A Different Way of Expected Wins

John Dattilo

Introduction:
I downloaded baseball statistics from Fangraph.com to gather data from the 2015 to 2019 seasons on how teams performed during each season. I originally wanted to see how these statistics would help predict the amount of runs a team would score or give up during the course of a season. I later decided that I want to create a model to predict the amount of wins a team would have and then compare this prediction to Bill James expected win-loss formula: Expected Wins = Number of Games * W%=[(Runs Scored)ˆ1.83]/[(Runs Scored)ˆ1.83 + (Runs Allowed)ˆ1.83], with an average difference slightly over 3 games per season. I ended up creating a multiple linear regression model that on average predicted the amount of wins within 4.36 games.

Reading and Manipulating Data:
I had to download two different csv files for each team, one for hitting and another for pitching statistics. For both files I had to clean the data such that I had to remove the percentage sign within the data and then I changed the datatype for those vairables from a character to numeric.I also renamed many of the column names to make the variables much easier to understand which team statistics the variable belonged too. I finally merged the two data files to be combined into one by using two variables, the team name and the year of the season which allowed for every team to have one row of variables per season.

Summary Statistics:
When looking the summary statistics of the data I noticed that average of the Weighted Runs Created Plus was not equal to 100, this is due to the fact that this statistic is adjusted for park factors which changes year to year, I will take caution when using the variable in the models. I also noticed that the average number of wins is 80.97 which is very close to what I expected of 81. This small difference is due to some teams not being able making up a game. The last thing I noticed that average team hitting stats such as hard hit percentage, flyball percentage, groundball percentage, and line drive percentage had the same average as the same pitching stats suggesting that there is no errors in these stats I also created a function to display the correlations between wins and all of the other stats to get a good idea at which stats to include in the model.

Modeling:
I decided to use a multiple regression model over a poisson regression model even though poisson models is used for count data because the assumption that the mean of the mean of the response variable Wins does not equal to the variance. I ended up running a step wise regression, which is a repetitive process which adds and removing variables until adding or removing variable does not improve the AIC (Akaike Information Criterion), which the lowest score means that the model is more likely to be the best model over the others in the dataset. I split up the data into a training set and a testing set such that I can make predictions on data that the model has not seen in order, as testing on data the model has seen can result in overfitting.The first model I came up with was

$$\begin{aligned}\text{Wins} = \alpha &+ \beta_1(\text{WHIP\_P}) + \beta_2(\text{wOBA\_H}) + \beta_3(\text{`LOB\%\_P`}) + \\ &\beta_4(\text{`Flyball\%\_P`}) + \beta_5(\text{`HardHit\%\_H`}) + \beta_6(\text{`HR/FB\_P`}) + \beta_7(\text{OPS\_H}) +. \\ &\beta_8(\text{SLG\_H}) + \beta_9(\text{ExitVelo\_P}) + \epsilon\end{aligned}$$

The probably with this model was that the variable OPS_H, SLG_H, and wOBA_H all had high variance inflation factors, which violates the assumption that all the predictors are independent from one another. In order to fix this solution I decided to only keep one of the three variables in the final model, which I choose to keep OPS_H instead of SLG_H or wOBA_H as OPS_H was highly significant in the model. I ended up with a final model of:

$$\text{Wins} = \alpha + \beta_1(\text{WHIP\_P}) + \beta_2(\text{OPS\_H}) + \beta_3(\text{`LOB\%\_P`}) + \\ \beta_4(\text{`Flyball\%\_P`}) + \beta_5(\text{`HardHit\%\_H`}) + \beta_6(\text{`HR/FB\_P`}) + \epsilon$$

This model passes all of the assumptions of linear regression that:
1. The residual vs fitted plot is approximately horizontal at 0 suggesting a linear relationship between the response variable and the predictor variables.
2. The homogeneity of variance plot is an approximately flat line suggesting that there is constant variances within the residuals
3. There is no high variance inflation factors suggesting at the predictors are independent from one another
4. The distribution of residuals follow the normal distribution validating the assumption that the residuals are normally distributed.

Conclusions:
The final model ended up predicting wins with an average error of 4.36 wins. This unfortunately does not improve on Bill Jame's expected win loss model that ends up with an average error of 1 less win compared to my multiple linear regression model. I also trying building tree-based models but they did not perform as well as the linear regression model. The benefit of using the linear regression model is that it is very easy to interpret the change in wins a team would have if the value of one of the variable in the model increase or decreases. Teams can make changes to the players that they play or try to trade or sign players that will help improve the team statistics that will lead to the team winning more games.

Code

```
hitting <- read.csv("FG_Custom_2015_2019.csv")
hitting[] <- lapply(hitting, gsub, pattern="%", replacement = "")

colnames(hitting)<- c("Season","Team","ExitVelocity_H","LaunchAngle_H","Barrel%_H","HardHit%_H","K%_H",

str(hitting)
```

```
pitching = read.csv("FG_Pitching_2015_2019.csv")
pitching[] <- lapply(pitching, gsub, pattern="%", replacement = "")

colnames(pitching) = c("Season","Team","Wins","BABIP_P","LOB%_P","HR/FB_P","ERA_P","FIP_P","xFIP_P","WA

str(pitching)
```

```
pitching_RunsAllowed = read.csv("FG_Pitching_RunsAllowed_2015_2019.csv")
pitching = cbind(pitching,pitching_RunsAllowed[,5])
colnames(pitching) = c("Season","Team","Wins","BABIP_P","LOB%_P","HR/FB_P","ERA_P","FIP_P","xFIP_P","WA
```

```
baseball = merge(x = hitting,y = pitching,by = c("Team", "Season"))
```

```
#Change all columns except team from characters to numeric
i = c(2:length(baseball))
baseball[ , i] <- apply(baseball[ , i], 2,
                    function(x) as.numeric(as.character(x)))
sapply(baseball, class)
```

```
library(psych)
#summary statistics
describe(baseball, fast = TRUE)
```

```
##                            vars   n    mean    sd     min     max  range   se
## Team                          1 150     NaN    NA     Inf    -Inf   -Inf   NA
## Season                        2 150 2017.00  1.42 2015.00 2019.00   4.00 0.12
## ExitVelocity_H                3 150   88.24  0.81   85.50   90.00   4.50 0.07
## LaunchAngle_H                 4 150   11.87  1.60    6.20   15.30   9.10 0.13
## Barrel%_H                     5 150    5.82  1.14    3.10    9.30   6.20 0.09
## HardHit%_H                    6 150   34.45  2.56   27.10   40.00  12.90 0.21
## K%_H                          7 150   21.68  2.09   15.90   26.40  10.50 0.17
## BB%_H                         8 150    8.27  1.00    6.30   10.50   4.20 0.08
## IsolatedPower_H               9 150    0.17  0.02    0.11    0.22   0.12 0.00
## LineDrive%_H                 10 150   20.97  1.00   18.70   24.60   5.90 0.08
## Groundball%_H                11 150   44.03  2.44   38.10   51.90  13.80 0.20
## Flyball%_H                   12 150   35.00  2.39   27.60   41.10  13.50 0.19
## WeightedRunsCreatedPlus_H    13 150   96.57  8.77   76.00  126.00  50.00 0.72
## Runs_H                       14 150  733.80 76.68  573.00  943.00 370.00 6.26
## OPS_H                        15 150    0.74  0.03    0.66    0.85   0.19 0.00
## SLG_H                        16 150    0.42  0.03    0.36    0.50   0.14 0.00
## BABIP_H                      17 150    0.30  0.01    0.28    0.33   0.06 0.00
## HR_H                         18 150  193.23 38.93  100.00  307.00 207.00 3.18
## wOBA_H                       19 150    0.32  0.01    0.29    0.36   0.07 0.00
```

```
## Wins                        20 150   80.97 12.68   47.00  108.00  61.00 1.03
## BABIP_P                     21 150    0.30  0.01    0.26    0.32   0.06 0.00
## LOB%_P                      22 150   72.80  2.21   68.00   79.40  11.40 0.18
## HR/FB_P                     23 150   13.17  1.79    9.40   19.00   9.60 0.15
## ERA_P                       24 150    4.24  0.53    2.94    5.67   2.73 0.04
## FIP_P                       25 150    4.23  0.44    3.23    5.56   2.33 0.04
## xFIP_P                      26 150    4.23  0.39    3.33    5.23   1.90 0.03
## WAR_P                       27 150   14.33  5.48    1.00   30.40  29.40 0.45
## WHIP_P                      28 150    1.32  0.09    1.10    1.51   0.41 0.01
## LineDrive%_P                29 150   20.96  0.95   18.70   23.10   4.40 0.08
## Groundball%_P               30 150   44.05  2.20   38.30   50.40  12.10 0.18
## Flyball%_P                  31 150   34.98  2.21   27.40   40.50  13.10 0.18
## SwingingStrike%_P           32 150   10.47  0.95    8.40   13.00   4.60 0.08
## K%_P                        33 150   21.70  2.36   17.00   28.50  11.50 0.19
## BB%_P                       34 150    8.27  0.86    6.10   10.30   4.20 0.07
## SIERA_P                     35 150    4.15  0.36    3.27    4.89   1.62 0.03
## Soft%_P                     36 150   18.31  1.33   15.00   21.60   6.60 0.11
## Med%_P                      37 150   48.66  3.10   41.90   54.30  12.40 0.25
## Hard%_P                     38 150   33.04  3.90   25.60   42.80  17.20 0.32
## ExitVelo_P                  39 150   88.23  0.70   86.20   89.90   3.70 0.06
## LaunchAngle_P               40 150   11.86  1.46    7.80   15.70   7.90 0.12
## Barrel%_P                   41 150    5.80  0.87    3.80    8.30   4.50 0.07
## HardHit%_P                  42 150   34.41  2.05   28.60   40.50  11.90 0.17
## RA_P                        43 150  733.80 88.43  525.00  981.00 456.00 7.22
```

```r
#Function to print correlations
attach(baseball)
x = 3
for (i in baseball[,3:43]) {
  cat(names(baseball[x]),  " and Wins correlation: " , cor(i,Wins),"\n")
  x = x + 1
}
```

```
## ExitVelocity_H  and Wins correlation:  0.3923636
## LaunchAngle_H  and Wins correlation:  0.1987292
## Barrel%_H  and Wins correlation:  0.3510248
## HardHit%_H  and Wins correlation:  0.4246023
## K%_H  and Wins correlation:  -0.2920551
## BB%_H  and Wins correlation:  0.5432014
## IsolatedPower_H  and Wins correlation:  0.4924674
## LineDrive%_H  and Wins correlation:  -0.05486168
## Groundball%_H  and Wins correlation:  -0.2153312
## Flyball%_H  and Wins correlation:  0.2459969
## WeightedRunsCreatedPlus_H  and Wins correlation:  0.7202837
## Runs_H  and Wins correlation:  0.6507682
## OPS_H  and Wins correlation:  0.6567219
## SLG_H  and Wins correlation:  0.5790466
## BABIP_H  and Wins correlation:  0.06382248
## HR_H  and Wins correlation:  0.445407
## wOBA_H  and Wins correlation:  0.7001235
## Wins  and Wins correlation:  1
## BABIP_P  and Wins correlation:  -0.4843451
## LOB%_P  and Wins correlation:  0.7465262
## HR/FB_P  and Wins correlation:  -0.2264793
```

```
## ERA_P  and Wins correlation:  -0.7808427
## FIP_P  and Wins correlation:  -0.6786824
## xFIP_P  and Wins correlation:  -0.6285034
## WAR_P  and Wins correlation:  0.7532432
## WHIP_P  and Wins correlation:  -0.7689023
## LineDrive%_P  and Wins correlation:  -0.05161117
## Groundball%_P  and Wins correlation:  0.2434329
## Flyball%_P  and Wins correlation:  -0.2171333
## SwingingStrike%_P  and Wins correlation:  0.509572
## K%_P  and Wins correlation:  0.6171353
## BB%_P  and Wins correlation:  -0.4190796
## SIERA_P  and Wins correlation:  -0.6037917
## Soft%_P  and Wins correlation:  0.3702824
## Med%_P  and Wins correlation:  0.03146366
## Hard%_P  and Wins correlation:  -0.1494446
## ExitVelo_P  and Wins correlation:  -0.3848747
## LaunchAngle_P  and Wins correlation:  -0.1652626
## Barrel%_P  and Wins correlation:  -0.3490348
## HardHit%_P  and Wins correlation:  -0.4159133
## RA_P  and Wins correlation:  -0.771469
```

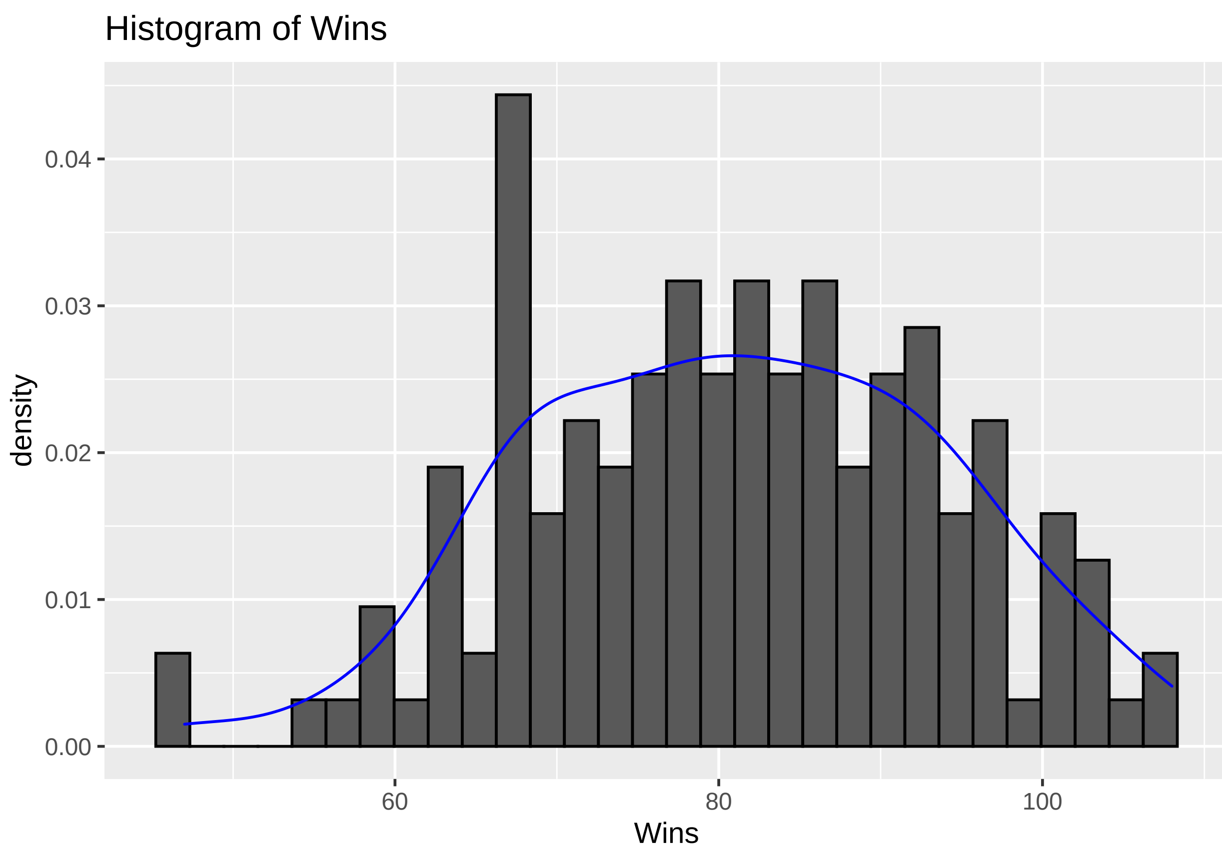
Function to calculation expected wins:

```
PythagoreanWinningPercentage = function(RS,RA)
{
  (RS^1.83/ (RS^1.83 + RA^1.83))
}
```

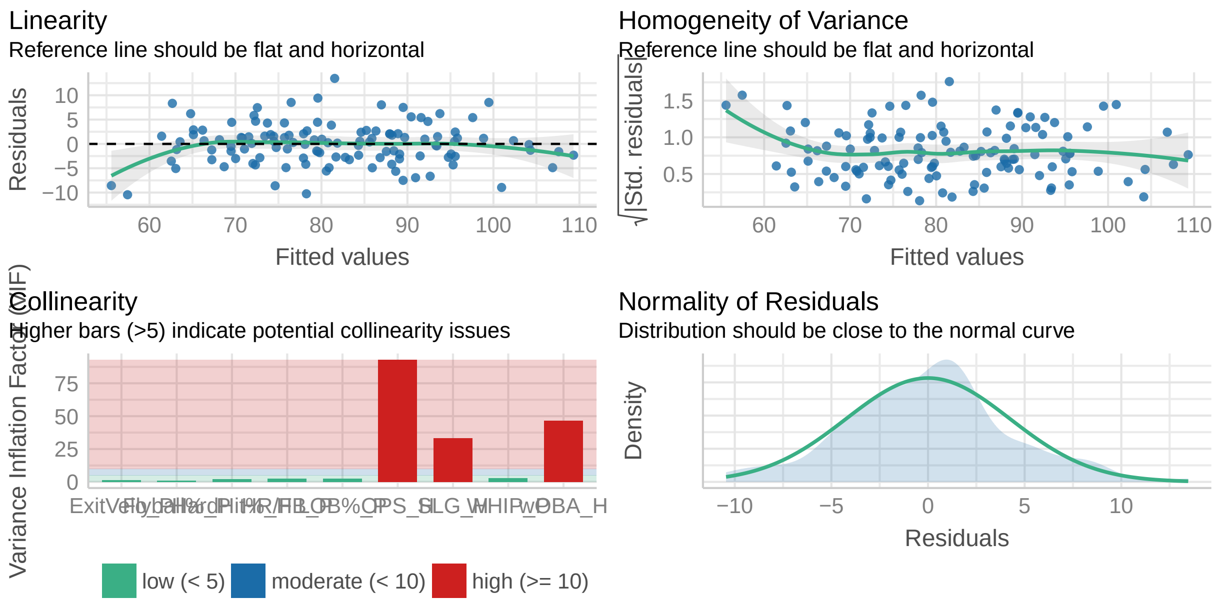
```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
ggplot(data = baseball,
       aes(x = Wins)) +
  geom_histogram(bins = 30, color = "black",aes(y=..density..)) +
  geom_density(color = "blue") +
  labs(title = "Histogram of Wins")
```




```
library(performance)
check_model(model, check = c("linearity","vif","homogeneity","normality"))
```

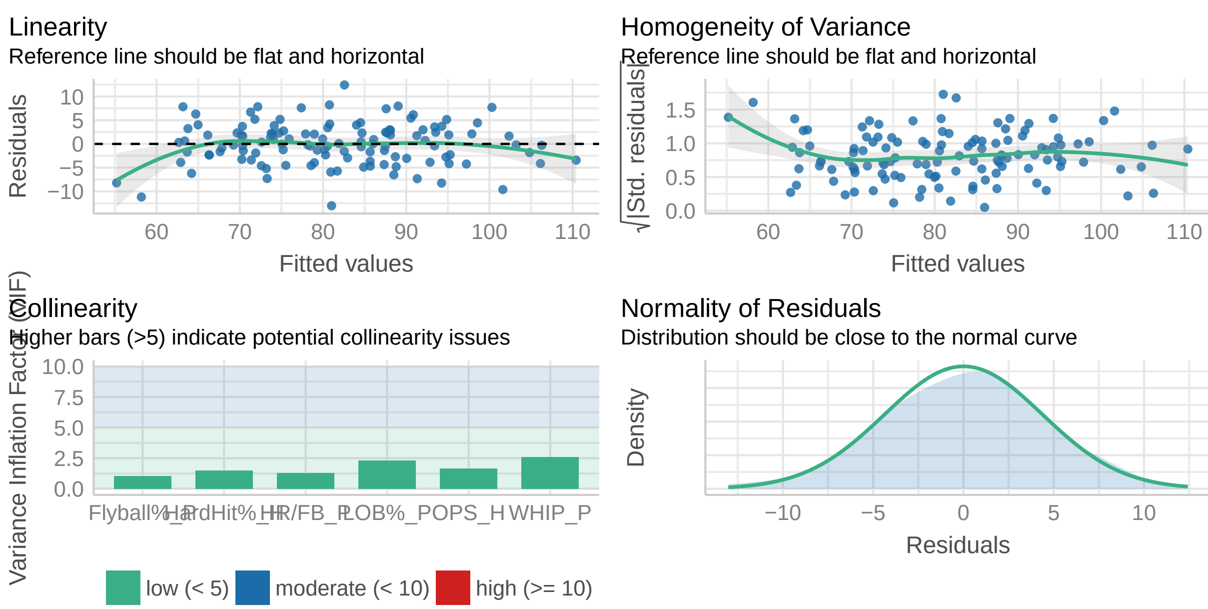


```r
model_performance(model)
```

```
## # Indices of model performance
##
## AIC     |     BIC |    R2 | R2 (adj.) |  RMSE | Sigma
## -----------------------------------------------------
## 708.640 | 739.302 | 0.882 |     0.872 | 4.229 | 4.417
```

Regression Diagnostics

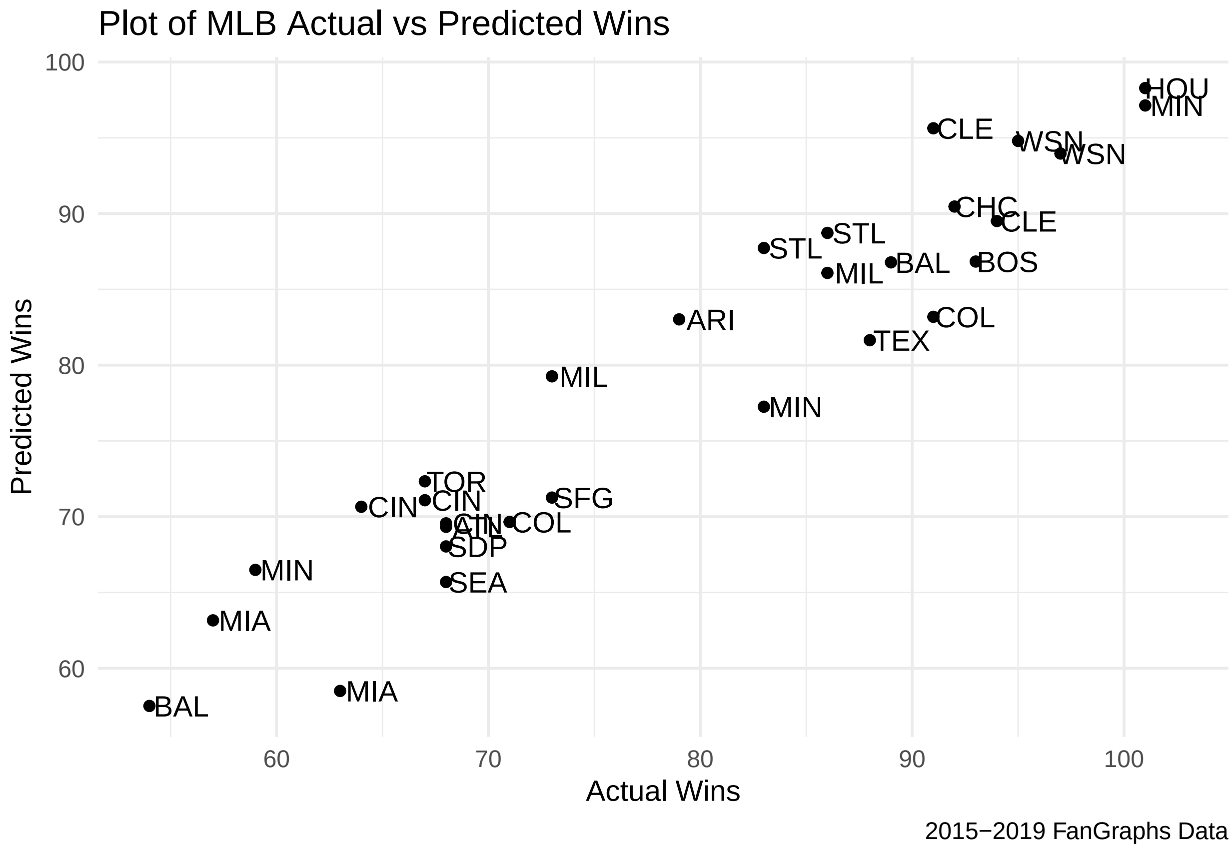
```r
model_final = lm(formula = Wins ~ WHIP_P + OPS_H + `LOB%_P` + `Flyball%_P` +
    `HardHit%_H` + `HR/FB_P` , data = train)

check_model(model_final, check = c("linearity","vif","homogeneity","normality"))
```




```r
model_performance(model_final)
```

```
## # Indices of model performance
##
## AIC     |     BIC |    R2 | R2 (adj.) |  RMSE | Sigma
## -----------------------------------------------------
## 709.889 | 732.189 | 0.874 |     0.868 | 4.359 | 4.492
```

```r
ggplot(data =actual_vs_pred, aes(x = Actual, y = Predicted)) + geom_point() +
  theme_minimal() +
  labs(x = "Actual Wins",
       y = "Predicted Wins",
       title = "Plot of MLB Actual vs Predicted Wins",
       caption = "2015-2019 FanGraphs Data") +
  geom_text(label = baseball$Team[-index], nudge_x = 1.5)
```


Plot of MLB Actual vs Predicted Wins
Predicted Wins
100
90
80
70
60
Actual Wins
60
70
80
90
100
HOU
MIN
CLE
WSN
WSN
CHC
CLE
STL
STL
BAL
BOS
MIL
ARI
COL
TEX
MIL
MIN
TOR
CIN
CIN
SFG
COL
SDP
SEA
MIN
MIA
MIA
BAL
2015−2019 FanGraphs Data