HOMEWORK 5

Yiqiao Yin [YY2502]

Contents

PROBLEM 1 (Ch3, Q14)

Let us refer to the data from Problem 1.22.

```
setwd("C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6101 - Applied Statistics I/Data")
data = read.csv("CH01PR22.csv", header = FALSE)
colnames(data) = c("Hardness", "ElapseTime")

print("Presents data: ")
```

```
## [1] "Presents data: "
```

```
head(data)
```

```
##   Hardness ElapseTime
## 1      199         16
## 2      205         16
## 3      196         16
## 4      200         16
## 5      218         24
## 6      220         24
```

(1) Let us state the hypothesis and perform F-test.

Hypothesis:

$$H_0 : \mathbb{E}(Y) = \beta_0 + \beta_1 X \text{ vs. } H_1 : \mathbb{E}(Y) \neq \beta_0 + \beta_1 X$$

Thus, H_0 postulates that μ_j in the full model, $Y_{ij} = \mu_j + \epsilon_{ij}$, is linear related to X_j:

$$\mu_j = \beta_0 + \beta_1 X_j$$

The reduced model under H_0 is:

$$Y_{ij} = \beta_0 + \beta_1 X_j + \epsilon_{ij}$$

To compute *F-statistics*, we need error sum of squares for the reduced model

$$\begin{aligned} \text{SSE}(R) &= \sum\sum[Y_{ij} - (\beta_0 + \beta_1 X_j)]^2 \\ &= \sum\sum(Y_{ij} - \hat{Y}_{ij})^2 \\ &= \text{SSE} \end{aligned}$$

with degrees of freedom $df_R = n - 2$.

In addition, we also need

$$\text{SSE}(F) = \sum_j \sum_i (Y_{ij} - \bar{Y}_j)^2$$

with degree of freedom $df_F = n - 1$.

```
n = nrow(data)
Y = data$Hardness
X = data$ElapseTime
aggregate(data, list(X=X), mean)
```

```
##    X Hardness ElapseTime
## 1 16   200.00         16
## 2 24   219.00         24
## 3 32   234.00         32
## 4 40   249.25         40
```

```
mu_j = aggregate(data, list(X=X), mean)[, 2]
```

```
# sample size
n = nrow(data)
print(paste0("Sample size ", n))
```

```
## [1] "Sample size 16"
```

```
# find SSE_R
lm0 = lm(data$Hardness~data$ElapseTime)
SSE_0 = sum(lm0$residuals^2)
print(paste0("SSE is ", SSE_0))
```

```
## [1] "SSE is 146.425"
```

```r
# find SSPE
e_vector = c()
for (j in data.frame(table(data$ElapseTime))[,1]) {
  data_curr = data[data$ElapseTime == as.numeric(j), ]
  lm_curr = lm(data_curr$Hardness~data_curr$ElapseTime)
  e_vector = c(e_vector, sum(lm_curr$residuals^2))
}
SSPE = sum(e_vector)
print(paste0("SSPE is ", SSPE))
```

```
## [1] "SSPE is 128.75"
```

```r
# find SSLF
SSLF = SSE_0 - SSPE
print(paste0("SSLF is ", SSLF))
```

```
## [1] "SSLF is 17.6749999999999"
```

```r
# df: SSLF
df_SSLF = 4-2

# df: SSPE
df_SSPE = n-4

# F-statistics
F_stat = SSLF / df_SSLF / (SSPE / df_SSPE)
print(paste0("F-statistics = ", F_stat))
```

```
## [1] "F-statistics = 0.823689320388343"
```

```r
# F critical value
alpha = 0.01
critical = qf(1 - alpha, df_SSLF, df_SSPE)
print(paste0("F critical value = ", critical))
```

```
## [1] "F critical value = 6.9266081401913"
```

From above computation, we fail to reject null since *F-stat* is less than critical value.

(2) Overall, it is preferable to have more sample size. In this example, for each j, we have equal number of X level. However, if I increase the data by an additional observation that fall in one of the partition of j, I would expect a slight increase of *F-stat*. This change will have to be dependent on the information added. For example, if this added observation is within range of Y that is in an existing partition, this will of course raise *F-stat*. However, if an observation that is added provides no information and fall in other Y's area, then I would expect *F-stat* to drop.

The advantage to have more sample size overall and also in each j is to have smaller variance of the data and hopefully more consistent performance. The disadvantage can be the noisy information added if the data is not informative.

(3) When a regression model is tested to lead to conclusion that the underlying model may not be linear, certain transformation can be used to tackle this sort of problems. For example, we can use logarithm, product, and polnomials to introduce additional covariates into the model.

PROBLEM 2 (Ch7, Q7)

Let us load the data first.

```
setwd("C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6101 - Applied Statistics I/Data")
data = read.csv("CH06PR18.csv", header = FALSE)
colnames(data) = c("Y", paste0("X", 1:(ncol(data)-1)))

print("Presents data: ")
```

```
## [1] "Presents data: "
```

```
head(data)
```

```
##      Y X1    X2   X3     X4
## 1 13.5  1  5.02 0.14 123000
## 2 12.0 14  8.19 0.27 104079
## 3 10.5 16  3.00 0.00  39998
## 4 15.0  4 10.70 0.05  57112
## 5 14.0 11  8.97 0.07  60000
## 6 10.5 15  9.45 0.24 101385
```

(1) Let us compute a variety of sum of squares residuals.

```
LM1 = lm(data$Y~data$X1)
LM2 = lm(data$Y~data$X2)
LM4 = lm(data$Y~data$X4)
LM12 = lm(data$Y~data$X1+data$X2)
LM14 = lm(data$Y~data$X1+data$X4)
LM124 = lm(data$Y~data$X1+data$X2+data$X4)
LM1234 = lm(data$Y~., data = data)

SSR_X2 = sum((mean(data$Y) - LM2$fitted.values)^2)
SSR_X4 = sum((mean(data$Y) - LM4$fitted.values)^2)
SSR_X1X2 = sum((mean(data$Y) - LM12$fitted.values)^2)
SSR_X1X4 = sum((mean(data$Y) - LM14$fitted.values)^2)
SSR_X1X2X4 = sum((mean(data$Y) - LM124$fitted.values)^2)
SSR_X1X2X3X4 = sum((mean(data$Y) - LM1234$fitted.values)^2)
SSE_X1X2X3X4 = sum((data$Y - LM1234$fitted.values)^2)

print(paste0("SSR(X2) = ", SSR_X2))
```

```
## [1] "SSR(X2) = 40.5033307305865"
```

```
print(paste0("SSR(X1|X2) = ", SSR_X1X2 - SSR_X2))
```

```
## [1] "SSR(X1|X2) = 47.1171999295226"
```

```r
print(paste0("SSR(X1|X4) = ", SSR_X1X4 - SSR_X4))
```

```
## [1] "SSR(X1|X4) = 42.2745683242814"
```

```r
print(paste0("SSR(X2|X1,X4) = ", SSR_X1X2X4 - SSR_X1X4))
```

```
## [1] "SSR(X2|X1,X4) = 27.8574934834163"
```

```r
print(paste0("SSR(X3|X1,X2,X4) = ", SSR_X1X2X3X4 - SSR_X1X2X4))
```

```
## [1] "SSR(X3|X1,X2,X4) = 0.41974626294018"
```

```r
print(paste0("SSE(X1,X2,X3,X4) = ", SSE_X1X2X3X4))
```

```
## [1] "SSE(X1,X2,X3,X4) = 98.2305939428886"
```

(2) Let us state the hypothesis

$$H_0 : \beta_3 = 0, \text{ vs. } H_1 : \beta_3 \neq 0$$

and let us compute *F-stat*

```r
n = nrow(data)
F_stat = (SSR_X1X2X3X4 - SSR_X1X2X4)/1 / ((SSE_X1X2X3X4)/(n-5))
print(paste0("F-stat = ", F_stat))
```

```
## [1] "F-stat = 0.324753365555346"
```

```r
alpha = 0.01
critical = qf(1 - alpha, 1, n-5)
print(paste0("F critical value = ", critical))
```

```
## [1] "F critical value = 6.98057781279638"
```

```r
p_val = df(F_stat, 1, n-5)
print(paste0("p-value = ", p_val))
```

```
## [1] "p-value = 0.592119971610236"
```

From *F-test* above, we observe that the test statistic is less than critical value, hence we fail to reject null hypothesis.

PROBLEM 3 (Ch7, Q10)

Recall the above data:

```r
setwd("C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6101 - Applied Statistics I/Data")
data = read.csv("CH06PR18.csv", header = FALSE)
colnames(data) = c("Y", paste0("X", 1:(ncol(data)-1)))

print("Presents data: ")
```

```
## [1] "Presents data: "
```

```r
head(data)
```

```
##      Y X1    X2   X3     X4
## 1 13.5  1  5.02 0.14 123000
## 2 12.0 14  8.19 0.27 104079
## 3 10.5 16  3.00 0.00  39998
## 4 15.0  4 10.70 0.05  57112
## 5 14.0 11  8.97 0.07  60000
## 6 10.5 15  9.45 0.24 101385
```

Recall the band problem in Problem 6.5 and let us load the data first.

```r
setwd("C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6101 - Applied Statistics I/Data")
data_0 = read.csv("CH06PR05.csv", header = FALSE)
colnames(data_0) <- c("Y", "X1", "X2")
```

Let us state the hypothesis first:

$$H_0 : \beta_1 = -0.1, \beta_2 = 0$$

$$H_1 : \text{at least one does not hold}$$

First, we have the full model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \epsilon_i$$

and next we have reduced model:

$$Y_i + 0.1X_{i1} - 0.4X_{i2} = \beta_0 + \beta_3 X_{i3} + \beta_4 X_{i4} + \epsilon_i$$

from PROBLEM 2, we found that

```r
print(paste0("SSE(X1,X2,X3,X4) = ", SSE_X1X2X3X4))
```

```
## [1] "SSE(X1,X2,X3,X4) = 98.2305939428886"
```

```r
print(paste0("df_full = ", n-5))
```

```
## [1] "df_full = 76"
```

Now we run linear regression for reduced model:

```r
LM_X3X4_H0 = lm((data$Y + 0.1*data$X1 - 0.4*data$X4)~data$X3 + data$X4)
SSE_X3X4 = sum(LM_X3X4_H0$residuals^2)
print(paste0("SSE(X3,X4) = ", SSE_X3X4))
```

```
## [1] "SSE(X3,X4) = 125.540751547772"
```

```r
print(paste0("df_reduced = ", n-3))
```

```
## [1] "df_reduced = 78"
```

Thus, the *f-statistic* can be computed

```r
F_stat = (SSE_X3X4 - SSE_X1X2X3X4) / 2 / (SSE_X1X2X3X4 / n-5)
print(paste0("F-stat = ", F_stat))
```

```
## [1] "F-stat = -3.60551398268141"
```

```r
alpha = 0.01
critical = qf(1 - alpha, 2, n-5)
print(paste0("F critical value = ", critical))
```

```
## [1] "F critical value = 4.89583988401818"
```

From results above, we observe that *F-stat* is less than critical value, so we fail to reject null hypothesis.

PROBLEM 4 (Ch7, Q16)

Let us refer to the band problem in 6.5.

```r
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")
head(data)
```

```
##    Y X1 X2
## 1 64  4  2
## 2 73  4  4
## 3 61  4  2
## 4 76  4  4
## 5 72  6  2
## 6 80  6  4
```

(1) Let us transform the data:

```r
n = nrow(data)
s1 = sqrt(sum((data$X1 - mean(data$X1))^2) / (n-1))
s2 = sqrt(sum((data$X2 - mean(data$X2))^2) / (n-1))
sy = sqrt(sum((data$Y - mean(data$Y))^2) / (n-1))
Y_star = (data$Y - mean(data$Y))/sy/sqrt(n-1)
x1_star = (data$X1 - mean(data$X1))/s1/sqrt(n-1)
x2_star = (data$X2 - mean(data$X2))/s2/sqrt(n-1)

LM_standard = lm(Y_star~x1_star+x2_star-1)
summary(LM_standard)
```

```
##
## Call:
## lm(formula = Y_star ~ x1_star + x2_star - 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.099209 -0.039740  0.000564  0.035794  0.094699
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## x1_star  0.89239    0.05852  15.250 4.09e-10 ***
## x2_star  0.39458    0.05852   6.743 9.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05852 on 14 degrees of freedom
## Multiple R-squared:  0.9521, Adjusted R-squared:  0.9452
## F-statistic:   139 on 2 and 14 DF,  p-value: 5.82e-10
```

From above results, we have standardized regression model:

$$\hat{Y}^* = 0.89X_1^* + 0.39X_2^*$$

(2) Interpretation: the standardized regression coefficient is actually related with the ordinary multiple regression model. In formula, we have

$$\beta_k = \left(\frac{S_Y}{S_k}\right)\beta_k^*$$

while $k = 1, 2, ..., p-1$. Moreover, for β_0, we have

$$\beta_0 = \bar{Y} - \beta_1\bar{X}_1 - \cdots - \beta_{p-1}\bar{X}_{p-1}$$

and we can see that this means that a unit of change of X_i of the original data produces β_i change on response variable Y. In relation with standardized data, a unit of change on standardized data X_i^* produces $\beta_k^* = \left(\frac{S_i}{S_Y}\right)\beta_k$ change on standardized response Y_i^*. In other words, the interpretation follows the same understanding of ordinary multiple regression model but with a scaling factors involving ratios of standard deviations.

(3) Transform back to original coefficients. In this problem, let us recall the following:

```r
print(paste0("s_Y = ", sy))
```

```
## [1] "s_Y = 11.4513463546141"
```

```r
print(paste0("s1 = ", s1))
```

```
## [1] "s1 = 2.3094010767585"
```

```r
print(paste0("s2 = ", s2))
```

```
## [1] "s2 = 1.03279555898864"
```

```r
print(paste0("Coefficients are: "))
```

```
## [1] "Coefficients are: "
```

```r
print(summary(LM_standard)$coefficient)
```

```
##          Estimate Std. Error   t value     Pr(>|t|)
## x1_star 0.8923929 0.05851802 15.249880 4.089332e-10
## x2_star 0.3945807 0.05851802  6.742892 9.426910e-06
```

and thus, we can compute the following:

```r
b1 = sy / s1 * summary(LM_standard)$coefficient[1,1]
b2 = sy / s2 * summary(LM_standard)$coefficient[2,1]
b0 = mean(data$Y) - b1*mean(data$X1) - b2*mean(data$X2)
print(paste0("Original coefficients are:"))
```

```
## [1] "Original coefficients are:"
```

```r
print(data.frame(b1=b1, b2=b2, b0=b0))
```

```
##      b1    b2    b0
## 1 4.425 4.375 37.65
```

and thus we conclude that the original model is

$$Y_i = 37.65 + 4.425X_1 + 4.375X_2$$

and we are done.

PROBLEM 5 (Ch7, Q24)

Referring to the band problem above, let us fit a simple regression model

(1) Simple Regression $Y\ X_1$:

```r
LM_Y_X1 = lm(data$Y~data$X1)
summary(LM_Y_X1)
```

```
##
## Call:
## lm(formula = data$Y ~ data$X1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.475 -4.688 -0.100  4.638  7.525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50.775      4.395  11.554 1.52e-08 ***
## data$X1        4.425      0.598   7.399 3.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.349 on 14 degrees of freedom
## Multiple R-squared:  0.7964, Adjusted R-squared:  0.7818
## F-statistic: 54.75 on 1 and 14 DF,  p-value: 3.356e-06
```

and we get the linear model

$$\hat{Y} = 50.775 + 4.425X_1$$

(2) Comparing with the previous results in 6.5b (which is the last homework), let me referring to the solution first.

From previous result, we had linear regression model

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

and we noticed that the coefficient on X_1 does not change.

(3) What happened?

Let us recall the previous model:

```r
LM_Y_X2 = lm(data$Y~data$X2)
LM = lm(Y~., data = data)

SSR_X1 = sum((mean(data$Y) - LM_Y_X1$fitted.values)^2)
SSR_X2 = sum((mean(data$Y) - LM_Y_X2$fitted.values)^2)
SSE_X1X2 = sum((mean(data$Y) - LM$fitted.values)^2)
SSR_X1X2 = SSE_X1X2 - SSR_X2

print(paste0("SSR(X1) = ", SSR_X1))
```

```
## [1] "SSR(X1) = 1566.45"
```

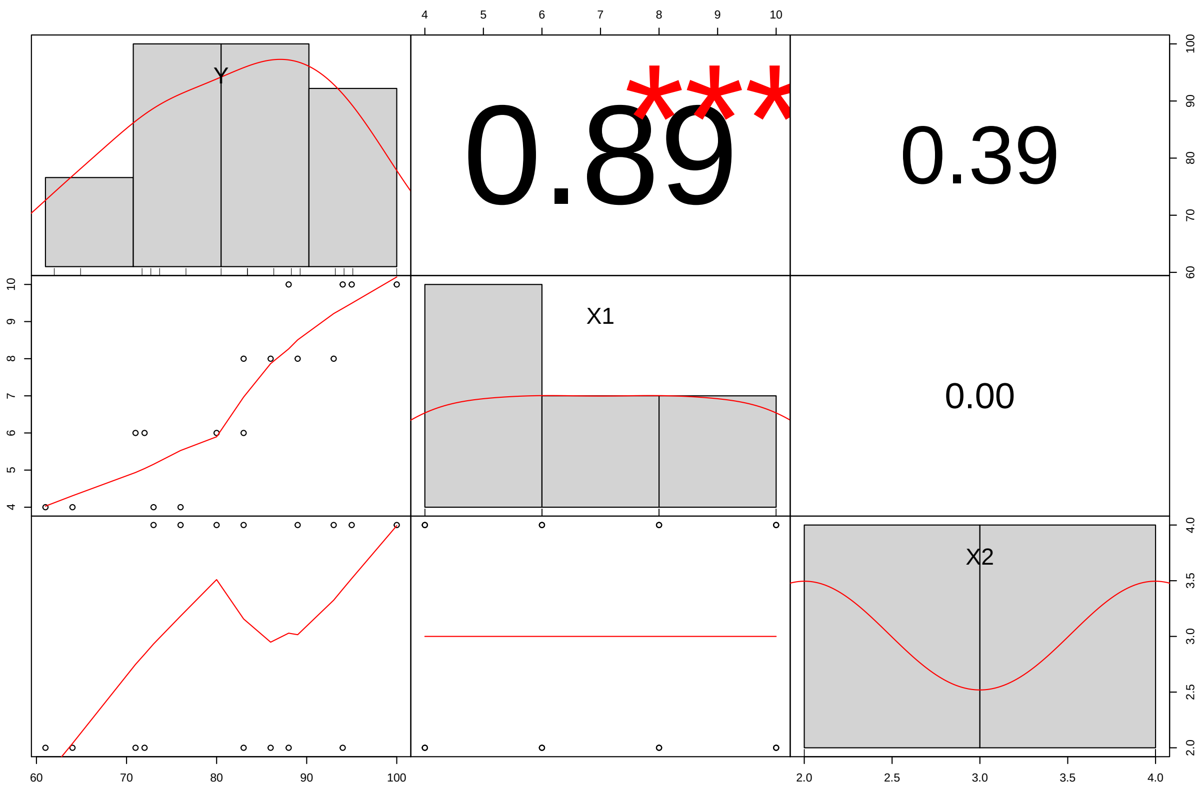
```r
print(paste0("SSR(X1|X2) = ", SSR_X1X2))
```

```
## [1] "SSR(X1|X2) = 1566.45"
```

and the answers are that the are the same values.

(4) Correlation matrix was obtained previously, let us recall the results:

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



From parts (2) and (3), we essentially concluded that given X_2, we have X_1 contributing to the sum of squares exactly the same. In other words, X_1 and X_2 are uncorrelated, which is a result from correlation table above.

PROBLEM 6 (Ch7, Q37)

Let us recall the data first:

```r
# Data
setwd("C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6101 - Applied Statistics I/Data")
data <- read.csv("APPENC02.csv", header = FALSE)
colnames(data) <- c(
  "ID",
```

```r
  "Country",
  "State",
  "Land_Area",
  "Total_Population",
  "Perc_Popu_18_34",
  "Perc_Popu_Over_65",
  "Num_Active_Phy",
  "Num_Hospital_Beds",
  "Total_Serious_Crimes",
  "Percent_High_School",
  "Percent_Bachelor_Deg",
  "Percent_Below_Poverty",
  "Percent_Unemployment",
  "Per_Capita_Income",
  "Total_Personal_Income",
  "Geographic_Region"
)
dim(data); head(data)
```

```
## [1] 440  17
```

```
##   ID     Country State Land_Area Total_Population Perc_Popu_18_34
## 1  1 Los_Angeles    CA      4060          8863164            32.1
## 2  2        Cook    IL       946          5105067            29.2
## 3  3      Harris    TX      1729          2818199            31.3
## 4  4   San_Diego    CA      4205          2498016            33.5
## 5  5      Orange    CA       790          2410556            32.6
## 6  6       Kings    NY        71          2300664            28.3
##   Perc_Popu_Over_65 Num_Active_Phy Num_Hospital_Beds Total_Serious_Crimes
## 1               9.7          23677             27700               688936
## 2              12.4          15153             21550               436936
## 3               7.1           7553             12449               253526
## 4              10.9           5905              6179               173821
## 5               9.2           6062              6369               144524
## 6              12.4           4861              8942               680966
##   Percent_High_School Percent_Bachelor_Deg Percent_Below_Poverty
## 1                70.0                 22.3                  11.6
## 2                73.4                 22.8                  11.1
## 3                74.9                 25.4                  12.5
## 4                81.9                 25.3                   8.1
## 5                81.2                 27.8                   5.2
## 6                63.7                 16.6                  19.5
##   Percent_Unemployment Per_Capita_Income Total_Personal_Income
## 1                  8.0             20786                184230
## 2                  7.2             21729                110928
## 3                  5.7             19517                 55003
## 4                  6.1             19588                 48931
## 5                  4.8             24400                 58818
## 6                  9.5             16803                 38658
##   Geographic_Region
## 1                 4
## 2                 2
## 3                 3
```

```
## 4                  4
## 5                  4
## 6                  1
```

Let us define some variables using the notation provided in the problems

```
Y = data$Num_Active_Phy
X1 = data$Total_Population
X2 = data$Total_Personal_Income
X3 = data$Land_Area
X4 = data$Perc_Popu_Over_65
X5 = data$Num_Hospital_Beds
X6 = data$Total_Serious_Crimes

LM12 = lm(Y~X1+X2)
LM123 = lm(Y~X1+X2+X3)
LM124 = lm(Y~X1+X2+X4)
LM125 = lm(Y~X1+X2+X5)
LM126 = lm(Y~X1+X2+X6)

SSR_12 = sum((Y - LM12$fitted.values)^2)
SSR_123 = sum((Y - LM123$fitted.values)^2)
SSR_124 = sum((Y - LM124$fitted.values)^2)
SSR_125 = sum((Y - LM125$fitted.values)^2)
SSR_126 = sum((Y - LM126$fitted.values)^2)

R_sq_3_12 = (SSR_12 - SSR_123) / SSR_12
R_sq_4_12 = (SSR_12 - SSR_124) / SSR_12
R_sq_5_12 = (SSR_12 - SSR_125) / SSR_12
R_sq_6_12 = (SSR_12 - SSR_126) / SSR_12

print(paste0("R_Y,3|12 = ", R_sq_3_12))
```

```
## [1] "R_Y,3|12 = 0.0288249536468882"
```

```
print(paste0("R_Y,4|12 = ", R_sq_4_12))
```

```
## [1] "R_Y,4|12 = 0.00384236729898615"
```

```
print(paste0("R_Y,5|12 = ", R_sq_5_12))
```

```
## [1] "R_Y,5|12 = 0.553818175062583"
```

```
print(paste0("R_Y,6|12 = ", R_sq_6_12))
```

```
## [1] "R_Y,6|12 = 0.00732340826775479"
```

(2) From the results above, we observe that introducing additional variable X_5 is the best because it raises the corresponding R^2 the highest, i.e. $R_{Y,5|12} = 0.55$ while the other raised R^2's are lower than this value.

Hence, X_5 is the best and the extra sum of squares associated with this variable is larger than that of the others.

(3) Let us consider full model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_5 X_{i5} + \epsilon_i$$

Let us state the hypothesis:

$$H_0 : \beta_5 = 0 \text{ vs. } H_1 : \beta_5 \neq 0$$

and we compute

```
print(paste0("SSR(X5|X1,X2) = ", SSR_125))
```

```
## [1] "SSR(X5|X1,X2) = 62896949.4876823"
```

```
SSR_5_12 = SSR_12 - SSR_125
print(paste0("SSR(X1,X2,X5) = ", SSR_5_12))
```

```
## [1] "SSR(X1,X2,X5) = 78070131.5817998"
```

```
F_stat = (SSR_5_12/1) / (SSR_125 / (nrow(data) - 4))
print(paste0("F-stat = ", F_stat))
```

```
## [1] "F-stat = 541.180099304034"
```

```
alpha = 0.01
critical = qf(1-alpha, 1, nrow(data)-3)
print(paste0("Critical value = ", critical))
```

```
## [1] "Critical value = 6.69322297445887"
```

and thus we reject null hypothesis and conclude that $\beta_5 \neq 0$.

PROBLEM 7

Let us derive IAC.

Proof Let denote the following: - Let $\hat{\theta}_n = \arg\max_\theta L(\theta|X_i)$ be the MLE. - Let $l_n = l(\hat{\theta}_n|X_i)$ be the maximal value of the empirical log-likelihood function. - Moreover, we define $\bar{l}(\theta) = \mathbb{E}(l(\theta|X_1))$

We want to show that in model selection AIC is essentially the following form

$$\text{AIC} = -2l_n - 2p$$

where p is the dimension of the model.

The idea is to adjust the empirical risk to be an unbiased estimator of the true risk of a parametric model. The loss function is the negative log-likelihood function so the empirical risk is

$$\hat{R}_n(\hat{\theta}_n) = -l_n = -l(\hat{\theta}_n|X_i)$$

The true risk of the MLE is

$$R(\hat{\theta}_n) = \mathbb{E}(-n\bar{l}(\hat{\theta}_n))$$

To analyze the true risk, we examine the asymptotic behavior of $\bar{l}(\hat{\theta}_n)$ around $\theta^* = \arg\max_\theta \bar{l}(\theta)$ which is the population MLE.

$$\begin{array}{rcl} \bar{l}(\hat{\theta}_n) & \approx & \bar{l}(\theta^*) + (\hat{\theta}_n - \theta^*)^T \nabla \bar{l}(\theta^8) + \frac{1}{2}(\hat{\theta}_n - \theta^*)^T \nabla\nabla \bar{l}(\theta^*)(\hat{\theta}_n - \theta^*) \\ & = & \bar{l}(\theta^*) + \frac{1}{2}(\hat{\theta}_n - \theta^*)^T I(\theta^*)(\hat{\theta}_n - \theta^*) \end{array}$$

thus the true risk is

$$R(\hat{\theta}_n) = -n\mathbb{E}(\bar{l}(\hat{\theta}_n)) \approx -n\bar{l}(\theta^*) - \frac{n}{2}\mathbb{E}\big((\hat{\theta}_n - \theta^*)^T I(\theta^*)(\hat{\theta_n} - \theta^*)\big)$$

For empirical risk, we expand l_n

$$\begin{array}{rcl} l_n & = & \sum_{i=1}^n l(\hat{\theta}_n|X_i) \\ & \approx & \sum_{i=1}^n l(\theta^*|X_i) + \underbrace{(\hat{\theta}_n - \theta^*)^T \sum_{i=1}^n \nabla l(\theta^*|X_i)}_{\mathrm{I}} + \underbrace{\frac{1}{2}(\hat{\theta}_n - \theta^*)^T \sum_{i=1}^n \nabla\nabla l(\theta^*|X_i)(\hat{\theta}_n - \theta^*)}_{\mathrm{II}} \end{array}$$

while

$$\mathrm{I} \approx -n(\hat{\theta}_n - \theta^*)^T I(\theta^*)(\hat{\theta}_n - \theta^*)$$

and

$$\mathrm{II} = \frac{n}{2}(\hat{\theta}_n - \theta^*)^T I(\theta^*)(\hat{\theta}_n - \theta^*)$$

Thus, putting everthing together and taking expectation, we obtain

$$\mathbb{E}(\hat{R}_n(\hat{\theta}_n)) = -\mathbb{E}(l_n) = -n\bar{l} + \frac{n}{2}\mathbb{E}\big((\hat{\theta}_n - \theta^*)^T I(\theta^*)(\hat{\theta_n} - \theta^*)\big)$$

From asymptotic behavior of MLE, we have

$$\sqrt{n}(\hat{\theta}_n - \theta^*) \approx \mathcal{N}(0, I^{-1}(\theta^*))$$

and thus

$$n(\hat{\theta}_n - \theta^*)^T I(\theta^*)(\hat{\theta}_n - \theta^*) \approx \chi_p^2$$

To ensure asymptotic estimator is unbiased, we need

$$\hat{R}_n(\hat{\theta}_n) + p = -l_n + p$$

and thus, by multiplying two, we have

$$\mathrm{AIC} = -2l_n + 2p = -2\log(L(\hat{\theta})) + 2p$$

and we are done.

QED