5.1.2 Multiple Linear Regression in R

Multiple Linear Regression :
It is the most common form of Linear Regression. Multiple Linear Regression basically describes how a single response variable Y depends linearly on a number of predictor variables.
The basic examples where Multiple Regression can be used are as follows:

1. The selling price of a house can depend on the desirability of the location, the number of bedrooms, the number of bathrooms, the year the house was built, the square footage of the lot, and a number of other factors.
2. The height of a child can depend on the height of the mother, the height of the father, nutrition, and environmental factors.

Estimation of the Model Parameters
Consider a multiple linear Regression model with k independent predictor variable x1, x2……, xk, and one response variable y.

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \epsilon$$

Suppose we have n observation on the k+1 variables and the variable of n should be greater than k.

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_i, \quad i = 1, \ldots, n$$

The basic goal in least-squares regression is to fit a hyper-plane into (k + 1)-dimensional space that minimizes the sum of squared residuals.

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2$$

Before taking the derivative with respect to the model parameters set them equal to zero and derive the least-squares normal equations that the parameters would have to fulfill.

The linear Regression model is written in the form as follows:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

In linear regression the least square parameters estimate b

$$\sum_{i=1}^{n} \epsilon_i^2 = \epsilon'\epsilon = (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$$

Imagine the columns of X to be fixed, they are the data for a specific problem and say b to be variable. We want to find the “best” b in the sense that the sum of squared residuals is minimized. The smallest that the sum of squares could be is zero.

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$$

Here y is the estimated response vector.
Following R code is used to implement Multiple Linear Regression on following dataset **data2 (https://drive.google.com/file/d/1LoSR4920Gqnh85IX5aSr30Izb81QHSJN/view?usp=sharing)** .

Note: Check this link to download the dataset: **https://drive.google.com/file/d/1LoSR4920Gqnh85IX5aSr30Izb81QHSJN/view?usp=sharing (https://drive.google.com/file/d/1LoSR4920Gqnh85IX5aSr30Izb81QHSJN/view?usp=sharing)**

the dataset looks like this:

```
> dataset
   R.D.Spend Administration Marketing.Spend      State   Profit
1   165349.2      136897.80        471784.1   New York 192261.8
2   162597.7      151377.59        443898.5 California 191792.1
3   153441.5      101145.55        407934.5    Florida 191050.4
4   144372.4      118671.85        383199.6   New York 182902.0
5   142107.3       91391.77        366168.4    Florida 166187.9
6   131876.9       99814.71        362861.4   New York 156991.1
7   134615.5      147198.87        127716.8 California 156122.5
8   130298.1      145530.06        323876.7    Florida 155752.6
9   120542.5      148718.95        311613.3   New York 152211.8
10  123334.9      108679.17        304981.6 California 149760.0
```

Step 1. Importing the Dataset

image.png

```
# Multiple Linear Regression

# Importing the dataset
dataset = read.csv('data2.csv')

# Encoding categorical data
dataset$State = factor(dataset$State,
                       levels = c('New York', 'California', 'Florida'),
                       labels = c(1, 2, 3))
dataset$State
```

Step 2: Splitting and scaling the dataset into training and test set

image.png

```
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Profit, SplitRatio = 0.8)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Feature Scaling
# training_set = scale(training_set)
# test_set = scale(test_set)
```

Step 3. Fitting the MLR to the training set

```
# Fitting Multiple Linear Regression to the Training set
regressor = lm(formula = Profit ~ .,
               data = training_set)
```

Step 4. Predicting the test results.

```
# Predicting the Test set results
y_pred = predict(regressor, newdata = test_set)
```

```
> regressor

Call:
lm(formula = Profit ~ ., data = training_set)

Coefficients:
    (Intercept)        R.D.Spend   Administration  Marketing.Spend
      2.816e+04        8.884e-01        5.670e-02        2.859e-02
         State2           State3
     -2.861e+03        9.172e+03
```

```
> y_pred
       5        8
179233.6 170602.2
```