COMP 310
Project 1 - Data Visualization

Fall 2018

Abstract

In this project you'll be extracting and visualizing data from the Lahman database. Your main learning objective is to learn how to interface a database with a high-level, general purpose programming language and understand the usage of databases from the application developer's standpoint.

Due in class on Tuesday 10/9

Measuring Performance

There is no one way to measure batting and pitching performance in baseball. Instead, several different statistics are used, each with their own merit. We've seen and explored some of these in class already. Now we'll take a wider, more engaged look at these statistics by looking at what they tell us about different slices of baseball history.

Batting

The classic batting performance statistic is the batting average; it is the ratio of hits to at bats. The results give us a percentage of time that a batter gets a hit. It's simple to compute and easy to understand.

$$\text{BAvg} = \frac{H}{AB} \tag{1}$$

What batting average fails to account for are the other ways a batter can get on base. An at bat is not the same as a plate appearance. A more nuanced look at offensive performance comes from the batter's on base percentage which awards credit for other ways a hitter might get on base and thereby make an offensive contribution: walks and getting hit by a pitch. The positive offensive outcomes weighted against total at bats, walks, hit by pitch occurrences, and sacrifice flies to get another percentage scale ratio.

$$\text{OBP} = \frac{H + BB + HBP}{AB + BB + HBP + SF} \tag{2}$$

Both batting average and OBP treat all hits equally. The slugging rate (SLG) for a hitter gives more credit to hits that result in more bases. It's results range from 4 down to 0 where a 4 is achieved by hitting a homerun on every single at bat and a zero by never getting a hit when at bat. We can interpret the result as the average number of bases achieved per at bat.

$$\text{SLG} = \frac{1B + 2 \times 2B + 3 \times 3B + 4 \times HR}{AB} \tag{3}$$

The batting average, on base percentage, and slugging rate each say something different about a player's offensive contributions to the game. What isn't immediately clear, and what we want to explore, is the extent to which they vary in practice from player to player and in particular for an individual player. Is it possible to rate highly with one statistics but less so with another? If so, by how much? Looking at different players in different years will allow us to explore these questions empirically.

Pitching

A standard measure of pitching quality is the ERA, or earned runs average. It approximates the average runs a pitcher gives up in 9 innings of baseball.

$$ERA = 9\frac{ER}{IP} = 3\frac{ER}{IPOuts} \tag{4}$$

Thankfully, a pitchers ERA is already tracked by the Lahman database. We don't need to compute it and in that way its very easy to use as a metric of pitching quality. The problem with ERA is that runs are a function of pitching, fielding, and the physical environment. Poor fielding allows more hits which in turn allows for more runs. Conversely, excellent fielding and save "bad" pitching and inflate the ERA of that pitcher. The ERA can also be affected by the parks in which the pitcher is playing. Balls travel differently in different conditions.

The ERA+, or adjusted ERA, attempts to control for the environment by incorporating the league average ERA (lgERA) and the pitching park factor, which attempts to account for differences in ballparks. An ERA+ of 100 means that the pitcher had average performance for the league and the parks played in where an ERA+ above or below 100 means the pitcher performed above or below average respectively. The pitching park factor (PPF) is tracked on a team by team basis and found in the Lahman database. League averages will need to be computed.

$$ERA+ = 100\left(2 - \frac{ERA}{lgERA} \times \frac{1}{PPF}\right) \tag{5}$$

While ERA+ does account for the physical batting environment, it doesn't necessarily account for the fielders backing up the pitcher. Towards this end the Fielding Independent Pitching (FIP) statistic looks only at the things the pitcher can more or less control for: allowed homeruns, walks, hit by pitches, and strike outs. A pitcher's FIP is typically adjusted with a , cFIP, so that the result is scaled to a value that is comparable to with ERA. Constants are determined by the league averages for the year in question and will need to be computed for each league and each year in question; they are not in the database.

$$\begin{aligned} FIP &= \tfrac{13\times HR + 3\times(BB+HBP) - 2\times K}{IP} + cFIP \\ cFIP &= lgERA - \tfrac{13\times lgHR + 3\times(lgBB+lgHBP) - 2\times lgK}{lgIP} \end{aligned} \tag{6}$$

A pitcher's ERA, ERA+, and FIP score says something about the quality of their pitching but it's not clear if the computational cost of ERA+ and FIP gives us a better snapshot of pitching quality. What is also not clear is whether or not we should look more closely at the park adjusted ERA+ or the fielding adjusted FIP. Once again, an examination of these statistics across historical baseball data should help us shed some light on these and other questions.

Visualization

Our goal is to visualize the 6 statistics above through historical baseball data so that we, and others, might be able to assess the merits of each statistics both on their own and comparatively. Our goal is not to draw conclusions but to provide concrete, empirical data from which we can gain insight and begin discussions. Multiple visualizations of each statistic, each presenting multiple dimensions of information, enables comparison and critique and meets this end.

Comparing Statistics with a Trio of 2D Point Plots

Pick two statistics for batting, ERA and ERA+ for example. Now pick a player in a given season. Imagine that player as an (x, y) point in a 2D space where the x dimension is ERA and y is ERA+, or vice versa. Plotting multiple players in this way lets us not only compare players but compare ERA to ERA+. Given that our goal is to compare all three stats and not just two, we should not produce a single plot but three, one for each possible pair of statistics. Each plot should be labeled and titled and stand on its own. By placing the plots side by side we can easily compare data points within a single plot to look for relationships between two statistics and from one plot to the next to look for relationships between the three statistics.

Comparing Statistics with a Trio of Tables

Never discount the power of a simple table for presenting data. By that token, multiple tables presented together in a single graphic invite deeper investigations. Imagine taking a set of pitchers, computing the ERA, ERA+, and FIP. Now, in a table, list the pitchers and their ERAs, sorted from highest to lowest, next to a similar table for ERA+, next to a similar table for FIP. With this three table graphic we can look at how each individual statistic ranks pitchers and compare that to how the other statistics rank them. It has the added benefit of making it easy to attach actual names to the statistics as well.

Comparing Statistics with a Trio of Histograms

A histogram for a given statistic lets us git a general feeling for how often players reached a particular value for a given statistic. They visualize the distribution of values across a particular data set. By placing a histograms of Batting Average, OBP, and SLG side by side we can not only explore the distribution of individual statistics but compare those distributions to other statistics.

Slicing the Data

We now know that we want to look at specific statistics using specific visualizations, but we don't know what set or sets of data we want visualize. To make life easier we'll set some ground rules.

- Our data points will always be one player in one season. Each statistic should be computed based on a single season of baseball for a single player. We will not look at aggregates that include multiple player nor will we compute a statistic for a single player over multiple seasons.
- Data sets should be drawn logically, i.e. a specific set of players, one or more teams, one or more divisions, one or both leagues. If we're interested in looking at a range of years we should redraw our logical group for each year. For example, we could choose to look at all the cardinals from 1945 until 2016 or maybe all of MLB in the year 1975.
- Data sets should only include players with a number of plate appearances, or innings pitched, that result in a meaningful statistic. No one hit wonders. This most definitely means filtering out anyone with zero plate appearances or innings pitched. It also means leaving out anyone below some threshold. Absent a reasoned cutoff we'll look at the upper 25th percentile and above for plate appearances/innings pitched.

Within these bounds, you can choose any kind of slicing pattern you want so long as you can generalize and vary it into a pattern. For example, if you're thinking of exploring Cardinals players only but over the time frame of 1945-1980, then a generalization of this data set could be any single team over any range of seasons in the 1945 to 2016 range. Our goal is clearly identify a whole family of data sets to choose from as opposed to one fixed data set.

Putting It All Together

Whatever sample of data points you pull is likely to be too big for a table and would produce a cluttered point plot graphic. We'll use the following strategy for managing this problem. Develop histograms for a large set of data. Then sample from that an interesting subset, like the top n or top and bottom n to display on both the table and the point plots. Because the top n in batting average might not be the top n in OBP nor the top n in SLG, we will want to union these subsets together and put them all in the point plot. This means the data points graphed on the plot are the same points shown in the tables and that they are all drawn from the distribution displayed in the histogram. For example, you might display the distributions of the ERA, ERA+, and FIP of all the Cardinals' pitchers from 1980 to 2000 in your histogram, then list the top 25 for each in their respective tables, and finally combine those lists into a set of 25-75 players that you plot in each of the three 2D point plots.

You Tasks

We can look at our project as implementing a three stage pipeline: extract the needed data form the database, prepare the data for visualization, and produce the visualization. Your code should be decomposed along these lines. Towards that end you have the following tasks in front of you.

1. *Develop python code to compute the league averages needed to compute ERA+ and the FIP constant for any year in the 1945 to 2016 range and any MLB league.* Your code must include one or more functions to compute the SQL query string needed to pull the data necessary for a given year and league. If your query itself does not compute the averages, then you should write functions to do so given the results of your queries.

2. *Develop python code to build the queries to pull the data needed for a trio graphic.* Once again, your goal is to design functions that construct SQL query strings. The arguments to these functions are determined by your data slicing pattern. For example, you might have a function that builds the query string to pull the batting data for a given team, starting at a given year, and ending at a given year. Given that each of these graphics is drawn from a single data set, you can choose to write a single query for that data set and use python for further slicing and dicing of the data or you can choose to write multiple simpler queries for each graphic. If you choose the later, be sure your data set is consistent with the design given in *Putting It All Together*.

3. *Develop python code to prepare the results of a query for visualization.* This code should take the results of one of your queries and wrangle data into the form needed for the intended graphic. It does not produce the graphic itself.

4. *Develop python code to prepare a trio graphic.* This code is given the data its going to visualize in a form that is ideal for the visualization at hand so that you can focus on tweaking the parameters of the graphic itself rather than wrangling the data.

You may find that you need additional supporting code to connect the stages of the pipeline. That's fine. Just develop each stage independently as discussed above.

Grading

Grading is based on meeting the following incremental benchmarks, i.e. you must complete the benchmarks in this order.

- (D) Be able to compute FIP constants for any year from 1945 to 2016 and for any league. Check your results against this site: `https://www.fangraphs.com/guts.aspx`.
- (C) Be able to produce a histogram trio graphic for any sample of your chosen sampling pattern.
- (B) Be able to produce the table trio for your sampling pattern.
- (A) Be able to produce the point plot trio for your sampling pattern.

Your code should produce the desired result without error. Failure to do so will reduce your grade. Partial credit will be rewarded if you can produce queries and process data but not produce the graphic, i.e. if you can get through some but not all of the process pipeline. For example, a C level project might achieve a C+ or possible B- if it demonstrably completes two of the three pipeline states for the B level project.

Your code should also be well organized and commented such that it is not a pain to read. Messy, difficult to follow code will result in a reduction of your grade. Such reductions will not kick you out of the range specified by your benchmark. For example, a B project with messy code can end up with a B- even if you've completed two of the three pipeline steps for the A level project.