

# Visualize Piecewise Regression with Negative Binomial Type I Error on Real Data using **brms** Custom Family

Michael Gilchrist

Created: 2023-03-29; Compiled: Thu Apr 6 18:23:44 2023

## Goal

- Visualize two piece negative binomial type 1 formulation to data

## Recap

- Update to 2023-03-09\_visualize.2023-02-28-fit... for use with more recent fits in 2023-03-18\_fit... folder.
- Part of challenge is bimodal nature of some male's x0 and y0 fits
  - T235 and T236 in particular

## Insights

## Set up

### Install libraries

```
# install packages user might not have by replacing FALSE with TRUE

## load libraries
library(stats)
library(MASS) # provides negative binomial fitting: glm.nb
library(ggplot2)
library(ggpubr)
library(grid)
library(gridExtra)
library(ggExtra)
library(cowplot)
library(GGally)
library(RColorBrewer) ## needed to have more than 8 colors with `palette="Set2"
library(broom)
library(tidyverse)
library(viridisLite)
library(cmdstanr)
library(rstan)
```

```

options(mc.cores = (parallel::detectCores()-2))
rstan_options(auto_write = TRUE)
library(brms)
library(bayesplot)
library(tidybayes)
library(loo)
library(modelr)

## options(ggplot2.continuous.colour="viridis",
##         ggplot2.discrete.colour="viridis",
##         ggplot2.scale_fill_discrete = scale_fill_viridis_d,
##         ggplot2.scale_fill_continuous = scale_fill_viridis_c)

library(reshape2)
library(latex2exp)

ggplot2::theme_set(theme_default(base_size = 10))
## Restore theme settings
## ggplot2::theme_set(theme_default())

n_cores <- 4

```

## Source family

```

source("../.../custom-brms-families/families/nbinom_type1.R")

## Define fmax() or case expose_functions()

fmax <- function(x, x0) max(x, x0)

```

## Load custom family

### Load input

```

load(file.path("input", "fit_tbl.Rda"), verbose = TRUE)

## Loading objects:
##   fit_tbl

fit_tbl_orig <- fit_tbl

fit_tbl <- fit_tbl %>% filter(!is.na(fit)))

```

## Define local functions

```

which_tbl_row <- function(filter_male = FALSE, x0_flag = "individual", y0_flag = "individual", disp_flag =
  which( tbl$filter_male %in% filter_male &
    tbl$x0_flag %in% x0_flag &
    tbl$y0_flag %in% y0_flag &
    tbl$disp_flag %in% disp_flag &
    tbl$disp_value %in% disp_value &
    tbl$model %in% model
    #tbl$sampling_dist %in% sampling &
  )
}

clean_var_names <- function(fit) {
  fit %>% setNames(gsub("b_", "", names(.)) %>%
    gsub("(x0|s0|y0)_male(T[0-9]{3})", "\\\2_\\1", .) %>%
    gsub("__", "_", .) %>%
    gsub("r_male_(x0|s0|y0)\\[(T[0-9]{3}),Intercept\\]", "\\\2_\\1_r", .) %>%
    gsub("\\.", " ", .))
}

```

## Visualize Model Fits

Set up functions, parameters, and results tibble

```

xmax <- 45.5

## Here I create a variable using a string and then access it using a string
for(name in names(fit_tbl)[1:8]) {
  # create
  var <- paste0(name, "_vec")
  assign(var, pull(fit_tbl, name) %>% unique())
  # access via get()
  print(paste0(var, ": ", paste0(get(var), sep = "\n\t")))
}

## [1] "model_vec: two_piece\n\t"
## [1] "x0_flag_vec: groups_1\n\t"    "x0_flag_vec: individual\n\t"
## [3] "x0_flag_vec: uniform_1\n\t"
## [1] "y0_flag_vec: individual\n\t"
## [1] "disp_value_vec: 0.01\n\t"
## [1] "disp_flag_vec: uniform_1\n\t"
## [1] "desc_vec: nbinom_type1; two_piece; x0: groups_1; y0: individual; disp_flag: uniform_1; disp pri
## [2] "desc_vec: nbinom_type1; two_piece; x0: groups_1; y0: individual; disp_flag: uniform_1; disp pri
## [3] "desc_vec: nbinom_type1; two_piece; x0: individual; y0: individual; disp_flag: uniform_1; disp pri
## [4] "desc_vec: nbinom_type1; two_piece; x0: individual; y0: individual; disp_flag: uniform_1; disp pri
## [5] "desc_vec: nbinom_type1; two_piece; x0: uniform_1; y0: individual; disp_flag: uniform_1; disp pri
## [6] "desc_vec: nbinom_type1; two_piece; x0: uniform_1; y0: individual; disp_flag: uniform_1; disp pri
## [1] "filter_male_vec: FALSE\n\t" "filter_male_vec: TRUE\n\t"
## [1] "x0_group_list_vec: NA\n\t"

```

## Print fits

```
pairs_include_lp = TRUE

for(row_index in 1:nrow(fit_tbl)) {

  print(paste("Row: ", row_index))

  row_values <- fit_tbl[row_index, ]
  desc <- row_values$desc
  filename_desc <- gsub("_", "-", desc) %>%
    gsub(";", "_", .) %>%
    gsub(":?", "-", .)
  fit_brms <- row_values$fit[[1]]
  data <- fit_brms[["data"]]
  fit_stan <- fit_brms$fit

  print_mcmc_pairs <- TRUE # individual level parameters
  print_mcmc_pairs_group <- TRUE # lp and group level parameters
  print_mcmc_scatter <- TRUE # individual level parameters, histograms printed on margins
  print_stan_hist <- FALSE

  print_get_prior <- TRUE ## reset value
  print_prior_summary <- TRUE

  desc_filename <- gsub("_", "-", desc) %>%
    gsub(";", "_", .) %>%
    gsub(":?", "-", .)
  desc_short <- desc %>% str_replace("nbinom_type1; two_piece; ","")

  title_row <- ggdraw() + draw_label(desc_short, fontface='bold', size = 12)

  ## Print and plot results, regardless of which fits one uses
  print(desc)
  # print(filename_desc)

  if(print_prior_summary) {
    print("Fit Prior Information")
    print(prior_summary(fit_brms)) # %>% filter(nlpar!="y0"))
    print_prior_summary <- TRUE
  }

  print("Fit Information")
  print(summary(fit_brms) ) # %>% gsub("disp_value", row_values[["disp_value"]]), .) #, pars = "x0*"))

  #clean up variable names
  fit_stan_rename <-
    fit_stan %>%
    clean_var_names()

##
```

```

vars_clean <- names(fit_stan_rename) %>% na.omit()

male_vec <- unique(data$male) %>% as.character()
## get male specific vars (start with "T")
vars_T <- grep("T[0-9]{3}", vars_clean, value = TRUE)
vars_Intercept <- grep("Intercept", vars_clean, value = TRUE)
vars_non_T <- vars_clean[!(vars_clean %in% c(vars_T, vars_Intercept))]
#print(vars_clean)

## Examine priors and exclude any constants from vars_non_T
priors_tmp <- prior_summary(fit_brms)
constants_non_T <- ggdist::parse_dist(priors_tmp) %>% filter(grepl("constant", .dist)) %>% pull(class)
vars_non_T <- vars_non_T[!(vars_non_T %in% constants_non_T)]

## Count occurrence of each male in model fit_brms
male_instance <- sapply(male_vec, function(x) {sum(str_detect(x, string=vars_clean))})

## ensure we can make a pairs plot for each male
if(all(unique(male_instance) == 2)) {

  if(print_mcmc_pairs) {
    ## Use mcmc_pairs
    pairs_list <- list()
    ## update panel.boarder
    ## Note brms uses 'variable' while stanfit uses 'pars'
    for(male in male_vec) {
      # print(male)
      vars_male <- grep(male, vars_T, value = TRUE)

      ## Generate pairs() plot using mcmc_pairs()
      ## Don't use pairs.stanfit which does not return an object
      pairs_pars <- vars_male
      if(pairs_include_lp) {
        pairs_pars <- c(pairs_pars, "lp_")
      }

      ## Change text size
      bayesplot_theme_update(text = element_text(size = 6), family = "sans")
      pairs_tmp <- mcmc_pairs(
        fit_stan_rename,
        pars = pairs_pars,
        ## set point size and transparency
        # grid_args = list(text = element_text(size = 20)),
        ## Try to use hex densities
        off_diag_fun = "hex"
      )

      bayesplot_theme_set() # same as bayesplot_theme_set(theme_default())

      # add density curves to off diagonal
      plots_tmp <- pairs_tmp$bayesplots

```

```

    if(length(plots_tmp) == 4) {
      plot_index <- c(2,3)
    }
    if(length(plots_tmp) == 9) {
      plot_index <- c(2, 3, 4, 6, 7, 8)
    }

    for(i in plot_index) {
      # pars = c(vars_male): list of 4 ggplot objects diagonal plots are 2 and 3
      plots_tmp[[i]] <- plots_tmp[[2]] +
        stat_density_2d(color = "black",
                        size = .25,
                        bins = 5)
    }

    pairs_tmp <- bayesplot_grid(plots = plots_tmp) ## Change font size
    pairs_list[[male]] <- pairs_tmp
  }

  p <- plot_grid(plotlist = pairs_list,
                 ncol = 2)
  plot_pairs_list <- plot_grid(title_row, p, ncol = 1,
                                 rel_heights=c(0.06, 1))

  print(plot_pairs_list)
  filename <- paste0("plot-pairs_", filename_desc, ".png")
  ggsave(filename = filename, path = file.path(output_dir, "figures"),
         width = 8, height = 11, units = "in",
         scale = 1,
         dpi=300,
         bg = "white")
}

if(print_mcmc_scatter) {

  scatter_list <- list()
  for(male in male_vec) {
    # print(male)
    vars_male <- grep(male, vars_T, value = TRUE)

    scatter_tmp <- mcmc_scatter( #was mcmc_scatter
      fit_stan_rename,
      pars = c(vars_male) #, vars_Intercept),
      #off_diag_args = list(size = 1, alpha = 0.35), # only used if `off_diag_fun = "scatter"`
      #off_diag_fun = c("hex")
    ) #+
    # stat_density_2d(color = "lightgray", size = .5, bins = 5) # doesn't work with mcmc_pairs

    ## ggMarginal doesn't work natively with mcmc_hex, so we need to make the
    ## points transparent and then add a hex layer
    scatter_list[[male]] <- ggExtra::ggMarginal(scatter_tmp +
      geom_point(col="transparent") +

```

```

        geom_hex() +
        theme(legend.position = "none"), type = "histogram")

    }

p <- plot_grid(plotlist = scatter_list,
               ncol = 3)

plot_scatter_list <- plot_grid(title_row, p, ncol = 1, rel_heights=c(0.1, 1))

print(plot_scatter_list)

filename <- paste0("plot-scatter_", filename_desc, ".png")
ggsave(filename = filename, path = file.path(output_dir, "figures"),
       width = 8, height = 11, units = "in",
       scale = 1,
       dpi=300,
       bg = "white")

} # end if(print_mcmc_scatter)
} else {} # end if(males == 2

if(print_mcmc_pairs_group) {
  pairs_list2 <- list()

  if(length(vars_Intercept) > 0) {
    pairs_list2[["Intercept"]] <- mcmc_pairs(
      fit_stan_rename, pars = c(vars_Intercept, "lp_"),
      off_diag_fun = c("hex"))
  }

  pairs_list2[["non-T"]] <- mcmc_pairs(fit_stan_rename,
                                         pars = vars_non_T,
                                         off_diag_fun = c("hex"))

}

p <- plot_grid(plotlist = pairs_list2,
               ncol = 1)
plot_pairs_list2 <- plot_grid(title_row, p, ncol = 1, rel_heights=c(0.1, 1))

print(plot_pairs_list2)

filename <- paste0("plot-pairs2_", filename_desc, ".png")
ggsave(filename = filename, path = file.path(output_dir, "figures"),
       width = 8, height = 11, units = "in",
       scale = 1,
       dpi=300,
       bg = "white")

}

```

```

vars_tmp <- vars_clean %>% str_subset("y0");

stan_plot(fit_stan_rename, pars = vars_tmp) +
  ggtitle("Initial Motif Counts", subtitle = desc)

vars_tmp <- vars_clean %>% str_subset("T[0-9]+_x0")
if(length(vars_tmp) > 0) {
  fit_stan <- stan_plot(fit_stan_rename, pars = vars_tmp) +
    ggtitle("Thresholds", subtitle = desc_short)

  filename <- paste0("fit-stan_", filename_desc, ".pdf")
  ggsave(filename = filename, path = file.path(output_dir, "figures"), dpi=300)
}

if(print_stan_hist) {
  ncol <- 4
  hist <- stan_hist(fit_stan_rename,
                     pars = vars_fit,
                     bins = 25,
                     ncol = ncol) +
    ggtitle(desc_short)
  filename <- paste0("histogram_", filename_desc, ".pdf")
  ggsave(filename = filename, path = file.path(output_dir, "figures"), dpi=300)
}
}

## [1] "Row: 1"
## [1] "nbinom_type1; two_piece; x0: groups_1; y0: individual; disp_flag: uniform_1; disp prior: 0.01; "
## [1] "Fit Prior Information"
##          prior class      coef group resp dpar nlpar lb    ub
##          uniform(32, 44.5)   b             x0 32 44.5
##          uniform(32, 44.5)   b Intercept   x0 32 44.5
##          uniform(10, 1000)   b             y0 10 1000
##          uniform(10, 1000)   b maleT234   y0 10 1000
##          uniform(10, 1000)   b maleT235   y0 10 1000
##          uniform(10, 1000)   b maleT236   y0 10 1000
##          uniform(10, 1000)   b maleT237   y0 10 1000
##          uniform(10, 1000)   b maleT243   y0 10 1000
##          uniform(10, 1000)   b maleT244   y0 10 1000
##          uniform(10, 1000)   b maleT246   y0 10 1000
##          uniform(10, 1000)   b maleT247   y0 10 1000
##          uniform(10, 1000)   b maleT257   y0 10 1000
##          uniform(10, 1000)   b maleT258   y0 10 1000
##          uniform(10, 1000)   b maleT260   y0 10 1000
## exponential(disp_value) disp           0    20
## student_t(3, 0, 66.7)   sd             x0 0 10
## student_t(3, 0, 66.7)   sd male        x0 0 10
## student_t(3, 0, 66.7)   sd Intercept male x0 0 10
##          source
##          user
## (vectorized)
##          user

```

```

##  (vectorized)
##  user
##  user
##  (vectorized)
##  (vectorized)
## [1] "Fit Information"

## Warning: Parts of the model have not converged (some Rhats are > 1.05). Be
## careful when analysing the results! We recommend running more iterations and/or
## setting stronger priors.

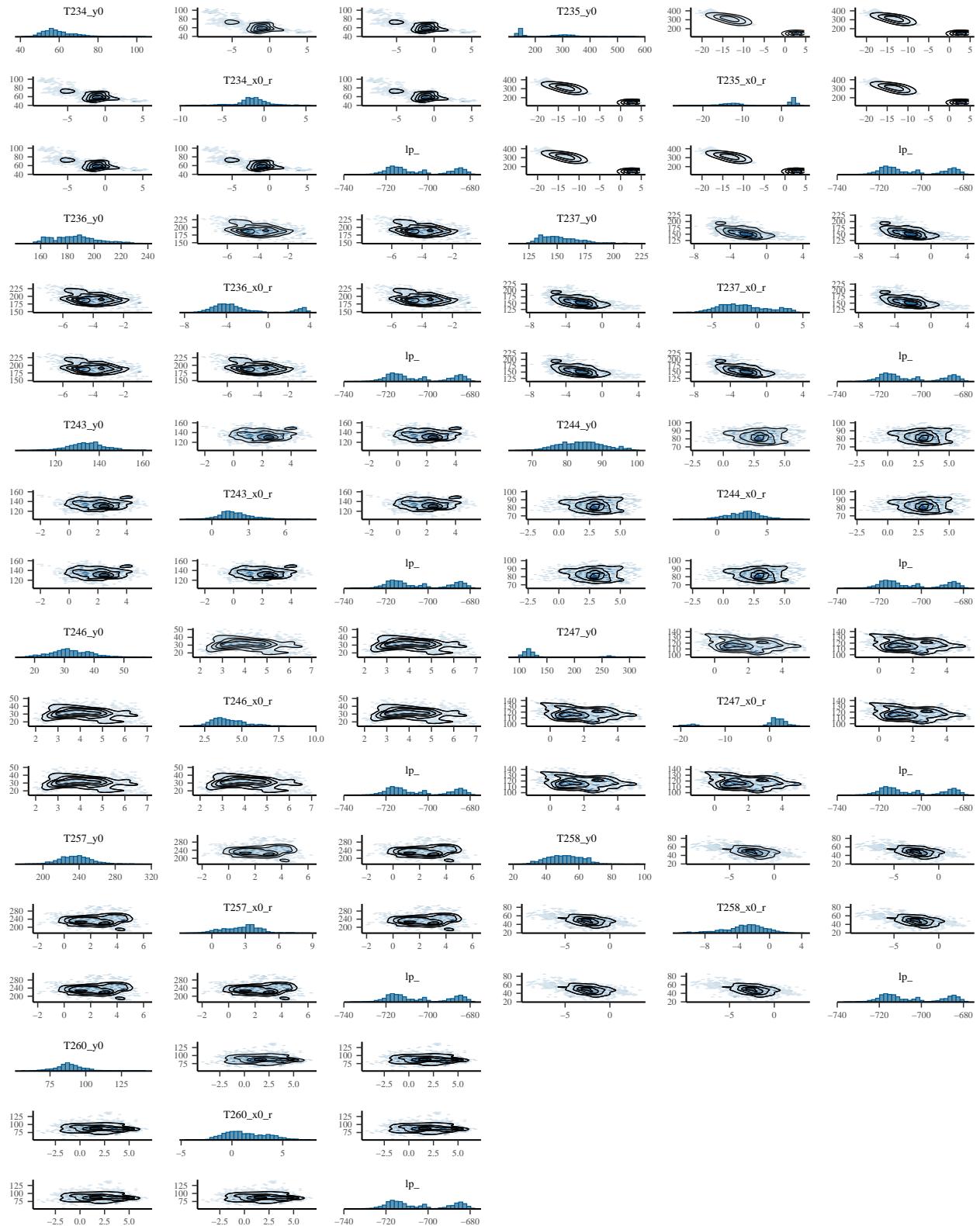
## Warning: There were 2995 divergent transitions after warmup. Increasing
## adapt_delta above 0.99 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Family: nbinom_type1
##   Links: mu = identity; disp = identity
## Formula: y ~ two_piece(x, x0, y0)
##           x0 ~ 0 + Intercept + (1 | male)
##           y0 ~ 0 + male
## Data: data (Number of observations: 107)
## Draws: 6 chains, each with iter = 5000; warmup = 3000; thin = 4;
##        total post-warmup draws = 3000
##
## Group-Level Effects:
## ~male (Number of levels: 11)
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(x0_Intercept)    5.58      1.99     2.39    9.22 1.88       9      32
##
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## x0_Intercept    41.60      1.25    38.51    43.58 1.70      10      20
## y0_maleT234    60.21      9.15    48.04    81.69 1.26      17      35
## y0_maleT235   231.17     96.09   133.01   412.91 1.86       9      37
## y0_maleT236   185.83     17.36   158.21   224.22 1.39      14      18
## y0_maleT237   153.58     18.12   128.05   198.66 1.26      17      34
## y0_maleT243   134.36      8.61   116.16   151.63 1.07      67      60
## y0_maleT244    84.13      6.44    72.35    97.01 1.17      27      72
## y0_maleT246    32.19      6.86    19.46    46.79 1.15      31      74
## y0_maleT247   142.46     58.36   101.87   297.94 1.47      12      11
## y0_maleT257   236.23     19.36   199.58   279.62 1.08      80     109
## y0_maleT258    51.53     10.62    32.11    70.97 1.21      23     181
## y0_maleT260    88.73      9.93    69.26   109.63 1.08      83     233

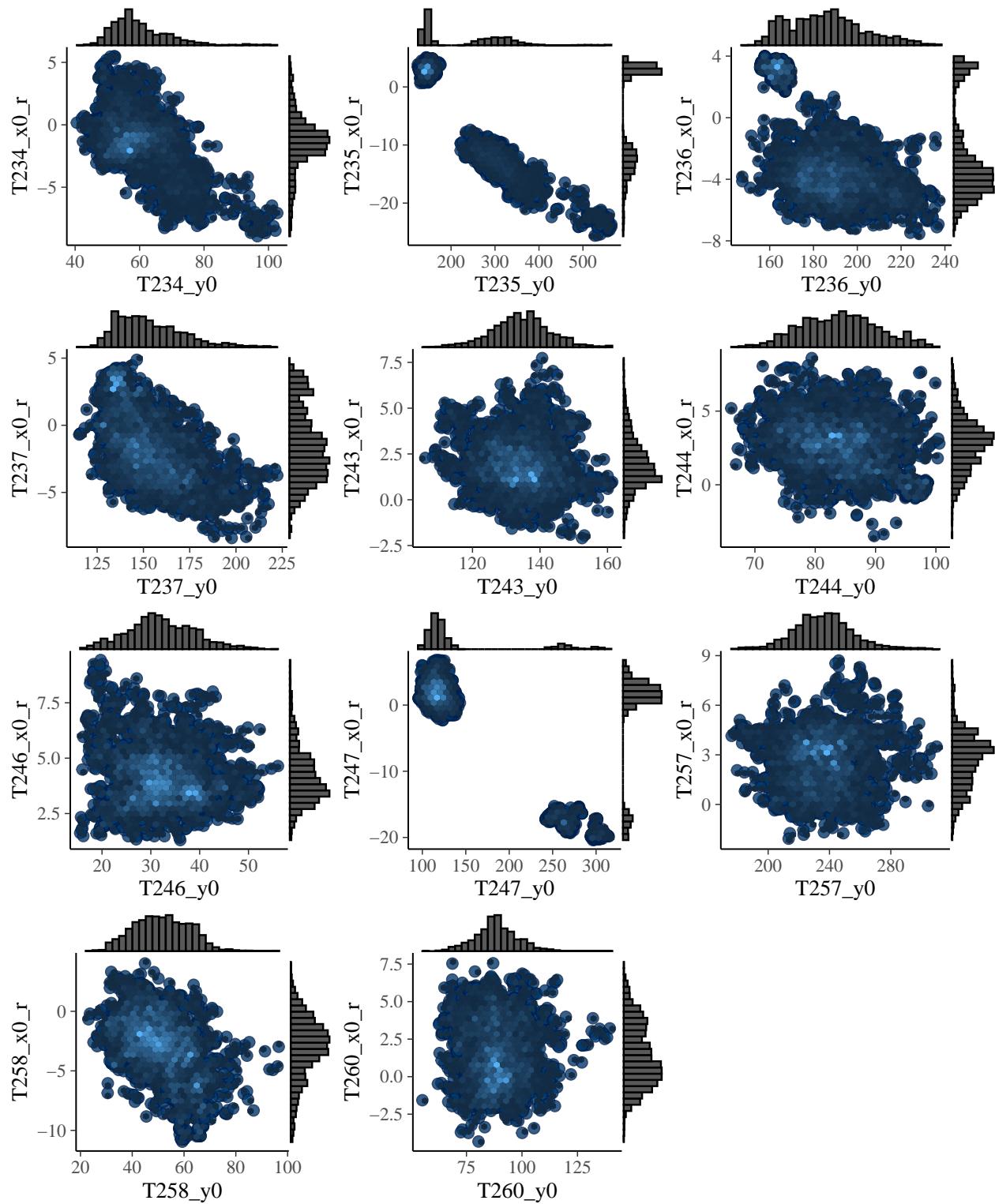
```

```
##  
## Family Specific Parameters:  
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS  
## disp      6.08      0.60     4.83    7.14 1.35       15       81  
##  
## 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).
```

**x0: groups\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE**



x0: groups\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE



```
## [1] "Row: 2"
## [1] "nbinom_type1; two_piece; x0: groups_1; y0: individual; disp_flag: uniform_1; disp prior: 0.01; "
## [1] "Fit Prior Information"
```

```

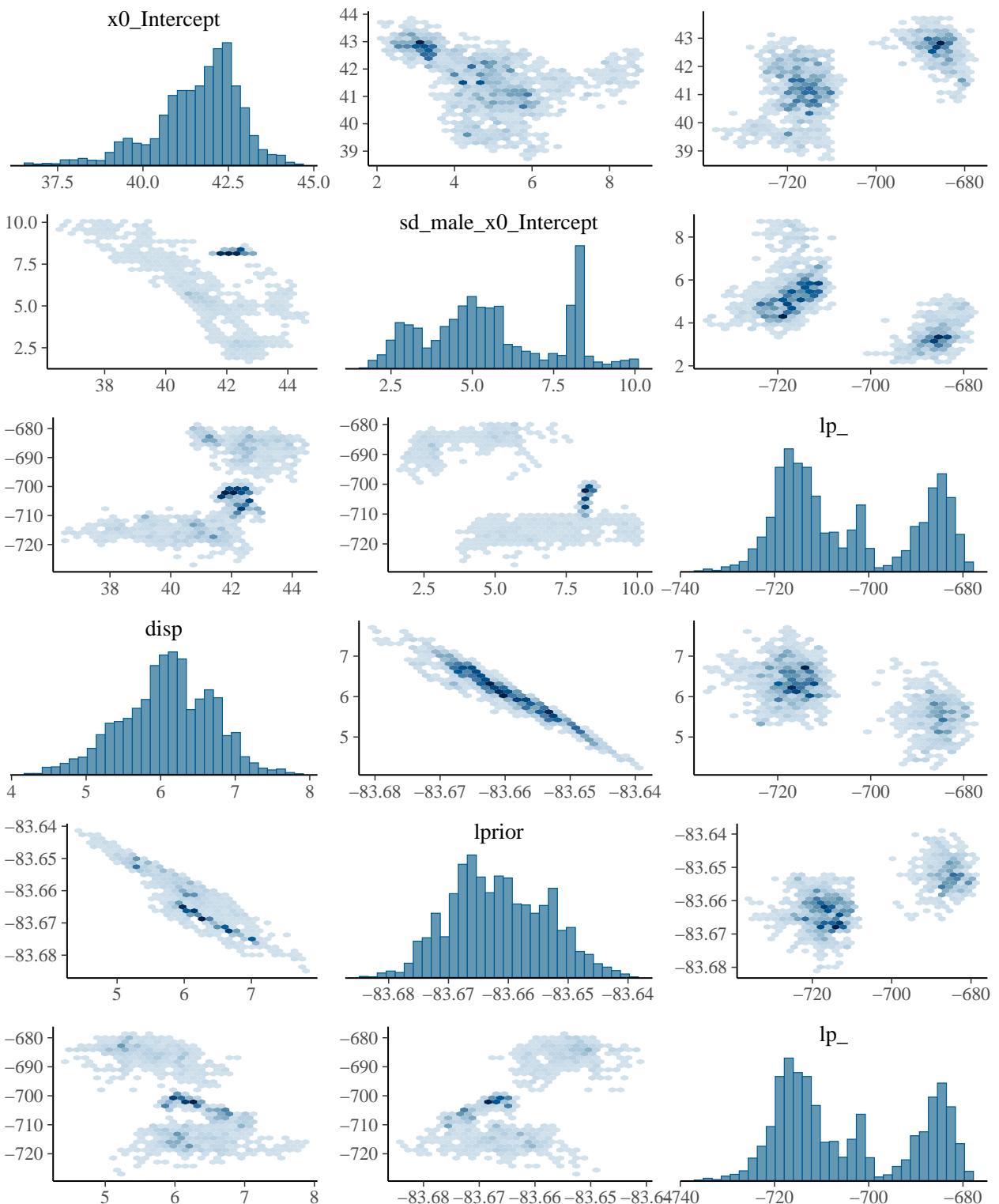
##          prior class      coef group resp dpar nelpar lb    ub
## uniform(32, 44.5)     b             x0 32 44.5
## uniform(32, 44.5)     b Intercept   x0 32 44.5
## uniform(10, 1000)     b             y0 10 1000
## uniform(10, 1000)     b maleT234   y0 10 1000
## uniform(10, 1000)     b maleT237   y0 10 1000
## uniform(10, 1000)     b maleT243   y0 10 1000
## uniform(10, 1000)     b maleT244   y0 10 1000
## uniform(10, 1000)     b maleT246   y0 10 1000
## uniform(10, 1000)     b maleT247   y0 10 1000
## uniform(10, 1000)     b maleT257   y0 10 1000
## uniform(10, 1000)     b maleT258   y0 10 1000
## uniform(10, 1000)     b maleT260   y0 10 1000
## exponential(disp_value) disp          0    20
## student_t(3, 0, 66.7) sd            x0 0 10
## student_t(3, 0, 66.7) sd male       x0 0 10
## student_t(3, 0, 66.7) sd Intercept male x0 0 10
##           source
##           user
## (vectorized)
##           user
## (vectorized)
##           user
##           user
## (vectorized)
## (vectorized)
## [1] "Fit Information"

## Warning: Parts of the model have not converged (some Rhats are > 1.05). Be
## careful when analysing the results! We recommend running more iterations and/or
## setting stronger priors.

## Warning: There were 2981 divergent transitions after warmup. Increasing
## adapt_delta above 0.99 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

```

**x0: groups\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE**



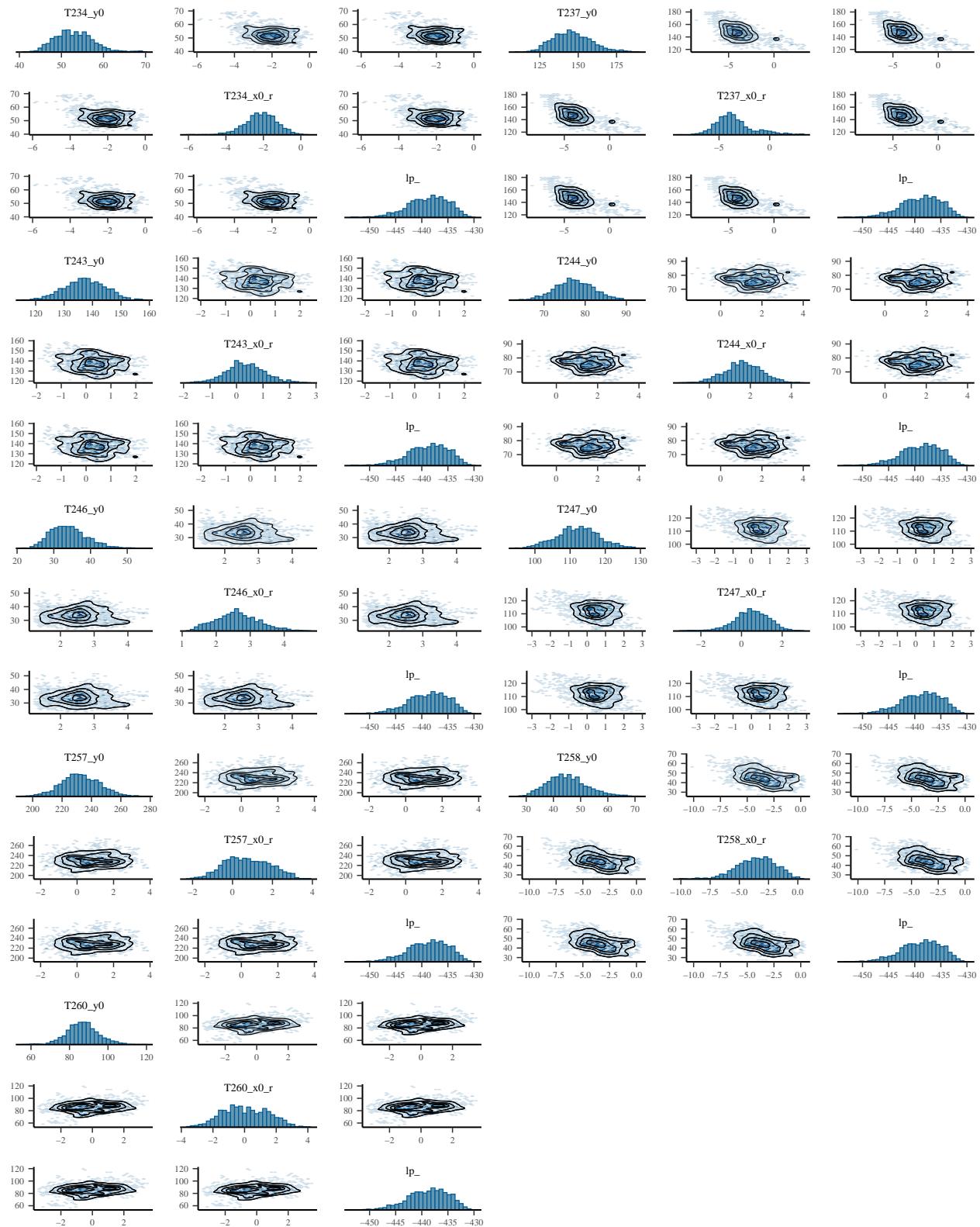
```
## Family: nbinom_type1
##   Links: mu = identity; disp = identity
```

```

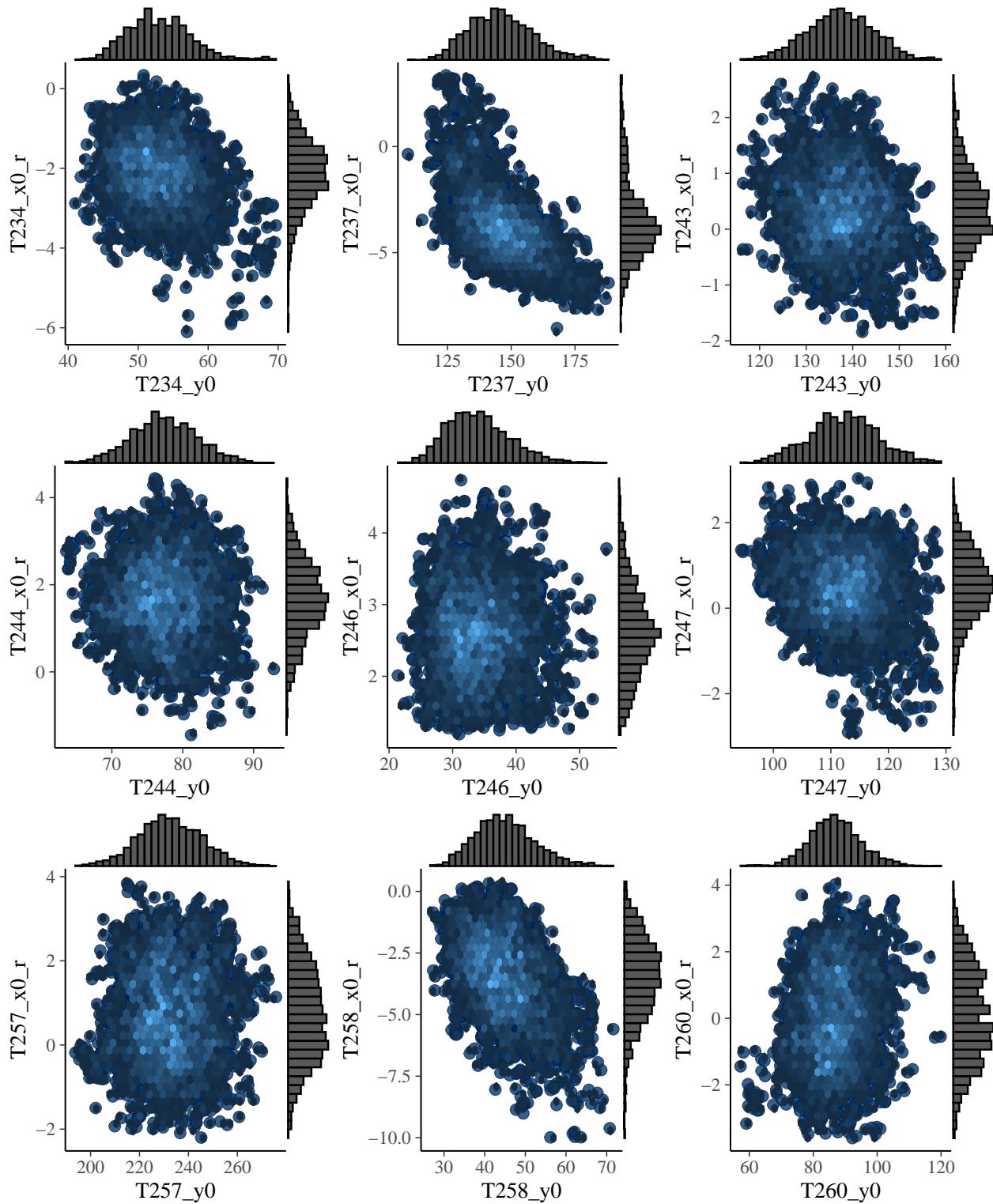
## Formula: y ~ two_piece(x, x0, y0)
##           x0 ~ 0 + Intercept + (1 | male)
##           y0 ~ 0 + male
## Data: data (Number of observations: 81)
## Draws: 6 chains, each with iter = 5000; warmup = 3000; thin = 4;
##        total post-warmup draws = 3000
##
## Group-Level Effects:
## ~male (Number of levels: 9)
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(x0_Intercept)    2.82      0.80     1.49    4.65 1.17      26      29
##
## Population-Level Effects:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## x0_Intercept     43.17      0.62    41.84    44.29 1.17      27      74
## y0_maleT234      53.06      4.52    45.35    63.22 1.08      89     141
## y0_maleT237     146.07     12.39   125.10   173.52 1.13      38     136
## y0_maleT243     137.35      7.16   123.19   150.86 1.08      58     280
## y0_maleT244     77.41       4.62    68.29    86.68 1.07      91     177
## y0_maleT246     34.06       4.91    25.85    44.63 1.02     234     366
## y0_maleT247    112.16      5.94   100.46   123.95 1.03     161     254
## y0_maleT257    232.52     12.86   207.44   257.84 1.05     136     243
## y0_maleT258     45.08      7.27    32.51    60.96 1.03     183     229
## y0_maleT260     86.90      8.08    71.37   103.87 1.03     172     218
##
## Family Specific Parameters:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disp        2.68      0.32     2.03    3.38 1.07      88     133
##
## 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).

```

**x0: groups\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**



**x0: groups\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**

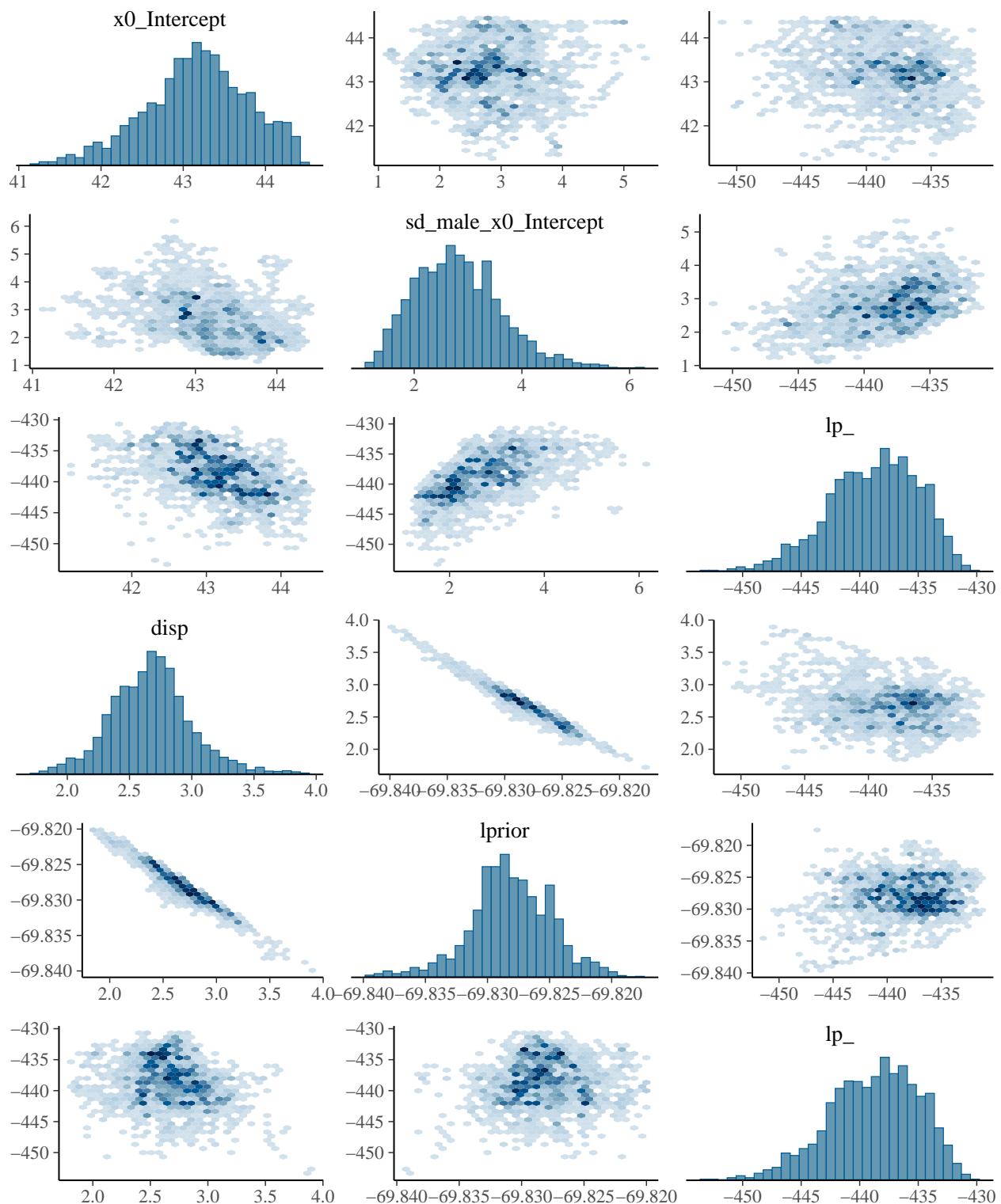


```
## [1] "Row: 3"
## [1] "nbinom_type1; two_piece; x0: individual; y0: individual; disp_flag: uniform_1; disp prior: 0.01
## [1] "Fit Prior Information"
```



```
## Warning: Parts of the model have not converged (some Rhats are > 1.05). Be  
## careful when analysing the results! We recommend running more iterations and/or  
## setting stronger priors.
```

**x0: groups\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**



```
## Family: nbinom_type1
##   Links: mu = identity; disp = identity
```

```

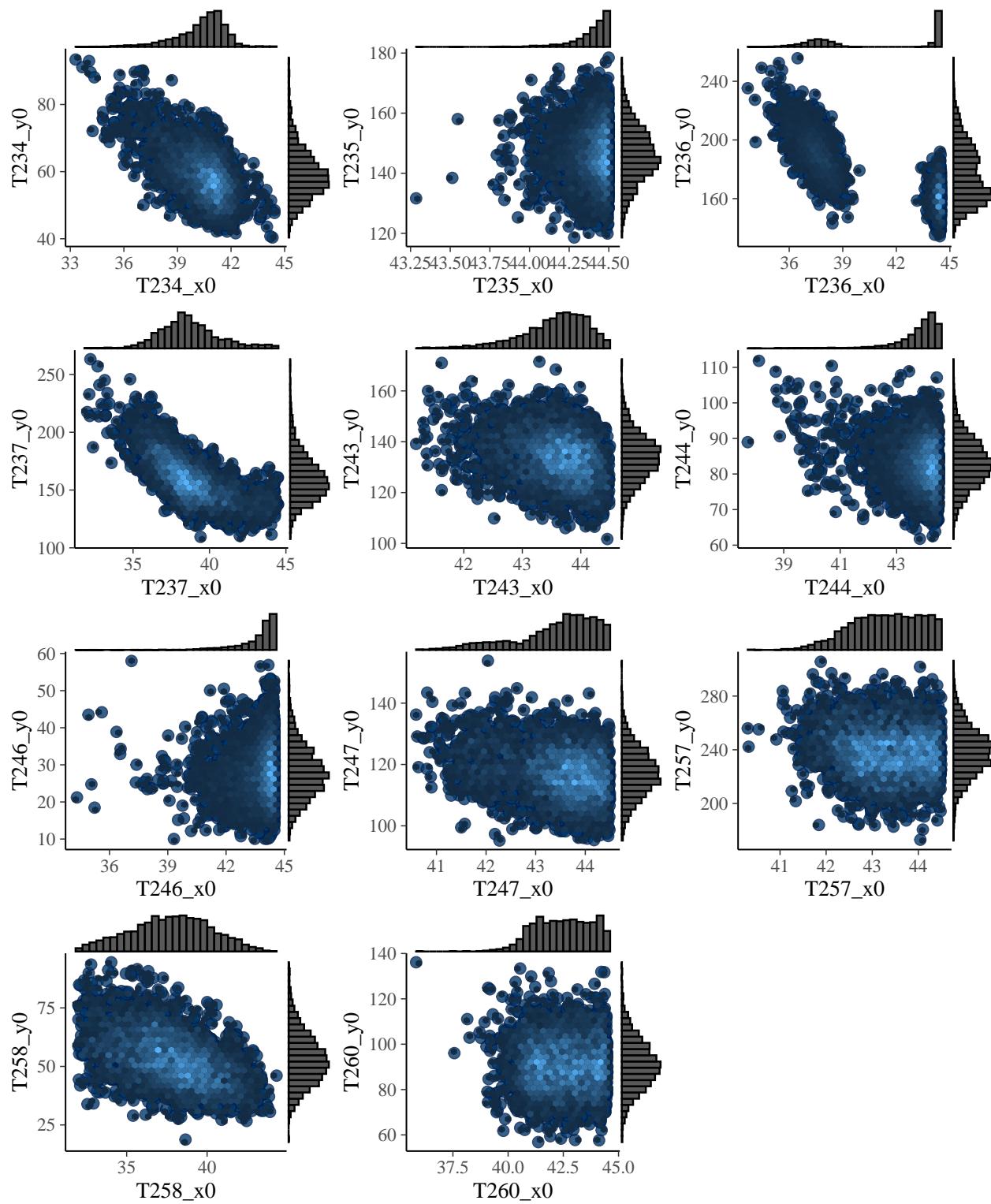
## Formula: y ~ two_piece(x, x0, y0)
##           x0 ~ 0 + male
##           y0 ~ 0 + male
## Data: data (Number of observations: 107)
## Draws: 6 chains, each with iter = 5000; warmup = 3000; thin = 4;
##        total post-warmup draws = 3000
##
## Population-Level Effects:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## x0_maleT234     40.34    1.40   36.69   42.26 1.00    2390    2493
## x0_maleT235     44.39    0.12   44.06   44.50 1.00    2901    2783
## x0_maleT236     40.93    3.48   35.94   44.49 1.71      9    244
## x0_maleT237     38.69    1.97   35.25   43.49 1.00    2586    2506
## x0_maleT243     43.55    0.57   42.16   44.38 1.00    2702    2597
## x0_maleT244     43.73    0.85   41.07   44.48 1.00    2677    2612
## x0_maleT246     43.62    1.11   40.73   44.48 1.00    3371    2955
## x0_maleT247     43.42    0.81   41.49   44.44 1.00    2772    2804
## x0_maleT257     43.29    0.75   41.77   44.44 1.00    2886    2957
## x0_maleT258     37.74    2.47   32.83   42.28 1.00    2884    2802
## x0_maleT260     42.47    1.25   40.13   44.40 1.00    2656    2447
## y0_maleT234     59.23    7.37   46.97   75.90 1.00    2784    2751
## y0_maleT235    146.29    9.13  128.38  164.79 1.00    2938    2957
## y0_maleT236    176.63   19.65  146.58  219.01 1.55      11    221
## y0_maleT237    160.51   19.52  128.66  206.01 1.00    2662    2732
## y0_maleT243    133.93    9.21  116.26  152.57 1.00    2982    2915
## y0_maleT244    83.20     6.88   70.38   97.71 1.00    2898    2516
## y0_maleT246    28.03     7.01   15.65   43.09 1.00    2699    2108
## y0_maleT247    117.36    7.86  102.44  132.99 1.00    2924    2974
## y0_maleT257    238.70   18.28  204.25  275.47 1.00    3115    2917
## y0_maleT258    53.22    10.91   34.19   76.57 1.00    3254    2937
## y0_maleT260    91.12    11.01   70.16  113.45 1.00    2985    2861
##
## Family Specific Parameters:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disp       6.00     0.50    5.11    7.05 1.01      622    2752
##
## 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).

```

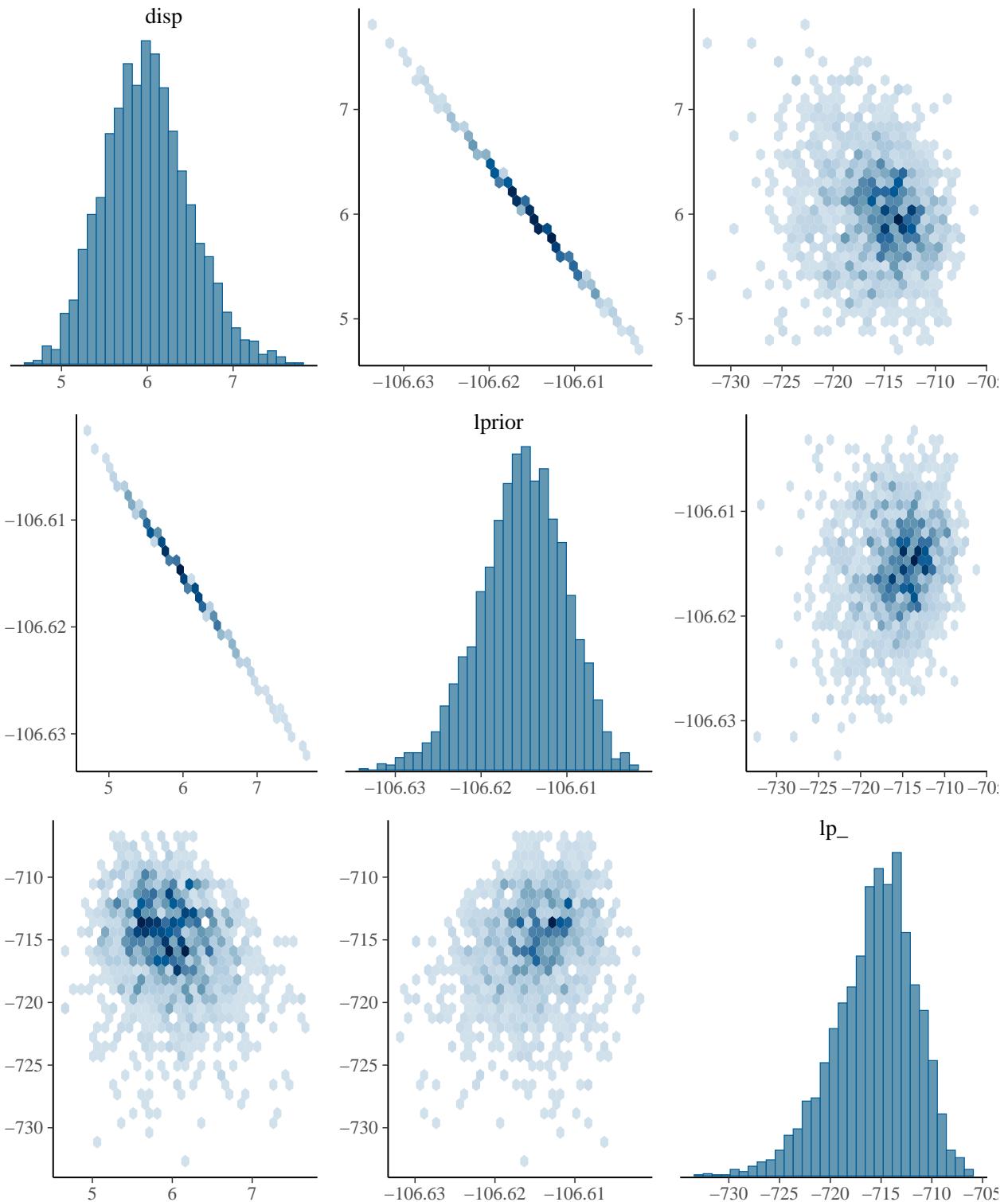
**x0: individual; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE**



**x0: individual; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE**



**x0: individual; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE**



```
## [1] "Row: 4"
## [1] "nbinom_type1; two_piece; x0: individual; y0: individual; disp_flag: uniform_1; disp prior: 0.01
## [1] "Fit Prior Information"
```

```

##          prior class      coef group resp dpar nlpar lb     ub
## uniform(32, 44.5)      b                   x0 32 44.5
## uniform(32, 44.5)      b maleT234           x0 32 44.5
## uniform(32, 44.5)      b maleT237           x0 32 44.5
## uniform(32, 44.5)      b maleT243           x0 32 44.5
## uniform(32, 44.5)      b maleT244           x0 32 44.5
## uniform(32, 44.5)      b maleT246           x0 32 44.5
## uniform(32, 44.5)      b maleT247           x0 32 44.5
## uniform(32, 44.5)      b maleT257           x0 32 44.5
## uniform(32, 44.5)      b maleT258           x0 32 44.5
## uniform(32, 44.5)      b maleT260           x0 32 44.5
## uniform(10, 1000)       b                   y0 10 1000
## uniform(10, 1000)       b maleT234           y0 10 1000
## uniform(10, 1000)       b maleT237           y0 10 1000
## uniform(10, 1000)       b maleT243           y0 10 1000
## uniform(10, 1000)       b maleT244           y0 10 1000
## uniform(10, 1000)       b maleT246           y0 10 1000
## uniform(10, 1000)       b maleT247           y0 10 1000
## uniform(10, 1000)       b maleT257           y0 10 1000
## uniform(10, 1000)       b maleT258           y0 10 1000
## uniform(10, 1000)       b maleT260           y0 10 1000
## exponential(disp_value) disp                      0    20
##         source
##         user
## (vectorized)
## [1] "Fit Information"
## Family: nbinom_type1
## Links: mu = identity; disp = identity
## Formula: y ~ two_piece(x, x0, y0)
##          x0 ~ 0 + male
##          y0 ~ 0 + male
## Data: data (Number of observations: 81)
## Draws: 6 chains, each with iter = 5000; warmup = 3000; thin = 4;
##        total post-warmup draws = 3000
##

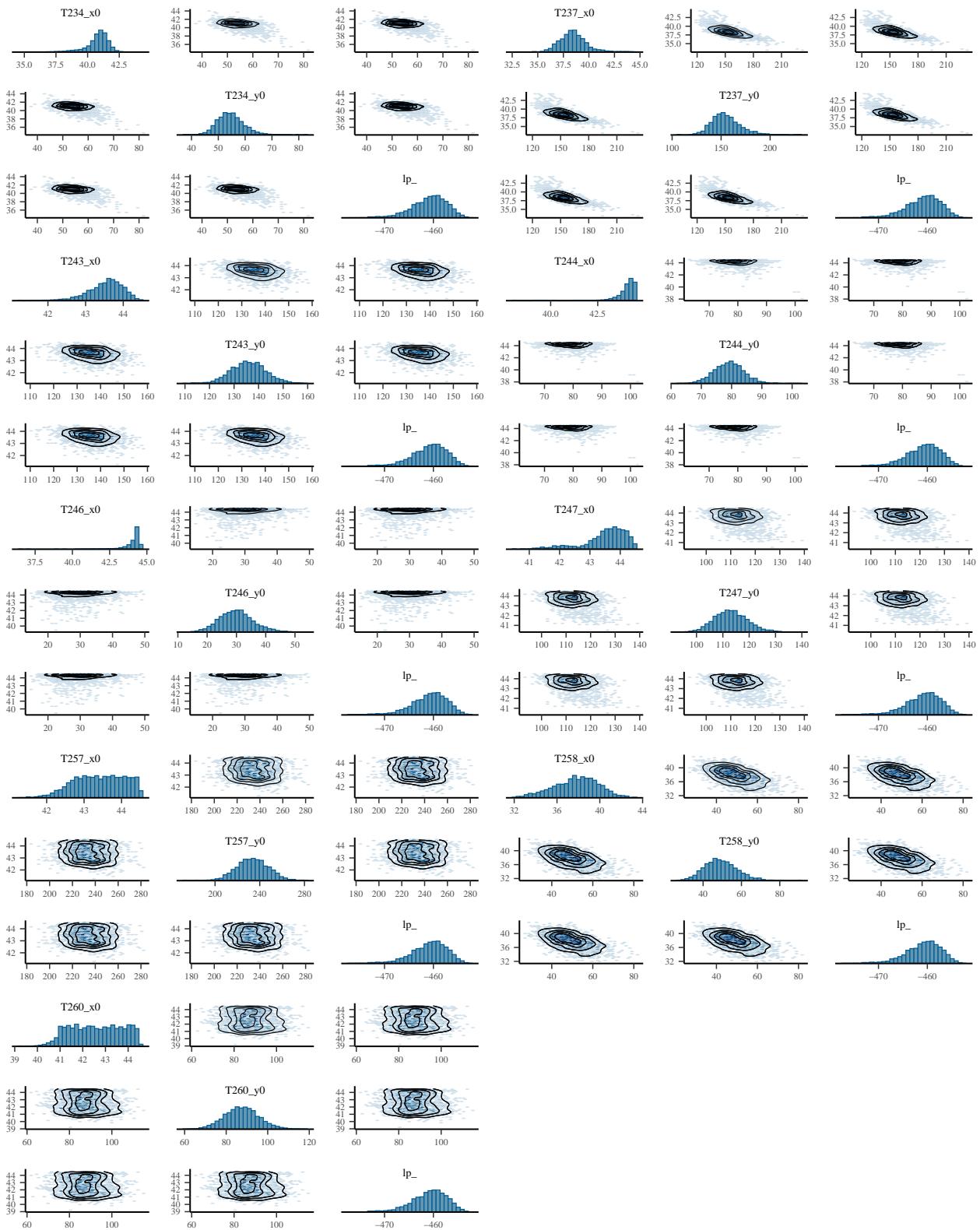
```

```

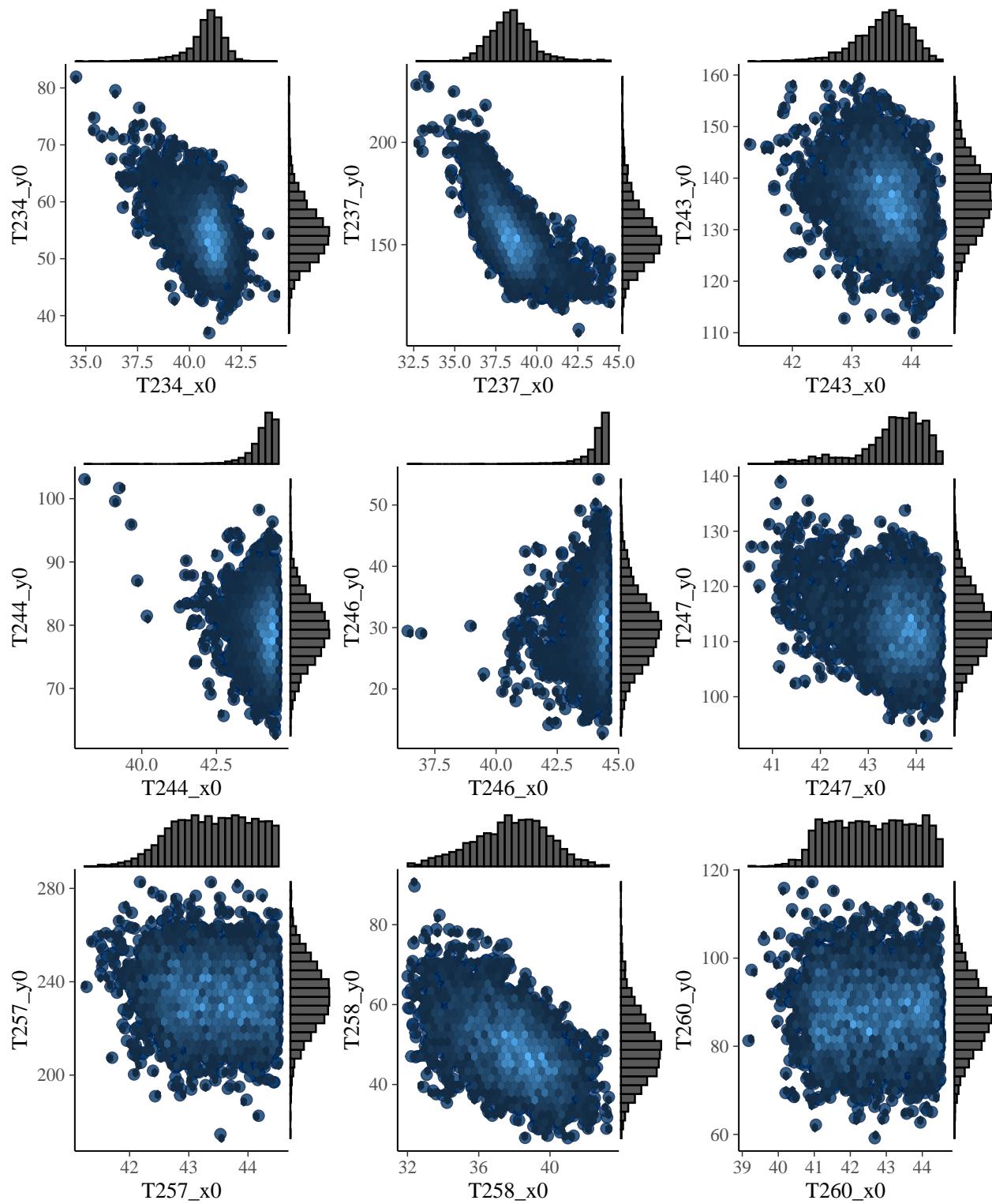
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## x0_maleT234    40.78     0.92   38.33   42.10 1.00    2646    2345
## x0_maleT237    38.46     1.40   36.05   41.78 1.00    2431    1958
## x0_maleT243    43.53     0.46   42.44   44.26 1.00    2945    2619
## x0_maleT244    44.04     0.48   42.86   44.48 1.00    2720    2587
## x0_maleT246    44.05     0.63   42.16   44.49 1.00    2987    2725
## x0_maleT247    43.51     0.71   41.59   44.42 1.00    2866    2585
## x0_maleT257    43.41     0.66   42.13   44.45 1.00    2780    2475
## x0_maleT258    37.79     2.08   33.42   41.53 1.00    2705    2885
## x0_maleT260    42.60     1.12   40.71   44.41 1.00    2653    2189
## y0_maleT234    54.64     4.97   45.94   65.33 1.00    2628    2730
## y0_maleT237   155.11    14.71  130.77  188.18 1.00    2771    2730
## y0_maleT243   135.92     7.11  122.34  150.17 1.00    2820    2725
## y0_maleT244    79.14     4.90   69.60   88.68 1.00    2813    2704
## y0_maleT246    30.00     5.77   19.47   42.26 1.00    2868    2802
## y0_maleT247   113.09     6.23  101.73  126.36 1.00    2927    2846
## y0_maleT257   233.72    14.22  207.04  261.48 1.00    2528    2612
## y0_maleT258    49.31     8.54   34.01   67.74 1.00    3018    2847
## y0_maleT260    87.51     8.26   71.53  104.42 1.00    3026    2959
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disp       3.04      0.34    2.44     3.76 1.00    3069    2879
##
## 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).

```

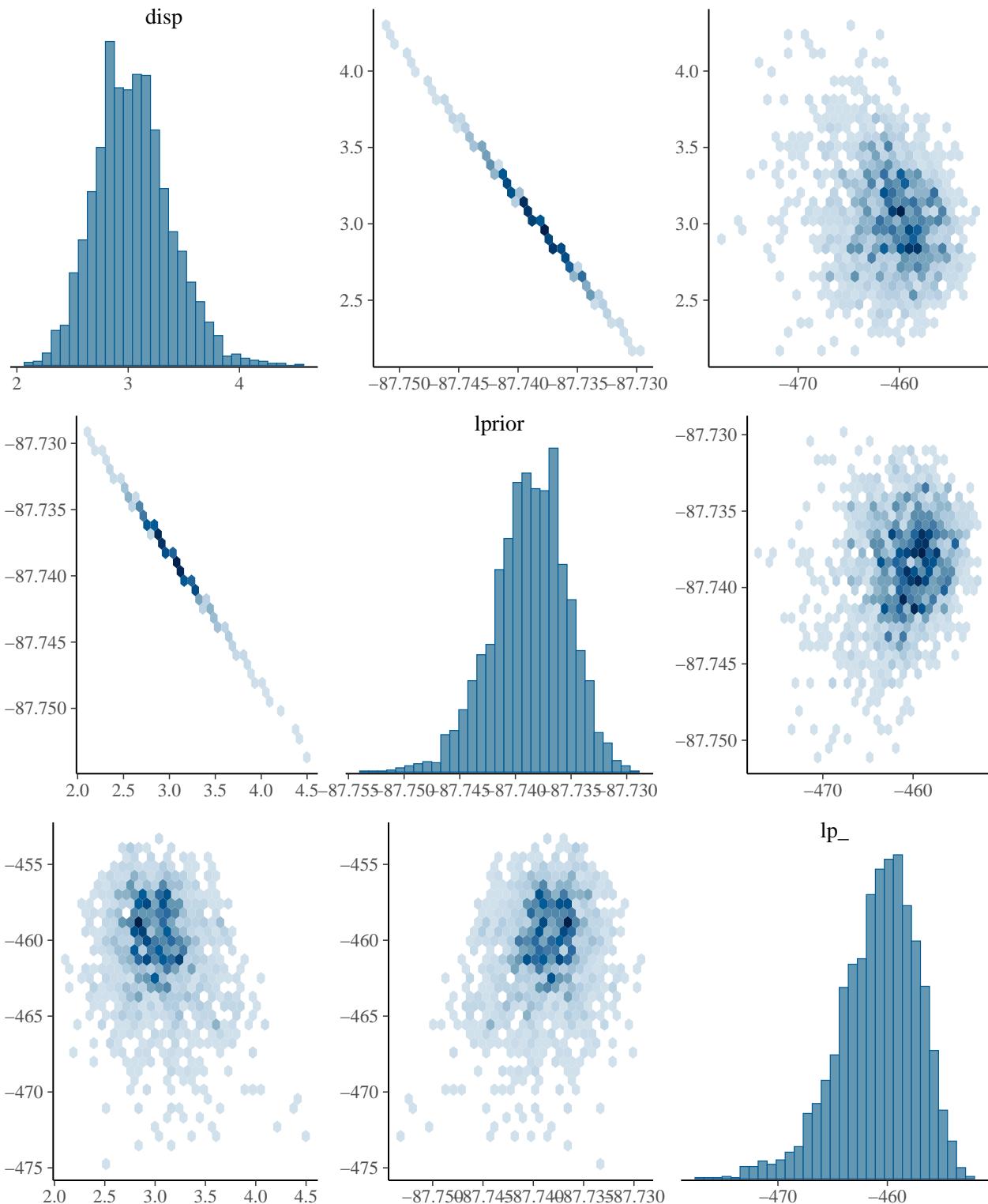
**x0: individual; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**



**x0: individual; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**



**x0: individual; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**



```
## [1] "Row: 5"
## [1] "nbinom_type1; two_piece; x0: uniform_1; y0: individual; disp_flag: uniform_1; disp prior: 0.01;"
```

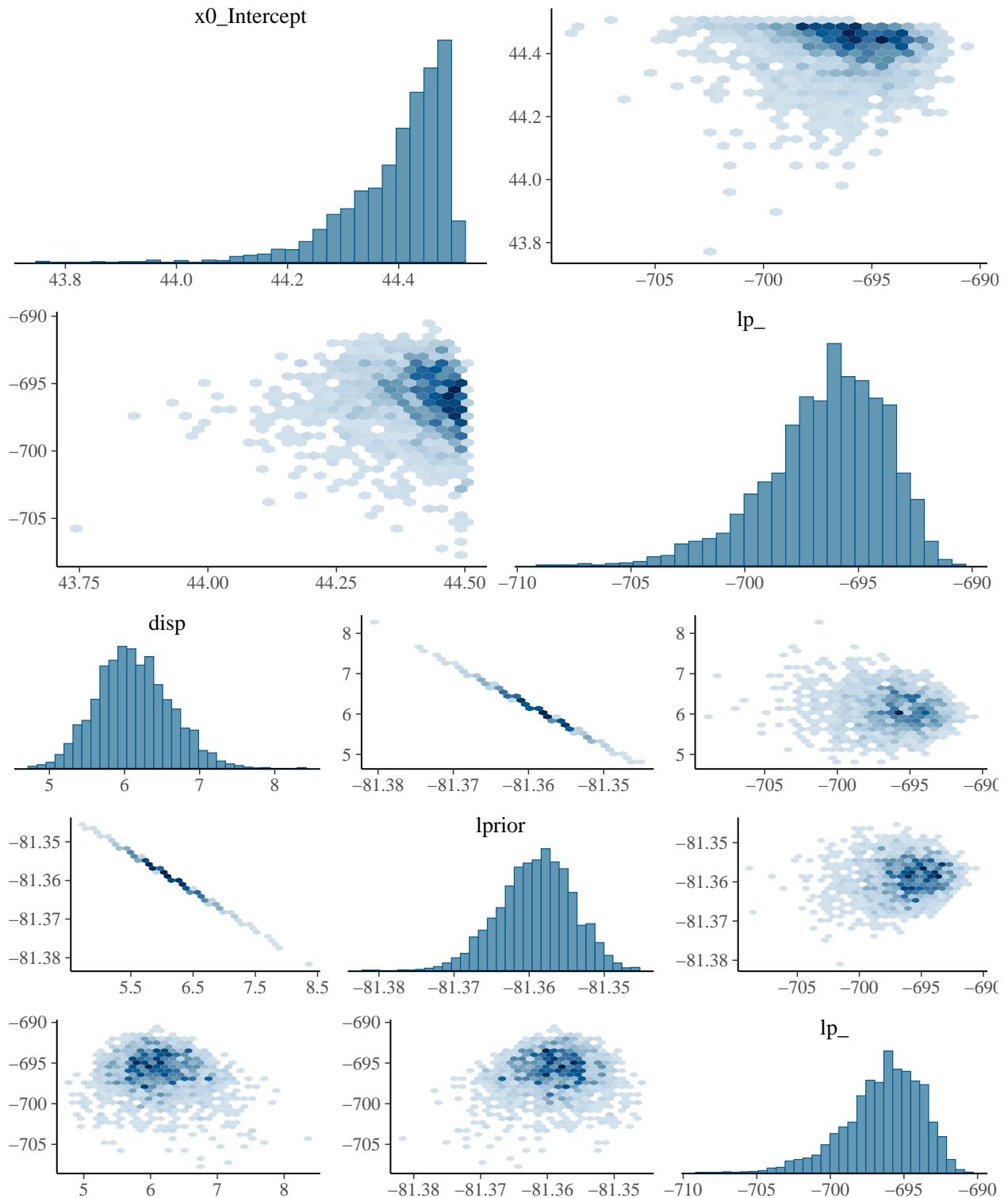
```

## [1] "Fit Prior Information"
##          prior class      coef group resp dpar nlnpar lb    ub
## uniform(32, 44.5)     b                      x0 32 44.5
## uniform(32, 44.5)     b Intercept              x0 32 44.5
## uniform(10, 1000)     b                      y0 10 1000
## uniform(10, 1000)     b maleT234             y0 10 1000
## uniform(10, 1000)     b maleT235             y0 10 1000
## uniform(10, 1000)     b maleT236             y0 10 1000
## uniform(10, 1000)     b maleT237             y0 10 1000
## uniform(10, 1000)     b maleT243             y0 10 1000
## uniform(10, 1000)     b maleT244             y0 10 1000
## uniform(10, 1000)     b maleT246             y0 10 1000
## uniform(10, 1000)     b maleT247             y0 10 1000
## uniform(10, 1000)     b maleT257             y0 10 1000
## uniform(10, 1000)     b maleT258             y0 10 1000
## uniform(10, 1000)     b maleT260             y0 10 1000
## exponential(disp_value) disp                  0    20
##          source
##          user
## (vectorized)
##          user
## (vectorized)
##          user
## [1] "Fit Information"
##   Family: nbinom_type1
##   Links: mu = identity; disp = identity
## Formula: y ~ two_piece(x, x0, y0)
##          x0 ~ 0 + Intercept
##          y0 ~ 0 + male
##   Data: data (Number of observations: 107)
##   Draws: 6 chains, each with iter = 5000; warmup = 3000; thin = 4;
##          total post-warmup draws = 3000
##
## Population-Level Effects:
##          Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## x0_Intercept  44.40     0.09   44.17   44.50 1.00  2642    2617
## y0_maleT234   49.49     4.88   40.59   59.64 1.00  2983    2955
## y0_maleT235  145.95     9.07  128.74  164.10 1.00  2925    2815
## y0_maleT236  161.37     9.03  144.51  179.66 1.00  3044    2747
## y0_maleT237  138.78    10.73  118.49  159.73 1.00  2675    2680
## y0_maleT243   127.80     8.61  111.62  145.51 1.00  2781    2700
## y0_maleT244   82.04      6.57   69.69   95.31 1.00  3046    2803
## y0_maleT246   28.67      6.92   16.46   43.23 1.00  2910    2787
## y0_maleT247  114.98     7.49  100.38  129.72 1.00  2780    2918

```

```
## y0_maleT257    237.00    18.09   201.82   272.35 1.00    2889    2613
## y0_maleT258     36.95     7.06   24.04    51.46 1.00    3026    2870
## y0_maleT260     91.16    11.03   69.67   112.62 1.00    2848    2831
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disp       6.12      0.49    5.22     7.11 1.00     2825     2723
##
## 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).
```

**x0: uniform\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: FALSE**



```
## [1] "Row: 6"
## [1] "nbinom_type1; two_piece; x0: uniform_1; y0: individual; disp_flag: uniform_1; disp prior: 0.01;
## [1] "Fit Prior Information"
```

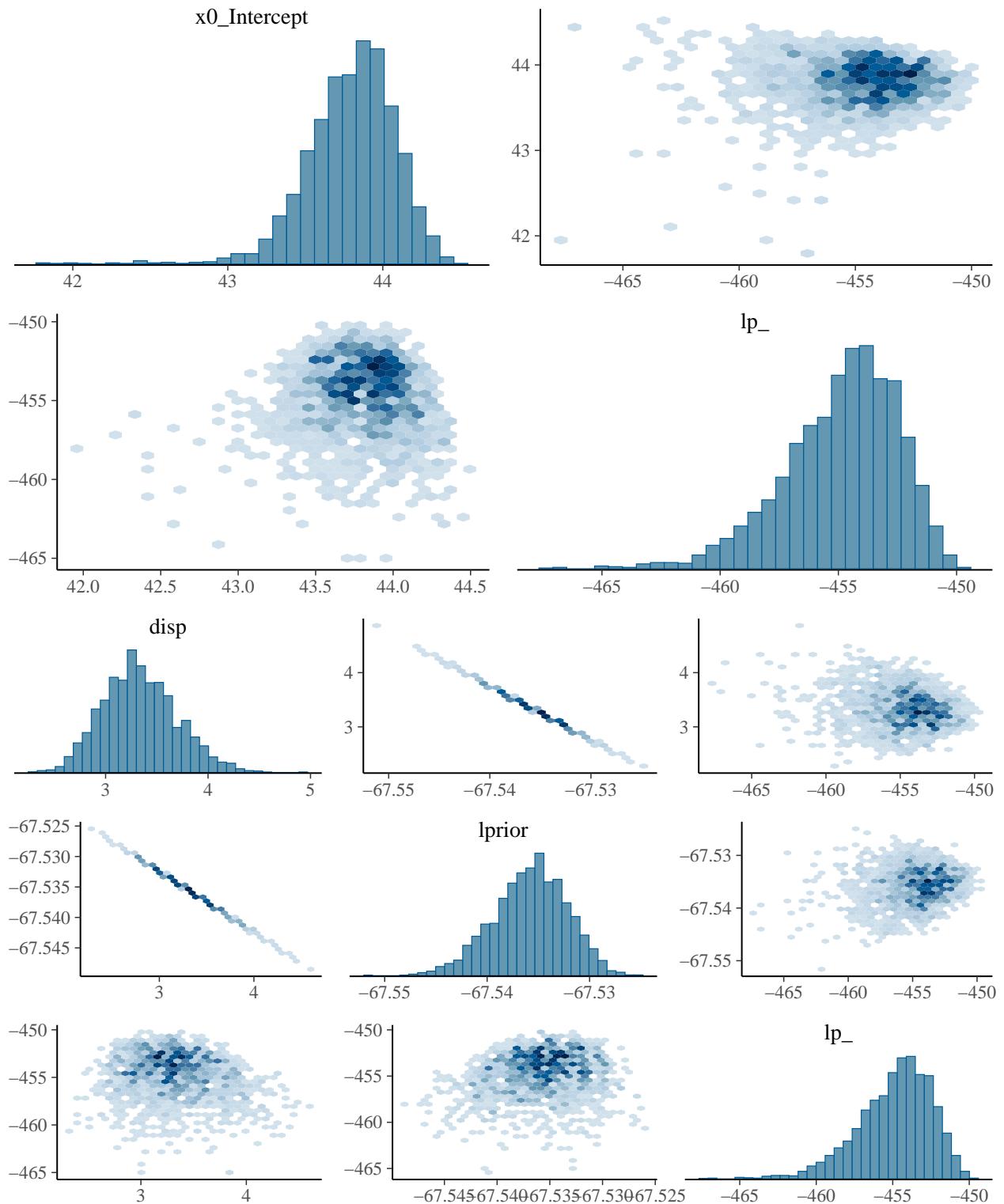
```

##          prior class      coef group resp dpar npar lb    ub
## uniform(32, 44.5)     b             x0 32 44.5
## uniform(32, 44.5)     b Intercept   x0 32 44.5
## uniform(10, 1000)     b             y0 10 1000
## uniform(10, 1000)     b maleT234   y0 10 1000
## uniform(10, 1000)     b maleT237   y0 10 1000
## uniform(10, 1000)     b maleT243   y0 10 1000
## uniform(10, 1000)     b maleT244   y0 10 1000
## uniform(10, 1000)     b maleT246   y0 10 1000
## uniform(10, 1000)     b maleT247   y0 10 1000
## uniform(10, 1000)     b maleT257   y0 10 1000
## uniform(10, 1000)     b maleT258   y0 10 1000
## uniform(10, 1000)     b maleT260   y0 10 1000
## exponential(disp_value) disp           0    20
##          source
##          user
## (vectorized)
##          user
## (vectorized)
##          user
## [1] "Fit Information"
## Family: nbinom_type1
## Links: mu = identity; disp = identity
## Formula: y ~ two_piece(x, x0, y0)
##          x0 ~ 0 + Intercept
##          y0 ~ 0 + male
## Data: data (Number of observations: 81)
## Draws: 6 chains, each with iter = 5000; warmup = 3000; thin = 4;
##        total post-warmup draws = 3000
##
## Population-Level Effects:
##          Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## x0_Intercept  43.79     0.29   43.19   44.27 1.00   2612    2366
## y0_maleT234   48.07     3.97   40.59   56.08 1.00   3067    3044
## y0_maleT237  134.90     8.26  119.11  151.44 1.00   2899    2849
## y0_maleT243  134.47     7.38  119.75  149.27 1.00   2860    2876
## y0_maleT244   79.85     4.97   70.58   89.79 1.00   2935    2843
## y0_maleT246   29.08     5.87   18.39   41.13 1.00   3163    2961
## y0_maleT247  112.78     5.97  101.28  124.84 1.00   2839    2959
## y0_maleT257  233.41    13.98  206.78  262.89 1.00   2906    2772
## y0_maleT258   37.34     5.75   26.91   49.20 1.00   2970    2813
## y0_maleT260   87.93     8.56   71.55  105.93 1.00   3083    2802
##
## Family Specific Parameters:
##          Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disp      3.33      0.36    2.69     4.08 1.00   2967    3002

```

```
##  
## 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).
```

**x0: uniform\_1; y0: individual; disp\_flag: uniform\_1; disp prior: 0.01; filter: TRUE**



## Plot Data and Predictions

### Set up colors

```
## "Set2" only has 8 colors by default.  
## This code expands that ability, based on: https://www.datanovia.com/en/blog/easy-way-to-expand-color  
n_colors <- length(male_vec)  
colors_male <- colorRampPalette(brewer.pal(8, "Set2"))(n_colors)
```

Data suggests that disp should vary between males?

### Plot Data and Expected Values vs. Temperature

```
curr_row_max <- switch(2,  
                        2,  
                        length(fit_tbl)  
                      )  
  
for(curr_row in 1:curr_row_max) {  
  
  curr_desc <- fit_tbl[[curr_row, "desc"]]  
  curr_desc_short <- curr_desc %>% str_replace("nbinom_type1; two_piece; ", "")  
  fit_brms <- fit_tbl[[curr_row, "fit"]][[1]]  
  
  expose_functions(fit_brms, vectorize = TRUE, show_compiler_warnings = FALSE)  
  data_fit_brms <- fit_brms$data  
  
  ## Create grid of x values for epred/predictions  
  
  data_grid <- data_fit_brms %>%  
    group_by(male) %>%  
    data_grid(x = seq_range(c(20, 46), n = 51))  
  
  ## add expected values  
  data_epred <- data_grid %>%  
    add_epred_draws(object = fit_brms, ndraws = 500, cores = n_cores)  
  
  male_vec_tmp <- male_vec #[c(1, 5, 8)]  
  ## Only plot a few males due to large size of `data_epred`  
  data_epred_tmp <- data_epred %>%  
    filter(male %in% male_vec_tmp)  
  
  data_fit_brms_tmp <- data_fit_brms %>% filter(male %in% male_vec_tmp)  
  y_max <- max(data_epred_tmp$.epred)*1.1  
  
  plot_tmp <- ggplot(data = data_epred_tmp, aes(x = x, y = .epred, color = male)) +  
    ## Combine Scatter Plots and Model vs Data Plots  
    stat_lineribbon(aes(y=.epred)) +
```

```

scale_fill_brewer(palette = "Greys") +
scale_color_manual(values = colors_male) +
#scale_color_brewer(palette = "Set2") +
geom_point(data = data_fit_brms_tmp,
           aes(x = x, y = y), color = "red") +
ylim(0, y_max)

plot_data_vs_epred <- plot_tmp + facet_wrap(vars(male)) +
  labs(title = "Data vs. Expected Values", subtitle = curr_desc_short)
last_plot()
filename <- paste0("data.and.epred-vs-x_", filename_desc, ".png")
ggsave(filename = filename, path = file.path(output_dir, "figures"),
       width = 8, height = 11, units = "in",
       scale = 1,
       dpi=300,
       bg = "white")
}

## Warning: Removed 1 rows containing missing values ('geom_point()').
## Removed 1 rows containing missing values ('geom_point()').
## Removed 1 rows containing missing values ('geom_point()').

## Error in 'fit_tbl[[curr_row, "desc"]]' :
## ! Can't extract rows past the end.
## i Location 7 doesn't exist.
## i There are only 6 rows.

```

## Plot Data and Simulated Values vs. Temperature

```

for(curr_row in 1:curr_row_max) {

  curr_desc <- fit_tbl[[curr_row, "desc"]]
  curr_desc_short <- curr_desc %>% str_replace("nbinom_type1; two_piece; ","")
  fit_brms <- fit_tbl[[curr_row, "fit"]][[1]]

  if(FALSE) {
    object.size(fit_brms)
    ## Unclear where function information is stored
    ## Fit object does not seem to change in size
    expose_functions(fit_brms, vectorize = TRUE, show_compiler_warnings=FALSE)
    object.size(fit_brms)
  }

  data_fit_brms <- fit_brms$data

  ## Create grid of x values for epred/predictions

  #dataframe_tmp <- crossing(x = seq_range(c(20, 30), n = 51), draw = 1:3, male = male_vec)

  data_grid <- data_fit_brms %>%

```

```

group_by(male) %>%
  data_grid(x = seq_range(c(20, 45.9), n = 51)) %>% #, .model = fit_brms) %>%
  ungroup()

## add simulated values
## Only getting 1 draw/(male temp)
data_pred <- data_grid %>%
  add_predicted_draws(object = fit_brms)

y_max <- max(data_pred$.prediction, na_rm = TRUE)*1.1

plot_tmp <- ggplot(data = data_pred, aes(x = x, y = .prediction, color = male)) +
  ## Combine Scatter Plots and Model vs Data Plots
  stat_lineribbon(aes(y=.prediction), .width = c(.95), color = "#08519C") +
  scale_fill_brewer(palette = "Greys") +
  scale_color_manual(values = colors_male) +
  #scale_color_brewer(palette = "Set2") +
  geom_point(data = data_fit_brms,
             aes(x = x, y = y), color = "red") +
  ylim(0, y_max)

plot_data_vs_pred <- plot_tmp + facet_wrap(vars(male)) +
  labs(title = "Data vs. Simulated Values", subtitle = curr_desc_short)
last_plot()

filename <- paste0("data-simulated-vs-x_", filename_desc, ".png")
ggsave(filename = filename, path = file.path(output_dir, "figures"),
       width = 8, height = 11, units = "in",
       scale = 1,
       dpi=300,
       bg = "white")
}

## Error in `fit_tbl[[curr_row, "desc"]]:`:
## ! Can't extract rows past the end.
## i Location 7 doesn't exist.
## i There are only 6 rows.

```

Exit rendering

```
knitr::knit_exit()
```