Gerber and Green (2012) Chapter 6 Problem 10

Margaret Moor and Alexander Coppock, Yale University

January 30, 2018

This script shows how to conduct the randomization inference procedure in Gerber and Green (2012) Chapter 6 Problem 10 three different ways: using the `ri2` package, using the `ri` package, and by hand with a loop.

Chapter 6 Problem 10

In her study of election monitoring in Indonesia, Hyde randomly assigned international election observers to monitor certain polling stations. Here, we consider a subset of her experiment where approximately 20% of the villages were assigned to the treatment group. Because of difficult terrain and time constraints, observers monitored 68 of the 409 polling places assigned to treatment. Observers also monitored 21 of the 1,562 stations assigned to the control group. The dependent variable here is the number of ballots that were declared invalid by polling station officials.

(a) Is monotonicity a plausible assumption in this application?

NOT SHOWN

(b) Under the assumption of monotonicity, what proportion of subjects (polling locations) would you estimate to be Compliers, Never-Takers, and Always-Takers?

SHOWN BELOW

(c) Explain what the non-interference assumption means in the context of this experiment.
(d) Download the sample dataset at http://isps.research.yale.edu/FEDAI and estimate the ITT and the CACE. Interpret the results.

NOT SHOWN

(e) Use randomization inference to test the sharp null hypothesis that there is no intent-to-treat effect for any polling location. Interpret the results. Explain why testing the null hypothesis that the ITT is zero for all subjects serves the same purpose as testing the null hypothesis that the ATE is zero for all Compliers.

SHOWN BELOW

```r
# Data from http://isps.yale.edu/FEDAI
library(haven)
data6.10 <- read_dta("datasets/6.10.dta")

# Number of sims the same for all three methods
sims <- 1000

# 409 of 1971 assigned to treatment
table(data6.10$Sample)
```

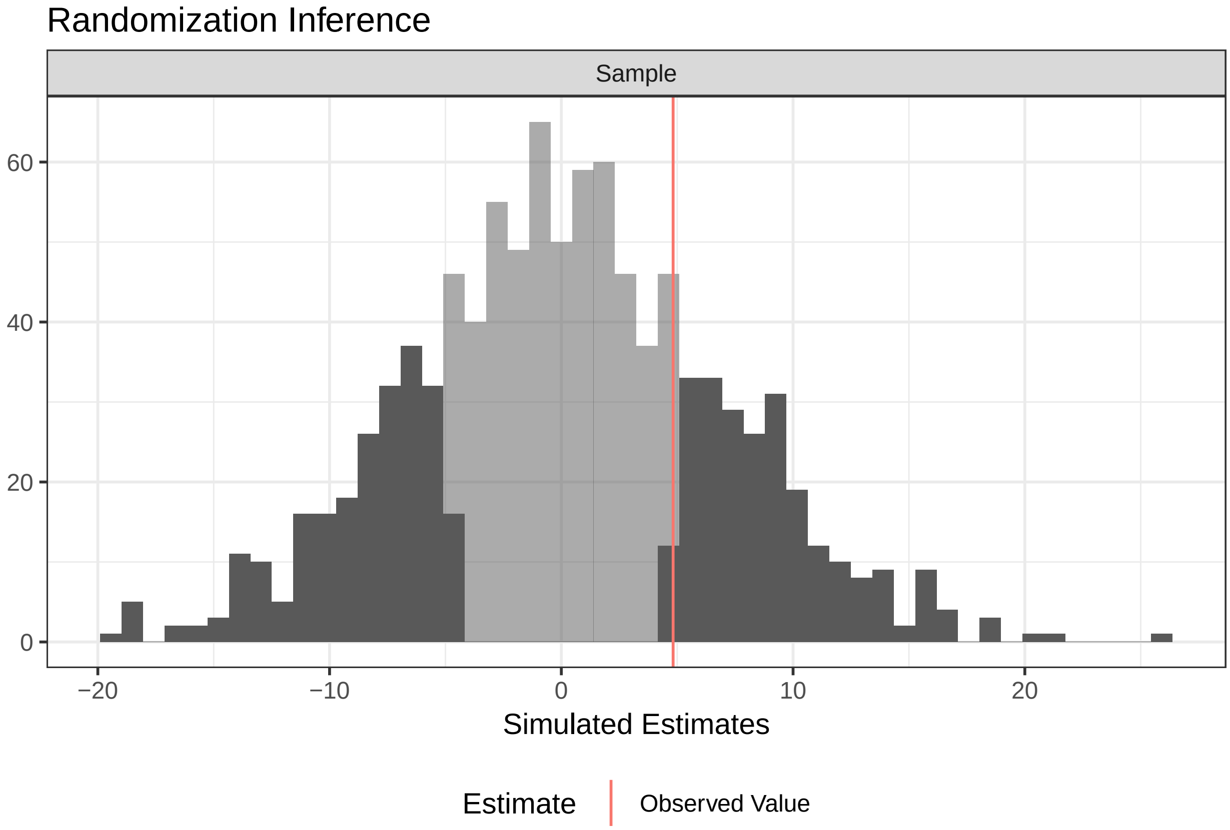
```
##
##    0    1
## 1562  409
```

In ri2

```
library(randomizr)
library(ri2)

# Declare randomization procedure
declaration <- declare_ra(N = 1971, m = 409)

# Conduct Randomization Inference
ri2_out <- conduct_ri(
  invalidballots ~ Sample,
  declaration = declaration,
  assignment = "Sample",
  sharp_hypothesis = 0,
  data = data6.10
  )

summary(ri2_out)
```

```
##   coefficient estimate two_tailed_p_value null_ci_lower null_ci_upper
## 1      Sample 4.824097              0.475     -13.26871      13.78311
```

```
plot(ri2_out)
```

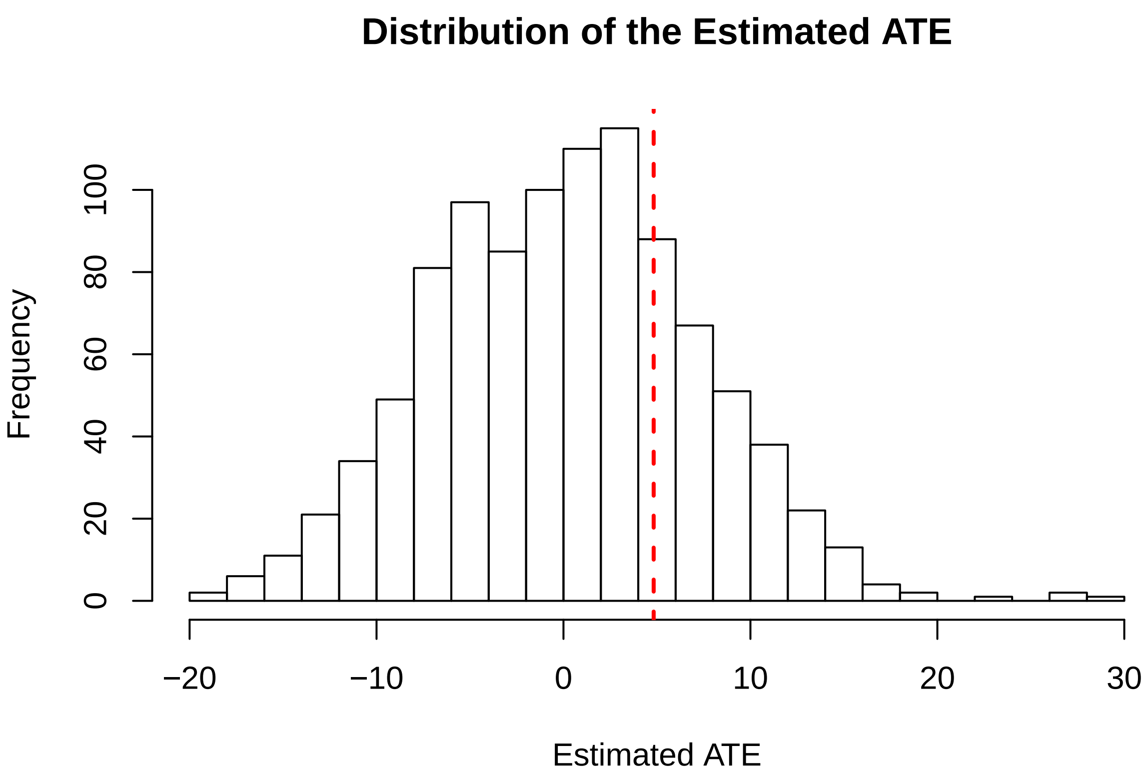


In ri

```r
library(ri)

# all possible permutations
perms <- genperms(data6.10$Sample, maxiter = sims)
```

```
## Too many permutations to use exact method.
## Defaulting to approximate method.
## Increase maxiter to at least Inf to perform exact estimation.
```

```r
# probability of treatment
probs <- genprobexact(data6.10$Sample)
# estimate the ITT
ate <- estate(data6.10$invalidballots, data6.10$Sample, prob = probs)

## Conduct Sharp Null Hypothesis Test of Zero Effect for Each Unit

# generate potential outcomes under sharp null of no effect
Ys <- genouts(data6.10$invalidballots, data6.10$Sample, ate = 0)
# generate sampling dist. under sharp null
distout <- gendist(Ys, perms, prob = probs)
# display characteristics of sampling dist. for inference
ri_out <- dispdist(distout, ate)
```




```r
ri_out
```

```
## $two.tailed.p.value
## [1] 0.502
##
```

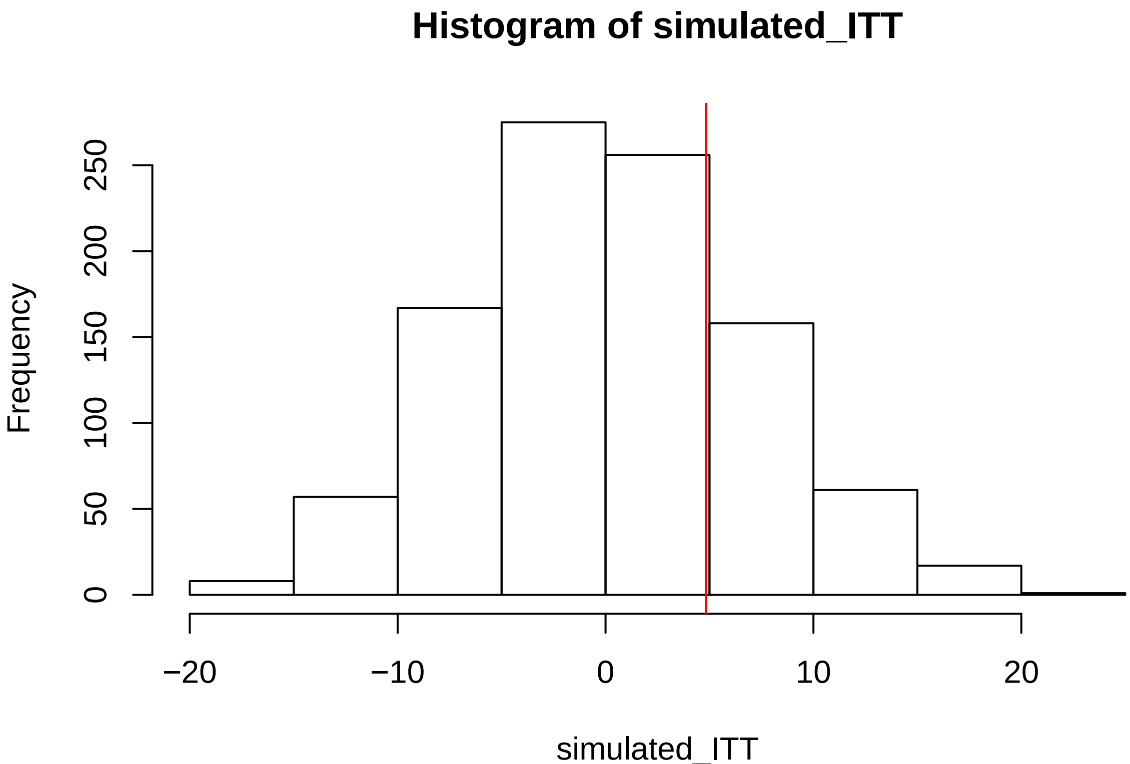
```
## $two.tailed.p.value.abs
## [1] 0.508
##
## $greater.p.value
## [1] 0.251
##
## $lesser.p.value
## [1] 0.749
##
## $quantile
##      2.5%     97.5%
## -13.25698  13.77447
##
## $sd
## [1] 7.107569
##
## $exp.val
## [1] 0.1206653
```

By hand

```r
library(randomizr)
N = 1971

ITT = with(data6.10, mean(invalidballots[Sample == 1]) - mean(invalidballots[Sample == 0]))

simulated_ITT <- rep(NA, sims)

for (i in 1:sims){
  data6.10$Z_sim <- complete_ra(N = 1971, 409)
  simulated_ITT[i] <- with(data6.10, mean(invalidballots[Z_sim == 1]) - mean(invalidballots[Z_sim == 0]
}

p_two_tailed <- mean(abs(simulated_ITT) >= abs(ITT))
p_upper <- mean(simulated_ITT >= ITT)
p_lower <- mean(simulated_ITT <= ITT)

c(ITT, p_two_tailed, p_upper, p_lower)
```

```
## [1] 4.824097 0.486000 0.242000 0.759000
```

```r
hist(simulated_ITT, breaks = 10)
abline(v = ITT, col = "red")
```


Histogram of simulated_ITT
Frequency
0
50
100
150
200
250
−20
−10
0
10
20
simulated_ITT