Assignment 4

Question 1

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.

Solution:

```
set.seed(1)
X <- rnorm(100)
eps <- rnorm(100)
```

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

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

Solution:

We assume that $\beta_0 = 1, \beta_1 = 2, \beta_2 = 3$, and $\beta_3 = 4$

```
# Select the coefficients
b0 <- 1
b1 <- 2
b2 <- 3
b3 <- 4

# Generate a response vector Y
Y <- b0 + b1*X + b2*X^2 + b3*X^3 + eps

length(Y)

## [1] 100
```

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.

Solution:

```
library(leaps)
# Create dataframe
df <- data.frame(Y = Y, X = X)

# Perform best subset selection to choose the best model
best.sub <- 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)
summary(best.sub)

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

Cp, BIC, and Adjusted R^2

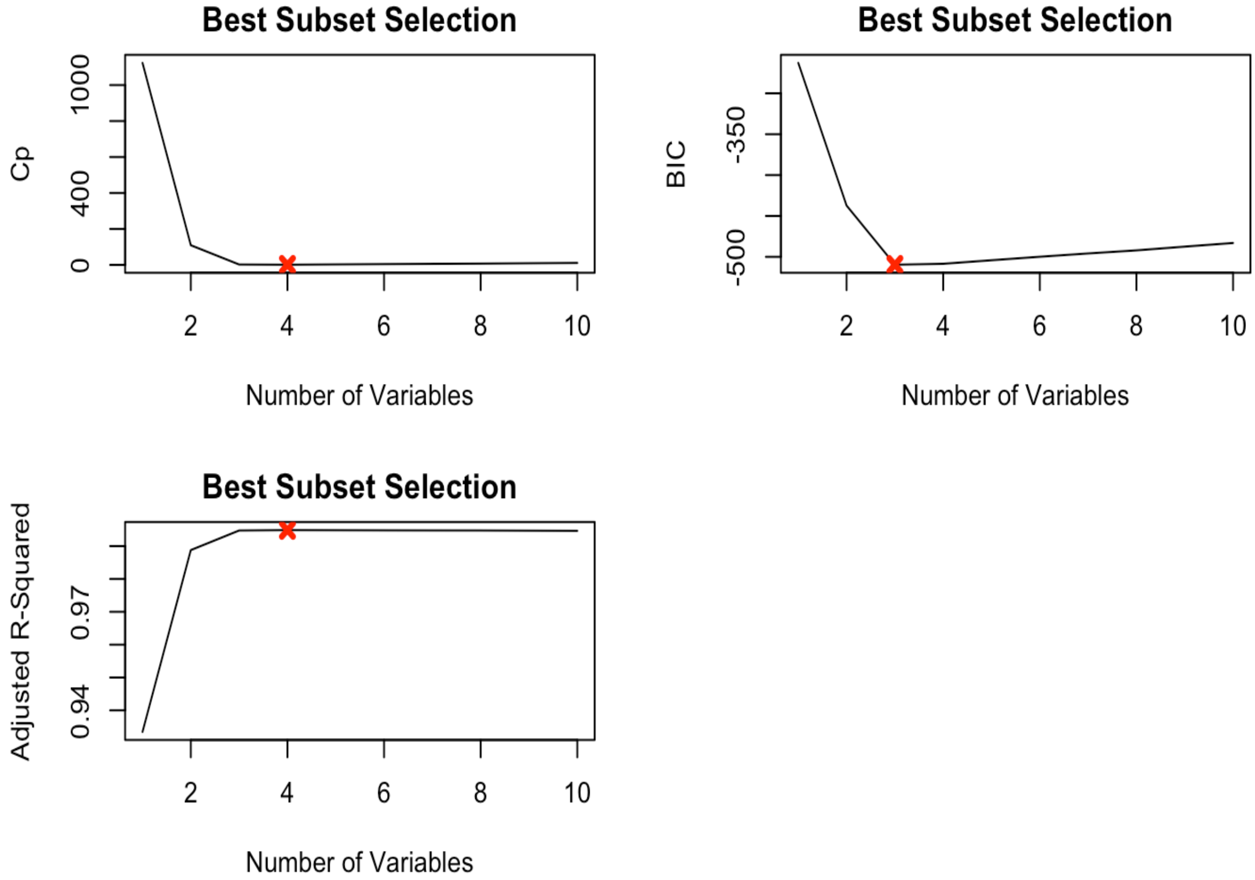
```
# Obtain Cp, BIC, and adjusted R2
sum <- summary(best.sub)
test.error <- data.frame(
  Cp = which.min(sum$cp),
  BIC = which.min(sum$bic),
  Adj.R2 = which.max(sum$adjr2)
)
print(test.error)

##   Cp BIC Adj.R2
## 1  4   3      4
```

We found that:

- The best model using the **lowest C_p** is the model includes **4 predictors**.

- The best model using the **lowest BIC** is the model includes **3 predictors**.
- The best model using the **highest Adjusted R^2** is the model includes **4 predictors**.

```
par(mfrow=c(2,2))
plot(sum$cp, xlab="Number of Variables", ylab="Cp", pch=20, type="l", main =
"Best Subset Selection")
points(4, sum$cp[4], pch=4, col="red", lwd=3)
plot(sum$bic, xlab="Number of Variables", ylab="BIC", pch=20, type="l", main
= "Best Subset Selection")
points(3, sum$bic[3], pch=4, col="red", lwd=3)
plot(sum$adjr2, xlab="Number of Variables", ylab="Adjusted R-Squared", pch=20
, type="l", main  = "Best Subset Selection")
points(4, sum$adjr2[4], pch=4, col="red", lwd=3)
```



We found that with C_p, BIC, and adjusted R^2, the best model contains set of 4, 3, and 4 predictors, respectively. Therefore, we will choose the model using the lowest C_p to obtain the coefficients of the best model.

```
# The coefficients of the best model
best.mod <- coef(best.sub,4)
print(best.mod)

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

The best subset is set of 4 predictors, which includes the predictors X, X^2, X^3, and X^5. We can write the formula of the selected model using best subset as follow:

$$\mathbf{1.072 + 2.3875X + 2.8458X^2 + 3.5580^3 + 0.0807X^5}$$

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

Solution:

Forward Stepwise Selection

```
library(leaps)
# Perform forward stepwise selection to choose the best model
forward <- 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")
summary(forward)

## 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, 
##     method = "forward")
## 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: forward
##           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)
```

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

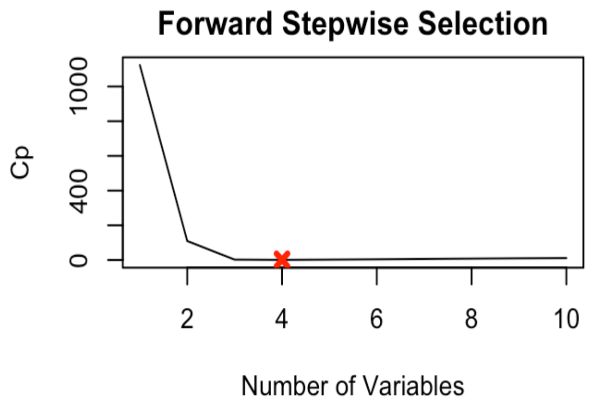
Cp, BIC, and Adjusted R^2

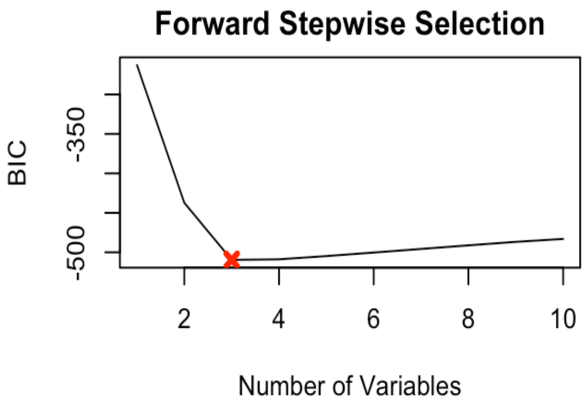
```
# Obtain Cp, BIC, and adjusted R2
sum <- summary(forward)
test.error <- data.frame(
  Cp = which.min(sum$cp),
  BIC = which.min(sum$bic),
  Adj.R2 = which.max(sum$adjr2)
)
print(test.error)

##   Cp BIC Adj.R2
## 1  4   3      4
```

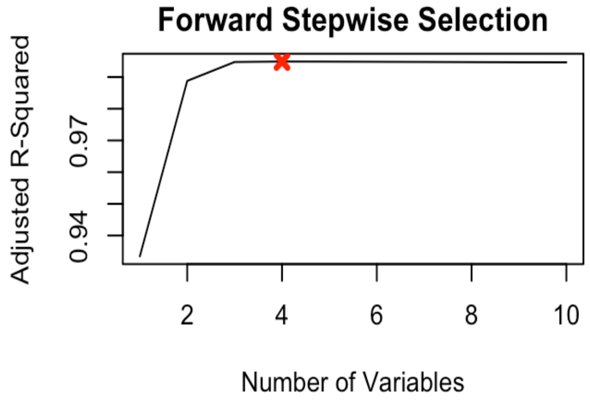
We found that:

- The best model using the **lowest C_p** is the model includes **4 predictors**.
- The best model using the **lowest BIC** is the model includes **3 predictors**.
- The best model using the **highest Adjusted R^2** is the model includes **4 predictors**.

```
par(mfrow=c(2,2))
plot(sum$cp, xlab="Number of Variables", ylab="Cp", pch=20, type="l", main =
"Forward Stepwise Selection")
points(4, sum$cp[4], pch=4, col="red", lwd=3)
plot(sum$bic, xlab="Number of Variables", ylab="BIC", pch=20, type="l", main
= "Forward Stepwise Selection")
points(3, sum$bic[3], pch=4, col="red", lwd=3)
plot(sum$adjr2, xlab="Number of Variables", ylab="Adjusted R-Squared", pch=20
, type="l", main  = "Forward Stepwise Selection")
points(4, sum$adjr2[4], pch=4, col="red", lwd=3)
```







We found that with C_p, BIC, and adjusted R^2, the best model contains set of 4, 3, and 4 predictors, respectively. Therefore, we will choose the model using the lowest C_p to obtain the coefficients of the best model.

```
# The coefficients of the best model
best.mod <- coef(forward,4)
print(best.mod)

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

The best subset is set of 4, which includes predictors X, X^2, X^3, and X^5. We found that the results are the same as using the best subset selection method. Therefore, we can write the formula of the selected model using forward stepwise selection as follow:

$$\mathbf{1.072 + 2.3875X + 2.8458X^2 + 3.5580^3 + 0.0807X^5}$$

Backward Stepwise Selection

```
library(leaps)
# Perform backward stepwise selection to choose the best model
backward <- 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")
summary(backward)
```

```
## 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, 
##     method = "backward")
## 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: backward
##           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)
## 1  ( 1 )  " " " "    "*"    " "    " "    " "    " "    " "    " "    " "
## 2  ( 1 )  " " "*"    "*"    " "    " "    " "    " "    " "    " "    " "
## 3  ( 1 )  "*" "*"    "*"    " "    " "    " "    " "    " "    " "    " "
## 4  ( 1 )  "*" "*"    "*"    " "    " "    " "    " "    " "    "*"    " "
## 5  ( 1 )  "*" "*"    "*"    " "    " "    " "    " "    "*"    "*"    " "
## 6  ( 1 )  "*" "*"    "*"    " "    " "    " "    " "    "*"    "*"    "*"
## 7  ( 1 )  "*" "*"    "*"    " "    " "    "*"    " "    "*"    "*"    "*"
## 8  ( 1 )  "*" "*"    "*"    "*"    " "    "*"    " "    "*"    "*"    "*"
## 9  ( 1 )  "*" "*"    "*"    "*"    "*"    "*"    " "    "*"    "*"    "*"
## 10  ( 1 ) "*" "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
```

Cp, BIC, and Adjusted R^2

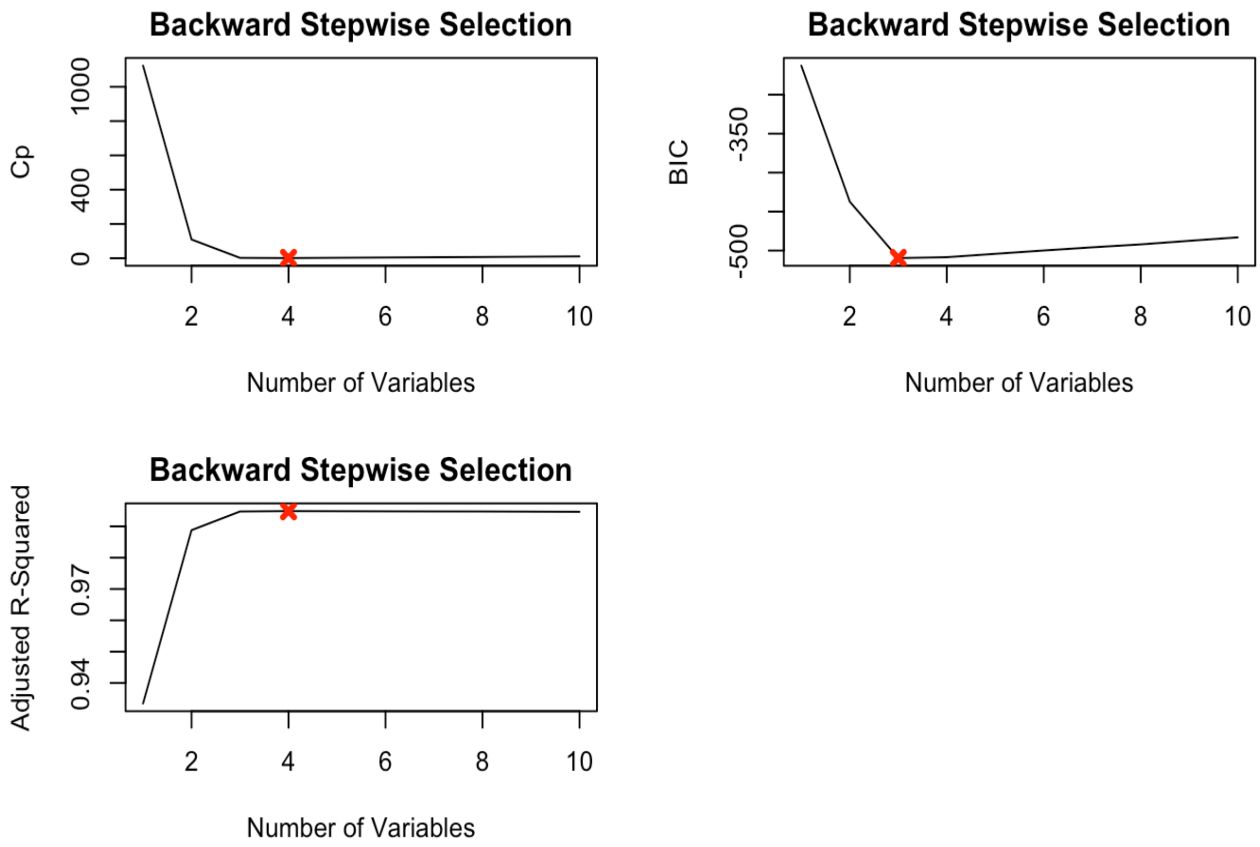
```
# Obtain Cp, BIC, and adjusted R2
sum <- summary(backward)
test.error <- data.frame(
  Cp = which.min(sum$cp),
  BIC = which.min(sum$bic),
  Adj.R2 = which.max(sum$adjr2)
)
print(test.error)
```

```
##   Cp BIC Adj.R2
## 1  4   3      4
```

We found that:

- The best model using the **lowest C_p** is the model includes **4 predictors**.
- The best model using the **lowest BIC** is the model includes **3 predictors**.
- The best model using the **highest Adjusted R^2** is the model includes **4 predictors.**

```
par(mfrow=c(2,2))
plot(sum$cp, xlab="Number of Variables", ylab="Cp", pch=20, type="l", main =
"Backward Stepwise Selection")
points(4, sum$cp[4], pch=4, col="red", lwd=3)
plot(sum$bic, xlab="Number of Variables", ylab="BIC", pch=20, type="l", main
= "Backward Stepwise Selection")
points(3, sum$bic[3], pch=4, col="red", lwd=3)
plot(sum$adjr2, xlab="Number of Variables", ylab="Adjusted R-Squared", pch=20
, type="l", main  = "Backward Stepwise Selection")
points(4, sum$adjr2[4], pch=4, col="red", lwd=3)
```



We found that with C_p, BIC, and adjusted R^2, the best model contains set of 4, 3, and 4 predictors, respectively. Therefore, we will choose the model using the lowest C_p to obtain the coefficients of the best model.

```
# The coefficients of the best model
best.mod <- coef(backward,4)
print(best.mod)

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

The best subset is set of 4 predictors, which includes predictors X, X^2, X^3, and X^9. The results are different than the best subset selection and forward stepwise selection methods. Therefore, we can write the formula of the selected model using backward stepwise selection as follow:

$$\mathbf{1.0792 + 2.2319X + 2.8335X^2 + 3.1896X^3 + 0.0013X^9}$$

Conclusion:

According to the results of three methods, we can observe that the best subset of all methods is the set of 4, based on the lowest C_p. The best subset selection and forward stepwise selection mothods chose the same predictors which are X, X^2, X^3 and X^5 to be the best model. However, with the backward stepwise selection method, the best subset contains X, X^2, X^3 and X^9.

(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.

Solution:

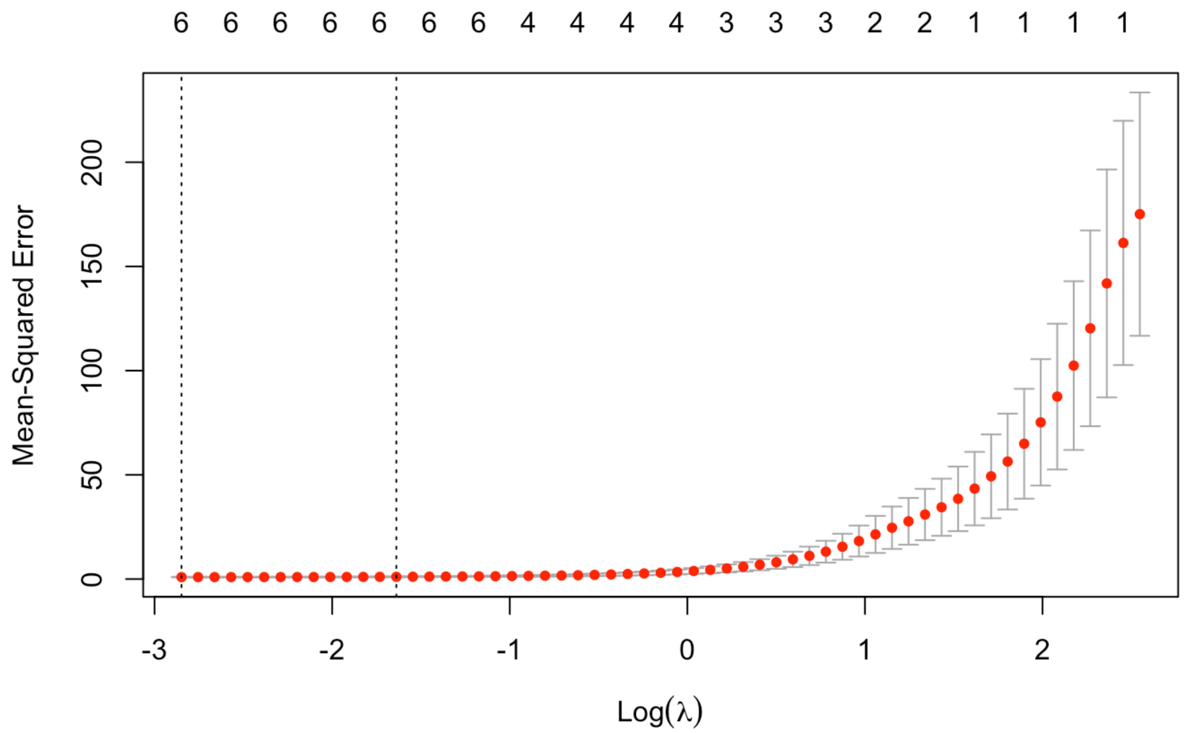
Select the optimal value of λ

```
set.seed(1)
library(glmnet)

X <- 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 = df)[,-1]
Y <- df$Y

# Using cross validation to choose the best lambda
cv <- cv.glmnet(X,Y, alpha = 1)
bestlam <- cv$lambda.min
sprintf('%s is %f', 'The best lambda obtained from the cross validation', bes
tlam)

## [1] "The best lambda obtained from the cross validation is 0.057947"
```

From the result, the optimal value of λ is 0.057947.

Plotting the cross-validation error as a function of lambda

```
# The cross-validation error as a function of lambda plot
plot(cv)
```



Estimate the coefficients

```
# Estimate the coefficients
predict(cv, s = bestlam, type = "coefficients")

## 11 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 1.168794337
## X           2.164793590
## I(X^2)      2.639485133
## I(X^3)      3.800683773
## I(X^4)      0.041512567
## I(X^5)      0.014068421
## I(X^6)      .
## I(X^7)      0.004039751
## I(X^8)      .
## I(X^9)      .
## I(X^10)     .
```

From the results, the model chosen by the lambda selected with cross-validation contains 6 predictors, including X, X^2, X^3, X^4, X^5, and X^7. We can see that the coefficients for X^6, X^8, X^9, and X^{10} have been shrunk to exactly zero.

Question 2

Use College data set by using library(ISLR)

```
library(ISLR)
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

(a) Split the data set into a training set and a test set.

Solution:

```
# Split data with the ratio 50:50
set.seed(1)
split<- sample(c(rep(0, 0.5 * nrow(College)), rep(1, 0.5 * nrow(College))))
train <- College[split == 0, ]
test <- College[split == 1, ]
sprintf("%s %i observations and %i variables", "Training set includes", dim(t
rain)[1],dim(train)[2])
```

```
## [1] "Training set includes 388 observations and 18 variables"
```

```
sprintf("%s %i observations and %i variables", "Test set includes", dim(test)
[1],dim(test)[2])
```

```
## [1] "Test set includes 389 observations and 18 variables"
```

(b) Fit a linear model using least squares on the training set and report the test error obtained.

Solution:

```
# fitting a linear regression model
lm.fit <- lm(Apps ~., data = train)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Apps ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2810.6  -408.0   -40.6   302.4  7207.2
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -322.27839  556.33831  -0.579  0.56275
## PrivateYes  -816.35852  174.19871  -4.686 3.91e-06 ***
## Accept         1.29774    0.06949  18.676  < 2e-16 ***
## Enroll        -0.26248    0.31390  -0.836  0.40359
## Top10perc     56.27171    7.89874   7.124 5.51e-12 ***
## Top25perc    -17.47324    6.31355  -2.768  0.00593 **
## F.Undergrad    0.03876    0.05363   0.723  0.47029
## P.Undergrad   -0.01328    0.04925  -0.270  0.78755
## Outstate      -0.06704    0.02478  -2.705  0.00714 **
## Room.Board     0.26167    0.06495   4.029 6.81e-05 ***
## Books         -0.29101    0.30145  -0.965  0.33500
## Personal       0.09030    0.09110   0.991  0.32223
## PhD           -7.41818    5.74007  -1.292  0.19704
## Terminal      -2.13184    6.32514  -0.337  0.73628
## S.F.Ratio      4.64207   18.76702   0.247  0.80477
## perc.alumni   -6.74075    5.59616  -1.205  0.22915
## Expend         0.04345    0.01673   2.597  0.00979 **
## Grad.Rate     11.03226    3.97480   2.776  0.00579 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 962.3 on 370 degrees of freedom
## Multiple R-squared:  0.9227, Adjusted R-squared:  0.9192
## F-statistic: 259.8 on 17 and 370 DF,  p-value: < 2.2e-16
```

```
# Reporting the test error
y.pred <- predict(lm.fit, test)
y.test <- test$Apps
lm.mse <- mean((y.test - y.pred)^2)
sprintf('%s is %f', 'The test error for the linear regression model', lm.mse)
```

```
## [1] "The test error for the linear regression model is 1476669.411939"
```

From the result, the test error for the linear regression model is 1476669.411939.

(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

Solution:

```
set.seed(1)
x <- model.matrix(Apps~., train)[,-1]
y <- train$Apps

# Using cross validation to choose the best lambda
cv <- cv.glmnet(x,y, alpha = 0)
bestlam.ridge <- cv$lambda.min
sprintf('%s is %f', 'The best lambda obtained from the cross validation', bes
tlam.ridge)

## [1] "The best lambda obtained from the cross validation is 317.263577"

# Fitting a Ridge regression model on the train data with lambda chosen by cr
oss-validation
ridge.mod <- glmnet(x,y, alpha = 0, lambda = bestlam.ridge)
```

We fitted the model with the chosen lambda, 317.263577.

```
# Evaluating test error on the test data
x.test <- model.matrix(Apps~., test)[,-1]
y.pred <- predict(ridge.mod, newx = x.test)
y.test <- test$Apps
ridge.mse <- mean((y.test - y.pred)^2)
sprintf('%s is %f', 'The test error for the ridge linear regression', ridge.m
se)

## [1] "The test error for the ridge linear regression is 2337451.466287"
```

From the result, the test error for the ridge regression model is 2337451.466287.

(d) Fit a lasso model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates.

Solution:

```
set.seed(1)
x <- model.matrix(Apps~., train)[,-1]
y <- train$Apps

# Using cross validation to choose the best lambda
cv <- cv.glmnet(x,y, alpha = 1)
bestlam.lasso <- cv$lambda.min
sprintf('%s is %f', 'The best lambda obtained from the cross validation', bes
tlam.lasso)
```

```
## [1] "The best lambda obtained from the cross validation is 6.228013"

# Fitting a Lasso regression model on the train data with lambda chosen by cr
oss-validation
lasso.mod <- glmnet(x,y, alpha = 1, lambda = bestlam.lasso)
```

We fitted the model with the chosen lambda, 6.228013.

```
# Evaluating test error on the test data
x.test <- model.matrix(Apps~., test)[,-1]
y.pred <- predict(lasso.mod, newx = x.test)
y.test <- test$Apps
lasso.mse <- mean((y.test - y.pred)^2)
sprintf('%s is %f', 'The test error for the lasso model', lasso.mse)

## [1] "The test error for the lasso model is 1510013.684253"
```

From the result, the test error for the lasso model is 1510013.684253.

The number of non-zero coefficient estimates

```
lasso.est <- predict(lasso.mod,type="coefficients", s = bestlam.lasso)[1:18,]
non.zero <- length(lasso.est[lasso.est != 0])
sprintf('%s is %i', 'The number of non-zero coefficient estimates', non.zero)

## [1] "The number of non-zero coefficient estimates is 16"
```

Clearly, the number of non-zero coefficient estimates is 16 including intercept coefficient. The non-zero coefficients can be shown as follows:

```
lasso.est[lasso.est != 0]

##   (Intercept)    PrivateYes        Accept     Top10perc     Top25perc 
## -389.75131393 -816.56771166    1.26056810   50.98629943  -13.41101525 
##    F.Undergrad      Outstate    Room.Board         Books      Personal 
##     0.00354925   -0.05972068    0.25338062   -0.22549393    0.07306761 
##           PhD      Terminal     S.F.Ratio   perc.alumni        Expend 
##   -6.54403623   -2.05540489    1.38926072   -6.75479651    0.04046896 
##     Grad.Rate 
##     9.91121234
```

Conclusion:

```
library(dplyr)

data.frame(Model = c("Linear Regression", "Ridge", "Lasso"), Test.Error = c(l
m.mse,ridge.mse, lasso.mse)) %>% arrange(Test.Error)
```

```
##               Model Test.Error
## 1 Linear Regression    1476669
## 2             Lasso    1510014
## 3             Ridge    2337451
```

From the result, we can conclude that the linear regression model seems to perform the best since it has the lowest test error, which is equal to 1476669, followed by the lasso model and then the ridge regression model.