Linear Model Selection

Langtao Chen

Initial: Jan 24, 2019 Update: Mar 13, 2021

Contents

1. Data

In this example, we use the used Toyota Corolla dataset to demonstrate how to conduct linear model selection. The response variable is price of the used car.

```
df <- read.csv('ToyotaCorolla_FullData.csv')
str(df)
```

```
## 'data.frame':    1436 obs. of  39 variables:
##  $ Id                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Model             : chr  "TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors" "TOYOTA Corolla 2.0 D4D HA
##  $ Price             : int  13500 13750 13950 14950 13750 12950 16900 18600 21500 12950 ...
##  $ Age_08_04         : int  23 23 24 26 30 32 27 30 27 23 ...
##  $ Mfg_Month         : int  10 10 9 7 3 1 6 3 6 10 ...
##  $ Mfg_Year          : int  2002 2002 2002 2002 2002 2002 2002 2002 2002 2002 ...
##  $ KM                : int  46986 72937 41711 48000 38500 61000 94612 75889 19700 71138 ...
##  $ Fuel_Type         : chr  "Diesel" "Diesel" "Diesel" "Diesel" ...
##  $ HP                : int  90 90 90 90 90 90 90 90 192 69 ...
##  $ Met_Color         : int  1 1 1 0 0 0 1 1 0 0 ...
##  $ Color             : chr  "Blue" "Silver" "Blue" "Black" ...
##  $ Automatic         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CC                : int  2000 2000 2000 2000 2000 2000 2000 2000 1800 1900 ...
##  $ Doors             : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Cylinders         : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Gears             : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Quarterly_Tax     : int  210 210 210 210 210 210 210 210 100 185 ...
##  $ Weight            : int  1165 1165 1165 1165 1170 1170 1245 1245 1185 1105 ...
##  $ Mfr_Guarantee     : int  0 0 1 1 1 0 0 1 0 0 ...
##  $ BOVAG_Guarantee   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Guarantee_Period  : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ ABS               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Airbag_1          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Airbag_2          : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ Airco             : int  0 1 0 0 1 1 1 1 1 1 ...
##  $ Automatic_airco   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Boardcomputer     : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ CD_Player         : int  0 1 0 0 0 0 0 1 0 0 ...
##  $ Central_Lock      : int  1 1 0 0 1 1 1 1 1 0 ...
##  $ Powered_Windows   : int  1 0 0 0 1 1 1 1 1 0 ...
##  $ Power_Steering    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Radio             : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Mistlamps         : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ Sport_Model       : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Backseat_Divider  : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ Metallic_Rim      : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Radio_cassette    : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Parking_Assistant: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Tow_Bar           : int  0 0 0 0 0 0 0 0 0 0 ...
```

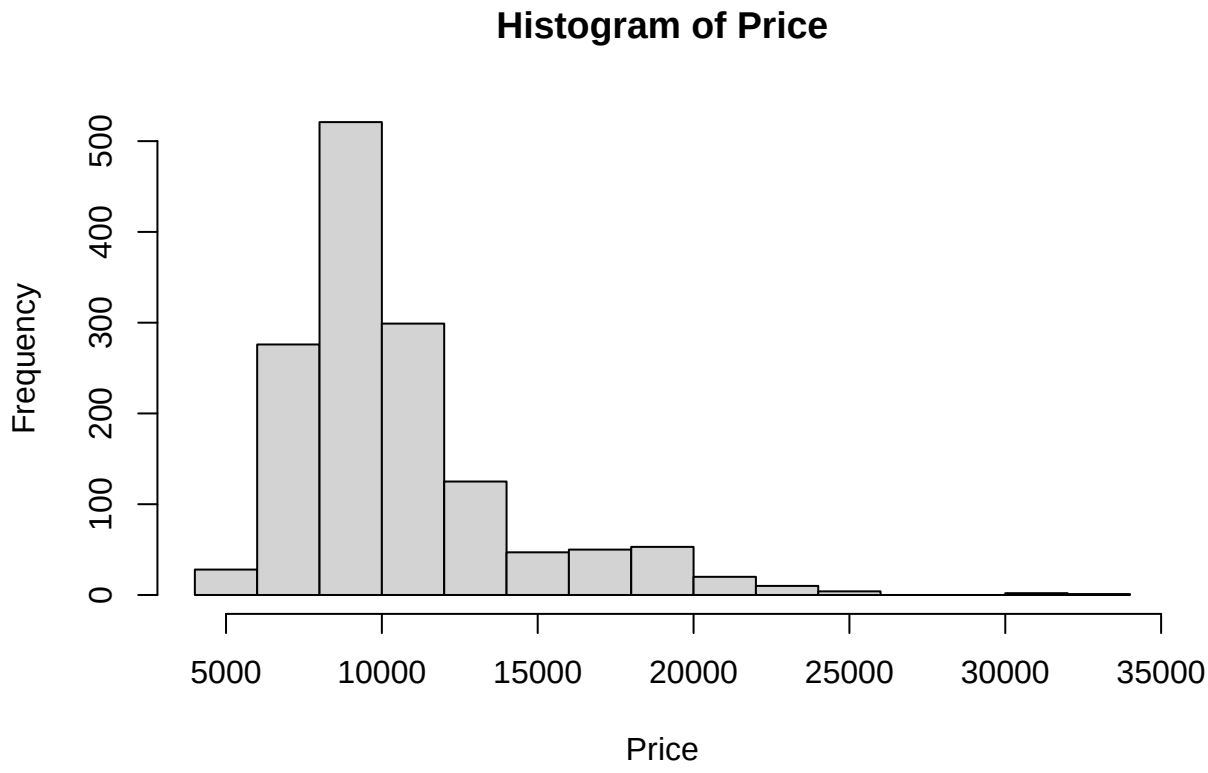
```
summary(df)
```

```
##        Id             Model                Price          Age_08_04
##  Min.   :   1.0   Length:1436        Min.   : 4350   Min.   : 1.00
```

```
##  1st Qu.: 361.8   Class :character   1st Qu.: 8450   1st Qu.:44.00
##  Median : 721.5   Mode  :character   Median : 9900   Median :61.00
##  Mean   : 721.6                      Mean   :10731   Mean   :55.95
##  3rd Qu.:1081.2                      3rd Qu.:11950   3rd Qu.:70.00
##  Max.   :1442.0                      Max.   :32500   Max.   :80.00
##    Mfg_Month         Mfg_Year          KM          Fuel_Type
##  Min.   : 1.000   Min.   :1998   Min.   :     1   Length:1436
##  1st Qu.: 3.000   1st Qu.:1998   1st Qu.: 43000   Class :character
##  Median : 5.000   Median :1999   Median : 63390   Mode  :character
##  Mean   : 5.549   Mean   :2000   Mean   : 68533
##  3rd Qu.: 8.000   3rd Qu.:2001   3rd Qu.: 87021
##  Max.   :12.000   Max.   :2004   Max.   :243000
##        HP          Met_Color         Color             Automatic
##  Min.   : 69.0   Min.   :0.0000   Length:1436        Min.   :0.00000
##  1st Qu.: 90.0   1st Qu.:0.0000   Class :character   1st Qu.:0.00000
##  Median :110.0   Median :1.0000   Mode  :character   Median :0.00000
##  Mean   :101.5   Mean   :0.6748                      Mean   :0.05571
##  3rd Qu.:110.0   3rd Qu.:1.0000                      3rd Qu.:0.00000
##  Max.   :192.0   Max.   :1.0000                      Max.   :1.00000
##        CC            Doors         Cylinders     Gears       Quarterly_Tax
##  Min.   : 1300   Min.   :2.000   Min.   :4   Min.   :3.000   Min.   : 19.00
##  1st Qu.: 1400   1st Qu.:3.000   1st Qu.:4   1st Qu.:5.000   1st Qu.: 69.00
##  Median : 1600   Median :4.000   Median :4   Median :5.000   Median : 85.00
##  Mean   : 1577   Mean   :4.033   Mean   :4   Mean   :5.026   Mean   : 87.12
##  3rd Qu.: 1600   3rd Qu.:5.000   3rd Qu.:4   3rd Qu.:5.000   3rd Qu.: 85.00
##  Max.   :16000   Max.   :5.000   Max.   :4   Max.   :6.000   Max.   :283.00
##      Weight     Mfr_Guarantee    BOVAG_Guarantee  Guarantee_Period
##  Min.   :1000   Min.   :0.0000   Min.   :0.0000   Min.   : 3.000
##  1st Qu.:1040   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.: 3.000
##  Median :1070   Median :0.0000   Median :1.0000   Median : 3.000
##  Mean   :1072   Mean   :0.4095   Mean   :0.8955   Mean   : 3.815
##  3rd Qu.:1085   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 3.000
##  Max.   :1615   Max.   :1.0000   Max.   :1.0000   Max.   :36.000
##       ABS            Airbag_1         Airbag_2          Airco
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median :1.0000   Median :1.0000
##  Mean   :0.8134   Mean   :0.9708   Mean   :0.7228   Mean   :0.5084
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  Automatic_airco   Boardcomputer      CD_Player       Central_Lock
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000   Median :0.0000   Median :1.0000
##  Mean   :0.05641   Mean   :0.2946   Mean   :0.2187   Mean   :0.5801
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  Powered_Windows Power_Steering       Radio          Mistlamps
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :1.000   Median :1.0000   Median :0.0000   Median :0.000
##  Mean   :0.562   Mean   :0.9777   Mean   :0.1462   Mean   :0.257
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

```
##   Sport_Model     Backseat_Divider  Metallic_Rim    Radio_cassette
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :1.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3001   Mean   :0.7702   Mean   :0.2047   Mean   :0.1455
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  Parking_Assistant     Tow_Bar
##  Min.   :0.000000   Min.   :0.0000
##  1st Qu.:0.000000   1st Qu.:0.0000
##  Median :0.000000   Median :0.0000
##  Mean   :0.002786   Mean   :0.2779
##  3rd Qu.:0.000000   3rd Qu.:1.0000
##  Max.   :1.000000   Max.   :1.0000
```

```r
hist(df$Price,
     main = 'Histogram of Price',
     xlab = 'Price')
```



2. Data Preparation

2.1. Data Cleansing

Let's remove Id from the dataset because it's not a good predictor. We remove the Cylinders column since it does not have variability in the dataset (all cars have 4 cylinders). We also remove the model column as it contains a large number of classes (i.e., 319 levels) that have redundant information with other columns.

```
df$Id <- NULL
df$Cylinders <- NULL
df$Model <- NULL
```

Let's explore the relationship between age and manufacturing year and month.

```
summary(lm(Age_08_04 ~ Mfg_Month + Mfg_Year, data = df))
```

```
##
## Call:
## lm(formula = Age_08_04 ~ Mfg_Month + Mfg_Year, data = df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.584e-08 -9.400e-12  1.730e-11  4.400e-11  1.317e-10
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  2.406e+04  2.347e-08  1.025e+12   <2e-16 ***
## Mfg_Month   -1.000e+00  5.392e-12 -1.855e+11   <2e-16 ***
## Mfg_Year    -1.200e+01  1.174e-11 -1.022e+12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.839e-10 on 1433 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.307e+23 on 2 and 1433 DF,  p-value: < 2.2e-16
```

Let's remove manufacturing month and year from the dataset since there is a linear dependence between the age and the two variables.

```
df$Mfg_Month <- NULL
df$Mfg_Year <- NULL
```

2.2. Data Split

Let's split the whole dataset into training (80%) and test (20%) sets.

```
set.seed(123)

train <- sample(1:nrow(df),nrow(df)*0.80)

train_df <- df[train,]
test_df <- df[-train,]

dim(train_df)
```

```
## [1] 1148   34
```

```
dim(test_df)
```

```
## [1] 288  34
```

2.3. Create Input Matrix

Many R functions such as lm() support using formula to specify model. But it's not convinient to use formula to conduc model selection when there are qualitative predictors. We can use the model.matrix() method to create the input matrix. The model.matrix() method can automatically transform qualitative variables into dummy variables.

```
# Create input matrix, removing the intercept
train_x <- model.matrix(Price ~ ., data = train_df)[,-1]
colnames(train_x)
```

```
##  [1] "Age_08_04"         "KM"                "Fuel_TypeDiesel"
##  [4] "Fuel_TypePetrol"   "HP"                "Met_Color"
##  [7] "ColorBlack"        "ColorBlue"         "ColorGreen"
## [10] "ColorGrey"         "ColorRed"          "ColorSilver"
## [13] "ColorViolet"       "ColorWhite"        "ColorYellow"
## [16] "Automatic"         "CC"                "Doors"
## [19] "Gears"             "Quarterly_Tax"     "Weight"
## [22] "Mfr_Guarantee"     "BOVAG_Guarantee"   "Guarantee_Period"
## [25] "ABS"               "Airbag_1"          "Airbag_2"
## [28] "Airco"             "Automatic_airco"   "Boardcomputer"
## [31] "CD_Player"         "Central_Lock"      "Powered_Windows"
## [34] "Power_Steering"    "Radio"             "Mistlamps"
## [37] "Sport_Model"       "Backseat_Divider"  "Metallic_Rim"
## [40] "Radio_cassette"    "Parking_Assistant" "Tow_Bar"
```

```
train_y <- train_df$Price
```

```
test_x <- model.matrix(Price ~ ., data = test_df)[,-1]
colnames(test_x)
```

```
##  [1] "Age_08_04"         "KM"                "Fuel_TypeDiesel"
##  [4] "Fuel_TypePetrol"   "HP"                "Met_Color"
##  [7] "ColorBlue"         "ColorGreen"        "ColorGrey"
## [10] "ColorRed"          "ColorSilver"       "ColorWhite"
## [13] "ColorYellow"       "Automatic"         "CC"
## [16] "Doors"             "Gears"             "Quarterly_Tax"
## [19] "Weight"            "Mfr_Guarantee"     "BOVAG_Guarantee"
## [22] "Guarantee_Period"  "ABS"               "Airbag_1"
## [25] "Airbag_2"          "Airco"             "Automatic_airco"
## [28] "Boardcomputer"     "CD_Player"         "Central_Lock"
## [31] "Powered_Windows"   "Power_Steering"    "Radio"
## [34] "Mistlamps"         "Sport_Model"       "Backseat_Divider"
## [37] "Metallic_Rim"      "Radio_cassette"    "Parking_Assistant"
## [40] "Tow_Bar"
```

```
test_y <- test_df$Price
```

3. Linear Model Selection

3.1. Best Subset Selection

We can use the regsubsets() method in leaps package to conduct the best subset selection.

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.4
```

```
old_time <- Sys.time()

fit_best <- regsubsets(x = train_x, y = train_y, nvmax = 360, really.big = TRUE)
fit_best_sum <- summary(fit_best)

new_time <- Sys.time()

cat('Time Spent in Best Subset Selection:', new_time - old_time, 'seconds')
```

```
## Time Spent in Best Subset Selection: 30.10058 seconds
```

We notice that best subset selection is time consuming, compared with step-wise selection.

```
# Print the adjusted R2
fit_best_sum$adjr2
```

```
##  [1] 0.7660608 0.8188271 0.8470323 0.8702456 0.8785259 0.8825244 0.8860379
##  [8] 0.8893102 0.8908328 0.8919930 0.8926888 0.8933197 0.8937603 0.8941953
## [15] 0.8946144 0.8949544 0.8952265 0.8954406 0.8956363 0.8957894 0.8959401
## [22] 0.8960726 0.8961985 0.8962982 0.8963806 0.8963977 0.8964318 0.8964670
## [29] 0.8964755 0.8964424 0.8964040 0.8963670 0.8963218 0.8962736 0.8962149
## [36] 0.8961437 0.8960896 0.8960163 0.8959373 0.8958464 0.8957533 0.8956590
```

```
# Find the position where adjusted R2 is the largest
which.max(fit_best_sum$adjr2)
```

```
## [1] 29
```

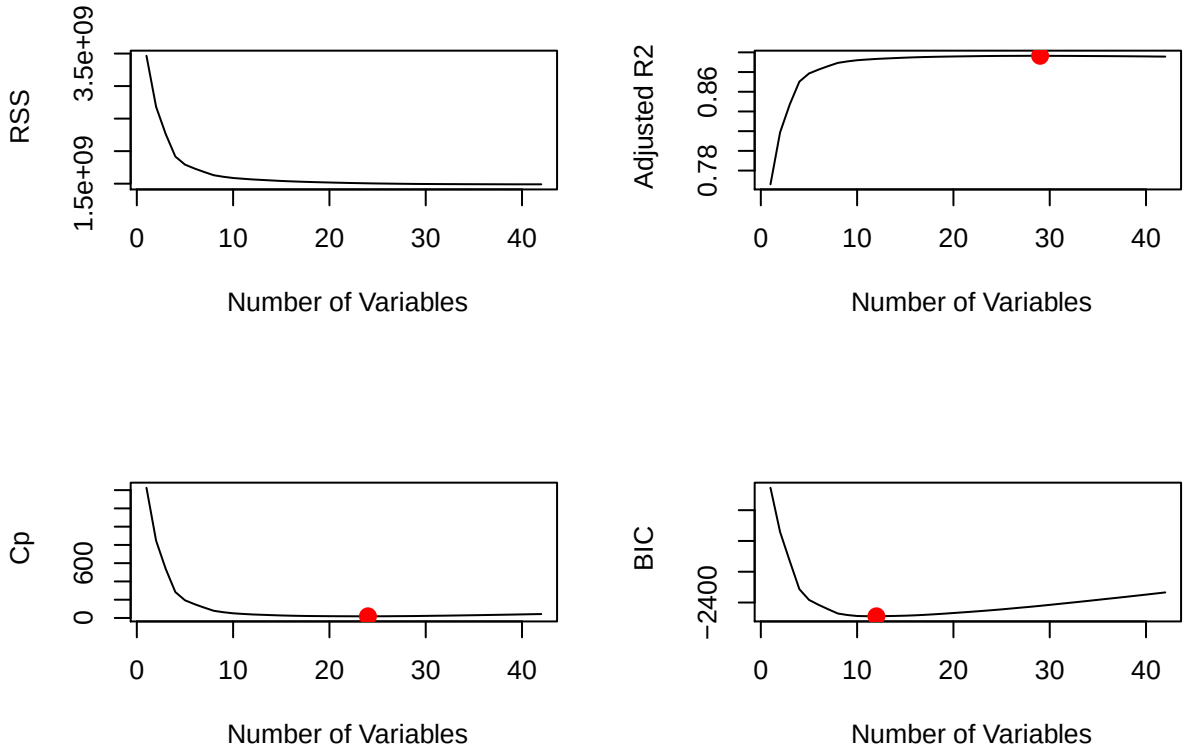
```
# Find the position where Cp is the smallest
which.min(fit_best_sum$cp)
```

```
## [1] 24
```

```
# Find the position where BIC is the smallest
which.min(fit_best_sum$bic)
```

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

```r
# Plot RSS, Adj R2, Cp, and BIC across the number of variables

par(mfrow =c(2,2))

plot(fit_best_sum$rss,
     xlab=" Number of Variables",
     ylab=" RSS",
     type="l")

plot(fit_best_sum$adjr2,
     xlab =" Number of Variables",
     ylab=" Adjusted R2",
     type="l")
points(which.max(fit_best_sum$adjr2),
       fit_best_sum$adjr2[which.max(fit_best_sum$adjr2)],
       col ="red",cex =2, pch =20)

plot(fit_best_sum$cp,
     xlab =" Number of Variables",
     ylab=" Cp",
     type="l")
points(which.min(fit_best_sum$cp),
       fit_best_sum$cp[which.min(fit_best_sum$cp)],
       col ="red",cex =2, pch =20)

plot(fit_best_sum$bic,
     xlab =" Number of Variables",
     ylab=" BIC",
     type="l")
points(which.min(fit_best_sum$bic),
       fit_best_sum$bic[which.min(fit_best_sum$bic)],
       col ="red",cex =2, pch =20)
```



If we use BIC as the criterion of model selection, the final model should include 12 predictors.

```
coef(fit_best, 12)
```

```
##      (Intercept)        Age_08_04               KM  Fuel_TypePetrol
##    -2.832474e+03    -1.144467e+02    -1.539628e-02     1.803959e+03
##               HP       ColorWhite    Quarterly_Tax           Weight
##     1.300159e+01    -7.397365e+02     1.381415e+01     1.493232e+01
##  BOVAG_Guarantee Guarantee_Period         Airbag_2  Automatic_airco
##     4.983760e+02     6.684807e+01    -2.613740e+02     2.673246e+03
##  Powered_Windows
##     4.105154e+02
```

3.2. Stepwise Selection

We can also use the regsubsets() method in leaps package to conduct the stepwise selection.

3.2.1. Forward Stepwise Selection

```
old_time <- Sys.time()

fit_fwd <- regsubsets(x = train_x, y = train_y, nvmax = 42, method = 'forward')
fit_fwd_sum <- summary(fit_fwd)
```

```
new_time <- Sys.time()
cat('Time Spent in Forward Stepwise Selection:', new_time - old_time, 'seconds')
```

```
## Time Spent in Forward Stepwise Selection: 0.01496005 seconds
```

We notice that stepwise selection is computationally cheap, compared with the best subset selection.

```
# Find the position where adjusted R2 is the largest
which.max(fit_fwd_sum$adjr2)
```

```
## [1] 29
```

```
# Find the position where Cp is the smallest
which.min(fit_fwd_sum$cp)
```

```
## [1] 24
```

```
# Find the position where BIC is the smallest
which.min(fit_fwd_sum$bic)
```

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

If we use BIC as the criterion of model selection, the final model should include 12 predictors.

```
coef(fit_fwd, 12)
```

```
##      (Intercept)        Age_08_04               KM  Fuel_TypePetrol
##    -2.832473e+03    -1.144467e+02    -1.539628e-02     1.803959e+03
##               HP       ColorWhite    Quarterly_Tax           Weight
##     1.300159e+01    -7.397365e+02     1.381415e+01     1.493232e+01
##  BOVAG_Guarantee Guarantee_Period         Airbag_2  Automatic_airco
##     4.983760e+02     6.684807e+01    -2.613740e+02     2.673246e+03
##  Powered_Windows
##     4.105154e+02
```

3.2.2. Backward Stepwise Selection

```
old_time <- Sys.time()

fit_bwd <- regsubsets(x = train_x, y = train_y, nvmax = 42, method = 'backward')
fit_bwd_sum <- summary(fit_bwd)

new_time <- Sys.time()
cat('Time Spent in Backward Stepwise Selection:', new_time - old_time, 'seconds')
```

```
## Time Spent in Backward Stepwise Selection: 0.01396394 seconds
```

```
# Find the position where adjusted R2 is the largest
which.max(fit_bwd_sum$adjr2)
```

```
## [1] 31
```

```
# Find the position where Cp is the smallest
which.min(fit_bwd_sum$cp)
```

```
## [1] 25
```

```
# Find the position where BIC is the smallest
which.min(fit_bwd_sum$bic)
```

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

If we use BIC as the criterion of model selection, the final model should include 11 predictors.

```
coef(fit_bwd, 11)
```

```
##       (Intercept)         Age_08_04                KM  Fuel_TypePetrol
##     -2.853828e+03     -1.143706e+02     -1.548583e-02      1.837074e+03
##                HP     Quarterly_Tax            Weight  BOVAG_Guarantee
##      1.286837e+01      1.395003e+01      1.488464e+01      5.047168e+02
## Guarantee_Period          Airbag_2  Automatic_airco  Powered_Windows
##      6.819308e+01     -2.492768e+02      2.684668e+03      4.308582e+02
```

We note that best subset, forward stepwise, and backward stepwise selection methods may result in different final models.

3.3. Test the Peformance of Linear Model Selection.

In the above, we used best subset selection and stepwise selection on the training dataset. We can compare the full model and the more parsimonous model on the test dataset. As an example, let's compare the full model with all predictors and the refined model suggested by best subset selection using Cp as the criterion.

```
# Predictor names in the best subset solution with 24 predictors
names(coef(fit_best, 24))[-1]
```

```
##  [1] "Age_08_04"        "KM"               "Fuel_TypeDiesel"  "Fuel_TypePetrol"
##  [5] "HP"               "Met_Color"        "ColorBlue"        "ColorGreen"
##  [9] "ColorRed"         "ColorWhite"       "Quarterly_Tax"    "Weight"
## [13] "Mfr_Guarantee"    "BOVAG_Guarantee"  "Guarantee_Period" "Airbag_2"
## [17] "Airco"            "Automatic_airco"  "CD_Player"        "Powered_Windows"
## [21] "Sport_Model"      "Backseat_Divider" "Metallic_Rim"     "Tow_Bar"
```

```
train_df_subset <- data.frame(train_x[,names(coef(fit_best, 24))[-1]])
train_df_subset$Price <- train_y
str(train_df_subset)
```

```
## 'data.frame':    1148 obs. of  25 variables:
##  $ Age_08_04       : num  49 46 8 52 41 68 73 79 72 73 ...
##  $ KM              : num  97600 69574 5000 49432 123425 ...
##  $ Fuel_TypeDiesel : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ Fuel_TypePetrol : num  1 1 1 1 0 1 1 1 1 1 ...
##  $ HP              : num  110 97 110 110 69 110 86 110 110 110 ...
##  $ Met_Color       : num  1 0 1 1 1 1 1 1 1 0 ...
##  $ ColorBlue       : num  0 1 0 1 1 0 0 0 1 0 ...
##  $ ColorGreen      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ColorRed        : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ ColorWhite      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Quarterly_Tax   : num  69 85 85 69 185 85 69 85 85 85 ...
##  $ Weight          : num  1045 1065 1130 1050 1140 ...
##  $ Mfr_Guarantee   : num  0 1 0 1 0 1 0 0 1 0 ...
##  $ BOVAG_Guarantee : num  1 1 1 1 1 1 1 1 1 0 ...
##  $ Guarantee_Period: num  6 3 3 3 3 3 3 3 3 3 ...
##  $ Airbag_2        : num  0 1 1 1 1 1 1 1 1 0 ...
##  $ Airco           : num  1 1 1 1 1 0 0 1 0 1 ...
##  $ Automatic_airco : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ CD_Player       : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ Powered_Windows : num  1 1 1 1 1 0 1 1 1 1 ...
##  $ Sport_Model     : num  0 0 1 0 0 1 0 0 0 0 ...
##  $ Backseat_Divider: num  0 1 1 1 1 1 0 1 1 1 ...
##  $ Metallic_Rim    : num  0 0 1 0 0 0 0 1 0 0 ...
##  $ Tow_Bar         : num  1 0 0 1 0 0 0 1 0 0 ...
##  $ Price           : int  10900 10750 21950 10250 13250 8950 7950 8950 7950 8500 ...
```

3.3.1. Performance of the Final Model

```
final_model <- lm(Price ~ ., data = train_df_subset)
summary(final_model)
```

```
##
## Call:
## lm(formula = Price ~ ., data = train_df_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8893.7  -724.7   -19.7   630.1  5239.0
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.054e+03  1.308e+03  -1.571 0.116465
## Age_08_04        -1.101e+02  3.167e+00 -34.764  < 2e-16 ***
## KM               -1.577e-02  1.305e-03 -12.084  < 2e-16 ***
## Fuel_TypeDiesel   5.362e+02  3.546e+02   1.512 0.130782
## Fuel_TypePetrol   2.164e+03  3.616e+02   5.986 2.90e-09 ***
## HP                1.550e+01  3.547e+00   4.369 1.36e-05 ***
## Met_Color        -1.187e+02  8.232e+01  -1.442 0.149512
## ColorBlue        -1.600e+02  9.299e+01  -1.720 0.085627 .
## ColorGreen       -3.058e+02  1.039e+02  -2.943 0.003316 **
## ColorRed         -2.344e+02  1.008e+02  -2.326 0.020208 *
```

```
## ColorWhite       -8.859e+02  2.516e+02  -3.521 0.000447 ***
## Quarterly_Tax     1.498e+01  1.833e+00   8.172 8.14e-16 ***
## Weight            1.356e+01  1.263e+00  10.739  < 2e-16 ***
## Mfr_Guarantee     2.173e+02  7.595e+01   2.862 0.004294 **
## BOVAG_Guarantee   4.490e+02  1.310e+02   3.427 0.000632 ***
## Guarantee_Period  6.208e+01  1.325e+01   4.684 3.15e-06 ***
## Airbag_2         -3.024e+02  1.054e+02  -2.868 0.004208 **
## Airco             1.462e+02  8.847e+01   1.653 0.098708 .
## Automatic_airco   2.486e+03  1.851e+02  13.427  < 2e-16 ***
## CD_Player         2.187e+02  9.820e+01   2.228 0.026107 *
## Powered_Windows   3.451e+02  8.706e+01   3.964 7.83e-05 ***
## Sport_Model       2.473e+02  8.589e+01   2.880 0.004057 **
## Backseat_Divider -2.468e+02  1.265e+02  -1.951 0.051353 .
## Metallic_Rim      1.482e+02  9.101e+01   1.628 0.103720
## Tow_Bar          -1.474e+02  8.081e+01  -1.825 0.068333 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1158 on 1123 degrees of freedom
## Multiple R-squared:  0.8985, Adjusted R-squared:  0.8963
## F-statistic: 414.1 on 24 and 1123 DF,  p-value: < 2.2e-16
```

```
# Calculate performance of the final model
price_pred <- predict(final_model, data.frame(test_x))

library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
postResample(pred = price_pred, obs = test_y)
```

```
##         RMSE     Rsquared          MAE
## 1093.3254572    0.9159974  847.5766989
```

3.3.2. Performance of the Full Model

```
# Fit a full linear model with all predictors
full_model <- lm(Price ~ ., data = train_df)
price_pred <- predict(full_model, test_df)
```

```
# Calculate performance of the final model
library(caret)
postResample(pred = price_pred, obs = test_y)
```

```
##         RMSE     Rsquared          MAE
## 1084.5416922    0.9174113  842.8434130
```

From the above results, we can find that the final model suggested by the best subset selection method and the full model with all predictors have similar performance.