test log-transformation

Victor Møller Poulsen

setup

data

```
d <- read_csv("/work/50114/MAG/data/modeling/psych_replication_matched.csv") %>%
  mutate(log_teamsize = log(n_authors),
         condition_coded = ifelse(condition == "experiment", 1, 0),
         condition_fct = as_factor(condition),
         teamsize_scaled = (n_authors-min(n_authors))/(max(n_authors)-min(n_authors)),
         days_after_2010_scaled = days_after_2010/max(days_after_2010),
         teamsize_log = log(n_authors),
         id_match = as_factor(match_group),
         id_fct = as_factor(PaperId)) %>% # because min = 0
  glimpse()
```

```
## Rows: 1560 Columns: 6

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): condition
## dbl (5): match_group, n_authors, PaperId, days_after_2010, c_5

##
## 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.

## Rows: 1,560
## Columns: 14
## $ match_group            <dbl> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8,~
## $ condition              <chr> "experiment", "control", "control", "experiment~
## $ n_authors              <dbl> 3, 3, 1, 1, 4, 4, 5, 5, 2, 2, 2, 2, 3, 3, 5, 5,~
## $ PaperId                <dbl> 2330249536, 2003350634, 2385753682, 2395494269,~
## $ days_after_2010        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ c_5                    <dbl> 10, 0, 0, 0, 310, 0, 2, 17, 0, 13, 2, 13, 0, 0,~
## $ log_teamsize           <dbl> 1.0986123, 1.0986123, 0.0000000, 0.0000000, 1.3~
## $ condition_coded        <dbl> 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,~
## $ condition_fct          <fct> experiment, control, control, experiment, exper~
## $ teamsize_scaled        <dbl> 0.03333333, 0.03333333, 0.00000000, 0.00000000,~
## $ days_after_2010_scaled <dbl> 0.000000000, 0.000000000, 0.000000000, 0.000000~
## $ teamsize_log           <dbl> 1.0986123, 1.0986123, 0.0000000, 0.0000000, 1.3~
## $ id_match               <fct> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8,~
## $ id_fct                 <fct> 2330249536, 2003350634, 2385753682, 2395494269,~
```

model formulae

same model-specifications as test_interactions2. just differs by being log of teamsize rather than 0-1 scaling.

```
f_logteam_0 <- bf(c_5 ~ 0 + condition_fct + condition_fct:teamsize_log + (1|id_match))
f_logteam_1 <- bf(c_5 ~ 1 + condition_fct + condition_fct:teamsize_log + (1|id_match))
f_logteam_01 <- bf(c_5 ~ 0 + Intercept + condition_fct + condition_fct:teamsize_log + (1|id_match))
```

Just doing negbinomial() for now, since we had Rhat issues for both negative binomial and zero-inflated negative binomial (does not seem to be the main cause of issues).

f_team_0: b, sd, shape f_team_1: b, Intercept, sd, shape f_team_01: b, sd, shape (Intercept becomes b).

set priors

```
# negbin baseline
negbin_0 <- c(prior(gamma(0.01, 0.01), class = shape),
              prior(normal(0, 1), class = b),
              prior(normal(0, 1), class = sd)) # a wild guess

# zinegbin baseline
negbin_1 = c(prior(gamma(0.01, 0.01), class = shape),
             prior(normal(0, 1), class = b),
             prior(normal(0, 1), class = Intercept),
             prior(normal(0, 1), class = sd)) # a wild guess

# can be used for all interactions (without thinking)
negbin_01 <- c(prior(gamma(0.01, 0.01), class = shape),
              prior(normal(0, 1), class = b),
              prior(normal(0, 1), class = sd)) # a wild guess
```

sample prior only

Some warnings and divergences.

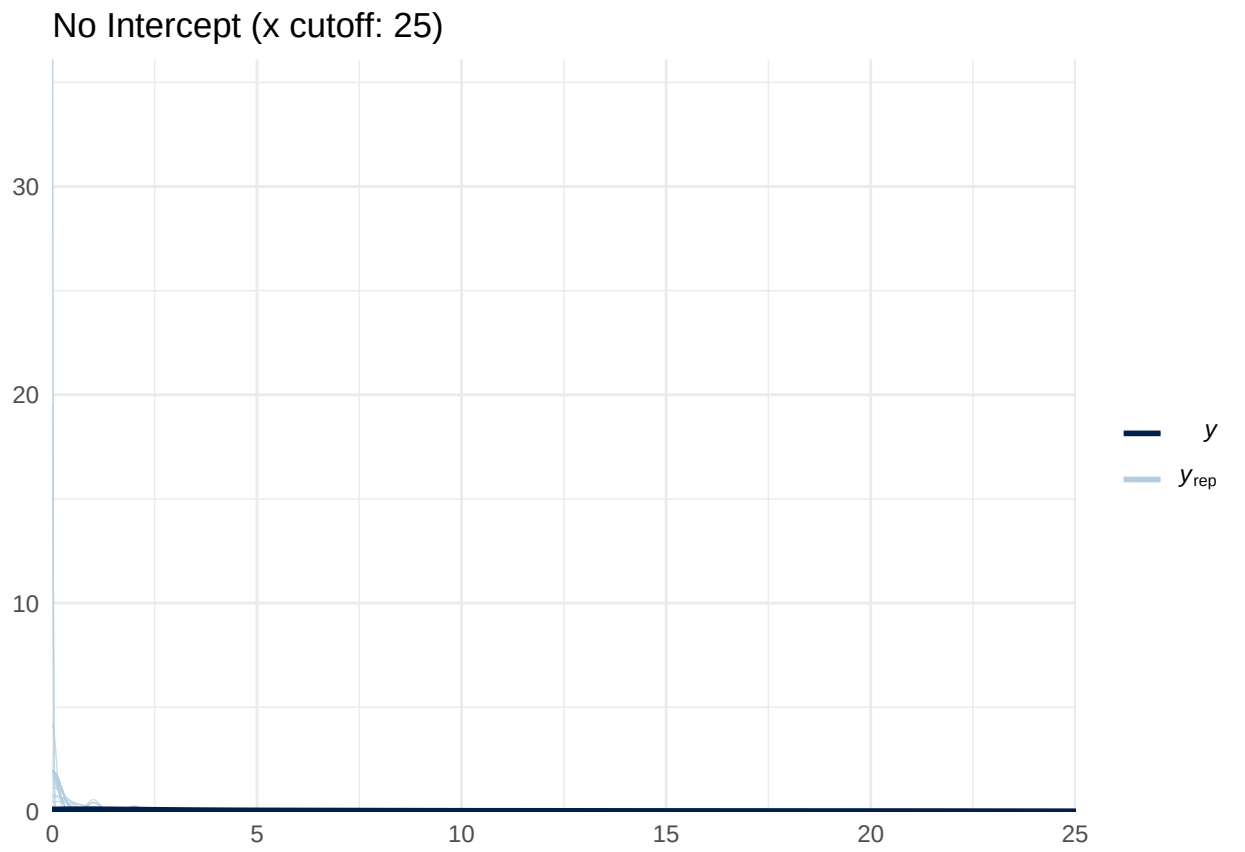
check priors

```
prior_check <- function(model, ndraws, title, xmax){

  pp_check(model,
           ndraws = ndraws) +
    labs(title = title) +
    theme_minimal() +
    xlim(0, xmax)

}
```

```
prior_check(negbin_prior_log_0, 100, "No Intercept (x cutoff: 25)", 25)
```

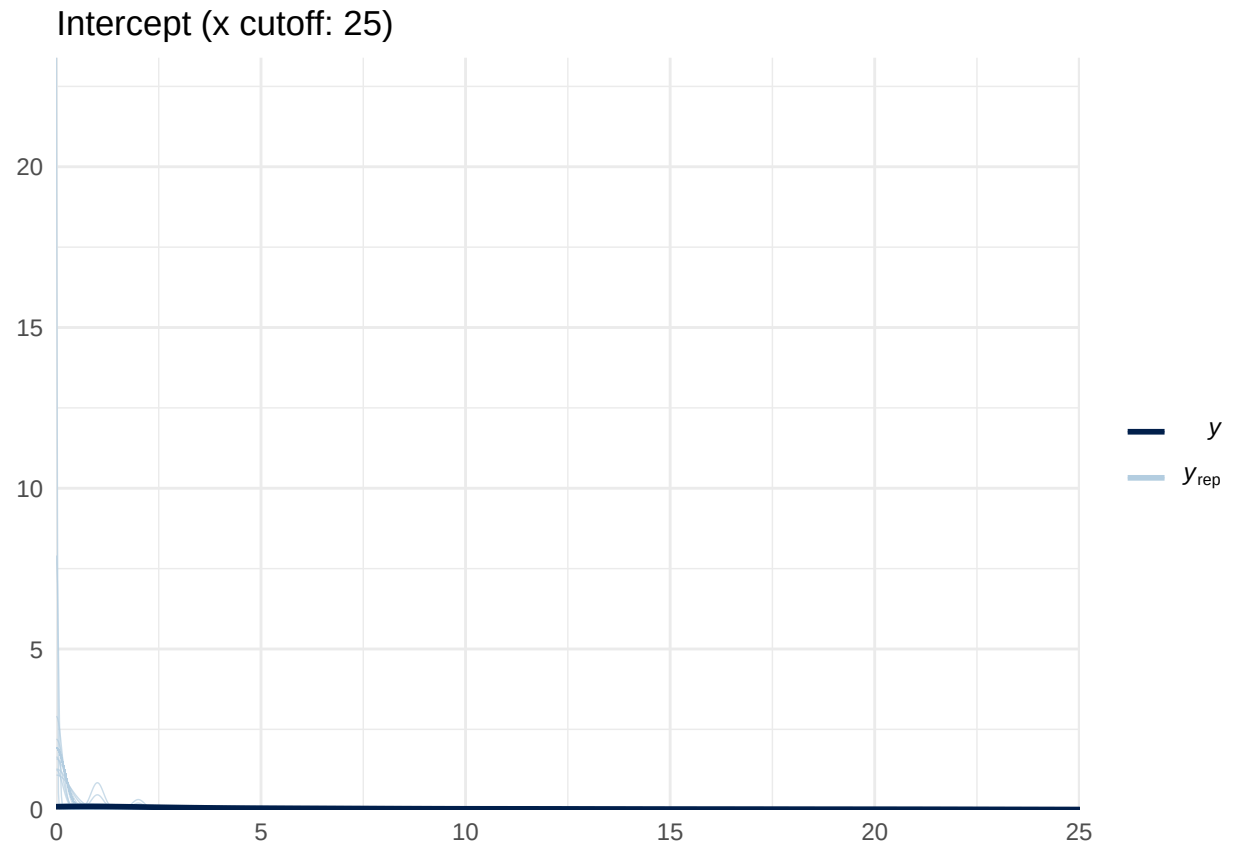
```
## Warning: Removed 1299 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 253 rows containing non-finite values (stat_density).
```




```
prior_check(negbin_prior_log_1, 100, "Intercept (x cutoff: 25)", 25)
```

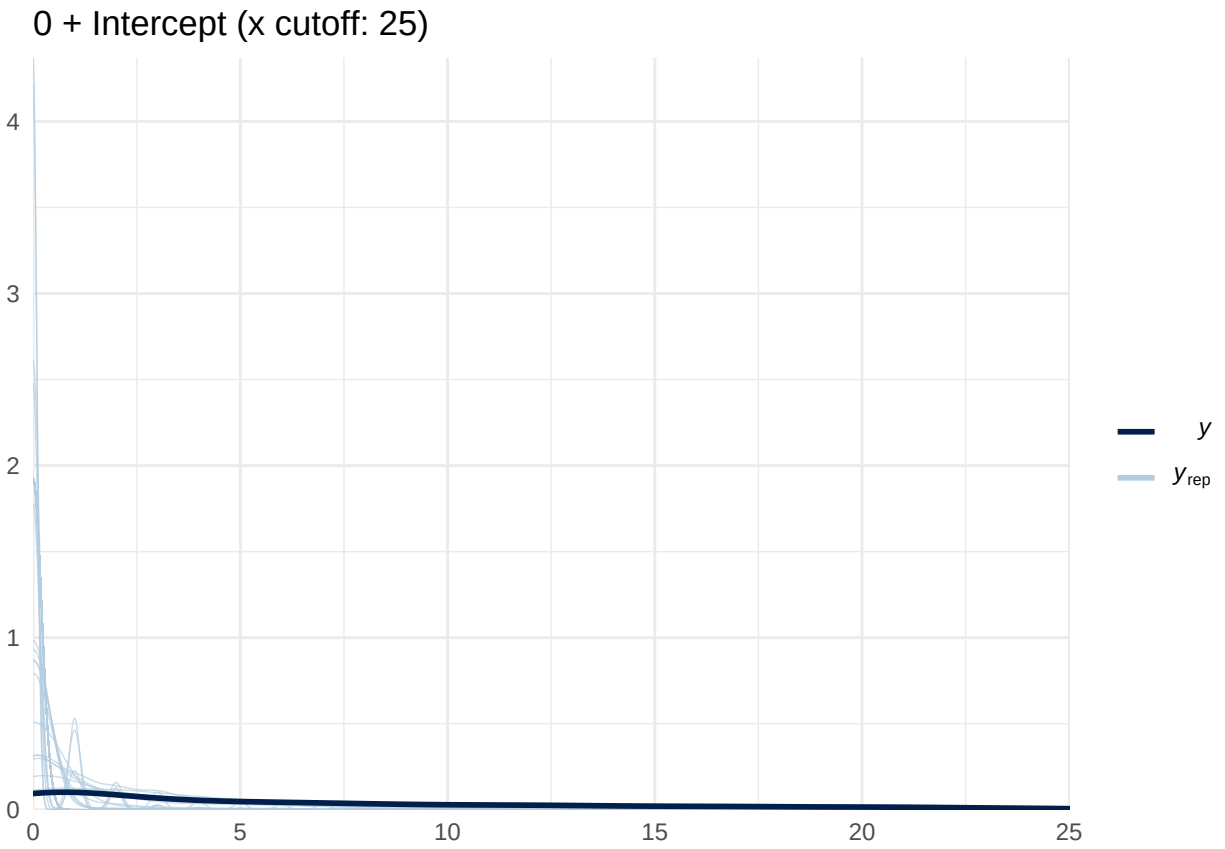
```
## Warning: Removed 60 rows containing non-finite values (stat_density).

## Warning: Removed 253 rows containing non-finite values (stat_density).
```




```
prior_check(negbin_prior_log_01, 100, "0 + Intercept (x cutoff: 25)", 25)
```

```
## Warning: Removed 1271 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 253 rows containing non-finite values (stat_density).
```



fit models

more pareto k than in the non-log-transformed? still fewest in the 0 + Intercept model.

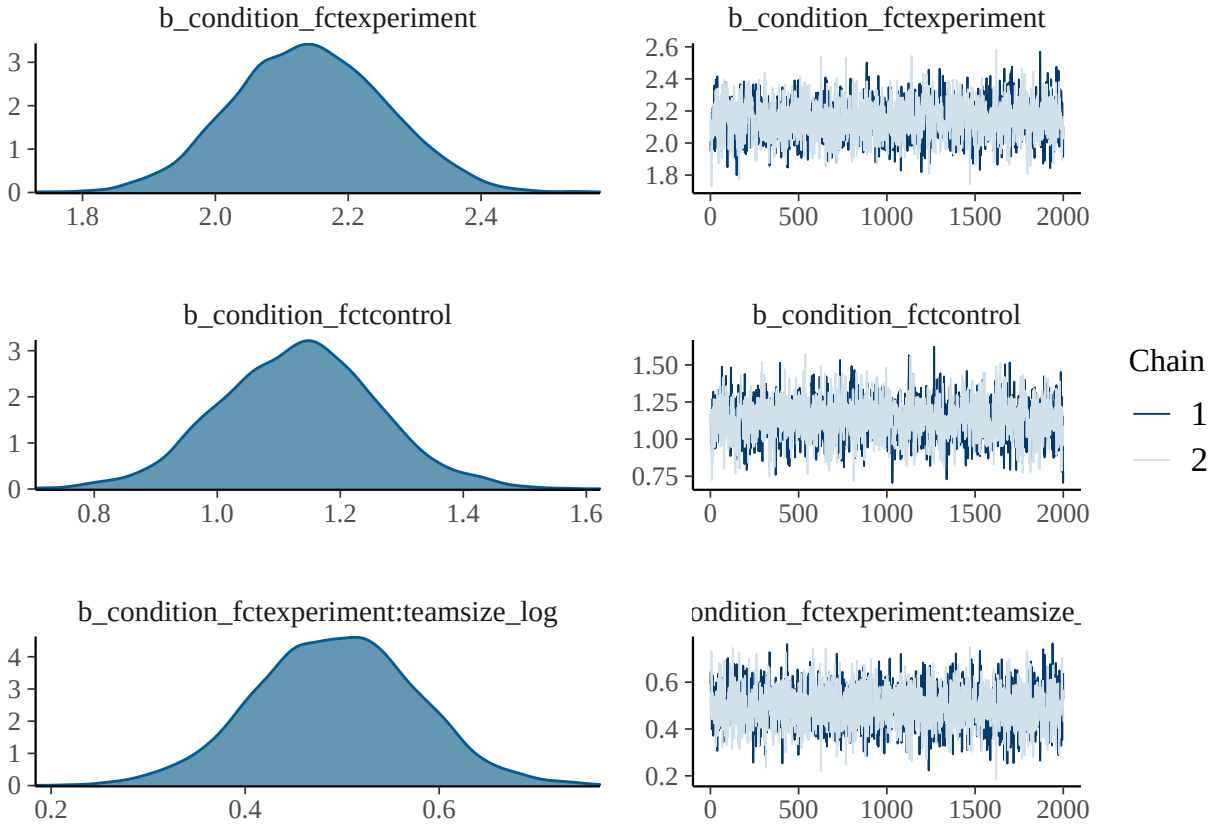
```
## baseline
negbin_post_log_0 <- fit_model(
  family = negbinomial(),
  formula = f_logteam_0,
  prior = negbin_0,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_log_0"
)

## baseline
negbin_post_log_1 <- fit_model(
  family = negbinomial(),
  formula = f_logteam_1,
  prior = negbin_1,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_log_1"
)

## baseline
negbin_post_log_01 <- fit_model(
```

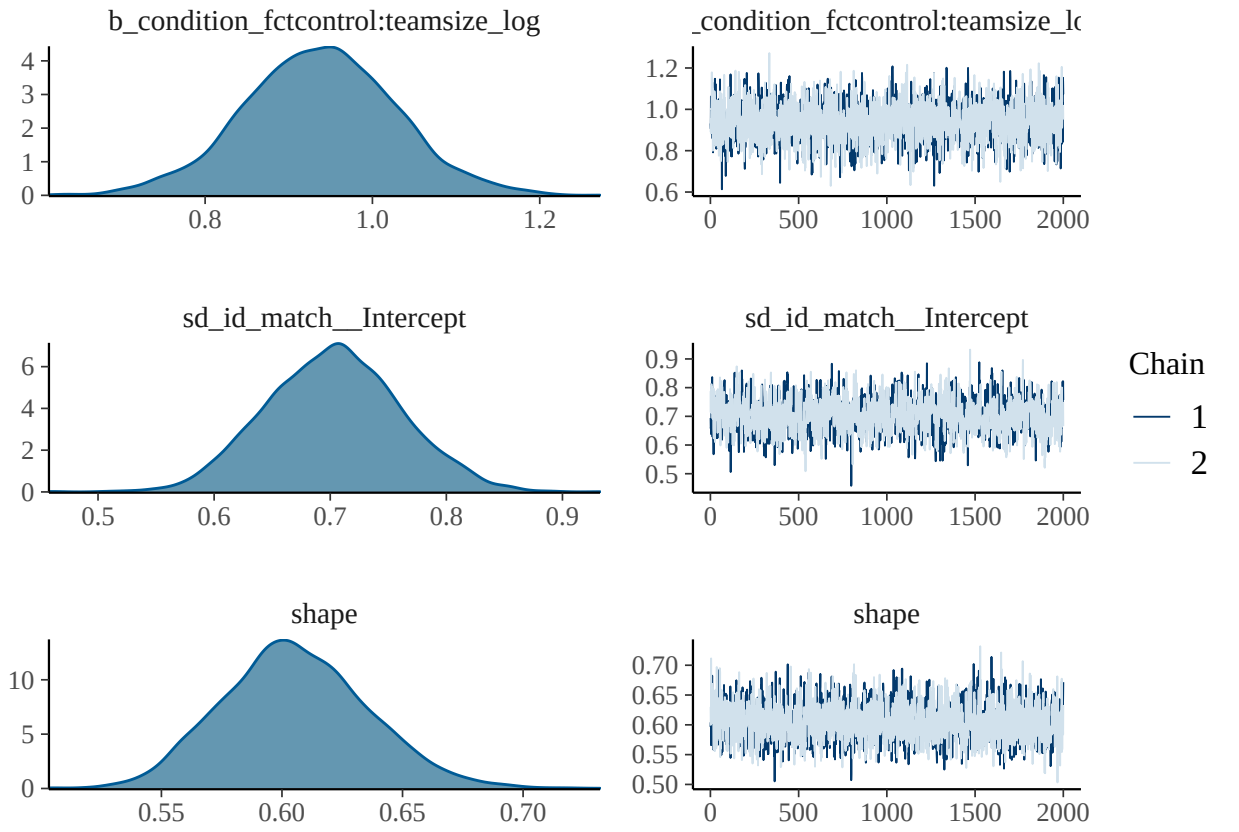
```
  family = negbinomial(),
  formula = f_logteam_01,
  prior = negbin_01,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_log_01"
)
```

check traces

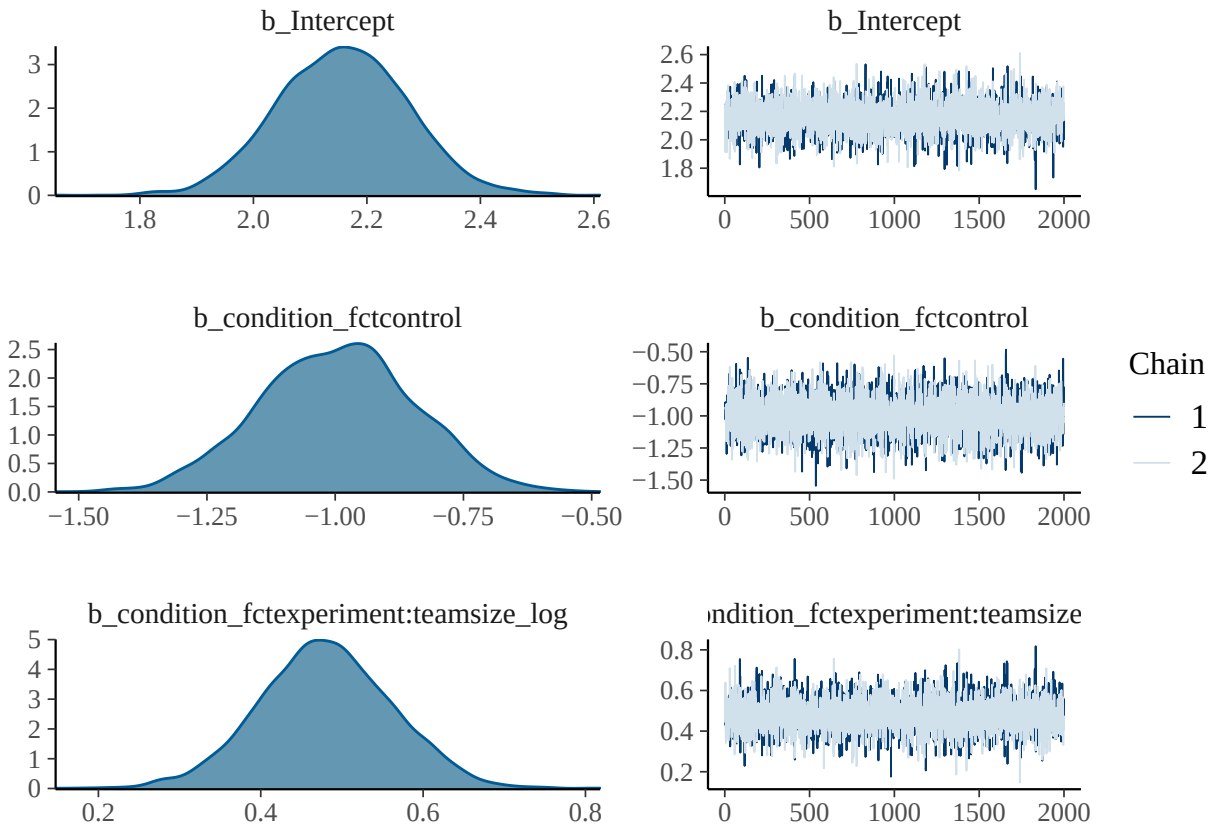
looks ok.

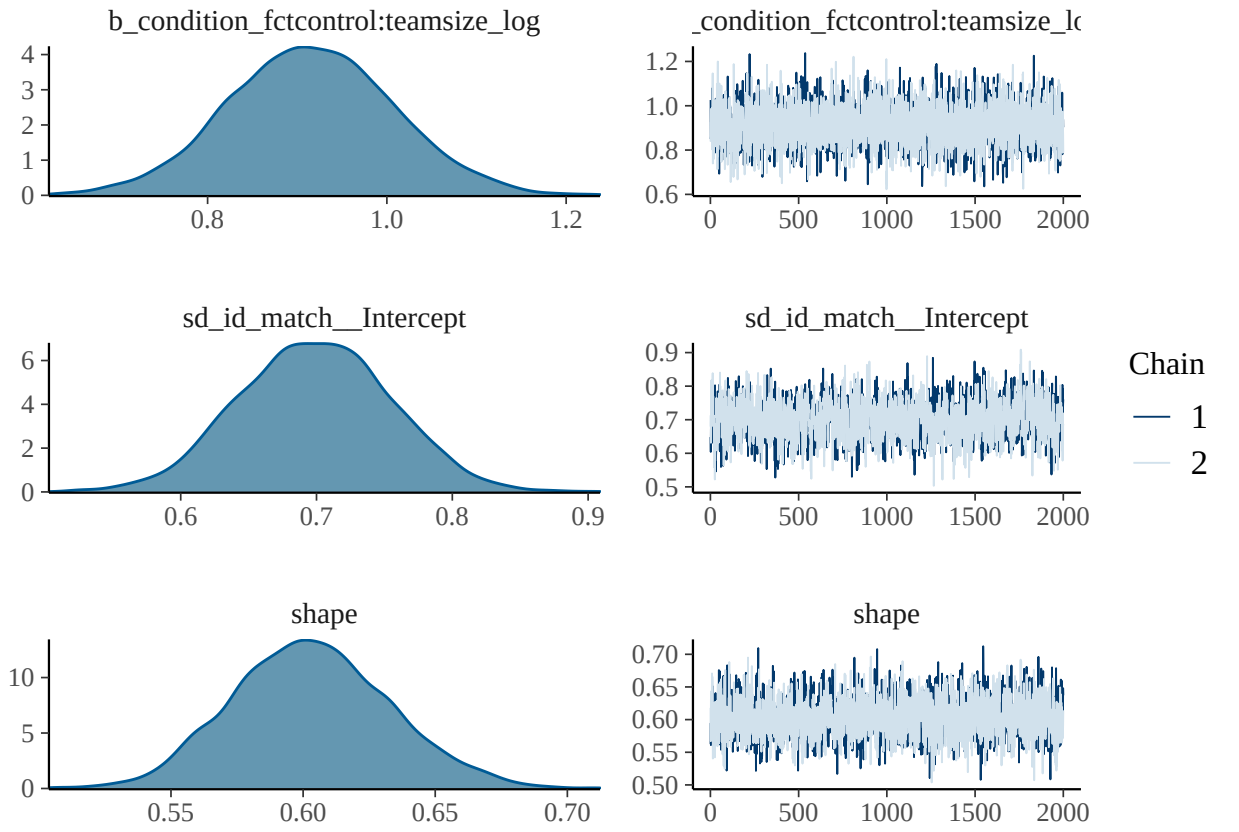
```
# some auto-correlation
# effect almost entirely in random effect
plot(negbin_post_log_0, N = 3)
```



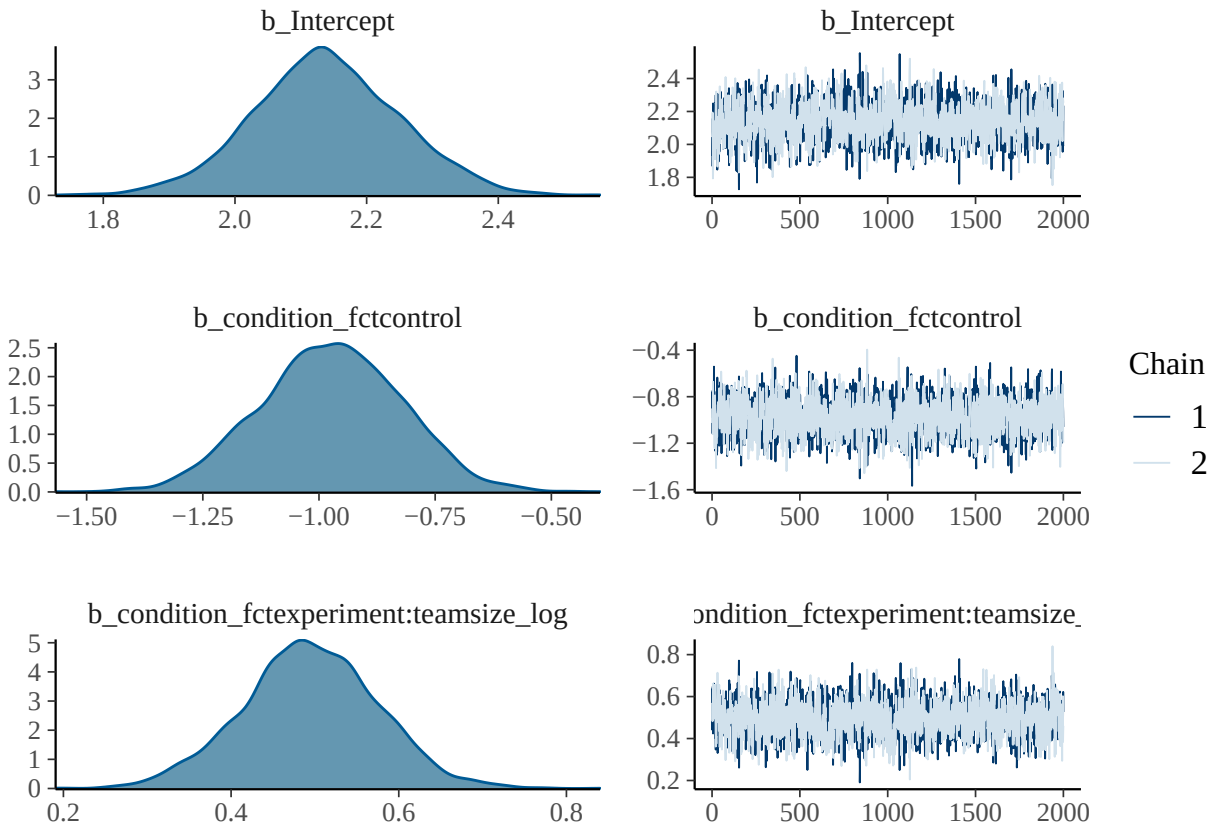


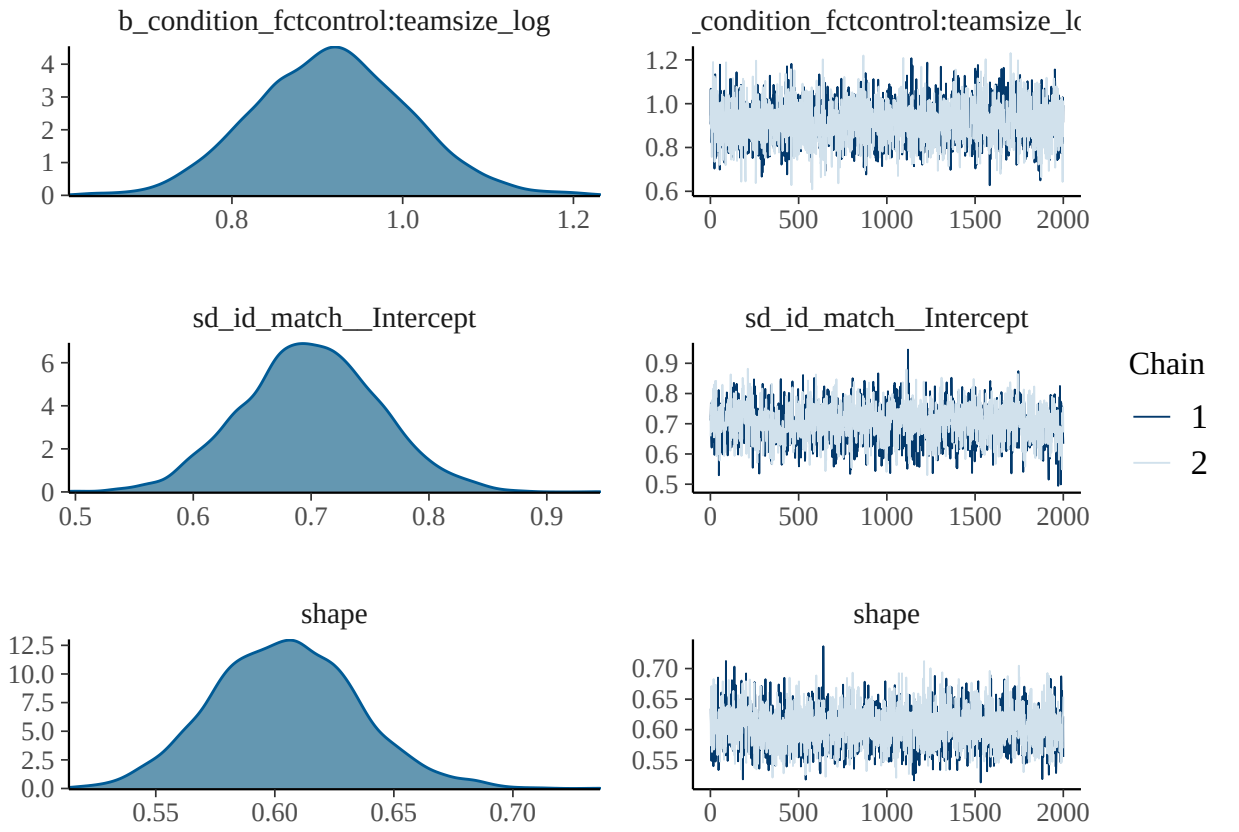

```
# some auto-correlation
# effect almost entirely in random effect
plot(negbin_post_log_1, N = 3)
```


b_Intercept
b_Intercept
b_condition_fctcontrol
b_condition_fctcontrol
b_condition_fctexperiment:teamsize_log
ndition_fctexperiment:teamsize
Chain
1
2




```
# same issues
plot(negbin_post_log_01, N = 3)
```


b_Intercept
b_Intercept
b_condition_fctcontrol
b_condition_fctcontrol
b_condition_fctexperiment:teamsize_log
ondition_fctexperiment:teamsize_
Chain
1
2



check estimates

Fewer effective samples for the log model than non-log (test_interactions2). Again, also fewer samples for the non-intercept model.

```
print(negbin_post_log_0)
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
## Formula: c_5 ~ 0 + condition_fct + condition_fct:teamsize_log + (1 | id_match)
##    Data: d (Number of observations: 1560)
##   Draws: 2 chains, each with iter = 2000; warmup = 0; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~id_match (Number of levels: 780)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.70      0.06     0.59     0.82 1.00     1011     2154
##
## Population-Level Effects:
##                                     Estimate Est.Error l-95% CI u-95% CI Rhat
## condition_fctexperiment                 2.14      0.11     1.92     2.37 1.00
## condition_fctcontrol                    1.13      0.13     0.88     1.39 1.00
## condition_fctexperiment:teamsize_log    0.49      0.08     0.33     0.65 1.00
```

```
## condition_fctcontrol:teamsize_log        0.94      0.09     0.75     1.12 1.00
##                                     Bulk_ESS Tail_ESS
## condition_fctexperiment                 1483     2657
## condition_fctcontrol                    1509     1972
## condition_fctexperiment:teamsize_log    1791     2445
## condition_fctcontrol:teamsize_log       2071     2214
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.61      0.03     0.55     0.67 1.00     1539     2655
##
## Draws were sampled using sample(hmc). 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).
```

```
print(negbin_post_log_1) # by far the highest number of effective samples
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
## Formula: c_5 ~ 1 + condition_fct + condition_fct:teamsize_log + (1 | id_match)
##    Data: d (Number of observations: 1560)
##   Draws: 2 chains, each with iter = 2000; warmup = 0; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~id_match (Number of levels: 780)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.70      0.06     0.59     0.81 1.00     1179     2148
##
## Population-Level Effects:
##                                     Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                               2.16      0.11     1.94     2.38 1.00
## condition_fctcontrol                   -1.00      0.15    -1.29    -0.71 1.00
## condition_fctexperiment:teamsize_log    0.48      0.08     0.32     0.64 1.00
## condition_fctcontrol:teamsize_log       0.92      0.09     0.73     1.10 1.00
##                                     Bulk_ESS Tail_ESS
## Intercept                               3553     3219
## condition_fctcontrol                    4698     3343
## condition_fctexperiment:teamsize_log    4308     3434
## condition_fctcontrol:teamsize_log       4488     3288
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.60      0.03     0.55     0.66 1.00     1828     3355
##
## Draws were sampled using sample(hmc). 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).
```

```
print(negbin_post_log_01)
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
```

```
## Formula: c_5 ~ 0 + Intercept + condition_fct + condition_fct:teamsize_log + (1 | id_match)
##    Data: d (Number of observations: 1560)
##   Draws: 2 chains, each with iter = 2000; warmup = 0; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~id_match (Number of levels: 780)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.70      0.06     0.59     0.81 1.00     1051     1987
##
## Population-Level Effects:
##                                     Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                               2.14      0.11     1.92     2.36 1.00
## condition_fctcontrol                   -0.97      0.15    -1.27    -0.69 1.00
## condition_fctexperiment:teamsize_log    0.49      0.08     0.33     0.65 1.00
## condition_fctcontrol:teamsize_log       0.92      0.09     0.74     1.10 1.00
##                                     Bulk_ESS Tail_ESS
## Intercept                               1789     2534
## condition_fctcontrol                    2505     2685
## condition_fctexperiment:teamsize_log    2084     2386
## condition_fctcontrol:teamsize_log       2668     2335
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.60      0.03     0.55     0.67 1.00     1717     2724
##
## Draws were sampled using sample(hmc). 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).
```

plot implications

```
y <- d$c_5
y_0 <- posterior_predict(negbin_post_log_0, draws = 500)
y_1 <- posterior_predict(negbin_post_log_1, draws = 500)
y_01 <- posterior_predict(negbin_post_log_01, draws = 500)
```

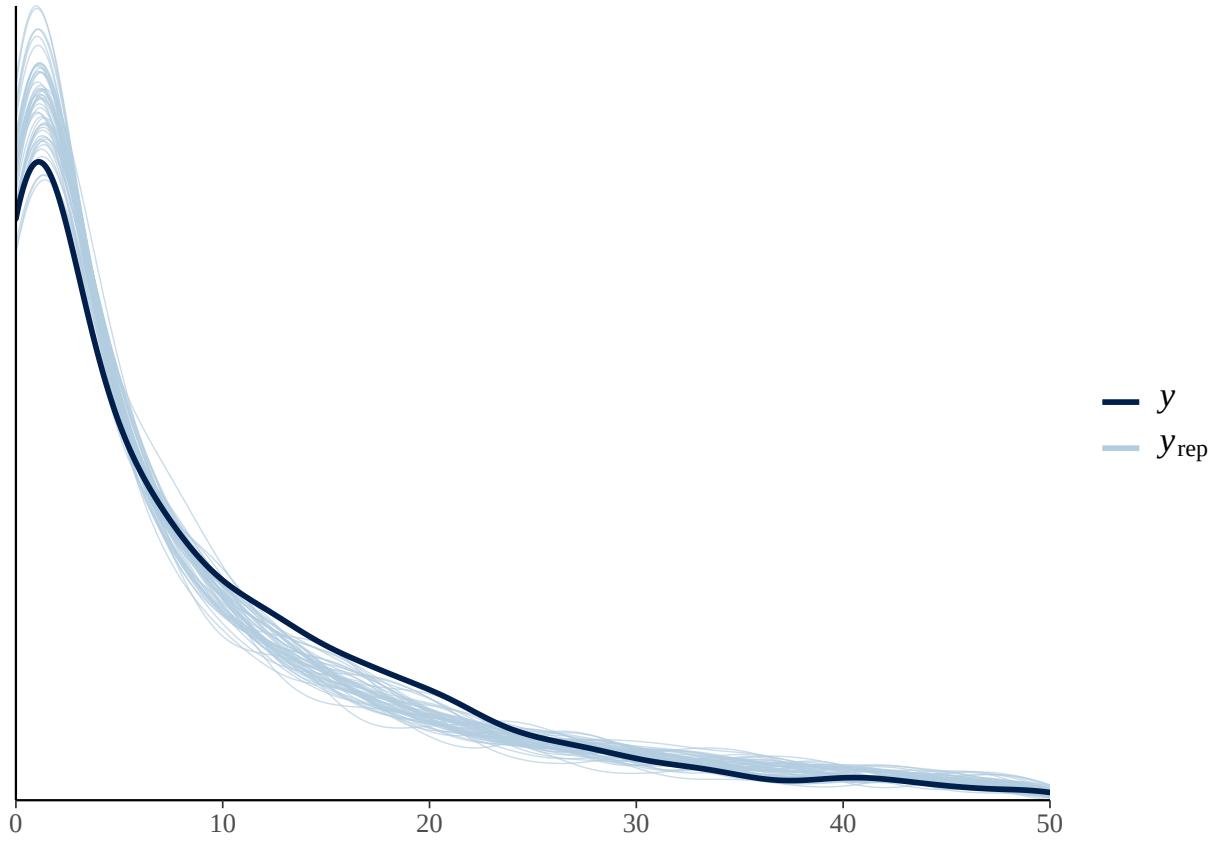
no intercept

looks pretty good. Does overestimate number of 1, and underestimate number of 2 & 3 a bit (always does) – perhaps zero-inflated better? a lot of uncertainty around 0 and 1 still.

```
ppc_dens_overlay(y, y_0[1:50, ]) + xlim(0, 50)
```

```
## Warning: Removed 6496 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 111 rows containing non-finite values (stat_density).
```

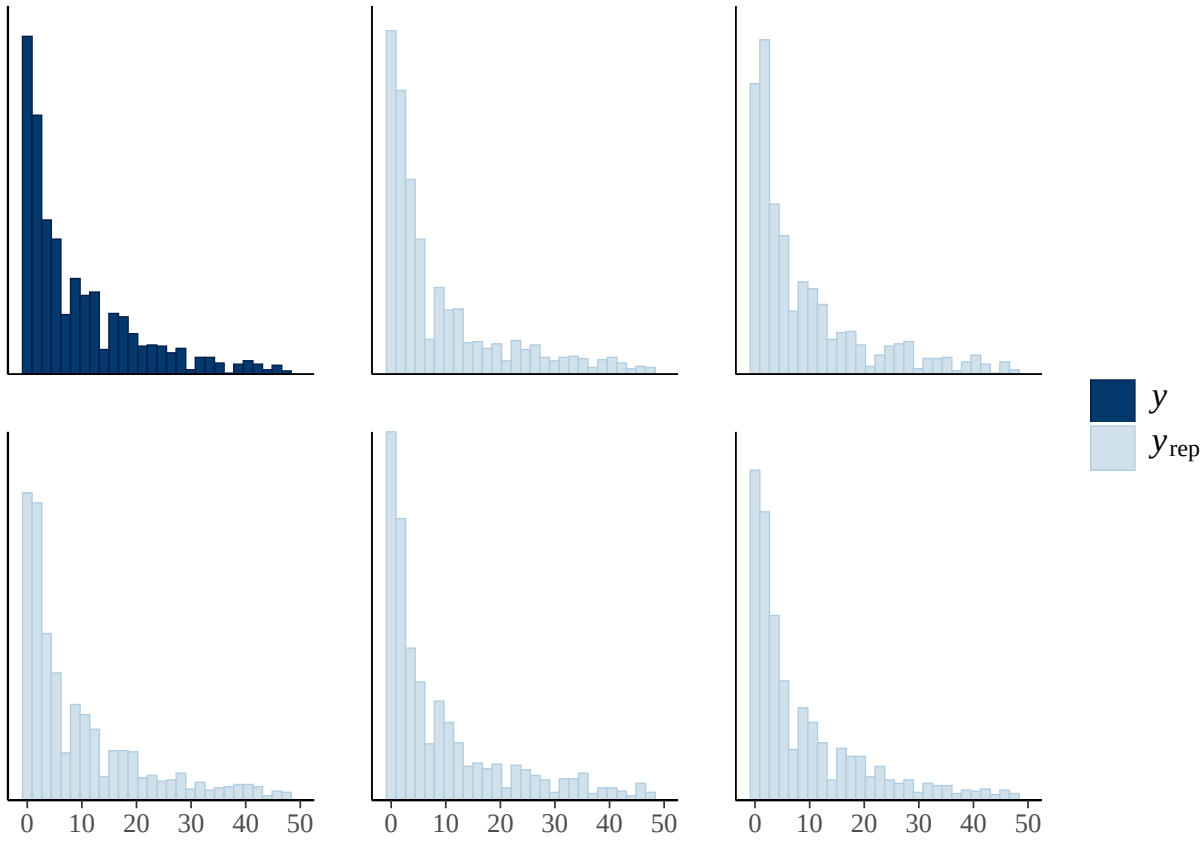



```
ppc_hist(y, y_0[1:5, ]) + xlim(-1, 50)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 749 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 12 rows containing missing values (geom_bar).
```



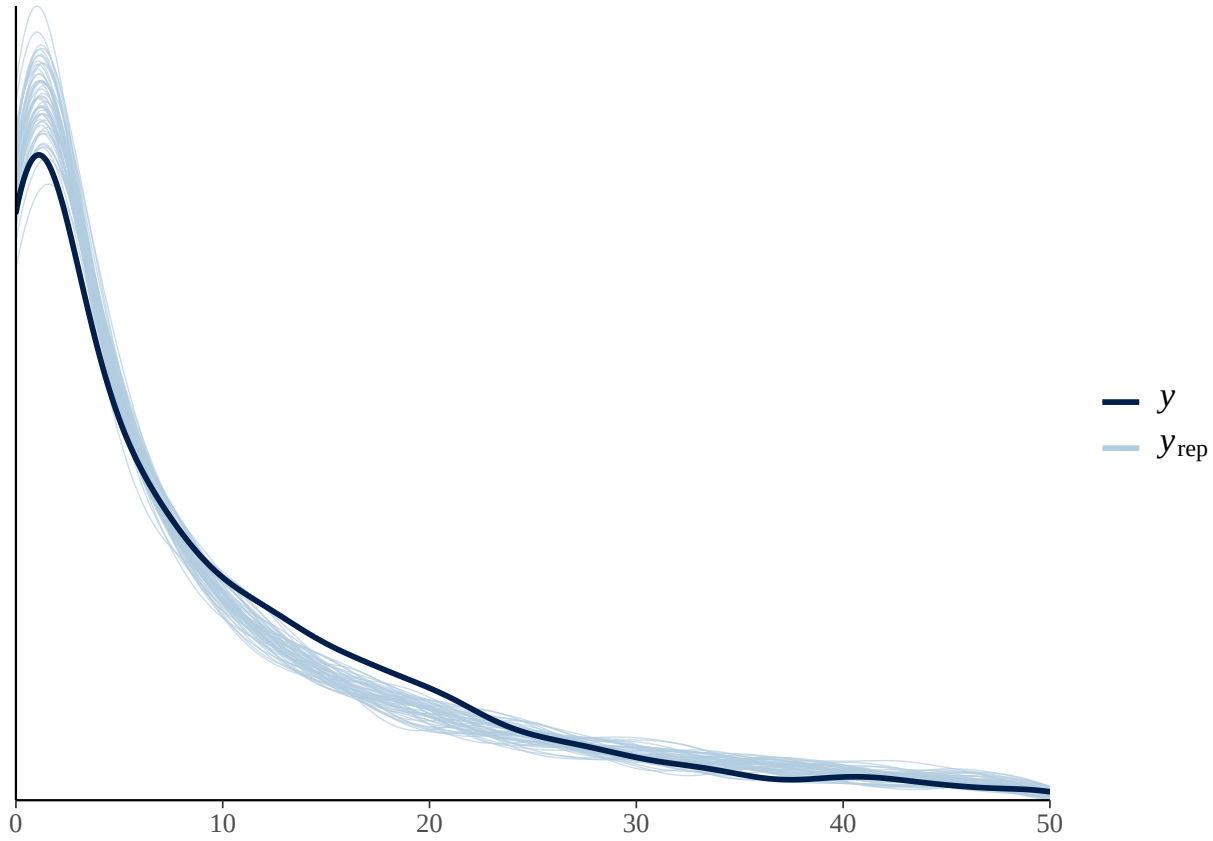
regular intercept

Perhaps a bit worse (e.g. with the undershoot at around c_5 = 5).

```
ppc_dens_overlay(y, y_1[1:50, ]) + xlim(0, 50)
```

```
## Warning: Removed 6319 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 111 rows containing non-finite values (stat_density).
```

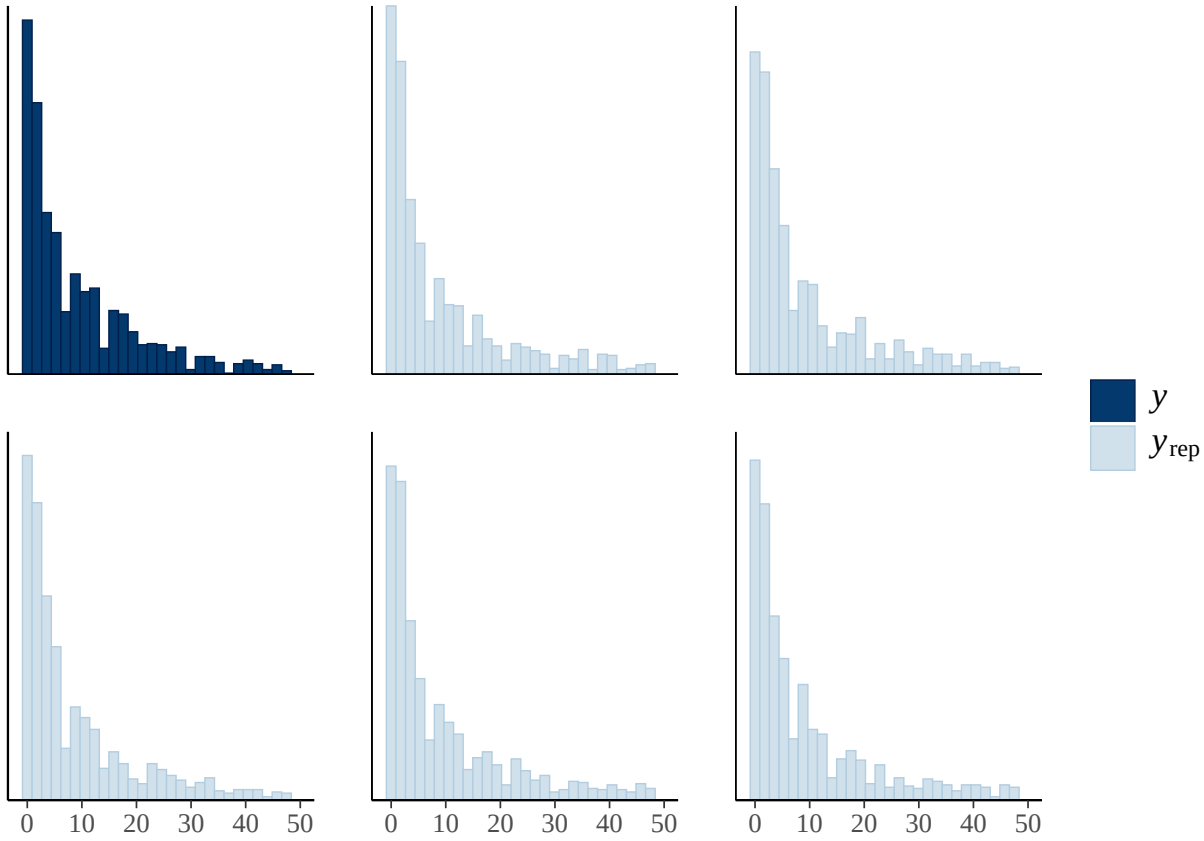



```
ppc_hist(y, y_1[1:5, ]) + xlim(-1, 50)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 730 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 12 rows containing missing values (geom_bar).
```



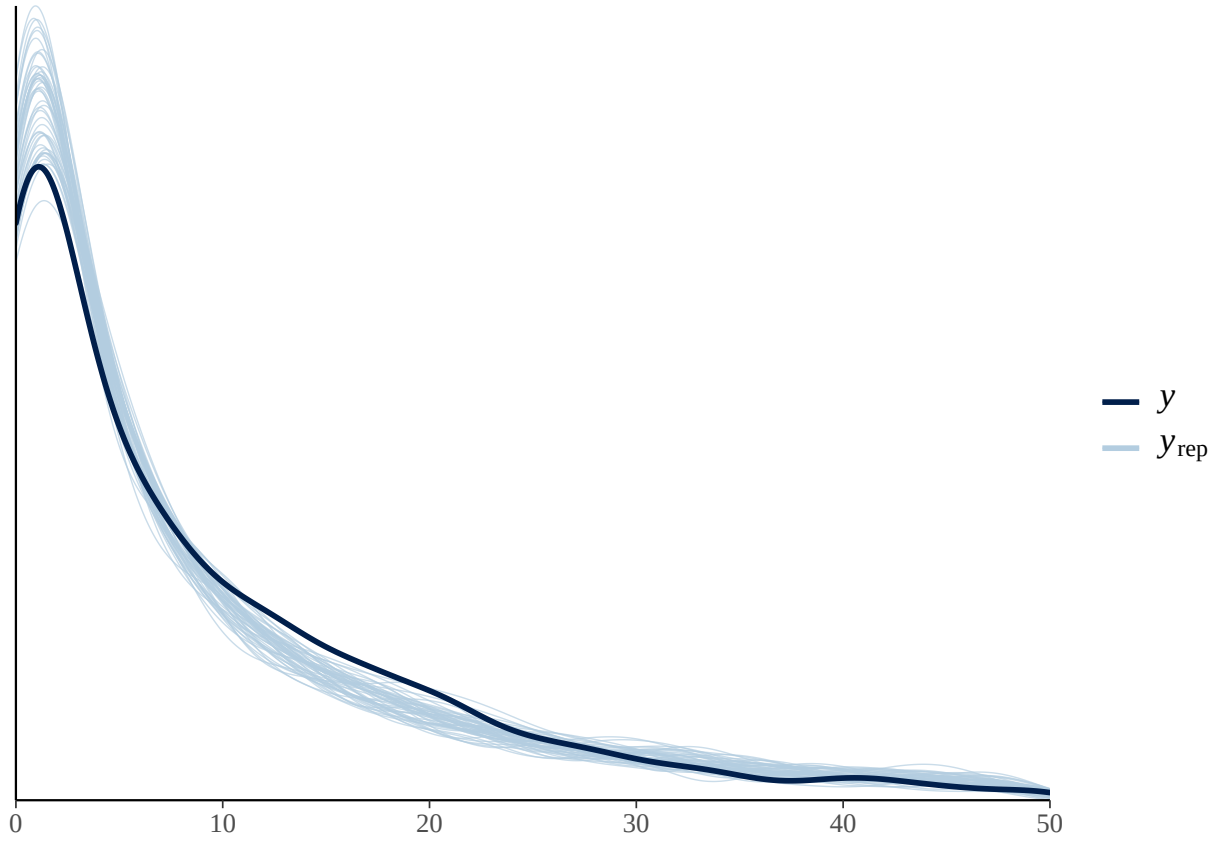
0 + Intercept

Looks more or less the same.

```
ppc_dens_overlay(y, y_01[1:50, ]) + xlim(0, 50)
```

```
## Warning: Removed 6316 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 111 rows containing non-finite values (stat_density).
```

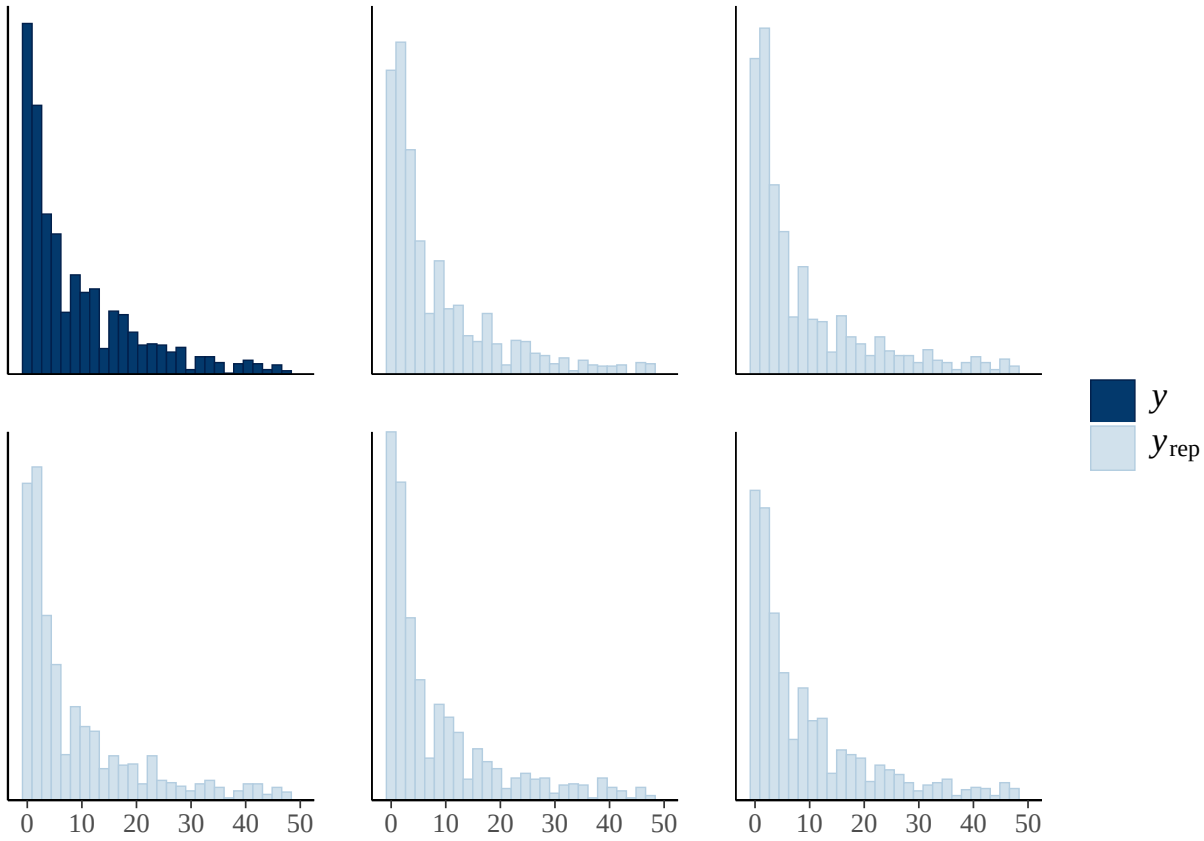



```
ppc_hist(y, y_01[1:5, ]) + xlim(-1, 50)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 761 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 12 rows containing missing values (geom_bar).
```



pareto k issues

https://bookdown.org/content/4857/monsters-and-mixtures.html

no intercept

Mainly studies that are (relatively) low teamsize and high citation. Also, several control studies that are the same (c_5 = 77, teamsize_log = 0.693, condition = control). Really need to fix this in the preprocessing. But here, we basically get that all studies with many citations are influential...

```
d %>%
  mutate(k = negbin_post_log_0$criteria$loo$diagnostics$pareto_k) %>%
  filter(k > .7) %>%
  select(c_5, teamsize_log, condition_fct, id_match, k)
```

```
## # A tibble: 68 x 5
##      c_5 teamsize_log condition_fct id_match     k
##    <dbl>        <dbl> <fct>         <fct>    <dbl>
##  1   310        1.39  experiment    3        1.00
##  2    41        1.61  control       8        0.704
##  3   150        1.79  experiment    15       0.819
##  4     6        0     control       30       0.732
##  5    69        1.39  control       46       0.815
```

```
##  6    72        1.61  experiment    47       0.711
##  7    53        0.693 experiment    59       0.738
##  8   236        1.39  experiment    66       0.938
##  9    70        2.20  control       71       0.903
## 10    30        1.10  experiment    77       0.702
## # ... with 58 more rows
```

1 + ...

some of the same here, mainly studies with high c_5 and low teamsize.

```
d %>%
  mutate(k = negbin_post_log_1$criteria$loo$diagnostics$pareto_k) %>%
  filter(k > .7) %>%
  select(c_5, teamsize_log, condition_fct, id_match, k)
```

```
## # A tibble: 71 x 5
##      c_5 teamsize_log condition_fct id_match     k
##    <dbl>        <dbl> <fct>         <fct>    <dbl>
##  1   310        1.39  experiment    3        0.746
##  2    33        0.693 control       35       0.873
##  3    99        2.40  experiment    44       0.766
##  4    69        1.39  control       46       0.821
##  5   236        1.39  experiment    66       0.969
##  6    70        2.20  control       71       0.779
##  7    13        0     control       73       0.701
##  8    77        1.39  experiment    74       0.767
##  9   995        2.20  experiment    85       1.25
## 10    54        0     control       95       0.757
## # ... with 61 more rows
```

0 + Intercept

Seems to handle influential observations a bit better. Still issue with the repeated value as above & some "experiment"-condition. Primarily those with high c_5.

```
# two studies that are the same in control (issue to be resolved earlier in the pipeline).
# the outlier study (experiment) which is max in teamsize and also extremely high citation
# whereas the
d %>%
  mutate(k = negbin_post_log_01$criteria$loo$diagnostics$pareto_k) %>%
  filter(k > .7) %>%
  select(c_5, teamsize_log, condition_fct, id_match, k)
```

```
## # A tibble: 71 x 5
##      c_5 teamsize_log condition_fct id_match     k
##    <dbl>        <dbl> <fct>         <fct>    <dbl>
##  1   310        1.39  experiment    3        1.18
##  2    58        1.39  experiment    13       0.752
##  3    33        0.693 control       35       0.760
##  4    84        1.39  control       42       0.780
```

```
##  5    51        1.79  experiment    43       0.745
##  6   169        1.61  experiment    45       0.783
##  7    53        0.693 experiment    59       0.965
##  8   236        1.39  experiment    66       0.847
##  9   995        2.20  experiment    85       0.873
## 10    54        0     control       95       1.00
## # ... with 61 more rows
```

Quick model comparison

Basically no difference, but appears to prefer the intercept models. Do we know why that is?

```
loo_compare(negbin_post_log_0,
            negbin_post_log_1,
            negbin_post_log_01)
```

```
##                    elpd_diff se_diff
## negbin_post_log_0   0.0       0.0
## negbin_post_log_01 -1.7       2.0
## negbin_post_log_1  -2.0       1.7
```

```
loo_model_weights(negbin_post_log_0,
                  negbin_post_log_1,
                  negbin_post_log_01)
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.

## Method: stacking
## ------
##                    weight
## negbin_post_log_0  0.879
## negbin_post_log_1  0.000
## negbin_post_log_01 0.121
```