```
############### 0 - safe choice A, 1 - risky choice B #####################
library(rstan); rstan_options(javascript=FALSE)
library(bayesplot)
library(dplyr)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = T)

dat <- read.csv('final_data.csv')
```

```
dat <- dat %>%
  filter(skew != 'control')
```

```
ids <- unique(dat$subject)
for(j in 1:length(ids)){
  dat$tid[dat$subject==ids[j]] <- j
}
tids <- unique(dat$tid)
dat$rt <- as.numeric(dat$rt/1000)

dat <- dat %>%
  filter(test_part == 'cc' | test_part == 'ss',
         rt >= 1,
         subject != '4ld6kjtr',
         subject != 'm73bj2hn')

dat <- dat %>%
  mutate(con = ifelse(test_part == "cc", 1, -1))
```

```
# only condition no time pressure
dataList  = list(cho = dat$cho, rt = dat$rt, participant = dat$tid, N=nrow(dat),
L = length(tids), starting_point=0.5, evd = dat$evd, sdd = dat$sdd, con =  dat$co
n)
```

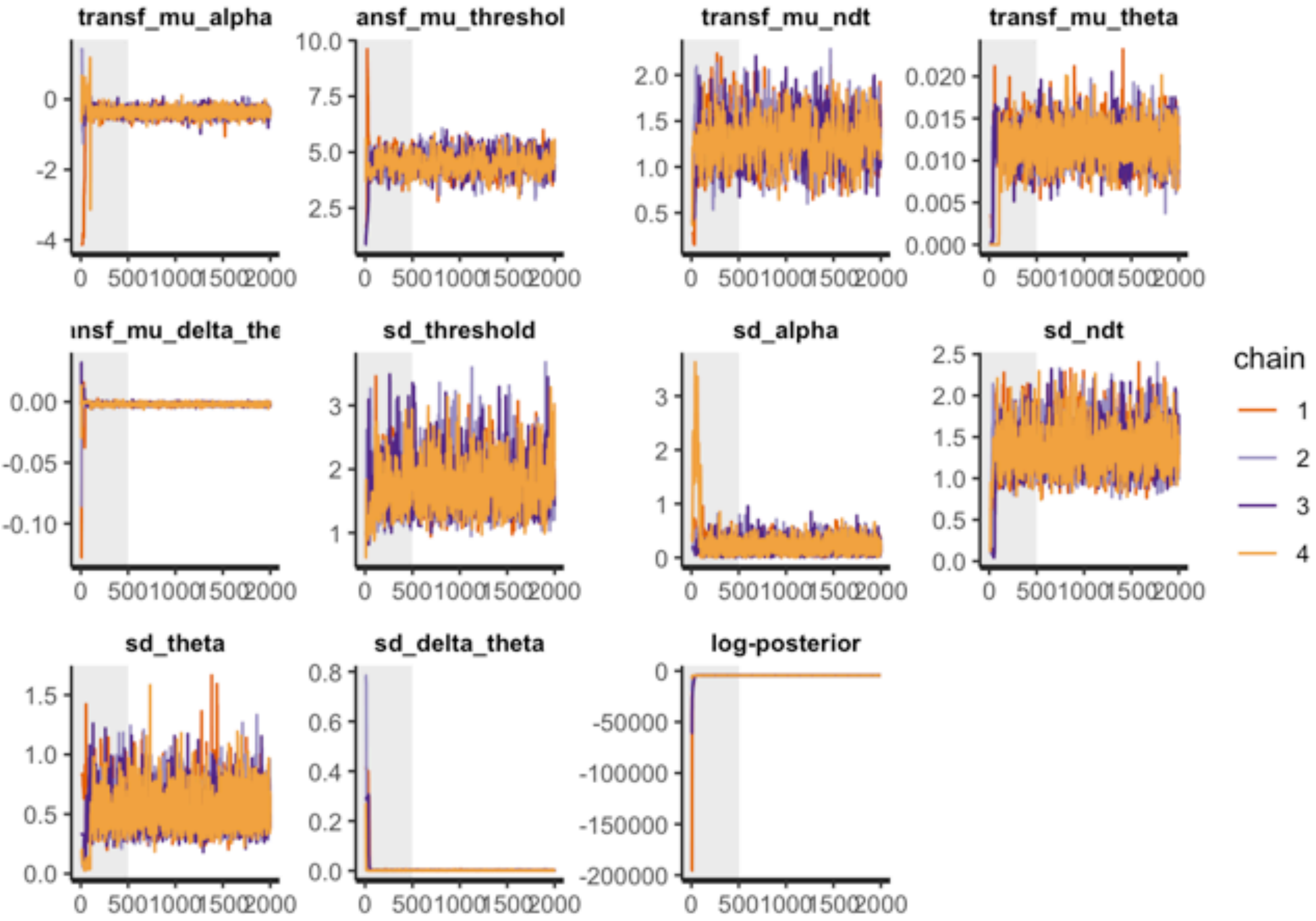
```
parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu
_theta","transf_mu_delta_theta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_theta', '
sd_delta_theta', "alpha_sbj","threshold_sbj","ndt_sbj",'theta_sbj','delta_theta_sb
j', "log_lik")

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, 5),
                          sd_threshold = runif(1, 0, 1),
                          mu_ndt = runif(1, -1.5, 0),
                          sd_ndt = runif(1, 0, 1),
                          mu_theta = runif(1,-20, 1),
                          sd_theta = runif(1,0,1),
                          mu_delta_theta = runif(1, -1, 1),
                          sd_delta_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),
                          z_delta_theta = runif(length(tids),-0.1,0.1)

    )
  }

  return(initList)
}
```

```
m <- stan_model("MV_Baseline.stan")
```

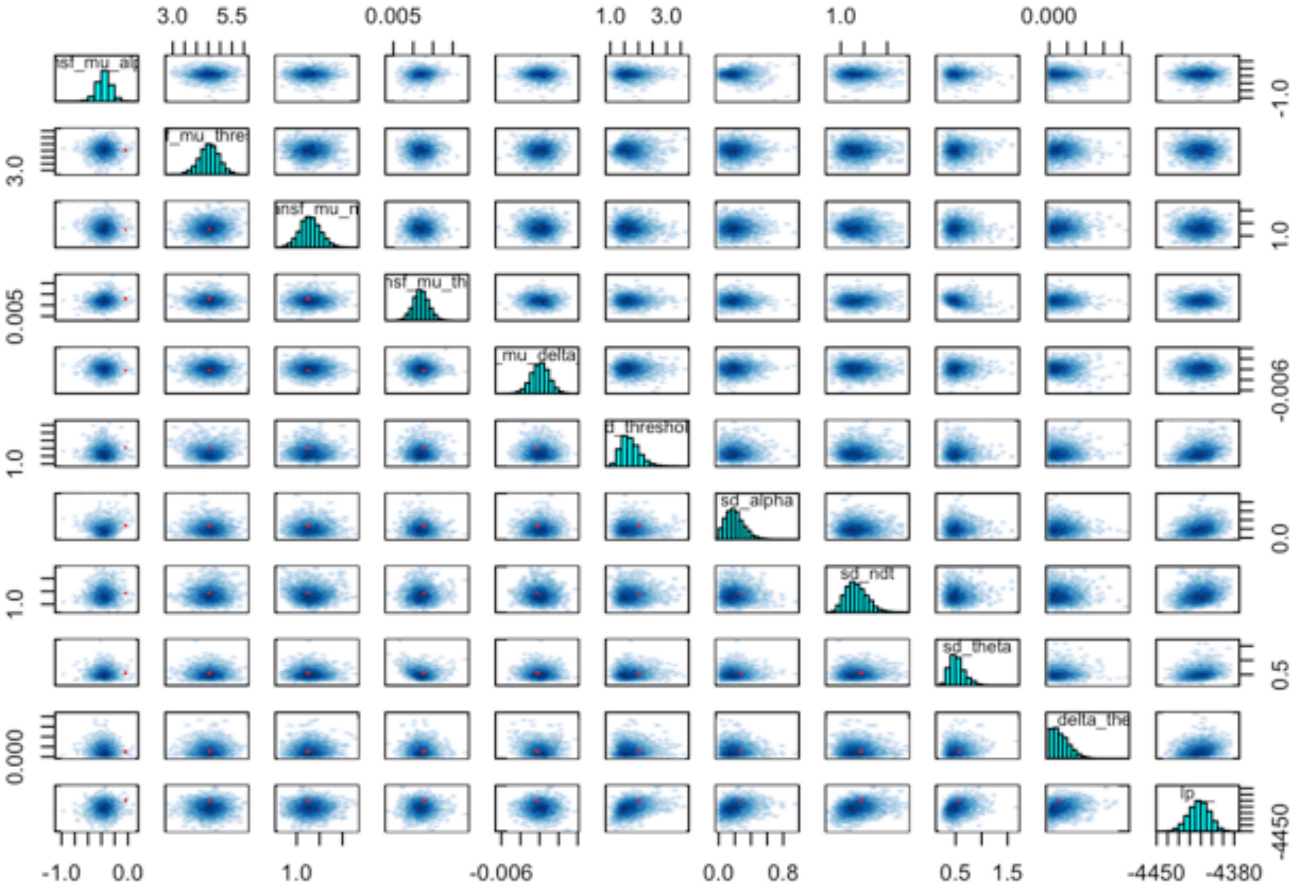
```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/i
nclude" -DNDEBUG   -I"/Library/Frameworks/R.framework/Versions/4.2/Resources/libra
ry/Rcpp/include/"  -I"/Library/Frameworks/R.framework/Versions/4.2/Resources/libra
ry/RcppEigen/include/"  -I"/Library/Frameworks/R.framework/Versions/4.2/Resources/
library/RcppEigen/include/unsupported"  -I"/Library/Frameworks/R.framework/Version
s/4.2/Resources/library/BH/include" -I"/Library/Frameworks/R.framework/Versions/4.
2/Resources/library/StanHeaders/include/src/"  -I"/Library/Frameworks/R.framework/
Versions/4.2/Resources/library/StanHeaders/include/"  -I"/Library/Frameworks/R.fra
mework/Versions/4.2/Resources/library/RcppParallel/include/"  -I"/Library/Framewor
ks/R.framework/Versions/4.2/Resources/library/rstan/include" -DEIGEN_NO_DEBUG  -DB
OOST_DISABLE_ASSERTS  -DBOOST_PENDING_INTEGER_LOG2_HPP  -DSTAN_THREADS  -DBOOST_NO
_AUTO_PTR  -include '/Library/Frameworks/R.framework/Versions/4.2/Resources/librar
y/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp'  -D_REENTRANT -DRCPP_PARAL
LEL_USE_TBB=1   -I/usr/local/include   -fPIC  -Wall -g -O2  -c foo.c -o foo.o
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/li
brary/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:
## In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/li
brary/RcppEigen/include/Eigen/Dense:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/li
brary/RcppEigen/include/Eigen/Core:88:
## /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/includ
e/Eigen/src/Core/util/Macros.h:628:1: error: unknown type name 'namespace'
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/includ
e/Eigen/src/Core/util/Macros.h:628:16: error: expected ';' after top level declara
tor
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/li
brary/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:
## In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/li
brary/RcppEigen/include/Eigen/Dense:1:
## /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/includ
e/Eigen/Core:96:10: fatal error: 'complex' file not found
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
```

```
dsamples <- sampling(m,
                data=dataList,
                pars=parameters,
                iter=2000,
                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
                )
```

```
#parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_m
u_theta","transf_mu_delta_theta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_theta',
'sd_delta_theta', "alpha_sbj","threshold_sbj","ndt_sbj",'theta_sbj','delta_theta_s
bj', "log_lik")
rstan::traceplot(dsamples, pars=c("transf_mu_alpha","transf_mu_threshold","transf_
mu_ndt", "transf_mu_theta","transf_mu_delta_theta", 'sd_threshold',"sd_alpha","sd_
ndt", 'sd_theta', 'sd_delta_theta', "lp__"), inc_warmup = TRUE, nrow = 3)
```



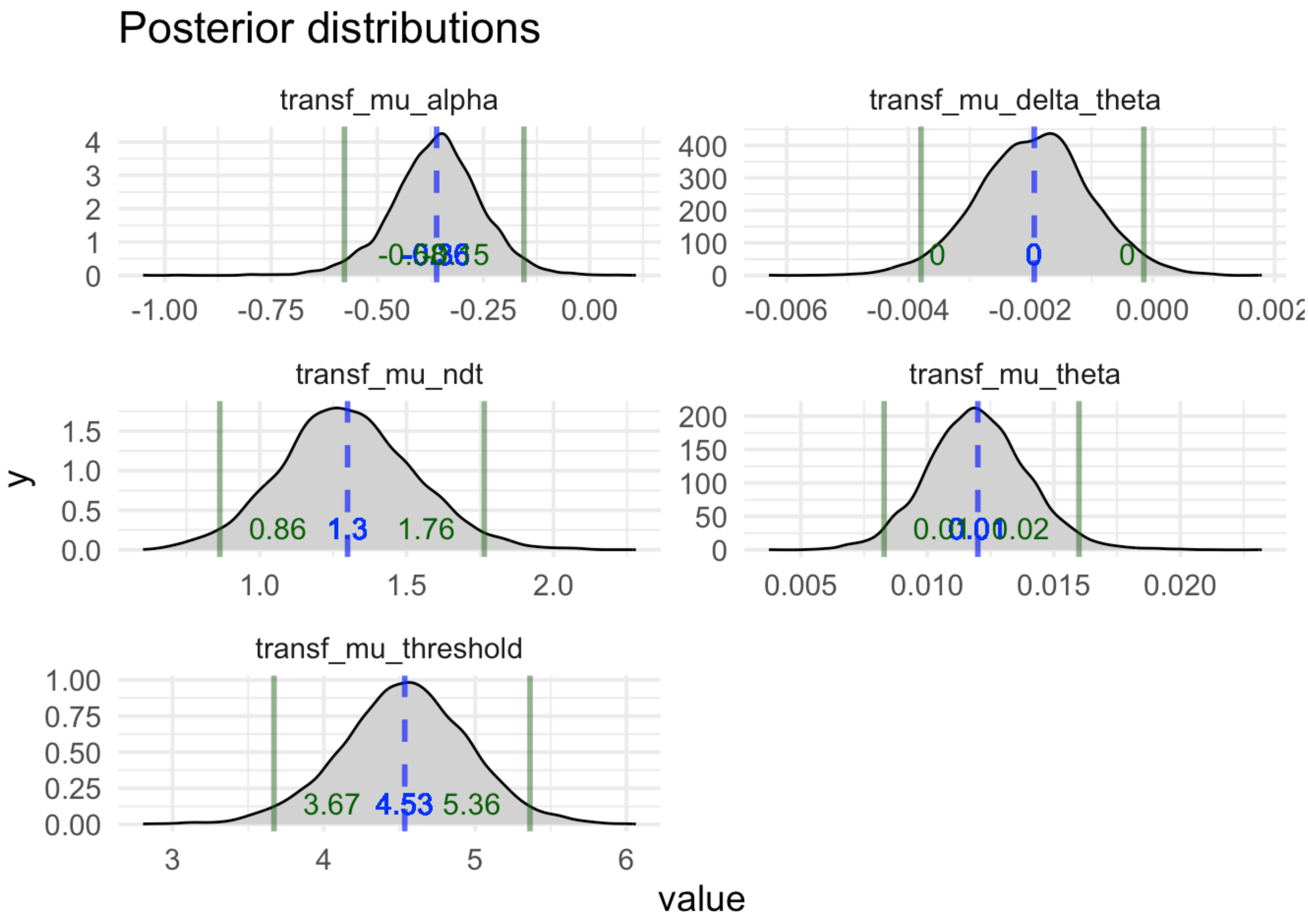
```
pairs(dsamples, pars = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt",
"transf_mu_theta","transf_mu_delta_theta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd
_theta', 'sd_delta_theta', "lp__"))
```



```
print(dsamples, pars = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt",
"transf_mu_theta","transf_mu_delta_theta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd
_theta', 'sd_delta_theta', "lp__"))
```

```
## Inference for Stan model: MV_Baseline.
## 4 chains, each with iter=2000; warmup=500; thin=1; 
## post-warmup draws per chain=1500, total post-warmup draws=6000.
## 
##                            mean se_mean   sd     2.5%      25%      50%      75%
## transf_mu_alpha           -0.36    0.00 0.11    -0.58    -0.42    -0.36    -0.29
## transf_mu_threshold        4.53    0.02 0.43     3.67     4.26     4.54     4.82
## transf_mu_ndt              1.30    0.01 0.23     0.86     1.15     1.29     1.44
## transf_mu_theta            0.01    0.00 0.00     0.01     0.01     0.01     0.01
## transf_mu_delta_theta      0.00    0.00 0.00     0.00     0.00     0.00     0.00
## sd_threshold               1.72    0.01 0.35     1.19     1.47     1.67     1.91
## sd_alpha                   0.20    0.00 0.12     0.02     0.12     0.19     0.27
## sd_ndt                     1.34    0.01 0.24     0.95     1.17     1.32     1.49
## sd_theta                   0.53    0.00 0.15     0.31     0.42     0.51     0.62
## sd_delta_theta             0.00    0.00 0.00     0.00     0.00     0.00     0.00
## lp__                   -4411.64    0.25 9.21 -4430.07 -4417.67 -4411.43 -4405.25
##                           97.5% n_eff Rhat
## transf_mu_alpha           -0.15  3149 1.00
## transf_mu_threshold        5.36   510 1.01
## transf_mu_ndt              1.76   733 1.00
## transf_mu_theta            0.02  1879 1.00
## transf_mu_delta_theta      0.00  5750 1.00
## sd_threshold               2.53  1145 1.00
## sd_alpha                   0.47  2572 1.00
## sd_ndt                     1.88  1380 1.00
## sd_theta                   0.89  2205 1.00
## sd_delta_theta             0.00  3309 1.00
## lp__                   -4394.18  1401 1.00
## 
## Samples were drawn using NUTS(diag_e) at Wed Nov 29 13:46:22 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(ggplot2)
library(tidyverse) # for the gather function

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", "transf_mu_delta_theta")

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

# 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")

print(p)
```


Posterior distributions
transf_mu_alpha
4
3
2
1
0
-0.58
-0.36
-0.15
-1.00
-0.75
-0.50
-0.25
0.00
transf_mu_delta_theta
400
300
200
100
0
0
0
0
-0.006
-0.004
-0.002
0.000
0.002
transf_mu_ndt
1.5
1.0
0.5
0.0
0.86
1.3
1.76
1.0
1.5
2.0
transf_mu_theta
200
150
100
50
0
0.01
0.01
0.02
0.005
0.010
0.015
0.020
transf_mu_threshold
1.00
0.75
0.50
0.25
0.00
3.67
4.53
5.36
3
4
5
6
y
value