R Notebook

***8. In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.***

1. 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(1)  
X = rnorm(100)  
eps = rnorm(100)

1. Generate a response vector Y of length n = 100 according to the model Y = β0 + β1X + β2X2 + β3X3 + ϵ, where β0, β1, β2, and β3 are constants of your choice.

By selecting β0 = 3, β1 = 2, β2 = -3, and β3 = 0.3

beta0 = 3  
beta1 = 2  
beta2 = -3  
beta3 = 0.3  
Y = beta0 + beta1 \* X + beta2 \* X^2 + beta3 \* X^3 + eps

1. Use the reg subsets() function to perform best subset selection in order to choose the best model containing the predictors X, X2,…,X10. 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 .

install.Packages("leaps")

library(leaps)

## Warning: package 'leaps' was built under R version 4.2.1

data.full = data.frame(y = Y, x = X)  
mod.full = regsubsets(y ~ poly(x, 10, raw = T), data = data.full, nvmax = 10)  
mod.summary = summary(mod.full)  
  
# Find the model size for best cp, BIC and adjr2  
which.min(mod.summary$cp)

## [1] 3

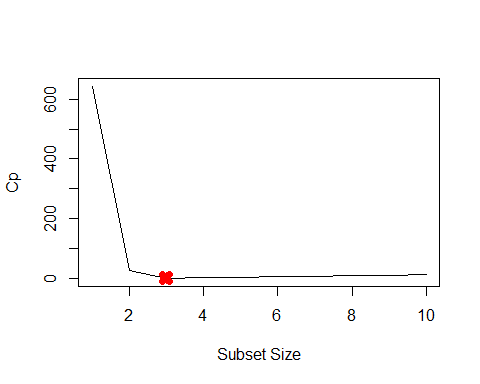
which.min(mod.summary$bic)

## [1] 3

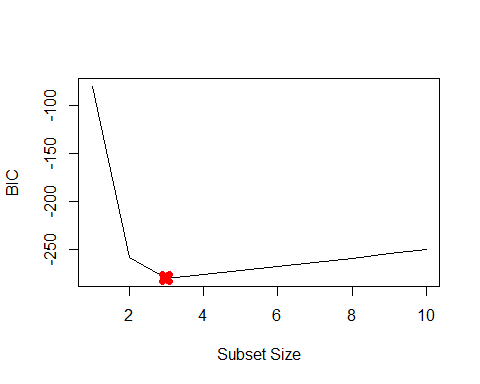
which.max(mod.summary$adjr2)

## [1] 3

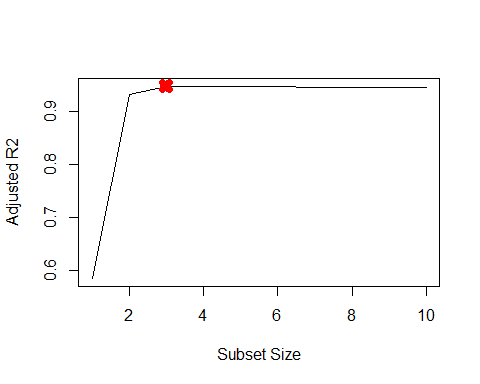
# Plot cp, BIC and adjr2  
plot(mod.summary$cp, xlab = "Subset Size", ylab = "Cp", pch = 20, type = "l")  
points(3, mod.summary$cp[3], pch = 4, col = "red", lwd = 7)



plot(mod.summary$bic, xlab = "Subset Size", ylab = "BIC", pch = 20, type = "l")  
points(3, mod.summary$bic[3], pch = 4, col = "red", lwd = 7)



plot(mod.summary$adjr2, xlab = "Subset Size", ylab = "Adjusted R2", pch = 20,   
 type = "l")  
points(3, mod.summary$adjr2[3], pch = 4, col = "red", lwd = 7)

 We find that with Cp, BIC and Adjusted R2 criteria, 3, 3, and 3 variable models are respectively picked.

coefficients(mod.full, id = 3)

## (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2   
## 3.07627412 2.35623596 -3.16514887   
## poly(x, 10, raw = T)7   
## 0.01046843

All statistics pick X7 over X3. The remaining coefficients are quite close to β s.

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

mod.fwd = regsubsets(y ~ poly(x, 10, raw = T), data = data.full, nvmax = 10,   
 method = "forward")  
mod.bwd = regsubsets(y ~ poly(x, 10, raw = T), data = data.full, nvmax = 10,   
 method = "backward")  
fwd.summary = summary(mod.fwd)  
bwd.summary = summary(mod.bwd)  
which.min(fwd.summary$cp)

## [1] 3

which.min(bwd.summary$cp)

## [1] 3

which.min(fwd.summary$bic)

## [1] 3

which.min(bwd.summary$bic)

## [1] 3

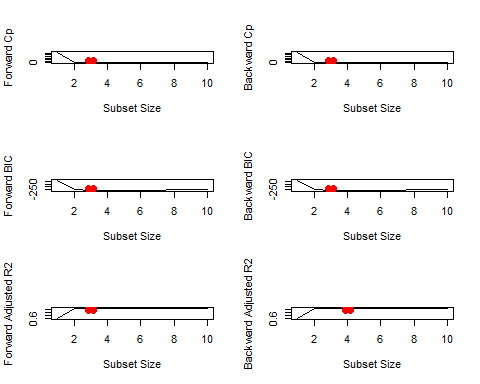
which.max(fwd.summary$adjr2)

## [1] 3

which.max(bwd.summary$adjr2)

## [1] 3

# Plot the statistics  
par(mfrow = c(3, 2))  
plot(fwd.summary$cp, xlab = "Subset Size", ylab = "Forward Cp", pch = 20, type = "l")  
points(3, fwd.summary$cp[3], pch = 4, col = "red", lwd = 7)  
plot(bwd.summary$cp, xlab = "Subset Size", ylab = "Backward Cp", pch = 20, type = "l")  
points(3, bwd.summary$cp[3], pch = 4, col = "red", lwd = 7)  
plot(fwd.summary$bic, xlab = "Subset Size", ylab = "Forward BIC", pch = 20,   
 type = "l")  
points(3, fwd.summary$bic[3], pch = 4, col = "red", lwd = 7)  
plot(bwd.summary$bic, xlab = "Subset Size", ylab = "Backward BIC", pch = 20,   
 type = "l")  
points(3, bwd.summary$bic[3], pch = 4, col = "red", lwd = 7)  
plot(fwd.summary$adjr2, xlab = "Subset Size", ylab = "Forward Adjusted R2",   
 pch = 20, type = "l")  
points(3, fwd.summary$adjr2[3], pch = 4, col = "red", lwd = 7)  
plot(bwd.summary$adjr2, xlab = "Subset Size", ylab = "Backward Adjusted R2",   
 pch = 20, type = "l")  
points(4, bwd.summary$adjr2[4], pch = 4, col = "red", lwd = 7)

 all statistics pick 3 variable models except backward stepwise with adjusted R2. Here are the coefficients

coefficients(mod.fwd, id = 3)

## (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2   
## 3.07627412 2.35623596 -3.16514887   
## poly(x, 10, raw = T)7   
## 0.01046843

coefficients(mod.bwd, id = 3)

## (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2   
## 3.078881355 2.419817953 -3.177235617   
## poly(x, 10, raw = T)9   
## 0.001870457

coefficients(mod.fwd, id = 4)

## (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2   
## 3.112358625 2.369858879 -3.275726574   
## poly(x, 10, raw = T)4 poly(x, 10, raw = T)7   
## 0.027673638 0.009997134

Here, X7 is chosen over X3 by forward stepwise. While backward stepwise with four variables selects X4 and X7, backward stepwise with three variables selects X9. Near s, all other coefficients are.

1. Now fit a lasso model to the simulated data, again using X, X2, …,X10 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.

Training Lasso on the data

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.2.1

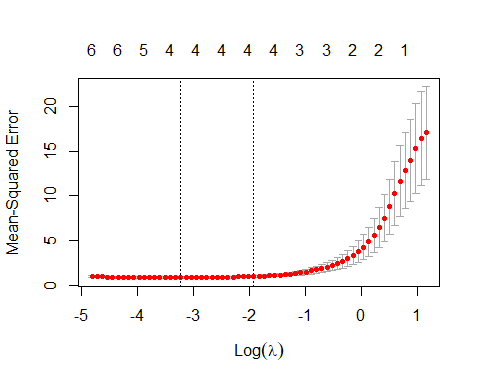
## Loading required package: Matrix

## Loaded glmnet 4.1-4

xmat = model.matrix(y ~ poly(x, 10, raw = T), data = data.full)[, -1]  
mod.lasso = cv.glmnet(xmat, Y, alpha = 1)  
best.lambda = mod.lasso$lambda.min  
best.lambda

## [1] 0.03991416

plot(mod.lasso)



# Next fit the model on entire data using best lambda  
best.model = glmnet(xmat, Y, alpha = 1)  
predict(best.model, s = best.lambda, type = "coefficients")

## 11 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 3.0398151056  
## poly(x, 10, raw = T)1 2.2303371338  
## poly(x, 10, raw = T)2 -3.1033192679  
## poly(x, 10, raw = T)3 .   
## poly(x, 10, raw = T)4 .   
## poly(x, 10, raw = T)5 0.0498410763  
## poly(x, 10, raw = T)6 .   
## poly(x, 10, raw = T)7 0.0008068431  
## poly(x, 10, raw = T)8 .   
## poly(x, 10, raw = T)9 .   
## poly(x, 10, raw = T)10 .

Lasso also picks X5 over X3. It also picks X7 with negligible coefficient.

1. Now generate a response vector Y according to the model Y = β0 + β7X7 + ϵ, and perform best subset selection and the lasso. Discuss the results obtained.

Create new Y with different β7=7.

beta7 = 7  
Y = beta0 + beta7 \* X^7 + eps  
# Predict using regsubsets  
data.full = data.frame(y = Y, x = X)  
mod.full = regsubsets(y ~ poly(x, 10, raw = T), data = data.full, nvmax = 10)  
mod.summary = summary(mod.full)  
  
# Find the model size for best cp, BIC and adjr2  
which.min(mod.summary$cp)

## [1] 2

which.min(mod.summary$bic)

## [1] 1

which.max(mod.summary$adjr2)

## [1] 4

coefficients(mod.full, id = 1)

## (Intercept) poly(x, 10, raw = T)7   
## 2.95894 7.00077

coefficients(mod.full, id = 2)

## (Intercept) poly(x, 10, raw = T)2 poly(x, 10, raw = T)7   
## 3.0704904 -0.1417084 7.0015552

coefficients(mod.full, id = 4)

## (Intercept) poly(x, 10, raw = T)1 poly(x, 10, raw = T)2   
## 3.0762524 0.2914016 -0.1617671   
## poly(x, 10, raw = T)3 poly(x, 10, raw = T)7   
## -0.2526527 7.0091338

xmat = model.matrix(y ~ poly(x, 10, raw = T), data = data.full)[, -1]  
mod.lasso = cv.glmnet(xmat, Y, alpha = 1)  
best.lambda = mod.lasso$lambda.min  
best.lambda

## [1] 12.36884

best.model = glmnet(xmat, Y, alpha = 1)  
predict(best.model, s = best.lambda, type = "coefficients")

## 11 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 3.820215  
## poly(x, 10, raw = T)1 .   
## poly(x, 10, raw = T)2 .   
## poly(x, 10, raw = T)3 .   
## poly(x, 10, raw = T)4 .   
## poly(x, 10, raw = T)5 .   
## poly(x, 10, raw = T)6 .   
## poly(x, 10, raw = T)7 6.796694  
## poly(x, 10, raw = T)8 .   
## poly(x, 10, raw = T)9 .   
## poly(x, 10, raw = T)10 .

Lasso also picks the best 1-variable model but intercet is quite off (3.8 vs 3).

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

1. Split the data set into a training set and a test set.

library(ISLR)  
set.seed(11)  
sum(is.na(College))

## [1] 0

train.size = dim(College)[1] / 2  
train = sample(1:dim(College)[1], train.size)  
test = -train  
College.train = College[train, ]  
College.test = College[test, ]

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

lm.fit = lm(Apps~., data=College.train)  
lm.pred = predict(lm.fit, College.test)  
mean((College.test[, "Apps"] - lm.pred)^2)

## [1] 1026096

Test RSS is 1026096.

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

library(glmnet)

train.mat = model.matrix(Apps~., data=College.train)  
test.mat = model.matrix(Apps~., data=College.test)  
grid = 10 ^ seq(4, -2, length=100)  
mod.ridge = cv.glmnet(train.mat, College.train[, "Apps"], alpha=0, lambda=grid, thresh=1e-12)  
lambda.best = mod.ridge$lambda.min  
lambda.best

## [1] 0.01

ridge.pred = predict(mod.ridge, newx=test.mat, s=lambda.best)  
mean((College.test[, "Apps"] - ridge.pred)^2)

## [1] 1026069

Test RSS is almost equal to OLS, 1026069.

1. 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.

mod.lasso = cv.glmnet(train.mat, College.train[, "Apps"], alpha=1, lambda=grid, thresh=1e-12)  
lambda.best = mod.lasso$lambda.min  
lambda.best

## [1] 0.01

lasso.pred = predict(mod.lasso, newx=test.mat, s=lambda.best)  
mean((College.test[, "Apps"] - lasso.pred)^2)

## [1] 1026036

Test RSS is almost equal to OLS, 1026036.

mod.lasso = glmnet(model.matrix(Apps~., data=College), College[, "Apps"], alpha=1)  
predict(mod.lasso, s=lambda.best, type="coefficients")

## 19 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) -471.39372052  
## (Intercept) .   
## PrivateYes -491.04485137  
## Accept 1.57033288  
## Enroll -0.75961467  
## Top10perc 48.14698892  
## Top25perc -12.84690695  
## F.Undergrad 0.04149116  
## P.Undergrad 0.04438973  
## Outstate -0.08328388  
## Room.Board 0.14943472  
## Books 0.01532293  
## Personal 0.02909954  
## PhD -8.39597537  
## Terminal -3.26800340  
## S.F.Ratio 14.59298267  
## perc.alumni -0.04404771  
## Expend 0.07712632  
## Grad.Rate 8.28950241