Gerber and Green (2012) Chapter 4 Problem 2

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 4 Problem 2 three different ways: using the `ri2` package, using the `ri` package, and by hand with a loop.

Chapter 4 Problem 2

A researcher working with Israeli elementary school students sought to improve students' ability to solve logic puzzles. 21 Students in the treatment and control group initially took a computer-administered test, and the number of correctly solved puzzles was recorded. A few days later, students assigned to the control group were then given 30 minutes to improve their puzzle-solving skills by playing on a computer. During the same allotment of time, students in the treatment group listened to an instructor describe some rules of thumb to keep in mind when solving logic puzzles. All subjects then took a computer-administered post-test, and the number of correctly solved puzzles was recorded.

(a) As a randomization check, use randomization inference to test the null hypothesis that the pre-test scores are unaffected by treatment assignment.

NOT SHOWN

(b) Use difference-in-means estimation to estimate the effect of the treatment on the post-test score. Form a 95% confidence interval.

SHOWN BELOW

(c) Use difference-in-differences estimation to estimate the effect of the treatment on the post-test score. Form a 95% confidence interval, and compare it to the interval in part (b).

NOT SHOWN

```
# Data from http://isps.yale.edu/FEDAI
library(haven)
data4.2 <- read_dta("datasets/4.2.dta")
# Number of sims the same for all three methods
sims <- 1000
```

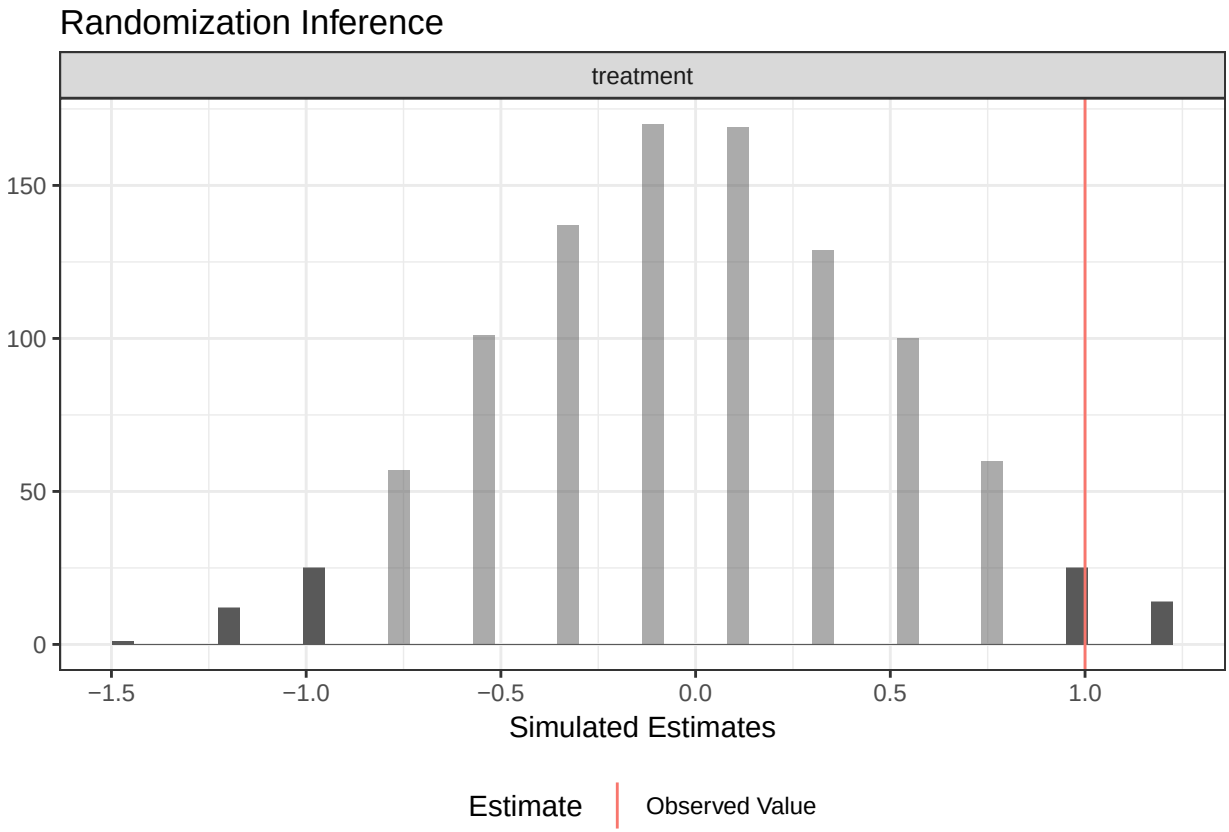
In ri2

```
library(ri2)

# Declare randomization procedure
declaration <- declare_ra(N = 18, m = 9)

# Conduct Randomization Inference
ri2_out <- conduct_ri(improvement ~ treatment,
                      declaration = declaration,
                      assignment = "treatment",
                      sharp_hypothesis = 0,
                      sims = sims,
                      data = data4.2)
```

```r
summary(ri2_out)
```

```
##   coefficient estimate two_tailed_p_value null_ci_lower null_ci_upper
## 1   treatment        1               0.077            -1              1
```

```r
plot(ri2_out)
```



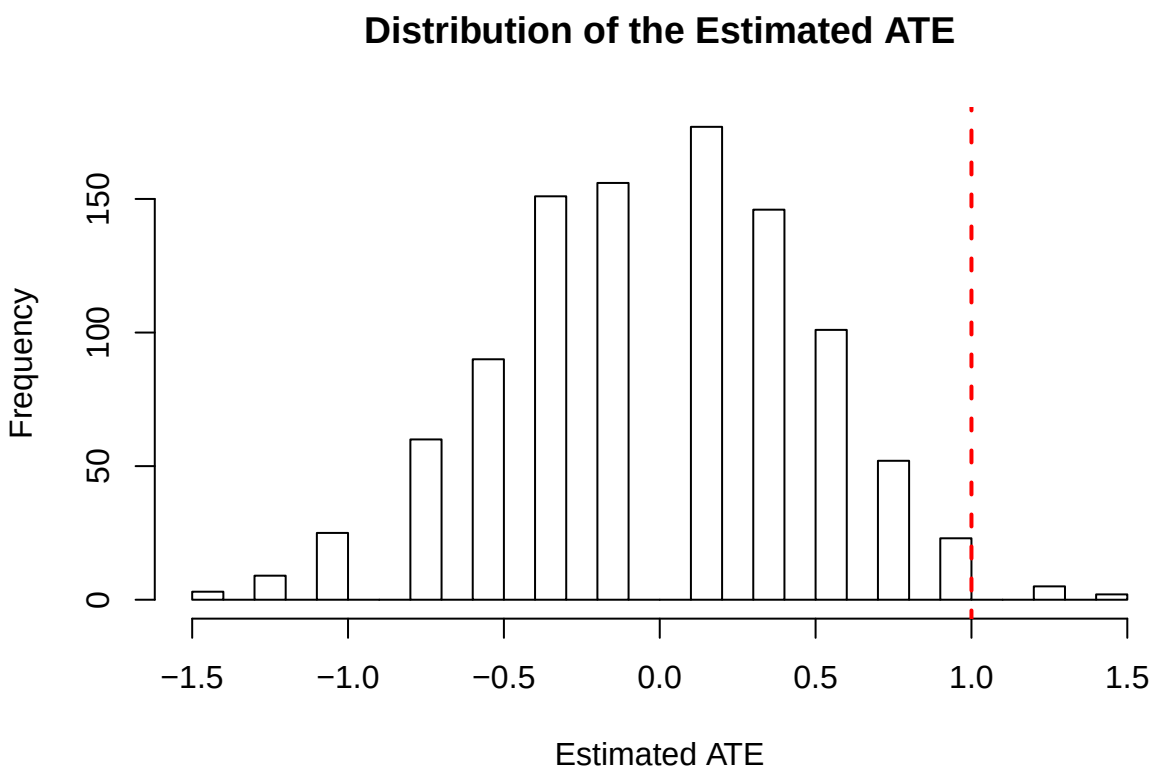
In ri

```r
library(ri)
# all possible permutations
perms <- genperms(data4.2$treatment, maxiter = sims)
```

```
## Too many permutations to use exact method.
## Defaulting to approximate method.
## Increase maxiter to at least 48620 to perform exact estimation.
```

```r
# probability of treatment
probs <- genprobexact(data4.2$treatment)
# estimate the ATE
ate <- estate(data4.2$improvement, data4.2$treatment, prob = probs)

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

# generate potential outcomes under sharp null of no effect
Ys <- genouts(data4.2$improvement, data4.2$treatment, ate = 0)
```

```r
# 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] 0.06
##
## $two.tailed.p.value.abs
## [1] 0.067
##
## $greater.p.value
## [1] 0.03
##
## $lesser.p.value
## [1] 0.993
##
## $quantile
##  2.5% 97.5%
##    -1     1
##
## $sd
## [1] 0.4930517
##
## $exp.val
## [1] -0.007777778
```

By hand

```
library(randomizr)

observed_ate <-
  with(data4.2, mean(improvement[treatment == 1]) - mean(improvement[treatment == 0]))

simulated_ates <- rep(NA, sims)

for (i in 1:sims){
  data4.2$Z_sim <- complete_ra(N = 18, m = 9)
  simulated_ates[i] <- with(data4.2, mean(improvement[Z_sim == 1]) - mean(improvement[Z_sim == 0]))
}

p_two_tailed <- mean(abs(simulated_ates) >= abs(observed_ate))
p_upper <- mean(simulated_ates >= observed_ate)
p_lower <- mean(simulated_ates <= observed_ate)

p_two_tailed <- mean(abs(simulated_ates) >= abs(observed_ate))
p_upper <- mean(simulated_ates >= observed_ate)
p_lower <- mean(simulated_ates <= observed_ate)
c(observed_ate, p_two_tailed, p_upper, p_lower)
```

```
## [1] 1.000 0.090 0.048 0.990
```

```
hist(simulated_ates, breaks = 10)
abline(v = observed_ate, col = "red")
```