Today, I am going to cover multiple linear regression.

Instead of using one predictor, we are asked to forecast math scores using a linear model with Reading Scores, Writing Score and Test Preparation Course.

Note here that there are two types of predictors:

- **Continuous** variables such as Reading Scores and Writing Score that have **infinite** many cases. We can use them **directly** in the R formula
- **Factor**/categorical variable such as Test Preparation Course that have only **finite** many cases: “none” and “completed”. We have to **convert it to a factor** before we use it in the **R formula**.

First, we look at the data types of all columns and find that TestPreparationCourse is a **character/string that cannot be used in R formula**.

```r
str(StudentsPerformance)
```

We need to **convert chr to factor before it can be used in the R formula**.

```r
#check the data type first; if it is not a factor yet, convert it
if(!is.factor(StudentsPerformance$TestPreparationCourse))
  #convert it to a factor
  StudentsPerformance$TestPreparationCourse <-         as.factor(StudentsPerformance$TestPreparationCourse)
#double check the data type
is.factor(StudentsPerformance$TestPreparationCourse)
```

Then, we can build the linear model using the following R formula

```r
lm.result2 <- lm(MathScore ~ ReadingScore + WritingScore + TestPreparationCourse, data= StudentsPerformance)
```

Note here, the “+” operator is used in the R formula.

- It only means “**include the variable**” in R formula.
- It does **not** mean regular addition operator in R formula.

The R formula

$$MathScore \sim ReadingScore + WritingScore + TestPreparationCourse$$

means the target, “Math Scores”, depends on “Reading scores”, “Writing scores” and “Test preparation course”.

It has corresponding math formula as follows

$$MathScore = \beta_0 + \beta_1 * ReadingScore + \beta_2 * WritingScore + \beta_3 TestPreparationCourse$$

Let’s figure out the corresponding coefficients by calling summary function in R.

```
summary(lm.result2 )$coefficients
```

We can get the coefficient of **continuous variables** such as Reading Scores and Writing Scores directly from the Estimate column as we did before. The coefficients are summarized in the following:

- The Intercept is 5.79
- The coefficient of Reading Score is 0.57
- The coefficient of Writing score is 0.29

But for the **factor/categorical variable**, there is a different story. When we look at the results above. We cannot find the variable name **TestPreparationCourse**. We can only find **TestPreparationCoursenone**. **TestPreparationCourse** is a factor with 2 levels/cases:

```
unique(StudentsPerformance$TestPreparationCourse)

## [1] none      completed
## Levels: completed none
```

The **TestPreparationCoursenone** concatenates the original variable name, **TestPreparationCoursen,** and one level: **none**. It means that if the TestPreparationCourse is none, then its coefficient is 1.48.

Therefore, we should **list all the finite cases with a different formula**:

$$MathScore = \begin{cases} 5.7903239 + 0.5700948 * ReadingScore + 0.2926499 * WritingScore; & \text{TestPreparationCourse is completed} \\ 5.7903239 + 0.5700948 * ReadingScore + 0.2926499 * WritingScore + 1.4794375; & \text{TestPreparationCourse is none} \end{cases}$$

Note that there is “no”" coefficient when the TestPreparationCourse is **completed**. Its effect is hidden in the y-intercept, which is 5.79. This case is the baseline. We need to understand the coefficient of factor in the following way:

When the TestPreparationCourse is none, the coefficient is 1.48, which means that the math score is 1.48 higher compared to the baseline (“completed”). This coefficient shows the relative effect instead of absolute effect.

Simplifying the math formula above, we obtain,

$$MathScore = \begin{cases} 5.7903239 + 0.5700948 * ReadingScore + 0.2926499 * WritingScore; & \text{TestPreparationCourse is completed} \\ 7.2697613 + 0.5700948 * ReadingScore + 0.2926499 * WritingScore; & \text{TestPreparationCourse is none} \end{cases}$$

In general, if a factor has n levels, we should have a piece wise function with n parts.

- The baseline contains one cases that is missing from the summary of the model results. It is hidden in the y-intercept.
- Other pieces are listed in the summary function which shows the relative effects with respective to the baseline.

Let’s look at the sample fit.

```
summary(lm.result2)$adj.r.squared

## [1] 0.6749004
```

It is 0.67 which is higher than the simple linear regression model. Therefore, the extra predictors helped improve the model performance.