Today I am going to cover how to define statistical models using R formula.

We are very good at math formula because we have learned them since elementary school. For example

There are several key features in the **math formula**:

* **=** means is equal to
* The variable on the left side of **=** is the output/target
* The variable(s) on the right side of **=** is the inputs/predictors/features
* \_\_\*\_\_ means multiplication in math formula
* **+** means addition in math formula

Let’s load the data set into memory:

library(readxl)  
StudentsPerformance <- read\_excel("C:/Users/yliu3/OneDrive - Maryville University/Online DSCI502 R Programming/DataSets/StudentsPerformance.xlsx")

We first look at the variable names.

colnames(StudentsPerformance)

We would like to build a simple linear model to forecast Math scores using Reading Scores. i.e. we want to have a mathematical formula

Where and are parameters to estimate. Unfortunately, R engine cannot understand the math formula above. We have to pass the **R formula** to the engine using the following command:

lm.result <- lm(MathScore ~ ReadingScore, data= StudentsPerformance)

Note here:

* We need to use that function **lm** which means linear model.
* The first argument is the **R formula** with ~. The **variables/column names of the data frame** are used in the R formula.
  + means “is model as a function of” or “depend on”.
  + The variable on the left side of the is the target/dependent variable
  + The variable(s) on the right side of the is the predictors/independent variables/features
  + The **R formula,** for the linear model is equivalent to the **math formula,**
* The 2nd argument specifying the data source using data = a data frame in memory.

We can see the major differences between the **R formula** and **math formula** are:

* The math formula uses “=” and R formula uses to separate the target and predictors
* The operations on the right hand side of formula are different. We will cover it in the later section.

The R engine generated a special object, lm.result, called **list** in R memory. We can look at the structure of it by running the following command:

str(lm.result)

This list object has 12 columns. Each column can have a **different length or different data types,** unlike the data frame. For example, the coefficients column has two numerical values, qr column is a list. Therefore, the list is a very flexible data structure used to hold the data.

To access the data in the list, we use similar syntax as data frame since data frame is a special list.

select columns using $ operator  
lm.result$coefficients

select columns using double brackets, [[]]  
the result is the data type of the item  
lm.result[["coefficients"]]

list subsetting using single brackets, []  
the result is still a list  
lm.result["coefficients"]

After we call the lm function, the R engine estimate the coefficients of and .

We can call the summary function by providing the model result to extract them.

summary(lm.result)

We mainly focus on the **Estimate** column in the **Coefficients item**. The , called the Intercept, can be found at the intersection of Estimate column and (Intercept) row. It is 7.36. The , the coefficient of Reading score, can be found at the intersection of the Estimate column and Reading score row. It is 0.85.

There is another way to get the coefficients. We can take out the coefficients item by using the $ operator of the list object, lm.result.

summary(lm.result)$coefficients

The third way to get the coefficients is using the coef() function by providing the model result.

coef(lm.result)

Therefore, we have the following **math** formula

We may ask how good the model fit is. There is a quantity to summarize this info. It is the coefficient of determination, often denoted by (R squared). It is the percent of the variance in the target explained by the predictors/features. We can show the has the following properties:

* is between 0 and 1, inclusively.
* is 0 means that the model explains none of the variability of the target data around the mean.
* is 1 means that the model perfectly fit the target data.
* The larger the , the better the sample fit.

We can get the using the following R code:

summary(lm.result)

We can see the R-Sqaured is 0.6684365.

Another command can be used to determine .

summary(lm.result)$r.squared

## [1] 0.6684365

When more predictors are used, the model can explain a greater percent of variance by intuition. To compare different models that have a different number of predictors, we need to “remove” the number of predictors from the model. We can use the adjusted R-squared to do it. When more predictors are used in the model, the adjusted R-squared may increase or decrease. If the additional predictor improves the model performance, the adjusted R squared will increase, otherwise, it may decrease.

summary(lm.result)$adj.r.squared