Linear Regression with R

This lab manual is to demonstrate how to build a linear regression model in R.

To build a linear regression model, use

lm(formula, data)

formula	a formula in the form of outcome ~ predictor1 + predictor2 + … + predictor*N* A predictor is an attribute.
data	A data frame in which to interpret the predictors named in the formula

Building the linear regression model

We will be using the diabetes dataset. The dataset contains 11 attributes (columns) as follows.

- age: Age of the patient
- sex: Gender of the patient
- bmi: Body mass index
- bp: Blood pressure
- S1, S2, S3, S4, S5, S6: Six blood serum measurements
- Diabetes: A quantitative measure of disease progression one year after baseline

Download the diabetes dataset. Load the package as follows.

```
> install.packages("Metrics")
> library(Metrics)
```

Load the data into R.

```
> diabetes <- read.csv("diabetes.csv")
```

To view the data frame

```
> View(diabetes)
```

Let's examine the columns in the data frame.

```
> str(diabetes)
```

```
'data.frame':   442 obs. of  11 variables:
 $ age     : num  0.03808 -0.00188 0.0853 -0.08906 0.00538 ...
 $ sex     : num  0.0507 -0.0446 0.0507 -0.0446 -0.0446 ...
 $ bmi     : num  0.0617 -0.0515 0.0445 -0.0116 -0.0364 ...
 $ bp      : num  0.02187 -0.02633 -0.00567 -0.03666 0.02187 ...
 $ s1      : num  -0.04422 -0.00845 -0.0456 0.01219 0.00393 ...
 $ s2      : num  -0.0348 -0.0192 -0.0342 0.025 0.0156 ...
 $ s3      : num  -0.0434 0.07441 -0.03236 -0.03604 0.00814 ...
 $ s4      : num  -0.00259 -0.03949 -0.00259 0.03431 -0.00259 ...
 $ s5      : num  0.01991 -0.06833 0.00286 0.02269 -0.03199 ...
 $ s6      : num  -0.01765 -0.0922 -0.02593 -0.00936 -0.04664 ...
 $ diabetes: num  151 75 141 206 135 97 138 63 110 310 ...
```

We would like to predict if a patient diabetes progression given the attributes. As we can see column "diabetes" is in num (continuous value).

Then, we split the data into training dan test sets, in a ratio of 70:30. The training set is used for training and creating the model. The test set is to evaluate the accuracy of the model.

```
> sample_ind <- sample(nrow(diabetes), nrow(diabetes)*0.7)
> train <- diabetes[sample_ind,]
> test <- diabetes[-sample_ind,]
```

Assuming we want to predict the patients' diabetes given all the attributes. Now, let's build a linear regression model. To build model

```
> lr <- lm(diabetes ~ ., train)
```

We can display the summary of the model.

```
> summary(lr)
```

```
Call:
lm(formula = diabetes ~ ., data = diabetes)

Residuals:
     Min       1Q   Median       3Q      Max
-155.829  -38.534   -0.227   37.806  151.355

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  152.133      2.576  59.061  < 2e-16 ***
age          -10.012     59.749  -0.168 0.867000
sex         -239.819     61.222  -3.917 0.000104 ***
bmi          519.840     66.534   7.813 4.30e-14 ***
bp           324.390     65.422   4.958 1.02e-06 ***
s1          -792.184    416.684  -1.901 0.057947 .
s2           476.746    339.035   1.406 0.160389
s3           101.045    212.533   0.475 0.634721
s4           177.064    161.476   1.097 0.273456
s5           751.279    171.902   4.370 1.56e-05 ***
s6            67.625     65.984   1.025 0.305998
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.15 on 431 degrees of freedom
Multiple R-squared:  0.5177,    Adjusted R-squared:  0.5066
F-statistic: 46.27 on 10 and 431 DF,  p-value: < 2.2e-16
```

The residuals tell how well does the model fit the data. In another words, the residuals are the vertical distances between data points and the regression line. The “Estimate” column list the coefficients (slope) of each attribute.

Now, let’s evaluate the linear model by performing prediction on the test set.

```
> test$pred <- predict(lr, test)
> rmse(test$diabetes, test$pred)
```

As we can see, the root mean squared error of the prediction is 57.9175.