Lab 3 Solutions

Name: Kaushik Raj V Nadar

Roll No.: 208160499

Date: 13/08/2024

P 1. Consider the data $y = (1, 4, 8, 9, 3, 8, 9)$, $x_1 = (-1, 1, -1, 1, 0, 0, 0)$, $x_2 = (-1, -1, 1, 1, 0, 1, 2)$. Let the following regression model is used:

$$\mathbb{E}(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2$$

Test the hypothesis $H_0 : \beta_0 = 0,\ \beta_1 - \beta_2 = 0$. You may use *'linearHypothesis'* function from **carData** package to verify your answer.

Part 1

Code:

```
library(Matrix)
y = c(1, 4, 8, 9, 3, 8, 9)
x1 = c(-1, 1, -1, 1, 0, 0, 0)
x2 = c(-1, -1, 1, 1, 0, 1, 2)

A = matrix(c(1,0,0,1,0,-1,0,0),nrow=2)
q = rankMatrix(A)[1]
C = rep(0,2)

model = summary(lm(y~x1+x2+I(x1^2)))

n = length(y)
p = 4

X = cbind(1,x1,x2,x1^2)

beta = model$coefficients[,1]
beta
rss_rssh =
t(A%*%beta)%*%solve(A%*%solve(t(X)%*%(X))%*%t(A))%*%A%*%beta
rss_rssh

RSS = t(y-X%*%beta)%*%(y-X%*%beta)

f.stat= ((rss_rssh)/q)/(RSS/(n-p))
f.stat
```

Output:

```
> beta
(Intercept)          x1          x2     I(x1^2)
   3.666667    1.000000    3.000000    1.833333
> rss_rssh = t(A%*%beta)%*%solve(A%*%solve(t(X)%*%(X))%*%t(A))%*%A%*%beta
> rss_rssh
         [,1]
[1,] 55.64103
>
> RSS = t(y-X%*%beta)%*%(y-X%*%beta)
>
> f.stat= ((rss_rssh)/q)/(RSS/(n-p))
> f.stat
         [,1]
[1,] 22.76224
```

Part 2

Code:

```
# Load necessary packages
library(car)

# Step 1: Prepare the data
y = c(1, 4, 8, 9, 3, 8, 9)
x1 = c(-1, 1, -1, 1, 0, 0, 0)
x2 = c(-1, -1, 1, 1, 0, 1, 2)

A = matrix(c(1,0,0,1,0,-1,0,0),nrow=2)

# Step 2: Fit the regression model
model <- lm(y ~ x1 + x2 + I(x1^2))

# Step 3: Test the hypotheses
# Test H0: beta0 = 0 and beta1 - beta2 = 0
linearHypothesis(model, hypothesis.matrix = A)
```

Output:

```
Linear hypothesis test

Hypothesis:
(Intercept) = 0
x1 - x2 = 0

Model 1: restricted model
Model 2: y ~ x1 + x2 + I(x1^2)

  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1      5 59.308
2      3  3.667  2    55.641 22.762 0.01537 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

P 2. Consider the data set and the model from **P 1.** Consider the following null hypothesis $H_0: \beta_1 = \beta_2 = \beta_3 = 0$.

(a) Compute the restricted LSE under the null hypothesis.

(b) Compute the restricted RSS and unrestricted RSS.

(c) Compute the F–statistics for testing H_0.

(d) Verify your results from the output of lm function.

Code:

```
# Data
y <- c(1, 4, 8, 9, 3, 8, 9)
x1 <- c(-1, 1, -1, 1, 0, 0, 0)
x2 <- c(-1, -1, 1, 1, 0, 1, 2)

# (a) Restricted LSE under H0
Beta0_hat_restricted <- mean(y)
cat("Restricted LSE for β0 under H0:", Beta0_hat_restricted,
"\n")

# (b) Restricted RSS
RSS_restricted <- sum((y - Beta0_hat_restricted)^2)
cat("Restricted RSS:", RSS_restricted, "\n")

# Unrestricted model
X <- cbind(1, x1, x2, x1^2)
Beta_hat_unrestricted <- solve(t(X) %*% X) %*% t(X) %*% y

# Unrestricted RSS
RSS_unrestricted <- sum((y - X %*% Beta_hat_unrestricted)^2)
cat("Unrestricted RSS:", RSS_unrestricted, "\n")
```

```r
# (c) Compute the F-statistic
n <- length(y)
p <- ncol(X)
q <- 3  # Number of restrictions

F_statistic <- ((RSS_restricted - RSS_unrestricted) / q) /
(RSS_unrestricted / (n - p))
cat("F-statistic for testing H0:", F_statistic, "\n")

# (d) Verify using lm function
unrestricted_model <- lm(y ~ x1 + x2 + I(x1^2))
summary(unrestricted_model)

# To test the null hypothesis using the anova function
restricted_model <- lm(y ~ 1)
anova(restricted_model, unrestricted_model)
```

Output:

```
> # (a) Restricted LSE under H0
> Beta0_hat_restricted <- mean(y)
> cat("Restricted LSE for β0 under H0:", Beta0_hat_restricted,
"\n")
Restricted LSE for β0 under H0: 6
>
> # (b) Restricted RSS
> RSS_restricted <- sum((y - Beta0_hat_restricted)^2)
> cat("Restricted RSS:", RSS_restricted, "\n")
Restricted RSS: 64
>
> # Unrestricted model
> X <- cbind(1, x1, x2, x1^2)
> Beta_hat_unrestricted <- solve(t(X) %*% X) %*% t(X) %*% y
>
> # Unrestricted RSS
> RSS_unrestricted <- sum((y - X %*% Beta_hat_unrestricted)^2)
> cat("Unrestricted RSS:", RSS_unrestricted, "\n")
Unrestricted RSS: 3.666667
>
> # (c) Compute the F-statistic
> n <- length(y)
> p <- ncol(X)
> q <- 3  # Number of restrictions
>
> F_statistic <- ((RSS_restricted - RSS_unrestricted) / q) /
(RSS_unrestricted / (n - p))
```

```
> cat("F-statistic for testing H0:", F_statistic, "\n")
F-statistic for testing H0: 16.45455
>
> # (d) Verify using lm function
> unrestricted_model <- lm(y ~ x1 + x2 + I(x1^2))
> summary(unrestricted_model)

Call:
lm(formula = y ~ x1 + x2 + I(x1^2))

Residuals:
      1       2       3       4       5       6       7
-0.5000  0.5000  0.5000 -0.5000 -0.6667  1.3333 -0.6667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.6667     0.7817   4.690  0.01832 *
x1            1.0000     0.5528   1.809  0.16815
x2            3.0000     0.4513   6.647  0.00694 **
I(x1^2)       1.8333     0.9574   1.915  0.15140
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.106 on 3 degrees of freedom
Multiple R-squared:  0.9427,  Adjusted R-squared:  0.8854
F-statistic: 16.45 on 3 and 3 DF,  p-value: 0.02288

>
> # To test the null hypothesis using the anova function
> restricted_model <- lm(y ~ 1)
> anova(restricted_model, unrestricted_model)
Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x1 + x2 + I(x1^2)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1      6 64.000
2      3  3.667  3    60.333 16.454 0.02288 *
```

P 3. Consider the following data

Table 1: Lack of fit data

x	1.0	1.0	2.0	3.3	3.3	4.0	4.0	4.0	4.7	5.0	5.6
y	10.84	9.30	16.35	22.88	24.35	24.56	25.86	29.16	24. 59	22.25	25.90
x	5.6	5.6	6.0	6.0	6.5	6.9					
y	27.20	25.61	25.45	26.56	21.03	21.46					

Perform a lack–of–test for the data given in the Table above. You may use *'ols_pure_error_anova'* function from the package **olsrr**.

Code:

```
library(olsrr)

# Input the data
x <- c(1.0, 1.0, 2.0, 3.3, 3.3, 4.0, 4.0, 4.0, 5.6, 5.6, 6.0,
6.0, 6.5, 6.9)
y <- c(10.84, 9.30, 16.35, 22.88, 24.35, 24.56, 25.86, 29.16,
27.20, 25.61, 25.45, 26.56, 21.03, 21.46)

# Fit the linear model
model <- lm(y ~ x)

# Perform the lack-of-fit test
lack_of_fit_test <- ols_pure_error_anova(model)

# Print the results
print(lack_of_fit_test)
```

Output:

```
Lack of Fit F Test
--------------
Response :   y
Predictor:   x

                      Analysis of Variance Table
-----------------------------------------------------------------------
                DF     Sum Sq      Mean Sq     F Value       Pr(>F)
-----------------------------------------------------------------------
x                1    225.2168    225.2168    87.78663    7.204966e-07
Residual        12    245.0711    20.42259
 Lack of fit     6    229.6781    38.27968    14.92093     0.002250367
 Pure Error      6    15.39302    2.565503
-----------------------------------------------------------------------
```