DATA 621 Business Analytics and Data Mining

Group 2 - Gabriel Campos, Melissa Bowman, Alexander Khaykin, & Jennifer Abinette

Last edited October 13, 2023

Homework #1 Assignment Requirements

Overview

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record 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.

Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. You can only use the variables given to you (or variables that you derive from the variables provided).

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

Variable Names	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

Deliverable:

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
- Assigned predictions (the number of wins for the team) for the evaluation data set.
- Include your R statistical programming code in an Appendix.

Write Up:

1. **DATA EXPLORATION (25 Points)** Describe the size and the variables in the moneyball training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas.

a. Mean / Standard Deviation / Median
b. Bar Chart or Box Plot of the data
c. Is the data correlated to the target variable (or to other variables?)
d. Are any of the variables missing and need to be imputed "fixed"?

2. **DATA PREPARATION (25 Points)** Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations.

a. Fix missing values (maybe with a Mean or Median value)
b. Create flags to suggest if a variable was missing
c. Transform data by putting it into buckets
d. Mathematical transforms such as log or square root (or use Box-Cox)
e. Combine variables (such as ratios or adding or multiplying) to create new variables

3. **BUILD MODELS (25 Points)** Using the training data set, build at least three different multiple linear regression models, using different variables (or the same variables with different transformations). Since we have not yet covered automated variable selection methods, you should select the variables manually (unless you previously learned Forward or Stepwise selection, etc.). Since you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done. Discuss the coefficients in the models, do they make sense? For example, if a team hits a lot of Home Runs, it would be reasonably expected that such a team would win more games. However, if the coefficient is negative (suggesting that the team would lose more games), then that needs to be discussed. Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

4. **SELECT MODELS (25 Points)** Decide on the criteria for selecting the best multiple linear regression model. Will you select a model with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your model. For the multiple linear regression model, will you use a metric such as Adjusted R2 , RMSE, etc.? Be sure to explain how you can make inferences from the model, discuss multi-collinearity issues (if any), and discuss other relevant model output. Using the training data set, evaluate the multiple linear regression model based on (a) mean squared error, (b) R2, (c) F-statistic, and (d) residual plots. Make predictions using the evaluation data set.

Evaluation

1. DATA EXPLORATION

```
## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'knitr' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'corrplot' was built under R version 4.2.3

## Warning: package 'ResourceSelection' was built under R version 4.2.3
```

Load the data

```
git_url<-
  "https://raw.githubusercontent.com/melbow2424/Data621_HW1/main/"

df_train <-
  read.csv(paste0(git_url,"moneyball-training-data.csv"))

df_evaluation <-
  read.csv(paste0(git_url,"moneyball-evaluation-data.csv"))
```

Summary of Variables

```
# Remove TEAM_ prefix from column names
df_train <-
  rename(df_train,
         "BATTING_HITS"="TEAM_BATTING_H", "BATTING_2B"="TEAM_BATTING_2B",
         "BATTING_3B"="TEAM_BATTING_3B", "BATTING_HR"="TEAM_BATTING_HR",
         "BATTING_BB"="TEAM_BATTING_BB", "BASERUN_SB"="TEAM_BASERUN_SB",
         "BASERUN_CS"="TEAM_BASERUN_CS", "BATTING_HBP"="TEAM_BATTING_HBP",
         "PITCHING_HITS"="TEAM_PITCHING_H", "PITCHING_HR"="TEAM_PITCHING_HR",
         "PITCHING_BB"="TEAM_PITCHING_BB", "FIELD_ERRORS"="TEAM_FIELDING_E",
         "FIELD_DBLPLY"="TEAM_FIELDING_DP","BATTING_SO"="TEAM_BATTING_SO",
         "PITCHING_SO"="TEAM_PITCHING_SO")
df_evaluation <-
  rename(df_evaluation,
         "BATTING_HITS"="TEAM_BATTING_H", "BATTING_2B"="TEAM_BATTING_2B",
         "BATTING_3B"="TEAM_BATTING_3B", "BATTING_HR"="TEAM_BATTING_HR",
         "BATTING_BB"="TEAM_BATTING_BB", "BASERUN_SB"="TEAM_BASERUN_SB",
         "BASERUN_CS"="TEAM_BASERUN_CS", "BATTING_HBP"="TEAM_BATTING_HBP",
         "PITCHING_HITS"="TEAM_PITCHING_H", "PITCHING_HR"="TEAM_PITCHING_HR",
         "PITCHING_BB"="TEAM_PITCHING_BB",  "FIELD_ERRORS"="TEAM_FIELDING_E",
```

```
        "FIELD_DBLPLY"="TEAM_FIELDING_DP","BATTING_SO"="TEAM_BATTING_SO",
        "PITCHING_SO"="TEAM_PITCHING_SO")

# Show variable stats for training dataset
print(skim(df_train))
```

```
## -- Data Summary ------------------------
##                            Values
## Name                       df_train
## Number of rows             2276
## Number of columns          17
## _______________________
## Column type frequency:
##   numeric                  17
## _______________________
## Group variables            None
##
## -- Variable type: numeric -------------------------------------------------------
##    skim_variable n_missing complete_rate   mean     sd   p0    p25   p50   p75
##  1 INDEX                 0        1      1268.   736.     1  631.  1270. 1916.
##  2 TARGET_WINS           0        1        80.8   15.8    0   71     82    92
##  3 BATTING_HITS          0        1      1469.   145.   891 1383   1454  1537.
##  4 BATTING_2B            0        1       241.    46.8   69  208    238   273
##  5 BATTING_3B            0        1        55.2   27.9    0   34     47    72
##  6 BATTING_HR            0        1        99.6   60.5    0   42    102   147
##  7 BATTING_BB            0        1       502.   123.     0  451    512   580
##  8 BATTING_SO          102        0.955   736.   249.     0  548    750   930
##  9 BASERUN_SB          131        0.942   125.    87.8    0   66    101   156
## 10 BASERUN_CS          772        0.661    52.8   23.0    0   38     49    62
## 11 BATTING_HBP        2085        0.0839   59.4   13.0   29   50.5   58    67
## 12 PITCHING_HITS         0        1      1779.  1407.  1137 1419   1518  1682.
## 13 PITCHING_HR           0        1       106.    61.3    0   50    107   150
## 14 PITCHING_BB           0        1       553.   166.     0  476    536.  611
## 15 PITCHING_SO         102        0.955   818.   553.     0  615    814.  968
## 16 FIELD_ERRORS          0        1       246.   228.    65  127    159   249.
## 17 FIELD_DBLPLY        286        0.874   146.    26.2   52  131    149   164
##     p100 hist
##  1  2535
##  2   146
##  3  2554
##  4   458
##  5   223
##  6   264
##  7   878
##  8  1399
##  9   697
## 10   201
## 11    95
## 12 30132
## 13   343
## 14  3645
## 15 19278
## 16  1898
```

```
## 17   228
```

Observations

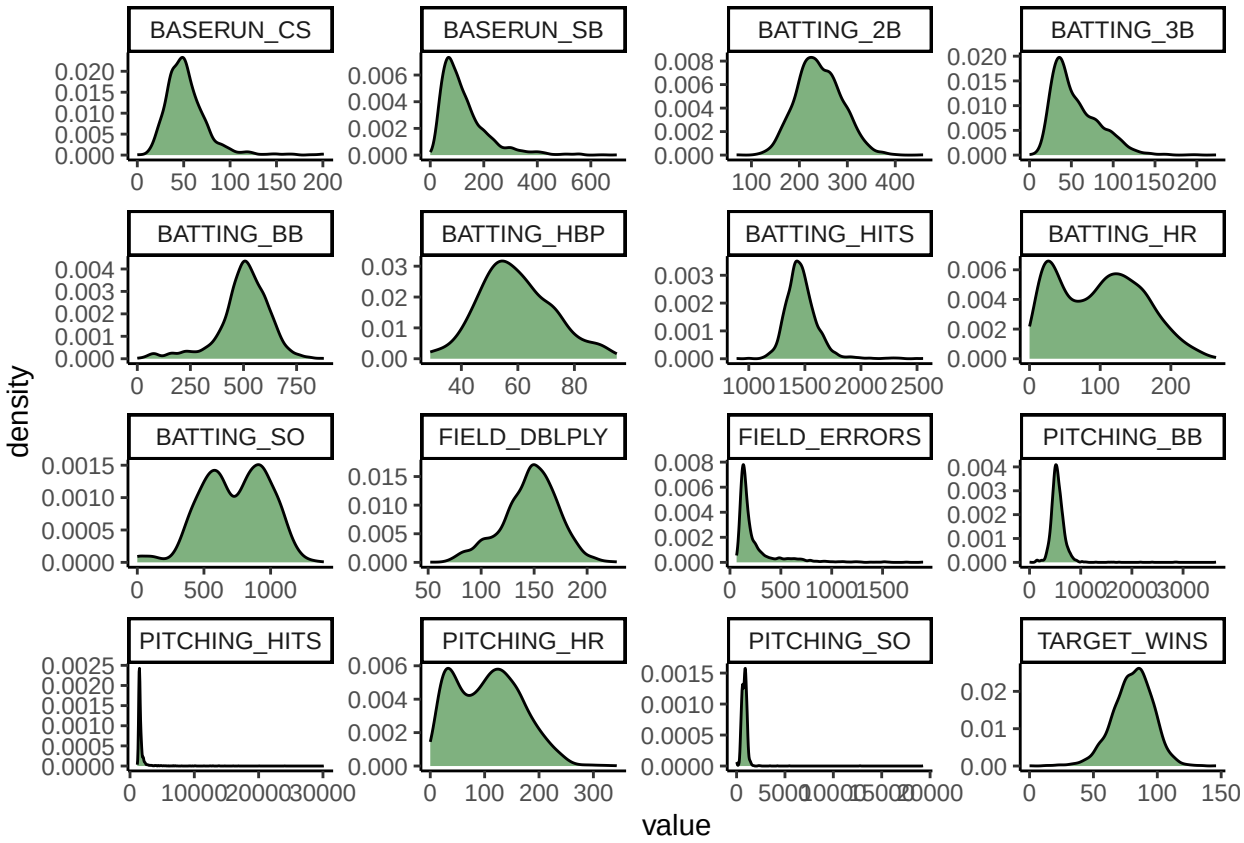
As shown above, the training data includes 17 variables (although Index is only for identification purposes) and 2,276 cases. We intend to create a Regression model that will predict the count of wins a team had over a 162 game season (TARGET_WINS) using the team's statistics on offensive and defensive plays. For these cases, teams had an average of 81 wins with a std deviation of 16 games with counts ranging from 0 to 146. The remaining variables include plays during offense, such as base hits and stealing bases (variables starting with BATTING or BASERUN), as well as defense plays, such as pitching strikeouts and double plays (PITCHING or FIELD).

Note that 6 variables are missing values including 2,085 values (92% missing) for batter hit by pitch (BATTING_HBP), 772 values (34%) for base runner caught stealing (BASERUN_CS), 286 values (13%) for fielding double plays (FIELD_DBLPLY), and the remaining three (stealing bases and strikeouts as batter and pitcher) missing less than 6% of values (BASERUN_SB, BATTING_SO, PITCHING_SO). We will explore how to handle these missing values later.

Distribution of Variables

```
df_train %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "darkgreen", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```

```
## Warning: Removed 3478 rows containing non-finite values (`stat_density()`).
```



Observations

Histograms a good way to visualize the distributions of the original variables as shown above. We can see the response variable (TARGET_WINS) appears normally distributed. Several variables, such as BATTING_SO appear to be bimodal, which may resolve after the missing data is dealt with. Other variables like PITCHING_HITS appear to be skewed far left and may present a challenge unless imputation of missing values corrects this.

Correlation with Target Wins

```
cor(df_train, y=df_train$TARGET_WINS)
```

```
##                     [,1]
## INDEX         -0.02105643
## TARGET_WINS    1.00000000
## BATTING_HITS   0.38876752
## BATTING_2B     0.28910365
## BATTING_3B     0.14260841
## BATTING_HR     0.17615320
## BATTING_BB     0.23255986
## BATTING_SO             NA
## BASERUN_SB             NA
## BASERUN_CS             NA
```

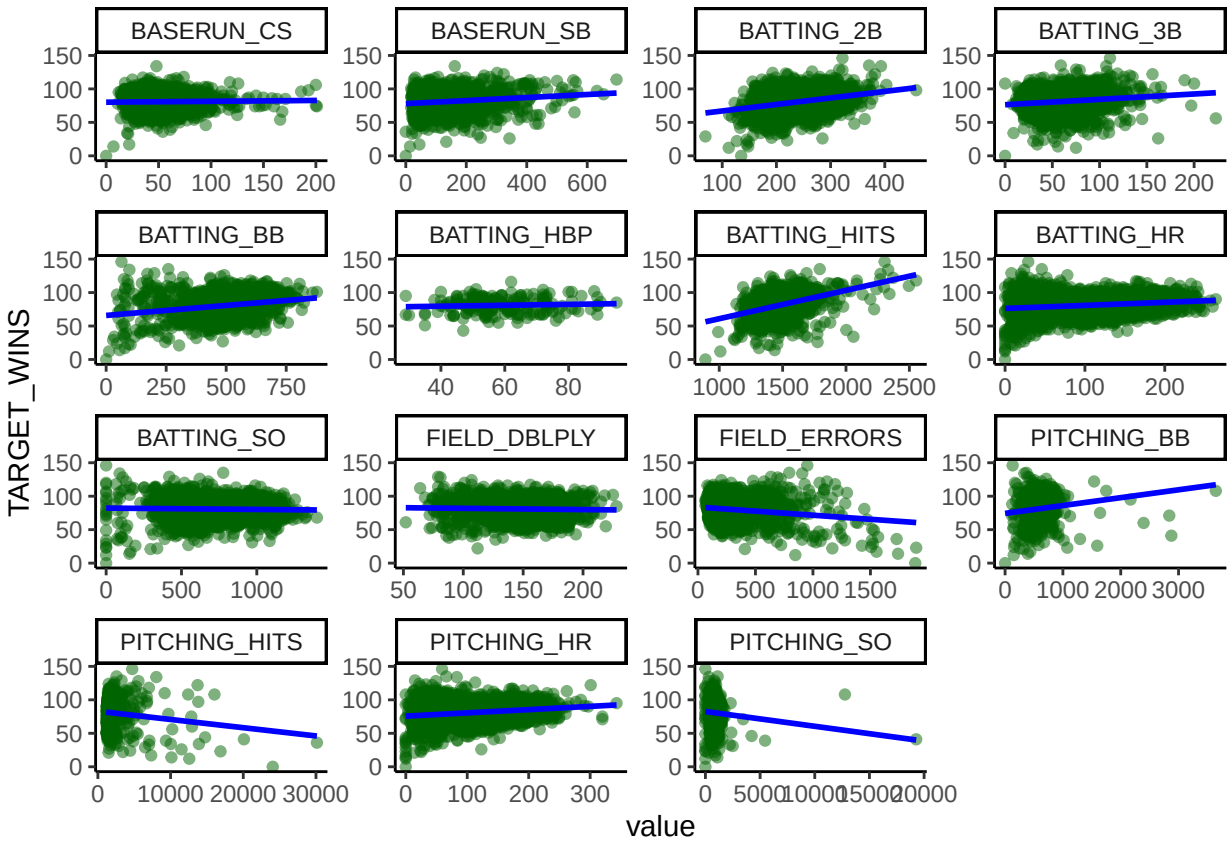
```
## BATTING_HBP             NA
## PITCHING_HITS -0.10993705
## PITCHING_HR    0.18901373
## PITCHING_BB    0.12417454
## PITCHING_SO             NA
## FIELD_ERRORS  -0.17648476
## FIELD_DBLPLY            NA
```

```
df_train %>%
  #pivot longer to plot all variables
  gather(variable, value, BATTING_HITS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value, y=TARGET_WINS)) + #plotting every variable
  geom_point(color = "darkgreen", alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 3478 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 3478 rows containing missing values (`geom_point()`).
```



Observations

The plots above demonstrate the relationships between number of wins (TARGET_WINS) and each remaining variable (besides the variables missing values) that we will be exploring as predictors. The predictor variables with the strongest correlation to number of wins are total base hits (BATTING_HITS), doubles by batters (BATTING_2B) and walks by batters (BATTING_BB) which have correlations of .39, .29 and .23, respectively.

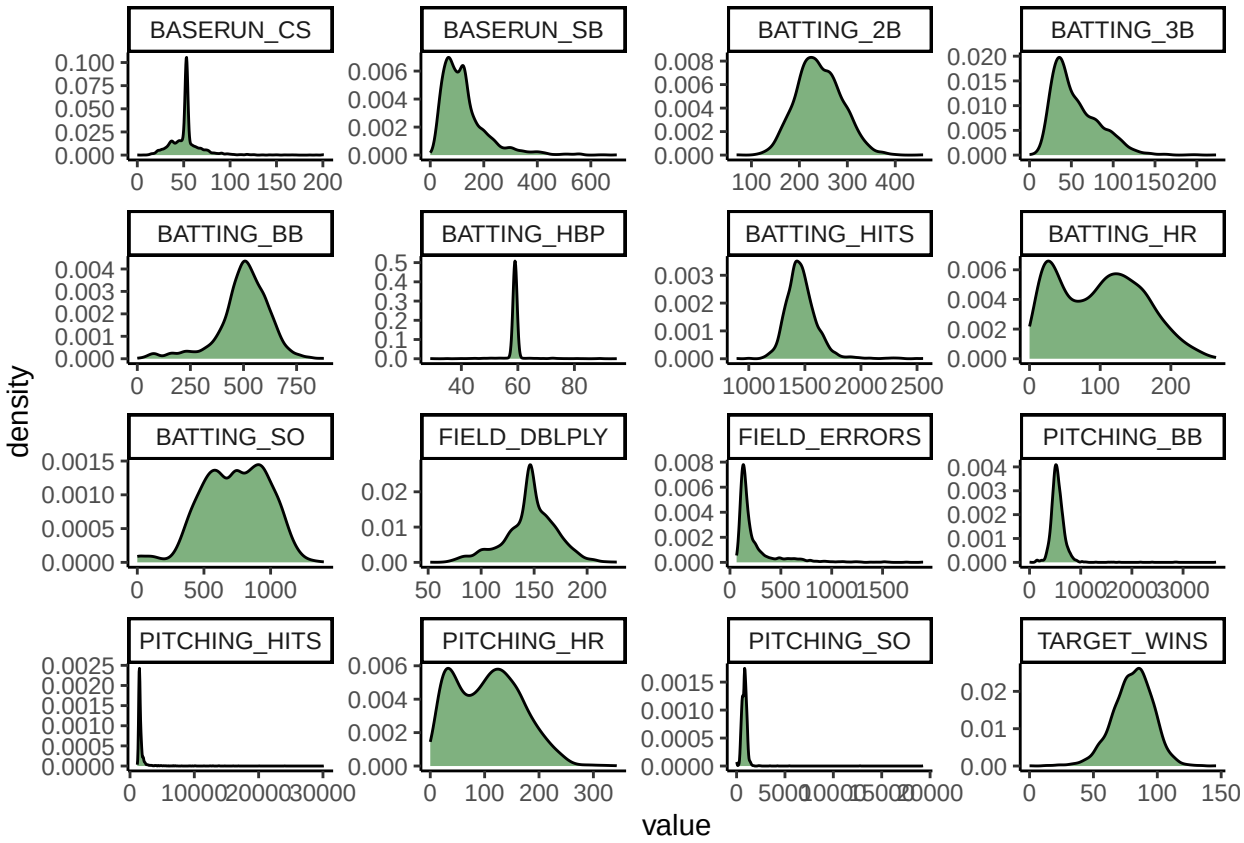
2. DATA PREPARATION

Dealing with Missing Values - replace NA with mean, median, zero, or remove cases

Mean Imputation

```
# Get the Means of columns in Data
train_means<-sapply(df_train, function(x) round(mean(x, na.rm = TRUE)))

# Replace NA values in 'column_name' with 'mean'
df_train_mn <- df_train %>%
  mutate(BATTING_SO =
           ifelse(is.na(BATTING_SO),
                  train_means[8],BATTING_SO))%>%
  mutate(BASERUN_SB =
           ifelse(is.na(BASERUN_SB),
                  train_means[9], BASERUN_SB))%>%
  mutate(BASERUN_CS =
           ifelse(is.na(BASERUN_CS),
                  train_means[10], BASERUN_CS))%>%
  mutate(BATTING_HBP =
           ifelse(is.na(BATTING_HBP),
                  train_means[11],BATTING_HBP))%>%
  mutate(PITCHING_SO =
           ifelse(is.na(PITCHING_SO),
                  train_means[15], PITCHING_SO))%>%
  mutate(FIELD_DBLPLY =
           ifelse(is.na(FIELD_DBLPLY),
                  train_means[17], FIELD_DBLPLY))
```

```
# Evaluate histograms
df_train_mn %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "darkgreen", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```




```
# Fit a multiple linear regression model using lm with variables imputed:mean
model_mn <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+BATTING_HR+
                 BATTING_BB+BATTING_SO+BASERUN_SB+BASERUN_CS+BATTING_HBP+
                 PITCHING_HITS+PITCHING_HR+PITCHING_BB+PITCHING_SO+
                 FIELD_ERRORS+FIELD_DBLPLY, data = df_train_mn)

# Summary of the regression model
summary(model_mn)$adj.r.squared
```

```
## [1] 0.3146964
```

Observations After imputation with means:

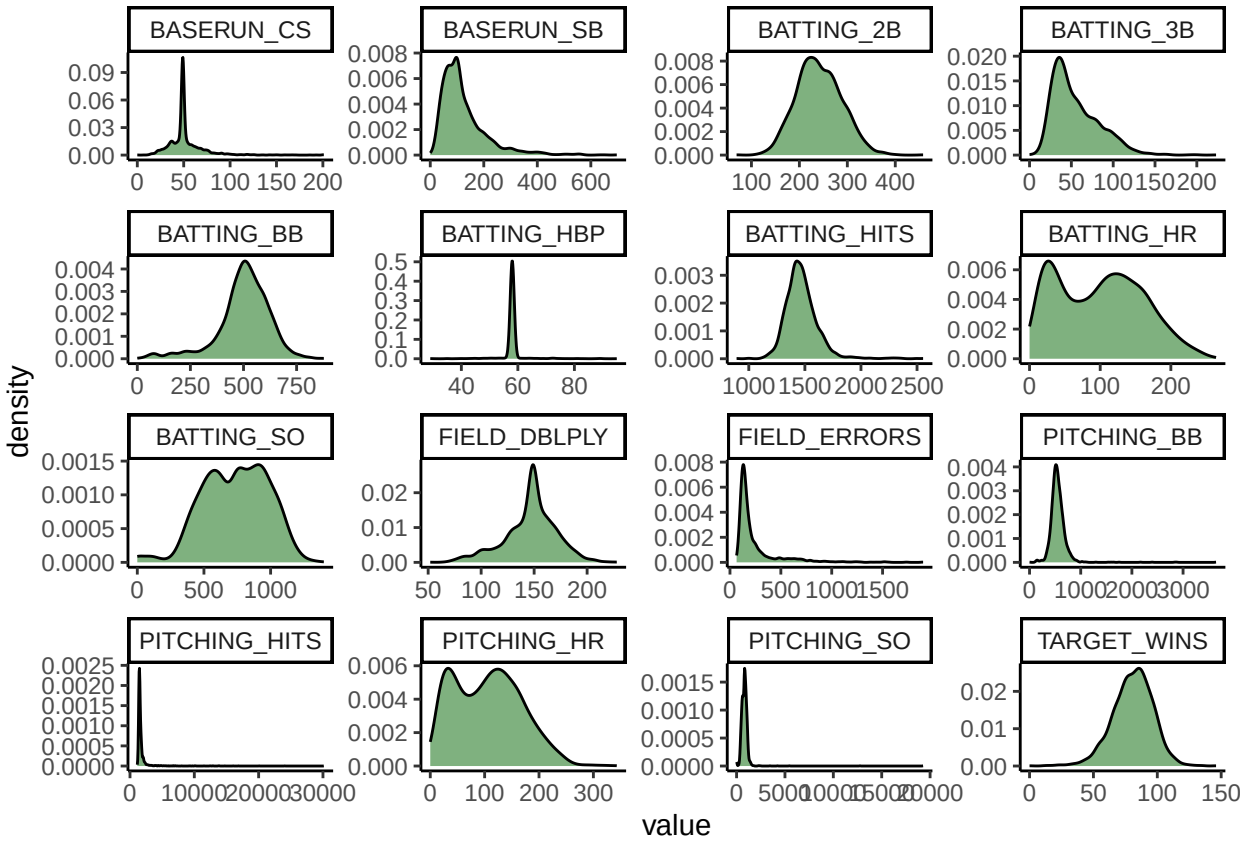
- The response variable **TARGET_WINS** still appears normally distributed
- The bimodality of the **BATTING** variables is largely unresolved
- The far left skew of the **PITCHING** variables is largely unresolved
- A Multiple Linear Regression with all variables has an adjusted R squared of .315

Median Imputation

```
# Get the Medians of columns in data
train_medians<-sapply(df_train, function(x) round(median(x, na.rm = TRUE)))
```

```r
# Replace NA values in 'column_name' with 'median'
df_train_md <- df_train %>%
  mutate(BATTING_SO =
           ifelse(is.na(BATTING_SO),
                  train_medians[8],BATTING_SO))%>%
  mutate(BASERUN_SB =
           ifelse(is.na(BASERUN_SB),
                  train_medians[9], BASERUN_SB))%>%
  mutate(BASERUN_CS =
           ifelse(is.na(BASERUN_CS),
                  train_medians[10], BASERUN_CS))%>%
  mutate(BATTING_HBP =
           ifelse(is.na(BATTING_HBP),
                  train_medians[11],BATTING_HBP))%>%
  mutate(PITCHING_SO =
           ifelse(is.na(PITCHING_SO),
                  train_medians[15], PITCHING_SO))%>%
  mutate(FIELD_DBLPLY =
           ifelse(is.na(FIELD_DBLPLY),
                  train_medians[17], FIELD_DBLPLY))
```

```r
df_train_md %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "darkgreen", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```




```
# Fit a multiple linear regression model using lm with variables imputed:median
model_md <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+BATTING_HR+
                 BATTING_BB+BATTING_SO+BASERUN_SB+BASERUN_CS+BATTING_HBP+
                 PITCHING_HITS+PITCHING_HR+PITCHING_BB+PITCHING_SO+FIELD_ERRORS+
                 FIELD_DBLPLY, data = df_train_md)

# Summary of the regression model
summary(model_md)$adj.r.squared
```

```
## [1] 0.3109674
```

Observations After imputation with the median:

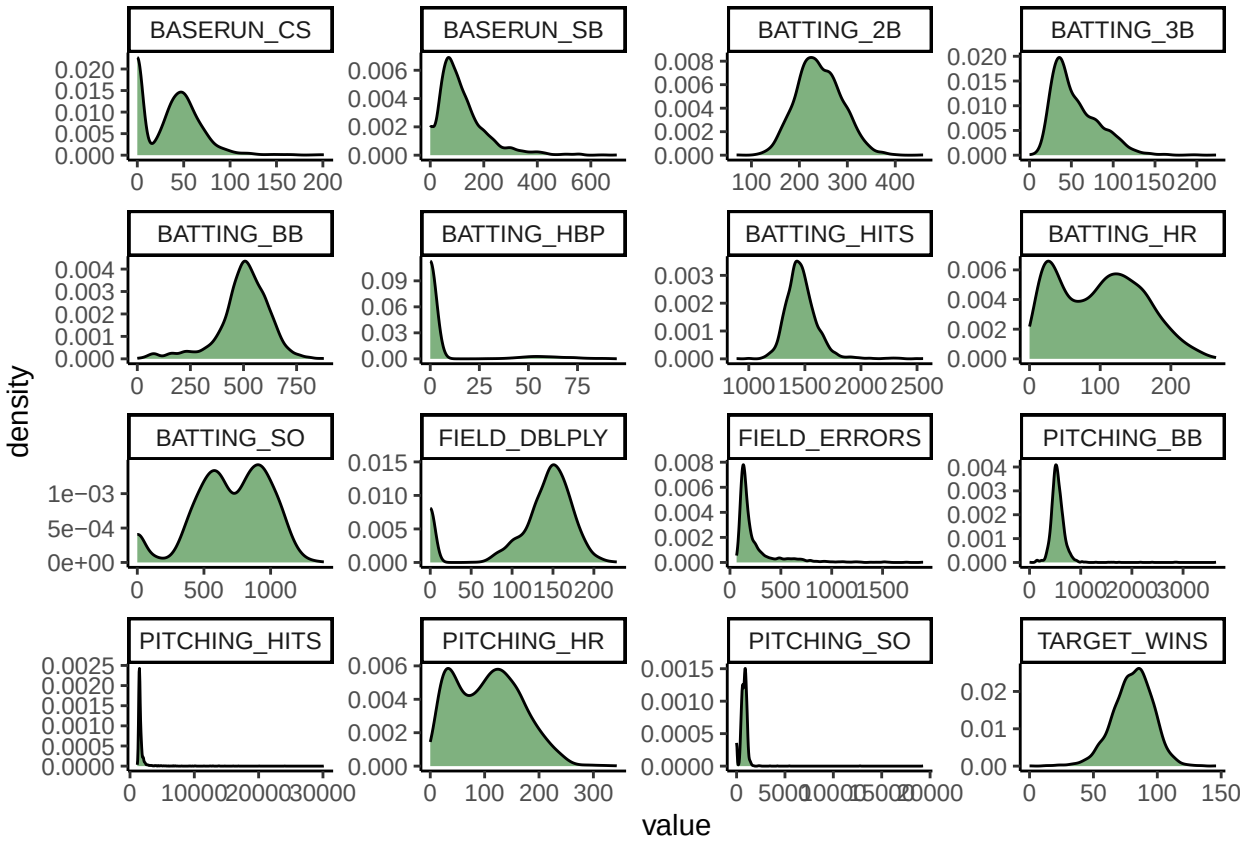
- The response variable **TARGET_WINS** still appears normally distributed
- The bimodality of the **BATTING** variables is largely unresolved
- The far left skew of the **PITCHING** variables is largely unresolved

- A Multiple Linear Regression with all variables has an adjusted R squared of .311

Zero Imputation

```
# Replace NA values with zero
df_train_0 <- df_train %>%
  replace_na( list( INDEX = 0,TARGET_WINS = 0,BATTING_HITS = 0,BATTING_2B = 0,
    BATTING_3B = 0,BATTING_HR = 0,BATTING_BB = 0,BATTING_SO = 0,BASERUN_SB = 0,
    BASERUN_CS = 0,BATTING_HBP = 0,PITCHING_HITS = 0,PITCHING_HR = 0,
    PITCHING_BB = 0,PITCHING_SO = 0,FIELD_ERRORS = 0,FIELD_DBLPLY = 0))
```

```
df_train_0 %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "darkgreen", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```




```
# Fit a multiple linear regression model using lm with variables imputed:zero
model_0 <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+BATTING_HR+
                BATTING_BB+BATTING_SO+BASERUN_SB+BASERUN_CS+BATTING_HBP+
                PITCHING_HITS+PITCHING_HR+PITCHING_BB+PITCHING_SO+FIELD_ERRORS+
                FIELD_DBLPLY,data = df_train_0)

# Summary of the regression model
summary(model_0)$adj.r.squared
```

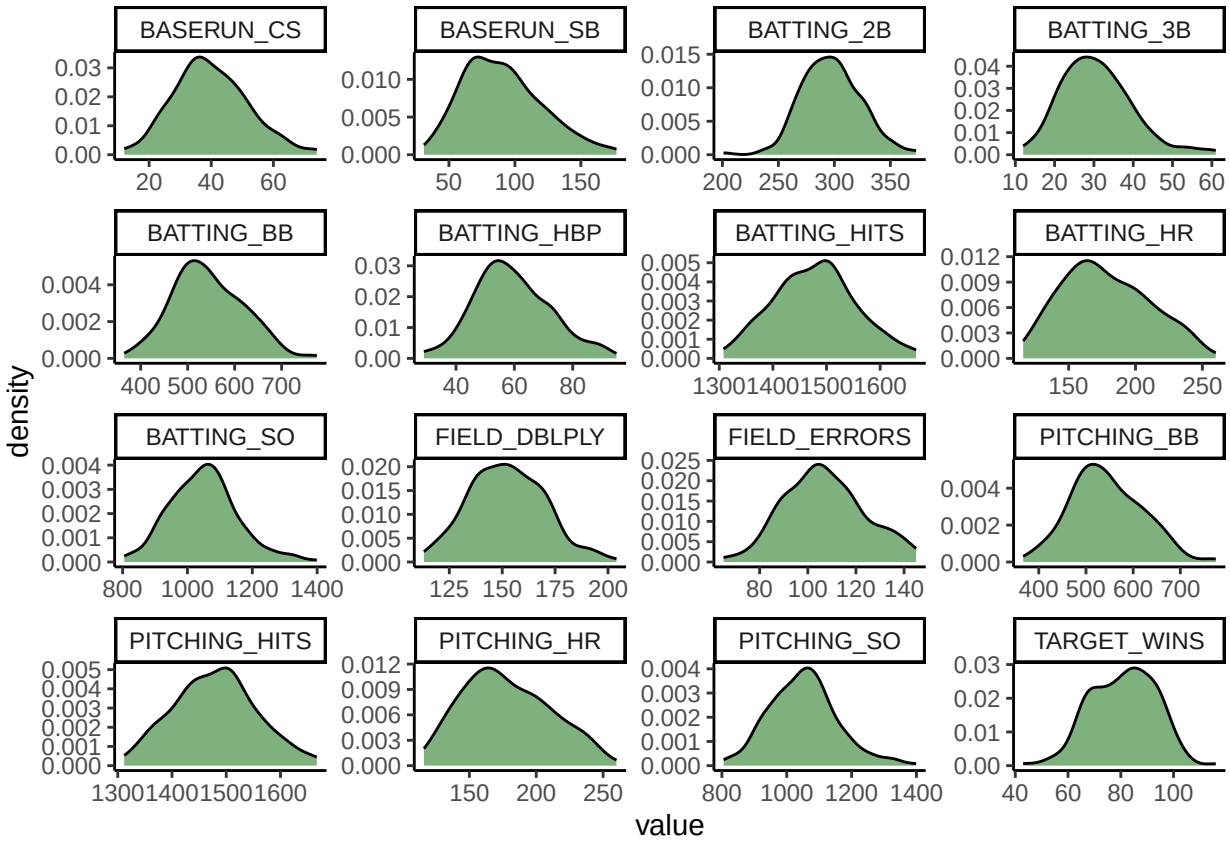
```
## [1] 0.2943554
```

Observations After imputation with the zero:

- Zero is a poor choice as it introduces strong peaks to the left of the distribution of many variables such as **FIELD_DBLPLY**
- A Multiple Linear Regression with all variables has an adjusted R squared of .294

Remove NA Values

```
# Remove all rows with NA
df_train_rm<- na.omit(df_train)

# Evaluate distributions
df_train_rm %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "darkgreen", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```




```
# Fit a multiple linear regression model using lm
model_rm <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+BATTING_HR+
                 BATTING_BB+BATTING_SO+BASERUN_SB+BASERUN_CS+BATTING_HBP+
```

```
                  PITCHING_HITS+PITCHING_HR+PITCHING_BB+PITCHING_SO+FIELD_ERRORS+
                  FIELD_DBLPLY,data = df_train_rm)

# Summary of the regression model
summary(model_rm)$adj.r.squared
```

```
## [1] 0.511555
```

```
# How many values are missing from BATTING_HBP in the evaluation dataset
# that we will need to use our regression model to predict wins for?
print(sum(is.na(df_evaluation$BATTING_HBP)))
```

```
## [1] 240
```

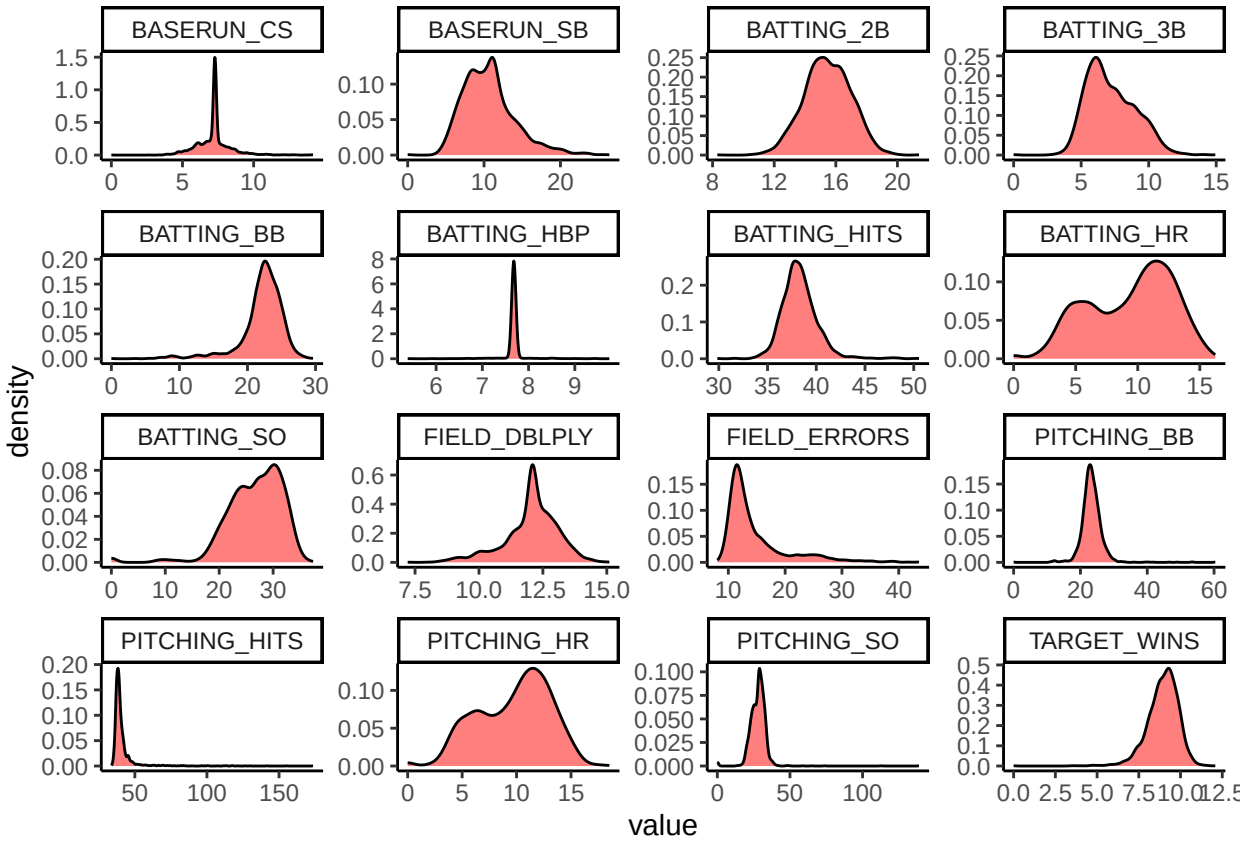
Observations After Removal of missing data:

- Although removal of missing data resolves the distributions of many variables, the sample size is reduced to under 10% of the original dataset from 2,276 observations to only 191 observations in the training dataset
- Additionally, batters hit by pitches (BATTING_HBP) is missing values for 240 of 259 cases (93%) in the evaluation dataset which we will be using our regression model to predict wins for.
- Observed a more favorable adjusted R-squared value than imputation with mean, median or zero
- A Multiple Linear Regression with all variables has an adjusted R squared of .512

Transformations

Square Root with Mean Imputation

```
df_train_sqrt <- sqrt(df_train_mn)

df_train_sqrt %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "red", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```



Observations Square root transformation is common to use when variables are counts to stabilize variance. Given mean imputation offered the best Multiple Regression effect size of .315 without removing observations, we used that dataset to perform square root transformations on. We can see from the histograms above that this transformation has improved the distribution of the variables to be more normal.

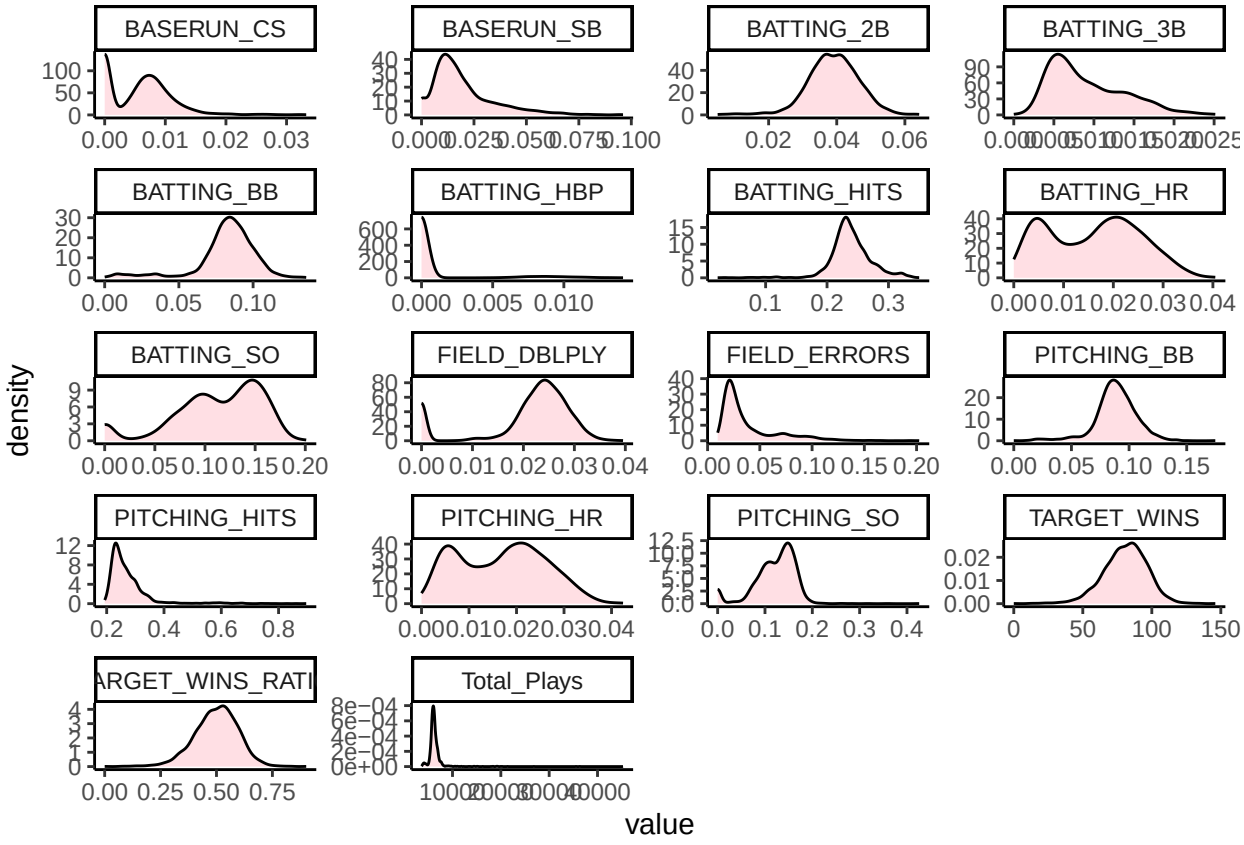
Ratio with Zero Imputation

Create dataset transforming variables to ratios. The Count of Wins will be divided by 162 to create a ratio of wins in the season. Additionally, we calculated new variable Total Plays that is the total number of plays for each team. Total Plays sums the number of pitching hits, batter hits and the other Base runner and fielding variables. Note that BATTING_HITS & PITCHING_HITS already include the counts for BATTING_2B, BATTING_3B, BATTING_HR and PITHCING_HR. The variables are then converted to a ratio using Total Plays as the denominator.

```
df_train_ratio <- df_train_0 %>%
  mutate(Total_Plays = BATTING_HITS + BATTING_BB + BATTING_SO + BATTING_HBP +
           BASERUN_SB + BASERUN_CS +
           PITCHING_BB + PITCHING_HITS + PITCHING_SO +
           FIELD_DBLPLY + FIELD_ERRORS) %>%
  ### Target_Wins is based on 162 game season
  mutate(TARGET_WINS_RATIO = TARGET_WINS/162) %>%
  mutate(BATTING_HITS = BATTING_HITS/Total_Plays) %>%
  mutate(BATTING_2B = BATTING_2B/Total_Plays) %>%
  mutate(BATTING_3B = BATTING_3B/Total_Plays) %>%
```

```
  mutate(BATTING_HR = BATTING_HR/Total_Plays) %>%
  mutate(BATTING_BB = BATTING_BB/Total_Plays) %>%
  mutate(BATTING_SO = BATTING_SO/Total_Plays) %>%
  mutate(BATTING_HBP = BATTING_HBP/Total_Plays) %>%
  mutate(BASERUN_SB = BASERUN_SB/Total_Plays) %>%
  mutate(BASERUN_CS = BASERUN_CS/Total_Plays) %>%
  mutate(PITCHING_BB = PITCHING_BB/Total_Plays) %>%
  mutate(PITCHING_HITS = PITCHING_HITS/Total_Plays) %>%
  mutate(PITCHING_HR = PITCHING_HR/Total_Plays) %>%
  mutate(PITCHING_SO = PITCHING_SO/Total_Plays) %>%
  mutate(FIELD_DBLPLY = FIELD_DBLPLY/Total_Plays) %>%
  mutate(FIELD_ERRORS = FIELD_ERRORS/Total_Plays)
```

```
df_train_ratio %>%
  #pivot longer to plot all variables
  gather(variable, value, TARGET_WINS: TARGET_WINS_RATIO)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "pink", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```



Observations Ratio transformation can also be used to stabilize variance and offers a model based on the total number of plays for each observation allowing us to use zero imputation for missing values. We can see from the histograms above that this transformation has improved the distribution of the variables to be more normal compared to the original variables, but there are still skewed variables from zero imputation.

3. BUILD MODELS

Initial Model - No Changes to Variables -Adj.R .512

```
model_initial <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B +BATTING_3B+
                      BATTING_HR+BATTING_BB+BATTING_SO+BASERUN_SB+ BASERUN_CS+
                      BATTING_HBP +PITCHING_HITS+ PITCHING_HR+PITCHING_BB+
                      PITCHING_SO+FIELD_ERRORS+ FIELD_DBLPLY, data = df_train)

summary(model_initial)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS +
##     BATTING_HBP + PITCHING_HITS + PITCHING_HR + PITCHING_BB +
##     PITCHING_SO + FIELD_ERRORS + FIELD_DBLPLY, data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8708  -5.6564  -0.0599   5.2545  22.9274
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   60.28826   19.67842   3.064  0.00253 **
## BATTING_HITS   1.91348    2.76139   0.693  0.48927
## BATTING_2B     0.02639    0.03029   0.871  0.38484
## BATTING_3B    -0.10118    0.07751  -1.305  0.19348
## BATTING_HR    -4.84371   10.50851  -0.461  0.64542
## BATTING_BB    -4.45969    3.63624  -1.226  0.22167
## BATTING_SO     0.34196    2.59876   0.132  0.89546
## BASERUN_SB     0.03304    0.02867   1.152  0.25071
## BASERUN_CS    -0.01104    0.07143  -0.155  0.87730
## BATTING_HBP    0.08247    0.04960   1.663  0.09815 .
## PITCHING_HITS -1.89096    2.76095  -0.685  0.49432
## PITCHING_HR    4.93043   10.50664   0.469  0.63946
## PITCHING_BB    4.51089    3.63372   1.241  0.21612
## PITCHING_SO   -0.37364    2.59705  -0.144  0.88577
## FIELD_ERRORS  -0.17204    0.04140  -4.155 5.08e-05 ***
## FIELD_DBLPLY  -0.10819    0.03654  -2.961  0.00349 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.467 on 175 degrees of freedom
##   (2085 observations deleted due to missingness)
## Multiple R-squared:  0.5501, Adjusted R-squared:  0.5116
## F-statistic: 14.27 on 15 and 175 DF,  p-value: < 2.2e-16
```

Observations

- This model is significant overall (p < 2.2e-16) with 175 degrees of freedom

- Adjusted R-squared of .512
- 2 of 15 variable coefficients and the intercept coefficient are significant ($p < .05$)
- 2085 observations deleted due to missing values

Let us explore three different models including:

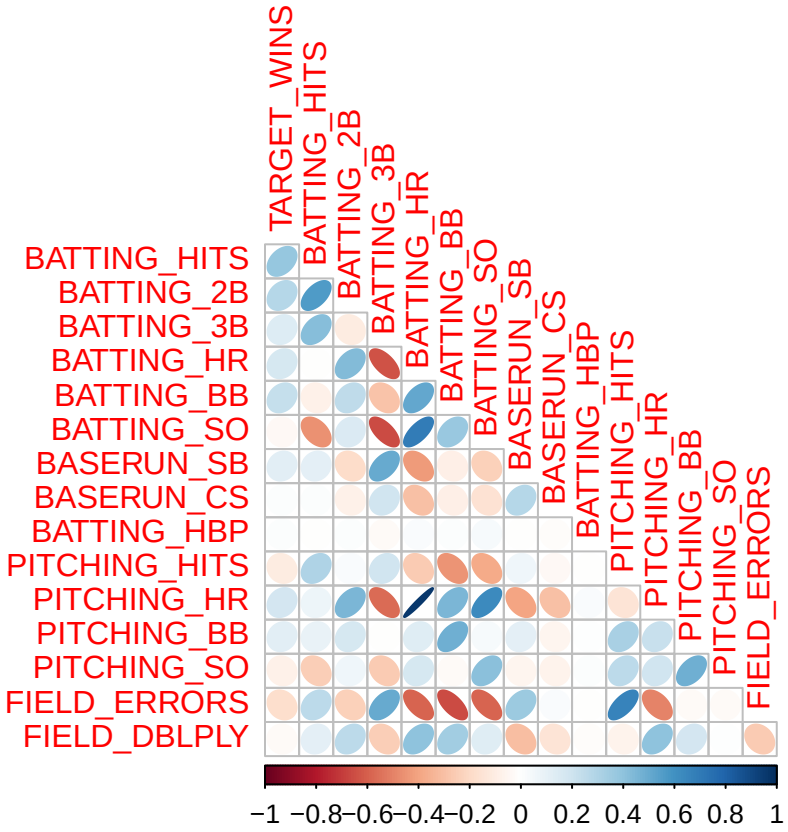
- A - Combining Variables due to Multicollinearity
- B - Backwards Selection and Intuitive Variable Coefficients
- C - Ratio of Wins in 162 Game Season

A - Combining Variables due to Multicollinearity

This model starts with the data that uses the Mean Imputation and focuses on multicollinearity between variables**

Correlations between Variables

```
df_train_mn %>%
  select(-INDEX) %>%
  cor(.,) %>%
  corrplot(.,method = "ellipse", type = "lower", diag = FALSE)
```



Observations

- The correlogram confirms that none of the variable are very strongly correlated with **TARGET_WINS**, with the exception of **BATTING_HITS** which has a modest positive correlation with wins
- Strong multicollinearity is seen between the following variable which needs to be taken into consideration when constructing the models: **FIELD_ERRORS**, **PITCHING_HR**, **BATTING_3B**, **BATTING_HR**
- It may also not be advisable to include **BATTING_HBP** in the model because it has no correlation with wins or any other variable in the dataset

A1 - ALL Variables -Adj.R .315

```
summary(model_mn)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS +
##     BATTING_HBP + PITCHING_HITS + PITCHING_HR + PITCHING_BB +
##     PITCHING_SO + FIELD_ERRORS + FIELD_DBLPLY, data = df_train_mn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.056  -8.639   0.151   8.359  58.647
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.136e+01  6.856e+00   3.116 0.001858 **
## BATTING_HITS   4.821e-02  3.687e-03  13.075  < 2e-16 ***
## BATTING_2B    -2.010e-02  9.152e-03  -2.196 0.028215 *
## BATTING_3B     6.047e-02  1.676e-02   3.608 0.000315 ***
## BATTING_HR     5.292e-02  2.743e-02   1.929 0.053834 .
## BATTING_BB     1.038e-02  5.818e-03   1.784 0.074624 .
## BATTING_SO    -9.415e-03  2.552e-03  -3.690 0.000230 ***
## BASERUN_SB     2.951e-02  4.465e-03   6.610 4.79e-11 ***
## BASERUN_CS    -1.163e-02  1.616e-02  -0.720 0.471658
## BATTING_HBP    6.409e-02  7.304e-02   0.877 0.380349
## PITCHING_HITS -7.302e-04  3.677e-04  -1.986 0.047146 *
## PITCHING_HR    1.491e-02  2.432e-02   0.613 0.540012
## PITCHING_BB    6.613e-05  4.146e-03   0.016 0.987275
## PITCHING_SO    2.846e-03  9.188e-04   3.097 0.001978 **
## FIELD_ERRORS  -2.123e-02  2.481e-03  -8.560  < 2e-16 ***
## FIELD_DBLPLY  -1.210e-01  1.304e-02  -9.275  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.04 on 2260 degrees of freedom
## Multiple R-squared:  0.3192, Adjusted R-squared:  0.3147
## F-statistic: 70.65 on 15 and 2260 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2260 degrees of freedom
- Adjusted R-squared of .315

- 9 of 15 variable coefficients and the intercept coefficient are significant (p < .05)
- 4 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_2B & Negative impact on Wins, but Positive coefficient: PITCHING_HR, PITCHING_BB, FIELD_DBLPLY

A2 - Removed BATTING_HBP for NA Values -Adj.R .315

- Let's test the model removing batters hit by pitch **BATTING_HBP** as more than 90% of values were missing and that the correlation matrix after mean imputation show no correlation with wins or any other variable.

```
model_mn_2 <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_BB+BATTING_SO+
                 BASERUN_SB+BASERUN_CS+FIELD_ERRORS+PITCHING_HR+BATTING_3B+
                   BATTING_HR ## +BATTING_HBP
                +PITCHING_HITS+PITCHING_BB+PITCHING_SO+FIELD_DBLPLY,
                data = df_train_mn)

# Summary of the regression model
summary(model_mn_2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_BB +
##     BATTING_SO + BASERUN_SB + BASERUN_CS + FIELD_ERRORS + PITCHING_HR +
##     BATTING_3B + BATTING_HR + PITCHING_HITS + PITCHING_BB + PITCHING_SO +
##     FIELD_DBLPLY, data = df_train_mn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.027  -8.575   0.137   8.342  58.611
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.507e+01  5.399e+00   4.643 3.63e-06 ***
## BATTING_HITS   4.824e-02  3.687e-03  13.084  < 2e-16 ***
## BATTING_2B    -2.004e-02  9.151e-03  -2.190 0.028604 *
## BATTING_BB     1.041e-02  5.818e-03   1.789 0.073671 .
## BATTING_SO    -9.351e-03  2.550e-03  -3.667 0.000251 ***
## BASERUN_SB     2.946e-02  4.464e-03   6.600 5.12e-11 ***
## BASERUN_CS    -1.173e-02  1.616e-02  -0.726 0.467830
## FIELD_ERRORS  -2.118e-02  2.480e-03  -8.542  < 2e-16 ***
## PITCHING_HR    1.481e-02  2.432e-02   0.609 0.542633
## BATTING_3B     6.040e-02  1.676e-02   3.604 0.000320 ***
## BATTING_HR     5.298e-02  2.743e-02   1.931 0.053548 .
## PITCHING_HITS -7.315e-04  3.676e-04  -1.990 0.046736 *
## PITCHING_BB    8.066e-05  4.145e-03   0.019 0.984477
## PITCHING_SO    2.841e-03  9.187e-04   3.092 0.002009 **
## FIELD_DBLPLY  -1.212e-01  1.304e-02  -9.298  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.04 on 2261 degrees of freedom
## Multiple R-squared:  0.319,  Adjusted R-squared:  0.3148
## F-statistic: 75.65 on 14 and 2261 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2261 degrees of freedom
- Adjusted R-squared is unchanged at .315

- 9 of 14 variable coefficients and the intercept coefficient are significant (p < .05)
- 5 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_2B & Negative impact on Wins, but Positive coefficient: PITCHING_HR, PITCHING_HITS, PITCHING_BB, FIELD_DBLPLY

A3 - Combining Variables with Multicollinearity -Adj.R .243

- When examining the correlations, we observed some multicollinearity among the following variables:

BATTING_HITS and PITCHING_HITS BATTING_HR and PITCHING_HR BATTING_BB and PITCHING_BB BATTING_SO and PITCHING_SO

- Due to the high correlation between these variables, we made the decision to combine them into single variables through addition.

```r
# Creating new data set with combined correlated variables and removed
# correlated variables
df_train_with_combo <- df_train_mn %>%
  mutate(team_H = BATTING_HITS + PITCHING_HITS,
         team_HR = BATTING_HR + PITCHING_HR,
         team_BB = BATTING_BB + PITCHING_BB,
         team_SO = BATTING_SO + PITCHING_SO)%>%
  select(-BATTING_HITS, -PITCHING_HITS, -BATTING_HR, -PITCHING_HR, -BATTING_BB,
         -PITCHING_BB,-BATTING_SO,-BATTING_HBP,- PITCHING_SO)

# Testing this new data set
model_mn_3_combo <- lm(TARGET_WINS ~ BATTING_2B + BATTING_3B
                  + BASERUN_SB + BASERUN_CS + FIELD_ERRORS + FIELD_DBLPLY
                  + team_H + team_HR + team_BB + team_SO,
                  data = df_train_with_combo)

# Summary of the regression model
summary(model_mn_3_combo)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_2B + BATTING_3B + BASERUN_SB +
##     BASERUN_CS + FIELD_ERRORS + FIELD_DBLPLY + team_H + team_HR +
##     team_BB + team_SO, data = df_train_with_combo)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.584  -8.848   0.052   8.566  64.909
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  62.4112789  2.8679760  21.761  < 2e-16 ***
## BATTING_2B    0.0603500  0.0073840   8.173 4.95e-16 ***
## BATTING_3B    0.1703875  0.0154484  11.029  < 2e-16 ***
## BASERUN_SB    0.0308966  0.0044801   6.896 6.90e-12 ***
## BASERUN_CS   -0.0080248  0.0169155  -0.474  0.63526
## FIELD_ERRORS -0.0200766  0.0024923  -8.056 1.27e-15 ***
## FIELD_DBLPLY -0.1030140  0.0135643  -7.595 4.49e-14 ***
## team_H        0.0009416  0.0003152   2.987  0.00285 **
## team_HR       0.0399146  0.0038915  10.257  < 2e-16 ***
## team_BB       0.0045957  0.0014547   3.159  0.00160 **
## team_SO      -0.0032839  0.0005131  -6.401 1.87e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.71 on 2265 degrees of freedom
## Multiple R-squared:  0.2461, Adjusted R-squared:  0.2428
## F-statistic: 73.94 on 10 and 2265 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2265 degrees of freedom
- Adjusted R-squared decreased to .243 (A1 & A2 -> .315)

- 9 of 10 variable coefficients and the intercept coefficient are significant (p < .01)
- 1 Variable coefficients has counter intuitive values: Positive impact on Wins, but Negative coefficient: FIELD_DBLPLY & Negative impact on Wins, but Positive coefficient: NONE
- Counter intuitive to create variables that add batting and pitching stats that should have an opposite impact on Wins

A4 - Remove BASERUN_CS for Non-significance -Adj.R .243

```r
# Testing this new data set
model_mn_4_combo <- lm(TARGET_WINS ~ BATTING_2B + BATTING_3B
                  + BASERUN_SB + FIELD_ERRORS + FIELD_DBLPLY
                  + team_H + team_HR + team_BB + team_SO,
                  data = df_train_with_combo)

# Summary of the regression model
summary(model_mn_4_combo)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_2B + BATTING_3B + BASERUN_SB +
##     FIELD_ERRORS + FIELD_DBLPLY + team_H + team_HR + team_BB +
```

```
##     team_SO, data = df_train_with_combo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.649  -8.815   0.055   8.574  64.906
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  61.9316105  2.6833783  23.080  < 2e-16 ***
## BATTING_2B    0.0601534  0.0073711   8.161 5.46e-16 ***
## BATTING_3B    0.1704151  0.0154456  11.033  < 2e-16 ***
## BASERUN_SB    0.0303981  0.0043544   6.981 3.83e-12 ***
## FIELD_ERRORS -0.0198336  0.0024386  -8.133 6.82e-16 ***
## FIELD_DBLPLY -0.1030898  0.0135610  -7.602 4.24e-14 ***
## team_H        0.0009287  0.0003140   2.958  0.00313 **
## team_HR       0.0403502  0.0037810  10.672  < 2e-16 ***
## team_BB       0.0046580  0.0014485   3.216  0.00132 **
## team_SO      -0.0032830  0.0005130  -6.400 1.88e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.7 on 2266 degrees of freedom
## Multiple R-squared:  0.246,  Adjusted R-squared:  0.243
## F-statistic: 82.16 on 9 and 2266 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2265 degrees of freedom
- Adjusted R-squared unchanged from previous model at .243 (A1 & A2 -> .315, A3 -> .243)

- 9 of 9 variable coefficients and the intercept coefficient are significant (p < .01)
- 1 Variable coefficients has counter intuitive values: Positive impact on Wins, but Negative coefficient: FIELD_DBLPLY & Negative impact on Wins, but Positive coefficient: NONE
- Counter intuitive to create variables that add batting and pitching stats that should have an opposite impact on Wins

A5 - Remove BATTING_HBP after Manual Review -Adj.R .221

- Let's redo the correlations to select the most highly correlated variables.

```
# Create a subset that includes all columns
# all_columns_subset <- df_train_with_combo[, ]
# kdepairs(all_columns_subset)
```

```
model_mn_5_combo <- lm(TARGET_WINS ~ team_H + team_BB + FIELD_ERRORS +
                         FIELD_DBLPLY+ team_SO+ team_HR ##+ BATTING_HBP
            + BATTING_3B +BASERUN_SB, data = df_train_with_combo)

summary(model_mn_5_combo)
```

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

```
## lm(formula = TARGET_WINS ~ team_H + team_BB + FIELD_ERRORS +
##     FIELD_DBLPLY + team_SO + team_HR + BATTING_3B + BASERUN_SB,
##     data = df_train_with_combo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.887  -8.814   0.199   8.594  72.705
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.8455389  2.4862050  28.495  < 2e-16 ***
## team_H        0.0015595  0.0003087   5.052 4.73e-07 ***
## team_BB       0.0046630  0.0014693   3.174  0.00153 **
## FIELD_ERRORS -0.0237537  0.0024252  -9.795  < 2e-16 ***
## FIELD_DBLPLY -0.0960015  0.0137276  -6.993 3.52e-12 ***
## team_SO      -0.0035237  0.0005195  -6.783 1.49e-11 ***
## team_HR       0.0518891  0.0035570  14.588  < 2e-16 ***
## BATTING_3B    0.1984428  0.0152752  12.991  < 2e-16 ***
## BASERUN_SB    0.0298278  0.0044164   6.754 1.82e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.9 on 2267 degrees of freedom
## Multiple R-squared:  0.2239, Adjusted R-squared:  0.2211
## F-statistic: 81.74 on 8 and 2267 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2267 degrees of freedom
- Adjusted R-squared decreased to .221 (A1 & A2 -> .315, A3 & A4 -> .243)

- 8 of 8 variable coefficients and the intercept coefficient are significant (p < .01)
- 1 Variable coefficients has counter intuitive values: Positive impact on Wins, but Negative coefficient: FIELD_DBLPLY & Negative impact on Wins, but Positive coefficient: NONE
- Counter intuitive to create variables that add batting and pitching stats that should have an opposite impact on Wins

Best from A: Model A2 model_mn_2 -Adj.R .315

The second version of the model (model_mn_2) has the highest effect size accounting for 31.5% of the variance in the Wins (TARGET_WINS). Additionally, the overall model and 9 of 14 coefficients are significant, though 5 counter intuitive coefficients are included and there is multicollinearity between predictor variables.

B - Backwards Selection and Intuitive Variable Coefficients

This model starts with a dataset that used Mean Imputation & Square Root Transformed Variables and then changes the model based on the variable coefficients.

B1 - ALL Variables -Adj.R .355

```
# Fit a multiple linear regression model using lm with square root
#transformed variables
model_sqrt_1 <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+
                     BATTING_HR+BATTING_BB+BATTING_SO+BASERUN_SB+
                   BASERUN_CS+BATTING_HBP+PITCHING_HITS+PITCHING_HR+
                     PITCHING_BB+PITCHING_SO+FIELD_ERRORS+FIELD_DBLPLY,
                 data = df_train_sqrt)

# Summary of the regression model
summary(model_sqrt_1)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS +
##     BATTING_HBP + PITCHING_HITS + PITCHING_HR + PITCHING_BB +
##     PITCHING_SO + FIELD_ERRORS + FIELD_DBLPLY, data = df_train_sqrt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7776 -0.4691  0.0289  0.4777  3.0592
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.838390   0.802792   3.536 0.000415 ***
## BATTING_HITS   0.243593   0.017380  14.016  < 2e-16 ***
## BATTING_2B    -0.041292   0.016465  -2.508 0.012217 *
## BATTING_3B     0.055966   0.015127   3.700 0.000221 ***
## BATTING_HR     0.030747   0.037571   0.818 0.413229
## BATTING_BB    -0.024306   0.024689  -0.984 0.324984
## BATTING_SO    -0.062899   0.013037  -4.825 1.50e-06 ***
## BASERUN_SB     0.058704   0.006748   8.699  < 2e-16 ***
## BASERUN_CS    -0.037046   0.015320  -2.418 0.015680 *
## BATTING_HBP    0.084764   0.064359   1.317 0.187956
## PITCHING_HITS -0.026829   0.004871  -5.508 4.03e-08 ***
## PITCHING_HR    0.035546   0.033722   1.054 0.291961
## PITCHING_BB    0.034832   0.020933   1.664 0.096252 .
## PITCHING_SO    0.034681   0.009031   3.840 0.000126 ***
## FIELD_ERRORS  -0.072716   0.006925 -10.500  < 2e-16 ***
## FIELD_DBLPLY  -0.168645   0.018086  -9.325  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.748 on 2260 degrees of freedom
## Multiple R-squared:  0.3589, Adjusted R-squared:  0.3546
## F-statistic: 84.33 on 15 and 2260 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2260 degrees of freedom

- Adjusted R-squared of .355 which is an improvement from the mean imputation models tested which are .315 or less.
- 10 of 15 variable coefficients and the intercept coefficient are significant ($p < .05$)
- 5 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_2B, BATTING_BB, FIELD_DBLPLY & Negative impact on Wins, but Positive coefficient: PITCHING_BB, PITCHING_HR

B2 - Removed BASERUN_CS and BATTING_HBP for NA Values -Adj.R .328

- Let's explore a model excluding the two variables with the most missing values including base runner caught stealing and batters hit by pitch

```
model_sqrt_2 <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+BATTING_HR+
                     BATTING_BB+BATTING_SO+BASERUN_SB+##BASERUN_CS + BATTING_HBP
                  +PITCHING_HITS+PITCHING_HR+PITCHING_BB+PITCHING_SO+FIELD_ERRORS,
                data = df_train_sqrt)
# Summary of the regression model
summary(model_sqrt_2)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB + +PITCHING_HITS +
##     PITCHING_HR + PITCHING_BB + PITCHING_SO + FIELD_ERRORS, data = df_train_sqrt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8866 -0.4833  0.0303  0.4923  2.9078
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.244593   0.599840   2.075 0.038111 *
## BATTING_HITS   0.245679   0.017738  13.850  < 2e-16 ***
## BATTING_2B    -0.048845   0.016771  -2.913 0.003620 **
## BATTING_3B     0.058233   0.015432   3.773 0.000165 ***
## BATTING_HR    -0.001726   0.037994  -0.045 0.963763
## BATTING_BB    -0.028313   0.025131  -1.127 0.260027
## BATTING_SO    -0.049021   0.013221  -3.708 0.000214 ***
## BASERUN_SB     0.062586   0.006313   9.913  < 2e-16 ***
## PITCHING_HITS -0.025643   0.004962  -5.168 2.57e-07 ***
## PITCHING_HR    0.045659   0.034371   1.328 0.184172
## PITCHING_BB    0.027355   0.021352   1.281 0.200258
## PITCHING_SO    0.033260   0.009215   3.609 0.000314 ***
## FIELD_ERRORS  -0.071576   0.006794 -10.535  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7636 on 2263 degrees of freedom
## Multiple R-squared:  0.3311, Adjusted R-squared:  0.3276
## F-statistic: 93.35 on 12 and 2263 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2263 degrees of freedom
- Adjusted R-squared decreased from .355 to .328

- 8 of 12 variable coefficients and the intercept coefficient are significant (p < .05)
- 5 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_2B, BATTING_BB, BATTING_HR & Negative impact on Wins, but Positive coefficient: PITCHING_BB, PITCHING_HR

B3 - Removed BASERUN_CS, BATTING_HBP, BATTING_HR, BATTING_BB, PITCHING_HR & PITCHING_BB for Non-significance -Adj.R .320

- Let's adjust this model to remove an additional 4 predictor variables that have non-significant coefficients including Batter & Pitcher home runs & walks

```
model_sqrt_3 <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+
                   BATTING_SO+BASERUN_SB+##BASERUN_CS + BATTING_HBP
           +PITCHING_HITS##+ BATTING_HR + BATTING_BB + PITCHING_HR + PITCHING_BB
                 +PITCHING_SO+FIELD_ERRORS,
               data = df_train_sqrt)
# Summary of the regression model
summary(model_sqrt_3)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_SO + BASERUN_SB + +PITCHING_HITS + PITCHING_SO +
##     FIELD_ERRORS, data = df_train_sqrt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9282 -0.4756  0.0200  0.5160  2.8059
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.050976   0.505878  -0.101 0.919745
## BATTING_HITS   0.278676   0.015907  17.519  < 2e-16 ***
## BATTING_2B    -0.049467   0.016794  -2.946 0.003256 **
## BATTING_3B     0.040506   0.014386   2.816 0.004909 **
## BATTING_SO    -0.037525   0.009850  -3.810 0.000143 ***
## BASERUN_SB     0.057276   0.005835   9.816  < 2e-16 ***
## PITCHING_HITS -0.017733   0.003679  -4.820 1.53e-06 ***
## PITCHING_SO    0.037464   0.006366   5.885 4.56e-09 ***
## FIELD_ERRORS  -0.077277   0.006083 -12.705  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7678 on 2267 degrees of freedom
## Multiple R-squared:  0.3224, Adjusted R-squared:  0.3201
## F-statistic: 134.9 on 8 and 2267 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2267 degrees of freedom
- Adjusted R-squared decreased slightly to .320 (B1 -> .355, B2 -> .328)
- 8 of 8 variable coefficients are significant (p < .01), though the intercept coefficient is not (p = .92)
- 1 Variable coefficient has counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_2B & Negative impact on Wins, but Positive coefficient: NONE

B4 - Added BATTING_HBP to increase effect size -Adj.R .320

- Let's try adding back in batters hit by pitch (BATTING_HBP) to see if that will increase our effect size back to when all variables were included.

```
model_sqrt_4 <- lm(TARGET_WINS ~ BATTING_HITS+BATTING_2B+BATTING_3B+BATTING_SO+
                     BASERUN_SB+ ##BASERUN_CS
                   + BATTING_HBP+PITCHING_HITS
                   ##+ BATTING_HR + BATTING_BB + PITCHING_HR + PITCHING_BB
                   +PITCHING_SO+FIELD_ERRORS,
                 data = df_train_sqrt)
# Summary of the regression model
summary(model_sqrt_4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_SO + BASERUN_SB + +BATTING_HBP + PITCHING_HITS +
##     PITCHING_SO + FIELD_ERRORS, data = df_train_sqrt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9281 -0.4768  0.0190  0.5160  2.8072
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.770099   0.712087  -1.081 0.279604
## BATTING_HITS   0.278462   0.015904  17.509  < 2e-16 ***
## BATTING_2B    -0.049408   0.016790  -2.943 0.003286 **
## BATTING_3B     0.040922   0.014385   2.845 0.004485 **
## BATTING_SO    -0.037652   0.009848  -3.823 0.000135 ***
## BASERUN_SB     0.057335   0.005834   9.828  < 2e-16 ***
## BATTING_HBP    0.094702   0.066013   1.435 0.151540
## PITCHING_HITS -0.017685   0.003678  -4.808 1.62e-06 ***
## PITCHING_SO    0.037480   0.006364   5.889 4.46e-09 ***
## FIELD_ERRORS  -0.077516   0.006083 -12.742  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7676 on 2266 degrees of freedom
## Multiple R-squared:  0.3231, Adjusted R-squared:  0.3204
## F-statistic: 120.2 on 9 and 2266 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2266 degrees of freedom

- Adjusted R-squared unchanged at .320 (B1 -> .355, B2 -> .328, B3 -> .320)
- 8 of 9 variable coefficients are significant (p < .01), and the intercept coefficient is still not significant, but has improved (p = .280)
- 1 Variable coefficient has counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_2B & Negative impact on Wins, but Positive coefficient: NONE

Best from B: Model B4 model_sqrt_4 -Adj.R .320

The fourth version of the square root transformation model (model_sqrt_4) explains 32% of the variance in Wins. Although this is 3.5% less variance than the first model, this fourth model ties more closely to what we would expect of the coefficient values and includes only significant predictor variables.

C - Ratio of Wins in 162 Game Season

This model starts with a dataset with Zero Imputation & Ratio Transformed Variables (df_train_ratio). The response variable (number of wins) has been converted to a ratio of wins out of a 162 game season. Additionally, the variable Total Plays was introduced that totals number of plays for each observation aka team. Total Plays sums the number of pitching hits, batter hits and the other Base runner and fielding variables. The remaining variables were converted to a ratio using Total Plays as the denominator.

C1 - ALL Ratio Variables -Adj.R .232

```
# Fit a multiple linear regression model
model_ratio_1 <- lm(TARGET_WINS_RATIO ~ BATTING_HITS+BATTING_2B+BATTING_3B+
                      BATTING_HR+BATTING_BB+BATTING_SO+BASERUN_SB+
                  BASERUN_CS+BATTING_HBP+PITCHING_HITS+PITCHING_HR+
                    PITCHING_BB+PITCHING_SO+FIELD_ERRORS+FIELD_DBLPLY,
                data = df_train_ratio)

# Summary of the regression model
summary(model_ratio_1)
```

```
##
## Call:
## lm(formula = TARGET_WINS_RATIO ~ BATTING_HITS + BATTING_2B +
##     BATTING_3B + BATTING_HR + BATTING_BB + BATTING_SO + BASERUN_SB +
##     BASERUN_CS + BATTING_HBP + PITCHING_HITS + PITCHING_HR +
##     PITCHING_BB + PITCHING_SO + FIELD_ERRORS + FIELD_DBLPLY,
##     data = df_train_ratio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43059 -0.05337  0.00114  0.05479  0.37273
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.17491    0.37364 -11.173  < 2e-16 ***
## BATTING_HITS   5.01341    0.43238  11.595  < 2e-16 ***
```

```
## BATTING_2B     1.15556    0.37045   3.119  0.00184 **
## BATTING_3B     4.62706    0.70287   6.583 5.71e-11 ***
## BATTING_HR    -0.07382    1.86734  -0.040  0.96847
## BATTING_BB     4.84746    0.65622   7.387 2.10e-13 ***
## BATTING_SO     4.00362    0.40006  10.007  < 2e-16 ***
## BASERUN_SB     4.76018    0.34628  13.747  < 2e-16 ***
## BASERUN_CS     4.70532    0.60337   7.798 9.49e-15 ***
## BATTING_HBP    2.24262    0.88266   2.541  0.01113 *
## PITCHING_HITS  4.72716    0.38085  12.412  < 2e-16 ***
## PITCHING_HR    3.86052    1.73189   2.229  0.02591 *
## PITCHING_BB    4.41518    0.52147   8.467  < 2e-16 ***
## PITCHING_SO    4.74611    0.41432  11.455  < 2e-16 ***
## FIELD_ERRORS   2.78209    0.32868   8.464  < 2e-16 ***
## FIELD_DBLPLY        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08523 on 2261 degrees of freedom
## Multiple R-squared:  0.2364, Adjusted R-squared:  0.2316
## F-statistic: 49.99 on 14 and 2261 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2261 degrees of freedom
- This model has an Adjusted R-squared of .232

- 13 of 15 variable coefficients and the Intercept coefficient are significant (p < .05)
- 7 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_HR & Negative impact on Wins, but Positive coefficient: BATTING_SO, BASERUN_CS, PITCHING_HITS, PITCHING_BB, FIELD_ERRORS & No Coefficient: FIELD_DBLPLY

C2 - Removed FIELD_DBLPLY & BATTING_HR for Non-significance -Adj.R .232

- Let us explore a model excluding the variables that do not have significant coefficients including field doubleplays, and batter home runs (FIELD_DBLPLY & BATTING_HR)

```
# Fit a multiple linear regression model using lm
model_ratio_2 <- lm(TARGET_WINS_RATIO ~ BATTING_HITS+BATTING_2B+BATTING_3B
                    ## +BATTING_HR + FIELD_DBLPLY
                    +BATTING_BB+BATTING_SO+BASERUN_SB
                    +BASERUN_CS+BATTING_HBP+PITCHING_HITS+PITCHING_HR+
                      PITCHING_BB+PITCHING_SO+FIELD_ERRORS,
                data = df_train_ratio)

# Summary of the regression model
summary(model_ratio_2)
```

```
##
## Call:
## lm(formula = TARGET_WINS_RATIO ~ BATTING_HITS + BATTING_2B +
##     BATTING_3B + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS +
##     BATTING_HBP + PITCHING_HITS + PITCHING_HR + PITCHING_BB +
```

```
##     PITCHING_SO + FIELD_ERRORS, data = df_train_ratio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43046 -0.05343  0.00114  0.05481  0.37282
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.1744     0.3734 -11.181  < 2e-16 ***
## BATTING_HITS    5.0129     0.4321  11.601  < 2e-16 ***
## BATTING_2B      1.1559     0.3703   3.122  0.00182 **
## BATTING_3B      4.6306     0.6970   6.644 3.82e-11 ***
## BATTING_BB      4.8372     0.6025   8.029 1.56e-15 ***
## BATTING_SO      4.0027     0.3993  10.023  < 2e-16 ***
## BASERUN_SB      4.7596     0.3459  13.762  < 2e-16 ***
## BASERUN_CS      4.7058     0.6031   7.803 9.17e-15 ***
## BATTING_HBP     2.2368     0.8700   2.571  0.01020 *
## PITCHING_HITS   4.7265     0.3804  12.424  < 2e-16 ***
## PITCHING_HR     3.7934     0.3355  11.308  < 2e-16 ***
## PITCHING_BB     4.4235     0.4768   9.278  < 2e-16 ***
## PITCHING_SO     4.7456     0.4140  11.462  < 2e-16 ***
## FIELD_ERRORS    2.7816     0.3284   8.471  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08521 on 2262 degrees of freedom
## Multiple R-squared:  0.2364, Adjusted R-squared:  0.232
## F-statistic: 53.86 on 13 and 2262 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2262 degrees of freedom
- Adjusted R-squared is unchanged at .232

- 13 of 13 variable coefficients and the Intercept coefficient are significant (p < .05)
- 4 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: NONE & Negative impact on Wins, but Positive coefficient: BASERUN_CS, PITCHING_HITS, PITCHING_HR, PITCHING_BB

C3 - Removed FIELD_DBLPLY, BATTING_HR & BATTING_HBP for Least significance -Adj.R .230

- Let's adjust this model to remove an additional predictor variables that has the least significant variable coefficient batter hit by pitch (BATTING_HBP p = .01) and was also the variable missing over 90% of values.

```
# Fit a multiple linear regression model using lm
model_ratio_3 <- lm(TARGET_WINS_RATIO ~ BATTING_HITS+BATTING_2B+BATTING_3B
                    ## +BATTING_HR + FIELD_DBLPLY + BATTING_HBP
                    +BATTING_BB+BATTING_SO+BASERUN_SB
                    +BASERUN_CS+PITCHING_HITS+PITCHING_HR+PITCHING_BB+
                      PITCHING_SO+FIELD_ERRORS,
```

```
               data = df_train_ratio)

# Summary of the regression model
summary(model_ratio_3)
```

```
##
## Call:
## lm(formula = TARGET_WINS_RATIO ~ BATTING_HITS + BATTING_2B +
##     BATTING_3B + BATTING_BB + BATTING_SO + BASERUN_SB + BASERUN_CS +
##     PITCHING_HITS + PITCHING_HR + PITCHING_BB + PITCHING_SO +
##     FIELD_ERRORS, data = df_train_ratio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43303 -0.05412  0.00163  0.05520  0.37179
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.8070     0.3454 -11.023  < 2e-16 ***
## BATTING_HITS    4.5580     0.3947  11.548  < 2e-16 ***
## BATTING_2B      1.4507     0.3525   4.115 4.00e-05 ***
## BATTING_3B      4.7008     0.6973   6.741 1.98e-11 ***
## BATTING_BB      4.3944     0.5780   7.602 4.23e-14 ***
## BATTING_SO      3.6856     0.3803   9.692  < 2e-16 ***
## BASERUN_SB      4.5011     0.3313  13.585  < 2e-16 ***
## BASERUN_CS      4.1212     0.5593   7.369 2.40e-13 ***
## PITCHING_HITS   4.3579     0.3528  12.352  < 2e-16 ***
## PITCHING_HR     3.9361     0.3312  11.883  < 2e-16 ***
## PITCHING_BB     4.0704     0.4571   8.905  < 2e-16 ***
## PITCHING_SO     4.3448     0.3841  11.313  < 2e-16 ***
## FIELD_ERRORS    2.4865     0.3080   8.072 1.11e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08532 on 2263 degrees of freedom
## Multiple R-squared:  0.2341, Adjusted R-squared:  0.2301
## F-statistic: 57.65 on 12 and 2263 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2263 degrees of freedom
- Adjusted R-squared decreased very slightly to .230 (C1 & C2 -> .232)
- 12 of 12 variable coefficients and the Intercept coefficient are significant (p < .001)
- 5 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: NONE & Negative impact on Wins, but Positive coefficient: BATTING_SO, BASERUN_CS, PITCHING_HITS, PITCHING_HR, PITCHING_BB

C4 - Removed BATTING_SO, BASERUN_CS, PITCHING_HITS, PITCHING_HR, PITCHING_BB for Counter intuitive coefficients -Adj.R .105

- Let's adjust this model to remove additional predictor variables that should have a negative impact on Wins, but have a positive coefficient (Batting strikeouts, base runner caught stealing, and pitching hits, home runs & walks)

```r
# Fit a multiple linear regression model using lm
model_ratio_4 <- lm(TARGET_WINS_RATIO ~ BATTING_HITS+BATTING_2B+BATTING_3B
                    ## +BATTING_HR + FIELD_DBLPLY + BATTING_HBP
                    +BATTING_BB+BASERUN_SB
                 ## +BATTING_SO+BASERUN_CS+PITCHING_HITS+PITCHING_HR+PITCHING_BB
                    + PITCHING_SO +  FIELD_ERRORS,
                data = df_train_ratio)

# Summary of the regression model
summary(model_ratio_4)
```

```
##
## Call:
## lm(formula = TARGET_WINS_RATIO ~ BATTING_HITS + BATTING_2B +
##     BATTING_3B + BATTING_BB + BASERUN_SB + PITCHING_SO + FIELD_ERRORS,
##     data = df_train_ratio)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46714 -0.05829  0.00217  0.05986  0.43548
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.55023    0.03243  16.966  < 2e-16 ***
## BATTING_HITS -0.27310    0.11660  -2.342  0.01926 *
## BATTING_2B    1.88110    0.37689   4.991 6.46e-07 ***
## BATTING_3B    4.29027    0.67122   6.392 1.98e-10 ***
## BATTING_BB   -0.24904    0.14038  -1.774  0.07619 .
## BASERUN_SB    0.48089    0.15839   3.036  0.00242 **
## PITCHING_SO  -0.34808    0.07670  -4.538 5.97e-06 ***
## FIELD_ERRORS -1.18547    0.13229  -8.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09199 on 2268 degrees of freedom
## Multiple R-squared:  0.1077, Adjusted R-squared:  0.105
## F-statistic: 39.12 on 7 and 2268 DF,  p-value: < 2.2e-16
```

Observations

- Model overall is significant (p < 2.2e-16) with 2268 degrees of freedom
- Adjusted R-squared decreased to .105 (C1 & C2 -> .232, C3 -> .230)
- 6 of 7 variable coefficients and the Intercept coefficient are significant (p < .05)
- 4 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: BATTING_HITS, BATTING_BB & Negative impact on Wins, but Positive coefficient: PITCHING_SO, FIELD_ERRORS

Best from C: Model C3 model_ratio_3 -Adj.R .230

The third version of the model (model_ratio_3) has essentially the same effect size as the previous 2 and is .13 higher than the fourth model. It accounts for 23% of the variance in the ratio of wins (TARGET_WINS_RATIO) and the overall model and coefficients are all significant, though 5 counter intuitive coefficients are included.

4. SELECT MODELS

Selection Criteria to Consider

As all three models are significant overall with similar degrees of freedom, we will focus on: * Adjusted R-squared value * Significance of Variable Coefficients * Variable Coefficients are Intuitive

Combining Variables due to Multicollinearity (Model A2):

- Adjusted R-squared of .315

- 9 of 14 variable coefficients and the intercept coefficient are significant ($p < .05$)
- 5 Variable coefficients have counter intuitive values: Positive impact on Wins, but Negative coefficient: batter doubles (BATTING_2B) & Negative impact on Wins, but Positive coefficient: field double play and pitching hits, home runs, & walks (PITCHING_HR, PITCHING_HITS, PITCHING_BB, FIELD_DBLPLY)

Backwards Selection & Intuitive Variable Coefficient (Model B4):

- Adjusted R-squared of .320
- 8 of 9 variable coefficients are significant ($p < .01$) though intercept coefficient is not
- 1 Variable coefficient has counter intuitive values as we would expect batter doubles (BATTING_2B) to have a positive impact on Wins, but it has a negative coefficient.

Percent Wins in 162 Game Seasons (Model C3):

- Adjusted R-squared of .230
- 12 of 12 variable coefficients and the Intercept coefficient are significant ($p < .001$)
- 5 Variable coefficients have counter intuitive values as we would predict a negative impact on Wins for batter strikeouts, base runners caught stealing, and pitching hits, home runs & walks (BATTING_SO, BASERUN_CS, PITCHING_HITS, PITCHING_HR, PITCHING_BB)

Selected Model

We chose Model B4 that incorporated backwards selection and focusing on variable coefficients being intuitive. It has the greatest adjusted R-squared accounting for 32% of the variance in Wins. It is also the model that makes the most intuitive sense overall as it focused on maximizing intuitiveness of the model through variable coefficient values, which are also all significant barring one.

**Regression Summary for Selected Model

```
summary(model_sqrt_4)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ BATTING_HITS + BATTING_2B + BATTING_3B +
##     BATTING_SO + BASERUN_SB + +BATTING_HBP + PITCHING_HITS +
##     PITCHING_SO + FIELD_ERRORS, data = df_train_sqrt)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -3.9281 -0.4768  0.0190  0.5160  2.8072
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.770099   0.712087  -1.081 0.279604
## BATTING_HITS   0.278462   0.015904  17.509  < 2e-16 ***
## BATTING_2B    -0.049408   0.016790  -2.943 0.003286 **
## BATTING_3B     0.040922   0.014385   2.845 0.004485 **
## BATTING_SO    -0.037652   0.009848  -3.823 0.000135 ***
## BASERUN_SB     0.057335   0.005834   9.828  < 2e-16 ***
## BATTING_HBP    0.094702   0.066013   1.435 0.151540
## PITCHING_HITS -0.017685   0.003678  -4.808 1.62e-06 ***
## PITCHING_SO    0.037480   0.006364   5.889 4.46e-09 ***
## FIELD_ERRORS  -0.077516   0.006083 -12.742  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7676 on 2266 degrees of freedom
## Multiple R-squared:  0.3231, Adjusted R-squared:  0.3204
## F-statistic: 120.2 on 9 and 2266 DF,  p-value: < 2.2e-16
```

```
# Save coefficients from Multiple Regression
df_coeff <- as.data.frame(model_sqrt_4$coefficients) %>%
  t()
```

Calculate Mean Squared Error

```
model_summ <-summary(model_sqrt_4)
#calculate MSE
print(mean(model_summ$residuals^2))
```

```
## [1] 0.5866497
```

Bivariate Plots of Wins by Significant Predictors

```
p1 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=BATTING_HITS)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY BATTING HITS",
       x="TOTAL HITS", y="WINS") +
  theme_bw()
p2 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=BATTING_2B)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY BATTING DOUBLES",
       x="DOUBLES HIT", y="WINS") +
```

```
  theme_bw()
p3 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=BATTING_3B)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY BATTING TRIPLES",
       x="TRIPLES HIT", y="WINS") +
  theme_bw()
p4 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=BATTING_SO)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY BATTING STRIKEOUTS",
       x="BATTER STRIKEOUTS", y="WINS") +
  theme_bw()
p5 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=BASERUN_SB)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY BASE RUNNER STOLEN BASES",
       x="STOLEN BASES", y="WINS") +
  theme_bw()
p6 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=PITCHING_HITS)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY HITS OFF PITCHER",
       x="TOTAL HITS", y="WINS") +
  theme_bw()
p7 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=PITCHING_SO)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY PITCHER STRIKEOUTS",
       x="PITCHER STRIEKOUTS", y="WINS") +
  theme_bw()
p8 <- model_sqrt_4 %>%
  ggplot(aes(y=TARGET_WINS, x=FIELD_ERRORS)) +
  geom_point(color="darkgreen") +
  geom_smooth(method = "lm", se=TRUE) +
  labs(title = "WINS BY FIELDING ERRORS",
       x="FIELD ERRORS", y="WINS") +
  theme_bw()

library(gridExtra)
```

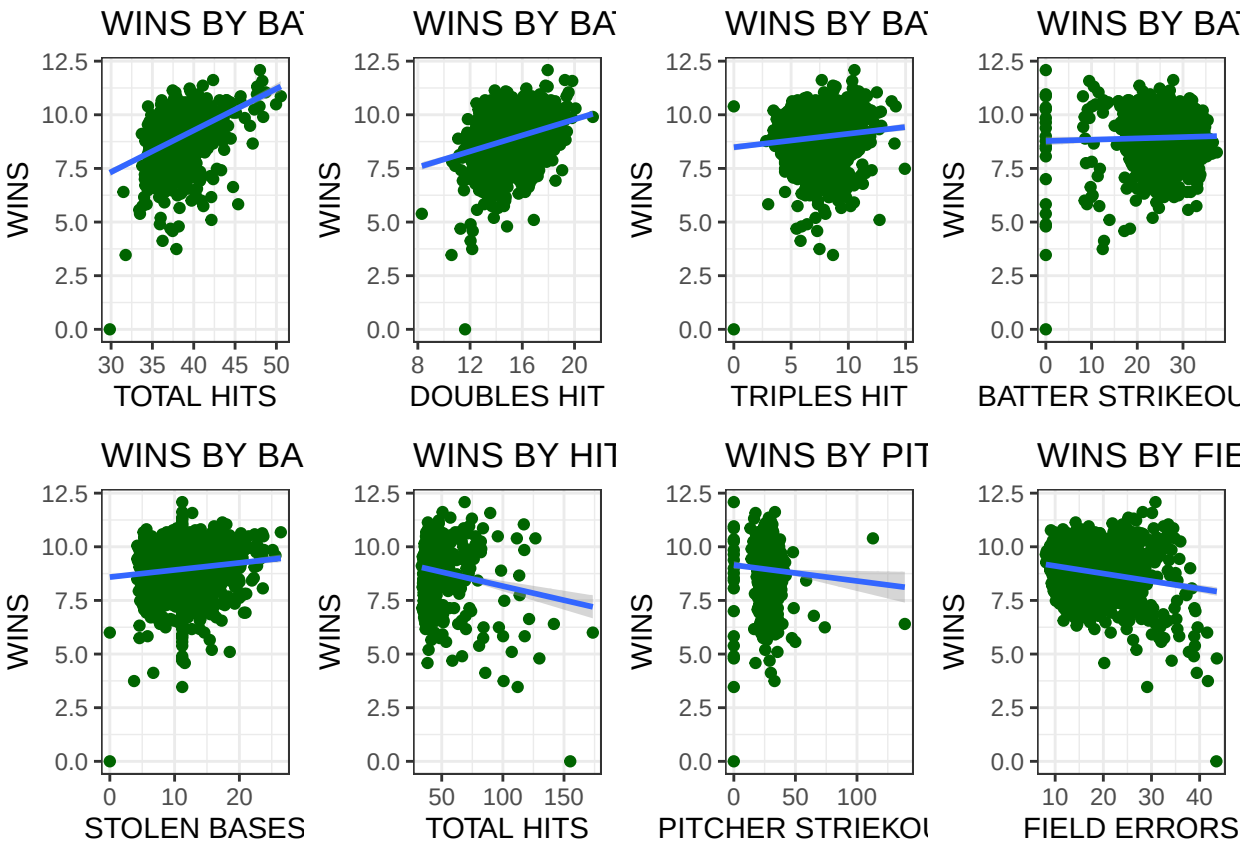
```
##
## Attaching package: 'gridExtra'
```

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

```
grid.arrange(p1, p2, p3, p4, p5, p6, p7,p8, nrow=2)
```

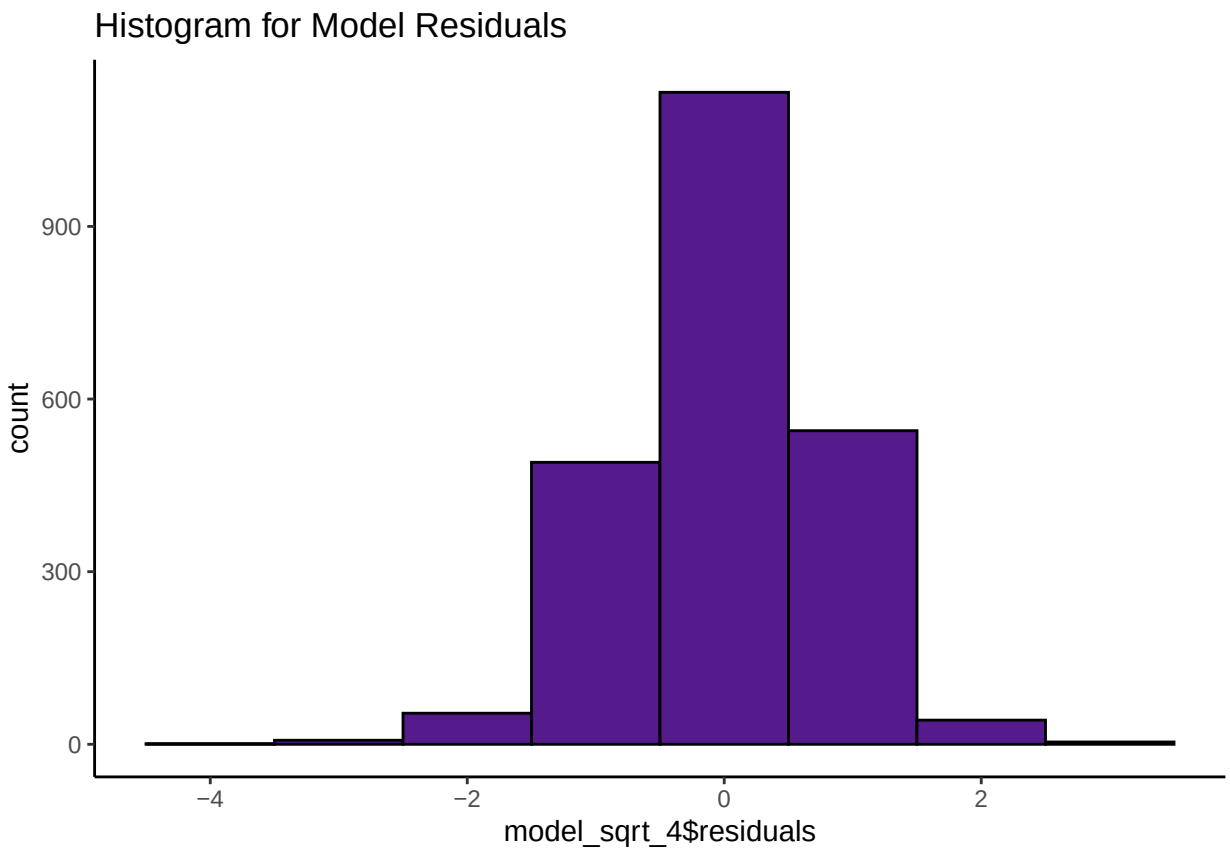
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

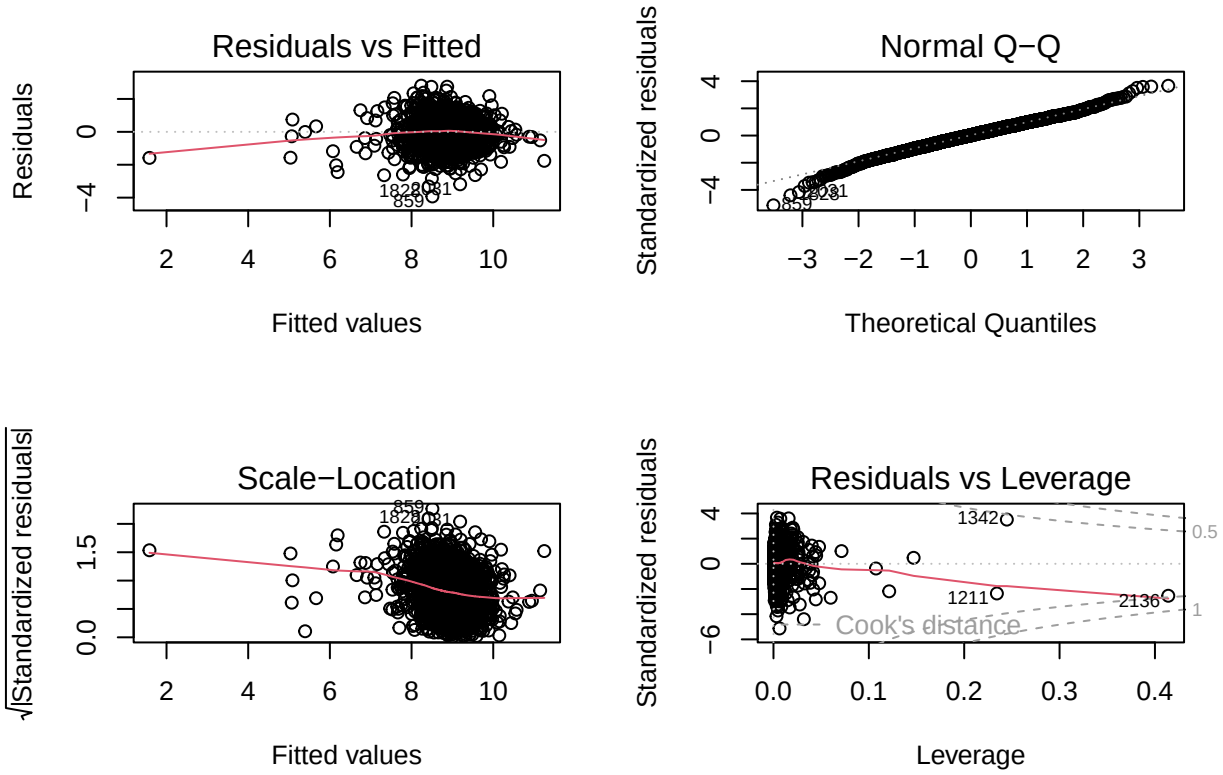


Diagnostic Plots for Selected Model

```
ggplot(data=model_sqrt_4, aes(model_sqrt_4$residuals)) +
  geom_histogram(binwidth = 1, color = "black", fill = "purple4") +
  theme(panel.background = element_rect(fill = "white"),
        axis.line.x=element_line(),
        axis.line.y=element_line()) +
  ggtitle("Histogram for Model Residuals")
```




```
par(mfrow = c(2, 2))
plot(model_sqrt_4)
```



Observations

A - Mean squared Error: 0.5866497 B - Adjusted R-squared is .320 indicating the model accounts for 32% of the variability in our training dataset C - F-statistic: 120.2 on 9 and 2266 DF, p-value: < 2.2e-16 D - Residual Plots: * Nearly Normal Residuals - Condition is met based on the histogram and normal probability plots though the 2 ends diverge on the Q-Q plot * Linearity and Constant Variability - There is no apparent pattern in the residuals plot indicating there is linearity and the points are scattered around zero for the most part showing constant variability * Leverage points - There are several bad leverage points affecting our model as indicated by Cook's distance

Predict Total Wins for Evaluation Dataset

Summary of Variables

```
print(skim(df_evaluation))
```

```
## -- Data Summary ------------------------
##                            Values
## Name                       df_evaluation
## Number of rows             259
## Number of columns          16
## _______________________
## Column type frequency:
```

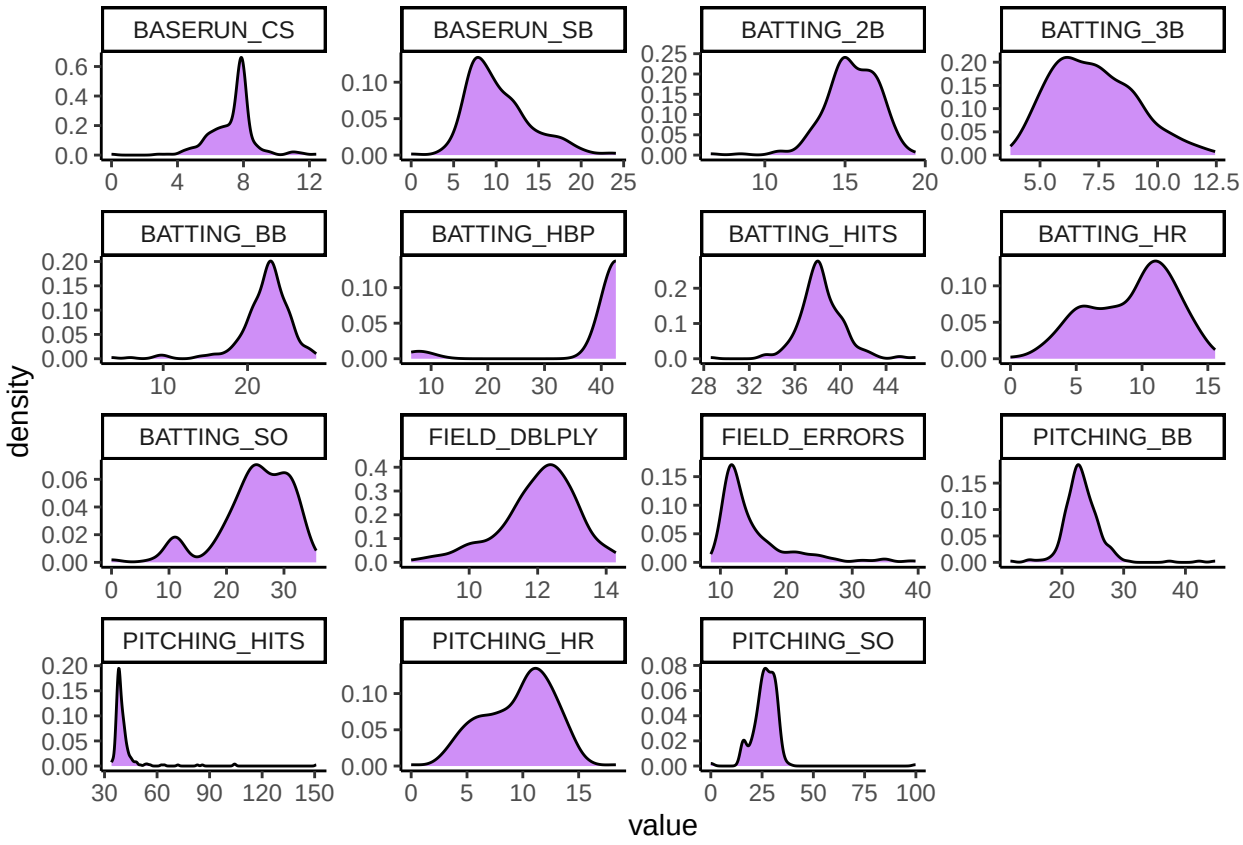
```
##   numeric                  16
## ________________________
## Group variables            None
##
## -- Variable type: numeric -------------------------------------------------------
##    skim_variable n_missing complete_rate   mean     sd   p0    p25    p50    p75
##  1 INDEX                 0        1      1264.   693.     9  708   1249   1832.
##  2 BATTING_HITS          0        1      1469.   151.   819 1387   1455   1548
##  3 BATTING_2B            0        1       241.    49.5   44  210    239    278.
##  4 BATTING_3B            0        1        55.9   27.1   14   35     52     72
##  5 BATTING_HR            0        1        95.6   56.3    0   44.5  101    136.
##  6 BATTING_BB            0        1       499.   121.    15  436.   509    566.
##  7 BATTING_SO           18        0.931   709.   243.     0  545    686    912
##  8 BASERUN_SB           13        0.950   124.    93.4    0   59     92    152.
##  9 BASERUN_CS           87        0.664    52.3   23.1    0   38     49.5   63
## 10 BATTING_HBP         240        0.0734   62.4   12.7   42   53.5   62     67.5
## 11 PITCHING_HITS         0        1      1813.  1663.  1155 1426.  1515   1681
## 12 PITCHING_HR           0        1       102.    57.7    0   52    104    142.
## 13 PITCHING_BB           0        1       552.   173.   136  471    526    606.
## 14 PITCHING_SO          18        0.931   800.   634.     0  613    745    938
## 15 FIELD_ERRORS          0        1       250.   231.    73  131    163    252
## 16 FIELD_DBLPLY         31        0.880   146.    25.9   69  131    148    164
##     p100 hist
##  1  2525
##  2  2170
##  3   376
##  4   155
##  5   242
##  6   792
##  7  1268
##  8   580
##  9   154
## 10    96
## 11 22768
## 12   336
## 13  2008
## 14  9963
## 15  1568
## 16   204
```

Mean Imputation then Square Root Transformation

```
# Get the Means of columns in Data
evaluation_means<-sapply(df_evaluation, function(x) round(mean(x, na.rm =TRUE)))

# Replace NA values in 'column_name' with 'mean'
df_evaluation_mn <- df_evaluation %>%
  mutate(BATTING_SO =
           ifelse(is.na(BATTING_SO),
                  evaluation_means[8],BATTING_SO))%>%
  mutate(BASERUN_SB =
           ifelse(is.na(BASERUN_SB),
```

```
                  evaluation_means[9], BASERUN_SB))%>%
  mutate(BASERUN_CS =
           ifelse(is.na(BASERUN_CS),
                  evaluation_means[10], BASERUN_CS))%>%
  mutate(BATTING_HBP =
           ifelse(is.na(BATTING_HBP),
                  evaluation_means[11],BATTING_HBP))%>%
  mutate(PITCHING_SO =
           ifelse(is.na(PITCHING_SO),
                  evaluation_means[15], PITCHING_SO))%>%
  mutate(FIELD_DBLPLY =
           ifelse(is.na(FIELD_DBLPLY),
                  evaluation_means[17], FIELD_DBLPLY))

df_evaluation_sqrt <- sqrt(df_evaluation_mn)

df_evaluation_sqrt %>%
  #pivot longer to plot all variables
  gather(variable, value, BATTING_HITS: FIELD_DBLPLY)%>%
  ggplot(.,aes(x=value)) + #plotting every variable
  geom_density(fill = "purple", alpha = 0.5) +
  facet_wrap(~ variable, scales = "free", ncol = 4) +
  theme_classic()
```

```
## Warning: Removed 31 rows containing non-finite values (`stat_density()`).
```



Predict Total Wins

- Use Regression Model built using Training data to Predict Wins for the Evaluation data

```
df_evaluation_sqrt$PREDICT_WINS_sqrt =
  predict(model_sqrt_4, new = df_evaluation_sqrt)
df_evaluation_sqrt$PREDICT_WINS =
  (df_evaluation_sqrt$PREDICT_WINS_sqrt)*(df_evaluation_sqrt$PREDICT_WINS_sqrt)
```

Reference

- "Pythagorean Theorem of Baseball." Baseball Reference, https://www.baseball-reference.com/bullpen/Pythagorean_Theorem_of_Baseball. Accessed 11 September 2023.
- No author listed. "Pythagorean Expectation in Major League Baseball." Digital Commons @ Cal Poly, https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1067&context=statsp. Accessed 11 September 2023.