DATA 621: Homework 1 (Baseball Regression)

Shoshanna Farber, Josh Forster, Glen Davis, Andrew Bowen, Charles Ugiagbe

2023-09-04

Setup:

```
## Warning: package 'car' was built under R version 4.3.1
```

```
## Warning: package 'carData' was built under R version 4.3.1
```

First, let's read in the provided dataset.

Data Exploration:

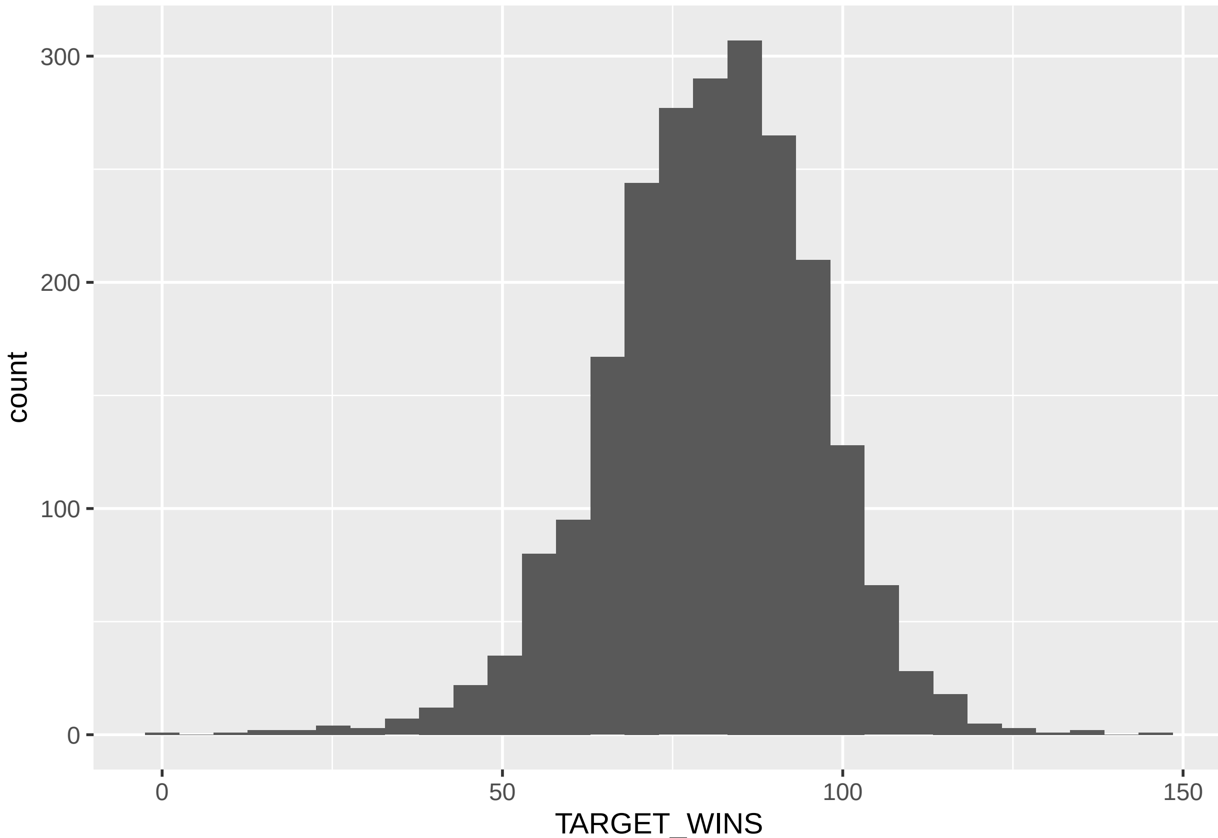
Next, let's print out some summary statistics. We're primarily interested in the `TARGET_WINS` variable, so we'll look at that first.

```
## The mean number of wins in a season is 80.7908611599297
```

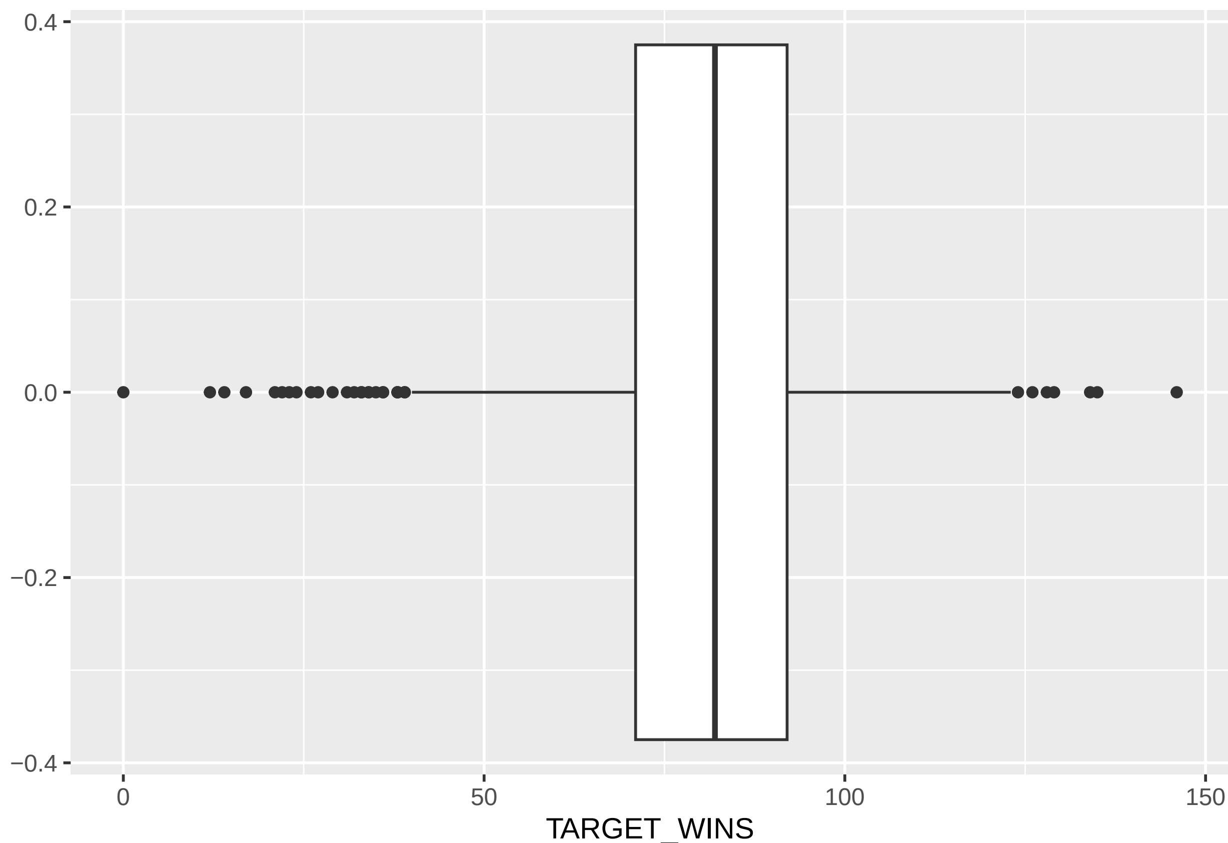
```
## The median number of wins in a season is 82
```

```
## The standard deviation for number of wins in a season is 15.7521524768421
```

Let's also make a boxplot and histogram of the `TARGET_WINS` variable. This should give us a sense of the distribution of wins for teams/seasons in our population.



Overall, the number of wins in a season for a given baseball team looks fairly normally distributed. We can also plot this distribution via a boxplot, which helps to highlight outliers.



Let's look at raw correlations between our other included variables and a team's win total for a season:

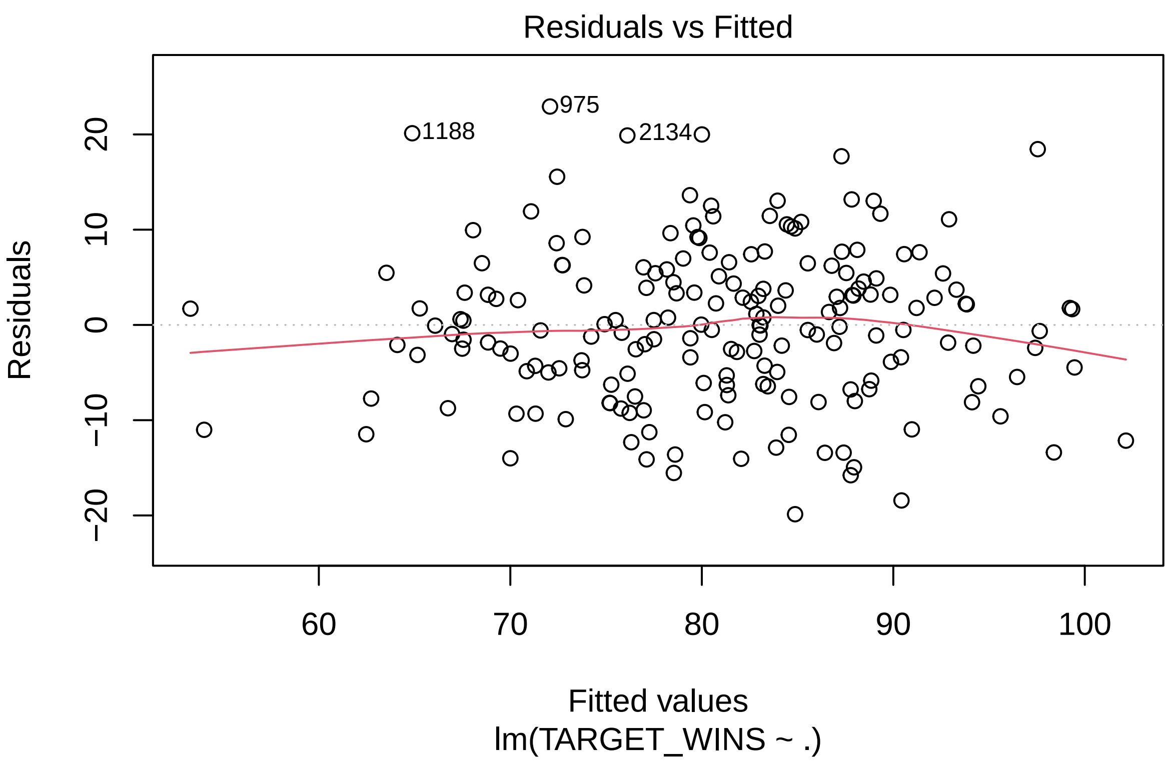
```
##                          [,1]
## TARGET_WINS       1.0000000
## TEAM_BATTING_H    0.3887675
## TEAM_BATTING_2B   0.2891036
## TEAM_BATTING_3B   0.1426084
## TEAM_BATTING_HR   0.1761532
## TEAM_BATTING_BB   0.2325599
## TEAM_BATTING_SO          NA
## TEAM_BASERUN_SB          NA
## TEAM_BASERUN_CS          NA
## TEAM_BATTING_HBP         NA
## TEAM_PITCHING_H  -0.1099371
## TEAM_PITCHING_HR  0.1890137
## TEAM_PITCHING_BB  0.1241745
## TEAM_PITCHING_SO         NA
## TEAM_FIELDING_E  -0.1764848
## TEAM_FIELDING_DP         NA
```

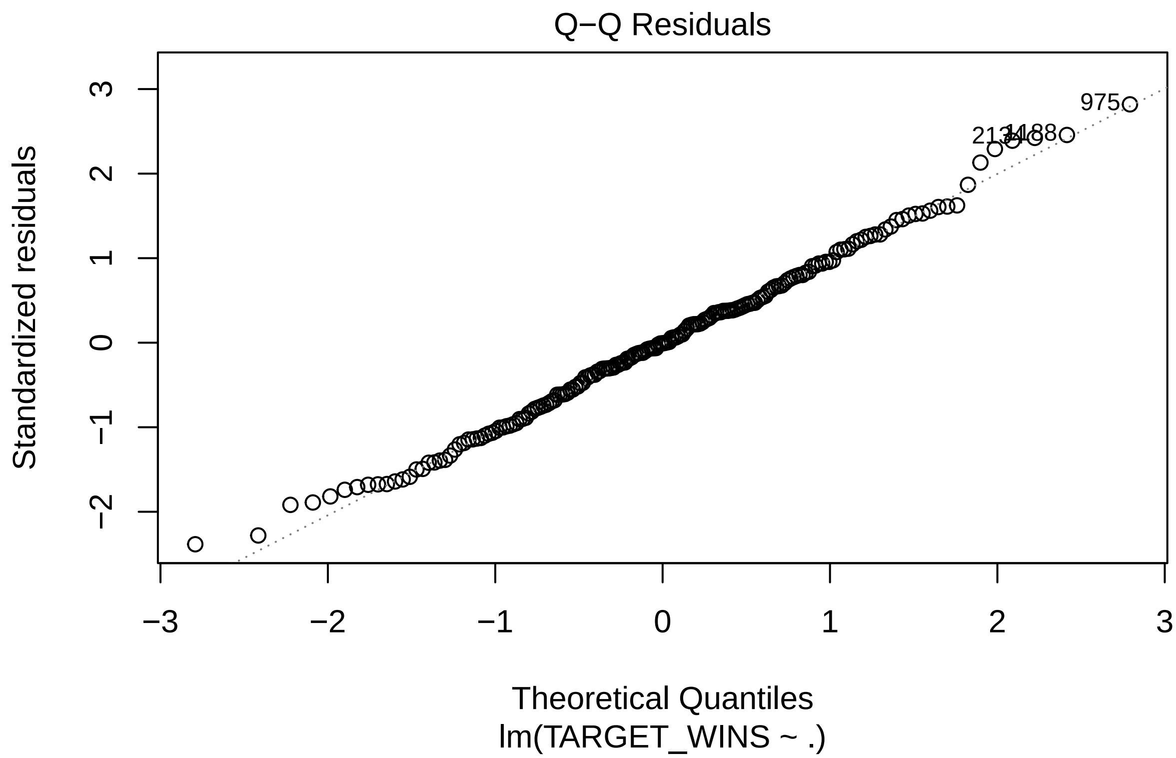
Let's make a basic model with all inputs:

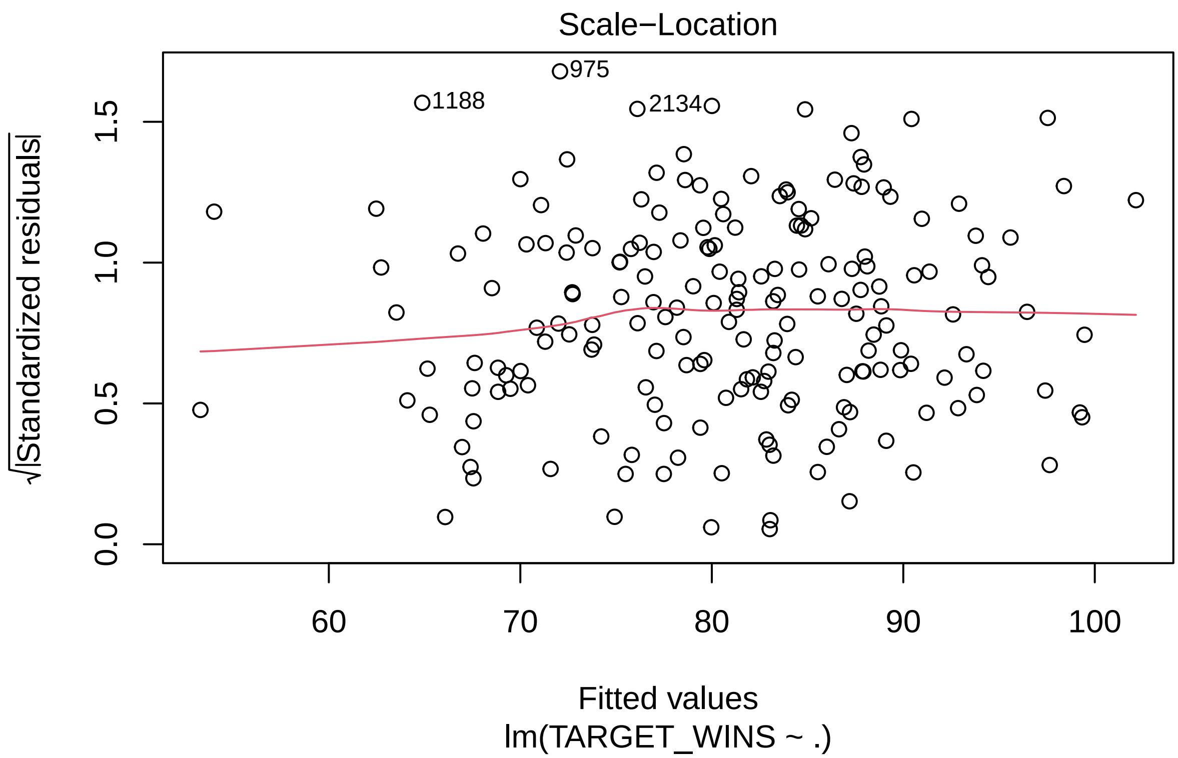
```
##      (Intercept)   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##      60.28826257       1.91347621       0.02638808      -0.10117554
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##      -4.84370721      -4.45969136       0.34196258       0.03304398
```

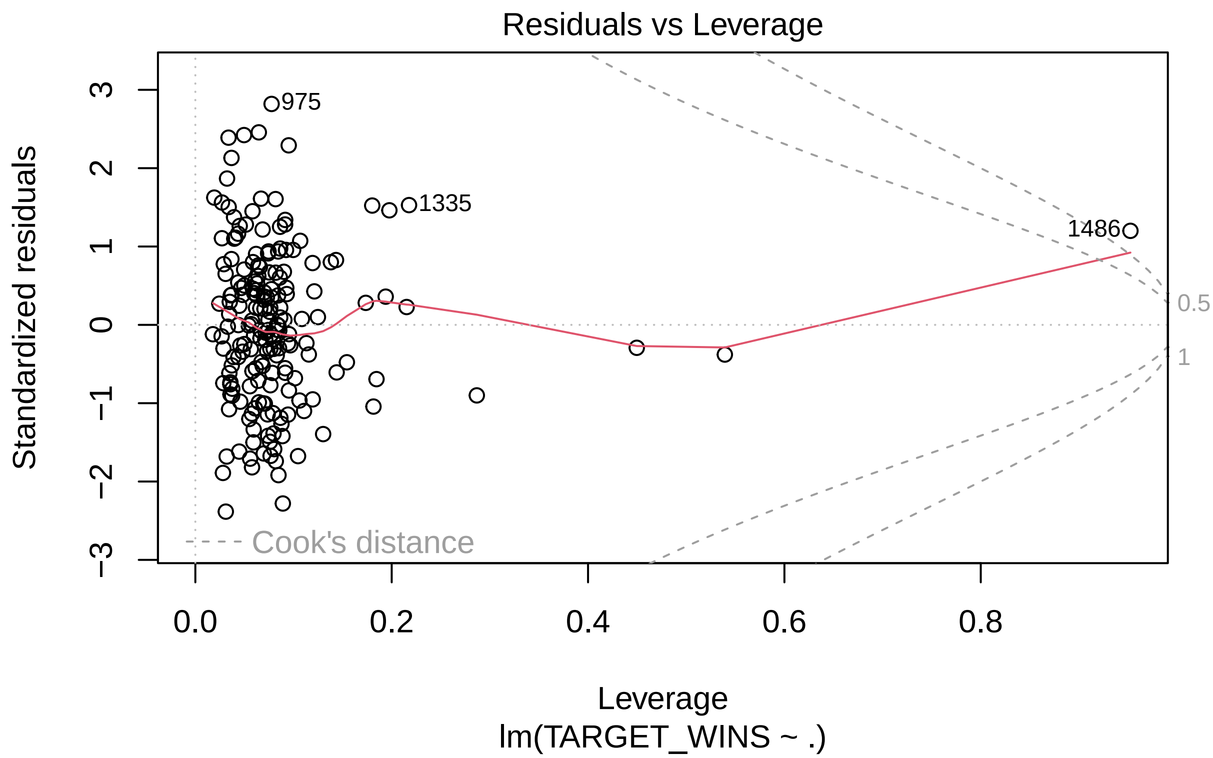
```
##  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H TEAM_PITCHING_HR
##      -0.01104427       0.08247269      -1.89095685       4.93043182
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##       4.51089069      -0.37364495      -0.17204198      -0.10819208
```

We can make some plots to help test our assumptions of our basic model using the `plot` function on our model variable:




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ .)
975
−3
−2
−1
0
1
2
3


Scale−Location
975
1188
2134
1.5
1.0
0.5
0.0
√|Standardized residuals|
60
70
80
90
100
Fitted values
lm(TARGET_WINS ~ .)



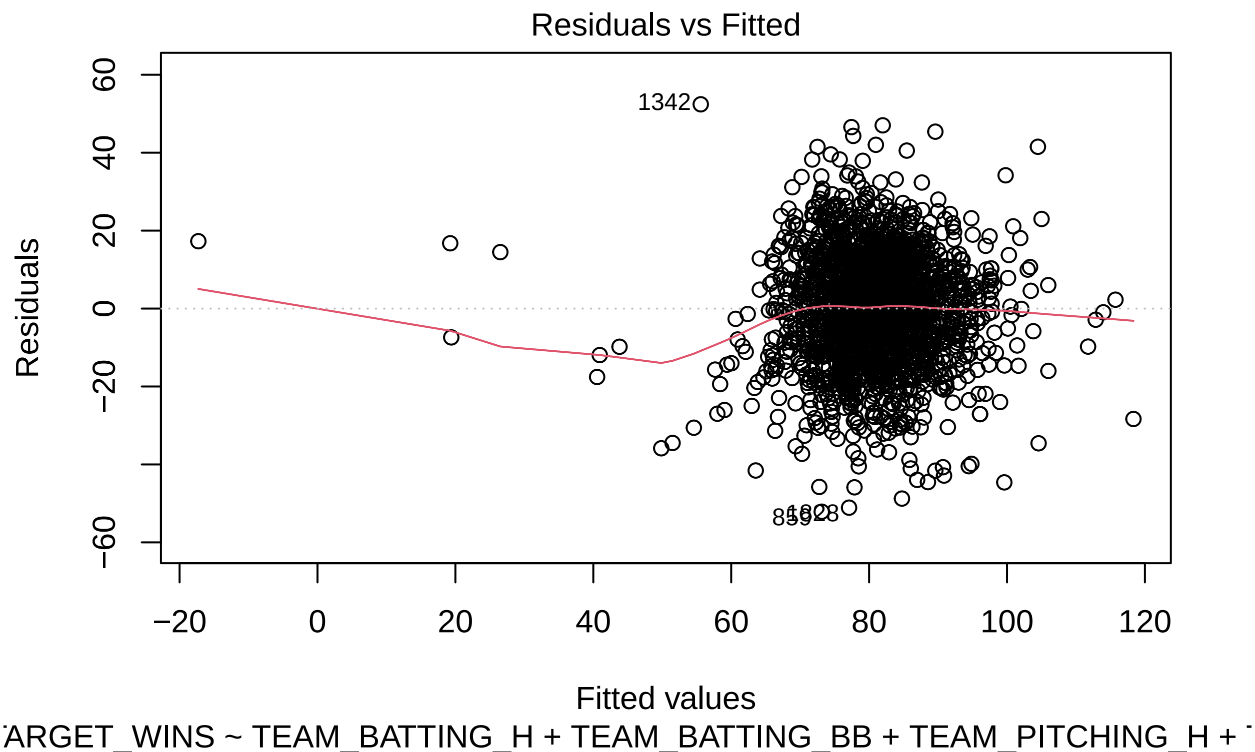
Now we can make a model with inputs that we know from baseball.

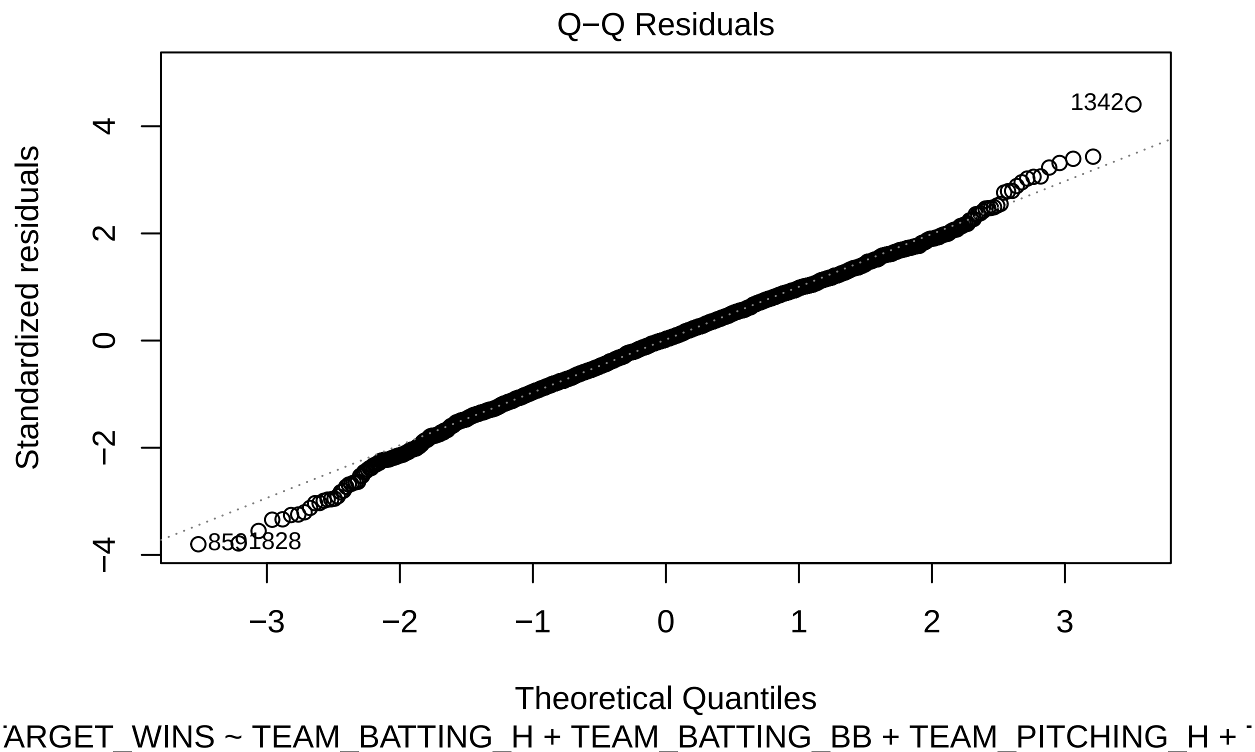
- Total hits (`TEAM_BATTING_H`)
- Total walks gained (`TEAM_BATTING_BB`)
- Total hits allowed (`TEAM_PITCHING_H`)
- Total walks allowed (`TEAM_PITCHING_BB`)

We chose these variables based on our understanding that good teams generally tend to get on base more frequently (positive predictor variables `TEAM_BATTING_HITS` and `TEAM_BATTING_BB`) while allowing *fewer* runners on base (negative predictor variables `TEAM_PITCHING_H` and `TEAM_PITCHING_BB`).

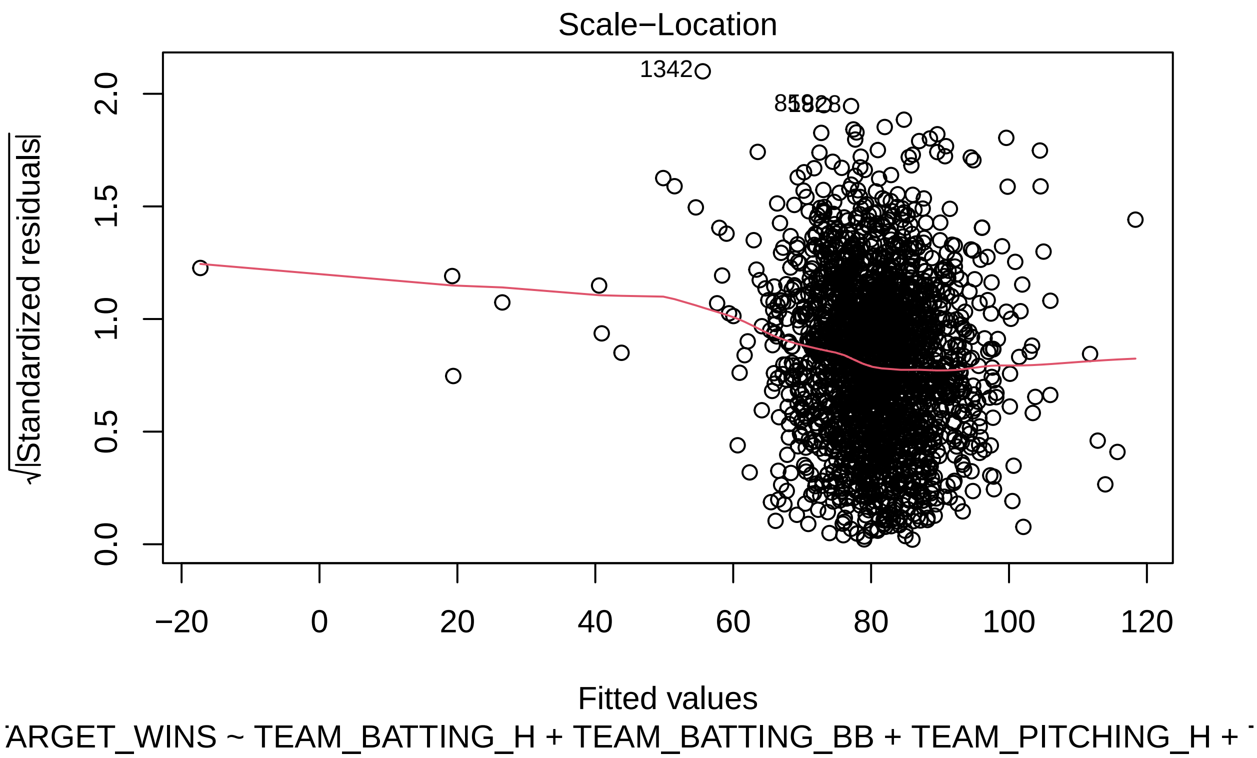
```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB +
##     TEAM_PITCHING_H + TEAM_PITCHING_BB, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.133  -8.860   0.379   9.373  52.416
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.3518000  3.2552864  -0.108 0.913949
## TEAM_BATTING_H    0.0497667  0.0021032  23.663  < 2e-16 ***
## TEAM_BATTING_BB   0.0148499  0.0039923   3.720 0.000204 ***
## TEAM_PITCHING_H  -0.0025469  0.0003317  -7.679 2.36e-14 ***
```

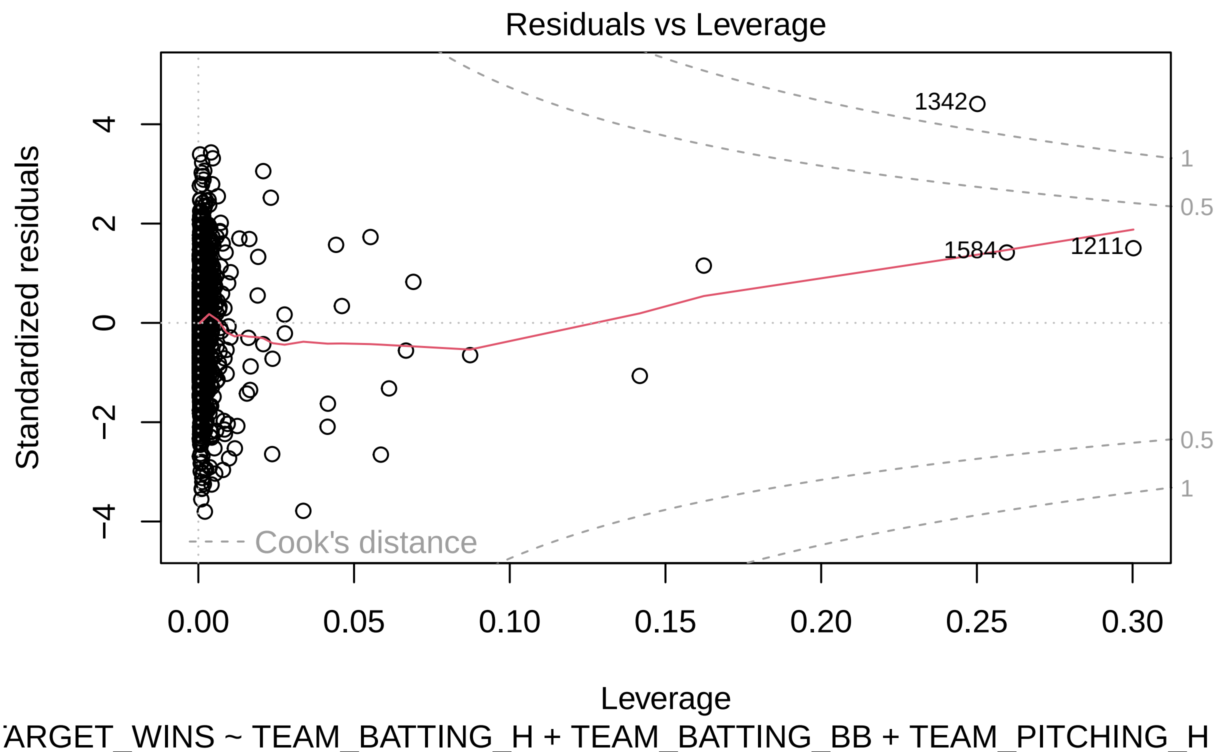
```
## TEAM_PITCHING_BB  0.0092317  0.0027681   3.335 0.000867 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.73 on 2271 degrees of freedom
## Multiple R-squared:  0.2416, Adjusted R-squared:  0.2403
## F-statistic: 180.9 on 4 and 2271 DF,  p-value: < 2.2e-16
```





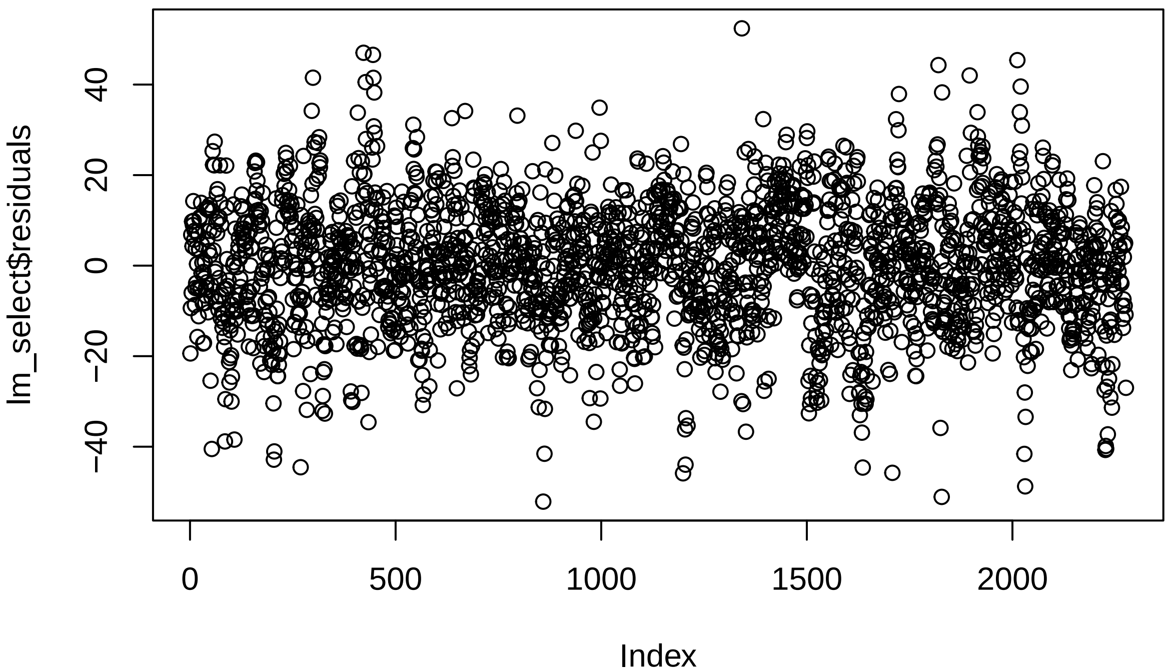
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
1342
859
1828
−3
−2
−1
0
1
2
3
4
2
0
−2
−4
ARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_PITCHING_H +


Scale−Location
1342
859
1828
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
−20
0
20
40
60
80
100
120
Fitted values
ARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_PITCHING_H +

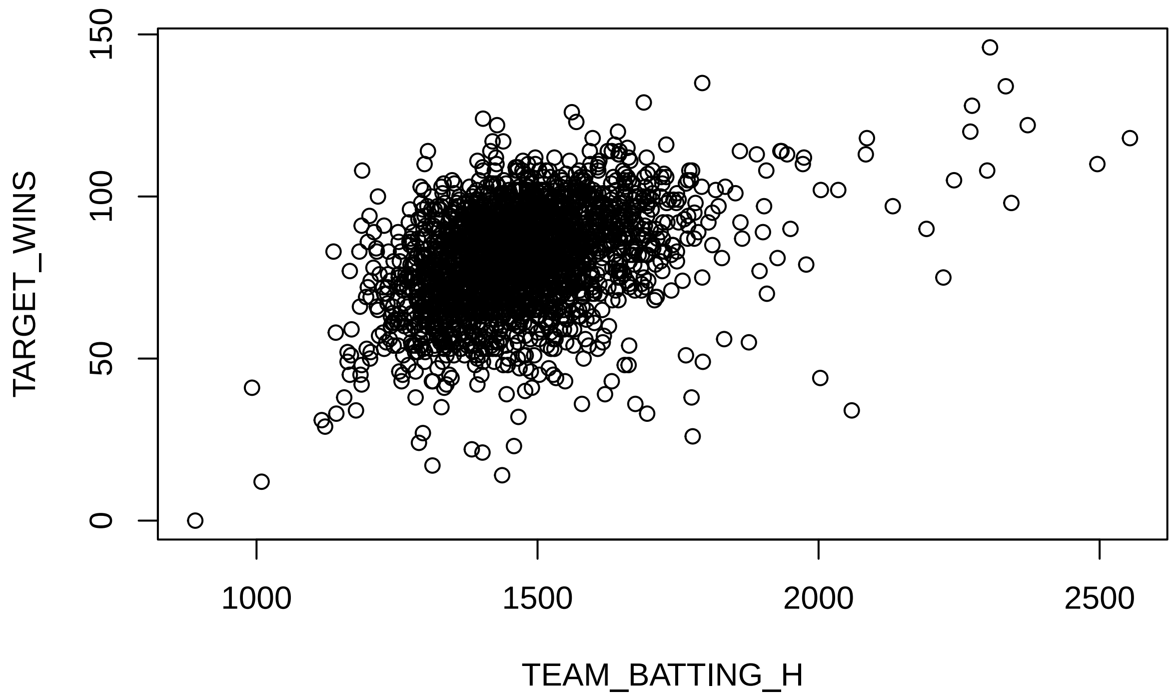


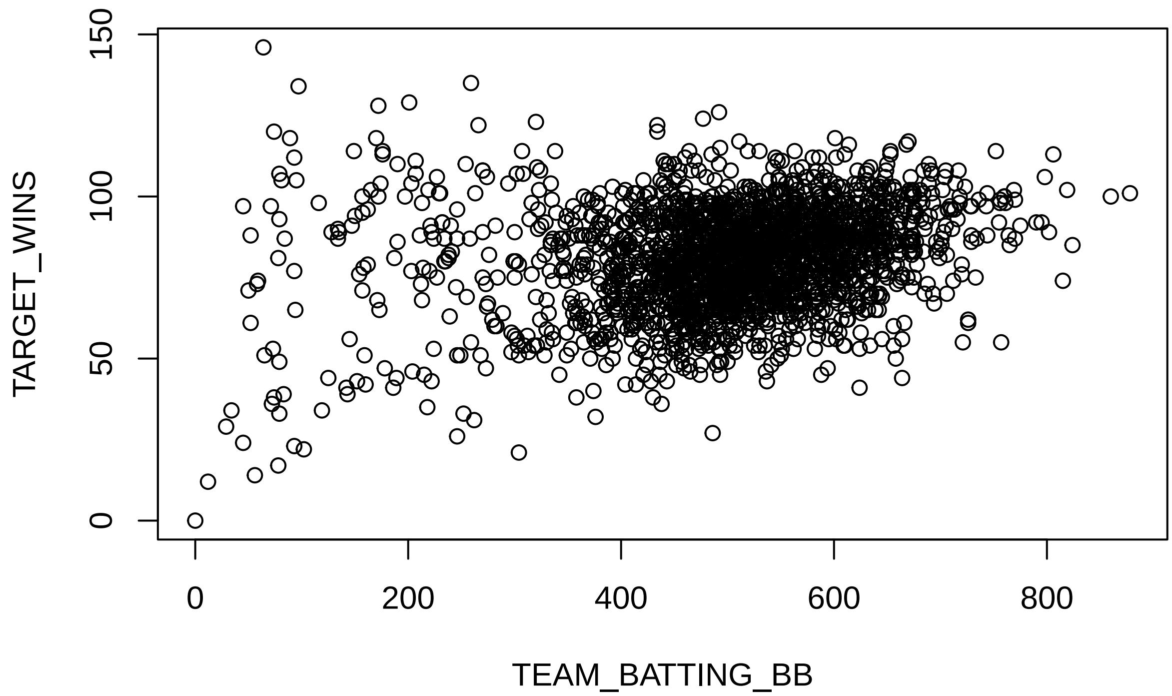
It's interesting to note that with selected variables (walks and hits gained/allowed per team) that our adjusted R^2 actually went *down*, indicating the amount of variability in `TARGET_WINS` explained by our more selective walks/hits model is *less* than the model including all variables.

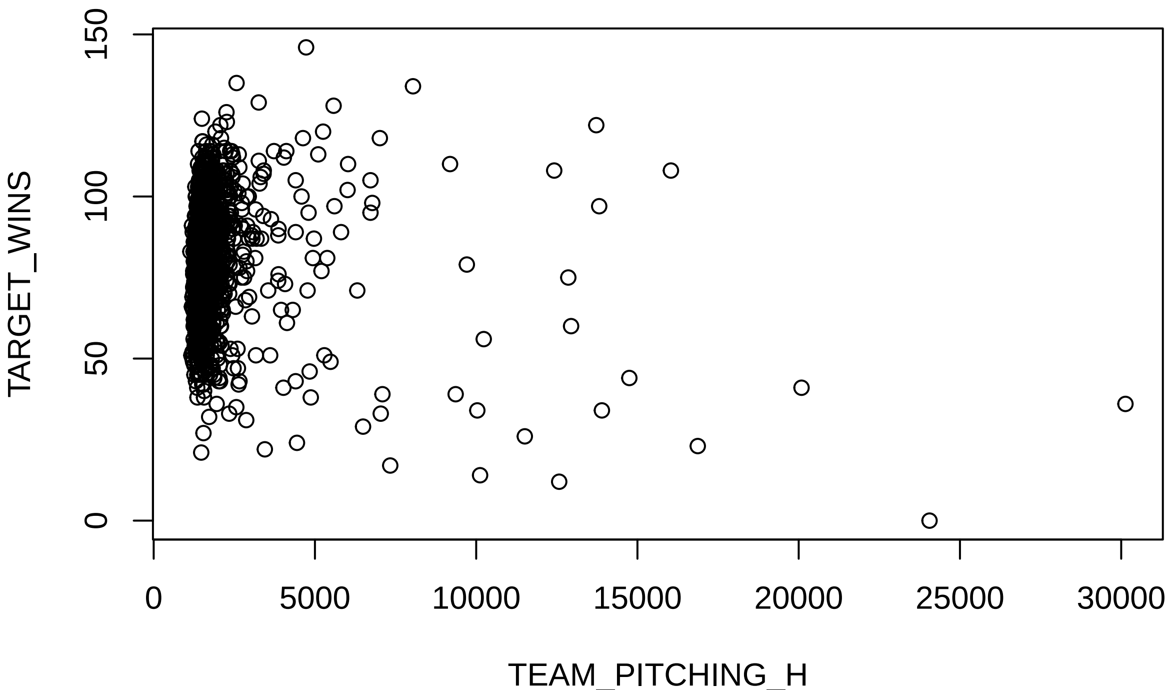
Looking at our residual plot above, there seems to be a clustering of residuals along the x-axis at $X \approx 80$. This shows a pattern in our residuals.

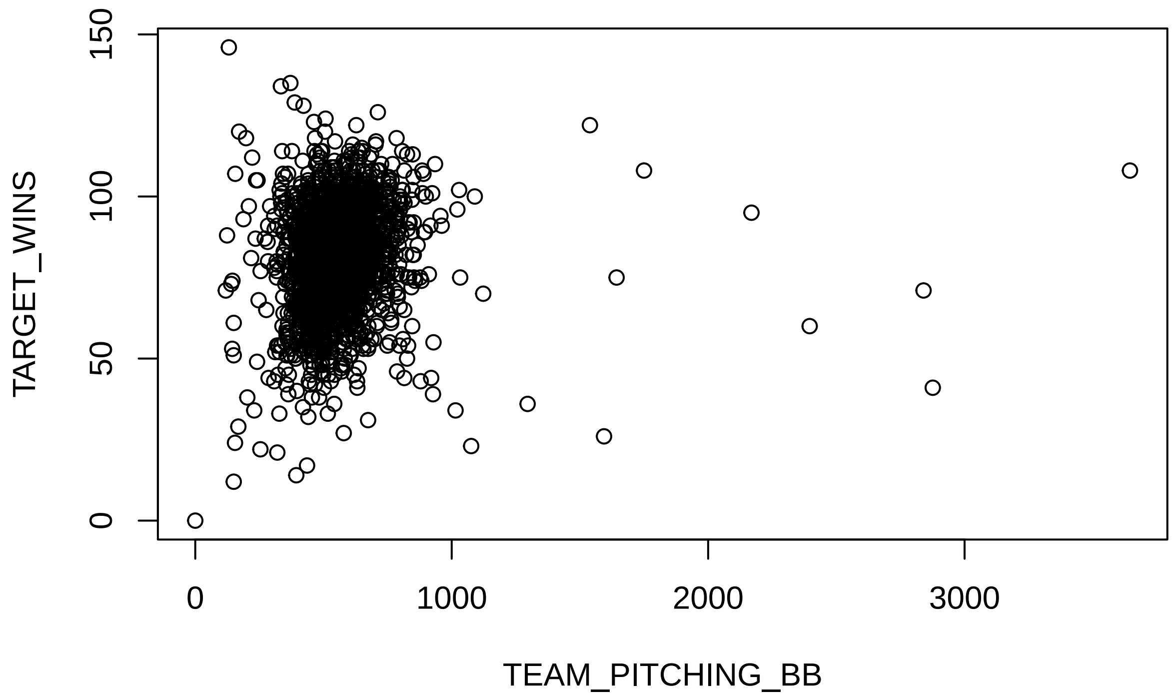


Let's plot our response variable (*Total Wins*) versus each of our predictor variables to get a sense of linear relationships.


TARGET_WINS
TEAM_BATTING_H
0
50
100
150
1000
1500
2000
2500


TARGET_WINS
0
50
100
150
0
200
400
600
800
TEAM_BATTING_BB


150
100
50
0
TARGET_WINS
0
5000
10000
15000
20000
25000
30000
TEAM_PITCHING_H



Model Evaluation

We'll need to read in our evaluation data, which is hosted on GitHub for reproduceability.

```r
predict(lm_all, test)
```

```
##        1        2        3        4        5        6        7        8
##       NA       NA       NA 79.60984       NA       NA       NA       NA
##        9       10       11       12       13       14       15       16
##       NA       NA       NA       NA       NA       NA       NA       NA
##       17       18       19       20       21       22       23       24
##       NA 78.95693       NA       NA       NA       NA       NA       NA
##       25       26       27       28       29       30       31       32
## 77.16939 86.81801       NA       NA       NA       NA       NA       NA
##       33       34       35       36       37       38       39       40
##       NA       NA       NA       NA       NA       NA       NA       NA
##       41       42       43       44       45       46       47       48
##       NA       NA       NA       NA       NA       NA       NA       NA
##       49       50       51       52       53       54       55       56
##       NA       NA       NA       NA       NA       NA       NA       NA
##       57       58       59       60       61       62       63       64
##       NA       NA       NA       NA       NA       NA       NA 85.05198
##       65       66       67       68       69       70       71       72
## 81.33195       NA       NA       NA       NA       NA       NA       NA
```

```
##        73       74       75       76       77       78       79       80
##        NA       NA       NA       NA       NA       NA       NA       NA
##        81       82       83       84       85       86       87       88
##        NA       NA       NA       NA       NA       NA       NA       NA
##        89       90       91       92       93       94       95       96
##        NA       NA       NA       NA       NA       NA       NA       NA
##        97       98       99      100      101      102      103      104
##        NA       NA       NA       NA       NA       NA       NA       NA
##       105      106      107      108      109      110      111      112
##        NA       NA       NA 72.39264 87.56175       NA       NA       NA
##       113      114      115      116      117      118      119      120
##        NA       NA       NA       NA       NA 74.49284 65.15701       NA
##       121      122      123      124      125      126      127      128
##        NA       NA       NA       NA       NA       NA       NA       NA
##       129      130      131      132      133      134      135      136
##        NA       NA       NA       NA       NA       NA 86.10463       NA
##       137      138      139      140      141      142      143      144
##        NA       NA       NA       NA       NA       NA       NA       NA
##       145      146      147      148      149      150      151      152
##        NA       NA       NA       NA       NA       NA       NA       NA
##       153      154      155      156      157      158      159      160
##        NA       NA       NA       NA 86.64915       NA       NA       NA
##       161      162      163      164      165      166      167      168
##        NA       NA       NA       NA       NA       NA       NA       NA
##       169      170      171      172      173      174      175      176
##        NA       NA       NA       NA       NA       NA       NA       NA
##       177      178      179      180      181      182      183      184
##        NA       NA       NA       NA       NA       NA       NA 88.27315
##       185      186      187      188      189      190      191      192
##        NA       NA       NA       NA       NA       NA       NA       NA
##       193      194      195      196      197      198      199      200
##        NA       NA       NA       NA       NA       NA       NA       NA
##       201      202      203      204      205      206      207      208
##        NA       NA       NA       NA       NA       NA       NA       NA
##       209      210      211      212      213      214      215      216
##        NA       NA       NA       NA       NA       NA       NA       NA
##       217      218      219      220      221      222      223      224
##        NA       NA       NA       NA       NA       NA 77.10932 65.54638
##       225      226      227      228      229      230      231      232
##        NA       NA       NA 69.38398 79.72822       NA       NA       NA
##       233      234      235      236      237      238      239      240
##        NA       NA       NA       NA       NA       NA       NA       NA
##       241      242      243      244      245      246      247      248
##        NA       NA       NA       NA       NA       NA       NA       NA
##       249      250      251      252      253      254      255      256
##        NA 78.12011 74.97230       NA       NA       NA       NA       NA
##       257      258      259
##        NA       NA       NA
```

Appendix: Report Code

Below is the code for this report to generate the models and charts above.

```r
knitr::opts_chunk$set(echo = TRUE)
library(glue)
library(tidyverse)
library(car)
df <- read.csv("https://raw.githubusercontent.com/andrewbowen19/businessAnalyticsDataMiningDATA621/main/
df <- data.frame(df)
mean_wins <- mean(df$TARGET_WINS)
median_wins <- median(df$TARGET_WINS)
sd_wins <- sd(df$TARGET_WINS)

# Print summary stats
print(glue("The mean number of wins in a season is {mean_wins}"))
print(glue("The median number of wins in a season is {median_wins}"))
print(glue("The standard deviation for number of wins in a season is {sd_wins}"))
ggplot(df, aes(x=TARGET_WINS)) + geom_histogram()
ggplot(df, aes(x=TARGET_WINS)) + geom_boxplot()
train <- subset(df, select=-c(INDEX))
cor(train, df$TARGET_WINS)
lm_all <- lm(TARGET_WINS~., train)
coef(lm_all)
plot(lm_all)

# Create model with select inputs (walks and hits allowed/gained)
lm_select <- lm(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_PITCHING_H + TEAM_PITCHING_BB, tra

summary(lm_select)
plot(lm_select)

# Plot selective model residuals
plot(lm_select$residuals)


# Plot our response variable for each predictor variable to get a sense of
plot(TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_PITCHING_H + TEAM_PITCHING_BB, data=train)

eval_data_url <- "https://raw.githubusercontent.com/andrewbowen19/businessAnalyticsDataMiningDATA621/mai

test <- read.csv(eval_data_url)
predict(lm_all, test)
```