Estimation, ATE, SE

Natalia Garbiras-Díaz

April 10, 2019

A simulation in R: sample mean as an unbiased estimator of the population mean

First, we will need to "create" a population (a study group)

```
population <- c(4, 5, 7, 12, 7, 8, 9, -3, 5, 8, 9, 3, 2, 3,

N <- length(population) # number of observations in the pop
N
```

```
[1] 23
```

```
pop_mean <- mean(population) # population mean
pop_mean
```

```
[1] 5.869565
```

We will draw several random samples of 8 observations (m) each *without* replacement

```
set.seed(12345)
s1 <- sample(population, size = 8, replace = FALSE)

s2 <- sample(population, size = 8, replace = FALSE)

s3 <- sample(population, size = 8, replace = FALSE)

s4 <- sample(population, size = 8, replace = FALSE)

samples <- rbind(s1, s2, s3, s4)

samples
```

```
   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
s1    3    6    6    9    5    7    8    9
s2    8   10   -3    9    8    4    3   12
s3   -3    3    7    5    3    2    8    9
s4    6    2    5    7    5   12   -3    9
```

Remember the population mean: 5.8695652

And the means of the four samples

```
apply(samples, MARGIN = 1, FUN = mean) # apply function to
```

```
   s1    s2    s3    s4
6.625 6.375 4.250 5.375
```

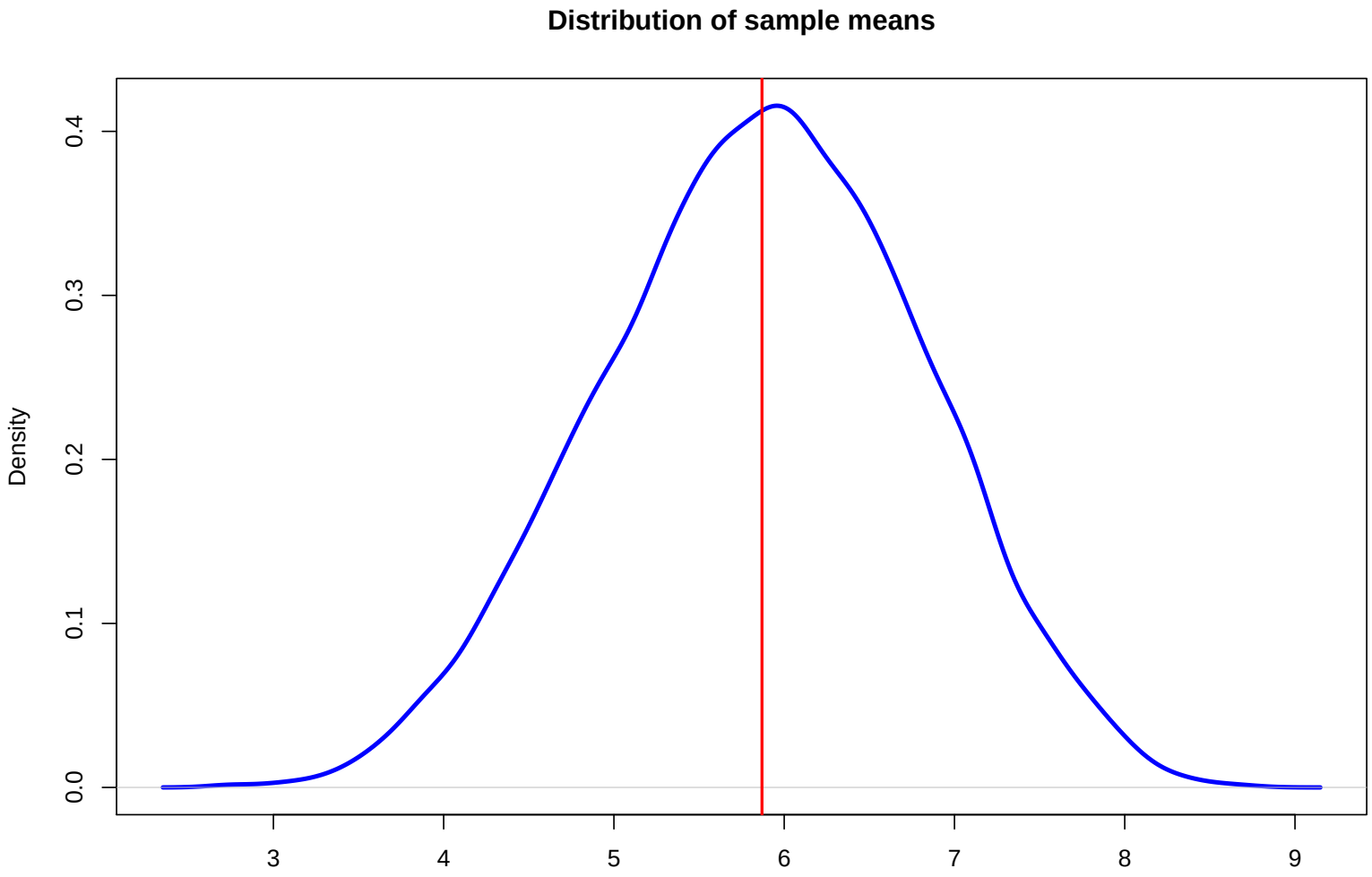
By chance each given sample mean may be a little higher or lower than the population mean.

We can use R to show that the sample mean is an unbiased estimator of the population mean.

For this, we will write a *simulation*. We will repeat the sampling process $10,000$ times.

```
sample_mean <- NA

for (i in 1:10000) {
  sample <- sample(population, size = 8, replace = FALSE)
  sample_mean[i] <- mean(sample)
}
```

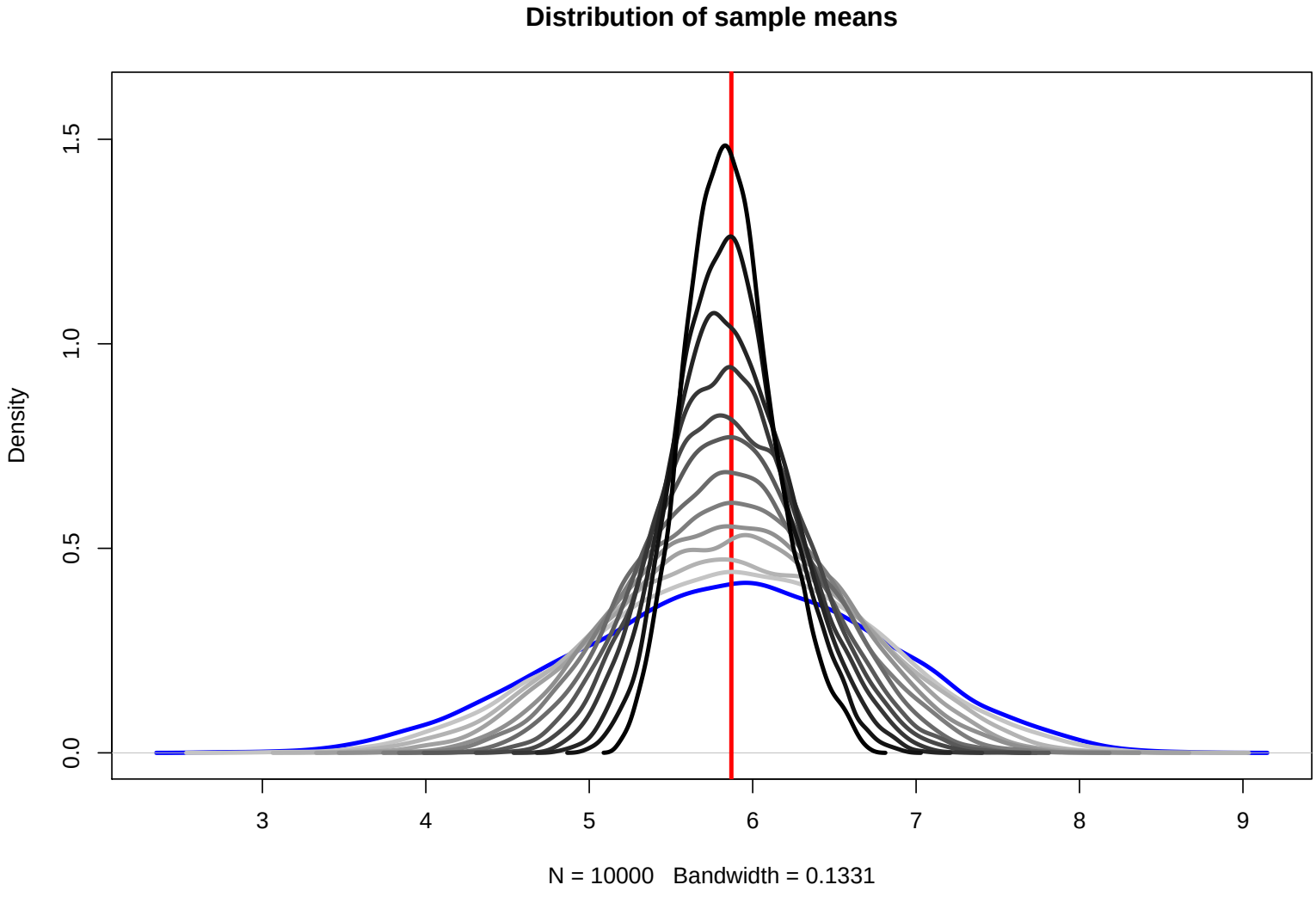
```r
par(mfrow = c(1, 1))
plot(density(sample_mean),
  col = "blue", lwd = 3,
  main = "Distribution of sample means"
)
abline(v = pop_mean, col = "red", lwd = 2)
```



Let's now look at the distribution of the sample mean as m gets closer to N.

So far, $m = 8$. We now need a new simulation that adds a new step: we need to vary the size of m. (Remember our population size, N, is 23)

```r
rep <- 10000

# The first loop varies m
for (m in 9:20) {
  sample_mean <- NA # creating an object to store the resu

  # The second loop goes through the 10,000 simulations
  for (i in 1:rep) {

    # we first get a random sample of size m from the popu
    sample <- sample(population, size = m, replace = FALSE)
    # and then calculate and store the sample mean
    sample_mean[i] <- mean(sample)
  }

  # finally, we plot the distribution of the 10,000 sample
  lines(density(sample_mean),
    lwd = 3,
    # note that this next line of code varies the color of
    # so that we can distinguish the different distributio
```


Distribution of sample means
Density
0.0
0.5
1.0
1.5
3
4
5
6
7
8
9
N = 10000 Bandwidth = 0.1331

The variance of the sample mean

The standard deviation of the sampling distribution gives us a measure of uncertainty about the mean:

```
var_sample_mean <- sum((sample_mean - mean(sample_mean))^2)
se_sample_mean <- sqrt(var_sample_mean)
se_sample_mean
```

```
[1] 0.9367243
```

Now, we can calculate this because we created our own population. This is not often the case (e.g., experiments)...

Remember the formula for the variance of the sample mean for the treatment group is:

$$\text{Var}(Y^T) = \frac{\sigma^2}{m}$$

We do not know σ^2, we can estimate this quantity with the variance of the assigned-to-treatment sample by:

$$\hat{\sigma}^2 = (\frac{1}{m-1}) \sum_{i=1}^{m} (Y_i - \bar{Y}^T)^2$$

Same with the variance of the sample mean for those units assigned to control.

2. Estimation of the ATE

We can write a function to estimate the ATE (or simply use the built-in function `t.test`).

```r
diff_means <- function(y, x) {

  # Calculating difference in means
  mean1 <- mean(y[x == 1], na.rm = T)
  mean0 <- mean(y[x == 0], na.rm = T)
  diff <- mean1 - mean0

  # Calculating number of observations
  N <- length(na.omit(y))

  # Preparing output
  res <- c(mean1, mean0, diff, N)
  names(res) <- c("Mean 1", "Mean 0", "Difference", "N")

  return(c(res))
}
```

To try our function, we will use the small dataset in Gerber & Green (2012)

```
gg_data <- as.data.frame(cbind(
  c(10, 15, 20, 20, 10, 15, 15),
  c(15, 15, 30, 15, 20, 15, 30)
))
names(gg_data) <- c("Y_i0", "Y_i1")
```

We will need to "create" a treatment vector...

```
# let's fix m=3 (units in the treatment group)
treat <- c(1, 1, 1, 0, 0, 0, 0)
gg_data$treat <- sample(treat, 7, replace = F)
gg_data$treat
```

```
[1] 1 1 0 0 1 0 0
```

...and a column with the "observed" outcomes

```
gg_data$observed <- ifelse(gg_data$treat == 1, gg_data$Y_i
## save(gg_data, file="gg_data.RData")
```

Let's see how the complete data set looks now:

```
head(gg_data)
```

```
  Y_i0 Y_i1 treat observed
1   10   15     1       15
2   15   15     1       15
3   20   30     0       20
4   20   15     0       20
5   10   20     1       20
6   15   15     0       15
```

```r
# mean of the treatment group
mean(gg_data$observed[gg_data$treat == 1])
```

```
[1] 16.66667
```

```r
# mean of the control group
mean(gg_data$observed[gg_data$treat == 0])
```

```
[1] 17.5
```

```r
# difference of means
mean(gg_data$observed[gg_data$treat == 1]) - mean(gg_data$o
```

```
[1] -0.8333333
```

```r
# with our function
diff_means(gg_data$observed, gg_data$treat)
```

```
    Mean 1     Mean 0 Difference          N
16.6666667 17.5000000 -0.8333333  7.0000000
```

Now, we can also estimate the same quantity using a regression:

```
lm_robust(observed ~ treat, data = gg_data)
```

```
              Estimate Std. Error    t value     Pr(>|t|)
(Intercept) 17.5000000   1.443376 12.1243557 6.743204e-05
treat       -0.8333333   2.204793 -0.3779645 7.209712e-01
              CI Upper DF
(Intercept) 21.210315  5
treat        4.834267  5
```

But notice that we are not relying on the assumptions of OLS regression. This is just math... the way β is estimated.

How can we get a distribution of the difference of means?

We can do this with a simulations. For each simulation,

- First: We will need to "create" a random treatment vector and generate the column with the associated observed outcomes.

How can we get a distribution of the difference of means?

We can do this with a simulations. For each simulation,

- First: We will need to "create" a random treatment vector and generate the column with the associated observed outcomes.
- Second: We will have to calculate the difference between the treatment and control means (by hand or using our new function).

```r
# 1.
gg_data$treat <- sample(treat, 7, replace = F)
gg_data$observed <- ifelse(gg_data$treat == 1, gg_data$Y_i
```

```r
# 2.
diff_means(gg_data$observed, gg_data$treat)
```

```
    Mean 1      Mean 0 Difference           N
     20.00       16.25       3.75        7.00
```

```r
# we should store this! so,
dm <- diff_means(gg_data$observed, gg_data$treat)
dm
```

```
    Mean 1      Mean 0 Difference           N
     20.00       16.25       3.75        7.00
```

```r
# but we only want the third element!
dm <- diff_means(gg_data$observed, gg_data$treat)[3]
dm
```

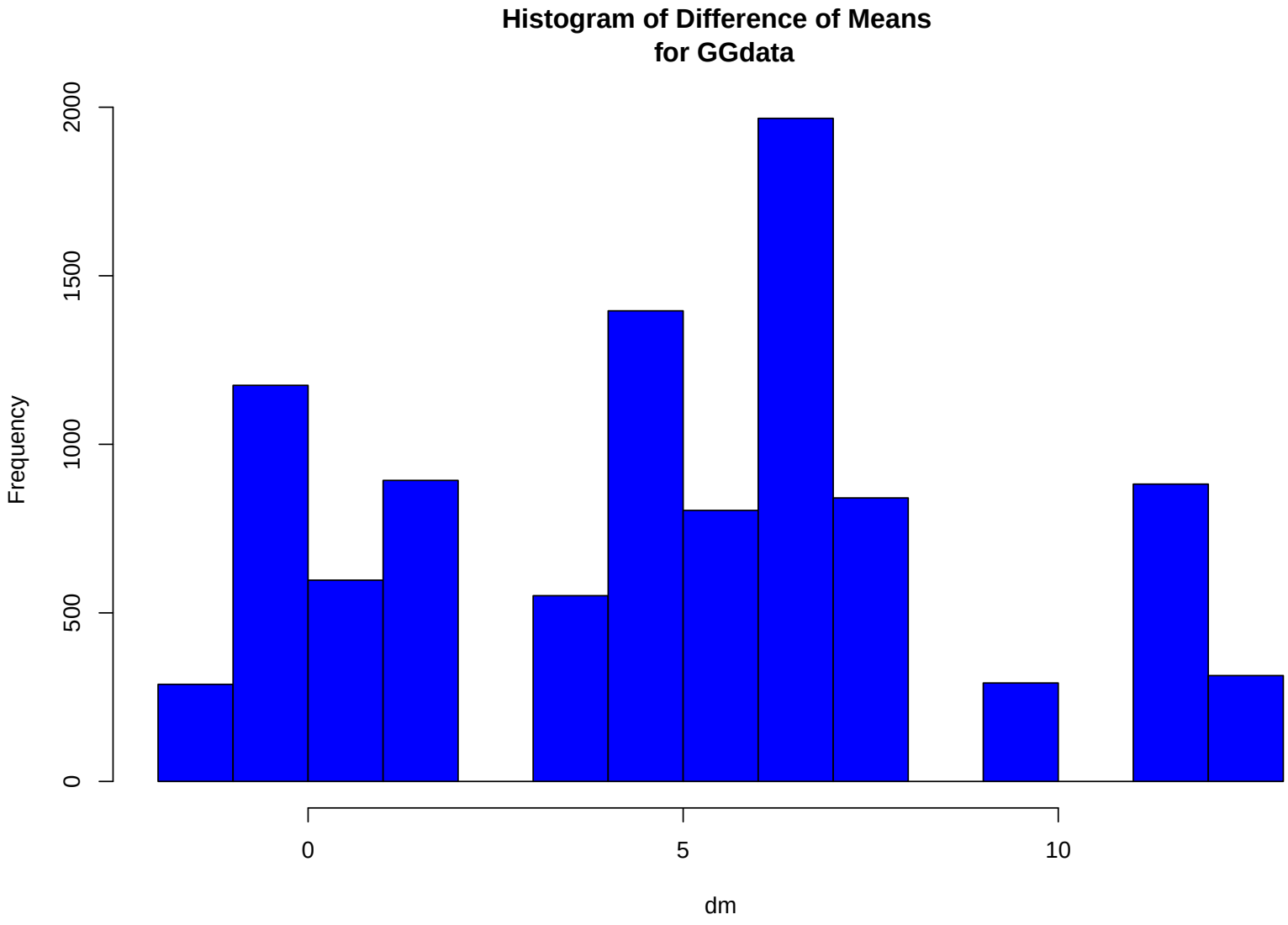
```
Difference
```

Now let's put this in a loop that allows us to repeat the process $10,000$ times (and saves the dom for each)...

```
dm <- NA # creating a placeholder to store all our doms...

for (i in 1:10000) {

  # 1.
  gg_data$treat_sim <- sample(treat, 7, replace = F)
  gg_data$observed <- ifelse(gg_data$treat_sim == 1, gg_dat

  # 2.
  dm[i] <- diff_means(gg_data$observed, gg_data$treat_sim)
}
```

Finally, let's plot the distribution

```
hist(dm, col = "blue", main = "Histogram of Difference of M
```



3. Standard Error for the ATE

1. Standard error for the difference in means

1. Standard error for the difference in means

- The difference in means is an unbiased estimator of the true ATE. However, by chance, in some realizations of our sample that estimate might be off the true ATE.
- The SE tells us the likely size of the amount off.

A conservative formula for the $\widehat{SE}$ for the $\widehat{ATE}$

$$\widehat{SE}(\widehat{ATE}) = \sqrt{\frac{\widehat{\text{Var}}(Y_i(0))}{N-m} + \frac{\widehat{\text{Var}}(Y_i(1))}{m}}$$

We are going to estimate the SE for the difference in means using the same data.

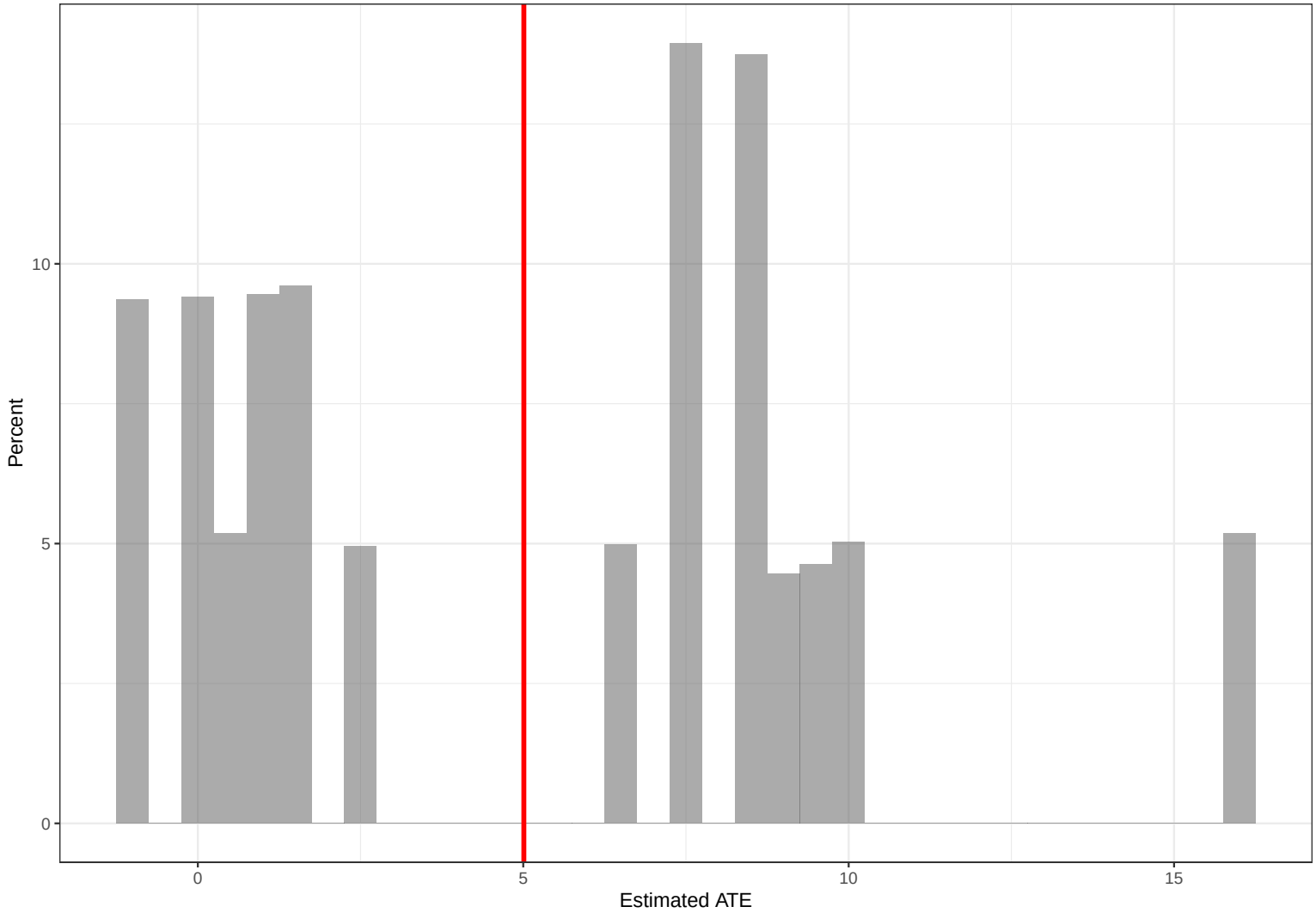
```
true_ate <- mean(gg_data$Y_i1) - mean(gg_data$Y_i0)
true_ate
```

```
[1] 5
```

```
est_ate <- mean(gg_data$observed[gg_data$treat == 1]) - mea
est_ate
```

```
[1] 10
```

```r
# generating empty dataframe to put the results
ate <- as.data.frame(matrix(NA, 10000, 2))
names(ate) <- c("estimated_ate", "estimated_se_ate")

# sampling
for (i in 1:10000) {

  # generating treatment vector for this replicate
  gg_data$treat_sim <- 0
  gg_data$treat_sim[sample(1:7, 2, replace = F)] <- 1

  treat_mean <- mean(gg_data$Y_i1[gg_data$treat_sim == 1])
  treat_var <- var(gg_data$Y_i1[gg_data$treat_sim == 1])

  control_mean <- mean(gg_data$Y_i0[gg_data$treat_sim == 0]
  control_var <- var(gg_data$Y_i0[gg_data$treat_sim == 0])

  ate[i, 1] <- treat_mean - control_mean
  ate[i, 2] <- sqrt(treat_var / 2 + control_var / 5)
}
```



- How could we use this graph to get the SE of the estimated ATE?

```r
# The SE of the estimated ATE is the standard deviation of
se_sampling <- sd(ate[, 1])
se_sampling
```

```
[1] 4.650395
```

- But in any given experiment, we don't have the sampling distribution. Instead, we can estimate the SE (using the conservative formula)

```
treat_var <- var(gg_data$Y_i1[gg_data$treat == 1])
control_var <- var(gg_data$Y_i0[gg_data$treat == 0])
est_se_cons <- sqrt(treat_var / 2 + control_var / 5)
est_se_cons
```

```
[1] 6.390097
```

```r
# Comparing the true standard error to the conservative fo
print(c(se_sampling, est_se_cons))
```

```
[1] 4.650395 6.390097
```

4. Blocked randomized experiments

Let's use the data from yesterday, with the example of water sanitazing devices. We had

```
table(data$complete.rand, data$female)
```

```
     0   1
  0 201 199
  1  99 101
```

```
# Block randomization using randomizr

data$block.rand <- block_ra(blocks = data$female, prob_eac

table(data$block.rand, data$female)
```

```
     0   1
  0 225 225
  1  75  75
```

```
data$block.obs <- with(data, Y1 * block.rand + Y0 * (1 - bl
```

How can we analyze these data?

- When analyzing data from blocked randomized experiments, we may ask different questions:
  - For instance, what is the ATE among women? Does the ATE vary by gender?
  - We may, instead, be interested in the overall ATE.

- Since we conducted a complete RA at the block level, we can estimate the ATE for each one of the groups created by our blocking variables

```
# Recall
effect.male <- -2
effect.female <- -7
```

```
female <- filter(data, data$female == 1)
dom_fem <- mean(female$block.obs[female$block.rand == 1]) -
dom_fem
```

```
[1] -6.888889
```

```
male <- filter(data, data$female == 0)
dom_male <- mean(male$block.obs[male$block.rand == 1]) - me
dom_male
```

```
[1] -2.408889
```

- Now, we can also estimate the overall ATE by estimating block-level ATEs.
- We then need to ask, how do we want to weight each block-level ATE in order to obtain the overal ATE?
- One way is to weight by the block size:

```
block_female <- sum(data$female == 1) / length(data$ID)
block_male <- sum(data$female == 0) / length(data$ID)

ate_overall <- block_female * dom_fem + block_male * dom_ma
ate_overall
```

```
[1] -4.648889
```

```
var_fem_treat <- var(data$block.obs[data$block.rand == 1 &
var_fem_control <- var(data$block.obs[data$block.rand == 0
var_male_treat <- var(data$block.obs[data$block.rand == 1 &
var_male_control <- var(data$block.obs[data$block.rand == (

se_est_fem <- sqrt(var_fem_control / sum(data$block.rand ==
se_est_male <- sqrt(var_male_control / sum(data$block.rand
```

We could have done this using the `difference_in_means` command from `estimatr`

```
difference_in_means(block.obs ~ block.rand, blocks = female
```

```
Design:  Blocked
             Estimate Std. Error   t value     Pr(>|t|) CI L
block.rand -4.648889   1.035855 -4.487971 8.632315e-06 -6.6
             DF
block.rand 596
```

```
(c(ate_overall, se_est_all))
```

```
[1] -4.648889  1.035855
```

Imagine we forget that we blocked:

```
lm_robust(block.obs ~ block.rand, data = data)
```

```
             Estimate Std. Error  t value      Pr(>|t|)  CI
(Intercept) 19.062222  0.5233487 36.42356 6.377196e-154 18.
block.rand  -4.648889  1.0594816 -4.38789  1.352959e-05 -6.
             CI Upper  DF
(Intercept) 20.090047 598
block.rand  -2.568132 598
```

```
(c(ate_overall, se_est_all))
```

```
[1] -4.648889  1.035855
```

We could also get to this quantity using a regression with block dummies (Least Squares Dummy Variables) or with weights (IPW):

```
table(data$block.rand, data$female)
```

```
     0   1
  0 225 225
  1  75  75
```

```
# Block dummies (LSDV):
# the weigths used here are: p_j * (1 - p_j) * n_j)
lm_robust(block.obs ~ block.rand + female, data = data)
```

```
             Estimate Std. Error   t value      Pr(>|t|)  (
(Intercept) 20.235556  0.7262745 27.862133 4.391278e-110 18
block.rand  -4.648889  1.0433060 -4.455921  9.976628e-06 -6
female      -2.346667  0.9058405 -2.590596  9.815024e-03 -4
              CI Upper  DF
(Intercept) 21.6619191 597
block.rand  -2.5998927 597
female      -0.5676453 597
```