Gerber and Green (2012) Chapter 4 Problem 5

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

Chapter 4 Problem 5

Randomizations are said to be "restricted" when the set of all possible random allocations is narrowed to exclude allocations that have inadequate covariate balance. Suppose, for example, that the assignment of treatments (di) in Table 4.1 was conducted subject to the restriction that a regression of di on Xi (the pre-test) generates an F-statistic whose reported p-value is greater than 0.05. In other words, had the researcher found that the assigned di were significantly predicted by Xi, the random allocation would have been conducted again, until the di met this criterion.

(a) Conduct a series of random assignments in order to calculate the weighting variable wi; for units in the treatment group, this weight is defined as the inverse of the probability of being assigned to treatment, and for units in the control group, this weight is defined as the inverse of the probability of being assigned to control. See Table 4.2 for an example. Does wi appear to vary within the treatment group or within the control group?

SHOWN BELOW

(b) Use randomization inference to test the sharp null hypothesis that di has no effect on Yi by regressing Yi on di and comparing the estimate to the sampling distribution under the null hypothesis. Make sure that your sampling distribution includes only random allocations that satisfy the restriction mentioned above. If the probability of treatment varies from one subject to the next, estimate the ATE by weighting each observation by wi. Calculate the p-value and interpret the results.

SHOWN BELOW

(c) Use randomization inference to test the sharp null hypothesis that di has no effect on Yi by regressing Yi on di and Xi (weighting the observations by wi, if necessary) and comparing the estimate to the sampling distribution under the null hypothesis. Calculate the p-value, and interpret the results.

(d) Compare the sampling distributions under the null hypothesis in parts (b) and (c) to the sampling distributions obtained in exercises 4(d) and 4(e), which assumed that the randomization was unrestricted.

NOT SHOWN

```r
# Data from http://isps.yale.edu/FEDAI
library(haven)
data4.5 <- read_dta("datasets/4.5.dta")
# Number of sims the same for all three methods
sims <- 1000
```

Setup

All three methods will require a custom randomization function that only returns "acceptable" random assignments.

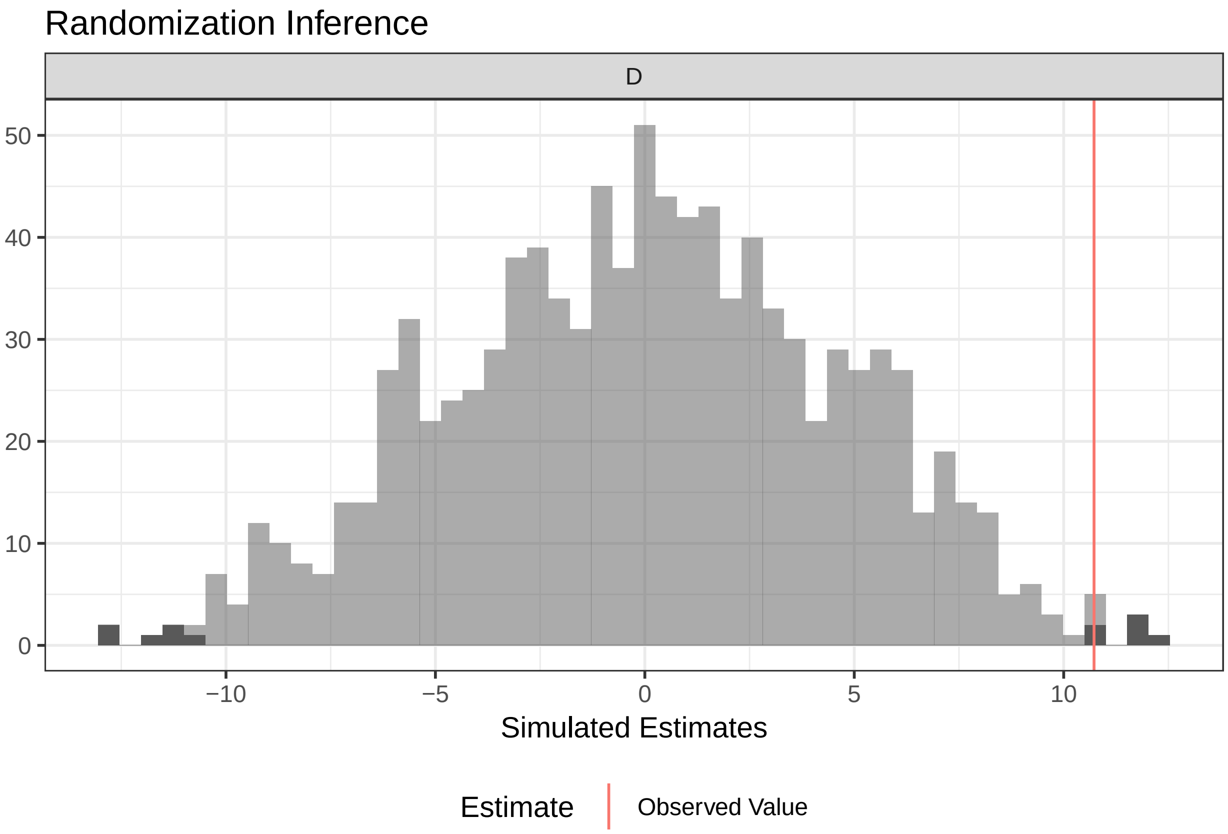
```r
custom_ra_function <- function() {
  bad_balance <- TRUE

  while (bad_balance) {
  data4.5$Z_sim <- complete_ra(nrow(data4.5))
  fit_sim <- lm(Z_sim ~ D + x, data = data4.5)

  # EXTRACT f stat
  f_stat <- summary(fit_sim)$fstatistic

  # calculate p
  p_val <- 1 - pf(q = f_stat[1],
  df1 = f_stat[2],
  df2 = f_stat[3])

    if (p_val > 0.05) {
  # if balance is "good", accept rand
      bad_balance <- FALSE
    }
  }
  return(data4.5$Z_sim)
}
```

In ri2

```r
library(ri2)
# Declare randomization procedure
permutation_matrix <- replicate(sims, custom_ra_function())
declaration <- declare_ra(permutation_matrix = permutation_matrix)

# Conduct Randomization Inference
ri2_out <- conduct_ri(Y ~ D,
                      declaration = declaration,
                      assignment = "D",
                      sharp_hypothesis = 0,
                      sims = sims,
                      data = data4.5)

summary(ri2_out)
```

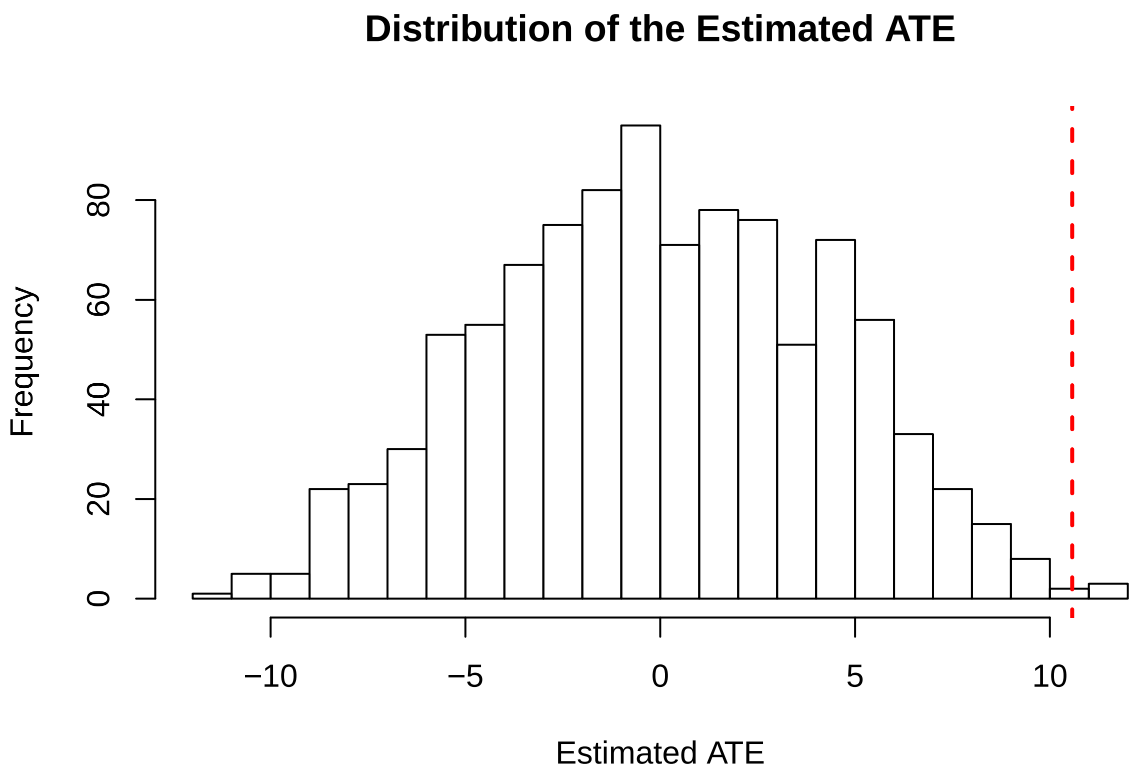
```
##   coefficient estimate two_tailed_p_value null_ci_lower null_ci_upper
## 1           D 10.72431              0.012      -9.09444      8.297476
```

```r
plot(ri2_out)
```



In ri

```
library(ri)

# all possible permutations
perms <- genperms.custom(numiter = sims, randfun = custom_ra_function)
# probability of treatment
probs <- genprob(perms)
# estimate the ATE
ate <- estate(data4.5$Y, data4.5$D, prob = probs)

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

# generate potential outcomes under sharp null of no effect
Ys <- genouts(data4.5$Y, data4.5$D, 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)
```



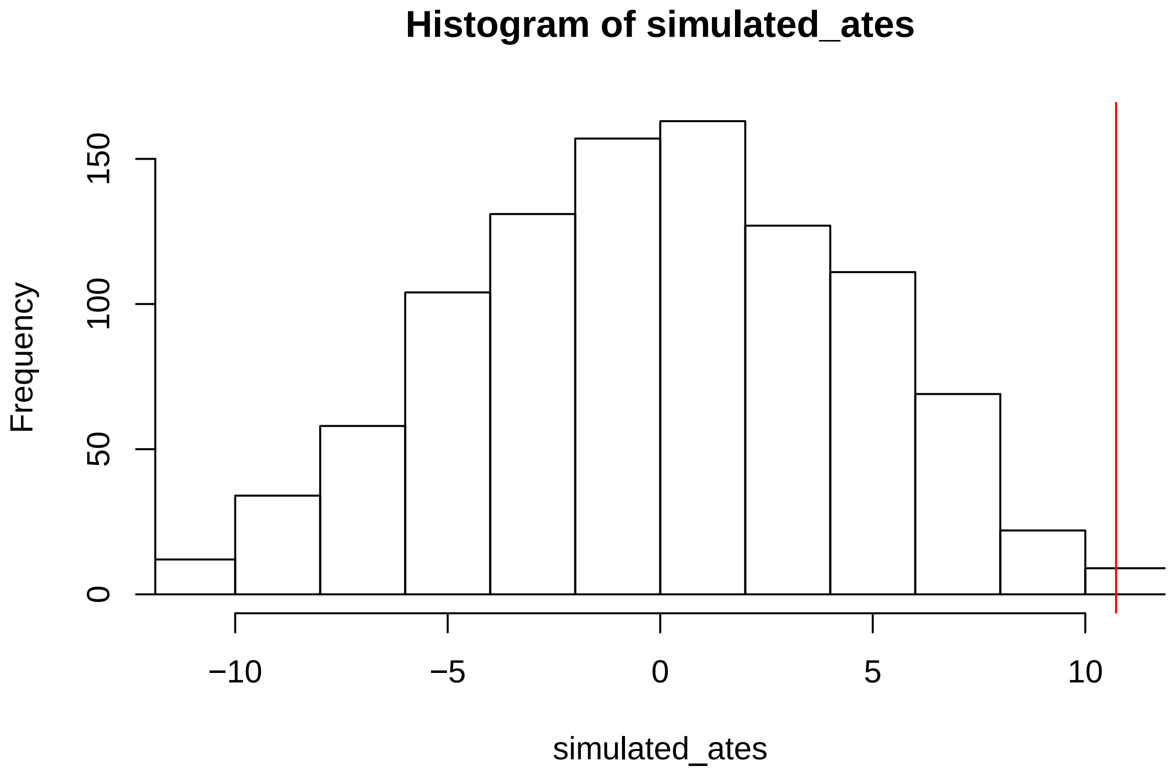
```
ri_out
```

```
## $two.tailed.p.value
## [1] 0.006
##
## $two.tailed.p.value.abs
## [1] 0.006
##
## $greater.p.value
## [1] 0.003
##
## $lesser.p.value
## [1] 0.997
##
## $quantile
##      2.5%     97.5%
## -8.316020  8.122009
##
## $sd
## [1] 4.350965
##
## $exp.val
## [1] 0.0004162978
```

By hand

```
library(randomizr)
prob_assignment <- rowMeans(permutation_matrix, na.rm = TRUE)
```

```r
data4.5$w <- with(data4.5, 1 / (D * prob_assignment + (1 - D) * (1 - prob_assignment)))
fit <- lm(Y ~ D, weights = w, data = data4.5)

observed_ate <- coef(fit)[2]

simulated_ates <- rep(NA, sims)

for (i in 1:sims){
  data4.5$Z_sim <- permutation_matrix[,i]
  data4.5$w_sim <- with(data4.5, 1 / (Z_sim * prob_assignment + (1 - Z_sim) * (1 - prob_assignment)))
  fit_sim <- lm(Y ~ Z_sim, weights = w_sim, data4.5)
  simulated_ates[i] <- coef(fit_sim)[2]
  }

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

hist(simulated_ates, breaks = 10, xlim = c(-11,11))
abline(v = observed_ate, col = "red")
```


Histogram of simulated_ates
Frequency
simulated_ates