|Name : Oluchi Jessica Onyekwere |Student ID: 3192227 | | 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

1. (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.

My assignment here

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 elipses ($\vdots$), use `\vdots`. Practice writing the following matrices and vectors in LaTeX. Write the following matrices in LaTeX.

2.

(a)

$$X'X = \begin{bmatrix} n & \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{2i} \\ \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{1i}^2 & \sum^n i = 1 x_{1i} x_{2i} \\ \sum_{i=1}^n x_{2i} & \sum_{i=1}^n x_{1i} x2i & \sum_1^n x_{2i}^2 \end{bmatrix}$$

(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`:

```r
set.seed(123)
n <- 25
x <- rnorm(n,mean=0.0,sd=1.0)
beta0 <- 1
beta1 <- 0
## x is irrelevant in this model, the data generating process is as follows:
dgp <- beta0 + beta1*x
## The residual is heteroskedastic by construction
e <- x^2*rnorm(n,mean=0.0,sd=1.0)
y <- dgp + e
```

   (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.

```r
lm(y ~ x)
```

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      0.889        0.461
```

(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.

```
X <- cbind(1,x)
beta_hat2 <- solve(t(X)%*% X) %*% t(X) %*% y

cbind(matrix =as.numeric(beta_hat2),
      lm = coef(lm(y~x)))
```

```
               matrix        lm
(Intercept) 0.8889841 0.8889841
x           0.4610238 0.4610238
```

(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.

```
resid <- y-X%*% beta_hat2

white_beta_hat2 <- matrix(0,
                 nrow = 2,
                 ncol = 2)
for(i in 1:n) {
  white_X <- X[i, , drop=FALSE]
  white_beta_hat2 <- white_beta_hat2 + as.numeric(resid[i]^2) * t(white_X) %*% white_X
}

white_Resid <- solve(t(X) %*% X)
white_Result <- white_Resid %*%  white_beta_hat2 %*% white_Resid

SE_white_beta_hat2 <- sqrt(white_Result[2,2])

beta_hat2
```

```
       [,1]
  0.8889841
x 0.4610238
```

```
SE_white_beta_hat2
```

```
[1] 0.4529172
```

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
```

```
x1 <- rnorm(n, mean=3, sd=6)
x2 <- rnorm(n, mean=2, sd=4)
epsilon <- rnorm(n,mean=0, sd=100)
y <- 2 + 3.5 * x1 -9.2 * x2 + epsilon
model <- lm(y ~ x1 + x2)
summary(model)
```

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
     Min       1Q   Median       3Q      Max 
-294.359  -43.645    0.202   63.692  263.941 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)    6.153     13.605   0.452 0.652071
x1             3.852      1.946   1.979 0.050596 .
x2           -10.537      2.737  -3.850 0.000212 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104.3 on 97 degrees of freedom
Multiple R-squared:  0.162, Adjusted R-squared:  0.1447
F-statistic: 9.377 on 2 and 97 DF,  p-value: 0.0001892
```

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

```r
lm(y~x1 + x2) $coefficient
```

```
(Intercept)          x1          x2
   6.153175    3.851836  -10.536670
```

```
(e) Using a `for()` loop, replicate the model above $M=1000$ times and save the coefficient e
```

```r
set.seed(1)
n = 100
M = 1000

coef_estimates <- matrix(0, nrow=M, ncol=3)
colnames(coef_estimates) <-c("intercept","x1_coef","x2_coef" )

for(i in 1:M){
x1 <- rnorm(n, mean = 3, sd = 6)
x2 <- rnorm(n, mean = 2, sd = 4)
epsilon <- rnorm(n, mean = 0, sd=100)
y <-  2 + 3.5 * x1 -9.2 * x2 + epsilon

model <- lm(y ~ x1 + x2)
coef_estimates[i,] <-coef(model)

}

summary(model)
```

```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-237.36  -61.37   18.30   58.46  207.67 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -16.822     11.283  -1.491  0.13922   
x1             2.759      1.361   2.028  0.04531 * 
x2            -6.237      2.212  -2.819  0.00584 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 87.71 on 97 degrees of freedom
Multiple R-squared:  0.1235,    Adjusted R-squared:  0.1055 
F-statistic: 6.837 on 2 and 97 DF,  p-value: 0.001669
```

```r
lm(y~x1+x2)$coefficients
```

```
(Intercept)          x1          x2 
 -16.821915    2.759031   -6.236846 
```

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

```r
hist(coef_estimates[, "x1_coef"],
     main = "Sampling Distribution from 1000 iterations",
     xlab = x1,
     col = "lightblue",)
```

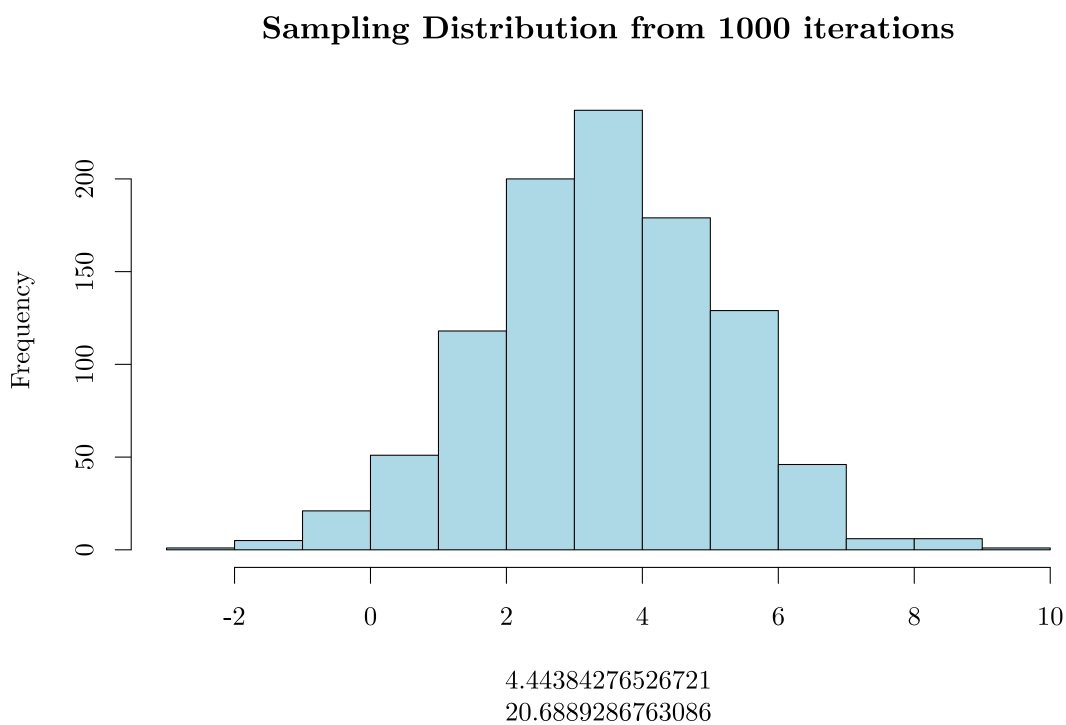

Figure 1: Sampling Distribution from 1000 iterations for x1_coef

```r
hist(coef_estimates[, "x2_coef"],
     main = "Sampling Distribution from 1000 iterations",
     xlab = x2,
     col = "pink")
```

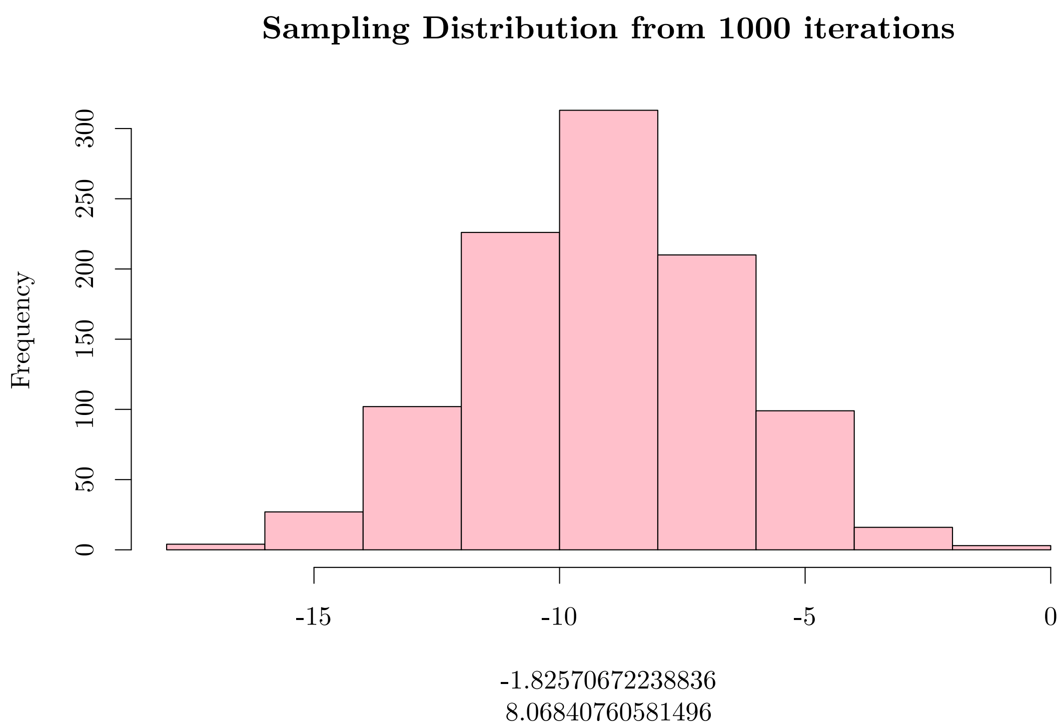

Figure 2: Sampling Distribution from 1000 iterations for x2_coef

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

```
hist(coef_estimates[, "x1_coef"],
     main =  "Sampling Distribution from 1000 iterations",
     xlab = x1,
     col = "lightblue",)


abline(v = mean(coef_estimates[,"x1_coef"]),
       col="black",
       lwd=2,
       lty=2)
```

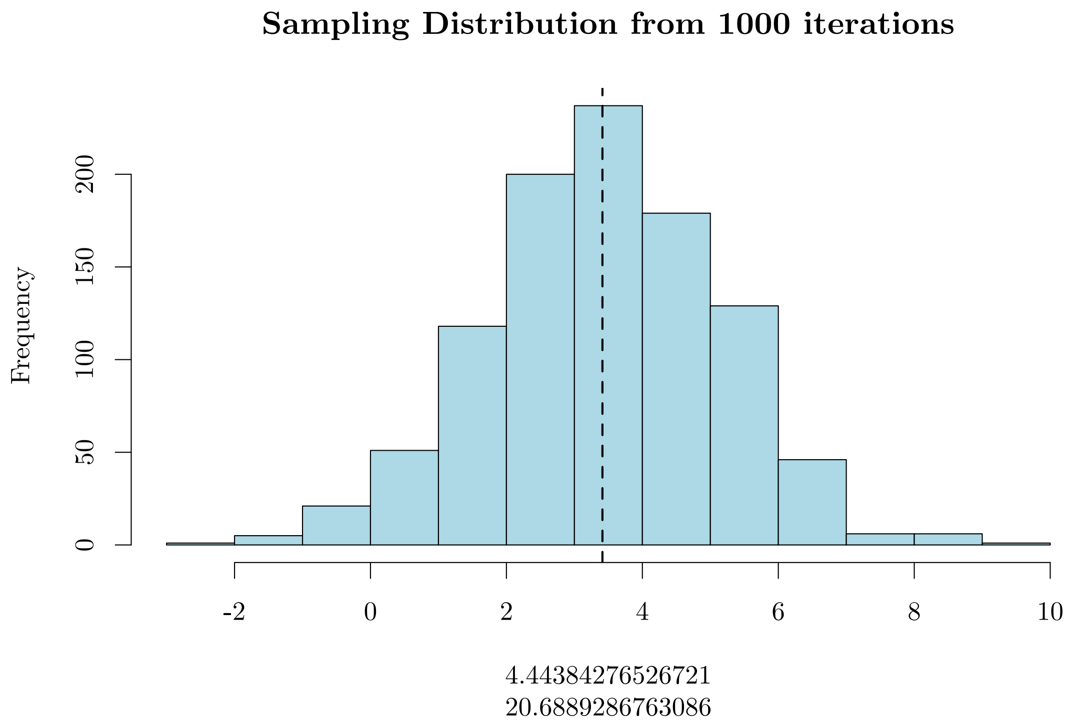

Figure 3: Sampling Distribution from 1000 iterations with the vertical line drawn at the mean of each respective variable for x1_coef

```
hist(coef_estimates[, "x2_coef"],
     main = "Sampling Distribution from 1000 iterations",
     xlab = x2,
     col = "pink")

abline(v=mean(coef_estimates[, "x2_coef"]),
       col="purple",
       lwd=2,
       lty=2)
```

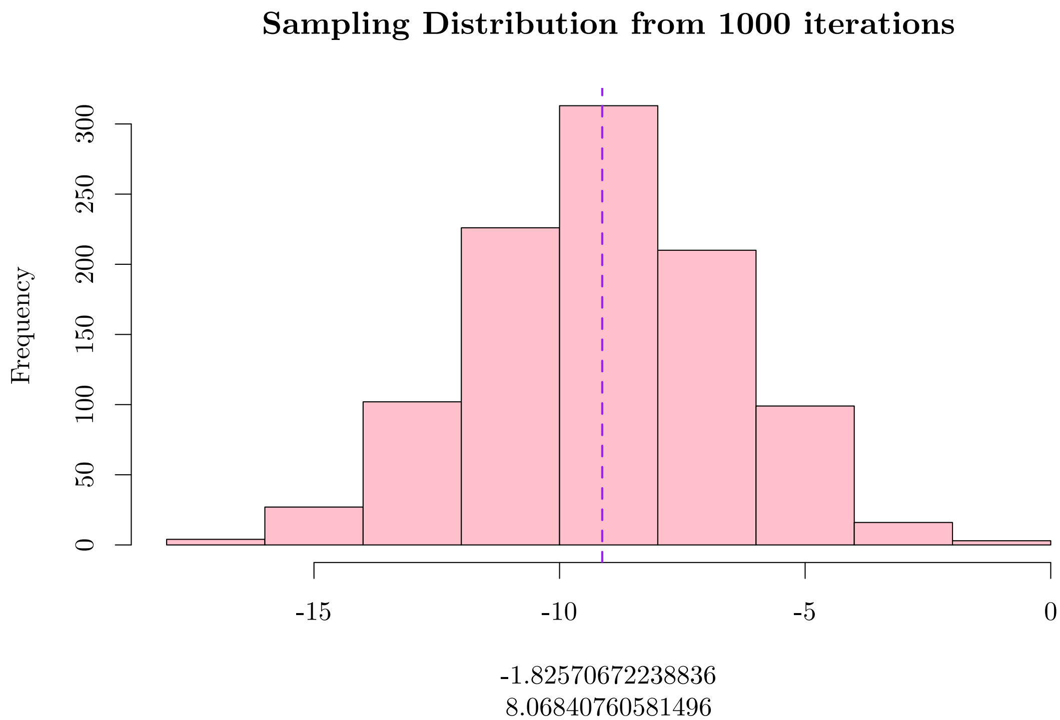

Figure 4: Sampling Distribution from 1000 iterations with the vertical line drawn at the mean of each respective variable for x2_coef