DS-6030 Homework Module 5

Tom Lever

06/24/2023

DS 6030 | Spring 2023 | University of Virginia

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 $n = 100$.

```
set.seed(2)
X <- rnorm(n = 100, mean = 0, sd = 1)
epsilon <- rnorm(n = 100, mean = 0, sd = 1*10^{-6})
X[1:3]
```

```
# [1] -0.8969145  0.1848492  1.5878453
```

```
epsilon[1:3]
```

```
# [1]  1.074459e-06  2.605978e-07 -3.142720e-07
```

   (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 β_0, β_1, β_2, β_3 are constants of your choice.

```
beta_0 <- 1
beta_1 <- 2
beta_2 <- 3
beta_3 <- 4
Y <- beta_0 + beta_1 * X + beta_2 * I(X^2) + beta_3 * I(X^3) + epsilon
Y[1:3]
```

```
# [1] -1.266573  1.497471 27.752887
```

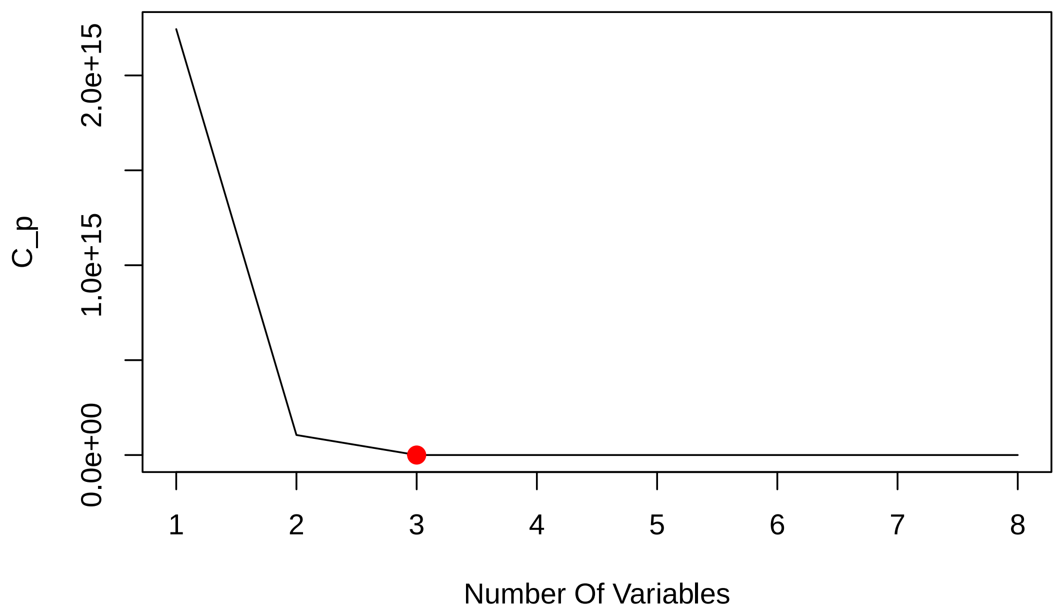
   (c) Use the `regsubsets()` function to perform best subset selection in order to choose the best model containing the predictors $X, X^2 \ldots, X^{10}$. What is the best model obtained according to Cp, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients 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.

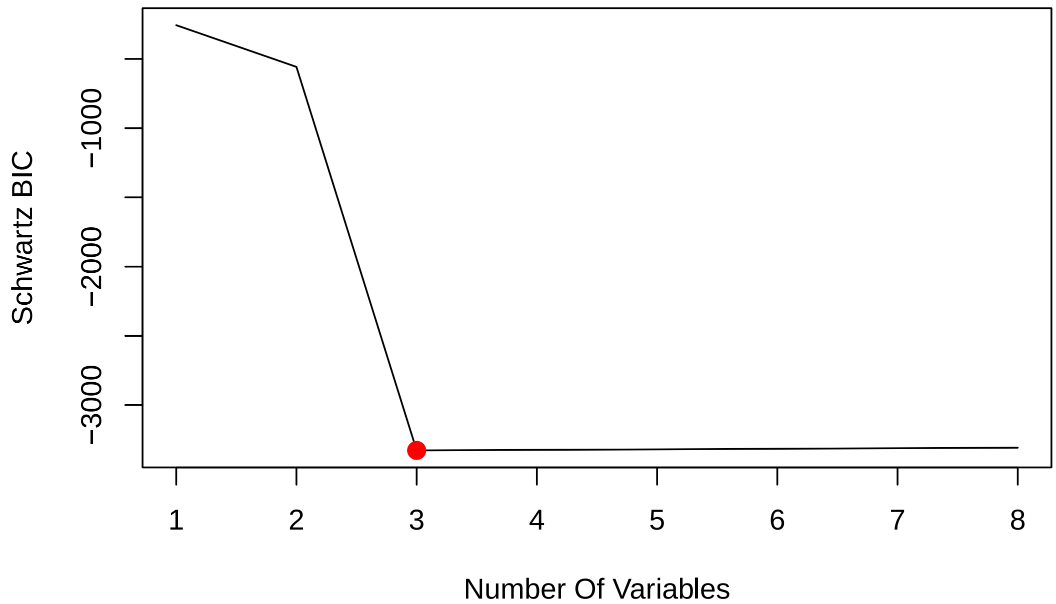
```
data_frame <- data.frame(X = X, Y = Y)
head(data_frame, n = 3)
```

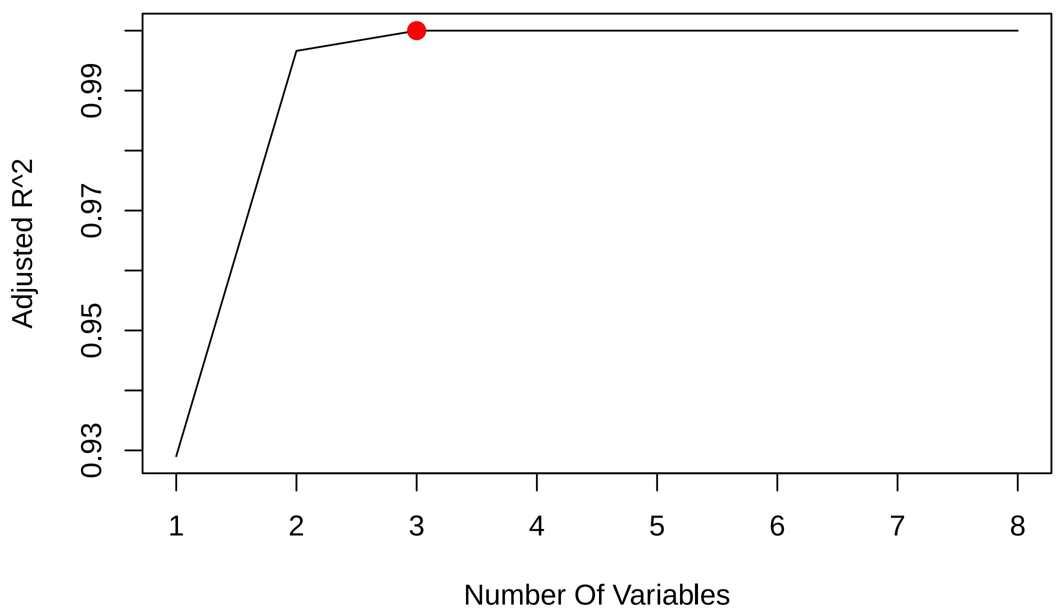
```
#           X          Y
# 1 -0.8969145 -1.26657....
# 2  0.1848492 1.497470....
# 3  1.5878453 27.75288....
```

```r
library(leaps)
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
subset_selection_object <- regsubsets(
    x = formula,
    data = data_frame,
    method = "exhaustive"
)
analyze_subset_selection_object <- function(subset_selection_object) {
 summary_for_subset_selection_object <- summary(object = subset_selection_object)
 Mallows_Cp <- summary_for_subset_selection_object$cp
 index_of_model_with_minimum_Mallows_Cp <- which.min(Mallows_Cp)
 coefficients_by_Mallows_Cp <- coef(
     subset_selection_object, index_of_model_with_minimum_Mallows_Cp
 )
 Schwartz_BIC <- summary_for_subset_selection_object$bic
 index_of_model_with_minimum_Schwartz_BIC <- which.min(Schwartz_BIC)
 coefficients_by_Schwartz_BIC <- coef(
     subset_selection_object, index_of_model_with_minimum_Schwartz_BIC
 )
 adjusted_R2 <- summary_for_subset_selection_object$adjr2
 index_of_model_with_maximum_adjusted_R2 <- which.max(adjusted_R2)
 coefficients_by_adjusted_R2 <- coef(
     subset_selection_object, index_of_model_with_maximum_adjusted_R2
 )
 plot(Mallows_Cp, xlab = "Number Of Variables", ylab = "C_p", type = "l")
 index_of_minimum_Mallows_Cp <- which.min(Mallows_Cp)
 minimum_Mallows_Cp <- Mallows_Cp[index_of_minimum_Mallows_Cp]
 points(
     index_of_minimum_Mallows_Cp,
     minimum_Mallows_Cp,
     col = "red",
     cex = 2,
     pch = 20
 )
 plot(Schwartz_BIC, xlab = "Number Of Variables", ylab = "Schwartz BIC", type = "l")
 index_of_minimum_Schwartz_BIC <- which.min(Schwartz_BIC)
 minimum_Schwartz_BIC <- Schwartz_BIC[index_of_minimum_Schwartz_BIC]
 points(
     index_of_minimum_Schwartz_BIC,
     minimum_Schwartz_BIC,
     col = "red",
     cex = 2,
     pch = 20
 )
 plot(adjusted_R2, xlab = "Number Of Variables", ylab = "Adjusted R^2", type = "l")
 index_of_maximum_adjusted_R2 <- which.max(adjusted_R2)
 maximum_adjusted_R2 <- adjusted_R2[index_of_maximum_adjusted_R2]
 points(
     index_of_maximum_adjusted_R2,
     maximum_adjusted_R2,
     col = "red",
     cex = 2,
     pch = 20
```

```
 )
 coefficients <- list(
     coefficients_by_Mallows_Cp = coefficients_by_Mallows_Cp,
     coefficients_by_Schwartz_BIC = coefficients_by_Schwartz_BIC,
     coefficients_by_adjusted_R2 = coefficients_by_adjusted_R2
 )
 return(coefficients)
}
analyze_subset_selection_object(subset_selection_object)
```








```
# $coefficients_by_Mallows_Cp
# (Intercept)           X      I(X^2)      I(X^3)
#           1           2           3           4
#
```

```
# $coefficients_by_Schwartz_BIC
# (Intercept)           X       I(X^2)      I(X^3)
#           1           2            3           4
#
# $coefficients_by_adjusted_R2
# (Intercept)           X       I(X^2)      I(X^3)
#           1           2            3           4
```

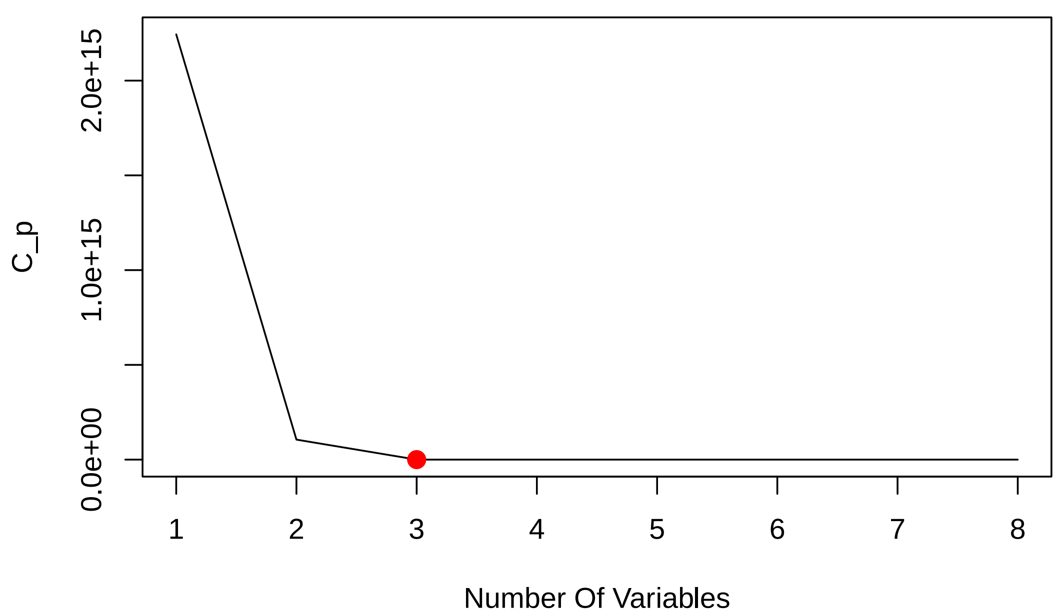
The best model obtained according to Mallow's C_p, the Schwartz Bayesian Information Criterion, and adjusted R^2 is

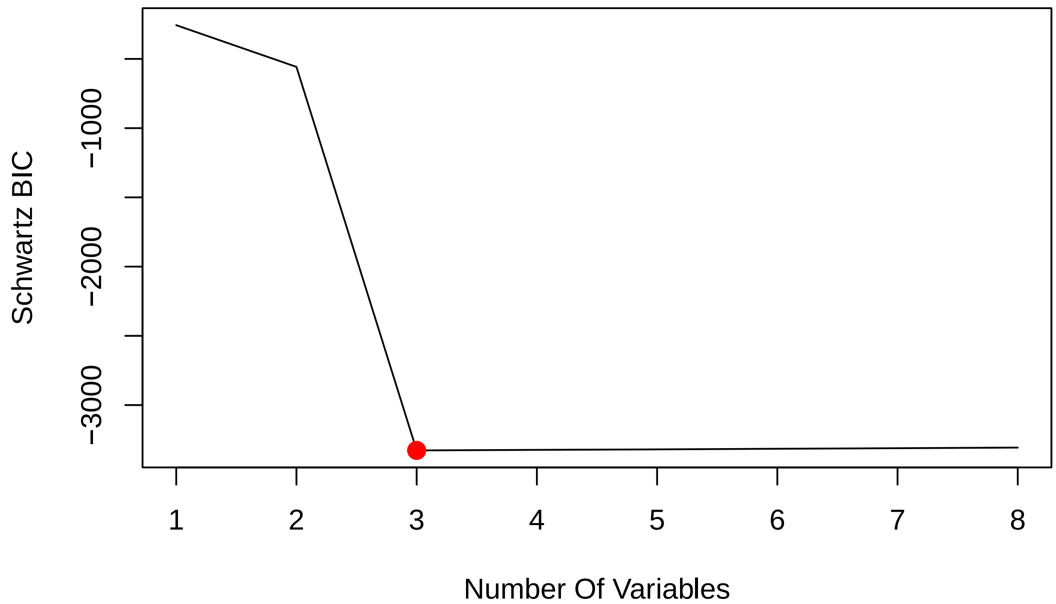
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 = 1 + 2X + 3X^2 + 4X^3$$

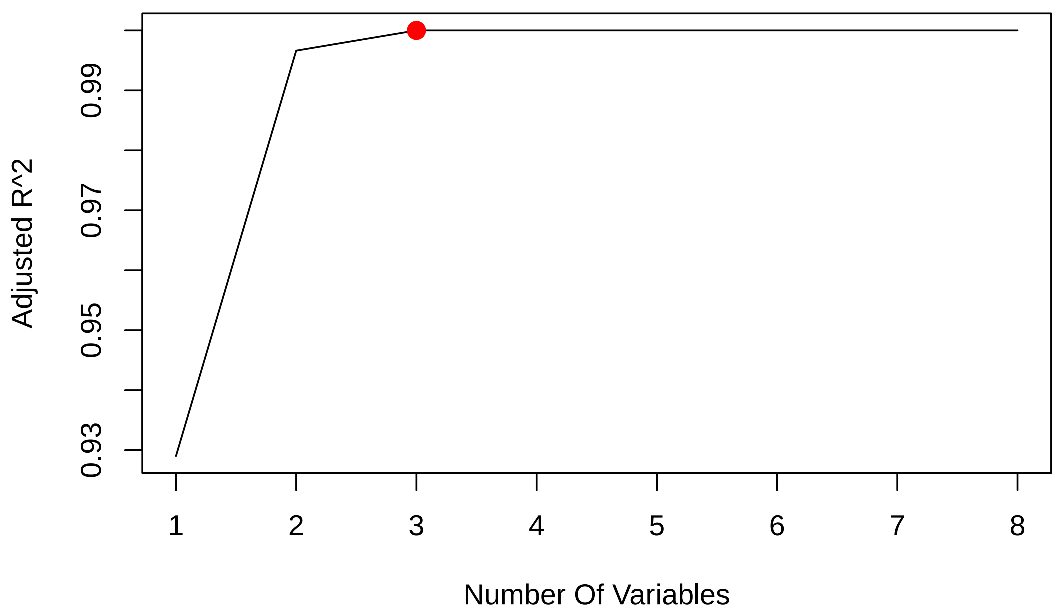
(d) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

The models chosen by forward and backward selection involve the true predictive terms and their coefficients.

```r
subset_selection_object <- regsubsets(
    x = formula,
    data = data_frame,
    method = "forward"
)
analyze_subset_selection_object(subset_selection_object)
```



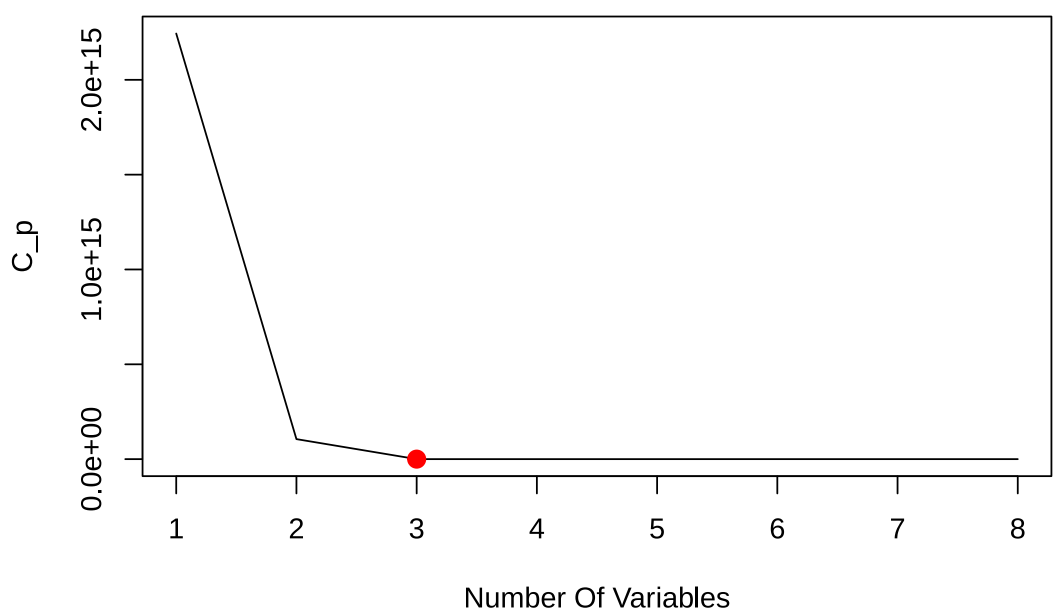


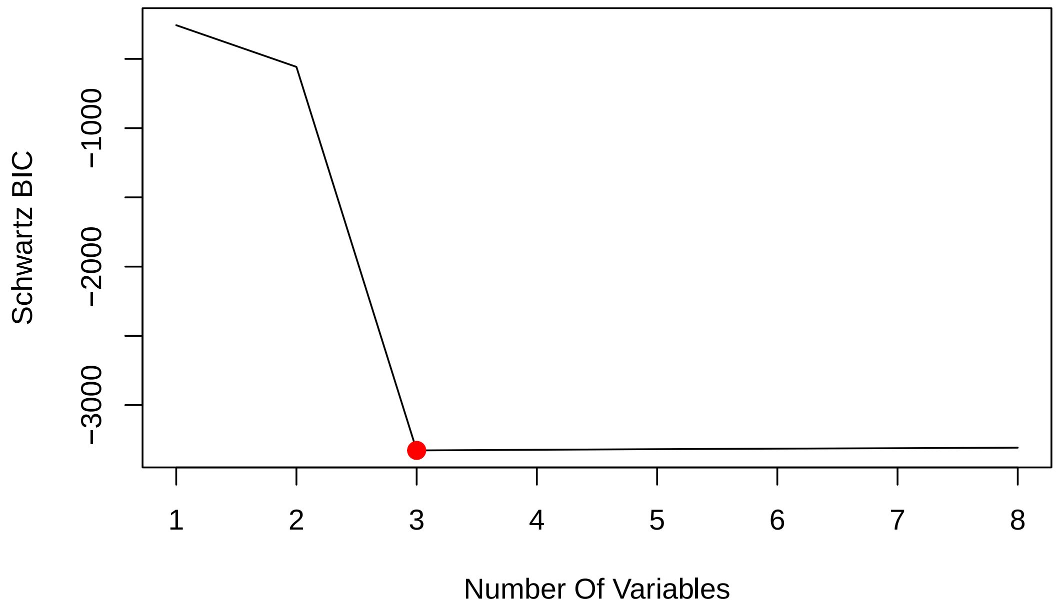


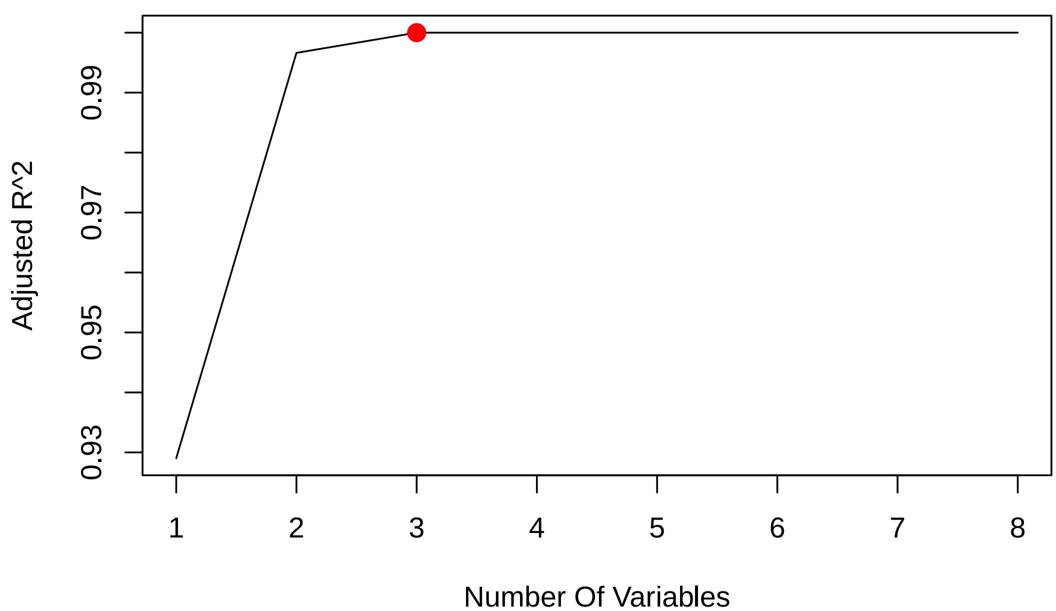

```
# $coefficients_by_Mallows_Cp
# (Intercept)           X      I(X^2)      I(X^3)
#           1           2           3           4
#
```

```
# $coefficients_by_Schwartz_BIC
# (Intercept)           X       I(X^2)       I(X^3)
#           1           2            3            4
#
# $coefficients_by_adjusted_R2
# (Intercept)           X       I(X^2)       I(X^3)
#           1           2            3            4
```

```r
subset_selection_object <- regsubsets(
    x = formula,
    data = data_frame,
    method = "backward"
)
analyze_subset_selection_object(subset_selection_object)
```








```
# $coefficients_by_Mallows_Cp
# (Intercept)           X      I(X^2)      I(X^3)
#           1           2           3           4
#
```

```
# $coefficients_by_Schwartz_BIC
# (Intercept)           X      I(X^2)      I(X^3)
#           1           2           3           4
#
# $coefficients_by_adjusted_R2
# (Intercept)           X      I(X^2)      I(X^3)
#           1           2           3           4
```

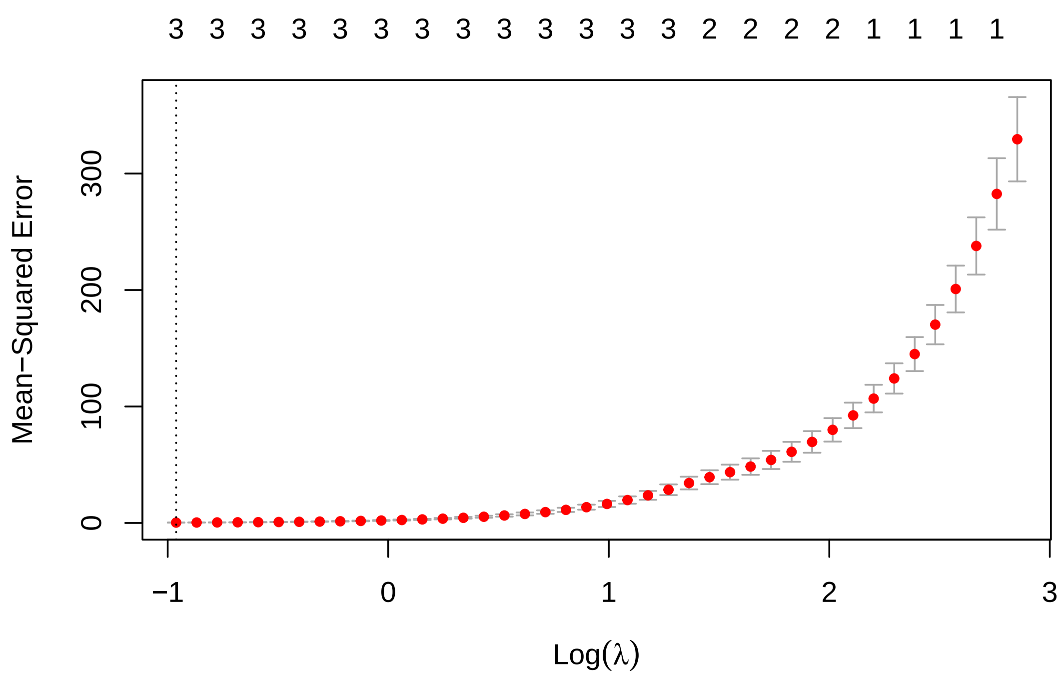
The best model obtained according to Mallow's C_p, the Schwartz Bayesian Information Criterion, and adjusted R^2 is

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 = 1 + 2X + 3X^2 + 4X^3$$

These results are identical to those in part (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.

```
full_model_matrix <- model.matrix(object = formula, data = data_frame)[, -1]
the_cv.glmnet <- glmnet::cv.glmnet(x = full_model_matrix, y = Y, alpha = 1)
plot(the_cv.glmnet)
```




```
optimal_value_of_lambda <- the_cv.glmnet$lambda.min
optimal_value_of_lambda
```

```
# [1] 0.3822143
```

```
polynomial_lasso_regression_model <- glmnet::glmnet(full_model_matrix, y = Y, alpha = 1)
predict(object = polynomial_lasso_regression_model, s = optimal_value_of_lambda, type = "coeffi
```

```
# 11 x 1 sparse Matrix of class "dgCMatrix"
#                   s1
# (Intercept) 1.364119
# X           1.990943
# I(X^2)      2.730444
# I(X^3)      3.899032
# I(X^4)      .
# I(X^5)      .
# I(X^6)      .
# I(X^7)      .
# I(X^8)      .
# I(X^9)      .
# I(X^10)     .
```

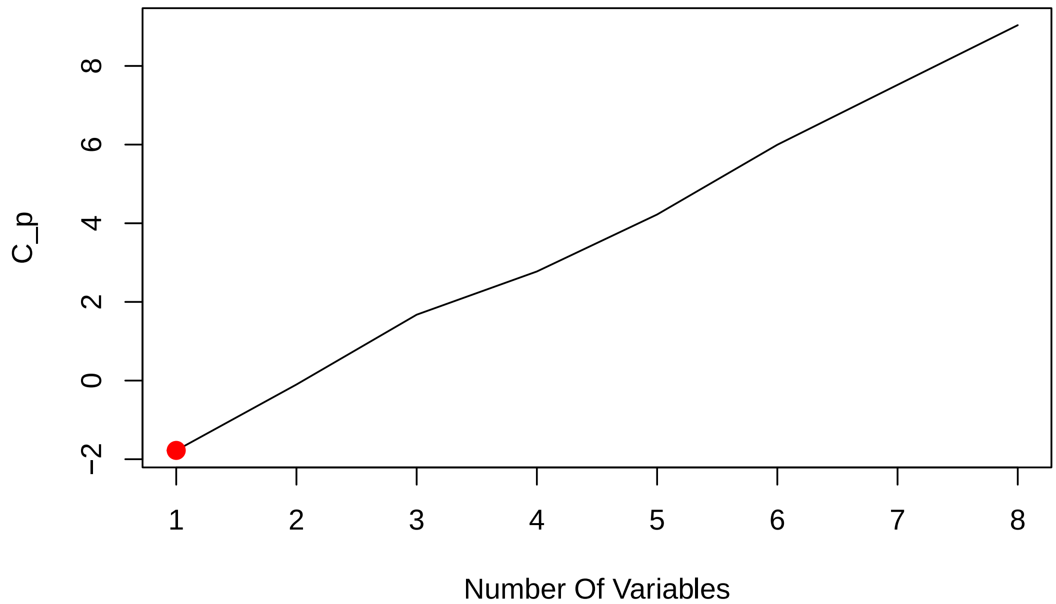
The models chosen by polynomial lasso regression involve the true predictive terms and rough approximations of their coefficients. $\beta_0 = 1.364$, $\beta_1 = 1.991$, $\beta_2 = 2.730$, and $\beta_3 = 3.899$.

(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.

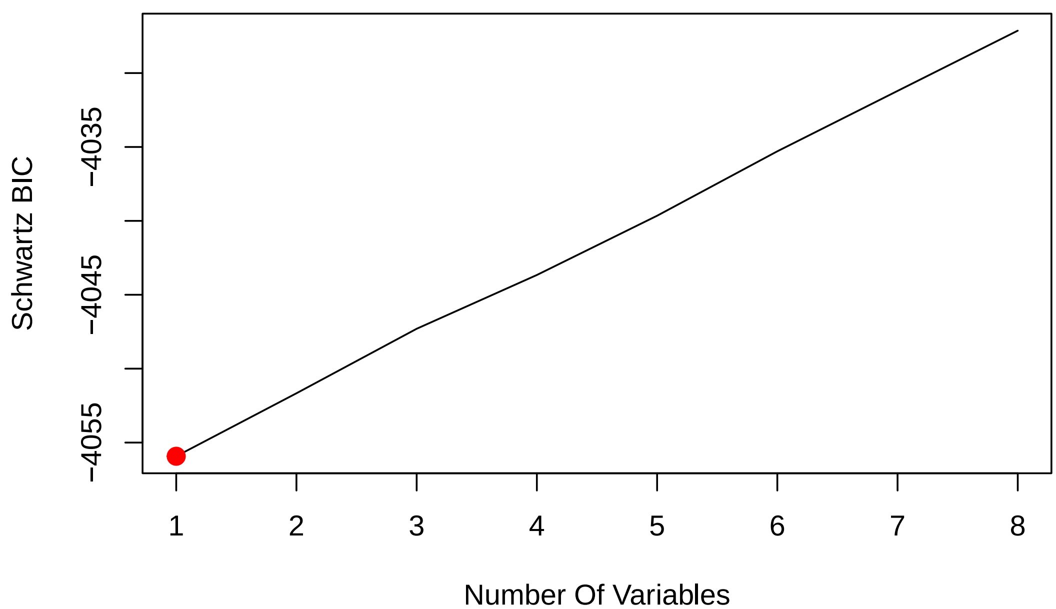
```r
beta_7 <- 8
Y <- beta_0 + beta_7 * I(X^7) + epsilon
data_frame <- data.frame(X = X, Y = Y)
head(data_frame, n = 3)
```

```
#            X            Y
# 1 -0.8969145 -2.73548....
# 2  0.1848492 1.000059....
# 3  1.5878453 204.5857....
```

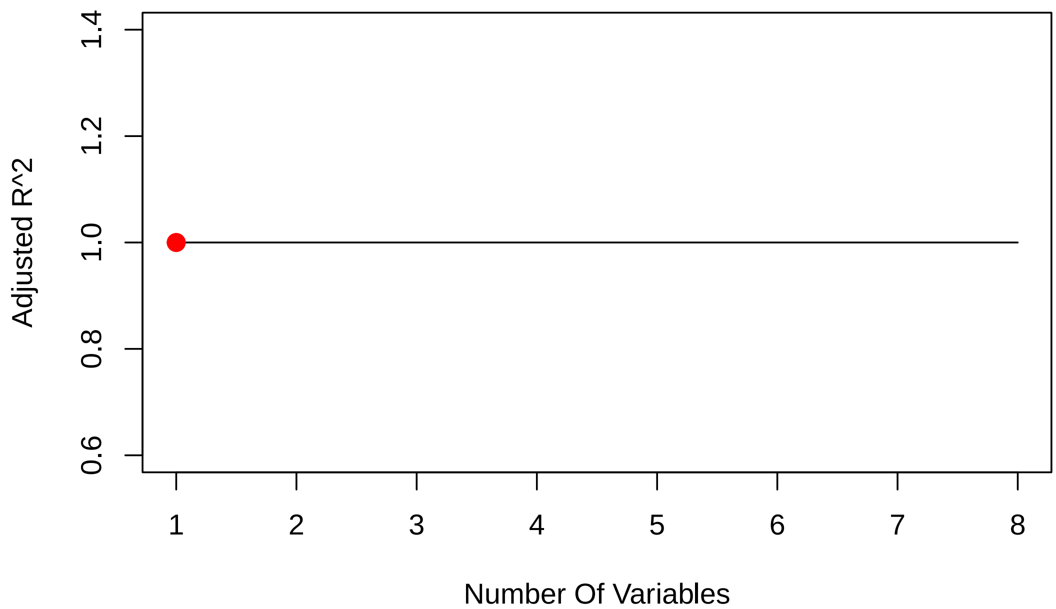
```r
#library(leaps)
formula <- Y ~ X + I(X^2) + I(X^3) + I(X^4) + I(X^5) + I(X^6) + I(X^7) + I(X^8) + I(X^9) + I(X
subset_selection_object <- regsubsets(
    x = formula,
    data = data_frame,
    method = "exhaustive"
)
analyze_subset_selection_object(subset_selection_object)
```



C_p
8
6
4
2
0
−2
1
2
3
4
5
6
7
8
Number Of Variables



Schwartz BIC
−4035
−4045
−4055
1
2
3
4
5
6
7
8
Number Of Variables

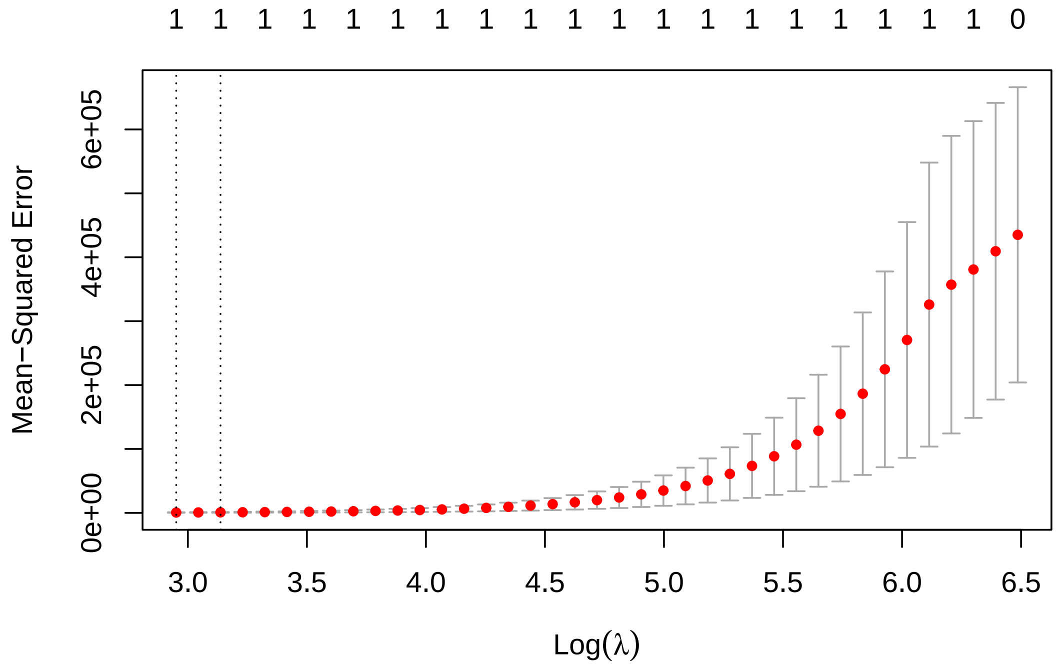



```
# $coefficients_by_Mallows_Cp
# (Intercept)      I(X^7)
#           1           8
#
# $coefficients_by_Schwartz_BIC
# (Intercept)      I(X^7)
#           1           8
#
# $coefficients_by_adjusted_R2
# (Intercept)      I(X^7)
#           1           8
```

The best model obtained according to Mallow's C_p, the Schwartz Bayesian Information Criterion, and adjusted R^2 is

$$Y = \beta_0 + \beta_7 X^7 = 1 + 8X^7$$

```
full_model_matrix <- model.matrix(object = formula, data = data_frame)[, -1]
the_cv.glmnet <- glmnet::cv.glmnet(x = full_model_matrix, y = Y, alpha = 1)
plot(the_cv.glmnet)
```




```
optimal_value_of_lambda <- the_cv.glmnet$lambda.min
optimal_value_of_lambda
```

```
# [1] 19.12382
```

```
polynomial_lasso_regression_model <- glmnet::glmnet(full_model_matrix, y = Y, alpha = 1)
predict(object = polynomial_lasso_regression_model, s = optimal_value_of_lambda, type = "coeff
```

```
# 11 x 1 sparse Matrix of class "dgCMatrix"
#                     s1
# (Intercept) 0.1238844
# X           .
# I(X^2)      .
# I(X^3)      .
# I(X^4)      .
# I(X^5)      .
# I(X^6)      .
# I(X^7)      7.7667958
# I(X^8)      .
# I(X^9)      .
# I(X^10)     .
```

The models chosen by polynomial lasso regression involve the true predictive terms and rough approximations of their coefficients. $\beta_0 = 1.364$ and $\beta_7 = 7.767$.

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.

```
colleges <- ISLR2::College
head(colleges, n = 3)
```

```
#                              Private Apps Accept Enroll Top10perc Top25perc
# Abilene Christian University     Yes 1660   1232    721        23        52
# Adelphi University               Yes 2186   1924    512        16        29
# Adrian College                   Yes 1428   1097    336        22        50
#                              F.Undergrad P.Undergrad Outstate Room.Board Books
# Abilene Christian University        2885         537     7440       3300   450
# Adelphi University                  2683        1227    12280       6450   750
# Adrian College                      1036          99    11250       3750   400
#                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
# Abilene Christian University     2200  70       78      18.1          12   7041
# Adelphi University               1500  29       30      12.2          16  10527
# Adrian College                   1165  53       66      12.9          30   8735
#                              Grad.Rate
# Abilene Christian University        60
# Adelphi University                  56
# Adrian College                      54
```

```
proportion_of_training_data <- 0.9
library(TomLeversRPackage)
split_data <- split_data_set_into_training_and_testing_data(
    data_frame = colleges,
    proportion_of_training_data = proportion_of_training_data
)
vector_of_indices_of_training_data <- split_data$vector_of_indices_of_training_data
vector_of_indices_of_training_data[1:3]
```

```
# [1] 464  37 678
```

```
vector_of_indices_of_testing_data <- split_data$vector_of_indices_of_testing_data
vector_of_indices_of_testing_data[1:3]
```

```
# [1] 728 491 550
```

```
training_data <- split_data$training_data
head(training_data, n = 3)
```

```
#                                     Private  Apps Accept Enroll Top10perc
# Quincy University                       Yes  1025    707    297        22
# Bard College                            Yes  1910    838    285        50
# University of Southern California       Yes 12229   8498   2477        45
#                                     Top25perc F.Undergrad P.Undergrad Outstate
# Quincy University                          66        1070          72    10100
# Bard College                               85        1004          15    19264
```

```
# University of Southern California        71       13259        1429    17230
#                                    Room.Board Books Personal PhD Terminal
# Quincy University                        4140   450     1080   69       71
# Bard College                             6206   750      750   98       98
# University of Southern California        6482   600     2210   90       94
#                                    S.F.Ratio perc.alumni Expend Grad.Rate
# Quincy University                       16.3          32   6880        80
# Bard College                            10.4          30  13894        79
# University of Southern California       11.4          10  17007        68
```

```r
testing_data <- split_data$testing_data
head(testing_data, n = 3)
```

```
#                                   Private Apps Accept Enroll Top10perc Top25perc
# Washington State University            No 6540   5839   2440        31        70
# Saint Francis College IN              Yes  213    166     85        13        36
# St. Mary's College of California      Yes 2643   1611    465        36        80
#                                   F.Undergrad P.Undergrad Outstate Room.Board
# Washington State University             14445        1344     8200       4210
# Saint Francis College IN                  513         247     8670       3820
# St. Mary's College of California         2615         248    13332       6354
#                                   Books Personal PhD Terminal S.F.Ratio
# Washington State University         800     2719  84       87      16.9
# Saint Francis College IN            450     1000  43       78      12.5
# St. Mary's College of California    630     1584  88       89      16.1
#                                   perc.alumni Expend Grad.Rate
# Washington State University                30  10912        56
# Saint Francis College IN                    4   7440        48
# St. Mary's College of California           17   9619        78
```

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

```r
formula = Apps ~ .
linear_model <- lm(formula, data = training_data)
calculate_mean_squared_error(linear_model)
```

```
# [1] 1043968
```

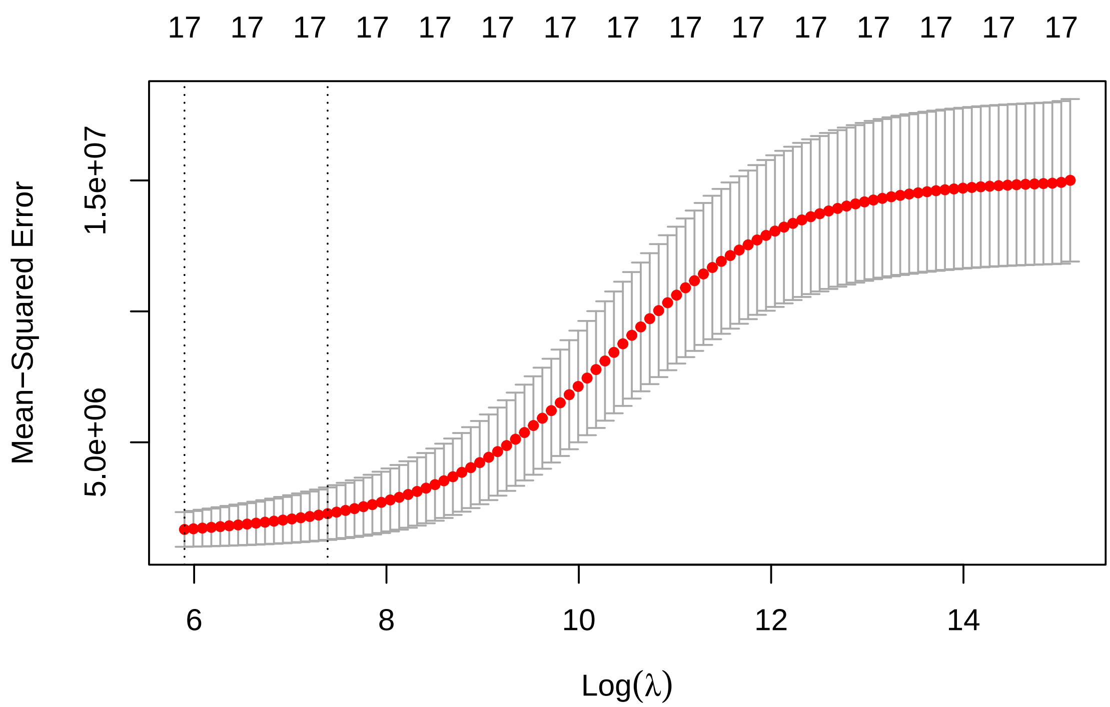
```r
y_hat_linear <- predict(
    object = linear_model,
    testing_data
)
testing_vector_of_numbers_of_applications <-
    colleges$Apps[vector_of_indices_of_testing_data]
y <- testing_vector_of_numbers_of_applications
y_bar <- mean(testing_vector_of_numbers_of_applications)
R_squared <- 1 - mean((y_hat_linear - y)^2) / mean((y_bar - y)^2)
R_squared
```

```
# [1] 0.8744088
```

The resulting mean squared error of prediction $MSEP_{linear} = 1,043,968$.

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

```
full_model_matrix <- model.matrix(object = formula, data = colleges)[, -1]
the_cv.glmnet <- glmnet::cv.glmnet(
    x = full_model_matrix,
    y = colleges$Apps,
    alpha = 0
)
plot(the_cv.glmnet)
```




```
optimal_value_of_lambda <- the_cv.glmnet$lambda.min
optimal_value_of_lambda
```

```
# [1] 364.8993
```

```
training_data_frame <- colleges[vector_of_indices_of_training_data, ]
training_model_matrix <- model.matrix(
    object = formula,
    data = training_data_frame
)[, -1]
training_vector_of_numbers_of_applications <-
    colleges$Apps[vector_of_indices_of_training_data]
polynomial_ridge_regression_model <- glmnet::glmnet(
    x = training_model_matrix,
    y = training_vector_of_numbers_of_applications,
```

```
    alpha = 0
)
testing_data_frame <- colleges[vector_of_indices_of_testing_data, ]
testing_model_matrix <- model.matrix(
    object = formula,
    data = testing_data_frame
)[, -1]
vector_of_predicted_numbers_of_applications <- predict(
    object = polynomial_ridge_regression_model,
    s = optimal_value_of_lambda,
    newx = testing_model_matrix
)[, 1]
vector_of_residuals <-
    vector_of_predicted_numbers_of_applications -
    testing_vector_of_numbers_of_applications
vector_of_squared_residuals <- vector_of_residuals^2
mean_squared_error <- mean(vector_of_squared_residuals)
mean_squared_error
```

```
# [1] 1071002
```

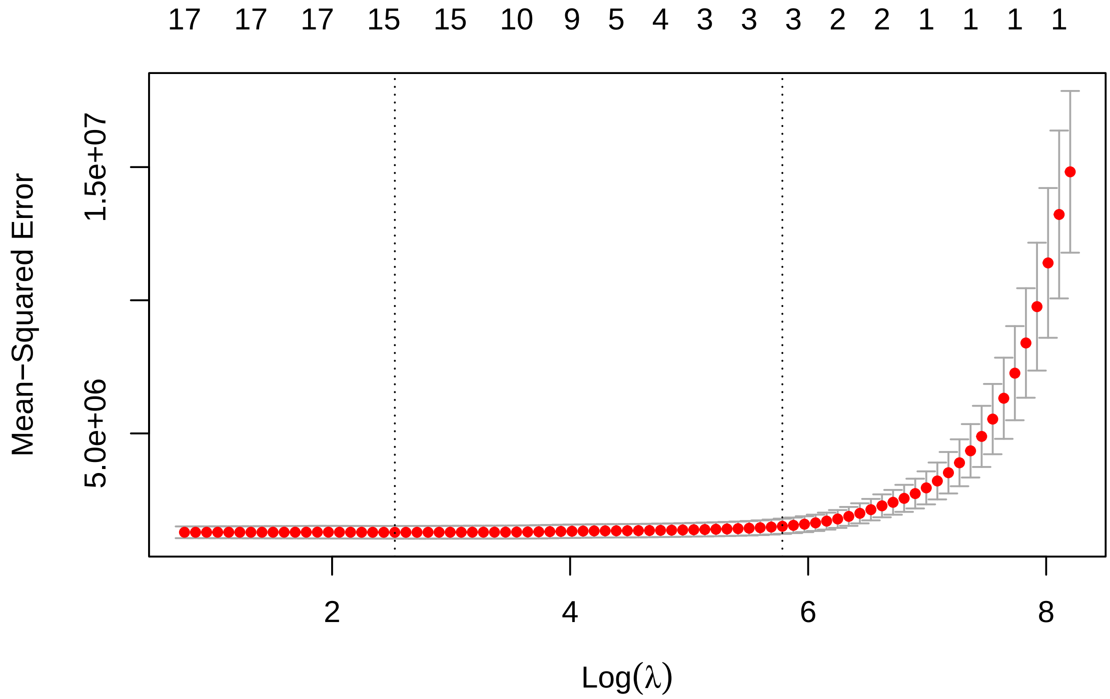
```
y_hat_ridge <- vector_of_predicted_numbers_of_applications
R_squared <- 1 - mean((y_hat_ridge - y)^2) / mean((y_bar - y)^2)
R_squared
```

```
# [1] 0.8908082
```

The resulting mean squared error of prediction $MSEP_{ridge} = 1,071,002$. $MSEP_{linear} < MSEP_{ridge}$.

(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.

```
full_model_matrix <- model.matrix(object = formula, data = colleges)[, -1]
the_cv.glmnet <- glmnet::cv.glmnet(
    x = full_model_matrix,
    y = colleges$Apps,
    alpha = 1
)
plot(the_cv.glmnet)
```




```
optimal_value_of_lambda <- the_cv.glmnet$lambda.min
optimal_value_of_lambda
```

```
# [1] 12.51776
```

```
training_model_matrix <- model.matrix(
    object = formula,
    data = training_data_frame
)[, -1]
training_vector_of_numbers_of_applications <-
    colleges$Apps[vector_of_indices_of_training_data]
testing_vector_of_numbers_of_applications <-
    colleges$Apps[vector_of_indices_of_testing_data]
polynomial_lasso_regression_model <- glmnet::glmnet(
    x = training_model_matrix,
    y = training_vector_of_numbers_of_applications,
    alpha = 1
)
testing_model_matrix <- model.matrix(
    object = formula,
    data = testing_data_frame
)[, -1]
vector_of_predicted_numbers_of_applications <- predict(
    object = polynomial_lasso_regression_model,
    s = optimal_value_of_lambda,
    newx = testing_model_matrix
```

```r
)[, 1]
vector_of_residuals <-
    vector_of_predicted_numbers_of_applications -
    testing_vector_of_numbers_of_applications
vector_of_squared_residuals <- vector_of_residuals^2
mean_squared_error <- mean(vector_of_squared_residuals)
mean_squared_error
```

```
# [1] 1168351
```

```r
predict(
    object = polynomial_lasso_regression_model,
    s = optimal_value_of_lambda,
    type = "coefficients"
)
```

```
# 18 x 1 sparse Matrix of class "dgCMatrix"
#                        s1
# (Intercept) -4.772317e+02
# PrivateYes  -4.945991e+02
# Accept       1.550834e+00
# Enroll      -4.885202e-01
# Top10perc    3.994779e+01
# Top25perc   -5.578160e+00
# F.Undergrad  1.402583e-03
# P.Undergrad  4.404712e-02
# Outstate    -7.179337e-02
# Room.Board   1.435632e-01
# Books        6.641890e-02
# Personal     9.897576e-03
# PhD         -6.550167e+00
# Terminal    -3.749746e+00
# S.F.Ratio    4.409518e+00
# perc.alumni -1.416008e-01
# Expend       6.819184e-02
# Grad.Rate    6.068215e+00
```

```r
y_hat_lasso <- vector_of_predicted_numbers_of_applications
R_squared <- 1 - mean((y_hat_lasso - y)^2) / mean((y_bar - y)^2)
R_squared
```

```
# [1] 0.8808833
```

The number of non-zero coefficients $M = 17$, which is equal to the number of non-zero coefficients. The resulting mean squared error of prediction $MSEP_{lasso} = 1,168,351$. $MSEP_{linear} < MSEP_{ridge} < MSEP_{lasso}$.

(e) Fit a PCR model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation.

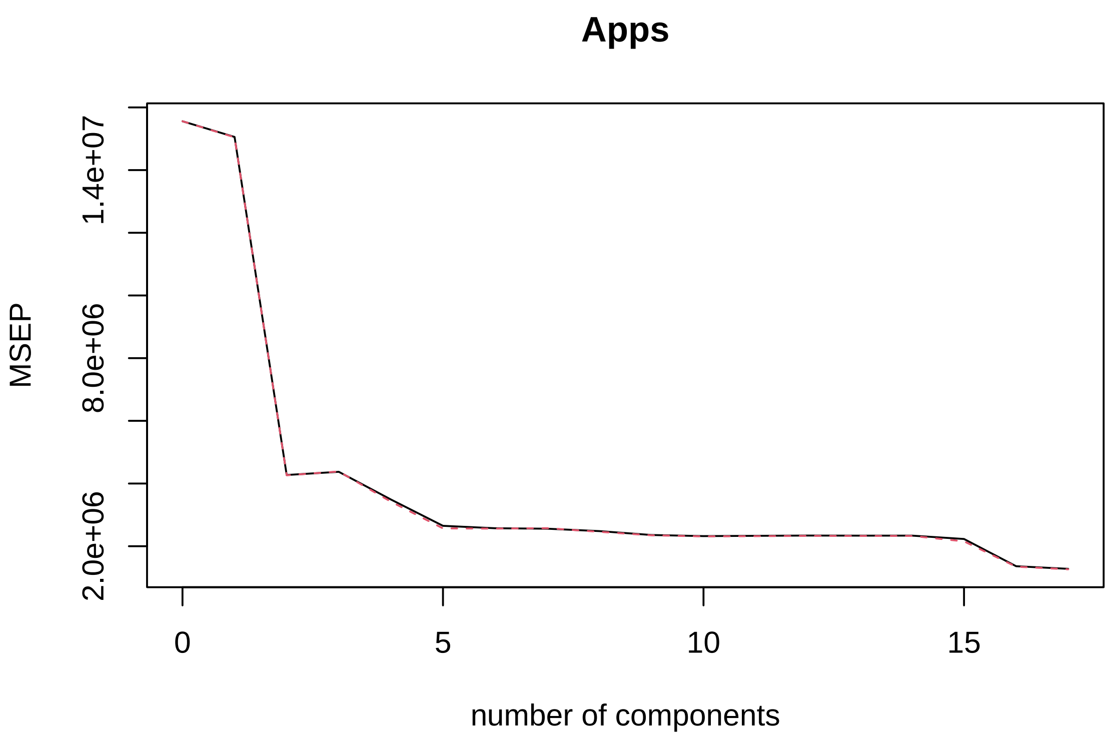
```r
library(pls)
```

```
# Warning: package 'pls' was built under R version 4.3.1

#
# Attaching package: 'pls'

# The following object is masked from 'package:stats':
#
#     loadings
```

```
the_mvr <- pcr(formula, data = training_data_frame, scale = TRUE, validation = "CV")
validationplot(the_mvr, val.type = "MSEP")
```




```
vector_of_predicted_numbers_of_applications <- predict(
    object = the_mvr,
    testing_data_frame,
    ncomp = 17
)
vector_of_residuals <-
    vector_of_predicted_numbers_of_applications -
    testing_vector_of_numbers_of_applications
vector_of_squared_residuals <- vector_of_residuals^2
mean_squared_error <- mean(vector_of_squared_residuals)
mean_squared_error
```

```
# [1] 1231856
```

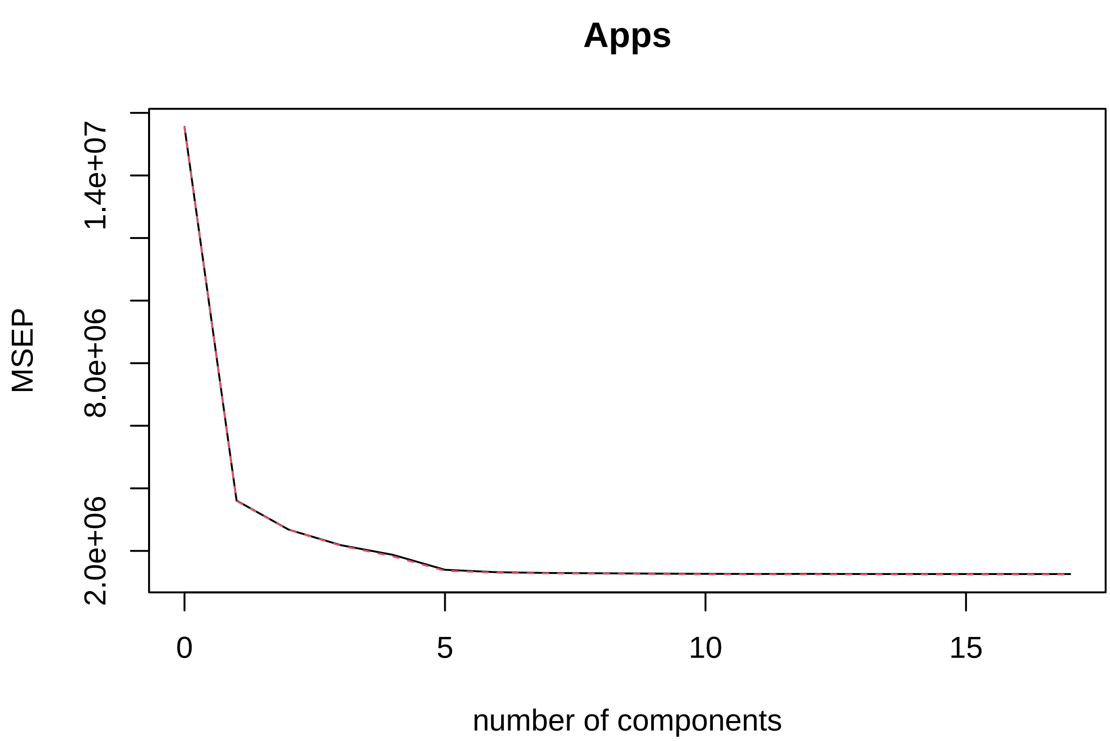
```r
y_hat_PCR <- vector_of_predicted_numbers_of_applications
R_squared <- 1 - mean((y_hat_PCR - y)^2) / mean((y_bar - y)^2)
R_squared
```

```
# [1] 0.8744088
```

The resulting mean squared error of prediction $MSEP_{PCR} = 1,231,856$. $MSEP_{linear} < MSEP_{ridge} < MSEP_{lasso} < MSEP_{PCR}$.

(f) Fit a PLS model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation.

```r
library(pls)
the_mvr <- plsr(formula, data = training_data_frame, scale = TRUE, validation = "CV")
validationplot(the_mvr, val.type = "MSEP")
```




```r
vector_of_predicted_numbers_of_applications <- predict(
    object = the_mvr,
    testing_data_frame,
    ncomp = 17
)
vector_of_residuals <-
    vector_of_predicted_numbers_of_applications -
    testing_vector_of_numbers_of_applications
vector_of_squared_residuals <- vector_of_residuals^2
mean_squared_error <- mean(vector_of_squared_residuals)
mean_squared_error
```

```
# [1] 1231856
```

```r
y_hat_PLSR <- vector_of_predicted_numbers_of_applications
R_squared <- 1 - mean((y_hat_PLSR - y)^2) / mean((y_bar - y)^2)
R_squared
```

```
# [1] 0.8744088
```

The number of non-zero coefficients estimates is 17. The resulting mean squared error of prediction $MSEP_{PLSR} = 1,231,856$. $MSEP_{linear} < MSEP_{ridge} < MSEP_{lasso} < MSEP_{PCR}; MSEP_{PCR} = MSEP_{PLSR}$.

(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?

As above, $MSEP_{linear} < MSEP_{ridge} < MSEP_{lasso} < MSEP_{PCR}; MSEP_{PCR} = MSEP_{PLSR}$. The mean squared error of prediction of the linear model $MSEP_{linear} = 1,043,968$. The mean squared error of prediction of the PLSR model $MSEP_{PLSR} = 1,231,856$. The rate of difference between these errors is -0.153. There is a strong effect for each model; ${R^2}_{linear} = {R^2}_{PCR} = {R^2}_{PLSR}; {R^2}_{linear} < {R^2}_{lasso} < {R^2}_{ridge}$.