CUNY SPS DATA 621 - CTG5 - HW1

Code

Betsy Rosalen, Gabrielle Bartomeo, Jeremy O'Brien, Lidiia Tronina, Rose Koh

February 27, 2019

---

1 DATA EXPLORATION

Professionals and gamblers alike are always seeking to optimize their chances of winning, whether it be sports, games, or their bets on them. Major League Baseball is a multibillion dollar industry (https://www.forbes.com/sites/mikeozanian/2018/04/11/baseball-team-values-2018/#4675cfd43fc0) where individual teams, players, and those who profit off of their success stand to benefit most from such optimization.

Data from 1871 to 2006 was collected in order to infer how many wins could be expected from the 162 games in a baseball team's season. Each observation represents a season for an unnamed team, and we have a total of 2,276 observations. For each team the target variable, `TARGET_WINS`, represents the number of wins in a given year and has a maximum value of 162 possible wins. In addition to that 15 continuous integer predictor variables were collected (not including the index) representing each team's: base hits, doubles, triples, homeruns, walks, and strikeouts by batters, batters hit by pitches, bases stolen by batters and the number of times they were caught stealing, the number of errors, double plays, walks, hits, and homeruns allowed, and strikeouts by pitchers. The testing data contains the same 15 predictor variables and no target variable so it will be impossible to check the accuracy of our predictions from the testing data.

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT ON WINS
TARGET_WINS	Number of wins	outcome variable

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT ON WINS
BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact
BATTING_2B	Doubles by batters (2B)	Positive Impact
BATTING_3B	Triples by batters (3B)	Positive Impact
BATTING_HR	Homeruns by batters (4B)	Positive Impact
BATTING_BB	Walks by batters	Positive Impact
BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact
BATTING_SO	Strikeouts by batters	Negative Impact
BASERUN_SB	Stolen bases	Positive Impact
BASERUN_CS	Caught stealing	Negative Impact
FIELDING_E	Errors	Negative Impact
FIELDING_DP	Double Plays	Positive Impact
PITCHING_BB	Walks allowed	Negative Impact
PITCHING_H	Hits allowed	Negative Impact
PITCHING_HR	Homeruns allowed	Negative Impact
PITCHING_SO	Strikeouts by pitchers	Positive Impact

1.1 Summary Statistics

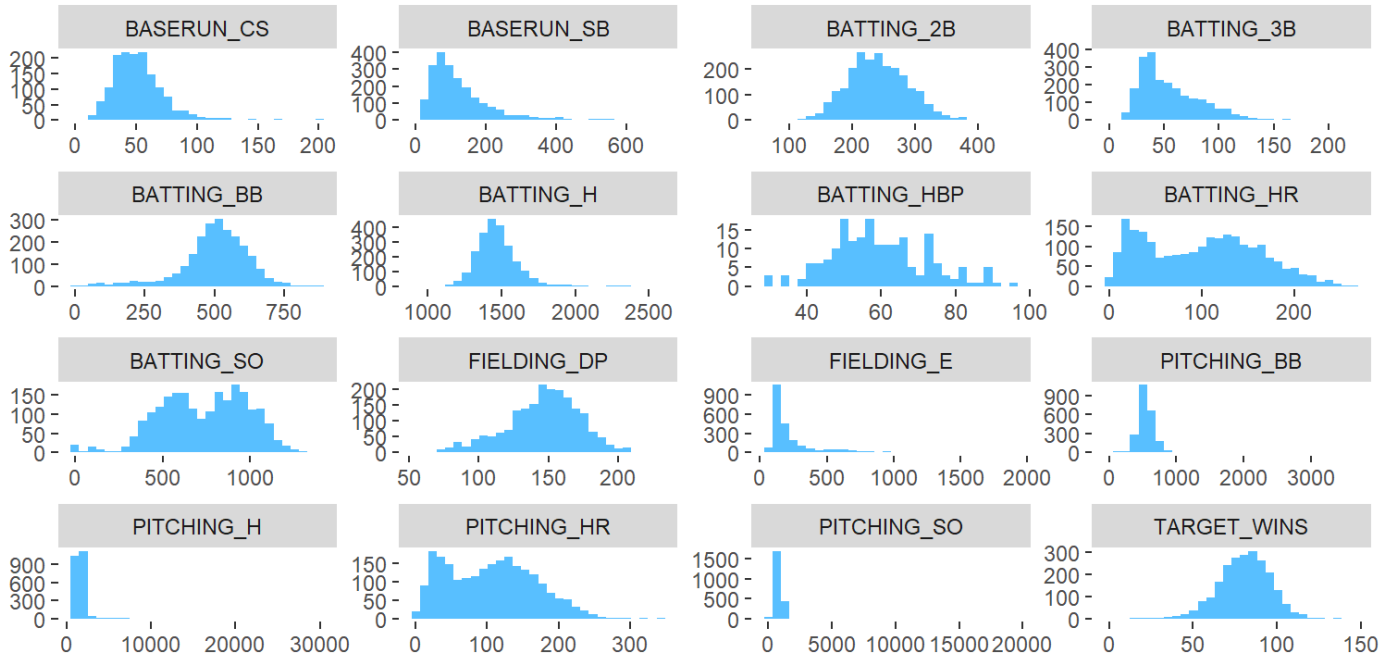
```
##                n  min    mean median   max      sd
## TARGET_WINS 2276    0   80.79   82.0   146   15.75
## BATTING_H   2276  891 1469.27 1454.0  2554  144.59
## BATTING_2B  2276   69  241.25  238.0   458   46.80
## BATTING_3B  2276    0   55.25   47.0   223   27.94
## BATTING_HR  2276    0   99.61  102.0   264   60.55
## BATTING_BB  2276    0  501.56  512.0   878  122.67
## BATTING_SO  2174    0  735.61  750.0  1399  248.53
## BASERUN_SB  2145    0  124.76  101.0   697   87.79
## BASERUN_CS  1504    0   52.80   49.0   201   22.96
## BATTING_HBP  191   29   59.36   58.0    95   12.97
## PITCHING_H  2276 1137 1779.21 1518.0 30132 1406.84
## PITCHING_HR 2276    0  105.70  107.0   343   61.30
## PITCHING_BB 2276    0  553.01  536.5  3645  166.36
## PITCHING_SO 2174    0  817.73  813.5 19278  553.09
## FIELDING_E  2276   65  246.48  159.0  1898  227.77
## FIELDING_DP 1990   52  146.39  149.0   228   26.23
```

Looking at the above, it can be easily noted that there are outliers present in more than one variable, with `PITCHING_H` being the worst offender. Even at three times the standard deviation, its maximum value lays far outiside of the 68-95-99.7 rule. `FIELDING_E`, on the other hand, has the curious case of having a large difference

between its mean and median, indicating there is skew present in this variable as well before any charts are actively looked at. Skewed variables cause bias in linear models and need treatment before being used.

1.2 Shape of Predictor Distributions

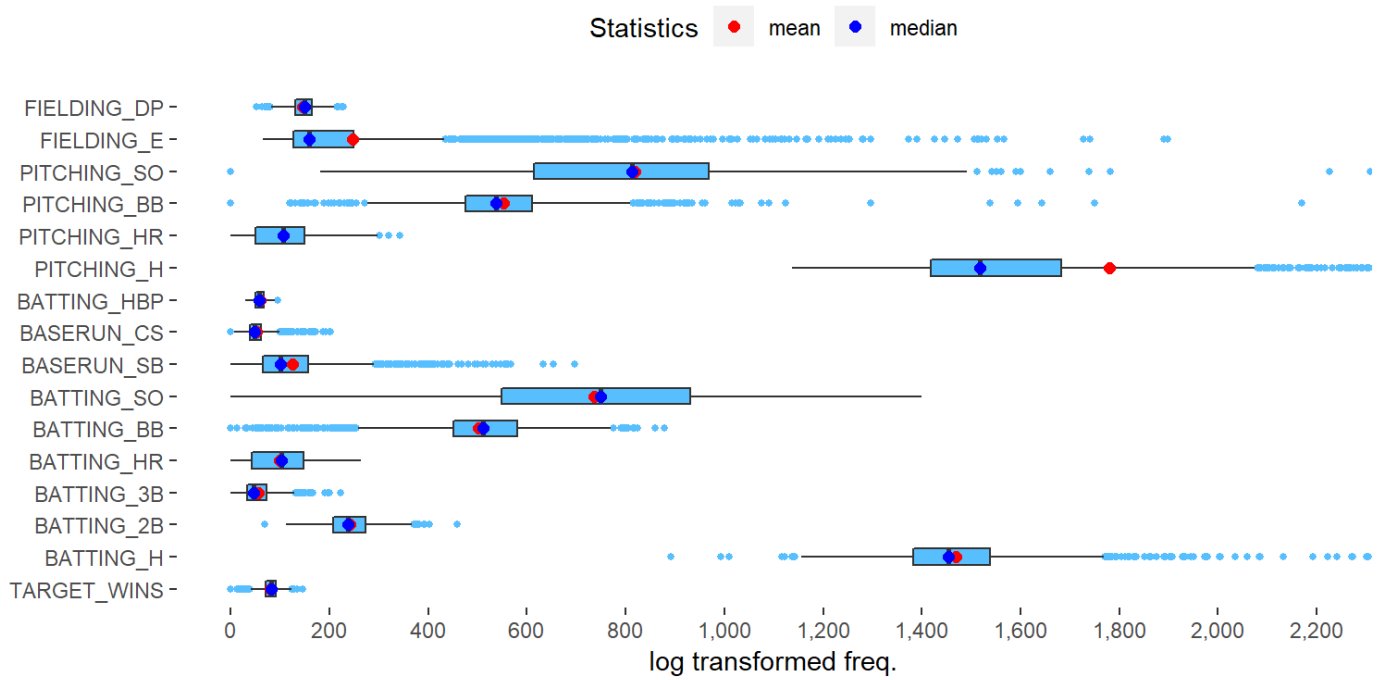
The distribution of most of the variables seems normal although `BASERUN_SB`, `BASERUN_CS`, and `BATTING_3B` have a slight to moderate right skew, `FIELDING_E`, `PITCHING_BB`, `PITCHING_H`, and `PITCHING_SO` have an extreme right skew, and `BATTING_HR`, `BATTING_SO`, and `PITCHING_HR` are bimodal. As a result some data transformation will most likely be necessary to improve the accuracy of our model. The standard deviation of the various variables also hints at the intense skewing of some of the variables.



Data Distributions

1.3 Outliers

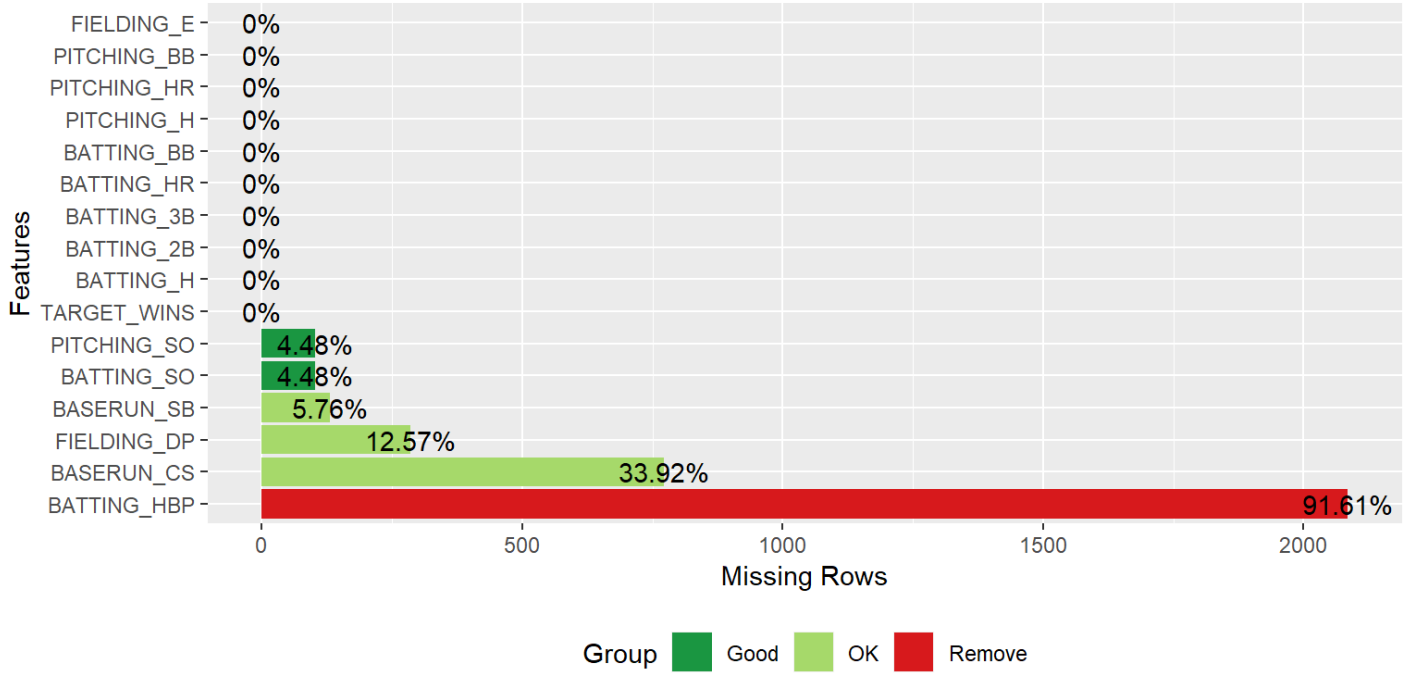
There are also a large number of outliers that need to be accounted for, most prevalently in `FIELDING_E` and `BATTING_H` based off of the boxplots below. One such extreme outlier removed implied that there were, on average per game in a single season, 186 hits allowed by pitchers. This is an unrealistic figure, even for those for whom baseball is outside of their realm of understanding.



Boxplots highlighting many outliers in the data.

1.4 Missing Values

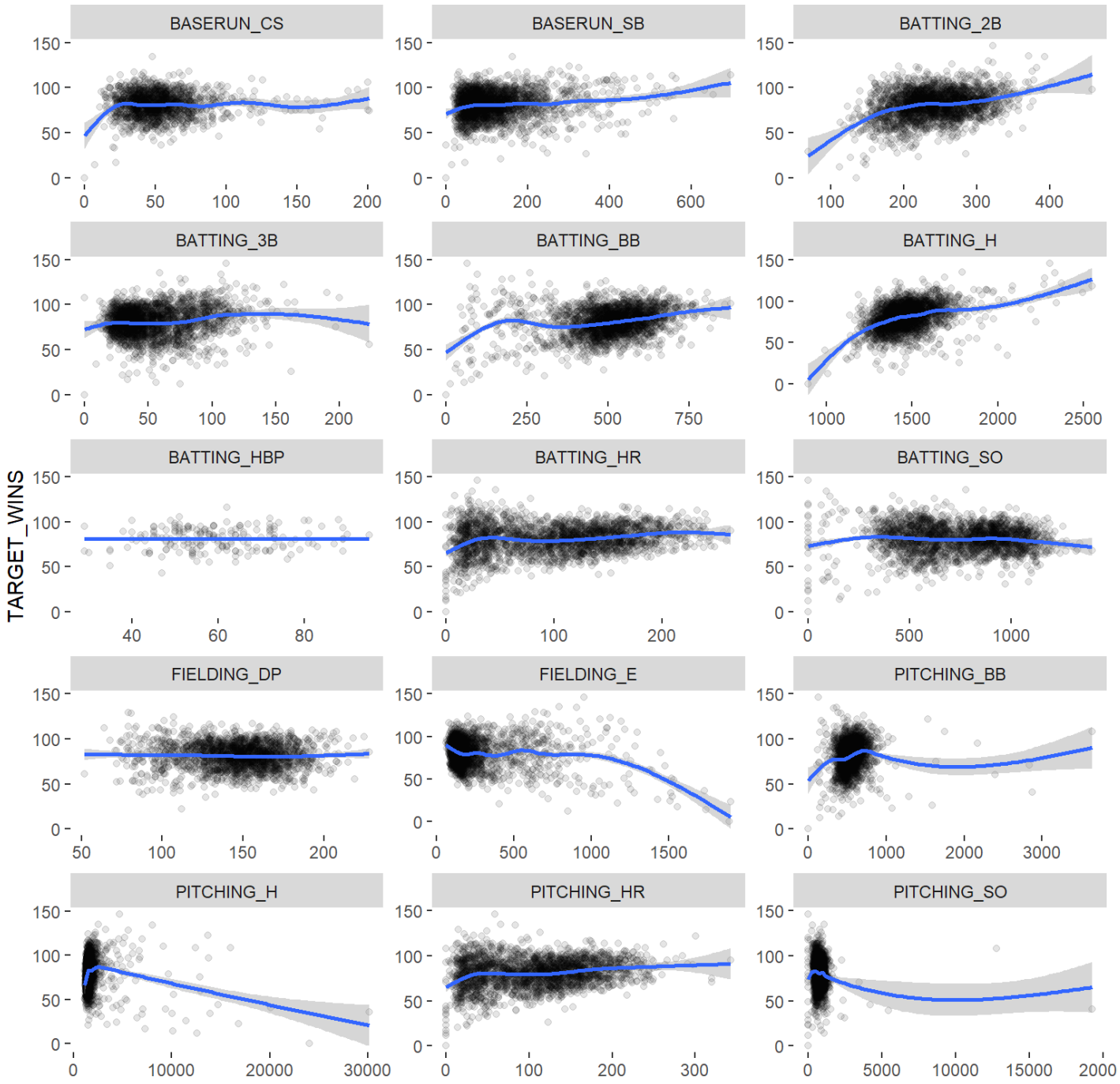
Of all the observations gathered across these fifteen variables, there are 3,478 missing values out of 36,416 total data points, which represents 10.187% of the data. Batters hit by pitches was missing the most, with 2,085 instances of missing information, which represents 91.61% of that variable missing. Additionally `Pitching_SO` and `Batting_SO` are missing exact same proportion 4.48% and are missing in the same observations. This data may not be missing at random and so there may be cause for removing it.



Missing values

1.5 Linearity

Each variable was plotted against the target variable in order to determine at a glance which had the most potential linearity before the dataset was modified.



Linear relationships between each predictors and the target

As can be observed, the most influential variables are the ones previously discussed to have severe outliers and skew, and their linear relationship is negative - the higher the variable, the lower the target wins. On the other hand, `BATTING_H`, `BATTING_BB` and `BATTING_2B` showed the most promise.

---

2 DATA PREPARATION

2.1 Missing Values

As previously mentioned, just north of 10% of the data was missing values. Missing values can lead to errors in a model, bias, and worse if left unaccounted for. Attempting to “fix” this by imputing values or guessing why the values are missing in the first place - such as concluding that the missing values are meant to be zeroes - are just as likely to help with creating a model as it is to help with creating a disaster.

One of the R packages utilized, DataExplorer, which was used for the chart in the “DATA EXPLORATION” section above, recommends removing null or missing values above a certain threshold as indicated in the graph.

Fixing missing values with imputation may help, but can also have a negative impact on the model if the assumed values do not correspond to the actual missing values. When it is just a few observations missing, modifications can be made, however, 91.61% is too large a proportion and would almost definitely distort the model, so we decided it was better to remove the `BATTING_HBP` column altogether. Deleting all cases with missing values, in this instance, would have shrunk the size of the dataset down to less than a tenth of its original size. If we simply delete all cases with missing values from the analysis, we will cause no bias, but we would most certainly lose a lot of important information.

Data that is Missing Completely at Random (MCAR), meaning the probability that a value is missing is the same for all cases can be imputed. Although there is some concern about whether or not `Pitching_SO` and `Batting_SO` are MCAR, we chose to leave all the remaining variables except `BATTING_HBP` and determine whether or not to remove themn during the modelling process.

2.1.1 NA Imputation

To deal with the remaining missing values, the bag imputation method was used via the caret package. A set of dummy variables were created and were used to predict the various values in the dataset. This dummy-set was then pre-processed and used against itself to predict the missing values.

```
##                n  min    mean median   max      sd
## TARGET_WINS 2276    0   80.79   82.0   146   15.75
## BATTING_H   2276  891 1469.27 1454.0  2554  144.59
## BATTING_2B  2276   69  241.25  238.0   458   46.80
## BATTING_3B  2276    0   55.25   47.0   223   27.94
## BATTING_HR  2276    0   99.61  102.0   264   60.55
## BATTING_BB  2276    0  501.56  512.0   878  122.67
## BATTING_SO  2174    0  735.61  750.0  1399  248.53
## BASERUN_SB  2145    0  124.76  101.0   697   87.79
## BASERUN_CS  1504    0   52.80   49.0   201   22.96
## PITCHING_H  2276 1137 1779.21 1518.0 30132 1406.84
## PITCHING_HR 2276    0  105.70  107.0   343   61.30
## PITCHING_BB 2276    0  553.01  536.5  3645  166.36
## PITCHING_SO 2174    0  817.73  813.5 19278  553.09
## FIELDING_E  2276   65  246.48  159.0  1898  227.77
## FIELDING_DP 1990   52  146.39  149.0   228   26.23
```

```
##                n  min    mean  median   max      sd
## TARGET_WINS 2276    0   80.79   82.00   146   15.75
## BATTING_H   2276  891 1469.27 1454.00  2554  144.59
## BATTING_2B  2276   69  241.25  238.00   458   46.80
## BATTING_3B  2276    0   55.25   47.00   223   27.94
## BATTING_HR  2276    0   99.61  102.00   264   60.55
## BATTING_BB  2276    0  501.56  512.00   878  122.67
## BATTING_SO  2276    0  728.23  739.00  1399  246.65
## BASERUN_SB  2276    0  124.63  105.00   697   85.28
## BASERUN_CS  2276    0   70.73   56.43   201   38.01
## PITCHING_H  2276 1137 1779.21 1518.00 30132 1406.84
## PITCHING_HR 2276    0  105.70  107.00   343   61.30
## PITCHING_BB 2276    0  553.01  536.50  3645  166.36
## PITCHING_SO 2276    0  807.63  803.50 19278  543.08
## FIELDING_E  2276   65  246.48  159.00  1898  227.77
## FIELDING_DP 2276   52  145.31  146.00   228   24.90
```

```
##                n min        mean    median max         sd
## TARGET_WINS    0   0   0.0000000  0.000000   0   0.000000
## BATTING_H      0   0   0.0000000  0.000000   0   0.000000
## BATTING_2B     0   0   0.0000000  0.000000   0   0.000000
## BATTING_3B     0   0   0.0000000  0.000000   0   0.000000
## BATTING_HR     0   0   0.0000000  0.000000   0   0.000000
## BATTING_BB     0   0   0.0000000  0.000000   0   0.000000
## BATTING_SO  -102   0   7.3754714 11.000000   0   1.876015
## BASERUN_SB  -131   0   0.1347715 -4.000000   0   2.507556
## BASERUN_CS  -772   0 -17.9280286 -7.427742   0 -15.053684
## PITCHING_H     0   0   0.0000000  0.000000   0   0.000000
## PITCHING_HR    0   0   0.0000000  0.000000   0   0.000000
## PITCHING_BB    0   0   0.0000000  0.000000   0   0.000000
## PITCHING_SO -102   0  10.1004968 10.000000   0  10.004072
## FIELDING_E     0   0   0.0000000  0.000000   0   0.000000
## FIELDING_DP -286   0   1.0780691  3.000000   0   1.327290
```

2.2 Remove Outliers

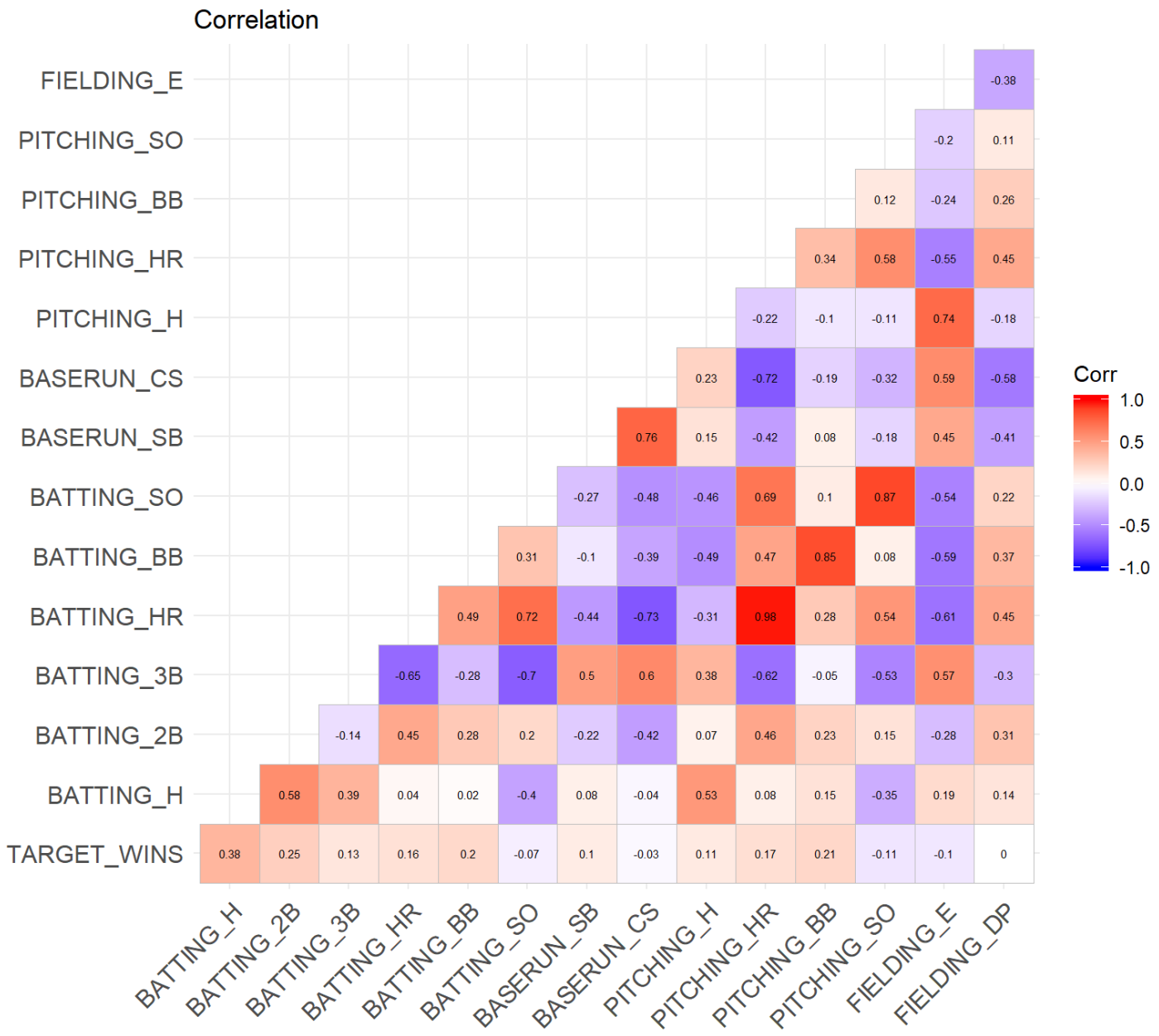
Outlier treament was done by placing a threshold of five times the standard deviation up from the mean and removing all observations that fell north of this boundary.

2.3 Correlation

The theoretical effect of strikeouts by batters, batters caught stealing, errors, walks, hits, and homeruns allowed were believed to have a negative impact on the number of wins of an individual team in a given year. A closer look at the correlation plot between the variables painted a different picture.

When compared to what was hypothesized, there was actually a positive impact for the number of wins for a team in a given year by walks, hits, and homeruns allowed; at the same time, variables previously thought to have a positive correlation - strikeouts by pitchers and double plays - had a negative correlation for the number of wins. The three variables with the greatest correlation to the number of wins were the hits allowed, the walks by batters,

and the walks allowed. Of these, the hits allowed had a relatively low correlation with the walks by batters and the walks allowed, whereas the walks allowed and the walks by batters had a direct positive correlation with one another.



2.4 Feature Engineering

Since there are four pairs of related variables that are two sides of the same coin, hits allowed vs. hits by batters, home runs allowed vs. home runs hit by batters, etc. and three of those pairs are highly correlated with each other we decided to try using the difference between them in place of the original variables in our original models. We decided to use offense (batting) minus defense (pitching). These arithmetically transformed offense / defense

variables are linearly related with BATTING and PITCHING variables, so we can include one or the other in a model, but not both. However, replacing original variables with these transforms did not improve R^2 in a base case.

---

3 BUILD MODELS

3.1 MODEL 1

Multiple regression can be created as a purely statistical model, through the use of significance tests, or it can be interpreted in a more practical, non-statistical manner. This approach is based on the subject-area expertise.

We've created the following categories from the most important to the least important variables according to the subject-area expert.

Very Important: BATTING_H, BATTING_HR, BATTING_SO ,FIELDING_E, PITCHING_SO

Fairly Important: BASERUN_SB, PITCHING_HR, BATTING_BB

Important: BATTING_2B, BATTING_3B, FIELDING_DP, PITCHING_H

Slightly Important: PITCHING_BB, BASERUN_CS

Not at all important: BATTING_HBP

'Batters hit by pitch' and 'Caught Stealing' have been eliminated as least important variables according to the expert.

```
##                Estimate  Std. Error     t value     Pr(>|t|)
## (Intercept) 36.59227159 5.663909946   6.4606026 1.278033e-10
## BATTING_H    0.01671944 0.005010218   3.3370691 8.606568e-04
## BATTING_HR   0.11918837 0.050659451   2.3527372 1.872280e-02
## BATTING_SO  -0.03113200 0.006598043  -4.7183691 2.525602e-06
## FIELDING_E  -0.03825885 0.003433165 -11.1439017 4.217234e-28
## PITCHING_SO  0.01996576 0.005419600   3.6839907 2.351124e-04
## BASERUN_SB   0.04144574 0.004934311   8.3994987 7.892888e-17
## PITCHING_HR -0.03498473 0.046875875  -0.7463269 4.555491e-01
## BATTING_BB   0.07833501 0.016216903   4.8304544 1.455372e-06
## BATTING_2B  -0.01106340 0.009389944  -1.1782184 2.388360e-01
## BATTING_3B   0.12637488 0.018154682   6.9610075 4.433682e-12
## FIELDING_DP -0.09122601 0.013110874  -6.9580422 4.525948e-12
## PITCHING_BB -0.05570586 0.014380420  -3.8737299 1.102908e-04
## PITCHING_H   0.01351302 0.001539308   8.7786357 3.239678e-18
```

We got 0.2793 on Adjusted R^2 after we removed these two variables. Once we tried to remove other not very important variables according to subject-area expert, we got an even lower R^2.

The next step we performed was backward elimunation, which was more effective compared to forward selection. BATTING_H and BATTING_2B have been removed based on the Backward Selection results.

This resulted in the following model:

```
TARGET_WINS ~ BATTING_H + BATTING_HR + BATTING_SO +
    FIELDING_E + PITCHING_SO + BASERUN_SB + BATTING_BB + BATTING_3B +
    FIELDING_DP + PITCHING_BB + PITCHING_H
```

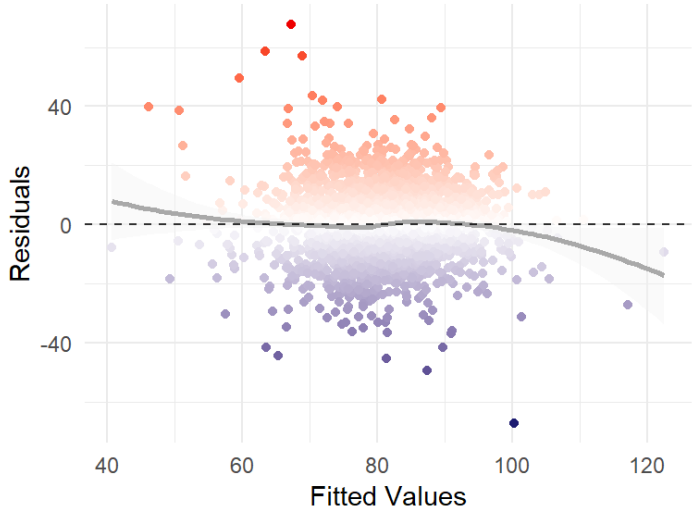
```
##                Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 50.96736132 3.815684658  13.357331 3.364909e-39
## BATTING_HR   0.13760271 0.050477986   2.725995 6.461210e-03
## BATTING_SO  -0.03318749 0.006582269  -5.041952 4.981386e-07
## FIELDING_E  -0.04164883 0.003244928 -12.835056 1.995570e-36
## PITCHING_SO  0.01869232 0.005383932   3.471871 5.267378e-04
## BASERUN_SB   0.04579083 0.004785457   9.568747 2.769643e-21
## PITCHING_HR -0.03887339 0.046964074  -0.827726 4.079147e-01
## BATTING_BB   0.08589278 0.016054744   5.349994 9.700147e-08
## BATTING_3B   0.15643670 0.016039022   9.753506 4.905956e-22
## FIELDING_DP -0.08357315 0.012949636  -6.453707 1.336225e-10
## PITCHING_BB -0.06196034 0.014270400  -4.341878 1.476224e-05
## PITCHING_H   0.01693109 0.001185182  14.285646 2.346317e-44
```

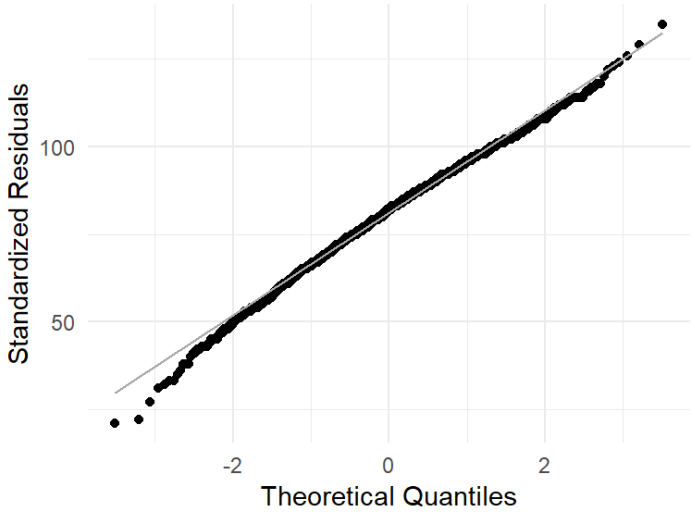
Our R^2 was still low (0.276), so we decided to look at the outliers, which can affect our model. Pitching_h had the high number of outliers which indicated a need for data transformation. We decided to use log transformation for this variable.

```
##                     Estimate   Std. Error     t value     Pr(>|t|)
## (Intercept)     -3.037377e+02 24.255675075 -12.5223350 8.231471e-35
## BATTING_HR       1.134496e-01  0.050029004   2.2676756 2.344480e-02
## BATTING_SO      -1.543857e-02  0.006667500  -2.3154965 2.067660e-02
## FIELDING_E      -4.201715e-02  0.003159393 -13.2991224 6.928751e-39
## PITCHING_SO      7.688177e-03  0.005335672   1.4409015 1.497537e-01
## BASERUN_SB       4.138238e-02  0.004680349   8.8417288 1.879618e-18
## PITCHING_HR     -4.352267e-02  0.046400753  -0.9379734 3.483602e-01
## BATTING_BB       1.011232e-01  0.016002125   6.3193606 3.164812e-10
## BATTING_3B       1.171124e-01  0.015935480   7.3491582 2.789677e-13
## FIELDING_DP     -9.429889e-02  0.012828663  -7.3506406 2.759637e-13
## PITCHING_BB     -7.559524e-02  0.014233269  -5.3111651 1.198021e-07
## log(PITCHING_H)  5.221724e+01  3.243446517  16.0993076 3.078483e-55
```

After we used the log trasformation our model's Adjusted R^2 increased to 0.2921.

*** Should have some explanation of this figure***





Model 1: Residual Plot and Q-Q Plot

*** Needs explanation***

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   60.82   76.09   81.12   81.44   86.10  108.26      54
```

*** Needs explanation***

```
## # A tibble: 1 x 7
##   r.squared adj.r.squared statistic   p.value    df deviance df.residual
##       <dbl>         <dbl>     <dbl>     <dbl> <int>    <dbl>       <int>
## 1     0.296         0.292      84.6 7.33e-160    12  347583.        2218
```

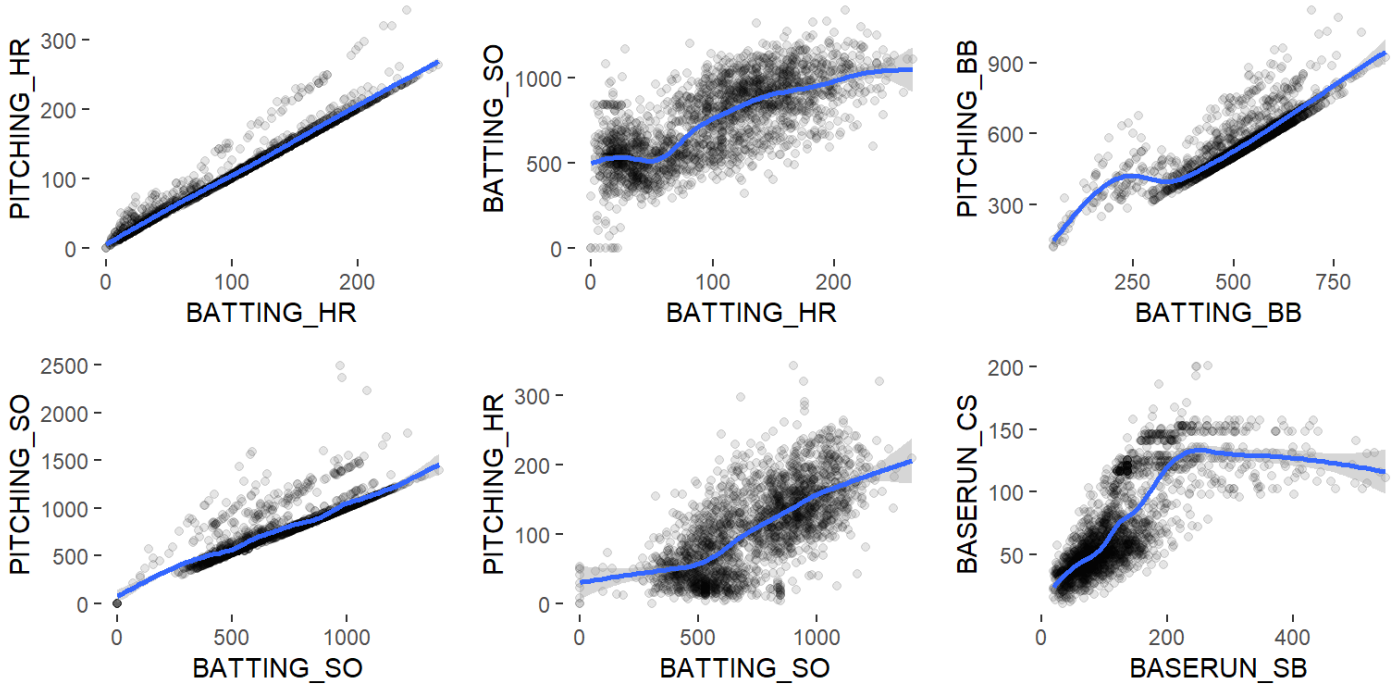
Summary of Results for Model 1:

The overall Subject-Area expertise wasn't as effective as a stand alone method of creating multiple regression models. Statistical iterations which were performed contradicted the subject area expert, such as, removing Batting_H from the model. Additionally log tranformation of PITCHING_H made a significant improvment in our model linearity.

3.2 MODEL 2

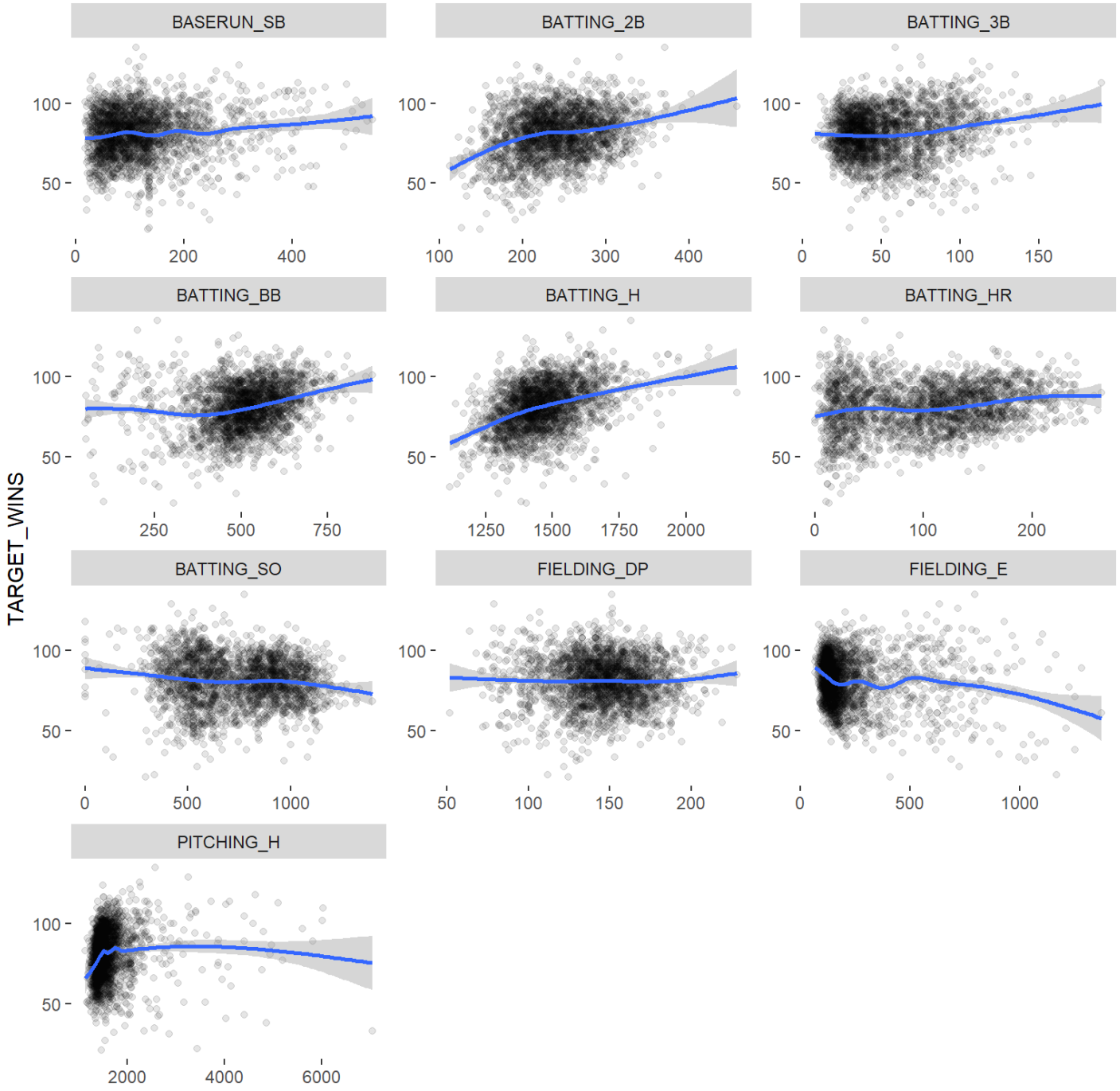
Our approach for Model 2 was to try to use as many of the tools as possible that are available in R and that we have learned thus far to determine a model based solely on the statistical qualities of the predictor variables without any regard to our expert's opinion.

We started by plotting the relationships between variables that had high correlation values to look for potential collinearity problems.

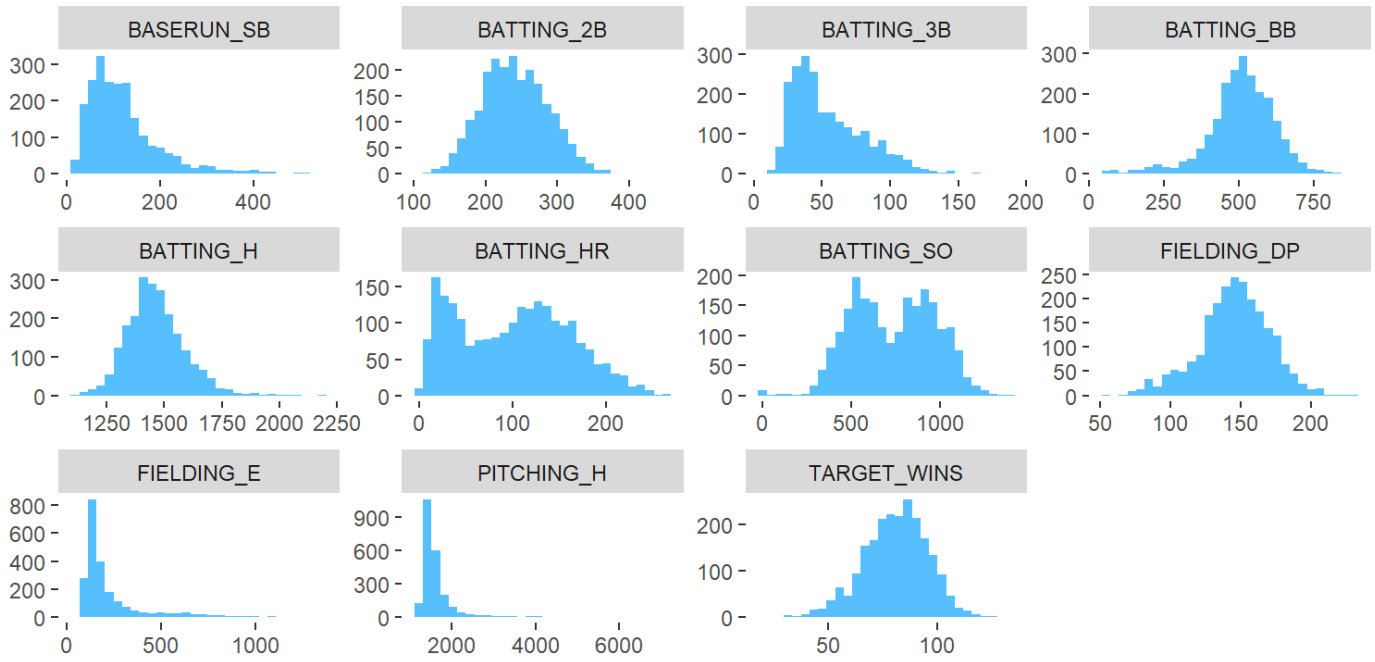


Scatterplots showing possible collinearity problems

Based on the charts above we decided somewhat arbitrarily to remove the three pitching variables (`PITCHING_HR` , `PITCHING_BB` , and `PITCHING_SO`) rather than the corresponding batting variables (`BATTING_HR` , `BATTING_BB` , and `BATTING_SO`) due to the extremely high correlation between these predictors. We then plotted the reamining variabes to see if they showed a linear relationship with the target variable. Most of the premaining predictors showed a clear linear relationship with the target, however, the extreme skew of `PITCHING_H` and `FIELDING_E` as well as a more moderate skew in `BASERUN_SB` and `BATTING_3B` , can be seen in the plots.



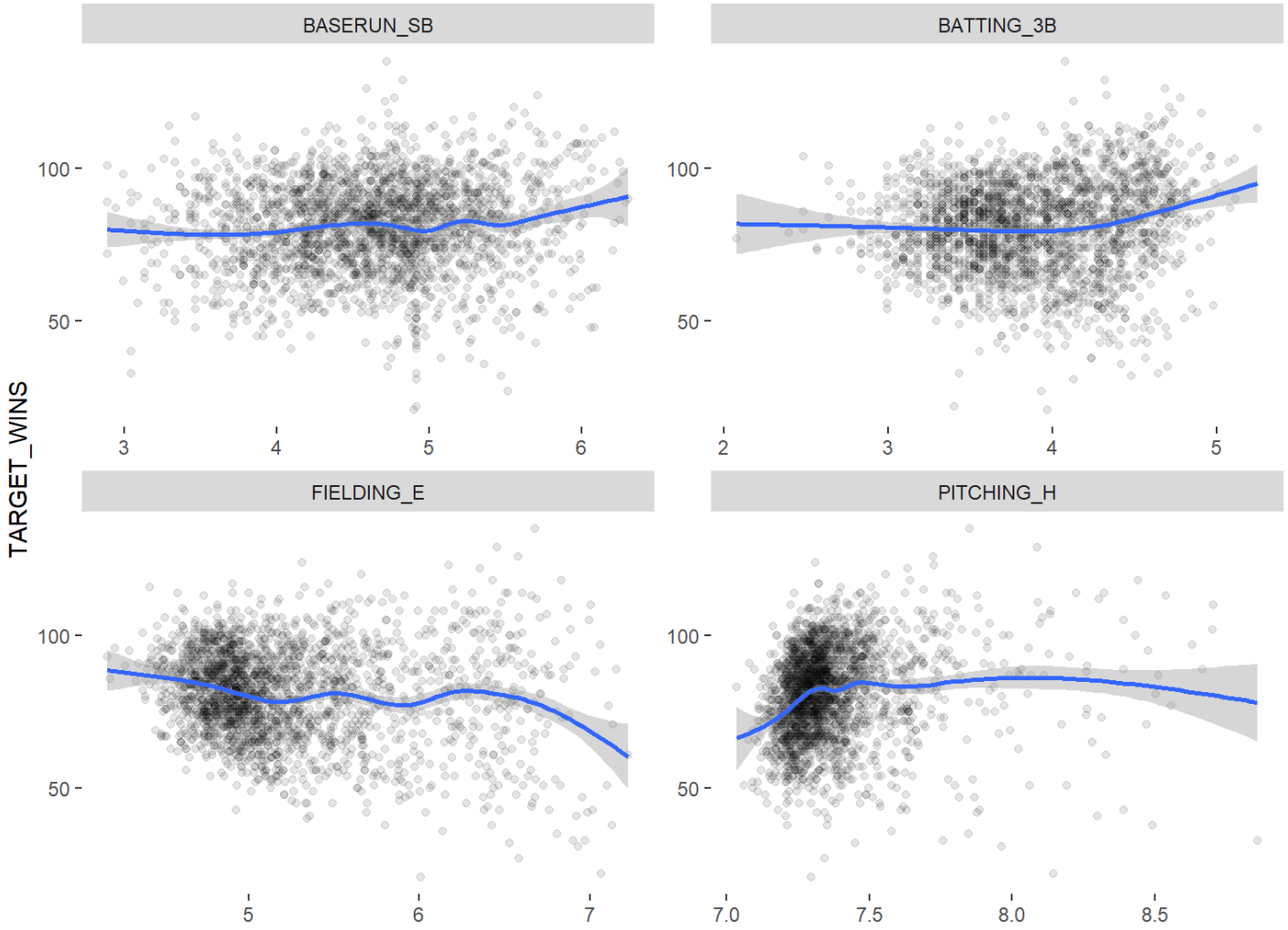
Linear relationship between each predictor and the arget



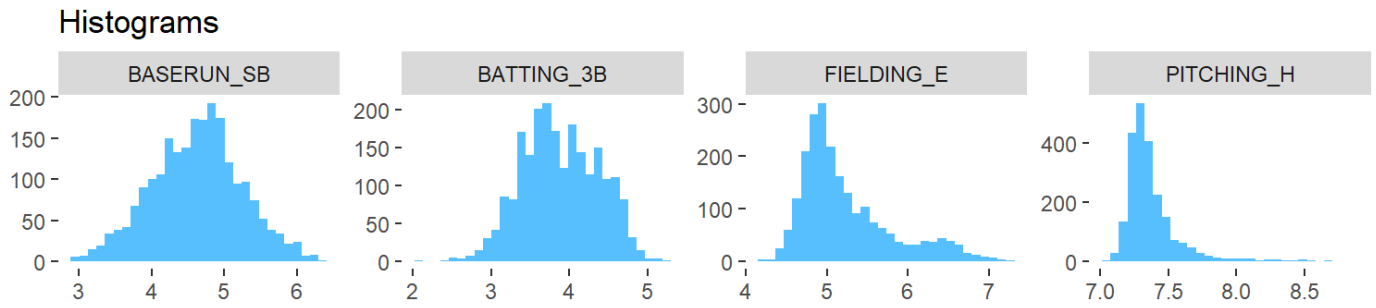
Predictor variable distributions

3.2.1 Log Transform Data

We decided to log transform `PITCHING_H`, `FIELDING_E`, `BASERUN_SB` and `BATTING_3B` in order to compensate for the skew. The resulting distributions can be seen in the revised plots below.



Linear relationship between each log transformed predictor and the Target showing decreased skew



Log transformed distributions showing decreased skew

3.2.2 Building the Model

Finally we built a model based on the selected variables including the log transformations where appropriate.

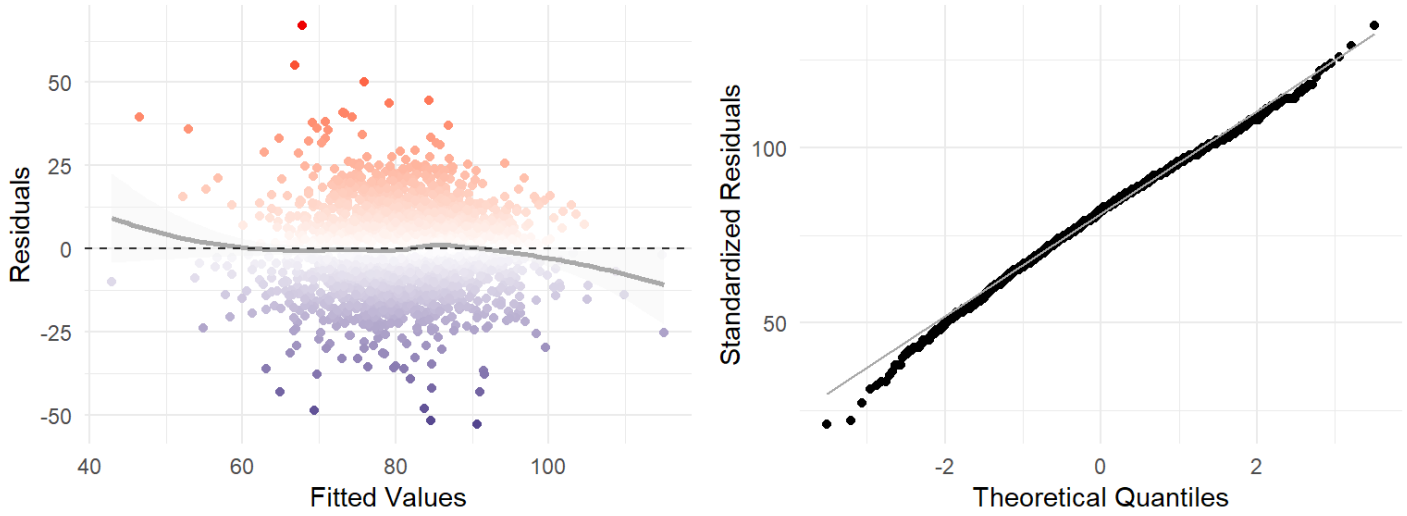
```
TARGET_WINS ~ BATTING_H + BATTING_2B + log(BATTING_3B) + BATTING_HR +
    BATTING_BB + BATTING_SO + log(BASERUN_SB) + log(PITCHING_H) +
    log(FIELDING_E) + FIELDING_DP
```

All of the variables had a very low p-value indicating a significant impact on our target, however our R^2 value was low at only 0.2889.

3.2.2.1 R^2 0.2888829

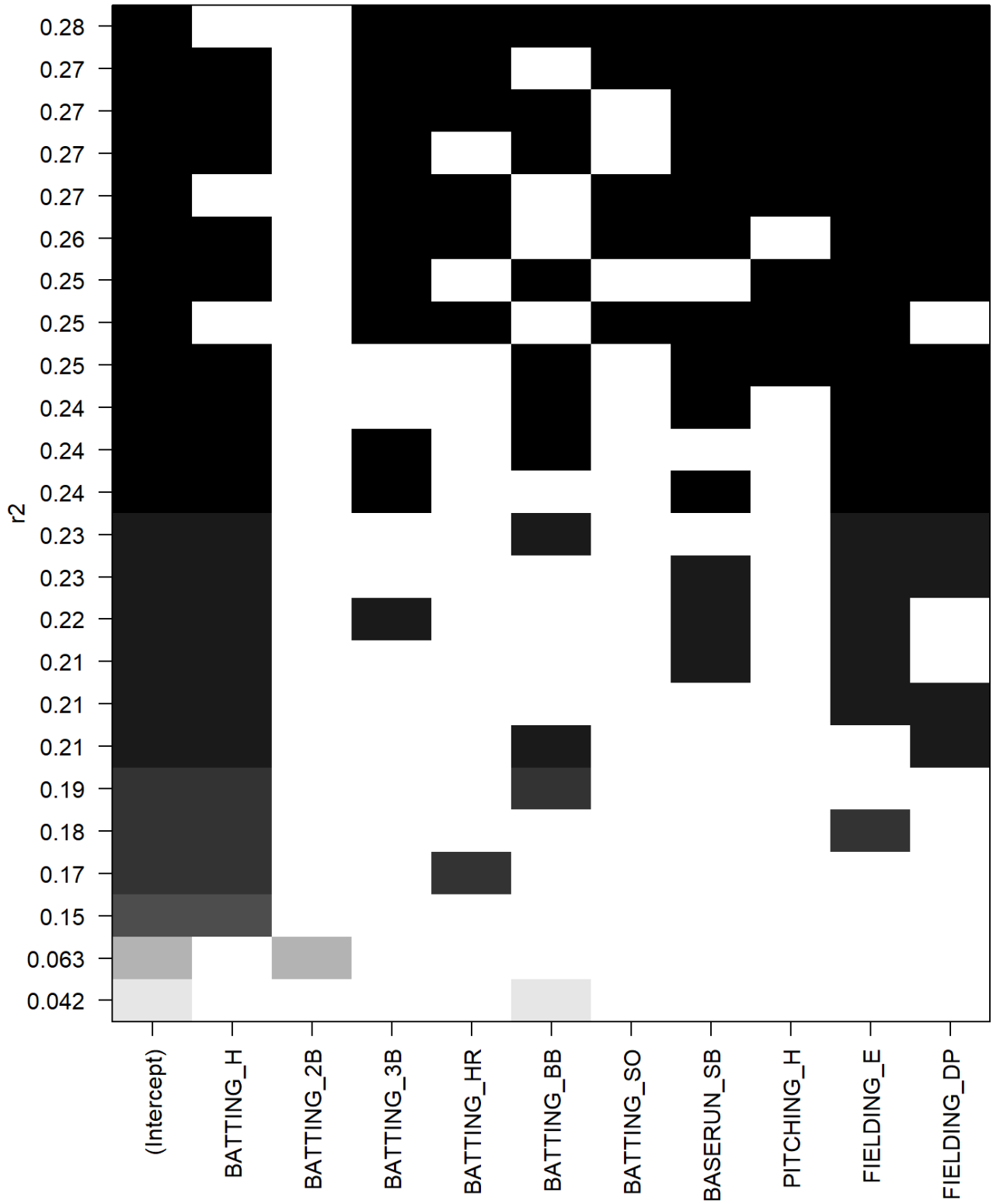
Full Model Coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-62.9724524	15.9139042	-3.9570710	0.0000783
BATTING_H	0.0255615	0.0046111	5.5434260	0.0000000
BATTING_2B	-0.0297598	0.0092155	-3.2293090	0.0012590
log(BATTING_3B)	6.9697812	0.9476551	7.3547650	0.0000000
BATTING_HR	0.0703282	0.0101256	6.9455790	0.0000000
BATTING_BB	0.0192290	0.0031872	6.0331920	0.0000000
BATTING_SO	-0.0112281	0.0024113	-4.6564500	0.0000034
log(BASERUN_SB)	4.5460080	0.5617657	8.0923560	0.0000000
log(PITCHING_H)	19.1408576	2.6161541	7.3164110	0.0000000
log(FIELDING_E)	-13.1027616	1.0618606	-12.3394370	0.0000000
FIELDING_DP	-0.1067225	0.0131384	-8.1229240	0.0000000



Model 2: Residual Plot and Q-Q Plot

Our residuals look normally distributed and random, and with constant variability, no indication of homoscedasticity. However we thought we may be able to use some other tools in R to refine our model and get a better R^2 value. So next we tried using the leaps package to see if it would recommend removing any of our chosen variables from the model. In the following plot you can see that we could remove `BATTING_H`, `BATTING_2B` without affecting out R^2 much, but it would not improve the model.


r2
0.28
0.27
0.27
0.27
0.27
0.26
0.25
0.25
0.25
0.24
0.24
0.24
0.23
0.23
0.22
0.21
0.21
0.21
0.19
0.18
0.17
0.15
0.063
0.042
(Intercept)
BATTING_H
BATTING_2B
BATTING_3B
BATTING_HR
BATTING_BB
BATTING_SO
BASERUN_SB
PITCHING_H
FIELDING_E
FIELDING_DP

```
## NULL
```

Next we tried standardizing the (non-log-transformed) variables to see what impact that might have on our model. As you can see standardzing actually resulted in a significant reduction in our R^2 value from 0.2889 to 0.274.

3.2.2.2 R^2 0.2739688

Full SCALED Model Coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	80.9717489	0.2690639	300.938712	0.0000000
BATTING_H	2.6935751	0.5977789	4.505972	0.0000070
BATTING_2B	-0.4725422	0.4272176	-1.106092	0.2688064
BATTING_3B	3.0938480	0.4776884	6.476708	0.0000000
BATTING_HR	4.8934200	0.5902167	8.290887	0.0000000
BATTING_BB	1.7865264	0.3813959	4.684178	0.0000030
BATTING_SO	-1.9686032	0.5564927	-3.537519	0.0004122
BASERUN_SB	2.6367694	0.3743523	7.043551	0.0000000
PITCHING_H	4.5352452	0.5648773	8.028726	0.0000000
FIELDING_E	-6.4017884	0.5957772	-10.745273	0.0000000
FIELDING_DP	-2.3837076	0.3295935	-7.232265	0.0000000

3.2.3 Test all of the predictors

Next we ran an ANOVA test to compare our model to the null model. With a p-value that is basically zero, clearly our model is statistically significant.

```
## Analysis of Variance Table
##
## Model 1: TARGET_WINS ~ 1
## Model 2: TARGET_WINS ~ BATTING_H + BATTING_2B + log(BATTING_3B) + BATTING_HR +
##     BATTING_BB + BATTING_SO + log(BASERUN_SB) + log(PITCHING_H) +
##     log(FIELDING_E) + FIELDING_DP
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2229 493421
## 2   2219 350880 10    142541 90.144 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2.4 Testing a subspace

We then tried testing a subspace. Since our initial models using the difference between the corresponding batting and pitching variables did not show promise we tried adding those two variables instead.

3.2.4.1 R^2 0.286633

Subspace Model Coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-47.5598798	14.1082067	-3.371079	0.0007616
I(BATTING_HR + PITCHING_HR)	0.0361009	0.0049624	7.274907	0.0000000
I(BATTING_BB + PITCHING_BB)	0.0077316	0.0014542	5.316742	0.0000001
I(BATTING_SO + PITCHING_SO)	-0.0052127	0.0010862	-4.798948	0.0000017

	Estimate	Std. Error	t value	Pr(>|t|)
BATTING_H	0.0273198	0.0048717	5.607838	0.0000000
BATTING_2B	-0.0248999	0.0093466	-2.664069	0.0077761
log(BATTING_3B)	6.4554813	0.9412999	6.858050	0.0000000
log(BASERUN_SB)	3.0279771	0.7904181	3.830855	0.0001312
BASERUN_CS	0.0460357	0.0177169	2.598400	0.0094279
log(PITCHING_H)	17.7088157	2.6101829	6.784511	0.0000000
log(FIELDING_E)	-13.7380684	1.0992719	-12.497426	0.0000000
FIELDING_DP	-0.0968425	0.0134958	-7.175776	0.0000000

Once again our model declined in performance rather than improving.

Last, but not least, we used the stepAIC function from the MASS package to see if it came up with different recommendations for what variabels to keep and which to exclude from our model. We started with all variables putting back the ones we had previously taken out due to collinearity issues and let the algorithm choose which to keep.

The final suggested model was:

```
Final Model:
TARGET_WINS ~ BATTING_H + BATTING_3B + BATTING_HR + BATTING_BB +
    BATTING_SO + BASERUN_SB + BASERUN_CS + PITCHING_H + PITCHING_BB +
    PITCHING_SO + FIELDING_E + FIELDING_DP
```

In comparison to our original model we had the following variables added to our model (`BASERUN_CS` , `PITCHING_BB` , and `PITCHING_SO`) and the following variable removed (`BATTING_2B`).

We tried multiple iterations of that model, without any log transformations, with log transformations, with and without the collinear variables, but whever we removed one of the collinear variables our model would decline in performance, so we decided to try our multiplying the corresponding collinear variables to gether and BINGO! We got and R^2 of 0.3247 using the following model:

```
TARGET_WINS ~ BATTING_3B + BATTING_HR + BATTING_BB*PITCHING_BB +
    BATTING_SO*PITCHING_SO + BASERUN_SB + BASERUN_CS + BATTING_H*log(PITCHING_H) +
    log(FIELDING_E) + FIELDING_DP
```

3.2.4.2 R^2 0.3247108

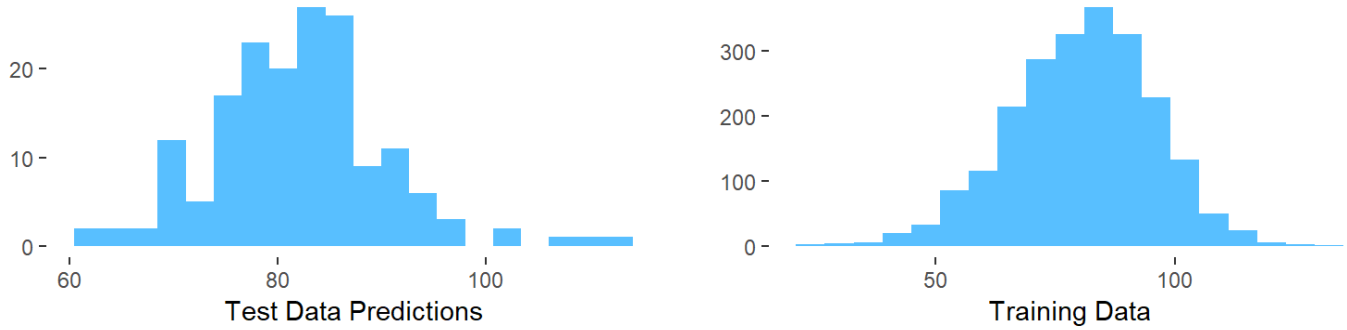
Full Model Coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	204.630208910	3.8266135	1.970884	00.0488615
BATTING_3B	0.1602499	0.0178747	8.965183	60.0000000
BATTING_HR	0.0721476	0.0102070	7.068457	30.0000000
BATTING_BB	0.0624848	0.0199429	3.133183	50.0017518
PITCHING_BB	-0.0863035	0.0130572	-6.609635	80.0000000
BATTING_SO	0.0016687	0.0091718	0.181943	30.8556439
PITCHING_SO	0.0285700	0.0064784	4.410052	50.0000108
BASERUN_SB	0.0304137	0.0059322	5.126867	10.0000003
BASERUN_CS	0.0630545	0.0157254	4.009722	90.0000628
BATTING_H	-0.2185464	0.0544015	-4.017284	50.0000608
log(PITCHING_H)	-8.4383898	14.3385415	-0.588511	00.5562494

	Estimate	Std. Error	t value	Pr(>|t|)
log(FIELDING_E)	-16.4657197	1.1044063	-14.9091144	0.0000000
FIELDING_DP	-0.0883833	0.0131868	-6.7024185	0.0000000
BATTING_BB:PITCHING_BB	0.0000469	0.0000169	2.7798125	0.0054850
BATTING_SO:PITCHING_SO	-0.0000271	0.0000044	-6.1888757	0.0000000
BATTING_H:log(PITCHING_H)	0.0303947	0.0074285	4.0916590	0.0000444

3.2.5 Predictions

We ran predictions on our final model and plotted the distribution next to the distribution from our target in the training data set to compare…

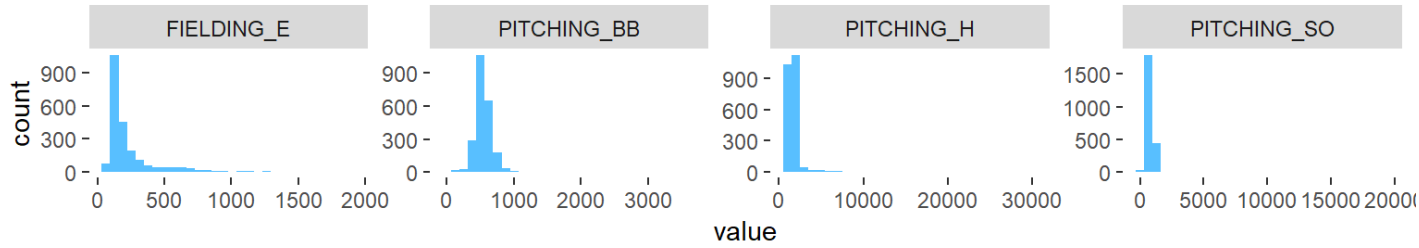


Predictions vs. training data

3.3 MODEL 3

We sought to explore whether there was a relationship between wins and the difference of specific offensive and defensive team capabilities - hits, homeruns, balls, and strike-outs. Incorporating variables that reflect those differences (i.e. subtracting batting hits from pitching hits, and so on), however, did not improve the explantory power of the model beyond using the original variables.

Given these variables did not yield improvements, in their place we explored a third model. As the histograms below highlight, a number of the independent variables - pitching hits, pitching homeruns, pitching strikeouts - demonstrate pronounced rightward-skew.



Histograms of variables showing pronounced rightward-skew

We corrected for that skew by transforming those three variables using natural logarithms. When we tested those log transformations in a model where they replaced the untransformed, original variables combined with all other variables, we found that neither the originals nor the log transformations for pitching homeruns and pitching strikeouts met the threshold of significance (a p-value below the α level of .05). Based on high p-values, over a series of backward steps we removed pitching homeruns, pitching strikeouts, and baserun caught stealing, yielding the following model:

[Jeremy: team, should we write LaTeX formulas for each model or just cable model coefficients?]

$y = $

Log Transform Model Coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-405.0139425	36.9977658	-10.9469840	0.0000000
BATTING_H	-0.0134654	0.0060337	-2.2316790	0.0257358
BATTING_2B	-0.0150055	0.0091463	-1.6406040	0.1010214
BATTING_3B	0.1405383	0.0176556	7.9599790	0.0000000
BATTING_HR	0.0723000	0.0097039	7.4506140	0.0000000
BATTING_BB	0.1246742	0.0126620	9.8463620	0.0000000
BATTING_SO	-0.0065541	0.0023297	-2.8132910	0.0049469
BASERUN_SB	0.0438045	0.0047630	9.1967740	0.0000000
log(PITCHING_H)	68.6642334	5.7601412	11.9205820	0.0000000
PITCHING_BB	-0.0951797	0.0106315	-8.9526030	0.0000000
FIELDING_E	-0.0473986	0.0035504	-13.3500850	0.0000000
FIELDING_DP	-0.0885260	0.0129387	-6.8419480	0.0000000

Based on this model's F-statistic and p-value, we can reject the null hypothesis that coefficients with values of zero would fit the data better. Per the adjusted r^2 value, this model explains approximately 29.56% of the variance in wins. However, in doing so it treats the batting hits and batting second base runs as drags on wins (with negative coefficients), and pitching hits as buoying wins - which is counterunituitive. While the other coefficients make more intuitive sense, these signs call into question how effectively we can use this model to understand the relationships between the independent variables and wins.

---

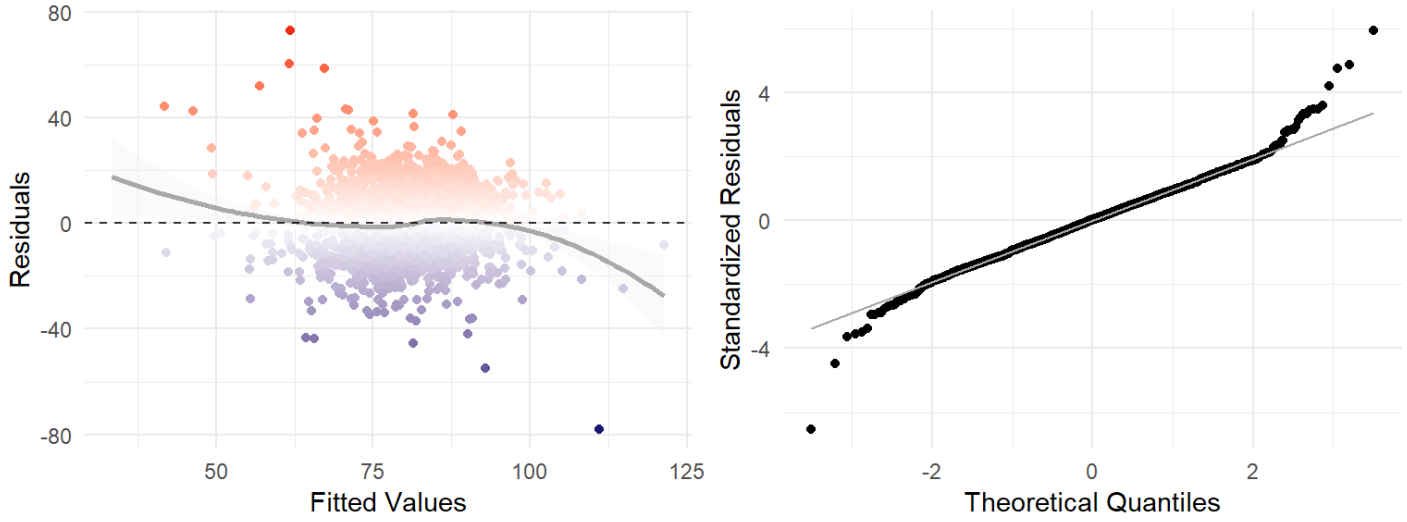
4 SELECT MODELS

4.1 Instructions:

Decide on the criteria for selecting the best multiple linear regression model. Will you select a model with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your model. For the multiple linear regression model, will you use a metric such as Adjusted R2, RMSE, etc.? Be sure to explain how you can make inferences from the model, discuss multi-collinearity issues (if any), and discuss other relevant model output. Using the training data set, evaluate the multiple linear regression model based on (a) mean squared error, (b) R2, (c) F-statistic, and (d) residual plots. Make predictions using the evaluation data set.

4.2 Comparison of models

[Jeremy: added in a chart for model 3]



Model 3: Residual Plot and Q-Q Plot

5 Appendix