ISL - Chapter 6 Exercises
Linear Model Selection and Regularization

An introduction to Statistical Learning, with Applications in R
- G. James, D. Witten, T. Hastie, R. Tibshirani

Thu Nguyen

26 June, 2019

Contents

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 $\beta_0, \beta_1, \beta_2, \beta_3$ 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 obtained 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.

(a) Generating random data: $X \sim \mathcal{N}(1, 2), \epsilon \sim \mathcal{N}(0, 1)$

```r
library(leaps)
set.seed(1)
x <- rnorm(100, 1, 2)
e <- rnorm(100, 0, 1)
```

(b) Generating $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$, where $\beta_0, \beta_1, \beta_2, \beta_3 \sim Unif(-5, 5)$ and are integers.

```r
betas <- round(runif(4, -5, 5), 0)
b0 <- betas[1]; b1 <- betas[2]; b2 <- betas[3]; b3 <- betas[4]
y <- b0 + b1*x + b2*x^2 + b3*x^3 + e
print(paste0('beta', 0:3, ': ', betas, collapse = '; '))
```

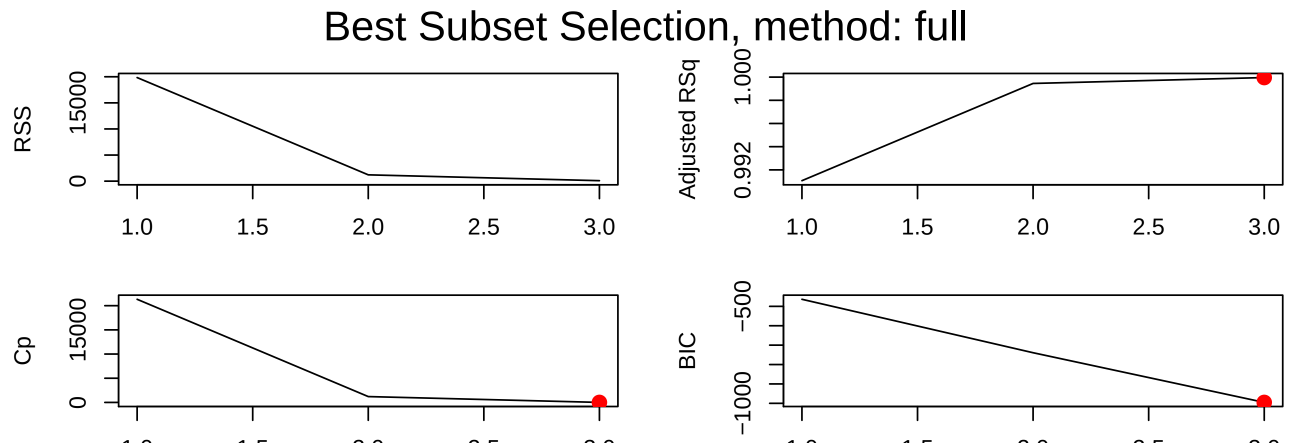
```
## [1] "beta0: 2; beta1: -3; beta2: 5; beta3: 4"
```

(c) `regsubsets()` to choose Best Subset Selection, Method: Full

```r
# create data.frame of data
dat <- data.frame(y = y, x1 = x, x2 = x^2, x3 = x^3)
# Best Subset Selection Full Model
regfit.full <- regsubsets(y ~ ., data = dat)
(reg.full.sum <- summary(regfit.full))
```

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

```r
par(mfrow=c(2,2), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.full.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.full.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.full.sum$adjr2), reg.full.sum$adjr2[which.max(reg.full.sum$adjr2)],
       col = 'red', cex = 2, pch = 20)
plot(reg.full.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.full.sum$cp), reg.full.sum$cp[which.min(reg.full.sum$cp)],
       col = 'red', cex = 2, pch = 20)
plot(reg.full.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.full.sum$bic), reg.full.sum$bic[which.min(reg.full.sum$bic)],
       col = 'red', cex = 2, pch = 20)
mtext('Best Subset Selection, method: full', outer = TRUE, cex = 1.5)
```



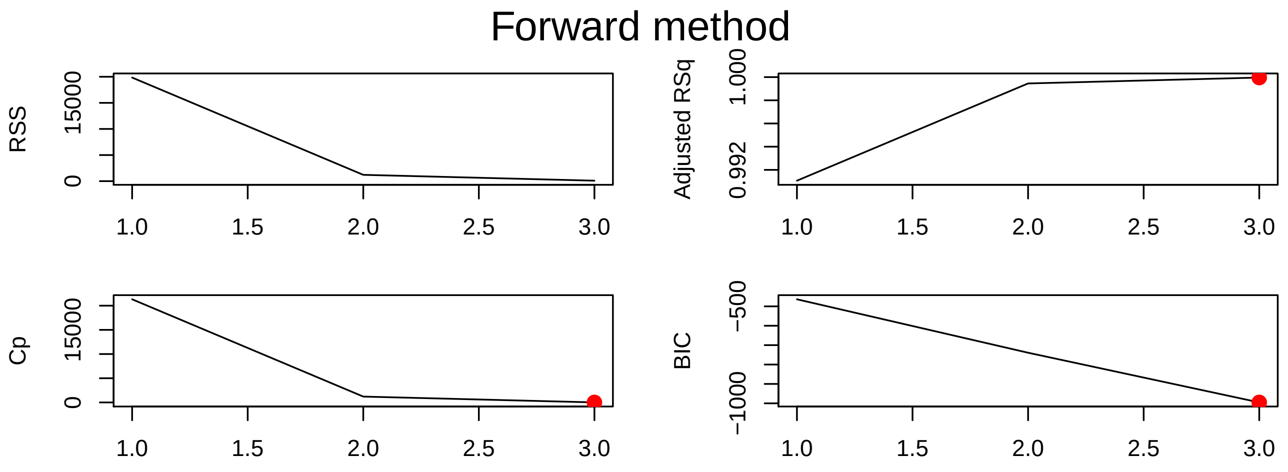
Comment: from plots, it appears that 3 variables would make a reasonable model, giving the model:

```r
coef(regfit.full, 3)
```

```
## (Intercept)          x1          x2          x3 
##    2.040714   -2.943850    4.962438    4.002205
```

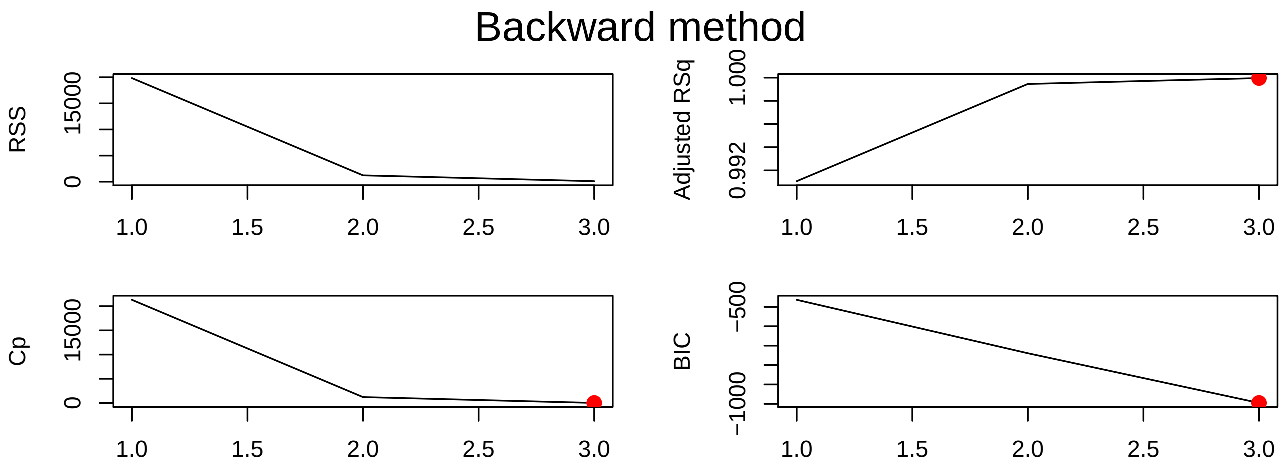
(d) Best Subset Selection, Method: Forward and Backward

```r
regfit.fwd <- regsubsets(y ~ ., data = dat, method = 'forward')
reg.fwd.sum <- summary(regfit.fwd)

par(mfrow=c(2,2), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.fwd.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.fwd.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.fwd.sum$adjr2), reg.fwd.sum$adjr2[which.max(reg.fwd.sum$adjr2)],
       col = 'red', cex = 2, pch = 20)
plot(reg.fwd.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.fwd.sum$cp), reg.fwd.sum$cp[which.min(reg.fwd.sum$cp)],
       col = 'red', cex = 2, pch = 20)
plot(reg.fwd.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.fwd.sum$bic), reg.fwd.sum$bic[which.min(reg.fwd.sum$bic)],
       col = 'red', cex = 2, pch = 20)
mtext('Forward method', outer = TRUE, cex = 1.5)
```



Comment: from plots, it appears that the *Forward* method also returns a model with 3 variables.

```r
regfit.bwd <- regsubsets(y ~ ., data = dat, method = 'backward')
reg.bwd.sum <- summary(regfit.bwd)

par(mfrow=c(2,2), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.bwd.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.bwd.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.bwd.sum$adjr2), reg.bwd.sum$adjr2[which.max(reg.bwd.sum$adjr2)],
       col = 'red', cex = 2, pch = 20)
plot(reg.bwd.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.bwd.sum$cp), reg.bwd.sum$cp[which.min(reg.bwd.sum$cp)],
       col = 'red', cex = 2, pch = 20)
plot(reg.bwd.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.bwd.sum$bic), reg.bwd.sum$bic[which.min(reg.bwd.sum$bic)],
       col = 'red', cex = 2, pch = 20)
mtext('Backward method', outer = TRUE, cex = 1.5)
```



Comment: from plots, it appears that the *Backward* method also returns a model with 3 variables, similar to the full model and *Forward* method.

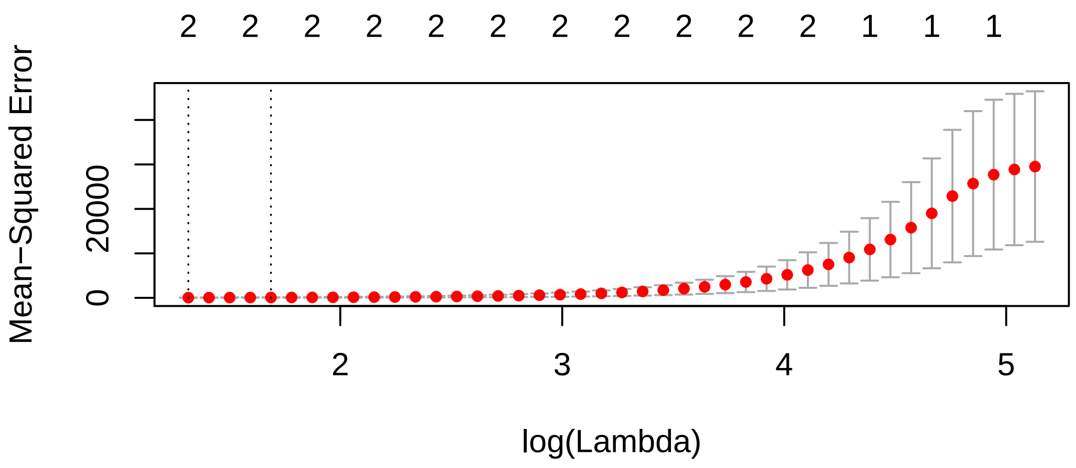
```r
data.frame(Betas = betas,
           Full = round(coef(regfit.full, 3),4),
           Forward = round(coef(regfit.fwd, 3),4),
           Backward = round(coef(regfit.bwd, 3),4))
```

```
##               Betas    Full Forward Backward
## (Intercept)       2  2.0407  2.0407   2.0407
## x1               -3 -2.9438 -2.9438  -2.9438
## x2                5  4.9624  4.9624   4.9624
## x3                4  4.0022  4.0022   4.0022
```

Comment: All 3 different methods yield the same model with 3 variables, whose coefficients are the same.

(e) The Lasso

```r
library(glmnet)
grid <- 10^seq(10, -2, length = 100)
# Convert data.frame() to matrix
x <- model.matrix(y ~ ., dat)[,-1]
y <- dat$y
# Sample split
set.seed(1)
train <- sample(100, 50, replace = FALSE)
y.test <- y[-train]     # observed data to be tested against for test-set MSE
# Model
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)

# Cross-Validation: cv.glmnet()
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
plot(cv.out)
```



```r
bestlam <- cv.out$lambda.min
# predict() and test set MSE based on best lambda
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[-train,])
mse <- mean((lasso.pred - y.test)^2)
```

Under the Lasso and Cross-Validation, the best λ (giving the $\min(MSE)$) is 3.7268791, and the associated MSE on the test set is 13.1452921.

```r
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.fit <- predict(out, type = 'coefficients', s = bestlam)
data.frame(Betas = betas,
           Subset_Selection = round(coef(regfit.full, 3),4),
           Lass0 = round(c(lasso.fit[1], lasso.fit[2], lasso.fit[3], lasso.fit[4]),4))
```

```
##             Betas Subset_Selection  Lass0
## (Intercept)     2           2.0407 2.2141
## x1             -3          -2.9438 0.0000
## x2              5           4.9624 4.7428
## x3              4           4.0022 3.7932
```

Comment:

- whereas the *Lasso* returns a model with only X^2 and X^3, while *Best Subset Selection* returns a model with all 3 variables X, X^2, X^3.

■

Exercise 9

In this exercise, we will predict the number of applications received using the other variables in the `College` data set.

(a) Split the data set into a training set and a test set.
(b) Fit a linear model using least squares on the training set, and report the test error obtained.
(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.
(d) Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.
(e) Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.
(f) Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.
(g) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?

(a) Splitting `College` into train and test sets

```r
library(ISLR)
data(College)
set.seed(1)
idx <- sample(nrow(College), round(nrow(College)*.8,0), replace = FALSE) # 80% train, 20% test
train <- College[idx,]
test <- College[-idx,]
```

(b) Linear Regression model

```r
lin.reg <- lm(Apps ~ ., data = train)
lin.pred <- predict(lin.reg, newdata = test)
lin.mse <- round(mean((lin.pred - test$Apps)^2),0)
print(paste0('Linear Regression MSE on test dataset: ', lin.mse))
```

```
## [1] "Linear Regression MSE on test dataset: 1082005"
```

(c) Ridge Regreesion model with best λ by Cross-Validation

```r
library(glmnet)
x <- model.matrix(Apps ~ ., College)[,-1]
y <- College$Apps
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x[idx,], y[idx], alpha = 0, lambda = grid)
# Cross-Validation
set.seed(1)
cv.out <- cv.glmnet(x[idx,], y[idx], alpha = 0)
ridge.bestlam <- round(cv.out$lambda.min,2)
# predict() and test set MSE based on best lambda
ridge.pred <- predict(ridge.mod, s = ridge.bestlam, newx = x[-idx,])
ridge.mse <- round(mean((ridge.pred - y[-idx])^2),0)
```

Under *Ridge Regression* and Cross-Validation, the best $\lambda = 382.08$, and the test set $MSE = 1188614$.

(d) The Lasso

```
lasso.mod <- glmnet(x[idx,], y[idx], alpha = 1, lambda = grid)
# Cross-Validation
set.seed(1)
cv.out <- cv.glmnet(x[idx,], y[idx], alpha = 1)
lasso.bestlam <- round(cv.out$lambda.min,2)
# predict() and test set MSE based on best lambda
lasso.pred <- predict(lasso.mod, s = lasso.bestlam, newx = x[-idx,])
lasso.mse <- round(mean((lasso.pred - y[-idx])^2),0)
```

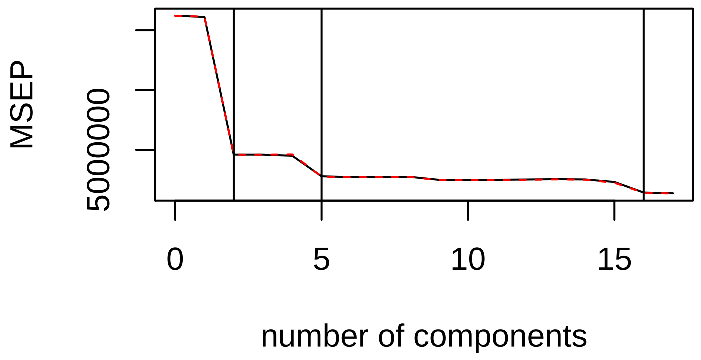
Under *Lasso* and Cross-Validation, the best $\lambda = 2.04$, and the test set $MSE = 1083743$.

```
lasso.nonzero.coef <- sum(predict(lasso.mod, type = 'coefficients', s = lasso.bestlam) != 0)
print(paste('Number of non-zero coefficient estimates from the Lasso:', lasso.nonzero.coef))
```

```
## [1] "Number of non-zero coefficient estimates from the Lasso: 18"
```

(e) Principal Components Regression

```
library(pls)
par(mar=c(4,5,1,1))
set.seed(1)
pcr.fit <- pcr(Apps ~ ., data = College, subset = idx, scale = TRUE, validation = 'CV')
validationplot(pcr.fit, val.type = 'MSEP', main = '')
abline(v = c(2,5,16))
```



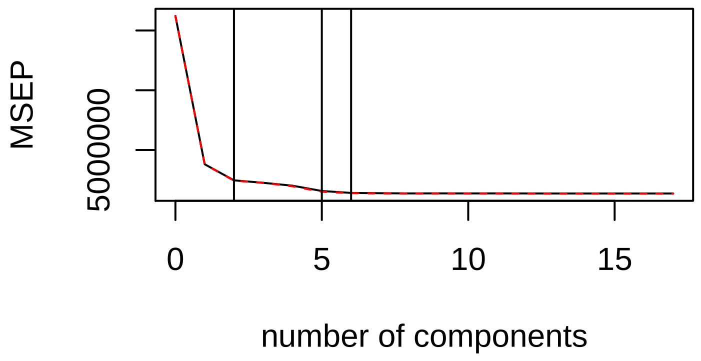
Comment: According to the plot, *Pincipal Components Regression* $M = 16$ components would yield the smallest $MSEP$ and thus MSE, although $M = 16$ is not much different from the *Linear Regression* model. At the same time, while not yielding the smallest $MSEP$, $M = 2$ or 5 appear to be reasonable estimates since they significantly reduce $MSEP$ from previous M.

```r
pcr.pred <- predict(pcr.fit, x[-idx,], ncomp = 16)
pcr.mse <- round(mean((pcr.pred - y[-idx])^2),0)
print(paste('Pricipal Components Regression: M = 16, and MSE on test dataset: ', pcr.mse))
```

```
## [1] "Pricipal Components Regression: M = 16, and MSE on test dataset:  1118477"
```

(f) Partial Least Squares Regreesion

```r
par(mar=c(4,5,1,1))
set.seed(1)
pls.fit <- plsr(Apps ~ ., data = College, subset = idx, scale = TRUE, validation = 'CV')
validationplot(pls.fit, val.type = 'MSEP', main = '')
abline(v = c(2,5,6))
```



Comment: According to the plot, *Partial Least Squares Regression* $M = 6$ components would yield the smallest $MSEP$ and thus MSE, which is a significant reduction from the original 18 variables.

```r
pls.pred <- predict(pls.fit, x[-idx,], ncomp = 6)
pls.mse <- round(mean((pls.pred - y[-idx])^2),0)
print(paste('Partial Least Squares Regression: M = 6, and MSE on test dataset: ', pls.mse))
```

```
## [1] "Partial Least Squares Regression: M = 6, and MSE on test dataset:  1118053"
```

(g) Comparison of all models

```r
models <- c('Linear Regression', 'Ridge Regression', 'Lasso', 'PCR', 'PLS')
mses <- c(lin.mse, ridge.mse, lasso.mse, pcr.mse, pls.mse)
data.frame(Models = models, MSE = mses)
```

```
##              Models     MSE
## 1 Linear Regression 1082005
## 2  Ridge Regression 1188614
## 3             Lasso 1083743
## 4               PCR 1118477
## 5               PLS 1118053
```

Comment:

- Among all the models, there are no significant differences in MSE, in part because of the random splits and cross-validation which determines the best parameter λ, all of which can change under a different iteration.

■

Exercise 11

We will now try to predict per capita crime rate in the `Boston` data set.

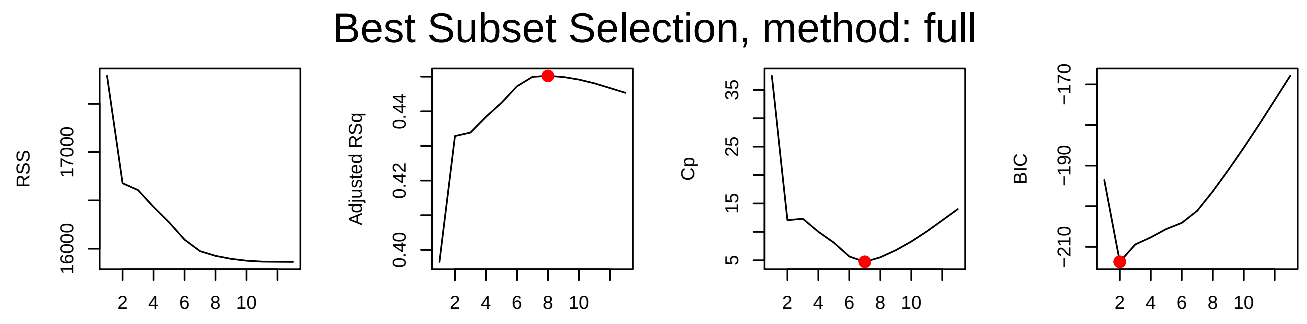
(a) Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider.
(b) Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross validation, or some other reasonable alternative, as opposed to using training error.
(c) Does your chosen model involve all of the features in the data set? Why or why not?

```r
library(MASS)
data(Boston)
set.seed(1)
idx <- sample(nrow(Boston), round(nrow(Boston)*.8,0), replace = FALSE)
train <- Boston[idx,]
test <- Boston[-idx,]
x <- model.matrix(crim ~ ., Boston)[,-1]
y <- Boston$crim
```

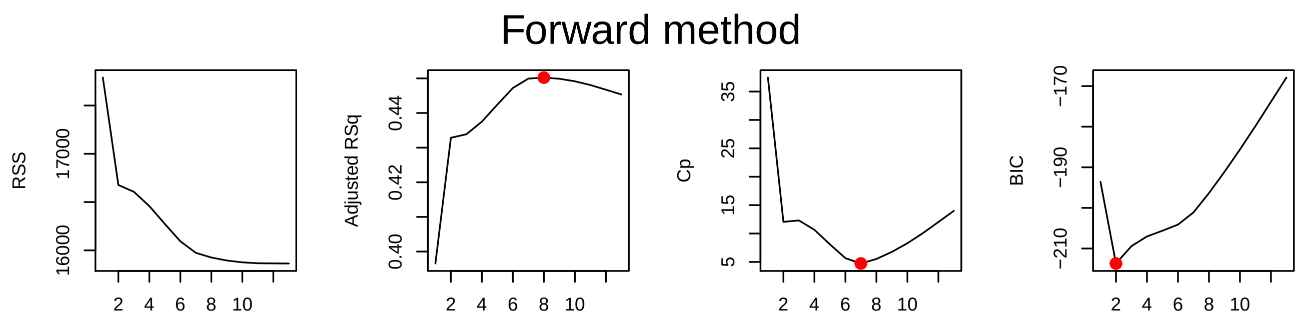
(a) Fitting models

```r
# Linear Regression
lin.reg <- lm(crim ~ ., data = train)
lin.pred <- predict(lin.reg, newdata = test)
lin.mse <- round(mean((lin.pred - test$crim)^2),2)
```

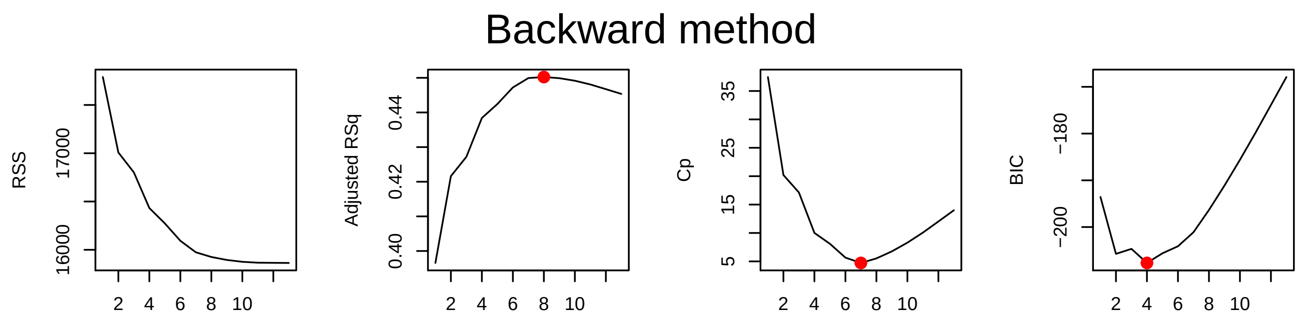
```r
# Best Subset Selection: method: Full
regfit.full <- regsubsets(crim ~ ., data = train, nvmax = ncol(Boston))
reg.full.sum <- summary(regfit.full)
par(mfrow=c(1,4), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.full.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.full.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.full.sum$adjr2), reg.full.sum$adjr2[which.max(reg.full.sum$adjr2)],
col = 'red', cex = 2, pch = 20)
plot(reg.full.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.full.sum$cp), reg.full.sum$cp[which.min(reg.full.sum$cp)],
col = 'red', cex = 2, pch = 20)
plot(reg.full.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.full.sum$bic), reg.full.sum$bic[which.min(reg.full.sum$bic)],
col = 'red', cex = 2, pch = 20)
mtext('Best Subset Selection, method: full', outer = TRUE, cex = 1.5)
```



```r
# Best Subset Selection: method: Forward
regfit.fwd <- regsubsets(crim ~ ., data = train, method = 'forward', nvmax = ncol(Boston))
reg.fwd.sum <- summary(regfit.fwd)
par(mfrow=c(1,4), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.fwd.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.fwd.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.fwd.sum$adjr2), reg.fwd.sum$adjr2[which.max(reg.fwd.sum$adjr2)],
col = 'red', cex = 2, pch = 20)
plot(reg.fwd.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.fwd.sum$cp), reg.fwd.sum$cp[which.min(reg.fwd.sum$cp)],
col = 'red', cex = 2, pch = 20)
plot(reg.fwd.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.fwd.sum$bic), reg.fwd.sum$bic[which.min(reg.fwd.sum$bic)],
col = 'red', cex = 2, pch = 20)
mtext('Forward method', outer = TRUE, cex = 1.5)
```



```r
# Best Subset Selection: method: Backward
regfit.bwd <- regsubsets(crim ~ ., data = train, method = 'backward', nvmax = ncol(Boston))
reg.bwd.sum <- summary(regfit.bwd)
par(mfrow=c(1,4), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.bwd.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.bwd.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.bwd.sum$adjr2), reg.bwd.sum$adjr2[which.max(reg.bwd.sum$adjr2)],
col = 'red', cex = 2, pch = 20)
plot(reg.bwd.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.bwd.sum$cp), reg.bwd.sum$cp[which.min(reg.bwd.sum$cp)],
col = 'red', cex = 2, pch = 20)
plot(reg.bwd.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.bwd.sum$bic), reg.bwd.sum$bic[which.min(reg.bwd.sum$bic)],
col = 'red', cex = 2, pch = 20)
mtext('Backward method', outer = TRUE, cex = 1.5)
```



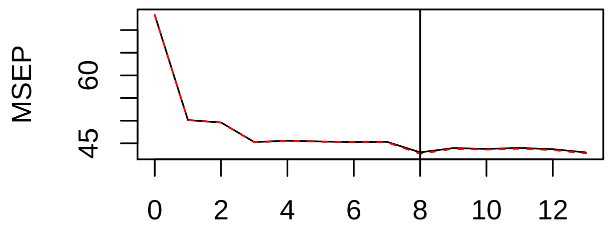
From the plots, it appears that all the Full, Forward, and Backward methods for *Best Subset Selection* would select 7 or 8 variables, and they all select the same variables. As such, it is sufficient to pick the Full method model and look at the MSE.

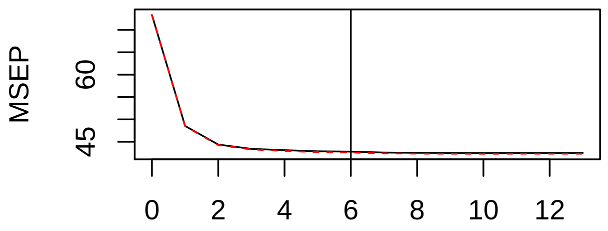
```r
# Best Subset Selection
test.mat <- model.matrix(crim ~ ., data = Boston[-idx,])
coef.7 <- coef(regfit.full, id = 7)
bss.pred.7 <- test.mat[, names(coef.7)] %*% coef.7
bss.mse.7 <- round(mean(bss.pred.7 - y[-idx])^2,2)
# Backward
coef.8 <- coef(regfit.full, id = 8)
bss.pred.8 <- test.mat[, names(coef.8)] %*% coef.8
bss.mse.8 <- round(mean(bss.pred.8 - y[-idx])^2,2)
```

```r
# Ridge Regression
library(glmnet)
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x[idx,], y[idx], alpha = 0, lambda = grid)
# Cross-Validation
set.seed(1)
cv.out <- cv.glmnet(x[idx,], y[idx], alpha = 0)
ridge.bestlam <- round(cv.out$lambda.min,2)
# predict() and test set MSE based on best lambda
ridge.pred <- predict(ridge.mod, s = ridge.bestlam, newx = x[-idx,])
ridge.mse <- round(mean((ridge.pred - y[-idx])^2),2)
```

```r
# The Lasso
lasso.mod <- glmnet(x[idx,], y[idx], alpha = 1, lambda = grid)
# Cross-Validation
set.seed(1)
cv.out <- cv.glmnet(x[idx,], y[idx], alpha = 1)
lasso.bestlam <- round(cv.out$lambda.min,2)
# predict() and test set MSE based on best lambda
lasso.pred <- predict(lasso.mod, s = lasso.bestlam, newx = x[-idx,])
lasso.mse <- round(mean((lasso.pred - y[-idx])^2),0)
```

```r
# Principal Components Regression
library(pls)
par(mfrow=c(1,2)); par(mar=c(2,5,1,1))
set.seed(1)
pcr.fit <- pcr(crim ~ ., data = Boston, subset = idx, scale = TRUE, validation = 'CV')
validationplot(pcr.fit, val.type = 'MSEP', main = '')
pcr.m <- 8
abline(v = pcr.m)
# Partial Least Squares Regression
pls.fit <- plsr(crim ~ ., data = Boston, subset = idx, scale = TRUE, validation = 'CV')
validationplot(pls.fit, val.type = 'MSEP', main = '')
pls.m <- 6
abline(v = pls.m)
```





```
# Principal Components Regression
pcr.pred <- predict(pcr.fit, x[-idx,], ncomp = pcr.m)
pcr.mse <- round(mean((pcr.pred - y[-idx])^2),2)
# Partial Least Squares Regression
pls.pred <- predict(pls.fit, x[-idx,], ncomp = pls.m)
pls.mse <- round(mean((pls.pred - y[-idx])^2),2)
```

Comparison of all models

```
models <- c('Linear Regression', 'Best Subset Selection: 7 variables',
            'Best Subset Selection: 8 variables', 'Ridge Regression', 'Lasso', 'PCR', 'PLS')
mses <- c(lin.mse, bss.mse.7, bss.mse.8, ridge.mse, lasso.mse, pcr.mse, pls.mse)
data.frame(Models = models, MSE = mses)
```

```
##                                Models   MSE
## 1                   Linear Regression 46.61
## 2 Best Subset Selection: 7 variables  0.29
## 3 Best Subset Selection: 8 variables  0.35
## 4                    Ridge Regression 46.99
## 5                               Lasso 47.00
## 6                                 PCR 48.50
## 7                                 PLS 46.73
```

Comment:

- *Best Subset Selection* models return significantly better test set MSE, even reducing to effectively 0.
- Among the other 5 models, MSE are effectively the same, indicating that given the `Boston` dataset, in predicting the per capita crime rate, any of *Ridge Regression, Lasso, PCR, PLS* would return a similar model.

(b) Model proposal

From part (a), it appears that *Best Subset Selection* returns substantially better models, as such, we can take a closer look into how well it performs, through all 3 methods and cross-validation.

```
predict.regsubsets <- function(object, newdata, id, ...) {
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[,xvars]%*%coefi
}
set.seed(1)
k <- 10
```

```r
folds <- sample(1:k, nrow(Boston), replace = TRUE)

# 10-fold Cross-Validation
p <- ncol(Boston) - 1
methods = c('exhaustive', 'forward', 'backward')
bss.summary <- as.data.frame(matrix(ncol = 2))
# Loop 1: m for each method
for (m in 1:3) {
  cv.errors <- matrix(NA, k, p, dimnames = list(NULL, paste(1:p)))
  # Loop 2: j for each of 10-fold cv
  for (j in 1:k) {
    best.fit <- regsubsets(crim ~ ., data = Boston[folds != j,], nvmax = p, method = methods[m])
    # Loop 3: i for each variable
    for (i in 1:p) {
      pred <- predict(best.fit, Boston[folds == j,], id = i)
      cv.errors[j,i] <- mean( (Boston$crim[folds == j] - pred)^2 )
    }
  }
  mean.cv.errors <- round(apply(cv.errors, 2, mean),2)
  bss.summary[m,] <- c(which.min(mean.cv.errors)[[1]], min(mean.cv.errors))
}

colnames(bss.summary) <- c('Number of Variables', 'Average 10-fold CV MSE')
rownames(bss.summary) <- c('Full', 'Forward', 'Backward')
bss.summary
```

```
##          Number of Variables Average 10-fold CV MSE
## Full                      12                  41.03
## Forward                   12                  41.03
## Backward                  12                  41.03
```

Comment:

- while the average k-fold Cross Validation MSE are slightly different across different methods, all 3 methods agree on the same number of variables that would return the lowest MSE, which is 12 variables. Additionally, while the average MSE is different from the Validation Set approach from part (a), *Best Subset Selection* still produce significantly smaller MSE and are thus better models.

```r
regfit.full <- regsubsets(crim ~ ., data = Boston, nvmax = p)
round(coef(regfit.full, 12),4)
```

```
## (Intercept)          zn       indus        chas         nox          rm
##     16.9857      0.0447     -0.0638     -0.7444    -10.2022      0.4396
##         dis         rad         tax     ptratio       black       lstat
##     -0.9936      0.5877     -0.0038     -0.2699     -0.0075      0.1281
##        medv
##     -0.1989
```

■