

DATA 621 - Homework 1

Joshua Sturm

February 26, 2018

Introduction

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season. Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team.

1. Data Exploration

1.1 Load packages

1.2 Read in data

1.2.1 Create data dictionary

Variable Name	Definition	Theoretical effect
TARGET_WINS	Number of wins	Outcome variable
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_SO	Strikeouts by batter	Negative impact on wins
TEAM_BASERUN_SB	Stolen bases	Positive impact on wins
TEAM_BASERUN_CS	Caught stealing	Negative impact on wins
TEAM_BATTING_HBP	Batters hit by pitch (free base)	Positive impact on wins
TEAM_PITCHING_H	Hits allowed	Negative impact on wins
TEAM_PITCHING_HR	Homeruns allowed	Positive impact on wins
TEAM_PITCHING_BB	Walks allowed	Negative impact on wins
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive impact on wins
TEAM_FIELDING_E	Errors	Negative impact on wins
TEAM_FIELDING_DP	Double plays	Positive impact on wins

1.3 Basic variable statistics

	n	mean	sd	median	min	max	skew	kurtosis	se
INDEX	2276	1268.46353	736.34904	1270.5	1	2535	0.0042149	-1.2167564	15.4346788
TARGET_WINS	2276	80.79086	15.75215	82.0	0	146	-0.3987232	1.0274757	0.3301823
BATTING_H	2276	1469.26977	144.59120	1454.0	891	2554	1.5713335	7.2785261	3.0307891
BATTING_2B	2276	241.24692	46.80141	238.0	69	458	0.2151018	0.0061609	0.9810087
BATTING_3B	2276	55.25000	27.93856	47.0	0	223	1.1094652	1.5032418	0.5856226
BATTING_HR	2276	99.61204	60.54687	102.0	0	264	0.1860421	-0.9631189	1.2691285
BATTING_BB	2276	501.55888	122.67086	512.0	0	878	-1.0257599	2.1828544	2.5713150
BATTING_SO	2174	735.60534	248.52642	750.0	0	1399	-0.2978001	-0.3207992	5.3301912
BASERUN_SB	2145	124.76177	87.79117	101.0	0	697	1.9724140	5.4896754	1.8955584
BASERUN_CS	1504	52.80386	22.95634	49.0	0	201	1.9762180	7.6203818	0.5919414
BATTING_HBP	191	59.35602	12.96712	58.0	29	95	0.3185754	-0.1119828	0.9382681
PITCHING_H	2276	1779.21046	1406.84293	1518.0	1137	30132	10.3295111	141.8396985	29.4889618
PITCHING_HR	2276	105.69859	61.29875	107.0	0	343	0.2877877	-0.6046311	1.2848886
PITCHING_BB	2276	553.00791	166.35736	536.5	0	3645	6.7438995	96.9676398	3.4870317
PITCHING_SO	2174	817.73045	553.08503	813.5	0	19278	22.1745535	671.1891292	11.8621151
FIELDING_E	2276	246.48067	227.77097	159.0	65	1898	2.9904656	10.9702717	4.7743279
FIELDING_DP	1990	146.38794	26.22639	149.0	52	228	-0.3889390	0.1817397	0.5879114

The training data has 2276 cases, with 17 variables.

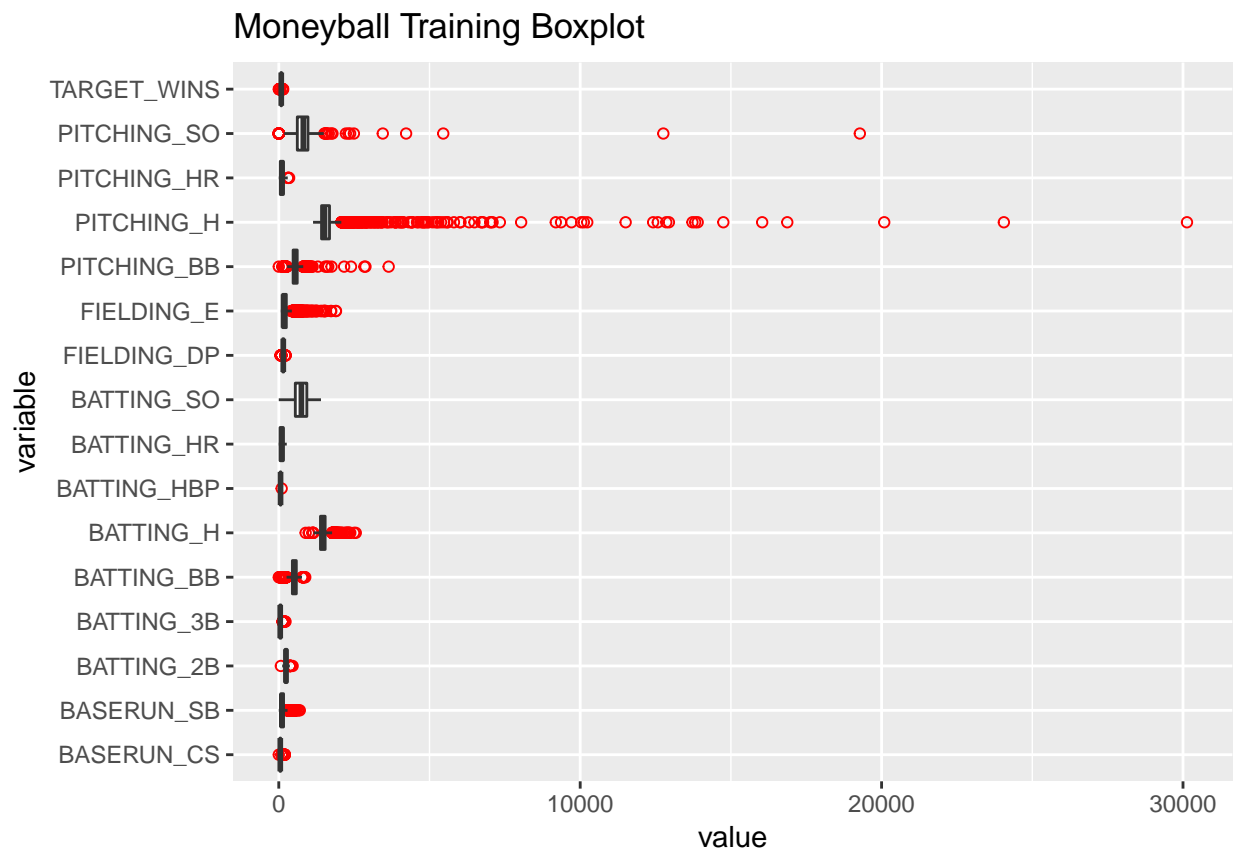
INDEX, as its name suggest, is simply an index, which we can remove.

TARGET_WINS is our response variable, leaving the remaining 15 variables as our predictors.

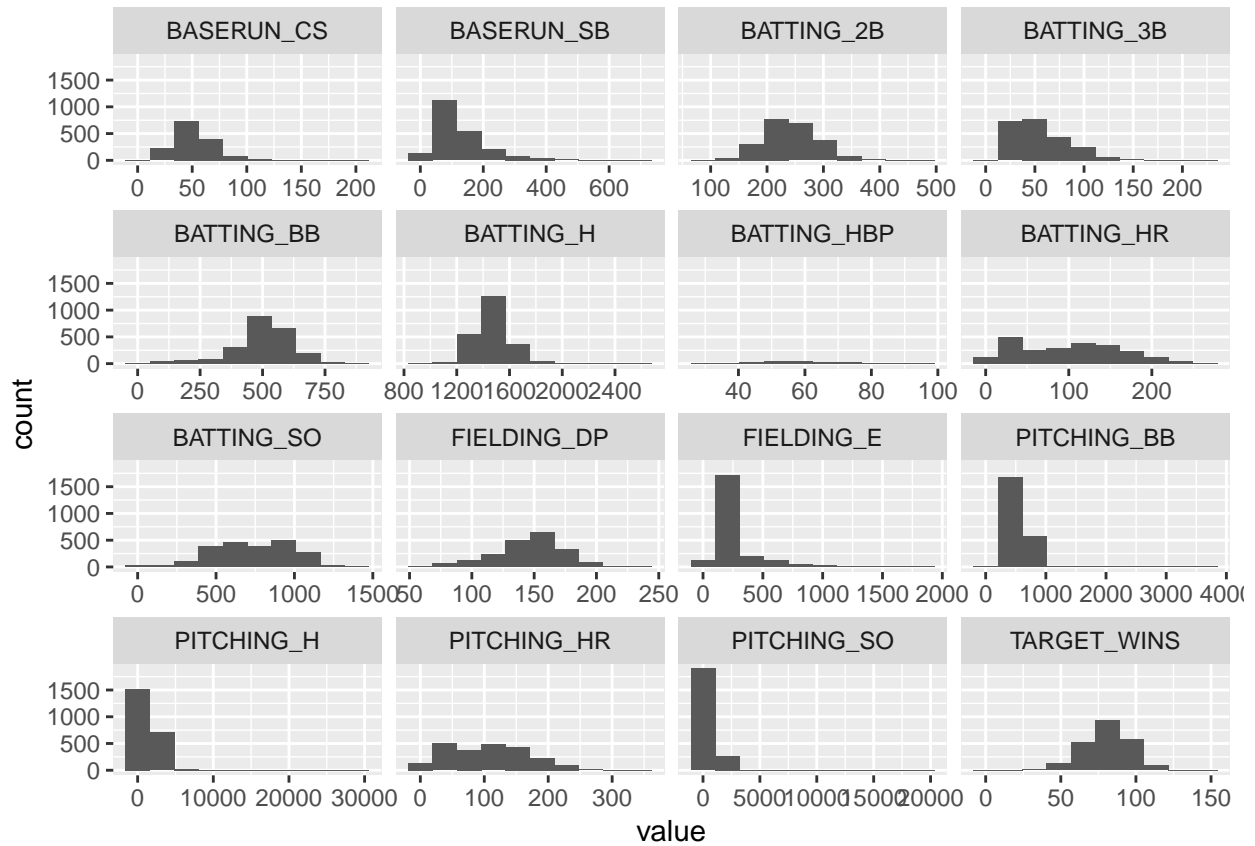
Right off the bat :), we notice the data has some oddities. Some variables are relatively sparse, particularly BASERUN_CS and BATTING_HBP. The latter two have so many missing cases, 772 (34%) and 2085 (92%) respectively, that it would be unreasonable to include them for use in any meaningful statistical model, so they will require further examining in part 3.

1.4 Summary Graphs

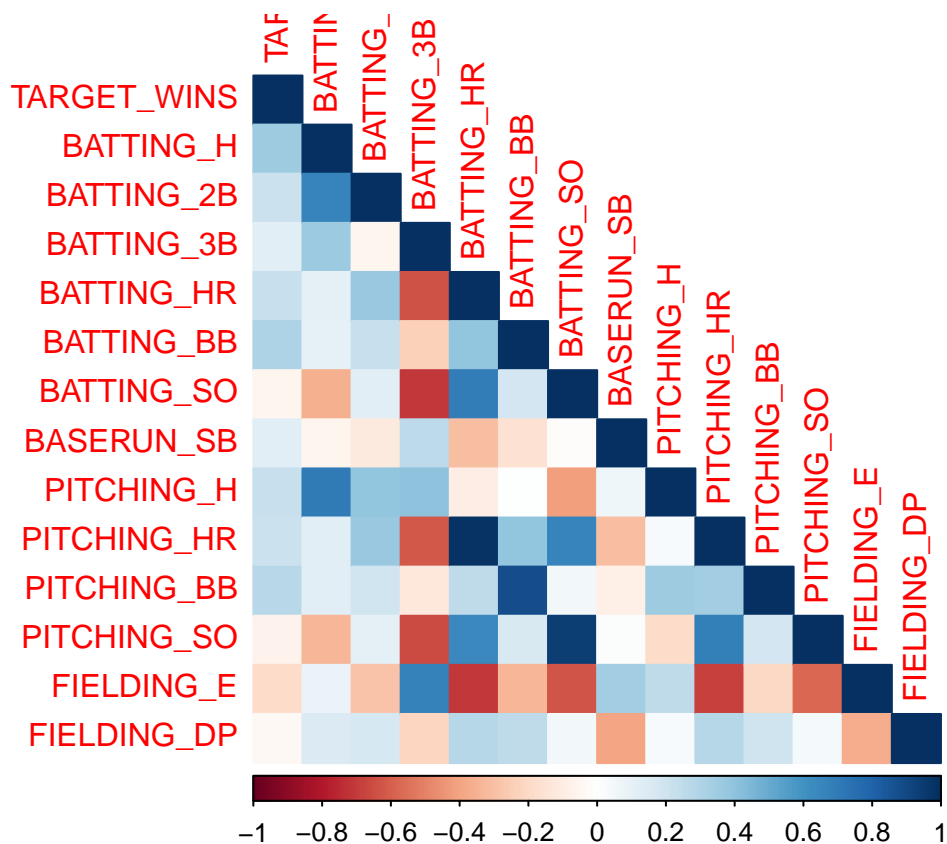
1.4.1 Boxplot



1.4.2 Histogram



1.5 Correlation



2. Data Preparation

2.1 Adding and removing variables

From the above graphs, we notice a few things. Firstly, our response variable, `INDEX`, is approximately normal.

Many of the predicting variables, however, are far from normal. I'll address this in a few ways. Firstly, as can be seen in the data dictionary, there are entries for `doubles`, `triples`, `homeruns`, and an all-encompassing `base hits`. Notably missing is a variable for singles. I can create my own by simply taking the difference between `base hits` and all three present variables, i.e. `singles = base hits - doubles - triples - homeruns`. Once I've created my `singles` variable, I'll need to remove `base hits` from the model to prevent multicollinearity.

Another issue is the strong correlation between `PITCHING_HR` and `BATTING_HR` (97%), which makes sense - they're essentially the same thing from opposite viewpoints. Since these variables are practically the same as each other, we can safely drop one of them.

As we mentioned in part 1, we're removing the two variables with a significant amount of missing data - `BATTING_HBP` and `BASERUN_CS`.

Additionally, `FIELDING_DP` is missing 12.5% of its total cases, so we will drop it. The remaining variables have less than 6% cases missing, so we'll attempt to model with those in place.

2.2 Missing data imputation

While we removed the variables that were missing a large portion of data, we're still left with others that have NA values.

BATTING_SO, BASERUN_SB, and PITCHING_SO have an average of ~ 112 missing cases.

After reading up on dealing with missing data in linear regression, it seems that imputation by means of regression is preferred over a basic statistical method such as mean, for example. To handle this, I'll make use of the `Hmisc` package.

2.3 Outliers

I'll begin by printing out the range for each variable, to see if I can identify any outliers based on the variable's extremes.

TARGET_WINS	BATTING_2B	BATTING_3B	BATTING_HR	BATTING_BB	BASERUN_SB	PITCHING_H	PITCHING_BB	PITCHING_SO	FIELDING_E	BATTING_1B	BATTING_SO
0	69	0	0	0	0	1137	0	0	65	709	0
146	458	223	264	878	697	30132	3645	19278	1898	2112	1399

When reviewing this table, together with the plots generated in part 1, one can instantly notice oddities in the data. Going in order, I'll begin with **BASERUN_SB**. The team with the most stolen bases in a season was the 1887 St. Louis Cardinals with 581. I'll remove the `sum(mb3$BASERUN_SB > 581)` rows which are greater than 581.

MLB statistics show the 1915 Philadelphia Athletics to have the most walks in a season, with 827. That works out to ~859 in a full 162-game season. Interestingly, removing the outliers for this variable actually makes the model *less* accurate; nevertheless, in the interest of staying consistent, I will discard any row with a value greater than 859.

Next is **PITCHING_H**. I couldn't find any data on most hits allowed, but according to this list on wikipedia, the most hits by a team in a season was 1,783, by the Phillies in 1930. This raises questions about the original dataset's max of **BATTING_H** of 2,554. If a team allowed 30,132 hits in a single season, that's an average of 186 hits per game... which is impossible. The record of 1,783 comes out to 11 hits per game. If they played a full 162-game season, it would come out to 1,874.

Since such a large portion of this variable is in outlier territory, it leads me to believe that it would be a mistake to discard the whole thing. However, I'm not exactly sure how to deal with it; statistical transformations, such as square root or log, have done little to rectify the issue. I will resort to removing the 320 rows with values larger than 1,874.

The next variable of concern is **PITCHING_SO**. According to official statistics kept by MLB (mlb.com), the team with the most (pitching) strikeouts in a season was the 2017 Cleveland Indians, with a staggering total of 1,614 strikeouts. Our data, however, has a max of 19278, which is several orders of magnitude larger than reality. So I will remove the 12 rows larger than 1,614.

The third variable of concern is **FIELDING_E**. The team with the most fielding errors in a season was the ~~1886 Washington Nationals, with a total of 867~~ 1883 Baltimore Orioles, with 517. Since there were only 98 games played then, that works out to ~854 in a full 162-game season. Thus, I will remove any row with more errors than 854. This is also an interesting variable, in that removing the outliers lowers the model's accuracy.

3. Build models

3.1 Model 1

I guess it makes the most sense to begin with a full model with all (non-transformed) variables included.

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = mb.tr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0626  -5.4196  -0.0423   5.2111  22.9355
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.4562317  19.7385030   3.063  0.00254 **
## INDEX        -0.0002478   0.0008508  -0.291  0.77122
## BATTING_H      1.8111103   2.7908648   0.649  0.51723
## BATTING_2B     0.0267462   0.0303941   0.880  0.38008
## BATTING_3B    -0.1018043   0.0777401  -1.310  0.19208
## BATTING_HR    -4.6100155  10.5666083  -0.436  0.66317
## BATTING_BB    -4.4606275   3.6457882  -1.224  0.22279
## BATTING_SO     0.4303282   2.6231874   0.164  0.86988
## BASERUN_SB     0.0335937   0.0288100   1.166  0.24519
## BASERUN_CS    -0.0130338   0.0719436  -0.181  0.85645
## BATTING_HBP    0.0837038   0.0499097   1.677  0.09532 .
## PITCHING_H    -1.7887761   2.7903398  -0.641  0.52233
## PITCHING_HR    4.6958245  10.5649821   0.444  0.65725
## PITCHING_BB    4.5120283   3.6432611   1.238  0.21721
## PITCHING_SO   -0.4618971   2.6214432  -0.176  0.86034
## FIELDING_E    -0.1724513   0.0415365  -4.152 5.16e-05 ***
## FIELDING_DP   -0.1063200   0.0371964  -2.858  0.00478 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.489 on 174 degrees of freedom
## (2085 observations deleted due to missingness)
## Multiple R-squared:  0.5503, Adjusted R-squared:  0.509
## F-statistic: 13.31 on 16 and 174 DF,  p-value: < 2.2e-16
```

Not a very robust model; many insignificant variables, a low R^2 value, and a high AIC and BIC. There are also a lot of weird coefficients in this model. Triples, *home runs*, walks, (pitching) strikeouts, and double plays all have negative coefficients, meaning they are negatively correlated with the model.

3.2 Model 2

The second model I'll try contains the dataset with the added and removed variables, but none of them were transformed.

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = mb2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -44.840  -7.961   0.168   7.573  50.911
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 17.7010995  5.1280679   3.452 0.000568 ***
## BATTING_2B  -0.0081692  0.0071957  -1.135 0.256385
## BATTING_3B   0.1205824  0.0165717   7.276 4.87e-13 ***
## BATTING_HR   0.1201699  0.0083181  14.447 < 2e-16 ***
## BATTING_BB   0.0255959  0.0064707   3.956 7.90e-05 ***
## BATTING_SO  -0.0081180  0.0050803  -1.598 0.110213
## BASERUN_SB   0.0585237  0.0042963  13.622 < 2e-16 ***
## PITCHING_H   0.0024770  0.0004047   6.120 1.12e-09 ***
## PITCHING_BB -0.0027189  0.0051155  -0.531 0.595139
## PITCHING_SO -0.0040271  0.0044839  -0.898 0.369221
## FIELDING_E  -0.0499358  0.0032267 -15.476 < 2e-16 ***
## BATTING_1B   0.0396067  0.0037802  10.477 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.63 on 2031 degrees of freedom
## (233 observations deleted due to missingness)
## Multiple R-squared:  0.3611, Adjusted R-squared:  0.3577
## F-statistic: 104.4 on 11 and 2031 DF,  p-value: < 2.2e-16
```

This model has more sensible coefficients, even though it scored lower (R^2).

3.3 Model 3

The third model will be the modified dataset with all the transformed variables.

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = mb3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -37.163  -7.661   0.146   7.627  38.457
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.746279   5.765368   5.333 1.08e-07 ***
## BATTING_2B  -0.080753   0.019091  -4.230 2.45e-05 ***
## BATTING_3B   0.148211   0.024128   6.143 9.82e-10 ***
## BATTING_HR   0.045947   0.019711   2.331 0.019853 *
## BATTING_BB   0.204607   0.050556   4.047 5.39e-05 ***
## BASERUN_SB   0.089149   0.004888  18.240 < 2e-16 ***
## PITCHING_H   0.060620   0.016882   3.591 0.000338 ***
## PITCHING_BB -0.165753   0.047778  -3.469 0.000533 ***
## PITCHING_SO  0.015239   0.008604   1.771 0.076693 .
## FIELDING_E  -0.087109   0.004906 -17.757 < 2e-16 ***
## BATTING_1B  -0.033850   0.018148  -1.865 0.062293 .
## BATTING_SO  -0.028905   0.008846  -3.267 0.001104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.08 on 1940 degrees of freedom
## Multiple R-squared:  0.3794, Adjusted R-squared:  0.3759
## F-statistic: 107.8 on 11 and 1940 DF,  p-value: < 2.2e-16
```


The transformed model further improves on the second, with respect to the coefficients. PITCHING_H is positive instead of negative, while BATTING_1B and BATTING_2B are negative.

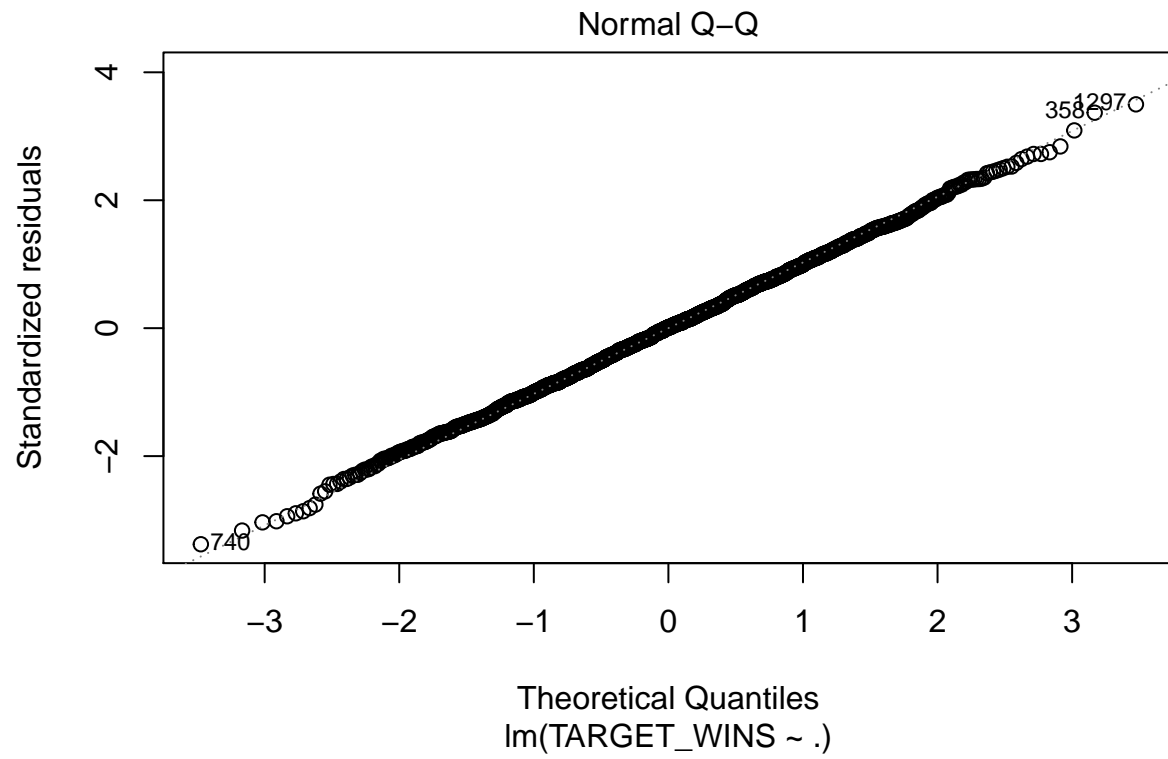
4. Model selection

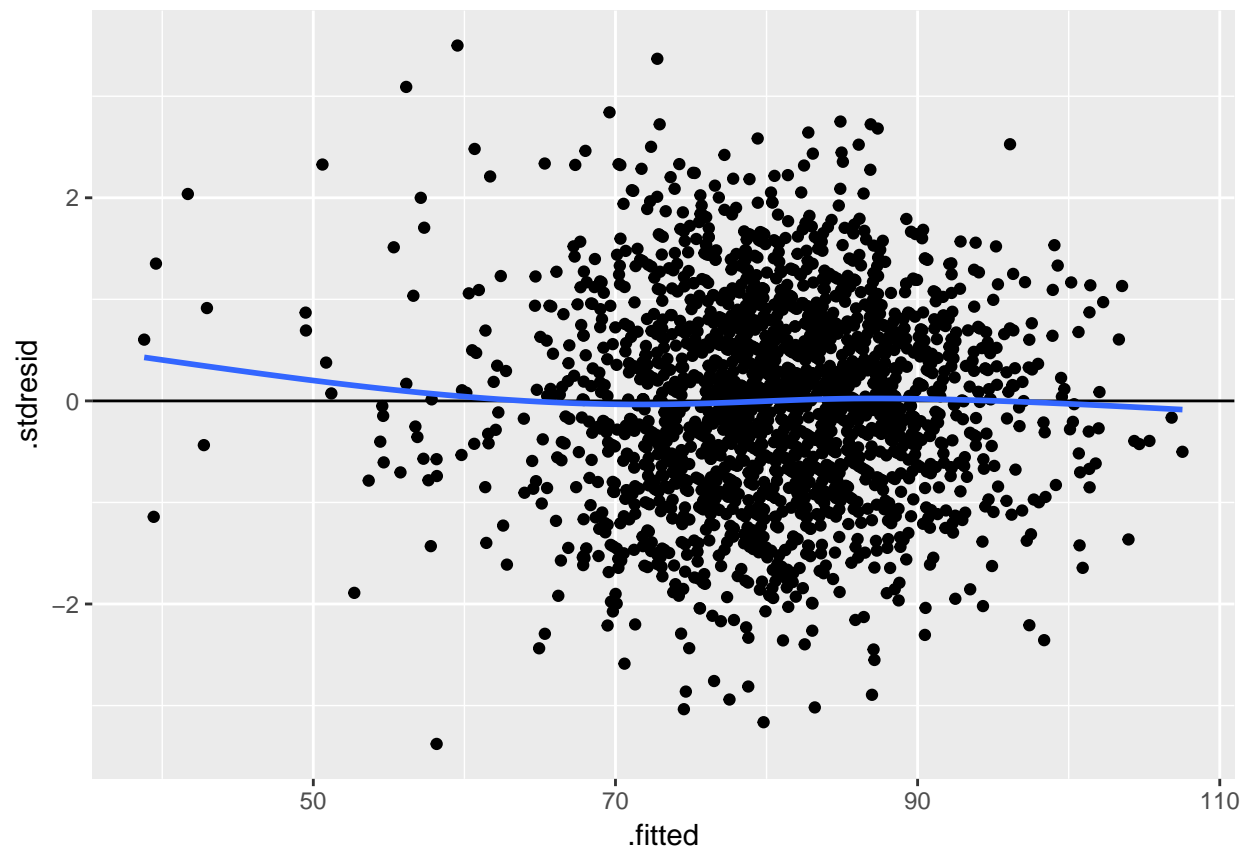
To ensure uniformity between the datasets, I performed the same transformations on the evaluation set.

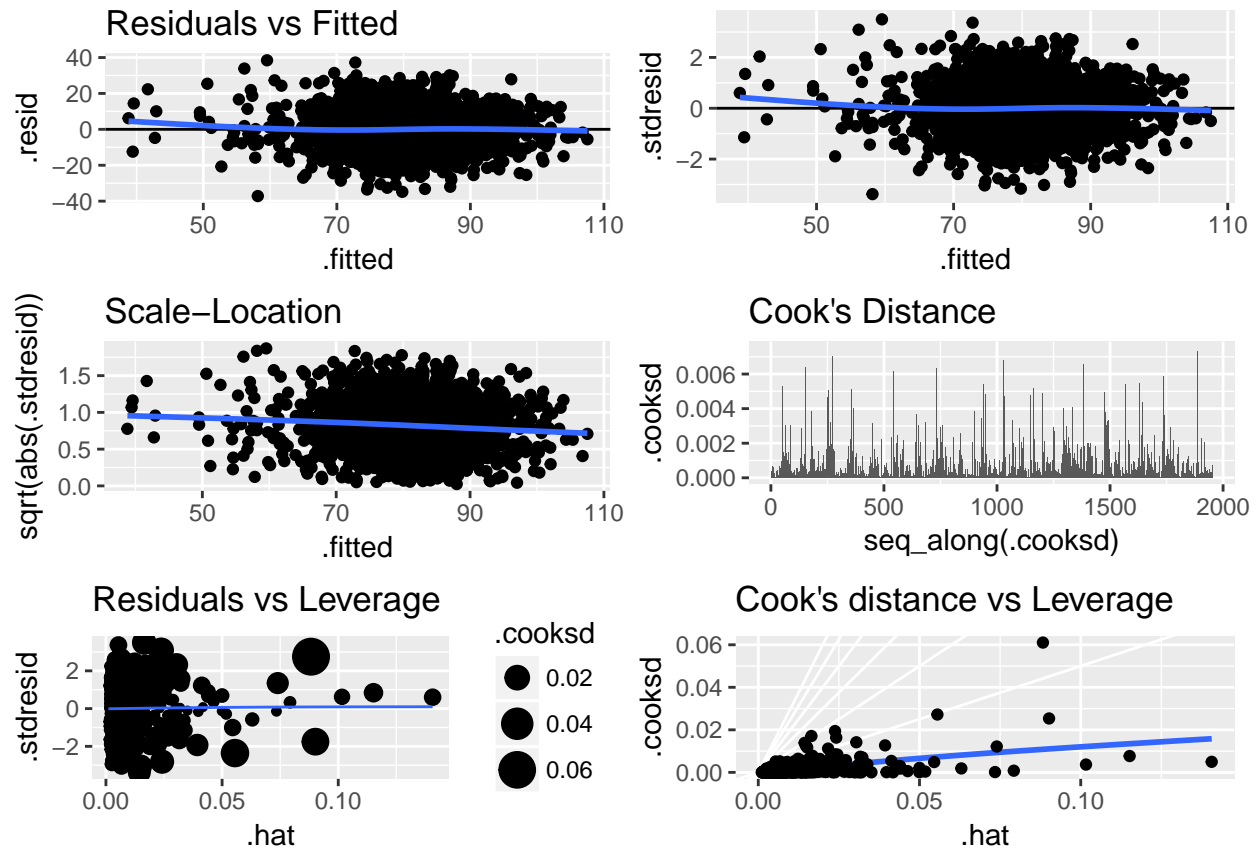
Here is a sample of the model's results:

actuals	predicted	error	percerror
70	63.70946	-6.290544	-8.99%
86	69.05946	-16.940545	-19.7%
70	72.50420	2.504199	3.58%
82	84.20480	2.204797	2.69%
75	76.53630	1.536295	2.05%
80	70.07014	-9.929861	-12.41%

```
## [1] "The mean error is: 1.78783769655456"
```







5. Closing words

I picked the third model, because I felt the data was the most honest - that is, it had the fewest outliers, and was closer to what it was meant to represent. The full model had a higher R^2 , but that could be attributed to collinearity amongst the variables. The model is not perfect, and I likely would not use it in practice, but, given the many issues with the data, I believe I optimized it as best I could.

References

- <https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/>
- <https://www.baseball-reference.com>
- <https://www.baseball-almanac.com>
- <https://www.mlb.com>
- <https://sports.stackexchange.com/questions/16246/what-is-the-mlb-record-for-most-errors-by-one-team-in-one-season-d>
- <http://r-statistics.co/Linear-Regression.html>
- <http://ggplot2.tidyverse.org/reference/fortify.lm.html>