CS100: Studio 8

Simple Linear Regression

October 30 and 31, 2019

Instructions

In this week’s studio, you will be learning how to create and interpret a simple linear regression model in R. You will also get some practice with data transformations.

Upon completion of all tasks, a TA will give you credit for today’s studio. If you do not manage to complete all the assigned work during the studio period, do not worry. You can continue to work on this assignment until the following Wednesday at 6 PM. Come by TA hours any time before then to show us your completed work and get credit for today’s studio.

Objectives

By the end of this studio, you will know:
- How to determine when two variables are related linearly, so that a simple linear regression model is applicable
- How to create a simple linear regression model in R
- How to interpret a simple linear regression model in R
- What Tukey’s ladder of transformations is
- How to transform data

Regression

Regression is a type of supervised learning used to model a relationship among numerical variables where it is hypothesized that one depends on the others. An example is modelling the value of a house using features like square footage, the number of bedrooms, and the age of the house.

Linear Regression

Linear regression is the most widely used tool for establishing a link between two (or more) variables. A linear regression represents the “line of best fit” between a response variable

\[ Y \]

and one or more explanatory variables

\[ X_1, \ldots, X_2, \ldots, \ldots, 1 \]
An explanatory variable (also called a predictor) is a variable that is (usually) independent of the others, meaning its value does not change with the values of the other variables. Explanatory variables are typically plotted on the $x$-axis. On the other hand, a response variable is a variable whose value may depend on the value of the explanatory variables. For example, if we were investigating the effect of parents’ heights on children’s heights, the parents’ heights would be the explanatory variable and the children’s heights would be the response variable.

In the case of just one explanatory variable $X$, you can create a scatter plot of pairs of $x_i$ and corresponding $y_i$ values. Generalizing from the data in such a scatter plot, a regression line is a model of how $Y$ depends (linearly) on $X$.

Given an unobserved values of $X$, such a model can be used to predict the corresponding value of $Y$. A linear regression is simply the line that best fits the data, meaning the line that minimizes the error between the predicted ($a.k.a.$ fitted) and the observed values.

Aside: Recall that the equation of a line is $Y = mX + b$, where $m$
is the slope of the line (which tells you how steep it is) and

\[ b \]

is the

\[ y \]

intercept (which tells you where the line intersects the

\[ y \]

axis). For any given value of

\[ x \]

, you can find the corresponding value of

\[ y \]

by calculating

\[ m \cdot x + b \]

.

There are two types of linear regression: simple and multiple. Simple linear regression is a method of modelling a quantitative response

\[ Y \]

on the basis of a single predictor

\[ X \]

. The model assumes that there is a linear relationship between and

\[ X \]

and

\[ Y \]

. We write this linear relationship as follows:

\[ Y = \beta_0 + \beta_1 X \]

Here

\[ \beta_0 \]

and

\[ \beta_1 \]

are the two coefficients of the linear model, representing the intercept and slope, respectively. Specifically,
describes how \( Y \) changes in response to changes in \( X \).

**Note:** Because regression generalizes to multiple variables, we replaced \( b \) by \( \beta_0 \), and \( m \) by \( \beta_1 \) in the familiar

\[
Y = mX + b
\]

formula.

Multiple linear regression is an extension of simple linear regression that handles multiple explanatory variables. In general, when there are \( n \) distinct predictors, then multiple linear regression takes the following form:

\[
Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n
\]

Here the \( \beta_i \), for \( i \in \{0, \ldots, n\} \), are the coefficients of the linear model, representing the gradients (i.e., slopes) in each direction, respectively. Specifically, \( \beta_i \) describes how \( Y \) changes in response to changes in variable \( X_i \), holding the values of all other variables fixed.
Linear Regression in R

To carry out a linear regression in R between explanatory variable $x$ and response variable $y$, you can use the command `lm(y ~ x)`.

The syntax $y \sim x$ tells R to model $y$ using $x$. The function `lm` tells R to create a linear model; that is, to use linear regression to predict $y$ from $x$.

To understand the output of `lm` in R, we’ll go through an example.

Here’s a scatterplot of weights in pounds (Weight) versus heights in inches (Height) for 250 Americans, plus the line of best fit.

![heightweight_scatter](image)

Figure 1: heightweight_scatter

Let’s fit a linear regression model to these data and analyze the output.

```r
fit <- lm(healthMeasures$Weight ~ healthMeasures$Height)
summary(fit)
```

For this example, our model is $\text{Weight} = -194.4861 + 5.2995(\text{Height})$.

Here’s how we interpret a few of the metrics in the R output (highlighted in red):

- The
-intercept of -194.4861 is the predicted weight of a person whose height is 0. Because no one is 0 inches tall, this

\[ y \]

-intercept is meaningless. (More generally, predictions are not usually valid outside the range of the

\[ x \]

values observed in the data, and become more and more precise the closer the queried

\[ x \]

value is to the sample mean

\[ \bar{x} \]

.)

- The slope of 5.2995 indicates that for every one inch increase in height, we can expect a 5.2995 pound increase in weight.

- The \( p \)

-value for the \( y \)
-intercept is less than $1.67 \times 10^{-6}$ (i.e., nearly 0). Likewise, the $p$ -value for the slope is less than $(2 \times 10^{-16})$ (i.e., essentially 0). When a $p$ -value is less than 0.05, we usually conclude that the predictor is significant (i.e., including this predictor in the model yields better results than not, where “not” means assuming a coefficient of 0 for that predictor).

- The $R^2$ value is 0.2631. This value of $R^2$ tells us how much of the variation (from the mean) in weight can be explained by variation (from the mean) in height. An $R^2$ value can range from 0 to 1; the closer it is to 1, the better the model explains the data. In this example, we might expect $R^2$ to be low, because there are many other factors besides height that affect a person’s weight.

**Part I: Randomly Generated Data**

Now it’s time for you to create your own linear models. Before working with real data, you are going to practice with randomly generated data.

**Part 1a**

The `runif(n, min, max)` command generates a vector of $n$ random numbers uniformly distributed between $min$ and $max$. (*Note: runif stands for “random uniform.”*)

Using `runif()`, generate two vectors, $x$ and $y$, where each has 50 numbers ranging between 0 and 100.

Create a scatterplot of $x$ and $y$ using the command `plot(x, y)`. Do you see any patterns?
Report the correlation between x and y using `cor(x, y)`. Is there a strong correlation between the two variables?

Regardless of your answer, run a linear regression using `my_model <- lm(y ~ x)`. Add a line of best fit to your scatterplot using `abline(my_model)`. Display the output of your model using `summary(my_model)`.

What is the slope of your model? Is it significant? Explain why or why not. (Hint: Check the \( p \) -value.) Look at the \( R^2 \) value. What does it tell you about how well your model explains the data?

Plot the residuals. *Hint:* To do so, use the following command: `plot(my_model, which = 1)`. The parameter `which = 1` tells R that you are interested in the residual plot. If you’d like to see some other diagnostic plots as well, you can omit this argument.

**Part 1b**

The uniformly random data you just generated did not exhibit a linear relationship. In this problem, you are going to randomly generate data according to a linear model:

\[ Y = a + bX + \epsilon \]

, where \( \epsilon \) denotes the noise in the model. (Test yourself: What are the coefficients of this linear model?)

Using `runif()` again, generate a vectors, `x` of 50 numbers ranging from, say, 0 and 100, and a second vector `e` (which stands for error) of 50 numbers ranging from, say, -10 to 10.

Choose values for `a` and `b`, and then combine `x` and `e` to produce a vector `y` as follows: `y <- a + b * x + e`.

Calculate the correlation coefficient between `y` and `x`. These variables should be very highly correlated!

Now, create a scatter plot displaying `y` as a function of `x`. Discuss with your partner what you observe.

Create a linear model to predict `y` from `x`. Comment on the model obtained. How do the estimated values of the slope and the intercept compare to their true values?
Add the regression line to your scatterplot using `abline`. You can define the `col` parameter to specify a color: `abline(my_model, col = "red")`. Additionally, plot the true linear model: i.e.,

\[ Y = a + bX \]

You can do so using `abline(a = a, b = b, col = "blue")`. \textit{Hint}: You can use the `legend()` command to add an appropriate legend to your plot as follows: `legend("topleft", legend = c("Best Fit", "True Model"), col = c("red", "blue"), lty = 1)`.

How do the two lines differ? How well does the best fit line fit the data? Compare the `r` and \( R^2 \) values to their counterparts in Part 1a. Lastly, as a final sanity check, plot the residuals. What do you observe?

\textbf{Part II: World Population Growth}

Not all relationships among data are linear. For example, the world population is growing exponentially: i.e., it is growing very quickly over time.

Load world population data into R as follows:

```r
world_pop <- read.csv("http://cs.brown.edu/courses/cs100/studios/data/6/worldpop.csv")
```

For convenience, rename the inconveniently named population variable, and filter `world_pop` to only include years since 1000.

Now plot “Population” vs. “Year”: i.e., create a scatter plot with “Year” on the x-axis and Population on the y-axis. What kind of trend do you observe? Although “Year” appears to be a significant predictor of “Population”, the relationship between the two variables is not linear. Still, it is possible we could transform one or both of these variables to make the relationship linear. Try plotting a few transformations of the data. What do you notice? How does population appear to have grown over time? What, then, is an appropriate transformation?

Many data relationships become linear after an appropriate transformation. The famous statistician, John Tukey, defined what he called a \textit{ladder of transformations}, defined by a parameter \( \lambda \)

over which one could search for a suitable transformation that would transform an arbitrary relationship into a linear one, simply by finding the most appropriate \( \lambda \).
The transformation Tukey applied was a power transformation. If the lambda you are currently considering is negative, and you are transforming some variable $v$ then the transformation is $-v^\lambda$.

If the lambda you are considering is 0, the transformation is $\log v$.

Finally, if the lambda you are considering is positive, the transformation is $v^\lambda$.

You can read more about Tukey’s ladder here.

For the remainder of this studio, you will search for the best positive value of $\lambda$ you can find under which a power transformation creates a linear relationship in population data. As you know, there are multiple diagnostics that evaluate the quality of a linear relationship, but for the purpose of this studio $R^2$ will suffice.

**Hint:** In this part of studio, you will repeat the same steps multiple times for different values of $\lambda$.

Therefore, you will probably find it worthwhile to encapsulate these steps in a function or a loop.

**Initialization:** Create a vector of a few values of $\lambda$ to try. **Hint:** Use the `c` function initially; you can add any additional values of $\lambda$ that turn out to be of interest later using the function `append`.

Next, initialize an empty vector in which to hold the $R^2$.
values of the linear models that correspond to the power transformations of each value of
\[ \lambda \]

. Do so like this: \( R2 \leftarrow c() \), and then use the \texttt{append} function to append values to the end of this vector.

**Loop:** For each such \( \lambda \) value, transform the \( X \) variable, “Year”. For example, to transform “Year” by raising it to your first \( \lambda \) power, you might write code like this: \texttt{world_pop$lambda1 <- world_pop$Year ** lambdas[1]}. Once you’ve transformed the “Year” variable, plot it against “Population” to check whether it looks linear. Create a linear model to summarize the relationship.

For each value of \( \lambda \) that you try, save the \( R^2 \) value of the linear model. You can access this value as follows: \texttt{summary(my_model)$r.squared}. Your goal is to keep trying new values of \( \lambda \) until you achieve an \( R^2 \) value above .9. To guide your search, you should plot your \( \lambda \) s against the corresponding \( R^2 \) values. Use this plot to help identify new values of \( \lambda \) to try. (*Hint:* You must ensure that your \( \lambda \) and \( R^2 \) vectors are of the same length before you can plot them against one another.)
End of Studio

When you are done please call over a TA to review your work, and check you off for this studio. If you do not finish within the two hour studio period, remember to come to TA office hours to get checked off.