Introduction to linear regression

Your reproducible lab report: Before you get started, download the R Markdown template for this lab. Remember all of your code and answers go in this document:

```
download.file("https://dyurovsky.github.io/85309/post/rmd/lab9.Rmd",
              destfile = "lab9.Rmd")
```

The movie *Moneyball* (http://en.wikipedia.org/wiki/Moneyball_(film)) focuses on the "quest for the secret of success in baseball." It follows a low-budget team, the Oakland Athletics, who believed that underused statistics, such as a player's ability to get on base, better predict the ability to score runs than typical statistics like home runs, RBIs (runs batted in), and batting average. Obtaining players who excelled in these underused statistics turned out to be much more affordable for the team.

In this lab we'll be looking at data from all 30 Major League Baseball teams and examining the linear relationship between runs scored in a season and a number of other player statistics. Our aim will be to summarize these relationships both graphically and numerically in order to find which variable, if any, helps us best predict a team's runs scored in a season.

Getting started

As usual, we're going to load the `tidyverse` package for data manipulation. We'll also be reading in a dataset to work with just like we usually do. This time, the data won't just be a comma separated value file so we'll use the `load` function to load an `RData` file that has both the data and some functions you will need.

```
library(tidyverse)

load(url("https://dyurovsky.github.io/85309/data/lab9/mlb11.RData"))
```

The data

Let’s load up the data for the 2011 season.

In addition to runs scored, there are seven traditionally-used variables (https://en.wikipedia.org/wiki/Baseball_statistics#Commonly_used_statistics) in the data set: at-bats, hits, home runs, batting average, strikeouts, stolen bases, and wins. There are also three newer variables: on-base percentage, slugging percentage, and on-base plus slugging. For the first portion of the analysis we’ll consider the seven traditional variables. At the end of the lab, you’ll work with the three newer variables on your own.

Exercise 1

What type of plot would you use to display the relationship between `runs` and one of the other numerical variables? Plot this relationship using the variable `at_bats` as the predictor. Does the relationship look linear? If you knew a team’s `at_bats` , would you be comfortable using a linear model to predict the number of runs?

If the relationship looks linear, we can quantify the strength of the relationship with the correlation coefficient.

```
mlb11 %>%
  summarise(cor = cor(runs, at_bats)) %>%
  pull()
```

Sum of squared residuals

> In this section you will use an interactive function to investigate what we mean by “sum of squared residuals”. You will need to run this function in your console, not in your markdown document. Running the function also requires that the `mlb11` dataset is loaded in your environment.

Think back to the way that we described the distribution of a single variable. Recall that we discussed characteristics such as center, spread, and shape. It’s also useful to be able to describe the relationship of two numerical variables, such as `runs` and `at_bats` above.

Exercise 2

Looking at your plot from the previous exercise, describe the relationship between these two variables. Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations.

Just as we used the mean and standard deviation to summarize a single variable, we can summarize the relationship between these two variables by finding the line that best follows their association. Use the following interactive function to select the line that you think does the best job of going through the cloud of points.

NOTE: Interactive commands don't always play nicely with RMarkdown. You may need to run in this in your R Console

```
plot_ss(mlb11, x = at_bats, y = runs)
```

After running this command, you'll be prompted to click two points on the plot to define a line. Once you've done that, the line you specified will be shown in black and the residuals in blue. Note that there are 30 residuals, one for each of the 30 observations. Recall that the residuals are the difference between the observed values and the values predicted by the line:

$$e_i = y_i - \hat{y}_i$$

The most common way to do linear regression is to select the line that minimizes the sum of squared residuals. To visualize the squared residuals, you can rerun the plot command and add the argument `showSquares = TRUE`.

```
plot_ss(mlb11, x = at_bats, y = runs, showSquares = TRUE)
```

Note that the output from the `plot_ss` function provides you with the slope and intercept of your line as well as the sum of squares.

Exercise 3

Using `plot_ss`, choose a line that does a good job of minimizing the sum of squares. Run the function several times. What was the smallest sum of squares that you got? How does it compare to your neighbors?

The linear model

It is rather cumbersome to try to get the correct least squares line, i.e. the line that minimizes the sum of squared residuals, through trial and error. Instead we can use the `lm` function in R to fit the linear model (a.k.a. regression line).

```
m1 <- lm(runs ~ at_bats, data = mlb11)
```

The first argument in the function `lm` is a formula that takes the form `y ~ x`. Here it can be read that we want to make a linear model of `runs` as a function of `at_bats`. The second argument specifies that R should look in the `mlb11` tibble to find the `runs` and `at_bats` variables.

The output of `lm` is an object that contains all of the information we need about the linear model that was just fit. We can access this information using the summary function.

```
summary(m1)
```

Let’s consider this output piece by piece. First, the formula used to describe the model is shown at the top. After the formula you find the five-number summary of the residuals. The “Coefficients” table shown next is key; its first column displays the linear model’s y-intercept and the coefficient of `at_bats`. With this table, we can write down the least squares regression line for the linear model:

$$\hat{y} = -2789.2429 + 0.6305 \times \mathit{at\_bats}$$

One last piece of information we will discuss from the summary output is the Multiple R-squared, or more simply, R^2. The R^2 value represents the proportion of variability in the response variable that is explained by the explanatory variable. For this model, 37.3% of the variability in runs is explained by at-bats.

Exercise 4

Fit a new model that uses `homeruns` to predict `runs`. Using the estimates from the R output, write the equation of the regression line. What does the slope tell us in the context of the relationship between success of a team and its home runs?

Prediction and prediction errors

Let’s create a scatterplot with the least squares line for `m1` laid on top.

```
ggplot(mlb11, aes(x = at_bats, y = runs)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

Here we are literally adding a layer on top of our plot. `geom_smooth` creates the line by fitting a linear model. It can also show us the standard error `se` associated with our line, but we’ll suppress that for now.

This line can be used to predict y at any value of x. When predictions are made for values of x that are beyond the range of the observed data, it is referred to as *extrapolation* and is not usually recommended. However, predictions made within the range of the data are more reliable. They’re also used to compute the residuals.

Exercise 5

If a team manager saw the least squares regression line and not the actual data, how many runs would he or she predict for a team with 5,579 at-bats? Is this an overestimate or an underestimate, and by how much? In other words, what is the residual for this prediction?

Model diagnostics

To assess whether the linear model is reliable, we need to check for (1) linearity, (2) nearly normal residuals, and (3) constant variability.

Linearity: You already checked if the relationship between runs and at-bats is linear using a scatterplot. We should also verify this condition with a plot of the residuals vs. fitted (predicted) values.

```
m1_residuals <- tibble(x = nrow(mlb11),
                       fitted = fitted(m1),
                       resid = residuals(m1))

ggplot(m1_residuals, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```

Notice here that our model object `m1` can also serve as a data set because stored within it are the fitted values ($\hat{y}$) and the residuals. Also note that we're getting fancy with the code here. After creating the scatterplot on the first layer (first line of code), we overlay a horizontal dashed line at $y = 0$ (to help us check whether residuals are distributed around 0), and we also adjust the axis labels to be more informative.

Exercise 6

Is there any apparent pattern in the residuals plot? What does this indicateabout the linearity of the relationship between runs and at-bats?

Nearly normal residuals: To check this condition, we can look at a histogram

```
ggplot(m1_residuals, aes(x = resid)) +
  geom_histogram(binwidth = 25) +
  xlab("Residuals")
```

Exercise 7

Based on the histogram, does the nearly normal residuals condition appear to be met?

Constant variability:

Exercise 8

Based on the residuals vs. fitted plot, does the constant variability condition appear to be met?

More Practice

Exercise 9

Choose another one of the seven traditional variables from `mlb11` besides `at_bats` that you think might be a good predictor of `runs`. Produce a scatterplot of the two variables and fit a linear model. At a glance, does there seem to be a linear relationship?

Exercise 10 How does this relationship compare to the relationship between `runs` and `at_bats` ? Use the R^2 values from the two model summaries to compare. Does your variable seem to predict `runs` better than `at_bats` ? How can you tell?

Exercise 11 Now that you can summarize the linear relationship between two variables, investigate the relationships between `runs` and each of the other five traditional variables. Which variable best predicts `runs` ? Support your conclusion using the graphical and numerical methods we've discussed (for the sake of conciseness, only include output for the best variable, not all five).

Exercise 12 Now examine the three newer variables. These are the statistics used by the central character (https://en.wikipedia.org/wiki/Paul_DePodesta) in *Moneyball* to predict a team's success. In general, are they more or less effective at predicting runs that the old variables? Explain using appropriate graphical and numerical evidence. Of all ten variables we've analyzed, which seems to be the best predictor of `runs` ? Using the limited (or not so limited) information you know about these baseball statistics, does your result make sense?

Exercise 13 Check the model diagnostics for the regression model with the variable you decided was the best predictor for runs.