STA6703 SML HW6

Christopher Marais

Chapter 6

Question 1

a.)

The best subset method has lowest RSS. This is because the best subset method is optimized fully to the training data taking into account all possible combinations of predictors. The step wise methods do not take into consideration all possible combination of predictors and would not be fully optimized on the training data. The larger k the more precise the fit for the best subset method will be.

b.)

There is no way of telling which of the methods would perform best on the testing data. This is totally dependent on how well the training data represents the testing data. if the testing data is represented well by the training data then the best subset method might perform better as it results in a more precise fit on the training data (sometimes resulting in over fitting), however if the testing data is not represented very well by the training data then the step wise methods might perform better as they generalize better than the best subset method.

c.)

i.) True
ii.) True
iii.) False
iv.) False
v.) False

Question 2

a.)

iii. Some coefficients to less informative predictors are 0 and thus decreasing their influence on prediction which in turn could increase the bias.

b.)

iii. Some coefficients to less informative predictors are very small and thus decreasing their influence on prediction allowing for a more general fit to the data which in turn could increase the bias.

c.)

ii. The model will be more flexible allowing for a more precise fit to the data which in turn could increase the variance.

Question 4

a.)

iii. As lambda increases the effect of beta will decrease. This in turn will cause the training RSS to steadily increase.

b.)

ii. Increasing lambda will decrease the variance and increase the bias. The decrease in variance is initially much faster than the increase in bias and thus the test RSS will decrease. However, the rate of decrease in variance decreases and the rate of increase in bias increases as lambda increases. Therefore at some point the increase in bias will outweigh the decrease in variance and the test RSS will start increasing again.

c.)

iv. As lambda increases the effect of beta decreases which in turn allows a decrease in variance as the model becomes more flexible.

d.)

iii. As lambda increases the effect of beta decreases which in turn allows an increase in bias as the model becomes more flexible.

e.)

v. The irreducible error is the error that is not possible to remove with a model and is inherent to the data. Therefore this stays the same.

Question 8

a.)

```
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

```r
# Generate data
set.seed(0)
X = c(rnorm(100))
e = rnorm(100, mean=0, sd=0.25)
```

b.)

```r
# Specify variables
b_0 = 5
b_1 = 7
b_2 = 9
b_3 = 11

Y = c(b_0 + b_1*X + b_2*X^2 + b_3*X^3 + e)
```

c.)

```r
# Best subset
# Save data in dataframe
df = data.frame(X,X^2,X^3,X^4,X^5,X^6,X^7,X^8,X^9,X^10,Y)

# Find best subsets
fit <- regsubsets(Y ~ ., data = df, nvmax=10)
summary(fit)
```

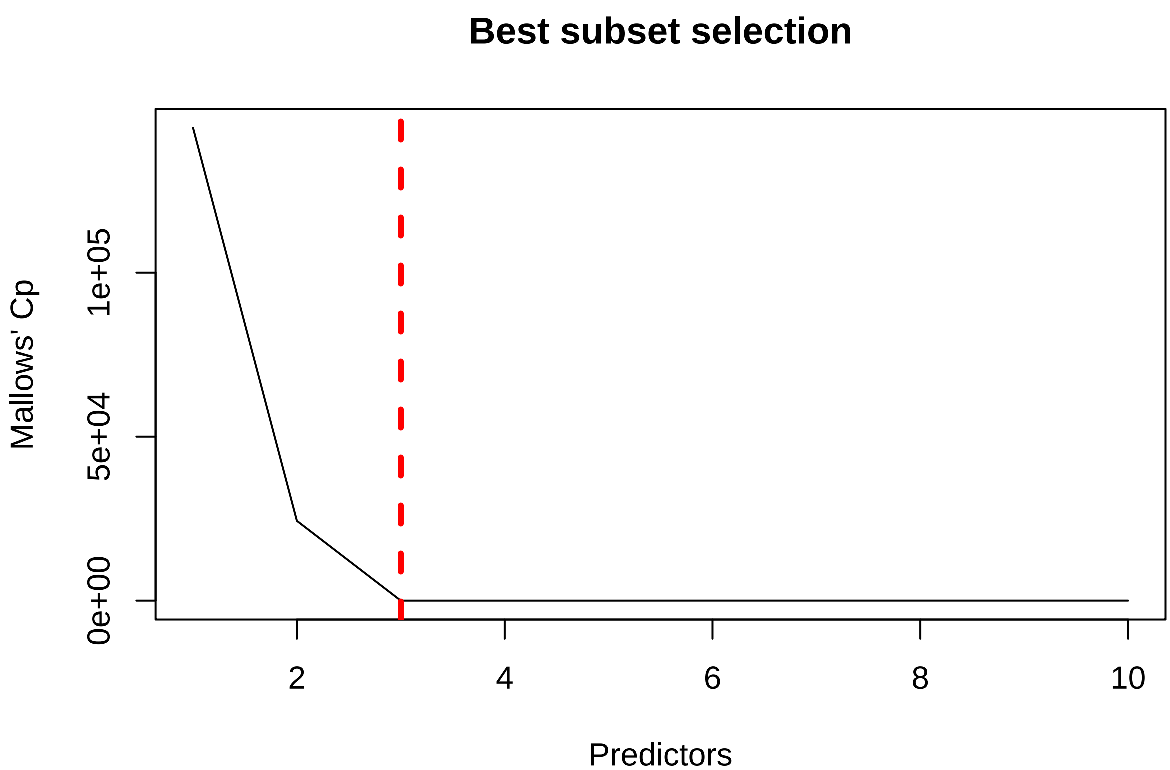
```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = df, nvmax = 10)
## 10 Variables  (and intercept)
##       Forced in Forced out
## X         FALSE      FALSE
## X.2       FALSE      FALSE
## X.3       FALSE      FALSE
## X.4       FALSE      FALSE
## X.5       FALSE      FALSE
## X.6       FALSE      FALSE
## X.7       FALSE      FALSE
## X.8       FALSE      FALSE
## X.9       FALSE      FALSE
## X.10      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##            X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
## 3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " "
## 4  ( 1 )  "*" "*" "*" "*" " " " " " " " " " " " "
## 5  ( 1 )  "*" "*" "*" " " " " " " " " "*" " " "*"
## 6  ( 1 )  "*" "*" "*" "*" " " " " " " "*" " " "*"
```

```
## 7  ( 1 )  "*" "*" "*" " " "*" "*" " " "*" " " "*"
## 8  ( 1 )  "*" "*" "*" " " "*" "*" "*" "*" " " "*"
## 9  ( 1 )  "*" "*" "*" " " "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
# Extract cp, bic and adjr2
cp = summary(fit)$cp
bic = summary(fit)$bic
adjusted_r_sq = summary(fit)$adjr2

# print coefficients
coef(fit, id=which.min(cp))
```

```
## (Intercept)           X         X.2         X.3
##    4.991504    7.007547    8.992661   11.012597
```

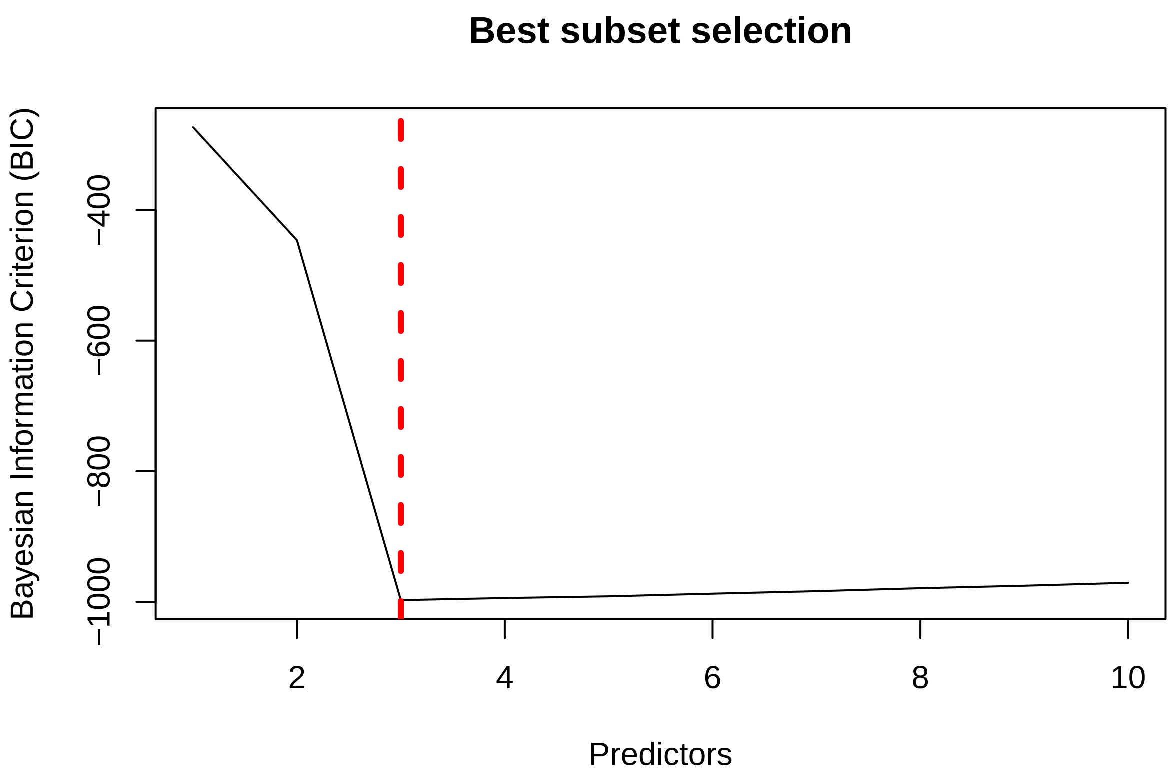
```
# Plot Cp
{plot(cp,
      main="Best subset selection",
     xlab="Predictors",
     ylab="Mallows' Cp",
     type="l")
  abline(v=which.min(cp),
         col="red",
         lwd=3,
         lty=2)}
```



```r
print(paste("The Cp decreases and then plateaus at the lowest value of",
            which.min(cp),
            "predictors."))
```

```
## [1] "The Cp decreases and then plateaus at the lowest value of 3 predictors."
```

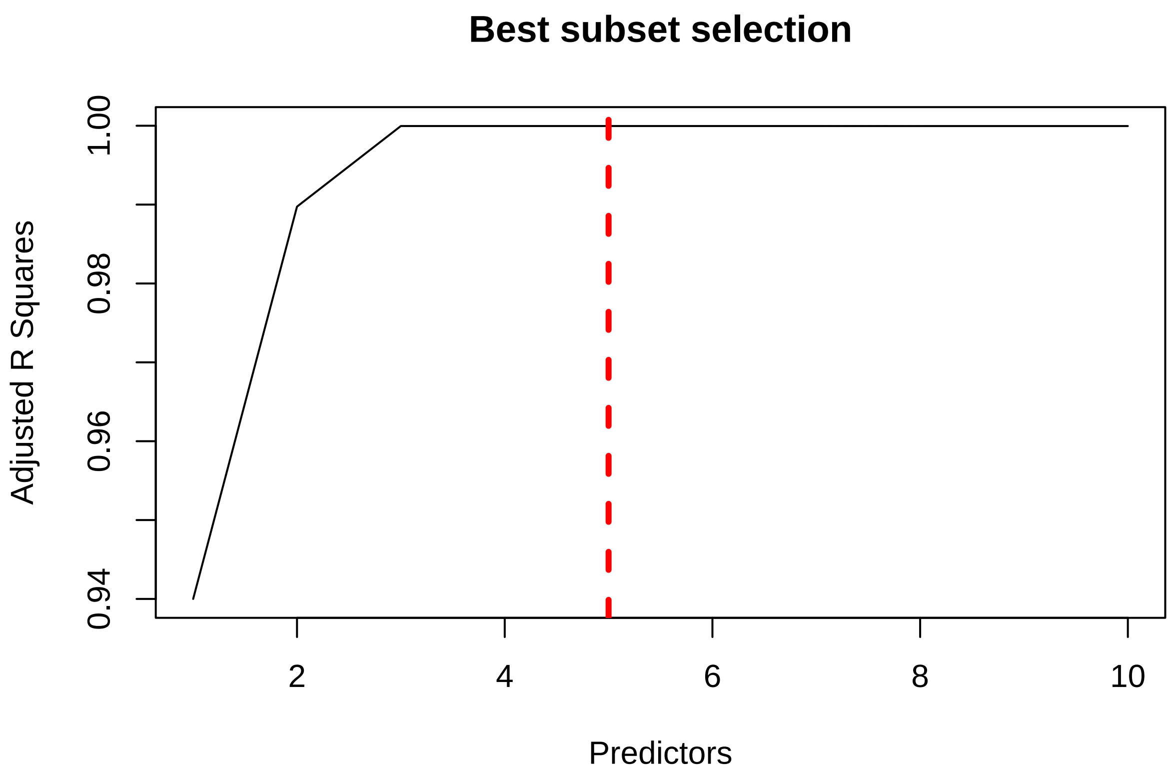
```r
# Plot BIC
{plot(bic,
     main="Best subset selection",
     xlab="Predictors",
     ylab="Bayesian Information Criterion (BIC)",
     type="l")
  abline(v=which.min(bic),
         col="red",
         lwd=3,
         lty=2)}
```




```r
print(paste("The BIC decreases and then slowly increases as the number of predictors increases as we
            which.min(bic),
            "predictors."))
```

```
## [1] "The BIC decreases and then slowly increases as the number of predictors increases as well wi
```

```r
# Plot adjr2
{plot(adjusted_r_sq,
      main="Best subset selection",
     xlab="Predictors",
     ylab="Adjusted R Squares",
     type="l")
  abline(v=which.max(adjusted_r_sq),
         col="red",
         lwd=3,
         lty=2)}
```




```r
print(paste("The Adjusted R Squares increases and then plateaus at the highest value of",
            which.max(adjusted_r_sq),
            "predictors. However, the Adjusted R Squares does not increase much after 3 predictors."
```

```
## [1] "The Adjusted R Squares increases and then plateaus at the highest value of 5 predictors. How
```

d.)

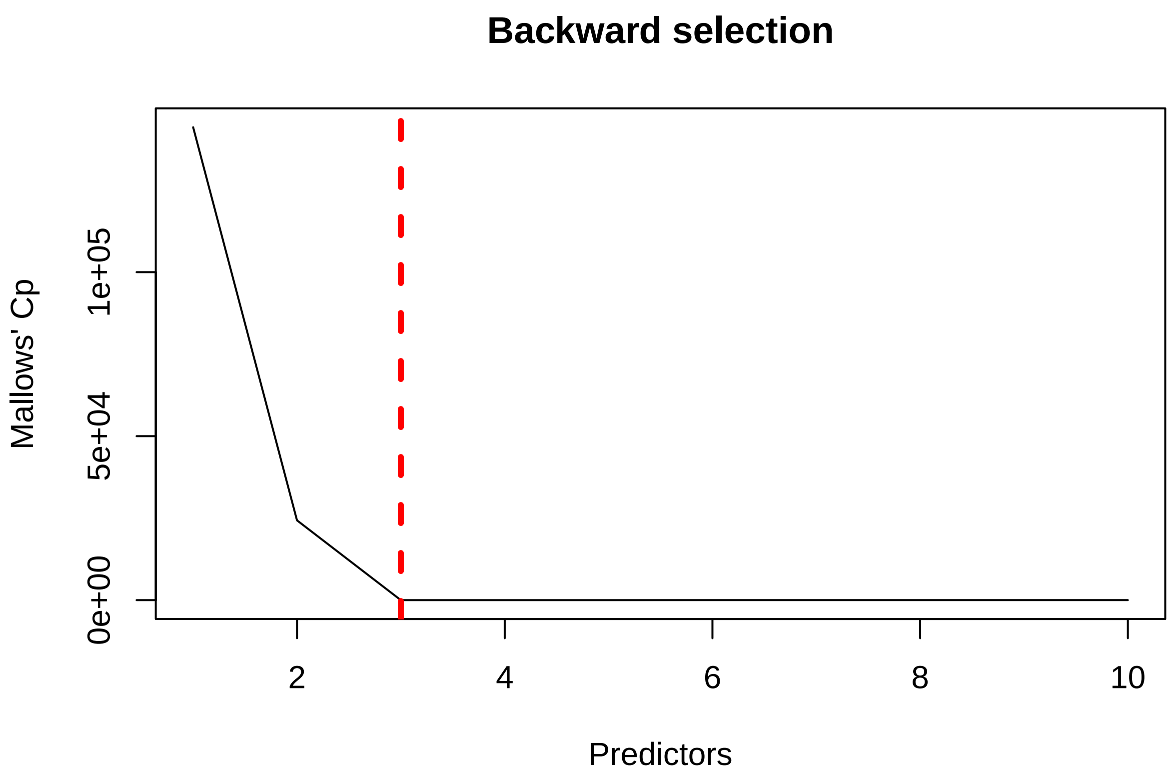
```r
# Backward step wise selection
# Find best subsets
fit <- regsubsets(Y ~ ., data = df, method = "backward", nvmax=10)
summary(fit)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = df, method = "backward", nvmax = 10)
## 10 Variables  (and intercept)
##      Forced in Forced out
## X        FALSE      FALSE
## X.2      FALSE      FALSE
## X.3      FALSE      FALSE
## X.4      FALSE      FALSE
## X.5      FALSE      FALSE
## X.6      FALSE      FALSE
## X.7      FALSE      FALSE
## X.8      FALSE      FALSE
## X.9      FALSE      FALSE
## X.10     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: backward
##           X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
## 1  ( 1 ) " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 ) " " "*" "*" " " " " " " " " " " " " " "
## 3  ( 1 ) "*" "*" "*" " " " " " " " " " " " " " "
## 4  ( 1 ) "*" "*" "*" " " " " " " " " "*" " " " "
## 5  ( 1 ) "*" "*" "*" " " " " " " " " "*" " " "*"
## 6  ( 1 ) "*" "*" "*" " " " " "*" " " "*" " " "*"
## 7  ( 1 ) "*" "*" "*" " " "*" "*" " " "*" " " "*"
## 8  ( 1 ) "*" "*" "*" " " "*" "*" "*" "*" " " "*"
## 9  ( 1 ) "*" "*" "*" " " "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
# Extract cp, bic and adjr2
cp = summary(fit)$cp
bic = summary(fit)$bic
adjusted_r_sq = summary(fit)$adjr2

# print coefficients
coef(fit, id=which.min(cp))
```

```
## (Intercept)           X         X.2         X.3
##    4.991504    7.007547    8.992661   11.012597
```

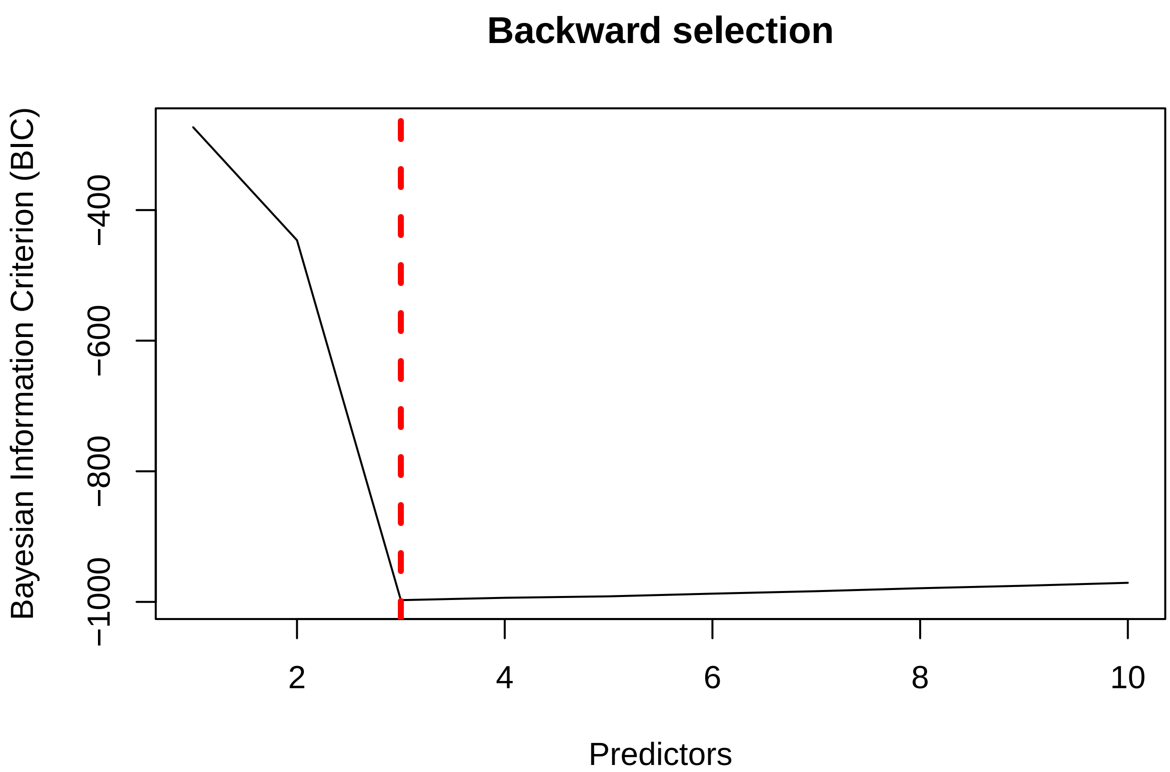
```
# Plot Cp
{plot(cp,
      main="Backward selection",
     xlab="Predictors",
     ylab="Mallows' Cp",
     type="l")
  abline(v=which.min(cp),
         col="red",
         lwd=3,
         lty=2)}
```




```
print(paste("The Cp decreases and then plateaus at the lowest value of",
            which.min(cp),
            "predictors."))
```

```
## [1] "The Cp decreases and then plateaus at the lowest value of 3 predictors."
```

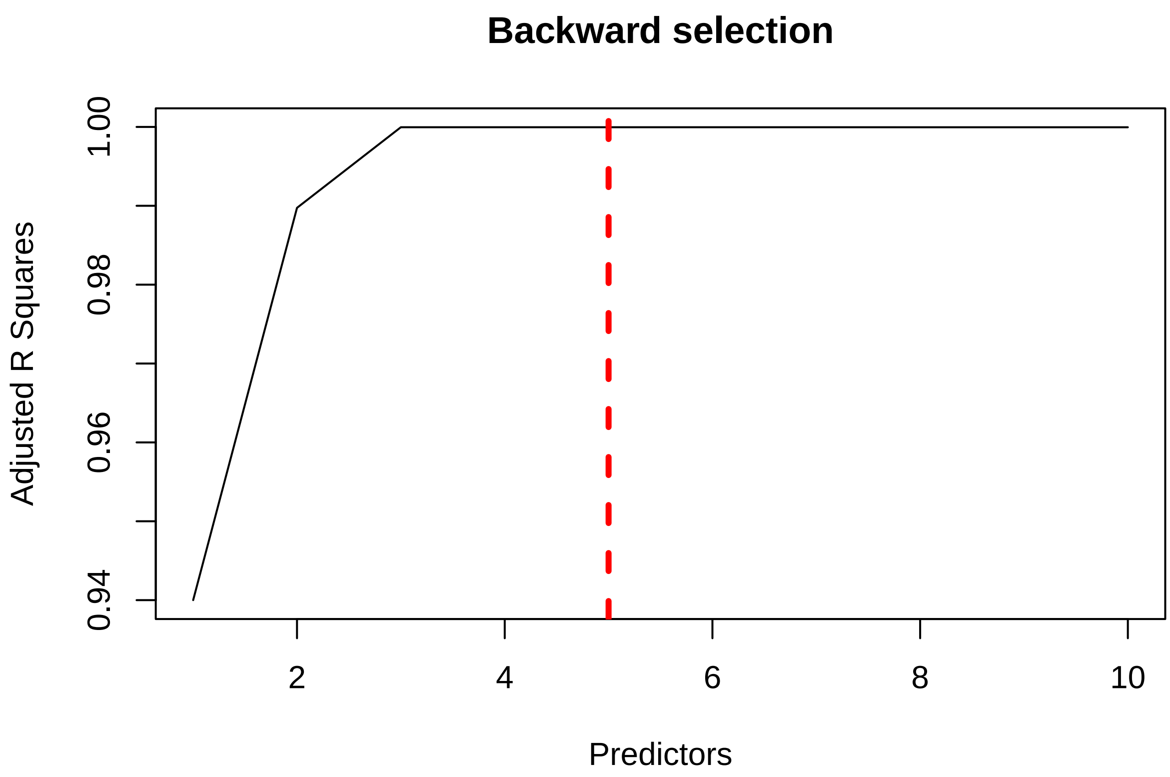
```
# Plot BIC
{plot(bic,
      main="Backward selection",
     xlab="Predictors",
     ylab="Bayesian Information Criterion (BIC)",
     type="l")
  abline(v=which.min(bic),
         col="red",
         lwd=3,
         lty=2)}
```




```r
print(paste("The BIC decreases and then slowly increases as the number of predictors increases as we
            which.min(bic),
            "predictors."))
```

```
## [1] "The BIC decreases and then slowly increases as the number of predictors increases as well wi
```

```r
# Plot adjr2
{plot(adjusted_r_sq,
      main="Backward selection",
     xlab="Predictors",
     ylab="Adjusted R Squares",
     type="l")
  abline(v=which.max(adjusted_r_sq),
         col="red",
         lwd=3,
         lty=2)}
```




```
print(paste("The Adjusted R Squares increases and then plateaus at the highest value of",
            which.max(adjusted_r_sq),
            "predictors. However, the Adjusted R Squares does not increase much after 3 predictors."
```

```
## [1] "The Adjusted R Squares increases and then plateaus at the highest value of 5 predictors. How
```

```
print("Therefore the first 3 predictors (X, X^2, X^3) seem top be the best predictors to include whe
```

```
## [1] "Therefore the first 3 predictors (X, X^2, X^3) seem top be the best predictors to include wh
```

```
# Forward step wise selection
# Find best subsets
fit <- regsubsets(Y ~ ., data = df, method = "forward", nvmax=10)
summary(fit)
```

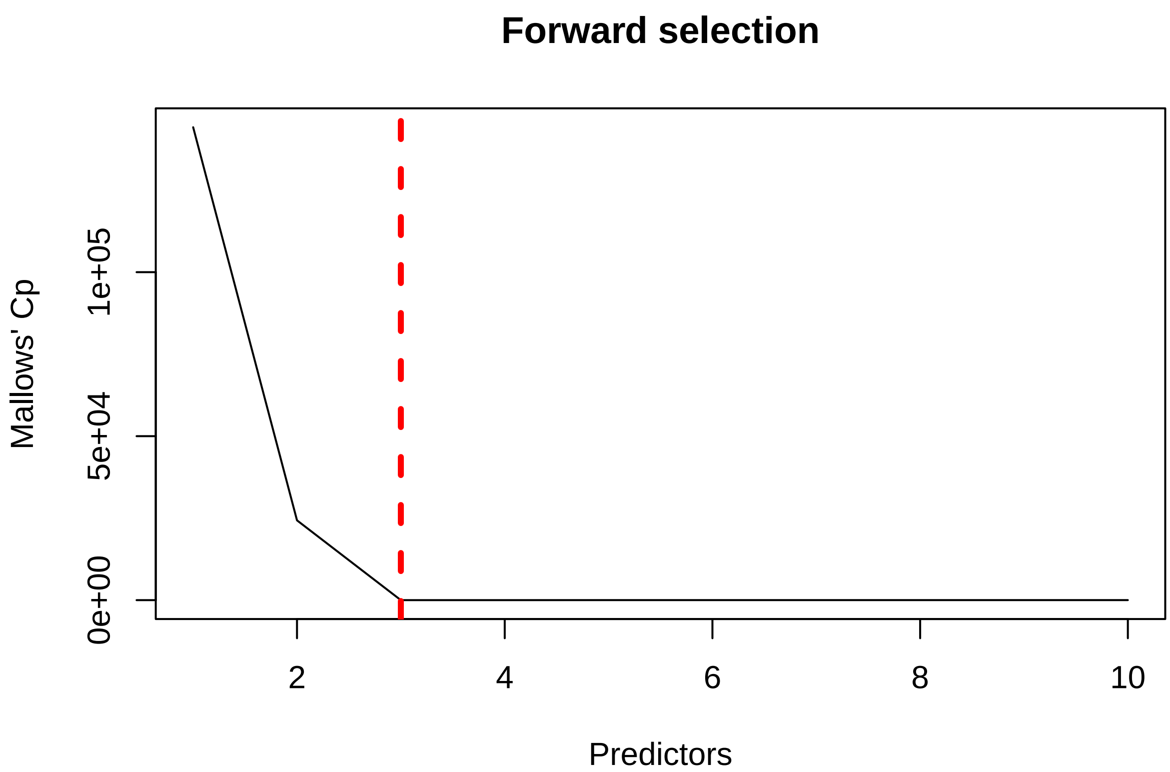
```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = df, method = "forward", nvmax = 10)
## 10 Variables  (and intercept)
##      Forced in Forced out
## X        FALSE      FALSE
## X.2      FALSE      FALSE
## X.3      FALSE      FALSE
## X.4      FALSE      FALSE
## X.5      FALSE      FALSE
## X.6      FALSE      FALSE
```

```
## X.7      FALSE     FALSE
## X.8      FALSE     FALSE
## X.9      FALSE     FALSE
## X.10     FALSE     FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##           X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
## 3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " "
## 4  ( 1 )  "*" "*" "*" "*" " " " " " " " " " " " "
## 5  ( 1 )  "*" "*" "*" "*" " " " " " " " " "*" " "
## 6  ( 1 )  "*" "*" "*" "*" "*" " " " " " " "*" " "
## 7  ( 1 )  "*" "*" "*" "*" "*" "*" " " " " "*" " "
## 8  ( 1 )  "*" "*" "*" "*" "*" "*" " " " " "*" "*"
## 9  ( 1 )  "*" "*" "*" "*" "*" "*" " " "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
# Extract cp, bic and adjr2
cp = summary(fit)$cp
bic = summary(fit)$bic
adjusted_r_sq = summary(fit)$adjr2

# print coefficients
coef(fit, id=which.min(cp))
```

```
## (Intercept)           X         X.2         X.3
##    4.991504    7.007547    8.992661   11.012597
```

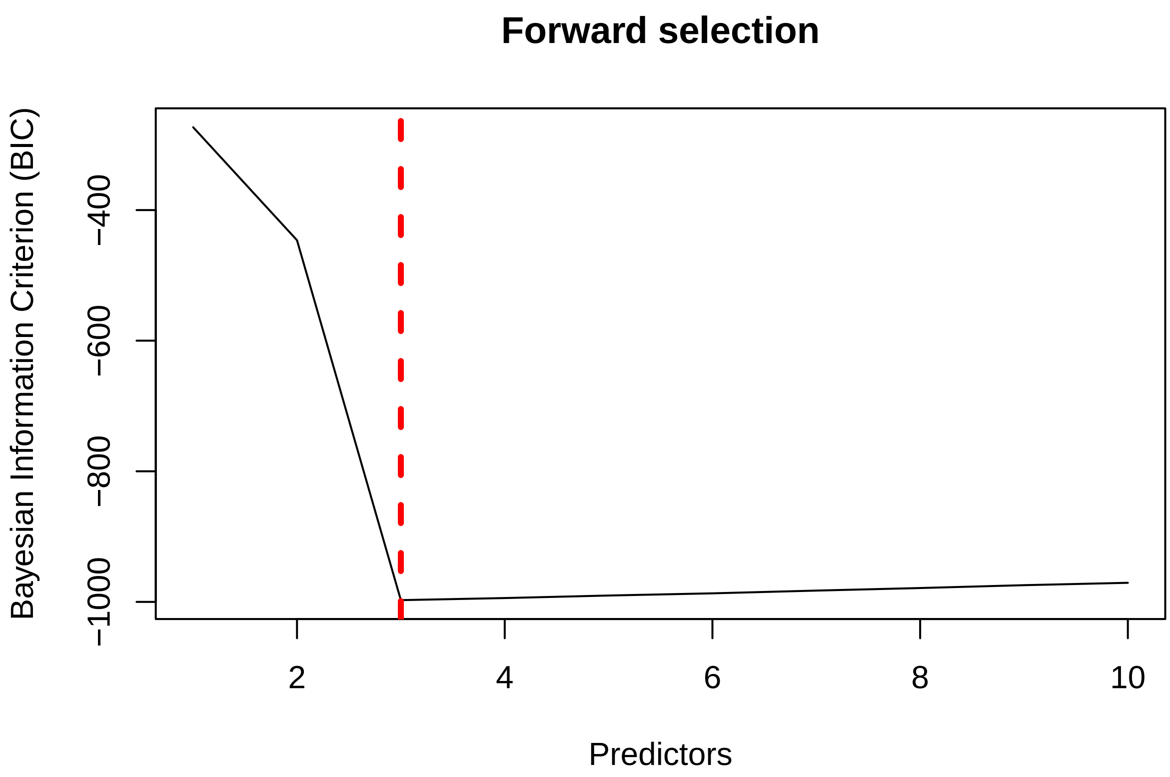
```
# Plot Cp
{plot(cp,
      main="Forward selection",
     xlab="Predictors",
     ylab="Mallows' Cp",
     type="l")
  abline(v=which.min(cp),
         col="red",
         lwd=3,
         lty=2)}
```




```r
print(paste("The Cp decreases and then plateaus at the lowest value of",
            which.min(cp),
            "predictors."))
```

```
## [1] "The Cp decreases and then plateaus at the lowest value of 3 predictors."
```

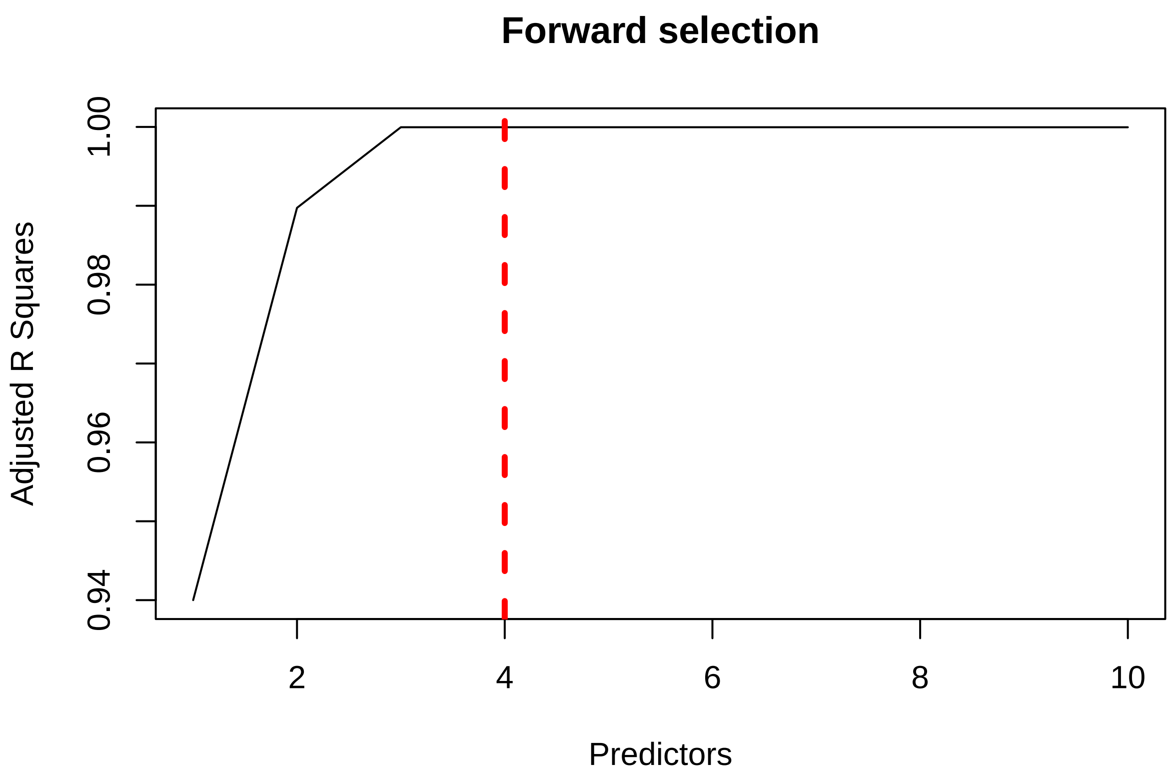
```r
# Plot BIC
{plot(bic,
      main="Forward selection",
     xlab="Predictors",
     ylab="Bayesian Information Criterion (BIC)",
     type="l")
  abline(v=which.min(bic),
         col="red",
         lwd=3,
         lty=2)}
```




```
print(paste("The BIC decreases and then slowly increases as the number of predictors increases as we
            which.min(bic),
            "predictors."))
```

```
## [1] "The BIC decreases and then slowly increases as the number of predictors increases as well wi
```

```
# Plot adjr2
{plot(adjusted_r_sq,
      main="Forward selection",
     xlab="Predictors",
     ylab="Adjusted R Squares",
     type="l")
  abline(v=which.max(adjusted_r_sq),
         col="red",
         lwd=3,
         lty=2)}
```




```
print(paste("The Adjusted R Squares increases and then plateaus at the highest value of",
            which.max(adjusted_r_sq),
            "predictors."))
```

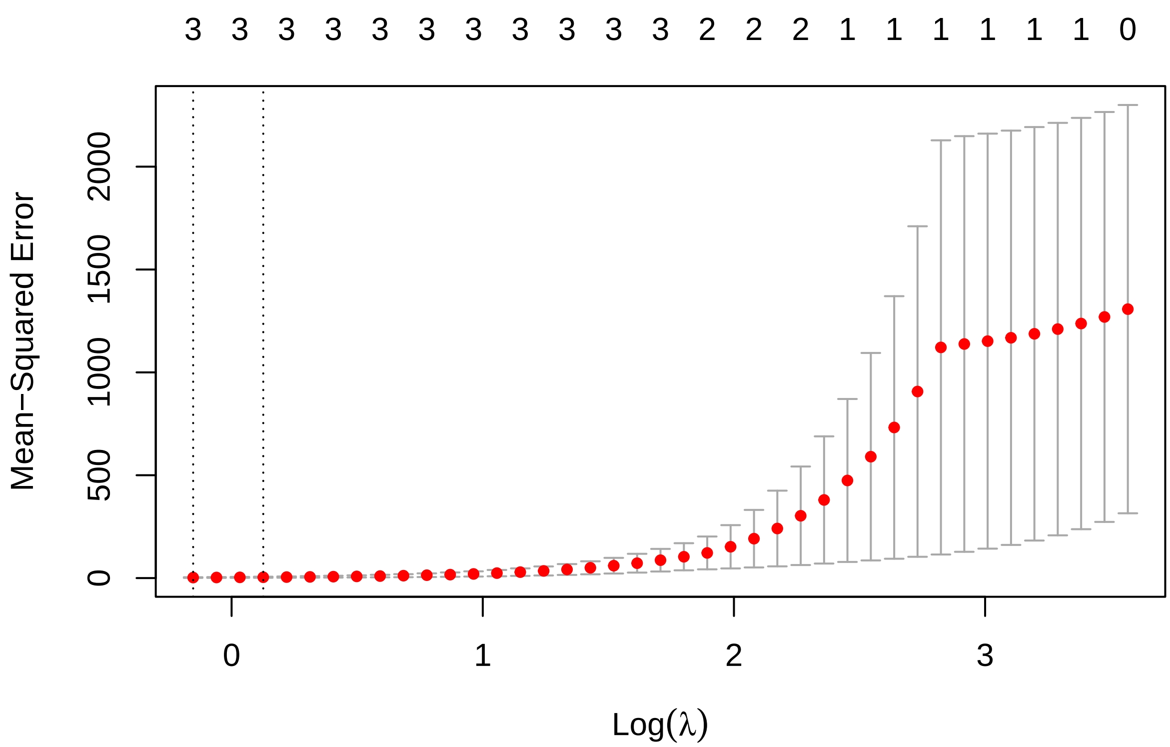
```
## [1] "The Adjusted R Squares increases and then plateaus at the highest value of 4 predictors."
```

This shows that the backward, forward, and best subset methods of selection all produce similar results with basically the same conclusion.

e.)

```
# specify data
x_mat = data.matrix(df[1:10])
y_mat = data.matrix(Y)

# train/test split
train_idx = sample(1:nrow(x_mat), nrow(x_mat)/2)
test_idx = (-train_idx)

# cross validation
folds_fit = cv.glmnet(x_mat[train_idx,],
                      y_mat[train_idx,],
                      alpha = 1,
                      folds=10)
```

```r
# plot results
plot(folds_fit)
```




```r
# Test data MSE
pred = predict(folds_fit,
               s = folds_fit$lambda.min,
               newx = x_mat[test_idx, ])
test_mse = mean((pred - Y[test_idx])^2)
print(paste("Test MSE: ", test_mse))
```

```
## [1] "Test MSE:  1.44424010506996"
```

```r
print(paste("Mininum lambda: ", folds_fit$lambda.min))
```

```
## [1] "Mininum lambda:  0.858061060097578"
```

When using Lasso we can se that the model with 3 predictors performs the best

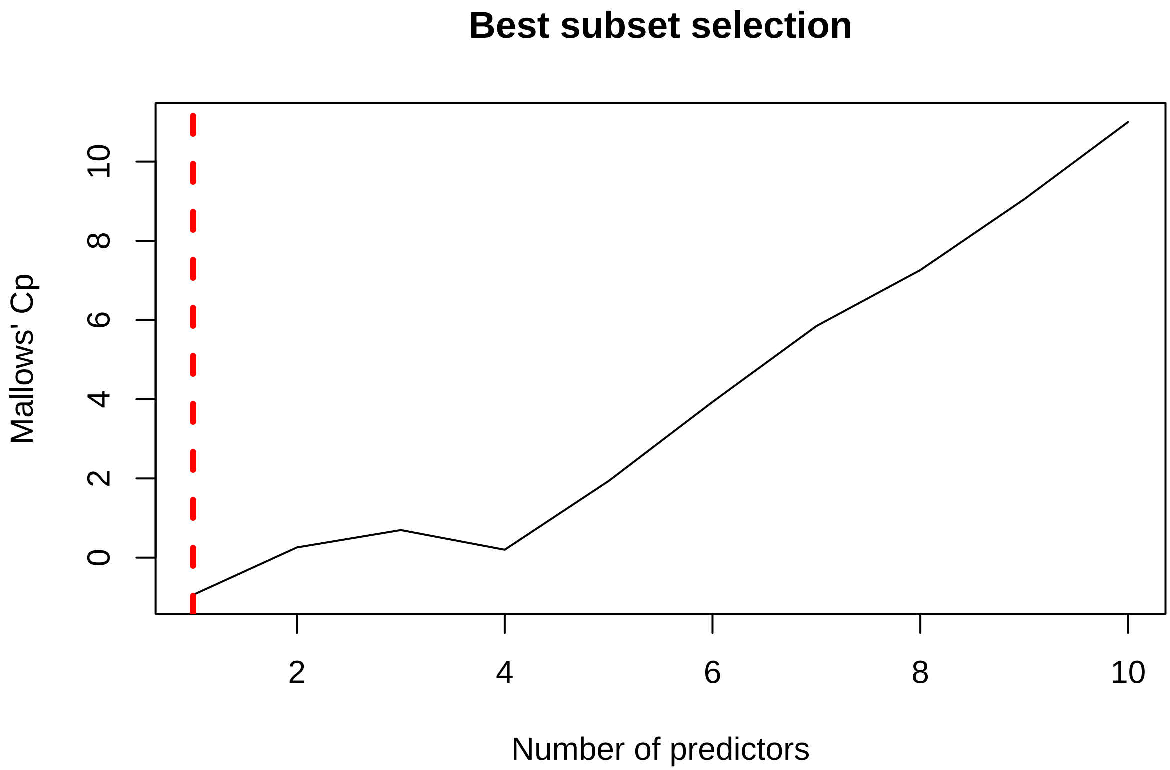
f.)

```r
# define new response
b_7 = 13
Y = c(b_0 + b_7*X^7 + e)
```

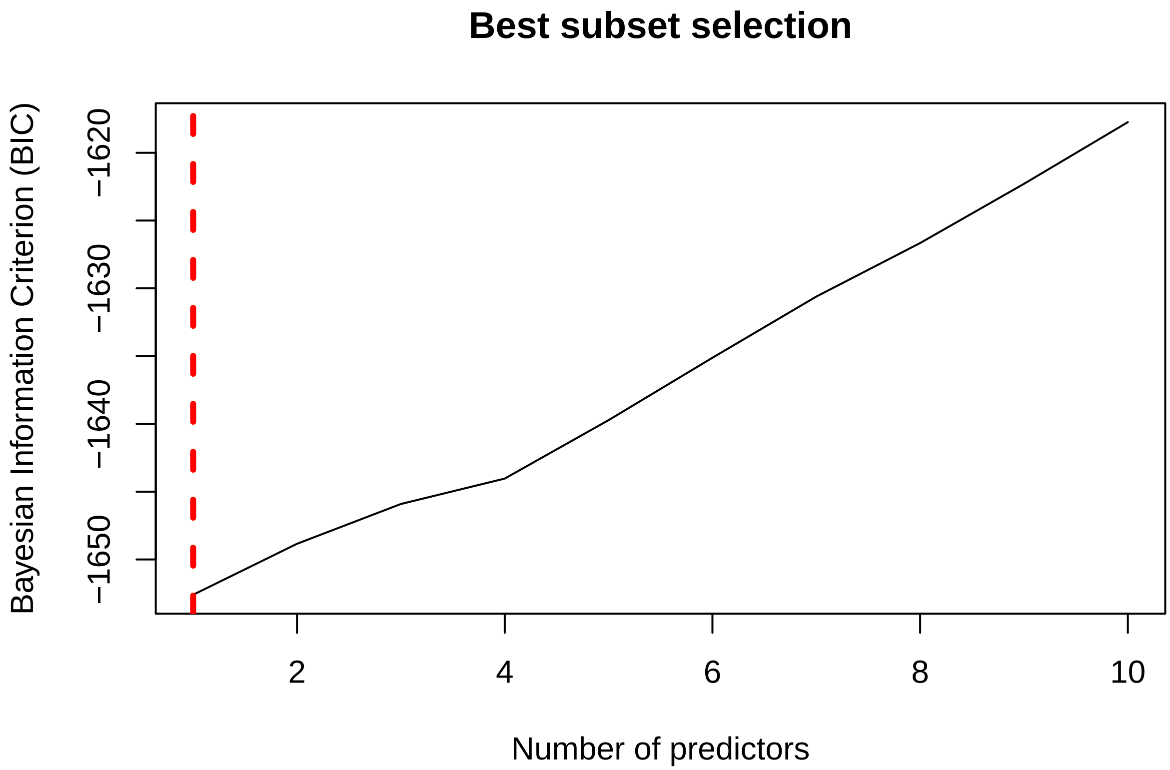
```r
# Best subset selection
# Save data in dataframe
df = data.frame(X,X^2,X^3,X^4,X^5,X^6,X^7,X^8,X^9,X^10,Y)

# Find best subsets
fit <- regsubsets(Y ~ ., data = df, nvmax=10)
summary(fit)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = df, nvmax = 10)
## 10 Variables  (and intercept)
##      Forced in Forced out
## X        FALSE      FALSE
## X.2      FALSE      FALSE
## X.3      FALSE      FALSE
## X.4      FALSE      FALSE
## X.5      FALSE      FALSE
## X.6      FALSE      FALSE
## X.7      FALSE      FALSE
## X.8      FALSE      FALSE
## X.9      FALSE      FALSE
## X.10     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           X   X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9 X.10
## 1  ( 1 )  " " " " " " " " " " " " "*" " " " " " "
## 2  ( 1 )  " " " " " " " " "*" " " "*" " " " " " "
## 3  ( 1 )  " " " " " " " " " " " " "*" " " "*" "*"
## 4  ( 1 )  " " " " " " " " " " "*" "*" "*" " " "*"
## 5  ( 1 )  "*" " " " " " " " " "*" "*" "*" " " "*"
## 6  ( 1 )  " " " " " " " " "*" "*" "*" "*" "*" "*"
## 7  ( 1 )  "*" " " "*" " " "*" "*" "*" "*" " " "*"
## 8  ( 1 )  "*" " " "*" " " "*" "*" "*" "*" "*" "*"
## 9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

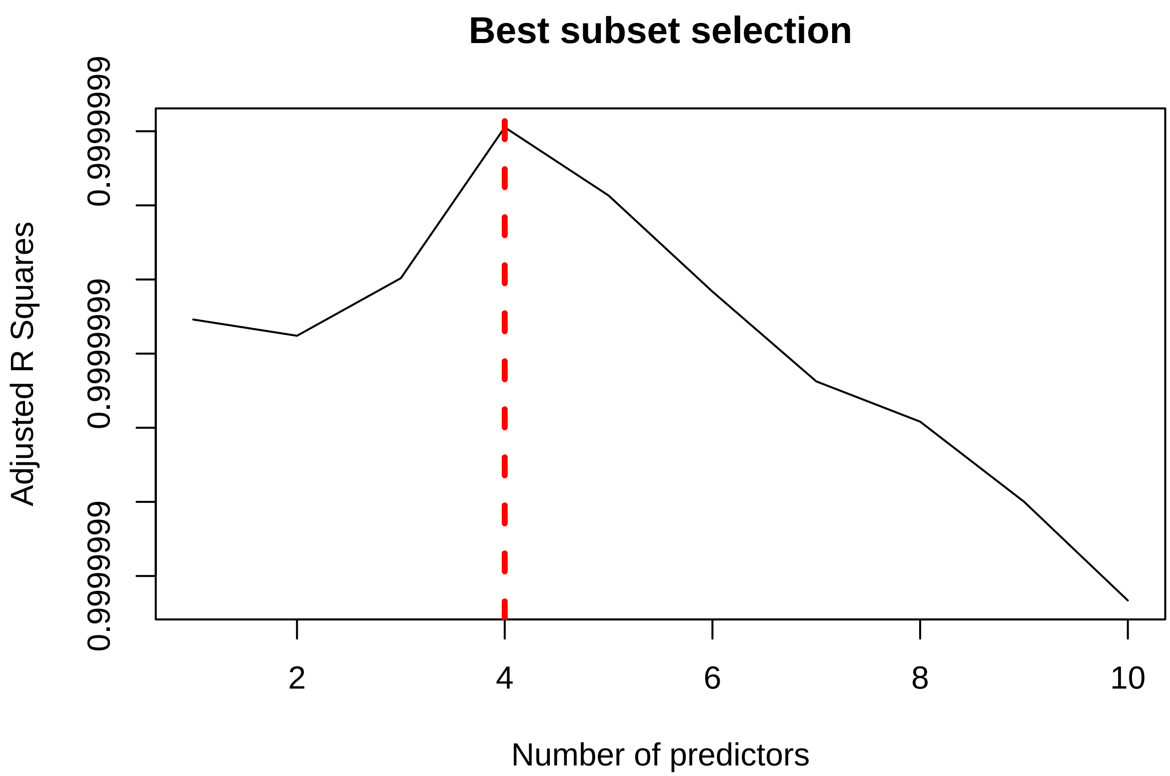
```r
# Extract cp, bic and adjr2
cp = summary(fit)$cp
bic = summary(fit)$bic
adjusted_r_sq = summary(fit)$adjr2

# Plot Cp
{plot(cp,
      main="Best subset selection",
     xlab="Number of predictors",
     ylab="Mallows' Cp",
     type="l")
  abline(v=which.min(cp),
         col="red",
         lwd=3,
         lty=2)}
```




```
# Plot BIC
{plot(bic,
     main="Best subset selection",
     xlab="Number of predictors",
     ylab="Bayesian Information Criterion (BIC)",
     type="l")
  abline(v=which.min(bic),
         col="red",
         lwd=3,
         lty=2)}
```



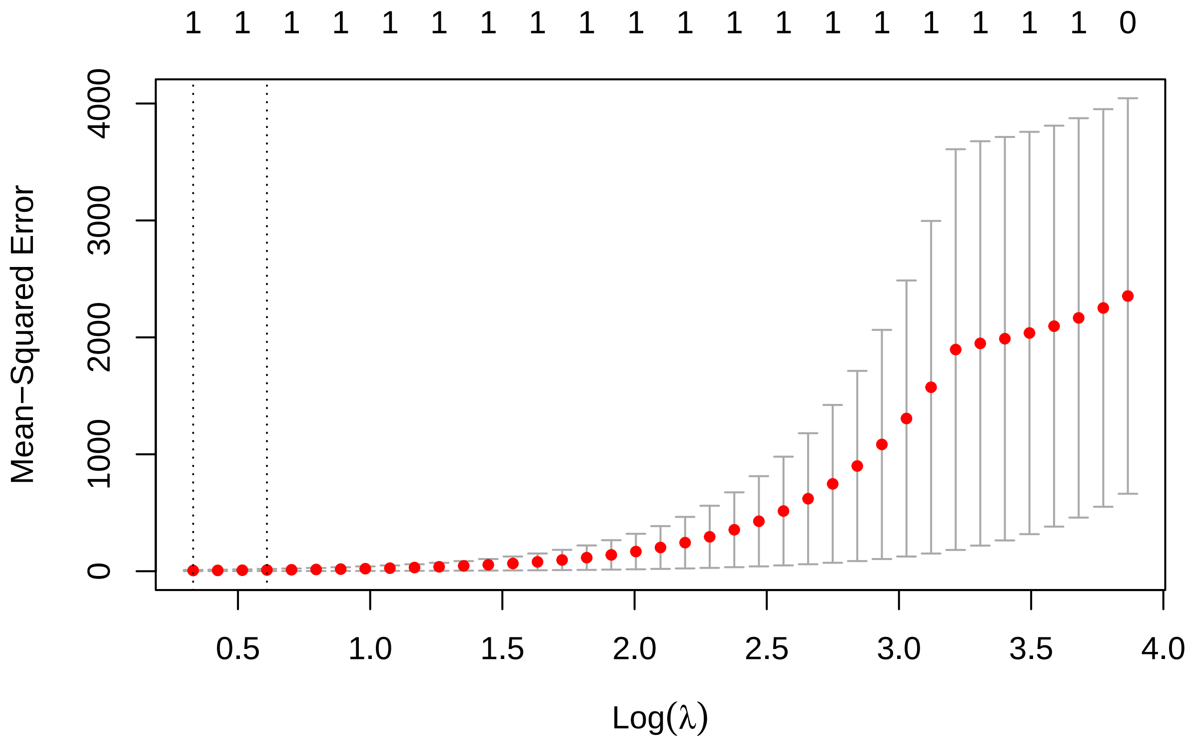

```r
# Plot adjr2
{plot(adjusted_r_sq,
      main="Best subset selection",
     xlab="Number of predictors",
     ylab="Adjusted R Squares",
     type="l")
  abline(v=which.max(adjusted_r_sq),
         col="red",
         lwd=3,
         lty=2)}
```




```r
# print coefficients
coef(fit, id=which.min(cp))
```

```
## (Intercept)         X.7
##    4.985807   13.000382
```

```r
# LASSO
# specify data
x_mat = data.matrix(df[1:10])
y_mat = data.matrix(Y)

# train/test split
train_idx = sample(1:nrow(x_mat), nrow(x_mat)/2)
test_idx = (-train_idx)

# cross validation
folds_fit = cv.glmnet(x_mat[train_idx,],
                      y_mat[train_idx,],
                      alpha = 1,
                      folds=10)

# plot results
plot(folds_fit)
```




```r
# Test data MSE
pred = predict(folds_fit,
               s = folds_fit$lambda.min,
               newx = x_mat[test_idx, ])
test_mse = mean((pred - Y[test_idx])^2)
print(paste("Test MSE: ", test_mse))
```

```
## [1] "Test MSE:  1497.8713152662"
```

```r
print(paste("Mininum lambda: ", folds_fit$lambda.min))
```

```
## [1] "Mininum lambda:  1.39164858912155"
```

These results show that there is only one predictor worth looking at according to the Cp and the BIC. The Adjusted R squares indicates that there might be 4 predictors, however the difference between the adjusted R squares for 1 predictor and 4 is extremely small so 1 is likely the best option. From this we can also see that the best single predictor is the Xˆ7 predictor which is most similar to what was used in the response.