```
############### 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$oa3-dat$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)
```

```
# transform to +/- 1; safe - 1, risky +1
dat$cho <- ifelse(dat$choice==0,-1,ifelse(dat$choice==1,1,NA))
dat$cho2 <- ifelse(dat$choice==0,1,ifelse(dat$choice==1,0,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$cho, accuracy_flipped = rcontrol_dat$cho2, rt = rcontrol_dat$rt, pa
```

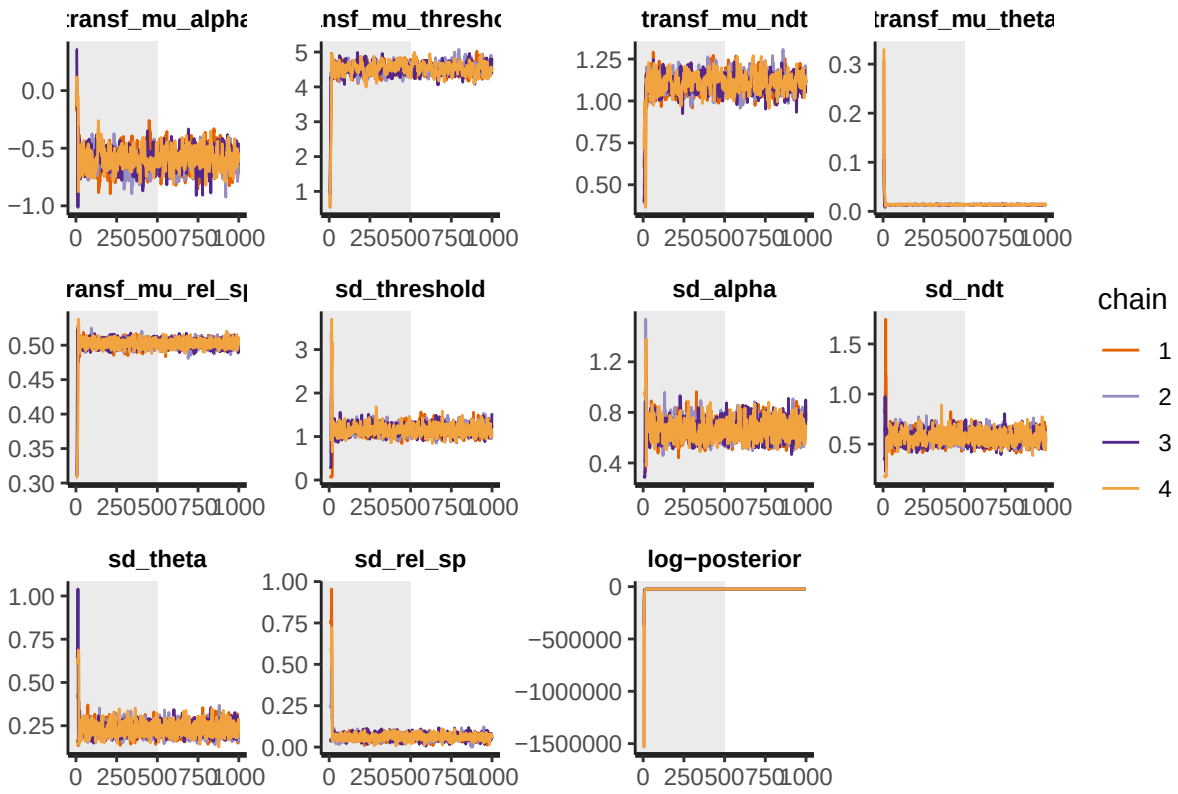
```
parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu_theta",'transf_mu_rel

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, 0.5),
                          sd_threshold = runif(1,0,1),
                          mu_ndt = runif(1, -1.5, 0),
                          sd_ndt = runif(1, 0, 1),
                          mu_theta = runif(1,-0.5, -0.5),
                          sd_theta = runif(1,0,1),
                          mu_rel_sp = runif(1,-0.5, -0.5),
                          sd_rel_sp = 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),
                          z_rel_sp = runif(length(tids),-0.1,0.1)
    )
  }

  return(initList)
}
```

```
m <- stan_model("MV_SP.stan")
dsamples <- sampling(m,
                data=dataList,
                pars=parameters,
                iter=1000,
                chains=4,#If not specified, gives random inits
                init = initFunc(4),
                warmup = 500,  # Stands for burn-in; Default = iter/2
                seed = 12,  # Setting seed; Default is random seed
                refresh = 0
                )
```

```
## 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
```

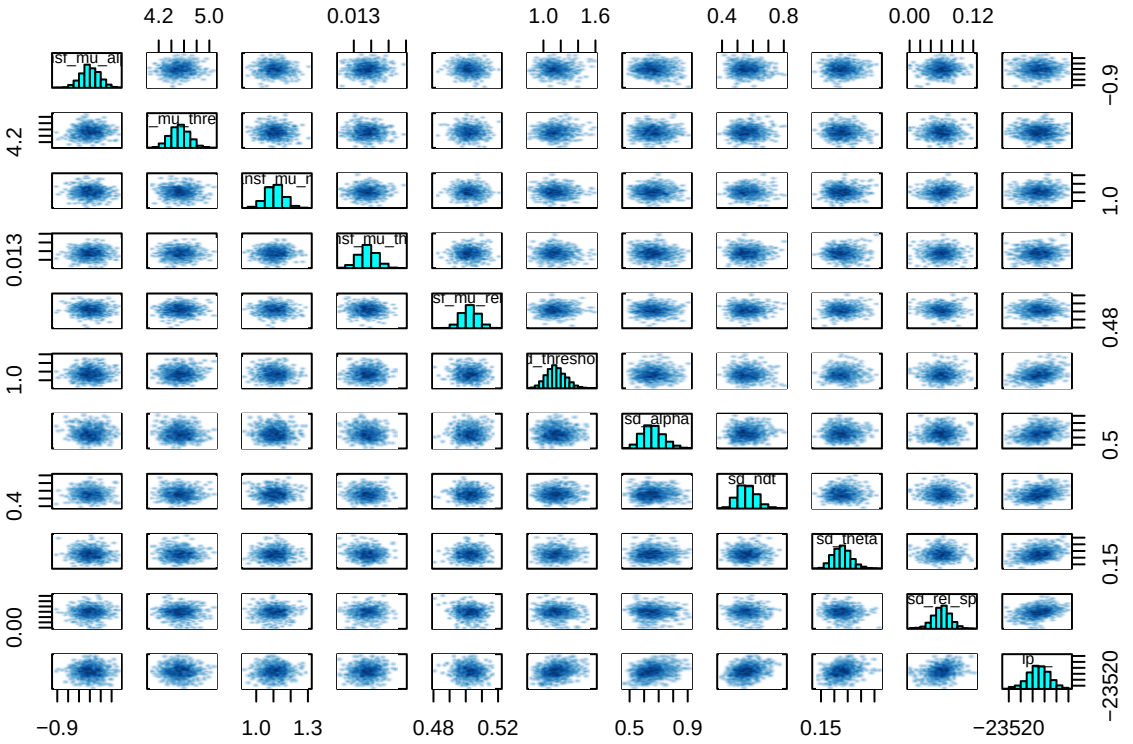
```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

```
#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_threshold","transf_mu_ndt", "transf_mu_th
```




```
pairs(dsamples, pars = c( "transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu_theta",'t
```

```
## 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

## 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",'tra
```

```
## Inference for Stan model: MV_SP.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##                          mean se_mean    sd      2.5%       25%       50%
## transf_mu_alpha         -0.60    0.00  0.09     -0.76     -0.65     -0.60
## transf_mu_threshold      4.53    0.01  0.15      4.25      4.43      4.53
## transf_mu_ndt            1.11    0.00  0.05      1.01      1.07      1.11
## transf_mu_theta          0.01    0.00  0.00      0.01      0.01      0.01
## transf_mu_rel_sp         0.50    0.00  0.01      0.49      0.50      0.50
## sd_threshold             1.15    0.01  0.11      0.95      1.07      1.14
## sd_alpha                 0.66    0.00  0.07      0.53      0.61      0.66
## sd_ndt                   0.56    0.00  0.06      0.45      0.52      0.56
## sd_theta                 0.23    0.00  0.03      0.17      0.20      0.23
## sd_rel_sp                0.06    0.00  0.02      0.02      0.05      0.06
## lp__                -23470.09    0.95 17.86 -23507.94 -23481.54 -23469.89
##                          75%     97.5% n_eff Rhat
## transf_mu_alpha         -0.54     -0.42   368 1.01
## transf_mu_threshold      4.64      4.83   229 1.02
## transf_mu_ndt            1.14      1.21   258 1.01
## transf_mu_theta          0.01      0.01  1400 1.00
## transf_mu_rel_sp         0.51      0.51  2701 1.00
## sd_threshold             1.22      1.39   497 1.01
```

```
## sd_alpha                 0.71      0.82   709 1.00
## sd_ndt                   0.60      0.69   436 1.01
## sd_theta                 0.25      0.30  1122 1.00
## sd_rel_sp                0.07      0.09   487 1.01
## lp__                -23458.14 -23435.44   355 1.01
##
## Samples were drawn using NUTS(diag_e) at Wed Oct  4 13:20:54 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).
```

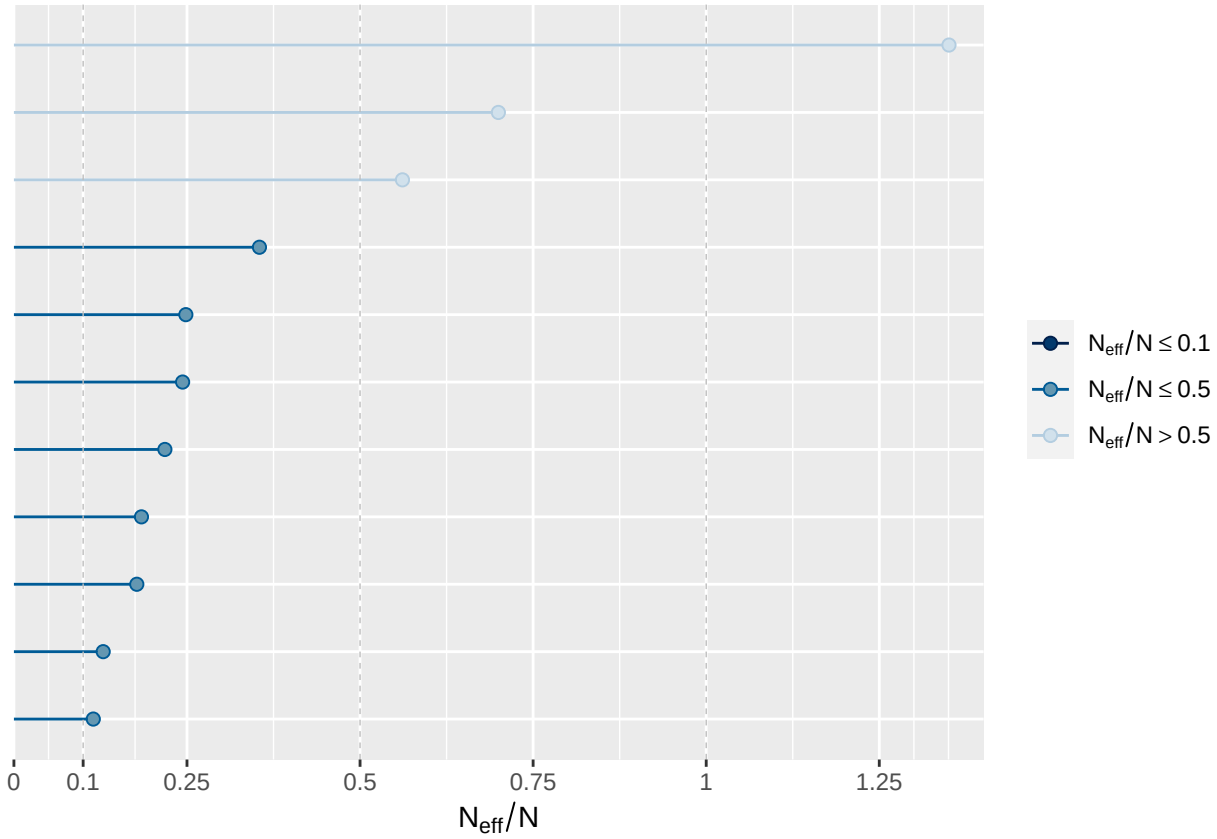
```
library(bayesplot)
```

```
## 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", "transf_mu_threshold","
df_ratios_cp <- as.data.frame(ratios_cp)
print(df_ratios_cp)
```

```
##                     ratios_cp
## transf_mu_alpha     0.1842285
## transf_mu_theta     0.6998025
## transf_mu_threshold 0.1146398
## transf_mu_ndt       0.1289420
## transf_mu_rel_sp    1.3507075
## sd_threshold        0.2483458
## sd_alpha            0.3546558
## sd_ndt              0.2180760
## sd_theta            0.5612301
## sd_rel_sp           0.2436870
## lp__                0.1775030
```

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


0
0.1
0.25
0.5
0.75
1
1.25
N_{eff}/N
$N_{eff}/N \leq 0.1$
$N_{eff}/N \leq 0.5$
$N_{eff}/N > 0.5$