Econ 293 Final Project Analysis Report

Chris Lin, Joseph Son
(clin17@stanford.edu, joeson@stanford.edu)

2019-06-06

```r
# Set up data and global variables
source("helpers.R")
source("load-data.R")
data_01 <- split_control_treated(0, 1, df, covariate_names)
data_02 <- split_control_treated(0, 2, df, covariate_names)
data_03 <- split_control_treated(0, 3, df, covariate_names)
NUM_TREES <- 1000
```

```r
# Estimate HTEs using forest-based learners
if (!file.exists("results/results_forests.rds")) {
  set.seed(123)
  results_forests_01 <- compare_forests(data_01$X, data_01$W, data_01$Y, data_01$ID, NUM_TREES)
  results_forests_01$compare <- "0 vs 1"

  results_forests_02 <- compare_forests(data_02$X, data_02$W, data_02$Y, data_02$ID, NUM_TREES)
  results_forests_02$compare <- "0 vs 2"

  results_forests_03 <- compare_forests(data_03$X, data_03$W, data_03$Y, data_03$ID, NUM_TREES)
  results_forests_03$compare <- "0 vs 3"

  results_forests <- Reduce(rbind, list(results_forests_01,
                                        results_forests_02,
                                        results_forests_03))
  saveRDS(results_forests, file = "results/results_forests.rds")
} else {
  results_forests <- readRDS(file = "results/results_forests.rds")
}
```

```r
# Estimate HTEs using ridge-based learners
if (!file.exists("results/results_glmnets.rds")) {
  set.seed(1234)
  results_glmnets_01 <- compare_glmnets(data_01$X, data_01$W, data_01$Y, data_01$ID, 10, 0, NUM_TREES)
  results_glmnets_01$compare <- "0 vs 1"

  results_glmnets_02 <- compare_glmnets(data_02$X, data_02$W, data_02$Y, data_02$ID, 10, 0, NUM_TREES)
  results_glmnets_02$compare <- "0 vs 2"

  results_glmnets_03 <- compare_glmnets(data_03$X, data_03$W, data_03$Y, data_03$ID, 10, 0, NUM_TREES)
  results_glmnets_03$compare <- "0 vs 3"

  results_glmnets <- Reduce(rbind, list(results_glmnets_01,
                                        results_glmnets_02,
                                        results_glmnets_03))
  saveRDS(results_glmnets, file = "results/results_glmnets.rds")
} else {
  results_glmnets <- readRDS(file = "results/results_glmnets.rds")
```

```r
}
```

```r
# Combine all the results
hte_results <- rbind(results_forests, results_glmnets)

# Summarize the learner results
hte_summary_01 <- summarize_hte_results(hte_results, "0 vs 1")
hte_summary_02 <- summarize_hte_results(hte_results, "0 vs 2")
hte_summary_03 <- summarize_hte_results(hte_results, "0 vs 3")

# Print out the results
cat("0 vs 1", "\n")
```

```
## 0 vs 1
```

```r
hte_summary_01$MRL <- scientific(hte_summary_01$MRL)
hte_summary_01$SDRL <- scientific(hte_summary_01$SDRL)
print(hte_summary_01)
```

```
## # A tibble: 7 x 3
##   learner       MRL       SDRL
##   <fct>         <chr>     <chr>
## 1 S-forest      1.32e-04  1.64e-03
## 2 T-forest      1.36e-05  7.03e-04
## 3 X-forest      8.47e-06  1.99e-03
## 4 Causal forest 3.47e-05  1.80e-03
## 5 S-glmnet      1.56e-06  2.32e-04
## 6 T-glmnet      9.63e-07  2.19e-04
## 7 X-glmnet      -8.64e-07 1.37e-04
```

```r
cat("\n\n\n")
```

```r
cat("0 vs 2", "\n")
```

```
## 0 vs 2
```

```r
hte_summary_02$MRL <- scientific(hte_summary_02$MRL)
hte_summary_02$SDRL <- scientific(hte_summary_02$SDRL)
print(hte_summary_02)
```

```
## # A tibble: 7 x 3
##   learner       MRL      SDRL
##   <fct>         <chr>    <chr>
## 1 S-forest      1.03e-04 1.35e-03
## 2 T-forest      1.32e-05 7.31e-04
## 3 X-forest      4.97e-06 1.97e-03
## 4 Causal forest 3.04e-05 1.87e-03
## 5 S-glmnet      2.47e-06 4.04e-04
## 6 T-glmnet      8.62e-07 2.64e-04
## 7 X-glmnet      4.63e-07 1.44e-04
```

```r
cat("\n\n\n")
```

```r
cat("0 vs 3", "\n")
```

```
## 0 vs 3
```

```r
hte_summary_03$MRL <- scientific(hte_summary_03$MRL)
hte_summary_03$SDRL <- scientific(hte_summary_03$SDRL)
print(hte_summary_03)
```

```
## # A tibble: 7 x 3
##   learner       MRL       SDRL
##   <fct>         <chr>     <chr>
## 1 S-forest      9.28e-05  1.20e-03
## 2 T-forest      3.62e-06  7.51e-04
## 3 X-forest      -1.67e-05 2.20e-03
## 4 Causal forest 1.24e-05  2.08e-03
## 5 S-glmnet      2.13e-06  4.42e-04
## 6 T-glmnet      2.59e-07  2.79e-04
## 7 X-glmnet      4.51e-06  3.21e-04
```

```r
cat("\n\n\n")
```

```r
# Find the best learners
hte_best_01 <- hte_results %>%
  filter(compare == "0 vs 1") %>%
  filter(learner == hte_summary_01$learner[hte_summary_01$MRL == min(hte_summary_01$MRL)])
hte_best_learner_01 <- unique(hte_best_01$learner)
print(paste0("Best learner for 0 vs 1: ", hte_best_learner_01))
```

```
## [1] "Best learner for 0 vs 1: X-glmnet"
```

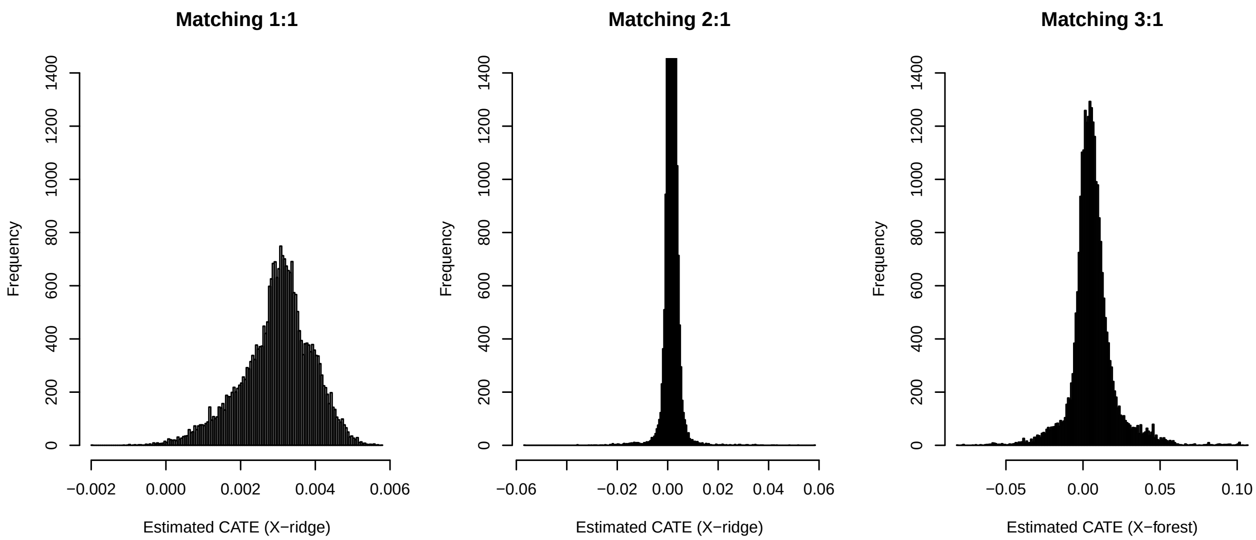
```r
hte_best_02 <- hte_results %>%
  filter(compare == "0 vs 2") %>%
  filter(learner == hte_summary_02$learner[hte_summary_02$MRL == min(hte_summary_02$MRL)])
hte_best_learner_02 <- unique(hte_best_02$learner)
print(paste0("Best learner for 0 vs 2: ", hte_best_learner_02))
```

```
## [1] "Best learner for 0 vs 2: S-forest"
```

```r
hte_best_03 <- hte_results %>%
  filter(compare == "0 vs 3") %>%
  filter(learner == hte_summary_03$learner[hte_summary_03$MRL == min(hte_summary_03$MRL)])
hte_best_leanrer_03 <- unique(hte_best_03$learner)
print(paste0("Best learner for 0 vs 3: ", hte_best_leanrer_03))
```

```
## [1] "Best learner for 0 vs 3: X-forest"
```

```r
par(mfrow = c(1, 3))
hist(hte_best_01$estimate, breaks = 250, ylim = c(0, 1400), xlab = "Estimated CATE (X-ridge)", main = "
hist(hte_best_02$estimate, breaks = 250, ylim = c(0, 1400), xlab = "Estimated CATE (X-ridge)", main = "
hist(hte_best_03$estimate, breaks = 250, ylim = c(0, 1400), xlab = "Estimated CATE (X-forest)", main =
```




```r
# Hypothesis test for overall treatment heterogeneity
if (!file.exists("results/test_hetero_summary_01.rds")) {
  test_hetero_01 <- test_hetero(as.matrix(hte_best_01[ , covariate_names]), hte_best_01$W,
                                hte_best_01$Y, hte_best_01$estimate, NUM_TREES)
  test_hetero_summary_01 <- summarize_test_hetero(test_hetero_01)
  saveRDS(test_hetero_summary_01, file = "results/test_hetero_summary_01.rds")
} else {
  test_hetero_summary_01 <- readRDS("results/test_hetero_summary_01.rds")
}


if (!file.exists("results/test_hetero_summary_02.rds")) {
  test_hetero_02 <- test_hetero(as.matrix(hte_best_02[ , covariate_names]), hte_best_02$W,
                                hte_best_02$Y, hte_best_02$estimate, NUM_TREES)
  test_hetero_summary_02 <- summarize_test_hetero(test_hetero_02)
  saveRDS(test_hetero_summary_02, file = "results/test_hetero_summary_02.rds")
} else {
  test_hetero_summary_02 <- readRDS("results/test_hetero_summary_02.rds")
}


if (!file.exists("results/test_hetero_summary_03.rds")) {
  test_hetero_03 <- test_hetero(as.matrix(hte_best_03[ , covariate_names]), hte_best_03$W,
                                hte_best_03$Y, hte_best_03$estimate, NUM_TREES)
  test_hetero_summary_03 <- summarize_test_hetero(test_hetero_03)
  saveRDS(test_hetero_summary_03, file = "results/test_hetero_summary_03.rds")
} else {
  test_hetero_summary_03 <- readRDS("results/test_hetero_summary_03.rds")
}

cat("Test for overall heterogeneity", "\n")
```

```
## Test for overall heterogeneity
```

```r
cat("0 vs 1", "\n")
```

```
## 0 vs 1
```

```r
test_hetero_summary_01$onesided_pval <- min(c(1, test_hetero_summary_01$onesided_pval * 3))
print(test_hetero_summary_01)
```

```
## $beta
## [1] 2.602721
##
## $beta_se
## [1] 1.822936
##
## $onesided_pval
## [1] 0.2300578
```

```r
cat("\n\n")
```

```r
cat("0 vs 2", "\n")
```

```
## 0 vs 2
```

```r
test_hetero_summary_02$onesided_pval <- min(c(1, test_hetero_summary_02$onesided_pval * 3))
print(test_hetero_summary_02)
```

```
## $beta
## [1] -0.4299797
##
## $beta_se
## [1] 1.818228
##
## $onesided_pval
## [1] 1
```

```r
cat("\n\n")
```

```r
cat("0 vs 3", "\n")
```

```
## 0 vs 3
```

```r
test_hetero_summary_03$onesided_pval <- min(c(1, test_hetero_summary_03$onesided_pval * 3))
print(test_hetero_summary_03)
```

```
## $beta
## [1] 0.6757758
##
## $beta_se
## [1] 0.1246052
##
## $onesided_pval
## [1] 8.852597e-08
```

```r
cat("\n\n")
```

```r
# Fit the models and get results
donate_cost_ratio <- c(10, 100, 1000, 10000)
costs <- 1 / donate_cost_ratio
if (!file.exists("results/policy_results_03.rds")) {
  set.seed(123)
  policy_results_03 <- list()
  for (i in 1:length(costs)) {
    policy_results_03[[i]] <- opt_policy_forest(data_03$X, data_03$W, data_03$Y, hte_best_03$estimate, c
```

```r
                                                   cost = costs[i], k = 10, num_trees = NUM_TREES)
  }
  saveRDS(policy_results_03, file = "results/policy_results_03.rds")
} else {
  policy_results_03 <- readRDS("results/policy_results_03.rds")
}

# Result summaries
policy_summary_dfs <- list()
for (i in 1:length(costs)) {
  policy_summary_dfs[[i]] <- summarize_opt_policy(policy_results_03[[i]]$Ahats_test)
  print(paste0("Donation:cost ratio = ", donate_cost_ratio[i]))
  print(policy_summary_dfs[[i]])
}
```

```
## [1] "Donation:cost ratio = 10"
##   Ahat_policy_mean Ahat_policy_sd Ahat_plugin_mean Ahat_plugin_sd
## 1       0.09381904    0.002706898       0.09382653    0.002713448
##   Ahat_everyone_mean Ahat_everyone_sd
## 1        -0.09381904      0.002706898
## [1] "Donation:cost ratio = 100"
##   Ahat_policy_mean Ahat_policy_sd Ahat_plugin_mean Ahat_plugin_sd
## 1       0.00401501    0.004897653        0.0104208     0.00399413
##   Ahat_everyone_mean Ahat_everyone_sd
## 1       -0.003815493      0.005984501
## [1] "Donation:cost ratio = 1000"
##   Ahat_policy_mean Ahat_policy_sd Ahat_plugin_mean Ahat_plugin_sd
## 1      0.004703359    0.004493456      0.009873607    0.005088723
##   Ahat_everyone_mean Ahat_everyone_sd
## 1        0.005160319      0.003941542
## [1] "Donation:cost ratio = 10000"
##   Ahat_policy_mean Ahat_policy_sd Ahat_plugin_mean Ahat_plugin_sd
## 1      0.005841853    0.002113163      0.009968833     0.00231166
##   Ahat_everyone_mean Ahat_everyone_sd
## 1        0.006024226      0.002224656
```

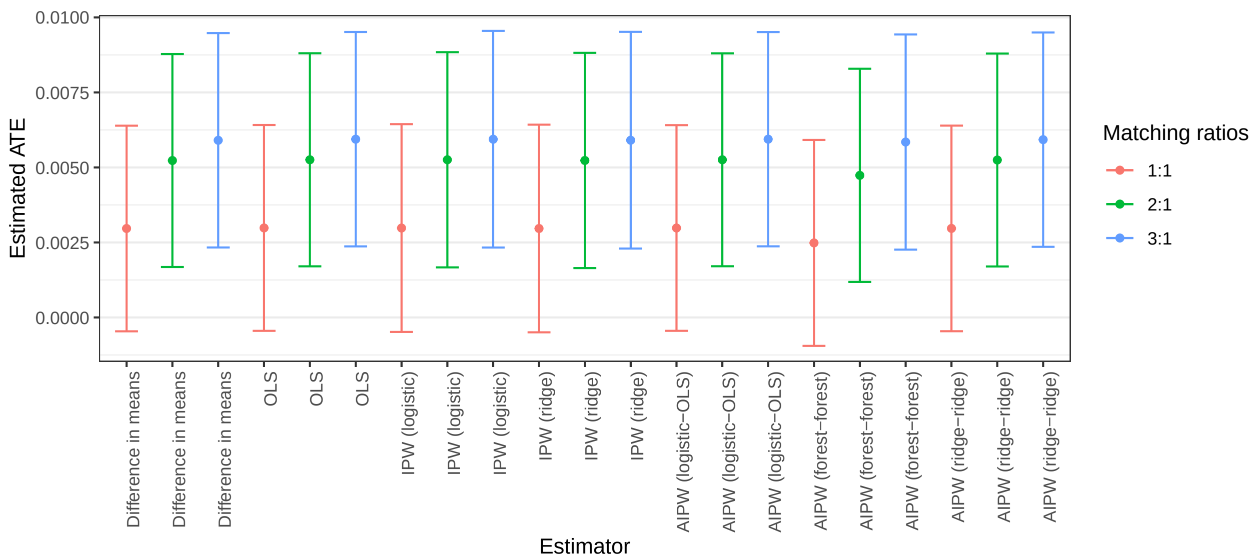
```r
df1 <- data_01$df %>% select(-ID)
df2 <- data_02$df %>% select(-ID)
df3 <- data_03$df %>% select(-ID)

if (!file.exists("results/ate_results_table.rds")) {
  ate_results <- calculate_ATE(df1, df2, df3, 1)
  saveRDS(ate_results, file = "results/ate_results_table.rds")
} else {
  ate_results <- readRDS("results/ate_results_table.rds")
}

if (!file.exists("results/ate_results_reduced_table.rds")) {
  ate_results_reduced <- calculate_ATE(df1, df2, df3, 0.5)
  saveRDS(ate_results_reduced, file = "results/ate_results_reduced_table.rds")
} else {
  ate_results_reduced <- readRDS("results/ate_results_reduced_table.rds")
}
print(ate_results)
```

```
## # A tibble: 21 x 5
##    Estimator                ATE  lower_ci upper_ci treatment_lvl
##    <chr>                  <dbl>     <dbl>    <dbl> <chr>
##  1 RCT_gold_standard_1 0.00296 -0.000462  0.00639 1
##  2 RCT_gold_standard_2 0.00523  0.00168   0.00878 2
##  3 RCT_gold_standard_3 0.00590  0.00233   0.00948 3
##  4 linear_regression_1 0.00298 -0.000447  0.00641 1
##  5 linear_regression_2 0.00526  0.00170   0.00881 2
##  6 linear_regression_3 0.00594  0.00237   0.00951 3
##  7 IPW_logistic_1      0.00298 -0.000482  0.00644 1
##  8 IPW_logistic_2      0.00526  0.00167   0.00884 2
##  9 IPW_logistic_3      0.00594  0.00233   0.00955 3
## 10 IPW_glmnet_1        0.00297 -0.000495  0.00643 1
## # ... with 11 more rows
```

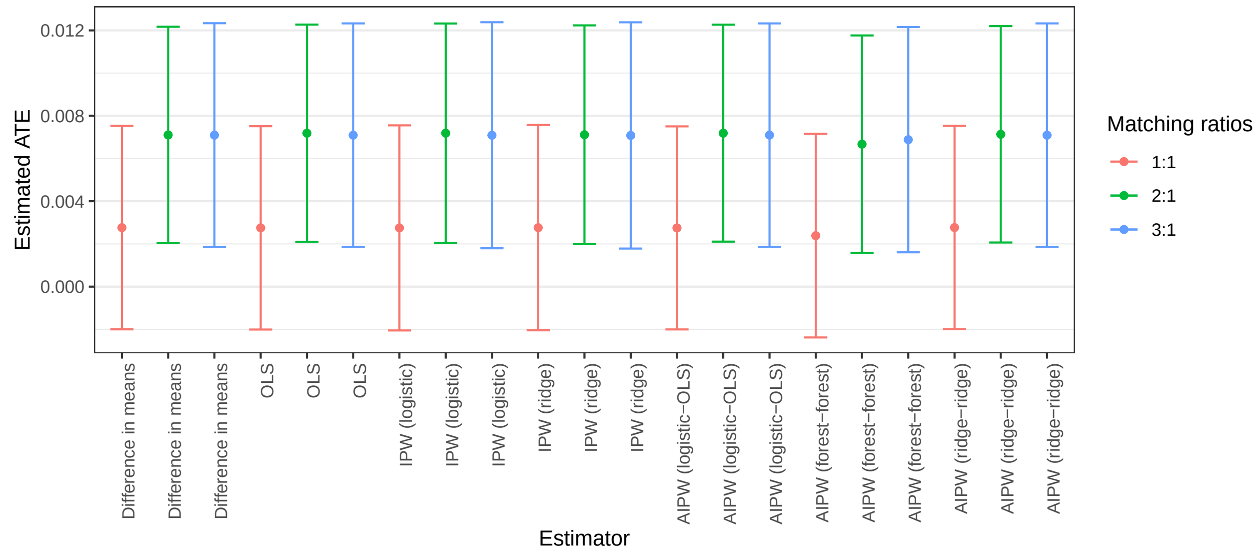
```r
print(ate_results_reduced)
```

```
## # A tibble: 21 x 5
##    Estimator                ATE lower_ci upper_ci treatment_lvl
##    <chr>                  <dbl>    <dbl>    <dbl> <chr>
##  1 RCT_gold_standard_1 0.00276 -0.00200  0.00753 1
##  2 RCT_gold_standard_2 0.00710  0.00204  0.0122  2
##  3 RCT_gold_standard_3 0.00709  0.00185  0.0123  3
##  4 linear_regression_1 0.00275 -0.00201  0.00751 1
##  5 linear_regression_2 0.00719  0.00210  0.0123  2
##  6 linear_regression_3 0.00709  0.00185  0.0123  3
##  7 IPW_logistic_1      0.00275 -0.00205  0.00755 1
##  8 IPW_logistic_2      0.00719  0.00205  0.0123  2
##  9 IPW_logistic_3      0.00709  0.00180  0.0124  3
## 10 IPW_glmnet_1        0.00276 -0.00205  0.00757 1
## # ... with 11 more rows
```

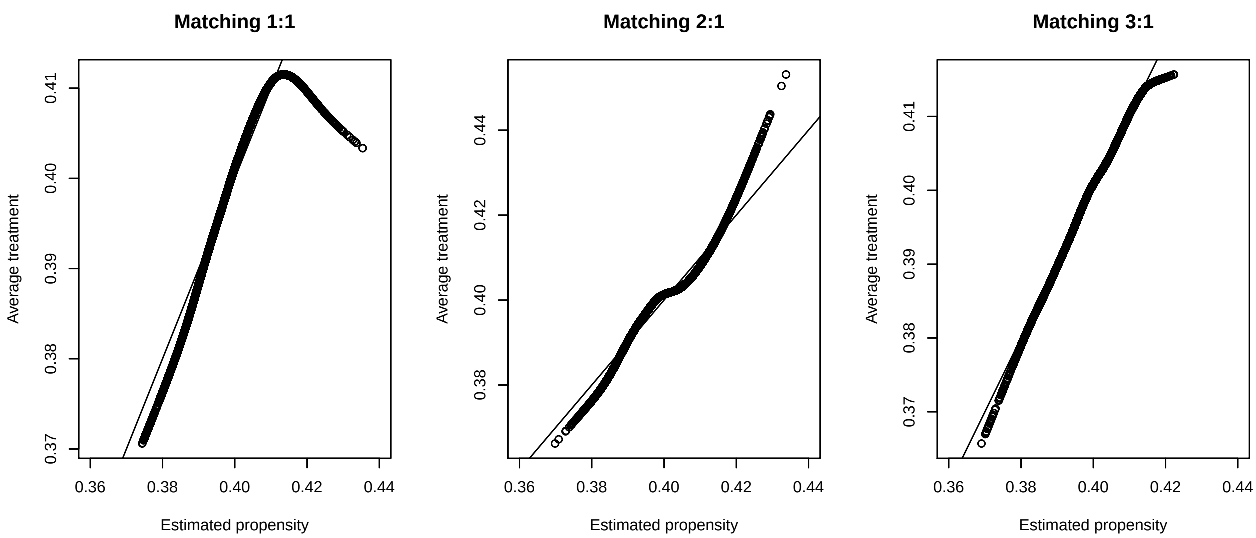
```r
ate_results$Estimator <- factor(ate_results$Estimator, levels = ate_results$Estimator)
ate_results_reduced$Estimator <- factor(ate_results_reduced$Estimator, levels = ate_results_reduced$Esti
plot_ATE_estimates(ate_results)
```




```r
plot_ATE_estimates(ate_results_reduced)
```

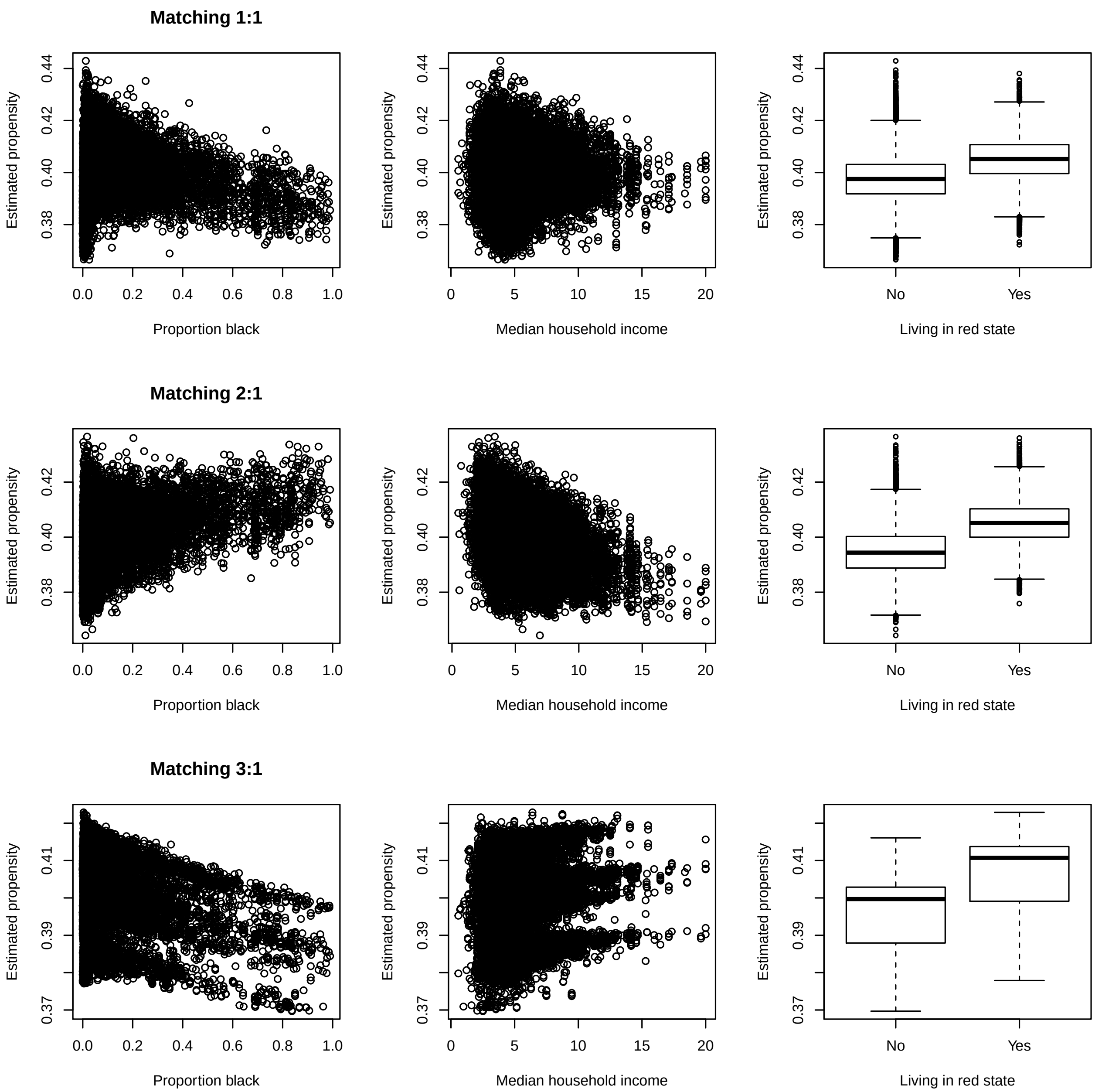



```
p_logistic1 = p_logistic(df1)
p_logistic2 = p_logistic(df2)
p_logistic3 = p_logistic(df3)
par(mfrow = c(1, 3))
plot_calibration(p_logistic1, df1$W, ylab = "Average treatment", xlab = "Estimated propensity",
                 xlim = c(0.36, 0.44), main = "Matching 1:1")
plot_calibration(p_logistic2, df2$W, ylab = "Average treatment", xlab = "Estimated propensity",
                 xlim = c(0.36, 0.44), main = "Matching 2:1")
plot_calibration(p_logistic3, df3$W, ylab = "Average treatment", xlab = "Estimated propensity",
                 xlim = c(0.36, 0.44), main = "Matching 3:1")
```



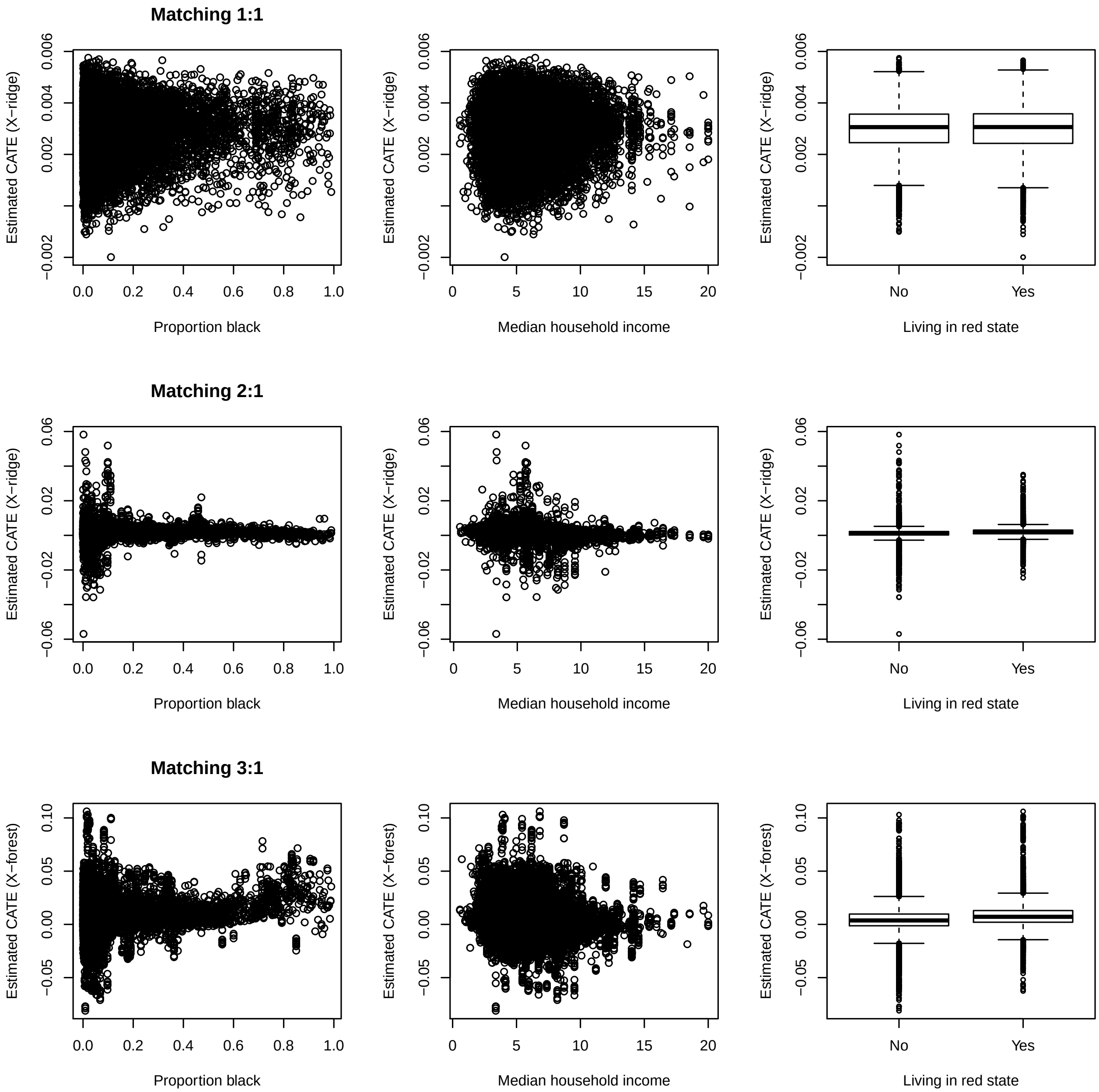

```
par(mfrow=c(3,3))
plot_yx(p_logistic(data_01$df), data_01$df$pblack, "Estimated propensity", "Proportion black", main = "I
plot_yx(p_logistic(data_01$df), data_01$df$median_hhincome, "Estimated propensity", "Median household i
plot_yx(p_logistic(data_01$df), as.factor(data_01$df$red0), "Estimated propensity", "Living in red stat
        names = c("No", "Yes"))

plot_yx(p_logistic(data_02$df), data_02$df$pblack, "Estimated propensity", "Proportion black", main = "I
plot_yx(p_logistic(data_02$df), data_02$df$median_hhincome, "Estimated propensity", "Median household i
```

```r
plot_yx(p_logistic(data_02$df), as.factor(data_02$df$red0), "Estimated propensity", "Living in red state
        names = c("No", "Yes"))

plot_yx(p_logistic(data_03$df), data_03$df$pblack, "Estimated propensity", "Proportion black", main = "M
plot_yx(p_logistic(data_03$df), data_03$df$median_hhincome, "Estimated propensity", "Median household in
plot_yx(p_logistic(data_03$df), as.factor(data_03$df$red0), "Estimated propensity", "Living in red state
        names = c("No", "Yes"))
```




```r
par(mfrow=c(3,3))
plot_yx(hte_best_01$estimate, hte_best_01$pblack, "Estimated CATE (X-ridge)", "Proportion black", main =
plot_yx(hte_best_01$estimate, hte_best_01$median_hhincome, "Estimated CATE (X-ridge)", "Median household
plot_yx(hte_best_01$estimate, as.factor(hte_best_01$red0), "Estimated CATE (X-ridge)", "Living in red st
        names = c("No", "Yes"))
```

```
plot_yx(hte_best_02$estimate, hte_best_02$pblack, "Estimated CATE (X-ridge)", "Proportion black", main =
plot_yx(hte_best_02$estimate, hte_best_02$median_hhincome, "Estimated CATE (X-ridge)", "Median househol
plot_yx(hte_best_02$estimate, as.factor(hte_best_02$red0), "Estimated CATE (X-ridge)", "Living in red s
        names = c("No", "Yes"))

plot_yx(hte_best_03$estimate, hte_best_03$pblack, "Estimated CATE (X-forest)", "Proportion black", main
plot_yx(hte_best_03$estimate, hte_best_03$median_hhincome, "Estimated CATE (X-forest)", "Median househo
plot_yx(hte_best_03$estimate, as.factor(hte_best_03$red0), "Estimated CATE (X-forest)", "Living in red
        names = c("No", "Yes"))
```




```
# Compute the doubly robust scores
if (!file.exists("results/hte_best_01.rds")) {
  drs_best_01 <- estimate_binary_drs(X_train = as.matrix(hte_best_01[ , covariate_names]),
                                     W_train = hte_best_01$W,
```

```r
                                    Y_train = hte_best_01$Y,
                                    cate_train = hte_best_01$estimate,
                                    X_new = NULL, W_new = NULL, Y_new = NULL, cate_new = NULL,
                                    num_trees = NUM_TREES, compute_new = FALSE)
  hte_best_01$drs <- drs_best_01$train
  saveRDS(hte_best_01, file = "results/hte_best_01.rds")
} else {
  hte_best_01 <- readRDS("results/hte_best_01.rds")
}

if (!file.exists("results/hte_best_02.rds")) {
  drs_best_02 <- estimate_binary_drs(X_train = as.matrix(hte_best_02[ , covariate_names]),
                                    W_train = hte_best_02$W,
                                    Y_train = hte_best_02$Y,
                                    cate_train = hte_best_02$estimate,
                                    X_new = NULL, W_new = NULL, Y_new = NULL, cate_new = NULL,
                                    num_trees = NUM_TREES, compute_new = FALSE)
  hte_best_02$drs <- drs_best_02$train
  saveRDS(hte_best_02, file = "results/hte_best_02.rds")
} else {
  hte_best_02 <- readRDS("results/hte_best_02.rds")
}

if (!file.exists("results/hte_best_03.rds")) {
  drs_best_03 <- estimate_binary_drs(X_train = as.matrix(hte_best_03[ , covariate_names]),
                                    W_train = hte_best_03$W,
                                    Y_train = hte_best_03$Y,
                                    cate_train = hte_best_03$estimate,
                                    X_new = NULL, W_new = NULL, Y_new = NULL, cate_new = NULL,
                                    num_trees = NUM_TREES, compute_new = FALSE)
  hte_best_03$drs <- drs_best_03$train
  saveRDS(hte_best_03, file = "results/hte_best_03.rds")
} else {
  hte_best_03 <- readRDS("results/hte_best_03.rds")
}

# Formal hypothesis tests
test_covs <- c("pblack", "median_hhincome", "red0")
test_cov_types <- c("continuous", "continuous", "binary")
df_lst <- list("0 vs 1" = hte_best_01, "0 vs 2" = hte_best_02, "0 vs 3" = hte_best_03)
for (comparison in names(df_lst)) {
  for (i in 1:length(test_covs)) {
    cat(paste0("Comparison: ", comparison), "\n")
    cat(paste0("Covariate: ", test_covs[i]), "\n")
    t_test <- test_covariate_hetero(df = df_lst[[comparison]],
                                   x_name = test_covs[i],
                                   x_type = test_cov_types[i])
    print(t_test)
    cat("Corrected p-value: ", min(c(t_test$p.value*length(test_covs)*length(df_lst), 1)), "\n\n")
  }
  cat("\n\n\n")
}
```

```
## Comparison: 0 vs 1
```

```
## Covariate: pblack
##
##  Welch Two Sample t-test
##
## data:  drs[!high_idx] and drs[high_idx]
## t = 0.42863, df = 26059, p-value = 0.6682
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.003515538  0.005483472
## sample estimates:
##   mean of x   mean of y
## 0.003288620 0.002304653
##
## Corrected p-value:  1
##
## Comparison: 0 vs 1
## Covariate: median_hhincome
##
##  Welch Two Sample t-test
##
## data:  drs[!high_idx] and drs[high_idx]
## t = 0.21332, df = 26003, p-value = 0.8311
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004009763  0.004989139
## sample estimates:
##   mean of x   mean of y
## 0.003043472 0.002553785
##
## Corrected p-value:  1
##
## Comparison: 0 vs 1
## Covariate: red0
##
##  Welch Two Sample t-test
##
## data:  drs[!pos_idx] and drs[pos_idx]
## t = -1.9775, df = 24229, p-value = 0.048
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.961753e-03 -3.960099e-05
## sample estimates:
##   mean of x   mean of y
## 0.001014136 0.005514814
##
## Corrected p-value:  0.4320074
##
##
##
##
## Comparison: 0 vs 2
## Covariate: pblack
##
##  Welch Two Sample t-test
```

```
##
## data:  drs[!high_idx] and drs[high_idx]
## t = -0.0069828, df = 25988, p-value = 0.9944
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004520740  0.004488643
## sample estimates:
##   mean of x   mean of y
## 0.005214524 0.005230572
##
## Corrected p-value:  1
##
## Comparison: 0 vs 2
## Covariate: median_hhincome
##
##  Welch Two Sample t-test
##
## data:  drs[!high_idx] and drs[high_idx]
## t = 0.60634, df = 25920, p-value = 0.5443
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.003111179  0.005898235
## sample estimates:
##   mean of x   mean of y
## 0.005919177 0.004525650
##
## Corrected p-value:  1
##
## Comparison: 0 vs 2
## Covariate: red0
##
##  Welch Two Sample t-test
##
## data:  drs[!pos_idx] and drs[pos_idx]
## t = -1.945, df = 23927, p-value = 0.05179
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0089389086  0.0000344735
## sample estimates:
##   mean of x   mean of y
## 0.003451801 0.007904018
##
## Corrected p-value:  0.4660769
##
##
##
##
## Comparison: 0 vs 3
## Covariate: pblack
##
##  Welch Two Sample t-test
##
## data:  drs[!high_idx] and drs[high_idx]
## t = 0.19059, df = 26075, p-value = 0.8489
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004073128  0.004950545
## sample estimates:
##   mean of x   mean of y
## 0.006350420 0.005911711
##
## Corrected p-value:  1
##
## Comparison: 0 vs 3
## Covariate: median_hhincome
##
##  Welch Two Sample t-test
##
## data:  drs[!high_idx] and drs[high_idx]
## t = 1.7588, df = 25978, p-value = 0.07862
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0004631758  0.0085599777
## sample estimates:
##   mean of x   mean of y
## 0.008155300 0.004106899
##
## Corrected p-value:  0.7075657
##
## Comparison: 0 vs 3
## Covariate: red0
##
##  Welch Two Sample t-test
##
## data:  drs[!pos_idx] and drs[pos_idx]
## t = -2.9333, df = 23903, p-value = 0.003357
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.011237163 -0.002235001
## sample estimates:
##   mean of x   mean of y
## 0.003450408 0.010186490
##
## Corrected p-value:  0.03021009
```

```
# test_CATE(hte_best_01, "estimate", "pblack", "continuous")
# test_CATE(hte_best_01, "estimate", "median_hhincome", "continuous")
# test_CATE(hte_best_01, "estimate", "red0", "factor")
#
# test_CATE(hte_best_02, "estimate", "pblack", "continuous")
# test_CATE(hte_best_02, "estimate", "median_hhincome", "continuous")
# test_CATE(hte_best_02, "estimate", "red0", "factor")
#
# test_CATE(hte_best_03, "estimate", "pblack", "continuous")
# test_CATE(hte_best_03, "estimate", "median_hhincome", "continuous")
# test_CATE(hte_best_03, "estimate", "red0", "factor")
```