Stat 897 Spring 2017 Data Analysis Assignment 4

Penn State

Due September 17, 2017

1. Use the College data found in the ISLR library. It contains 18 variables for 777 different universities and colleges in the US. The list of variables and their full description can be found on p. 54 of the book.

(a) Load the dataset College. Our objective is to predict the number of applications received using the other variables in the data set.

```
library(leaps)
library(ISLR)
data("College")
```

(b) Split your data set into a training set containing 100 observations and a test set containing the rest of the observations. For reproducibility of results use set.seed() with a value of your choosing.

```
set.seed (1)
trainingRows=sample (nrow(College), 100, replace = FALSE)
train = College[trainingRows,]
test = College[-trainingRows,]
```

(c) Perform best subset selection on the training set, and plot the training set MSE associated with the best model of each size.

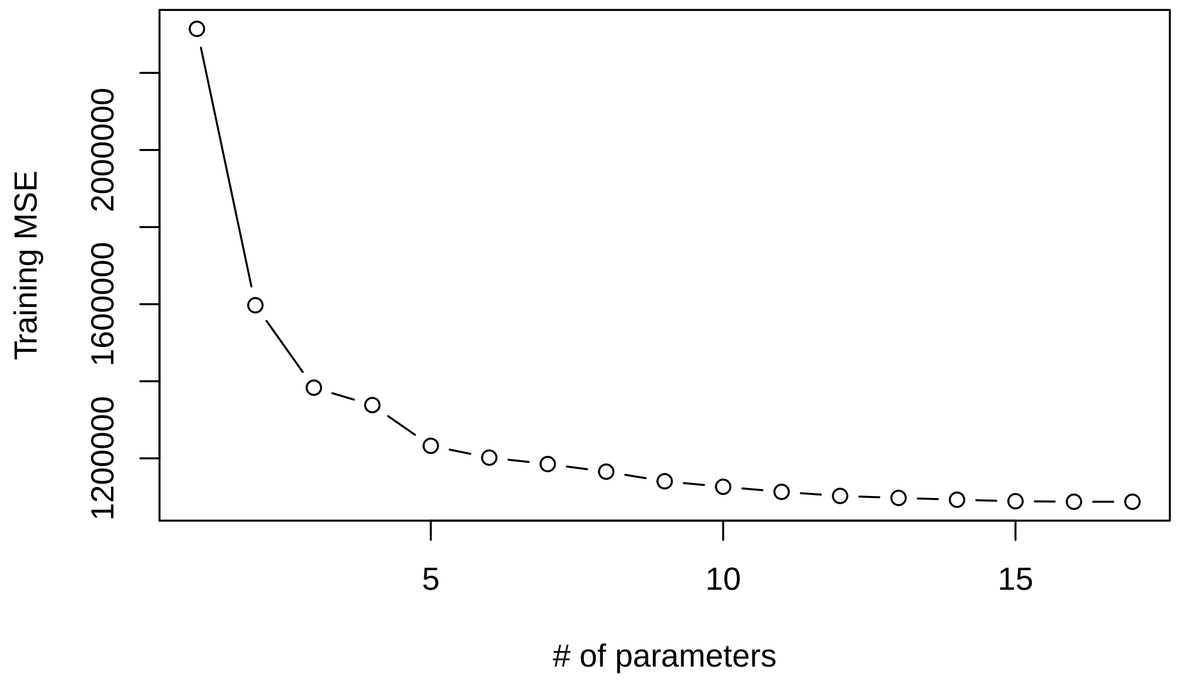
```
dim(train)
```

```
## [1] 100  18
```

```
names(train)
```

```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "Top10perc"
##  [6] "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"    "Room.Board"
## [11] "Books"       "Personal"    "PhD"         "Terminal"    "S.F.Ratio"
## [16] "perc.alumni" "Expend"      "Grad.Rate"
```

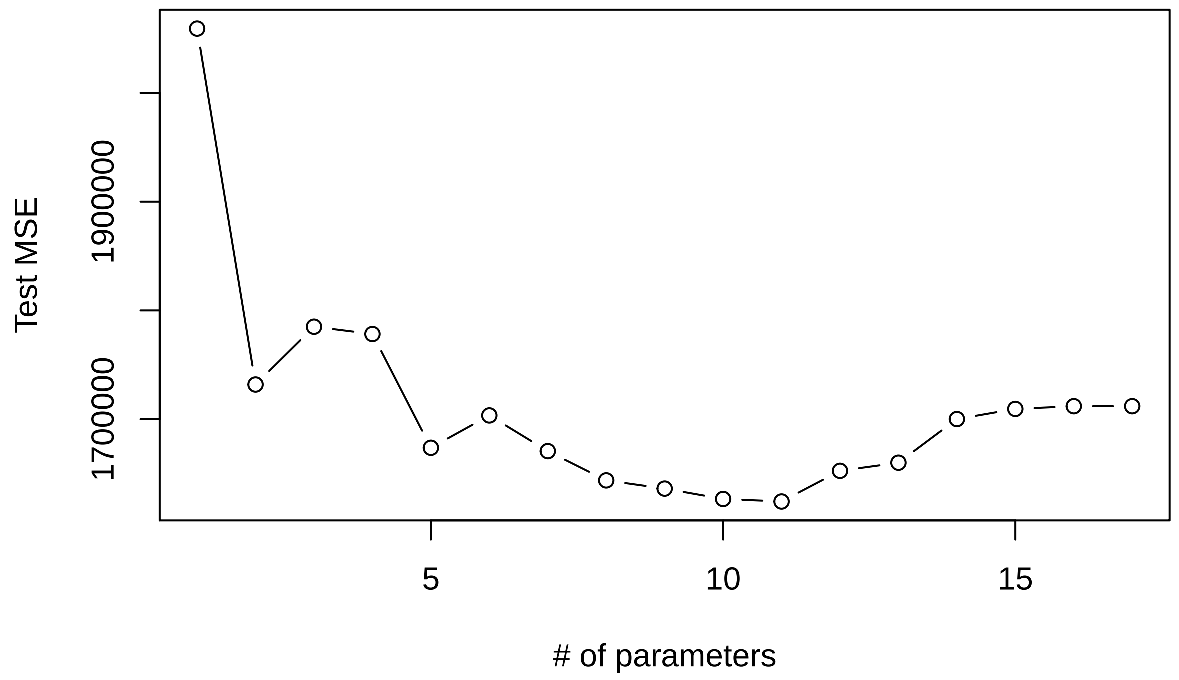
```
regfit.best=regsubsets (Apps~.,data=train, nvmax=17)
train.mat=model.matrix (Apps~.,data=train)
train.val.errors =rep(NA ,17)
for(i in 1:17) {
    coefi=coef(regfit.best ,id=i)
    pred=train.mat [,names(coefi)]%*% coefi
    train.val.errors [i]= mean(( train$Apps-pred)^2)
}
plot(train.val.errors ,type='b', xlab='# of parameters', ylab='Training MSE')
```



We can see that the training MSE continues to decrease but the rate of change drops significantly after the first few terms. This behavior is expected and is the reason why we can't use the training MSE to compare between models of different sizes. Looking at how steep is the decrease - the steepest decrease happens till 5 and after 11 is almost flat.

(d) Plot the test set MSE associated with the best model of each size.

```
test.mat=model.matrix (Apps~.,data=test)
test.val.errors =rep(NA ,17)
for(i in 1:17){
    coefi=coef(regfit.best ,id=i)
    pred=test.mat [,names(coefi)] %*% coefi
    test.val.errors [i]= mean(( test$Apps-pred)^2)
}
plot(test.val.errors ,type='b', xlab='# of parameters', ylab='Test MSE')
```



The minimum test MSE if for model with 11 parameters

(e) For which model size does the test set MSE take on its minimum value? Comment on your results.

```r
which.min(test.val.errors)
```

```
## [1] 11
```

The test set MSE takes on the minimum value for model 11. As we discussed before We can see that the training MSE continues to decrease but the rate of change drops significantly after the first few terms. On the other hand the test MSE first drops and then starts to increase indicating overfitting when the number of parameters is >= 12. Also when the parameters are < 11 we see indications of underfitting.

Please check that none of the best models is the intercept only model, or the model with ALL predictors. If they are, try using a different seed value to avoid it.

(f) Fit a regression model with all features to the full data containing 777 observations. Let the regression coefficients for this model be denoted by β_j. Let $\hat{\beta}_j^r$ be the estimated regression coefficient for the best model containing r features. Create a plot displaying

$$\sqrt{\sum_{j=1}(\beta_j - \hat{\beta}_j^r)^2}$$

for a range of values of r. Comment on what you observe. How does this plot compare to the test MSE plot from (d).

```
dim(College)
```

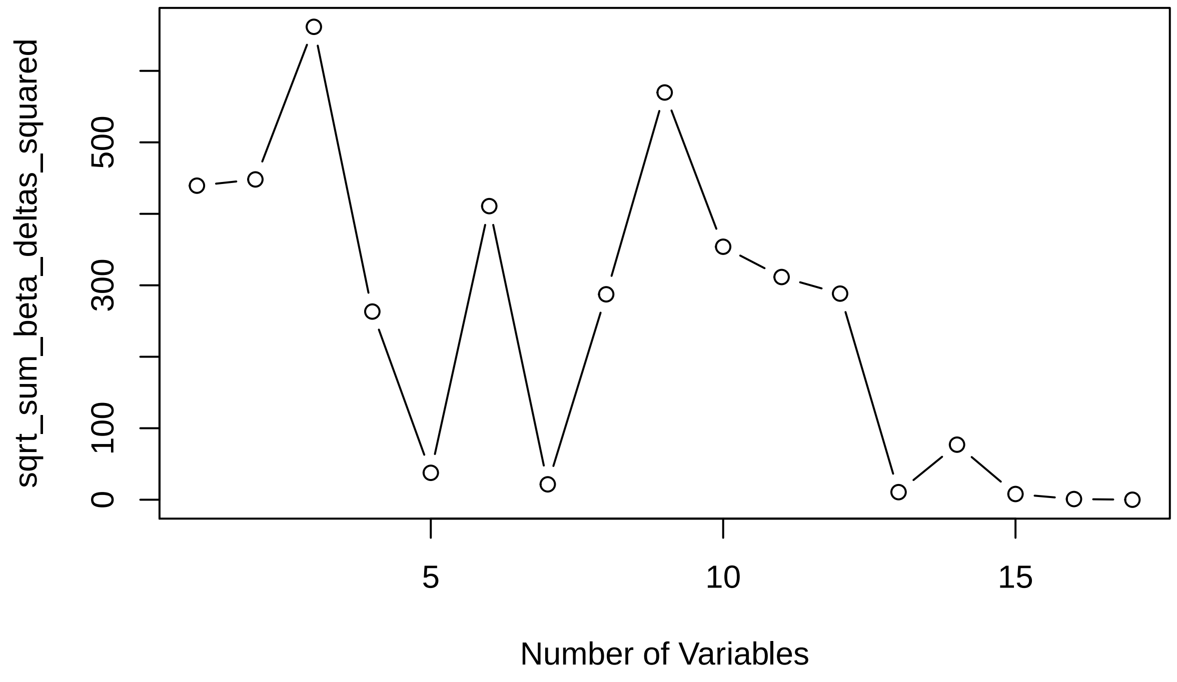
```
## [1] 777  18
```

```
lm.fit=lm(Apps~., data=College)
regfit.best=regsubsets (Apps~.,data=College, nvmax=17)
beta_j = coef(lm.fit)
beta_j
```

```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc
## -445.08412616 -494.14896913    1.58580720   -0.88069035   49.92627615
##     Top25perc    F.Undergrad   P.Undergrad      Outstate    Room.Board
##  -14.23447911     0.05739022    0.04444654   -0.08587004    0.15102708
##         Books      Personal           PhD      Terminal     S.F.Ratio
##    0.02089712     0.03110491   -8.67849712   -3.33065614   15.38960682
##   perc.alumni        Expend     Grad.Rate
##    0.17866507     0.07789731    8.66762545
```

```
sqrt_sum_beta_deltas_squared=rep(NA ,17)
for(i in 1:17){
  beta_hat_j=coef(regfit.best ,id=i)
  sqrt_sum_beta_deltas_squared [i] = sqrt(sum(
    (beta_j[names(beta_j)] - beta_hat_j[names(beta_j)])^2,
                 na.rm = TRUE))
}

plot(sqrt_sum_beta_deltas_squared, xlab ="Number of Variables", type = 'b')
```



```r
which.min(sqrt_sum_beta_deltas_squared)
```

```
## [1] 17
```

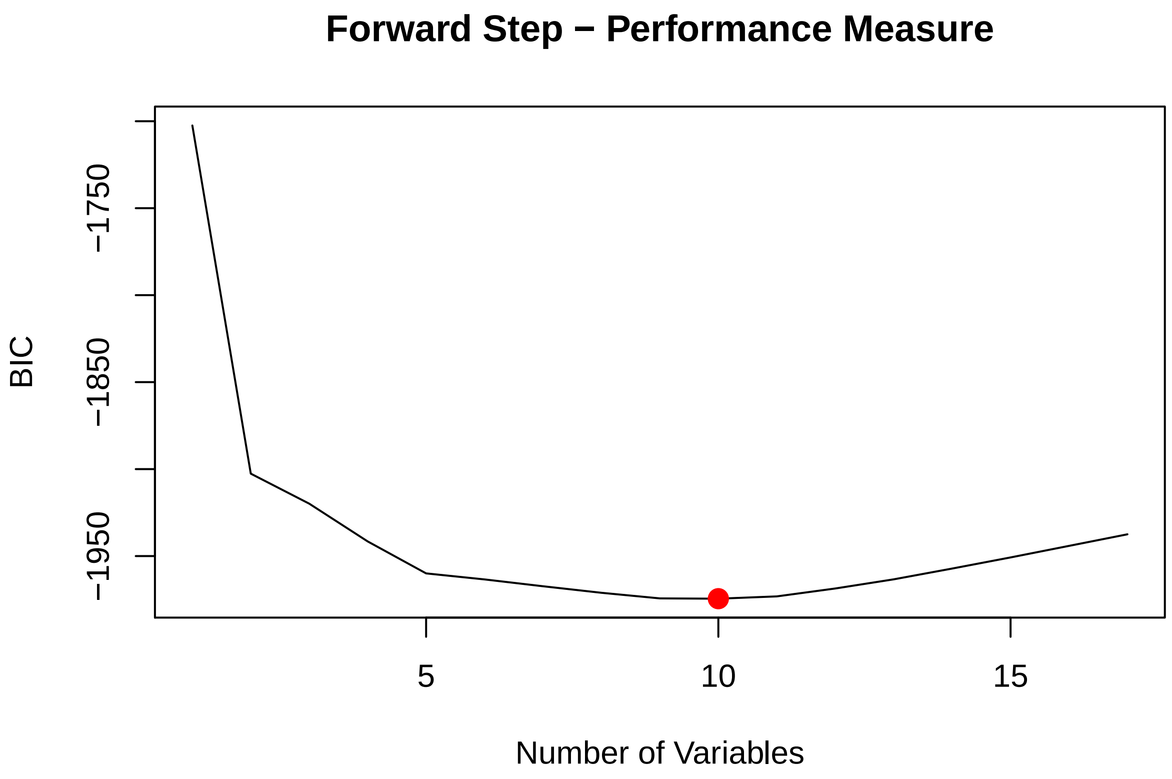
We see that as more terms are added to the best model, additional terms are added to the equation - we observe high volatility in the initial part of the plot. As we reach towards the max the volatility subsides and the value rapidly moves towards 0. At 17 parameters we get 0 as expected.

(g) Now use forward and backward stepwise selection with the BIC and AIC to select models (so you will get up to four best models). How do the results compare to what you obtained above in part (c) and (d)?

```r
# Using all the data in the data set
regfit.fwd=regsubsets (Apps~.,data=College, nvmax =17, method='forward')
reg.summary = summary (regfit.fwd)
plot(reg.summary$bic, xlab ="Number of Variables",ylab="BIC", type = 'l', main = 'Forward Step - Perforn
which.min (reg.summary$bic )
```

```
## [1] 10
```

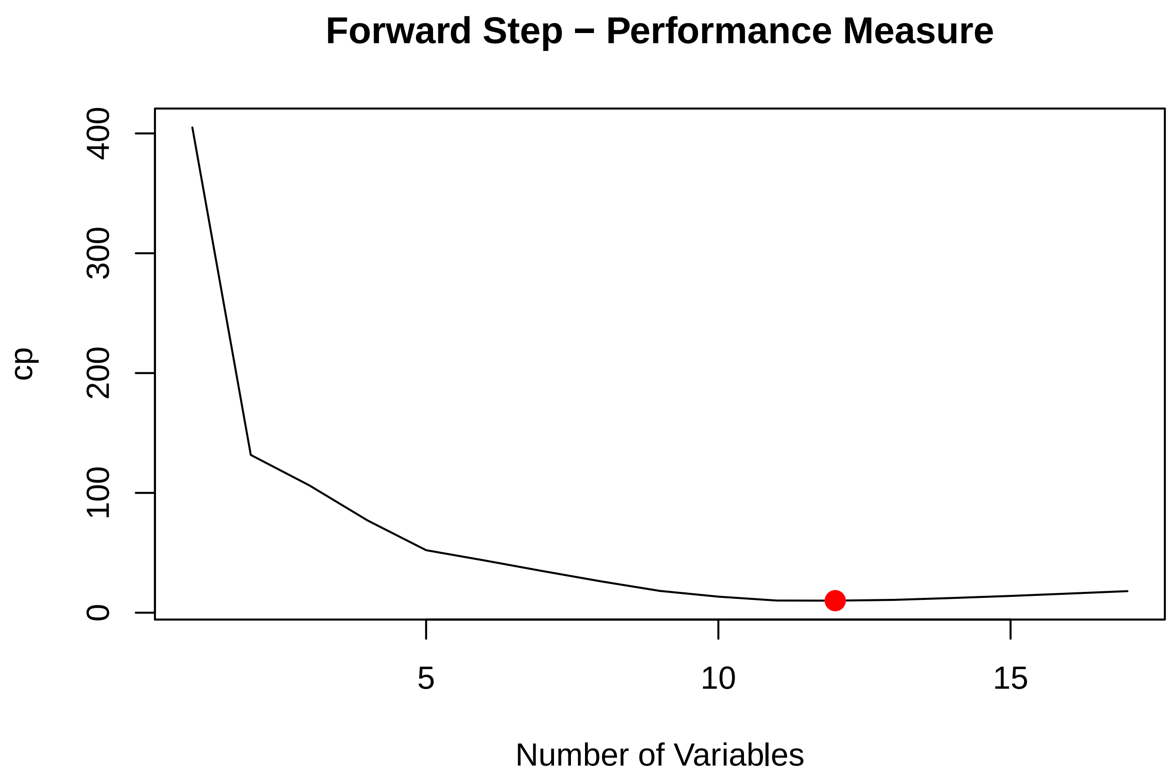
```r
points (10, reg.summary$bic[10], col ="red",cex =2, pch =20)
```



```r
plot(reg.summary$cp, xlab ="Number of Variables",ylab="cp", type = 'l', main = 'Forward Step - Performar
which.min (reg.summary$cp )
```

```
## [1] 12
```

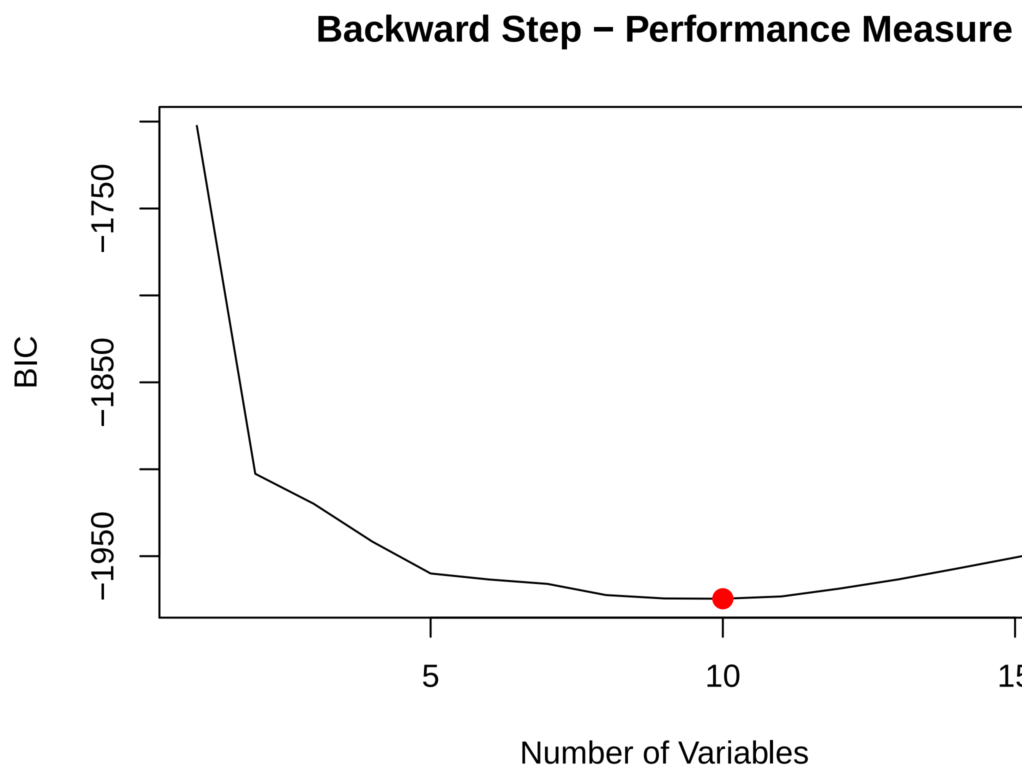
```
points (12, reg.summary$cp[12], col ="red",cex =2, pch =20)
```



```
regfit.bwd=regsubsets (Apps~.,data=College, nvmax =17, method='backward')
reg.summary =  summary(regfit.bwd)
plot(reg.summary$bic, xlab ="Number of Variables",ylab="BIC", type = 'l', main = 'Backward Step - Perfo
which.min (reg.summary$bic )
```

```
## [1] 10
```

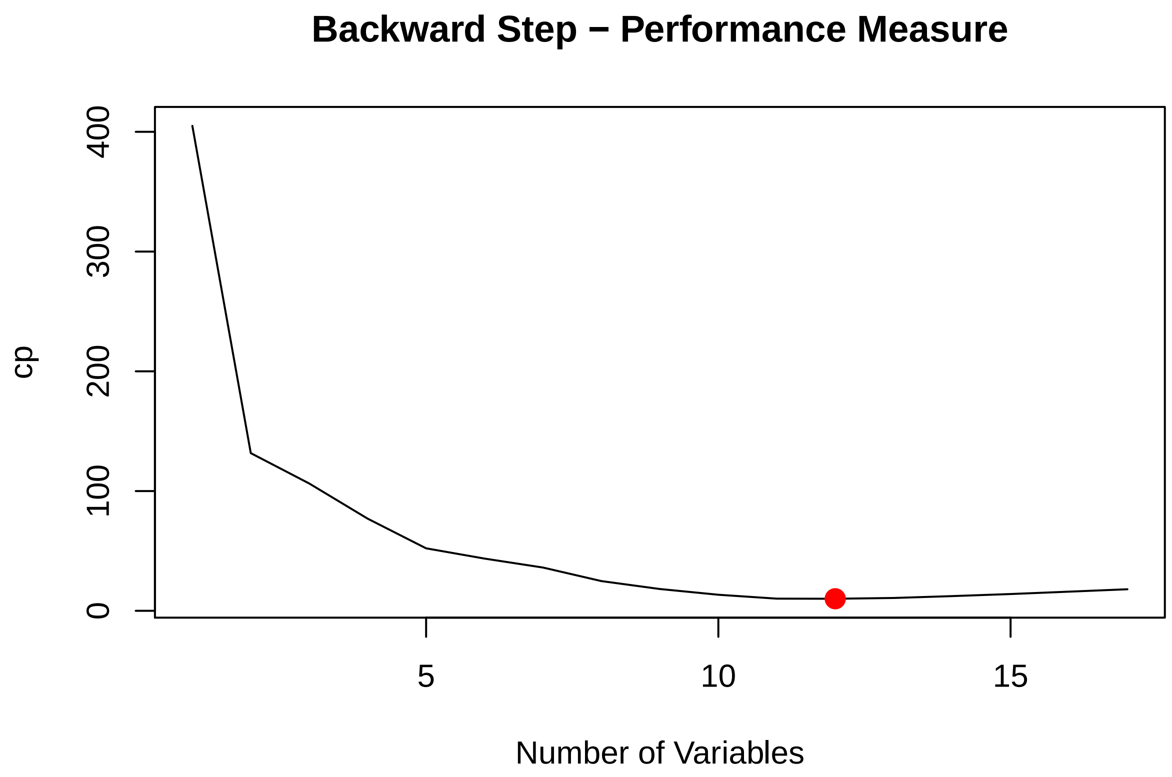
```
points (10, reg.summary$bic[10], col ="red",cex =2, pch =20)
```



```
plot(reg.summary$cp, xlab ="Number of Variables",ylab="cp", type = 'l', main = 'Backward Step - Performa
which.min (reg.summary$cp )
```

```
## [1] 12
```

```
points (12, reg.summary$cp[12], col ="red",cex =2, pch =20)
```



For both forward and backward techniques, we select the same models. BIC chooses 10 while CP/AIC will choose 12 The choices are different from the test MSE (selected 11) and training MSE is lowest for 17 ofcourse but the rate of decrease slows down after 5 and almost ends after 11.