Cy Young Prediction Project

Jonah Bonesteel

2022-11-07

For my final project, I am interested in dealing with baseball statistics. My idea for the project is to identify key metrics that are most significant in determining whether a pitcher receives points for the Cy Young Award. I also hope to create a regression model that can predict a how many Cy Young points a player will receive based on their current statistics.

Each league's award is voted on by members of the Baseball Writers' Association of America, with one representative from each team. As of the 2010 season, each voter places a vote for first, second, third, fourth, and fifth place among the pitchers of each league. The formula used to calculate the final scores is a weighted sum of the votes. The pitcher with the highest score in each league wins the award. If two pitchers receive the same number of votes, the award is shared. From 1970 to 2009, writers voted for three pitchers, with the formula of five points for a first-place vote, three for a second-place vote and one for a third-place vote. Prior to 1970, writers only voted for the best pitcher and used a formula of one point per vote.My hope for this project is that I will be able to run my models on current player statistics at the end of each season to predict who will win the Cy Young Award race. (source: Wikipedia)

The data I have is an MLB Cy Young Award data set that goes back to 1956. There are 56 variables including the predictor variable "pointsWon", which contains the number of Cy Young points a player received.

Part 1: Exploratory Analysis and Data Cleaning

```
library(readr)
cy_data <- read_csv("cy_data.csv")
```

```
## Rows: 803 Columns: 56
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (18): playerID, teamID, lgID, birthCountry, birthState, birthCity, death...
## dbl (38): yearID, pointsWon, stint, W, L, G, GS, CG, SHO, SV, IP, IPouts, H,...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

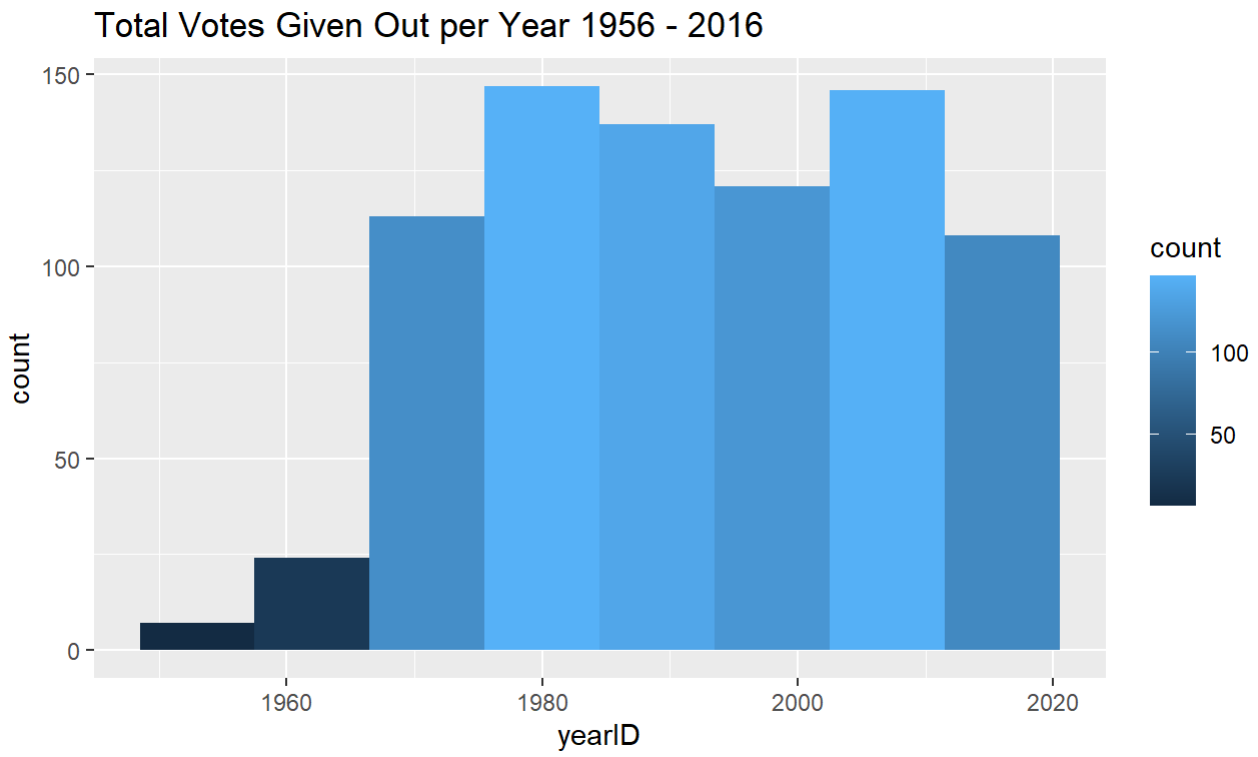
In order to alter the data to the format we would like, we must first know the type of each column.

```
sapply(cy_data, class)
```

```
##       yearID     playerID    pointsWon        stint       teamID        lgID
##    "numeric"  "character"    "numeric"    "numeric"  "character"  "character"
##            W            L            G           GS           CG          SHO
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##           SV           IP       IPouts            H           ER           HR
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##           BB           SO        BAOpp          ERA         WHIP          IBB
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##           WP          HBP           BK          BFP           GF            R
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##           SH           SF         GIDP    birthYear   birthMonth     birthDay
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
## birthCountry   birthState    birthCity    deathYear   deathMonth     deathDay
##  "character"  "character"  "character"    "numeric"    "numeric"    "numeric"
## deathCountry   deathState    deathCity    nameFirst     nameLast    nameGiven
##  "character"  "character"  "character"  "character"  "character"  "character"
##       weight       height         bats       throws        debut    finalGame
##    "numeric"    "numeric"  "character"  "character"  "character"  "character"
##      retroID      bbrefID
##  "character"  "character"
```

From part 1 of the project, we know that the number of votes given out dramatically increased to what it still is today in the year 1970. Let's visualize this again.

```
library(ggplot2)
ggplot(cy_data, aes(yearID)) + geom_histogram(aes(fill = ..count..), binwidth = 9) + ggtitle('To
tal Votes Given Out per Year 1956 - 2016')
```



Due to the dramatic increase in total Cy Young votes per year in 1970, lets start by eliminating values prior to that year.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

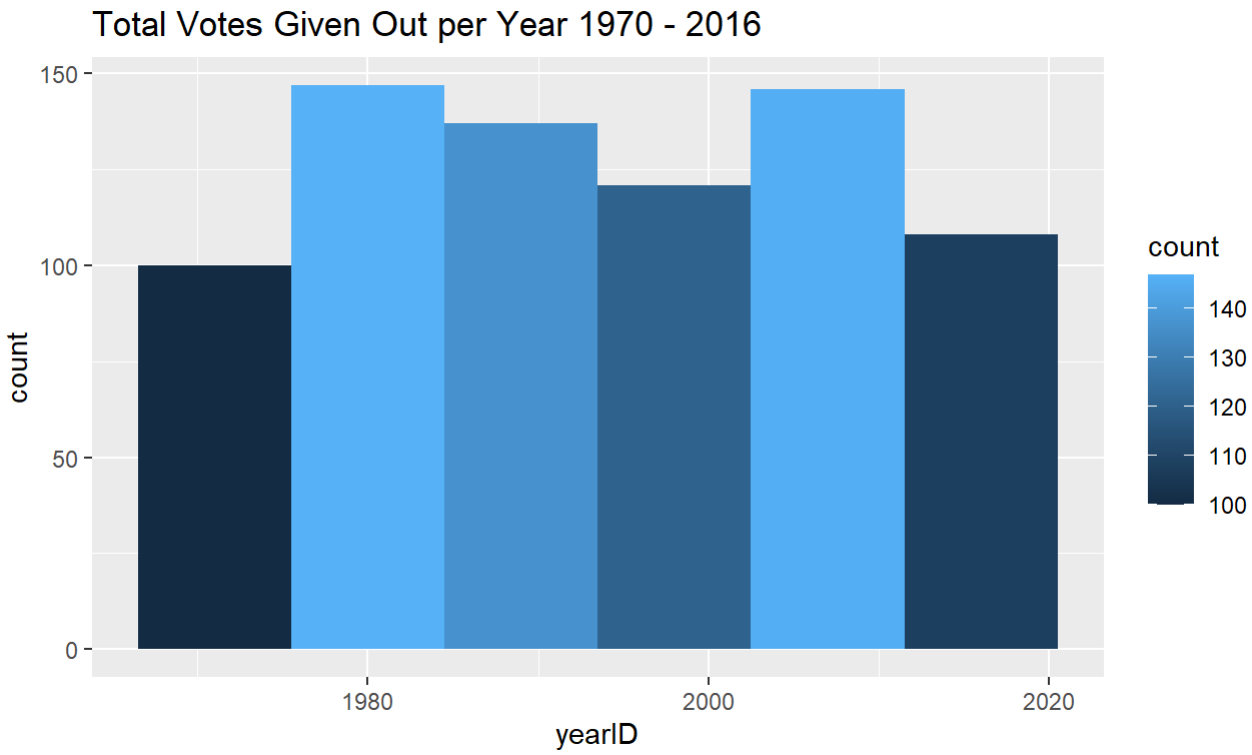
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
cy_data = filter(cy_data, yearID >= 1970)
```

Lets view the distribution of votes now.

```
library(ggplot2)
ggplot(cy_data, aes(yearID)) + geom_histogram(aes(fill = ..count..), binwidth = 9) + ggtitle('To
tal Votes Given Out per Year 1970 - 2016')
```



To do some preliminary exploratory analysis, lets create a subset object with only the variables we are interested in.

```
subset <- cy_data[,c("pointsWon","ERA","SO","HR","H","BAOpp","WHIP","BB","W","L","IP",'SV')]
```

Lets take a look at summary statistics and a basic correlation matrix for a few of the numeric variables to see if there are any patterns.

```
summary(subset)
```

```
##    pointsWon           ERA              SO              HR
##  Min.   :  0.00   Min.   :0.540   Min.   : 18.0   Min.   : 0.00
##  1st Qu.:  3.00   1st Qu.:2.400   1st Qu.:113.5   1st Qu.: 9.00
##  Median : 12.00   Median :2.840   Median :163.0   Median :16.00
##  Mean   : 35.75   Mean   :2.785   Mean   :162.5   Mean   :15.43
##  3rd Qu.: 55.50   3rd Qu.:3.200   3rd Qu.:205.0   3rd Qu.:21.00
##  Max.   :224.00   Max.   :5.590   Max.   :383.0   Max.   :41.00
##        H             BAOpp            WHIP             BB
##  Min.   : 14.0   Min.   :0.126   Min.   :0.550   Min.   :  2.00
##  1st Qu.:135.5   1st Qu.:0.212   1st Qu.:1.050   1st Qu.: 38.00
##  Median :189.0   Median :0.230   Median :1.130   Median : 56.00
##  Mean   :173.9   Mean   :0.227   Mean   :1.123   Mean   : 56.85
##  3rd Qu.:220.0   3rd Qu.:0.245   3rd Qu.:1.210   3rd Qu.: 73.00
##  Max.   :381.0   Max.   :0.298   Max.   :1.660   Max.   :204.00
##        W               L                IP              SV
##  Min.   : 0.00   Min.   : 0.000   Min.   : 29.0   Min.   : 0.00
##  1st Qu.:13.00   1st Qu.: 5.000   1st Qu.:167.8   1st Qu.: 0.00
##  Median :17.00   Median : 8.000   Median :221.7   Median : 0.00
##  Mean   :15.28   Mean   : 7.652   Mean   :203.6   Mean   : 6.56
##  3rd Qu.:20.00   3rd Qu.:10.000   3rd Qu.:249.5   3rd Qu.: 0.00
##  Max.   :27.00   Max.   :20.000   Max.   :376.7   Max.   :62.00
```

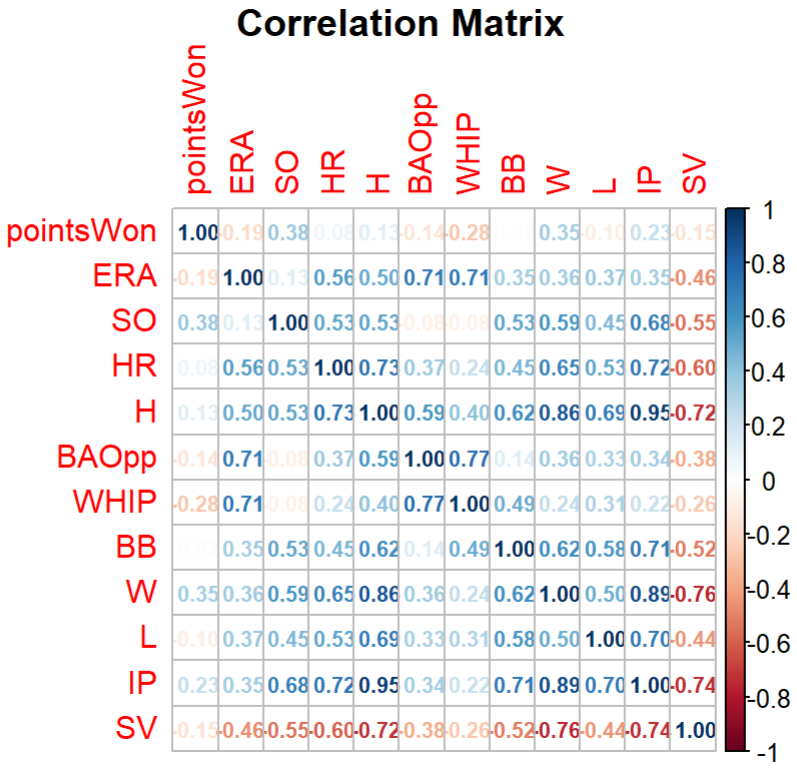
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
matrix<-cor(subset)
head(round(matrix,2))
```

```
##           pointsWon   ERA    SO   HR    H BAOpp  WHIP   BB    W     L   IP
## pointsWon      1.00 -0.19  0.38 0.08 0.13 -0.14 -0.28 0.02 0.35 -0.10 0.23
## ERA           -0.19  1.00  0.13 0.56 0.50  0.71  0.71 0.35 0.36  0.37 0.35
## SO             0.38  0.13  1.00 0.53 0.53 -0.08 -0.08 0.53 0.59  0.45 0.68
## HR             0.08  0.56  0.53 1.00 0.73  0.37  0.24 0.45 0.65  0.53 0.72
## H              0.13  0.50  0.53 0.73 1.00  0.59  0.40 0.62 0.86  0.69 0.95
## BAOpp         -0.14  0.71 -0.08 0.37 0.59  1.00  0.77 0.14 0.36  0.33 0.34
##              SV
## pointsWon -0.15
## ERA       -0.46
## SO        -0.55
## HR        -0.60
## H         -0.72
## BAOpp     -0.38
```

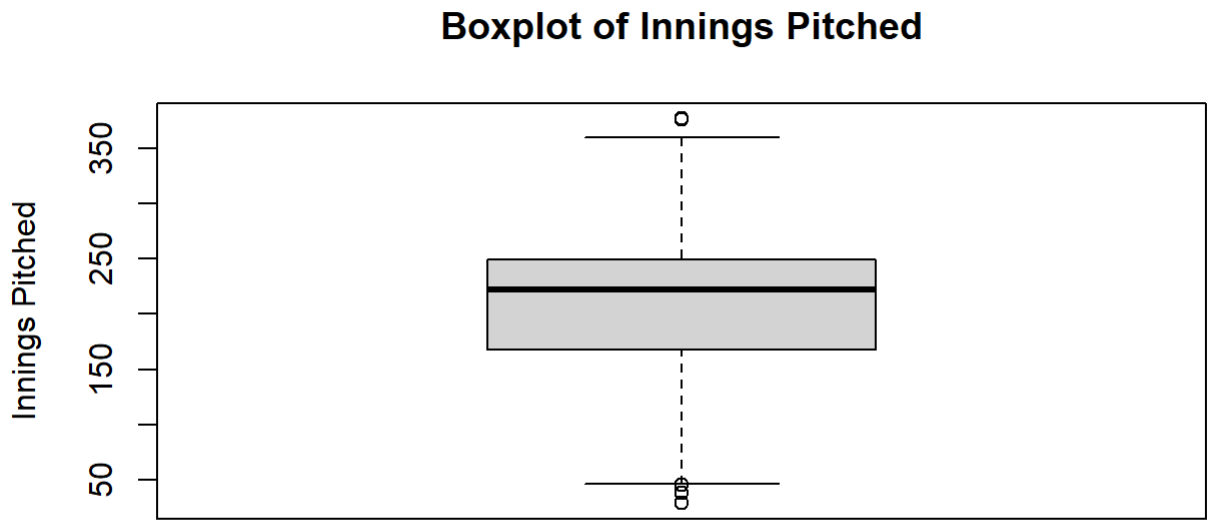
```
corrplot(matrix, method="number", title="Correlation Matrix", mar=c(0,0,1,0), number.cex=0.70)
```



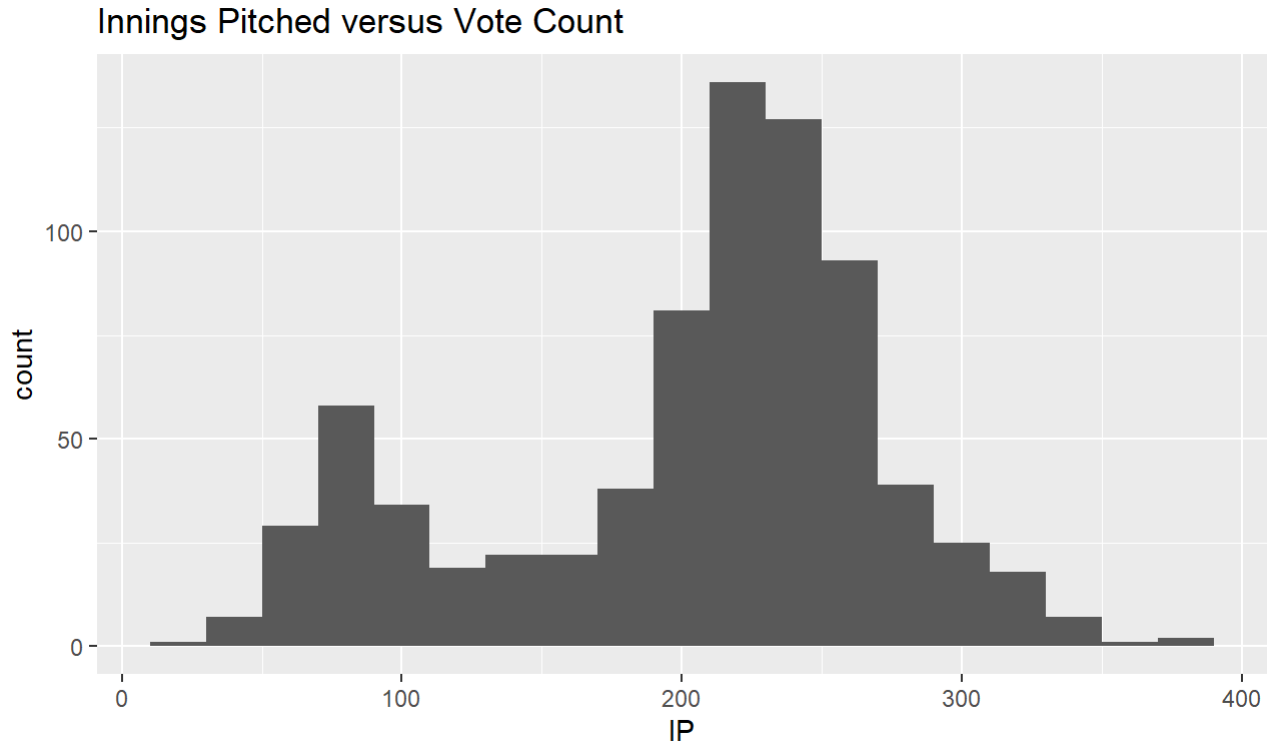
In preparation for data cleaning, lets do some more exploratory analysis to see if we can identify potential outliers or transformations.

I have a hunch that the Cy Young criteria for starters is different than for relievers. Lets look at some variables that would differ the most between starters and relievers, such as innings pitched and saves.

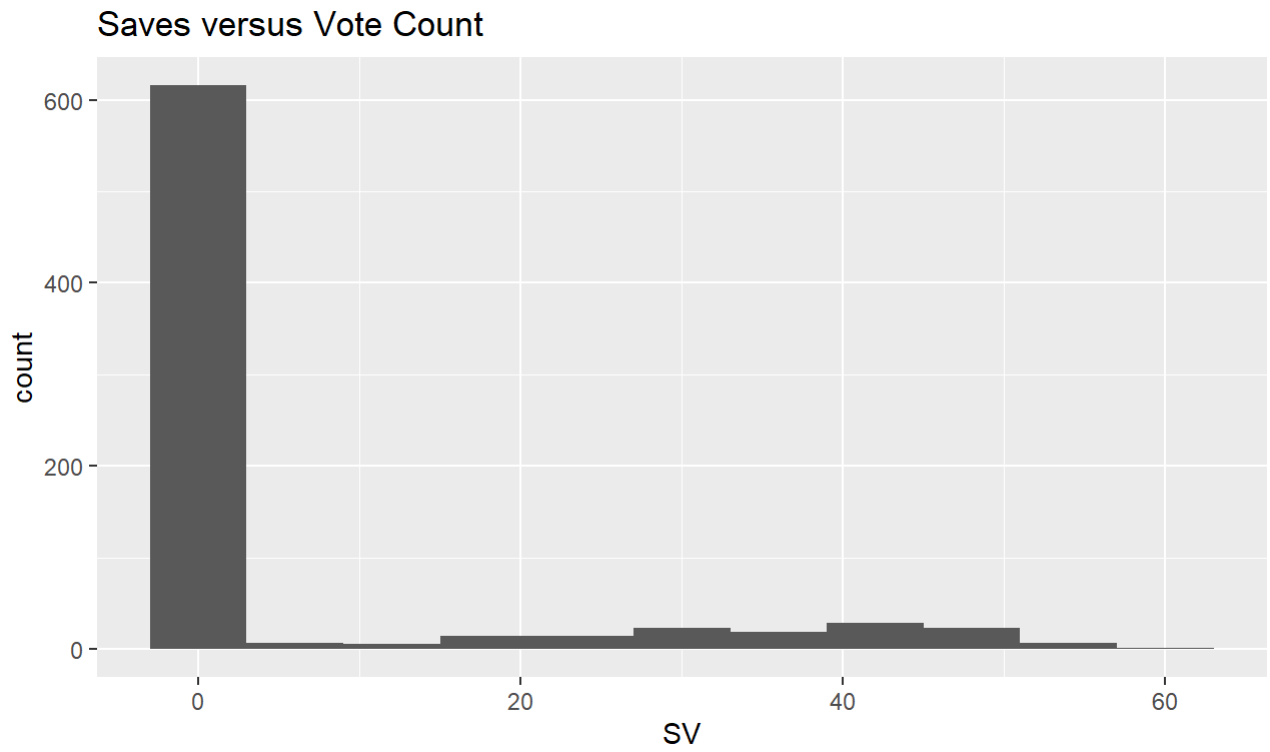
```
boxplot(subset$IP, main="Boxplot of Innings Pitched", ylab="Innings Pitched")
```



```
ggplot(subset, aes(IP)) + geom_histogram(aes(fill = pointsWon), binwidth = 20) + ggtitle('Innings Pitched versus Vote Count')
```




```
ggplot(subset, aes(SV)) + geom_histogram(aes(fill = pointsWon), binwidth = 6) + ggtitle('Saves versus Vote Count')
```



There does appear to be a clear divide among starters and relievers. The plot of innings pitched vs points shows there to be a bimodal distribution, where there is a large cluster of data points around the 75 to 100 innings pitched range as well as one around the 200 to 250 innings pitched range. This would make sense to explain starters vs relievers. Relievers would tend to be in the 75-100 innings pitched range, while starters would tend to be 150 innings pitched and above. Additionally, the plots surrounding the saves variable show us the clear divide between who are starters and who are relievers, as it is extremely rare that a starter records a save. However, there are relievers who do not earn saves.

A solution to this problem would be to use the ‘games started’ and ‘games appeared’ columns. A reliever ought to be classified as someone who is making lots of game appearances but having few starting appearances. Lets create a “games started rate” column where it is simply = games started / total games.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ tibble  3.1.7     ✓ stringr 1.4.0
## ✓ tidyr   1.2.0     ✓ forcats 0.5.1
## ✓ purrr   0.3.4
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

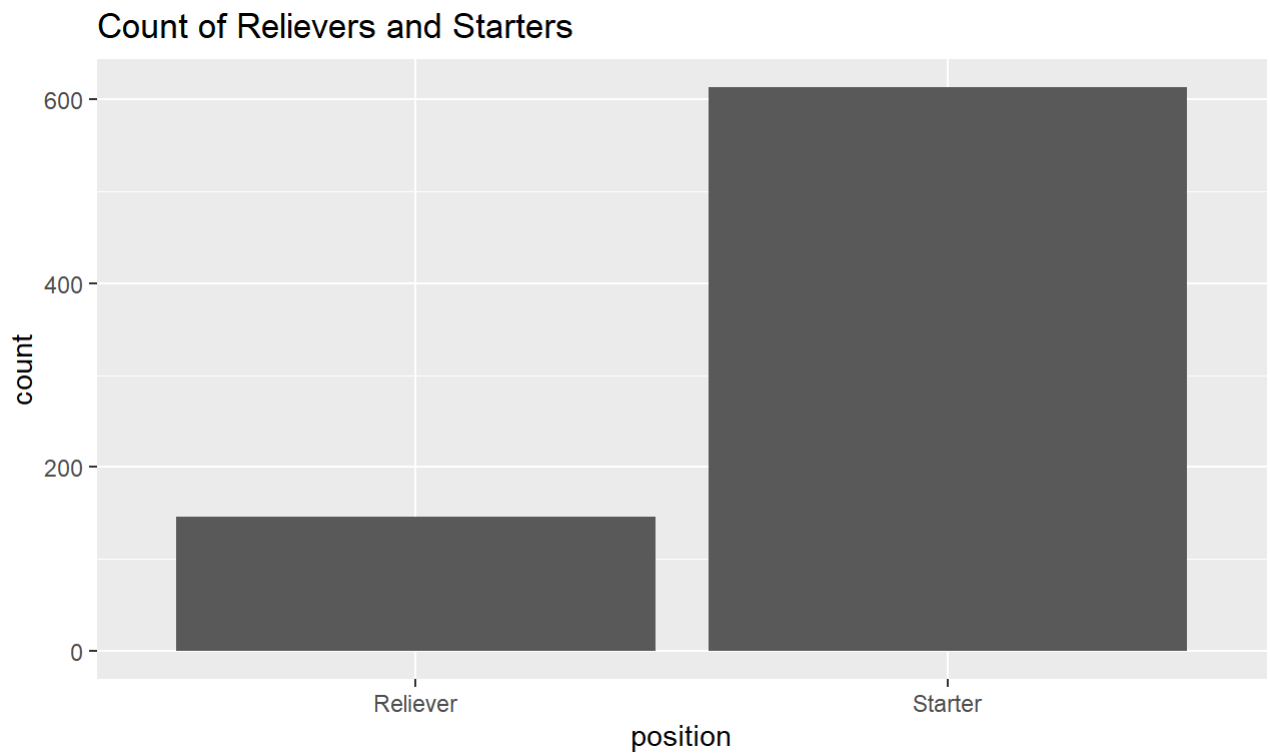
```
cy_data <- mutate(cy_data, start_rate = GS/G)
```

Define a reliever as someone with a start rate of lower than 25%.

```
cy_data$position <- ifelse(cy_data$start_rate < 0.25, 'Reliever', 'Starter')
```

Plot count of starters vs relievers in the data set.

```
library(ggplot2)
ggplot(data = cy_data, aes(x = position)) +
    geom_bar() + ggtitle('Count of Relievers and Starters')
```

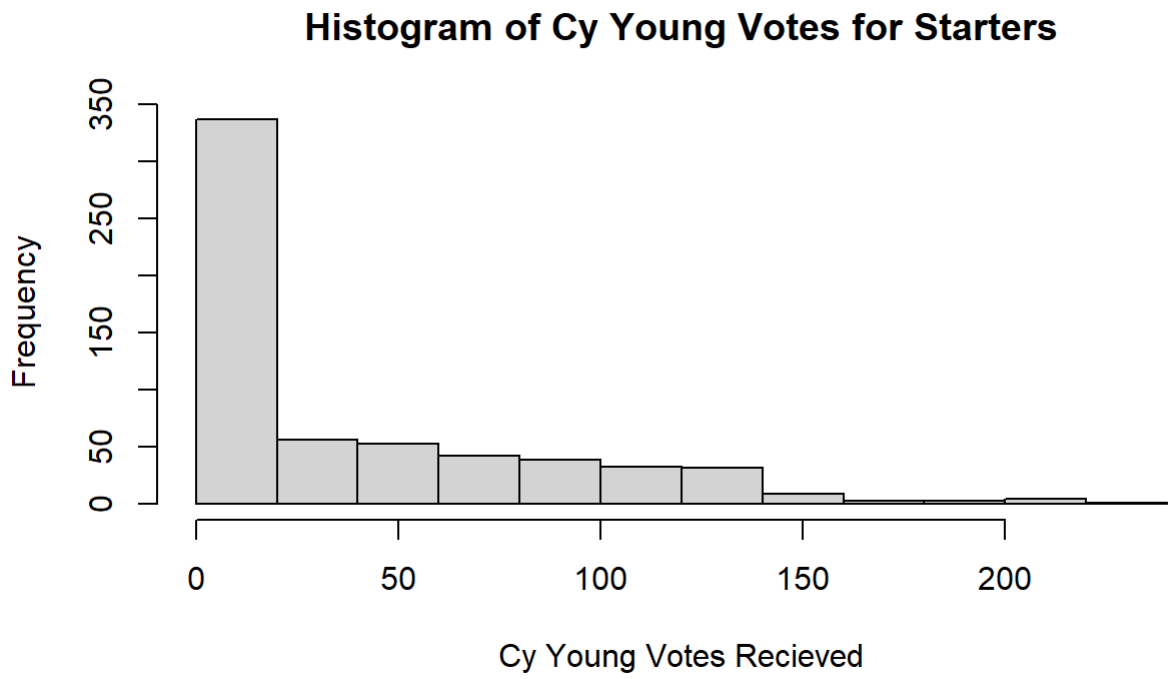


Because of the differences in each position's stats, I will be splitting up the data set to account for each position and run the analysis separately.

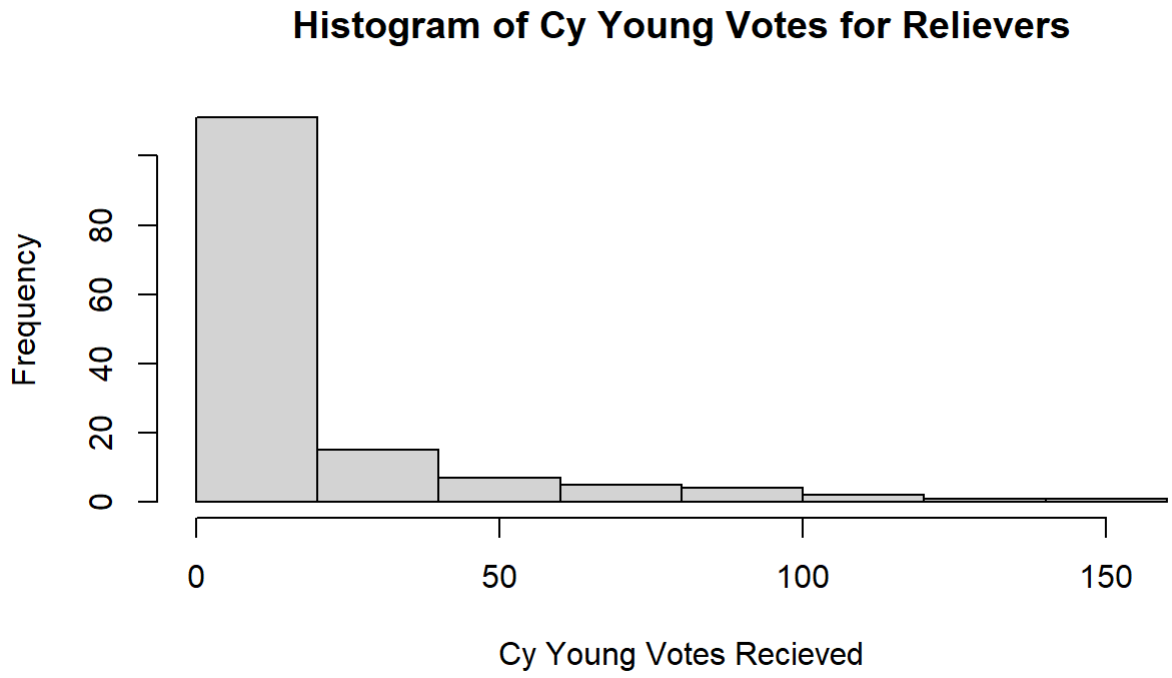
```
starterSubset <- subset(cy_data, position == 'Starter')
relieverSubset <- subset(cy_data, position == 'Reliever')
```

One problem I foresee is with the distribution of Cy Young Votes in our data set. Due to the nature of the voting system, the majority of players in this data set will have a low number of votes. Lets visualize this for both starters and relievers.

```
hist(starterSubset$pointsWon,main="Histogram of Cy Young Votes for Starters", xlab="Cy Young Vot
es Recieved")
```




```
hist(relieverSubset$pointsWon,main="Histogram of Cy Young Votes for Relievers", xlab="Cy Young V
otes Recieved")
```



Because of the right-skewness of the response variable in both the subsets, we will need to perform a log transformation on the pointsWon variable.

Part 2: Model Creation

Lets start with starters. First lets subset the whole data set into only include the variables we are interested in.

```
startersData <- starterSubset[,c('pointsWon','W','L','IP','H','HR','BB','SO','BAOpp','ERA','WHI
P')]
```

```
startersData %>% summarise_all(~ sum(is.na(.)))
```

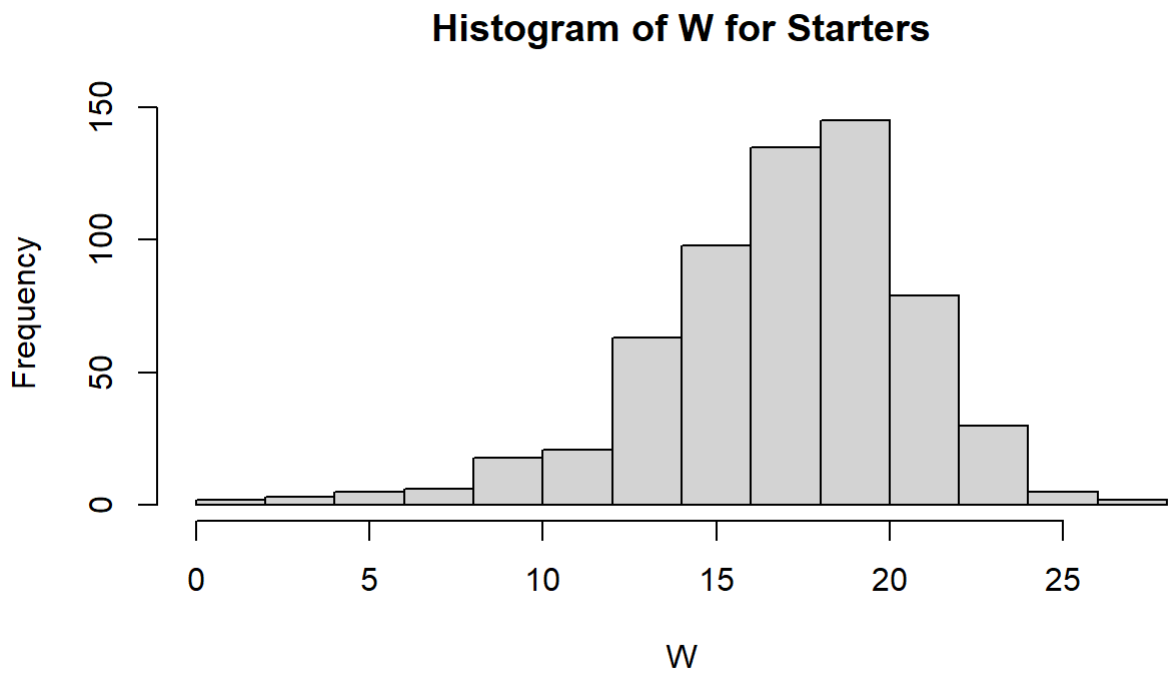
```
## # A tibble: 1 × 11
##   pointsWon     W     L    IP     H    HR    BB    SO BAOpp   ERA  WHIP
##       <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1         0     0     0     0     0     0     0     0     0     0     0
```

Lets remove the erroneous data value in row 169.

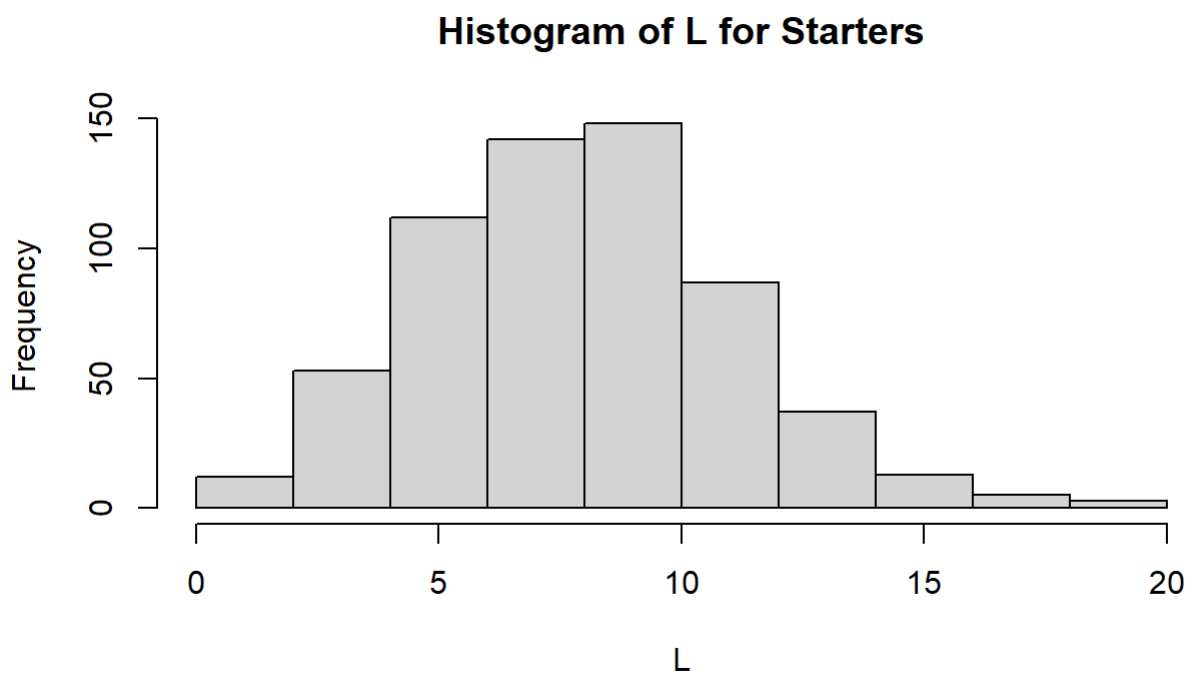
```
startersData <- startersData[-c(169), ]
```

Lets get an idea of the overall distributions of each variable.

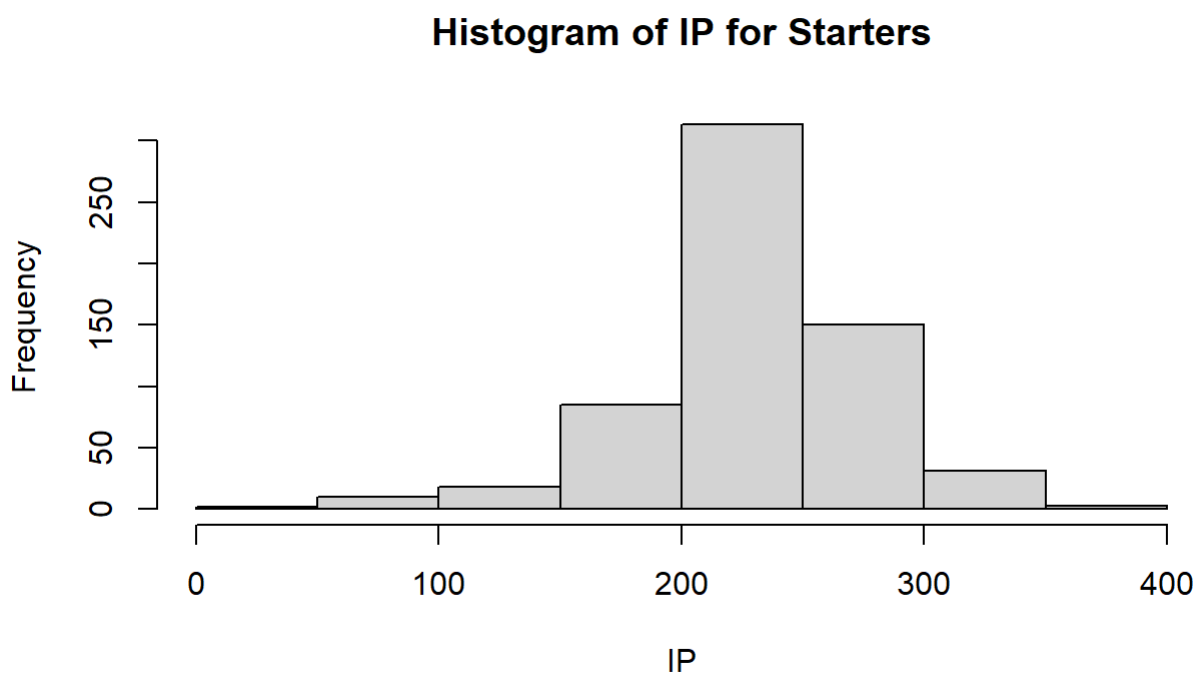
```
hist(startersData$W,main="Histogram of W for Starters", xlab="W")
```



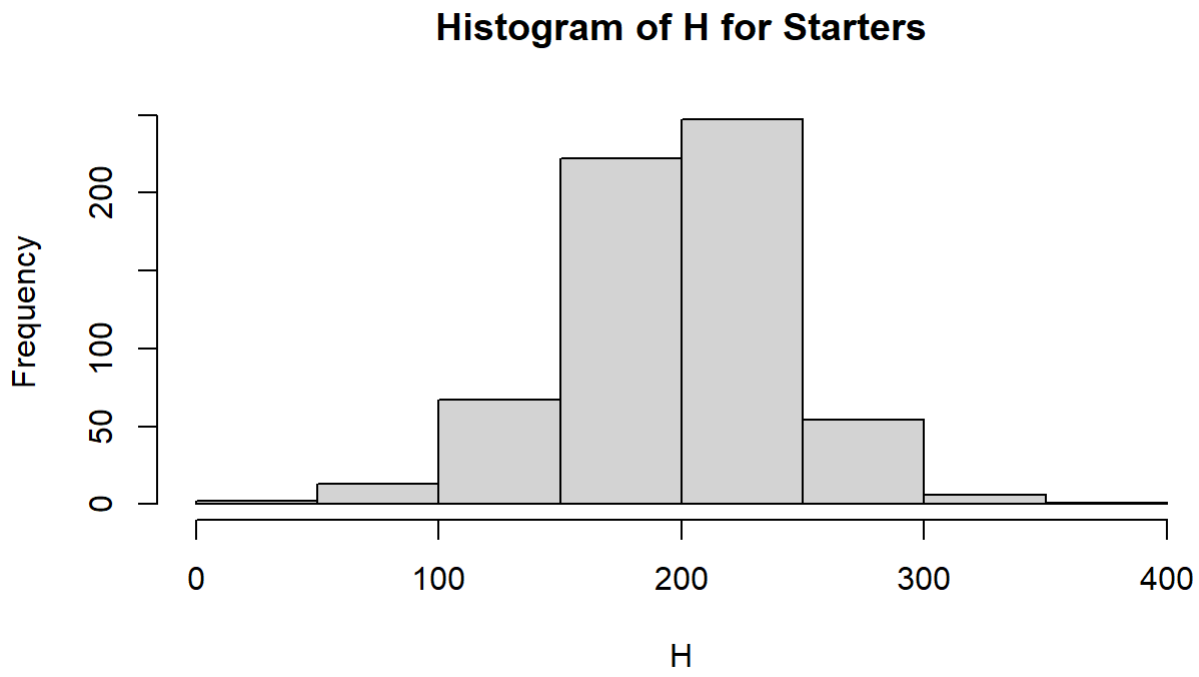

```
hist(startersData$L,main="Histogram of L for Starters", xlab="L")
```




```
hist(startersData$IP,main="Histogram of IP for Starters", xlab="IP")
```

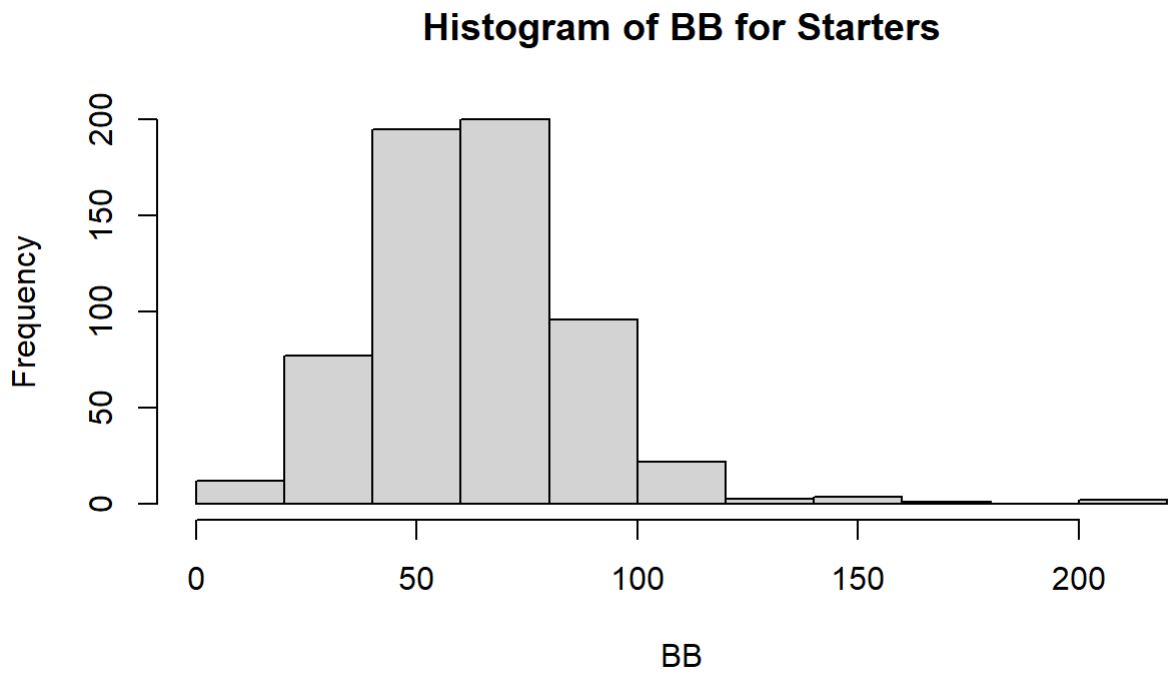



```
hist(startersData$H,main="Histogram of H for Starters", xlab="H")
```

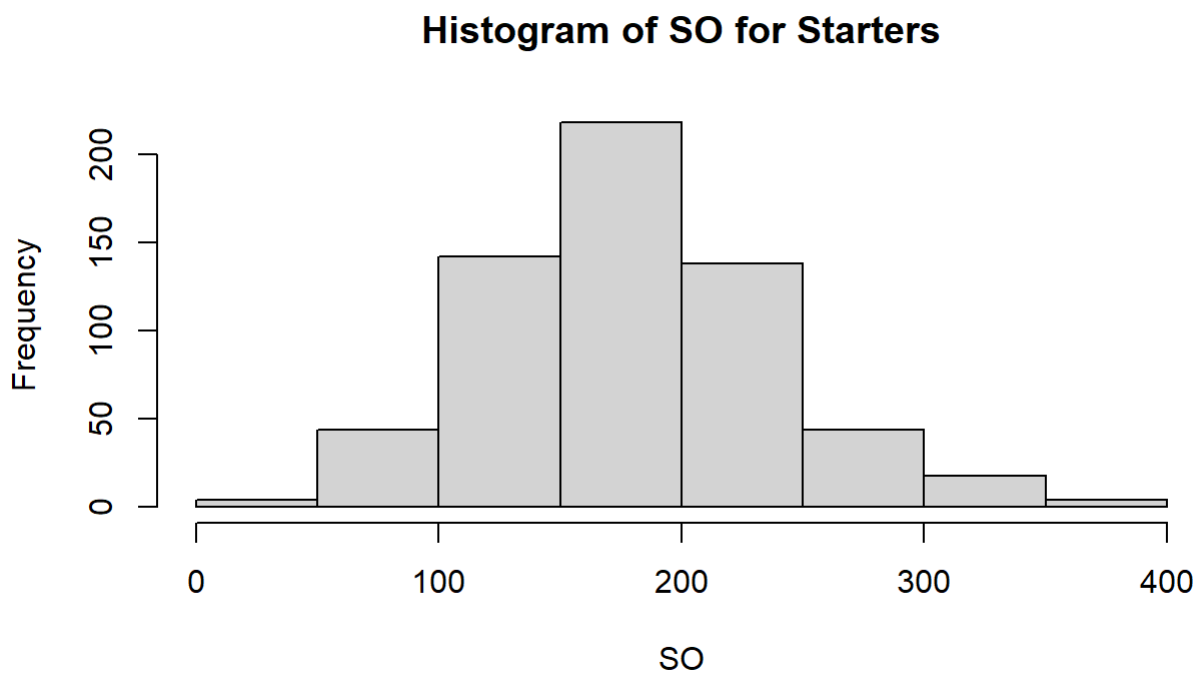



```
hist(startersData$HR,main="Histogram of HR for Starters", xlab="HR")
```

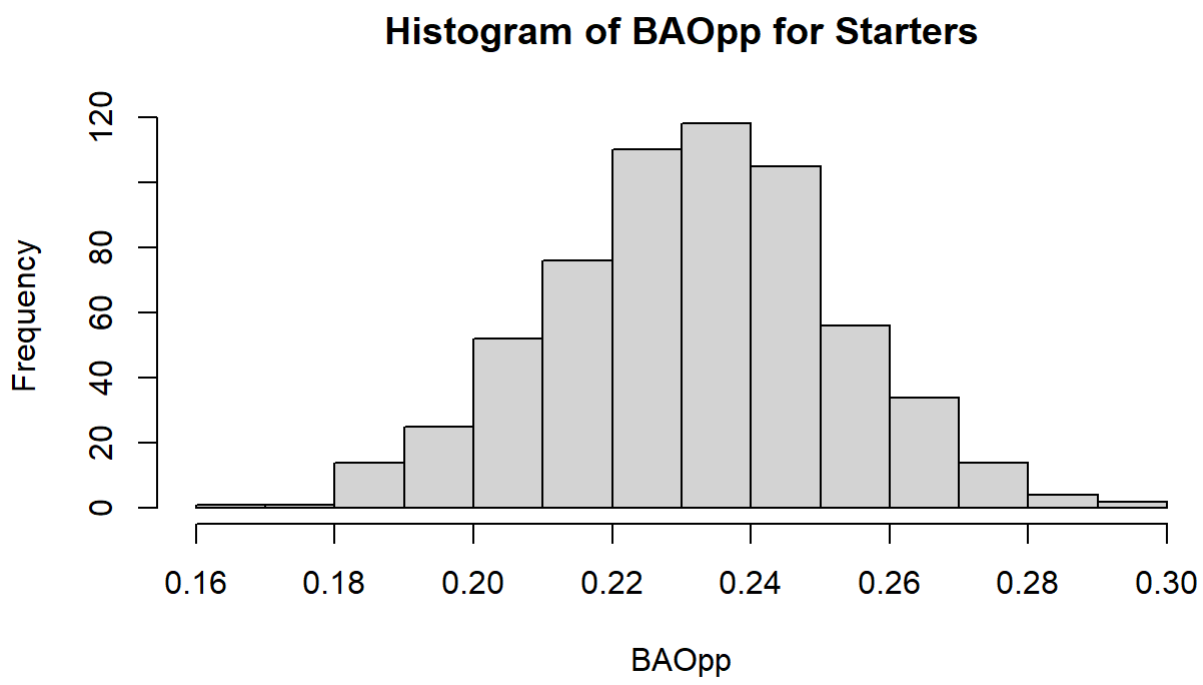

```
hist(startersData$BB,main="Histogram of BB for Starters", xlab="BB")
```



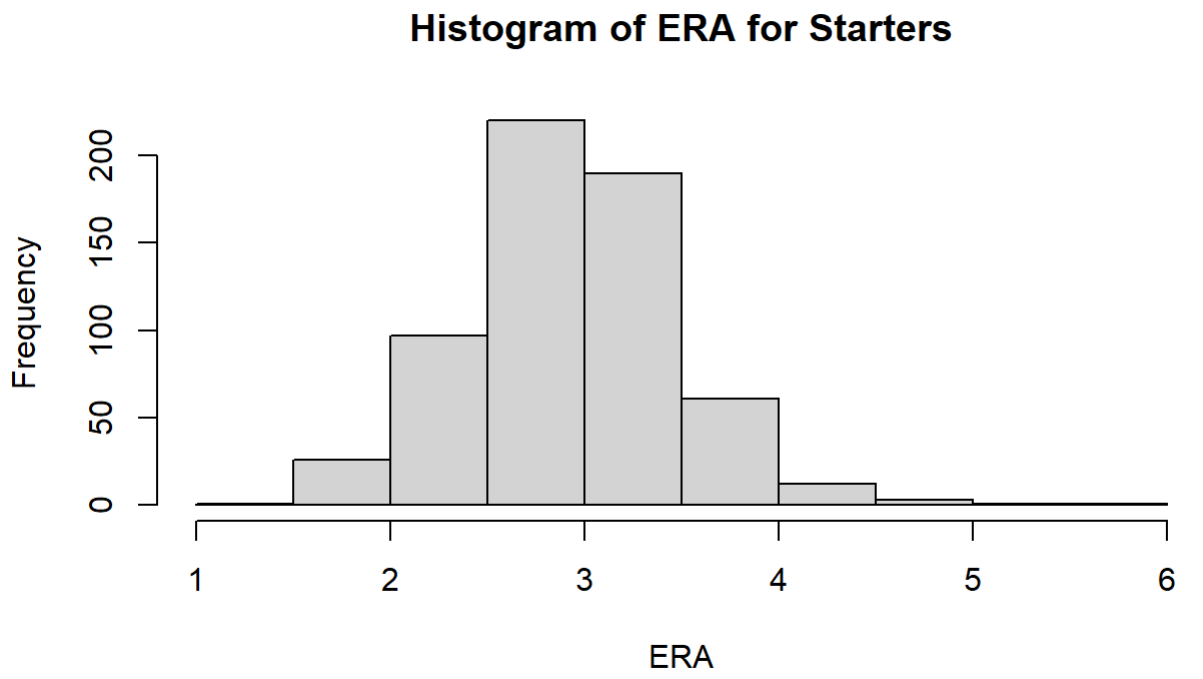

```
hist(startersData$SO,main="Histogram of SO for Starters", xlab="SO")
```



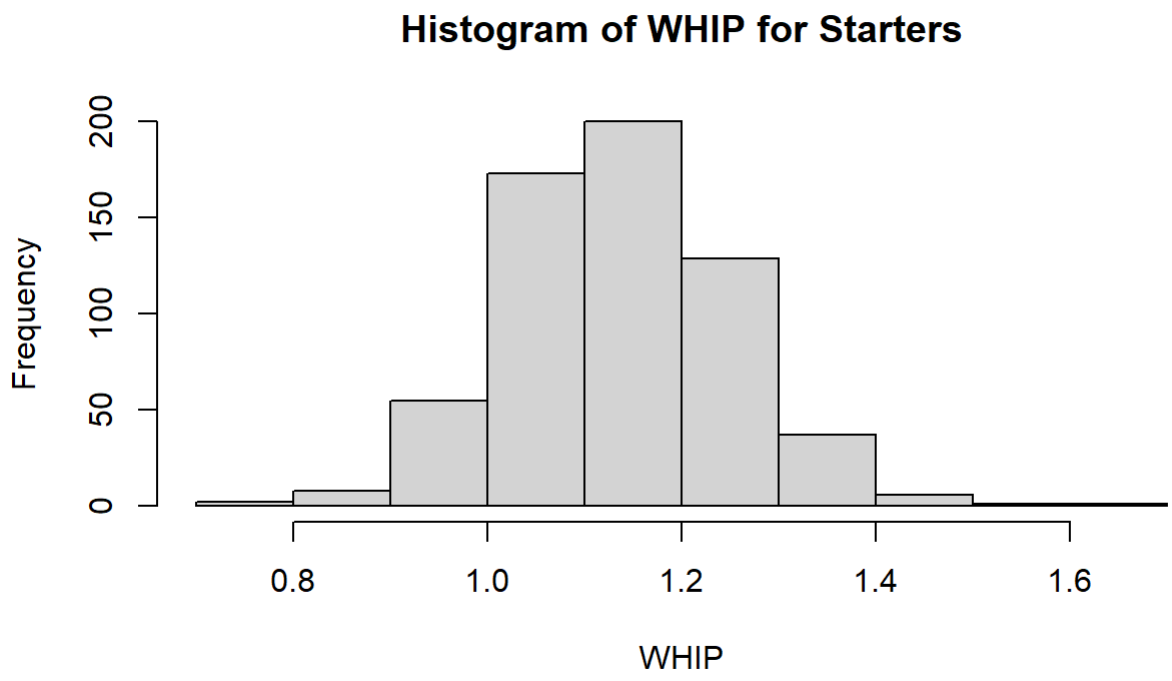

```
hist(startersData$BAOpp,main="Histogram of BAOpp for Starters", xlab="BAOpp")
```




```
hist(startersData$ERA,main="Histogram of ERA for Starters", xlab="ERA")
```

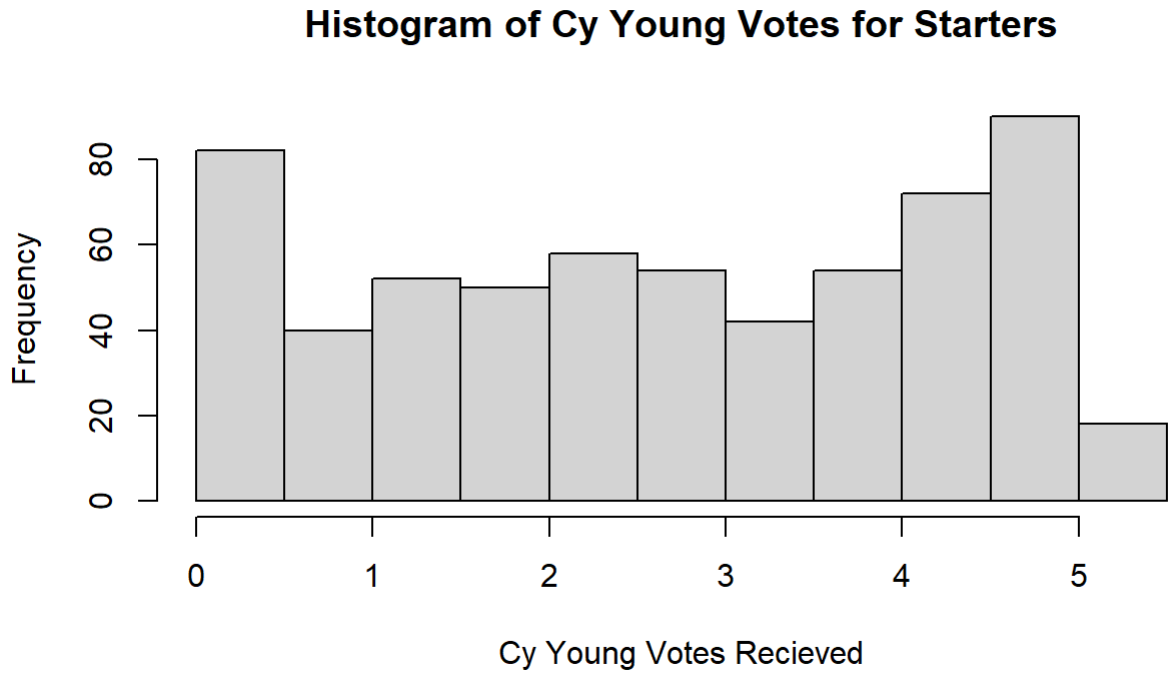



```
hist(startersData$WHIP,main="Histogram of WHIP for Starters", xlab="WHIP")
```



Let's log transform the variables who are not normally distributed.

```
logStarterVotes <- log(startersData$pointsWon)
hist(logStarterVotes,main="Histogram of Cy Young Votes for Starters", xlab="Cy Young Votes Recie
ved")
```



Multicollinearity might be an issue in this analysis. Lets look at the correlation matrix for starters subset.

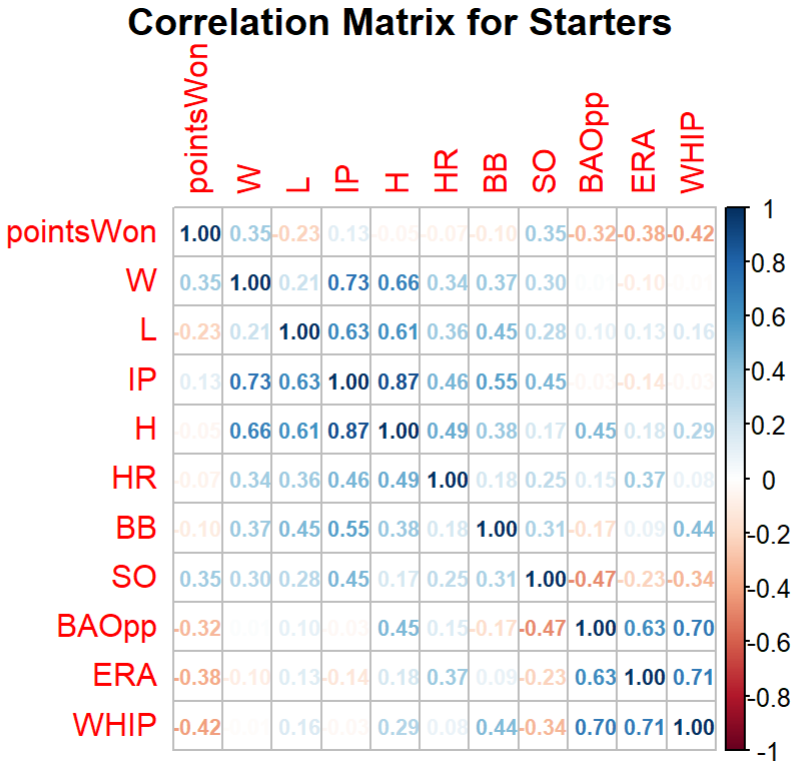
```
summary(startersData)
```

```
##    pointsWon            W               L               IP       
##  Min.   :  1.00   Min.   : 1.00   Min.   : 0.000   Min.   : 38.67  
##  1st Qu.:  3.00   1st Qu.:15.00   1st Qu.: 6.000   1st Qu.:208.00  
##  Median : 16.00   Median :18.00   Median : 8.000   Median :231.50  
##  Mean   : 40.25   Mean   :17.45   Mean   : 8.418   Mean   :230.00  
##  3rd Qu.: 65.25   3rd Qu.:20.00   3rd Qu.:10.000   3rd Qu.:256.67  
##  Max.   :224.00   Max.   :27.00   Max.   :20.000   Max.   :376.67  
##        H               HR              BB               SO       
##  Min.   : 40.0   Min.   : 1.00   Min.   :  6.00   Min.   : 18.0  
##  1st Qu.:174.0   1st Qu.:13.00   1st Qu.: 47.00   1st Qu.:141.0  
##  Median :201.0   Median :18.00   Median : 62.00   Median :179.0  
##  Mean   :198.5   Mean   :17.86   Mean   : 63.52   Mean   :180.8  
##  3rd Qu.:226.0   3rd Qu.:22.00   3rd Qu.: 77.00   3rd Qu.:213.2  
##  Max.   :381.0   Max.   :41.00   Max.   :204.00   Max.   :383.0  
##      BAOpp             ERA             WHIP      
##  Min.   :0.1670   Min.   :1.280   Min.   :0.72  
##  1st Qu.:0.2180   1st Qu.:2.580   1st Qu.:1.06  
##  Median :0.2325   Median :2.940   Median :1.14  
##  Mean   :0.2322   Mean   :2.944   Mean   :1.14  
##  3rd Qu.:0.2460   3rd Qu.:3.260   3rd Qu.:1.22  
##  Max.   :0.2980   Max.   :5.590   Max.   :1.66
```

```
library(corrplot)
matrix<-cor(startersData)
head(round(matrix,2))
```

```
##           pointsWon    W     L   IP     H    HR    BB   SO BAOpp   ERA  WHIP
## pointsWon      1.00 0.35 -0.23 0.13 -0.05 -0.07 -0.10 0.35 -0.32 -0.38 -0.42
## W              0.35 1.00  0.21 0.73  0.66  0.34  0.37 0.30  0.01 -0.10 -0.01
## L             -0.23 0.21  1.00 0.63  0.61  0.36  0.45 0.28  0.10  0.13  0.16
## IP             0.13 0.73  0.63 1.00  0.87  0.46  0.55 0.45 -0.03 -0.14 -0.03
## H             -0.05 0.66  0.61 0.87  1.00  0.49  0.38 0.17  0.45  0.18  0.29
## HR            -0.07 0.34  0.36 0.46  0.49  1.00  0.18 0.25  0.15  0.37  0.08
```

```
corrplot(matrix, method="number", title="Correlation Matrix for Starters", mar=c(0,0,1,0),numbe
r.cex = 0.70)
```



Lets keep this in mind when creating the final model. For now, lets fit an initial model, run the proper diagnostics to check the assumptions for a linear regression model, and determine if any other transformations are necessary

Declare other variables.

```
startersW <- startersData$W
startersL <- startersData$L
startersIP <- startersData$IP
startersH <- startersData$H
startersHR <- startersData$HR
startersBB <- startersData$BB
startersSO <- startersData$SO
startersBAOpp <- startersData$BAOpp
startersERA <- startersData$ERA
startersWHIP <- startersData$WHIP
```

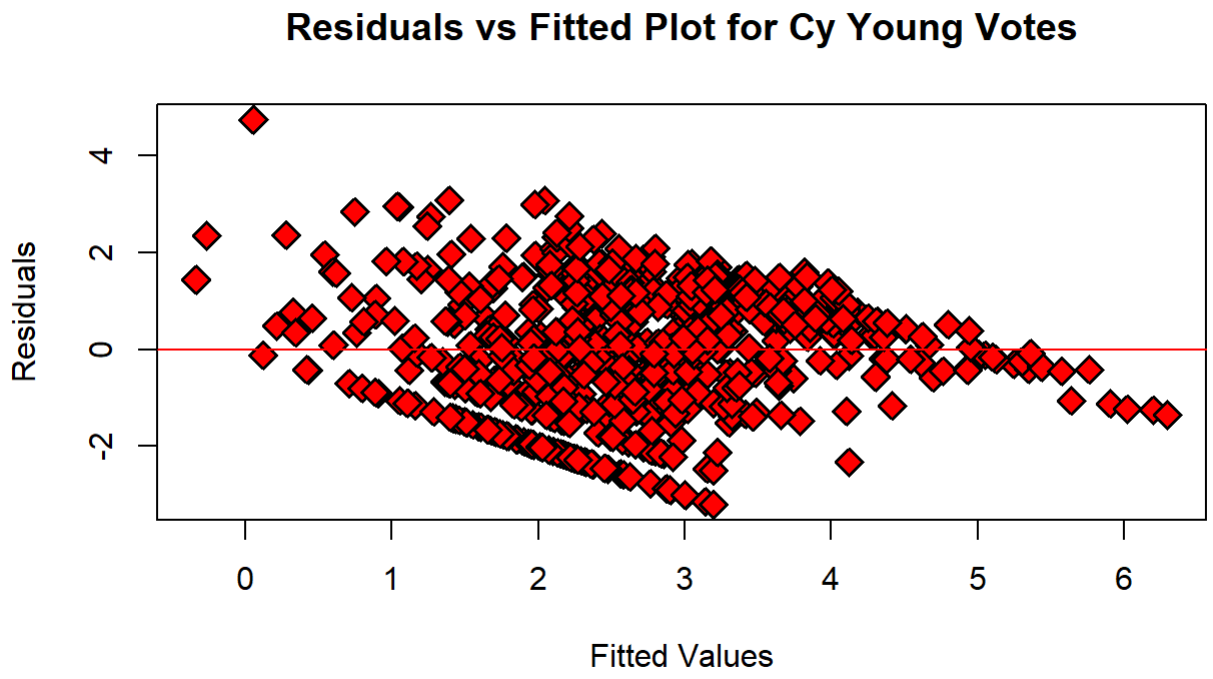
Fit an initial model with each variable from the subset.

```
startersFull <- lm(logStarterVotes ~ startersW + startersL + startersIP + startersH + startersHR
+
                     startersBB + startersSO + startersBAOpp + startersERA + startersWHIP)
summary(startersFull)
```

```
##
## Call:
## lm(formula = logStarterVotes ~ startersW + startersL + startersIP +
##     startersH + startersHR + startersBB + startersSO + startersBAOpp +
##     startersERA + startersWHIP)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1948 -0.9669  0.0428  0.9943  4.7327
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.285833   2.579038  -1.662 0.097075 .
## startersW      0.211830   0.023612   8.971  < 2e-16 ***
## startersL     -0.087823   0.026255  -3.345 0.000874 ***
## startersIP     0.028506   0.010243   2.783 0.005554 **
## startersH     -0.035999   0.010390  -3.465 0.000569 ***
## startersHR    -0.032029   0.011591  -2.763 0.005897 **
## startersBB    -0.014097   0.010799  -1.305 0.192242
## startersSO     0.006460   0.001242   5.200 2.74e-07 ***
## startersBAOpp 29.339474  13.917377   2.108 0.035434 *
## startersERA   -0.223825   0.180606  -1.239 0.215719
## startersWHIP  -1.116372   2.630760  -0.424 0.671460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.31 on 601 degrees of freedom
## Multiple R-squared:  0.3893, Adjusted R-squared:  0.3791
## F-statistic: 38.31 on 10 and 601 DF,  p-value: < 2.2e-16
```

The first assumption we will test is the homogeneity of errors assumption. Plot the residuals against the predicted Cy Young Votes.

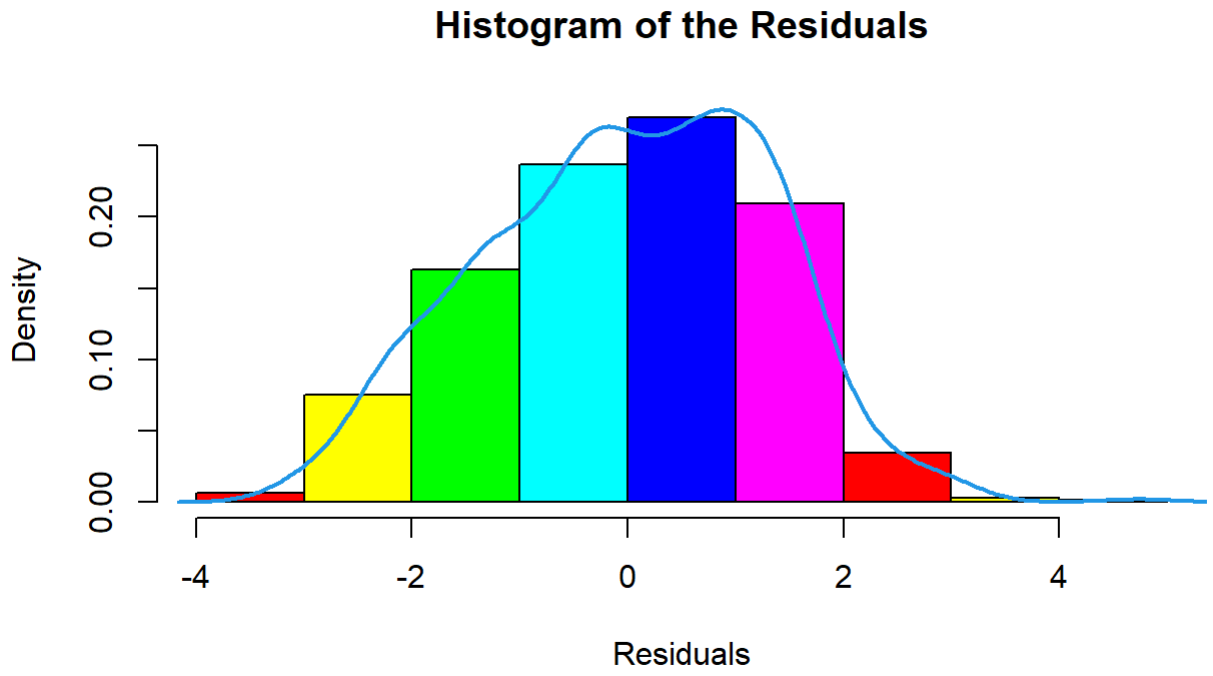
```
plot(startersFull$fitted, startersFull$residuals,main="Residuals vs Fitted Plot for Cy Young Vot
es", xlab="Fitted Values",ylab="Residuals", pch=23,bg="red",cex=1.5,lwd=1.5)
abline(h=0,col="red")
```



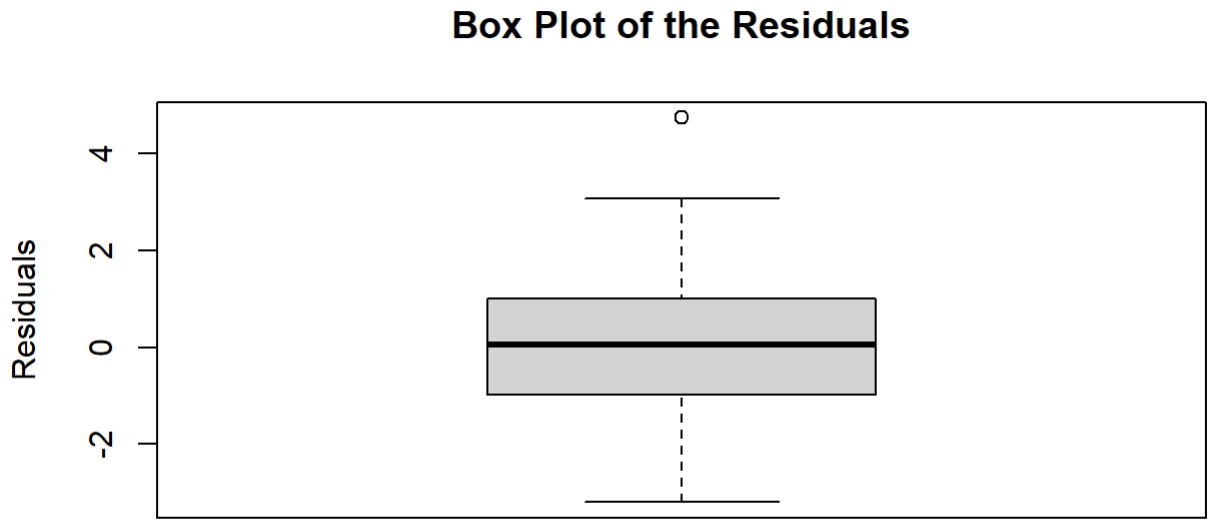
There should be no fanning effect in this graph. Since there is a fanning effect, we should be concerned about the homogeneity of errors assumption.

Lets look at a QQ-plot, boxplot and histogram of the residuals with normal curve.

```
res=startersFull$residuals
hist(res, prob = TRUE, main="Histogram of the Residuals", xlab="Residuals",ylab="Density", col=r
ainbow(6))
lines(density(res), col = 4, lwd = 2)
```



```
boxplot(res, main="Box Plot of the Residuals", ylab="Residuals")
```




```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.1
```

```
## Loading required package: carData
```

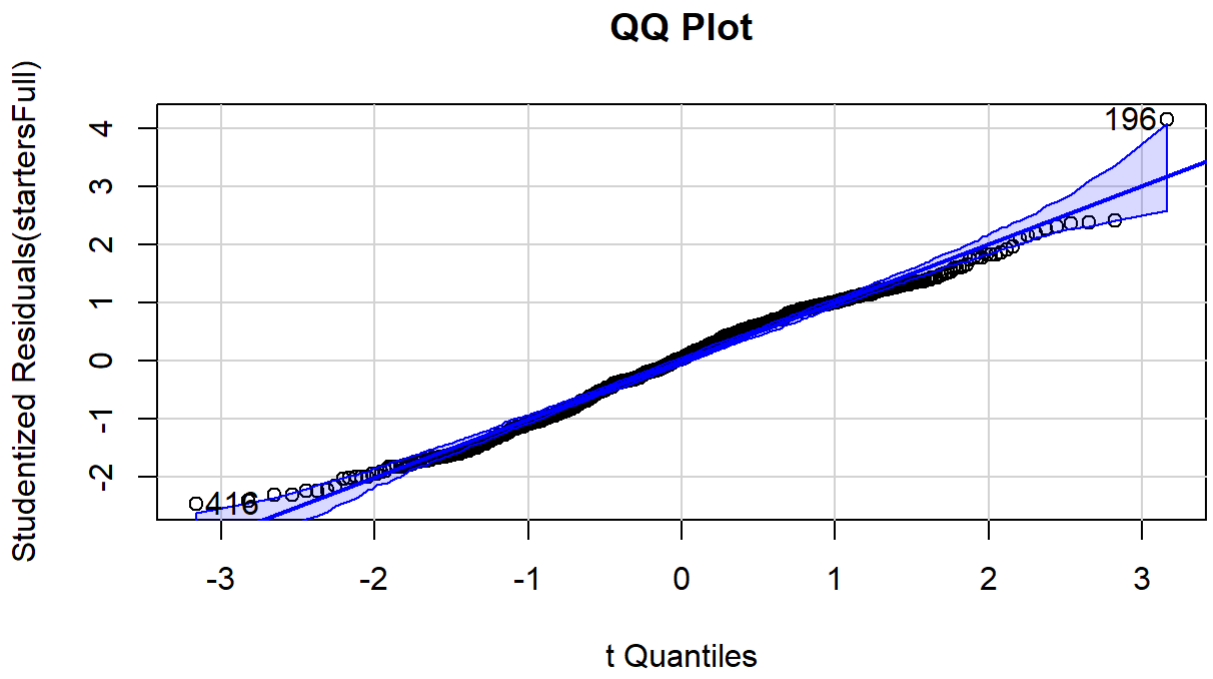
```
## Warning: package 'carData' was built under R version 4.2.1
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
qqPlot(startersFull, id.n=5, main='QQ Plot')
```




```
## [1] 196 416
```

The plots of the residuals look approximately normal, which is what we are looking for. Lets use the residuals from this model to remove outliers and extreme values and run another model.

Remove outliers:

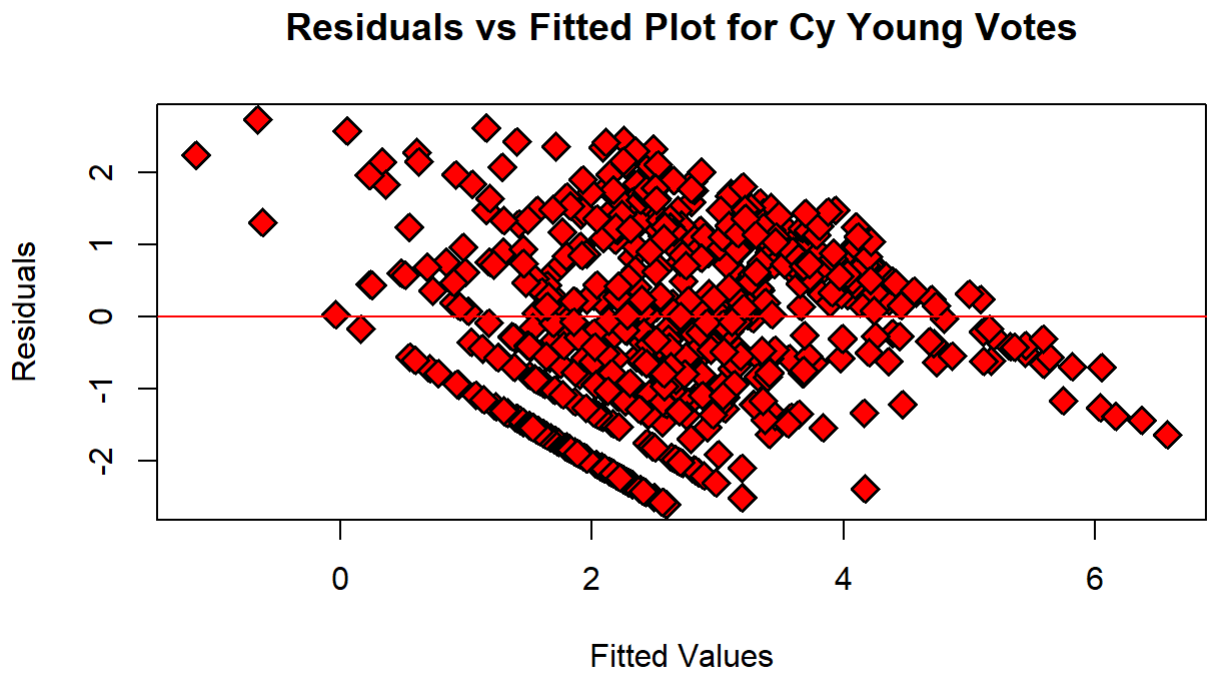
```
rstandard<-rstandard(startersFull)
nonoutlierdf<-data.frame(logStarterVotes,startersW,startersL,startersIP,startersH,startersHR,sta
rtersBB,startersSO,
                         startersBAOpp,startersERA,startersWHIP,rstandard)
nonoutlierdf<-nonoutlierdf[!(nonoutlierdf$rstandard >= 2 | nonoutlierdf$rstandard <= -2), ]
```

Create new model without the outliers.

```
startersNoOutlier <- lm(logStarterVotes ~ startersW + startersL + startersIP + startersH + start
ersHR +
                      startersBB + startersSO + startersBAOpp + startersERA + startersWHIP, data
= nonoutlierdf)
summary(startersNoOutlier)
```

```
##
## Call:
## lm(formula = logStarterVotes ~ startersW + startersL + startersIP +
##     startersH + startersHR + startersBB + startersSO + startersBAOpp +
##     startersERA + startersWHIP, data = nonoutlierdf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59419 -0.89044  0.08009  0.92217  2.73161
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.700252   2.640852  -0.644 0.519941
## startersW      0.233752   0.022232  10.514  < 2e-16 ***
## startersL     -0.088514   0.024489  -3.614 0.000327 ***
## startersIP     0.017593   0.010572   1.664 0.096620 .
## startersH     -0.026577   0.010451  -2.543 0.011246 *
## startersHR    -0.030579   0.010904  -2.805 0.005208 **
## startersBB    -0.004286   0.010822  -0.396 0.692229
## startersSO     0.006621   0.001161   5.705 1.86e-08 ***
## startersBAOpp 29.704330  13.048106   2.277 0.023177 *
## startersERA   -0.222168   0.174343  -1.274 0.203059
## startersWHIP  -3.827829   2.662373  -1.438 0.151042
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.207 on 582 degrees of freedom
## Multiple R-squared:  0.4658, Adjusted R-squared:  0.4567
## F-statistic: 50.76 on 10 and 582 DF,  p-value: < 2.2e-16
```

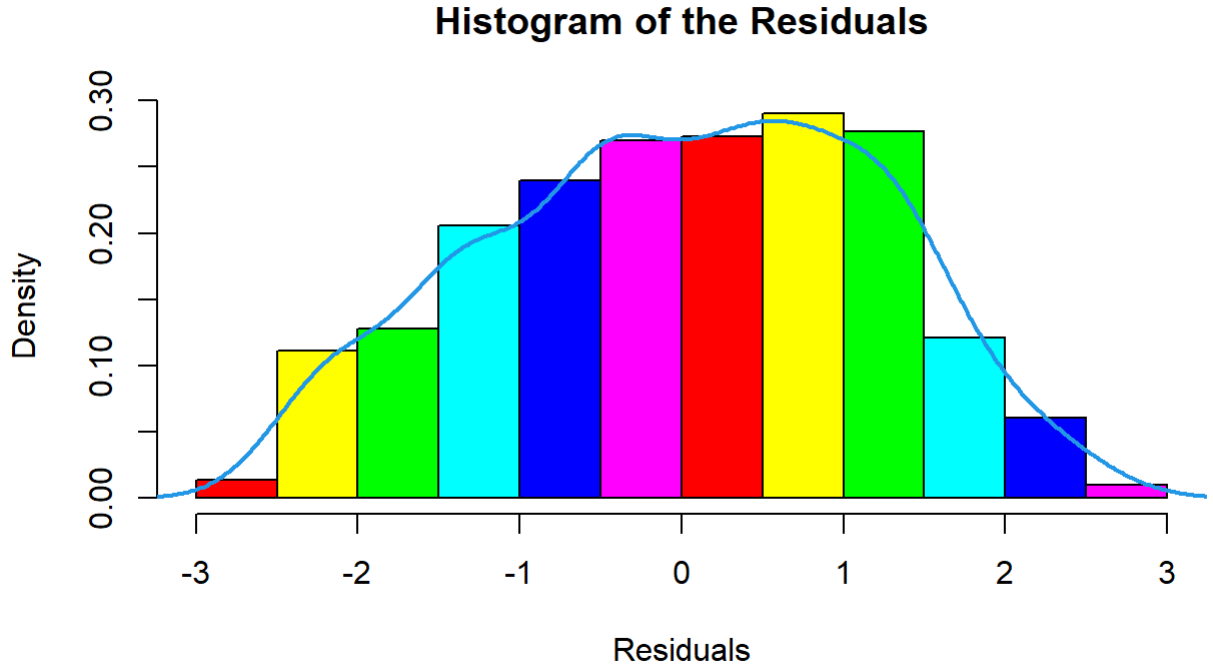
```
plot(startersNoOutlier$fitted, startersNoOutlier$residuals,main="Residuals vs Fitted Plot for Cy Young Votes", xlab="Fitted Values",ylab="Residuals", pch=23,bg="red",cex=1.5,lwd=1.5)
abline(h=0,col="red")
```



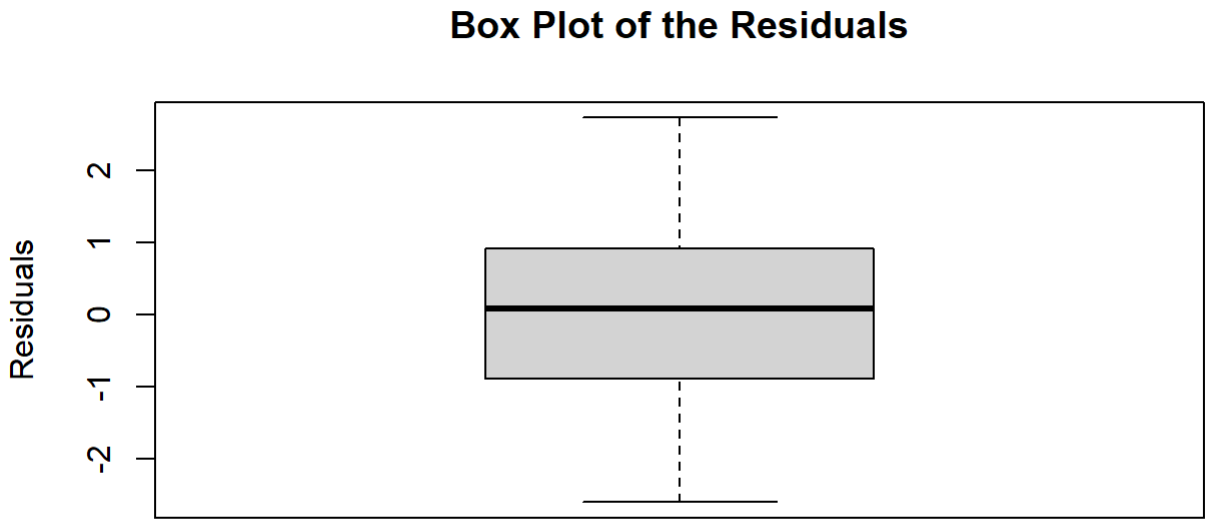
This residuals vs fitted plot looks better after removing outliers.

Lets look at a QQ-plot, boxplot and histogram of the residuals with normal curve.

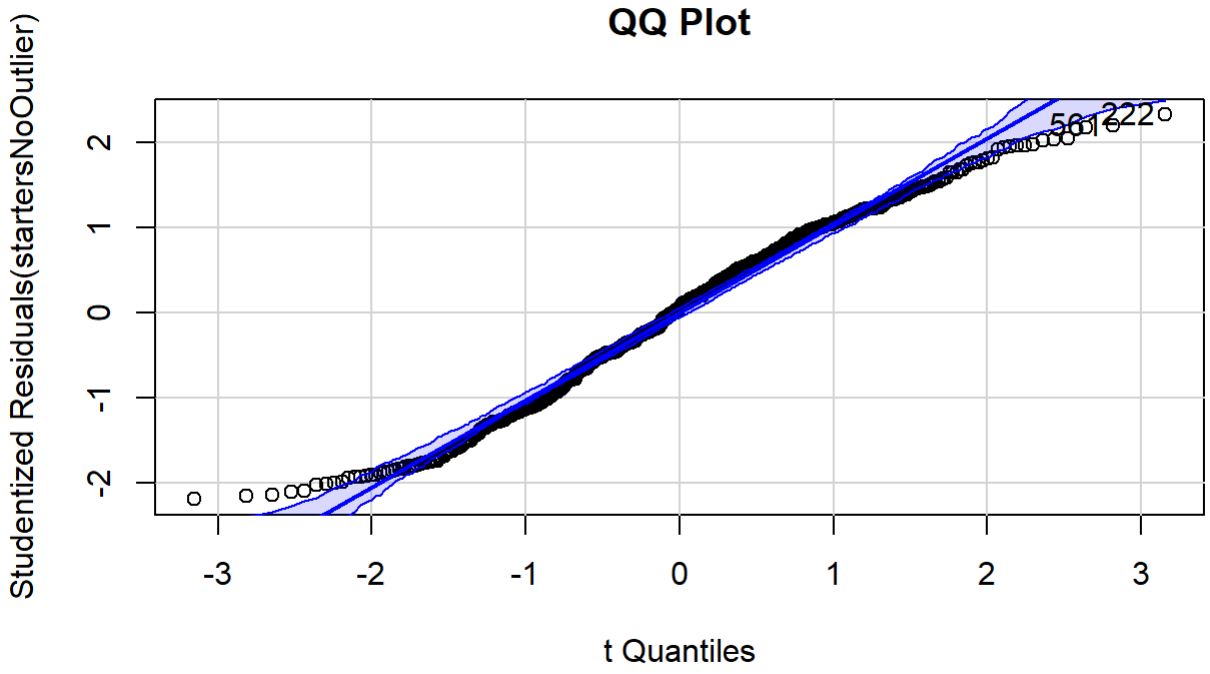
```
res=startersNoOutlier$residuals
hist(res, prob = TRUE, main="Histogram of the Residuals", xlab="Residuals",ylab="Density", col=r
ainbow(6))
lines(density(res), col = 4, lwd = 2)
```



```
boxplot(res, main="Box Plot of the Residuals", ylab="Residuals")
```




```
library(car)
qqPlot(startersNoOutlier, id.n=5, main='QQ Plot')
```




```
## 222 561
## 212 542
```

Now, lets make a final model by removing the insignificant variables from the model. We will do this by using the model selection criteria provided from the ‘leaps’ library.

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.2.2
```

```
startersDiagnostic <- regsubsets(logStarterVotes ~ startersW + startersL + startersIP + starters
H + startersHR +
                     startersBB + startersSO + startersBAOpp + startersERA + startersWHIP, data
= nonoutlierdf)
reg_summary<-summary(startersDiagnostic)
names(reg_summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

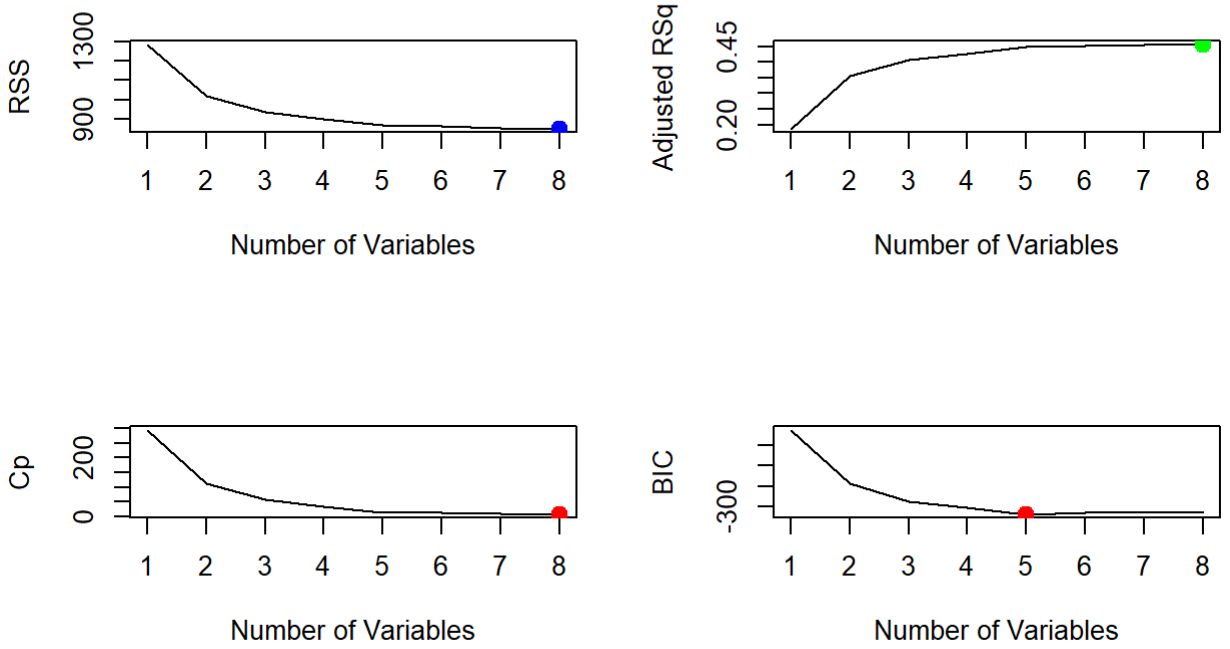
```
reg_summary$which
```

```
##   (Intercept) startersW startersL startersIP startersH startersHR startersBB
## 1        TRUE     FALSE     FALSE      FALSE     FALSE      FALSE      FALSE
## 2        TRUE      TRUE     FALSE      FALSE     FALSE      FALSE      FALSE
## 3        TRUE      TRUE      TRUE      FALSE     FALSE      FALSE      FALSE
## 4        TRUE      TRUE      TRUE      FALSE     FALSE      FALSE      FALSE
## 5        TRUE      TRUE      TRUE      FALSE     FALSE       TRUE      FALSE
## 6        TRUE      TRUE      TRUE       TRUE     FALSE       TRUE      FALSE
## 7        TRUE      TRUE      TRUE      FALSE      TRUE       TRUE      FALSE
## 8        TRUE      TRUE      TRUE       TRUE      TRUE       TRUE      FALSE
##   startersSO startersBAOpp startersERA startersWHIP
## 1      FALSE         FALSE       FALSE         TRUE
## 2      FALSE         FALSE       FALSE         TRUE
## 3      FALSE         FALSE       FALSE         TRUE
## 4       TRUE         FALSE       FALSE         TRUE
## 5       TRUE         FALSE       FALSE         TRUE
## 6       TRUE         FALSE       FALSE         TRUE
## 7       TRUE          TRUE       FALSE         TRUE
## 8       TRUE          TRUE       FALSE         TRUE
```

```
par(mfrow = c(2,2))
plot(reg_summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
rss_min<-which.min(reg_summary$rss)
points(rss_min, reg_summary$rss[rss_min],col="blue",cex = 2, pch = 20)
plot(reg_summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
adjr2_max<-which.max(reg_summary$adjr2)
points(adjr2_max, reg_summary$adjr2[adjr2_max],col="green",cex = 2, pch = 20)

plot(reg_summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
cp_min = which.min(reg_summary$cp) # 7
points(cp_min, reg_summary$cp[cp_min], col = "red", cex = 2, pch = 20)

plot(reg_summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
bic_min = which.min(reg_summary$bic) # 6
points(bic_min, reg_summary$bic[bic_min], col = "red", cex = 2, pch = 20)
```



Model selection using forward selection.

```
# Both Forward selection
startersForward <- regsubsets(logStarterVotes ~ startersW + startersL + startersIP + startersH +
startersHR +
                     startersBB + startersSO + startersBAOpp + startersERA + startersWHIP, data
 = nonoutlierdf, nvmax=10, method = 'forward')
summary(startersForward)
```

```
## Subset selection object
## Call: regsubsets.formula(logStarterVotes ~ startersW + startersL +
##     startersIP + startersH + startersHR + startersBB + startersSO +
##     startersBAOpp + startersERA + startersWHIP, data = nonoutlierdf,
##     nvmax = 10, method = "forward")
## 10 Variables  (and intercept)
##               Forced in Forced out
## startersW         FALSE      FALSE
## startersL         FALSE      FALSE
## startersIP        FALSE      FALSE
## startersH         FALSE      FALSE
## startersHR        FALSE      FALSE
## startersBB        FALSE      FALSE
## startersSO        FALSE      FALSE
## startersBAOpp     FALSE      FALSE
## startersERA       FALSE      FALSE
## startersWHIP      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##           startersW startersL startersIP startersH startersHR startersBB
## 1  ( 1 )  " "       " "       " "        " "       " "        " "
## 2  ( 1 )  "*"       " "       " "        " "       " "        " "
## 3  ( 1 )  "*"       "*"       " "        " "       " "        " "
## 4  ( 1 )  "*"       "*"       " "        " "       " "        " "
## 5  ( 1 )  "*"       "*"       " "        " "       "*"        " "
## 6  ( 1 )  "*"       "*"       "*"        " "       "*"        " "
## 7  ( 1 )  "*"       "*"       "*"        " "       "*"        " "
## 8  ( 1 )  "*"       "*"       "*"        "*"       "*"        " "
## 9  ( 1 )  "*"       "*"       "*"        "*"       "*"        " "
## 10  ( 1 ) "*"       "*"       "*"        "*"       "*"        "*"
##           startersSO startersBAOpp startersERA startersWHIP
## 1  ( 1 )  " "        " "           " "         "*"
## 2  ( 1 )  " "        " "           " "         "*"
## 3  ( 1 )  " "        " "           " "         "*"
## 4  ( 1 )  "*"        " "           " "         "*"
## 5  ( 1 )  "*"        " "           " "         "*"
## 6  ( 1 )  "*"        " "           " "         "*"
## 7  ( 1 )  "*"        "*"           " "         "*"
## 8  ( 1 )  "*"        "*"           " "         "*"
## 9  ( 1 )  "*"        "*"           "*"         "*"
## 10  ( 1 ) "*"        "*"           "*"         "*"
```

Based off the plots, the recommended number of variables we should keep in the model is 8. Going off the selection table, we will remove BB and ERA from the final model.

Re- name variables and create final predictive model:

```
logStarterVotes <- nonoutlierdf$logStarterVotes
startersW <- nonoutlierdf$startersW
startersL <- nonoutlierdf$startersL
startersIP <- nonoutlierdf$startersIP
startersH <- nonoutlierdf$startersH
startersHR <- nonoutlierdf$startersHR
startersSO <- nonoutlierdf$startersSO
startersBAOpp <- nonoutlierdf$startersBAOpp
startersWHIP <- nonoutlierdf$startersWHIP

startersFinal <- lm(logStarterVotes ~ startersW + startersL + startersIP + startersH + startersH
R + startersSO + startersBAOpp + startersWHIP)
summary(startersFinal)
```

```
##
## Call:
## lm(formula = logStarterVotes ~ startersW + startersL + startersIP +
##     startersH + startersHR + startersSO + startersBAOpp + startersWHIP)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5292 -0.9164  0.0683  0.9221  2.6589
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.961773   2.374627  -0.405 0.685610
## startersW      0.232496   0.022181  10.482  < 2e-16 ***
## startersL     -0.094125   0.024078  -3.909 0.000104 ***
## startersIP     0.016039   0.008444   1.899 0.058006 .
## startersH     -0.024820   0.009692  -2.561 0.010691 *
## startersHR    -0.038161   0.008980  -4.249 2.49e-05 ***
## startersSO     0.006398   0.001146   5.584 3.61e-08 ***
## startersBAOpp 31.117543  10.776057   2.888 0.004025 **
## startersWHIP  -5.353081   0.623313  -8.588  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.206 on 584 degrees of freedom
## Multiple R-squared:  0.4643, Adjusted R-squared:  0.457
## F-statistic: 63.28 on 8 and 584 DF,  p-value: < 2.2e-16
```

Now, lets create the model for relievers.

```
relieverData <- relieverSubset[,c('pointsWon','W','L','IP','H','HR','BB','SO','BAOpp','ERA','WHI
P','SV')]
```

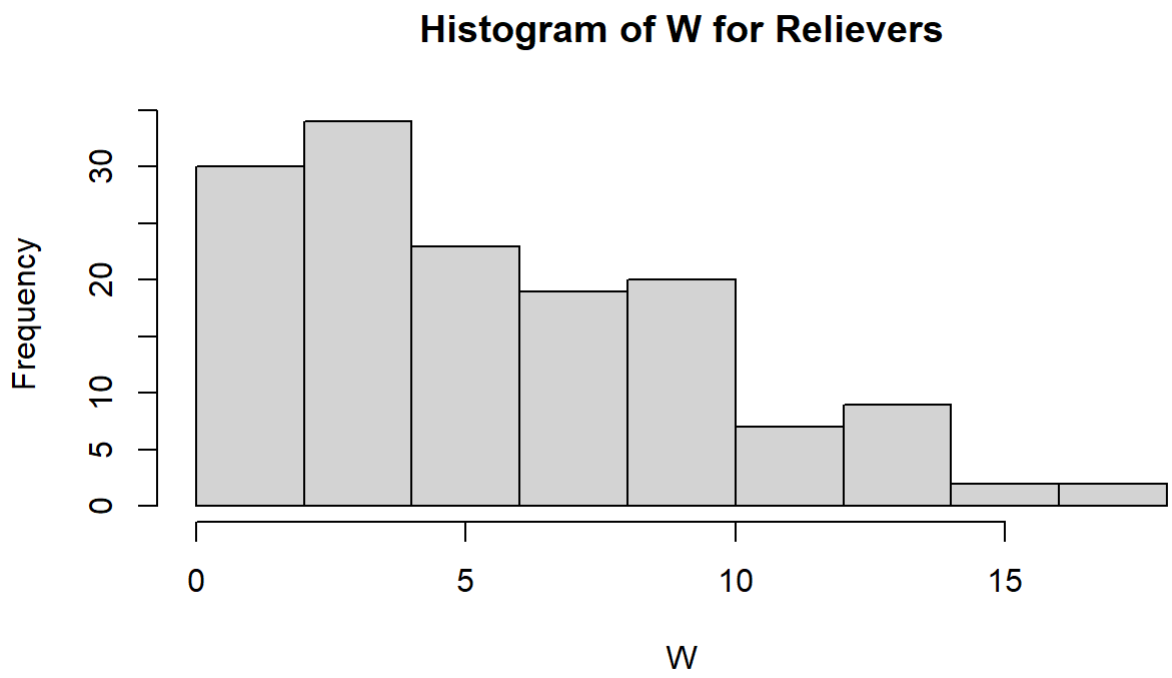
Check if any columns have null values.

```
relieverData %>% summarise_all(~ sum(is.na(.)))
```

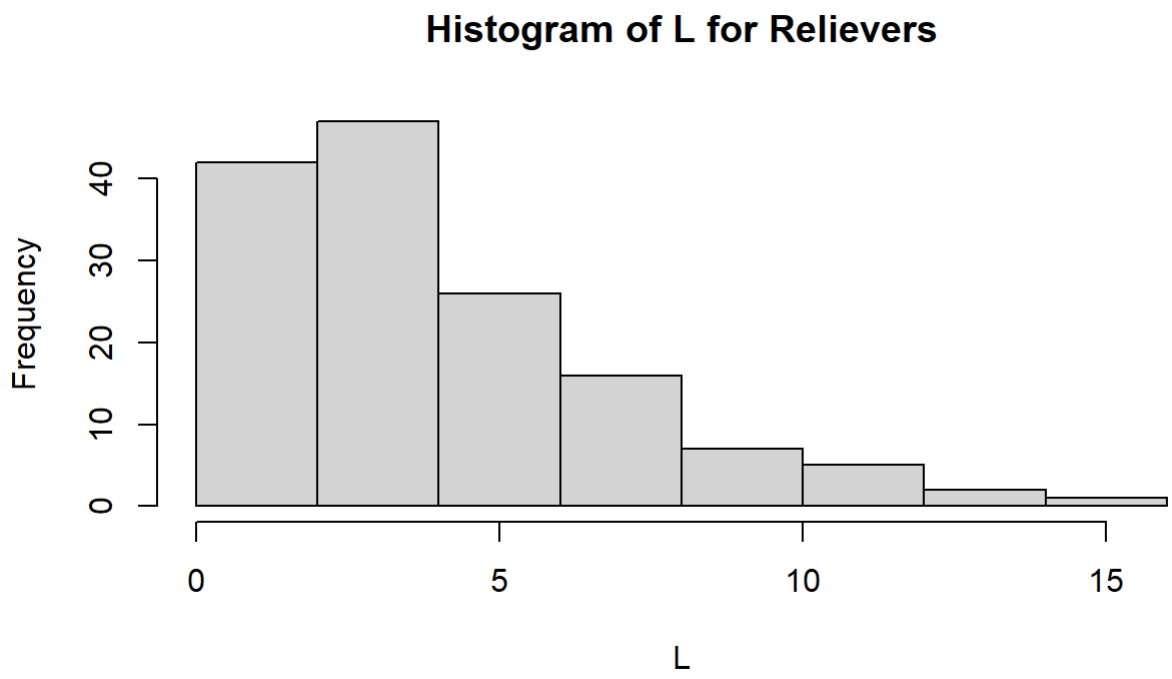
```
## # A tibble: 1 × 12
##   pointsWon     W     L    IP     H    HR    BB    SO BAOpp   ERA  WHIP    SV
##       <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1         0     0     0     0     0     0     0     0     0     0     0     0
```

Lets get an idea of the overall distributions of each variable.

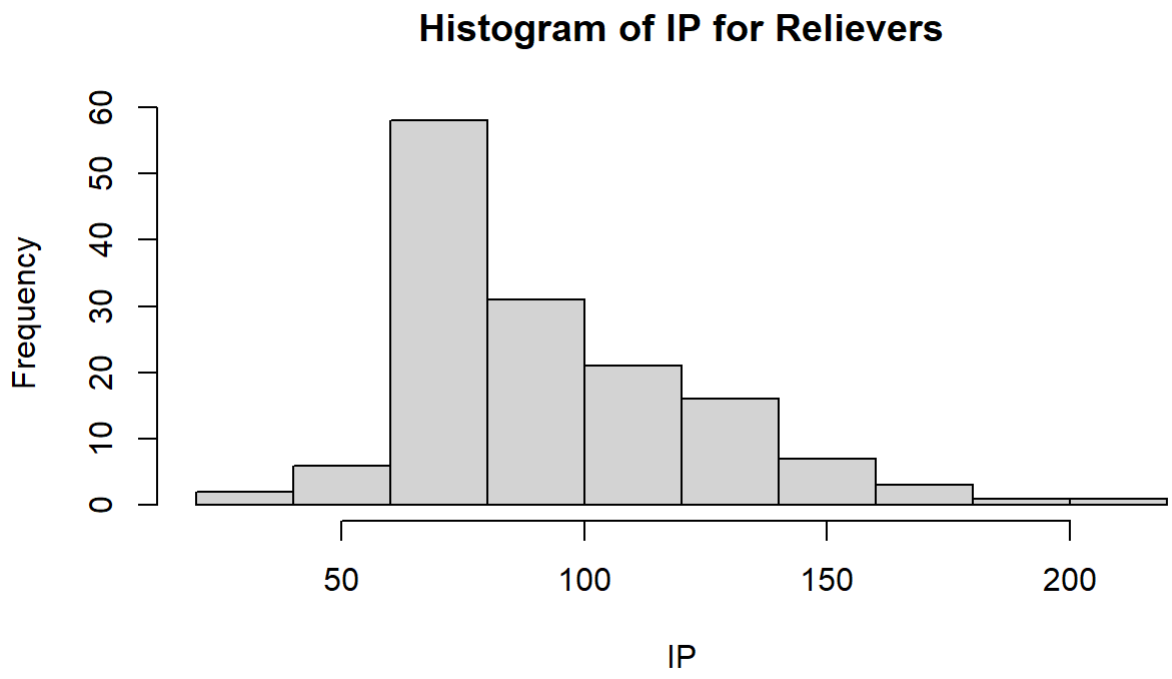
```
hist(relieverData$W,main="Histogram of W for Relievers", xlab="W")
```



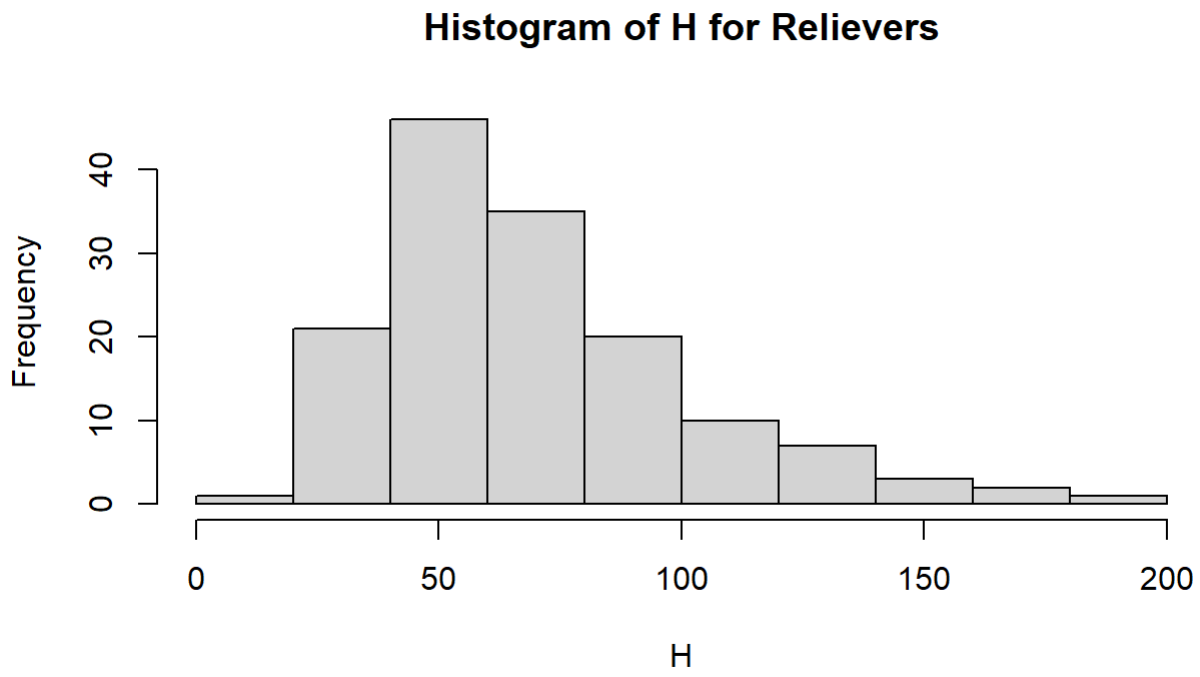

```
hist(relieverData$L,main="Histogram of L for Relievers", xlab="L")
```



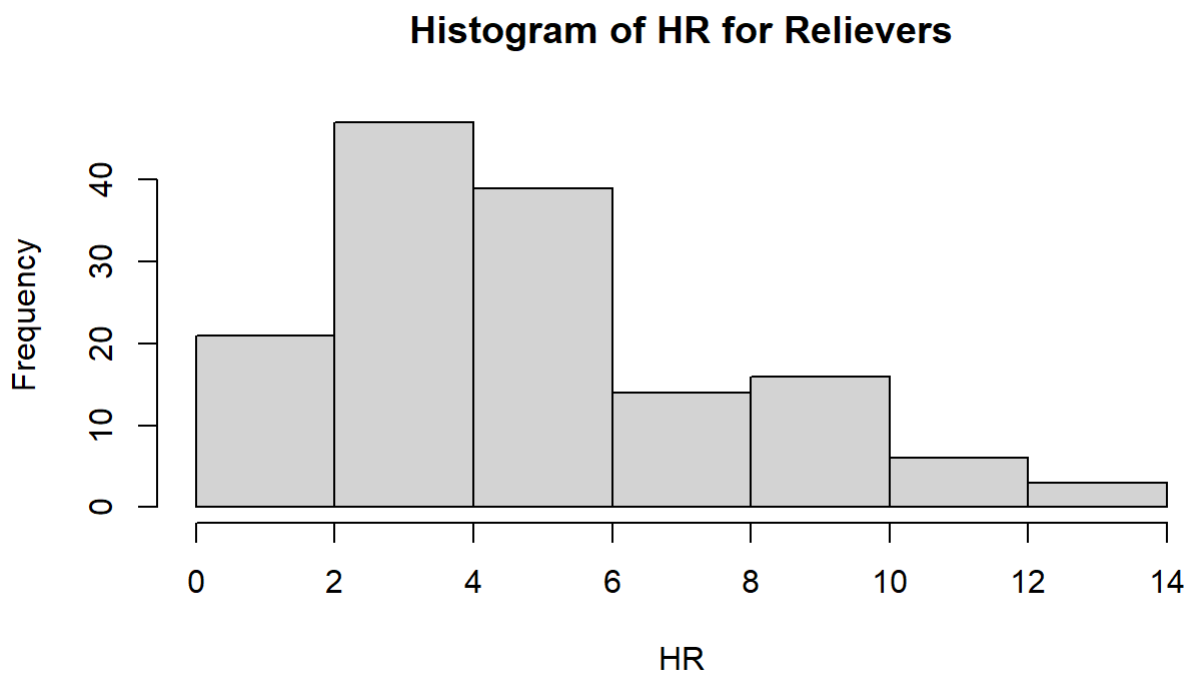

```
hist(relieverData$IP,main="Histogram of IP for Relievers", xlab="IP")
```



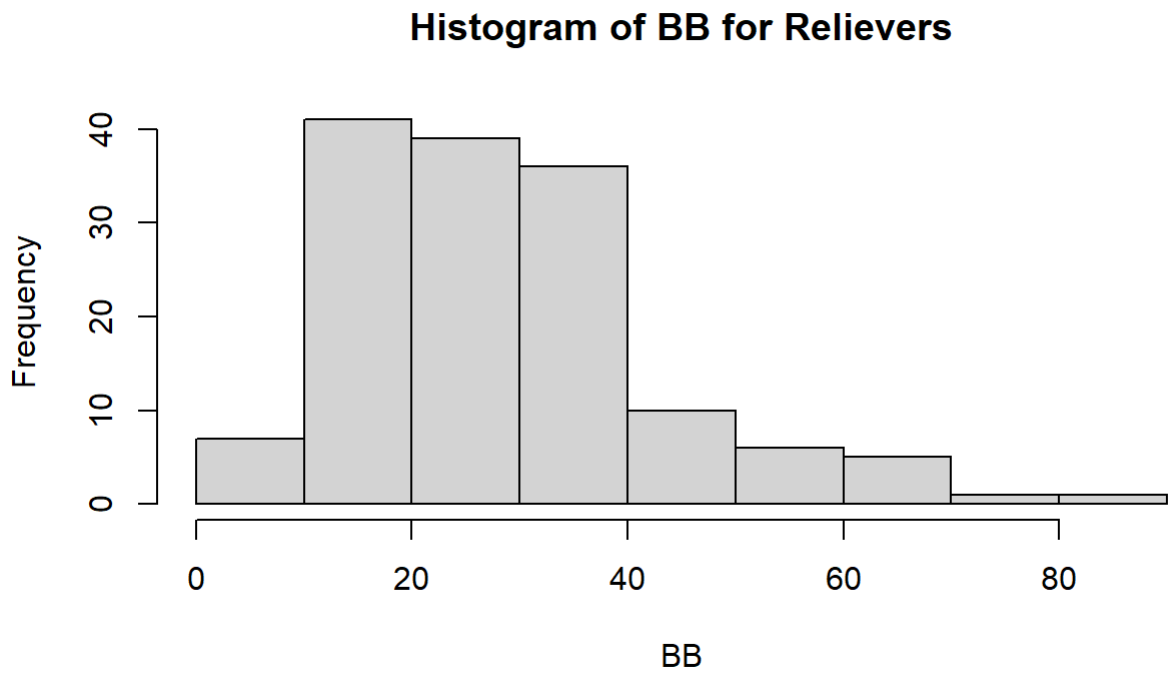

```
hist(relieverData$H,main="Histogram of H for Relievers", xlab="H")
```



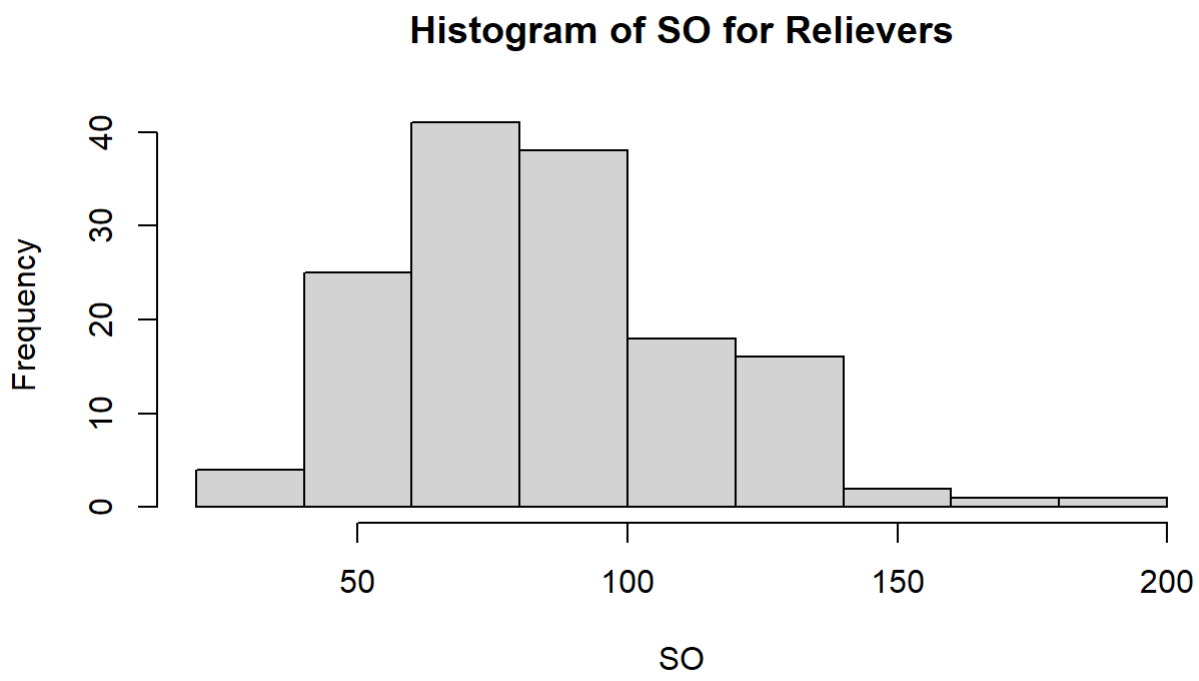

```
hist(relieverData$HR,main="Histogram of HR for Relievers", xlab="HR")
```



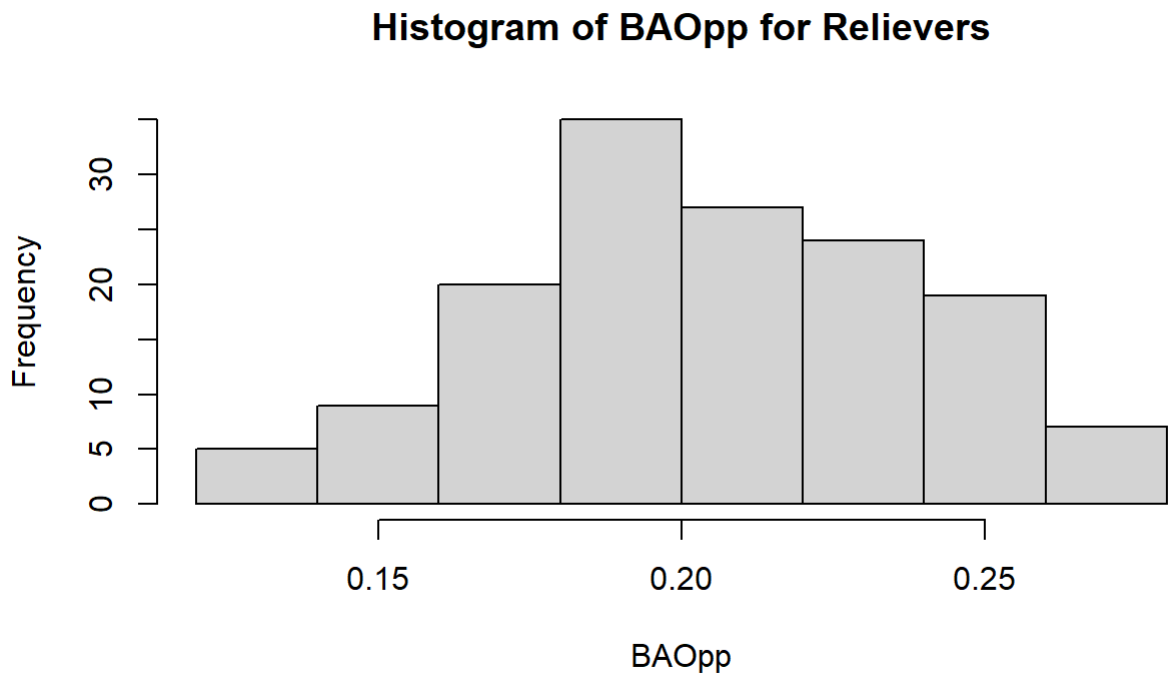

```
hist(relieverData$BB,main="Histogram of BB for Relievers", xlab="BB")
```



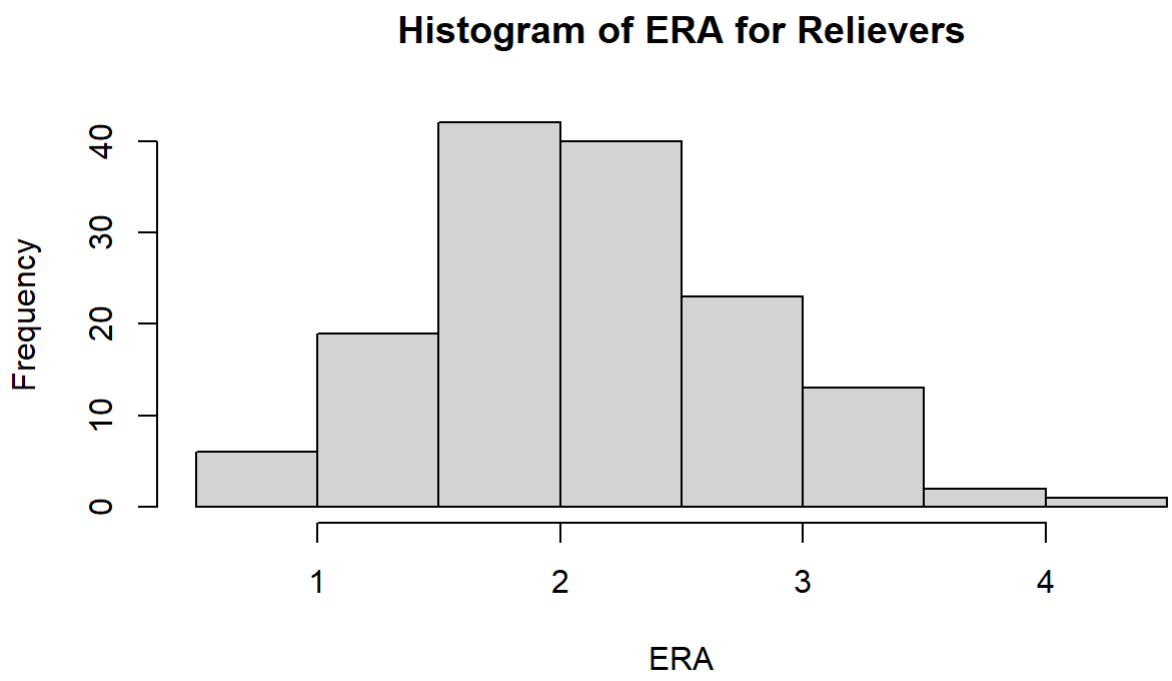

```
hist(relieverData$SO,main="Histogram of SO for Relievers", xlab="SO")
```



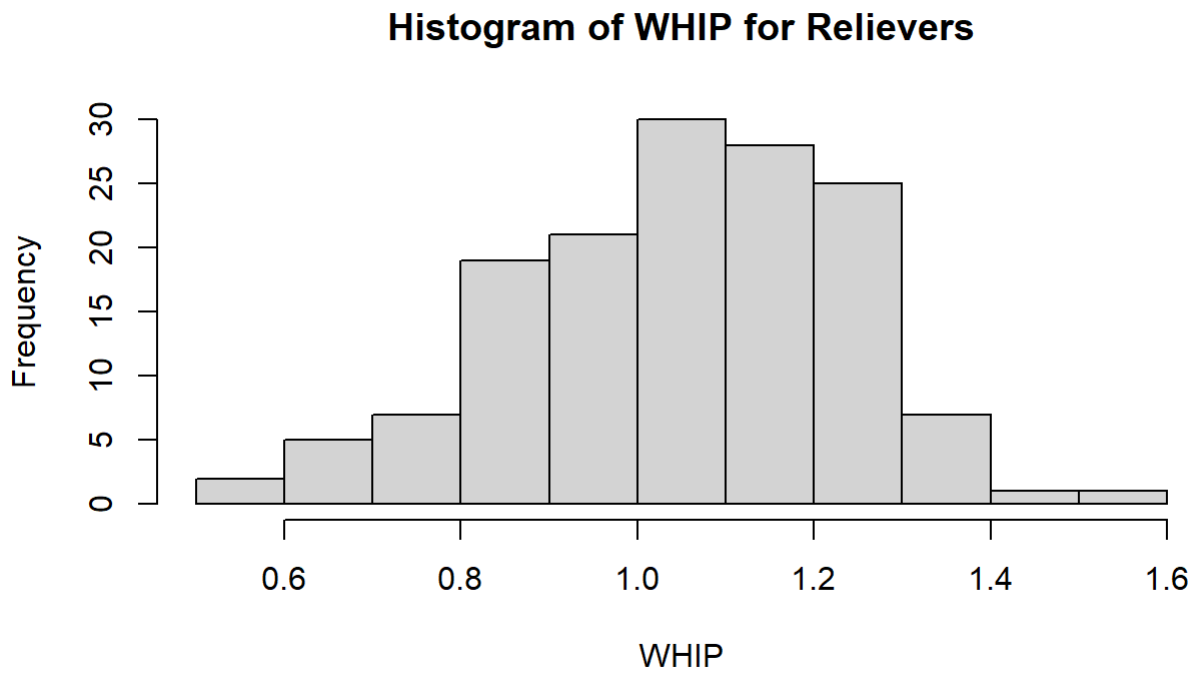

```
hist(relieverData$BAOpp,main="Histogram of BAOpp for Relievers", xlab="BAOpp")
```



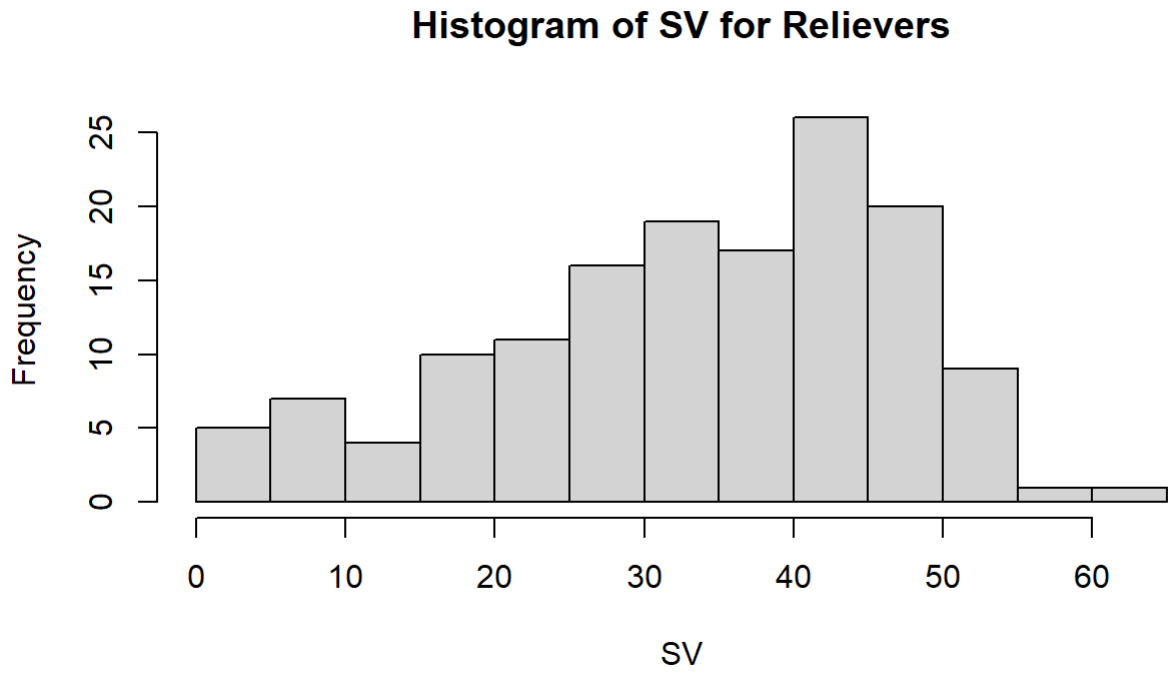

```
hist(relieverData$ERA,main="Histogram of ERA for Relievers", xlab="ERA")
```




```
hist(relieverData$WHIP,main="Histogram of WHIP for Relievers", xlab="WHIP")
```

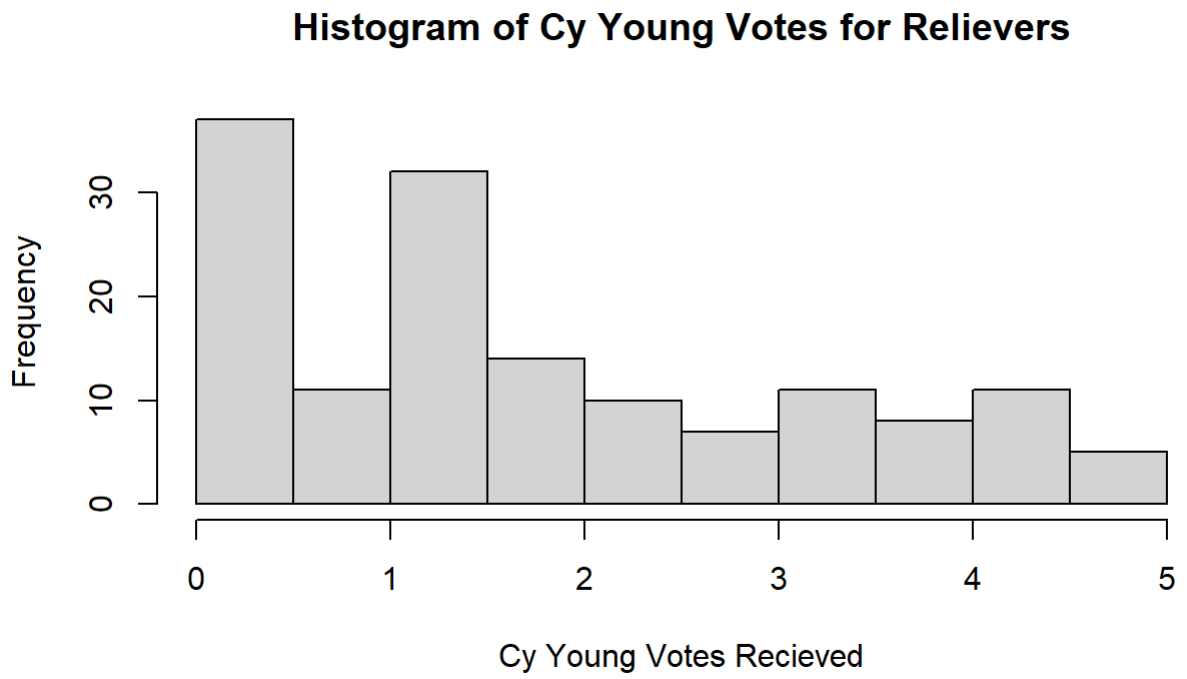



```
hist(relieverData$SV,main="Histogram of SV for Relievers", xlab="SV")
```



Let's log transform the response variable.

```
logRelieverVotes <- log(relieverData$pointsWon)
hist(logRelieverVotes,main="Histogram of Cy Young Votes for Relievers", xlab="Cy Young Votes Rec
ieved")
```



Multicollinearity might be an issue in this analysis. Lets look at the correlation matrix for relievers subset.

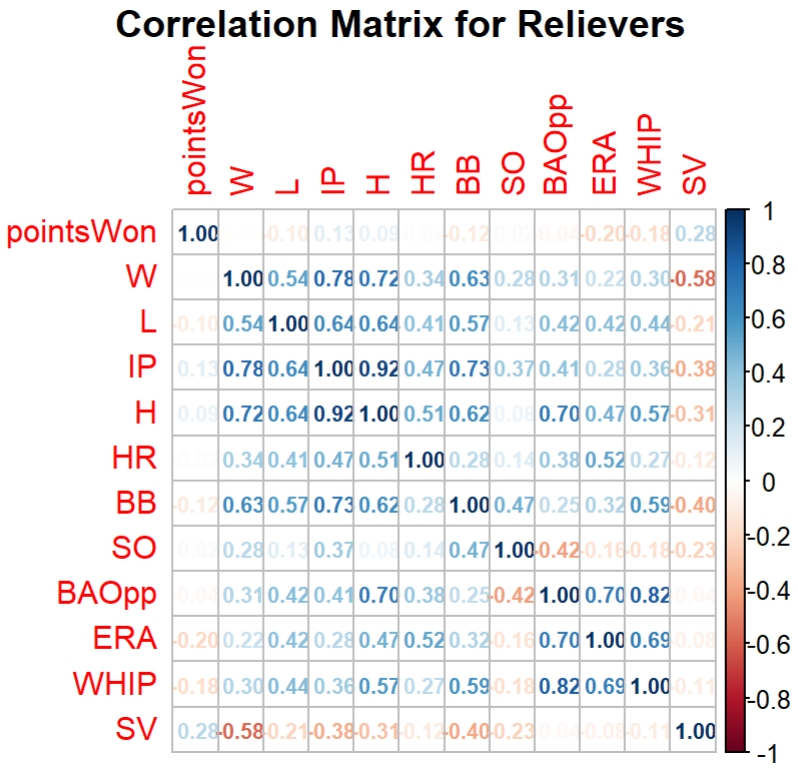
```
summary(relieverData)
```

```
##    pointsWon            W                L                IP        
##  Min.   :  1.00   Min.   : 0.000   Min.   : 0.000   Min.   : 29.00  
##  1st Qu.:  1.25   1st Qu.: 3.000   1st Qu.: 2.000   1st Qu.: 71.67  
##  Median :  4.00   Median : 6.000   Median : 4.000   Median : 83.17  
##  Mean   : 17.09   Mean   : 6.205   Mean   : 4.404   Mean   : 92.33  
##  3rd Qu.: 20.00   3rd Qu.: 9.000   3rd Qu.: 6.000   3rd Qu.:107.33  
##  Max.   :146.00   Max.   :17.000   Max.   :15.000   Max.   :208.33  
##        H                HR               BB              SO        
##  Min.   : 14.00   Min.   : 0.000   Min.   : 2.00   Min.   : 37.00  
##  1st Qu.: 48.00   1st Qu.: 3.000   1st Qu.:18.00   1st Qu.: 67.00  
##  Median : 63.50   Median : 5.000   Median :26.00   Median : 82.50  
##  Mean   : 70.36   Mean   : 5.226   Mean   :28.75   Mean   : 85.27  
##  3rd Qu.: 88.75   3rd Qu.: 7.000   3rd Qu.:35.75   3rd Qu.:101.00  
##  Max.   :191.00   Max.   :14.000   Max.   :84.00   Max.   :182.00  
##      BAOpp             ERA             WHIP             SV        
##  Min.   :0.1260   Min.   :0.540   Min.   :0.550   Min.   : 1.00  
##  1st Qu.:0.1822   1st Qu.:1.653   1st Qu.:0.910   1st Qu.:25.25  
##  Median :0.2045   Median :2.090   Median :1.060   Median :36.00  
##  Mean   :0.2053   Mean   :2.119   Mean   :1.051   Mean   :33.84  
##  3rd Qu.:0.2310   3rd Qu.:2.592   3rd Qu.:1.198   3rd Qu.:45.00  
##  Max.   :0.2800   Max.   :4.480   Max.   :1.510   Max.   :62.00
```

```
library(corrplot)
matrix<-cor(relieverData)
head(round(matrix,2))
```

```
##            pointsWon     W     L   IP    H    HR    BB   SO BAOpp   ERA  WHIP
## pointsWon       1.00 -0.01 -0.10 0.13 0.09 -0.02 -0.12 0.02 -0.04 -0.20 -0.18
## W              -0.01  1.00  0.54 0.78 0.72  0.34  0.63 0.28  0.31  0.22  0.30
## L              -0.10  0.54  1.00 0.64 0.64  0.41  0.57 0.13  0.42  0.42  0.44
## IP              0.13  0.78  0.64 1.00 0.92  0.47  0.73 0.37  0.41  0.28  0.36
## H               0.09  0.72  0.64 0.92 1.00  0.51  0.62 0.08  0.70  0.47  0.57
## HR             -0.02  0.34  0.41 0.47 0.51  1.00  0.28 0.14  0.38  0.52  0.27
##                  SV
## pointsWon  0.28
## W         -0.58
## L         -0.21
## IP        -0.38
## H         -0.31
## HR        -0.12
```

```
corrplot(matrix, method="number", title="Correlation Matrix for Relievers", mar=c(0,0,1,0),numbe
r.cex = 0.70)
```



Lets keep this in mind when creating the final model. For now, lets fit an initial model, run the proper diagnostics to check the assumptions for a linear regression model, and determine if any other transformations are necessary

Declare other variables.

```
relieverW <- relieverData$W
relieverL <- relieverData$L
relieverIP <- relieverData$IP
relieverH <- relieverData$H
relieverHR <- relieverData$HR
relieverBB <- relieverData$BB
relieverSO <- relieverData$SO
relieverBAOpp <- relieverData$BAOpp
relieverERA <- relieverData$ERA
relieverWHIP <- relieverData$WHIP
relieverSV <- relieverData$SV
```

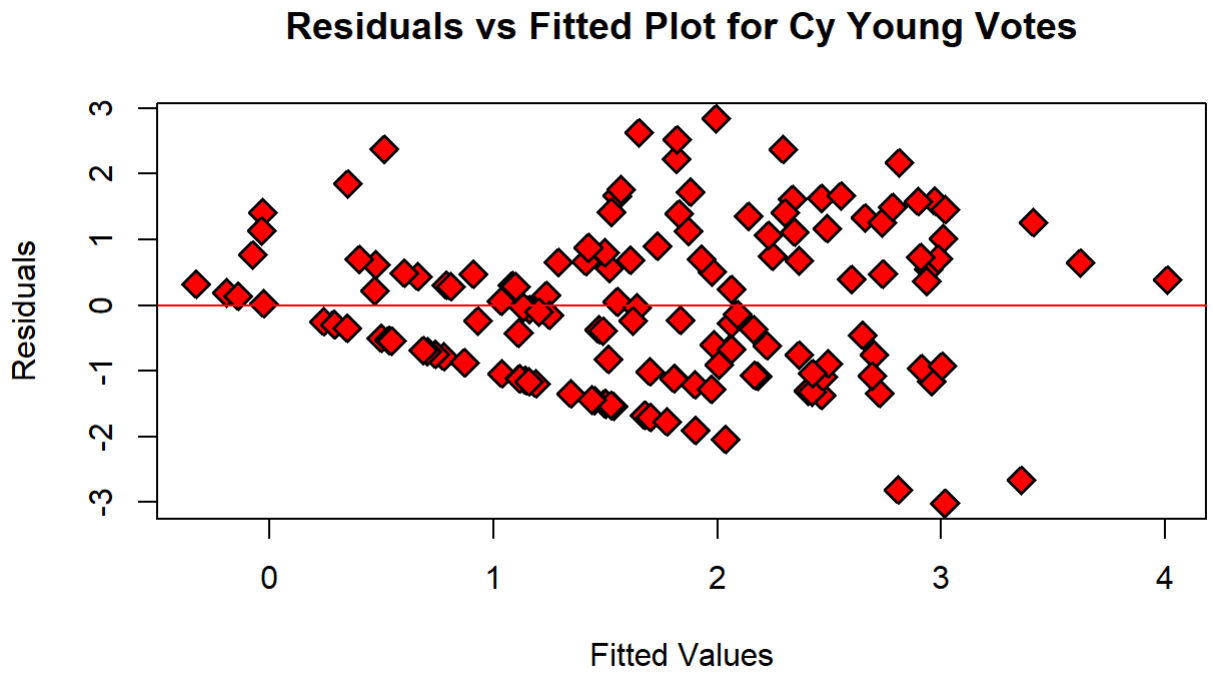
Fit an initial model with each variable from the subset.

```
relieverFull <- lm(logRelieverVotes ~ relieverW + relieverL + relieverIP + relieverH + relieverH
R +
                     relieverBB + relieverSO + relieverBAOpp + relieverERA + relieverWHIP + reli
everSV)
summary(relieverFull)
```

```
##
## Call:
## lm(formula = logRelieverVotes ~ relieverW + relieverL + relieverIP +
##     relieverH + relieverHR + relieverBB + relieverSO + relieverBAOpp +
##     relieverERA + relieverWHIP + relieverSV)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01608 -0.99632 -0.04893  0.76335  2.84119
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.193429   2.057563   0.094  0.92524
## relieverW      0.105577   0.049468   2.134  0.03464 *
## relieverL     -0.146307   0.047657  -3.070  0.00259 **
## relieverIP     0.009882   0.026599   0.372  0.71082
## relieverH      0.013260   0.026020   0.510  0.61117
## relieverHR    -0.007256   0.047888  -0.152  0.87979
## relieverBB    -0.003091   0.030830  -0.100  0.92029
## relieverSO    -0.003290   0.006371  -0.516  0.60646
## relieverBAOpp  1.803643  15.033478   0.120  0.90468
## relieverERA   -0.364682   0.257093  -1.418  0.15837
## relieverWHIP  -1.469118   2.856516  -0.514  0.60789
## relieverSV     0.059640   0.009802   6.085 1.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.243 on 134 degrees of freedom
## Multiple R-squared:  0.3568, Adjusted R-squared:  0.3041
## F-statistic: 6.759 on 11 and 134 DF,  p-value: 5.934e-09
```

The first assumption we will test is the homogeneity of errors assumption. Plot the residuals against the predicted Cy Young Votes.

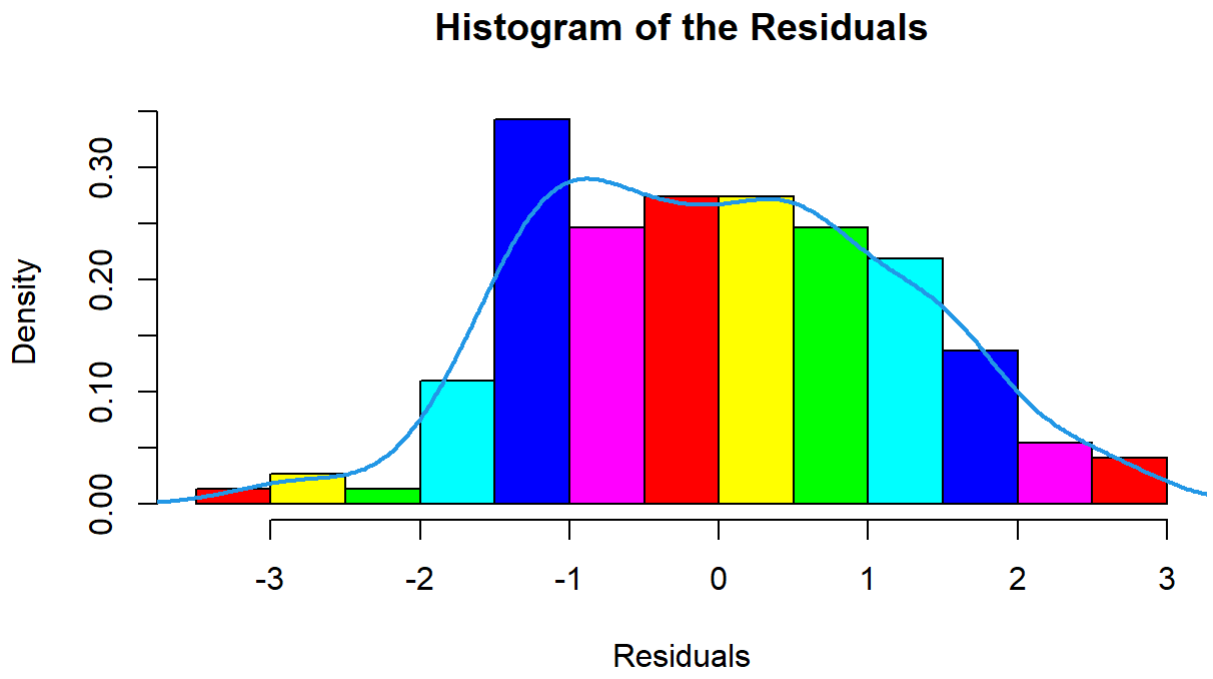
```
plot(relieverFull$fitted, relieverFull$residuals,main="Residuals vs Fitted Plot for Cy Young Vot
es", xlab="Fitted Values",ylab="Residuals", pch=23,bg="red",cex=1.5,lwd=1.5)
abline(h=0,col="red")
```



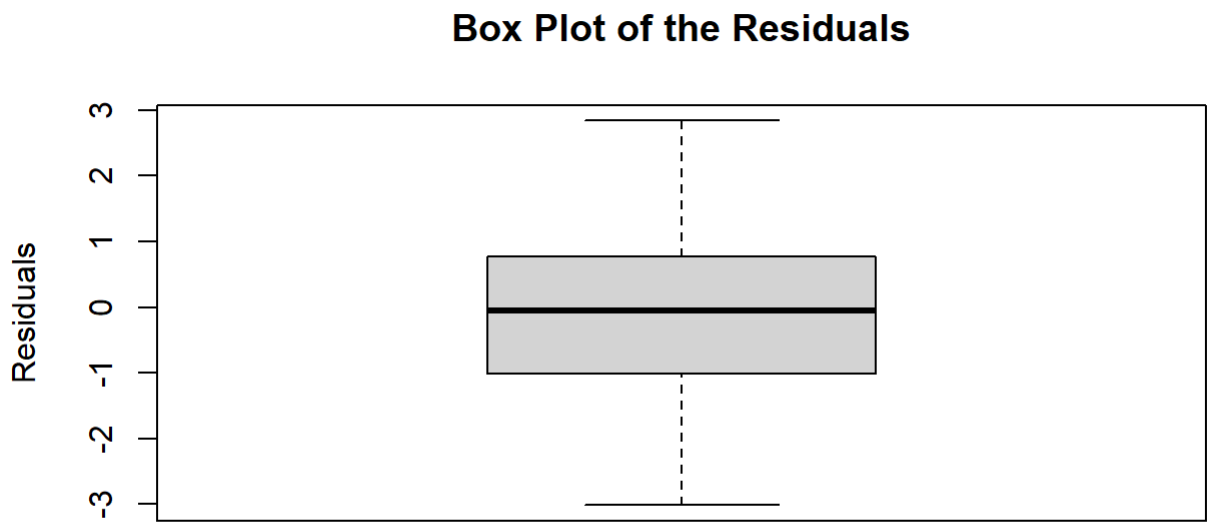
The randomness in this plot is what we are looking for when assessing the homogeneity of errors assumption.

Lets look at a QQ-plot, boxplot and histogram of the residuals with normal curve.

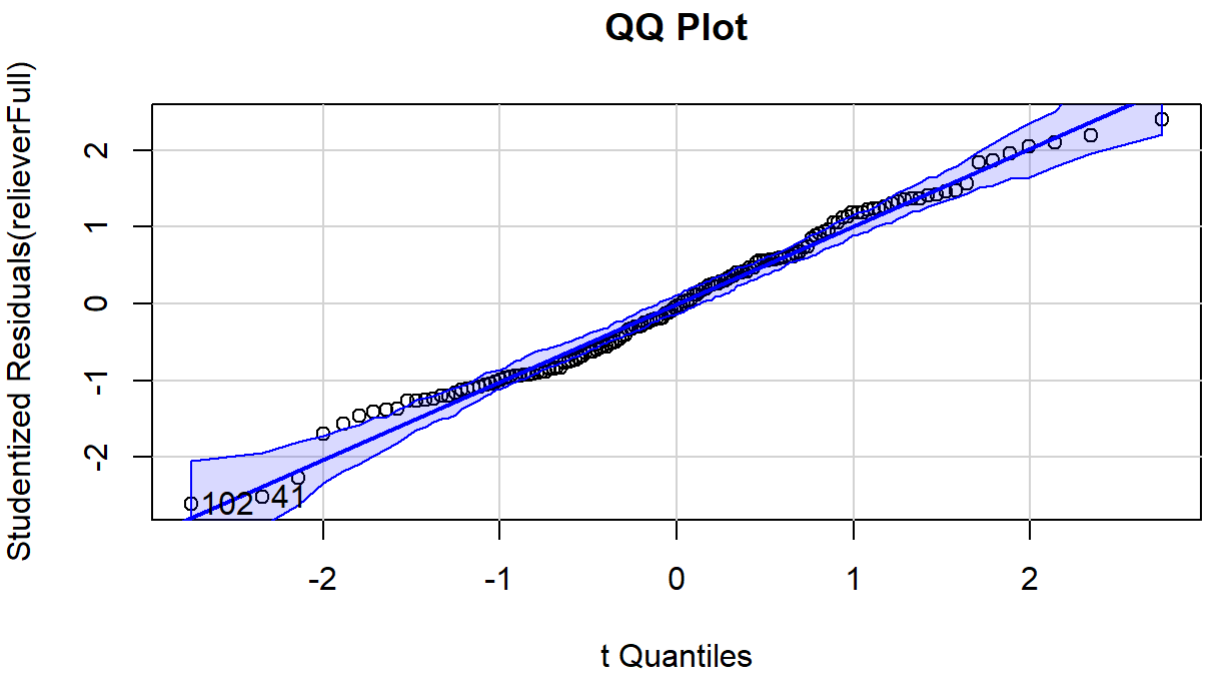
```
res=relieverFull$residuals
hist(res, prob = TRUE, main="Histogram of the Residuals", xlab="Residuals",ylab="Density", col=r
ainbow(6))
lines(density(res), col = 4, lwd = 2)
```




```
boxplot(res, main="Box Plot of the Residuals", ylab="Residuals")
```




```
library(car)
qqPlot(relieverFull, id.n=5, main='QQ Plot')
```




```
## [1]  41 102
```

The plots of the residuals look approximately normal, which is what we are looking for. Lets use the residuals from this model to remove outliers and extreme values and run another model.

Remove outliers:

```
rstandard<-rstandard(relieverFull)
nonoutlierdf<-data.frame(logRelieverVotes,relieverW,relieverL,relieverIP,relieverH,relieverHR,re
lieverBB,relieverSO,
                         relieverBAOpp,relieverERA,relieverWHIP,relieverSV,rstandard)
nonoutlierdf<-nonoutlierdf[!(nonoutlierdf$rstandard >= 2 | nonoutlierdf$rstandard <= -2), ]
```

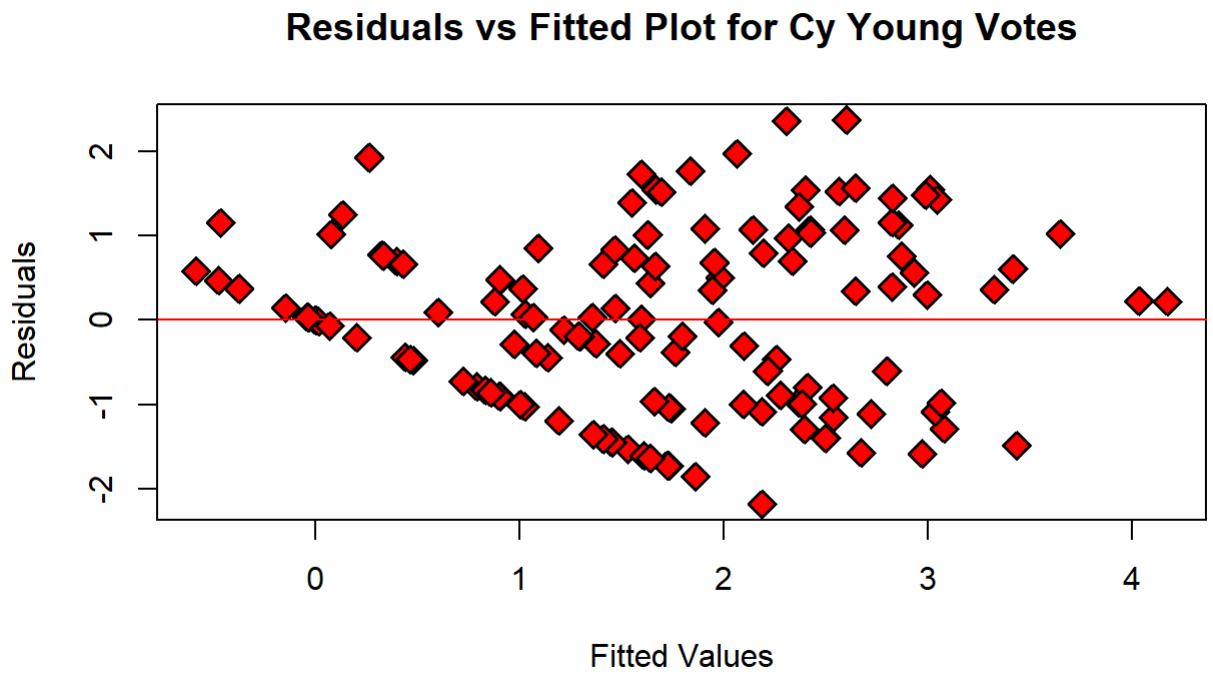
Create new model without the outliers.

```
relieversNoOutlier <- lm(logRelieverVotes ~ relieverW + relieverL + relieverIP + relieverH + rel
ieverHR +
                     relieverBB + relieverSO + relieverBAOpp + relieverERA + relieverWHIP + reli
everSV, data = nonoutlierdf)
summary(relieversNoOutlier)
```

```
##
## Call:
## lm(formula = logRelieverVotes ~ relieverW + relieverL + relieverIP +
##     relieverH + relieverHR + relieverBB + relieverSO + relieverBAOpp +
##     relieverERA + relieverWHIP + relieverSV, data = nonoutlierdf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18748 -0.94689  0.03021  0.76840  2.38187
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.127e+00  1.824e+00   0.618 0.537909
## relieverW      1.664e-01  4.481e-02   3.714 0.000304 ***
## relieverL     -1.922e-01  4.340e-02  -4.428 2.03e-05 ***
## relieverIP     2.476e-05  2.350e-02   0.001 0.999161
## relieverH      2.545e-02  2.314e-02   1.100 0.273603
## relieverHR     3.406e-02  4.421e-02   0.770 0.442563
## relieverBB    -2.631e-03  2.728e-02  -0.096 0.923332
## relieverSO    -7.372e-03  5.719e-03  -1.289 0.199694
## relieverBAOpp -5.793e+00  1.344e+01  -0.431 0.667151
## relieverERA   -4.021e-01  2.310e-01  -1.741 0.084151 .
## relieverWHIP  -1.107e+00  2.535e+00  -0.437 0.663073
## relieverSV     6.885e-02  8.892e-03   7.742 2.67e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.088 on 127 degrees of freedom
## Multiple R-squared:  0.4816, Adjusted R-squared:  0.4367
## F-statistic: 10.72 on 11 and 127 DF,  p-value: 9.201e-14
```

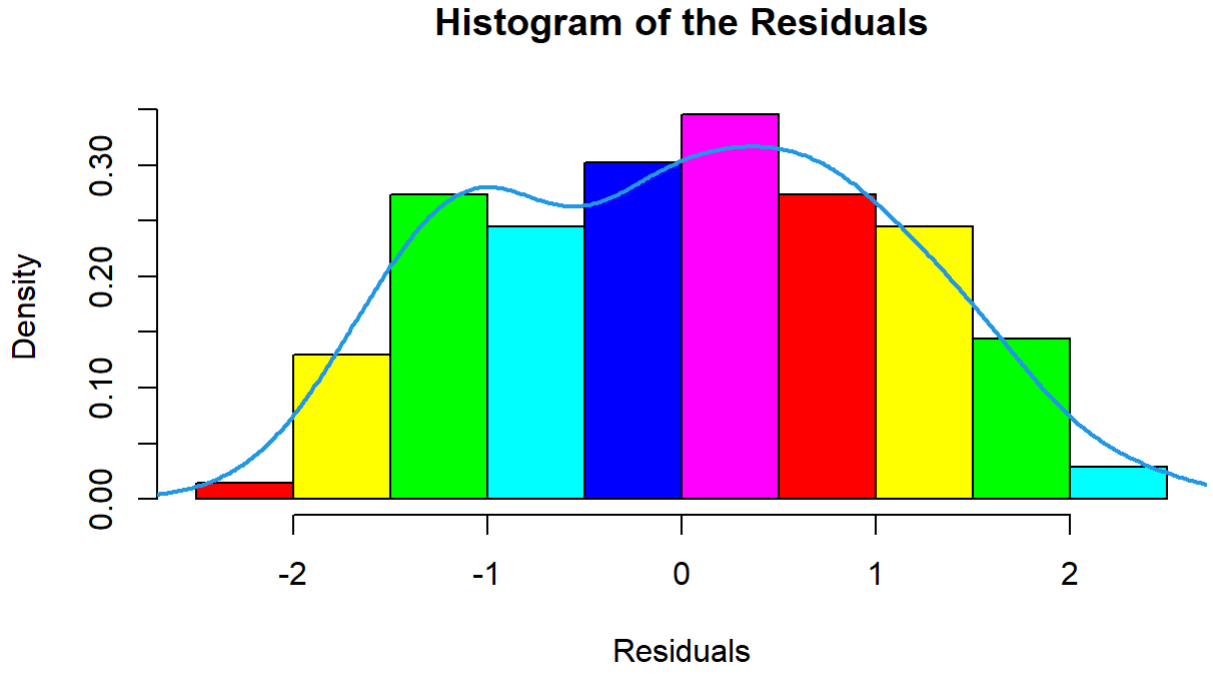
Lets look at the residuals vs fitted plot without outliers.

```
plot(relieversNoOutlier$fitted, relieversNoOutlier$residuals,main="Residuals vs Fitted Plot for
 Cy Young Votes", xlab="Fitted Values",ylab="Residuals", pch=23,bg="red",cex=1.5,lwd=1.5)
abline(h=0,col="red")
```

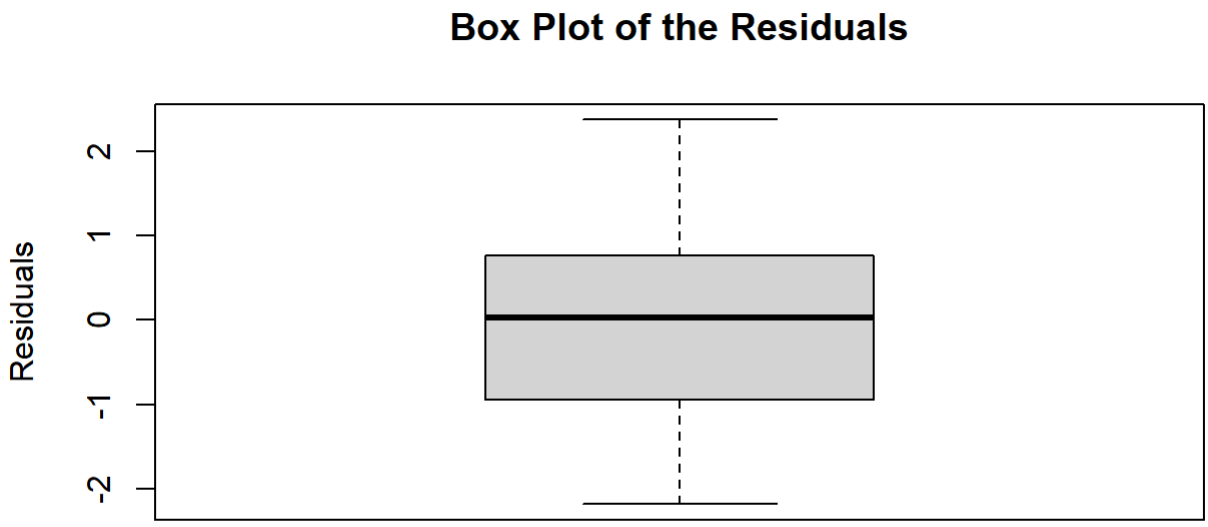


Lets look at a QQ-plot, boxplot and histogram of the residuals with normal curve.

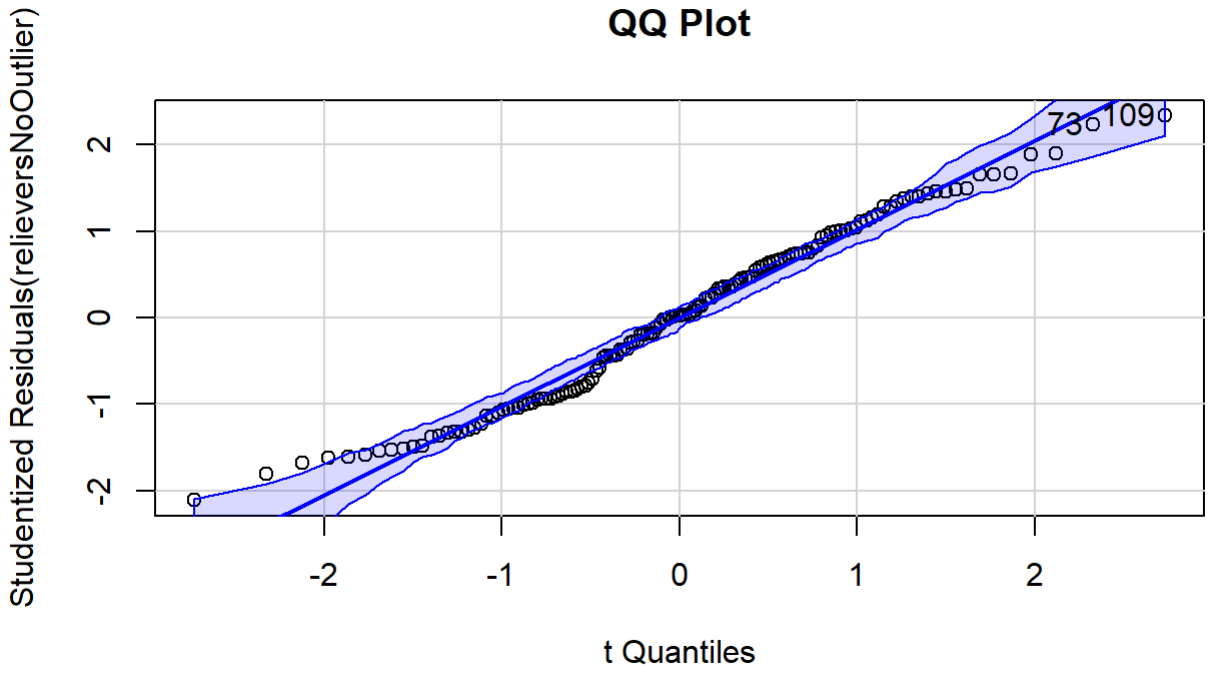
```
res=relieversNoOutlier$residuals
hist(res, prob = TRUE, main="Histogram of the Residuals", xlab="Residuals",ylab="Density", col=r
ainbow(6))
lines(density(res), col = 4, lwd = 2)
```




```
boxplot(res, main="Box Plot of the Residuals", ylab="Residuals")
```




```
library(car)
qqPlot(relieversNoOutlier, id.n=5, main='QQ Plot')
```




```
##  73 109
##  69 102
```

Now, lets make a final model by removing the insignificant variables from the model. We will do this by using the model selection criteria provided from the ‘leaps’ library.

```
library(leaps)
relieverDiagnostic <- regsubsets(logRelieverVotes ~ relieverW + relieverL + relieverIP + relieve
rH + relieverHR +
                      relieverBB + relieverSO + relieverBAOpp + relieverERA + relieverWHIP + reli
everSV, data = nonoutlierdf)
reg_summary<-summary(startersDiagnostic)
names(reg_summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

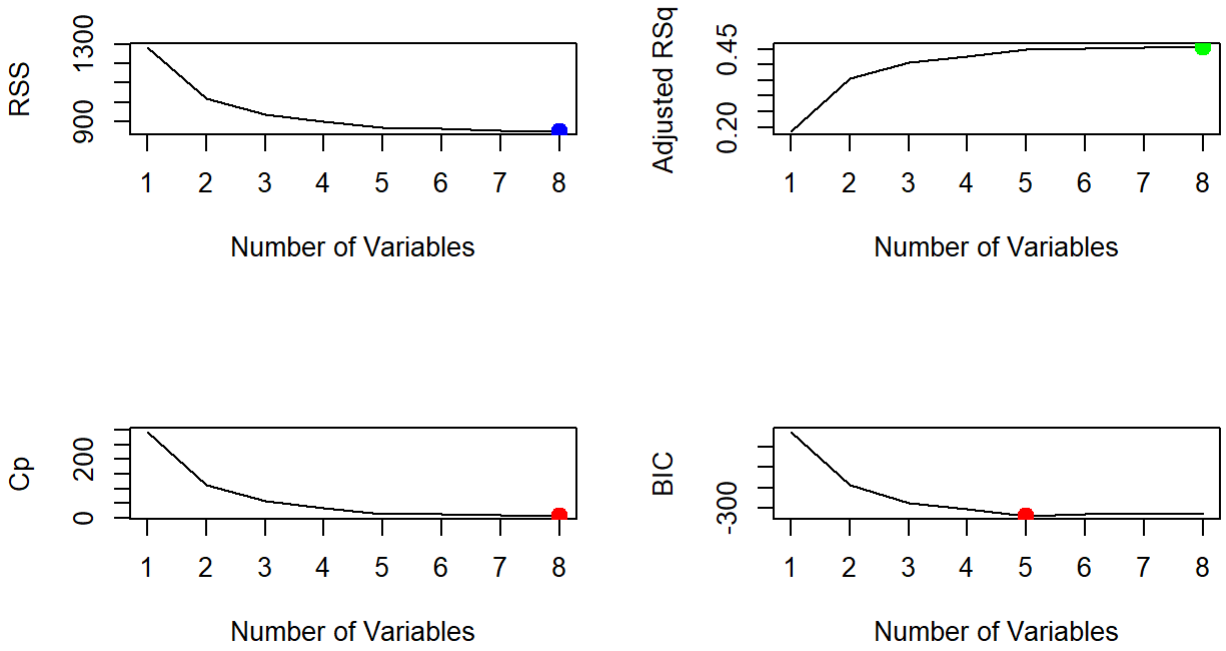
```
reg_summary$which
```

```
##   (Intercept) startersW startersL startersIP startersH startersHR startersBB
## 1        TRUE     FALSE     FALSE      FALSE     FALSE      FALSE      FALSE
## 2        TRUE      TRUE     FALSE      FALSE     FALSE      FALSE      FALSE
## 3        TRUE      TRUE      TRUE      FALSE     FALSE      FALSE      FALSE
## 4        TRUE      TRUE      TRUE      FALSE     FALSE      FALSE      FALSE
## 5        TRUE      TRUE      TRUE      FALSE     FALSE       TRUE      FALSE
## 6        TRUE      TRUE      TRUE       TRUE     FALSE       TRUE      FALSE
## 7        TRUE      TRUE      TRUE      FALSE      TRUE       TRUE      FALSE
## 8        TRUE      TRUE      TRUE       TRUE      TRUE       TRUE      FALSE
##   startersSO startersBAOpp startersERA startersWHIP
## 1      FALSE         FALSE       FALSE         TRUE
## 2      FALSE         FALSE       FALSE         TRUE
## 3      FALSE         FALSE       FALSE         TRUE
## 4       TRUE         FALSE       FALSE         TRUE
## 5       TRUE         FALSE       FALSE         TRUE
## 6       TRUE         FALSE       FALSE         TRUE
## 7       TRUE          TRUE       FALSE         TRUE
## 8       TRUE          TRUE       FALSE         TRUE
```

```
par(mfrow = c(2,2))
plot(reg_summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
rss_min<-which.min(reg_summary$rss)
points(rss_min, reg_summary$rss[rss_min],col="blue",cex = 2, pch = 20)
plot(reg_summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
adjr2_max<-which.max(reg_summary$adjr2)
points(adjr2_max, reg_summary$adjr2[adjr2_max],col="green",cex = 2, pch = 20)

plot(reg_summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
cp_min = which.min(reg_summary$cp) # 7
points(cp_min, reg_summary$cp[cp_min], col = "red", cex = 2, pch = 20)

plot(reg_summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
bic_min = which.min(reg_summary$bic) # 6
points(bic_min, reg_summary$bic[bic_min], col = "red", cex = 2, pch = 20)
```



Model selection using forward selection.

```
# Both Forward selection
relieverForward <- regsubsets(logRelieverVotes ~ relieverW + relieverL + relieverIP + relieverH
 + relieverHR +
                     relieverBB + relieverSO + relieverBAOpp + relieverERA + relieverWHIP + reli
everSV, data = nonoutlierdf, nvmax=11, method='forward')
summary(relieverForward)
```

```
## Subset selection object
## Call: regsubsets.formula(logRelieverVotes ~ relieverW + relieverL +
##     relieverIP + relieverH + relieverHR + relieverBB + relieverSO +
##     relieverBAOpp + relieverERA + relieverWHIP + relieverSV,
##     data = nonoutlierdf, nvmax = 11, method = "forward")
## 11 Variables  (and intercept)
##              Forced in Forced out
## relieverW         FALSE      FALSE
## relieverL         FALSE      FALSE
## relieverIP        FALSE      FALSE
## relieverH         FALSE      FALSE
## relieverHR        FALSE      FALSE
## relieverBB        FALSE      FALSE
## relieverSO        FALSE      FALSE
## relieverBAOpp     FALSE      FALSE
## relieverERA       FALSE      FALSE
## relieverWHIP      FALSE      FALSE
## relieverSV        FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: forward
##          relieverW relieverL relieverIP relieverH relieverHR relieverBB
## 1  ( 1 )  " "       " "       " "        " "       " "        " "
## 2  ( 1 )  "*"       " "       " "        " "       " "        " "
## 3  ( 1 )  "*"       "*"       " "        " "       " "        " "
## 4  ( 1 )  "*"       "*"       "*"        " "       " "        " "
## 5  ( 1 )  "*"       "*"       "*"        " "       " "        "*"
## 6  ( 1 )  "*"       "*"       "*"        " "       " "        "*"
## 7  ( 1 )  "*"       "*"       "*"        " "       " "        "*"
## 8  ( 1 )  "*"       "*"       "*"        " "       "*"        "*"
## 9  ( 1 )  "*"       "*"       "*"        " "       "*"        "*"
## 10  ( 1 ) "*"       "*"       "*"        "*"       "*"        "*"
## 11  ( 1 ) "*"       "*"       "*"        "*"       "*"        "*"
##          relieverSO relieverBAOpp relieverERA relieverWHIP relieverSV
## 1  ( 1 )  " "        " "           " "         " "          "*"
## 2  ( 1 )  " "        " "           " "         " "          "*"
## 3  ( 1 )  " "        " "           " "         " "          "*"
## 4  ( 1 )  " "        " "           " "         " "          "*"
## 5  ( 1 )  " "        " "           " "         " "          "*"
## 6  ( 1 )  " "        " "           "*"         " "          "*"
## 7  ( 1 )  "*"        " "           "*"         " "          "*"
## 8  ( 1 )  "*"        " "           "*"         " "          "*"
## 9  ( 1 )  "*"        " "           "*"         "*"          "*"
## 10  ( 1 ) "*"        " "           "*"         "*"          "*"
## 11  ( 1 ) "*"        "*"           "*"         "*"          "*"
```

Based off the plots, the recommended number of variables we should keep in the model is 8. We will remove H, BAOpp, and WHIP from the final model.

Lets re-declare objects for the variables used in the final model.

```
logRelieverVotes <- nonoutlierdf$logRelieverVotes
relieverW <- nonoutlierdf$relieverW
relieverL <- nonoutlierdf$relieverL
relieverIP <- nonoutlierdf$relieverIP
relieverBB <- nonoutlierdf$relieverBB
relieverSO <- nonoutlierdf$relieverSO
relieverHR <- nonoutlierdf$relieverHR
relieverERA <- nonoutlierdf$relieverERA
relieverSV <- nonoutlierdf$relieverSV
```

Final predictive model:

```
relieverFinal <- lm(logRelieverVotes ~ relieverW + relieverL + relieverIP + relieverHR +
                      relieverBB + relieverSO + relieverERA + relieverSV)
summary(relieverFinal)
```

```
##
## Call:
## lm(formula = logRelieverVotes ~ relieverW + relieverL + relieverIP +
##     relieverHR + relieverBB + relieverSO + relieverERA + relieverSV)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18908 -0.92891 -0.00557  0.77450  2.45659
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.018935   0.644363  -1.581 0.116238
## relieverW    0.171600   0.044049   3.896 0.000156 ***
## relieverL   -0.189907   0.043077  -4.409 2.16e-05 ***
## relieverIP   0.023436   0.006102   3.841 0.000191 ***
## relieverHR   0.037222   0.043774   0.850 0.396715
## relieverBB  -0.016696   0.010699  -1.560 0.121087
## relieverSO  -0.007487   0.004272  -1.753 0.082004 .
## relieverERA -0.484793   0.188812  -2.568 0.011372 *
## relieverSV   0.067454   0.008723   7.733 2.54e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.082 on 130 degrees of freedom
## Multiple R-squared:  0.4751, Adjusted R-squared:  0.4428
## F-statistic: 14.71 on 8 and 130 DF,  p-value: 3.604e-15
```

Part 3: Predictions for 2022 Awards Recipients

Lets test these models on data from the 2022 MLB season. The recipients of the awards are announced on November 7th, 2022. These predictions were made on November 6th.

The data sets I will be importing are of qualified starters and relievers from statcast.com.

```
stats_2022 <- read_csv("stats.csv")
```

```
## Rows: 45 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): last_name, first_name, LG
## dbl (8): IP, H, HR, SO, BAOpp, W, L, WHIP
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

This data only contains starting pitchers, so lets run our starters predictive model and make predictions.

```
startersW <- c(stats_2022$W)
startersL <- c(stats_2022$L)
startersIP <- c(stats_2022$IP)
startersH <- c(stats_2022$H)
startersHR <- c(stats_2022$HR)
startersSO <- c(stats_2022$SO)
startersBAOpp <- c(stats_2022$BAOpp)
startersWHIP <- c(stats_2022$WHIP)
predict_starters <- data.frame(startersW, startersL, startersIP, startersH, startersHR, starters
SO, startersBAOpp, startersWHIP)
starter_predictions <- predict(startersFinal, newdata = predict_starters)
starter_predictions <- as.data.frame(starter_predictions)
starter_predictions <- mutate(starter_predictions, predicted_votes = exp(starter_predictions))
starter_predictions <- mutate(starter_predictions, last_name = stats_2022$last_name)
starter_predictions <- mutate(starter_predictions, first_name = stats_2022$first_name)
starter_predictions <- mutate(starter_predictions, LG = stats_2022$LG)
colnames(starter_predictions)[1] <- 'log_prediction'
```

Lets import data on qualified relievers from 2022.

```
relieverstats <- read_csv("relieverstats - Copy.csv")
```

```
## Rows: 49 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): last_name, first_name, LG
## dbl (8): IP, HR, SO, BB, SV, W, L, ERA
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Run our relievers predictive model and make predictions.

```
relieverW <- c(relieverstats$W)
relieverL <- c(relieverstats$L)
relieverIP <- c(relieverstats$IP)
relieverHR <- c(relieverstats$HR)
relieverBB <- c(relieverstats$BB)
relieverSO <- c(relieverstats$SO)
relieverERA <- c(relieverstats$ERA)
relieverSV <- c(relieverstats$SV)
predict_relievers <- data.frame(relieverW, relieverL, relieverIP, relieverHR, relieverBB, reliev
erSO, relieverERA, relieverSV)
reliever_predictions <- predict(relieverFinal, newdata = predict_relievers)
reliever_predictions <- as.data.frame(reliever_predictions)
reliever_predictions <- mutate(reliever_predictions, predicted_votes = exp(reliever_prediction
s))
reliever_predictions <- mutate(reliever_predictions, last_name = relieverstats$last_name)
reliever_predictions <- mutate(reliever_predictions, first_name = relieverstats$first_name)
reliever_predictions <- mutate(reliever_predictions, LG = relieverstats$LG)
colnames(reliever_predictions)[1] <- 'log_prediction'
```

Merge starter and reliever predictions into one final 2022 Cy Young Predictions data frame and show the top 20 results.

```
predictions<-rbind(starter_predictions,reliever_predictions)
predictions<-predictions[order(-predictions$predicted_votes),]
head(predictions, n = 20)
```

```
##    log_prediction predicted_votes  last_name first_name LG
## 2        4.844915      127.092531  Verlander     Justin AL
## 34       3.864661       47.687085     Wright       Kyle NL
## 29       3.224317       25.136402      Urias      Julio NL
## 7        3.223400       25.113361    Darvish         Yu NL
## 41       3.199368       24.517036     Manoah       Alek AL
## 10       3.186899       24.213217   Anderson      Tyler NL
## 39       3.082362       21.809857     Valdez    Framber AL
## 36       3.040548       20.916697     Ohtani     Shohei AL
## 22       2.974215       19.574245      Rodon     Carlos NL
## 42       2.972248       19.535777     Gallen        Zac NL
## 24       2.902272       18.215491      Fried        Max NL
## 31       2.852231       17.326400  Alcantara      Sandy NL
## 38       2.434779       11.413299 McClanahan      Shane AL
## 45       2.416223       11.203463     Bieber      Shane AL
## 43       2.355718       10.545699     Burnes     Corbin NL
## 35       2.322736       10.203553       Webb      Logan NL
## 32       2.176268        8.813357      Cease      Dylan AL
## 11       2.124867        8.371782       Cole     Gerrit AL
## 19       2.107343        8.226358    Bassitt      Chris NL
## 16       2.073183        7.950086    Taillon    Jameson AL
```