

# First Fitting of Thermal Models using BRMS

Michael Gilchrist

date: 2022-10-20

## Goal

- Fit series of thermal models in rTPC using `brms`.

## Recap

- Previously spent time using `glm` and `glmer` models.
- These models had issues with over dispersion when using a poisson family or converging when using a negative binomial family.

## Current Approach

### Analyze directly with rTPC?

No

According to Padfield et al. (2021)

However, [the rTPC] pipeline does not accommodate non-independent (related) replicates, and clustered or stratified sampling (possibly with missing values). In such situations, nonlinear mixed effects model fitting (e.g. using the `nlme` r package; Oddi et al., 2019) or Bayesian approaches (e.g. using the `brms` r package; Bürkner, 2017) would be more appropriate. Nevertheless, for fitting massive TPC datasets to multiple mathematical models, rTPC offers a simple, reliable and reproducible computational pipeline with robust methods for calculation of model uncertainty, requiring minimal statistical and computational expertise, and suitable for a wide range of applications.

Thus, we can't incorporate individual or random effects. We could try and use all of the observations in `song_prop` to deal with the `male` term.

### Analyze with `brms`

Yes.

- Can't use rTPC functions directly (e.g. `lactin2_1995`) in `brm()` calls since that uses `stan` rather than R functions.
- Having issues fitting models, should consider using output from `glm` fits.

## Data and Explanatory Variables

- Data: Use song\_count from round 3

## Future Plans

### Data and Explanatory Variables

- Use additional rounds when we get temp\_mean and humidity\_mean data.
- Include a tensor spline to describe trial order effects.

### Including beak and mass data

- Include beak size (surface area) as additional explanatory variable. We do have bird mass as well
- Notes from Liz
  - Do birds with larger beaks maintain singing at higher rates at higher temps?
  - We calculated bill surface area approximately as the surface area of a cone: length \* pi \* (width + depth) / 4.
  - Because larger animals produce more heat (Kleiber 1932), we scaled bill size relative to heat production by dividing bill surface area by expected daily energy consumption (mass0.658; Speakman and Kr'ol 2010, Hudson et al. 2013).
  - So, looks like we need to calculate bill surface area and then scale relative to heat production (bill surface area/mass0.658)

### Additional Possibilities

- Follow up with Ray/Juan about ground versus surface temp for operative temperature.
- What is the 0 for operative temperature?
- Ponder utility of other ZF data on panting.

## Set up

### Load libraries

```
## load libraries

library(rstan)

## Loading required package: StanHeaders

##
## rstan version 2.26.13 (Stan version 2.26.1)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```

## For within-chain threading using 'reduce_sum()' or 'map_rect()' Stan functions,
## change 'threads_per_chain' option:
## rstan_options(threads_per_chain = 1)

#options(mc.cores = parallel::detectCores()))
rstan_options(auto_write = TRUE)
if(FALSE) example(stan_model, package = "rstan", run.dontrun = TRUE)

library(brms)

## Loading required package: Rcpp

## Loading 'brms' package (version 2.18.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').

##
## Attaching package: 'brms'

## The following object is masked from 'package:rstan':
## 
##     loo

## The following object is masked from 'package:stats':
## 
##     ar

if(FALSE) {
  ## This often causes core dumps
  prior1 <- prior(normal(0,10), class = b) +
    prior(cauchy(0,2), class = sd)
  fit1 <- brm(count ~ zAge + zBase * Trt + (1|patient),
             data = epilepsy, family = poisson(),
             prior = prior1, chains = 1, cores = 1)

  fit2 <- brm(count ~ zAge + zBase * Trt + (1|patient),
             data = epilepsy, family = poisson(),
             prior = prior1, chains = 2, cores = 1)

  fit3 <- brm(count ~ zAge + zBase * Trt + (1|patient),
             data = epilepsy, family = poisson(),
             prior = prior1, cores = 2, chains = 2)
}

library(rTPC)
library(nls.multstart)
library(broom)

library(stats)
#library(brms)
library(tidyverse)

```

```

## -- Attaching packages
## -----
## tidyverse 1.3.2 --

## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr    1.0.10
## v tidyr   1.2.1      v stringr  1.4.1
## v readr   2.1.3      vforcats  0.5.2
## -- Conflicts --
## x tidyr::extract() masks rstan::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(ggplot2)
library(ggpubr)
library(ggsci) ## provides scale_fill_npg()
library(viridisLite)
library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(reshape2)

## 
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyverse':
## 
##   smiths

library(humidity) ## provides VPD
library(weathermetrics)
library(latex2exp)

```

## Local Functions

```

## Taken from: https://stackoverflow.com/a/51330864/5322644
## Use to get model equations for models in rTPC

help_text <- function(...) {
  file <- help(...)
  path <- dirname(file)
  dirpath <- dirname(path)
  pkgname <- basename(dirpath)
  RdDB <- file.path(path, pkgname)
  rd <- tools:::fetchRdDB(RdDB, basename(file))
  capture.output(tools:::Rd2txt(rd, out="", options=list(underline_titles=FALSE)))

```

```

}

get_model_eq <- function(model) {
  txt <- help_text(model)
  eqn_line <- grep("^ +rate = .*\"", txt, value = TRUE)
  eqn <- gsub("(^ +rate = | *$)", "", eqn_line) %>% gsub("\\.(\\d{1,2})", " * \\1", .)
  df <- tibble(model = model, eq = eqn)
  return(df)
}

plot_brms_fit <- function(brmsfit) {
  plist = list()

  plist[[1]] <- plot(brmsfit, title = paste("brm() summary: ", prior_index), ask = FALSE)
  return(plist)
}

plot_stan_fit <- function(stanfit) {
  plist = list();
  i <- 1

  plist[[i]] <- plot(stanfit, title = paste("stan() summary: ", prior_index))

  ## Plotting routines from: www.weirdfishes.blog/blog/fitting-bayesian-models-with...
  ##
  mack_diagnostics <- rstan::get_sampler_params(stanfit) %>%
    set_names(1:4) %>%
    map_df(as_tibble, .id = 'chain') %>%
    group_by(chain) %>%
    mutate(iteration = 1:length(chain)) %>%
    mutate(warmup = iteration <= warmup) %>%
    mutate()
  i <- i + 1

  plist[[i]] <- mack_diagnostics %>%
    group_by(warmup, chain) %>%
    summarise(percent_divergent = mean(divergent_>0)) %>%
    ggplot() +
    geom_col(aes(chain, percent_divergent, fill = warmup), position = 'dodge', color = 'black') +
    scale_y_continuous(labels = scales::percent, name = "% Divergent Runs") +
    scale_fill_npg()

  i<- i+1
  plist[[i]] <- mack_diagnostics %>%
    ggplot(aes(iteration, treedepth_>, color = chain)) +
    geom_line() +
    geom_hline(aes(yintercept = max_treedepth), color = 'red') +
    scale_color_locuszoom()

  return(plist)
}

```

## Create Model Tibble

```
model_def_tbl <- lapply(get_model_names(), get_model_eq) %>% bind_rows(, .id = NULL)
print(model_def_tbl, n = 200)

## # A tibble: 26 x 2
##   model      eq
##   <chr>     <chr>
## 1 beta_2012 "(a * ((temp - b + ((c * (d-1))/(d + e - 2)))/c)^(d-1)~"
## 2 boatman_2017 "rmax * (sin(pi * ((temp - tmin)/(tmax - tmin)))^a))^b"
## 3 briere2_1999 "a * temp * (temp - tmin) * (tmax - temp)^(1/b)"
## 4 delong_2017 "c * exp(-(eb-(ef * (1-((temp + 273.15)/tm))+ehc * ((t~"
## 5 flinn_1991 "1 / (1 + a + b * temp + c * temp^2))"
## 6 gaussian_1987 "rmax * exp(-0.5 * (abs(temp - topt)/a)^2)"
## 7 hinshelwood_1947 "a * exp(-e/k * (temp + 273.15)) - b * exp(-eh/k * (te~"
## 8 joehnk_2008 "rmax * (1 + a * ((b^(temp - topt) - 1) - (log(b)/log(~"
## 9 johnsonlewin_1946 "(r0 * exp(-e/(k * (temp + 273.15)))) / ((1 + exp((-1/~"
## 10 kamykowski_1985 "a * (1 - exp(-b * (temp - tmin))) * (1 - exp(-c * (t~"
## 11 lactin2_1995 "exp(a * temp) - exp(a * tmax - ((tmax - temp) / delt~"
## 12 lrf_1991 "rmax * ((temp - tmax)(temp - tmin))^2 / ((topt - tmin~"
## 13 modifiedgaussian_2006 "rmax * exp(-0.5 * (abs(temp - topt)/a)^b)"
## 14 oneill_1972 "rmax * (ctmax - temp / ctmax - topt)^2 * exp(x * (tem~"
## 15 pawar_2018 "r_tref * exp(e/k * (1/tref - 1/(temp + 273.15))) / (1~"
## 16 quadratic_2008 "a + b * temp + c * temp^2"
## 17 ratkowsky_1983 "((a * (temp - tmin)) * (1 - exp(b * (temp - tmax))))^~"
## 18 rezende_2019 "(a.10^(log10(q10)/(10/temp)))^"
## 19 rezende_2019 "(a.10^(log10(q10)/(10/temp))) * (1-c * (b - temp)^2))"
## 20 sharpschoolfull_1981 "r_tref * exp(e/k * (1/tref - 1/(temp + 273.15))) / (1~"
## 21 sharpschoolhigh_1981 "r_tref * exp(e/k * (1/tref - 1/(temp + 273.15))) / (1~"
## 22 sharpschoollow_1981 "r_tref * exp(e/k * (1/tref - 1/(temp + 273.15))) / (1~"
## 23 spain_1982 "est = r0 * exp(a * temp) * (1 - b * exp(c * temp)~"
## 24 thomas_2012 "a * exp(b * temp) * (1 - ((temp - topt)/(c/2))^~"
## 25 thomas_2017 "a * exp(b * temp) - (c + d * (exp(e * temp)))"
## 26 weibull_1995 "((a * (((c-1)/c)^((1-c)/c)) * (((temp-topt)/b)+(((c~-
```

## Set Output

```
output_dir = "output"
if (!dir.exists(output_dir)) {dir.create(output_dir)}
```

## Load Data

```
## Read in ZEFI Data sets
## Treat 'repeatability' as round = 0
## Add round info

## Repeatability was done between round 1 and 2, female was present, but only one temp. so treating as
```

```

git_root <- system("git rev-parse --show-toplevel", intern = TRUE)

data_raw = list()

data_raw[[1]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Round-1-Heat-Trials.csv")) %>%
  ## Note T237 and T230 are missing numbers in the song_count column
  ## so we are filtering these observations out until they are found
  filter(!is.na(song_count)) %>%
  ungroup()

data_raw[[2]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Repeatability-Song-Count.csv")) %>%
  mutate(round = 2) %>%
  group_by(male) %>%
  mutate(test_order = rank(date)) %>%
  ungroup()

data_raw[[3]] <- read.csv(file.path(git_root, "data", "raw_data", "HSPi-Round-2-Heat-Trials.csv")) %>%
  mutate(round = 3) %>%
  ## Deal with missing temp_mean and humidity_mean values
  ## in round == 3
  ## 2022/10/19 - code no longer needed
  ## group_by(temp_target) %>%
  ##mutate(temp = if_else((round == 3 & is.na(temp_mean)),
  ##                      mean(temp_mean, na.rm = TRUE),
  ##                      temp_mean)) %>%
  ##mutate(humidity = if_else((round == 3 & is.na(humidity_mean)),
  ##                           mean(humidity_mean, na.rm = TRUE),
  ##                           humidity_mean)) %>%
  ungroup()

## Join data and discard empty columns
data_full <- full_join(data_raw[[1]], data_raw[[2]]) %>%
  full_join(data_raw[[3]]) %>%
  discard(~all(is.na(.) | . == "")) %>% ## get rid of columns of only NA
  mutate(trial_completed = !(is.na(song_count))) %>%
  mutate(song_count = ifelse(is.na(song_count), 0, song_count)) %>%
  mutate(song_count = song_count*1.0) %>% ## convert to a double so it's not treated as an integer
  mutate(chamber = as.factor(chamber), male = as.factor(male)) %>%
  ## create a global variable trial_order based on individual rounds
  mutate(trial_index = as.integer(round*10+test_order)) %>%
  mutate(song_count_plus_1 = (song_count + 1)) %>%
  mutate(log_song_count_plus_1 = log(song_count + 1)) %>%
  mutate(temp_target = as.numeric(temp_target)) %>%
  ## Create generic 'temp' column which is either
  ## temp_mean, if it exists, or temp_target, if it doesn't
  mutate(temp = if_else(is.na(temp_mean),
                       temp_target,
                       temp_mean)) %>%
  ## Add column with total song_count for a given round
  group_by(male, round) %>%
  mutate(count_total_round = sum(song_count)) %>%
  ungroup() %>%

```

```

    mutate(song_prop = song_count/count_total_round) %>%
    ## assuming poisson error
    ## From glm man page
    ## > Non-'NULL' 'weights' can be used to indicate that different
    ## > observations have different dispersions (with the values in
    ## > 'weights' being inversely proportional to the dispersions);
    ## add +1 to deal with single 0
    mutate(count_wt = 1/(song_count + 1)) %>%
    ## need to rescale wts for song_prop data
    mutate(prop_wt = count_wt * count_total_round^2) %>%
    ## Add vpd
    mutate(svp = SVP(t = temp_mean + 273.15, isK = TRUE), vpd = svp*(1-humidity_mean/100) ) %>%
    group_by(round) %>%
    mutate(vpd_offset = vpd - mean(vpd)) %>%
    ungroup() %>%
    relocate(song_count, song_prop, vpd, temp_mean, humidity_mean, .after = male) %>%
    mutate() ## Dummy function so we can comment out lines above it w/o any issues

## Joining, by = c("male", "chamber", "date", "song_count", "counter", "test_order", "temp_target", "round")
## Joining, by = c("male", "chamber", "date", "song_count", "counter", "test_order", "temp_target", "round")

```

## Examine Data

```

data_count_total <- data_full %>% group_by(round) %>%
  select(male, round, count_total_round) %>%
  distinct()

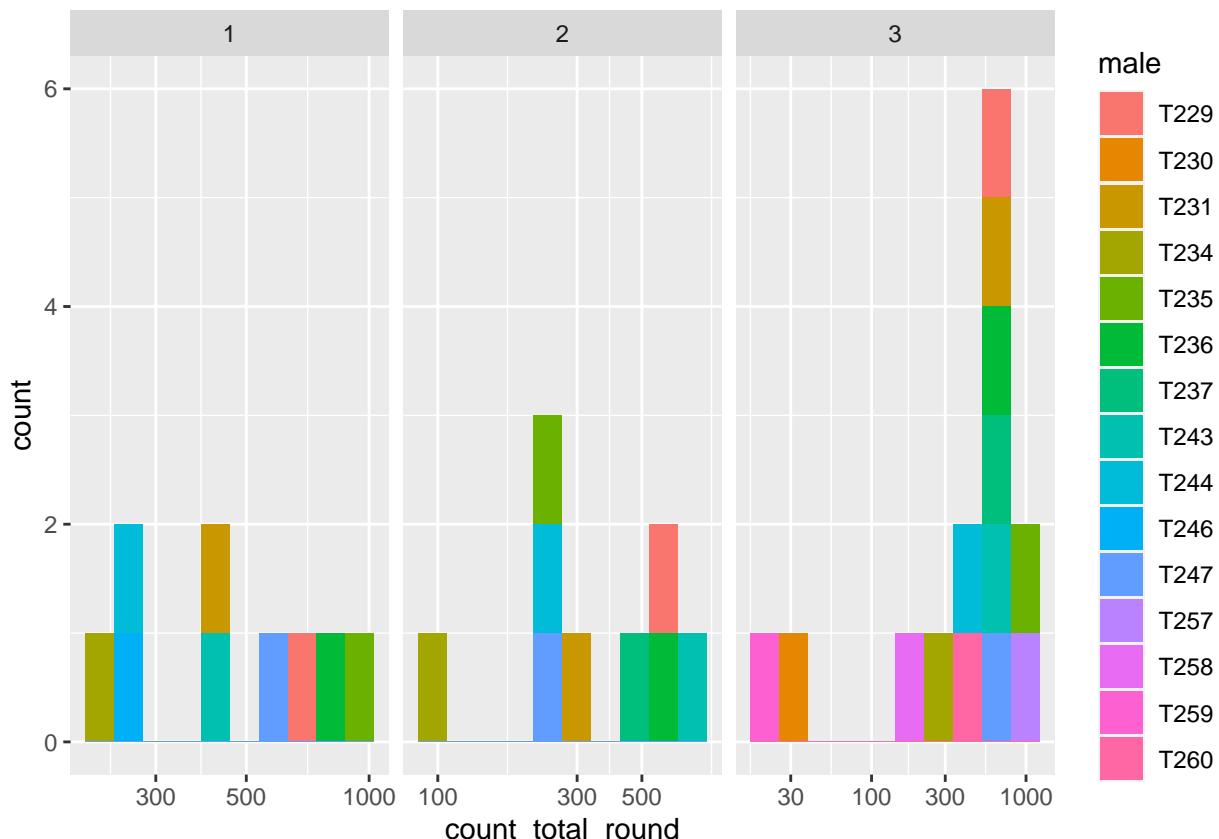
t <- ggplot(data_count_total, aes(count_total_round, fill = male)) +
  geom_histogram(bins = 10) +
  scale_x_log10()

hist_count_total <- t +
  facet_grid(cols = vars(round), scales = "free_x")
hist_count_total

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Removed 1 rows containing non-finite values (stat_bin).

```



```
## Compare count_total_round between round 1 and 3
```

- See third.fitting.Rmd

## Result

- As before, we see strong consistancy between round 1 and 3.
- Consistency with round 2 is weaker, but sample sizes are smaller: 3 trials/male in round 2 vs 6 trials/male in round 3.

## Create Filtered Data

```
data_ind <- data_full %>%
  filter(
    (round == 3 & count_total_round >= 150) |
    (round == 2 & count_total_round > 30)
  )

dim(data_ind)
```

```
## [1] 88 29
```

```

## copy data frame and assign `male = "combined"`
data_comb <- data_ind %>% mutate(male = "combined")

data <- bind_rows(data_ind, data_comb)

```

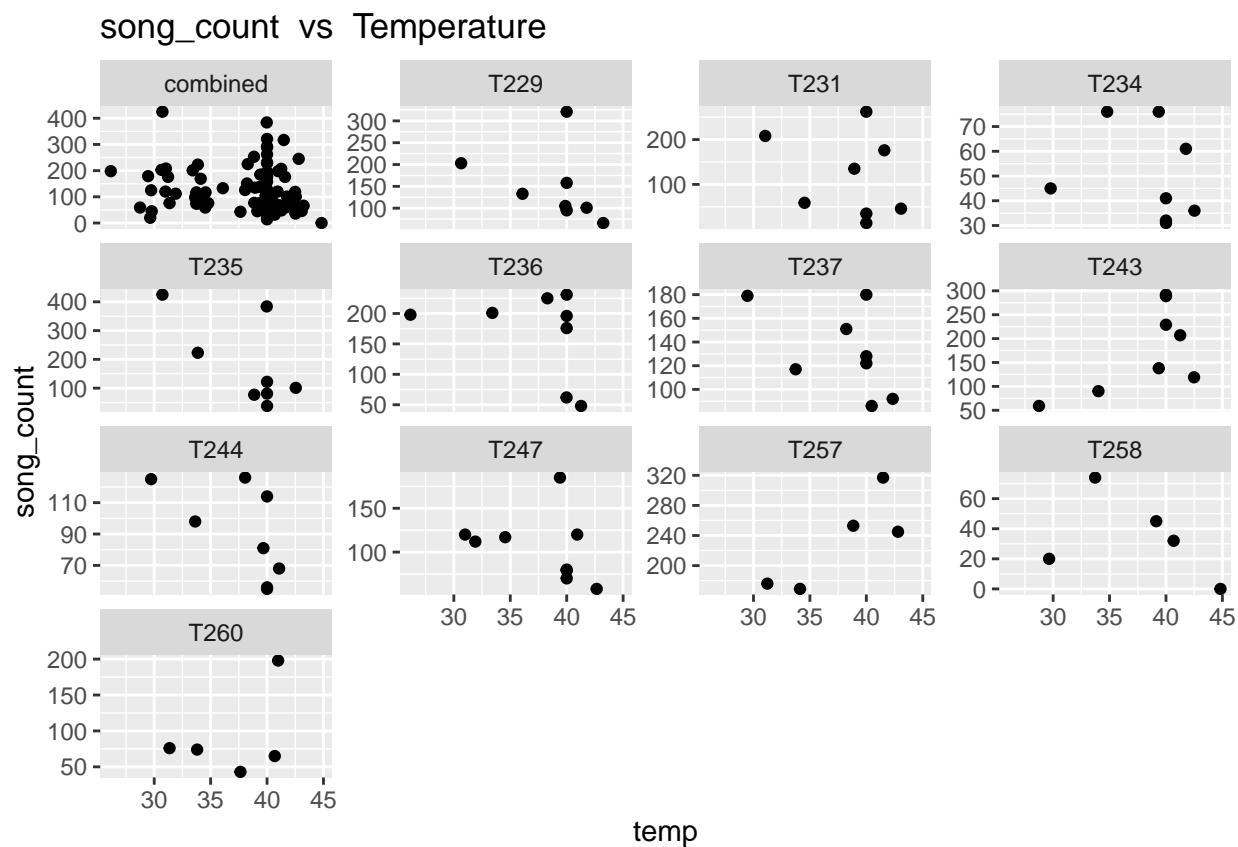
## Plot Filtered Data

```

xlab <- "Temperature"
ylab <- "song_count"

plot_temp_data <-
  ggplot(data) +
  aes(x = temp,
      y = song_count) +
  facet_wrap("male", scales = "free_y") +
  geom_point() +
  labs(title = paste( ylab, " vs ", xlab))
last_plot()

```



# First Analysis

## Formal Model Fits to song\_count

### Data Set Up

- Filter data

```
## Filter the data a bit more to make things easier
males_in_round_3 <- data_ind %>% select(male, round) %>% unique() %>% filter(round == 3) %>% pull(male)

data <- data_ind %>%
  ##   filter( male %in% c("T229", "T234", "T244", "T243", "T247", "T258")) %>%
  filter(male %in% males_in_round_3) %>% ## This is apparently unnecessary
  mutate()

dim(data)

## [1] 88 29
```

### Fit modifiedgaussian

- rTPC model: modifiedgaussian
- male as a factor

```
fit.file <- file.path(output_dir, "mgauss_nb.Rda")

refit.model <- FALSE

nlform <-
  bf(song_count ~ rmax*exp(-0.5*(fabs(temp - topt)/a)^b),
    rmax ~ male -1,
    topt ~ 1,
    a ~ 1,
    b ~ 1,
    nl = TRUE)

iter = 10000
warmup = floor(iter/3)
max_treedepth = 15

if(!file.exists(fit.file) | refit.model) {

  fit_list = list()

  for(prior_index in 1:2) {

    print(paste("fitting prior_index: ", prior_index))

    switch(prior_index,
```

```

{
  ## bounded normals
  nlprior <- c(prior(normal(400, 400), nlpar = "rmax", lb = 0),
                prior(normal(35, 20), nlpar = "topt", lb = 0, ub = 45),
                prior(normal(0, 3), nlpar = "a", lb = 0),
                prior(normal(0, 3), nlpar = "b", lb = 0)),

  ## wider bounded normals
  nlprior <- c(prior(normal(400, 800), nlpar = "rmax", lb = 0),
                prior(normal(35, 20), nlpar = "topt", lb = 0, ub = 45),
                prior(normal(0, 10), nlpar = "a", lb = 0),
                prior(normal(0, 6), nlpar = "b", lb = 0)),

  ## gamma(alpha = a , beta = b); E(x) = a/b; var(x) = a/b^2
  nlprior <- c(prior(normal(400, 800), nlpar = "rmax", lb = 0),
                prior(normal(35, 20.), nlpar = "topt", lb = 0),
                prior(gamma(1, 0.01), nlpar = "a", lb = 0),
                prior(gamma(1, 0.01), nlpar = "b", lb = 0)),

  fit_list[[prior_index]] <- brm(formula = nlform,
                                 data = data,
                                 family = negbinomial(link = "identity",
                                                       link_shape = "identity"),
                                 prior = nlprior,
                                 chains = 4, cores = 4,
                                 iter = iter,
                                 warmup = warmup,
                                 save_model = "model_mgauss_nb_1.stan",
                                 control = list(adapt_delta = 0.96))

}

mgauss_nb <- fit_list

save(mgauss_nb, fit.file)
} else {
  load(fit.file)
}
}

## [1] "fitting prior_index: 1"

## Compiling Stan program...

## Start sampling

## Warning: There were 5 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 11614 transitions after warmup that exceeded the maximum treedepth. Increase max
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

```

```

## Warning: Examine the pairs() plot to diagnose sampling problems

## [1] "fitting prior_index: 2"

## Compiling Stan program...
## Start sampling

## Warning: There were 12 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 6667 transitions after warmup that exceeded the maximum treedepth. Increase max_treedepth.
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: The largest R-hat is 1.59, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quantiles
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

## Error in save(mgauss_nb, fit.file): 'file' must be specified

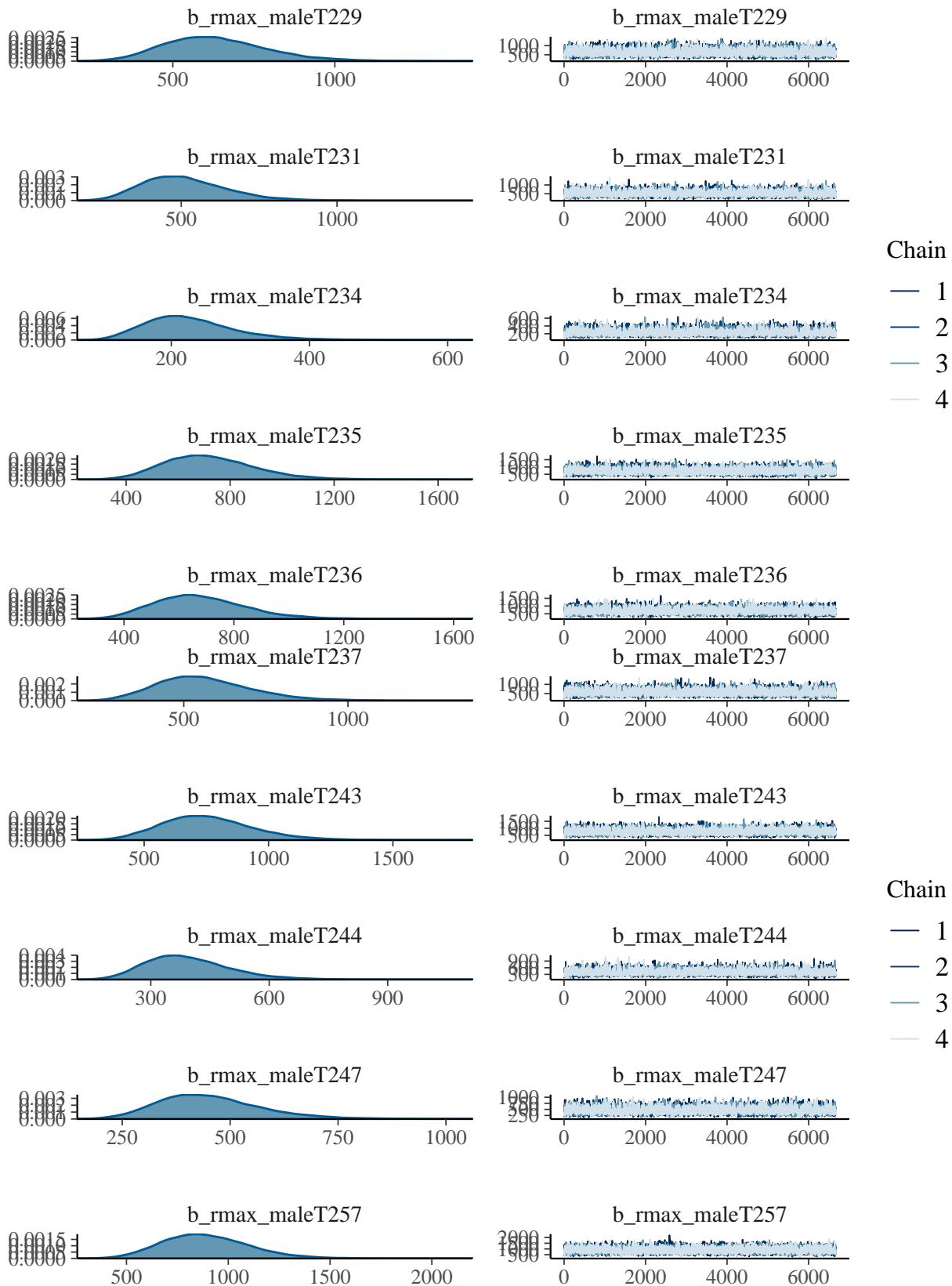
```

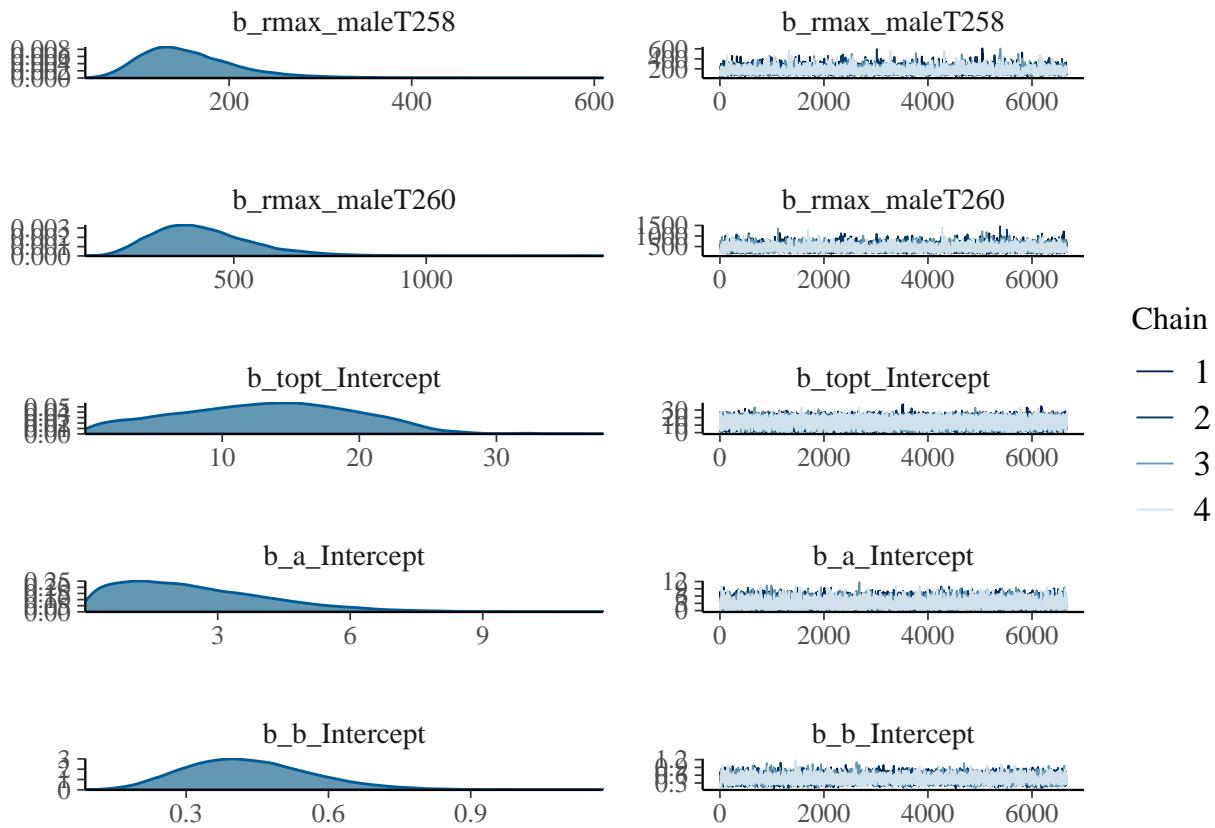
## Plot Fits

```

for(prior_index in 1:2) {
  model <- mgauss_nb[[prior_index]]
  plot_brms_fit(model)
  print(plot_stan_fit(model$fit))
}

```





```

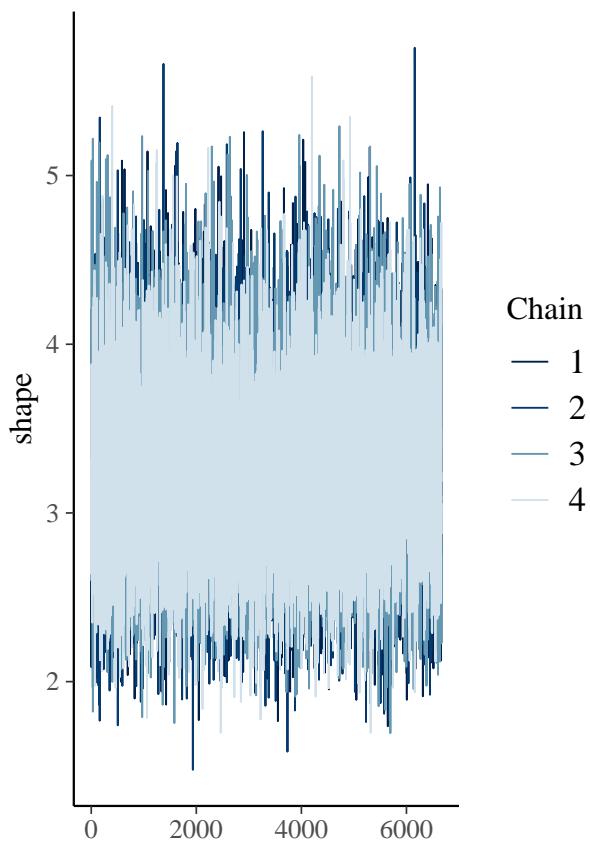
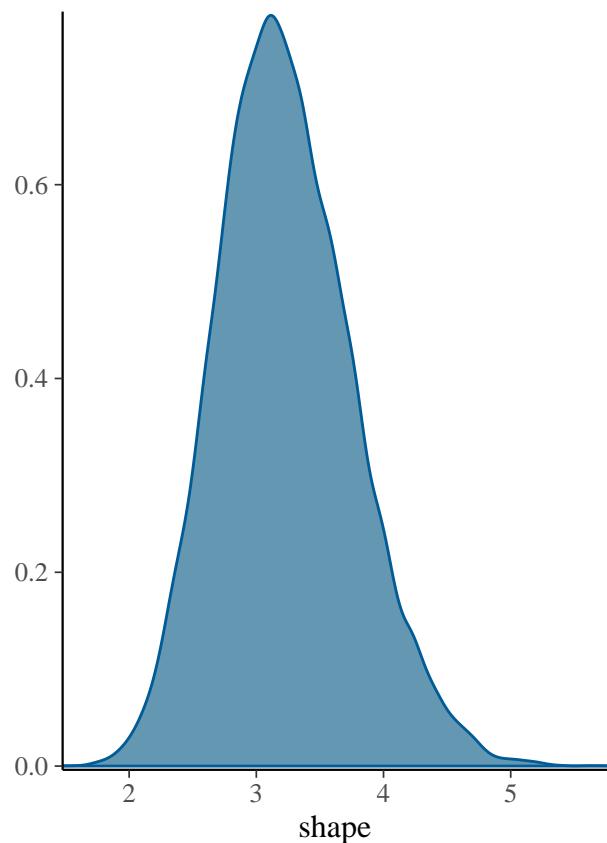
## 'pars' not specified. Showing first 10 parameters by default.

## ci_level: 0.8 (80% intervals)

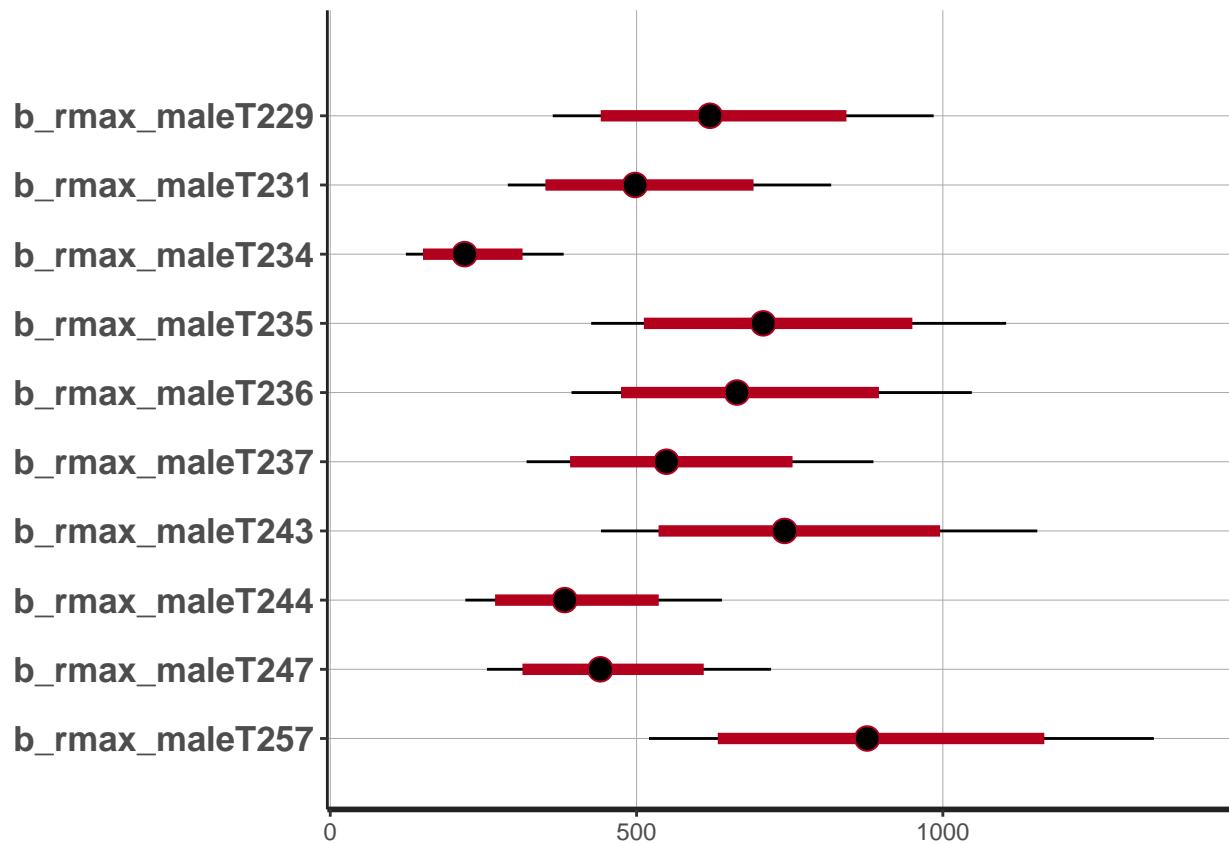
## outer_level: 0.95 (95% intervals)

## `summarise()` has grouped output by 'warmup'. You can override using the
## '.groups' argument.

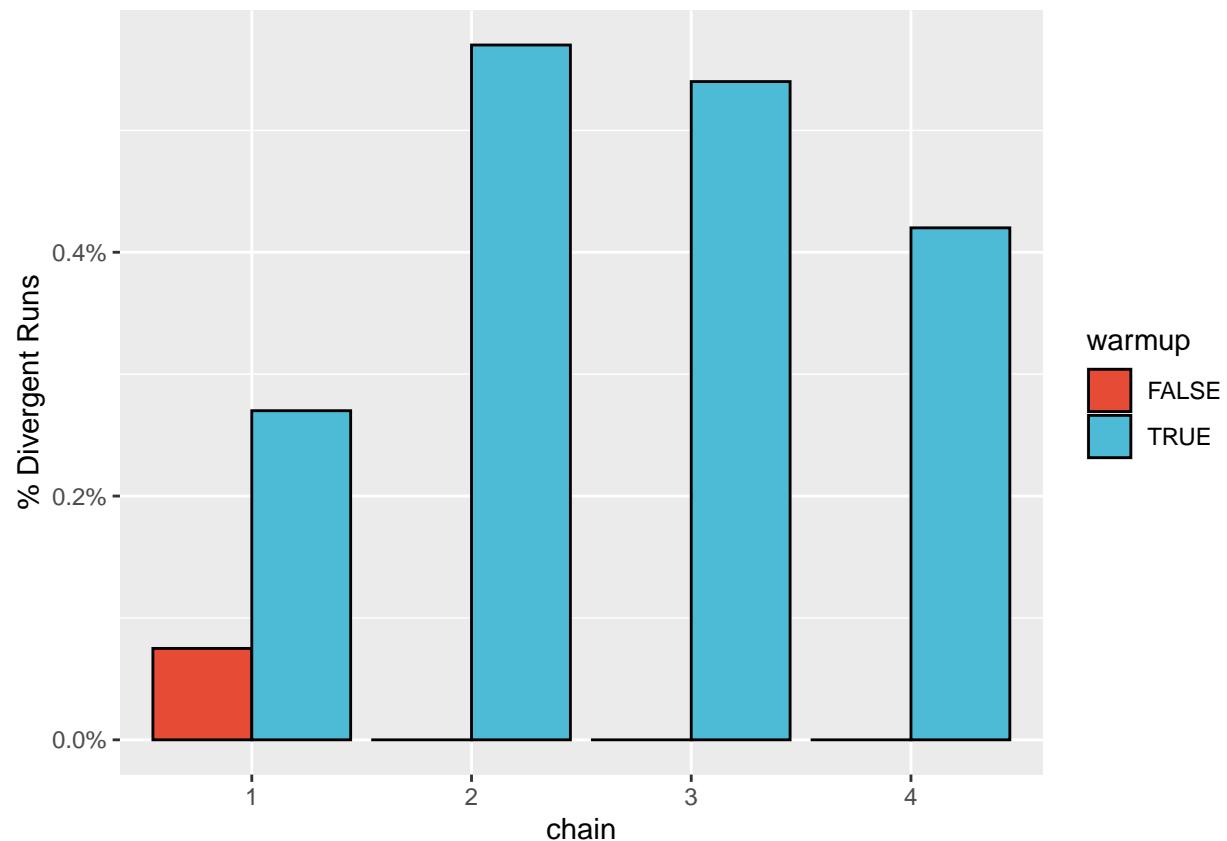
```



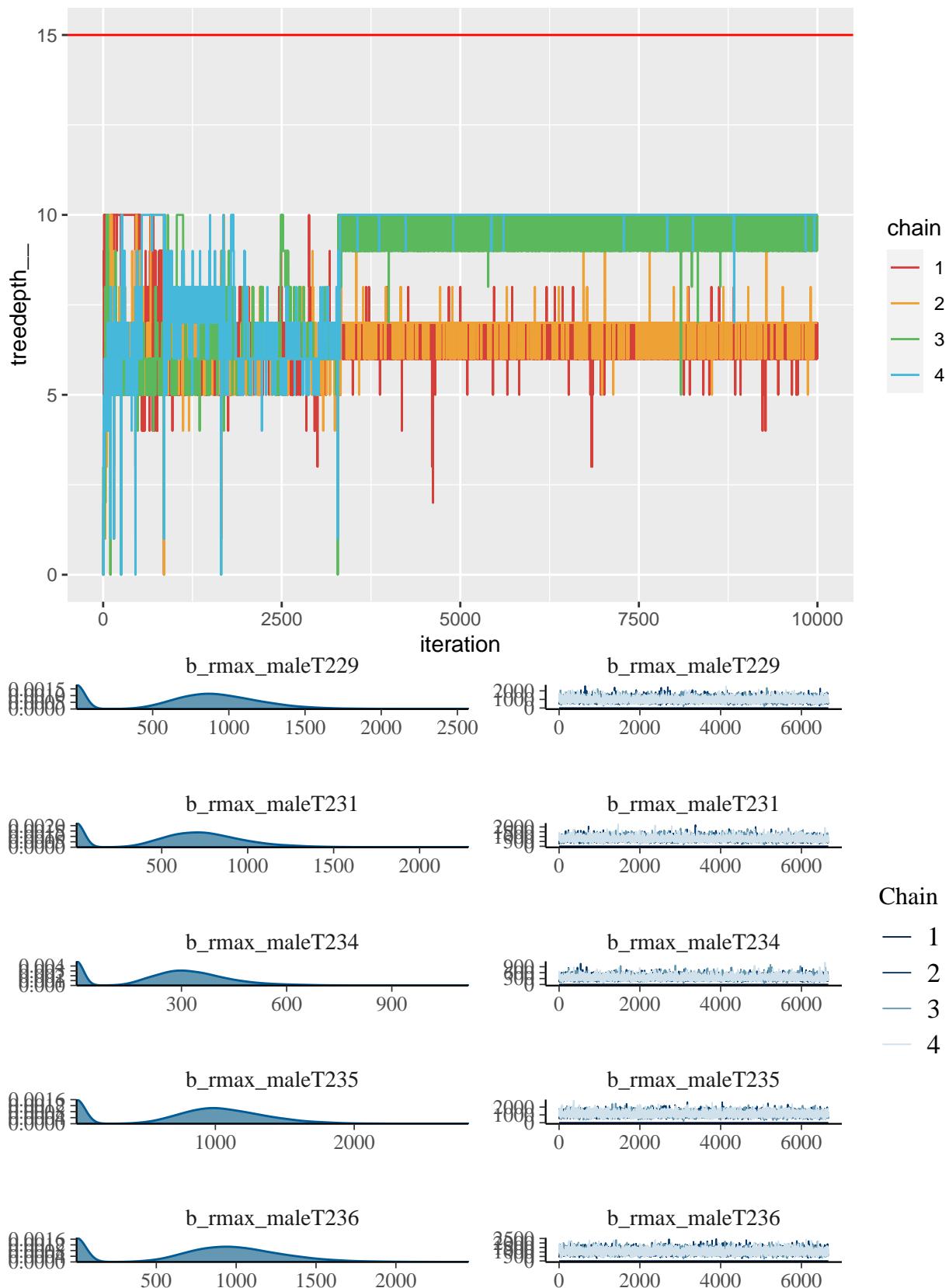
```
## [[1]]
```

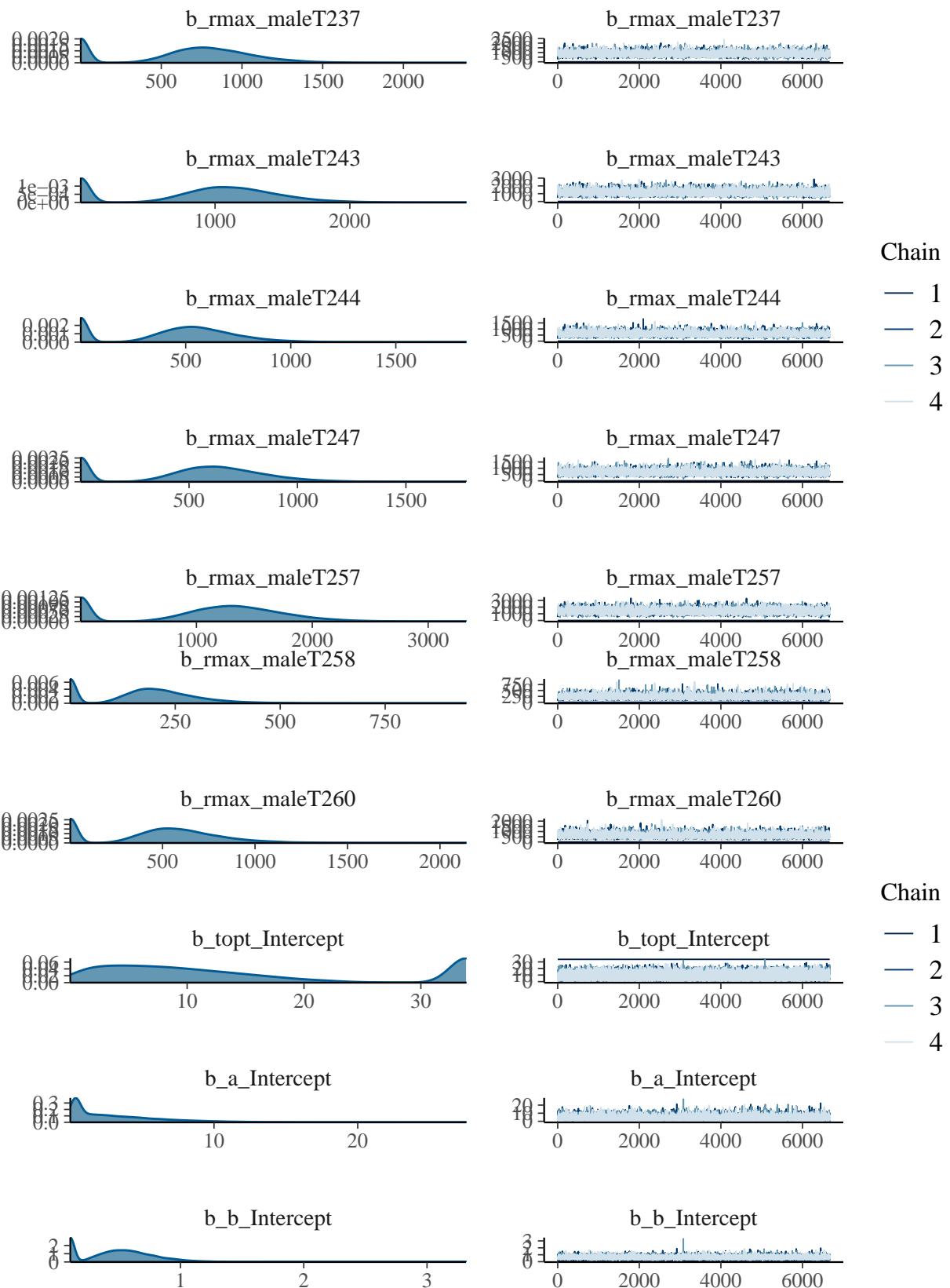


```
##  
## [[2]]
```



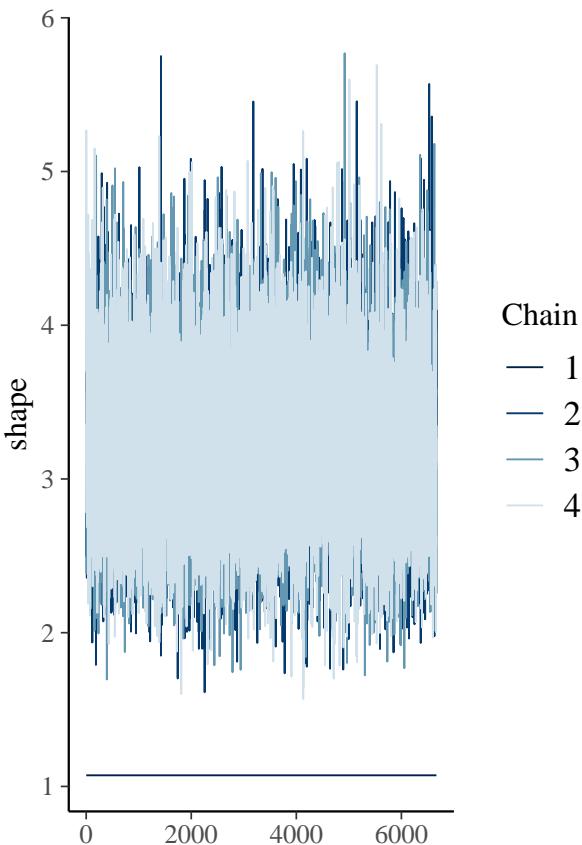
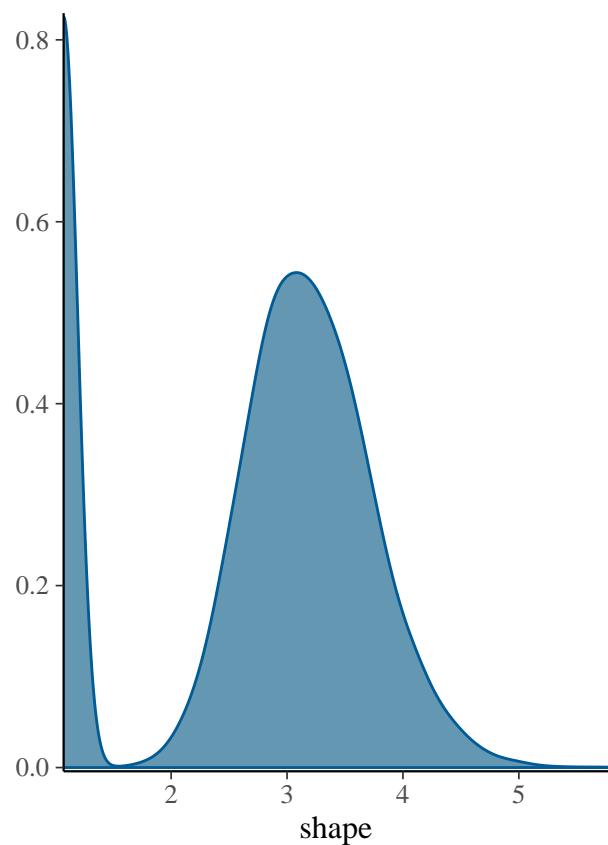
```
##  
## [[3]]
```



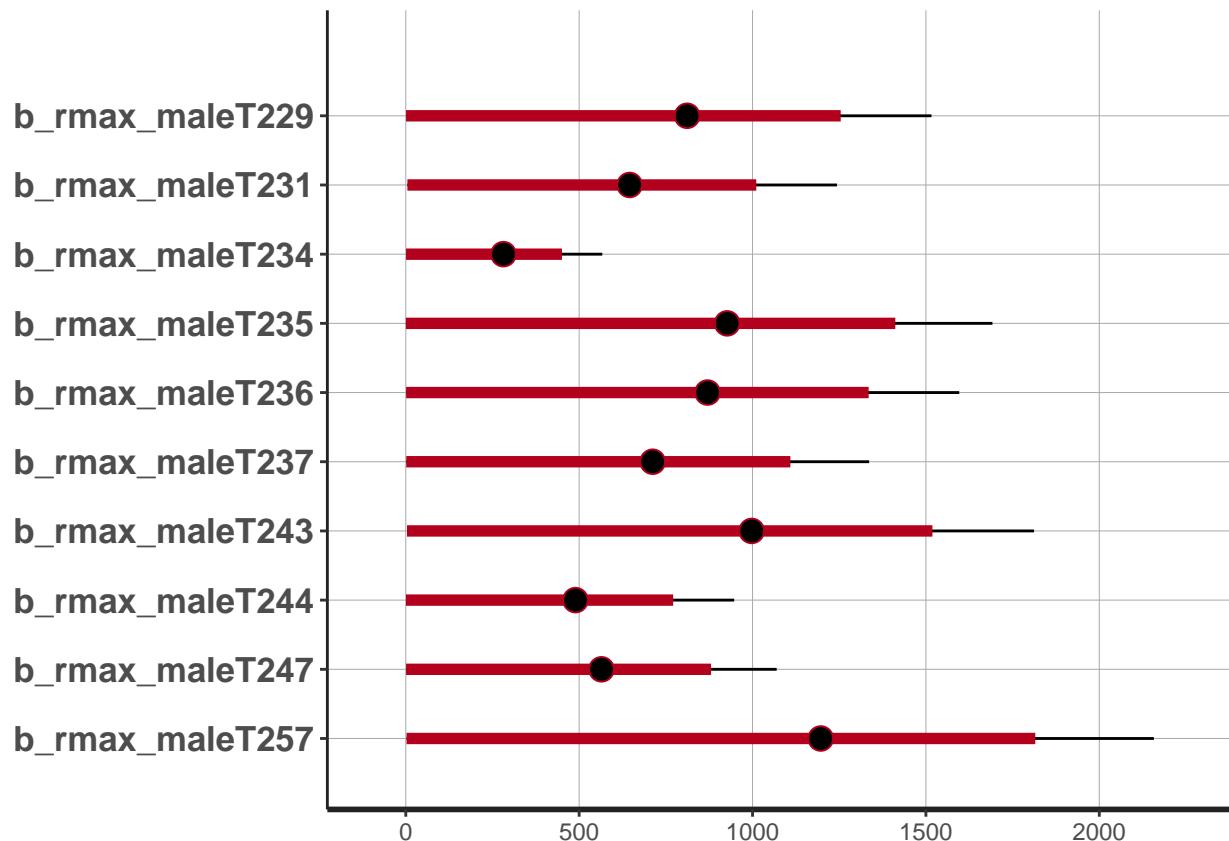


```
## 'pars' not specified. Showing first 10 parameters by default.
## ci_level: 0.8 (80% intervals)
```

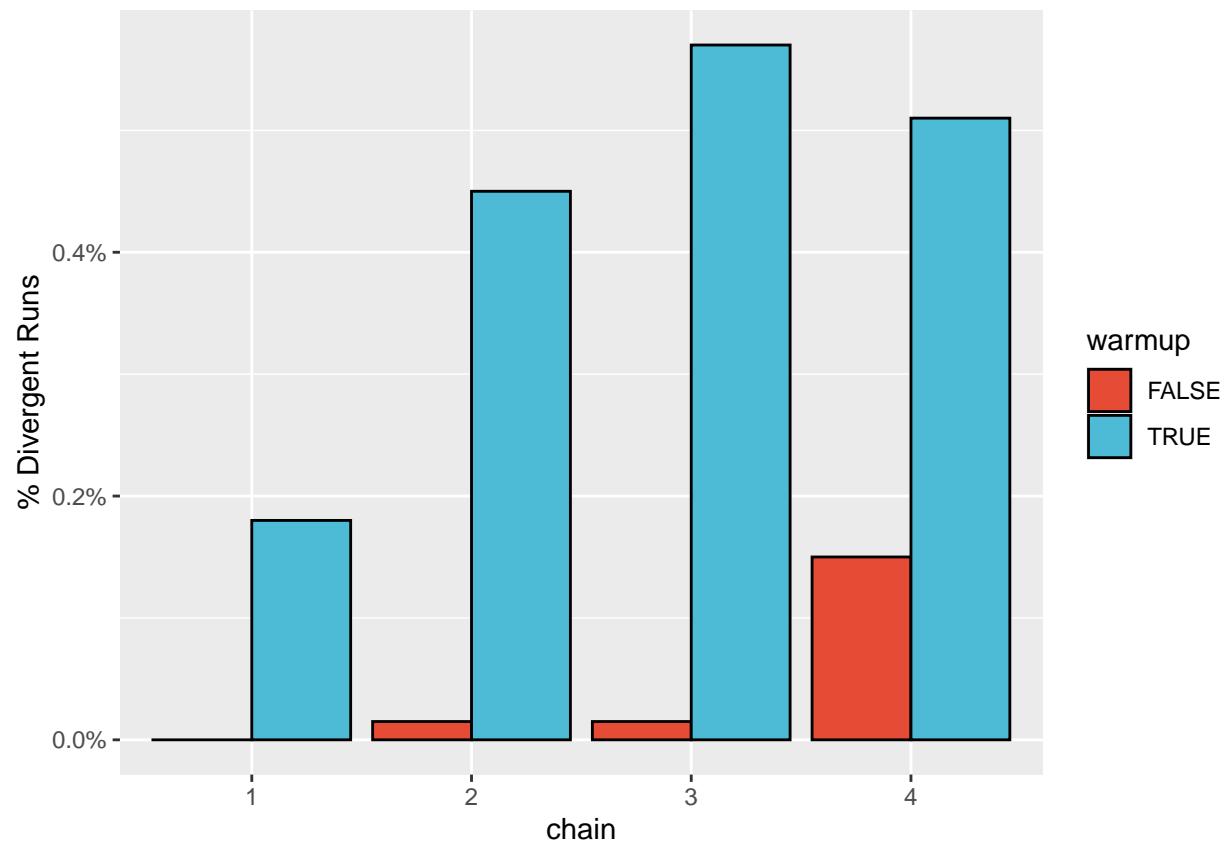
```
## outer_level: 0.95 (95% intervals)
## `summarise()` has grouped output by 'warmup'. You can override using the '.groups' argument.
```



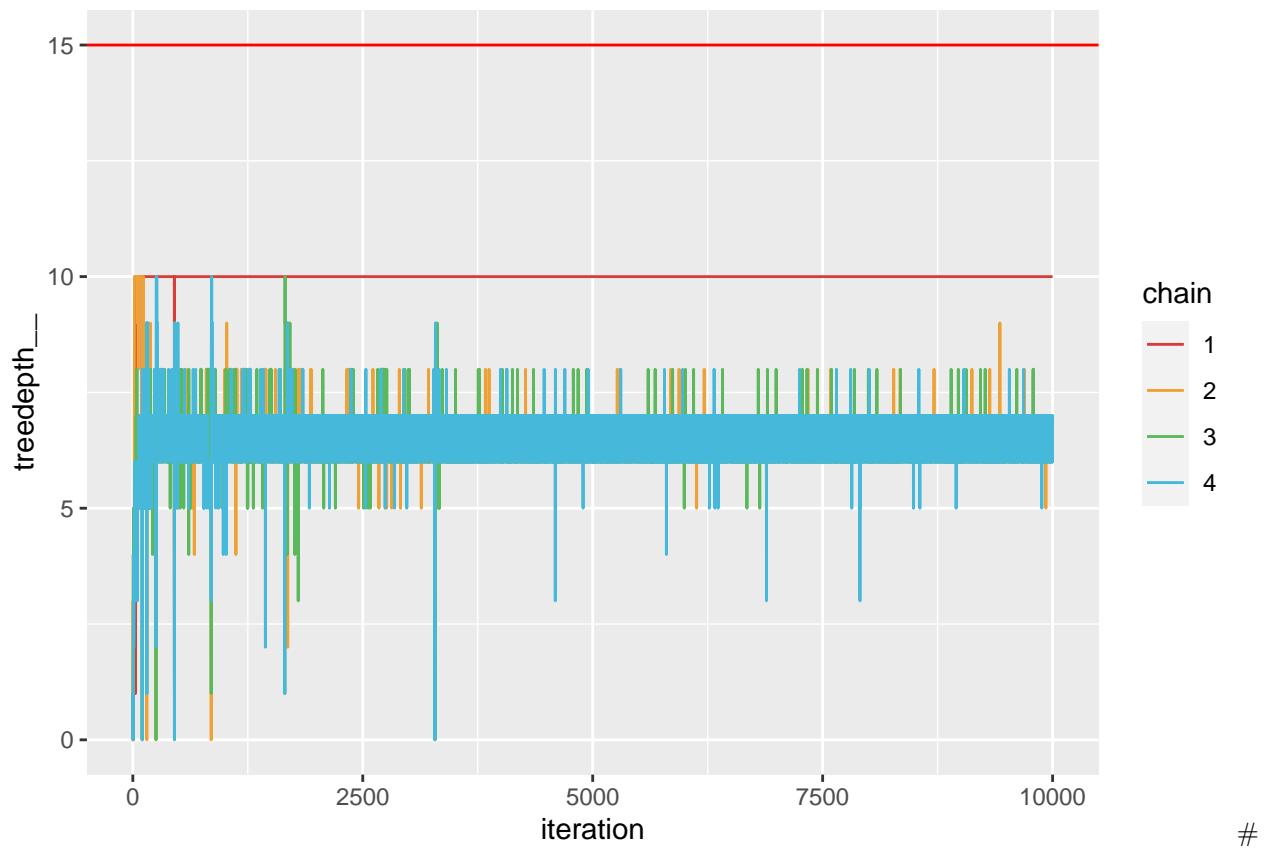
```
## [[1]]
```



```
##  
## [[2]]
```



```
##  
## [[3]]
```



End

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