Problem Set #2

W241 - Field Experiments

Alex Smith

January 30, 2016

FE exercise 3.6

The Clingingsmith, Khwaja, and Kremer study discussed in section 3.5 may be be used to test the sharp null hypothesis that winning the visa lottery for the pilgrimage to Mecca had no effect on the views of Pakistani Muslims toward people from other countries. Assume that the Pakistani authorities assigned visas using complete random assignment.

```
# set working directory for Problem Set #2
setwd("/Users/Alex/Documents/Berkeley/1601Spring/W241/PS2")

# read in the data and simplify just to the columns we want to analyze
pakistani <- read.csv("1Pakistani.csv")
pakistani2 <- pakistani[,c(1,8)]
head(pakistani2)
```

```
##   success views
## 1       0     2
## 2       0     1
## 3       0     0
## 4       0     5
## 5       0     3
## 6       0     2
```

a. Conduct 10,000 simulated random assignments under the sharp null hypothesis.

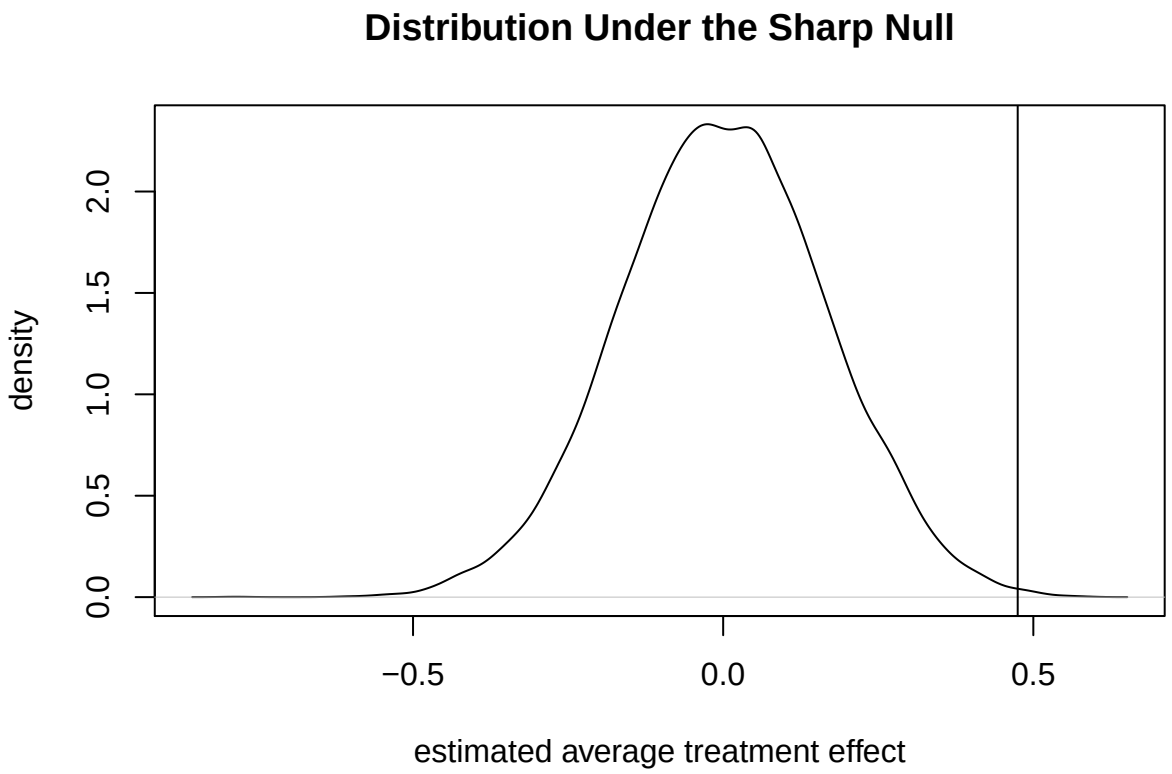
```
# create a function that uses simple random assignment to divide observations
# into treatment and control groups
observations <- nrow(pakistani2)
randomize <- function(random_qty) {
  original_random_qty = random_qty
  if (random_qty %% 2 == 1) {random_qty = random_qty + 1}
  random_assignment <- sample(c(rep(0,random_qty/2),rep(1,random_qty/2)))
  return(random_assignment[1:original_random_qty])
}

# create a function that calculates the average treatment effect (ATE) based on two
# vectors, the first determining treatment or control assignment and the second
# the observed outcome
ate <- function(assignment_vector, outcome_vector) {
```

```
  outcomes <- assignment_vector * outcome_vector +
    (1-assignment_vector) * outcome_vector
  return(mean(outcomes[assignment_vector == 1]) -
           mean(outcomes[assignment_vector == 0]))
}

# calculate the actual estimated ATE
actualATE <- ate(pakistani2[,1],pakistani2[,2])

# perform 10,000 simulated random assignments and calculate estimated ATEs for each
SIMULATIONS <- 10000
sharp_null_dist <- replicate(SIMULATIONS,ate(randomize(observations),pakistani2[,2]))
plot(density(sharp_null_dist),main = "Distribution Under the Sharp Null",
     xlab = "estimated average treatment effect", ylab = "density")
abline(v = actualATE)
```



b. How many of the simulated random assignments generate an estimated ATE that is at least as large as the actual estimate of the ATE?

```
# calculate how many of the simulations produced an ATE at least as large as the
# actual estimated ATE
simulation_greater <- sum(sharp_null_dist >= actualATE)
```

There were **18** simulations that produced an estimated ATE at least as large as the actual estimated ATE.

c. What is the implied one-tailed p-value?

```
# calculate how many of the simulations produced an ATE at least as large as the
# actual estimated ATE
percent_greater <- simulation_greater / SIMULATIONS
```

The implied one-tailed p-value is **0.0018**.

d. How many of the simulated random assignments generate an estimated ATE that is at least as large *in absolute value* as the actual estimate of the ATE?

```
# calculate how many of the simulations produced an absolute ATE at least as large
# as the actual estimated ATE
simulation_greater_absolute <- sum(abs(sharp_null_dist) >= actualATE)
```

There were **39** simulations that produced an absolute estimated ATE at least as large as the actual estimated ATE.

e. What is the implied two-tailed p-value?

```
# calculate the percentage of simulations that produced an absolute ATE at least
# as large as the actual estimated ATE
percent_greater_absolute <- simulation_greater_absolute / SIMULATIONS
```

The implied two-tailed p-value is **0.0039**.

FE exercise 3.8

Naturally occurring experiments sometimes involve what is, in effect, block random assignment. For example, Titunik studies the effect of lotteries that determine whether state senators in TX and AR serve two-year or four-year terms in the aftermath of decennial redistricting. These lotteries are conducted within each state, and so there are effectively two distinct experiments on the effects of term length. An interesting outcome variable is the number of bills (legislative proposals) that each senator introduces during a legislative session. The table below lists the number of bills introduced by senators in both states during 2003.

```
# read in the data using the foreign library and define state values
library(foreign)
termlotteries <- read.dta("2TexasLottery.dta")
TEXAS <- 0
ARKANSAS <- 1
head(termlotteries)
```

```
##   term2year bills_introduced texas0_arkansas1
## 1         0               18                0
## 2         0               29                0
## 3         0               41                0
## 4         0               53                0
## 5         0               60                0
## 6         0               67                0
```

a. For each state, estimate the effect of having a two-year term on the number of bills introduced.

```
# create new ATE formula that considers an additional parameter, the state
ate_parameter <- function(assignment_vector, outcome_vector,
                          parameter_vector,parameter) {
  reduced <- data.frame(assignment_vector[parameter_vector == parameter],
                        outcome_vector[parameter_vector == parameter])
  outcomes <- reduced[,1] * reduced[,2] +
    (1-reduced[,1]) * reduced[,2]
  calculated_ate <- mean(outcomes[reduced[,1] == 1])-mean(outcomes[reduced[,1] == 0])
  return(calculated_ate)
}

# calculate the effect of a two year term for each state
ate_Texas <- ate_parameter(termlotteries[,1],termlotteries[,2],
                           termlotteries[,3],TEXAS)
ate_Arkansas <- ate_parameter(termlotteries[,1],termlotteries[,2],
                              termlotteries[,3],ARKANSAS)
```

The average treatment effect of a 2 year term for a Texas senator is **-16.7416667**. The average treatment effect of a 2 year term for an Arkansas senator is **-10.0947712**.

b. For each state, estimate the standard error of the estimated ATE.

```
# write a function to calculate the standard error for a list of numbers
standard_error <- function(numbers) {
  sqrt(var(numbers)/length(numbers))
}

# find the Texas and Arkansas bills passed
just_Texas <- subset(termlotteries, texas0_arkansas1 == TEXAS)
just_Arkansas <- subset(termlotteries, texas0_arkansas1 == ARKANSAS)

# compute the standard errors for each state
standard_error_Texas <- standard_error(just_Texas[,2])
standard_error_Arkansas <- standard_error(just_Arkansas[,2])
```

The standard error for the estimated ATE of Texas is **4.8973791**. The standard error for the estimated ATE of Arkansas is **1.861314**.

c. Use equation (3.10) to estimate the overall ATE for both states combined.

```
# equation 3.10 is the summation of the weighted ATEs for each block where each
# block is weighted by the share of all subjects who belong to that block

# calculate the total number of senators in each state and the total
# number of senators
total_senators <- length(termlotteries[,2])
```

```
total_Texas_senators <- length(just_Texas[,2])
total_Arkansas_senators <- length(just_Arkansas[,2])

# write a formula to calculate the weighted ATE based on equation 3.10
weighted_ATE <- function(first_ATE,first_subjects,second_ATE,second_subjects){
  first_weight <- (first_subjects/(first_subjects + second_subjects)) * first_ATE
  second_weight <- (second_subjects/(first_subjects + second_subjects)) * second_ATE
  final_weight <- first_weight + second_weight
  return(final_weight)
}

pooled_ATE <- weighted_ATE(ate_Texas,total_Texas_senators,
                           ate_Arkansas,total_Arkansas_senators)
```

The pooled ATE for both states combined is **-13.2167979**.

d. Explain why, in this study, simply pooling the data for the two states and comparing the average number of bills introduced by two-year senators to the average number of bills introduced by four-year senators leads to biased estimate of the overall ATE.

Simply pooling the data from the two states and comparing the average number of bills between the two-year and four-year senators leads to biased results because the two states have different standard deviations of data and very different average treatment effects. The randomization of term lengths was not done from a pooled population but from individual state populations. It is better to compare an Arkansas senator with another Arkansas senator than an Arkansas senator with a Texas senator.

e. Insert the estimated standard errors into equation (3.12) to estimate the standard error for the overall ATE.

```
# write a function to estimate the standard error based on equation 3.12
weighted_SE <- function(first_SE,first_population,second_SE,second_population){
  weighted_first_SE <- (first_SE)^2 *
    (first_population/(first_population+second_population))^2
  weighted_second_SE <- (second_SE)^2 *
    (second_population/(first_population+second_population))^2
  final_SE <- sqrt(weighted_first_SE + weighted_second_SE)
  return(final_SE)
}

overall_SE <- weighted_SE(standard_error_Texas, total_Texas_senators,
                          standard_error_Arkansas, total_Arkansas_senators)
```

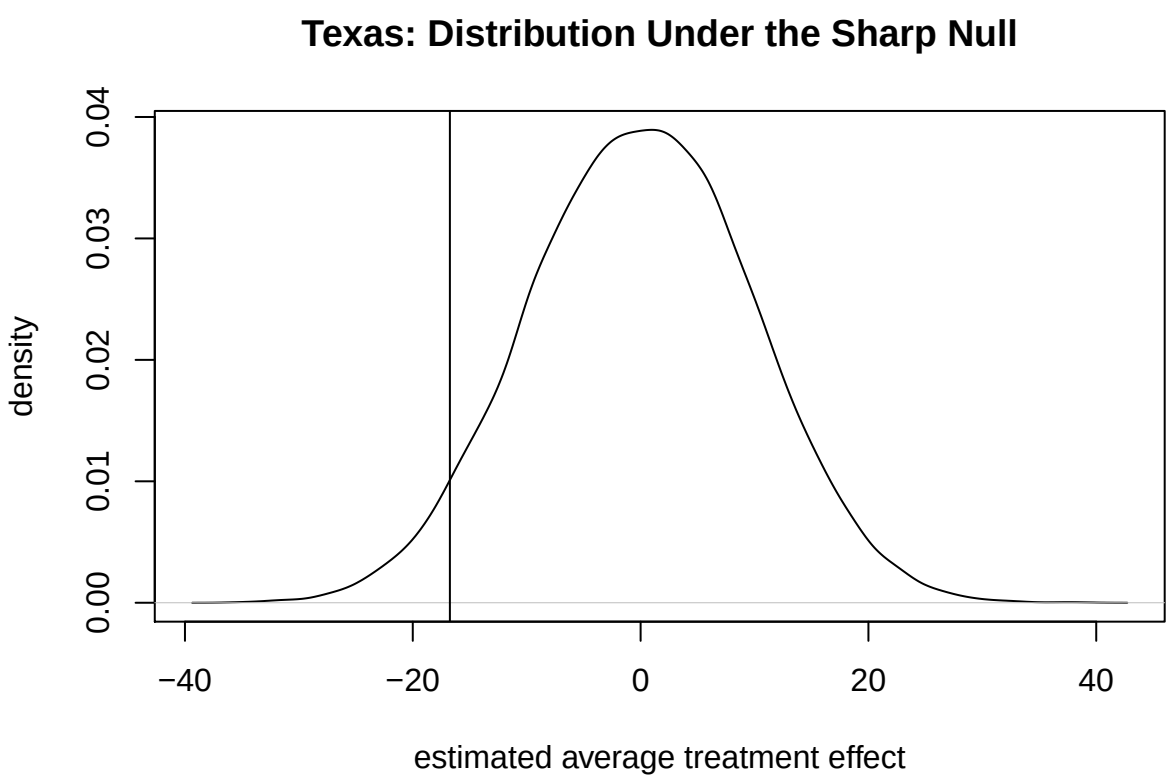
The standard error for the overall ATE, based on equation 3.12, is **2.5031171**.

f. Use randomization inference to test the sharp null hypothesis that the treatment effect is zero for senators in both states.

```
# recall our 10,000 simulations from FE Excercise 3.6
SIMULATIONS
```

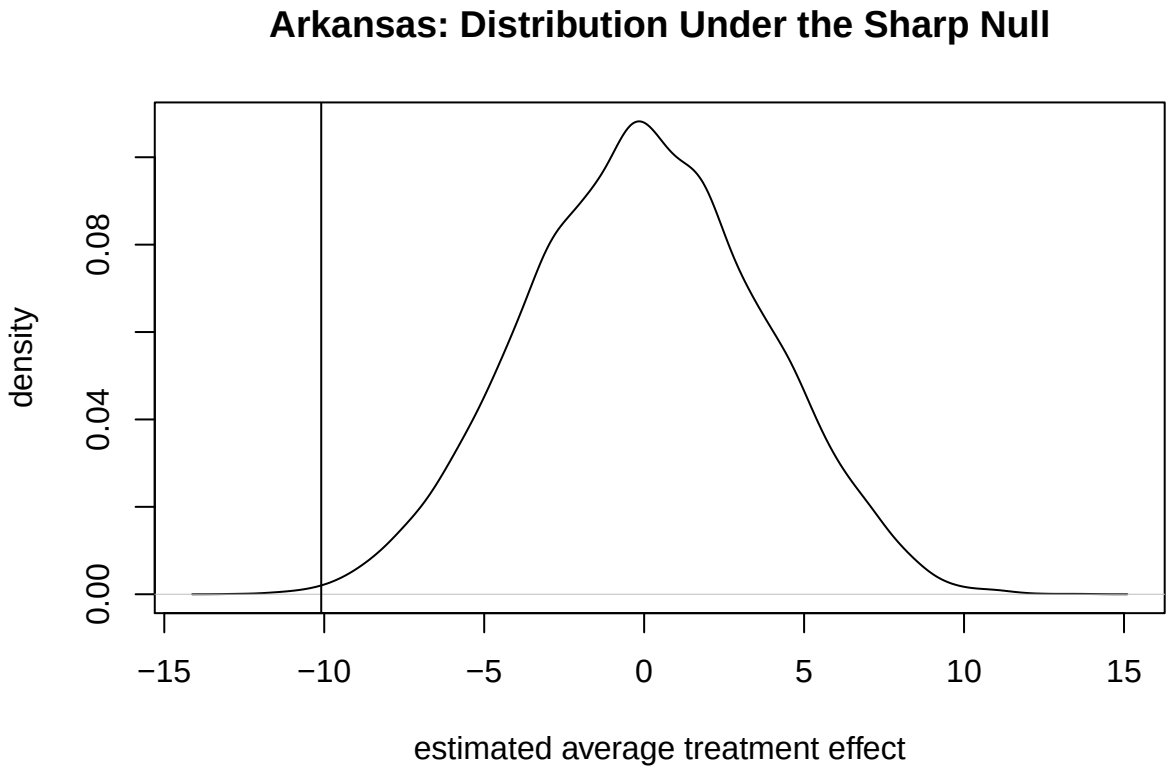
```
## [1] 10000
```

```
# create a sharp null distribution for Texas using the functions we
# already wrote for Excercise 3.6
sharp_null_dist_Texas <- replicate(SIMULATIONS,
                                   ate(randomize(total_Texas_senators),just_Texas[,2]))
plot(density(sharp_null_dist_Texas),main = "Texas: Distribution Under the Sharp Null",
     xlab = "estimated average treatment effect", ylab = "density")
abline(v = ate_Texas)
```




```
# test the Texas distribution to determine how likely our actual estimated Texas
# ATE is
Texas_simulation_greater <- sum(abs(sharp_null_dist_Texas) >= abs(ate_Texas))
Texas_p_value <- Texas_simulation_greater / SIMULATIONS

# create a sharp null distribution for Arkansas using the functions we
# already wrote for Excercise 3.6
sharp_null_dist_Arkansas <- replicate(SIMULATIONS,
                                      ate(randomize(total_Arkansas_senators),
                                          just_Arkansas[,2]))
plot(density(sharp_null_dist_Arkansas),
     main = "Arkansas: Distribution Under the Sharp Null",
     xlab = "estimated average treatment effect", ylab = "density")
abline(v = ate_Arkansas)
```

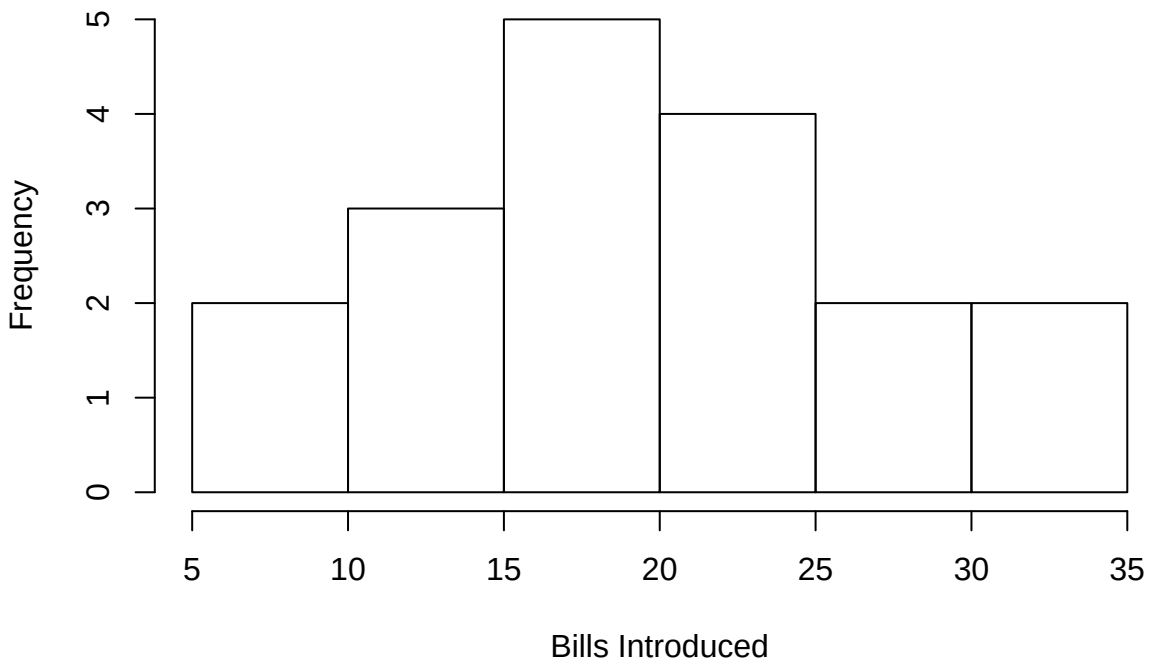



```
# test the Arkansas distribution to determine how likely our actual estimated Arkansas
# ATE is
Arkansas_simulation_greater <- sum(abs(sharp_null_dist_Arkansas) >= abs(ate_Arkansas))
Arkansas_p_value <- Arkansas_simulation_greater / SIMULATIONS
```

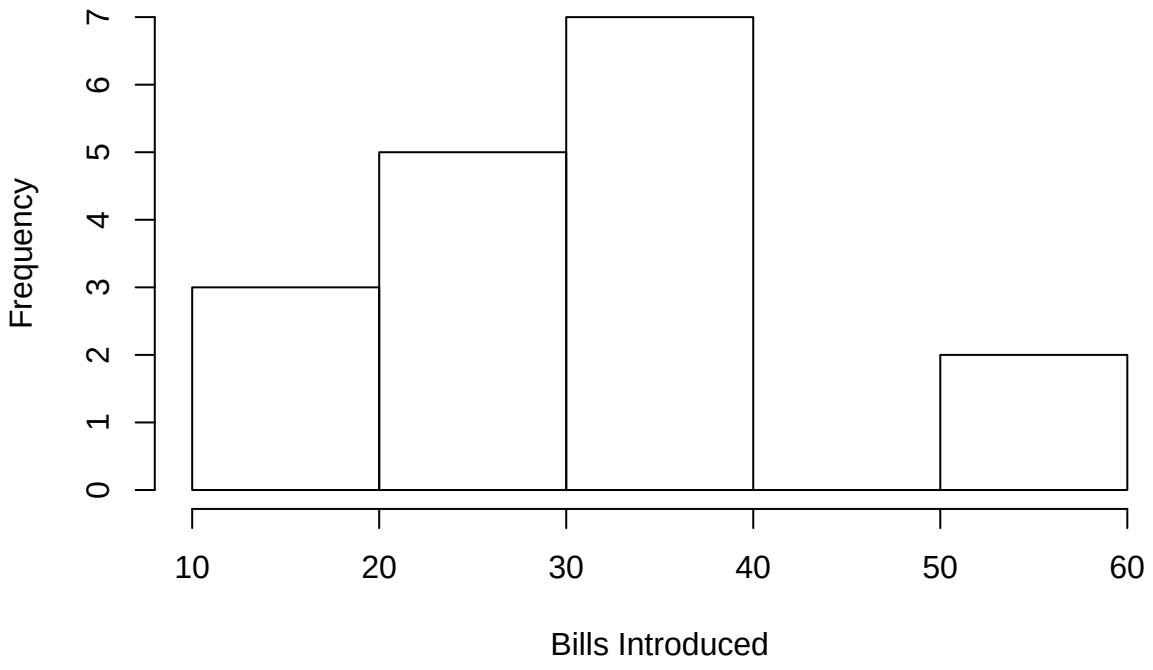
I choose to answer this question for each state individually because the lotteries for each state are independent and unrelated. I test for a difference (2-tailed test), not a directional difference. Under the sharp null hypothesis, the likelihood that we would have the average treatment effect we saw for Texas senators is **0.0849**. Under the sharp null hypothesis, the likelihood that we would have the average treatment effect we saw for Arkansas senators is **0.0033**.

g. IN Addition: Plot histograms for both the treatment and control groups in each state (for 4 histograms in total).

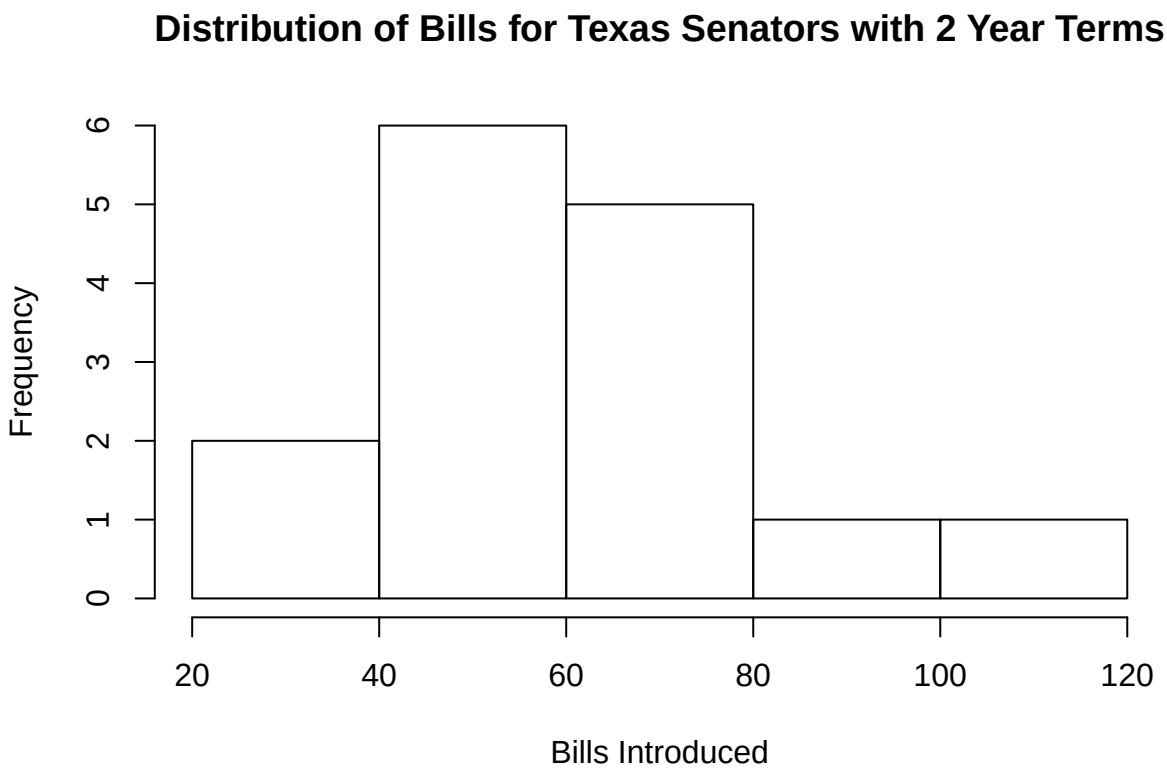
```
# histogram for Arkansas treatment group
hist(subset(termlotteries,term2year == 1 & texas0_arkansas1 == 1)[,2],
     main = "Distribution of Bills for Arkansas Senators with 2 Year Terms",
     xlab = "Bills Introduced")
```



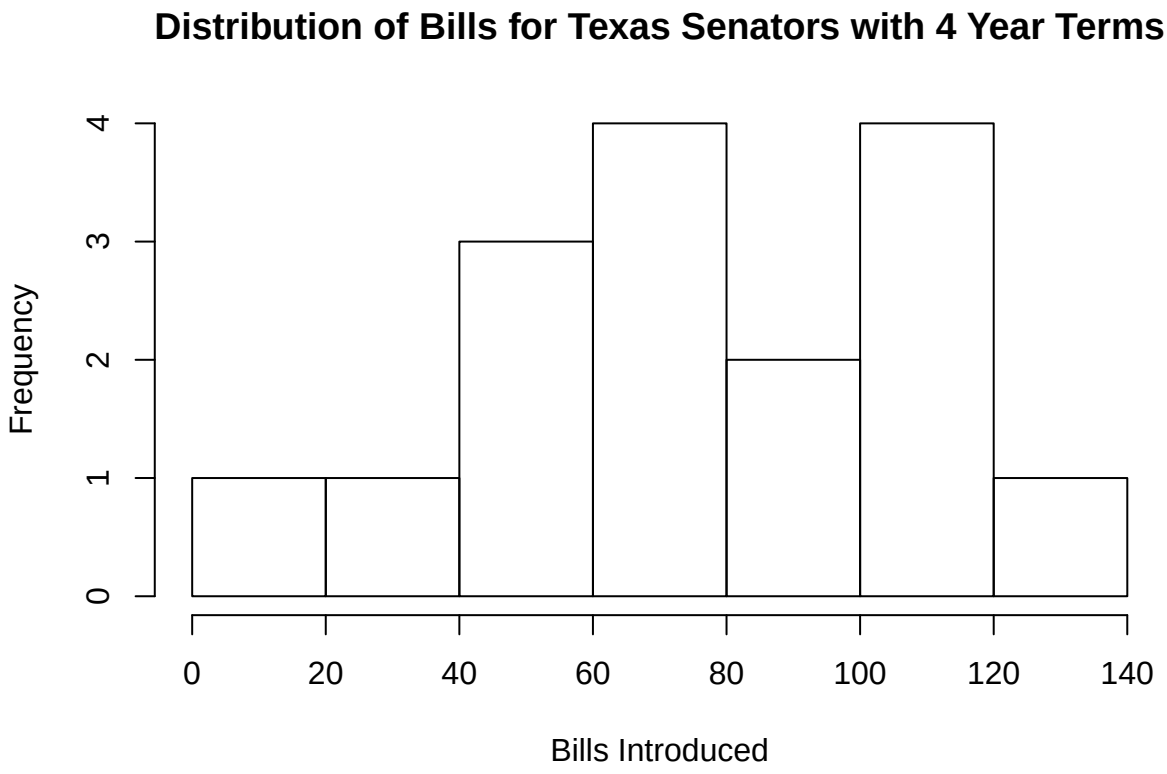

```
# histogram for Arkansas control group
hist(subset(termlotteries,term2year == 0 & texas0_arkansas1 == 1)[,2],
     main = "Distribution of Bills for Arkansas Senators with 4 Year Terms",
     xlab = "Bills Introduced")
```



```r
# histogram for Texas treatment group
hist(subset(termlotteries,term2year == 1 & texas0_arkansas1 == 0)[,2],
     main = "Distribution of Bills for Texas Senators with 2 Year Terms",
     xlab = "Bills Introduced")
```




```r
# histogram for Texas control group
hist(subset(termlotteries,term2year == 0 & texas0_arkansas1 == 0)[,2],
     main = "Distribution of Bills for Texas Senators with 4 Year Terms",
     xlab = "Bills Introduced")
```



FE exercise 3.11

Use the data in table 3.3 to simulate cluster randomized assignment. (*Notes: (a) Assume 3 clusters in treatment and 4 in control; and (b) When Gerber and Green say "simulate", they do not mean "run simulations with R code", but rather, in a casual sense "take a look at what happens if you do this this way." There is no randomization inference necessary to complete this problem.*)

a. Suppose the clusters are formed by grouping observations {1,2}, {3,4}, {5,6}, . . . , {13,14}. Use equation (3.22) to calculate the standard error assuming half of the clusters are randomly assigned to treatment.

```
# read in the dataset downloaded from the textbook website
publicworks <- read.csv("3VillagePublic.csv")

# create a vector to hold the cluster assignments
clusters1 <- vector()

# populate the vector using a for loop
for(i in 1:(nrow(publicworks)/2)){
  cluster_i <- rep(i,2)
  clusters1 <- append(clusters1,cluster_i)
}

# add the vector to the dataframe as a column
publicworks["cluster1"] <- clusters1
```

```r
# randomize the clusters into treatment and control groups
public_random <- randomize(nrow(publicworks)/2)

# write a function to assign each random assignment to the correct row
# in the dataframe based on the cluster number
random_assignment <- function(cluster_vector,random_vector){
  assignment_vector <- vector()
  for(i in 1:length(cluster_vector)){
    assignment <- random_vector[cluster_vector[i]]
    assignment_vector <- append(assignment_vector,assignment)
  }
  return(assignment_vector)
}

# write a function to find the cluster-level mean potential outcomes (table 3.7)
clusters_averages <- function(control_vector,treatment_vector,clusters){

  # create empty vectors to hold our values for the control and treatment potential
  # outcomes and for the cluster assignment
  cluster_controls <- vector()
  cluster_treatments <- vector()
  cluster_assignment <- vector()

  for(i in 1:max(clusters)){
    average_control <- mean(control_vector[clusters==i])
    average_treatment <- mean(treatment_vector[clusters==i])
    cluster_controls <- append(cluster_controls, average_control)
    cluster_treatments <- append(cluster_treatments, average_treatment)
    cluster_assignment <- append(cluster_assignment, i)
  }

  new_dataframe <- data.frame(c(cluster_assignment))
  new_dataframe["Y"] <- cluster_controls
  new_dataframe["D"] <- cluster_treatments
  colnames(new_dataframe) <- (c("clusters","Y","D"))

  return(new_dataframe)
}

# create a new clusters dataframe and populate with the cluster effects
clusters1_data <- clusters_averages(publicworks[,2], publicworks[,3], publicworks[,5])

# write equation 3.22 as an R function
clustering_SE <- function(clusters, treatment_outcomes, control_outcomes,
                          treatment_number, total_number) {
  k <- max(clusters)
  m <- treatment_number
  N <- total_number
  control_variance <- var(control_outcomes)
  treatment_variance <- var(treatment_outcomes)
  covariance <- cov(treatment_outcomes, control_outcomes)

  calcualted_clustered_SE <- sqrt((1/(k-1)) *
```

```
                                    (
                                      ((m * control_variance) / (N - m)) +
                                      (((N - m) * treatment_variance) / m) +
                                        2 * covariance
                                    ))

  return(calcualted_clustered_SE)
                            }

# calculate the standard error for this first cluster
cluster1_SE <- clustering_SE(clusters1_data[,1],
                             clusters1_data[,3],
                             clusters1_data[,2],
                             nrow(publicworks)/2,
                             nrow(publicworks))
```

Using equation 3.22, the standard error is **4.9489457**.

b. Suppose that clusters are instead formed by grouping observations {1,14}, {2,13}, {3,12}, ... , {7,8}. Use equation (3.22) to calculate the standard error assuming half of the clusters are randomly assigned to treatment.

```
# create the first seven assignments, and then flip them to get the
# last seven assignments, append these to the clusters2 vector
clusters2 <- vector()
for(i in 1:(nrow(publicworks)/2)){
  cluster_i <- rep(i)
  clusters2 <- append(clusters2,cluster_i)
}
clusters2 <- append(clusters2,rev(clusters2))

# append this cluster assignment to the main dataframe
publicworks["cluster2"] <- clusters2

# get the cluster level pooled outcomes
clusters2_data <- clusters_averages(publicworks[,2], publicworks[,3], publicworks[,6])

# calculate the standard error using equation 3.22
cluster2_SE <- clustering_SE(clusters2_data[,1],
                             clusters2_data[,3],
                             clusters2_data[,2],
                             nrow(publicworks)/2,
                             nrow(publicworks))
```

Using equation 3.22, the standard error is **1.1452109**.

c. Why do the two methods of forming clusters lead to different standard errors? What are the implications for the design of cluster randomized experiments?

The two methods of forming clusters lead to such different standard errors because the first clustering method created groups that were more internally homogenous and externally heterogenous. In other words, the

subjects in the first clustering method, were more likely to be in clusters with subjects similar to themselves. The second clustering method created clusters that were more similar to one another. The second method reduced the variance of the cluster-level potential outcomes.

More Practice #1

You are an employee of a newspaper and are planning an experiment to demonstrate to Apple that online advertising on your website causes people to buy iPhones. Each site visitor shown the ad campaign is exposed to $0.10 worth of advertising for iPhones. (Assume all users could see ads.) There are 1,000,000 users available to be shown ads on your newspaper's website during the one week campaign.

Apple indicates that they make a profit of $100 every time an iPhone sells and that 0.5% of visitors to your newspaper's website buy an iPhone in a given week in general, in the absence of any advertising.

a. By how much does the ad campaign need to increase the probability of purchase in order to be "worth it" and a positive ROI (supposing there are no long-run effects and all the effects are measured within that week)?

```
# constants of the ad campaign
AD_COST <- 0.10
USERS <- 1000000
PHONE_PROFIT <- 100
REGULAR_BUYERS <- 0.05

# assume all users will see the ad campaign
costs <- AD_COST * USERS

# calculate the number of buyers needed to break even
people_break_even <- costs / PHONE_PROFIT

# calculate the total number of buyers included those who would buy the phone anyway
total_buyers <- people_break_even + REGULAR_BUYERS * USERS

# calculate the new probability of purchase to be "worth it" and how much more
# likely this is
worth_it <- total_buyers / USERS
worth_it_difference <- worth_it - REGULAR_BUYERS
```

The ad campaign must increase the probability of purchase by **0.001**.

b. Assume the measured effect is 0.2 percentage points. If users are split 50:50 between the treatment group (exposed to iPhone ads) and control group (exposed to unrelated advertising or nothing; something you can assume has no effect), what will be the confidence interval of your estimate on whether people purchase the phone?

```
# set the split value and assumed measured effects as constants
PHONE_SPLIT <- 0.50
MEASURED_EFFECT <- 0.002
```

```r
# let us write a function that generates sample data based on the inputs
# in the prompt
make_phone_sample <- function(individuals, split, outcome_anyway, outcome_treatment){

  # simulate the outcomes for control as those individuals who would
  # buy an iPhone anyway. the individuals who buy a phone are given
  # an outcome of 1, others are given an outcome of 0
  phone_control_group_buyers <- rep(1, split * individuals * outcome_anyway)
  phone_control_group_nonbuyers <- rep(0, split * individuals * (1 - outcome_anyway))
  phone_control_group <- c(phone_control_group_buyers, phone_control_group_nonbuyers)

  # simulate the outcomes for control as those individuals who would
  # buy an iPhone anyway AND the individuals who were affected by the
  # the ad. the individuals who buy a phone are given an outcome of 1,
  # others are given an outcome of 0
  phone_treatment_group_buyers <- rep(1, split * individuals *
                                        (outcome_anyway + outcome_treatment))
  phone_treatment_group_nonbuyers <- rep(0, split * individuals *
                                        (1-(outcome_anyway + outcome_treatment)))
  phone_treatment_group <- c(phone_treatment_group_buyers,
                             phone_treatment_group_nonbuyers)

  # create the simulation for assignment to control and treatment
  phone_assignments <- c(rep(0, split * individuals), rep(1, split * individuals))

  # create the data frame that holds the assignment to control/treatment and
  # the outcome for each individual
  phone_data <- data.frame(phone_assignments,
                           c(phone_control_group, phone_treatment_group))
  colnames(phone_data) <- c("assignment", "outcome")
  return(phone_data)
}

# generate a dataframe that simulates the question prompt, measured effect of 0.2
# and a 50:50 split
iphone_simulation1 <- make_phone_sample(USERS, PHONE_SPLIT,
                                        REGULAR_BUYERS, MEASURED_EFFECT)

# create a linear regression using R's built-in function
iphone1 <- lm(outcome ~ assignment, iphone_simulation1)

# calculate the confidence intervals
iphone1_confidence <- confint(iphone1,'assignment',level=0.95)
iphone1_confidence
```

```
##                   2.5 %      97.5 %
## assignment 0.001137632 0.002862368
```

```r
# alternatively, we could use the standard error formula provided below
# I just noticed this standard error formula and didn't want to
# delete all my work

two_sample_SE <- function(x1, x2, n1, n2) {
```

```
  p <- (x1 + x2) / (n1 + n2)
  return(sqrt(
    p * (1 - p) * (1 / n1 + 1 / n2)
  ))
}

# calculate the successes (or purchases) and trials (or site visits)
successes_control1 <- USERS * PHONE_SPLIT * (REGULAR_BUYERS)
successes_treatment1 <- USERS * PHONE_SPLIT * (REGULAR_BUYERS + MEASURED_EFFECT)
trials_control1 <- USERS * PHONE_SPLIT
trials_treatment1 <- USERS * PHONE_SPLIT

iphone1_SE <- two_sample_SE(successes_control1, successes_treatment1,
                            trials_control1, trials_treatment1)

# both approaches provide the same standard error. I calcualte the confidence
# interval for all people purchasing the phone (NOT just the treatment effect)
# because I'm interested in reporting how many people purchased the iPhone
CONFIDENCE_MULTIPLIER <- 1.96

bottom_iphone1 <- REGULAR_BUYERS + MEASURED_EFFECT - CONFIDENCE_MULTIPLIER * iphone1_SE
top_iphone1 <- REGULAR_BUYERS + MEASURED_EFFECT + CONFIDENCE_MULTIPLIER * iphone1_SE
```

My 95% confidence interval for portion of users purchasing iPhones after treatment is from **0.0511376** to **0.0528624**.

- **Note:** The standard error for a two-sample proportion test is $\sqrt{p(1-p) * (\frac{1}{n_1} + \frac{1}{n_2})}$ where $p = \frac{x_1+x_2}{n_1+n_2}$, where x and n refer to the number of "successes" (here, purchases) over the number of "trials" (here, site visits). The length of each tail of a 95% confidence interval is calculated by multiplying the standard error by 1.96.

c. Is this confidence interval precise enough that you would recommend running this experiment? Why or why not?

In order to break even, we needed to increase the probability of purchase by **0.001**. The bottom of our confidence interval is just slightly above this threshold. This means that I would recommend the advertising campaign provided I knew of no other more cost-effective means of increasing iPhone sales.

d. Your boss at the newspaper, worried about potential loss of revenue, says he is not willing to hold back a control group any larger than 1% of users. What would be the width of the confidence interval for this experiment if only 1% of users were placed in the control group?

```
# calculate the successes (or purchases) and trials (or site visits) for this new
# experiment scenario
PHONE_SPLIT2 <- 0.01

successes_control2 <- USERS * PHONE_SPLIT2 * (REGULAR_BUYERS)
successes_treatment2 <- USERS * (1 - PHONE_SPLIT2) * (REGULAR_BUYERS + MEASURED_EFFECT)
trials_control2 <- USERS * PHONE_SPLIT2
trials_treatment2 <- USERS * (1 - PHONE_SPLIT2)
```

```
iphone2_SE <- two_sample_SE(successes_control2, successes_treatment2,
                            trials_control2, trials_treatment2)

# calculate the bottom and top of the confidence intervals
bottom_iphone2 <- REGULAR_BUYERS + MEASURED_EFFECT - CONFIDENCE_MULTIPLIER * iphone2_SE
top_iphone2 <- REGULAR_BUYERS + MEASURED_EFFECT + CONFIDENCE_MULTIPLIER * iphone2_SE
```

The new 95% confidence interval with only 1% of the users in the treatment group is from **0.0476271** to **0.0563729**.

More Practice #2

Here you will find a set of data from an auction experiment by John List and David Lucking-Reiley (2000).

```
auction <- read.csv("5Auction.csv")
head(auction)
```

```
##   bid uniform_price_auction
## 1   5                     1
## 2   5                     1
## 3  20                     0
## 4   0                     1
## 5  20                     1
## 6   0                     1
```

In this experiment, the experimenters invited consumers at a sports card trading show to bid against one other bidder for a pair trading cards. We abstract from the multi-unit-auction details here, and simply state that the treatment auction format was theoretically predicted to produce lower bids than the control auction format. We provide you a relevant subset of data from the experiment.

a. Compute a 95% confidence interval for the difference between the treatment mean and the control mean, using analytic formulas for a two-sample t-test from your earlier statistics course.

```
# create a t-test using R's built in function
cards_model <- t.test(bid ~ uniform_price_auction, data = auction)
cards_model
```

```
##
##  Welch Two Sample t-test
##
## data:  bid by uniform_price_auction
## t = 2.8211, df = 61.983, p-value = 0.006421
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.557141 20.854624
## sample estimates:
## mean in group 0 mean in group 1
##        28.82353        16.61765
```

The 95% confidence for the difference between the treatment mean and the control mean is **3.5571406** to **20.8546241**.

b. In plain language, what does this confidence interval mean?

In plain language, this confidence interval means that I am 95% certain that the true treatment effect lies between **3.5571406** and **20.8546241**. Or, another way to say it, if I were to replicate this experiment again and again, 95% of the confidence intervals I constructed would have the true treatment effect.

c. Regression on a binary treatment variable turns out to give one the same answer as the standard analytic formula you just used. Demonstrate this by regressing the bid on a binary variable equal to 0 for the control auction and 1 for the treatment auction.

```
# create a linear regression using R's built-in function
cards_linear_model <- lm(bid ~ uniform_price_auction, data = auction)
summary(cards_linear_model)
```

```
##
## Call:
## lm(formula = bid ~ uniform_price_auction, data = auction)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.824 -11.618  -3.221   8.382  58.382
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              28.824      3.059   9.421 7.81e-14 ***
## uniform_price_auction  -12.206      4.327  -2.821  0.00631 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.84 on 66 degrees of freedom
## Multiple R-squared:  0.1076, Adjusted R-squared:  0.09409
## F-statistic: 7.959 on 1 and 66 DF,  p-value: 0.006315
```

d. Calculate the 95% confidence interval you get from the regression.

```
# use the confint function to calculate the confidence intervals for this
cards_linear_confidence <- confint(cards_linear_model,
                                   'uniform_price_auction',level=0.95)
cards_linear_confidence
```

```
##                           2.5 %    97.5 %
## uniform_price_auction -20.84416 -3.567603
```

The 95% confidence interval for the regression is from **-20.844162** to **-3.5676027**.

e. On to p-values. What p-value does the regression report? Note: please use two-tailed tests for the entire problem.

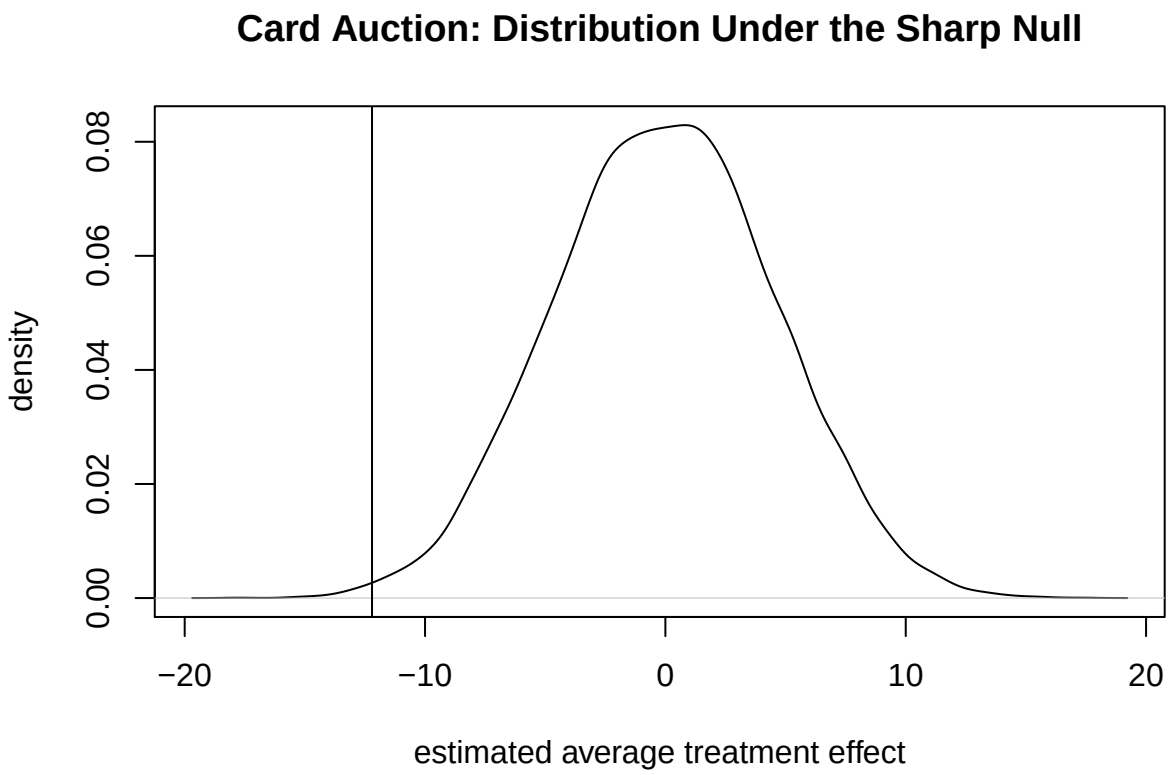
```
# grab the p-value using the anova function
cards_linear_p_value <- anova(cards_linear_model)$'Pr(>F)'[1]
```

The regression reports a p-value of **0.0063148**.

f. Now compute the same p-value using randomization inference.

```
# create a sharp null distribution for the card auction
sharp_null_dist_cards <- replicate(SIMULATIONS,
                                   ate(randomize(nrow(auction)),auction[,1]))

# plot the sharp null distribution and add the treatment effect seen
plot(density(sharp_null_dist_cards),
     main = "Card Auction: Distribution Under the Sharp Null",
     xlab = "estimated average treatment effect", ylab = "density")
cards_ATE <- ate(auction[,2],auction[,1])
abline(v = cards_ATE)
```




```
# calculate the p-value
cards_sharp_p_value <- sum(abs(sharp_null_dist_cards) >= abs(cards_ATE)) / SIMULATIONS
```

The p-value using randomization inference is **0.006**.

g. Compute the same p-value again using analytic formulas for a two-sample t-test from your earlier statistics course. (Also see part (a).)

```
# call the model calculated in part (a) and pull the p-value
cards_t_p_value <- cards_model$p.value[1]
```

The t-test reports a p-value of **0.0064208**.

h. Compare the two p-values in parts (e) and (f). Are they much different? Why or why not? How might your answer to this question change if the sample size were different?

The difference between the p-values calculated with linear regression and with randomization inference is **0**. The difference is very small. The p-value is greater for the sharp null hypothesis because the sharp null is a stricter test. The sharp null is that there is no difference between the treatment and control for every individual. The "regular" null from the linear regression only tests that there is no difference between the treatment and control. As my sample size increases, I expect that the difference between these p-values will disappear because larger sample sizes will give me greater and greater certainty as my p-value approaches 0.