---

title: "A2 Q3"

author: "Zhengyang Fei"

format: html

editor: visual

```{r}

# Setup

set.seed(1)

nobs <- 250 # Run over: 250, 500

m <- 1000

mboot <- 100

plots <- FALSE

theta <- 0.5

```

```{r}

#| echo: false

#|

# Create empty variables for simulation

star_coef <- matrix(NA, m, 1)

star_se <- matrix(NA, m, 1)

tilde_coef <- matrix(NA, m, 1)
```

```
tilde_se <- matrix(NA, m, 1)

# bootstrap
bootmeans <- matrix(NA, m, 1)
bootses <- matrix(NA, m, 1)
bootcil <- matrix(NA, m, 1)
bootciu <- matrix(NA, m, 1)

bootmeans_t <- matrix(NA, m, 1)
bootses_t <- matrix(NA, m, 1)
bootcil_t <- matrix(NA, m, 1)
bootciu_t <- matrix(NA, m, 1)
```

```{r}
#| echo: false
#|
for (j in 1:m) {
  A <- rnorm(nobs, 0, 1)
  U <- rnorm(nobs, 0, 1)
  t <- rnorm(nobs, 0, 1)
  e <- rnorm(nobs, 0 ,1)
  Z <- 0.5*A + U + t
  Y <- theta*Z + U + e

  # Estimator 1: theta hat star
```

```
mod1 <- lm(Y ~ Z)
coef_vector <- as.numeric(summary(mod1)$coefficients[, "Estimate"])
se_vector <- as.numeric(summary(mod1)$coefficients[, "Std. Error"])

## Getting estimates for Estimator 1
star_coef[j, 1] <- coef_vector[2]
star_se[j, 1] <- se_vector[2]

# Estimator 2: theta tilde
mod2 <- lm(Y ~ A)
coef1 <- as.numeric(summary(mod2)$coefficients[, "Estimate"])
mod3 <- lm(Z ~ A)
coef2 <- as.numeric(summary(mod3)$coefficients[, "Estimate"])

## Getting estimates for Estimator 2
tilde_coef[j, 1] <- coef1[2] / coef2[2]
tilde_se[j, 1] <- sqrt((var(Y-tilde_coef[j, 1]*Z)) / (nobs * var(Z) * cor(Z, A)^2))

# Bootstrap
bm <- rep(NA, mboot)
bm.tilde <- rep(NA, mboot)
for (k in 1:mboot) {
  bootidx <- sample(1:nobs, size = nobs, replace=TRUE)
  aboot <- A[bootidx]
  zboot <- Z[bootidx]
  yboot <- Y[bootidx]
```

```
 # Bootstrap Estimator 1
 mod4bs <- lm(yboot~ zboot)
 estimated_coef <- summary(mod4bs)$coefficients[, "Estimate"]
 coeff_vectors <- as.numeric(estimated_coef)
 bm[k] <- coeff_vectors[2]

 # Bootstrap Estimator 2
 mod5bs <- lm(yboot ~ aboot)
 estimated_coef <- summary(mod5bs)$coefficients[, "Estimate"]
 mod5bs_coef_vec <- as.numeric(estimated_coef)

 mod6bs <- lm(zboot ~ aboot)
 estimated_coef <- summary(mod6bs)$coefficients[, "Estimate"]
 mod6bs_coef_vec <- as.numeric(estimated_coef)

 bm.tilde[k] <- mod5bs_coef_vec[2] / mod6bs_coef_vec[2]
}

bootmeans[j, 1] <- mean(bm)
bootses[j, 1] <- sd(bm)
bootcil[j, 1] <- quantile(bm, probs = c(0.025))
bootciu[j, 1] <- quantile(bm, probs = c(0.975))

bootmeans_t[j, 1] <- mean(bm.tilde)
bootses_t[j, 1] <- sd(bm.tilde)
```

```
  bootcil_t[j, 1] <- quantile(bm.tilde, probs = c(0.025))
  bootciu_t[j, 1] <- quantile(bm.tilde, probs = c(0.025))

  print(j)
}
```

```{r}
#| echo: false
# For Estimator 1
starthetamean <- colMeans(star_coef)

results <- cbind(
  m,
  nobs,
  starthetamean,
  starthetamean - theta,
  apply(star_coef, 2, sd),
  apply(star_coef, 2, sd)/ sqrt(m),
  colMeans(star_se))
colnames(results) <- c('No. runs', 'nobs', 'Mean', 'Bias', 'SD', 'MCE', 'Mean SE' )


# 95% coverage probability
cover <- matrix(NA, m, 2)
```

```
# Normal approximation CI coverage:

cover <- (star_coef + qnorm(0.975) * star_se) > theta & (star_coef + qnorm(0.025) * star_se) < theta

results <- cbind(results, colMeans(cover))

colnames(results)[ncol(results)] <- c('Normal CI coverage')


# Bootstrap standard error

results <- cbind(results, colMeans(bootses))

colnames(results)[ncol(results)] <- c('Mean BS SE')


# Bootstrap normal approximation CI coverage:

cover <- (star_coef + qnorm(0.975) * bootses) > theta & (star_coef + qnorm(0.025) * bootses) < theta

results <- cbind(results, colMeans(cover))

colnames(results)[ncol(results)] <- c('Normal BS CI coverage')


# Bootstrap percentile confidence intervals

results <- cbind(results, colMeans(bootcil))

colnames(results)[ncol(results)] <- c('BS Percent lower')

results <- cbind(results, colMeans(bootciu))

colnames(results)[ncol(results)] <- c('BS Percent upper')


results <- round(results, 3)

print("theta_star")

results

```
```
```

```{r}
#| echo: false
tildethetamean <- colMeans(tilde_coef)

results <- cbind(
 m,
 nobs,
 tildethetamean,
 tildethetamean - theta,
 apply(tilde_coef, 2, sd),
 apply(tilde_coef, 2, sd)/ sqrt(m),
 colMeans(tilde_se))
colnames(results) <- c('#runs', 'nobs', 'Mean', 'Bias', 'SD', 'MCE', 'Mean SE' )



# 95% coverage probability
cover <- matrix(NA, m, 2)

# Normal approximation CI coverage:
cover <- (tilde_coef + qnorm(0.975) * tilde_se) > theta & (tilde_coef + qnorm(0.025) * tilde_se) < theta
results <- cbind(results, colMeans(cover))
colnames(results)[ncol(results)] <- c('Normal CI coverage')

# Bootstrap standard error
```

```
results <- cbind(results, colMeans(bootses_t))

colnames(results)[ncol(results)] <- c('Mean BS SE')


# Bootstrap normal approximation CI coverage:

cover <- (tilde_coef + qnorm(0.975) * bootses_t) > theta & (tilde_coef + qnorm(0.025) * bootses_t) < theta

results <- cbind(results, colMeans(cover))

colnames(results)[ncol(results)] <- c('Normal BS CI coverage')


# Bootstrap percentile confidence intervals

results <- cbind(results, colMeans(bootcil_t))

colnames(results)[ncol(results)] <- c('BS Percent lower')

results <- cbind(results, colMeans(bootciu_t))

colnames(results)[ncol(results)] <- c('BS Percent upper')


results <- round(results, 3)

print("theta_tilde")

results
```