Alexander Booth

Predict 411, Section 58

Professor Ott

October 8, 2017

Unit 01: "Moneyball Baseball Problem"

Introduction:

In this assignment, I will use OLS (“Linear Regression”) and the given statistics to predict the number of wins for a given baseball team. I will only use the variables given to me, or variables that I derive from the data provided. The data for this assignment contains approximately 2,200 records. Each record represents a professional baseball team from the years 1871 to 2006, inclusive. Each record contains the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season. I plan to use Ordinary Least Squares (OLS) regression to predict the number of wins for a team. I will construct multiple models and select a single model based on both model comparison criteria, such as AIC, BIC, and Adjusted R^2, as well as interpretability. The data manipulation steps, as well as the parameters from the model, will comprise a strategy for predicting wins on a test set.

I am provided with two data sets, one for training the model, the other to feed into the predicting strategy. The training data seems to have occurred during a long span of time, yet the data itself does not provide a variable to indicate time. The game of baseball has evolved over the period of 1871 to 2006, from the deadball era, through pitcher friendly and batter friendly periods. It seems unlikely that this dataset is comprised of homogeneous records from relatively similar eras. As such, I expect what is being measured will have experienced some changes over the history of baseball and these changes will likely manifest in the data. As I move further into examining the data, I will keep an eye out for outliers since they may be indication of previous ways in which baseball was being played.

I will use techniques learned in the, 410, such as dummy coding, automatic variable selection, and dimensionality reduction as well as those learned in the first few weeks of this class, such as variable transformation, fixing missing data and outliers, as well as linear regression. The hope is that through exploratory data analysis I can become intimate with the data, and use some of the above techniques to create a set of variables that will perform well when moving into model selection.

Exploratory Data Analysis:

My plan to learn and explore the data set will consist of the following steps:

- Examine the data, as well as the data dictionary. Fix any errors (e.g. label-switching)
- Consider the arithmetic relationship between variables (e.g. combining or decomposing), as well as contextual relationships between variables (e.g. variables that represent conceptually opposite measurements)
- Understand what data is missing
- Understand the initial relationship variables have with our dependent variable
- Impute variables that have missing data
- Examine variable distributions and consider further imputation or indicator coding
- Re-examine relationship between re-expressed/imputed independent variables and dependent variable
- Begin model construction

There are two data sets provided, one is the training data set comprised of 2276 observations. The other is a testing data set comprised of 259 observations. I will begin my exploratory data analysis (EDA) by examining the variables provided to us in the data dictionary.

Table 1: Data Dictionary with Proposed Theoretical Effect

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET_WINS		
TEAM_BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact on Wins
TEAM_BATTING_2B	Doubles by batters (2B)	Positive Impact on Wins
TEAM_BATTING_3B	Triples by batters (3B)	Positive Impact on Wins
TEAM_BATTING_HR	Homeruns by batters (4B)	Positive Impact on Wins
TEAM_BATTING_BB	Walks by batters	Positive Impact on Wins
TEAM_BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact on Wins
TEAM_BATTING_SO	Strikeouts by batters	Negative Impact on Wins
TEAM_BASERUN_SB	Stolen bases	Positive Impact on Wins
TEAM_BASERUN_CS	Caught stealing	Negative Impact on Wins
TEAM_FIELDING_E	Errors	Negative Impact on Wins
TEAM_FIELDING_DP	Double Plays	Positive Impact on Wins
TEAM_PITCHING_BB	Walks allowed	Negative Impact on Wins
TEAM_PITCHING_H	Hits allowed	Negative Impact on Wins
TEAM_PITCHING_HR	Homeruns allowed	Negative Impact on Wins
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive Impact on Wins

First, it is apparent that this data has been prepared for analysis since all of the variable labels are consistent. Additionally, each variable is continuous, and appears to be a count of a specific metric. Some of these metrics could likely be arithmetically combined, such as computing the number of “1B” hits by taking the difference between overall hits and “2B”, “3B”, “HR”. Another metric called “Total Bases” can be calculated by 1*1B + 2*2B + 3*3B + 4*HR. Finally, a metric called “On-Base Count” can be calculated by adding together “Hits” and “Walks” for batting and pitching respectively. I will create all four of those variables after missing data has been imputed and outliers have been fixed.

It is imperative to remain diligent to potential relationships between variables that can be explored, where one variable can help to provide context for examining another variable. Some examples of these presumed relationships are:

- TEAM_BATTING_H & TEAM_PITCHING_H
- TEAM_BATTING_BB & TEAM_PITCHING_BB

- TEAM_BATTING_HR & TEAM_PITCHING_HR

I believe that by looking at the descriptive statistics of one of these variables, I could compare those to the descriptive statistics of the other variable and have contextually informed observations. However, first I will examine the training data for missing values.

Table 2: Missing Values

TEAM_BATTING_H	0
TEAM_BATTING_2B	0
TEAM_BATTING_3B	0
TEAM_BATTING_HR	0
TEAM_BATTING_BB	0
TEAM_BATTING_SO	102
TEAM_BASERUN_SB	131
TEAM_BASERUN_CS	772
TEAM_BATTING_HBP	2085
TEAM_PITCHING_H	0
TEAM_PITCHING_HR	0
TEAM_PITCHING_BB	0
TEAM_PITCHING_SO	102
TEAM_FIELDING_E	0
TEAM_FIELDING_DP	286

A few of the variables have missing data. We will create two new variables in this process; one with the IMP_* prefix for the imputed variable, leaving the original variable untouched, and one with the m_* prefix as an indicator variable for the imputed variable. Sometimes, the fact that a variable was missing can actually be predictive. This means the indicator variables might be entered into the predictive model. For instance HBP and CS have only recently been kept track of. If they are missing, the data could be from an earlier period in baseball's history, which could have predictive capability. I will impute each of these variables with missing data with their median, since as I will soon show, most of these distributions are highly skewed and irregular. The median, unlike the mean, is less susceptible to extreme values.

Next, after imputing these values, I will examine the distribution of each variable via a histogram and boxplot for extreme values, outliers, or wrongly inserted data. As I said earlier, the visual examination of the variables as histograms and boxplots shows wildly differing from

expected normal shapes for several variables, some with straight-up impossible values. I will attach all of these graphics in the Appendix.

Some of these variables have an alarming shape to their distribution, with indication that there are some extreme values. I am looking for values that appear to be so extreme that I should simply impute them. With this line of reasoning I found several unreasonable seasonal averages for a couple variables. Here are some of my favorite outliers. One team in one season had their pitchers apparently log 19,278 strikeouts. Averaging that out per game gives 19,278/162 = 119 strikeouts per game. Unfortunately, there are only 9*3 = 27 outs that are possible in a game making this point impossible. As such, this is not an outlier, but wrong data. Another one is 30,132 pitching hits given up over one season. This equates to 186 hits given up a game which is just absurd. These are a just a couple of impossible examples. However, many other points, while not impossible, break records listed on baseball-reference and baseball-almanac. For instance, one team apparently had 0 wins in a season, when the record for single season losses is 20.

I am now going to consider historical reference data from Michael Bein's 'Century at a Glance' Graphical History of Baseball to further validate this point for a subset of variables. While it is scaled slightly differently, the distributions should still be similar.

Figure 1 Distribution of Pitcher Hits Allowed

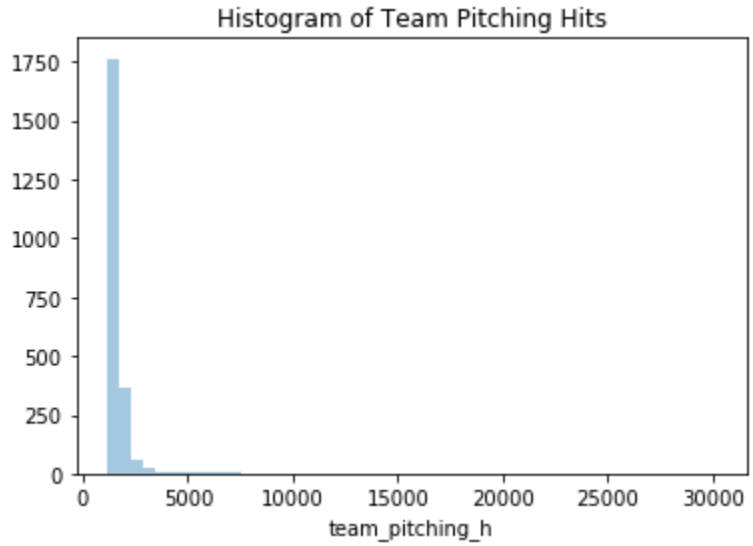
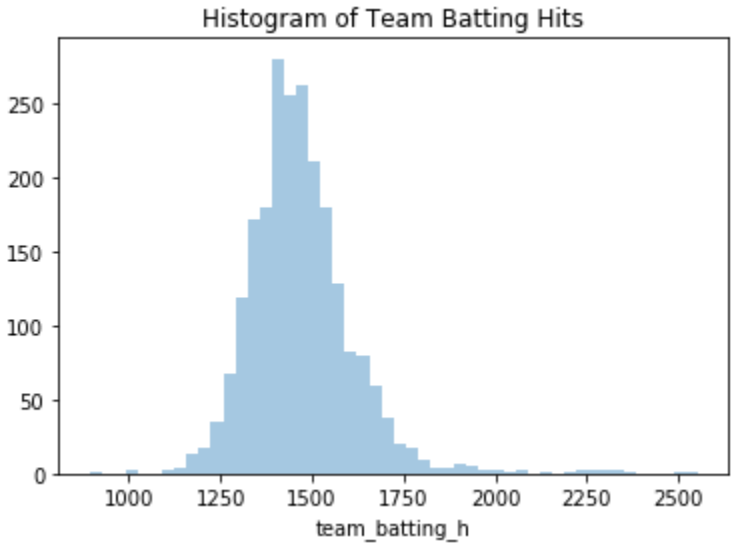


Figure 2: Historical Hits Per Game

The maximum hits allowed is 11. Over 162 games, this provides a maximum reference of 1,782 hits. This means that any points over that number is unreasonable to say the least. The

possible relationship TEAM_BATTING_H & TEAM_PITCHING_H should provide context for one another.

Figure 3: Batting Hits



This distribution looks more normal, but also contains extreme values, well over the 1,782 number determined above. Action needs to be taken to fix the outliers in both of those variables. Next is stolen bases per team per season.

Figure 4: Stolen Bases

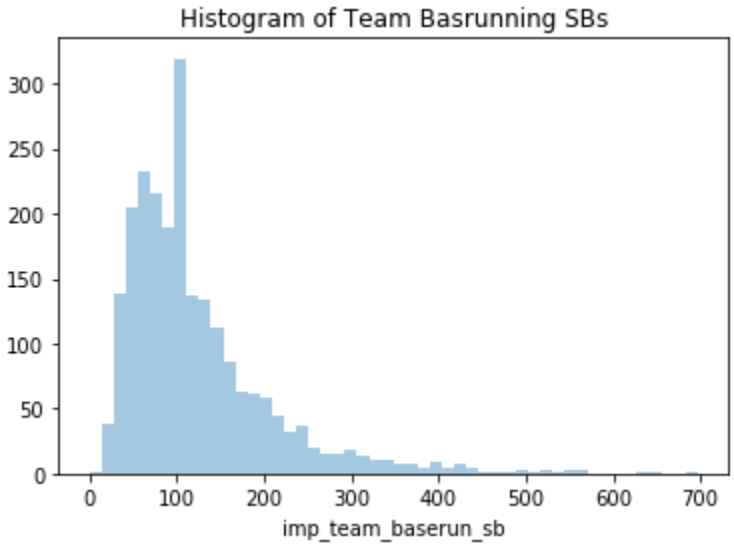


Figure 5: Historical Stolen Bases per Game

Again, the maximum rate per game is 1.5, which extends to 243 steals over a 162 game season. Our training data distribution of stolen bases has a long right tail extending up to 700! All of those points between 250 and 700 are outliers, and impossible. Next is pitching walks allowed, which should again be similar to batter walks.

Figure 6: Pitching Walks Allowed

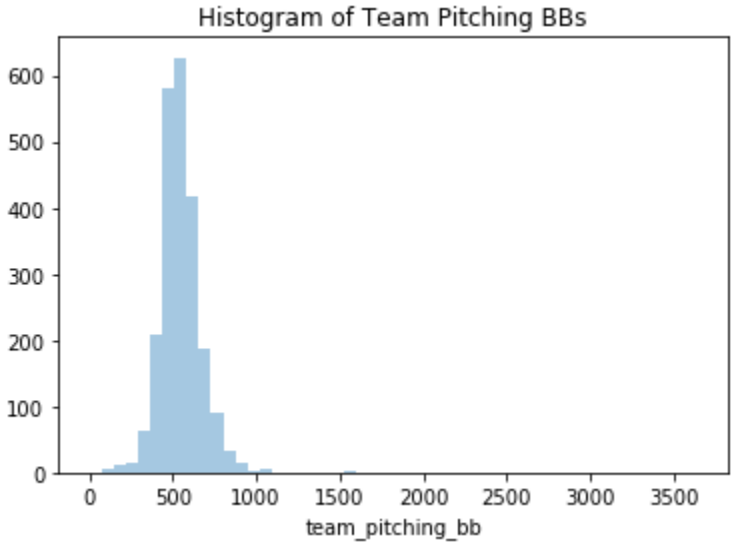


Figure 7: Batters Walked

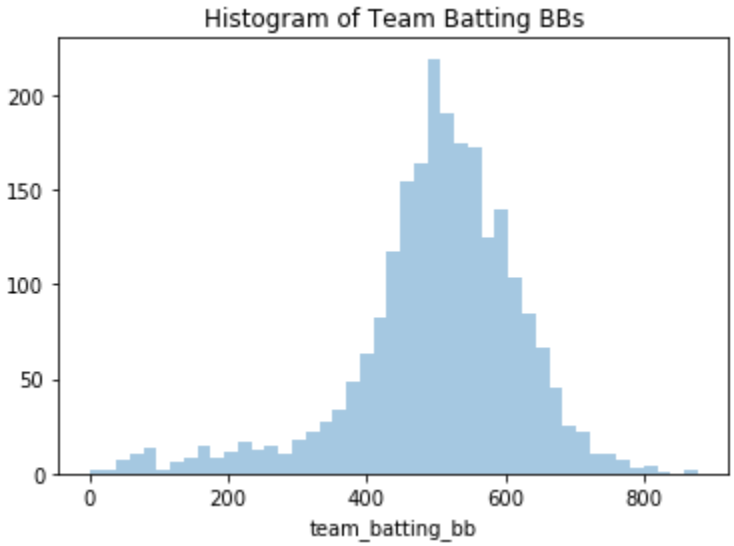
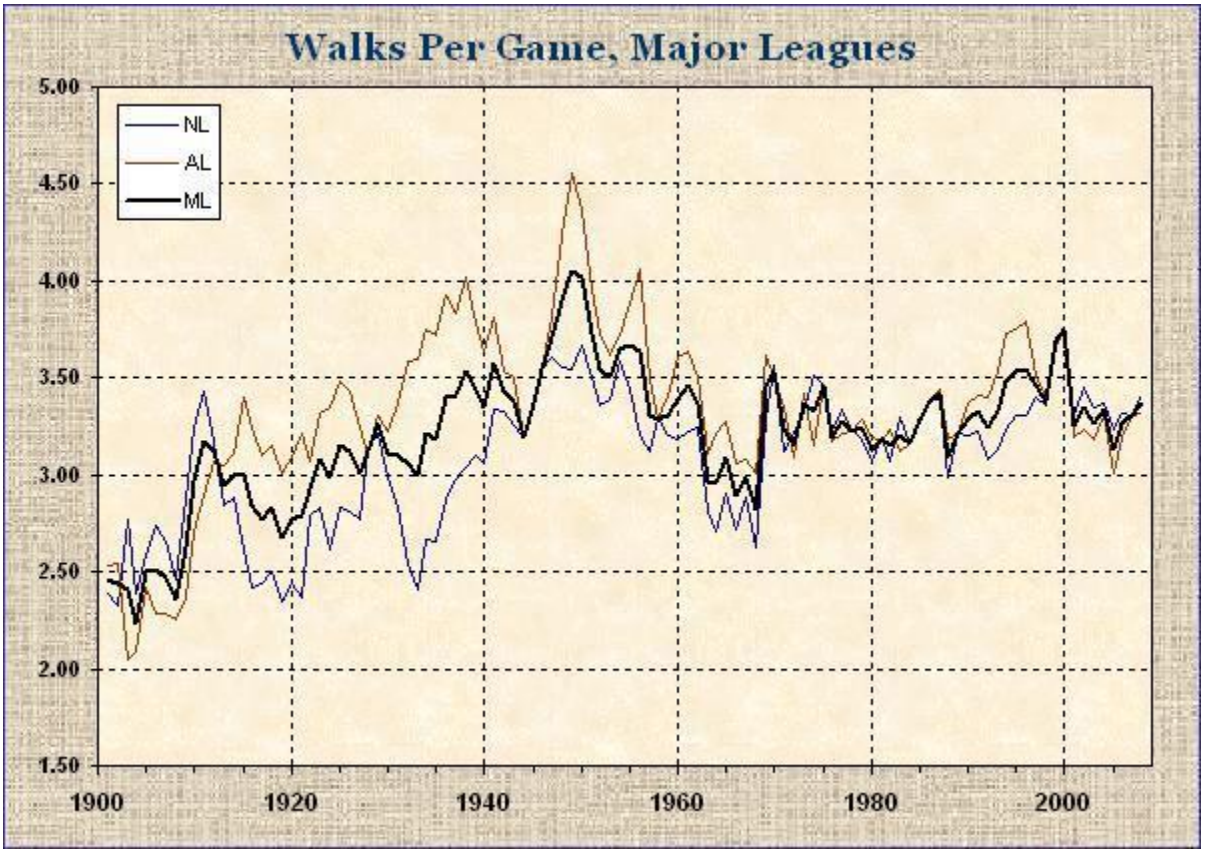


Figure 8: Historical Walks per Game



There are a couple of things wrong here. First, the maximum rate is around 4.5 walks per game which extends to 730 for a full season. Both walks allowed and batter walks have seasons greater than this number. The lowest historical rate is around 2 per game which extends to 324 for a season. Both walks allowed and batter walks have seasons less than this number. Finally, I will end with fielder's errors.

Figure 9: Distribution of Fielder's Errors

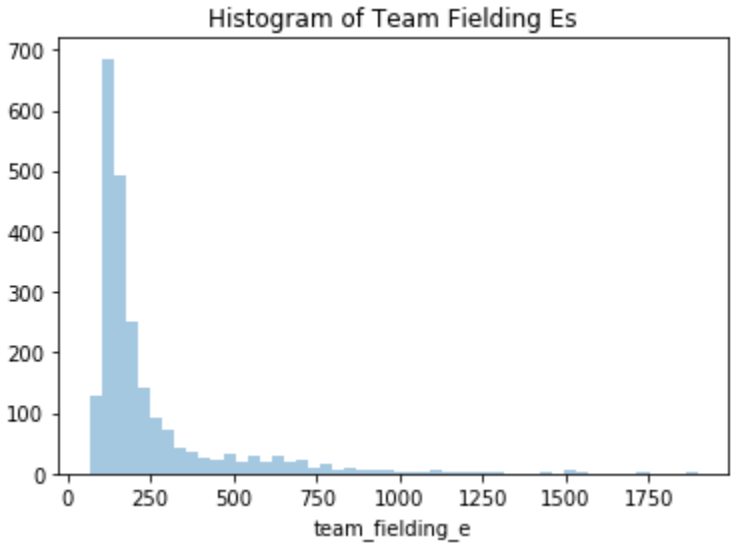


Figure 10: Historical Fielder's Errors

Again, the same kind of bad data can be seen. The maximum rate of errors was 2.5. Over a full season, that equates to 405 errors a season. However, our histogram shows a large

right tail with many values greater than 500 errors. After analyzing the data in this way, almost all of the variables had issues. Here is the full list. Wins, Batting Hits, Batting 2B, Batting 3B, Batting Walks, Batting SOs, Baserunning SB, Baserunning CS, Fielding Errors, Pitching Walks, Pitching Hits, and Pitching Strikeouts.

From our studies, I know that we have several options to fix this data:

- Truncating at a specific value (or set of values, e.g. ranges)
- log transformations
- Standardization (e.g. Z-Score)
- Binning (e.g. Buckets, Quantiles)
- Combining above techniques (e.g. log followed by binning)

Since I am committed to using the OLS regression as our model, I cannot simply choose to alter the modeling technique in the face of the extreme values. Even though this is an assignment, there are many industries that have regulatory limitations that would prevent one from pursuing a different model out of convenience. I chose to use the truncating strategy at this time based off of quantiles. For the variables listed above, if any values exceeded the 99th percentile, then they were replaced with the value of the 99th percentile. Likewise for values less than the 1th percentile. Finally, with missing values imputed and outliers mostly fixed, I created the four variables of 1Bs, Total Bases, and Batting and Pitching On-Base Count, as defined earlier. It was important to remember to do these same actions with the test data. As such, I imputed missing data with the medians from the training data and truncating the variables using the original 99th and 1th percentiles of the same variables from the training set.

Modeling:

Before moving onto to creating models, I first looked at the correlations of all of our variables with a heatmap and a pair plot.

Figure 11: Correlation Heatmap

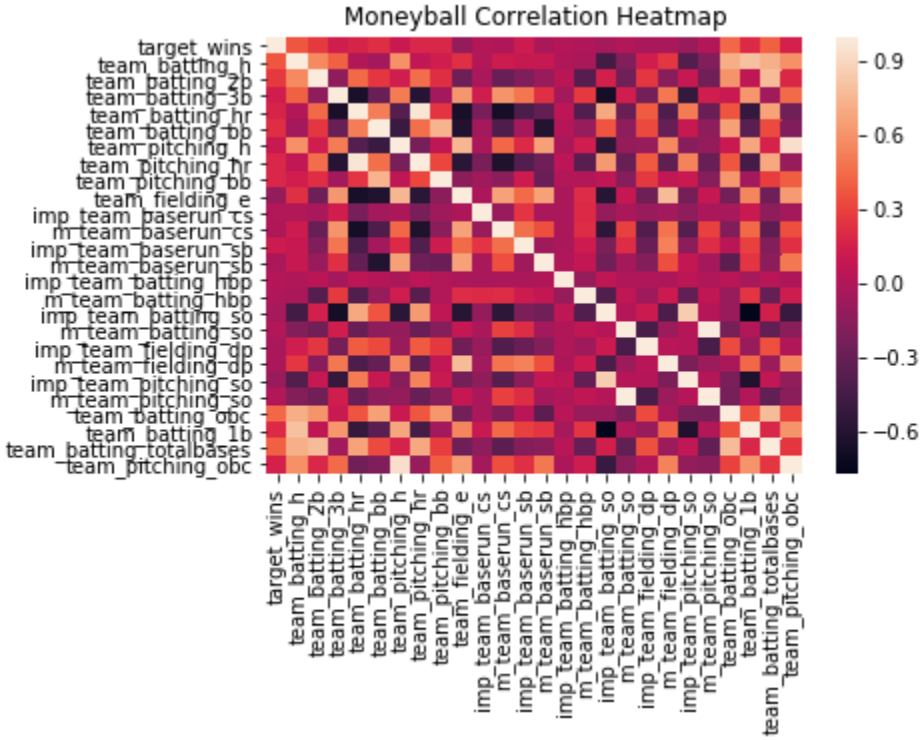
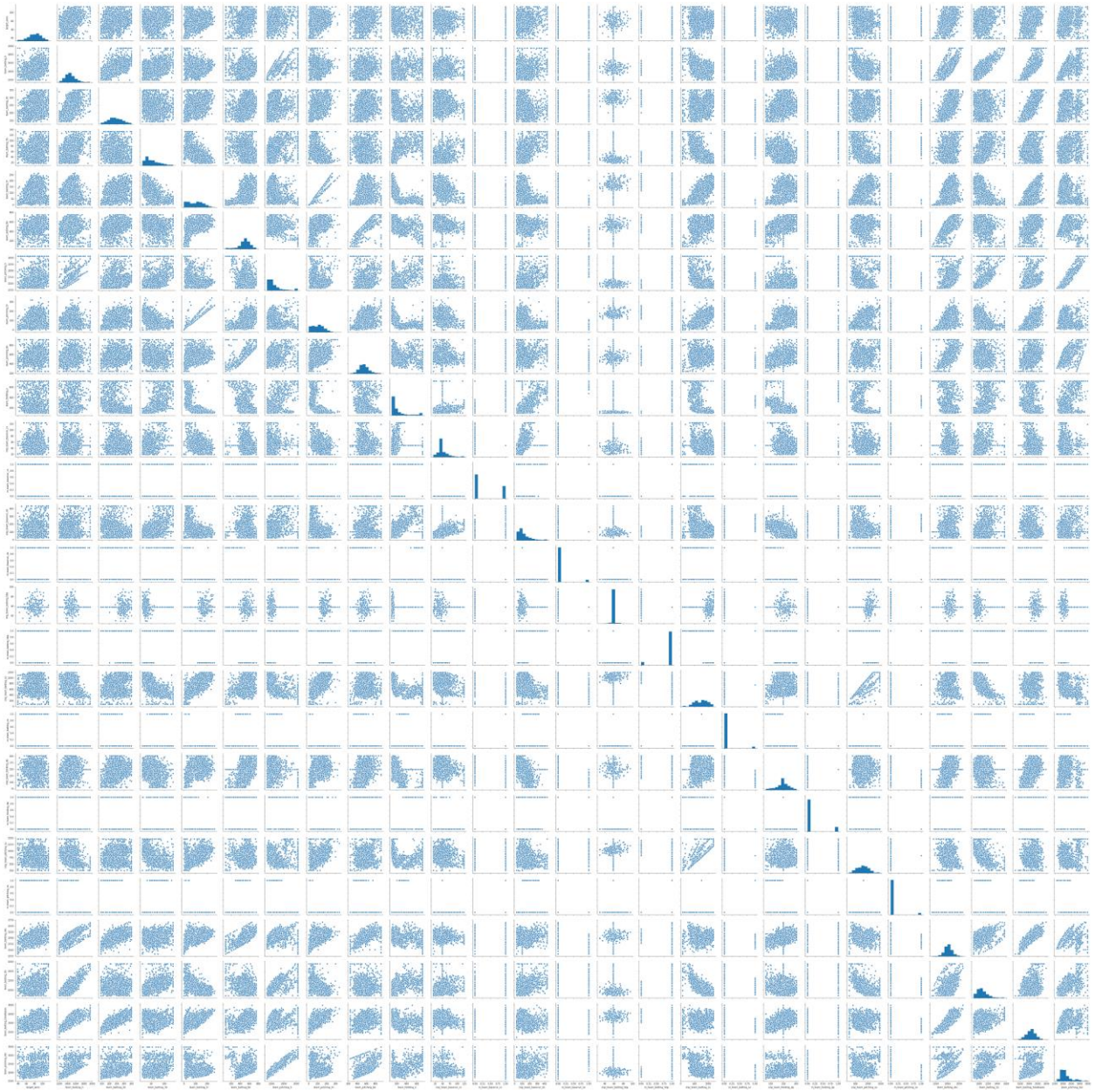


Figure 12: Correlation PairPlot



Perhaps unsurprisingly, total bases and the batting on base count had the highest correlation with target wins. These statistics are similar to On-Base Percentage and Slugging Percentage which sabermetricians have proved are correlated to runs scored, a precursor of wins and the numbers behind the OPS baseball statistic. Unfortunately, the correlation maps hint at potentially strong multi correlation issues in models developed, as many variables seemed to have a relationship with each other.

The first predictive model I created was a simple OLS regression model comprising of all 26 independent variables currently in the data set, including all imputed, fixed, and indicator variables. Here are the results from that model.

Table 3: Model 1 Summary

OLS Regression Results

Dep. Variable:	target_wins	R-squared:	0.388
Model:	OLS	Adj. R-squared:	0.382
Method:	Least Squares	F-statistic:	68.01
Date:	Fri, 06 Oct 2017	Prob (F-statistic):	6.94e-222
Time:	17:21:54	Log-Likelihood:	-8857.8
No. Observations:	2276	AIC:	1.776e+04
Df Residuals:	2254	BIC:	1.789e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	28.8234	7.166	4.022	0.000	14.770	42.877
team_batting_h	-0.0543	0.026	-2.131	0.033	-0.104	-0.004
team_batting_2b	-0.0079	0.013	-0.595	0.552	-0.034	0.018
team_batting_3b	0.0682	0.015	4.549	0.000	0.039	0.098
team_batting_hr	-0.0482	0.015	-3.198	0.001	-0.078	-0.019
team_batting_bb	0.0129	0.019	0.681	0.496	-0.024	0.050
team_pitching_h	-0.0024	0.004	-0.673	0.501	-0.010	0.005
team_pitching_hr	0.0455	0.028	1.647	0.100	-0.009	0.100
team_pitching_bb	-0.0019	0.005	-0.383	0.702	-0.012	0.008
team_fielding_e	-0.0748	0.005	-14.890	0.000	-0.085	-0.065
imp_team_baserun_cs	-0.0624	0.018	-3.381	0.001	-0.099	-0.026
m_team_baserun_cs	0.0450	0.901	0.050	0.960	-1.722	1.812
imp_team_baserun_sb	0.0678	0.006	12.298	0.000	0.057	0.079
m_team_baserun_sb	39.2427	2.378	16.504	0.000	34.580	43.905
imp_team_batting_hbp	0.0582	0.050	1.152	0.249	-0.041	0.157
m_team_batting_hbp	4.0423	1.093	3.699	0.000	1.899	6.185
imp_team_batting_so	-0.0218	0.007	-3.299	0.001	-0.035	-0.009
m_team_batting_so	3.8588	0.749	5.154	0.000	2.390	5.327
imp_team_fielding_dp	-0.1036	0.014	-7.631	0.000	-0.130	-0.077
m_team_fielding_dp	7.1433	1.618	4.415	0.000	3.970	10.316
imp_team_pitching_so	0.0078	0.005	1.428	0.153	-0.003	0.019
m_team_pitching_so	3.8588	0.749	5.154	0.000	2.390	5.327
team_batting_obc	0.0167	0.018	0.931	0.352	-0.019	0.052
team_batting_1b	0.0437	0.017	2.613	0.009	0.011	0.077
team_batting_totalbases	0.0395	0.010	3.936	0.000	0.020	0.059
team_pitching_obc	-0.0043	0.002	-1.996	0.046	-0.009	-7.6e-05

Omnibus:	28.509	Durbin-Watson:	1.217
Prob(Omnibus):	0.000	Jarque-Bera (JB):	49.219
Skew:	0.008	Prob(JB):	2.05e-11
Kurtosis:	3.720	Cond. No.	1.00e+16

Table 4: Model 1 Anova

	sum_sq	df	F	PR(>F)
team_batting_h	644.573618	1.0	4.540962	3.320093e-02
team_batting_2b	50.284166	1.0	0.354247	5.517784e-01
team_batting_3b	2937.984165	1.0	20.697829	5.663133e-06
team_batting_hr	1451.892541	1.0	10.228450	1.402086e-03
team_batting_bb	65.884401	1.0	0.464150	4.957610e-01
team_pitching_h	64.330044	1.0	0.453199	5.008886e-01
team_pitching_hr	385.265325	1.0	2.714159	9.960077e-02
team_pitching_bb	20.799883	1.0	0.146533	7.019066e-01
team_fielding_e	31472.540519	1.0	221.721161	6.677853e-48
imp_team_baserun_cs	1622.526027	1.0	11.430547	7.347838e-04
m_team_baserun_cs	0.353671	1.0	0.002492	9.601940e-01
imp_team_baserun_sb	21466.890870	1.0	151.232277	1.092550e-33
m_team_baserun_sb	38665.023037	1.0	272.391541	7.384246e-58
imp_team_batting_hbp	188.470660	1.0	1.327759	2.493266e-01
m_team_batting_hbp	1942.393261	1.0	13.683982	2.214661e-04
imp_team_batting_so	1544.487525	1.0	10.880773	9.867909e-04
m_team_batting_so	3770.036963	1.0	26.559565	2.777629e-07
imp_team_fielding_dp	8266.377222	1.0	58.235869	3.408187e-14
m_team_fielding_dp	2766.383905	1.0	19.488921	1.059600e-05
imp_team_pitching_so	289.383339	1.0	2.038679	1.534813e-01
m_team_pitching_so	3770.036963	1.0	26.559565	2.777629e-07
team_batting_obc	123.059032	1.0	0.866940	3.519037e-01
team_batting_1b	969.004116	1.0	6.826545	9.040963e-03
team_batting_totalbases	2198.504761	1.0	15.488264	8.553347e-05
team_pitching_obc	565.579317	1.0	3.984454	4.604209e-02
Residual	319947.387514	2254.0	NaN	NaN

Figure 13: Model 1 Residual vs Predicted Plot

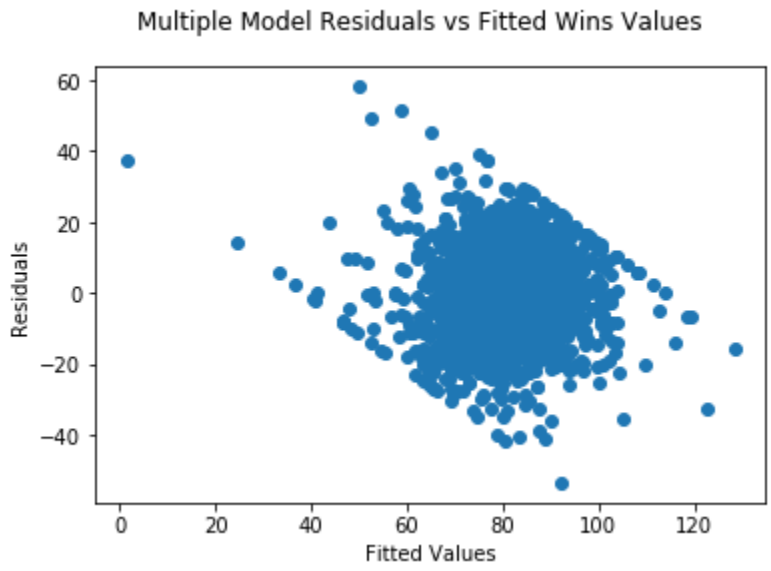
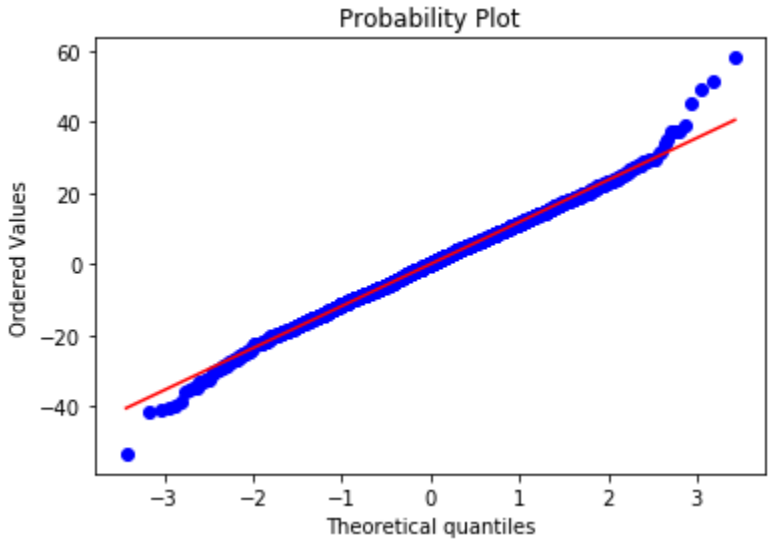


Figure 14: Model 1 Residual QQ-Plot



While this model seems to perform adequately, based off its relatively high, at least to the rest of the models, Adjusted-R^2 value, there a few things to comment. First, the many variables used present a good case for overfitting. Essentially, this generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study. This model is quite complex. Additionally, not all of the independent variables are significant at the .05 alpha level. Furthermore, not all of variables are actually independent and multicollinearity does manifest itself in this model. The residual vs predicted graph shows two very clear linear limits, and is not randomized at the more extreme game wins. The QQ-Plot supports this as it shows that the residuals detour away from normality at the extremes. Finally, the mean absolute error of this model was 9.36. This model suffers from overfitting, multi-collinearity, and poor estimations for extreme values of the dependent variable.

For the second model I created, I decided to eliminate multicollinearity as defined by the Variance Inflation Factor or VIF. After creating a feature auto-selection function, I set the VIF limit to be three. The following features were then used in the second model:

Table 5: Features with VIF less than 3

	VIF Factor	Features
0	491.679650	Intercept
1	2.339449	team_batting_3b
2	2.284751	team_batting_bb
3	1.330407	imp_team_baserun_cs
4	2.704105	m_team_baserun_cs
5	1.852084	imp_team_baserun_sb
6	2.611806	m_team_baserun_sb
7	1.011498	imp_team_batting_hbp
8	1.355563	m_team_batting_hbp
9	1.496432	imp_team_fielding_dp
10	1.385947	m_team_pitching_so
11	2.806899	team_batting_1b
12	2.148044	team_batting_totalbases
13	2.775525	team_pitching_obc

The output from a model created from these features are described below.

Table 6: Model 2 Summary

OLS Regression Results

Dep. Variable:	target_wins	R-squared:	0.311
Model:	OLS	Adj. R-squared:	0.307
Method:	Least Squares	F-statistic:	78.40
Date:	Sat, 07 Oct 2017	Prob (F-statistic):	7.26e-172
Time:	23:22:09	Log-Likelihood:	-8993.0
No. Observations:	2276	AIC:	1.801e+04
Df Residuals:	2262	BIC:	1.809e+04
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-7.9987	5.866	-1.364	0.173	-19.502	3.505
team_batting_3b	0.0290	0.015	1.933	0.053	-0.000	0.058
team_batting_bb	0.0441	0.003	13.341	0.000	0.038	0.051
imp_team_baserun_cs	-0.0017	0.018	-0.093	0.926	-0.037	0.034
m_team_baserun_cs	-0.3961	0.919	-0.431	0.666	-2.198	1.406
imp_team_baserun_sb	0.0283	0.004	6.359	0.000	0.020	0.037
m_team_baserun_sb	19.7925	1.836	10.782	0.000	16.193	23.392
imp_team_batting_hbp	0.0492	0.071	0.696	0.487	-0.089	0.188
m_team_batting_hbp	5.7629	1.111	5.188	0.000	3.584	7.941
imp_team_fielding_dp	-0.0953	0.013	-7.227	0.000	-0.121	-0.069
m_team_pitching_so	6.3556	1.505	4.222	0.000	3.404	9.308
team_batting_1b	0.0214	0.004	5.738	0.000	0.014	0.029
team_batting_totalbases	0.0301	0.002	19.508	0.000	0.027	0.033
team_pitching_obc	-0.0094	0.001	-7.146	0.000	-0.012	-0.007

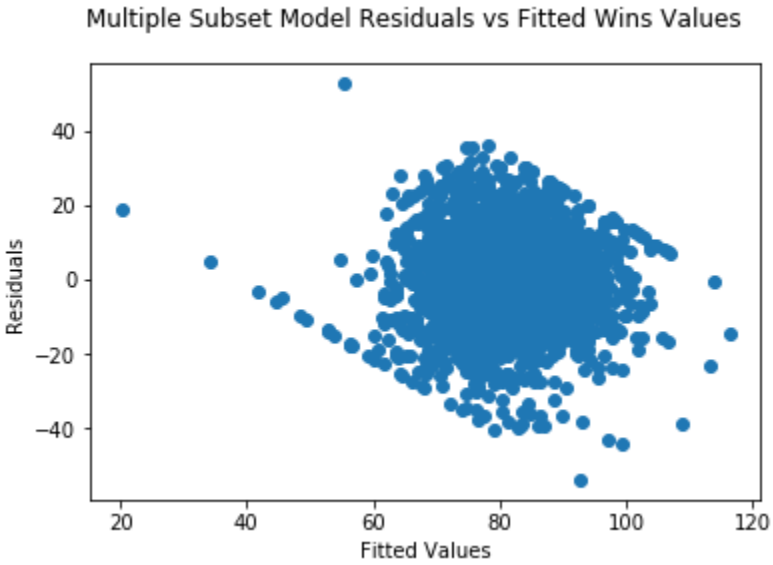
Omnibus:	20.278	Durbin-Watson:	1.134
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21.742
Skew:	-0.193	Prob(JB):	1.90e-05
Kurtosis:	3.284	Cond. No.	7.29e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.29e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

Table 7: Model 2 ANOVA Table

	sum_sq	df	F	PR(>F)
team_batting_3b	595.117347	1.0	3.735970	5.337742e-02
team_batting_bb	28351.524243	1.0	177.982450	3.868267e-39
imp_team_baserun_cs	1.375161	1.0	0.008633	9.259808e-01
m_team_baserun_cs	29.603323	1.0	0.185841	6.664421e-01
imp_team_baserun_sb	6440.428351	1.0	40.431097	2.456301e-10
m_team_baserun_sb	18517.657520	1.0	116.248355	1.822458e-26
imp_team_batting_hbp	77.132248	1.0	0.484213	4.865915e-01
m_team_batting_hbp	4286.782322	1.0	26.911146	2.319816e-07
imp_team_fielding_dp	8319.168571	1.0	52.225270	6.734816e-13
m_team_pitching_so	2839.603244	1.0	17.826186	2.516013e-05
team_batting_1b	5244.027586	1.0	32.920448	1.088747e-08
team_batting_totalbases	60623.501301	1.0	380.576339	1.829309e-78
team_pitching_obc	8134.745612	1.0	51.067517	1.198681e-12
Residual	360322.872929	2262.0	NaN	NaN

Figure 15: Model 2 Residual vs Predicted Plot



While this model reduced the effects of multicollinearity, it did so in expense of the ability to predict target wins. The Adjusted-R^2 decreased, the AIC and BIC both increased and the mean absolute error increased to 10.02. This model also seems to suffer most from predicting wins at the extremes, as smaller values are overpredicted and larger values are underpredicted. Additionally, a couple of variables are used which are deemed to be statistically insignificant. These are baserunning caught stealing and batting hit by pitch.

For my third model, I decided to reduce the dimensionality of the data by performing Principal Component Analysis, or PCA. The results of this analysis indicated that 7 vectors were significant, or had an eigenvalue of over 1.0. Using these 7 vectors, I performed a third regression analysis. The results are enumerated below.

Table 8: PCA Eigenvalues

```
Eigenvalues in descending order:
7.76612594775
4.92148728511
2.41770016668
1.90231437101
1.64910818575
1.12172234131
1.03966673667
0.944909507503
0.708564030372
0.600998816166
0.417003401123
0.336785000277
0.311005705564
0.274914181318
0.264729483459
0.173853287898
0.0719851188188
0.0433167204431
0.0188905908673
0.0111799588496
0.0037391630716
6.45489313779e-16
3.50719774835e-16
1.47276183948e-16
6.27260731302e-17
```

Figure 16: Scree Plot

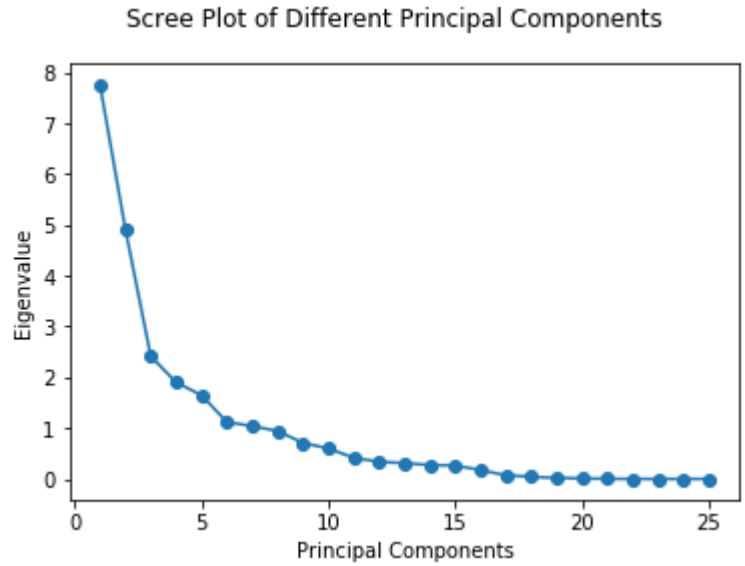


Figure 17: PCA Cumulative Explained Variance

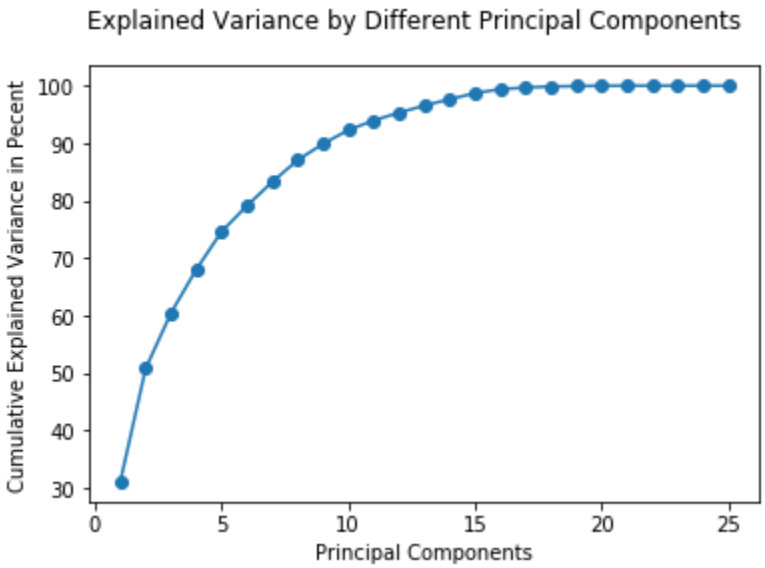


Figure 18: PCA Proportion of Explained Variance

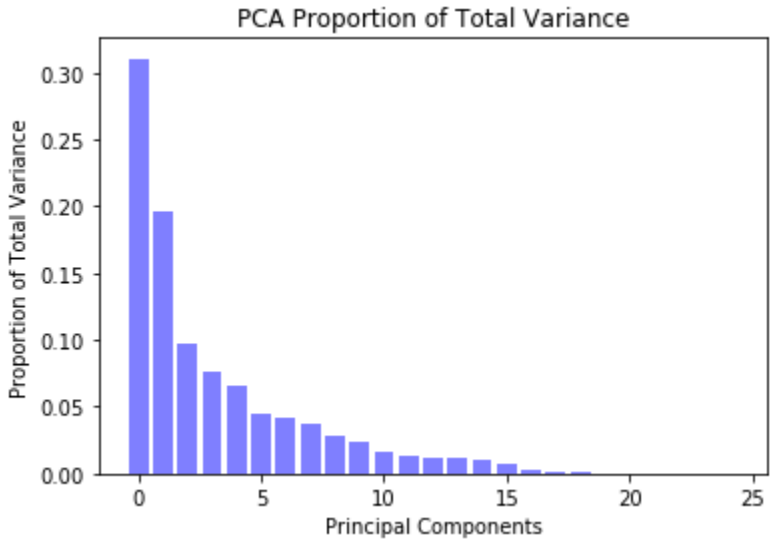


Table 9: Model 3 Summary

OLS Regression Results

Dep. Variable:	target_wins	R-squared:	0.219
Model:	OLS	Adj. R-squared:	0.216
Method:	Least Squares	F-statistic:	90.71
Date:	Sat, 07 Oct 2017	Prob (F-statistic):	8.12e-117
Time:	23:55:05	Log-Likelihood:	-9135.4
No. Observations:	2276	AIC:	1.829e+04
Df Residuals:	2268	BIC:	1.833e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	80.8235	0.281	287.363	0.000	80.272	81.375
pca1	-0.2219	0.101	-2.199	0.028	-0.420	-0.024
pca2	2.4426	0.127	19.266	0.000	2.194	2.691
pca3	-1.8370	0.181	-10.156	0.000	-2.192	-1.482
pca4	2.1228	0.204	10.410	0.000	1.723	2.523
pca5	-1.1041	0.219	-5.041	0.000	-1.534	-0.675
pca6	0.9776	0.266	3.681	0.000	0.457	1.498
pca7	-0.8029	0.276	-2.911	0.004	-1.344	-0.262

Omnibus:	21.251	Durbin-Watson:	1.022
Prob(Omnibus):	0.000	Jarque-Bera (JB):	22.066
Skew:	-0.214	Prob(JB):	1.62e-05
Kurtosis:	3.224	Cond. No.	2.79

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Table 10: Model 3 ANOVA

	sum_sq	df	F	PR(>F)
pca1	870.304851	1.0	4.833775	2.800813e-02
pca2	66828.784063	1.0	371.174861	1.007333e-76
pca3	18569.038859	1.0	103.134608	9.991121e-24
pca4	19510.805611	1.0	108.365290	8.022617e-25
pca5	4575.635690	1.0	25.413614	4.992490e-07
pca6	2439.718527	1.0	13.550481	2.376517e-04
pca7	1525.470805	1.0	8.472643	3.640427e-03
Residual	408345.763864	2268.0	NaN	NaN

Figure 19: Model 3 Residual vs Predicted

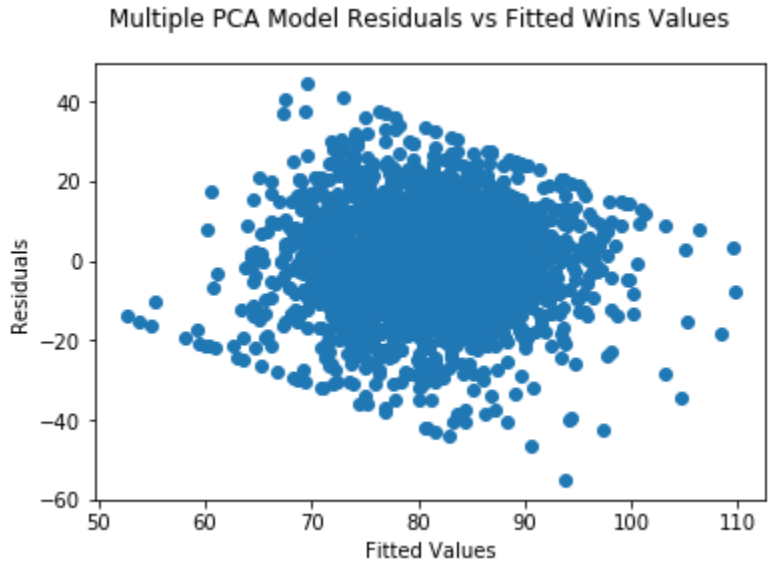
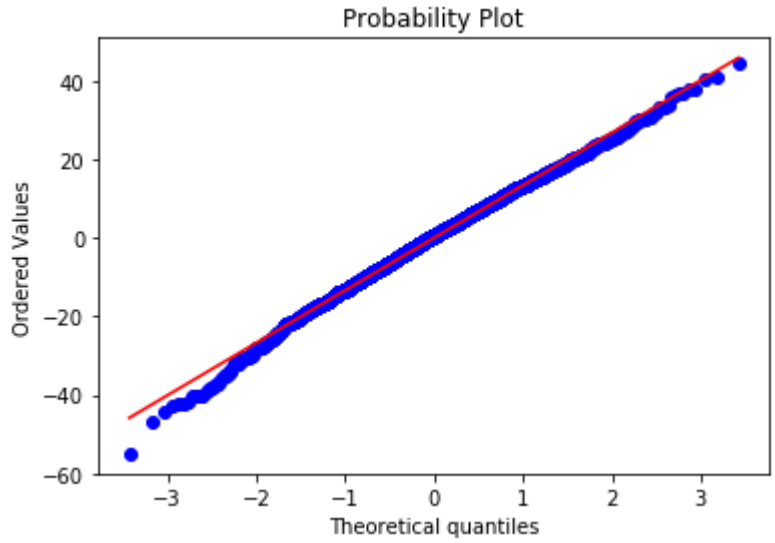


Figure 20: Model 3 QQ-Plot



This model completely eradicated multicollinearity. However, only using the top 7 PCA vectors reduces the Adjusted-R^2 tremendously as well as increases the AIC and BIC. However, the residual plot is a lot more normal and the qq-plot shows less of an issue at the extremes. In actuality, this model performs poorly in its predictive ability.

For my final model, I used a combination of variable transformations and automatic feature selection of statistically significant variables. I also created a variable called Batting_Fake_OPS which was just the total bases added to the on base count. The variables I transformed are the following: Hits, Total Bases, On Base Count, and Caught Stealing were all transformed by the log function, and Fielding Errors and Fielding Double Plays were transformed by the square root function. Here are the results from that final model:

Table 11: Model 4 Summary

OLS Regression Results

Dep. Variable:	target_wins	R-squared:	0.409
Model:	OLS	Adj. R-squared:	0.405
Method:	Least Squares	F-statistic:	92.04
Date:	Sun, 08 Oct 2017	Prob (F-statistic):	5.33e-243
Time:	00:10:19	Log-Likelihood:	-8817.2
No. Observations:	2276	AIC:	1.767e+04
Df Residuals:	2258	BIC:	1.777e+04
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-223.9113	246.429	-0.909	0.364	-707.163	259.340
team_batting_fakeops	0.0318	0.010	3.034	0.002	0.011	0.052
team_batting_bb	-0.1193	0.023	-5.182	0.000	-0.164	-0.074
team_batting_2b	-0.0664	0.011	-5.879	0.000	-0.089	-0.044
team_batting_hr	-0.1094	0.021	-5.216	0.000	-0.151	-0.068
log_team_batting_h	-258.6462	37.119	-6.968	0.000	-331.437	-185.855
log_team_batting_totalbases	61.6530	21.944	2.810	0.005	18.621	104.685
log_team_batting_obc	230.6075	42.639	5.408	0.000	146.992	314.223
sqrt_team_fielding_e	-2.9351	0.167	-17.546	0.000	-3.263	-2.607
log_imp_team_baserun_cs	-3.5408	0.957	-3.699	0.000	-5.418	-1.664
m_team_baserun_cs	1.9206	0.853	2.252	0.024	0.248	3.593
imp_team_baserun_sb	0.0689	0.005	12.852	0.000	0.058	0.079
m_team_baserun_sb	37.1982	1.865	19.945	0.000	33.541	40.856
m_team_batting_hbp	4.7744	1.071	4.458	0.000	2.674	6.874
imp_team_batting_so	-0.0161	0.002	-7.024	0.000	-0.021	-0.012
m_team_batting_so	4.8884	0.761	6.424	0.000	3.396	6.381
sqrt_imp_team_fielding_dp	-2.5695	0.321	-8.004	0.000	-3.199	-1.940
m_team_fielding_dp	6.6335	1.511	4.391	0.000	3.671	9.596
m_team_pitching_so	4.8884	0.761	6.424	0.000	3.396	6.381

Omnibus:	29.001	Durbin-Watson:	1.262
Prob(Omnibus):	0.000	Jarque-Bera (JB):	49.369
Skew:	-0.045	Prob(JB):	1.90e-11
Kurtosis:	3.716	Cond. No.	7.33e+19

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 7.73e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Table 12: Model 4 ANOVA Table

	sum_sq	df	F	PR(>F)
team_batting_fakeops	1258.994394	1.0	9.207852	2.437274e-03
team_batting_bb	3671.348848	1.0	26.850983	2.392647e-07
team_batting_2b	4725.693611	1.0	34.562099	4.741644e-09
team_batting_hr	3719.873124	1.0	27.205873	1.995672e-07
log_team_batting_h	6638.732345	1.0	48.553406	4.201468e-12
log_team_batting_totalbases	1079.342852	1.0	7.893943	5.002784e-03
log_team_batting_obc	3999.477035	1.0	29.250800	7.029672e-08
sqrt_team_fielding_e	42094.019553	1.0	307.861188	1.033146e-64
log_imp_team_baserun_cs	1870.881213	1.0	13.682982	2.215741e-04
m_team_baserun_cs	693.302375	1.0	5.070575	2.443105e-02
imp_team_baserun_sb	22585.936407	1.0	165.185775	1.539493e-36
m_team_baserun_sb	54393.982727	1.0	397.818891	1.176174e-81
m_team_batting_hbp	2717.710277	1.0	19.876399	8.665794e-06
imp_team_batting_so	6746.381717	1.0	49.340717	2.836464e-12
m_team_batting_so	5643.336346	1.0	41.273422	1.609030e-10
sqrt_imp_team_fielding_dp	8760.237034	1.0	64.069362	1.902625e-15
m_team_fielding_dp	2636.828948	1.0	19.284860	1.177921e-05
m_team_pitching_so	5643.336346	1.0	41.273422	1.609030e-10
Residual	308737.508021	2258.0	NaN	NaN

Figure 21: Model 4 Residual Plot

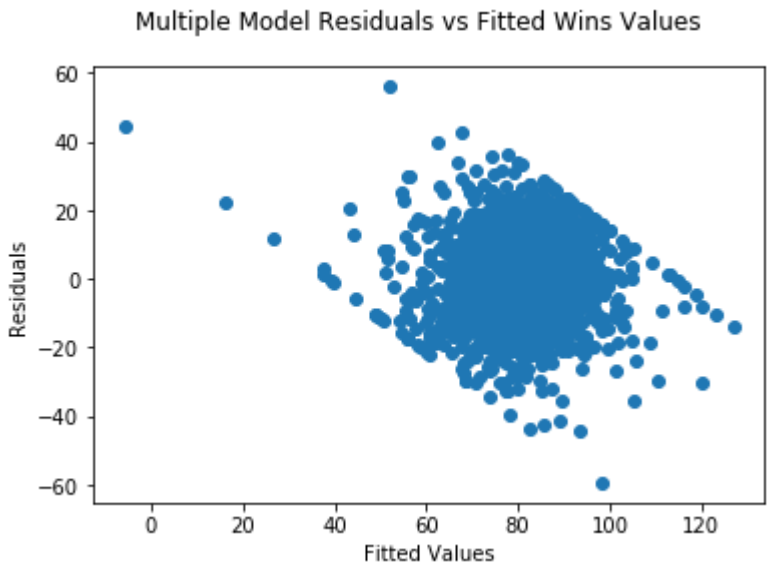
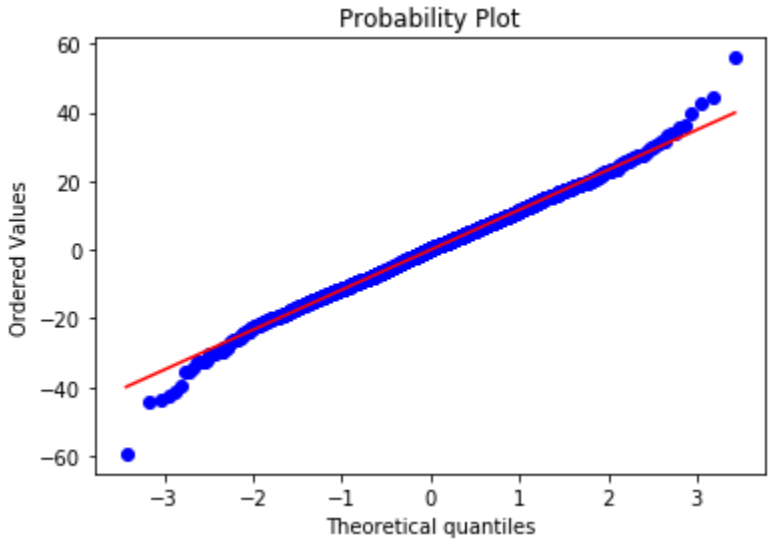


Figure 22: Model 4 QQ_Plot



This model performs the best out of the bunch. The Adjusted_R^squared has jumped above .40, the AIC and BIC have decreased, and every predictor variable is statistically significant at the alpha = .05 level. Unfortunately, this model does suffer from multicollinearity and predicts the edge cases of wins wrong quite drastically. The QQ-Plot shows a large deviation in normality at the extreme values, reinforcing these idea. However, this model seems to predict target wins the best. In fact, when used to predict wins on the test data set, it produced the best score on Kaggle as of early Sunday morning, 10/08/2017.

Here is a quick caveat. I did try and bin the fake OPS variable, however the resulting variable was not statistically significant and reduced the Adjusted-R^2 value and increased the AIC and BIC. As such, I did not include it in the final model. Additionally, of this final model, a few coefficients did not make much sense. For instance, the log of hits is quite negative. I would expect this value to be positive. However, due to collinearity with OBC and fake OPS, it makes some sense that this value is backwards. Other wrong coefficients are walks, doubles, and homeruns, which can be attributed to the strong multicollinearity issue, as well as fielding double plays, which is off by itself. Double plays have no collinearity with other variables and should increase the amount of wins a team has, however this model says the opposite is true.

Model Selection:

The best model in effect can account for 40% of the variability that is apparent in the training data set. I am more accustomed to working with data sets where I can build models that have much higher goodness- of-fit diagnostics without nearly the amount of effort put into this model. Between all of the models constructed, a few of which are not documented in this report, I have to consider that the expected predictive ability on this data set is likely going to be quite low. It is highly likely that Simpson's Paradox in in effect, which states, "what is true for the whole population may not be true for any of the subpopulations". I am pretty confident that there are multiple sub-populations within this data, most likely due to how long of a period the data was being collected over. My attempts at trying to isolate this phenomena have only served to convolute the interpretation of the best model.

In dealing with wins in baseball, most analysts have determined that they are quite random and hard to predict. This is why many statistics used in sabermetrics deal with runs scored and runs allowed. Runs are the precursor to wins, and while wins are hard to predict, runs are much easier to correlate with and predict by simple statistics. In fact, if this study was done with runs scored, on-base percentage and slugging percentage percentage would perform the best in tandem to predict runs scored.

However, I am going to select the best model based off of Adjusted-R^2, AIC, BIC, and statistically significant predictor variables. In this case, I am forced to recommend my last model In this model, target wins is equal to 0.318 * fake_OPS - 0.1193 * batting_bb - .0664 * batting_2b - 0.1094 * batting_hr - 258.6462 * log(batting_hits) + 61.653 * log(batting_totalBases) + 230.6075 * log(batting_OBC) - 2.9351 * sqrt(fielding_errors) - 3.5408 * log(baserunning_cs) + 1.9206 * m_team_baserun_cs + .0689 * baserun_sb + 37.1982 * m_team_baserun_sb + 4.7744 * m_team_batting_hbp - 0.0161 * batting_so + 4.8884 * m_team_batting_so - 2.5695 * sqrt(fielding_dp) + 6.6335 * m_team_fielding_dp + 4.8884 * team_pitching_so - 223.9113

Conclusion:

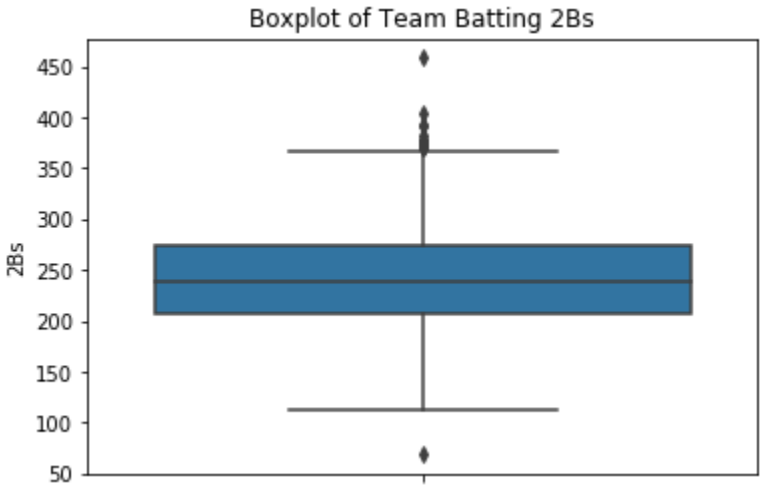
Obviously, this model is quite complex. However, wins in baseball are a complex entity. Interestingly, none of the pitching statistics made it into the final model. This is confusing, as pitching should make up about 50% percent of the games and have an effect on wins. This model does enforce that players with a high OPS and low error rate should be sought after on offense. This makes intuitive sense as more people on base, or getting more bases per plate appearance, creates more runs scored which should increase the number of wins for a team.

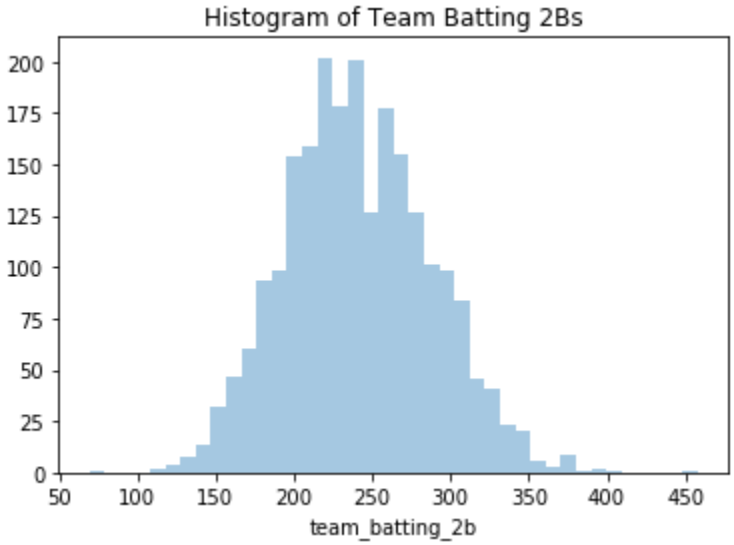
Unfortunately, I did not even comprehensively exhaust all of the modeling options at my disposal. The confidence in deploying this model for predictive value is relatively low. Given a data set that provided context, such as year or league, would have allowed me to build multiple models for what is likely very different subpopulations. Further, if the data set was not so error ridden, a more comprehensive and accurate model could have been created.

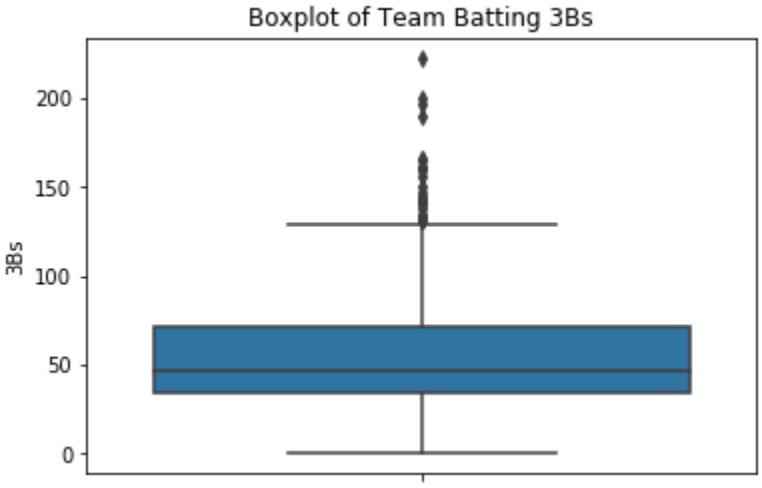
It may be best for teams in the future to predict runs scored or allowed through various statistics. OPS for runs scored and opponent OPS and pitcher WHIP are known to be good and correlated variables. Predicting runs provides a robust estimate for predicted wins, through formulas such as the Pythagorean formula. Additionally, predicting runs eliminates the noise and random error associated with predicting wins. I would recommend further analysis into the study of predicting runs, and then using that to predict wins, instead of this analysis to solely predict wins from basic baseball statistics.

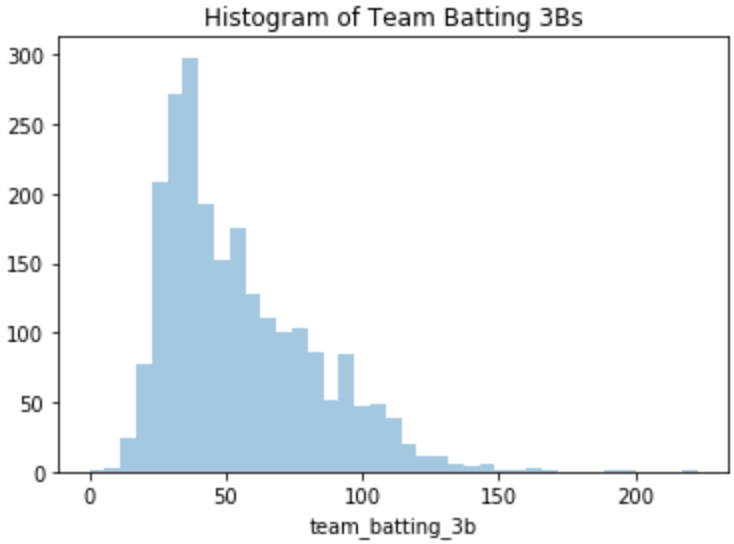
Appendix

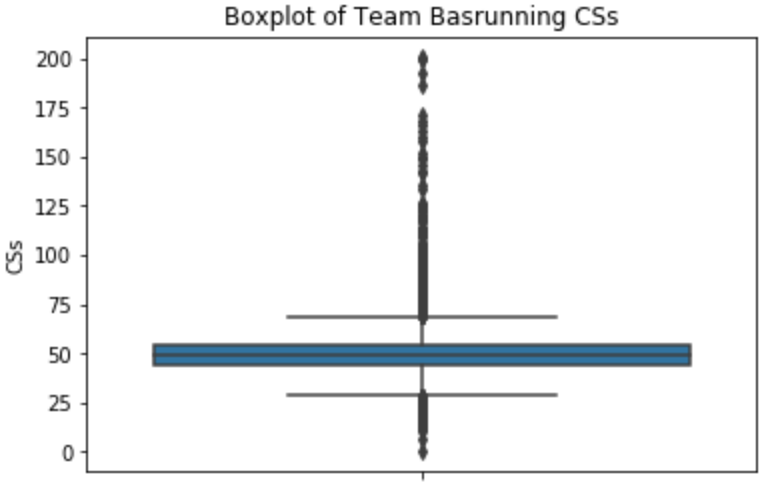
EDA Graphics

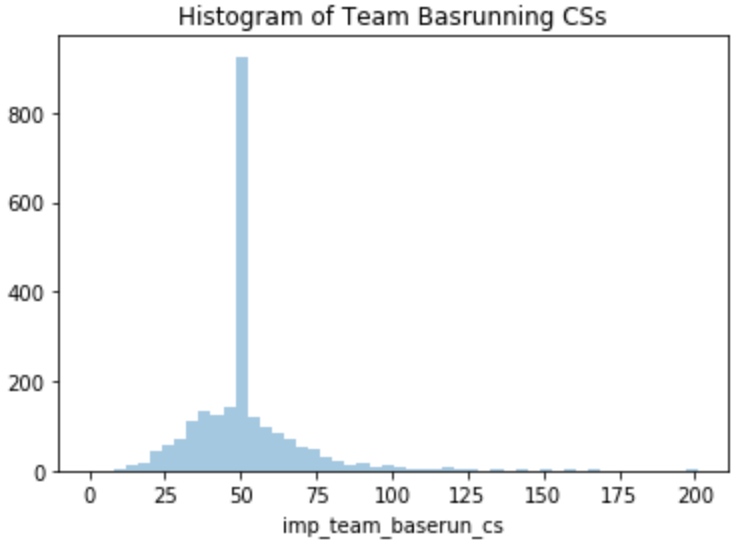


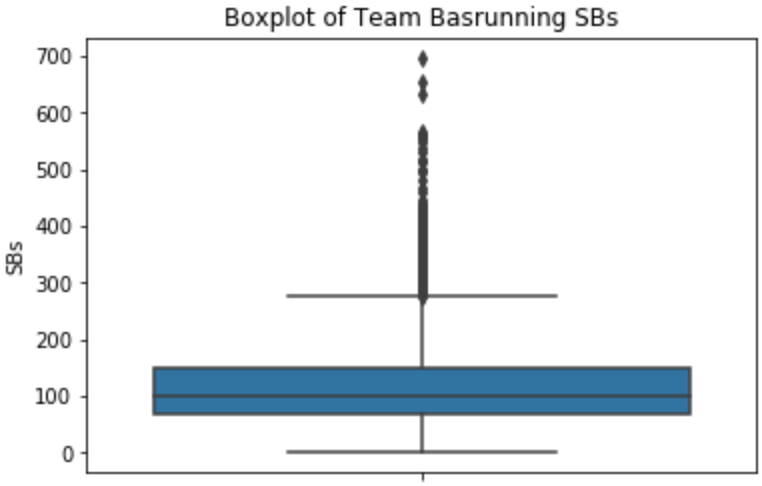


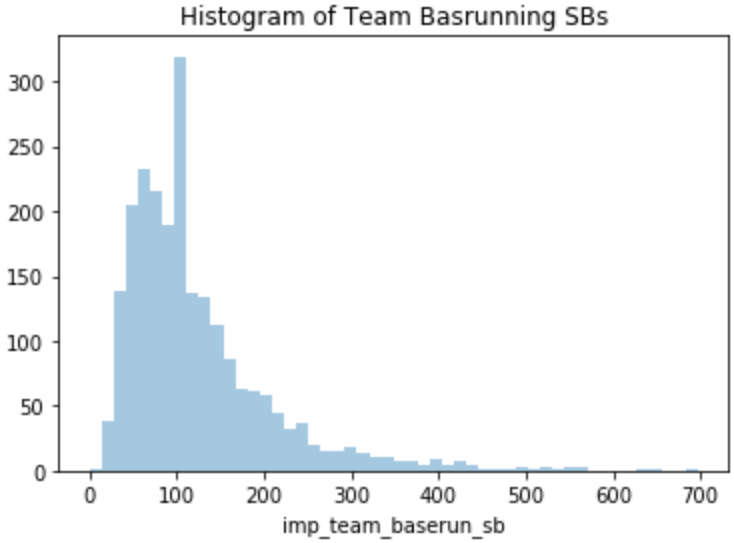


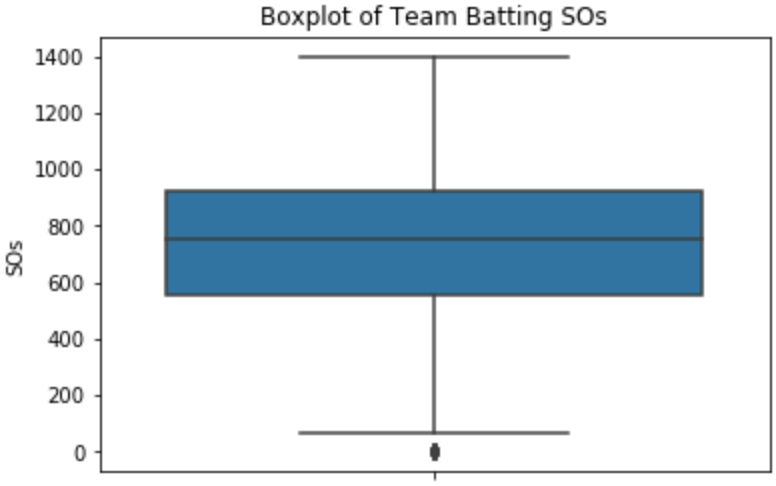

Histogram of Team Batting 3Bs
300
250
200
150
100
50
0
0
50
100
150
200
team_batting_3b

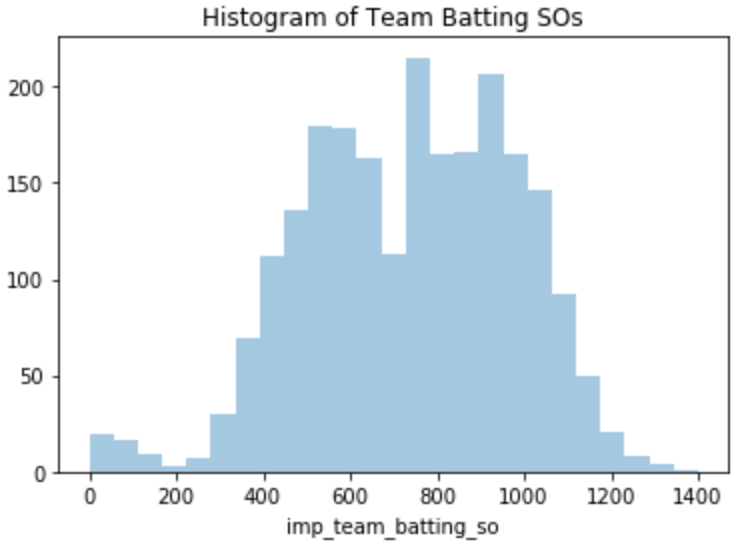

Boxplot of Team Basrunning CSs
200
175
150
125
100
75
50
25
0
CSs

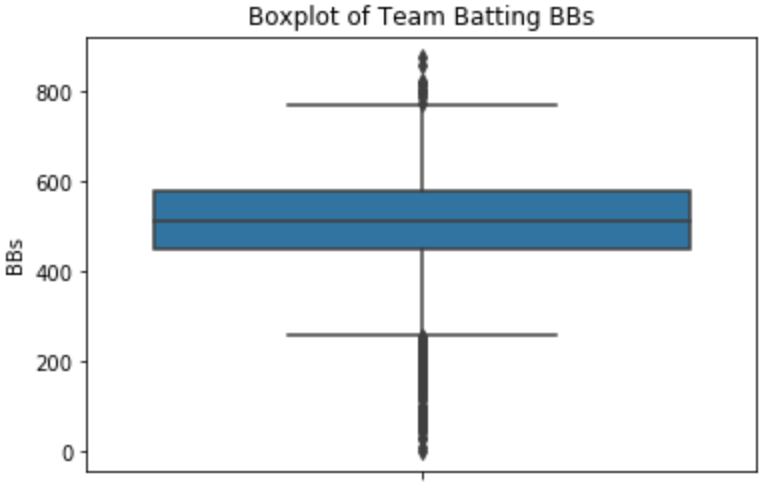

Histogram of Team Basrunning CSs
800
600
400
200
0
0
25
50
75
100
125
150
175
200
imp_team_baserun_cs

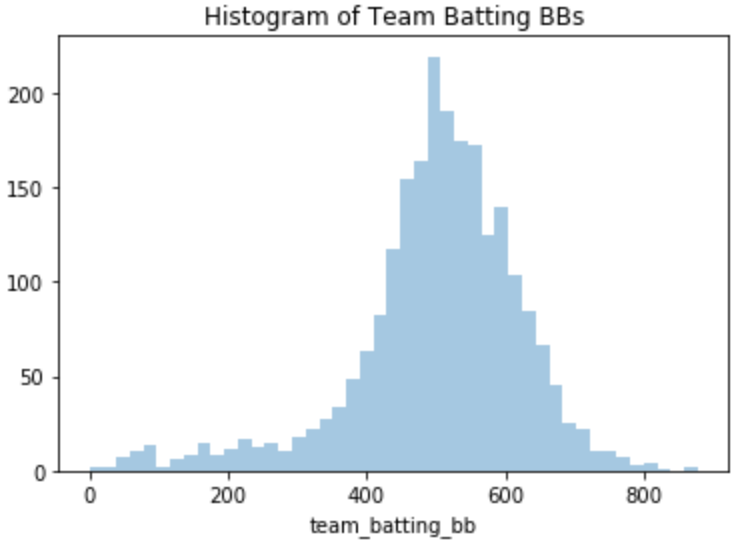

Boxplot of Team Basrunning SBs
SBs
700
600
500
400
300
200
100
0

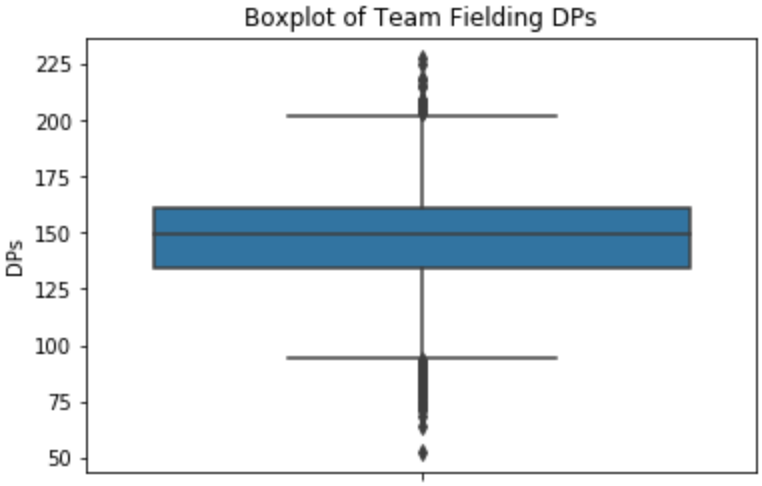

Histogram of Team Basrunning SBs
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
imp_team_baserun_sb

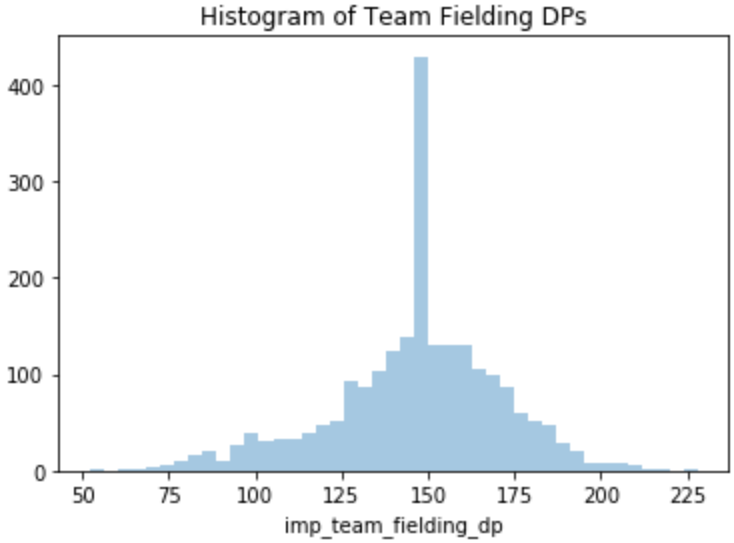

Boxplot of Team Batting SOs
SOs
1400
1200
1000
800
600
400
200
0

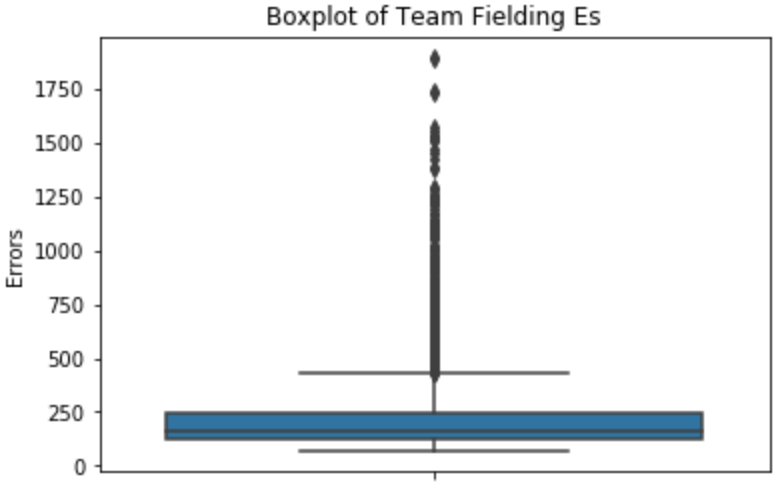

Histogram of Team Batting SOs
200
150
100
50
0
0
200
400
600
800
1000
1200
1400
imp_team_batting_so

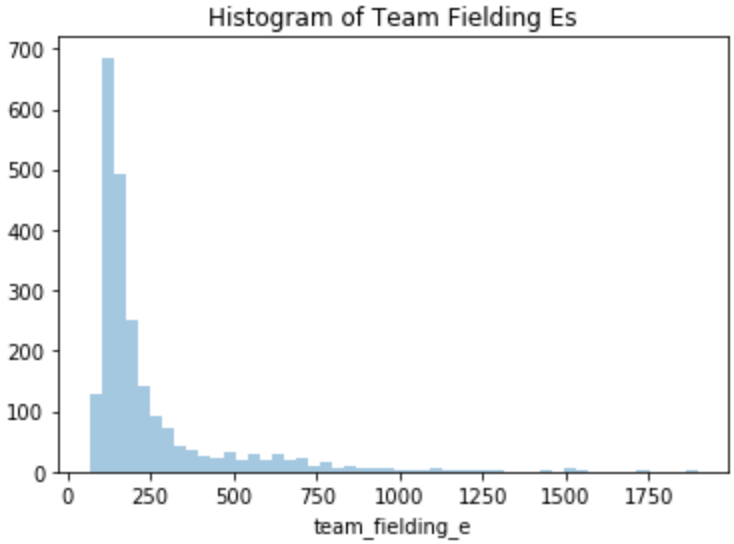

Boxplot of Team Batting BBs
800
600
400
200
0
BBs

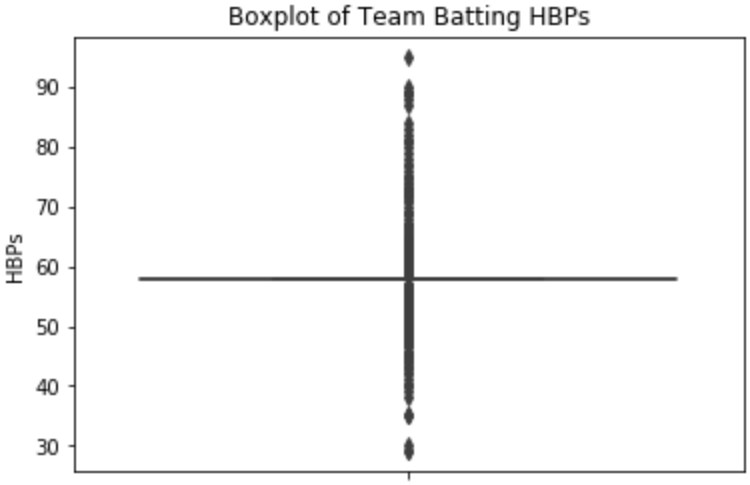

Histogram of Team Batting BBs
200
150
100
50
0
0
200
400
600
800
team_batting_bb

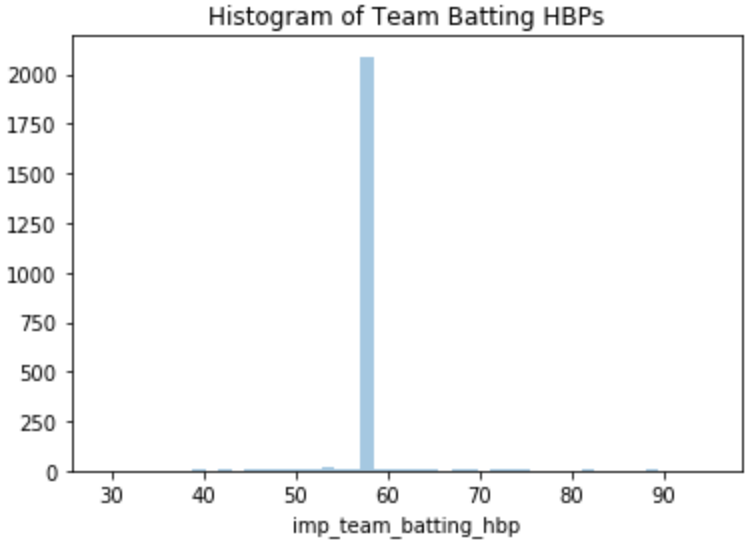

Boxplot of Team Fielding DPs
DPs
225
200
175
150
125
100
75
50

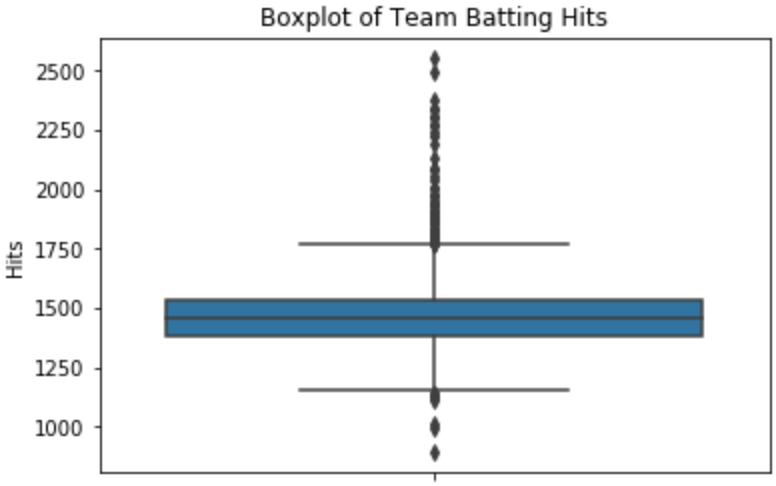

Histogram of Team Fielding DPs
400
300
200
100
0
50
75
100
125
150
175
200
225
imp_team_fielding_dp

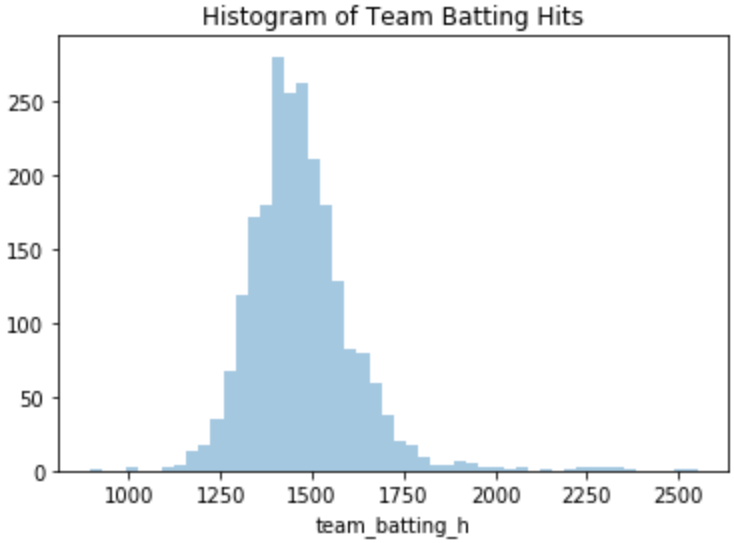

Boxplot of Team Fielding Es
Errors
1750
1500
1250
1000
750
500
250
0

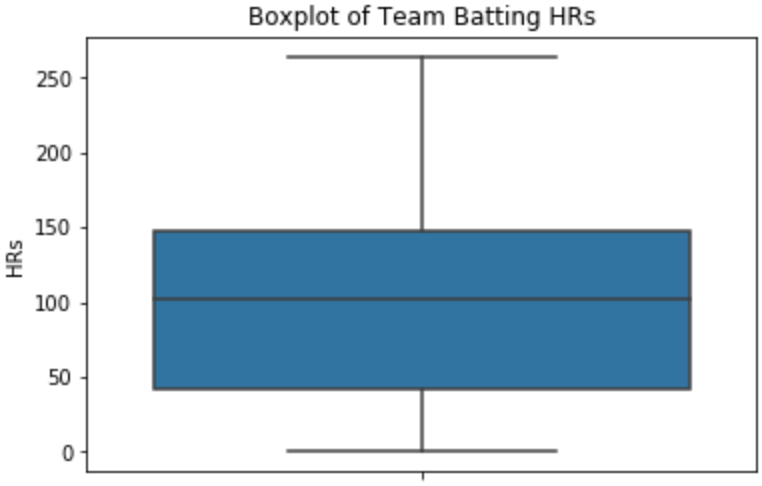

Histogram of Team Fielding Es
700
600
500
400
300
200
100
0
0
250
500
750
1000
1250
1500
1750
team_fielding_e

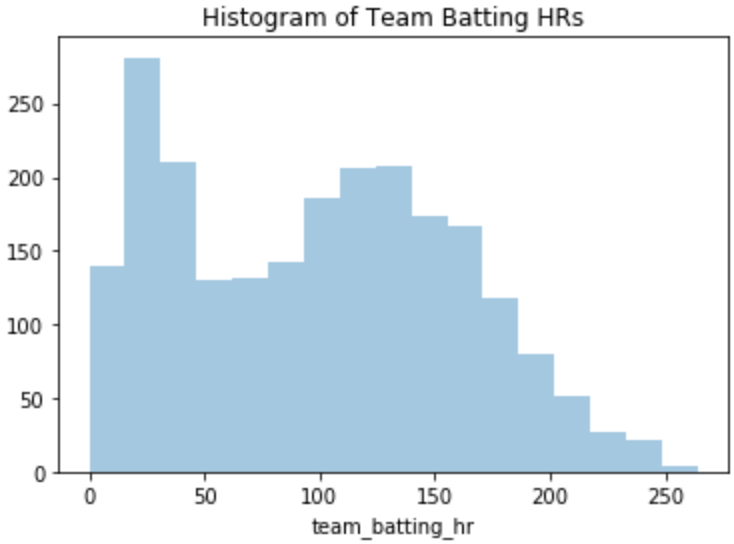

Boxplot of Team Batting HBPs
HBPs
90
80
70
60
50
40
30

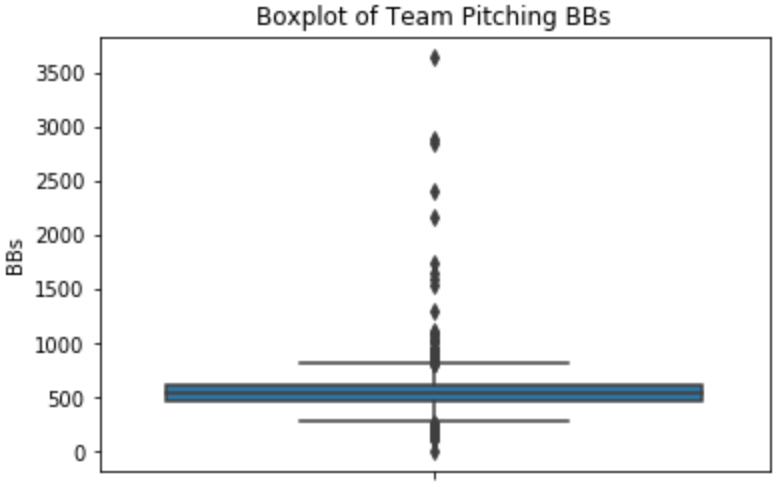

Histogram of Team Batting HBPs
2000
1750
1500
1250
1000
750
500
250
0
30
40
50
60
70
80
90
imp_team_batting_hbp

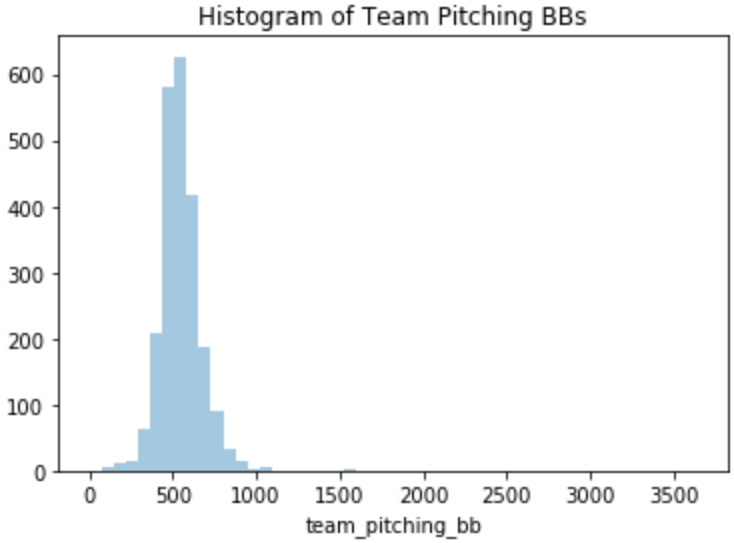

Boxplot of Team Batting Hits
Hits
2500
2250
2000
1750
1500
1250
1000

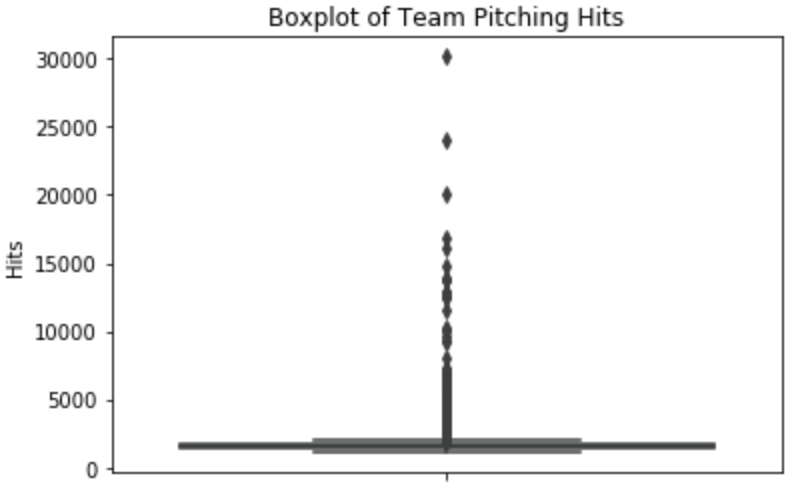

Histogram of Team Batting Hits
250
200
150
100
50
0
1000
1250
1500
1750
2000
2250
2500
team_batting_h

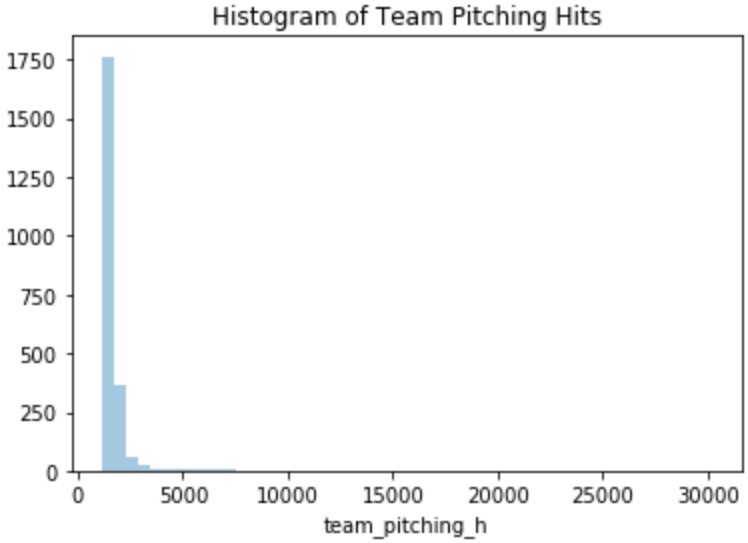

Boxplot of Team Batting HRs
HRs
250
200
150
100
50
0

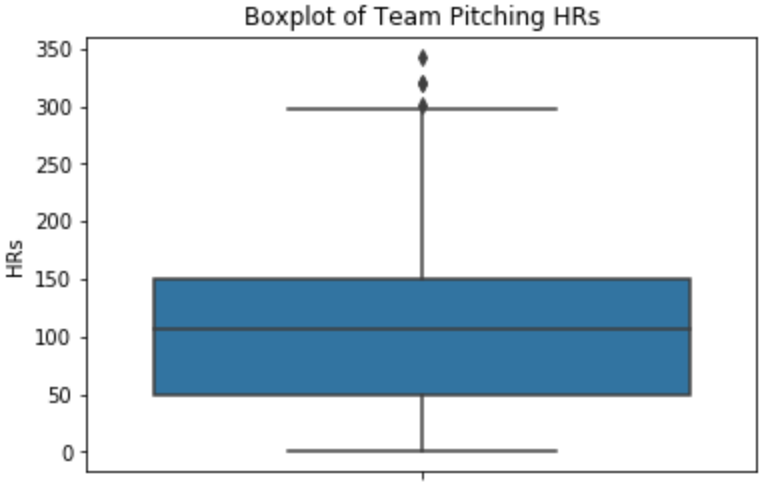

Histogram of Team Batting HRs
250
200
150
100
50
0
0
50
100
150
200
250
team_batting_hr

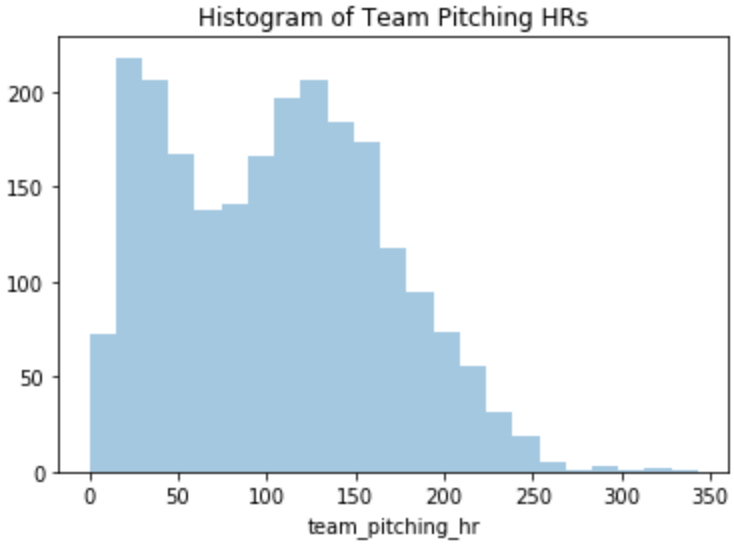

Boxplot of Team Pitching BBs
BBs
3500
3000
2500
2000
1500
1000
500
0

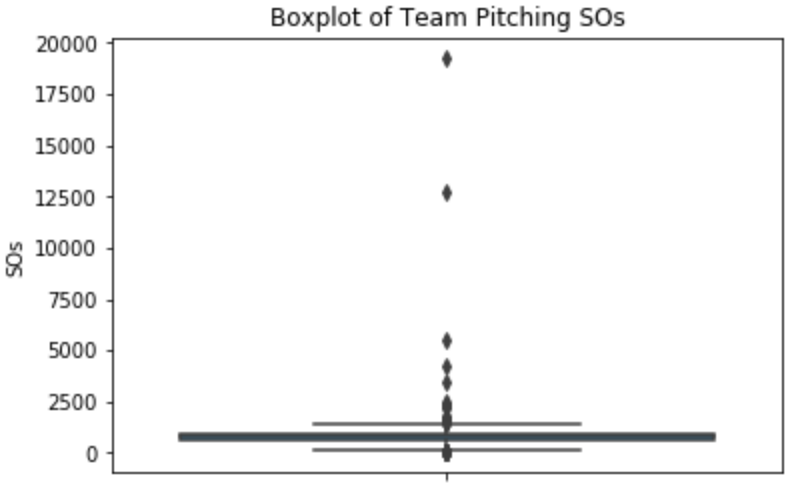

Histogram of Team Pitching BBs
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500
team_pitching_bb

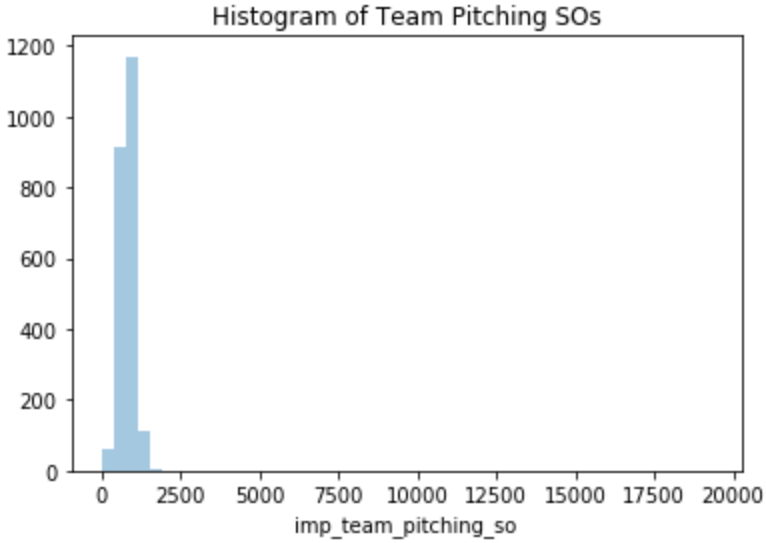

Boxplot of Team Pitching Hits
Hits
30000
25000
20000
15000
10000
5000
0

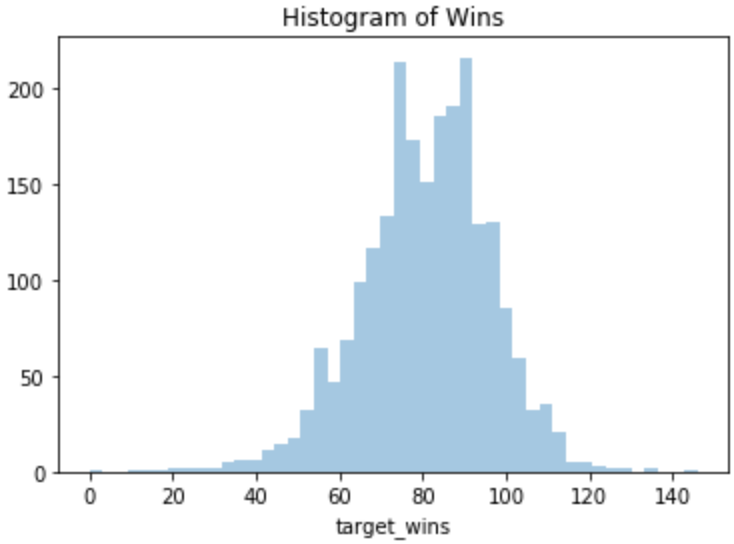

Histogram of Team Pitching Hits
1750
1500
1250
1000
750
500
250
0
0
5000
10000
15000
20000
25000
30000
team_pitching_h

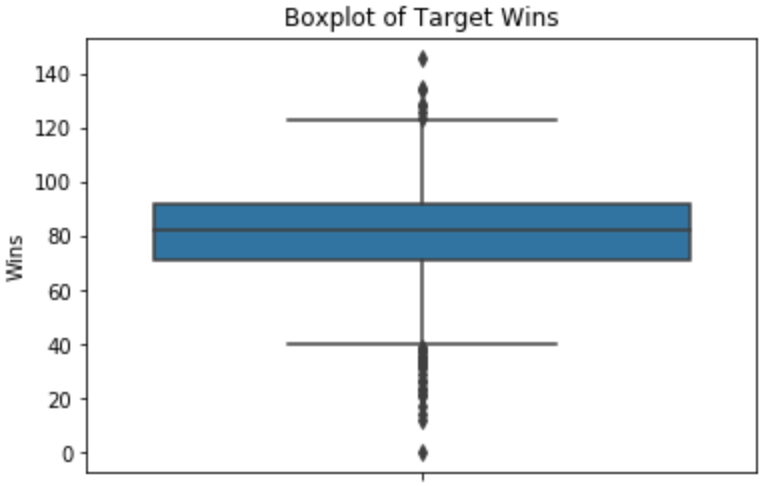

Boxplot of Team Pitching HRs
HRs
350
300
250
200
150
100
50
0


Histogram of Team Pitching HRs
200
150
100
50
0
0
50
100
150
200
250
300
350
team_pitching_hr


Boxplot of Team Pitching SOs
20000
17500
15000
12500
10000
7500
5000
2500
0
SOs


Histogram of Team Pitching SOs
1200
1000
800
600
400
200
0
0
2500
5000
7500
10000
12500
15000
17500
20000
imp_team_pitching_so


Histogram of Wins
200
150
100
50
0
0
20
40
60
80
100
120
140
target_wins


Boxplot of Target Wins
Wins
140
120
100
80
60
40
20
0