MATH - 4360: Linear Statistical Models

Chapter 10 - Variable Selection and Model Building

Suthakaran Ratnasingam

Criteria for Evaluating Subset Regression Models

- Two key aspects of the variable selection problem are generating the subset models and deciding if one subset is better than another.
- How can we evaluate and compare different candidate models?
  - Coefficient of multiple determination
  - Adjusted R^2
  - Residual mean square
  - Mallows C_p Statistic
  - AIC or BIC

Let's fit a full model for the `Hald Cement Data` given in Example 10.1

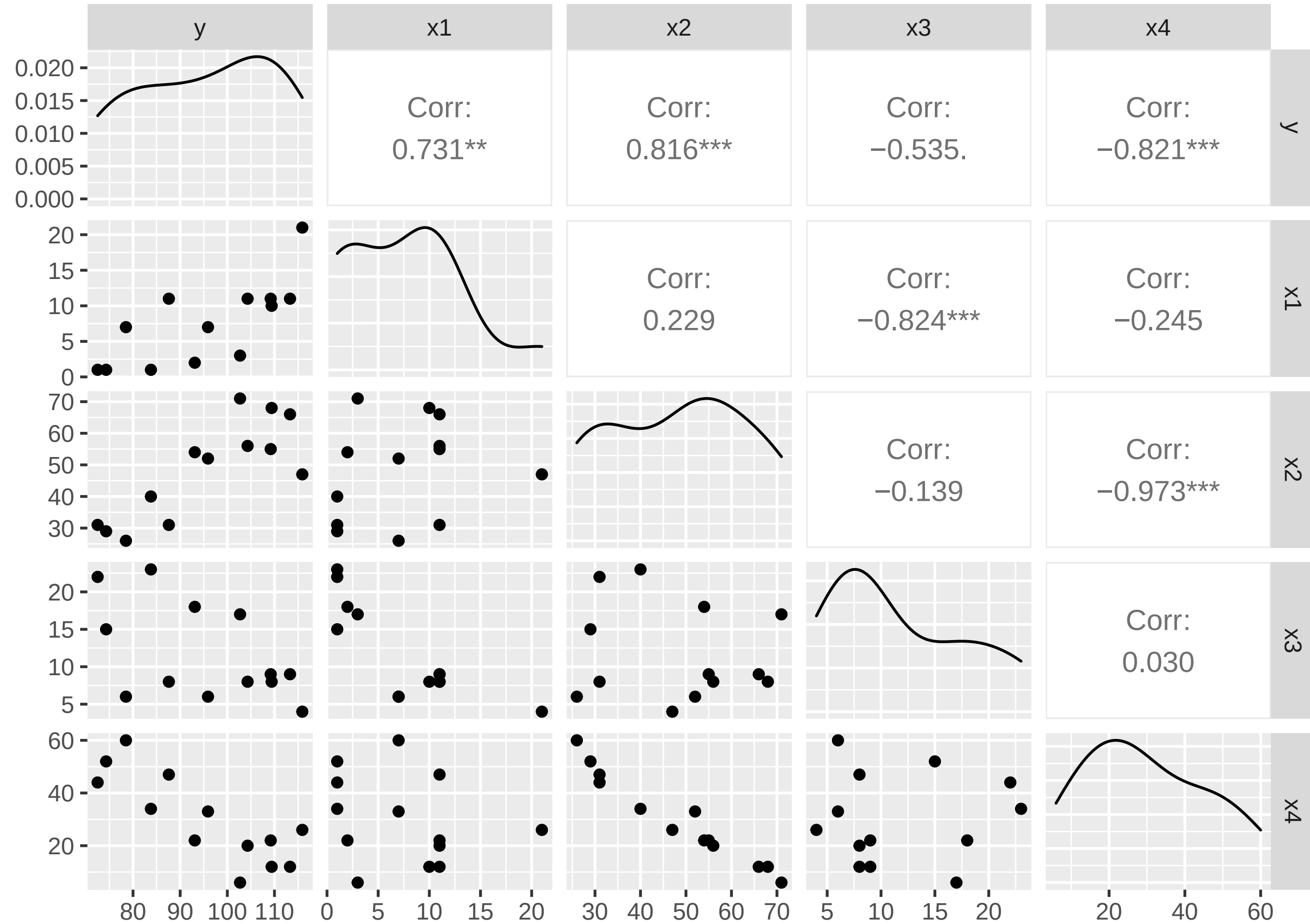
```
rm(list = ls())
# I assume that you have installed the following R packages. If not, please install
# them using the R function: install.packages('package_name')
library(olsrr)
library(ggfortify)
library(ggplot2)
library(tidyverse)
library(car)
library(Rcpp)
library(GGally)
library(leaps)
library(matlib) # enables function inv()
data1 = read.table("D:\\CSUSB\\Fall 2021\\MATH 4360\\RNotes\\ex101.txt", header = TRUE)
head(data1)
```

```
##       y x1 x2 x3 x4
## 1  78.5  7 26  6 60
## 2  74.3  1 29 15 52
## 3 104.3 11 56  8 20
## 4  87.6 11 31  8 47
## 5  95.9  7 52  6 33
## 6 109.2 11 55  9 22
```

```
names(data1)
```

```
## [1] "y"  "x1" "x2" "x3" "x4"
```

```
n = nrow(data1)
GGally::ggpairs(data1, progress=FALSE)
```




```
n = nrow(data1)
full_model = lm(y ~ ., data = data1)
```

Coefficient of Multiple Determination

Let R_p^2 denote the coefficient of multiple determination for a subset regression model with p terms, that is, $p-1$ regressors and an intercept term β_0. Computationally,

$$R_p^2 = \frac{SS_R(p)}{SS_T} = 1 - \frac{SS_{Res}(p)}{SS_T}$$

where $SS_R(p)$ and $SS_{Res}(p)$ denote the regression sum of squares and the residual sum of squares, respectively, for a $p-$ term subset model.

- Choose a best model by comparing R^2 for different models. Unfortunately, R_p^2 increases with p with a maximum when $p = K$
- Models having large values of R_p^2 are preferred.

Adjusted R^2

Instead of R^2, we may use the adjusted R^2 as it may be more interpretable. The ajusted R^2 for a $p-$term model is

$$R^2_{Adj,p} = 1 - \left(\frac{n-1}{p-1}\right)(1 - R_p^2)$$

- We can then choose a model based on the largest $R^2_{Adj,p}$

```
summary(full_model)$r.squared
```

```
## [1] 0.9823756
```

Residual Mean Square

The residual mean square for a subset regression model, for example, with p regressors

$$MS_{res}(p) = \frac{SS_{Res}(p)}{n-p}$$

may be used as a model evaluation criterion

```
summary(full_model)$adj.r.squared
```

```
## [1] 0.9735634
```

Akaike's information criterion (AIC)

The Akaike's information criterion statistic is given as

$$AIC = -2l(\hat{\beta}, \sigma^2; y) + 2p$$
$$AIC = n \ln\left(\frac{SS_{Res}}{n}\right) + 2p$$

where $SS_{Res} = y'Hy = y'X(X'X)^{-1}X'y$

```
AIC(full_model)
```

```
## [1] 65.83669
```

Bayesian information criterion (BIC)

Similar to AIC, the Bayesian information criterion is based on maximizing the posterior distribution of the model given the observations y. In the case of linear regression model, it is defined as

$$BIC = -2\ln(L) + p\ln(n)$$

```
BIC(full_model)
```

```
## [1] 69.22639
```

Mallows C_p Statistic

Mallows C_p Statistic Mallows [1964, 1966, 1973, 1995] has proposed a criterion that is related to the mean square error of a fitted value

$$C_p = \frac{SS_{Res}(p)}{\hat{\sigma}^2} - n + 2p$$

I wrote two `R` functions `find_Cp()` which computes the Mallows 's C_p Statistic

```r
# Compute Cp statistics
find_Cp = function(model){
  SS_res = deviance(model)
  sigma_sq = (summary(model)$sigma)^2
  p = length(coef(model))
  Cp = SS_res/sigma_sq - n + 2 * p
  return(Cp)
}
find_Cp(full_model)
```

```
## [1] 5
```

Here I have another simple R function `find_all_measures()` gives all diagnostic measures for the fitted model. For example, we will find all measures for our full model, `full_model`.

```r
# Find all measures
find_all_measures = function(model){
  SS_res = deviance(model)
  MS_res = anova(model)['Residuals', 'Mean Sq']
  R_Sq = summary(model)$r.squared
  p = length(coef(model))
  Adj_R_Sq = summary(model)$adj.r.squared
  Cp = find_Cp(model)
  AIC = AIC(model)
  BIC = BIC(model)
  return(data.frame(p = p, SS_res =  SS_res, R_Sq = R_Sq,
                    Adj_R_Sq = Adj_R_Sq, MS_res = MS_res,
                    Cp =  Cp, AIC = AIC, BIC = BIC))
}
find_all_measures(full_model)
```

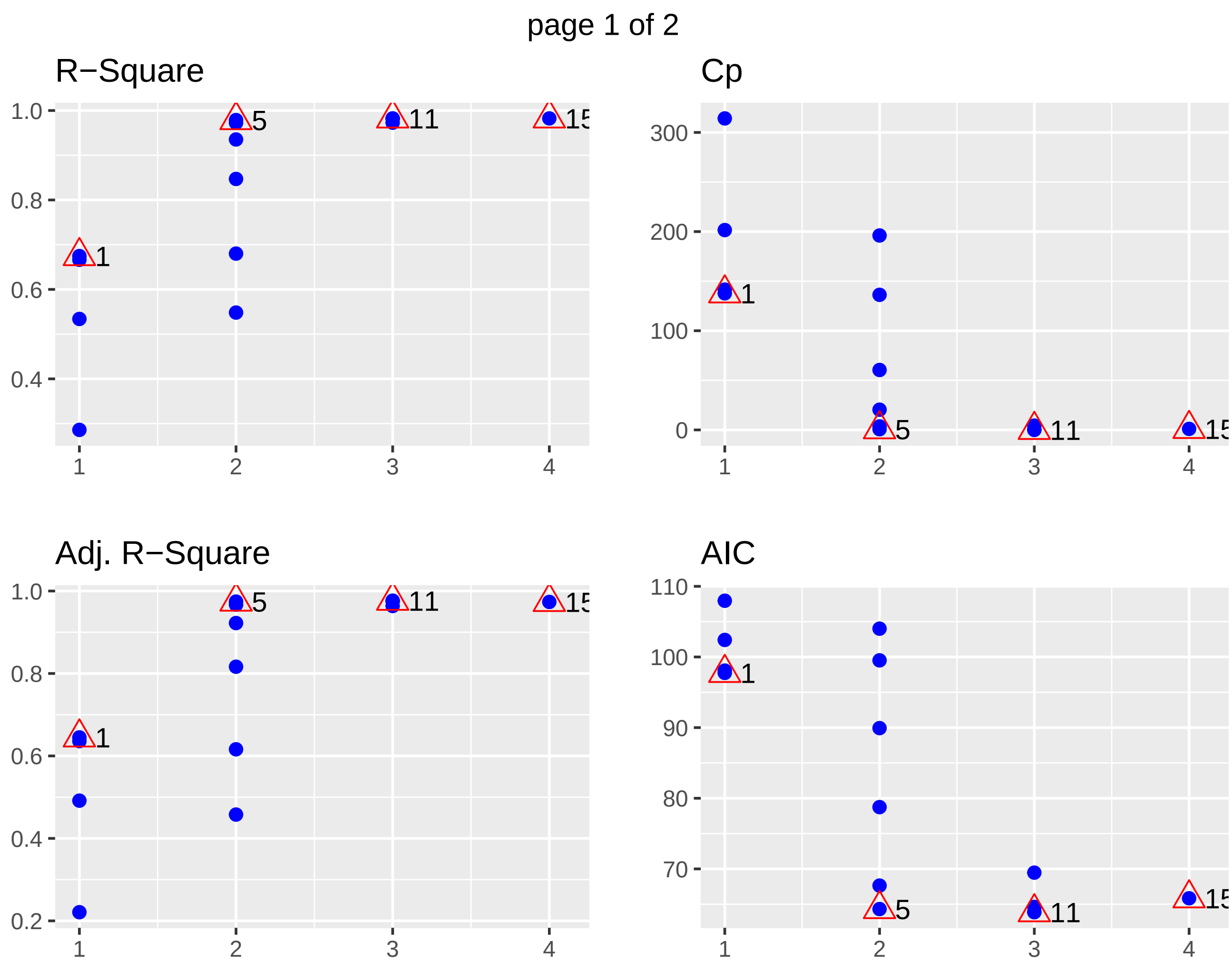
```
##   p   SS_res      R_Sq  Adj_R_Sq   MS_res Cp      AIC      BIC
## 1 5 47.86364 0.9823756 0.9735634 5.982955  5 65.83669 69.22639
```

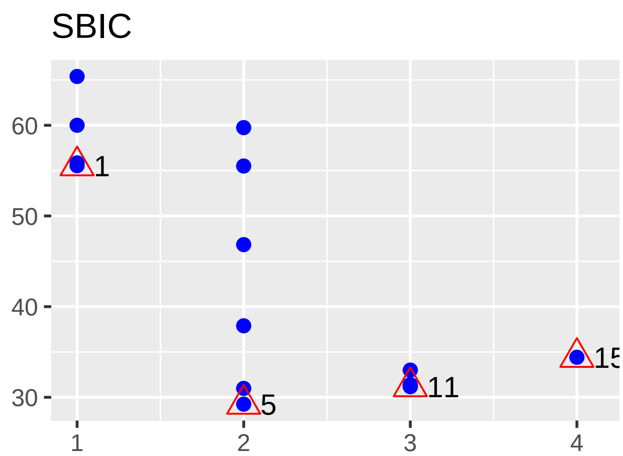
Let's find the summary of all possible regressions for the Hald Cement Data. Since there are $k = 4$ candidate regressors, there are $2^4 = 16$ possible regression equations if we always include the intercept β_0. The R function `leaps()` in the `leaps` package performs an all possible regressions methodology. However, I am using the R function `ols_step_all_possible()` in the `olsrr` package.

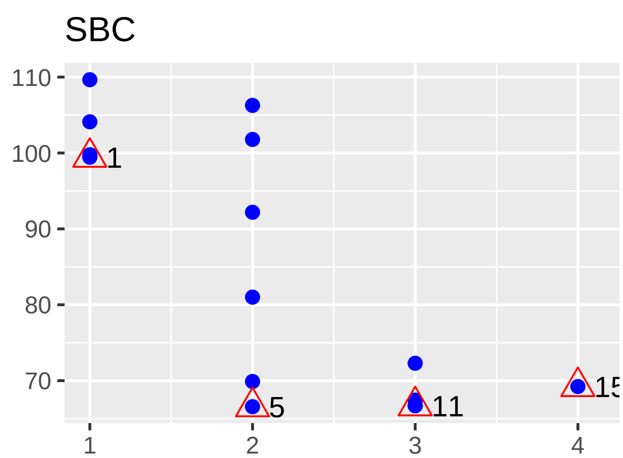
```r
all_cand = ols_step_all_possible(full_model)
all_cand
```

```
##    Index N  Predictors  R-Square Adj. R-Square Mallow's Cp
## 4      1 1          x4 0.6745420     0.6449549  138.730833
## 2      2 1          x2 0.6662683     0.6359290  142.486407
## 1      3 1          x1 0.5339480     0.4915797  202.548769
## 3      4 1          x3 0.2858727     0.2209521  315.154284
## 5      5 2       x1 x2 0.9786784     0.9744140    2.678242
## 7      6 2       x1 x4 0.9724710     0.9669653    5.495851
## 10     7 2       x3 x4 0.9352896     0.9223476   22.373112
## 8      8 2       x2 x3 0.8470254     0.8164305   62.437716
## 9      9 2       x2 x4 0.6800604     0.6160725  138.225920
## 6     10 2       x1 x3 0.5481667     0.4578001  198.094653
## 12    11 3    x1 x2 x4 0.9823355     0.9764473    3.018233
## 11    12 3    x1 x2 x3 0.9822847     0.9763796    3.041280
## 13    13 3    x1 x3 x4 0.9812811     0.9750415    3.496824
## 14    14 3    x2 x3 x4 0.9728200     0.9637599    7.337474
## 15    15 4 x1 x2 x3 x4 0.9823756     0.9735634    5.000000
```

```
plot(all_cand)
```







Computational Techniques for Variable Selection

- To find the subset of variables to use in the final equation, it is natural to consider fitting models with various combinations of the candidate regressors.
- All Possible Regressions
  - This procedure requires that the analyst fit all the regression equations involving one candidate regressor, two candidate regressors, and so on. These equations are evaluated according to some suitable criterion and the "best" regression model selected.
  - If we assume that the intercept term β_0 is included in all equations, then if there are k candidate regressors, there are 2^k total equations to be estimated and examined.
  - For example, if $k = 4$, then there are $2^4 = 16$ possible equations, while if $k = 10$, there are $2^{10} = 1024$ possible regression equations.
- Stepwise Regression Methods
  - Evaluating all possible regressions can be burdensome computationally and for the analyst
  - An alternative might be to compare a scientifically meaningful subset of models.
  - These fit into 3 categories
    * Forward selection
    * Backward elimination
    * Stepwise regression

Forward Selection

This methodology assumes that there is no explanatory variable in the model except an intercept term. It adds variables one by one and tests the fitted model at each step using some suitable criterion.

```r
fit_intercpt_model = lm(y~1,data=data1)
step(fit_intercpt_model, direction="forward", scope = ~ x1 + x2 + x3 + x4)
```

```
## Start:  AIC=71.44
## y ~ 1
##
##        Df Sum of Sq     RSS    AIC
## + x4    1   1831.90  883.87 58.852
## + x2    1   1809.43  906.34 59.178
## + x1    1   1450.08 1265.69 63.519
## + x3    1    776.36 1939.40 69.067
## <none>             2715.76 71.444
##
## Step:  AIC=58.85
## y ~ x4
##
##        Df Sum of Sq    RSS    AIC
## + x1    1    809.10  74.76 28.742
## + x3    1    708.13 175.74 39.853
## <none>             883.87 58.852
## + x2    1     14.99 868.88 60.629
##
## Step:  AIC=28.74
## y ~ x4 + x1
##
##        Df Sum of Sq    RSS    AIC
## + x2    1    26.789 47.973 24.974
## + x3    1    23.926 50.836 25.728
## <none>             74.762 28.742
##
## Step:  AIC=24.97
## y ~ x4 + x1 + x2
##
##        Df Sum of Sq    RSS    AIC
## <none>             47.973 24.974
## + x3    1   0.10909 47.864 26.944
```

```
##
## Call:
## lm(formula = y ~ x4 + x1 + x2, data = data1)
##
## Coefficients:
## (Intercept)           x4           x1           x2
##     71.6483      -0.2365       1.4519       0.4161
```

```r
# You can also use the following R function. However, this function needs full model for computation
full_model = lm(y ~ ., data = data1)
forw = ols_step_forward_p(full_model)
forw
```

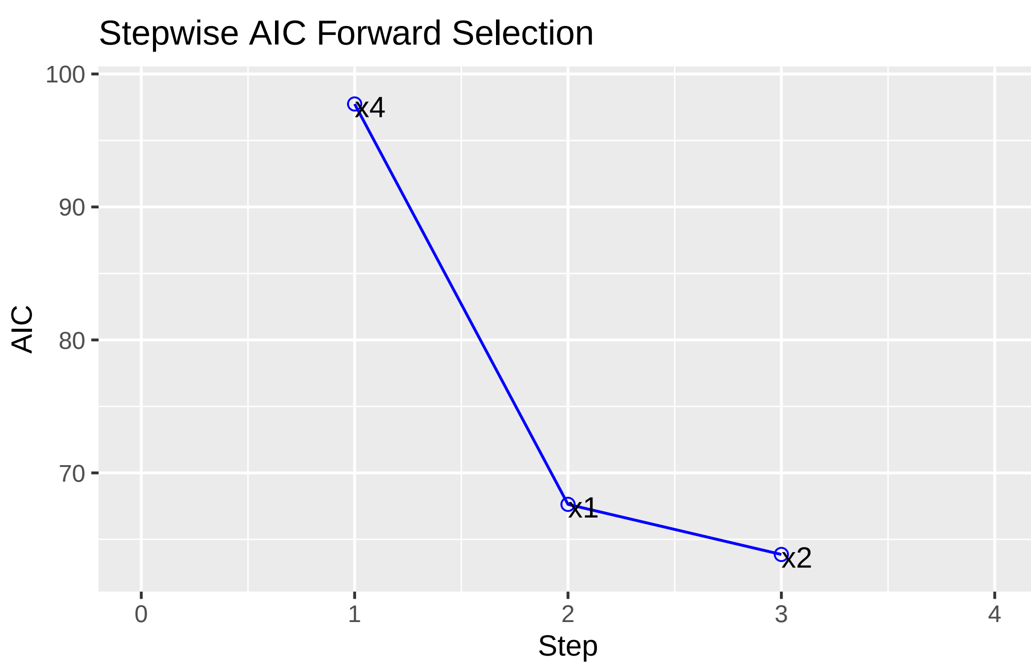
```
##
##                               Selection Summary
## ---------------------------------------------------------------------------
##         Variable                   Adj.
## Step    Entered     R-Square    R-Square      C(p)        AIC       RMSE
```

```
## -------------------------------------------------------------------------
##    1    x4           0.6745      0.6450    138.7308    97.7440    8.9639
##    2    x1           0.9725      0.9670      5.4959    67.6341    2.7343
##    3    x2           0.9823      0.9764      3.0182    63.8663    2.3087
## -------------------------------------------------------------------------
```

```
#plot(forw) # using the plot() function, you can visualize graphically
forw_AIC = ols_step_forward_aic(full_model) # Here I specified the selection criterion AIC
forw_AIC
```

```
##
##                          Selection Summary
## ------------------------------------------------------------------
## Variable      AIC       Sum Sq       RSS       R-Sq      Adj. R-Sq
## ------------------------------------------------------------------
## x4           97.744    1831.896    883.867    0.67454      0.64495
## x1           67.634    2641.001     74.762    0.97247      0.96697
## x2           63.866    2667.790     47.973    0.98234      0.97645
## ------------------------------------------------------------------
```

```
plot(forw_AIC)
```



Backward Elimination Procedure

- This methodology is contrary to the forward selection procedure. The forward selection procedure starts with no explanatory variable in the model and keeps on adding one variable at a time until a suitable model is obtained.
- The backward elimination methodology begins with all explanatory variables and keeps on deleting one variable at a time until a suitable model is obtained.
- It is based on the following steps:
  - Consider all k explanatory variables and fit the model.
  - Drop the predictor that improves your selection criterion the least
  - Continue until there is no predictor that can be dropped and result in an improvement of your selection criterion, then all the remaining predictors define your final model.

```r
step(lm(y~.,data=data1),direction="backward")
```

```
## Start:  AIC=26.94
## y ~ x1 + x2 + x3 + x4
##
##        Df Sum of Sq    RSS    AIC
## - x3    1    0.1091 47.973 24.974
## - x4    1    0.2470 48.111 25.011
## - x2    1    2.9725 50.836 25.728
## <none>              47.864 26.944
## - x1    1   25.9509 73.815 30.576
##
## Step:  AIC=24.97
## y ~ x1 + x2 + x4
##
##        Df Sum of Sq    RSS    AIC
## <none>               47.97 24.974
## - x4    1      9.93  57.90 25.420
## - x2    1     26.79  74.76 28.742
## - x1    1    820.91 868.88 60.629
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4, data = data1)
##
## Coefficients:
## (Intercept)           x1           x2           x4
##     71.6483       1.4519       0.4161      -0.2365
```

```r
# Alternatively, you can use the following R function as well.
# backw = ols_step_backward_p(full_model, details = TRUE)  # if you need details summary
backw = ols_step_backward_p(full_model)
backw
```

```
##
##
##                          Elimination Summary
## -----------------------------------------------------------------------
##         Variable                  Adj.
## Step    Removed     R-Square    R-Square     C(p)       AIC       RMSE
## -----------------------------------------------------------------------
##    1    x3            0.9823      0.9764    3.0182    63.8663    2.3087
## -----------------------------------------------------------------------
```

```r
#plot(backw) # using the plot() function, you can visualize graphically
```

Stepwise Regression Procedure

A combination of forward selection and backward elimination procedure is the stepwise regression. It is a modification of forward selection procedure and has the following steps.

```r
fit_full = lm(y ~ ., data = data1)
step = ols_step_both_p(fit_full)
step
```

```
##
##                               Stepwise Selection Summary
## ---------------------------------------------------------------------------------------
```

```
##                       Added/                      Adj.
## Step     Variable     Removed      R-Square      R-Square      C(p)         AIC        RMSE
## ----------------------------------------------------------------------------------------------
##    1        x4        addition        0.675         0.645      138.7310     97.7440     8.9639
##    2        x1        addition        0.972         0.967        5.4960     67.6341     2.7343
##    3        x2        addition        0.982         0.976        3.0180     63.8663     2.3087
## ----------------------------------------------------------------------------------------------
```

```
#plot(step) # using the plot() function, you can visualize graphically
```

References

- Introduction to Linear Regression Analysis, 5th Edition, by Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining (Wiley), ISBN: 978-0-470-54281-1.
- R Core Team (2020). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.
- RStudio Team (2020). RStudio: Integrated Development Environment for R. Boston, MA: RStudio, PBC.