DATA 608: Homework 1 (Baseball Regression)

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

2023-09-04

First, let's read in the provided dataset

Data Exploration

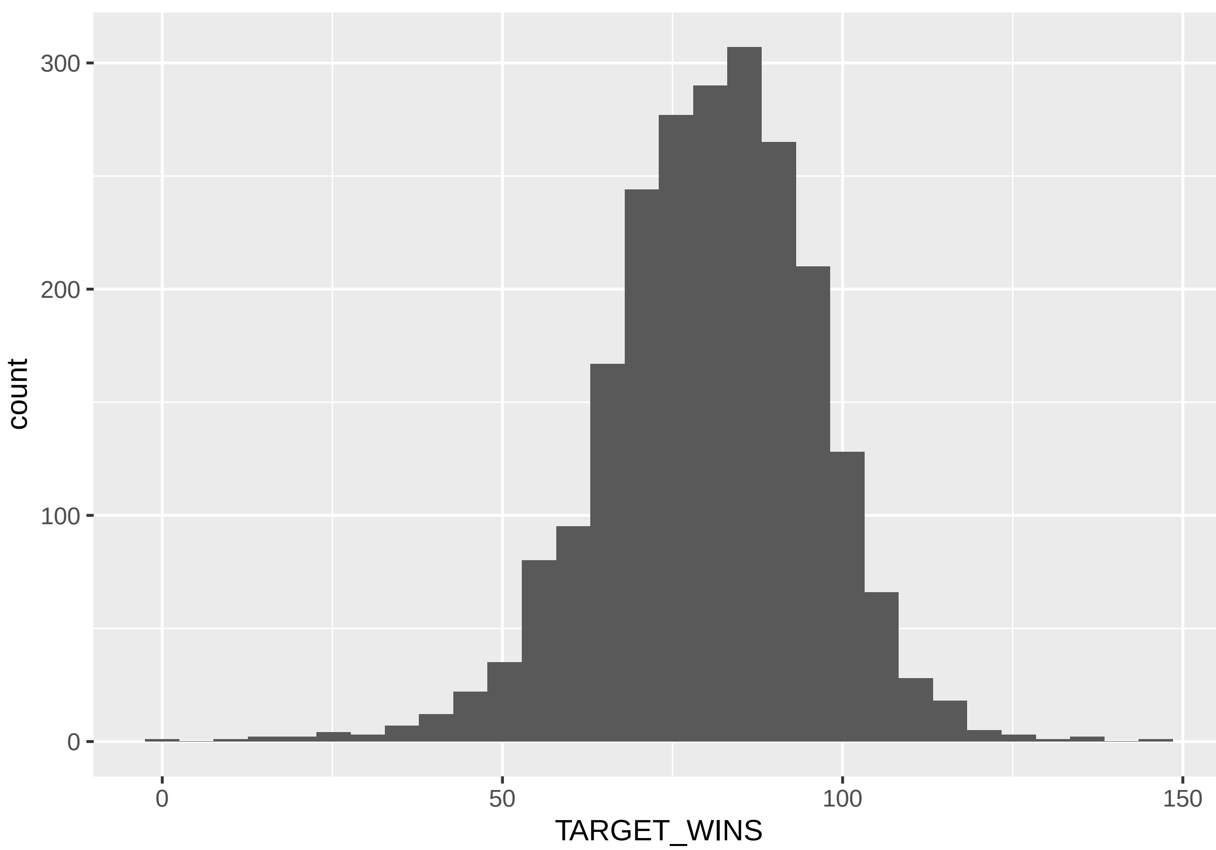
First, let's print out some summary statistics. We're primarily interested in the `TARGET_WINS` feature, so we'll look at that first

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

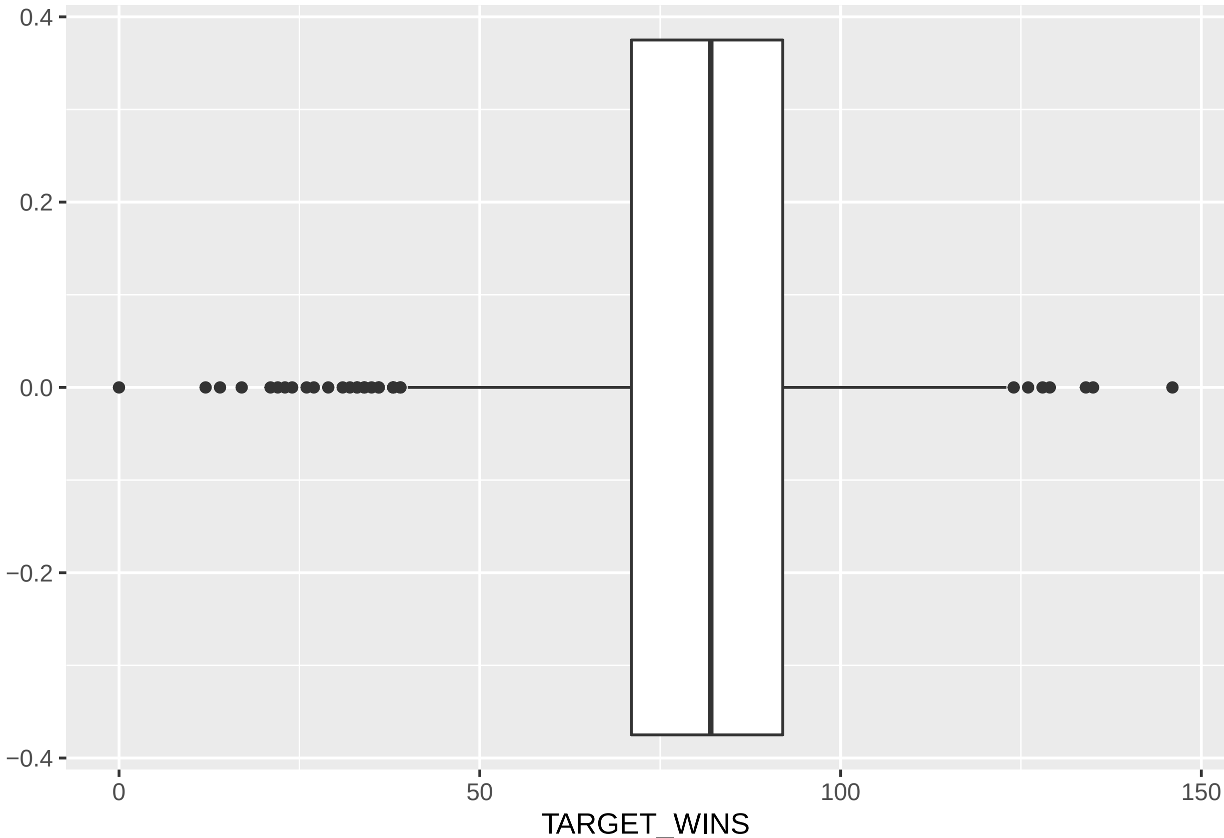
```
## The median number of wins in a season is 82
```

```
## 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:

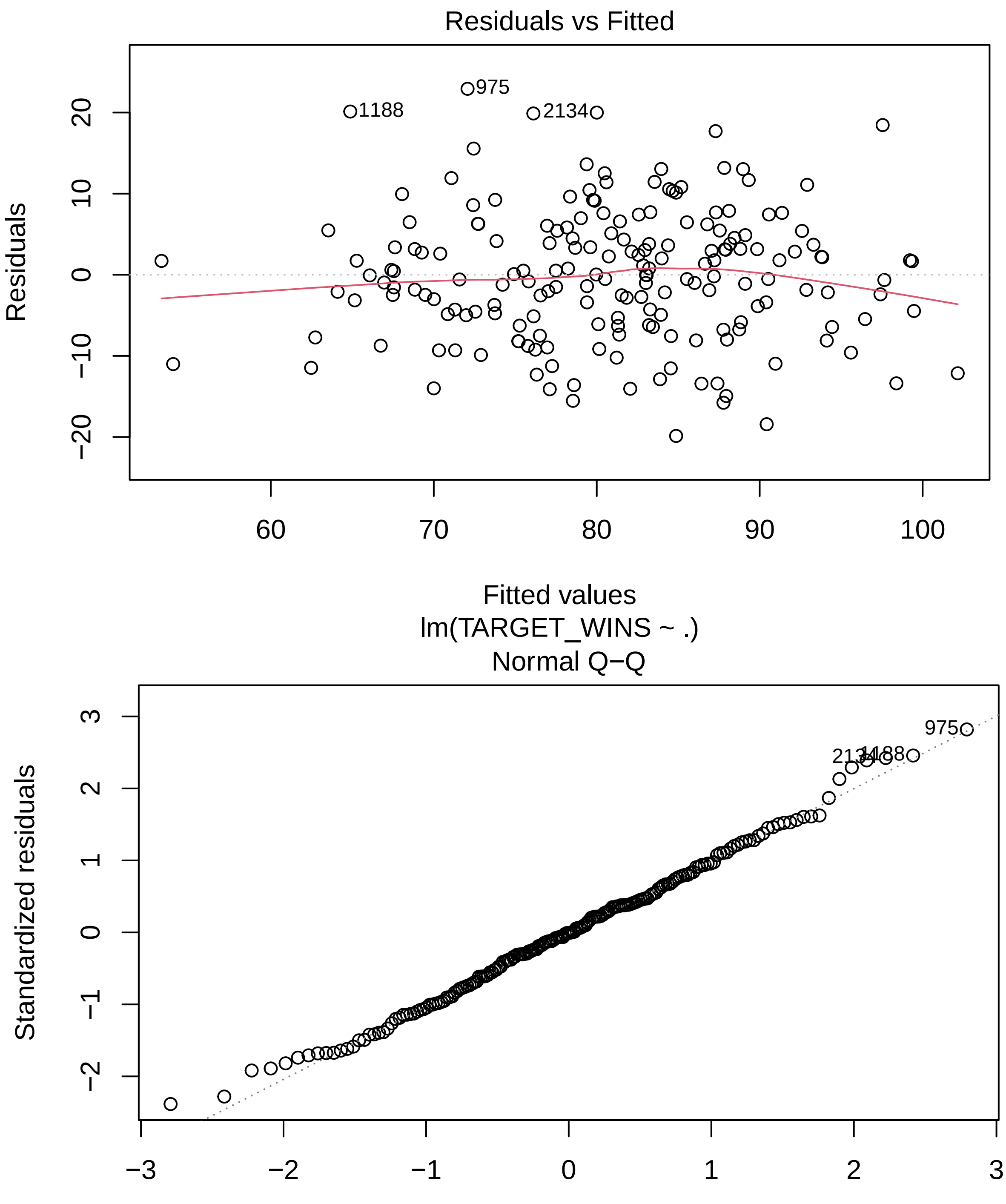
```
##                         [,1]
## TARGET_WINS       1.0000000
## TEAM_BATTING_H    0.3887675
## TEAM_BATTING_2B   0.2891036
## TEAM_BATTING_3B   0.1426084
## TEAM_BATTING_HR   0.1761532
## TEAM_BATTING_BB   0.2325599
## TEAM_BATTING_SO          NA
## TEAM_BASERUN_SB          NA
## TEAM_BASERUN_CS          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
```

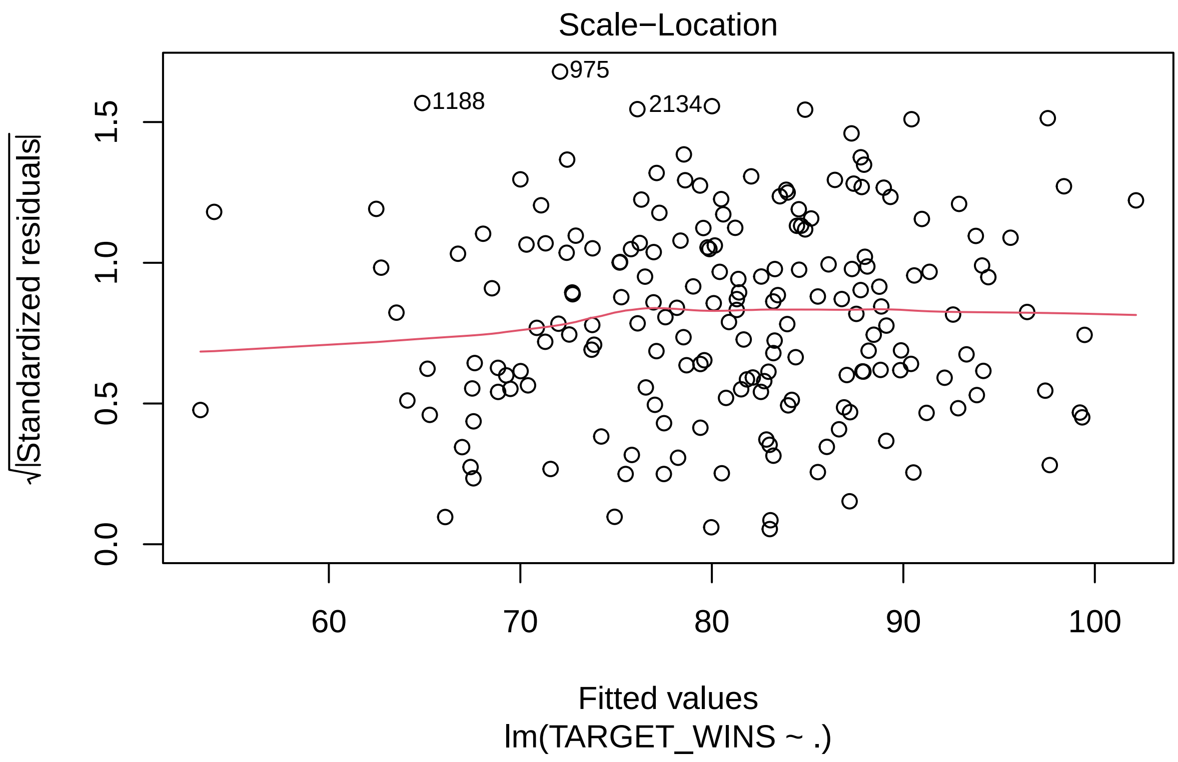
Let's make a basic model with some offensive inputs (hits, 2B, 3B, Home Runs)

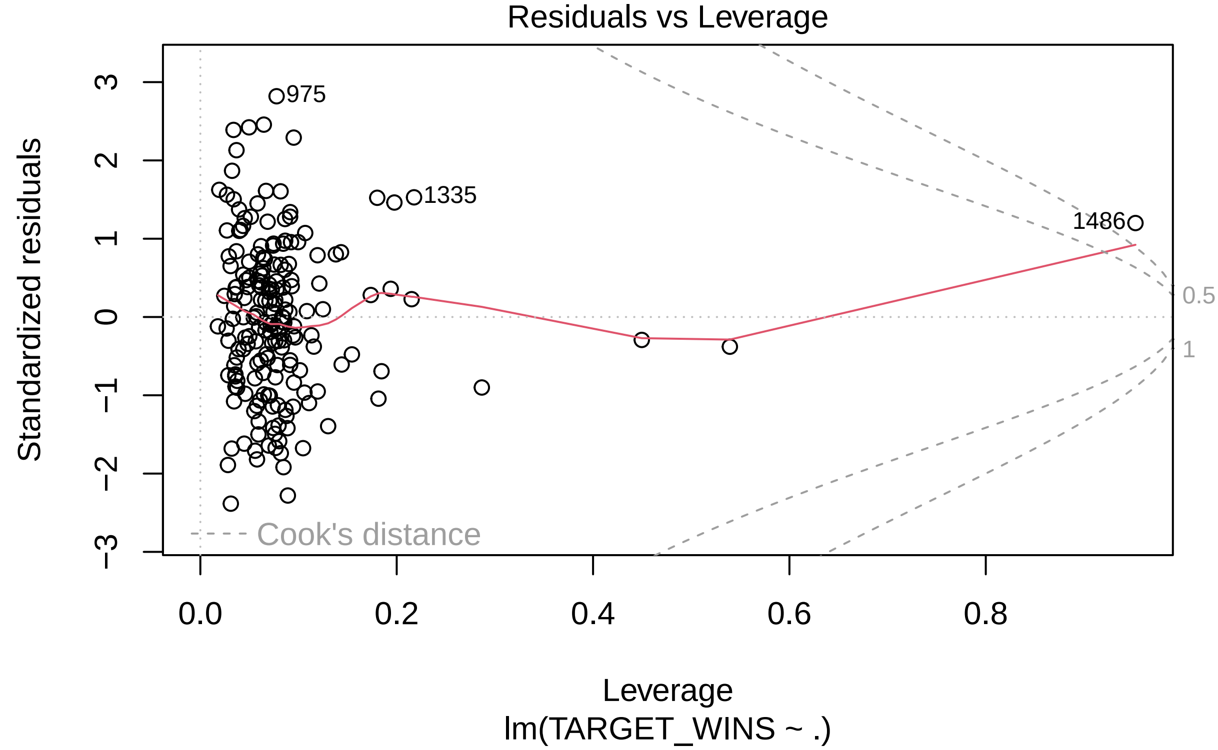
```
##
## Call:
```

```
## lm(formula = TARGET_WINS ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8708  -5.6564  -0.0599   5.2545  22.9274
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      60.28826   19.67842   3.064  0.00253 **
## TEAM_BATTING_H    1.91348    2.76139   0.693  0.48927
## TEAM_BATTING_2B   0.02639    0.03029   0.871  0.38484
## TEAM_BATTING_3B  -0.10118    0.07751  -1.305  0.19348
## TEAM_BATTING_HR  -4.84371   10.50851  -0.461  0.64542
## TEAM_BATTING_BB  -4.45969    3.63624  -1.226  0.22167
## TEAM_BATTING_SO   0.34196    2.59876   0.132  0.89546
## TEAM_BASERUN_SB   0.03304    0.02867   1.152  0.25071
## TEAM_BASERUN_CS  -0.01104    0.07143  -0.155  0.87730
## TEAM_BATTING_HBP  0.08247    0.04960   1.663  0.09815 .
## TEAM_PITCHING_H  -1.89096    2.76095  -0.685  0.49432
## TEAM_PITCHING_HR  4.93043   10.50664   0.469  0.63946
## TEAM_PITCHING_BB  4.51089    3.63372   1.241  0.21612
## TEAM_PITCHING_SO -0.37364    2.59705  -0.144  0.88577
## TEAM_FIELDING_E  -0.17204    0.04140  -4.155 5.08e-05 ***
## TEAM_FIELDING_DP -0.10819    0.03654  -2.961  0.00349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.467 on 175 degrees of freedom
##   (2085 observations deleted due to missingness)
## Multiple R-squared:  0.5501, Adjusted R-squared:  0.5116
## F-statistic: 14.27 on 15 and 175 DF,  p-value: < 2.2e-16
```

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


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET_WINS ~ .)
975
1188
2134
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(TARGET_WINS ~ .)
975
2134
1188

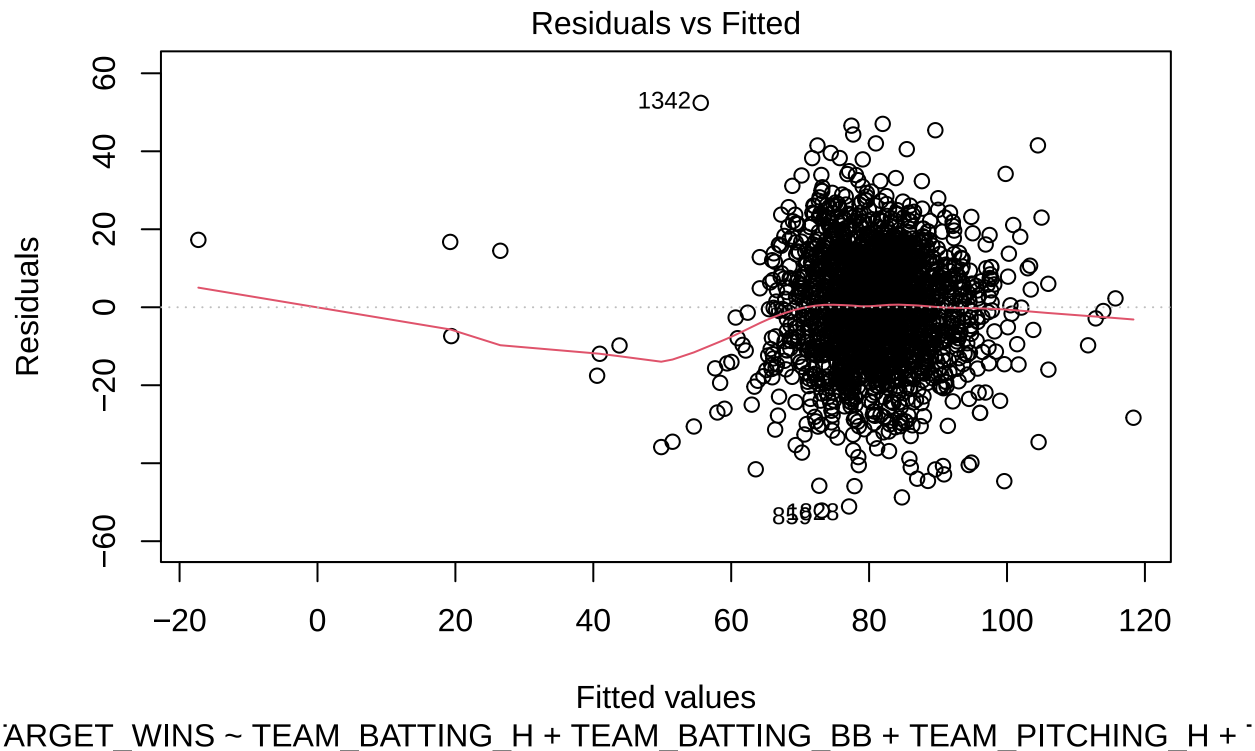


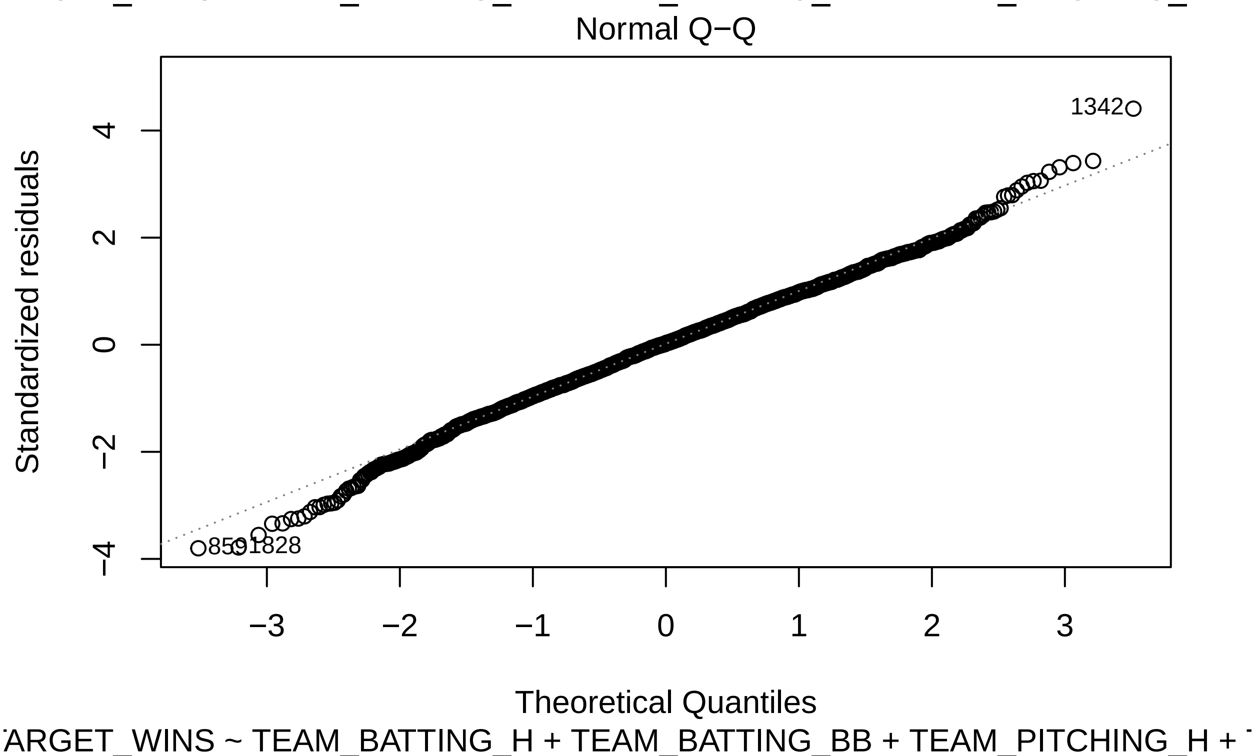


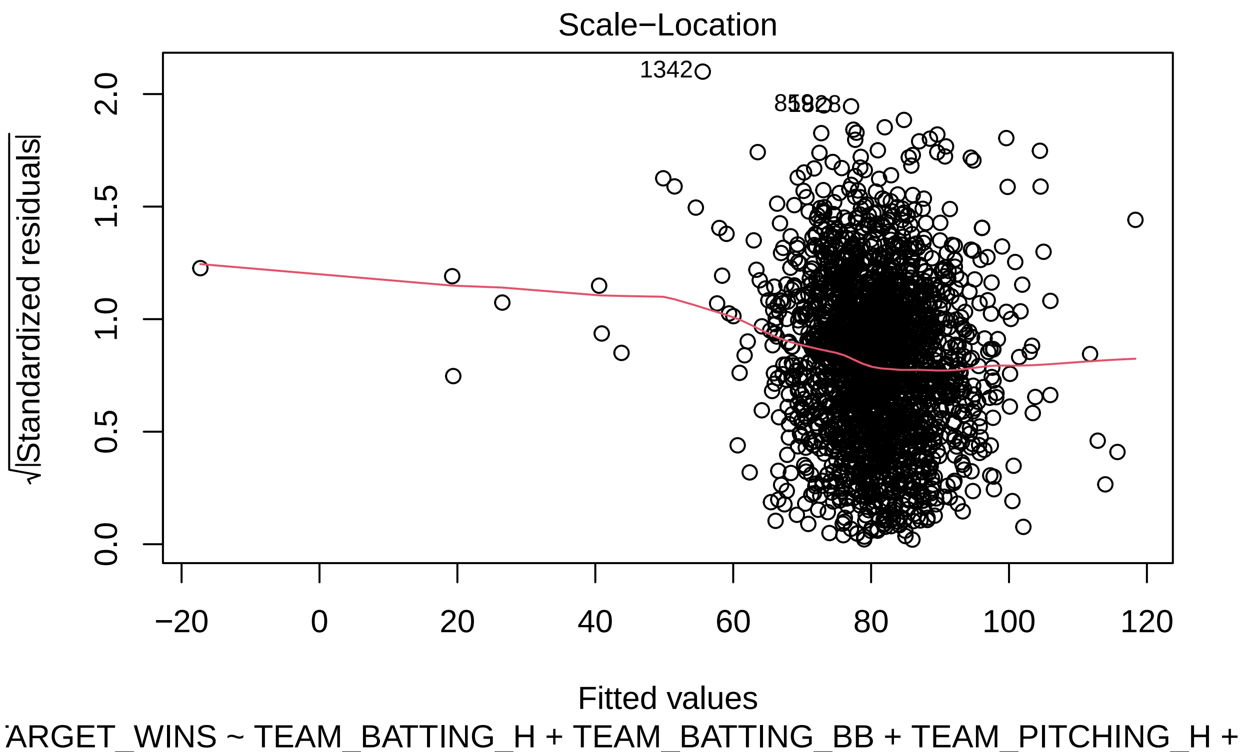
Now we can make a model with inputs that we know from baseball.

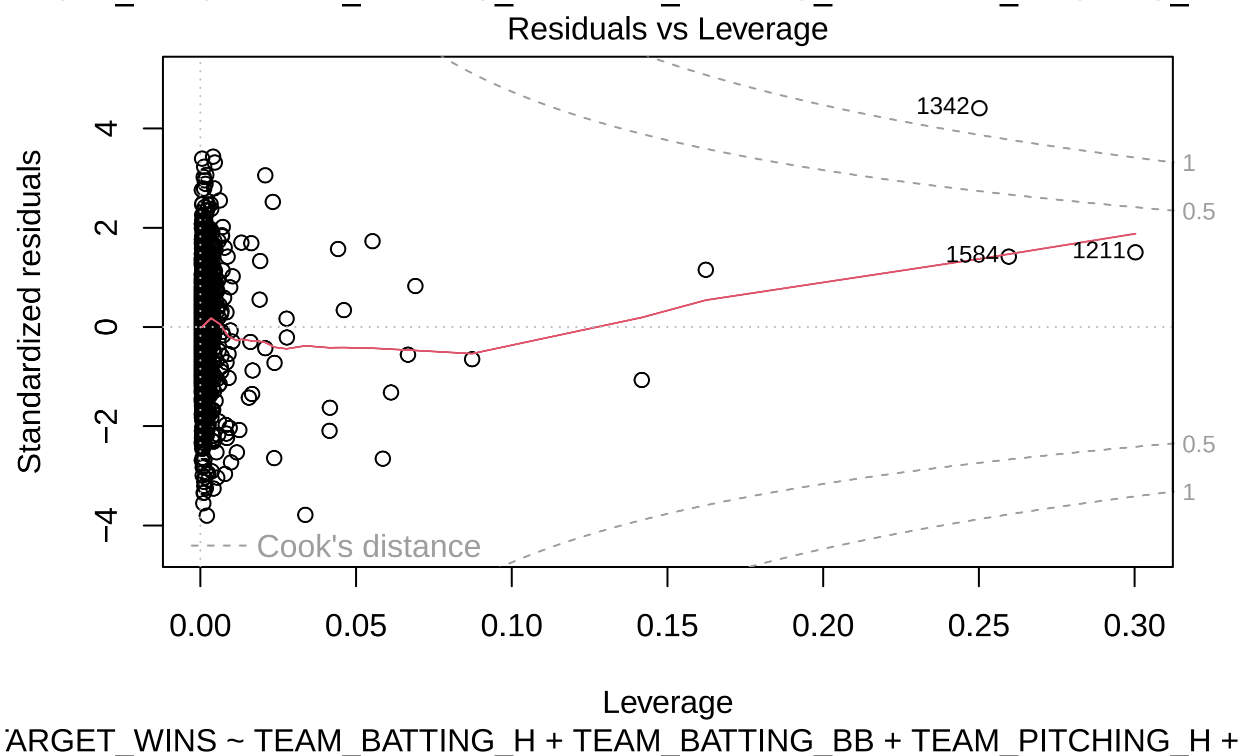
- Total hits (`TEAM_BATTING_H`)
- Total walks gained (`TEAM_BATTING_BB`)
- Total hits allowed (`TEAM_PITCHING_H`)
- Total walks allowed (`TEAM_PITCHING_BB`)

The thinking being here that good teams generally tend to get on base more frequently (`TEAM_BATTING_HITS` and `TEAM_BATTING_BB`) while allowing *fewer* runners on base (Negative predictor variables `TEAM_PITCHING_H` and `TEAM_PITCHING_BB`)

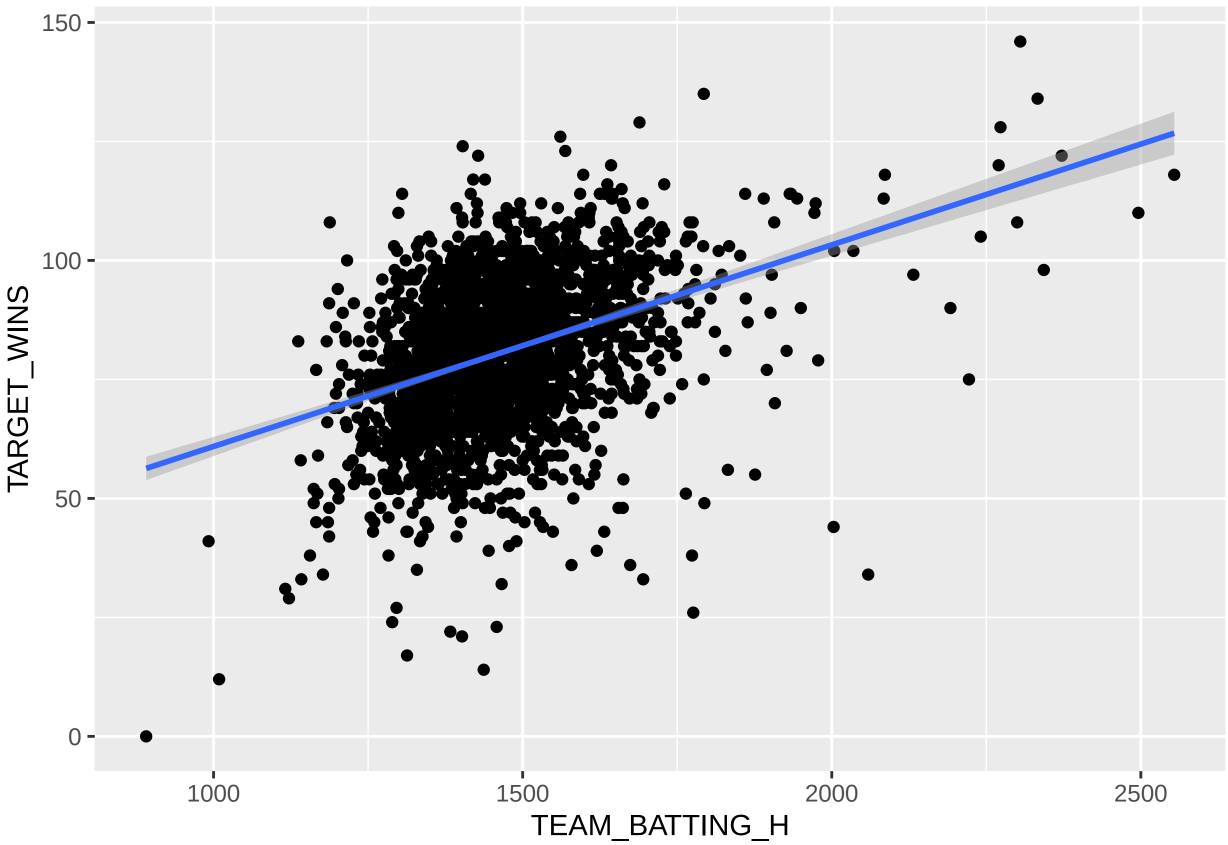








It's interesting to not that with selected variables (walks and hits gained/allowed per team) that our adjusted R^2 actually went *down*, indicating the amount of variability in `TARGET_WINS` explained by our more selectvie walks/hits model is *less* than the model including all variables.



Model Evaluation

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

Appendix: Report Code

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

```
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)
summary(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

plot(lm_select)
# Plotting total wins  versus a teams hits
ggplot(train, aes(x=TEAM_BATTING_H, y=TARGET_WINS)) +
  geom_point() +
  stat_smooth(method = "lm")
eval_data_url <- "https://raw.githubusercontent.com/andrewbowen19/businessAnalyticsDataMiningDATA621/mai

test <- read.csv(eval_data_url)
```