Assignment #1
Kevin Wong
PREDICT 411 Section 58
Kaggle name: kevinwong

INTRODUCTION:

The objective of this assignment is to analyze baseball team data from 1871 to 2006 and to predict the number of wins for each baseball team in a season. This will be accomplished by performing exploratory data analysis to understand the data such as the distribution, relationships between variables, and missing values. After exploring the data, variables with missing values will need to be fixed in order to produce the models. Models will be generated using stepwise regression. Once there are several viable models, the best one will be selected based on performance, interpretability, and intuition.

DATA EXPLORATION

Based on initial observations, the data contains 2276 teams with a variety of baseball performance statistics. Figure 1 show summary statistics of the target wins. The noteworthy statistics are the average number of wins in a season is 81 games, the median number of wins in a season is 82 games, and the standard deviation is 16 games.

Basic Statistical Measures			
Location		**Variability**	
Mean	80.79086	**Std Deviation**	15.75215
Median	82.00000	**Variance**	248.13031
Mode	83.00000	**Range**	146.00000
		Interquartile Range	21.00000

Figure 1

A quick look at Figure 2 will reveal the distribution of the target wins. The distribution is approximately normal with a majority of the target wins falling in the center of the distribution. The approximate normal distribution is confirmed by the QQ plot below the distribution plot.

Most of the target wins fall on the line in the QQ plot with some data points diverging at the ends. This indicates possibility of outliers where some teams are winning more games or losing more games than what is expected in the normal range. In the boxplot, there are points that fall outside the whiskers which confirms our suspicions of outliers seen in the QQ plot.

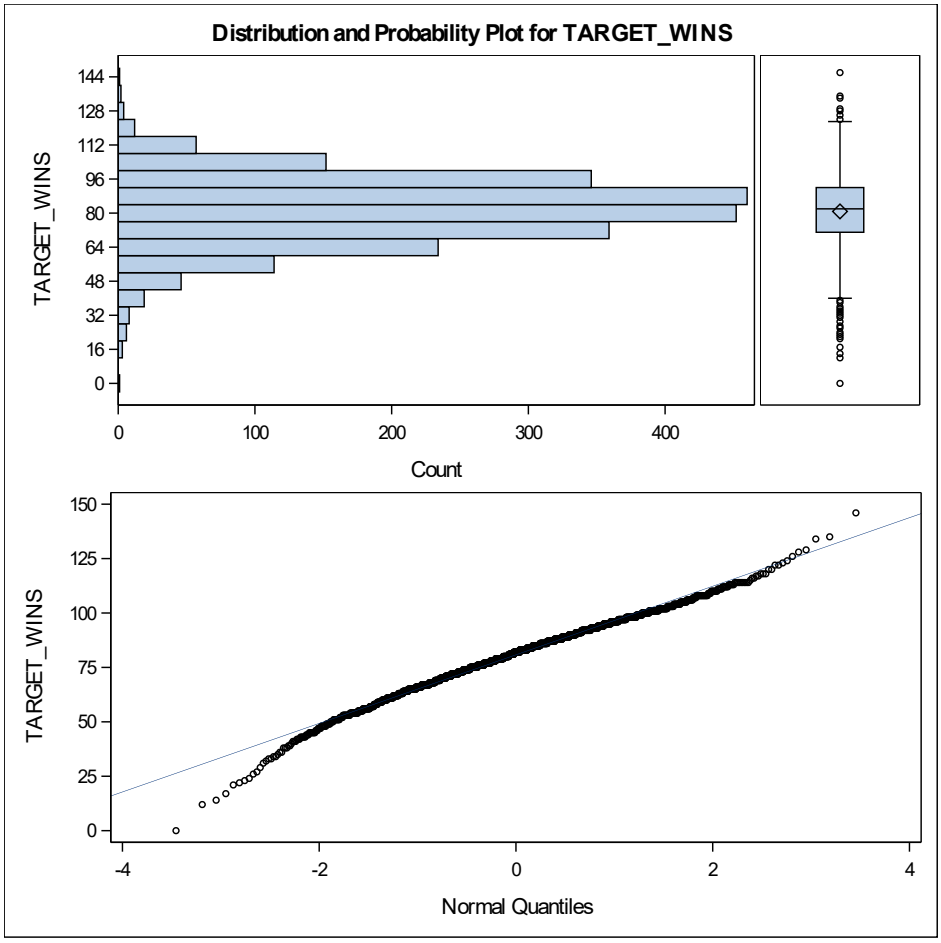


Figure 2

By examining the target wins variable in detail, there is a clear guideline of how many wins each team should approximately win. Most teams will likely win the average number of games (81), but there will be some variability from the average with some teams winning more or less than 81 games.

The other variables also play an important role in understanding the data. In Figure 3, summary statistics are presented for all the variables. Although this is very high level view of the data, it is sufficient in getting the gist of each variable's distribution. For example, the average Base Hits by batters per team is 1469 with the minimum base hits at 891 and maximum base hits at 2554. Remember that the dataset contains baseball statistics on 2276 teams. In the case of the Base Hits by batters variable, every team has data. However, this isn't the case for some variables, i.e. Strikeouts by batters, Batters hit by pitch, Stolen bases, Caught stealing, Strikeouts by pitchers, and Double plays. This can be determined based on examining the "N" column of Figure 3. Any variable with less than 2276 indicates missing values. Notice how Batters hit by pitch only has 191 non-missing fields out of 2276 total fields. It may be better to drop this variable than to fix it prior to building a model. Missing values will be dealt with in the data preparation section of this report.

Simple Statistics							
Variable	**N**	**Mean**	**Std Dev**	**Sum**	**Minimum**	**Maximum**	**Label**
TARGET_WINS	2276	80.79086	15.75215	183880	0	146.00000	
INDEX	2276	1268	736.34904	2887023	1.00000	2535	
TEAM_BATTING_H	2276	1469	144.59120	3344058	891.00000	2554	Base Hits by batters
TEAM_BATTING_2B	2276	241.24692	46.80141	549078	69.00000	458.00000	Doubles by batters
TEAM_BATTING_3B	2276	55.25000	27.93856	125749	0	223.00000	Triples by batters
TEAM_BATTING_HR	2276	99.61204	60.54687	226717	0	264.00000	Homeruns by batters
TEAM_BATTING_BB	2276	501.55888	122.67086	1141548	0	878.00000	Walks by batters
TEAM_BATTING_SO	2174	735.60534	248.52642	1599206	0	1399	Strikeouts by batters
TEAM_BASERUN_SB	2145	124.76177	87.79117	267614	0	697.00000	Stolen bases
TEAM_BASERUN_CS	1504	52.80386	22.95634	79417	0	201.00000	Caught stealing
TEAM_BATTING_HBP	191	59.35602	12.96712	11337	29.00000	95.00000	Batters hit by pitch
TEAM_PITCHING_H	2276	1779	1407	4049483	1137	30132	Hits allowed
TEAM_PITCHING_HR	2276	105.69859	61.29875	240570	0	343.00000	Homeruns allowed
TEAM_PITCHING_BB	2276	553.00791	166.35736	1258646	0	3645	Walks allowed
TEAM_PITCHING_SO	2174	817.73045	553.08503	1777746	0	19278	Strikeouts by pitchers
TEAM_FIELDING_E	2276	246.48067	227.77097	560990	65.00000	1898	Errors
TEAM_FIELDING_DP	1990	146.38794	26.22639	291312	52.00000	228.00000	Double Plays

Figure 3

It is possible that not all variables will need to be used in creating an accurate model. In Figure 4, a correlation value is computed for each variable against target wins. The correlation values are a measure of how strong the linear relationship is between variables and whether it is positive or negative. Some variables are highly correlated with target wins, while other variables are not. For example, Base Hits by batters has a value of 0.38877 which is high while Caught stealing is barely correlated with target wins with a value of 0.0224. There is also a column for p-values which indicates whether the correlations are significant. We can use a decision rule of 95% meaning any variable with a p-value of less than 0.05 is significant. It appears that Strikeouts by batters (TEAM_BATTING_SO), Caught stealing (TEAM_BASERUN_CS), Batters hit by pitch (TEAM_BATTING_HBP), and Double plays (TEAM_FIELDING_DP) do not meet our decision rule and could be excluded from use.

Correlation with TARGET_WINS		
Variable	**Correlation**	**P-value**
TEAM_BATTING_H	0.38877	<.0001
TEAM_BATTING_2B	0.2891	<.0001
TEAM_BATTING_3B	0.14261	<.0001
TEAM_BATTING_HR	0.17615	<.0001
TEAM_BATTING_BB	0.23256	<.0001
TEAM_BATTING_SO	-0.03175	0.1389
TEAM_BASERUN_SB	0.13514	<.0001
TEAM_BASERUN_CS	0.0224	0.3853
TEAM_BATTING_HBP	0.0735	0.3122
TEAM_PITCHING_H	-0.10994	<.0001
TEAM_PITCHING_HR	0.18901	<.0001
TEAM_PITCHING_BB	0.12417	<.0001
TEAM_PITCHING_SO	-0.07844	0.0003
TEAM_FIELDING_E	-0.17648	<.0001
TEAM_FIELDING_DP	-0.03485	0.1201

Figure 4

Before entirely excluding variables, it is a good idea to transform the data by fixing missing values or combining variables and reexamine the viability of those variables for predicting wins.

DATA PREPARATION

Missing values need to be handled before building models. They can be handled by either dropping the records, dropping the entire variable, or imputation. In this case, it was determined that Batters hit by pitch variable should be dropped altogether prior to model building because it has too many missing values to properly impute. All other variables with missing values will be considered for the model because a majority of the records are not missing. These variables will be imputed.

When it comes to fixing missing values, there are several methods at our disposal. The first technique is to impute the missing values with the mean values of each variable. Earlier in Figure 3, the mean value for Base Hits by batters was 1469. Any missing value in Base Hits by batters will be imputed with 1469. The same procedure will be used for the other variables with missing values. Most of the time, mean imputation will lead to good results. Later this will be our first model.

The second technique for imputing missing values is to use a decision tree. This is slightly more involved, but will likely give the better results. A decision tree will be created in R for each variable with missing values. In mean imputation, a fixed value is used for missing values of an entire variable whereas in decision tree imputation, a value is used based on certain conditions. For example, in Figure 5,

Strikeouts by pitchers would be given the value 5663 if Walks allowed (TEAM_PITCHING_BB) is greater than 1618.

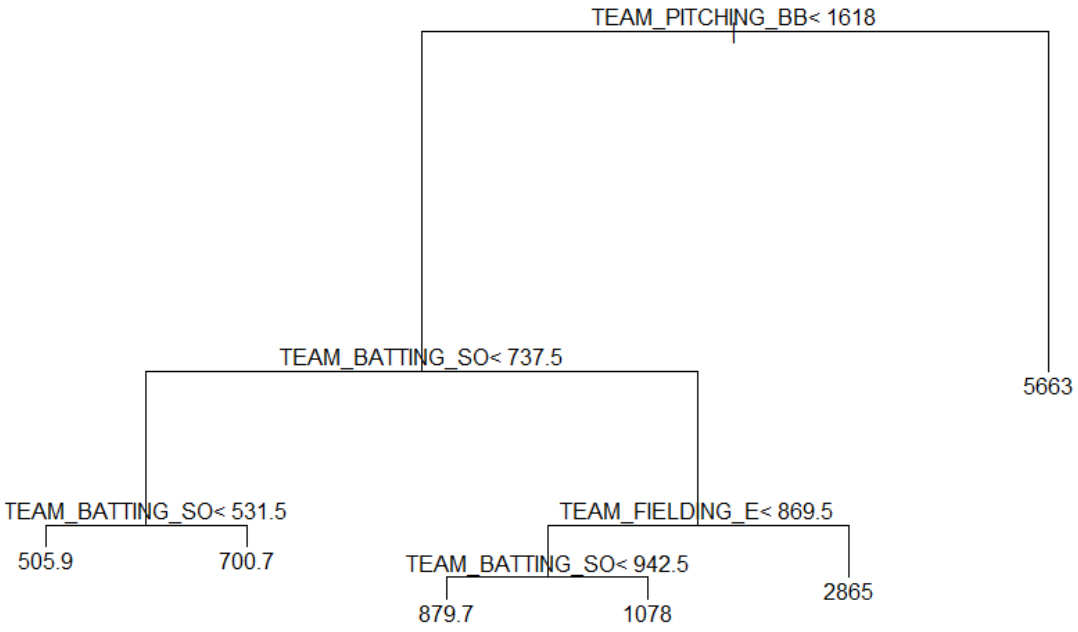


Figure 5

Another way of transforming the data is to drop variables. As mentioned earlier, Batters hit by pitch was dropped from the dataset because it contains too many missing values. It is better to remove this variable than to impute it with too little data.

BUILD MODELS:

MODEL 1: MEAN FULL MODEL

This is a full model containing all the variables with the mean used for missing values. This is a good starting model to determine how well each variable helps predict wins. The mean is generally an adequate guess for missing values. In this model, no selection technique is used. All variables are manually included.

The overall p-value for Model 1 is less than 0.0001, which indicates a significant model in predicting wins. The Adjusted R-Squared and Mean Square Error (MSE) will be the metrics used to determine the best model. A higher Adjusted R-Squared is better and a lower MSE is better. In this case, the Adjusted R-Squared is 0.3148 and MSE is 170.03, which will be the current benchmark. Here is the equation for predicting wins using Model 1:

WINS =	+	25.06831		
	+	0.04824	*	Base Hits by batters
	-	0.02004	*	Doubles by batters
	+	0.06040	*	Triples by batters
	+	0.05298	*	Homeruns by batters
	+	0.01041	*	Walks by batters
	-	0.00935	*	Strikeouts by batters
	+	0.02946	*	Stolen bases
	-	0.01173	*	Caught stealing
	-	0.00073149	*	Hits allowed
	+	0.01481	*	Homeruns allowed
	+	0.00008066	*	Walks allowed
	+	0.00284	*	Strikeouts by pitchers
	-	0.02118	*	Errors
	-	0.12121	*	Double plays

Most of Model 1 makes sense as positive measures of success like Base Hits, Triples, Homeruns, Walks by batters, and Stolen bases are positive coefficients in the equation while negative measures of success like Strikeouts by batters, Caught stealing, Hits allowed, and Errors are negative coefficients in the equation. All these values make intuitive sense.

On the other hand, Doubles and Double plays are shown as negative coefficients when they should have a positive impact on wins. Also, Homeruns allowed and Walks allowed are shown as positive coefficients when they should have negative impact on wins. These values are counterintuitive. The counterintuitive parts of the model may need to be further investigated if this model were to be chosen for deployment. However, for now, the model will be kept as a benchmark despite certain measures not making sense.

Figure 6 shows some diagnostics on the fit of Model 1. The diagnostics give information about the validity of the model by examining the residuals (error). Some notable plots to examine are the residual versus predicted values. The data seem to be clustered in a ball which indicates there might be some

noise in the data. It is preferred that the residuals are randomly distributed throughout. However, it is not too extreme so, but it is possible some transformation of the variables may fix it. Other diagnostics like the QQ plot show the residuals to be relatively normal and the Cook's D reveals one or two points that are possible outliers. Overall, the diagnostics reveal little to be concerned about.

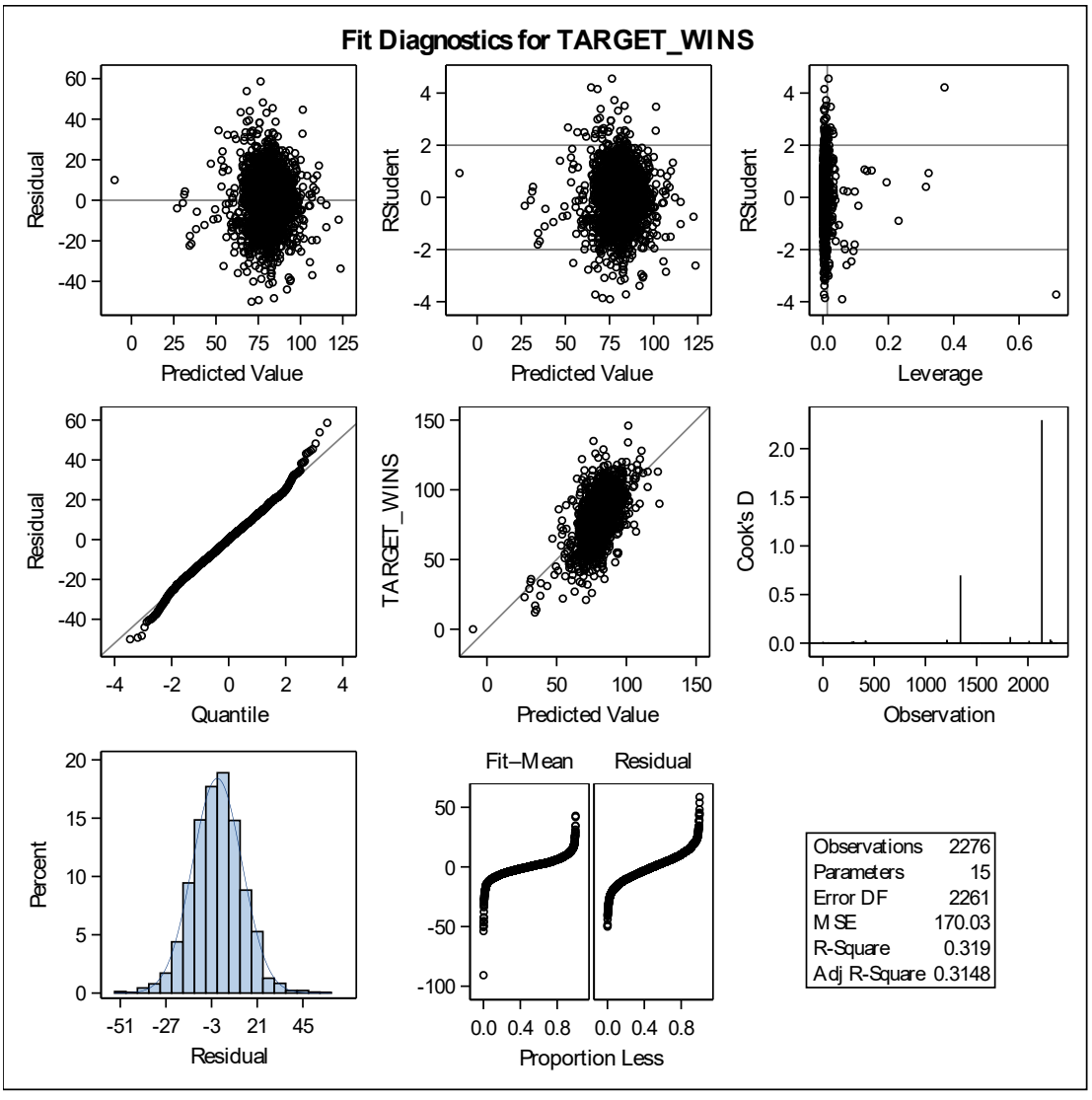


Figure 6

MODEL 2: MEAN WITH STEPWISE

Model 2 is a mean model using the stepwise selection technique. Just like in Model 1, the mean is used to fill in any missing values. Instead of including all variables for consideration, the stepwise selection, variables are added to and removed from the model based on some criteria. In this case, the adjusted R-Squared is used. If the adjusted R-Squared improves, the variable is added to the model, if it is not, then it is removed. This is useful in possibly creating a simpler model. Model #2 is a significant model with a p-value of less than 0.0001. The Adjusted R-Squared is 0.3154 and MSE is 169.88. The resulting equation for Model #2 is:

WINS =	+	23.72283		
	+	0.04845	*	Base Hits by batters
	-	0.02049	*	Doubles by batters
	+	0.06238	*	Triples by batters
	+	0.06976	*	Homeruns by batters
	+	0.01073	*	Walks by batters
	-	0.00930	*	Strikeouts by batters
	+	0.02875	*	Stolen bases
	-	0.00068943	*	Hits allowed
	+	0.00288	*	Strikeouts by pitchers
	-	0.02067	*	Errors
	-	0.12118	*	Double plays

The majority of coefficient values make intuitive sense. Base Hits, Triples, Homeruns, Walks by batters, Stolen bases, and Strikeouts by pitchers contribute positively to wins and Strikeouts by batters, Hits allowed and Errors contribute negatively to wins as expected.

Again, Doubles and Double plays seem counterintuitive as the model shows them having negative impact on wins when it is common knowledge they have positive impact. However, further investigation of these variables is needed to determine the true impact. A possible remedy is to check for multicollinearity among variables. If there is no explanation, then it may be better to drop the variables from the model, which will be illustrated later in Model 4.

When comparing the two models so far, Model 2 has 11 variables compared to Model 1's 16 variables. There is also an improvement in the Adjusted R-Squared from 0.3148 to 0.3154 and MSE from 170.03 to 169.88. At this point in time, Model 2 is the better model based on the Adjusted R-Squared, MSE, and number of variables.

The fit diagnostics in Figure 7 for Model 2 look very similar to Model 1. The previous assessment applies here. The residuals versus predicted values show some pattern, but not too extreme. The QQ plot is approximately normal with skewness to the right. The Cook's D reveals possible outliers. But overall, the validity of the model is good.

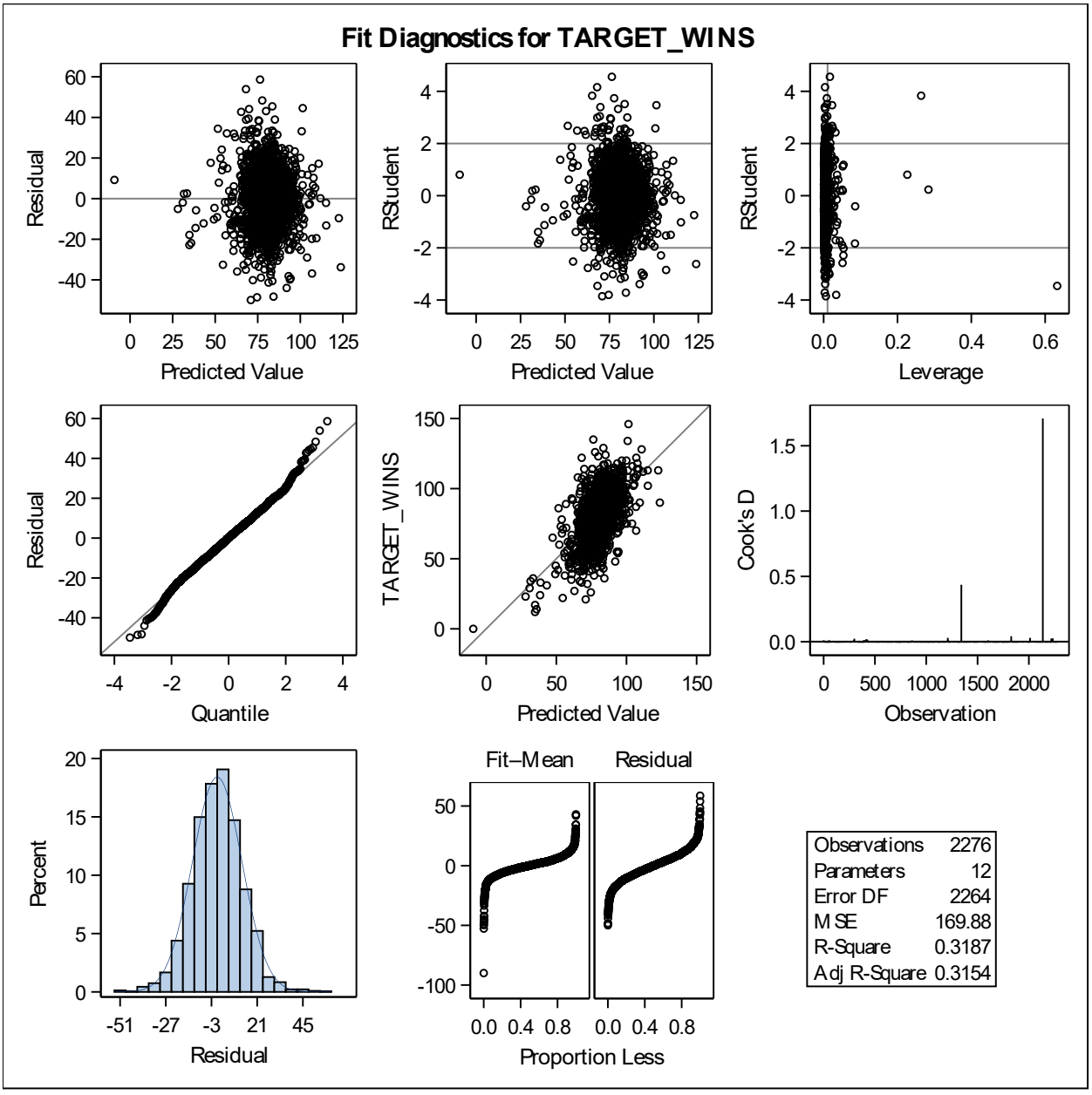


Figure 7

MODEL 3: MEDIAN WITH STEPWISE

Earlier in the exploration of the data, the analysis revealed the possibility of outliers present in the data. Because the mean is highly influenced by outliers, this model attempts to remedy that by using the median to impute missing values. Model 3 is a significant model based on the p-value of less than 0.0001. The Adjusted R-Squared is 0.3117 and MSE is 170.78. The resulting equation for Model 3 is:

WINS =	+	22.34404		
	+	0.04909	*	Base Hits by batters
	-	0.02137	*	Doubles by batters
	+	0.06658	*	Triples by batters
	+	0.06740	*	Homeruns by batters
	+	0.01155	*	Walks by batters
	-	0.00852	*	Strikeouts by batters
	+	0.02492	*	Stolen bases
	-	0.00077704	*	Hits allowed
	+	0.00297	*	Strikeouts by pitchers
	-	0.01901	*	Errors
	-	0.12179	*	Double plays

Model 3 echoes the same issues with the previous two models. That is the equation contains counterintuitive coefficients specifically for Doubles and Double plays. Again, this could be an issue with multicollinearity or other combination of factors.

Compared to Model 2, Model 3 performs slightly worse with Adjusted R-Squared of 0.3117 compared to Model 2's Adjusted R-Squared of 0.3154. Because using median to fix missing values does not seem to improve model performance, Model 2 is the best current model.

Figure 8 shows similar fit diagnostics to the previous two models. The residual versus predicted value plot show some clustering, but there is no discernible pattern to be concerned about. The residuals are approximately normal as revealed by the QQ plot with some right skewing. And the Cook's D shows one or two potential outlier data points. Overall, there is no cause for concern in regards to the validity of Model 3.

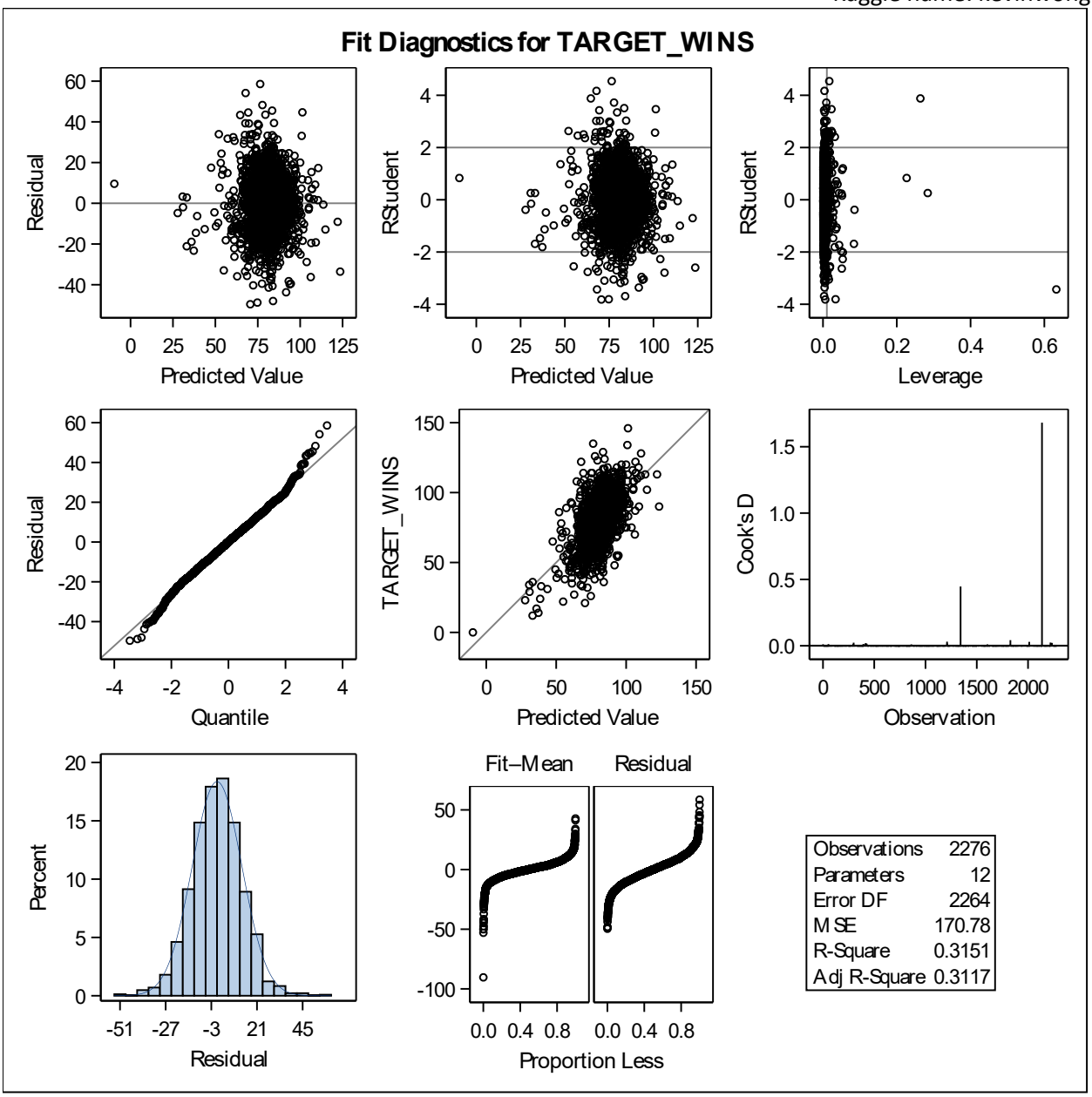


Figure 8

MODEL 4: DECISION TREE AND STEPWISE

Decision trees are extremely useful in fixing missing values. In Model 4, a decision tree will be used. The missing values will be filled in based on conditions of other variables. Decision trees can provide more accurate guesses which can produce more accurate models. Stepwise variable selection will be used again because of its ability to add and remove variables at each iteration.

Model 4 is significant in predicting wins with a p-value less than 0.0001. The Adjusted R-Squared is 0.3373 and MSE is 164.44. The resulting equation for Model 4 is:

WINS =	+	31.16673		
	+	0.04432	*	Base Hits by batters
	-	0.02202	*	Doubles by batters
	+	0.03185	*	Triples by batters
	+	0.07665	*	Homeruns by batters
	+	0.00660	*	Walks by batters
	-	0.01649	*	Strikeouts by batters
	+	0.05665	*	Stolen bases
	-	0.03358	*	Caught stealing
	+	0.00221	*	Strikeouts by pitchers
	-	0.03366	*	Errors
	-	0.06881	*	Double plays

The equation makes sense for most variables. Base hits, Triples, Homeruns, Walks by batters, Stolen bases, and Strikeouts by pitchers are all measures of good performance which is reflected by the equation. Strikeouts by batters, Caught stealing, and Errors are measures of poor performance which is reflected by the equation. The same counterintuitive coefficients for Doubles and Double plays still exist in this model. This requires further investigation of multicollinearity or consultation from experts on baseball statistics.

Model 4 has shown noticeable improvement in performance compared to the previous models. By using decision trees to fix missing values, Adjusted R-Squared has improved to 0.3373 from the previous best of 0.3154 (Model 2).

Another issue to note is that certain variables included in the model are on the border of statistical significance meaning it is possible to exclude them from the model. These variables are Doubles by batters, Triples by batters, Walks by batters, and Caught stealing. In the next and final model, the decision rule will be to exclude variables that do not meet a p-value threshold of 0.01 or less.

The diagnostics in Figure 9 look good. The residuals are fairly randomly distributed and approximately normal. There are some instances of outliers revealed by the QQ plot and Cook's D plot. But again, Model 4 does relatively well based on examination of the diagnostics.

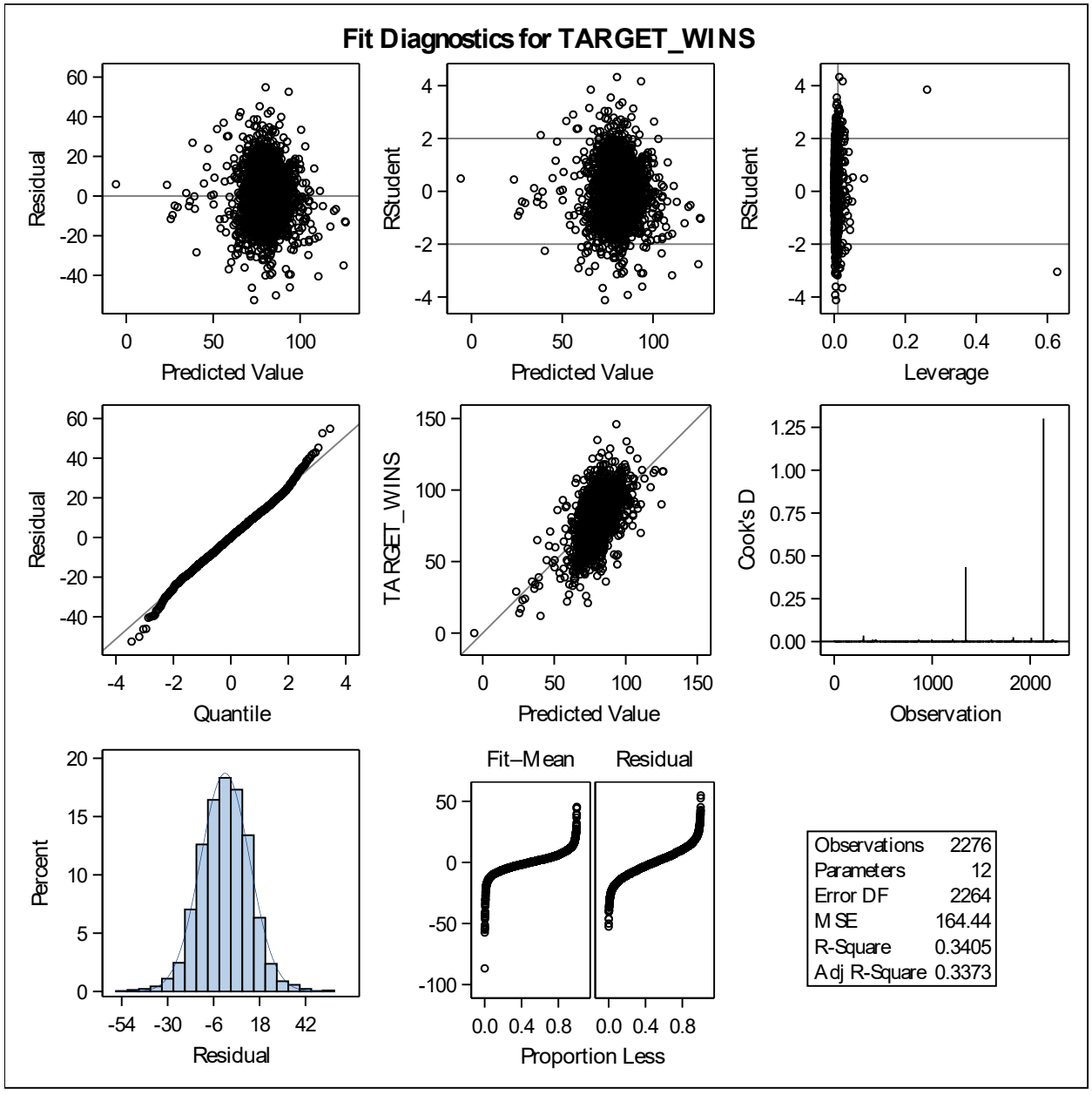


Figure 9

MODEL 5:

The final model will use decision trees for fixing missing values, stepwise for variable selection, and remove variables that are counterintuitive and not statistically significant. The goal is to create a parsimonious model that does not sacrifice performance.

Model 5 is significant in predicting wins with a p-value of less than 0.0001. The Adjusted R-Squared is 0.3263 and MSE is 167.18. The resulting equation of Model 5 is:

WINS =	+	25.03673		
	+	0.04049	*	Base Hits by batters
	+	0.07812	*	Homeruns by batters
	-	0.01868	*	Strikeouts by batters
	+	0.06323	*	Stolen bases
	+	0.00191	*	Strikeouts by pitchers
	-	0.03251	*	Errors

Model 5 is simpler than all previous models. In comparison to Model 4, counterintuitive variables (Doubles by batters and Double plays) were dropped in Model 5 to produce a fully intuitive model. Doubles by batters, Triples by batters, Walks by batters, and Caught stealing were also dropped from the model because they were not statistically significant based on 0.01 threshold. As a result, Model 5 is simpler and more intuitive.

Base Hits, Homeruns, Stolen bases, and Strikeouts by pitchers contribute positively towards wins as we expect. Strikeouts by batters and Errors contribute negatively towards wins as we expect. This model has no counterintuitive variables.

The drawback to this model is that the performance suffered slightly compared to Model 4. Model 5 has an Adjusted R-Squared of 0.3263 compared with Model 4's Adjusted R-Squared of 0.3373. The MSE is also worse at 167.18 compared to Model 4's MSE of 164.44

The fit diagnostics in Figure 10 reveal little to be concerned about. The residuals are distributed fairly randomly and are approximately normal. There are some outliers present, but nothing invalidating the fit of this model.

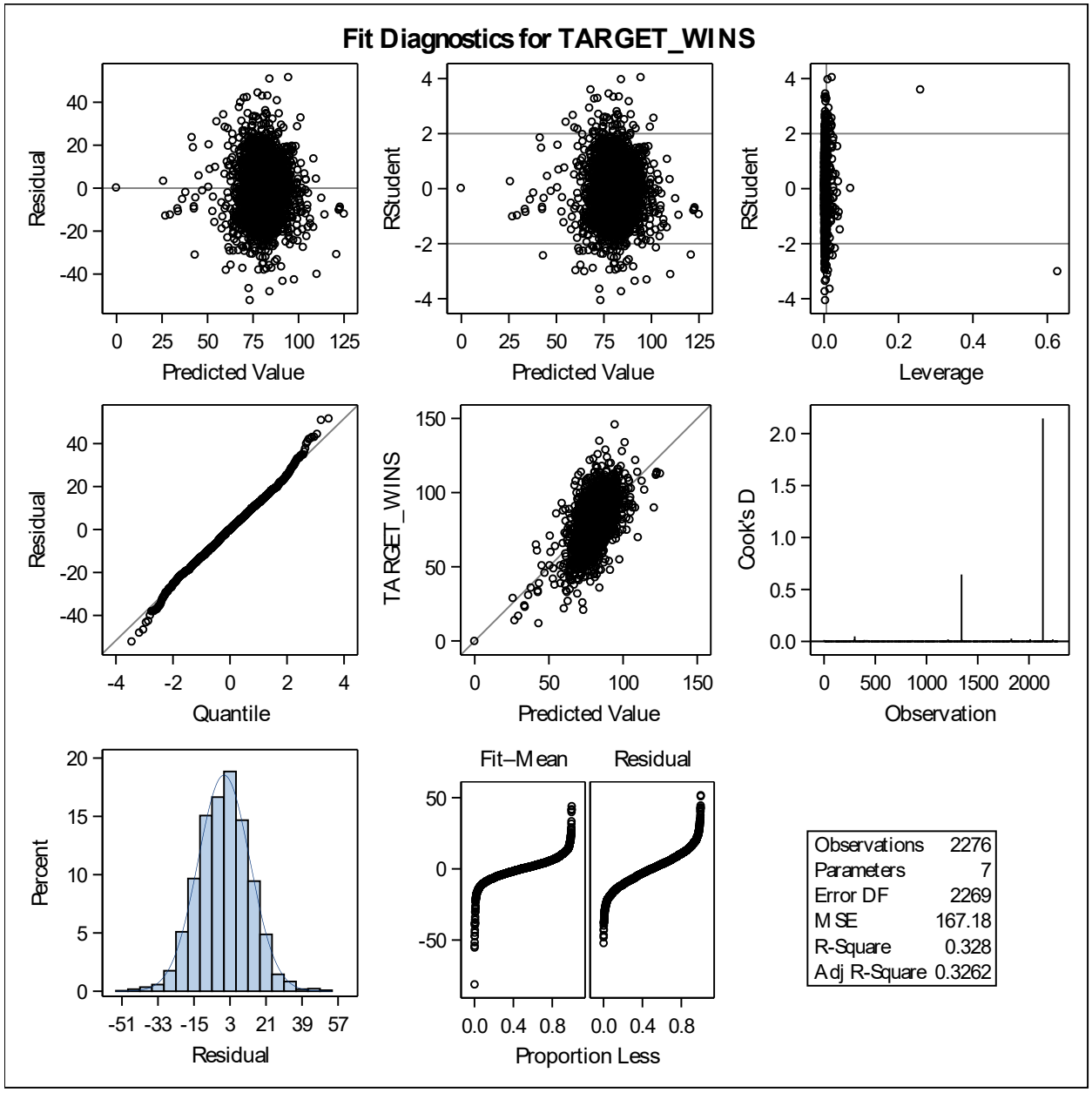


Figure 10

SELECT THE MODEL:

As mentioned earlier, the main decision criterion is the Adjusted R-Squared. A higher Adjusted R-Squared is indicative of better performance. MSE is also used as a secondary criterion which measures the difference between actual and predicted values. A lower MSE is better. Aside from these criteria, the goal is also to have a highly interpretable model that makes sense. Model 5 fits the bill.

When looking at Adjusted R-Squared, Model 1 is 0.3148, Model 2 is 0.3154, Model 3 is 0.3117, Model 4 is 0.3373, and Model 5 is 0.3263. The respective MSE's are 170.03, 169.88, 170.78, 164.44, and 167.18. Based on these criteria, Model 4 is clearly the best performer. However, Model 4 also has variables that are counterintuitive and are not statistically significant which is why Model 5 is the best overall model. It is easy to interpret, the variables make intuitive sense, and the performance is relatively good.

CONCLUSION:

Five models were generated from baseball team data from 1871 to 2006 to predict number of wins. Prior to generating the models, the baseball data was analyzed to better understand the relationship between variables. The chosen model was Model 5 which was created using decision trees and stepwise variable selection. Model 5 exhibited good performance with a relatively high Adjusted R-Squared value compared to other models. Although it did not have the best overall performance, the tradeoff is a more intuitive and simple model.

BINGO BONUS:

Attempted points: 50

I created a PROC GLM and PROC GENMOD step in my SAS code submitted to canvas and have a short writeup of the differences below with SAS output (20 points). I also created decision tree in R with code and output below (20 points). Lastly, I used SAS macros in my SAS code (10 points).

PROC GLM

1) Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	185765.4989	18576.5499	111.10	<.0001
Error	2265	378730.9510	167.2101		
Corrected Total	2275	564496.4499			

R-Square	Coeff Var	Root MSE	TARGET_WINS Mean
0.329082	16.00549	12.93098	80.79086

Source	DF	Type I SS	Mean Square	F Value	Pr > F
TEAM_BATTING_H	1	85318.09812	85318.09812	510.24	<.0001
TEAM_BATTING_HR	1	18026.73476	18026.73476	107.81	<.0001
TEAM_BATTING_BB	1	22048.82591	22048.82591	131.86	<.0001
TEAM_PITCHING_H	1	10111.30758	10111.30758	60.47	<.0001
TEAM_PITCHING_HR	1	298.39870	298.39870	1.78	0.1817
TEAM_PITCHING_BB	1	2566.62871	2566.62871	15.35	<.0001
TEAM_FIELDING_E	1	4859.06810	4859.06810	29.06	<.0001
IMP_TEAM_BASERUN_SB	1	31982.72182	31982.72182	191.27	<.0001

Source	DF	Type I SS	Mean Square	F Value	Pr > F
IMP_TEAM_PITCHING_SO	1	48.91936	48.91936	0.29	0.5886
IMP_TEAM_BATTING_SO	1	10504.79586	10504.79586	62.82	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
TEAM_BATTING_H	1	43483.99220	43483.99220	260.06	<.0001
TEAM_BATTING_HR	1	613.90041	613.90041	3.67	0.0555
TEAM_BATTING_BB	1	519.72723	519.72723	3.11	0.0780
TEAM_PITCHING_H	1	18.92384	18.92384	0.11	0.7366
TEAM_PITCHING_HR	1	148.96003	148.96003	0.89	0.3453
TEAM_PITCHING_BB	1	236.59372	236.59372	1.41	0.2344
TEAM_FIELDING_E	1	24507.15777	24507.15777	146.57	<.0001
IMP_TEAM_BASERUN_SB	1	39577.83687	39577.83687	236.70	<.0001
IMP_TEAM_PITCHING_SO	1	1507.83588	1507.83588	9.02	0.0027
IMP_TEAM_BATTING_SO	1	10504.79586	10504.79586	62.82	<.0001

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	22.28275330	4.47190197	4.98	<.0001
TEAM_BATTING_H	0.04031234	0.00249980	16.13	<.0001
TEAM_BATTING_HR	0.05089593	0.02656228	1.92	0.0555
TEAM_BATTING_BB	0.01007128	0.00571253	1.76	0.0780
TEAM_PITCHING_H	0.00012592	0.00037431	0.34	0.7366

Parameter	Estimate	Standard Error	t Value	Pr > |t|
TEAM_PITCHING_HR	0.02252159	0.02386138	0.94	0.3453
TEAM_PITCHING_BB	-0.00485945	0.00408524	-1.19	0.2344
TEAM_FIELDING_E	-0.03072033	0.00253753	-12.11	<.0001
IMP_TEAM_BASERUN_SB	0.06235227	0.00405282	15.38	<.0001
IMP_TEAM_PITCHING_SO	0.00268782	0.00089507	3.00	0.0027
IMP_TEAM_BATTING_SO	-0.01895404	0.00239133	-7.93	<.0001

The results are different compared to the champion model (Model 5). The R-Square value slightly higher at 0.329 while Model 5 had an R-Squared value of 0.328. The PROC GLM also produced more variables, 10 compared to 6 in Model 5. I was not able to use stepwise selection in the PROC GLM step which is why there are more variables and I was not able to specify adjusted R-Squared value to be output. All variables appear to make intuitive sense. According to the PROC GLM, TEAM_PITCHING_H is considerably insignificant. This variable would be removed.

PROC GENMOD

Model Information	
Data Set	WORK.TEMPFILE
Distribution	Normal
Link Function	Identity
Dependent Variable	TARGET_WINS

Criteria For Assessing Goodness Of Fit			
Criterion	**DF**	**Value**	**Value/DF**
Deviance	2265	378730.9510	167.2101
Scaled Deviance	2265	2276.0000	1.0049
Pearson Chi-Square	2265	378730.9510	167.2101
Scaled Pearson X2	2265	2276.0000	1.0049
Log Likelihood		-9049.6987	
Full Log Likelihood		-9049.6987	
AIC (smaller is better)		18123.3975	
AICC (smaller is better)		18123.5354	
BIC (smaller is better)		18192.1596	

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	**DF**	**Estimate**	**Standard Error**	**Wald 95% Confidence Limits**		**Wald Chi-Square**	**Pr > ChiSq**
Intercept	1	22.2828	4.4611	13.5392	31.0263	24.95	<.0001
TEAM_BATTING_H	1	0.0403	0.0025	0.0354	0.0452	261.32	<.0001
TEAM_BATTING_HR	1	0.0509	0.0265	-0.0010	0.1028	3.69	0.0548
TEAM_BATTING_BB	1	0.0101	0.0057	-0.0011	0.0212	3.12	0.0772
TEAM_PITCHING_H	1	0.0001	0.0004	-0.0006	0.0009	0.11	0.7359
TEAM_PITCHING_HR	1	0.0225	0.0238	-0.0241	0.0692	0.90	0.3441

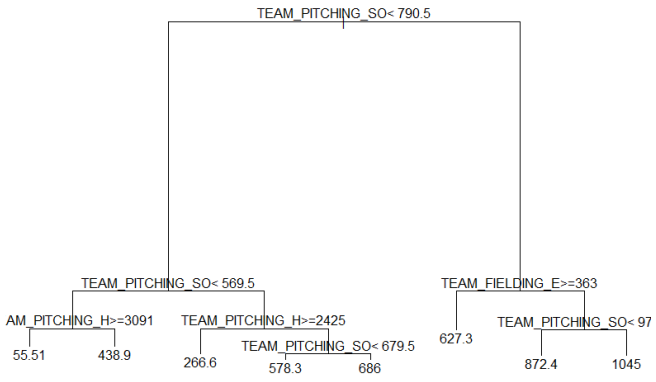
Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
TEAM_PITCHING_BB	1	-0.0049	0.0041	-0.0128	0.0031	1.42	0.2331
TEAM_FIELDING_E	1	-0.0307	0.0025	-0.0357	-0.0258	147.28	<.0001
IMP_TEAM_BASERUN_SB	1	0.0624	0.0040	0.0544	0.0703	237.84	<.0001
IMP_TEAM_PITCHING_SO	1	0.0027	0.0009	0.0009	0.0044	9.06	0.0026
IMP_TEAM_BATTING_SO	1	-0.0190	0.0024	-0.0236	-0.0143	63.13	<.0001
Scale	1	12.8997	0.1912	12.5303	13.2799		

The PROC GENMOD procedure contains different output compared to PROC GLM AND PROC REG. It seems to use maximum likelihood estimation. The PROC GENMOD uses several metrics to analyze parameter estimates. The resulting equation is very similar to PROC GLM. Again, the results here are different than OLS in Model 5 due to ability to specify stepwise selection.

Decision Tree in R (CODE AND OUTPUT)

```
### Moneyball Assignment ###

# Using decision trees to impute missing values #


# load training dataset and libraries

library(rpart)

setwd("./PREDICT 411/Moneyball")

mb <- as.data.frame(read.csv("moneyball.csv"))

#test <- as.data.frame(read.csv("moneyball_test.csv"))


# examine the dataset

str(mb)

summary(mb)

cor(mb)
```

```
# drop unused columns prior to decision tree

mb <- mb[,-c(1,2,11)]

names(mb)


# create a decision tree for each variable with missing values. cross validate to see if we can prune the tree to make simpler

# TEAM_BATTING_SO, TEAM_BASERUN_CS, TEAM_BASERUN_SB, TEAM_PITCHING_SO, TEAM_FIELDING_DP


batting_so_tree <- rpart(TEAM_BATTING_SO~.,data=mb)

plot(batting_so_tree)

text(batting_so_tree)

printcp(batting_so_tree)

plotcp(batting_so_tree) # keep all 8 branches
```

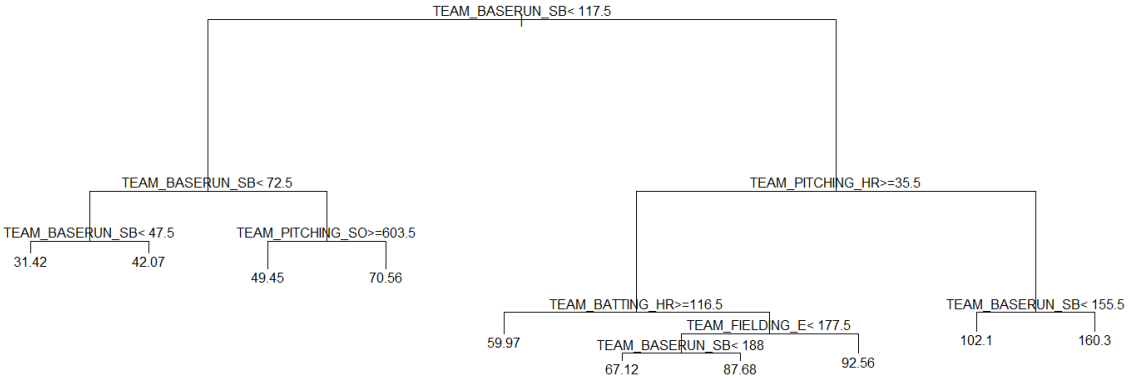



```
baserun_cs <- rpart(TEAM_BASERUN_CS~.,data=mb)

plot(baserun_cs)
```

text(baserun_cs)

printcp(baserun_cs)

plotcp(baserun_cs) # keep all 10 branches



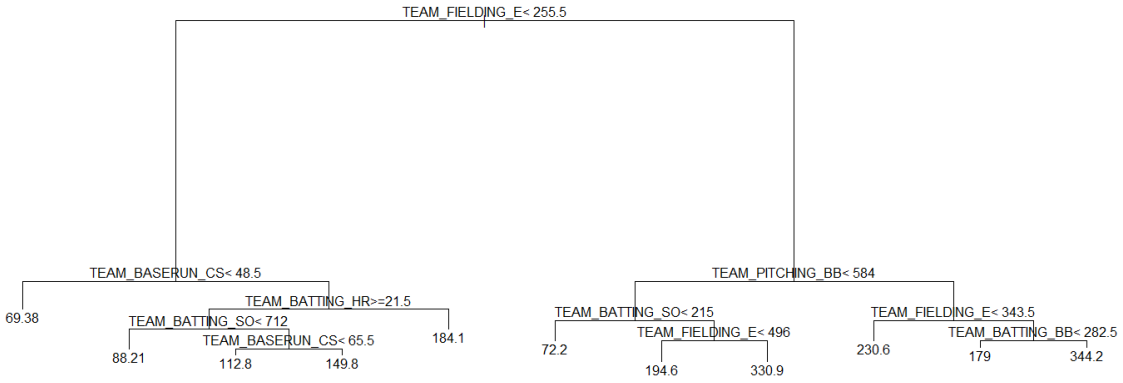
baserun_sb <- rpart(TEAM_BASERUN_SB~.,data=mb)

plot(baserun_sb)

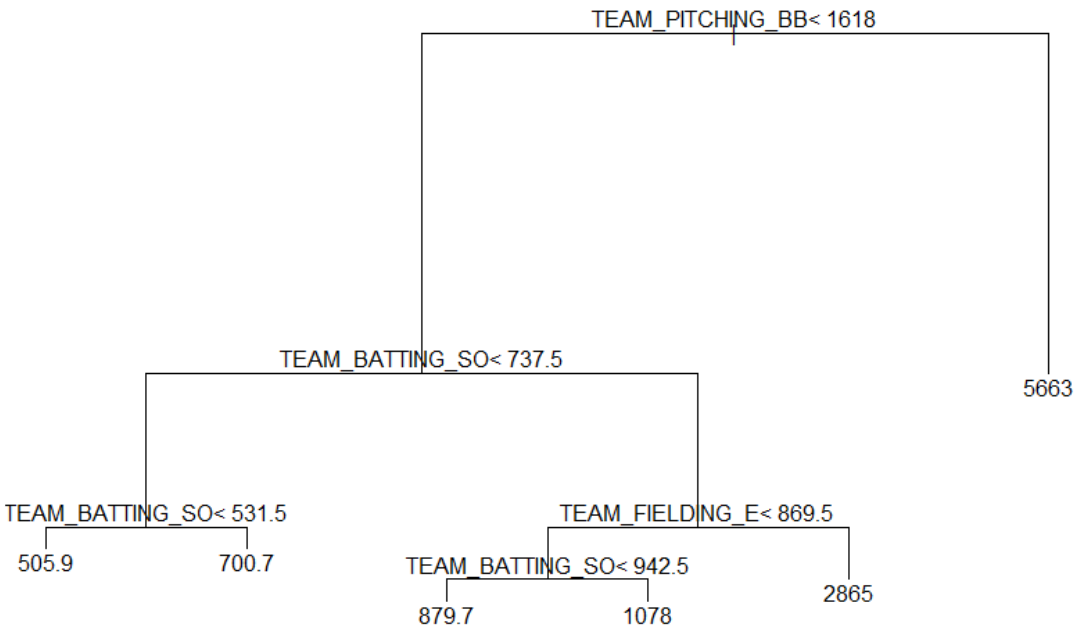
text(baserun_sb)

printcp(baserun_sb)

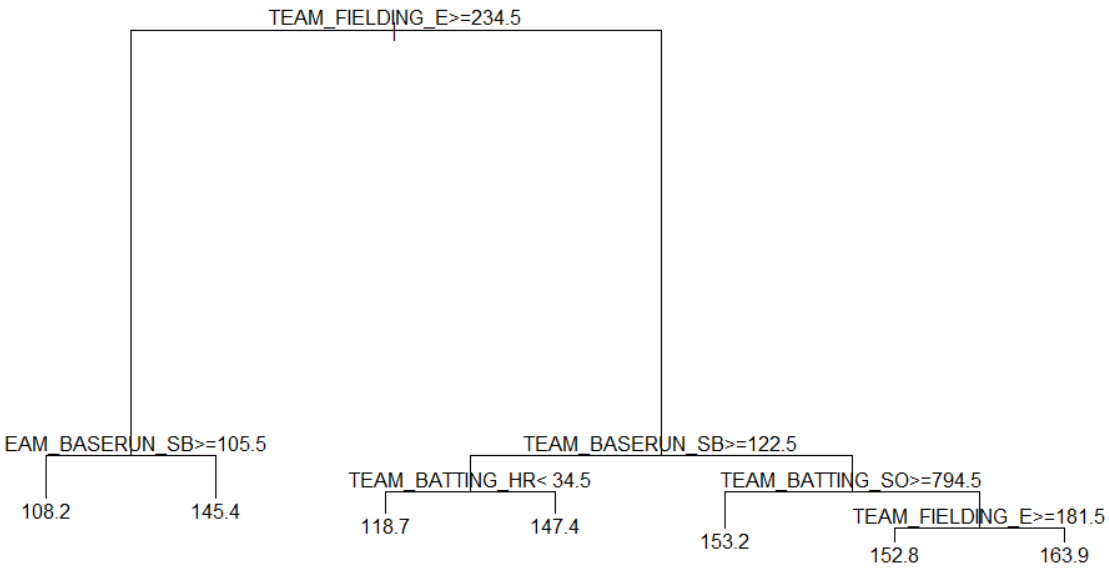
plotcp(baserun_sb) # keep all 10 branches (maybe 8)




```
pitching_so <- rpart(TEAM_PITCHING_SO~.,data=mb)

plot(pitching_so)

text(pitching_so)

printcp(pitching_so)

plotcp(pitching_so) # keep all 6 branches (maybe 5)
```




```
fielding_dp <- rpart(TEAM_FIELDING_DP~.,data=mb)

plot(fielding_dp)

text(fielding_dp)

printcp(fielding_dp)

plotcp(fielding_dp) # keep all 6 branches
```



SAS Macros

See SAS code file.