```
############### 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)
```

```
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
```

```
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)
```

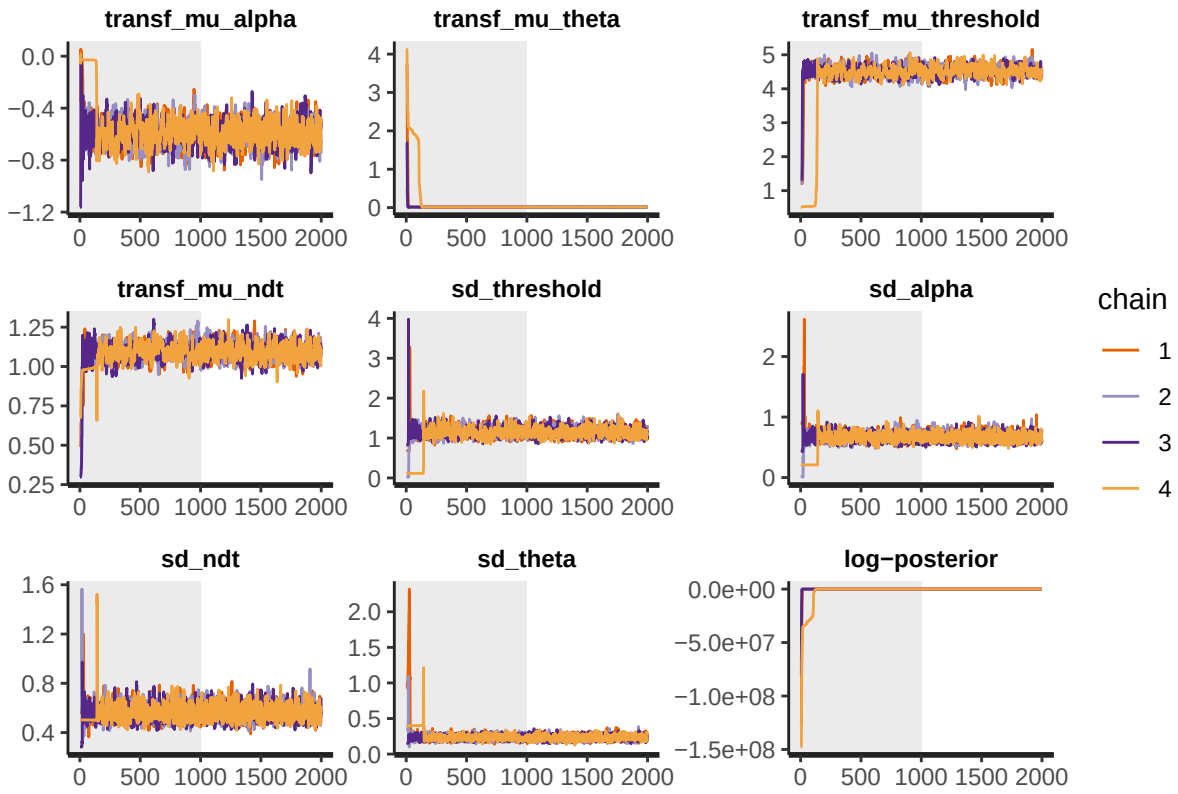
```
# gamble characteristics
dat$eva = dat$oa1*dat$pa1+dat$oa2*dat$pa2 + dat$oa3*dat$pa3+dat$oa4*dat$pa4
dat$evb = dat$ob1*dat$pb1+dat$ob2*dat$pb2 + dat$ob3*dat$pb3+dat$ob4*dat$pb4
dat$evd = dat$evb - dat$eva
dat$sda = sqrt((dat$oa1-dat$eva)^2*dat$pa1 + (dat$oa2-dat$eva)^2*dat$pa2 + (dat$oa3-dat$eva)^2*dat$pa3 +
dat$sdb = sqrt((dat$ob1-dat$evb)^2*dat$pb1 + (dat$ob2-dat$evb)^2*dat$pb2 + (dat$ob3-dat$evb)^2*dat$pb3 +
dat$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$rt, participant = rcontrol_dat$tid, N=nrow(r
```

```r
parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu_theta", 'sd_threshol

initFunc <-function (i) {
  initList=list()
  for (ll in 1:i){
    initList[[ll]] = list(mu_alpha = runif(1, -5, 5),
                          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,-4, 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)
}
```

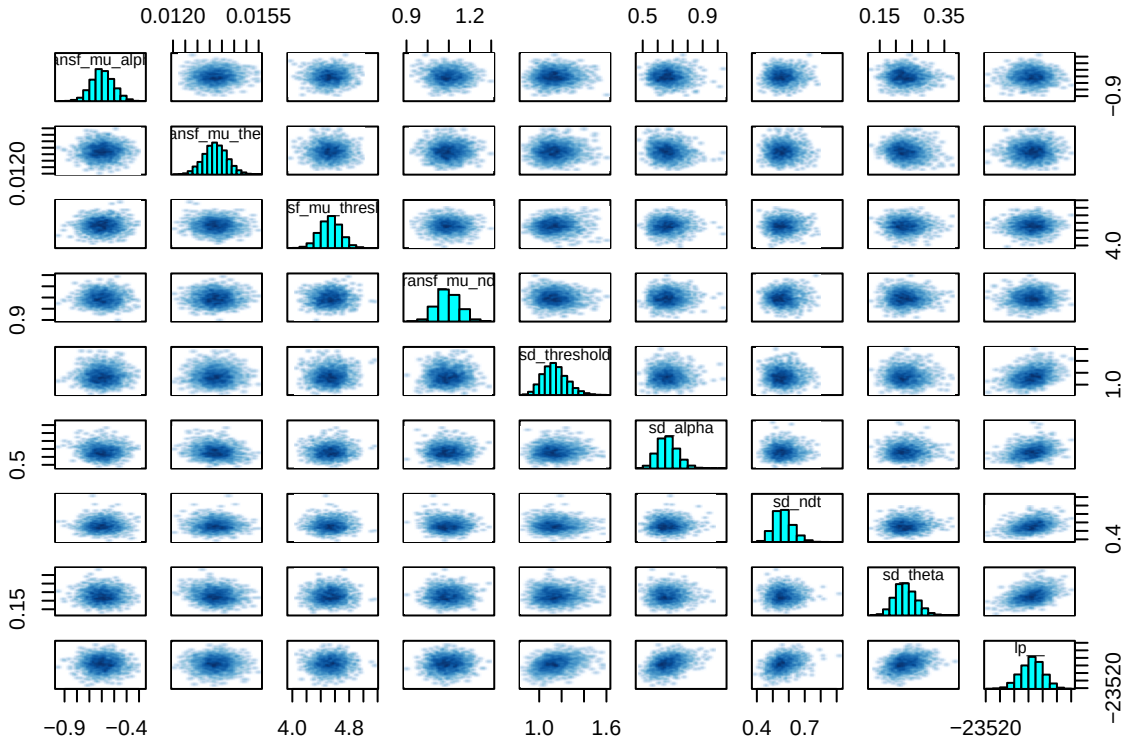
```r
m <- stan_model("MV_Baseline.stan")
dsamples <- sampling(m,
                data=dataList,
                pars=parameters,
                iter=2000,
                chains=4,#If not specified, gives random inits
                init = initFunc(4),
                warmup = 1000,  # Stands for burn-in; Default = iter/2
                seed = 12, # Setting seed; Default is random seed
                refresh = 0
                )
```

```r
#parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu_theta",'sd_threshol

rstan::traceplot(dsamples, pars=c("transf_mu_alpha","transf_mu_theta", "transf_mu_threshold","transf_mu
```




```
pairs(dsamples, pars = c("transf_mu_alpha","transf_mu_theta","transf_mu_threshold","transf_mu_ndt", 'sd
```

```
## 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_theta", "transf_mu_threshold","transf_mu_ndt", 's
```

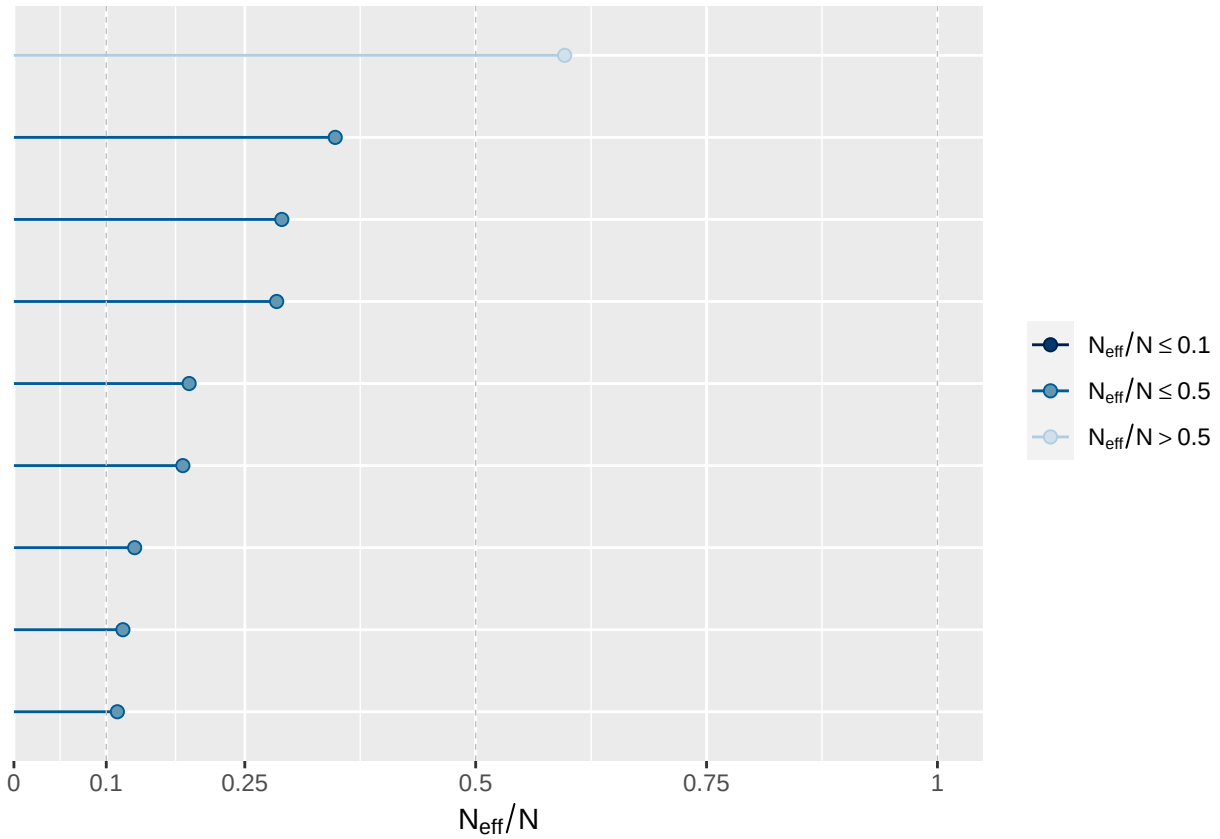
```
## Inference for Stan model: MV_Baseline.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##                          mean se_mean    sd      2.5%       25%       50%
## transf_mu_alpha         -0.59    0.00  0.09     -0.76     -0.65     -0.60
## transf_mu_theta          0.01    0.00  0.00      0.01      0.01      0.01
## transf_mu_threshold      4.53    0.01  0.15      4.24      4.43      4.53
## transf_mu_ndt            1.10    0.00  0.05      1.00      1.06      1.09
## sd_threshold             1.14    0.00  0.11      0.95      1.06      1.13
## sd_alpha                 0.67    0.00  0.07      0.55      0.62      0.66
## sd_ndt                   0.56    0.00  0.06      0.46      0.52      0.56
## sd_theta                 0.23    0.00  0.03      0.17      0.21      0.23
## lp__                -23455.11    0.59 16.05 -23487.16 -23465.53 -23454.88
##                          75%     97.5% n_eff Rhat
## transf_mu_alpha         -0.54     -0.42   523 1.00
## transf_mu_theta          0.01      0.01  2385 1.00
## transf_mu_threshold      4.63      4.82   448 1.00
## transf_mu_ndt            1.13      1.20   472 1.02
## sd_threshold             1.21      1.38   759 1.00
## sd_alpha                 0.71      0.81  1138 1.00
## sd_ndt                   0.60      0.69  1160 1.00
## sd_theta                 0.25      0.30  1392 1.00
```

```
## lp__                -23444.13 -23424.77   732 1.00
##
## Samples were drawn using NUTS(diag_e) at Mon Oct  9 22:55:28 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).
```

```
ratios_cp <- neff_ratio(dsamples, pars = c("transf_mu_alpha","transf_mu_theta", "transf_mu_threshold","
df_ratios_cp <- as.data.frame(ratios_cp)
print(df_ratios_cp)
```

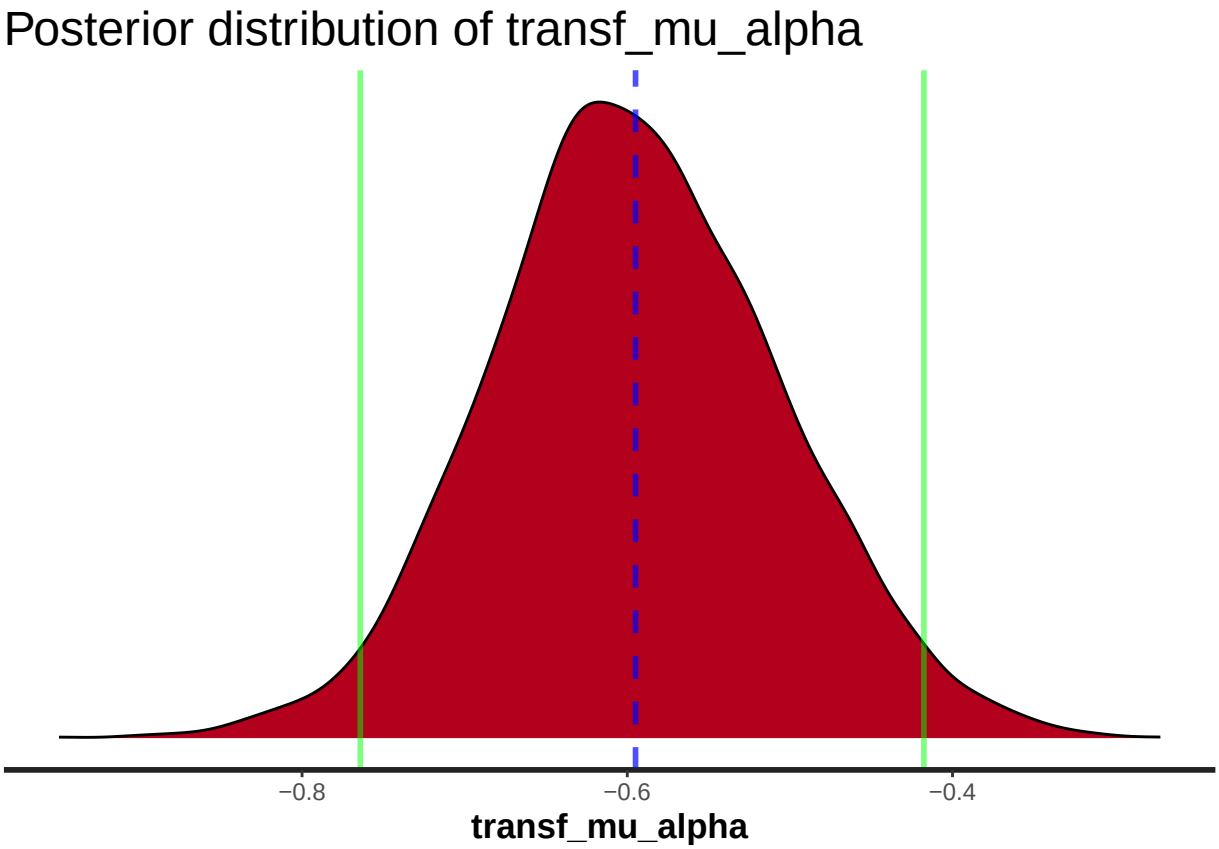
```
##                      ratios_cp
## transf_mu_alpha     0.1307550
## transf_mu_theta     0.5963391
## transf_mu_threshold 0.1120171
## transf_mu_ndt       0.1180074
## sd_threshold        0.1897305
## sd_alpha            0.2845123
## sd_ndt              0.2900805
## sd_theta            0.3479072
## lp__                0.1829276
```

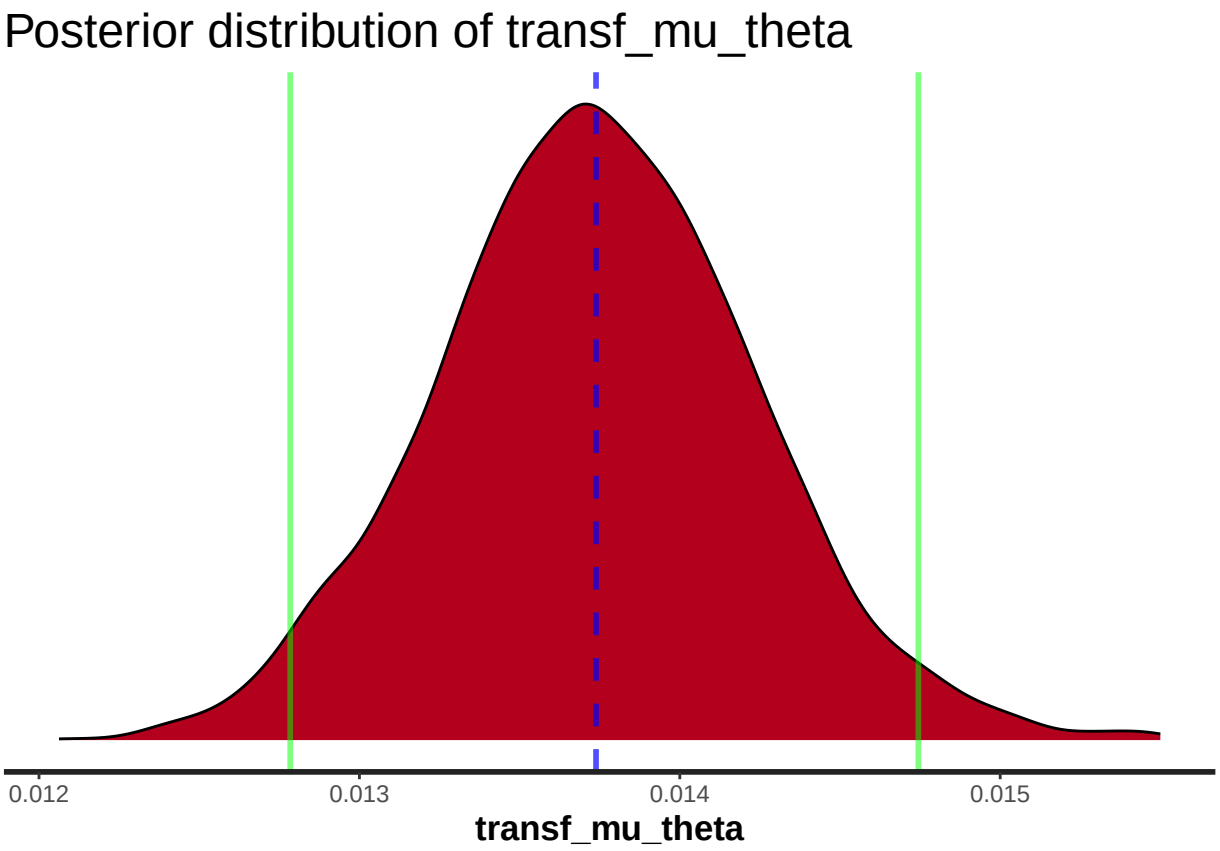
```
mcmc_neff(ratios_cp, size = 2)
```

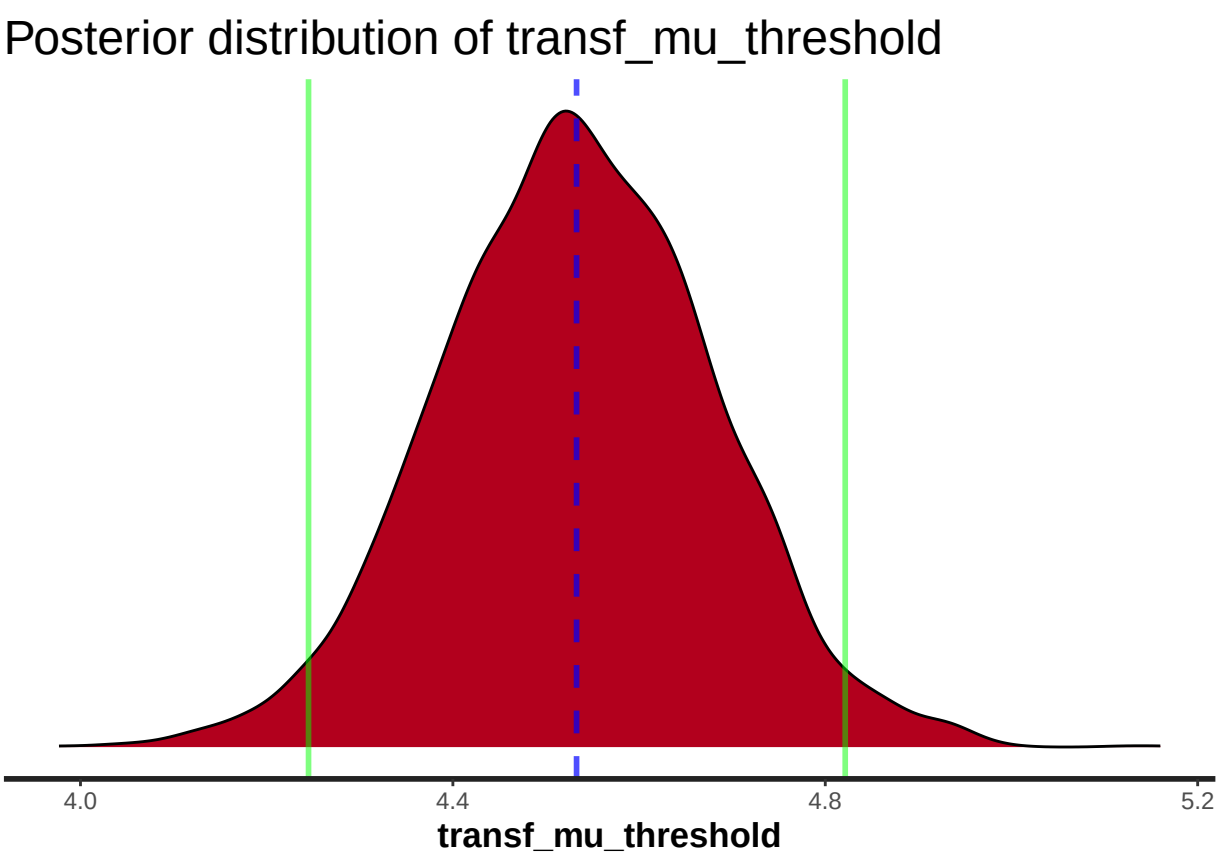


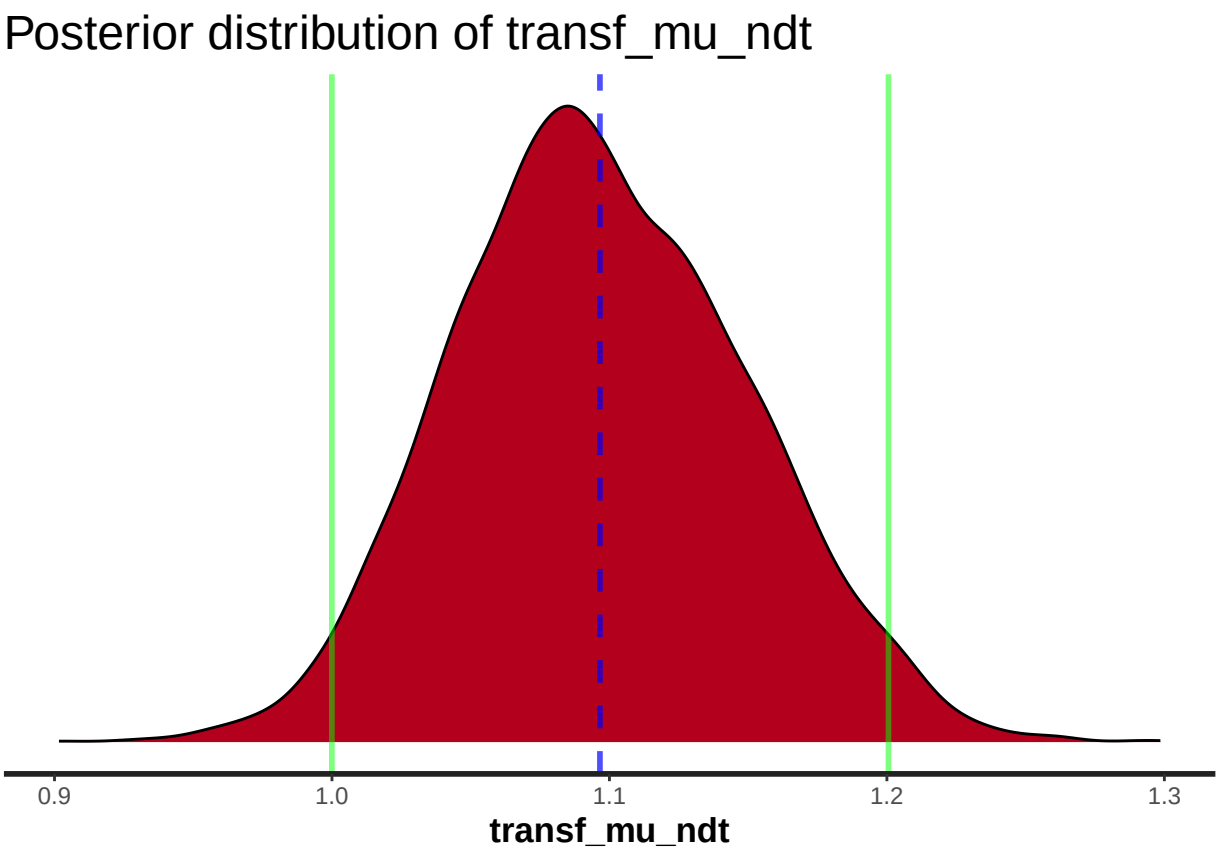
```r
library(ggplot2)

# Assuming 'fit' is your fitted model
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", "sd_threshold", "sd_alpha", "sd_ndt", "sd_theta")

# Loop through each parameter and plot
for (param in parameters) {
  p <- stan_dens(dsamples, pars = param)
  p <- p +
    geom_vline(aes(xintercept=means[param]), color="blue", linetype="dashed", size=1, alpha=0.7, show.l
    geom_vline(aes(xintercept=hpd_interval[param, 1]), color="green", linetype="solid", size=1, alpha=0
    geom_vline(aes(xintercept=hpd_interval[param, 2]), color="green", linetype="solid", size=1, alpha=0
    labs(title=paste("Posterior distribution of", param))

  print(p)
}
```

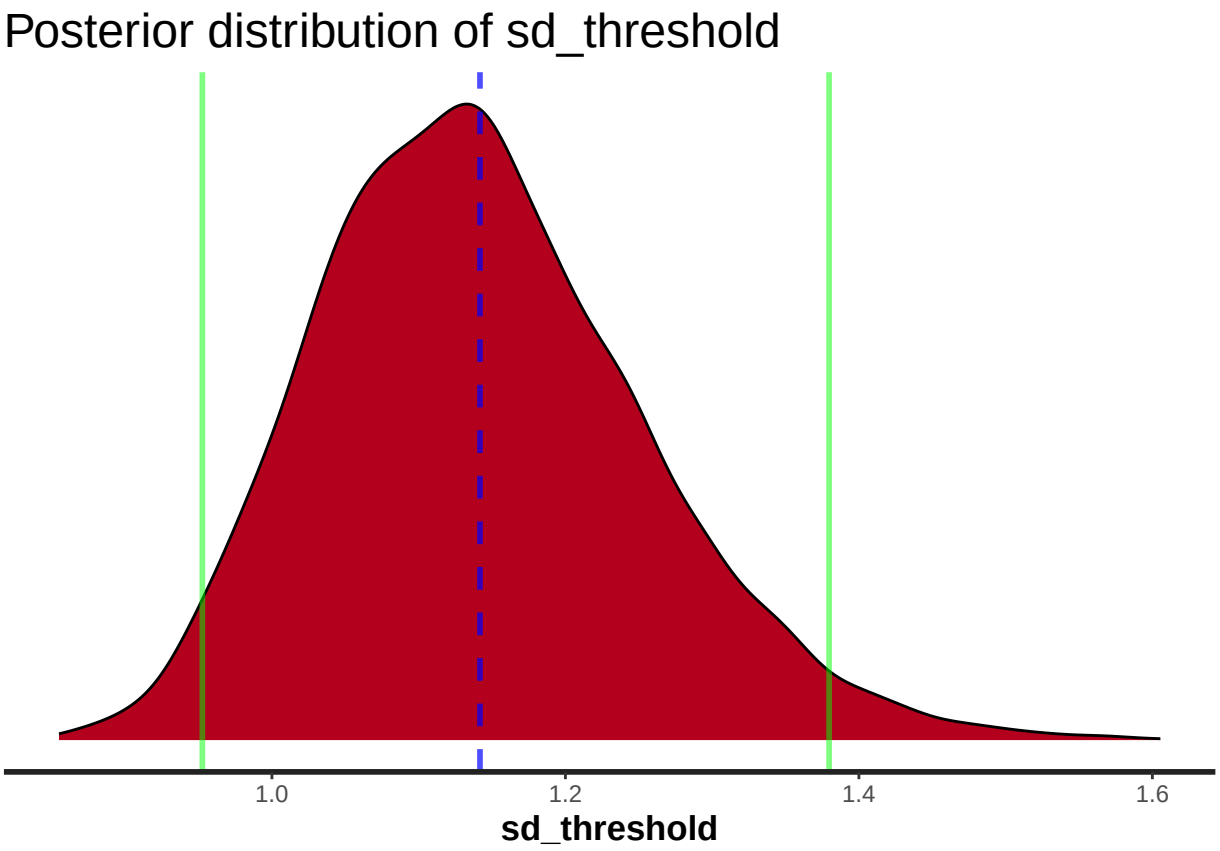
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

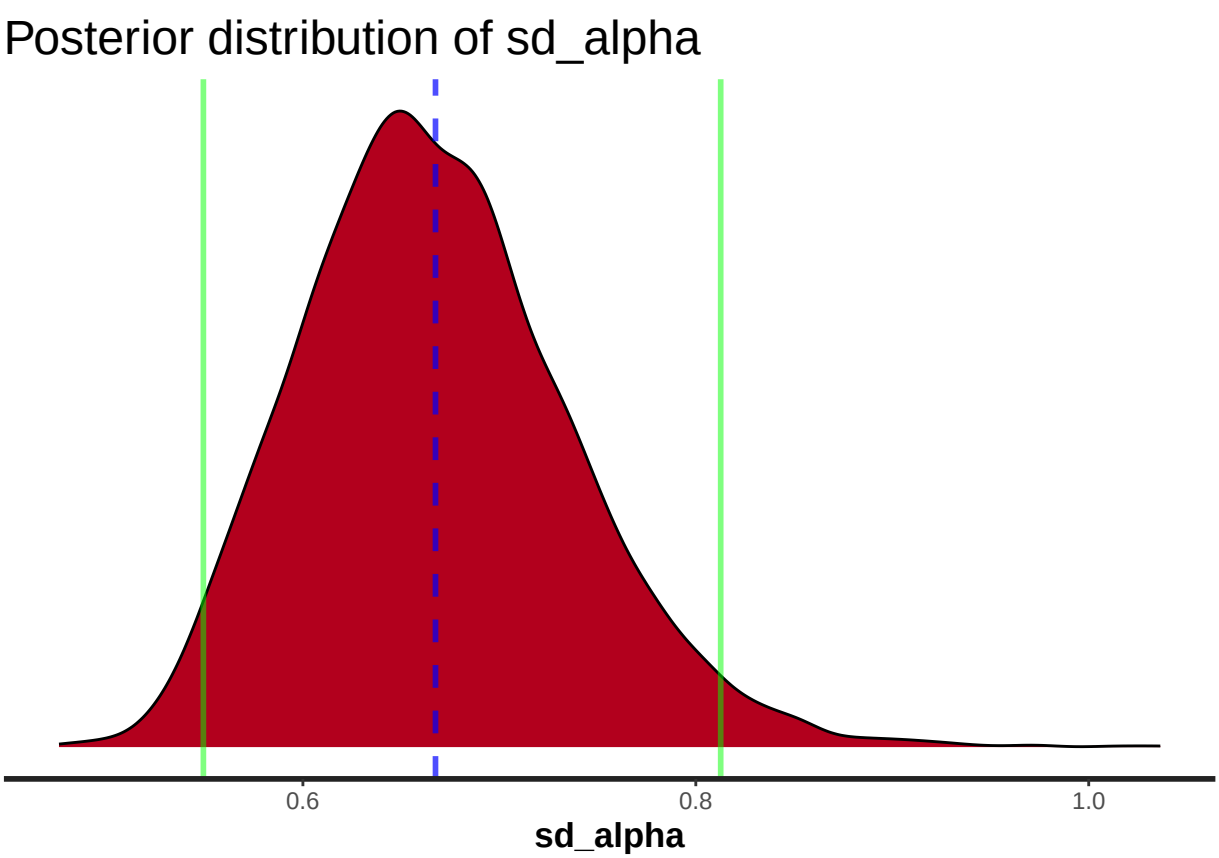

Posterior distribution of transf_mu_alpha
−0.8
−0.6
−0.4
transf_mu_alpha

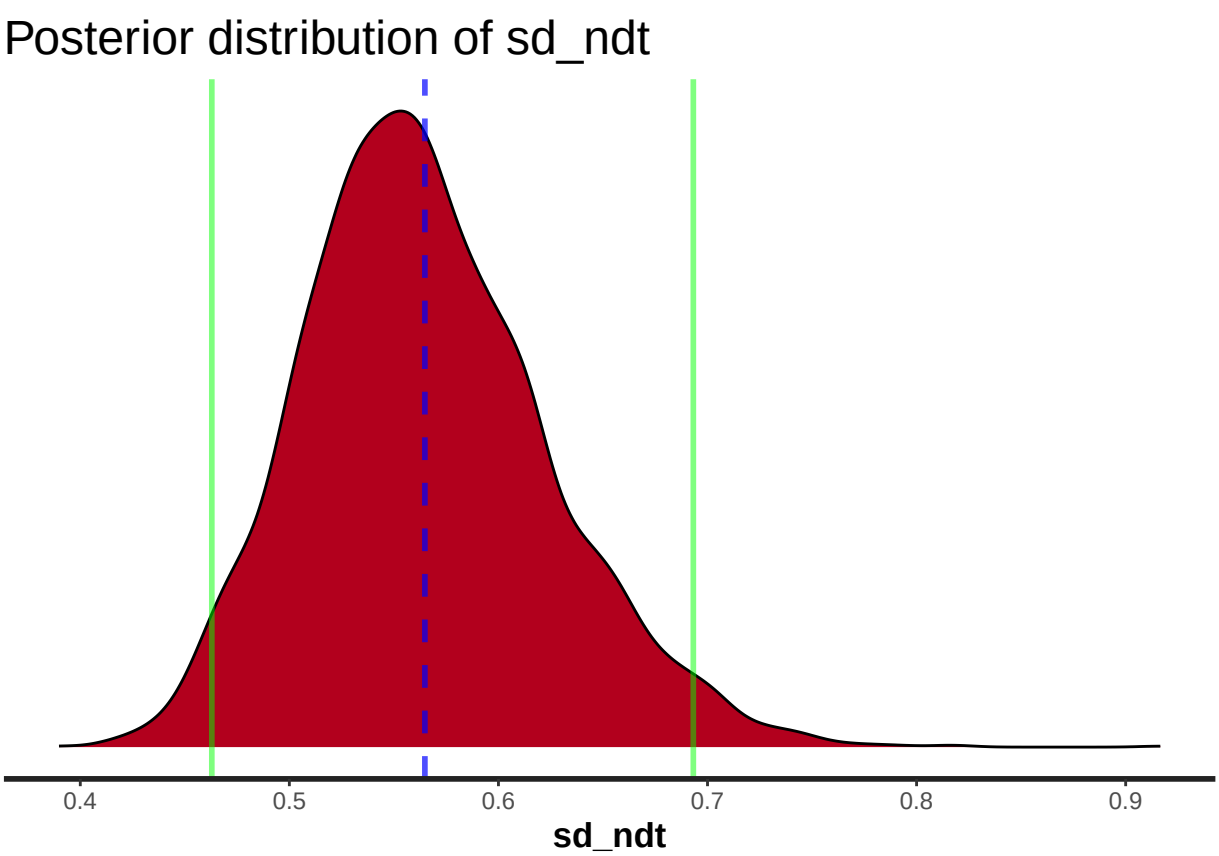

Posterior distribution of transf_mu_theta
0.012
0.013
0.014
0.015
transf_mu_theta

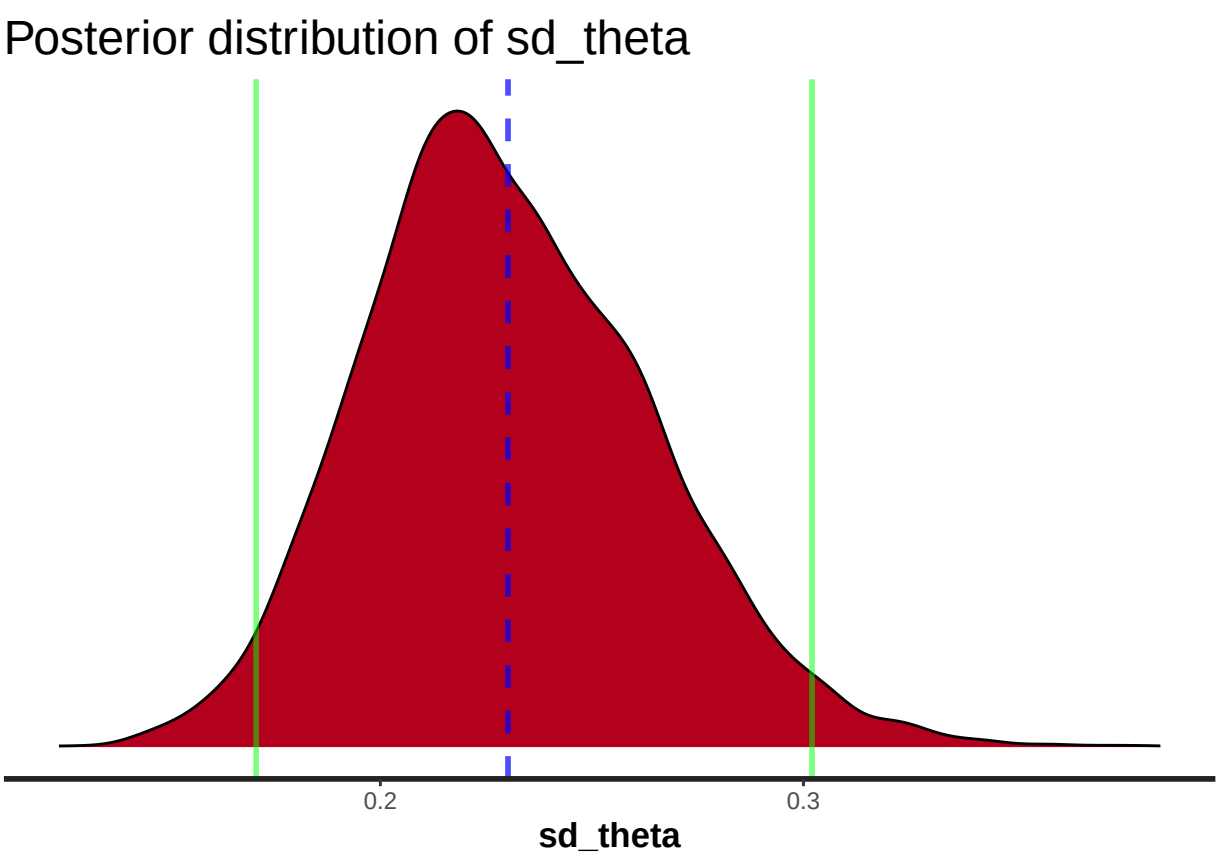

Posterior distribution of transf_mu_threshold
4.0
4.4
4.8
5.2
transf_mu_threshold


Posterior distribution of transf_mu_ndt
0.9
1.0
1.1
1.2
1.3
transf_mu_ndt


Posterior distribution of sd_threshold
1.0
1.2
1.4
1.6
sd_threshold


Posterior distribution of sd_alpha
0.6
0.8
1.0
sd_alpha


Posterior distribution of sd_ndt
0.4
0.5
0.6
0.7
0.8
0.9
sd_ndt


Posterior distribution of sd_theta
0.2
0.3
sd_theta