Regression Techniques

Subhrajyoty Roy (MB-1911)

August 8, 2019

Introduction

Every statistical model is based on some assumptions. When applying this model to fit real world data, it is essential to check whether those assumptions are actually valid for the specific dataset in concern. For Regression model, the method to verify the assumptions is called **Residual Diagnostics**.

The another practical problem that we are often faced with is that there might be many predictor variables which affect the response variable which we are trying to model. However, as we add more and more variables, the model kind of overfits to the specific dataset that we are working with (that is the error for the training dataset becomes smaller) and does not generalize well. Also, the model complexity increases and it becomes cubersome (sometimes impossible) to make a prediction using a model that uses many predictors. In that case, it is essential to figure out a subset of predictors which can explain the change in response variable better, while being reasonably smaller in size in order to make prediction possible. The method to create such subsets of variables is called **Model Selection** or **Variable Selection**.

Dataset

Here, we are working with *CarSeats* dataset from *ISLR* package. This is a data set containing sales of child car seats at 400 different stores. It contains the following columns.

1. **Sales:** Unit sales (in thousands) at each location
2. **CompPrice:** Price charged by competitor at each location
3. **Income:** Community income level (in thousands of dollars)
4. **Advertising:** Local advertising budget for company at each location (in thousands of dollars)
5. **Population:** Population size in region (in thousands)
6. **Price:** Price company charges for car seats at each site
7. **ShelveLoc:** A factor with levels Bad, Good and Medium indicating the quality of the shelving location for the car seats at each site
8. **Age:** Average age of the local population
9. **Education:** Education level at each location
10. **Urban:** A factor with levels No and Yes to indicate whether the store is in an urban or rural location
11. **US:** A factor with levels No and Yes to indicate whether the store is in the US or not

Let us first take a look at the data.

```
carseats = ISLR::Carseats
knitr::kable(head(carseats))
```

Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
9.50	138	73	11	276	120	Bad	42	17	Yes	Yes

Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
11.22	111	48	16	260	83	Good	65	10	Yes	Yes
10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
7.40	117	100	4	466	97	Medium	55	14	Yes	Yes
4.15	141	64	3	340	128	Bad	38	13	Yes	No
10.81	124	113	13	501	72	Bad	78	16	No	Yes

Since this data contains some factor variables, we remove those variables.

```
carseats = carseats[, -c(7,10,11)]
knitr::kable(head(carseats))
```

Sales	CompPrice	Income	Advertising	Population	Price	Age	Education
9.50	138	73	11	276	120	42	17
11.22	111	48	16	260	83	65	10
10.06	113	35	10	269	80	59	12
7.40	117	100	4	466	97	55	14
4.15	141	64	3	340	128	38	13
10.81	124	113	13	501	72	78	16

Now, we have a dataset with 400 observations and 8 variables. We choose the *Sales* as our response variable which we are trying to predict from the other predictor variables.

```
summary(carseats)
```

```
     Sales          CompPrice       Income        Advertising    
 Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000  
 1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000  
 Median : 7.490   Median :125   Median : 69.00   Median : 5.000  
 Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635  
 3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000  
 Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000  
   Population        Price            Age          Education   
 Min.   : 10.0   Min.   : 24.0   Min.   :25.00   Min.   :10.0  
 1st Qu.:139.0   1st Qu.:100.0   1st Qu.:39.75   1st Qu.:12.0  
 Median :272.0   Median :117.0   Median :54.50   Median :14.0  
 Mean   :264.8   Mean   :115.8   Mean   :53.32   Mean   :13.9  
 3rd Qu.:398.5   3rd Qu.:131.0   3rd Qu.:66.00   3rd Qu.:16.0  
 Max.   :509.0   Max.   :191.0   Max.   :80.00   Max.   :18.0  
```

We also have a missing entry in the observations, which is represented by the *Sales* column taking value 0. Hence, we need to remove that observations.

```
carseats = subset(carseats, Sales > 0)
nrow(carseats)
```

```
[1] 399
```

Now, we have 399 observations in our dataset.

Residual Diagnostics

To perform residual diagnostics, firstly, we have to consider a linear regression model with some of the predictors. From economic theory, it is a known fact that demand mostly depends on the price of the commodity, the competitive prices and the average income of the market group. Therefore, we only choose variables *Sales, CompPrice, Income, Price* to work with.

```
carseats2 = carseats[, c("Sales","CompPrice","Income","Price")]
knitr::kable(head(carseats2))
```

Sales	CompPrice	Income	Price
9.50	138	73	120
11.22	111	48	83
10.06	113	35	80
7.40	117	100	97
4.15	141	64	128
10.81	124	113	72

Next, we fit a linear model with *Sales* as response variable and the rest as predictors.

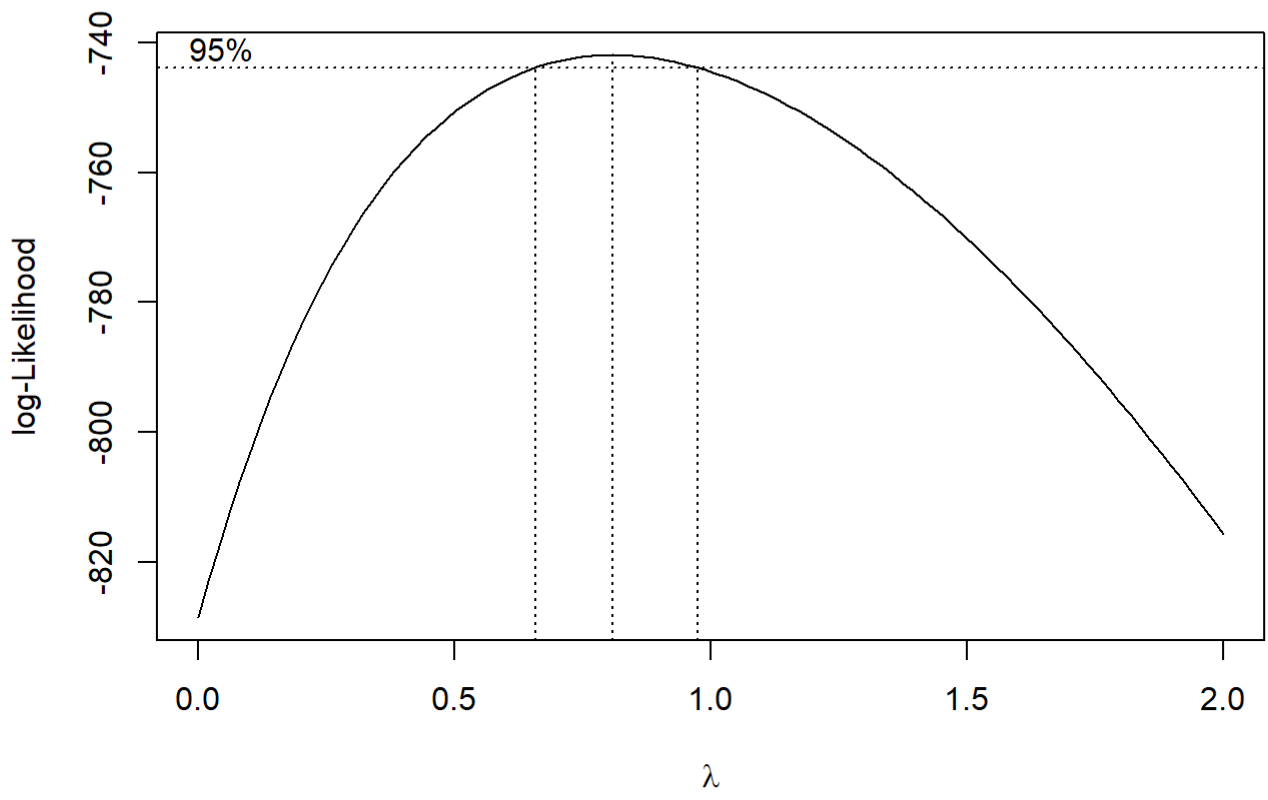
```
model <- lm(Sales ~ Price + CompPrice + Income, data = carseats2)
summary(model)
```

```


Call:
lm(formula = Sales ~ Price + CompPrice + Income, data = carseats2)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.1365 -1.5099 -0.2152  1.4739  6.1060 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.938123   0.981591   5.031 7.43e-07 ***
Price       -0.086375   0.005881 -14.686  < 2e-16 ***
CompPrice    0.092323   0.009010  10.247  < 2e-16 ***
Income       0.014963   0.004017   3.725 0.000223 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.231 on 395 degrees of freedom
Multiple R-squared:  0.3708,    Adjusted R-squared:  0.366 
F-statistic: 77.59 on 3 and 395 DF,  p-value: < 2.2e-16
```

We find that multiple R squared is coming out to be 0.3708, suggesting a poor fit of the linear model. Note that, all variables seems to have a significant contribution in the linear model at significance level $\alpha = 0.05$.

Firstly, we use the box-cox transformation to find out whether a suitable power transformation of the response variable can make it linearly related with the predictor variables.

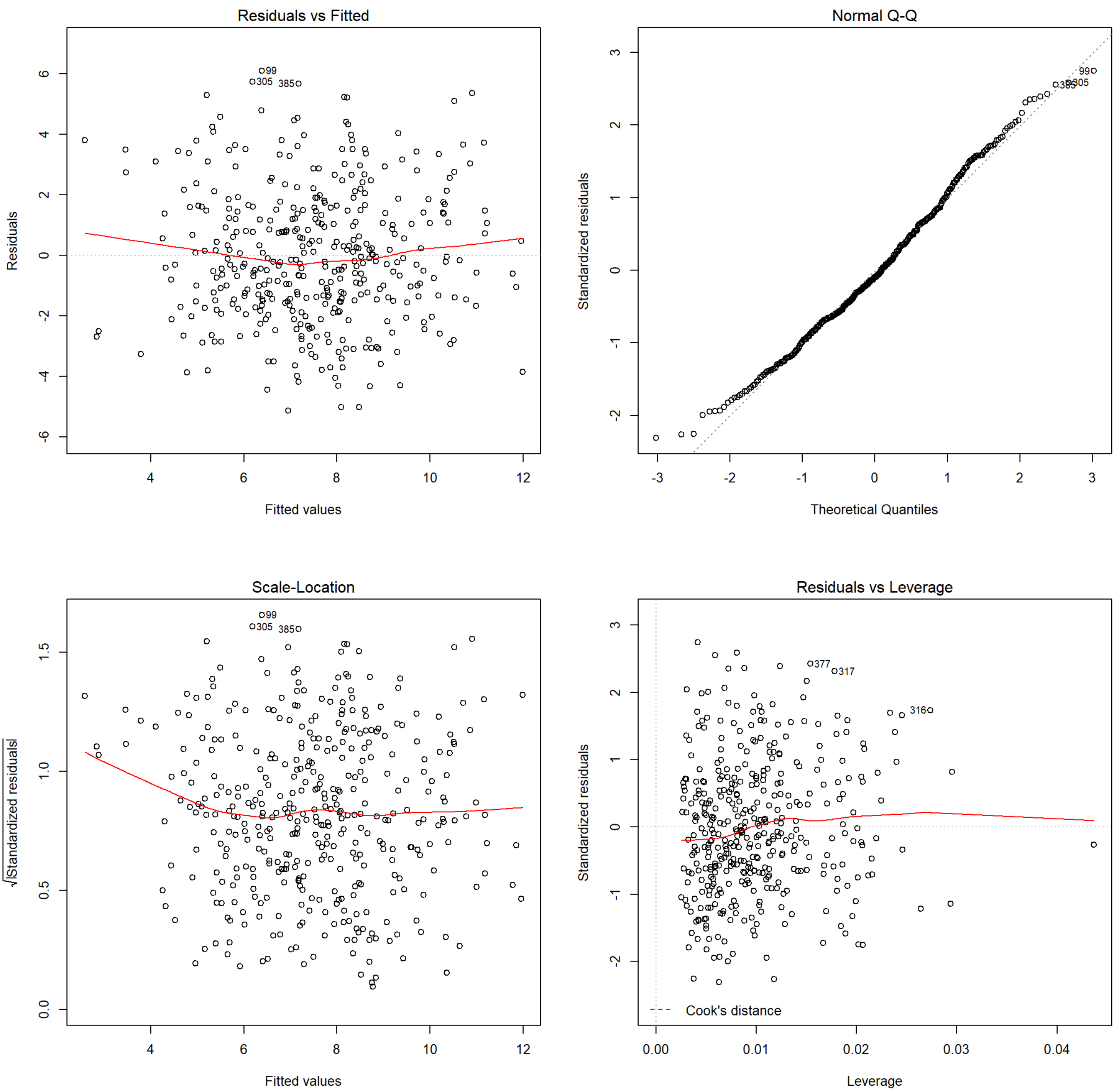
```
library(MASS)
boxcox(model, lambda = seq(0,2, 1/10))
```



Note that, the appropriate power transformation which yields the most log-Liikelihood is about 0.8. However, since the point $\lambda = 1$ is within the 95% confidence interval range for actual λ, it seems that power transformation would not greatly benifit us than a multiple linear regression model with no transformation made on response variable.

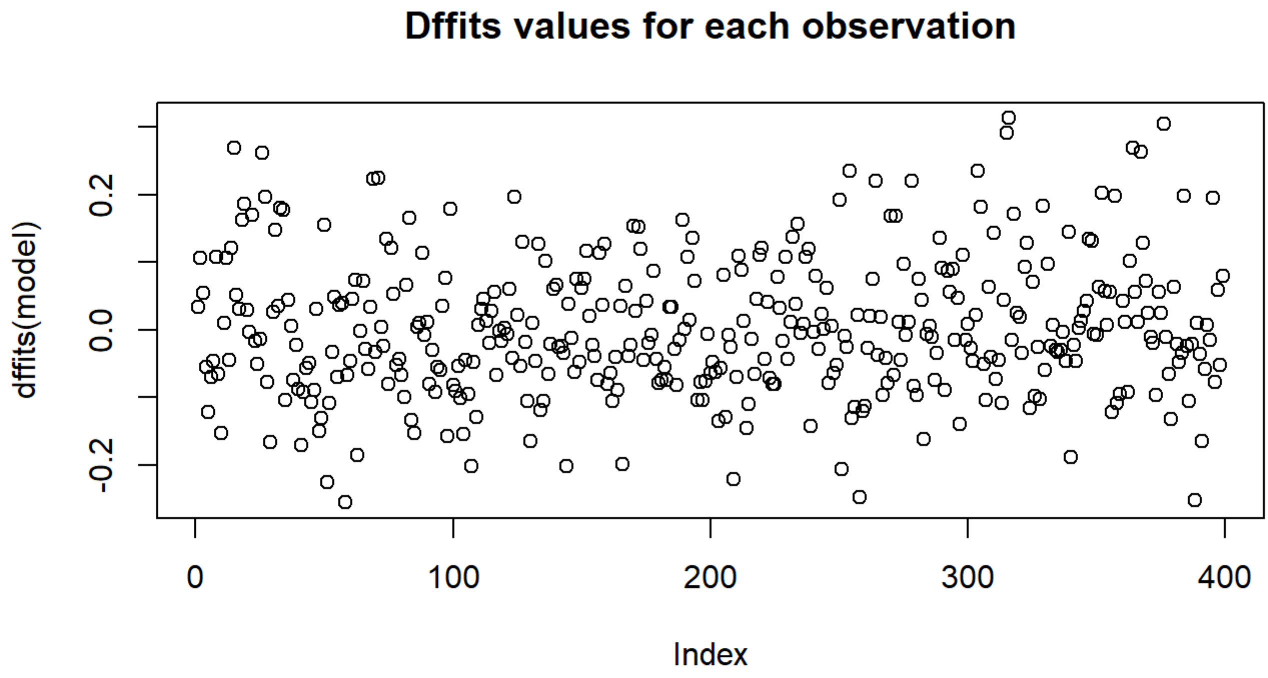
To check whether the assumptions of linear regression model is valid, we consider the following 4 plots.

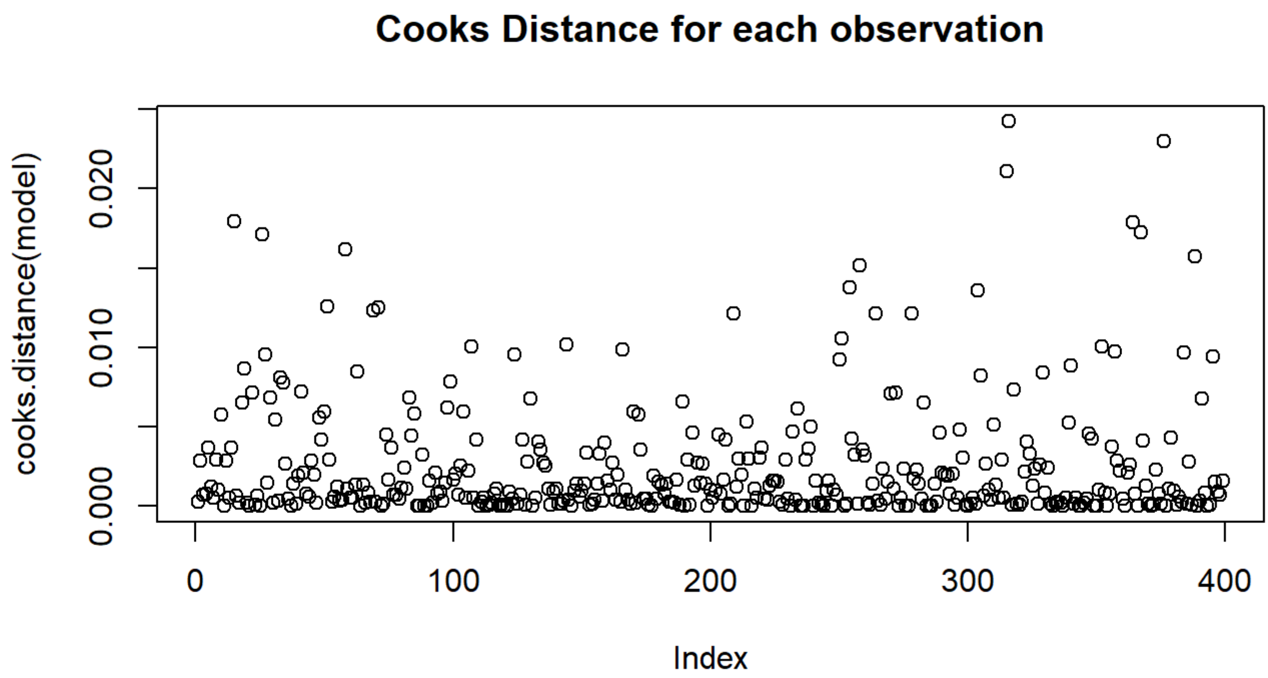
```
par(mfrow = c(2,2))
plot(model)
```



Note that, the residuals vs fitted plot does not show any evident pattern, as expected. The Normal Q-Q plot shows that the sample quantiles of standardized residuals do agree with theoretical standard normal quantiles, however, there is some minor deviations in both tails. The plot of Cook's distance does show one high leverage point which might be influential.

```
par(mfrow = c(2,1))
plot(dffits(model), main = "Dffits values for each observation")
plot(cooks.distance(model), main = "Cooks Distance for each observation")
```





We find that there are some observations for which Cooks distance is relatively large, however, there is no obvious influential points.

Model Selection

We shall use the R package *leaps* for performing Model selection in *carseats* data.

```
library(leaps)
```

```
Warning: package 'leaps' was built under R version 3.5.3
```

We first consider all possible models of the available 7 variables (i.e. 127 models) and for each number of predictors, we output the **best** subset with that many predictors in our model, where **best** is determined by maximum R^2 or minimum residual sum of squares. For example, if we conisder those models with only two predictors, then we fit all $\binom{7}{2} = 21$ models and report only the one model which has maximum R^2. This gives the *best* linear model with 2 predictor variables. Finally, the models containing different number of variables are compared against each other using **Mallow's C_p criterion** or **Adjusted R^2**.

Firstly, we use C_p criterion to figure out the best model.

```
fits = leaps(x = carseats[,2:8], y = carseats[,1], names = names(carseats)[2:8], nbest = 1, m
ethod = "Cp")

fits
```

```
$which
  CompPrice Income Advertising Population Price   Age Education
1     FALSE  FALSE       FALSE      FALSE  TRUE FALSE     FALSE
2      TRUE  FALSE       FALSE      FALSE  TRUE FALSE     FALSE
3      TRUE  FALSE        TRUE      FALSE  TRUE FALSE     FALSE
4      TRUE  FALSE        TRUE      FALSE  TRUE  TRUE     FALSE
5      TRUE   TRUE        TRUE      FALSE  TRUE  TRUE     FALSE
6      TRUE   TRUE        TRUE      FALSE  TRUE  TRUE      TRUE
7      TRUE   TRUE        TRUE       TRUE  TRUE  TRUE      TRUE

$label
[1] "(Intercept)" "CompPrice"   "Income"      "Advertising" "Population"
[6] "Price"       "Age"         "Education"

$size
[1] 2 3 4 5 6 7 8

$Cp
[1] 285.200044 152.810271  71.429683  17.364066   5.167873   6.034701
[7]   8.000000
```

The best model according to C_p criterion is the model for which C_p value is closest to the number of predictors in the model. From the results above, we find that the best reduced submodel is the one which only 6 predictor variable (including Intercept, $p = 6$ and $C_p = 5.167873$), and the corresponding model contains *CompPrice, Income, Advertising, Price* and *Age* as predictors as well as an intercept component.

```
submodel1 = lm(Sales ~ CompPrice + Income + Advertising + Price + Age, data = carseats)
```

We perform similar treatment with **Adjusted R^2**, where the best model is defined by maximum value of adjusted R^2.

```
fits = leaps(x = carseats[,2:8], y = carseats[,1], names = names(carseats)[2:8], nbest = 1, m
ethod = "adjr2")

fits
```

```
$which
  CompPrice Income Advertising Population Price   Age Education
1     FALSE  FALSE       FALSE      FALSE  TRUE FALSE     FALSE
2      TRUE  FALSE       FALSE      FALSE  TRUE FALSE     FALSE
3      TRUE  FALSE        TRUE      FALSE  TRUE FALSE     FALSE
4      TRUE  FALSE        TRUE      FALSE  TRUE  TRUE     FALSE
5      TRUE   TRUE        TRUE      FALSE  TRUE  TRUE     FALSE
6      TRUE   TRUE        TRUE      FALSE  TRUE  TRUE      TRUE
7      TRUE   TRUE        TRUE       TRUE  TRUE  TRUE      TRUE

$label
[1] "(Intercept)" "CompPrice"   "Income"      "Advertising" "Population"
[6] "Price"       "Age"         "Education"  

$size
[1] 2 3 4 5 6 7 8

$adjr2
[1] 0.1862737 0.3453959 0.4439923 0.5101633 0.5260728 0.5262367 0.5250671
```

In this case, the *best* model comes out to be the one with 7 predictors (including the intercept). It only leaves out the variable *Population* and use the rest of the variables to model the response variable *Sales*. We, therefore, also consider this reduced submodel.

```
submodel2 = lm(Sales ~ CompPrice + Income + Advertising + Price + Age + Education, data = car
seats)
```

Now, we use Information criterion to find out the best reduced submodel which can explain the variation in response variable *Sales* properly. We compare the two submodel using *AIC, BIC* and *PRESS* criterion.

```
print(paste("The AIC for first submodel is", extractAIC(submodel1)[2]))
```

```
[1] "The AIC for first submodel is 530.374705575129"
```

```
print(paste("The AIC for second submodel is", extractAIC(submodel2)[2]))
```

```
[1] "The AIC for second submodel is 531.220123416082"
```

We see that, the first submodel has lower AIC than second submodel.

```
print(paste("The BIC for first submodel is", extractAIC(submodel1, k = log(nrow(carseats)))[2
]))
```

```
[1] "The BIC for first submodel is 554.308474076468"
```

```
print(paste("The BIC for second submodel is", extractAIC(submodel2, k = log(nrow(carseats)))[
2]))
```

```
[1] "The BIC for second submodel is 559.142853334311"
```

We also see that, the first submodel has lower BIC than second submodel.

```
print(paste("The PRESS value for first submodel is", sum((submodel1$residuals/(1-hatvalues(submodel1)))^2) ))
```

```
[1] "The PRESS value for first submodel is 1508.93798604069"
```

```
print(paste("The PRESS value for second submodel is", sum((submodel2$residuals/(1-hatvalues(submodel2)))^2) ))
```

```
[1] "The PRESS value for second submodel is 1512.0134529052"
```

Note that, the first submodel also has lower PRESS value than second submodel. Hence, the first submodel should be the *best* reduced model, under any criterion we use.

The summary of first submodel is given as follows;

```
summary(submodel1)
```

```


Call:
lm(formula = Sales ~ CompPrice + Income + Advertising + Price +
    Age, data = carseats)

Residuals:
    Min      1Q  Median      3Q     Max
-4.9070 -1.3233 -0.1939  1.1544  4.6976

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.113720   0.946122   7.519 3.78e-13 ***
CompPrice    0.093947   0.007813  12.024  < 2e-16 ***
Income       0.013118   0.003478   3.772 0.000187 ***
Advertising  0.130697   0.014614   8.943  < 2e-16 ***
Price       -0.092626   0.005113 -18.115  < 2e-16 ***
Age         -0.045030   0.006028  -7.471 5.22e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.929 on 393 degrees of freedom
Multiple R-squared:  0.532, Adjusted R-squared:  0.5261
F-statistic: 89.36 on 5 and 393 DF,  p-value: < 2.2e-16
```

Forward and Backward Selection Method

In previous part, *leaps()* function had to check all 127 possible combinations of 7 predictor variables. If the number of predictor variables i.e. p is large, then fitting $2^p - 1$ many regression line would be tedious and computationally expensive.

In such case, one workaround is to add predictor variables sequentially to a base model or remove predictor variables sequentially from the full model. The first method is **Forward Selection Method** and the second method is **Backward Selection Method**.

We first create a Base model and a Full model. To create a base model with a single predictor variable, we choose the one which has highest magnitude of correlation with the response variable.

```r
sapply(carseats, function(x) {abs(cor(x, carseats[,1]))} )
```

```
      Sales   CompPrice      Income Advertising  Population       Price 
 1.00000000  0.07088258  0.14303466  0.26554315  0.05520510  0.43395646 
        Age   Education 
 0.22393720  0.04960767 
```

We find that, the variable *Price* has highest correlation with *Sales*. Therefore, our base model will include *Price* variable and an intercept term.

```r
base = lm(Sales ~ Price, data = carseats)
full = lm(Sales ~ ., data = carseats)
```

Firstly, we use forward selection method.

```r
step(base, scope = list( upper=full, lower= ~1 ), direction = "forward", trace=FALSE)
```

```


Call:
lm(formula = Sales ~ Price + CompPrice + Advertising + Age + 
    Income, data = carseats)

Coefficients:
(Intercept)        Price    CompPrice  Advertising          Age  
    7.11372     -0.09263      0.09395      0.13070     -0.04503  
     Income  
    0.01312  
```

We find that the *best* model returned by Forward selection method includes the 5 predictor variables leaving *Population* and *Education*, as the *global best* model returned by exhaustive search.

Using the backward selection method, we get the same set of predictor variables defining the *best* model as before.

```r
step(full, direction = "backward", trace=FALSE)
```

```


Call:
lm(formula = Sales ~ CompPrice + Income + Advertising + Price + 
    Age, data = carseats)

Coefficients:
(Intercept)    CompPrice       Income  Advertising        Price  
    7.11372      0.09395      0.01312      0.13070     -0.09263  
        Age  
   -0.04503  
```

THANK YOU