Homework 8

Yunting Chiu

2021-04-14

1. (**9.1**) A speaker stated: "In well-designed experiments involving quantitative explanatory variables, a procedure for reducing the number of explanatory variables after the data are obtained is not necessary." Do you agree? Discuss.

- An explanatory variable is a type of independent variable.
- Assume that these variables have met the assumptions and that there is no collinearity in the well-designed experiments. The better approach is to carry out procedures to reduce the number of explanatory variables that are not significant, because these variables cannot explain the model and may reduce the precision of the outcome.

2. (**9.5**) In forward stepwise regression, what advantage is there in using a relatively small α to-enter value for adding variables? What advantage is there in using a larger α-to-enter value?

-

4. (**Continuing 6.27 from an earlier homework**) In a small-scale regression study, the following data were obtained,

Y	X1	X2
42.0	7.0	33.0
33.0	4.0	41.0
75.0	16.0	7.0
28.0	3.0	49.0
91.0	21.0	5.0
55.0	8.0	31.0

Make a data frame

```
Y <- c(42, 33, 75, 28, 91, 55)
X1 <- c(7, 4, 16, 3, 21, 8)
X2 <- c(33, 41, 7, 49, 5, 31)

df <- data.frame(Y, X1, X2)
df
```

```
##    Y X1 X2
## 1 42  7 33
## 2 33  4 41
## 3 75 16  7
## 4 28  3 49
## 5 91 21  5
## 6 55  8 31
```

Model selection

```
library(leaps)
df.fit <-  regsubsets(Y ~ X1 + X2, data = df)
summary(df.fit)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ X1 + X2, data = df)
## 2 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## 1 subsets of each size up to 2
## Selection Algorithm: exhaustive
##          X1  X2
## 1  ( 1 ) "*" " "
## 2  ( 1 ) "*" "*"
```

1.1 Find out the largest adjusted R squares

```
summary(df.fit)$adjr2
```

```
## [1] 0.9724995 0.9663230
```

```
which.max(summary(df.fit)$adjr2)
```

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

1.2 Find out the smallest CP

```
summary(df.fit)$cp
```

```
## [1] 1.266385 3.000000
```

```
which.min(summary(df.fit)$cp)
```

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

1.3 Find out the smallest BIC (penalized-likelihood criteria)

```
summary(df.fit)$bic
```

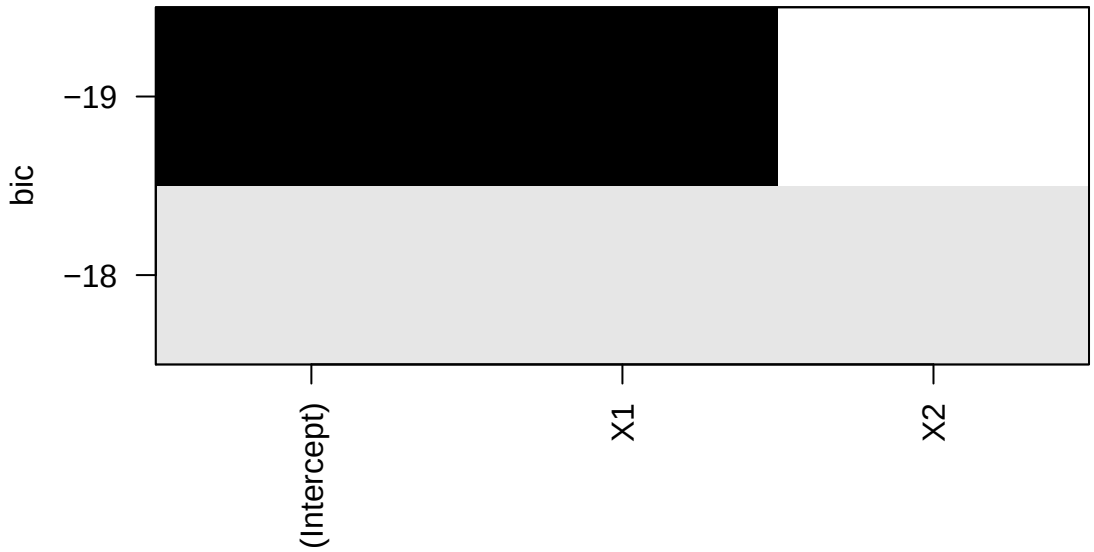
```
## [1] -19.31664 -18.03531
```

```
which.min(summary(df.fit)$bic)
```

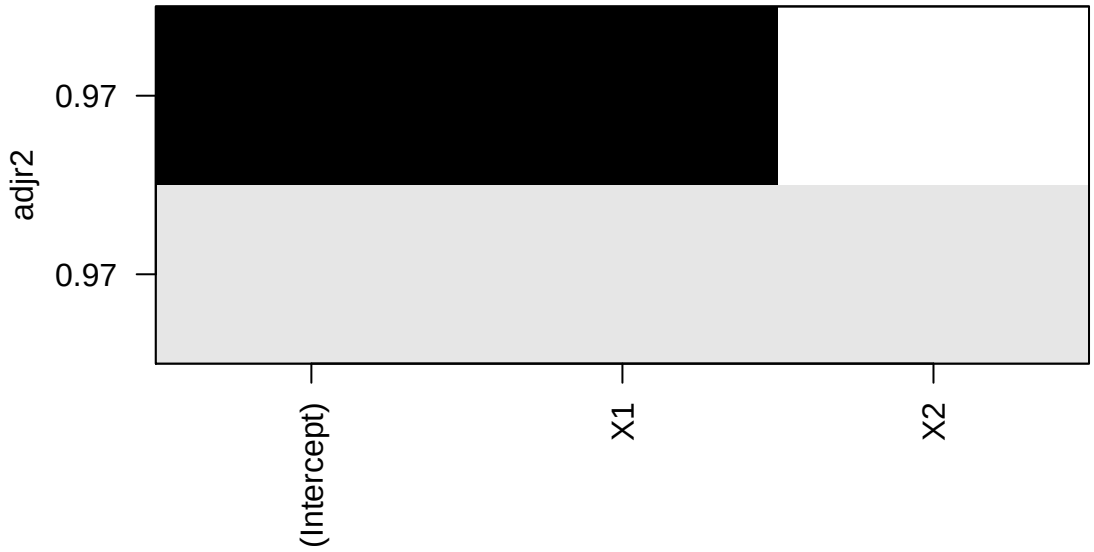
```
## [1] 1
```

2.1 Find out the proper adjusted R and BIC using plot

```
df.fit2 <-  regsubsets( Y ~ ., data = df, method = "backward" )
plot(df.fit2, sclae = "bic")
```




```
plot(df.fit2, scale = "adjr2")
```



According to the result above, the best model is:

```
best1 <- lm(Y ~ X1, data = df)
summary(best1)
```

```
##
## Call:
## lm(formula = Y ~ X1, data = df)
##
## Residuals:
##       1       2       3       4       5       6
## -2.2714 -0.9706 -0.1740 -2.5370 -1.3421  7.2950
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.2361     3.0376   6.662 0.002638 **
## X1            3.4336     0.2575  13.335 0.000183 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.111 on 4 degrees of freedom
## Multiple R-squared:  0.978,  Adjusted R-squared:  0.9725
## F-statistic: 177.8 on 1 and 4 DF,  p-value: 0.0001829
```

3.1 We can also choose the best model by means of a stepwise procedure

```
null = lm( Y ~ 1, data = df )
full = lm( Y ~ ., data = df )

step( null, scope = list(lower = null, upper = full), direction = "forward" )
```

```
## Start:  AIC=39.43
## Y ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + X1    1    3004.4   67.59 18.530
## + X2    1    2913.4  158.64 23.649
## <none>              3072.00 39.430
##
## Step:  AIC=18.53
## Y ~ X1
##
##        Df Sum of Sq    RSS    AIC
## <none>              67.585 18.530
## + X2    1    5.5118 62.074 20.019
```

```
##
## Call:
## lm(formula = Y ~ X1, data = df)
##
## Coefficients:
## (Intercept)           X1
##      20.236        3.434
```

```
step( null, scope = list(lower = null, upper = full), direction = "backward" )
```

```
## Start:  AIC=39.43
## Y ~ 1
```

```
##
## Call:
## lm(formula = Y ~ 1, data = df)
##
## Coefficients:
## (Intercept)
##          54
```

```
step( null, scope = list(lower = null, upper = full), direction = "both" )
```

```
## Start:  AIC=39.43
## Y ~ 1
##
```

```
##        Df Sum of Sq     RSS    AIC
## + X1    1    3004.4   67.59 18.530
## + X2    1    2913.4  158.64 23.649
## <none>               3072.00 39.430
##
## Step:  AIC=18.53
## Y ~ X1
##
##        Df Sum of Sq     RSS    AIC
## <none>                 67.59 18.530
## + X2    1      5.51   62.07 20.019
## - X1    1   3004.41 3072.00 39.430
```

```
##
## Call:
## lm(formula = Y ~ X1, data = df)
##
## Coefficients:
## (Intercept)           X1
##      20.236        3.434
```

In summary, the smaller RSS is `Y~X1`, so the best performance of this model is:

```
best2 <- lm(Y ~ X1, data = df)
summary(best2)
```

```
##
## Call:
## lm(formula = Y ~ X1, data = df)
##
## Residuals:
##       1       2       3       4       5       6
## -2.2714 -0.9706 -0.1740 -2.5370 -1.3421  7.2950
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.2361     3.0376   6.662 0.002638 **
## X1            3.4336     0.2575  13.335 0.000183 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.111 on 4 degrees of freedom
## Multiple R-squared:  0.978,  Adjusted R-squared:  0.9725
## F-statistic: 177.8 on 1 and 4 DF,  p-value: 0.0001829
```