Data Analysis 2

Ziying Li, Matt Capaldi, Yujuan Gao, Olivia Morales

October 26, 2022

```r
## libraries
libs <- c("tidyverse", "haven", "bibtex",
          "psych", "knitr", "pastecs", "kableExtra","survey",
          "cobalt", "randomForest", "ipred","rpart", "baguette",
          "parsnip", "SimDesign", "bartCause", "lme4", "grf",
          "GenericML", "car", "bartMachine", "gtools", "patchwork")

sapply(libs, require, character.only = TRUE)
```

```
##    tidyverse        haven       bibtex        psych        knitr      pastecs
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##   kableExtra       survey       cobalt randomForest        ipred        rpart
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##     baguette      parsnip    SimDesign    bartCause         lme4          grf
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##    GenericML          car  bartMachine       gtools    patchwork
##         TRUE         TRUE         TRUE         TRUE         TRUE
```

```r
covariateNames <- c(
    "X1RTHETK1",
    "X1MTHETK1",
    "X1TCHAPP",
    "X1TCHCON",
    "X1TCHPER",
    "X1TCHEXT",
    "X1TCHINT",
    "X1ATTNFS",
    "X1INBCNT",
    "X12MOMAR",
    "X1NUMSIB",
    "P1OLDMOM",
    "P1CHLDBK",
    "P2DISTHM",
    "P1NUMPLA",
    "T2PARIN",
    "X12PAR1ED_I",
    "X12PAR2ED_I",
    "X2INCCAT_I",
    "X1PAR1EMP",
    "S2LUNCH",
    "X2KRCETH",
    "S2NGHBOR",
    "S2OUTSID",
    "S2USDABR",
    "S2PUBSOC",
    "X1LOCALE",
    "S1_ID",
    "W1_2P0PSU",
    "prop.missing",
    "X_CHSEX_R",
    "X_RACETH_R",
    "P1HSCALE")

## directory path (assignment_2 as current working directory)

data_dir <- file.path(".", "data")

## loading data
load(file.path(data_dir, "chapter_10_data_cleaned_and_imputed.Rdata"))

#standardized continuous predictors
for (var in covariateNames) {
  if (class(data[,var])!="factor") { data[,var] = (data[,var]-mean(data[,var]))/sd(data[,var]) } }
```

In this paper, we explore heterogeneity of treatment using three different methods; CausalBART, GenericML, and CausalForest. We will explore the heterogeneity (Questions 1, 3, and 4) separately for each model before concluding and comparing CATEs (Question 2) at the conclusion of the paper.

```
psFormula <- paste(covariateNames, collapse="+")
psFormula <- formula(paste("treated~", psFormula, sep=""))
print(psFormula)
```

```
## treated ~ X1RTHETK1 + X1MTHETK1 + X1TCHAPP + X1TCHCON + X1TCHPER +
##     X1TCHEXT + X1TCHINT + X1ATTNFS + X1INBCNT + X12MOMAR + X1NUMSIB +
##     P1OLDMOM + P1CHLDBK + P2DISTHM + P1NUMPLA + T2PARIN + X12PAR1ED_I +
##     X12PAR2ED_I + X2INCCAT_I + X1PAR1EMP + S2LUNCH + X2KRCETH +
##     S2NGHBOR + S2OUTSID + S2USDABR + S2PUBSOC + X1LOCALE + S1_ID +
##     W1_2P0PSU + prop.missing + X_CHSEX_R + X_RACETH_R + P1HSCALE
```

BART

```
set.seed(123)
# credit to Matt, esp for the covs matrices!
train <- data %>%
  sample_frac(size = 0.5)
test <- anti_join(data, train)

train <- train %>% mutate(across(.fns = as.numeric))
test <- test %>% mutate(across(.fns = as.numeric))
train$treated <- train$treated - 1
test$treated <- test$treated - 1

#matrix_covs <- as.matrix(train %>% select(all_of(covariateNames)) %>%
                          # mutate(across(.fns = as.numeric)))
```

```r
# estimating conditional average treatment effects (CATEs) using BART

bart <- bartc(response = train$X2MTHETK1,
             treatment = train$treated,
             confounders = data.frame(train[,covariateNames]),
             method.rsp = "bart",
             method.trt = "glm",
             keepTrees = TRUE,
             estimand = "ate")
```

```
## fitting treatment model via method 'glm'
## fitting response model via method 'bart'
```

```r
cate <- predict(bart,
               newdata = data.frame(test[,covariateNames]),
               type = "icate")

cate_m <- apply(cate, 2, mean)

test$cate <- apply(cate, 2, mean)

modely <- summary(bart, target = "cate")
```

We first attempted estimating the conditional average treatment effects (CATEs) of early childhood care on mathematics scores using Bayesian Additive Regression Trees (BART) for the entire dataset and for all the covariates estimated in the original propensity score model. We found that splitting the data into training and testing data mitigated severe overfitting issues (particularly in calculating the propensity scores).

note: the bartc function does not allow the typical adjustments to hyperparamters characteristic of the other functions like genericML and causal_forest, so we decided to focus just on making adjustments to the arguments of the functions as needed.

We also tried estimating the propensity scores & CATEs both with BART, but found that logistic regression was better suited in estimating the scores while still using BART for the CATEs.

The final model (modely) provides the CATE derived from the fit BART model, demonstrating no substantive treatment effect heterogeneity.

```r
#library(bartMachine) already in libs list

set.seed(123)
bart_machine <- bartMachine(X=data.frame(train[,covariateNames]),
                            y=cate_m,
                            serialize = T,
                            mem_cache_for_speed = F,
                            seed = 20221026)
```

```
## bartMachine initializing with 50 trees...
## bartMachine vars checked...
## bartMachine java init...
## bartMachine factors created...
## bartMachine before preprocess...
## bartMachine after preprocess... 33 total features...
## bartMachine sigsq estimated...
## bartMachine training data finalized...
## Now building bartMachine for regression...
## evaluating in sample data...done
## serializing in order to be saved for future R sessions...done
```

```
var.importance <- investigate_var_importance(bart_machine,
                                             num_replicates_for_avg = 20,
                                             plot = F)
```

```
## ....................
```

```
#select variables whose importance is greater than the median
important.vars = names(var.importance$avg_var_props)[
  var.importance$avg_var_props > median(var.importance$avg_var_props)]

bart2 <- bartc(response = test$X2MTHETK1,
              treatment = test$treated,
              confounders = data.frame(test[,important.vars]),
              method.rsp = "bart",
              method.trt = "glm",
              estimand = "ate",
              keepTrees = T)
```

```
## fitting treatment model via method 'glm'
## fitting response model via method 'bart'
```

```
cate2 <- predict(bart2,
                newdata = data.frame(test[,important.vars]),
                type = "icate")

test$cate2 <- apply(cate2, 2, mean)

modely2 <- summary(bart2, target = "cate")

model <- paste(important.vars, collapse="+")
model <- paste(c("cate2",model), collapse="~")
model.cate <- lm(model,data = test)
summary(model.cate)
```

```
##
## Call:
## lm(formula = model, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27755 -0.01348 -0.00312  0.00768  0.60862
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.880e-02  2.487e-03   7.560 4.61e-14 ***
## W1_2P0PSU     1.729e-03  5.281e-04   3.274  0.00107 **
## P1CHLDBK     -5.252e-04  5.929e-04  -0.886  0.37575
## X1NUMSIB     -2.381e-03  5.221e-04  -4.561 5.18e-06 ***
## X1TCHCON     -1.104e-03  7.200e-04  -1.533  0.12532
## X1MTHETK1    -2.314e-02  8.536e-04 -27.112  < 2e-16 ***
## S1_ID        -3.049e-03  5.214e-04  -5.848 5.23e-09 ***
## X2INCCAT_I    9.841e-04  6.005e-04   1.639  0.10129
## T2PARIN       4.882e-04  7.157e-04   0.682  0.49518
## P1NUMPLA     -2.936e-04  4.773e-04  -0.615  0.53859
## prop.missing  5.987e-04  5.374e-04   1.114  0.26531
## X_RACETH_R    2.536e-04  4.387e-04   0.578  0.56330
## X1RTHETK1     1.173e-02  8.397e-04  13.972  < 2e-16 ***
## X1TCHINT      1.633e-03  5.418e-04   3.015  0.00258 **
## X1ATTNFS      6.761e-05  6.671e-04   0.101  0.91928
## X1TCHPER     -1.838e-03  7.359e-04  -2.498  0.01253 *
## X1LOCALE     -7.646e-04  4.786e-04  -1.598  0.11018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04137 on 6325 degrees of freedom
## Multiple R-squared:  0.151,  Adjusted R-squared:  0.1488
## F-statistic:  70.3 on 16 and 6325 DF,  p-value: < 2.2e-16
```

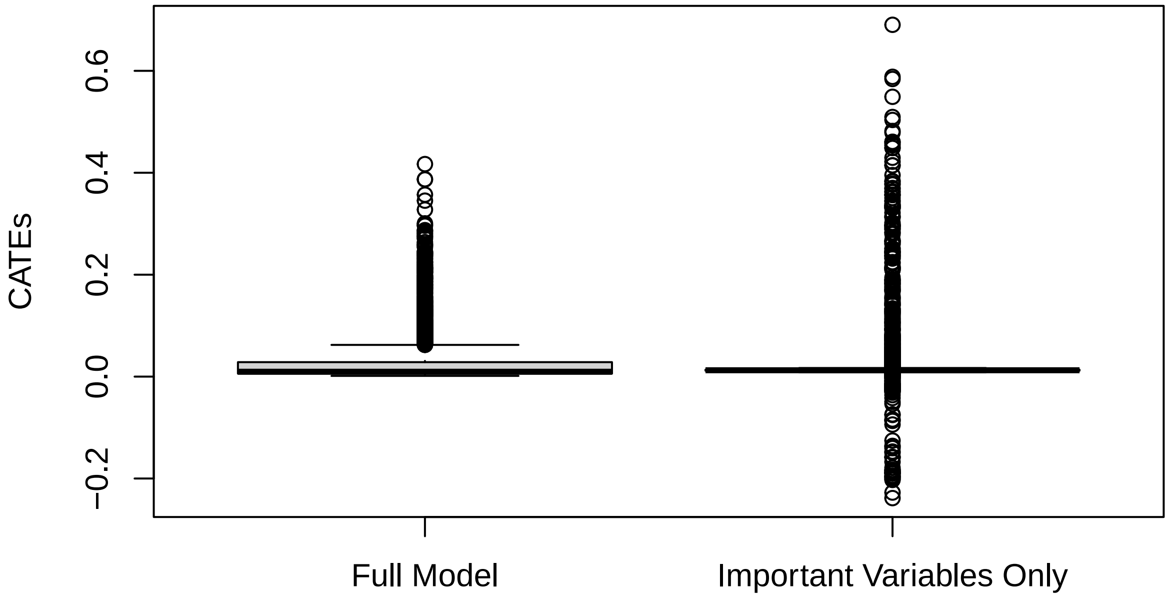
We then identified the most important variables (selecting those variables with inclusion proportions greater than the median proportion for the BART model). Those variables were then used to fit another BART object and predict CATEs once again. The model CATE was slightly greater than the first estimation with all the variables included, but not by much.

note: we reduced the num_replicates_for_avg due to questions of efficiency; we decided to decrease from 25 to 20. We also noticed the top variables in the proportion figure kept changing, even when setting a seed. As such, we identified the two predictors that occurred the most in multiple runs of the investigate_var_importance function. Results are identified below:

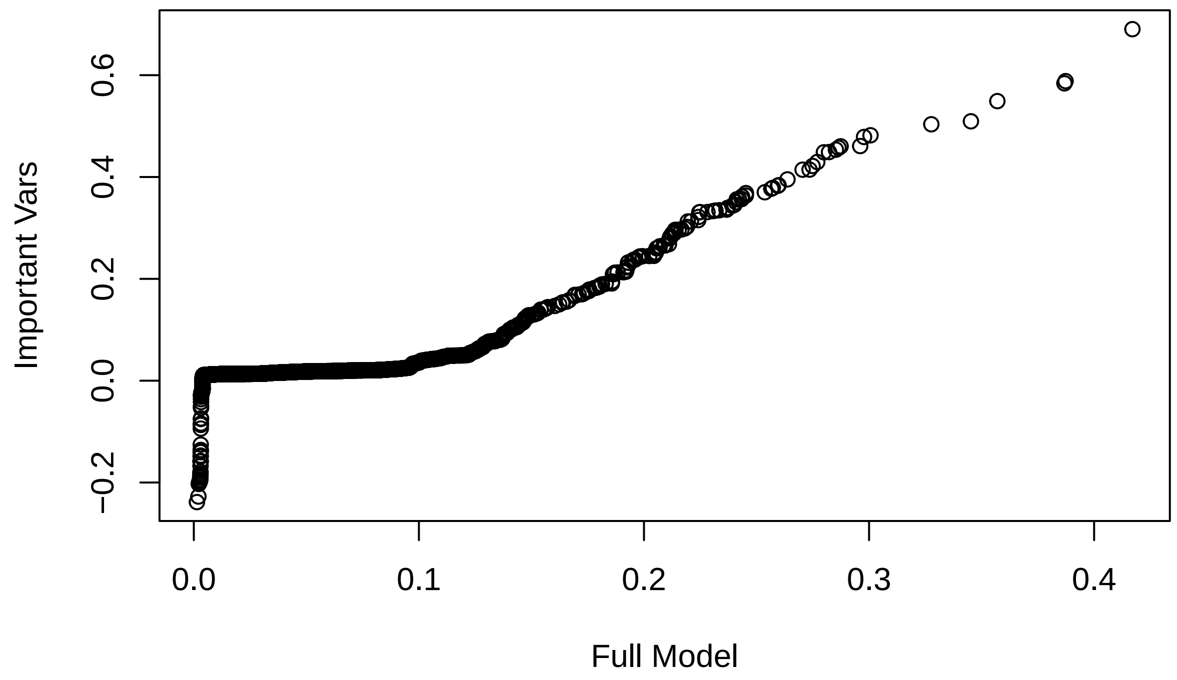
```
blabels = c("Full Model", "Important Variables Only")

# correlation matrix

cor(test$cate, test$cate2)
```

```
## [1] 0.4779716
```

```
#boxplot
plot1 <- boxplot(test$cate, test$cate2, names = blabels, ylab = "CATEs")
```



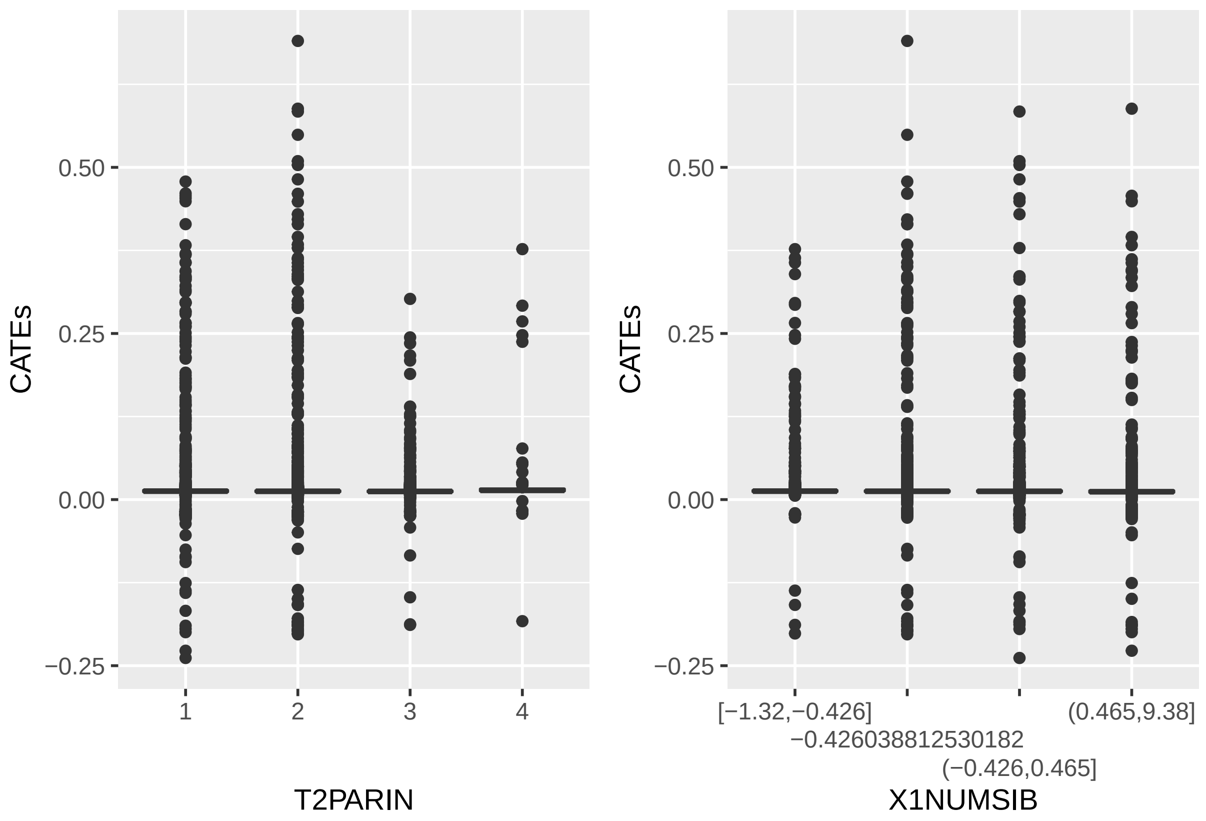
```
# qqplot
qplot <- qqplot(test$cate, test$cate2, xlab = "Full Model", ylab = "Important Vars")
```



Especially from the boxplot, it seems that the CATEs between models are not that different. From the correlation matrix, there is a ~~weak~~ moderate correlation noted.

Two most important CATE predictors (BART), plot CATE/predictor relationship

```
## top 2 predictors
top.pred <- c("T2PARIN", "X1NUMSIB")

cut <- quantcut(test$X1NUMSIB, q = 4, na.rm = T)

## plot relationship

bplot1 <- ggplot(test, aes(x = as.factor(T2PARIN), y = cate2)) +
  geom_boxplot() + labs(y = "CATEs", x = "T2PARIN")

bplot2 <- ggplot(test, aes(x = cut, y = cate2)) +
  geom_boxplot() + labs(y = "CATEs", x = "X1NUMSIB") +
  scale_x_discrete(guide = guide_axis(n.dodge = 3))

library(patchwork)
bplot1 + bplot2
```



From the previous estimations of variable importance, we found that T2PARIN and X1NUMSIB were the two most significant predictors (according to their relative inclusion proportion in the BART model). The two plots demonstrate a similar lack of heterogeneity, as there is no identifiable relationship between the estimates and the two predictors.

note: the X1NUMSIB variable was previously standardized, that's why the plots are divided into not the most informative quartiles.

GenericML

Q1)

Initially, we set up the hyper-parameters for GenericML. We decided to use the learners ranger (with 300 trees, for a balance of efficiency and effectiveness), and lasso with the default settings as a comparison.

As GenericML is set up to be used on experimental data and therefore does not accept propensity scores outside of 0.05 and 0.95, propensity scores were calculated and then rounded at the extreme ends to avoid this issue.

The other primary hyperparameter we were concerned with was the num_splits argument in the GenericML function (how many times the data is split for recalculating), which was set to 10. This again was chosen for a balance efficiency and effectiveness.

```
# Setup for GML
learners <- c("mlr3::lrn('ranger', num.trees = 300)", "lasso")
matrix_covs <- as.matrix(data %>% select(all_of(covariateNames)) %>%
                           mutate(across(.fns = as.numeric)))
X1 <- setup_X1(funs_Z = c("B", "S"))
vcov <- setup_vcov(estimator = "vcovHC")

# Estimate ps scores (with 0.05/0.95 adjustment to work with GenML)
library(parsnip)
ps_rf <- rand_forest(mode = "classification",
               engine = "ranger",
               trees = 1000) %>%
  fit(psFormula,
      data = data)
data$ps_rf <- predict(ps_rf,
                      new_data = data,
                      type = "prob")[,2]
data$ps_rf <- data$ps_rf$.pred_1 ## remove the $column.name
data <- data %>%
  mutate(ps_rf = ifelse(ps_rf >= 0.95, 0.94, ps_rf), #Rounding to avoid error Dr. L
         ps_rf = ifelse(ps_rf <= 0.05, 0.06, ps_rf))

# Run initial GenML
genML <- GenericML(
  Z = matrix_covs, #covariates
  D = as.numeric(as.character(data$treated)), #treatment
  Y = as.numeric(data$X2MTHETK1), #outcome
  learners_GenericML = learners,  # learners specified above
  learner_propensity_score = as.numeric(data$ps_rf), #as.numeric(data$ps)  #ps
  num_splits = 10,                        # number splits of the data
  quantile_cutoffs = c(0.2, 0.4, 0.6, 0.8), # grouping for CATEs
  significance_level = 0.05,                # significance level
  X1_BLP = X1, X1_GATES = X1,               # regression setup
  vcov_BLP = vcov, vcov_GATES = vcov,       # covariance setup
  parallel = F, #num_cores = 6L, # parallelization
  seed = 20220621)                         # RNG seed
```

As the below printouts show, the best ranger performed more effectively than lasso, and therefore, the results calculated with ranger become the default for the rest of the analysis

```
get_best(genML)
```

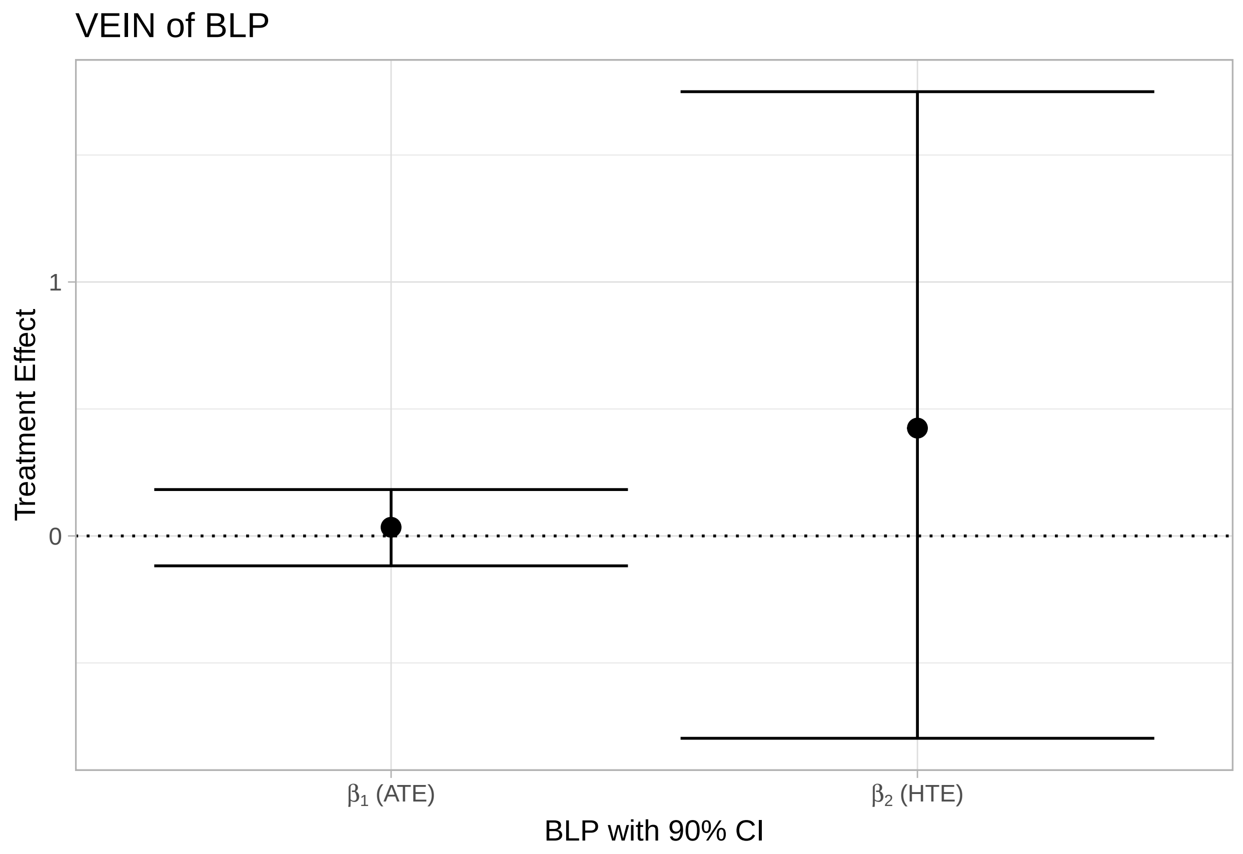
```
##                                          lambda lambda.bar
## mlr3::lrn('ranger', num.trees = 300) 0.003275   0.008701
## lasso                                0.001050   0.007221
## ---
## The best learner for BLP is mlr3::lrn('ranger', num.trees = 300) with lambda = 0.0033.
## The best learner for GATES and CLAN is mlr3::lrn('ranger', num.trees = 300) with lambda.bar = 0.0087
```

```
## ranger is best, becomes the default for all future GenML functions
```

Unlike when initially calculating heterogeneity on attendance, GenML does not find evidence of significant heterogeneity when looking at spring math scores. For the purpose of the assignment however, we continue analyzing as if there was heterogeneity to be explored.

```
get_BLP(genML)
```

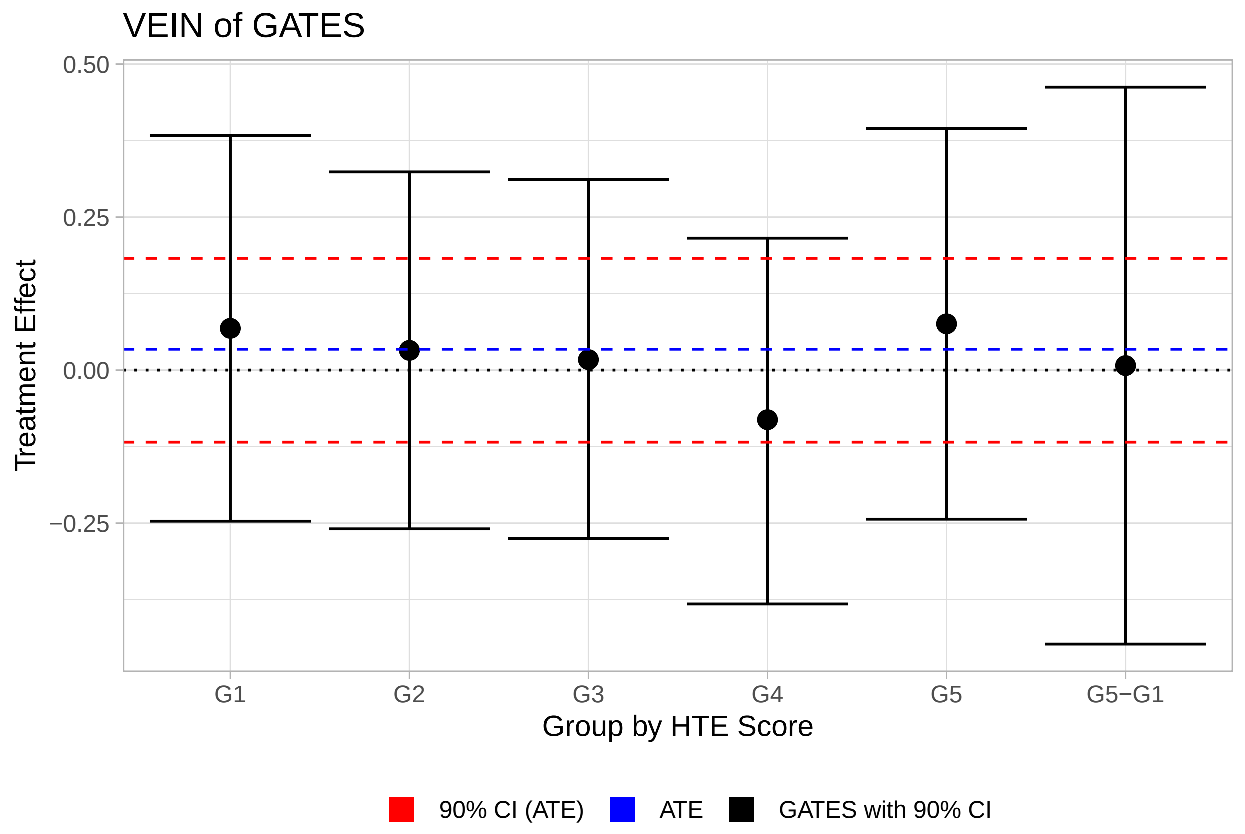
```
## BLP generic targets
## ---
##        Estimate CI lower CI upper p value
## beta.1  0.03413 -0.11769  0.18266   0.630
## beta.2  0.42461 -0.79681  1.74942   0.426
## ---
## Confidence level of confidence interval [CI lower, CI upper]: 90 %
```



```
## (no longer) significant indicating treatment heterogeneity
```

```
get_GATES(genML)
```

```
## GATES generic targets
## ---
##                  Estimate  CI lower  CI upper p value
## gamma.1          0.068156 -0.246833  0.383144   0.649
## gamma.2          0.032178 -0.259401  0.323758   0.714
## gamma.3          0.017048 -0.274932  0.311441   0.903
## gamma.4         -0.081083 -0.382124  0.215560   0.594
## gamma.5          0.075532 -0.243578  0.394642   0.491
## gamma.5-gamma.1  0.007318 -0.447657  0.462292   0.888
## ---
## Confidence level of confidence interval [CI lower, CI upper]: 90 %
```



The below code predicts CATEs with all covariates for use in Q2 at the end of the paper.

```
# Predict CATEs with full dataset for plotting/comparison
genML_Q2 <- proxy_CATE(Z = matrix_covs,
                       D = as.numeric(as.character(data$treated)),
                       Y = as.numeric(data$X2MTHETK1),
                       A_set = sample(1:12684, size = 12684/2), #obs sample half
                       learner = "mlr3::lrn('ranger', num.trees = 300)")
data$GenML_CATEa <- genML_Q2$estimates$CATE
# proxy_CATE builds model with half but then provides estimates for all, so need to
# cut down to half sample to compare with other methods later. No better method
# appears to be available
GenML_CATEhalf <- sample(data$GenML_CATEa, 6342)
```

Q3)

The next stage for GenericML took a workaround to best imitate the variable importance or BartMachine functions of the other two methods. Using heterogeneity_CLAN() we were able to get the p-values for each variable in terms of covariates in terms of their influence on heterogeneity.

```r
genML_het <- heterogeneity_CLAN(genML)

genML_sig <- as.data.frame(genML_het$p_values) %>%
  pivot_longer(cols = everything()) %>%
  arrange(value)

# Selecting variables above median significance
genML_imps <- genML_sig %>%
  filter(value < median(value)) %>%
  select(name) %>%
  as.list()
genML_imps <- genML_imps[["name"]]
```

Then, a new covariate matrix was created with only variables whose p-value was less than the median of all. As there is no directly comparable function to variable importance in causal forests and bart machine for causal bart, this was our best approximation of a similar concept.

We then recalculated CATEs with only these reduced covariates.

```r
# Create reduced covariates matrix
matrix_covsGML2 <- as.matrix(data %>% select(all_of(genML_imps)) %>%
                           mutate(across(.fns = as.numeric)))

# Calculate CATES using reduced covaraites
genML_Q3 <- proxy_CATE(Z = matrix_covsGML2,
                    D = as.numeric(as.character(data$treated)),
                    Y = as.numeric(data$X2MTHETK1),
                    A_set = sample(1:12684, size = 12684/2), #obs sample half
                    learner = "mlr3::lrn('ranger', num.trees = 300)")
data$GenML_CATEr <- genML_Q3$estimates$CATE
```

The resulting CATEs are summarized below. These results indicate that there is some variation, however, as noted above the omnibus test failed to find significant heterogeneity, so cannot assume this variation is anything other than noise. The correlation between the reduced and all variable CATE is 0.49 indicating a moderate relationship as demonstrated by the plot provided below as well.

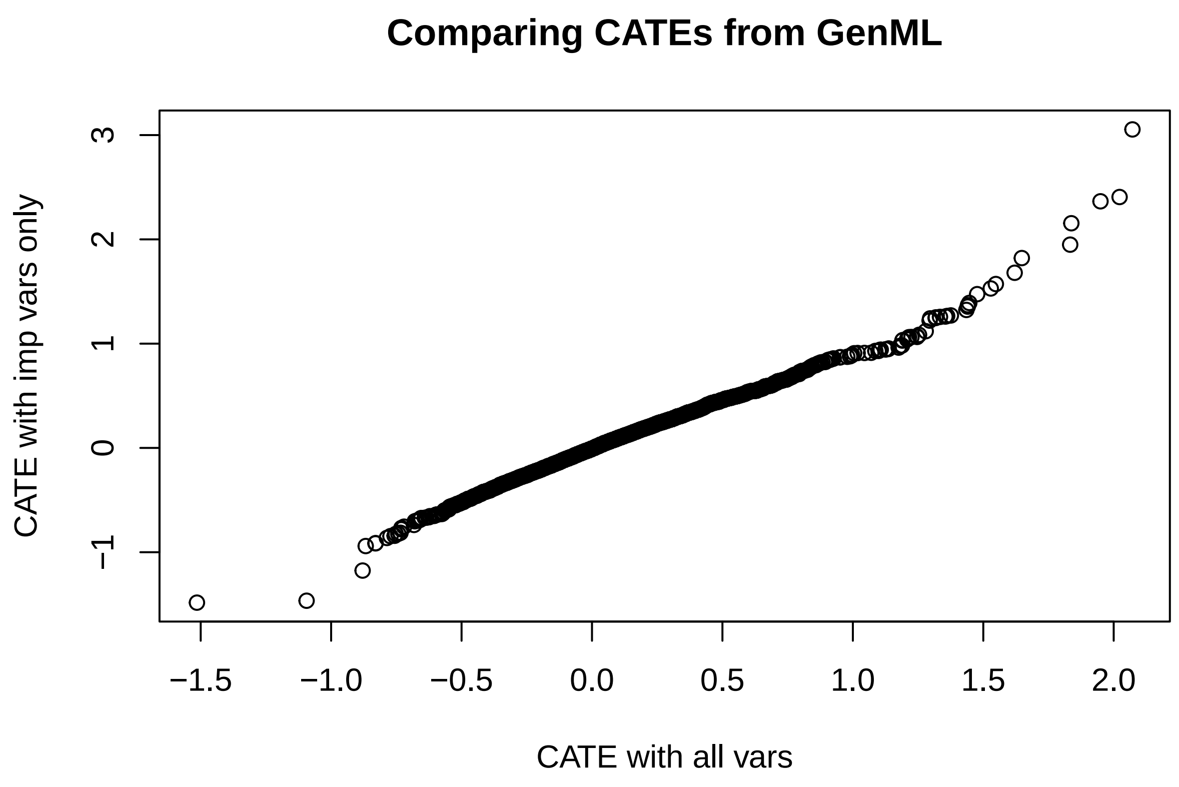
```r
summary(data$GenML_CATEr)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.48356 -0.08867  0.01050  0.02193  0.11897  3.05418
```

```r
cor(data$GenML_CATEa, data$GenML_CATEr)
```

```
## [1] 0.5037823
```

```r
qqplot(data$GenML_CATEa, data$GenML_CATEr,
     main = "Comparing CATEs from GenML",
     ylab = "CATE with imp vars only",
     xlab = "CATE with all vars")
```



Q4)

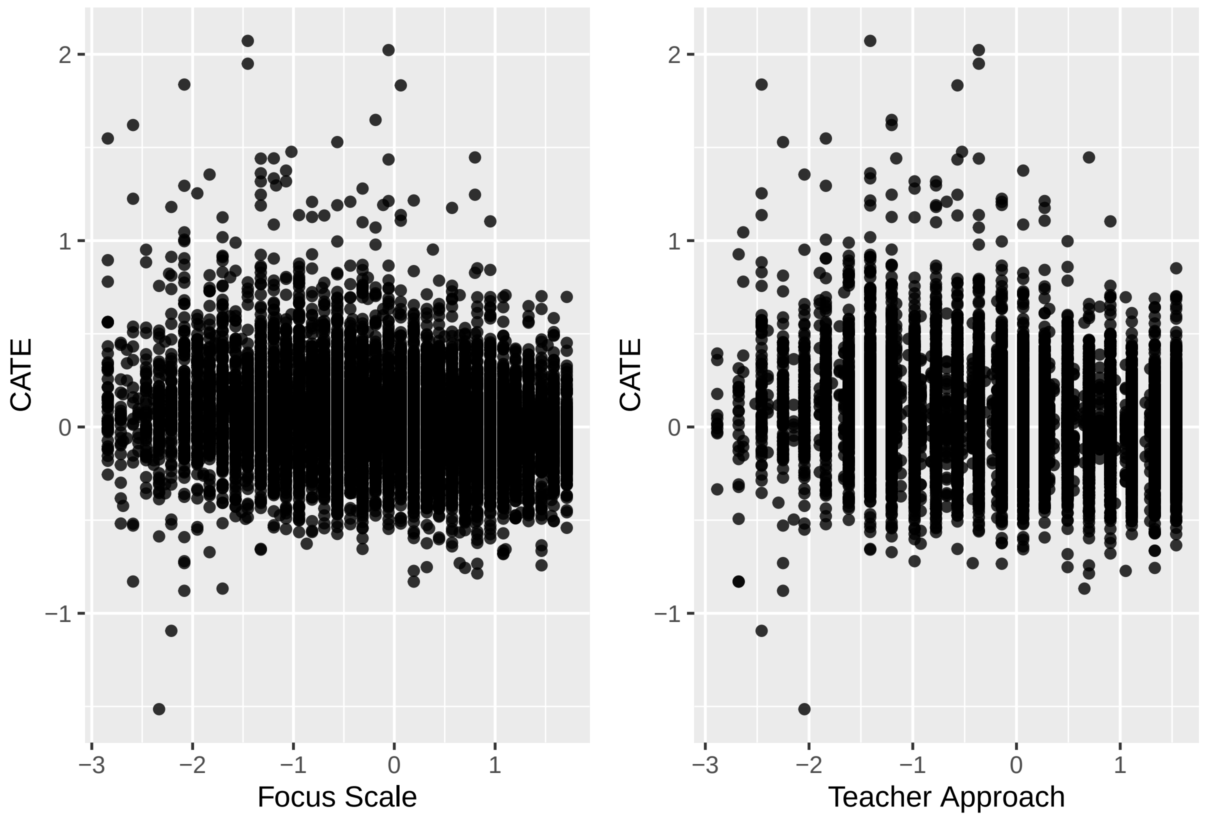
Lastly, we draw back on the calculation of individual predictors of treatment and pull out the 2 with the lowest p-values, X1ATTNFS (focus scale) and X1TCHAPP (teacher approaches to learning). Below are the plots of these variable against the CATE.

```
genML_sig %>% head(n = 2)
```

```
## # A tibble: 2 x 2
##   name          value
##   <chr>         <dbl>
## 1 X1MTHETK1 0
## 2 X1RTHETK1 3.28e-176
```

```
# Show the 2 lowest p values of het, i.e. most significant predictors of het

a <- ggplot(data) +
  geom_point(aes(x = X1ATTNFS,
                 y = GenML_CATEa),
             alpha = 0.8) +
  labs(y = "CATE",
       x = "Focus Scale")


b <- ggplot(data) +
  geom_point(aes(x = X1TCHAPP,
                 y = GenML_CATEa),
             alpha = 0.8) +
  labs(y = "CATE",
       x = "Teacher Approach")
```

```
library(patchwork)
a + b
```



Even though these variables were identified as the mostly likely contributors to treatment heterogeneity, the 2 plots show quite there is no clear relationship to be seen between them and the CATE, supporting the finding that GenericML suggests there is no treatment heterogeneity.

Causal Forests

Q1)

```r
data2 <- data

#fit logistic regression model for propensity score estimation ignoring clustering
ps.model0 <- glm(psFormula, data=data2, family=binomial)

#obtain propensity scores that ignore clustering
data2$ps <- fitted(ps.model0)

#the grf package only takes numeric covariates
#So convert those factor variables to be the numeric class
for (i in 1:length(covariateNames)) {
  if(class(data2[,covariateNames[i]])=="factor"){
    data2[, covariateNames[i]] <- as.numeric(as.character(data2[,covariateNames[i]]))
  }
}

#Step 1: Split data into training data set and testing data set
#In this case, we split it to be 50/50
set.seed(123)
train_index <- sample(1:nrow(data2), nrow(data2)/2)
train_index <- train_index[order(train_index)]

train_data <- data2[train_index,]
test_data <- data2[-train_index,]

#Step 2: model fit, using causal forest
#Tuning mtry and min.node.size parameters by setting tune.parameters
train.forest = causal_forest(X=train_data[,covariateNames],
                             Y = train_data$X2MTHETK1, num.trees = 5000,
                             W = as.numeric(as.character(train_data$treated)),
                             W.hat = train_data$ps,
                             tune.parameters = c("mtry", "min.node.size"),
                             seed = 0)

#The best tunning parameters of mtry and min.node.size were shown below,
 #which indicated a better performance than the default setting
train.forest[["tuning.output"]]
```

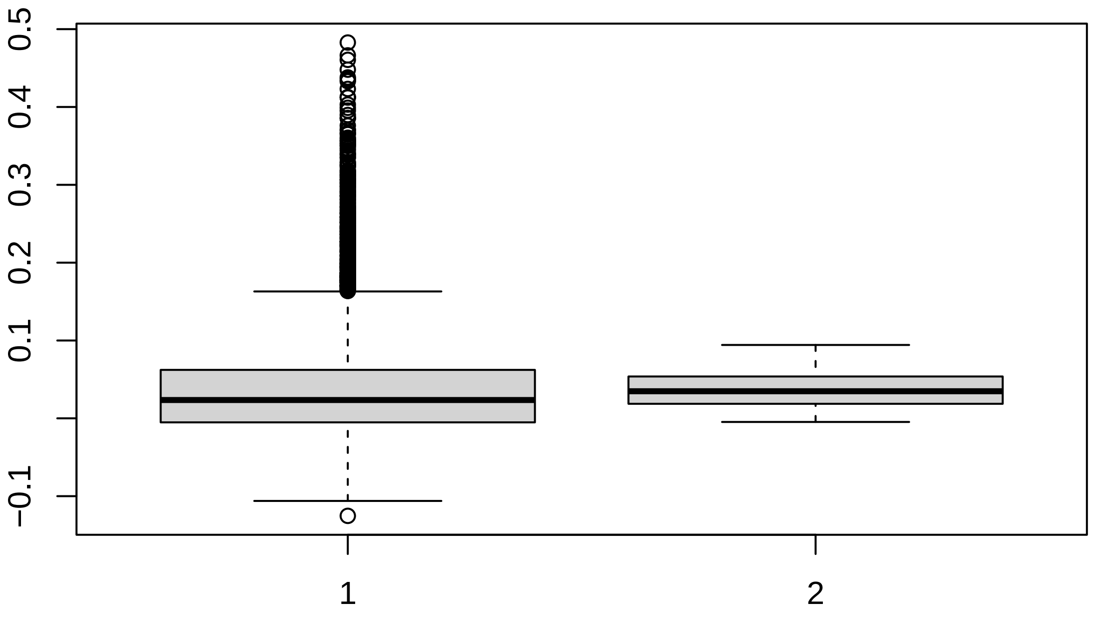
```
## Tuning status: tuned.
## This indicates tuning found parameters that are expected to perform better than default.
##
## Predicted debiased error: 0.190954597186443
##
## Tuned parameters:
## mtry: 16
##  min.node.size: 1
##
## Average error by 5-quantile:
##
##     mtry     error
##    [1,6] 0.1927644
##   (6,11] 0.1926376
##  (11,16] 0.1926894
##  (16,21] 0.1926488
##  (21,26] 0.1924563
##
##  min.node.size     error
##          [1,2] 0.1911759
##          (2,9] 0.1918496
##       (9,32.4] 0.1926654
##     (32.4,124] 0.1935464
##      (124,393] 0.1940741
```

Q2) Correlation and Figures Comparing Different CF fits

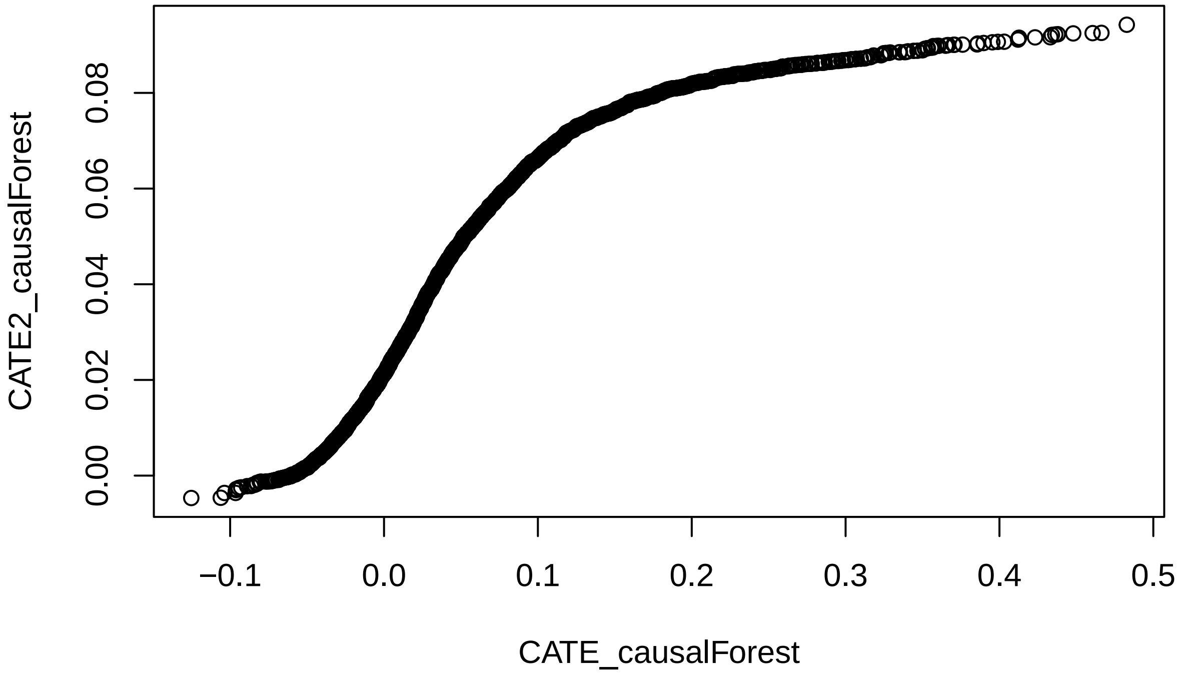
```
#Obtain estimates of the conditional average treatment effect (CATE)
#with standard errors
tau.hat = predict(train.forest,X= test_data[,covariateNames], estimate.variance = T)
CATE_causalForest = tau.hat$predictions


# Causal Forests
#1. correlation matrix
#causal forest only output the best tunning parameters' model fit outcomes
#To answer Q2, I run one more model fit with mtry = 4 and min.node.size = 50
train.forest2 = causal_forest(X=train_data[,covariateNames],
                              Y = train_data$X2MTHETK1, num.trees = 5000,
                              W = as.numeric(as.character(train_data$treated)),
                              W.hat = train_data$ps,
                              mtry = 4, min.node.size = 50,
                              seed = 0)
tau.hat2 = predict(train.forest2,X= test_data[,covariateNames], estimate.variance = T)
CATE2_causalForest = tau.hat2$predictions

cor(CATE_causalForest, CATE2_causalForest)
```

```
## [1] 0.7619943
```

```
#2.box plot
boxplot(CATE_causalForest, CATE2_causalForest)
```



```r
#3.QQ plot
qqplot(CATE_causalForest, CATE2_causalForest)
```



The correlation between the two tunning methods was strong. But the distributions of CATEs between these two tunning methods were slightly different (see box plot), which caused a nonlinear QQ plot.

Q3) Determine Best Linear Projection of CATE/Variable Importance

```r
#Causal Forests
#Step 1: Subset important variables
importance_cf = variable_importance(train.forest)
rownames(importance_cf) = names(train_data[,covariateNames])
```

```
#select variables above the median of importance of the aggregated importances
#across imputed datasets
important.var_cf = rownames(importance_cf)[importance_cf>median(importance_cf)]

#Step 2
#run test forest with the best hyperparameters
test.forest = causal_forest(X = test_data[,important.var_cf],
                            Y = test_data$X2MTHETK1,
                            W = as.numeric(as.character(test_data$treated)),
                            W.hat = test_data$ps,
                            mtry = 16, num.trees=5000,
                            min.node.size = 1, seed = 0)

#Step 3: Estimate the best linear projection of CATE
predictors = test_data[,important.var_cf]

CATE.prediction = best_linear_projection(test.forest, A=predictors)
CATE.prediction
```

```
##
## Best linear projection of the conditional average treatment effect.
## Confidence intervals are cluster- and heteroskedasticity-robust (HC3):
##
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1653968  0.1279545  1.2926  0.19619
## X1RTHETK1     0.0515876  0.0529552  0.9742  0.33001
## X1MTHETK1    -0.0906360  0.1641318 -0.5522  0.58082
## X1TCHAPP      0.0134252  0.0419533  0.3200  0.74898
## X1TCHPER     -0.0333595  0.0301400 -1.1068  0.26841
## X1TCHEXT      0.0700968  0.0352924  1.9862  0.04706 *
## X1TCHINT     -0.0188982  0.0344273 -0.5489  0.58307
## X1ATTNFS     -0.0742941  0.0401170 -1.8519  0.06408 .
## X1INBCNT      0.1045508  0.0677221  1.5438  0.12268
## P1OLDMOM      0.0410309  0.0239838  1.7108  0.08717 .
## P1CHLDBK      0.0029568  0.0124860  0.2368  0.81281
## X2INCCAT_I    0.0189513  0.0437704  0.4330  0.66505
## S2LUNCH       0.0564428  0.0370794  1.5222  0.12801
## X2KRCETH     -0.0039526  0.0418722 -0.0944  0.92480
## S2OUTSID     -0.0812555  0.0540147 -1.5043  0.13255
## S1_ID        -0.0548927  0.0476948 -1.1509  0.24981
## prop.missing  0.0055268  0.0178410  0.3098  0.75674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#The predictor has a p-value smaller than 0.05,
 # which indicates the existence of heterogeneity.

#Further check the herterogeneity
test_calibration(test.forest)
```

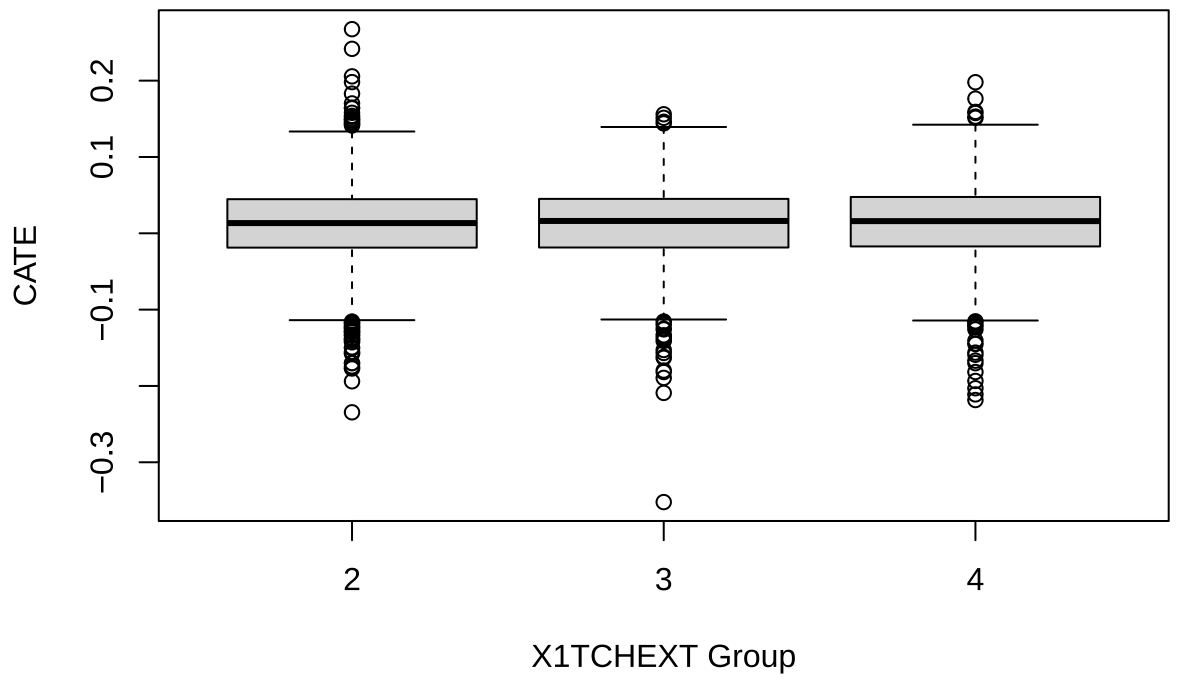
```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
```

```
##                                Estimate Std. Error t value Pr(>t)
## mean.forest.prediction          1.17996    1.27144  0.9280 0.1767
## differential.forest.prediction  0.20666    0.35460  0.5828 0.2800
```

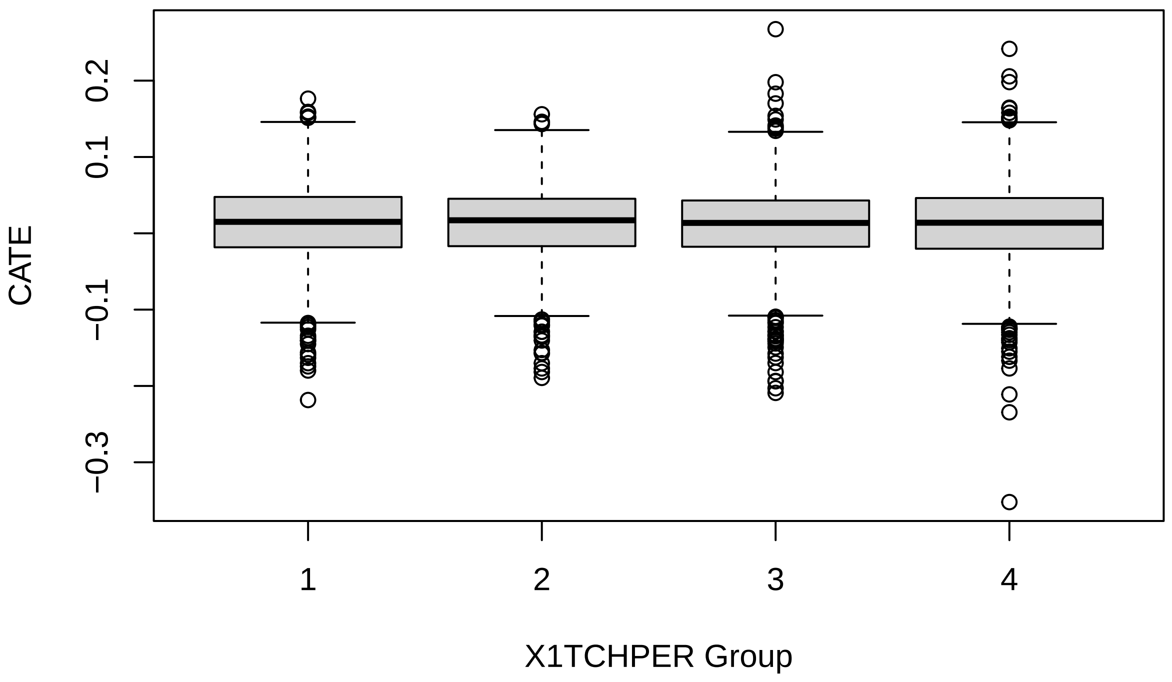
```
#The outcomes showed that the coefficient of the mean forest prediction was 1
 #which indicated the mean forest prediction was correct.

#Also, the results indicated that no heterogeneity
 #had been detected in the overall selected covariates.
```

Q4) Two most important CATE predictors (Causal Forests), plot CATE/predictor relationship

```
#The most two predictors are
Top2predictors <- c("X1TCHEXT", "X1TCHPER")

#Obtain estimates of the conditional average treatment effect (CATE)
#with standard errors
tau.hat = predict(test.forest,X= test_data[,important.var_cf],
estimate.variance = T)
CATE_causalForest = tau.hat$predictions


#Predictor 1
group1 <- quantile(train_data[, Top2predictors[1]])
train_data$groupX1M <- ifelse(train_data[,Top2predictors[1]] >= group1[4], 4,
                             ifelse(train_data[,Top2predictors[1]]>=group1[3] &
                             train_data[,Top2predictors[1]]<group1[4], 3,
                                    ifelse(train_data[,Top2predictors[1]]>=group1[2] &
                                    train_data[,Top2predictors[1]]<group1[3], 2,
                                           1)))
train_data$groupX1M <- factor(train_data$groupX1M)
boxplot(CATE_causalForest ~ train_data$groupX1M,
xlab = paste(Top2predictors[1], "Group"), ylab = "CATE")
```



```r
#Predictor 2
group2 <- quantile(train_data[, Top2predictors[2]])
train_data$group2 <- ifelse(train_data[,Top2predictors[2]] >= group2[4], 4,
                            ifelse(train_data[,Top2predictors[2]]>=group2[3] &
                            train_data[,Top2predictors[2]]<group2[4], 3,
                                   ifelse(train_data[,Top2predictors[2]]>=group2[2] &
                                   train_data[,Top2predictors[2]]<group2[3], 2,
                                          1)))
train_data$group2 <- factor(train_data$group2)
boxplot(CATE_causalForest ~ train_data$group2,
xlab = paste(Top2predictors[2], "Group"), ylab = "CATE")
```



Both two plots for two important predictors did not show an obvious different CATEs among groups.

Comparisons across Methods (Q2)

```r
labels = c("BART", "Causal Forests", "GenericML")

# correlation matrix

cor(cate_m, CATE_causalForest)
```

```
## [1] -0.1196529
```

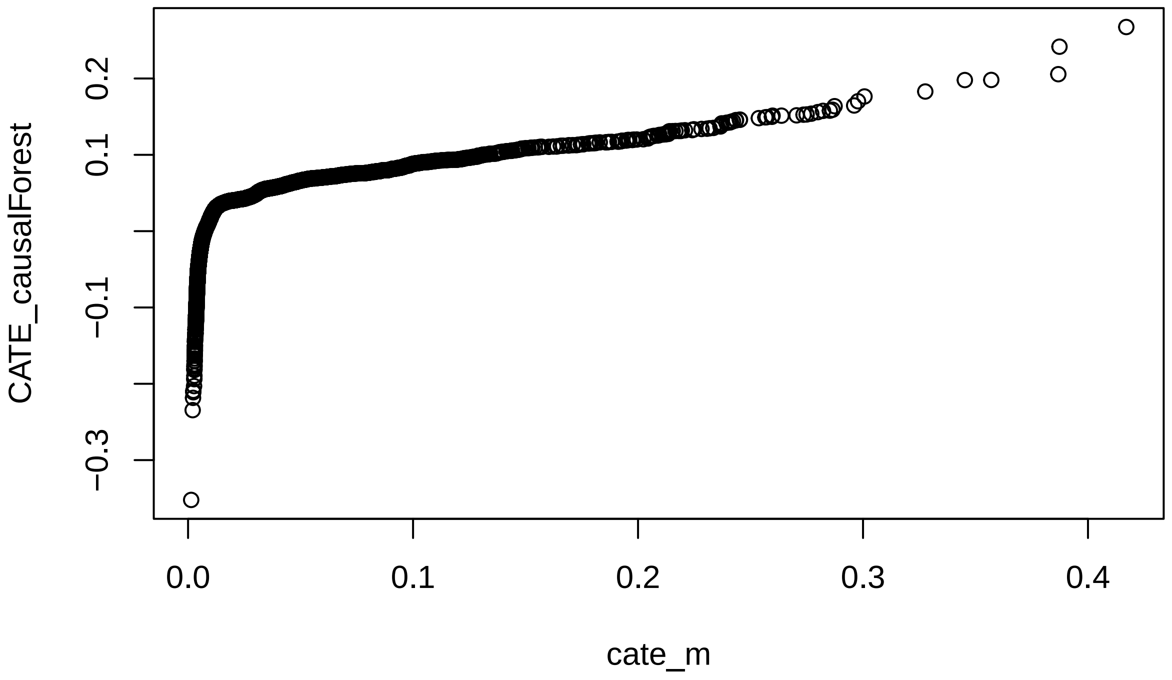
```r
cor(cate_m, GenML_CATEhalf)
```

```
## [1] 0.02273236
```

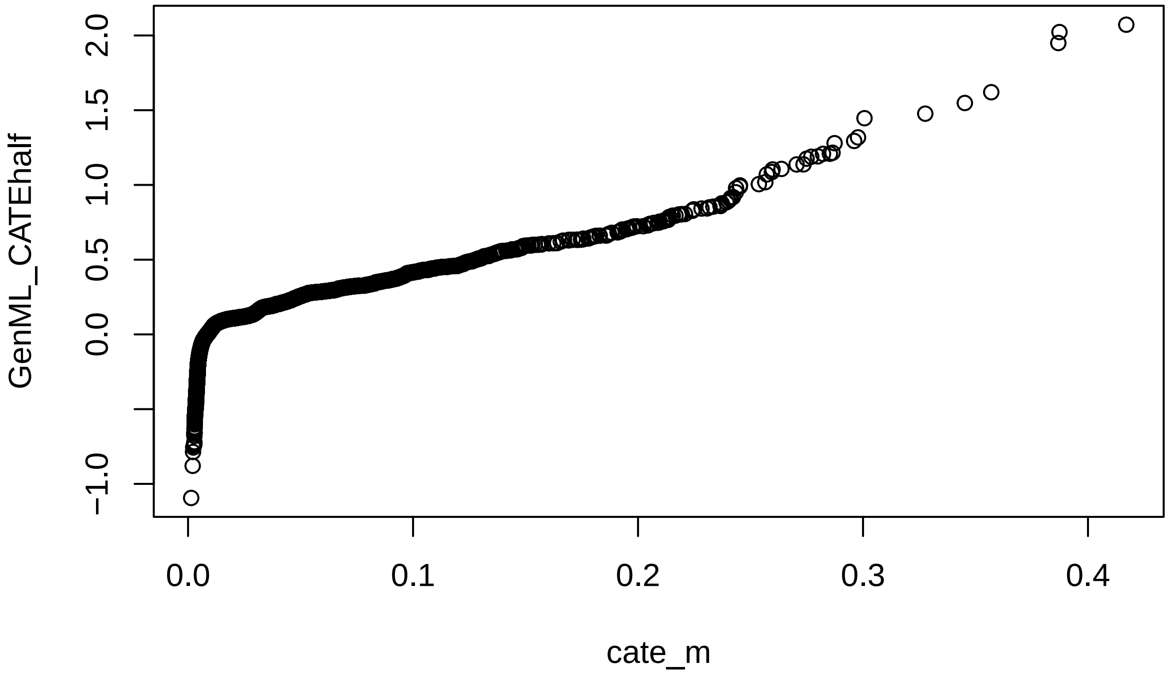
```r
cor(CATE_causalForest, GenML_CATEhalf)
```

```
## [1] -0.002439439
```

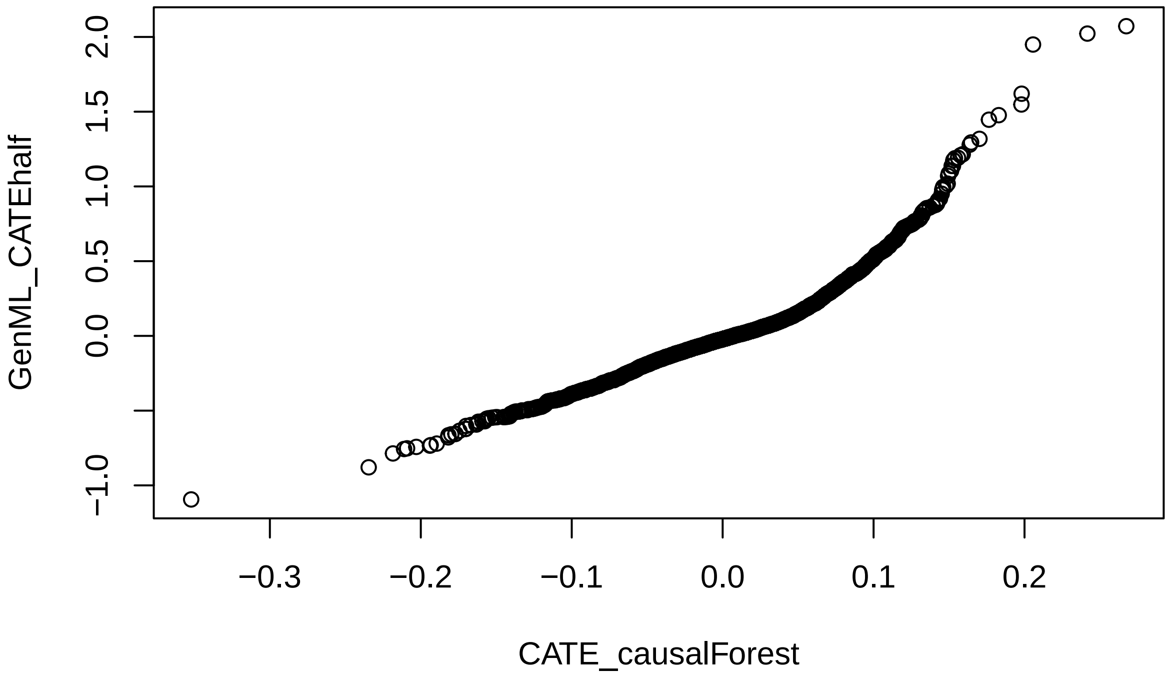
```r
# qqplot

qqplot(cate_m, CATE_causalForest)
```



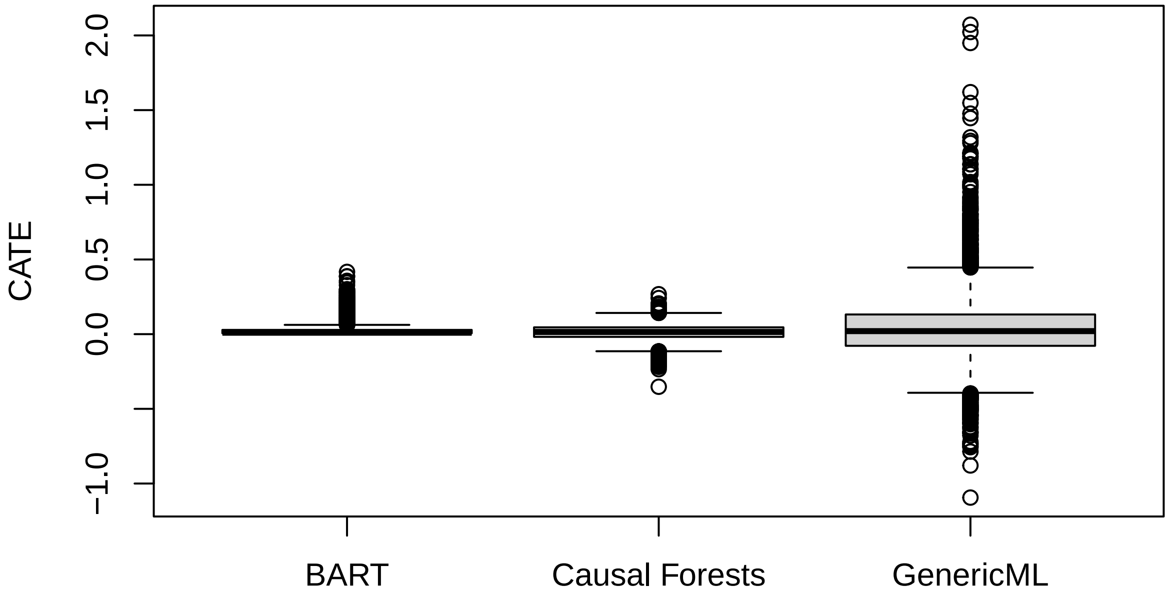
```r
qqplot(cate_m, GenML_CATEhalf)
```



```
qqplot(CATE_causalForest, GenML_CATEhalf)
```



```
# boxplot

boxplot(cate_m, CATE_causalForest, GenML_CATEhalf,
                        names = labels, ylab = "CATE")
```



Method Comparisons and Conclusion

BART detected no treatment effect heterogeneity, as evidenced by the CATE parameter estimate of almost 0. The CATE estimates between the full model with all covariates and the important variables model were slightly correlated. The two most important predictors identified were not related to our CATE estimates, further evidence of lack of treatment heterogeneity.

GenericML failed to find any evidence of treatment heterogeneity on the math score outcome variable. The p-value of the omnibus test for heterogeneity was 0.41, so not even close to marginal significance. The CATEs calculated with all vars and only important vars were loosely similar to each other. The two most important variables still showed no real evidence of treatment heterogeneity.

Causal Forest (CF) did not fit very well with the data set. Because when using the training data set, CF with the best tuning parameters detected heterogeneity, but it failed to detect heterogeneity by using the testing data set. When plotting the top two important predictors with CATE, we found no difference between groups.

Comparing CATEs estimated between methods we found there was surprisingly little correlation between any of the methods. Although they all predicted a CATE average close to 0, BART and Causal Forest had much narrower spreads than GenericML. This may have been in part due to the more awkward nature of pulling the CATE out GenericML, as it is not available from the primary GenericML object and needs a separate calculation.

In summary, this paper attempted to explore treatment effect heterogeneity with Causal Forests, GenericML, and Causal BART. Unfortunately, there was no significant heterogeneity to be found in terms of math score outcome. All three methods came to similar conclusions.