DATA621 Homework 1

John Ferrara

2025-03-01

Overview

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season.

Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. You can only use the variables given to you (or variables that you derive from the variables provided). Below is a short description of the variables of interest in the data set:

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

Variable Description Table

Deliverables:

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.

- Assigned predictions (the number of wins for the team) for the evaluation data set.
- Include your R statistical programming code in an Appendix.

Write Up INSTRUCTIONS:

1) DATA EXPLORATION

Describe the size and the variables in the moneyball training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas. a. Mean / Standard Deviation / Median b. Bar Chart or Box Plot of the data c. Is the data correlated to the target variable (or to other variables?) d. Are any of the variables missing and need to be imputed "fixed"?

2) DATA PREPARATION

Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations. a. Fix missing values (maybe with a Mean or Median value) b. Create flags to suggest if a variable was missing c. Transform data by putting it into buckets d. Mathematical transforms such as log or square root (or use Box-Cox) e. Combine variables (such as ratios or adding or multiplying) to create new variables

3) BUILD MODELS

Using the training data set, build at least three different multiple linear regression models, using different variables (or the same variables with different transformations). Since we have not yet covered automated variable selection methods, you should select the variables manually (unless you previously learned Forward or Stepwise selection, etc.). Since you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done. Discuss the coefficients in the models, do they make sense? For example, if a team hits a lot of Home Runs, it would be reasonably expected that such a team would win more games. However, if the coefficient is negative (suggesting that the team would lose more games), then that needs to be discussed. Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

4) SELECT MODELS

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 * (d) residual plots

Make predictions using the evaluation data set.

—- WRITE UP START —-

DATA EXPLORATION:

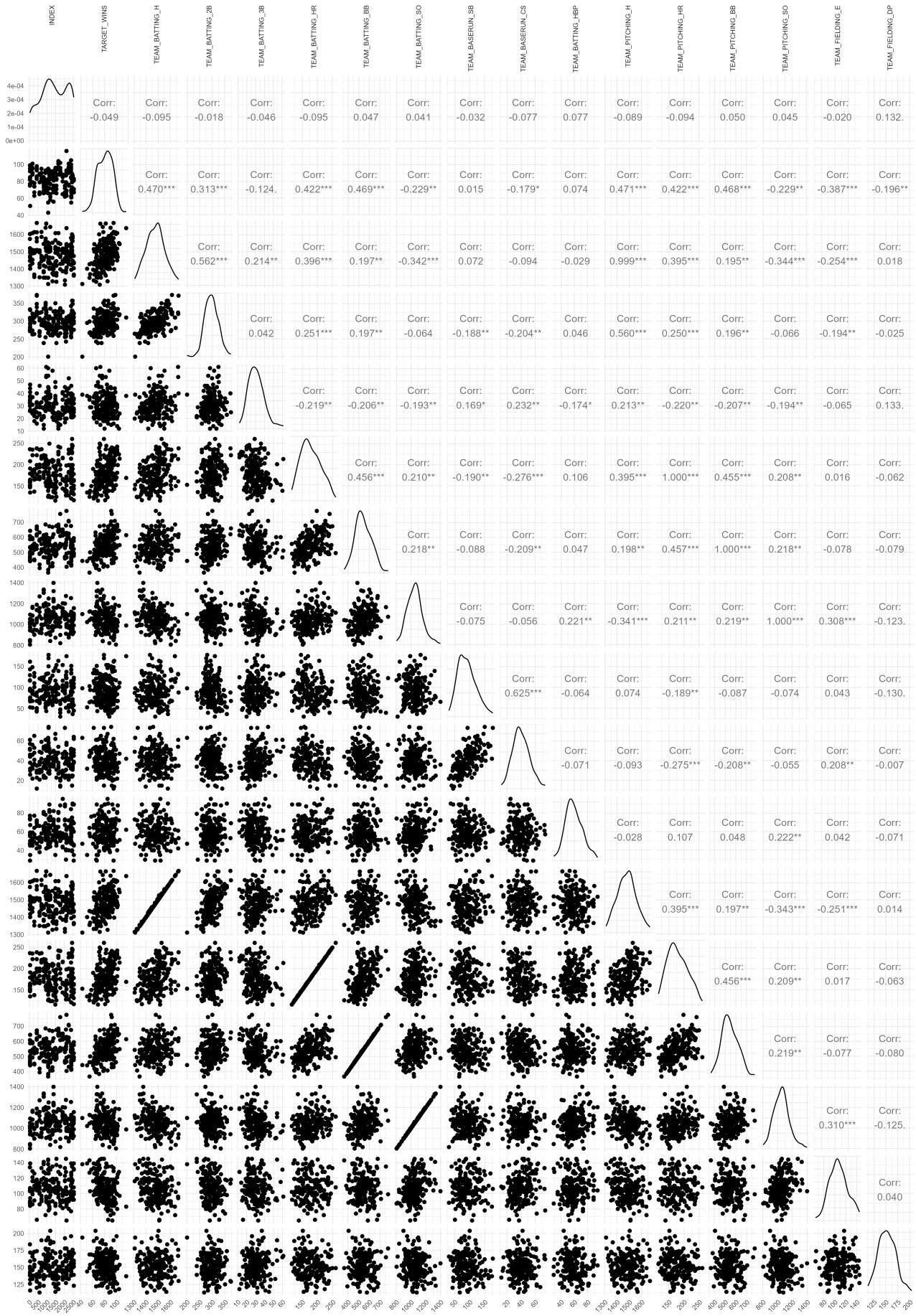
The training data set received had 2276 rows of data, with no duplicate rows. There were many variables within the initial training data set, some of which were redundant or aggregate versions of other variables. After inspection the dependent variable that we want to inspect, and subsequently predict is TARGET_WINS. WIth the exception of the INDEX column, the rest are the independent X variables. Below is a break down of all of the non-index columns and their respective Null value counts, along with the range of their values. Note: All of the columns were continuous integers.

Overview of Non-Index Columns

Column Name	Data Type	Nulls	Range	Notes
TARGET_WINS	<int>	No Nulls	0 - 146	DEPENDENT_VAR
TEAM_BATTING_H	<int>	No Nulls	891 - 2,554	
TEAM_BATTING_2B	<int>	No Nulls	69 - 458	
TEAM_BATTING_3B	<int>	No Nulls	0 - 223	
TEAM_BATTING_HR	<int>	No Nulls	0 - 264	
TEAM_BATTING_BB	<int>	No Nulls	0 - 878	
TEAM_BATTING_SO	<int>	102 Nulls	0 - 1,399	
TEAM_BASERUN_SB	<int>	131 Nulls	0 - 697	
TEAM_BASERUN_CS	<int>	772 Nulls	0 - 201	
TEAM_BATTING_HBP	<int>	2085 Nulls	29 - 95	
TEAM_PITCHING_H	<int>	No Nulls	1,137 - 30,132	
TEAM_PITCHING_HR	<int>	No Nulls	0 - 343	
TEAM_PITCHING_BB	<int>	No Nulls	0 - 3,645	
TEAM_PITCHING_SO	<int>	102 Nulls	0 - 19,278	
TEAM_FIELDING_E	<int>	No Nulls	65 - 1,898	
TEAM_FIELDING_DP	<int>	286 Nulls	52 - 228	

Of these columns there are several that have null values in need of resolution. After confirming that null values were not just “zeros in disguise”, the scale of the presence of nulls in the data was examined, only 8 percent of rows from the original 2,276 rows have no nulls. Leading us to conclude strategies for accommodating these null columns will be necessary. Ultimately, several different techniques were leveraged. In addition to these techniques there were one or two other column that were parsed out of larger “aggregate” type columns. Lastly, for a brief overview on all of the initial variable columns and their relationship to one another for those rows without nulls please see the chart below:

GGPairs Table For Training Data



GGPairs Table For Training Data

The main takeaways from this glimpse of the raw data with nulls removed were:

- Several Variables have noticeable Positive relationships and noticable negative relationships with the TARGET_WINS Variable.
- TEAM_BATTING_H and TEAM_PITCHING_H have a direct linear relationship, not independent
- TEAM_BATTING_SO and TEAM_PITCHING_SO have a direct linear relationship, not independent.
- TEAM_BATTING_BB and TEAM_PITCHING_BB have a direct linear relationship, not independent.
- Positive relationship between TEAM_BASERUN_SB and TEAM_BASERUN_CS, may want to merge these to one var.

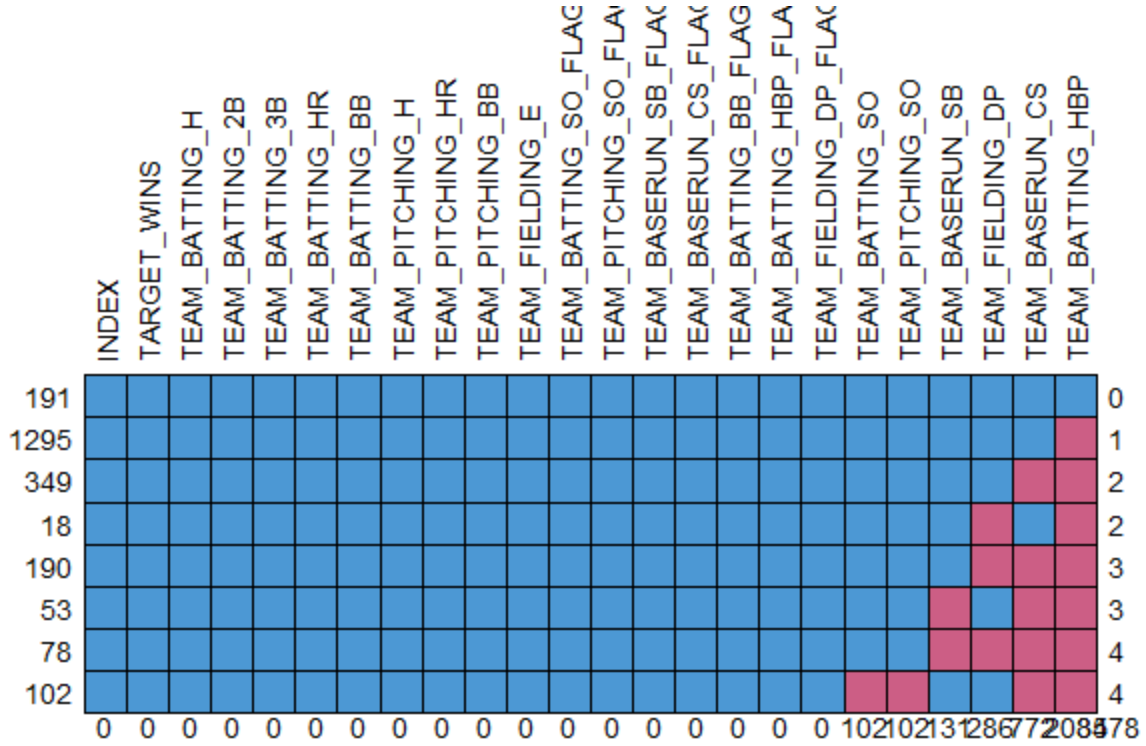
After this initial evaluation, the data cleaning and processing was carried out.

DATA PROCESSING

There were multiple steps taken to process the training data and prepare it for modeling. Firstly, as mentioned, the columns that contained null values needed to be addressed. Please see the image below for a coverage map for the presence of nulls in the training data. Each column with nulls was examined and addressed with a specific solution. These solutions were as follows:

- Before imputation several flagging columns were generated in order to keep track of which row values were imputed if needed.
- There were enough non-null values for TEAM_BATTING_SO and TEAM_PITCHING_SO, so imputation using the MICE packed with Predictive Mean Modeling (PMM) methodology was used. Enough data to impute TEAM_BATTING_SO and TEAM_PITCHING_SO. Need to impute.
- Additionally, before imputation of the TEAM_PITCHING_SO column there were two rows that contained extreme outliers. These two rows were removed from the data before imputation for any data.
- Similarly, the TEAM_FIELDING_DP column has 286 Nulls, which was roughly 14% null values. The remaining 1,990 values were used to impute values for this nulls using the same MICE PMM methodology.
- The other columns were handled without the use of imputation. This as in part due to the large number of null values in the data compared with the non null values for select columns. The actions taken were:
    - The columns TEAM_BASERUN_SB and TEAM_BASERUN_CS were combined into TEAM_BASERUN_STEAL_ATTEMPTS. This yeilded no nulls for this new aggregate column.
    - The columns TEAM_BATTING_HBP was combined with TEAM_BATTING_BB (no null values) to generate TEAM_BATTING_WALK_TOTAL. This yielded no null values.

Null Values Visualized with MICE Package



Null Values Visualized with MICE Package

Beyond imputation and accommodating nulls there were other aspects of the data that were edited, this varied by model and the concept behind them. These tactics, because they were specific for the models are discussed in the next section.

MODEL BUILDING

There were multiple modeling concepts conceived and tested before ulimately landing on one to use for TARGET_WINS prediction. The models that were devised, and their respcective data alterations are as follows:

1. TARGET_WINS VS VARIOUS TYPES OF BASE HITS (1, 2, 3, and HOME RUNS)

This model sought to examine the types of batting gains with respect to base number in order to see if there was any significance in variance explaination for TARGET_WINS of a team. It parsed out TEAM_BATTING_1B from the larger aggregate TEAM_BATTING_H, and modeled the relevant columns.The adjusted R^2 here was around 20% of the variance being explained by these X variables with a decent F-score of 145. P-values show statistical significance. The coefficents here were highest for 3 base hits(0.15), and homeruns (0.11). THis was interesting because one would think Home Runs would have the highest coefficient for impact on wins.

CUSTOM DATA PROCESSING

- Parsed out TEAM_BATTING_1B from TEAM_BATTING_H. This was done by subtracting the sum of TEAM_BATTING_1B, TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BATTING_HR from TEAM_BATTING_H.
- The TEAM_BATTING_H was dropped from the model because of its aggregate nature.

INITIAL MODEL PERFORMANCE

MODEL 1 SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +
    TEAM_BATTING_2B + TEAM_BATTING_HR, data = model_1_df)

Residuals:
    Min      1Q  Median      3Q     Max
-71.390  -8.854   0.603   9.632  49.361

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     18.031397   3.188915   5.654 1.76e-08 ***
TEAM_BATTING_1B  0.032043   0.003092  10.364  < 2e-16 ***
TEAM_BATTING_3B  0.157431   0.015193  10.362  < 2e-16 ***
TEAM_BATTING_2B  0.034017   0.007679   4.430 9.89e-06 ***
TEAM_BATTING_HR  0.114952   0.007682  14.964  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.04 on 2269 degrees of freedom
Multiple R-squared:  0.2039,    Adjusted R-squared:  0.2025
F-statistic: 145.3 on 4 and 2269 DF,  p-value: < 2.2e-16
```

MODEL 1 SUMMARY

2. TARGET_WINS VS DEFENSIVE AND OFFENSIVE INDICES (MODEL ULTIMATELY SELECTED)

This model sought to examine the influences of defense and offensive actions on the chances of winning the game. These indices were used see if there was any significance in variance explanation for TARGET_WINS of a team. This model was subsequently selected to fine tune and improve because it was the best performing model. The Adjusted R^2 for this model was ~21.3 percent of the variance explained by these two aggregate indices.P-values show statistical significance. The coefficients are pretty low though, with the defense indec being higher than the offense index.

CUSTOM DATA PROCESSING

- DEFENSE_INDEX was created by adding up the sum of TEAM_PITCHING_SO + TEAM_FIELDING_DP for a row and subtracting the TEAM_FIELDING_E column.
- OFFENSE_INDEX was created by adding up TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BASERUN_SB for a row.

INITIAL MODEL PERFORMANCE

MODEL 2 SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX, data = model_2_dfa)

Residuals:
    Min      1Q  Median      3Q     Max
-62.424  -8.976   0.481   9.035  57.093

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0615756  3.0863725   1.640  0.10115
OFFENSE_INDEX 0.0353272  0.0014213  24.855  < 2e-16 ***
DEFENSE_INDEX 0.0025885  0.0007198   3.596  0.00033 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.95 on 2271 degrees of freedom
Multiple R-squared:  0.214,     Adjusted R-squared:  0.2133
F-statistic: 309.2 on 2 and 2271 DF,  p-value: < 2.2e-16
```

MODEL 2 SUMMARY

SUBMODELS WITH NUANCE

Within this model, the columns that fed each index were also used in two separate models in order to get a sense of how each influenced the model. Within the main model the granualar models I ran to see the nuanced break down for each column were interesting. The Defensive Granular model was not relevant, as the coefficients were super small. although it should be noted they were negative. However, the TEAM_FIELDING_DP was not statistically significant. Pivoting to the Offensive Granualr Model, this one has all significant p-scores as well as a decent adjusted R^2 of 0.225. The F-score was higher at 222. The coefficients were still small, with TEAM_BATTING_H having the highest by just a bit.

MODEL 2 DEFENSIVE COLUMNS GRANULAR SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_FIELDING_DP,
    data = model_2_df_granular_defense)

Residuals:
    Min      1Q  Median      3Q     Max
-85.908  -9.571   1.057  10.775  59.718

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      86.970356   1.718626  50.605  < 2e-16 ***
TEAM_PITCHING_SO -0.006285   0.001162  -5.410 6.97e-08 ***
TEAM_FIELDING_DP -0.008299   0.011101  -0.748    0.455
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.62 on 2271 degrees of freedom
Multiple R-squared:  0.01382,   Adjusted R-squared:  0.01295
F-statistic: 15.92 on 2 and 2271 DF,  p-value: 1.366e-07
```

MODEL 2 OFFENSIVE COLUMNS GRANULAR SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + 
    TEAM_BASERUN_SB, data = model_2_df_granualar_offense)

Residuals:
    Min      1Q  Median      3Q     Max 
-62.823  -8.763   0.490   9.235  49.663 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -2.318103   3.299134  -0.703    0.482    
TEAM_BATTING_H  0.043789   0.002029  21.581  < 2e-16 ***
TEAM_BATTING_BB 0.033815   0.002379  14.214  < 2e-16 ***
TEAM_BASERUN_SB 0.014734   0.003381   4.358 1.37e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.84 on 2270 degrees of freedom
Multiple R-squared:  0.2268,    Adjusted R-squared:  0.2258 
F-statistic:   222 on 3 and 2270 DF,  p-value: < 2.2e-16
```

3. TARGET_WINS VS HIGH RISK AND LOW RISK ACTION INDICES

This model sought to examine the influences of high risk and low risk actions on the chances of winning the game. High risk actions were multiple base hits, excluding home runs, while low risk behavior included single base gain hits, walking base gains, and batting home runs. This model had statistically significant p=values, and an adjusted R^2 of 0.20. The F-score is relatively higher compared to the other models too. While both coefficients are low, the Low Risk actions had a slightly higher one at 0.039, while the High Risk actions had 0.018. This may make sense depending on how you look at the game, and high risk may be higher reward, but more lower risk activites may help win the game without as much risk.

CUSTOM DATA PROCESSING

- HIGH RISK INDEX was created by taking the sum of “high risk” types of playing activity. It summed up TEAM_BASERUN_STEAL_ATTEMPTS, TEAM_BATTING_2B and TEAM_BATTING_3B.
- LOW RISK INDEX was created by taking the sum of “low risk” types of playing activities. It summed up TEAM_BATTING_1B, TEAM_BATTING_WALK_TOTAL, and TEAM_BATTING_HR.

INITIAL MODEL PERFORMANCE

MODEL 3 SUMMARY

```
Call:
lm(formula = TARGET_WINS + 1 ~ HIGH_RISK + LOW_RISK, data = model_3_dfa)

Residuals:
    Min      1Q  Median      3Q     Max 
-57.078  -9.081   0.197   9.110  56.827 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.664325   3.207541   2.389    0.017 *  
HIGH_RISK   0.018733   0.002900   6.459 1.29e-10 ***
LOW_RISK    0.039120   0.002029  19.277  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.06 on 2271 degrees of freedom
Multiple R-squared:  0.201,     Adjusted R-squared:  0.2002 
F-statistic: 285.6 on 2 and 2271 DF,  p-value: < 2.2e-16
```

SUBMODELS WITH NUANCE

Within this model, the columns that fed each index were also used in two separate models in order to get a sense of how each influenced the model. Both of the granular models yielded worse R^2 values than the initial Model 3 performance. All the p-values in these 2 models were statistically significant, but the coefficents are all still pretty low. The highest coefficient in both of these models was that for the TEAM_BATTING_2B (0.10) in the high risk activities model.

MODEL 3 HIGH RISK GRANULAR SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + 
    TEAM_BASERUN_STEAL_ATTEMPTS, data = model_3_df_high_risk_granular)

Residuals:
    Min      1Q  Median      3Q     Max 
-71.784  -9.394   0.655   9.728  54.924 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              47.976139   1.857844  25.824  < 2e-16 ***
TEAM_BATTING_2B           0.105327   0.006649  15.842  < 2e-16 ***
TEAM_BATTING_3B           0.082740   0.011532   7.175 9.74e-13 ***
TEAM_BASERUN_STEAL_ATTEMPTS  0.018618   0.003500   5.320 1.14e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.72 on 2270 degrees of freedom
Multiple R-squared:  0.1249,    Adjusted R-squared:  0.1237 
F-statistic:   108 on 3 and 2270 DF,  p-value: < 2.2e-16
```

MODEL 3 LOW RISK GRANULAR SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + 
    TEAM_BATTING_HR, data = model_3_df_low_risk_granular)

Residuals:
    Min      1Q  Median      3Q     Max 
-64.285  -8.924   0.259   9.346  52.517 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)           -0.383255   3.557273  -0.108    0.914    
TEAM_BATTING_1B        0.054282   0.002683  20.229   <2e-16 ***
TEAM_BATTING_WALK_TOTAL  0.031993   0.002860  11.187   <2e-16 ***
TEAM_BATTING_HR        0.067185   0.006324  10.624   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.08 on 2270 degrees of freedom
Multiple R-squared:  0.1996,    Adjusted R-squared:  0.1986 
F-statistic: 188.7 on 3 and 2270 DF,  p-value: < 2.2e-16
```

4. TARGET_WINS VS DEFENSIVE PITCHING VS DEFENSIVE FIELD ACTION INDICES

This model sought to look at the differences in defensive type actions on the TARGET_WINS. Defensive Field activities such as double plays and errors made when a team is in the field. Defensive Pitch activities took a look at the Pitcher’s strike outs, as well as the pitcher’s walks and hits allowed. The adjusted R^2 for this main model was super low at ~2% as was the F-score of 32.

CUSTOM DATA PROCESSING

- FIELD WORK INDEX was created by taking the TEAM_FIELDING_DP and subtracting the TEAM_FIELDING_E column.
- PITCH WORK INDEX was created by taking the TEAM_PITCHING_SO and subtracting the sum of TEAM_PITCHING_BB and TEAM_PITCHING_H columns.

INITIAL MODEL PERFORMANCE

MODEL 4 SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX,
    data = model_4_dfa)

Residuals:
    Min      1Q  Median      3Q     Max
-76.219  -9.996   0.544  10.294  72.538

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      81.0242086  0.4966042 163.156  < 2e-16 ***
FIELD_WORK_INDEX  0.0128692  0.0017484   7.361 2.55e-13 ***
PITCH_WORK_INDEX -0.0007428  0.0003085  -2.408   0.0161 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.52 on 2271 degrees of freedom
Multiple R-squared:  0.0275,    Adjusted R-squared:  0.02664
F-statistic: 32.11 on 2 and 2271 DF,  p-value: 1.778e-14
```

SUBMODELS WITH NUANCE

Within this model, the columns that fed each index were also used in two separate models in order to get a sense of how each influenced the model. The granular models performed similarly to the main MODEL 4, so it isnt worth digging in further.

MODEL 4 PITCH WORK GRANULAR SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_PITCHING_BB +
    TEAM_PITCHING_H, data = model_4_df_defensive_pitch_granular)

Residuals:
    Min      1Q  Median      3Q     Max
-59.621  -9.412   0.789  10.052  72.966

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      79.0080593  1.4307605  55.221  < 2e-16 ***
TEAM_PITCHING_SO -0.0093176  0.0011475  -8.120 7.56e-16 ***
TEAM_PITCHING_BB  0.0228060  0.0022950   9.937  < 2e-16 ***
TEAM_PITCHING_H  -0.0018958  0.0002471  -7.673 2.48e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.2 on 2270 degrees of freedom
Multiple R-squared:  0.06762,   Adjusted R-squared:  0.06639
F-statistic: 54.88 on 3 and 2270 DF,  p-value: < 2.2e-16
```

MODEL 4 FIELD WORK GRANULAR SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ TEAM_FIELDING_DP + TEAM_FIELDING_E,
    data = model_4_df_defensive_field_granular)

Residuals:
    Min      1Q  Median      3Q     Max
-60.906  -9.951   0.547   9.827  73.382

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     101.004634   2.153992  46.892  < 2e-16 ***
TEAM_FIELDING_DP  -0.108006   0.013182  -8.193 4.19e-16 ***
TEAM_FIELDING_E   -0.020423   0.001734 -11.779  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.27 on 2271 degrees of freedom
Multiple R-squared:  0.05863,   Adjusted R-squared:  0.0578
F-statistic: 70.72 on 2 and 2271 DF,  p-value: < 2.2e-16
```

MODEL SELECTION AND FURTHER MODIFICATION

Of all the models that were initially thought of and executed Model 2 was the best performing, so it was ultimately chosen for refinement and improvement.

Alterations and Improvements to MODEL 2

After selecting MODEL 2 due to its relatively better performance, the negative valued in the Defense index needed to be corrected before attempting to use Box-Cox to identify a proper lambda value to help fit the dependent variable. This was done by shifting the Defense index to be calulated by dividing the sum of TEAM_PITCHING_SO and TEAM_FIELDING_DP by the TEAM_FIELDING_E column. This yeilded positive values for this index. Additionally, for both the Defensive and Offensive indicies multiple transformations were applied with the results of each examined in order to determine the best transformation. These results can be seen in the table below. Lastly, because there were zeros in the TARGET_WINS column, this column was altered to have 1 added to it.

TRANSFORMATION COMPARISONS

Transformation	R^2 (%)	F-score
No Change (X Vars)	21.7	317
Log Transform	**22.75**	**335**
Square Root (SqRt)	22.43	329
Squared	21.17	317
Cubed	21.77	317

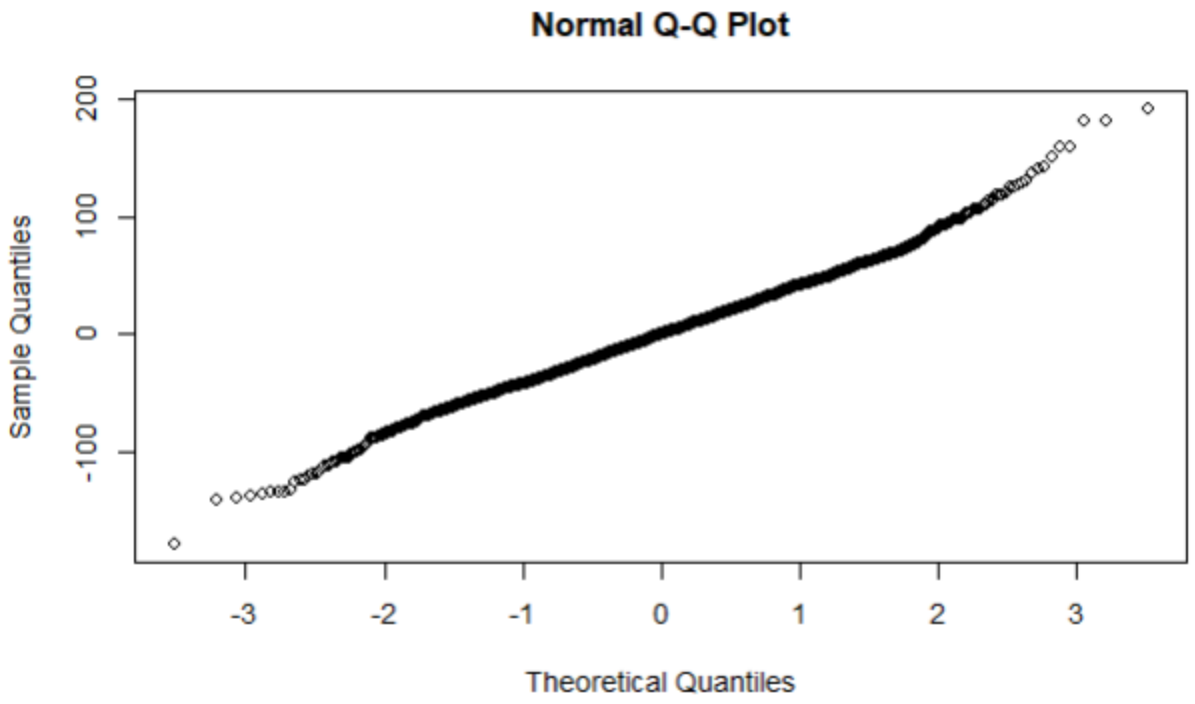
These transformations were applied to the data, and Model 2 was run again with updated results. You can see below that the transformed model has normally distributed residuals on the histgram, and mostly normal QQ plot as well. However, there is some slight deviation at the ends on the QQ line plot.

TRANSFORMED MODEL 2 MODEL SUMMARY

```
Call:
lm(formula = TARGET_WINS + 1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX),
    data = model_2_dfa_transform)

Residuals:
     Min       1Q   Median       3Q      Max
-186.614  -30.177    0.178   28.870  211.998

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1624.048     73.178 -22.193   <2e-16 ***
log(DEFENSE_INDEX)    3.726      1.131   3.294    0.001 **
log(OFFENSE_INDEX)  239.523      9.573  25.022   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 45.34 on 2271 degrees of freedom
Multiple R-squared:  0.219,     Adjusted R-squared:  0.2183
F-statistic: 318.3 on 2 and 2271 DF,  p-value: < 2.2e-16
```

TRANSFORMED MODEL 2 HISTOGRAM PLOT

TRANSFORMED MODEL 2 QQ PLOT

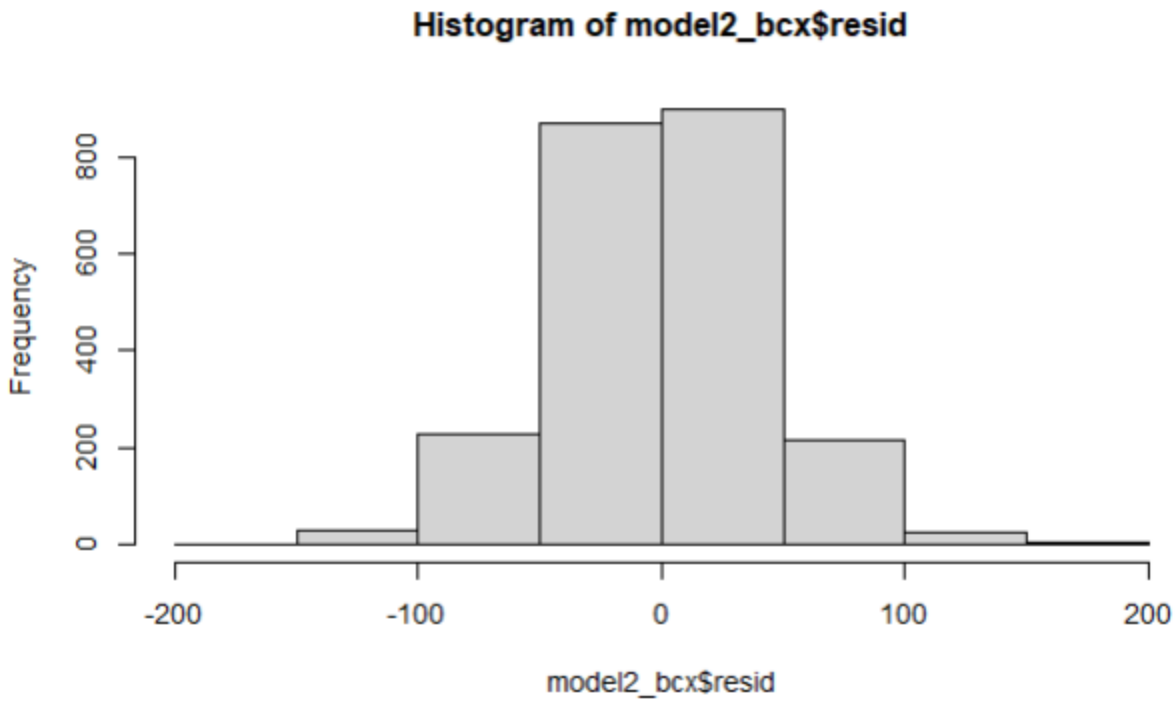


Following these transformations, Box-Cox was used on the best performing transformed model acquire a lambda value for TARGET_WINS. The optimal lambda value was ~1.27. The lambda value was applied to the TARGET_WINS column, and the model was run.

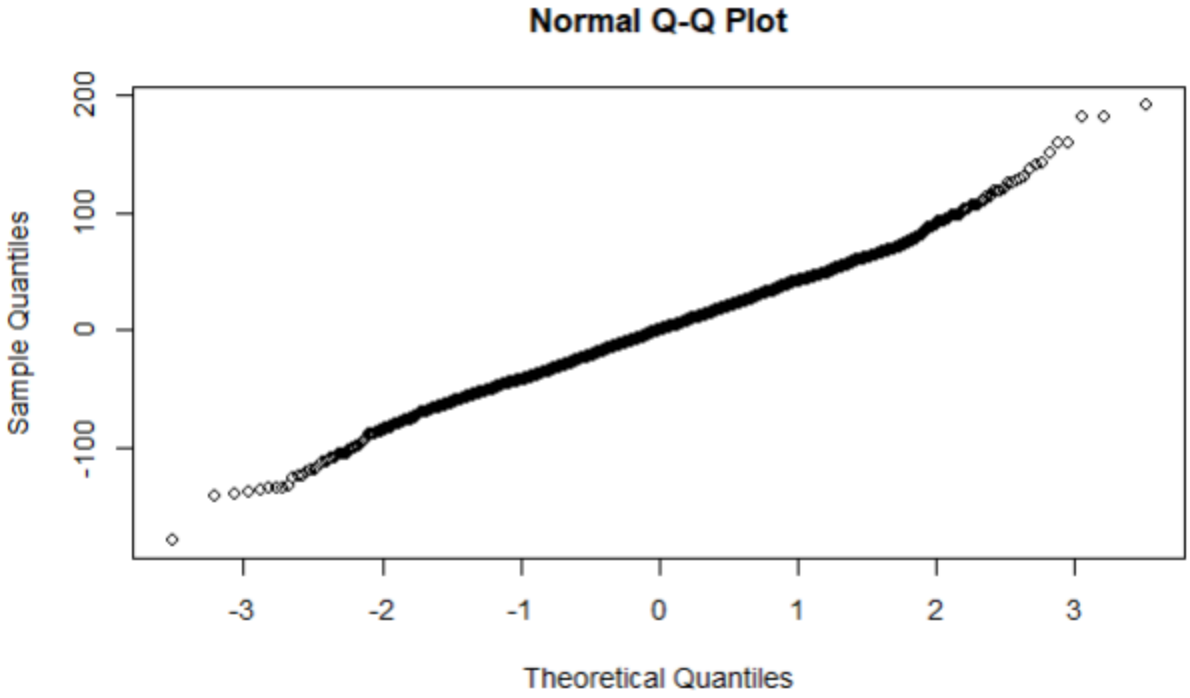
TRANSFORMED MODEL 2 BOXCOX MODEL SUMMARY

```
Call:
lm(formula = TARGET_WINS ~ DEFENSE_INDEX + OFFENSE_INDEX, data = model_2_dfa_transform)

Residuals:
     Min       1Q   Median       3Q      Max
-178.077  -28.910    0.281   27.596  191.053

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -29.63206    9.37499  -3.161  0.00159 **
DEFENSE_INDEX   1.09433    0.25295   4.326 1.58e-05 ***
OFFENSE_INDEX   0.10744    0.00437  24.587  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43.14 on 2271 degrees of freedom
Multiple R-squared:  0.2132,    Adjusted R-squared:  0.2125
F-statistic: 307.7 on 2 and 2271 DF,  p-value: < 2.2e-16
```

TRANSFORMED MODEL 2 BOXCOX HISTOGRAM PLOT



TRANSFORMED MODEL 2 BOXCOX QQ PLOT



Overall, the Box Cox lambda application seems to have worsened the model based on the R^2 and F-Score values, as well as the skewness of it's residuals, going from a -0.012 with the transformed model to a 0.106 with the Box-Cox lambda application. The MODEL 2 Transformed with logs and the adjusted Defensive index was the best model of those tried. With that being said this model still only explained around 21.8 percent of the variance in the TARGET_WINS column. The F-Score was fairly high, helping to support the legitimacy of this modal. The coefficients in this model indicate that the Offensive Index is more important for the TARGET_WIN prediction than the Deffensive index, as their respective coefficients are 239,5 and 3.72. If there was another

variable similar to TEAM_FIELD_E for the offensive side of things to divide by in the Offensive index i wonder how this would shift the model.

Finally, This model was then used to predict TARGET_WIN values in the moneyball_eval_data, specifically after the identical restructuring and processing was completed on that dataset. See below (commented out for knitting):

```
#predictions <- predict(model2_2, newdata=moneyball_eval_formatted)  # Predict on TestData

# JOINING WITH NON-INDEX COLUMN DF, the original imputed data with imputation flags ("M" - MI
ssing | "P" - Present) for any potentatial recreation.
#predicted_win_data <- cbind(moneyball_eval_imputed, PRED_TARGET_WINS = predictions)
#head(predicted_win_data)

#write.csv(predicted_win_data, "TARGET_WINS_Prediction_moneyball_data.csv")
```

The appendix below contains all of my code, most of it relavant to my thought process and some of it unused scrap.

Appendix: R Code

```
#Reading in both data sets, However only inspecting the Training Data for now because that is
what we want to work with.

moneyball_train_data = read.csv(url("https://raw.githubusercontent.com/jhnboyy/CUNY_SPS_WORK/
refs/heads/main/Spring2025/DATA621/data/moneyball-training-data.csv"))
moneyball_eval_data = read.csv(url("https://raw.githubusercontent.com/jhnboyy/CUNY_SPS_WORK/r
efs/heads/main/Spring2025/DATA621/data/moneyball-evaluation-data.csv"))
```

```
# Taking a look at the first five rows, along with the columns and their respective data type
s.
print(head(moneyball_train_data))
```

```
##   INDEX TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
## 1     1          39           1445             194              39
## 2     2          70           1339             219              22
## 3     3          86           1377             232              35
## 4     4          70           1387             209              38
## 5     5          82           1297             186              27
## 6     6          75           1279             200              36
##   TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB
## 1              13             143             842              NA
## 2             190             685            1075              37
## 3             137             602             917              46
## 4              96             451             922              43
## 5             102             472             920              49
## 6              92             443             973             107
##   TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
## 1              NA               NA            9364               84
## 2              28               NA            1347              191
## 3              27               NA            1377              137
## 4              30               NA            1396               97
## 5              39               NA            1297              102
## 6              59               NA            1279               92
##   TEAM_PITCHING_BB TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
## 1              927             5456            1011               NA
## 2              689             1082             193              155
## 3              602              917             175              153
## 4              454              928             164              156
## 5              472              920             138              168
## 6              443              973             123              149
```

```
## Confirming distinct values in df before count and summary
print(nrow(moneyball_train_data)) # 2276
```

```
## [1] 2276
```

```
moneyball_train_data <- moneyball_train_data |> distinct()

print(nrow(moneyball_train_data)) # no Dupes, 2276
```

```
## [1] 2276
```

```
### Summary Stats of the training data.
print(summary(moneyball_train_data))
```

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##                                                    NA's   :102
##  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
##  Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137
##  1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419
##  Median :101.0   Median : 49.0   Median :58.00    Median : 1518
##  Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
##  3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682
##  Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132
##  NA's   :131     NA's   :772     NA's   :2085
##  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :  0.0    Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
##  1st Qu.: 50.0    1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0
##  Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
##  Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
##  3rd Qu.:150.0    3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2
##  Max.   :343.0    Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
##                                    NA's   :102
##  TEAM_FIELDING_DP
##  Min.   : 52.0
##  1st Qu.:131.0
##  Median :149.0
##  Mean   :146.4
##  3rd Qu.:164.0
##  Max.   :228.0
##  NA's   :286
```

```
## Overview Non-Index Columns <datatypes>:  Notes
#TARGET_WINS <int>: No nulls, Range: 0 - 146
#TEAM_BATTING_H <int>: No Nulls, 891 - 2,554
#TEAM_BATTING_2B <int>: No Nulls, 69 - 458
#TEAM_BATTING_3B <int>: No Nulls, 0 - 223
#TEAM_BATTING_HR <int>: No Nulls, 0 - 264
#TEAM_BATTING_BB <int>: No Nulls, 0 - 878
#TEAM_BATTING_SO <int>: 102 Nulls, 0 - 1,399
#TEAM_BASERUN_SB <int>: 131 Nulls, 0 - 697
#TEAM_BASERUN_CS <int>: 772 Nulls, 0 - 201
#TEAM_BATTING_HBP <int>: 2085 Nulls, 29 - 95
#TEAM_PITCHING_H <int>: No Nulls, 1137 - 30,132
#TEAM_PITCHING_HR <int>: No Nulls, 0 - 343
#TEAM_PITCHING_BB <int>: No Nulls, 0 - 3645
#TEAM_PITCHING_SO <int>: 102 Nulls, 0 - 19278
#TEAM_FIELDING_E <int>: No Nulls, 65 - 1898
#TEAM_FIELDING_DP <int>: 286 Nulls, 52 - 228

## TARGET_WINS is the dependent var.
## The other various predictors would be independent vars.
## May need to use imputation for some of these because of nulls.


## Based on Description Table the following relationships may be true:
## TEAM_BATTING_H may be somewhat an aggregate of  TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BAT
TING_HR
## - The sum of the latter three variables minus the TEAM_BATTING_H should be "TEAM_BATTING_1
B"
## - Confirming my Understanding of this:

temp_hit_df <- moneyball_train_data  |>
  dplyr::select(TEAM_BATTING_H,TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BATTING_HR) |>
  mutate("TEAM_BATTING_1B" = TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R))

print(head(temp_hit_df))
```

```
##   TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 1           1445             194              39              13
## 2           1339             219              22             190
## 3           1377             232              35             137
## 4           1387             209              38              96
## 5           1297             186              27             102
## 6           1279             200              36              92
##   TEAM_BATTING_1B
## 1            1199
## 2             908
## 3             973
## 4            1044
## 5             982
## 6             951
```

```
## Are Errors (TEAM_FIELDING_E) and aggregate of the "allowed" categories: TEAM_PITCHING_BB,T
EAM_PITCHING_H,TEAM_PITCHING_HR

temp_error_df <- moneyball_train_data |>
  dplyr::select(TEAM_FIELDING_E,TEAM_PITCHING_BB,TEAM_PITCHING_H,TEAM_PITCHING_HR) |>
  mutate("Allowed_Sum" =  TEAM_PITCHING_BB+TEAM_PITCHING_H+TEAM_PITCHING_HR,
         "equiv_test" = TEAM_FIELDING_E == Allowed_Sum)

print(head(temp_error_df))
```

```
##   TEAM_FIELDING_E TEAM_PITCHING_BB TEAM_PITCHING_H TEAM_PITCHING_HR Allowed_Sum
## 1            1011              927            9364               84       10375
## 2             193              689            1347              191        2227
## 3             175              602            1377              137        2116
## 4             164              454            1396               97        1947
## 5             138              472            1297              102        1871
## 6             123              443            1279               92        1814
##   equiv_test
## 1      FALSE
## 2      FALSE
## 3      FALSE
## 4      FALSE
## 5      FALSE
## 6      FALSE
```

```
## This is not the case, there are no instances where this is true.
temp_error_df |>
  filter(equiv_test==TRUE)
```

```
## [1] TEAM_FIELDING_E  TEAM_PITCHING_BB TEAM_PITCHING_H  TEAM_PITCHING_HR
## [5] Allowed_Sum      equiv_test
## <0 rows> (or 0-length row.names)
```

```
## According to the descriptive document for this homework, these stats are seasonal perforan
ce for a teams over time (1871 - 2006)
## Curious how many teams are in this data, although there is no way to parse them out.
total_years <- 2006-1871
print(total_years)
```

```
## [1] 135
```

```
# Combining both aspects of the data because this is the same data but split for this task:
total_rows <- nrow(moneyball_train_data)+nrow(moneyball_eval_data)

## Total teams covered:
print(total_rows/total_years) ## About 18 - 19 teams.
```

```
## [1] 18.77778
```

```
## ABOVE TEAM WORK NOT APPLICABLE AFTER REREADING THE DESCROPTION OF THE DATA
```

```
# Looking at nulls, For those columns with Nulls See what the other columns look like. Are th
ey the same rows? Are they different rows?

## --- TEAM_BATTING_SO (Batting Strike out nulls)
so_nulls <- moneyball_train_data |> filter(is.na(TEAM_BATTING_SO))

#so_nulls

## Seems that TEAM_BASERUN_CS,TEAM_BATTING_HBP, and TEAM_PITCHING_SO are null in the same ins
tances as the the strike out nulls. Lets Check,

print(unique(so_nulls$TEAM_BASERUN_CS))
```

```
## [1] NA
```

```
print(unique(so_nulls$TEAM_BATTING_HBP))
```

```
## [1] NA
```

```
print(unique(so_nulls$TEAM_PITCHING_SO))
```

```
## [1] NA
```

```
## Confirmed all of these values are null when the Pitching Strike outs are null

## Are the nulls here just zeros? Lets see if zeros exist anywhere in their respective non nu
ll values.
## Using Summary again:
print(summary(moneyball_train_data))
```

```
##      INDEX         TARGET_WINS     TEAM_BATTING_H TEAM_BATTING_2B
##  Min.   :   1.0   Min.   :  0.00   Min.   : 891   Min.   : 69.0
##  1st Qu.: 630.8   1st Qu.: 71.00   1st Qu.:1383   1st Qu.:208.0
##  Median :1270.5   Median : 82.00   Median :1454   Median :238.0
##  Mean   :1268.5   Mean   : 80.79   Mean   :1469   Mean   :241.2
##  3rd Qu.:1915.5   3rd Qu.: 92.00   3rd Qu.:1537   3rd Qu.:273.0
##  Max.   :2535.0   Max.   :146.00   Max.   :2554   Max.   :458.0
##
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.: 34.00   1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 548.0
##  Median : 47.00   Median :102.00   Median :512.0   Median : 750.0
##  Mean   : 55.25   Mean   : 99.61   Mean   :501.6   Mean   : 735.6
##  3rd Qu.: 72.00   3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 930.0
##  Max.   :223.00   Max.   :264.00   Max.   :878.0   Max.   :1399.0
##                                                    NA's   :102
##  TEAM_BASERUN_SB TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H
##  Min.   :  0.0   Min.   :  0.0   Min.   :29.00    Min.   : 1137
##  1st Qu.: 66.0   1st Qu.: 38.0   1st Qu.:50.50    1st Qu.: 1419
##  Median :101.0   Median : 49.0   Median :58.00    Median : 1518
##  Mean   :124.8   Mean   : 52.8   Mean   :59.36    Mean   : 1779
##  3rd Qu.:156.0   3rd Qu.: 62.0   3rd Qu.:67.00    3rd Qu.: 1682
##  Max.   :697.0   Max.   :201.0   Max.   :95.00    Max.   :30132
##  NA's   :131     NA's   :772     NA's   :2085
##  TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##  Min.   :  0.0    Min.   :   0.0   Min.   :    0.0   Min.   :  65.0
##  1st Qu.: 50.0    1st Qu.: 476.0   1st Qu.:  615.0   1st Qu.: 127.0
##  Median :107.0    Median : 536.5   Median :  813.5   Median : 159.0
##  Mean   :105.7    Mean   : 553.0   Mean   :  817.7   Mean   : 246.5
##  3rd Qu.:150.0    3rd Qu.: 611.0   3rd Qu.:  968.0   3rd Qu.: 249.2
##  Max.   :343.0    Max.   :3645.0   Max.   :19278.0   Max.   :1898.0
##                                    NA's   :102
##  TEAM_FIELDING_DP
##  Min.   : 52.0
##  1st Qu.:131.0
##  Median :149.0
##  Mean   :146.4
##  3rd Qu.:164.0
##  Max.   :228.0
##  NA's   :286
```

```
#TEAM_BATTING_SO - Zeros present
##TEAM_BASERUN_SB - Zeros present
#TEAM_BASERUN_CS - Zeros present
#TEAM_BATTING_HBP - No Zeros
#TEAM_PITCHING_SO - Zeros present
#TEAM_FIELDING_DP - No zeros present

## Most of the columns that have nulls also have zeros, so structurally speaking the nulls ar
e most likely not zeros and will need imputation, or we will need to remove the rows.

## What proportion of the data is null, would it be worth removing?
no_nulls<-moneyball_train_data |>
  filter(!is.na(TEAM_BATTING_SO),
         !is.na(TEAM_BASERUN_SB),
         !is.na(TEAM_BASERUN_CS),
         !is.na(TEAM_BATTING_HBP),
         !is.na(TEAM_PITCHING_SO),
         !is.na(TEAM_FIELDING_DP))


#print(paste("Only", round(nrow(no_nulls)/ nrow(moneyball_train_data),2)*100,"percent of rows
from the original",nrow(moneyball_train_data),"rows have no nulls, we will need to impute."))
```

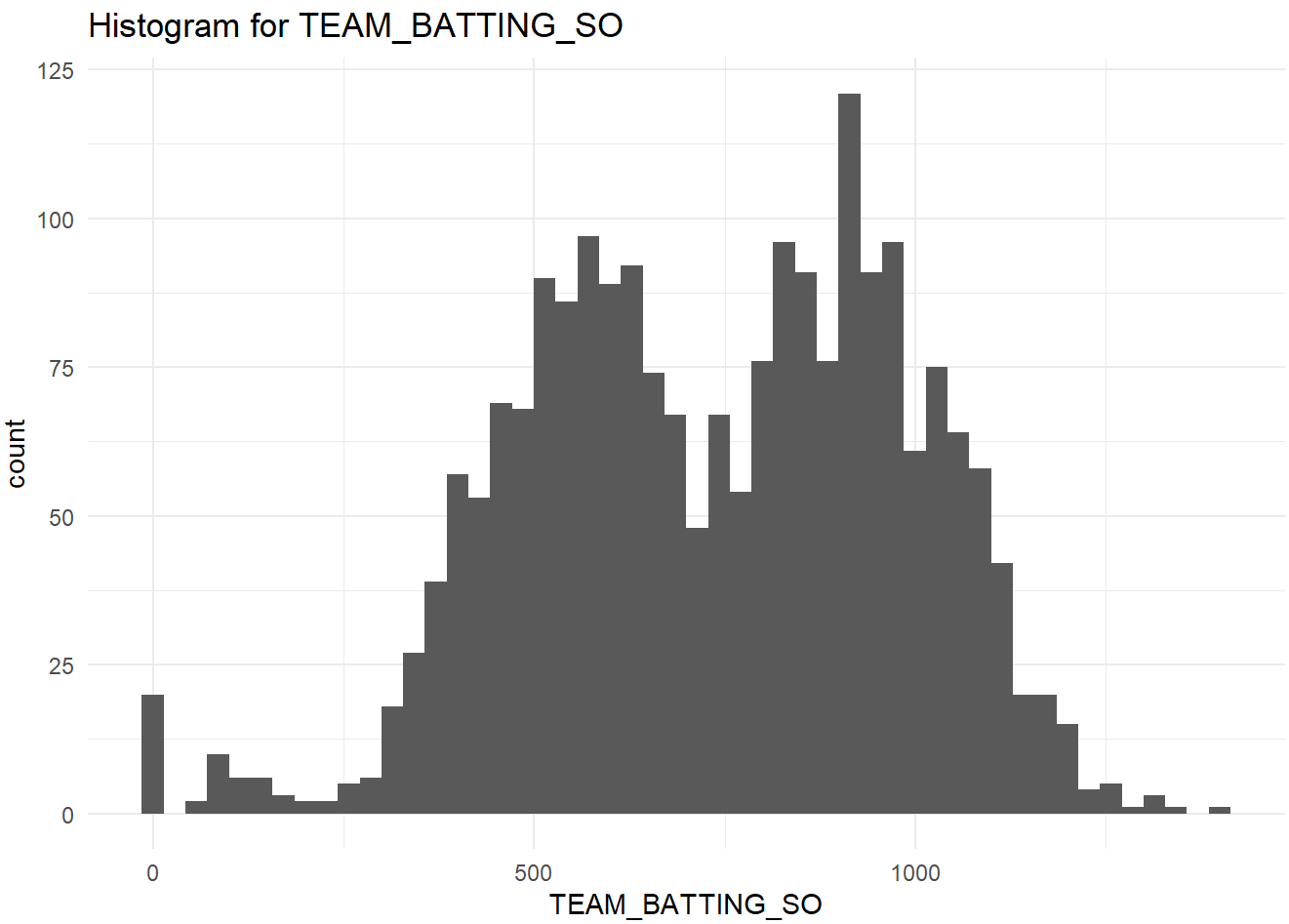
```
## Taking a look at the data with pair plots
#colnames(moneyball_train_data)
#"TARGET_WINS", "TEAM_BATTING_H"   "TEAM_BATTING_2B"  "TEAM_BATTING_3B"  "TEAM_BATTING_HR"  "
TEAM_BATTING_BB"  "TEAM_BATTING_SO" "TEAM_BASERUN_SB"  "TEAM_BASERUN_CS"  #"TEAM_BATTING_HBP"
"TEAM_PITCHING_H"  "TEAM_PITCHING_HR" "TEAM_PITCHING_BB" "TEAM_PITCHING_SO" "TEAM_FIELDING_E"
"TEAM_FIELDING_DP"

moneyball_train_nona <- na.omit(moneyball_train_data)
## NOTE THIS IS A VERY SMALL SUBSET BECAUSE OF THE NULLS
print(nrow(moneyball_train_nona)) #191
```

```
## [1] 191
```

```
p <- ggpairs(moneyball_train_nona,
           progress = FALSE) +
          theme_minimal(base_size=9) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
                                     strip.text.x = element_text(angle = 90, hjust = 1),
                                    strip.text.y = element_text(angle = 0, hjust = 1))
## Saving Image for reference in PDF #
#ggsave("ggpairs_plot.png", plot = p, width = 15, height = 20, units = "in", dpi = 300)

### Notesworthy Points from the GGPAIRs plot.
#-- TEAM_BATTING_H and TEAM_PITCHING_H have a direct linear relationship, not independent
#-- TEAM_BATTING_SO and TEAM_PITCHING_SO have a direct linear relationship, not independent.
#-- TEAM_BATTING_BB and TEAM_PITCHING_BB have a direct linear relationship, not independent.
#-- Positive relationship between TEAM_BASERUN_SB and TEAM_BASERUN_CS, may want to merge thes
e to one var.
#-- There may be some other relationships that slip past the visual checks, but will deal wit
h those later.

### Action Items for modeling:
## Should keep just one of each of these variables when modeling for sake of predictor indepe
ndence.
## Similarly, we should parse out the 1H from the TEAM_BATTING_H or just keep only the agg va
r for modeling.
## Similar actions needed for TEAM_PITCHING_H and TEAM_PITCHING_HR, these variables are relat
ed as HR is a subset of hits.
## TEAM_BASERUN_SB and TEAM_BASERUN_CS aggregated into one Stealing base index, because notic
able relationship in plot
```

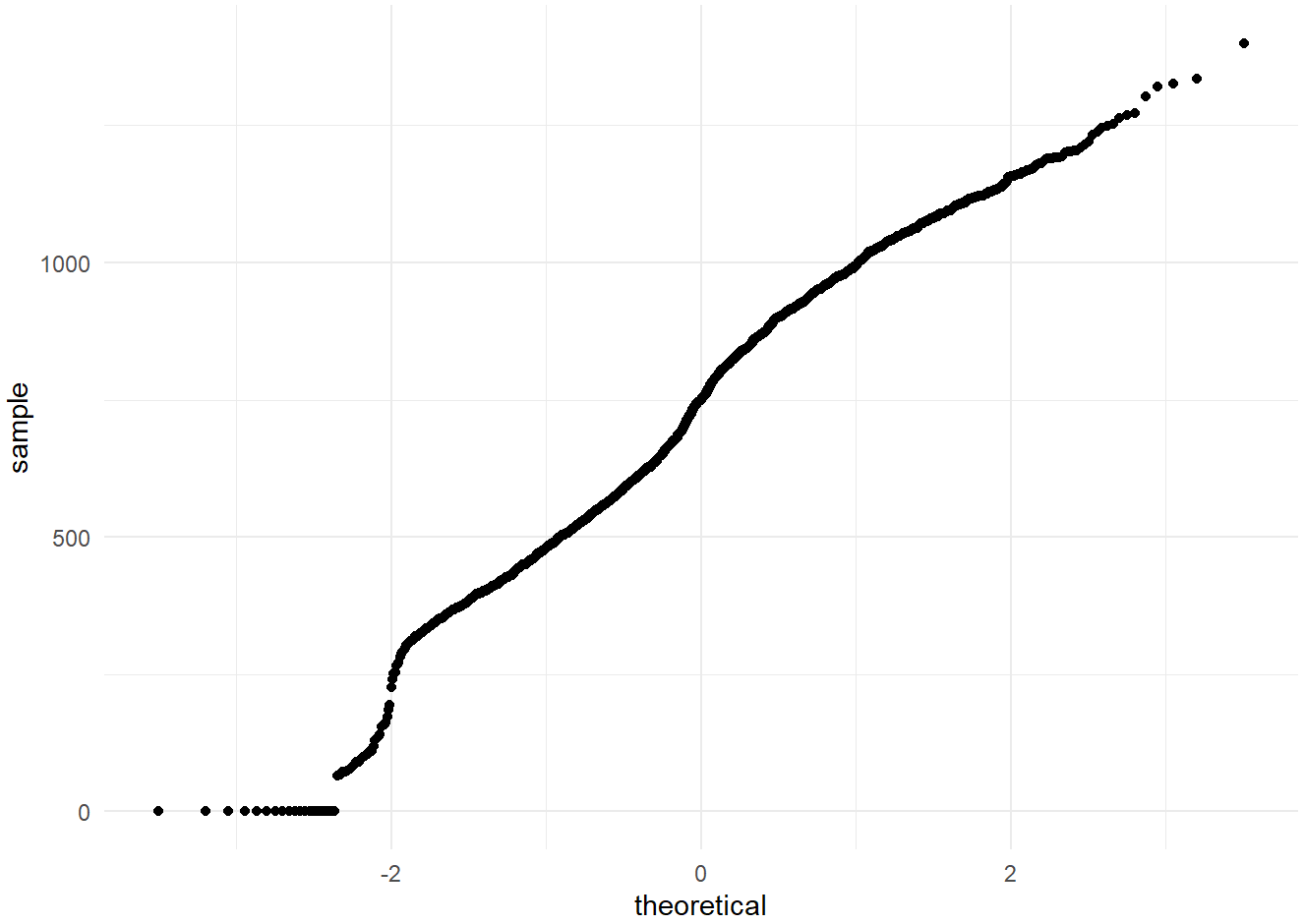
```
## Looking at the columns with nulls.

## Plot work for moneyball_train_data$TEAM_BATTING_SO
ggplot(moneyball_train_data, aes(x = TEAM_BATTING_SO)) +
  geom_histogram( bins = 50) +
  labs(title = "Histogram for TEAM_BATTING_SO") +
  theme_minimal()
```

```
## Warning: Removed 102 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



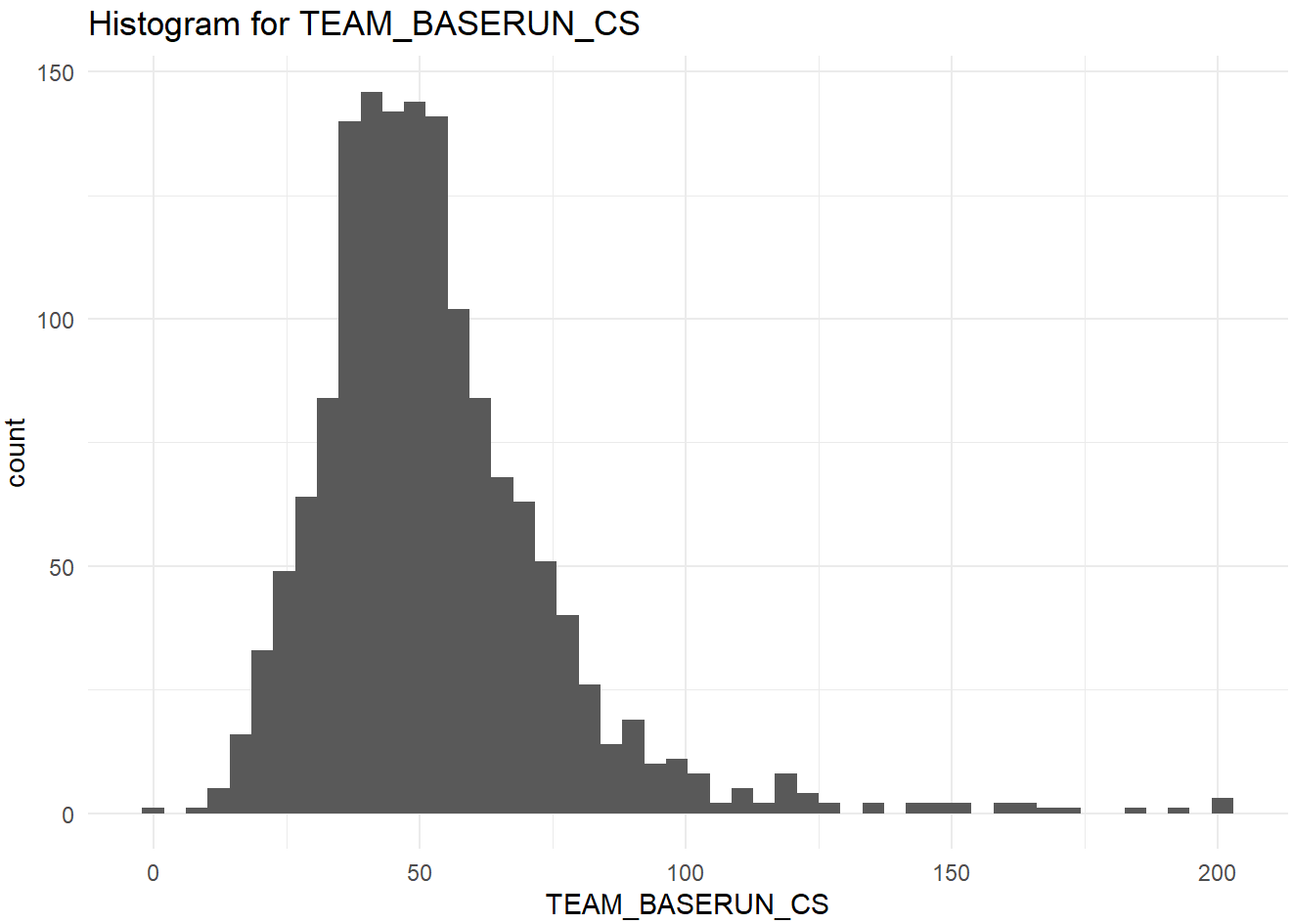

```
ggplot(moneyball_train_data, aes(sample = TEAM_BATTING_SO)) +
  stat_qq() +
  theme_minimal()
```

```
## Warning: Removed 102 rows containing non-finite outside the scale range
## (`stat_qq()`).
```



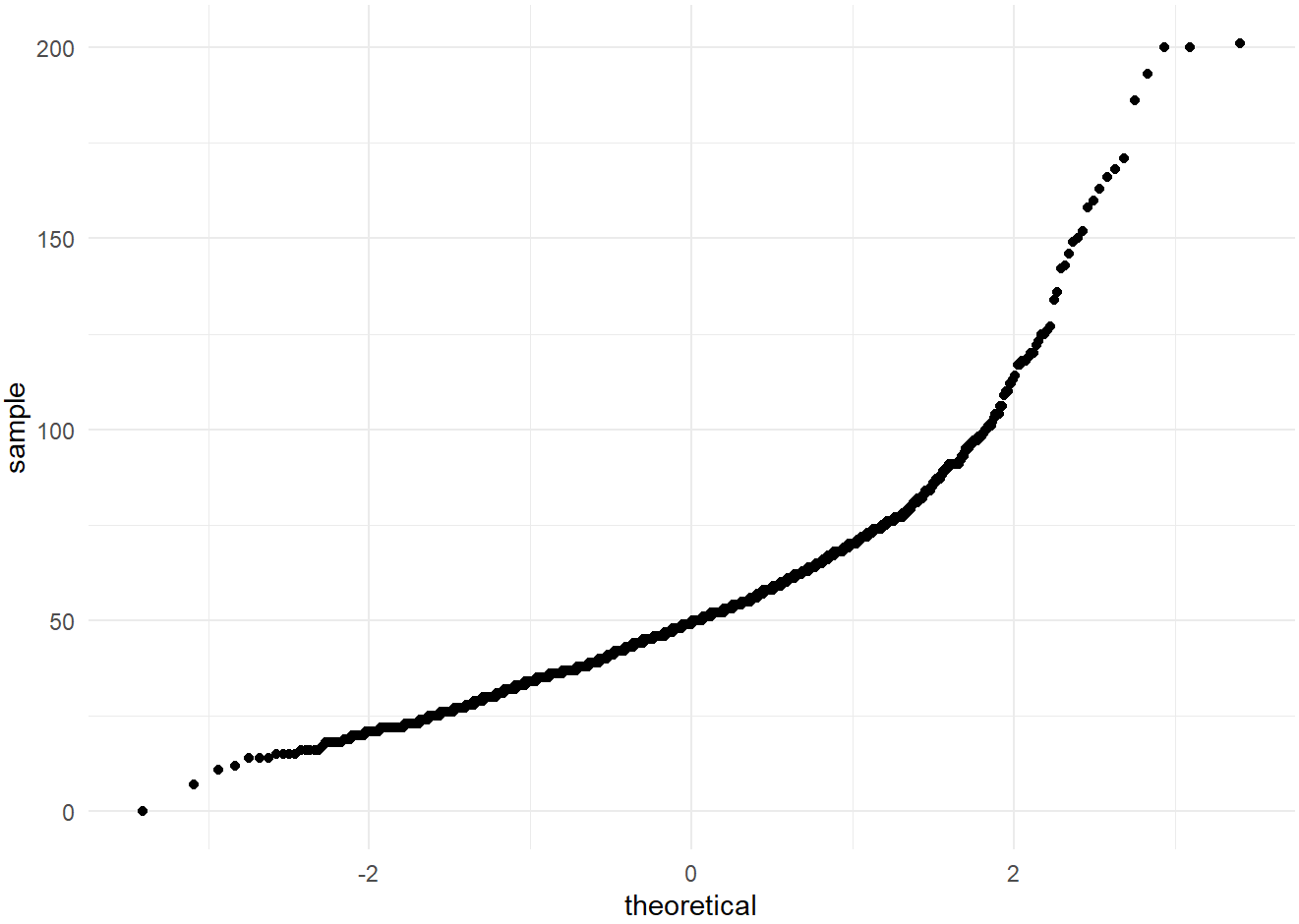

```
## Distribution conclusion: Not normal, camel shaped

## Plot work for moneyball_train_data$TEAM_BASERUN_CS
ggplot(moneyball_train_data, aes(x = TEAM_BASERUN_CS)) +
  geom_histogram( bins = 50) +
  labs(title = "Histogram for TEAM_BASERUN_CS") +
  theme_minimal()
```

```
## Warning: Removed 772 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



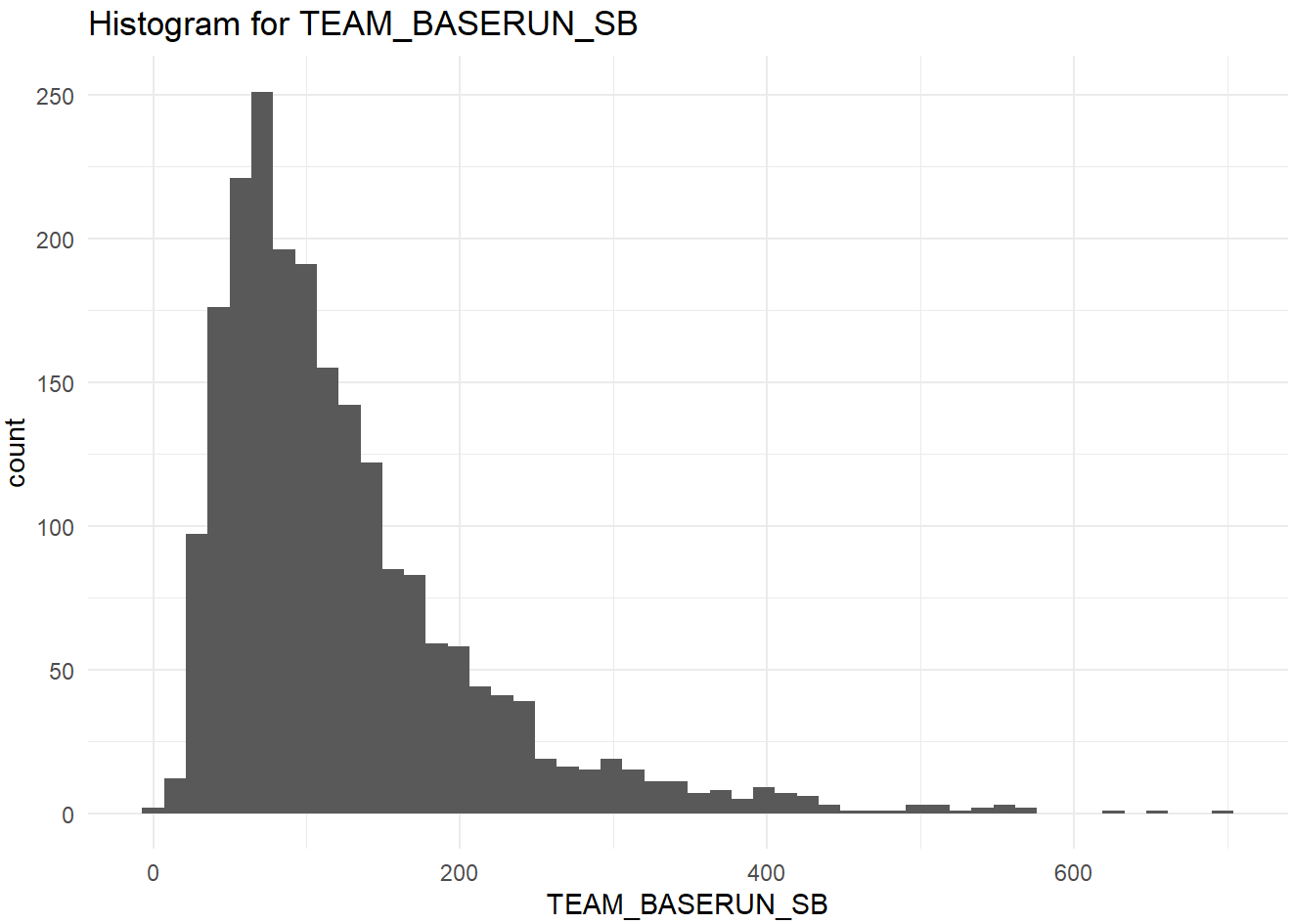

```
ggplot(moneyball_train_data, aes(sample = TEAM_BASERUN_CS)) +
  stat_qq() +
  theme_minimal()
```

```
## Warning: Removed 772 rows containing non-finite outside the scale range
## (`stat_qq()`).
```



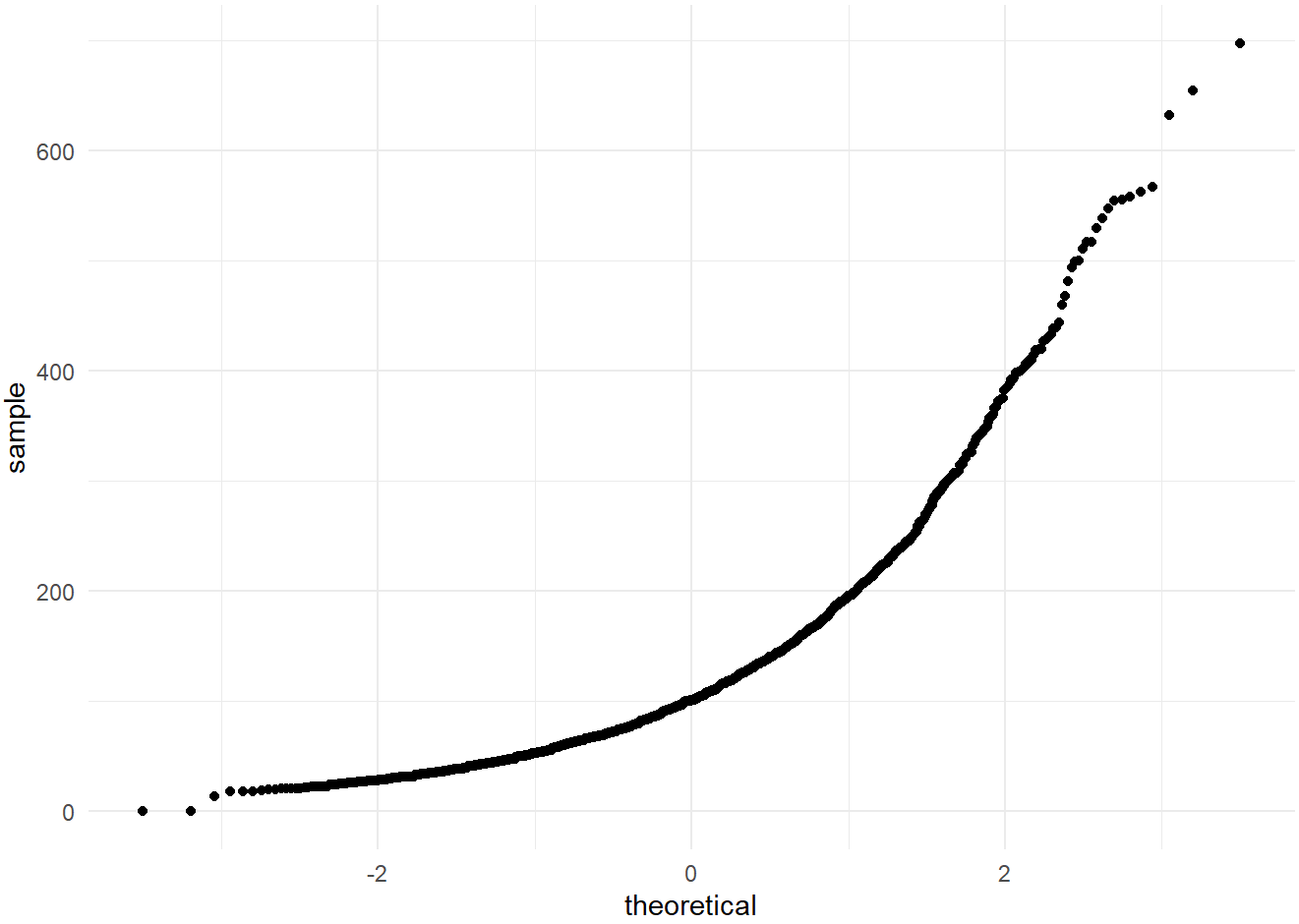

```
## Distribution conclusion: Right Skew in the data, not normal

## Plot work for moneyball_train_data$TEAM_BASERUN_SB
ggplot(moneyball_train_data, aes(x = TEAM_BASERUN_SB)) +
  geom_histogram( bins = 50) +
  labs(title = "Histogram for TEAM_BASERUN_SB") +
  theme_minimal()
```

```
## Warning: Removed 131 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



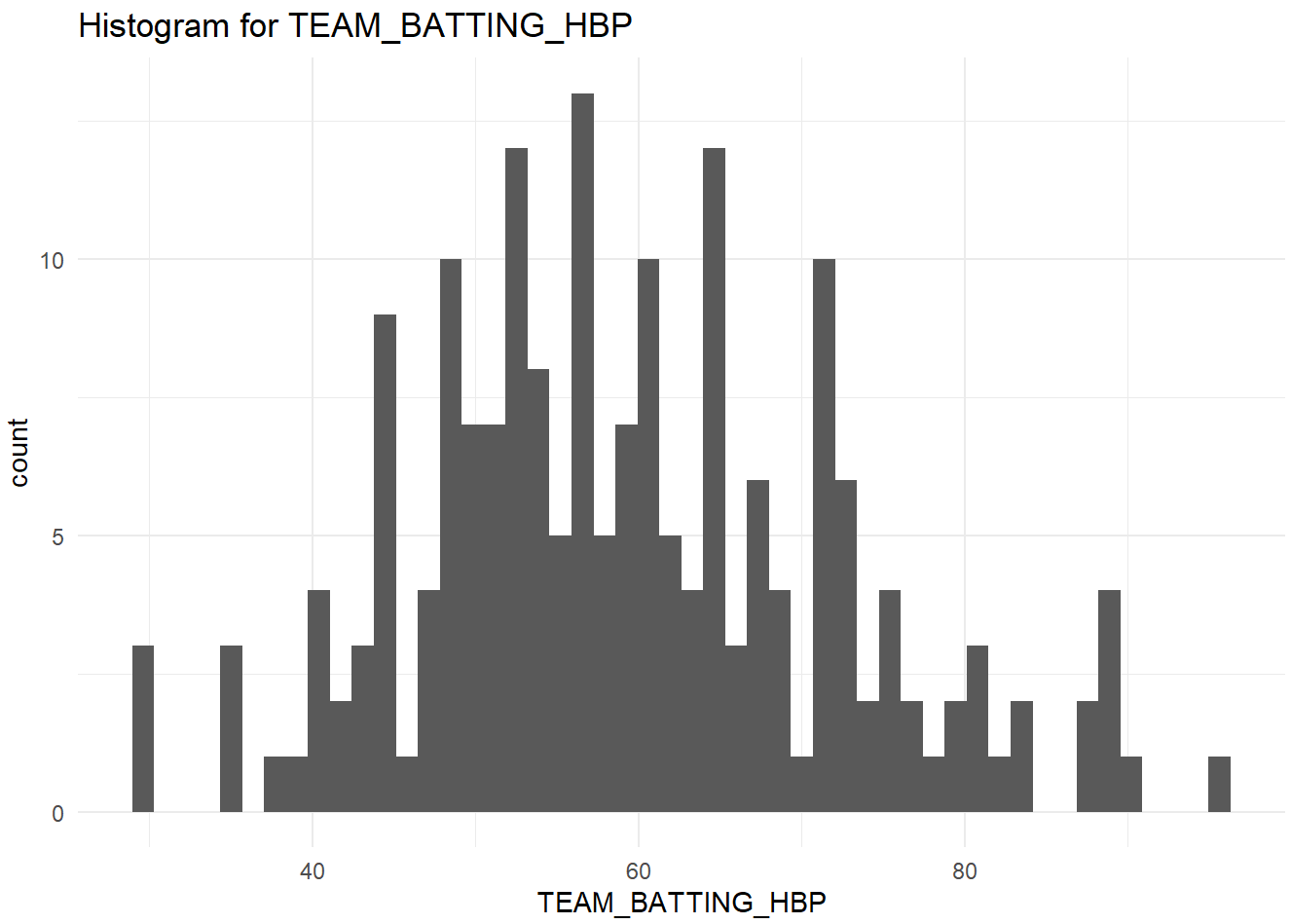

```
ggplot(moneyball_train_data, aes(sample = TEAM_BASERUN_SB)) +
  stat_qq() +
  theme_minimal()
```

```
## Warning: Removed 131 rows containing non-finite outside the scale range
## (`stat_qq()`).
```



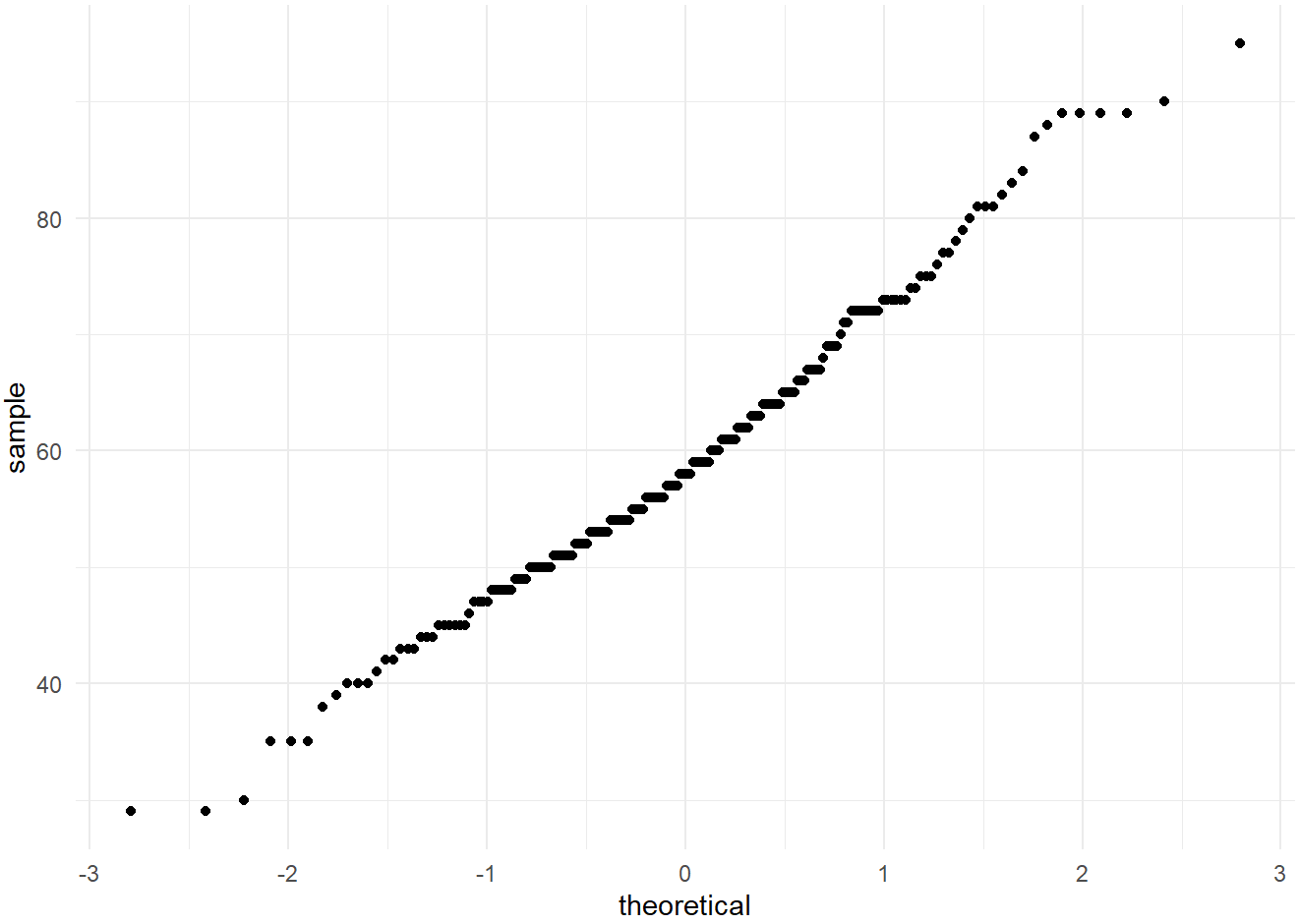

```
## Distribution conclusion: Right Skew in the data, not normal.

## Plot work for moneyball_train_data$TEAM_BATTING_HBP
ggplot(moneyball_train_data, aes(x = TEAM_BATTING_HBP)) +
  geom_histogram( bins = 50) +
  labs(title = "Histogram for TEAM_BATTING_HBP") +
  theme_minimal()
```

```
## Warning: Removed 2085 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



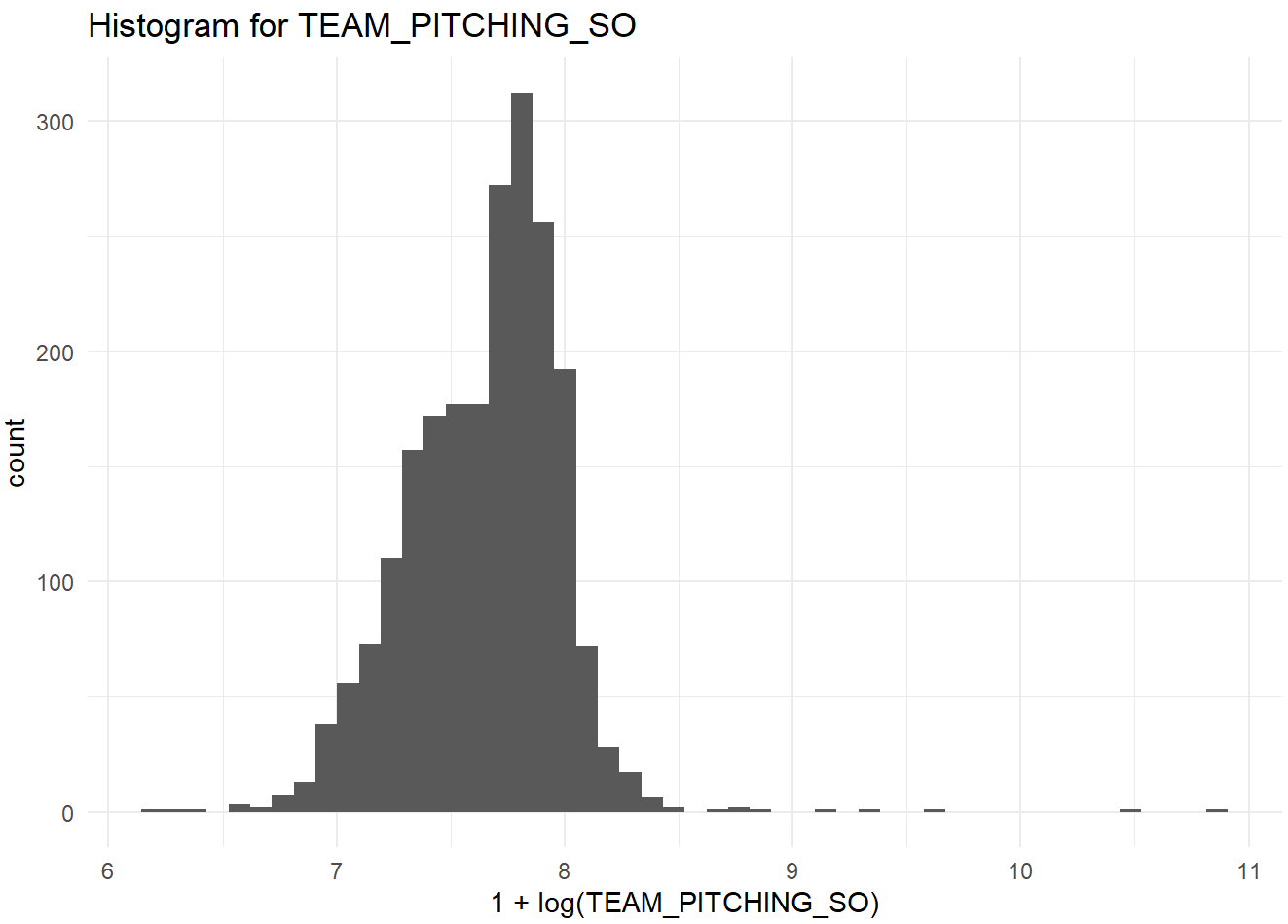

```
ggplot(moneyball_train_data, aes(sample = TEAM_BATTING_HBP)) +
  stat_qq() +
  theme_minimal()
```

```
## Warning: Removed 2085 rows containing non-finite outside the scale range
## (`stat_qq()`).
```



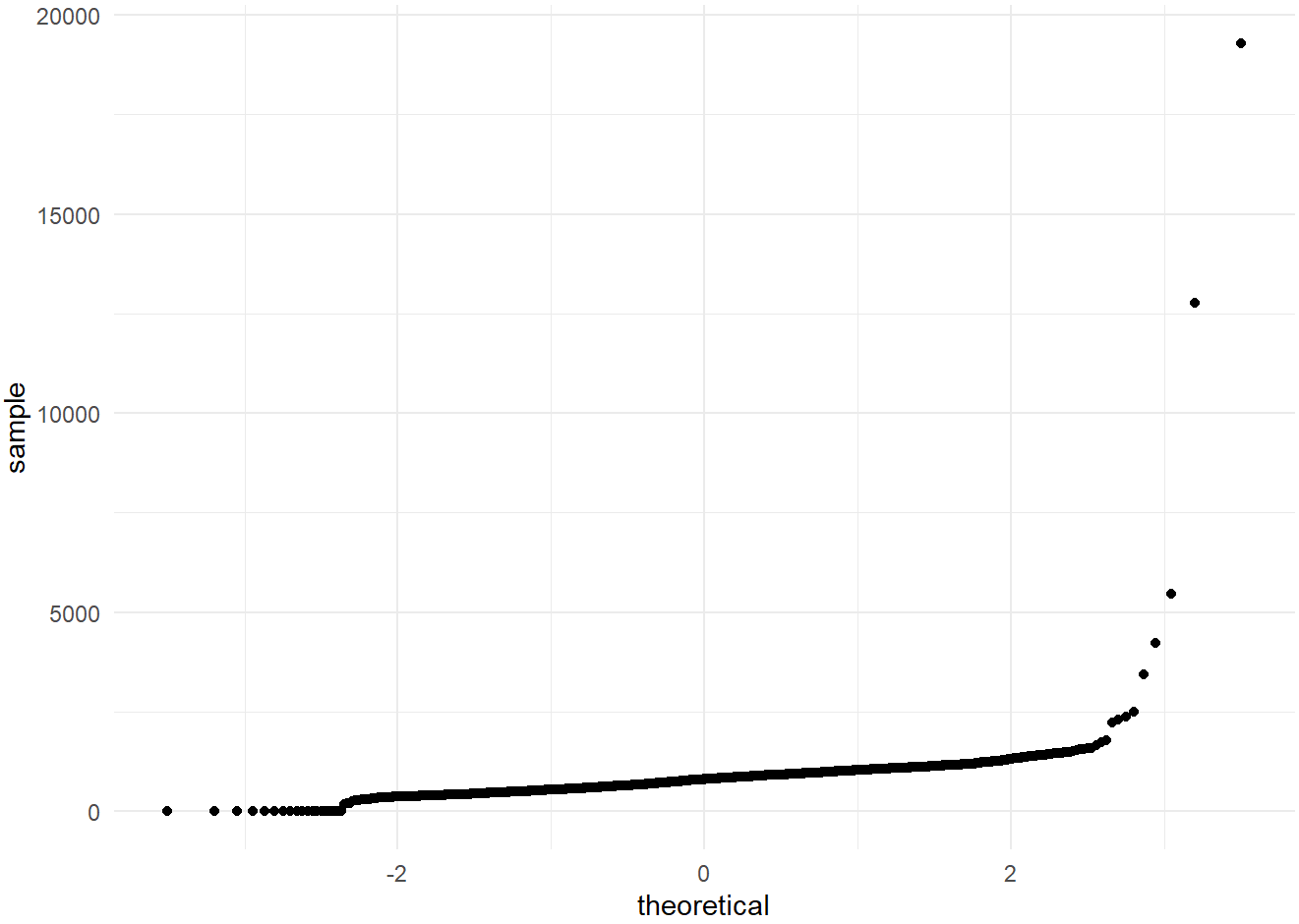

```
## Distribution conclusion: Seems normal. Easily seen when bin number is lowered, qq plot mos
tly comfirms

## Plot work for moneyball_train_data$TEAM_PITCHING_SO
ggplot(moneyball_train_data, aes(x = 1+log(TEAM_PITCHING_SO))) +
  geom_histogram( bins = 50) +
  labs(title = "Histogram for TEAM_PITCHING_SO") +
  theme_minimal()
```

```
## Warning: Removed 122 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(moneyball_train_data, aes(sample = TEAM_PITCHING_SO)) +
  stat_qq() +
  theme_minimal()
```

```
## Warning: Removed 102 rows containing non-finite outside the scale range
## (`stat_qq()`).
```




```
## Distribution conclusion: Seems normal, with a super high outlier. May need to deal with th
is
summary(moneyball_train_data$TEAM_PITCHING_SO)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   615.0   813.5   817.7   968.0 19278.0     102
```

```
sort(unique(moneyball_train_data$TEAM_PITCHING_SO)) ## Two very large outliers: 12758, 19278
```

```
##   [1]    0  181  205  208  252  270  272  292  301  306  310  316
##  [13]  320  326  329  341  345  358  361  362  363  364  366  367
##  [25]  369  373  374  375  376  378  380  381  383  384  386  387
##  [37]  390  391  392  393  395  396  398  400  401  402  406  407
##  [49]  409  410  411  412  414  416  417  419  420  422  423  426
##  [61]  427  431  432  433  434  436  437  438  440  443  444  446
##  [73]  447  448  449  450  451  454  455  459  460  461  462  463
##  [85]  464  465  466  467  468  469  470  471  473  475  477  478
##  [97]  479  480  481  483  484  485  486  488  489  490  491  493
## [109]  494  495  496  497  498  499  500  502  503  504  505  506
## [121]  507  508  510  511  512  513  514  515  516  517  518  519
## [133]  520  521  522  524  525  526  527  528  529  530  531  533
## [145]  534  535  536  537  539  540  541  542  543  545  546  547
## [157]  548  549  551  552  553  554  555  556  557  558  559  560
## [169]  561  562  563  564  565  566  567  568  569  570  571  572
## [181]  573  574  575  576  577  578  579  580  581  582  583  584
## [193]  585  586  587  588  589  590  591  592  593  594  595  596
## [205]  597  598  599  600  601  602  603  604  605  606  607  608
## [217]  609  610  611  612  613  614  615  616  617  619  620  621
## [229]  622  623  624  625  626  627  628  629  630  631  633  634
## [241]  635  636  637  638  639  640  641  642  643  644  645  646
## [253]  647  648  649  650  651  652  653  654  655  656  657  658
## [265]  659  660  661  662  663  664  665  666  667  668  669  670
## [277]  671  672  673  674  675  678  679  680  681  682  683  684
## [289]  685  686  687  688  690  691  692  693  694  695  696  697
## [301]  698  700  702  703  704  705  707  708  709  710  711  712
## [313]  713  714  715  716  717  718  719  720  721  722  723  724
## [325]  725  726  727  728  729  730  731  732  733  734  735  736
## [337]  737  738  739  740  741  742  743  744  745  746  747  749
## [349]  750  751  752  753  754  755  756  757  758  759  760  761
## [361]  762  764  765  766  767  768  769  770  771  772  774  775
## [373]  777  778  779  781  782  784  785  787  788  789  790  791
## [385]  792  793  794  795  796  797  799  800  801  802  803  804
## [397]  805  806  807  808  809  810  811  812  813  814  815  816
## [409]  817  818  819  820  821  822  823  824  825  826  827  828
## [421]  829  830  831  832  833  834  835  836  837  838  839  840
## [433]  841  842  843  844  845  846  847  848  849  850  851  852
## [445]  853  854  855  856  857  858  860  861  862  863  864  865
## [457]  866  867  868  869  870  871  872  873  874  875  876  877
## [469]  879  880  881  882  883  884  885  887  888  889  890  891
## [481]  892  893  896  897  898  899  900  901  902  903  904  905
## [493]  906  907  908  909  910  911  912  913  914  915  916  917
## [505]  918  919  920  921  922  923  924  925  926  927  928  929
## [517]  930  931  932  933  934  935  936  937  938  939  940  941
## [529]  942  943  944  946  947  948  949  950  951  952  953  954
## [541]  955  956  957  958  959  960  961  962  963  964  965  966
## [553]  967  968  969  970  971  972  973  974  975  976  977  978
## [565]  979  980  981  982  983  984  985  986  987  988  989  990
## [577]  992  993  994  995  996  997  998 1000 1001 1003 1004 1005
## [589] 1006 1008 1009 1010 1011 1012 1013 1014 1015 1016 1018 1019
## [601] 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031
```

```
## [613]  1032  1033  1034  1035  1036  1037  1038  1039  1040  1041  1042  1043
## [625]  1044  1045  1046  1047  1048  1049  1052  1053  1054  1055  1056  1057
## [637]  1058  1059  1060  1061  1062  1063  1066  1067  1068  1069  1070  1071
## [649]  1072  1073  1074  1075  1076  1077  1078  1079  1080  1081  1082  1083
## [661]  1084  1085  1087  1088  1089  1090  1091  1092  1093  1094  1095  1096
## [673]  1097  1098  1099  1102  1103  1104  1105  1106  1107  1108  1109  1110
## [685]  1111  1112  1113  1114  1116  1117  1119  1120  1121  1122  1123  1124
## [697]  1125  1126  1129  1130  1132  1133  1134  1136  1138  1141  1143  1144
## [709]  1145  1146  1148  1153  1155  1156  1157  1158  1159  1160  1161  1162
## [721]  1164  1167  1168  1169  1170  1172  1173  1176  1179  1180  1181  1185
## [733]  1188  1189  1190  1191  1192  1193  1194  1197  1198  1200  1203  1204
## [745]  1205  1206  1208  1210  1211  1217  1221  1222  1233  1237  1239  1245
## [757]  1247  1248  1249  1253  1256  1257  1258  1261  1269  1273  1275  1276
## [769]  1284  1293  1296  1299  1303  1305  1313  1326  1328  1335  1342  1349
## [781]  1350  1351  1354  1356  1364  1370  1371  1385  1386  1387  1399  1405
## [793]  1412  1419  1424  1427  1430  1434  1436  1459  1461  1464  1474  1481
## [805]  1491  1511  1541  1552  1561  1590  1600  1659  1739  1781  2225  2309
## [817]  2367  2492  3450  4224  5456 12758 19278
```

```
## Is this ONLY two rows?
nrow(moneyball_train_data |> filter(TEAM_PITCHING_SO>12000)) # Confirmed only two rows, may j
ust want ot remove these.
```

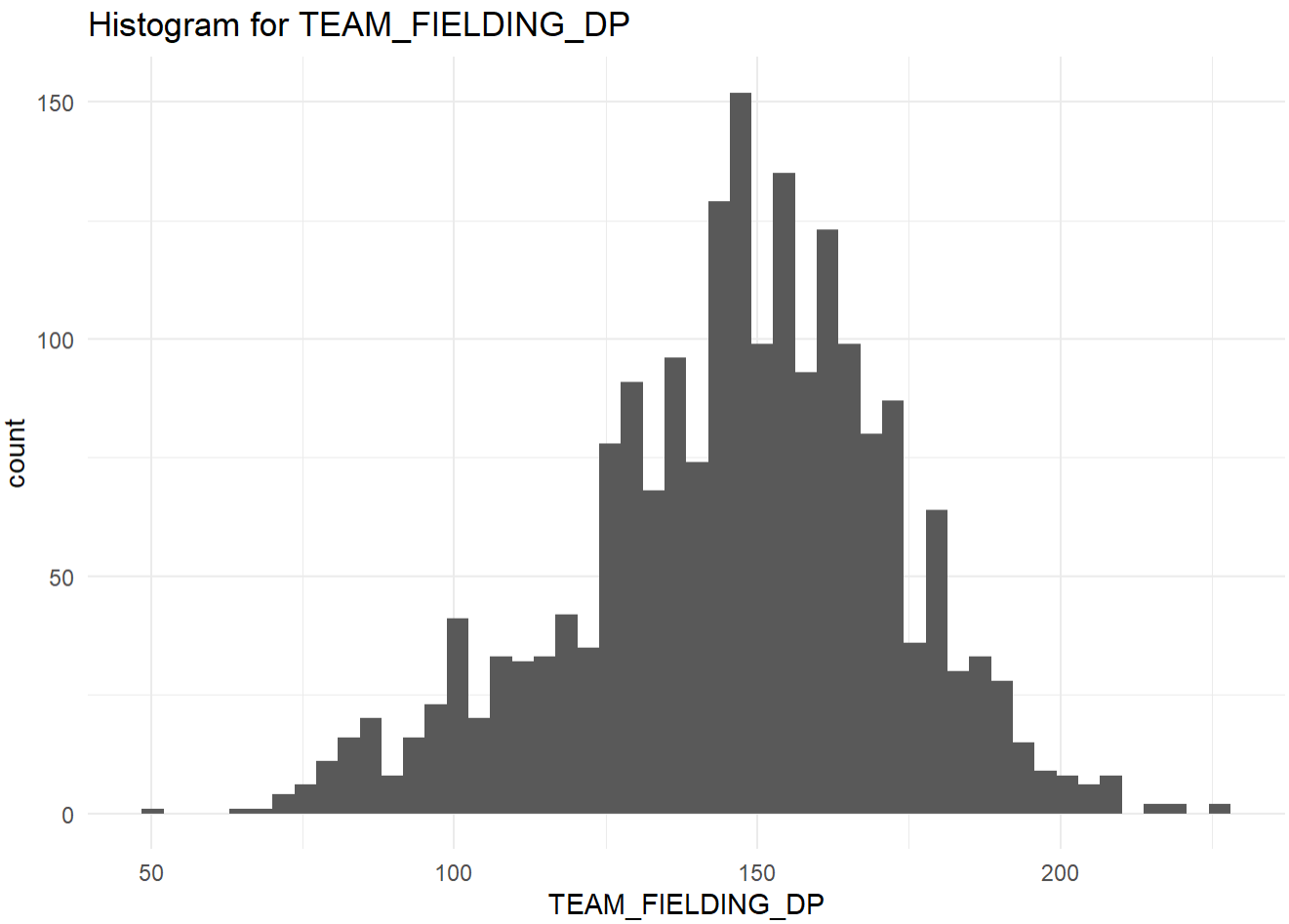
```
## [1] 2
```

```
moneyball_train_data |> filter(TEAM_PITCHING_SO>12000)
```

```
##   INDEX TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
## 1  1494         108           1188             338               0
## 2  2380          41            992             263              20
##   TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB
## 1               0             270             945              NA
## 2               0             142             952              NA
##   TEAM_BASERUN_CS TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
## 1              NA               NA           16038                0
## 2              NA               NA           20088                0
##   TEAM_PITCHING_BB TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
## 1             3645            12758             716               NA
## 2             2876            19278             952               NA
```

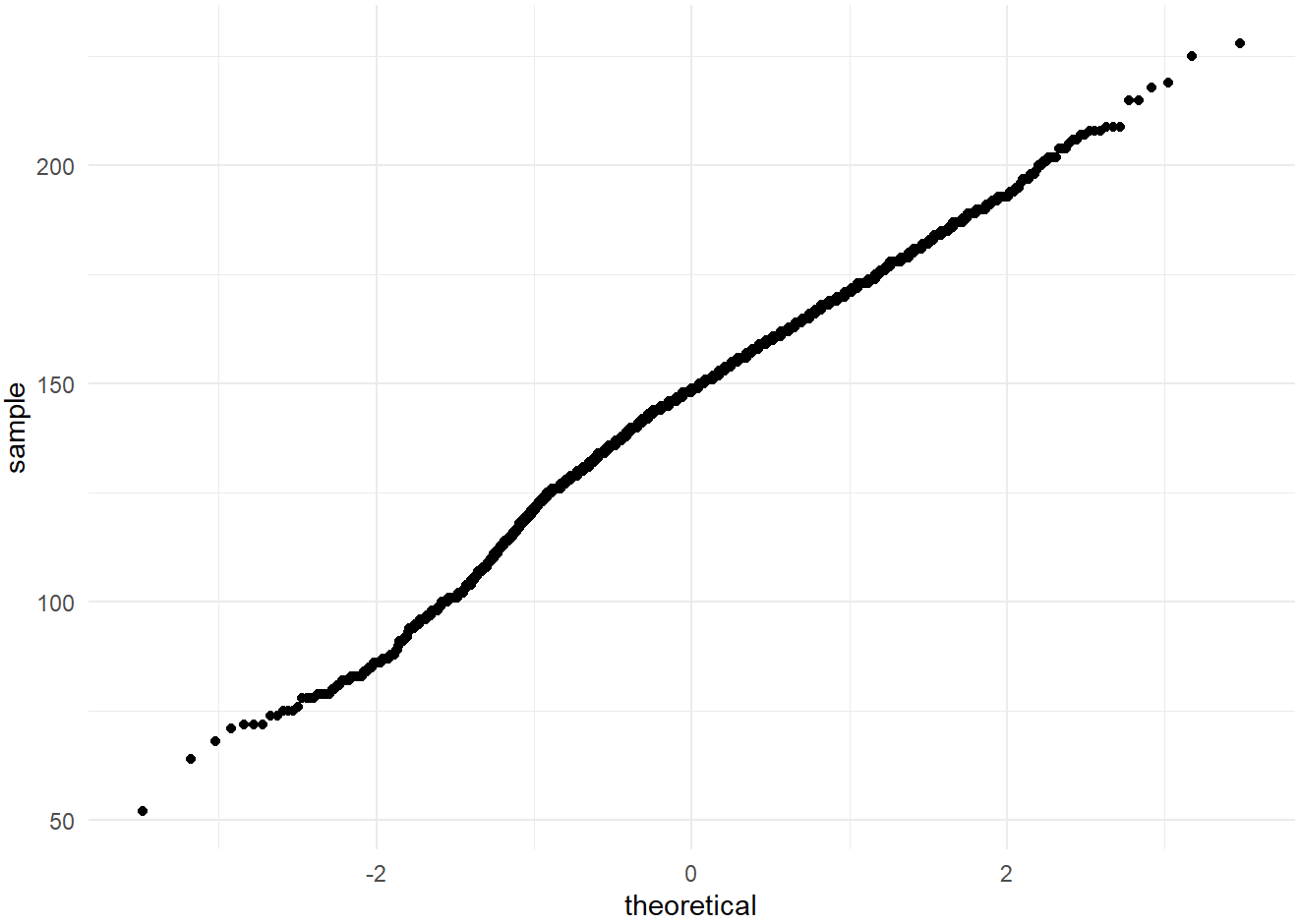
```
## Plot work for moneyball_train_data$TEAM_FIELDING_DP
ggplot(moneyball_train_data, aes(x = TEAM_FIELDING_DP)) +
  geom_histogram( bins = 50) +
  labs(title = "Histogram for TEAM_FIELDING_DP") +
  theme_minimal()
```

```
## Warning: Removed 286 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(moneyball_train_data, aes(sample = TEAM_FIELDING_DP)) +
  stat_qq() +
  theme_minimal()
```

```
## Warning: Removed 286 rows containing non-finite outside the scale range
## (`stat_qq()`).
```




```
## Distribution conclusion: Seems normal
```

```
#, fig.width = 5, fig.height =10}
## Using Mice Native tool to example the pattern of nulls:
##Imputation Reference: https://www.r-bloggers.com/2015/10/imputing-missing-data-with-r-mice-package/
#md.pattern(moneyball_train_data,rotate.names = TRUE)


## Viz Notes:
#- Enough data to impute TEAM_BATTING_SO and TEAM_PITCHING_SO. Need to impute.
#- TEAM_BASERUN_CS and TEAM_BASERUN_SB it may be worth just to agg to a column that is total attempts at base stealing for team, as this strategy's impact is what we want to know.No need to impute
#- Can probably combine the TEAM_BATTING_HBP and TEAM_BATTING_BB as they are essentially equi valents for impact on game. As total batting walks. No need to impute with HBP because way to o many nulls.

## Looking into TEAM_FIELDING_DP (286 Nulls vs)

nrow(moneyball_train_data |> filter(!is.na(TEAM_FIELDING_DP)))#1,990 non null rows can impute . ~14% nulls
```

```
## [1] 1990
```

```
### Takeaways for Imputation Process:
# - I was going to try different methods within MICE, but decided to just keep all of them PM
M in order to be consistent for explaining. The TEAM_PITCHING_SO outliers shouild be handled
separately though.
# - Overall To Do's:
# --- TEAM_BATTING_SO - Imputation with MICE PMM Needed
# --- TEAM_PITCHING_SO - Imputation with MICE PMM Needed (Probably need to remove the two out
liers from the TEAM_PITCHING_SO column before imputation)
# --- TEAM_BASERUN_SB and TEAM_BASERUN_CS to be combined into TEAM_BASERUN_STEAL_ATTEMPTS; No
Imputation needed.
# --- TEAM_BATTING_HBP can be combined with TEAM_BATTING_BB (no null values) into TEAM_BATTIN
G_WALK_TOTAL
# --- TEAM_FIELDING_DP -Imputation with MICE PMM Needed
```

```
### USING ACTION ITEMS AS GUIDE
## Adding imputation /missing data Flags to keep track of the changes:
## Present (P) and Missing (M)
moneyball_train_data$TEAM_BATTING_SO_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_BATTING_SO), 'M', 'P')
moneyball_train_data$TEAM_PITCHING_SO_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_PITCHING_SO), 'M', 'P')
moneyball_train_data$TEAM_BASERUN_SB_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_BASERUN_SB), 'M', 'P')
moneyball_train_data$TEAM_BASERUN_CS_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_BASERUN_CS), 'M', 'P')
moneyball_train_data$TEAM_BATTING_BB_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_BATTING_BB), 'M', 'P')
moneyball_train_data$TEAM_BATTING_HBP_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_BATTING_HBP), 'M', 'P')
moneyball_train_data$TEAM_FIELDING_DP_FLAG <- ifelse(is.na(moneyball_train_data$TEAM_FIELDING_DP), 'M', 'P')


# --- TEAM_BASERUN_SB and TEAM_BASERUN_CS to be combined into TEAM_BASERUN_STEAL_ATTEMPTS; No Imputation needed for certain models
# --- TEAM_BATTING_HBP can be combined with TEAM_BATTING_BB (no null values) into TEAM_BATTING_WALK_TOTAL
moneyball_train_processed <- moneyball_train_data |>
  mutate(TEAM_BASERUN_STEAL_ATTEMPTS = ifelse(is.na(TEAM_BASERUN_SB), 0, TEAM_BASERUN_SB) + ifelse(is.na(TEAM_BASERUN_CS), 0, TEAM_BASERUN_CS),
         TEAM_BATTING_WALK_TOTAL = ifelse(is.na(TEAM_BATTING_BB), 0, TEAM_BATTING_BB) + ifelse(is.na(TEAM_BATTING_HBP), 0, TEAM_BATTING_HBP)) |>
  dplyr::select(-TEAM_BASERUN_CS, -TEAM_BATTING_HBP) #-TEAM_BASERUN_SB,-TEAM_BATTING_BB
## In these aggregate derivative rows, im considering the nulls to be zero for summation.



# --- TEAM_PITCHING_SO,TEAM_BATTING_SO,TEAM_FIELDING_DP,TEAM_BASERUN_SB,TEAM_BATTING_BB - Imputation with MICE PMM Needed. Probably need to remove the two outlier columns from the TEAM_PITCHING_SO column, and the data at large.
outlier_rows <- moneyball_train_processed |> filter(TEAM_PITCHING_SO>12000)
no_outlier_moneyball_train_processed <- moneyball_train_processed |> filter(TEAM_PITCHING_SO<12000 | is.na(TEAM_PITCHING_SO))
imputed <- mice(no_outlier_moneyball_train_processed,m=5,meth='pmm',seed=5)
```

```
##
##  iter imp variable
##   1   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
```

```
## Warning: Number of logged events: 7
```

```
moneyball_train_imputed<-complete(imputed)

#Checking for nulls where there shouldnt be.
print(nrow(moneyball_train_imputed |> filter(is.na(TEAM_PITCHING_SO))))
```

```
## [1] 0
```

```
## Working DF
print(head(moneyball_train_imputed))
```

```
##   INDEX TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
## 1     1          39           1445             194              39
## 2     2          70           1339             219              22
## 3     3          86           1377             232              35
## 4     4          70           1387             209              38
## 5     5          82           1297             186              27
## 6     6          75           1279             200              36
##   TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB
## 1              13             143             842              39
## 2             190             685            1075              37
## 3             137             602             917              46
## 4              96             451             922              43
## 5             102             472             920              49
## 6              92             443             973             107
##   TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
## 1            9364               84              927             5456
## 2            1347              191              689             1082
## 3            1377              137              602              917
## 4            1396               97              454              928
## 5            1297              102              472              920
## 6            1279               92              443              973
##   TEAM_FIELDING_E TEAM_FIELDING_DP TEAM_BATTING_SO_FLAG TEAM_PITCHING_SO_FLAG
## 1            1011              190                    P                     P
## 2             193              155                    P                     P
## 3             175              153                    P                     P
## 4             164              156                    P                     P
## 5             138              168                    P                     P
## 6             123              149                    P                     P
##   TEAM_BASERUN_SB_FLAG TEAM_BASERUN_CS_FLAG TEAM_BATTING_BB_FLAG
## 1                    M                    M                    P
## 2                    P                    P                    P
## 3                    P                    P                    P
## 4                    P                    P                    P
## 5                    P                    P                    P
## 6                    P                    P                    P
##   TEAM_BATTING_HBP_FLAG TEAM_FIELDING_DP_FLAG TEAM_BASERUN_STEAL_ATTEMPTS
## 1                     M                     M                           0
## 2                     M                     P                          65
## 3                     M                     P                          73
## 4                     M                     P                          73
## 5                     M                     P                          88
## 6                     M                     P                         166
##   TEAM_BATTING_WALK_TOTAL
## 1                     143
## 2                     685
## 3                     602
## 4                     451
## 5                     472
## 6                     443
```

```
## Now that the data is imputed, and some of the columns have been aggregated, taking the nex
t step in processing and/or variable preparation and selection for the models.
## Printing this to look at the columns again:
#colnames(moneyball_train_imputed)

## Thoughts
#- Batting vs BaseRun vs Fielding vs Pitching Agg vars -> Offesne vs Defense
#- Parse out the 1HB, se can do a correlation with the type of hits vs total wins.
#- Steals vs caught steals, maybe also this coirrelation with the hits?
#- Need to keep in mind we need to predict winds for the team.


## Three main Model Ideas that will dictate further processing:
# --- MODEL 1: Parse out "TEAM_BATTING_1B" from TEAM_BATTING_H, then can do analysis on the d
ifferent types of base hits and their correlation with wins.
# --- MODEL 2: Aggregate into Two index Defense and Offense (Pitching and Field VAS Batting a
nd Baserun Metrics) to see which is more important
# --- MODEL 3: High Risk Offensive Model, Looking at if the "higher risk" offensive plays are
worth it.
# --- MODEL 4: Pitching Index Model vs Field Index for Defense analysis.

#### Keeping two dfs for each model (except 1), one with the non aggregated values and one wi
th custom agg vars.

### MODEL 1 Processing
# --- Limiting to TARGET_WINS and the parsed out Hit columns
model_1_df <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_BATTING_H,TEAM_BATTING_2B,TEAM_BATTING_3B,TEAM_BATTING_HR)|>
  mutate (TEAM_BATTING_1B = (TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R))) |>
  dplyr::select(-TEAM_BATTING_H)
print(head(model_1_df))
```

```
##   TARGET_WINS TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR TEAM_BATTING_1B
## 1          39             194              39              13            1199
## 2          70             219              22             190             908
## 3          86             232              35             137             973
## 4          70             209              38              96            1044
## 5          82             186              27             102             982
## 6          75             200              36              92             951
```

```
### MODEL 2 Processing
## Grouping Proper Defense and Offense. Ensuring that those non-independent vars (e.g.,TEAM_B
ATTING_H and TEAM_PITCHING_H) as accounted for.
# --- OTHER INCLUSION:
#     TARGET_WINS
#     TEAM_BATTING_SO_FLAG
#     TEAM_PITCHING_SO_FLAG
#     TEAM_BATTING_BB_FLAG
#     TEAM_BATTING_HBP_FLAG
#     TEAM_FIELDING_DP_FLAG
#     TEAM_BASERUN_SB_FLAG

# --- DEFENSE INCLUSION:
      #-- TEAM_PITCHING_SO
      #-- TEAM_FIELDING_DP
      #-- TEAM_FIELDING_E

# --- OFFENSE INCLUSION:
      #-- TEAM_BATTING_H
      #-- TEAM_BATTING_BB
      #-- TEAM_BASERUN_SB

#-- NOTE: In order to account for the non-independent vars on both sides, for each index we a
re only considering those that positively impact the index from the redundant variables. For
instance, from the variables TEAM_PITCHING_SO and TEAM_BATTING_SO, only TEAM_PITCHING_SO will
be used because it would positively contribute to a good defense. Howver, variables like TEAM
_FIELDING_E that do not have another 1:1 relationship with another column will be used as a f
actor in defense index.

## Custom Aggregate
model_2_dfa <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_BATTING_SO_FLAG,TEAM_PITCHING_SO_FLAG,TEAM_BATTING_BB_FLAG,T
EAM_BATTING_HBP_FLAG,TEAM_FIELDING_DP_FLAG,TEAM_BASERUN_SB_FLAG,
         TEAM_PITCHING_SO,TEAM_FIELDING_DP,TEAM_FIELDING_E,TEAM_BATTING_H,TEAM_BATTING_WALK_T
OTAL,TEAM_BATTING_BB,TEAM_BASERUN_SB) |>
  mutate(DEFENSE_INDEX=((TEAM_PITCHING_SO+TEAM_FIELDING_DP)-TEAM_FIELDING_E),
         OFFENSE_INDEX=(TEAM_BATTING_H+TEAM_BATTING_BB+TEAM_BASERUN_SB)) |>
  dplyr::select(-TEAM_PITCHING_SO,-TEAM_FIELDING_DP,-TEAM_FIELDING_E,-TEAM_BATTING_H,-TEAM_BA
TTING_WALK_TOTAL,-TEAM_BATTING_BB,-TEAM_BASERUN_SB)

print(head(model_2_dfa))
```

```
##   TARGET_WINS TEAM_BATTING_SO_FLAG TEAM_PITCHING_SO_FLAG TEAM_BATTING_BB_FLAG
## 1          39                    P                     P                    P
## 2          70                    P                     P                    P
## 3          86                    P                     P                    P
## 4          70                    P                     P                    P
## 5          82                    P                     P                    P
## 6          75                    P                     P                    P
##   TEAM_BATTING_HBP_FLAG TEAM_FIELDING_DP_FLAG TEAM_BASERUN_SB_FLAG
## 1                     M                    M                   M
## 2                     M                    P                   P
## 3                     M                    P                   P
## 4                     M                    P                   P
## 5                     M                    P                   P
## 6                     M                    P                   P
##   DEFENSE_INDEX OFFENSE_INDEX
## 1          4635          1627
## 2          1044          2061
## 3           895          2025
## 4           920          1881
## 5           950          1818
## 6           999          1829
```

```
## GRANULAR DEFENSE - TEAM_PITCHING_SO TEAM_FIELDING_DP TEAM_FIELDING_E
model_2_df_granular_defense <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_BATTING_SO_FLAG,TEAM_PITCHING_SO_FLAG,TEAM_BATTING_BB_FLAG,T
EAM_BATTING_HBP_FLAG,TEAM_FIELDING_DP_FLAG,TEAM_BASERUN_SB_FLAG,
         TEAM_PITCHING_SO,TEAM_FIELDING_DP,TEAM_FIELDING_E)
print(head(model_2_df_granular_defense))
```

```
##   TARGET_WINS TEAM_BATTING_SO_FLAG TEAM_PITCHING_SO_FLAG TEAM_BATTING_BB_FLAG
## 1          39                    P                     P                    P
## 2          70                    P                     P                    P
## 3          86                    P                     P                    P
## 4          70                    P                     P                    P
## 5          82                    P                     P                    P
## 6          75                    P                     P                    P
##   TEAM_BATTING_HBP_FLAG TEAM_FIELDING_DP_FLAG TEAM_BASERUN_SB_FLAG
## 1                     M                    M                   M
## 2                     M                    P                   P
## 3                     M                    P                   P
## 4                     M                    P                   P
## 5                     M                    P                   P
## 6                     M                    P                   P
##   TEAM_PITCHING_SO TEAM_FIELDING_DP TEAM_FIELDING_E
## 1             5456              190            1011
## 2             1082              155             193
## 3              917              153             175
## 4              928              156             164
## 5              920              168             138
## 6              973              149             123
```

```
## GRANULAR OFFENSE - TEAM_BATTING_H TEAM_BATTING_BB TEAM_BASERUN_SB (Keeping TEAM_BATTING_WA
LK_TOTAL b/c agg of HBP is not really offense, out of team's control)
model_2_df_granualar_offense <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_BATTING_SO_FLAG,TEAM_PITCHING_SO_FLAG,TEAM_BATTING_BB_FLAG,T
EAM_BATTING_HBP_FLAG,TEAM_FIELDING_DP_FLAG,TEAM_BASERUN_SB_FLAG,
         TEAM_BATTING_H,TEAM_BATTING_BB,TEAM_BASERUN_SB)


### MODEL 3 Risk Index

# --- OTHER INCLUSION:
#     TARGET_WINS
#     TEAM_BATTING_BB_FLAG
#     TEAM_BASERUN_CS_FLAG
#     TEAM_BATTING_HBP_FLAG
#     TEAM_BASERUN_SB_FLAG

#-- HIGH RISK OFFENSE:
#     TEAM_BASERUN_STEAL_ATTEMPTS
#     TEAM_BATTING_2B
#     TEAM_BATTING_3B
#-- LOW RISK OFFENSE:
#     TEAM_BATTING_1B
#     TEAM_BATTING_WALK_TOTAL

## Custom Aggregate
model_3_dfa <- moneyball_train_imputed |>
  mutate (TEAM_BATTING_1B = (TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R))) |>
  dplyr::select( TARGET_WINS,TEAM_BATTING_BB_FLAG,TEAM_BASERUN_CS_FLAG, TEAM_BATTING_HBP_FLA
G,TEAM_BASERUN_SB_FLAG,TEAM_BATTING_HR,TEAM_BATTING_1B,
          TEAM_BATTING_WALK_TOTAL,TEAM_BATTING_2B,TEAM_BATTING_3B,TEAM_BASERUN_STEAL_ATTEMPT
S) |>
  mutate (HIGH_RISK =   (TEAM_BASERUN_STEAL_ATTEMPTS+TEAM_BATTING_2B+TEAM_BATTING_3B),
          LOW_RISK = (TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL+TEAM_BATTING_HR)) |>
  dplyr::select(-TEAM_BATTING_1B,-TEAM_BATTING_WALK_TOTAL,-TEAM_BATTING_2B,-TEAM_BATTING_3B,-
TEAM_BASERUN_STEAL_ATTEMPTS,-TEAM_BATTING_HR)

print(head(model_3_dfa))
```

```
##   TARGET_WINS TEAM_BATTING_BB_FLAG TEAM_BASERUN_CS_FLAG TEAM_BATTING_HBP_FLAG
## 1          39                    P                    M                     M
## 2          70                    P                    P                     M
## 3          86                    P                    P                     M
## 4          70                    P                    P                     M
## 5          82                    P                    P                     M
## 6          75                    P                    P                     M
##   TEAM_BASERUN_SB_FLAG HIGH_RISK LOW_RISK
## 1                    M       233     1355
## 2                    P       306     1783
## 3                    P       340     1712
## 4                    P       320     1591
## 5                    P       301     1556
## 6                    P       402     1486
```

```
## High Risk Activities Granular: TEAM_BASERUN_STEAL_ATTEMPTS , TEAM_BATTING_2B,  TEAM_BATTIN
G_3B
model_3_df_high_risk_granular <- moneyball_train_imputed |>
  mutate (TEAM_BATTING_1B = (TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R))) |>
  dplyr::select(TARGET_WINS,TEAM_BATTING_BB_FLAG,TEAM_BASERUN_CS_FLAG, TEAM_BATTING_HBP_FLA
G,TEAM_BASERUN_SB_FLAG,
          TEAM_BATTING_2B,TEAM_BATTING_3B,TEAM_BASERUN_STEAL_ATTEMPTS)

## Low Risk Activities Granular: TEAM_BATTING_1B, TEAM_BATTING_WALK_TOTAL
model_3_df_low_risk_granular <- moneyball_train_imputed |>
  mutate (TEAM_BATTING_1B = (TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R))) |>
  dplyr::select( TARGET_WINS,TEAM_BATTING_BB_FLAG,TEAM_BASERUN_CS_FLAG, TEAM_BATTING_HBP_FLA
G,TEAM_BASERUN_SB_FLAG,
          TEAM_BATTING_1B, TEAM_BATTING_HR, TEAM_BATTING_WALK_TOTAL)


## MODEL 4 Processing

#--DEFENSE Analysis - Which is more important pitching or field play
# --- OTHER INCLUSION:
#     TARGET_WINS
#     TEAM_PITCHING_SO_FLAG
#     TEAM_FIELDING_DP_FLAG

# --- FIELDING INCLUSION:
#TEAM_FIELDING_DP
#TEAM_FIELDING_E

# --- PITCHING INCLUSION:
#TEAM_PITCHING_BB
#TEAM_PITCHING_H (TEAM_PITCHING_HR is included in TEAM_PITCHING_H)
#TEAM_PITCHING_SO

## Custom Aggregate
model_4_dfa <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_PITCHING_SO_FLAG, TEAM_FIELDING_DP_FLAG, TEAM_FIELDING_DP, T
EAM_FIELDING_E, TEAM_PITCHING_BB, TEAM_PITCHING_H, TEAM_PITCHING_SO) |>
  mutate(FIELD_WORK_INDEX = (TEAM_FIELDING_DP-TEAM_FIELDING_E),
         PITCH_WORK_INDEX = (TEAM_PITCHING_SO-(TEAM_PITCHING_BB+TEAM_PITCHING_H)))|>
  dplyr::select(-TEAM_FIELDING_DP,-TEAM_FIELDING_E,-TEAM_PITCHING_BB,-TEAM_PITCHING_H,-TEAM_P
ITCHING_SO)

print(head(model_4_dfa))
```

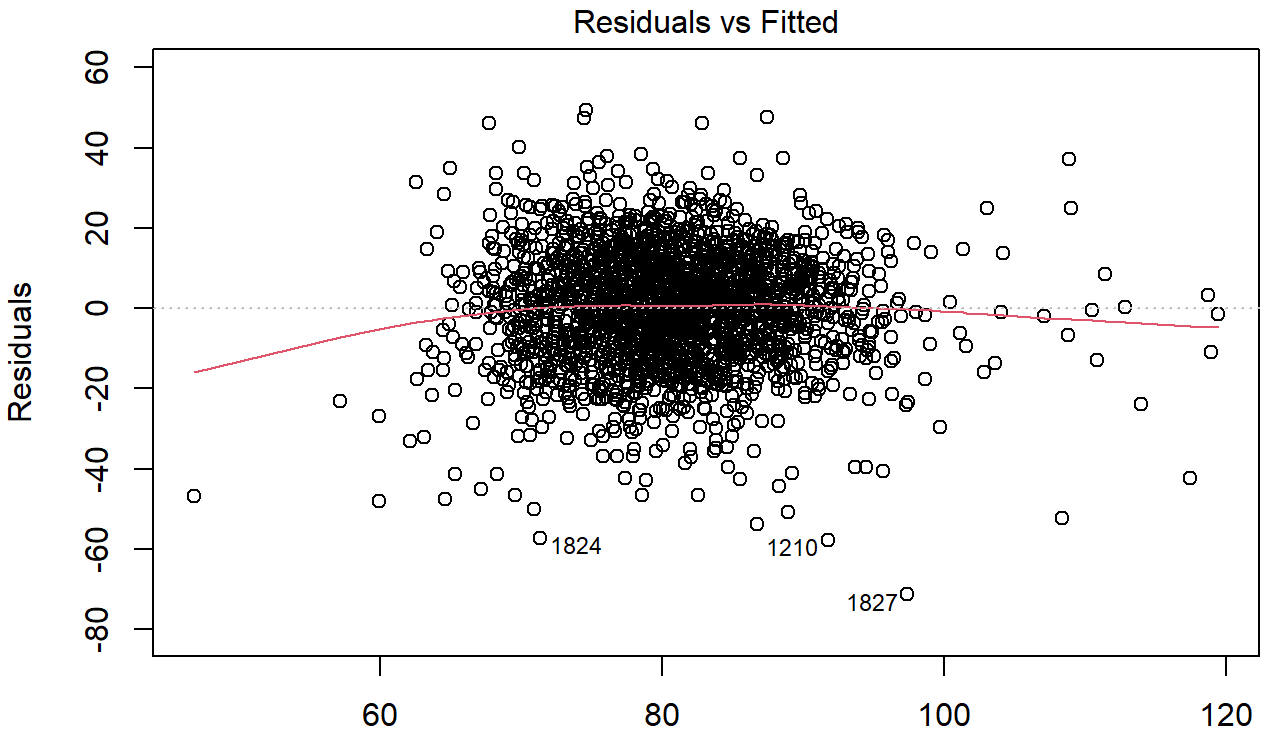
```
##   TARGET_WINS TEAM_PITCHING_SO_FLAG TEAM_FIELDING_DP_FLAG FIELD_WORK_INDEX
## 1          39                     P                     M             -821
## 2          70                     P                     P              -38
## 3          86                     P                     P              -22
## 4          70                     P                     P               -8
## 5          82                     P                     P               30
## 6          75                     P                     P               26
##   PITCH_WORK_INDEX
## 1            -4835
## 2             -954
## 3            -1062
## 4             -922
## 5             -849
## 6             -749
```

```
## Defensive Fieldwork Granular
model_4_df_defensive_field_granular <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_PITCHING_SO_FLAG, TEAM_FIELDING_DP_FLAG,
         TEAM_FIELDING_DP, TEAM_FIELDING_E)

## Defensive Pitchwork Granular
model_4_df_defensive_pitch_granular<-moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_PITCHING_SO_FLAG, TEAM_FIELDING_DP_FLAG,
         TEAM_PITCHING_SO,TEAM_PITCHING_BB,TEAM_PITCHING_H)
```

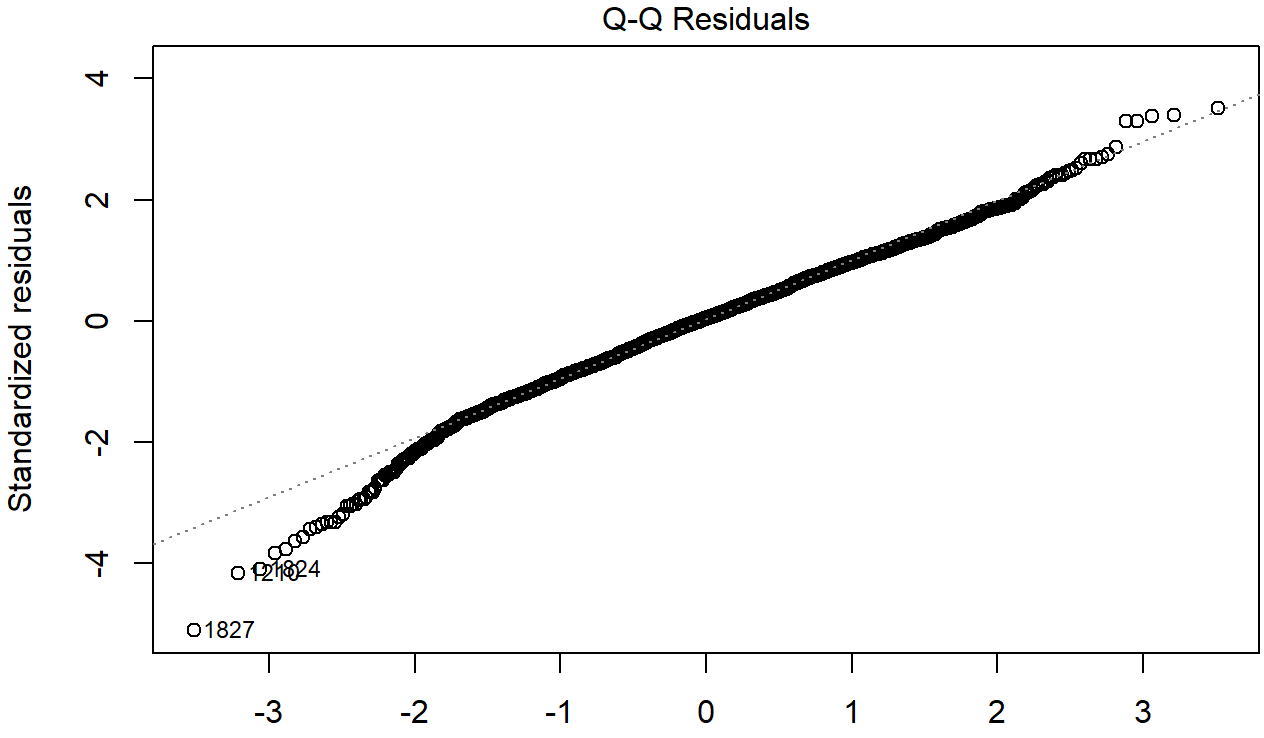
```
## Running Model with no change
model1 <- lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +TEAM_BATTING_2B + TEAM_BATTING_
HR, data=model_1_df)
print(summary(model1))
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +
##     TEAM_BATTING_2B + TEAM_BATTING_HR, data = model_1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -71.390  -8.854   0.603   9.632  49.361
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     18.031397   3.188915   5.654 1.76e-08 ***
## TEAM_BATTING_1B  0.032043   0.003092  10.364  < 2e-16 ***
## TEAM_BATTING_3B  0.157431   0.015193  10.362  < 2e-16 ***
## TEAM_BATTING_2B  0.034017   0.007679   4.430 9.89e-06 ***
## TEAM_BATTING_HR  0.114952   0.007682  14.964  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.04 on 2269 degrees of freedom
## Multiple R-squared:  0.2039, Adjusted R-squared:  0.2025
## F-statistic: 145.3 on 4 and 2269 DF,  p-value: < 2.2e-16
```

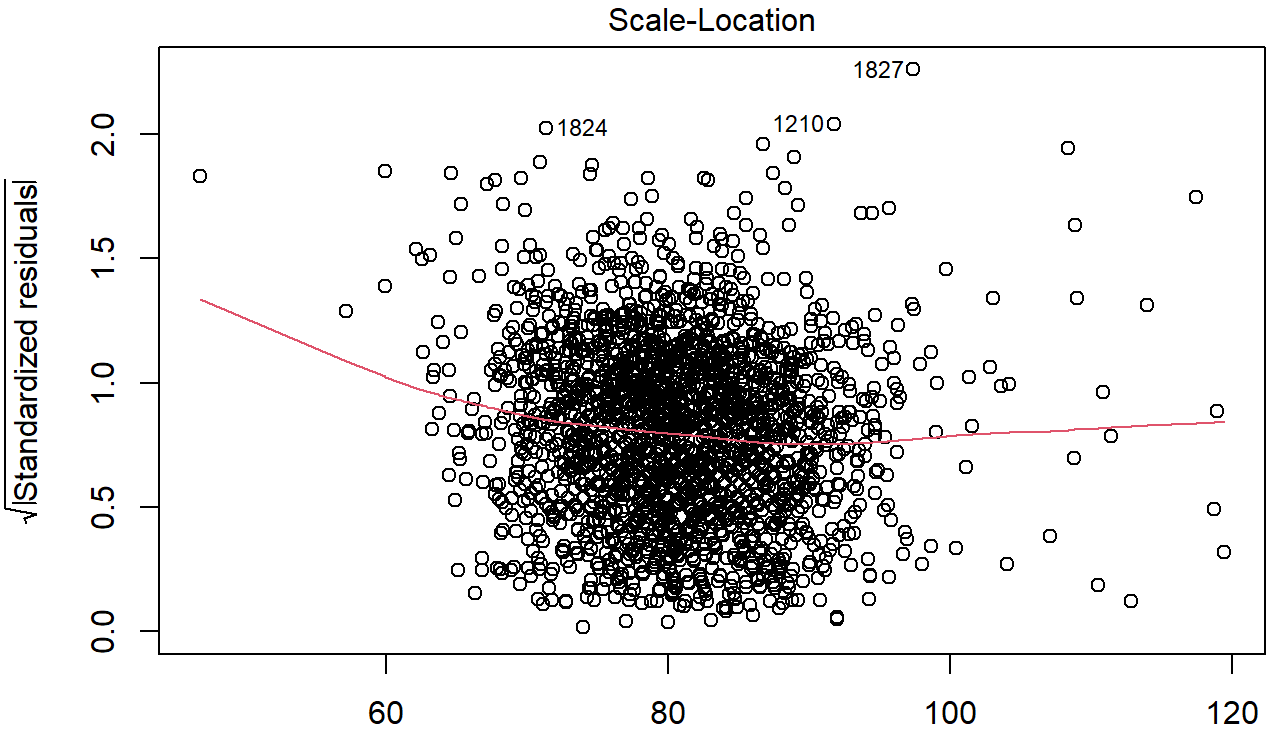
```
plot(model1)
```

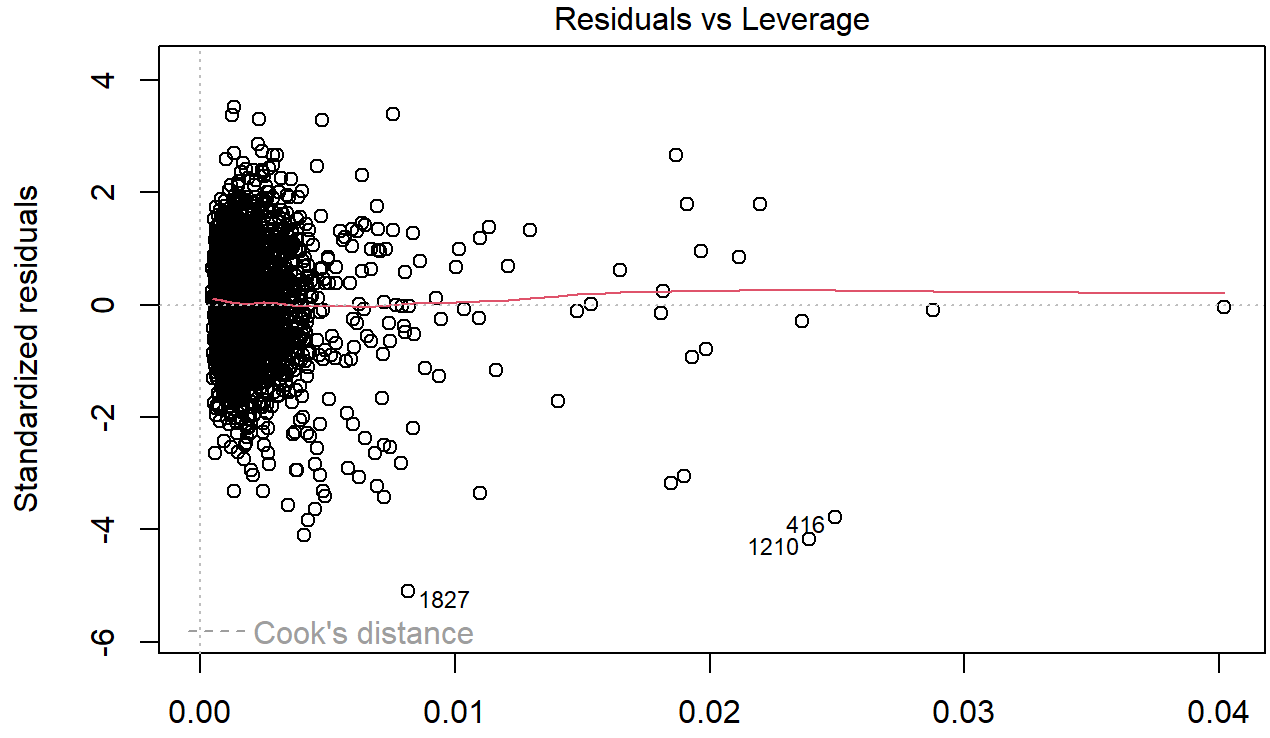


m(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE



m(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE


Scale-Location
1827
1824
1210
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE

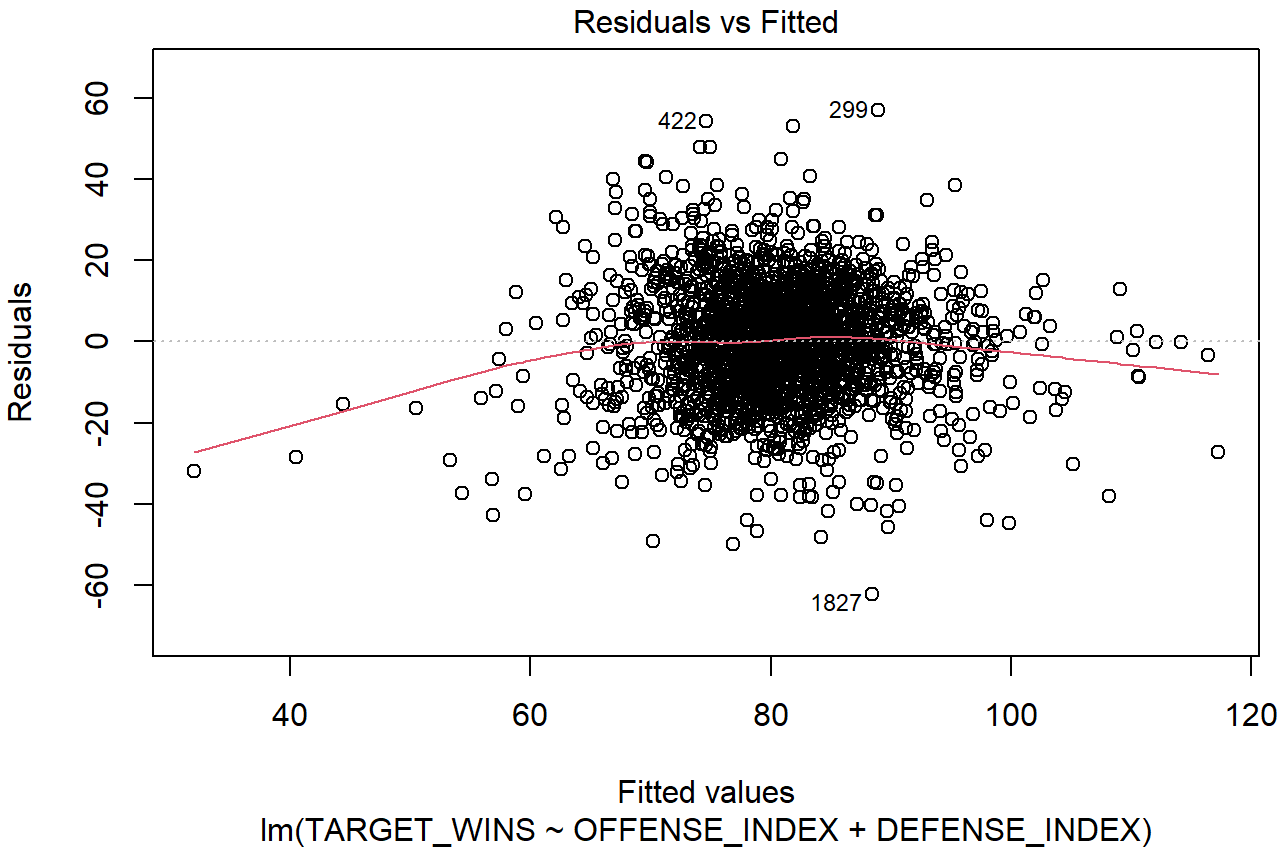

Residuals vs Leverage
Standardized residuals
-6
-4
-2
0
2
4
416
1210
1827
Cook's distance
0.00
0.01
0.02
0.03
0.04
Leverage
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE

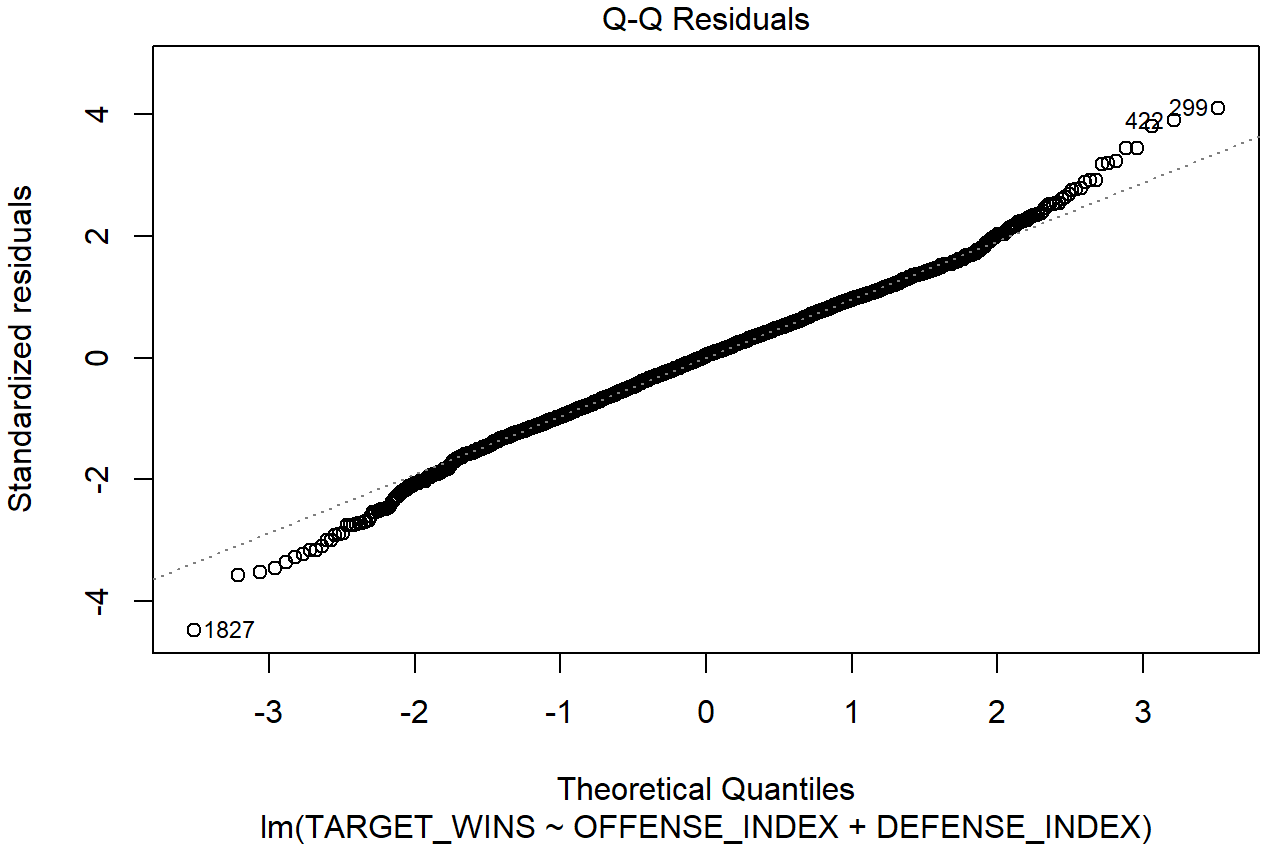
```
#### Overall adjusted r^2 is 20%, with 3b having the highest coefficient. All p values show s
tatistical significance. Lowest coeffeicsnets ate 1B and 2B, may want ot remove and adjust.

##BELOW NOT WORTH IT
# No 1b and 2b; Adjusted R2 goes down, but coefficients increase a bit for 3B
#model1_3plus <- lm(TARGET_WINS ~ TEAM_BATTING_3B + TEAM_BATTING_HR, data=model_1_df)
#print(summary(model1_3plus))
#plot(model1_3plus)
```

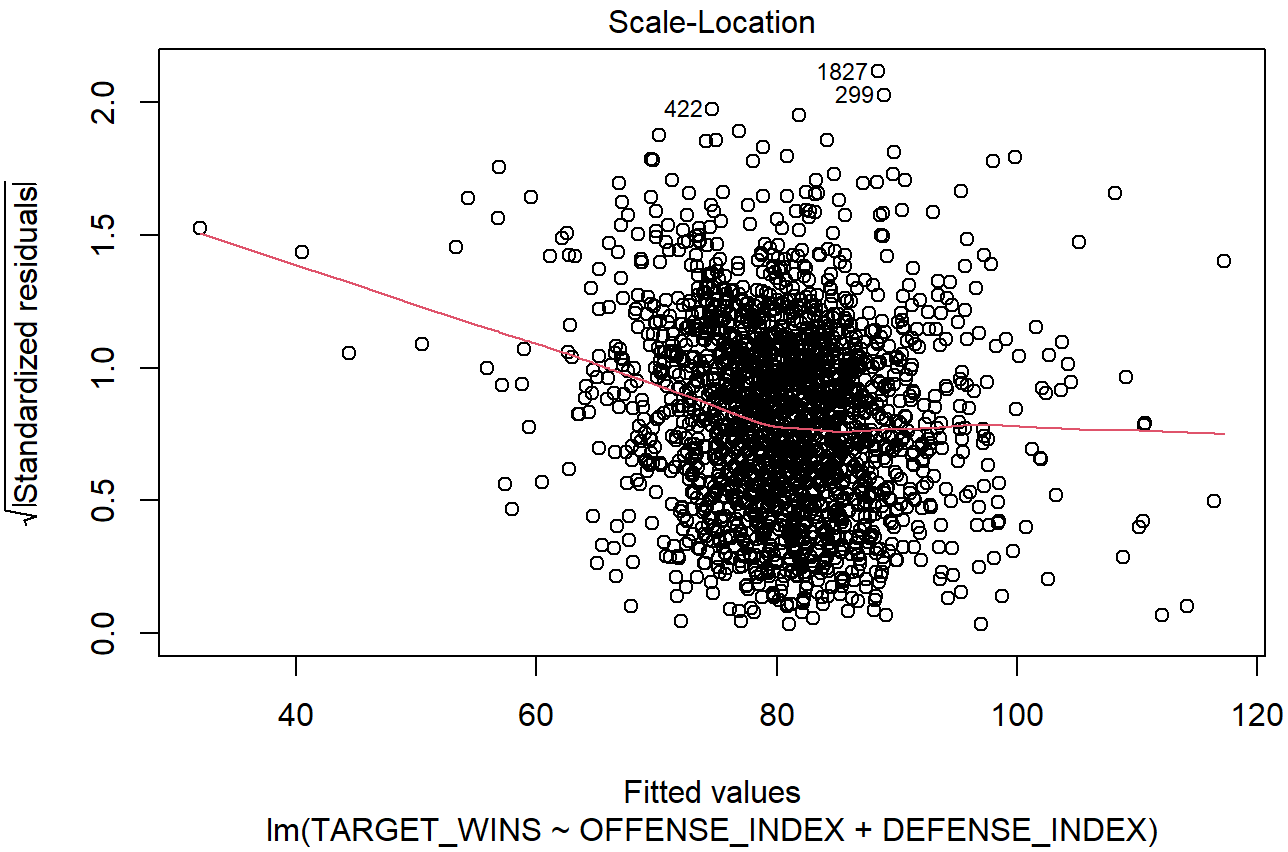
```
## Custom Agg work (R^2 is ~21%)
model2 <- lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX, data=model_2_dfa)
print(summary(model2))
```

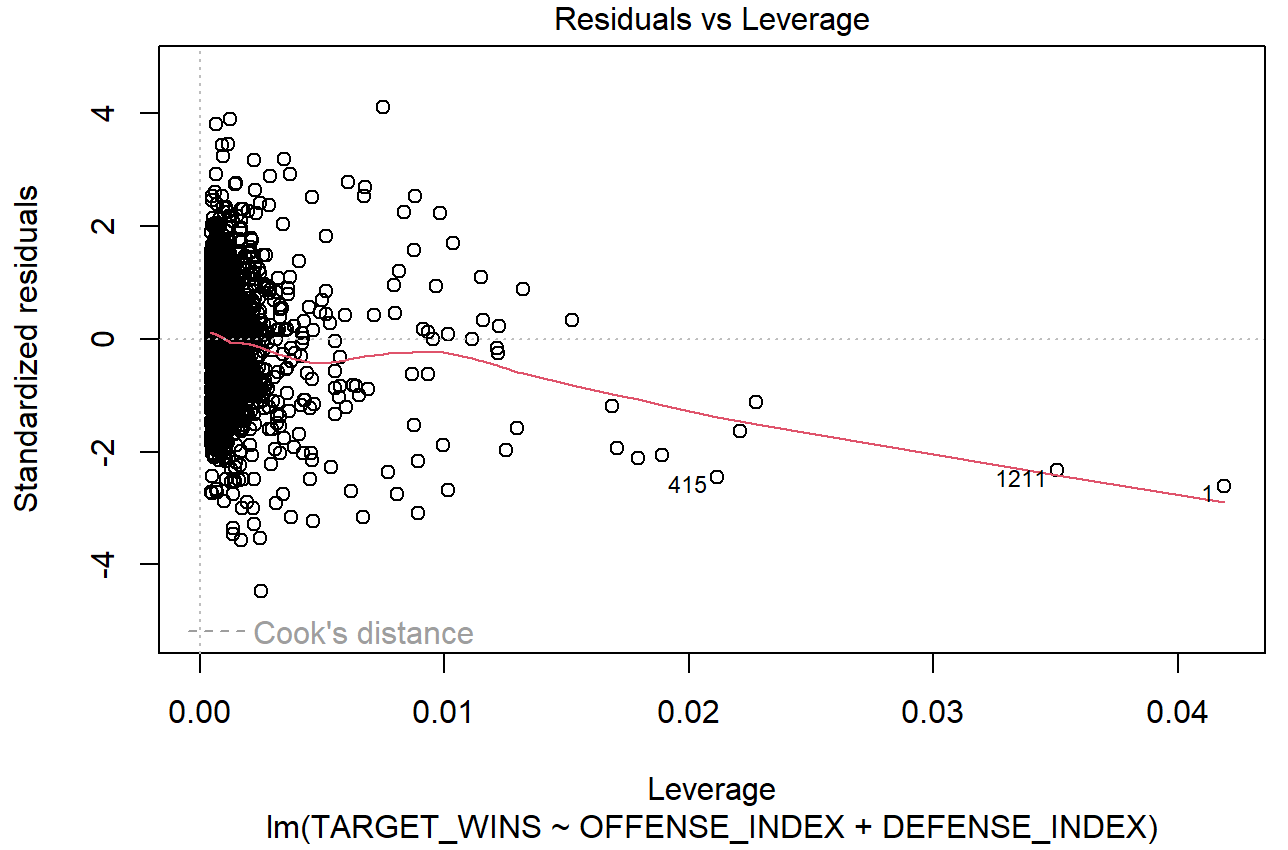
```
##
## Call:
## lm(formula = TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX, data = model_2_dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.424  -8.976   0.481   9.035  57.093
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0615756  3.0863725   1.640  0.10115
## OFFENSE_INDEX 0.0353272  0.0014213  24.855  < 2e-16 ***
## DEFENSE_INDEX 0.0025885  0.0007198   3.596  0.00033 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.95 on 2271 degrees of freedom
## Multiple R-squared:  0.214,  Adjusted R-squared:  0.2133
## F-statistic: 309.2 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)
422
299
1827
-60
-40
-20
0
20
40
60
40
60
80
100
120


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)
422
299
1827
-4
-2
0
2
4
-3
-2
-1
0
1
2
3

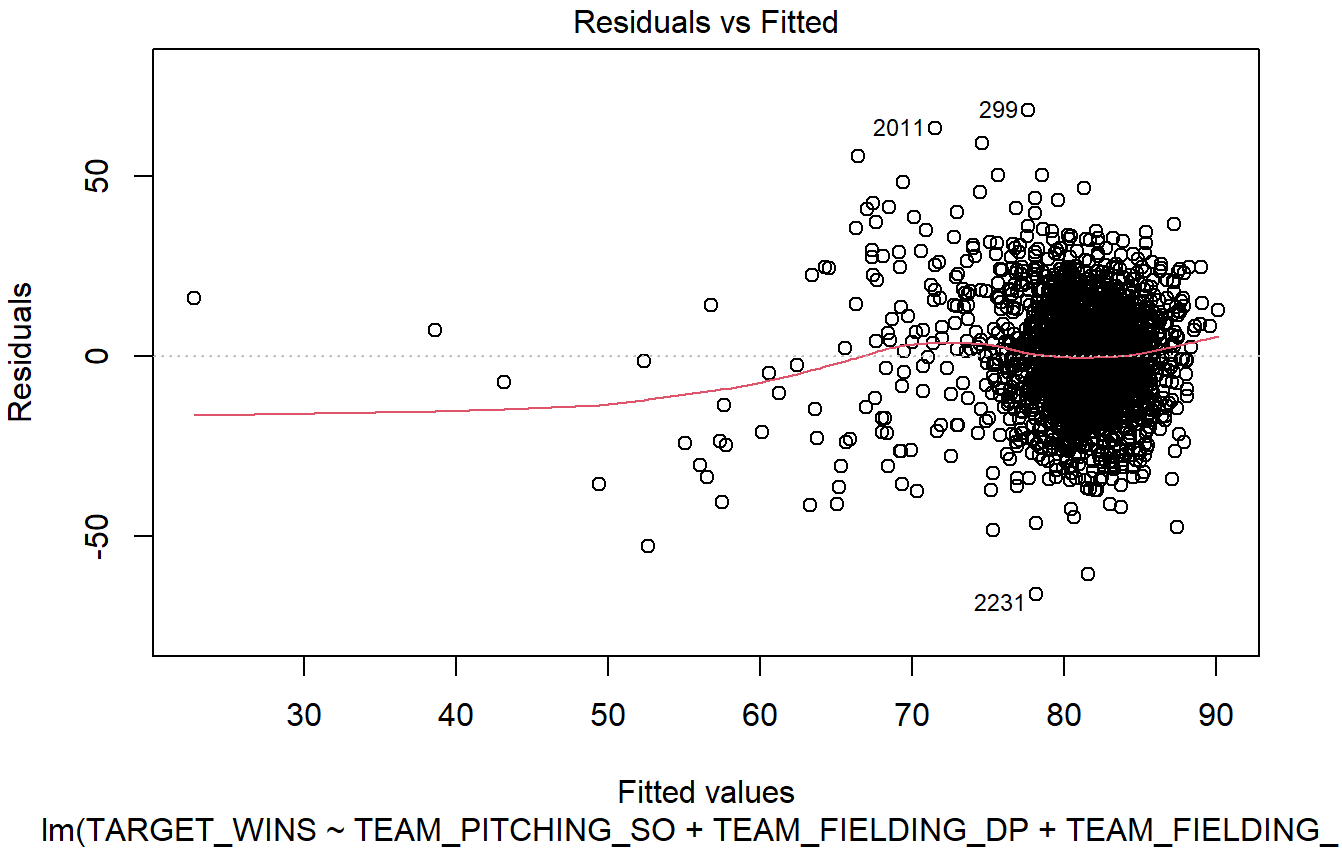

Scale-Location
1827
299
422
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)

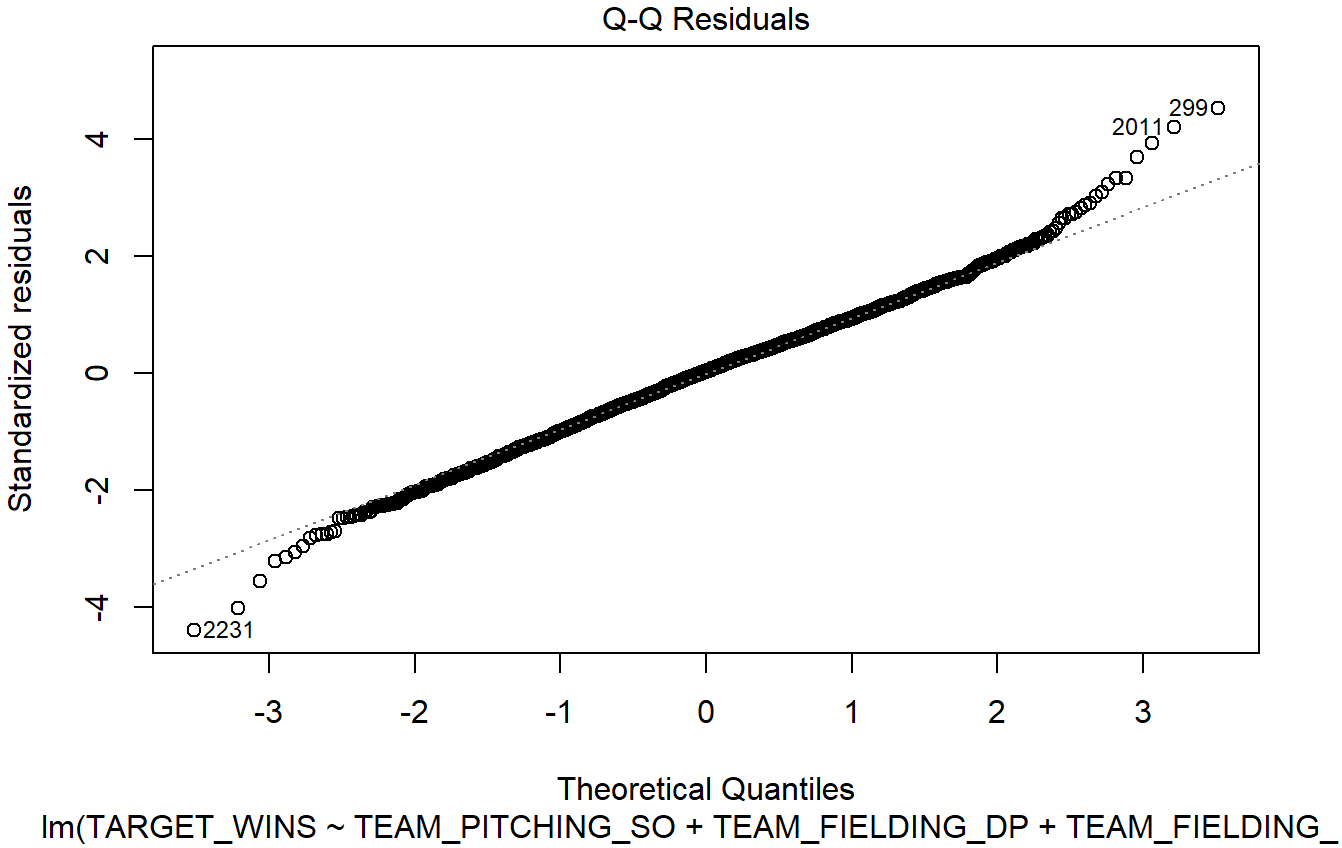

Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
415
1211
1
Cook's distance
0.00
0.01
0.02
0.03
0.04
Leverage
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)

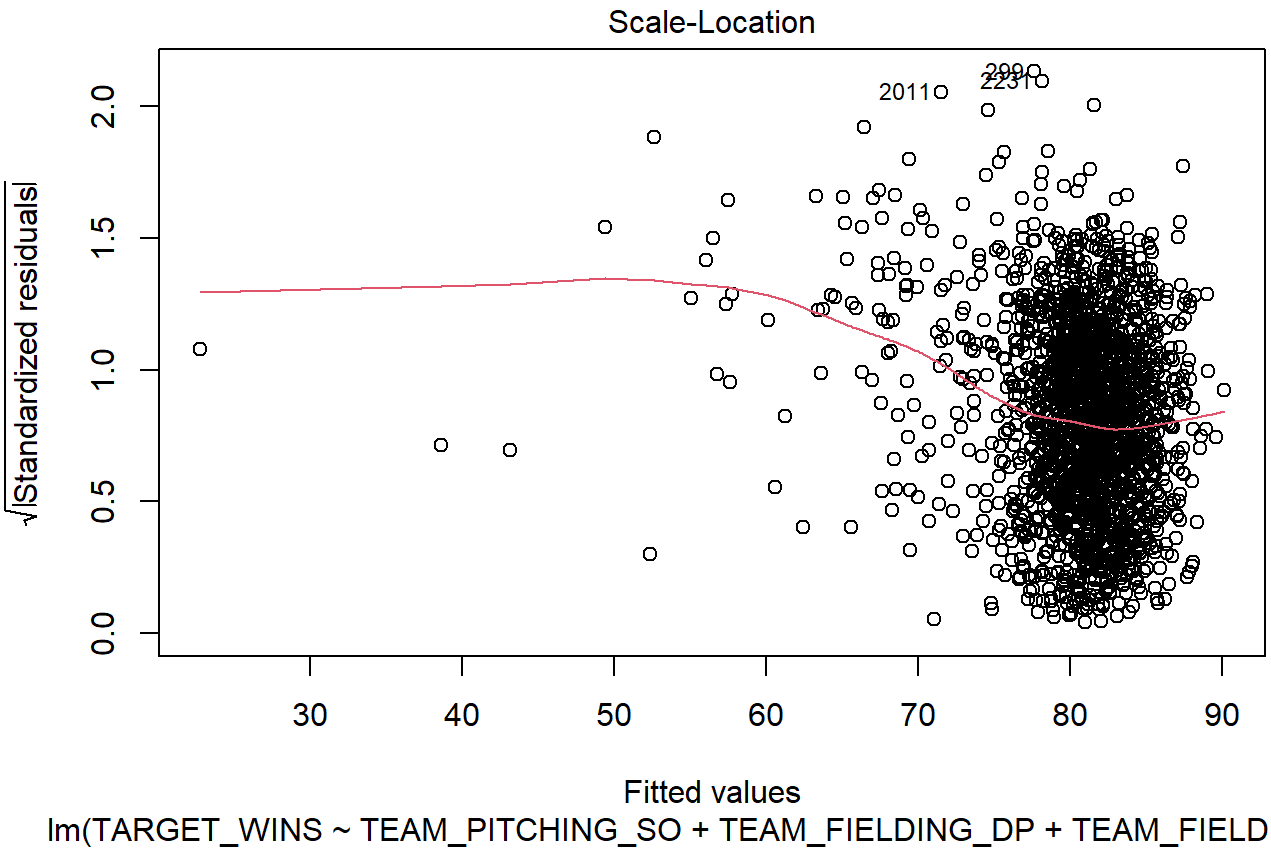
```
## Granular work  (R^2 is ~7%); All coefficients are negative. This model instance may need a transform to see impact. Low F scroe
model2_def_gran <- lm(TARGET_WINS ~ TEAM_PITCHING_SO+ TEAM_FIELDING_DP+ TEAM_FIELDING_E, data =model_2_df_granular_defense)
print(summary(model2_def_gran))
```

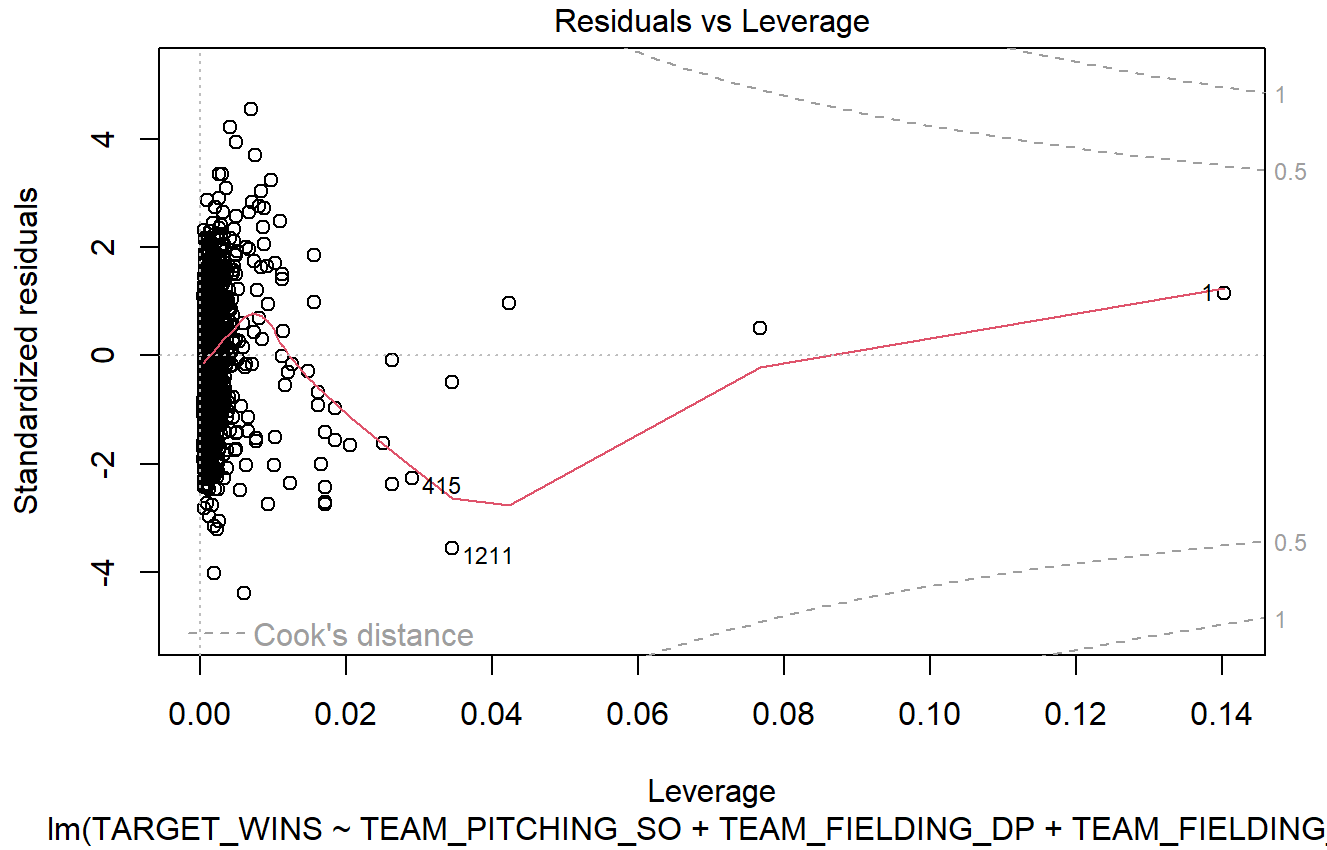
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_FIELDING_DP +
##     TEAM_FIELDING_E, data = model_2_df_granular_defense)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.122  -9.705   0.576   9.630  68.399
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      106.722892   2.286944  46.666  < 2e-16 ***
## TEAM_PITCHING_SO  -0.007813   0.001130  -6.914 6.12e-12 ***
## TEAM_FIELDING_DP  -0.102159   0.013076  -7.813 8.47e-15 ***
## TEAM_FIELDING_E   -0.021707   0.001726 -12.574  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.11 on 2270 degrees of freedom
## Multiple R-squared:  0.07804,    Adjusted R-squared:  0.07682
## F-statistic: 64.05 on 3 and 2270 DF,  p-value: < 2.2e-16
```

```
plot(model2_def_gran)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_FIELDING_DP + TEAM_FIELDING_
2011
299
2231
50
0
-50
30
40
50
60
70
80
90


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_FIELDING_DP + TEAM_FIELDING_
299
2011
2231
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

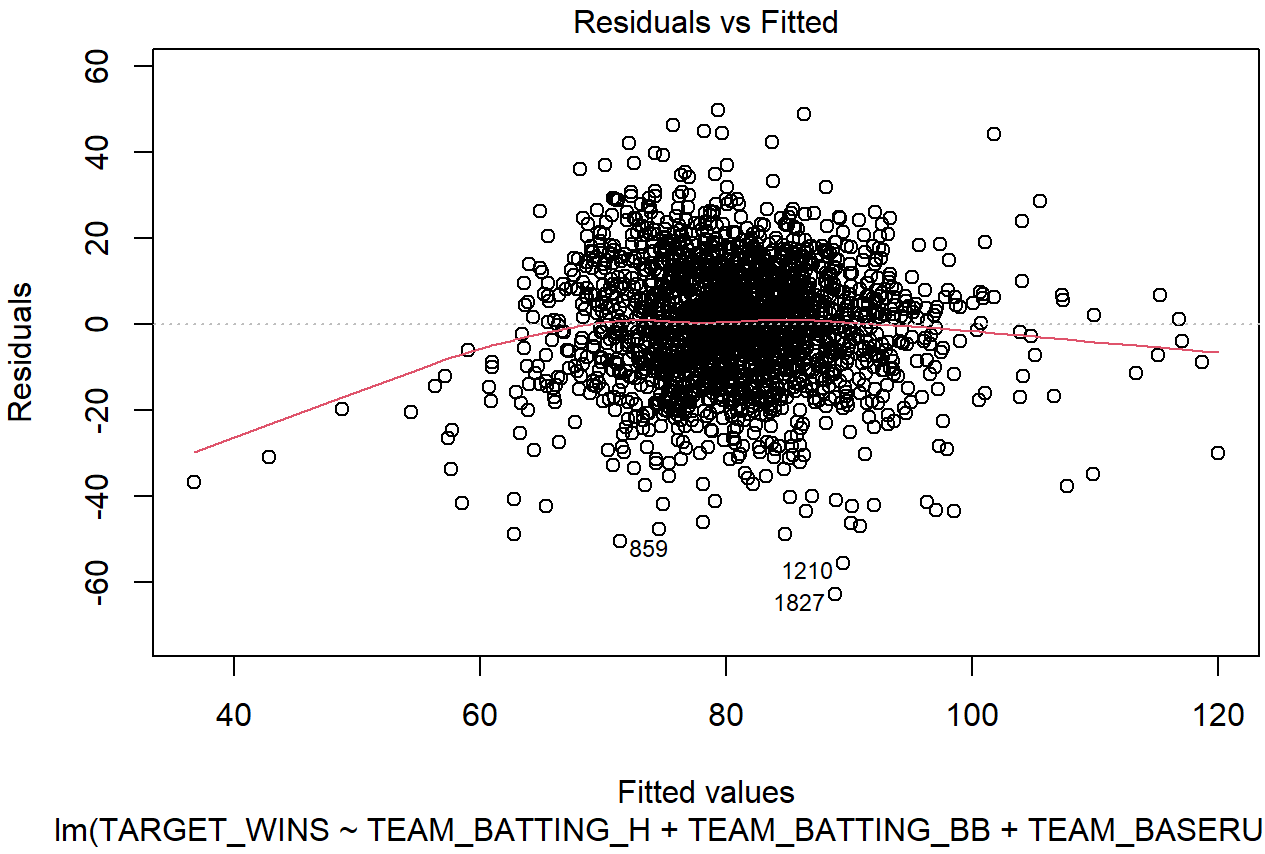

Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
2011
2231
30
40
50
60
70
80
90
Fitted values
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_FIELDING_DP + TEAM_FIELDING_

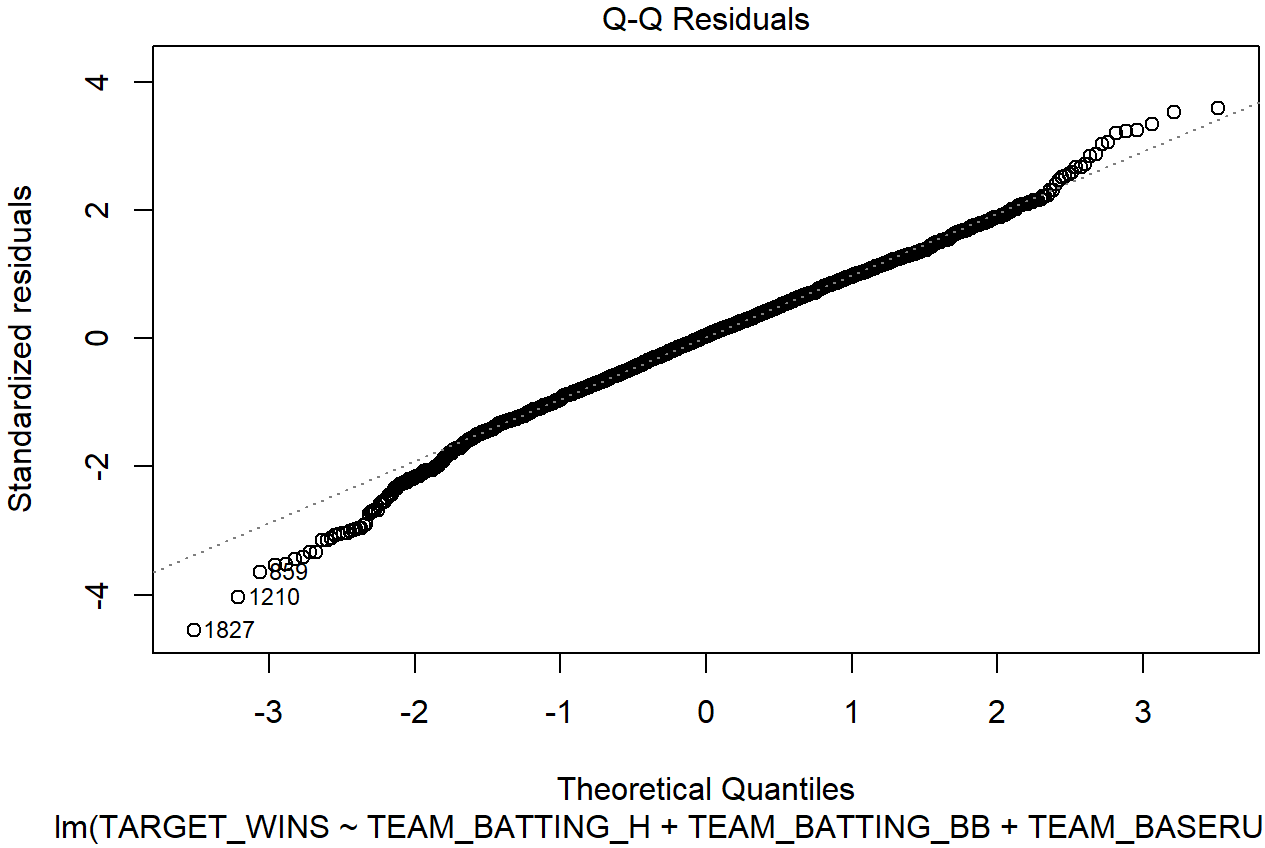

Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
1
0.5
0.5
1
1
415
1211
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Leverage
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_FIELDING_DP + TEAM_FIELDING_

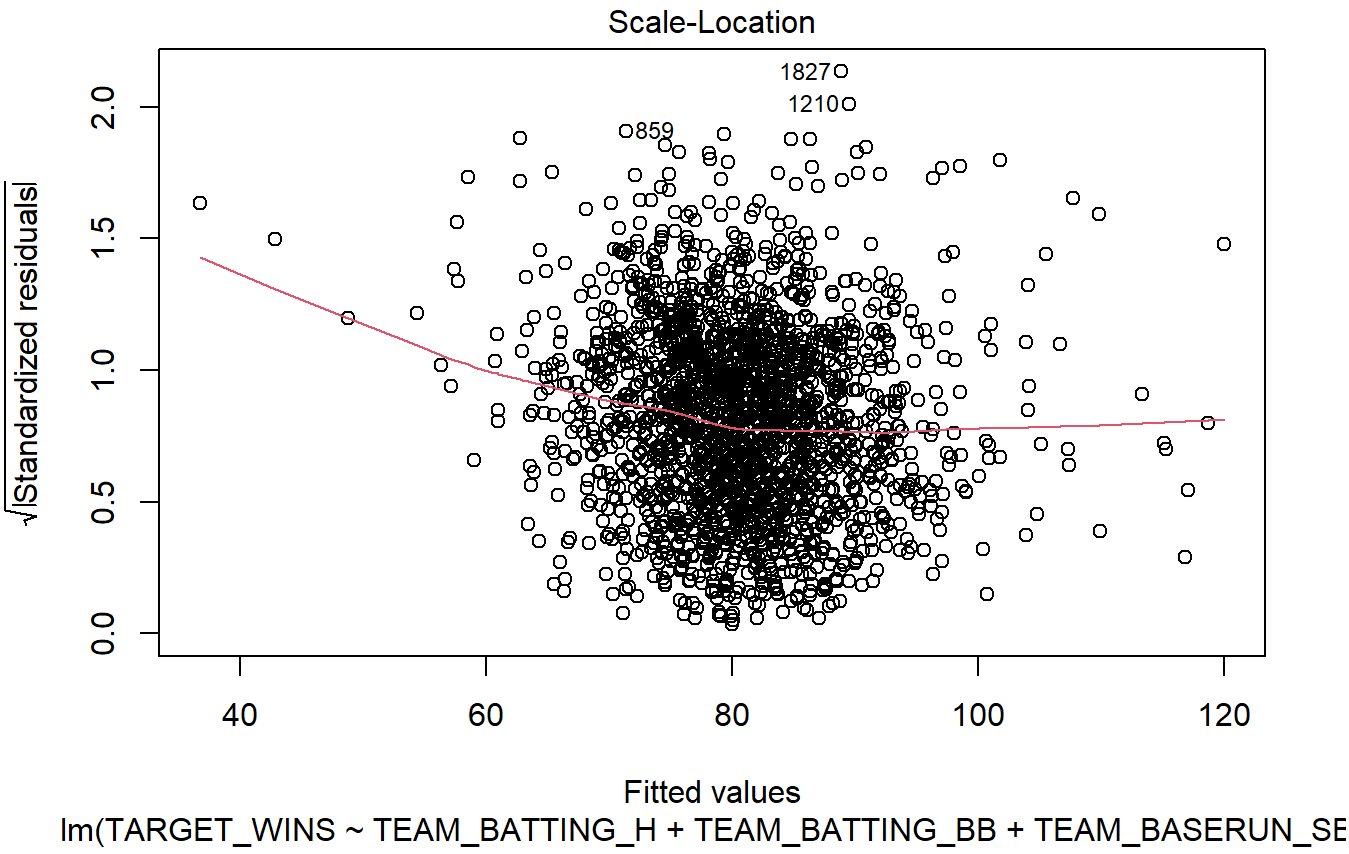
```
# R^2 is 22.5% with total hits having highest coefficient (.04) Worth keeping
model2_off_gran <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BASERUN_SB, data=
model_2_df_granualar_offense)
print(summary(model2_off_gran))
```

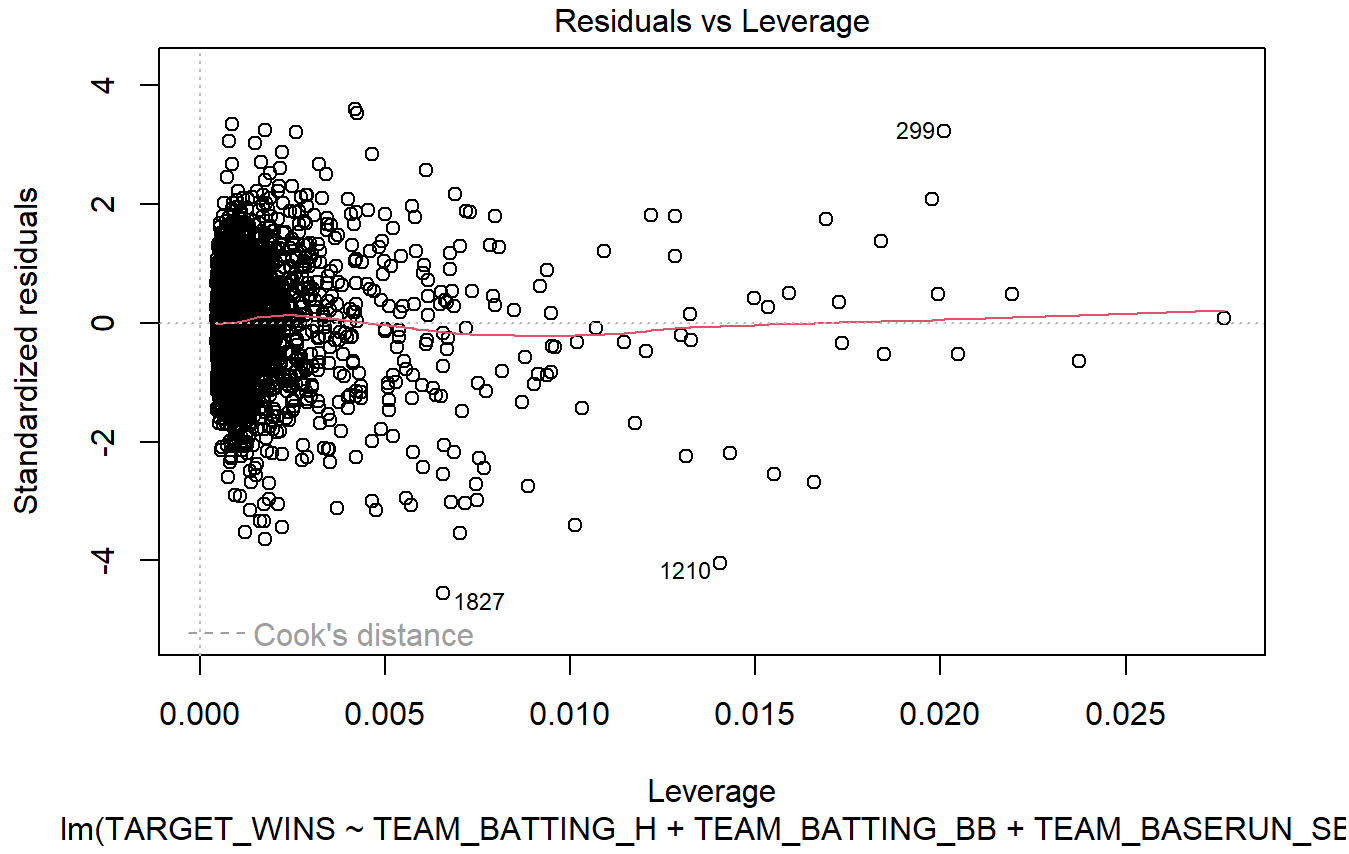
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB +
##     TEAM_BASERUN_SB, data = model_2_df_granualar_offense)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.823  -8.763   0.490   9.235  49.663
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.318103   3.299134  -0.703    0.482
## TEAM_BATTING_H   0.043789   0.002029  21.581  < 2e-16 ***
## TEAM_BATTING_BB  0.033815   0.002379  14.214  < 2e-16 ***
## TEAM_BASERUN_SB  0.014734   0.003381   4.358 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.84 on 2270 degrees of freedom
## Multiple R-squared:  0.2268, Adjusted R-squared:  0.2258
## F-statistic:   222 on 3 and 2270 DF,  p-value: < 2.2e-16
```

```
plot(model2_off_gran)
```


Residuals vs Fitted
Residuals
60
40
20
0
-20
-40
-60
859
1210
1827
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BASERUN_SE


Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
859
1210
1827
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BASERUN_SE

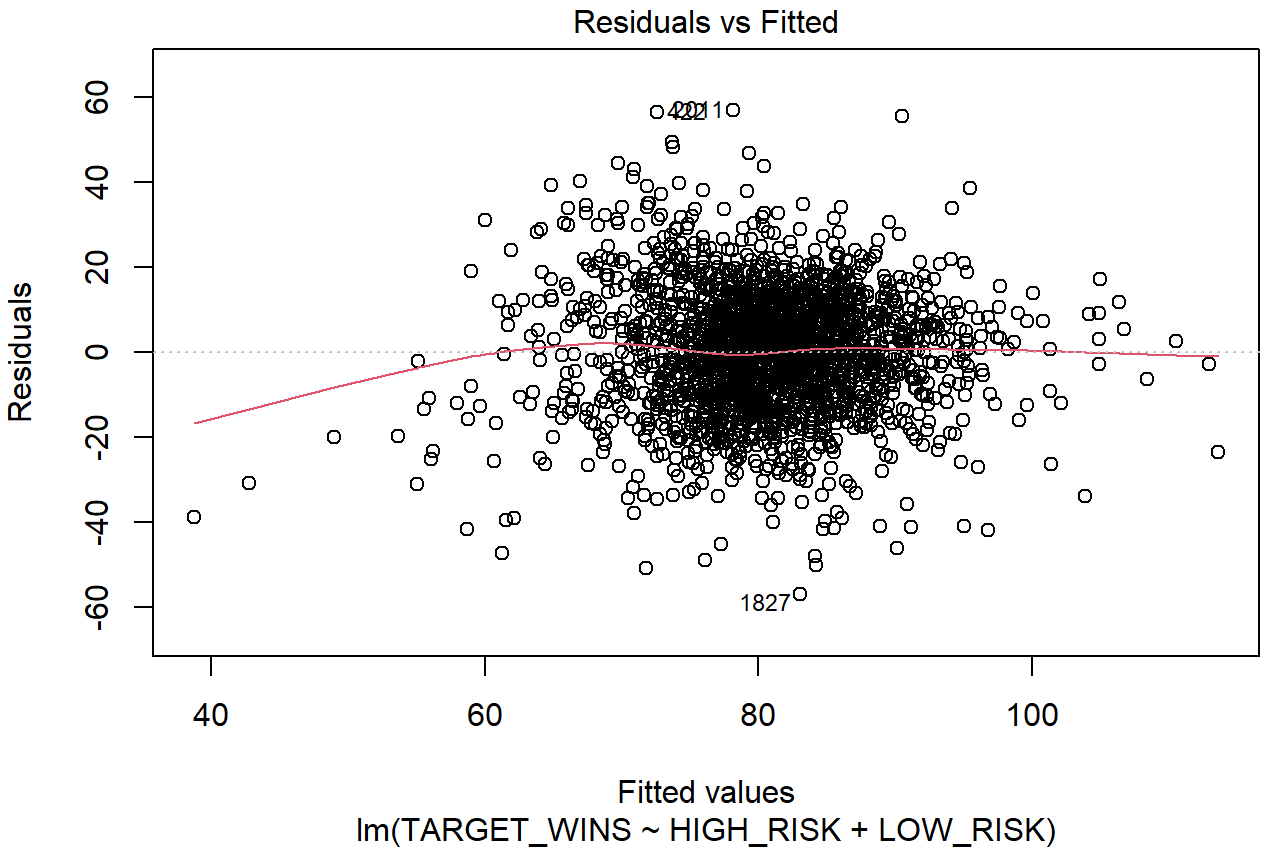

Scale-Location
1827
1210
859
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BASERUN_SE

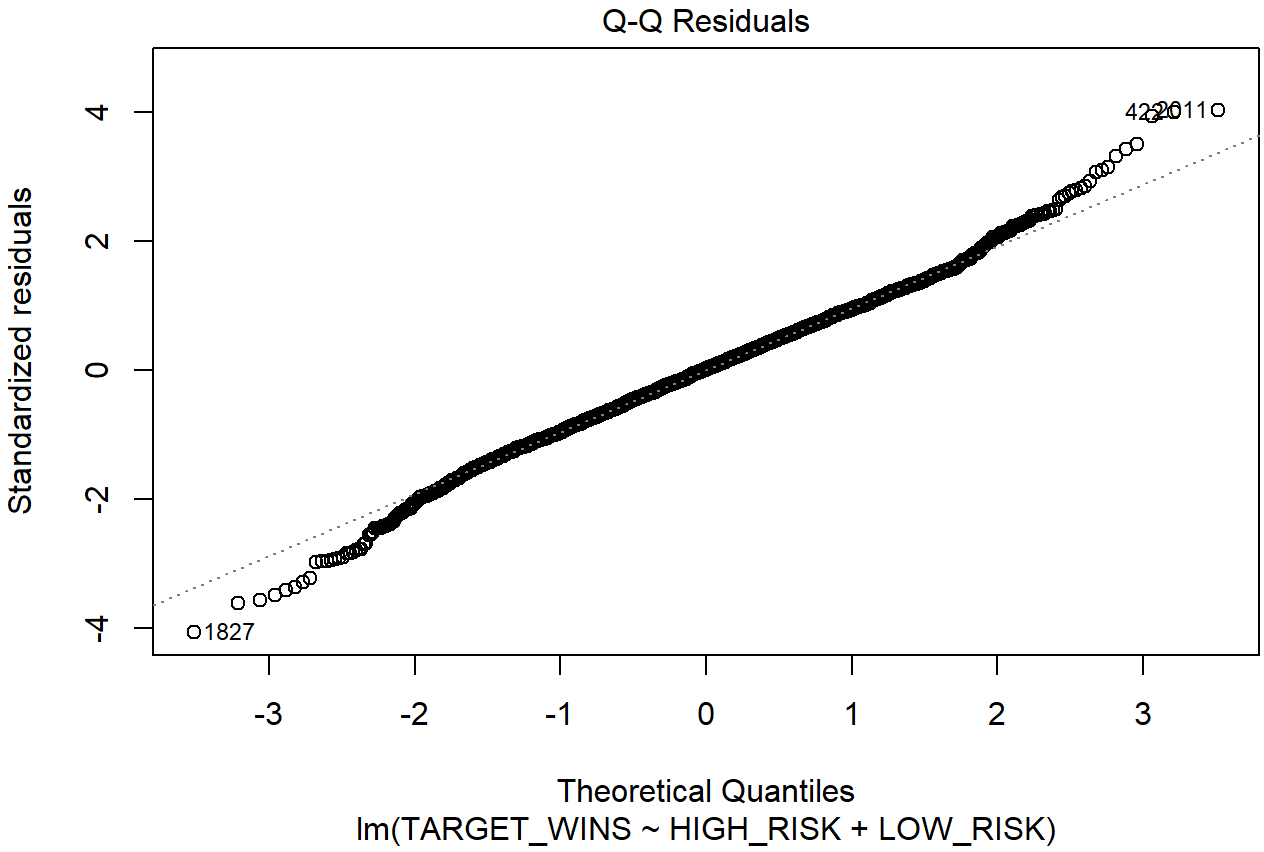

Residuals vs Leverage
299
1210
1827
Cook's distance
Standardized residuals
-4
-2
0
2
4
0.000
0.005
0.010
0.015
0.020
0.025
Leverage
lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BASERUN_SE

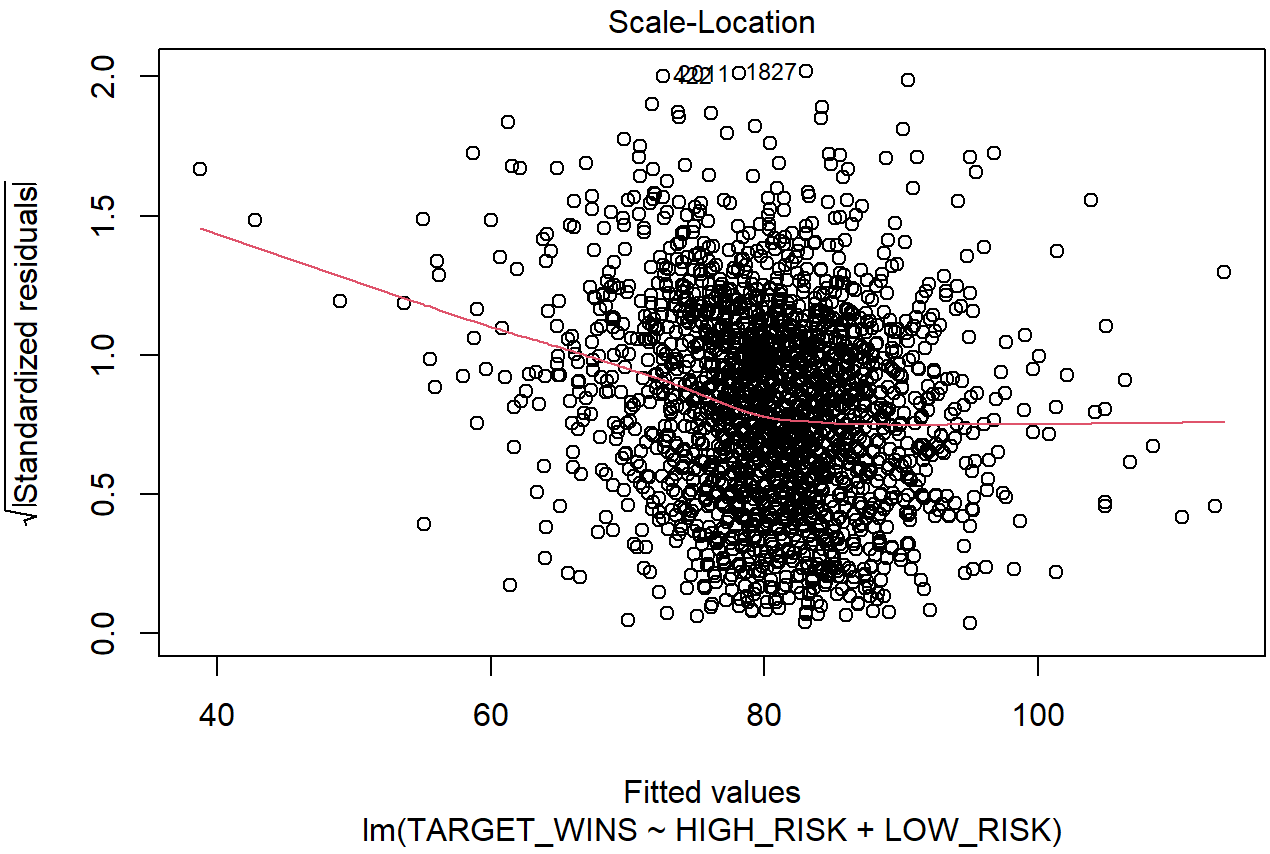
```
## Custom Agg (R^2 ~20%) Low coeff.m high f score
model3 <- lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK, data=model_3_dfa)
print(summary(model3))
```

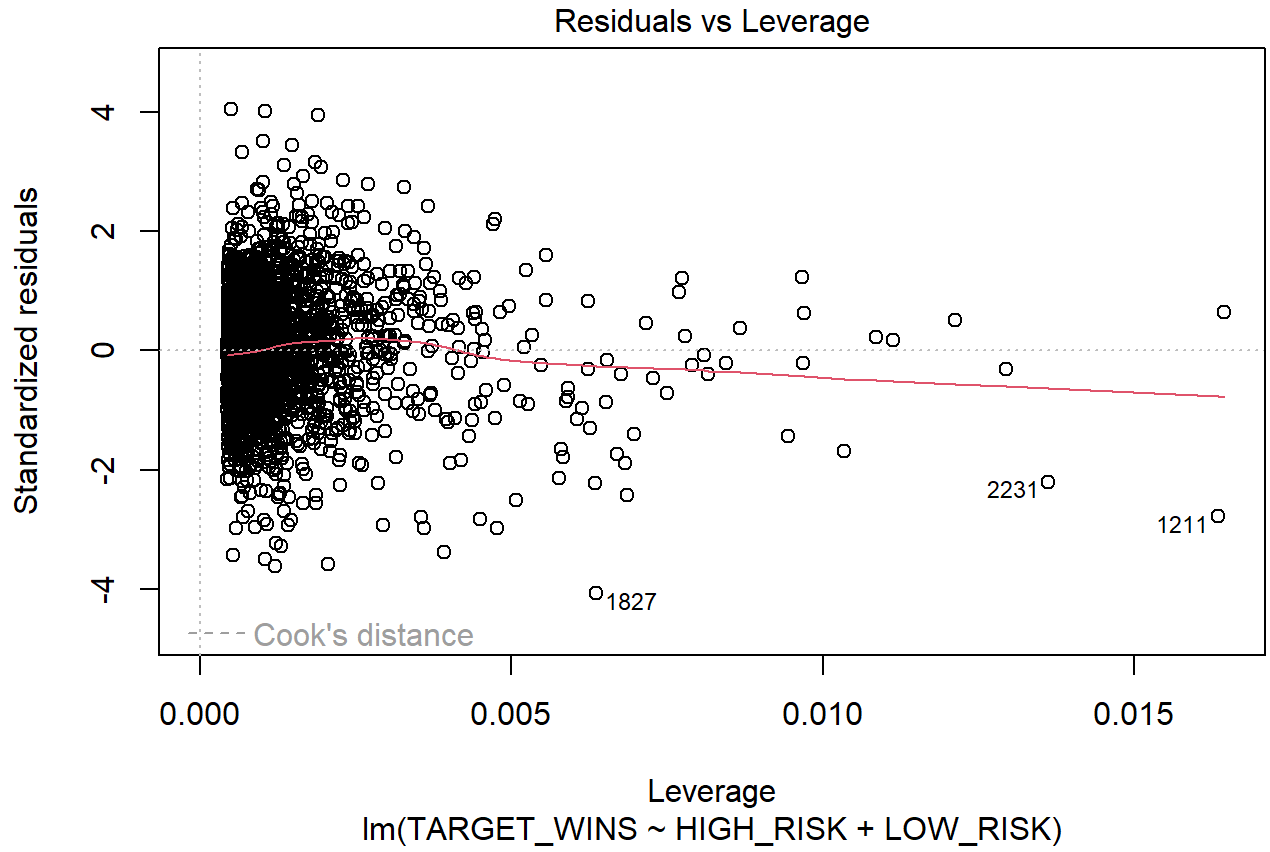
```
##
## Call:
## lm(formula = TARGET_WINS ~ HIGH_RISK + LOW_RISK, data = model_3_dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.078  -9.081   0.197   9.110  56.827
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.664325   3.207541   2.078   0.0378 *
## HIGH_RISK   0.018733   0.002900   6.459 1.29e-10 ***
## LOW_RISK    0.039120   0.002029  19.277  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.06 on 2271 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.2002
## F-statistic: 285.6 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model3)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)
1827


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)
1827

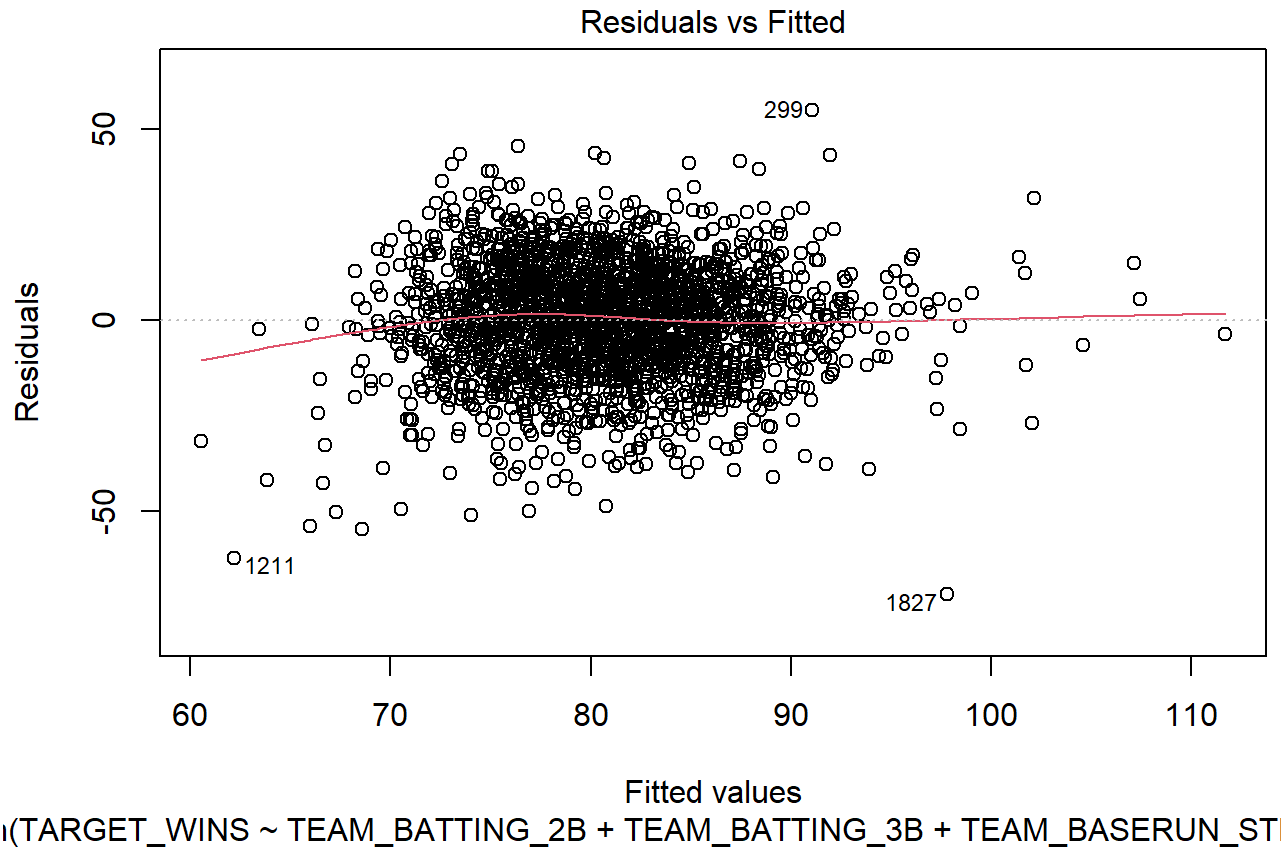

Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
40
60
80
100
Fitted values
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)


Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
0.000
0.005
0.010
0.015
Leverage
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)
Cook's distance
1827
2231
1211

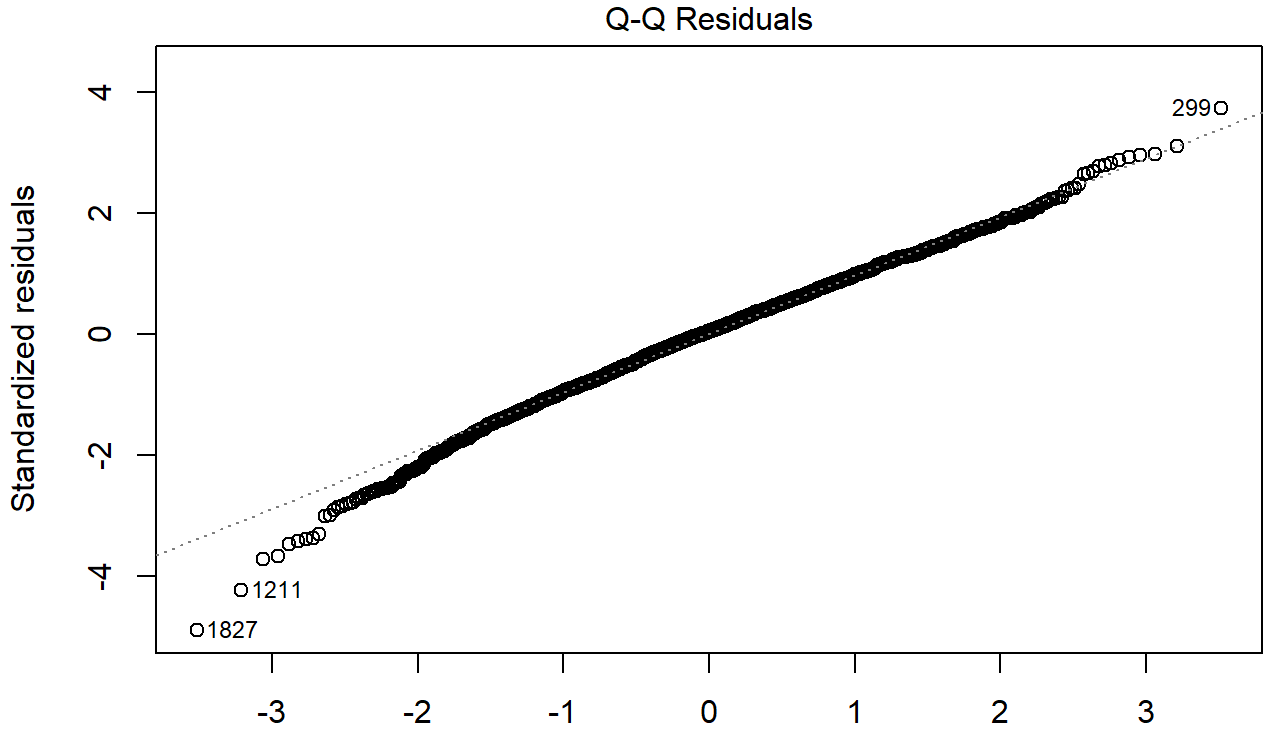
```
## R^2 is ~12%
model3_gran_high_risk <- lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B +TEAM_BASERUN_STE
AL_ATTEMPTS, data=model_3_df_high_risk_granular)
print(summary(model3_gran_high_risk))
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B +
##     TEAM_BASERUN_STEAL_ATTEMPTS, data = model_3_df_high_risk_granular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -71.784  -9.394   0.655   9.728  54.924
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 47.976139   1.857844  25.824  < 2e-16 ***
## TEAM_BATTING_2B              0.105327   0.006649  15.842  < 2e-16 ***
## TEAM_BATTING_3B              0.082740   0.011532   7.175 9.74e-13 ***
## TEAM_BASERUN_STEAL_ATTEMPTS  0.018618   0.003500   5.320 1.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.72 on 2270 degrees of freedom
## Multiple R-squared:  0.1249, Adjusted R-squared:  0.1237
## F-statistic:   108 on 3 and 2270 DF,  p-value: < 2.2e-16
```

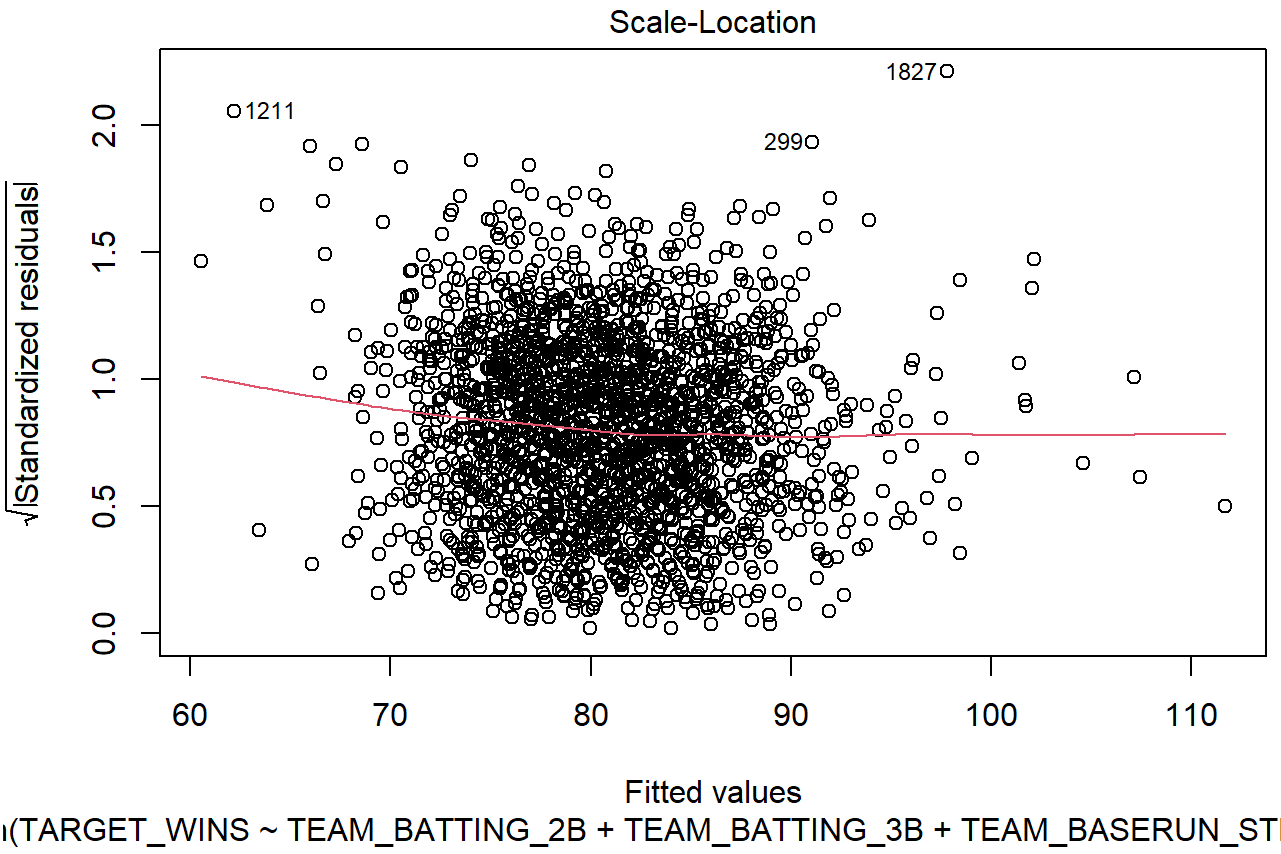
```
plot(model3_gran_high_risk)
```

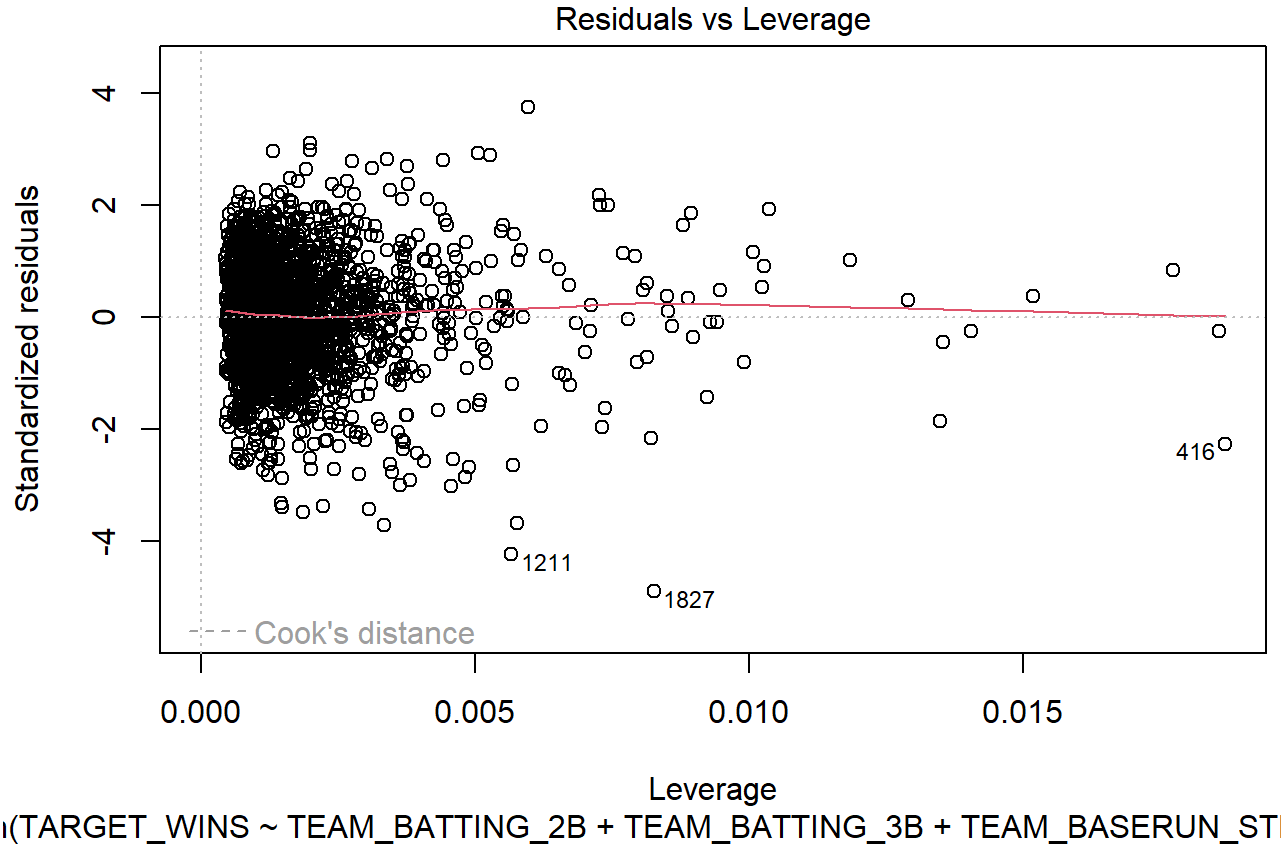


lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BASERUN_STEAL_



lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BASERUN_STEAL_

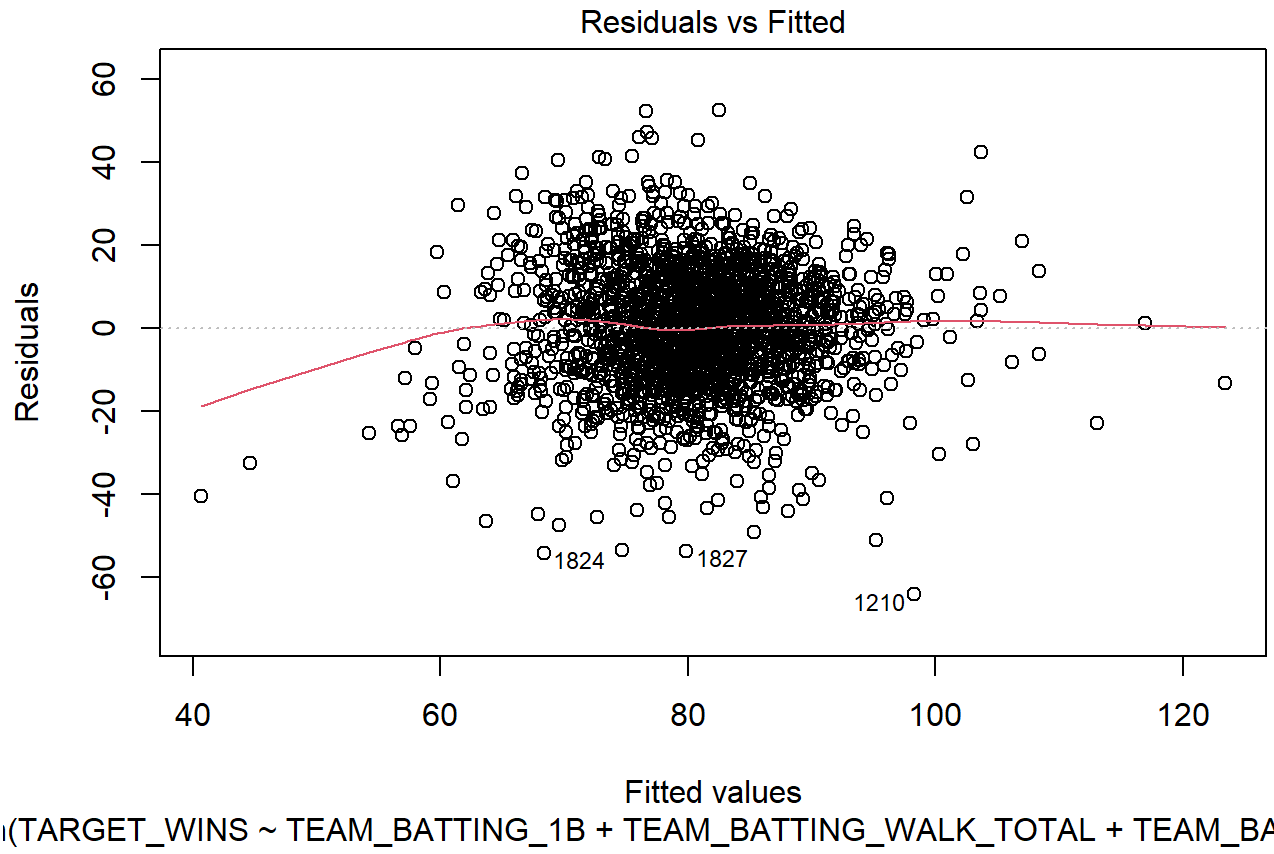

Scale-Location
1827
1211
299
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
60
70
80
90
100
110
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BASERUN_STEAL_

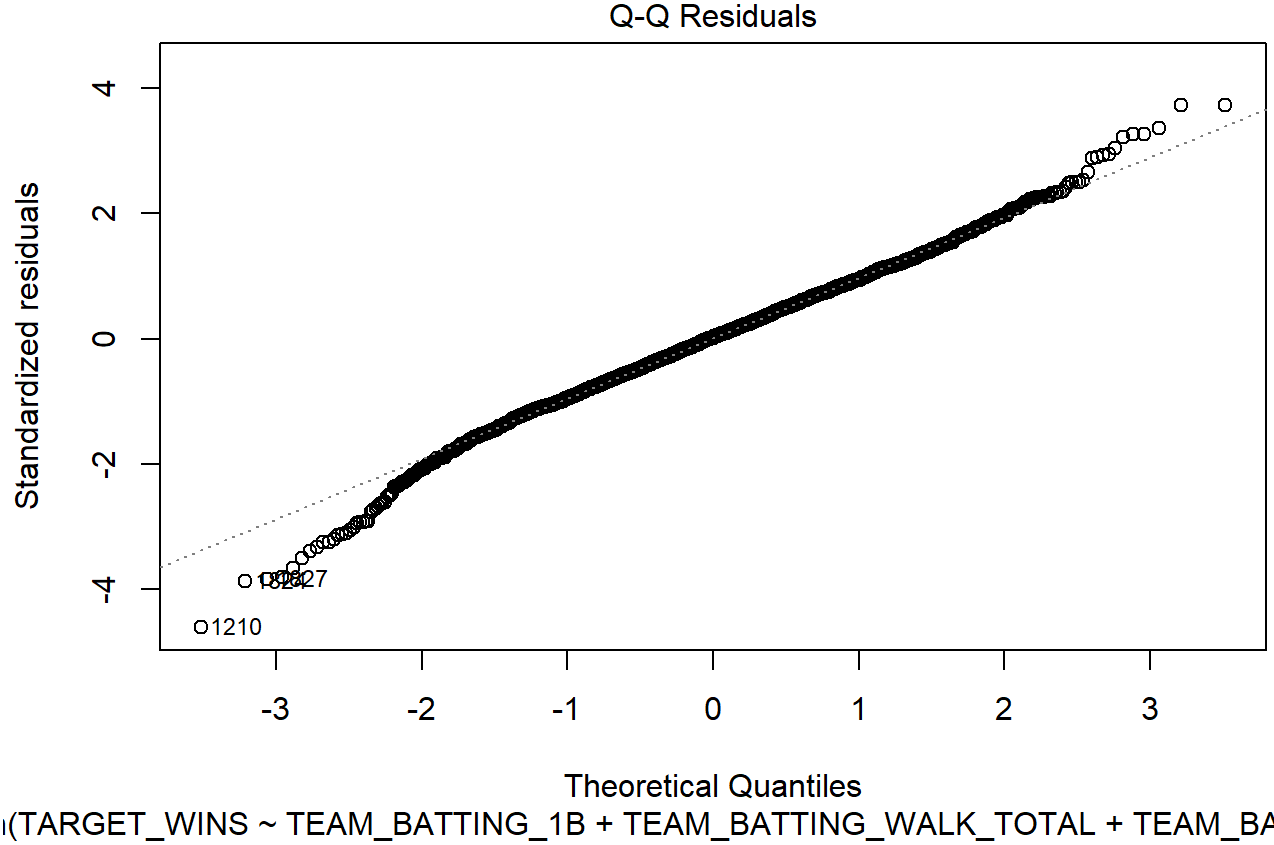

Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
416
1211
1827
Cook's distance
0.000
0.005
0.010
0.015
Leverage
lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BASERUN_STEAL_

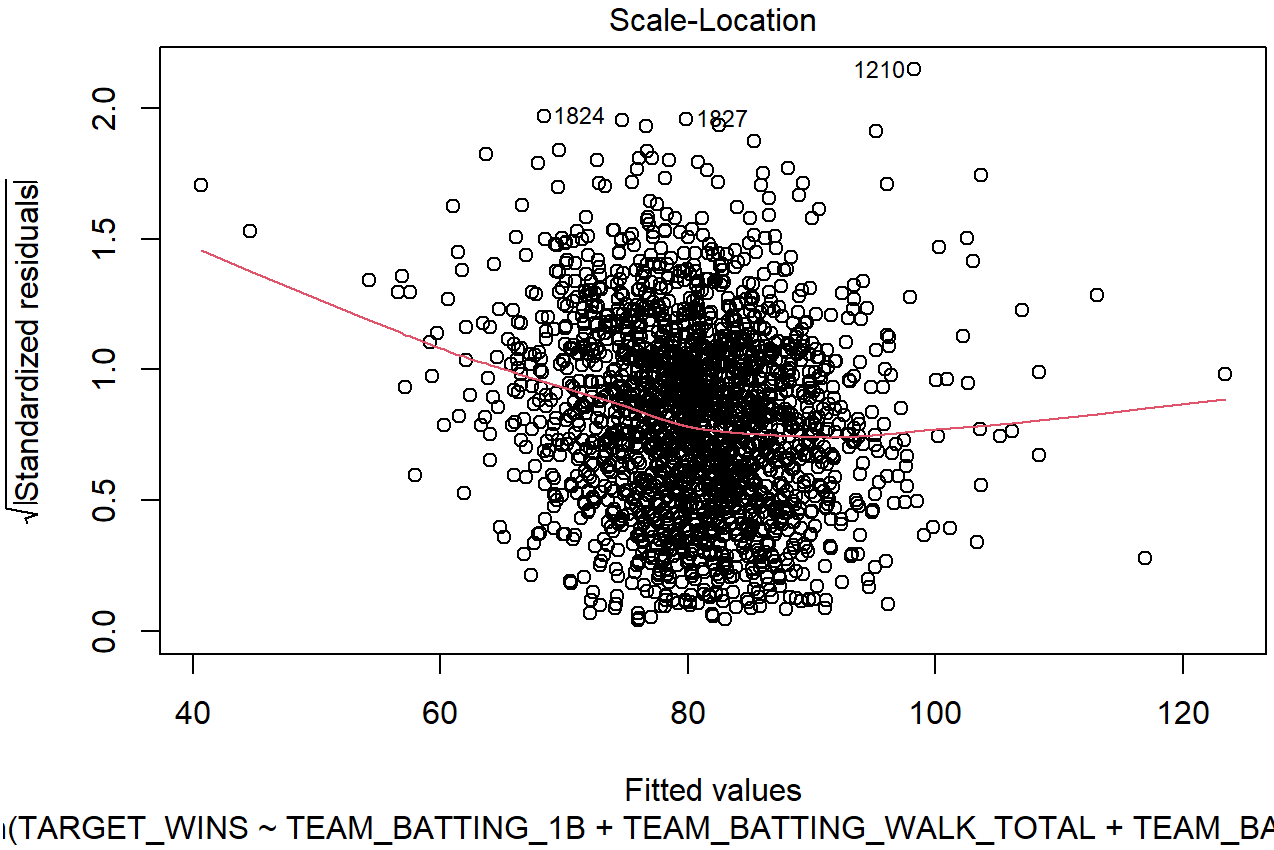
```
## R^2 is 19.8%, a ll low coefficients though, f scroe higher
model3_gran_low_risk <- lm(TARGET_WINS ~ TEAM_BATTING_1B +TEAM_BATTING_WALK_TOTAL+TEAM_BATTIN
G_HR, data=model_3_df_low_risk_granular)
print(summary(model3_gran_low_risk))
```

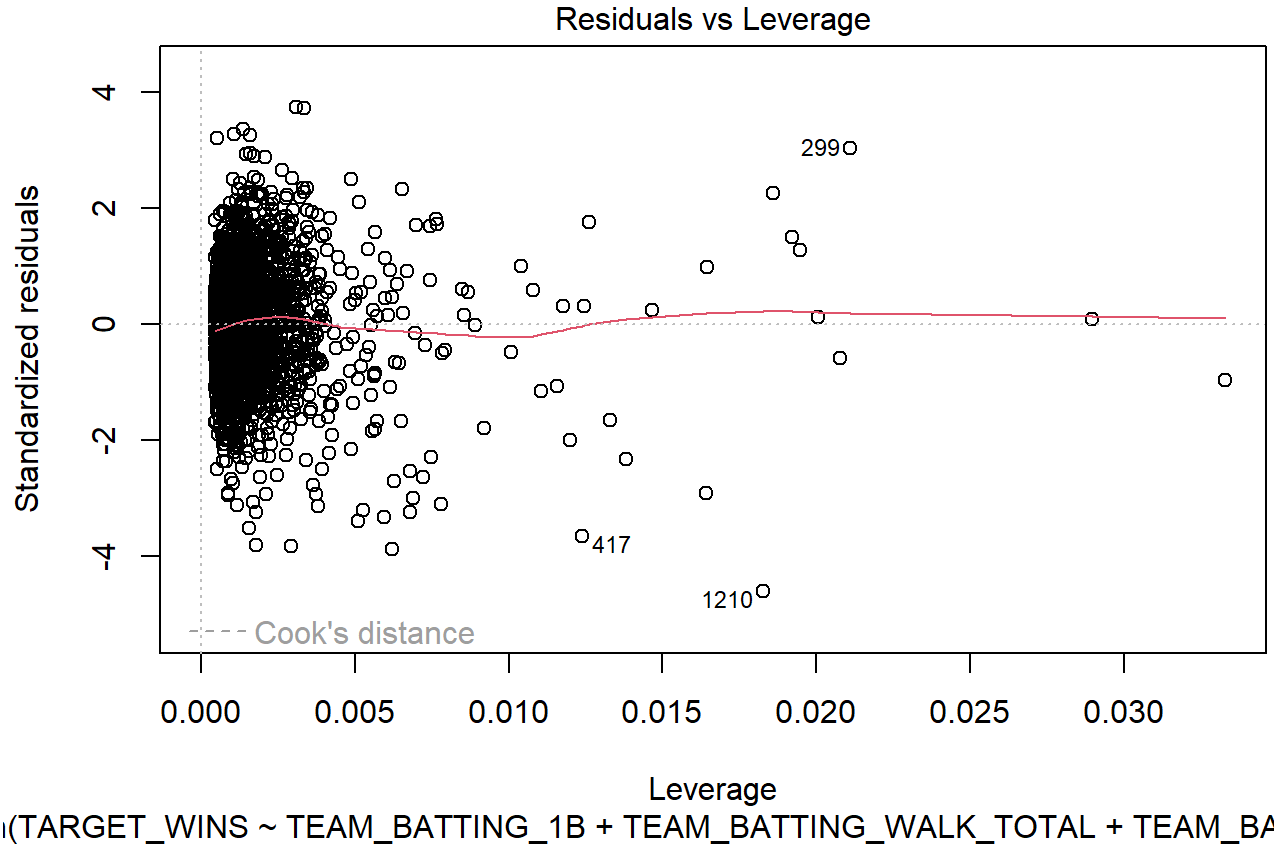
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL +
##     TEAM_BATTING_HR, data = model_3_df_low_risk_granular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.285  -8.924   0.259   9.346  52.517
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.383255   3.557273  -0.108    0.914
## TEAM_BATTING_1B          0.054282   0.002683  20.229   <2e-16 ***
## TEAM_BATTING_WALK_TOTAL  0.031993   0.002860  11.187   <2e-16 ***
## TEAM_BATTING_HR          0.067185   0.006324  10.624   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.08 on 2270 degrees of freedom
## Multiple R-squared:  0.1996, Adjusted R-squared:  0.1986
## F-statistic: 188.7 on 3 and 2270 DF,  p-value: < 2.2e-16
```

```
plot(model3_gran_low_risk)
```


Residuals vs Fitted
Residuals
60
40
20
0
-20
-40
-60
1824
1827
1210
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN


Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
1210
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN

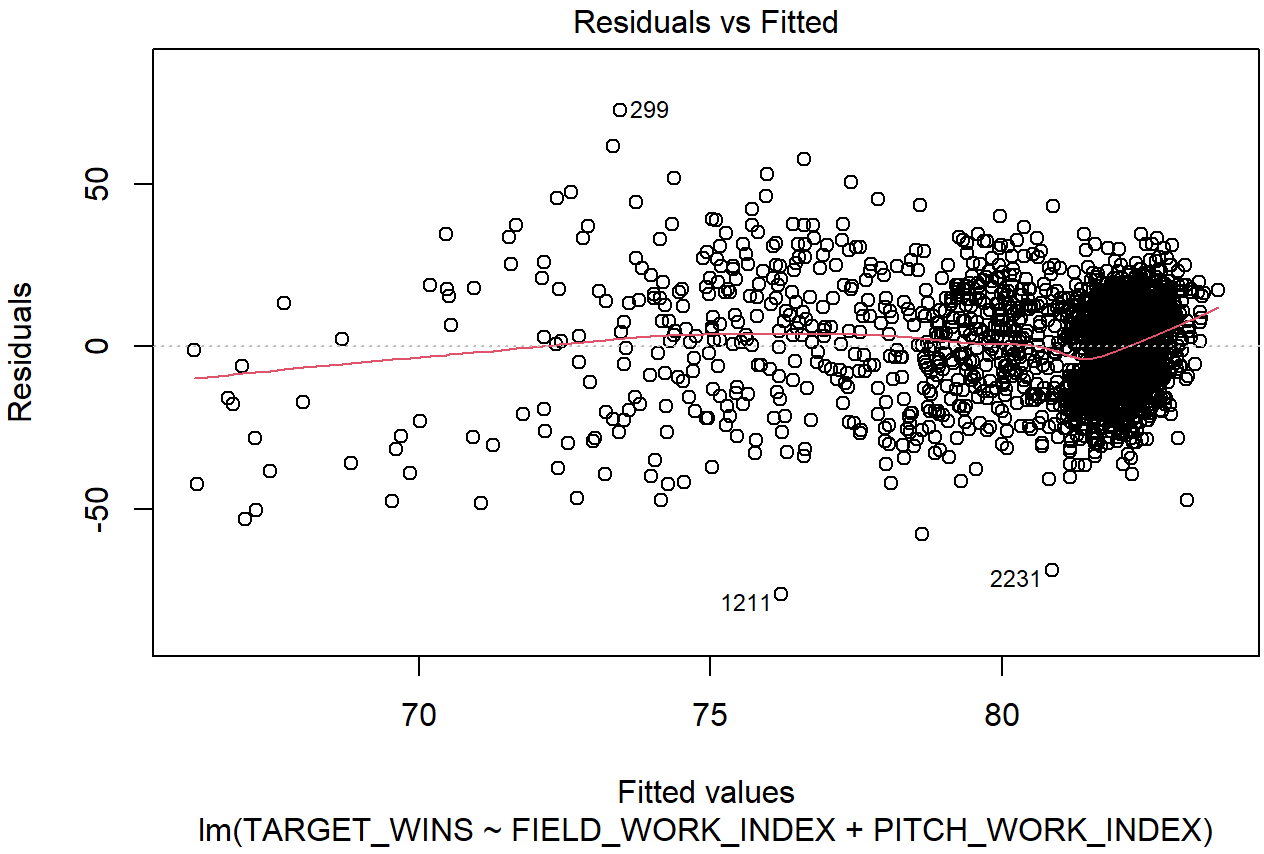

Scale-Location
1210
1824
1827
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN

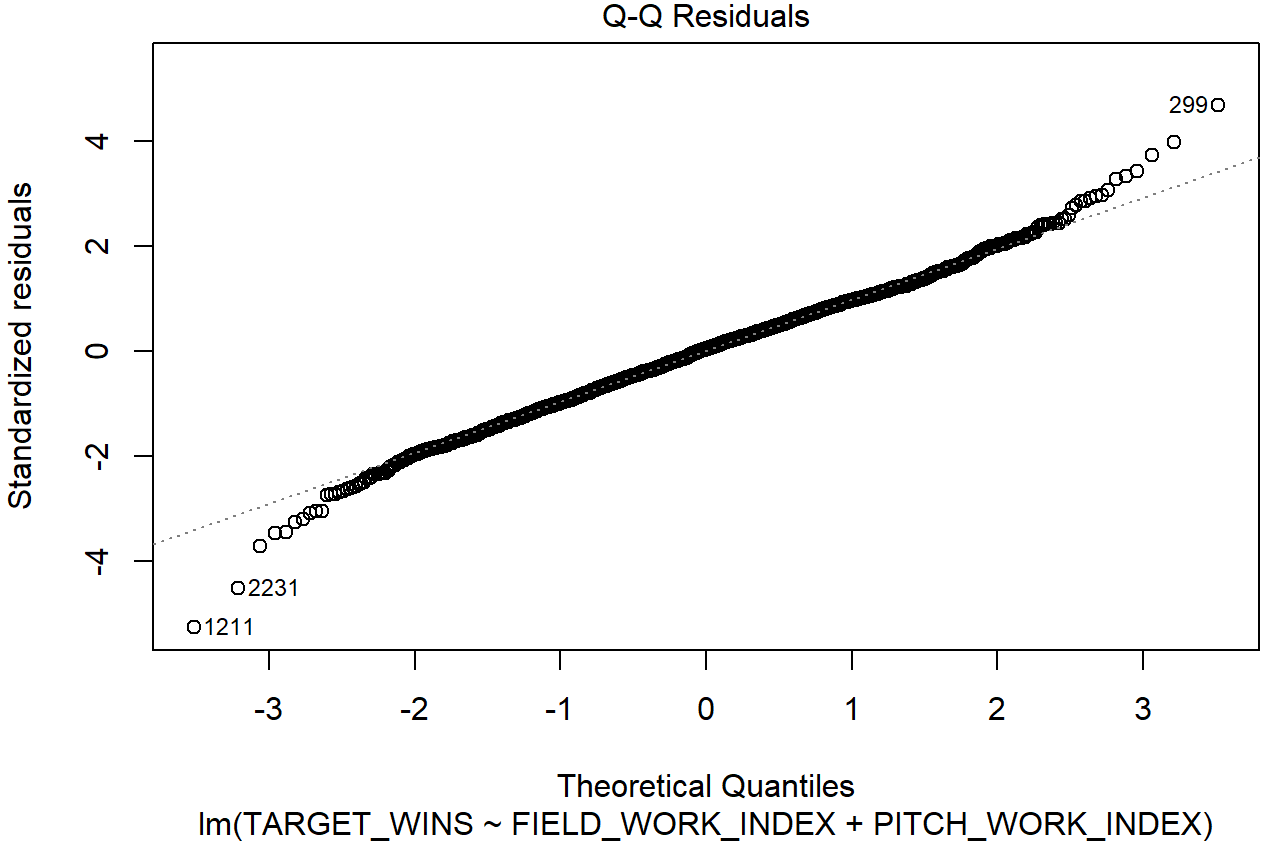

Residuals vs Leverage
299
417
1210
Cook's distance
Standardized residuals
4
2
0
-2
-4
0.000
0.005
0.010
0.015
0.020
0.025
0.030
Leverage
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN

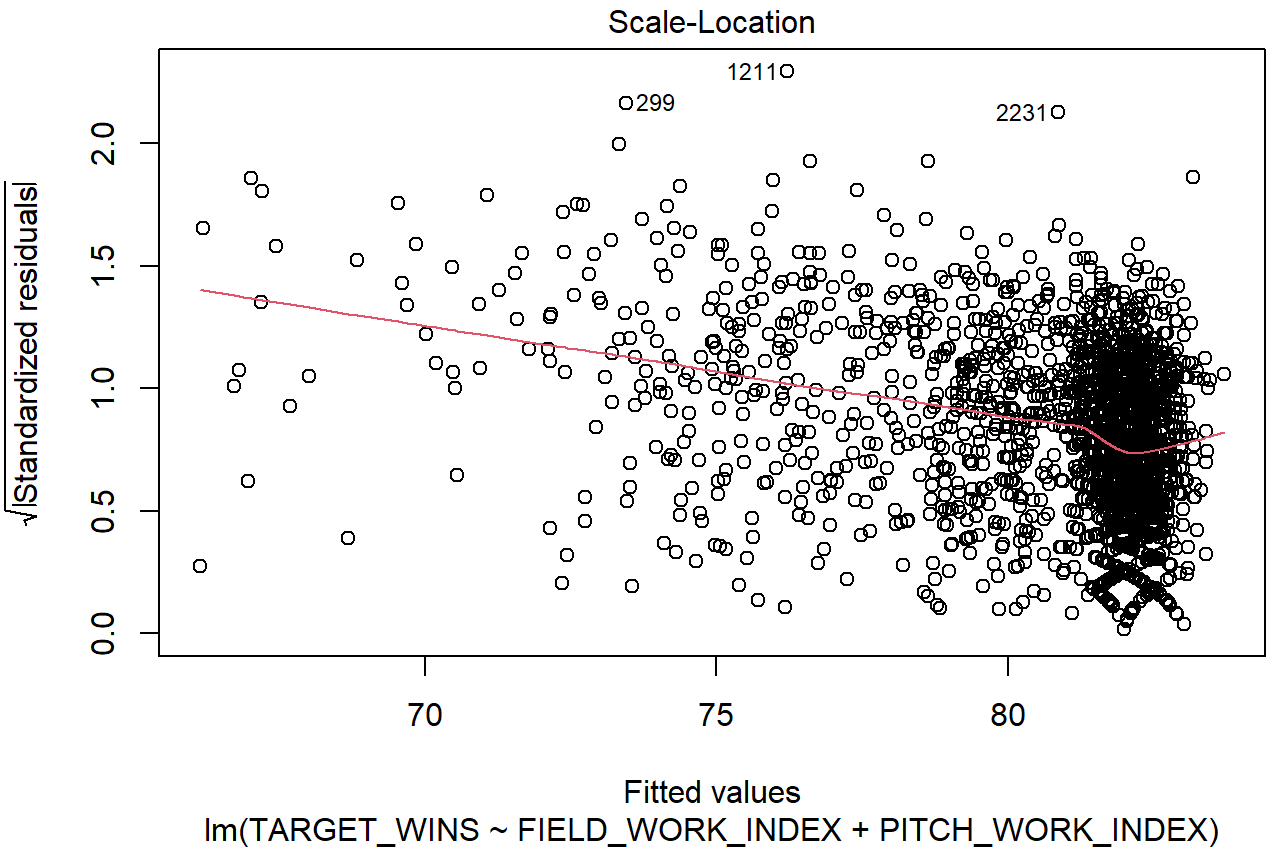
```
## R^2 is 2.6%, field coefficient is neg probably b/c of  TEAM_FIELDING_E. Should transform,
model4 <- lm(TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX, data=model_4_dfa)
print(summary(model4))
```

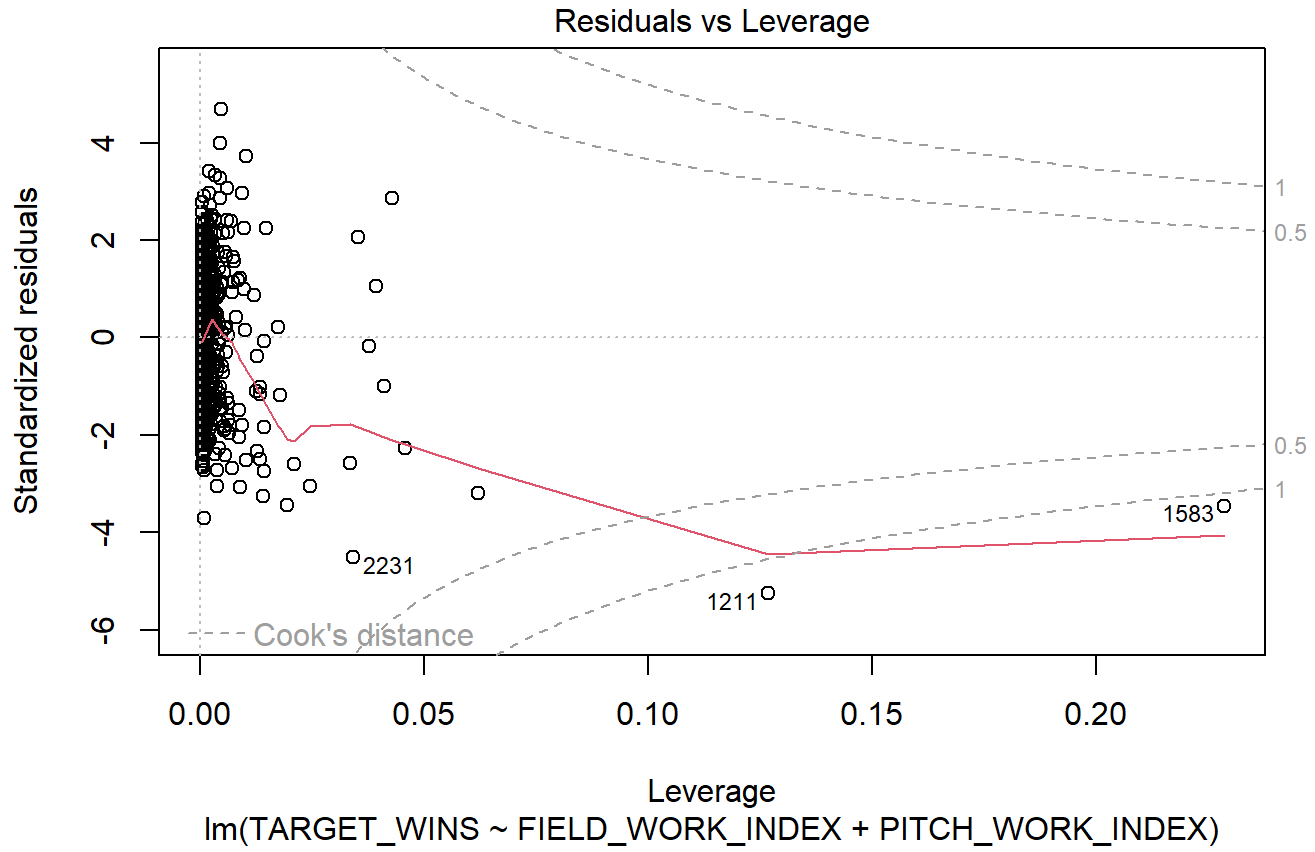
```
##
## Call:
## lm(formula = TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX,
##     data = model_4_dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -76.219  -9.996   0.544  10.294  72.538
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      81.0242086  0.4966042 163.156  < 2e-16 ***
## FIELD_WORK_INDEX  0.0128692  0.0017484   7.361 2.55e-13 ***
## PITCH_WORK_INDEX -0.0007428  0.0003085  -2.408   0.0161 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.52 on 2271 degrees of freedom
## Multiple R-squared:  0.0275, Adjusted R-squared:  0.02664
## F-statistic: 32.11 on 2 and 2271 DF,  p-value: 1.778e-14
```

```
plot(model4)
```


Residuals vs Fitted
Residuals
50
0
-50
299
1211
2231
70
75
80
Fitted values
lm(TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX)


Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
299
2231
1211
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX)

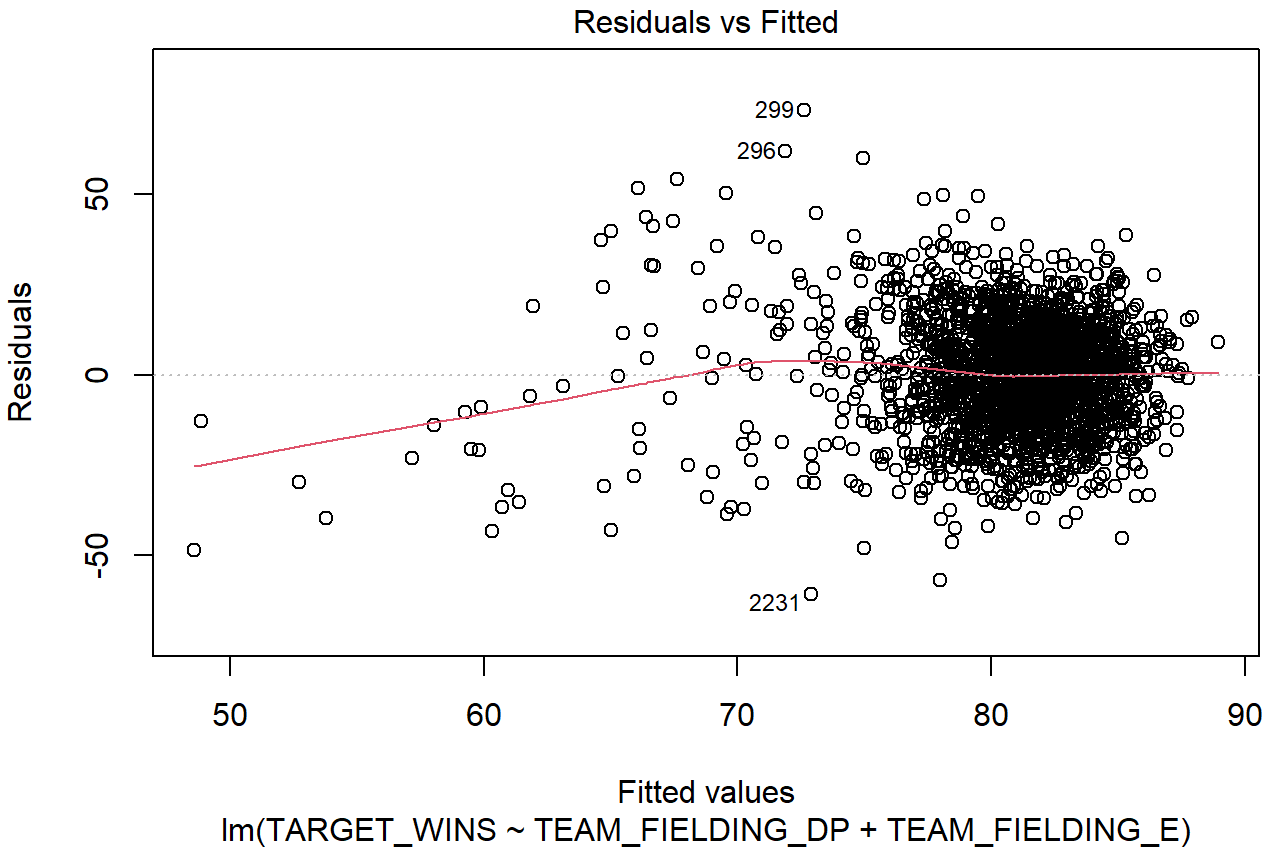

Scale-Location
1211
299
2231
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
70
75
80
Fitted values
lm(TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX)

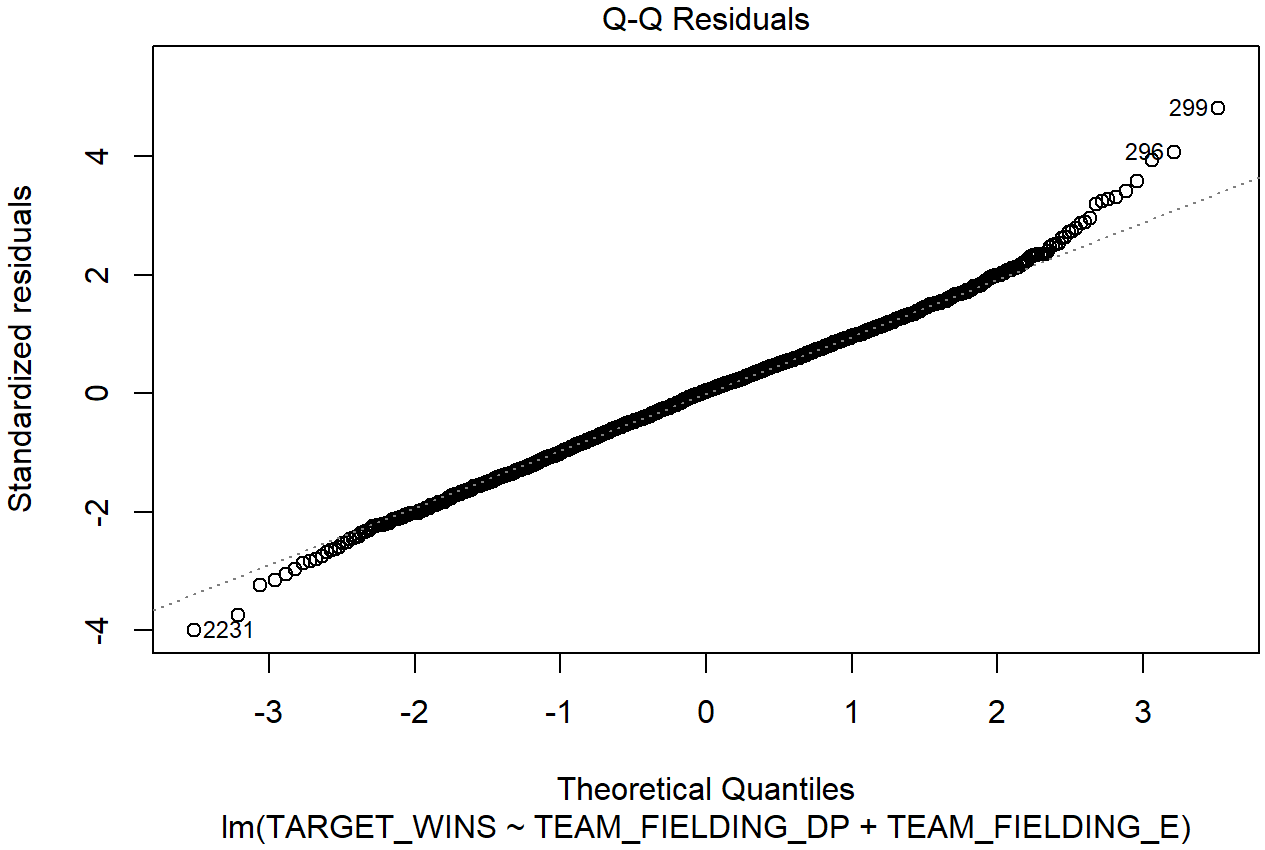

Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
-6
1
0.5
0.5
1
1583
2231
1211
Cook's distance
0.00
0.05
0.10
0.15
0.20
Leverage
lm(TARGET_WINS ~ FIELD_WORK_INDEX + PITCH_WORK_INDEX)

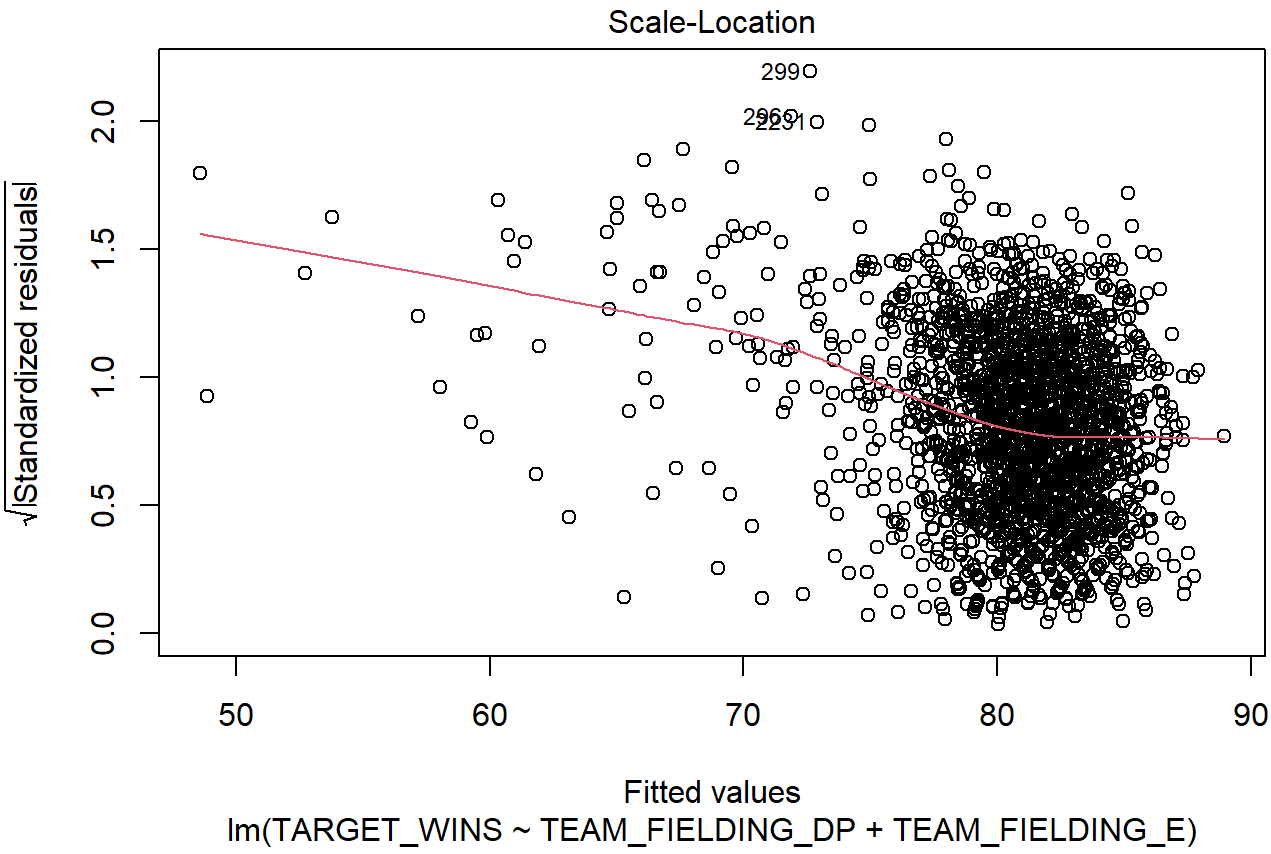
```
## R^2 is 5.7%
model4_def_field <- lm(TARGET_WINS ~ TEAM_FIELDING_DP+ TEAM_FIELDING_E, data=model_4_df_defen
sive_field_granular)
print(summary(model4_def_field))
```

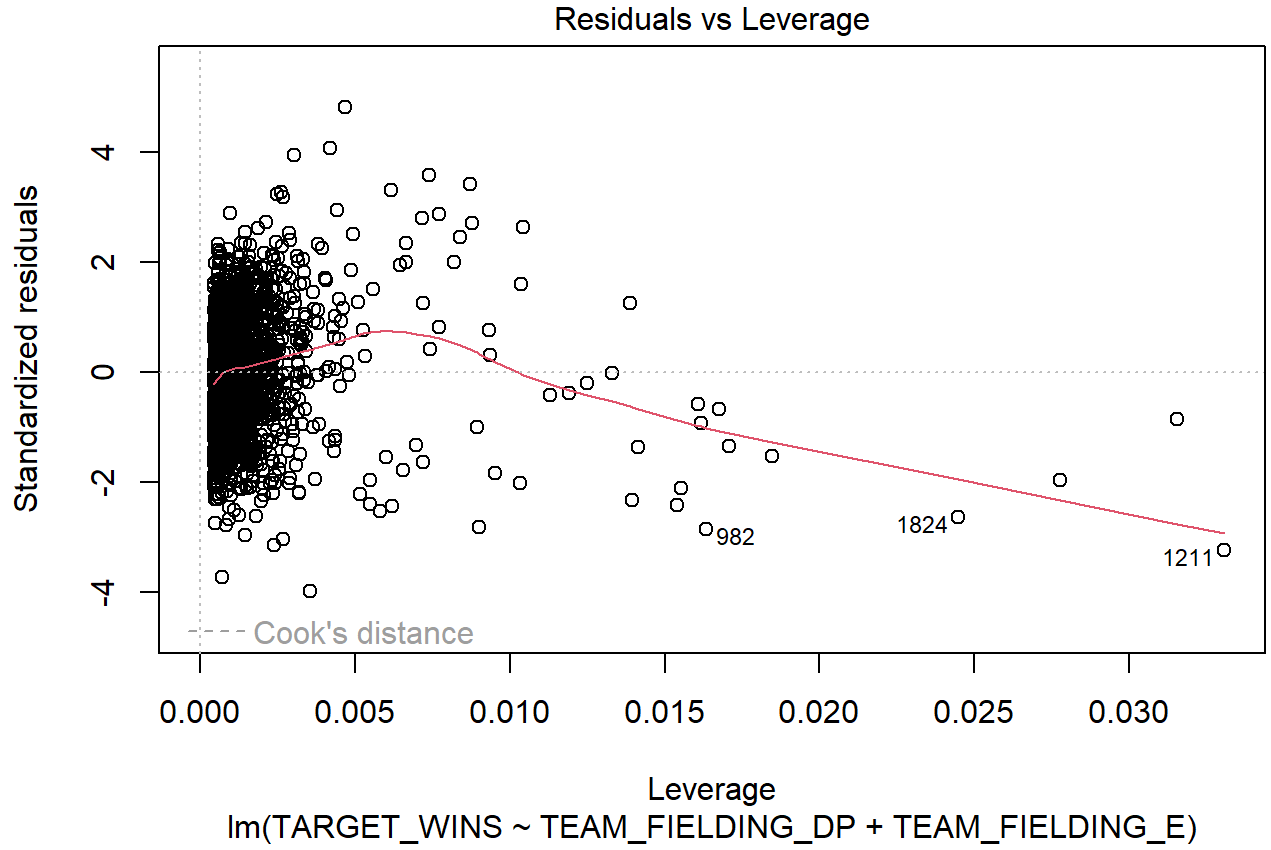
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_FIELDING_DP + TEAM_FIELDING_E,
##     data = model_4_df_defensive_field_granular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.906  -9.951   0.547   9.827  73.382
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      101.004634   2.153992  46.892  < 2e-16 ***
## TEAM_FIELDING_DP  -0.108006   0.013182  -8.193 4.19e-16 ***
## TEAM_FIELDING_E   -0.020423   0.001734 -11.779  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.27 on 2271 degrees of freedom
## Multiple R-squared:  0.05863,    Adjusted R-squared:  0.0578
## F-statistic: 70.72 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model4_def_field)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ TEAM_FIELDING_DP + TEAM_FIELDING_E)
299
296
2231
50
0
-50
50
60
70
80
90


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ TEAM_FIELDING_DP + TEAM_FIELDING_E)
299
296
2231
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

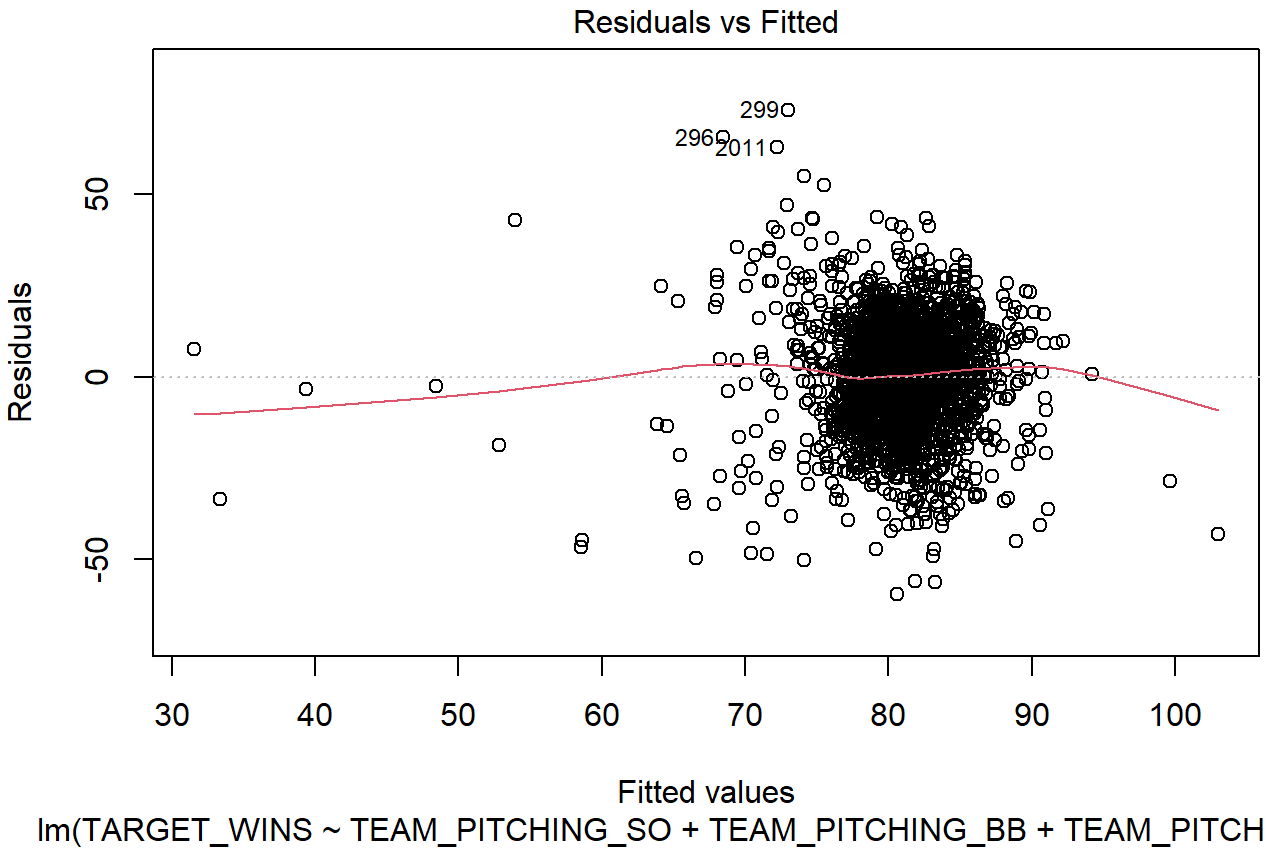

Scale-Location
299
2991
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
50
60
70
80
90
Fitted values
lm(TARGET_WINS ~ TEAM_FIELDING_DP + TEAM_FIELDING_E)

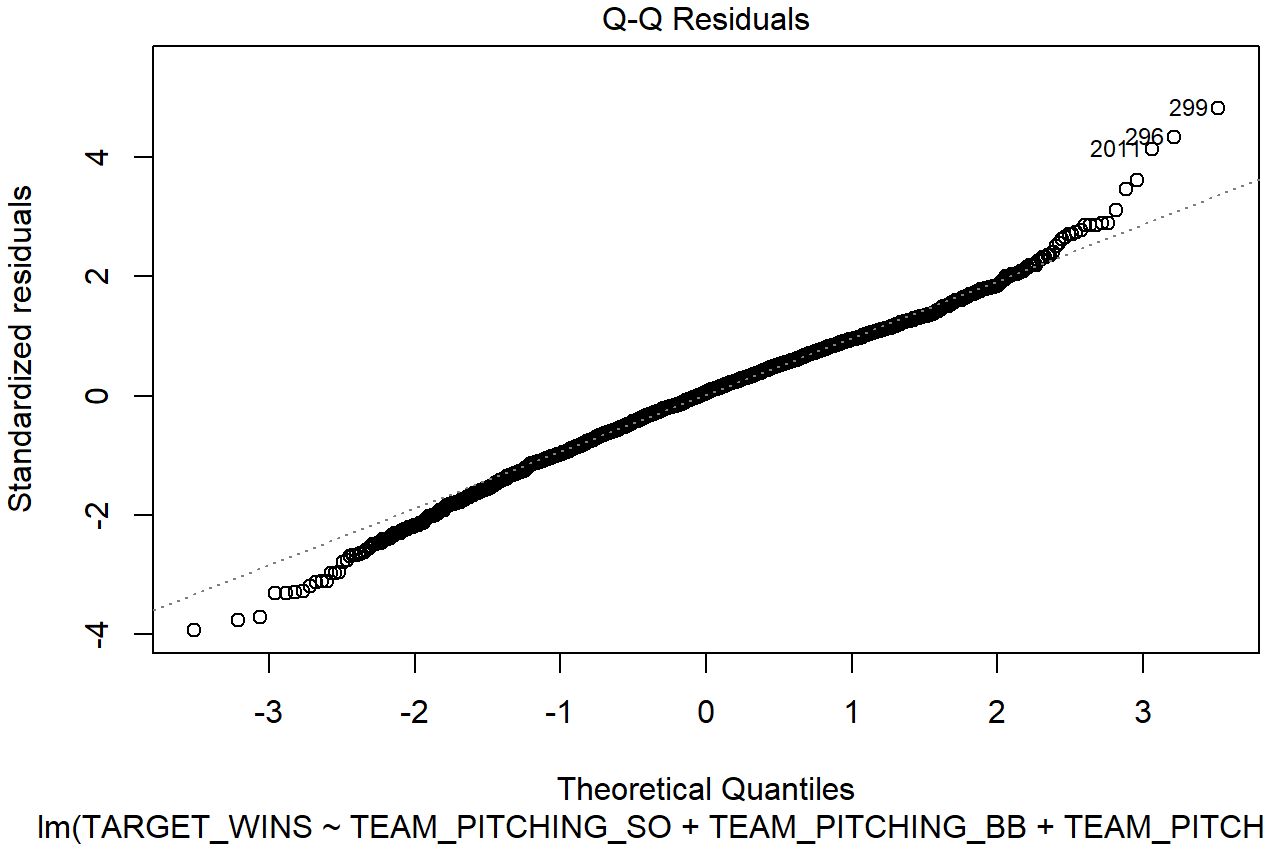

Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
Cook's distance
982
1824
1211
0.000
0.005
0.010
0.015
0.020
0.025
0.030
Leverage
lm(TARGET_WINS ~ TEAM_FIELDING_DP + TEAM_FIELDING_E)

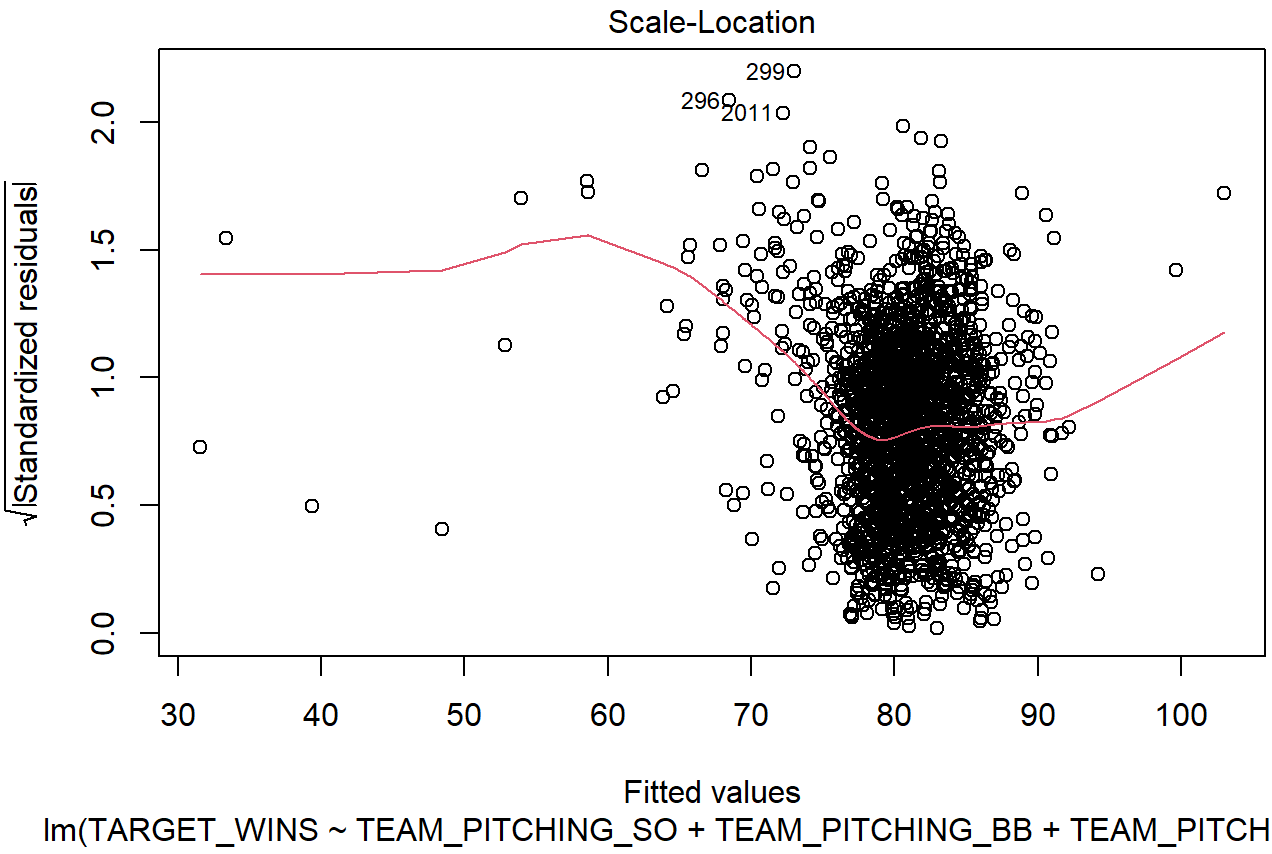
```
## R^2 is 6.6%, neg correlation makes sense
model4_def_pitch <- lm(TARGET_WINS ~ TEAM_PITCHING_SO+TEAM_PITCHING_BB+TEAM_PITCHING_H, data=
model_4_df_defensive_pitch_granular)
print(summary(model4_def_pitch))
```

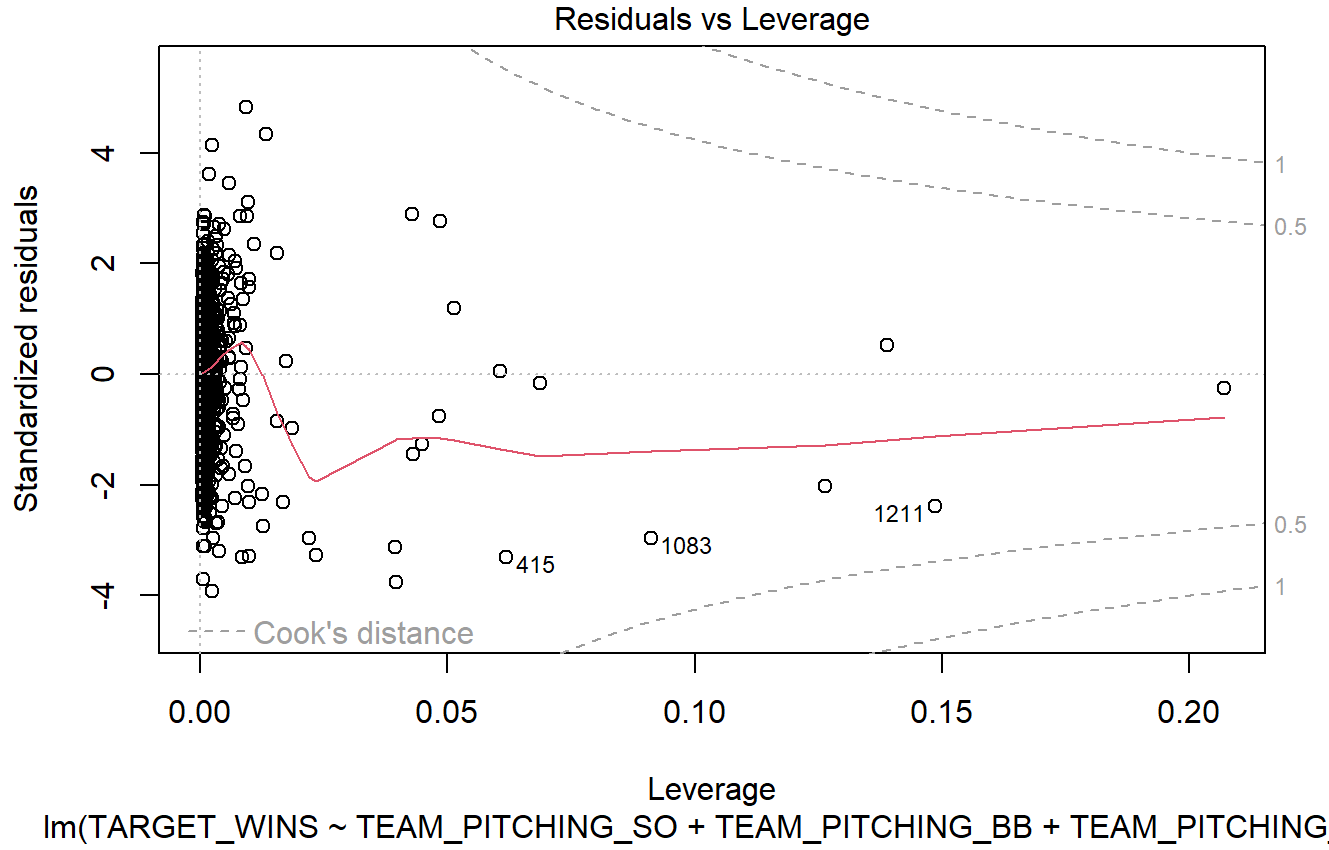
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_PITCHING_BB +
##     TEAM_PITCHING_H, data = model_4_df_defensive_pitch_granular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -59.621  -9.412   0.789  10.052  72.966
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      79.0080593  1.4307605  55.221  < 2e-16 ***
## TEAM_PITCHING_SO -0.0093176  0.0011475  -8.120 7.56e-16 ***
## TEAM_PITCHING_BB  0.0228060  0.0022950   9.937  < 2e-16 ***
## TEAM_PITCHING_H  -0.0018958  0.0002471  -7.673 2.48e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.2 on 2270 degrees of freedom
## Multiple R-squared:  0.06762,    Adjusted R-squared:  0.06639
## F-statistic: 54.88 on 3 and 2270 DF,  p-value: < 2.2e-16
```

```
plot(model4_def_pitch)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_PITCHING_BB + TEAM_PITCHING_
299
296
2011
50
0
-50
30
40
50
60
70
80
90
100


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_PITCHING_BB + TEAM_PITCHING_
299
296
2011
4
2
0
-2
-4
-3
-2
-1
0
1
2
3

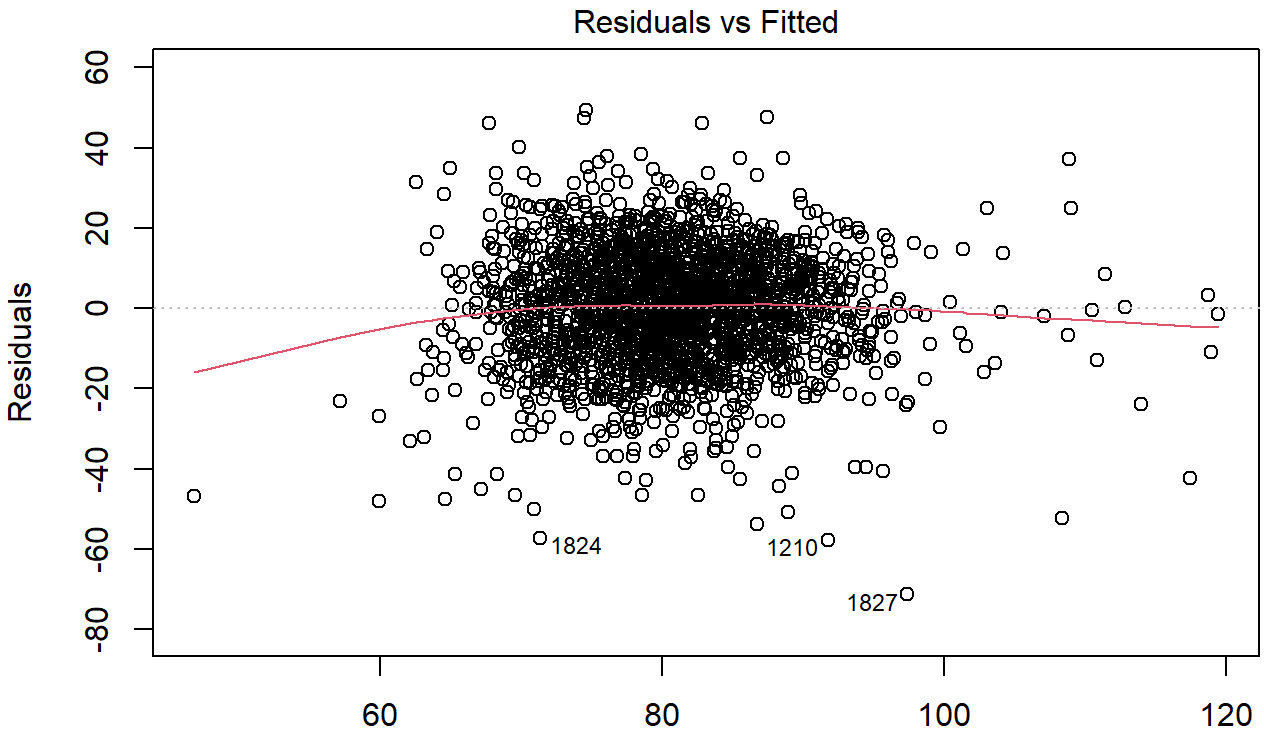

Scale-Location
299
296
2011
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
30
40
50
60
70
80
90
100
Fitted values
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_PITCHING_BB + TEAM_PITCHING_


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
1
0.5
0.5
1
1211
1083
415
Cook's distance
0.00
0.05
0.10
0.15
0.20
Leverage
lm(TARGET_WINS ~ TEAM_PITCHING_SO + TEAM_PITCHING_BB + TEAM_PITCHING_

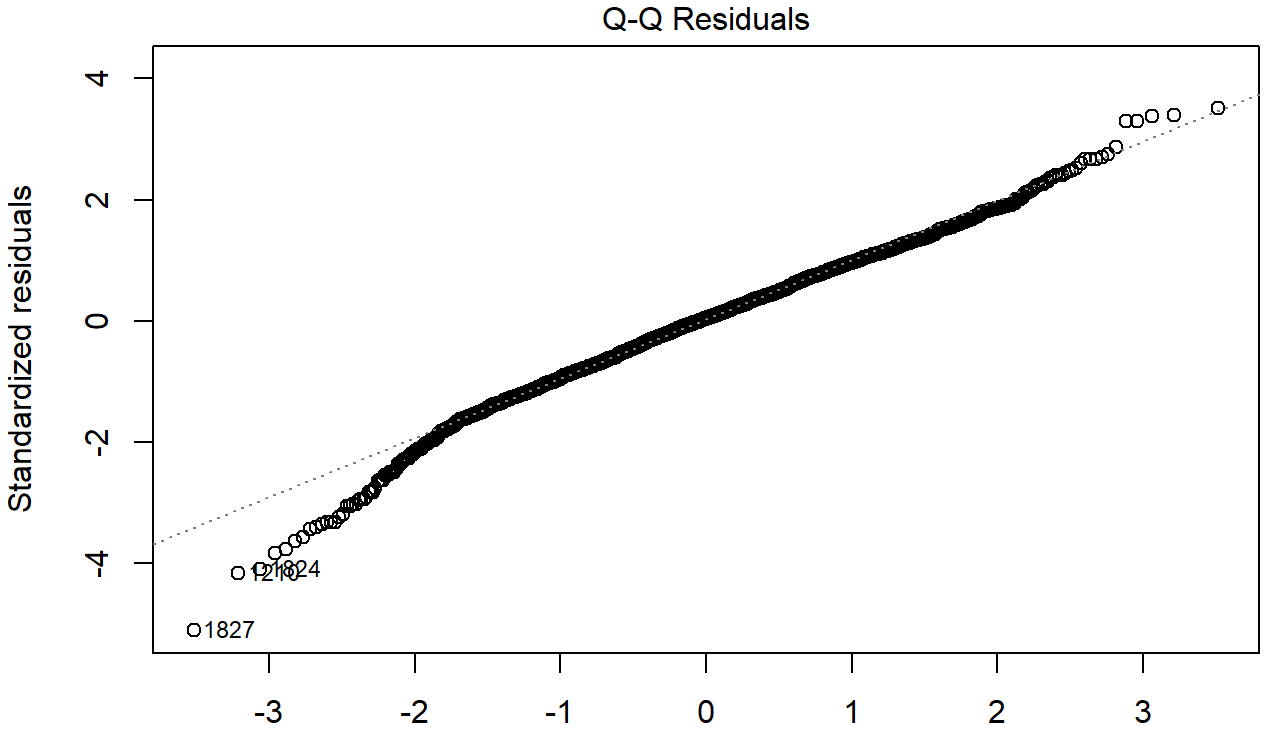
```
### Selecting the best performing models/ most interesting models, and figuring out possible transforms for improvement.

## Model1 is decent, not amazing though
summary(model1)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B +
##     TEAM_BATTING_2B + TEAM_BATTING_HR, data = model_1_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -71.390  -8.854   0.603   9.632  49.361
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     18.031397   3.188915   5.654 1.76e-08 ***
## TEAM_BATTING_1B  0.032043   0.003092  10.364  < 2e-16 ***
## TEAM_BATTING_3B  0.157431   0.015193  10.362  < 2e-16 ***
## TEAM_BATTING_2B  0.034017   0.007679   4.430 9.89e-06 ***
## TEAM_BATTING_HR  0.114952   0.007682  14.964  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.04 on 2269 degrees of freedom
## Multiple R-squared:  0.2039, Adjusted R-squared:  0.2025
## F-statistic: 145.3 on 4 and 2269 DF,  p-value: < 2.2e-16
```

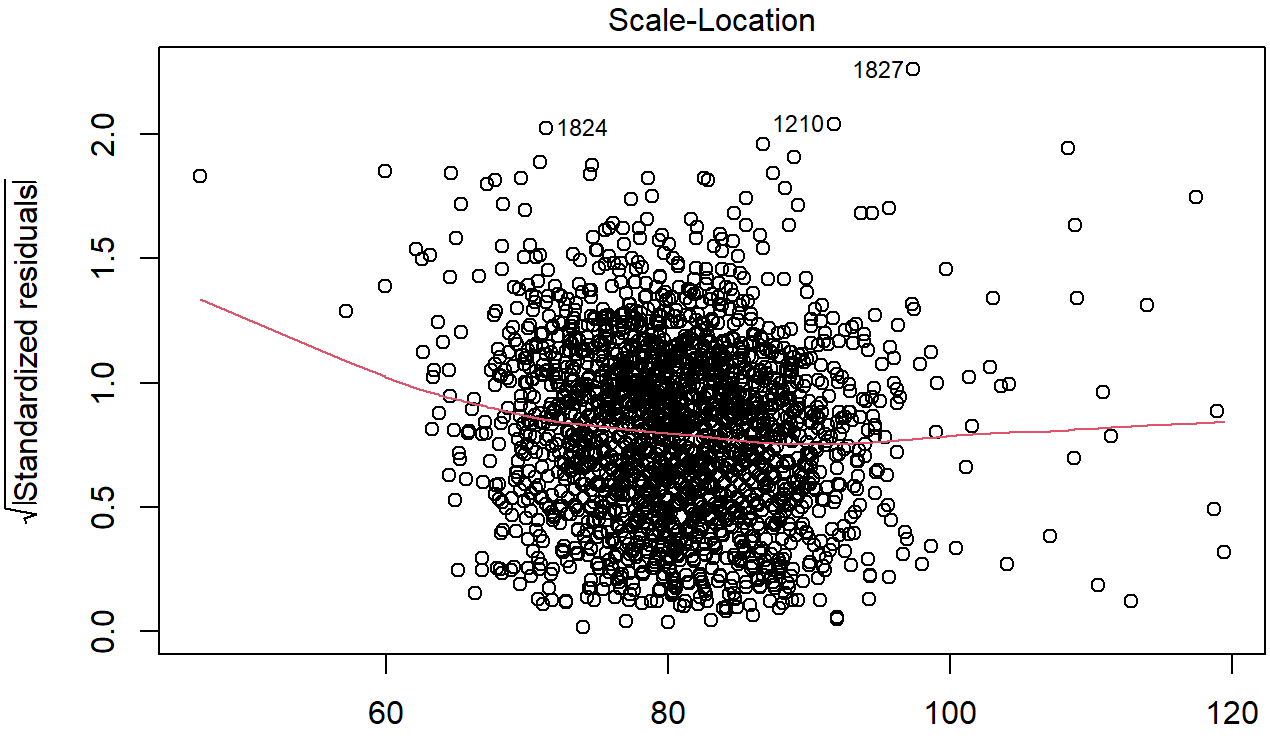
```
plot(model1) ## some leveraged points to look at, slight curve in scale location plot (homosc edascity),
```

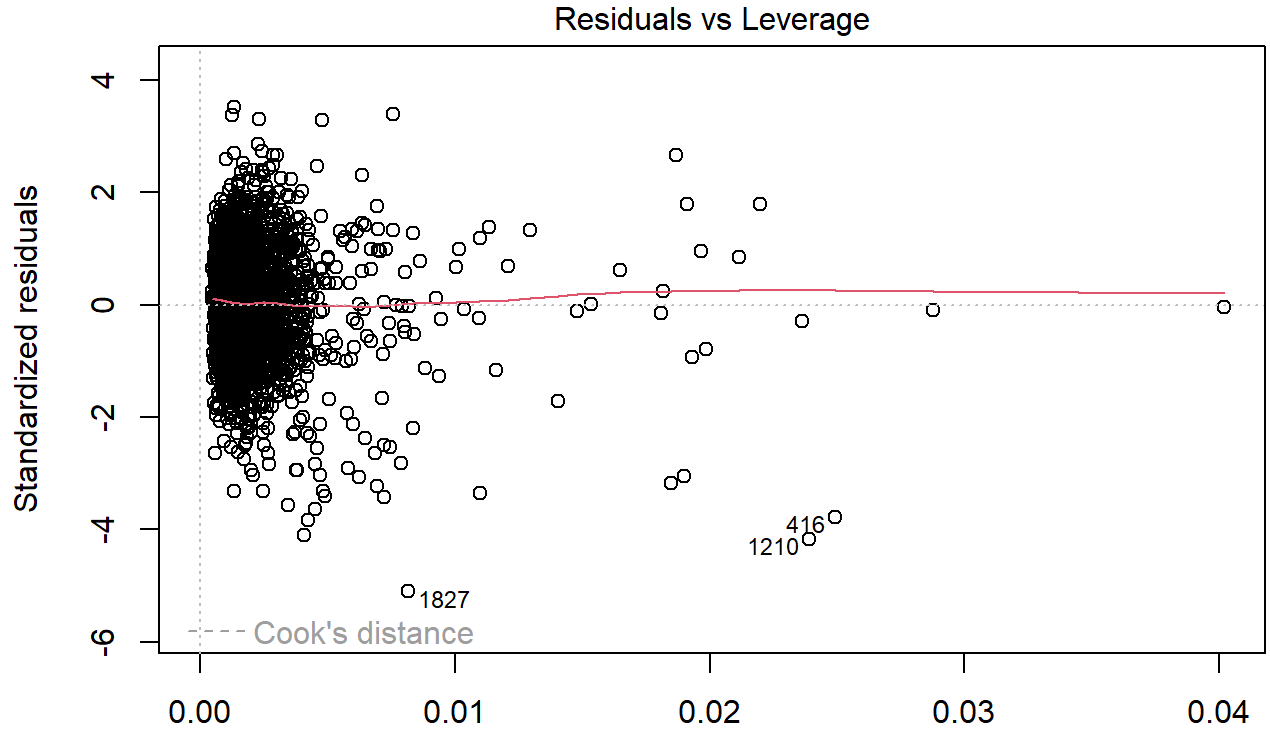


m(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE



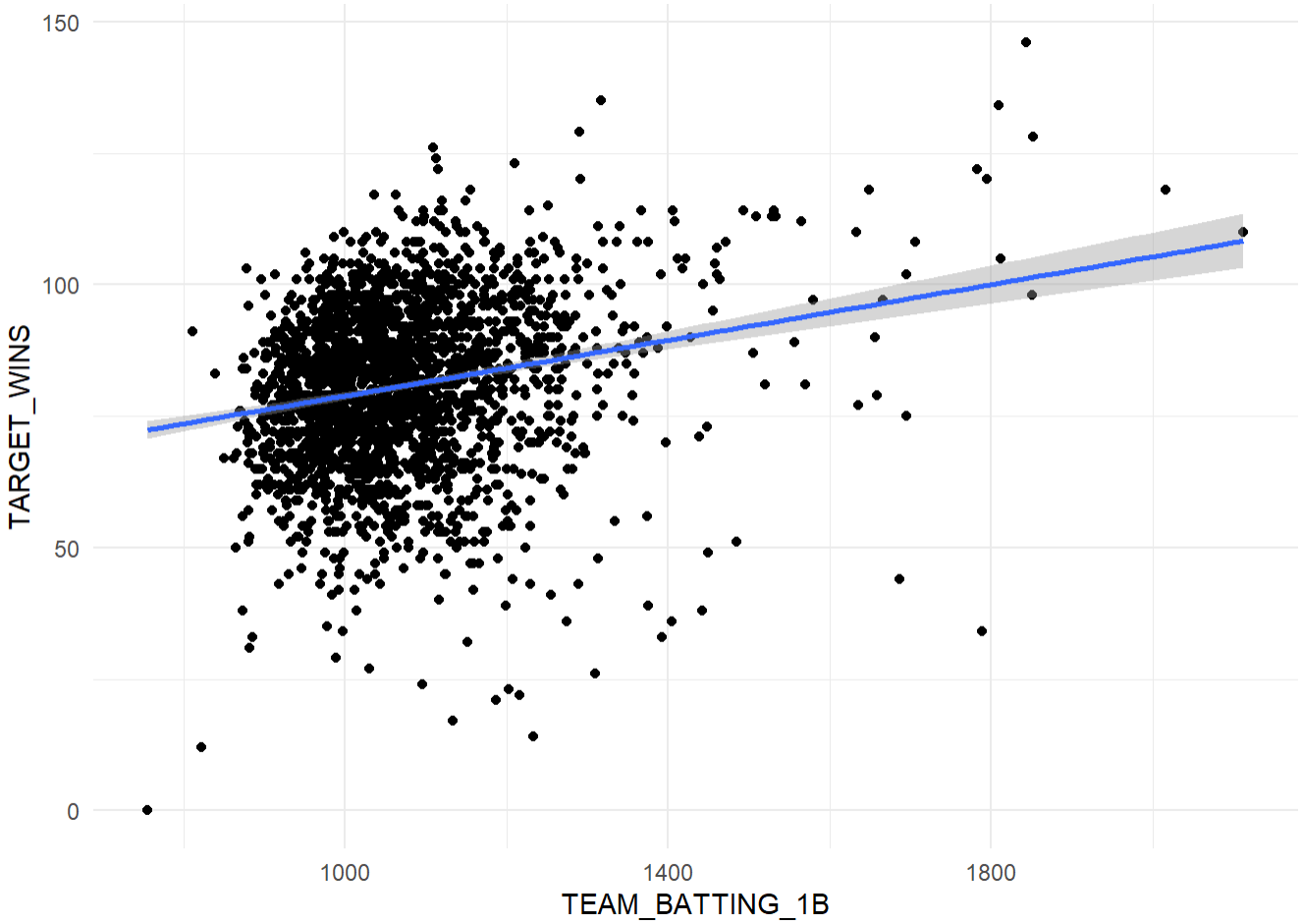
m(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE


Scale-Location
1827
1824
1210
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE


Residuals vs Leverage
Standardized residuals
-6
-4
-2
0
2
4
416
1210
1827
Cook's distance
0.00
0.01
0.02
0.03
0.04
Leverage
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_3B + TEAM_BATTING_2B + TE

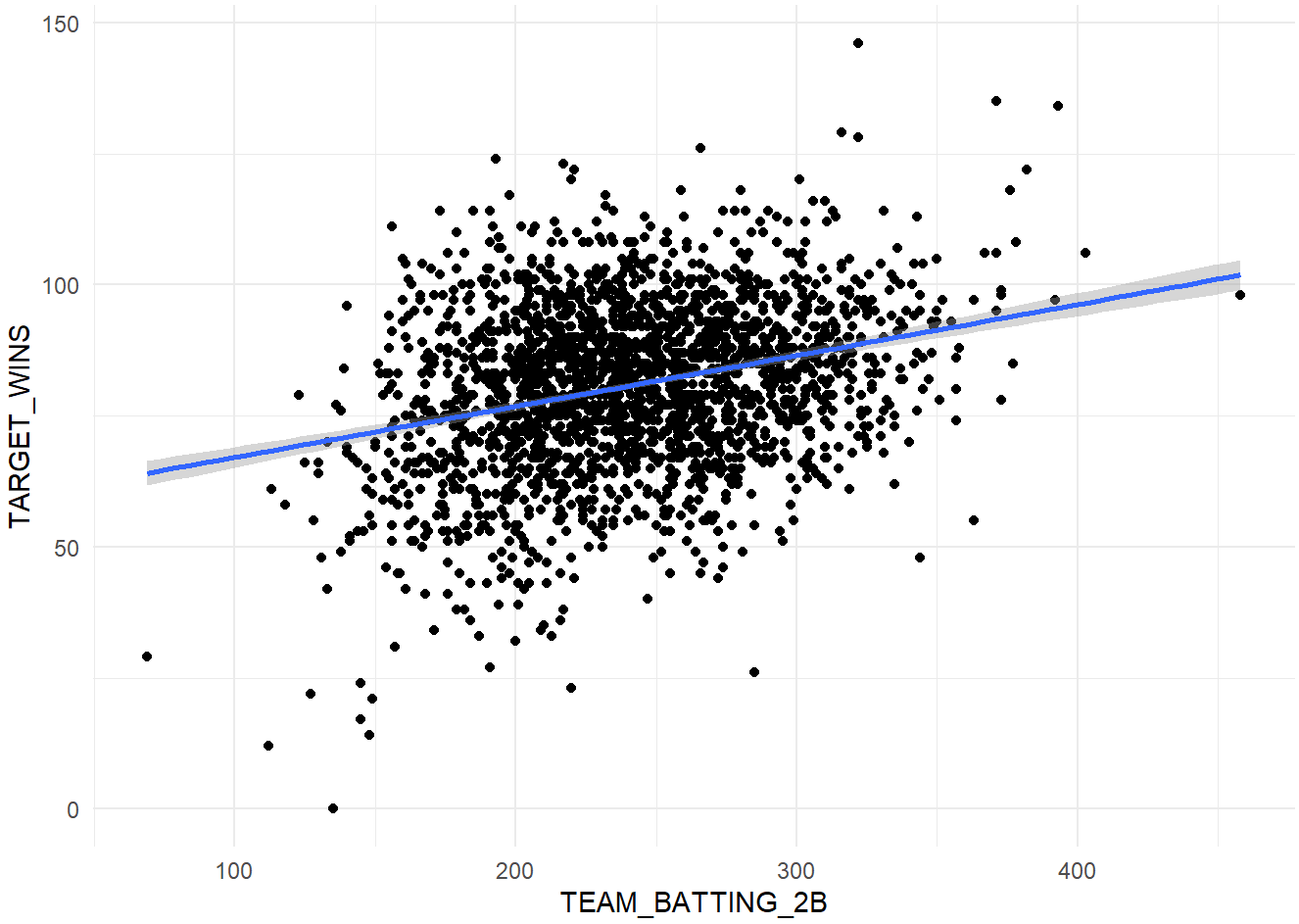
```
#colnames(model_1_df)
## PLOTTING
ggplot(model_1_df, aes(x = TEAM_BATTING_1B, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



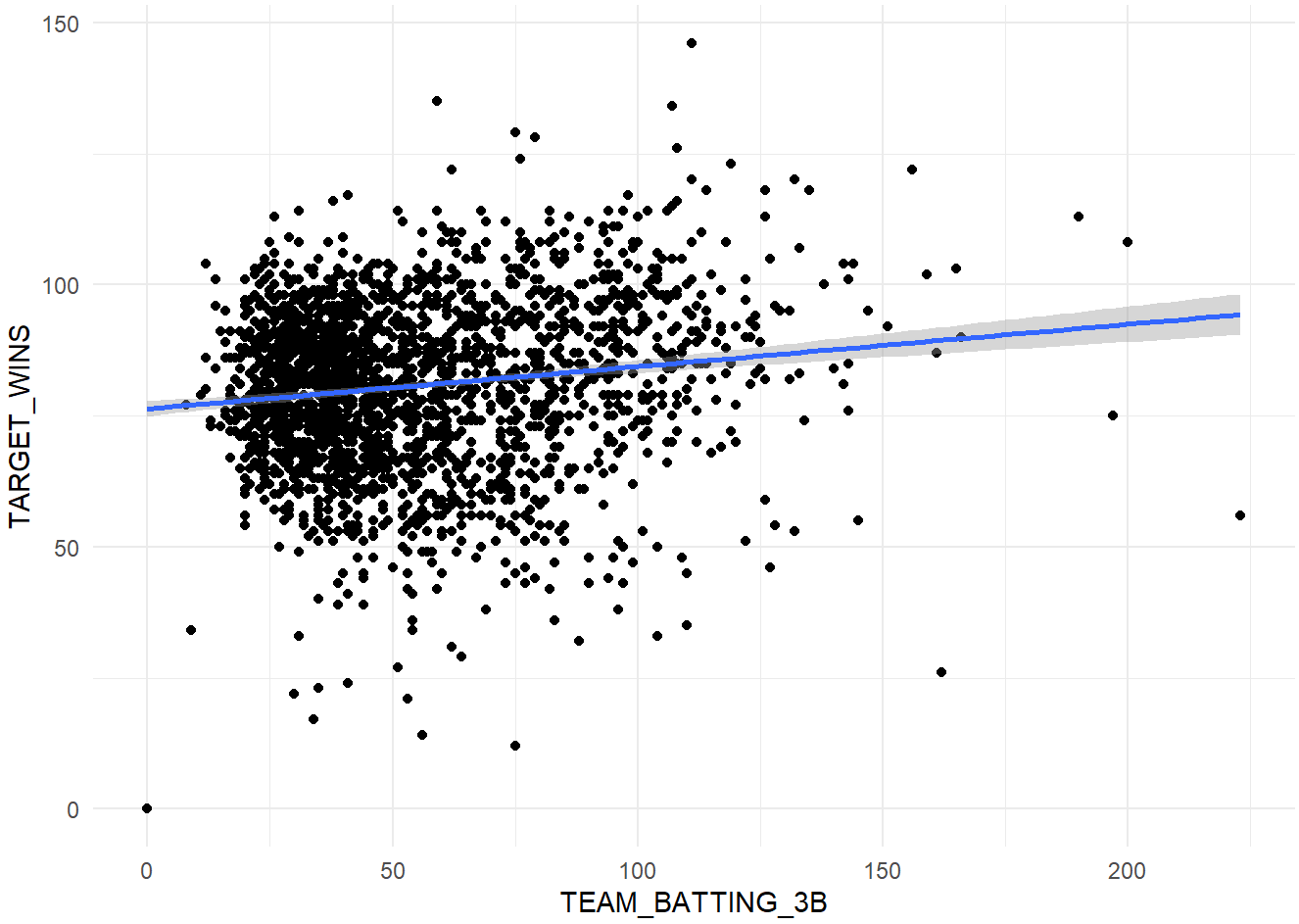

```
ggplot(model_1_df, aes(x = TEAM_BATTING_2B, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



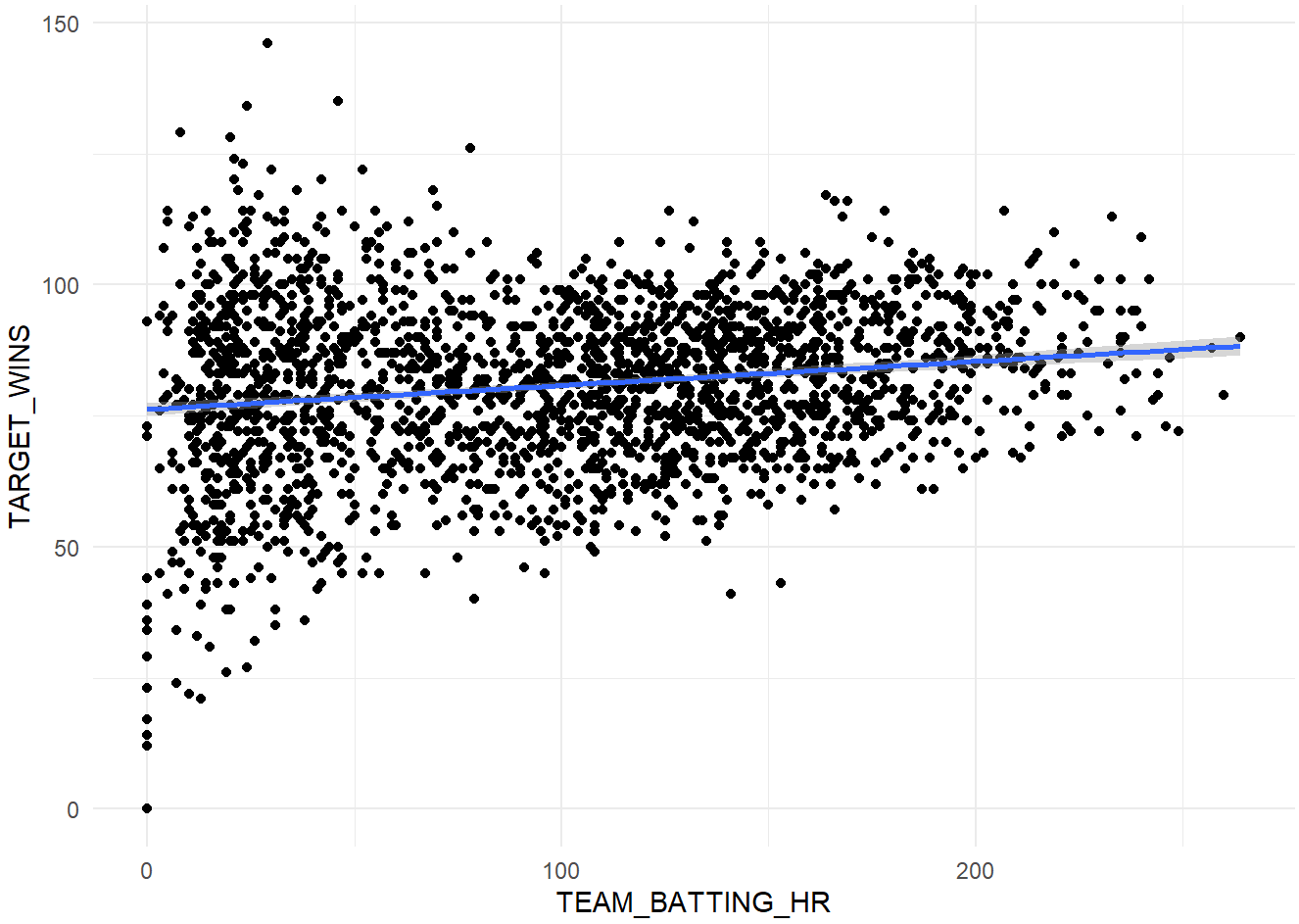

```
ggplot(model_1_df, aes(x = TEAM_BATTING_3B, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(model_1_df, aes(x = TEAM_BATTING_HR, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



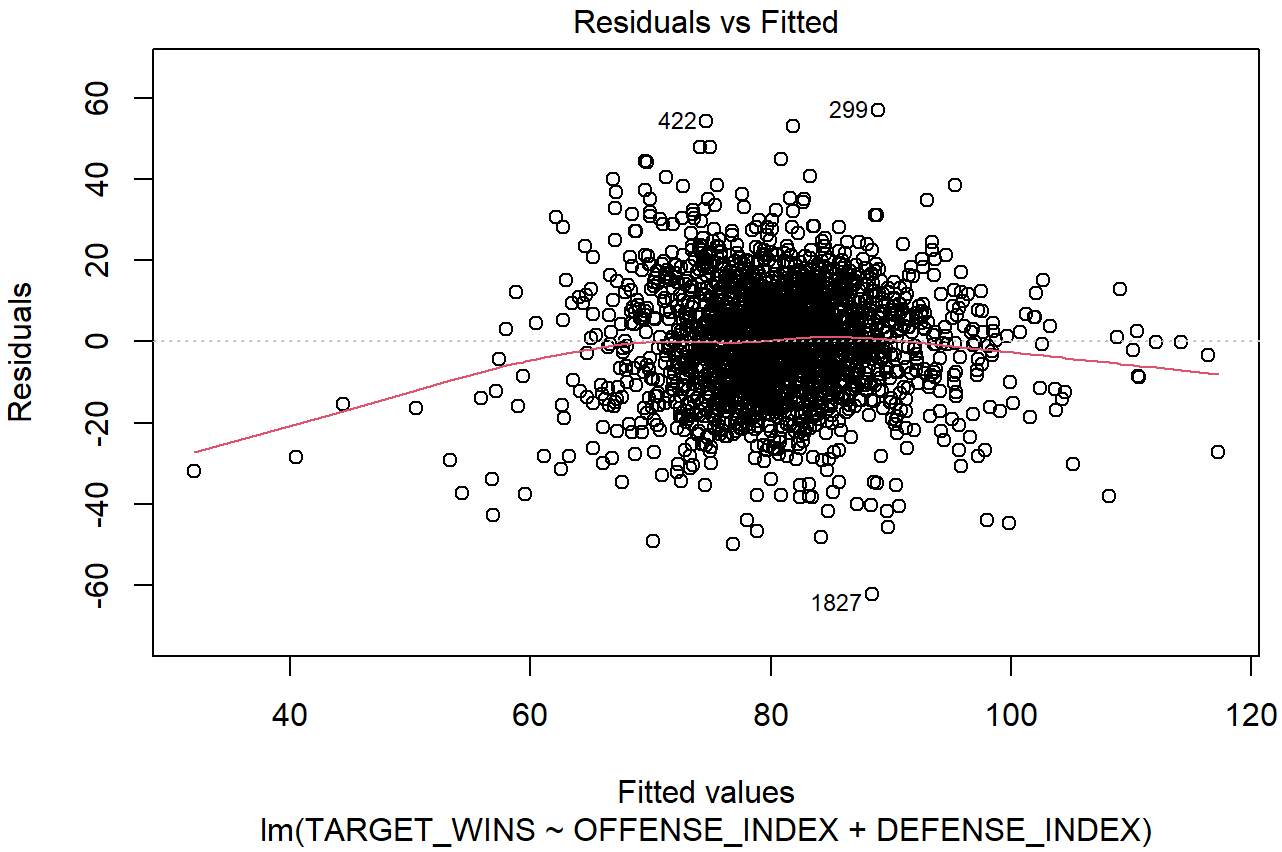

```
### MODEL 2
summary(model2)
```

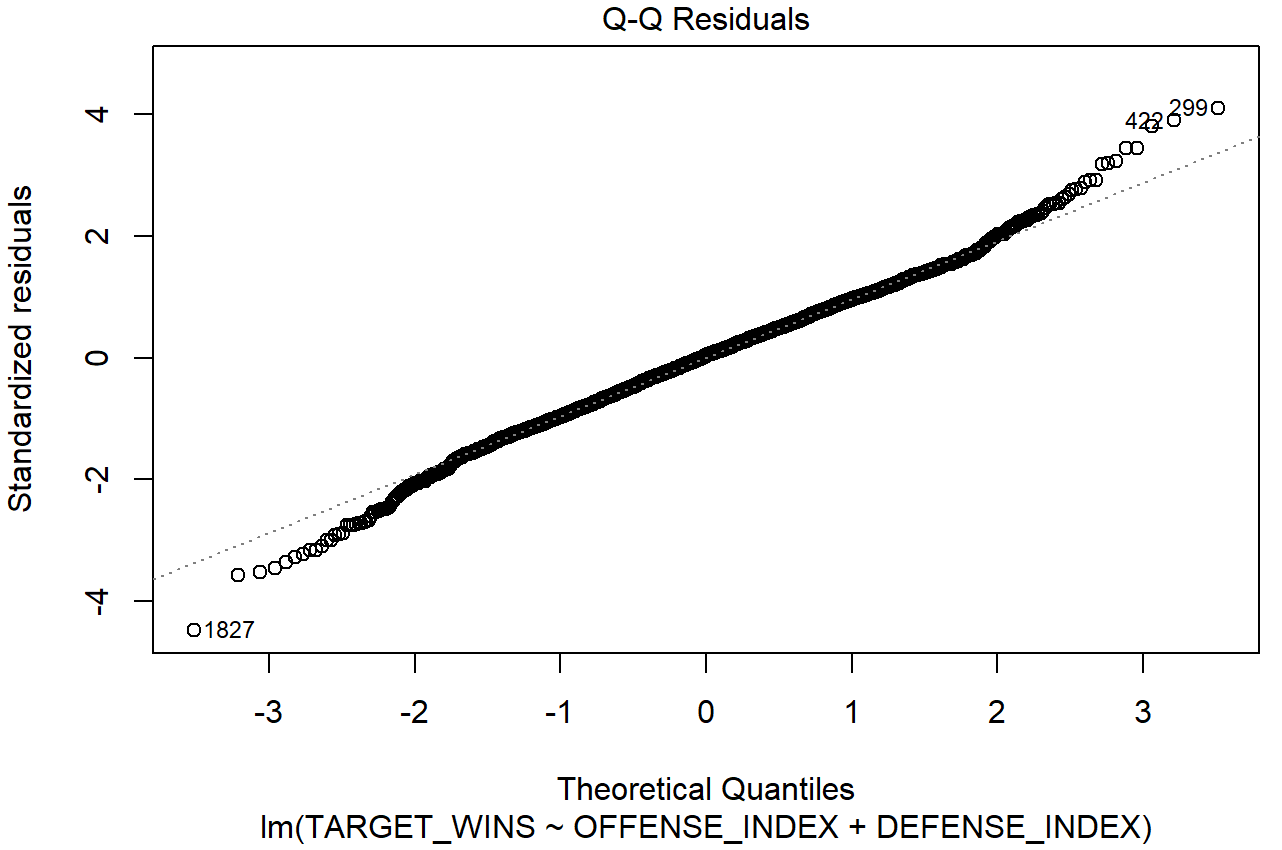
```
##
## Call:
## lm(formula = TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX, data = model_2_dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.424  -8.976   0.481   9.035  57.093
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0615756  3.0863725   1.640  0.10115
## OFFENSE_INDEX 0.0353272  0.0014213  24.855  < 2e-16 ***
## DEFENSE_INDEX 0.0025885  0.0007198   3.596  0.00033 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.95 on 2271 degrees of freedom
## Multiple R-squared:  0.214,  Adjusted R-squared:  0.2133
## F-statistic: 309.2 on 2 and 2271 DF,  p-value: < 2.2e-16
```

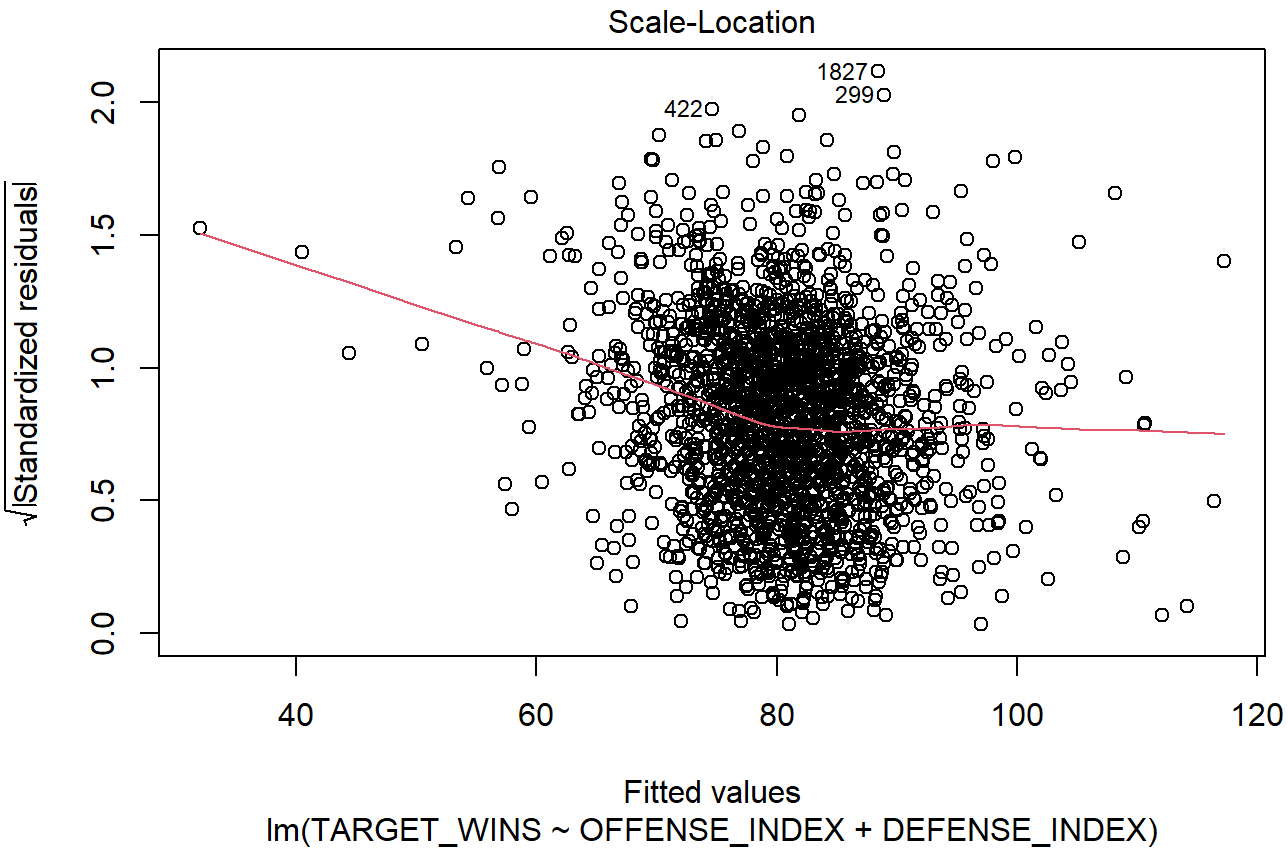
```
summary(model2_off_gran)
```

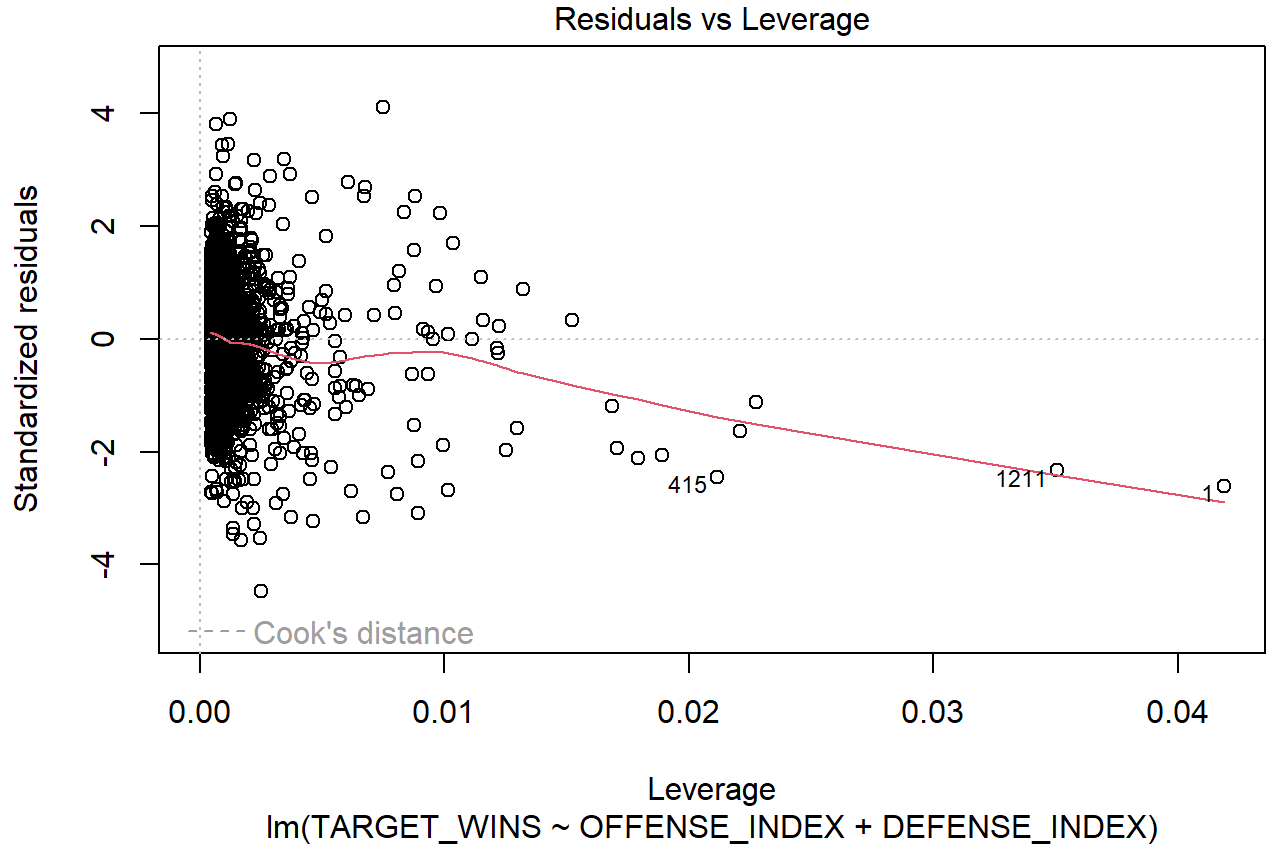
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB +
##     TEAM_BASERUN_SB, data = model_2_df_granualar_offense)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.823  -8.763   0.490   9.235  49.663
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.318103   3.299134  -0.703    0.482
## TEAM_BATTING_H   0.043789   0.002029  21.581  < 2e-16 ***
## TEAM_BATTING_BB  0.033815   0.002379  14.214  < 2e-16 ***
## TEAM_BASERUN_SB  0.014734   0.003381   4.358 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.84 on 2270 degrees of freedom
## Multiple R-squared:  0.2268, Adjusted R-squared:  0.2258
## F-statistic:   222 on 3 and 2270 DF,  p-value: < 2.2e-16
```

```
plot(model2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)
422
299
1827
-60
-40
-20
0
20
40
60
40
60
80
100
120

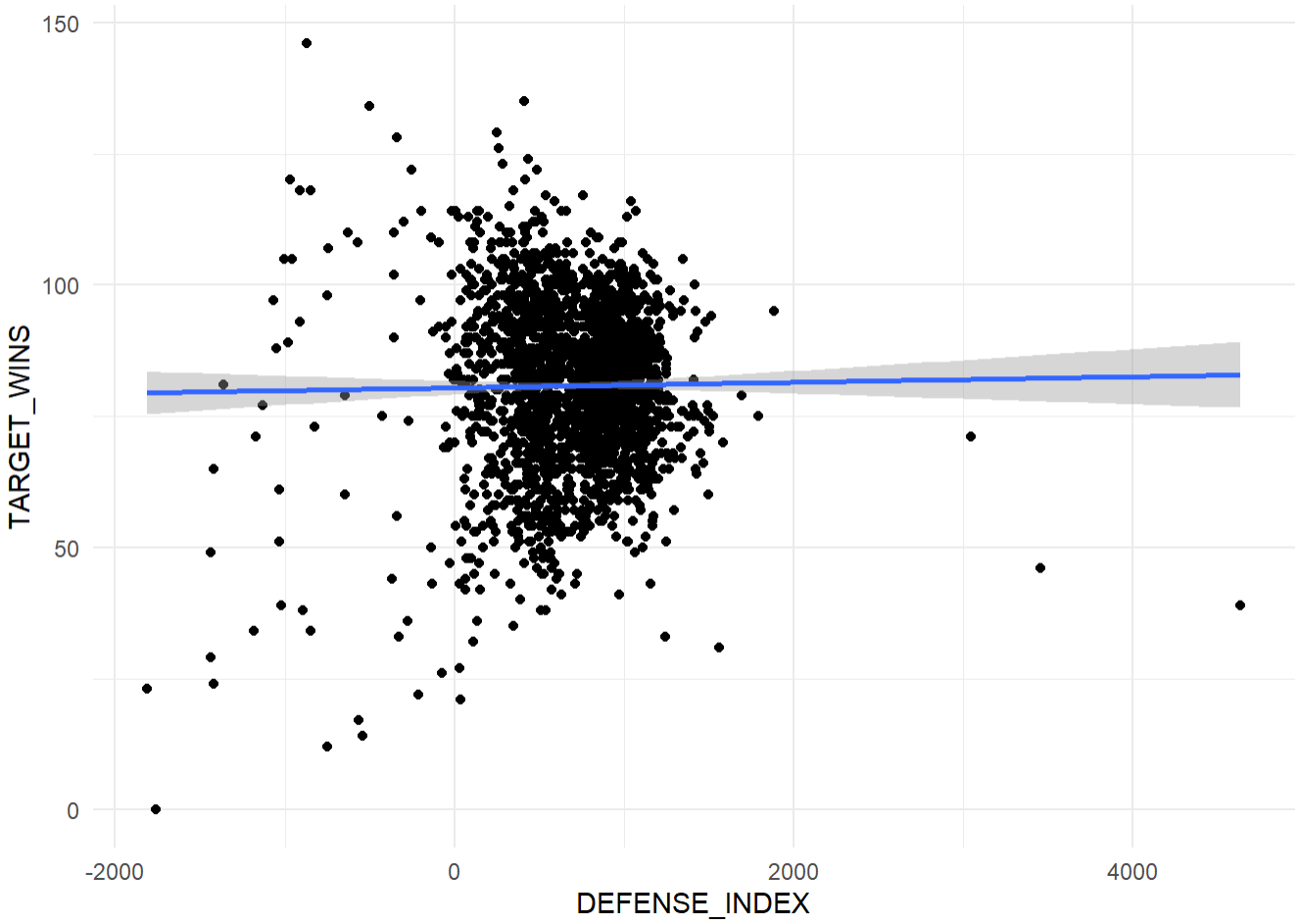

Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)
422
299
1827
-4
-2
0
2
4
-3
-2
-1
0
1
2
3


Scale-Location
1827
299
422
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
415
1211
1
Cook's distance
0.00
0.01
0.02
0.03
0.04
Leverage
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)

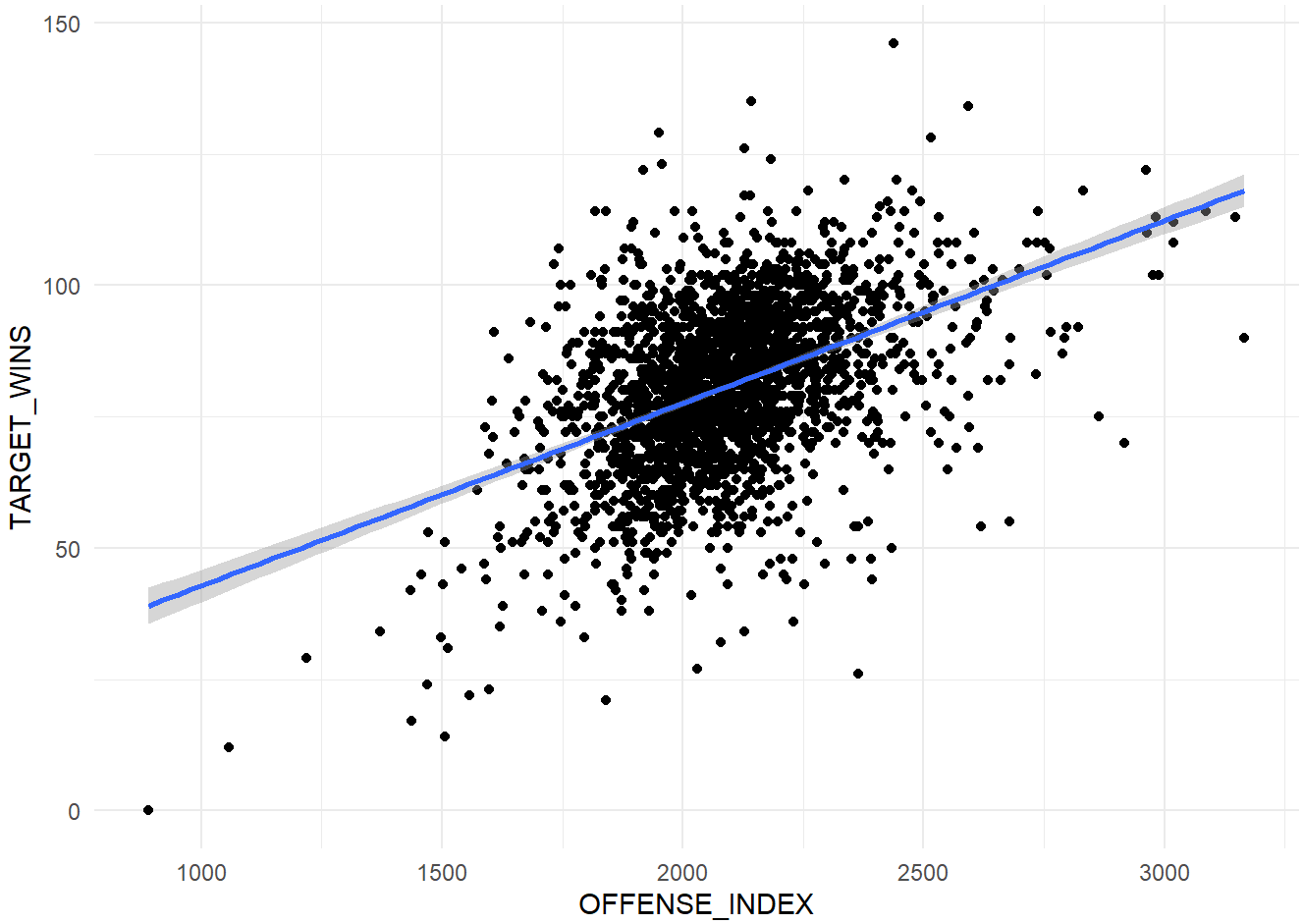
```
## PLOTTING
ggplot(model_2_dfa, aes(x = DEFENSE_INDEX, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



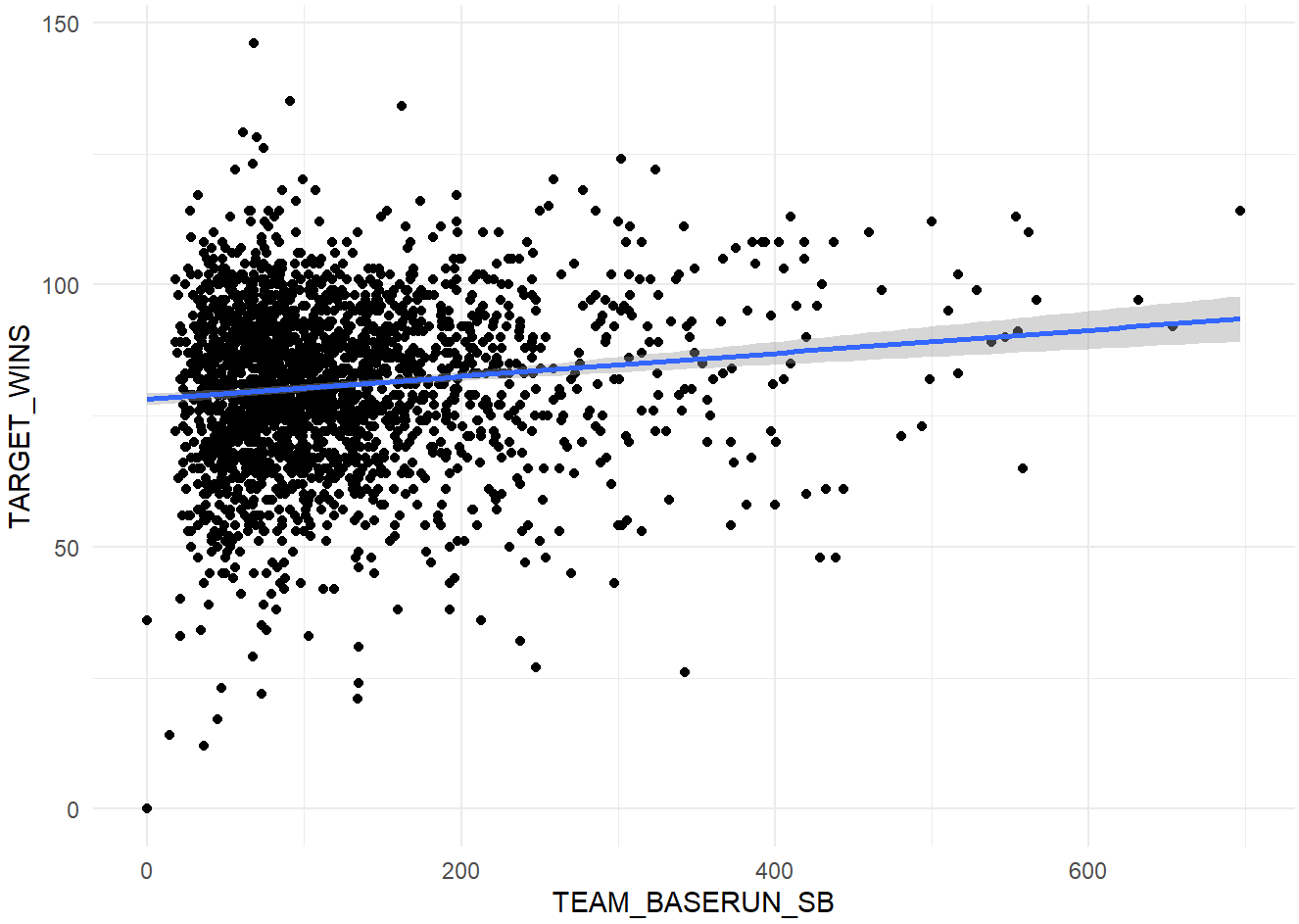

```
ggplot(model_2_dfa, aes(x = OFFENSE_INDEX, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



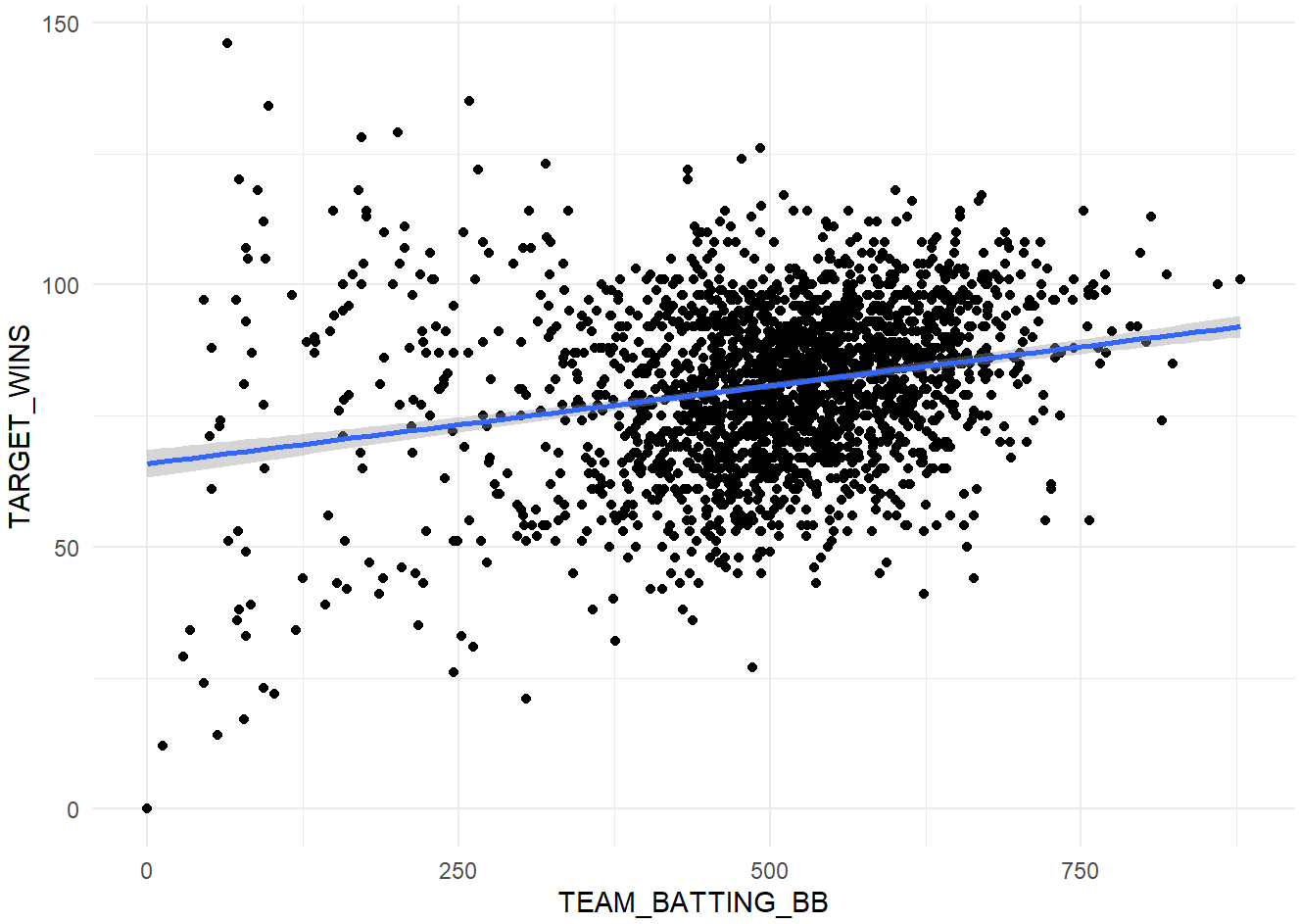

```
## Model Two Granular Offense
ggplot(model_2_df_granualar_offense, aes(x = TEAM_BASERUN_SB, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



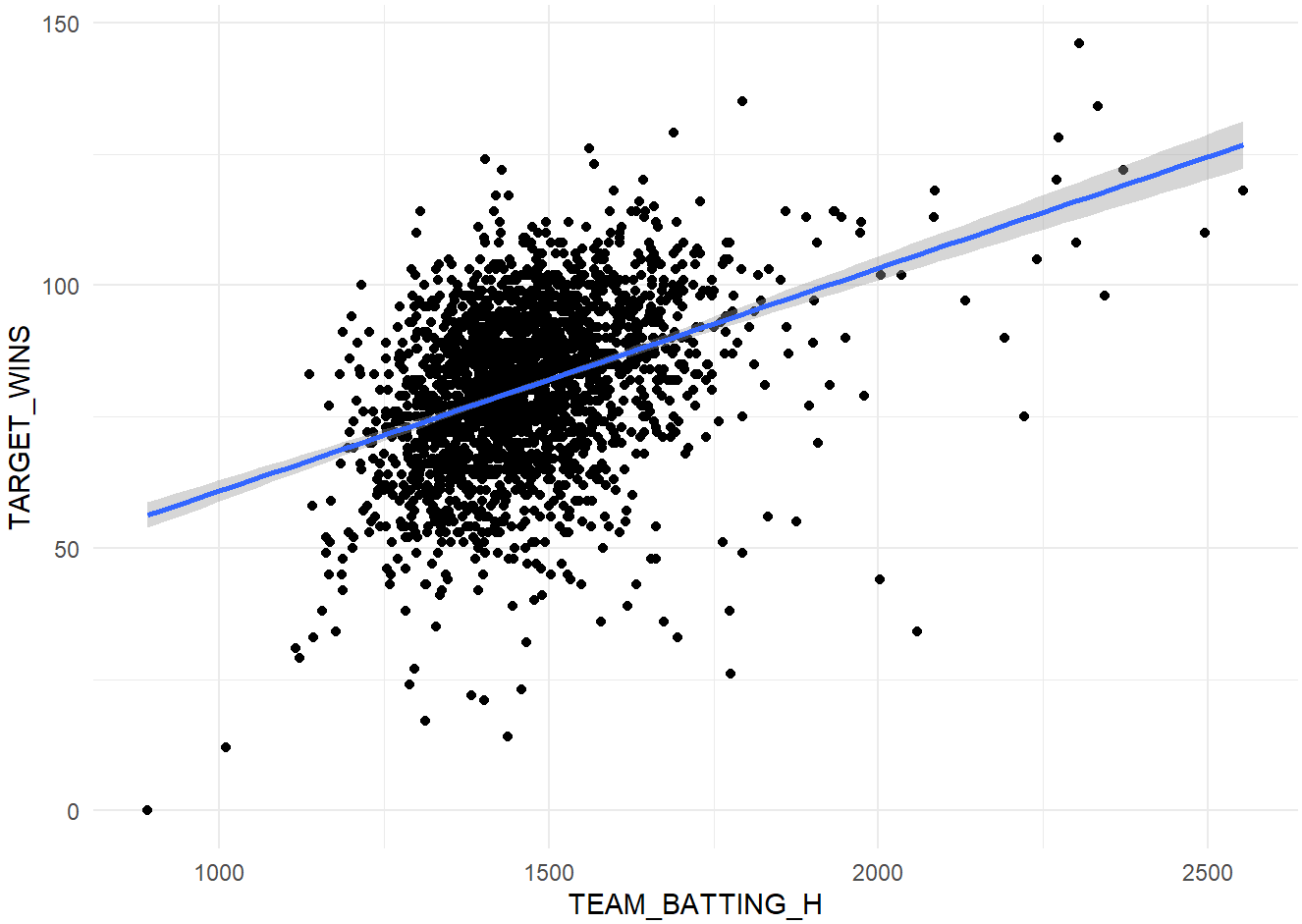

```
ggplot(model_2_df_granualar_offense, aes(x = TEAM_BATTING_BB, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(model_2_df_granualar_offense, aes(x = TEAM_BATTING_H, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



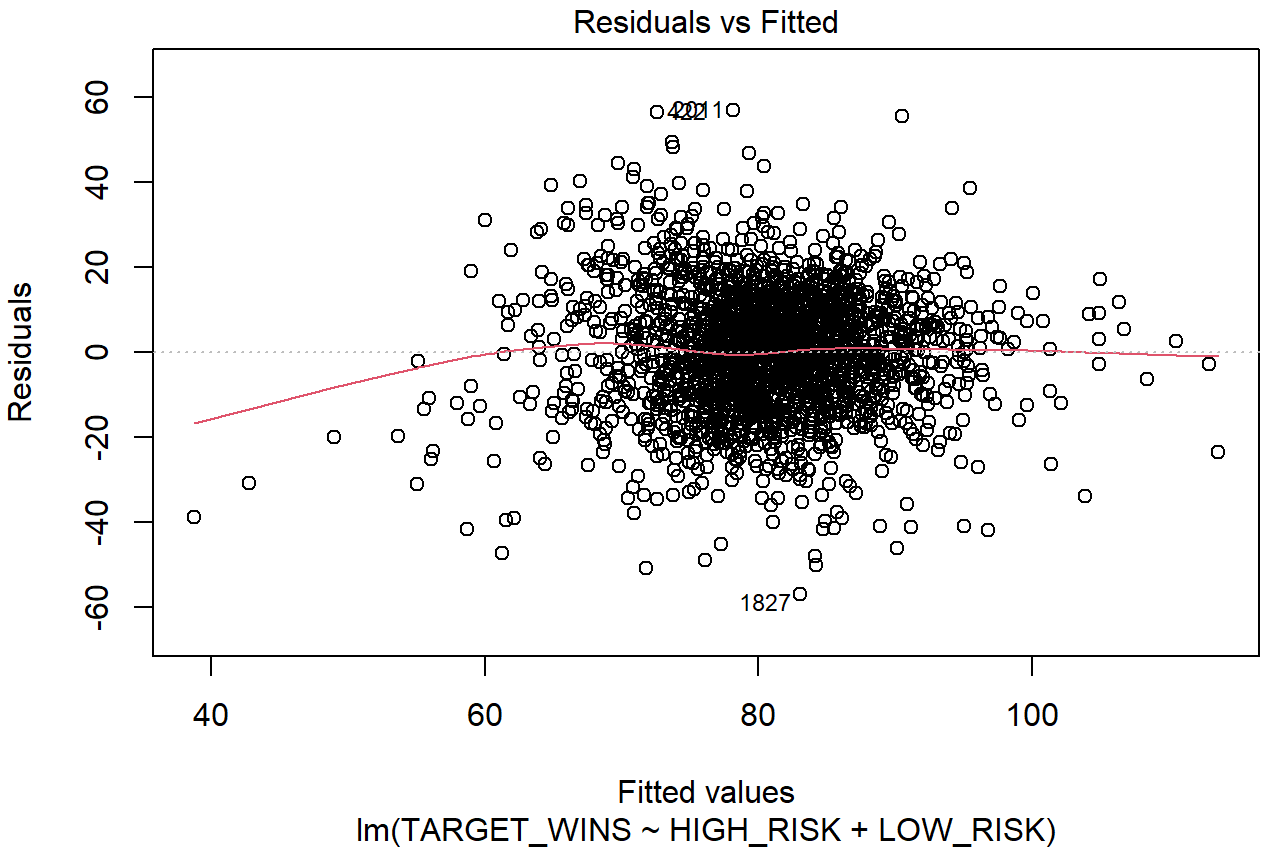

```
## MODEL 3
summary(model3)
```

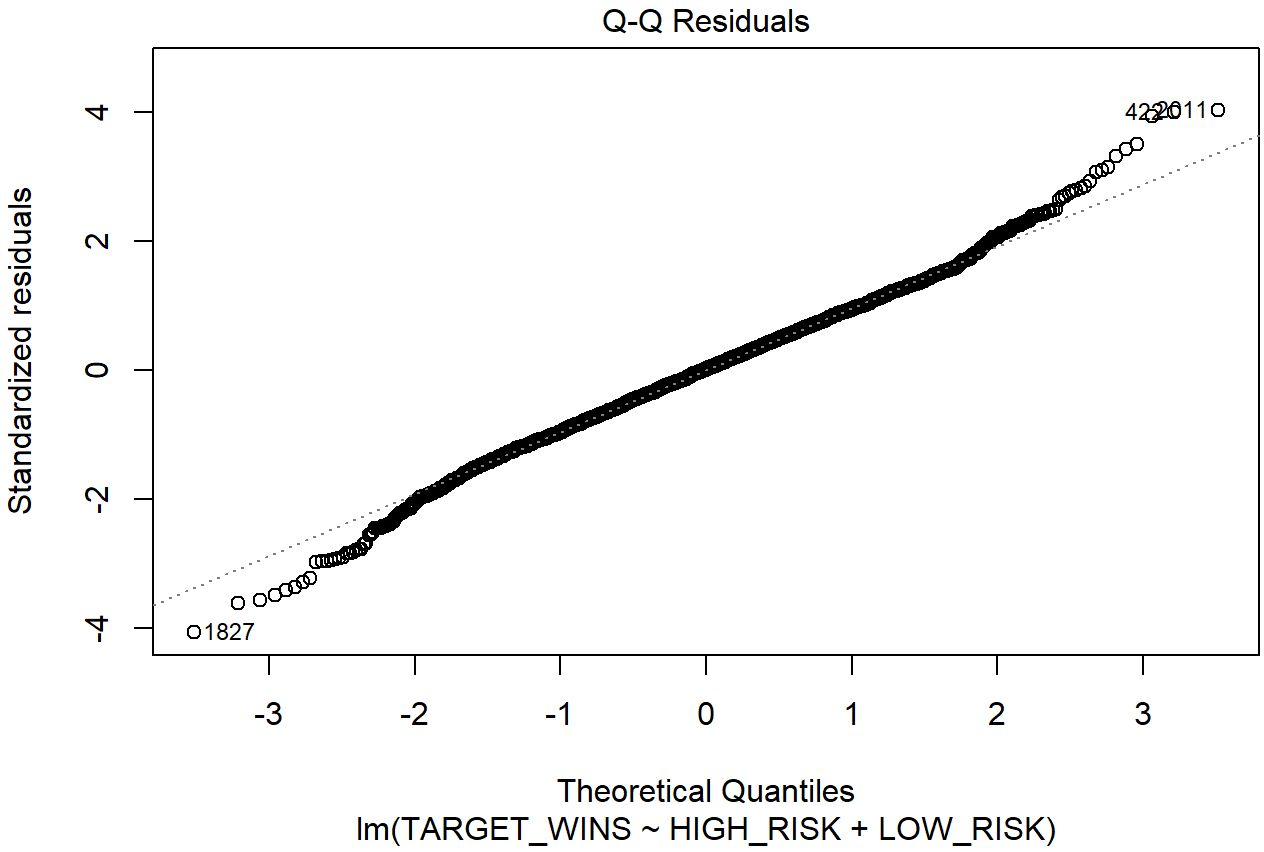
```
##
## Call:
## lm(formula = TARGET_WINS ~ HIGH_RISK + LOW_RISK, data = model_3_dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.078  -9.081   0.197   9.110  56.827
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.664325   3.207541   2.078   0.0378 *
## HIGH_RISK   0.018733   0.002900   6.459 1.29e-10 ***
## LOW_RISK    0.039120   0.002029  19.277  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.06 on 2271 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.2002
## F-statistic: 285.6 on 2 and 2271 DF,  p-value: < 2.2e-16
```

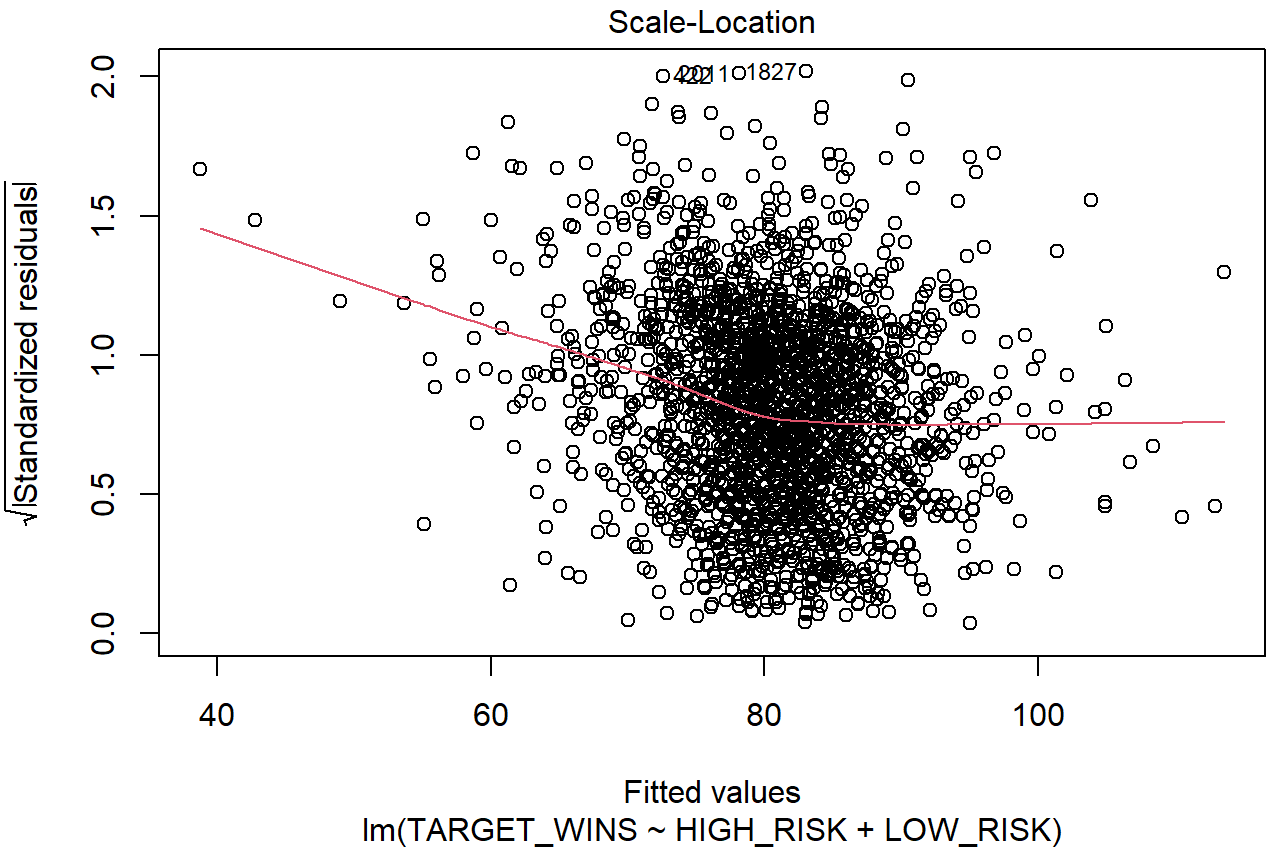
```
summary(model3_gran_low_risk)
```

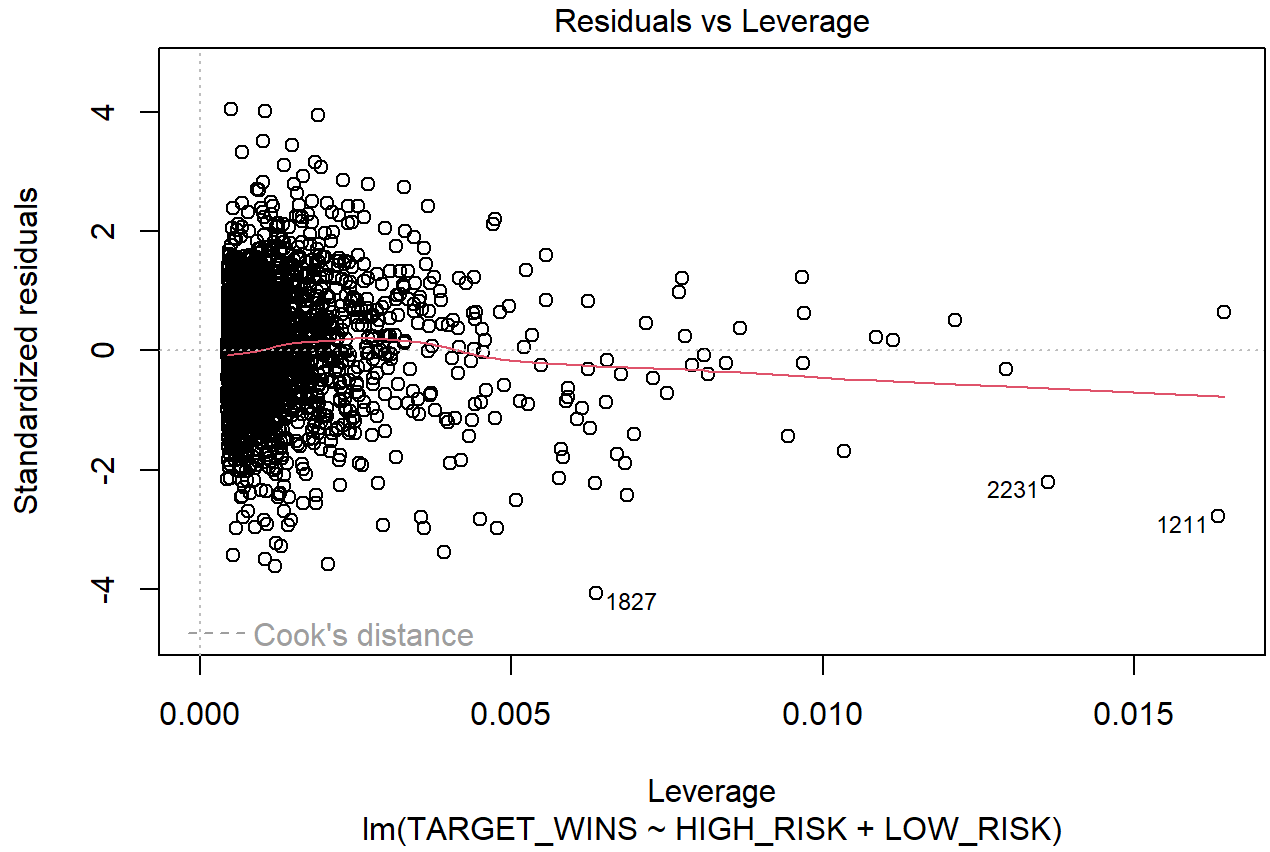
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL +
##     TEAM_BATTING_HR, data = model_3_df_low_risk_granular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.285  -8.924   0.259   9.346  52.517
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.383255   3.557273  -0.108    0.914
## TEAM_BATTING_1B          0.054282   0.002683  20.229   <2e-16 ***
## TEAM_BATTING_WALK_TOTAL  0.031993   0.002860  11.187   <2e-16 ***
## TEAM_BATTING_HR          0.067185   0.006324  10.624   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.08 on 2270 degrees of freedom
## Multiple R-squared:  0.1996, Adjusted R-squared:  0.1986
## F-statistic: 188.7 on 3 and 2270 DF,  p-value: < 2.2e-16
```

```
plot(model3) # Few leverage points influencing model, should try transform
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)
1827

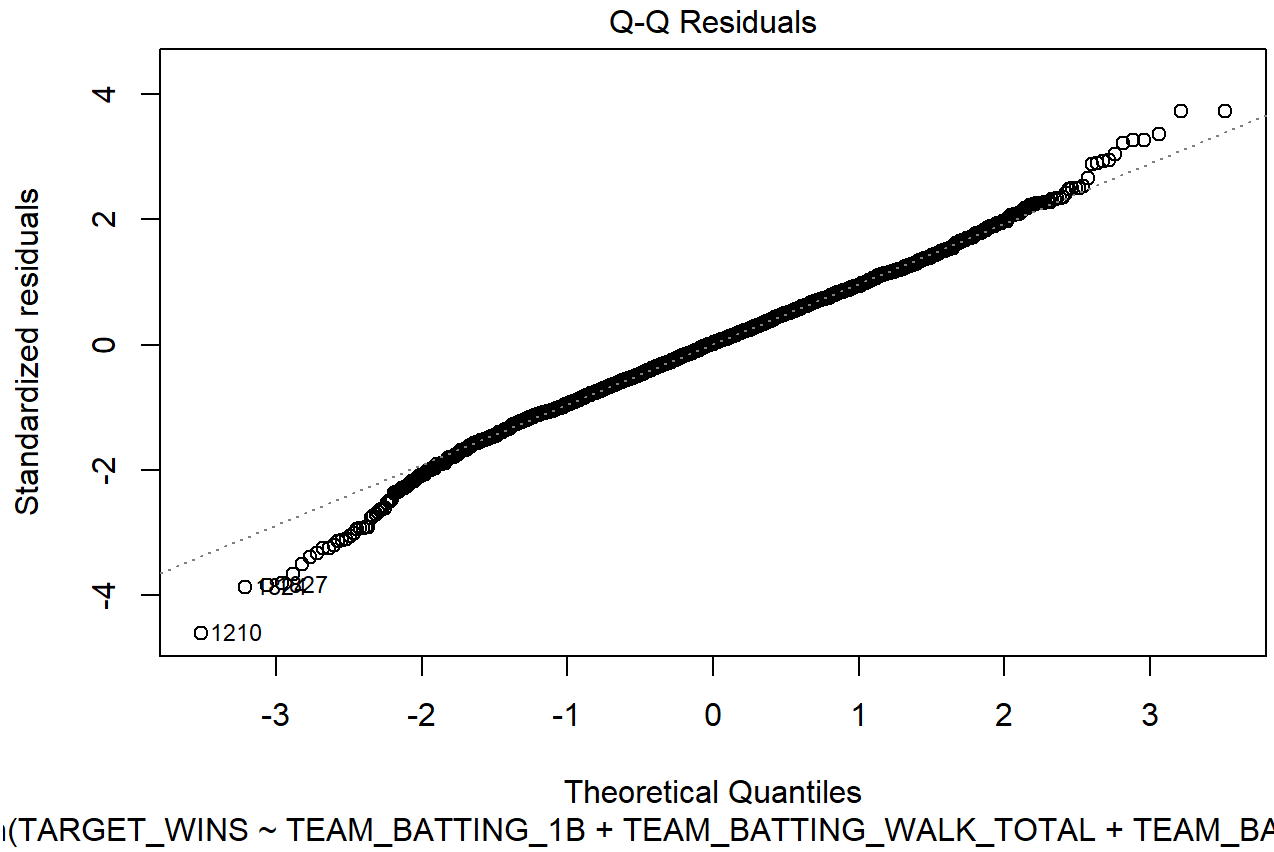

Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)
1827

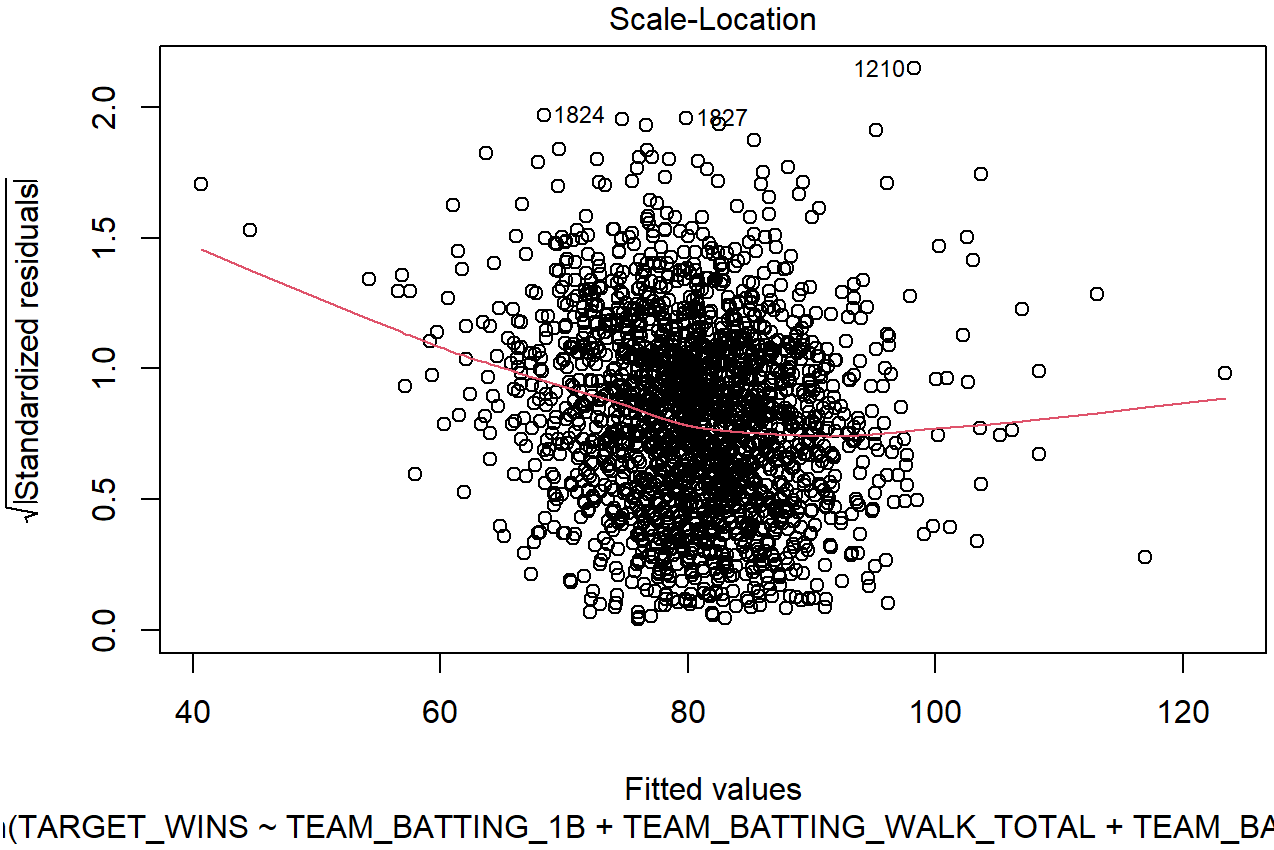

Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
40
60
80
100
Fitted values
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)

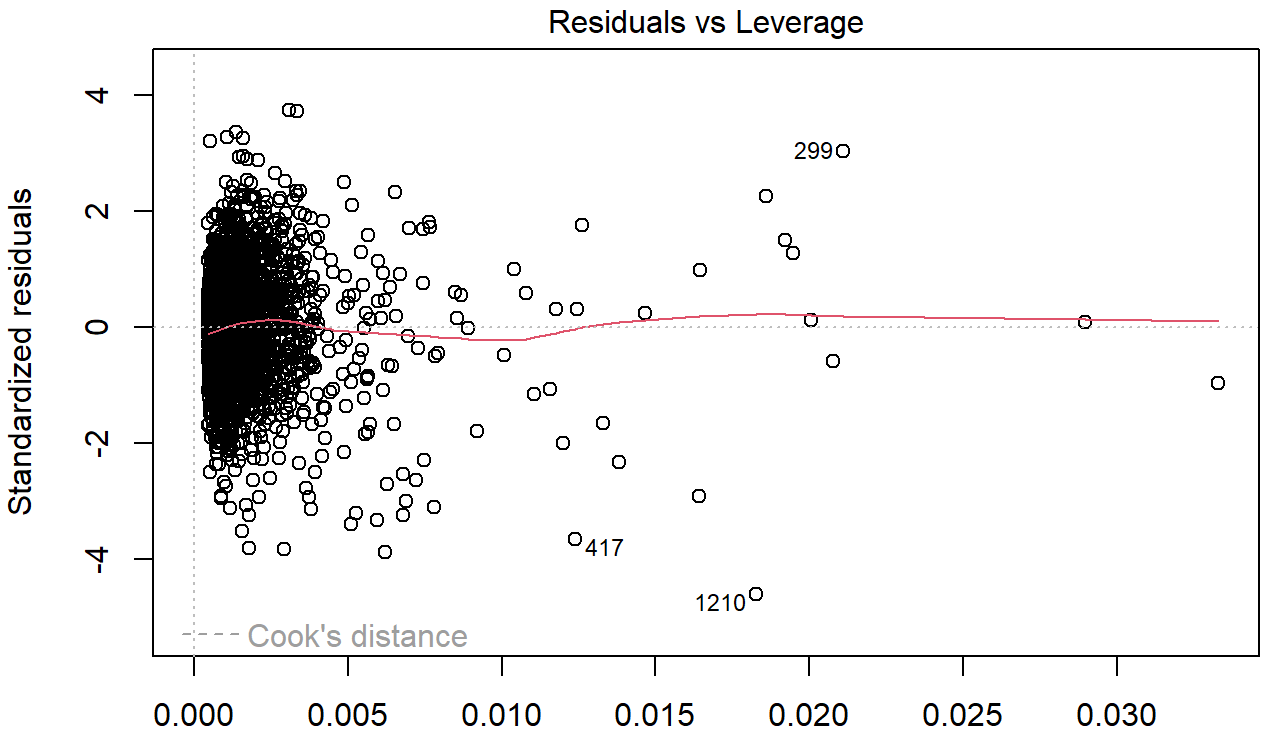

Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
0.000
0.005
0.010
0.015
Leverage
lm(TARGET_WINS ~ HIGH_RISK + LOW_RISK)
Cook's distance
1827
2231
1211

```
plot(model3_gran_low_risk)
```

(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN


Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
1210
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN

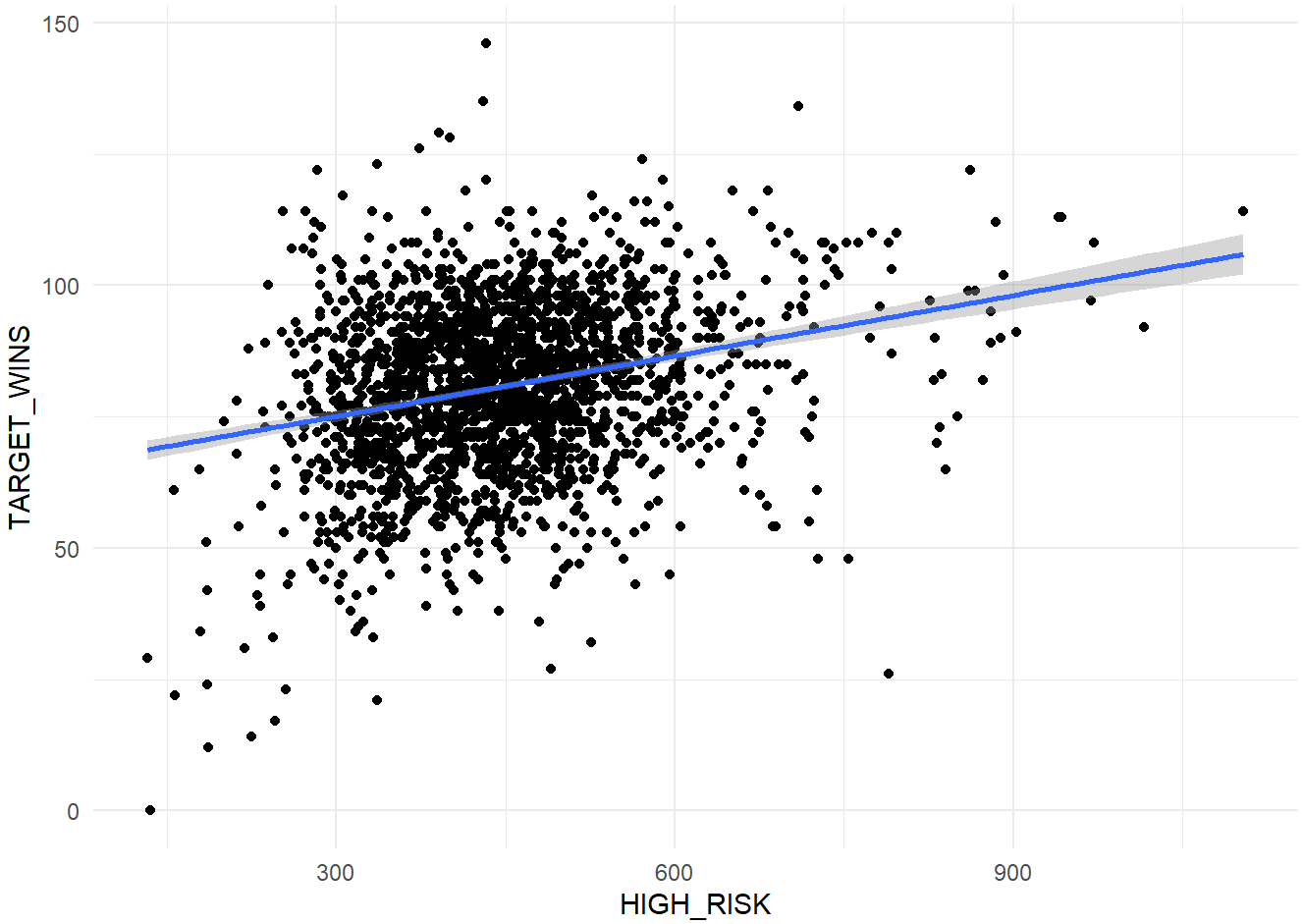

Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
1210
1824
1827
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN



lm(TARGET_WINS ~ TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL + TEAM_BATTIN

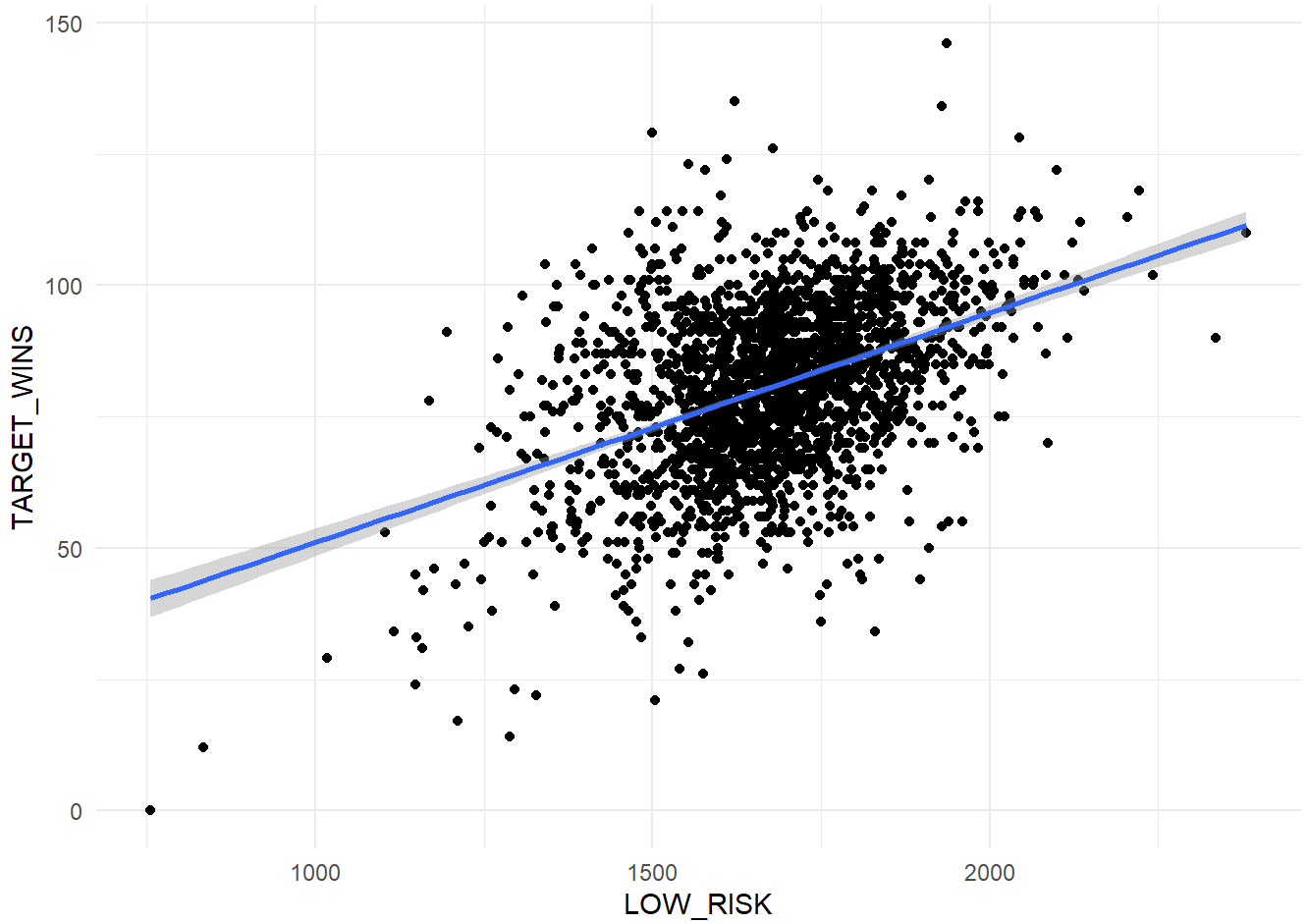
```
## PLOTTING
ggplot(model_3_dfa, aes(x = HIGH_RISK, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



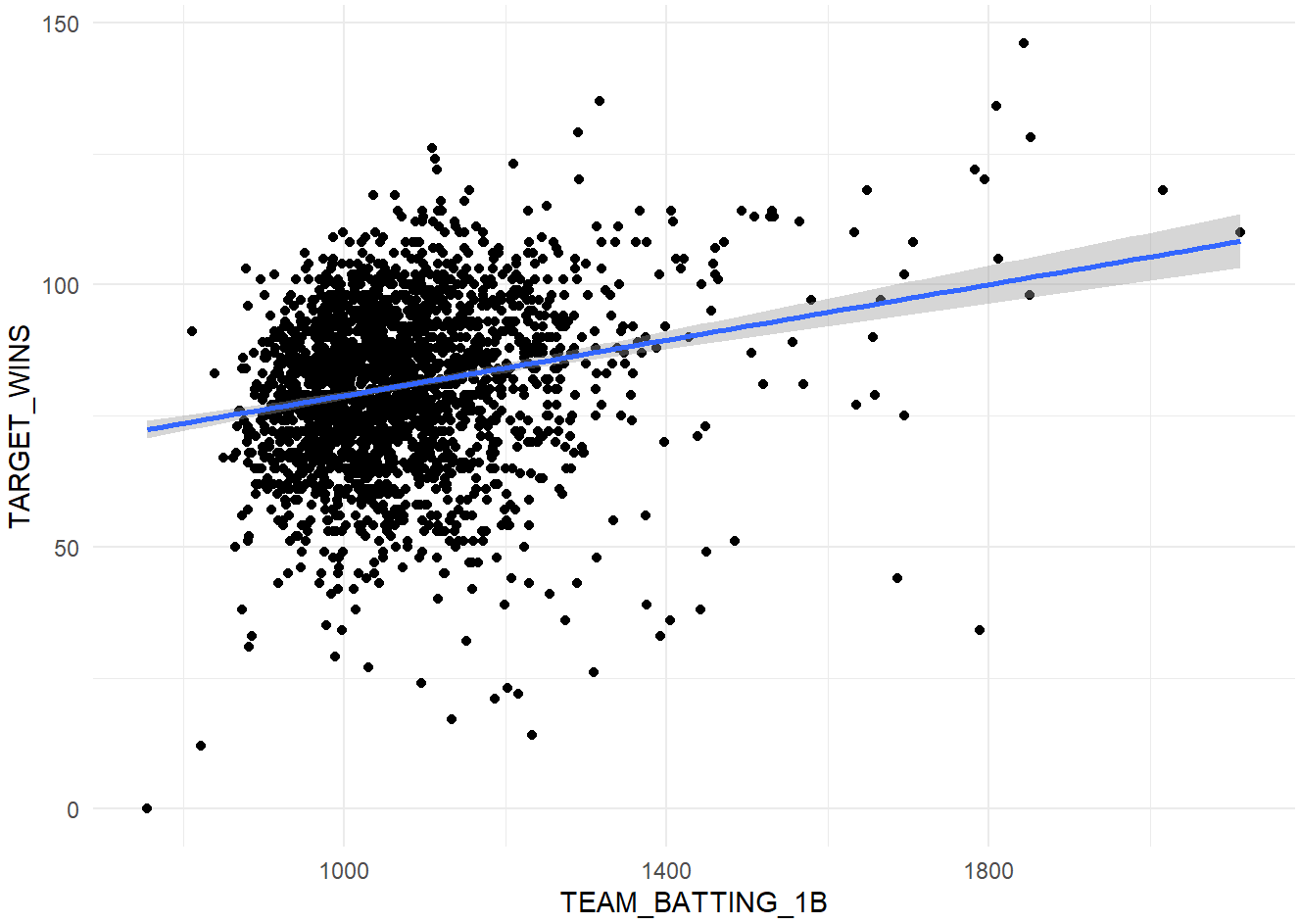

```
ggplot(model_3_dfa, aes(x = LOW_RISK, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



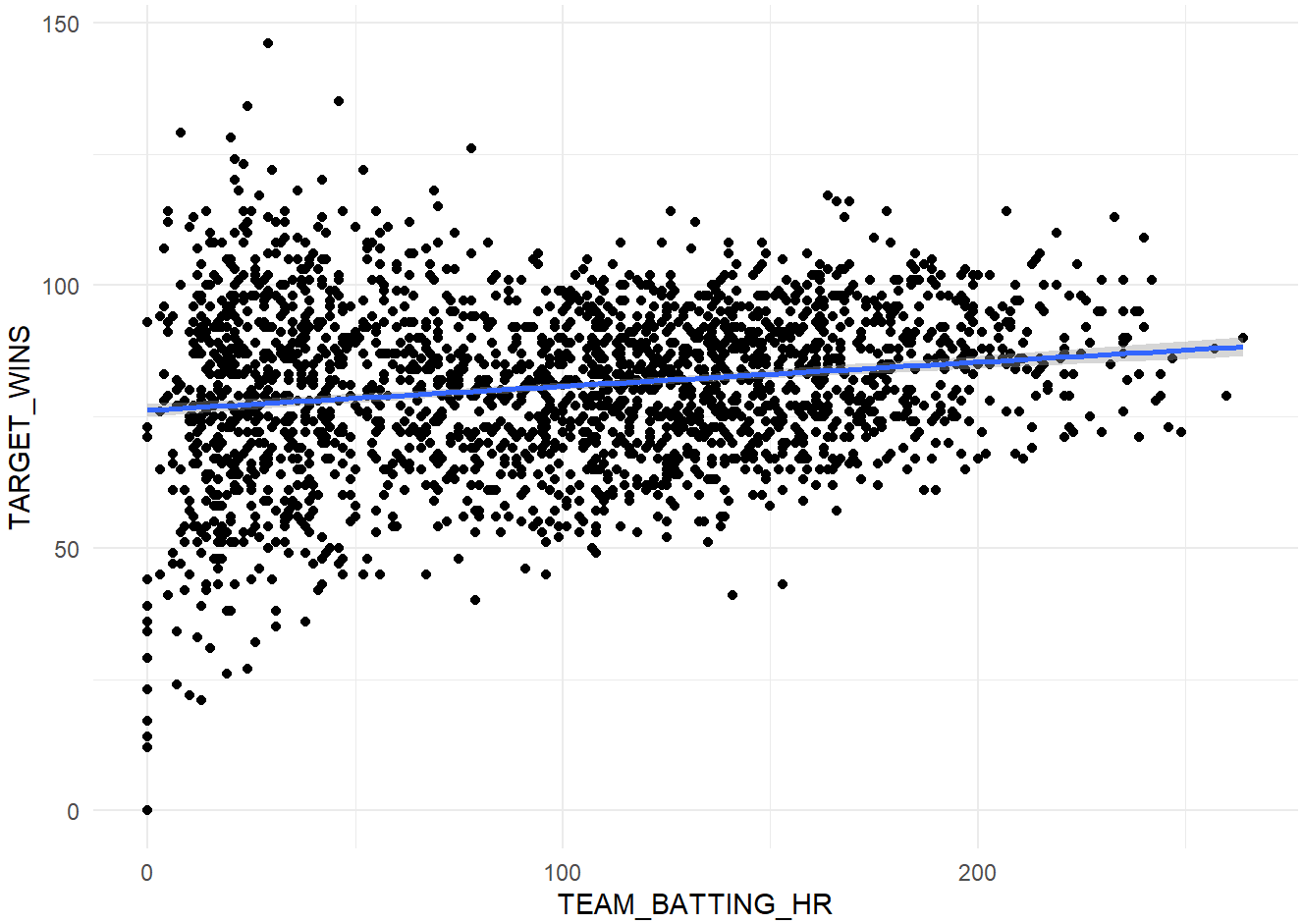

```
## --
ggplot(model_3_df_low_risk_granular, aes(x = TEAM_BATTING_1B, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



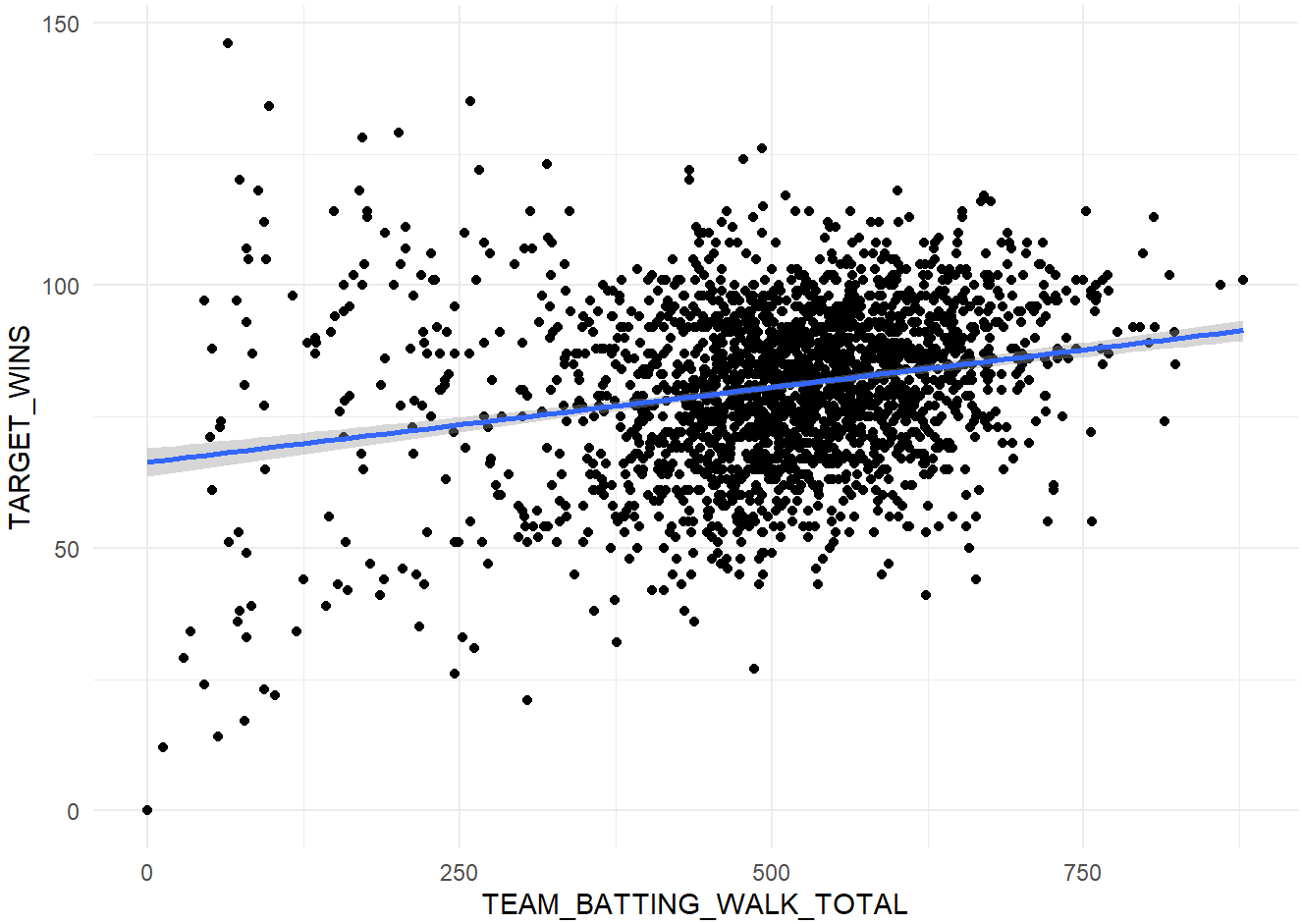

```
ggplot(model_3_df_low_risk_granular, aes(x = TEAM_BATTING_HR, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



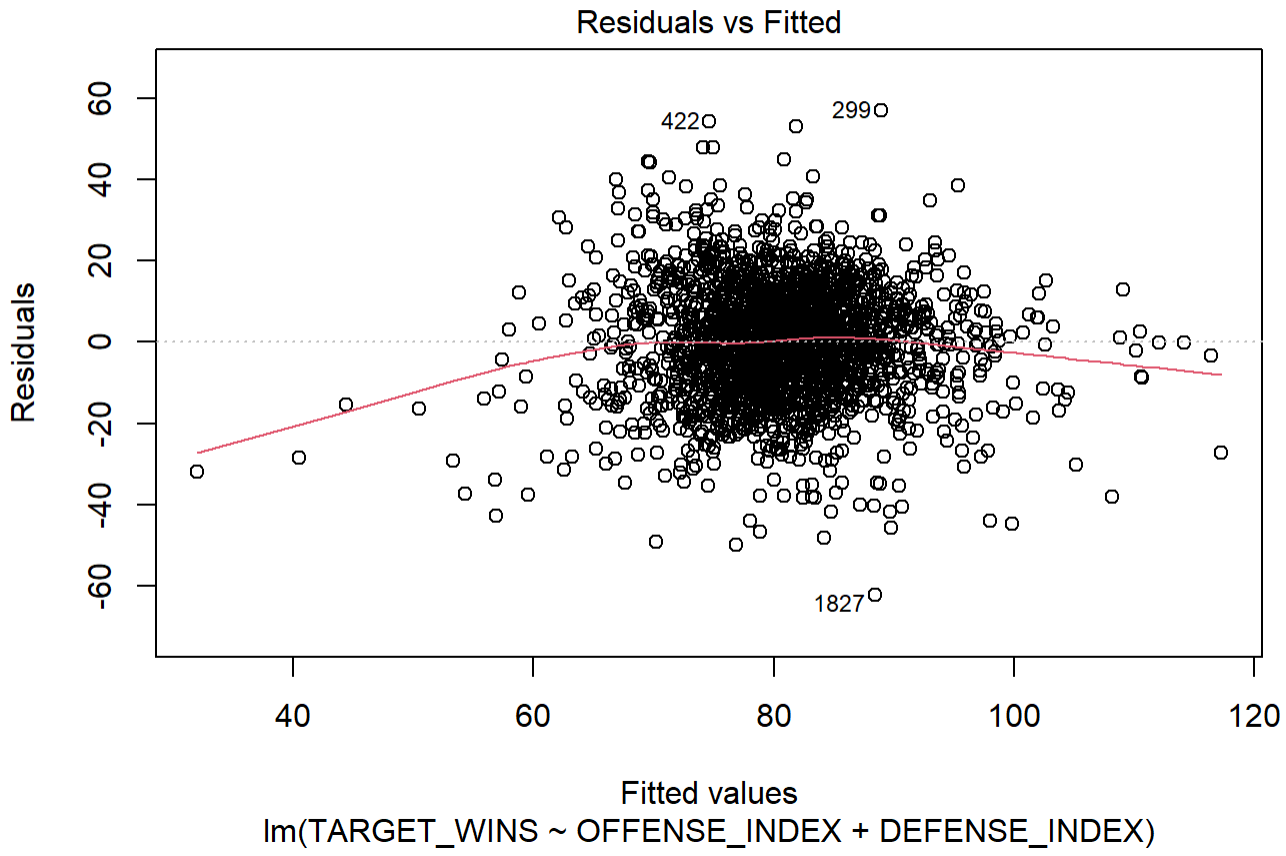

```
ggplot(model_3_df_low_risk_granular, aes(x = TEAM_BATTING_WALK_TOTAL, y = TARGET_WINS)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_minimal()
```

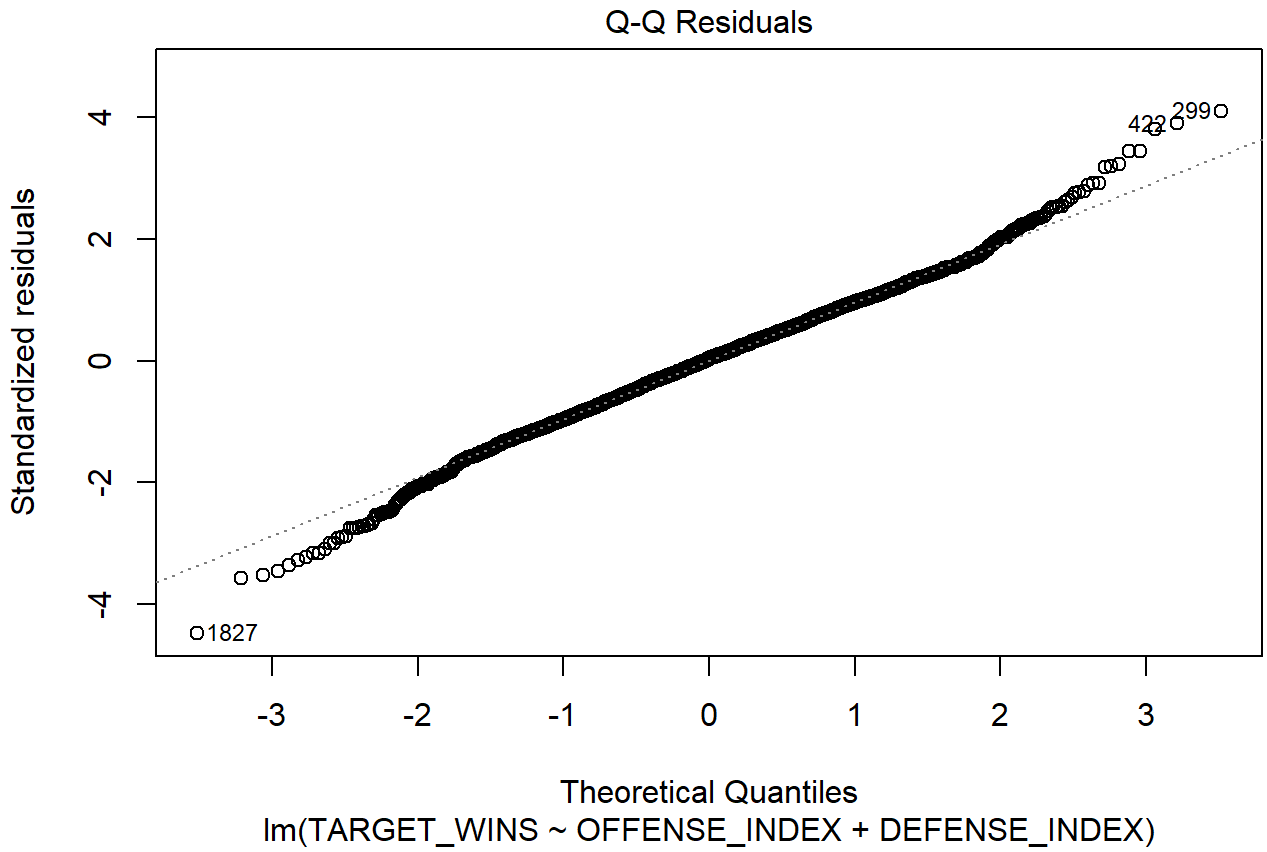
```
## `geom_smooth()` using formula = 'y ~ x'
```

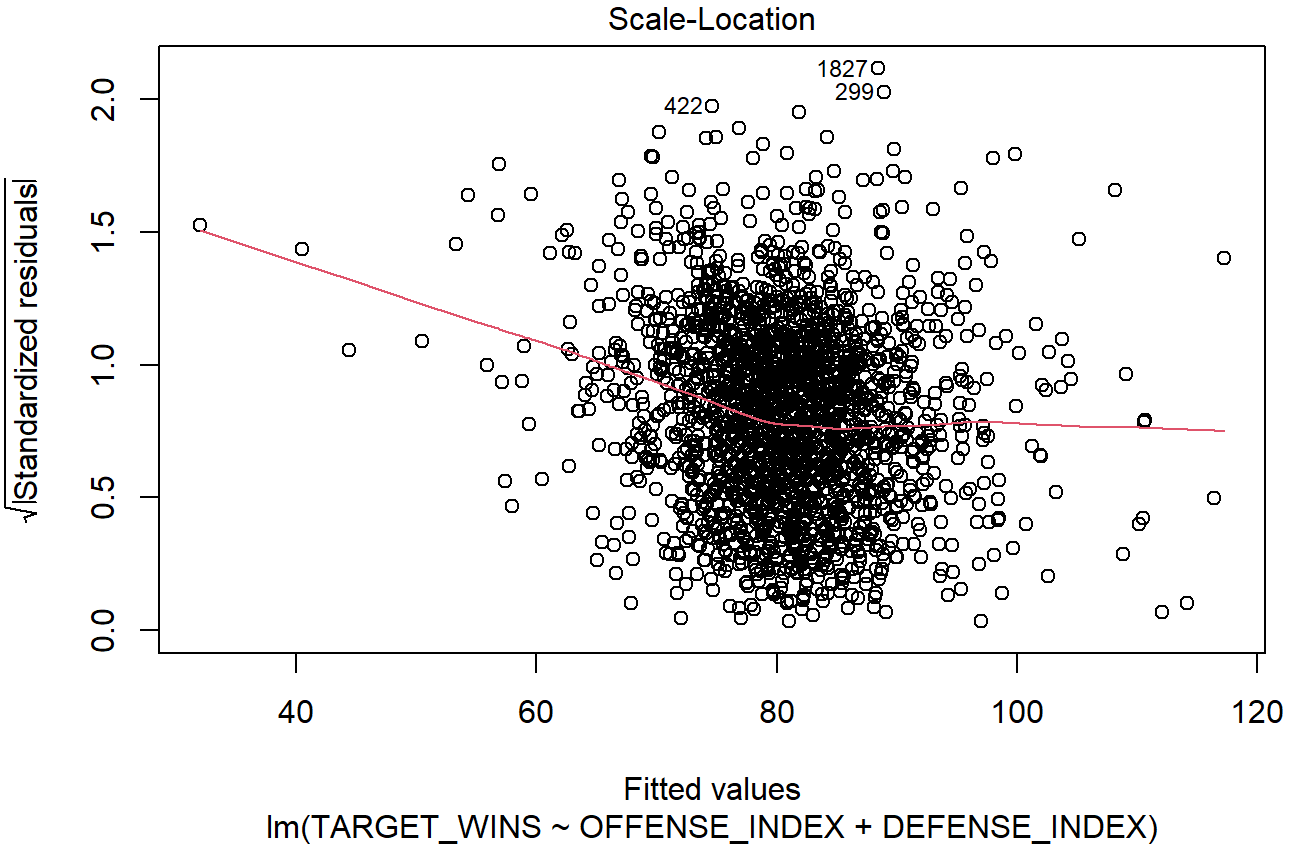


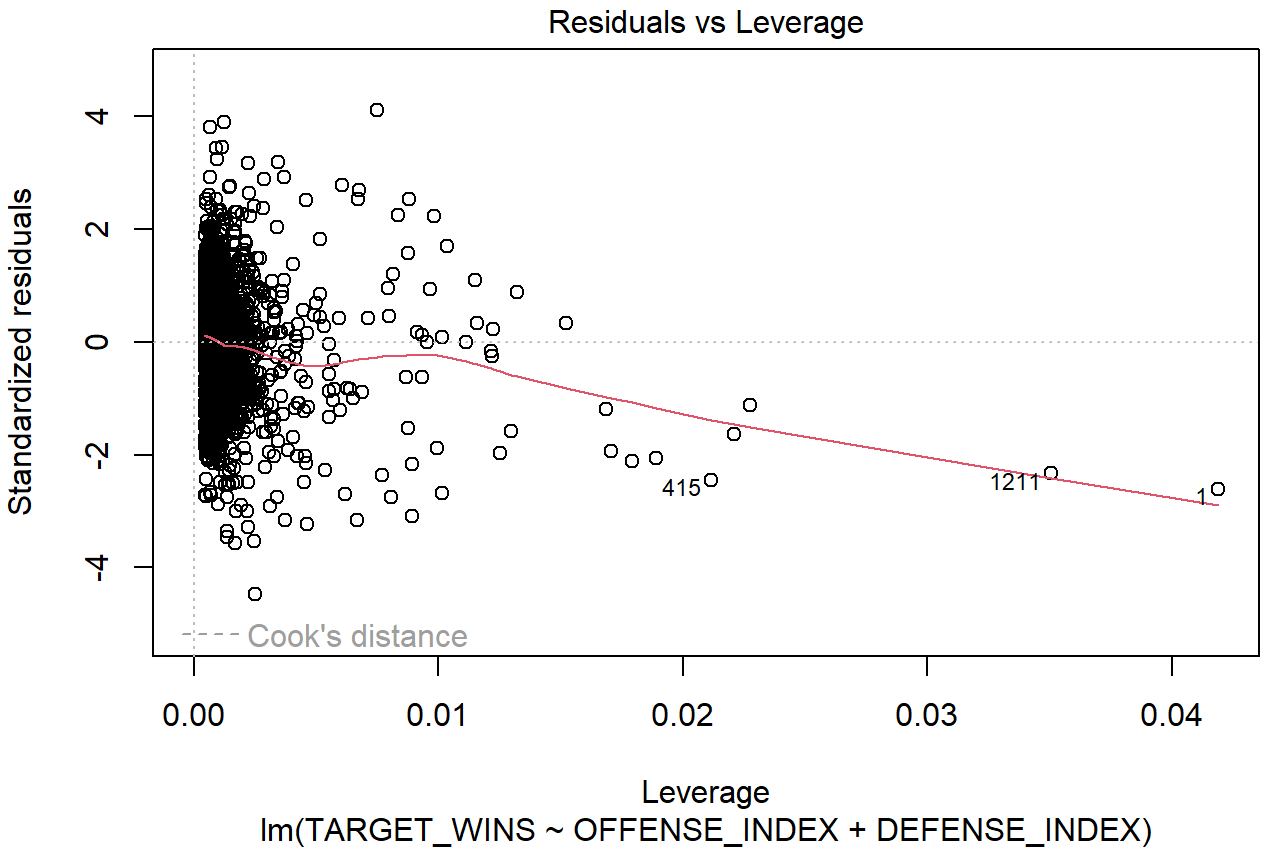
I Want to attempt to perfect model 2 and/or 3, they are decent and i think convey more in teresting information than 1.

```
### MODEL 2 Processing & Improving

## Grouping Proper Defense and Offense. Ensuring that those non-independent vars (e.g.,TEAM_B
ATTING_H and TEAM_PITCHING_H) as accounted for.
# --- OTHER INCLUSION:
#     TARGET_WINS
#     TEAM_BATTING_SO_FLAG
#     TEAM_PITCHING_SO_FLAG
#     TEAM_BATTING_BB_FLAG
#     TEAM_BATTING_HBP_FLAG
#     TEAM_FIELDING_DP_FLAG
#     TEAM_BASERUN_SB_FLAG

# --- DEFENSE INCLUSION:
      #-- TEAM_PITCHING_SO
      #-- TEAM_FIELDING_DP
      #-- TEAM_FIELDING_E

# --- OFFENSE INCLUSION:
      #-- TEAM_BATTING_H
      #-- TEAM_BATTING_BB
      #-- TEAM_BASERUN_SB
      #--TEAM_BATTING_SO
plot(model2)
```




Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
299
422
1827
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
1827
299
422
40
60
80
100
120
Fitted values
lm(TARGET_WINS ~ OFFENSE_INDEX + DEFENSE_INDEX)



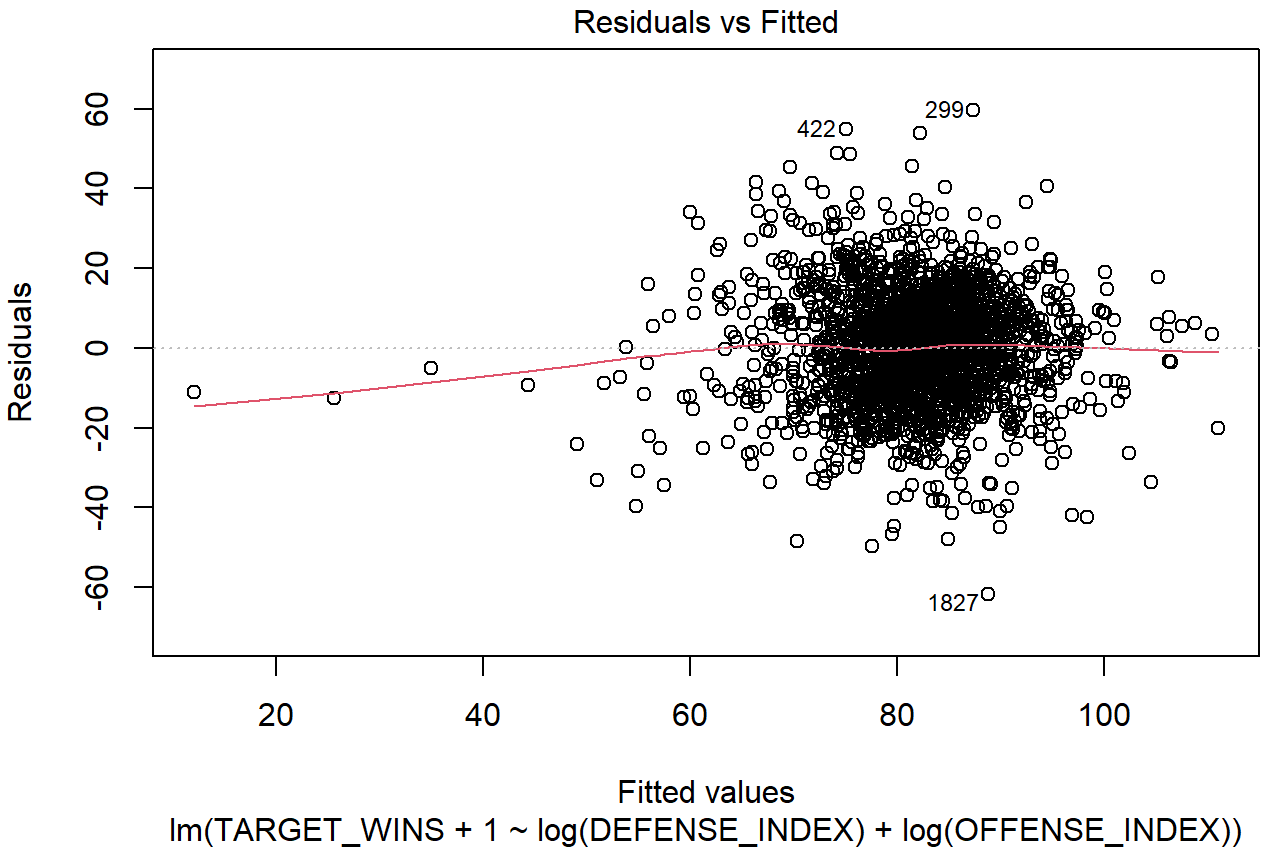

```
colnames(moneyball_train_imputed)
```

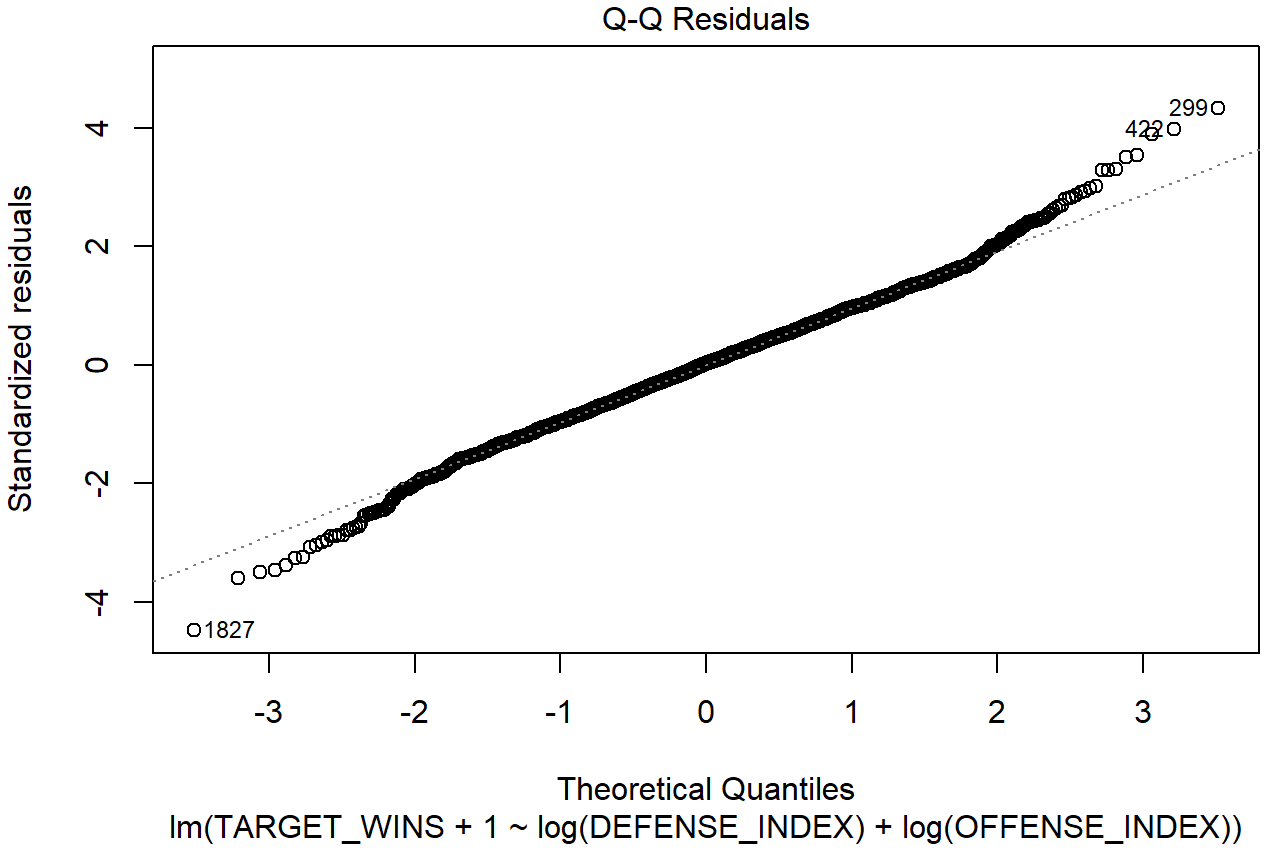
```
##  [1] "INDEX"                      "TARGET_WINS"
##  [3] "TEAM_BATTING_H"             "TEAM_BATTING_2B"
##  [5] "TEAM_BATTING_3B"            "TEAM_BATTING_HR"
##  [7] "TEAM_BATTING_BB"            "TEAM_BATTING_SO"
##  [9] "TEAM_BASERUN_SB"            "TEAM_PITCHING_H"
## [11] "TEAM_PITCHING_HR"           "TEAM_PITCHING_BB"
## [13] "TEAM_PITCHING_SO"           "TEAM_FIELDING_E"
## [15] "TEAM_FIELDING_DP"           "TEAM_BATTING_SO_FLAG"
## [17] "TEAM_PITCHING_SO_FLAG"      "TEAM_BASERUN_SB_FLAG"
## [19] "TEAM_BASERUN_CS_FLAG"       "TEAM_BATTING_BB_FLAG"
## [21] "TEAM_BATTING_HBP_FLAG"      "TEAM_FIELDING_DP_FLAG"
## [23] "TEAM_BASERUN_STEAL_ATTEMPTS" "TEAM_BATTING_WALK_TOTAL"
```

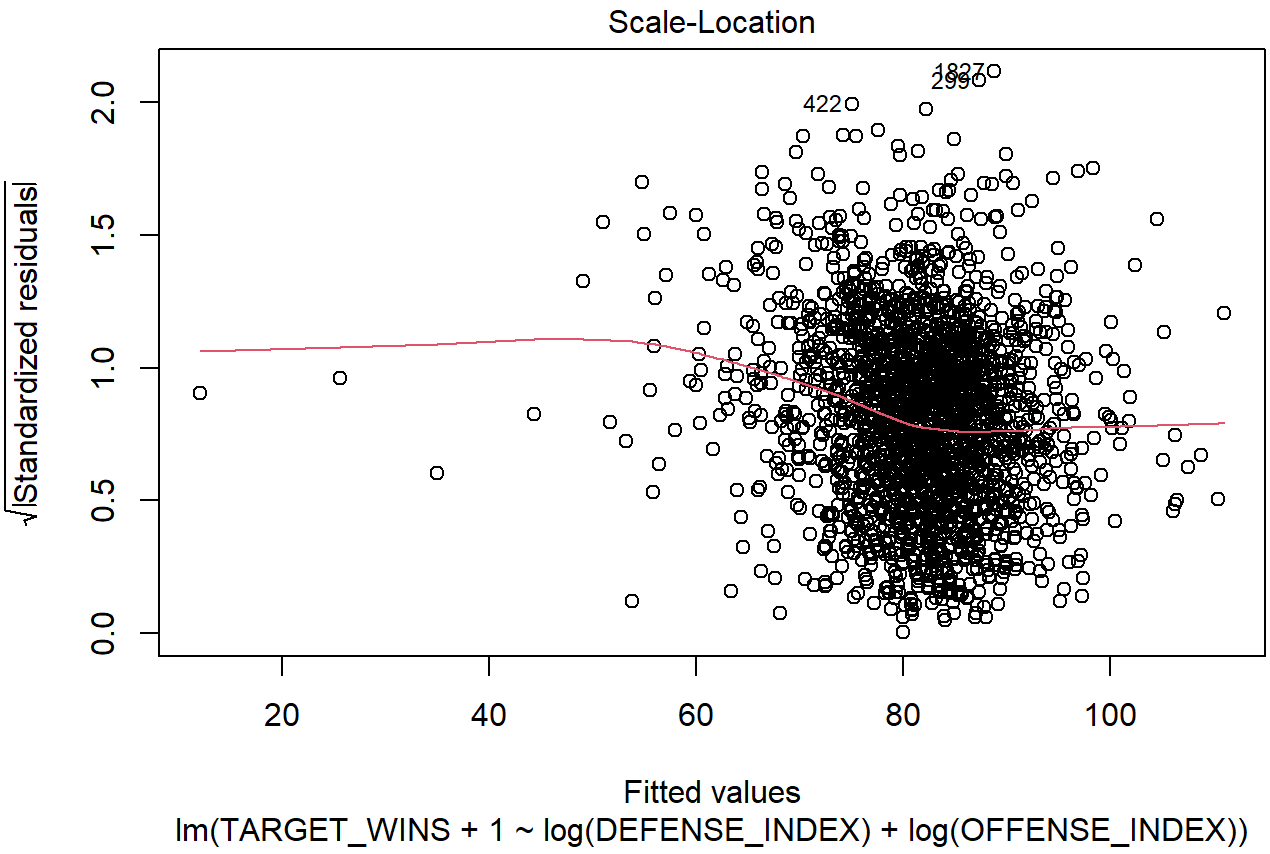
```
## Custom Aggregate with Transform, Making Better
model_2_dfa_transform <- moneyball_train_imputed |>
  dplyr::select(TARGET_WINS,TEAM_BATTING_SO_FLAG,TEAM_PITCHING_SO_FLAG,TEAM_BATTING_BB_FLAG,T
EAM_BATTING_HBP_FLAG,TEAM_FIELDING_DP_FLAG,TEAM_BASERUN_SB_FLAG,
         TEAM_PITCHING_SO,TEAM_FIELDING_DP,TEAM_FIELDING_E,TEAM_BATTING_H,TEAM_BATTING_WALK_T
OTAL,TEAM_BATTING_BB,TEAM_BASERUN_SB,TEAM_BATTING_SO) |>
  mutate( DEFENSE_INDEX = ((TEAM_PITCHING_SO + TEAM_FIELDING_DP) / TEAM_FIELDING_E),## Shifti
ng this to keep things positive, but maintain the error influence
         OFFENSE_INDEX=(TEAM_BATTING_H+TEAM_BATTING_BB+TEAM_BASERUN_SB)) |>
  dplyr::select(-TEAM_PITCHING_SO,-TEAM_FIELDING_DP,-TEAM_FIELDING_E,-TEAM_BATTING_H,-TEAM_BA
TTING_WALK_TOTAL,-TEAM_BATTING_BB,-TEAM_BASERUN_SB)

## Adding 1 to Target Wins Ver to use Box Cox
model2_2 <- lm(TARGET_WINS+1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX), data=model_2_dfa_tran
sform)
print(summary(model2_2))
```

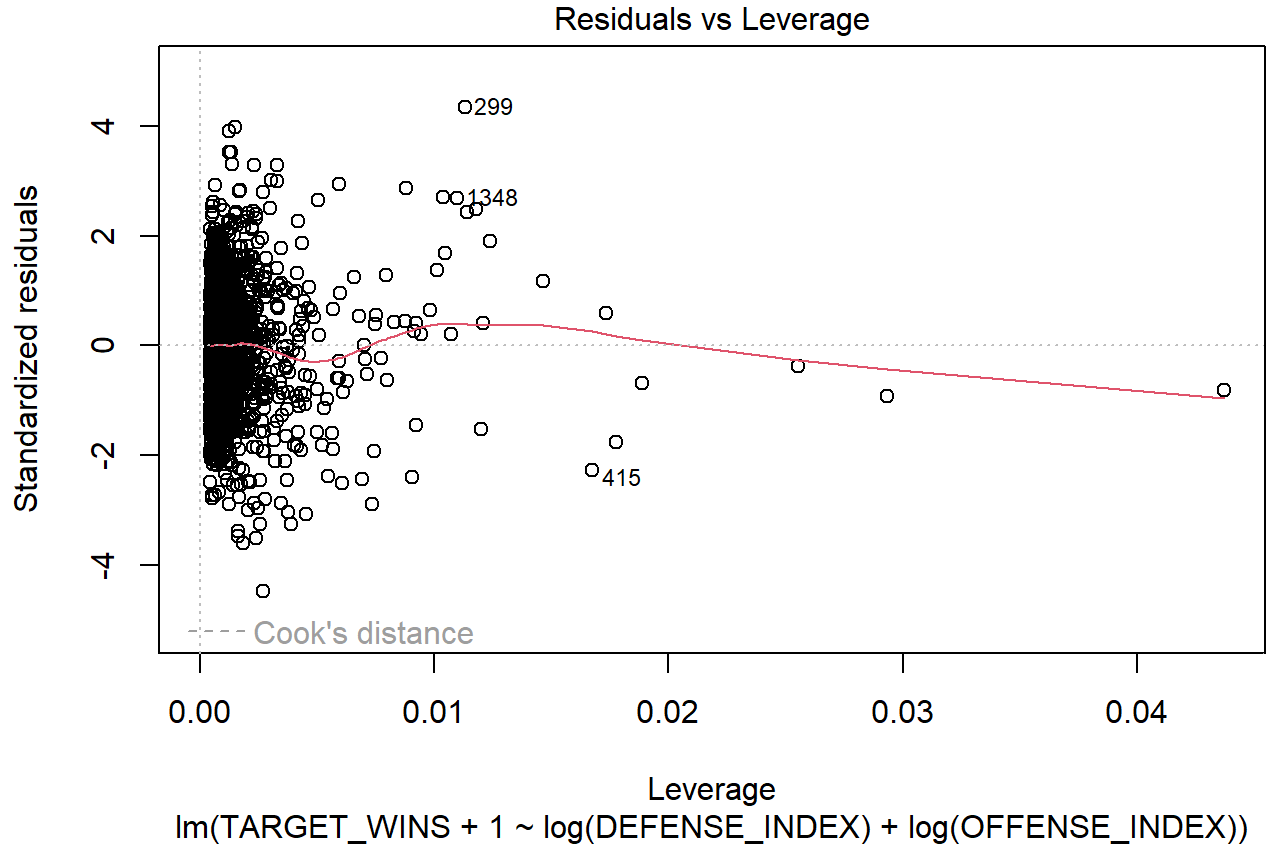
```
##
## Call:
## lm(formula = TARGET_WINS + 1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX),
##     data = model_2_dfa_transform)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.831  -9.038   0.316   8.871  59.607
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -489.8544    22.3094 -21.957  < 2e-16 ***
## log(DEFENSE_INDEX)    1.5386     0.3449   4.462 8.54e-06 ***
## log(OFFENSE_INDEX)   74.5013     2.9183  25.529  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.82 on 2271 degrees of freedom
## Multiple R-squared:  0.2281, Adjusted R-squared:  0.2275
## F-statistic: 335.6 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model2_2)
```


Residuals vs Fitted
Residuals
60
40
20
0
-20
-40
-60
422
299
1827
20
40
60
80
100
Fitted values
lm(TARGET_WINS + 1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX))

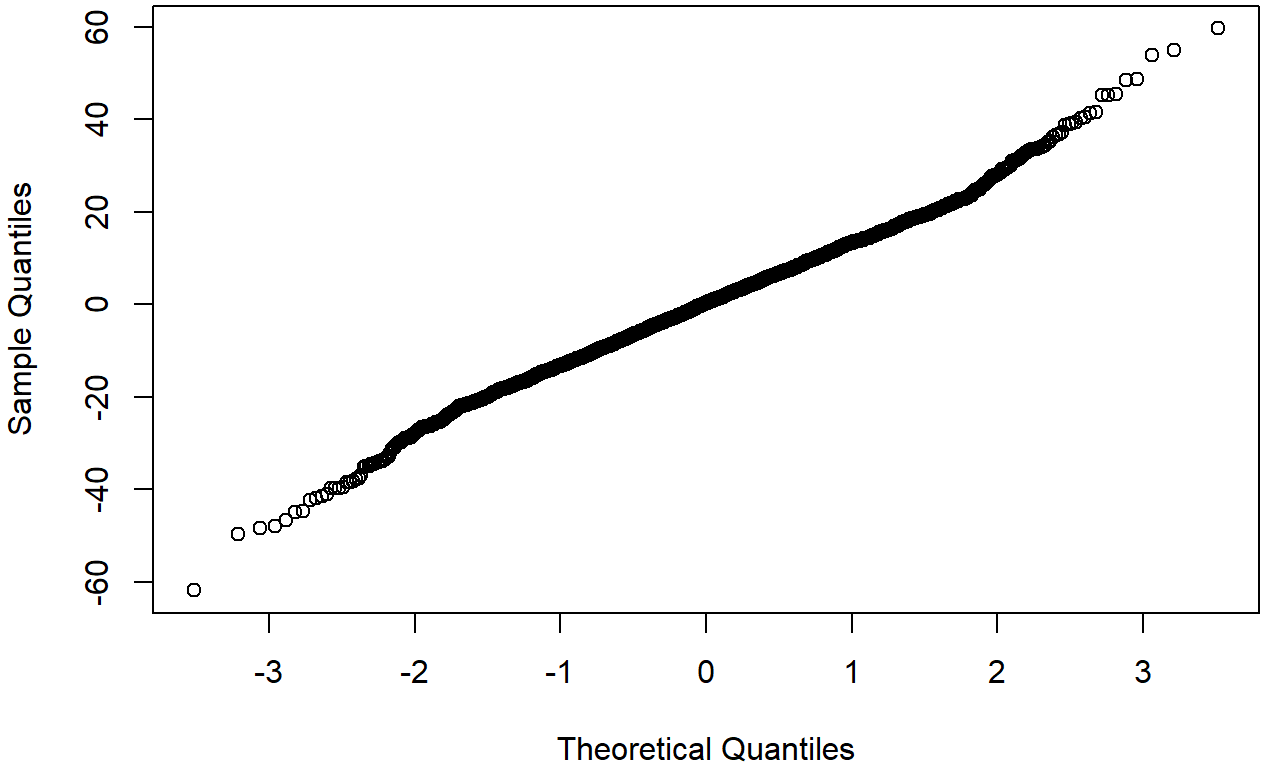

Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
299
422
1827
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS + 1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX))


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
20
40
60
80
100
1827
299
422
Fitted values
lm(TARGET_WINS + 1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX))


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
0.00
0.01
0.02
0.03
0.04
299
1348
415
Cook's distance
Leverage
lm(TARGET_WINS + 1 ~ log(DEFENSE_INDEX) + log(OFFENSE_INDEX))

```
hist(model2_2$resid)
```

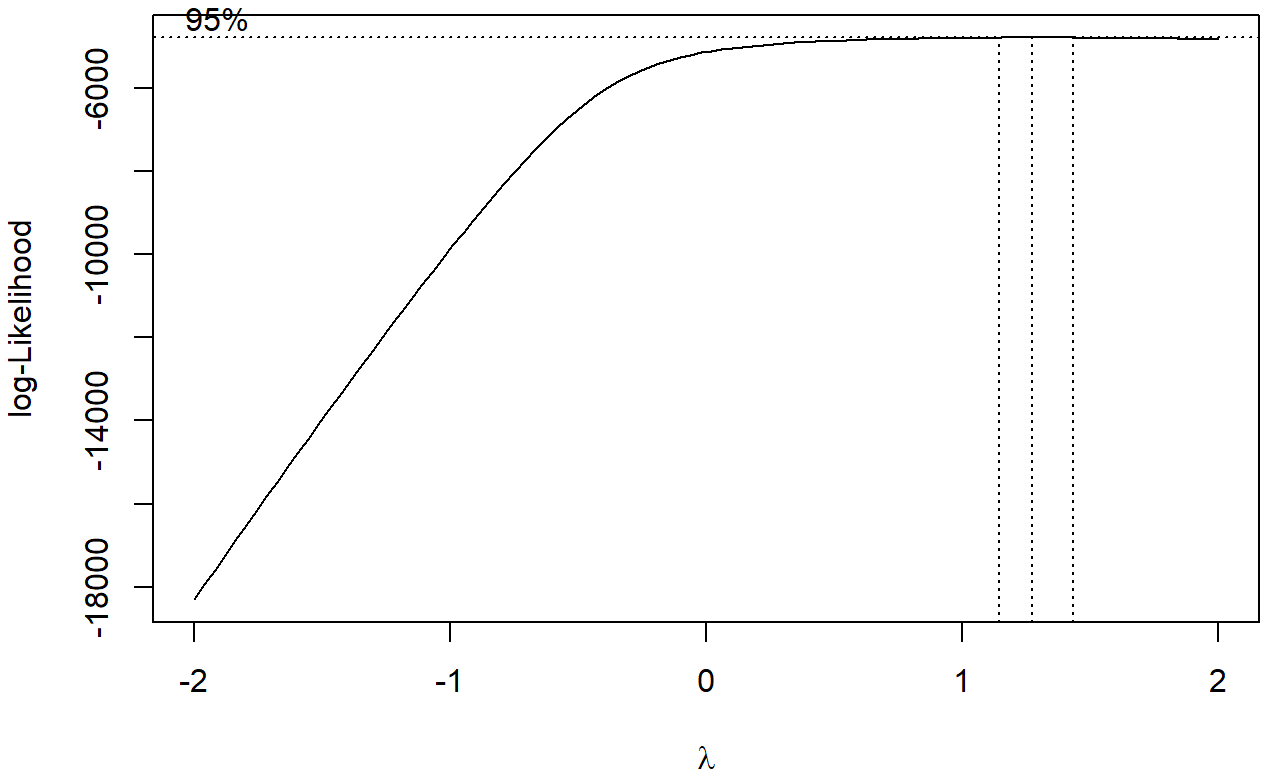

```
qqnorm(model2_2$resid)
```




```
skewness(model2_2$resid) #-0.012
```

```
## [1] -0.01202184
```

```
## No Change X Vars: r^2 21.7% | F-score 317
## Log Transform: r^2 22.75% | F-score 335 [THIS MODEL IS BEST BY SMALL AMOUNT]
## SqRt: r^2 22.43% | F-score 329
## Squared: r^2 21.17% | F-score 317
## Cubed: r^2 21.77 | F-score 317

#sort(unique(moneyball_train_imputed$TEAM_BATTING_SO ))
#sort(unique(model_2_dfa_transform$DEFENSE_INDEX))
#sort(unique(model_2_dfa_transform$OFFENSE_INDEX))

bcx_model2_2<- boxcox(model2_2)
```



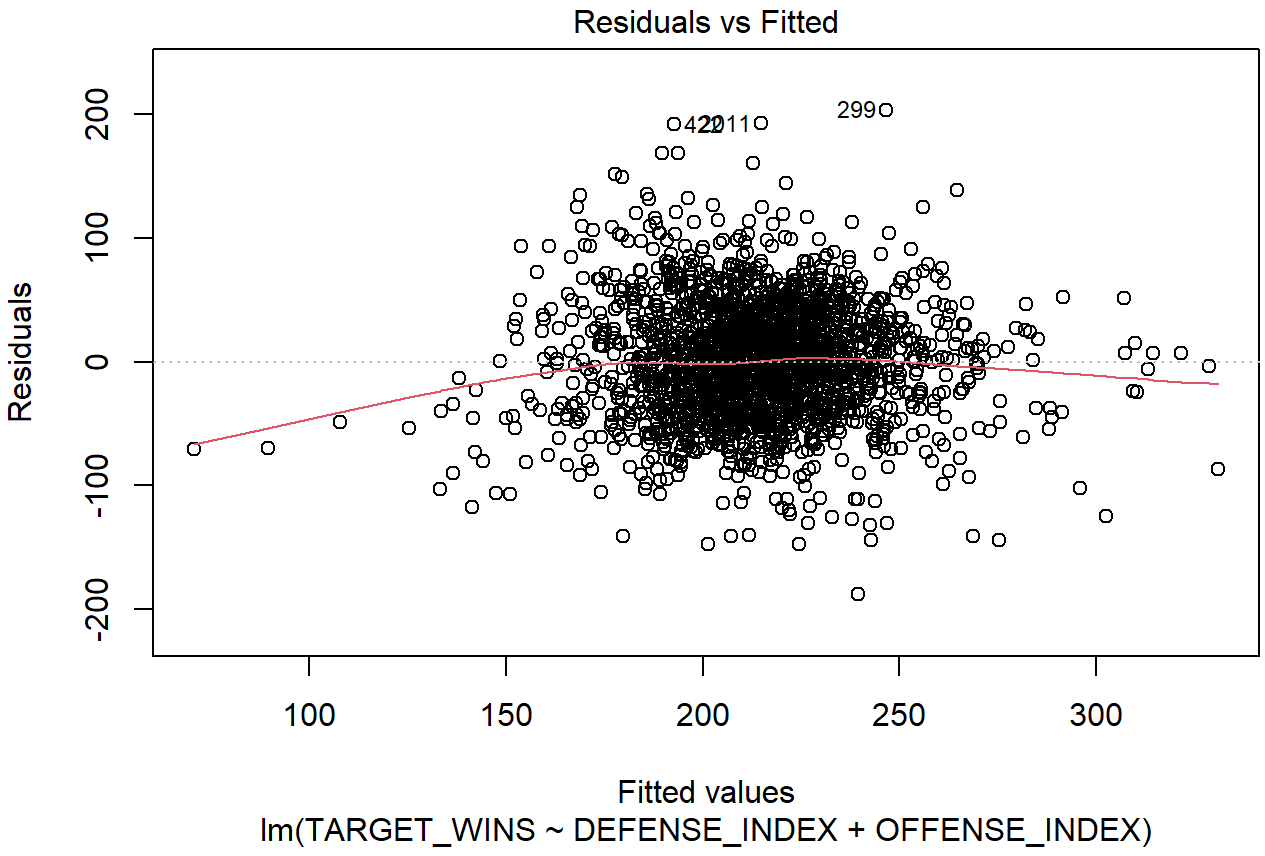

```
optimal_lambda <- bcx_model2_2$x[which.max(bcx_model2_2$y)]
print(optimal_lambda) ## 1.27
```

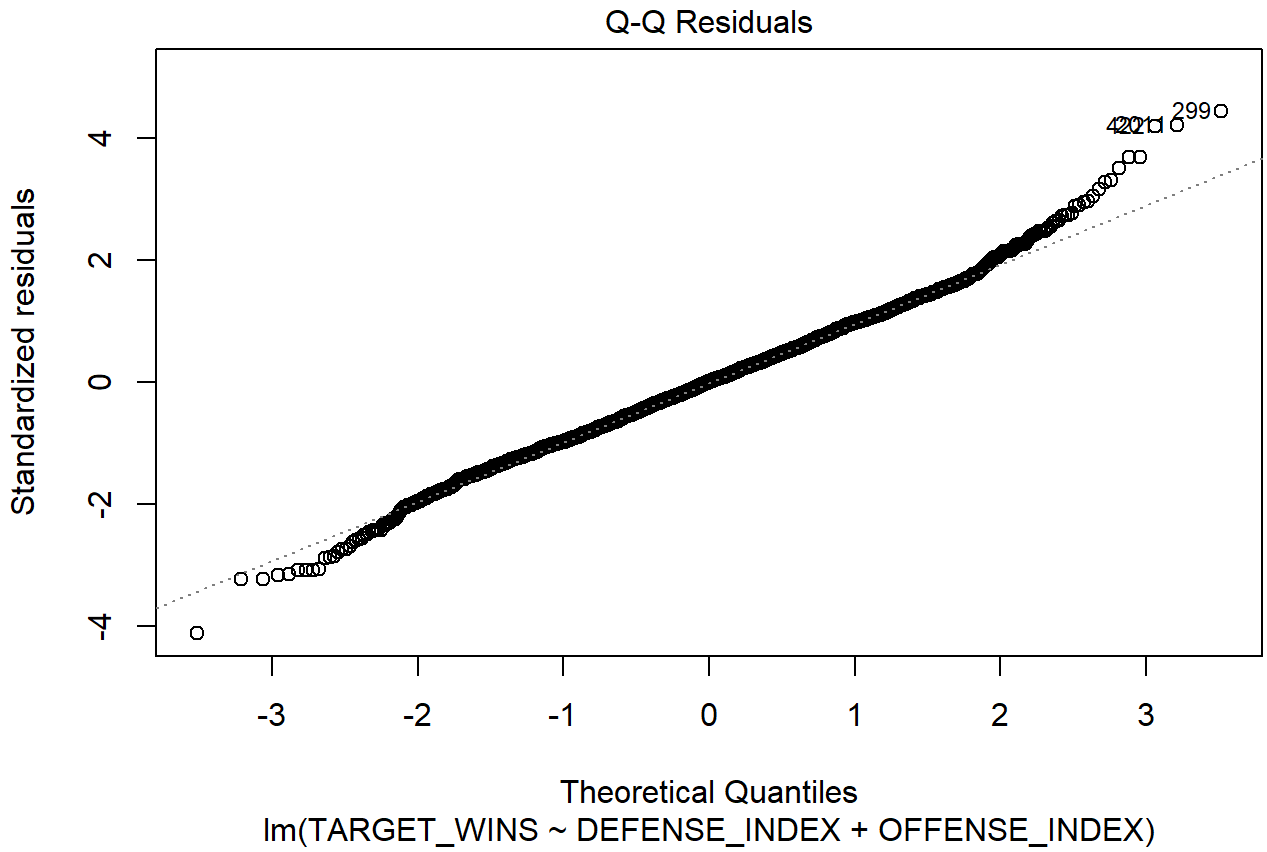
```
## [1] 1.272727
```

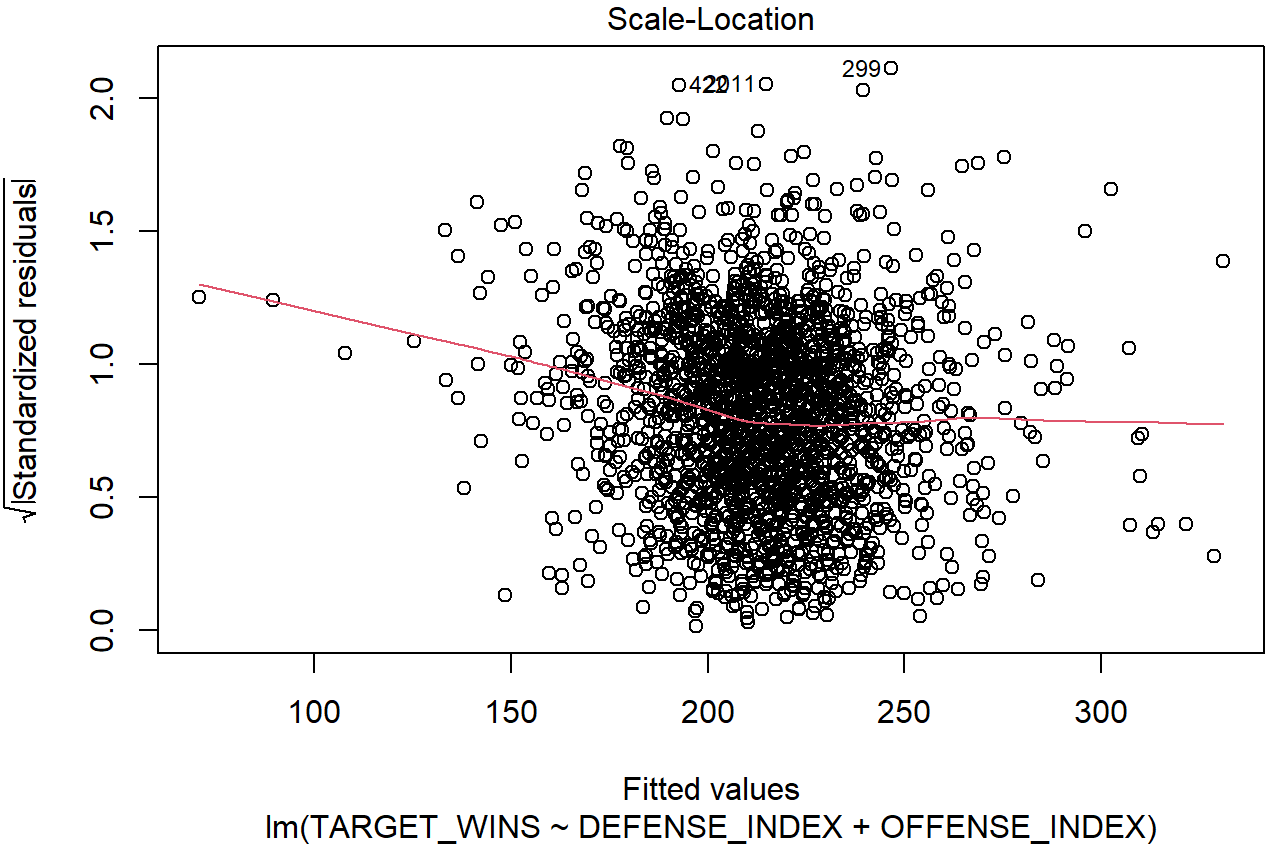
```
## Applying Box Cox to column
##REFERENCE: https://www.statology.org/box-cox-transformation-in-r/

model_2_dfa_transform$TARGET_WINS <- (((model_2_dfa_transform$TARGET_WINS+1)^optimal_lambda -
1) / optimal_lambda)

## Adjusted model w/ BoxCox
model2_bcx<- lm(TARGET_WINS ~ DEFENSE_INDEX + OFFENSE_INDEX, data=model_2_dfa_transform)
print(summary(model2_bcx))
```

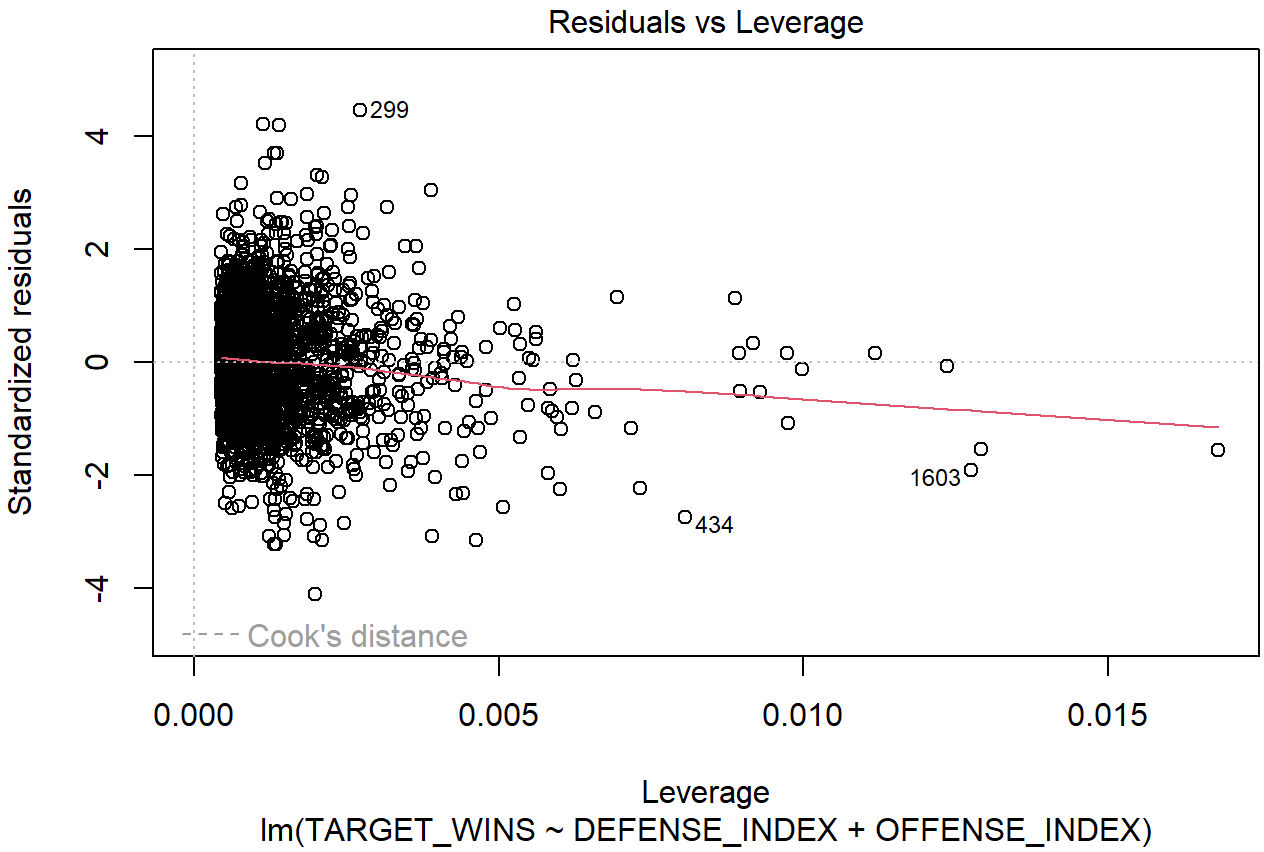
```
##
## Call:
## lm(formula = TARGET_WINS ~ DEFENSE_INDEX + OFFENSE_INDEX, data = model_2_dfa_transform)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -188.094  -30.629    0.259   29.205  203.063
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -30.670467   9.926942  -3.090  0.00203 **
## DEFENSE_INDEX   1.152300   0.267844   4.302 1.76e-05 ***
## OFFENSE_INDEX   0.113744   0.004627  24.581  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.68 on 2271 degrees of freedom
## Multiple R-squared:  0.2131, Adjusted R-squared:  0.2124
## F-statistic: 307.5 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model2_bcx)
```

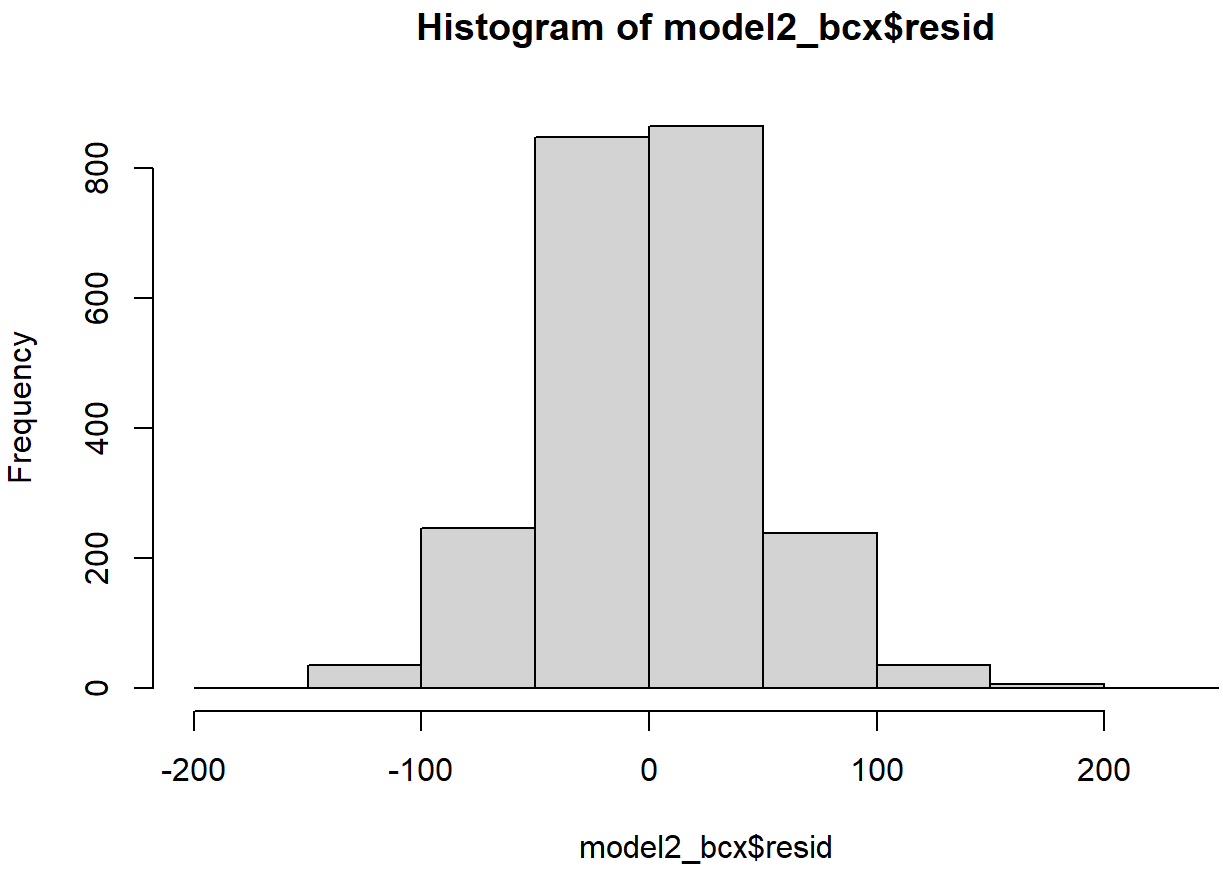



Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ DEFENSE_INDEX + OFFENSE_INDEX)

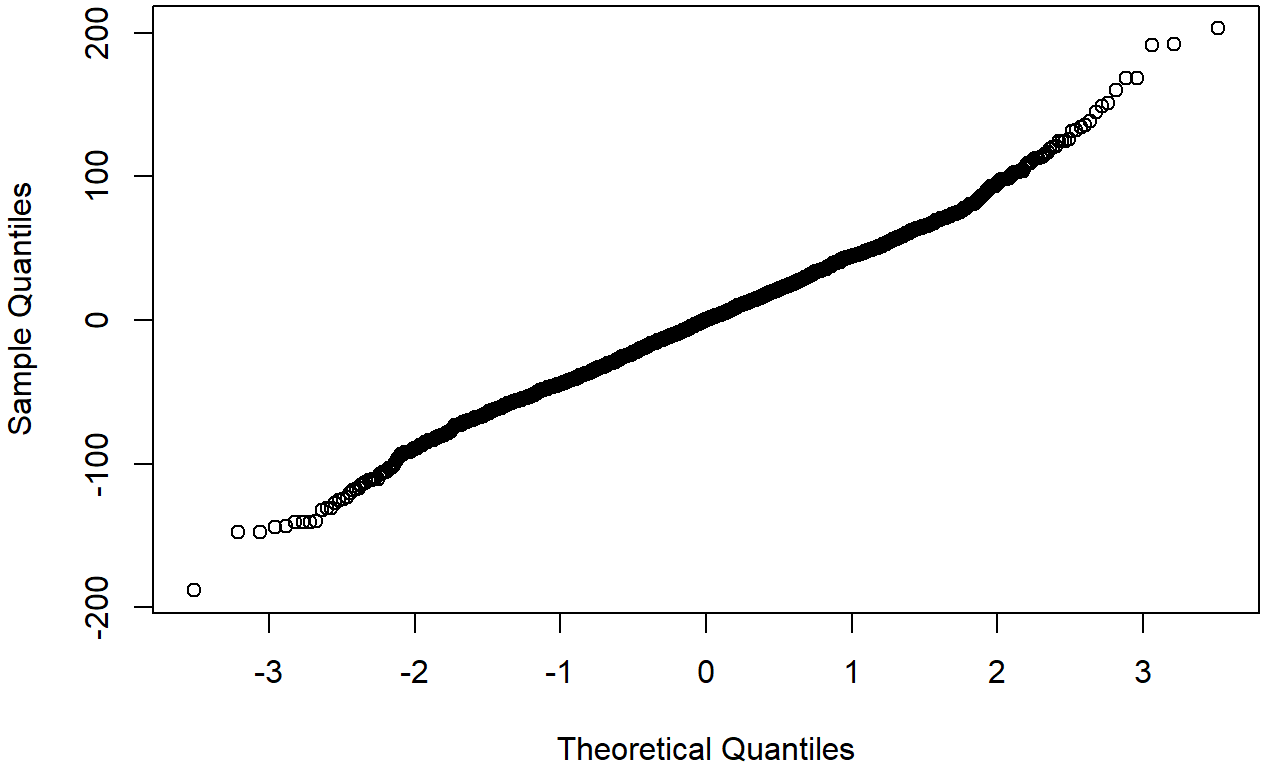

Scale-Location
√|Standardized residuals|
Fitted values
lm(TARGET_WINS ~ DEFENSE_INDEX + OFFENSE_INDEX)




```
hist(model2_bcx$resid)
```




```
qqnorm(model2_bcx$resid)
```




```
skewness(model2_bcx$resid) ## 0.106
```

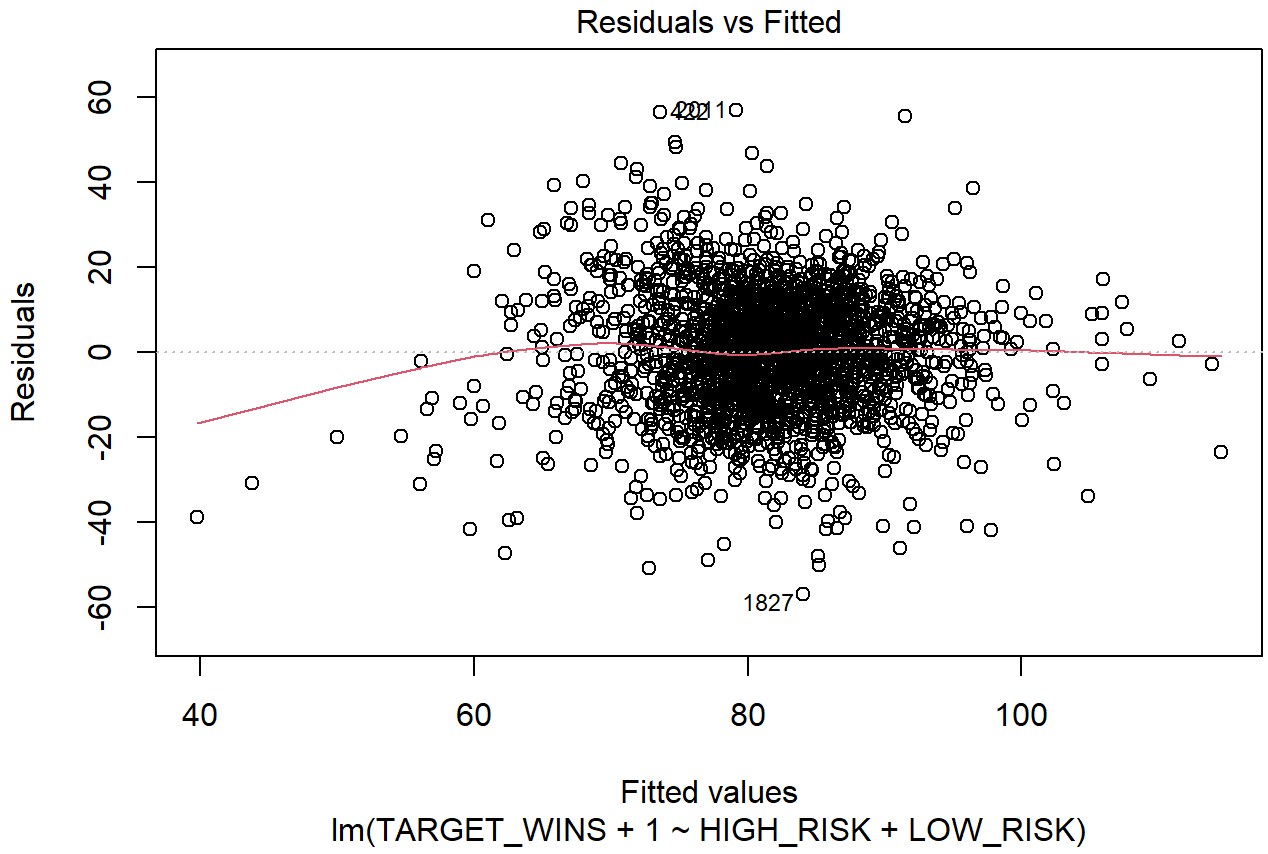
```
## [1] 0.1122622
```

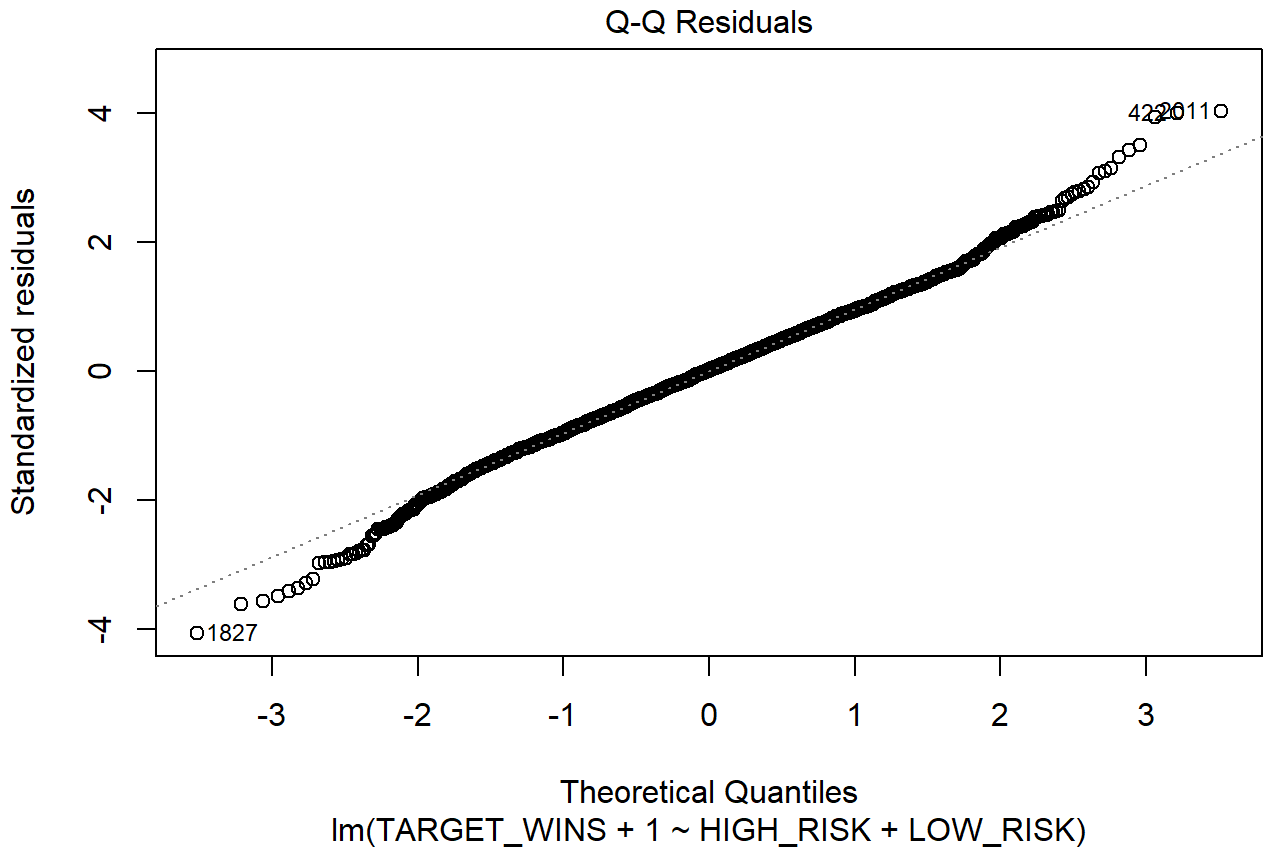
```
## BOX COX Transform improves skewness, but lower the R^2 of the model.
```

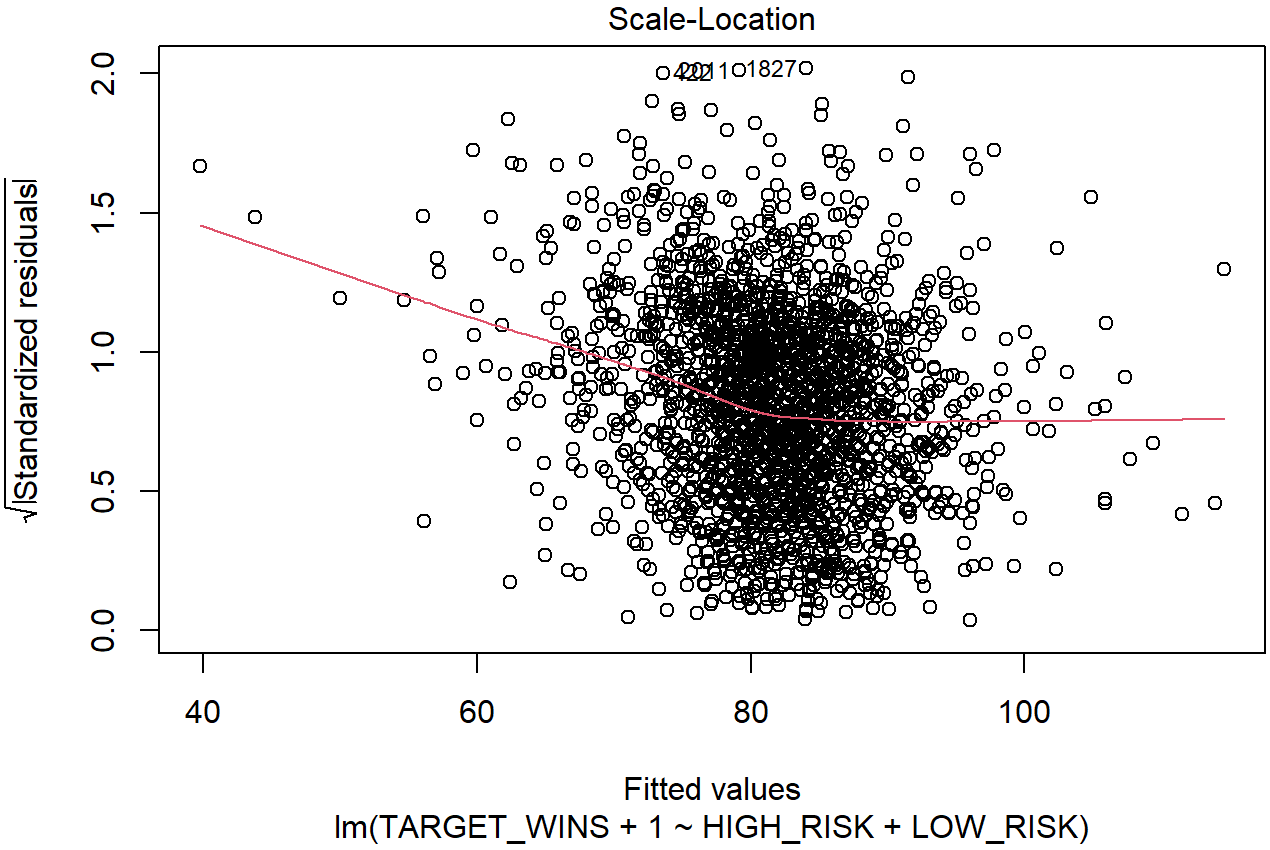
```
### MODEL 3 Risk Index

# --- OTHER INCLUSION:
#     TARGET_WINS
#     TEAM_BATTING_BB_FLAG
#     TEAM_BASERUN_CS_FLAG
#     TEAM_BATTING_HBP_FLAG
#     TEAM_BASERUN_SB_FLAG

#-- HIGH RISK OFFENSE:
#     TEAM_BASERUN_STEAL_ATTEMPTS
#     TEAM_BATTING_2B
#     TEAM_BATTING_3B
#-- LOW RISK OFFENSE:
#     TEAM_BATTING_1B
#     TEAM_BATTING_WALK_TOTAL

## Custom Aggregate
model_3_dfa <- moneyball_train_imputed |>
  mutate (TEAM_BATTING_1B = (TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R))) |>
  dplyr::select( TARGET_WINS,TEAM_BATTING_BB_FLAG,TEAM_BASERUN_CS_FLAG, TEAM_BATTING_HBP_FLA
G,TEAM_BASERUN_SB_FLAG,TEAM_BATTING_HR,TEAM_BATTING_1B,
          TEAM_BATTING_WALK_TOTAL,TEAM_BATTING_2B,TEAM_BATTING_3B,TEAM_BASERUN_STEAL_ATTEMPT
S) |>
  mutate (HIGH_RISK =   (TEAM_BASERUN_STEAL_ATTEMPTS+TEAM_BATTING_2B+TEAM_BATTING_3B),
          LOW_RISK = (TEAM_BATTING_1B + TEAM_BATTING_WALK_TOTAL+TEAM_BATTING_HR)) |>
  dplyr::select(-TEAM_BATTING_1B,-TEAM_BATTING_WALK_TOTAL,-TEAM_BATTING_2B,-TEAM_BATTING_3B,-
TEAM_BASERUN_STEAL_ATTEMPTS,-TEAM_BATTING_HR)

model3_log <- lm(TARGET_WINS+1 ~ HIGH_RISK + LOW_RISK, data=model_3_dfa)
print(summary(model3_log))
```

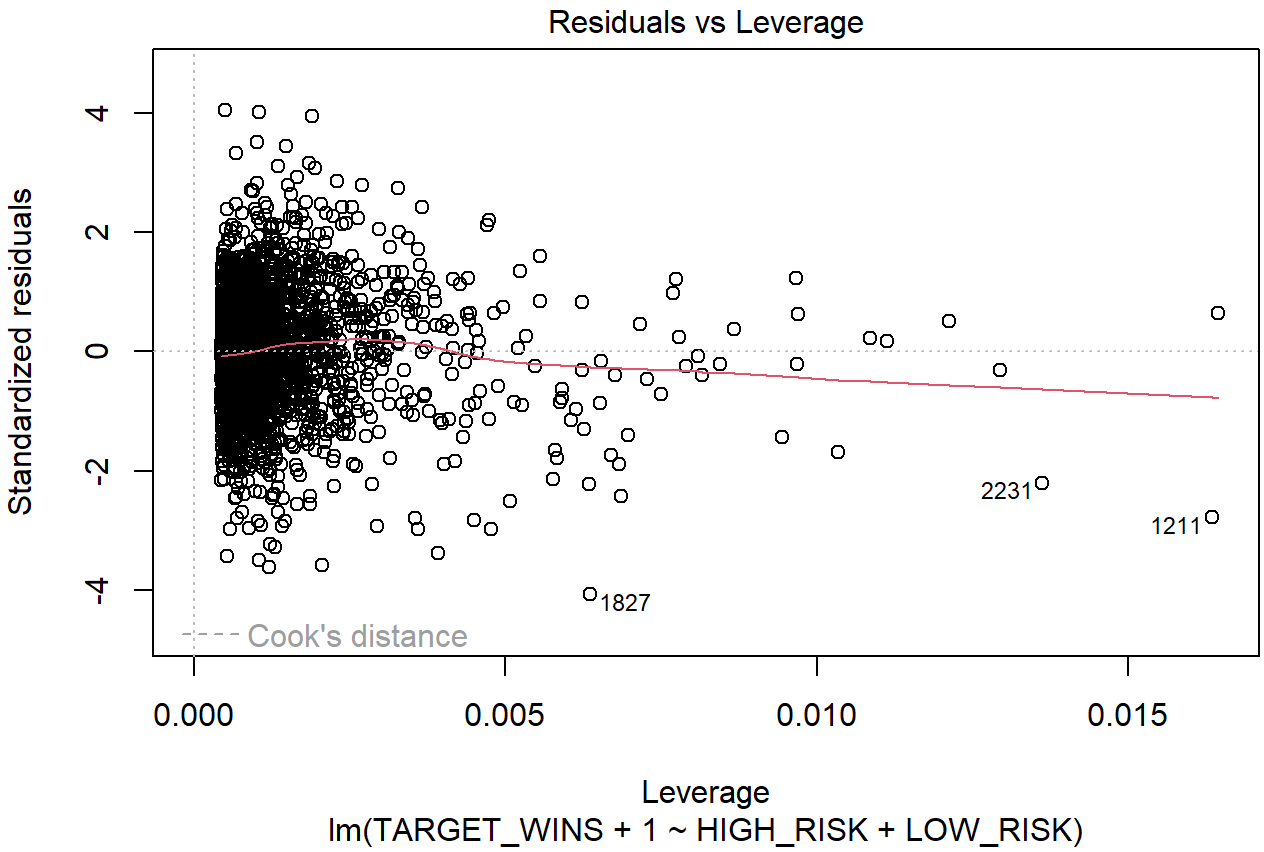
```
##
## Call:
## lm(formula = TARGET_WINS + 1 ~ HIGH_RISK + LOW_RISK, data = model_3_dfa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.078  -9.081   0.197   9.110  56.827
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.664325   3.207541   2.389    0.017 *
## HIGH_RISK   0.018733   0.002900   6.459 1.29e-10 ***
## LOW_RISK    0.039120   0.002029  19.277  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.06 on 2271 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.2002
## F-statistic: 285.6 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model3_log)
```

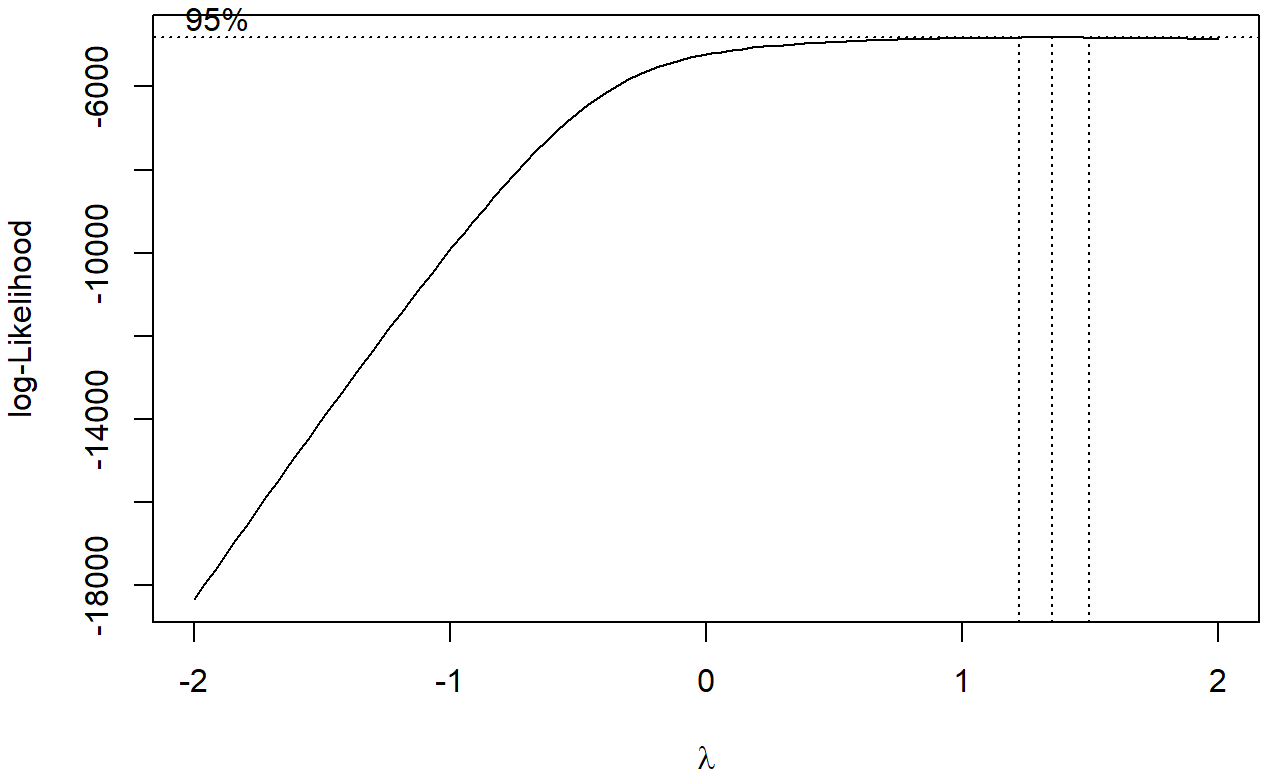



Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
1827
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TARGET_WINS + 1 ~ HIGH_RISK + LOW_RISK)


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
1827
40
60
80
100
Fitted values
lm(TARGET_WINS + 1 ~ HIGH_RISK + LOW_RISK)




```
## Comfirmign the Optimal Lambda
bcx_model3<- boxcox(model3_log)
```



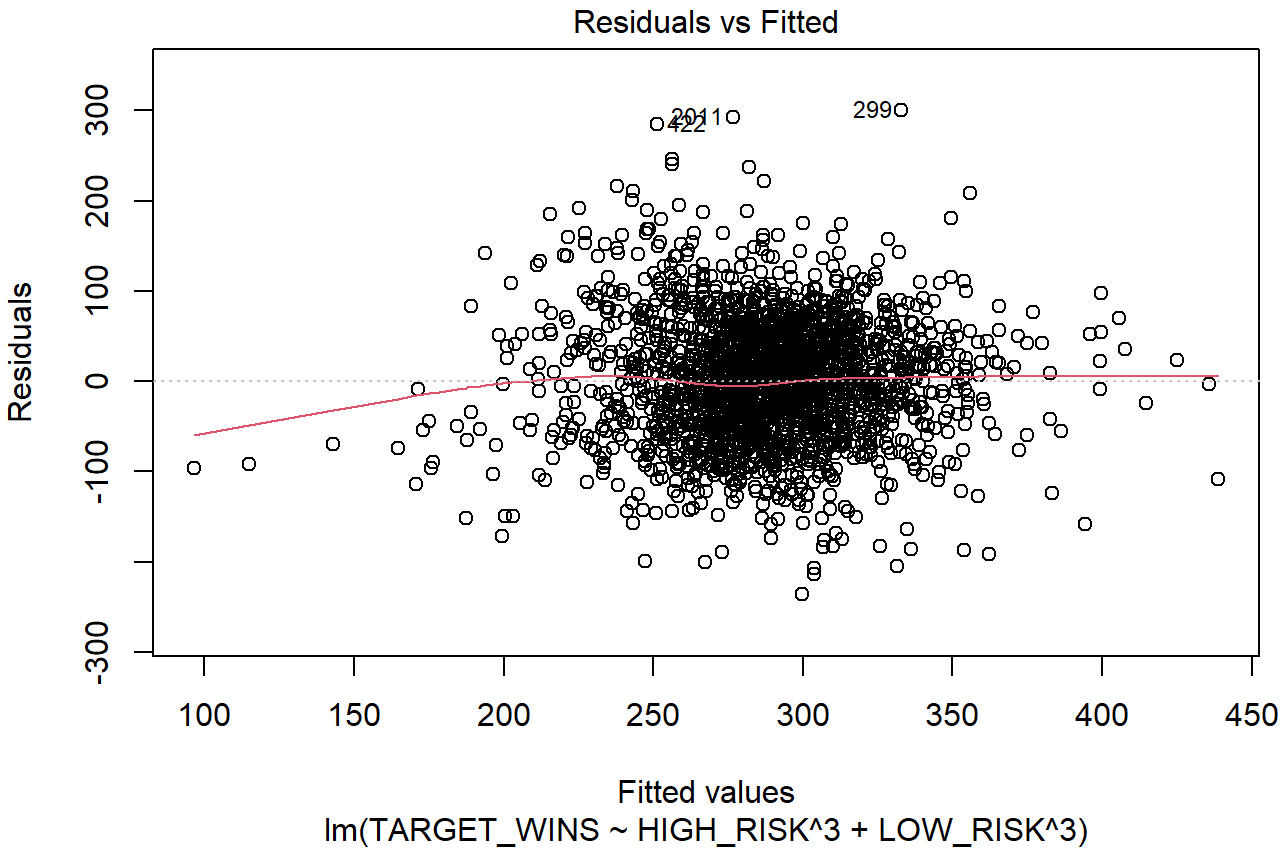

```
optimal_lambda <- bcx_model3$x[which.max(bcx_model3$y)]
print(optimal_lambda)
```

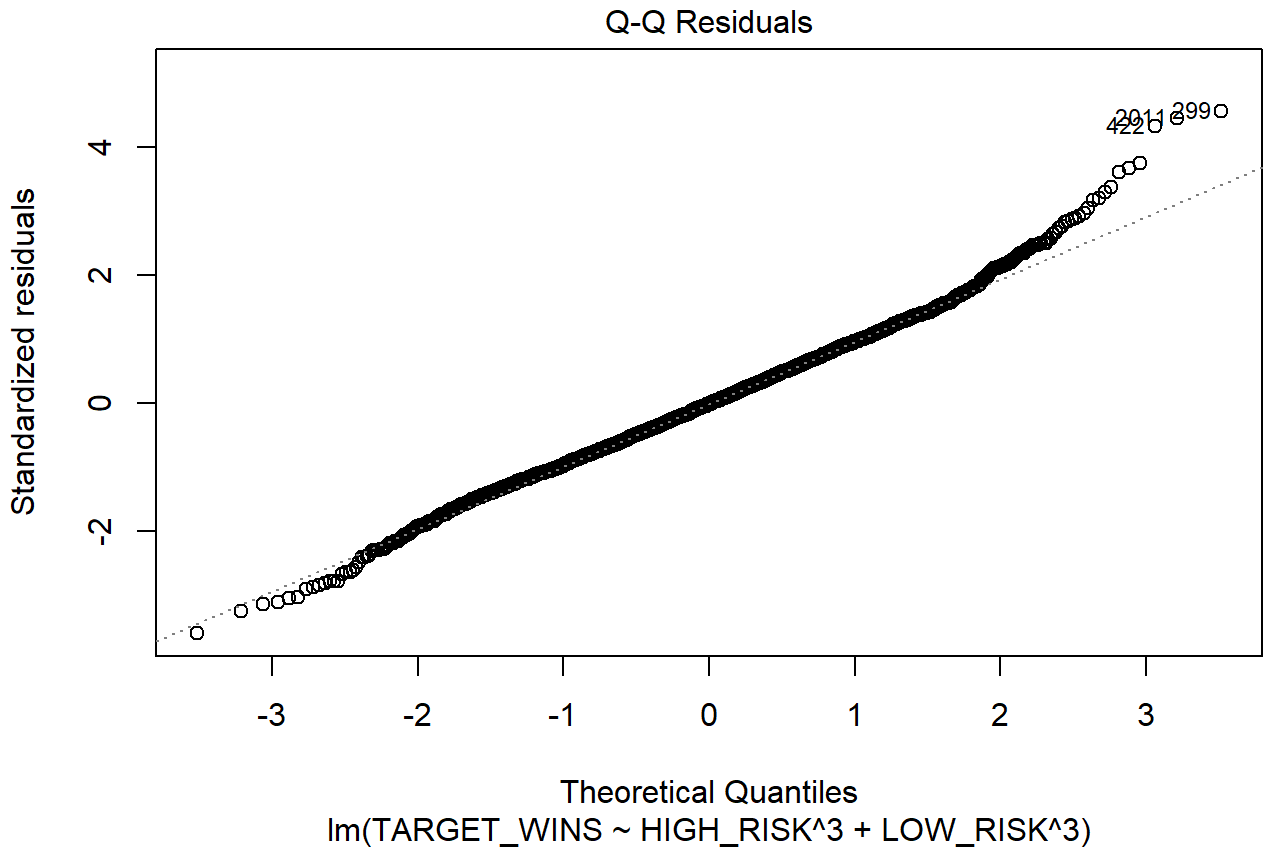
```
## [1] 1.353535
```

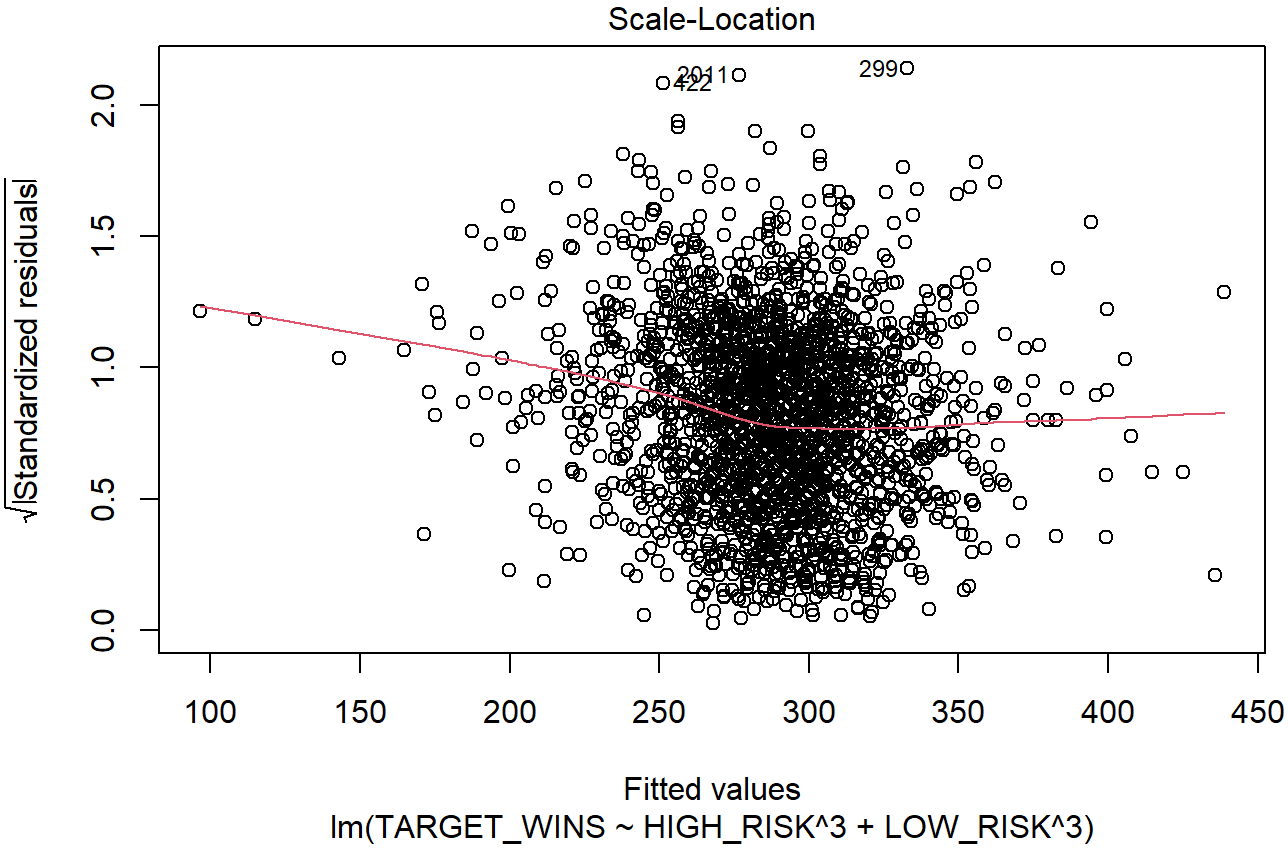
```
model_3a_transform<-model_3_dfa
model_3a_transform$TARGET_WINS <- (((model_3a_transform$TARGET_WINS+1)^optimal_lambda - 1) /
optimal_lambda)


model3_trans <- lm(TARGET_WINS ~ HIGH_RISK^3 + LOW_RISK^3, data=model_3a_transform)
print(summary(model3_trans))
```

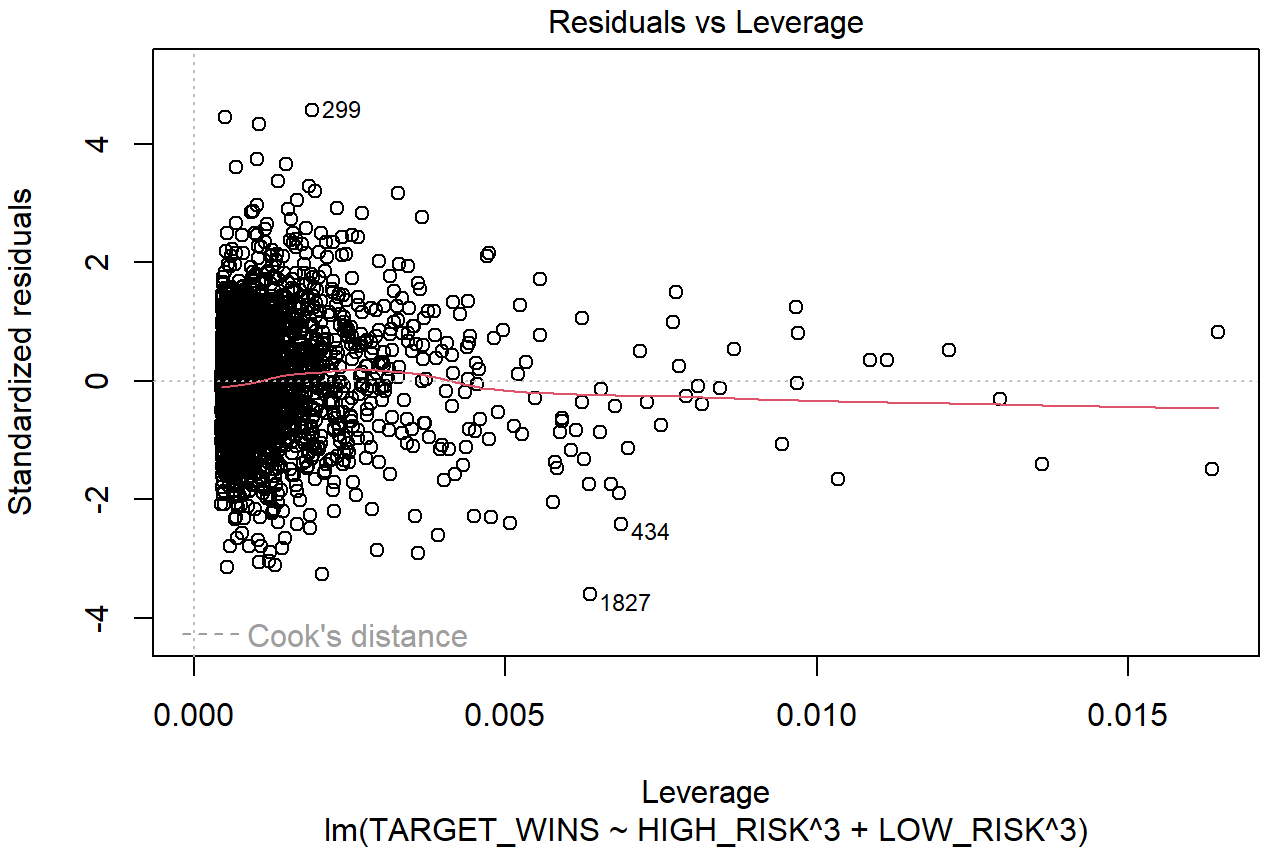
```
##
## Call:
## lm(formula = TARGET_WINS ~ HIGH_RISK^3 + LOW_RISK^3, data = model_3a_transform)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -236.38  -44.09   -1.20   42.45  300.35
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -50.056186  15.004017  -3.336 0.000863 ***
## HIGH_RISK     0.087089   0.013567   6.419 1.66e-10 ***
## LOW_RISK      0.178321   0.009493  18.785  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.79 on 2271 degrees of freedom
## Multiple R-squared:  0.1938, Adjusted R-squared:  0.1931
## F-statistic:   273 on 2 and 2271 DF,  p-value: < 2.2e-16
```

```
plot(model3_trans)
```




Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ HIGH_RISK^3 + LOW_RISK^3)
2011
299
422


Scale-Location
√|Standardized residuals|
Fitted values
lm(TARGET_WINS ~ HIGH_RISK^3 + LOW_RISK^3)
2011
299
422




```
## DROPPING MODEL 3 B/C Lower Fscore and Lower R^2.
## CHOOSING MODEL 2
```

```
### Preparing Test Data using the same methodology of imputation etc as the training

## Adding imputation /missing data Flags to keep track of the changes:
## Present (P) and Missing (M)
moneyball_eval_data$TEAM_BATTING_SO_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_BATTING_S
O), 'M', 'P')
moneyball_eval_data$TEAM_PITCHING_SO_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_PITCHING_S
O), 'M', 'P')
moneyball_eval_data$TEAM_BASERUN_SB_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_BASERUN_S
B), 'M', 'P')
moneyball_eval_data$TEAM_BASERUN_CS_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_BASERUN_C
S), 'M', 'P')
moneyball_eval_data$TEAM_BATTING_BB_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_BATTING_B
B), 'M', 'P')
moneyball_eval_data$TEAM_BATTING_HBP_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_BATTING_HB
P), 'M', 'P')
moneyball_eval_data$TEAM_FIELDING_DP_FLAG <- ifelse(is.na(moneyball_eval_data$TEAM_FIELDING_D
P), 'M', 'P')

moneyball_eval_data_processed <- moneyball_eval_data |>
  mutate(TEAM_BASERUN_STEAL_ATTEMPTS = ifelse(is.na(TEAM_BASERUN_SB), 0, TEAM_BASERUN_SB) + i
felse(is.na(TEAM_BASERUN_CS), 0, TEAM_BASERUN_CS),
         TEAM_BATTING_WALK_TOTAL = ifelse(is.na(TEAM_BATTING_BB), 0, TEAM_BATTING_BB) + ifels
e(is.na(TEAM_BATTING_HBP), 0, TEAM_BATTING_HBP)) |>
  dplyr::select(-TEAM_BASERUN_CS, -TEAM_BATTING_HBP) #-TEAM_BASERUN_SB,-TEAM_BATTING_BB


## Confirming no major outliers like in the original training set
sort(unique(moneyball_eval_data_processed$TEAM_PITCHING_SO))
```

```
##   [1]    0  315  323  324  352  361  367  368  378  401  408  413  420  421  426
##  [16]  430  436  437  446  462  466  468  471  478  479  489  504  509  511  520
##  [31]  524  526  527  529  535  540  545  554  558  559  569  573  576  577  579
##  [46]  583  588  595  597  599  600  604  613  619  621  622  623  625  627  630
##  [61]  631  632  634  635  642  644  645  648  653  656  660  662  663  665  666
##  [76]  667  672  673  675  681  684  693  696  698  703  705  709  712  714  715
##  [91]  717  719  722  725  727  728  729  731  734  735  736  745  747  758  761
## [106]  777  779  780  781  782  785  790  792  794  802  805  811  815  816  817
## [121]  824  827  828  829  831  832  841  842  849  850  854  855  857  860  870
## [136]  871  873  878  889  893  902  903  906  909  912  913  914  917  921  926
## [151]  929  930  932  938  944  947  953  958  963  964  966  969  970  971  973
## [166]  976  977  980  984  990  999 1002 1006 1008 1011 1021 1025 1027 1030 1031
## [181] 1032 1041 1042 1054 1056 1060 1061 1071 1080 1084 1090 1092 1094 1105 1119
## [196] 1123 1142 1145 1170 1177 1231 1268 1295 1462 9963
```

```
## Looking at the columns that need imputation (4/5 columns need imputation that needed imput
ation in trianing)
summary(moneyball_eval_data_processed)
```

```
##      INDEX      TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   :   9   Min.   : 819   Min.   : 44.0   Min.   : 14.00
##  1st Qu.: 708   1st Qu.:1387   1st Qu.:210.0   1st Qu.: 35.00
##  Median :1249   Median :1455   Median :239.0   Median : 52.00
##  Mean   :1264   Mean   :1469   Mean   :241.3   Mean   : 55.91
##  3rd Qu.:1832   3rd Qu.:1548   3rd Qu.:278.5   3rd Qu.: 72.00
##  Max.   :2525   Max.   :2170   Max.   :376.0   Max.   :155.00
##
##  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
##  Min.   :  0.00   Min.   : 15.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 44.50   1st Qu.:436.5   1st Qu.: 545.0   1st Qu.: 59.0
##  Median :101.00   Median :509.0   Median : 686.0   Median : 92.0
##  Mean   : 95.63   Mean   :499.0   Mean   : 709.3   Mean   :123.7
##  3rd Qu.:135.50   3rd Qu.:565.5   3rd Qu.: 912.0   3rd Qu.:151.8
##  Max.   :242.00   Max.   :792.0   Max.   :1268.0   Max.   :580.0
##                                   NA's   :18       NA's   :13
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
##  Min.   : 1155   Min.   :  0.0    Min.   : 136.0   Min.   :   0.0
##  1st Qu.: 1426   1st Qu.: 52.0    1st Qu.: 471.0   1st Qu.: 613.0
##  Median : 1515   Median :104.0    Median : 526.0   Median : 745.0
##  Mean   : 1813   Mean   :102.1    Mean   : 552.4   Mean   : 799.7
##  3rd Qu.: 1681   3rd Qu.:142.5    3rd Qu.: 606.5   3rd Qu.: 938.0
##  Max.   :22768   Max.   :336.0    Max.   :2008.0   Max.   :9963.0
##                                                    NA's   :18
##  TEAM_FIELDING_E  TEAM_FIELDING_DP TEAM_BATTING_SO_FLAG TEAM_PITCHING_SO_FLAG
##  Min.   :  73.0   Min.   : 69.0    Length:259          Length:259
##  1st Qu.: 131.0   1st Qu.:131.0    Class :character    Class :character
##  Median : 163.0   Median :148.0    Mode  :character    Mode  :character
##  Mean   : 249.7   Mean   :146.1
##  3rd Qu.: 252.0   3rd Qu.:164.0
##  Max.   :1568.0   Max.   :204.0
##                   NA's   :31
##  TEAM_BASERUN_SB_FLAG TEAM_BASERUN_CS_FLAG TEAM_BATTING_BB_FLAG
##  Length:259           Length:259           Length:259
##  Class :character     Class :character     Class :character
##  Mode  :character     Mode  :character     Mode  :character
##
##
##
##
##  TEAM_BATTING_HBP_FLAG TEAM_FIELDING_DP_FLAG TEAM_BASERUN_STEAL_ATTEMPTS
##  Length:259            Length:259            Min.   :  0.0
##  Class :character      Class :character      1st Qu.: 88.5
##  Mode  :character      Mode  :character      Median :137.0
##                                              Mean   :152.2
##                                              3rd Qu.:198.0
##                                              Max.   :580.0
##
##  TEAM_BATTING_WALK_TOTAL
##  Min.   : 15.0
##  1st Qu.:443.0
```

```
##  Median :512.0
##  Mean   :503.5
##  3rd Qu.:573.0
##  Max.   :792.0
##
```

```
# --- TEAM_PITCHING_SO,TEAM_BATTING_SO,TEAM_FIELDING_DP,TEAM_BASERUN_SB - Imputation with MIC
E PMM Needed. Probably need to remove the two outlier columns from the TEAM_PITCHING_SO colum
n, and the data at large.

imputed <- mice(moneyball_eval_data_processed,m=5,meth='pmm',seed=5)
```

```
##
##  iter imp variable
##   1   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   1   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   2   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   3   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   4   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   2  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   3  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   4  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
##   5   5  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_PITCHING_SO  TEAM_FIELDING_DP
```

```
## Warning: Number of logged events: 7
```

```
moneyball_eval_imputed<-complete(imputed)

#Checking for nulls where there shouldnt be.
print(nrow(moneyball_eval_imputed |> filter(is.na(TEAM_PITCHING_SO))))
```

```
## [1] 0
```

```
## Working DF
print(head(moneyball_eval_imputed))
```

```
##   INDEX TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 1     9           1209             170              33              83
## 2    10           1221             151              29              88
## 3    14           1395             183              29              93
## 4    47           1539             309              29             159
## 5    60           1445             203              68               5
## 6    63           1431             236              53              10
##   TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_PITCHING_H
## 1             447            1080              62            1209
## 2             516             929              54            1221
## 3             509             816              59            1395
## 4             486             914             148            1539
## 5              95             416              52            3902
## 6             215             377              55            2793
##   TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO TEAM_FIELDING_E
## 1               83              447             1080             140
## 2               88              516              929             135
## 3               93              509              816             156
## 4              159              486              914             124
## 5               14              257             1123             616
## 6               20              420              736             572
##   TEAM_FIELDING_DP TEAM_BATTING_SO_FLAG TEAM_PITCHING_SO_FLAG
## 1              156                    P                     P
## 2              164                    P                     P
## 3              153                    P                     P
## 4              154                    P                     P
## 5              130                    P                     P
## 6              105                    P                     P
##   TEAM_BASERUN_SB_FLAG TEAM_BASERUN_CS_FLAG TEAM_BATTING_BB_FLAG
## 1                    P                    P                    P
## 2                    P                    P                    P
## 3                    P                    P                    P
## 4                    P                    P                    P
## 5                    M                    M                    P
## 6                    M                    M                    P
##   TEAM_BATTING_HBP_FLAG TEAM_FIELDING_DP_FLAG TEAM_BASERUN_STEAL_ATTEMPTS
## 1                     M                     P                         112
## 2                     M                     P                          93
## 3                     M                     P                         106
## 4                     P                     P                         205
## 5                     M                     P                           0
## 6                     M                     P                           0
##   TEAM_BATTING_WALK_TOTAL
## 1                     447
## 2                     516
## 3                     509
## 4                     528
## 5                      95
## 6                     215
```

```
# Predicting TARGET_WINS using Model 2 (Box Cox) improved model with the eval data.

moneyball_eval_formatted <- moneyball_eval_imputed |>
  dplyr::select(TEAM_BATTING_SO_FLAG,TEAM_PITCHING_SO_FLAG,TEAM_BATTING_BB_FLAG,TEAM_BATTING_
HBP_FLAG,TEAM_FIELDING_DP_FLAG,TEAM_BASERUN_SB_FLAG,
         TEAM_PITCHING_SO,TEAM_FIELDING_DP,TEAM_FIELDING_E,TEAM_BATTING_H,TEAM_BATTING_WALK_T
OTAL,TEAM_BATTING_BB,TEAM_BASERUN_SB,TEAM_BATTING_SO) |>
  mutate( DEFENSE_INDEX = ((TEAM_PITCHING_SO + TEAM_FIELDING_DP) / TEAM_FIELDING_E),## Shifti
ng this to keep things positive, but maintain the error influence
         OFFENSE_INDEX=(TEAM_BATTING_H+TEAM_BATTING_BB+TEAM_BASERUN_SB)) |>
  dplyr::select(-TEAM_PITCHING_SO,-TEAM_FIELDING_DP,-TEAM_FIELDING_E,-TEAM_BATTING_H,-TEAM_BA
TTING_WALK_TOTAL,-TEAM_BATTING_BB,-TEAM_BASERUN_SB)


predictions <- predict(model2_2, newdata=moneyball_eval_formatted)  # Predict on TestData

## PREDICTED WINS
predicted_win_data <- cbind(moneyball_eval_imputed, PRED_TARGET_WINS = predictions)
head(predicted_win_data)
```

```
##   INDEX TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 1     9           1209             170              33              83
## 2    10           1221             151              29              88
## 3    14           1395             183              29              93
## 4    47           1539             309              29             159
## 5    60           1445             203              68               5
## 6    63           1431             236              53              10
##   TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_PITCHING_H
## 1             447            1080              62            1209
## 2             516             929              54            1221
## 3             509             816              59            1395
## 4             486             914             148            1539
## 5              95             416              52            3902
## 6             215             377              55            2793
##   TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO TEAM_FIELDING_E
## 1               83              447             1080             140
## 2               88              516              929             135
## 3               93              509              816             156
## 4              159              486              914             124
## 5               14              257             1123             616
## 6               20              420              736             572
##   TEAM_FIELDING_DP TEAM_BATTING_SO_FLAG TEAM_PITCHING_SO_FLAG
## 1              156                    P                     P
## 2              164                    P                     P
## 3              153                    P                     P
## 4              154                    P                     P
## 5              130                    P                     P
## 6              105                    P                     P
##   TEAM_BASERUN_SB_FLAG TEAM_BASERUN_CS_FLAG TEAM_BATTING_BB_FLAG
## 1                    P                    P                    P
## 2                    P                    P                    P
## 3                    P                    P                    P
## 4                    P                    P                    P
## 5                    M                    M                    P
## 6                    M                    M                    P
##   TEAM_BATTING_HBP_FLAG TEAM_FIELDING_DP_FLAG TEAM_BASERUN_STEAL_ATTEMPTS
## 1                     M                     P                         112
## 2                     M                     P                          93
## 3                     M                     P                         106
## 4                     P                     P                         205
## 5                     M                     P                           0
## 6                     M                     P                           0
##   TEAM_BATTING_WALK_TOTAL PRED_TARGET_WINS
## 1                     447         68.45039
## 2                     516         71.41741
## 3                     509         77.84143
## 4                     528         85.91625
## 5                      95         60.51704
## 6                     215         64.95147
```

```
#write.csv(predicted_win_data, "TARGET_WINS_Prediction_moneyball_data.csv")

### NO Actual TARGET_WINS in the test / eval data, so cant demonstrate accuracy.
```

```
## NOT USING BECAUSE INSTRUCTIONS SAID TO MANUALLY SELECT VARIABLES
#Pivoting to the technique used in Week 5 Content videos, as none of the models I manually th
ought of were great.


#install.packages("leaps")

#library(leaps)

## Limiting to certain variables, and keeping only one version of each of the directly linear
ly related variables.
#moneyball_train_imputed_lim <-moneyball_train_imputed |>
#  mutate(TEAM_BATTING_1B = TEAM_BATTING_H - (TEAM_BATTING_2B+TEAM_BATTING_3B+TEAM_BATTING_H
R)) |>
#  dplyr::select(TARGET_WINS,TEAM_BATTING_1B,TEAM_BATTING_2B,TEAM_BATTING_2B,TEAM_BATTING_H
R,TEAM_PITCHING_SO,
#                TEAM_BATTING_WALK_TOTAL,TEAM_BASERUN_STEAL_ATTEMPTS,TEAM_FIELDING_E,TEAM_FIE
LDING_DP)

#moneyball_train_imputed_lim

## Limited df for the ModelSelection Route.
#regfit.full = regsubsets(TARGET_WINS~.,data=moneyball_train_imputed_lim, nvmax=9)

#reg.summary<-summary(regfit.full)

### Plottign the CP in order to pick the ideal number of variables
#plot(reg.summary$cp, xlab="Number of Variables", ylab="Cp")
## Only 9 var ,so lower may be higher vars. Confirming:
#which.min(reg.summary$cp)
## Model number 6 is ideal.Looking at the other CP chart from video

# Looking and 7 is actually the lowest for
#plot(regfit.full, scale="Cp")
#coef(regfit.full, 6)
```