Bruce Campbell ST-617 Homework 2

Tue Jul 12 09:44:26 2016

Chapter 6

Problem 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.

```
rm(list = ls())
set.seed(123)
X <- rnorm(100, mean = 0, sd = 1)

epsilon <- rnorm(100, mean = 0, sd = 1)
```

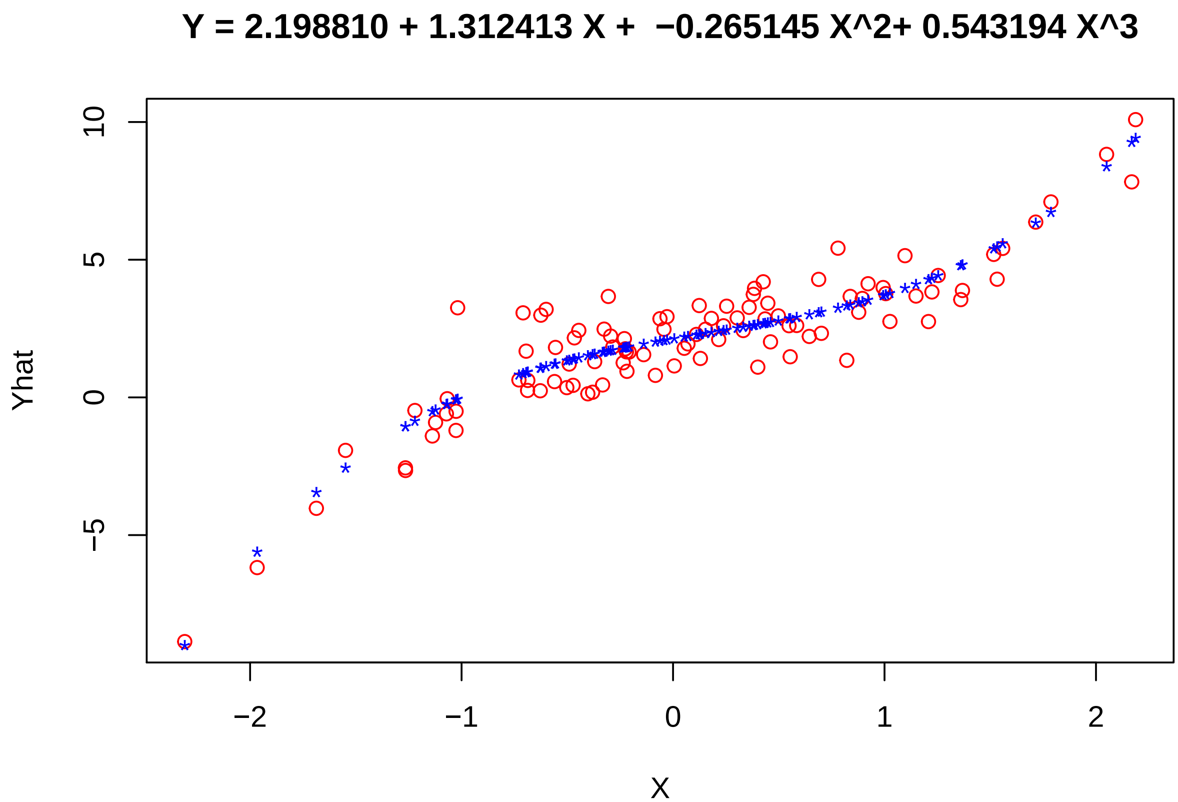
b)

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.

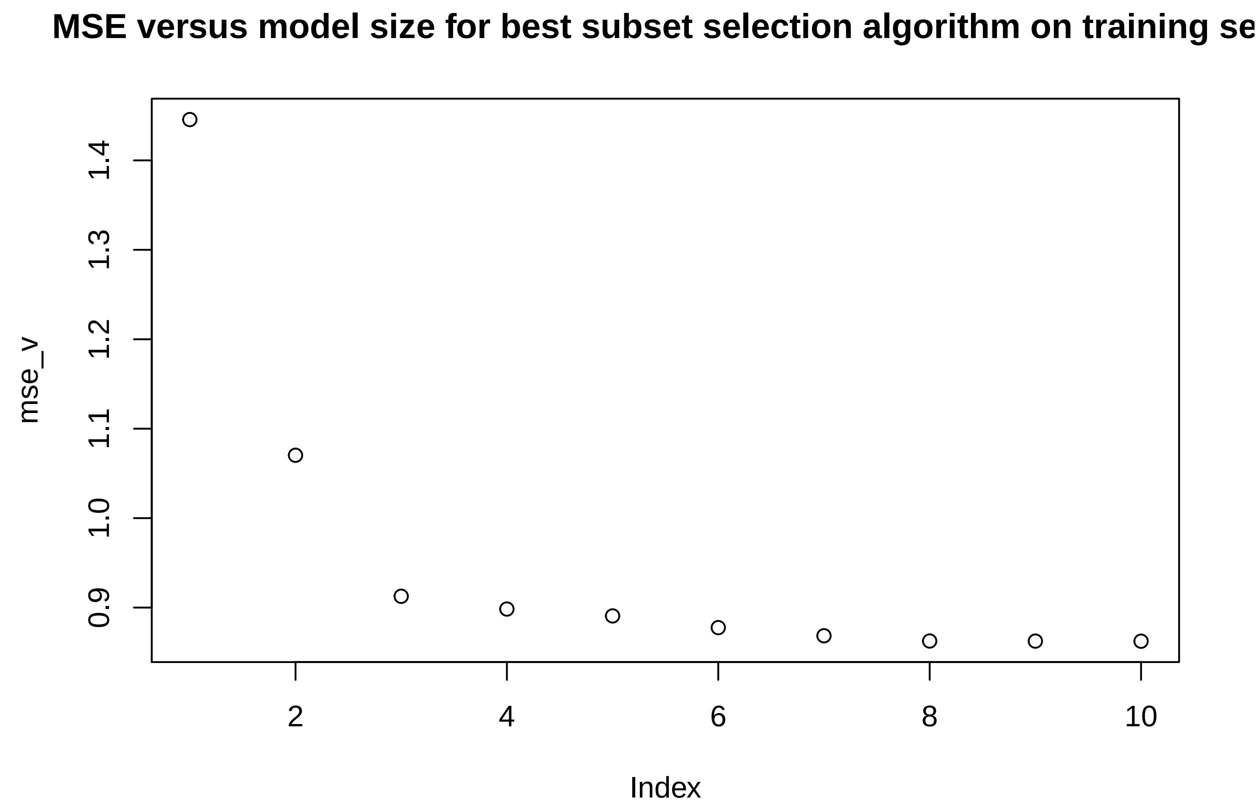
```
beta = rnorm(4, mean = 0, sd = 1)
Y <- matrix(NA, nrow = 100, ncol = 1)
Yhat <- matrix(NA, nrow = 100, ncol = 1)

for (i in 1:100) {
    Y[i] = beta[1] + beta[2] * X[i] + beta[3] * X[i]^2 + beta[4] * X[i]^3

    Yhat[i] = beta[1] + beta[2] * X[i] + beta[3] * X[i]^2 + beta[4] * X[i]^3 +
        epsilon[i]
}

plot(X, Yhat, col = "red")
points(X, Y, pch = "*", col = "blue")
title(main = sprintf("Y = %f + %f X +  %f X^2+ %f X^3", beta[1], beta[2], +beta[3],
    beta[4]), cex = 4.6)
```



c)

Use the regsubsets() function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, ..., X^10$. What is the best model obtained according to Cp, BIC, and adjusted R2? 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 .

```
DF <- as.data.frame(X)
DF <- cbind(DF, Yhat)
names(DF) = c("X", "Y")
library(leaps)

regfit.full <- 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 = DF, nvmax = 10)

reg.summary <- summary(regfit.full)

mse_v <- reg.summary$rss/nrow(DF)

plot(mse_v)
title("MSE versus model size for best subset selection algorithm on training set.")
```



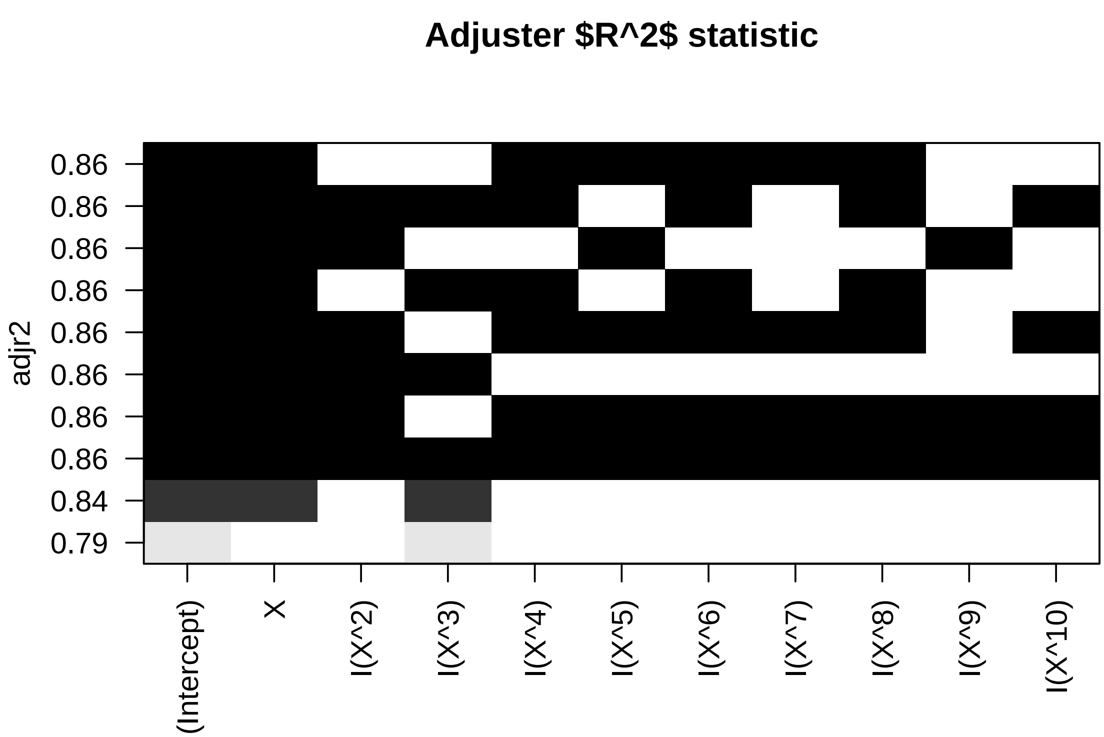
We see that there is a sharp drop in the training set MSE until 3 or 4 predictors are included and that there is a steady decrease as additional polynomial terms are included. This is attributed to over fitting to the training data.

```
summary(regfit.full)
```

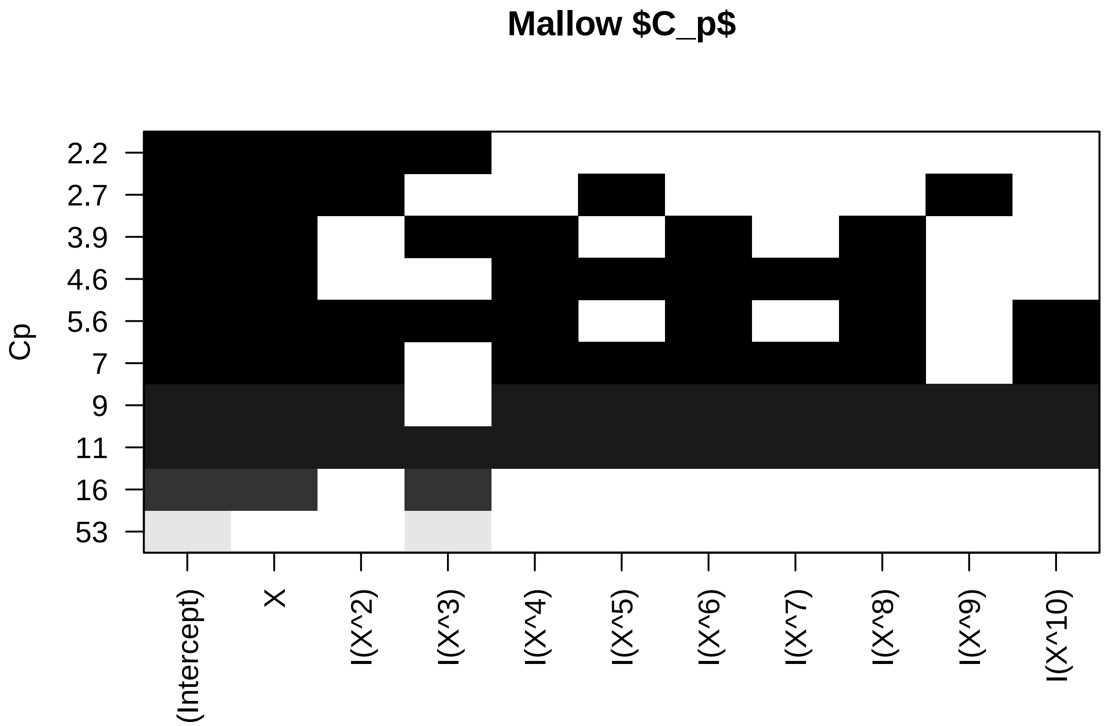
```
## Subset selection object
## Call: regsubsets.formula(Y ~ X + I(X^2) + I(X^3) + I(X^4) + I(X^5) +
##     I(X^6) + I(X^7) + I(X^8) + I(X^9) + I(X^10), data = DF, nvmax = 10)
## 10 Variables  (and intercept)
##         Forced in Forced out
## X           FALSE      FALSE
## I(X^2)      FALSE     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

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

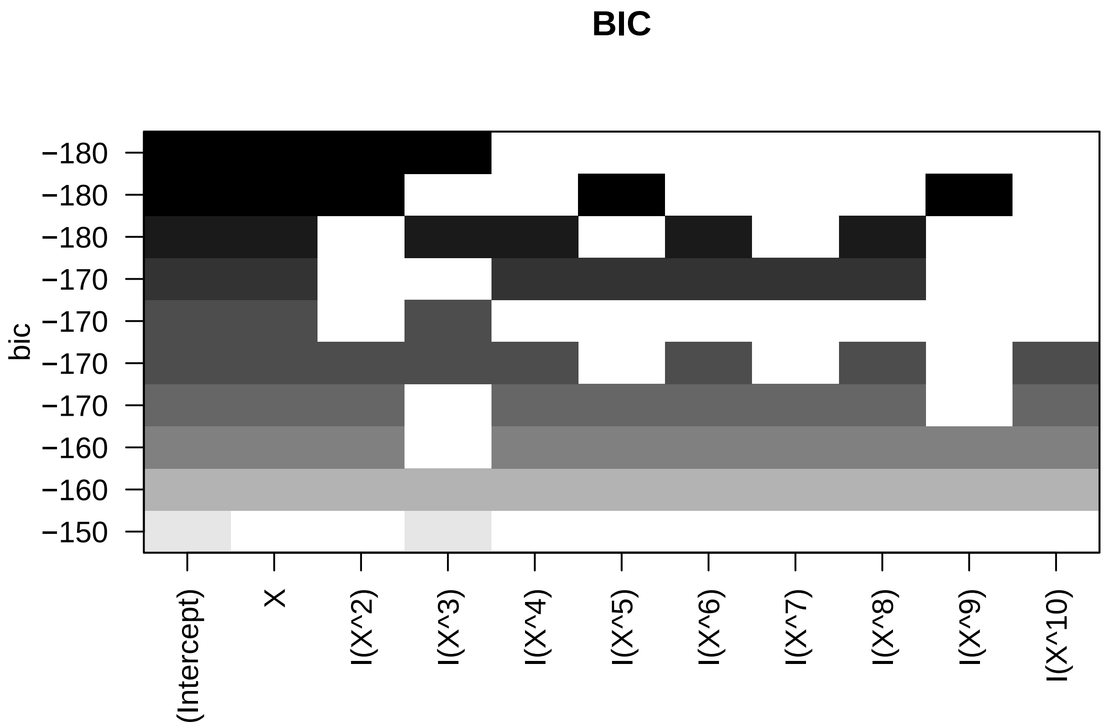
We notice the best subset algorithm has correctly included the proper terms in the third sets of predictors.



We need to remember that with R^2 statistic the values closer to 1 are a better fit. Among those with a value of 0.86 we see that the model with $Intercept, X, X^2, X^3$ is selected.



The C_p statistic indicates that the best model contains $Intercept, X, X^2, X^3$

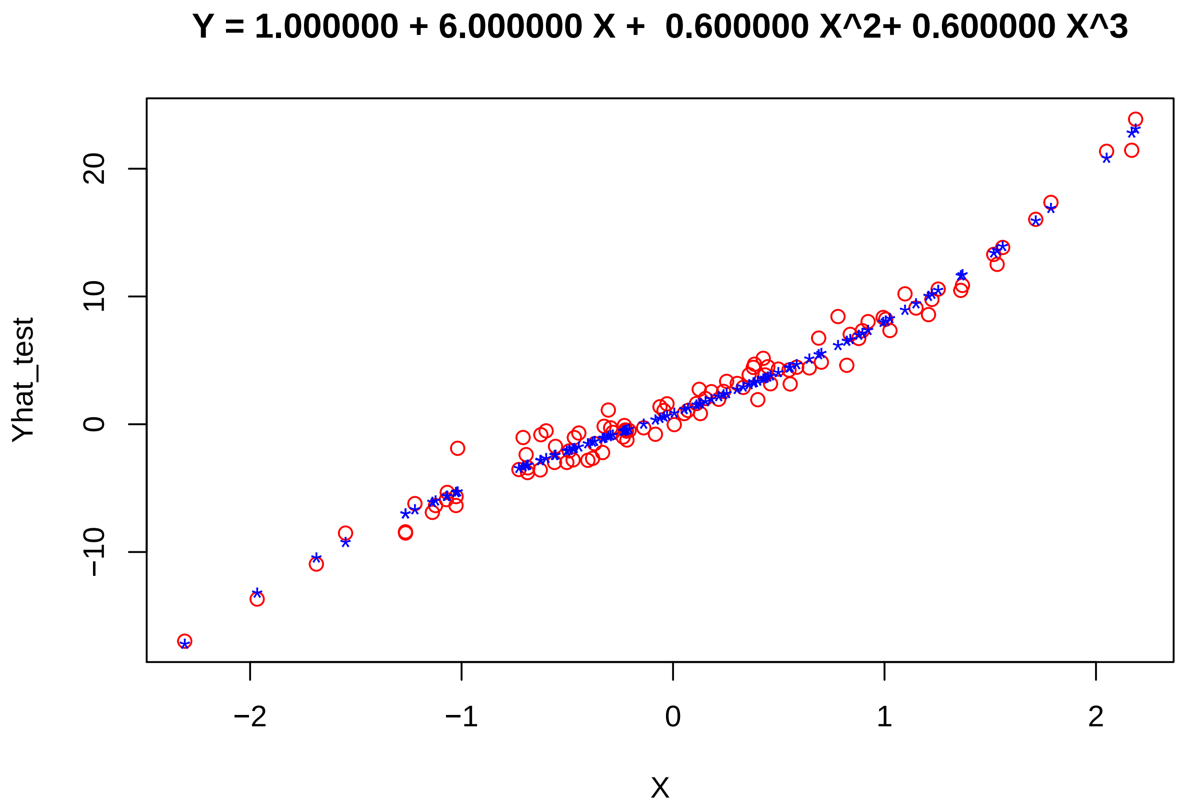


The BIC statistic indicates that the best model contains $Intercept, X, X^2, X^3$

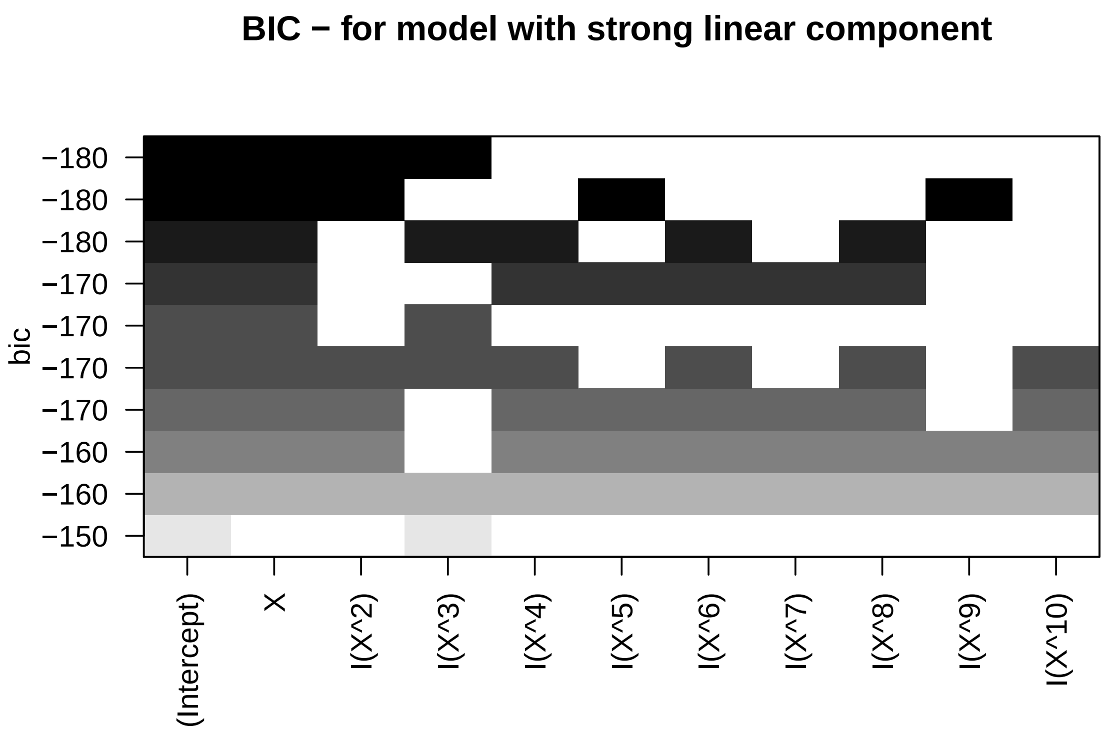
There was a problem with knitr where the cache was corrupted and the plots from old models were included. The section below is retained for that purpose.

```
beta_test <- matrix(NA, nrow = 4, ncol = 1)
beta_test[1] = 1
beta_test[2] = 6
beta_test[3] = 0.6
beta_test[4] = 0.6

Y_test <- matrix(NA, nrow = 100, ncol = 1)
Yhat_test <- matrix(NA, nrow = 100, ncol = 1)

for (i in 1:100) {
    Y_test[i] = beta_test[1] + beta_test[2] * X[i] + beta_test[3] * X[i]^2 +
        beta_test[4] * X[i]^3

    Yhat_test[i] = beta_test[1] + beta_test[2] * X[i] + beta_test[3] * X[i]^2 +
        beta_test[4] * X[i]^3 + epsilon[i]
}
DF_test <- as.data.frame(X)
DF_test <- cbind(DF, Yhat_test)
names(DF) = c("X", "Y")

plot(X, Yhat_test, col = "red")
```

```r
points(X, Y_test, pch = "*", col = "blue")
title(main = sprintf("Y = %f + %f X +  %f X^2+ %f X^3", beta_test[1], beta_test[2],
    +beta_test[3], beta_test[4]), cex = 4.6)
```



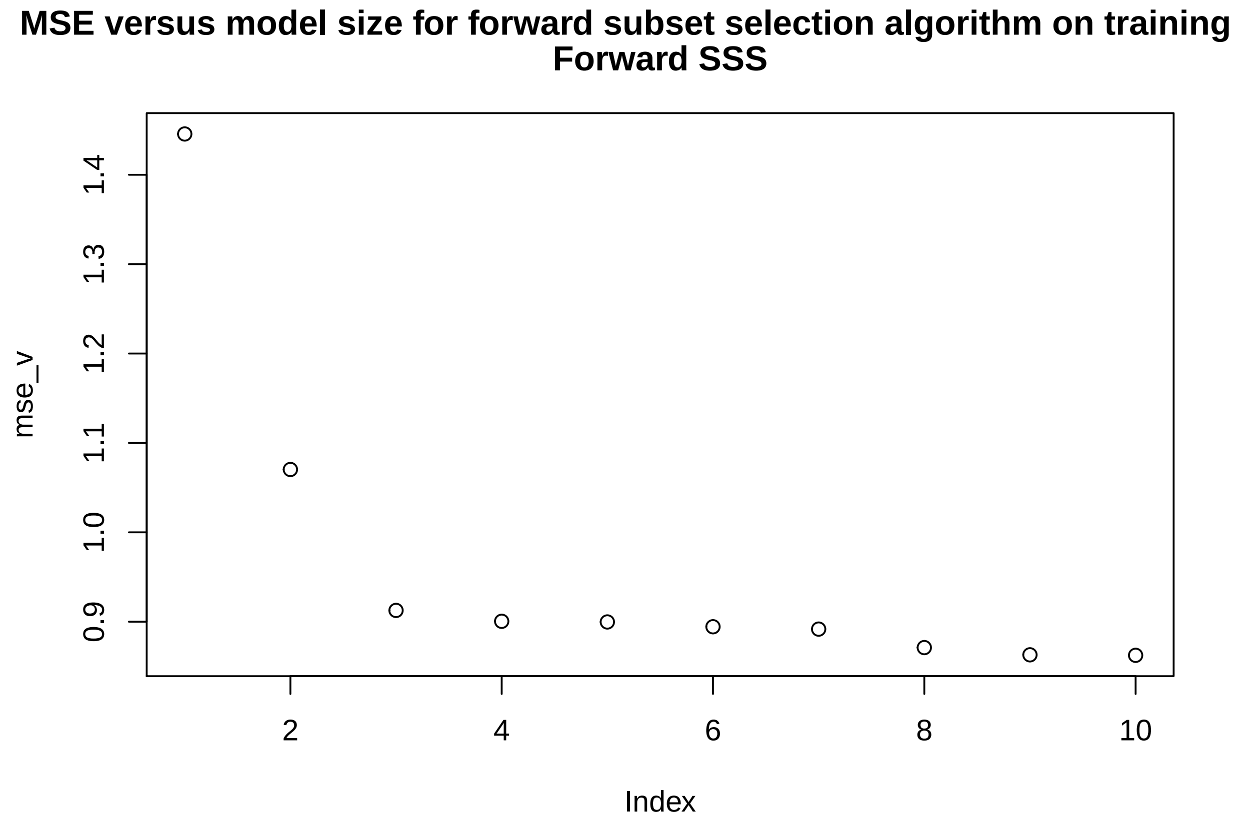

```r
regfit.full <- regsubsets(Y ~ X + I(X^2) + I(X^3) + I(X^4) + I(X^5) + I(X^6) +
    I(X^7) + I(X^8) + I(X^9) + I(X^10), data = DF_test, nvmax = 10)
plot(regfit.full, scale = "bic")
title("BIC - for model with strong linear component ")
```



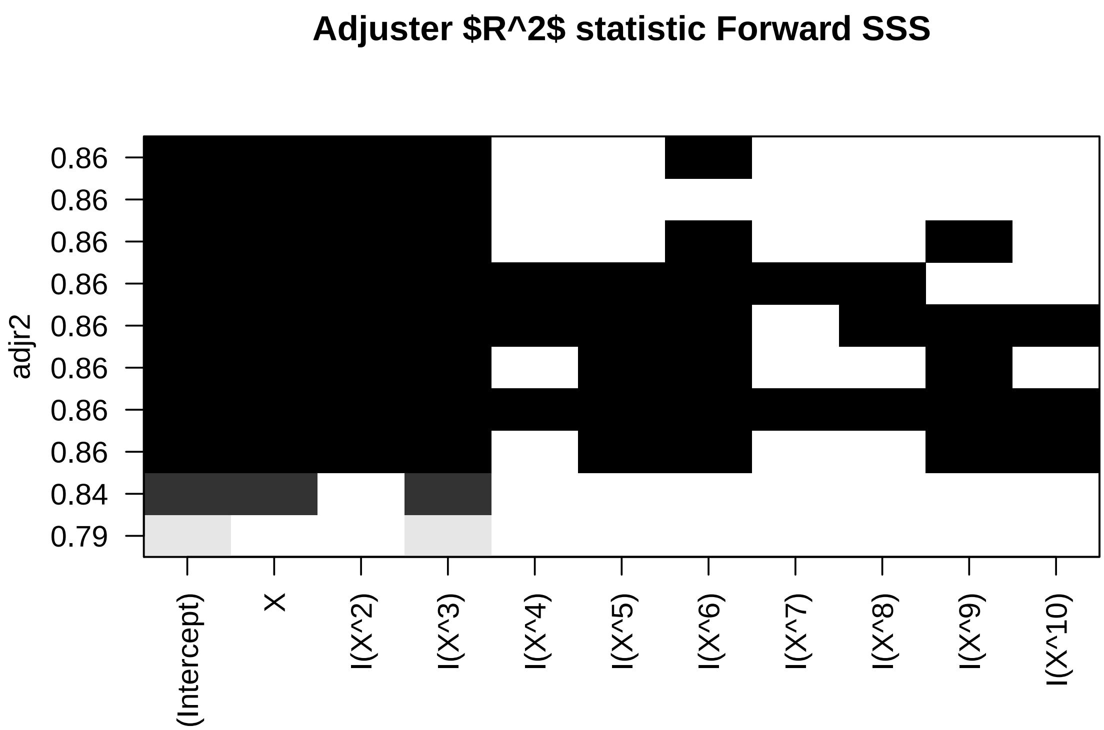
###d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

FORWARD SSS

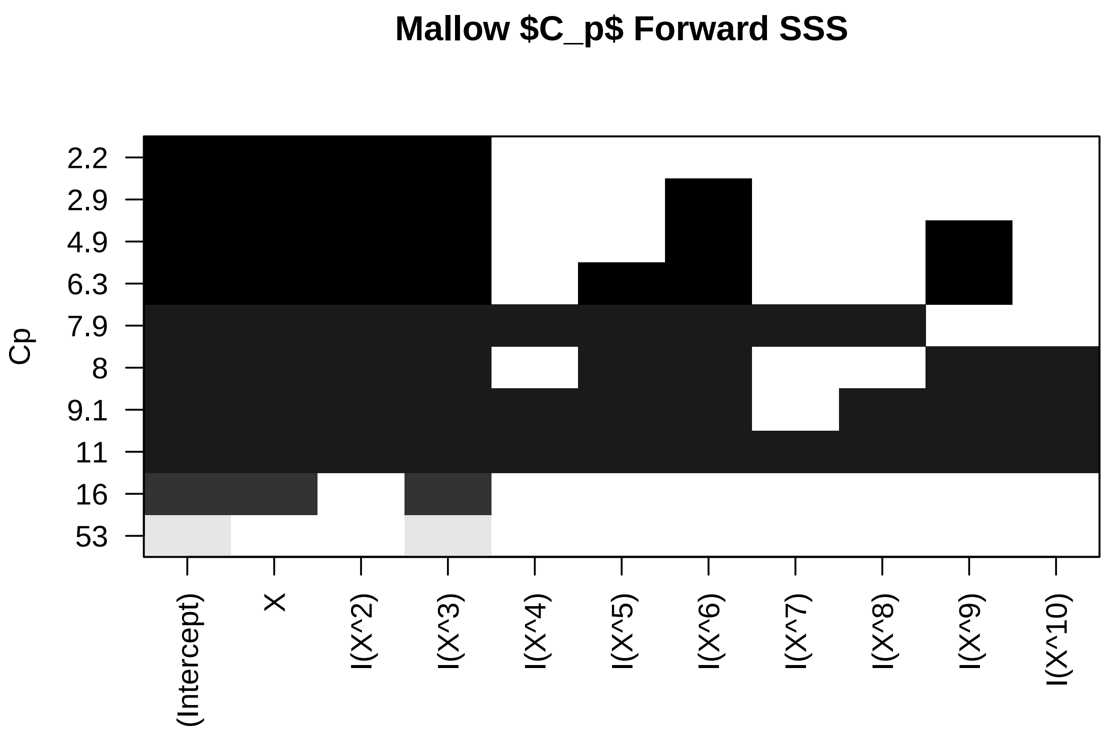
```
regfit.full <- regsubsets(Y ~ X + I(X^2) + I(X^3) + I(X^4) + I(X^5) + I(X^6) +
    I(X^7) + I(X^8) + I(X^9) + I(X^10), data = DF, nvmax = 10, method = "forward")

reg.summary <- summary(regfit.full)

mse_v <- reg.summary$rss/nrow(DF)

plot(mse_v)
title(c("MSE versus model size for forward subset selection algorithm on training set.",
    "Forward SSS"))
```



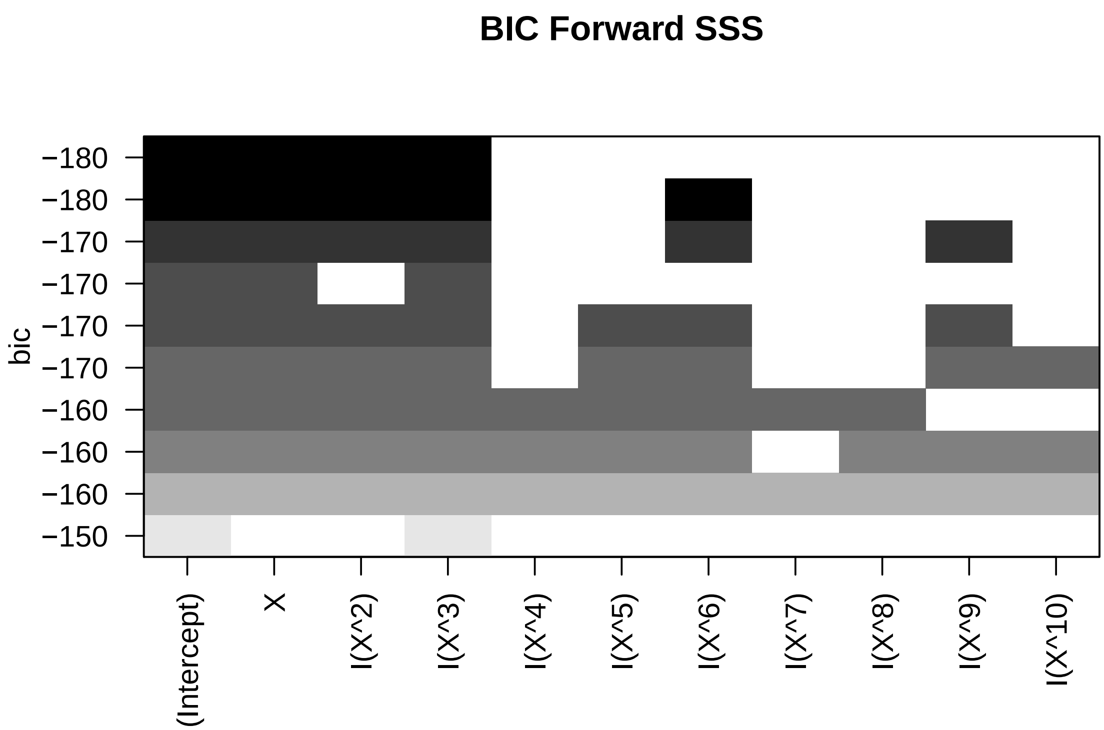
We see that there is a sharp drop in the training set MSE until 3 or 4 predictors are included and there is a steady decrease as additional polynomial terms are included. This is attributed to over fitting to the training data.



Among those with a value of 0.86 we see that the model with $Intercept, X, X^2, X^3$ is selected.



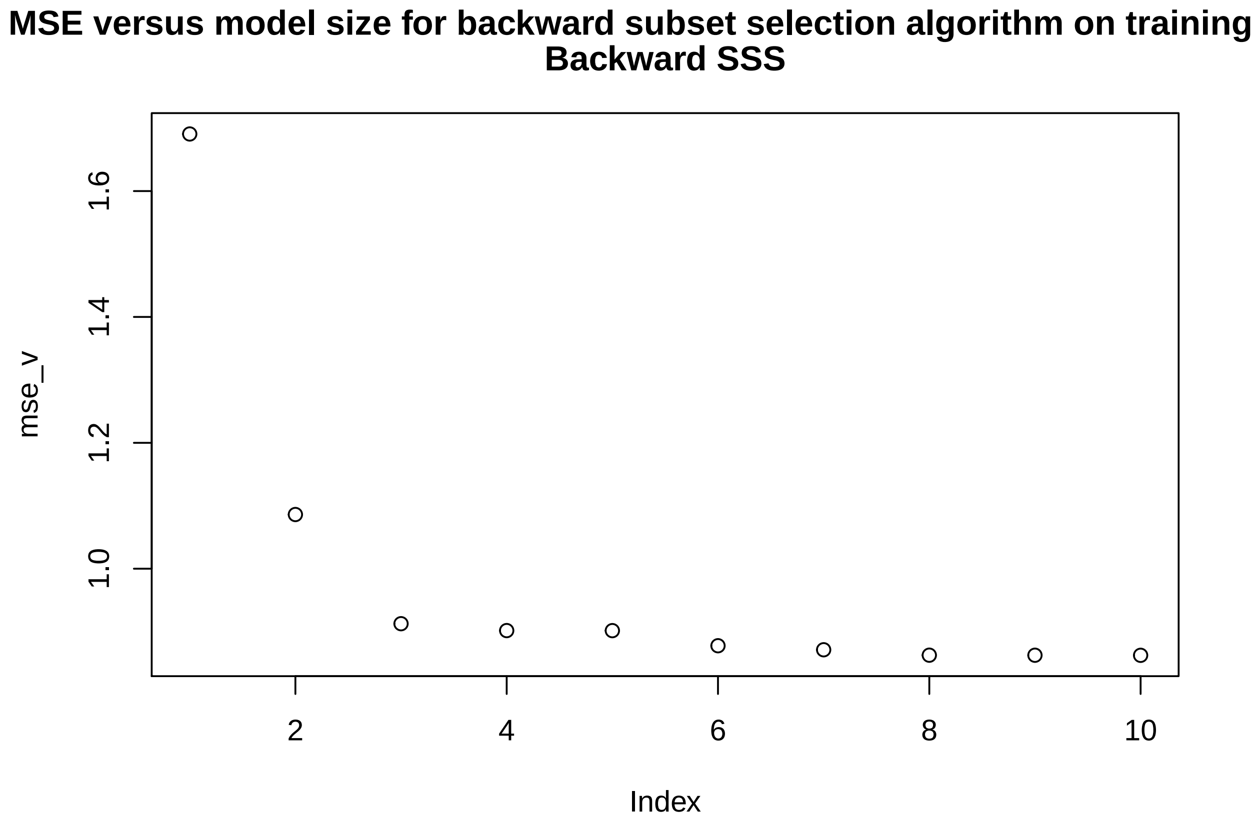
The C_p statistic indicates that the best model contains $Intercept, X, X^2, X^3$



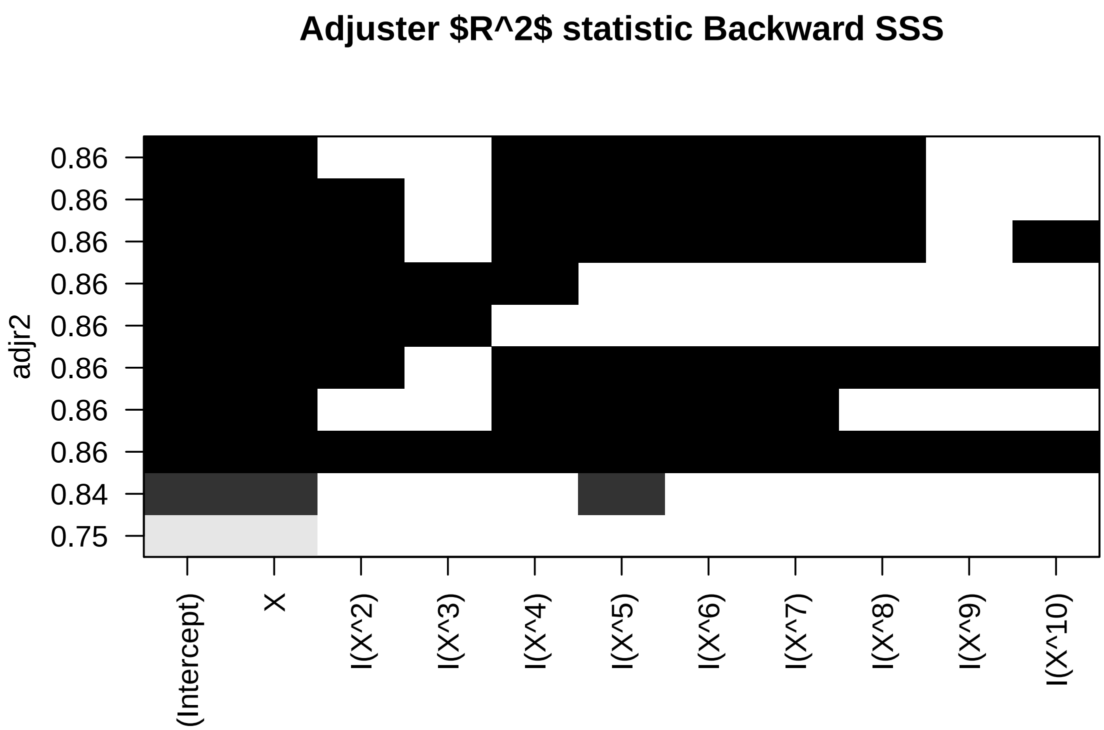
The BIC statistic indicates that the best model contains $Intercept, X, X^2, X^3$

BACKWARD SSS

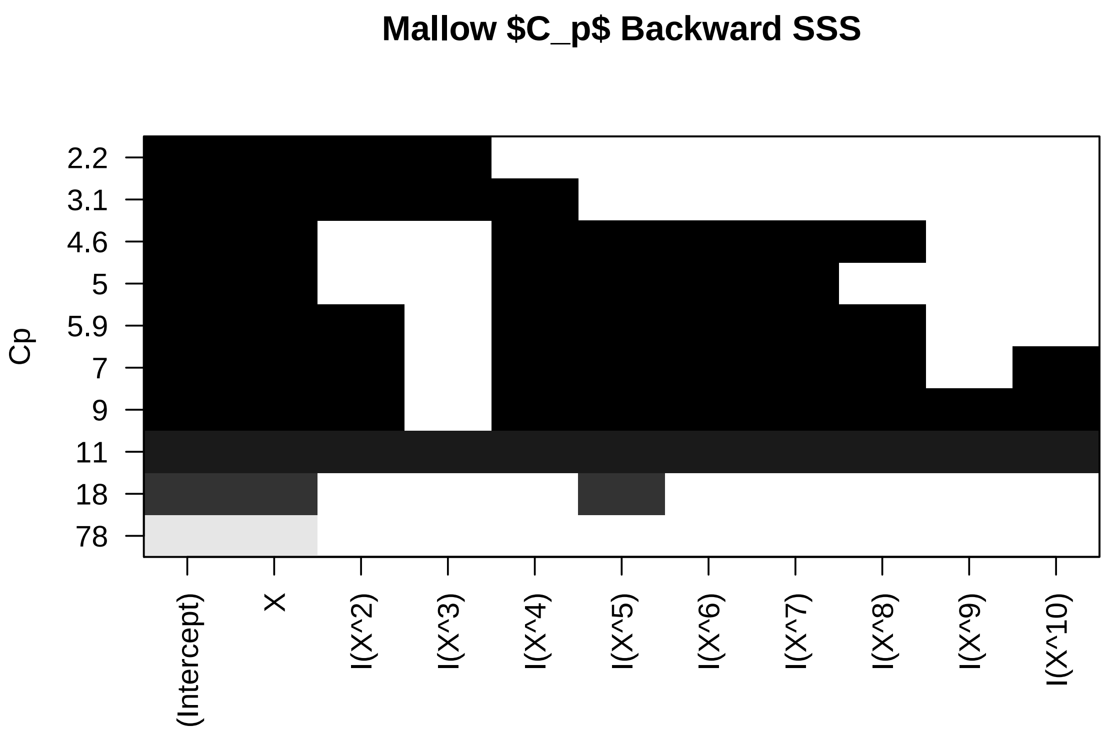
```
regfit.full <- regsubsets(Y ~ X + I(X^2) + I(X^3) + I(X^4) + I(X^5) + I(X^6) +
    I(X^7) + I(X^8) + I(X^9) + I(X^10), data = DF, nvmax = 10, method = "backward")

reg.summary <- summary(regfit.full)

mse_v <- reg.summary$rss/nrow(DF)

plot(mse_v)
title(c("MSE versus model size for backward subset selection algorithm on training set.",
    "Backward SSS"))
```



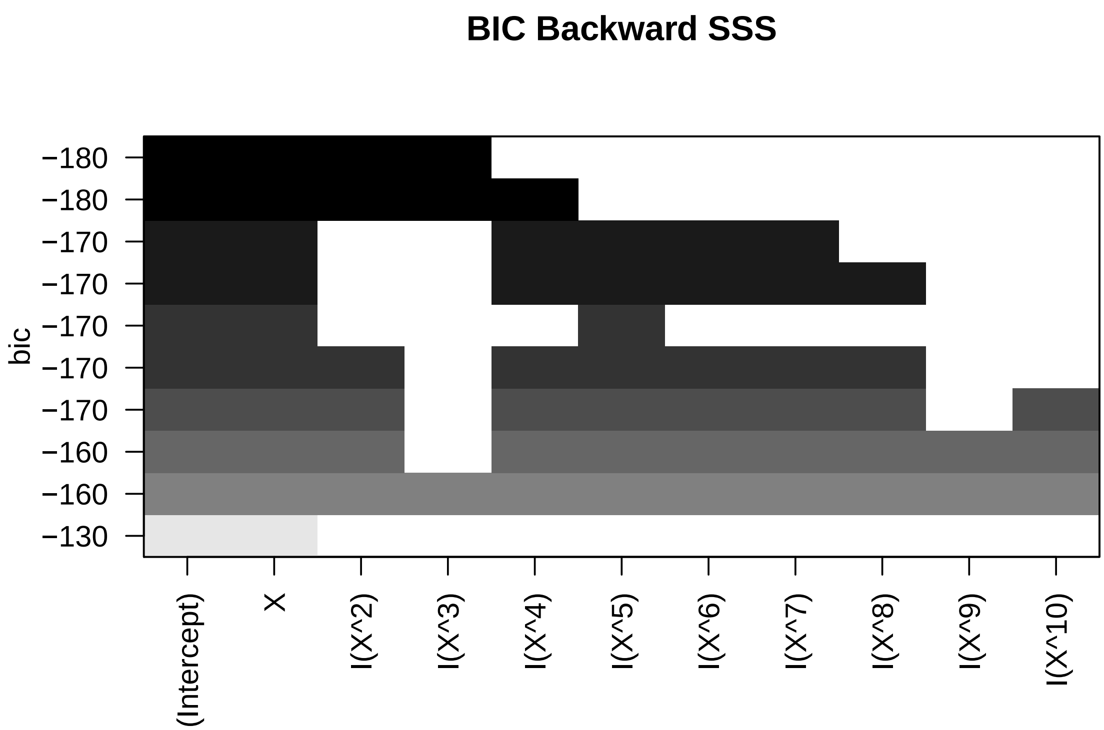
We see that there is a sharp drop in the training set MSE until 2 predictors are included and there is a steady decrease as additional polynomial terms are included. This is attributed to over fitting to the training data.



We need to remember that with R^2 statistic the values closer to 1 are a better fit. Among those with a value of 0.86 we see the full model $(Intercept), X, X^2, X^3$ has been selected.



The C_p statistic indicates that the best model contains $(Intercept), X, X^2, X^3$

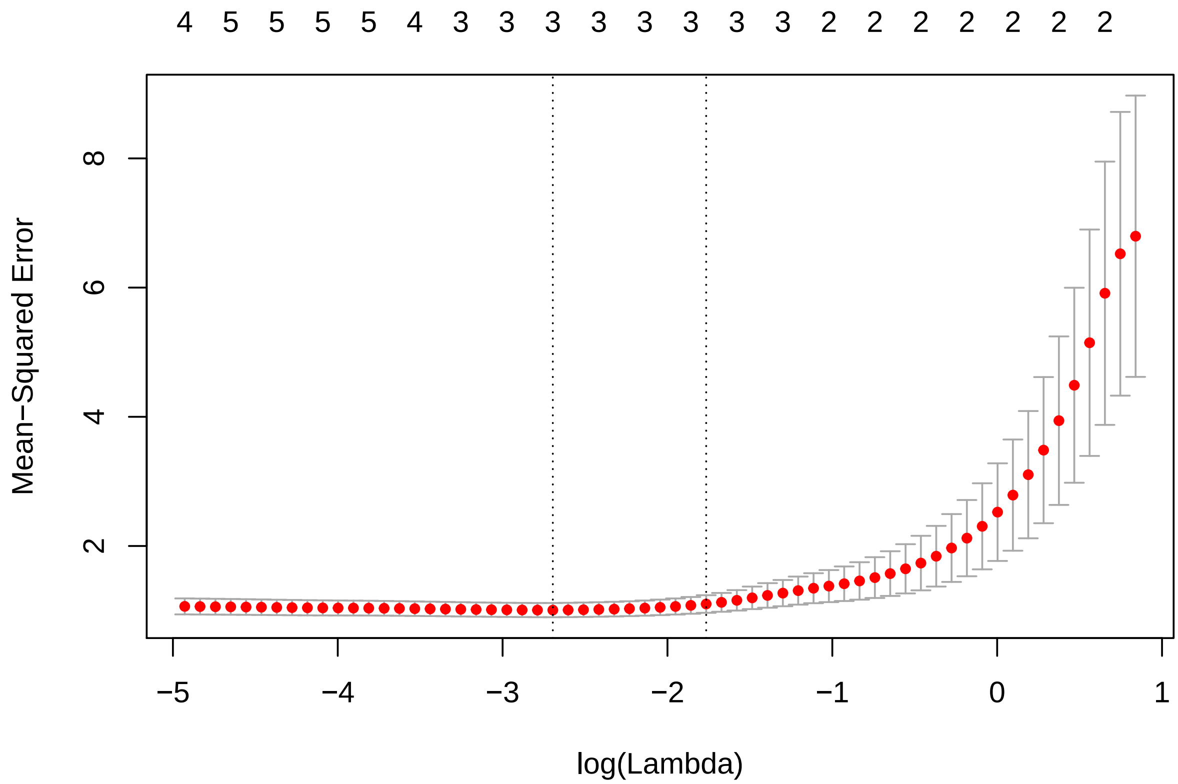


The BIC statistic indicates that the best model contains $(Intercept), X, X^2, X^3$

e)

Now fit a lasso model to the simulated data, again using $X, X^2, ..., X^10$ 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.

```
library(glmnet)
x_lasso = 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), DF)[, -1]
y_lasso = DF$Y
cv.out = cv.glmnet(x_lasso, y_lasso, alpha = 1)
plot(cv.out)
```




```r
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 0.06750938
```

```r
best_lasso = glmnet(x_lasso, y_lasso, alpha = 1, lambda = bestlam)
predict(best_lasso, type = "coefficients", s = bestlam)[1:10, ]
```

```
## (Intercept)           X      I(X^2)      I(X^3)      I(X^4)      I(X^5)
##   2.1157969   1.1876459  -0.2813453   0.5440925   0.0000000   0.0000000
##      I(X^6)      I(X^7)      I(X^8)      I(X^9)
##   0.0000000   0.0000000   0.0000000   0.0000000
```

We see that the lasso for a value of lambda given by cross validation has driven all the extra model coedfficeints to 0 as expected.

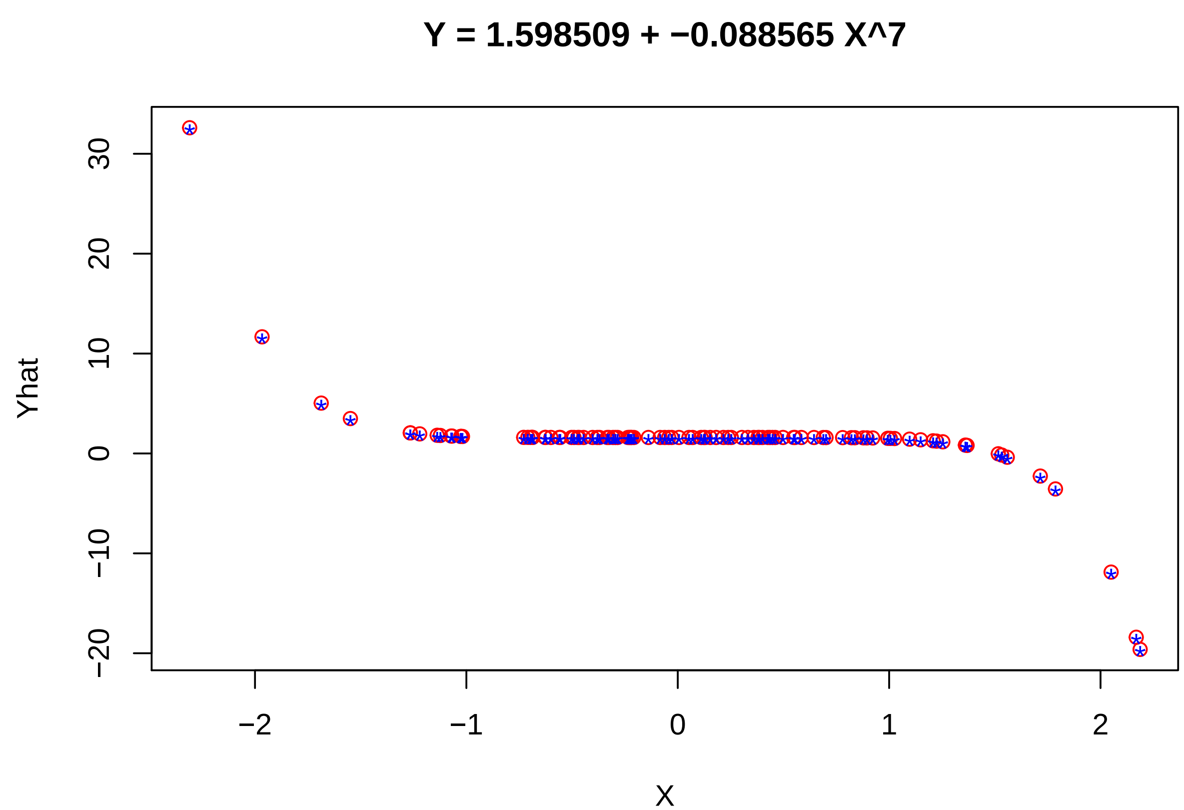
f)

Now generate a response vector Y according to the model

$$Y = \beta_0 + \beta_7 X_7 + \epsilon$$

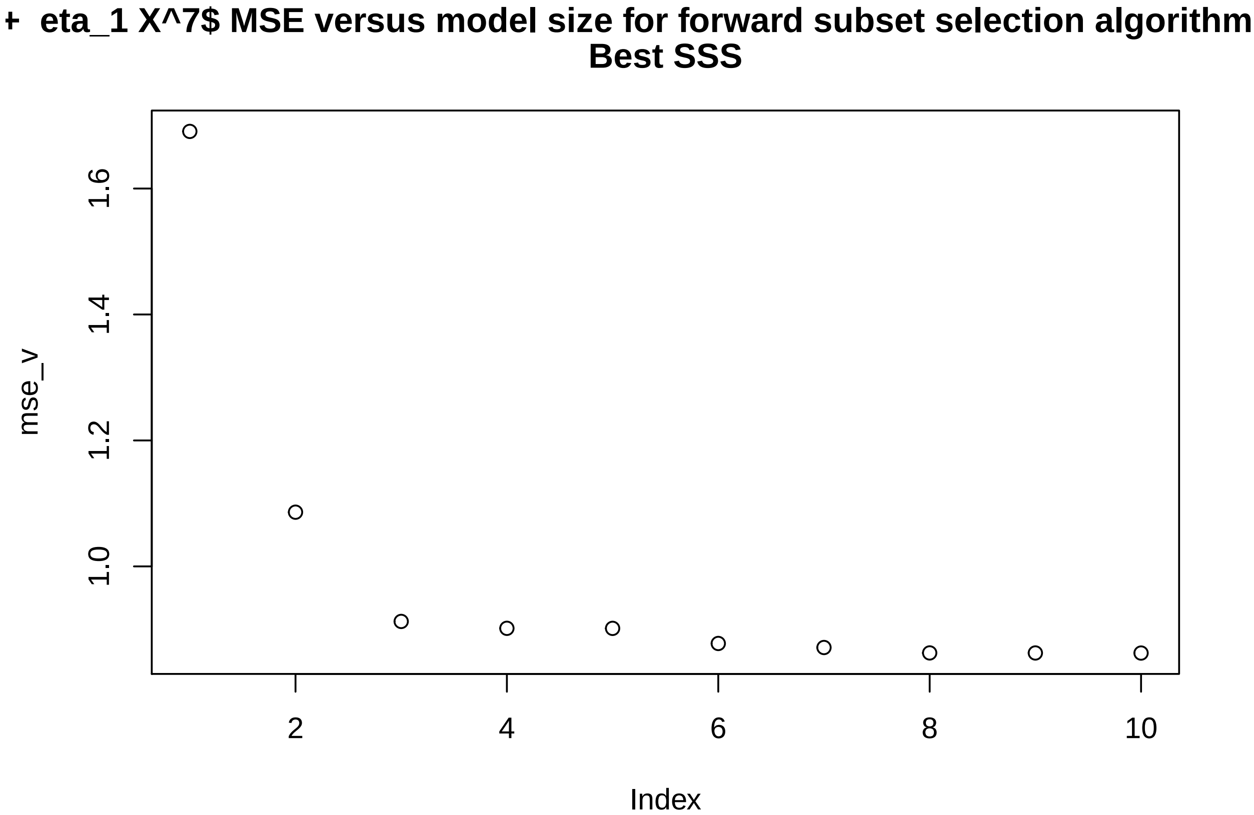
, and perform best subset selection and the lasso. Discuss the results obtained.

```
beta = rnorm(2, mean = 0, sd = 1)
Y <- matrix(NA, nrow = 100, ncol = 1)
Yhat <- matrix(NA, nrow = 100, ncol = 1)

for (i in 1:100) {
    Y[i] = beta[1] + beta[2] * X[i]^7

    Yhat[i] = beta[1] + beta[2] * X[i]^7
}

plot(X, Yhat, col = "red")
points(X, Y, pch = "*", col = "blue")
title(main = sprintf("Y = %f + %f X^7", beta[1], beta[2]), cex = 4.6)
```

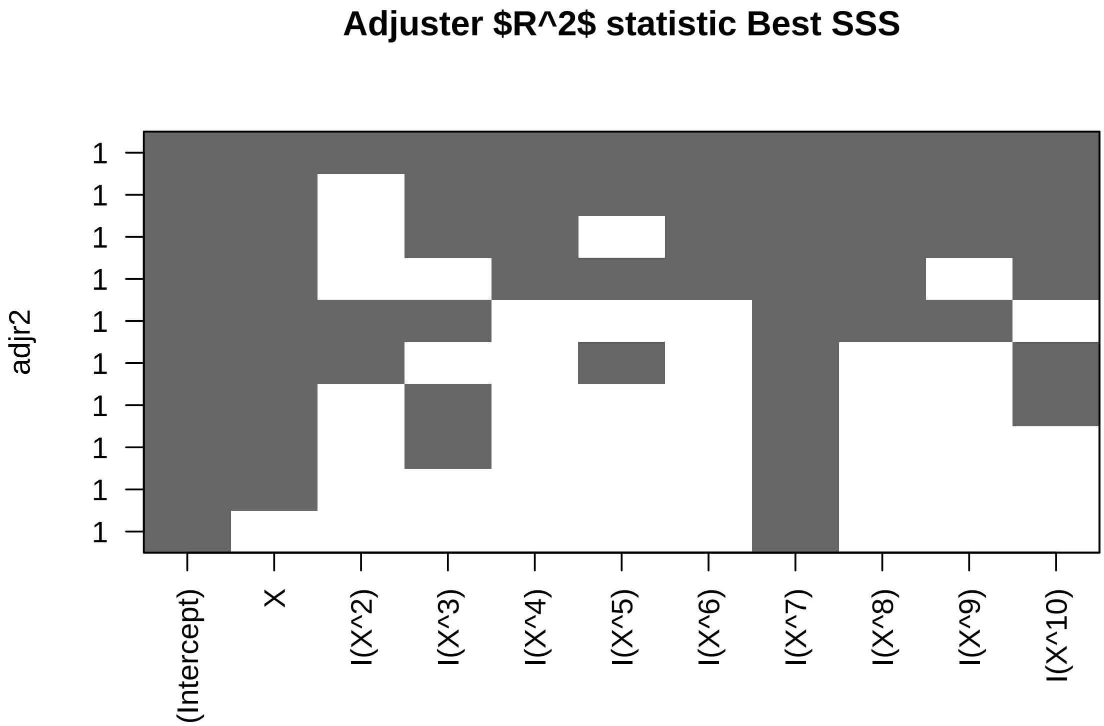



```
DF <- as.data.frame(X)
DF <- cbind(DF, Yhat)
names(DF) = c("X", "Y")
library(leaps)

regfit.full <- 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 = DF, nvmax = 10)

reg.summary <- summary(regfit.full)
```

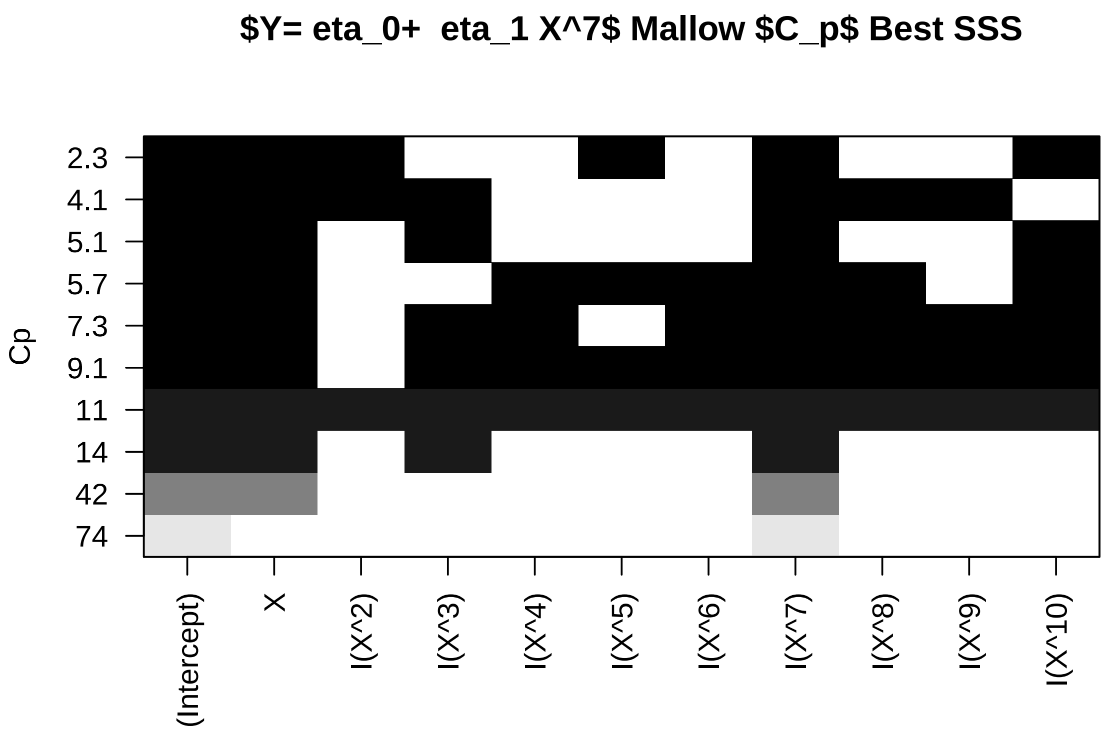
```
plot(mse_v)
title(c("$Y=\beta_0+ \beta_1 X^7$ MSE versus model size for forward subset selection algorithm on trainin
    "Best SSS"))
```



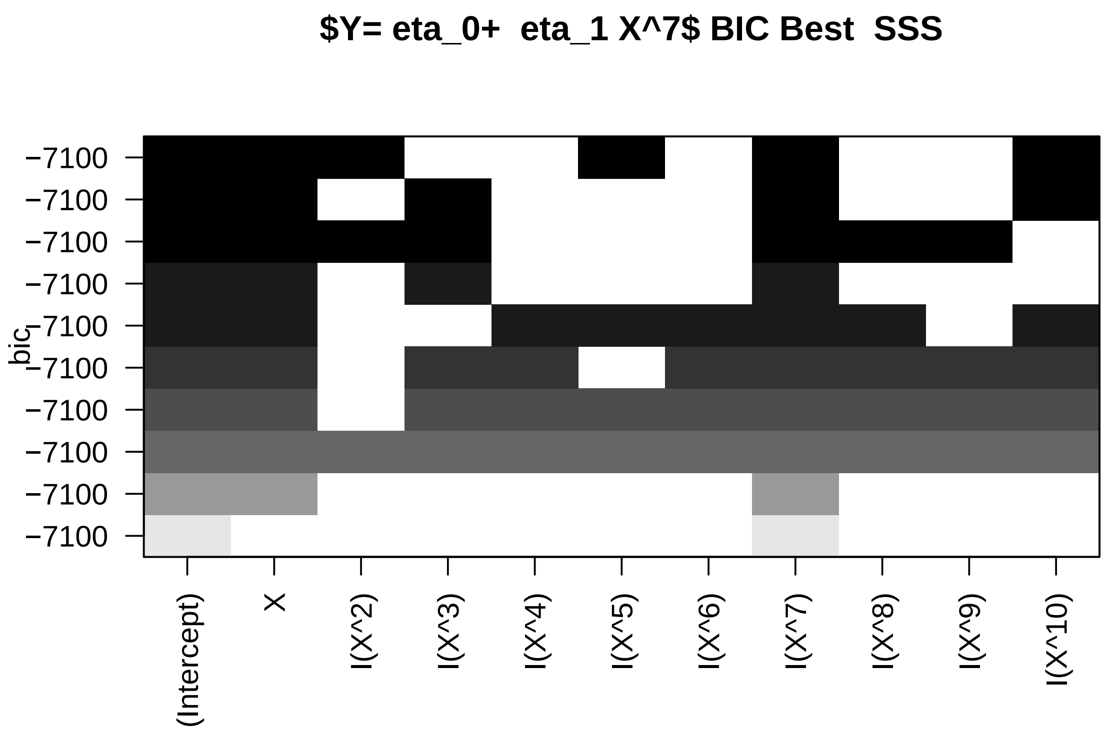

```
plot(regfit.full, scale = "adjr2")
title("Adjuster $R^2$ statistic Best SSS")
```



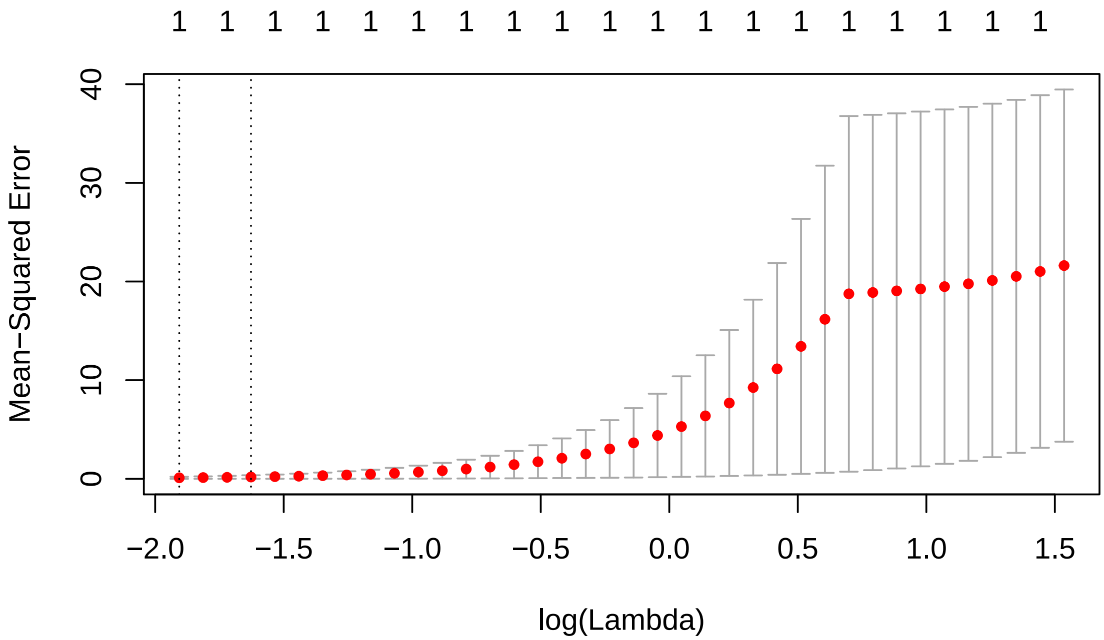

```
plot(regfit.full, scale = "Cp")
title("$Y=\beta_0+ \beta_1 X^7$ Mallow $C_p$ Best SSS")
```




```r
plot(regfit.full, scale = "bic")
title("$Y=\beta_0+ \beta_1 X^7$ BIC Best  SSS")
```




```
x_lasso = 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), DF)[, -1]
y_lasso = DF$Y
cv.out = cv.glmnet(x_lasso, y_lasso, alpha = 1)
plot(cv.out)
```




```r
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 0.1486221
```

```r
best_lasso = glmnet(x_lasso, y_lasso, alpha = 1, lambda = bestlam)
predict(best_lasso, type = "coefficients", s = bestlam)[1:10, ]
```

```
##   (Intercept)             X        I(X^2)        I(X^3)        I(X^4)
##  1.590678e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##        I(X^5)        I(X^6)        I(X^7)        I(X^8)        I(X^9)
##  0.000000e+00  0.000000e+00 -8.557615e-02  0.000000e+00 -2.777786e-05
```

All subset selection methods select the 7th term, but none have just that term and the intercept. The adjusted RSS statistic is confused, we suspect because the coefficient for X^7 (randomly generated) was small. The other statistics for Best subset models include additional terms. The lasso with a regularization parameter chose by cross validation comes very close to correctly selecting the model $Y = \beta_0 + \beta_1 X^7$. There is a X^9 term with a very small coefficient.

We note that this model may be difficult to fit since the range of the predictor is close to [-1,1] where a high order term like x^7 is relatively constant.