Heterogeneity in voter preferences for female candidates

Marek Chadim

10-06-2024

1. Run an OLS regression of picked_cand_a on cand_a_female. What is the average preference for or against female candidates?

```r
load("politician_gender_prefs.rdata")
data <- df
data$picked_cand_a <- as.numeric(data$picked_cand_a)
data$cand_a_female <- as.numeric(data$cand_a_female)
summary(lm(picked_cand_a ~ cand_a_female, data = data))
```

```
##
## Call:
## lm(formula = picked_cand_a ~ cand_a_female, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5716 -0.4723  0.4284  0.4284  0.5277
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.47230    0.01899  24.871  < 2e-16 ***
## cand_a_female  0.09933    0.02674   3.715 0.000212 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4974 on 1382 degrees of freedom
## Multiple R-squared:  0.009886,   Adjusted R-squared:  0.009169
## F-statistic:  13.8 on 1 and 1382 DF,  p-value: 0.0002116
```

Run regressions of picked_cand_a on cand_a_female and one of each of the other included regressions, main effects and interactions.

```r
regressors <- setdiff(names(data), c("picked_cand_a", "cand_a_female"))
regression_results <- list()
for (regressor in regressors) {
  formula <- as.formula(paste("picked_cand_a ~ cand_a_female *", regressor))
  model <- lm(formula, data = data)
  regression_results[[regressor]] <- broom::tidy(model)
}
```

What is the most significant heterogeneity?

```
most_significant <- NULL
lowest_p_value <- Inf

for (regressor in regressors) {
  interaction_row <- regression_results[[regressor]] |>
    dplyr::filter(term == paste("cand_a_female:", regressor, sep = ""))

  if (nrow(interaction_row) > 0) {
    p_value <- interaction_row$p.value
    if (p_value < lowest_p_value) {
      lowest_p_value <- p_value
      most_significant <- regressor
    }
  }
}
most_significant_model <- lm(as.formula(paste("picked_cand_a ~ cand_a_female *", most_significant)), da
summary(most_significant_model)
```

```
##
## Call:
## lm(formula = as.formula(paste("picked_cand_a ~ cand_a_female *",
##     most_significant)), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6830 -0.5238  0.3806  0.4443  0.7494
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.57379    0.03212  17.866  < 2e-16 ***
## cand_a_female                 -0.04997    0.04459  -1.121    0.263
## considerwoman_n               -0.06464    0.01655  -3.906 9.85e-05 ***
## cand_a_female:considerwoman_n  0.09648    0.02330   4.141 3.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4943 on 1380 degrees of freedom
## Multiple R-squared:  0.02335,    Adjusted R-squared:  0.02123
## F-statistic:    11 on 3 and 1380 DF,  p-value: 3.881e-07
```

Why can’t we use the standard errors from these estimates for analyzing hetergeneity?

We need to correct for the fact that we are testing for multiple hypotheses, or we will end up with many false positives. We can use Bonferroni or Romano-Wolf correction to adjust the p-values for multiple testing.

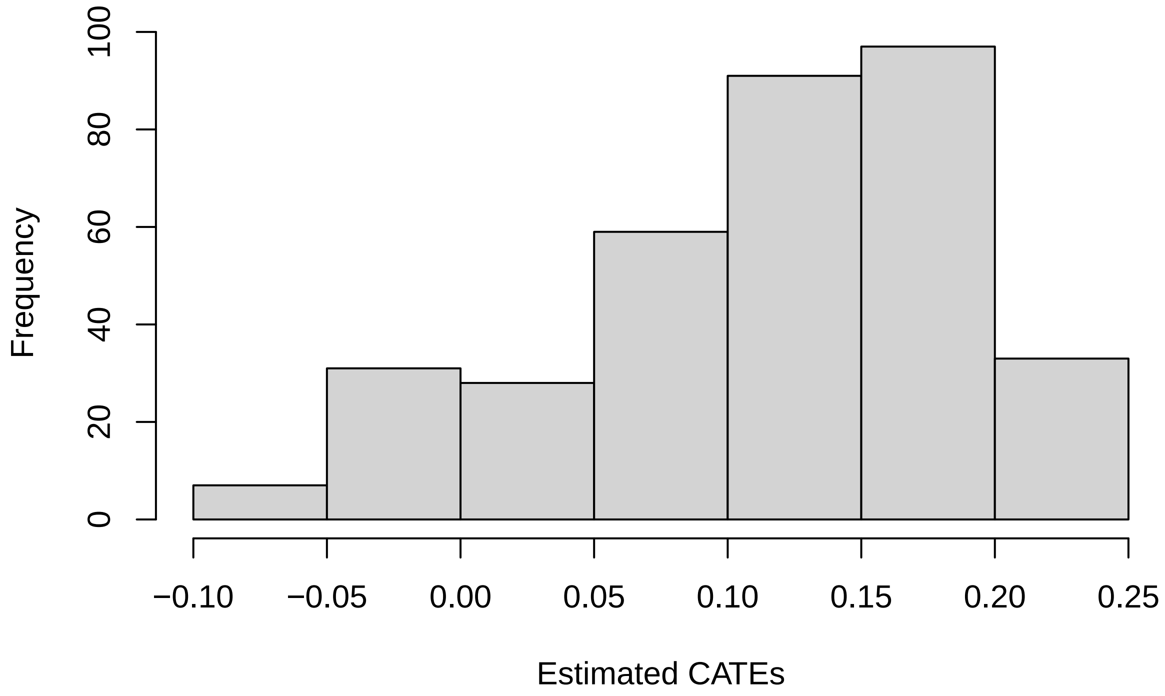
2. Estimate a causal forest model

```
Y = as.numeric(df$picked_cand_a)
W = as.numeric(df$cand_a_female)
X <- mltools::one_hot(
data.table::data.table(
```

```r
df |>
dplyr::select(-cand_a_female, -picked_cand_a) ) )

# Split data into a train and test sample.
set.seed(42)
train = sample(nrow(X), 0.75 * nrow(X))
test = -train

# Fit a CATE function on training data.
cate.forest = causal_forest(X[train, ], Y[train], W[train], num.trees = 5000, seed = 42)
```

3. Predict the heterogenous treatment effect

```r
X.test = X[test, ]
tau.hat.test = predict(cate.forest, X.test, estimate.variance=TRUE)$predictions

# A histogram of CATE estimates.
hist(tau.hat.test, xlab = "Estimated CATEs", main = "")
```



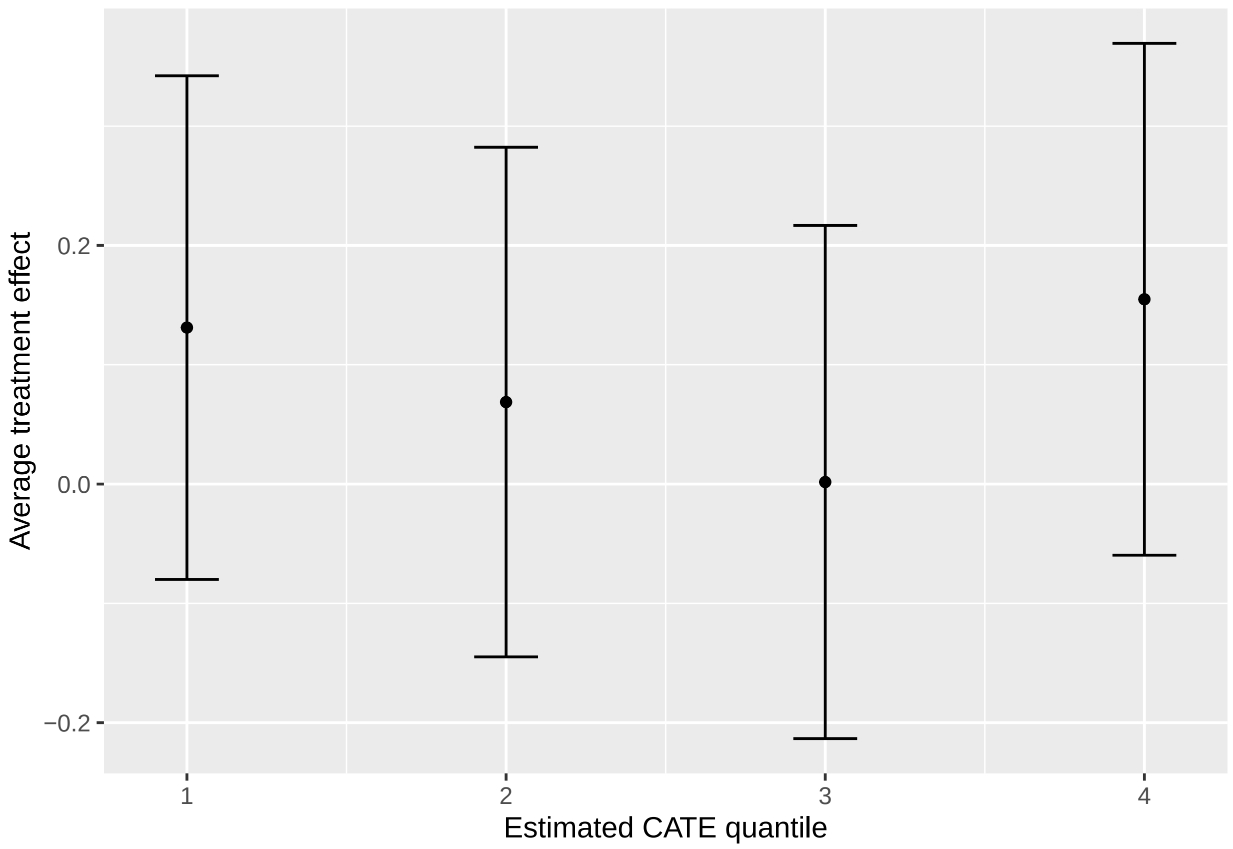
Does the predicted treatment effects predict treatment heterogeneity in the test data?

```r
summary(lm(Y[test] ~ W[test] + tau.hat.test + W[test] * tau.hat.test))
```

```
##
## Call:
## lm(formula = Y[test] ~ W[test] + tau.hat.test + W[test] * tau.hat.test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5749 -0.4531 -0.4287  0.5124  0.5750
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.43206    0.07726   5.592 4.59e-08 ***
## W[test]               0.05413    0.10095   0.536    0.592
## tau.hat.test          0.10399    0.56377   0.184    0.854
## W[test]:tau.hat.test  0.31334    0.74535   0.420    0.674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5003 on 342 degrees of freedom
## Multiple R-squared:  0.009789,   Adjusted R-squared:  0.001103
## F-statistic: 1.127 on 3 and 342 DF,  p-value: 0.3381
```

No, the interaction is not significant.

```r
# Vizualize
num.groups = 4
quartile = cut(tau.hat.test,
               quantile(tau.hat.test, seq(0, 1, by = 1 / num.groups)),
               labels = 1:num.groups,
               include.lowest = TRUE)
# Create a list of test set samples by CATE quartile.
samples.by.quartile = split(seq_along(quartile), quartile)

# Look at ATEs in each of these quartiles. To calculate these we fit a separate evaluation forest.
eval.forest = causal_forest(X.test, Y[test], W[test])

# Calculate ATEs for each group.
ate.by.quartile = lapply(samples.by.quartile, function(samples) {
  average_treatment_effect(eval.forest, subset = samples)
})

# Plot group ATEs along with 95% confidence bars.
df.plot.ate = data.frame(
  matrix(unlist(ate.by.quartile), num.groups, byrow = TRUE, dimnames = list(NULL, c("estimate","std.err
  group = 1:num.groups
)

ggplot(df.plot.ate, aes(x = group, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = estimate - 1.96 * std.err, ymax = estimate + 1.96 * std.err, width = 0.2)) +
  xlab("Estimated CATE quantile") +
  ylab("Average treatment effect")
```



4. What are the ten most important variables for hetergeneity?

```
varimp.cate = variable_importance(cate.forest)
ranked.variables = order(varimp.cate, decreasing = TRUE)
print(colnames(X)[ranked.variables[1:10]])
```

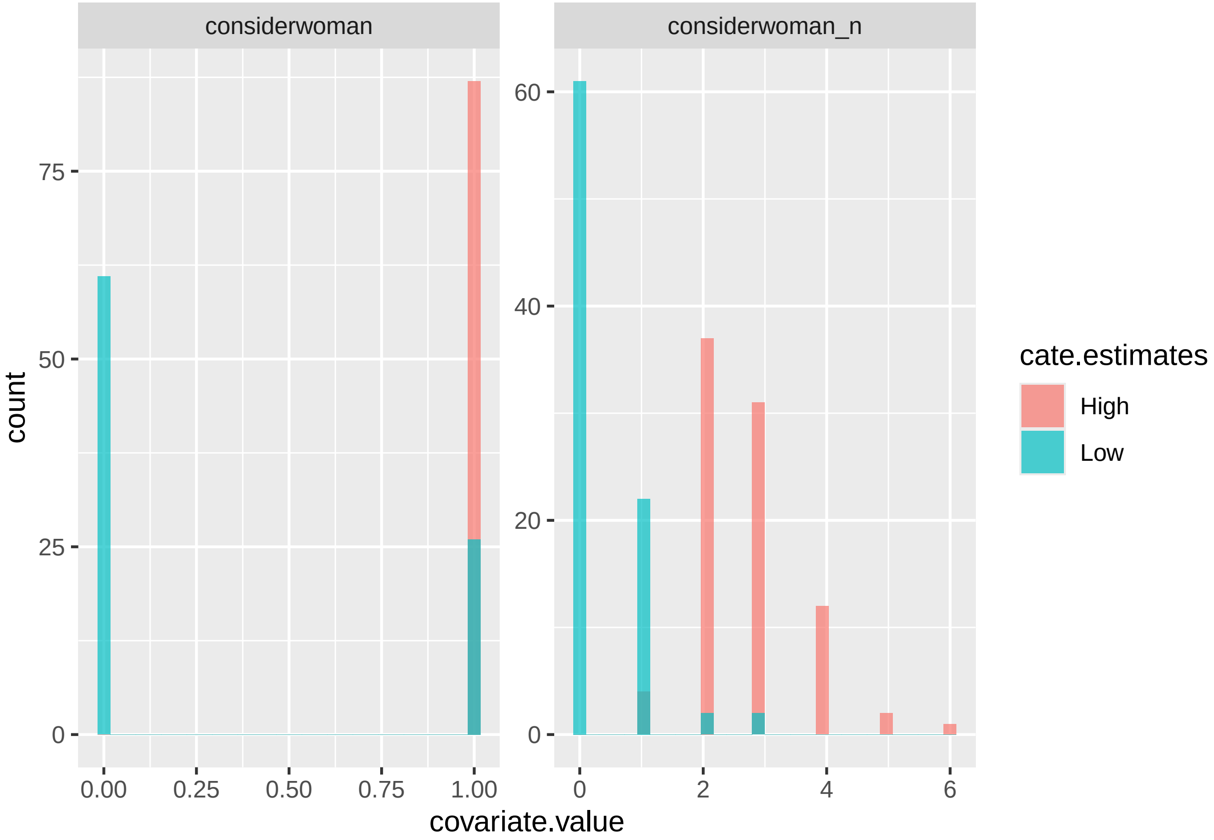
```
##  [1] "considerwoman_n"  "considerwoman"    "age"              "sdo"
##  [5] "progressive"      "candwarm"         "believe_Believe"  "horserace_Warren"
##  [9] "favmetoo"         "suppclean"
```

Plot the predicted treatment effect as a function of two most important variables.

```
top.varnames = colnames(X)[ranked.variables[1:2]]

# Select the test set samples predicted to have low/high CATEs.
low = samples.by.quartile[[1]]
high = samples.by.quartile[[num.groups]]

# Make some long format data frames for ggplot.
df.lo = data.frame(
  covariate.value = unlist(as.vector(X.test[low, ..top.varnames])),
  covariate.name = rep(top.varnames, each = length(low)),
  cate.estimates = "Low"
)
```

```
df.hi = data.frame(
  covariate.value = unlist(as.vector(X.test[high, ..top.varnames])),
  covariate.name = rep(top.varnames, each = length(high)),
  cate.estimates = "High"
)
df.plot.hist = rbind(df.lo, df.hi)

# Plot overlaid histograms of the selected covariates by low/high classification.
ggplot(df.plot.hist, aes(x = covariate.value, fill = cate.estimates)) +
  geom_histogram(alpha = 0.7, position = "identity") +
  facet_wrap(~ covariate.name, scales = "free", ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


considerwoman
considerwoman_n
count
covariate.value
cate.estimates
High
Low