**COMP3354 - Statistical Learning**

**Assignment 2**

**Name: Pranav Talwar**

**UID: 3035435462**

**Chapter 6, Question 8**

a) Rcommand: > set.seed(5462)

> library(leaps)

> library(glmnet)

> # Part A

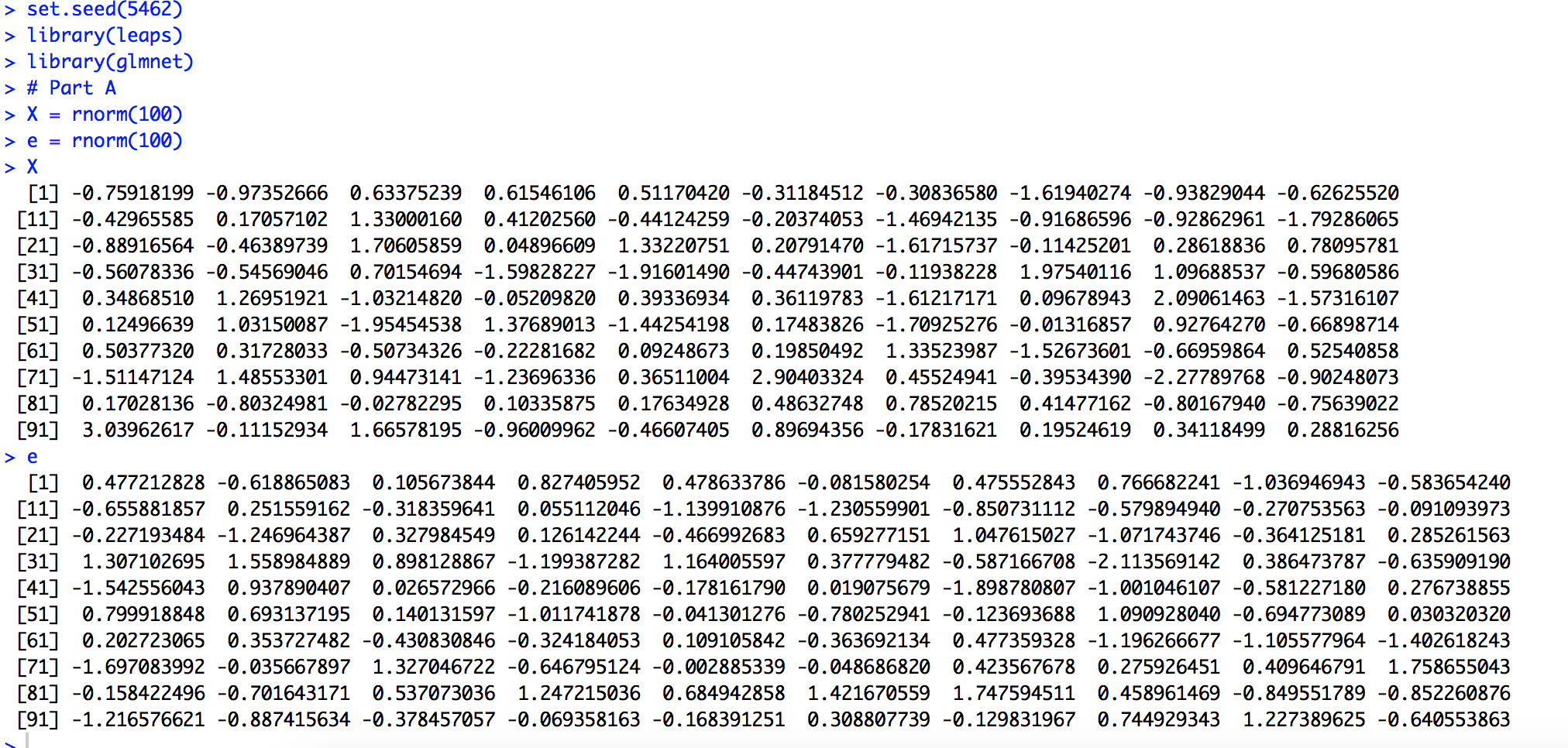
> X = rnorm(100)

> e = rnorm(100)

> X

> e

Output:

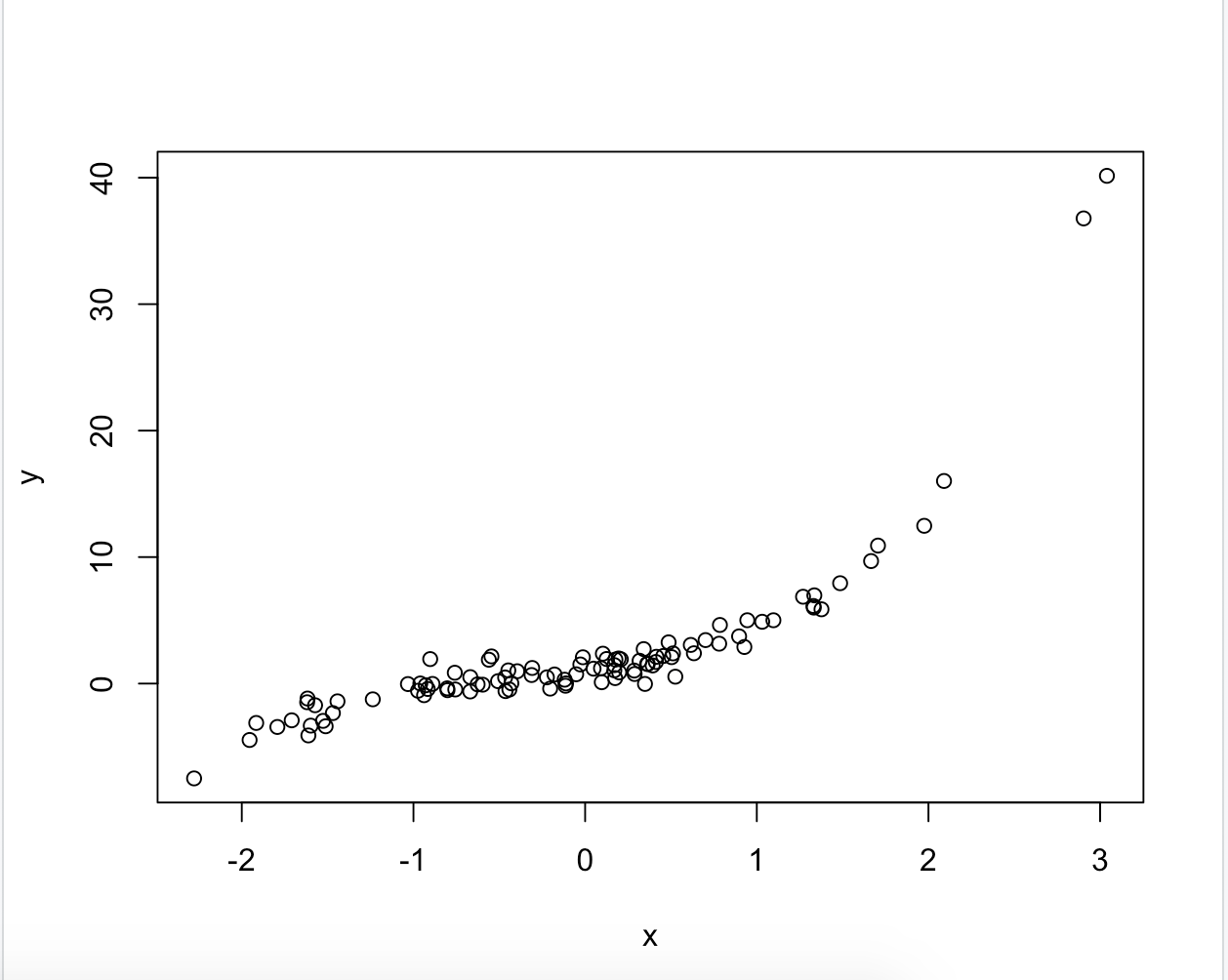


b) Rcommand: > Y = 1 + X + X\*X + X\*X\*X + e

> data.new = data.frame(y = Y, x = X)

> plot(data.new)

Output:



In the following model Y = β0 +β1\*X +β2\*X2 +β3\*X3 +e, the values chosen are 1 for all the betas.

c) Rcommand: > subsets = regsubsets(y~poly(x,10, raw = TRUE),data = data.new,nvmax = 10)

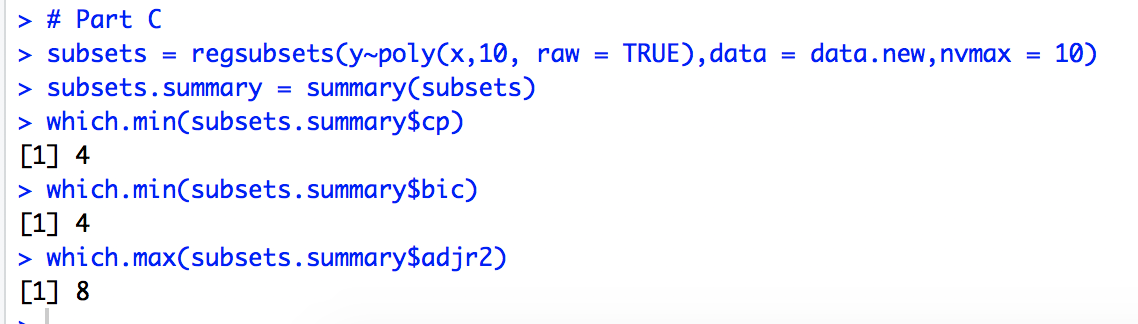
> subsets.summary = summary(subsets)

> which.min(subsets.summary$cp)

> which.min(subsets.summary$bic)

> which.max(subsets.summary$adjr2)

Output :



According to the best subset selection method, the best models are;

1. CP: Model 4
2. BIC: Model 4
3. ADJR2: Model 8

Rcommand: > par(mfrow=c(2,2))

> plot(subsets.summary$cp, xlab = "Size of the Subset", ylab ="CP", type = "l")

> points(4, subsets.summary$cp[4],col = "red")

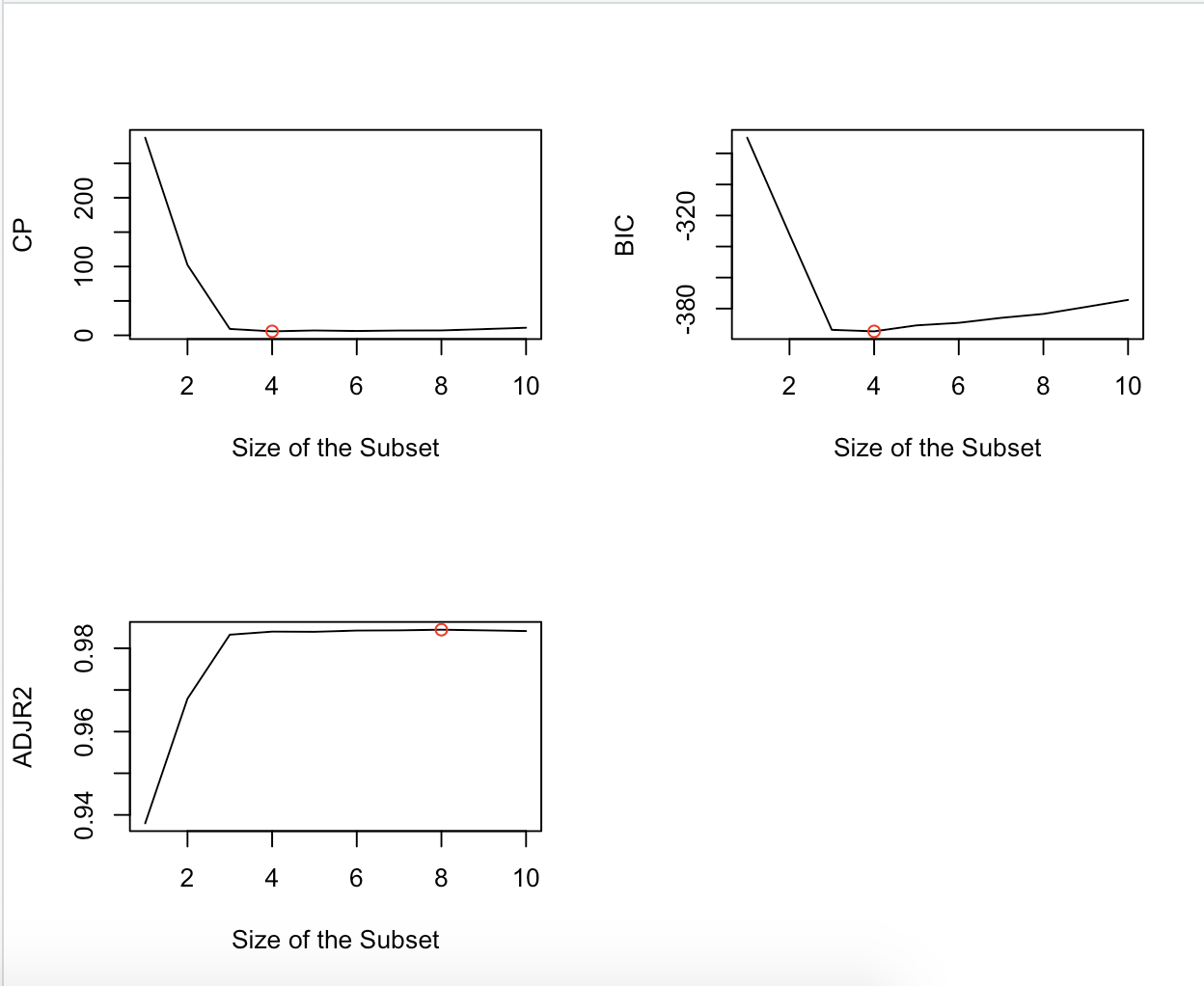
> plot(subsets.summary$bic, xlab = "Size of the Subset", ylab = "BIC", type ="l")

> points(4, subsets.summary$bic[4],col = "red")

> plot(subsets.summary$adjr2, xlab = "Size of the Subset", ylab ="ADJR2", type = "l")

> points(8, subsets.summary$adjr2[8],col = "red")

Output:



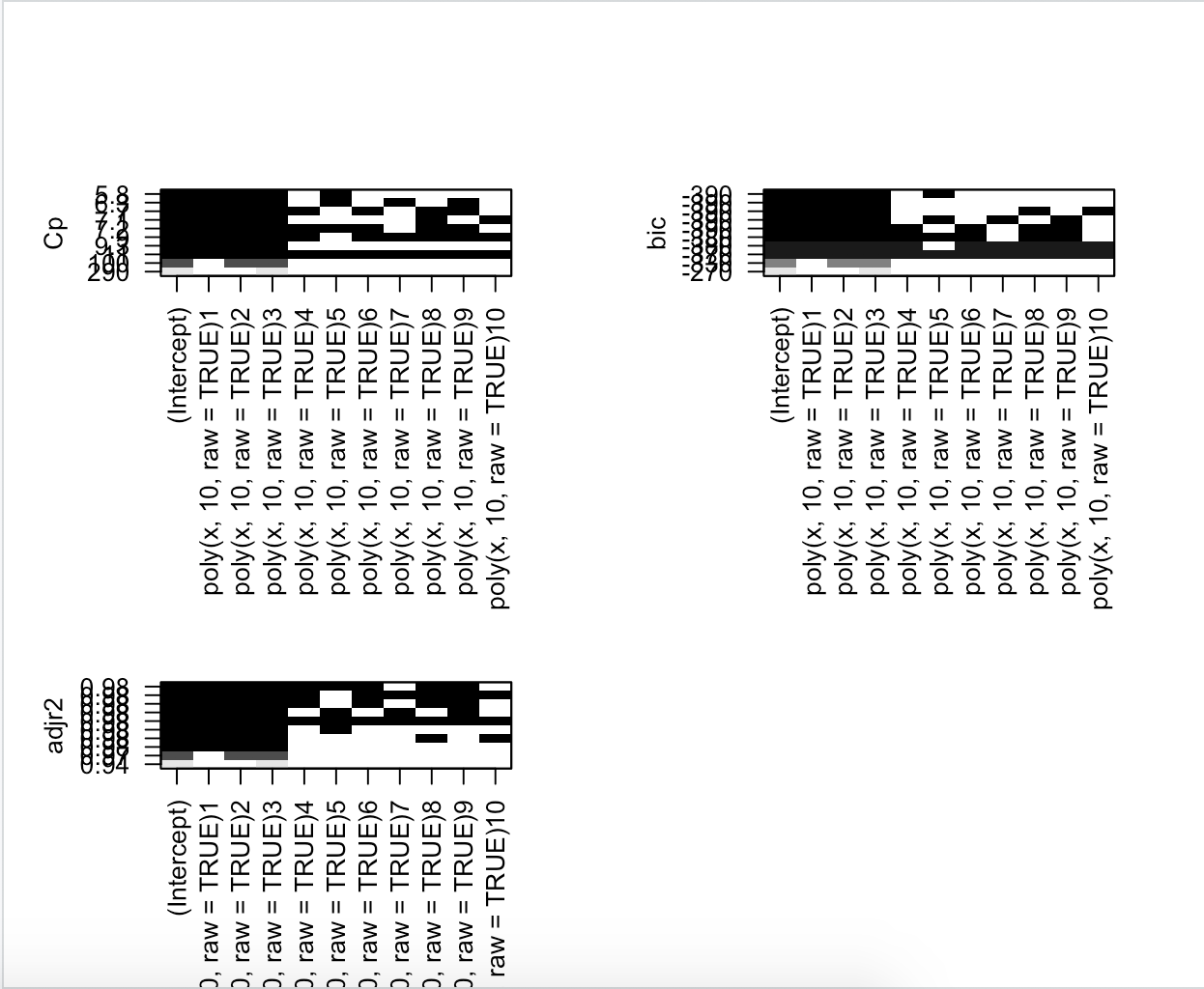
Rcommand: > par(mfrow=c(2,2))

> plot(subsets, scale="Cp")

> plot(subsets, scale="bic")

> plot(subsets, scale="adjr2")

Output:

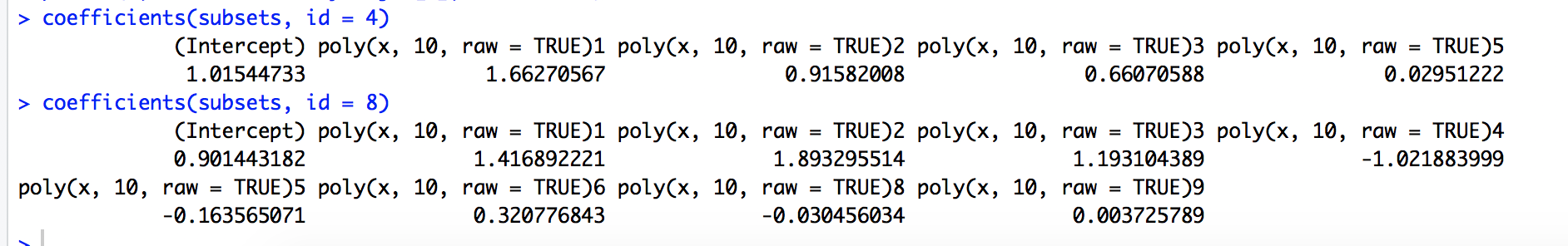


From these plots we can see that the model 4 is the best when CP and BIC are used in best subset selection. Model 4 has the lowest CP and BIC among all the models. Whereas, Model 8 is the best when ADJR2 is used in the best subset selection method since it has the highest ADJR2 among all the other models.

Rcommand: > coefficients(subsets, id = 4)

> coefficients(subsets, id = 8)

Output:



The coefficients of model 4 and model 8 are given above. We can see that model 3 includes the coefficients till X^3 as well as X^5 (very small) whereas model 8 also includes variables X^4, X^6, X^8 and X^10. The coefficients of model 3 are more or less accurate as compared to the real model, whereas the coefficient of X^5 is negligible.

d) **Using Forward Selection**

Rcommand: > subsets.fwd = regsubsets(y~poly(x,10, raw = TRUE), data = data.new, nvmax=10, method = "forward")

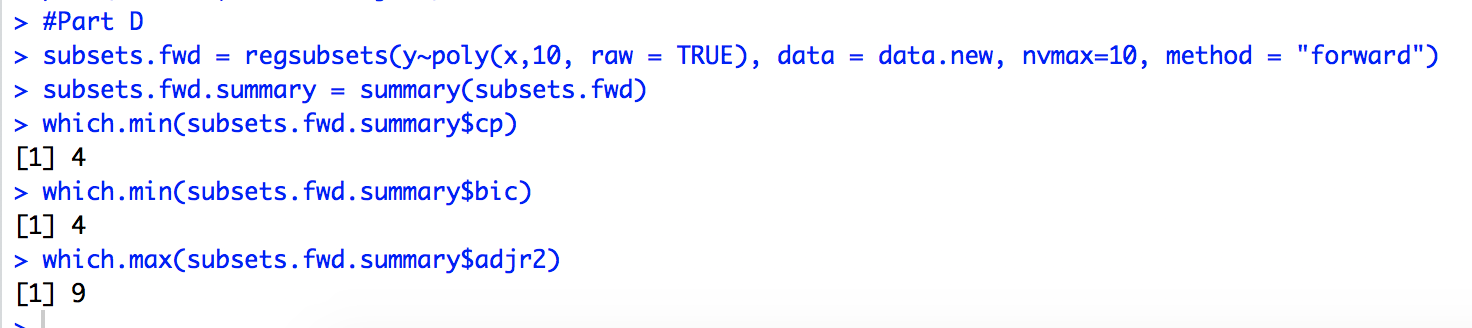
> subsets.fwd.summary = summary(subsets.fwd)

> which.min(subsets.fwd.summary$cp)

> which.min(subsets.fwd.summary$bic)

> which.max(subsets.fwd.summary$adjr2)

Output:



Rcommand: > par(mfrow=c(2,2))

> plot(subsets.fwd.summary$cp, xlab = "Size of the Subset", ylab ="CP", type = "l")

> points(4, subsets.fwd.summary$cp[4],col = "red")

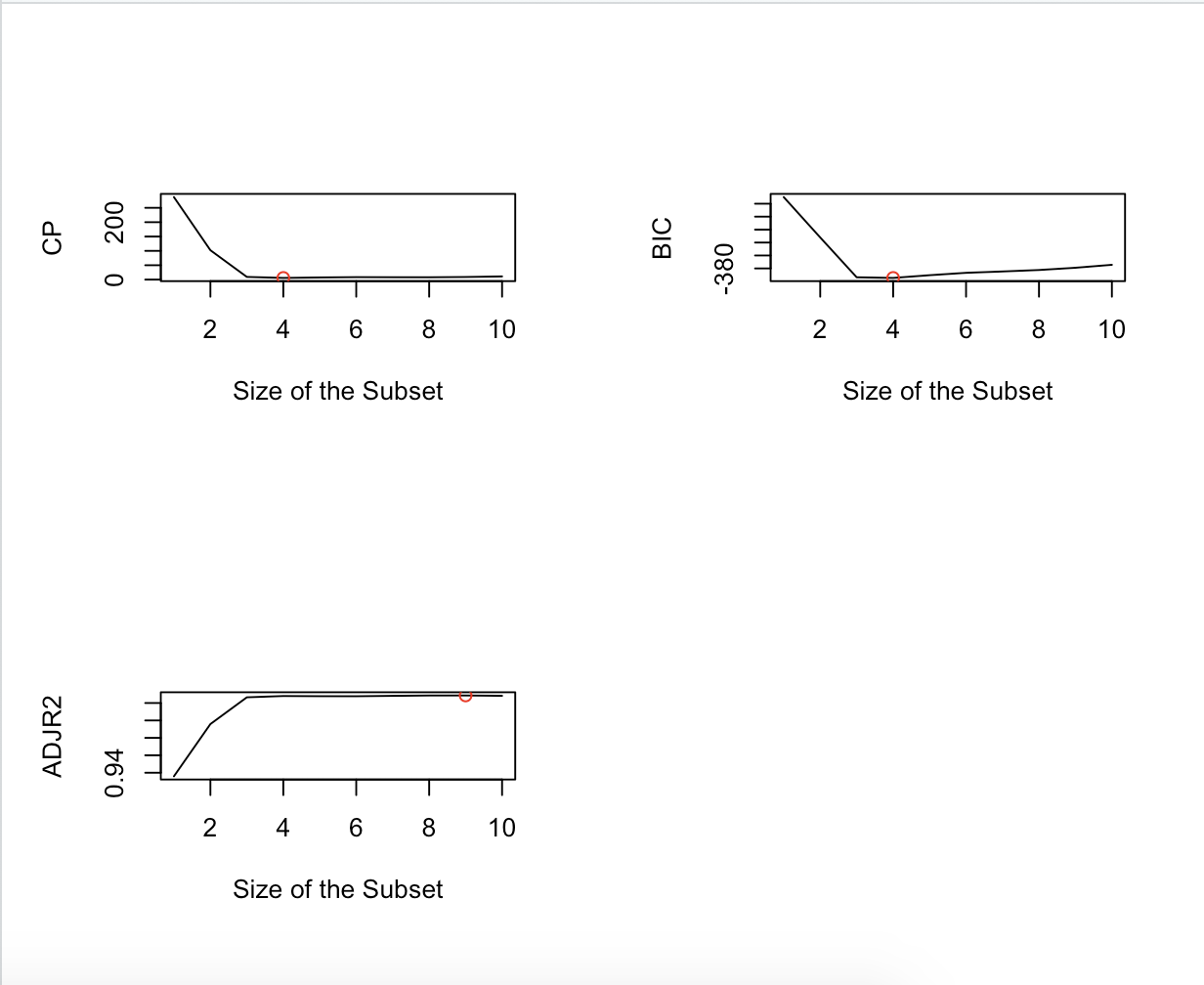
> plot(subsets.fwd.summary$bic, xlab = "Size of the Subset", ylab = "BIC", type ="l")

> points(4, subsets.fwd.summary$bic[4],col = "red")

> plot(subsets.fwd.summary$adjr2, xlab = "Size of the Subset", ylab ="ADJR2", type = "l")

> points(9, subsets.fwd.summary$adjr2[9],col = "red")

Output:

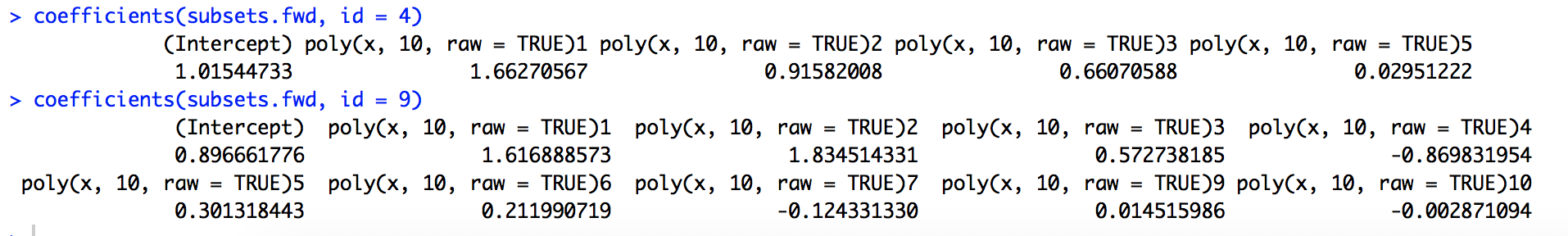


The best model according to forward selection approach when CP, BIC and ACJR2 are used is model 4 (CP, BIC)and model 9 (ADJR2) as can be seen in the plots given above.

Rcommand: > coefficients(subsets.fwd, id = 4)

> coefficients(subsets.fwd, id = 9)

Output:

****

The coefficients of model 4 and model 9 are given above. The coefficients of model 4 are the same as given by best subset selection method, whereas the coefficients for model 9 contain all the variables up till X^10 with the exception of X^8.

**Using Backward selection**

Rcommand: > subsets.bwd = regsubsets(y~poly(x,10, raw = TRUE), data = data.new, nvmax=10, method = "backward")

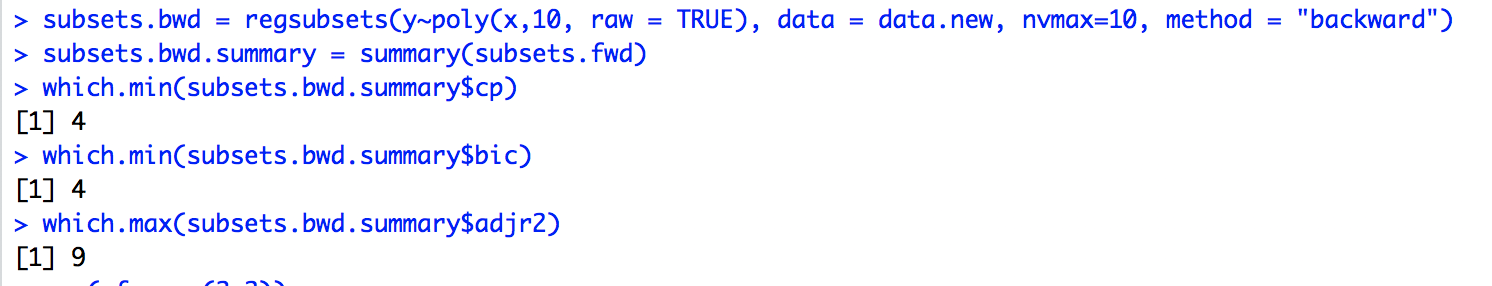
> subsets.bwd.summary = summary(subsets.fwd)

> which.min(subsets.bwd.summary$cp)

> which.min(subsets.bwd.summary$bic)

> which.max(subsets.bwd.summary$adjr2)

Output:



Rcommand: > par(mfrow=c(2,2))

> plot(subsets.bwd.summary$cp, xlab = "Size of the Subset", ylab ="CP", type = "l")

> points(4, subsets.bwd.summary$cp[4],col = "red")

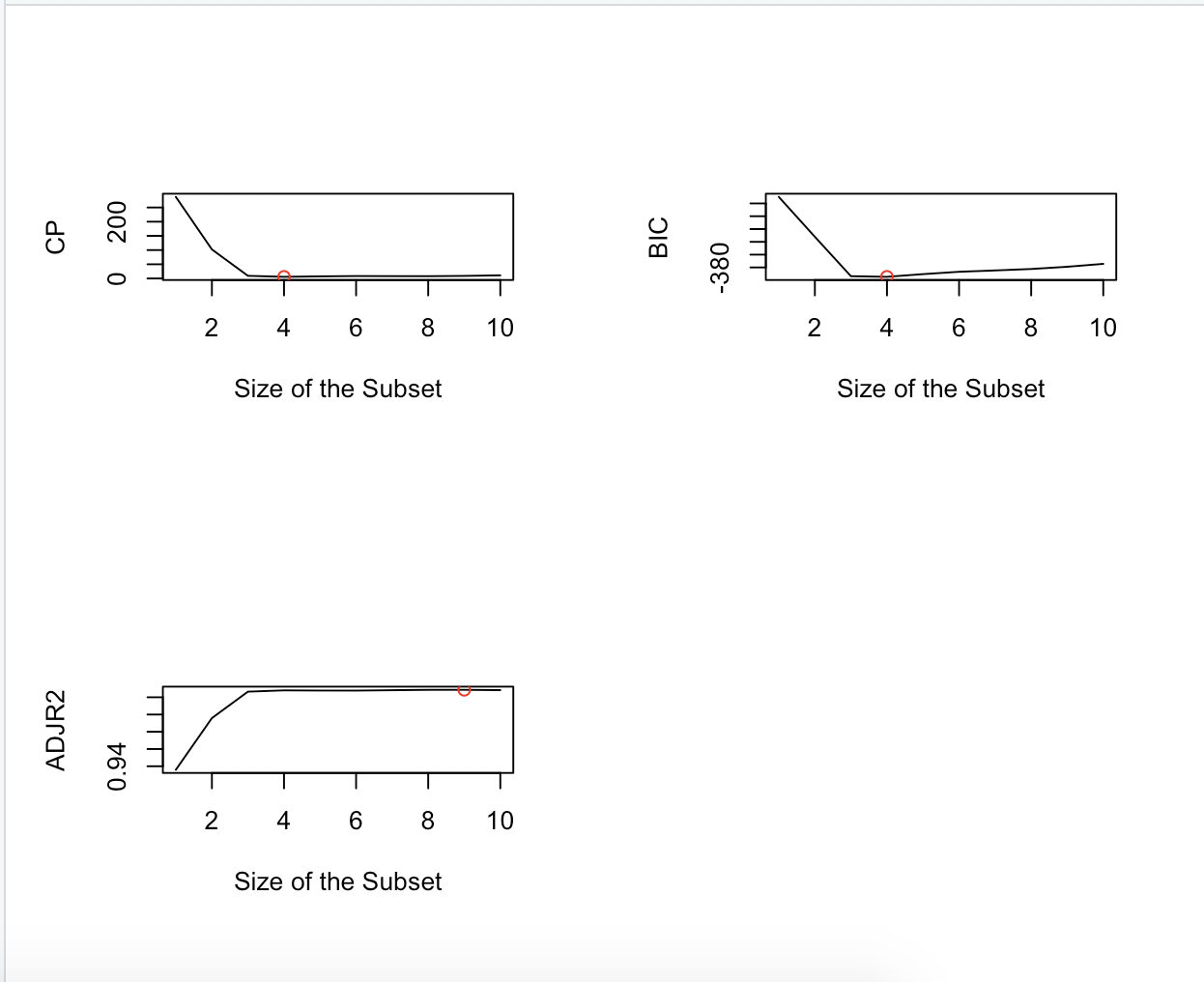
> plot(subsets.bwd.summary$bic, xlab = "Size of the Subset", ylab = "BIC", type ="l")

> points(4, subsets.bwd.summary$bic[4],col = "red")

> plot(subsets.bwd.summary$adjr2, xlab = "Size of the Subset", ylab ="ADJR2", type = "l")

> points(9, subsets.bwd.summary$adjr2[9],col = "red")

Output:

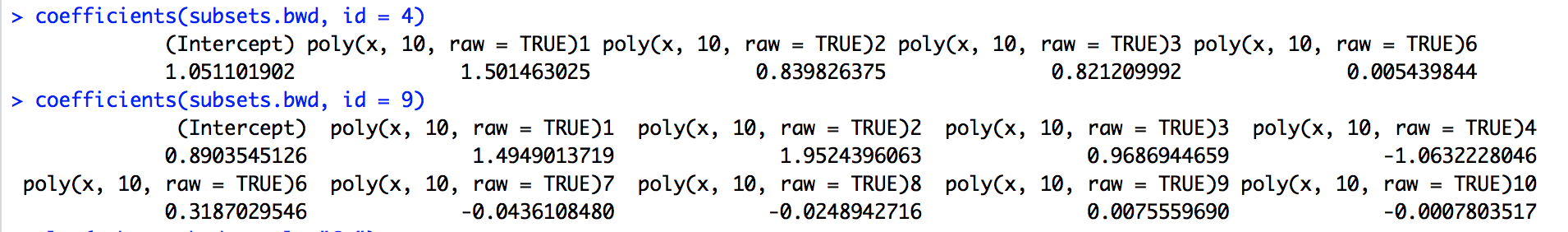


The best model according to backward selection when CP, BIC and ADJR2 are used is model 4 (CP, BIC) and model 9 (ADJR2).

Rcommand: > coefficients(subsets.bwd, id = 4)

> coefficients(subsets.bwd, id = 9)

Output:



The coefficients of model 4 and model 9 are given above. The coefficients of model 4 are close to what was given by best subset selection method and forward selection method except it has a X^6 term in the model selected by it, whereas the coefficients for model 9 contain all the variables uptill X^10 with the exception of X^5.

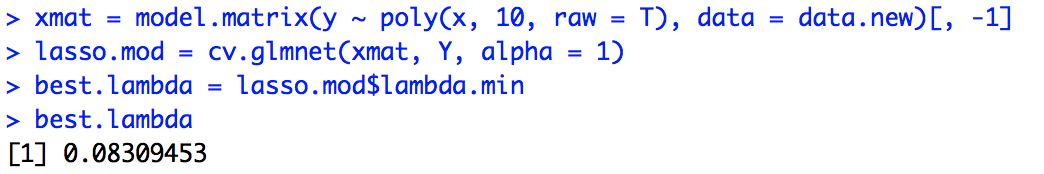
e) Rcommand: > xmat = model.matrix(y ~ poly(x, 10, raw = T), data = data.new)[, -1]

> lasso.mod = cv.glmnet(xmat, Y, alpha = 1)

> best.lambda = lasso.mod$lambda.min

> best.lambda

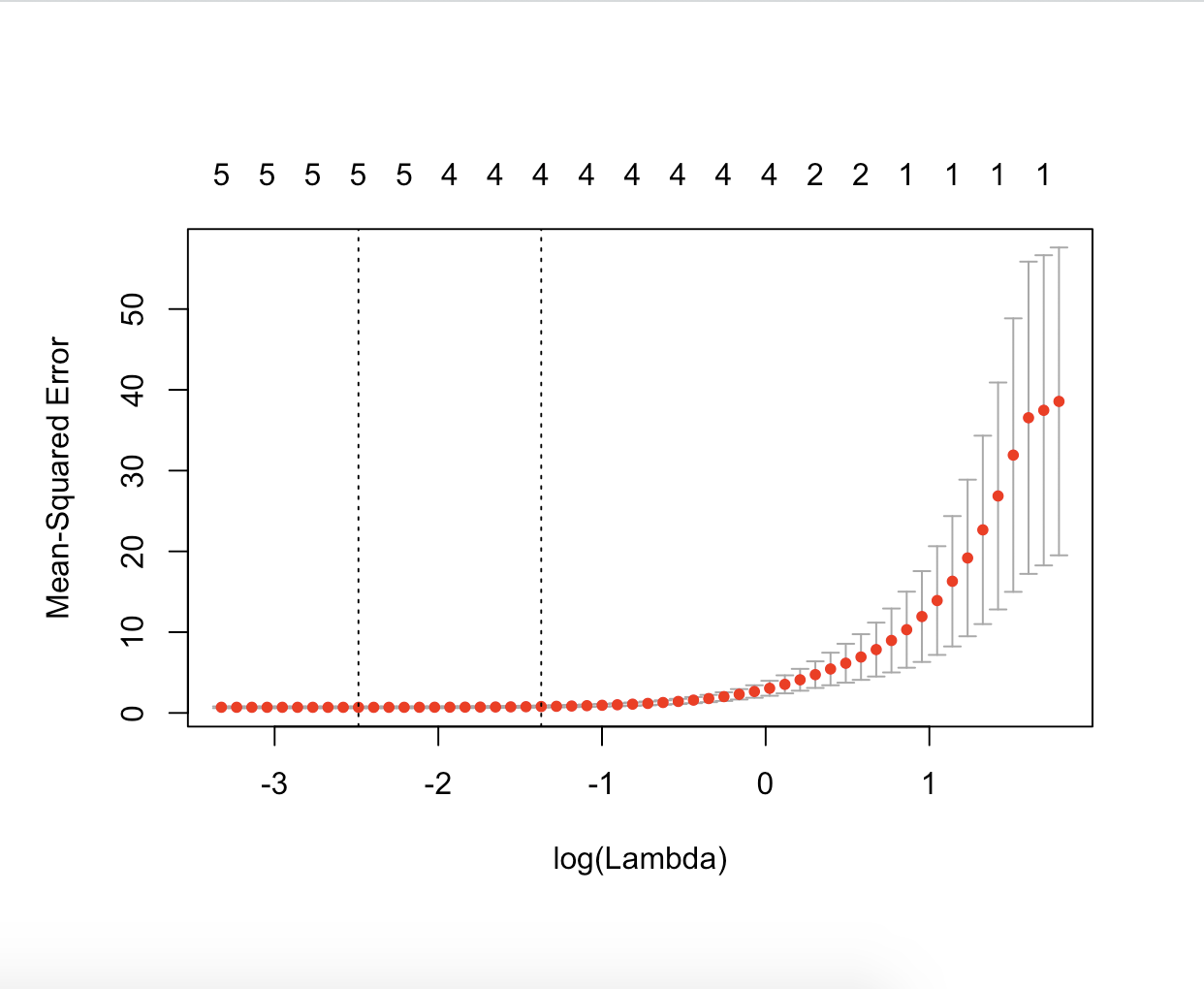
Output:



The best value of lambda as given by the cross-validation method is 0.08309453.

Rcommand: > plot(lasso.mod)

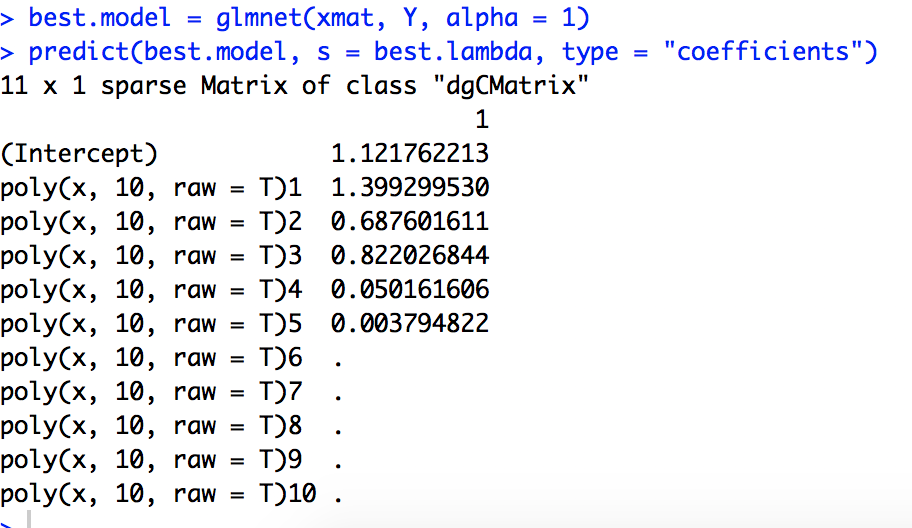
Output:



Rcommand: > best.model = glmnet(xmat, Y, alpha = 1)

> predict(best.model, s = best.lambda, type = "coefficients")

Output:



Lasso also picks X^5 and X^4 (insignificant coefficient). The values of the other coefficients are close to the real values.

f) Rcommand: > Y2 = 1 + X^7 + e

> data.new = data.frame(y = Y2, x = X)

> regfit.7 <- regsubsets(y~poly(x,10,raw=T), data=data.new, nvmax=10)

> reg.summary = summary(regfit.7)

> which.min(reg.summary$cp)

> which.min(reg.summary$bic)

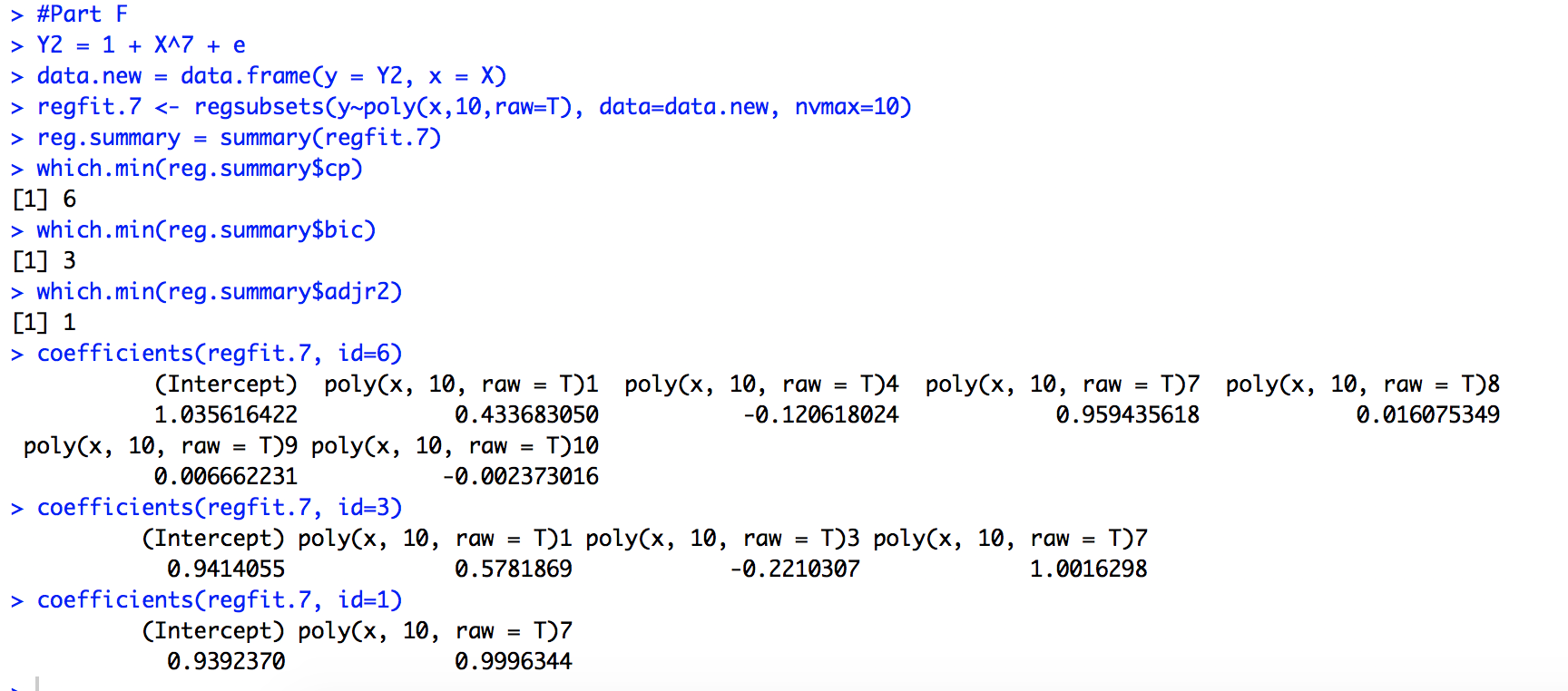
> which.min(reg.summary$adjr2)

> coefficients(regfit.7, id=6)

> coefficients(regfit.7, id=3)

> coefficients(regfit.7, id=1)

Output:



Here we can see that the model selected by the best subset selection method is model 6 (CP), model 3 (BIC) and model 1 (ADJR2). As we can see from the coefficients from the various models generated, model 1 is the best since its selects accurate coefficients and imitates the original model. Whereas the model 3, chooses X^3 coefficient as well, whereas model 6 is completely inaccurate and chooses a large number of extra variables.

Rcommand: > xmat = model.matrix(y ~ poly(x, 10, raw = T), data = data.new)[, -1]

> lasso.mod = cv.glmnet(xmat, Y, alpha = 1)

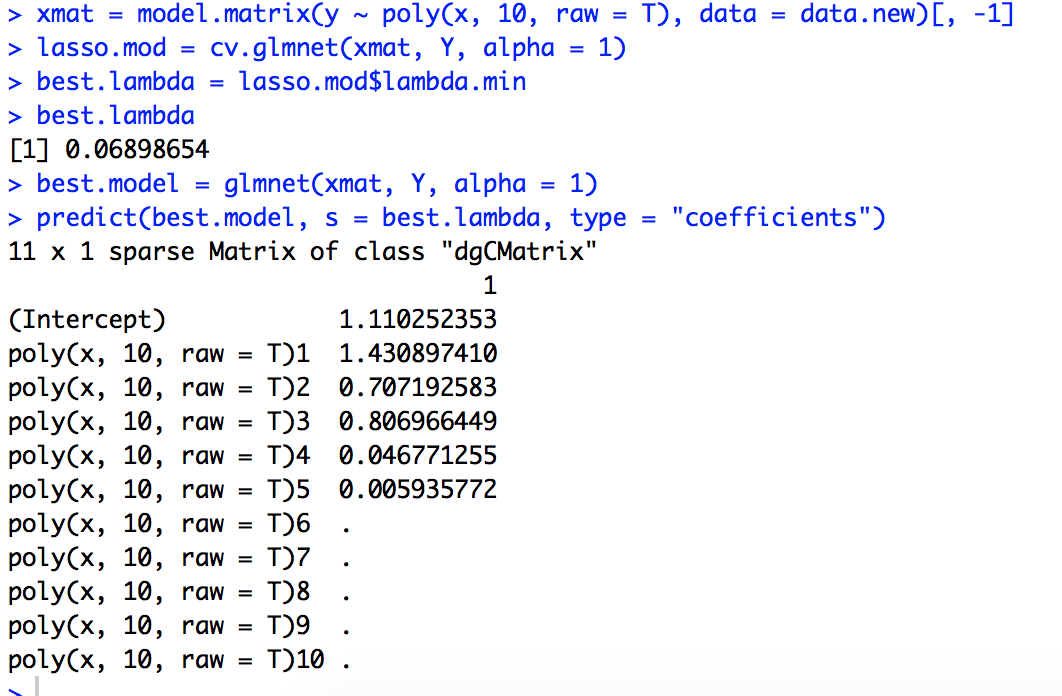
> best.lambda = lasso.mod$lambda.min

> best.lambda

> best.model = glmnet(xmat, Y, alpha = 1)

> predict(best.model, s = best.lambda, type = "coefficients")

Output:



The lambda generated from cross validation comes out to be 0.06898654. The model generated from lasso method is not so accurate as the one generated by best subset selection (ADJR2), since it picks variables X to X^5.

Hence after using the best subset selection method and lasso method, the best model was selected by best subdset selection (adjr2) as it was closest to the real model.

**Chapter 8, Question 8**

a) Rcommand: >set.seed(5462)

> library(ISLR)

> library(tree)

>library(randomForest)

>nrow(Carseats)

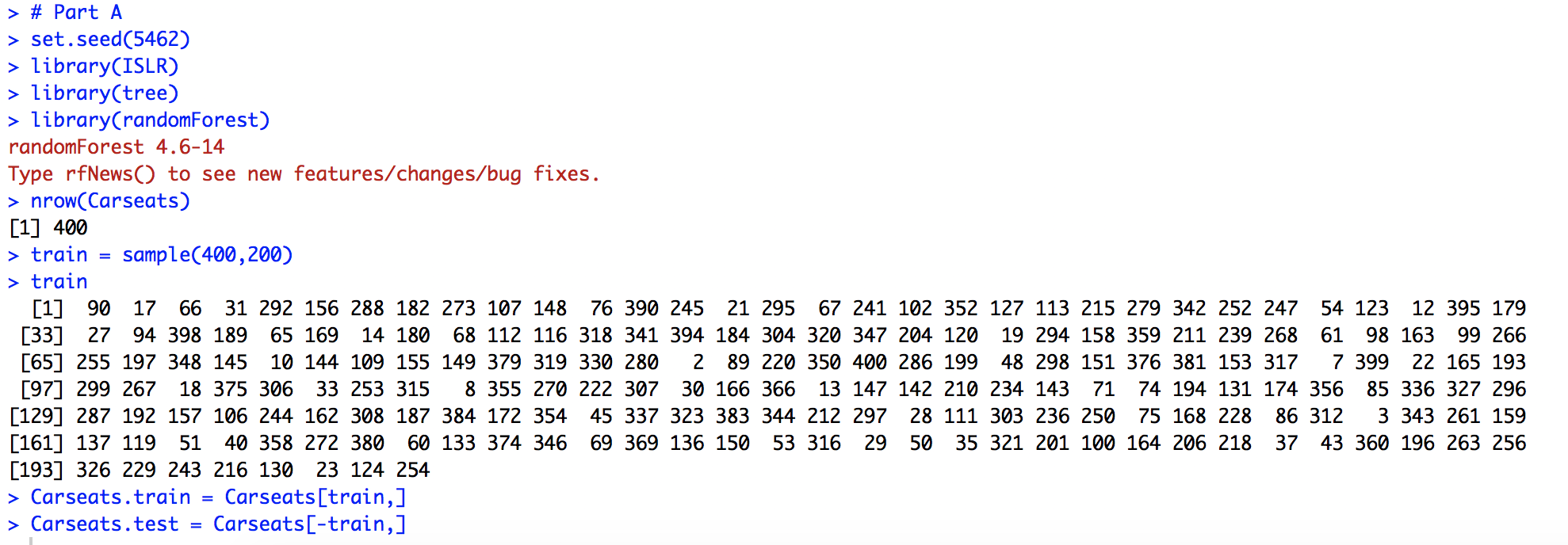
> train = sample(400,200)

> train

> Carseats.train = Carseats[train,]

> Carseats.test = Carseats[-train,]

Output:



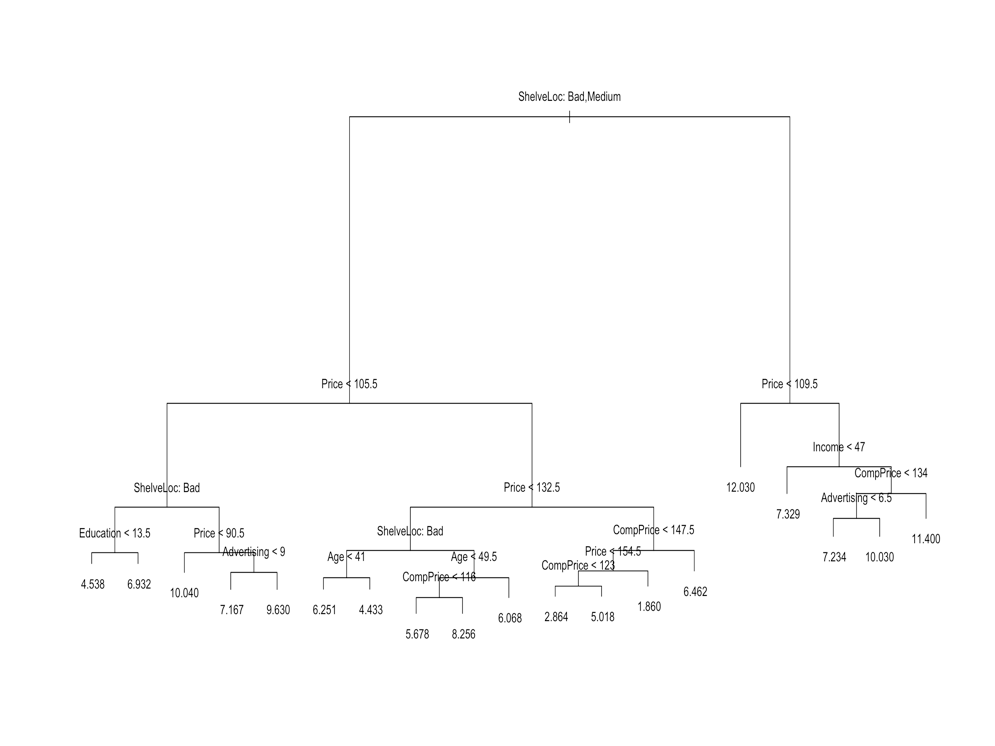
**The data is split into a training and a testing set on a 1:1 ratio.**

b) Rcommand: >carseats.tree = tree(Sales~.,data=Carseats.train)

>plot(carseats.tree)

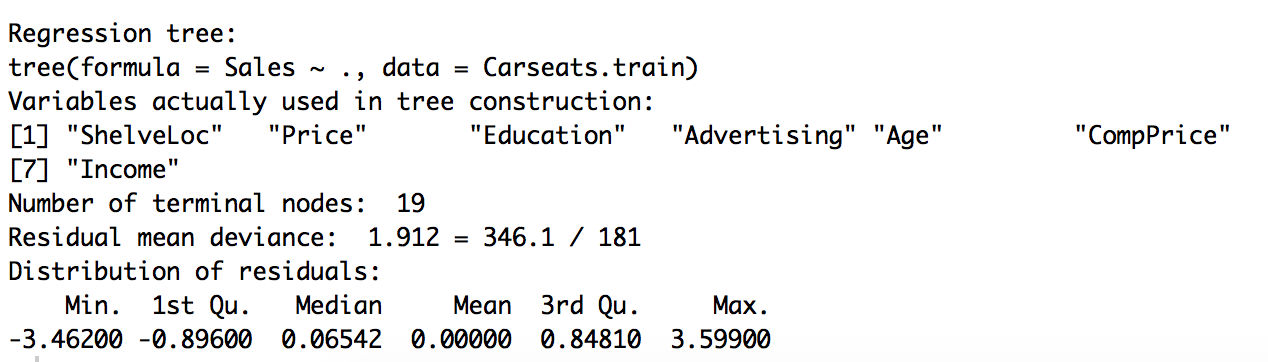
>text(carseats.tree, pretty = 0)

Output:



Rcommand: summary(carseats.tree)

Output:

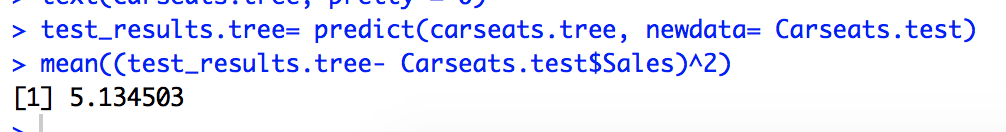


**The tree produced has 19 terminal nodes.**

Rcommand: >test\_results.tree= predict(carseats.tree, newdata= Carseats.test)

>mean((test\_results.tree- Carseats.test$Sales)^2)

Output:

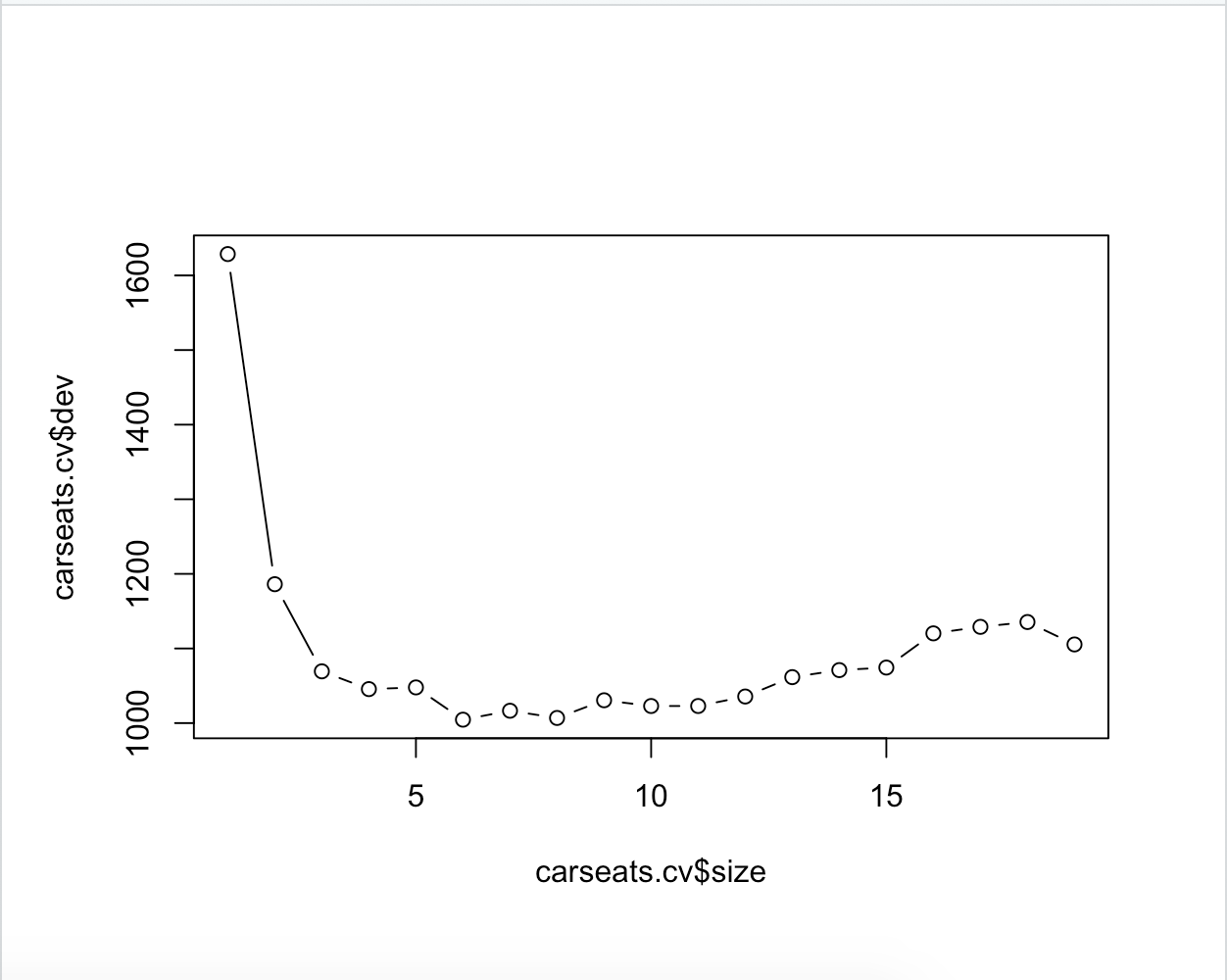


The test MSE comes out to be 5.134503.

c) Rcommand: > carseats.cv = cv.tree(carseats.tree)

>plot(carseats.cv$size, carseats.cv$dev, type="b")

Output:



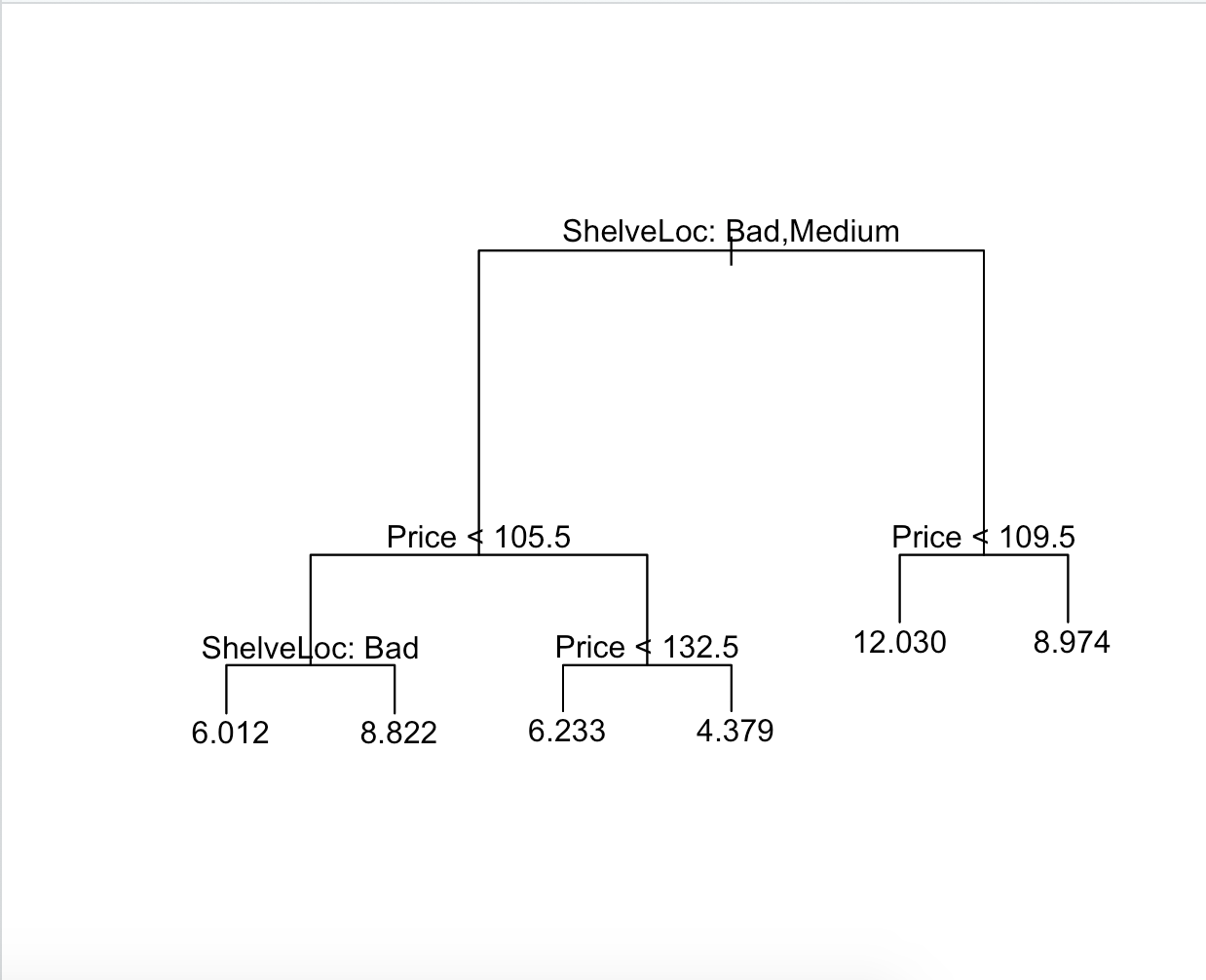
**As we can see, the deviance of is the lowest when the size is 6, and hence we can use 6 to prune the tree.**

Rcommand: > prune.carseats = prune.tree(carseats.tree, best = 6)

> plot(prune.carseats)

> text(prune.carseats, pretty = 0)

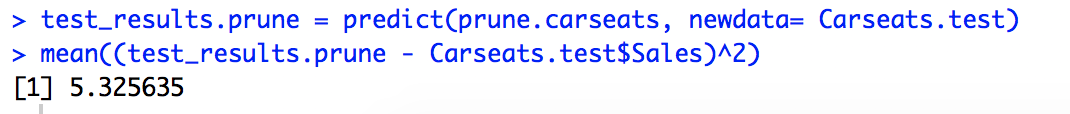
Output:



Rcommand: > test\_results.prune = predict(prune.carseats, newdata= Carseats.test)

>mean((test\_results.prune - Carseats.test$Sales)^2)

Output:



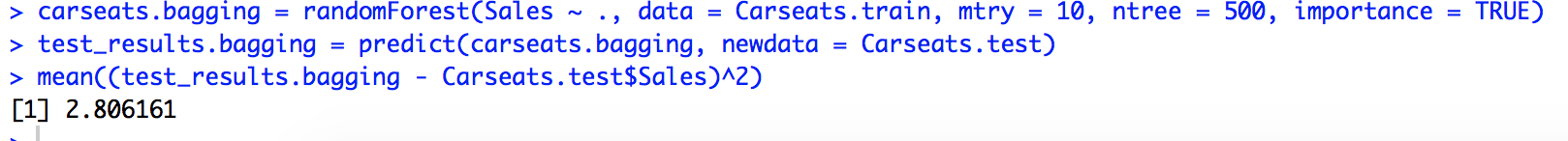
**The test MSE in the case of pruning the tree comes out to be 5.325635 which is higher than the case in which the tree was not pruned. Hence, the test MSE increases.**

d) Rcommand: > carseats.bagging = randomForest(Sales ~ ., data = Carseats.train, mtry = 10,ntree = 500, importance = TRUE)

> test\_results.bagging = predict(carseats.bagging, newdata = Carseats.test)

> mean((test\_results.bagging - Carseats.test$Sales)^2)

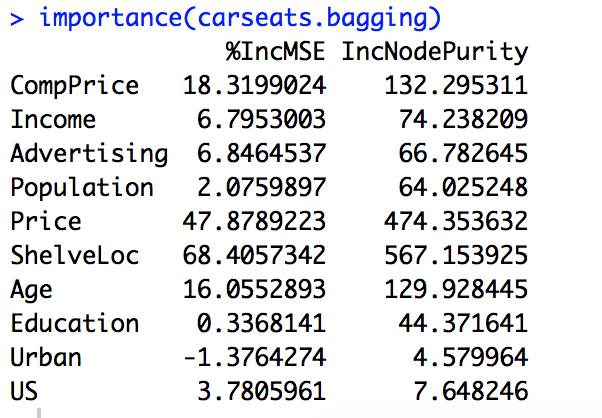
Output:



The test MSE when the bagging approach is used comes out to be **2.806161.**

Rcommand: > importance(carseats.bagging)

Output:



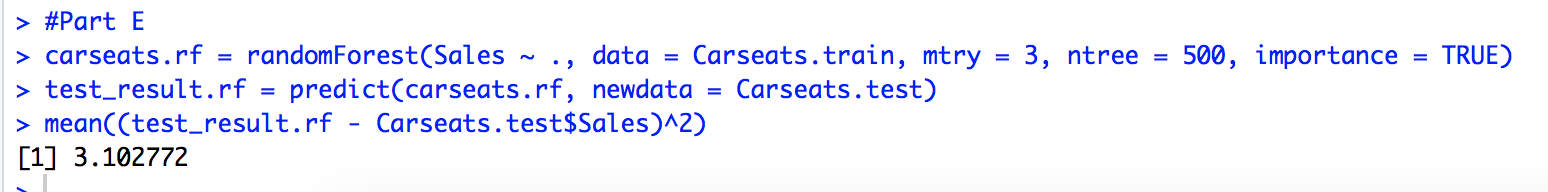
Using the importance() function we can determine that the most important variables are **“Price”, “Shelveloc” and “CompPrice”.**

e) Rcommand: > carseats.rf = randomForest(Sales ~ ., data = Carseats.train, mtry = 3, ntree = 500, importance = TRUE)

> test\_result.rf = predict(carseats.rf, newdata = Carseats.test)

> mean((test\_result.rf - Carseats.test$Sales)^2)

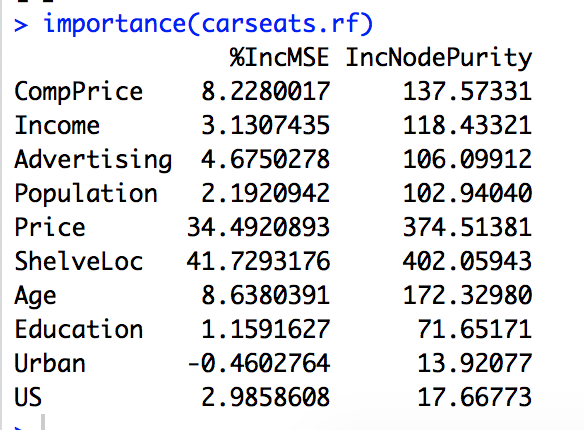
Output:



The test MSE in this case where Random Forest approach is used comes out to be 3.102772. The test error rate becomes worse when the value of m changes. **The test MSE increases when the value of m is decreased from 10 to 3.**

Rcommand: > importance(carseats.rf)

Output:



The most important variables according to the importance() function come out to be “**Price” and “ShelveLoc”** when the Random Forest approach is used.