**Goal of the Project**

In this project I chose to analyze a data set containing all historical Major League Baseball data. Specifically, I want to look and see if there is a model, without using a pitcher’s earned runs and innings pitched, that can be used to predict a pitcher’s Earned Run Average (ERA – calculated as **earned runs / innings pitched \* 9**). This statistically significant model would change sports prediction greatly which is why it interests me. While current models to predict ERA are naïve and simply take a predictive model for earned runs and a predictive model for innings pitched and then run them through the ERA formula, a model which would allow for a more holistic prediction of ERA would potentially bring a new, more accurate way of predicting pitcher efficiency in Major League Baseball.

The dataset used is the History of Baseball dataset downloaded from *kaggle* (SeanLahman)*.*

**Data Shaping and Import**

The first step was simply to download the dataset, extract it, and find out where my data would be located. Fortunately, I opened the Zip file to discover that its contents were a set of approximately 15 well-named CSV files. After a quick search I found a file called pitchers.csv. That took care of the problem of finding and sifting through my data. I could simply read that file into RStudio – and eventually MySQL and be ready to run some calculations. The next step was to connect to a database instance.

Connecting to the database required me to make a lot more assumptions and executive decisions right off the bat than I expected. I hadn’t thought, before starting the project, about the MySQL environment, what the database structure might look like when the project is opened, what server and port the MySQL server is running on, and the user accounts that might be configured. Unfortunately, there’s no way to configure this through R so I just had to make assumptions about much of that. I assume that there’s an active, healthy MySQL server running on **127.0.0.1** with a password-less user **root** who has full write access. From there, the dropping and recreating of the database to ensure data integrity is trivial and can be done through the **dbSendQuery** method in the **RMySQL** package as follows:

After reading in the CSV file, it looked like the data was in relatively decent shape, but there were some modifications that needed to be made. First, I wanted to filter the data to only contain pitchers from 1919 to today. Prior to 1919 was an era known as the “dead ball era” in which pitchers were the focus of the game, not batters. As a result, the pitchers’ ERAs were much lower. This period would skew the predictive model significantly so I remove those data points. I also remove 4 columns (intentional walks, sacrifice bunts, sacrifice fly outs, and grounded into double plays) from the dataset because they were not tracked until the mid-20th century and thus can’t be used as part of the predictive model because a large portion of the data points don’t contain values for those fields. Lastly, I converted the **league\_id** column to contain integers representing the league of the player instead of the strings “AL” and “NL.” The American League is represented by a 0 and the National League by a 1. I chose to do this because of the rule differences between the American and National Leagues, there’s a chance that the league is a factor and thus I wanted to be able to confirm or deny that theory.

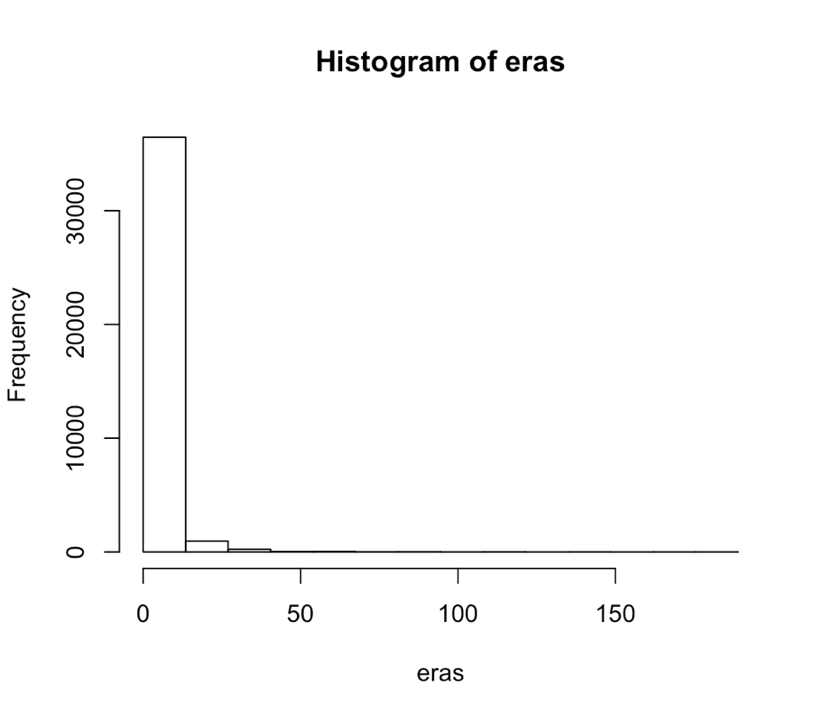
**modernEra <- raw[which(raw$year >= 1919), c(-21, -28, -29, -30)]**

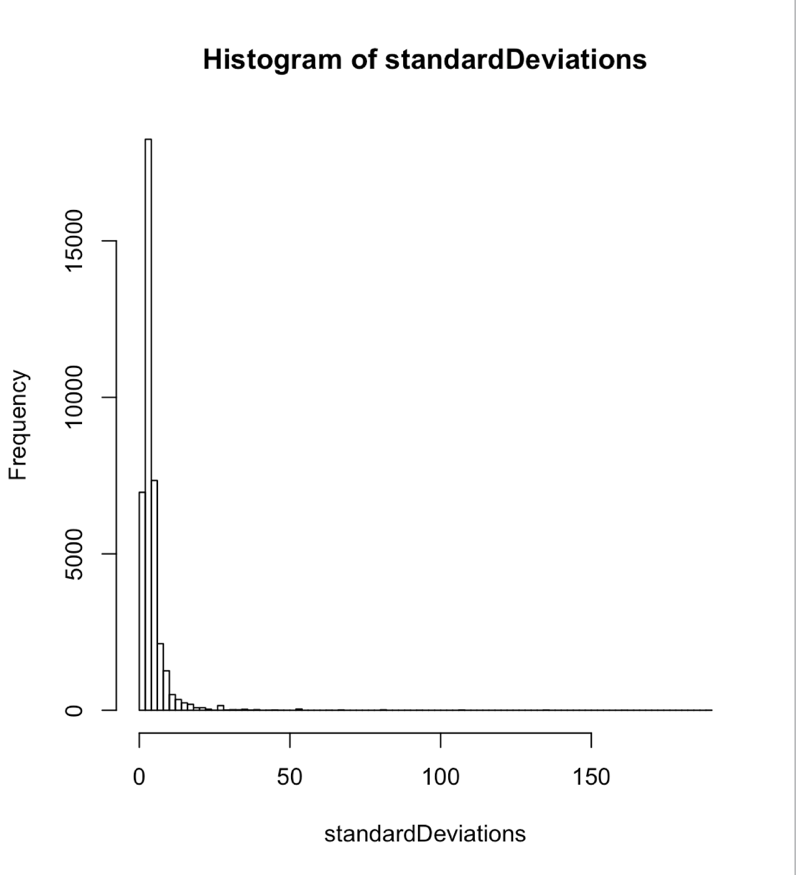
**ifelse(modernEra$league\_id == 'AL',**

**modernEra$league\_id <- 0, modernEra$league\_id <- 1)**

**Beginning the Analysis**

At this point, knowing we have all the data we need in our database, it’s time to start looking at the data just to get a general feel for the data. However, when I initially plotted the data on a histogram, I immediately noticed that outliers were going to be a huge factor in the dataset.



This logically makes sense, as there were plenty of pitchers with just a few innings pitched that could easily have ERAs of over 50. As such, I decided to calculate the standard deviation of each ERA and plot those. The graph looks very similar to the histogram of the ERAs themselves. This histogram was binned in sets of two standard deviations. Seeing that after the 14th column the numbers are effectively zero, it appears that the majority of ERAs fall within about 28 standard deviations of the mean. I tried to calculate how many pitchers’ ERAs fell outside of that range using the following calculation:

**> fetch(dbSendQuery(baseballDb,**

**'select count(\*) from pitchers where eraDev > 28'), n=-1)[1,1]**

**[1] 188**

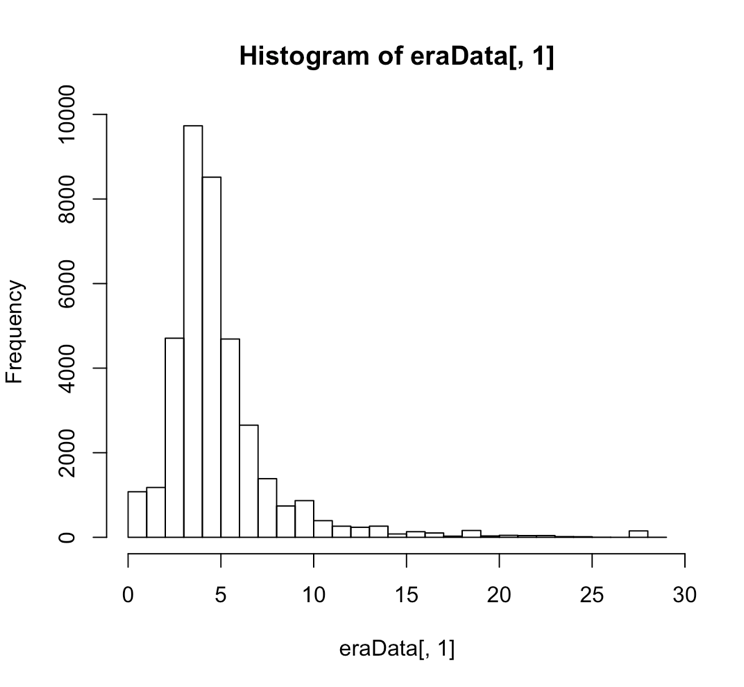
**> fetch(dbSendQuery(baseballDb, 'select count(\*) from pitchers'), n=-1)[1,1]**

**[1] 37815**

**> 188/37815**

**[1] 0.004971572**

As you can see, my calculations led to the finding that 99.5 percent of all pitchers ERAs fell within 28 standard deviations of the mean. While this is obviously more than the 3 standard deviations accepted as the true definition of an outlier, the wide variety of factors affecting ERAs clearly force us to break that rule. This meant that I had to change my data import to calculate the absolute deviation from the mean and then filter out all pitchers whose ERAs fell outside of 28 standard deviations. I’m consciously choosing to filter these pitchers out because there were likely other external circumstances that led to their abnormally high or low ERAs that we can’t account for in a statistical model. However, it may be interesting to later return and reprocess the data with the outliers added and see how the models change.

Once I removed those outliers, getting a feel for the data became much more feasible. I was able to plot the data on a histogram much more effectively, and though there are still a few outliers, the graph looks much more normal now. In addition to re-running the models with the outlier data included, I may similarly come back later and try to cut out some more of the data, seeing that there are still a considerable number of pitchers with ERAs above 15, this may cause the data to be skewed a bit.

**Building The Model**

Now that I had sanitized and processed the data using R, in order to generate the model I wanted to use Excel to pull data in from the SQL server. However, I quickly found out that I needed to install some drivers to allow me to connect to my local MySQL instance. After navigating OpenLink Software’s poorly-designed website, I finally found the drivers I thought I needed but after playing with Excel for more than an hour I couldn’t get anything to work. I guess we’re doing all of this in R.

After a quick search through the R docs I found the **lm()** function which, handily, performs a multiple regression analysis for us. Running the **lm()**function quickly showed a promising model. I chose to include 11 factors in the initial regression: **stint, wins, losses, hits, home runs, walks, strikeouts, opponent batting average, wild pitches, hit by pitches, and balks.** The initial regression provided interesting results pretty quickly:

**> model <- lm(eraData$era ~ eraData$stint +**

**+ eraData$w + eraData$l +**

**+ eraData$h + eraData$hr + eraData$bb + eraData$so +**

**+ eraData$baopp + eraData$wp + eraData$hbp + eraData$bk, eraData)**

**> summary(model)**

**Coefficients:**

**Estimate Std. Error t value Pr(>|t|)**

**(Intercept) -0.6937707 0.0749206 -9.260 < 2e-16 \*\*\***

**eraData$stint -0.2554949 0.0456467 -5.597 2.19e-08 \*\*\***

**eraData$w -0.0187580 0.0065877 -2.847 0.00441 \*\***

**eraData$l 0.0280837 0.0070788 3.967 7.28e-05 \*\*\***

**eraData$h -0.0165710 0.0006656 -24.895 < 2e-16 \*\*\***

**eraData$hr 0.0956475 0.0033100 28.896 < 2e-16 \*\*\***

**eraData$bb 0.0133892 0.0011366 11.780 < 2e-16 \*\*\***

**eraData$so -0.0084784 0.0006164 -13.756 < 2e-16 \*\*\***

**eraData$baopp 24.1526936 0.1801094 134.100 < 2e-16 \*\*\***

**eraData$wp 0.0093620 0.0063925 1.465 0.14306**

**eraData$hbp 0.0233496 0.0065704 3.554 0.00038 \*\*\***

**eraData$bk -0.0381781 0.0168480 -2.266 0.02346 \***

As you can see, the first model yielded several factors with microscopic P-values, and only one factor – wild pitches – above a 0.05, making it the only insignificant factor in the model. After removing the wild pitches from the model, I can now run the formula again.

**> model <- lm(eraData$era ~ eraData$stint +**

**+ eraData$w + eraData$l +**

**+ eraData$h + eraData$hr + eraData$bb + eraData$so +**

**+ eraData$baopp + eraData$hbp + eraData$bk, eraData)**

**> summary(model)**

**Coefficients:**

**Estimate Std. Error t value Pr(>|t|)**

**(Intercept) -0.6920283 0.0749124 -9.238 < 2e-16 \*\*\***

**eraData$stint -0.2554623 0.0456474 -5.596 2.20e-08 \*\*\***

**eraData$w -0.0198174 0.0065480 -3.026 0.002476 \*\***

**eraData$l 0.0281146 0.0070789 3.972 7.15e-05 \*\*\***

**eraData$h -0.0166025 0.0006653 -24.955 < 2e-16 \*\*\***

**eraData$hr 0.0956588 0.0033101 28.899 < 2e-16 \*\*\***

**eraData$bb 0.0138591 0.0010904 12.710 < 2e-16 \*\*\***

**eraData$so -0.0082326 0.0005931 -13.881 < 2e-16 \*\*\***

**eraData$baopp 24.1549309 0.1801057 134.115 < 2e-16 \*\*\***

**eraData$hbp 0.0238585 0.0065613 3.636 0.000277 \*\*\***

**eraData$bk -0.0371418 0.0168334 -2.206 0.027360 \***

Seeing that we’ve already removed all factors that are significant, this means we have theoretically constructed our “idea” model. At this point, I’ll also take a look at the R-squared to see how closely this model fits with the original data.

**Multiple R-squared: 0.4052, Adjusted R-squared: 0.405**

So it looks like our model is not a great fit for the original data. The next step is to try and calculate the estimates for each pitcher’s ERA and see exactly how far off we were. Since we have close to 30,000 records, I’m just going to look at aggregate statistics to determine the answer to this question rather than examining individual records. To do this, we simply calculate the absolute value of the difference between the actual ERAs and the predicted ones from the model above. Once we average all of those numbers we should have a good idea of how far off we were in our measurements. That is accomplished using the following snippets:

**> eraData <- fetch(dbSendQuery(baseballDb, 'select \* from pitchers'), n=-1)**

**> eraData$predictedEra <- -0.6920283 + (-0.2554623 \* eraData$stint) +**

**+ (-0.0198174 \* eraData$w) + (0.0281146 \* eraData$l) +**

**+ (-0.0166025 \* eraData$h) + (0.0956588 \* eraData$hr) +**

**+ (0.0138591 \* eraData$bb) + (-0.0082326 \* eraData$so) +**

**+ (24.1549309 \* eraData$baopp) + (0.0238585 \* eraData$hbp) +**

**+ (-0.0371418 \* eraData$bk)**

**> eraData$eraPredDev <- abs(eraData$era - eraData$predictedEra)**

**> mean(eraData$eraPredDev, na.rm=TRUE)**

**[1] 1.396108**

As we can see, our model’s predictions are approximately 1.40 earned runs off of the actual value for any given pitcher. Though we saw that our model didn’t fit the original data well (R2 = .405), depending on the use case this may be a reasonable-enough predictor of ERA to use in a real context. If we’re only off by about one and a half runs per game then for small-scale use cases, like predicting fantasy baseball performances, this model might actually suffice to produce something usable. For a higher-stakes use case, like a manager trying to determine whether he should accept a trade, this model may not be sufficient. The difference between a starting pitcher with an ERA of 3.00 and a pitcher with an ERA of 4.30 is the difference between an All-Star and a mediocre pitcher.

**Verification**

Just to compare, I’d now like to do a multiple variable regression using just a pitcher’s Earned Runs and Innings Pitched to see if using the naïve model will be better or worse than this model which theoretically should be a better predictor. This code is pretty much the same as the larger regression I just ran.

**> eraData <- fetch(dbSendQuery(baseballDb, 'select era, er, ipouts from pitchers'), n=-1)**

**> model <- lm(eraData$era ~ eraData$er + eraData$ipouts, data=eraData)**

**> summary(model)**

**Coefficients:**

**Estimate Std. Error t value Pr(>|t|)**

**(Intercept) 5.5046238 0.0222637 247.25 <2e-16 \*\*\***

**eraData$er 0.0905811 0.0013135 68.96 <2e-16 \*\*\***

**eraData$ipouts -0.0159866 0.0001828 -87.47 <2e-16 \*\*\***

**Multiple R-squared: 0.1914, Adjusted R-squared: 0.1914**

This regression is alarmingly bad, likely because we’re trying to predict a value based off of just two factors. However, I would’ve expected the R2 to be at least a little higher than 0.19. Similar to our activity with the other model, we should also see what our MAD is to see on average how close to the actual value our model gets.

**> eraData$predictedEra <- 5.5046238 + (0.0905811 \* eraData$er) + (-0.0159866 \* eraData$ipouts)**

**> eraData$eraPredDev <- abs(eraData$era - eraData$predictedEra)**

**> mean(eraData$eraPredDev, na.rm=TRUE)**

**[1] 1.645667**

As we can see, the new model, at least as far as MAD can measure is only about 18 percent worse than our larger model. This doesn’t tell us much, other than affirming that for any serious analysis of baseball statistics neither of these models would be usable.

# Bibliography

SeanLahman. *The History of Baseball*. September 2016. Zip File. <https://www.kaggle.com/seanlahman/the-history-of-baseball>.