# Regression HW 8 Solutions

Jun Lu, Mingfan Meng

Fall 2023

## Contents



30 pts

Reminder:

- Lab 5 (separate submission)
- Project progress update

### 1. Problem 9.18. (Job Proficiency, p. 379. 30pts. a: 5, b: 15 (10 for the statistics, 5 for comments.), c: 10)

```
jp.data <- read.table("./CH09PR10.txt", header=F)
colnames(jp.data) <- c("proficiency", "x1", "x2", "x3", "x4")
# head(jp.data)
```

#### a. Variable selection (use stepwise in R) 5 pts.

```
jp.all <- lm(proficiency ~ x1 + x2 + x3 + x4, data = jp.data)
step(jp.all, direction="both")
```

```
## Start:  AIC=74.95
## proficiency ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq     RSS     AIC
## - x2    1     12.22  348.20  73.847
## <none>               335.98  74.954
## - x4    1    260.74  596.72  87.314
## - x1    1    759.83 1095.81 102.509
## - x3    1   1064.15 1400.13 108.636
##
## Step:  AIC=73.85
## proficiency ~ x1 + x3 + x4
##
##        Df Sum of Sq     RSS     AIC
## <none>               348.20  73.847
## + x2    1     12.22  335.98  74.954
## - x4    1    258.46  606.66  85.727
## - x1    1    763.12 1111.31 100.861
```

```
## - x3    1   1324.39 1672.59 111.081
##
## Call:
## lm(formula = proficiency ~ x1 + x3 + x4, data = jp.data)
##
## Coefficients:
## (Intercept)           x1           x3           x4
##   -124.2000       0.2963       1.3570       0.5174
```

- R's stepwise selection algorithm use AIC (by default) as the selection criterion. Based on AIC, the predictors (X1, X3, X4) are selected to fit a regression model for job proficiency.
- The estimated regression function is:

$$\widehat{proficiency} = -124.2000 + 0.2963X_1 + 1.3570X_3 + 0.5174X_4.$$

**b. Compute the model selection criteria. (15 pts. 10 for the statistics, 5 for comments.)**

- Copy the `selcri()` function from the Lab 5.

```
selcri<-function(lmout)
  {
  n <- length(lmout$fit)
  rsq <- summary(lmout)$r.sq
  adj.rsq <- summary(lmout)$adj.r.sq
  aic <- extractAIC(lmout)[2]
  bic <- extractAIC(lmout, k = log(n))[2]
  press <- sum((lmout$residuals/(1 - hatvalues(lmout)))^2)
  cbind(rsq, adj.rsq, aic, bic, press)
  }
```

- To compute Mallows' C, we need to set the pool of all possible predictors. In this problem the pool is ($X_1$, $X_2$, $X_3$, $X_4$). Get the MSE from the 4-predictor model.

```
anova(jp.all)
```

```
## Analysis of Variance Table
##
## Response: proficiency
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 2395.9  2395.9 142.620 1.480e-10 ***
## x2         1 1807.0  1807.0 107.565 1.708e-09 ***
## x3         1 4254.5  4254.5 253.259 8.045e-13 ***
## x4         1  260.7   260.7  15.521   0.00081 ***
## Residuals 20  336.0    16.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mse.all <- 16.8  # from output
```

- Alternatively, you can get the "Residual standard error: 4.099" from the `summary()` output. Then use $MSE = (4.099)^2 = 16.8$.

```
jp.4par <- lm(proficiency ~ x1 + x3 + x4, data=jp.data)
anova(jp.4par)
```

**i. For the 3-predictor (X1, X3, X4), 4-parameter model from part (a)**

```
## Analysis of Variance Table
##
## Response: proficiency
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 2395.9  2395.9 144.496 7.054e-11 ***
## x3         1 6051.5  6051.5 364.969 9.359e-15 ***
## x4         1  258.5   258.5  15.588 0.0007354 ***
## Residuals 21  348.2    16.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sse.4 <- 348.2  # from above output

selcri(jp.4par)
```

```
##           rsq   adj.rsq      aic      bic   press
## [1,] 0.9615422 0.9560482 73.84732 78.72282 471.452
```

```
n <- nrow(jp.data)
p <- length(jp.4par$coefficients)
mc.4 <- sse.4/mse.all - (n-2*p)
noquote(paste("Mallows' C: ", mc.4))
```

```
## [1] Mallows' C:  3.72619047619047
```

```
anova(jp.all)
```

**ii. For the 4-predictor (X1, X2, X3, X4), 5-parameter model (jp.all)**

```
## Analysis of Variance Table
##
## Response: proficiency
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 2395.9  2395.9 142.620 1.480e-10 ***
## x2         1 1807.0  1807.0 107.565 1.708e-09 ***
## x3         1 4254.5  4254.5 253.259 8.045e-13 ***
## x4         1  260.7   260.7  15.521   0.00081 ***
## Residuals 20  336.0    16.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sse.5 <- 336.0  # from above output

selcri(jp.all)
```

```
##           rsq   adj.rsq      aic      bic    press
## [1,] 0.9628918 0.9554702 74.95421 81.04859 518.9885
```

```
n <- nrow(jp.data)
p <- length(jp.all$coefficients)
mc.5 <- sse.5/mse.all - (n-2*p)
noquote(paste("Mallows' C: ", round(mc.5)))  # it should be 5.
```

```
## [1] Mallows' C:  5
```

**iii. Comments**

- $R^2$ show both models explain a large proportion of the variation in job proficiency (Y). However, we should not use $R^2$ for model selection since these two models have different number of parameters.
- $Adj - R^2$ for the 3-predictor (4-parameter) model is larger, but only by a small amount.
- Both AIC and SBC (SBC is also known as BIC) for the 3-predictor (4-parameter) model are smaller.
- PRESS for the 3-predictor (4-parameter) model is smaller.
- Mallows $C_p$ for the 3-predictor (4-parameter) model is preferred because $C_4 = 3.7$ is close to the 4 (the number of parameters), and it is smaller than $C_5 = 5$ from the 4-predictor model.
- The above measures all indicate the 3-predictor (X1, X3, X4), 4-parameter model is preferred over the model using all 4 predictors (5 parameters)

**c. Use `regsubsets()` function, and compare the resulting model comparison criteria. (10 pts)**

```
library(leaps)
temp <- regsubsets(proficiency ~ x1 + x2 + x3 + x4, data = jp.data)
jp.bestsub <- summary(temp)
jp.bestsub
```

```
## Subset selection object
## Call: regsubsets.formula(proficiency ~ x1 + x2 + x3 + x4, data = jp.data)
## 4 Variables  (and intercept)
##    Forced in Forced out
## x1     FALSE      FALSE
## x2     FALSE      FALSE
## x3     FALSE      FALSE
## x4     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4
## 1  ( 1 ) " " " " "*" " "
## 2  ( 1 ) "*" " " "*" " "
## 3  ( 1 ) "*" " " "*" "*"
## 4  ( 1 ) "*" "*" "*" "*"
```

```
data.frame(n.pred = seq(1:4), n.par = seq(2:5),
           adj.R2 = jp.bestsub$adjr2, SBC = jp.bestsub$bic,
           Mallows.C = jp.bestsub$cp)
```

```
##   n.pred n.par    adj.R2       SBC Mallows.C
## 1      1     1 0.7962344 -34.39587 84.246496
## 2      2     2 0.9269043 -57.91831 17.112978
## 3      3     3 0.9560482 -68.57933  3.727399
## 4      4     4 0.9554702 -66.25356  5.000000
```

- The function `regsubsets()` can compute adjusted-$R^2$, Mallows-C, and SBC (BIC).
- The adjusted-$R^2$ and Mallows-C from `regsubsets()` are the same as the values computed in b.
- SBC (BIC) values from `regsubsets()` are different from those in b.
- (FYI. NO deduction if not included.) Why the SBC are "different?" Different R functions may use slightly different function to compute AIC and SBC (BIC). They offer differ by a constant that depends on the sample size $n$. For example, for the 4-parameter (3 predictor) model:

$$\widehat{proficiency} = -124.2000 + 0.2963X_1 + 1.3570X_3 + 0.5174X_4.$$

```
extractAIC(jp.4par, k=log(25))  # Recall n = 25
```

```
## [1]  4.00000 78.72282
```

```
AIC(jp.4par, k=log(25))
```

```
## [1] 152.8886
```

```
jp.bestsub$bic[3]
```

```
## [1] -68.57933
```

That said, as long as we use the AIC or SBC (BIC) from the same function, we can safely use them for model comparison. The difference in AIC or SBC (BIC) between 2 candidate models will not be affected by which function we use.

```
extractAIC(jp.4par, k=log(25))[2] - extractAIC(jp.all, k=log(25))[2]
```

```
## [1] -2.325772
```

```
AIC(jp.4par, k=log(25)) - AIC(jp.all, k=log(25))
```

```
## [1] -2.325772
```

```
jp.bestsub$bic[3] -  jp.bestsub$bic[4]
```

```
## [1] -2.325772
```

**This is the end of HW 8.**