Homework #1: Baseball Analysis

Data 621 Business Analytics and Data Mining

Aadi Kalloo, Nathan Lim, Asher Meyers, Daniel Smilowitz, Logan Thomson

Due June 19, 2016

Contents

Data Exploration

The data analyzed in this report includes 2276 professional baseball teams for the years 1871-2006. In total, 16 variables were present in the data provided. Included below is a summary of descriptive statistics, correlations to wins, and the number of missing values for each variable in the provided data set:

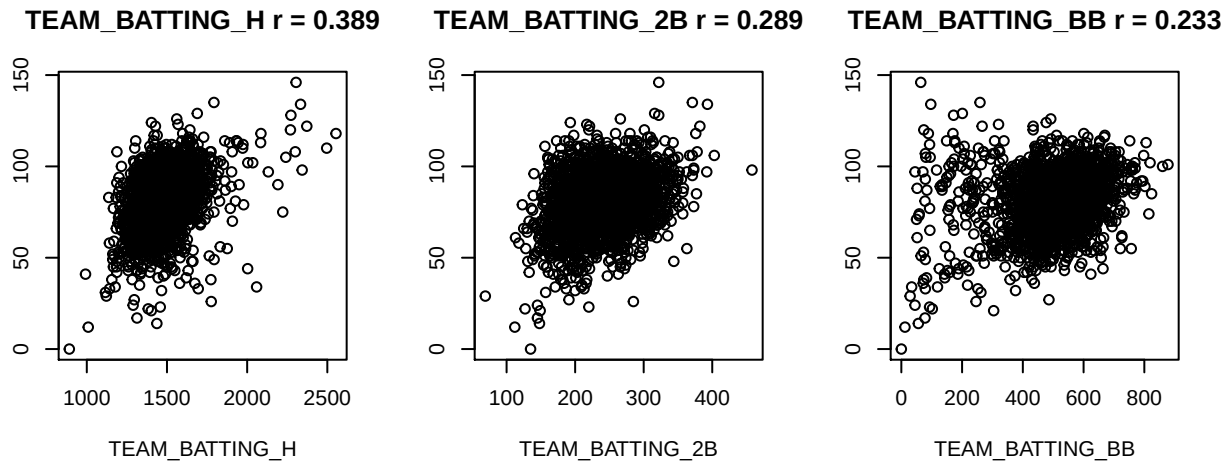
Table 1

	VAR_NAME	MEAN	MEDIAN	CORRELATION TO WINS (r)	NUM_MISSING
2	TARGET_WINS	80.79086	82.0	NA	NA
1	TEAM_BASERUN_CS	52.80386	49.0	0.0224041	772
21	TEAM_BASERUN_SB	124.76177	101.0	0.1351389	131
3	TEAM_BATTING_2B	241.24692	238.0	0.2891036	0
4	TEAM_BATTING_3B	55.25000	47.0	0.1426084	0
5	TEAM_BATTING_BB	501.55888	512.0	0.2325599	0
6	TEAM_BATTING_H	1469.26977	1454.0	0.3887675	0
7	TEAM_BATTING_HBP	59.35602	58.0	0.0735042	2085
8	TEAM_BATTING_HR	99.61204	102.0	0.1761532	0
9	TEAM_BATTING_SO	735.60534	750.0	-0.0317507	102
10	TEAM_FIELDING_DP	146.38794	149.0	-0.0348506	286
11	TEAM_FIELDING_E	246.48067	159.0	-0.1764848	0
12	TEAM_PITCHING_BB	553.00791	536.5	0.1241745	0
13	TEAM_PITCHING_H	1779.21046	1518.0	-0.1099371	0
14	TEAM_PITCHING_HR	105.69859	107.0	0.1890137	0
15	TEAM_PITCHING_SO	817.73045	813.5	-0.0784361	102

It can be seen that there are missing values in 6 of the variables in the data set, and these missing values range from approximately 5-92% of the data provided for their respective variables. However, in only two exceptions do the missing data

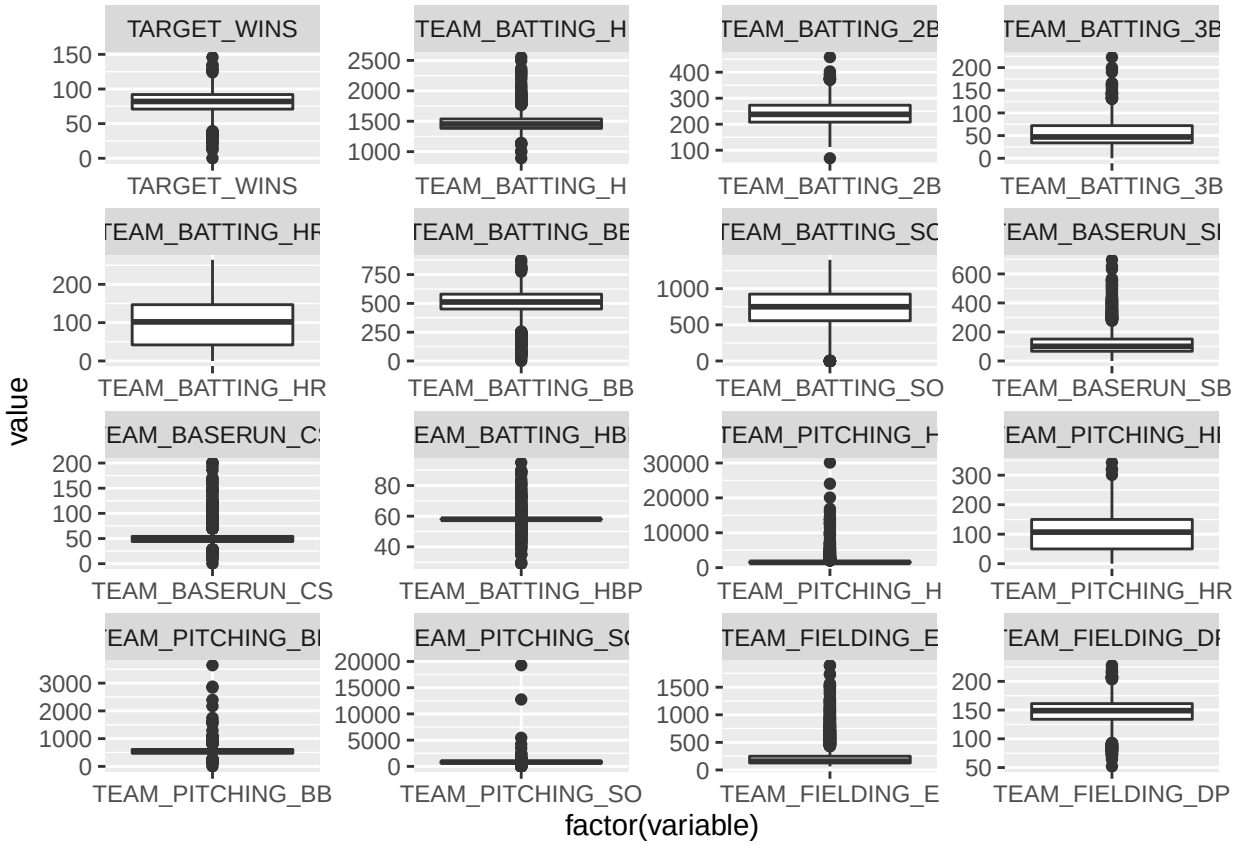
account for more than 11% of the missing data.

Below are graphs that show the relationship to *Target Wins* for the three variables with the highest correlation coefficient:



As can be seen from Table 1, there are few variables that have any particularly strong correlation with `TARGET_WINS`.The full array of scatterplots representing correlations between `TARGET_WINS` and other variables may be found in Appendix A.

The distribution of values and outliers is also of significant importance in understanding the baseball data set. Here it can be seen that many variables have a skewed distribution:



In summary, the baseball data set provided includes many variables with a skewed distribution, few variables that correlate well with `TARGET_WINS`, and several variables that have missing data and should either require data imputation or should be excluded. The following sections serve to review these issues and go on to create a working regression model that can predict `TARGET_WINS`.

Data Preparation

The dataset contains 17 columns - an index column (INDEX), a response column (TARGET_WINS) and 15 predictor columns. There are 2,276 observations - but there are many missing values for many of the predictors.

Two predictors in particular stand out:

	Predictor Name	Description	Impact	% Missing	r with Response	p-Value
a	TEAM_BATTING_HBP	Batters hit by pitch (free base)	Positive	91.6%	0.07	0.31
b	TEAM_BASERUN_CS	Strikeouts by batters	Negative	33.9%	0.02	0.39

Including these predictors in our dataset would mean that we would either have to forego a significant portion of our data (34% or 92%), or impute a large number of data points. Their correlation coefficients with the response are less than an absolute value of 0.07; the p values of a simple one variable linear regression using them and the response yields models of no statistical significance (i.e. $p > 0.05$). Thus, it seems safe to exclude these predictors from our models. In this way, we avoid the twin pitfalls of mass exclusion and imputation.

It was determined that the *Hits By Pitch* variable had too many missing values to be useful for regression, and thus this variable was excluded from the model building process. As shown in Table 1 above, there are several variables that have missing values. The attempted solution to this problem involved imputation using the median for each variable in the data set. A summary of the data is shown here again for inspection and confirmation of similarity between the old and new data sets:

Missing Values Imputed With Median

	VAR_NAME	MEAN	MEDIAN	CORRELATION TO WINS (r)	NUM_MISSING
2	TEAM_BATTING_H	1469.26977	1454.0	NA	NA
1	TEAM_BASERUN_CS	51.51362	49.0	0.0159598	0
21	TEAM_BASERUN_SB	123.39411	101.0	0.1236109	0
3	TEAM_BATTING_2B	241.24692	238.0	0.2891036	0
4	TEAM_BATTING_3B	55.25000	47.0	0.1426084	0
5	TEAM_BATTING_BB	501.55888	512.0	0.2325599	0
6	TEAM_BATTING_HBP	58.11380	58.0	0.0165164	0
7	TEAM_BATTING_HR	99.61204	102.0	0.1761532	0
8	TEAM_BATTING_SO	736.25044	750.0	-0.0305814	0
9	TEAM_FIELDING_DP	146.71617	149.0	-0.0300863	0
10	TEAM_FIELDING_E	246.48067	159.0	-0.1764848	0
11	TEAM_PITCHING_BB	553.00791	536.5	0.1241745	0
12	TEAM_PITCHING_H	1779.21046	1518.0	-0.1099371	0
13	TEAM_PITCHING_HR	105.69859	107.0	0.1890137	0
14	TEAM_PITCHING_SO	817.54086	813.5	-0.0757997	0

Further exclusions to the data were made:

Exclusion	Explanation
INDEX == 1347	This row had a suspicious set of zero entries
TEAM_BATTING_BB == 0	Anomalously low walk count (expected occurences of a zero value for this predictor are zero)
TEAM_BATTING_SO	Outside of recognized records link
TEAM_BATTING_HR	Outside of recognized records link

It should be noted that the records excluded from the first two rows of the table above are similar. This suggests that strikeouts were not recorded for those rows, but were marked as zero. Those two predictors have the same number of NA values, 102, suggesting their recording method could have been linked.

Many of the more advanced baseball statistics are simply combinations of other statistics (i.e. slugging percentage is total bases divided by at-bats). Using the predictors given in the data set, we wanted to see if combining predictors and/or calculating

new values would increase any significance in a model that is trying to predict wins. Total bases and extra bases are both stats that can easily be calculated with the given data. Assuming that the number of doubles, triples and home runs are included in the `TEAM_BATTING_H` variable, we can subtract these out to obtain the number of singles. Adding this to the doubles, triples, and homeruns, each multiplied by the number of bases each is worth (2,3, and 4 respectively) would give total bases. Doing the same thing, but excluding single base hits would give the number of extra bases as well. These two statistics were used in the model, along with many of the other predictors.

Since the fielding errors predictor was drastically right-skewed (see Appendix A), a log transformation was done on this predictor, resulting in a better fit of the model.

Model Creation

Model Summary Table

Model #	# of Predictors	Adj. R^2	F-Statistic	P-Value	Residual Standard Error	Degrees of Freedom
1	11	0.31	95	2.2e-16	13.07	2264
2	8	0.22	82	2.2e-16	13.82	2266
3	7	0.35	157	2.2e-16	11.07	1993
4	4	0.28	225	2.2e-16	11.78	2271
5	8	0.31	125	2.2e-16	13.15	2267

Model 1: Simple Full Linear Regression, With Removal of Non-Significant Predictors

Description: Missing values were replaced with the median values from the associated predictor, to retain all data points for making a regression; a linear regression was fit to all predictors; All non-significant predictors (p<.05) were removed sequentially. The final iteration of this regression model is shown here:

	Estimate	Std. Error	t value	Pr(>|t|)
TEAM_BATTING_H	0.04909	0.00367	13.38	2.469e-39
TEAM_BATTING_2B	-0.02137	0.009163	-2.333	0.01975
TEAM_BATTING_3B	0.06658	0.01662	4.005	6.4e-05
TEAM_BATTING_HR	0.0674	0.009632	6.998	3.399e-12
TEAM_BATTING_BB	0.01155	0.003375	3.421	0.0006342
TEAM_BATTING_SO	-0.008521	0.002453	-3.474	0.0005227
TEAM_BASERUN_SB	0.02492	0.004209	5.92	3.699e-09
TEAM_PITCHING_H	-0.000777	0.0003209	-2.421	0.01555
TEAM_PITCHING_SO	0.002966	0.0006719	4.415	1.059e-05
TEAM_FIELDING_E	-0.01901	0.002392	-7.948	2.972e-15
TEAM_FIELDING_DP	-0.1218	0.01293	-9.419	1.079e-20
(Intercept)	22.34	5.234	4.269	2.043e-05

Table 7: Fitting linear model: TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_E + TEAM_FIELDING_DP

Observations	Residual Std. Error	R^2	Adjusted R^2
2276	13.07	0.3151	0.3117

Model 2: SLR Bounded by Recent MLB Data (1962-)

Description: Missing values were replaced with the median values from the associated predictor. Data was compared against

records from 1962 and onwards, aka the MLB dataset, and data outside the bounds of that external dataset were replaced with the medians of the associated predictor. Eg, if one of the records in our dataset had more home runs hit than in all of the MLB dataset, then that home run data point was replaced with the median home run figure in our dataset.

Then, a linear regression was fitted to all predictors; predictors were removed in order of significance, to obtain a model with a higher f-statistic.

	Estimate	Std. Error	t value	Pr(>|t|)
TEAM_BATTING_H	0.03724	0.003922	9.496	5.32e-21
TEAM_BATTING_2B	0.01864	0.008774	2.124	0.03376
TEAM_BATTING_BB	0.03099	0.003743	8.28	2.084e-16
TEAM_BATTING_SO	-0.01306	0.00211	-6.191	7.083e-10
TEAM_BASERUN_SB	0.03816	0.0053	7.201	8.087e-13
TEAM_PITCHING_HR	0.05392	0.008428	6.397	1.92e-10
TEAM_FIELDING_E	-0.01014	0.001499	-6.764	1.706e-11
TEAM_FIELDING_DP	-0.1141	0.01365	-8.357	1.107e-16
(Intercept)	25.55	5.856	4.362	1.344e-05

Table 9: Fitting linear model: TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_PITCHING_HR + TEAM_FIELDING_E + TEAM_FIELDING_DP

Observations	Residual Std. Error	R^2	Adjusted R^2
2275	13.82	0.2248	0.2221

Model 3: Data Bounded by 1880-2015 Records

Three predictors were removed: Predictors removed, due to high NA count: TEAM_BATTING_HBP and TEAM_BASERUN_CS Predictor removed, due to lack of relevance: TEAM_FIELDING_DP

Bounds for predictors were set by minimum and maximum all-time MLB records, with citations shown in the code. Where our records were outside the bounds of these external records, they were replaced with NA values. Predictors were removed sequentially removed, in order of significance; multiple predictors were created and tested, in the hopes of improving fit statistics, and one proved useful - the number of 1st base hits. This predictor was then added to the model.

	Estimate	Std. Error	t value	Pr(>|t|)
TEAM_BATTING_3B	0.1804	0.01727	10.45	6.657e-25
TEAM_BATTING_HR	0.1149	0.007506	15.31	4.252e-50
TEAM_BATTING_BB	0.02858	0.003096	9.231	6.64e-20
TEAM_BATTING_SO	-0.017	0.002323	-7.32	3.592e-13
TEAM_BASERUN_SB	0.07938	0.004722	16.81	2.048e-59
TEAM_FIELDING_E	-0.0736	0.003888	-18.93	1.226e-73
TEAM_BATTING_1B	0.02327	0.004018	5.79	8.143e-09
(Intercept)	36.71	5.549	6.616	4.739e-11

Table 11: Fitting linear model: TARGET_WINS ~ TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_FIELDING_E + TEAM_BATTING_1B

Observations	Residual Std. Error	R^2	Adjusted R^2
2001	11.07	0.3557	0.3535

Model 4: Using only significant predictors from model using all variables

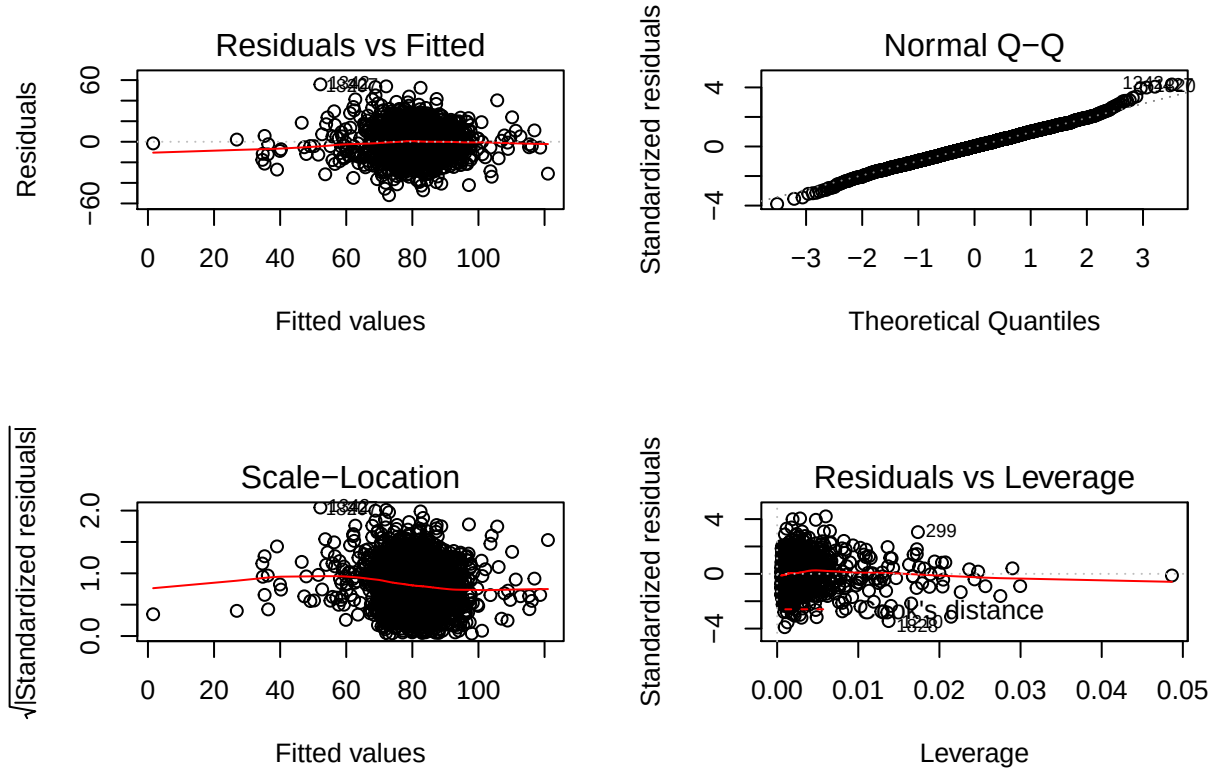
When the model using all variables to predict `TARGET_WINS` was considered, it was decided to create a new model using only the most significant predictors from that regression. This method uses 4 variables only: Base hits by batters, Stolen Bases, Double plays, and Errors. This model produced an F statistic of 225 and $R^2 = 0.283$.

	Estimate	Std. Error	t value	Pr(>|t|)
TEAM_BATTING_H	0.05359	0.002051	26.13	8.068e-132
TEAM_BASERUN_SB	0.02968	0.003554	8.353	1.144e-16
TEAM_FIELDING_DP	-0.08779	0.01224	-7.17	1.008e-12
TEAM_FIELDING_E	-0.027	0.001367	-19.74	3.322e-80
(Intercept)	17.93	3.198	5.605	2.329e-08

Table 13: Fitting linear model: TARGET_WINS ~ TEAM_BATTING_H + TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_FIELDING_E

Observations	Residual Std. Error	R^2	Adjusted R^2
2276	13.34	0.2841	0.2829

Here the residuals and QQ plots can be examined. These plots provide sufficient information to upload the assumptions taken when creating the regression model:




```
[1] ""
```

	Estimate	Std. Error	t value	Pr(>|t|)
TEAM_BATTING_H	0.05359	0.002051	26.13	8.068e-132
TEAM_BASERUN_SB	0.02968	0.003554	8.353	1.144e-16
TEAM_FIELDING_DP	-0.08779	0.01224	-7.17	1.008e-12
TEAM_FIELDING_E	-0.027	0.001367	-19.74	3.322e-80
(Intercept)	17.93	3.198	5.605	2.329e-08

Table 15: Fitting linear model: TARGET_WINS ~ TEAM_BATTING_H + TEAM_BASERUN_SB + TEAM_FIELDING_DP + TEAM_FIELDING_E

Observations	Residual Std. Error	R^2	Adjusted R^2
2276	13.34	0.2841	0.2829

Model 5: Feature engineering and new variables

Using a step-wise methodology, less significant predictors such as caught stealing, pitching walks & hits were left out. To avoid including predictors that were related, the base hits/singles and other multiple base hit predictors were left out. Most of the predictors have coefficients that behave the way we would expect, given the predictors effect on the game. The extra bases predictor ended up having a negative coefficient, when we would expect there to be a positive one. This may be due to the fact that it is similar stats that make up these predictors, and there is collinearity between the total bases and extra bases predictors. Removing extra bases from the model results in a lower coefficient for total bases, and an overall lower adjusted R^2 value for the model.

	Estimate	Std. Error	t value	Pr(>|t|)
TEAM_BATTING_TOT_BASES	0.04703	0.003591	13.1	7.868e-38
TEAM_BATTING_XTRA_BASE	-0.05788	0.01165	-4.969	7.221e-07
TEAM_BATTING_BB	0.01479	0.003173	4.661	3.328e-06
TEAM_BATTING_SO	-0.007654	0.002161	-3.543	0.0004043
TEAM_BASERUN_SB	0.03415	0.004144	8.24	2.869e-16
TEAM_PITCHING_HR	-0.03628	0.01042	-3.483	0.0005049
log(TEAM_FIELDING_E)	-8.255	0.853	-9.677	9.833e-22
TEAM_FIELDING_DP	-0.1334	0.01294	-10.31	2.104e-24
(Intercept)	64.92	7.546	8.604	1.413e-17

Table 17: Fitting linear model: TARGET_WINS ~ TEAM_BATTING_TOT_BASES + TEAM_BATTING_XTRA_BASE + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_PITCHING_HR + log(TEAM_FIELDING_E) + TEAM_FIELDING_DP

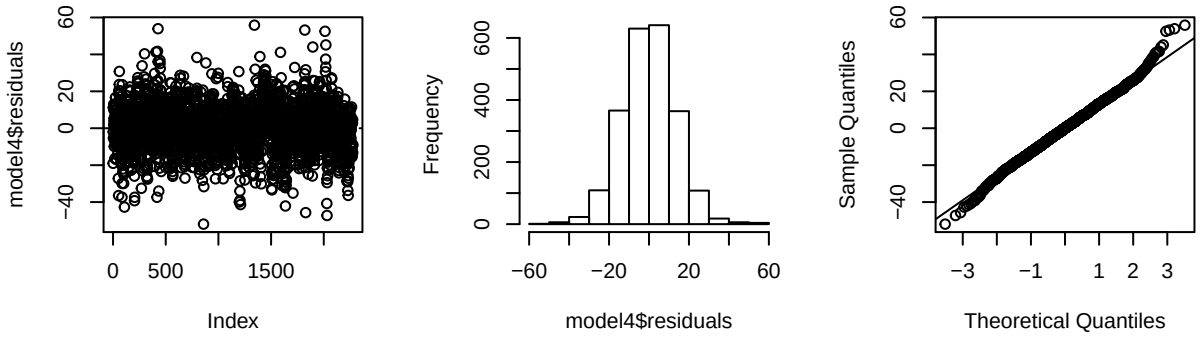
Observations	Residual Std. Error	R^2	Adjusted R^2
2276	13.15	0.3054	0.303

Model Selection and Prediction

The model utilizing only the significant predictors (Model 4) is selected as the best model for prediction of team wins in a 162-game baseball season. While the R^2 value of this model the second-lowest of the five models tested, its high F-score indicates that it is the most statistically significant. Additionally, it is the most parsimonious models tested, and the simplicity lends itself to easier understanding of the model by other users.

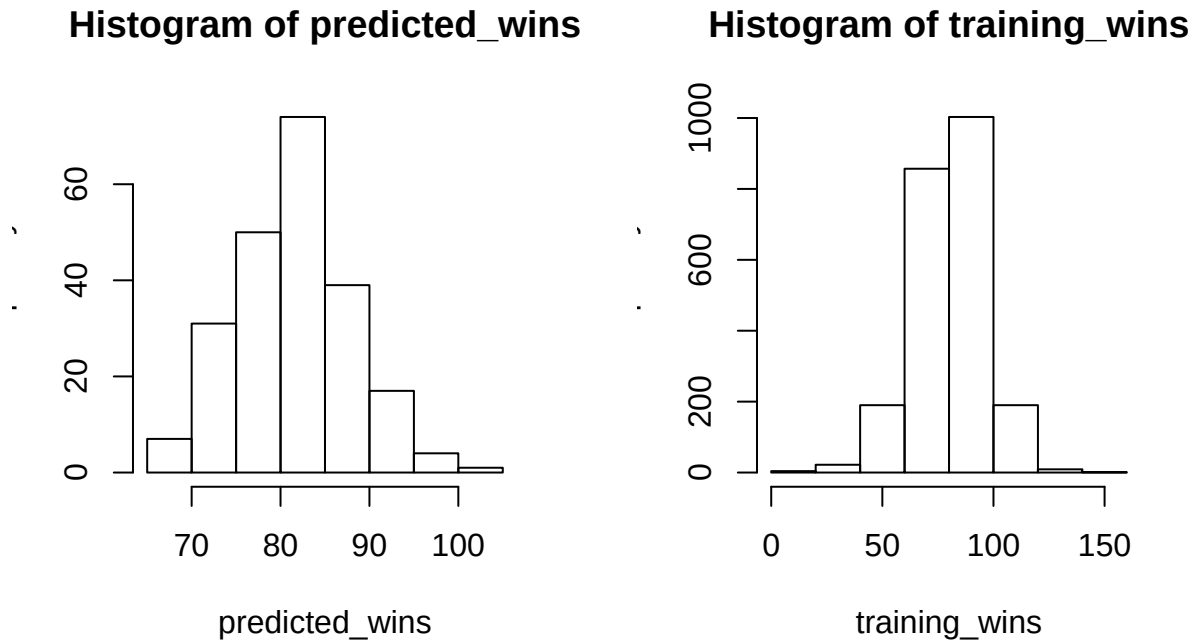
This model has a root-mean-square error of 13.3394, an R^2 of 0.28, and an F-statistic of 225.3. The F statistic has a corresponding p-value of <2.2e-16.

The residual plots for this model are presented below:



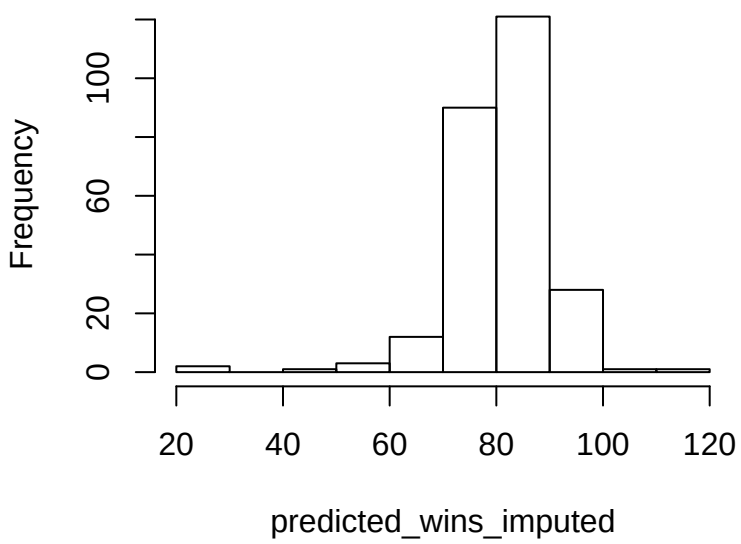
There does not appear to be any pattern in the residuals in the scatterplot, so the condition of linearity can be accepted. The histogram and Q-Q plot indicate that the residuals are roughly normally distributed, albeit with short tails. Finally, the scatterplot and Q-Q plot indicate that the residuals indicate near-constant variability. Because the conditions are met, the validity of the use of a linear model is accepted.

The linear model is applied to an evaluation dataset containing response variables for 259 cases. A histogram of the predicted team wins is presented below.



The predicted wins appear roughly normally distributed, with a slight right-skewness. As expected, the distribution is centered near 82, which represents a 0.500 season. Further investigation shows that the median is indeed roughly 82 wins, with the mean slightly lower at roughly 81 wins.

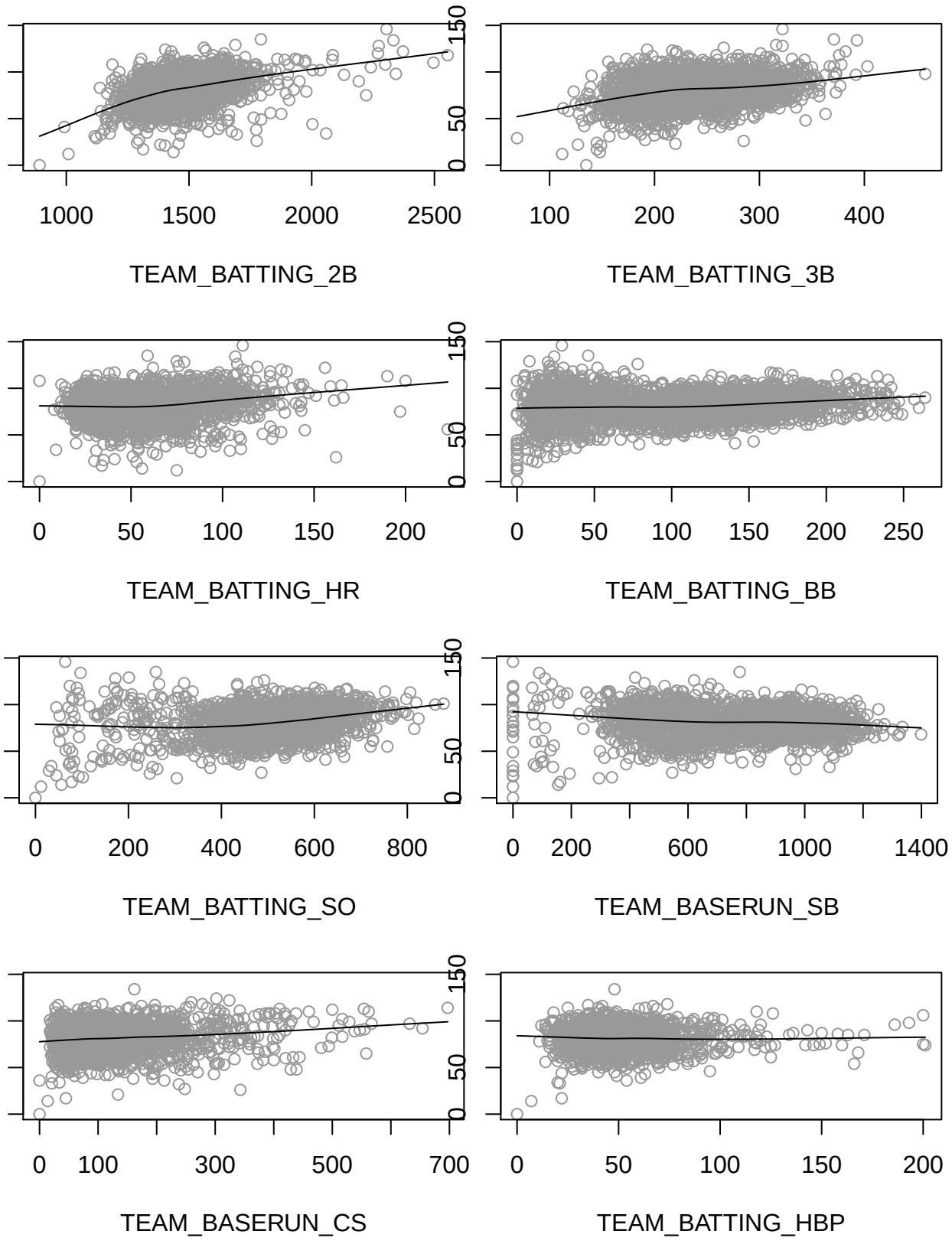
Due to missing values, however, there are 89 missing predictions, representing roughly 34% of the total dataset. In order to allow for predictions of the full 259 cases in the evaluation dataset, missing values are filled with the median for each given missing variable. The linear model is again applied, this time to the evaluation dataset with imputed median values. A histogram of this modified predicted team wins is presented below.

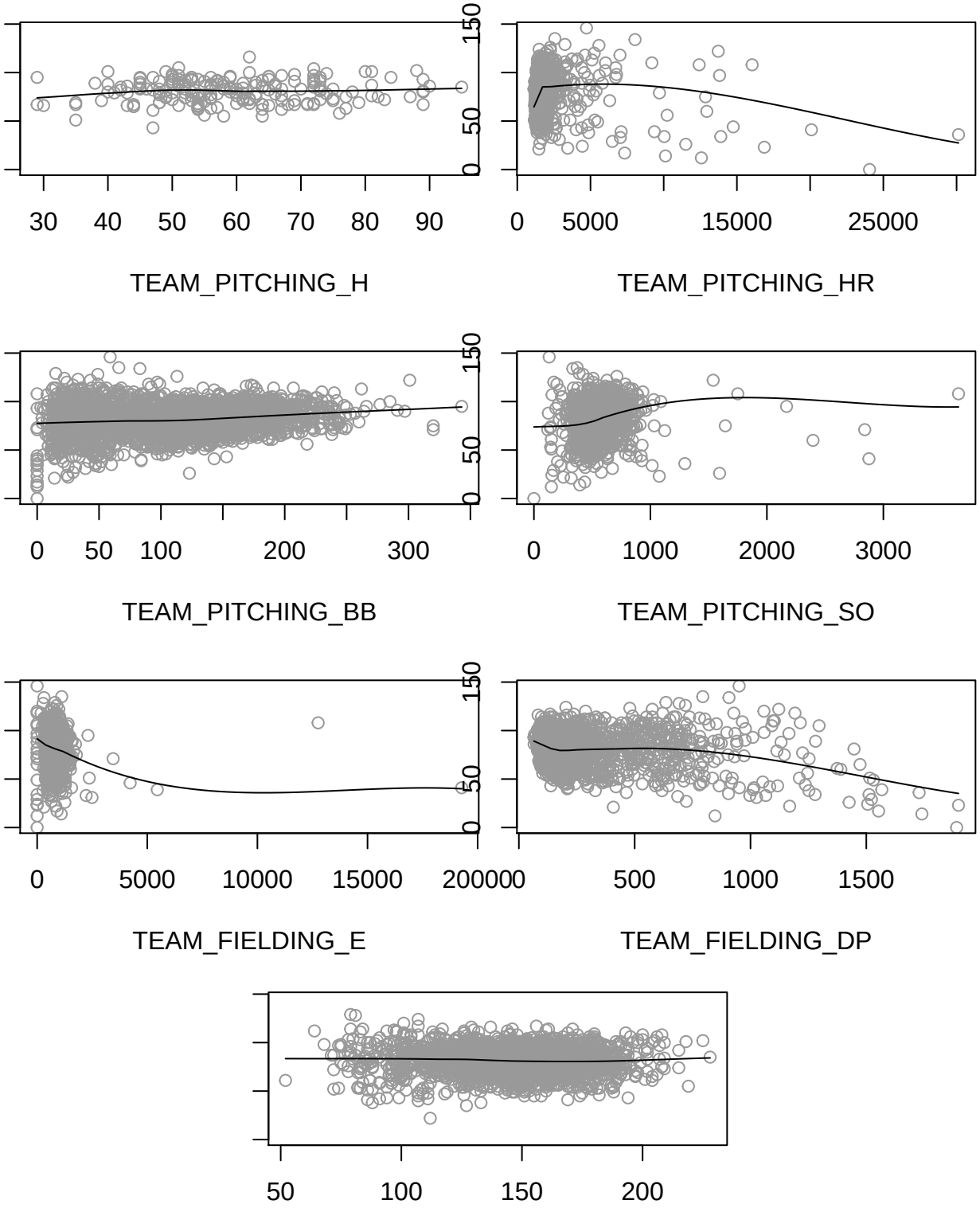


In contrast to the predictions for the raw evaluation dataset, this set of predictions is left-skewed. The median and mean for this set of predictions are both roughly 81. The shape of the distribution, which is seemingly condensed towards the median, suggests that imputation using the median may have introduced a bias towards the center of the distribution (which, again, corresponds to a 0.500 season with 82 wins). Although it is outside the scope of this investigation, more advanced imputation that will avoid the introduction of bias may be advisable for cases with missing predictors.

A comparison of the full sets of predictions for the evaluation dataset is available in Appendix B.

Appendix A – Correlations with TARGET_WINS




0
50
150
30
40
50
60
70
80
90
TEAM_PITCHING_H
0
5000
15000
25000
TEAM_PITCHING_HR
0
50
100
200
300
TEAM_PITCHING_BB
0
1000
2000
3000
TEAM_PITCHING_SO
0
5000
10000
15000
200000
TEAM_FIELDING_E
500
1000
1500
TEAM_FIELDING_DP
50
100
150
200

Appendix B – Index-wise Results from Predictive Model

Index	Predicted Value	Index	Predicted Value
9	67.08	1253	88.46
10	66.92	1261	81.49
14	76.79	1305	80.91
47	87.93	1314	84.94
60	NA	1323	83.96
63	NA	1328	NA
74	NA	1353	74.44
83	79.65	1363	79.96
98	72.16	1371	89.46
120	76.64	1372	81.91
123	74.83	1389	71.76
135	79.84	1393	74.76
138	76.58	1421	91.63
140	79.55	1431	76.85
151	82.18	1437	73.96
153	76.72	1442	71.85
171	74.71	1450	76.95
184	79.5	1463	80.79
193	74.99	1464	81.19
213	90.96	1470	82.37
217	83.31	1471	79.89
226	85.65	1484	84.8
230	78.71	1495	NA
241	72.71	1507	74.52
291	84.58	1514	78.84
294	88.45	1526	73.44
300	NA	1549	86.67
348	79.17	1552	NA
350	86.58	1556	92.06
357	82.53	1564	76.47
367	87.91	1585	97.51
368	83.83	1586	99.86
372	81.34	1590	90.02
382	81.51	1591	97.03
388	79.6	1592	90.23
396	81.4	1603	83.87
398	75.58	1612	78.62
403	85.64	1634	78.59
407	85.99	1645	74.5
410	89.59	1647	81.08
412	82.95	1673	90.1
414	90.63	1674	88.93
436	NA	1687	80.95
440	NA	1688	94.57
476	NA	1700	83.64
479	NA	1708	74.1
481	NA	1713	75.47
501	77.39	1717	71.86
503	72.46	1721	74.5
506	80.86	1730	78
519	79.94	1737	83.81
522	88.59	1748	84.28
550	77.27	1749	84.65
554	73.53	1763	81.6
566	78.52	1768	NA
578	80.76	1778	NA

Index	Predicted Value	Index	Predicted Value
596	89.47	1780	NA
599	77.73	1782	NA
605	65.44	1784	NA
607	81.23	1794	NA
614	89.2	1803	75.12
644	74.99	1804	84.22
692	86.87	1819	80.44
699	82.25	1832	82.05
700	82.59	1833	86.72
716	NA	1844	71.13
721	78.01	1847	79.48
722	86.08	1854	78.67
729	78.97	1855	76.35
731	86.76	1857	83.58
746	86.44	1864	75.94
763	75.01	1865	81.23
774	74.75	1869	75.2
776	81.76	1880	88.4
788	87.44	1881	82.28
789	85.36	1882	80.25
792	80.21	1894	81.85
811	82.77	1896	80.55
835	76.19	1916	NA
837	81.56	1918	NA
861	86.94	1921	NA
862	90.53	1926	NA
863	97.27	1938	80.07
871	79.21	1979	66.36
879	80.84	1982	68.9
887	80.66	1987	84.53
892	80.19	1997	80.63
904	83.76	2004	92.76
909	85.9	2011	80.52
925	88.37	2015	78.7
940	NA	2022	80.69
951	NA	2025	79.49
976	75.16	2027	82.78
981	88.98	2031	77.32
983	88.54	2036	NA
984	84.65	2066	74.46
989	85.56	2073	82.55
995	102.2	2087	80.12
1000	88.33	2092	81.86
1001	87.29	2125	NA
1007	82.95	2148	80.19
1016	74.59	2162	91.87
1027	84.75	2191	74.87
1033	79.25	2203	84.81
1070	78.8	2218	78.45
1081	NA	2221	76.34
1084	NA	2225	85.67
1098	80.81	2232	76.65
1150	84.03	2267	NA
1160	NA	2291	73.84
1169	81.77	2299	88.77
1172	85.58	2317	85.91
1174	93.95	2318	86.37
1176	92.06	2353	85.41

Index	Predicted Value	Index	Predicted Value
1178	83.49	2403	65.12
1184	82.58	2411	88.87
1193	87.06	2415	82.7
1196	80.97	2424	83.92
1199	79.25	2441	74.03
1207	NA	2464	83.58
1218	NA	2465	81.27
1223	71.28	2472	NA
1226	71.61	2481	NA
1227	67.38	2487	NA
1229	70.69	2500	70.5
1241	90.14	2501	74.19
1244	92.75	2520	84.48
1246	78.45	2521	83.32
1248	91.52	2525	79.7
1249	92.58	NA	NA

Appendix C – R Code

```
library(stringr)
library(pander)
library(knitr)
library(ggplot2)
library(gridExtra)
library(reshape2)
library(MASS)
library(leaps)

trainingdata = read.csv("https://raw.githubusercontent.com/aadikalloo/AadiMSDA/master/IS621-Data-Mining/moneyb

createSummaryTable <- function(trainingdata1) {
    ######## Mean and Medians Table
    mean_median_df = data.frame(matrix(0, nrow = ncol(trainingdata1),
        ncol = 2))

    mean2 <- function(x) {
        mean(x, na.rm = TRUE)
    }
    median2 <- function(x) {
        median(x, na.rm = TRUE)
    }

    means = as.data.frame(lapply(trainingdata1, mean2))
    medians = as.data.frame(lapply(trainingdata1, median2))
    lengths = as.data.frame(lapply(trainingdata1, length))

    mean_median_df[, 1] = names(means)
    mean_median_df[, 2] = t(means[1, ])
    mean_median_df[, 3] = t(medians[1, ])
    # mean_median_df[, 4] = t(lengths[1, ])

    names(mean_median_df) = c("VAR_NAME", "MEAN", "MEDIAN")
    # kable(mean_median_df, digits = 2)

    ######## Correlations to Wins

    cor_df = data.frame(matrix(0, nrow = ncol(trainingdata1) -
        2, ncol = 2))

    cors = as.data.frame(cor(trainingdata1$TARGET_WINS, trainingdata1[,
        3:ncol(trainingdata1)], use = "pairwise.complete.obs"))
    cor_df[, 1] = names(cors)
    cor_df[, 2] = t(cors[1, ])

    names(cor_df) = c("VAR_NAME", "CORRELATION TO WINS (r)")

    # kable(cor_df, digits = 2)

    ######## Missing Values per variable
    mv_df = data.frame(matrix(0, nrow = ncol(trainingdata1),
        ncol = 2))

    num_missing <- function(x) {
        sum(is.na(x))
    }
```

```
    missingvalues = as.data.frame(lapply(trainingdata1, num_missing))
    mv_df[, 1] = names(missingvalues)
    mv_df[, 2] = t(missingvalues[1, ])

    names(mv_df) = c("VAR_NAME", "NUM_MISSING")

    # kable(mv_df, digits = 2)


    data_exp = merge(mean_median_df, cor_df, by.x = "VAR_NAME",
        by.y = "VAR_NAME")
    data_exp = merge(data_exp, mv_df, by.x = "VAR_NAME", by.y = "VAR_NAME")
    temp = as.data.frame(cbind(mean_median_df[2, ], NA, NA))
    names(temp) = names(data_exp)
    data_exp = rbind(temp, data_exp)
}


trainingdata_bk = trainingdata

data_exp = createSummaryTable(trainingdata)
kable(data_exp)

par(mfrow = c(1, 3), pin = c(3/2, 3/2))
top3correlations = c(0, 0, 7, 4, 0, 0, 6)
for (plot_count in c(3, 4, 7)) {
    plot(x = trainingdata[, plot_count], y = trainingdata$TARGET_WINS,
        xlab = names(trainingdata)[plot_count], ylab = "Target Wins",
        main = paste0(names(trainingdata)[plot_count], " r = ",
            round(data_exp[top3correlations[plot_count], 4],
                3)))
}

trainingDataRaw = trainingdata
data_no_index <- trainingDataRaw[, c(2:17)]  # delete index

for (i in 1:16) {
    data_no_index[, i][is.na(data_no_index[, i])] = median(data_no_index[,
        i], na.rm = TRUE)
}

df_new = data_no_index

par(mfrow = c(6, 3))

m <- melt(df_new)
p <- ggplot(m, aes(factor(variable), value))
p + geom_boxplot() + facet_wrap(~variable, scale = "free")

trainingDataRaw = trainingdata
data_no_index <- trainingDataRaw[, c(2:17)]  # delete index

for (i in 1:16) {
    data_no_index[, i][is.na(data_no_index[, i])] = median(data_no_index[,
        i], na.rm = TRUE)
}

df_new = data_no_index
# summary(df_new)
```

```r
imp_data = createSummaryTable(df_new)
kable(imp_data)
trainingdata = df_new


a = c("TEAM_BATTING_HBP", "Batters hit by pitch (free base)",
    "Positive", "91.6%", "0.07", "0.31")
b = c("TEAM_BASERUN_CS", "Strikeouts by batters", "Negative",
    "33.9%", "0.02", "0.39")

names(a) = c("Predictor Name", "Description", "Impact", "% Missing",
    "r with Response", "p-Value")
names(b) = names(a)
c = as.data.frame(rbind(a, b))
kable(c)

dfraw <- read.csv(url("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW1/data/moneyball-training-dat
data_no_index <- dfraw[, c(2:17)]  #Remove INDEX column
for (i in 1:16) {
    data_no_index[, i][is.na(data_no_index[, i])] <- median(data_no_index[,
        i], na.rm = TRUE)
}
df_new <- data_no_index
# summary(df_new)


fit <- lm(TARGET_WINS ~ ., df_new)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_PITCHING_BB)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BATTING_HBP)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BASERUN_CS)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_PITCHING_HR)
# pander(summary(fit))

fit <- lm(TARGET_WINS ~ ., df_new)
# pander(summary(fit))
fit <- update(fit, . ~ . - TEAM_PITCHING_BB)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BATTING_HBP)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BASERUN_CS)
# summary(fit)

fit <- update(fit, . ~ . - TEAM_PITCHING_HR)
pander(summary(fit))

model1 = fit

dfraw <- read.csv(url("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW1/data/moneyball-training-dat
dfremove <- subset(dfraw, INDEX == 1347)$INDEX
df <- subset(dfraw, !(INDEX %in% dfremove))
data_no_index <- df[, c(2:17)]  #Remove INDEX column
for (i in 1:16) {
    data_no_index[, i][is.na(data_no_index[, i])] <- median(data_no_index[,
        i], na.rm = TRUE)
}
```

```r
df <- data_no_index
MLB <- read.csv(url("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW1/data/MLB.Stats.1962-2015.csv"

df$TEAM_BATTING_BB <- ifelse(df$TEAM_BATTING_BB < min(MLB$BB) |
    df$TEAM_BATTING_BB > max(MLB$BB), median(df$TEAM_BATTING_BB),
    df$TEAM_BATTING_BB)
df$TEAM_BATTING_H <- ifelse(df$TEAM_BATTING_H < min(MLB$H) |
    df$TEAM_BATTING_H > max(MLB$H), median(df$TEAM_BATTING_H),
    df$TEAM_BATTING_H)
df$TEAM_BATTING_2B <- ifelse(df$TEAM_BATTING_2B < min(MLB$X2B) |
    df$TEAM_BATTING_2B > max(MLB$X2B), median(df$TEAM_BATTING_2B),
    df$TEAM_BATTING_2B)
df$TEAM_BATTING_3B <- ifelse(df$TEAM_BATTING_3B < min(MLB$X3B) |
    df$TEAM_BATTING_3B > max(MLB$X3B), median(df$TEAM_BATTING_3B),
    df$TEAM_BATTING_3B)
df$TEAM_BATTING_HR <- ifelse(df$TEAM_BATTING_HR < min(MLB$HR) |
    df$TEAM_BATTING_HR > max(MLB$HR), median(df$TEAM_BATTING_HR),
    df$TEAM_BATTING_HR)
df$TEAM_BATTING_SO <- ifelse(df$TEAM_BATTING_SO < min(MLB$SO) |
    df$TEAM_BATTING_SO > max(MLB$SO), median(df$TEAM_BATTING_SO),
    df$TEAM_BATTING_SO)
df$TEAM_PITCHING_SO <- ifelse(df$TEAM_PITCHING_SO < min(MLB$SO.1) |
    df$TEAM_PITCHING_SO > max(MLB$SO.1), median(df$TEAM_PITCHING_SO),
    df$TEAM_PITCHING_SO)
df$TEAM_BASERUN_SB <- ifelse(df$TEAM_BASERUN_SB < min(MLB$SB) |
    df$TEAM_BASERUN_SB > max(MLB$SB), median(df$TEAM_BASERUN_SB),
    df$TEAM_BASERUN_SB)

fit <- lm(TARGET_WINS ~ ., df)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_PITCHING_SO)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BATTING_3B)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BATTING_HBP)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_PITCHING_BB)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BASERUN_CS)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_PITCHING_H)
# summary(fit)
fit <- update(fit, . ~ . - TEAM_BATTING_HR)
pander(summary(fit))

model2 = fit

# Model 3 using 1880- Data
dfraw <- read.csv(url("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW1/data/moneyball-training-dat
dfremove <- subset(dfraw, INDEX == 1347)$INDEX
df <- subset(dfraw, !(INDEX %in% dfremove))
df <- df[, -c(1, 10, 11, 17)]  #Remove caught stealing and hit by pitcher variables and fielding double play,

df$TEAM_BASERUN_SB <- ifelse(df$TEAM_BASERUN_SB < 13 | df$TEAM_BASERUN_SB >
    638, NA, df$TEAM_BASERUN_SB)
# http://www.baseball-almanac.com/recbooks/rb_stba2.shtml
df$TEAM_BATTING_3B <- ifelse(df$TEAM_BATTING_3B < 11 | df$TEAM_BATTING_3B >
    153, NA, df$TEAM_BATTING_3B)
# http://www.baseball-almanac.com/rb_trip2.shtml
```

```
df$TEAM_BATTING_HR <- ifelse(df$TEAM_BATTING_HR < 3 | df$TEAM_BATTING_HR >
    264, NA, df$TEAM_BATTING_HR)
# http://www.baseball-almanac.com/recbooks/rb_hr7.shtml
df$TEAM_BATTING_SO <- ifelse(df$TEAM_BATTING_SO < 308 | df$TEAM_BATTING_SO >
    1535, NA, df$TEAM_BATTING_SO)
# http://www.baseball-almanac.com/recbooks/rb_strike2.shtml
df$TEAM_PITCHING_SO <- ifelse(df$TEAM_PITCHING_SO < 333 | df$TEAM_PITCHING_SO >
    1450, NA, df$TEAM_PITCHING_SO)
# http://www.baseball-almanac.com/recbooks/rb_strik.shtml
df$TEAM_BASERUN_SB <- ifelse(df$TEAM_BASERUN_SB < 13 | df$TEAM_BASERUN_SB >
    638, NA, df$TEAM_BASERUN_SB)
# http://www.baseball-almanac.com/recbooks/rb_stba2.shtml

fit <- lm(TARGET_WINS ~ ., df)
# summary(fit)
fit1 <- update(fit, . ~ . - TEAM_BATTING_H)
# summary(fit1)

fit2 <- update(fit1, . ~ . - TEAM_PITCHING_HR)
# summary(fit2)
fit3 <- update(fit2, . ~ . - TEAM_PITCHING_SO)
# summary(fit3)


fit4 <- update(fit3, . ~ . - TEAM_BATTING_2B)
# summary(fit4)
fit5 <- update(fit4, . ~ . - TEAM_PITCHING_BB)
# summary(fit5)
fit6 <- update(fit5, . ~ . - TEAM_PITCHING_H)
# summary(fit6)

df$TEAM_BATTING_1B <- df$TEAM_BATTING_H - df$TEAM_BATTING_2B -
    df$TEAM_BATTING_3B - df$TEAM_BATTING_HR  #Singles - 1st Base Hits
fit7 <- update(fit6, . ~ . + TEAM_BATTING_1B)
pander(summary(fit7))

# pairs(~TARGET_WINS + TEAM_BATTING_3B + TEAM_BATTING_HR +
# TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB
# +TEAM_FIELDING_E, df, upper.panel = panel.smooth,
# lower.panel = NULL)


# These are variables that I tried but didn't turn out to be
# valuable

df$TEAM_BATTING_HRP <- df$TEAM_BATTING_HR/df$TEAM_BATTING_H  #Home runs as a percentage of base hits
# summary(update(fit6,.~.+TEAM_BATTING_HRP))

df$TEAM_BATTING_HSO <- df$TEAM_BATTING_H/df$TEAM_BATTING_SO  #Ratio of hits to strikeouts
# summary(update(fit6,.~.+TEAM_BATTING_HSO))
model3 = fit7

model4 <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BASERUN_SB +
    TEAM_FIELDING_DP + TEAM_FIELDING_E, df_new)
pander(summary(model4))

# pairs(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BASERUN_SB +
# TEAM_FIELDING_E + TEAM_FIELDING_DP, df_new, upper.panel =
# panel.smooth, lower.panel = NULL)
```

```r
par(mfrow = c(2, 2))
plot(model4)
# par(mfrow=c(1,1)); plot(model4$residuals); abline(h=0,
# col='red')
par(mfrow = c(1, 1))
par(mfrow = c(1, 1), pin = c(3, 3/2))
print("")
pander(summary(model4))

# plot(fitted(model4), residuals(model4)) abline(h=0,
# col='red')

# Get Singles from Base Hits
df_new$TEAM_BATTING_1B <- with(df_new, TEAM_BATTING_H - TEAM_BATTING_2B -
    TEAM_BATTING_3B - TEAM_BATTING_HR)

# Create Extra Bases and Total Bases predictors
df_new$TEAM_BATTING_XTRA_BASE <- with(df_new, TEAM_BATTING_2B +
    TEAM_BATTING_3B + TEAM_BATTING_HR)

df_new$TEAM_BATTING_TOT_BASES <- with(df_new, TEAM_BATTING_1B +
    (2 * TEAM_BATTING_2B) + (3 * TEAM_BATTING_3B) + (4 * TEAM_BATTING_HR))


lt_model3 <- lm(TARGET_WINS ~ TEAM_BATTING_TOT_BASES + TEAM_BATTING_XTRA_BASE +
    TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_SB + TEAM_PITCHING_HR +
    log(TEAM_FIELDING_E) + TEAM_FIELDING_DP, df_new)


# boxcox((TARGET_WINS + 1) ~ TEAM_FIELDING_E, data=df_new,
# lambda = seq(1.2, 1.5, length=10)) #MASS package

pander(summary(lt_model3))
stepReduced = model4

par(mfrow = c(1, 3))
plot(stepReduced$residuals, main = NULL)
hist(stepReduced$residuals, main = NULL)
qqnorm(stepReduced$residuals, main = NULL)
qqline(stepReduced$residuals)
par(mfrow = c(1, 1))

evaluation_data <- read.csv("https://raw.githubusercontent.com/dsmilo/DATA621/master/HW1/data/moneyball-evalua
par(mfrow = c(1, 1), pin = c(2, 2))

predicted_wins <- predict(stepReduced, evaluation_data)
hist(predicted_wins)
training_wins = trainingdata_bk$TARGET_WINS
hist(training_wins, breaks = 8)

evaluation_data_imputed <- evaluation_data
for (i in 2:ncol(evaluation_data_imputed)) {
    evaluation_data_imputed[, i][is.na(evaluation_data_imputed[,
        i])] <- median(evaluation_data_imputed[, i], na.rm = TRUE)
}

par(mfrow = c(1, 1), pin = c(3, 2))
predicted_wins_imputed <- predict(model4, evaluation_data_imputed)
```

```r
hist(predicted_wins_imputed)

par(pin = c(3, 1))
for (var_count in 3:17) {
    # plot(x = trainingdata_bk[, var_count], y =
    # trainingdata$TARGET_WINS, xlab =
    # names(trainingdata)[var_count], ylab = 'Target Wins')
    scatter.smooth(x = trainingdata_bk[, var_count], y = trainingdata$TARGET_WINS,
        xlab = names(trainingdata)[var_count], ylab = "Target Wins",
        col = "#999999")
}

appendixB = data.frame(matrix(NA, nrow = 130, ncol = 4))
appendixB[, 1] = evaluation_data$INDEX[1:130]
appendixB[, 2] = predicted_wins[1:130]
appendixB[, 3] = c(evaluation_data$INDEX[131:259], NA)
appendixB[, 4] = c(predicted_wins[131:259], NA)
# appendixB = appendixB[-130:259,]
names(appendixB) = c("Index", "Predicted Value", "Index", "Predicted Value")

pander(appendixB)
```