

ECON 7201

Applied Econometrics

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).

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 (\vdots), 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
```

$\beta_2 = 0.889$

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

Standard error

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

- (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)
y_vec <- matrix(y, ncol = 1)
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y_vec
u_hat <- y_vec - X %*% beta_hat
sigma2_hat <- sum(u_hat^2) / (n - ncol(X))
cov_beta_hat <- sigma2_hat * solve(t(X) %*% X)
```

The standard errors are

```
se_beta_hat <- sqrt(diag(cov_beta_hat))
```

```
ols_model <- lm(y ~ x)
coef_lm <- coef(ols_model)
X <- cbind(1, x)
y_vec <- matrix(y, ncol = 1)
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y_vec
```

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

```
all.equal(as.numeric(beta_hat), as.numeric(coef_lm))
```

- (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.

Ans. # Diagonal matrix of squared residuals

```
e1 <- diag(as.vector(u_hat^2))
```

```
V_White <- solve(t(X) %*% X) %*% t(X) %*% e1 %*% X %*% solve(t(X) %*% X)
```

White standard errors

```
se_White <- sqrt(diag(V_White))
```

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 $x_1 \sim N(3, 6)$ and $x_2 \sim N(2, 4)$ and $\epsilon_i \sim N(0, 100)$.

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

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

```
::: {.cell}
```

```
```{r .cell-code}
```

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

```
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

```
n <- 100
M <- 1000
```

Create vectors to store coefficients

```
beta0_hat <- numeric(M)
beta1_hat <- numeric(M)
beta2_hat <- numeric(M)
```

## Generate new random data

```

for (m in 1:M) {
 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

 #Estimation of model

 model <- lm(y~ x1 + x2)

 # Save coefficients

 beta0_hat[m] <- coef(model)[1]
 beta1_hat[m] <- coef(model)[2]
 beta2_hat[m] <- coef(model)[3]
}

```

(f) Using `hist()`, plot the sampling distributions of the coefficient estimates,  $\beta_1$  and  $\beta_2$ .

## Histogram for Beta 1

```

hist(beta1_hat,
 breaks = 18,
 col = "blue",
 main = expression("Sampling Distribution of " * hat(beta)[1]),
 xlab = expression(hat(beta)[1]))

```

## Histogram for Beta 2

```

hist(beta2_hat,
 breaks = 18,
 col = "orange",
 main = expression("Sampling Distribution of " * hat(beta)[2]),
 xlab = expression(hat(beta)[2]))

```

- (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
```

```
::: {.cell}
```

```
```{r .cell-code}
```

```
hist(beta1_hat,
      breaks = 18,
      col = "green",
      main = expression("Sampling Distribution of " * hat(beta)[1]),
      xlab = expression(hat(beta)[1]))
abline(v = mean(beta1_hat),
      col = "orange",
      lwd = 2)
```

```
:::
```

Adding A Vertical line for Beta 2

```
hist(beta2_hat,
      breaks = 18,
      col = "green",
      main = expression("Sampling Distribution of " * hat(beta)[2]),
      xlab = expression(hat(beta)[2]))
abline(v = mean(beta2_hat), col = "orange", lwd = 3)
abline(v = -9.2, col = "darkgreen", lwd = 3, lty = 3)
```