HOMEWORK 4

Yiqiao Yin [YY2502]

Contents

PROBLEM 1

Consider n-dimensional matrix X with mean 0 and covariance Σ. Let A be a $k \times n$ matrix.

Proof

$$\begin{aligned} \text{cov}(Y) &= \text{cov}(AX) \\ &= A\text{cov}(X)A^T \\ &= A\Sigma A^T \end{aligned}$$

Q.E.D.

PROBLEM 2

Consider linear model $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and hypothesis $H_0 : \beta_1 = 0$ and $H_1 : \beta_1 \neq 0$.

Proof

Let us consider linear regression model $Y \sim X_1 + \dots X_p$ while p is the number of covariates in the data. Let us also dentoe n to be total sample size (number of rows) in the data. Let us also denote conventional terminologies: - $\text{SS}_{XX} = \sum_{i=1}^n (X_i - \bar{X})^2$ - $\text{SS}_{YY} = \sum_{i=1}^n (Y_i - \bar{Y})^2$ - $\text{SS}_{XY} = \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$

In terms of the framework of analysis of variance, let us clarify the following terminologies: - $\text{SSE} = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$ - $\text{SSR} = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$ - $\text{SST} = \sum_{i=1}^n (Y_i - \bar{Y})^2$

Now let us start by realizing, in *t-test*, we have

$$t - \text{stat} = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)}$$

while $\beta_1 = 0$ is the statement of null hypothesis. Under null hypothesis, *t-statistics* follows t-distribution. In other words, we have

$$t - \text{stat} = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)} \sim t(1 - \frac{\alpha}{2}, n - 1)$$

We can carry out the following derivation:

$$\begin{array}{lll} t^2 & = & \left[\frac{\hat{\beta}_1}{\text{SE}(\hat{\beta})_1}\right]^2 \\ & = & \hat{\beta}_1 \hat{\beta}_1 \left[\frac{\text{SS}_{XX}}{\text{MSE}}\right] \\ & = & \hat{\beta}_1 \left[\frac{\text{SS}_{XY}}{\text{SS}_{XX}}\right] \left[\frac{\text{SS}_{XX}}{\text{MSE}}\right] \\ & = & \hat{\beta}_1 \frac{\text{SS}_{XY}}{\text{MSE}} \\ & = & \frac{\text{SSR}}{\text{SS}_{XY}} \frac{\text{SS}_{XY}}{\text{MSE}} \\ & = & \frac{\text{SSR}}{\text{MSE}} = \frac{\text{SSR}/1}{\text{MSE}} \\ & = & F \end{array}$$

This is because the null hypothesis states $\beta_1 = 0$ which essentially gives degree of freedom of 1. This means $\text{SSR} = \text{SSR}/1$ can be used as the definition of MSE for reduced model. This formation satisfies exactly that of the F-distribution.

Q.E.D.

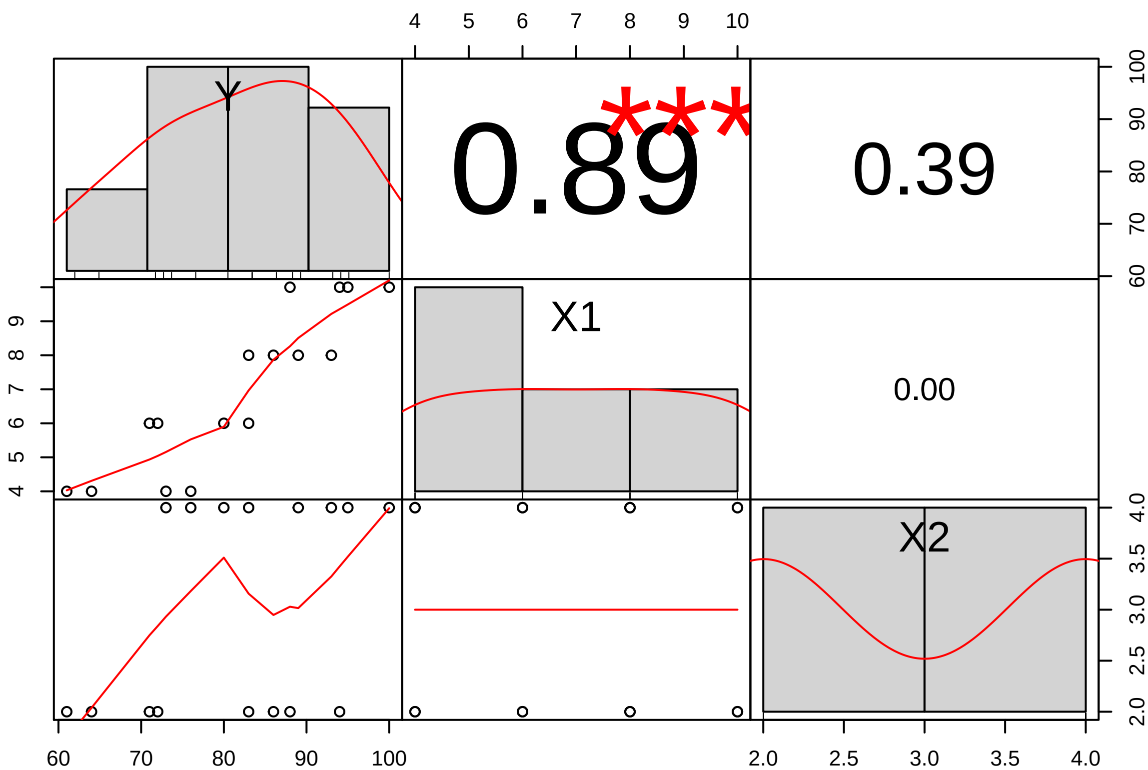
PROBLEM 3 (Ch6, Q5, a.b.c)

Let us load the data first.

```
setwd("C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6101 - Applied Statistics I/Data")
data = read.csv("CH06PR05.csv", header = FALSE)
colnames(data) <- c("Y", "X1", "X2")
```

(a) Scatter Plot and Correlation Plot

```r
PerformanceAnalytics::chart.Correlation(data)
```



From above results, we have the correlation matrix to be

$$\rho = \begin{bmatrix} 1 & 0.89 & 0.39 \\ 0.89 & 1 & 0 \\ 0.39 & 0 & 1 \end{bmatrix}$$

From the scatter plot, we observe that there is a positive association between Y and X_1. Moreover, this association can be confirmed by looking at the correlation matrix plot which states $\text{cor}(Y, X_1) = 0.89$, a positive association. The association between Y and X_2, however, is not as strong as that between Y and X_1. This pattern from scatter plot can be confirmed by $\text{cor}(Y, X_2) = 0.39$ in correlation matrix plot.

(b) Regression Model

Let us build regression model using *lm()* function.

```r
LM = lm(Y~., data = data); summary(LM)
```

```
##
## Call:
## lm(formula = Y ~ ., data = data)
##
## Residuals:
```

```
##     Min      1Q Median      3Q     Max
## -4.400 -1.762  0.025  1.587  4.200
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.6500     2.9961  12.566 1.20e-08 ***
## X1            4.4250     0.3011  14.695 1.78e-09 ***
## X2            4.3750     0.6733   6.498 2.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.693 on 13 degrees of freedom
## Multiple R-squared:  0.9521, Adjusted R-squared:  0.9447
## F-statistic: 129.1 on 2 and 13 DF,  p-value: 2.658e-09
```

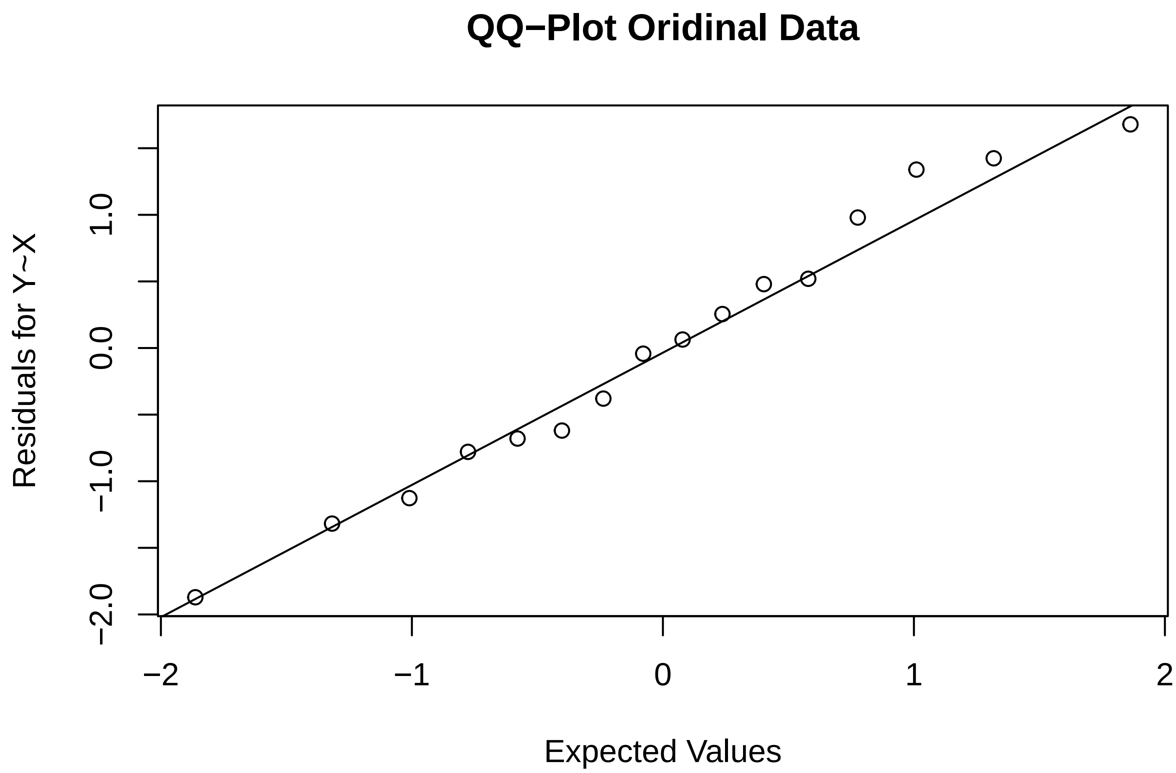
From regression results above, we conclude the following linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 = 37.65 + 4.425 X_1 + 4.375 X_2$$

and we can interpret β_1 as the following. A unit change of X_1 has a positive impact on Y and it will increase Y by a value of $\beta_1 = 4.425$ marginally. Here maginally refers to the changes made that is with respect to X_1 alone.

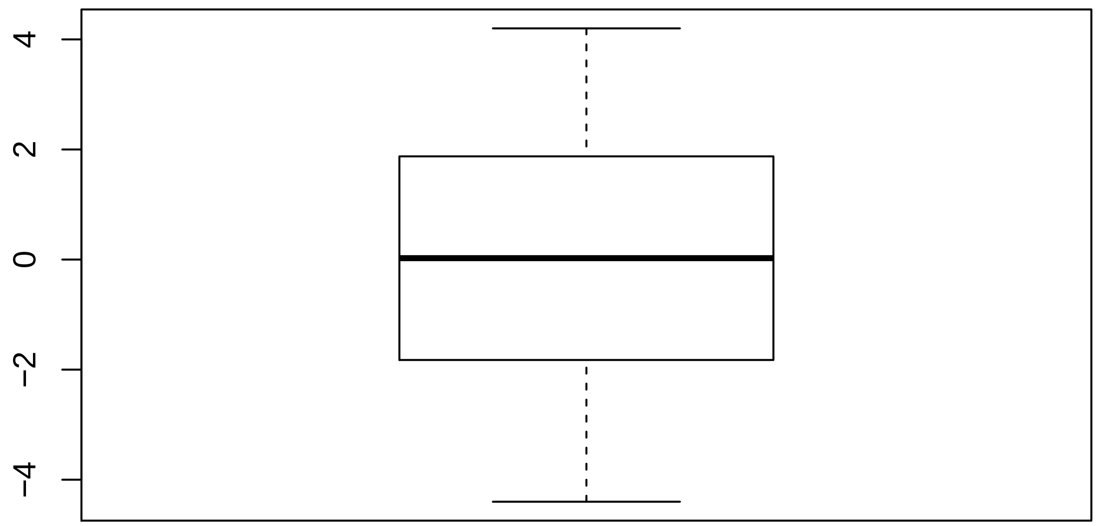
(3) Residuals

```
qqnorm(
  rstandard(LM),
  main = paste0("QQ-Plot Oridinal Data"),
  xlab = "Expected Values",
  ylab = "Residuals for Y~X"); qqline(rstandard(LM))
```



We observe from the QQ-plot of residuals that most of the residuals fall on the straight line crossing (0,0). We can say that the residuals look very much like normal distribution.

```
boxplot(LM$residuals)
```



From results of Box-plot, we observe that the residuals in the middle, from 25th percentile to 75th percentile stay within ± 2 ranges. We also have observed that the max and min are around ± 4 which is almost twice as the body of the Box-plot. This relates to the fact that under standard normal distribution the range of two standard deviations is about $1.96 \approx 2$ that of one standard deviation.

PROBLEM 4 (Ch6, Q7)

(a) Coefficient of determinant, i.e. R^2. Let us present multiple approaches of obtaining this results.

```
preds <- LM$fitted.values
actual <- data$Y
rss <- sum((preds - actual) ^ 2)  ## residual sum of squares
tss <- sum((actual - mean(actual)) ^ 2)  ## total sum of squares
rsq <- 1 - rss/tss
print(c("R-square is ", rsq))
```

```
## [1] "R-square is "      "0.952058973055414"
```

We can get to the same results by using *LM* results:

```
1 - sum(LM$residuals^2) / tss
```

```
## [1] 0.952059
```

Alternatively, we can simply extract from *LM* results:

```
summary(LM)$r.sq
```

```
## [1] 0.952059
```

(b) Simple determination between Y and $\hat{Y}$ is

```
Rsimple = cor(data$Y, LM$fitted.values)^2; Rsimple
```

```
## [1] 0.952059
```

We observe that this calculation results in a value that is higher than the multiple coefficient of determination.

PROBLEM 5 (Ch6, Q8)

Assume in the previous example we have residuals to be independent standard normal. In other words, let us assume $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

(a) Compute confidence interval at $X = 5$ anad $X = 4$ respectively. We can use *predict()* function in *R*.

```
confidence <- predict(
  LM,
  newdata = data.frame('X1' = 5, 'X2' = 4),
  interval = 'confidence', level = 0.99)
print(
  paste0(
    "Confidence interval is: [",
    round(confidence[2],3), ", ",
    round(confidence[,3],3), "]"))
```

```
## [1] "Confidence interval is: [73.881, 80.669]"
```

It gives us confidence interval of $[73.881, 80.669]$. We can also compute this from scratch:

```
n = nrow(data)
yhat = 37.65 + 4.425*5 + 4.375*4
print(paste0("Yhat is ", yhat))
```

```
## [1] "Yhat is 77.275"
```

```
X = as.matrix(cbind(1L, data[, -1]))
mse = mean((LM$residuals)^2)
XTX_inv = matlib::inv(t(X) %*% X)
Xh = matrix(c(1, 5, 4), nrow=3)
SE_Y_2 = mse*(1/(n-3) + t(Xh) %*% XTX_inv %*% Xh) # formula from page 245
print(paste0("SE is ", sqrt(SE_Y_2)))
```

```
## [1] "SE is 1.21851205764054"
```

```r
critical = qt(1-0.01/2, n-3)
print(paste0("Critical value is ", critical))
```

```
## [1] "Critical value is 3.01227583871658"
```

```r
lowerB = yhat - critical * sqrt(SE_Y_2)
upperB = yhat + critical * sqrt(SE_Y_2)
print(paste0("Confidence interval is: [", round(lowerB, 3), ", ", round(upperB, 3), "]"))
```

```
## [1] "Confidence interval is: [73.605, 80.945]"
```

Thus, with 99% confidence coefficient, we estiamte that the mean Y given $X_1 = 5$ and $X_2 = 4$ is $[74.216, 80.334]$.

(b) Using formula provided in page 56 of textbook, we may compute prediction interval

```r
Y <- data$Y
X <- cbind(One=1L, data[, c(2,3)])
b <- c(37.65, 4.425, 4.375)
MSE <- (t(Y)  %*% Y - t(b)  %*% t(X)  %*% Y)/(16-3)
s <- (MSE + 1.127^2)^(1/2)
print(paste0("SE is ", s))
```

```
## [1] "SE is 2.91958475709242"
```

```r
s_pred_2 <- s
lowerBB = yhat - critical * s_pred_2
upperBB = yhat + critical * s_pred_2
print(paste0("Confidence interval is: [", round(lowerBB, 3), ", ", round(upperBB, 3), "]"))
```

```
## [1] "Confidence interval is: [68.48, 86.07]"
```

With 99% confidence, we predict that the Y will be within the range of $[68.48, 86.07]$. The same results can also be confirmed using *predict()* function in *R*.

```r
prediction <- predict(
  LM,
  newdata = data.frame('X1' = 5, 'X2' = 4),
  interval = 'prediction',  level = 0.99)
print(
  paste0(
    "Confidence interval is: [",
    round(prediction[2], 2), ", ",
    round(prediction[3], 2), "]"))
```

```
## [1] "Confidence interval is: [68.48, 86.07]"
```

PROBLEM 6 (Ch6, Q25)

We can approach this problem in the following ways. First, if we were to force $\beta_2 = 4$, that would mean that we may treat $\beta_2 \cdot X_2 = 4 \cdot X_2$ as a constant. In other words, we would like to build a model that takes the following form:

$$Y \sim \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \; \beta_0 + \beta_2 X_2 + \beta_1 X_1 + \beta_3 X_3 + \epsilon$$

$$\Rightarrow Y \sim \tilde{\beta}_0 + \beta_1 X_1 + \beta_3 X_3 + \epsilon$$

while $\tilde{\beta}_0 = \beta_0 + \beta_2 X_2$.

In practice, we can approach with the following. We build an artificial model of which we know the parameters. Then we run two linear models. The first one *LM1* will run a linear regression model as we used to know. The second one *LM2* will regress $\tilde{Y} := Y - 4X_2$ on $\beta_1 X_1 + \beta_3 X_3$. The second model is what we are interested in.

```r
# Creat data
n = 1e3
set.seed(2020)
X1 = rnorm(n, 0, 1)
X2 = rnorm(n, 0, 1)
X3 = rnorm(n, 0, 1)
beta1 = 3
beta2 = 4
beta3 = 5

# Model
Y = beta1 * X1 + beta2 * X2 + beta3 * X3 + rnorm(n, 0, 1)

# LM1
LM1 = lm(Y~X1+X2+X3); summary(LM1)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9053 -0.6629 -0.0450  0.7065  2.7347
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01934    0.03270  -0.592    0.554
## X1           3.03333    0.03153  96.193   <2e-16 ***
## X2           4.04372    0.03249 124.459   <2e-16 ***
## X3           4.94284    0.03240 152.538   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.033 on 996 degrees of freedom
## Multiple R-squared:  0.9806, Adjusted R-squared:  0.9805
## F-statistic: 1.676e+04 on 3 and 996 DF,  p-value: < 2.2e-16
```

```r
# LM2
LM2 = lm((Y-4*X2)~X1+X3); summary(LM2)
```

```
##
## Call:
## lm(formula = (Y - 4 * X2) ~ X1 + X3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8743 -0.6635 -0.0451  0.7257  2.7790
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01920    0.03271  -0.587    0.557
## X1           3.03476    0.03153  96.255   <2e-16 ***
## X3           4.94343    0.03241 152.508   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.033 on 997 degrees of freedom
## Multiple R-squared:  0.971,  Adjusted R-squared:  0.971
## F-statistic: 1.67e+04 on 2 and 997 DF,  p-value: < 2.2e-16
```

From the first linear model, *LM1*, we have

$$Y = -0.019 + 3.033X_1 + 4.044X_2 + 4.943X_3$$

while the second linear model, *LM2*, we have

$$Y = -0.019 + 3.035X_1 + 4X2 + 4.934X_3$$

Let us compare the residuals of two models

```r
MSE1 = mean(LM1$residuals^2)
MSE2 = mean(LM2$residuals^2)
print(paste0("MSE for the first model: ", round(MSE1, 3)))
```

```
## [1] "MSE for the first model: 1.062"
```

```r
print(paste0("MSE for the first model: ", round(MSE2, 3)))
```

```
## [1] "MSE for the first model: 1.064"
```

As we can observe, this approach to create the second linear model *LM2* produces results quite similar to the first model *LM1*.