HW8

Ben Tankus

2/23/2021

1. (2 points) (ISLR 2.4 Exercise #1, page 52) For each of the following parts, indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answers.

(a) The sample size n is extremely large, and the number of predictors p is small.

Flexile is usually BETTER than inflexible: with the small number of predictors it's unlikely that we will have an overfitting senerio

(b) The number of predictors p is extremely large, and the number of observations n is small.

Flexible is usually WORSE than inflexible: With the high-dimentional "large p" the model already has a tendency to be too flexible with the many predictor variables, and overfit.

(c) The relationship between the predictors and response is highly non-linear.

Flexibile is usually BETTER than inflexible: If the relationship is non-linear, a model will have to be flexible to fit the data, not linear.

(d) The variance of the error terms, i.e., sigmasq = Var(epsilon), is extremely high.

Flexible is usually WORSE than inflexible: If the variance is high, the model should be flexible to accurately respond.

2. (4 points) (ISLR 5.4 Exercise #8, page 201) We will now perform cross-validation on a simulated dataset.

(a) Generate a simulated data set as follows:

```
set.seed(1)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
df = data.frame(x,y)

df['x2'] <- df['x']^2
df['x3'] <- df['x']^3
df['x4'] <- df['x']^4
```

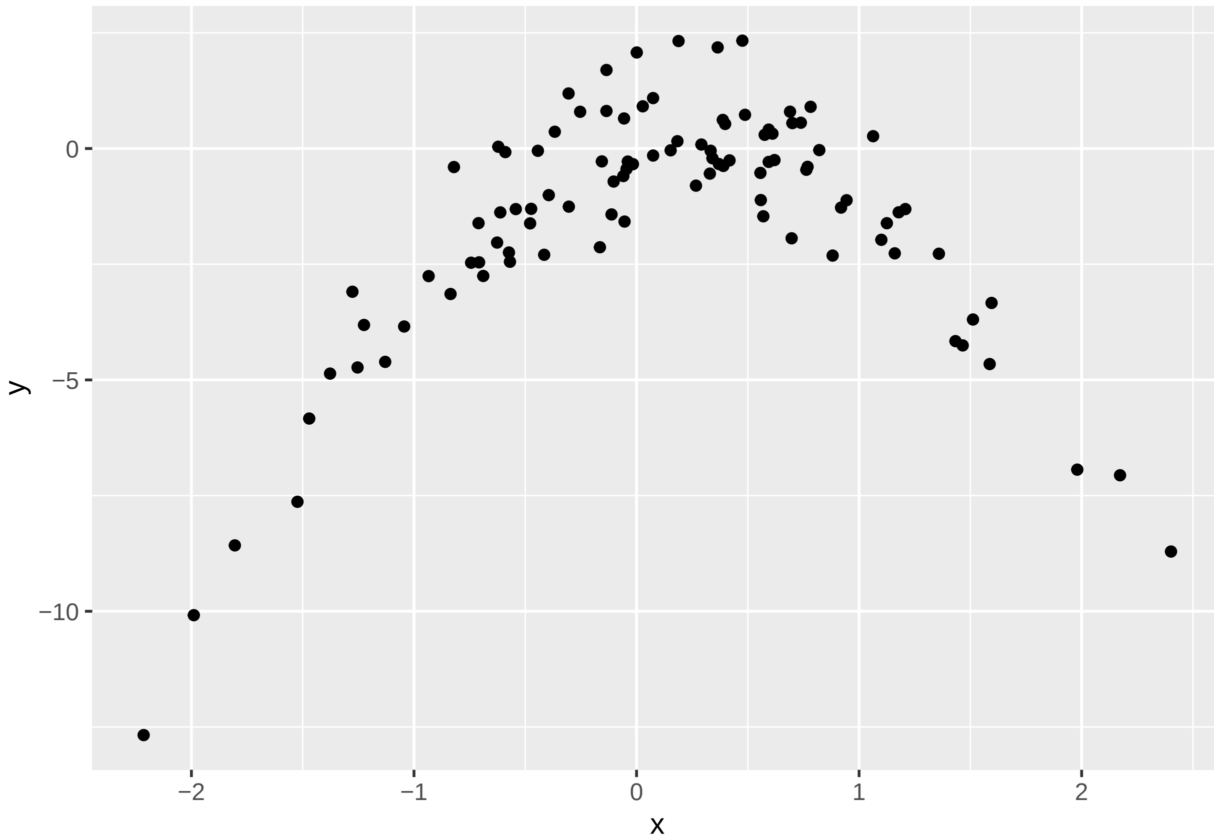
In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

In this model n is 100 and p is 2 (x and x^2)

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

(b) Create a scatter plot of X against Y using the data you generated above. Comment on what you see.

```
qplot(x, y)
```



It looks like there is a clear negative, non-linear relationship between x and y.

(c) Set a random seed, and then compute the leave-one-out cross-validation (LOOCV) errors that result from fitting the following four models using least squares:

- Y = beta0 + beta1X + epsilon Error is 7.2881616
- Y = beta0 + beta1X + beta2X2 + epsilon Error is 0.9374236
- Y = beta0 + beta1X + beta2X2 + beta3X3 + epsilon Error is 0.9566218
- Y = beta0 + beta1X + beta2X2 + beta3X3 + beta4X4 + epsilon
  Error is 0.9539049

Note that you may find it helpful to use the data.frame() function to create a single data set containing both X and Y .

```
set.seed(1)

cv.error <- rep(0,4)
```

```r
for(i in 1:4){
  print(i)
  glm.fit <- glm(y ~ ., data= df[,1:(i+1)])
  cv.error[i] <- cv.glm(df[,1:(i+1)], glm.fit)$delta[1]
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

```r
cv.error
```

```
## [1] 7.2881616 0.9374236 0.9566218 0.9539049
```

There is a large drop in MSE values from a linear model (7.29) to the second order polynomial model (0.937). From this drop, I can reccomend fitting this data using the second order polynomial of X Y = beta0 + beta1X + beta2X2 + epsilon.

(d) Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why or why not?

```r
set.seed(2)
x.d <- rnorm(100)
y.d <- x.d - 2*x.d^2 + rnorm(100)
df.d = data.frame(x.d,y.d)
df.d['x2.d'] <- df.d['x.d']^2
df.d['x3.d'] <- df.d['x.d']^3
df.d['x4.d'] <- df.d['x.d']^4
cv.error <- rep(0,4)

for(i in 1:4){
  print(i)
  glm.fit <- glm(y.d ~ ., data= df.d[,1:(i+1)])
  cv.error[i] <- cv.glm(df.d[,1:(i+1)], glm.fit)$delta[1]
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

```r
cv.error
```

```
## [1] 9.858301 1.004410 1.018030 1.035601
```

The MSE values are different but the best model pattern remains the same (second order model). This is because the transformations from X to Y is what creates the relationship between X and Y, not the values themselves. The second order polynomial of X Y = beta0 + beta1X + beta2X2 + epsilon is still the model with the best MSE.

(e) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

The second model *Y = beta0 + beta1X + beta2X2 + epsilon* has the lowest LOOCV error. This is what I expected because there is a clear second order polynomial shape to the data.This matches the second order equation selected.

(f) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

```r
summary(glm.fit)
```

```
##
## Call:
## glm(formula = y.d ~ ., data = df.d[, 1:(i + 1)])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.08635  -0.78633   0.06263   0.76755   2.11807
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03008    0.16417  -0.183    0.855
## x.d          0.98184    0.20180   4.865 4.53e-06 ***
## x2.d        -1.96901    0.23512  -8.374 4.86e-13 ***
## x3.d        -0.01033    0.06292  -0.164    0.870
## x4.d         0.00451    0.05212   0.087    0.931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.001533)
##
##     Null deviance: 1013.122  on 99  degrees of freedom
## Residual deviance:   95.146  on 95  degrees of freedom
## AIC: 290.81
##
## Number of Fisher Scoring iterations: 2
```

The coefficients of x and x^2 in the model are both statistically significantly different from zero, while x^3 and x^4 are not. This aligns with the crossvalidation method suggesting the p = 2 model to be best.

3. (4 points) (Based on ISLR 6.8 Exercise #11, page 264 — Predicting crime rates in Boston data.) The Boston data set is in the MASS package, you'll need to load that first.

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.3
```

```r
?Boston
```

```
## starting httpd help server ... done
```

```r
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

Your job is to build a regression model to predict the crime rate (crim) in Boston suburbs based on the other provided variables.

Your solution should include: - A brief exploratory analysis (some summary statistics, and a few plots of any obvious relationships). - A description of the set of regression models you considered. - A description of how the models were evaluated. - A summary of one (or a few) models that based on your analysis are the best among those you considered.

```r
summary(Boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
```

```
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
dev.new(width = 100, height = 100, unit = "in")

plot(Boston)


boxplot(x = as.list(as.data.frame(Boston)), las = 2, main = 'All columns')
boxplot(x = as.list(as.data.frame(Boston[,-c(4,5,6,7,10,12)])), las = 2, main = 'Extra small and Extra l


# Check corelation
corBucket <- round(cor(Boston),3)


# Get lower triangle of the correlation matrix
  get_lower_tri<-function(corBucket){
    corBucket[upper.tri(corBucket)] <- NA
    return(corBucket)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(corBucket){
    corBucket[lower.tri(corBucket)]<- NA
    return(corBucket)
  }

upper_tri <- get_upper_tri(corBucket)

melted_cormat <- melt(upper_tri, na.rm = TRUE)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill = value)) +
  geom_tile() +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = 'red', high = "green", mid = 'white', midpoint = 0, limit = c(-1,1), space
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 1, size = 8))

qplot(rad, crim, data = Boston, main = 'Rad vs Crim')
qplot(dis, crim, data = Boston, main = 'Dis vs Crim')
qplot(medv, crim, data = Boston, main = 'Medv vs Crim')
```

It looks like the tax and black columns are much higher than the others, and crime, zoned lots, age, and median value all have sizable skew to their distributions. Many columns also have strong multi-colinearity such as distance to employment centers on non-retail business acres, nitrogen oxides, and age as well as tax on accessibility to radial highways. It also looks as though the charles river dummy variable is not related to any metric. I analysed a few specific promising metric's scatter plots below.

Rad vs crime shows that there is a small amount of crime in the early indexes, but as the scale increases there is a huge jump in crime around 24.

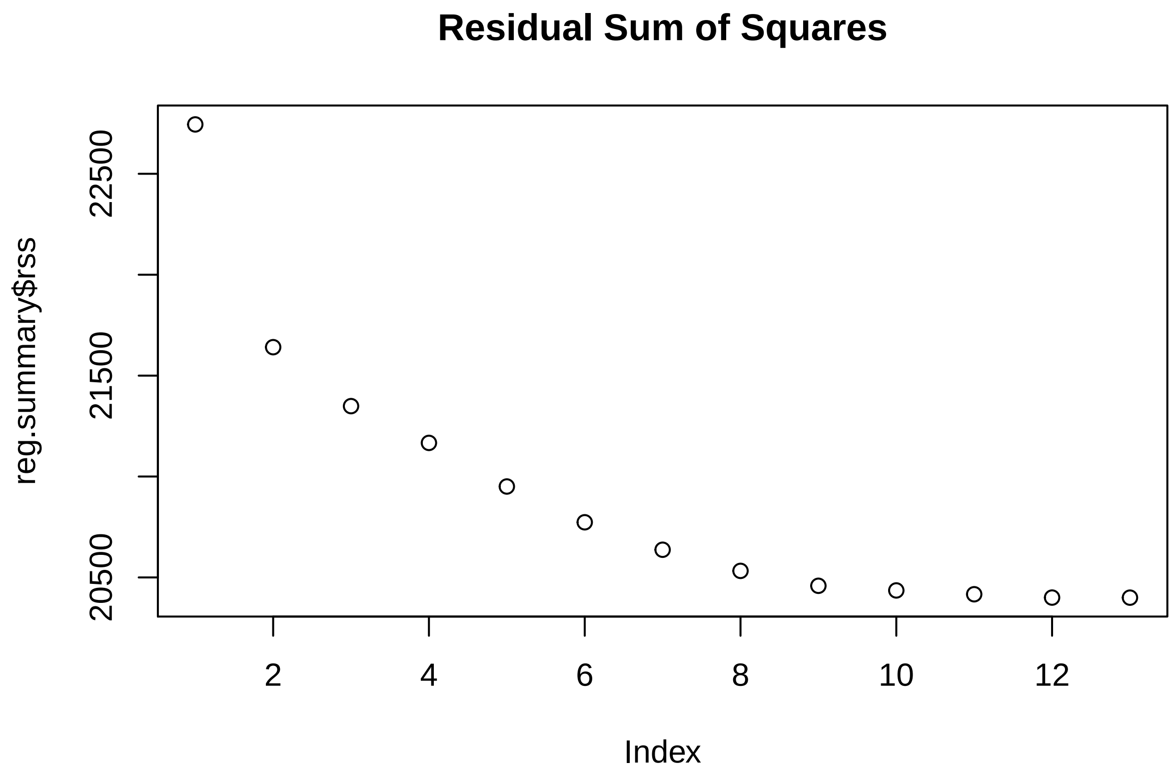
Dis vs crime shows that as the distance to the employment centers decreases the crime increases.

Medv vs crime shows as the median value of the homes increases, the crime decreases.

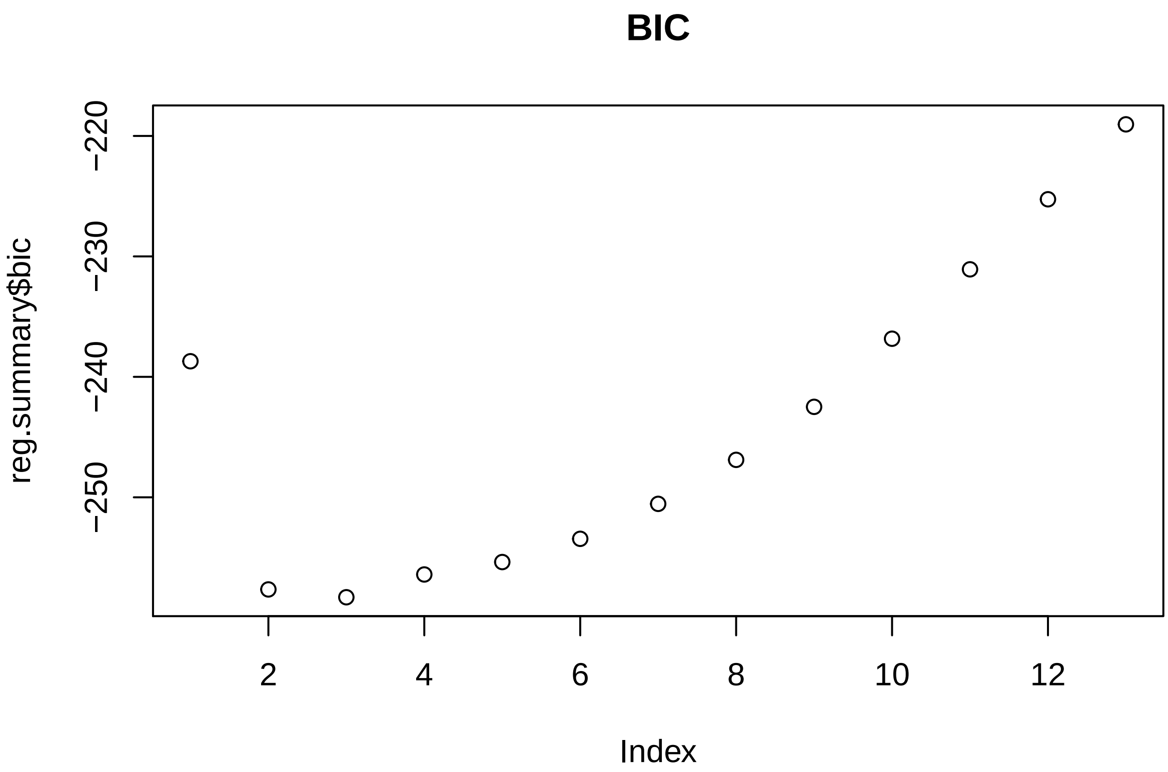
```r
# Regression: Best Subset Method
regfit.full <- regsubsets(crim ~ . , data = Boston, nvmax = 100)
reg.summary <- summary(regfit.full)

names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

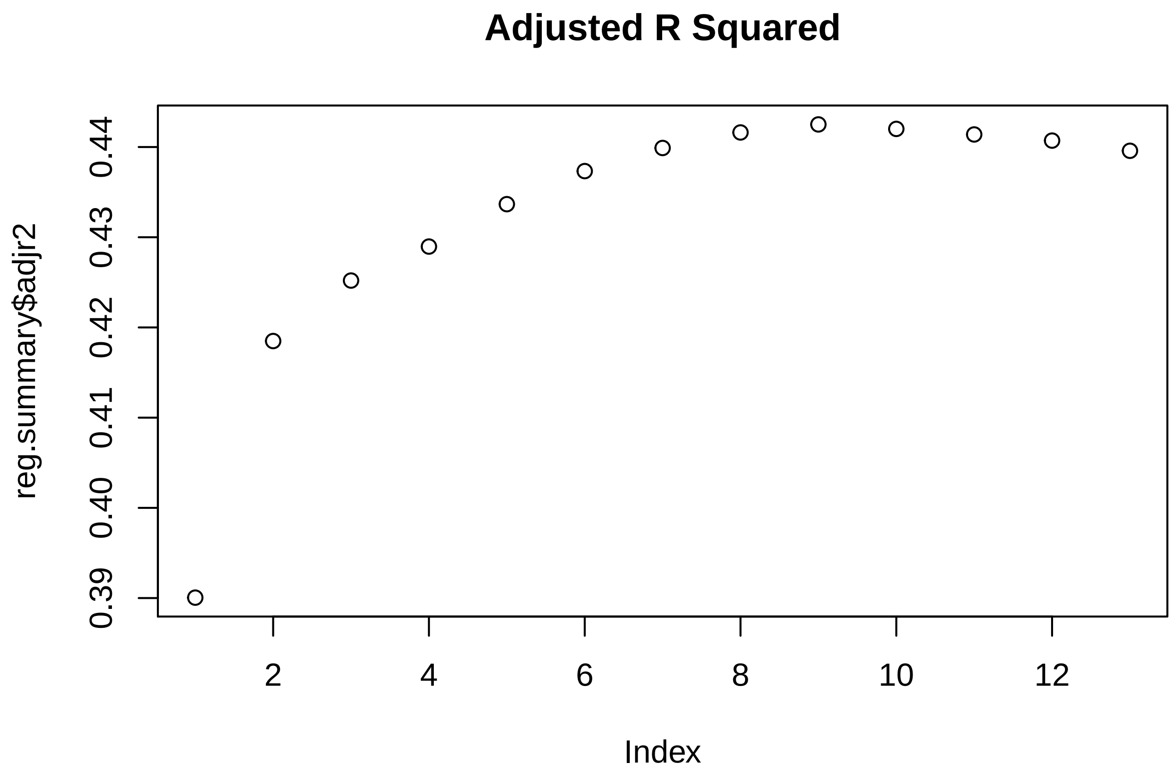
```r
plot(reg.summary$rss, main = 'Residual Sum of Squares')
```



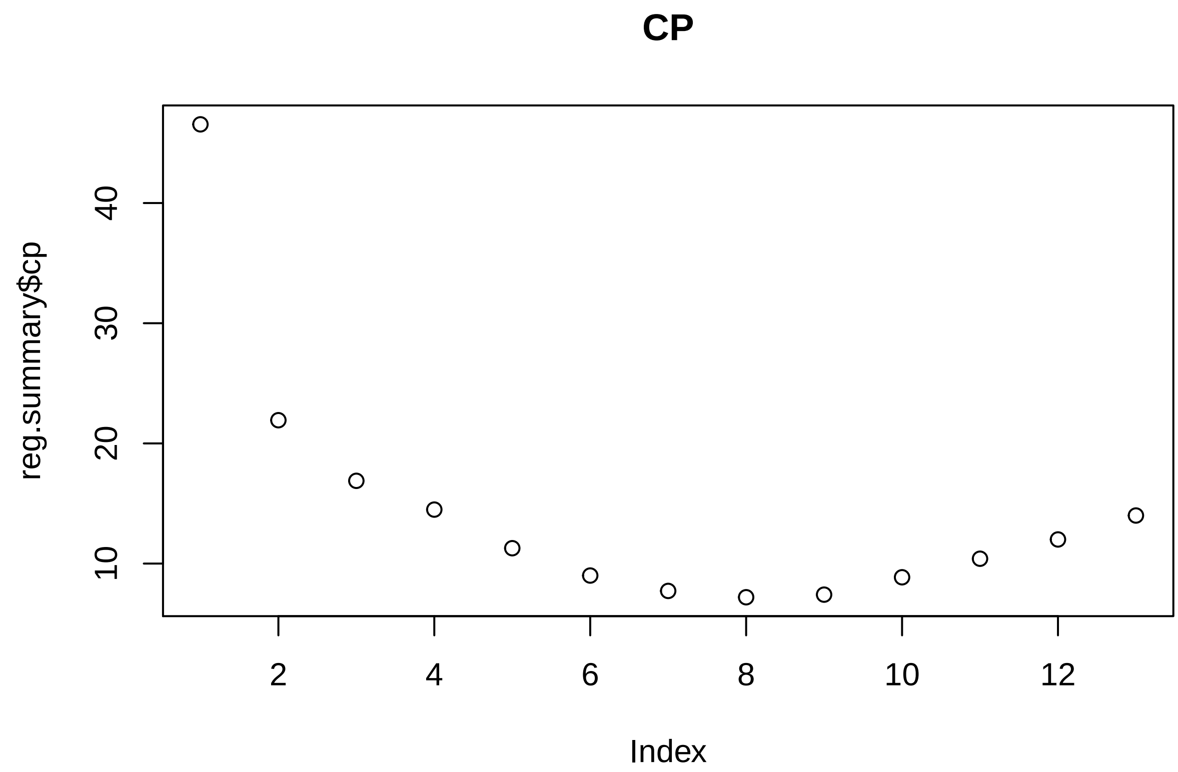

```
plot(reg.summary$bic, main = 'BIC')
```




```r
plot(reg.summary$adjr2, main = 'Adjusted R Squared')
```




```r
plot(reg.summary$cp, main = 'CP')
```




```r
coef(regfit.full, 7)
```

```
##   (Intercept)            zn           nox           dis           rad
##  22.711289450   0.044886656 -12.185035028  -1.017202266   0.541197849
##       ptratio         black          medv
##  -0.331185681  -0.008097571  -0.228833182
```

```r
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ ., data = Boston, nvmax = 100)
## 13 Variables  (and intercept)
##           Forced in Forced out
## zn            FALSE      FALSE
## indus         FALSE      FALSE
## chas          FALSE      FALSE
## nox           FALSE      FALSE
## rm            FALSE      FALSE
## age           FALSE      FALSE
## dis           FALSE      FALSE
## rad           FALSE      FALSE
## tax           FALSE      FALSE
## ptratio       FALSE      FALSE
## black         FALSE      FALSE
```

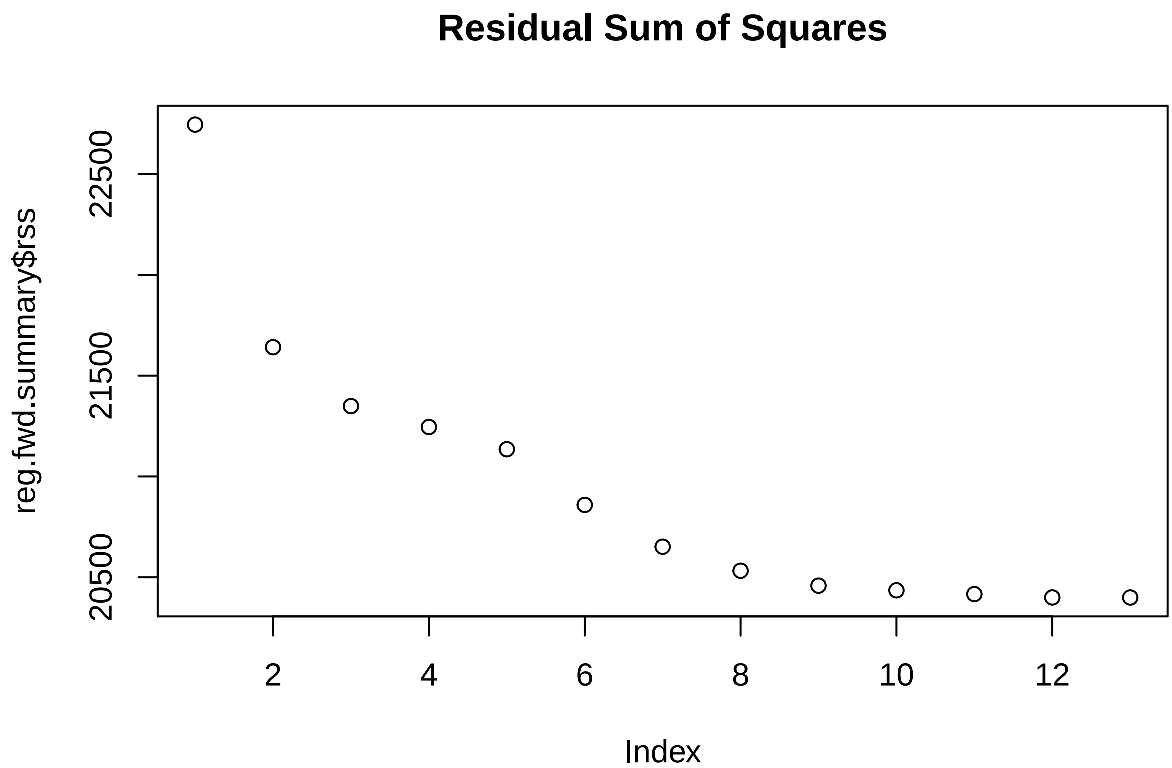
```
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: exhaustive
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " "
## 2  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   "*"   " "
## 3  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " "
## 4  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     " "   " "   "*"
## 5  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     "*"   " "   "*"
## 6  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   " "   "*"
## 7  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   " "   "*"
## 8  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
## 9  ( 1 )  "*" "*"   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
## 10  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" " " "*"     "*"   "*"   "*"
## 11  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 12  ( 1 ) "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
```

Using the Best Subset method the Adjusted r^2 is best at 9, BIC at 3, RSS at 13, and Cp at 8. Looking at the graphs, 7 is the best tradeoff of the 4 metrics. This shows zn, nox, dis, rad, ptratio, black, and medv as significant factors

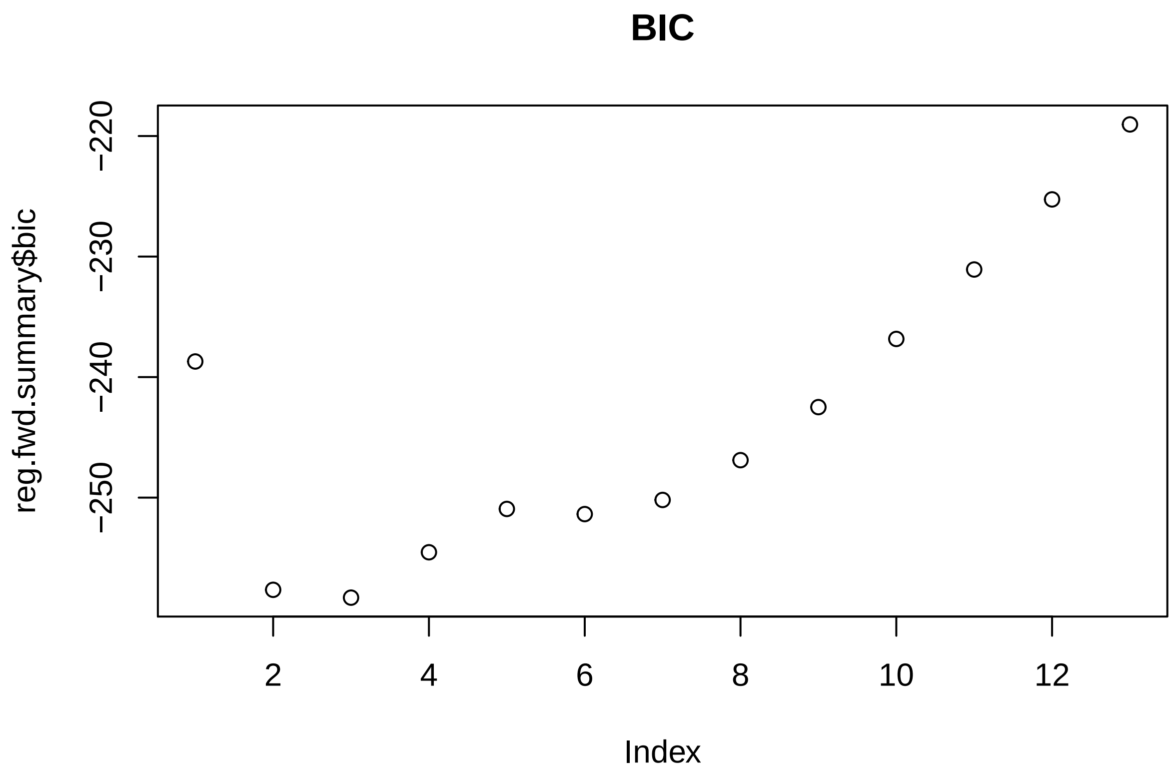
```r
#Forwards Stepwise Selection
regfit.fwd <- regsubsets(crim ~ . , data = Boston, nvmax = 100, method = 'forward')
reg.fwd.summary <- summary(regfit.fwd)

names(reg.fwd.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

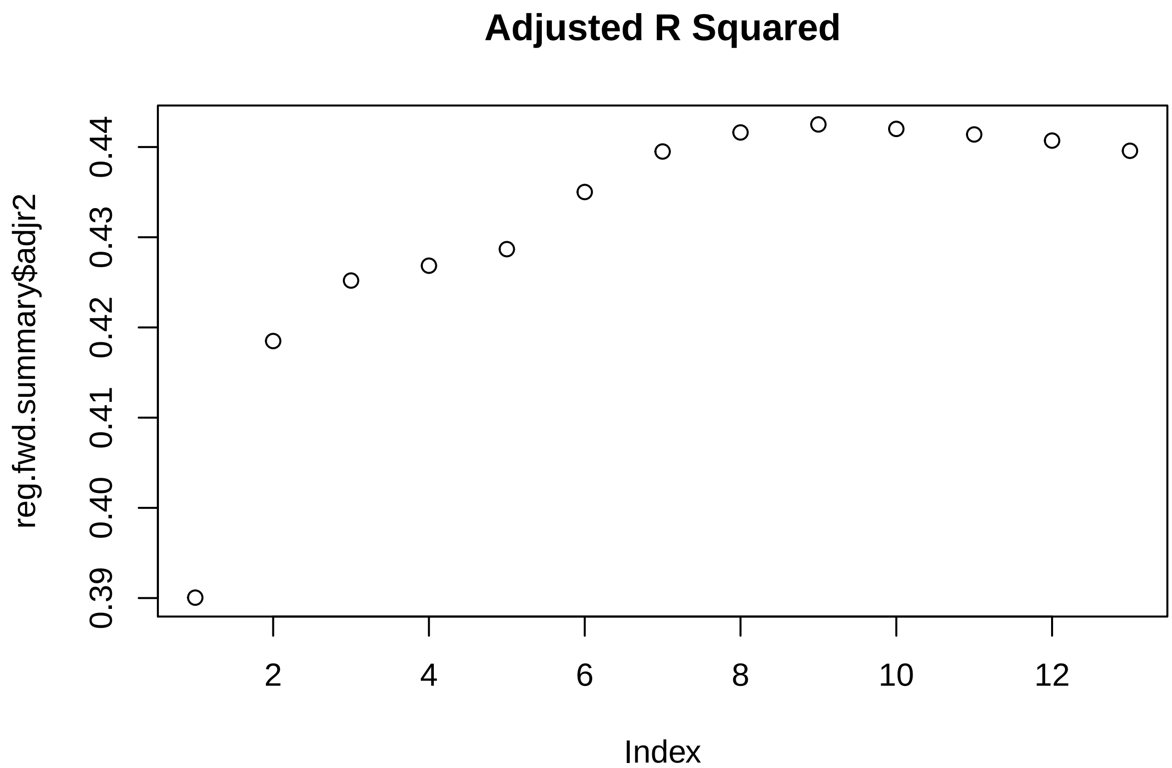
```r
plot(reg.fwd.summary$rss, main = 'Residual Sum of Squares')
```



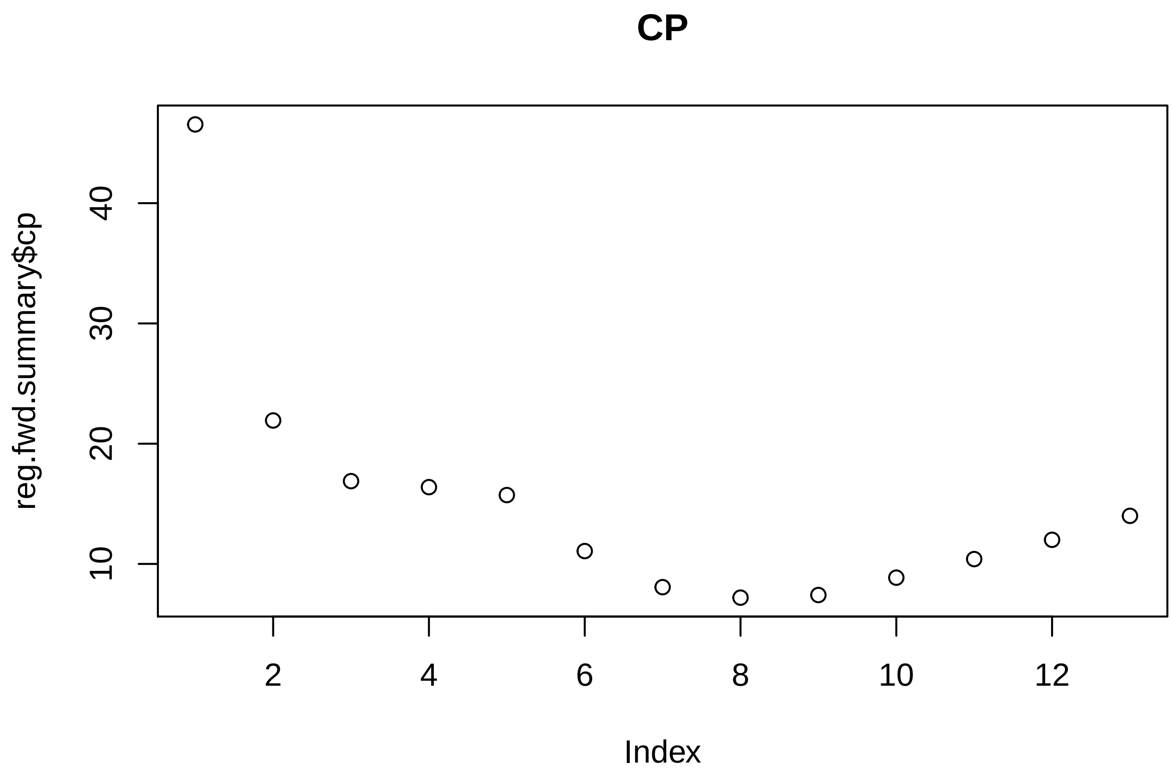

```r
plot(reg.fwd.summary$bic, main = 'BIC')
```




```r
plot(reg.fwd.summary$adjr2, main = 'Adjusted R Squared')
```




```r
plot(reg.fwd.summary$cp, main = 'CP')
```




```r
coef(regfit.fwd, 7)
```

```
##   (Intercept)            zn           nox           dis           rad
##  11.926501632   0.051640782 -10.047986363  -0.888085025   0.493381375
##         black         lstat          medv
##  -0.008481075   0.118503848  -0.139435501
```

```r
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ ., data = Boston, nvmax = 100, method = "forward")
## 13 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
```

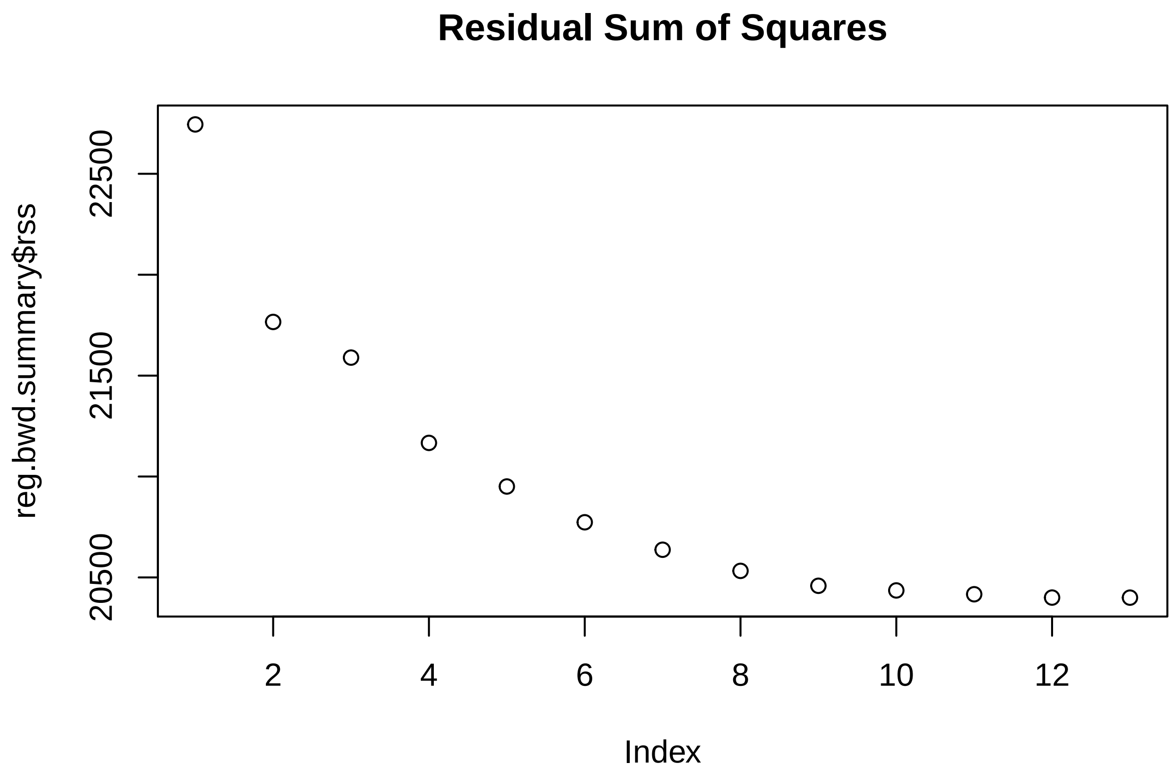
```
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: forward
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " " 
## 2  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   "*"   " " 
## 3  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " " 
## 4  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   "*" 
## 5  ( 1 )  "*" " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   "*" 
## 6  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     "*"   "*"   "*" 
## 7  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   "*"   "*" 
## 8  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*" 
## 9  ( 1 )  "*" "*"   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*" 
## 10  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" " " "*"     "*"   "*"   "*" 
## 11  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*" 
## 12  ( 1 ) "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*" 
## 13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*" 
```

The forwards model has the best RSS value at 13, BIC at 3, Adjusted R-squared at 9, and CP at 8. This matches the full model with the best model at 7 factors, zn, nox, dis, lsat, rad, black, and medv

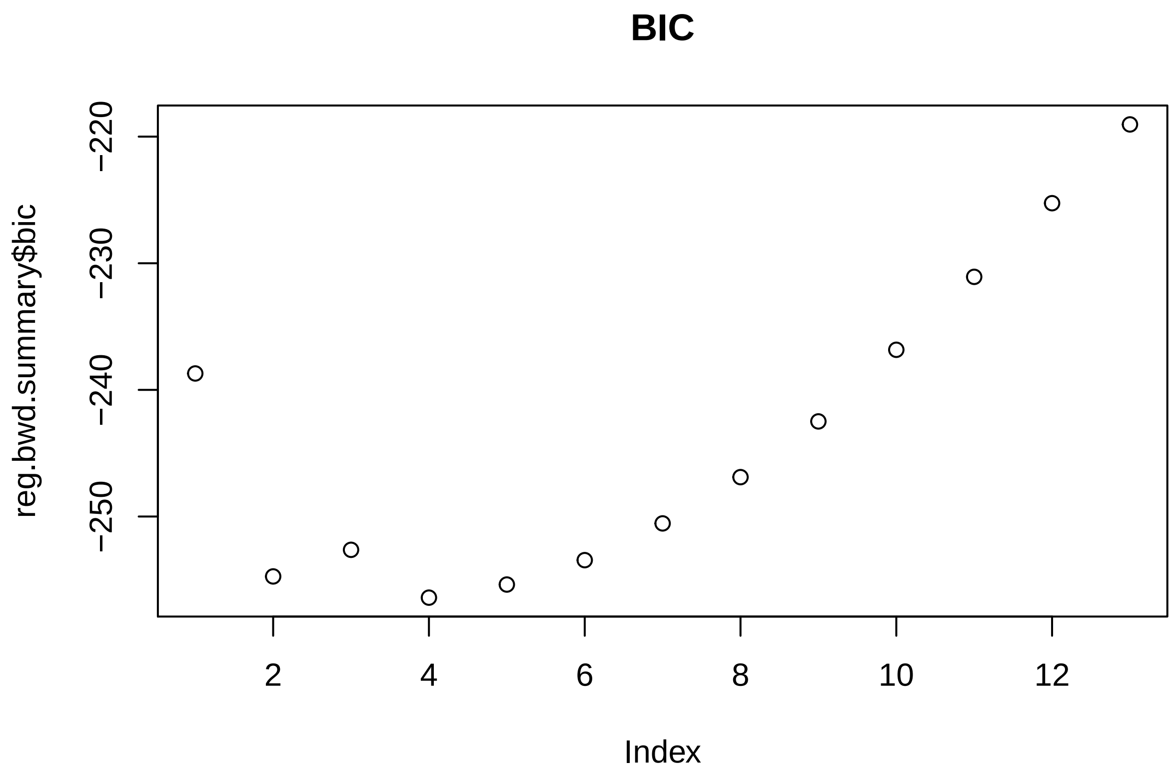
```
#Backwards Stepwise Selection
regfit.bwd <- regsubsets(crim ~ . , data = Boston, nvmax = 100, method = 'backward')
reg.bwd.summary <- summary(regfit.bwd)

names(reg.bwd.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

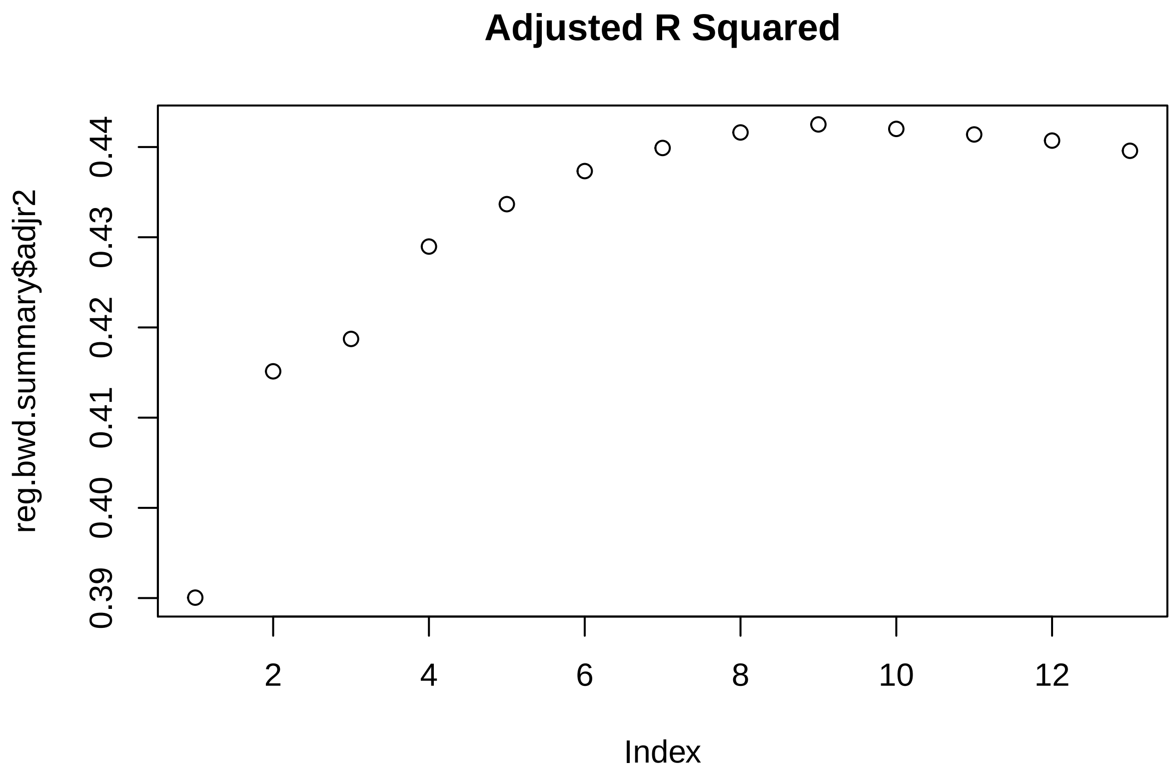
```
plot(reg.bwd.summary$rss, main = 'Residual Sum of Squares')
```



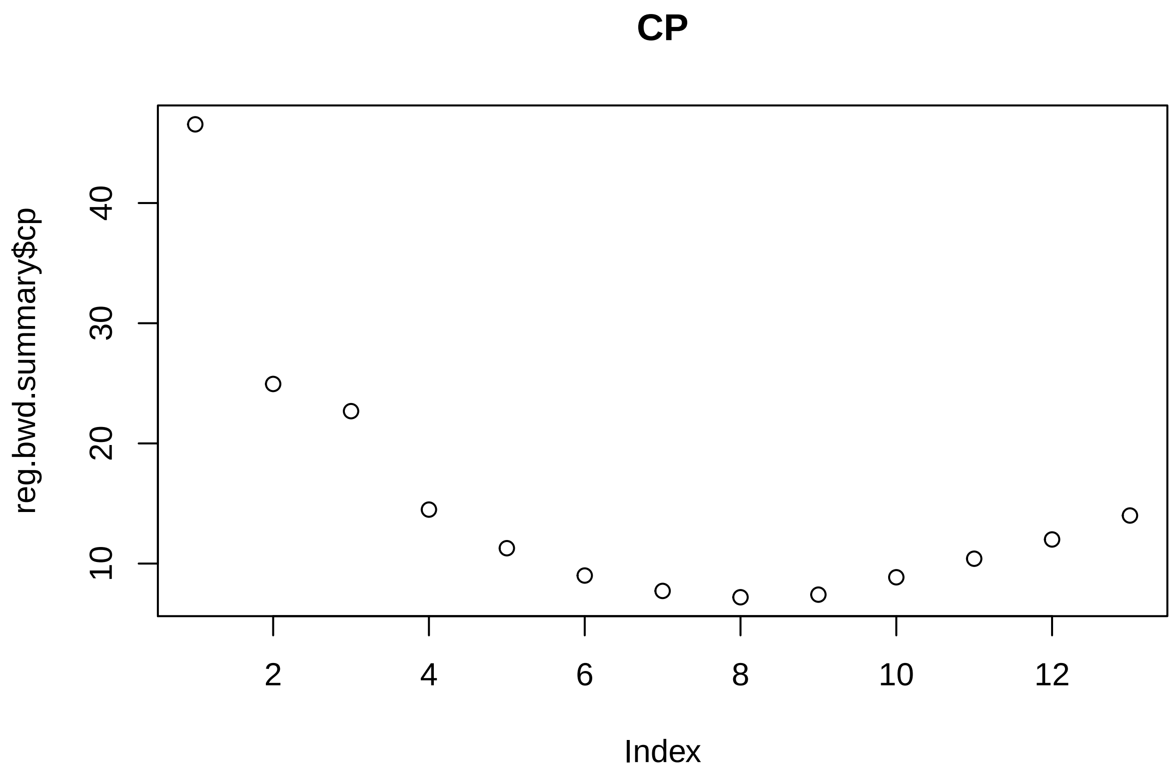

```
plot(reg.bwd.summary$bic, main = 'BIC')
```




```
plot(reg.bwd.summary$adjr2, main = 'Adjusted R Squared')
```




```
plot(reg.bwd.summary$cp, main = 'CP')
```




```r
coef(regfit.bwd, 7)
```

```
##   (Intercept)            zn           nox           dis           rad
##  22.711289450   0.044886656 -12.185035028  -1.017202266   0.541197849
##       ptratio         black          medv
##  -0.331185681  -0.008097571  -0.228833182
```

```r
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ ., data = Boston, nvmax = 100, method = "backward")
## 13 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
```

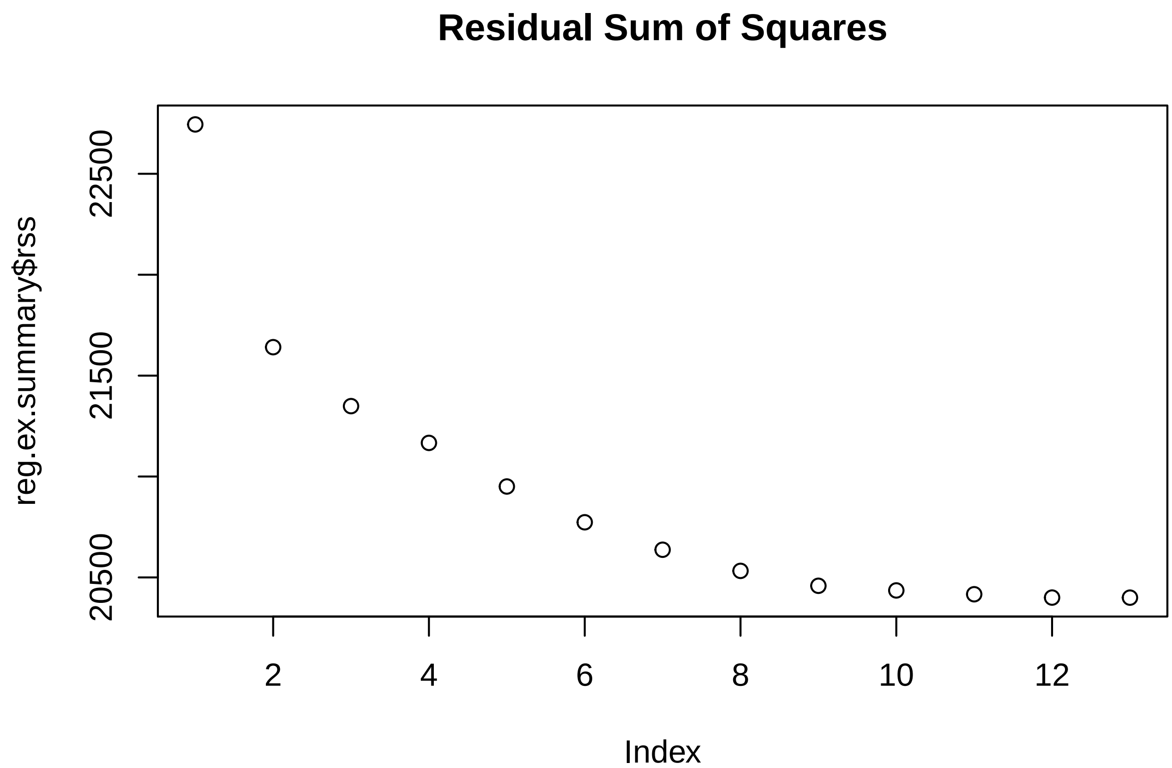
```
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: backward
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " "
## 2  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   "*"
## 3  ( 1 )  " " " "   " "  " " " " " " "*" "*" " " " "     " "   " "   "*"
## 4  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     " "   " "   "*"
## 5  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     "*"   " "   "*"
## 6  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   " "   "*"
## 7  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   " "   "*"
## 8  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
## 9  ( 1 )  "*" "*"   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
## 10  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" " " "*"     "*"   "*"   "*"
## 11  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 12  ( 1 ) "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
```

The backwards model has the best RSS value at 13, BIC at 4, Adjusted R-squared at 9, and CP at 8. This matches the full model with the best model at 7 factors, zn, nox, dis, rad, ptratio, black, and medv

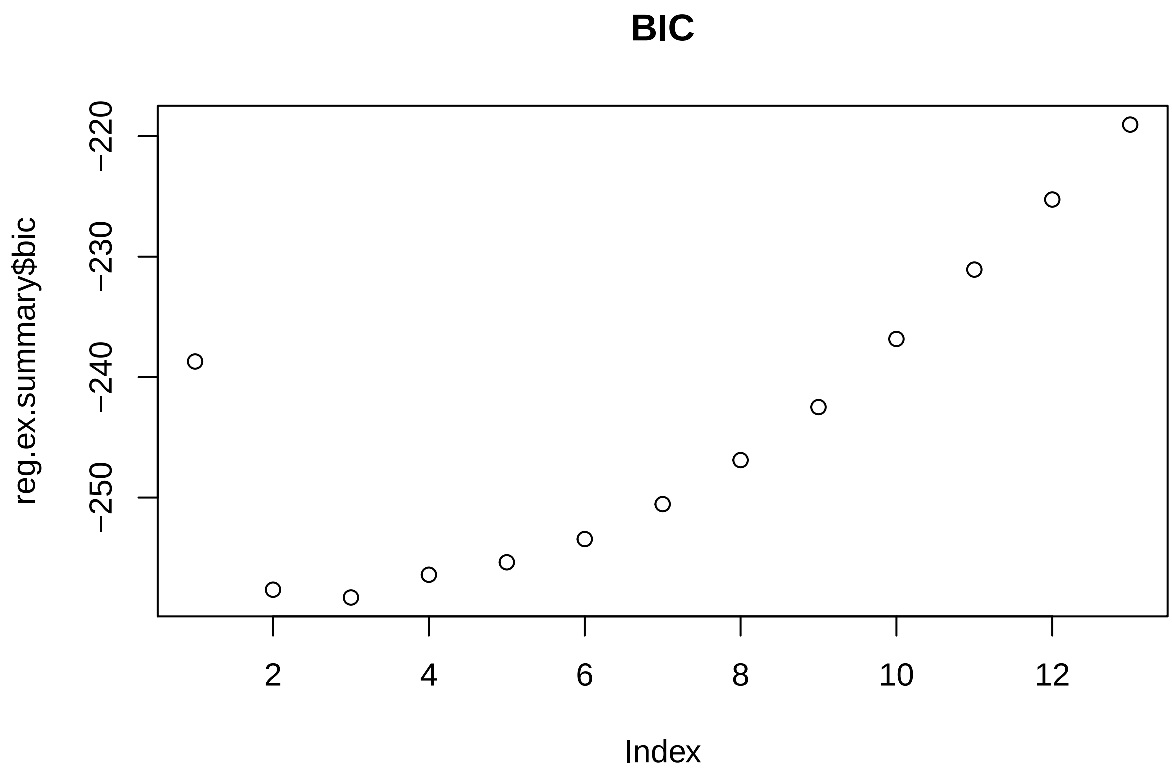
```r
#exhaustive Stepwise Selection
regfit.ex <- regsubsets(crim ~ . , data = Boston, nvmax = 100, method = 'exhaustive')
reg.ex.summary <- summary(regfit.ex)

names(reg.ex.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

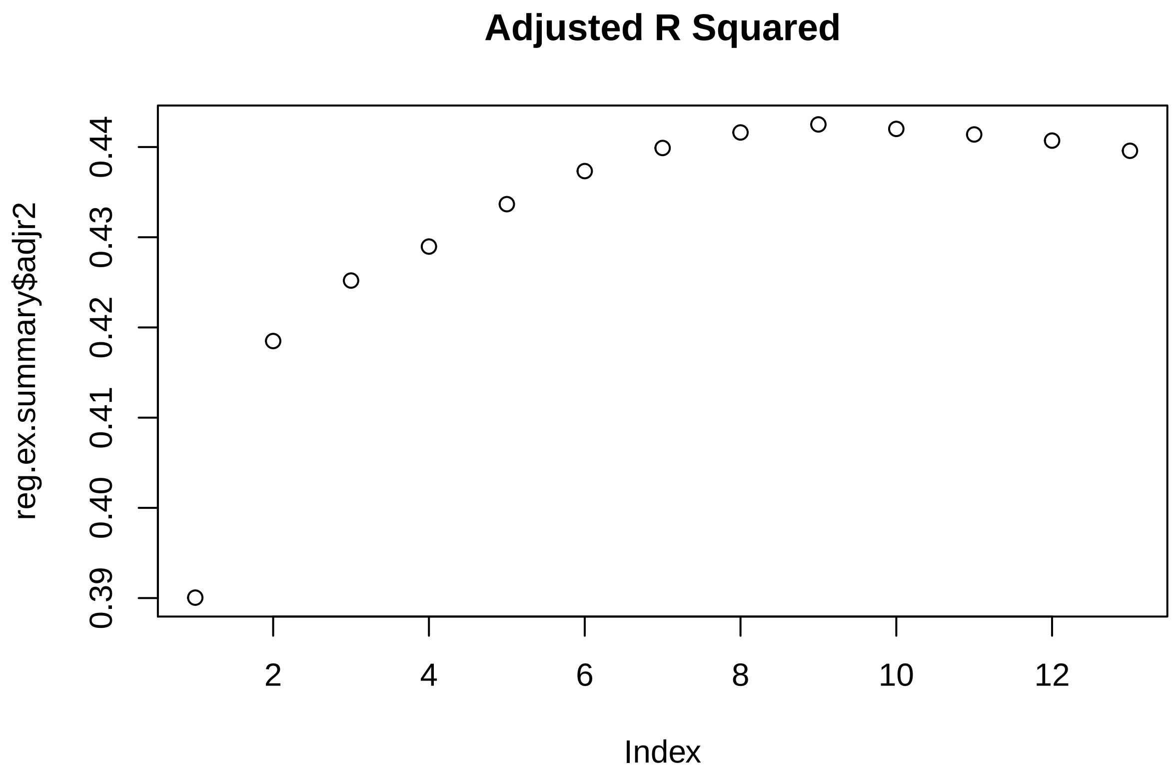
```r
plot(reg.ex.summary$rss, main = 'Residual Sum of Squares')
```



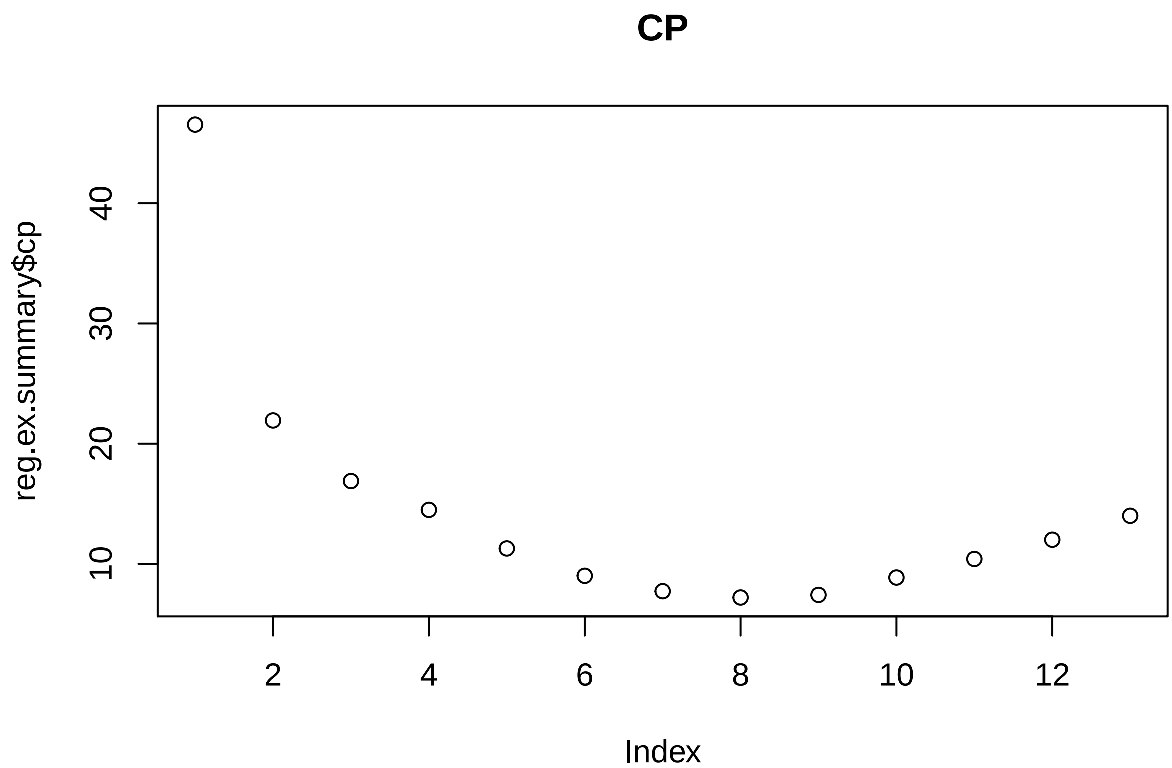

```r
plot(reg.ex.summary$bic, main = 'BIC')
```




```r
plot(reg.ex.summary$adjr2, main = 'Adjusted R Squared')
```




```r
plot(reg.ex.summary$cp, main = 'CP')
```




```r
coef(regfit.ex, 7)
```

```
##   (Intercept)            zn           nox           dis           rad
##  22.711289450   0.044886656 -12.185035028  -1.017202266   0.541197849
##       ptratio         black          medv
##  -0.331185681  -0.008097571  -0.228833182
```

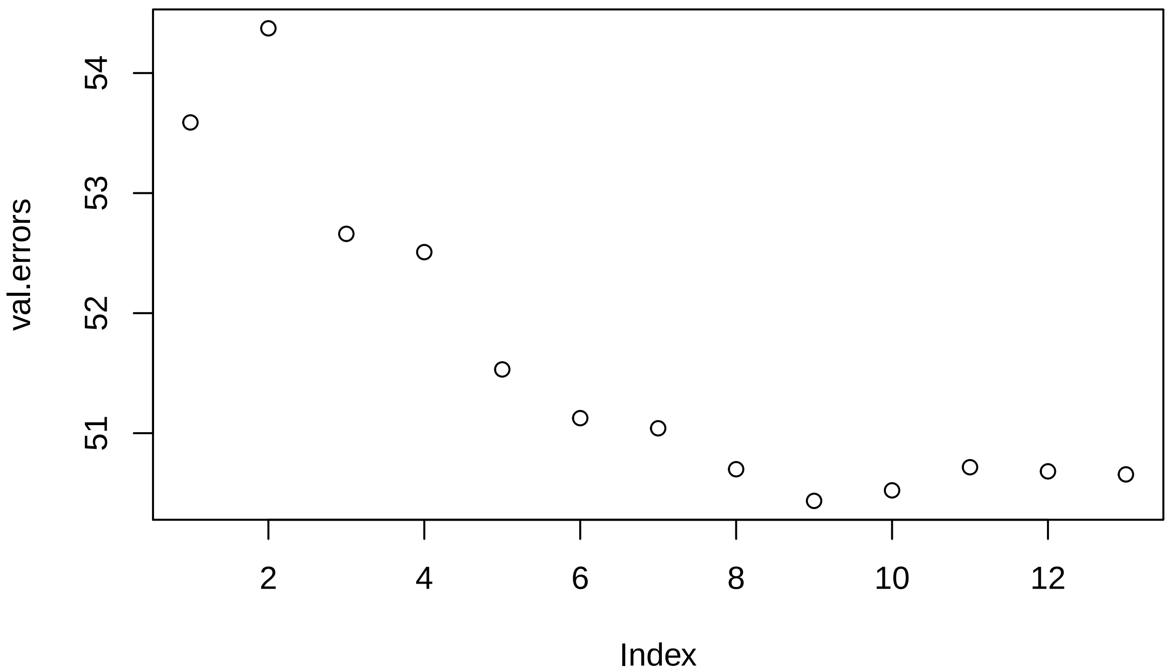
```r
summary(regfit.ex)
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ ., data = Boston, nvmax = 100, method = "exhaustive")
## 13 Variables  (and intercept)
##           Forced in Forced out
## zn            FALSE      FALSE
## indus         FALSE      FALSE
## chas          FALSE      FALSE
## nox           FALSE      FALSE
## rm            FALSE      FALSE
## age           FALSE      FALSE
## dis           FALSE      FALSE
## rad           FALSE      FALSE
## tax           FALSE      FALSE
## ptratio       FALSE      FALSE
## black         FALSE      FALSE
```

```
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 13
## Selection Algorithm: exhaustive
##           zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " "
## 2  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     " "   "*"   " "
## 3  ( 1 )  " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " "
## 4  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     " "   " "   "*"
## 5  ( 1 )  "*" " "   " "  " " " " " " "*" "*" " " " "     "*"   " "   "*"
## 6  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   " "   "*"
## 7  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   " "   "*"
## 8  ( 1 )  "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
## 9  ( 1 )  "*" "*"   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
## 10  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" " " "*"     "*"   "*"   "*"
## 11  ( 1 ) "*" "*"   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 12  ( 1 ) "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"   "*"
## 13  ( 1 ) "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"   "*"
```

The exhaustive model has the best RSS value at 13, BIC at 3, Adjusted R-squared at 9, and CP at 8. This matches the full model with the best model at 7 factors, zn, nox, dis, rad, ptratio, black, and medv

```r
set.seed(1)
train <- sample(c(TRUE, FALSE), nrow(Boston), replace = TRUE)
test <- (!train)

regfit.best = regsubsets(crim ~ . , data = Boston[train,], nvmax = 13)

test.mat <- model.matrix(crim ~., data = Boston[test,])

val.errors <- rep(NA,13)

for(i in 1:13){
  coefi <- coef(regfit.best, id=i)
  pred <- test.mat[,names(coefi)] %*% coefi
  val.errors[i] <- mean((Boston$crim[test]-pred)^2)
}
plot(val.errors)
```




```r
coef(regfit.best,which.min(val.errors))
```

```
## (Intercept)          zn       indus         nox         dis         rad
## 16.49784501  0.04428501 -0.11356470 -6.80041892 -0.87067024  0.48133294
##     ptratio       black       lstat        medv
## -0.17759119 -0.01438142  0.12943566 -0.13215744
```

Using cross-validation it looks as though the best model is at 9, with similar values at 7, 8, and 10 with coefficients of zn, indus, nox, dis, rad, ptratio, black, lstat, and medv. I reccomend choosing the 8 (zn, indus, nox, dis, rad, ptratio, lstat, and medv) variable model as a happy medium between the cross-validation 9, and forward & backwards model of 7.