jh8186 solutionsRmdHW3

Jerry Huang

2024-07-12

```r
# Library loading
library(readxl)
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
```

```r
library(tidyr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(haven) # Read dta
```

Problem 1 - CATE using GOTV

Part a. Data preparation

```r
# 1.
gotv <- read_xlsx("gotv_individual.xlsx")
gotv <- gotv %>%
  mutate(
    g2000 = as.numeric(g2000),
    p2000 = as.numeric(p2000),
    g2002 = as.numeric(g2002),
    p2002 = as.numeric(p2002),
    p2004 = as.numeric(p2004)
  )
gotv <- gotv %>%
  group_by(hh_id) %>%
  mutate(
    num_voted = g2000 + p2000 + g2002 + p2002 + p2004
  )

# 2.
cleaner <- gotv %>%
  select(hh_id, hh_size, num_voted, voted, treatment) %>%
  filter(treatment %in% c('Neighbors', 'Control')) # %in% operator checks if the treatment column value

# 3.
household <- cleaner %>%
  group_by(hh_id) %>%
  summarise(
    hh_size = mean(hh_size),
    num_voted = ceiling(mean(num_voted)), # Use ceiling to round up
    voted = mean(voted),
    treatment = first(treatment) # Use first to filter out duplicated rows
  )

# 4.
household_count <- household %>%
  group_by(num_voted, treatment) %>%
  summarise(num_households = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'num_voted'. You can override using the
## `.groups` argument.
```

```r
household_count %>%
  spread(key = treatment, value = num_households) %>%
  print()
```

```
## # A tibble: 6 x 3
##   num_voted Control Neighbors
##       <dbl>   <int>     <int>
## 1         0      74        14
## 2         1    3238       646
## 3         2   25397      5126
## 4         3   45511      9078
## 5         4   22901      4521
## 6         5    2878       615
```

4. From the table, we observe a drop in the number of households for ‘num_voted’ level 5 in both treatment groups. This suggests that only a small proportion of households have voted in all five previous elections.

Part b. CATE for subgroups

```r
# 1.

cate_hats <- household %>%
  group_by(num_voted, treatment) %>%
  summarise(mean_voted = mean(voted), n = n(), .groups = 'drop') %>%
  pivot_wider(names_from = treatment, values_from = c(mean_voted, n)) %>%
  mutate(
    CATE = mean_voted_Neighbors - mean_voted_Control,
    n_neighbors = n_Neighbors,
    n_control = n_Control
  )

cate_hats <- cate_hats %>%
  rowwise() %>%
  mutate(
    var_neighbors = var(household$voted[household$num_voted == num_voted & household$treatment == "Neigh
    var_control = var(household$voted[household$num_voted == num_voted & household$treatment == "Control
    var = var_neighbors / n_neighbors + var_control / n_control,
    se = sqrt(var)
  )

# 2.
cate_hats <- cate_hats %>%
  mutate(
    lower_ci = CATE - 1.96 * se,
    upper_ci = CATE + 1.96 * se
  )

print(cate_hats)
```

```
## # A tibble: 6 x 14
## # Rowwise:
##   num_voted mean_voted_Control mean_voted_Neighbors n_Control n_Neighbors   CATE
##       <dbl>              <dbl>                <dbl>     <int>       <int>  <dbl>
## 1         0              0.182                0.286        74          14 0.103
## 2         1              0.208                0.292      3238         646 0.0838
## 3         2              0.214                0.278     25397        5126 0.0640
## 4         3              0.296                0.387     45511        9078 0.0908
## 5         4              0.406                0.508     22901        4521 0.102
## 6         5              0.524                0.570      2878         615 0.0460
## # i 8 more variables: n_neighbors <int>, n_control <int>, var_neighbors <dbl>,
## #   var_control <dbl>, var <dbl>, se <dbl>, lower_ci <dbl>, upper_ci <dbl>
```

3. In conclusion, the 'neighbors' treatment from level 1 to 5 seems to significantly increase the probability of voting because their confidence intervals do not include -. For 'num_voted' levels at 0, the treatment effect is not statistically significant at the 5% level.

Part c. Effect modification

```
tau_0 <- cate_hats$CATE[cate_hats$num_voted == 0]
tau_5 <- cate_hats$CATE[cate_hats$num_voted == 5]

var_tau_0 <- cate_hats$var[cate_hats$num_voted == 0]
var_tau_5 <- cate_hats$var[cate_hats$num_voted == 5]

# estimated delta value
delta_hat <- tau_0 - tau_5

# variance of the estimated delta value
```

```r
var_delta_hat <- var_tau_0 + var_tau_5

# standard error of delta hat
se_delta_hat <- sqrt(var_delta_hat)

# 95% confidence interval
ci_delta_hat <- c(delta_hat - 1.96 * se_delta_hat, delta_hat + 1.96 * se_delta_hat)

print(delta_hat)
```

```
## [1] 0.0573247
```

```r
print(var_delta_hat)
```

```
## [1] 0.01799012
```

```r
print(ci_delta_hat)
```

```
## [1] -0.2055647  0.3202141
```

1. We cannot say that there's significant difference in the treatment effect for people who always vote and people who never vote.
2. The comparison between num_voted levels 0 and 5 showed that the estimated difference in treatment effects (delta_hat) was 0.05732, indicating a higher treatment effect for those who always vote compared to those who never vote. However, the 95% confidence interval for this difference (-0.2056, 0.3202) included zero, suggesting that the difference in treatment effects between these two extreme groups is not statistically significant. The treatment significantly increases voting probability for individuals who have voted between 1 to 5 times. There is no strong evidence of a significant difference in treatment effects between individuals who never vote and those who always vote. The findings highlight the importance of considering voting history when designing interventions to increase voter turnout, suggesting a focus on individuals with some voting history for more effective results.

Part d. Sample sizes and significance effect

1) When the overall sample size is fixed, dividing the sample into multiple subgroups reduces the number of observations in each subgroup. For example, if you have a total sample size of 1200 and divide it into 6 subgroups, each subgroup will have an average of 200 observations.
2) Statistical power is the probability of detecting a true effect if it exists. Power is influenced by the sample size, effect size, and significance level. Smaller sample sizes in each subgroup reduce the power to detect significant differences because there is less data to estimate the effect accurately. With fewer observations, the variability within each subgroup increases, making it harder to distinguish between the effect of the treatment and random noise.
3) Smaller subgroups have higher variance in their estimates because each observation has a larger impact on the mean and variance of the group. Higher variability within subgroups leads to larger standard errors, which in turn widen the confidence intervals of the estimated effects.
4) When testing multiple hypotheses, the likelihood of Type I errors (false positives) increases. To control for this, adjustments such as the Bonferroni correction are often applied, which makes it harder to achieve statistical significance. Each additional hypothesis tested reduces the effective significance level for each individual test, requiring stronger evidence to claim significance.

Problem 2 - Stratification using GOTV

Part a.

```
neighbors <- subset(cleaner, treatment == 'Neighbors')
control <- subset(cleaner, treatment == 'Control')

mean_neighbors <- mean(neighbors$voted, na.rm = TRUE)
mean_control <- mean(control$voted, na.rm = TRUE)

ate <- mean_neighbors - mean_control

var_neighbors <- var(neighbors$voted, na.rm = TRUE)
var_control <- var(control$voted, na.rm = TRUE)
se <- sqrt(var_neighbors / length(neighbors$voted) + var_control / length(control$voted))

lower_ci <- ate - 1.96 * se
upper_ci <- ate + 1.96 * se

result <- data.frame(
  ATE = ate,
  SE = se,
  Lower_CI = lower_ci,
  Upper_CI = upper_ci
)


print(result)
```

```
##          ATE          SE   Lower_CI   Upper_CI
## 1 0.08130991 0.002691753 0.07603408 0.08658575
```

Part b.

```
# Calculate the block-level estimator for ATE
total_sample_size <- sum(cate_hats$n_neighbors + cate_hats$n_control)

cate_hats <- cate_hats %>%
  mutate(
    N = n_neighbors + n_control,
    weights = N / total_sample_size
  )

tau_block_hat <- sum(cate_hats$weights * cate_hats$CATE)

cate_hats <- cate_hats %>%
  mutate(
    weighted_var = (weights^2) * (var_neighbors / n_neighbors + var_control / n_control)
  )

var_tau_block_hat <- sum(cate_hats$weighted_var)
se_tau_block_hat <- sqrt(var_tau_block_hat)

lower_ci_tau_block_hat <- tau_block_hat - qnorm(0.975) * se_tau_block_hat
upper_ci_tau_block_hat <- tau_block_hat + qnorm(0.975) * se_tau_block_hat

result_tau_block <- data.frame(
  tau_block_hat = tau_block_hat,
  SE = se_tau_block_hat,
  Lower_CI = lower_ci_tau_block_hat,
  Upper_CI = upper_ci_tau_block_hat
)

print(result_tau_block)
```

```
##   tau_block_hat          SE   Lower_CI   Upper_CI
## 1    0.08499853 0.003337304 0.07845754 0.09153953
```

In part a, the ATE is 0.08131, which means that the treatment group (Neighbors) had a 8.131% higher voting rate compared to the control group. The effect is statistically significant because the 95% confidence interval is (0.07603, 0.08659), which does not include 0. In part b, the ATE is 0.085,which means that the treatment group (Neighbors) had a 8.5% higher voting rate compared to the control group. The effect is statistically significant because the 95% confidence interval is (0.07846, 0.09154), which does not include 0. The standard error in part b is higher than part a (0.003337 > 0.002692), because the population within each stratum (block) is smaller compared to the overall sample.

Part c.

Problem 3

Part a.

M and Y are colliders. A, X, and Z are non-colliders. ## Part b. We should condition on X, because it affects both A and Y, which makes it a confounder in this path. Conditioning on a non-collider will block the path. We should not condition on Z because it will block the causal path between A and Y. ## Part c. Path 1: M <- Z -> Y Path 2: M <- A -> Z -> Y Path 3: M <- A <- X -> Y For path 1, we should condition on Z. For path 2, we should condition on A or Z. For path 3, we should condition on A or X.

Problem 4

Part a.

```
trc <- read_dta("trc_data.dta")

mean_treated <- mean(trc$RUSTAND[trc$TRCKNOW == 1])
mean_control <- mean(trc$RUSTAND[trc$TRCKNOW == 0])

ATE_trc <- mean_treated - mean_control

se_ATE_trc <- sqrt(var(trc$RUSTAND[trc$TRCKNOW == 1])/length(trc$RUSTAND[trc$TRCKNOW == 1]) +
var(trc$RUSTAND[trc$TRCKNOW == 0])/length(trc$RUSTAND[trc$TRCKNOW == 0]))
CI_lower <- ATE_trc - qnorm(0.975)*se_ATE_trc
CI_upper <- ATE_trc + qnorm(0.975)*se_ATE_trc

result_trc <- data.frame(
  ATE = ATE_trc,
  SE = se_ATE_trc,
  CI_lower = CI_lower,
  CI_upper = CI_upper
)
print(result_trc)
```

```
##           ATE         SE   CI_lower   CI_upper
## 1 -0.2177317 0.04433111 -0.3046191 -0.1308444
```

As the table shown, we have an ATE of -0.2177 meaning that on average the subjects who are exposure to TRC will have lower racial attitudes, with a confidence interval (-0.3046, -0.1308). As the confidence interval does not include 0, we can say that TRC may reduce the racial attitudes.

Part b.

```
trc <- trc %>%
  mutate(
    age_std = age/sd(age),
    female_std = female/sd(female),
    wealth_std = wealth/sd(wealth),
    religiosity_std = religiosity/sd(religiosity),
    ethsalience_std = ethsalience/sd(ethsalience),
    rcblack_std = rcblack/sd(rcblack),
    rcwhite_std = rcwhite/sd(rcwhite),
    rccol_std = rccol/sd(rccol),
    EDUC_std = EDUC/sd(EDUC)
  )

balance_table <- trc %>%
  group_by(TRCKNOW) %>%
  summarise(
    age_std = mean(age_std),
    female_std = mean(female_std),
    wealth_std = mean(wealth_std),
    religiosity_std = mean(religiosity_std),
    ethsalience_std = mean(ethsalience_std),
    rcblack_std = mean(rcblack_std),
    rcwhite_std = mean(rcwhite_std),
    rccol_std = mean(rccol_std),
    EDUC_std = mean(EDUC_std)
  )
balance_table
```

```
## # A tibble: 2 x 10
##   TRCKNOW age_std female_std wealth_std religiosity_std ethsalience_std
##     <dbl>   <dbl>      <dbl>      <dbl>           <dbl>           <dbl>
## 1       0    2.62      0.866      0.774            2.15            4.69
## 2       1    2.52      1.08       0.928            2.11            4.73
## # i 4 more variables: rcblack_std <dbl>, rcwhite_std <dbl>, rccol_std <dbl>,
## #   EDUC_std <dbl>
```

```
abs_balance_diff <- abs(balance_table[1, 2:ncol(balance_table)] - balance_table[2, 2:ncol(balance_table
abs_balance_diff
```

```
##     age_std female_std wealth_std religiosity_std ethsalience_std rcblack_std
## 1 0.0980385  0.2106527  0.1539068      0.04139381      0.03641136  0.07762829
##   rcwhite_std rccol_std  EDUC_std
## 1  0.03762548 0.1367213 0.3840434
```

After standardization, we observe a significant difference in wealth, female, Respondent is coloured, and the Level of education between the treated and control subjects. These imbalances suggest that simple comparisons might be biased and that further adjustment is necessary to control for these covariates and obtain a more accurate estimate of the treatment effect.

Part c.

```
# 1. Fit the logistic regression model
propensity_score_model <- glm(TRCKNOW ~ age + female + wealth + religiosity + ethsalience + rcblack + rc

# 2. Inverse propensity score for each observation
trc$propensity <- predict(propensity_score_model, type = "response")

# 3.
trc$stb_wt <- NA
trc$stb_wt[trc$TRCKNOW == 1] <- mean(trc$TRCKNOW==1)/trc$propensity[trc$TRCKNOW==1]
trc$stb_wt[trc$TRCKNOW == 0] <- mean(trc$TRCKNOW==0)/(1 - trc$propensity[trc$TRCKNOW==0])

trc$stb_wt <- (mean(trc$TRCKNOW==1)/trc$propensity) * trc$TRCKNOW +  (mean(trc$TRCKNOW==0)/(1 - trc$prop

# Verify: means of stabilized weights should be close to 1 in treated/control
mean(trc$stb_wt[trc$TRCKNOW == 1])
```

```
## [1] 1.000541
```

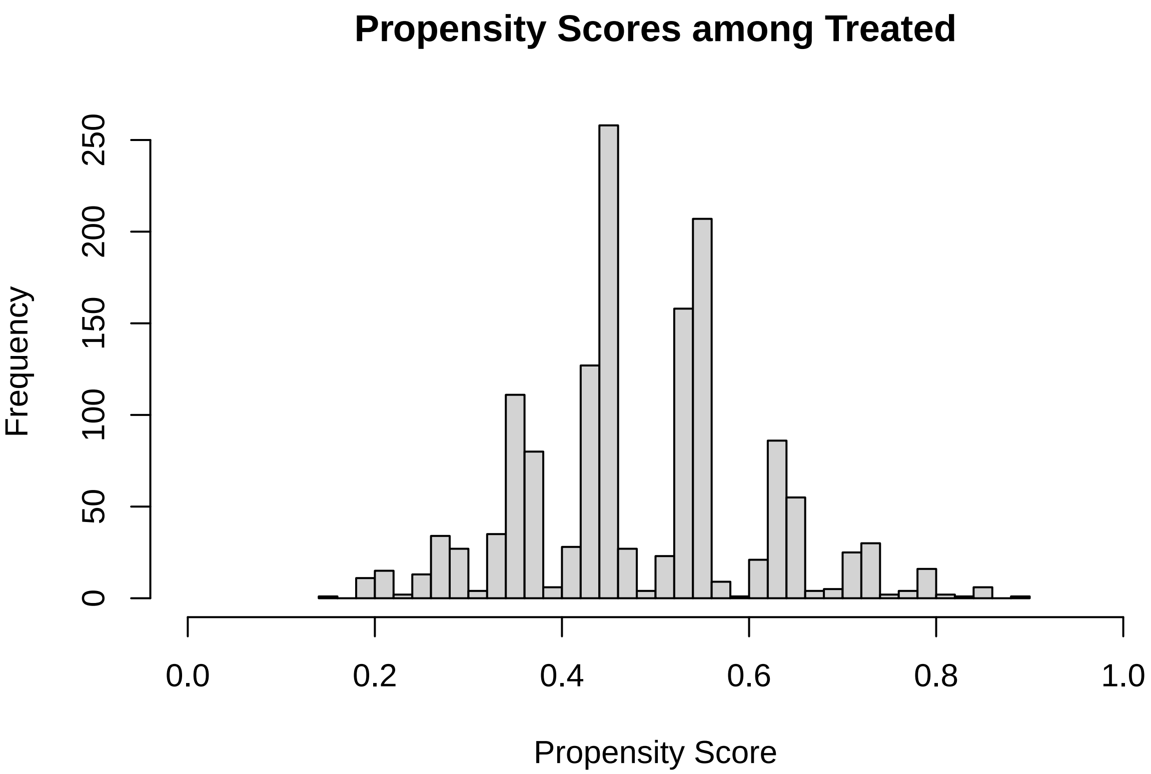
```
mean(trc$stb_wt[trc$TRCKNOW == 0])
```

```
## [1] 1.000198
```

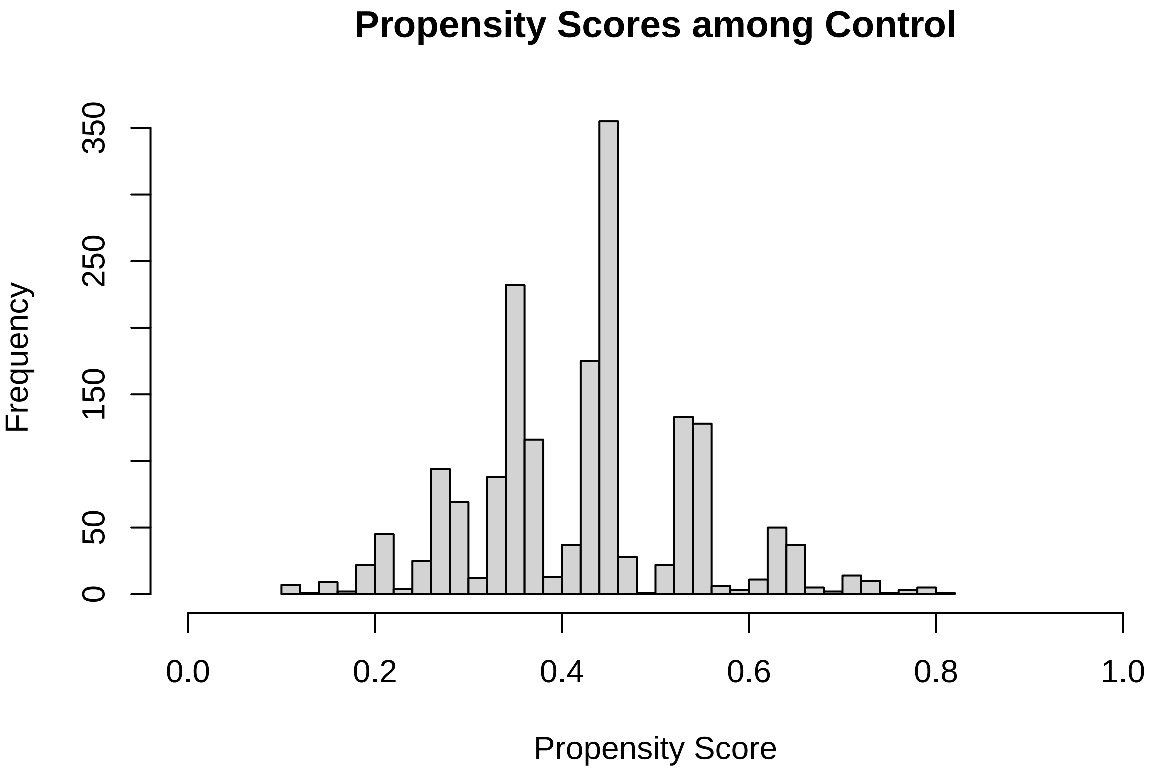
```
# Weighted outcomes for treated and control groups
weighted_outcome_treated <- sum(trc$RUSTAND[trc$TRCKNOW == 1] * trc$stb_wt[trc$TRCKNOW == 1]) / sum(trc
weighted_outcome_control <- sum(trc$RUSTAND[trc$TRCKNOW == 0] * trc$stb_wt[trc$TRCKNOW == 0]) / sum(trc

# IPW estimator for the ATE
ipw_ate <- weighted_outcome_treated - weighted_outcome_control

# Print the IPW estimator for the ATE
print(ipw_ate)
```

```
## [1] -0.1631028
```

```
# 4.
hist(trc$propensity[trc$TRCKNOW == 1], xlab="Propensity Score", main="Propensity Scores among Treated",
```



```r
hist(trc$propensity[trc$TRCKNOW == 0], xlab="Propensity Score", main="Propensity Scores among Control",
```



## Part d

```r
set.seed(1000)
nBoot <- 1000 # Number of iterations
```

```
ate_boot <- rep(NA, nBoot) # Placeholder to store estimates

# For each iteration
for(boot in 1:nBoot){
  # Resample rows with replacement
  trc_boot <- trc[sample(1:nrow(trc), nrow(trc), replace=T),] #replace = T is key!

  # Fit the propensity score model on the bootstrapped data
  pscore_model_boot <- glm(TRCKNOW ~ age + female + wealth +
                             religiosity + ethsalience + rcblack + rcwhite + rccol + EDUC, data = trc,

  # Save the propensities
  trc_boot$propensity <- predict(pscore_model_boot, type = "response")

  # Calculate the (stabilized) weights
  trc_boot$stb_wt <- (mean(trc_boot$TRCKNOW==1)/trc_boot$propensity) * trc_boot$TRCKNOW +
    (mean(trc_boot$TRCKNOW==0)/(1 - trc_boot$propensity)) * (1 - trc_boot$TRCKNOW)

  # Store the weighted difference-in-means
  ate_boot[boot] <- weighted.mean(trc_boot$RUSTAND[trc_boot$TRCKNOW == 1], trc_boot$stb_wt[trc_boot$TRC
  weighted.mean(trc_boot$RUSTAND[trc_boot$TRCKNOW == 0], trc_boot$stb_wt[trc_boot$TRCKNOW == 0])

}

# Take the SD of the ate_boot to get our estimated SE - can do asymptotic inference
sd(ate_boot)
```

```
## [1] 0.04638179
```

```
# Asymptotic 95\% CI
point_wtd <- mean(ate_boot)
c(point_wtd - qnorm(.975)*sd(ate_boot),
  point_wtd + qnorm(.975)*sd(ate_boot))
```

```
## [1] -0.3099755 -0.1281622
```

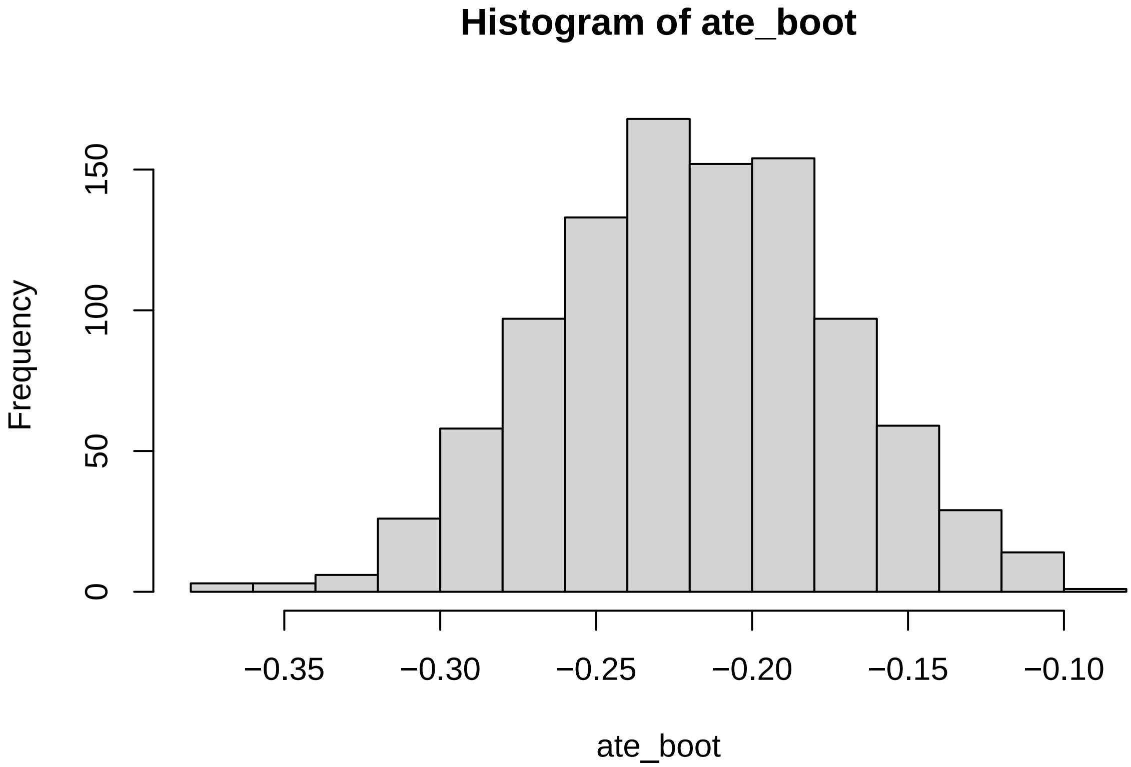
```
# Take quantiles to get CIs directly from the bootstrapped distribution (esp. if skewed)
quantile(ate_boot, c(.025, .975))
```

```
##       2.5%      97.5%
## -0.3083216 -0.1286676
```

```
result_boot <- data.frame(
  Estimate = point_wtd,
  SD = sd(ate_boot),
  CI_lower = point_wtd - qnorm(.975)*sd(ate_boot),
  CI_upper = point_wtd + qnorm(.975)*sd(ate_boot)
)
print(result_boot)
```

```
##     Estimate         SD   CI_lower   CI_upper
## 1 -0.2190688 0.04638179 -0.3099755 -0.1281622
```

```r
hist(ate_boot)
```



The bootstrap estimate of ATE is -0.2191 and the 95% confidence interval is (-0.31, -0.1282). This suggests that the true ATE lies somewhere within this interval and since this interval does not contain zero, the treatment effect is statistically significant at the 5% level. In Part A we have found out that the point estimate is -0.2177. However, the ATE from the IPTW approach is slightly more negative than the ATE from Part A, suggesting a stronger effect when accounting for the propensity to receive the treatment based on the observed covariates. In Part B we found out that the ATE is -0.1631. This indicates that when controlling for the covariates, the TRC exposure is still associated with a reduction in racial attitudes, reinforcing the results from Part A, but with a smaller magnitude. The effect is statistically significant. The 95% confidence intervals from both the simple differences and bootstrap methods do not include zero, suggesting that the reduction in racial attitudes is unlikely to be due to random chance.