Goal

The final goal of this ETL project will be a dataset that gives us a clear picture of pitchers in the MLB that characterizes their pitching style and success rate in the 2018 season. To do this we need to grab data from multiple tables that are related and as such the best option for our analysis would be a relational database.

Extraction

Data was pulled from a kaggle repository that contained five different csv’s. These had several commonalities between them that could be used as foreign keys.

Transformation

1.     Joining player\_names table with atbats table on player\_id. Gives a clearer picture on which players are right vs. left handed.

2.     Lots of columns stand to be dropped from pitches table (and others) because they lack any explanatory value. For example, in pitches table there are various pitch parameters that aren’t characterized in any way (ax, ay, vz, break angle, etc.)t

3.     Create one single pitchers table with information on which hand they throw with, their most frequent pitch and most frequent pitch outcomes

4.  Many records had empty fields. Drop all rows that are incomplete.

5. Trimmed records from games that were before 2018.

6. Determine if certain pitchers perform better against right or left handed batters using     using all three tables.

Load

1. We will create a relational database in Postgres and create tables by importing the above referenced CSVs as well as keys that connect the various tables. We will use this database to write a series of queries regarding parts of the baseball data that we find interesting/unique per some of the guidelines outlined above as well as ad-hoc requests we are inspired by during the development process.

Transformation Process

Four relevant csv’s were loaded into pandas dataframes, those being: *atbats*, *player\_names*, and *pitches*, and *games*.

The *atbats* csv contained a unique id (atbat\_id) and columns describing the outcome of each at bat. Relevant information found here would be *p\_throws* (L or R to denote the pitcher’s dominant hand)*, pitcher\_id, game\_id*, and *stand* (L or R to denote the batter’s dominant hand). The *games* csv’s only desirable information was the date of each game which was associated with a unique *game\_id* which could tie it to the *atbats* dataframe.

The *player\_names* csv had three columns: *player\_id, first\_name,* and *last\_name* with player\_id being its unique primary key.

The *pitches* csv contained various columns that characterized each pitch such as indicators of speed, spin, and position. Each row had a non-unique *atbat\_id* associated with it. We chose to only keep a few prescient columns such as speed, break\_angle, pitch\_type and situational descriptors such as strike and ball count and number of players on base at the time of the pitch. There was no clear primary key in this table so for our purposes, a serialized primary key would be created to serve as our unique identifier of each pitch.

The first transformation we did was to concatenate the first and last name columns in *player\_names* for readability. Next, we joined *atbats* with *games* to associate a date with every at bat. This joined dataframe was then filtered to only include at bats from 2018.

The next step was to perform a left join of *atbats* onto *pitches* because there were multiple instances of atbat\_id found in *pitches*. At this point, we could drop all null values from *pitches* which would drop pitches thrown in any year other than 2018 as well as incomplete rows. Finally, an *average\_speed* column was created for ease of use and the final *pitches* table was joined with the *player\_names* table on *pitcher\_id*. We then reorganized the columns and sorted the final table by pitcher name and re-exported this dataframe to a csv to then be loaded into a PostGres database.