Making predictions from your data in R using linear regression

Emahn Novid

10/18/2017

Linear regression is one of the earliest and most common techniques in practical data analysis. Regression is a relatively simple, yet powerful, mathematical process that we can use to predict unknown values. In other words, predict the future!

It's likely that you've already seen examples of regression. You can spot the use of regression analysis when you see a scatterplot overlayed with a straight line indicating a general up or down trend in the data.

It's easy to understand the value of predicting the future using data that we've collected. In this post we're going to explore one way of carrying out this procedure called **simple linear regression**.

What is simple linear regression?

Given a set of data with two variables (`x` and `y`), you can use simple linear regression to generate a straight line that's the best fit to your data. Using this line equation you can predict unknown values of `y` by plugging in values for `x` . `x` and `y` are the variables in an ab-line equation whose coefficients are determined through the mathematical procedure of linear regression.

Example: The relationship between a person's height and their body weight

The description above refers to variables `x` and `y` and that might sound cryptic, but it's actually representing something simple that we can relate to.

For example, imagine we take a group of people and record the heights and weight sof each person in two separate lists. * `x` = height measurements * `y` = weight measurements

There's an intuitive relationship here between `x` and `y` : As a person gets taller they also tend to weigh more.

Given this obvious relationship between height and weight we expect to see a similar pattern in our data. If we graph this data using weight as the y-axis and height as the x-axis we expect the points to align on an upward trend where the tallest people have the highest weight all grouped in the top right corner of the graph and the shortest people grouped in the bottom left indicating lower weight.

We can draw a straight line between the tallest and heaviest and the shortest and lightest but that would only be a rough eye-ball approximation.

Our attempt to formally analyze this raises an important question:

> How can we use this data to predict the weight of a non-existant ten foot tall human?

Simple linear regression gives us a way to answer this question.

Our goal is to predict the weight of someone who's ten feet tall. In other words, we're trying to predict an unknown weight variable using a known height variable (ten feet).

Simple linear regression looks at this problem in the form of an equation for a straight line:

`Y = a + b * X + e`

Where `a` is the y-intercept and `b` is the slope. `e` is the error term which accounts for the part of `Y` that the model can't explain - essentially our uncertainty about the future.

We won't get into the math here but it's important to know that this line is produced by finding values for `a` and `b` that result in the best fit line to our data.

After using simple linear regression to find an equation for this line, we can plug in our known `X` value (10 feet) and receive a prediction of weight in the `Y` value.

Performing simple linear regression using R

R provides a common syntax for linear regression and it abstracts away most of the math that's involved.

Using R, there are a number of steps we go through before creating the final line equation allowing us to make our predictions.

1. Gather our data

```
# Weight values
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
# Height values
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
```

2. Use the function `lm()` to get the coefficients for the regression line

The function `lm()` provides the coefficients `a` and `b` (mentioned above) used to complete our line equation.

The syntax for `lm()` is the following:

> lm(formula, data)

`formula` represents the two variables we're comparing `y` is a list of the values we want to predict and `x` is the values we're basing the predictions on. `data` is an optional parameter that can be a data frame that `formula` refers to.

```
# Apply the lm() function.
relation <- lm(y~x)

print(relation)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##    -38.4551       0.6746
```

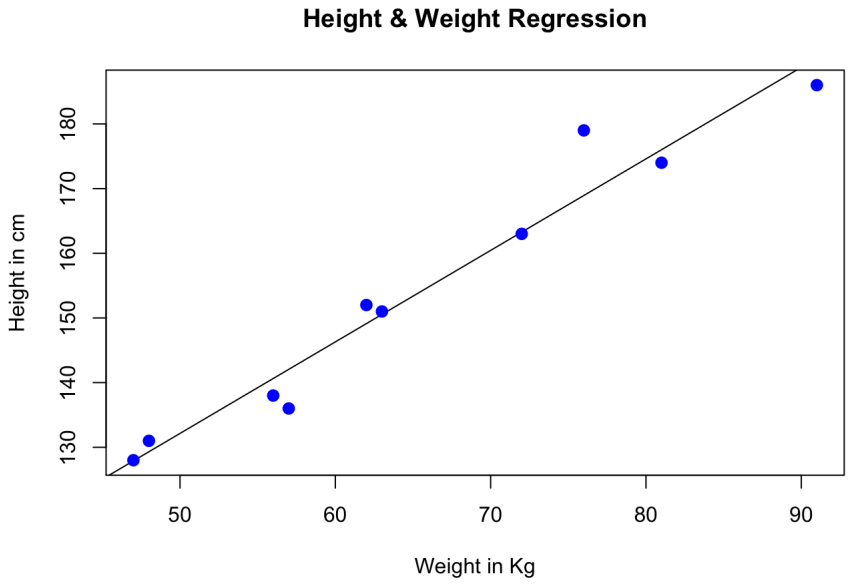
3. Use the function `predict()` to generate predictions using the coefficients

```
# Apply the lm() function.
relation <- lm(y~x)

# Find weight of a person with height = 170
a <- data.frame(x = 170)

result <- predict(relation, a)

print(result)
```

```
##        1
## 76.22869
```

Visualization

```
# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")
```



Conclusion

We've gone over an example of making predictions with two variables. The important thing to remember is this method can be extended to any number of variables. Performing regression on multiple variables and even non-linear equations opens up many areas of practical application - both in industry and academia. This relatively simple mathematical procedure is a fundamental tool of many disciplines like economics and the correlational studies done by academics and scientists. Linear regression provides us with important insights into the nature of our data and we can build on that knowledge to better understand and adapt to the world around us.

References

1. http://www.montefiore.ulg.ac.be/~kvansteen/GBIO0009-1/ac20092010/Class8/Using%20R%20for%20linear%20regression.pdf
2. http://r-statistics.co/Linear-Regression.html
3. http://www.r-tutor.com/elementary-statistics/simple-linear-regression/estimated-simple-regression-equation
4. https://www.tutorialspoint.com/r/r_linear_regression.htm

5. https://onlinecourses.science.psu.edu/stat501/node/251
6. https://books.google.com/books?id=MjNv6rGv8NIC&pg=PA1#v=onepage&q&f=false
7. http://www.cyclismo.org/tutorial/R/linearLeastSquares.html