DSO 530 - Homework 4

Xueyan Gu

4/10/2019

ISLR Chapter 6

6.8 Exercises

1. Answer:

(a)

The model with k predictors which has the smallest training RSS is the model performed from the best subset selection because when performing best subset selection, the model with k predictors is the model with the smallest RSS among all the C_p^k models with k predictors.

On the other hand, when conducting farward stepwise selection, the model with k predictors is the model with the smallest RSS among the p-k models which increase the predictors in M_{k-1} with one additional predictor. When conducting backward stepwise selection, the model with k predictors is the model with the smallest RSS among the k models which contains all but one of the predictors in M_{k+1}.

(b)

It is hard to answer because each selection method is likely to have the smallest test RSS. Best subset selection may have the smallest test RSS because it takes into account more models than the other methods. However, the other methods might also pick a model with smaller test RSS by just luck.

(c)

i. True. Since the model with (k+1) predictors is obtained by increasing the predictors in the model with k predictors with one additional predictor.

ii. True. Since the model with k predictors is obtained by removing one predictor from the model with (k+1) predictors.

iii. False. Since there is no direct relationship between the models obtained from forward and backward selection.

iv. False. Since there is no direct relationship between the models obtained from forward and backward selection.

v. False. Since the model with (k+1) predictors is obtained by selecting among all possible models with (k+1) predictors, and so does not necessarily contain all the predictors selected for the k-variable model.

2. Answer:

(a)

iii is correct.

Reason: The lasso limits the number of predictors, so it reduces the inherent variance at the cost of an increase in bias. In other words, removing a predictor from the model is equivalent to saying that the removed feature does not have a strong relationship with the target value. So this may be a biased statement but it decreases the variance as a lower number of values need to be estimated from data.

(b)

iii is correct.

Reason: The ridge regression will produce more biased models because it shrinks predictors that don't have as a strong relationship with the target variable. So the variance will decrease at the cost of an increase in bias.

8. Answer:

(a)

```
set.seed(1)
x = rnorm(100)
eps = rnorm(100)
```

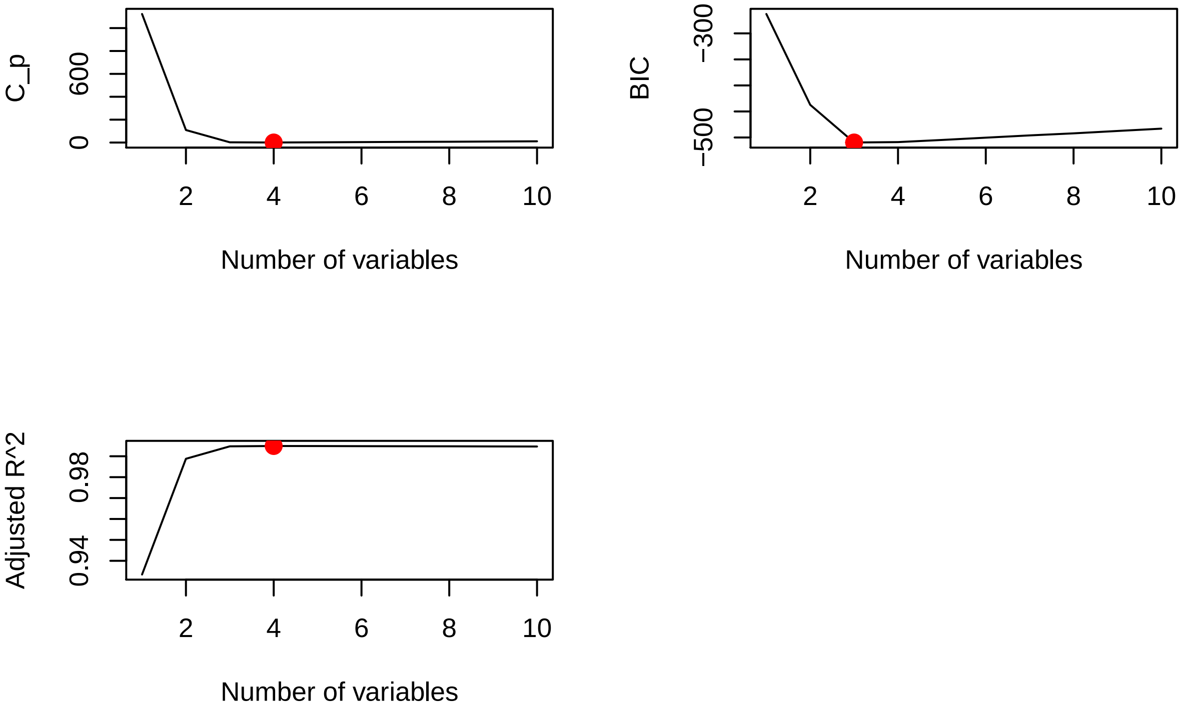
(b)

```
beta_0 = 1
beta_1 = 2
beta_2 = 3
beta_3 = 4
y = beta_0 + beta_1 * x + beta_2 * x^2 + beta_3 * x^3 + eps
```

(c)

```
library(leaps)
data = data.frame(y = y, x = x)

# Using best subset selection
regfit = regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
                      I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data, nvmax = 10)
reg_summary = summary(regfit)
```

```
# Plot the results
par(mfrow = c(2, 2))

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

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

plot(reg_summary$adjr2, xlab = "Number of variables", ylab = "Adjusted R^2", type = "l")
points(which.max(reg_summary$adjr2), reg_summary$adjr2[which.max(reg_summary$adjr2)],
       col = "red", cex = 2, pch = 20)
```



As we can see above, the best model is the one obtained according to adjusted R^2. Based on the plot, we pick the 4-variables model with C_p, we pick the 3-variables model with BIC, and we pick the 4-variables model with adjusted R^2.

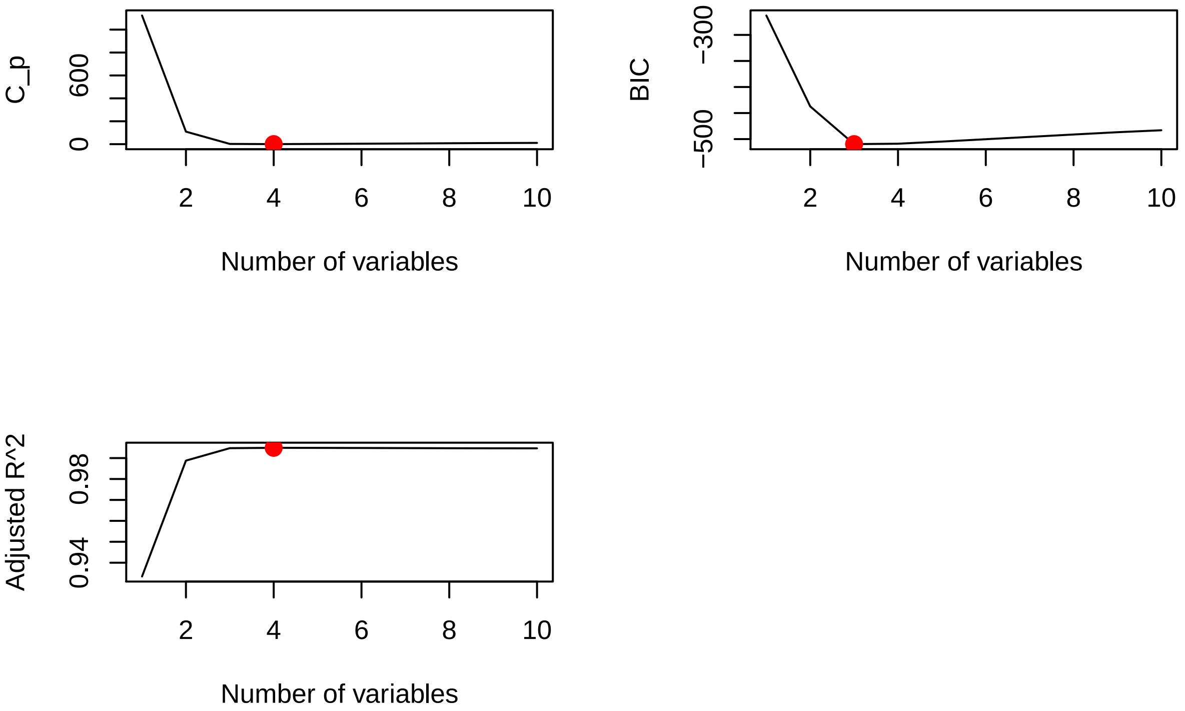
```
coef(regfit, which.max(reg_summary$adjr2))
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^5) 
##  1.07200775  2.38745596  2.84575641  3.55797426  0.08072292
```

As we can see, the coefficients of the mobel obtained according to R^2 are 2.38745596, 2.84575641, 3.55797426, and 0.08072292.

(d)

```
# Using forward stepwise selection
regfit_fwd = regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
                          I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data,
                        nvmax = 10, method = "forward")
reg_summary_fwd = summary(regfit_fwd)

# plot the results
par(mfrow = c(2, 2))

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

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

plot(reg_summary_fwd$adjr2, xlab = "Number of variables", ylab = "Adjusted R^2", type = "l")
points(which.max(reg_summary_fwd$adjr2), reg_summary_fwd$adjr2[which.max(reg_summary_fwd$adjr2)],
       col = "red", cex = 2, pch = 20)
```



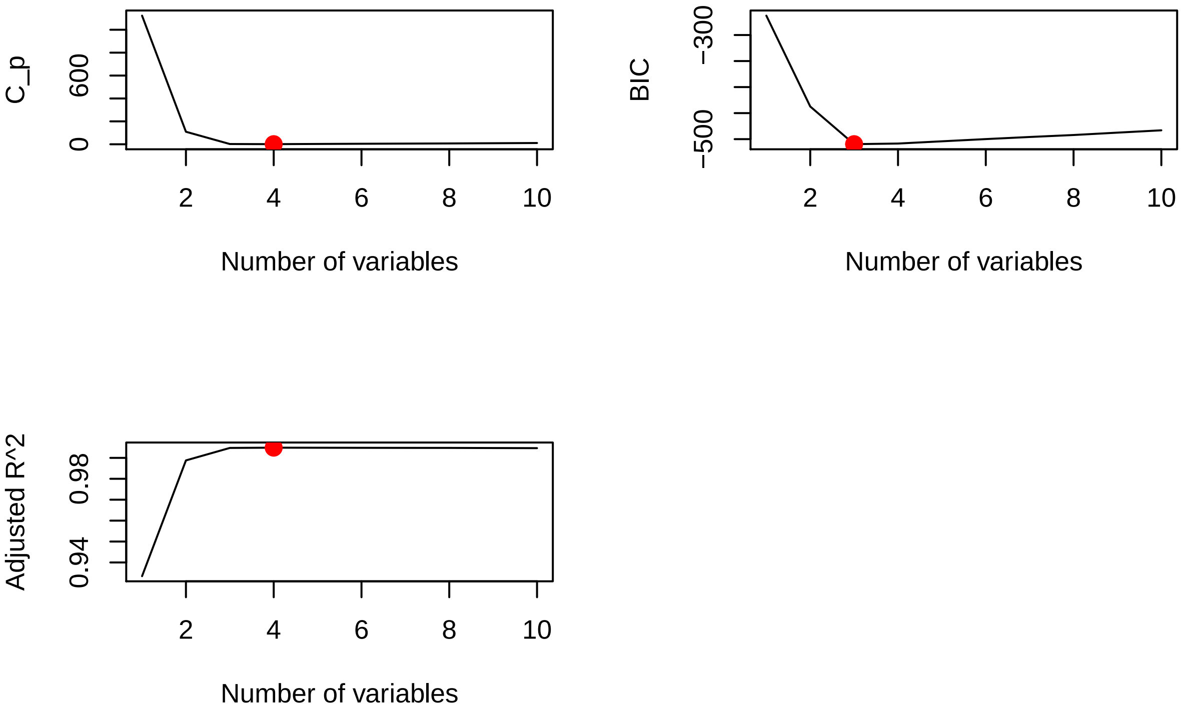
As we can see above, the best model is the one obtained according to adjusted R^2. Based on the plot, we pick the 4-variables model with C_p, we pick the 3-variables model with BIC, and we pick the 4-variables model with adjusted R^2. The result is the same as the model obtained from best subset selection.

```
coef(regfit_fwd, which.max(reg_summary_fwd$adjr2))
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^5)
##  1.07200775  2.38745596  2.84575641  3.55797426  0.08072292
```

As we can see, the coefficients of the mobel obtained according to R^2 are 2.38745596, 2.84575641, 3.55797426, and 0.08072292.

```
# Using backward stepwise selection
regfit_bwd = regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
                          I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data, nvmax = 10,
                        method = "backward")
reg_summary_bwd = summary(regfit_bwd)

# Plot the results
par(mfrow = c(2, 2))
plot(reg_summary_bwd$cp, xlab = "Number of variables", ylab = "C_p", type = "l")
points(which.min(reg_summary_bwd$cp), reg_summary_bwd$cp[which.min(reg_summary_bwd$cp)],
       col = "red", cex = 2, pch = 20)

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

plot(reg_summary_bwd$adjr2, xlab = "Number of variables", ylab = "Adjusted R^2", type = "l")
points(which.max(reg_summary_bwd$adjr2), reg_summary_bwd$adjr2[which.max(reg_summary_bwd$adjr2)],
       col = "red", cex = 2, pch = 20)
```



As we can see above, the best model is the one obtained according to adjusted R^2. Based on the plot, we pick the 4-variables model with C_p, we pick the 3-variables model with BIC, and we pick the 4-variables model with adjusted R^2.

```
coef(regfit_bwd, which.max(reg_summary_bwd$adjr2))
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^9)
## 1.079236362 2.231905828 2.833494180 3.819555807 0.001290827
```

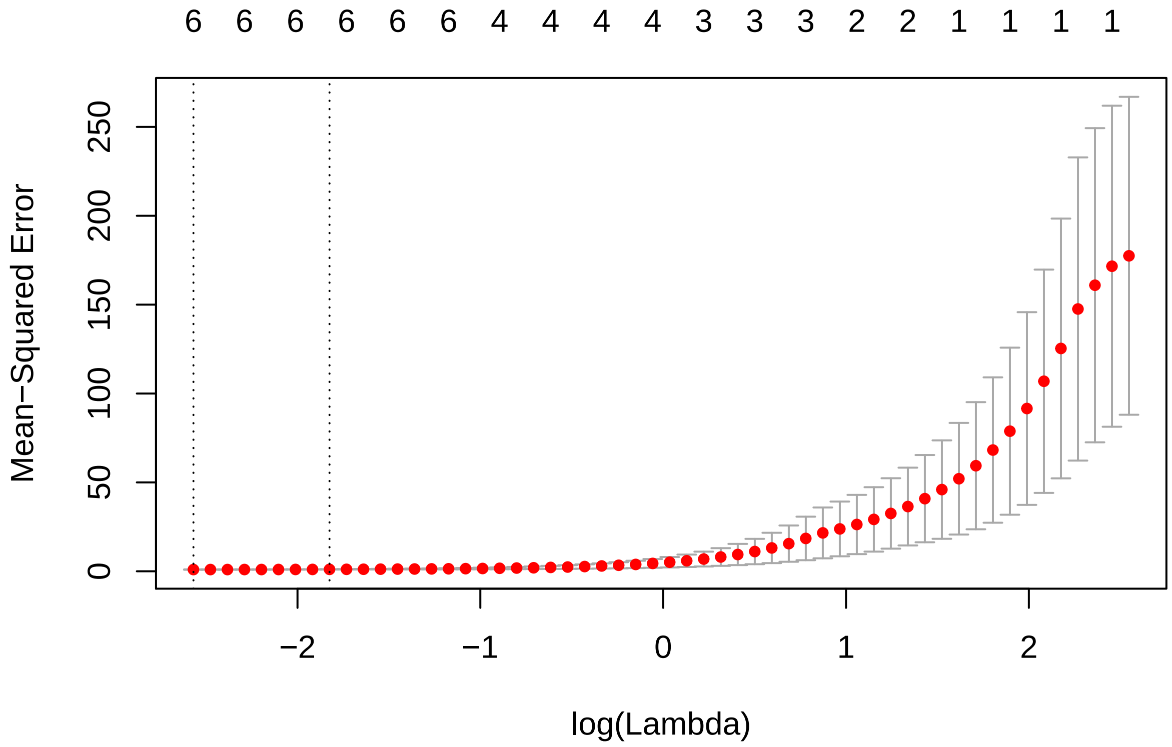
As we can see, the coefficients of the mobel obtained according to R^2 are 2.231905828, 2.833494180, 3.819555807, and 0.001290827.

(e)

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

```
x_matrix = model.matrix(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
                          I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data)[, -1]
cv_lasso <- cv.glmnet(x_matrix, y, alpha = 1)
plot(cv_lasso)
```



```
lambda = cv_lasso$lambda.min
lambda
```

```
## [1] 0.07660225
```

As we can see, the optimal value of λ is 0.07660225. We use the optimal value to refit the lasso model.

```
fit_lasso = glmnet(x_matrix, y, alpha = 1)
predict(fit_lasso, s = lambda, type = "coefficients")[1:11, ]
```

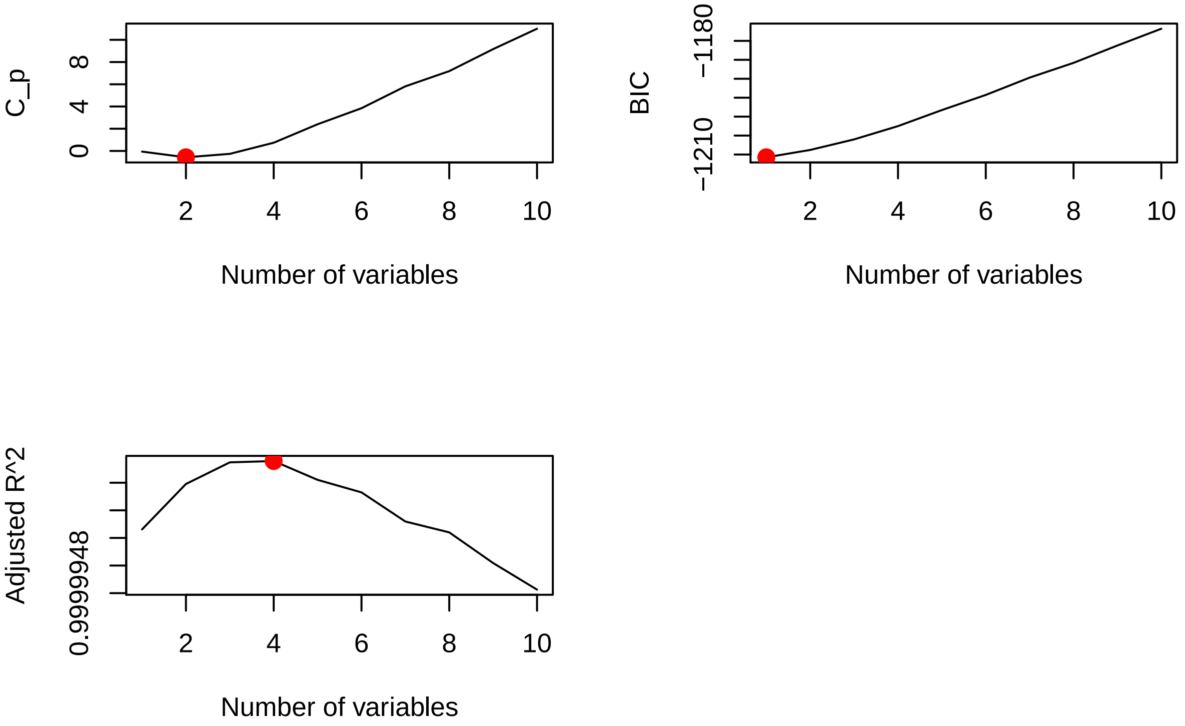
```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^4)      I(x^5) 
## 1.182646169 2.137739131 2.623547995 3.813195738 0.042303133 0.012404464 
##      I(x^6)      I(x^7)      I(x^8)      I(x^9)     I(x^10) 
## 0.000000000 0.003849104 0.000000000 0.000000000 0.000000000
```

As we can see above, the lasso method picks X, X^2, X^3, X^4 and X^5 as variables for the model.

(f)

```
beta_7 = 7
y = beta_0 + beta_7 * x^7 + eps
data2 = data.frame(y = y, x = x)

# Using best subset selection
regfit2 = regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
                       I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data2, nvmax = 10)
reg_summary2 = summary(regfit2)

# Plot the results
par(mfrow = c(2, 2))
plot(reg_summary2$cp, xlab = "Number of variables", ylab = "C_p", type = "l")
points(which.min(reg_summary2$cp), reg_summary2$cp[which.min(reg_summary2$cp)],
       col = "red", cex = 2, pch = 20)
```

```
plot(reg_summary2$bic, xlab = "Number of variables", ylab = "BIC", type = "l")
points(which.min(reg_summary2$bic), reg_summary2$bic[which.min(reg_summary2$bic)],
       col = "red", cex = 2, pch = 20)

plot(reg_summary2$adjr2, xlab = "Number of variables", ylab = "Adjusted R^2", type = "l")
points(which.max(reg_summary2$adjr2), reg_summary2$adjr2[which.max(reg_summary2$adjr2)],
       col = "red", cex = 2, pch = 20)
```



Based on the plot, we pick the 2-variables model with C_p, we pick the 1-variables model with BIC, and we pick the 4-variables model with adjusted R^2.

```
coef(regfit2, 4)
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^7)
##   1.0762524   0.2914016  -0.1617671  -0.2526527   7.0091338
```

As we can see, best subset selection with BIC picks the most accurate 1-variable model with matching coefficients.

```
# Using lasso
x_matrix2 = model.matrix(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
                             I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data2)[, -1]
cv_lasso2 = cv.glmnet(x_matrix2, y, alpha = 1)
lambda2 = cv_lasso2$lambda.min
lambda2
```

```
## [1] 13.57478
```

```
fit_lasso2 = glmnet(x_matrix2, y, alpha = 1)
predict(fit_lasso, s = lambda2, type = "coefficients")[1:11, ]
```

```
## (Intercept)           x      I(x^2)      I(x^3)      I(x^4)      I(x^5)
##    4.454162    0.000000    0.000000    0.000000    0.000000    0.000000
##      I(x^6)      I(x^7)      I(x^8)      I(x^9)     I(x^10)
##    0.000000    0.000000    0.000000    0.000000    0.000000
```

As we can see, lasso picks the most accurate 1-variable model, but the intercept is quite off.