Name: Christopher Allwood
Student ID: 3174698

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.

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)

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

(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:

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

summary(OLS_Estimator)
```

```

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.8764 -0.1604  0.0928  0.5748  2.1268 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   0.8890     0.2379   3.737  0.00108 **
x             0.4610     0.2563   1.799  0.08520 . 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.189 on 23 degrees of freedom
Multiple R-squared:  0.1233,	Adjusted R-squared:  0.0852 
F-statistic: 3.235 on 1 and 23 DF,  p-value: 0.0852
```

(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.

```r
X <- cbind(1, x)
residuals <-resid(OLS_Estimator)
sigmasqr_hat <-sum(residuals^2/(n-2))
trasposeX.Xinv <-solve(t(X) %*% X)
VarianceMatrix <-sigmasqr_hat * trasposeX.Xinv
beta2_se <-sqrt(VarianceMatrix[2,2])

#Comparing the result of beta_se with the standard error of x from the OLS_Estimates regress

beta2_se
```

```
[1] 0.2563086
```

```r
coef(summary(OLS_Estimator))["x", "Std. Error"]
```

```
[1] 0.2563086
```

```r
cbind(matrix= as.numeric(beta2_se), coef(summary(OLS_Estimator))["x", "Std. Error"])
```

```
        matrix          
[1,] 0.2563086 0.2563086
```

(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.

```
White_X <- cbind(1,x)
transposeX.Xinv <-solve(t(X)%*%X)

#Extracting the residuals from the original regression output

resid_extract <-resid(OLS_Estimator)

# Transforming the residuals
resid_extract_hat <-diag(resid_extract^2)

White_resid <- t(White_X)%*% resid_extract_hat %*% White_X

CovWhite <- transposeX.Xinv %*% White_resid %*% transposeX.Xinv

# Extracting the standard corrected standard errors from

White_SE_Beta2 <- sqrt(CovWhite[2,2])

# Comparing results

White_SE_Beta2
```

```
[1] 0.4529172
```

```
coef(summary(OLS_Estimator))["x", "Std. Error"]
```

```
[1] 0.2563086
```

```
cbind(White_SE_Beta2,coef(summary(OLS_Estimator))["x", "Std. Error"])
```

```
     White_SE_Beta2
[1,]      0.4529172 0.2563086
```

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)`.

```
#nrm(list=ls())
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
```

```
x_1 <- rnorm(n,mean=3,sd=sqrt(6))
x_2 <-rnorm(n,mean=2,sd=sqrt(4))
epsilon <- rnorm(n,mean=0,sd=sqrt(100))
y <-2+3.5*x_1-9.2*x_2+epsilon
model4c <- lm(y ~x_1+x_2)
summary(model4c)
```

```

Call:
lm(formula = y ~ x_1 + x_2)

Residuals:
     Min       1Q   Median       3Q      Max 
-29.4359  -4.3645   0.0202   6.3692  26.3941 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.5297     2.1508   1.176    0.242    
x_1           3.5862     0.4766   7.524 2.72e-11 ***
x_2          -9.4673     0.5474 -17.295  < 2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.43 on 97 degrees of freedom
Multiple R-squared:  0.7859,    Adjusted R-squared:  0.7815
F-statistic:   178 on 2 and 97 DF,  p-value: < 2.2e-16
```

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

```r
lm(y~x_1+x_2)$coefficients
```

```
(Intercept)         x_1         x_2
   2.529657    3.586182   -9.467334
```

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

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

estimates_hat <- matrix( , nrow=M, ncol=3)

for (i in 1:M){
  x_1hat <- rnorm(n,mean=3,sd=sqrt(6))
  x_2hat <-rnorm(n,mean=2,sd=sqrt(4))
  epsilonhat <- rnorm(n,mean=0,sd=sqrt(100))
  yhat <-2+3.5*x_1hat-9.2*x_2hat+epsilonhat
  model4c_hat <- lm(yhat ~x_1hat+x_2hat)
  summary(model4c_hat)
  estimates_hat[i, ]<-coef(model4c_hat)
}

summary(model4c_hat)
```

```
Call:
lm(formula = yhat ~ x_1hat + x_2hat)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-23.736  -6.136   1.829   5.846  20.767 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.1526     1.7297  -0.088     0.93    
x_1hat        3.3185     0.3333   9.958   <2e-16 ***
x_2hat       -8.6074     0.4425 -19.452   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.771 on 97 degrees of freedom
Multiple R-squared:  0.8461,    Adjusted R-squared:  0.8429 
F-statistic: 266.6 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
colnames(estimates_hat) <-c("Intercept","Beta1_hat","Beta2_hat" )
lm(yhat~x_1hat+x_2hat)$coefficients
```

```
(Intercept)      x_1hat      x_2hat 
  -0.152614    3.318500   -8.607369 
```

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

```r
hist(estimates_hat[, "Beta1_hat"],
     main="Sampling Distribution from for Beta-1 from 1000 iterations",
     xlab = "Beta-1-hat",
     col= "blue")
```

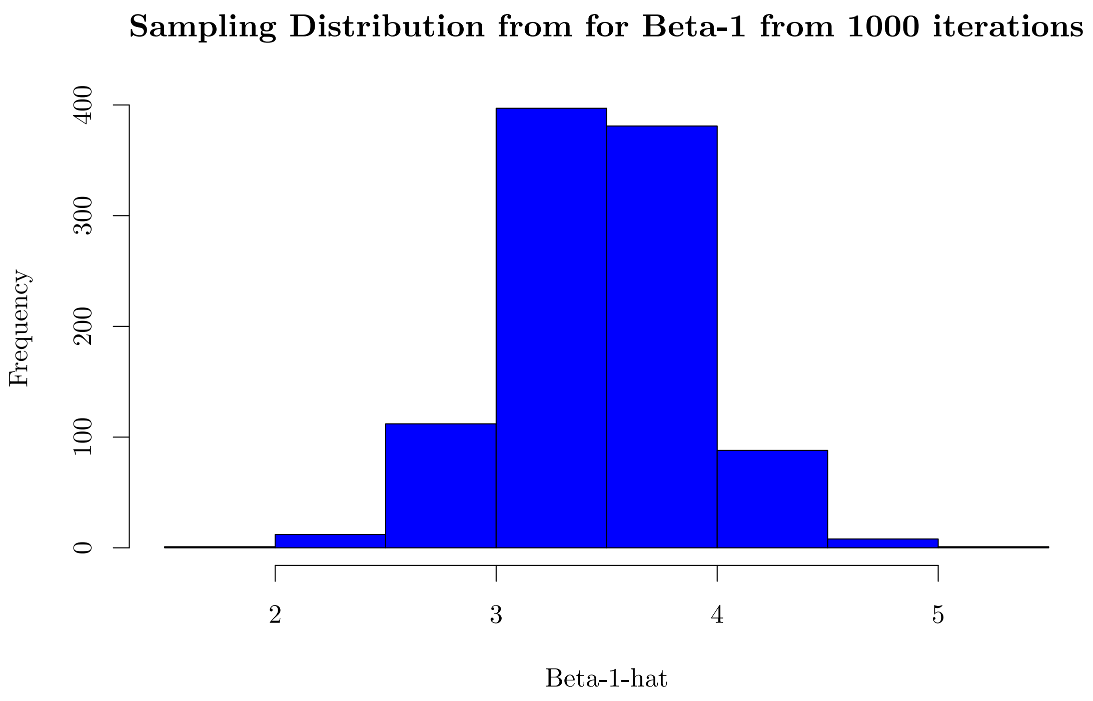

Figure 1: Sampling distribution from 1000 iterations for Beta-1-hat

```r
hist(estimates_hat[, "Beta2_hat"],
     main="Sampling Distribution from for Beta-2-hat from 1000 iterations",
     xlab = "Beta-2-hat",
     col= "green")
```

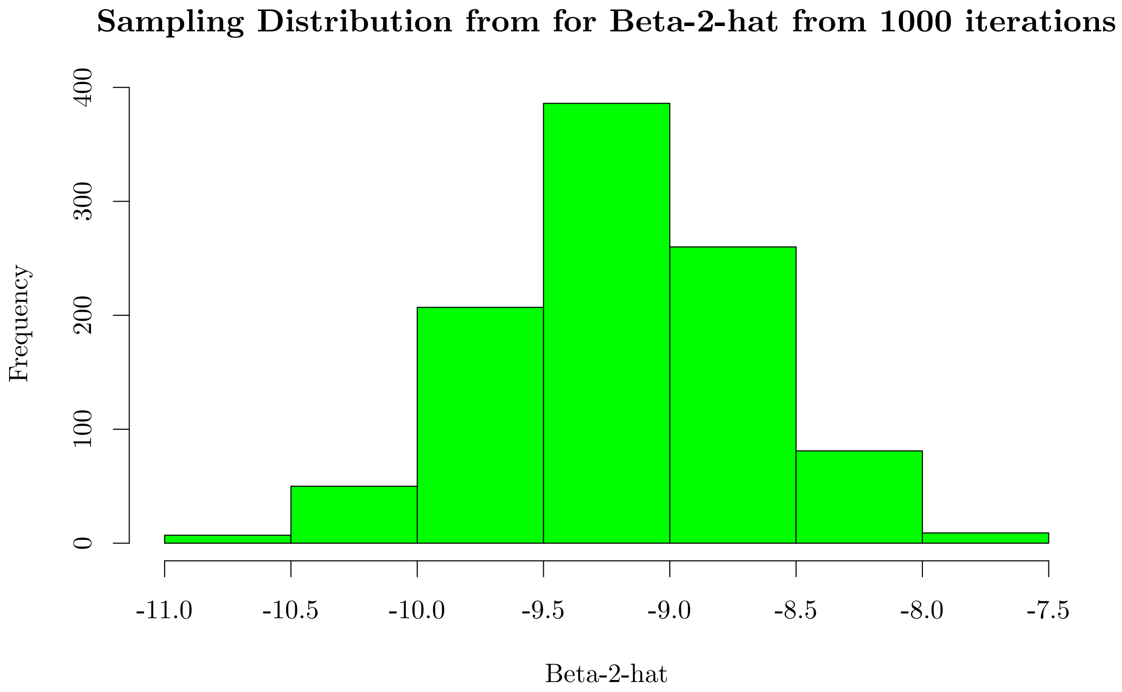

Figure 2: Sampling distribution from 1000 iterations for Beta-2-hat

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

```r
hist(estimates_hat[ , "Beta1_hat"],
     main= "Sampling Distribution from for Beta-1-hat from 1000 iterations",
     xlab = "Beta-1-hat",
     col= "blue")


abline(v= mean(estimates_hat[, "Beta1_hat"]),
       col= "red",
       lwd=2,
       lty="dashed")
```

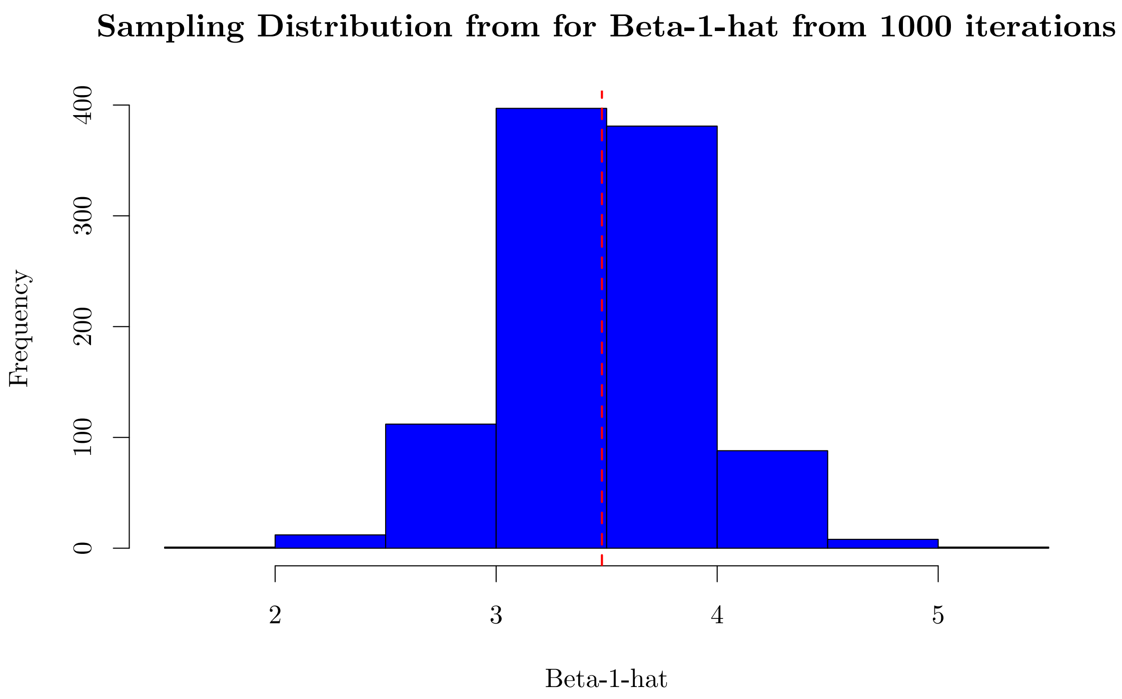

Figure 3: Sampling distribution from 1000 iterations for Beta-1-hat

```
hist(estimates_hat[ , "Beta2_hat"],
     main="Sampling Distribution from for Beta-2-hat from 1000 iterations",
     xlab = "Beta-2-hat",
     col= "green")

abline(v= mean(estimates_hat[, "Beta2_hat"]),
       col= "red",
       lwd=2,
       lty= "dashed")
```

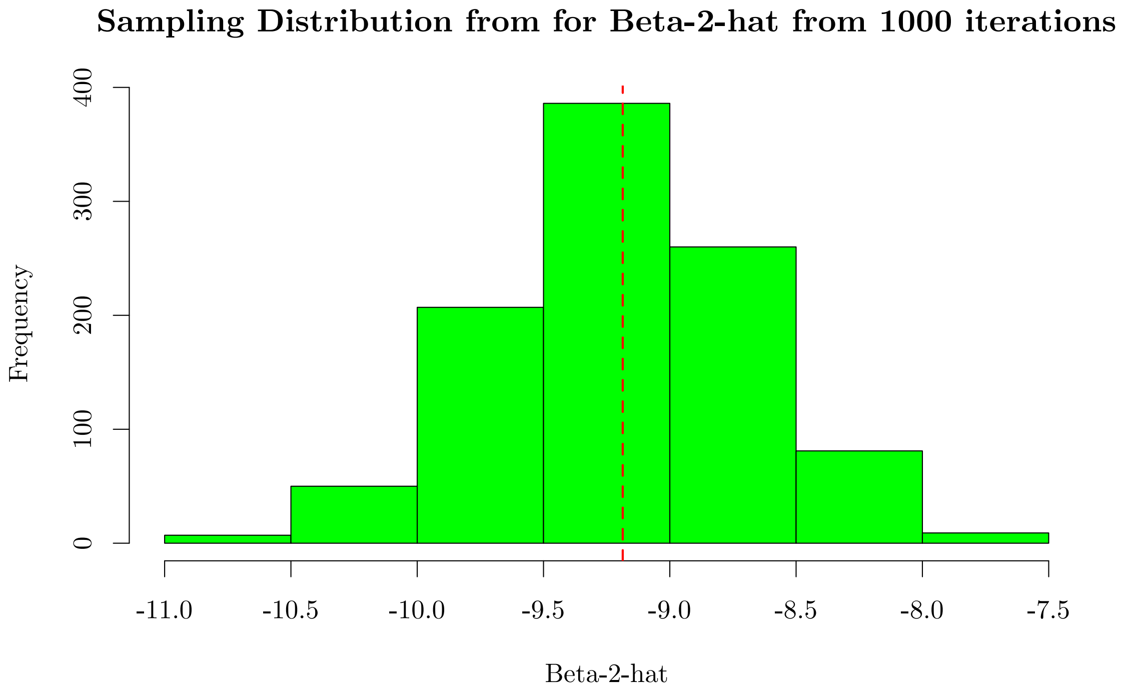

Figure 4: Sampling distribution from 1000 iterations for Beta-2-hat