Problem set 5

Nathaniel Williams

2025-12-07

Part 1

Step 1: Create a simulated data set with a dependent variable that is a linear function of a

treatment variable and a confounding variable. Fit a linear model for the true data generating process and print the summary table.

```
set.seed(123)
# Confounder first its unrelated to others
n<-1000
C<- rnorm(n, mean=0, sd=1)
# Treatment, is impacted by confounder
X<- C*0.35 + rnorm(n, mean=0, sd=1)
#Outcome dependent on x and confounder
Y<- X*.5 + C*.25 + rnorm(n, mean=0, sd=1)
#true model

model<- lm(Y~X+C)
summary(model)
```

```
##
## Call:
## lm(formula = Y ~ X + C)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8360 -0.6277 -0.0370  0.6538  3.3787
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02093    0.03098  -0.676    0.499
## X            0.52751    0.03079  17.135  < 2e-16 ***
## C            0.21888    0.03401   6.435 1.91e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9788 on 997 degrees of freedom
## Multiple R-squared:  0.3346, Adjusted R-squared:  0.3332
## F-statistic: 250.6 on 2 and 997 DF,  p-value: < 2.2e-16
```

```
print(summary(model))
```

```
##
```

```
## Call:
## lm(formula = Y ~ X + C)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8360 -0.6277 -0.0370  0.6538  3.3787
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02093    0.03098  -0.676    0.499
## X            0.52751    0.03079  17.135  < 2e-16 ***
## C            0.21888    0.03401   6.435 1.91e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9788 on 997 degrees of freedom
## Multiple R-squared:  0.3346, Adjusted R-squared:  0.3332
## F-statistic: 250.6 on 2 and 997 DF,  p-value: < 2.2e-16
```

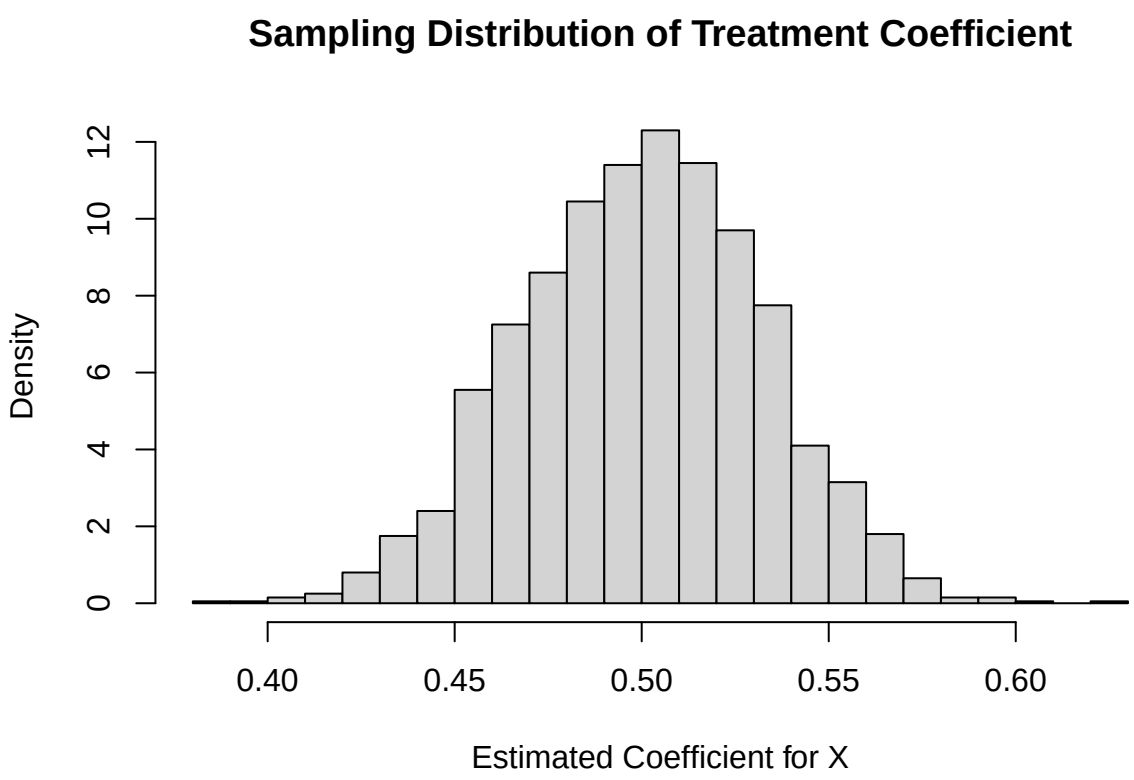
A. Using the true model, demonstrate that the coefficient for your treatment variable

follows the central limit theorem. That is, demonstrate that the coefficient's sampling distribution is approximately normal

```
#I think monte carlo loop
#number of iterations
sims<-2000
#empty storage for treatment estimates
treatment_hat <- numeric(sims)

for(i in 1:sims){
  #true dgp
  C<- rnorm(n, mean=0, sd=1)
  X<- C*0.35 + rnorm(n, mean=0, sd=1)
  Y<- X*.5 + C*.25 + rnorm(n, mean=0, sd=1)
#true model
  model2<-lm(Y~X+C)
  #store
  treatment_hat[i] <- coef(model2)['X']
}
```

Need to visualize the data

```
hist(treatment_hat,
     breaks = 30,
     probability = TRUE,
     main = "Sampling Distribution of Treatment Coefficient",
     xlab = "Estimated Coefficient for X")
```



Its a bell shaped distrubition/roughly normal

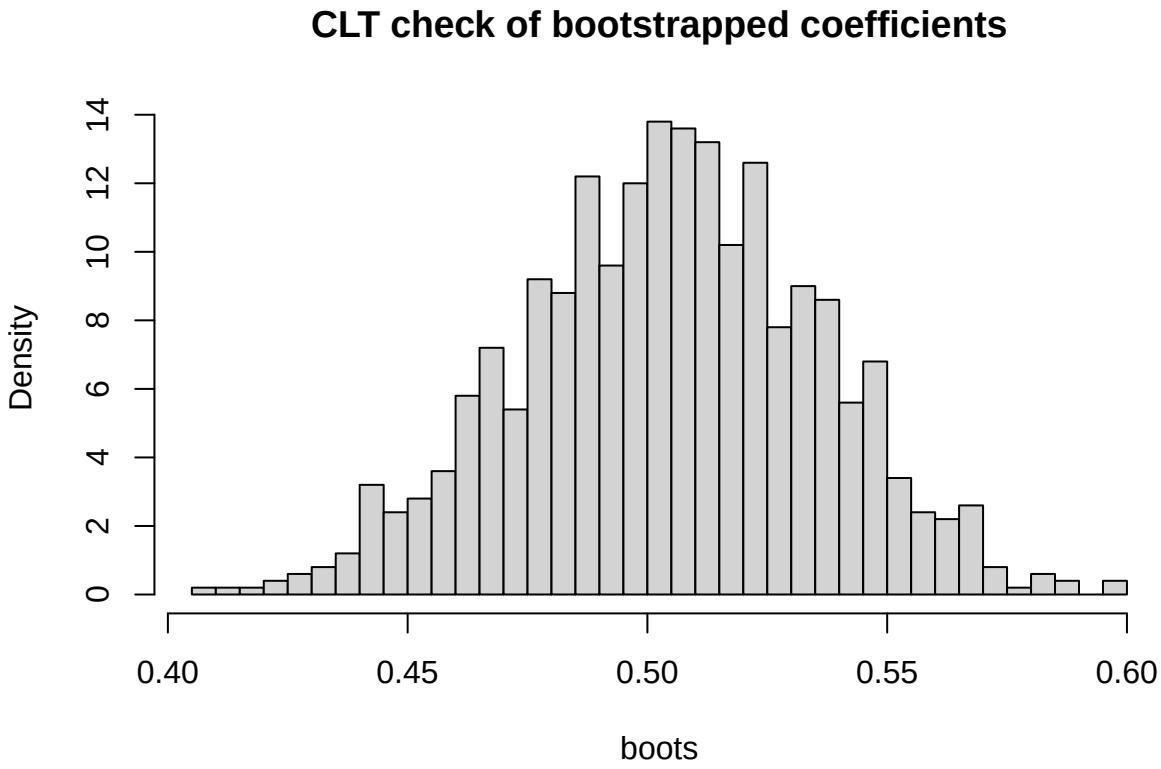
B. Compute the bootstrapped standard error for the coefficient of the treatment variable.

```
#data frame to store and plug into model later
data<- data.frame(Y, X, C)

#Storage for bootsrapped coef, n is 1000 still so should be fine
boots<-numeric(n)

#loop
for(i in 1:n) {
  #indices with replacement for bootstrapping
  boot_indices <- sample(1:n, n, replace = TRUE)

  #bootstrap sample using indices
  boot_sample <- data[boot_indices, ]

  #Fit the new bootsrap model
  boot_model <- lm(Y ~ X + C, data = boot_sample)

  #Store X
  boots[i] <- coef(boot_model)['X']
}
```

```r
#Standard error is just of the bootsrap coeffficients
boot_se <- sd(boots)
print(boot_se)
```

```
## [1] 0.03108411
```

```r
hist(boots,
     breaks = 30,
     probability = TRUE,
     main = 'CLT check of bootstrapped coefficients')
```



C. Fit a model that omits the confounding variable. Repeat part (a) for this new model

and plot the sampling distribution of the treatment variable's coefficient. How do your results differ? What does this imply about statistical tests based on a coefficient's sampling distribution

```r
#New model omitting C
model3<-lm(Y~X)
summary(model3)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
```

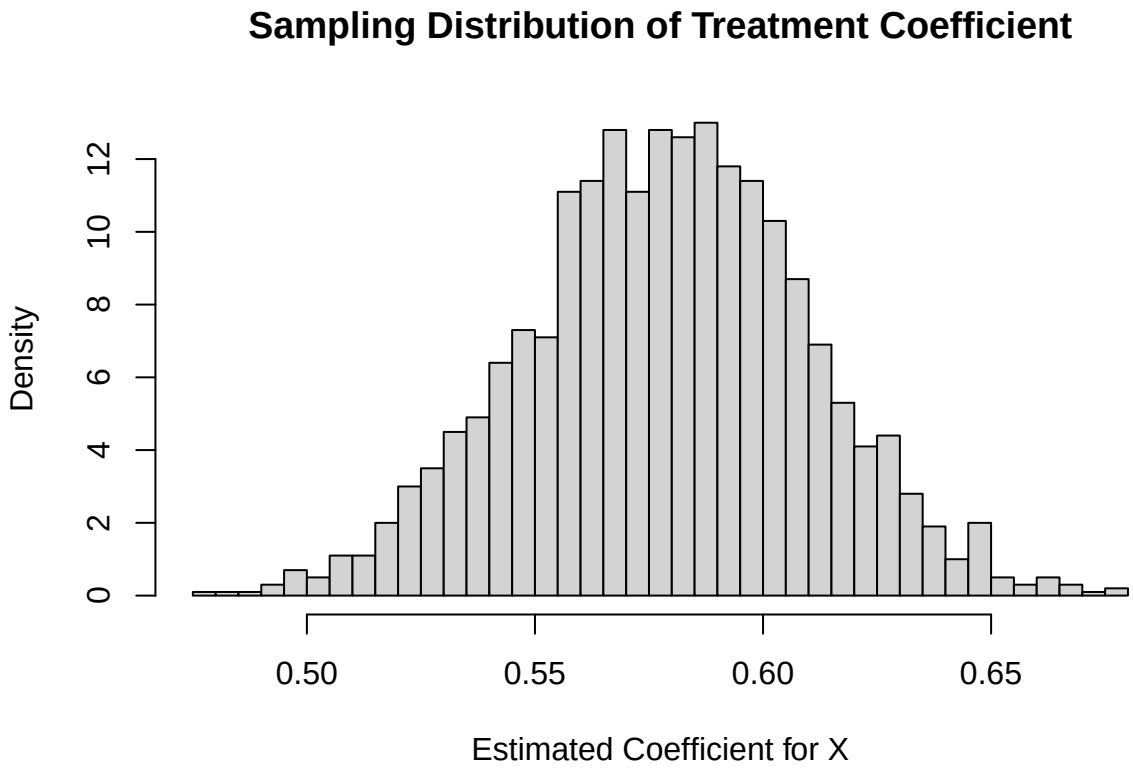
```
##     Min      1Q  Median      3Q     Max
## -3.0341 -0.7575  0.0036  0.7019  3.4117
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02283    0.03329  -0.686    0.493
## X            0.61829    0.03135  19.721   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.052 on 998 degrees of freedom
## Multiple R-squared:  0.2804, Adjusted R-squared:  0.2797
## F-statistic: 388.9 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
#power of X has been inflated to .6 instead of .5


#Repeat A (im just copy and pasting part a and editing what needs to be edited)
confounded_treatment <- numeric(sims)

for(i in 1:sims){
  #true dgp
  C<- rnorm(n, mean=0, sd=1)
  X<- C*0.35 + rnorm(n, mean=0, sd=1)
  Y<- X*.5 + C*.25 + rnorm(n, mean=0, sd=1)
#new model with no confounder accounted for
  modelconfounded<-lm(Y~X)
  #store
  confounded_treatment[i] <- coef(modelconfounded)['X']
}


summary(modelconfounded)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.05539 -0.69421  0.02508  0.69170  3.02944
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.05189    0.03140  -1.653   0.0987 .
## X            0.54253    0.03000  18.082   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9927 on 998 degrees of freedom
## Multiple R-squared:  0.2468, Adjusted R-squared:  0.246
## F-statistic:   327 on 1 and 998 DF,  p-value: < 2.2e-16
```

Need to visualize the data

```r
hist(confounded_treatment,
     breaks = 30,
     probability = TRUE,
     main = "Sampling Distribution of Treatment Coefficient",
     xlab = "Estimated Coefficient for X")
```



The distribution has become less of a bell curve, data is a bit more spread out indicating our standard error is wider. Despite this we have gotten close to the true population parameters with the bootstrapped model, showing the power of the method in eliminating the impact of unobservable confounders through repeated sampling within the data.

Part 2

I am going to use the star data set from the book/class as it is easily accesible and user friendly.

Conduct a hypothesis test for a difference in means. You decide what the hypotheses are, whether you use a t-test or a z-test, and what the level of significance is. Explain your decisions, and interpret your results both substantively and statistically.

```r
star<-read.csv('https://raw.githubusercontent.com/MLBurnham/pols_602/refs/heads/main/data/STAR.csv')
```

Just gonna stick with the hypothesis that small class sizes lead to higher average test scores small class -> bettter test score

```r
#Im going to just look at cumulative score for math and reading
star$total<- star$reading + star$math
```

```r
#t test 95 percetn confidence as is standard
results_star <- t.test(total ~ classtype, data = star,
                       subset = classtype %in% c('small', 'regular'),
                       alternative = "less",
                       conf.level = 0.95
                       )


print(results_star)
```

```
##
##  Welch Two Sample t-test
##
## data:  total by classtype
## t = -3.3562, df = 1220.5, p-value = 0.0004073
## alternative hypothesis: true difference in means between group regular and group small is less than 0
## 95 percent confidence interval:
##       -Inf -6.725996
## sample estimates:
## mean in group regular   mean in group small
##              1254.329              1267.530
```

R is doing reg - small, so anticipated t-value for my hypothsis would be negative as the differene in means here is going to be negative if regular score< small score and regular - small

I used a t-test as we discussed that we can relax some assumptions about the distrubtion of data and as the sample size grows the t test and z test difference become negligable.

The t value and p-value show that the odds of our observation are statisically distinguishable from zero and allow us to make causal inference (provided all usual assumptions about representative samples and confounders are met)

Having a small class size does contribute to better test scores, confirming the hypothesis.

B. Using the same data, fit a linear model. Interpret the coe>icient, standard error, tvalue, and p-value

```r
#model is just that test scores are a function of class size, there are no control factors in the data
startreatment<-star$classtype
staroutcome<-star$total

starmodel<- lm(staroutcome~startreatment)
summary(starmodel)
```

```
##
## Call:
## lm(formula = staroutcome ~ startreatment)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -223.53  -47.13   -3.93   45.67  209.67
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1254.329      2.656 472.229  < 2e-16 ***
## startreatmentsmall   13.200      3.920   3.368 0.000781 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.72 on 1272 degrees of freedom
## Multiple R-squared:  0.008837,   Adjusted R-squared:  0.008058
## F-statistic: 11.34 on 1 and 1272 DF,  p-value: 0.0007809
```

The estimated effect of moving from a regular class to a small class is a 13 point increase on total test scores (math + reading) Standard error of + - 3.9 points

t-value and p-value suggest a strong correlation and our findings are significantly distinguishable from zero.