

DATA 621: Homework 1 (Group 2)

Moneyball Linear Regression

Group 2 members: *Alice Friedman, Diego Correa, Jagdish Chhabria, Orli Khaimova, Richard Zheng, Stephen Haslett.*

Assignment Objective

In this assignment, we analyze and model a baseball dataset containing multi-year game statistics for different teams. The objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. We can only use the variables given to us (or variables that we derive from the variables provided).

Data

There are 2 datasets provided - The Moneyball training dataset contains 17 columns and 2276 rows. Each record in the Money Ball training dataset represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season. For this assignment, the target variable in the dataset is TARGET_WINS.

Below is a short description of the variables of interest in the data set:

Purpose of Analysis

The purpose of the analysis is to find which of the predictors have significant ability to explain the variation in the response variable (number of wins by a team), and to make a prediction for all the records provided in the test data set.

Method

The method used is a multiple linear regression model on the training data to predict the number of wins for the team.

Data Exploration

The first variable in the above table (INDEX) was dropped from the dataset due to the fact that it is nearly a row identifier, and has no impact on the target variable (TARGET_WINS).

Summary Statistics

The first step in our data exploration was to compile summary statistics to give us some insight into the data prior to preparing the data for modeling. To make the variable names more readable, we removed the "TEAM_" prefix from each variable.

##	Wins	Hits	Doubles	Triples
##	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

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

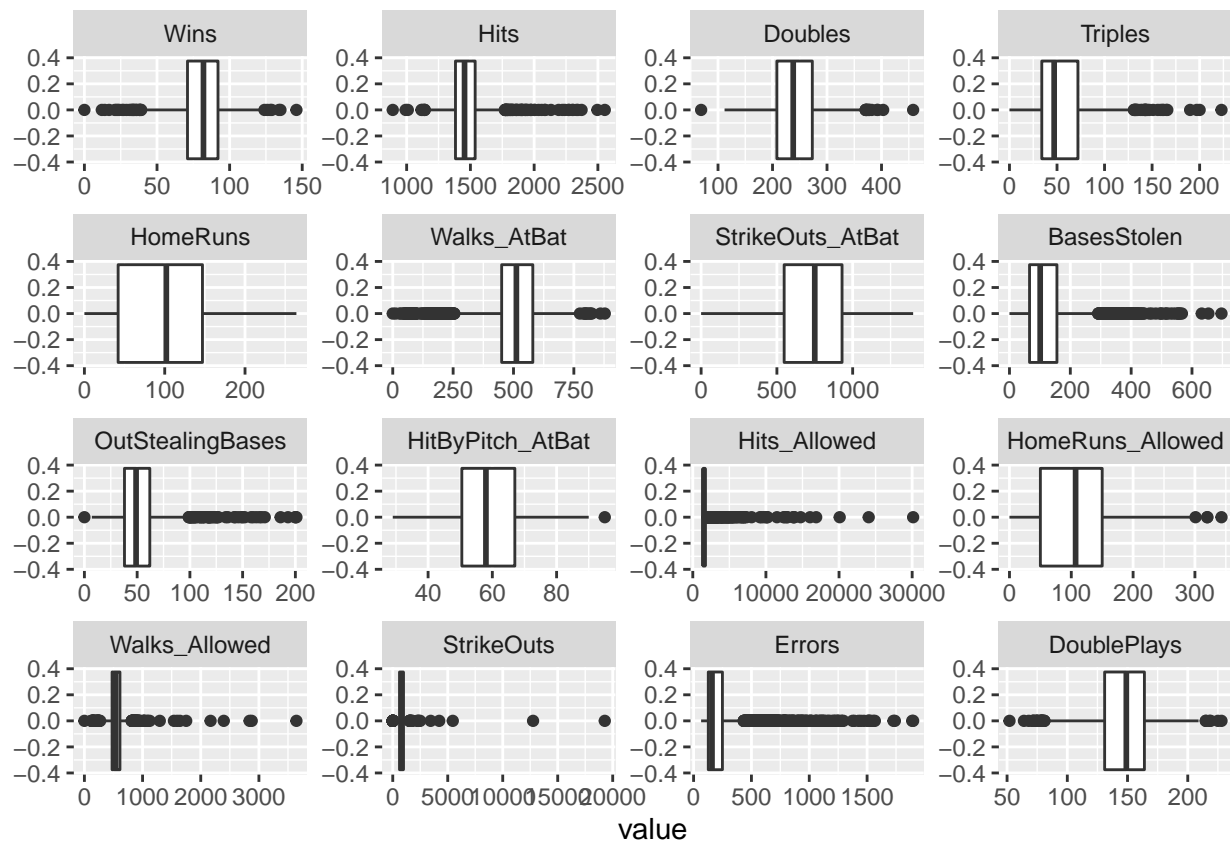
Figure 1: Variables of Interest

```

## 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
##
##      HomeRuns      Walks_AtBat      StrikeOuts_AtBat      BasesStolen
## 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
## OutStealingBases HitByPitch_AtBat Hits_Allowed HomeRuns_Allowed
## 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
## Walks_Allowed      StrikeOuts      Errors      DoublePlays
## 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

```

From the above, we see that there are 15 predictors and 1 response variable (Wins). Of the predictors, 6 have missing values. We then plotted boxplots for all the variables to get a sense of outliers.

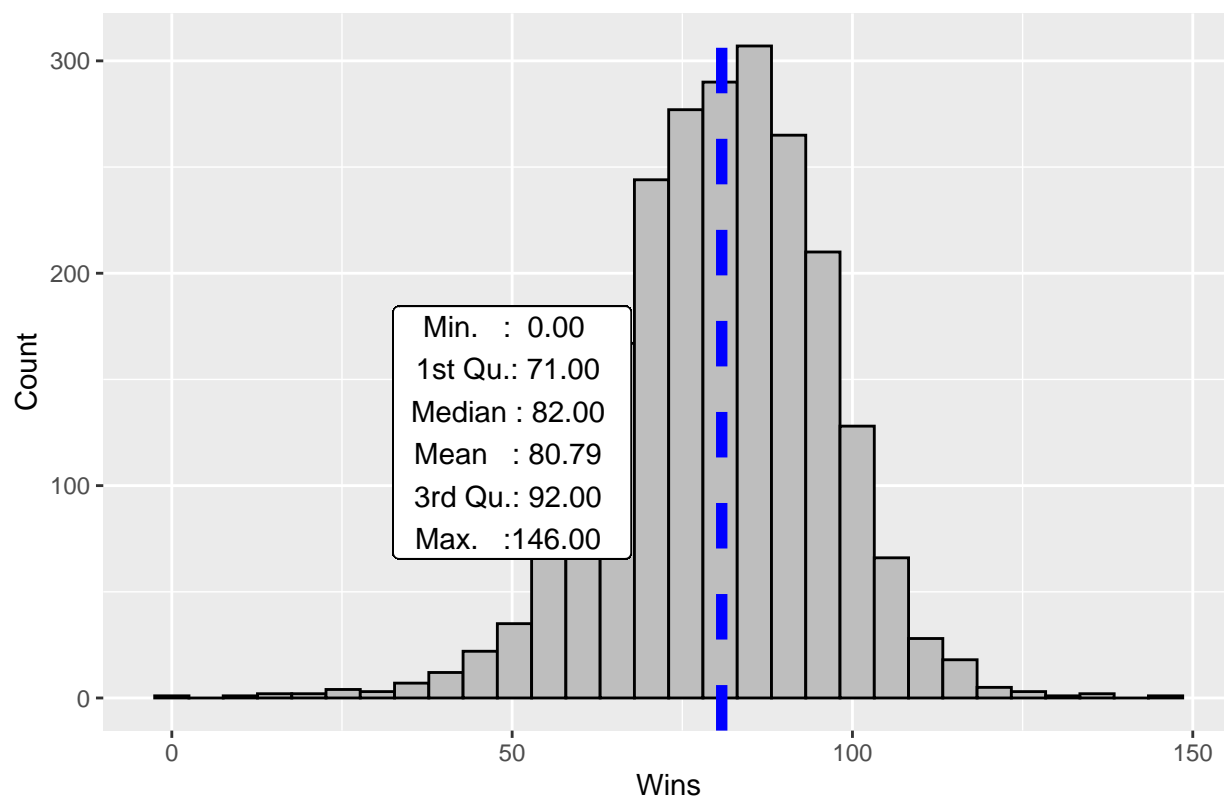


From the box plots, we can see that quite a few predictors are very skewed in nature, such as Walks_Allowed and Hits_Allowed.

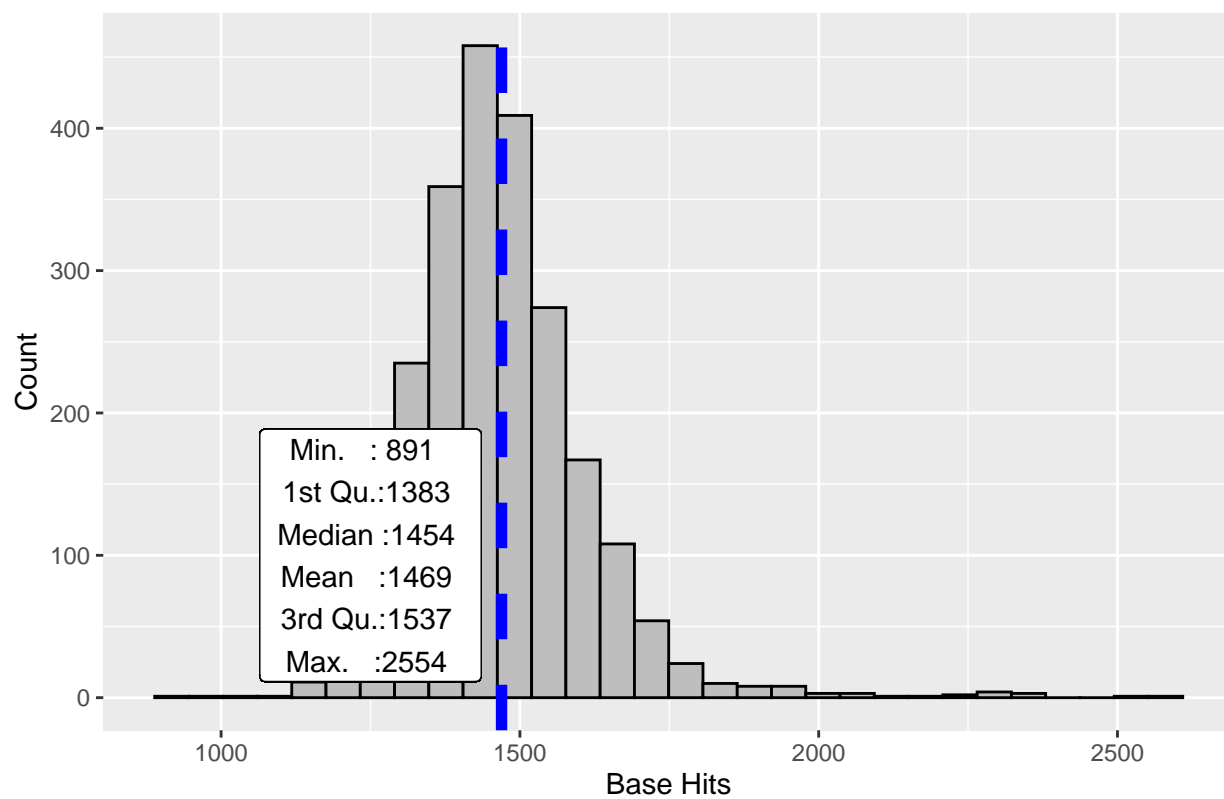
Variable Distributions

We created distribution plots for all the variables to check their shape visually and get a high-level, intuitive sense of normality.

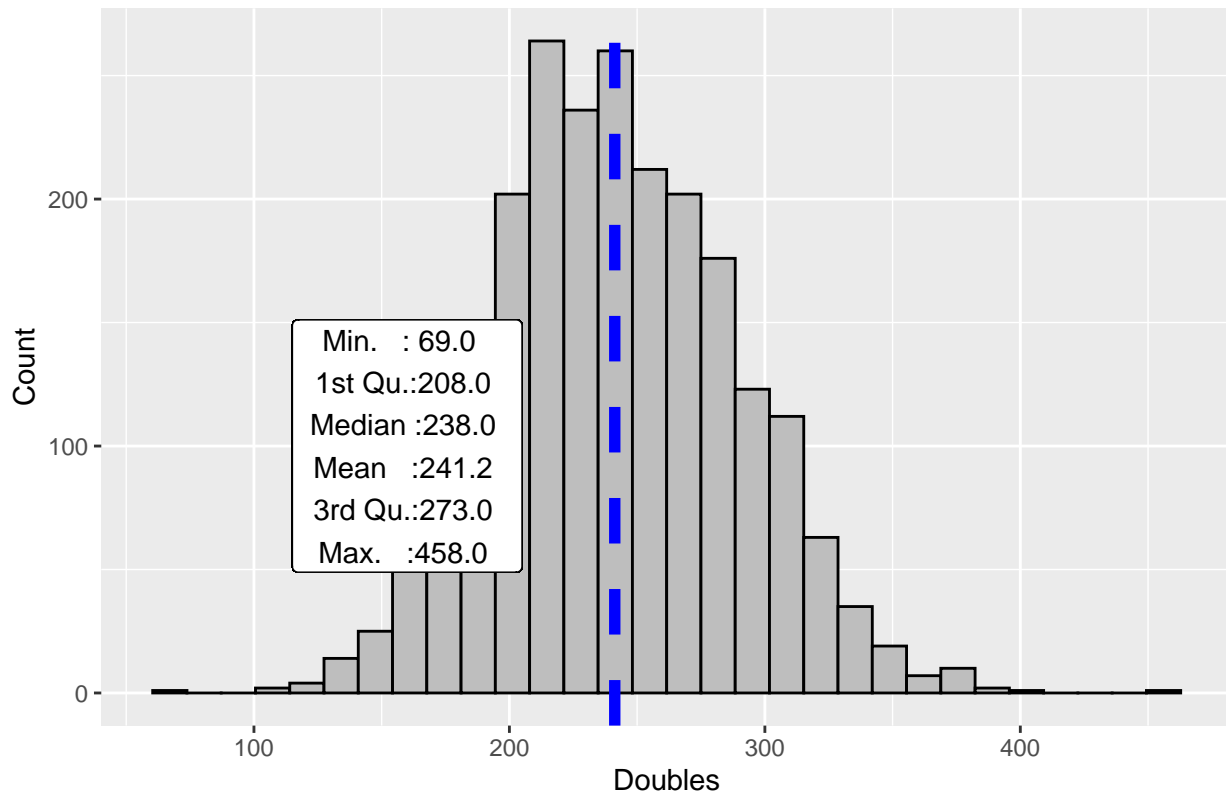
Wins Histogram Plot



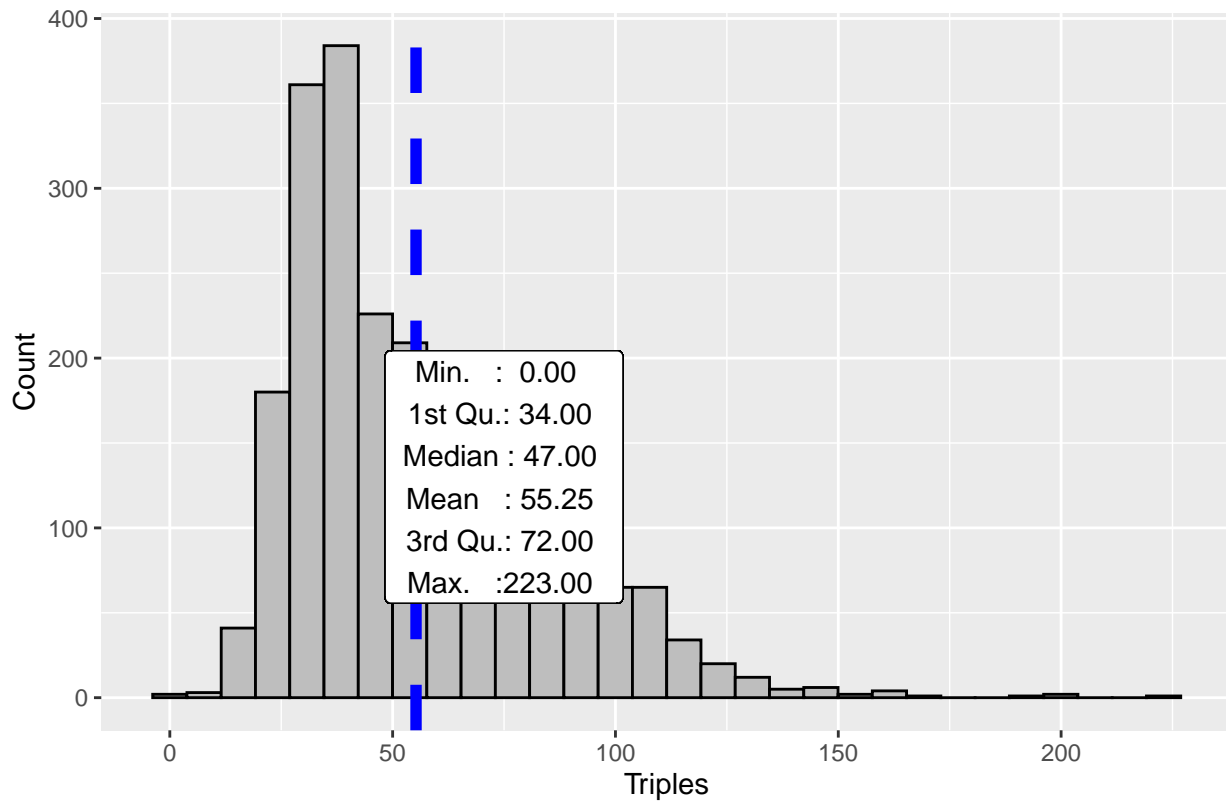
Hits Histogram Plot



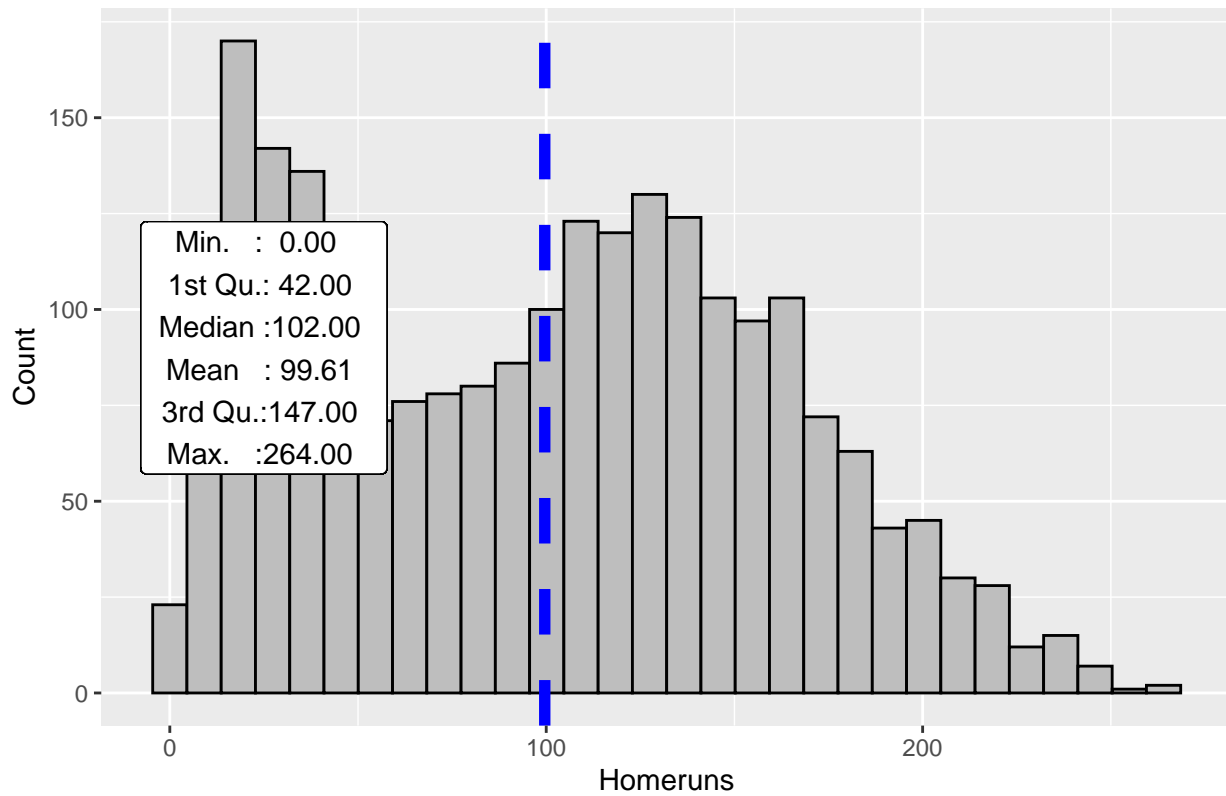
Doubles Histogram Plot



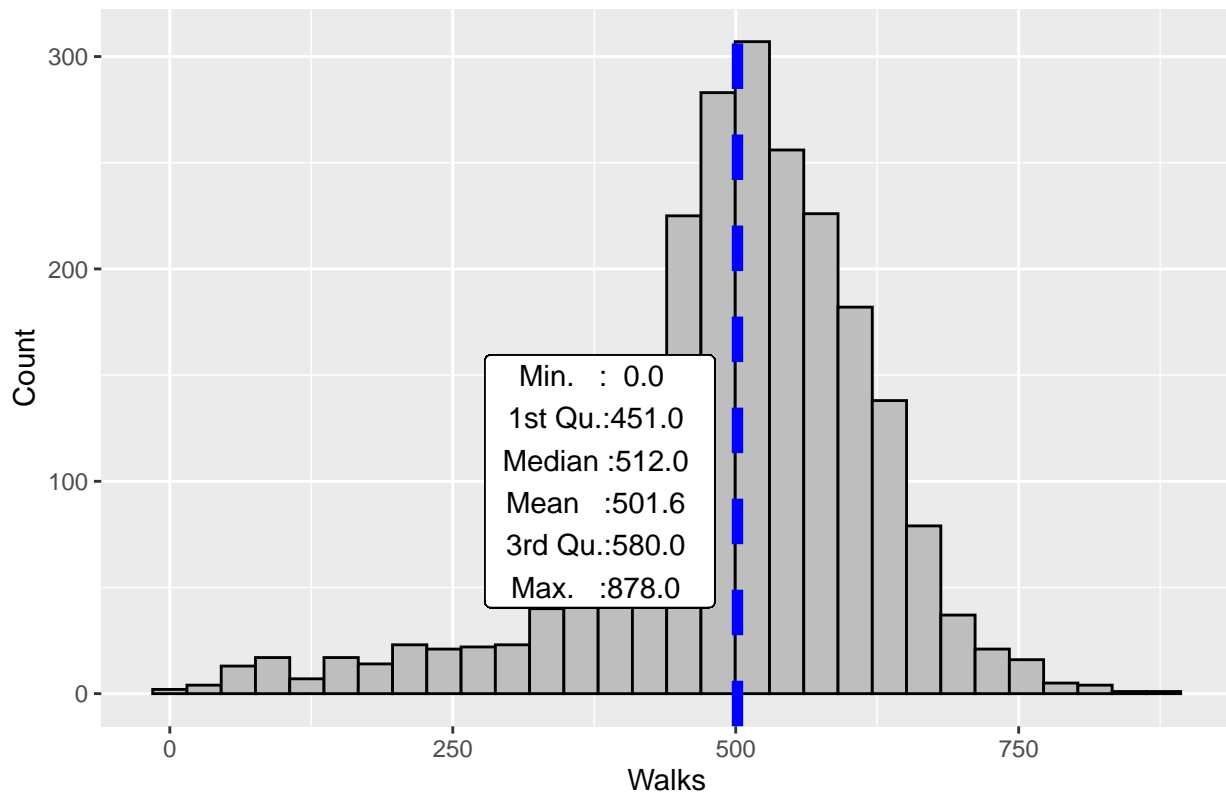
Triples Histogram Plot



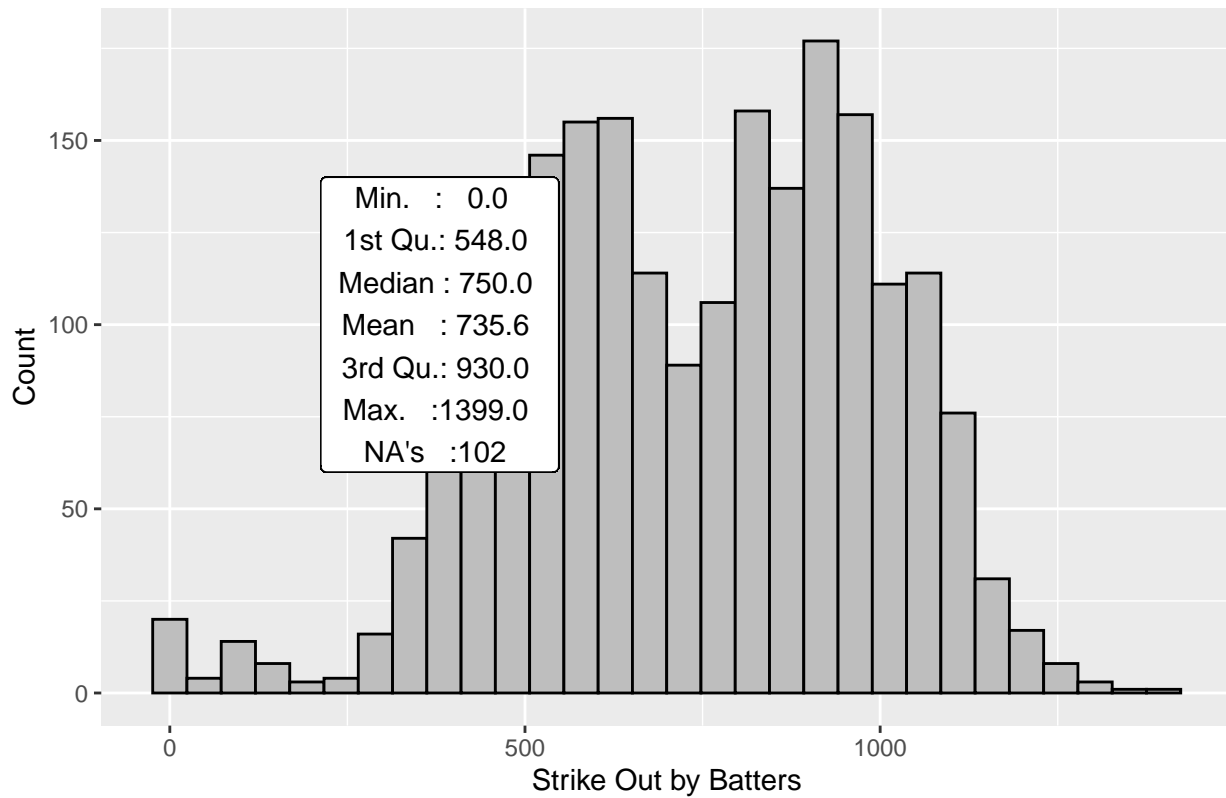
Homeruns Histogram Plot



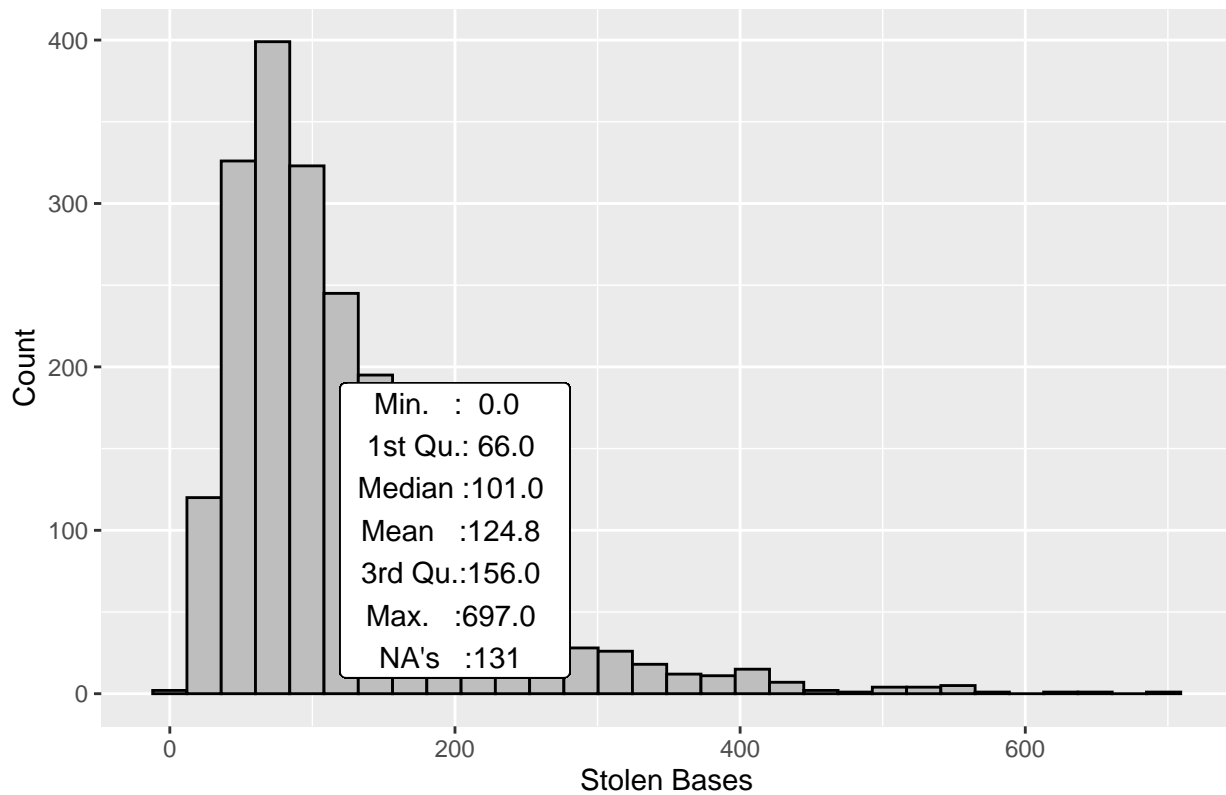
Walks Histogram Plot



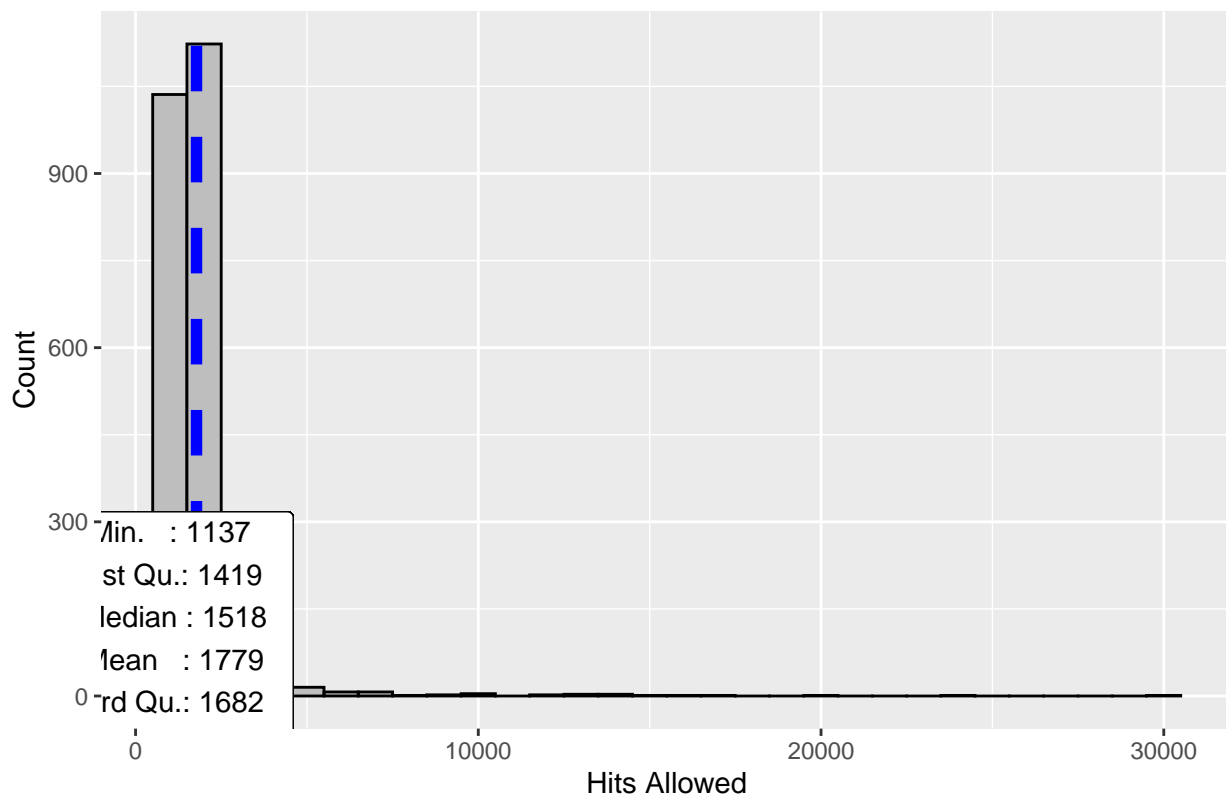
Strike Out by Batters Histogram Plot



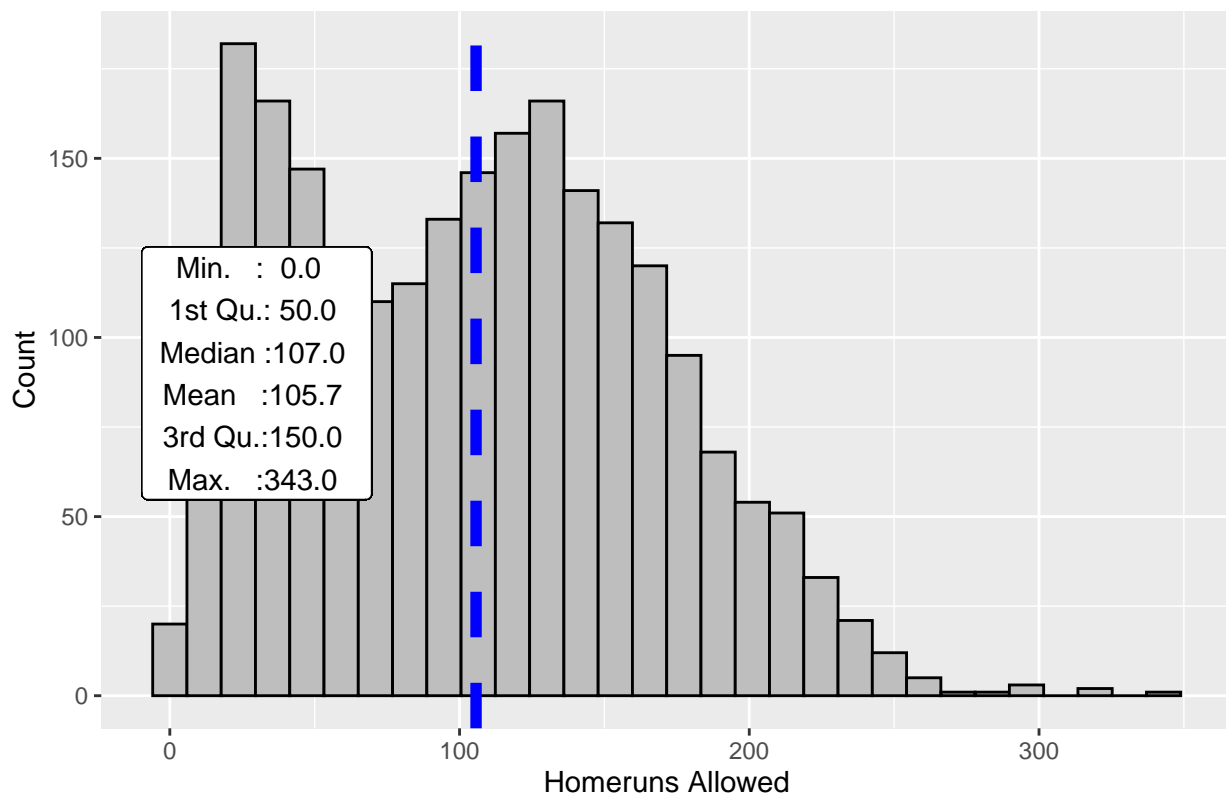
Stolen Bases Histogram Plot



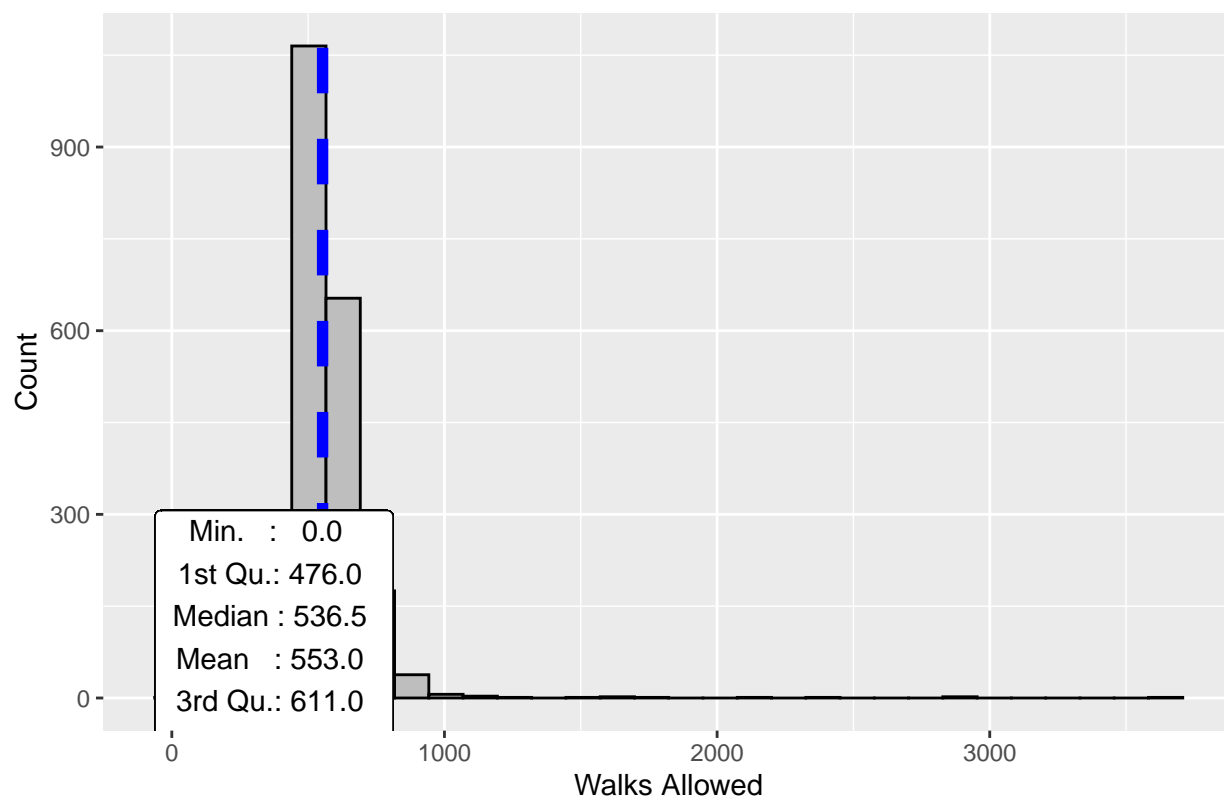
Hits Allowed Histogram Plot



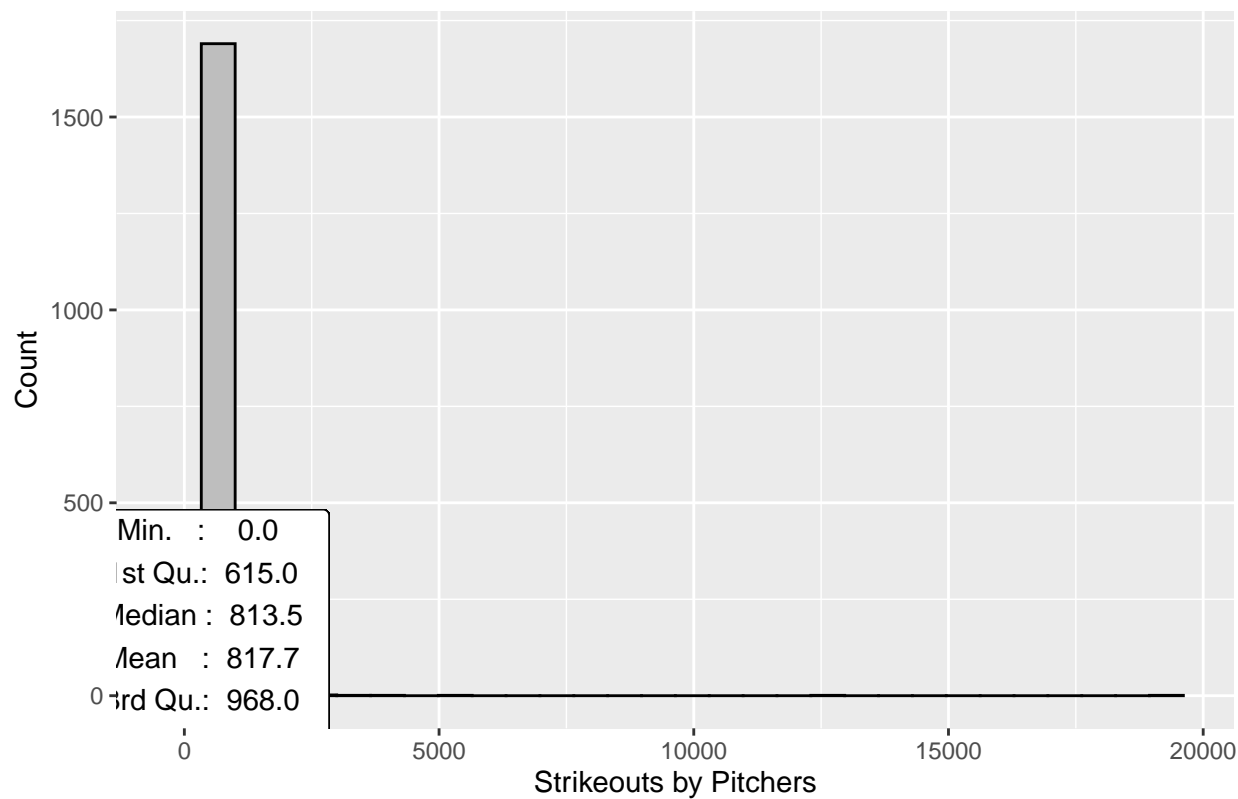
Homeruns Allowed Histogram Plot

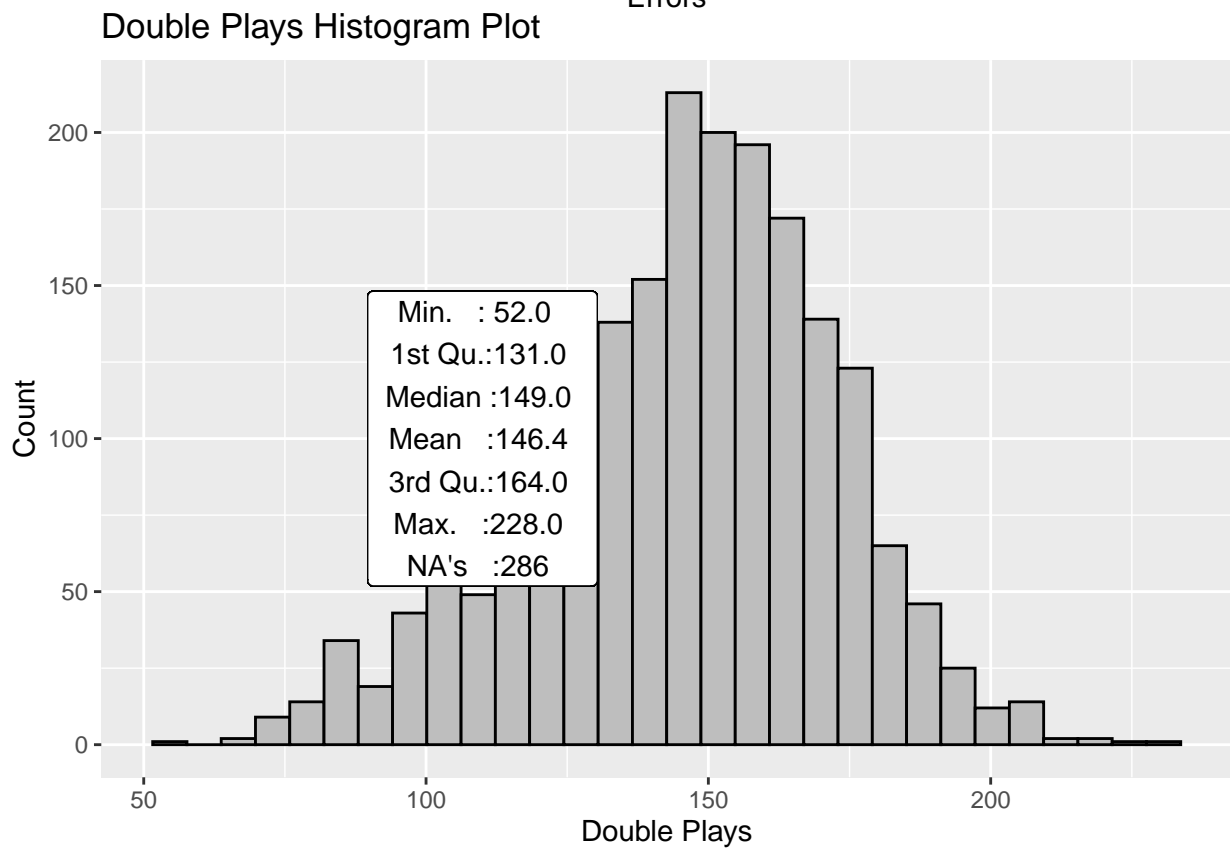
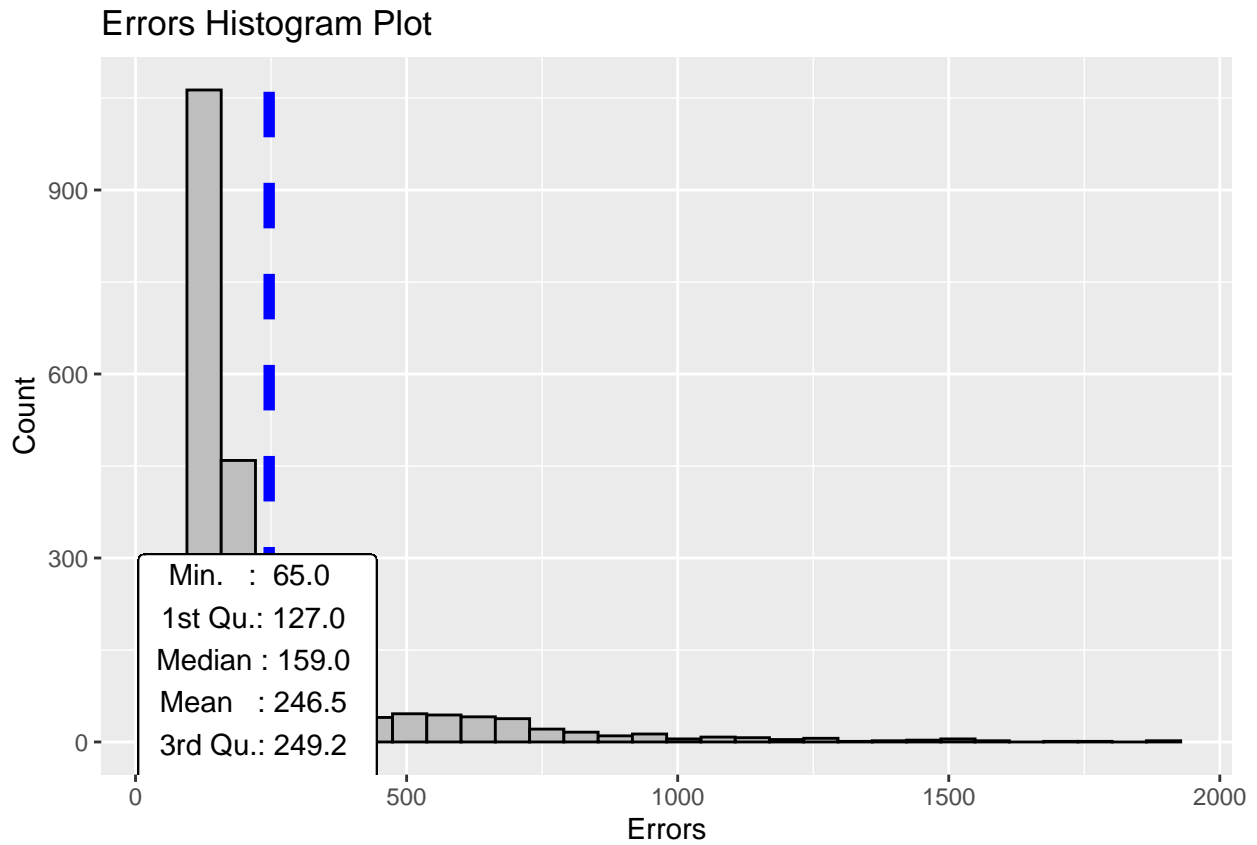


Walks Allowed Histogram Plot



Strikeouts by Pitchers Histogram Plot





The histograms provide additional confirmation that some of the variables are quite skewed. For example:

Table 1: Breakdown of Variables by Percentage of Missing Data

	x
HitByPitch_AtBat	91.61
OutStealingBases	33.92
DoublePlays	12.57
BasesStolen	5.76
StrikeOuts_AtBat	4.48
StrikeOuts	4.48
Wins	0.00
Hits	0.00
Doubles	0.00
Triples	0.00
HomeRuns	0.00
Walks_AtBat	0.00
Hits_Allowed	0.00
HomeRuns_Allowed	0.00
Walks_Allowed	0.00
Errors	0.00

Errors, Triples and Walks_AtBat. There are other variables with what look like bi-modal type of distributions. For example: StrikeOuts_AtBat. There are a couple of variables that look closer to the normal distribution. For example - the response variable Wins.

Data Preparation

Missing Data - Handling NA values

We now dig deeper into the extent of missing data for the predictors.

91.61% percent of the rows are missing from the HitByPitch_AtBat variable, so we will remove this variable from the dataset completely. The percentage of missing data for the remaining variables with missing data is much less, and so excluding them from the final model could skew the results.

We now need to deal with 2 more data issues: 1) significant outliers 2) missing values

We could possibly drop rows with either of the 2 issues mentioned above, but then we may end up losing a fair amount of data. We therefore decided to remove the outliers for some of the more extreme cases, and then from the updated dataset, we imputed the missing values with the median of the respective predictor variable.

The following columns look like they have significant outliers: - Walks_Allowed - BasesStolen - StrikeOuts - Hits_Allowed - Errors - Triples

Handling extreme values

##	Wins	Hits	Doubles	Triples
##	Min. : 21.00	Min. :1137	Min. :130.0	Min. : 11.0
##	1st Qu.: 72.00	1st Qu.:1385	1st Qu.:215.0	1st Qu.: 32.0
##	Median : 82.00	Median :1447	Median :244.0	Median : 42.0
##	Mean : 80.77	Mean :1457	Mean :246.9	Mean : 48.1
##	3rd Qu.: 90.00	3rd Qu.:1524	3rd Qu.:276.0	3rd Qu.: 60.0
##	Max. :120.00	Max. :1876	Max. :392.0	Max. :126.0
##				
##	HomeRuns	Walks_AtBat	StrikeOuts_AtBat	BasesStolen

```

## Min. : 4.0 Min. :273.0 Min. : 268 Min. : 18.0
## 1st Qu.: 75.0 1st Qu.:472.0 1st Qu.: 598 1st Qu.: 62.0
## Median :118.0 Median :523.0 Median : 814 Median : 91.0
## Mean :115.6 Mean :527.9 Mean : 783 Mean :100.2
## 3rd Qu.:156.0 3rd Qu.:585.0 3rd Qu.: 955 3rd Qu.:131.0
## Max. :264.0 Max. :775.0 Max. :1399 Max. :289.0
##
## OutStealingBases Hits_Allowed HomeRuns_Allowed Walks_Allowed
## Min. : 11.00 Min. :1137 Min. : 4.0 Min. :320.0
## 1st Qu.: 38.00 1st Qu.:1407 1st Qu.: 79.0 1st Qu.:487.0
## Median : 49.00 Median :1490 Median :121.0 Median :537.0
## Mean : 52.82 Mean :1510 Mean :118.4 Mean :546.2
## 3rd Qu.: 62.00 3rd Qu.:1590 3rd Qu.:158.0 3rd Qu.:601.0
## Max. :201.00 Max. :2069 Max. :264.0 Max. :810.0
## NA's :356
## StrikeOuts Errors DoublePlays
## Min. : 301.0 Min. : 65.0 Min. : 72.0
## 1st Qu.: 639.0 1st Qu.:122.0 1st Qu.:136.0
## Median : 824.0 Median :144.0 Median :151.0
## Mean : 805.5 Mean :161.9 Mean :150.4
## 3rd Qu.: 962.0 3rd Qu.:184.0 3rd Qu.:165.0
## Max. :1481.0 Max. :430.0 Max. :225.0
## NA's :21
## Wins Hits Doubles Triples
## Min. : 21.00 Min. :1137 Min. :130.0 Min. : 11.0
## 1st Qu.: 72.00 1st Qu.:1385 1st Qu.:215.0 1st Qu.: 32.0
## Median : 82.00 Median :1447 Median :244.0 Median : 42.0
## Mean : 80.77 Mean :1457 Mean :246.9 Mean : 48.1
## 3rd Qu.: 90.00 3rd Qu.:1524 3rd Qu.:276.0 3rd Qu.: 60.0
## Max. :120.00 Max. :1876 Max. :392.0 Max. :126.0
## HomeRuns Walks_AtBat StrikeOuts_AtBat BasesStolen
## Min. : 4.0 Min. :273.0 Min. : 268 Min. : 18.0
## 1st Qu.: 75.0 1st Qu.:472.0 1st Qu.: 598 1st Qu.: 62.0
## Median :118.0 Median :523.0 Median : 814 Median : 91.0
## Mean :115.6 Mean :527.9 Mean : 783 Mean :100.2
## 3rd Qu.:156.0 3rd Qu.:585.0 3rd Qu.: 955 3rd Qu.:131.0
## Max. :264.0 Max. :775.0 Max. :1399 Max. :289.0
## OutStealingBases Hits_Allowed HomeRuns_Allowed Walks_Allowed
## Min. : 11.00 Min. :1137 Min. : 4.0 Min. :320.0
## 1st Qu.: 41.00 1st Qu.:1407 1st Qu.: 79.0 1st Qu.:487.0
## Median : 49.00 Median :1490 Median :121.0 Median :537.0
## Mean : 52.06 Mean :1510 Mean :118.4 Mean :546.2
## 3rd Qu.: 58.00 3rd Qu.:1590 3rd Qu.:158.0 3rd Qu.:601.0
## Max. :201.00 Max. :2069 Max. :264.0 Max. :810.0
## StrikeOuts Errors DoublePlays
## Min. : 301.0 Min. : 65.0 Min. : 72.0
## 1st Qu.: 639.0 1st Qu.:122.0 1st Qu.:136.0
## Median : 824.0 Median :144.0 Median :151.0
## Mean : 805.5 Mean :161.9 Mean :150.4
## 3rd Qu.: 962.0 3rd Qu.:184.0 3rd Qu.:165.0
## Max. :1481.0 Max. :430.0 Max. :225.0

```

Table 2: Correlation of Variables to Wins

	x
Hits	0.3505170
Doubles	0.2177959
Triples	0.1065282
HomeRuns	0.2269131
Walks_AtBat	0.3031098
StrikeOuts_AtBat	-0.0435931
BasesStolen	0.1064962
OutStealingBases	-0.0072786
Hits_Allowed	0.2476797
HomeRuns_Allowed	0.2286214
Walks_Allowed	0.2883711
StrikeOuts	-0.0563730
Errors	-0.1921452
DoublePlays	-0.0336770

Correlation between predictors, and between predictors and response variable

We now check which of the predictors are more correlated with the response variable as a mechanism to select which variables to include in the linear regression model. We also check the correlation between the predictors, since we'd like to avoid multi-collinearity.

	Wins	Hits	Doubles	Triples	HomeRuns	Walks_AtBat	StrikeOuts_AtBat	BasesStolen
Wins	1.00	0.35	0.22	0.11	0.23	0.30	-0.04	0.11
Hits	0.35	1.00	0.65	0.38	0.09	0.09	-0.37	-0.05
Doubles	0.22	0.65	1.00	-0.06	0.38	0.24	0.13	-0.13
Triples	0.11	0.38	-0.06	1.00	-0.65	-0.25	-0.72	0.20
HomeRuns	0.23	0.09	0.38	-0.65	1.00	0.40	0.71	-0.31
Walks_AtBat	0.30	0.09	0.24	-0.25	0.40	1.00	0.21	-0.15
StrikeOuts_AtBat	-0.04	-0.37	0.13	-0.72	0.71	0.21	1.00	-0.03
BasesStolen	0.11	-0.05	-0.13	0.26	-0.31	-0.15	-0.03	1.00
OutStealingBases	-0.01	0.00	-0.09	0.25	-0.34	-0.14	-0.18	0.40
Hits_Allowed	0.25	0.83	0.42	0.55	-0.18	-0.04	-0.56	0.05
HomeRuns_Allowed	0.23	0.11	0.38	-0.63	0.99	0.41	0.68	-0.32
Walks_Allowed	0.29	0.12	0.19	-0.11	0.26	0.95	0.05	-0.11
StrikeOuts	-0.06	-0.38	0.11	-0.69	0.68	0.20	0.98	-0.01
Errors	-0.19	0.11	-0.30	0.69	-0.71	-0.35	-0.65	0.30
DoublePlays	-0.03	0.17	0.18	-0.20	0.29	0.24	0.06	-0.39

##	row	column	cor	p
## 1	Wins	Hits	0.350517030	0.0000000000000000000000
## 2	Wins	Walks_AtBat	0.303109751	0.0000000000000000000000
## 3	Wins	Walks_Allowed	0.288371147	0.0000000000000000000000
## 4	Wins	Hits_Allowed	0.247679739	0.0000000000000000000000
## 5	Wins	HomeRuns_Allowed	0.228621353	0.0000000000000000000000
## 6	Wins	HomeRuns	0.226913134	0.0000000000000000000000
## 7	Wins	Doubles	0.217795857	0.0000000000000000000000
## 8	Wins	Errors	-0.192145179	0.00000000000000002220446
## 9	Wins	Triples	0.106528221	0.0000063172528612653878
## 10	Wins	BasesStolen	0.106496161	0.0000063581242804033877

Based on the p-values, we could exclude the following variables from the regression model: StrikeOuts_AtBat, DoublePlays and OutStealingBases

[illegible]

Model 1

Model 1 Statistics

15

```
## Walks_AtBat      0.200978    0.064606    3.111                0.00190 **
## BasesStolen      0.076969    0.006573   11.709 < 0.0000000000000002 ***
## Hits_Allowed     0.065127    0.020521    3.174                0.00153 **
## HomeRuns_Allowed -0.145831    0.134764   -1.082                0.27935
## Errors           -0.124236    0.007365  -16.869 < 0.0000000000000002 ***
## Walks_Allowed    -0.159245    0.061932   -2.571                0.01021 *
## StrikeOuts       0.001674    0.032200    0.052                0.95854
## StrikeOuts_AtBat -0.023747    0.033436   -0.710                0.47765
## OutStealingBases -0.039104    0.014502   -2.696                0.00707 **
## DoublePlays      -0.109783    0.012606   -8.709 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.26 on 1774 degrees of freedom
## Multiple R-squared:  0.4045, Adjusted R-squared:  0.3998
## F-statistic: 86.07 on 14 and 1774 DF, p-value: < 0.00000000000000022
```

We see that the adjusted R-squared for this model is 0.40 i.e. these predictors explain about 40% of the variability in the response variable.

Model 1 R Squared

```
## [1] 0.4044997
```

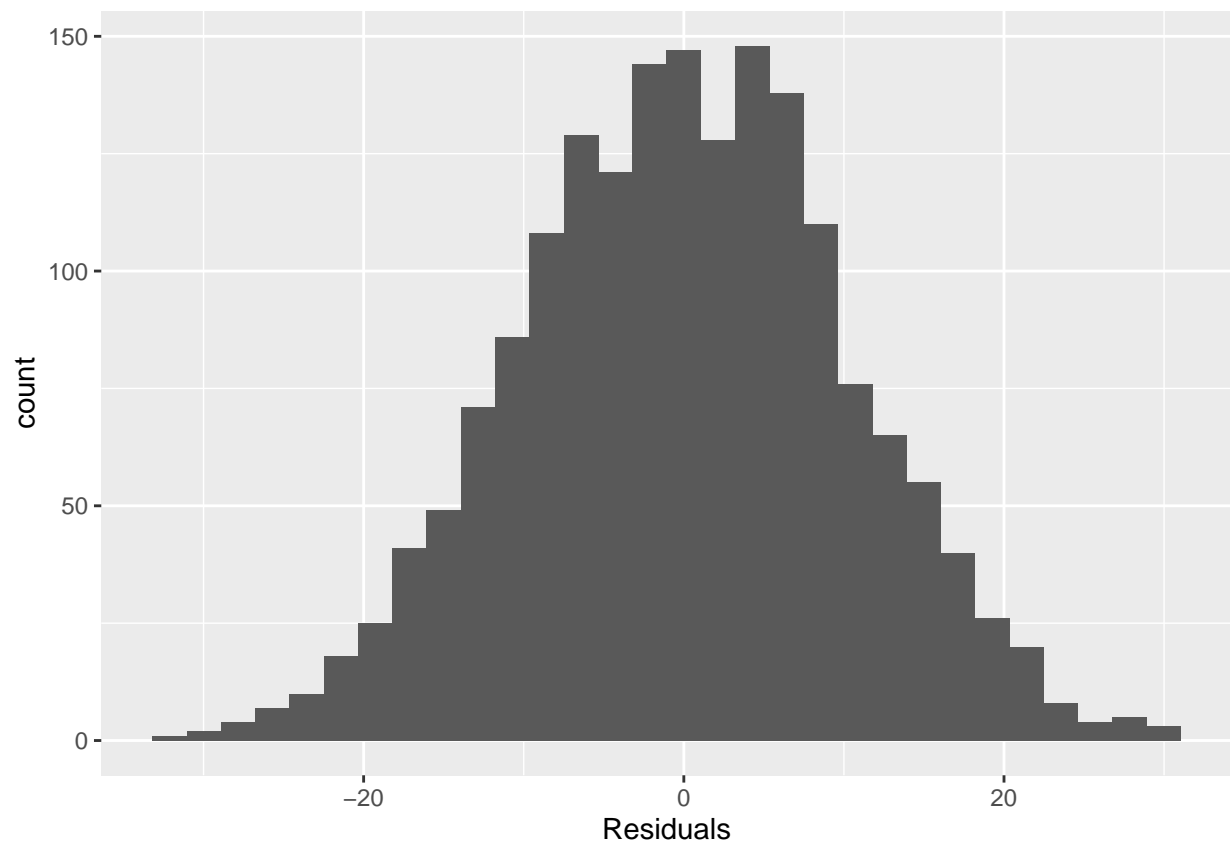
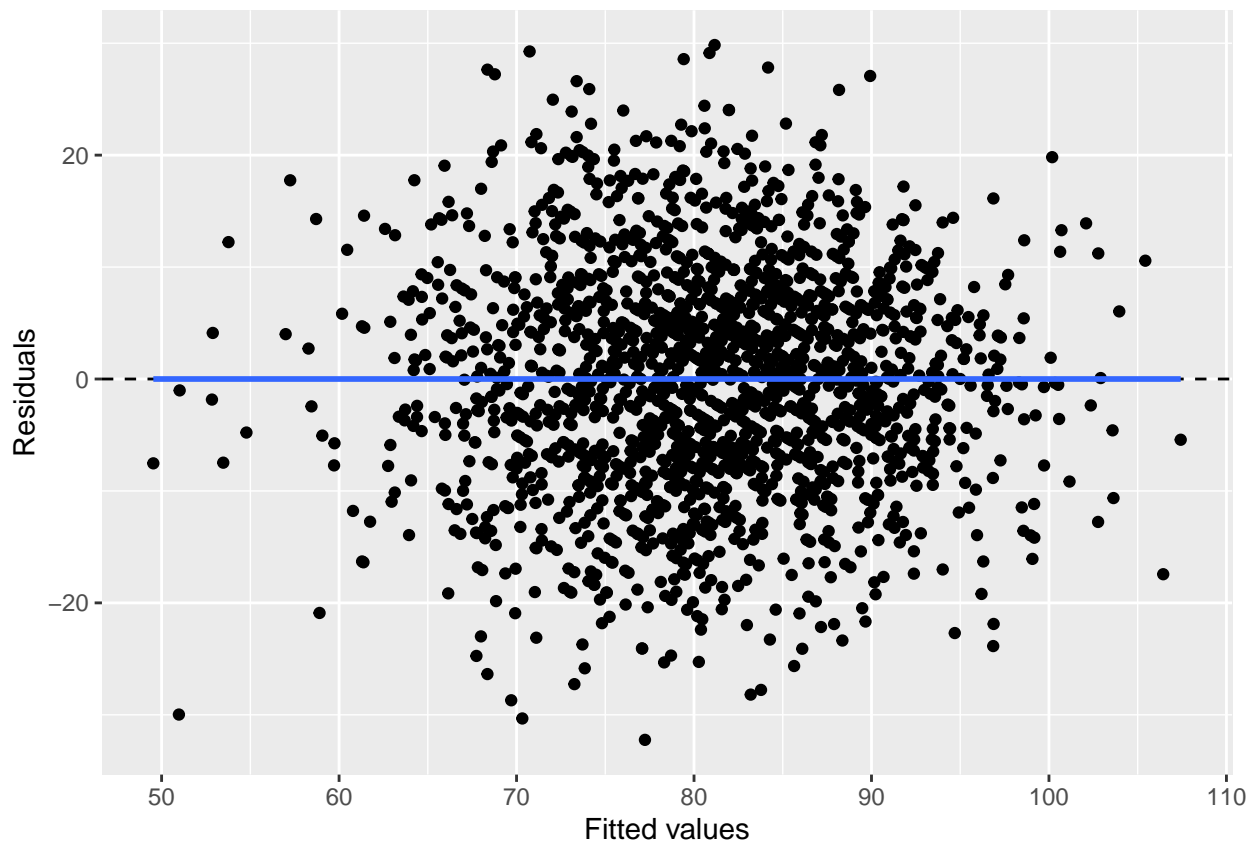
Model 1 Confidence Intervals

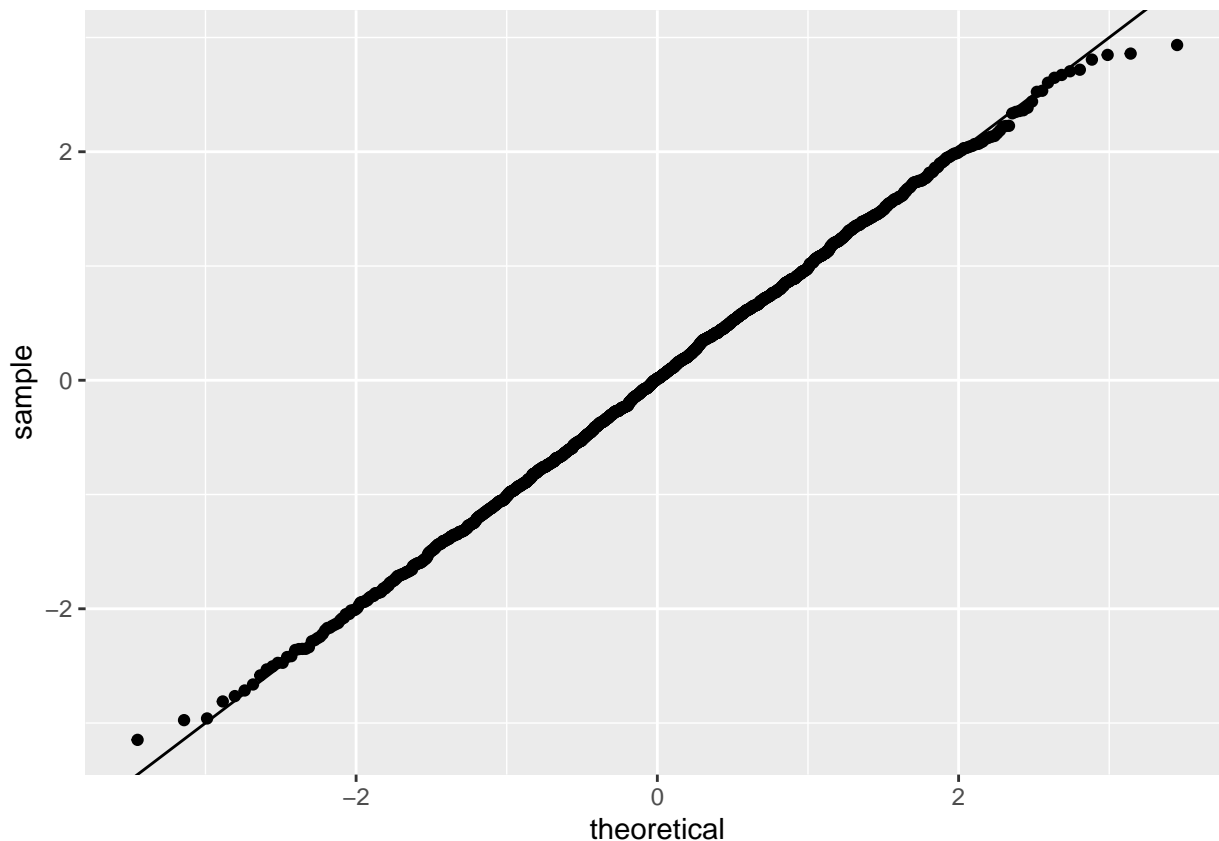
We calculate the 95% confidence intervals for each of the co-efficients and the intercept for this model.

```
##              2.5 %      97.5 %
## (Intercept)  44.94044362 69.127482832
## Hits        -0.07886950  0.007872103
## Doubles      -0.07225451 -0.036849474
## Triples      0.14798073  0.225304552
## HomeRuns     -0.03079800  0.513988472
## Walks_AtBat  0.07426677  0.327689717
## BasesStolen  0.06407675  0.089861282
## Hits_Allowed 0.02487963  0.105374808
## HomeRuns_Allowed -0.41014380 0.118482229
## Errors       -0.13868039 -0.109790758
## Walks_Allowed -0.28071180 -0.037777706
## StrikeOuts   -0.06148025  0.064828416
## StrikeOuts_AtBat -0.08932553 0.041830860
## OutStealingBases -0.06754688 -0.010661147
## DoublePlays  -0.13450771 -0.085058563
```

Model 1 Plots

We plot the residuals versus the fitted values - it shows that the residuals are scattered fairly evenly and there doesn't seem to be a trend. The distribution of the residuals does not seem very skewed. The same can be seen through the qq-plot as well.





Model 2

Model 2 uses stepwise regression on the variables in Model 1 to create the best performing model.

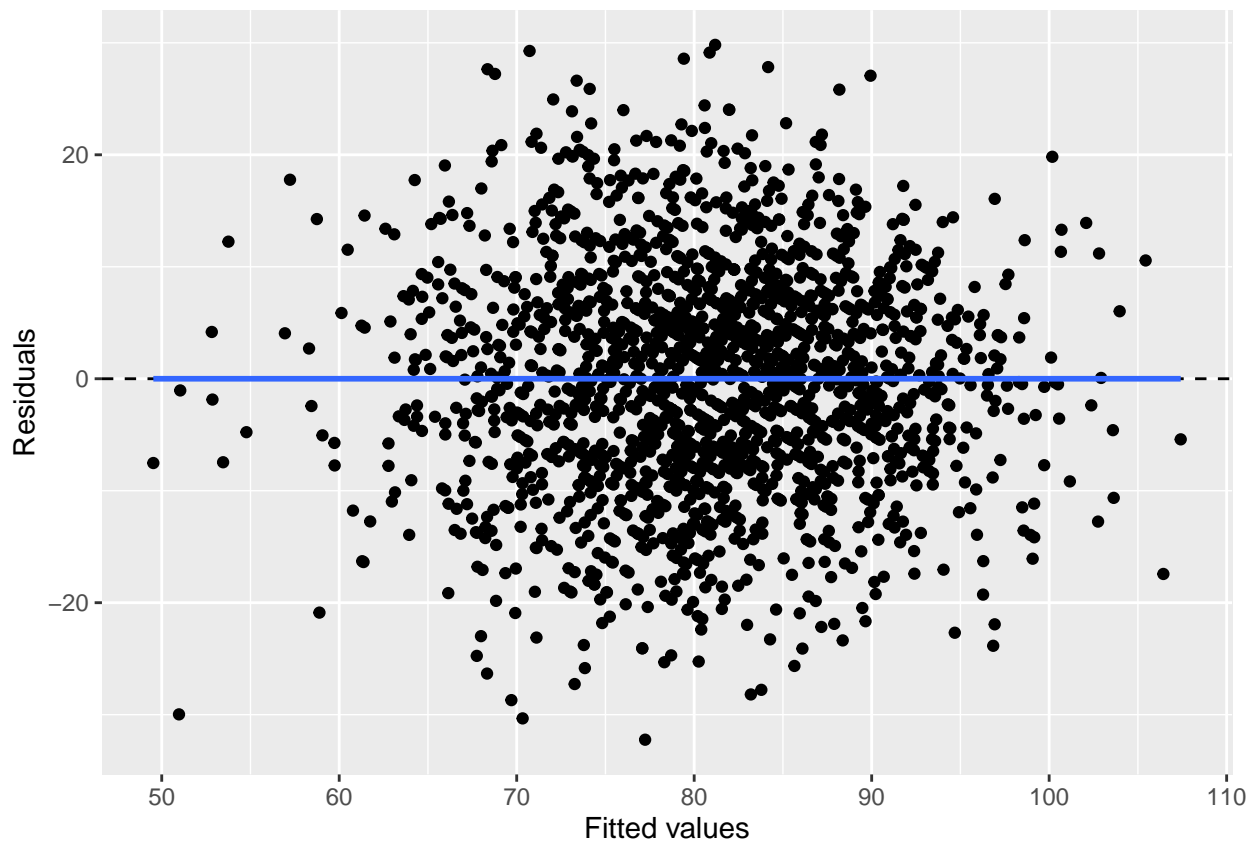
Model 2 Summary Stats

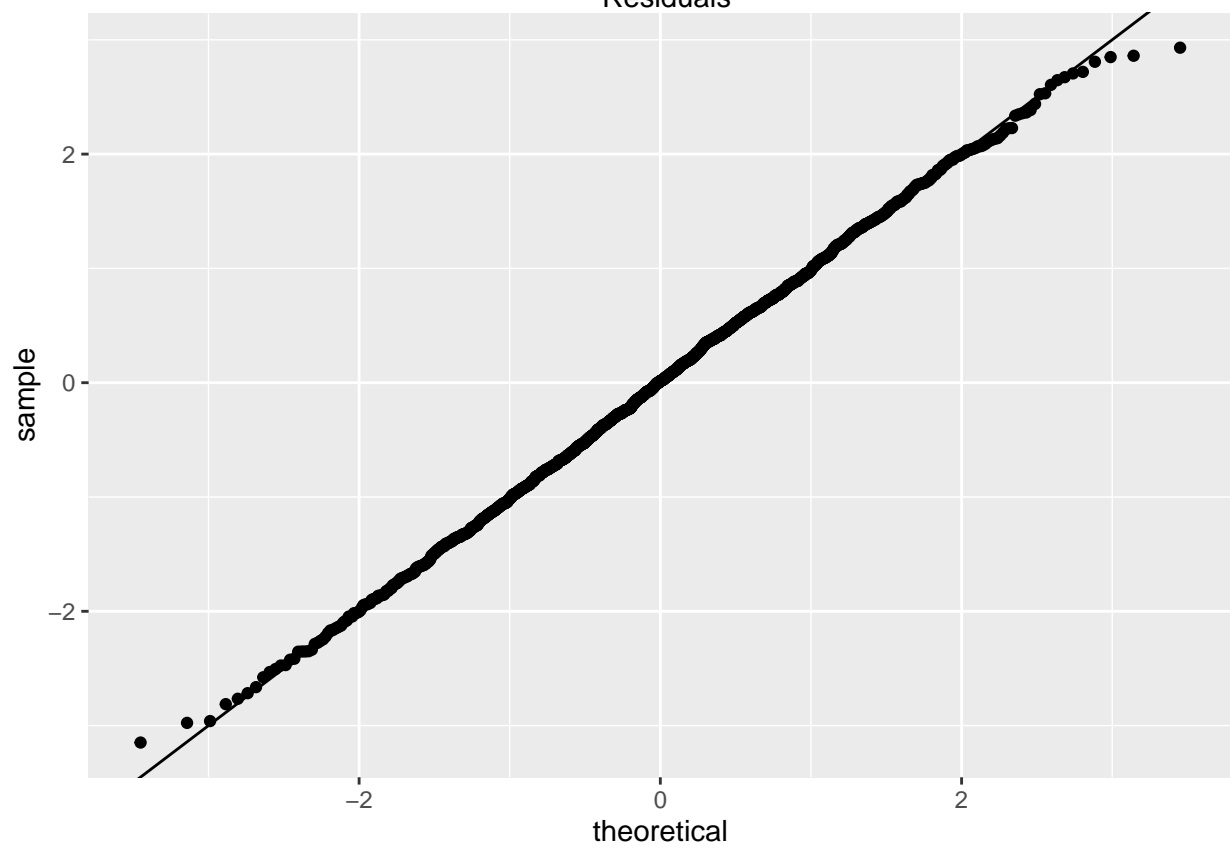
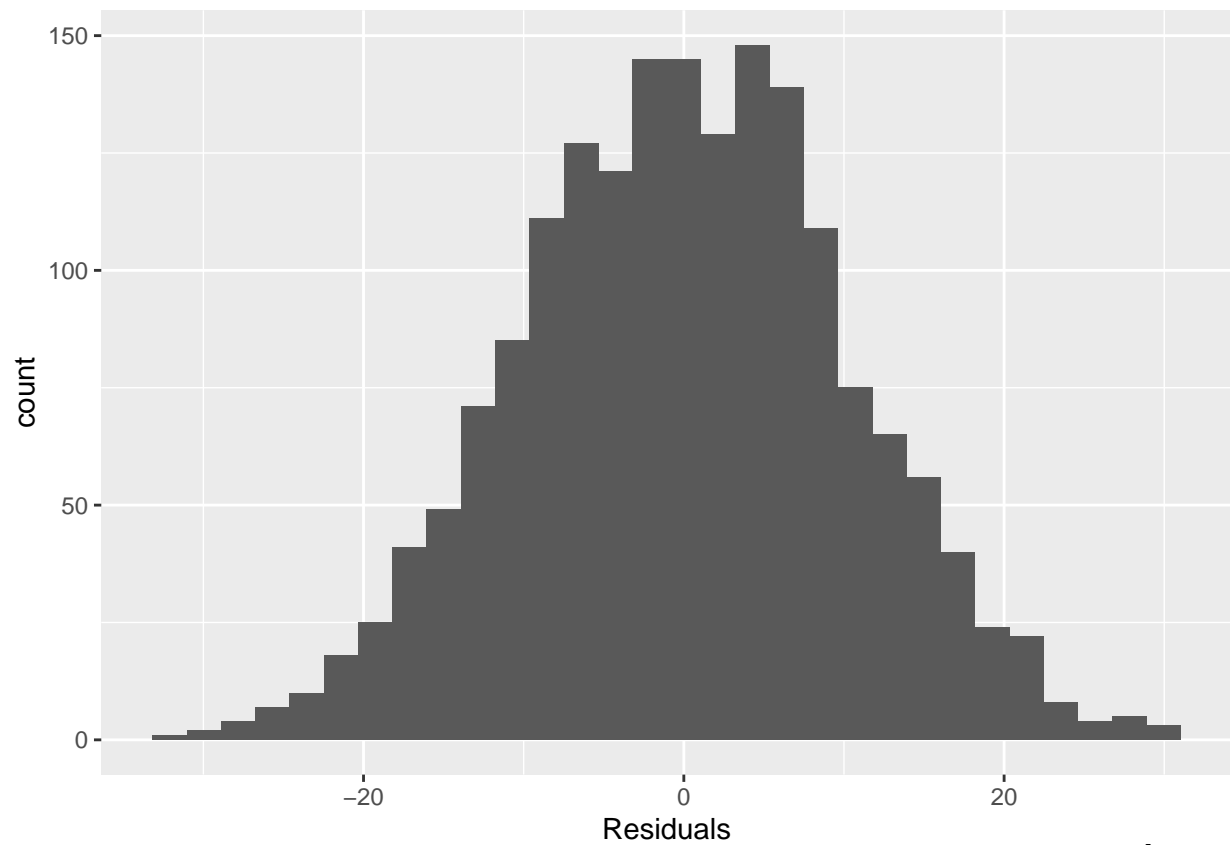
```
##
## Call:
## lm(formula = Wins ~ Hits + Doubles + Triples + HomeRuns + Walks_AtBat +
##      BasesStolen + Hits_Allowed + HomeRuns_Allowed + Errors +
##      Walks_Allowed + StrikeOuts_AtBat + OutStealingBases + DoublePlays,
##      data = mb_training_updated)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -32.235  -7.017   0.138   6.908  29.818
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  57.054236   6.151995   9.274 < 0.0000000000000002 ***
## Hits        -0.035845   0.021081  -1.700   0.089244 .
## Doubles     -0.054553   0.009023  -6.046   0.00000000181 ***
## Triples      0.186556   0.019637   9.500 < 0.0000000000000002 ***
## HomeRuns     0.236397   0.096373   2.453   0.014264 *
## Walks_AtBat  0.200201   0.062835   3.186   0.001467 **
## BasesStolen  0.076979   0.006568  11.720 < 0.0000000000000002 ***
## Hits_Allowed  0.065450   0.019551   3.348   0.000832 ***
## HomeRuns_Allowed -0.140764  0.093058  -1.513   0.130547
```

```
## Errors      -0.124249   0.007359 -16.885 < 0.0000000000000002 ***
## Walks_Allowed -0.158502   0.060243  -2.631      0.008586 **
## StrikeOuts_AtBat -0.022013   0.002395  -9.192 < 0.0000000000000002 ***
## OutStealingBases -0.039065   0.014479  -2.698      0.007039 **
## DoublePlays   -0.109815   0.012588  -8.724 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.26 on 1775 degrees of freedom
## Multiple R-squared:  0.4045, Adjusted R-squared:  0.4001
## F-statistic: 92.74 on 13 and 1775 DF,  p-value: < 0.00000000000000022
```

However we see minimal impact to the R-squared value, which remains around 0.40.

Model 2 Plots





Wins = Target_wins, Hits = Batting_h, Doubles = Batting_2b, Triples = Batting_3b, HomeRuns =

Batting_hr, Walks_AtBat = Batting_bb, StrikeOuts_AtBat = Batting_so, BasesStolen = Baserun_sb, OutStealingBases = Baserun_cs, Hits_Allowed = Pitching_h, HitByPitch_AtBat = Batting_hbp, Errors = Fielding_e, HomeRuns_Allowed = Pitching_hr, Walks_Allowed = Pitching_bb, StrikeOuts = Pitching_so, DoublePlays = Fielding_dp

Model 3

For Model 3, we create a new dataframe and derive some new variables by transforming existing predictors to include in this dataframe: - Singles is derived as the difference between all Hits and Doubles, Triples and Home Runs - Homeruns difference is the difference between home runs scored and allowed.

We also include certain variables derived on the fly in the model - for example: the ratio between Home runs allowed and scores, the product of home runs allowed and scored, the reciprocal of Double plays and the cube of the stolen basis variable.

Model 3 Summary Stats

```
##
## Call:
## lm(formula = Wins ~ Hits + Doubles + Triples + Walks_AtBat +
##      BasesStolen + Hits_Allowed + Errors + Walks_Allowed + StrikeOuts +
##      Singles + Homeruns_diff + StrikeOuts_AtBat + I(HomeRuns_Allowed/HomeRuns) +
##      I(HomeRuns_Allowed * HomeRuns) + I(1/DoublePlays) + I(OutStealingBases^3),
##      data = mb_training_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31.826  -7.049   0.066   6.960  31.025
##
## Coefficients:
##              Estimate      Std. Error t value
## (Intercept)    109.9453169993    40.2856609726   2.729
## Hits           0.0281580282     0.0440891477   0.639
## Doubles        -0.1735673380     0.0277600015  -6.252
## Triples         0.0602076781     0.0303537212   1.984
## Walks_AtBat     0.1904325935     0.0648354833   2.937
## BasesStolen     0.0687665386     0.0060907199  11.290
## Hits_Allowed    0.1198392065     0.0310629804   3.858
## Errors         -0.1235278847     0.0075827849 -16.291
## Walks_Allowed  -0.1489662240     0.0621728192  -2.396
## StrikeOuts      0.0166499515     0.0340842858   0.488
## Singles        -0.1206931596     0.0272655274  -4.427
## Homeruns_diff  -0.2229976419     0.1427124139  -1.563
## StrikeOuts_AtBat -0.0391231155     0.0354323522  -1.104
## I(HomeRuns_Allowed/HomeRuns) -85.2314174131    37.8641514196  -2.251
## I(HomeRuns_Allowed * HomeRuns) -0.0000688546     0.0000877940  -0.784
## I(1/DoublePlays) 2390.0964728930    257.2489261894   9.291
## I(OutStealingBases^3) 0.0000001113     0.0000004862   0.229
##              Pr(>|t|)
## (Intercept)      0.006413 **
## Hits             0.523126
## Doubles          0.000000000505 ***
## Triples          0.047462 *
## Walks_AtBat      0.003355 **
## BasesStolen      < 0.0000000000000002 ***
## Hits_Allowed     0.000118 ***
```

```
## Errors < 0.0000000000000002 ***
## Walks_Allowed 0.016678 *
## StrikeOuts 0.625261
## Singles 0.000010157283 ***
## Homeruns_diff 0.118333
## StrikeOuts_AtBat 0.269672
## I(Homeruns_Allowed/HomeRuns) 0.024509 *
## I(Homeruns_Allowed * HomeRuns) 0.432984
## I(1/DoublePlays) < 0.0000000000000002 ***
## I(OutStealingBases^3) 0.819000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.25 on 1772 degrees of freedom
## Multiple R-squared:  0.4069, Adjusted R-squared:  0.4015
## F-statistic: 75.97 on 16 and 1772 DF, p-value: < 0.0000000000000002
```

We don't see much change to the R-squared value.

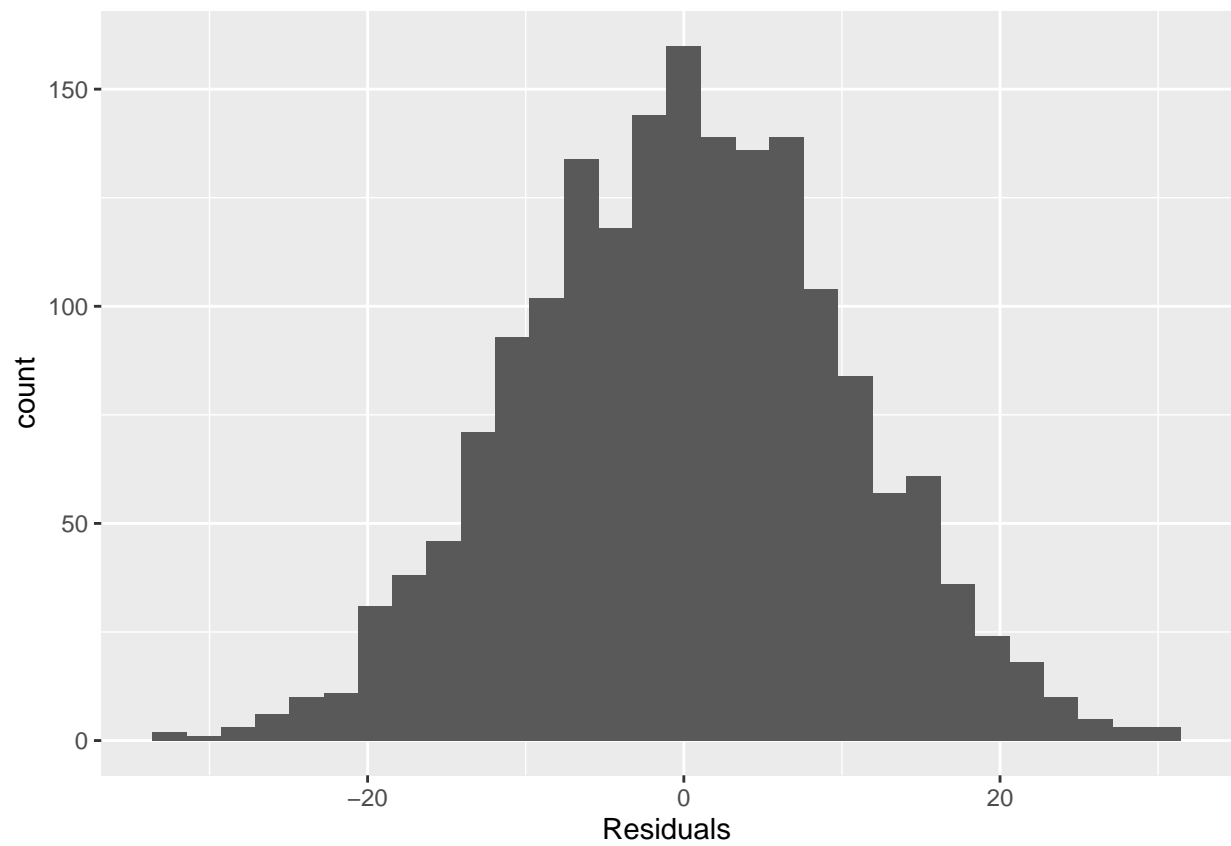
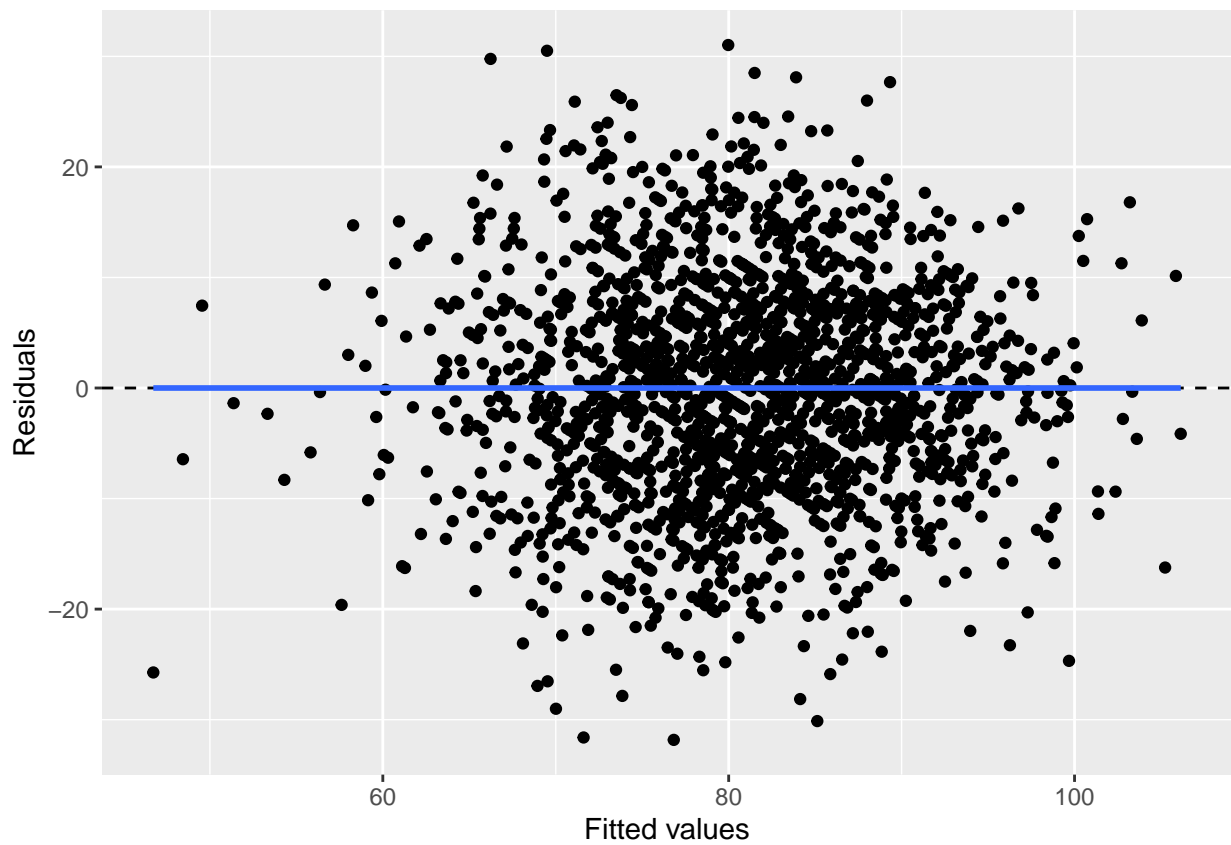
Model 3 R-Squared

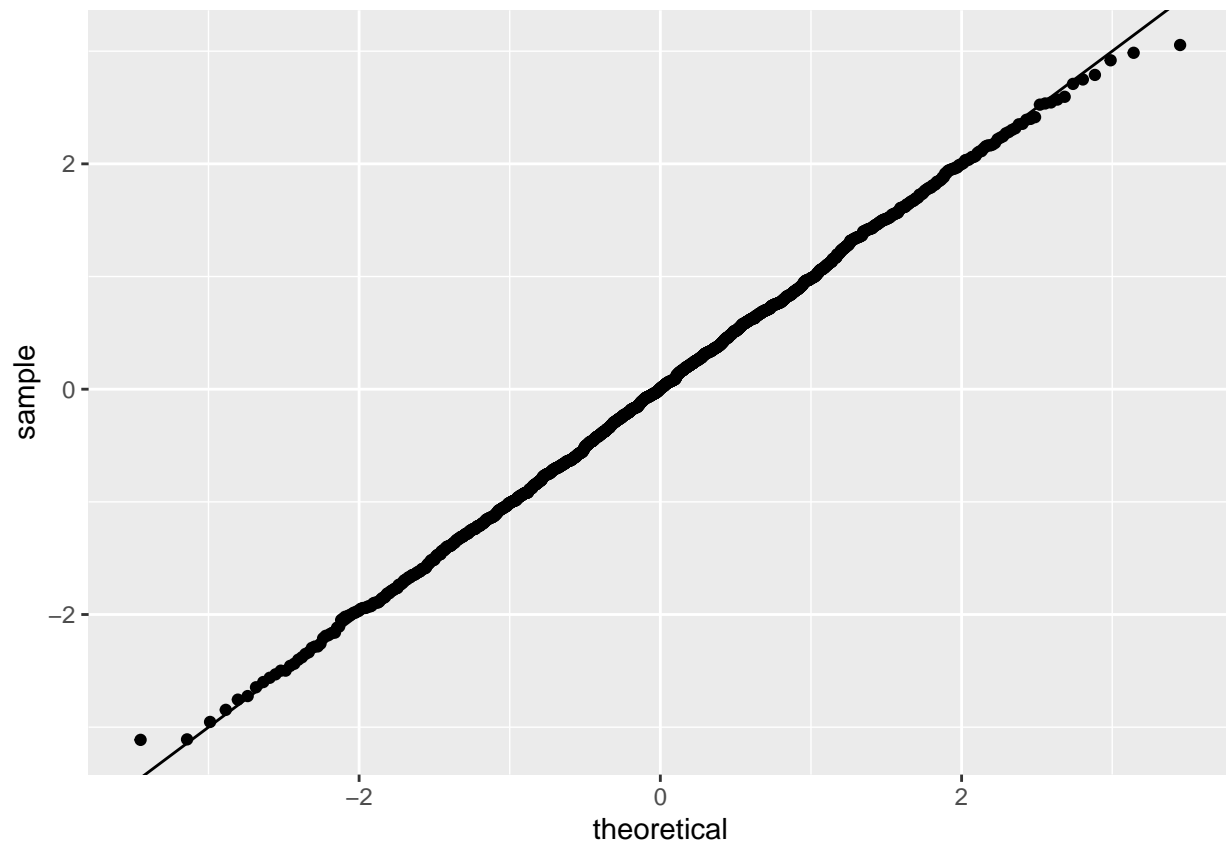
```
## [1] 0.4068615
```

Model 3 Confidence Intervals

	2.5 %	97.5 %
## (Intercept)	30.9329035984562	188.957730400225
## Hits	-0.0583141776004	0.114630233945
## Doubles	-0.2280131298677	-0.119121546135
## Triples	0.0006748143466	0.119740541945
## Walks_AtBat	0.0632705241893	0.317594662719
## BasesStolen	0.0568207875157	0.080712289602
## Hits_Allowed	0.0589152700712	0.180763142896
## Errors	-0.1384000282943	-0.108655741084
## Walks_Allowed	-0.2709060004136	-0.027026447629
## StrikeOuts	-0.0501996822336	0.083499585329
## Singles	-0.1741691375868	-0.067217181579
## Homeruns_diff	-0.5029000183358	0.056904734612
## StrikeOuts_AtBat	-0.1086167168480	0.030370485818
## I(Homeruns_Allowed/HomeRuns)	-159.4945153196878	-10.968319506518
## I(Homeruns_Allowed * HomeRuns)	-0.0002410453595	0.000103336128
## I(1/DoublePlays)	1885.5532182390721	2894.639727546878
## I(OutStealingBases^3)	-0.000008422653	0.000001064801

Model 3 Plots

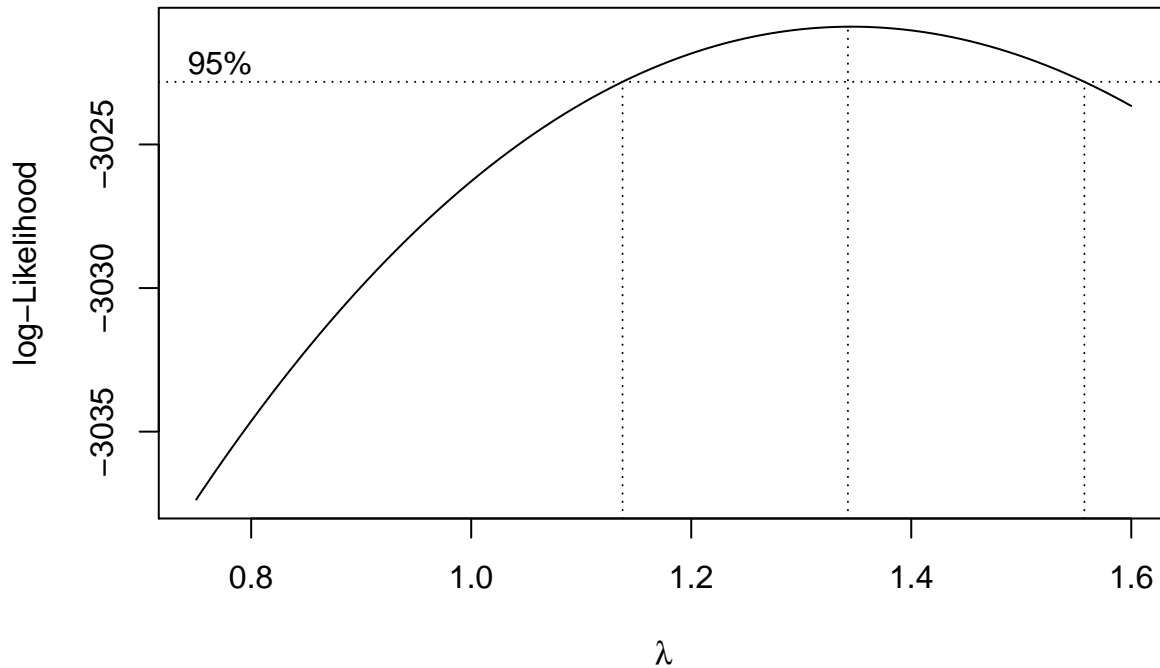




There is not much change in the scatter plot of the residuals with the fitted values, and the distribution of errors does not seem to have changed much.

Model 4 - Box Cox transformation

For our final model (Model 4), we do a Box Cox transformation on the response variable from Model 1 to see if it provides a better-fitting model. We plot the lambda and based on the plot, a lambda value of around 1.35 seems like the best value.



Model 4 Statistics

Model 4 Summary Stats

```
##
## Call:
## lm(formula = (((Wins^1.35) - 1)/1.35) ~ Hits + Doubles + Triples +
##      HomeRuns + Walks_AtBat + BasesStolen + Hits_Allowed + HomeRuns_Allowed +
##      Errors + Walks_Allowed + StrikeOuts + StrikeOuts_AtBat +
##      OutStealingBases + DoublePlays, data = mb_training_updated)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-138.325	-33.262	-0.183	31.243	143.981

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	169.730793	28.446990	5.967	0.00000000292	***
Hits	-0.160624	0.102019	-1.574	0.11556	
Doubles	-0.251500	0.041641	-6.040	0.00000000188	***
Triples	0.859596	0.090943	9.452	< 0.0000000000000002	***
HomeRuns	1.088291	0.640737	1.698	0.08959	.
Walks_AtBat	0.922916	0.298057	3.096	0.00199	**
BasesStolen	0.351828	0.030326	11.602	< 0.0000000000000002	***
Hits_Allowed	0.296026	0.094672	3.127	0.00180	**
HomeRuns_Allowed	-0.640745	0.621730	-1.031	0.30288	
Errors	-0.560633	0.033978	-16.500	< 0.0000000000000002	***
Walks_Allowed	-0.729399	0.285721	-2.553	0.01077	*
StrikeOuts	0.007101	0.148555	0.048	0.96188	
StrikeOuts_AtBat	-0.110581	0.154256	-0.717	0.47355	
OutStealingBases	-0.183480	0.066905	-2.742	0.00616	**
DoublePlays	-0.502375	0.058158	-8.638	< 0.0000000000000002	***

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.33 on 1774 degrees of freedom
## Multiple R-squared:  0.4028, Adjusted R-squared:  0.3981
## F-statistic: 85.46 on 14 and 1774 DF,  p-value: < 0.00000000000000022
```

Model 4 R Squared

```
## [1] 0.4027743
```

We don't see much impact on R-squared, possibly because the response variable was close to normal to begin with.

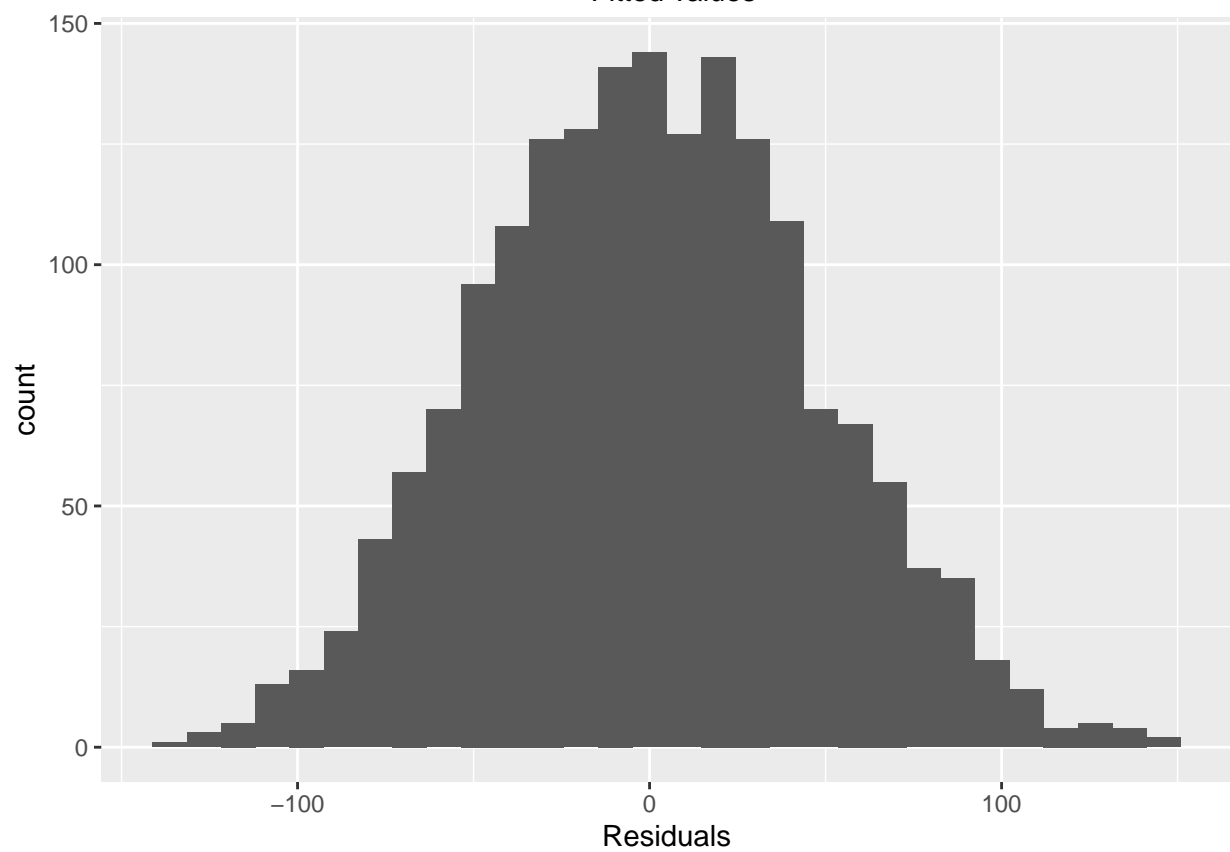
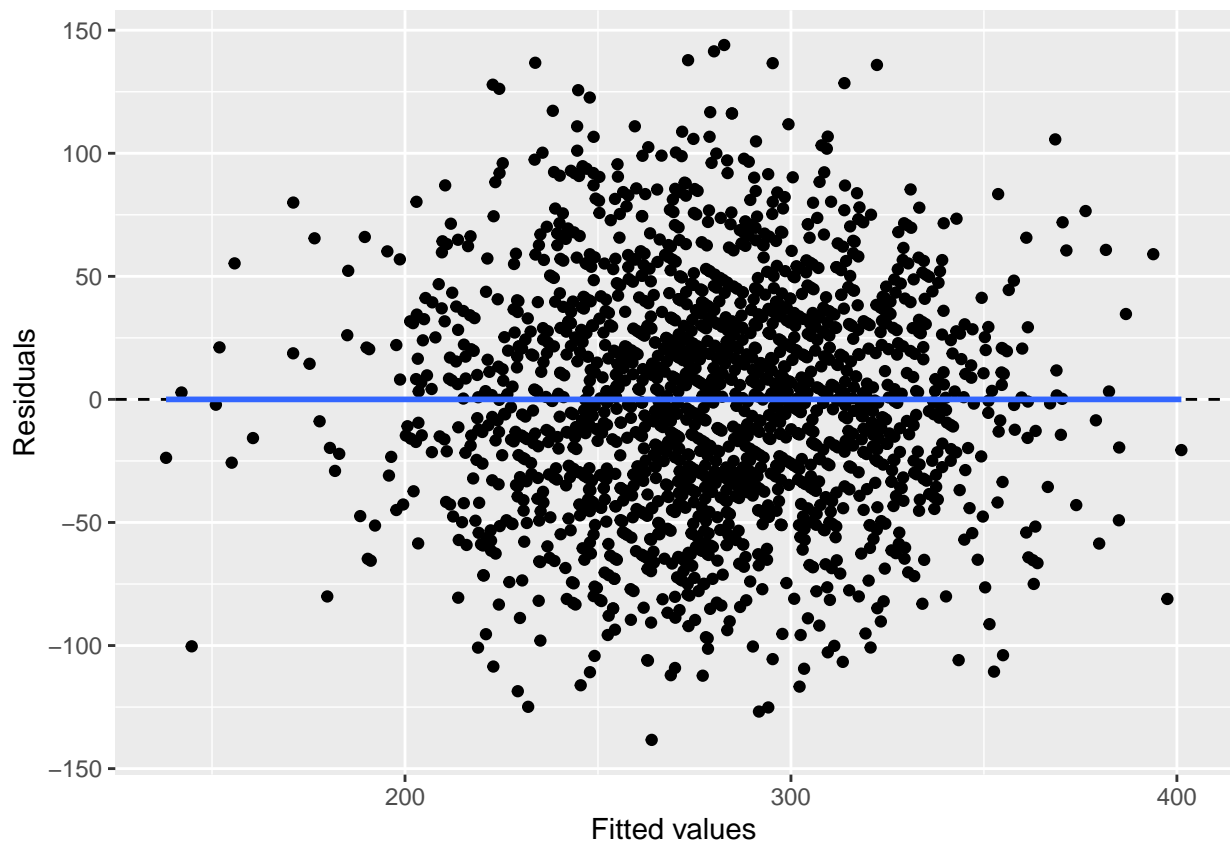
Model 4 Confidence Intervals

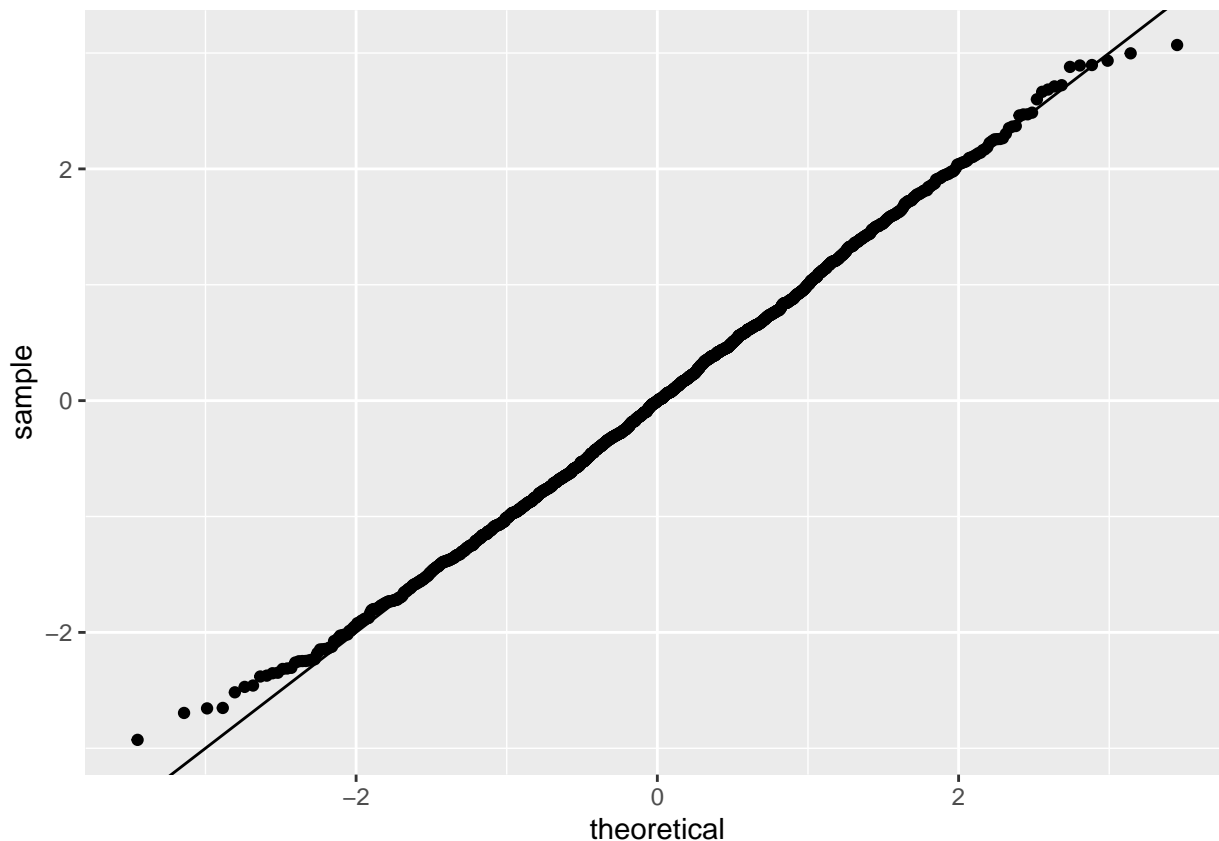
We calculate the 95% confidence intervals for each of the co-efficients and the intercept for this model.

##	2.5 %	97.5 %
## (Intercept)	113.9376512	225.52393496
## Hits	-0.3607137	0.03946645
## Doubles	-0.3331698	-0.16982958
## Triples	0.6812301	1.03796157
## HomeRuns	-0.1683880	2.34497038
## Walks_AtBat	0.3383354	1.50749568
## BasesStolen	0.2923498	0.41130605
## Hits_Allowed	0.1103447	0.48170712
## HomeRuns_Allowed	-1.8601466	0.57865598
## Errors	-0.6272741	-0.49399258
## Walks_Allowed	-1.2897845	-0.16901436
## StrikeOuts	-0.2842602	0.29846157
## StrikeOuts_AtBat	-0.4131244	0.19196222
## OutStealingBases	-0.3147009	-0.05225998
## DoublePlays	-0.6164411	-0.38830872

Model 4 Plots

We plot the residuals versus the fitted values - it shows that the residuals are scattered fairly evenly and there doesn't seem to be a trend. The distribution of the residuals does not seem very skewed. The same can be seen through the qq-plot as well.





The residuals for this model behave similarly to the residuals from the previous model.

Model Selection

We decide to use model one for making predictions for the test dataset, since the other models do not provide a significant improvement over it.

Predicting the response variable for the test dataset

We now predict the number of wins for the test data using model one.

```
## TEAM_BATTING_H TEAM_BATTING_2B TEAM_BATTING_3B TEAM_BATTING_HR
## Min. : 819 Min. : 44.0 Min. : 14.00 Min. : 0.00
## 1st Qu.:1387 1st Qu.:210.0 1st Qu.: 35.00 1st Qu.: 44.50
## Median :1455 Median :239.0 Median : 52.00 Median :101.00
## Mean :1469 Mean :241.3 Mean : 55.91 Mean : 95.63
## 3rd Qu.:1548 3rd Qu.:278.5 3rd Qu.: 72.00 3rd Qu.:135.50
## Max. :2170 Max. :376.0 Max. :155.00 Max. :242.00
##
## TEAM_BATTING_BB TEAM_BATTING_SO TEAM_BASERUN_SB TEAM_BASERUN_CS
## Min. : 15.0 Min. : 0.0 Min. : 0.0 Min. : 0.00
## 1st Qu.:436.5 1st Qu.: 545.0 1st Qu.: 59.0 1st Qu.: 38.00
## Median :509.0 Median : 686.0 Median : 92.0 Median : 49.50
## Mean :499.0 Mean : 709.3 Mean :123.7 Mean : 52.32
## 3rd Qu.:565.5 3rd Qu.: 912.0 3rd Qu.:151.8 3rd Qu.: 63.00
## Max. :792.0 Max. :1268.0 Max. :580.0 Max. :154.00
## NA's :18 NA's :13 NA's :87
## TEAM_BATTING_HBP TEAM_PITCHING_H TEAM_PITCHING_HR TEAM_PITCHING_BB
## Min. :42.00 Min. : 1155 Min. : 0.0 Min. : 136.0
```

```
## 1st Qu.:53.50    1st Qu.: 1426    1st Qu.: 52.0    1st Qu.: 471.0
## Median :62.00    Median : 1515    Median :104.0    Median : 526.0
## Mean   :62.37    Mean   : 1813    Mean   :102.1    Mean   : 552.4
## 3rd Qu.:67.50    3rd Qu.: 1681    3rd Qu.:142.5    3rd Qu.: 606.5
## Max.   :96.00    Max.   :22768    Max.   :336.0    Max.   :2008.0
## NA's   :240
## TEAM_PITCHING_SO TEAM_FIELDING_E TEAM_FIELDING_DP
## Min.    : 0.0    Min.    : 73.0    Min.    : 69.0
## 1st Qu.: 613.0    1st Qu.: 131.0    1st Qu.:131.0
## Median : 745.0    Median : 163.0    Median :148.0
## Mean   : 799.7    Mean   : 249.7    Mean   :146.1
## 3rd Qu.: 938.0    3rd Qu.: 252.0    3rd Qu.:164.0
## Max.   :9963.0    Max.   :1568.0    Max.   :204.0
## NA's   :18      NA's   :31
```

Rename Columns

We now rename the columns for this test dataset to be more intuitive.

We impute the missing values with the median

We now check the summary to see if the missing values have been filled.

```
##      Hits      Doubles      Triples      HomeRuns
## Min.   : 819    Min.   : 44.0    Min.   : 14.00   Min.   : 0.00
## 1st Qu.:1387    1st Qu.:210.0    1st Qu.: 35.00   1st Qu.: 44.50
## Median :1455    Median :239.0    Median : 52.00   Median :101.00
## Mean   :1469    Mean   :241.3    Mean   : 55.91   Mean   : 95.63
## 3rd Qu.:1548    3rd Qu.:278.5    3rd Qu.: 72.00   3rd Qu.:135.50
## Max.   :2170    Max.   :376.0    Max.   :155.00   Max.   :242.00
## Walks_AtBat  StrikeOuts_AtBat  BasesStolen  OutStealingBases
## Min.   : 15.0    Min.   : 0.0    Min.   : 0.0    Min.   : 0.00
## 1st Qu.:436.5    1st Qu.: 565.0    1st Qu.: 60.5    1st Qu.: 44.00
## Median :509.0    Median : 686.0    Median : 92.0    Median : 49.50
## Mean   :499.0    Mean   : 707.7    Mean   :122.1    Mean   : 51.37
## 3rd Qu.:565.5    3rd Qu.: 904.5    3rd Qu.:149.0    3rd Qu.: 56.00
## Max.   :792.0    Max.   :1268.0    Max.   :580.0    Max.   :154.00
## HitByPitch_AtBat  Hits_Allowed  HomeRuns_Allowed  Walks_Allowed
## Min.   :42.00    Min.   :1155    Min.   : 0.0    Min.   :136.0
## 1st Qu.:62.00    1st Qu.:1426    1st Qu.: 52.0    1st Qu.:471.0
## Median :62.00    Median :1515    Median :104.0    Median :526.0
## Mean   :62.03    Mean   :1813    Mean   :102.1    Mean   :552.4
## 3rd Qu.:62.00    3rd Qu.:1681    3rd Qu.:142.5    3rd Qu.:606.5
## Max.   :96.00    Max.   :22768    Max.   :336.0    Max.   :2008.0
## StrikeOuts      Errors      DoublePlays
## Min.   : 0.0    Min.   : 73.0    Min.   : 69.0
## 1st Qu.: 622.5    1st Qu.:131.0    1st Qu.:134.5
## Median : 745.0    Median :163.0    Median :148.0
## Mean   : 795.9    Mean   :249.7    Mean   :146.3
## 3rd Qu.: 927.5    3rd Qu.:252.0    3rd Qu.:160.5
## Max.   :9963.0    Max.   :1568.0    Max.   :204.0
```

We create a table for the predicted wins.

```
##      fit      lwr      upr
## Min.   : -11.14    Min.   : -298.56    Min.   : 17.65
## 1st Qu.: 72.01     1st Qu.: 51.04     1st Qu.: 92.83
```

## Median :	80.28	Median :	59.67	Median :	100.67
## Mean :	85.60	Mean :	57.08	Mean :	114.11
## 3rd Qu.:	87.29	3rd Qu.:	66.81	3rd Qu.:	107.79
## Max. :	1256.60	Max. :	440.64	Max. :	2072.55

Conclusion

We conclude that model one which includes a majority of the predictors except one provides the best overall fit. While we did try additional models based on transformed variables, they did not provide a significant improvement, so we decided to go with model one. This model does not seem to violate the assumptions of linear regression.

References

Sellmair, Reinhard. “How to handle correlated Features?” June 25, 2018. <https://www.kaggle.com/reisel/how-to-handle-correlated-features>

Xie, Yihui, J. J. Allaire, and Garrett Golemund, *R Markdown: The Definitive Guide*, CRC Press December 14, 2020 <https://bookdown.org/yihui/rmarkdown/r-code.html>.

<https://rstatisticsblog.com/data-science-in-action/data-preprocessing/six-amazing-function-to-create-train-test-split-in-r/>

R Code

```
# =====
# Load Libraries and Disable Scientific Notation for Readability Purposes
# =====
```

```
knitr::opts_chunk$set(echo = TRUE)
# Disable scientific numbers for readability purposes.
options(scipen = 999)
```

```
library(MASS)
library(tidyverse)
library(dplyr)
library(reshape2)
library(kableExtra)
library(corrplot)
library(ggplot2)
library(Hmisc)
library(PerformanceAnalytics)
library(GGally)
library(ggpubr)
library(car)
```

```
# =====
# Load The Dataset and Summarize the Data
# =====
```

```
# Load in the training data.
```

```
url = "https://raw.githubusercontent.com/Jagdish16/CUNY_DATA_621/main/project_1/moneyball-training-data"
mb_training <- read.csv(url)
```

```
# Remove the INDEX variable as it is of no value in the data evaluation.
```

```

mb_training <- subset(mb_training, select = -c(INDEX))

# Summarize the test data.
summary(mb_training)

# =====
# Rename the Variables to be More Intuitive
# =====

# Rename the columns to be more intuitive.
mb_training <- mb_training %>%
  rename_with(~ gsub("TEAM_", "", .x)) %>%
  rename_with(stringr::str_to_title) %>%
  dplyr::rename(
    Wins = Target_wins,
    Hits = Batting_h,
    Doubles = Batting_2b,
    Triples = Batting_3b,
    HomeRuns = Batting_hr,
    Walks_AtBat = Batting_bb,
    StrikeOuts_AtBat = Batting_so,
    BasesStolen = Baserun_sb,
    OutStealingBases = Baserun_cs,
    Hits_Allowed = Pitching_h,
    HitByPitch_AtBat = Batting_hbp,
    Errors = Fielding_e,
    HomeRuns_Allowed = Pitching_hr,
    Walks_Allowed = Pitching_bb,
    StrikeOuts = Pitching_so,
    DoublePlays = Fielding_dp
  )

# =====
# Box Plots
# =====

# Plot boxplots for all variables.
long <- mb_training %>% as.data.frame() %>% melt()

long %>%
  ggplot(aes(x=value)) + geom_boxplot() + facet_wrap(~variable, scales = 'free')

# =====
# Distribution Plots
# =====

# Wins.
ggplot(mb_training, aes(x = Wins)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Wins)),
    linetype = 'dashed', size = 2, color = 'blue') +

```

```

geom_label(aes(x = 50, y = 125,
               label = str_replace_all(toString(summary(mb_training['Wins'])), ',', '\n')
               )) +
labs(title = 'Wins Histogram Plot', y = 'Count', x = 'Wins')

# Hits.
ggplot(mb_training, aes(x = Hits)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Hits)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 1250, y = 100,
                 label = str_replace_all(toString(summary(mb_training['Hits'])), ',', '\n')
                 )) +
  labs(title = 'Hits Histogram Plot', y = 'Count', x = 'Base Hits')

# Doubles.
ggplot(data = mb_training, aes(x = Doubles)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Doubles)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 160, y = 100,
                 label = str_replace_all(toString(summary(mb_training['Doubles'])), ',', '\n')
                 )) +
  labs(title = 'Doubles Histogram Plot', y = 'Count', x = 'Doubles')

# Triples.
ggplot(data = mb_training, aes(x = Triples)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Triples)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 75, y = 130,
                 label = str_replace_all(toString(summary(mb_training['Triples'])), ',', '\n')
                 )) +
  labs(title = 'Triples Histogram Plot', y = 'Count', x = 'Triples')

# Homeruns.
ggplot(data = mb_training, aes(x = HomeRuns)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(HomeRuns)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 25, y = 90,
                 label = str_replace_all(toString(summary(mb_training['HomeRuns'])), ',', '\n')
                 )) +
  labs(title = 'Homeruns Histogram Plot', y = 'Count', x = 'Homeruns')

# Walks.
ggplot(data = mb_training, aes(x = Walks_AtBat)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Walks_AtBat)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 380, y = 100,

```



```

        label = str_replace_all(toString(summary(mb_training['Walks_AtBat'])), ',', '\n')
    )) +
labs(title = 'Walks Histogram Plot', y = 'Count', x = 'Walks')

# Strike Out by Batters.
ggplot(data = mb_training, aes(x = StrikeOuts_AtBat)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(StrikeOuts_AtBat)),
    linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 380, y = 100,
    label = str_replace_all(toString(summary(mb_training['StrikeOuts_AtBat'])), ',', '\n')
  )) +
labs(title = 'Strike Out by Batters Histogram Plot', y = 'Count', x = 'Strike Out by Batters')

# Stolen Bases.
ggplot(data = mb_training, aes(x = BasesStolen)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(BasesStolen)),
    linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 200, y = 100,
    label = str_replace_all(toString(summary(mb_training['BasesStolen'])), ',', '\n')
  )) +
labs(title = 'Stolen Bases Histogram Plot', y = 'Count', x = 'Stolen Bases')

# Hits Allowed.
ggplot(data = mb_training, aes(x = Hits_Allowed)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Hits_Allowed)),
    linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 1250, y = 100,
    label = str_replace_all(toString(summary(mb_training['Hits_Allowed'])), ',', '\n')
  )) +
labs(title = 'Hits Allowed Histogram Plot', y = 'Count', x = 'Hits Allowed')

# Homeruns Allowed.
ggplot(data = mb_training, aes(x = HomeRuns_Allowed)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(HomeRuns_Allowed)),
    linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 30, y = 90,
    label = str_replace_all(toString(summary(mb_training['HomeRuns_Allowed'])), ',', '\n')
  )) +
labs(title = 'Homeruns Allowed Histogram Plot', y = 'Count', x = 'Homeruns Allowed')

# Walks Allowed.
ggplot(data = mb_training, aes(x = Walks_Allowed)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Walks_Allowed)),
    linetype = 'dashed', size = 2, color = 'blue') +

```

```

geom_label(aes(x = 375, y = 100,
               label = str_replace_all(toString(summary(mb_training['Walks_Allowed'])), ',', '\n')
               )) +
labs(title = 'Walks Allowed Histogram Plot', y = 'Count', x = 'Walks Allowed')

# Strikeouts by Pitchers.
ggplot(data = mb_training, aes(x = StrikeOuts)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(StrikeOuts)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 425, y = 100,
                 label = str_replace_all(toString(summary(mb_training['StrikeOuts'])), ',', '\n')
                 )) +
labs(title = 'Strikeouts by Pitchers Histogram Plot', y = 'Count', x = 'Strikeouts by Pitchers')

# Errors.
ggplot(data = mb_training, aes(x = Errors)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(Errors)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 225, y = 100,
                 label = str_replace_all(toString(summary(mb_training['Errors'])), ',', '\n')
                 )) +
labs(title = 'Errors Histogram Plot', y = 'Count', x = 'Errors')

# Double Plays.
ggplot(data = mb_training, aes(x = DoublePlays)) +
  geom_histogram(color = 'black', fill = 'gray', bins = 30) +
  geom_vline(aes(xintercept = mean(DoublePlays)),
             linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 110, y = 100,
                 label = str_replace_all(toString(summary(mb_training['DoublePlays'])), ',', '\n')
                 )) +
labs(title = 'Double Plays Histogram Plot', y = 'Count', x = 'Double Plays')

# =====
# Missing Data
# =====

# Create a table of variables sorted by percentage of missing data.
missing_data <- colSums(mb_training %>% sapply(is.na))
percentage_missing <- round(missing_data / nrow(mb_training) * 100, 2)
missing_values_table <- sort(percentage_missing, decreasing = TRUE)

missing_values_table %>%
  kable(caption = 'Breakdown of Variables by Percentage of Missing Data') %>%
  kable_styling()

# Drop the HitByPitch_AtBat variable from the dataset.
mb_training <- mb_training %>% dplyr::select(-HitByPitch_AtBat)

```

```

# =====
# Handle Outliers
# =====

# Remove outlier rows for the 6 predictor variables.
mb_training_updated <- mb_training

# Remove outliers - Method 2.
for (n in c("Walks_Allowed", "BasesStolen", "StrikeOuts", "Hits_Allowed", "Errors", "Triples")) {
  Q <- quantile(mb_training[,n], probs = c(.25, .75), na.rm = TRUE)
  iqr <- IQR(mb_training[,n], na.rm = TRUE)
  # Upper Range.
  up <- Q[2] + 1.5 * iqr
  # Lower Range.
  low <- Q[1] - 1.5 * iqr
  mb_training_updated <- subset(mb_training_updated, mb_training_updated[,n] > (Q[1]-1.5 * iqr)&mb_traini
}

# Check the summary for the updated dataframe.
summary(mb_training_updated)

# Impute missing values with the median value for each remaining column.
mb_training_updated <- data.frame(sapply(mb_training_updated, function(x) ifelse(is.na(x), median(x, na

# Check the summary for the updated dataframe.
summary(mb_training_updated)

# =====
# Data Correlation
# =====

# Perform a correlation analysis on the data. In this analysis, we are only interested in the
# correlation of the predictor variables and the "TARGET_WINS" variable.
correlation_table <- cor(mb_training_updated, method = 'pearson', use = 'complete.obs')[,1]

# Remove the TARGET_WINS variable from the correlation table as it is redundant
# within the context of our correlation analysis.
correlation_table <- correlation_table[-c(1)]

correlation_table %>%
  kable(caption = 'Correlation of Variables to Wins') %>% kable_styling()

# Calculate correlation between variables.
mb_training_updated_corr_matrix <- mb_training_updated %>% cor() %>% round(2) %>% as.matrix()
mb_training_updated_corr_matrix %>% kable() %>% kable_styling()

# flattenCorrMatrix
# cormat : matrix of the correlation coefficients.
# pmat : matrix of the correlation p-values.
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],

```

```

    cor =(cormat)[ut],
    p = pmat[ut]
  )
}

# Another method to check correlations and their significance.
corr.mat<-rcorr(as.matrix(mb_training_updated))

flattenCorrMatrix(corr.mat$r, corr.mat$p)%>% filter(row=='Wins') %>% arrange(-abs(cor))

# =====
# Check Normality of Predictors
# =====

# Run the Shapiro wilkes test for normality.
do.call(rbind, lapply(mb_training_updated, function(x) shapiro.test(x)[c("statistic", "p.value"])))

# =====
# Model 1
# =====

model_one <- lm(Wins ~ Hits + Doubles + Triples + HomeRuns +
               Walks_AtBat + BasesStolen + Hits_Allowed +
               HomeRuns_Allowed + Errors + Walks_Allowed + StrikeOuts +
               StrikeOuts_AtBat + OutStealingBases + DoublePlays,
               mb_training_updated)

# Model 1 summary stats.
summary(model_one)

# Model 1 R Squared.
summary(model_one)$r.squared

# Model 1 Confidence Intervals.
confint(model_one)

# Model 1 plots - residuals vs fitted values, residuals distribution.
ggplot(data = model_one, aes(x = .fitted, y = .resid)) +
  geom_point() + geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE) + xlab("Fitted values") + ylab("Residuals")

ggplot(data = model_one, aes(x = .resid)) + geom_histogram() + xlab("Residuals")

ggplot(data = model_one) + stat_qq(aes(sample = .stdresid)) + geom_abline()

# =====
# Model 2
# =====

# Model 2 uses stepwise regression on the variables in Model 1.
model_two <- stepAIC(model_one, direction = 'both', trace = FALSE)

```

```

# Model 2 summary stats.
summary(model_two)

# Model 2 plots - residuals vs fitted values, residuals distribution.
ggplot(data = model_two, aes(x = .fitted, y = .resid)) +
  geom_point() + geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE) + xlab("Fitted values") + ylab("Residuals")

ggplot(data = model_two, aes(x = .resid)) + geom_histogram() + xlab("Residuals")

ggplot(data = model_two) + stat_qq(aes(sample = .stdresid)) + geom_abline()

# =====
# Model 3
# =====

# Derive 2 new variables for Singles and Home run difference.
mb_training_new <- mb_training_updated %>% mutate(Singles = Hits - Doubles - Triples - HomeRuns)
mb_training_new <- mb_training_new %>% mutate(Homeruns_diff = HomeRuns_Allowed - HomeRuns)

model_three <- lm(Wins ~ Hits + Doubles + Triples + Walks_AtBat +
  BasesStolen + Hits_Allowed + Errors + Walks_Allowed +
  StrikeOuts + Singles + Homeruns_diff + StrikeOuts_AtBat +
  I(HomeRuns_Allowed/HomeRuns) + I(HomeRuns_Allowed*HomeRuns) +
  I(1/DoublePlays) + I(OutStealingBases^3),
  mb_training_new)

# Model 3 summary stats.
summary(model_three)

# Model 3 R-Squared.
summary(model_three)$r.squared

# Model 3 confidence intervals.
confint(model_three)

# Model 3 plots - residuals vs fitted values, residuals distribution.
ggplot(data = model_three, aes(x = .fitted, y = .resid)) +
  geom_point() + geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE) + xlab("Fitted values") + ylab("Residuals")

ggplot(data = model_three, aes(x = .resid)) + geom_histogram() + xlab("Residuals")

ggplot(data = model_three) + stat_qq(aes(sample = .stdresid)) + geom_abline()

# =====
# Model 4
# =====

# Model 4 - Box Cox method.

```

```

MASS::boxcox(model_one, lambda = seq(0.75, 1.6, by = 0.05), plotit = TRUE)

# Fit a model using a lambda value of 1.35 for the response variable.
model_cox = lm((((Wins ~ 1.35) - 1)/ 1.35) ~ Hits + Doubles + Triples + HomeRuns + Walks_AtBat +
  BasesStolen + Hits_Allowed + HomeRuns_Allowed + Errors +
  Walks_Allowed + StrikeOuts + StrikeOuts_AtBat + OutStealingBases +
  DoublePlays,
  mb_training_updated)

# Model 4 summary stats.
summary(model_cox)

# Model 4 R Squared.
summary(model_cox)$r.squared

# Model 4 confidence intervals.
confint(model_cox)

# Model 4 plots - residuals vs fitted values, residuals distribution.
ggplot(data = model_cox, aes(x = .fitted, y = .resid)) +
  geom_point() + geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE) + xlab("Fitted values") + ylab("Residuals")

ggplot(data = model_cox, aes(x = .resid)) + geom_histogram() + xlab("Residuals")

ggplot(data = model_cox) + stat_qq(aes(sample = .stdresid)) + geom_abline()

# =====
# Model Selection
# =====

# Predict the number of wins for the test data using model one.

# Load in the test data.
url2 <- 'https://raw.githubusercontent.com/Jagdish16/CUNY_DATA_621/main/project_1/moneyball-evaluation-
mb_test <- read.csv(url2)

# Remove the INDEX variable as it is of no value in the data evaluation.
mb_test <- subset(mb_test, select = -c(INDEX))

# Summarize the test data.
summary(mb_test)

# Rename the test data variables to be more intuitive.
mb_test <- mb_test %>%
  rename_with(~ gsub("TEAM_", "", .x)) %>%
  rename_with(stringr::str_to_title) %>%
  dplyr::rename(
    Hits = Batting_h,
    Doubles = Batting_2b,
    Triples = Batting_3b,
    HomeRuns = Batting_hr,

```

```

Walks_AtBat = Batting_bb,
StrikeOuts_AtBat = Batting_so,
BasesStolen = Baserun_sb,
OutStealingBases = Baserun_cs,
Hits_Allowed = Pitching_h,
HitByPitch_AtBat = Batting_hbp,
Errors = Fielding_e,
HomeRuns_Allowed = Pitching_hr,
Walks_Allowed = Pitching_bb,
StrikeOuts = Pitching_so,
DoublePlays = Fielding_dp
)

# Impute missing values with the median value for each column.
mb_test_updated <- data.frame(sapply(mb_test, function(x) ifelse(is.na(x), median(x, na.rm = TRUE), x)))

# Summarize the test data.
summary(mb_test_updated)

# Predicting Wins in the test data and looking at the distribution.
mb_test_updated$predicted_wins <- predict(model_one, type = 'response', newdata = mb_test_updated)

ggplot(data = mb_test_updated, aes(x = predicted_wins)) +
  geom_histogram( color = 'black', fill = 'gray') +
  geom_vline(aes(xintercept = mean(predicted_wins)), linetype = 'dashed', size = 2, color = 'blue') +
  geom_label(aes(x = 500, y = 150, label= str_replace_all(toString(summary(mb_test_updated['predicted_wins']),
labs(title = 'Wins Prediction Histogram Plot', y = 'Count', x = 'Wins Prediction'))

# Create a table of prediction and confidence intervals.
test_data <- predict(model_one, newdata = mb_test_updated, interval = 'prediction')
summary(test_data)

```