explore likelihood

Victor Møller Poulsen

load data

```
# 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,~
```

read models (from fit__likelihood.Rmd)

```
zip_prior <- readRDS("/work/50114/MAG/modeling/models/zip_prior.rds")
zip_post <- readRDS("/work/50114/MAG/modeling/models/zip_post.rds")
negbin_prior <- readRDS("/work/50114/MAG/modeling/models/negbin_prior.rds")
negbin_post <- readRDS("/work/50114/MAG/modeling/models/negbin_post.rds")
zinegbin_prior <- readRDS("/work/50114/MAG/modeling/models/zinegbin_prior.rds")
zinegbin_post <- readRDS("/work/50114/MAG/modeling/models/zinegbin_post.rds")
```

prior predictive checks

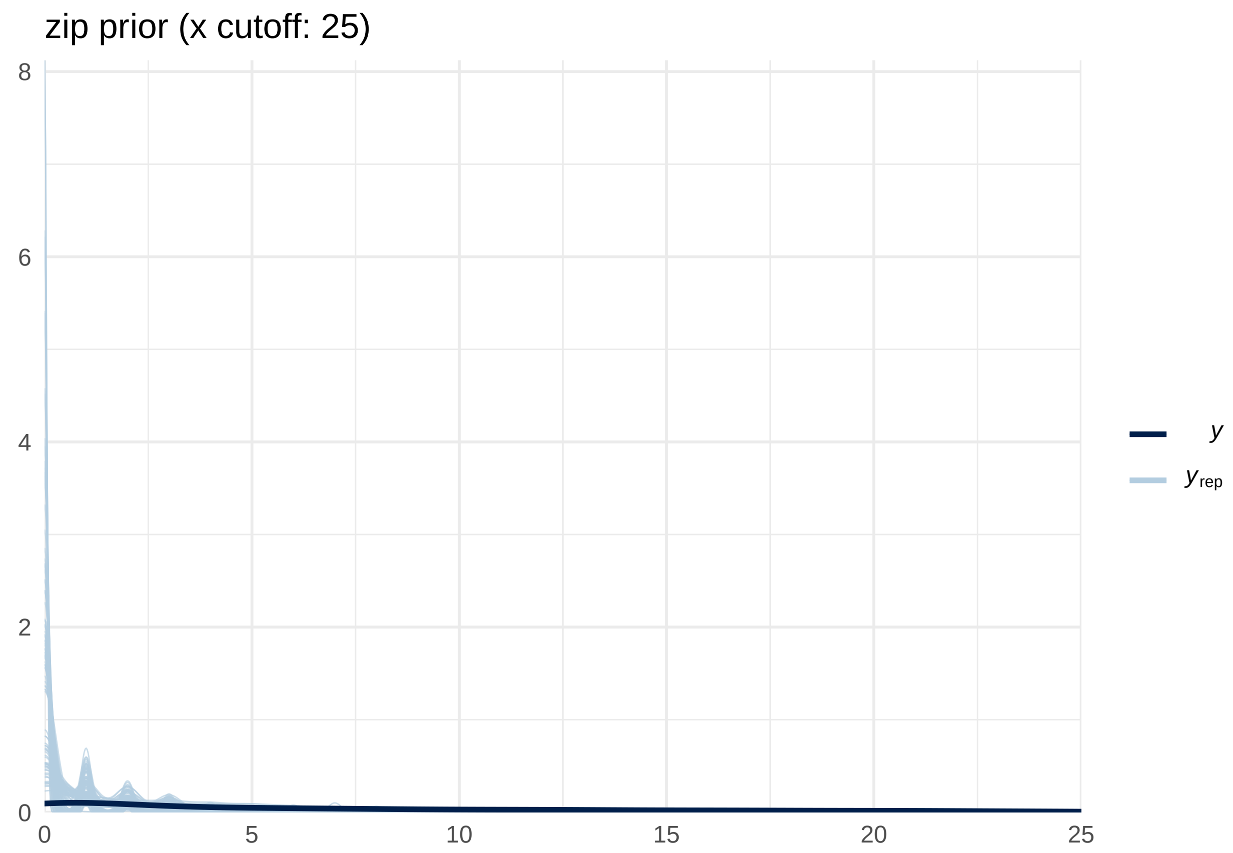
```
prior_check <- function(model, ndraws, title, xmax){

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

}
```

ZIP prior

```
prior_check(zip_prior, 100, "zip prior (x cutoff: 25)", 25)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_density).
```

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

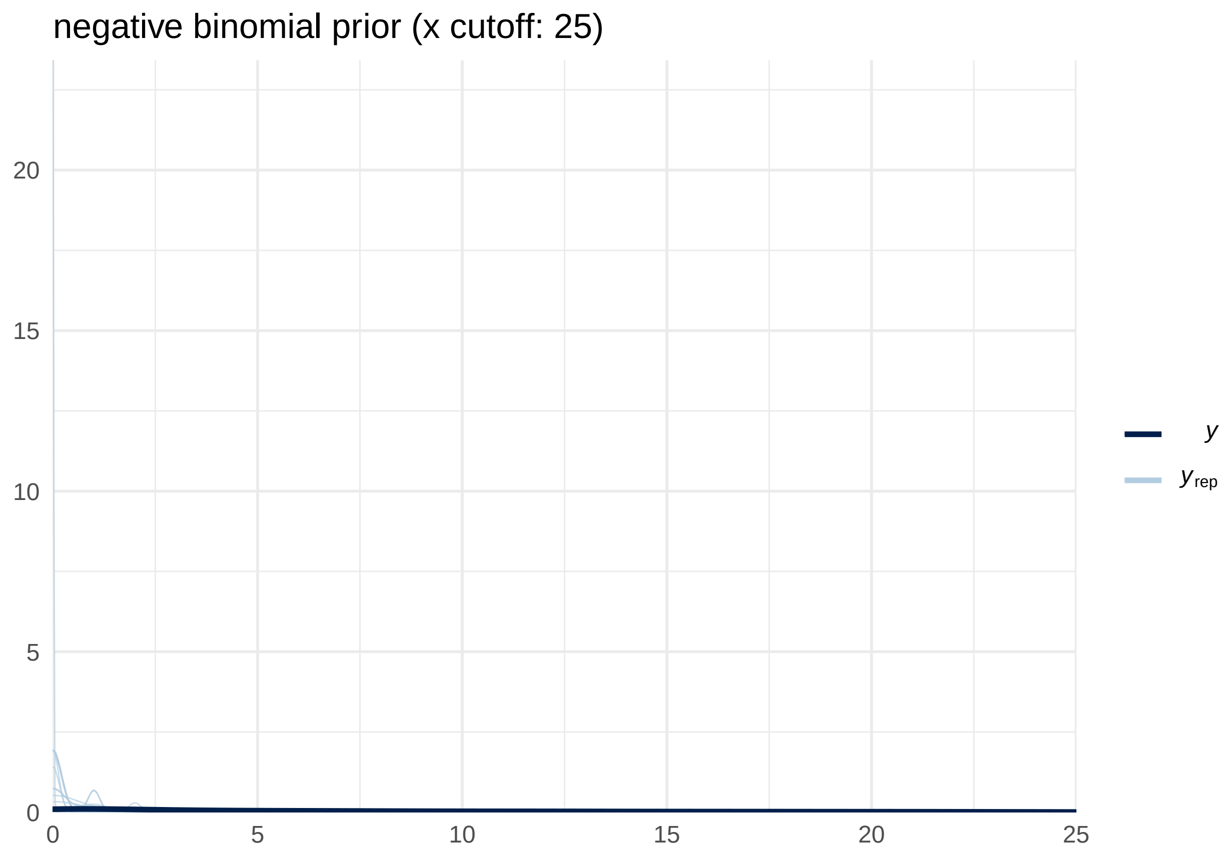


negative binomial prior

```
prior_check(negbin_prior, 100, "negative binomial prior (x cutoff: 25)", 25)
```

```
## Warning: Removed 19 rows containing non-finite values (stat_density).
```

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

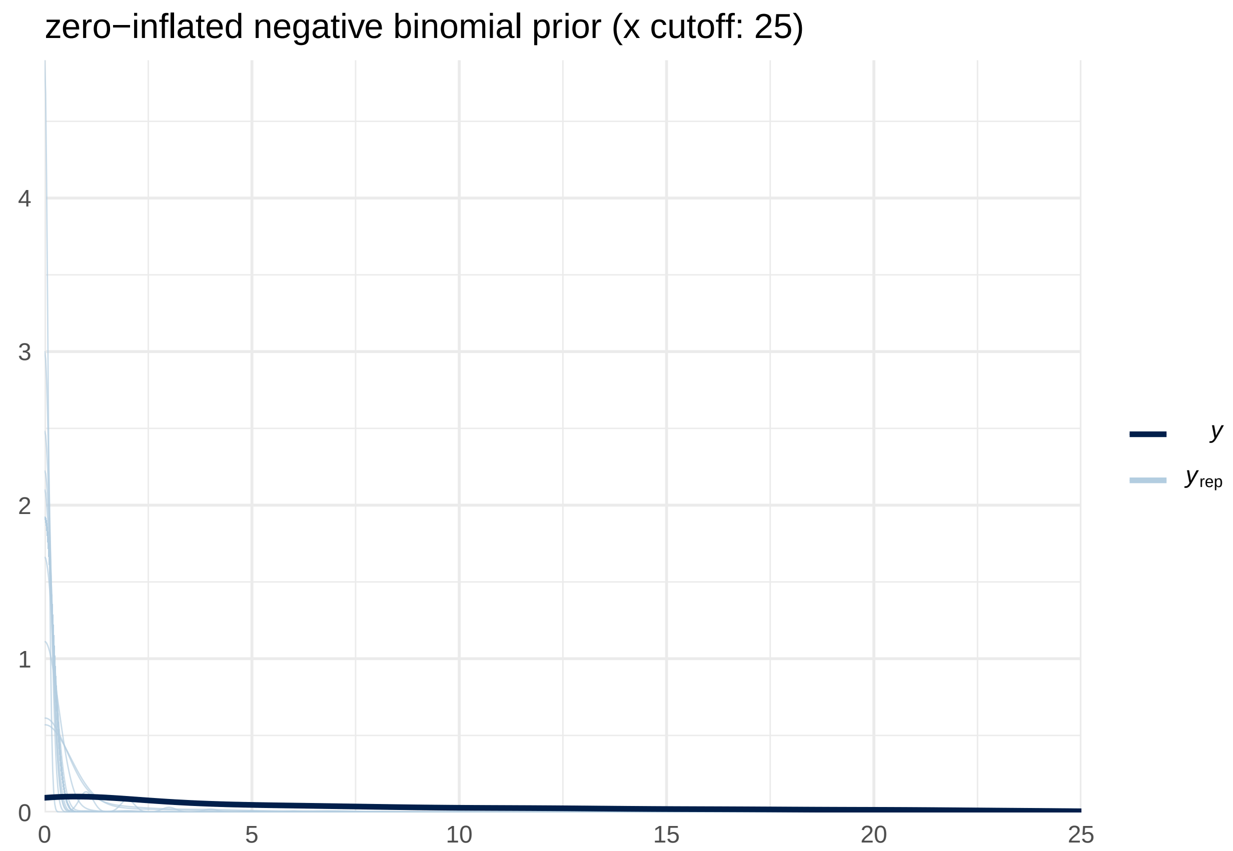


zero-inflated negative binomial prior

```
prior_check(zinegbin_prior, 100, "zero-inflated negative binomial prior (x cutoff: 25)", 25)
```

```
## Warning: Removed 56 rows containing non-finite values (stat_density).
```

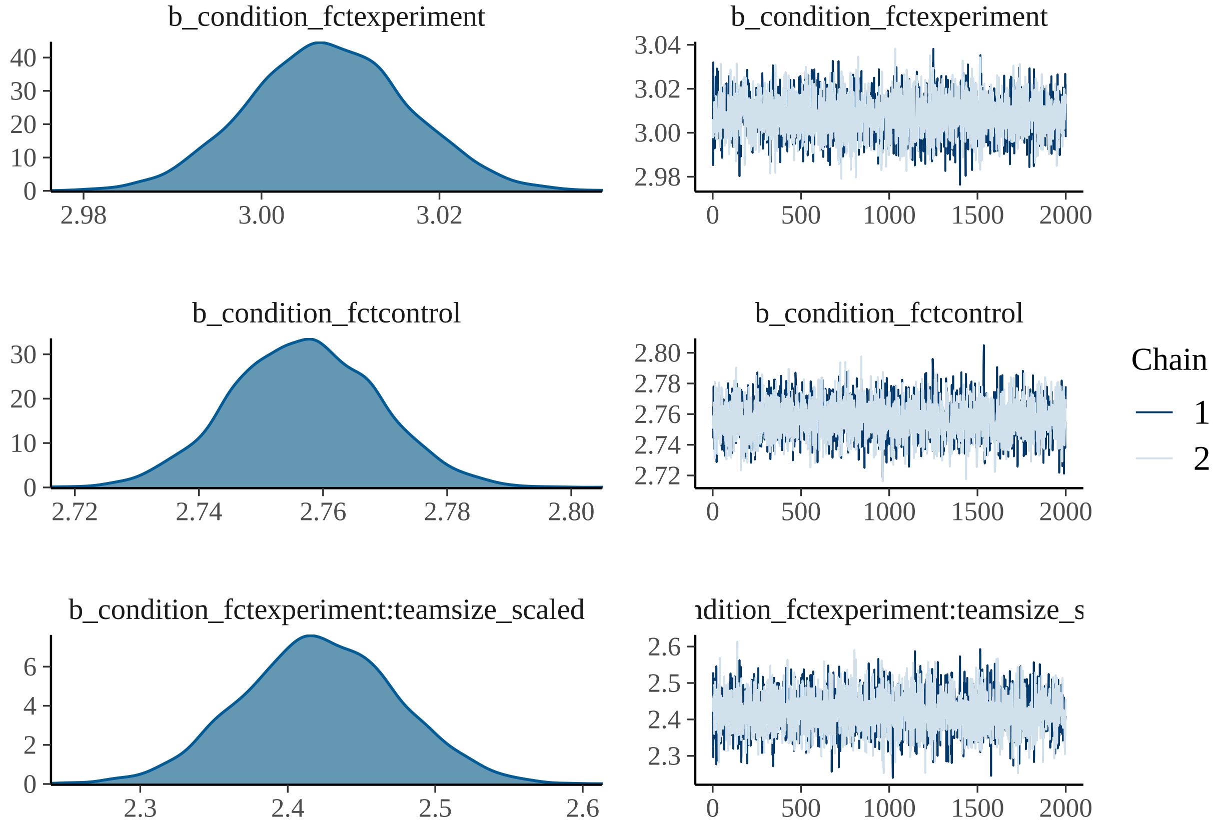
```
## Warning: Removed 253 rows containing non-finite values (stat_density).
```

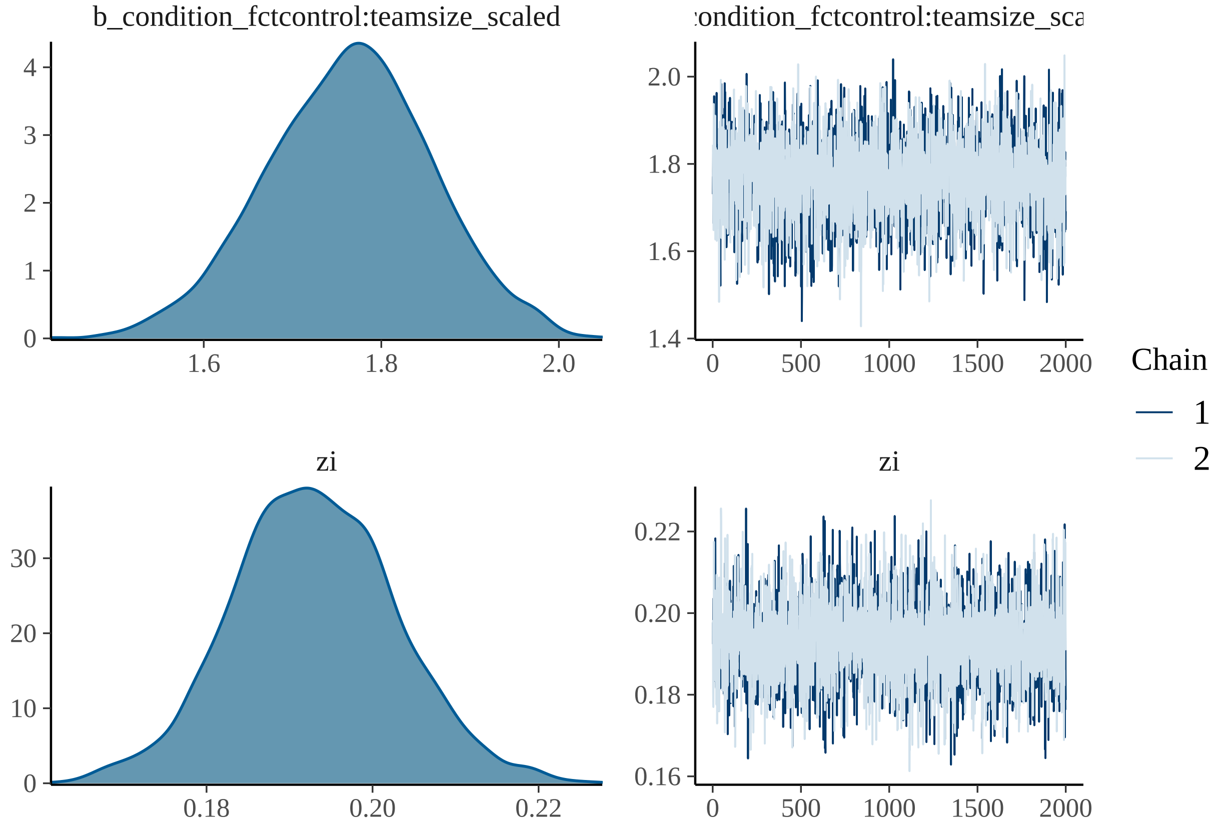


plot traces

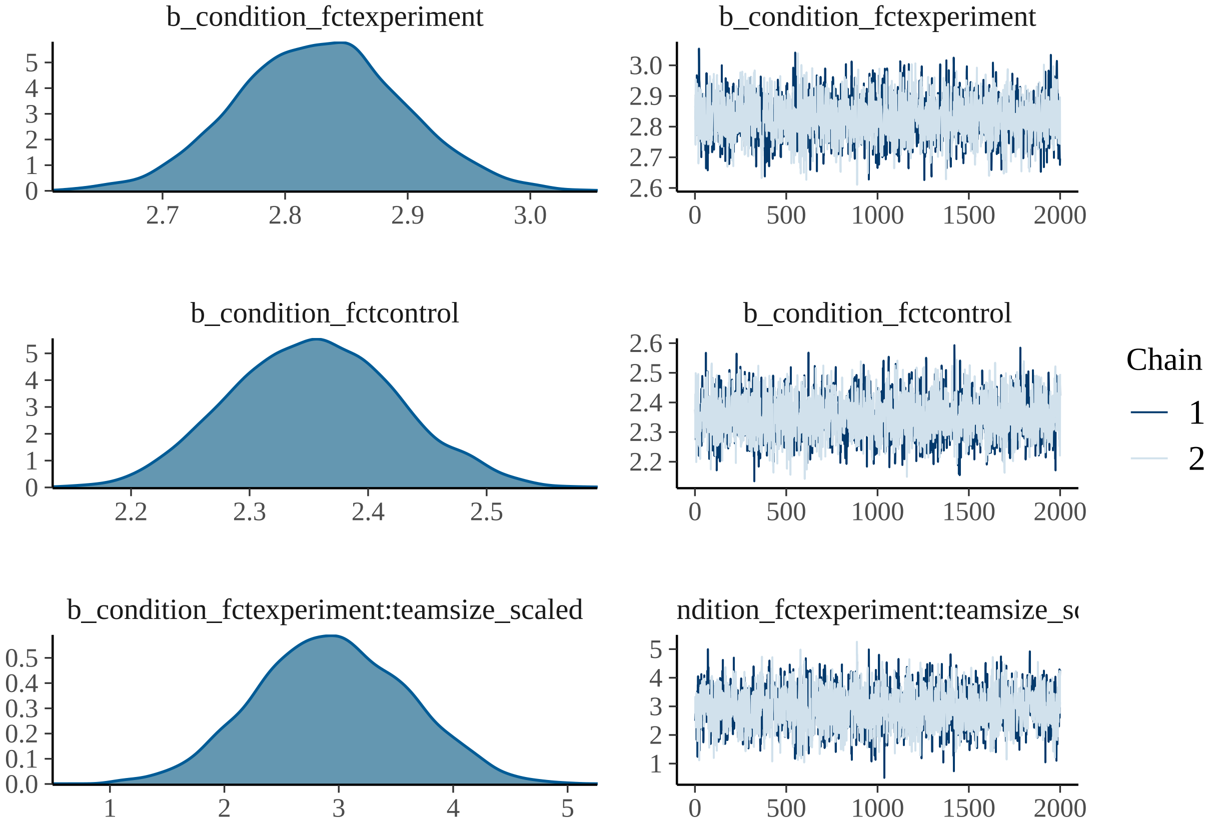
reasonable for negative binomial and for zero-inflated negative binomial

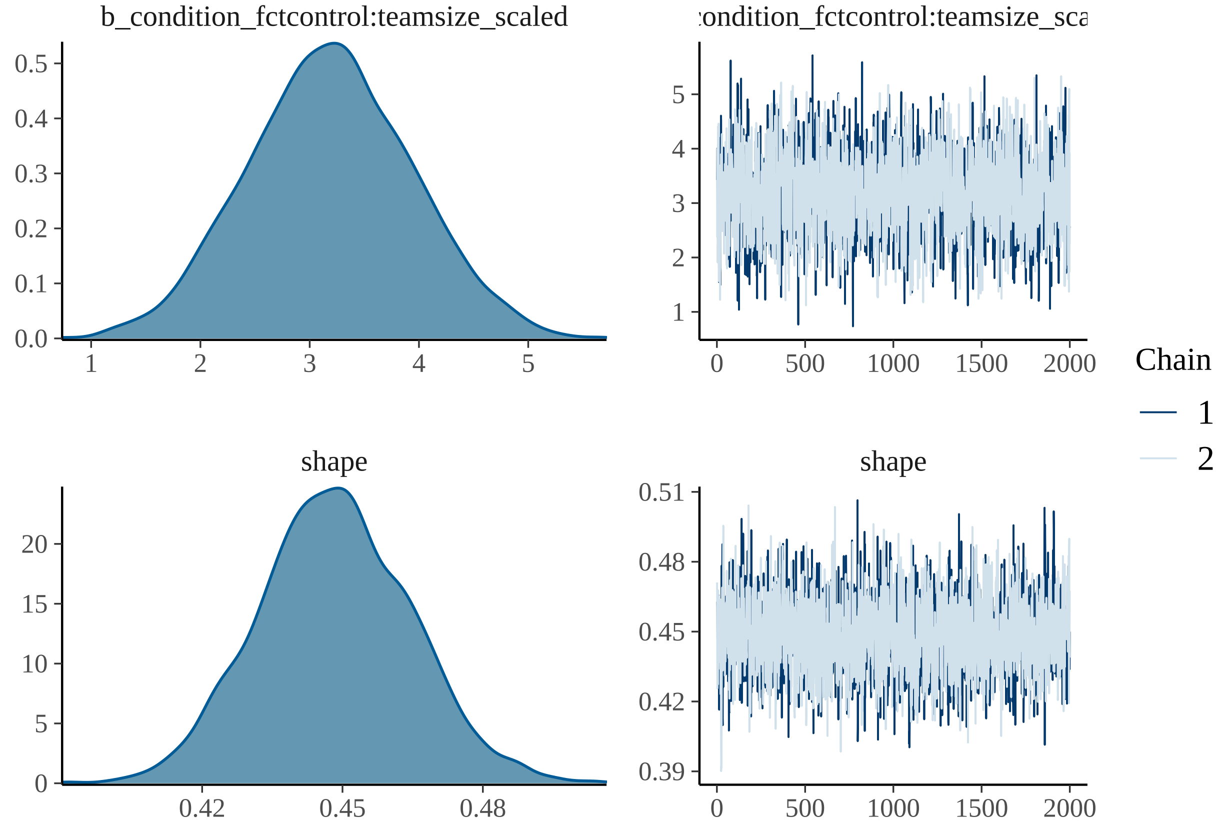
```
plot(zip_post, N = 3) # gives weird stuff
```


b_condition_fctexperiment
40
30
20
10
0
2.98
3.00
3.02
b_condition_fctexperiment
3.04
3.02
3.00
2.98
0
500
1000
1500
2000
b_condition_fctcontrol
30
20
10
0
2.72
2.74
2.76
2.78
2.80
b_condition_fctcontrol
2.80
2.78
2.76
2.74
2.72
0
500
1000
1500
2000
Chain
1
2
b_condition_fctexperiment:teamsize_scaled
6
4
2
0
2.3
2.4
2.5
2.6
ndition_fctexperiment:teamsize_s
2.6
2.5
2.4
2.3
0
500
1000
1500
2000

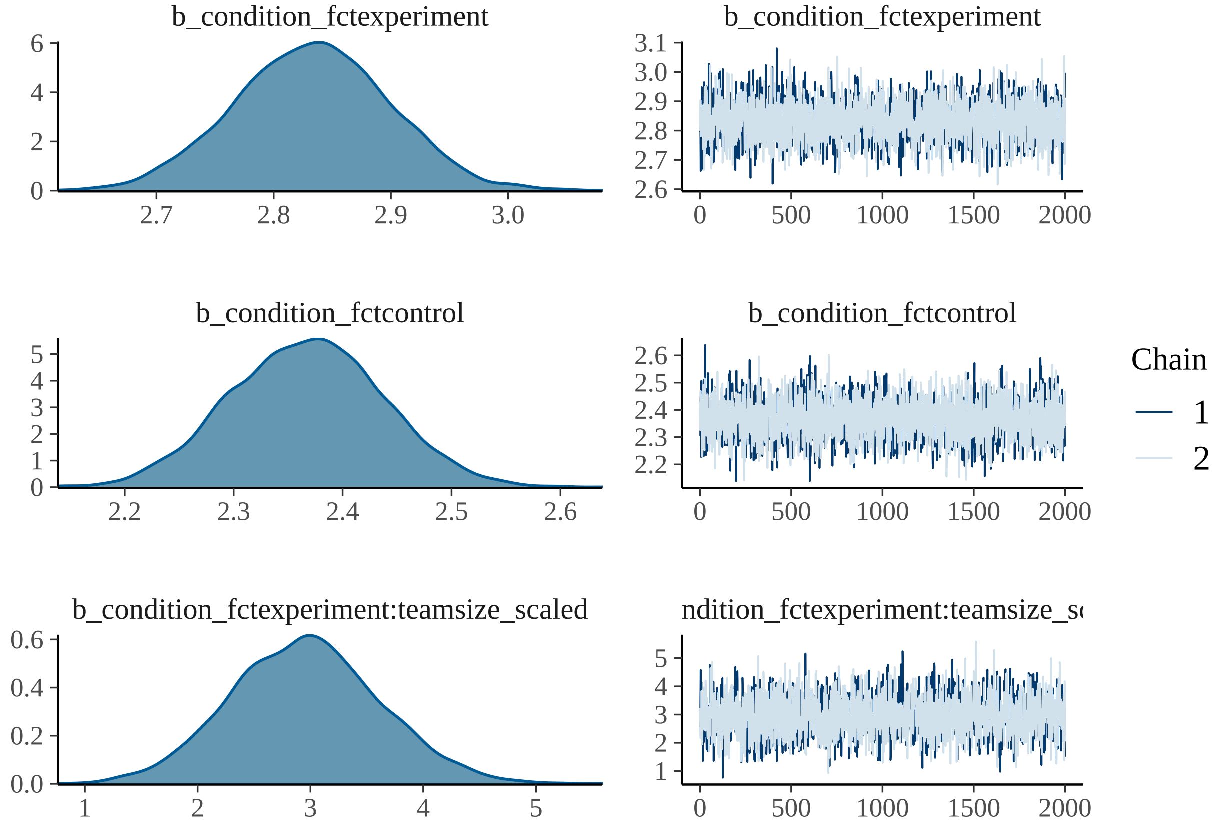


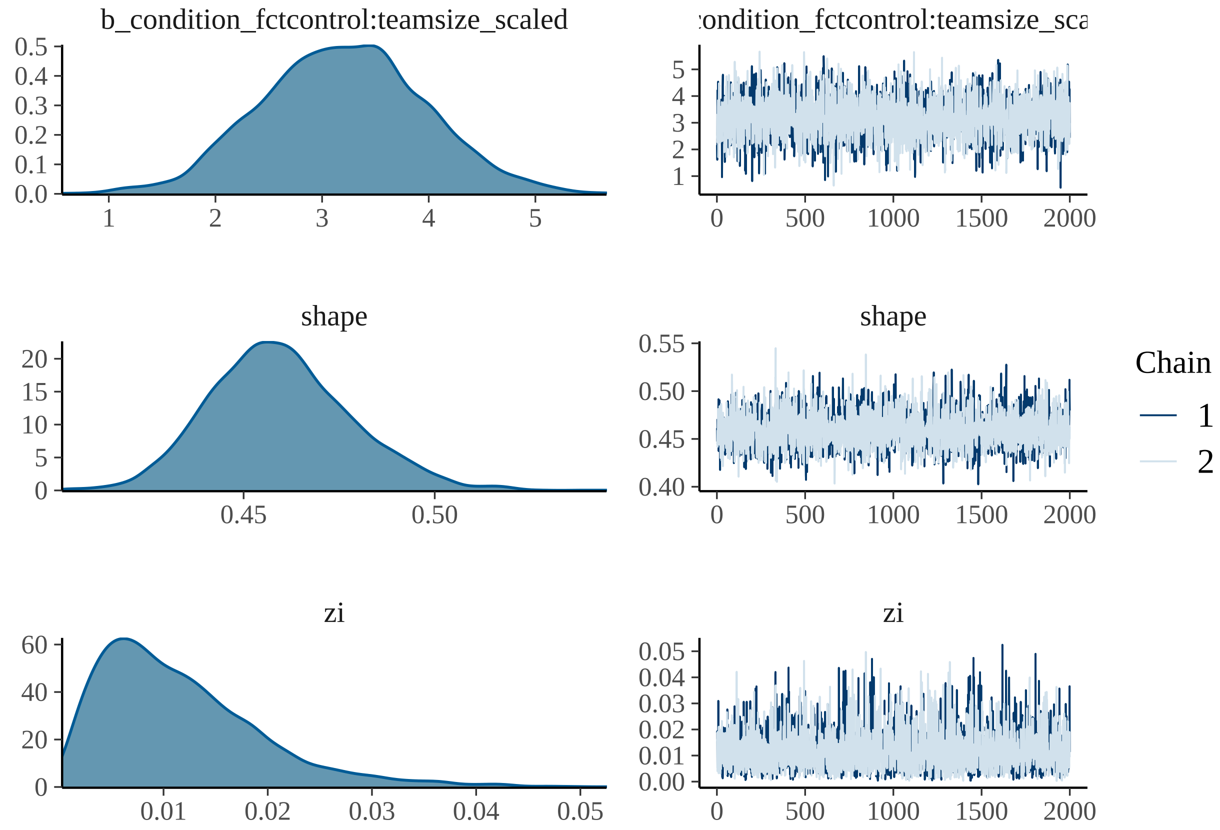
```
plot(negbin_post, N = 3) # much more reasonable
```


b_condition_fctexperiment
5
4
3
2
1
0
2.7
2.8
2.9
3.0
b_condition_fctexperiment
3.0
2.9
2.8
2.7
2.6
0
500
1000
1500
2000
b_condition_fctcontrol
5
4
3
2
1
0
2.2
2.3
2.4
2.5
b_condition_fctcontrol
2.6
2.5
2.4
2.3
2.2
0
500
1000
1500
2000
Chain
1
2
b_condition_fctexperiment:teamsize_scaled
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
ndition_fctexperiment:teamsize_sc
5
4
3
2
1
0
500
1000
1500
2000



```
plot(zinegbin_post, N = 3) # also reasonable
```


b_condition_fctexperiment
6
4
2
0
2.7
2.8
2.9
3.0
b_condition_fctexperiment
3.1
3.0
2.9
2.8
2.7
2.6
0
500
1000
1500
2000
b_condition_fctcontrol
5
4
3
2
1
0
2.2
2.3
2.4
2.5
2.6
b_condition_fctcontrol
2.6
2.5
2.4
2.3
2.2
0
500
1000
1500
2000
Chain
1
2
b_condition_fctexperiment:teamsize_scaled
0.6
0.4
0.2
0.0
1
2
3
4
5
ndition_fctexperiment:teamsize_sc
5
4
3
2
1
0
500
1000
1500
2000



check posterior sampling

https://mc-stan.org/bayesplot/articles/graphical-ppcs.html

```
y <- d$c_5
y_zip <- posterior_predict(zip_post, draws = 500)
y_negbin <- posterior_predict(negbin_post, draws = 500)
y_zinegbin <- posterior_predict(zinegbin_post, draws = 500)
```

overlay & histogram

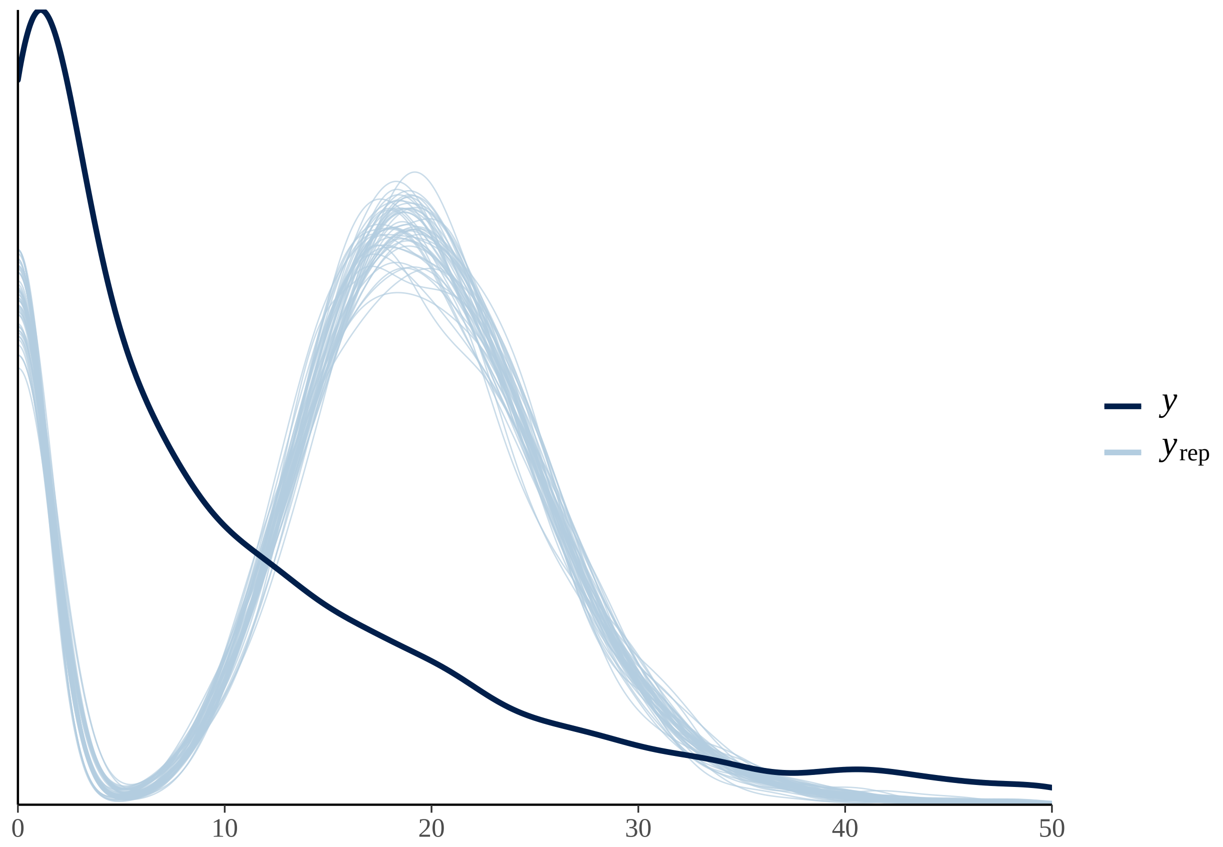
ZIP

we see the key issue again here… we have two distributions which is not reflective of the actual data.

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

```
## Warning: Removed 198 rows containing non-finite values (stat_density).
```

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

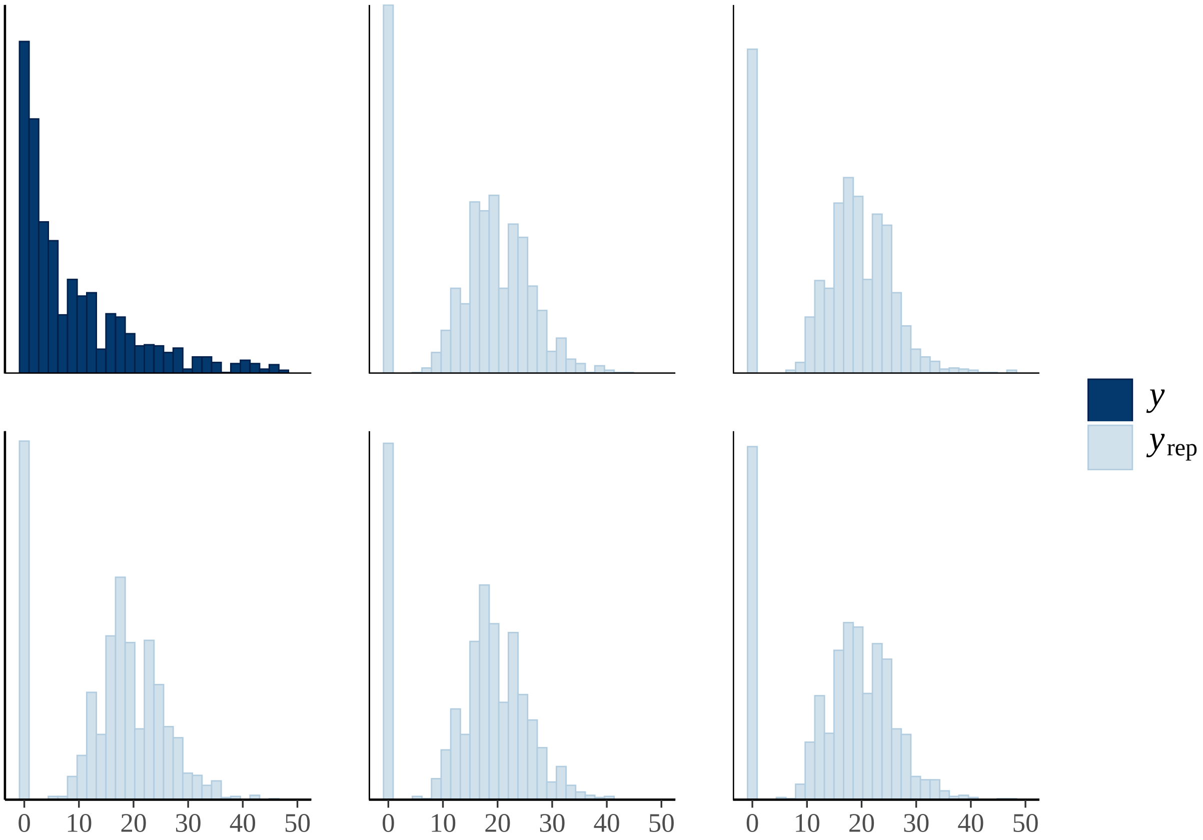


```
ppc_hist(y, y_zip[1:5, ]) + xlim(-1, 50)
```

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

```
## Warning: Removed 126 rows containing non-finite values (stat_bin).
```

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



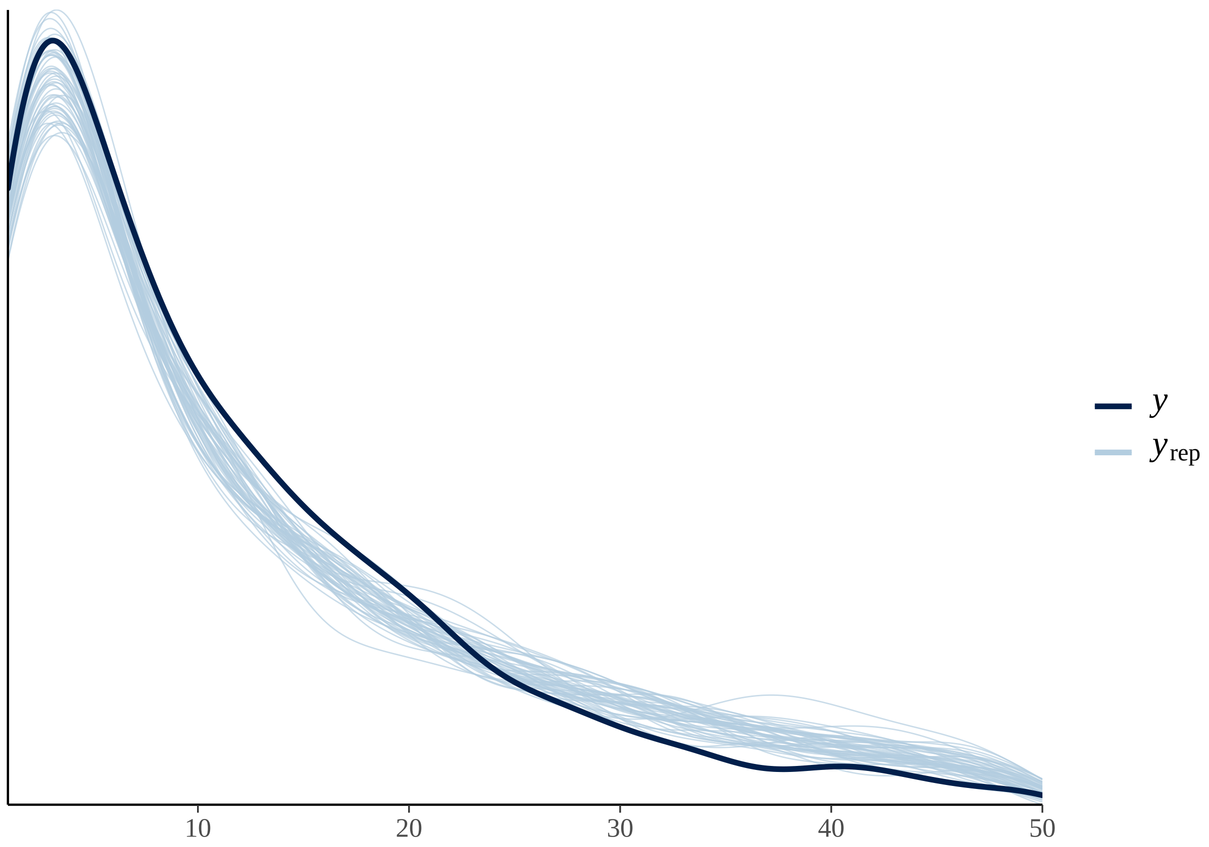
negative binomial

looks really good. has a hard time figuring out the correct number of 0s.

```
ppc_dens_overlay(y, y_negbin[1:50, ]) + xlim(01, 50)
```

```
## Warning: Removed 22597 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 411 rows containing non-finite values (stat_density).
```

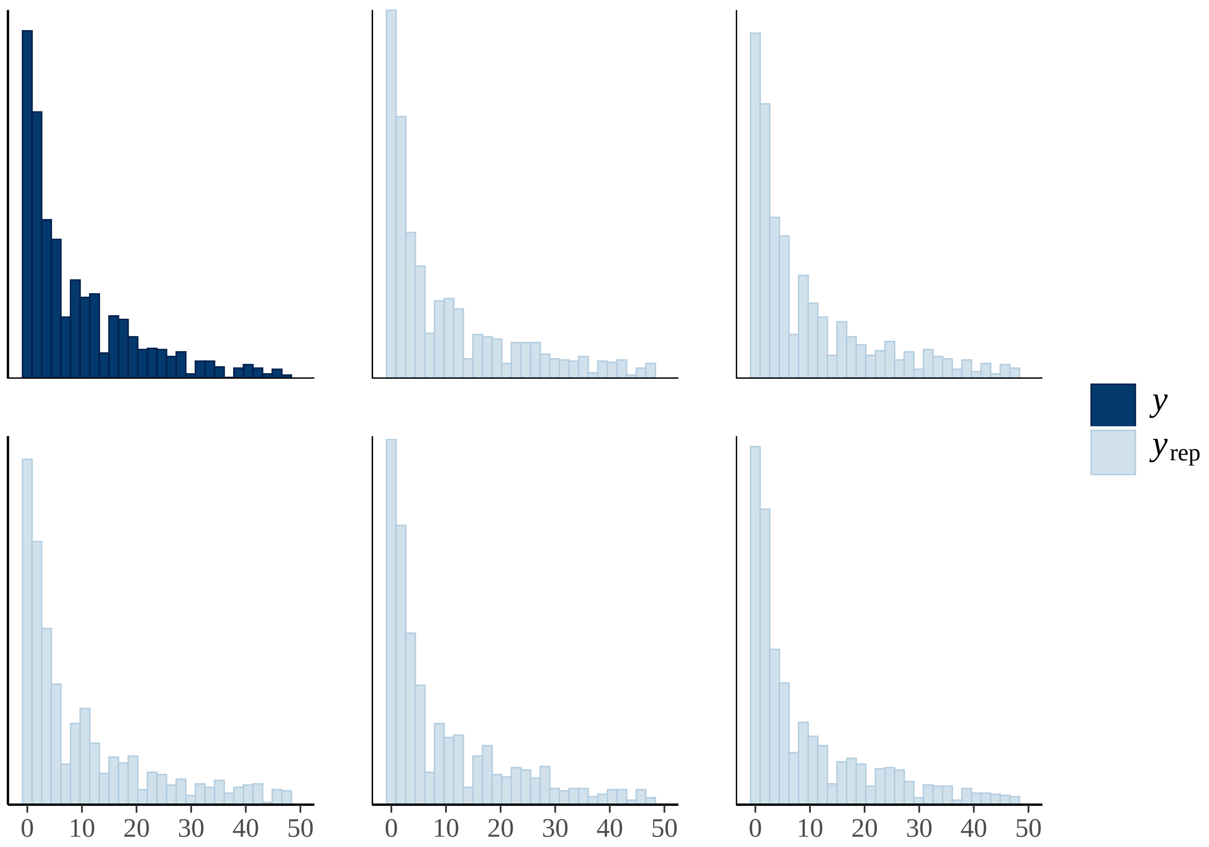


```
ppc_hist(y, y_negbin[1:5, ]) + xlim(-1, 50)
```

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

```
## Warning: Removed 801 rows containing non-finite values (stat_bin).
```

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



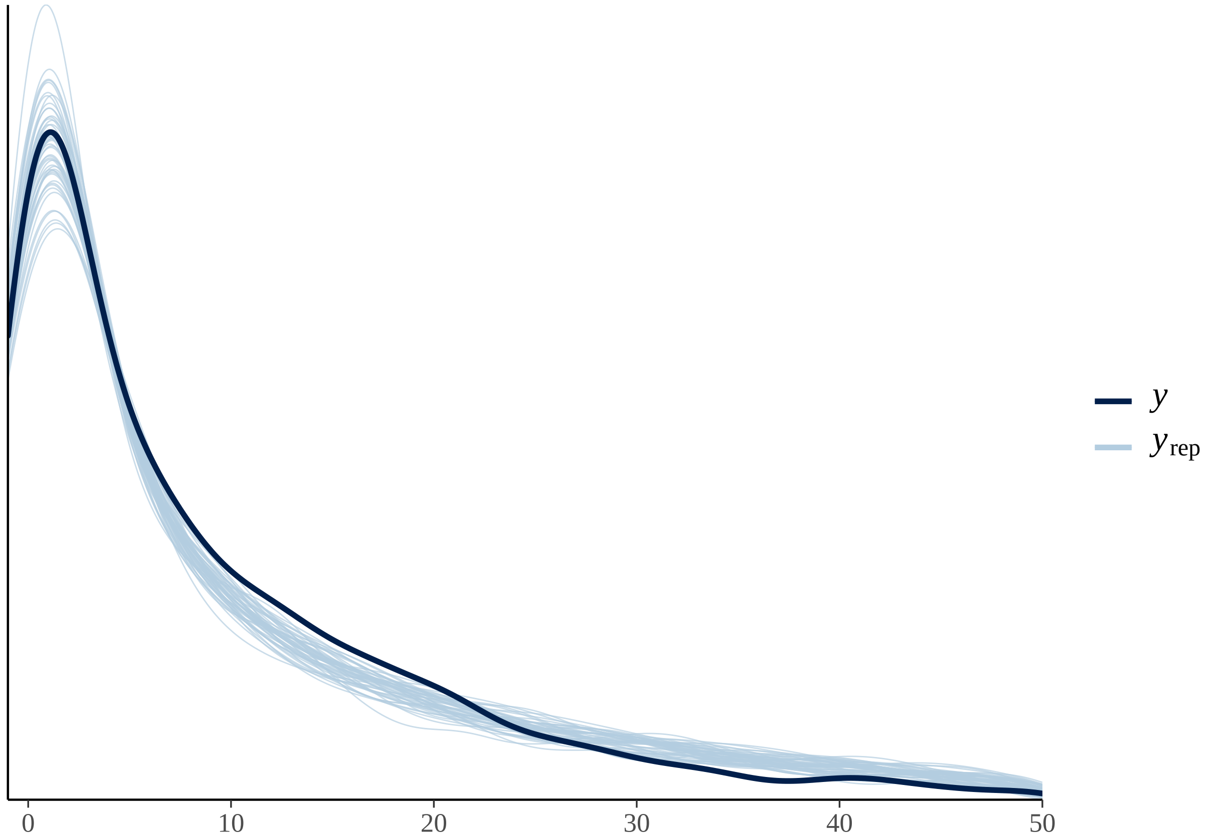
zero-inflated negative binomial

looks really good. very similar to negbin.

```
ppc_dens_overlay(y, y_zinegbin[1:50, ]) + xlim(-1, 50)
```

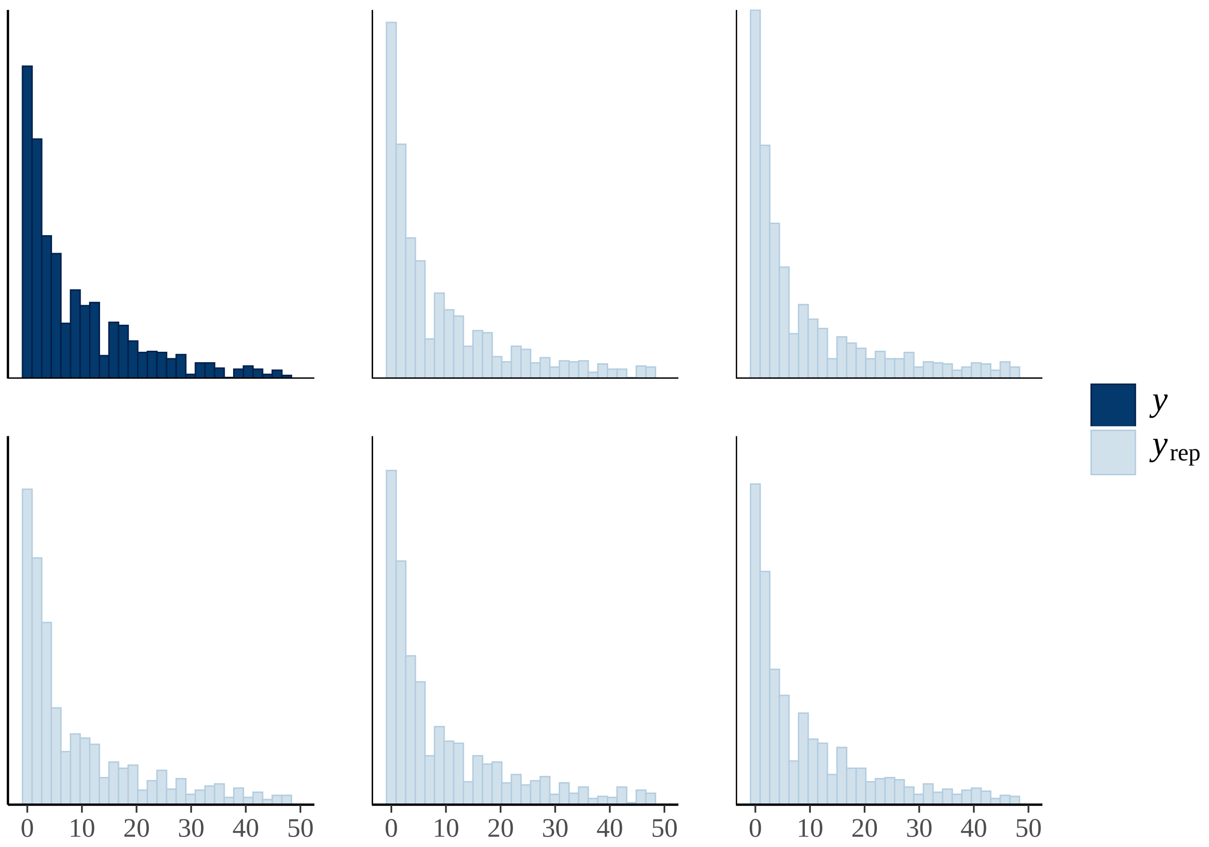
```
## Warning: Removed 6480 rows containing non-finite values (stat_density).
```

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



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

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

## Warning: Removed 706 rows containing non-finite values (stat_bin).

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

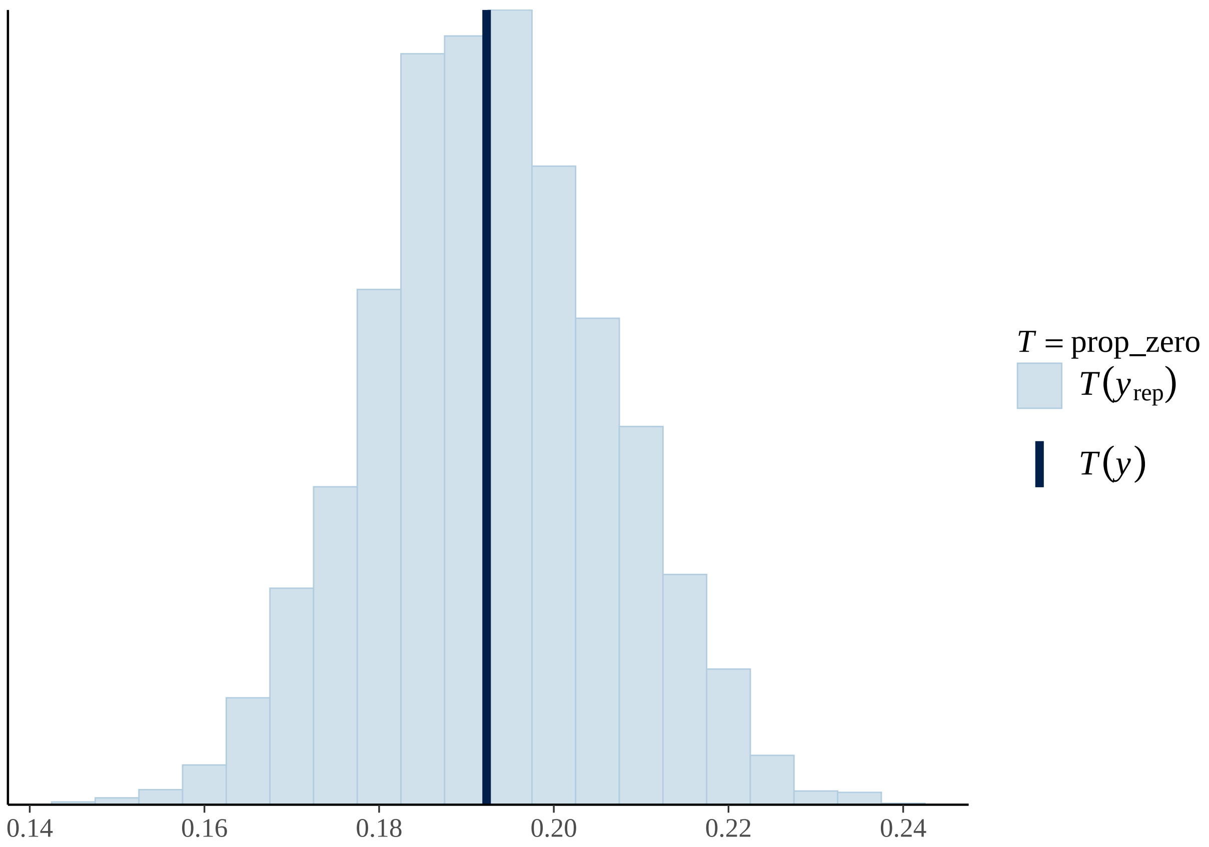


more PPC

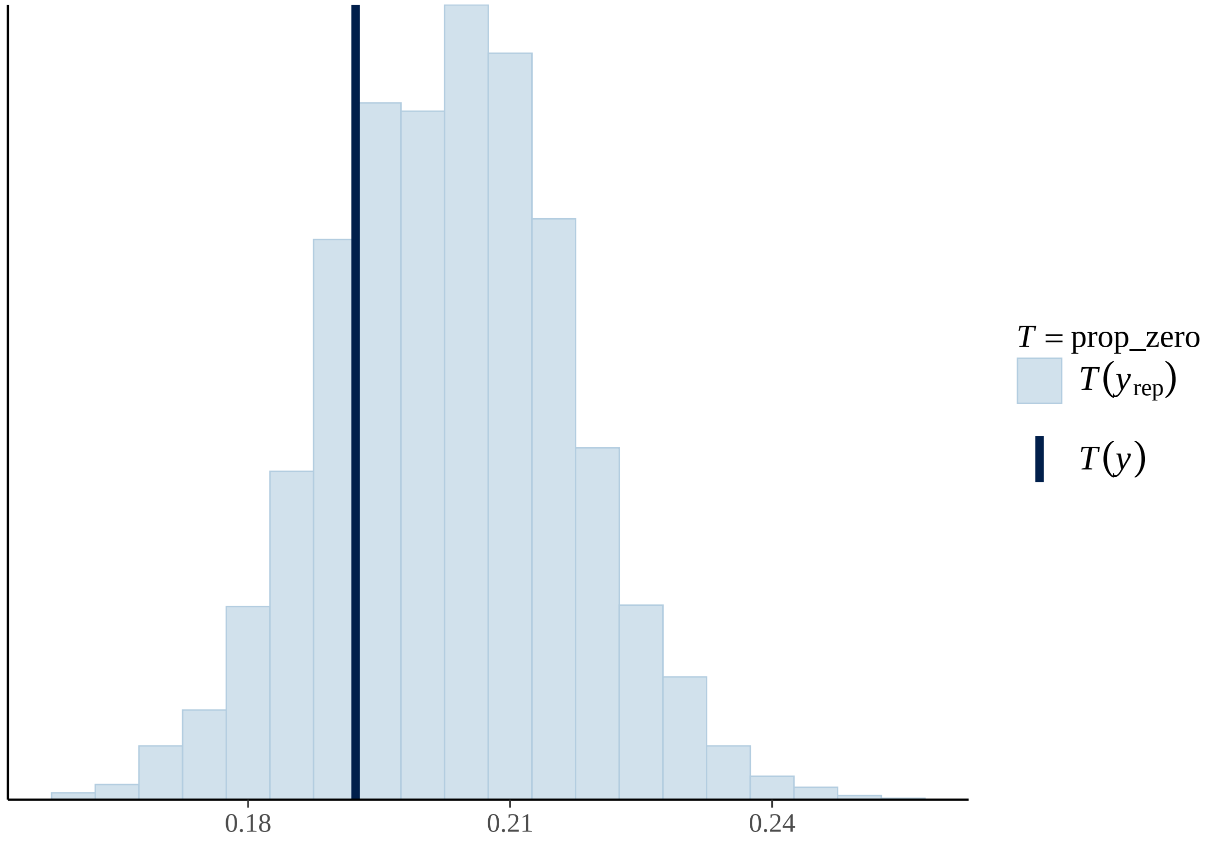
ability of models to produce zeros

All models pretty good at producing the right amount of zeros.

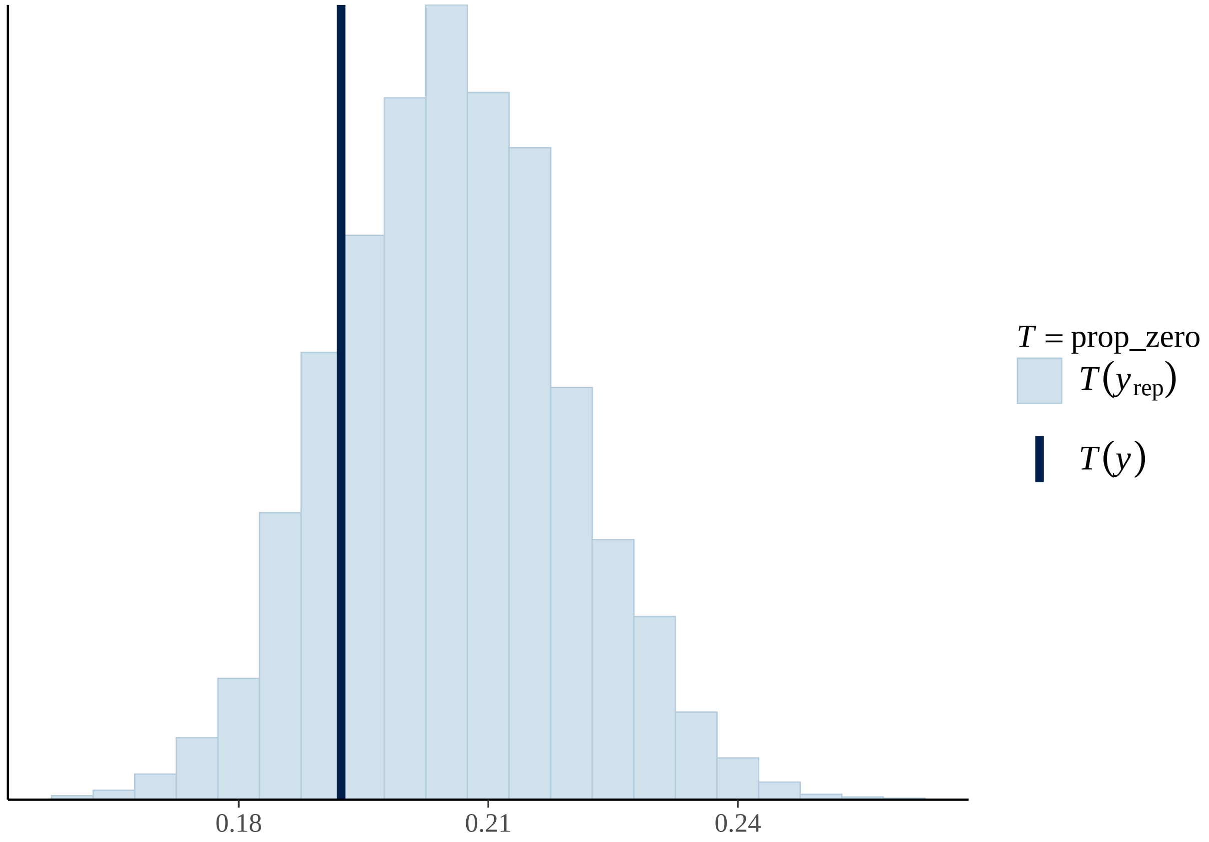
```
prop_zero <- function(x) mean(x == 0)
ppc_stat(y, y_zip, stat = "prop_zero", binwidth = 0.005)
```



```
ppc_stat(y, y_negbin, stat = "prop_zero", binwidth = 0.005)
```



```
ppc_stat(y, y_zinegbin, stat = "prop_zero", binwidth = 0.005)
```

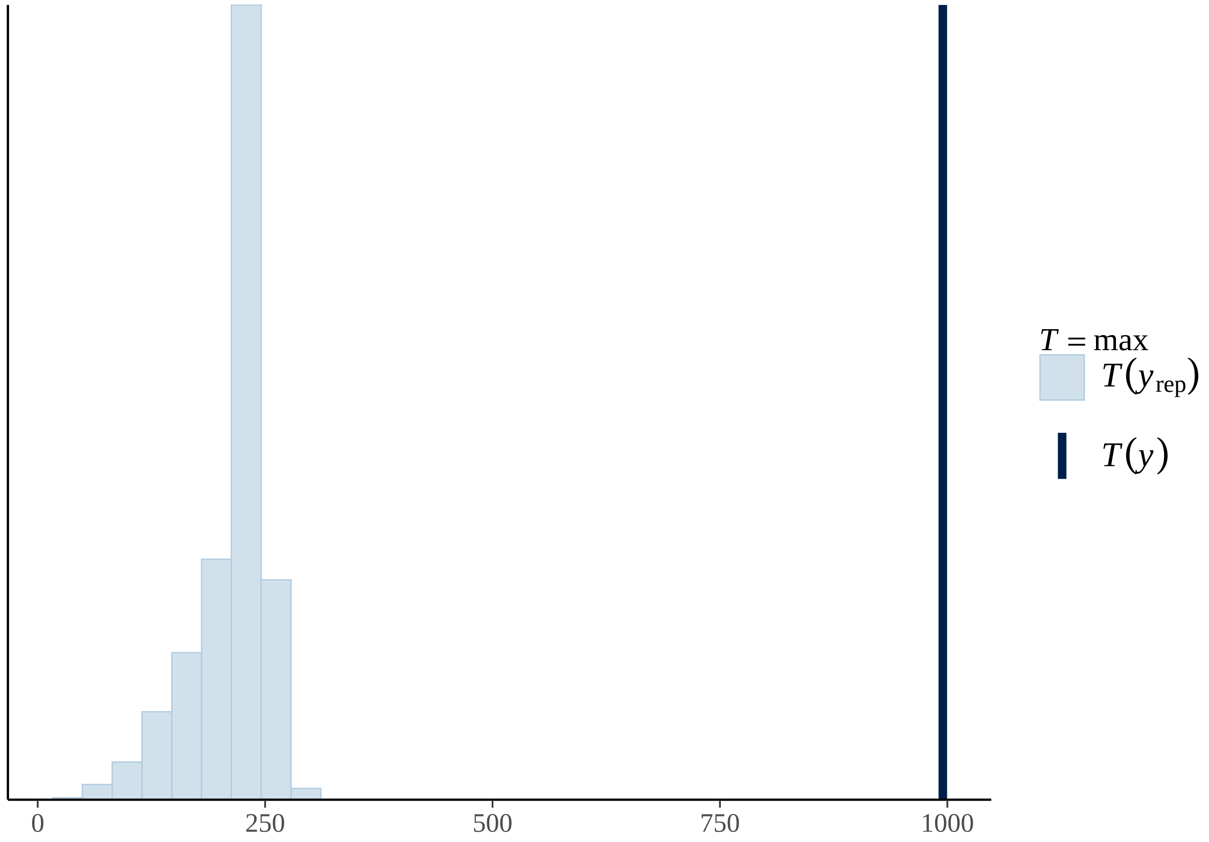


ability to generate max

Both negative binomial and zero-inflated negative binomial has some crazy outliers here, but looks pretty reasonable.

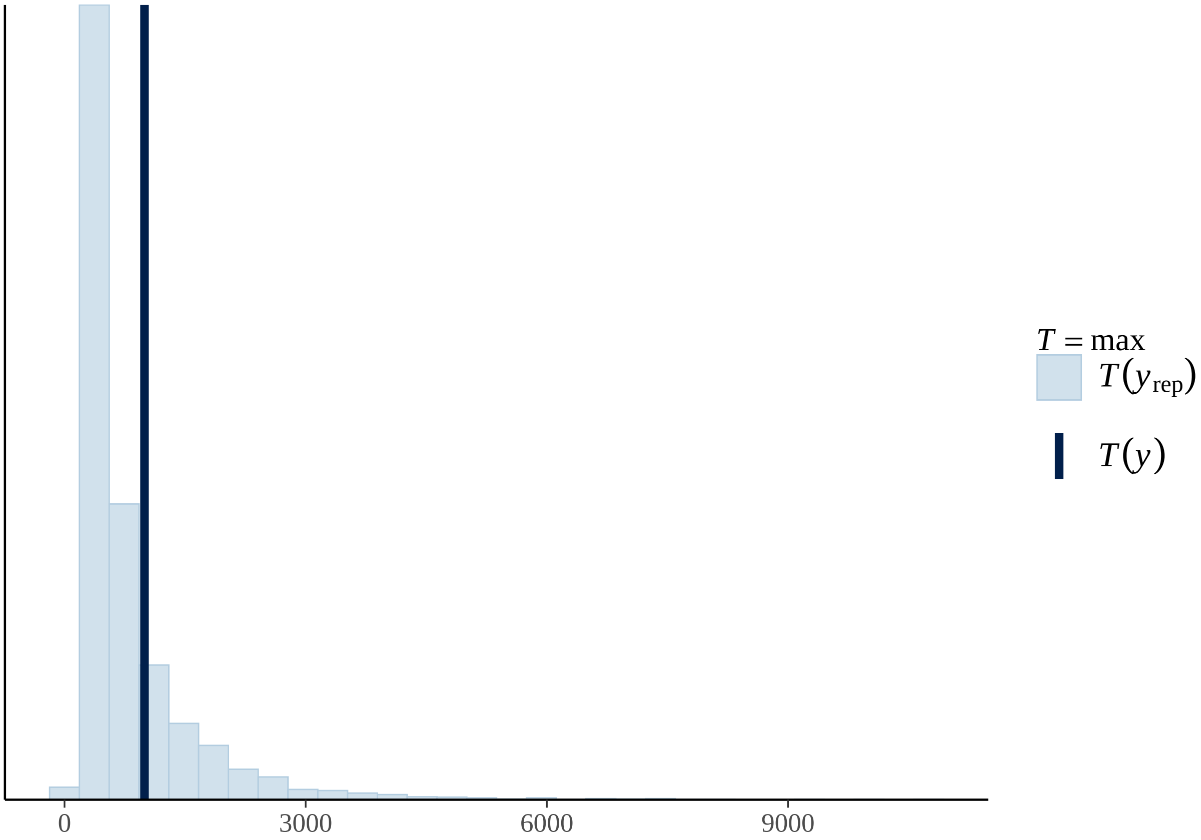
```
ppc_stat(y, y_zip, stat = "max")
```

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



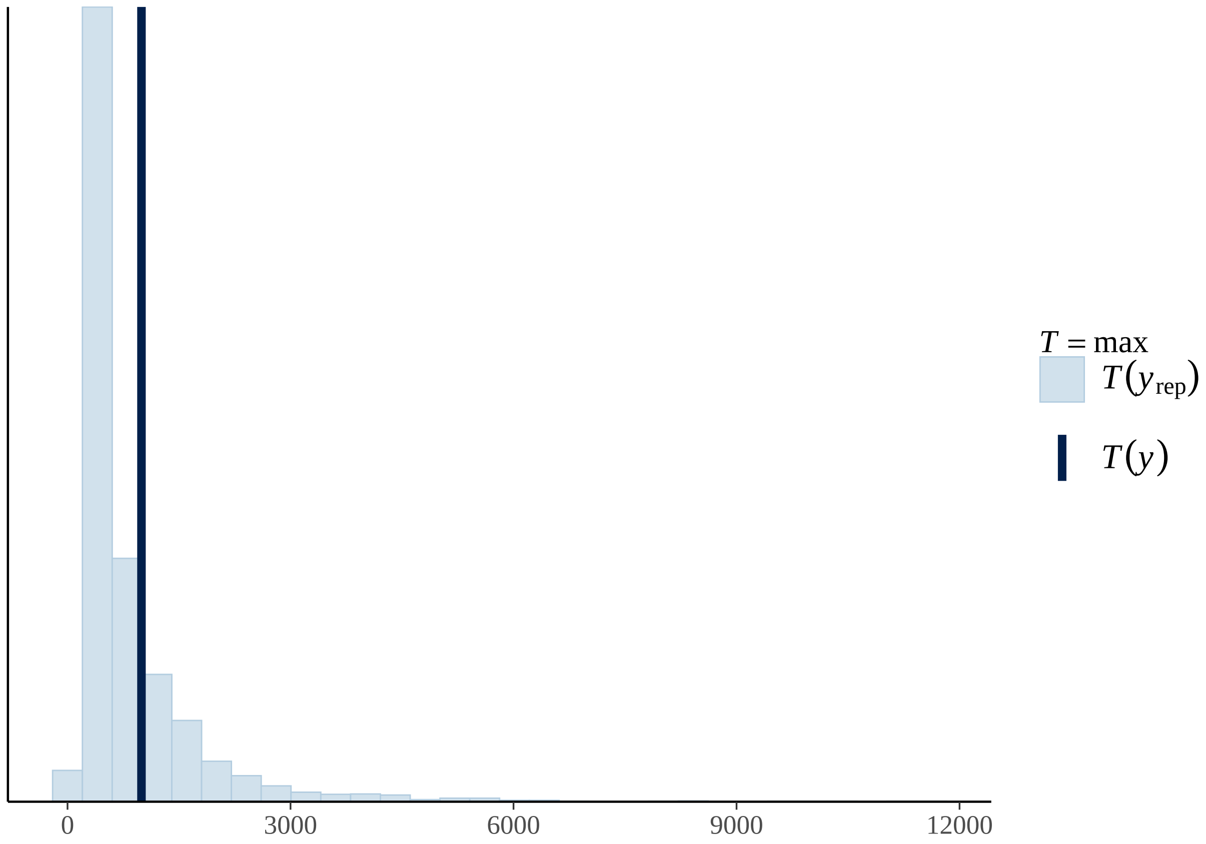
```r
ppc_stat(y, y_negbin, stat = "max")
```

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



```
ppc_stat(y, y_zinegbin, stat = "max")
```

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

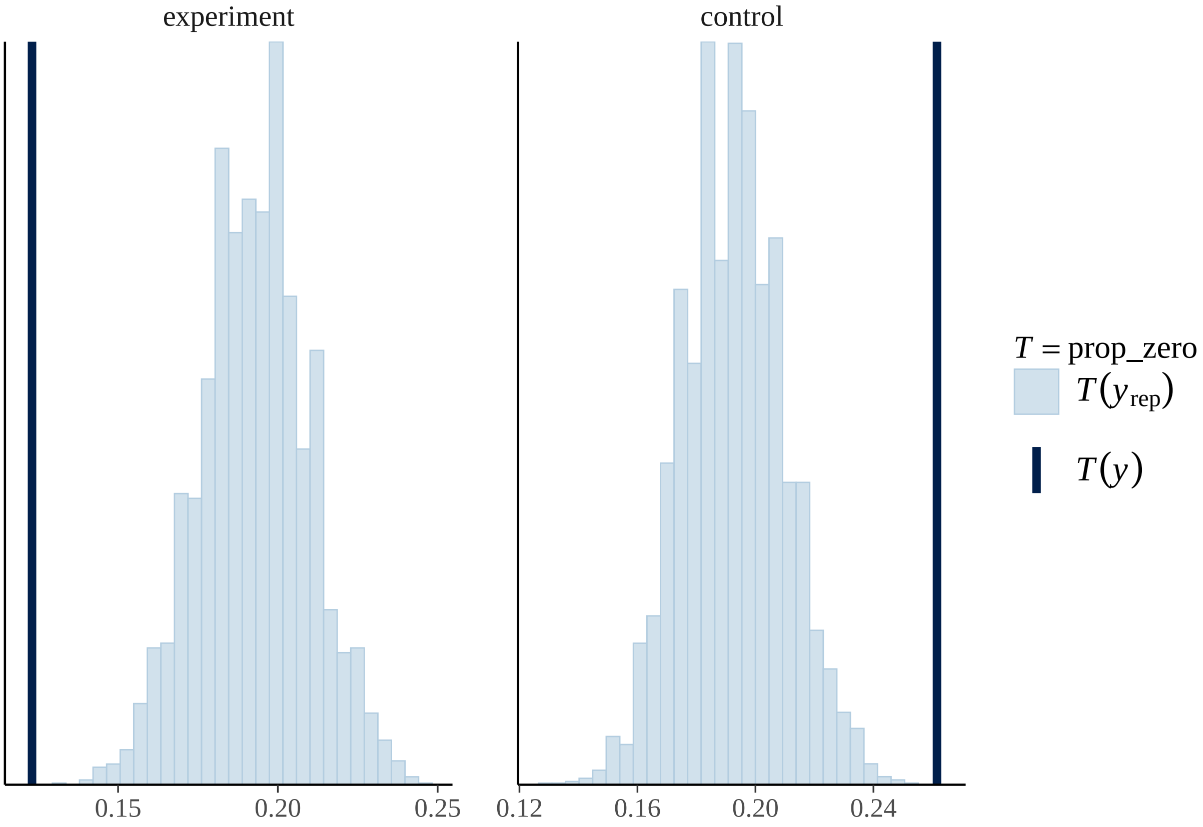


by group

They all predict “too much” of the same for each group (although might be good that they are more sceptical). Best model seems to be the negative binomial.

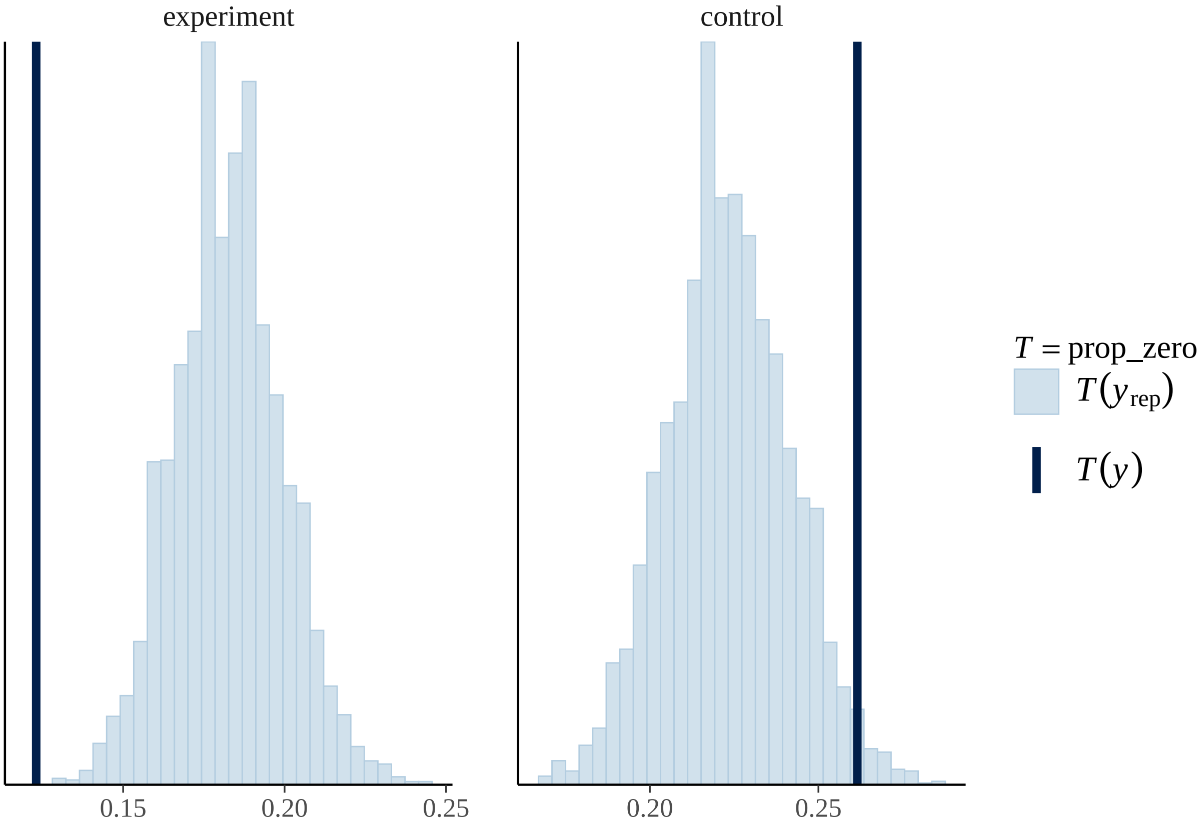
```
ppc_stat_grouped(y, y_zip, group = d$condition_fct, stat = "prop_zero") # predicts same for groups
```

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



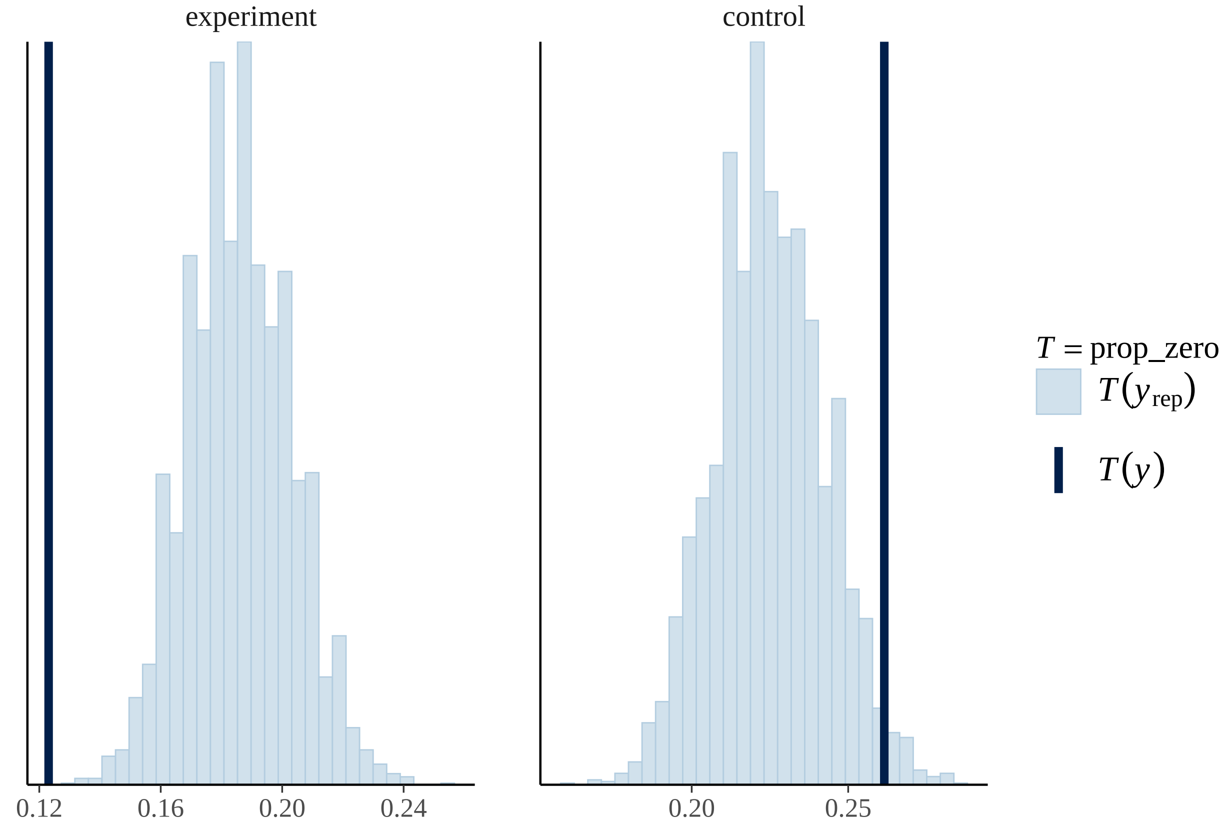
```
ppc_stat_grouped(y, y_negbin, group = d$condition_fct, stat = "prop_zero")
```

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



```
ppc_stat_grouped(y, y_zinegbin, group = d$condition_fct, stat = "prop_zero")
```

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



check summary

Only "interesting" thing here is that large teamsize seems to have a stronger effect for replication studies. Probably nothing (as we will see later) & definitely not significant. However, there has been a lot of attention on the large replication studies (e.g. ManyLabs).

```
summary(zip_post)
```

```
##  Family: zero_inflated_poisson
##   Links: mu = log; zi = identity
## Formula: c_5 ~ 0 + condition_fct + condition_fct:teamsize_scaled
##    Data: d (Number of observations: 1560)
##   Draws: 2 chains, each with iter = 2000; warmup = 0; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##                                         Estimate Est.Error l-95% CI u-95% CI
## condition_fctexperiment                     3.01      0.01     2.99     3.03
## condition_fctcontrol                        2.76      0.01     2.73     2.78
## condition_fctexperiment:teamsize_scaled     2.42      0.05     2.32     2.52
## condition_fctcontrol:teamsize_scaled        1.76      0.09     1.58     1.94
##                                         Rhat Bulk_ESS Tail_ESS
## condition_fctexperiment                 1.00     3193     2646
## condition_fctcontrol                    1.00     2683     2836
## condition_fctexperiment:teamsize_scaled 1.00     2886     2894
## condition_fctcontrol:teamsize_scaled    1.00     2685     2534
##
## Family Specific Parameters:
##    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## zi     0.19      0.01     0.17     0.21 1.00     3575     2546
##
## 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).
```

```
summary(negbin_post)
```

```
##  Family: negbinomial
##   Links: mu = log; shape = identity
## Formula: c_5 ~ 0 + condition_fct + condition_fct:teamsize_scaled
##    Data: d (Number of observations: 1560)
##   Draws: 2 chains, each with iter = 2000; warmup = 0; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##                                         Estimate Est.Error l-95% CI u-95% CI
## condition_fctexperiment                     2.83      0.07     2.70     2.96
## condition_fctcontrol                        2.35      0.07     2.22     2.49
## condition_fctexperiment:teamsize_scaled     2.95      0.66     1.67     4.23
## condition_fctcontrol:teamsize_scaled        3.19      0.75     1.73     4.67
##                                         Rhat Bulk_ESS Tail_ESS
## condition_fctexperiment                 1.00     2253     2354
## condition_fctcontrol                    1.00     2198     1986
## condition_fctexperiment:teamsize_scaled 1.00     2141     2418
## condition_fctcontrol:teamsize_scaled    1.00     2281     2578
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.45      0.02     0.42     0.48 1.00     2671     2495
##
## 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).
```

```
summary(zinegbin_post)
```

```
##  Family: zero_inflated_negbinomial
##   Links: mu = log; shape = identity; zi = identity
## Formula: c_5 ~ 0 + condition_fct + condition_fct:teamsize_scaled
##    Data: d (Number of observations: 1560)
##   Draws: 2 chains, each with iter = 2000; warmup = 0; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##                                         Estimate Est.Error l-95% CI u-95% CI
## condition_fctexperiment                     2.83      0.07     2.70     2.96
## condition_fctcontrol                        2.37      0.07     2.23     2.50
## condition_fctexperiment:teamsize_scaled     2.96      0.66     1.70     4.29
## condition_fctcontrol:teamsize_scaled        3.20      0.76     1.77     4.71
##                                         Rhat Bulk_ESS Tail_ESS
## condition_fctexperiment                 1.00     2080     2612
## condition_fctcontrol                    1.00     2659     2666
## condition_fctexperiment:teamsize_scaled 1.00     2012     2619
## condition_fctcontrol:teamsize_scaled    1.00     2697     2801
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## shape     0.46      0.02     0.43     0.50 1.00     3317     2879
## zi        0.01      0.01     0.00     0.03 1.00     2373     1547
##
## 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 model fit (focus on negative binomial)

notes

effect of interaction probably driven by the one outlier (more data and/or robust model?).

could also consider doing something else to teamsize (e.g. log) - would make sense that it is not a linear relationship.

also consider what McElreath does with forcing it through origin.

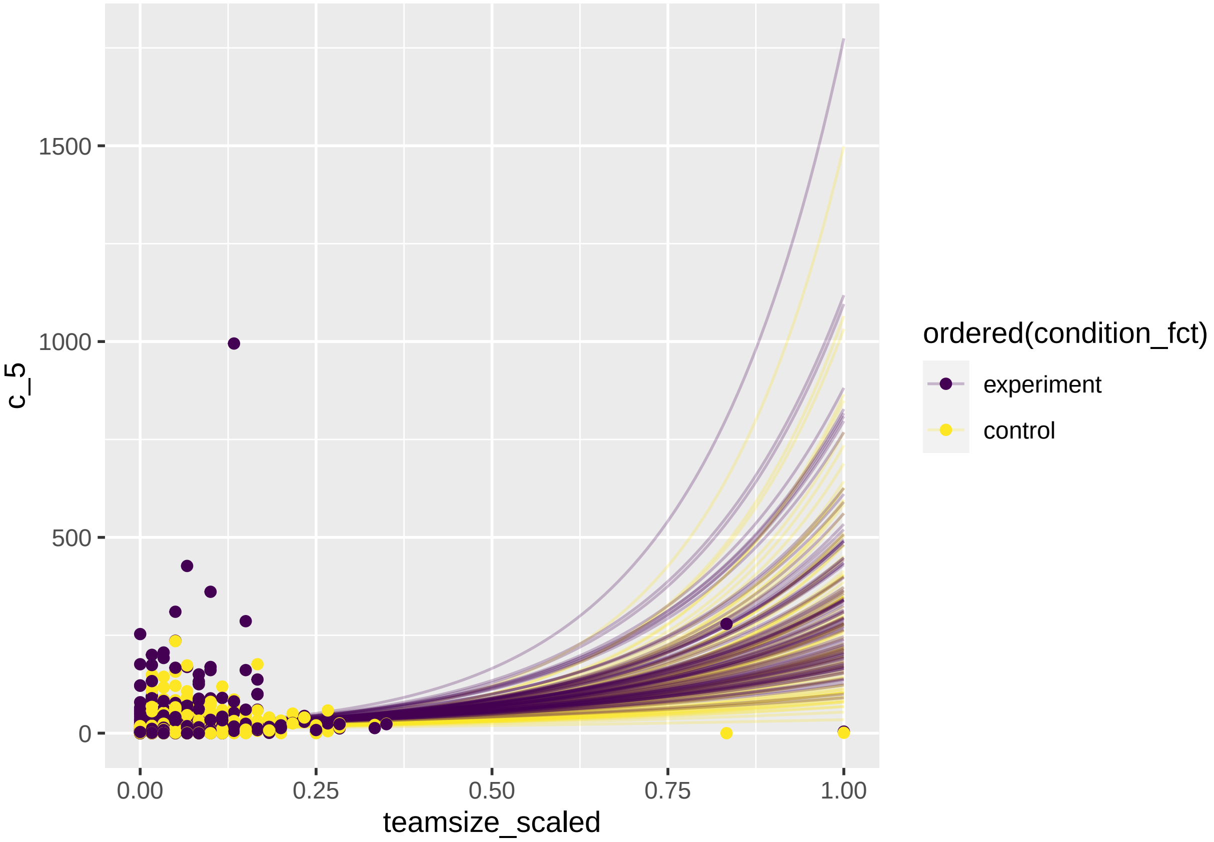
add_fitted_draws() is now add_epred_draws().

https://mjskay.github.io/tidybayes/reference/add_predicted_draws.html

fitted draws

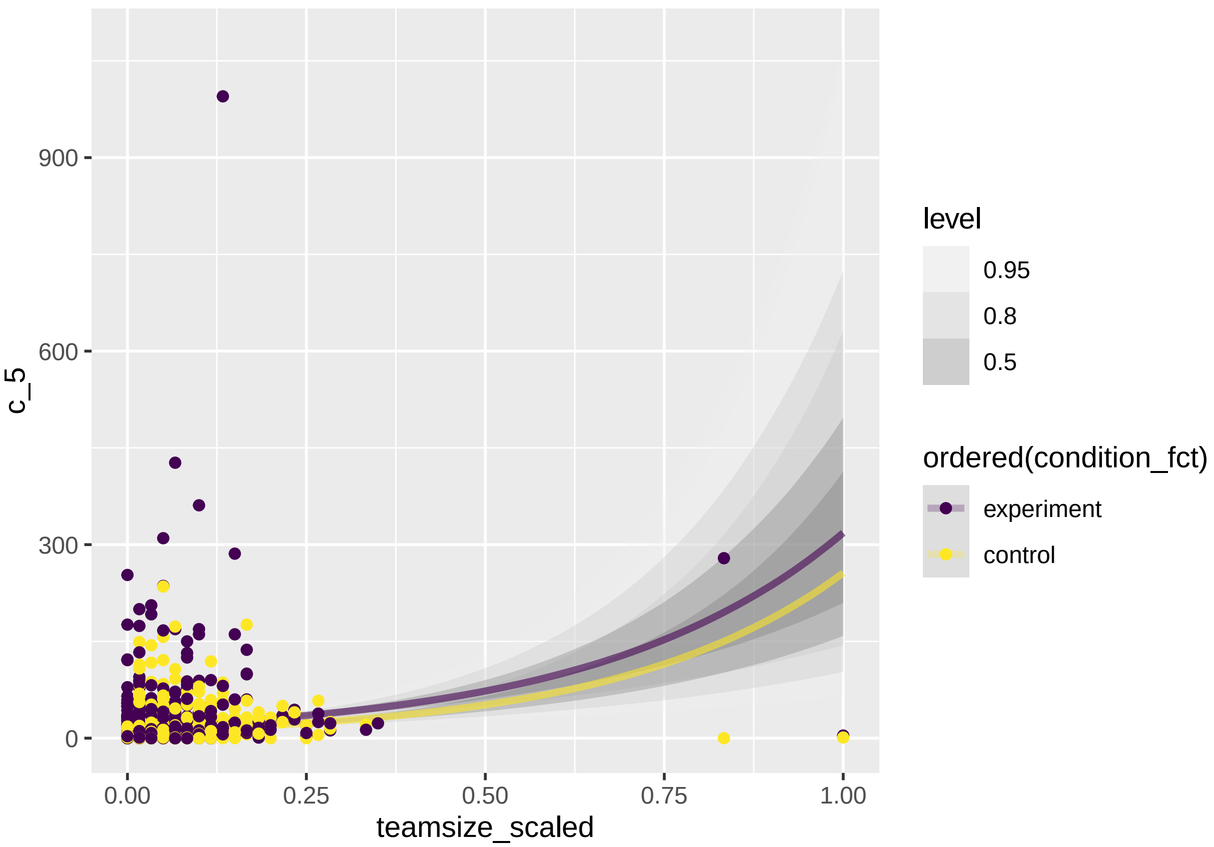
Clear that there is a lot of weight for two large studies, where the "control" is not cited & the experiment is cited a lot. Really seems to drive the (probably spurious) effect.

```
# with draws
d %>%
  group_by(condition_fct) %>%
  data_grid(teamsize_scaled = seq_range(teamsize_scaled, n = 101)) %>%
  add_epred_draws(negbin_post, ndraws = 100) %>% # same as fitted
  ggplot(aes(x = teamsize_scaled, y = c_5, color = ordered(condition_fct))) +
  geom_line(aes(y = .epred, group = paste(condition_fct, .draw)), alpha = 0.25) +
  geom_point(data = d)
```



fitted CI

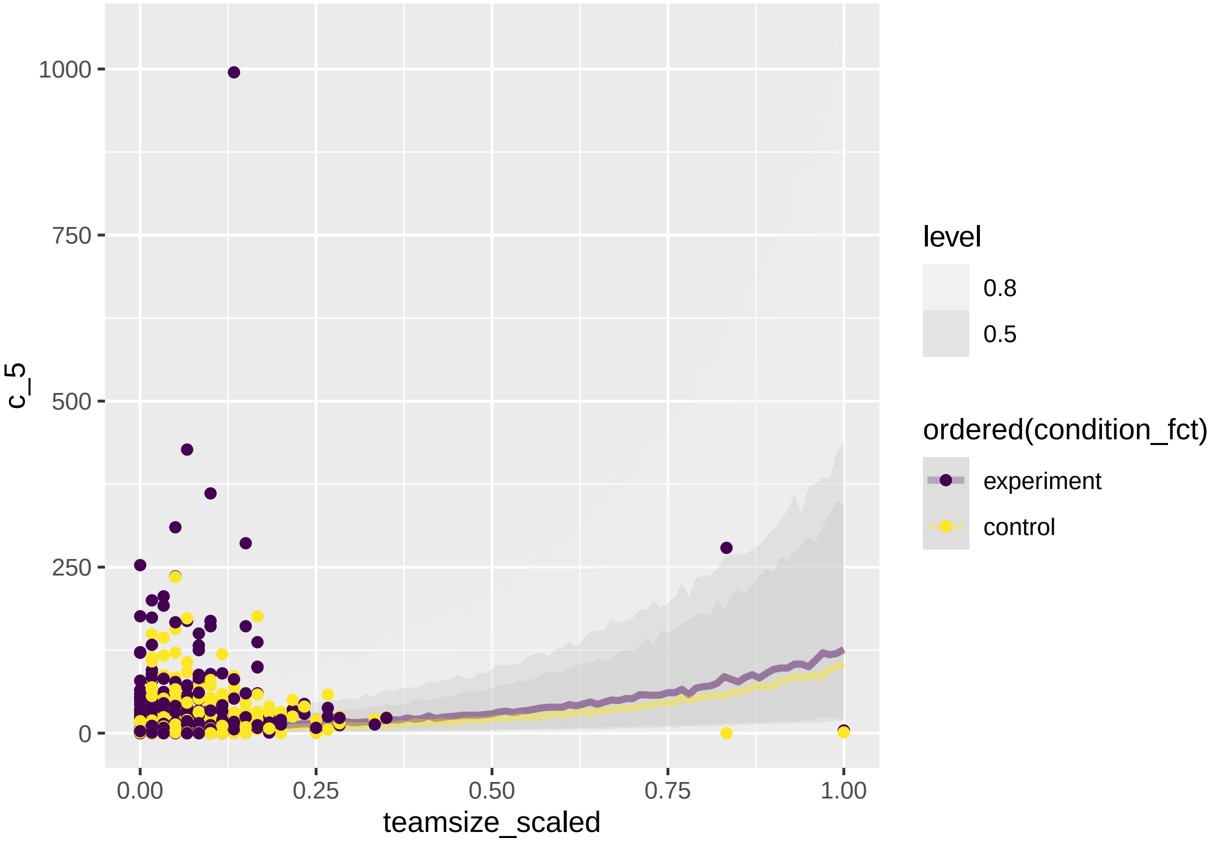
```
d %>%
  group_by(condition_fct) %>%
  data_grid(teamsize_scaled = seq_range(teamsize_scaled, n = 101)) %>%
  add_epred_draws(negbin_post) %>% # same as fitted
  ggplot(aes(x = teamsize_scaled, y = c_5, color = ordered(condition_fct))) +
  stat_lineribbon(aes(y = .epred), .width = c(.95, .8, .5), alpha = 0.25) +
  geom_point(data = d) +
  scale_fill_brewer(palette = "Greys")
```



predicted draws

Zero is the most common value for almost any value of teamsize & condition. Not super informative here. Perhaps more interesting with draws?

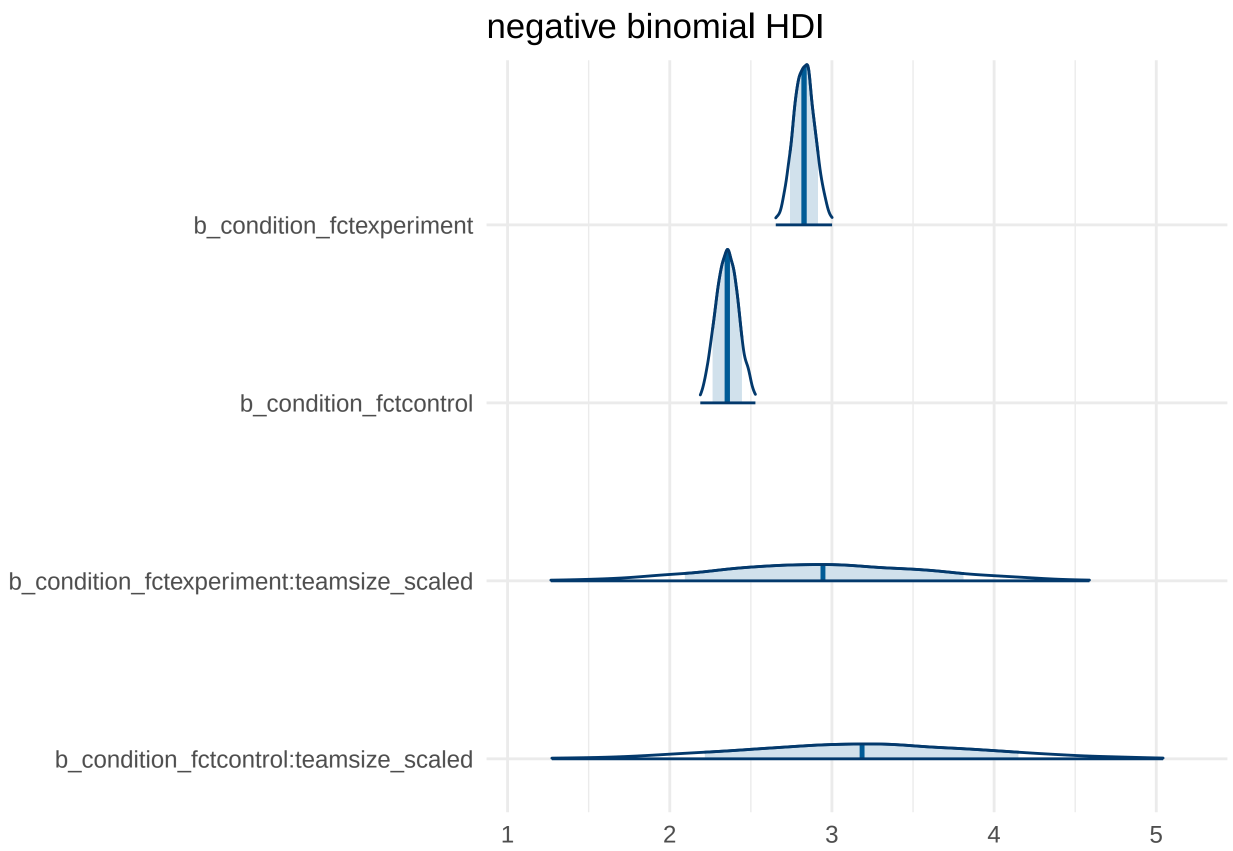
```
# does a poor job predicting (but makes sense that 0 is always the value to predict...)
d %>%
  group_by(condition_fct) %>%
  data_grid(teamsize_scaled = seq_range(teamsize_scaled, n = 101)) %>%
  add_predicted_draws(negbin_post) %>% # same as fitted
  ggplot(aes(x = teamsize_scaled, y = c_5, color = ordered(condition_fct))) +
  stat_lineribbon(aes(y = .prediction), .width = c(.8, .5), alpha = 0.25) +
  geom_point(data = d) +
  scale_fill_brewer(palette = "Greys")
```



HDI for parameters

```
mcmc_hdi <- function(fit, title){

  mcmc_areas(
    fit,
    pars = c("b_condition_fctexperiment",
             "b_condition_fctcontrol",
             "b_condition_fctexperiment:teamsize_scaled",
             "b_condition_fctcontrol:teamsize_scaled"),
    prob = 0.8, # 80% intervals
    prob_outer = 0.99, # 99%
    point_est = "mean") +
    theme_minimal() +
    ggtitle(title)

}
```

```
mcmc_hdi(negbin_post, "negative binomial HDI")
```



model comparison

consider criterion, i.e. waic, loo, bayes_R2 (I think the latter is for stacking weights?). best at the moment is actually the zero-inflated negative binomial, but very close to negative binomial. should probably keep both in mind moving forward (as they both looked reasonable).

```
# add criterions
zip_post <- add_criterion(zip_post,
                          criterion = c("loo", "bayes_R2"))

negbin_post <- add_criterion(negbin_post,
                             criterion = c("loo", "bayes_R2"))

zinegbin_post <- add_criterion(zinegbin_post,
                               criterion = c("loo", "bayes_R2")) # one problematic observation.
# run loo compare
loo_compare(zip_post,
            negbin_post,
            zinegbin_post)
```

```
##                elpd_diff se_diff
## negbin_post        0.0       0.0
## zinegbin_post     -1.8       0.6
## zip_post      -19106.5    3067.5
```

```
# also uses stacking (and gives the same as pyMC3)
loo_model_weights(zip_post,
                  negbin_post,
                  zinegbin_post)
```

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

```
## Method: stacking
## ------
##               weight
## zip_post      0.007
## negbin_post   0.993
## zinegbin_post 0.000
```

Check influential observations (pareto k)

check recoded (of rethinking)