TF Moneyball

Tyler Frankenberg

03/06/2022

Import packages

```
library(tidyverse)
library(ggcorrplot)
library(rms)
```

Import data & Glimpse its Structure

We'll start by viewing the Moneyball dataset's structure and the summary statistics for each of its columns. We can see there are 16 columns of numeric data.

```
url <- "https://raw.githubusercontent.com/curdferguson/data621/main/datasets/moneyball-training-data.csv"
moneyball_raw <- url %>% read_csv(na='') %>% column_to_rownames(var="INDEX")

moneyball_raw %>% head(5)
```

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

```
moneyball_raw %>% summary()
```

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

Data Cleaning

Our initial summary reveals a few immediate concerns:

Investigate Zero-win Record Season

There are implausible minimum values of `0` for several columns, including TARGET_WINS. According to Wikipedia, the worst professional baseball season on record was achieved by 1875 Brooklyn Atlantics, of the short-lived “National Association”. Even the lowly Atlantics were able to scrape together two wins that year. Filtering reveals these zero minimums to come from a single row, which has several `NA` values as well, so appears to be incomplete or erroneous. We can safely remove this row from the data.

Source: https://en.wikipedia.org/wiki/List_of_worst_Major_League_Baseball_season_records (https://en.wikipedia.org/wiki/List_of_worst_Major_League_Baseball_season_records)

```
moneyball_raw0 <- moneyball_raw[moneyball_raw$TARGET_WINS > 0, ]
```

Handle NA Values

There are a high number of of `NA` values in the columns TEAM_BATTING_SO, TEAM_BASERUN_SB, TEAM_BASERUN_CS, TEAM_BATTING_HBP, TEAM_PITCHING_SO, and TEAM_FIELDING_DP. There is only one clear-cut case here, which is for the column TEAM_BATTING_HBP, which includes `772` NA values. This will limit any linear model it is a part of, so we'll start by dropping this column from the dataset.

Next, we'll examine how many rows contain more than one `NA` value. Since they are relatively few, but still number in the hundreds of rows, we will drop those rows from the dataset, rather than try to impute missing data.

```
moneyball_raw1 <- moneyball_raw0 %>% select(!c("TEAM_BATTING_HBP"))

NArows <- data.frame(`NAvals>=4` = nrow(moneyball_raw1[rowSums(is.na(moneyball_raw1)) >= 4, ]),
           `NAvals==3` = nrow(moneyball_raw1[rowSums(is.na(moneyball_raw1)) == 3, ]),
           `NAvals==2` = nrow(moneyball_raw1[rowSums(is.na(moneyball_raw1)) == 2, ]),
           `NAvals==1` = nrow(moneyball_raw1[rowSums(is.na(moneyball_raw1)) == 1, ]),
           `NAvals==0` = nrow(moneyball_raw1[rowSums(is.na(moneyball_raw1)) == 0, ]))

NArows
```

```
##   NAvals..4 NAvals..3 NAvals..2 NAvals..1 NAvals..0
## 1         0       180       243       366      1486
```

```
#source: https://statisticsglobe.com/r-remove-data-frame-rows-with-some-or-all-na#:~:text=The%20output%20is%20the%20same%20as%20in%20the,you%20can%20replace%20%E2%80%9C%3D%3D%200%E2%80%9D%20by%20%E2%80%9C%3E%3D%202%E2%80%9D.
```

Next, we'll take a quick look at the rows with one `NA` value. These fall either in the column TEAM_BASERUN_CS, or in TEAM_FIELDING_DP. TEAM_BASERUN_CS is an interesting data point, because when added to TEAM_BASERUN_SB, it quantifies how often teams are willing to risk an out in order to advance a runner on the base path. This is often mentioned as the type of "old school" baseball behavior that today's more conservative, analytically-minded organizations discourage. Thus, it may help generate an important proxy for 'aggressiveness' in a team's offensive play. Given the information we therefore can glean from this field, we'll keep the column though it contains hundreds of `NA` values, and drop the `NA` rows.

```
head(moneyball_raw1[rowSums(is.na(moneyball_raw1)) == 1, ], 10)
```

```
##     TARGET_WINS TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 82           77           1647             191              80              56
## 84           86           1393             168              46              17
## 92           56           1334             183              52              33
## 93           47           1520             267              58              40
## 94           55           1551             240              72              37
## 95           74           1414             205              64              34
## 96          100           1384             226              64              37
## 99           76           1355             179              79              23
## 100          69           1325             140              77              17
## 101          66           1400             165              72              28
##     TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
## 82              464             326             214              NA
## 84              471             569             223              NA
## 92              380             572             161              NA
## 93              594             619             181              NA
## 94              481             731             145              NA
## 95              524             687             190              NA
## 96              532             653             147              NA
## 99              452             622             164              NA
## 100             457             572             108              NA
## 101             414             561             169              NA
##     TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO
## 82             1933               66               545              383
## 84             1647               20               557              673
## 92             1412               35               402              606
## 93             1631               43               637              664
## 94             1642               39               509              774
## 95             1517               36               562              737
## 96             1465               39               563              691
## 99             1435               24               479              659
## 100            1731               22               597              747
## 101            1632               33               483              654
##     TEAM_FIELDING_E TEAM_FIELDING_DP
## 82              318              101
## 84              279              106
## 92              323              145
## 93              372              118
## 94              312              137
## 95              293               88
## 96              251              151
## 99              237              129
## 100             240              116
## 101             238              129
```

This leaves just 18 NAs on the column TEAM_FIELDING_DP. This is certainly a low enough number of rows to consider imputation of a mean value; however, further investigation into this subset reveals additional concerns; namely, absurdly high values for TEAM_PITCHING_H (max = `30132`) and TEAM_PITCHING_SO (max = `192781`).

According to Baseball Almanac, the Major League record for most strikeouts in a season by a pitching staff is `1687` by the 2018 Houston Astros. Since our data goes all the way back to the earliest days of the sport in 1871, it likely includes team totals for clubs and leagues outside the modern definition of “Major” such as semi-professional teams, or “barnstorming” teams who dominated poorly-matched local competition as they toured the country.

Whether these values actually represent a herculean effort out of baseball’s mythic age, or are simply erroneous, they are so outside the realm of plausibility for a modern-day team that we can consider them invalid and delete the rows.

Source: https://www.baseball-almanac.com/recbooks/rb_strike2.shtml (https://www.baseball-almanac.com/recbooks/rb_strike2.shtml)

```
moneyball_raw2 <- moneyball_raw1[rowSums(is.na(moneyball_raw1)) == 0, ]
```

Transform Columns into Meaningful Measures

Now that we have cleaned our raw data, we must consider the relevance of each column to a team’s success, and therefore its conceptual appropriateness in a predictive model. In other words, rather than taking each column at face value, we need to consider how it the underlying activity impacts the game of baseball, both on its own and in combination with the other measures. We come up with the following for our test dataset, which we’ll explain in more detail below.

```
moneyball_train <- moneyball_raw2 %>% summarise(
    TARGET_WINS = TARGET_WINS,
    TEAM_BATTING_TB = ((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) +
2*TEAM_BATTING_2B + 3*TEAM_BATTING_3B + 4*TEAM_BATTING_HR),
    TEAM_BATTING_BB = TEAM_BATTING_BB,
    TEAM_BATTING_SO = TEAM_BATTING_SO,
    TEAM_BASERUN_ATT = TEAM_BASERUN_SB + TEAM_BASERUN_CS,
    TEAM_BASERUN_PCT = TEAM_BASERUN_SB / TEAM_BASERUN_ATT,
    TEAM_FIELDING_E = TEAM_FIELDING_E,
    TEAM_FIELDING_DP = TEAM_FIELDING_DP,
    TEAM_PITCHING_BB = TEAM_PITCHING_BB,
    TEAM_PITCHING_H = TEAM_PITCHING_H - TEAM_PITCHING_HR,
    TEAM_PITCHING_HR = TEAM_PITCHING_HR,
    TEAM_PITCHING_SO = TEAM_PITCHING_SO
)

summary(moneyball_train)
```

```
##   TARGET_WINS    TEAM_BATTING_TB TEAM_BATTING_BB TEAM_BATTING_SO
##  Min.   : 41.0   Min.   :1509    Min.   :309.0   Min.   : 326.0
##  1st Qu.: 72.0   1st Qu.:2031    1st Qu.:488.0   1st Qu.: 717.0
##  Median : 81.5   Median :2175    Median :534.0   Median : 870.5
##  Mean   : 81.0   Mean   :2178    Mean   :541.9   Mean   : 841.7
##  3rd Qu.: 90.0   3rd Qu.:2322    3rd Qu.:592.8   3rd Qu.: 983.0
##  Max.   :117.0   Max.   :2832    Max.   :878.0   Max.   :1399.0
##  TEAM_BASERUN_ATT TEAM_BASERUN_PCT TEAM_FIELDING_E TEAM_FIELDING_DP
##  Min.   : 34.0    Min.   :0.3485   Min.   : 65.0   Min.   : 87.0
##  1st Qu.:104.0    1st Qu.:0.5789   1st Qu.:117.0   1st Qu.:140.0
##  Median :141.0    Median :0.6419   Median :136.0   Median :153.0
##  Mean   :148.8    Mean   :0.6328   Mean   :143.1   Mean   :153.7
##  3rd Qu.:184.0    3rd Qu.:0.6950   3rd Qu.:160.0   3rd Qu.:167.0
##  Max.   :465.0    Max.   :0.8403   Max.   :360.0   Max.   :228.0
##  TEAM_PITCHING_BB TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_SO
##  Min.   : 325.0   Min.   :1028    Min.   : 12.0    Min.   : 345.0
##  1st Qu.: 495.2   1st Qu.:1261    1st Qu.:101.0    1st Qu.: 738.0
##  Median : 548.0   Median :1327    Median :134.0    Median : 884.0
##  Mean   : 561.6   Mean   :1371    Mean   :134.1    Mean   : 869.3
##  3rd Qu.: 614.0   3rd Qu.:1423    3rd Qu.:167.0    3rd Qu.:1005.8
##  Max.   :1090.0   Max.   :2237    Max.   :343.0    Max.   :1781.0
```

Batting Measures

Since TEAM_BATTING_H is by its nature inclusive of the data in the 2B, 3B, and HR columns, we need to subtract their values from it so that it is only measuring single-base hits. But we should also consider the fact that 2B, 3B, and HR are worth more in terms of progress toward runs scored - runs are of course the currency with which wins are acquired - and weight them appropriately.

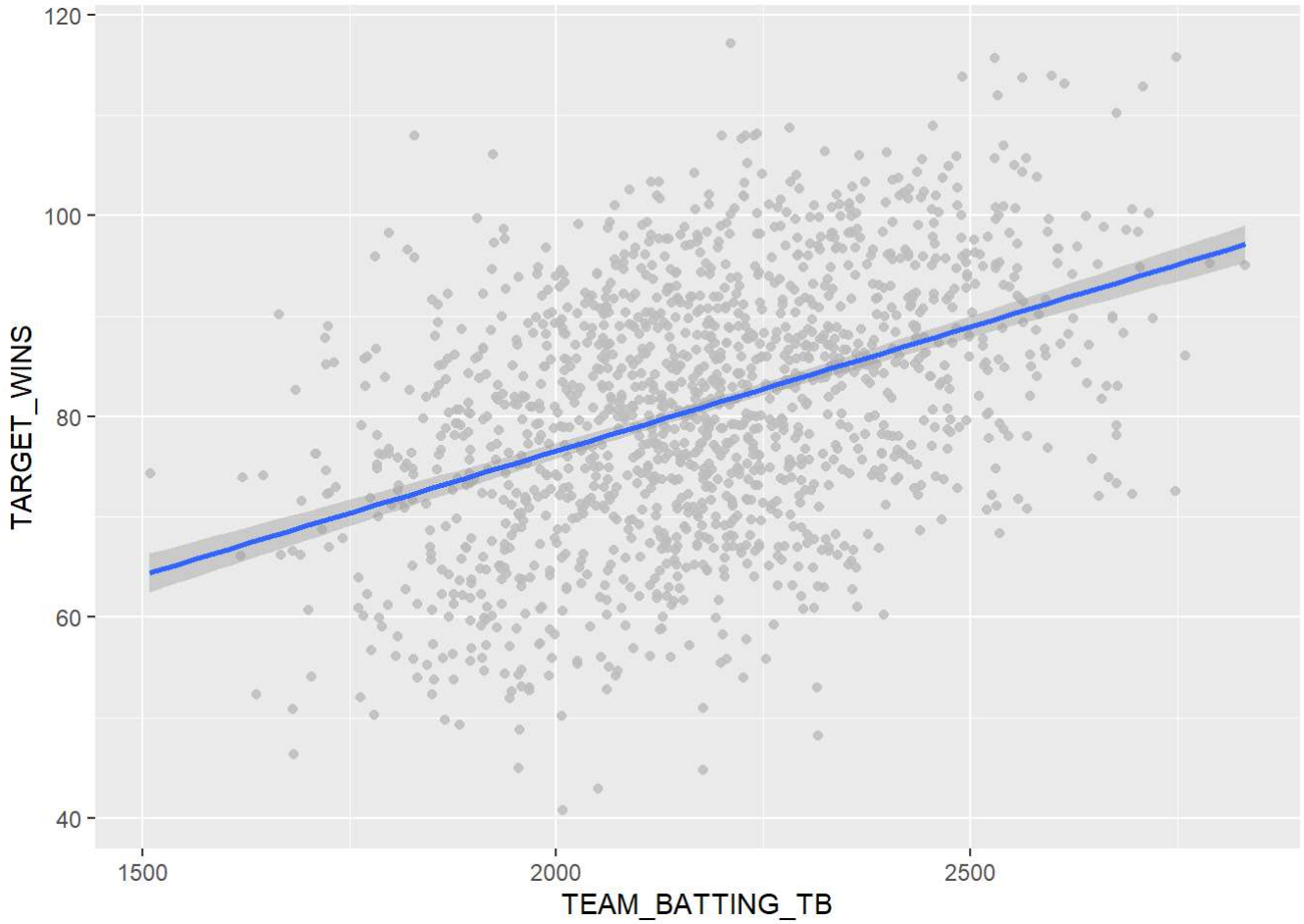
We can conveniently achieve these ends while making the model more concise by combining all of these columns together into the measure TEAM_BATTING_TB (“Total Bases”), a real-life baseball statistic which simply counts the number of bases achieved through hits in fair territory.

TEAM_BATTING_BB and TEAM_BATTING_SO are somewhat more straightforward as measures in their own right, and both are relevant to the outcome of the game (though they may well be negatively correlated, as ‘plate discipline’, or the batter’s ability to refrain from swinging at pitches thrown outside the “strike zone”, should both increase his number of walks and reduce his number of strikeouts).

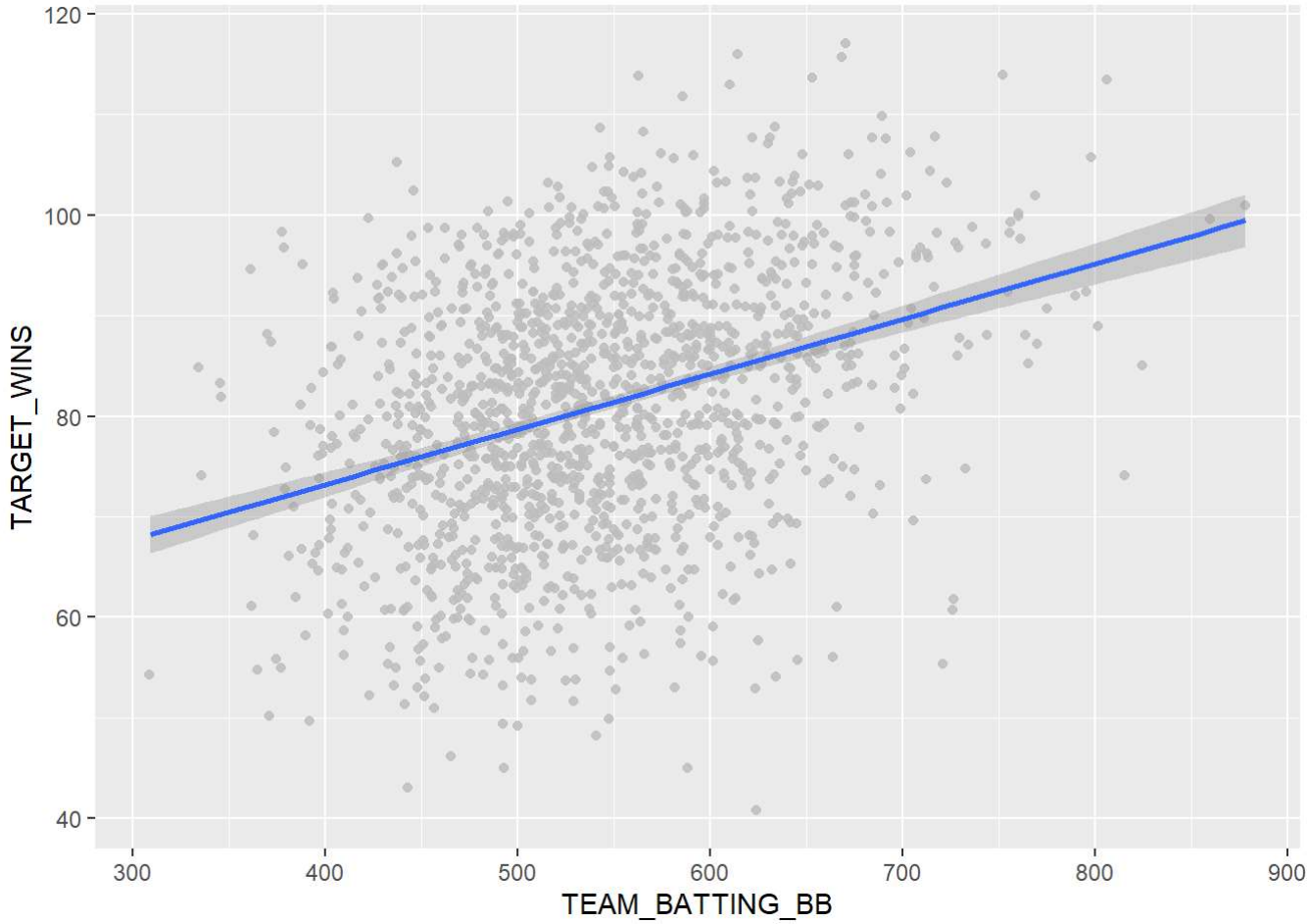
Scatter plots vs. TARGET_WINS show clear positive linear associations for Total Bases and Walks, while Strikeouts seem to have a less impactful relationship.

```
batting_plots <- lapply(colnames(moneyball_train[, 2:4]), function(c) {
  ggplot(moneyball_train, aes(moneyball_train[, c], TARGET_WINS)) + geom_jitter(color="gray", al
pha=0.85) + geom_smooth(method = "lm") + xlab(c)
 })

batting_plots
```

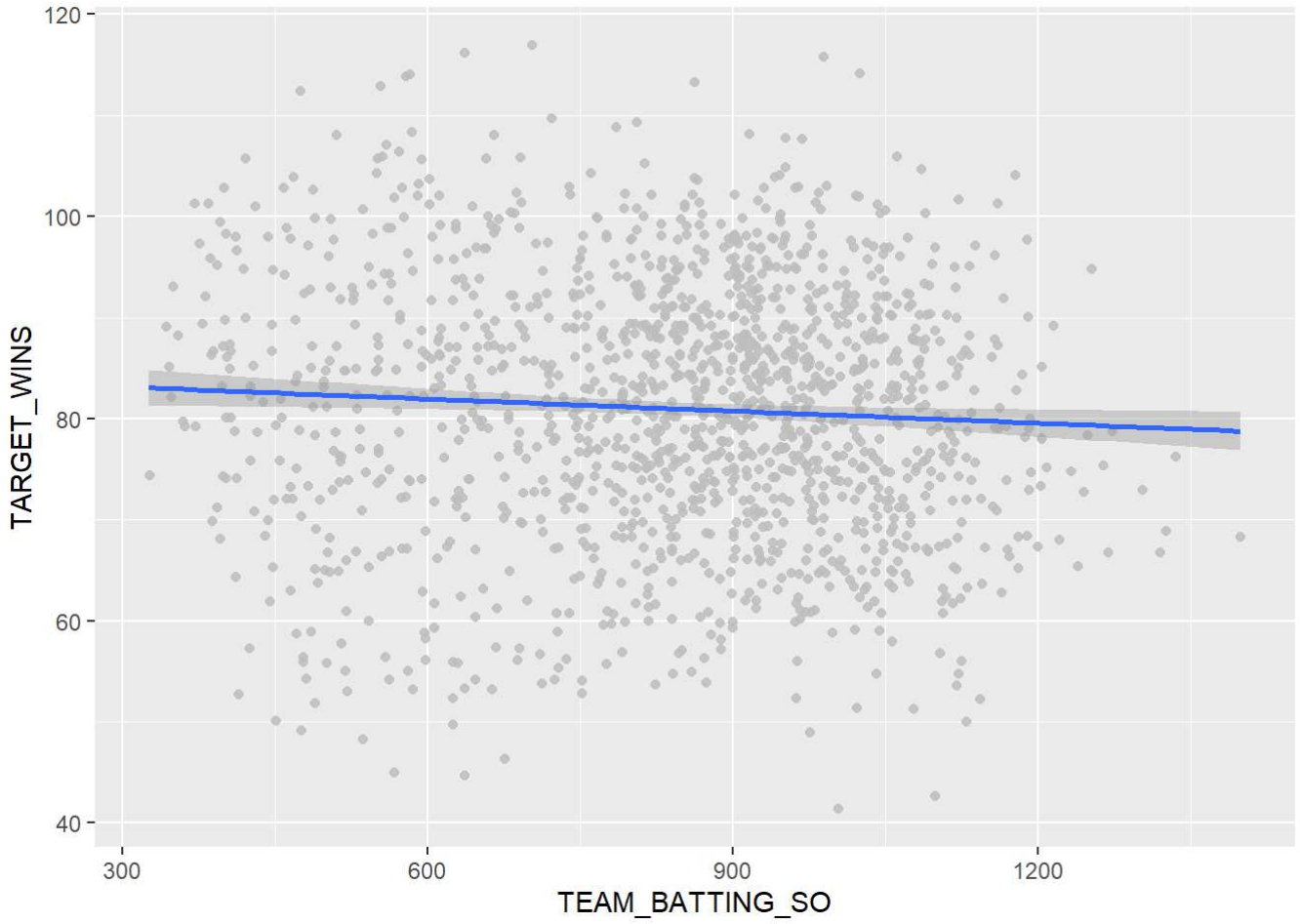
```
## [[1]]
```




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


120
100
80
60
40
TARGET_WINS
300
400
500
600
700
800
900
TEAM_BATTING_BB

```
##
## [[3]]
```



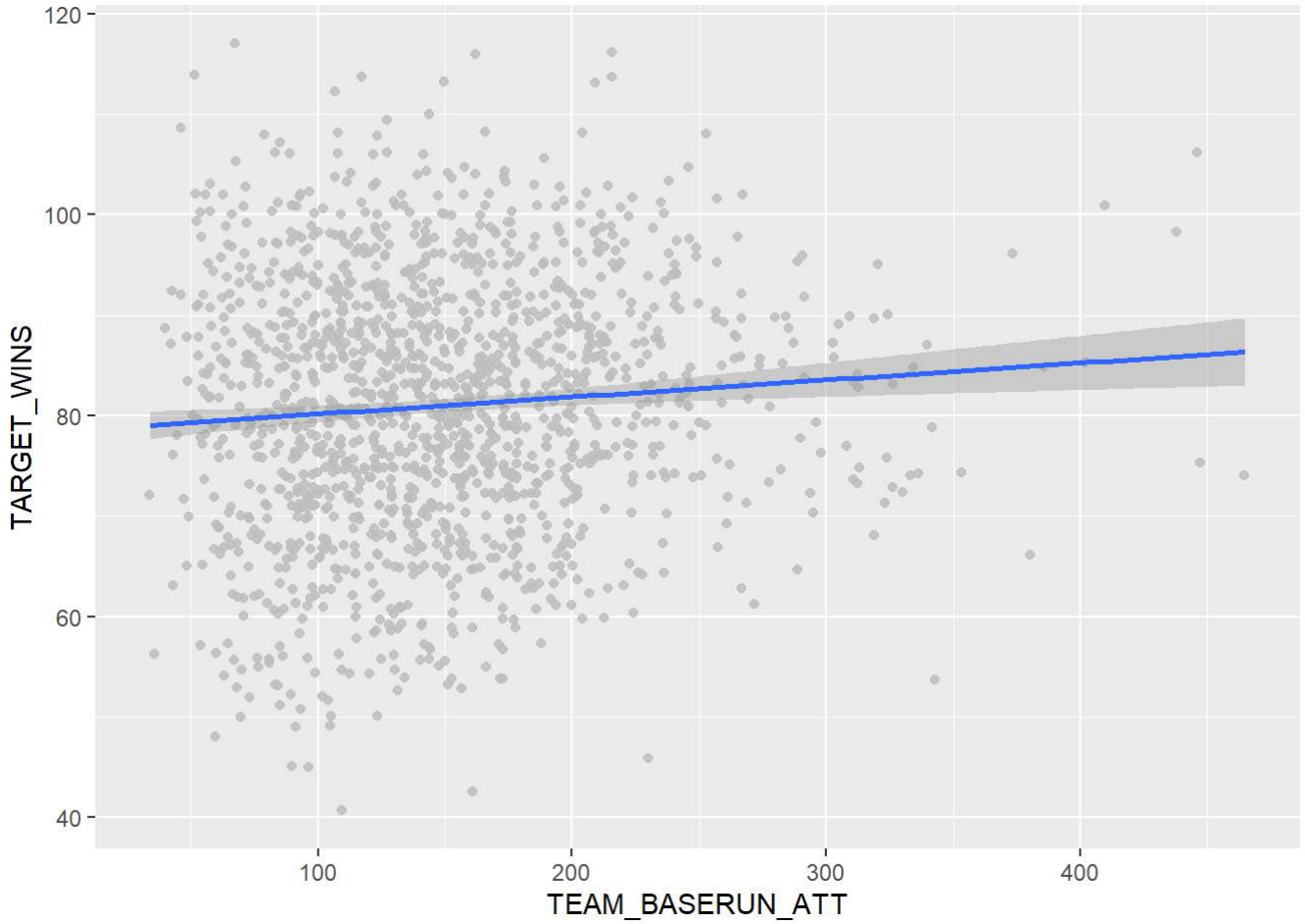
Base Running Measures

As we discussed above, we'll add TEAM_BASERUN_SB and TEAM_BASERUN_CS together to get TEAM_BASERUN_ATT ("Attempts"). This is a number we might expect to be higher for older teams and lower for modern-day teams, as contemporary analytics on which baseball strategy is now often based have come to frown upon the risk involved in base stealing. But over the long term it should be a good proxy for "aggressive" (high TEAM_BASERUN_ATT) versus "conservative" (low TEAM_BASERUN_ATT) offensive play. For good measure, we'll compute the percentage of attempts that resulted in successful steals as TEAM_BASERUN_PCT.

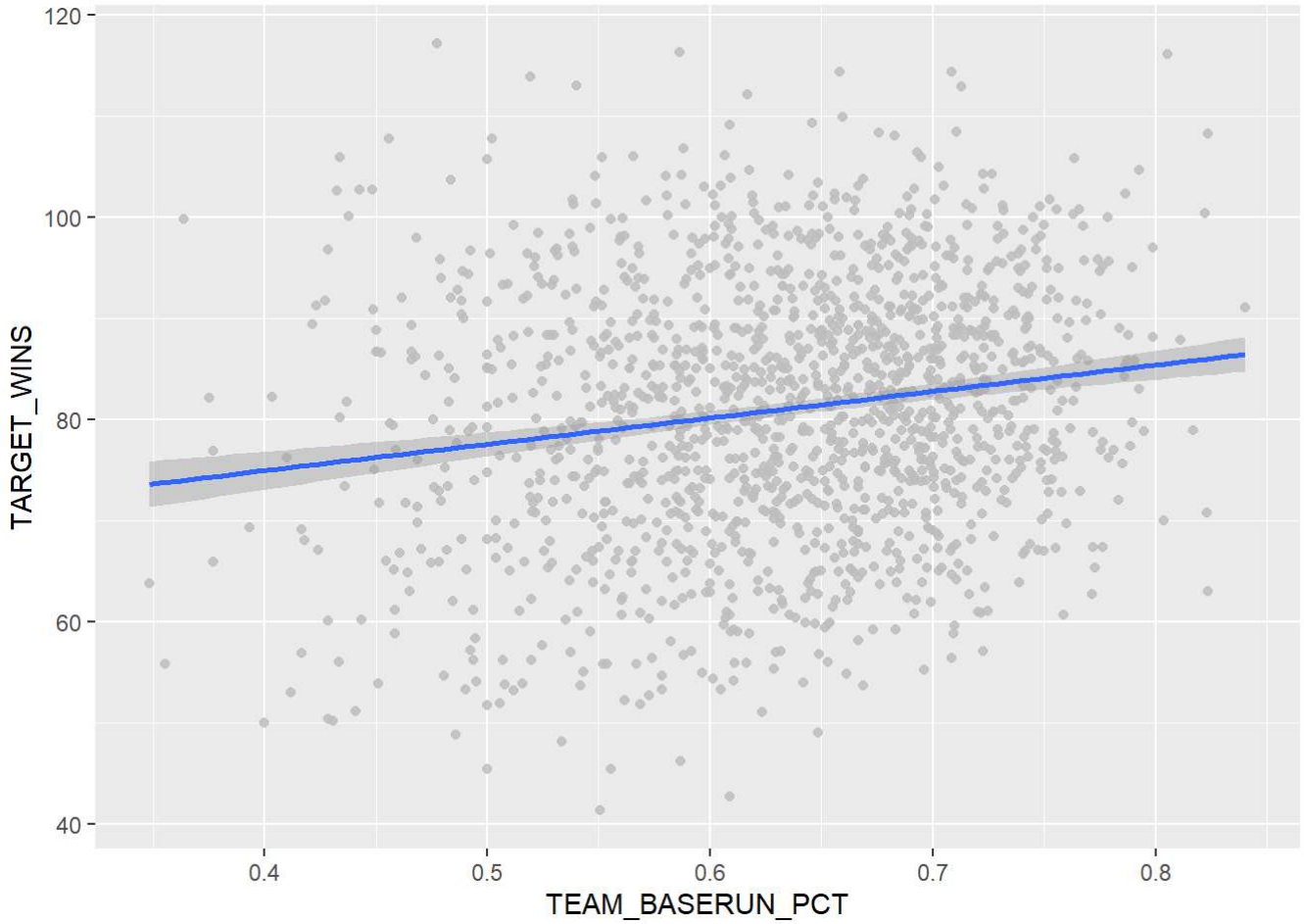
There is an apparent positive linear association between both baserunning measures and TARGET_WINS.

```
baserun_plots <- lapply(colnames(moneyball_train[, 5:6]), function(c) {
  ggplot(moneyball_train, aes(moneyball_train[, c], TARGET_WINS)) + geom_jitter(color="gray", al
pha=0.85) + geom_smooth(method = "lm") + xlab(c)
 })

baserun_plots
```

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




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



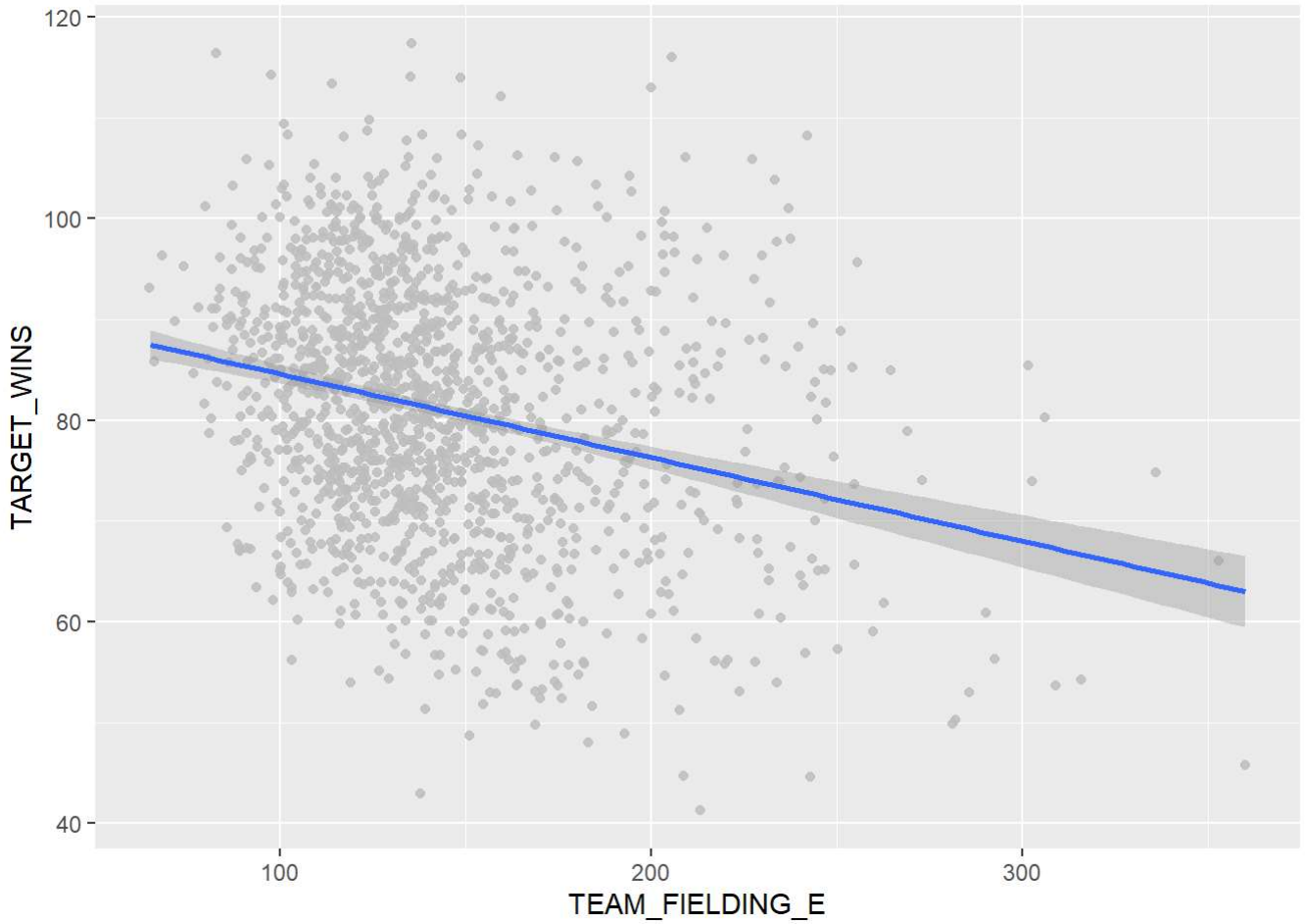
Fielding Measures

Team fielding statistics, in combination with pitching, give a picture of the team's defensive play. We should be on the lookout for correlation with pitching measures, as for example, some pitchers strategically throw pitches that by their speed and movement are less likely to result in strikeouts and more likely to be hit softly and result in double plays.

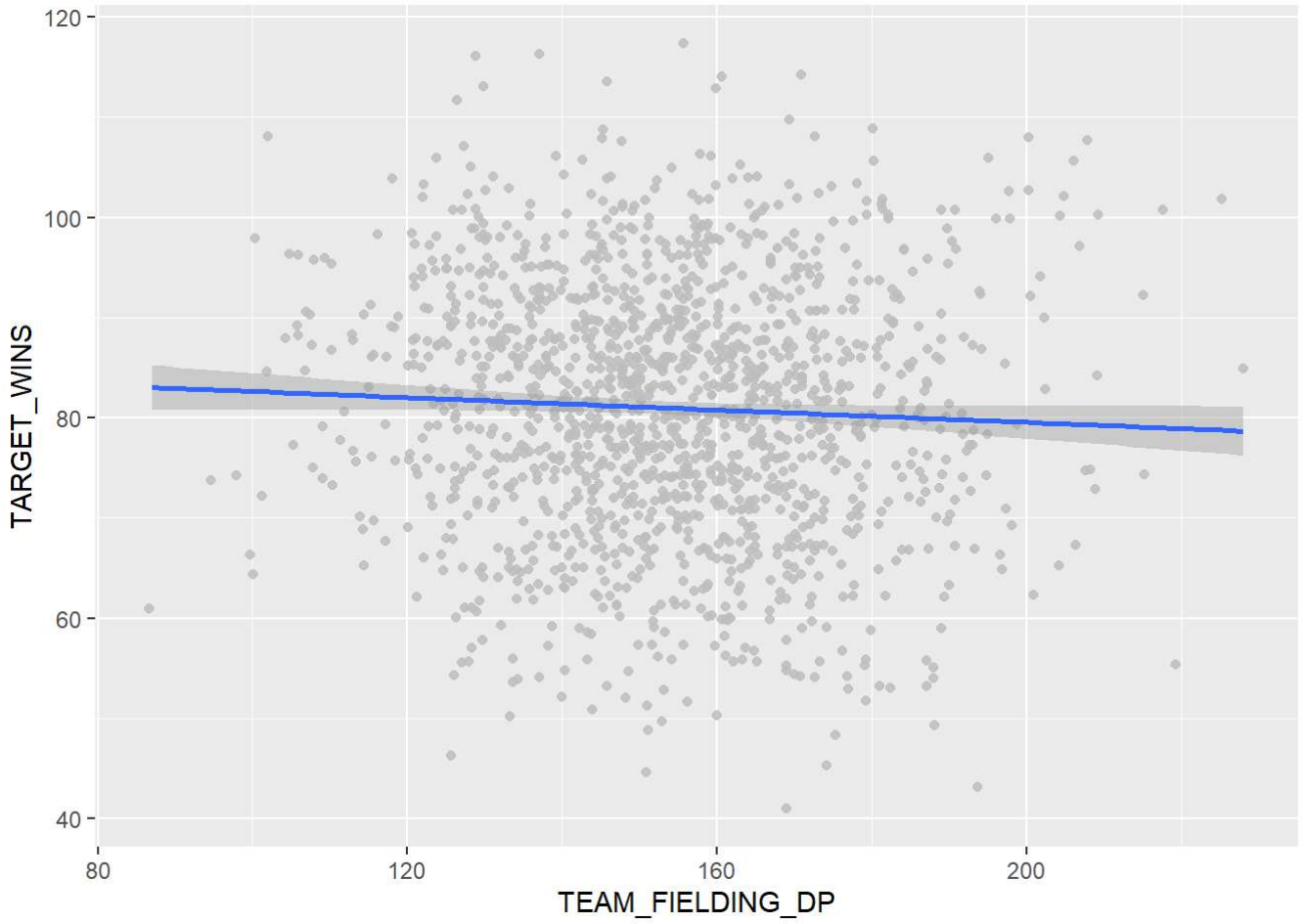
Fewer team errors are likely to result in more wins (as we might expect); however, the impact of double plays remains unclear from our initial "eyeball test".

```
fielding_plots <- lapply(colnames(moneyball_train[, 7:8]), function(c) {
  ggplot(moneyball_train, aes(moneyball_train[, c], TARGET_WINS)) + geom_jitter(color="gray", al
pha=0.85) + geom_smooth(method = "lm") + xlab(c)
 })

fielding_plots
```

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




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



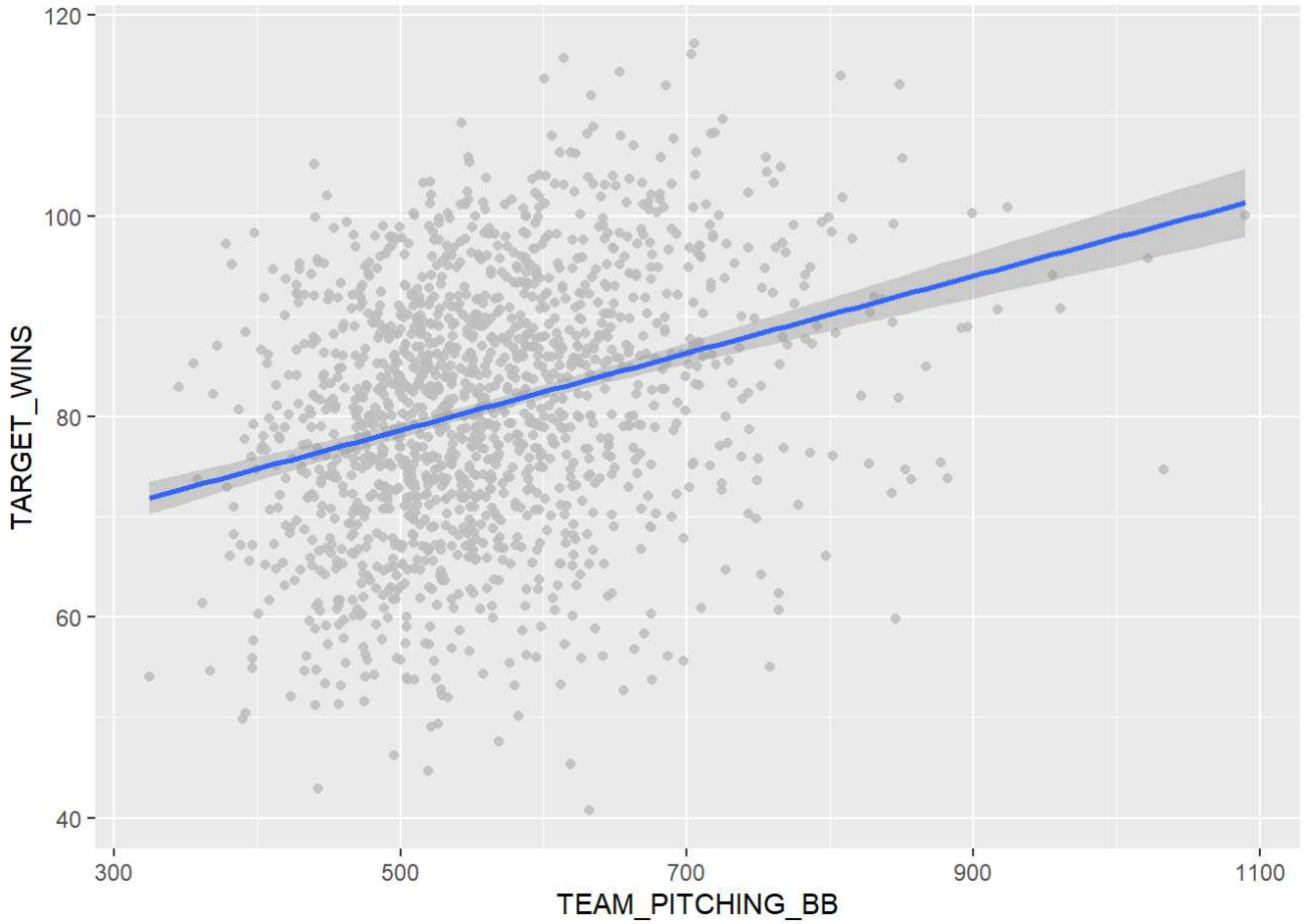
Pitching Measures

The team pitching statistics require no obvious transformations; however, their impact on target wins as evidenced in this dataset is curious. We would expect pitching BB, H, and HR to have a negative association with TARGET_WINS and SO to have a positive association, and yet…

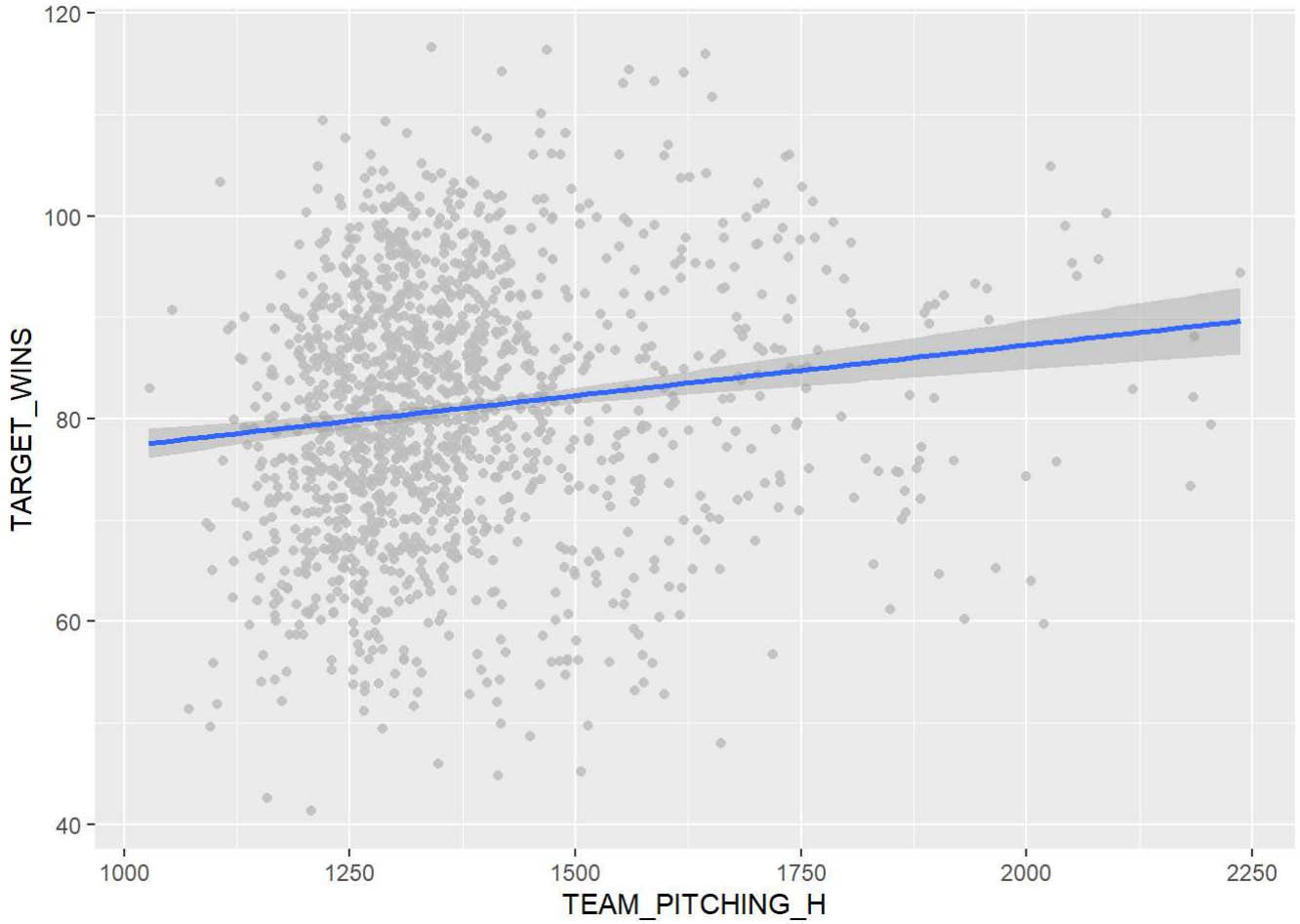
While we can't weigh in definitively on the cause for our relationships being the way they are, further investigation might explore whether the rates of pitching statistics have fluctuated over the course of time and whether they associate positively or negatively with the level of parity in the league over those same periods.

```
pitching_plots <- lapply(colnames(moneyball_train[, 9:12]), function(c) {
  ggplot(moneyball_train, aes(moneyball_train[, c], TARGET_WINS)) + geom_jitter(color="gray", al
pha=0.85) + geom_smooth(method = "lm") + xlab(c)
 })

pitching_plots
```

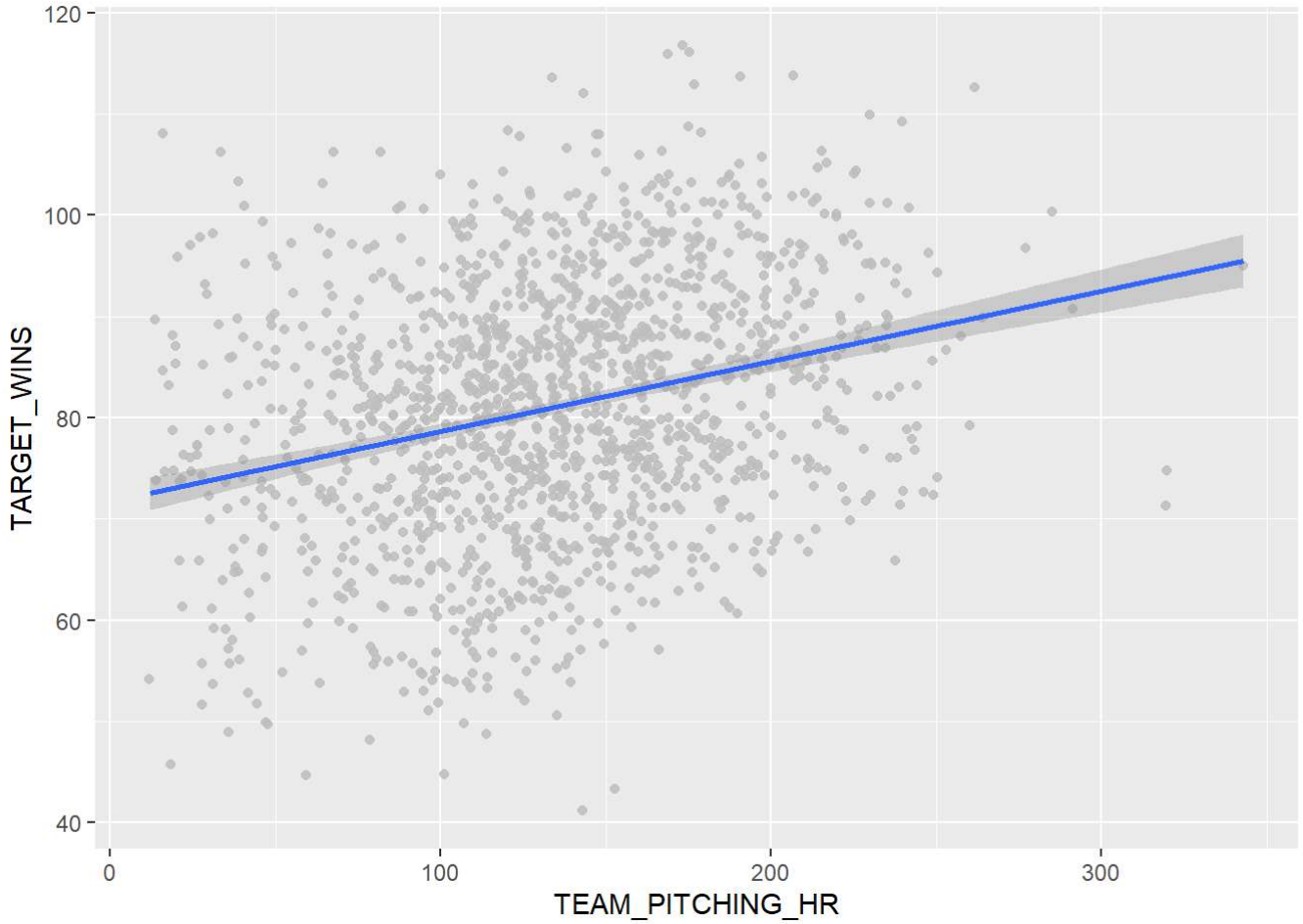
```
## [[1]]
```


120
100
80
60
40
TARGET_WINS
300
500
700
900
1100
TEAM_PITCHING_BB

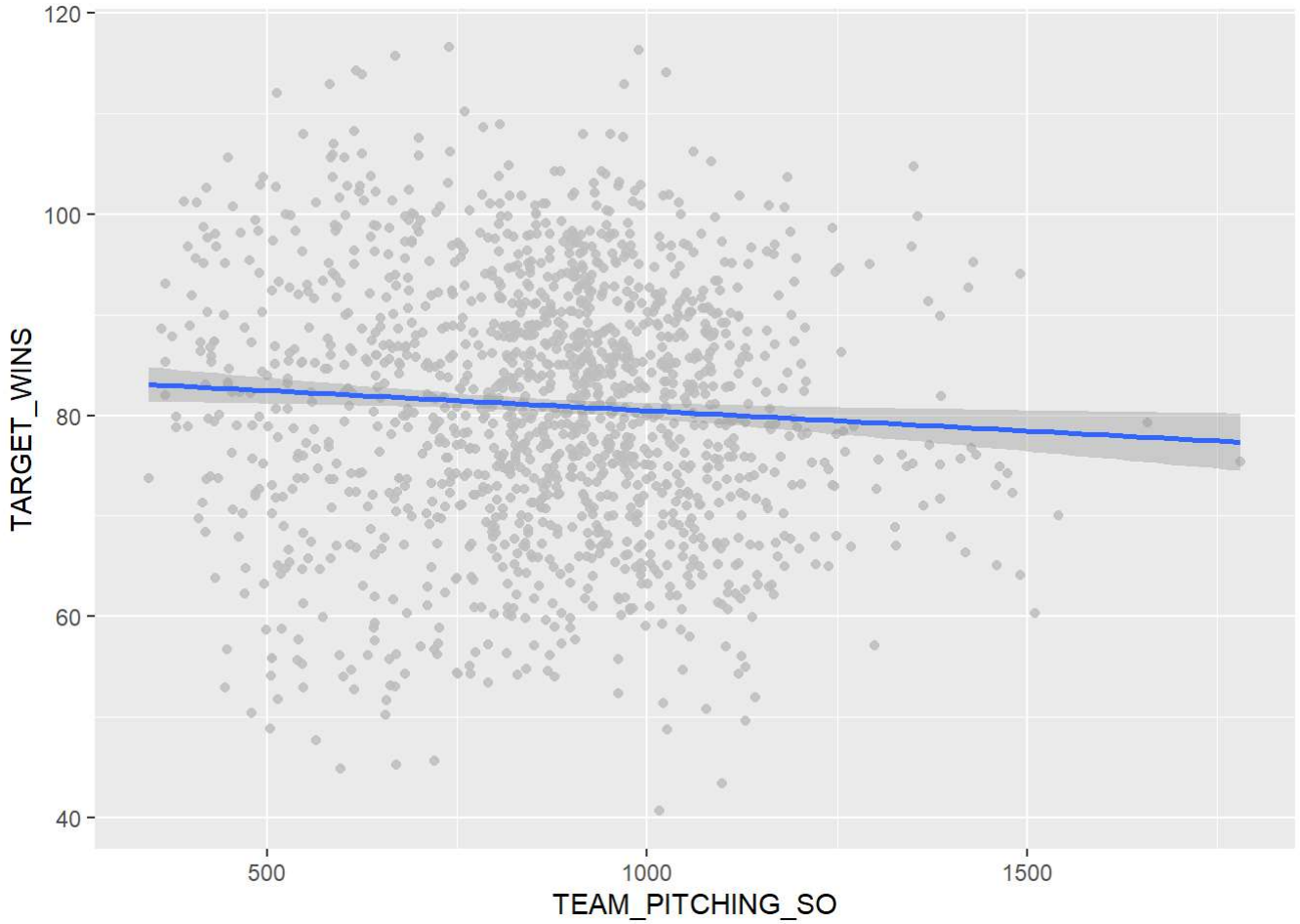
```
##
## [[2]]
```


120
100
80
60
40
TARGET_WINS
1000
1250
1500
1750
2000
2250
TEAM_PITCHING_H

```
##
## [[3]]
```




```
##
## [[4]]
```



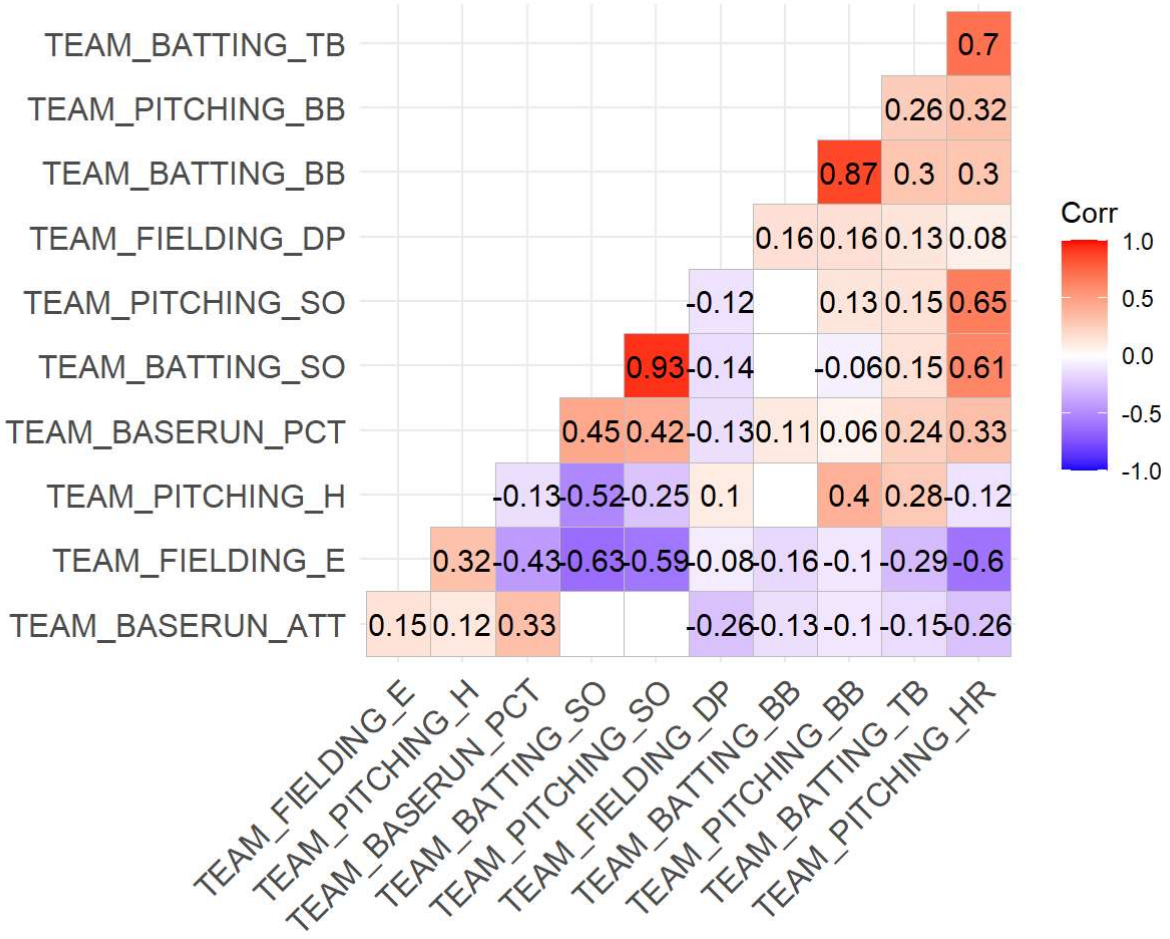
Correlation among variables

As we noted previously, correlation among variables - and thus the potential for multicolinearity in any given linear model derived from the data - is of some concern. A lower-half correlation matrix reveals this is indeed the case. These are, after all, team statistics contributing toward wins in a team sport, and we expect some reasonable amount of relatedness amongst the individual performances on offense and defense.

A few fields that stand out here:

- TEAM_BATTING_BB and TEAM_PITCHING_BB, and TEAM_BATTING_SO and TEAM_PITCHING_SO are strongly correlated with one another. This is curious, as the batting and pitching performances are, at face value, about as independent of one another as it gets in team sports. Without a plausible explanation for the correlation, we'll proceed by adding interaction terms for each relationship to our linear model.
- TEAM_FIELDING_E is negatively correlated with many of the variables in the dataset. At first glance this might seem plausible, as we'd expect a high number of fielding errors to indicate a team that generally lacked discipline or was having a "down year". However, it's notable that the columns with the highest degree of negative correlation are team batting strikeouts and team pitching strikeouts.
- TEAM_BATTING_TB and TEAM_PITCHING_HR - the high degree of positive correlation in these measures may be due to the impact of confounding variables outside this dataset. For example, homeruns and power hitting in general were notoriously rare during the "dead ball era" of the late 1800s and early 1900s when baseballs were constructed of less aerodynamic material and outfield fences ranged well over 400 yards, and have achieved record highs over the last 3 decades due variously to the use of performance enhancing drugs by hitters, the short distance of home run fences in modern baseball stadiums, and the adaptation of new hitting styles through the use of advancd motion-capture analytics.

```
corr <- cor(moneyball_train[, 2:12])
p.mat <- cor_pmat(moneyball_train[, 2:12])

ggcorrplot(corr, p.mat = p.mat, hc.order = TRUE,
    type = "lower", insig = "blank", lab=TRUE)
```




```
#http://sthda.com/english/wiki/ggcorrplot-visualization-of-a-correlation-matrix-using-ggplot2#:~:text=ggcorrplot%20main%20features%20It%20provides%20a%20solution%20for,function%20for%20computing%20a%20matrix%20of%20correlation%20p-values.
```

Create and Train a Linear Model

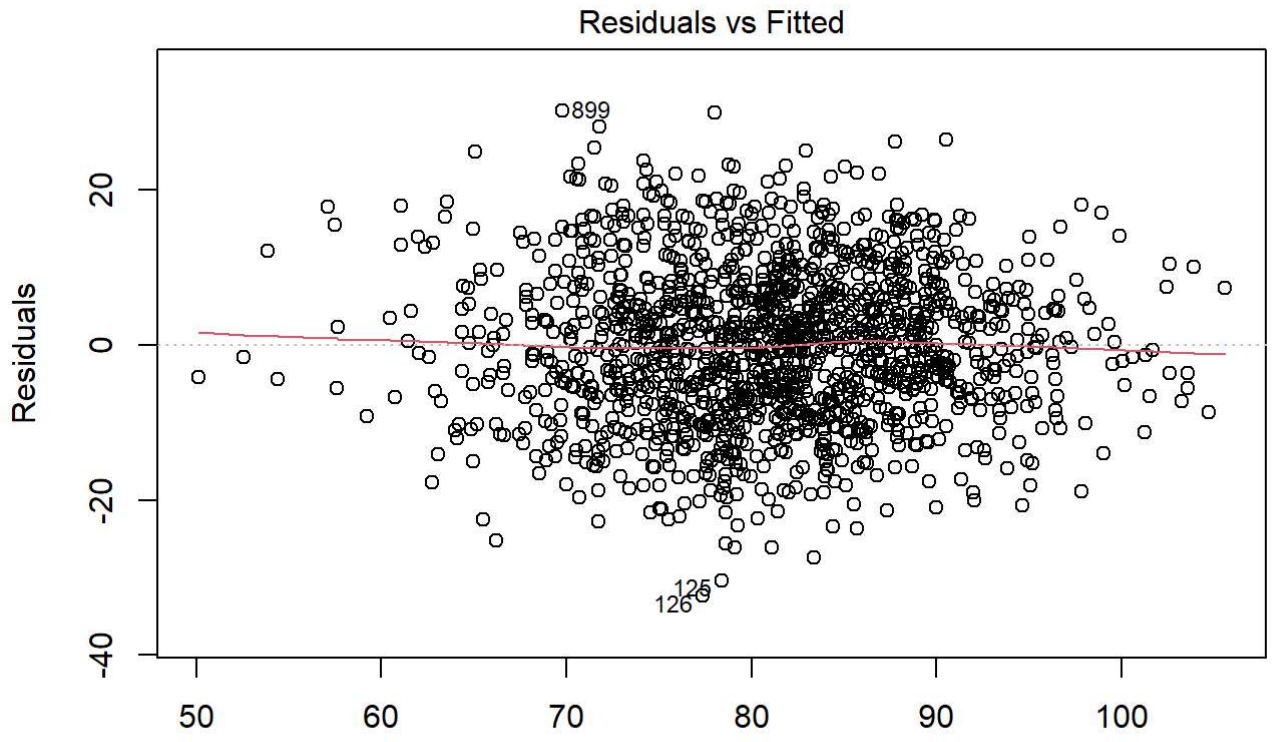
We'll start by testing a linear model with our 11 relevant statistics plus our two interaction terms. Based on the output of our `lm` and `summary` functions, we appear to be off to a promising start:

- we have an adjusted R^2 value of `0.3908` and an F-statistic that is statistically significant at the 99% level.
- we have a number of coefficients that are statistically significant at the 95% level, though most have a relatively small t-value.
- the distribution of residuals appears approximately normal, with a median about =0.133, and our diagnostic plots give us no concerns regarding outliers with high leverage or non-linear relationships.

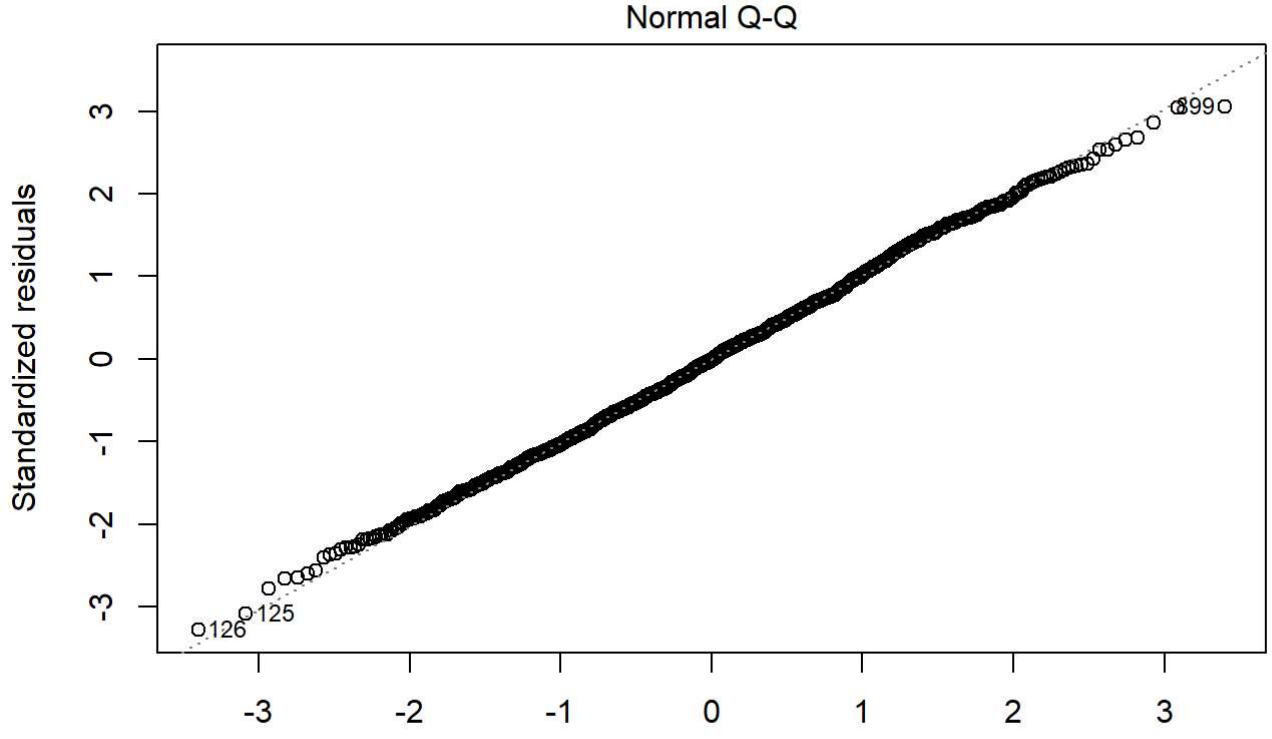
```
lm1 <- lm(TARGET_WINS ~ TEAM_BATTING_TB + TEAM_BATTING_BB + TEAM_BATTING_SO + TEAM_BASERUN_ATT +
TEAM_BASERUN_PCT +
            TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR +
            TEAM_PITCHING_SO +
            TEAM_BATTING_SO*TEAM_PITCHING_SO + TEAM_BATTING_BB*TEAM_PITCHING_BB,
          data=moneyball_train)

lm1_sum <- summary(lm1)
lm1_sum
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_TB + TEAM_BATTING_BB +
##     TEAM_BATTING_SO + TEAM_BASERUN_ATT + TEAM_BASERUN_PCT + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP + TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR +
##     TEAM_PITCHING_SO + TEAM_BATTING_SO * TEAM_PITCHING_SO + TEAM_BATTING_BB *
##     TEAM_PITCHING_BB, data = moneyball_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.373  -6.729  -0.133   6.699  30.226
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       7.337e+01  1.153e+01   6.366 2.58e-10 ***
## TEAM_BATTING_TB                  -1.699e-03  6.944e-03  -0.245 0.806763
## TEAM_BATTING_BB                   5.442e-02  3.591e-02   1.515 0.129869
## TEAM_BATTING_SO                   1.954e-02  2.139e-02   0.913 0.361145
## TEAM_BASERUN_ATT                  4.485e-02  5.141e-03   8.724  < 2e-16 ***
## TEAM_BASERUN_PCT                 -3.469e+00  4.034e+00  -0.860 0.390005
## TEAM_FIELDING_E                  -1.192e-01  9.821e-03 -12.139  < 2e-16 ***
## TEAM_FIELDING_DP                 -1.122e-01  1.365e-02  -8.225 4.24e-16 ***
## TEAM_PITCHING_BB                 -7.390e-02  4.083e-02  -1.810 0.070527 .
## TEAM_PITCHING_H                   2.645e-02  9.776e-03   2.706 0.006895 **
## TEAM_PITCHING_HR                  1.123e-01  2.924e-02   3.840 0.000128 ***
## TEAM_PITCHING_SO                 -2.172e-02  2.161e-02  -1.005 0.315023
## TEAM_BATTING_SO:TEAM_PITCHING_SO -1.496e-05  6.263e-06  -2.389 0.017025 *
## TEAM_BATTING_BB:TEAM_PITCHING_BB  5.085e-05  2.474e-05   2.055 0.040041 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.908 on 1472 degrees of freedom
## Multiple R-squared:  0.3961, Adjusted R-squared:  0.3908
## F-statistic: 74.27 on 13 and 1472 DF,  p-value: < 2.2e-16
```

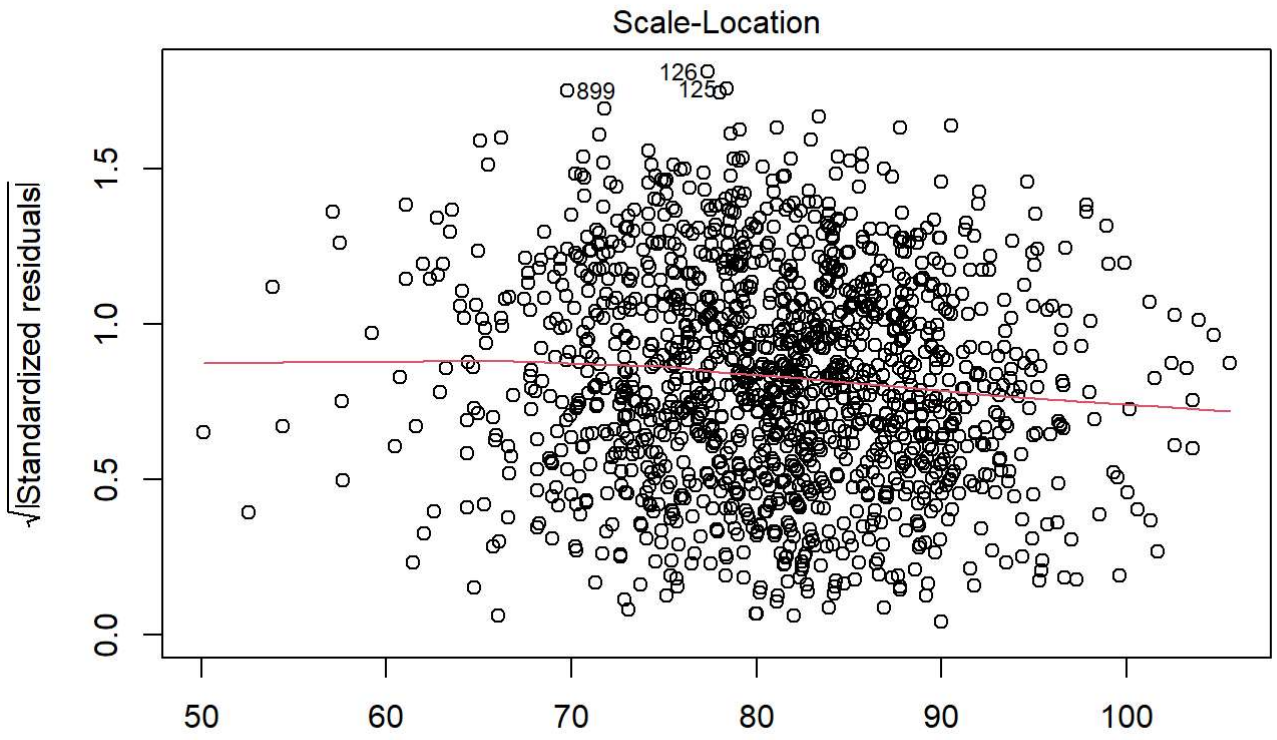
```
plot(lm1)
```



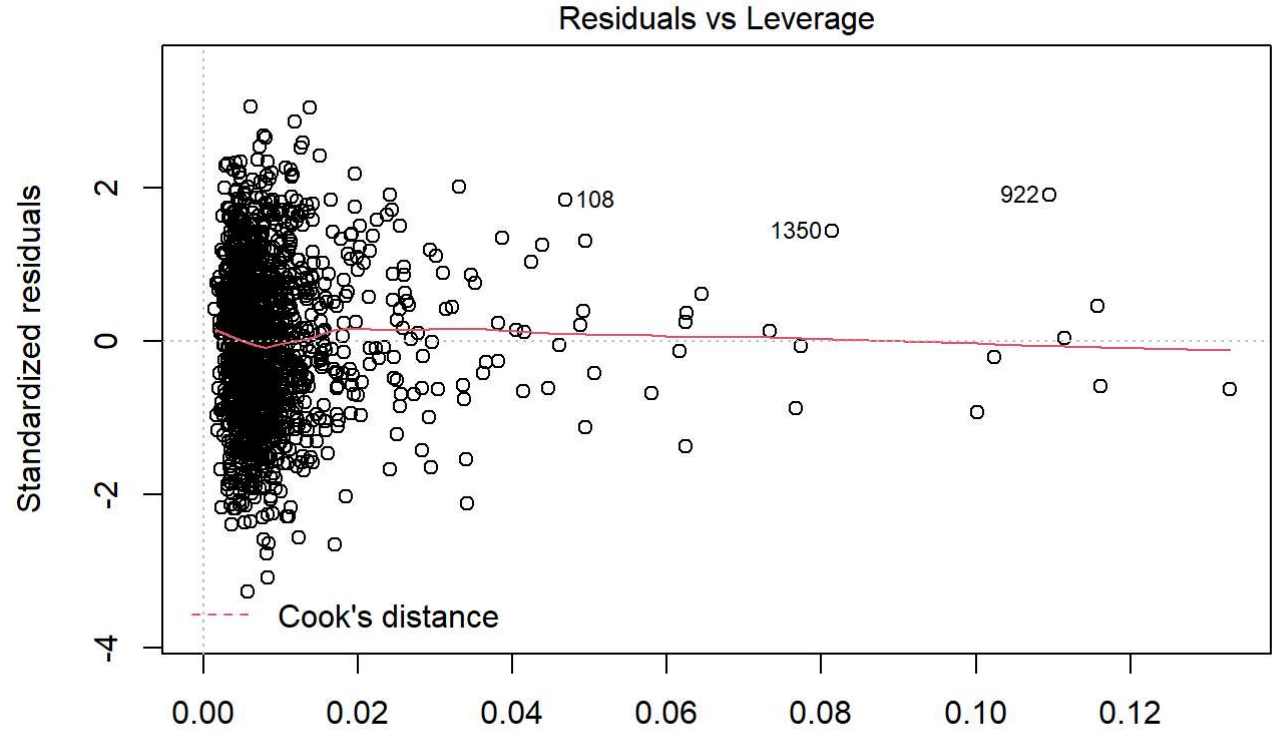
n(TARGET_WINS ~ TEAM_BATTING_TB + TEAM_BATTING_BB + TEAM_BATTING_SO + TE



n(TARGET_WINS ~ TEAM_BATTING_TB + TEAM_BATTING_BB + TEAM_BATTING_SO + TE



n(TARGET_WINS ~ TEAM_BATTING_TB + TEAM_BATTING_BB + TEAM_BATTING_SO + TE



n(TARGET_WINS ~ TEAM_BATTING_TB + TEAM_BATTING_BB + TEAM_BATTING_SO + TE

Backward Stepwise Elimination

We'll now iterate through a series of models in which we will eliminate the statistically insignificant coefficient with the lowest t-value. We'll pause when we have achieved a model with only statistically significant coefficients.

```
#Eliminate TEAM_BATTING_TB

lm2 <- lm(TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BATTING_SO +
            TEAM_BASERUN_ATT + TEAM_BASERUN_PCT +
            TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO +
            TEAM_BATTING_SO*TEAM_PITCHING_SO + TEAM_BATTING_BB*TEAM_PITCHING_BB,
          data=moneyball_train)

lm2_sum <- summary(lm2, cor=TRUE)


#Eliminate TEAM_BASERUN_PCT

lm3 <- lm(TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BATTING_SO +
            TEAM_BASERUN_ATT +
            TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO +
            TEAM_BATTING_SO*TEAM_PITCHING_SO + TEAM_BATTING_BB*TEAM_PITCHING_BB,
          data=moneyball_train)

lm3_sum <- summary(lm3, cor=TRUE)


# Eliminate TEAM_BATTING_SO and SO interaction term

lm4 <- lm(TARGET_WINS ~ TEAM_BATTING_BB +
            TEAM_BASERUN_ATT +
            TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO +
            TEAM_BATTING_BB*TEAM_PITCHING_BB,
          data=moneyball_train)

lm4_sum <- summary(lm4, cor=TRUE)

# Eliminate TEAM_BATTING_BB*TEAM_PITCHING_BB interaction term

lm5 <- lm(TARGET_WINS ~ TEAM_BATTING_BB +
            TEAM_BASERUN_ATT +
            TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_PITCHING_BB + TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO,
          data=moneyball_train)

lm5_sum <- summary(lm5, cor=TRUE)
```

Review Viable Model

The first model with only statistically significant coefficients is our 6th… model `lm6`.

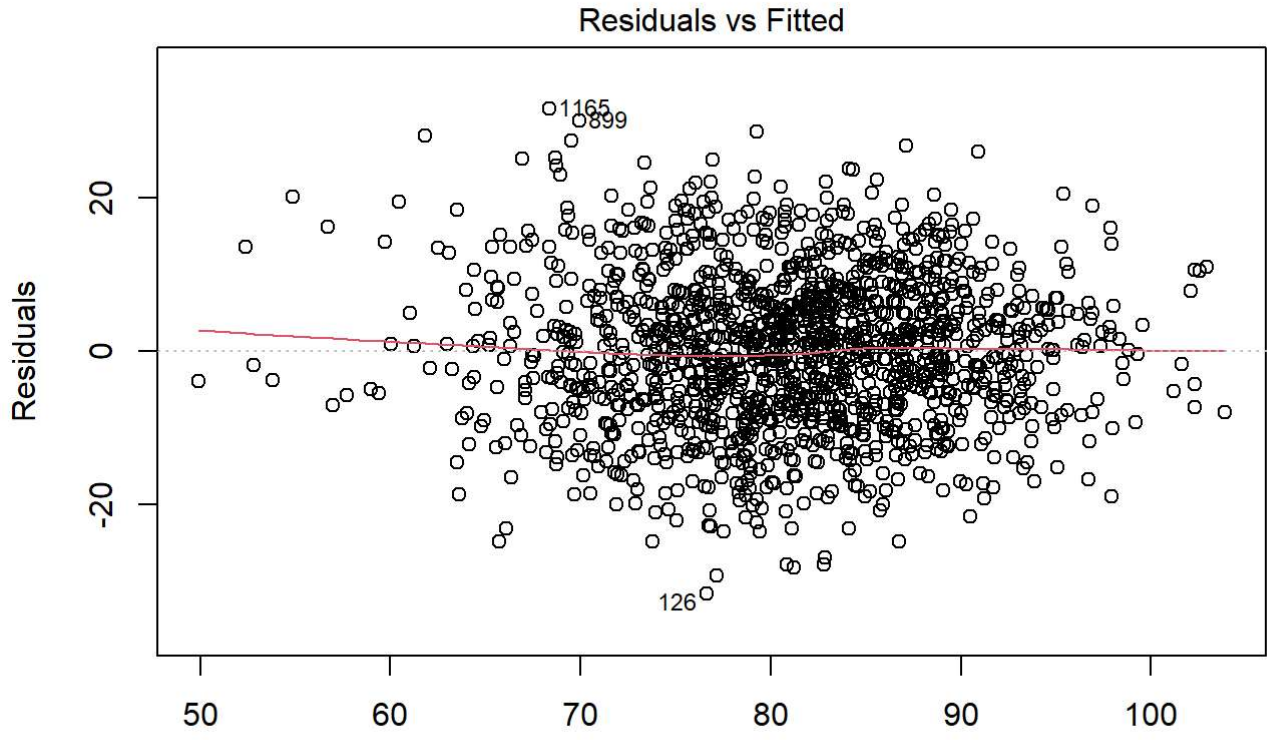
- The adjusted R^2 is still high for this model, at `0.3837`.
- The Residuals are normally distributed about a median at `0.006`; the absolute values of Min and Max residuals are nearly equal, as are the 1Q and 3Q values.
- Nothing in the residual-fitted or scale-location plots indicates a nonlinear or heteroscedastic distribution of residuals, and the outliers in our dataset do not exert unwelcome leverage.

As regards the coefficients themselves, there are no surprises here that we haven't already commented on in the analysis of scatter plots above; it is interesting, however, that TEAM_FIELDING_E and TEAM_FIELDING_DP show the greatest impact on the data.

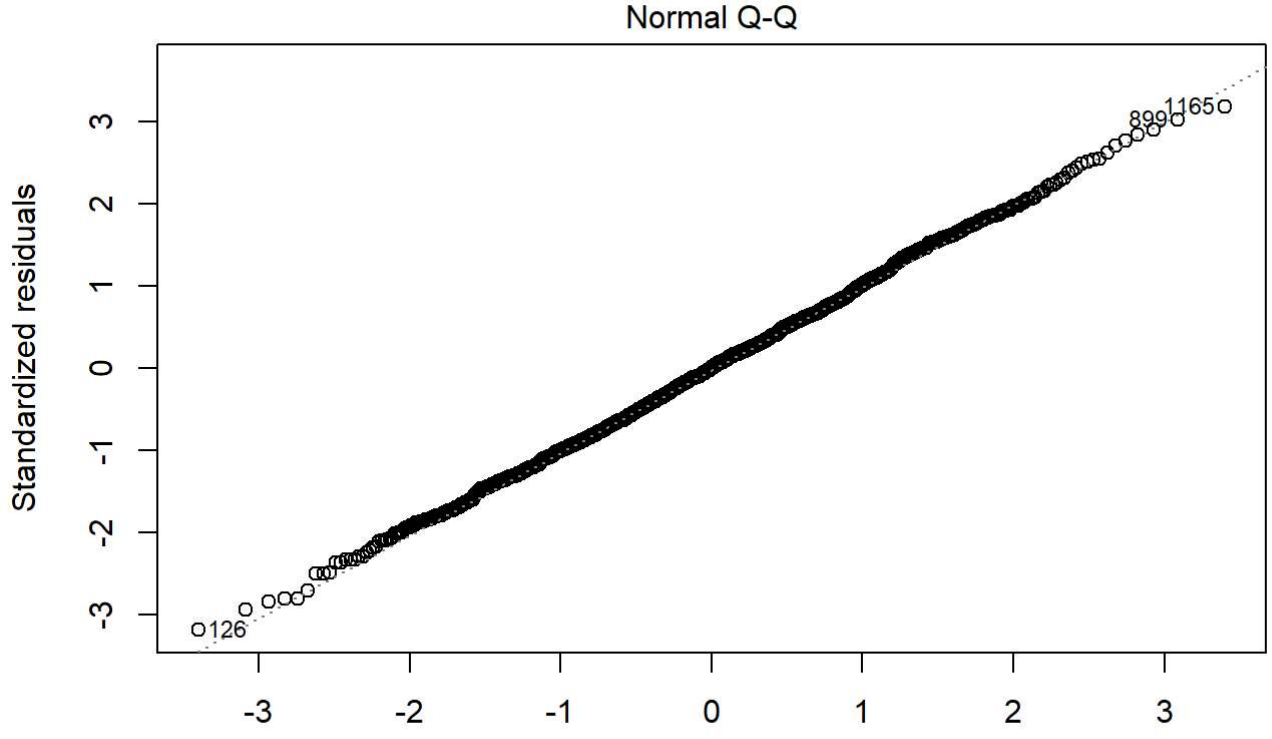
```
# Eliminate TEAM_PITCHING_BB

lm6 <- lm(TARGET_WINS ~ TEAM_BATTING_BB +
            TEAM_BASERUN_ATT +
            TEAM_FIELDING_E + TEAM_FIELDING_DP +
            TEAM_PITCHING_H + TEAM_PITCHING_HR + TEAM_PITCHING_SO,
          data=moneyball_train)

lm6_sum <- summary(lm6, cor=TRUE)
summary(lm6)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BASERUN_ATT +
##     TEAM_FIELDING_E + TEAM_FIELDING_DP + TEAM_PITCHING_H + TEAM_PITCHING_HR +
##     TEAM_PITCHING_SO, data = moneyball_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.636  -6.951   0.006   6.518  31.662
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      89.027309   4.172693  21.336  < 2e-16 ***
## TEAM_BATTING_BB   0.035825   0.003466  10.336  < 2e-16 ***
## TEAM_BASERUN_ATT  0.046653   0.004643  10.048  < 2e-16 ***
## TEAM_FIELDING_E  -0.125721   0.009207 -13.656  < 2e-16 ***
## TEAM_FIELDING_DP -0.109702   0.013651  -8.036 1.88e-15 ***
## TEAM_PITCHING_H   0.011300   0.001641   6.886 8.45e-12 ***
## TEAM_PITCHING_HR  0.110036   0.007983  13.783  < 2e-16 ***
## TEAM_PITCHING_SO -0.034248   0.001835 -18.663  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.965 on 1478 degrees of freedom
## Multiple R-squared:  0.3866, Adjusted R-squared:  0.3837
## F-statistic: 133.1 on 7 and 1478 DF,  p-value: < 2.2e-16
```

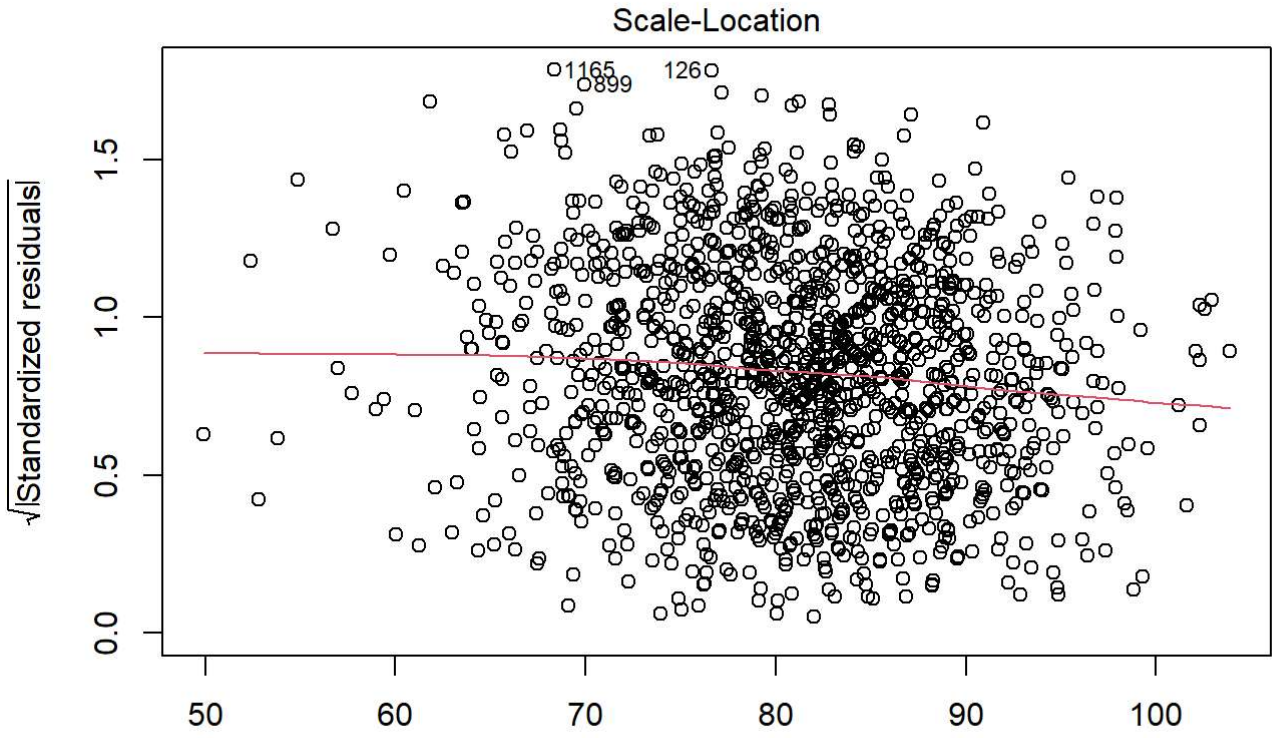
```
plot(lm6)
```



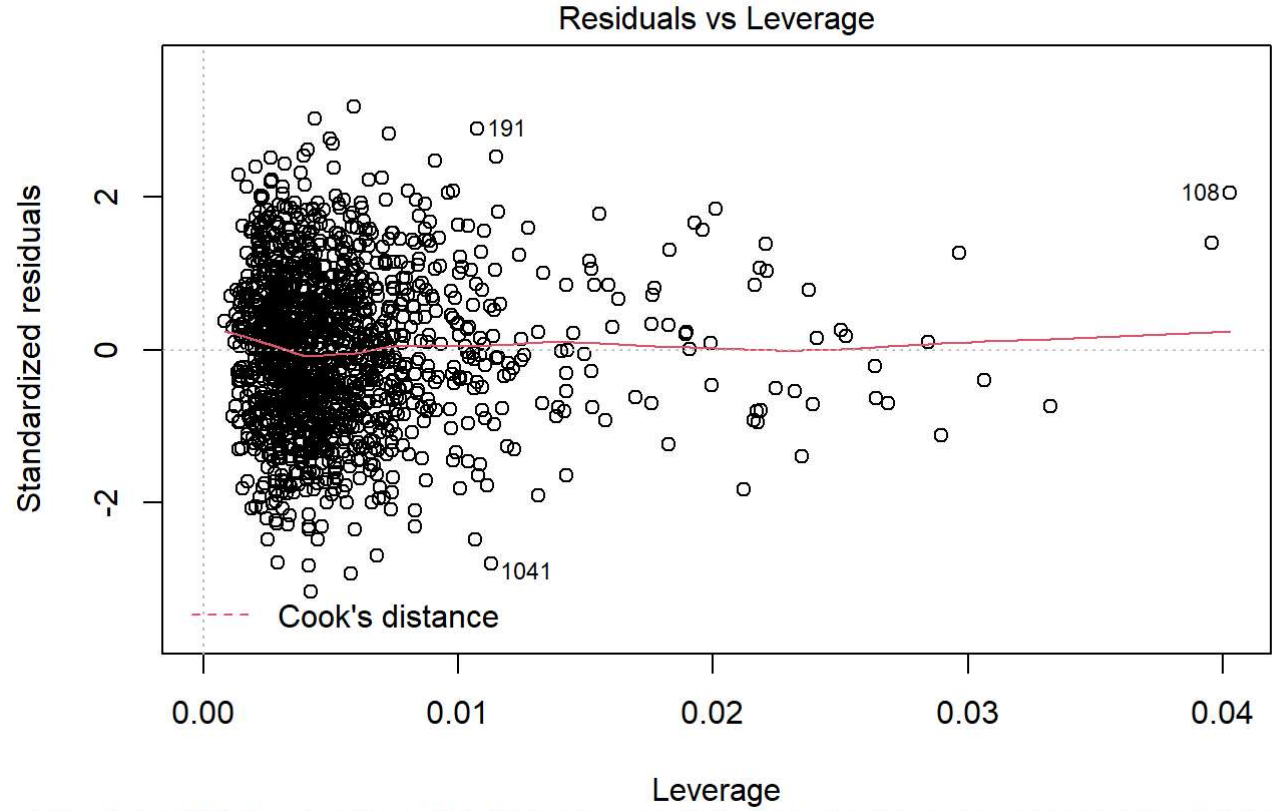
n(TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BASERUN_ATT + TEAM_FIELDING_E +



n(TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BASERUN_ATT + TEAM_FIELDING_E +



lm(TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BASERUN_ATT + TEAM_FIELDING_E +



lm(TARGET_WINS ~ TEAM_BATTING_BB + TEAM_BASERUN_ATT + TEAM_FIELDING_E +

Addressing Multicolinearity

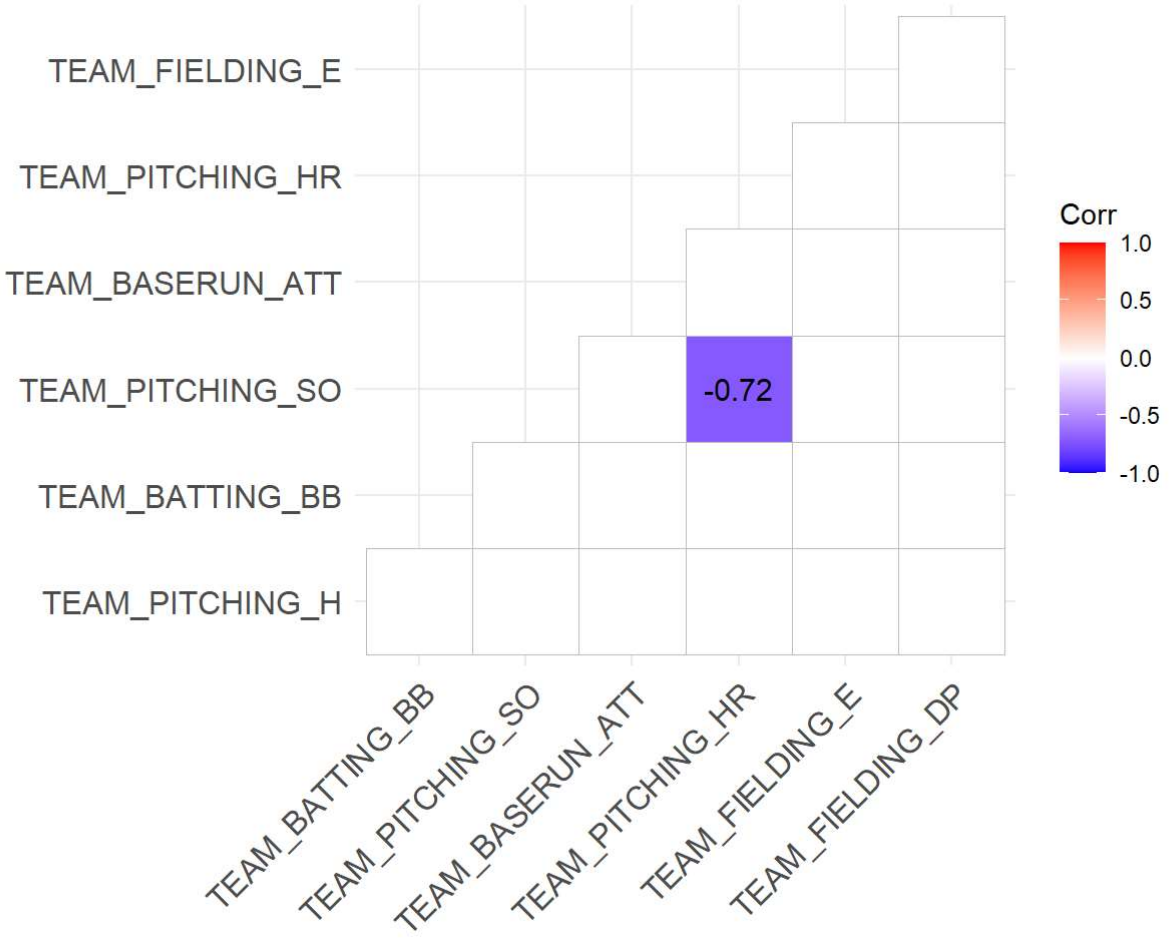
What of the potential for multicolinearity we discussed above? Displaying a correlation matrix for the coefficients in the data, we see we have one statistically-significant negative correlation between TEAM_PITCHING_SO and TEAM_PITCHING_HR.

Intuitively this isn't outside the realm of plausibility, though we'll still do our due diligence and investigate further by calculating the variance inflation factor of each item in the model. This shows the extent to which the model variance is inflated due to the colinearity of two variables. In this case, our VIF output shows values of less than `2.5`, well below the threshold of `4` at which we'd want to adjust the model.

Therefore, we can assume that the interaction of TEAM_PITCHING_SO and TEAM_PITCHING_HR is not impactul enough to warrant further adjustment to the model.

Source: https://corporatefinanceinstitute.com/resources/knowledge/other/variance-inflation-factor-vif/ (https://corporatefinanceinstitute.com/resources/knowledge/other/variance-inflation-factor-vif/)

```
# correlation matrix

corr6 <- cor(lm6_sum$correlation)[2:8, 2:8]
p.mat6 <- cor_pmat(lm6_sum$correlation)[2:8, 2:8]

ggcorr6 <- ggcorrplot(corr6, p.mat= p.mat6, hc.order = TRUE,
    type = "lower", insig = "blank", lab=TRUE)

ggcorr6
```



```
# Variance Inflation Factor
v6 <- vif(lm6)
v6
```

```
##  TEAM_BATTING_BB TEAM_BASERUN_ATT  TEAM_FIELDING_E TEAM_FIELDING_DP
##         1.166197         1.228479         1.923433         1.150906
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_SO
##         1.193338         2.469536         2.260331
```

Test the Model

Import Test Dataset and Transform Variables

We'll start by importing our evaluation dataset and transforming its columns to include the measures we developed for the previous model.

```
url1 <- "https://raw.githubusercontent.com/curdferguson/data621/main/datasets/moneyball-evaluati
on-data.csv"

moneyball_test <- url1 %>% read_csv(na='') %>% column_to_rownames(var="INDEX")
```

```
## Rows: 259 Columns: 16
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (16): INDEX, TEAM_BATTING_H, TEAM_BATTING_2B, TEAM_BATTING_3B, TEAM_BATT...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
moneyball_test1 <- moneyball_test %>% select(!c("TEAM_BATTING_HBP"))

moneyball_test <- moneyball_test1 %>% summarise(
    TEAM_BATTING_TB = ((TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR) +
2*TEAM_BATTING_2B + 3*TEAM_BATTING_3B + 4*TEAM_BATTING_HR),
    TEAM_BATTING_BB = TEAM_BATTING_BB,
    TEAM_BATTING_SO = TEAM_BATTING_SO,
    TEAM_BASERUN_ATT = TEAM_BASERUN_SB + TEAM_BASERUN_CS,
    TEAM_BASERUN_PCT = TEAM_BASERUN_SB / TEAM_BASERUN_ATT,
    TEAM_FIELDING_E = TEAM_FIELDING_E,
    TEAM_FIELDING_DP = TEAM_FIELDING_DP,
    TEAM_PITCHING_BB = TEAM_PITCHING_BB,
    TEAM_PITCHING_H = TEAM_PITCHING_H - TEAM_PITCHING_HR,
    TEAM_PITCHING_HR = TEAM_PITCHING_HR,
    TEAM_PITCHING_SO = TEAM_PITCHING_SO
)

summary(moneyball_test)
```

```
##  TEAM_BATTING_TB TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_ATT
##  Min.   :1089    Min.   : 15.0   Min.   :   0.0   Min.   :  0.00
##  1st Qu.:1934    1st Qu.:436.5   1st Qu.: 545.0   1st Qu.: 94.75
##  Median :2116    Median :509.0   Median : 686.0   Median :129.00
##  Mean   :2109    Mean   :499.0   Mean   : 709.3   Mean   :142.08
##  3rd Qu.:2294    3rd Qu.:565.5   3rd Qu.: 912.0   3rd Qu.:177.00
##  Max.   :2864    Max.   :792.0   Max.   :1268.0   Max.   :467.00
##                                  NA's   :18       NA's   :87
##  TEAM_BASERUN_PCT TEAM_FIELDING_E  TEAM_FIELDING_DP TEAM_PITCHING_BB
##  Min.   :0.2593   Min.   :  73.0   Min.   : 69.0    Min.   : 136.0
##  1st Qu.:0.5535   1st Qu.: 131.0   1st Qu.:131.0    1st Qu.: 471.0
##  Median :0.6076   Median : 163.0   Median :148.0    Median : 526.0
##  Mean   :0.6151   Mean   : 249.7   Mean   :146.1    Mean   : 552.4
##  3rd Qu.:0.6827   3rd Qu.: 252.0   3rd Qu.:164.0    3rd Qu.: 606.5
##  Max.   :0.8114   Max.   :1568.0   Max.   :204.0    Max.   :2008.0
##  NA's   :88                        NA's   :31
##  TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_SO
##  Min.   : 1124   Min.   :  0.0    Min.   :   0.0
##  1st Qu.: 1300   1st Qu.: 52.0    1st Qu.: 613.0
##  Median : 1402   Median :104.0    Median : 745.0
##  Mean   : 1711   Mean   :102.1    Mean   : 799.7
##  3rd Qu.: 1606   3rd Qu.:142.5    3rd Qu.: 938.0
##  Max.   :22768   Max.   :336.0    Max.   :9963.0
##                                   NA's   :18
```

Calculate predictions using the training model

```
moneyball_test <- moneyball_test %>% mutate(predicted_wins = lm6$coefficients[1] + TEAM_BATTING_
BB*lm6$coefficients[2] + TEAM_BASERUN_ATT*lm6$coefficients[3] +  TEAM_FIELDING_E*lm6$coefficient
s[4] + TEAM_FIELDING_DP*lm6$coefficients[5] + TEAM_PITCHING_H*lm6$coefficients[6] + TEAM_PITCHIN
G_HR*lm6$coefficients[7] + TEAM_PITCHING_SO*lm6$coefficients[8])
```

```
```