Test Interaction

Victor Møller Poulsen

```r
# check 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),
         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: 12
## $ 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~
## $ id_fct               <fct> 2330249536, 2003350634, 2385753682, 2395494269,~
```

Different ways of specifying something similar. We had issues with model (f_team_0) earlier. Trying to troubleshoot whether it is related to intercept & I think that (0 + Intercept) syntax is actually more appropriate since it does not assume mean centering (something like that).

```r
f_team_0 <- bf(c_5 ~ 0 + condition_fct + condition_fct:teamsize_scaled + (1|id_fct))
f_team_1 <- bf(c_5 ~ 1 + condition_fct + condition_fct:teamsize_scaled + (1|id_fct))
f_team_01 <- bf(c_5 ~ 0 + Intercept + condition_fct + condition_fct:teamsize_scaled + (1|id_fct))
```

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

```r
# 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

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

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

}
```

```r
prior_check(negbin_prior_0, 100, "No Intercept (x cutoff: 25)", 25)
```

```
## Warning: Removed 169 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 253 rows containing non-finite values (stat_density).
```
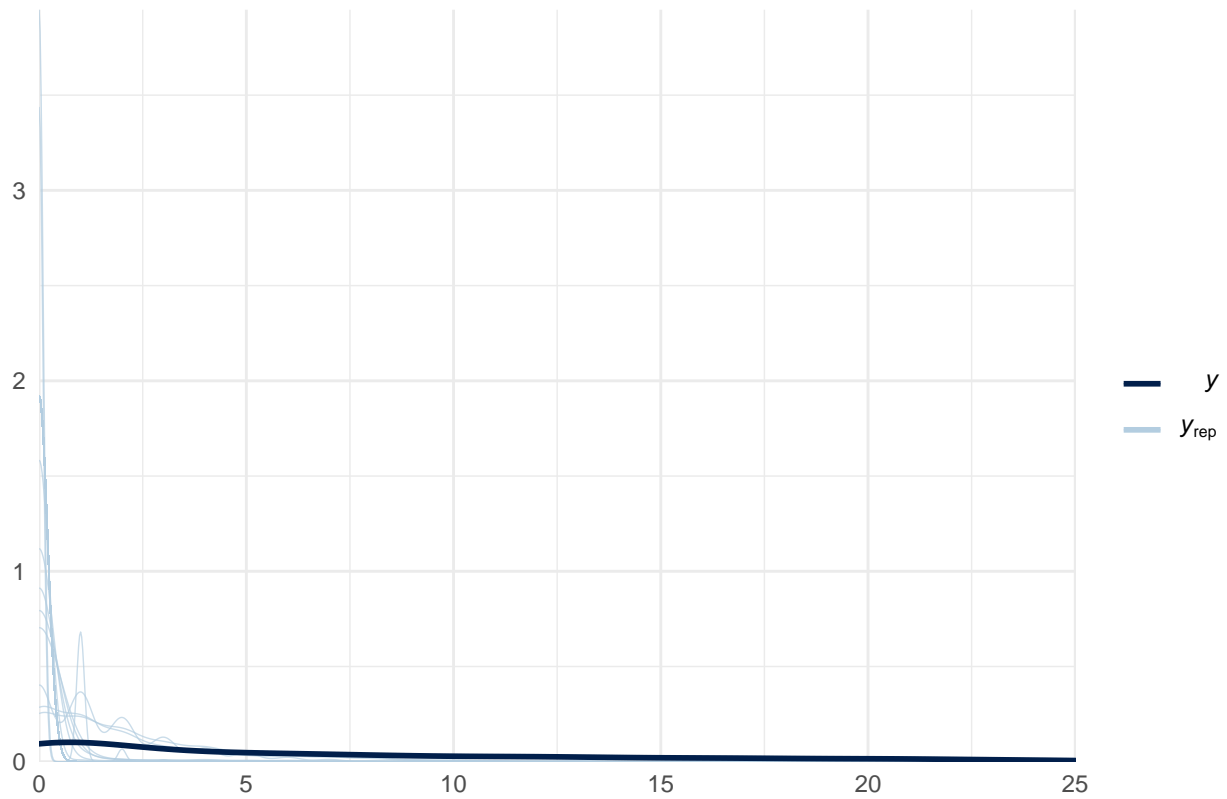
No Intercept (x cutoff: 25)


```
prior_check(negbin_prior_1, 100, "Intercept (x cutoff: 25)", 25)
```
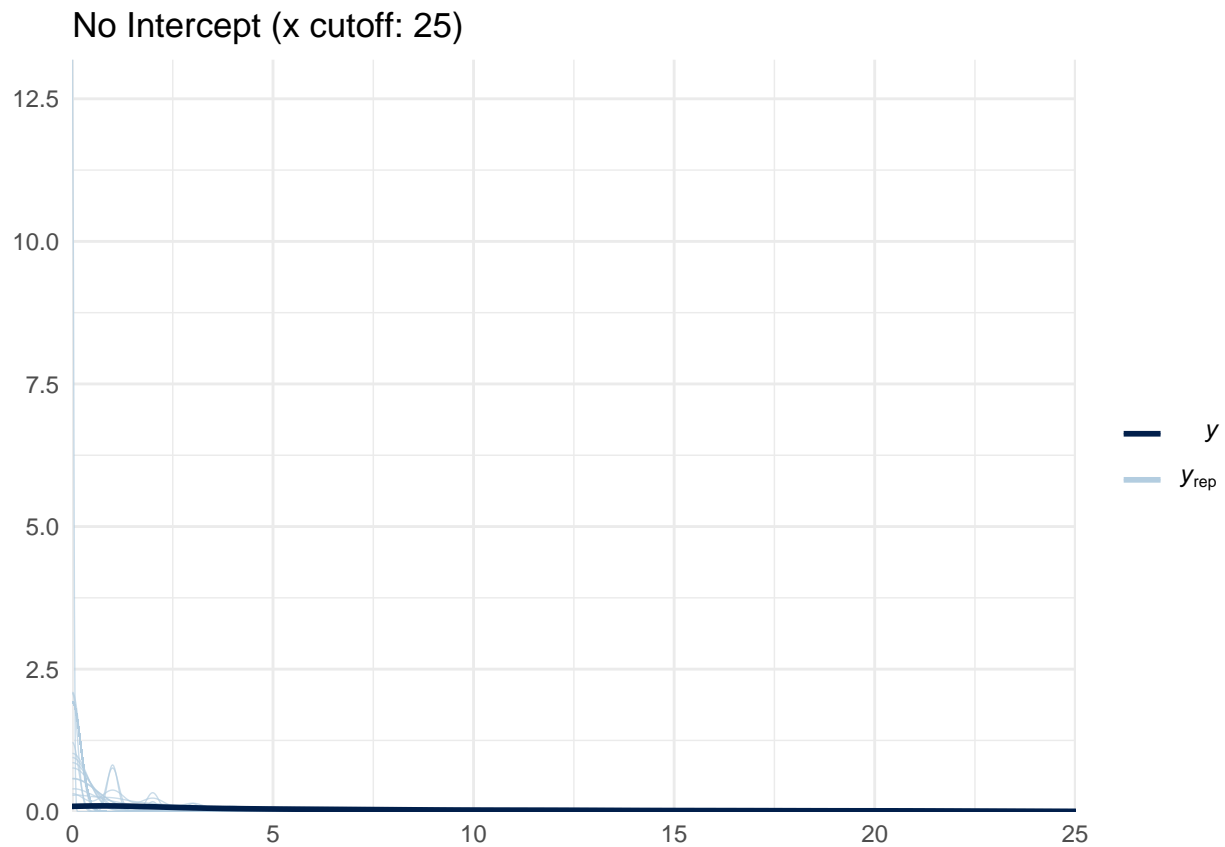
```
## Warning: Removed 130 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 253 rows containing non-finite values (stat_density).
```
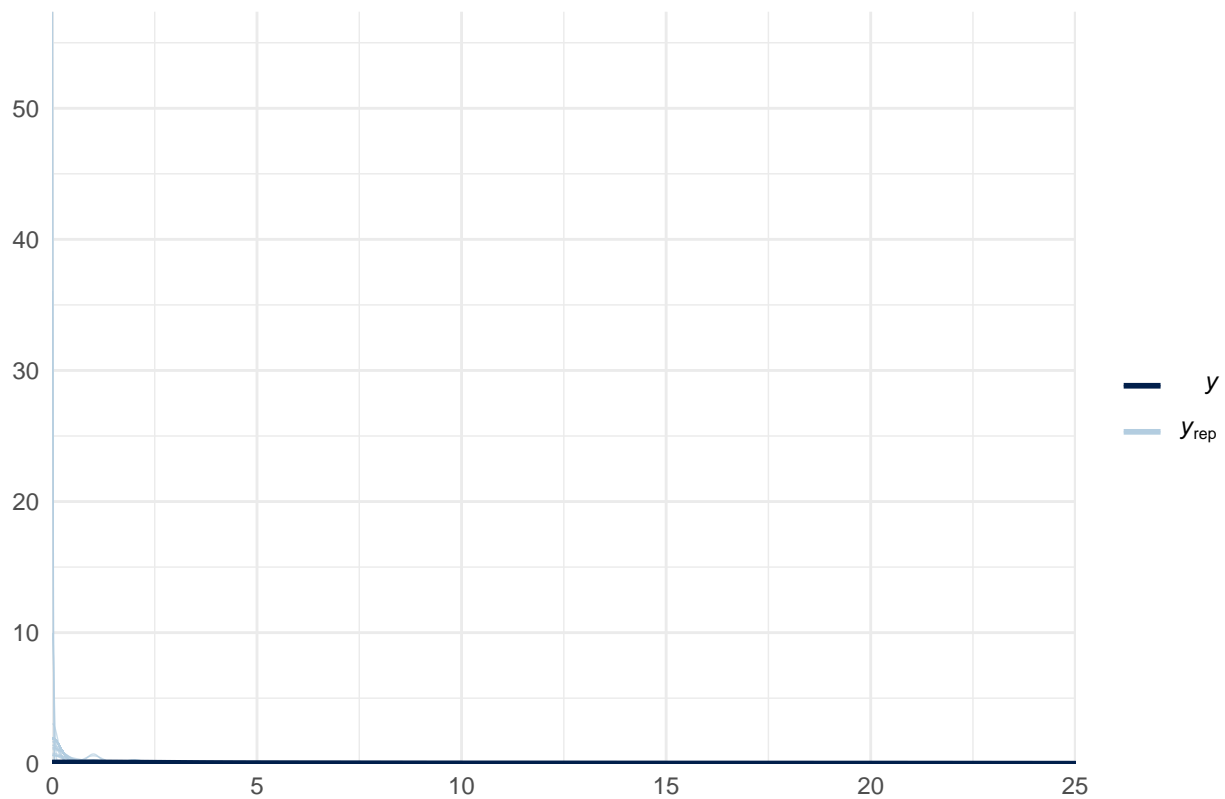
Intercept (x cutoff: 25)


```
prior_check(negbin_prior_01, 100, "0 + Intercept (x cutoff: 25)", 25)
```

Warning: Removed 370 rows containing non-finite values (stat_density).

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

0 + Intercept (x cutoff: 25)

fit models

```r
## baseline
negbin_post_0 <- fit_model(
  family = negbinomial(),
  formula = f_team_0,
  prior = negbin_0,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_0"
)

## baseline
negbin_post_1 <- fit_model(
  family = negbinomial(),
  formula = f_team_1,
  prior = negbin_1,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_1"
)

## baseline
negbin_post_01 <- fit_model(
  family = negbinomial(),
```

```
    formula = f_team_01,
    prior = negbin_01,
    sample_prior = TRUE,
    file = "/work/50114/MAG/modeling/models/negbin_post_01"
)
```
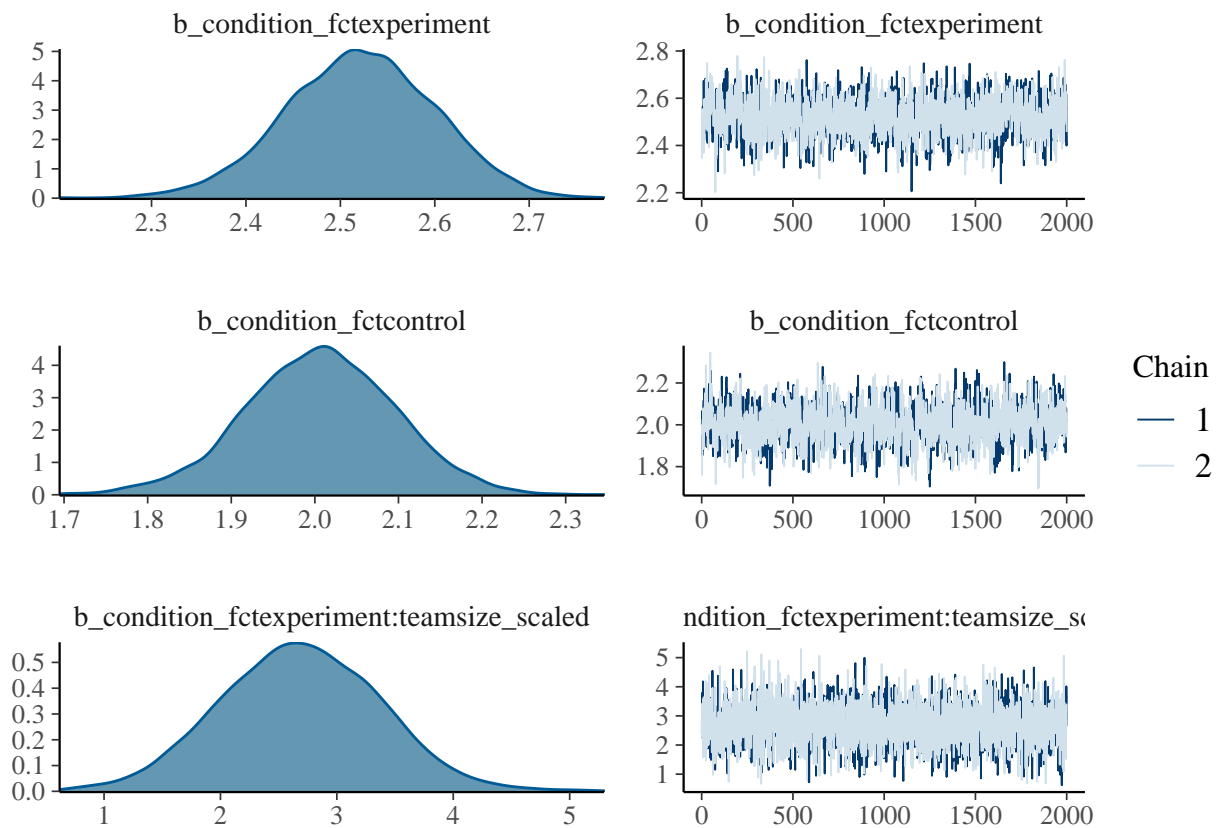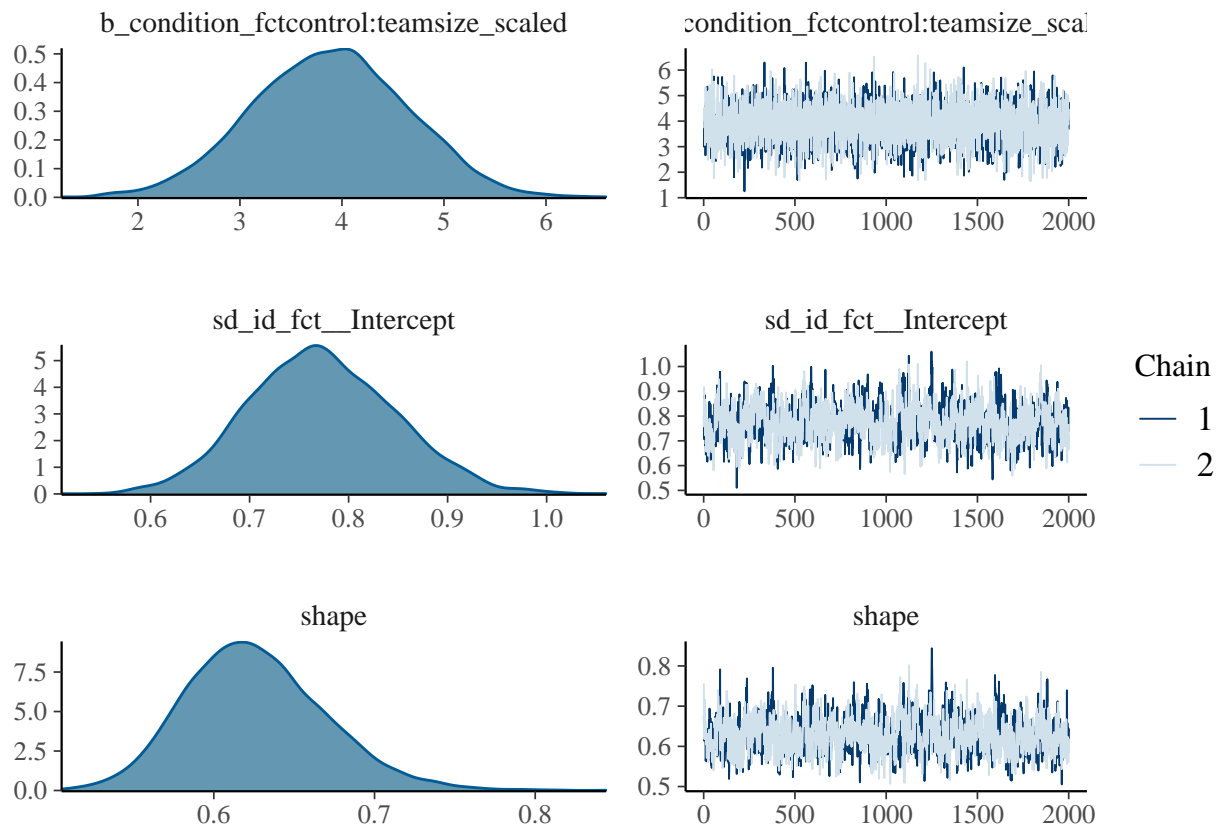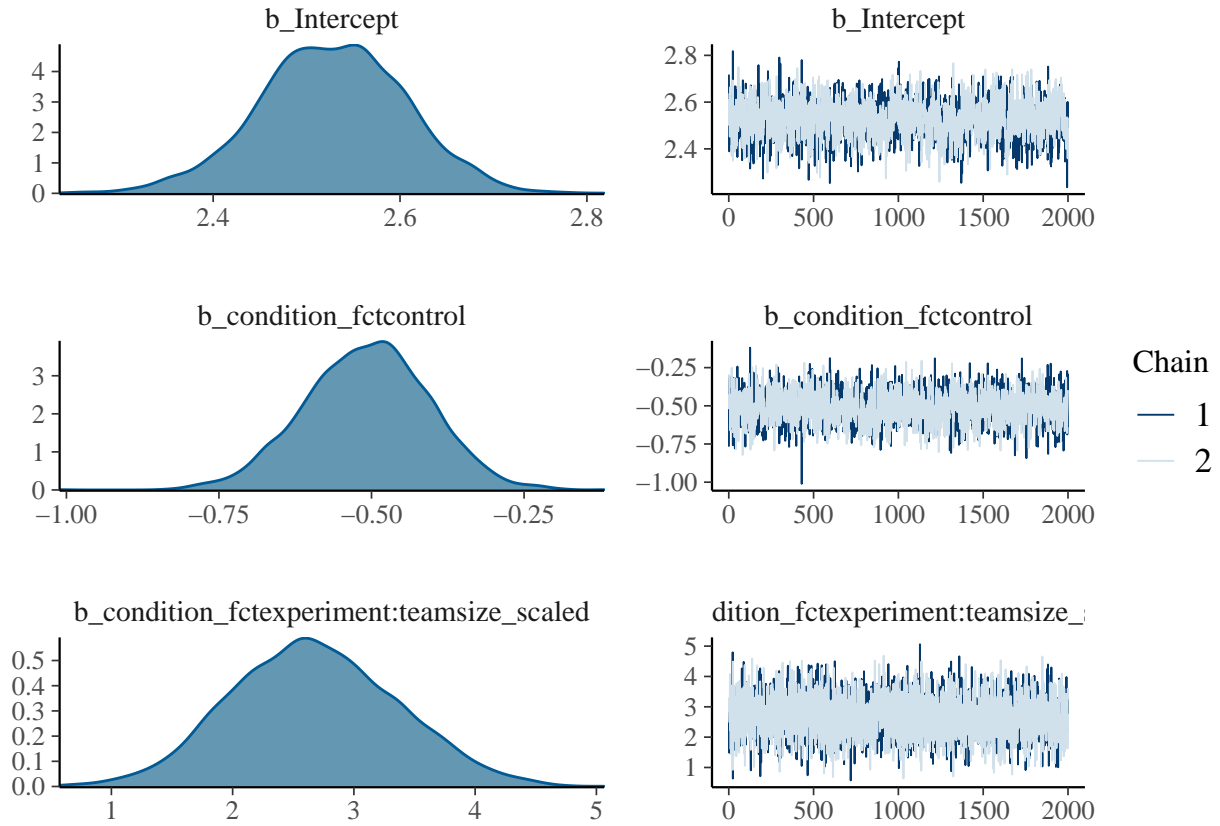
check traces

```
# some auto-correlation
# effect almost entirely in random effect
plot(negbin_post_0, N = 3)
```
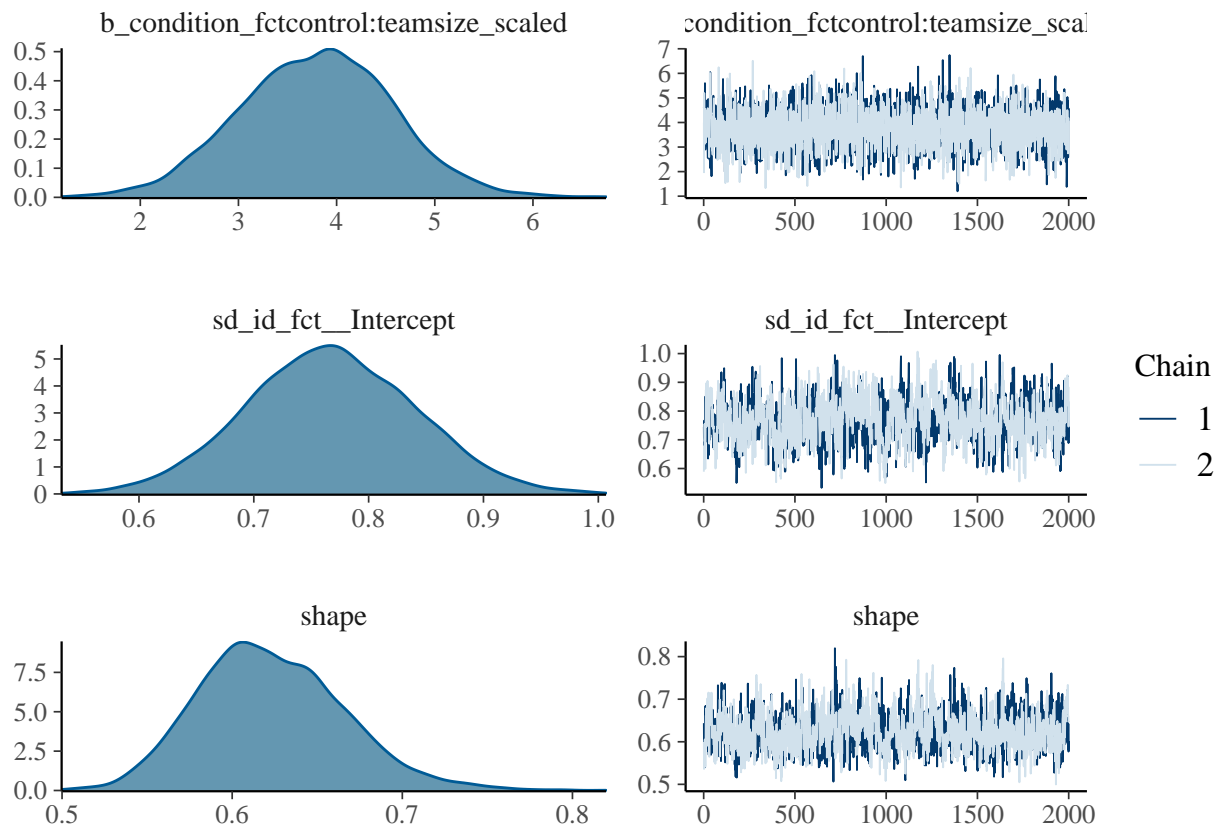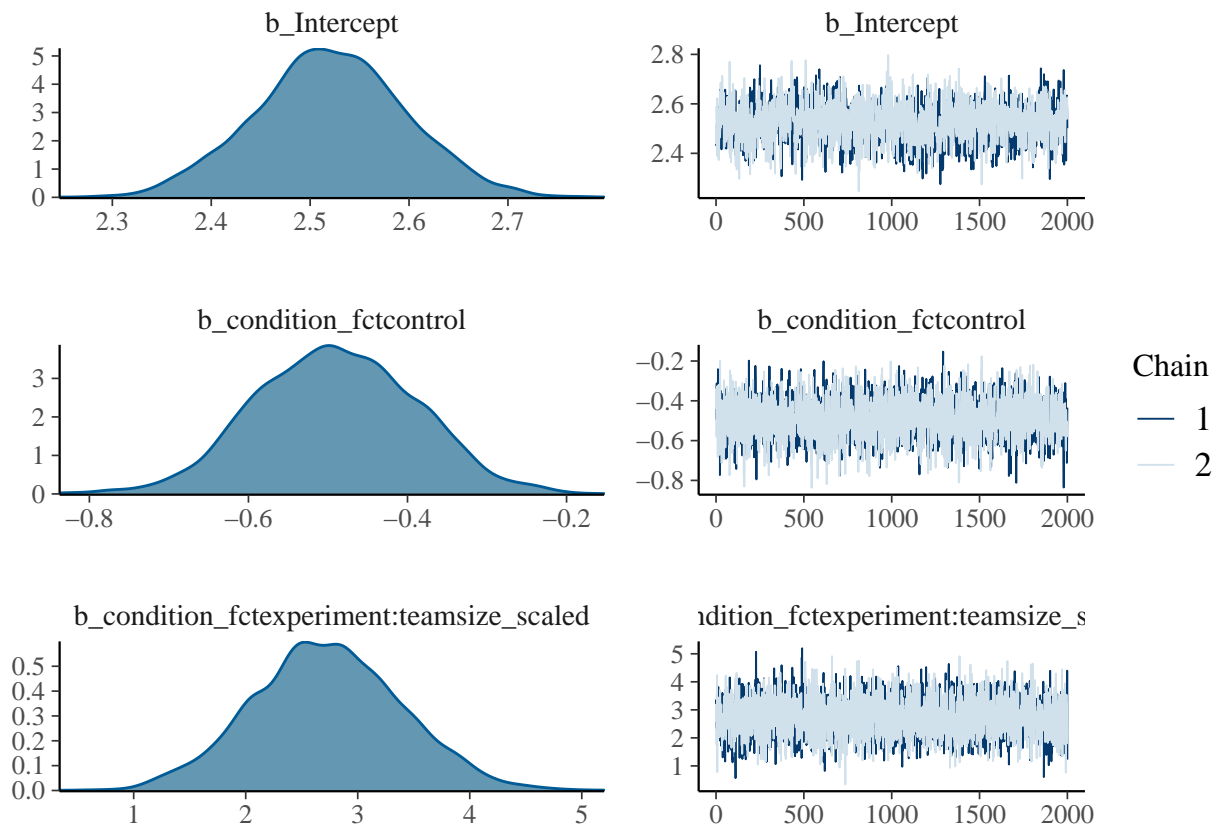
```
# some auto-correlation
# effect almost entirely in random effect
plot(negbin_post_1, N = 3)
```
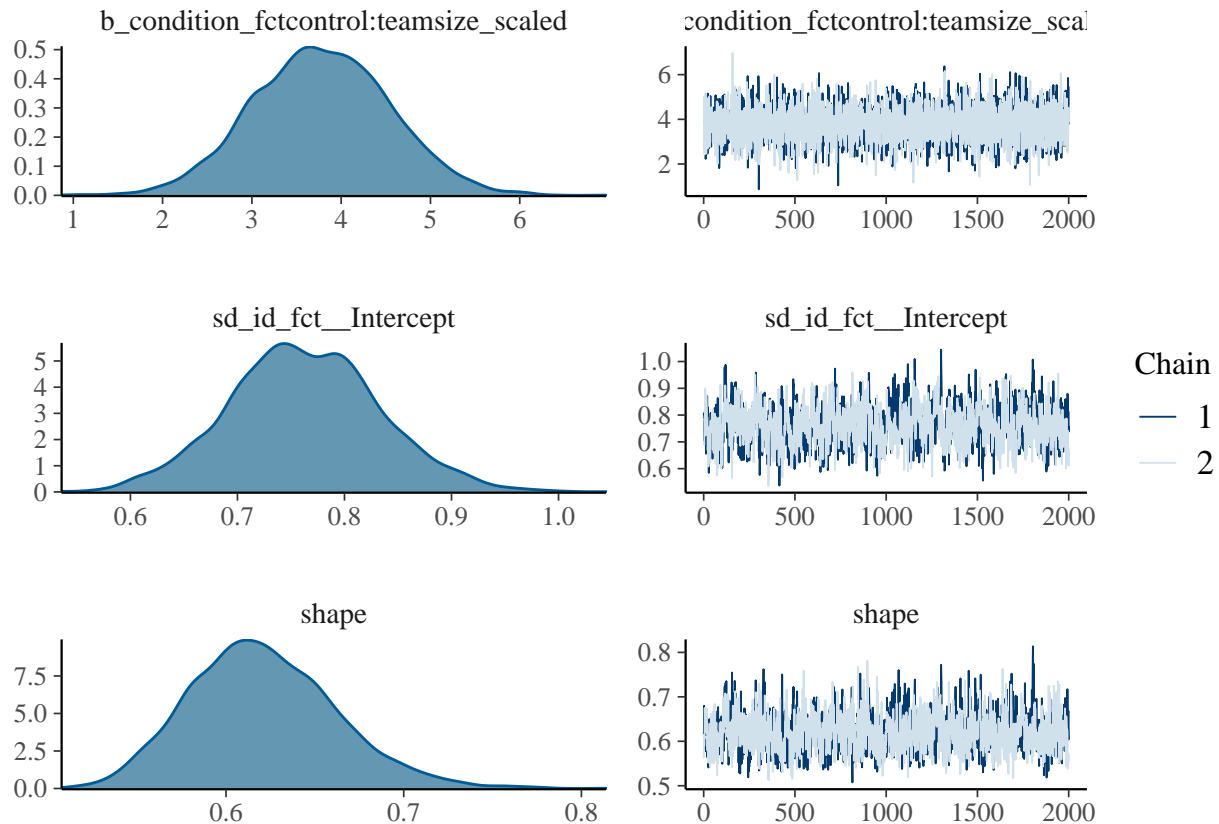
```r
# same issues
plot(negbin_post_01, N = 3)
```

Seems like the issue is not the particular specification: This post explains why it is problematic to have random effect for single observation: https://stats.stackexchange.com/questions/242821/how-will-random-effects-with-only-1-observation-affect-a-generalized-linear-mixe

Try with random effect per group (i.e. experiment + control)

create the variable (as factor)

```
d <- d %>%
  mutate(id_match = as_factor(match_group))
```

new model formulae

```
f_team_match_0 <- bf(c_5 ~ 0 + condition_fct + condition_fct:teamsize_scaled + (1|id_match))
f_team_match_1 <- bf(c_5 ~ 1 + condition_fct + condition_fct:teamsize_scaled + (1|id_match))
f_team_match_01 <- bf(c_5 ~ 0 + Intercept + condition_fct + condition_fct:teamsize_scaled + (1|id_match)
```

prior

```
## baseline
negbin_prior_match_0 <- fit_model(
  family = negbinomial(),
  formula = f_team_match_0,
  prior = negbin_0, # same prior
  sample_prior = "only",
  file = "/work/50114/MAG/modeling/models/negbin_prior_match_0"
)

## baseline
negbin_prior_match_1 <- fit_model(
  family = negbinomial(),
  formula = f_team_match_1,
  prior = negbin_1,
  sample_prior = "only",
  file = "/work/50114/MAG/modeling/models/negbin_prior_match_1"
)

## baseline
negbin_prior_match_01 <- fit_model(
  family = negbinomial(),
  formula = f_team_match_01,
  prior = negbin_01,
  sample_prior = "only",
  file = "/work/50114/MAG/modeling/models/negbin_prior_match_01"
)
```
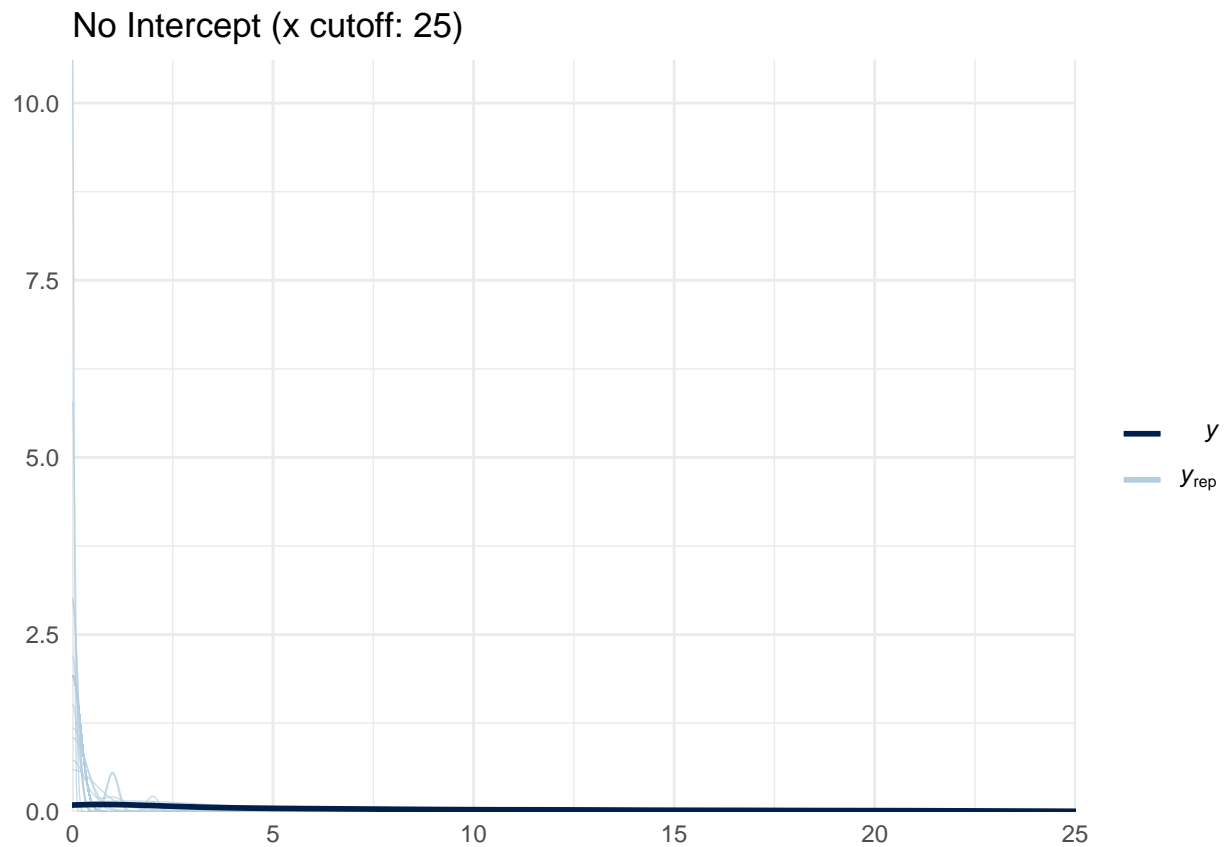
prior check

A few divergent transitions (for all of them, between 7-9/4000)

```
prior_check(negbin_prior_0, 100, "No Intercept (x cutoff: 25)", 25)
```

```
## Warning: Removed 196 rows containing non-finite values (stat_density).
```

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

No Intercept (x cutoff: 25)


```
prior_check(negbin_prior_1, 100, "Intercept (x cutoff: 25)", 25)
```

```
## Warning: Removed 100 rows containing non-finite values (stat_density).
```

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

Intercept (x cutoff: 25)


```
prior_check(negbin_prior_01, 100, "0 + Intercept (x cutoff: 25)", 25)
```

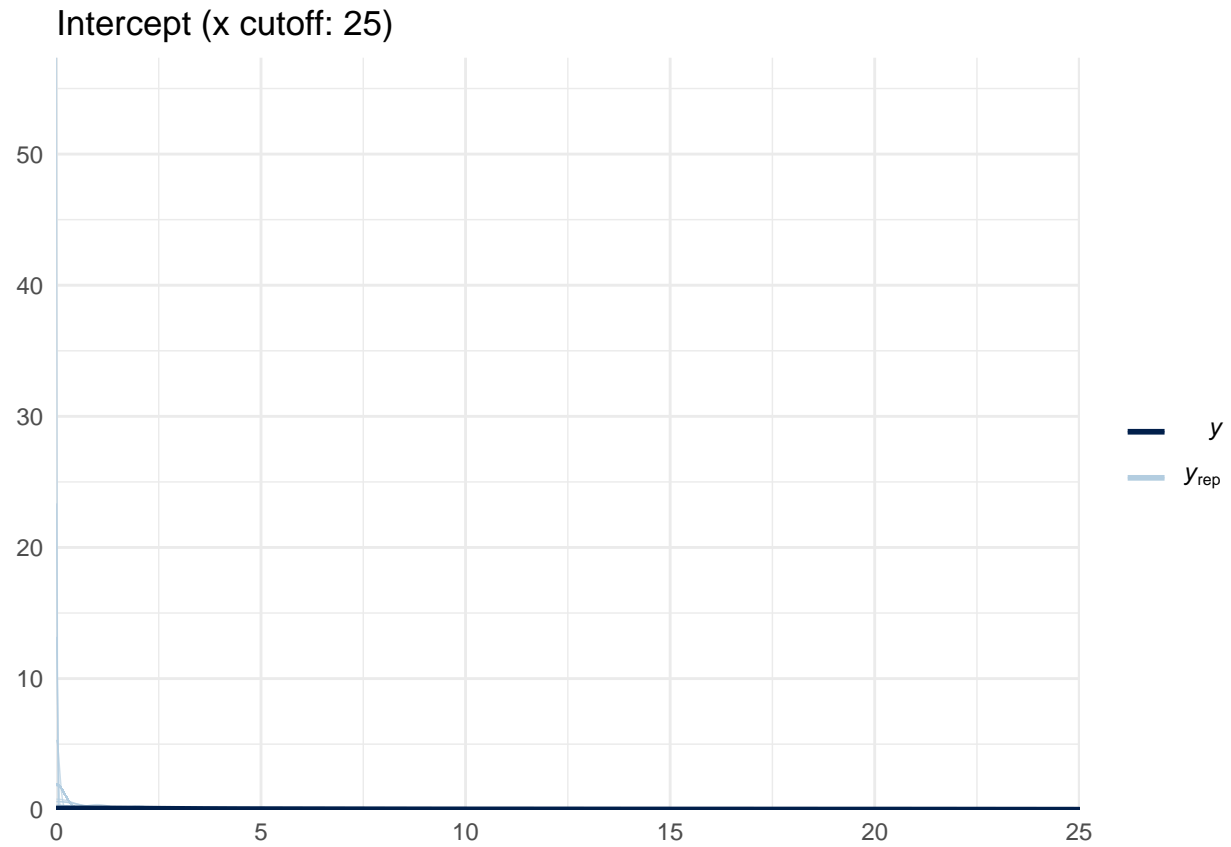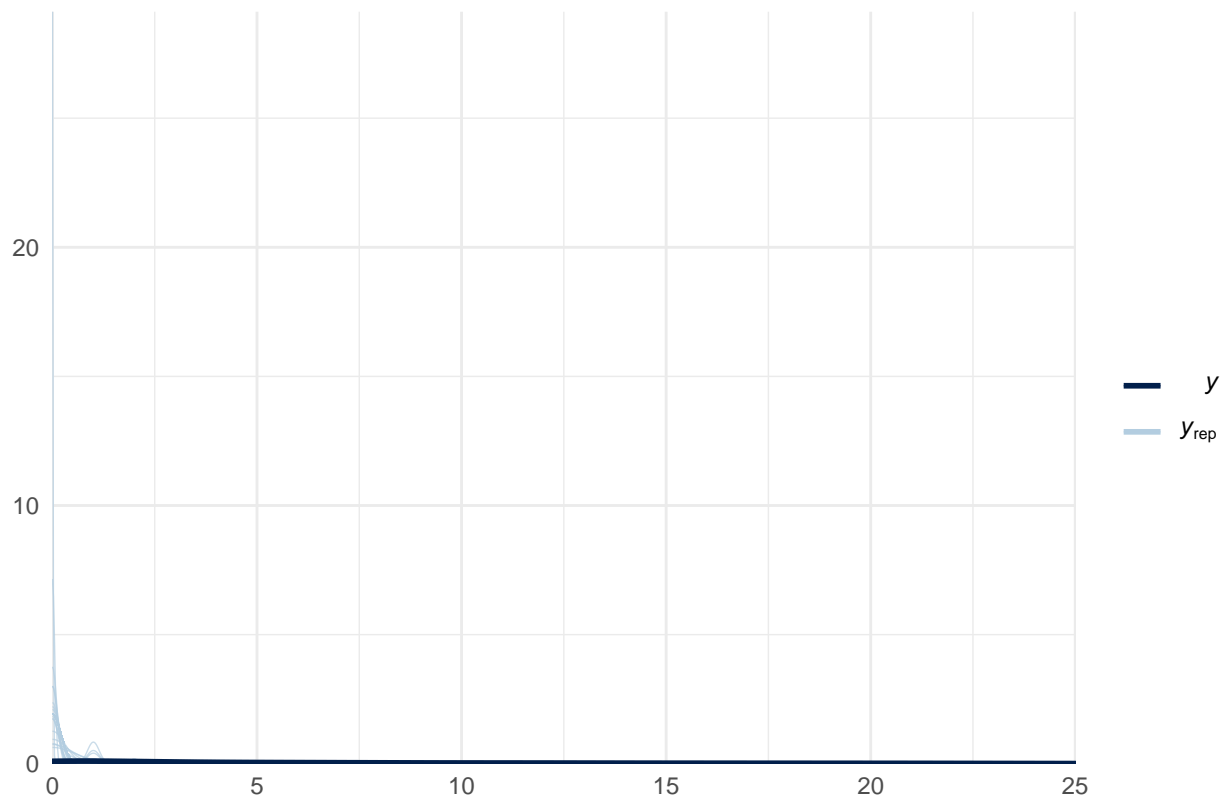## Warning: Removed 185 rows containing non-finite values (stat_density).

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

0 + Intercept (x cutoff: 25)

fit models

a few pareto_k > 0.7 (but few compared to above).

```
## baseline
negbin_post_match_0 <- fit_model(
  family = negbinomial(),
  formula = f_team_match_0,
  prior = negbin_0,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_match_0"
)

## baseline
negbin_post_match_1 <- fit_model(
  family = negbinomial(),
  formula = f_team_match_1,
  prior = negbin_1,
  sample_prior = TRUE,
  file = "/work/50114/MAG/modeling/models/negbin_post_match_1"
)

## baseline
negbin_post_match_01 <- fit_model(
```

```
    family = negbinomial(),
    formula = f_team_match_01,
    prior = negbin_01,
    sample_prior = TRUE,
    file = "/work/50114/MAG/modeling/models/negbin_post_match_01"
)
```
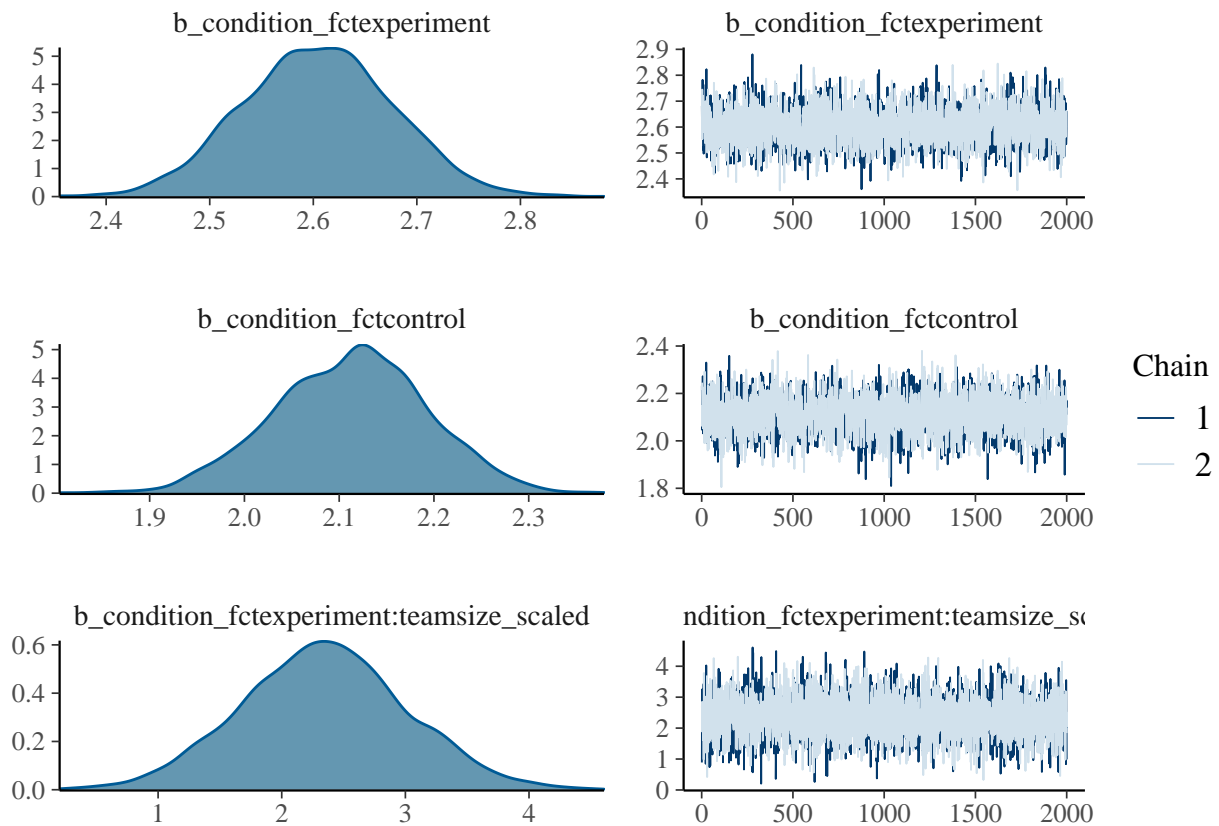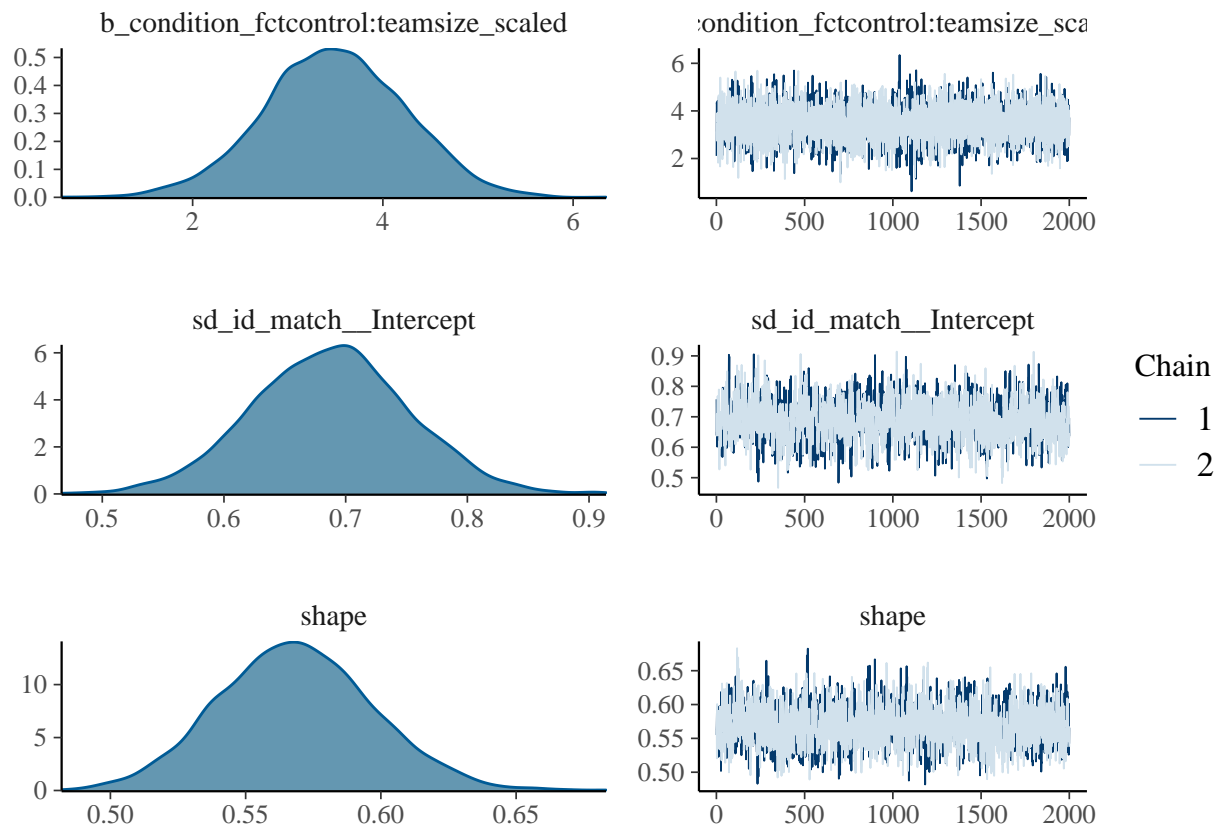
traces

Looks MUCH better now than before.

```
# looks pretty good.
plot(negbin_post_match_0, N = 3)
```
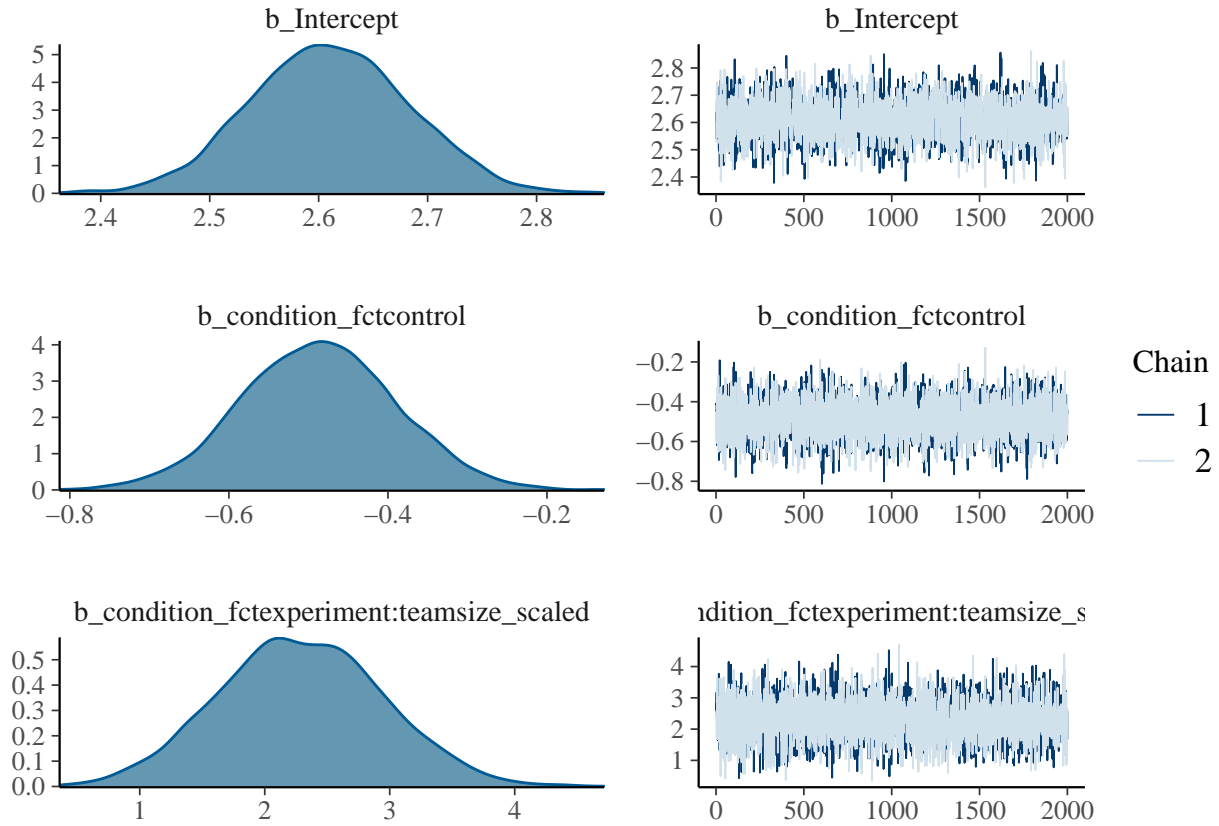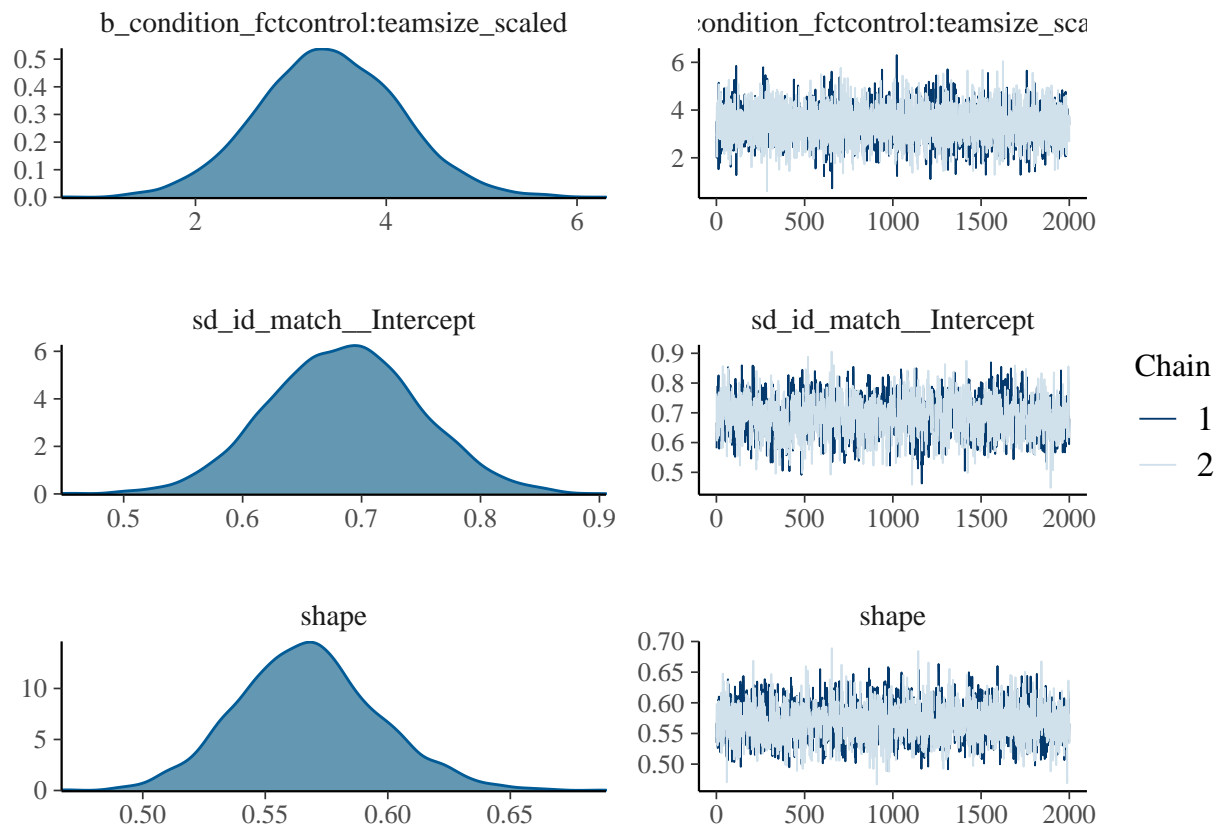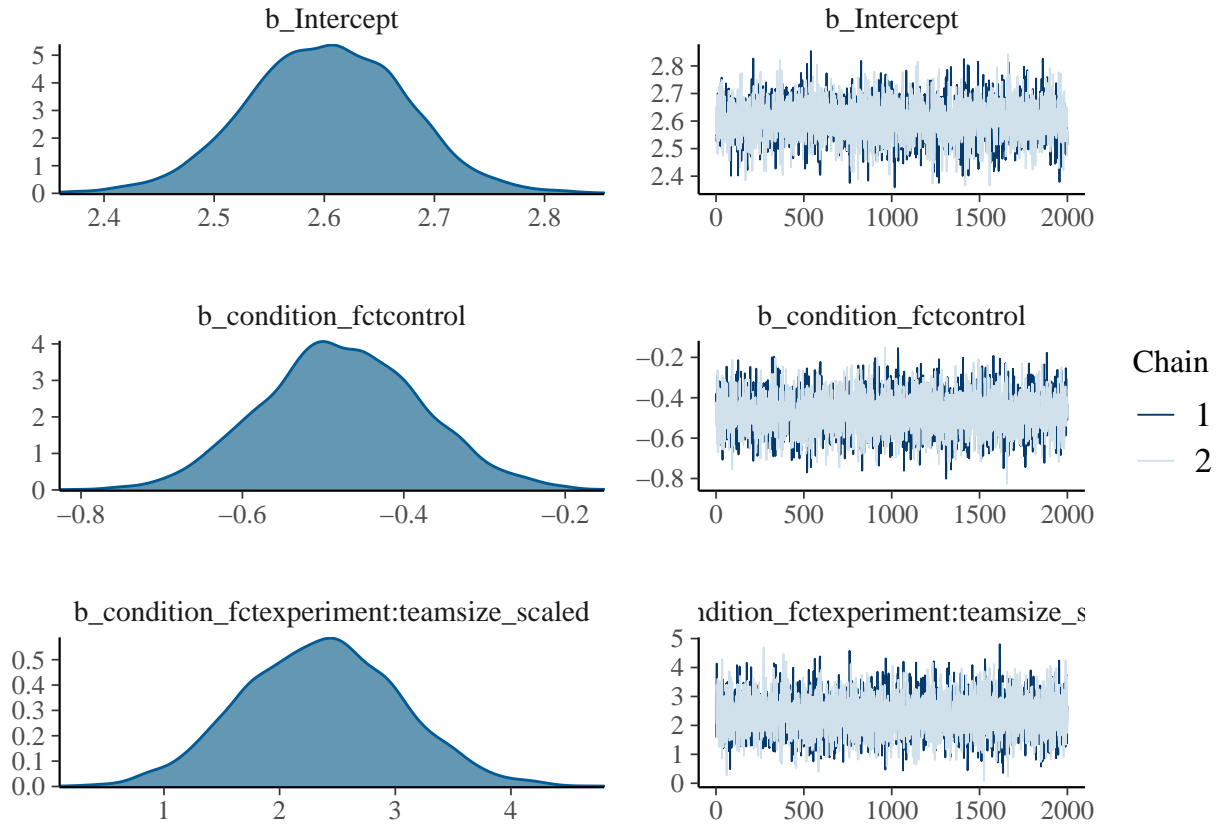
```
# same as above
plot(negbin_post_match_1, N = 3)
```

```
# slightly different: might be more appropriate?
plot(negbin_post_match_01, N = 3)
```
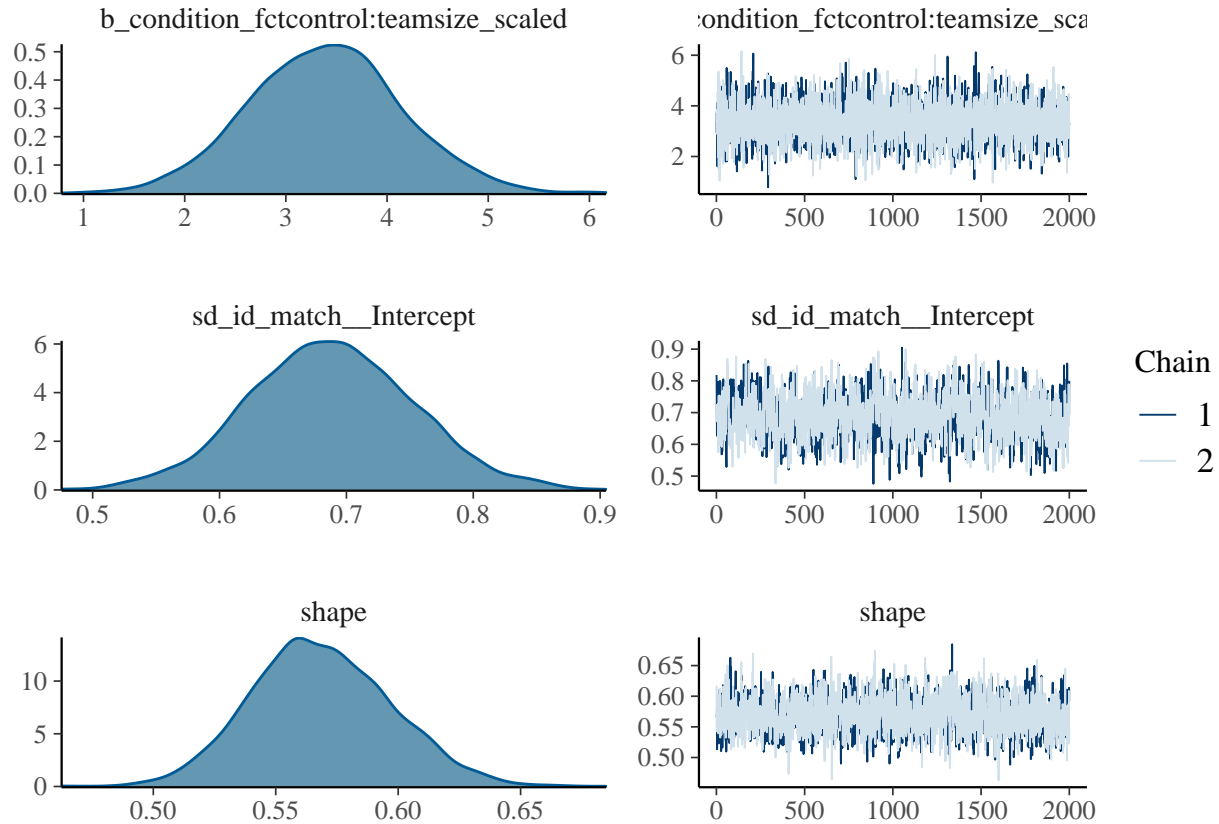
check estimates

More effective samples for intercept models. No Rhat issues, looks pretty good. Not strictly "significant" given 95% CI, but close – and also stronger effect of teamsize (although problematic because of outlier) – connected with the pareto k issue.

```
print(negbin_post_match_0)
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
## Formula: c_5 ~ 0 + condition_fct + condition_fct:teamsize_scaled + (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)     0.69      0.06     0.56     0.81 1.00     1003     1975
##
## Population-Level Effects:
##                           Estimate Est.Error l-95% CI u-95% CI
## condition_fctexperiment       2.60      0.07     2.46     2.75
## condition_fctcontrol          2.11      0.08     1.95     2.27
```

```
## condition_fctexperiment:teamsize_scaled    2.34       0.67      1.03      3.67
## condition_fctcontrol:teamsize_scaled        3.49       0.74      2.01      4.92
##                                             Rhat Bulk_ESS Tail_ESS
## condition_fctexperiment                     1.00     3504     3187
## condition_fctcontrol                        1.00     2803     3093
## condition_fctexperiment:teamsize_scaled 1.00         7463     3430
## condition_fctcontrol:teamsize_scaled    1.00         8158     3507
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.57      0.03     0.51     0.63 1.00     1637     2868
##
## 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_match_1) # more effective samples
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
## Formula: c_5 ~ 1 + condition_fct + condition_fct:teamsize_scaled + (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)     0.69      0.06     0.56     0.81 1.00     1048     1849
##
## Population-Level Effects:
##                                         Estimate Est.Error l-95% CI u-95% CI
## Intercept                                   2.61      0.07     2.47     2.75
## condition_fctcontrol                       -0.49      0.10    -0.68    -0.30
## condition_fctexperiment:teamsize_scaled     2.29      0.66     1.04     3.58
## condition_fctcontrol:teamsize_scaled        3.40      0.74     1.95     4.88
##                                         Rhat Bulk_ESS Tail_ESS
## Intercept                               1.00     3820     3084
## condition_fctcontrol                    1.00     7469     3303
## condition_fctexperiment:teamsize_scaled 1.00     7164     3311
## condition_fctcontrol:teamsize_scaled    1.00     7060     3153
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.57      0.03     0.51     0.63 1.00     1945     2756
##
## 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_match_01)
```

```
##  Family: negbinomial
```

```
##    Links: mu = log; shape = identity
## Formula: c_5 ~ 0 + Intercept + condition_fct + condition_fct:teamsize_scaled + (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)     0.69      0.06     0.56     0.82 1.00      827     2134
##
## Population-Level Effects:
##                                     Estimate Est.Error l-95% CI u-95% CI
## Intercept                               2.60      0.07     2.46     2.74
## condition_fctcontrol                   -0.48      0.10    -0.66    -0.28
## condition_fctexperiment:teamsize_scaled  2.36      0.68     1.07     3.69
## condition_fctcontrol:teamsize_scaled    3.39      0.76     1.92     4.90
##                                     Rhat Bulk_ESS Tail_ESS
## Intercept                           1.00     2524     2527
## condition_fctcontrol                1.00     3456     3075
## condition_fctexperiment:teamsize_scaled 1.00  3954     3094
## condition_fctcontrol:teamsize_scaled 1.00    5324     3313
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.57      0.03     0.52     0.63 1.00     1402     2284
##
## 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_match_0, draws = 500)
y_1 <- posterior_predict(negbin_post_match_1, draws = 500)
y_01 <- posterior_predict(negbin_post_match_01, draws = 500)
```
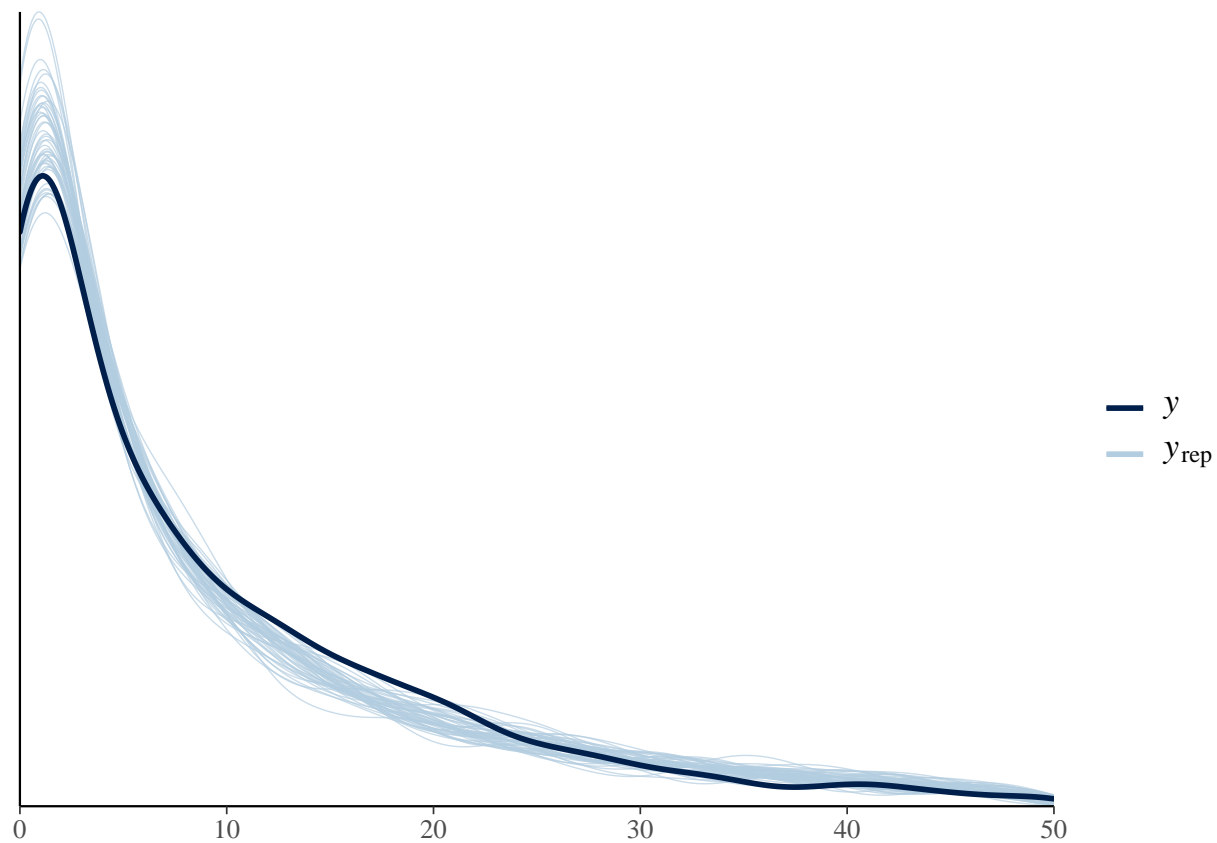
no intercept

looks pretty good. a lot of uncertainty around 0 and 1 still.

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

```
## Warning: Removed 6170 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 751 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 6154 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 111 rows containing non-finite values (stat_density).
```
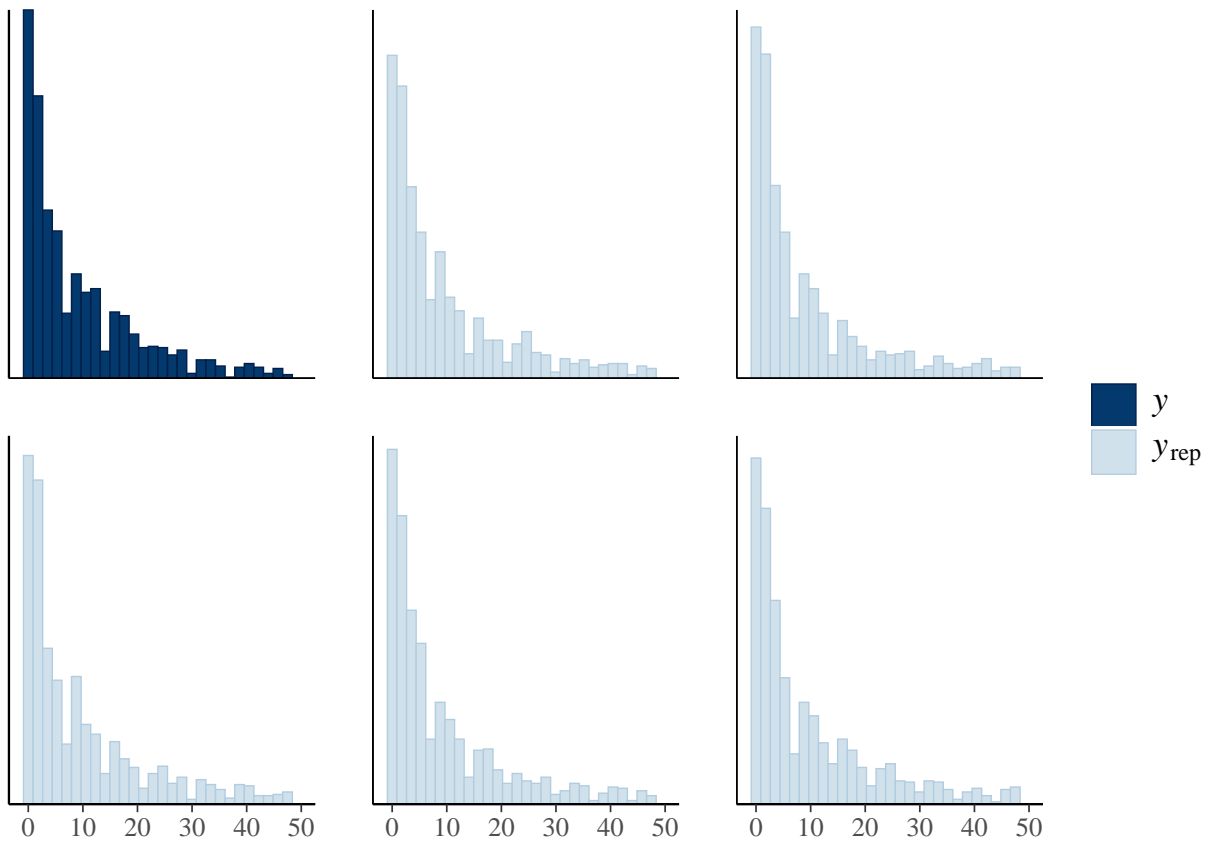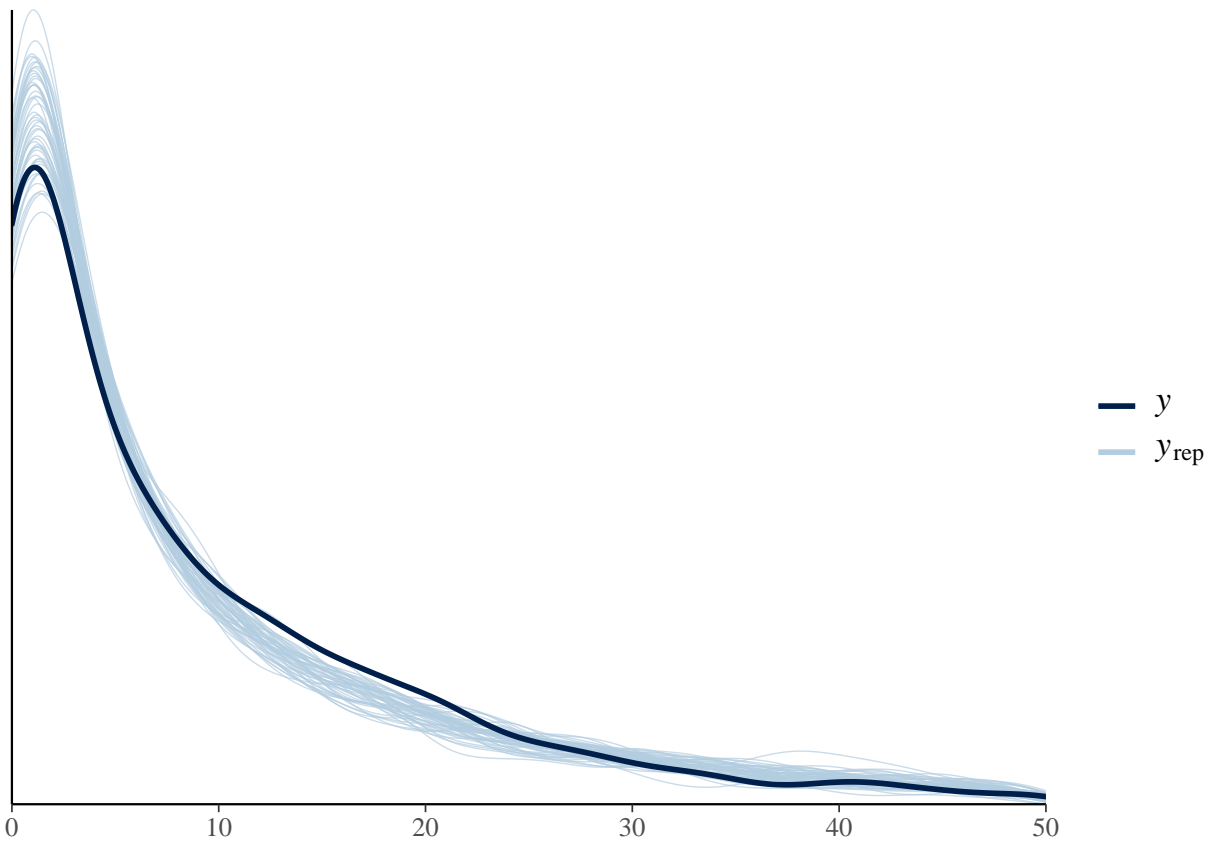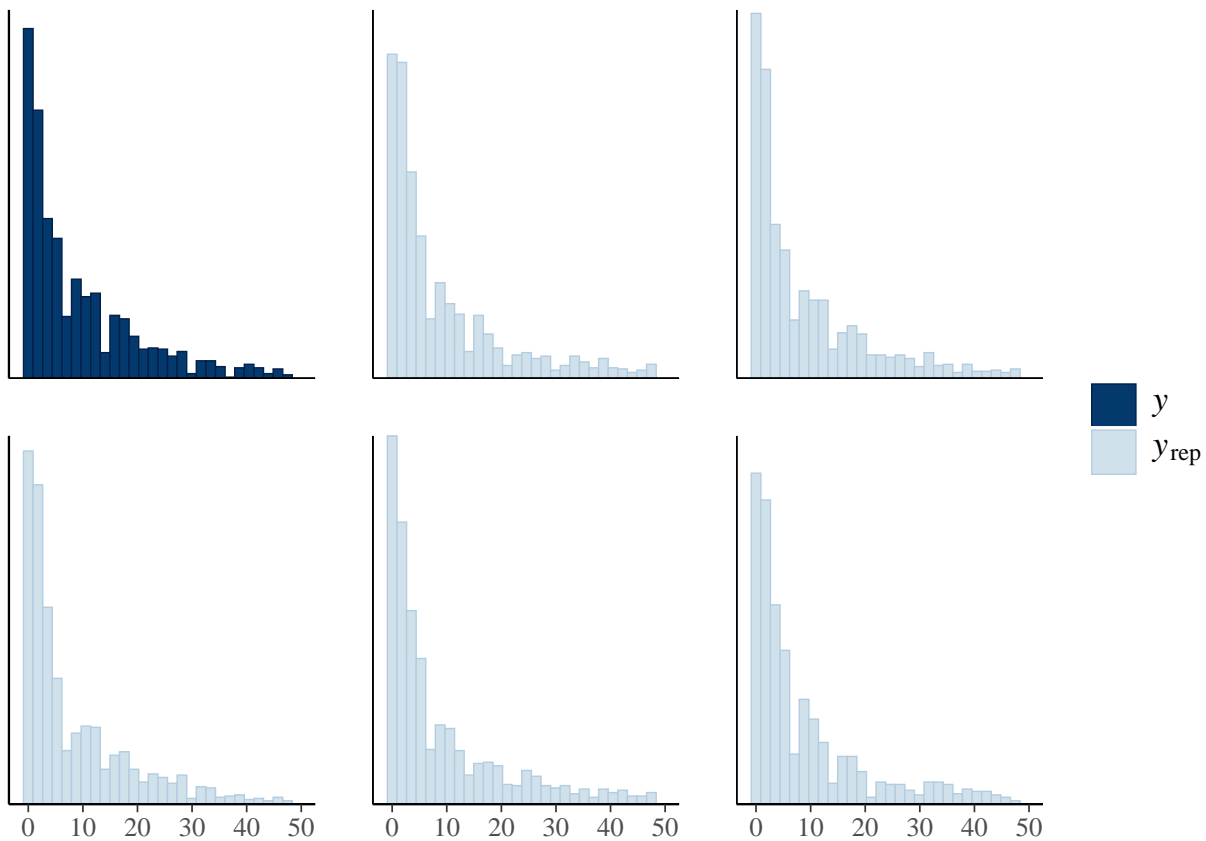
```
ppc_hist(y, y_1[1:5, ]) + xlim(-1, 50)
```

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

```
## Warning: Removed 724 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 6188 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 724 rows containing non-finite values (stat_bin).

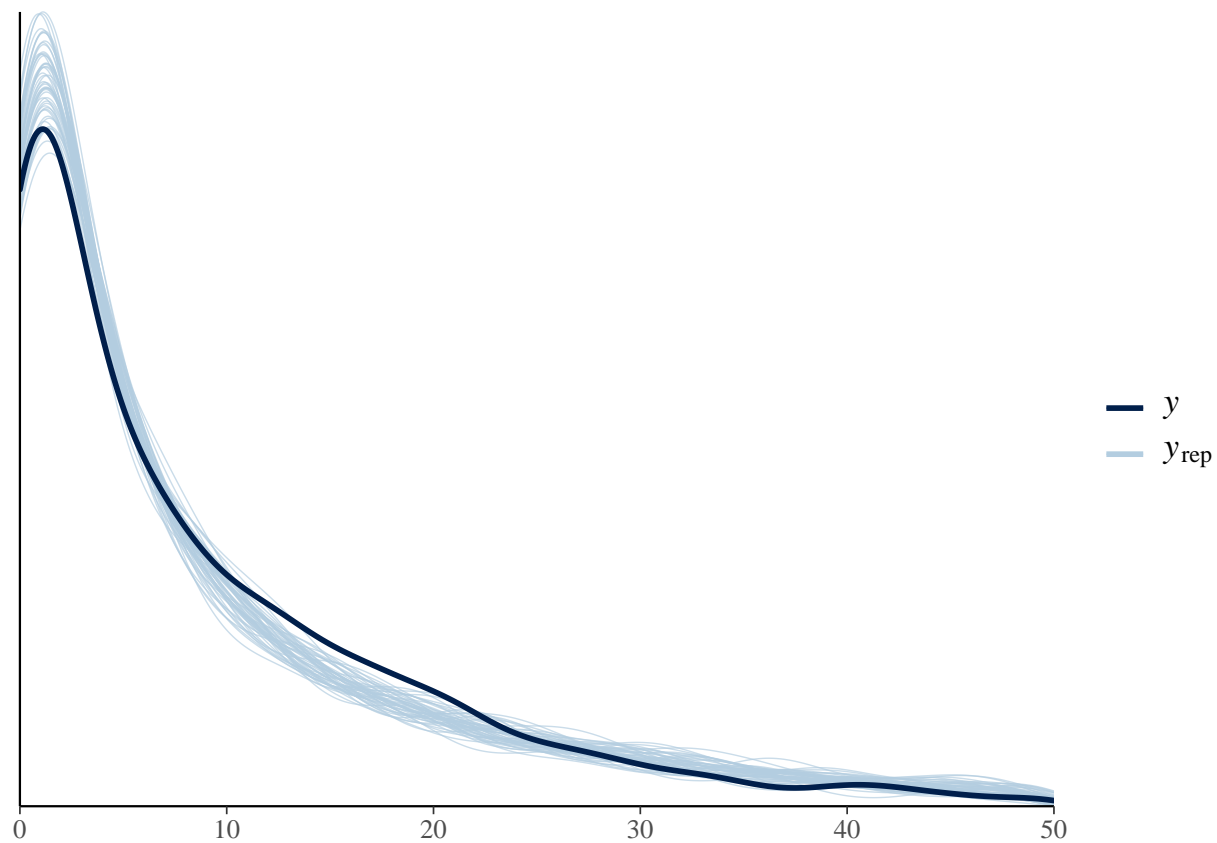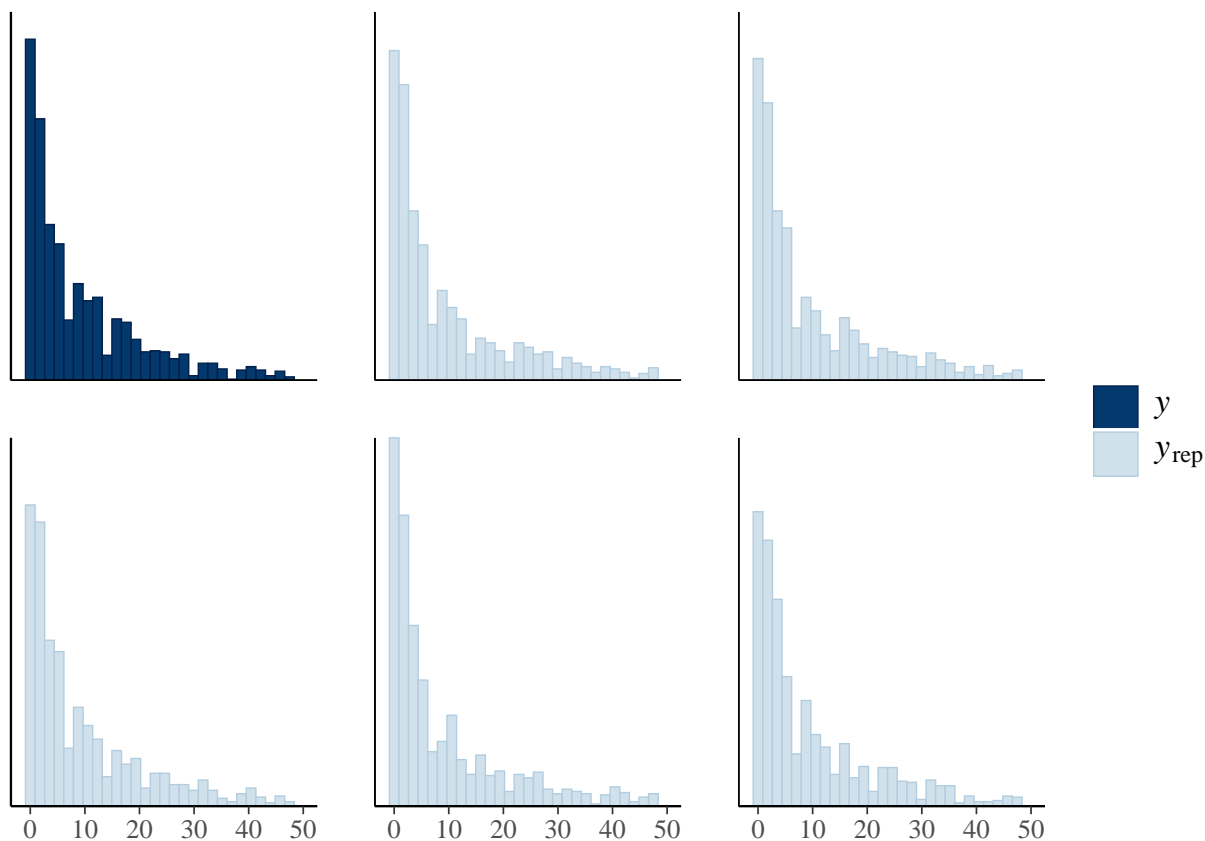## Warning: Removed 12 rows containing missing values (geom_bar).

pareto k issues

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

no intercept

mainly studies that are (relatively) low teamsize and high citation

```
d %>%
  mutate(k = negbin_post_match_0$criteria$loo$diagnostics$pareto_k) %>%
  filter(k > .7) %>%
  select(c_5, teamsize_scaled, condition_fct, id_match, k)
```

```
## # A tibble: 70 x 5
##     c_5 teamsize_scaled condition_fct id_match     k
##   <dbl>           <dbl> <fct>         <fct>    <dbl>
## 1   310          0.05   experiment    3        0.928
## 2    41          0.0667 control       8        0.871
## 3   125          0.0833 experiment    17       0.949
## 4    26          0.0333 control       39       0.924
## 5    99          0.167  experiment    44       0.798
## 6   169          0.0667 experiment    45       0.825
## 7    69          0.05   control       46       0.771
```

```
## 8    70       0.0833 control        62      0.757
## 9    236      0.05   experiment     66      0.885
## 10   995      0.133  experiment     85      1.00
## # ... with 60 more rows
```

$1 + \ldots$

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

```
d %>%
  mutate(k = negbin_post_match_1$criteria$loo$diagnostics$pareto_k) %>%
  filter(k > .7) %>%
  select(c_5, teamsize_scaled, condition_fct, id_match, k)
```

```
## # A tibble: 53 x 5
##      c_5 teamsize_scaled condition_fct id_match     k
##    <dbl>           <dbl> <fct>         <fct>    <dbl>
## 1   310          0.05    experiment    3        0.964
## 2    41          0.0667  control       8        0.731
## 3    78          0.0167  experiment    23       0.740
## 4    84          0.05    control       42       1.15
## 5    51          0.0833  experiment    43       0.768
## 6    52          0.0667  control       54       0.716
## 7    70          0.0833  control       62       0.889
## 8   236          0.05    experiment    66       0.726
## 9    77          0.05    experiment    74       0.839
## 10  161          0.1     experiment    83       0.860
## # ... with 43 more rows
```

$0 + \text{Intercept}$

Seems to handle influential observations much better. Good explanation for all of the outliers: (1) the two control studies are high influence because they are the same (fix earlier in pipeline) (2) the experiment study is high influence because there are very few studies with high teamsize (could do log-transformation) and the corresponding control study has zero citations... so this has an outsize influence on the interaction with condition and teamsize.

```
# 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_match_01$criteria$loo$diagnostics$pareto_k) %>%
  filter(k > .7) %>%
  select(c_5, teamsize_scaled, condition_fct, id_match, k)
```

```
## # A tibble: 53 x 5
##      c_5 teamsize_scaled condition_fct id_match     k
##    <dbl>           <dbl> <fct>         <fct>    <dbl>
## 1   310          0.05    experiment    3        1.24
## 2    41          0.0667  control       8        0.789
## 3    58          0.05    experiment    13       0.923
```

```
## 4    125             0.0833 experiment    17       0.898
## 5     84             0.05   control       42       0.901
## 6    169             0.0667 experiment    45       0.704
## 7     30             0.0333 experiment    77       0.859
## 8    995             0.133  experiment    85       0.950
## 9     82             0.0167 control       99       0.785
## 10   235             0.05   control       138      1.06
## # ... with 43 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_match_0,
            negbin_post_match_1,
            negbin_post_match_01)
```

```
##                       elpd_diff se_diff
## negbin_post_match_01   0.0       0.0
## negbin_post_match_1   -2.8       2.1
## negbin_post_match_0   -4.7       2.1
```

```
loo_model_weights(negbin_post_match_0,
                  negbin_post_match_1,
                  negbin_post_match_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_match_0  0.000
## negbin_post_match_1  0.040
## negbin_post_match_01 0.960
```