# Problem Set 5

Neeraj Tom Savio

2025-12-11

**Part 1: Simulation**

**Question 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.

```
## Introductory Code
rm(list=ls())
set.seed(67)
library(ggplot2)
```

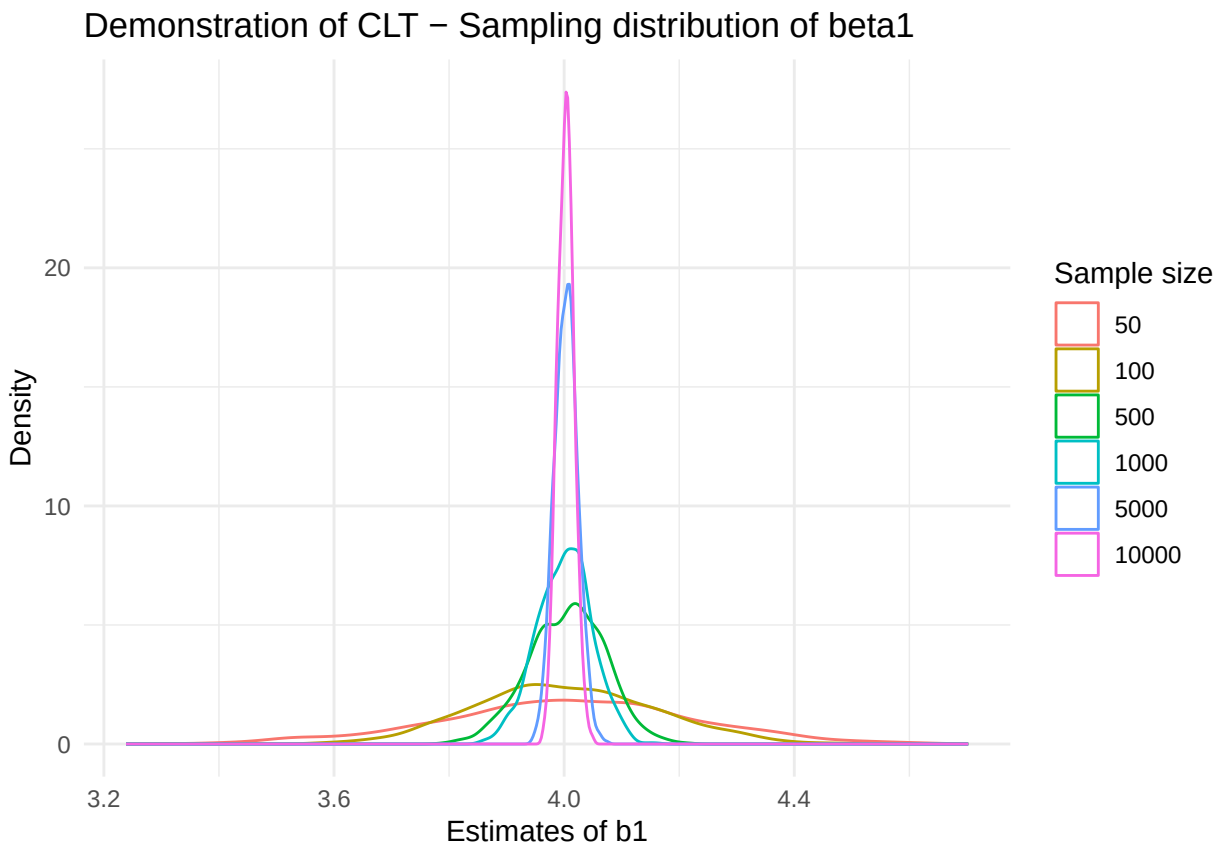
```
## Creating the population

#Variables
z <- rnorm(100000, 6, 7) # Confounder
x <- rnorm(100000, 3, 4) + 2*z #Independent Variable
y <- rnorm(100000, 7, 6) + 4*x + 3*z  #Dependent Variable

#Population
pop <- data.frame(x, y, z)
model_pop <- lm(y ~ x + z, data = pop) #Saving population to the dataframe
summary(model_pop)
```

```
##
## Call:
## lm(formula = y ~ x + z, data = pop)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31.2714  -4.0261   0.0143   4.0477  25.0701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.970699   0.028704   242.8   <2e-16 ***
## x           4.002774   0.004738   844.8   <2e-16 ***
## z           2.997329   0.009857   304.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5.998 on 99997 degrees of freedom
## Multiple R-squared:  0.9943, Adjusted R-squared:  0.9943
## F-statistic: 8.653e+06 on 2 and 99997 DF,  p-value: < 2.2e-16
```

```r
#Sample Size
n_samp <- c(50, 100, 500, 1000, 5000, 10000)
#Number of draws - How many times do we want regressions to be run
n_sim <- 1000

#Creating a results dataframe
#number of sample sizes(6)*number of regressions(1000) = 6000
res <- data.frame(
  sample_size = numeric(6000),
  sim = integer(6000),
  beta0 = numeric(6000),
  beta1 = numeric(6000),
  beta2 = numeric(6000)
)
N <- nrow(pop) #Number of observations in the population
row <- 1 #Starting at row 1

#Creating a FOR loop to run the code for each sample size
for (n in n_samp) {
  #Creating a FOR loop to repeat the regression 1000 times
  for (i in 1:n_sim) {
  #Randomly determining which observations should be drawn from the population.
    samp_indices <- sample(1:N, size = n, replace = FALSE)
    samp <- pop[samp_indices,] #Creating the sample

    #Fitting the linear model for each sample
    model <- lm(y ~ x + z, data = samp)
    coefs <- coef(model) #Collecting the coefficients for each model

    #Saving results
    res$sample_size[row] <- n #denoting the sample size
    res$sim[row] <- i #denoting the regression iteration
    res$beta0[row] <- coefs["(Intercept)"] #Saving coefficients
    res$beta1[row] <- coefs["x"]
    res$beta2[row] <- coefs["z"]

    row <- row + 1 #Denoting the move to the next row
  }
}

#Converting sample size into a categorical variable
res$sample_size <- as.factor(res$sample_size)

#Creating a plot to visualize the sampling distribution for each sample size
ggplot(res, aes(x = beta1, color = sample_size)) +
  geom_density() +
  labs(
    x = "Estimates of b1",
    y = "Density",
    color = "Sample size",
```

```
    title = "Demonstration of CLT - Sampling distribution of beta1"
  ) +
  theme_minimal()
```




The density plot demonstrates the Central Limit Theorem by depicting the density distributions for different sample sizes. We can see that as the sample size increases, the distribution of coefficients approximates a normal distribution.

**Question b.** Compute the bootstrapped standard error for the coefficient of the treatment variable.

```
#Creating a new results dataframe
res1 <- data.frame(
  sample_size = numeric(6000),
  sim = integer(6000),
  beta0 = numeric(6000),
  beta1 = numeric(6000),
  beta2 = numeric(6000)
)

N <- nrow(pop) #Number of observations in the population
row <- 1 #Starting at row 1

#Creating a FOR loop to run the code for each sample size
for (n in n_samp) {
  #Creating a FOR loop to repeat the regression 1000 times
  for (i in 1:n_sim) {
```

```
  #Randomly determining which observations should be drawn from the population.
    samp_indices1 <- sample(1:N, size = n, replace = TRUE)
    samp1 <- pop[samp_indices1,] #Creating the sample

    #Fitting the linear model for each sample
    model1 <- lm(y ~ x + z, data = samp1)
    coefs1 <- coef(model1) #Collecting the coefficients for each model

    #Saving results
    res1$sample_size[row] <- n #denoting the sample size
    res1$sim[row] <- i #denoting the regression iteration
    res1$beta0[row] <- coefs1["(Intercept)"] #Saving coefficients
    res1$beta1[row] <- coefs1["x"]
    res1$beta2[row] <- coefs1["z"]

    row <- row + 1 #Denoting the move to the next row
  }
}

print(sd(res1$beta1))
```

```
## [1] 0.1173851
```

Computing the bootstrapped standard error involves the following steps: 1. Collect data and define statistic. 2. Resample data with replacement. 3. Calculating the statistic for each sample. 4. Calculating the standard deviation of the calculated statistics.

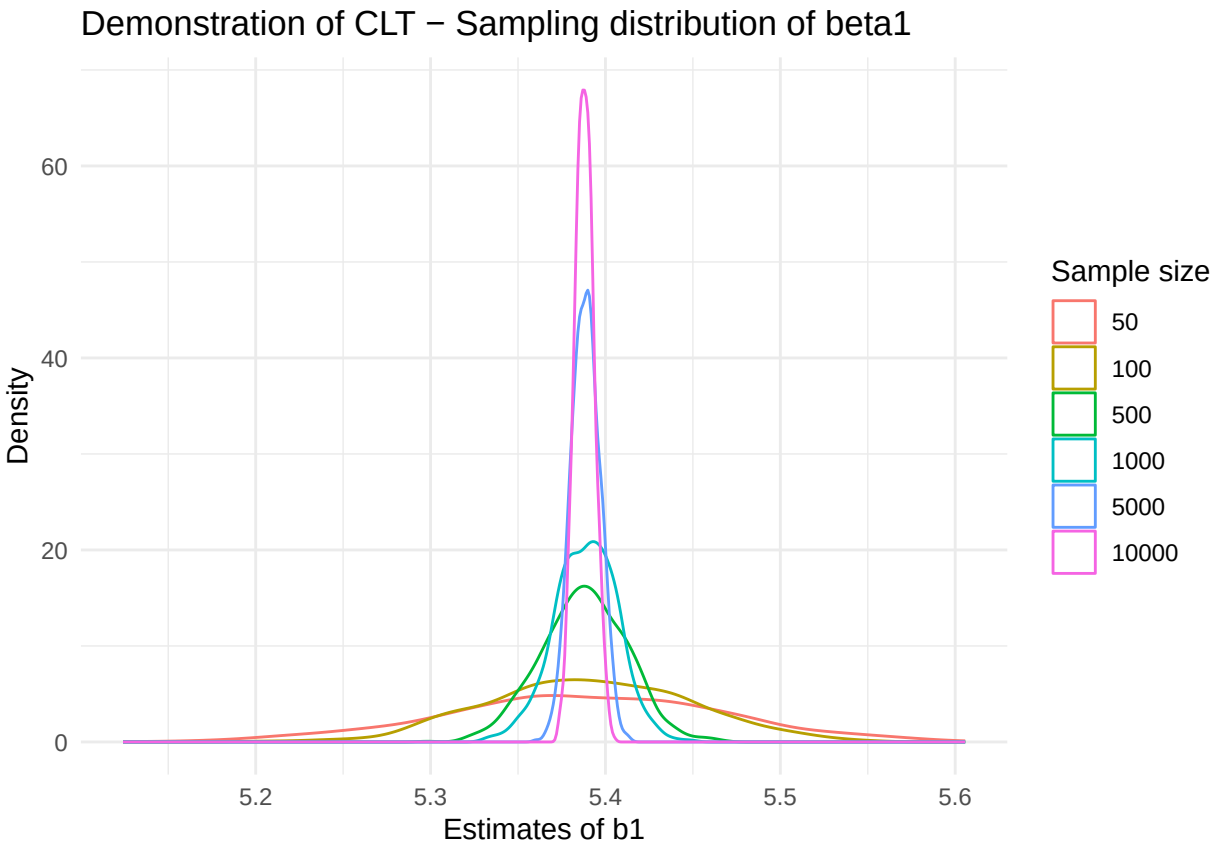
Based on the above steps, the bootstrapped standard error is 0.1173851

### Question 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 coeficient's sampling distribution?

```
#Creating a different results dataframe
#number of sample sizes(6)*number of regressions(1000) = 6000
res2 <- data.frame(
  sample_size = numeric(6000),
  sim = integer(6000),
  beta0 = numeric(6000),
  beta1 = numeric(6000),
  beta2 = numeric(6000)
)

N <- nrow(pop) #Number of observations in the population
row <- 1 #Starting at row 1

#Creating a FOR loop to run the code for each sample size
for (n in n_samp) {
  #Creating a FOR loop to repeat the regression 1000 times
  for (i in 1:n_sim) {
  #Randomly determining which observations should be drawn from the population.
```

```r
    samp_indices <- sample(1:N, size = n, replace = FALSE)
    samp2 <- pop[samp_indices,] #Creating the sample

    #Fitting the linear model for each sample
    model2 <- lm(y ~ x, data = samp2)
    coefs2 <- coef(model2) #Collecting the coefficients for each model

    #Saving results
    res2$sample_size[row] <- n #denoting the sample size
    res2$sim[row] <- i #denoting the regression iteration
    res2$beta0[row] <- coefs2["(Intercept)"] #Saving coefficients
    res2$beta1[row] <- coefs2["x"]
    res2$beta2[row] <- coefs2["z"]

    row <- row + 1 #Denoting the move to the next row
  }
}

res2$sample_size <- as.factor(res$sample_size) #Converting sample size into a categorical variable

#Creating a plot to visualize the sampling distribution for each sample size
ggplot(res2, aes(x = beta1, color = sample_size)) +
  geom_density() +
  labs(
    x = "Estimates of b1",
    y = "Density",
    color = "Sample size",
    title = "Demonstration of CLT - Sampling distribution of beta1"
  ) +
  theme_minimal()
```




The sampling distribution for the linear regression coefficients without controlling for the confounder shows a bias aand its centered around a value different from the actual regression coefficient. Therefore, this shows that statistical tests that are predicated on properties of a normal distribution are unreliable if there is bias in the model.

**Part 2: Data Analysis**

**Question a** 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.

```
#Loading STAR Dataset from Mike Burnham's Github pols_602 repository at https://github.com/MLBurnham/po
star <- read.csv("https://raw.githubusercontent.com/MLBurnham/pols_602/refs/heads/main/data/STAR.csv")

#Creating a new numeical treatment variable
star$treatment <- ifelse(star$classtype == "small", 1, 0)

#Creating new datasets to hold treatment and control groups
str_treatment <- star[star$treatment == 1, ]
str_control <- star[star$treatment == 0, ]

#Running Difference in Means Test Using Welch Two Sample T-Test
two_tailed <- t.test(str_treatment$math, str_control$math, mu = 0, alternative = "two.sided", conf.leve
print(two_tailed)
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  str_treatment$math and str_control$math
## t = 2.7404, df = 1219.3, p-value = 0.006227
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   1.701545 10.278265
## sample estimates:
## mean of x mean of y
##  634.8274  628.8374
```

I am using the STAR dataset that looks at the relationship between scores and classroom sizes. I hypothesize that small classroom sizes (treatment) has an effect on the math scores. I use the t-statistic cause of the low sample size. I find that the t-statistic is 2.7404, which means that my observed difference in means value is 2.7404 standard deviations away from a t-statistic sampling distribution centered at 0. The probability of seeing this value occur by chance is 0.006227. Therefore, we can reasonably conclude that the observed difference in means value is statistically significant.

**Question b.** Using the same data, fit a linear model. Interpret the coefficient, standard error, tvalue, and p-value.

```
model <- lm(math ~ treatment, data = star)
summary(model)
```

```
##
## Call:
## lm(formula = math ~ treatment, data = star)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.827  -27.585   -0.827   26.163  145.163
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  628.837      1.476  426.09  < 2e-16 ***
## treatment      5.990      2.178    2.75  0.00604 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.74 on 1272 degrees of freedom
## Multiple R-squared:  0.005911,   Adjusted R-squared:  0.00513
## F-statistic: 7.564 on 1 and 1272 DF,  p-value: 0.006039
```

Putting a child into a small classroom (treatment condition) has an average effect of 5.990 on his math scores. On a t-statistic sampling distribution centered at 0, this value is 2.75 standard deviations away. Therfore, this is a 0.00604 % probability that we got this value by chance due to sampling variance.