BHao_HW1

```
train_raw = read.csv("~/Google Drive/CUNY/git/DATA621/HW1/moneyball-training-data.csv")
```

Data Exploration

Our primary objectives for data exploration are two fold:

1) To get a general sense for our data via summary statistics and visualizations

2) To highlight potential data integrity issues that will need to be addressed in the following data preparation phase

Note: data exploration, data preparation and model building may be somewhat iterative processes

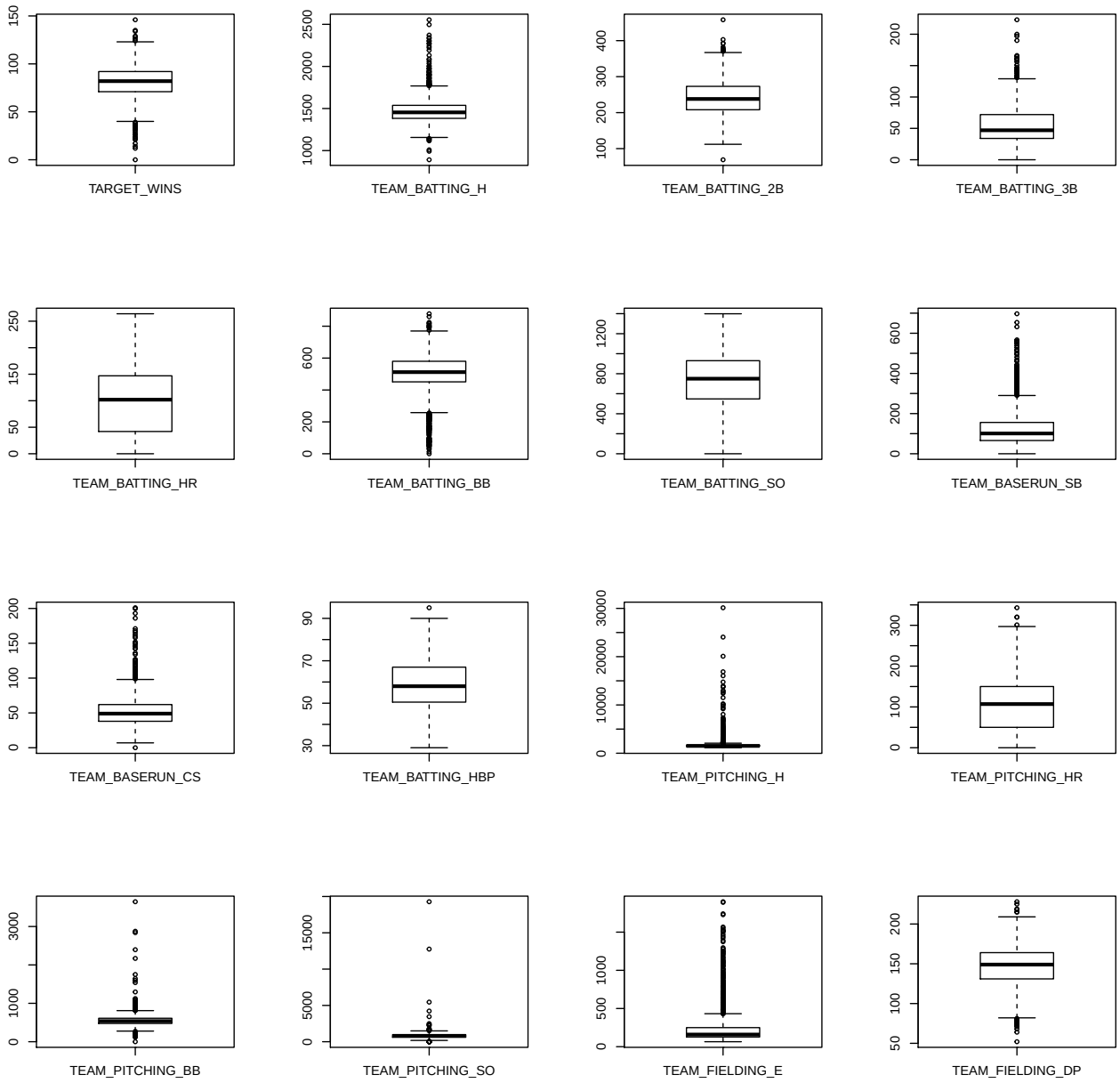
- We start by summarizing our data. A couple things immediately jump out:
  - 0 minimum values, especially in the TARGET_WINS column
  - NAs in several columns

```
##      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
```

- Next, let's look at each column's distribution graphically with some box plots and outline the highlights below:
  - TARGET_WINS: There are a large number of outliers
  - TEAM_BATTING_H: There are a large number of outliers, especially in the higher values
  - TEAM_BATTING_3B: The lower whisker is close to zero, and there are many outliers in the higher values
  - TEAM_BATTING_BB: The IQR is relatively tight; there are a large number of outliers
  - TEAM_BASERUN_SB: There are a large number of outliers in the higher values
  - TEAM_BASERUN_CS: There are a large number of outliers in the higher values
  - TEAM_PITCHING_H: There are an extremely large number of outliers in the higher values
  - TEAM_PITCHING_BB: There are an extremely large number of outliers in the higher values
  - TEAM_PITCHING_SO: There are an extremely large number of outliers in the higher values
  - TEAM_FIELDING_E: There are an extremely large number of outliers in the higher values

Note: obviously, not having access to the data providers makes it more difficult to make sense of these potential anomalies; however, knowing the total number of at bats and pitches would be very helpful to this end.



- Next, we examine the correlation between each column and the response variable TARGET_WINS:
  - TEAM_BATTING_H exhibits the highest correlation to the response variable, while TEAM_FIELDING_E exhibits the lowest correlation
  - Surprisingly both TEAM_PITCHING_HR and TEAM_PITCHING_BB exhibit positive correlations to the response variable; we will revisit these anomalies after the data cleansing process to see if they still appear

```
##             rowname   With NAs Ignoring NAs
## 1    TEAM_BATTING_H  0.3887675   0.38876752
## 2   TEAM_BATTING_2B  0.2891036   0.28910365
## 3   TEAM_BATTING_BB  0.2325599   0.23255986
## 4  TEAM_PITCHING_HR  0.1890137   0.18901373
## 5   TEAM_BATTING_HR  0.1761532   0.17615320
## 6   TEAM_BATTING_3B  0.1426084   0.14260841
## 7   TEAM_BASERUN_SB         NA   0.13513892
```

```
## 8  TEAM_PITCHING_BB  0.1241745   0.12417454
## 9  TEAM_BATTING_HBP         NA   0.07350424
## 10  TEAM_BASERUN_CS         NA   0.02240407
## 11  TEAM_BATTING_SO         NA  -0.03175071
## 12 TEAM_FIELDING_DP         NA  -0.03485058
## 13 TEAM_PITCHING_SO         NA  -0.07843609
## 14  TEAM_PITCHING_H -0.1099371  -0.10993705
## 15  TEAM_FIELDING_E -0.1764848  -0.17648476
```

- Data exploration summary: The exploration phase revealed that there are a number of potential data issues that will need to be addressed in the data preparation phase:
  - 0 values
  - Missing values
  - Outliers
  - Missing variables - again, it would be very helpful to have the total number of 'at bats' and 'pitches'; however, we can make do without those variables

Data Preparation

In this phase, we will look at each anomaly and decide how best to address it so that our models can work correctly.

- Since the TEAM_BATTING_H column is inclusive of three other columns, we will start by replacing it with a TEAM_BATTING_1B column

We'll replace TEAM_BATTING_H with TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B - TEAM_BATTING_HR.

```
train_clean = train_raw
train_clean = train_clean %>% mutate(TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3
  select(-TEAM_BATTING_H)
```

- Let's next look at the row with 0 wins
  - Most of its columns are either 0 or NA
  - Furthermore, it has a value for TEAM_PITCHING_H (24057) which just does not make sense as that would equate to giving up 148.5 hits on average over 162 games. The median for this field is 1518 which is close to the median value for TEAM_BATTING_H of 1454 - which makes sense as they should be nearly mirror images of each other on average

It's safe to assume that this row contains faulty data and should be removed.

```
train_clean %>% filter(TARGET_WINS == 0)
```

```
##   INDEX TARGET_WINS TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## 1  1347           0              135               0               0
##   TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
## 1               0               0               0               0
##   TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
## 1               NA           24057                0                0
##   TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP TEAM_BATTING_1B
## 1                0            1890               NA             756
```

```
train_clean = train_clean %>% filter(TARGET_WINS != 0)
```

- Let's turn our attention to columns with missing data
  - Specifically let's start with TEAM_BATTING_SO and TEAM_PITCHING_SO, both of which are missing values for the same 102 rows
  - These two columns should be nearly mirror images of each other on average

We'll fill in the missing values using their respective median values.

```
train_clean = train_clean %>% mutate(
  TEAM_PITCHING_SO = ifelse(is.na(TEAM_PITCHING_SO), median(TEAM_PITCHING_SO, na.rm = TRUE), TEAM_PITCH
  TEAM_BATTING_SO = ifelse(is.na(TEAM_BATTING_SO), median(TEAM_BATTING_SO, na.rm = TRUE), TEAM_BATTING_S
```

We'll use the same approach to fill in the missing values for TEAM_BASERUN_SB, TEAM_BASERUN_CS and TEAM_FIELDING_DP. While we could have used the ratio of average 'stolen bases' to 'caught steals' multiplied by the 'stolen bases' column, we would often then be estimating based on an estimate when the 'stolen bases' itself was filled in.

```
train_clean = train_clean %>% mutate(
  TEAM_BASERUN_SB = ifelse(is.na(TEAM_BASERUN_SB), median(TEAM_BASERUN_SB, na.rm = TRUE), TEAM_BASERUN_S
  TEAM_BASERUN_CS = ifelse(is.na(TEAM_BASERUN_CS), median(TEAM_BASERUN_CS, na.rm = TRUE), TEAM_BASERUN_C
  TEAM_FIELDING_DP = ifelse(is.na(TEAM_FIELDING_DP), median(TEAM_FIELDING_DP, na.rm = TRUE), TEAM_FIELDI
```

- Missing data (cont.)
  - The final column with NAs is the TEAM_BATTING_HBP. Since there are 2084 rows with missing data, it may be best to create a categorical variable to indicate whether TEAM_BATTING_HBP exists or not

We'll add a new variable TEAM_BATTING_HBP_YN that is 1 when the TEAM_BATTING_HBP exists and 0 when it does not.

```
train_clean = train_clean %>% mutate(TEAM_BATTING_HBP_YN = ifelse(is.na(TEAM_BATTING_HBP), 0, 1))
summary(train_clean)
```

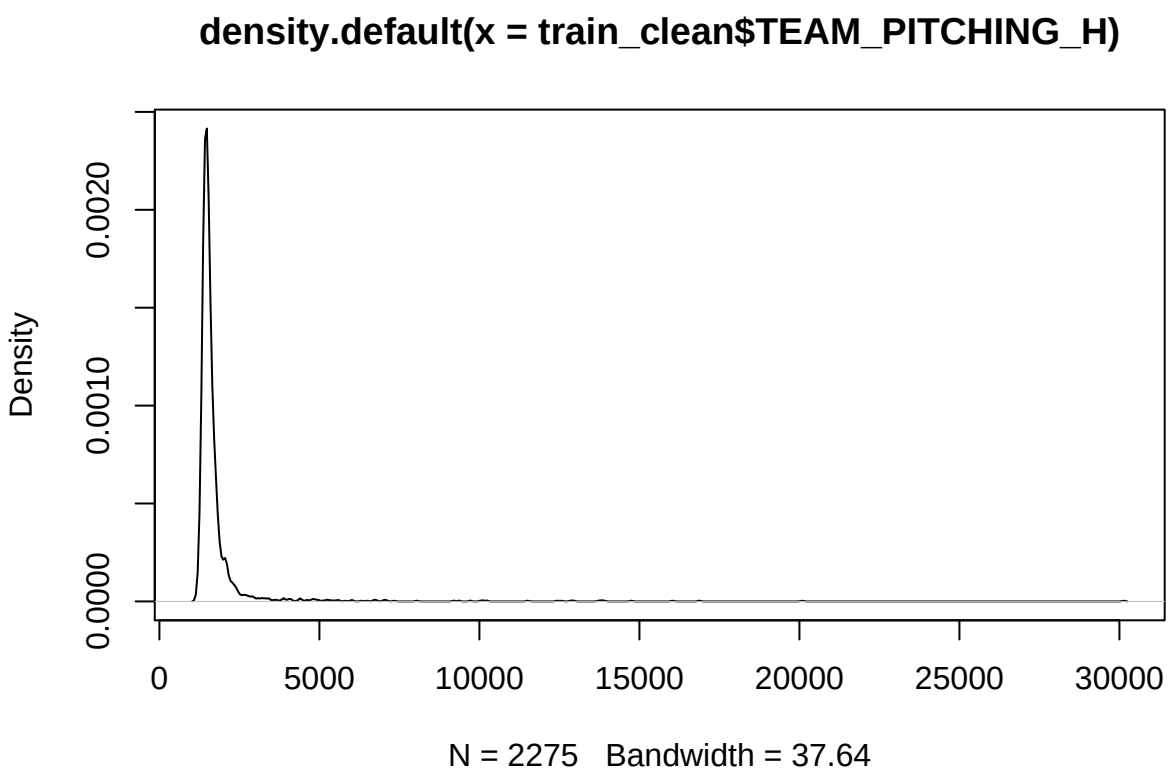
```
##      INDEX         TARGET_WINS     TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   :   1.0   Min.   : 12.00   Min.   : 69.0   Min.   :  0.00
##  1st Qu.: 630.5   1st Qu.: 71.00   1st Qu.:208.0   1st Qu.: 34.00
##  Median :1270.0   Median : 82.00   Median :238.0   Median : 47.00
##  Mean   :1268.4   Mean   : 80.83   Mean   :241.3   Mean   : 55.27
##  3rd Qu.:1916.0   3rd Qu.: 92.00   3rd Qu.:273.0   3rd Qu.: 72.00
##  Max.   :2535.0   Max.   :146.00   Max.   :458.0   Max.   :223.00
##
##  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
##  Min.   :  0.00   Min.   : 12.0   Min.   :   0.0   Min.   :  0.0
##  1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 557.5   1st Qu.: 67.0
##  Median :102.00   Median :512.0   Median : 750.0   Median :101.0
##  Mean   : 99.66   Mean   :501.8   Mean   : 736.6   Mean   :123.4
##  3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 925.0   3rd Qu.:151.0
##  Max.   :264.00   Max.   :878.0   Max.   :1399.0   Max.   :697.0
##
##  TEAM_BASERUN_CS  TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
##  Min.   :  7.00   Min.   :29.00    Min.   : 1137   Min.   :  0.0
##  1st Qu.: 44.00   1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0
##  Median : 49.00   Median :58.00    Median : 1518   Median :107.0
##  Mean   : 51.54   Mean   :59.36    Mean   : 1769   Mean   :105.7
##  3rd Qu.: 54.50   3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0
##  Max.   :201.00   Max.   :95.00    Max.   :30132   Max.   :343.0
##                   NA's   :2084
##  TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
```

```
##  Min.   : 119.0   Min.   :    0.0   Min.   :  65.0   Min.   : 52.0
##  1st Qu.: 476.0   1st Qu.:  626.0   1st Qu.: 127.0   1st Qu.:134.0
##  Median : 537.0   Median :  814.0   Median : 159.0   Median :149.0
##  Mean   : 553.3   Mean   :  817.9   Mean   : 245.8   Mean   :146.7
##  3rd Qu.: 611.0   3rd Qu.:  957.0   3rd Qu.: 249.0   3rd Qu.:161.5
##  Max.   :3645.0   Max.   :19278.0   Max.   :1898.0   Max.   :228.0
##
##  TEAM_BATTING_1B TEAM_BATTING_HBP_YN
##  Min.   : 709    Min.   :0.00000
##  1st Qu.: 991    1st Qu.:0.00000
##  Median :1050    Median :0.00000
##  Mean   :1073    Mean   :0.08396
##  3rd Qu.:1129    3rd Qu.:0.00000
##  Max.   :2112    Max.   :1.00000
##
```

- Looking deeper into the TEAM_PITCHING_H column, we see that it contains a large number of outliers
  - Quantifying that shows 213 rows with TEAM_PITCHING_H values greater than 1.5 IQRs above the 3rd quartile

Instead of removing these rows since they still might contain useful data in the other columns, let's create a flag to indicate which rows contain outliers in this particular column - this happens to address outliers across all of the pitcher-related columns. Let's also do the same for other column categories as well.

```
plot(density(train_clean$TEAM_PITCHING_H))
```




```
train_clean %>% filter(TEAM_PITCHING_H > quantile(TEAM_PITCHING_H, 0.75) + 1.5 * IQR(TEAM_PITCHING_H)) %
  summarise(n = n())
```

```
##     n
## 1 213
```

```
train_clean = train_clean %>%
  mutate('PITCHER_OUTLIER_YN' = ifelse(TEAM_PITCHING_H > quantile(TEAM_PITCHING_H, 0.75) + 1.5 * IQR(TE
                                       TEAM_PITCHING_H < quantile(TEAM_PITCHING_H, 0.25) - 1.5 * IQR(TE
                                       1, 0))
train_clean = train_clean %>%
  mutate('BATTING_OUTLIER_YN' = ifelse(TEAM_BATTING_1B > quantile(TEAM_BATTING_1B, 0.75) + 1.5 * IQR(TE
                                       TEAM_BATTING_1B < quantile(TEAM_BATTING_1B, 0.25) - 1.5 * IQR(TE
                                       1, 0))
train_clean = train_clean %>%
  mutate('BATTING_OUTLIER_YN' = ifelse(TEAM_BATTING_1B > quantile(TEAM_BATTING_1B, 0.75) + 1.5 * IQR(TE
                                       TEAM_BATTING_1B < quantile(TEAM_BATTING_1B, 0.25) - 1.5 * IQR(TE
                                       1, 0))
train_clean = train_clean %>%
  mutate('BASERUN_OUTLIER_YN' = ifelse(TEAM_BASERUN_SB > quantile(TEAM_BASERUN_SB, 0.75) + 1.5 * IQR(TE
                                       TEAM_BASERUN_SB < quantile(TEAM_BASERUN_SB, 0.25) - 1.5 * IQR(TE
                                       1, 0))
train_clean = train_clean %>%
  mutate('FIELDING_OUTLIER_YN' = ifelse(TEAM_FIELDING_E > quantile(TEAM_FIELDING_E, 0.75) + 1.5 * IQR(T
                                       TEAM_FIELDING_E < quantile(TEAM_FIELDING_E, 0.25) - 1.5 * IQR(TE
                                       1, 0))
```

- Next, we'll create some new variables based on ratios of the existing variables. Not only will this scale and normalize the variables, it may also create variables that have a better explain the response variable
  - TARGET_WINS_Ratio = TARGET_WINS / 162 (i.e. the percentage of wins)
  - TEAM_H_Ratio = (TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR) / TEAM_PITCHING_H (i.e. the ratio of hits earned to hits allowed)
  - TEAM_BASERUN_Ratio = TEAM_BASERUN_SB / TEAM_BASERUN_CS (i.e. the ratio of successful steals to unsuccessful ones)
  - TEAM_HR_SO_Ratio = TEAM_BATTING_HR / TEAM_BATTING_SO (i.e. the ratio of home runs to strikeouts)

```
train_clean = train_clean %>%
  mutate(TARGET_WINS_Ratio = TARGET_WINS / 162,
         TEAM_H_Ratio = (TEAM_BATTING_1B + TEAM_BATTING_2B + TEAM_BATTING_3B + TEAM_BATTING_HR) / TEAM_
         TEAM_BASERUN_Ratio = TEAM_BASERUN_SB / TEAM_BASERUN_CS,
         TEAM_HR_SO_Ratio = TEAM_BATTING_HR / ifelse(TEAM_BATTING_SO == 0, median(TEAM_BATTING_SO), TEA
```

- Let's now revisit our earlier data explorations. Well, we certainly removed the missing values, but it does not yet look like our new variables will be very promising

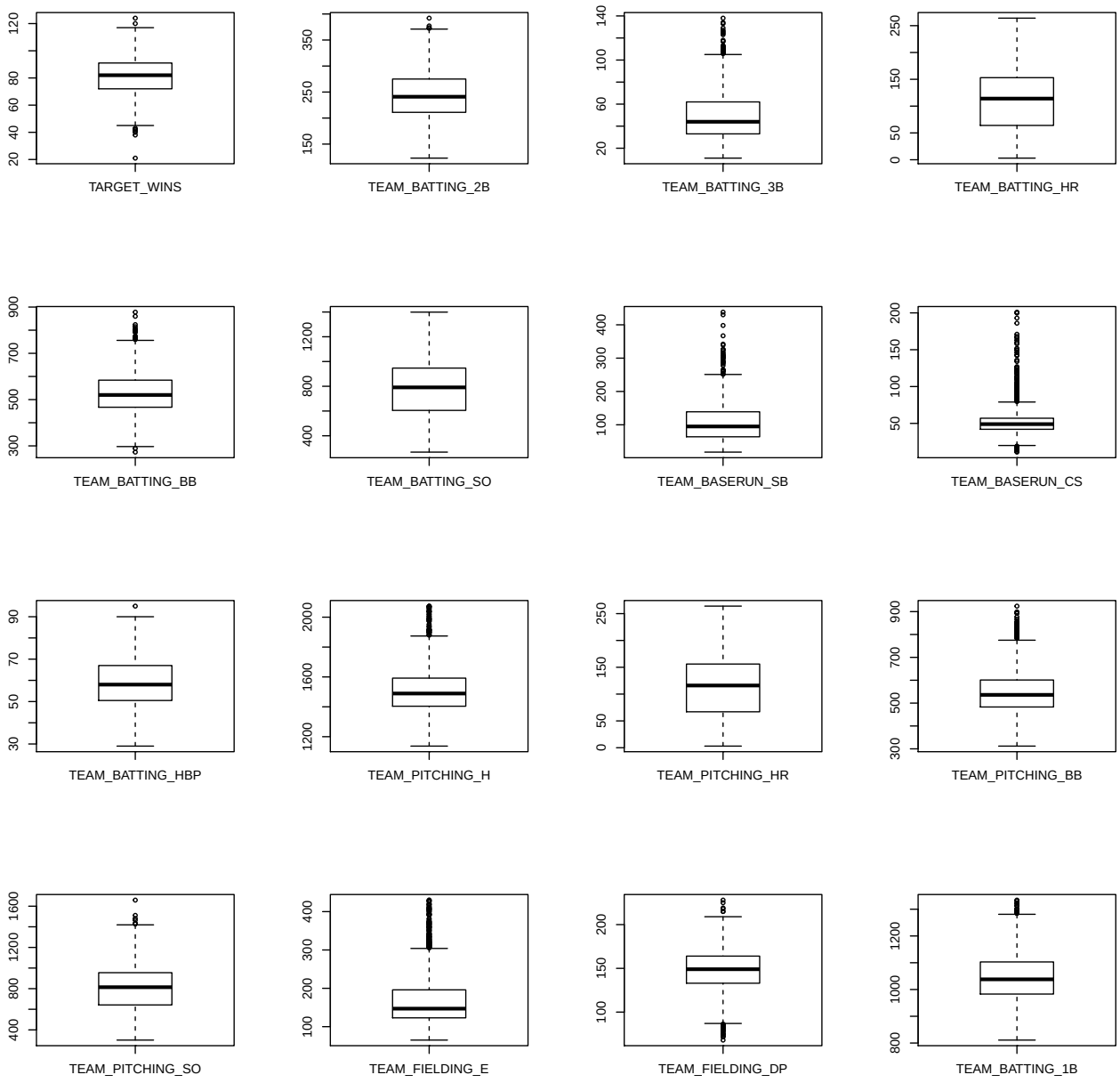
```
##      INDEX         TARGET_WINS     TEAM_BATTING_2B TEAM_BATTING_3B
##  Min.   :   1.0   Min.   : 12.00   Min.   : 69.0   Min.   :  0.00
##  1st Qu.: 630.5   1st Qu.: 71.00   1st Qu.:208.0   1st Qu.: 34.00
##  Median :1270.0   Median : 82.00   Median :238.0   Median : 47.00
##  Mean   :1268.4   Mean   : 80.83   Mean   :241.3   Mean   : 55.27
##  3rd Qu.:1916.0   3rd Qu.: 92.00   3rd Qu.:273.0   3rd Qu.: 72.00
##  Max.   :2535.0   Max.   :146.00   Max.   :458.0   Max.   :223.00
##
##  TEAM_BATTING_HR  TEAM_BATTING_BB TEAM_BATTING_SO  TEAM_BASERUN_SB
##  Min.   :  0.00   Min.   : 12.0   Min.   :   0.0   Min.   :  0.0
```

```
##  1st Qu.: 42.00   1st Qu.:451.0   1st Qu.: 557.5   1st Qu.: 67.0
##  Median :102.00   Median :512.0   Median : 750.0   Median :101.0
##  Mean   : 99.66   Mean   :501.8   Mean   : 736.6   Mean   :123.4
##  3rd Qu.:147.00   3rd Qu.:580.0   3rd Qu.: 925.0   3rd Qu.:151.0
##  Max.   :264.00   Max.   :878.0   Max.   :1399.0   Max.   :697.0
##
##  TEAM_BASERUN_CS  TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR
##  Min.   :  7.00   Min.   :29.00    Min.   : 1137   Min.   :  0.0
##  1st Qu.: 44.00   1st Qu.:50.50    1st Qu.: 1419   1st Qu.: 50.0
##  Median : 49.00   Median :58.00    Median : 1518   Median :107.0
##  Mean   : 51.54   Mean   :59.36    Mean   : 1769   Mean   :105.7
##  3rd Qu.: 54.50   3rd Qu.:67.00    3rd Qu.: 1682   3rd Qu.:150.0
##  Max.   :201.00   Max.   :95.00    Max.   :30132   Max.   :343.0
##                   NA's   :2084
##  TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E  TEAM_FIELDING_DP
##  Min.   : 119.0   Min.   :    0.0   Min.   :  65.0   Min.   : 52.0
##  1st Qu.: 476.0   1st Qu.:  626.0   1st Qu.: 127.0   1st Qu.:134.0
##  Median : 537.0   Median :  814.0   Median : 159.0   Median :149.0
##  Mean   : 553.3   Mean   :  817.9   Mean   : 245.8   Mean   :146.7
##  3rd Qu.: 611.0   3rd Qu.:  957.0   3rd Qu.: 249.0   3rd Qu.:161.5
##  Max.   :3645.0   Max.   :19278.0   Max.   :1898.0   Max.   :228.0
##
##  TEAM_BATTING_1B TEAM_BATTING_HBP_YN PITCHER_OUTLIER_YN BATTING_OUTLIER_YN
##  Min.   : 709    Min.   :0.00000     Min.   :0.00000    Min.   :0.00000
##  1st Qu.: 991    1st Qu.:0.00000     1st Qu.:0.00000    1st Qu.:0.00000
##  Median :1050    Median :0.00000     Median :0.00000    Median :0.00000
##  Mean   :1073    Mean   :0.08396     Mean   :0.09363    Mean   :0.03341
##  3rd Qu.:1129    3rd Qu.:0.00000     3rd Qu.:0.00000    3rd Qu.:0.00000
##  Max.   :2112    Max.   :1.00000     Max.   :1.00000    Max.   :1.00000
##
##  BASERUN_OUTLIER_YN FIELDING_OUTLIER_YN TARGET_WINS_Ratio
##  Min.   :0.00000    Min.   :0.0000      Min.   :0.07407
##  1st Qu.:0.00000    1st Qu.:0.0000      1st Qu.:0.43827
##  Median :0.00000    Median :0.0000      Median :0.50617
##  Mean   :0.05978    Mean   :0.1327      Mean   :0.49893
##  3rd Qu.:0.00000    3rd Qu.:0.0000      3rd Qu.:0.56790
##  Max.   :1.00000    Max.   :1.0000      Max.   :0.90123
##
##   TEAM_H_Ratio     TEAM_BASERUN_Ratio TEAM_HR_SO_Ratio
##  Min.   :0.04938   Min.   : 0.000     Min.   :0.00000
##  1st Qu.:0.93223   1st Qu.: 1.448     1st Qu.:0.08543
##  Median :0.95069   Median : 2.059     Median :0.13463
##  Mean   :0.91475   Mean   : 2.466     Mean   :0.13034
##  3rd Qu.:1.00000   3rd Qu.: 2.806     3rd Qu.:0.17306
##  Max.   :1.01854   Max.   :14.224     Max.   :0.54545
##
```

```
##                rowname  TARGET_WINS
## 1    TARGET_WINS_Ratio  1.000000000
## 2      TEAM_BATTING_2B  0.285964582
## 3     TEAM_HR_SO_Ratio  0.276486605
## 4      TEAM_BATTING_BB  0.225471523
## 5      TEAM_BATTING_1B  0.213402040
## 6     TEAM_PITCHING_HR  0.186326615
```

```
## 7      TEAM_BATTING_HR  0.173551988
## 8      TEAM_BATTING_3B  0.139073830
## 9   BATTING_OUTLIER_YN  0.128735272
## 10     TEAM_BASERUN_SB  0.121110012
## 11  TEAM_BASERUN_Ratio  0.119491785
## 12    TEAM_PITCHING_BB  0.117643912
## 13  BASERUN_OUTLIER_YN  0.102242413
## 14        TEAM_H_Ratio  0.083696128
## 15  PITCHER_OUTLIER_YN  0.046631794
## 16     TEAM_BASERUN_CS  0.009835166
## 17 TEAM_BATTING_HBP_YN  0.001939440
## 18    TEAM_FIELDING_DP -0.030050754
## 19     TEAM_BATTING_SO -0.037712096
## 20 FIELDING_OUTLIER_YN -0.053236802
## 21     TEAM_PITCHING_H -0.079145655
## 22    TEAM_PITCHING_SO -0.079711792
## 23     TEAM_FIELDING_E -0.163022098
## 24    TEAM_BATTING_HBP           NA
```

- Let's also revisit our box plots filtered to exclude the outliers. Although some outliers remain, the vast majority of them have been filtered out



Build Model

Model1: Backward elimination

- Let's start with a backward elimination by including all variables (except we'll remove -INDEX, -TARGET_WINS_Ratio, -TEAM_BATTING_HBP on the first pass since these are either irrelevant or captured in other columns) and then remove variables until we are left with only statistically significant variables

```
train_model1 = train_clean
train_model1 = train_model1 %>% select(-INDEX, -TARGET_WINS_Ratio, -TEAM_BATTING_HBP)
model1 = train(TARGET_WINS ~ ., data = train_model1, method = 'lm')
summary(model1)
```

```
##
```

```
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.126  -8.533   0.222   8.115  58.895
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.330e+01  8.493e+00   7.453 1.29e-13 ***
## TEAM_BATTING_2B      2.865e-02  7.671e-03   3.735 0.000193 ***
## TEAM_BATTING_3B      1.280e-01  1.598e-02   8.009 1.84e-15 ***
## TEAM_BATTING_HR      2.359e-01  3.245e-02   7.272 4.87e-13 ***
## TEAM_BATTING_BB      1.989e-02  6.226e-03   3.195 0.001418 **
## TEAM_BATTING_SO     -2.198e-03  3.326e-03  -0.661 0.508734
## TEAM_BASERUN_SB      4.358e-02  2.403e-02   1.814 0.069867 .
## TEAM_BASERUN_CS     -3.547e-02  4.016e-02  -0.883 0.377269
## TEAM_PITCHING_H     -3.362e-04  4.557e-04  -0.738 0.460705
## TEAM_PITCHING_HR    -1.439e-01  3.266e-02  -4.407 1.10e-05 ***
## TEAM_PITCHING_BB    -4.376e-03  4.463e-03  -0.980 0.326960
## TEAM_PITCHING_SO     9.948e-04  9.272e-04   1.073 0.283435
## TEAM_FIELDING_E     -4.989e-02  4.111e-03 -12.135  < 2e-16 ***
## TEAM_FIELDING_DP    -1.271e-01  1.318e-02  -9.638  < 2e-16 ***
## TEAM_BATTING_1B      4.440e-02  4.293e-03  10.343  < 2e-16 ***
## TEAM_BATTING_HBP_YN -4.078e+00  1.209e+00  -3.375 0.000752 ***
## PITCHER_OUTLIER_YN   5.210e+00  1.740e+00   2.995 0.002775 **
## BATTING_OUTLIER_YN   1.400e+00  2.179e+00   0.642 0.520708
## BASERUN_OUTLIER_YN  -1.793e+00  1.882e+00  -0.953 0.340935
## FIELDING_OUTLIER_YN  8.933e+00  1.839e+00   4.857 1.27e-06 ***
## TEAM_H_Ratio        -3.578e+01  7.408e+00  -4.830 1.46e-06 ***
## TEAM_BASERUN_Ratio  -7.192e-01  1.205e+00  -0.597 0.550746
## TEAM_HR_SO_Ratio     1.548e+01  1.289e+01   1.201 0.230031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.76 on 2252 degrees of freedom
## Multiple R-squared:  0.3429, Adjusted R-squared:  0.3365
## F-statistic: 53.42 on 22 and 2252 DF,  p-value: < 2.2e-16
```

```
model1$results
```

```
##   intercept     RMSE  Rsquared   RMSESD RsquaredSD
## 1      TRUE 13.42168 0.2879774 0.512064 0.03956896
```

- We then remove the insignificant columns, -TEAM_HR_SO_Ratio, -TEAM_BASERUN_Ratio, -BASERUN_OUTLIER_YN, -BATTING_OUTLIER_YN, -TEAM_PITCHING_SO, -TEAM_PITCHING_BB, -TEAM_PITCHING_H, -TEAM_BASERUN_CS, -TEAM_BATTING_SO
- Nice, not only do we see that the RMSE decreased slightly but that the RMSESD decreased significantly, which indicates a more precise estimate of the RMSE figure

```
train_model1 = train_model1 %>% select(-TEAM_HR_SO_Ratio, -TEAM_BASERUN_Ratio, -BASERUN_OUTLIER_YN, -BA
                                       -TEAM_PITCHING_SO, -TEAM_PITCHING_BB, -TEAM_PITCHING_H, -TEAM_BAS
                                       -TEAM_BATTING_SO)
model1 = train(TARGET_WINS ~ ., data = train_model1, method = 'lm')
summary(model1)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.778  -8.432   0.132   8.195  57.842
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          53.155060   6.555099   8.109 8.28e-16 ***
## TEAM_BATTING_2B       0.027764   0.007493   3.705 0.000216 ***
## TEAM_BATTING_3B       0.135822   0.015148   8.966  < 2e-16 ***
## TEAM_BATTING_HR       0.251312   0.030586   8.217 3.48e-16 ***
## TEAM_BATTING_BB       0.015211   0.003307   4.600 4.46e-06 ***
## TEAM_BASERUN_SB       0.022143   0.004220   5.248 1.68e-07 ***
## TEAM_PITCHING_HR     -0.144617   0.027408  -5.277 1.44e-07 ***
## TEAM_FIELDING_E      -0.050733   0.003739 -13.570  < 2e-16 ***
## TEAM_FIELDING_DP     -0.126957   0.012828  -9.897  < 2e-16 ***
## TEAM_BATTING_1B       0.048458   0.003122  15.520  < 2e-16 ***
## TEAM_BATTING_HBP_YN  -4.651387   1.131601  -4.110 4.09e-05 ***
## PITCHER_OUTLIER_YN    5.538817   1.651010   3.355 0.000807 ***
## FIELDING_OUTLIER_YN   9.414990   1.680125   5.604 2.35e-08 ***
## TEAM_H_Ratio        -31.644840   5.914721  -5.350 9.67e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.76 on 2261 degrees of freedom
## Multiple R-squared:  0.3404, Adjusted R-squared:  0.3366
## F-statistic: 89.77 on 13 and 2261 DF,  p-value: < 2.2e-16
```

```
model1$results
```

```
##   intercept     RMSE  Rsquared    RMSESD RsquaredSD
## 1      TRUE 12.86653 0.3231218 0.3690171 0.02598453
```

Model2: Ratio-based

- I'm skeptical based on the correlation results, but let's see how the ratio-based model performs; we'll throw the TEAM_PITCHING_HR column as well since no pitching columns were converted to ratios and thus also the PITCHER_OUTLIER_YN column

- Since the response variable was converted into a ratio, the RMSE figures are not directly comparable, but based on the R-squared values, this model clearly underperforms the previous model

```
train_model2 = train_clean
train_model2 = train_model2 %>% select(TARGET_WINS_Ratio, TEAM_H_Ratio, TEAM_BASERUN_Ratio, TEAM_HR_SO_
model2 = train(TARGET_WINS_Ratio ~ ., data = train_model2, method = 'lm')
summary(model2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33772 -0.05691  0.00511  0.05786  0.44478
```

```
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.454e-01  2.143e-02  11.453  < 2e-16 ***
## TEAM_H_Ratio         1.718e-01  2.286e-02   7.515 8.12e-14 ***
## TEAM_BASERUN_Ratio   1.271e-02  1.176e-03  10.806  < 2e-16 ***
## TEAM_HR_SO_Ratio     6.113e-01  4.979e-02  12.277  < 2e-16 ***
## TEAM_PITCHING_HR    -2.186e-04  5.478e-05  -3.991 6.79e-05 ***
## PITCHER_OUTLIER_YN   9.032e-02  1.114e-02   8.108 8.36e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08913 on 2269 degrees of freedom
## Multiple R-squared:  0.1521, Adjusted R-squared:  0.1502
## F-statistic: 81.41 on 5 and 2269 DF,  p-value: < 2.2e-16
```

```
model2$results
```

```
##   intercept       RMSE  Rsquared      RMSESD RsquaredSD
## 1      TRUE 0.09026956 0.1406657 0.001956328  0.0262628
```

Model3: Remove highly collinear variables

- Time to return to the original model, but we'll remove the most collinear variables based on the cross correlation and variance inflation factors shown below:
  - TEAM_BATTING_HR: 0.969 correlation with TEAM_PITCHING_HR
  - FIELDING_OUTLIER_YN & TEAM_H_Ratio: both are highly correlated to TEAM_FIELDING_E and to each other
- Although the RMSE increases slightly, the model is simpler and more stable without the collinear variables, so we will use Model3 as the final model
- Coefficient discussion - in this final model, below is a brief discussion of the coefficients for the selected variables:
  - TEAM_FIELDING_DP: The negative coefficient suggests that more double plays is associated with fewer wins. While this differs from the suggested impact as per the homework assignment, it could be possible as teams that have more double play opportunities likely have more opposing runners on base at a given time which should have a negative effect. Further research is needed to isolate the impact of double plays.
  - TEAM_BATTING_HBP_YN & PITCHER_OUTLIER_YN: Both are flags created to deal with either missing data or outliers; there coefficients may not have intuitive interpretations
  - Remaining variables: The remaining variables have signs that are consistent with their expected impact on the number of wins.

```
cor(train_model1)
```

```
##                    TARGET_WINS TEAM_BATTING_2B TEAM_BATTING_3B
## TARGET_WINS          1.00000000      0.28596458       0.1390738
## TEAM_BATTING_2B      0.28596458      1.00000000      -0.1094983
## TEAM_BATTING_3B      0.13907383     -0.10949828       1.0000000
## TEAM_BATTING_HR      0.17355199      0.43450589      -0.6379259
## TEAM_BATTING_BB      0.22547152      0.25286114      -0.2921176
## TEAM_BASERUN_SB      0.12111001     -0.18514149       0.4851236
## TEAM_PITCHING_HR     0.18632661      0.45364008      -0.5701993
```

```
## TEAM_FIELDING_E     -0.16302210     -0.23086449       0.5225142
## TEAM_FIELDING_DP    -0.03005075      0.25735317      -0.2278874
## TEAM_BATTING_1B      0.21340204      0.08476283       0.5995733
## TEAM_BATTING_HBP_YN  0.00193944      0.36203847      -0.2660418
## PITCHER_OUTLIER_YN   0.04663179      0.02967777       0.3225772
## FIELDING_OUTLIER_YN -0.05323680     -0.16160694       0.4759790
## TEAM_H_Ratio         0.08369613      0.11531561      -0.4122639
##                    TEAM_BATTING_HR TEAM_BATTING_BB TEAM_BASERUN_SB
## TARGET_WINS              0.1735520      0.22547152      0.12111001
## TEAM_BATTING_2B          0.4345059      0.25286114     -0.18514149
## TEAM_BATTING_3B         -0.6379259     -0.29211757      0.48512355
## TEAM_BATTING_HR          1.0000000      0.51296999     -0.40836484
## TEAM_BATTING_BB          0.5129700      1.00000000     -0.04547002
## TEAM_BASERUN_SB         -0.4083648     -0.04547002      1.00000000
## TEAM_PITCHING_HR         0.9693345      0.45843864     -0.38157640
## TEAM_FIELDING_E         -0.5892503     -0.65289124      0.33474354
## TEAM_FIELDING_DP         0.3919539      0.33102671     -0.27029947
## TEAM_BATTING_1B         -0.5000378     -0.35934482      0.27225401
## TEAM_BATTING_HBP_YN      0.3922217      0.10289528     -0.11540717
## PITCHER_OUTLIER_YN      -0.2831195     -0.48896379      0.13759716
## FIELDING_OUTLIER_YN     -0.4389749     -0.47985250      0.42852241
## TEAM_H_Ratio             0.4794442      0.62376581     -0.18395997
##                    TEAM_PITCHING_HR TEAM_FIELDING_E TEAM_FIELDING_DP
## TARGET_WINS              0.1863266      -0.1630221      -0.03005075
## TEAM_BATTING_2B          0.4536401      -0.2308645       0.25735317
## TEAM_BATTING_3B         -0.5701993       0.5225142      -0.22788742
## TEAM_BATTING_HR          0.9693345      -0.5892503       0.39195386
## TEAM_BATTING_BB          0.4584386      -0.6528912       0.33102671
## TEAM_BASERUN_SB         -0.3815764       0.3347435      -0.27029947
## TEAM_PITCHING_HR         1.0000000      -0.4936765       0.38992182
## TEAM_FIELDING_E         -0.4936765       1.0000000      -0.23034283
## TEAM_FIELDING_DP         0.3899218      -0.2303428       1.00000000
## TEAM_BATTING_1B         -0.4181906       0.5628440      -0.08789672
## TEAM_BATTING_HBP_YN      0.3579617      -0.1865065       0.06933660
## PITCHER_OUTLIER_YN      -0.1455180       0.6774971      -0.12675890
## FIELDING_OUTLIER_YN     -0.3523016       0.8472196      -0.10923311
## TEAM_H_Ratio             0.3222699      -0.8694651       0.18631621
##                    TEAM_BATTING_1B TEAM_BATTING_HBP_YN PITCHER_OUTLIER_YN
## TARGET_WINS            0.21340204          0.00193944        0.04663179
## TEAM_BATTING_2B        0.08476283          0.36203847        0.02967777
## TEAM_BATTING_3B        0.59957334         -0.26604179        0.32257722
## TEAM_BATTING_HR       -0.50003777          0.39222167       -0.28311953
## TEAM_BATTING_BB       -0.35934482          0.10289528       -0.48896379
## TEAM_BASERUN_SB        0.27225401         -0.11540717        0.13759716
## TEAM_PITCHING_HR      -0.41819055          0.35796169       -0.14551795
## TEAM_FIELDING_E        0.56284404         -0.18650650        0.67749711
## TEAM_FIELDING_DP      -0.08789672          0.06933660       -0.12675890
## TEAM_BATTING_1B        1.00000000         -0.23669901        0.47124119
## TEAM_BATTING_HBP_YN   -0.23669901          1.00000000       -0.09730010
## PITCHER_OUTLIER_YN     0.47124119         -0.09730010        1.00000000
## FIELDING_OUTLIER_YN    0.43438136         -0.11844255        0.62582226
## TEAM_H_Ratio          -0.52640742          0.17224454       -0.80830086
##                    FIELDING_OUTLIER_YN TEAM_H_Ratio
## TARGET_WINS                 -0.0532368   0.08369613
```

```
## TEAM_BATTING_2B              -0.1616069   0.11531561
## TEAM_BATTING_3B               0.4759790  -0.41226395
## TEAM_BATTING_HR              -0.4389749   0.47944418
## TEAM_BATTING_BB              -0.4798525   0.62376581
## TEAM_BASERUN_SB               0.4285224  -0.18395997
## TEAM_PITCHING_HR             -0.3523016   0.32226994
## TEAM_FIELDING_E               0.8472196  -0.86946508
## TEAM_FIELDING_DP             -0.1092331   0.18631621
## TEAM_BATTING_1B               0.4343814  -0.52640742
## TEAM_BATTING_HBP_YN          -0.1184426   0.17224454
## PITCHER_OUTLIER_YN            0.6258223  -0.80830086
## FIELDING_OUTLIER_YN           1.0000000  -0.73987516
## TEAM_H_Ratio                 -0.7398752   1.00000000
```

```
vif(model1$finalModel)
```

```
##     TEAM_BATTING_2B     TEAM_BATTING_3B     TEAM_BATTING_HR
##            1.715161            2.499062           47.876589
##     TEAM_BATTING_BB     TEAM_BASERUN_SB    TEAM_PITCHING_HR
##            2.282792            1.813637           39.399739
##     TEAM_FIELDING_E    TEAM_FIELDING_DP     TEAM_BATTING_1B
##            9.904034            1.384859            2.258869
## TEAM_BATTING_HBP_YN  PITCHER_OUTLIER_YN FIELDING_OUTLIER_YN
##            1.376493            3.233135            4.542274
##        TEAM_H_Ratio
##           10.788789
```

```
train_model3 = train_model1 %>% select(-TEAM_BATTING_HR, -FIELDING_OUTLIER_YN, -TEAM_H_Ratio)
model3 = train(TARGET_WINS ~ ., data = train_model3, method = 'lm')
summary(model3)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.505  -8.475   0.093   8.349  57.846
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         24.232637   3.516409   6.891 7.14e-12 ***
## TEAM_BATTING_2B      0.033594   0.007636   4.400 1.13e-05 ***
## TEAM_BATTING_3B      0.106657   0.014722   7.244 5.92e-13 ***
## TEAM_BATTING_BB      0.015579   0.003362   4.634 3.80e-06 ***
## TEAM_BASERUN_SB      0.024497   0.003988   6.143 9.52e-10 ***
## TEAM_PITCHING_HR     0.077009   0.007130  10.800  < 2e-16 ***
## TEAM_FIELDING_E     -0.028218   0.002283 -12.361  < 2e-16 ***
## TEAM_FIELDING_DP    -0.111020   0.012815  -8.663  < 2e-16 ***
## TEAM_BATTING_1B      0.043179   0.003135  13.774  < 2e-16 ***
## TEAM_BATTING_HBP_YN -3.046056   1.138163  -2.676   0.0075 **
## PITCHER_OUTLIER_YN   7.929821   1.406515   5.638 1.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 13.05 on 2264 degrees of freedom
## Multiple R-squared:  0.3087, Adjusted R-squared:  0.3057
## F-statistic: 101.1 on 10 and 2264 DF,  p-value: < 2.2e-16
```

```
model3$results
```

```
##   intercept    RMSE  Rsquared    RMSESD RsquaredSD
## 1      TRUE 13.1298 0.3022075 0.2923675 0.02466683
```

```
vif(model3$finalModel)
```

```
##     TEAM_BATTING_2B     TEAM_BATTING_3B     TEAM_BATTING_BB
##            1.701453            2.255375            2.254677
##     TEAM_BASERUN_SB    TEAM_PITCHING_HR     TEAM_FIELDING_E
##            1.547392            2.547793            3.527537
##    TEAM_FIELDING_DP     TEAM_BATTING_1B TEAM_BATTING_HBP_YN
##            1.320439            2.175392            1.330373
##  PITCHER_OUTLIER_YN
##            2.241767
```

SELECT MODELS

Model selection rationale

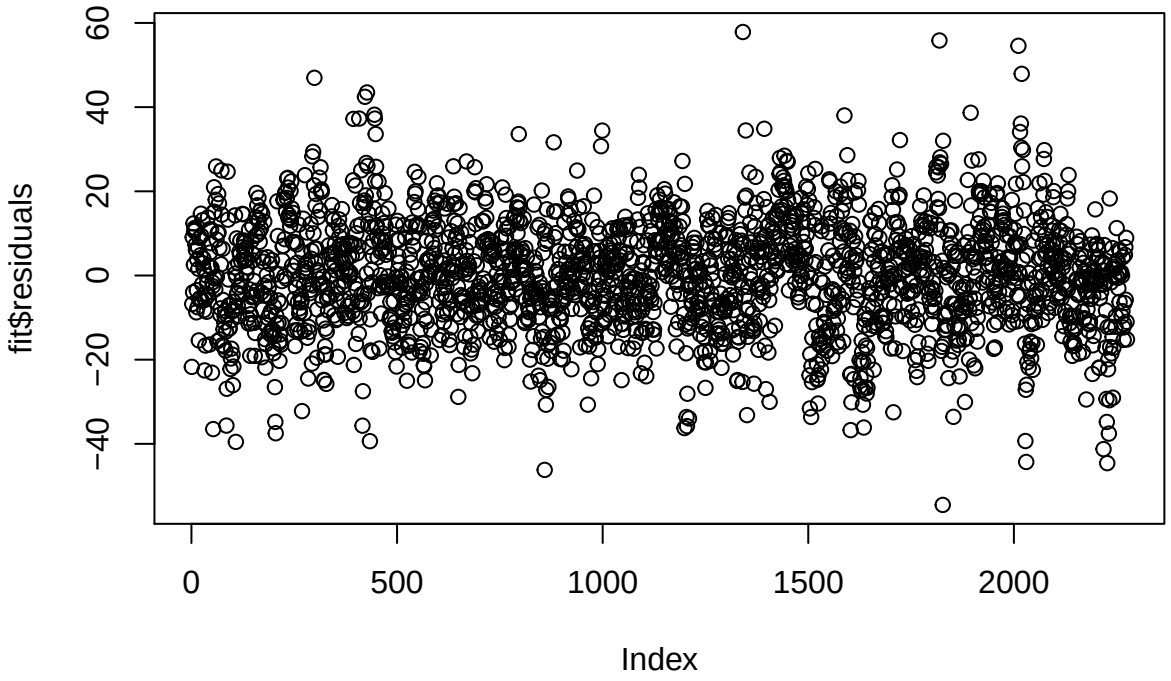
As discussed above, we selected Model3, which was based on backward elimination, followed by removal of any highly collinear variables. Although Model3 did not have the lowest RMSE, it was the most parsimonious (fewest variables) and stable (little collinearity between variables).

Inference and regression diagnostics

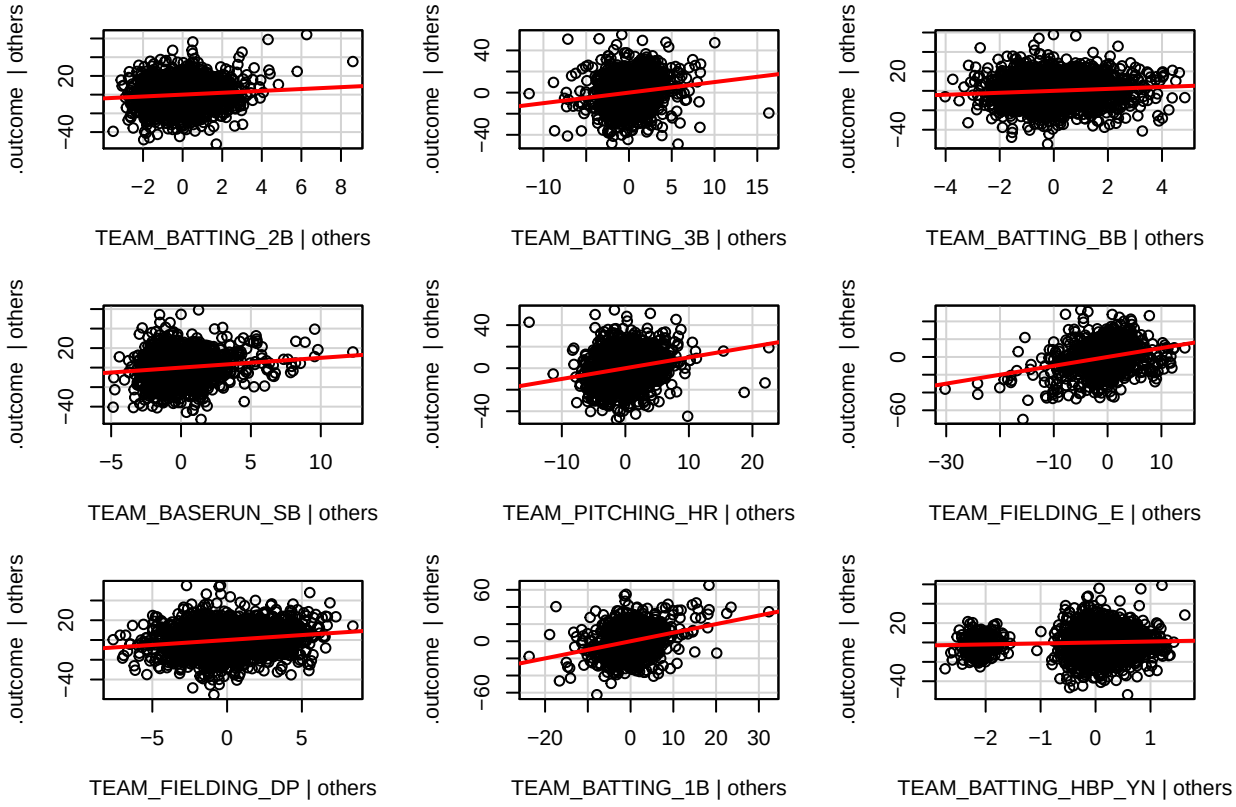
For our inferences to be valid, we need to perform some regression diagnostics and validate some assumptions:

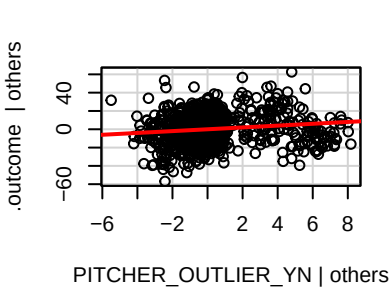
- Independence of errors: Based on the residual plot below, the residuals appear random over the index values
- Outliers and leverage: Based on the leverage plots below, there do not appear to be any data points exerting undue leverage on the regression
- Normality: Based on the qq-plot below, the residuals are fairly normally distributed, although there are some outliers in the tails
- Constant variance: Based on the spread-level plot below, variance appears relatively constant, although again with a few outliers

```
fit = model3$finalModel
plot(fit$residuals)
```

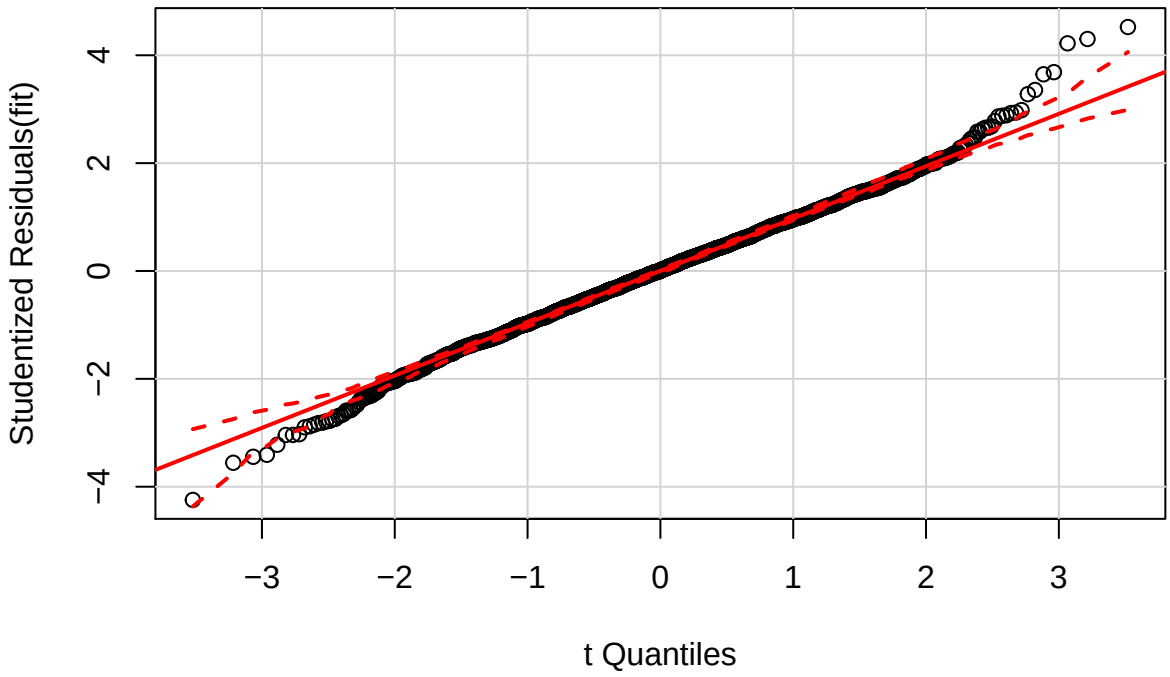



```
leveragePlots(fit)
```

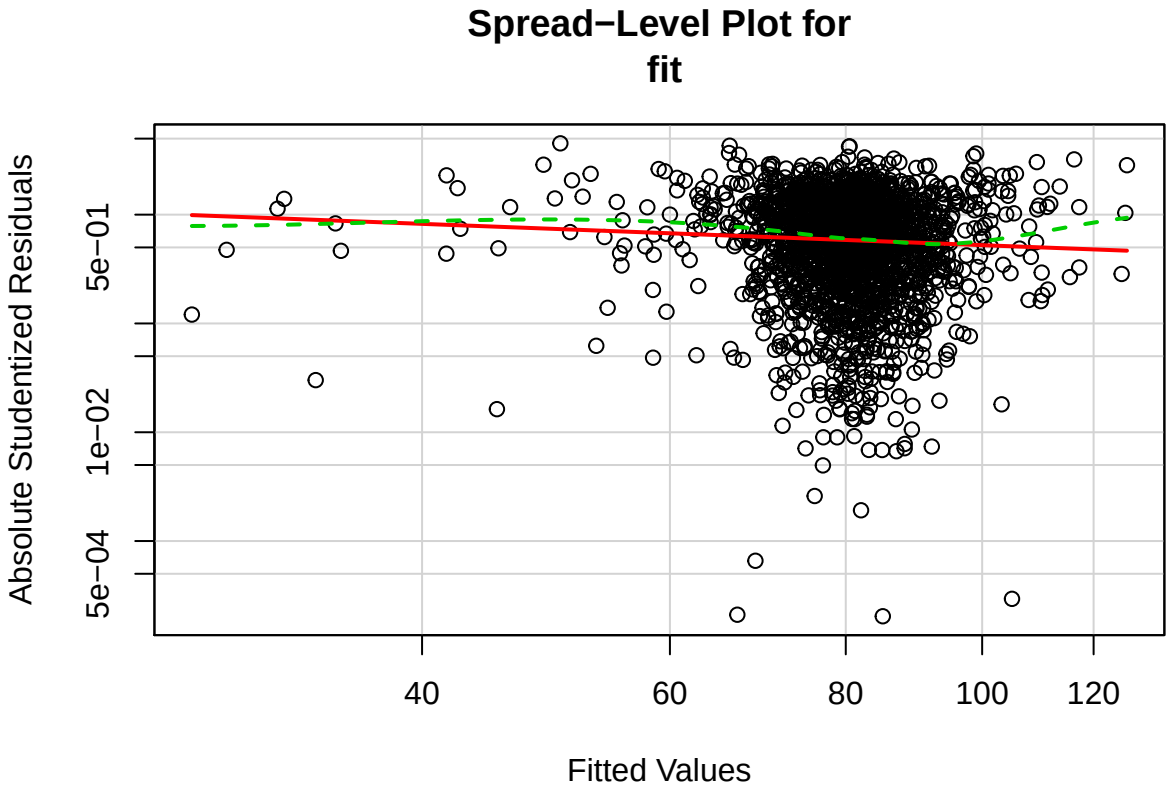





```
qqPlot(fit)
```




```
spreadLevelPlot(fit)
```




```
##
## Suggested power transformation:  1.492206
```

Evaluation of regression model

Based on the above regression diagnostics results, we can comfortably make inferences from our model results:

- RMSE: 13.05
- Adjusted R2: 0.3057
- F-statistic: 101.1
- residual plots: see regression diagnostics above

```
summary(model3)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.505  -8.475   0.093   8.349  57.846
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        24.232637   3.516409   6.891 7.14e-12 ***
## TEAM_BATTING_2B     0.033594   0.007636   4.400 1.13e-05 ***
## TEAM_BATTING_3B     0.106657   0.014722   7.244 5.92e-13 ***
## TEAM_BATTING_BB     0.015579   0.003362   4.634 3.80e-06 ***
```

```
## TEAM_BASERUN_SB      0.024497   0.003988   6.143 9.52e-10 ***
## TEAM_PITCHING_HR     0.077009   0.007130  10.800  < 2e-16 ***
## TEAM_FIELDING_E     -0.028218   0.002283 -12.361  < 2e-16 ***
## TEAM_FIELDING_DP    -0.111020   0.012815  -8.663  < 2e-16 ***
## TEAM_BATTING_1B      0.043179   0.003135  13.774  < 2e-16 ***
## TEAM_BATTING_HBP_YN -3.046056   1.138163  -2.676   0.0075 **
## PITCHER_OUTLIER_YN   7.929821   1.406515   5.638 1.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.05 on 2264 degrees of freedom
## Multiple R-squared:  0.3087, Adjusted R-squared:  0.3057
## F-statistic: 101.1 on 10 and 2264 DF,  p-value: < 2.2e-16
```

Predictions on evaluation set

```
# prep evaluation data for prediction
test_raw = read.csv("~/Google Drive/CUNY/git/DATA621/HW1/moneyball-evaluation-data.csv")
test_clean = test_raw
# create TEAM_BATTING_1B column
test_clean = test_clean %>% mutate(TEAM_BATTING_1B = TEAM_BATTING_H - TEAM_BATTING_2B - TEAM_BATTING_3B
  select(-TEAM_BATTING_H)
# fill in missing values
test_clean = test_clean %>% mutate(
  TEAM_PITCHING_SO = ifelse(is.na(TEAM_PITCHING_SO), median(TEAM_PITCHING_SO, na.rm = TRUE), TEAM_PITCH
  TEAM_BATTING_SO = ifelse(is.na(TEAM_BATTING_SO), median(TEAM_BATTING_SO, na.rm = TRUE), TEAM_BATTING_S
  TEAM_BASERUN_SB = ifelse(is.na(TEAM_BASERUN_SB), median(TEAM_BASERUN_SB, na.rm = TRUE), TEAM_BASERUN_S
  TEAM_BASERUN_CS = ifelse(is.na(TEAM_BASERUN_CS), median(TEAM_BASERUN_CS, na.rm = TRUE), TEAM_BASERUN_C
  TEAM_FIELDING_DP = ifelse(is.na(TEAM_FIELDING_DP), median(TEAM_FIELDING_DP, na.rm = TRUE), TEAM_FIELDI
# add TEAM_BATTING_HBP_YN
test_clean = test_clean %>% mutate(TEAM_BATTING_HBP_YN = ifelse(is.na(TEAM_BATTING_HBP), 0, 1))
# add outlier flags
test_clean = test_clean %>%
  mutate('PITCHER_OUTLIER_YN' = ifelse(TEAM_PITCHING_H > quantile(TEAM_PITCHING_H, 0.75) + 1.5 * IQR(TEA
                                       TEAM_PITCHING_H < quantile(TEAM_PITCHING_H, 0.25) - 1.5 * IQR(TEA
                                       1, 0))
test_clean = test_clean %>%
  mutate('BATTING_OUTLIER_YN' = ifelse(TEAM_BATTING_1B > quantile(TEAM_BATTING_1B, 0.75) + 1.5 * IQR(TEA
                                       TEAM_BATTING_1B < quantile(TEAM_BATTING_1B, 0.25) - 1.5 * IQR(TEA
                                       1, 0))
test_clean = test_clean %>%
  mutate('BATTING_OUTLIER_YN' = ifelse(TEAM_BATTING_1B > quantile(TEAM_BATTING_1B, 0.75) + 1.5 * IQR(TEA
                                       TEAM_BATTING_1B < quantile(TEAM_BATTING_1B, 0.25) - 1.5 * IQR(TEA
                                       1, 0))
test_clean = test_clean %>%
  mutate('BASERUN_OUTLIER_YN' = ifelse(TEAM_BASERUN_SB > quantile(TEAM_BASERUN_SB, 0.75) + 1.5 * IQR(TEA
                                       TEAM_BASERUN_SB < quantile(TEAM_BASERUN_SB, 0.25) - 1.5 * IQR(TEA
                                       1, 0))
test_clean = test_clean %>%
  mutate('FIELDING_OUTLIER_YN' = ifelse(TEAM_FIELDING_E > quantile(TEAM_FIELDING_E, 0.75) + 1.5 * IQR(TE
                                       TEAM_FIELDING_E < quantile(TEAM_FIELDING_E, 0.25) - 1.5 * IQR(TEA
                                       1, 0))

# make predictions and save as csv file
pred = predict(model3, newdata = test_clean)
```

```
write.csv(pred, "~/Google Drive/CUNY/git/DATA621/HW1/moneyball-evaluation-pred.csv")
```