

Demo of Bayesian Causal Forests on a Randomized Controlled Trial

Jared Murray, Drew Herren

2025-01-06

Introduction

In this vignette we re-analyze data from study 6 in “A synergistic mindsets intervention protects adolescents from stress (@yeager2022synergistic)”. The study was a randomized controlled trial of a light-touch synergistic mindsets intervention. Variables include:

- **testanxiety**: A measure of the student’s test anxiety
- **sex**: 1 if M, 0 if F
- **stressmindset_baseline**: Pre-treatment stress mindset; lower values indicate positive stress-can-be-enhancing mindsets
- **fixedmindset_baseline**: Pre-treatment growth/fixed mindset; lower values indicate positive growth mindsets
- **pss_impute**: A measure of perceived social stress at baseline
- **fgen_moth**: 1 if the student is a first-generation college student on their mother’s side
- **fgen_fath**: 1 if the student is a first-generation college student on their father’s side
- **ses**: A measure of socioeconomic status

The outcome **anxiety** is a measure of anxiety (GAD-7).

First, we load requisite packages.

```
library(stochtree)
library(ggplot2)
library(coda)
library(possum)
library(rpart)
library(rpart.plot)
library(tidyverse)
library(here)
```

Dataset

Next we import and process the data:

```
# Import dataset
file_path <- file.path(here(), "data/rct/synergistic_study6.csv")
df <- read.csv(file_path)
```

And unpack the data into BCF components: outcome, treatment, and covariates.

We also create two new variables from the baseline fixed and stress mindset scores (where higher scores correspond to more fixed mindsets and less stress-can-be-enhancing mindsets) by adding and multiplying them. Since the hypothesis was that the treatment would be most effective for students with poor growth and stress mindsets, adding these coordinates will allow the trees to make one split to isolate groups that are high or low on both mindset scores. Feature engineering like this can be very helpful when using tree-based methods in general, especially when the sample size is small.

```

# Outcome
y <- df$anxiety

# Treatment
Z <- df$treatment

# Covariates
covariate_df <- dplyr::select(df, -treatment, -anxiety) %>%
  mutate(bothplus = stressmindset_baseline + fixedmindset_baseline,
         bothprod = scale(stressmindset_baseline)*scale(fixedmindset_baseline)) %>%
  data.frame()

```

Model Run

Fit the BCF model

$$y_i = \mu(x_i) + \tau(x_i)z_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

with 100 “warm-start” iterations and 3000 MCMC iterations after 100 burn-in iterations. Since this is a randomized controlled trial we omit a fitted propensity score.

We also tweak a couple of the BCF defaults:

- We turn off adaptive coding of the treatment dummies, and
- we increase the probability that a heterogeneity tree splits at least once (`alpha_tau` from 0.25 to 0.95).

The BCF defaults were developed for scenarios with moderate to strong confounding and weak prior evidence of heterogeneous effects. Here we have a randomized trial, and lots of prior evidence and theory that suggests effects should be heterogeneous.

```

random_seed <- 121
set.seed(random_seed)
t0 <- Sys.time()
num_gfr <- 100
num_burnin <- 100
num_mcmc <- 3000
bcf_params <- list(
  propensity_covariate = 'mu',
  cutpoint_grid_size = 100,
  sample_sigma_leaf_tau = T,
  sample_sigma_leaf_mu = T,
  adaptive_coding = F,
  alpha_tau = 0.95,
  random_seed = random_seed
)
bcf_model <- bcf(X_train = covariate_df, Z_train = Z, y_train = y,
  pi_train = rep(0.5, length(y)),
  num_gfr = num_gfr,
  num_burnin = num_burnin, num_mcmc = num_mcmc,
  params = bcf_params)
t1 <- Sys.time()
t1 - t0

```

```
## Time difference of 9.986763 secs
```

Analysis

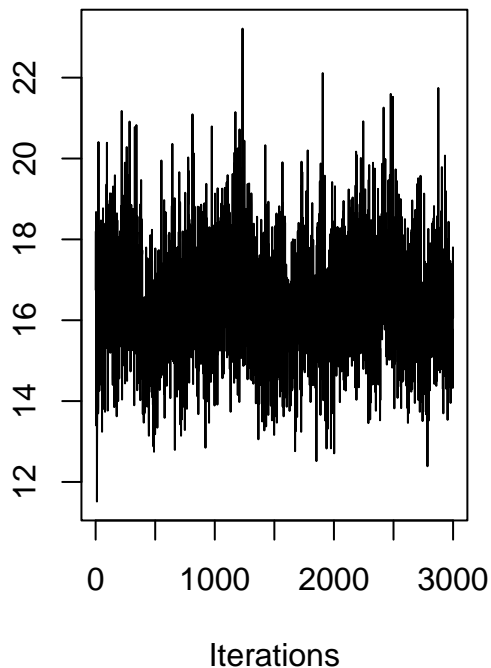
First some high-level diagnostics using coda:

```
coda_s2 = mcmc(bcf_model$sigma2_samples)
summary(coda_s2)
```

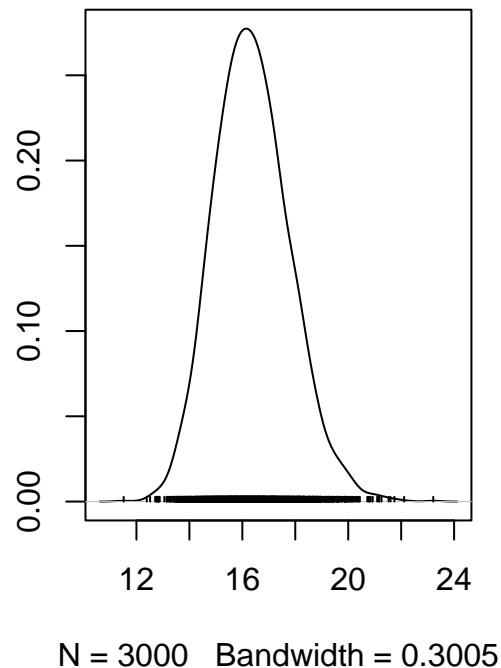
```
##
## Iterations = 1:3000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 3000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean           SD      Naive SE Time-series SE
##      16.3699      1.4188      0.0259      0.0564
##
## 2. Quantiles for each variable:
##
##  2.5%   25%   50%   75%  97.5%
## 13.82 15.38 16.30 17.26 19.37
```

```
plot(coda_s2)
```

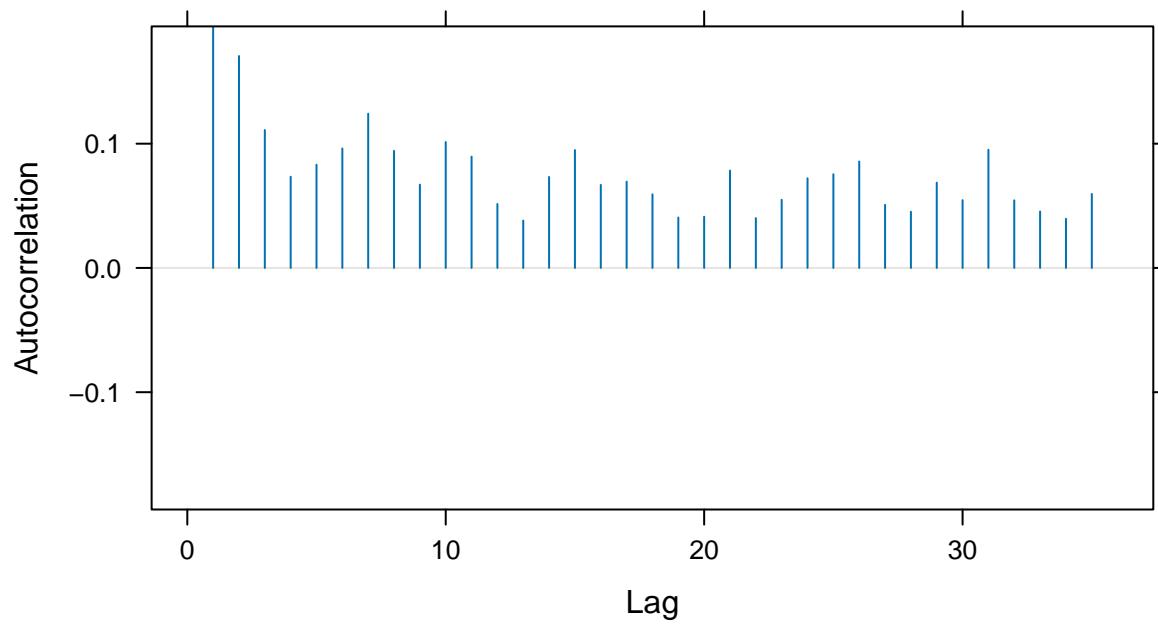
Trace of var1



Density of var1



```
acfplot(coda_s2)
```



```
effectiveSize(coda_s2)
```

```
##      var1  
## 632.928
```

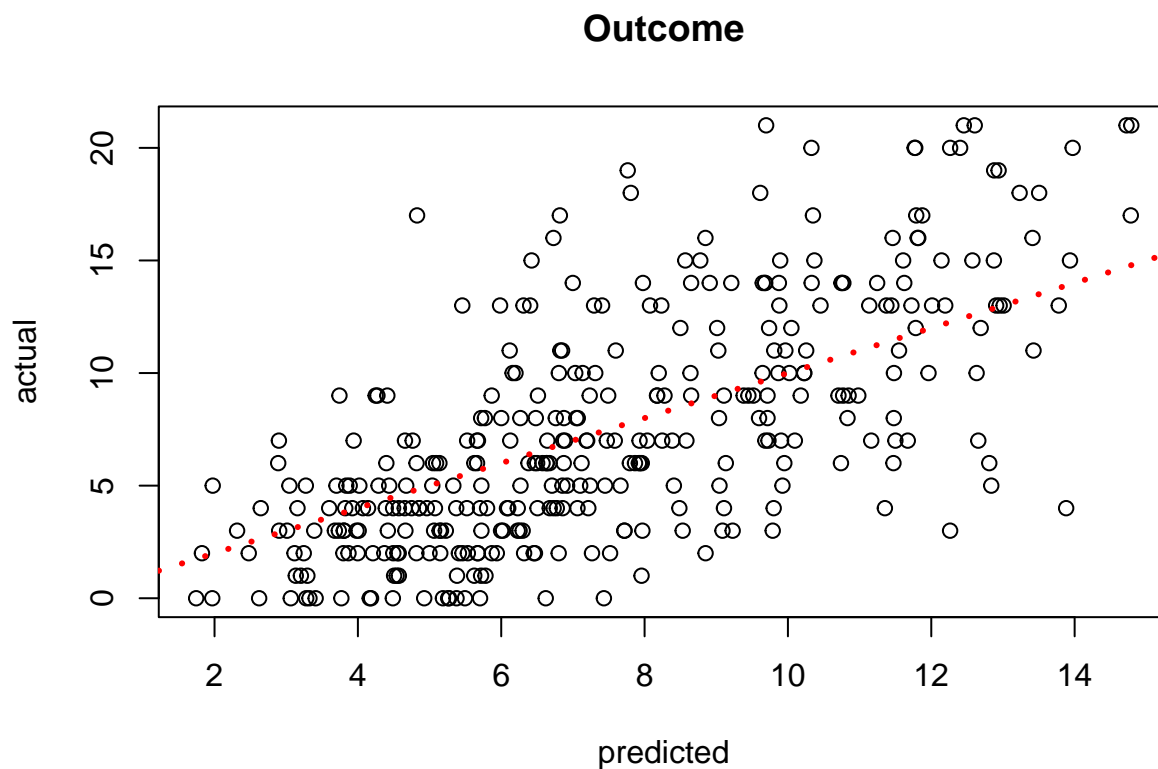
Extract observation-level outputs of the model: Fitted values, predicted values under control, and the estimated treatment effects.

```
yhat_posterior <- bcf_model$y_hat_train  
muhat_posterior <- bcf_model$mu_hat_train  
tauhat_posterior <- bcf_model$tau_hat_train/sd(y)
```

Predictive accuracy

Compare the average predicted outcome from the BCF model versus actual y , for a sanity check.

```
plot(rowMeans(yhat_posterior), y, xlab = "predicted", ylab = "actual",  
     main = "Outcome")  
abline(0,1,col="red",lty=3,lwd=3)
```



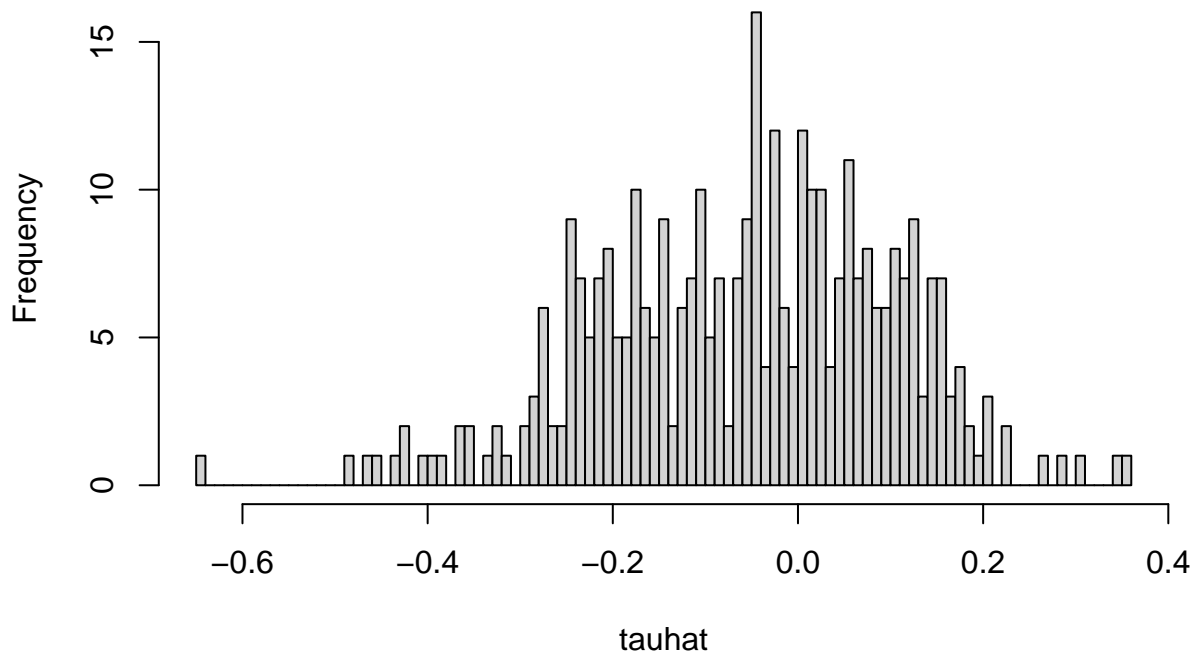
Estimated conditional average treatment effects (CATEs)

`tauhat_posterior` contains posterior samples of CATEs for each observation. Typically these are very uncertain, and inference will focus on aggregations or summaries of CATEs.

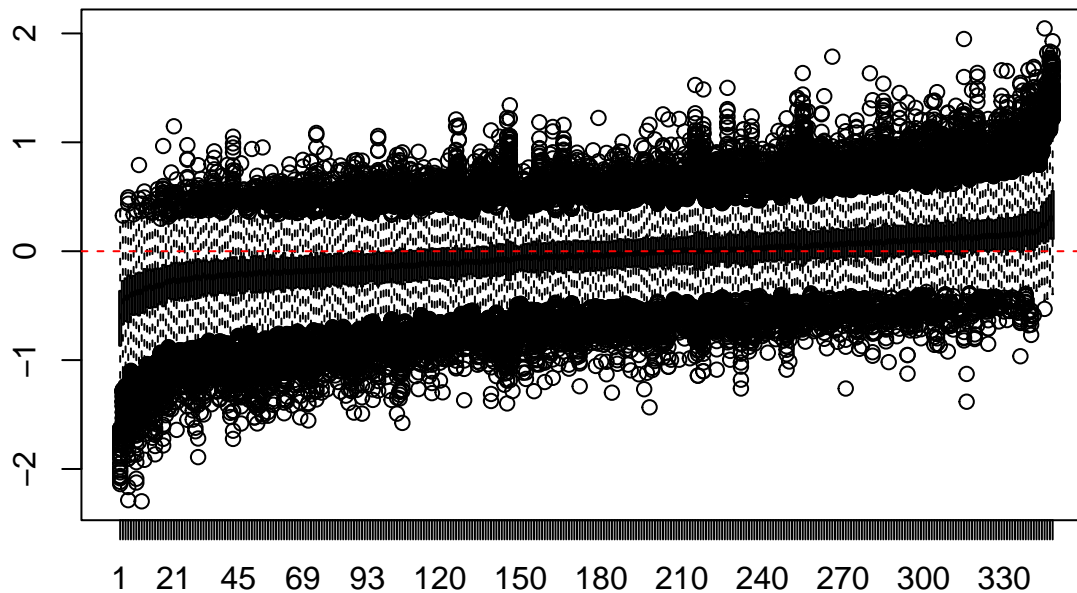
```
tauhat = rowMeans(tauhat_posterior)
```

```
hist(tauhat, breaks=100)
```

Histogram of tauhat



```
boxplot(t(tauhat_posterior)[,order(tauhat)])
abline(h=0, lty=2, col='red')
```



Average treatment effects

Extract and examine the posterior distribution of the ATE. There is little evidence to suggest a negative (beneficial) treatment effect overall. But this doesn't mean that effects are zero everywhere!

```
ate_dist <- colMeans(bcf_model$tau_hat_train)

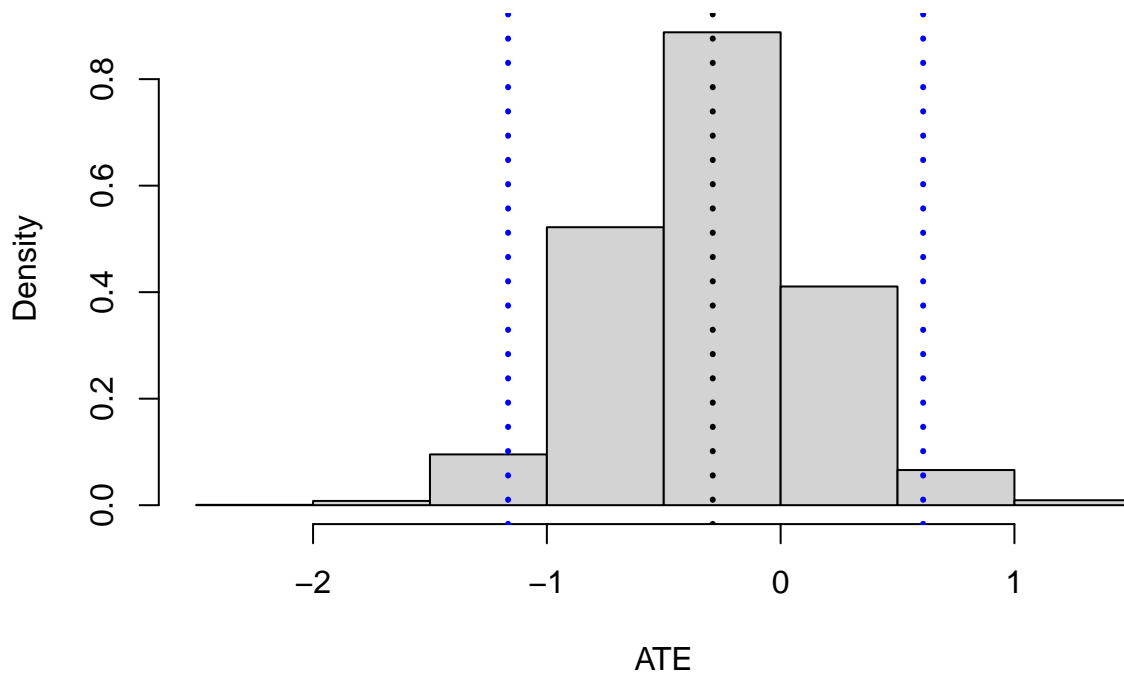
hist(ate_dist, xlab = "ATE", ylab = "Density", main = "Average Treatment Effect", freq = F)
```

```

ate <- mean(ate_dist)
ate_lb <- quantile(ate_dist, 0.025)
ate_ub <- quantile(ate_dist, 0.975)
abline(v = ate, lty = 3, lwd = 3, col = "black")
abline(v = ate_lb, lty = 3, lwd = 3, col = "blue")
abline(v = ate_ub, lty = 3, lwd = 3, col = "blue")

```

Average Treatment Effect



Checking diagnostics again

```

coda_ate = mcmc(ate_dist)
summary(coda_ate)

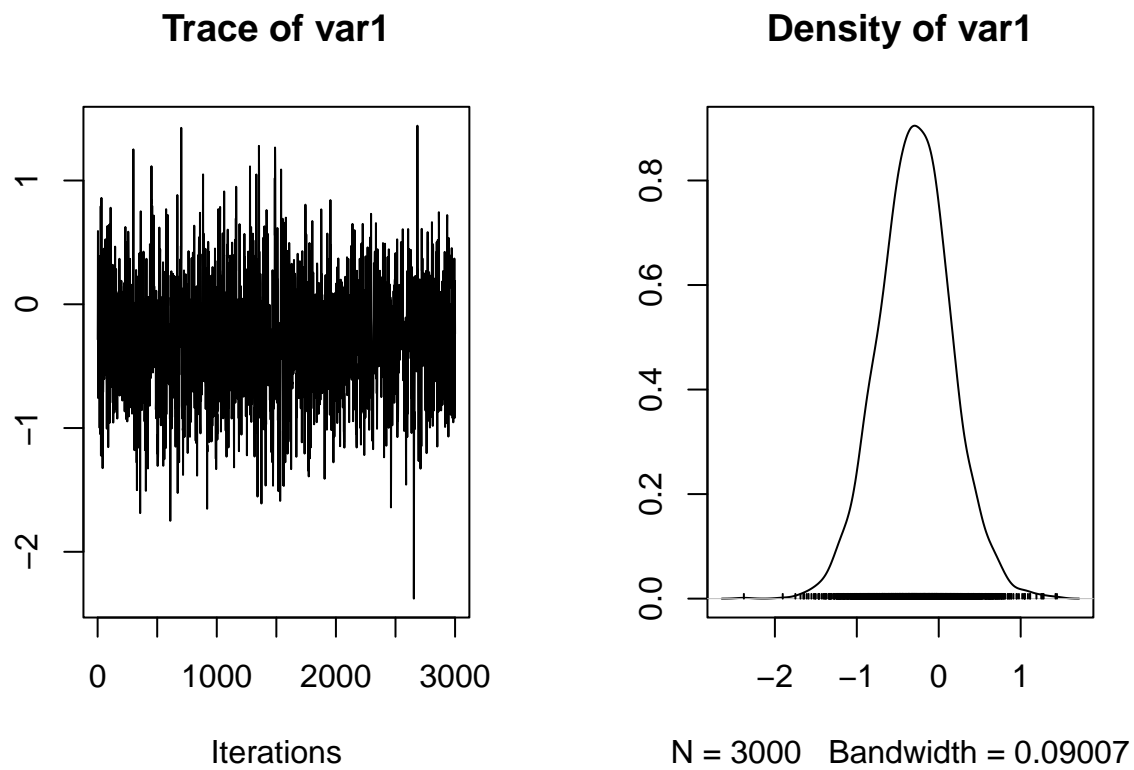
```

```

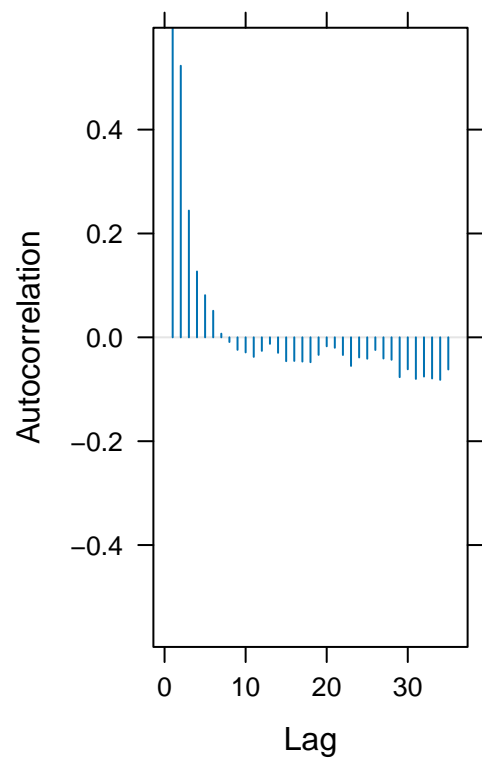
##
## Iterations = 1:3000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 3000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##      Mean      SD      Naive SE Time-series SE
##      -0.2905   0.4437   0.0081      0.0139
##
## 2. Quantiles for each variable:
##
##      2.5%    25%    50%    75%    97.5%
##      -1.16578 -0.57517 -0.29094 -0.01049  0.60955

```

```
plot(coda_ate)
```



```
acfplot(coda_ate)
```



```
effectiveSize(coda_ate)
```

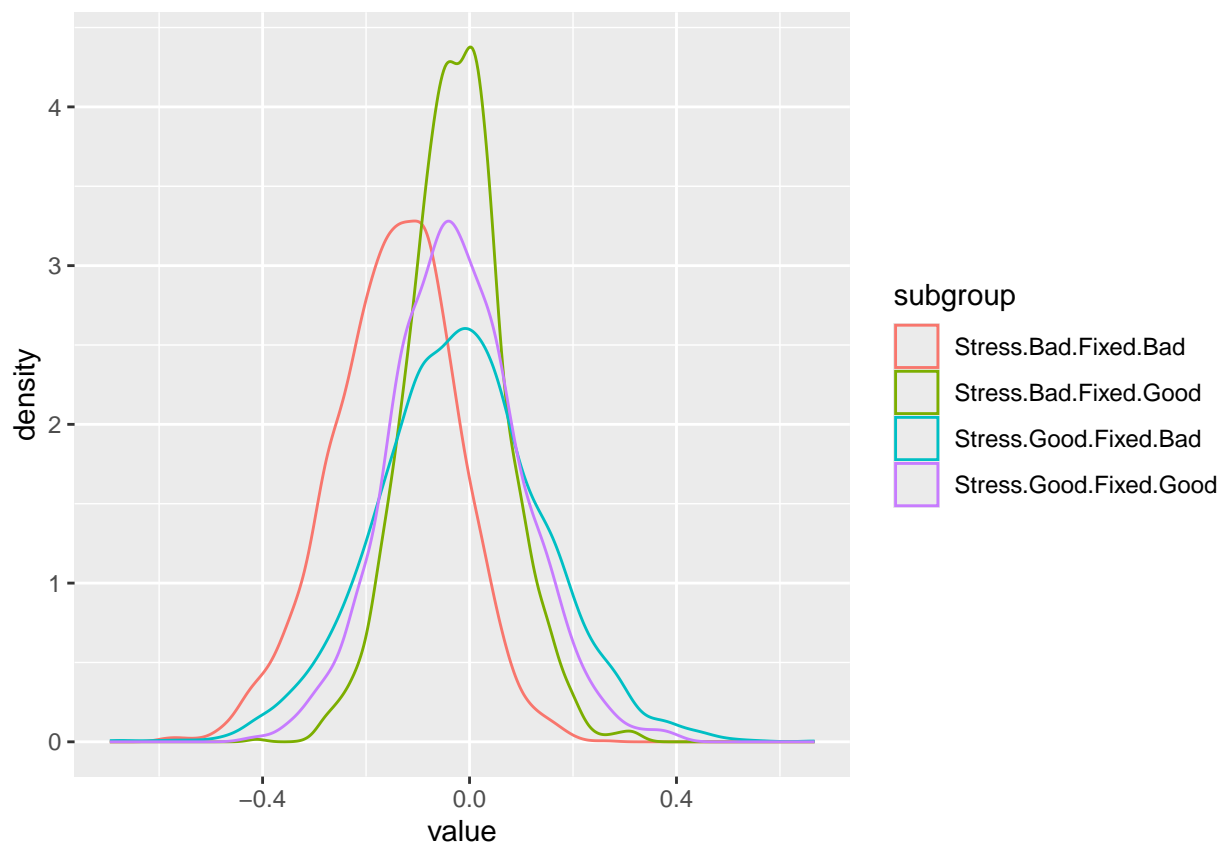


```
##      var1
## 1018.308
```

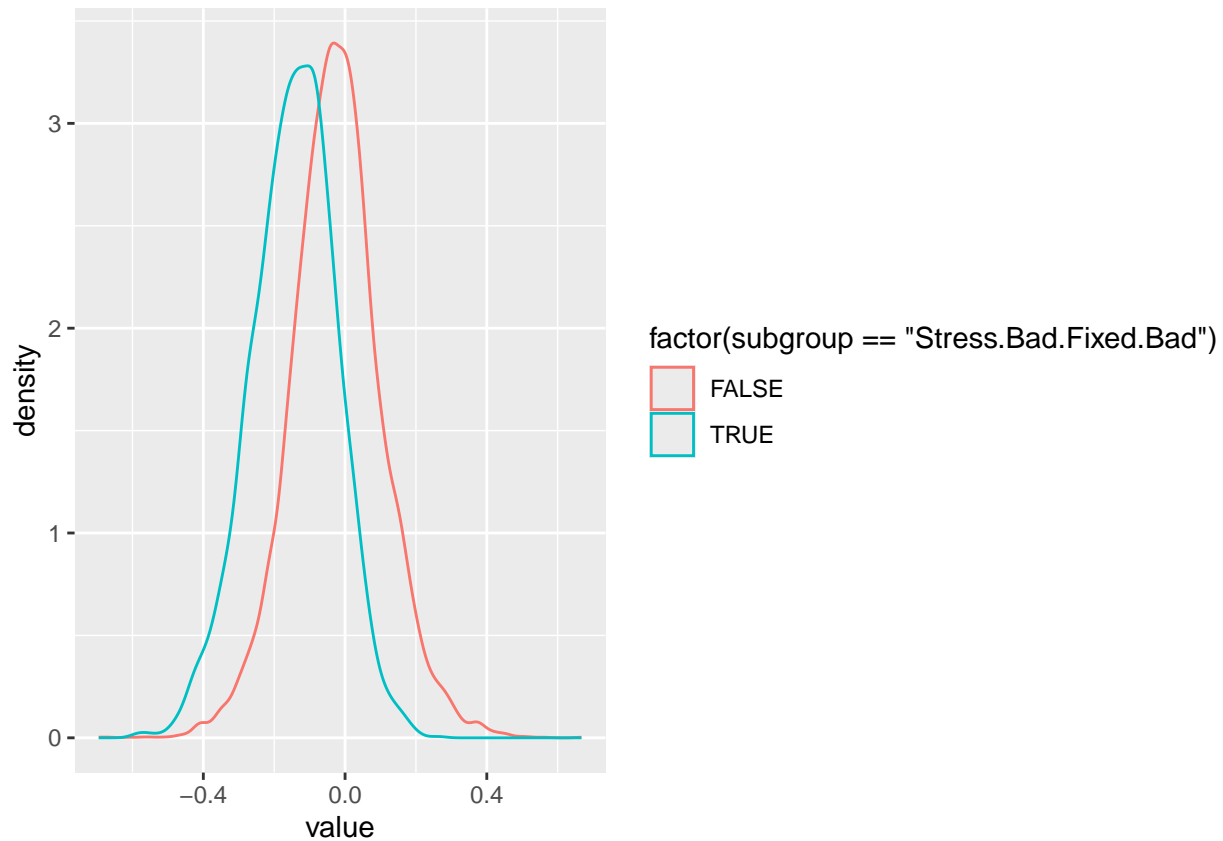
Subgroup ATEs We can start examining heterogeneity by looking at subgroup averages, first by their baseline mindsets and then by a measure of socioeconomic status.

```
subgroup_defs = covariate_df %>%
  mutate(stress_group = ifelse(stressmindset_baseline>3, "Stress.Bad", "Stress.Good"),
         fixed_group = ifelse(fixedmindset_baseline>3, "Fixed.Bad", "Fixed.Good"),
         doublemindset = interaction(stress_group, fixed_group),
         doublebad = ifelse(doublemindset=="Stress.Bad.Fixed.Bad", "Neg Prior Mindset", "Other Prior Mindset")
  )

subgroup_ates = subgroup_average_posterior(t(tauhat_posterior), subgroup_defs$doublemindset)
ggplot(aes(x=value, color=subgroup), data=gather(subgroup_ates, key="subgroup"))+geom_density()
```



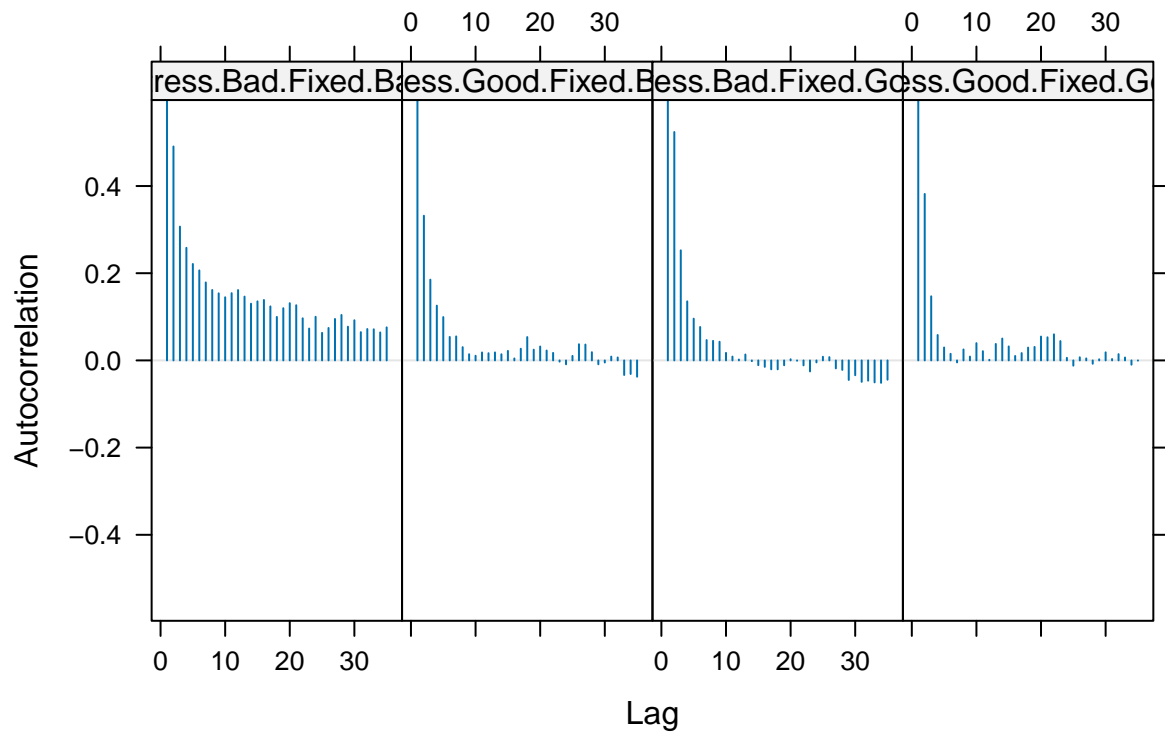
```
ggplot(aes(x=value, color=factor(subgroup=="Stress.Bad.Fixed.Bad")), data=gather(subgroup_ates, key="subgroup"))+geom_density()
```



```
coda_subate = mcmc(subgroup_ates)
summary(coda_subate)
```

```
##
## Iterations = 1:3000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 3000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## Stress.Bad.Fixed.Bad -0.14545 0.11970 0.002185      0.007780
## Stress.Good.Fixed.Bad -0.01995 0.15935 0.002909      0.004868
## Stress.Bad.Fixed.Good -0.02427 0.09412 0.001718      0.003142
## Stress.Good.Fixed.Good -0.02665 0.12544 0.002290      0.003425
##
## 2. Quantiles for each variable:
##
##              2.5%      25%      50%      75%      97.5%
## Stress.Bad.Fixed.Bad -0.3973 -0.22248 -0.13962 -0.06395 0.07424
## Stress.Good.Fixed.Bad -0.3383 -0.12296 -0.02080 0.07941 0.29366
## Stress.Bad.Fixed.Good -0.2073 -0.08497 -0.02494 0.03342 0.16623
## Stress.Good.Fixed.Good -0.2695 -0.11250 -0.03149 0.05373 0.22735
```

```
acfplot(coda_subate)
```

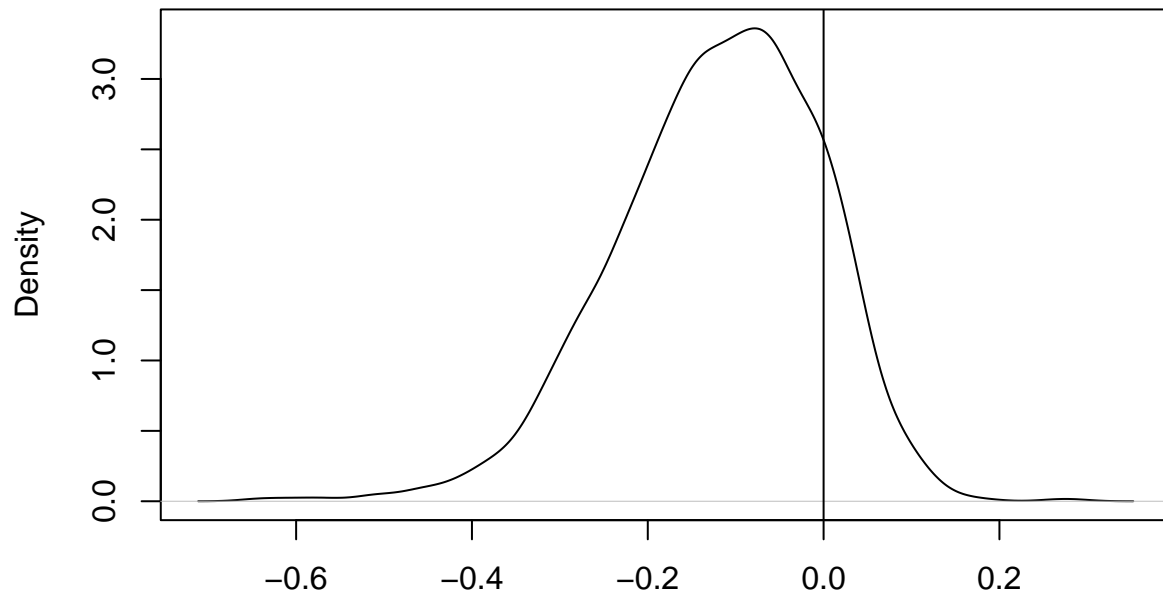


```
effectiveSize(coda_subate)
```

```
## Stress.Bad.Fixed.Bad Stress.Good.Fixed.Bad Stress.Bad.Fixed.Good
##           236.7052           1071.7127           897.5527
## Stress.Good.Fixed.Good
##           1341.5880
```

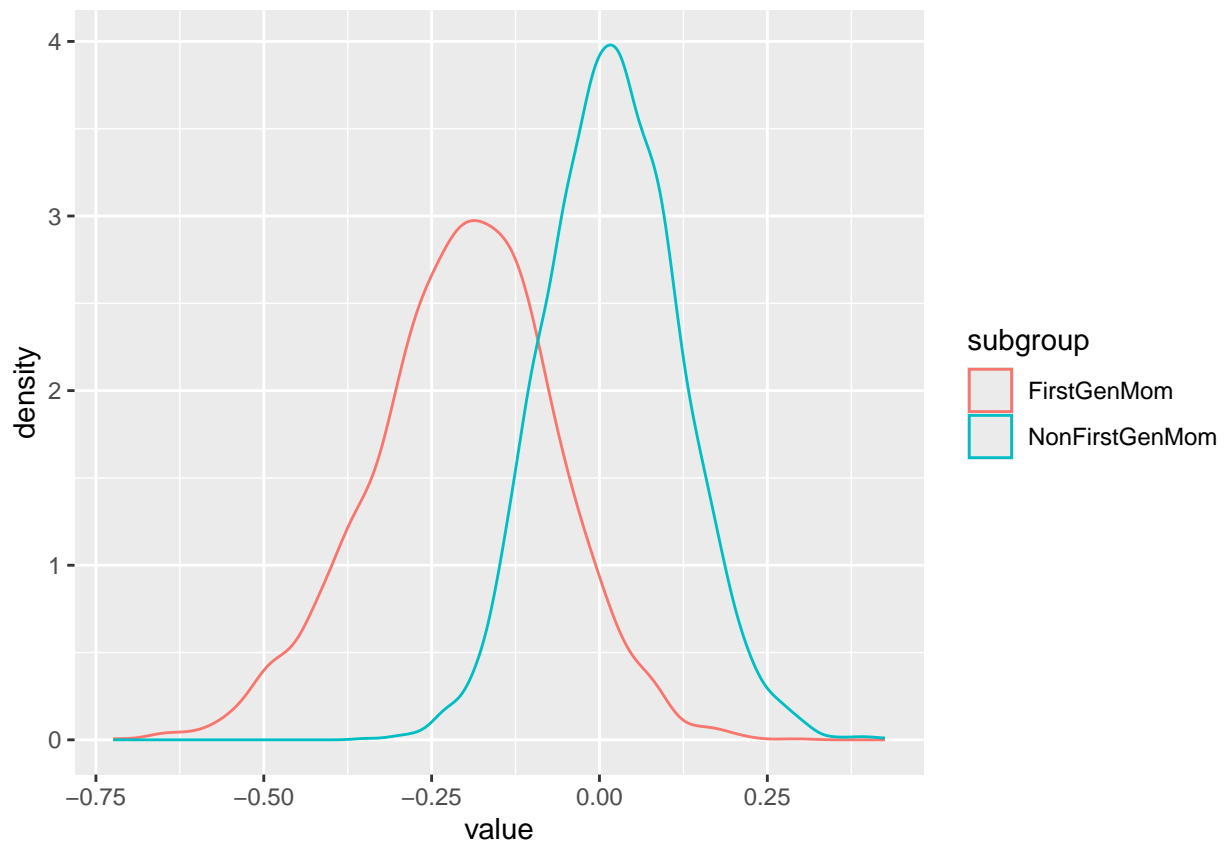
```
db_ates = subgroup_average_posterior(t(tauhat_posterior),
                                     subgroup_defs$doublebad)
plot(density(db_ates[,1] - db_ates[,2])); abline(v=0)
```

density(x = db_ates[, 1] - db_ates[, 2])



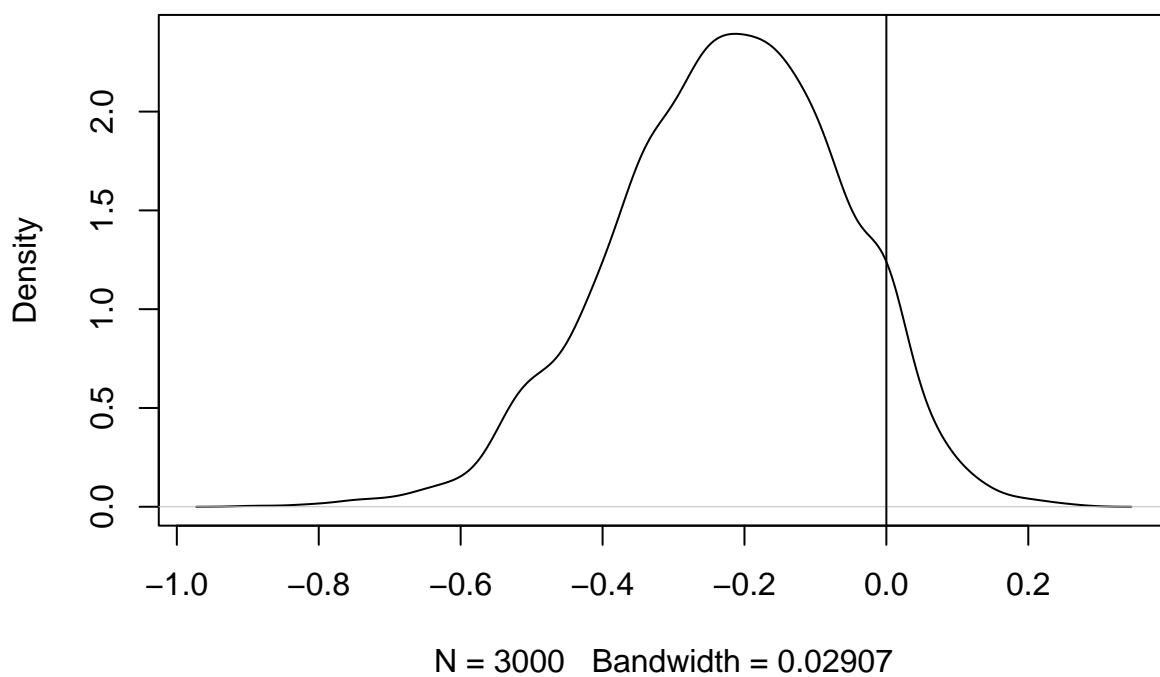
N = 3000 Bandwidth = 0.02111

```
fgen_subgroup_ates = subgroup_average_posterior(t(tauhat_posterior), ifelse(df$fgen_moth, "FirstGenMom", "SecondGenMom"))  
  
# Posterior densities  
ggplot(aes(x=value, color=subgroup),  
  data=gather(fgen_subgroup_ates, key="subgroup"))+geom_density()
```



```
plot(density(fgen_subgroup_ates[,1] - fgen_subgroup_ates[,2])); abline(v=0)
```

density(x = fgen_subgroup_ates[, 1] - fgen_subgroup_ates[, 2])

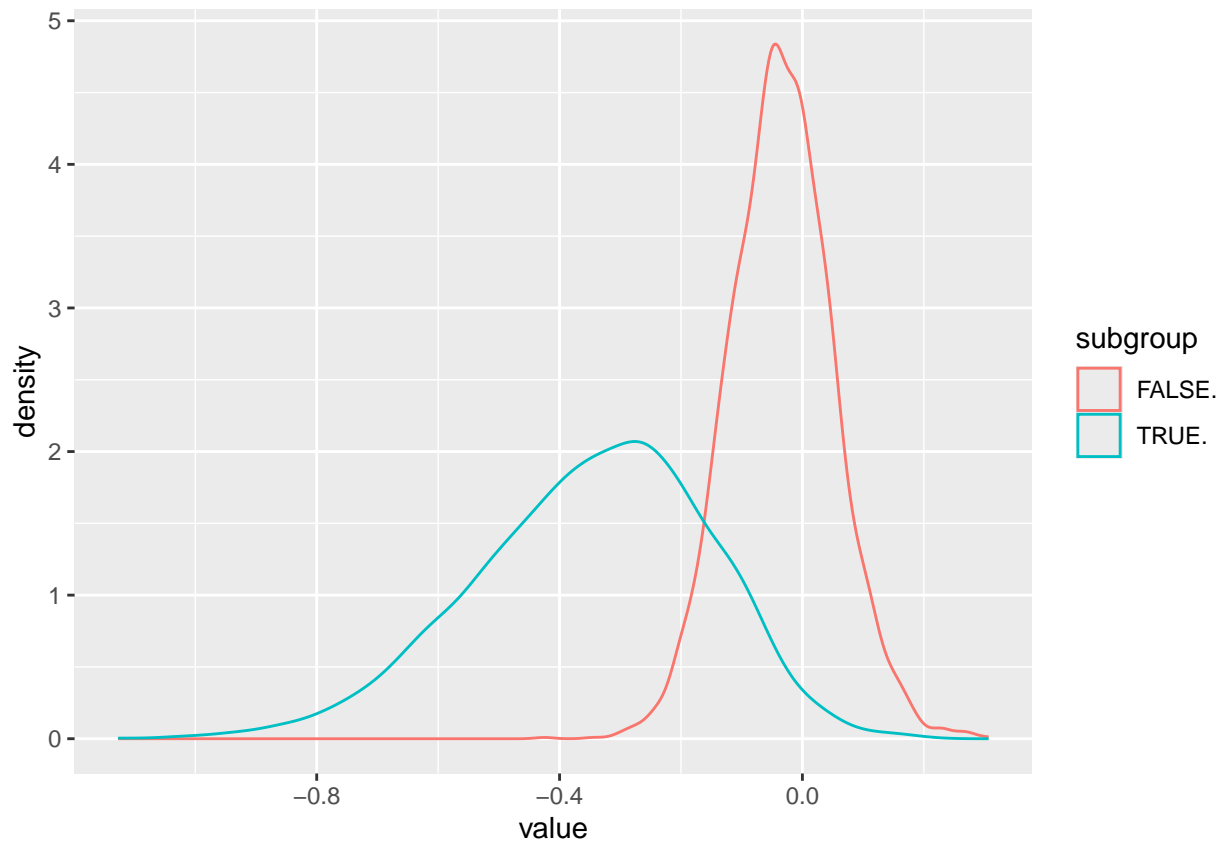


```
# Posterior prob < 0
colMeans(fgen_subgroup_ates>0)
```

```
##      FirstGenMom NonFirstGenMom
##      0.05866667      0.57100000
```

```
# Negative mindsets & firstgen
```

```
fgen_db_subgroup_ates = subgroup_average_posterior(t(tauhat_posterior), df$fgen_moth & subgroup_defs$do)
ggplot(aes(x=value, color=subgroup),
       data=gather(fgen_db_subgroup_ates, key="subgroup"))+geom_density()
```

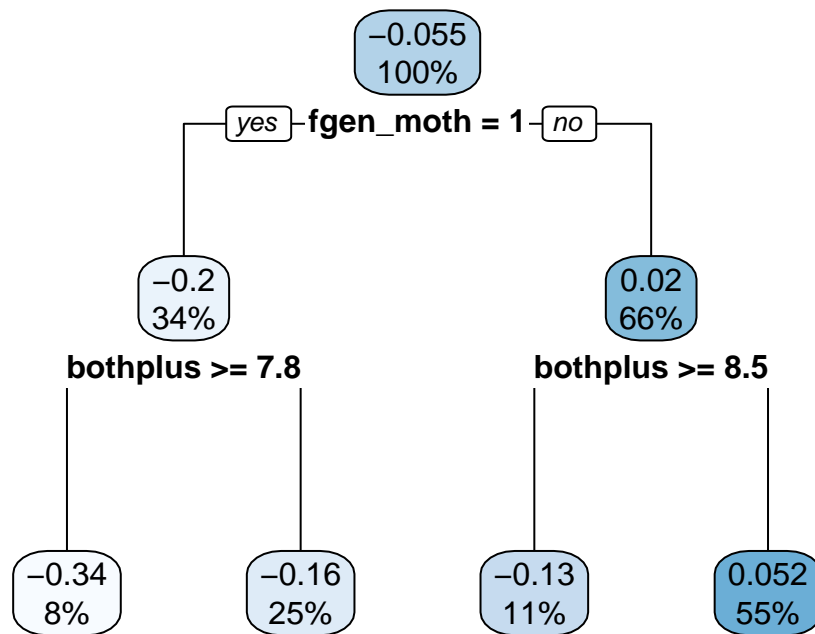


```
colMeans(fgen_db_subgroup_ates<0)
```

```
##      FALSE.      TRUE.
## 0.6683333 0.9793333
```

CART Subgroup Search We can search over possible subgroups by “fitting the fit” with a CART tree – we get back a set of subgroups that (approximately) maximally separate their subgroup ATEs.

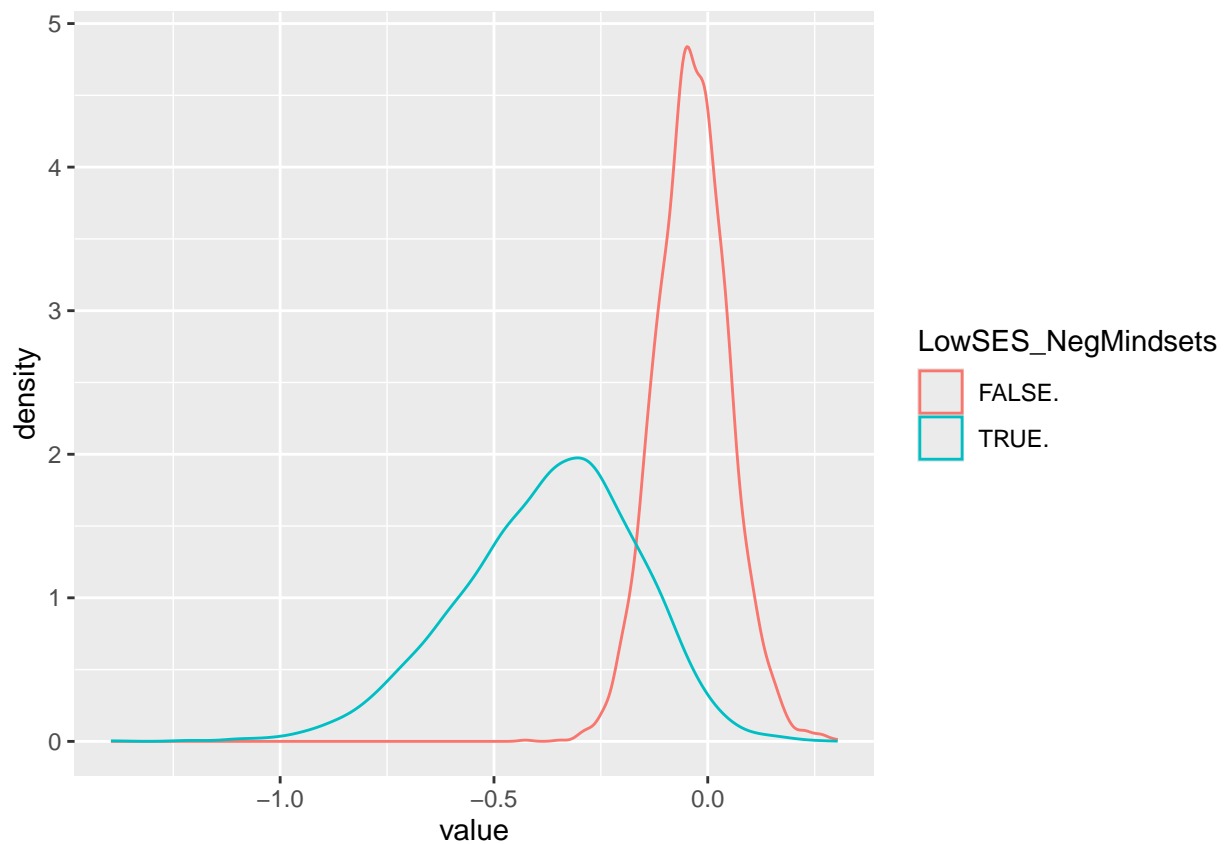
```
treefit = rpart(rowMeans(tauhat_posterior)~., data=covariate_df,
               control=rpart.control(maxdepth=2))
rpart.plot(treefit)
```



```

LowSES_NegMindsets = covariate_df$fgen_moth==1 & covariate_df$bothplus>8.2
subgroup_ates = subgroup_average_posterior(t(tauhat_posterior), LowSES_NegMindsets)
ggplot(aes(x=value, color=LowSES_NegMindsets),
  data=gather(subgroup_ates, key="LowSES_NegMindsets"))+geom_density()

```

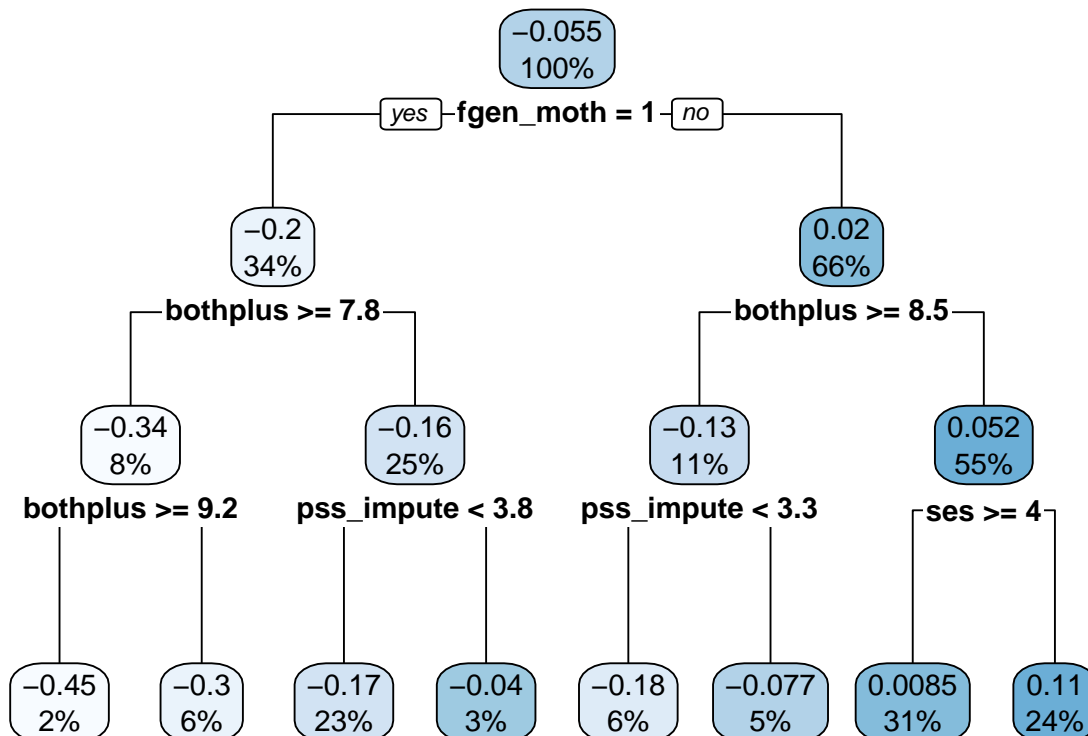


```
apply(subgroup_ates, 2, function(x) quantile(x, c(0.025, 0.1, 0.5, 0.9, 0.975)))
```

```
##          FALSE.          TRUE.
## 2.5%   -0.20142309 -0.80485811
## 10%    -0.14332841 -0.65199016
## 50%    -0.03738122 -0.35455137
## 90%     0.06959524 -0.12019726
## 97.5%  0.13570108 -0.01870874
```

Deeper trees better approximate τ at the cost of complexity and a loss of interpretability.

```
treefit = rpart(rowMeans(tauhat_posterior)~., data=covariate_df,
               control=rpart.control(maxdepth=3))
rpart.plot(treefit)
```



Additive summaries and partial effects of prior mindsets Perceived social stress (PSS) was recorded primarily to *exclude* it as a driver of any observed moderation by baseline mindsets – that is, we’d like to partial it out of our estimates of effect variation by baseline mindsets. Subgroup analysis is not well suited to estimating partial effects. Instead we try additive summaries using the **possum** package:

```
# Appending the model fits to the data frame
model_df <- covariate_df
model_df$tauhat <- rowMeans(bcf_model$tau_hat_train)

# Specify the model summary formula using our three focal variables, plus fgen_moth

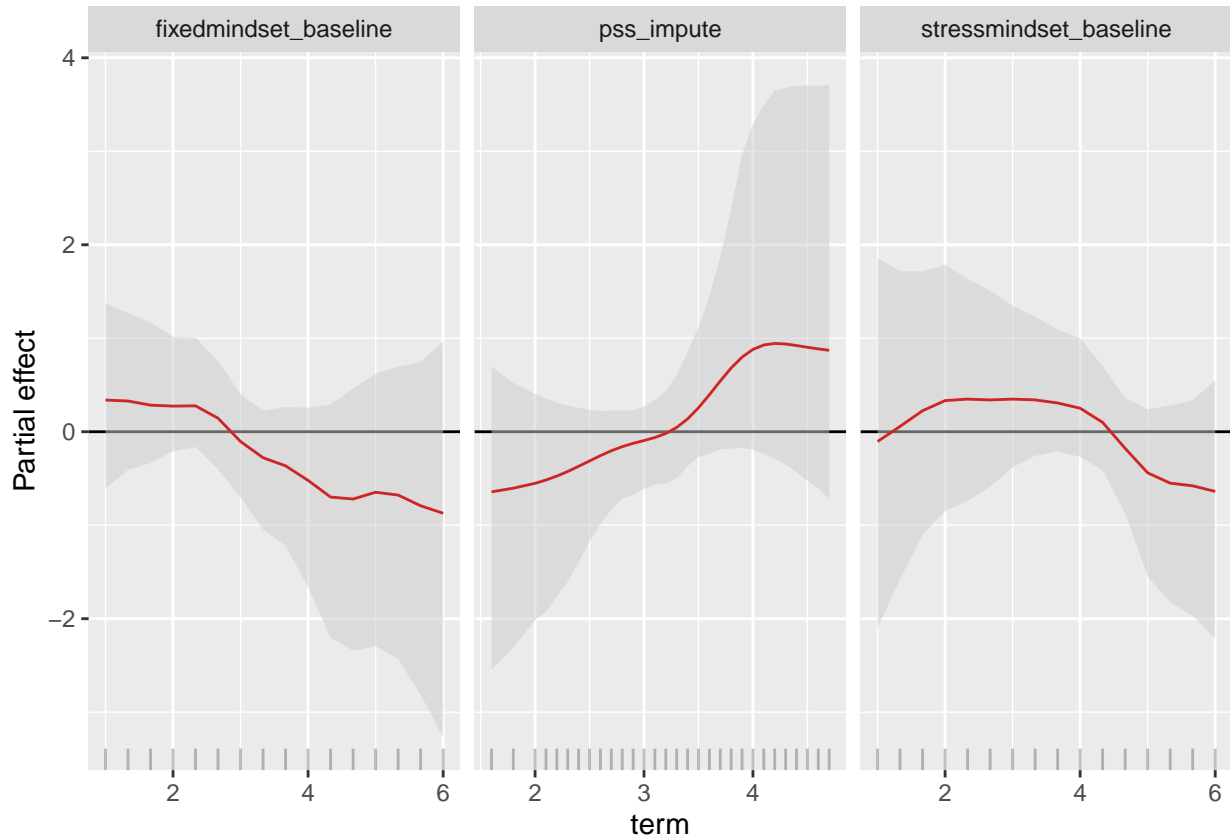
# Explicitly call out categorical variables by wrapping them in factor()
# in the formula specification
ff = tauhat ~ factor(fgen_moth) +
  s(stressmindset_baseline) + s(fixedmindset_baseline) + s(pss_impute)
```



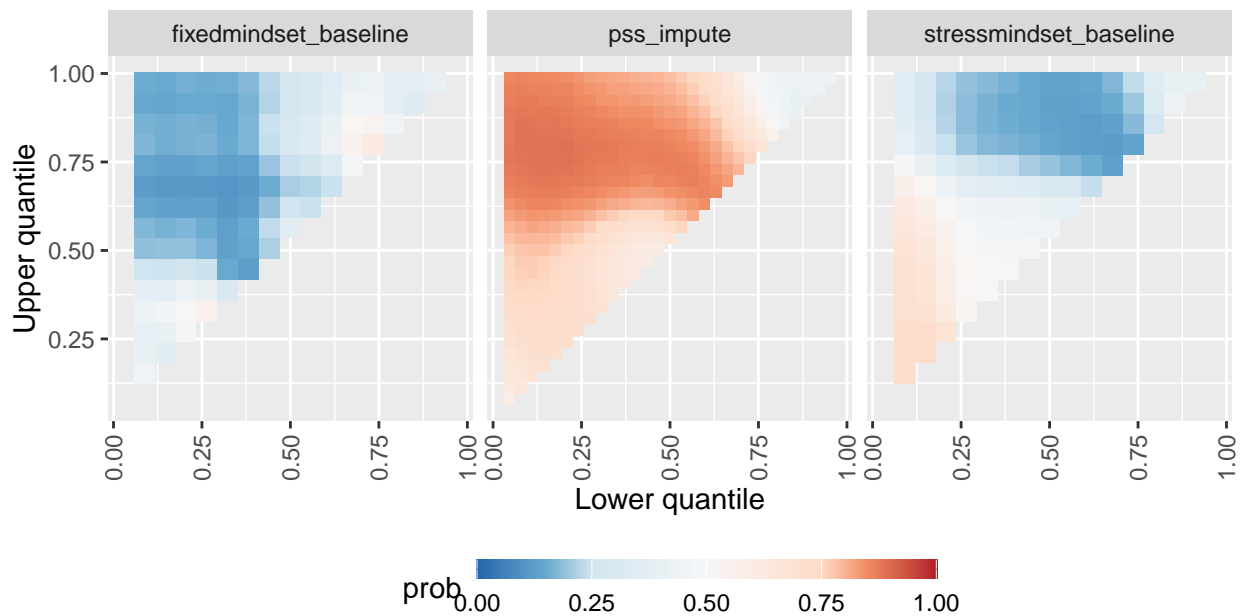
```
gf = additive_summary(ff, bcf_model$tau_hat_train, df=model_df, fast=FALSE)
```

```
## 348 3000348 3000
```

```
additive_summary_plot(gf)
```



```
additive_summary_triangle_plot(gf)
```



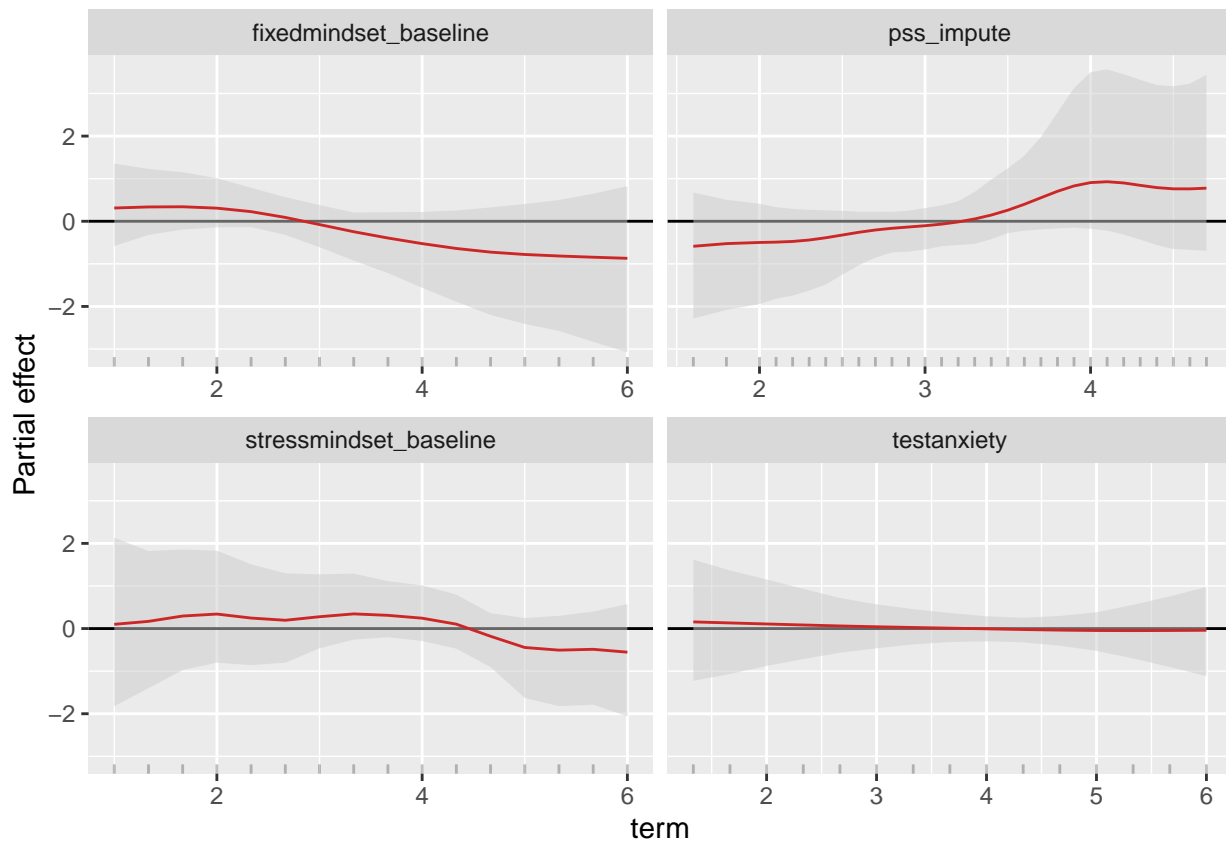
Once more, with all the variables:

```
ff = tauhat ~ factor(fgen_moth) + factor(fgen_fath) + factor(ses) + factor(sex) +  
  s(testanxiety) + s(stressmindset_baseline) + s(fixedmindset_baseline) + s(pss_impute)
```

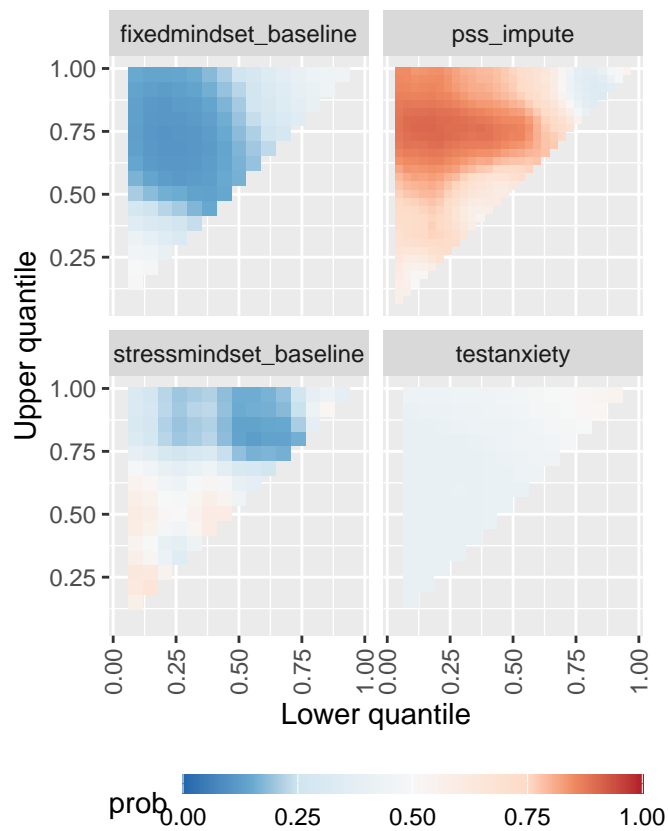
```
gf = additive_summary(ff, bcf_model$tau_hat_train, df=model_df, fast=FALSE)
```

```
## 348 3000348 3000
```

```
additive_summary_plot(gf)
```



```
additive_summary_triangle_plot(gf)
```



```

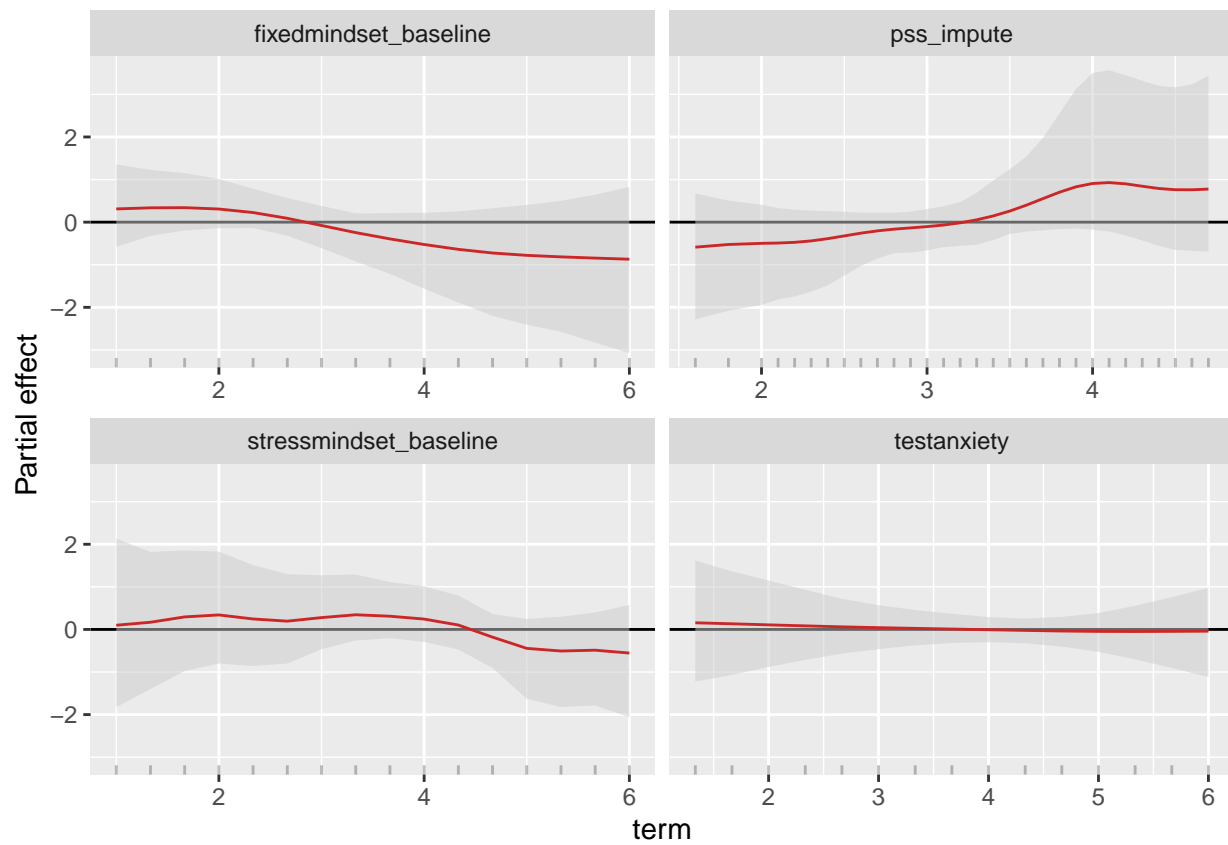
gfpm = gam(rowMeans(bcf_model$tau_hat_train) ~ factor(fgen_moth) + factor(fgen_fath) +
  factor(ses) + factor(sex) + factor(ses) +
  s(testanxiety) + s(fixedmindset_baseline, stressmindset_baseline) + s(pss_impute), data=covariate_df)

gf = additive_summary(ff, bcf_model$tau_hat_train, df=covariate_df, fast=FALSE)

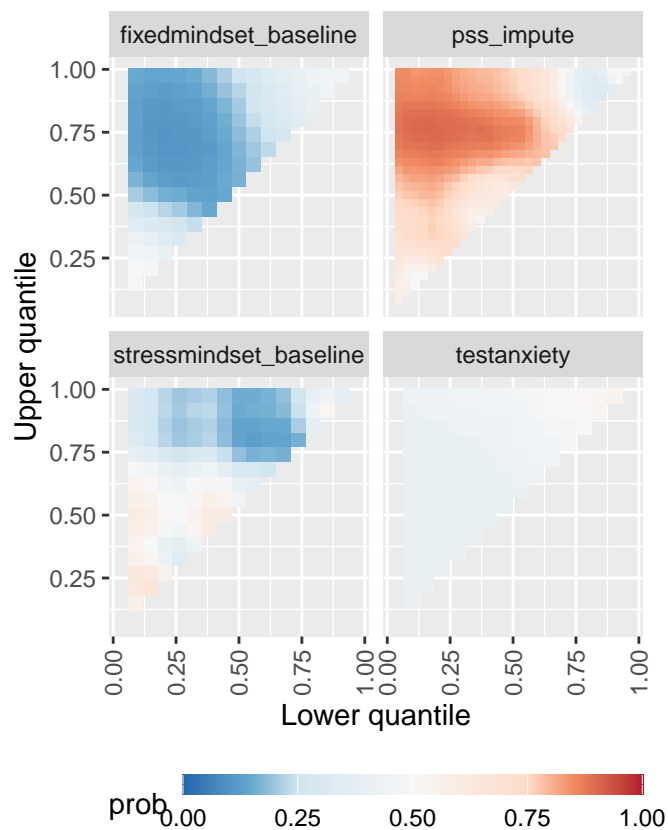
## 348 3000348 3000

additive_summary_plot(gf)

```



```
additive_summary_triangle_plot(gf)
```

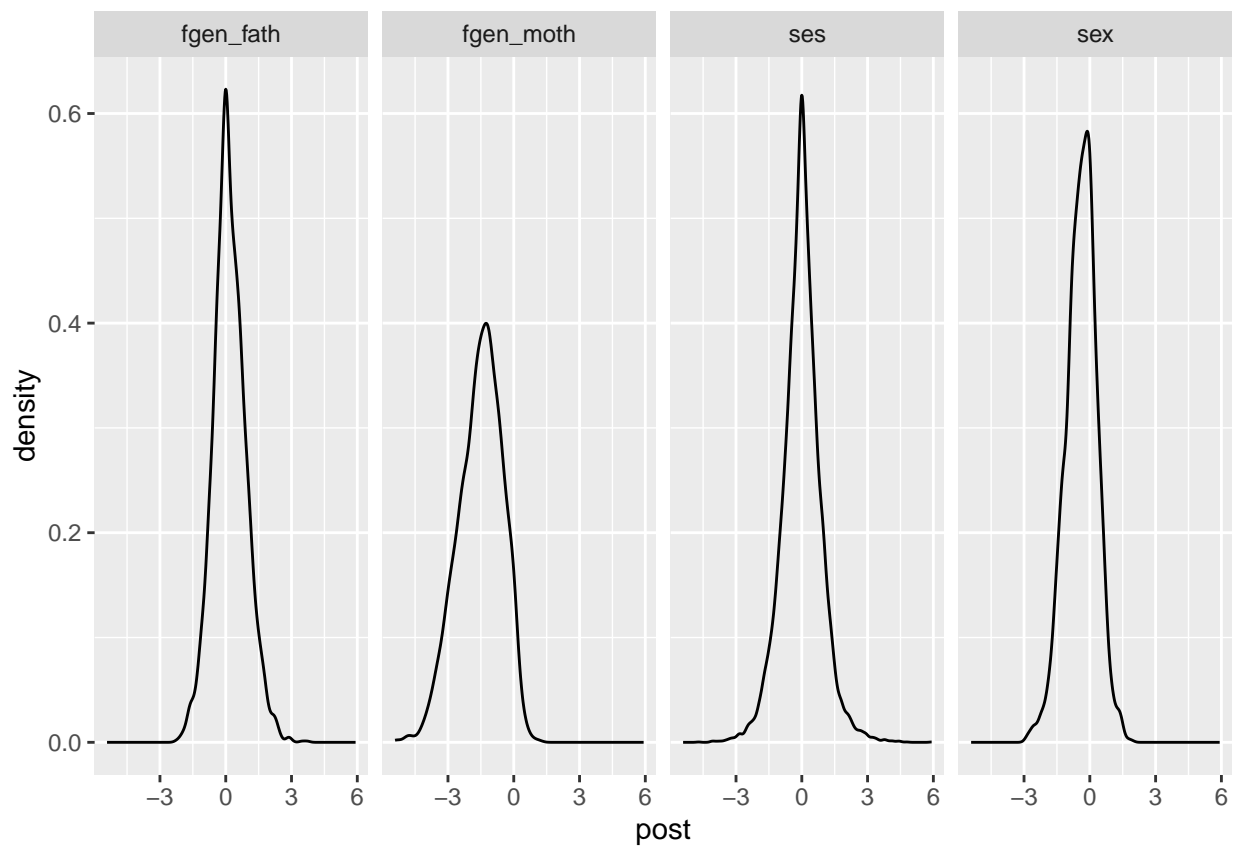


Looking at (approximate) partial effects of categorical variables:

```
factor_post = get_additive_factor_posterior(gf)
factor_post %>%
  group_by(term) %>%
  summarize(est = mean(post), lo=quantile(post, 0.05), hi=quantile(post, 0.95))
```

```
## # A tibble: 4 x 4
##   term      est    lo    hi
##   <chr>    <dbl> <dbl> <dbl>
## 1 fgen_fath 0.172 -1.03  1.46
## 2 fgen_moth -1.54  -3.33 -0.0171
## 3 ses      -0.0114 -1.47  1.39
## 4 sex      -0.399  -1.59  0.682
```

```
ggplot(aes(x=post), data=factor_post) + geom_density() + facet_grid(~term)
```



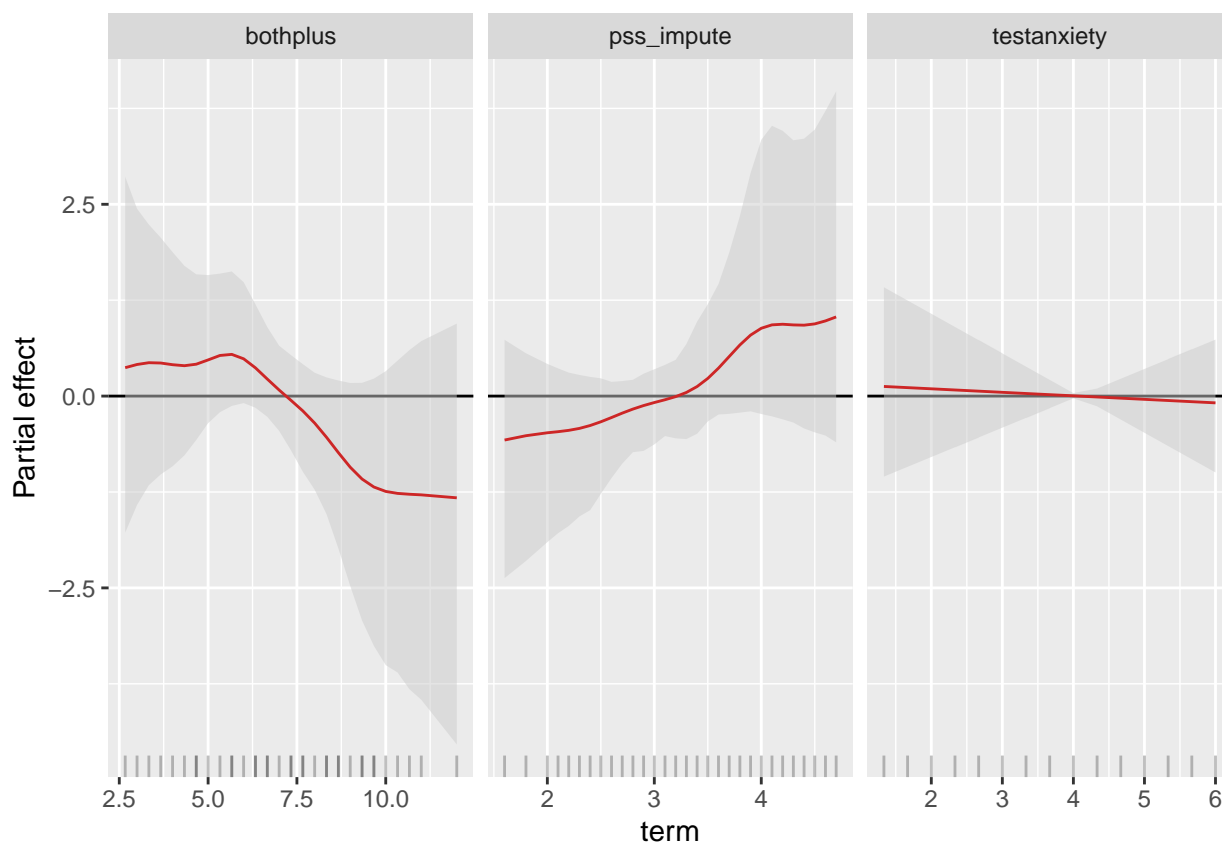
And again, summarizing by the sum of the two mindset scores instead of the individual scores

```
ff = tauhat ~ factor(fgen_moth) + factor(fgen_fath) + factor(ses) + factor(sex) +
  s(testanxiety) + s(bothplus) + s(pss_impute)

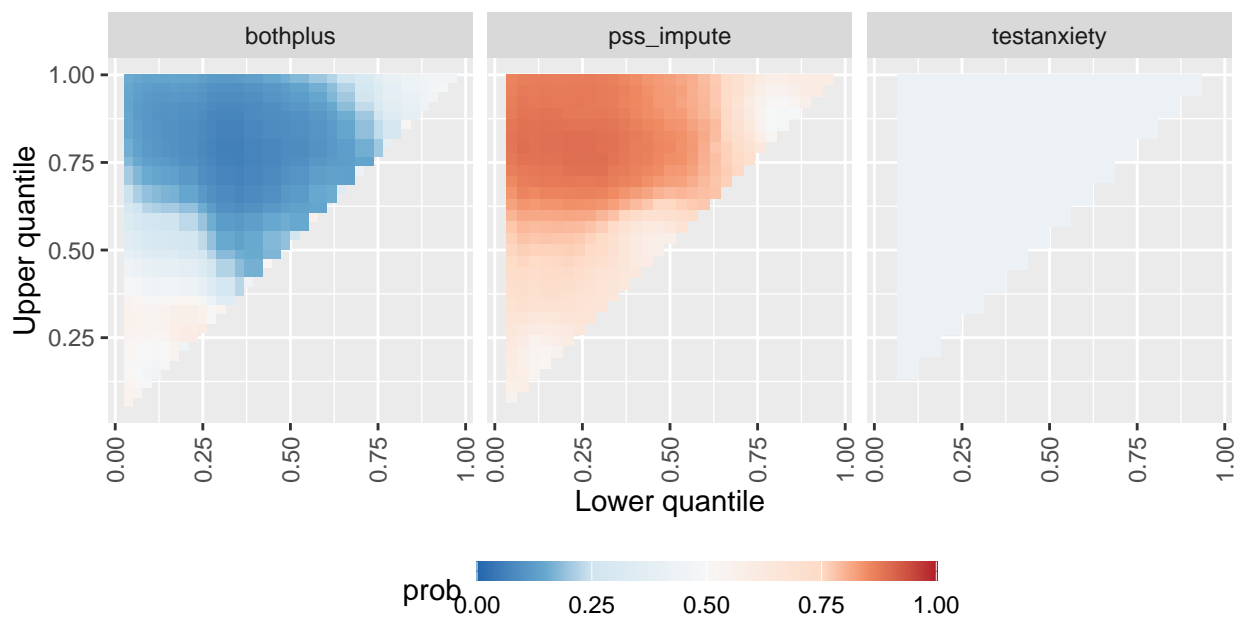
gf = additive_summary(ff, bcf_model$tau_hat_train, df=model_df, fast=FALSE)
```

```
## 348 3000348 3000
```

```
additive_summary_plot(gf)
```



```
additive_summary_triangle_plot(gf)
```



References