```
library(htmltools)


############### 0 - safe choice A, 1 - risky choice B #####################
library(rstan); rstan_options(javascript=FALSE)
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.8, GitRev: 2e1f913d3ca3)
```

```
## 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)
```

```
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = T)


# Get list of files in 'data_2' folder with the pattern "riskytimed"
files <- dir(path = "data_2", pattern="riskytimed")

# Read all csv files in the list
data_list <- lapply(paste0("data_2/", files), read.table, header = TRUE, skip = 0,
fill = TRUE, sep= ";")

# Concatenate rows of all items in the list into a data frame
dat <- do.call("rbind", data_list)
```

```
# gamble characteristics
dat$eva = dat$oa1*dat$pa1+dat$oa2*dat$pa2 + dat$oa3*dat$pa3+dat$oa4*dat$pa4
dat$evb = dat$ob1*dat$pb1+dat$ob2*dat$pb2 + dat$ob3*dat$pb3+dat$ob4*dat$pb4
dat$evd = dat$evb - dat$eva
dat$sda = sqrt((dat$oa1-dat$eva)^2*dat$pa1 + (dat$oa2-dat$eva)^2*dat$pa2 + (dat$oa
3-dat$eva)^2*dat$pa3 + (dat$oa4-dat$eva)^2*dat$pa4)
dat$sdb = sqrt((dat$ob1-dat$evb)^2*dat$pb1 + (dat$ob2-dat$evb)^2*dat$pb2 + (dat$ob
3-dat$evb)^2*dat$pb3 + (dat$ob4-dat$evb)^2*dat$pb4)
dat$sdd = dat$sdb - dat$sda
dat$evdummy = ifelse(dat$evd>0,1,0)
```

```r
# transform to +/- 1; safe - 1, risky +1
dat$cho <- ifelse(dat$choice==0,-1,ifelse(dat$choice==1,1,NA))

ids <- unique(dat$id)
for(j in 1:length(ids)){
  dat$tid[dat$id==ids[j]] <- j
}
tids <- unique(dat$tid)
# only control data
control_dat <- dat[dat$cond=="control",]
# remove fast RTs
rcontrol_dat <- control_dat[control_dat$rt>1,]

# only condition no time pressure
dataList  = list(cho = rcontrol_dat$cho,rt = rcontrol_dat$rt, participant = rcontr
ol_dat$tid,N=nrow(rcontrol_dat),  L = length(tids),starting_point=0.5, evd = rcont
rol_dat$evd, sdd = rcontrol_dat$sdd)
```

```r
oa = as.matrix(rcontrol_dat[, c("oa1", "oa2", "oa3", "oa4")])
ob = as.matrix(rcontrol_dat[, c("ob1", "ob2", "ob3", "ob4")])
pa = as.matrix(rcontrol_dat[, c("pa1", "pa2", "pa3", "pa4")])
pb = as.matrix(rcontrol_dat[, c("pb1", "pb2", "pb3", "pb4")])
```

```r
dataList  = list(cho = rcontrol_dat$cho,rt = rcontrol_dat$rt, participant = rcontr
ol_dat$tid,N=nrow(rcontrol_dat),  L = length(tids),starting_point=0.5,
                 oa = as.matrix(rcontrol_dat[, c("oa1", "oa2", "oa3", "oa4")]),
                 ob = as.matrix(rcontrol_dat[, c("ob1", "ob2", "ob3", "ob4")]),
                 pa = as.matrix(rcontrol_dat[, c("pa1", "pa2", "pa3", "pa4")]),
                 pb = as.matrix(rcontrol_dat[, c("pb1", "pb2", "pb3", "pb4")])
                 )

parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu
_theta",'sd_threshold',"sd_alpha","sd_ndt", 'sd_theta', "alpha_sbj","threshold_sb
j","ndt_sbj",'theta_sbj',"log_lik")
```

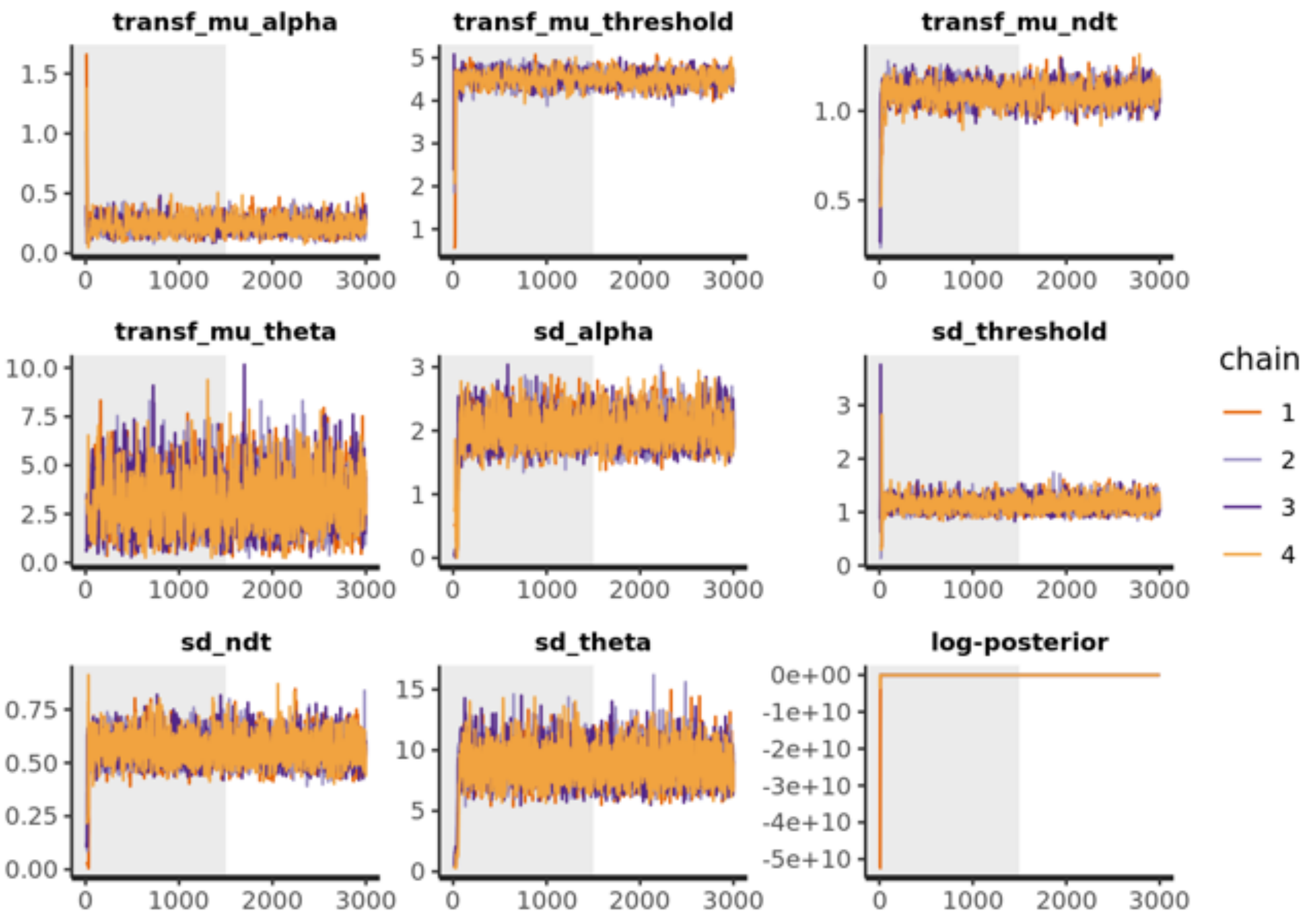
```r
initFunc <-function (i) {
  initList=list()
  for (ll in 1:i){
    initList[[ll]] = list(mu_alpha = runif(1,-1.4578,2.5413),
                          sd_alpha = runif(1,0,1),
                          mu_threshold = runif(1,-0.5, 2.5),
                          sd_threshold = runif(1,0,1),
                          mu_ndt = runif(1, -1.5, 0),
                          sd_ndt = runif(1, 0, 1),
                          mu_theta = runif(1,0,6),
                          sd_theta = runif(1,0,1),
                          z_alpha = runif(length(tids),-0.1,0.1),
                          z_theta = runif(length(tids),-0.1,0.1),
                          z_threshold = runif(length(tids),-0.1,0.1),
                          z_ndt = runif(length(tids),-0.1,0.1)

    )
  }

  return(initList)
}
```

```r
m <- stan_model("EU_Baseline.stan")
dsamples <- sampling(m,
                data=dataList,
                pars=parameters,
                iter=3000,
                chains=4,#If not specified, gives random inits
                init = initFunc(4),
                warmup = 1500,  # Stands for burn-in; Default = iter/2
                seed = 12, # Setting seed; Default is random seed
                refresh = 0
                )
```

```
## Warning: There were 19 transitions after warmup that exceeded the maximum treed
epth. Increase max_treedepth above 10. See
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

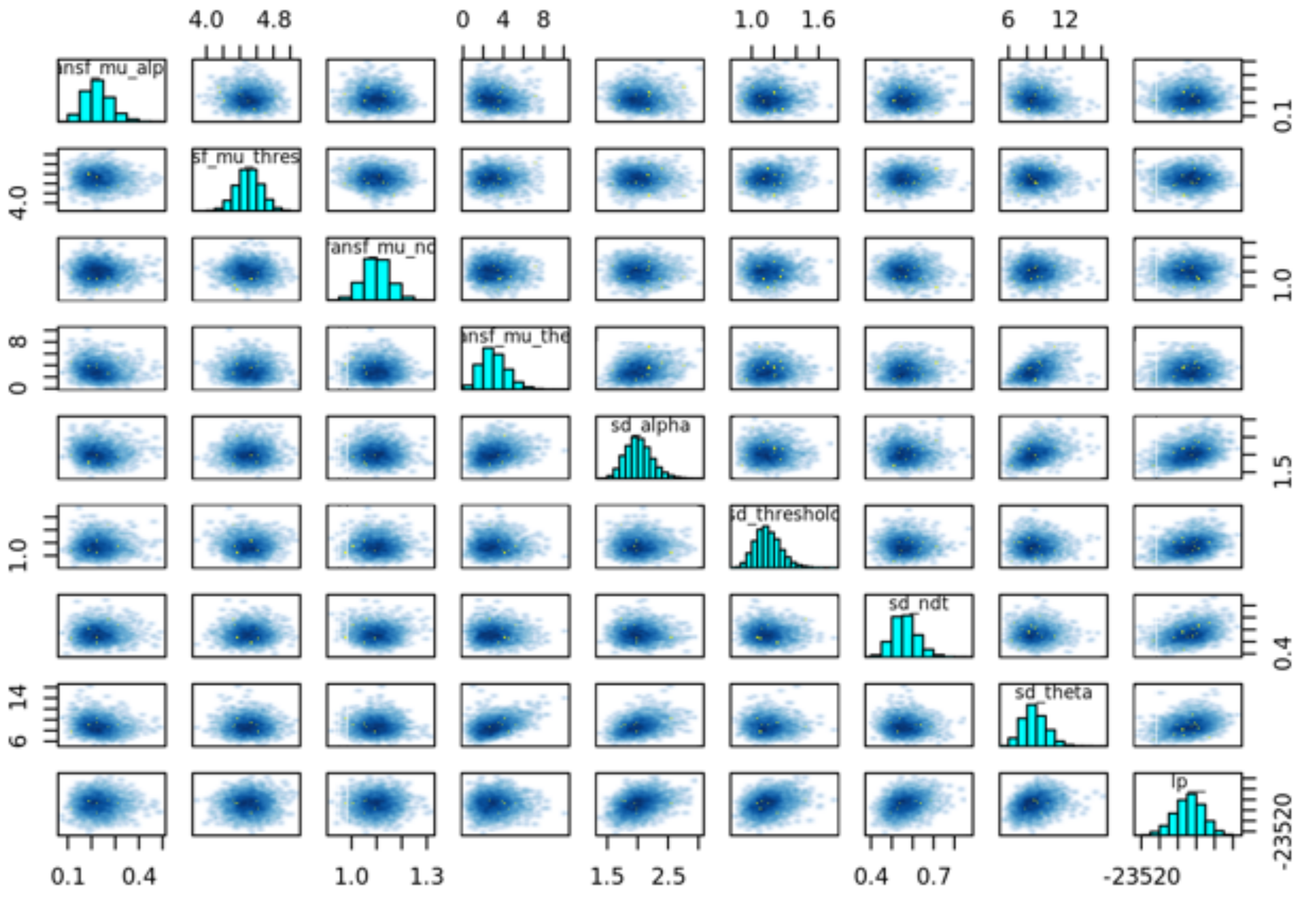
```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```r
rstan::traceplot(dsamples, pars=c("transf_mu_alpha","transf_mu_threshold","transf_
mu_ndt","transf_mu_theta", "sd_alpha","sd_threshold","sd_ndt",'sd_theta', "lp__"),
inc_warmup = TRUE, nrow = 3)
```



```
pairs(dsamples, pars = c( "transf_mu_alpha","transf_mu_threshold","transf_mu_nd
t","transf_mu_theta", "sd_alpha","sd_threshold","sd_ndt",'sd_theta', "lp__"))
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



```
print(dsamples, pars = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt","
transf_mu_theta", "sd_alpha","sd_threshold","sd_ndt",'sd_theta', "lp__"))
```

```
## Inference for Stan model: EU_Baseline.
## 4 chains, each with iter=3000; warmup=1500; thin=1;
## post-warmup draws per chain=1500, total post-warmup draws=6000.
##
##                          mean se_mean    sd      2.5%       25%       50%
## transf_mu_alpha          0.23    0.00  0.06      0.13      0.19      0.22
## transf_mu_threshold      4.50    0.01  0.15      4.21      4.41      4.50
## transf_mu_ndt            1.10    0.00  0.05      1.00      1.06      1.10
## transf_mu_theta          3.07    0.04  1.29      0.94      2.13      2.95
## sd_alpha                 2.02    0.01  0.22      1.64      1.87      2.01
## sd_threshold             1.15    0.00  0.12      0.95      1.06      1.14
## sd_ndt                   0.56    0.00  0.06      0.46      0.52      0.56
## sd_theta                 8.80    0.03  1.30      6.55      7.90      8.69
## lp__                -23467.93    0.49 15.68 -23500.04 -23478.41 -23467.37
##                           75%     97.5% n_eff Rhat
## transf_mu_alpha          0.26      0.35   715 1.01
## transf_mu_threshold      4.60      4.80   554 1.00
## transf_mu_ndt            1.13      1.20   575 1.00
## transf_mu_theta          3.87      5.95   955 1.00
## sd_alpha                 2.16      2.50  1111 1.00
## sd_threshold             1.22      1.40  1014 1.00
## sd_ndt                   0.60      0.69  1357 1.01
## sd_theta                 9.61     11.60  1925 1.00
## lp__                -23457.32 -23438.00  1010 1.00
##
## Samples were drawn using NUTS(diag_e) at Tue Oct 17 04:39:25 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```r
library(bayesplot)
```

```
## This is bayesplot version 1.10.0
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

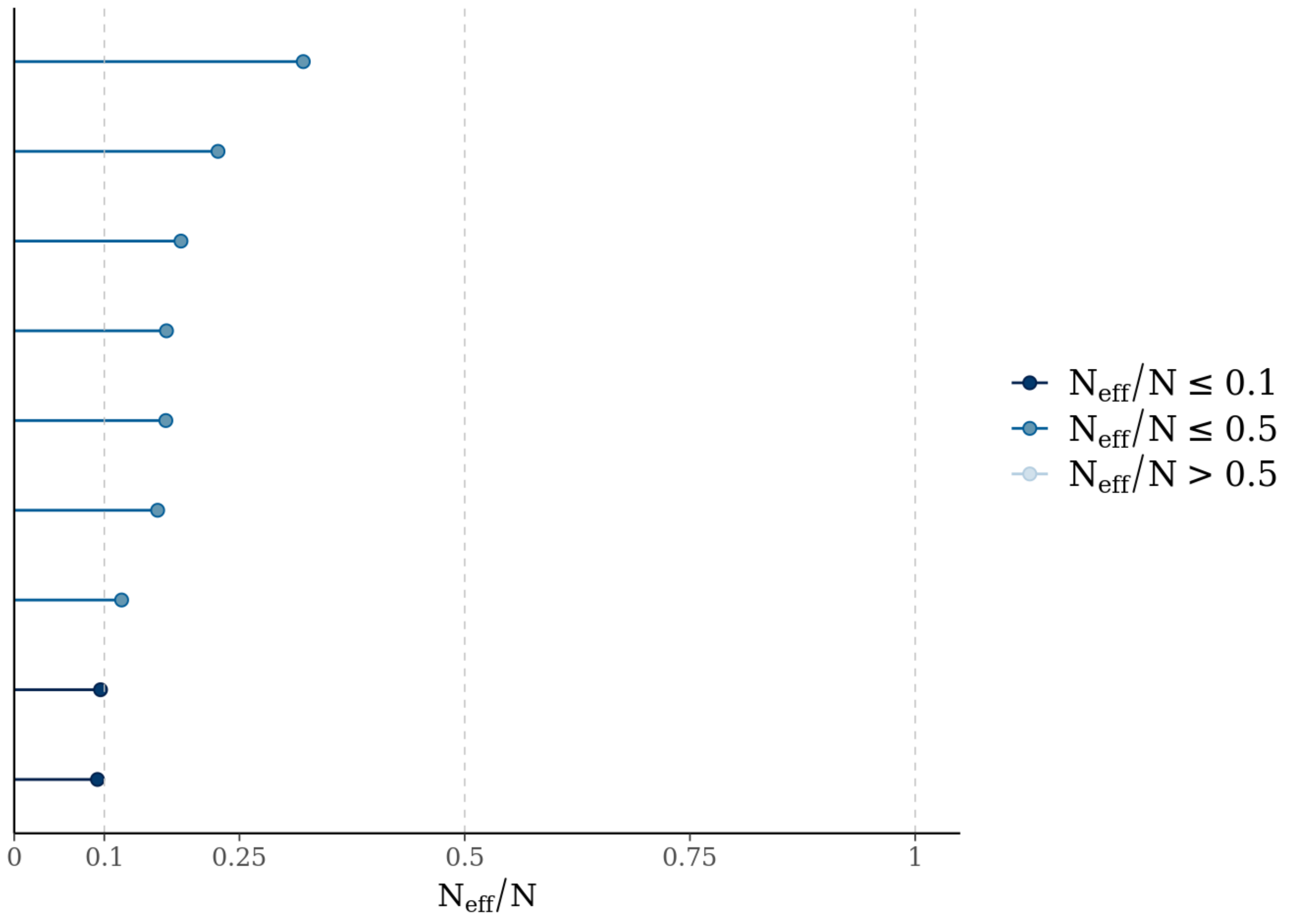
```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```
ratios_cp <- neff_ratio(dsamples, pars = c("transf_mu_alpha","transf_mu_theta", "t
ransf_mu_threshold","transf_mu_ndt", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_thet
a', "lp__"))
df_ratios_cp <- as.data.frame(ratios_cp)
print(df_ratios_cp)
```

```
##                          ratios_cp
## transf_mu_alpha     0.11918999
## transf_mu_theta     0.15917447
## transf_mu_threshold 0.09238789
## transf_mu_ndt       0.09581390
## sd_threshold        0.16900667
## sd_alpha            0.18508883
## sd_ndt              0.22617162
## sd_theta            0.32084126
## lp__                0.16829710
```

```
mcmc_neff(ratios_cp, size = 2)
```



```r
library(ggplot2)
library(tidyverse) # for the gather function
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ─
─
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.1     ✔ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ─
─
## ✖ tidyr::extract() masks rstan::extract()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conf
licts to become errors
```

```r
samples_matrix <- as.matrix(dsamples)
means <- colMeans(samples_matrix)
hpd_interval <- t(apply(samples_matrix, 2, function(x) quantile(x, probs=c(0.025,
0.975))))



parameters <- c("transf_mu_alpha", "transf_mu_theta", "transf_mu_threshold",
                "transf_mu_ndt")

# Reshape data to a long format
df_long <- as.data.frame(samples_matrix) %>%
  gather(key = "parameter", value = "value", parameters)
```

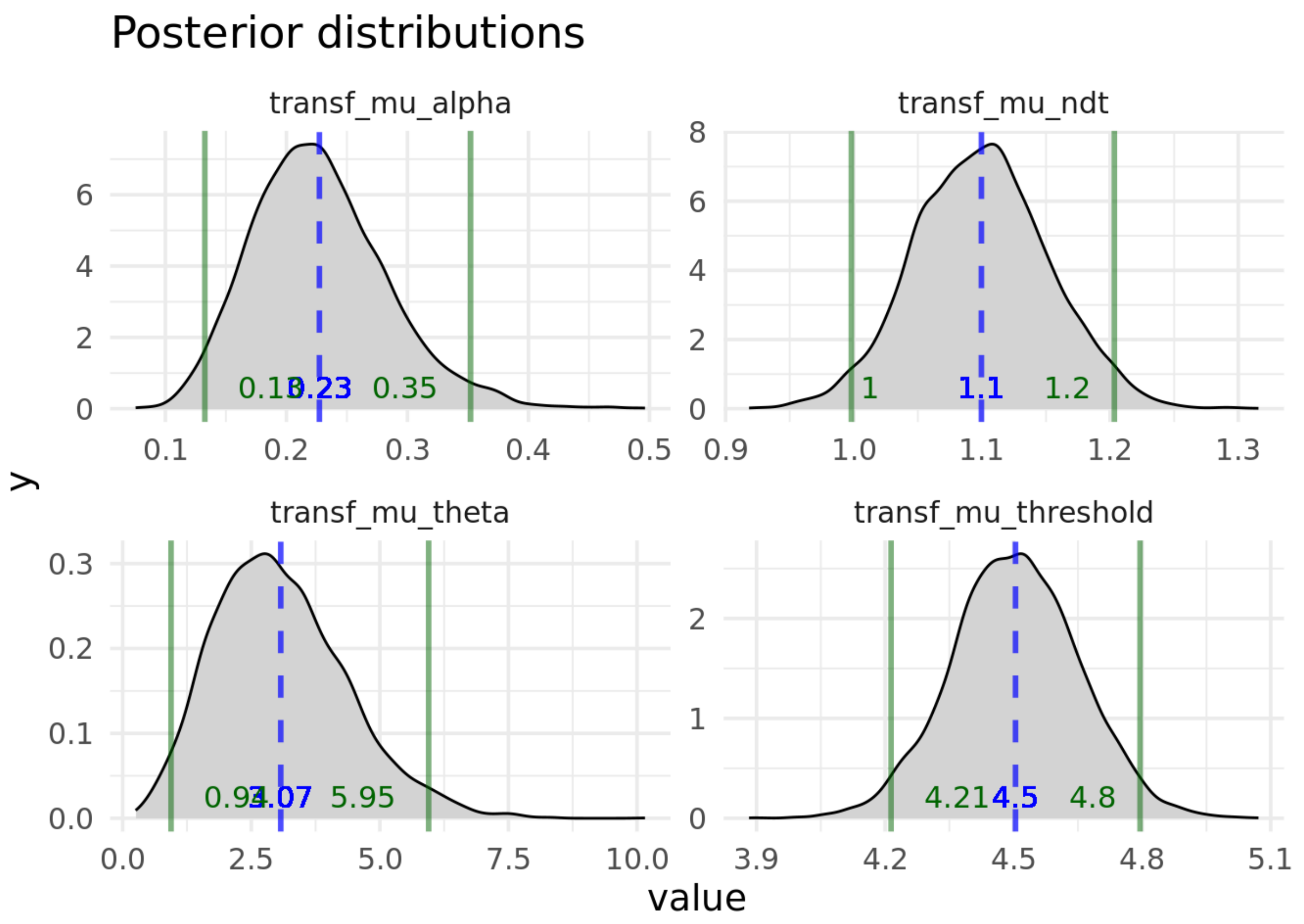
```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.
1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(parameters)
##
##   # Now:
##   data %>% select(all_of(parameters))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# Convert hpd_interval to a data frame and name the columns
hpd_interval_sub <- hpd_interval[parameters, ]
hpd_df <- as.data.frame(hpd_interval_sub)
colnames(hpd_df) <- c("lower", "upper")
rownames(hpd_df) <- parameters
hpd_df$parameter <- rownames(hpd_df)




# Aesthetic enhancements
theme_set(theme_minimal(base_size = 14)) # Set the default theme

custom_palette <- c("density_fill" = "lightgray",
                    "mean_line" = "blue",
                    "hpd_line" = "darkgreen")

# Add text labels for mean, lower, and upper HPD values
df_long <- df_long %>%
  group_by(parameter) %>%
  mutate(mean = means[parameter])

hpd_df <- hpd_df %>%
  mutate(mid = (lower + upper) / 2)

p <- ggplot(df_long, aes(x = value)) +
  geom_density(aes(fill = "density_fill")) +
  scale_fill_manual(values = custom_palette, guide = FALSE) +
  geom_vline(aes(xintercept = mean, color = "mean_line"), linetype = "dashed", siz
e = 1, alpha = 0.7) +
  geom_text(data = df_long, aes(x = mean, y = 0, label = round(mean, 2)), vjust =
-0.5, hjust = 0.5, size = 4, color = custom_palette["mean_line"]) +
  geom_vline(data = hpd_df, aes(xintercept = lower, color = "hpd_line"), linetype
= "solid", size = 1, alpha = 0.5) +
  geom_text(data = hpd_df, aes(x = lower, y = 0, label = round(lower, 2)), vjust =
-0.5, hjust = -0.5, size = 4, color = custom_palette["hpd_line"]) +
  geom_vline(data = hpd_df, aes(xintercept = upper, color = "hpd_line"), linetype
= "solid", size = 1, alpha = 0.5) +
  geom_text(data = hpd_df, aes(x = upper, y = 0, label = round(upper, 2)), vjust =
-0.5, hjust = 1.5, size = 4, color = custom_palette["hpd_line"]) +
  facet_wrap(~ parameter, scales = "free", ncol = 2) +
  scale_color_manual(values = custom_palette, guide = FALSE) +
  labs(title = "Posterior distributions")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
print(p)
```

```
## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This was deprec
ated in
## ggplot2 3.3.4.
## ℹ Please use "none" instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Posterior distributions
transf_mu_alpha
transf_mu_ndt
transf_mu_theta
transf_mu_threshold
y
value