Chapter 6 Exercises: Linear Model Selection and Regularization

Statistical Learning with R

Jon Geiger

March 13, 2022

Conceptual Exercise 4

Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

for a particular value of λ. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

(a) As we increase λ from 0, the training RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

(b) Repeat (a) for test RSS.
(c) Repeat (a) for variance.
(d) Repeat (a) for squared bias.
(e) Repeat (a) for the irreducible error.

Solution

(a) iii: Steadily increase. When $\lambda = 0$, that's a least squares fit, which minimizes the RSS. So, adding squared coefficients to this RSS can only increase it. In other words, for least squares, we minimize the parenthetical expression, so for any $\lambda > 0$, the RSS must increase.

(b) ii: Decrease initially, then start increasing in a U shape. We know that even though the training RSS will steadily increase, we hope that the reduction in variance is enough to overcome the fact that the ridge coefficients are shrinking towards zero. The test RSS should reveal over- or under-fitting, so by regularizing and shrinking the coefficients, the decrease in variance will make for an optimal value of λ, after which point the variance decrease will not outweigh the bias increase.

(c) iv: Steadily Decrease. Shrinkage methods necessarily decrease the variance at the cost of increasing the bias, so increasing λ decreases the ℓ_2 norm of the ridge coefficients, shrinking down the magnitude of the coefficient vector.

(d) iii: Steadily Increase. By shrinking the coefficients down, we are increasing the impact that the intercept term has on the prediction, which lends itself toward a more biased model. A model which is just equal to the intercept β_0 is one with zero variance but maximized bias.

(e) v: Remain Constant. The irreducible error is, as it says, not reducible. That is, it is inherent to the data and the noise introduced into that data, so there is no model we can choose, nor any value of λ, that will reduce the irreducible error.

Applied Exercise 8

In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Use the `rnorm()` function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of length $n = 100$.

(b) Generate a response vector Y of length $n = 100$ according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

where $\beta_0, \beta_1, \beta_2$, and β_3 are constants of your choice.

(c) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model according to C_p, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the `data.frame()` function to create a single data set containing both X and Y.

(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

(e) Now fit a lasso model to the simulated data, again using $X, X^2, \ldots, X^{10}$ as predictors. use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

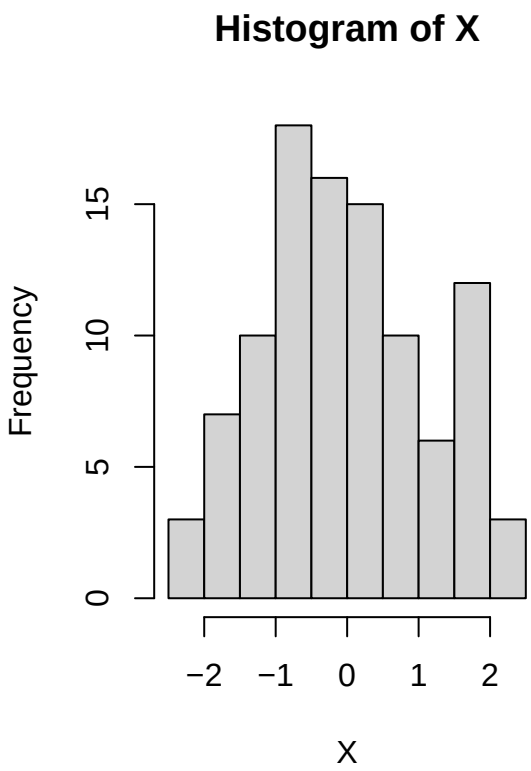
(f) Now generate a response vector Y according to the model

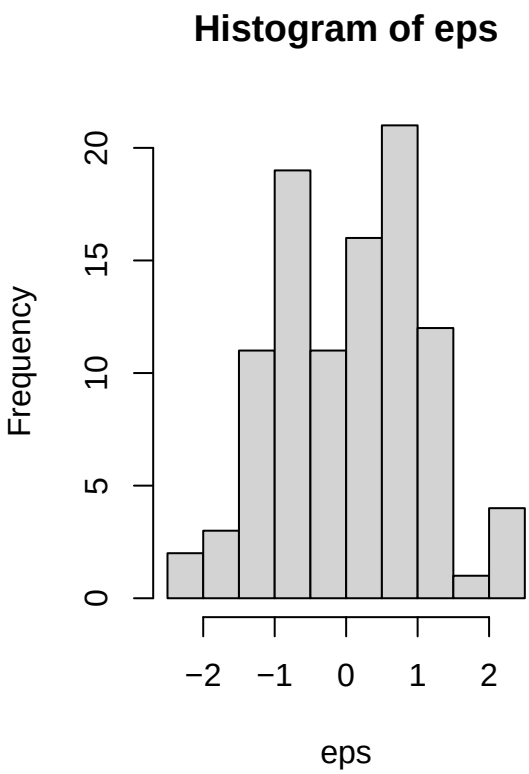
$$Y = \beta_0 + \beta_7 X^7 + \epsilon$$

and perform best subset selection and the lasso. Discuss the results obtained.

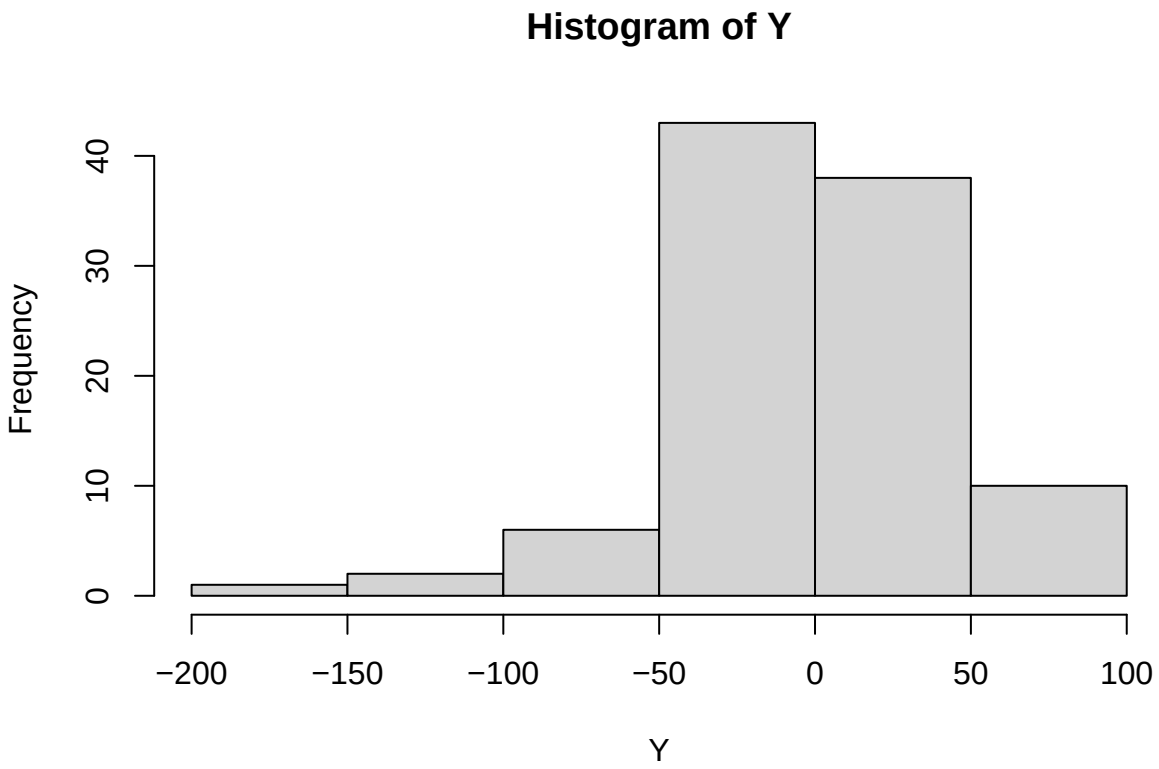
Solution

(a)
```
set.seed(2)
X <- rnorm(100)
eps <- rnorm(100)
par(mfrow=c(1,2))
hist(X); hist(eps)
```


Histogram of X
Frequency
0
5
10
15
−2
−1
0
1
2
X

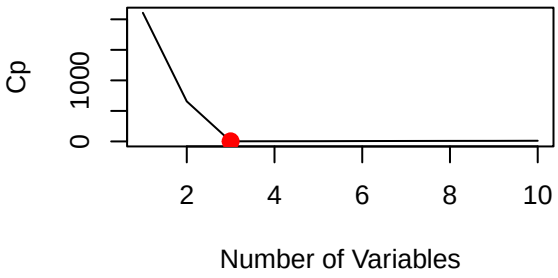

Histogram of eps
Frequency
0
5
10
15
20
−2
−1
0
1
2
eps

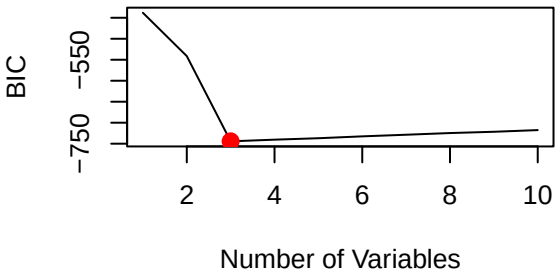
(b)
```
beta_0 <- 1; beta_1 <- 5
beta_2 <- -3; beta_3 <- 10
Y <- beta_0 + beta_1*X + beta_2*X^2 + beta_3*X^3 + eps
hist(Y)
```

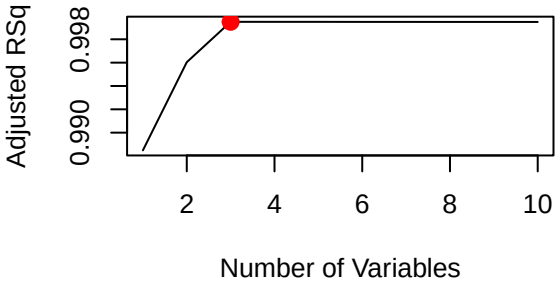


(c)
```
X_matrix <- poly(X, 10, raw = T, simple = T)

model_select <- function(X, Y, method) {

    model <- regsubsets(y = Y,
                        x = X,
                        nvmax = 10,
                        method = method)

    model_summary <- summary(model)

    par(mfrow=c(2,2))

    plot(model_summary$cp, xlab = "Number of Variables",
         ylab = "Cp", type = "l")
    points(which.min(model_summary$cp),
           model_summary$cp[which.min(model_summary$cp)],
           col = "red", cex = 2, pch = 20)

    plot(model_summary$bic, xlab = "Number of Variables",
         ylab = "BIC", type = "l")
    points(which.min(model_summary$bic),
           model_summary$bic[which.min(model_summary$bic)],
           col = "red", cex = 2, pch = 20)

    plot(model_summary$adjr2, xlab = "Number of Variables",
         ylab = "Adjusted RSq", type = "l")
    points(which.max(model_summary$adjr2),
           model_summary$adjr2[which.max(model_summary$adjr2)],
           col = "red", cex = 2, pch = 20)

    print(model_summary$outmat)
    print(coef(model, id=which.min(model_summary$bic)))

}

model_select(X_matrix, Y, "exhaustive")
```

```
##          1   2   3   4   5   6   7   8   9   10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " "
## 3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " "
## 4  ( 1 )  "*" "*" "*" " " "*" " " " " " " " " "
## 5  ( 1 )  "*" "*" "*" " " "*" " " " " " " "*" " "
## 6  ( 1 )  "*" "*" "*" " " "*" " " " " " " "*" "*"
## 7  ( 1 )  "*" "*" "*" "*" " " "*" "*" " " "*" " "
## 8  ( 1 )  "*" "*" "*" "*" "*" "*" " " "*" " " "*"
## 9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
## (Intercept)           1           2           3
##   0.9621529   4.9867296  -2.9495582   9.9875115
```





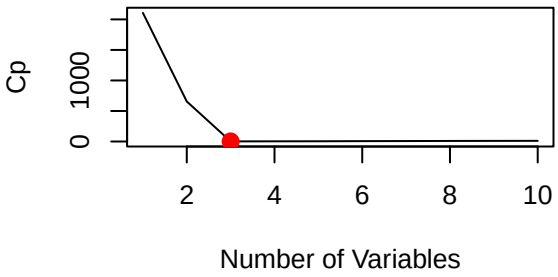


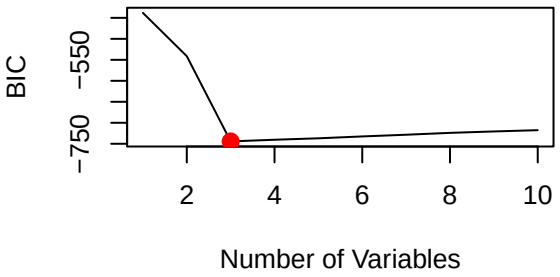
We can see that three predictors yields the lowest C_p, the lowest BIC, and the highest Adjusted R^2. The output matrix shows that these are the first three predictors, and gives the coefficients which are very close to the coefficients we specified in the above code.

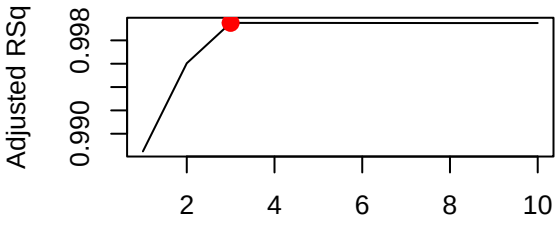
(d) Using forward stepwise selection, we have:

```
model_select(X_matrix, Y, "forward")
```

```
##          1   2   3   4   5   6   7   8   9   10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
## 3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " "
## 4  ( 1 )  "*" "*" "*" " " "*" " " " " " " " " " "
## 5  ( 1 )  "*" "*" "*" " " "*" " " " " " " "*" " "
## 6  ( 1 )  "*" "*" "*" " " "*" " " " " " " "*" "*"
## 7  ( 1 )  "*" "*" "*" "*" "*" " " " " " " "*" "*"
## 8  ( 1 )  "*" "*" "*" "*" "*" "*" " " " " "*" "*"
## 9  ( 1 )  "*" "*" "*" "*" "*" "*" " " "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
## (Intercept)           1           2           3
##   0.9621529   4.9867296  -2.9495582   9.9875115
```





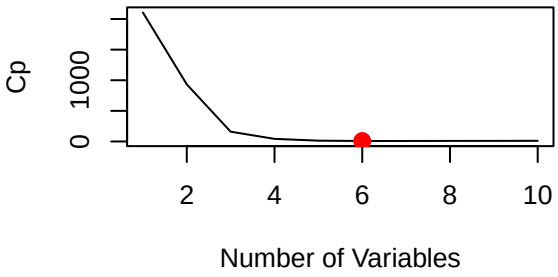


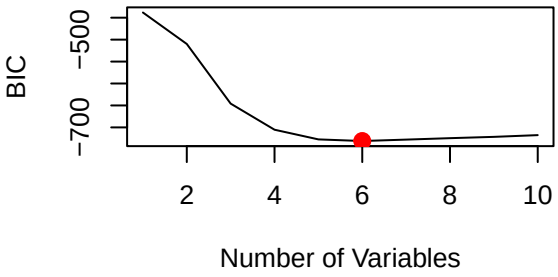
Forward Stepwise selection selects the same model as best subset selection, which is a three-variable model. Using backward stepwise selection, we have:

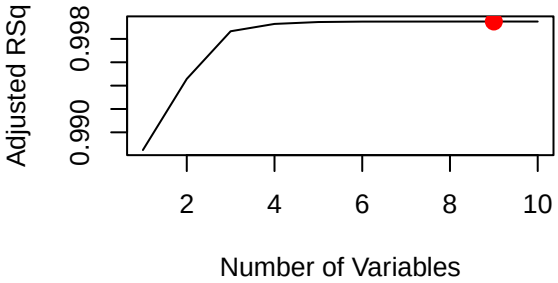
```
model_select(X_matrix, Y, "backward")
```

```
##          1   2   3   4   5   6   7   8   9   10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " " " "*" "*" " " " " " " " " " " " "
## 3  ( 1 )  "*" " " "*" "*" " " " " " " " " " " " "
## 4  ( 1 )  "*" " " "*" "*" " " "*" " " " " " " " "
## 5  ( 1 )  "*" " " "*" "*" " " "*" " " "*" " " " "
## 6  ( 1 )  "*" " " "*" "*" " " "*" " " "*" " " "*"
## 7  ( 1 )  "*" " " "*" "*" " " "*" "*" "*" " " "*"
```

```
## 8  ( 1 )  "*" " " "*" "*" " " "*" "*" "*" "*" "*"
## 9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
## (Intercept)           1           3           4           6           8
##   0.5890997   5.0354846   9.9661887  -3.8493495   1.7655796  -0.3328523
##          10
##   0.0220322
```



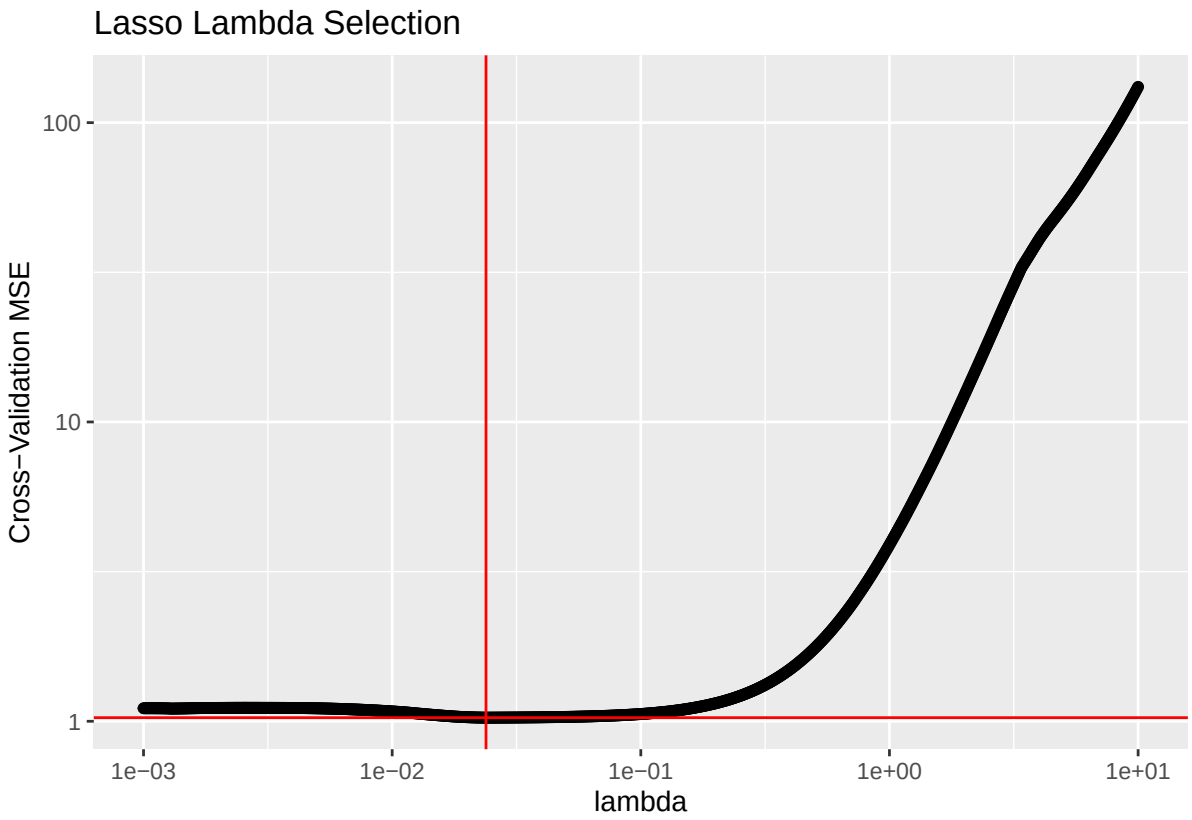




Backward stepwise selection minimizes C_p and BIC with a six-variable model, while the Adjusted R^2 is maximized with a nine-variable model. The coefficients for the six-variable model are displayed since that is the model which minimizes the BIC.

(e) Now we'll fit a lasso model with 5-fold CV with 100 values of lambda ranging from 0.01 to 1.

```
lambdas = 10^seq(1,-3, length = 1000)
model_lasso <- cv.glmnet(x = X_matrix,
                         y = Y,
                         lambda = lambdas,
                         nfolds = 5)

lambda_MSE_df <- data.frame("lambda" = model_lasso$lambda,
                            "CV_MSE" = model_lasso$cvm)

lambda_MSE_df %>%
    ggplot(aes(x = lambda, y = CV_MSE)) +
    geom_point() +
    geom_line() +
    geom_vline(xintercept = model_lasso$lambda.min, col = "red") +
    geom_hline(yintercept = min(model_lasso$cvm), col = "red") +
    scale_x_log10() +
    scale_y_log10() +
    labs(
        x = "lambda", y = "Cross-Validation MSE",
        title = "Lasso Lambda Selection"
    )
```




```
(lambda_best <- model_lasso$lambda.min)
```

```
## [1] 0.0238439
```

The best value of λ is 0.0238439 for this data set. With this value of λ, we can refit the model on the whole dataset to get more accurate coefficients, and they will be reported below:

```
model_lasso_best <- glmnet(x = X_matrix, y = Y, alpha = 1)
predict(model_lasso_best, s = lambda_best, type = "coefficients")
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.28485764
## 1            4.38500478
## 2           -2.41560241
## 3            9.98009656
## 4           -0.01330873
## 5            .
## 6            .
## 7            .
## 8            .
## 9            .
## 10           .
```

We can note that the lasso regression has shrunken the coefficients for $X^5 - X^{10}$ down to zero, performing variable selection. It has also shrunken X^4 down close to zero, but not quite all the way. The coefficients $\hat{\beta}_0 - \hat{\beta}_3$ are all relatively close to their true values, with the largest error seen in $\hat{\beta}_2$

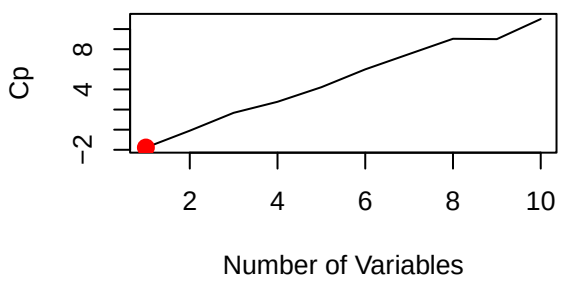
(f) We'll generate a new response Y with the same coefficient β_0 and a new coefficient β_7:

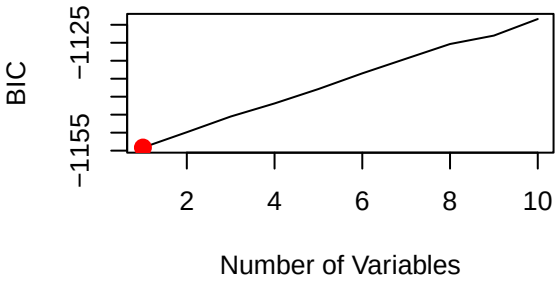
```
beta_7 <- 4
Y <- beta_0 + beta_7*X^7 + eps
```

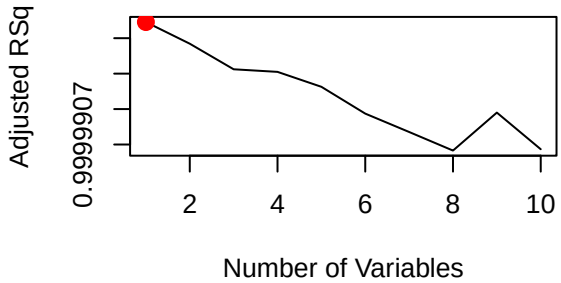
We'll now perform best subset selection using the function we created earlier:

```
model_select(X_matrix, Y, "exhaustive")
```

```
##          1   2   3   4   5   6   7   8   9   10
## 1  ( 1 )  " " " " " " " " " " " " "*" " " " " " "
## 2  ( 1 )  " " "*" " " " " " " " " " " " " " " " "
## 3  ( 1 )  " " " " " " " " " " " " "*" " " " " " "
## 4  ( 1 )  "*" " " "*" " " " " " " " " " " " " " "
## 5  ( 1 )  "*" " " "*" " " " " " " " " " " " " " "
## 6  ( 1 )  " " "*" " " "*" " " " " " " " " " " " "
## 7  ( 1 )  "*" " " " " " " " " " " " " " " " " " "
## 8  ( 1 )  " " "*" " " "*" "*" "*" "*" "*" "*" "*"
## 9  ( 1 )  " " "*" "*" "*" "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
## (Intercept)           7
##    1.025124    3.998911
```



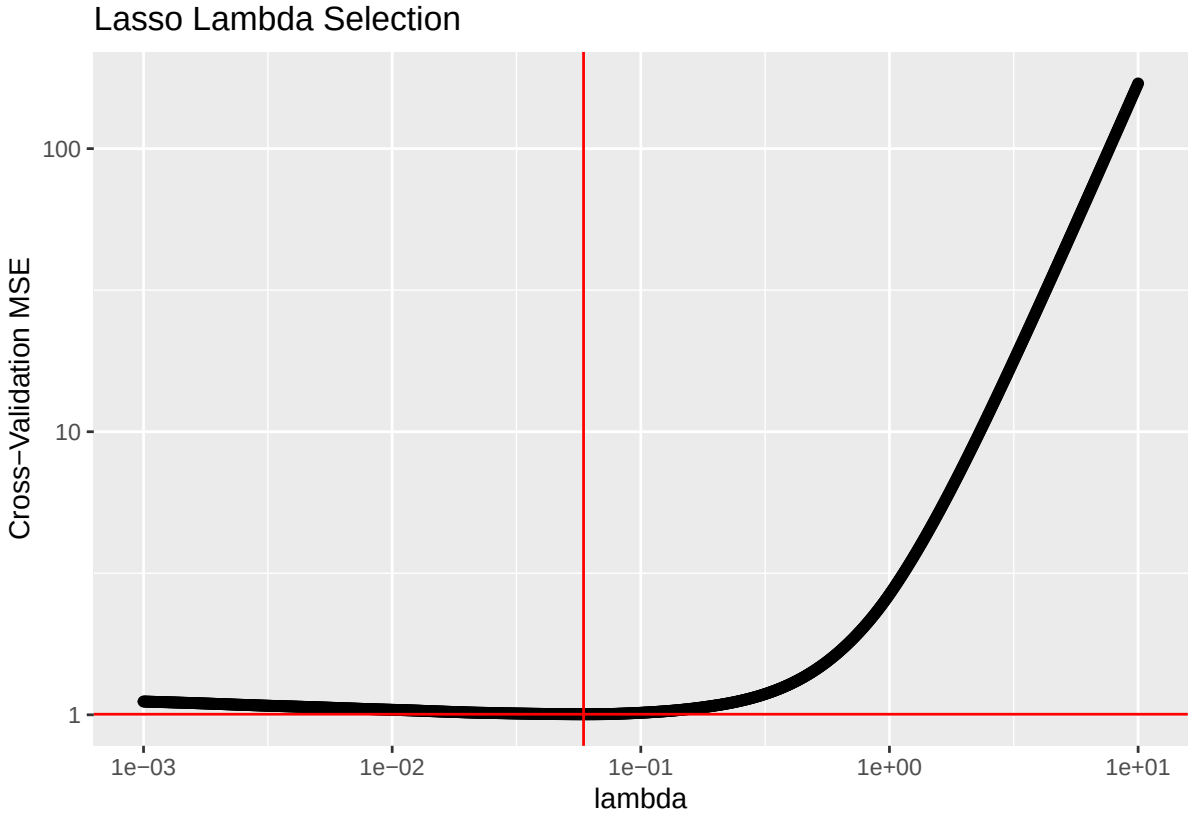




We can see that all three of the criteria choose the one-variable model, with X^7 as the single predictor. We'll now perform the lasso to see if it does the same thing:

```
model_lasso <- cv.glmnet(x = X_matrix,
                         y = Y,
                         lambda = lambdas,
                         nfolds = 5)
```

```r
lambda_MSE_df <- data.frame("lambda" = model_lasso$lambda,
                            "CV_MSE" = model_lasso$cvm)

lambda_MSE_df %>%
    ggplot(aes(x = lambda, y = CV_MSE)) +
    geom_point() +
    geom_line() +
    geom_vline(xintercept = model_lasso$lambda.min, col = "red") +
    geom_hline(yintercept = min(model_lasso$cvm), col = "red") +
    scale_x_log10() +
    scale_y_log10() +
    labs(
        x = "lambda", y = "Cross-Validation MSE",
        title = "Lasso Lambda Selection"
    )
```




```r
(lambda_best <- model_lasso$lambda.min)
```

```
## [1] 0.05885316
```

```r
model_lasso_best <- glmnet(x = X_matrix, y = Y, alpha = 1)
predict(model_lasso_best, s = lambda_best, type = "coefficients")
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept) 0.587185
```

```
## 1           .
## 2           .
## 3           .
## 4           .
## 5           .
## 6           .
## 7           3.882341
## 8           .
## 9           .
## 10          .
```

We can see that the lasso performed variable selection quite well, and only includes the values of the intercept and the X^7 term.