```
############### 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$oa
3-dat$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)
```

```
# 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,]

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
rcontrol_dat <- rcontrol_dat %>%
  rowwise() %>%
  mutate(
    oa_condition = sum(c_across(starts_with("oa")) == 0),
    ob_condition = sum(c_across(starts_with("ob")) == 0)
  ) %>%
  filter(
    (oa_condition == 2 & ob_condition == 2) |
    (oa_condition == 0 & ob_condition == 0)
  )
```

```
rcontrol_dat
```

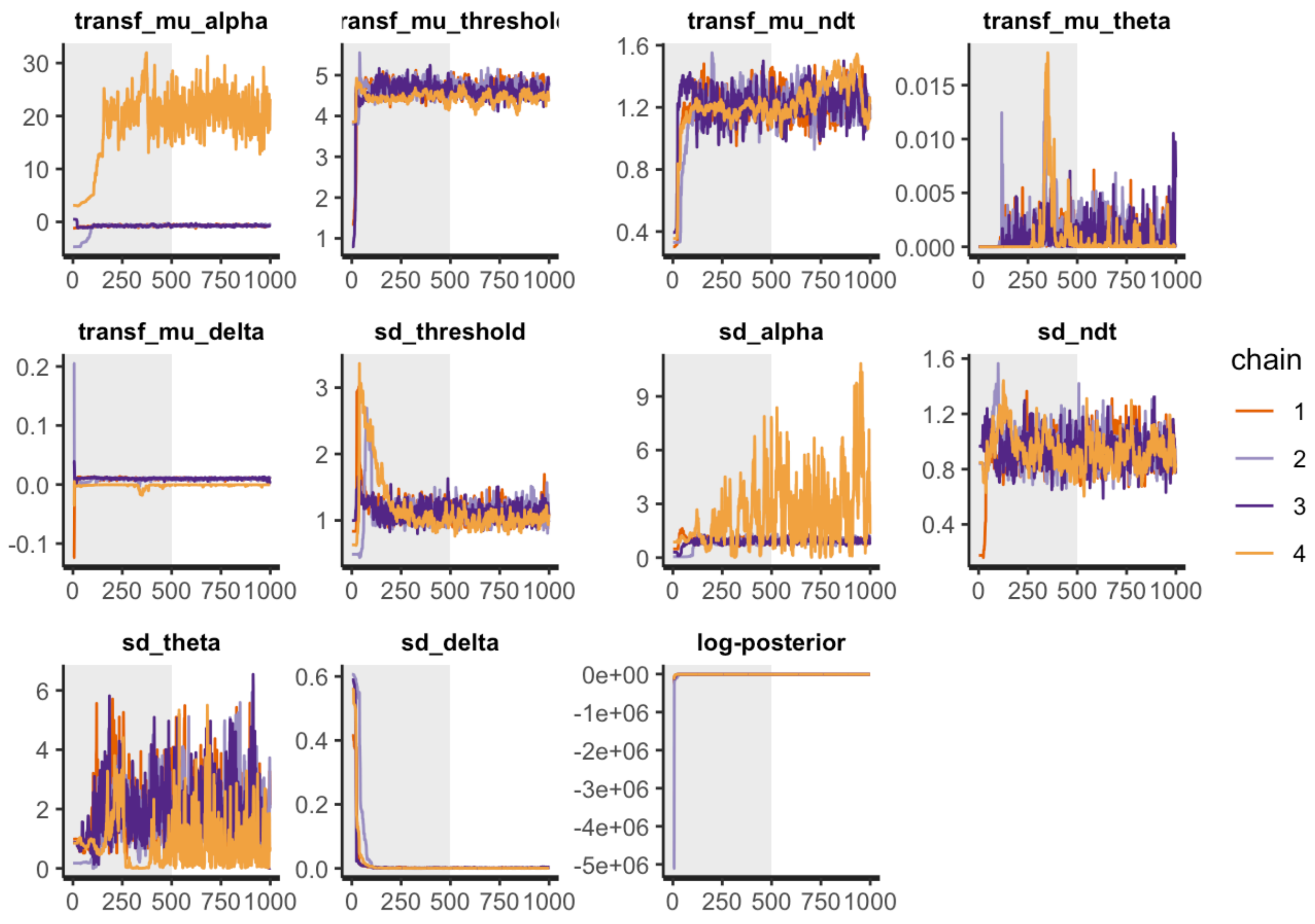
```
## # A tibble: 2,995 × 42
## # Rowwise:
##    date      id block order cond  trial gameid leftright   oa1   oa2   oa3   oa
4
##    <chr>  <int> <int> <int> <chr> <int>  <int>     <int> <int> <int> <int> <int
>
##  1 2019_…     1     1     0 cont…     1    144         1    99    93    90     7
1
##  2 2019_…     1     1     0 cont…     3    121         1    42    51    47     7
0
##  3 2019_…     1     1     0 cont…     5    118         1    62    52    65     6
8
##  4 2019_…     1     1     0 cont…     8    126         0    16    55    51     4
2
##  5 2019_…     1     1     0 cont…     9    142         1    98    65    64     8
4
##  6 2019_…     1     1     0 cont…    10    129         0    12    29    81     3
6
##  7 2019_…     1     1     0 cont…    27    123         1    26    24    51     9
2
##  8 2019_…     1     1     0 cont…    29    122         1    35    58    25     3
1
##  9 2019_…     1     1     0 cont…    32    117         1    65    85    92     4
1
## 10 2019_…     1     1     0 cont…    37    139         0    44    31    16
8
## # ℹ 2,985 more rows
## # ℹ 30 more variables: pa1 <dbl>, pa2 <dbl>, pa3 <dbl>, pa4 <dbl>, ob1 <int>,
## #   ob2 <int>, ob3 <int>, ob4 <int>, pb1 <dbl>, pb2 <dbl>, pb3 <dbl>,
## #   pb4 <dbl>, choice <int>, rt <dbl>, paytrial <int>, payout <int>,
## #   alter <int>, geschlecht <int>, comment <chr>, eva <dbl>, evb <dbl>,
## #   evd <dbl>, sda <dbl>, sdb <dbl>, sdd <dbl>, evdummy <dbl>, cho <dbl>,
## #   tid <int>, oa_condition <int>, ob_condition <int>
```

```
# only condition no time pressure
dataList  = list(cho = rcontrol_dat$cho, rt = rcontrol_dat$rt, participant = rcont
rol_dat$tid, N=nrow(rcontrol_dat),  L = length(tids), starting_point=0.5, evd = rc
ontrol_dat$evd, sdd = rcontrol_dat$sdd, con =  rep(1, length(rcontrol_dat$trial)))
```

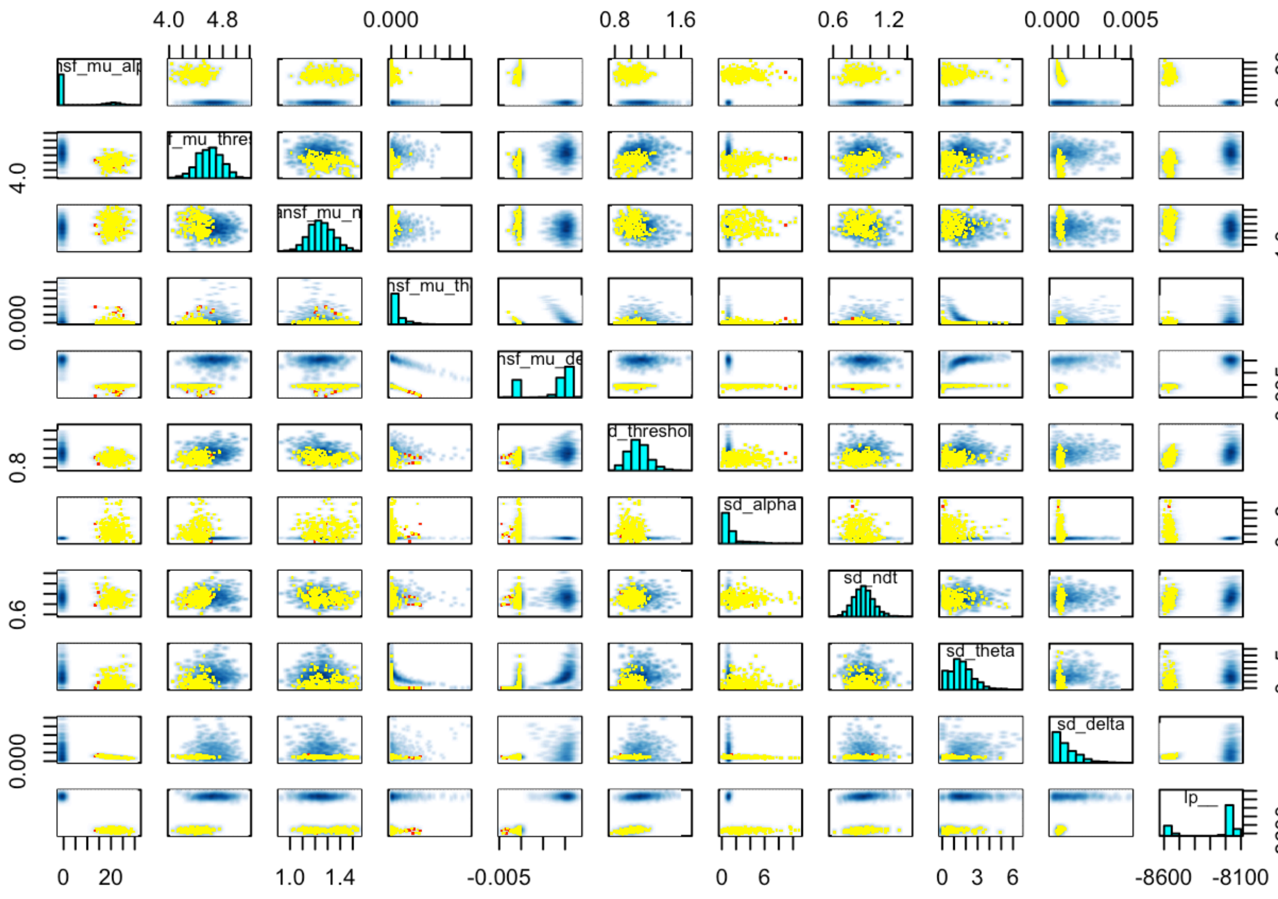
```
parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_mu
_theta","transf_mu_delta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_theta', 'sd_del
ta', "alpha_sbj","threshold_sbj","ndt_sbj",'theta_sbj','delta_sbj', "log_lik")

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 = runif(1, -1, 1),
                          sd_delta = 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 = runif(length(tids),-0.1,0.1)

    )
  }

  return(initList)
}
```

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

```
#parameters = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt", "transf_m
u_theta","transf_mu_delta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_theta', 'sd_de
lta', "alpha_sbj","threshold_sbj","ndt_sbj",'theta_sbj','delta_sbj', "log_lik")
rstan::traceplot(dsamples, pars=c("transf_mu_alpha","transf_mu_threshold","transf_
mu_ndt", "transf_mu_theta","transf_mu_delta", 'sd_threshold',"sd_alpha","sd_ndt",
'sd_theta', 'sd_delta', "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", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_thet
a', 'sd_delta', "lp__"))
```



```
print(dsamples, pars = c("transf_mu_alpha","transf_mu_threshold","transf_mu_ndt",
"transf_mu_theta","transf_mu_delta", 'sd_threshold',"sd_alpha","sd_ndt", 'sd_thet
a', 'sd_delta', "lp__"))
```

```
## Inference for Stan model: MV_Baseline.
## 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%      75%
## transf_mu_alpha          4.71    6.62   9.47    -0.91    -0.74    -0.64     2.99
## transf_mu_threshold      4.60    0.08   0.18     4.22     4.48     4.61     4.73
## transf_mu_ndt            1.25    0.01   0.10     1.07     1.18     1.24     1.31
## transf_mu_theta          0.00    0.00   0.00     0.00     0.00     0.00     0.00
## transf_mu_delta          0.01    0.00   0.00     0.00     0.00     0.01     0.01
## sd_threshold             1.09    0.03   0.12     0.88     1.00     1.08     1.16
## sd_alpha                 1.44    0.58   1.36     0.63     0.86     0.95     1.11
## sd_ndt                   0.93    0.01   0.11     0.73     0.86     0.92     1.00
## sd_theta                 1.66    0.31   1.05     0.04     0.95     1.56     2.23
## sd_delta                 0.00    0.00   0.00     0.00     0.00     0.00     0.00
## lp__                 -8265.09  121.82 173.05 -8586.26 -8296.69 -8172.84 -8157.84
##                         97.5% n_eff  Rhat
## transf_mu_alpha         24.68     2  6.67
## transf_mu_threshold      4.95     5  1.31
## transf_mu_ndt            1.45    45  1.11
## transf_mu_theta          0.00    40  1.07
## transf_mu_delta          0.01     2  4.69
## sd_threshold             1.36    16  1.11
## sd_alpha                 5.59     5  1.36
## sd_ndt                   1.16   319  1.02
## sd_theta                 4.08    11  1.15
## sd_delta                 0.00    21  1.12
## lp__                 -8136.54     2 11.04
##
## Samples were drawn using NUTS(diag_e) at Thu Nov 16 04:09:39 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(ggplot2)
library(tidyverse) # for the gather function
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ─
─
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.1.8
## ✔ purrr     1.0.1     ✔ tidyr     1.3.0
## ✔ readr     2.1.4
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ─
─
## ✖ tidyr::extract() masks rstan::extract()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force a
ll conflicts to become errors
```

```
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)
```

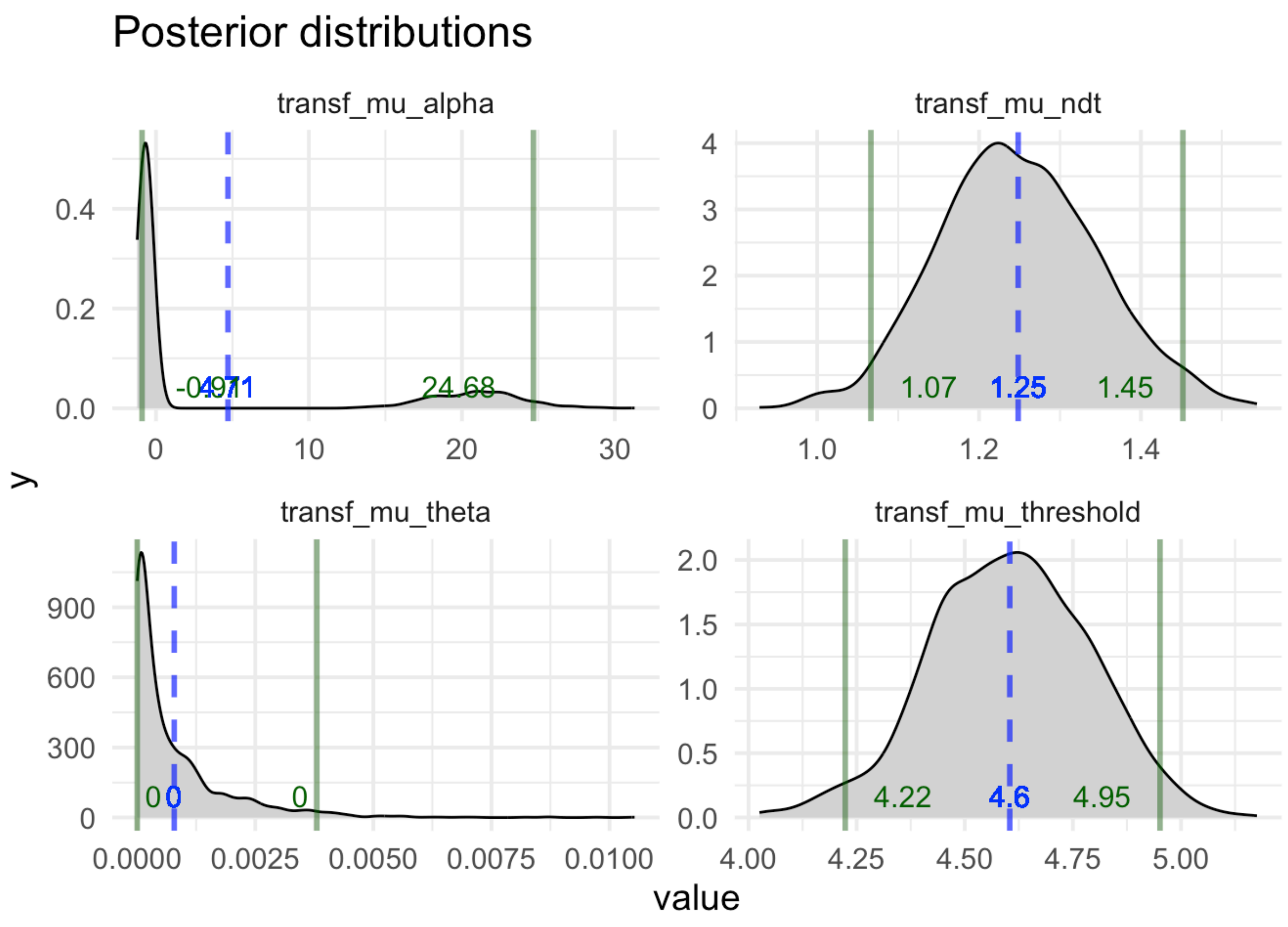
```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.
1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(parameters)
##
##   # Now:
##   data %>% select(all_of(parameters))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## 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