Speaking Science

W241 Final Project

Haerang Lee, Aris F, Mumin Khan

March 17th, 2020

Contents


```
read_data <- function(data_file) {
  d <- fread(data_file, )
  d <- na.omit(d)

  d$q1_correct <- as.factor(d$q1_correct)
  d$q2_correct <- as.factor(d$q2_correct)
  d$q3_correct <- as.factor(d$q3_correct)
  d$q4_correct <- as.factor(d$q4_correct)
  d$q5_correct <- as.factor(d$q5_correct)
  d$q6_correct <- as.factor(d$q6_correct)
  d$treatment <- as.factor(d$treatment)

  return(d)
}


data_file <- 'speaking_science_data_03-24_clean.csv'
d <- read_data(data_file)
head(d)
```

```
##         start_date        end_date     ip_address duration_in_seconds
## 1: 3/23/2020 21:00 3/23/2020 21:03 68.105.189.229                  185
## 2: 3/23/2020 20:59 3/23/2020 21:03  76.176.54.192                  284
## 3: 3/23/2020 21:04 3/23/2020 21:06     71.6.87.50                  163
## 4: 3/23/2020 21:02 3/23/2020 21:06  72.216.72.106                  275
## 5: 3/23/2020 21:04 3/23/2020 21:07  196.17.67.191                  193
## 6: 3/23/2020 21:05 3/23/2020 21:07 104.247.222.40                  127
##      recorded_date       response_id latitude longitude       mturk_id
## 1: 3/23/2020 21:03 R_1jqhwROmmrLPaoy 36.05881 -115.3104 A12ATVBE1I4567
## 2: 3/23/2020 21:03 R_2ASHk9ILabLrZCB 33.02870 -117.0846  A9OOV3976AFYF
## 3: 3/23/2020 21:06 R_3kbIZqjBaOkb1pG 37.76880 -122.2620 A1OROXYXMV5MBO
## 4: 3/23/2020 21:06 R_sRUrOCfBuUjTAuB 32.89461 -111.7493  A83OLM1ZQCO83
## 5: 3/23/2020 21:07 R_dcfqmDLDif7R5Pr 34.05440 -118.2440 A2XCEMBRPHIWEG
```

```
## 6: 3/23/2020 21:07 R_wLFYgmJWP3ZB51n 38.97630  -87.3667 A1PQEHT7M68ZK3
##    browser_type browser_version      browser_os browser_resolution
## 1:       Chrome   80.0.3987.149 Windows NT 10.0           1536x864
## 2:       Chrome   79.0.3945.136     Android 7.0            360x640
## 3:       Chrome   80.0.3987.132       Macintosh           1440x900
## 4:       Chrome   80.0.3987.149 Windows NT 10.0           1536x864
## 5:       Chrome   80.0.3987.149  Windows NT 6.1           1366x768
## 6:       Chrome   80.0.3987.149 Windows NT 10.0           1364x768
##    time_read_intro time_read_article credibility importance q1 q1_correct q2
## 1:           7.588            29.649           5          4  3          1  5
## 2:          13.345           166.381           6          7  1          0  3
## 3:          18.607            47.366           6          6  2          0  1
## 4:          10.128           163.601           7          7  3          1  1
## 5:           3.328            23.648           7          7  2          0  1
## 6:          16.888            19.636           5          6  3          1  3
##    q2_correct q3 q3_correct q4 q4_correct q5 q5_correct q6 q6_correct
## 1:          0  3          1  1          0  4          1  1          1
## 2:          1  1          0  4          0  4          1  1          1
## 3:          0  1          0  2          0  1          0  1          1
## 4:          0  3          1  1          0  3          0  1          1
## 5:          0  2          0  4          0  2          0  2          0
## 6:          1  4          0  1          0  1          0  3          0
##    questions_correct time_answering_questions donation time_donation
## 1:                 4                 125.852        1         4.044
## 2:                 3                  67.334        1         7.657
## 3:                 1                  49.709       50         7.145
## 4:                 3                  72.550        0        14.611
## 5:                 0                  70.295        2        68.819
## 6:                 2                  34.879       20        15.105
##           city state   zip treatment
## 1:   Las Vegas    NV 89113         0
## 2:   San Diego    CA 92127         0
## 3:     Vallejo    CA 94589         1
## 4: Casa Grande    AZ 85122         1
## 5: Los Angeles    CA 90009         1
## 6: Los Angeles    CA 90004         1
```

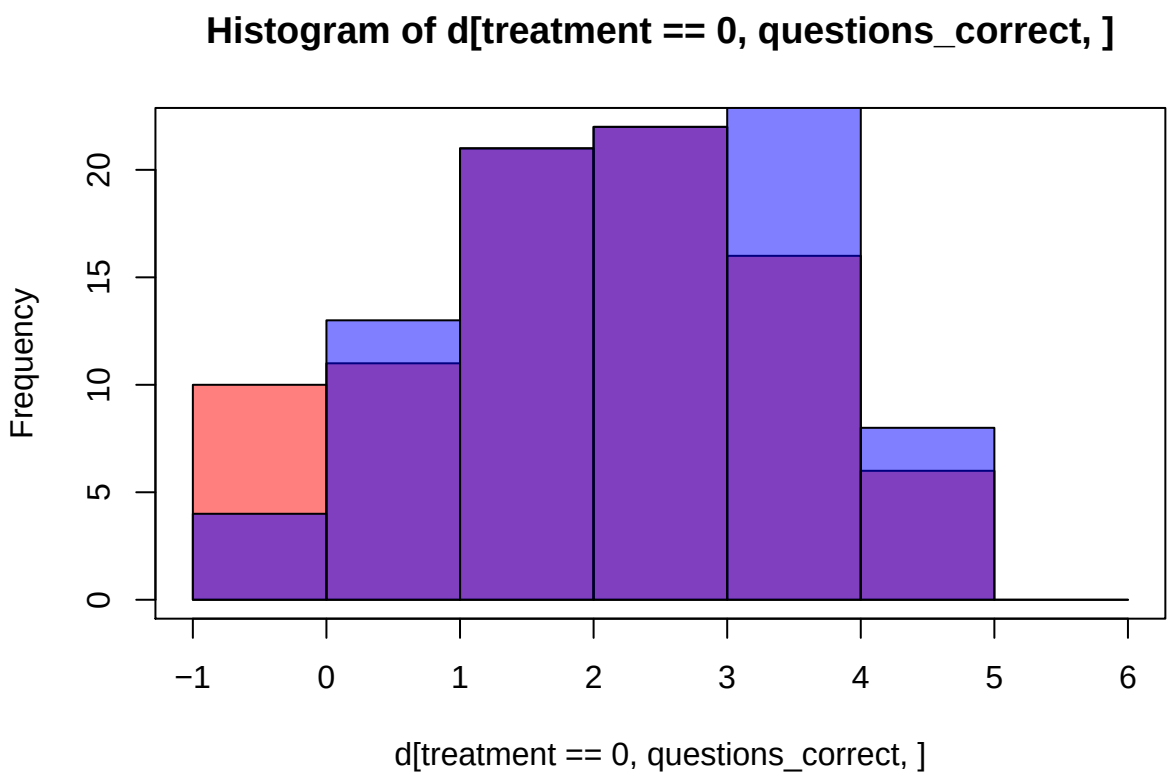
Simple Linear Regression

```
mod <- lm(questions_correct ~ treatment, data=d)
stargazer(mod, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                           questions_correct
## ------------------------------------------------
## treatment1                      0.303
##                                (0.207)
##
## Constant                      2.477***
##                                (0.148)
```

```
##
## ------------------------------------------------
## Observations                      177
## R2                               0.012
## Adjusted R2                      0.007
## Residual Std. Error       1.375 (df = 175)
## F Statistic              2.153 (df = 1; 175)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
hist(d[treatment == 0, questions_correct,], col=rgb(1,0,0,0.5), breaks=seq(-1,6, by=1))
hist(d[treatment == 1, questions_correct,], col=rgb(0,0,1,0.5), breaks=seq(-1,6, by=1), add = T)
box()
```

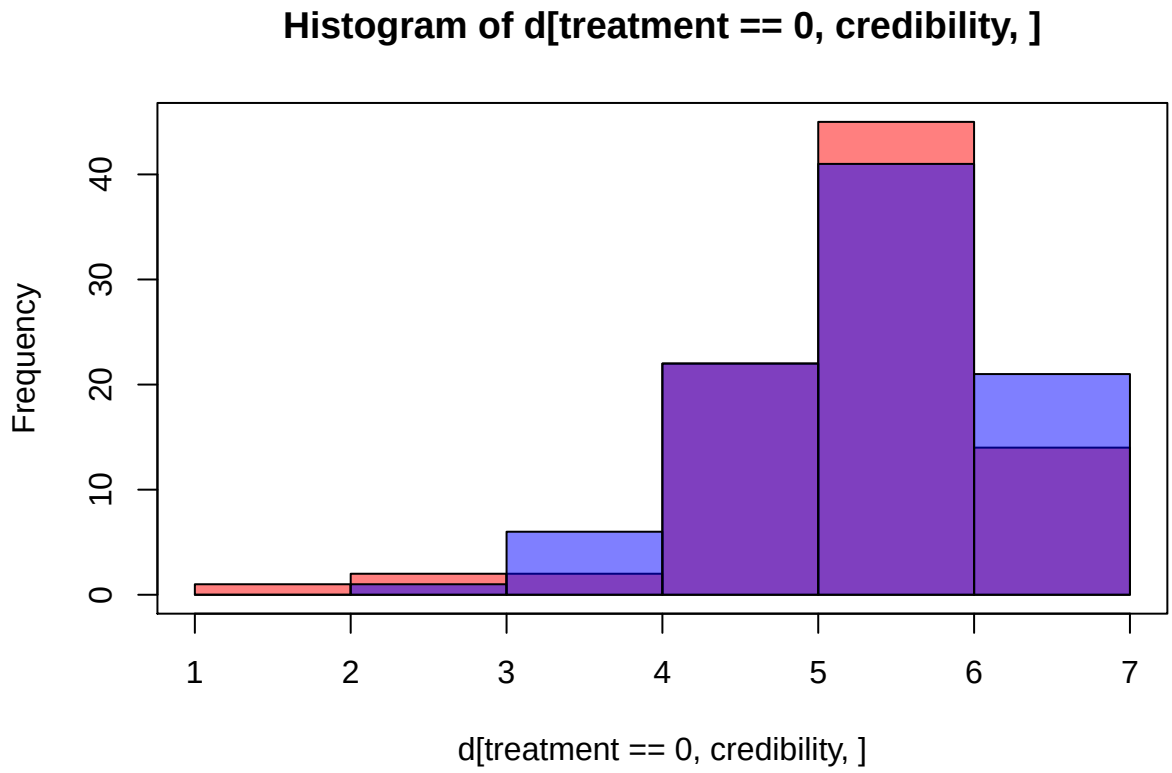



```
#Non parametric test
mod <- lm(credibility ~ treatment, data=d)
summary(mod)
```

```
##
## Call:
## lm(formula = credibility ~ treatment, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7442 -0.7442  0.1758  0.2558  1.2558
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.74419    0.09834  58.409   <2e-16 ***
## treatment1   0.07999    0.13716   0.583    0.561
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.912 on 175 degrees of freedom
## Multiple R-squared:  0.00194,    Adjusted R-squared:  -0.003763
## F-statistic: 0.3401 on 1 and 175 DF,  p-value: 0.5605
```

```
hist(d[treatment == 0, credibility,], col=rgb(1,0,0,0.5), breaks=seq(1,7, by=1))
hist(d[treatment == 1, credibility,], col=rgb(0,0,1,0.5), breaks=seq(1,7, by=1), add = T)
box()
```

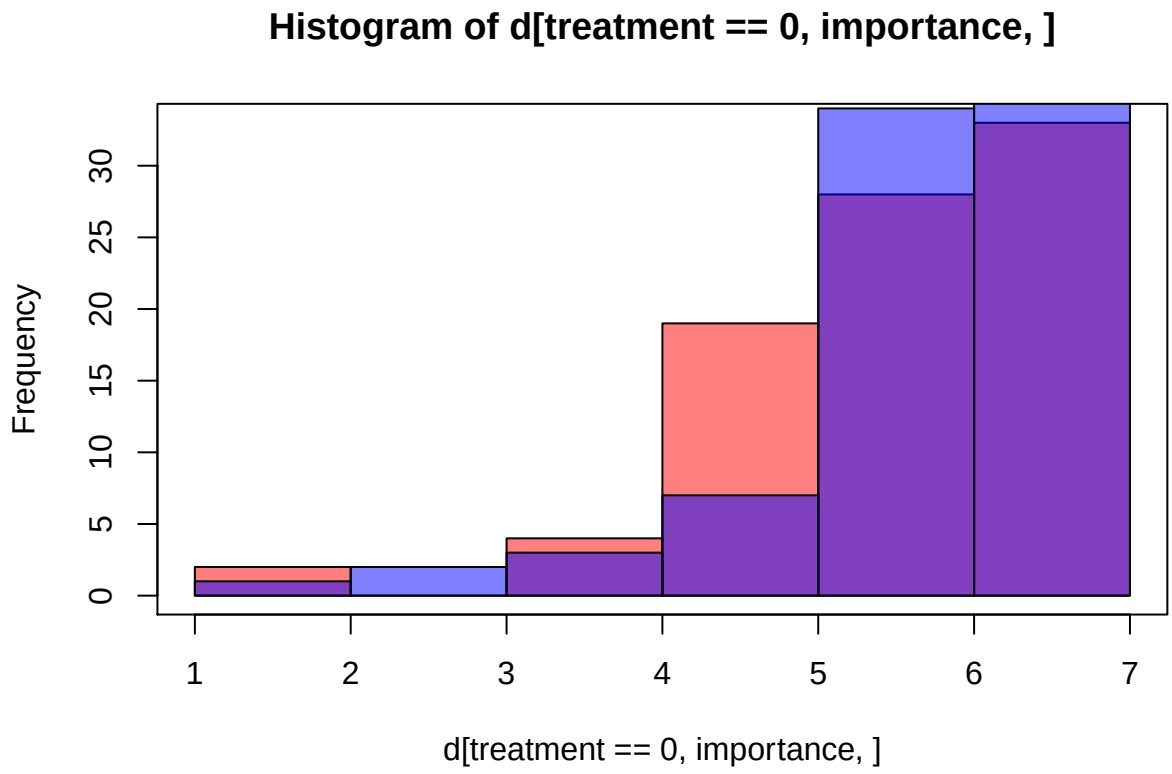



```
mod <- lm(importance ~ treatment, data=d)
stargazer(mod, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                               importance
## -----------------------------------------------
## treatment1                      0.266
##                                (0.168)
##
## Constant                       5.953***
```

```
##                                (0.121)
##
## ------------------------------------------------
## Observations                     177
## R2                              0.014
## Adjusted R2                     0.008
## Residual Std. Error       1.120 (df = 175)
## F Statistic             2.501 (df = 1; 175)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
hist(d[treatment == 0, importance,], col=rgb(1,0,0,0.5), breaks=seq(1,7, by=1))
hist(d[treatment == 1, importance,], col=rgb(0,0,1,0.5), breaks=seq(1,7, by=1), add = T)
box()
```

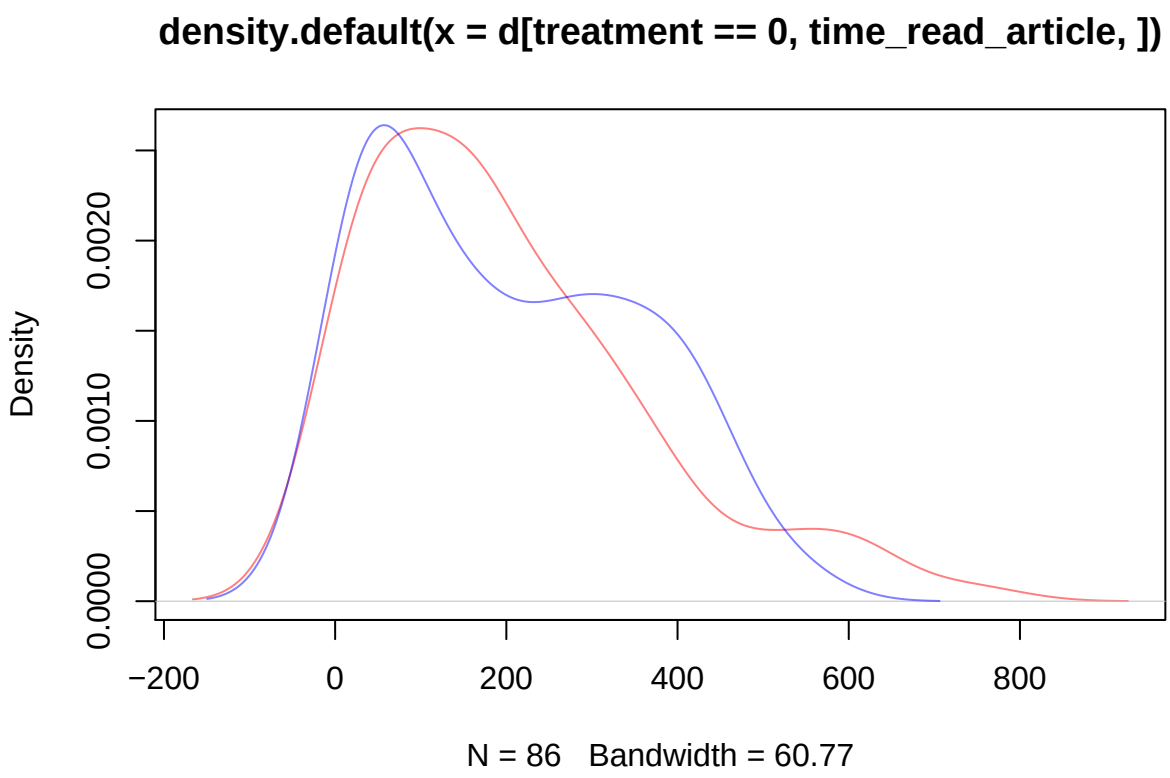



```
mod <- lm(time_read_article ~ treatment, data=d)
summary(mod)
```

```
##
## Call:
## lm(formula = time_read_article ~ treatment, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -186.16 -140.60  -35.82  107.43  541.70
##
## Coefficients:
```

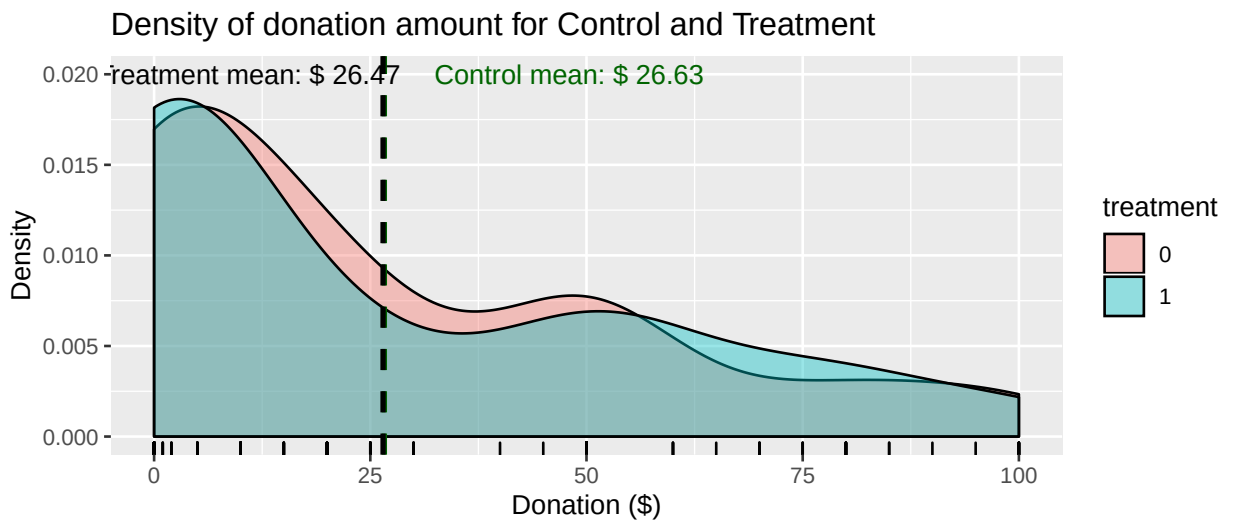
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 202.1964    17.1446  11.794   <2e-16 ***
## treatment1   -0.6931    23.9108  -0.029    0.977
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 159 on 175 degrees of freedom
## Multiple R-squared:  4.801e-06,  Adjusted R-squared:  -0.005709
## F-statistic: 0.0008403 on 1 and 175 DF,  p-value: 0.9769
```

```r
d1 <- density(d[treatment == 0, time_read_article,])
d2 <- density(d[treatment == 1, time_read_article,])

plot(d1, col=rgb(1,0,0,0.5))
lines(d2, col=rgb(0,0,1,0.5))
```




```r
control_mean_donation <- mean(d[treatment == 0, donation])
treat_mean_donation <- mean(d[treatment == 1, donation])

ggplot(d, aes(x = donation, fill = treatment)) +
  geom_density(alpha = 0.4) +
  geom_vline(aes(xintercept = control_mean_donation), color = "darkgreen", linetype = "dashed", size =
  geom_vline(aes(xintercept = treat_mean_donation), color = "black", linetype = "dashed", size = 1) +
  annotate("text", x = 48, y = 0.02, label = paste("Control mean: $", round(control_mean_donation, 2)),
  annotate("text", x = 11, y = 0.02, label = paste("Treatment mean: $", round(treat_mean_donation, 2)),
  ggtitle("Density of donation amount for Control and Treatment") + geom_rug() +
  ylab("Density") +
```

```
  xlab("Donation ($)")
```




```
wilcox.test(d[treatment == 0, donation], d[treatment == 1, donation])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d[treatment == 0, donation] and d[treatment == 1, donation]
## W = 4101.5, p-value = 0.5728
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(d[treatment == 0, importance], d[treatment == 1, importance])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d[treatment == 0, importance] and d[treatment == 1, importance]
## W = 3316.5, p-value = 0.06096
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(d[treatment == 0, credibility], d[treatment == 1, credibility])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  d[treatment == 0, credibility] and d[treatment == 1, credibility]
## W = 3759.5, p-value = 0.6286
## alternative hypothesis: true location shift is not equal to 0
```

Randomization Inference

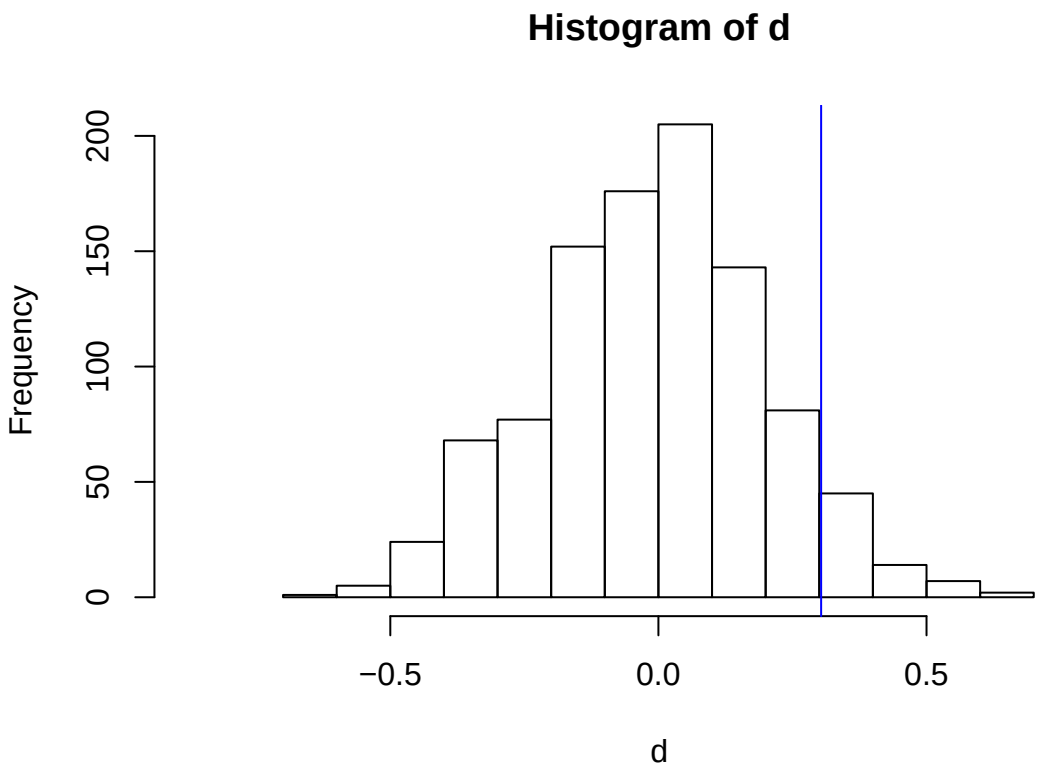
Testing the sharp null hypothesis that the treatment has no effect for anyone.

```
# Actual ATE's
ate_questions_correct <- d[, .('group_mean' = mean(questions_correct)), by=treatment][, diff(group_mean
ate_credibility       <- d[, .('group_mean' = mean(credibility)),       by=treatment][, diff(group_mean
ate_importance        <- d[, .('group_mean' = mean(importance)),        by=treatment][, diff(group_mean
ate_time_read_article <- d[, .('group_mean' = mean(time_read_article)), by=treatment][, diff(group_mean
```

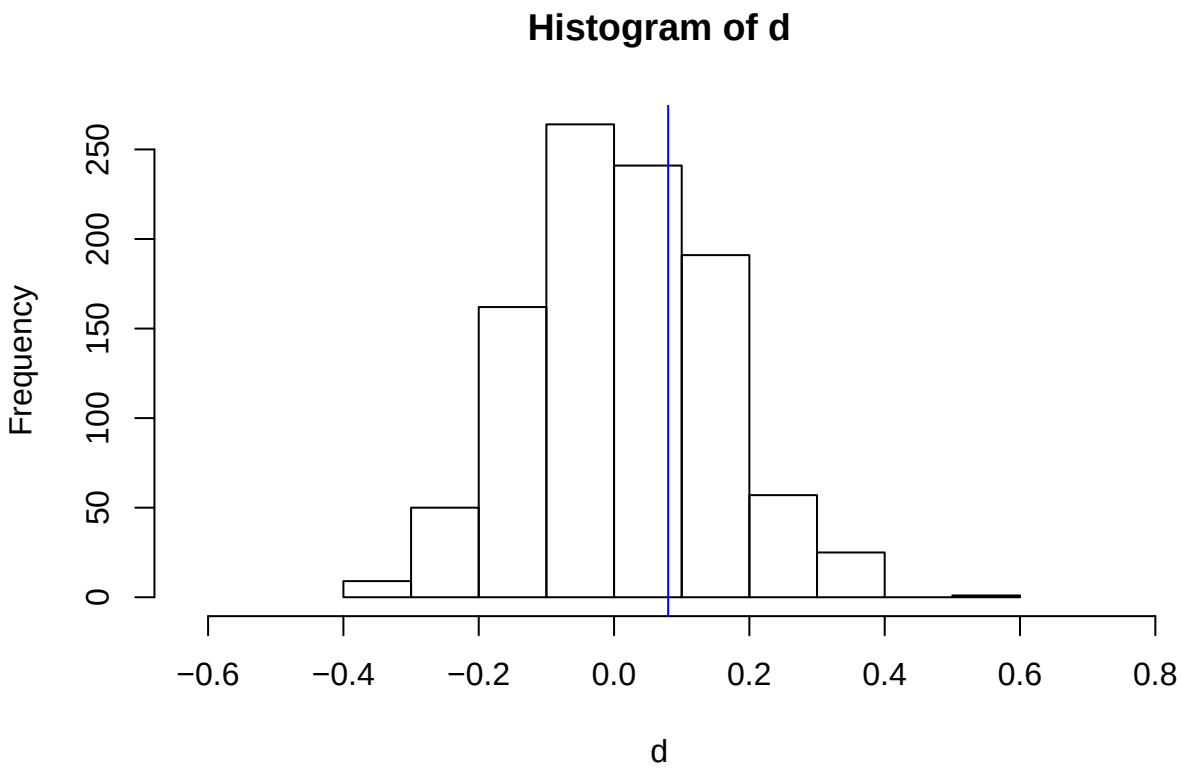
```r
ate_donation          <- d[, .('group_mean' = mean(donation)),          by=treatment][, diff(group_mean
```

```r
n <- 1000

# Initialize randomization inference
ri_questions_correct <- rep(NA, n)
ri_credibility       <- rep(NA, n)
ri_importance        <- rep(NA, n)
ri_time_read_article <- rep(NA, n)
ri_donation          <- rep(NA, n)

for(i in 1:n){
  d_ri <- copy(d)
  d_ri$treatment <- sample(d_ri$treatment)

  # Is there any way to do this with a loop and col.names(ri)? I hate R...
  ri_questions_correct[i] <- d_ri[, .('group_mean' = mean(questions_correct)), by=treatment][, diff(grou
  ri_credibility[i]       <- d_ri[, .('group_mean' = mean(credibility)),       by=treatment][, diff(grou
  ri_importance[i]        <- d_ri[, .('group_mean' = mean(importance)),        by=treatment][, diff(grou
  ri_time_read_article[i] <- d_ri[, .('group_mean' = mean(time_read_article)), by=treatment][, diff(grou
  ri_donation[i]          <- d_ri[, .('group_mean' = mean(donation)),          by=treatment][, diff(grou

}

ri <- data.table(
  questions_correct = ri_questions_correct,
  credibility       = ri_credibility,
  importance        = ri_importance,
  time_read_article = ri_time_read_article,
  donation          = ri_donation
)
```

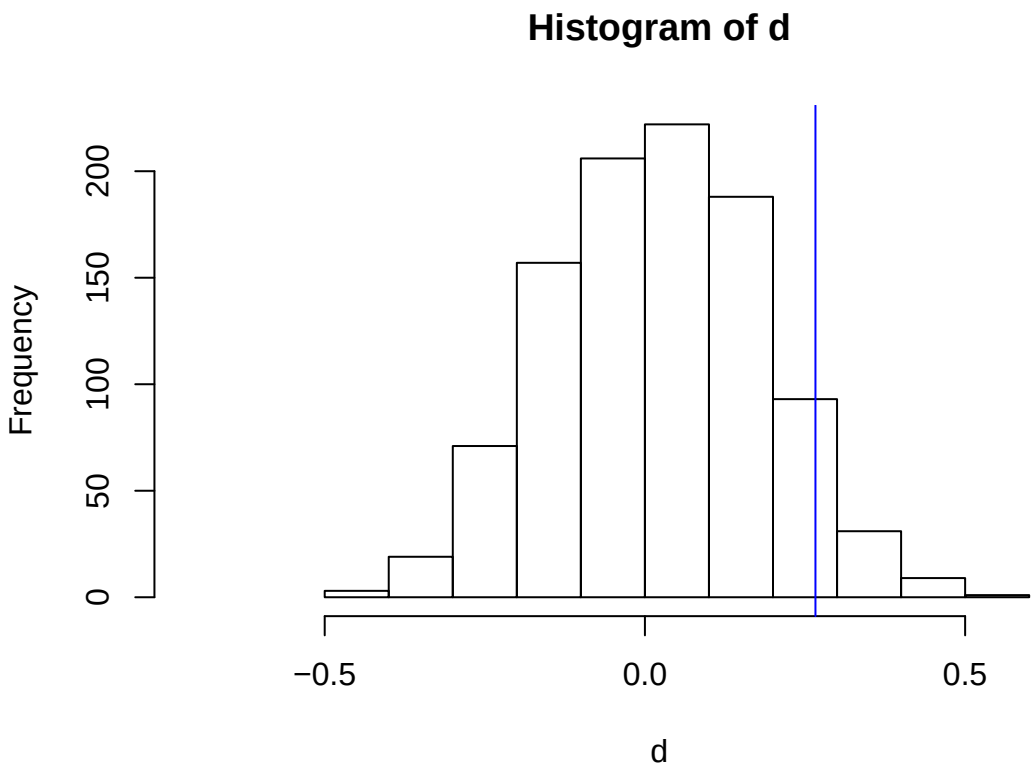
```r
visualize_ri <- function(d, ate) {
  hist(d, xlim = c(min(d)-0.25, max(d)+0.25))
  abline(v=ate, col='blue', lwd = 1)
}

visualize_ri(ri$questions_correct, ate_questions_correct)
```



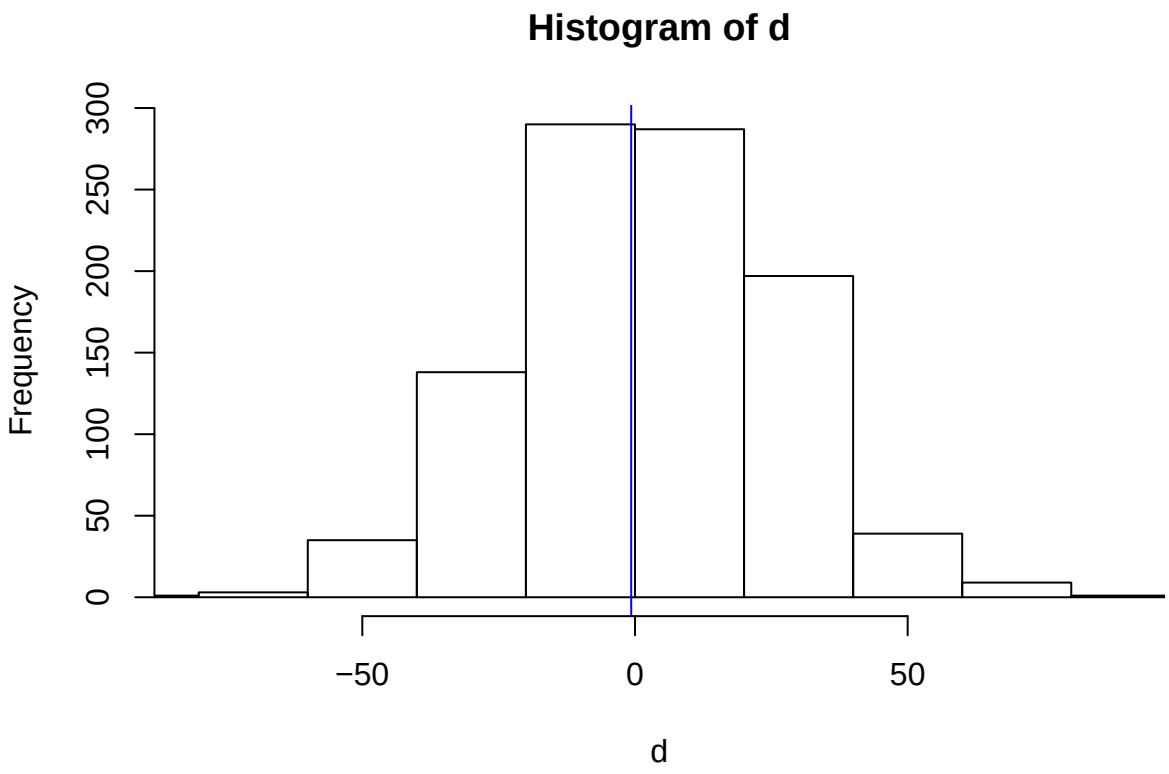

```
visualize_ri(ri$credibility,       ate_credibility)
```



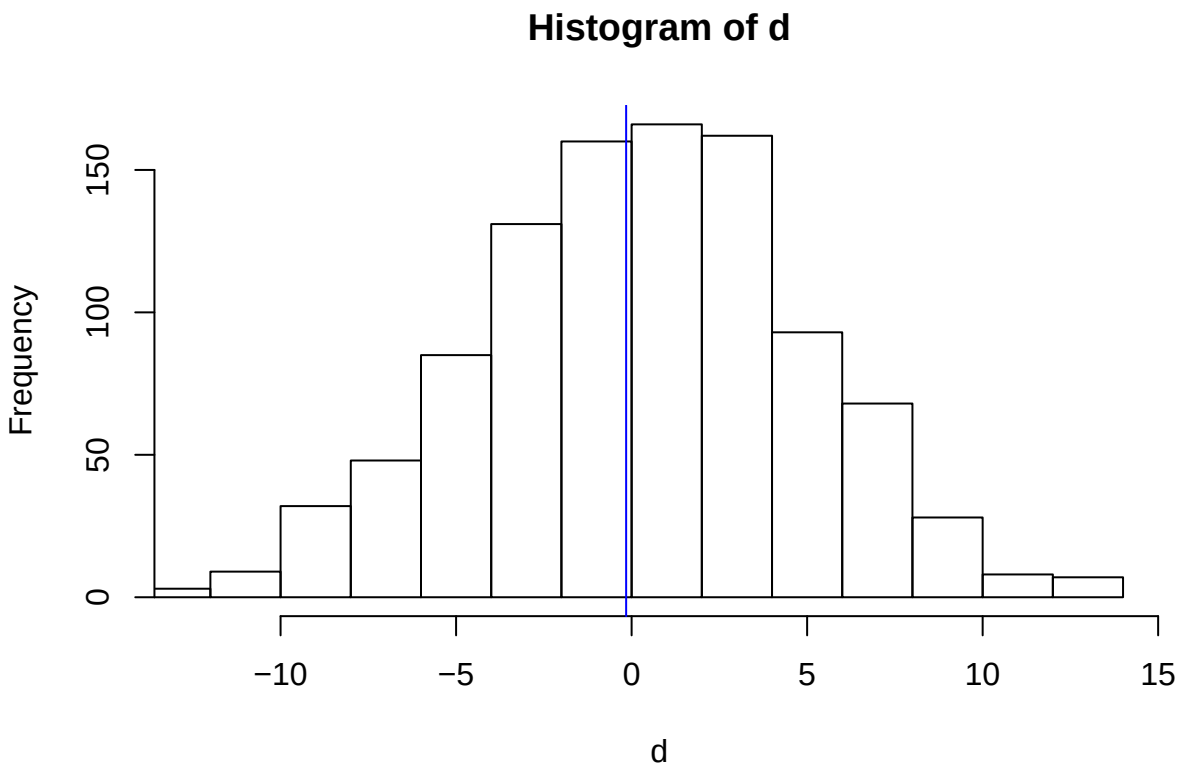

```
visualize_ri(ri$importance,        ate_importance)
```




```
visualize_ri(ri$time_read_article, ate_time_read_article)
```




```
visualize_ri(ri$donation,          ate_donation)
```



Subgroup testing

```
f <- read_data(data_file)
f <- f[time_read_article > 120, ]

#plot(density(f$time_read_article))
```

```
wilcox.test(f[treatment == 0, importance],  f[treatment == 1, importance])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  f[treatment == 0, importance] and f[treatment == 1, importance]
## W = 1435, p-value = 0.492
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(f[treatment == 0, credibility], f[treatment == 1, credibility])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  f[treatment == 0, credibility] and f[treatment == 1, credibility]
## W = 1493.5, p-value = 0.7715
## alternative hypothesis: true location shift is not equal to 0
```

```
mod <- lm(donation ~ treatment, data=f)
stargazer(mod, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                donation
## ------------------------------------------------
## treatment1                      -3.752
##                                 (4.456)
##
## Constant                       20.127***
##                                 (3.165)
##
## ------------------------------------------------
## Observations                      111
## R2                               0.006
## Adjusted R2                     -0.003
## Residual Std. Error       23.473 (df = 109)
## F Statistic              0.709 (df = 1; 109)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
control_mean_donation <- mean(f[treatment == 0, donation])
treat_mean_donation <- mean(f[treatment == 1, donation])

ggplot(f, aes(x = donation, fill = treatment)) +
  geom_density(alpha = 0.4) +
  geom_vline(aes(xintercept = control_mean_donation), color = "darkgreen", linetype = "dashed", size = 
  geom_vline(aes(xintercept = treat_mean_donation), color = "black", linetype = "dashed", size = 1) +
  annotate("text", x = 48, y = 0.02, label = paste("Control mean: $", round(control_mean_donation, 2)),
  annotate("text", x = 11, y = 0.02, label = paste("Treatment mean: $", round(treat_mean_donation, 2)),
  ggtitle("Density of donation amount for Control and Treatment") + geom_rug() +
  ylab("Density") +
  xlab("Donation ($)")
```