MTH441 Lab 2 Solutions

08/08/2024

Name: Kaushik Raj V Nadar

Roll No.: 208160499

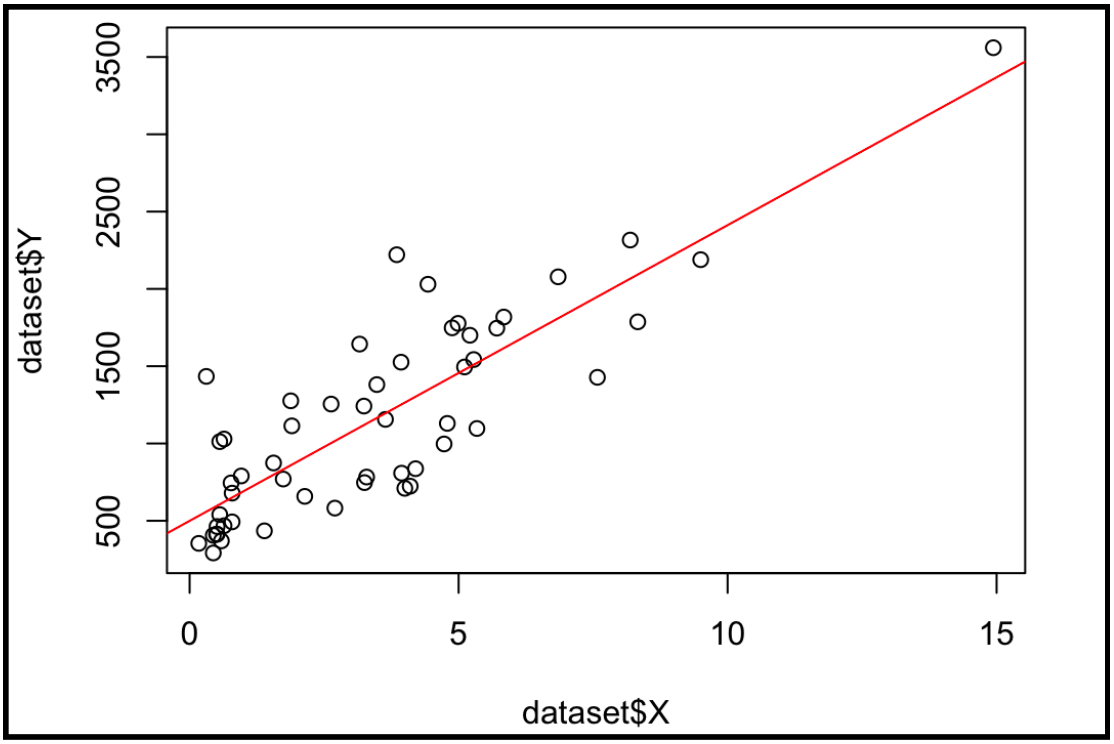
P 1. ***The Rocket Propellant Data:*** Consider the Rocket Propellant Data from Example 2.1 on page 15 of the book Introduction to Linear Regression Analysis by Montgomery et.al. Fit a simple linear regression model for the Rocket Propellant Data.

Code:

```
library(readxl)

dataset <- read_excel("Electricity_Data.xlsx")

model <- lm(Y~X, data=dataset)

summary(model)

plot(dataset$X, dataset$Y)

abline(model, col=‘red')
```

Output:

```
Call:
lm(formula = Y ~ X, data = dataset)

Residuals:
   Min     1Q Median     3Q    Max
-559.9 -285.9  -39.9  249.6  984.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   499.48      77.09   6.479 3.62e-08 ***
X             191.32      17.35  11.030 4.11e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 359.5 on 51 degrees of freedom
Multiple R-squared:  0.7046,    Adjusted R-squared:  0.6988
F-statistic: 121.7 on 1 and 51 DF,  p-value: 4.106e-15
```



P 2. ***The Delivery Time Data:*** Consider the Delivery Time Data from Example 3.1 on page 74 of the book Introduction to Linear Regression Analysis by Montgomery et.al. Fit a multiple linear regression model for the Delivery Time Data.

Code:

```
dataset <- read_excel("TimeDeliveryData.xlsx")

model <- lm(Y~X1+X2, data=dataset)

summary(model)
```

Output:

```
Call:
lm(formula = Y ~ X1 + X2, data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7880 -0.6629  0.4364  1.1566  7.4197 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.341231   1.096730   2.135 0.044170 *  
X1          1.615907   0.170735   9.464 3.25e-09 ***
X2          0.014385   0.003613   3.981 0.000631 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.259 on 22 degrees of freedom
Multiple R-squared:  0.9596,	Adjusted R-squared:  0.9559 
F-statistic: 261.2 on 2 and 22 DF,  p-value: 4.687e-16
```

P 3. For the data in **P 1** compute the following:

(1) Value of the unbiased estimator of σ^2.

(2) $t_j = \widehat{\beta}_j / \sqrt{\mathrm{Var}(\widehat{\beta}_j)}$.

Code:

```
res <- model$residuals

hatsigma <- (t(res)%*%res)/(length(res)-3)

cat("σ2 = ", hatsigma)

ones<-rep(1, length(res))

X<-cbind(ones, dataset$X)
```

```
C<-solve(t(X)%*%X)
est.beta <- model$coefficients
t.beta0 <- est.beta[1]/(sqrt(hatsigma*C[1,1]))
cat("t0 = ", t.beta0)
t.beta1 <- est.beta[2]/(sqrt(hatsigma*C[2,2]))
cat("t1 = ", t.beta1)
```

Output:

```
σ2 =  131814.5
t0 =  6.414929
t1 =  10.92121
```

P 4. Generating sample from a Chi–square random variable:

(1) Generate $n = 5000$ samples using $W = \sum_{i=1}^{3} Z_i^2$, where $Z_i \sim \mathcal{N}(0,1)$ for $i = 1,2,3$.

(2) Plot the histogram of W. Compute the sample mean and variance of W.

(3) Compute the theoretical mean and variance of chi–square random variable with 3 degrees of freedom.

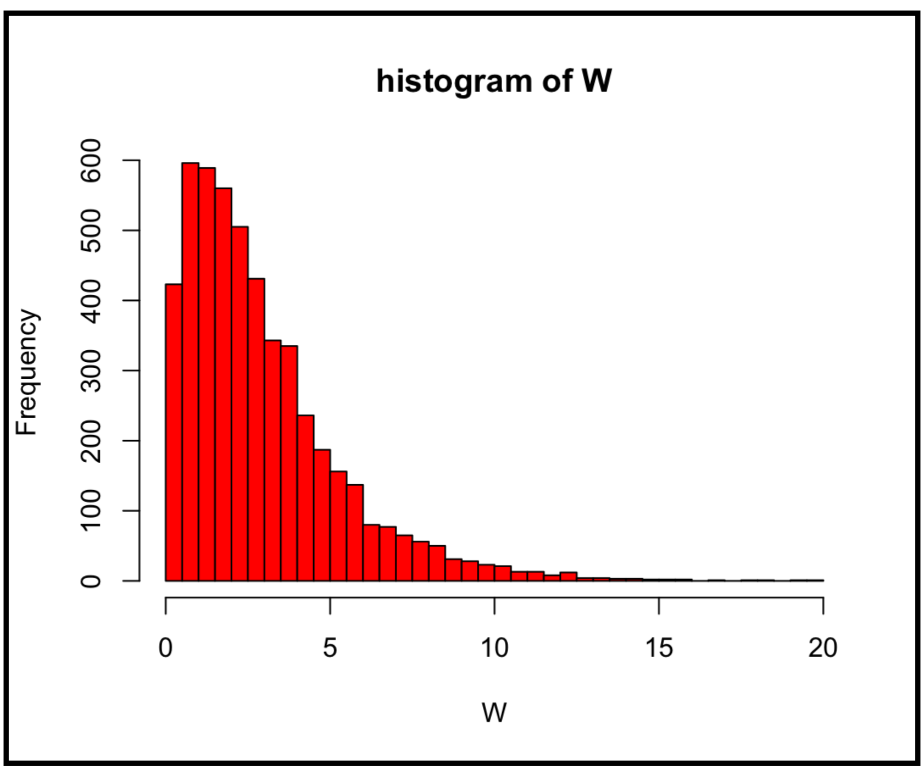
(4) Compare the results of (2) and (3).

Code:

```
n <- 5000
Z1 <- rnorm(n, mean = 0, sd = 1)
Z2 <- rnorm(n, mean = 0, sd = 1)
Z3 <- rnorm(n, mean = 0, sd = 1)
W <- Z1^2 + Z2^2 + Z3^2
head(W)
```

```
hist(W, breaks = 50, main = "histogram of W", xlab = "W", col
= "red")

mean_W <- mean(W)

var_W <- var(W)

cat("Sample mean of W:", mean_W, "\n")

cat("Sample variance of W:", var_W, "\n")


theoretical_mean_W <- 3

theoretical_var_W <- 6

mean_difference <- abs(theoretical_mean_W - mean_W)

variance_difference <- abs(theoretical_var_W - var_W)

cat("Difference between theoretical mean and sample mean:",
mean_difference, "\n")

cat("Difference between theoretical variance and sample
variance:", variance_difference, "\n")
```

Output:



Sample mean of W: 2.934132

Sample variance of W: 5.808767

Difference between theoretical mean and sample mean: 0.06586763

Difference between theoretical variance and sample variance: 0.1912327

P 5. Generating sample from a Chi–square random variable:

(1) Given a full rank 8×5 matrix X (create using *matrix* function) compute $\mathrm{P_X} = \mathrm{X}(\mathrm{X'X})^{-1}\mathrm{X'}$. Verify if $\mathrm{P_X}$ is idempotent.

(2) Generate 5000 samples using $u = Y'\mathrm{P_X}Y$ where $Y \sim \mathcal{N}(\boldsymbol{\mu}, \mathrm{I})$, $\boldsymbol{\mu} = (0,0,0,0,0,0,0,0)'$ and I is a 8×8 diagonal matrix with all ones as diagonal.

(3) Plot the histogram of u. Compute the sample mean and variance of u.

(4) Compute the theoretical mean and variance of chi–square random variable with 5 degrees of freedom.

(5) Compare the results of (3) and (4).

Code:

```
X <- matrix(rnorm(8 * 5), nrow = 8, ncol = 5)
XtX_inv <- solve(t(X) %*% X)
PX <- X %*% XtX_inv %*% t(X)

PX_squared <- PX %*% PX
is_idempotent <- all.equal(PX, PX_squared, tolerance = 1e-10)

cat("Projection matrix P_X:\n")
print(PX)

cat("\nIs P_X idempotent? ", is_idempotent, "\n")
```

```
n_samples <- 5000
mu <- rep(0, 8)
I <- diag(8)
Y <- mvrnorm(n_samples, mu, I)

u <- numeric(n_samples)
for (i in 1:n_samples) {
  u[i] <- t(Y[i, ]) %*% PX %*% Y[i, ]
}

cat("First few samples of u:\n")
print(head(u))

hist(u, breaks = 50, main = "Histogram of u", xlab = "u", col
= "green")

sample_mean_u <- mean(u)
sample_var_u <- var(u)
cat("Sample mean of u:", sample_mean_u, "\n")
cat("Sample variance of u:", sample_var_u, "\n")

theoretical_mean_u <- 5
theoretical_var_u <- 10
cat("Theoretical mean of u:", theoretical_mean_u, "\n")
cat("Theoretical variance of u:", theoretical_var_u, "\n")
```

```
mean_difference <- abs(theoretical_mean_u - sample_mean_u)

variance_difference <- abs(theoretical_var_u - sample_var_u)

cat("Difference between theoretical mean and sample mean:",
mean_difference, "\n")

cat("Difference between theoretical variance and sample
variance:", variance_difference, "\n")
```

Output:

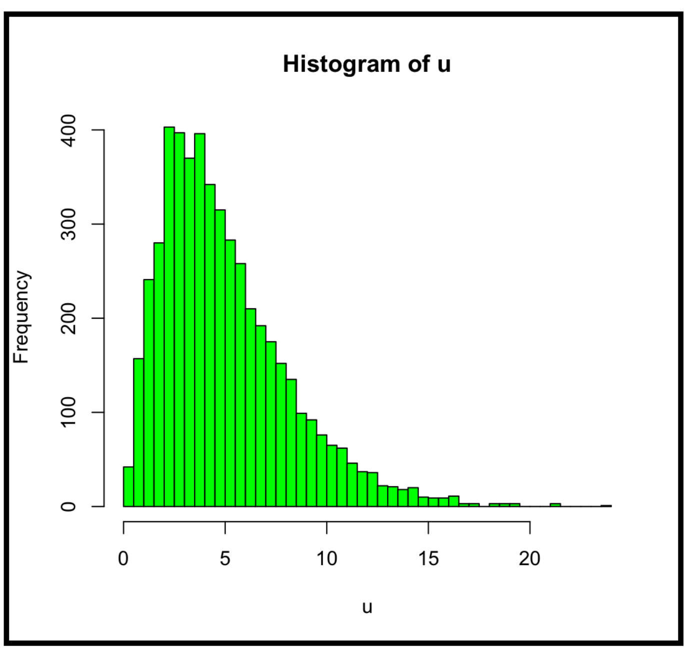
Projection matrix P_X:

```
            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]        [,7]        [,8]
[1,]  0.74880530 -0.27480166 -0.25327509  0.12627321 -0.02215265  0.10821517  0.09648372  0.10476833
[2,] -0.27480166  0.29595123  0.12528763  0.24133148  0.03819803  0.03751440  0.21494368 -0.09921223
[3,] -0.25327509  0.12528763  0.27612254  0.04416468  0.04302235  0.28382015 -0.11040036  0.15326469
[4,]  0.12627321  0.24133148  0.04416468  0.90431340 -0.04237888 -0.06138018 -0.04748382  0.05075567
[5,] -0.02215265  0.03819803  0.04302235 -0.04237888  0.74596716 -0.15658060  0.17881797  0.35694550
[6,]  0.10821517  0.03751440  0.28382015 -0.06138018 -0.15658060  0.80552834  0.11834129  0.14384379
[7,]  0.09648372  0.21494368 -0.11040036 -0.04748382  0.17881797  0.11834129  0.78865818 -0.22526226
[8,]  0.10476833 -0.09921223  0.15326469  0.05075567  0.35694550  0.14384379 -0.22526226  0.43465385
```

Is P_X idempotent? TRUE

First few samples of u:

```
[1] 10.749035  2.522771  4.599951  3.190109  5.954706  4.548479
```



Sample mean of u: 4.984684

Sample variance of u: 9.901843

Theoretical mean of u: 5

Theoretical variance of u: 10

Difference between theoretical mean and sample mean: 0.01531644

Difference between theoretical variance and sample variance: 0.09815717

P 6. Generating sample from a F–distribution:

(2) Generate 5000 samples using $f = (Y'\mathrm{P}_{\mathrm{X}_2}Y/df_1)/(Y'\mathrm{P}_{\mathrm{X}_1}Y/df_2)$ where $Y \sim \mathcal{N}(\boldsymbol{\mu}, \mathrm{I}_8)$, $\boldsymbol{\mu} = (0,0,0,0,0,0,0,0)'$ and I_8 is a 8×8 diagonal matrix with all ones in diagonal, $\mathrm{P}_{\mathrm{X}_2} = \mathrm{I}_8 - \mathrm{P}_{\mathrm{X}_1}$, $df_1 = Rank(\mathrm{P}_{\mathrm{X}_2})$, $df_2 = Rank(\mathrm{P}_{\mathrm{X}_1})$ and X_1 is same as X from **P 5.** (1).

(3) Plot the histogram of f. Compute the sample mean and variance of f.

(4) Compute the theoretical mean and variance of F–random variable with (3,5) degrees of freedom.

(5) Compare the results of (3) and (4).

Code:

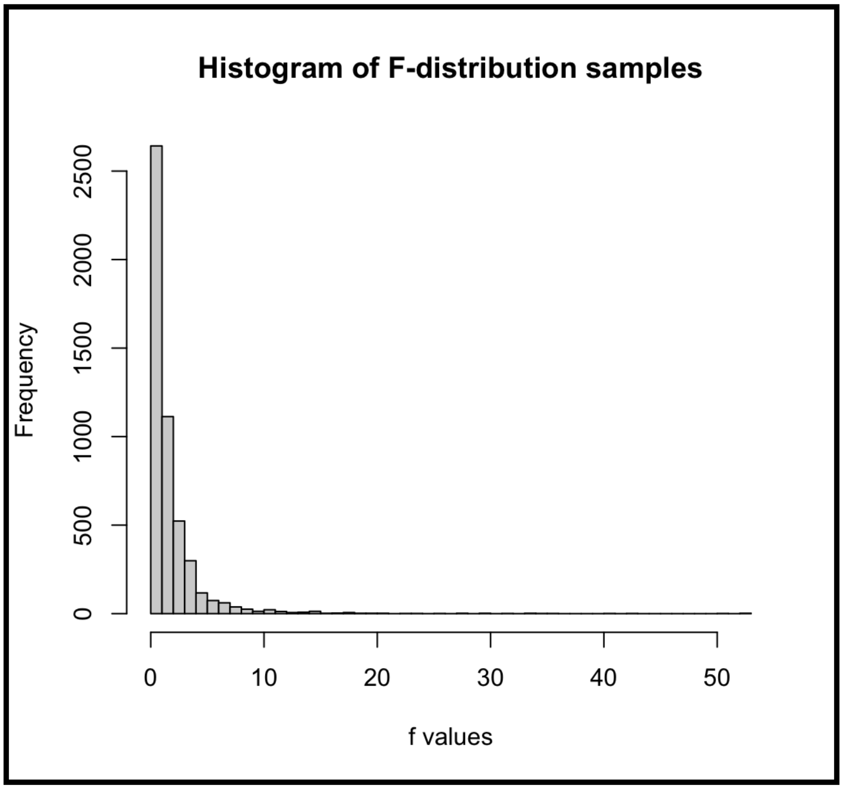
```
# Define parameters
n <- 5000
mu <- rep(0, 8)
I8 <- diag(8)
df1 <- 3
df2 <- 5

# Generate Y from N(mu, I8)
Y <- mvrnorm(n, mu, I8)
```

```
# Define X1
X1 <- matrix(rnorm(8 * df2), nrow = 8, ncol = df2)

# Compute PX1 and PX2
PX1 <- X1 %*% solve(t(X1) %*% X1) %*% t(X1)
PX2 <- I8 - PX1

# Compute f values
f_values <- apply(Y, 1, function(y) {
  (t(y) %*% PX2 %*% y / df1) / (t(y) %*% PX1 %*% y / df2)
})

# Plot the histogram of f
hist(f_values, breaks = 50, main = "Histogram of F-
distribution samples", xlab = "f values")

# Compute sample mean and variance
sample_mean <- mean(f_values)
sample_variance <- var(f_values)

# Compute theoretical mean and variance
theoretical_mean <- df2 / (df2 - 2)
theoretical_variance <- (2 * df2^2 * (df1 + df2 - 2)) / (df1
* (df2 - 2)^2 * (df2 - 4))

# Print results
```

```
cat("Sample Mean:", sample_mean, "\n")

cat("Sample Variance:", sample_variance, "\n")

cat("Theoretical Mean:", theoretical_mean, "\n")

cat("Theoretical Variance:", theoretical_variance, “\n")


mean_difference <- abs(theoretical_mean - sample_mean)

variance_difference <- abs(theoretical_variance -
sample_variance)


cat("Difference between theoretical mean and sample mean:",
mean_difference, "\n")

cat("Difference between theoretical variance and sample
variance:", variance_difference, "\n")
```

Output:

```
Sample Mean: 1.737454

Sample Variance: 8.720356

Theoretical Mean: 1.666667

Theoretical Variance: 11.11111

Difference between theoretical mean and sample mean: 0.07078695

Difference between theoretical variance and sample variance: 2.390755
```


Histogram of F-distribution samples
Frequency
0
500
1000
1500
2000
2500
0
10
20
30
40
50
f values