

# Working title: Examining treatment and generalization effects using adaptive distributed practice and stimuli variability, a single-case experimental design

Last updated: 2021-07-07

Yina Quique <sup>a\*</sup>, Robert Cavanaugh <sup>b</sup>, Erica Lescht <sup>b</sup>, William S. Evans <sup>b</sup>

<sup>a</sup>Northwestern University Chicago IL, USA; <sup>b</sup>Department of Communication Sciences and Disorders, University of Pittsburgh, Pittsburgh, PA USA;

## Description

This document reports model comparison code using a bayesfactor approach: [https://easystats.github.io/bayestestR/articles/bayes\\_factors.html](https://easystats.github.io/bayestestR/articles/bayes_factors.html)

- Makowski, D., Ben-Shachar, M. S., & Lüdtke, D. (2019). bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework. Journal of Open Source Software, 4(40), 1541. <https://doi.org/10.21105/joss.01541>
- Makowski, D., Ben-Shachar, M. S., Chen, S. H. A., & Lüdtke, D. (2019). Indices of Effect Existence and Significance in the Bayesian Framework. Retrieved from 10.3389/fpsyg.2019.02767

Under which model are the observed data more probable?

We will use the model with the squared term if there is at least moderate evidence that the data are more probable in this model. If there is weak or uncertain evidence, or evidence against the model with the squared term, we will use the more parsimonious model which excludes it.

## Load packages

```
# install.packages("easystats", repos = "https://easystats.r-universe.dev")
# install.packages("here")
# install.packages("janitor")

# Data manipulation and reporting
library(knitr)
library(here)
library(tidyverse)
library(janitor)
# Stats
library(brms)
library(bayesplot)
library(tidybayes)
library(effectsize)
library(bayestestR)
library(see)
```

```
library(parallel)
ncores = detectCores()
options(mc.cores = parallel::detectCores())
```

## Read in data:

Read in and clean data

```
df_all <- read_csv(here('data', '2020-12-14-hutiema-normal-DEID.csv')) %>%
  select(target_word, gen, complexity, group, participant,
         phase, session, date, response = hand_correct_acc,
         baselineSlope, levelChange, slopeChange) %>%
  distinct() %>%
  mutate(treated = as.factor(ifelse(group == 'untreated', 0, 1)),
         levelChange = as.factor(levelChange))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

- Setup separate data-frames for direct and generalization.
- Separate out followup data.

```
# direct training with untreated
df_direct <- df_all %>%
  filter(gen == 'direct' & session <= 16)%>%
  mutate(slopeChange2 = slopeChange^2)

# generalization with untreated
df_gen <- df_all %>%
  filter(gen != 'direct' | group == 'untreated',
         session <= 16)%>%
  mutate(slopeChange2 = slopeChange^2)
```

## Prior distributions

BRMS uses a flat (uninformative) prior on the beta coefficients. Here, we will use a weakly informative prior to constrain sampling to reasonable values. A Student-t distribution as indicated produces prior distributions roughly within  $\pm 5/-5$  log-odds, which is a range of reasonable values (e.g. between roughly 0 and 1). The default prior on the variance term is a half-normal with a standard deviation of 2. The prior on the correlation matrix is weakly informative around 0. These priors are based on recommendations from the Stan team and our prior work using the interrupted time series models previously.

The following is a prior predictive check to visualize the choice of prior distributions.

```
beta_coef_prior = c(prior(student_t(3, 0, 2), class = b),
                    prior(normal(0,2), class = sd),
                    prior(lkj(2), class = L))
```

Interrupted time series formula. The 0 + Intercept syntax allows for a prior on the intercept.

```
ITTS.formula <- bf(response ~ 0 + Intercept +
                  baselineSlope +
                  levelChange +
                  slopeChange +
                  (baselineSlope | target_word) +
                  (0+levelChange | target_word) +
                  (0+slopeChange | target_word))
```

```

ITTS.formula_sq <- bf(response ~ 0 + Intercept +
  baselineSlope +
  levelChange +
  slopeChange +
  slopeChange2 +
  (baselineSlope | target_word) +
  (0+levelChange | target_word) +
  (0+slopeChange | target_word) +
  (0+slopeChange2 | target_word))

# not run to save knitting time.
prior_check <- brm(ITTS.formula,
  data = df_direct %>%
    filter(participant == 'participant1' & treated == 1),
  iter = 1000,
  warmup = 500,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  sample_prior = "only",
  prior = beta_coef_prior,
  file = here("output2", "prior_check"),
  file_refit = "on_change"
)

```

Visualize the prior distributions:

```

data = mcmc_plot(prior_check, type = "hist")$data
choices = unique(data$Parameter)[c(1,2,7,11)]
data %>%
  filter(Parameter %in% choices) %>%
  mutate(Parameter = gsub('_', '.', Parameter)) %>%
  ggplot(aes(x = value, fill = Parameter)) +
  geom_density() +
  facet_wrap(~Parameter, scales = 'free') +
  theme_tidybayes()

```

We can also look to see where the data lies.

```

data %>%
  filter(Parameter %in% choices) %>%
  mutate(Parameter = gsub('_', '.', Parameter)) %>%
  group_by(Parameter) %>%
  summarize(val = round(quantile(value, probs = c(0.05, .34, .5, .66, .95)), 2)) %>%
  mutate(quantile = paste0(
    as.character(rep(c(0.05, .34, .5, .66, .95)*100), 3), "%")
  ) %>%
  pivot_wider(names_from = quantile, values_from = val) %>%
  kable()

```

Parameter	5%	34%	50%	66%	95%
b	-4.89	-0.92	0.01	1.02	4.95
cor	-0.73	-0.21	0.00	0.21	0.70

Parameter	5%	34%	50%	66%	95%
sd	0.15	0.82	1.30	1.88	3.99

## Models: Learning Effects - Treated, Control, & Generalization items

### Treated & Control:

```
treated_direct_p1 <- brm(ITTS.formula,
  data = df_direct %>%
    filter(participant == 'participant1' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  save_pars = save_pars(all = TRUE),
  control = list(adapt_delta = .9),
  prior = beta_coef_prior,
  file = here("output2", "treated_direct_p1"),
  file_refit = "on_change"
)
```

```
treated_direct_p1_sq <- brm(ITTS.formula_sq,
  data = df_direct %>%
    filter(participant == 'participant1' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  save_pars = save_pars(all = TRUE),
  control = list(adapt_delta = .9),
  prior = beta_coef_prior,
  file = here("output2", "treated_direct_p1_sq"),
  file_refit = "on_change"
)
```

```
bf_tx_d_p1 <- bayestestR::bayesfactor_models(treated_direct_p1, treated_direct_p1_sq) %>%
  rownames_to_column(var = "model") %>%
  select(1,3)
```

```
print(paste("There is", interpret_bf(bf_tx_d_p1[2,2], log = T, include_value = T), "the model with the s
```

#### 1. model treated items, participant 1

```
## [1] "There is extreme evidence (BF = 1/206.91) against the model with the square term."
```

```
rm(treated_direct_p1)
rm(treated_direct_p1_sq)
```

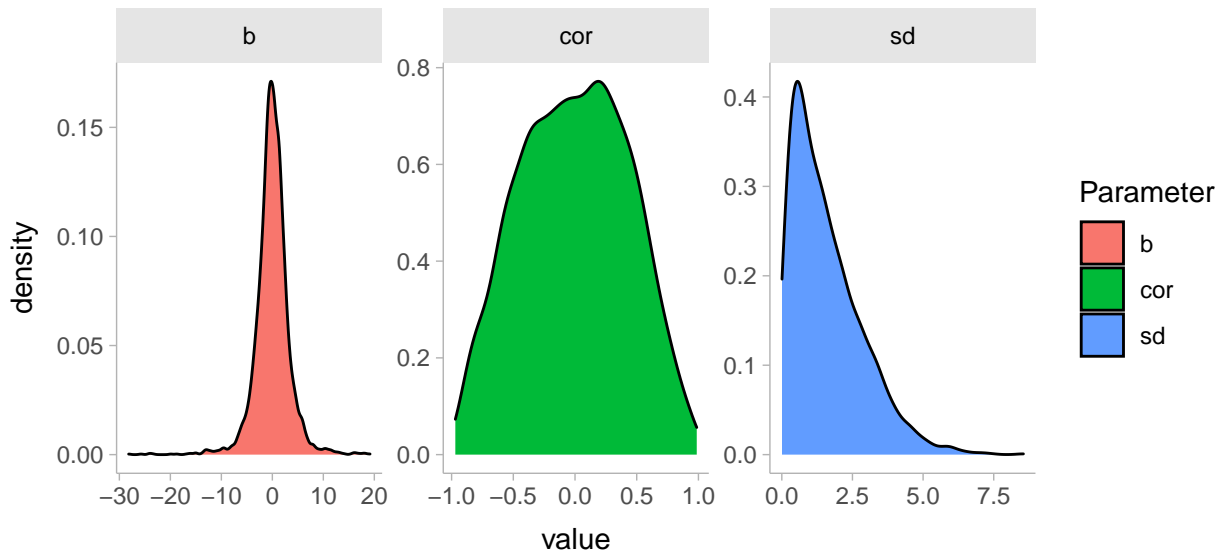


Figure 1: Density plots of weakly informative prior distributions

```
untreated_direct_p1 <- brm(ITTS.formula,
  data = df_direct %>%
    filter(participant == 'participant1' & treated == 0),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "untreated_direct_p1"),
  file_refit = "on_change"
)
```

```
untreated_direct_p1_sq <- brm(ITTS.formula_sq,
  data = df_direct %>%
    filter(participant == 'participant1' & treated == 0),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "untreated_direct_p1_sq"),
  file_refit = "on_change"
)
```

```
bf_untx_d_p1 <- bayestestR::bayesfactor_models(untreated_direct_p1, untreated_direct_p1_sq) %>%
  rownames_to_column(var = "model") %>%
  select(1,3)
```

```
print(paste("There is", interpret_bf(bf_untx_d_p1[2,2], log = T, include_value = T), "the model with the"))
```

## 2. model untreated items, participant 1

```
## [1] "There is extreme evidence (BF = 1/1.89e+04) against the model with the square term."
```

```
rm(untreated_direct_p1)
rm(untreated_direct_p1_sq)
```

```
treated_direct_p2 <- brm(ITTS.formula,
  data = df_direct %>%
    filter(participant == 'participant2' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "treated_direct_p2"),
  file_refit = "on_change"
)
```

```
treated_direct_p2_sq <- brm(ITTS.formula_sq,
  data = df_direct %>%
    filter(participant == 'participant2' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "treated_direct_p2_sq"),
  file_refit = "on_change"
)
```

```
bf_tx_d_p2 <- bayestestR::bayesfactor_models(treated_direct_p2, treated_direct_p2_sq) %>%
  rownames_to_column(var = "model") %>%
  select(1,3)
```

```
print(paste("There is", interpret_bf(bf_tx_d_p2[2,2], log = T, include_value = T), "the model with the
```

### 3. model treated items, participant 2

```
## [1] "There is anecdotal evidence (BF = 1.72) in favour of the model with the square term."
```

```
rm(treated_direct_p2)
rm(treated_direct_p2_sq)
```

```
untreated_direct_p2 <- brm(ITTS.formula,
  data = df_direct %>%
    filter(participant == 'participant2' & treated == 0),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "untreated_direct_p2"),
  file_refit = "on_change"
)
```

```
untreated_direct_p2_sq <- brm(ITTS.formula_sq,
  data = df_direct %>%
    filter(participant == 'participant2' & treated == 0),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "untreated_direct_p2_sq"),
  file_refit = "on_change"
)
```

```
bf_untx_d_p2 <- bayestestR::bayesfactor_models(untreated_direct_p2, untreated_direct_p2_sq) %>%
  rownames_to_column(var = "model") %>%
  select(1,3)
print(paste("There is", interpret_bf(bf_untx_d_p2[2,2], log = T, include_value = T), "the model with the
```

### 4. model untreated items, participant 2

```
## [1] "There is extreme evidence (BF = 1/1.05e+04) against the model with the square term."
```

```
rm(untreated_direct_p2)
rm(untreated_direct_p2_sq)
```

## Generalization Items

```
treated_gen_p1 <- brm(ITTS.formula,
  data = df_gen %>%
    filter(participant == 'participant1' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "treated_gen_p1"),
  file_refit = "on_change"
)
```

```
treated_gen_p1_sq <- brm(ITTS.formula_sq,
  data = df_gen %>%
    filter(participant == 'participant1' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "treated_gen_p1_sq"),
  file_refit = "on_change"
)
```

```
bf_tx_gen_p1 <- bayestestR::bayesfactor_models(treated_gen_p1, treated_gen_p1_sq) %>%
  rownames_to_column(var = "model") %>%
  select(1,3)
```

```
print(paste("There is", interpret_bf(bf_tx_gen_p1[2,2], log = T, include_value = T), "the model with the"))
```

### 5. model with treated items, participant 1

```
## [1] "There is very strong evidence (BF = 49.82) in favour of the model with the square term."
```

```
rm(treated_gen_p1)
rm(treated_gen_p1_sq)
```



```
treated_gen_p2 <- brm(ITTS.formula,
  data = df_gen %>%
  filter(participant == 'participant2' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "treated_gen_p2"),
  file_refit = "on_change"
)
```

```
treated_gen_p2_sq <- brm(ITTS.formula_sq,
  data = df_gen %>%
  filter(participant == 'participant2' & treated == 1),
  iter = 14000,
  warmup = 2000,
  inits = 'random',
  family = bernoulli(),
  chains = 4,
  cores = 4,
  seed = 42,
  save_pars = save_pars(all = TRUE),
  prior = beta_coef_prior,
  file = here("output2", "treated_gen_p2_sq"),
  file_refit = "on_change"
)
```

```
bf_tx_gen_p2 <- bayestestR::bayesfactor_models(treated_gen_p2, treated_gen_p2_sq) %>%
  rownames_to_column(var = "model") %>%
  select(1,3)
```

```
print(paste("There is", interpret_bf(bf_tx_gen_p2[2,2], log = T, include_value = T), "the model with the"))
```

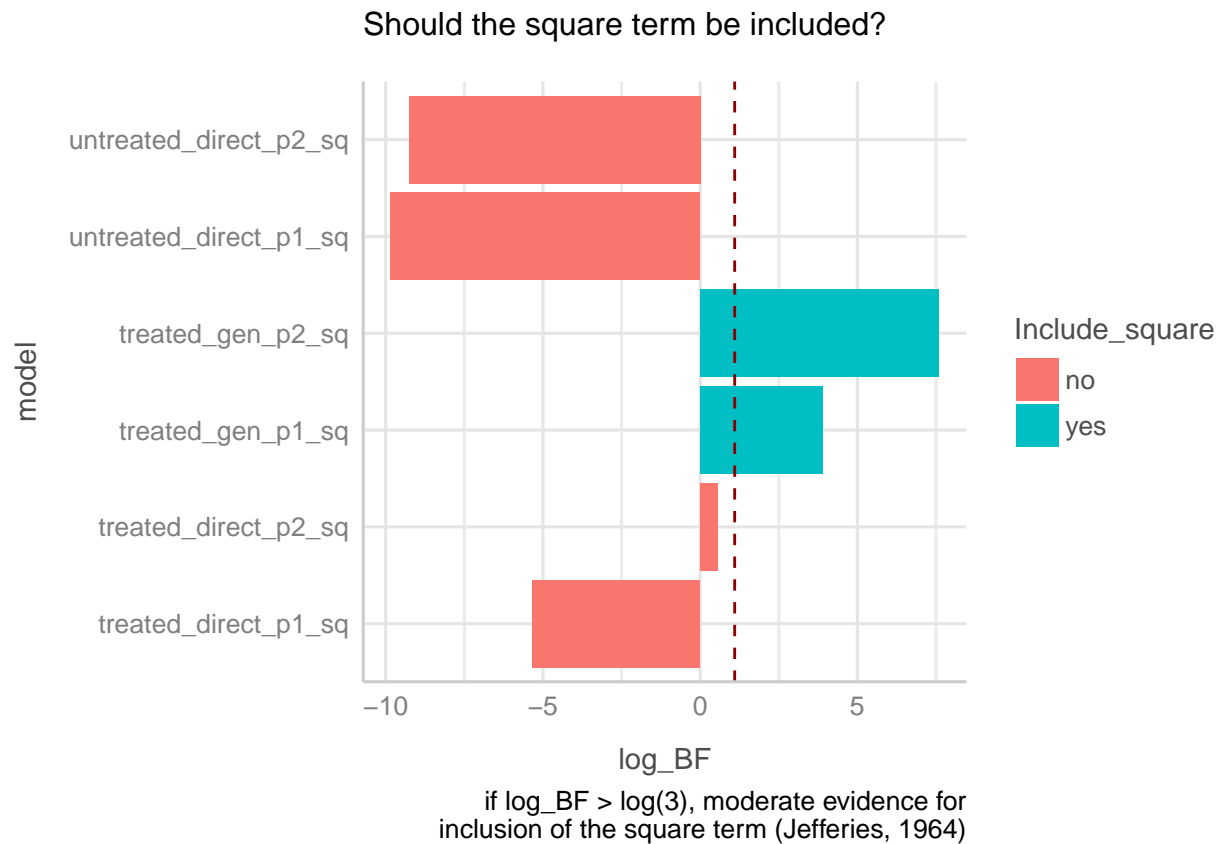
## 6. model with treated items, participant 2

```
## [1] "There is extreme evidence (BF = 1.99e+03) in favour of the model with the square term."
```

```
rm(treated_gen_p2)
rm(treated_gen_p2_sq)
```

```
df <- bind_rows(bf_tx_d_p1, bf_tx_d_p2, bf_untx_d_p1,
  bf_untx_d_p2, bf_tx_gen_p1, bf_tx_gen_p2)
attributes(df) <- NULL
df <- bind_cols(df)
colnames(df) <- c("model", "log_BF")
```

```
df %>%
  filter(
    str_detect(model, "_sq")) %>%
  mutate(Include_square = ifelse(log_BF < log(3), "no", "yes")) %>%
  ggplot(aes(x = model,
             y = log_BF,
             fill = Include_square)) +
  geom_col() +
  geom_hline(aes(yintercept = log(3)), color = "darkred", linetype = "dashed") +
  coord_flip(clip = "off") +
  labs(title = "Should the square term be included?",
       caption = "if log_BF > log(3), moderate evidence for
inclusion of the square term (Jefferies, 1964)")
```



## Session Info

Note, this was run via docker/rocker on digital ocean using <https://hub.docker.com/r/asachet/rocker-stan> with a few additional packages install:

- <https://github.com/easystats/easystats>
- <https://garhtarr.github.io/meatR/janitor.html>

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.2 LTS
##
```

```

## Matrix products: default
## BLAS/LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-pthread0.3.8.so
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=C
## [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel stats      graphics grDevices utils      datasets methods
## [8] base
##
## other attached packages:
## [1] see_0.6.4      bayestestR_0.10.5  effectsize_0.4.5.1 tidybayes_2.3.1
## [5] bayesplot_1.8.1 brms_2.15.0        Rcpp_1.0.6          janitor_2.1.0
## [9] forcats_0.5.1  stringr_1.4.0      dplyr_1.0.7         purrr_0.3.4
## [13] readr_1.4.0     tidyr_1.1.3        tibble_3.1.2        ggplot2_3.3.4
## [17] tidyverse_1.3.1 here_1.0.1          knitr_1.33
##
## loaded via a namespace (and not attached):
## [1] readxl_1.3.1      backports_1.2.1    plyr_1.8.6
## [4] igraph_1.2.6      splines_4.1.0      svUnit_1.0.6
## [7] crosstalk_1.1.1   rstantools_2.1.1   inline_0.3.19
## [10] digest_0.6.27     htmltools_0.5.1.1  rsconnect_0.8.18
## [13] fansi_0.5.0       magrittr_2.0.1     modelr_0.1.8
## [16] RcppParallel_5.1.4 matrixStats_0.59.0 xts_0.12.1
## [19] prettyunits_1.1.1 colorspace_2.0-2   rvest_1.0.0
## [22] ggdist_2.4.1      haven_2.4.1        xfun_0.24
## [25] callr_3.7.0       crayon_1.4.1       jsonlite_1.7.2
## [28] lme4_1.1-27.1     zoo_1.8-9          glue_1.4.2
## [31] gtable_0.3.0      V8_3.4.2           distributional_0.2.2
## [34] pkgbuild_1.2.0    rstan_2.21.2       abind_1.4-5
## [37] scales_1.1.1      mvtnorm_1.1-2      DBI_1.1.1
## [40] miniUI_0.1.1.1    xtable_1.8-4       stats4_4.1.0
## [43] StanHeaders_2.21.0-7 DT_0.18            datawizard_0.1.0.9000
## [46] htmlwidgets_1.5.3 httr_1.4.2         threejs_0.3.3
## [49] arrayhelpers_1.1-0 ellipsis_0.3.2     farver_2.1.0
## [52] pkgconfig_2.0.3   loo_2.4.1          dbplyr_2.1.1
## [55] utf8_1.2.1        labeling_0.4.2     tidyselect_1.1.1
## [58] rlang_0.4.11      reshape2_1.4.4     later_1.2.0
## [61] munsell_0.5.0     cellranger_1.1.0   tools_4.1.0
## [64] cli_2.5.0         generics_0.1.0     broom_0.7.8
## [67] ggirges_0.5.3     evaluate_0.14      fastmap_1.1.0
## [70] yaml_2.2.1        processx_3.5.2     fs_1.5.0
## [73] nlme_3.1-152      mime_0.11          projpred_2.0.2
## [76] xml2_1.3.2        compiler_4.1.0     shinythemes_1.2.0
## [79] rstudioapi_0.13   gamm4_0.2-6        curl_4.3.2
## [82] reprex_2.0.0      stringi_1.6.2      highr_0.9
## [85] ps_1.6.0          parameters_0.14.0.1 Brodingtonag_1.2-6
## [88] lattice_0.20-44   Matrix_1.3-3       nloptr_1.2.2.2
## [91] markdown_1.1      shinyjs_2.0.0      vctrs_0.3.8

```

## [94] pillar_1.6.1	lifecycle_1.0.0	bridgesampling_1.1-2
## [97] insight_0.14.2	httpuv_1.6.1	R6_2.5.0
## [100] promises_1.2.0.1	gridExtra_2.3	codetools_0.2-18
## [103] boot_1.3-28	colourpicker_1.1.0	MASS_7.3-54
## [106] gtools_3.9.2	assertthat_0.2.1	rprojroot_2.0.2
## [109] withr_2.4.2	shinystan_2.5.0	mgcv_1.8-35
## [112] hms_1.1.0	grid_4.1.0	coda_0.19-4
## [115] minqa_1.2.4	rmarkdown_2.9	snakecase_0.11.0
## [118] shiny_1.6.0	lubridate_1.7.10	base64enc_0.1-3
## [121] dygraphs_1.1.1.6		