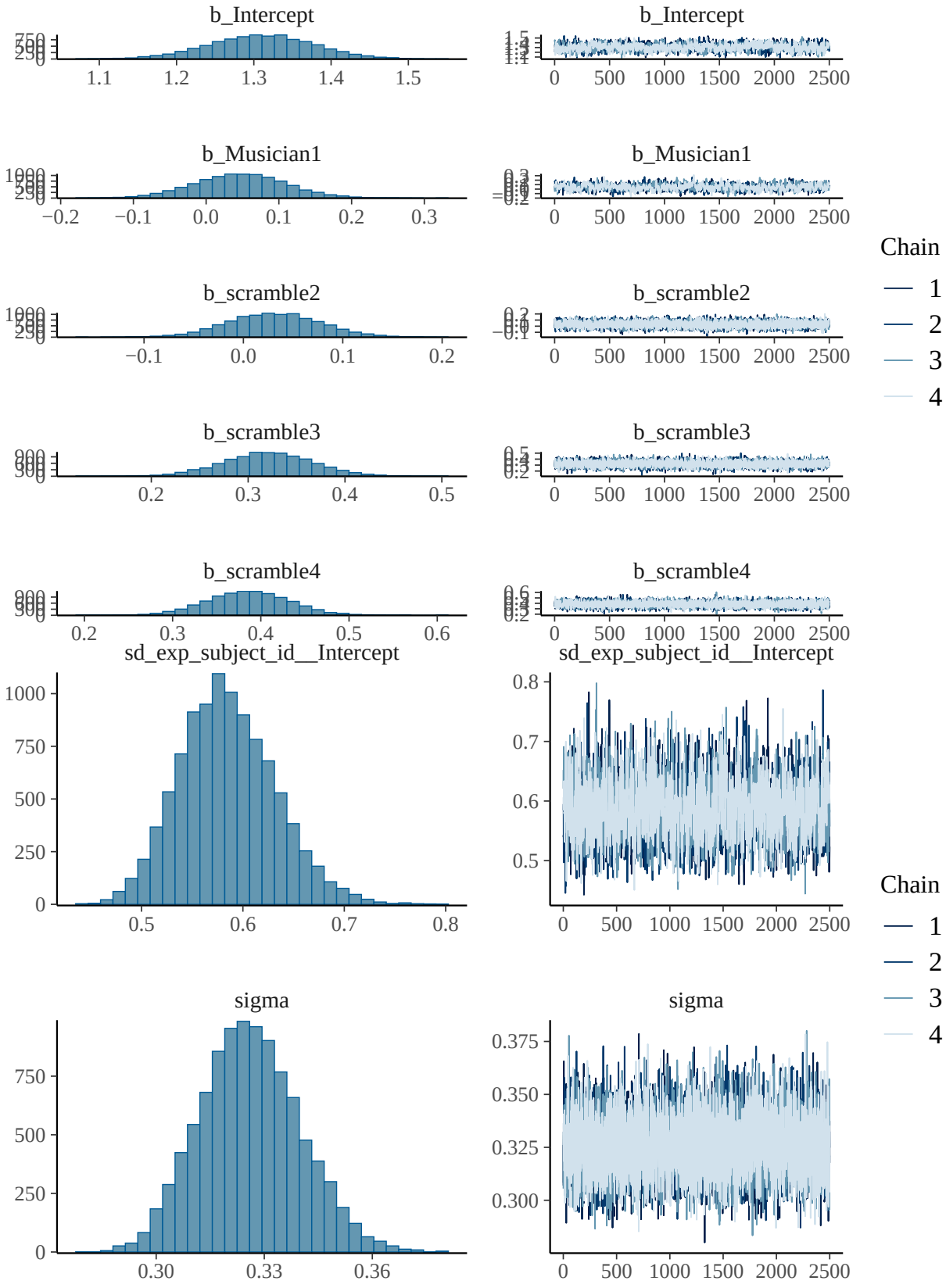
```
get_prior(log_rate ~ Musician + scramble + (1|exp_subject_id), data = data)
```

```
##                   prior     class       coef          group resp dpar nlpar lb
##                  (flat)         b                                              
##                  (flat)         b MusicianNo                                   
##                  (flat)         b  scramble2                                   
##                  (flat)         b  scramble3                                   
##                  (flat)         b  scramble4                                   
##  student_t(3, 1.5, 2.5) Intercept                                              
##    student_t(3, 0, 2.5)        sd                                             0
##    student_t(3, 0, 2.5)        sd            exp_subject_id                   0
##    student_t(3, 0, 2.5)        sd  Intercept exp_subject_id                   0
##    student_t(3, 0, 2.5)     sigma                                             0
##  ub       source
##          default
##     (vectorized)
##     (vectorized)
##     (vectorized)
##     (vectorized)
##          default
##          default
##     (vectorized)
##     (vectorized)
##          default
```

```
these_priors <- c(
  set_prior('normal(0, 0.5)', coef = "Musician1"), # don't necessarily expect a difference between grou
  set_prior('normal(0, 0.5)', coef = "scramble2"), # intact vs 8B
  set_prior('normal(0, 0.5)', coef = "scramble3"), # intact vs 2B
  set_prior('normal(0, 0.5)', coef = "scramble4") # intact vs 1B
)
```

```
brm_log_rate <- brm(log_rate ~ Musician + scramble + (1|exp_subject_id), data = data,
                    prior = these_priors,
                    save_pars = save_pars(all = TRUE), iter = 5000,
                    file = '../models/E3_log_rate')
```

```
plot(brm_log_rate)
```



```r
print(summary(brm_log_rate), digits = 4)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: log_rate ~ Musician + scramble + (1 | exp_subject_id)
##    Data: data (Number of observations: 380)
##   Draws: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup draws = 10000
##
## Multilevel Hyperparameters:
## ~exp_subject_id (Number of levels: 95)
##               Estimate Est.Error l-95% CI u-95% CI   Rhat Bulk_ESS Tail_ESS
## sd(Intercept)   0.5848    0.0471   0.4990   0.6833 1.0069     1669     3427
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI   Rhat Bulk_ESS Tail_ESS
## Intercept   1.3098    0.0682   1.1743   1.4424 1.0040     1206     2673
## Musician1   0.0469    0.0626  -0.0746   0.1704 1.0043      953     2011
## scramble2   0.0308    0.0464  -0.0596   0.1221 1.0001     8142     8185
## scramble3   0.3206    0.0465   0.2293   0.4115 1.0002     7677     7800
## scramble4   0.3858    0.0466   0.2949   0.4781 1.0006     7585     8083
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI   Rhat Bulk_ESS Tail_ESS
## sigma   0.3261    0.0139   0.3004   0.3543 1.0005     9571     8167
##
## Draws were sampled 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
emm_log_rate <- emmeans(brm_log_rate, specs = c("scramble", "Musician"))
summary(emm_log_rate)
```

```
##  scramble Musician emmean lower.HPD upper.HPD
##  Intact   Yes        1.26      1.07      1.43
##  8B       Yes        1.29      1.11      1.47
##  2B       Yes        1.58      1.39      1.76
##  1B       Yes        1.65      1.47      1.83
##  Intact   No         1.36      1.18      1.54
##  8B       No         1.39      1.21      1.57
##  2B       No         1.68      1.49      1.85
##  1B       No         1.74      1.56      1.92
##
## Point estimate displayed: median
## HPD interval probability: 0.95
```

```r
emm_log_rate_s <- emmeans(brm_log_rate, specs = "scramble")
summary(emm_log_rate_s)
```

```
##  scramble emmean lower.HPD upper.HPD
##  Intact     1.31      1.17      1.44
##  8B         1.34      1.21      1.47
##  2B         1.63      1.50      1.76
##  1B         1.70      1.56      1.82
##
## Results are averaged over the levels of: Musician
## Point estimate displayed: median
## HPD interval probability: 0.95
```

```r
contrast(emm_log_rate_s, method = "pairwise")
```

```
##  contrast    estimate lower.HPD upper.HPD
##  Intact - 8B  -0.0306    -0.124    0.0574
##  Intact - 2B  -0.3209    -0.407   -0.2253
##  Intact - 1B  -0.3858    -0.474   -0.2923
##  8B - 2B      -0.2896    -0.379   -0.1981
##  8B - 1B      -0.3550    -0.448   -0.2608
##  2B - 1B      -0.0652    -0.158    0.0265
##
## Results are averaged over the levels of: Musician
## Point estimate displayed: median
## HPD interval probability: 0.95
```

```
log_rate_BF <- describe_posterior(brm_log_rate,
                                  estimate = "median", dispersion = TRUE,
                                  ci = .95, ci_method = "HDI",
                                  test = c("bayes_factor"))
```

```
## Warning: Bayes factors might not be precise.
##   For precise Bayes factors, sampling at least 40,000 posterior samples is
##   recommended.
```

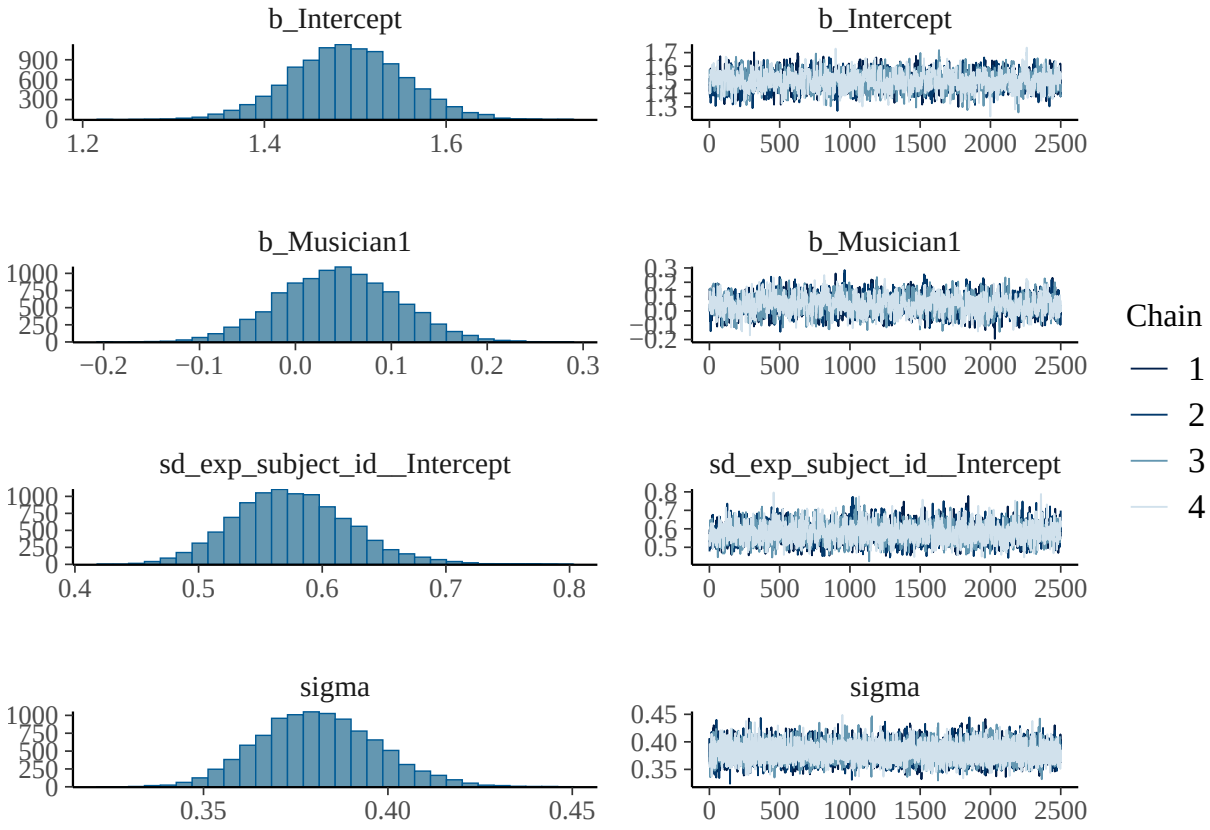
```
print(log_rate_BF, digits = 4)
```

```
## Summary of Posterior Distribution
##
## Parameter   | Median |    MAD |        95% CI |       BF |  Rhat |       ESS
## ------------------------------------------------------------------------------
## (Intercept) | 1.3102 | 0.0684 | [ 1.17, 1.44] | 8.75e+21 | 1.004 | 1195.0000
## Musician1   | 0.0469 | 0.0626 | [-0.07, 0.17] |    0.173 | 1.004 |  892.0000
## scramble2   | 0.0306 | 0.0471 | [-0.06, 0.12] |    0.114 | 1.000 | 8063.0000
## scramble3   | 0.3209 | 0.0461 | [ 0.23, 0.41] | 4.17e+04 | 1.000 | 7619.0000
## scramble4   | 0.3858 | 0.0465 | [ 0.29, 0.47] | 1.21e+07 | 1.000 | 7569.0000
```

Compare the full model to a model without scramble condition.

```
brm_log_rate_null <- brm(log_rate ~ Musician + (1|exp_subject_id), data = data,
                         prior = set_prior('normal(0, 0.5)', class = 'b'),
                         save_pars = save_pars(all = TRUE), iter = 5000,
                         file = '../models/E3_log_rate_null')
```

```
plot(brm_log_rate_null)
```




```
print(summary(brm_log_rate_null), digits = 4)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: log_rate ~ Musician + (1 | exp_subject_id)
##    Data: data (Number of observations: 380)
##   Draws: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
##          total post-warmup draws = 10000
##
## Multilevel Hyperparameters:
## ~exp_subject_id (Number of levels: 95)
##               Estimate Est.Error l-95% CI u-95% CI   Rhat Bulk_ESS Tail_ESS
## sd(Intercept)   0.5760    0.0472   0.4914   0.6771 1.0024     2141     4245
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI   Rhat Bulk_ESS Tail_ESS
## Intercept   1.4916    0.0619   1.3691   1.6143 1.0015     1503     3251
## Musician1   0.0453    0.0613  -0.0731   0.1657 1.0015     1409     2696
##
## Further Distributional Parameters:
```

```
##       Estimate Est.Error l-95% CI u-95% CI   Rhat Bulk_ESS Tail_ESS
## sigma   0.3812    0.0161   0.3517   0.4147 1.0000     9665     7663
##
## Draws were sampled 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).
```

```
BF_log_rate <- bayes_factor(brm_log_rate, brm_log_rate_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
```

```
print(BF_log_rate)
```

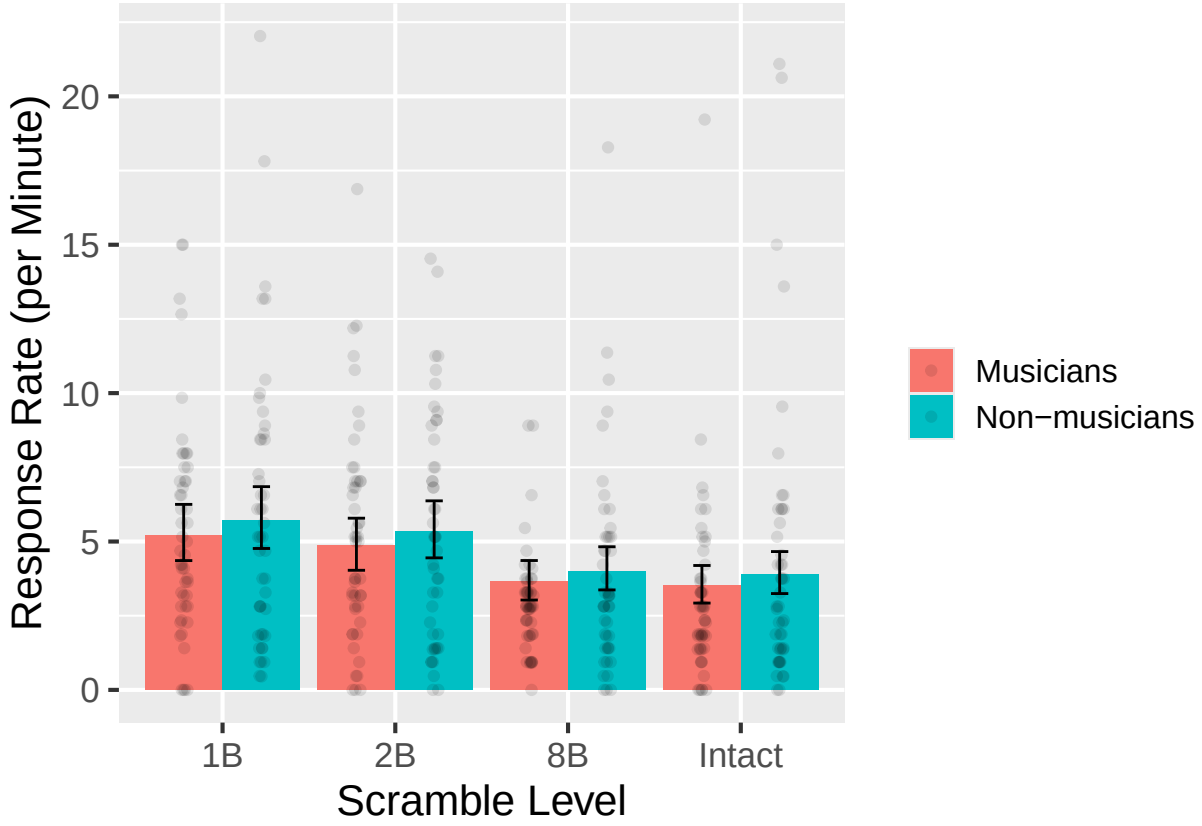
```
## Estimated Bayes factor in favor of brm_log_rate over brm_log_rate_null: 24732049331808900.00000
```

Visualize with posterior estimates and 95% CrI

```
posterior_est <- as.data.frame(emm_log_rate)
```

```
ggplot() +
  geom_col(aes(x = scramble, y = exp(emmean), fill = Musician), data = posterior_est,
           position = "dodge") +
  geom_errorbar(aes(x = scramble, ymin = exp(lower.HPD), ymax = exp(upper.HPD), fill = Musician),
                data = posterior_est, position = position_dodge(width = 0.9), width = 0.2) +
  geom_point(aes(x = scramble, y = mean_response_rate, fill = Musician), data = data,
             position = position_jitterdodge(dodge.width = 0.9, jitter.width = 0.1), alpha = 0.1) +
  theme_gray(base_size = 16) +
  scale_x_discrete(limits = rev) +
  xlab('Scramble Level') +
  ylab('Response Rate (per Minute)') +
  scale_fill_discrete(name="", labels=c('Musicians', 'Non-musicians')) +
  theme(legend.text = element_text(size = 12))
```

```
## Warning in geom_errorbar(aes(x = scramble, ymin = exp(lower.HPD), ymax =
## exp(upper.HPD), : Ignoring unknown aesthetics: fill
```




```
ggsave('../figures/Fig3_rate.png', width = 7, height = 5)
```