

ECON 7201

Applied Econometrics

Emmanuel

Assignment 2

Due Date

Sunday October 5, 2025 at 11:59 PM

Directions

Answer all questions. Submit both a PDF and Quarto file to the nexus assignment portal.

Git and GitHub

- (a) Create a new R project in your **econ_3201** directory called **assignment_2**.
- (b) Download the assignment PDF and Quarto file the **assignment_2** folder.
- (c) Commit and push the changes to your **econ_3201** repository on [GitHub.com](https://github.com).

Ans. I have created a new R project (Assignment 2) and pushed all the changes to my repository on github.com (https://github.com/Emmanuel-spec493/econ_3201)

LaTeX

Matrices are created in LaTeX using the `\begin{bmatrix}...\end{bmatrix}` command. To separate entries along the same row, use `&`. To end a line, use `\\`. To make vertical ellipses (:), use `\vdots`. Practice writing the following matrices and vectors in LaTeX. Write the following matrices in LaTeX.

2.

$$(a) X'X = \begin{bmatrix} n & \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{2i} \\ \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{1i}^2 & \sum_{i=1}^n x_{1i}x_{2i} \\ \sum_{i=1}^n x_{2i} & \sum_{i=1}^n x_{1i}x_{2i} & \sum_{i=1}^n x_{2i}^2 \end{bmatrix}$$

$$(b) \Omega = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$$

R

3. In this question we compare standard errors based on (incorrect) asymptotic assumptions with those based on alternate (appropriate) estimator (White). Consider one sample drawn from the following data generating process (DGP) which we will simulate in R:

(a) Compute the OLS estimator of β_2 and its standard error using the `lm()` command in R for the model $y_i = \beta_1 + \beta_2 x_i + \epsilon_i$ based on the DGP given above.

```
set.seed(123)
n <- 25
x <- rnorm(n, mean = 0, sd = 1)
beta0 <- 1
beta1 <- 0
dgp <- beta0 + beta1 * x
e <- x^2 * rnorm(n, mean = 0, sd = 1)
y <- dgp + e
```

$$\beta_2 = 0.461$$

```
model <- lm(y ~ x)
```

Standard error

```
se <- summary(model)$coef[, "Std. Error"]
```

$$\text{se of } \beta_2 = 0.2563086$$

(b) Next, compute the standard error of $\hat{\beta}_2$ by computing $\hat{\sigma}^2(X'X)^{-1}$ in R using matrix commands, and verify that the two standard error estimates are identical. Ans.

The standard errors using matrix command

```

X <- cbind(1, x)

beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y

# Residual variance

u_hat <- y - X %*% beta_hat

sigma2_hat <- as.numeric(t(u_hat) %*% u_hat / (n - ncol(X)))

# variance-covariance matrix

Var_beta_hat <- sigma2_hat * solve(t(X) %*% X)

# Standard errors
se_beta_hat <- sqrt(diag(Var_beta_hat))

```

se of $\beta_2 = 0.2563086$

Checking to see if standard errors are same for β_2 using the two methods

```
all.equal(as.numeric(se_beta_hat["x"]), as.numeric(summary(model)$coef["x", "Std. Error"]))
```

[1] TRUE

- (c) Compute White's heteroskedasticity consistent covariance matrix estimator using matrices in R and report the White estimator of the standard error of $\hat{\beta}_2$. Compare this with that from 3 (a) above.

```

# Diagonal matrix of squared residuals
e2 <- diag(as.vector(u_hat^2))

# Covariance matrix
V_White <- solve(t(X) %*% X) %*% t(X) %*% e2 %*% X %*% solve(t(X) %*% X)

# White standard errors
se_White <- sqrt(diag(V_White))

```

(se_White) of $\beta_2 = 0.4529172$

```
# Comparing se_White and se of beta_2

all.equal(as.numeric(se_White["x"]), as.numeric(summary(model)$coef["x", "Std. Error"]))

[1] "Mean relative difference: 0.4340939"
```

The OLS SE is higher than the White SE in this case

4. Let $\hat{\theta}$ be an estimator for the population parameter θ . $\hat{\theta}$ is said to be unbiased if $E(\hat{\theta}) = \theta$. That is, if the mean of the sampling distribution of $\hat{\theta}$ is equal to the true population value.

Consider the model

$$y_i = \beta_0 + \beta_1 x_{1,i} - \beta_2 x_{2,i} + \epsilon_i.$$

Lets provide empirical evidence that the ordinary least squares estimators $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ are unbiased estimators of β_0 , β_1 , β_2 , respectively, using R.

- (a) Set the seed to 1, i.e., `set.seed(1)`.

```
set.seed(1)
```

- (b) Set the number of observations $n = 100$

```
n <- 100
```

- (c) Generate the following model

$$y_i = 2 + 3.5x_{1,i} - 9.2x_{2,i} + \epsilon_i,$$

where $x_1 \sim N(3, 6)$, $x_2 \sim N(2, 4)$, and $\epsilon \sim N(0, 100)$. To create a normally distributed variable, use the `rnorm(n, mean, sd)` command in R.

```
x1 <- rnorm(n, mean = 3 , sd = sqrt(6))
x2 <- rnorm(n, mean = 2 , sd = sqrt(4))
epsilon <- rnorm(n, mean = 0 , sd = sqrt(100))

y <- 2+3.5*x1-9.2*x2+epsilon
```

- (d) Estimate the model coefficients using the `lm()` command. (Search `?lm()` in the console for more info).

```
model <- lm(y ~x1 + x2)
```

- (e) Using a `for()` loop, replicate the model above $M = 1000$ times and save the coefficient estimates from each iteration.

Number of repetitions

```
set.seed(1)
N <- 1000
n <- 100
x1 <- rnorm(N,mean =3 , sd = sqrt(6))
x2 <- rnorm(N,mean =2 , sd = sqrt(4))
epsilon <- rnorm(N,mean =0 , sd = sqrt(100))

X <- cbind(1,x1 , x2)
num_sim <- 1000
y <- 2+3.5*x1-9.2*x2+epsilon
beta0_estimates <- numeric(num_sim)
beta1_estimates <- numeric(num_sim)
beta2_estimates <- numeric(num_sim)

for (i in 1:num_sim) {
  sample_indices <- sample(1:N , n)
  X_sample <- X[sample_indices, ]
  y_sample <- y[sample_indices]

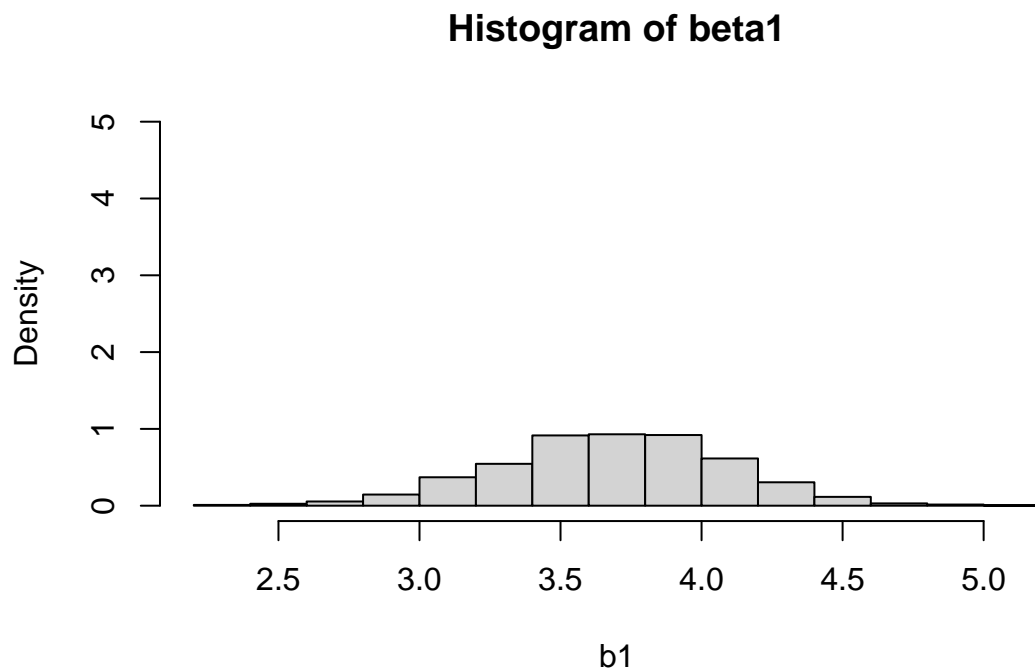
  beta_hat <- solve(t(X_sample)%*%X_sample)%*%t(X_sample)%*%y_sample

  beta0_estimates[i] <- beta_hat[1]
  beta1_estimates[i] <- beta_hat[2]
  beta2_estimates[i] <- beta_hat[3]
}
```

- (f) Using `hist()`, plot the sampling distributions of the coefficient estimates, β_1 and β_2 .

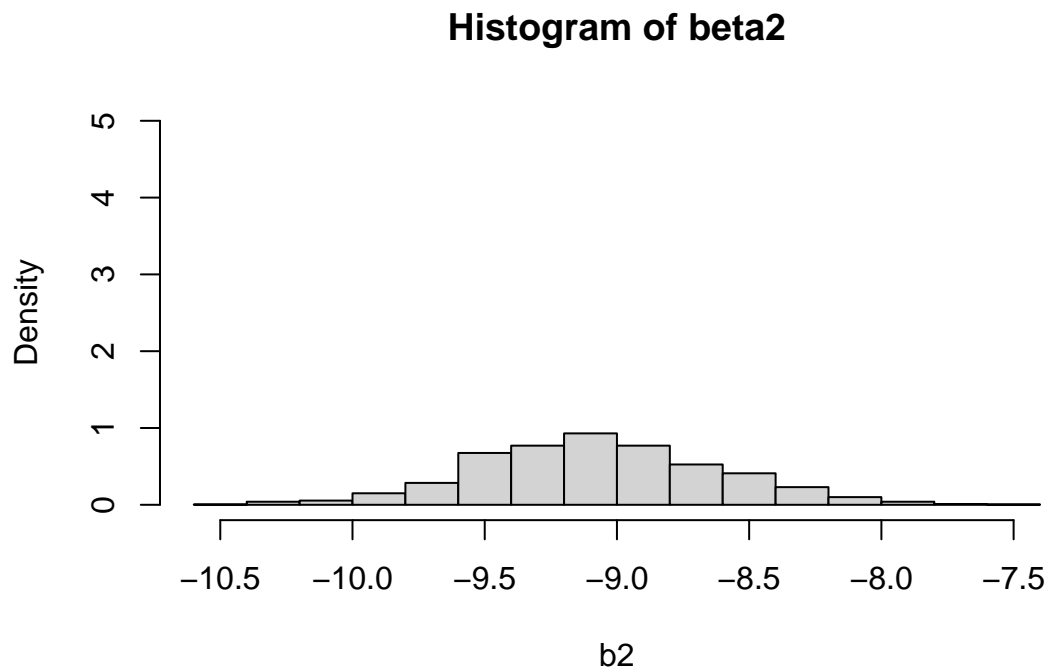
Histogram for Beta 1

```
hist(beta1_estimates,  
     probability = TRUE,  
     ylim = c(0 , 5),  
     main = "Histogram of beta1",  
     xlab = "b1")
```



Histogram for Beta 2

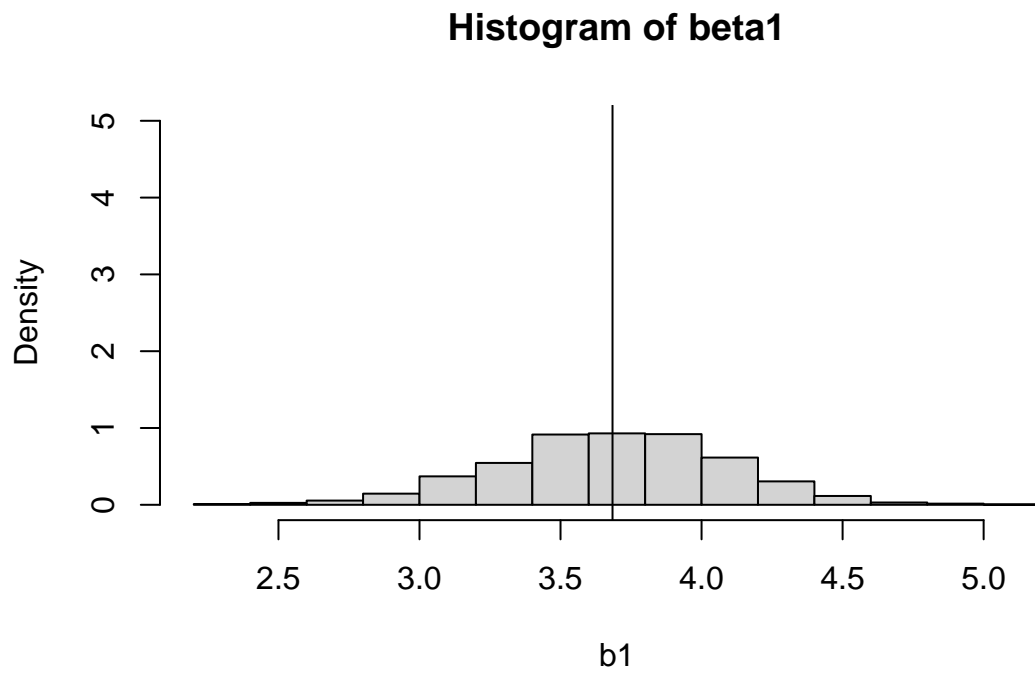
```
hist(beta2_estimates,  
     probability = TRUE,  
     ylim = c(0 , 5),  
     main = "Histogram of beta2",  
     xlab = "b2")
```



- (g) Add a vertical line to each figure at the mean of the respective variable. Search `?abline()` in your console.

Adding A vertical line for Beta 1

```
hist(beta1_estimates,  
probability = TRUE,  
ylim = c(0, 5),  
main = "Histogram of beta1",  
xlab = "b1")  
abline(v=mean(beta1_estimates))
```



Adding A Vertical line for Beta 2

```
hist(beta2_estimates,  
probability = TRUE,  
ylim = c(0 , 5),  
main = "Histogram of beta2",  
xlab = "b2")  
abline(v=mean(beta2_estimates))
```