CSI 2300: Lecture 19 – Simple Linear Regression

Outline

In this lecture we'll discuss:

- Recap simple linear regression with `lm()`
- What does linear regression actually do?
- Investigate how well the line fits the data

Remember Linear Regression

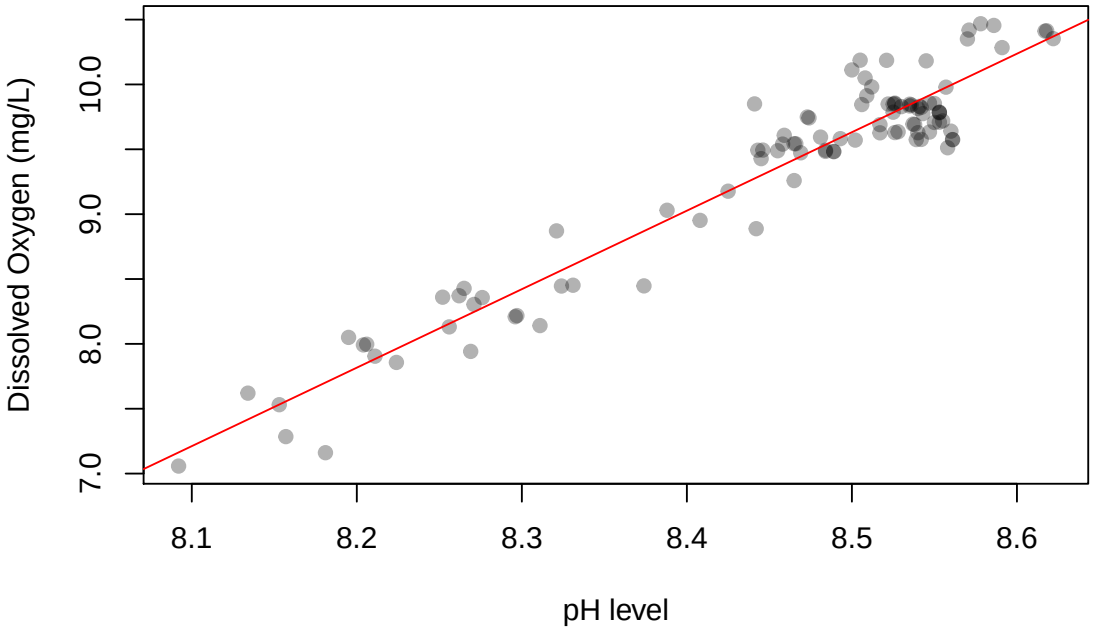
Let's review the mechanics of linear regression quickly. We'll return to the Eagle Mountain Lake (EML) dataset we used when we first discussed linear regression.

```
# load in the dataset...
suppressMessages(library(mowateR))
data(eml)

# let's use a small amount of data, so we can see the examples more clearly
N <- 100
eml_small <- eml[1:N,]

# plot with some transparency (alpha = 0.1) to better see the examples that
# overlap
plot(eml_small$DO ~ eml_small$pH, col=rgb(0, 0, 0, 0.3), pch=19,
     xlab="pH level", ylab="Dissolved Oxygen (mg/L)", main="Eagle Mountain Lake")

# use lm() to find a linear model
model <- lm(DO ~ pH, data=eml_small)
abline(model, col="red")
```



Note our use of the R formula notation: `DO ~ pH`. The right side (pH level) is the **independent variable**, and the left side (dissolved oxygen) is the **dependent** variable. The implication is that pH affects dissolved oxygen.

Recall that we can investigate the coefficients of the model that `lm()` returns:

```
model
```

```
##
## Call:
## lm(formula = DO ~ pH, data = eml_small)
##
## Coefficients:
## (Intercept)           pH
##     -41.798        6.051
```

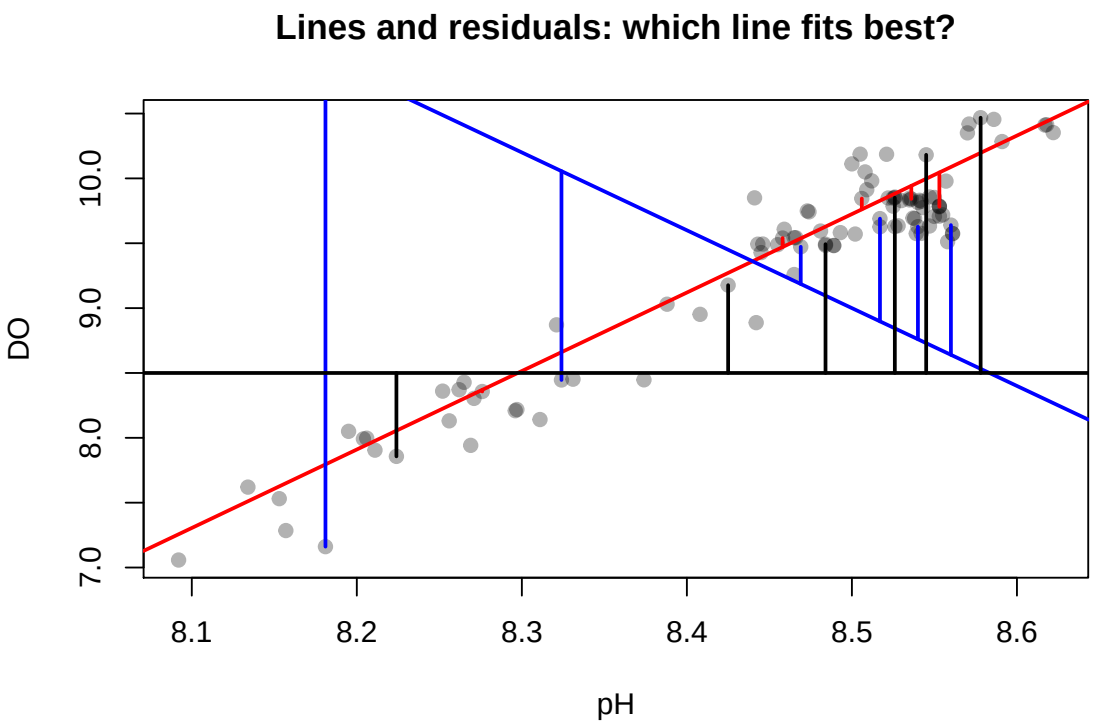
So we have an intercept (called "(Intercept)") and a slope (called "pH", the same name as our independent variable). Recall what those two quantities mean.

- slope: when pH goes up by 1, DO goes up by 6.05.
- intercept: when pH is 0, DO is predicted to be -41.8 (which is absurd).

What Is Linear Regression Doing?

Linear regression fits a line to some examples. But how? Why does it choose one line over another? This has to do with how we define the **fit function** or the **error function**, which is based on "sum of squared residuals" or just "sum of squares".

- Consider **one** observation pair (x_i, y_i), where x is the independent variable and y is the dependent variable.
  - i here is the index of the observation (in our case, ranging from 1 to 100 of our small data frame).
  - For our example, x_i is a particular pH value and y_i is its associated dissolved oxygen level.
- The prediction that the model makes is $\hat{y}_i = x_i \cdot a + b$, where a is the slope and b is the intercept.
- The **difference between the prediction and the observation** (y_i) is called the **residual** $(y_i - \hat{y}_i)$.
  - You can visualize this as the vertical distance between the line and the example.



The Goal: Minimize the Residuals (for all examples).

So we define the **sum-of-squared residuals**:

$$SSR = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - x_i \cdot a + b)^2$$

When we run `lm()`, it is choosing the slope and intercept (values of a and b) that minimize SSR.

The SSR is the sum of the square of all of these residual values. So it's always positive (because the squares are positive).

There's an easy way to get the residuals in R: `residuals()`. Let's calculate them by hand anyway.

```
# use the residuals() built-in function
r_residuals <- residuals(model)

# compute the residuals manually using a loop
hand_residuals <- 1:nrow(eml_small)
for (i in 1:length(hand_residuals)) {
    prediction <-  model$coefficients[1] + model$coefficients[2] * eml_small[i,]$pH
    hand_residuals[i] <- eml_small[i,]$DO - prediction
```

```
}

# square each of the residuals
r_squared_residuals <- r_residuals ^ 2
```

Explore Residuals with a Shiny App!

Go look at the Shiny app, modeling_19_shiny_regression.R!

How well does the line fit?

Getting a line to fit is easy. But is that line any good? Does it fit the data well? We can answer this question in a few ways:

- Look at it – how **close** are the examples to the line?
- Measure it – how **small** is the SSR?
- Test it – are the coefficients **significant**?

Giving Context to SSR

The sum of squared residuals (SSR) is not very informative by **itself**. It's not normalized or easily comparable to anything. Here are some good ways to convert the number to something more understandable:

- Take the average (mean), and then the square root of that mean. This gives the **standard deviation of the residuals**, which doesn't increase or decrease just because you add or remove examples from the data.
- **Divide the SSR by the SST**, or "total sum of squares". The SST can be thought of as the residuals for a horizontal line at the average value of the dependent variable (or "y" value). The SST is defined as

$$SST = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

  If the SSR is about the same as the SST, then $SSR/SST \approx 1$, and this implies the line doesn't fit very well. If the SSR is much smaller than the SST, then $SSR/SST \approx 0$, and this implies that the line fits the data very well (it "explains" a lot of the variation of the dependent variable).

- Compute the **coefficient of determination**, or R^2 value, as

$$R^2 = 1 - \frac{SSR}{SST}$$

  Note that this is just 1 minus the value we just discussed (SSR/SST). Thus when the line is a good fit, $SSR/SST \approx 0$, and $R^2 = 1 - SSR/SST \approx 1$. Conversely, when the line is not much better than just predicting the average of the dependent variable, then $R^2 \approx 0$.

Coefficient Significance (p values)

Consider this thought experiment: if we fit a line to two points, we can fit it *exactly*. Then $R^2 = 1$, hooray! Despite that, two points does **not** provide much evidence that there is a strong underlying linear relationship between the independent and dependent variables. But if there were 100 points, or 10 000 points that gave a $R^2 = 1$, that would be strong evidence. But we cannot judge that based on R^2!

As mentioned in a previous lecture, many software packages (like R) will give us not only the coefficients (slope and intercept) of the best-fit line, but also the **levels of significance** of those coefficients.

The level of significance can be thought of as the **probability that the coefficient's value is non-zero just due to random chance** – where "random" here refers to the data we observed for fitting the line.

```
summary(model)
```

```
##
## Call:
## lm(formula = DO ~ pH, data = eml_small)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5406 -0.1611 -0.0043  0.1540  0.5752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -41.7978     1.5147  -27.59   <2e-16 ***
## pH            6.0505     0.1792   33.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2303 on 98 degrees of freedom
## Multiple R-squared:  0.9209, Adjusted R-squared:   0.92
## F-statistic:  1140 on 1 and 98 DF,  p-value: < 2.2e-16
```

If you look at the line labeled "`## Signif. codes:  0 '***' ...`", those numbers are considered significance levels (sometimes known as p values). The smaller they are, the more significant the result (lower probability that the coefficient is actually zero). As we see in this model, the coefficients `(Intercept)` and `pH` both have labels `Pr(>|t|)` that indicate `<2e-16 ***`, or very significant. So that **is strong evidence** that the coefficients are non-zero, and that a line is explanatory for predicting DO from pH.

We're just fitting a line. What does "random" mean here?

Here's a way to think about what "randomness" and "probability" means in this context. Suppose you have a very large "pool" of data on the thing you're modeling. But at any given time you're only looking at a handful of observations (say, N of them) from that pool. You run a regression on those N observations and get the coefficients of a fitted line. What if you randomly selected **another** set of N observations from the same original pool, and fit another line to those observations? Is it likely that, **just by chance (due to the observations that happened to be selected)**, you would get significantly different coefficients? The smaller N is, the more likely it is that the coefficients would differ significantly across different sets of observations. So the randomness is (again) due to the data we happen to observe, not the method of fitting the line.