DATA 621 Project 1

Michael Muller

February 25, 2018

DATA EXPLORATION pt.1

The moneyball dataset contains roughly 2200 observations of baseball team statistics from 1871 to 2006.

All the data is represented in the form of integers; the point of this project is to create a multiple linear regression model that best predicts the number of wins from their in-game statistics of that year.

Below is a brief description of all our datapoints.

The dataset contains

7 discrete variables on *Batting*

2 discrete variables on *Base Running*

2 discrete variables on *Fielding*

4 discrete variables on *Pitching*

Below are the only 6 variables that theoretically impact number of wins NEGATIVELY

Variable Desc | [Variable Name]

Strikeouts by batter | *[TEAM_BATTING_SO]*

Caught stealing (bases) | *[TEAM_BASERUN_CS]*

Errors | *[TEAM_FIELDING_E]*

Walks Allowed | *[TEAM_PITCHING_BB]*

Hits Allowed | *[TEAM_PITCHING_H]*

Homeruns Allowed | *[TEAM_PITCHING_HR]*

Thoughts

All of these variables contain metrics on 'bad actions' during baseball. The players want to minimize these numbers with little exception (Advanced pitching tactics/strategies (A coach may want the pitcher to allow a walk, rather than risk a hit or homerun)) in order to win their current game.

The inverse of this is also true; all 'good actions' during baseball are considered theoretically positive impacts on wins.

DATA EXPLORATION pt.2

There are 191 complete cases

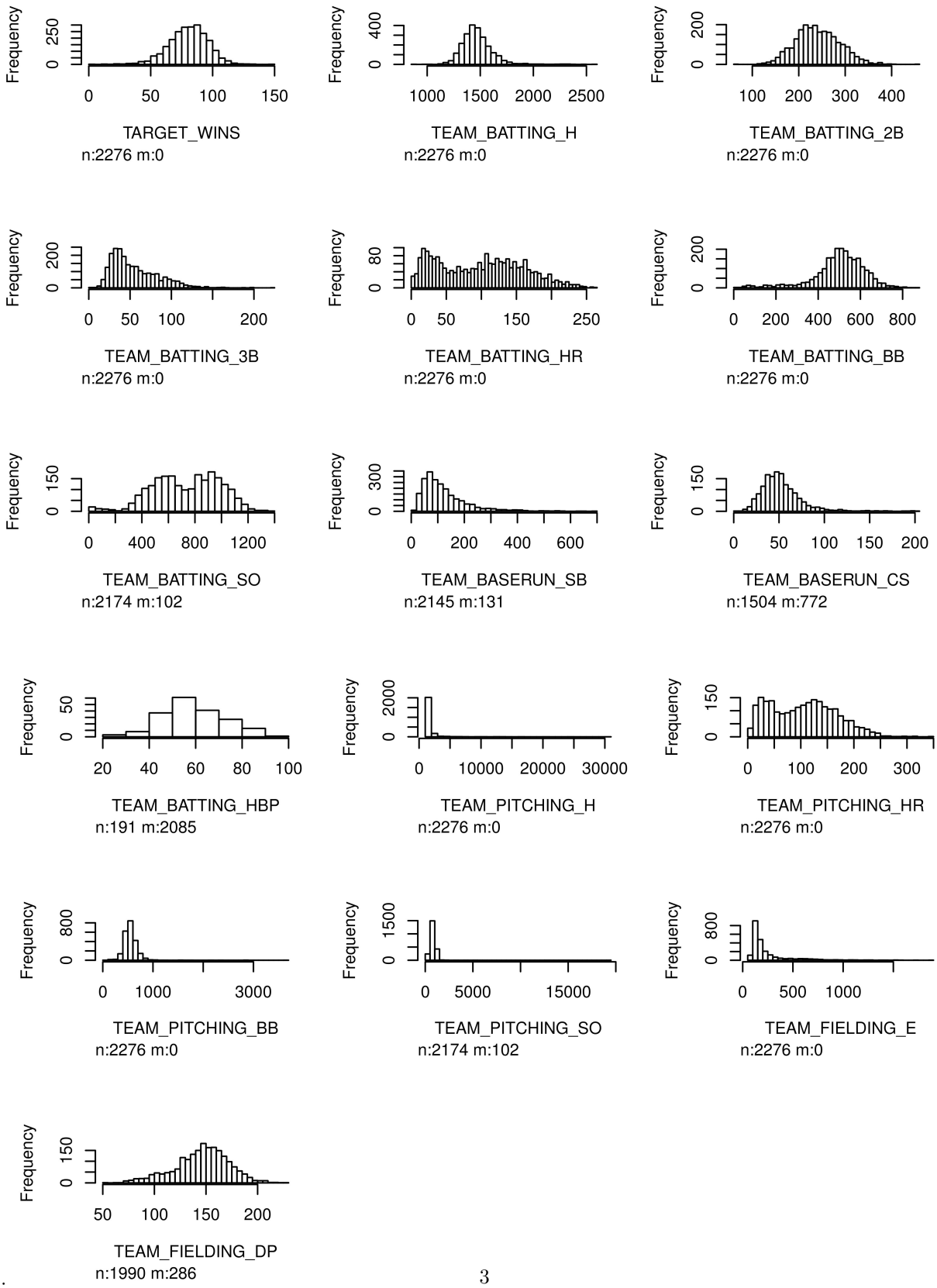
	Variable	# Observed	# Missing
1	INDEX	2276	0
2	TARGET_WINS	2276	0
3	TEAM_BATTING_H	2276	0
4	TEAM_BATTING_2B	2276	0
5	TEAM_BATTING_3B	2276	0
6	TEAM_BATTING_HR	2276	0
7	TEAM_BATTING_BB	2276	0
8	TEAM_BATTING_SO	2174	102
9	TEAM_BASERUN_SB	2145	131
10	TEAM_BASERUN_CS	1504	772
11	TEAM_BATTING_HBP	191	2085
12	TEAM_PITCHING_H	2276	0
13	TEAM_PITCHING_HR	2276	0
14	TEAM_PITCHING_BB	2276	0
15	TEAM_PITCHING_SO	2174	102
16	TEAM_FIELDING_E	2276	0
17	TEAM_FIELDING_DP	1990	286

vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurto
1	2276	1268.46353	736.34904	1270.5	1268.56970	952.5705	1	2535	2534	0.0042149	-1.21675
2	2276	80.79086	15.75215	82.0	81.31229	14.8260	0	146	146	-0.3987232	1.02747
3	2276	1469.26977	144.59120	1454.0	1459.04116	114.1602	891	2554	1663	1.5713335	7.27852
4	2276	241.24692	46.80141	238.0	240.39627	47.4432	69	458	389	0.2151018	0.00616
5	2276	55.25000	27.93856	47.0	52.17563	23.7216	0	223	223	1.1094652	1.50324
6	2276	99.61204	60.54687	102.0	97.38529	78.5778	0	264	264	0.1860421	-0.96311
7	2276	501.55888	122.67086	512.0	512.18331	94.8864	0	878	878	-1.0257599	2.18285
8	2174	735.60534	248.52642	750.0	742.31322	284.6592	0	1399	1399	-0.2978001	-0.32079
9	2145	124.76177	87.79117	101.0	110.81188	60.7866	0	697	697	1.9724140	5.48967
10	1504	52.80386	22.95634	49.0	50.35963	17.7912	0	201	201	1.9762180	7.62038
11	191	59.35602	12.96712	58.0	58.86275	11.8608	29	95	66	0.3185754	-0.11198
12	2276	1779.21046	1406.84293	1518.0	1555.89517	174.9468	1137	30132	28995	10.3295111	141.83969
13	2276	105.69859	61.29875	107.0	103.15697	74.1300	0	343	343	0.2877877	-0.60463
14	2276	553.00791	166.35736	536.5	542.62459	98.5929	0	3645	3645	6.7438995	96.96763
15	2174	817.73045	553.08503	813.5	796.93391	257.2311	0	19278	19278	22.1745535	671.18912
16	2276	246.48067	227.77097	159.0	193.43798	62.2692	65	1898	1833	2.9904656	10.97027
17	1990	146.38794	26.22639	149.0	147.57789	23.7216	52	228	176	-0.3889390	0.18173

A few noticeable figures would be

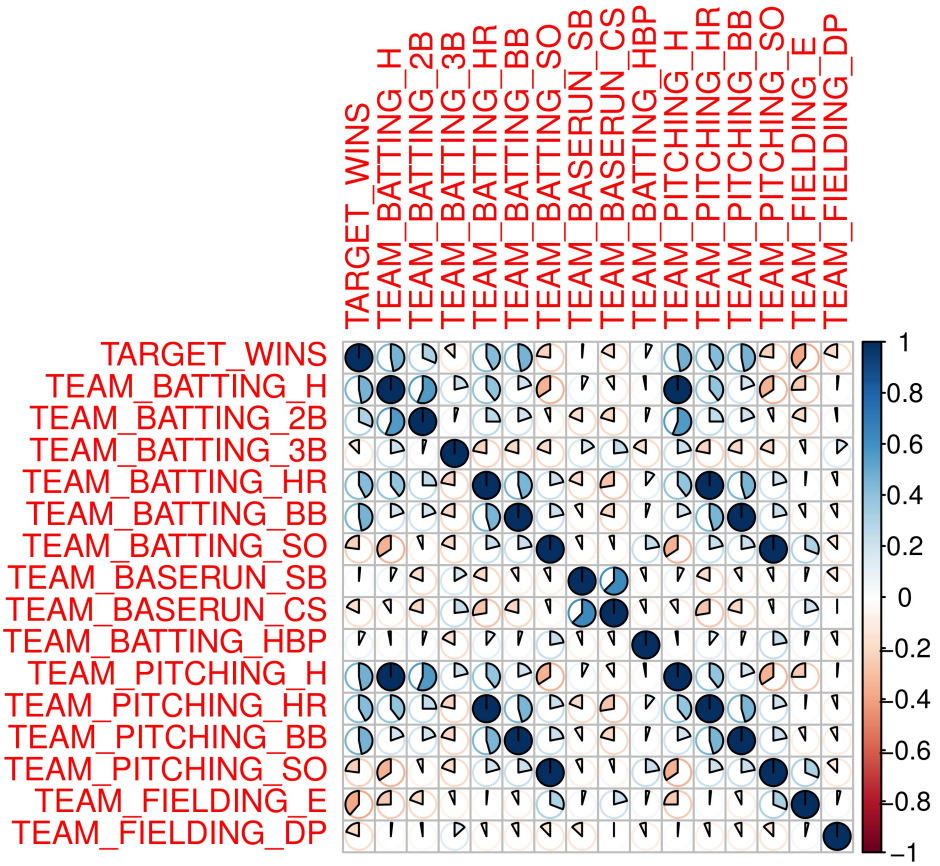
1) The strong positive skews on Pitching|Allowed hits, and Pitching|Strikeouts.
2) Were going to need to scale down a few variables with high ranges
3) Definitely going to remove TEAM_BATTING_HBP for its lack of observations and low range

Figure 1 : Histograms



.

Figure 2 : Correlation matrix plot



In *Figure 1* we can see near normal distributions for most variables. However we can see strong skews from TEAM_FIELDING_DP, TEAM_BATTING_BB, TEAM_BATTING_3B, TEAM_BASERUN_SB. We will need to fix this.
We also see two bimodal distributions in TEAM_PITCHING_HR, TEAM_BATTING_SO, and what appears to be a multi-modal dist. from TEAM_BATTING_HR.
Extreme outliers distorting our views of TEAM_PITCHING_H, and TEAM_PITCHING_SO.

In *Figure 2* we can see strong correlations in the following sets.

TEAM_PITCHING_H :: TEAM_BATTING_H

TEAM_PITCHING_HR :: TEAM_BATTING_HR

TEAM_PITCHING_BB :: TEAM_BATTING_BB

TEAM_PITCHING_SO :: TEAM_PITCHING_SO

Relation? I'm not sure, I don't know too much about baseball, but we don't want our predictor variables to predict themselves. This brings up issues of multicollinearity which can make our parameters indeterminate and increase standard errors across the board.
Note that they are not direct correlations because special circumstance through umpires who can alter rules and give freebees
What we will do though, is remove one corresponding variable, least normally distributed from each pairing.

Also in *Figure 2* we see our most influential observation! The variables most strongly correlated with winning. Behold as they appear to be the same variables, in which we will be removing one half (Minus strikeouts).

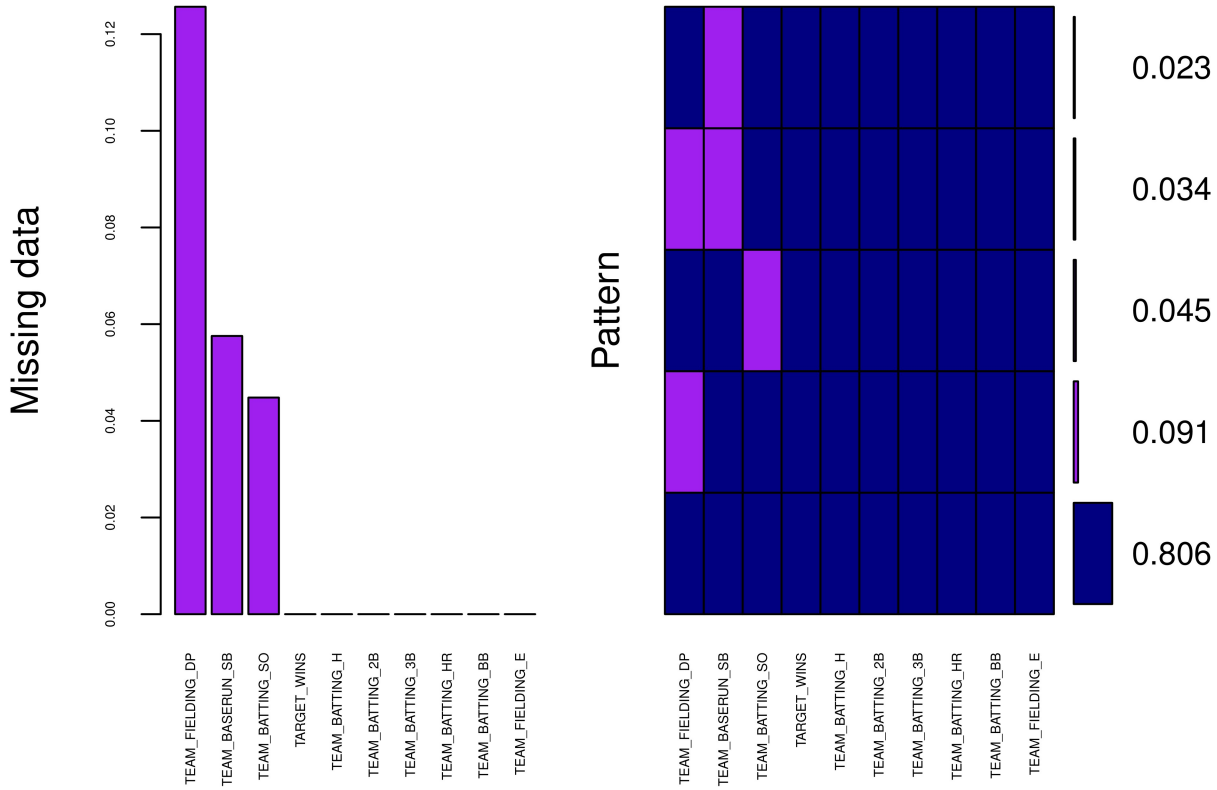
[TEAM_FIELDING_E,TEAM_FIELDING_DP,TEAM_BATTING_2B] have a second tier priority in our upcoming regressions due to their favorable correlations under big 3 [Walks,Homeruns,Basehits].

DATA PREPARATION

We remove the Pitching pairings that are too closely correlated with Batting

We remove TEAM_BATTING_HBP and TEAM_BASERUN_CS for having too many missing data points

Removal of the 7 most problematic variables(80.6% of our observations are complete)




```
##
##  Variables sorted by number of missings:
##          Variable      Count
##  TEAM_FIELDING_DP 0.12565905
##   TEAM_BASERUN_SB 0.05755712
##   TEAM_BATTING_SO 0.04481547
##       TARGET_WINS 0.00000000
##    TEAM_BATTING_H 0.00000000
##   TEAM_BATTING_2B 0.00000000
##   TEAM_BATTING_3B 0.00000000
##   TEAM_BATTING_HR 0.00000000
##   TEAM_BATTING_BB 0.00000000
##   TEAM_FIELDING_E 0.00000000
```

It is clear were going to need to impute some values to make a great predictive model.

We have two objectives left to fix this data.

1) Impute data to fill missing data

2) Cap extreme + infrequent outliers that exist outside 1.5 x IQR

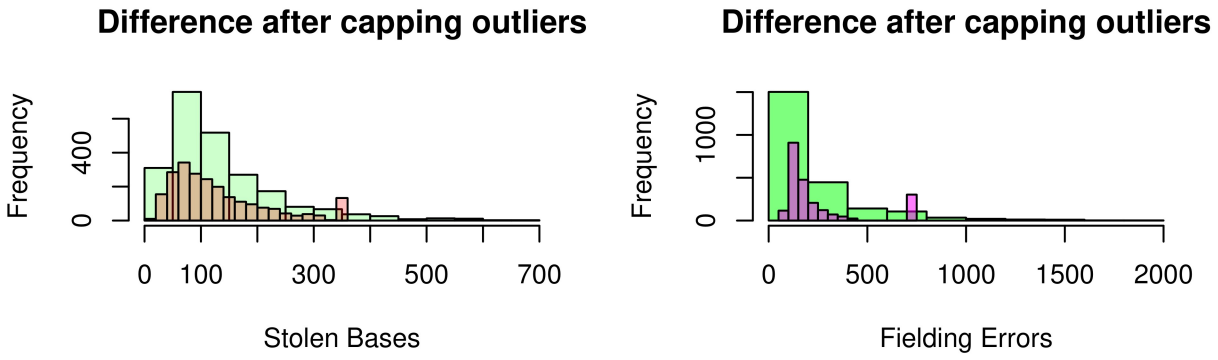
We use 40 iterations of predictive mean matching to complete our dataset, then we can remove some bad leverage points by capping outliers.

```
##
##  iter imp variable
##   1   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   2   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   3   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   4   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   5   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   6   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   7   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   8   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   9   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   10   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   11   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   12   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   13   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   14   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   15   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   16   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   17   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   18   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   19   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   20   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   21   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   22   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   23   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   24   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   25   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   26   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   27   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   28   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   29   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   30   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   31   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   32   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   33   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   34   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   35   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   36   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   37   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   38   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   39   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
##   40   1  TEAM_BATTING_SO  TEAM_BASERUN_SB  TEAM_FIELDING_DP
```

.

Moving on to capping outliers

Figure 1 identified [BASERUN_SB,FIELDING_E] as the variables with extreme & infrequent outliers



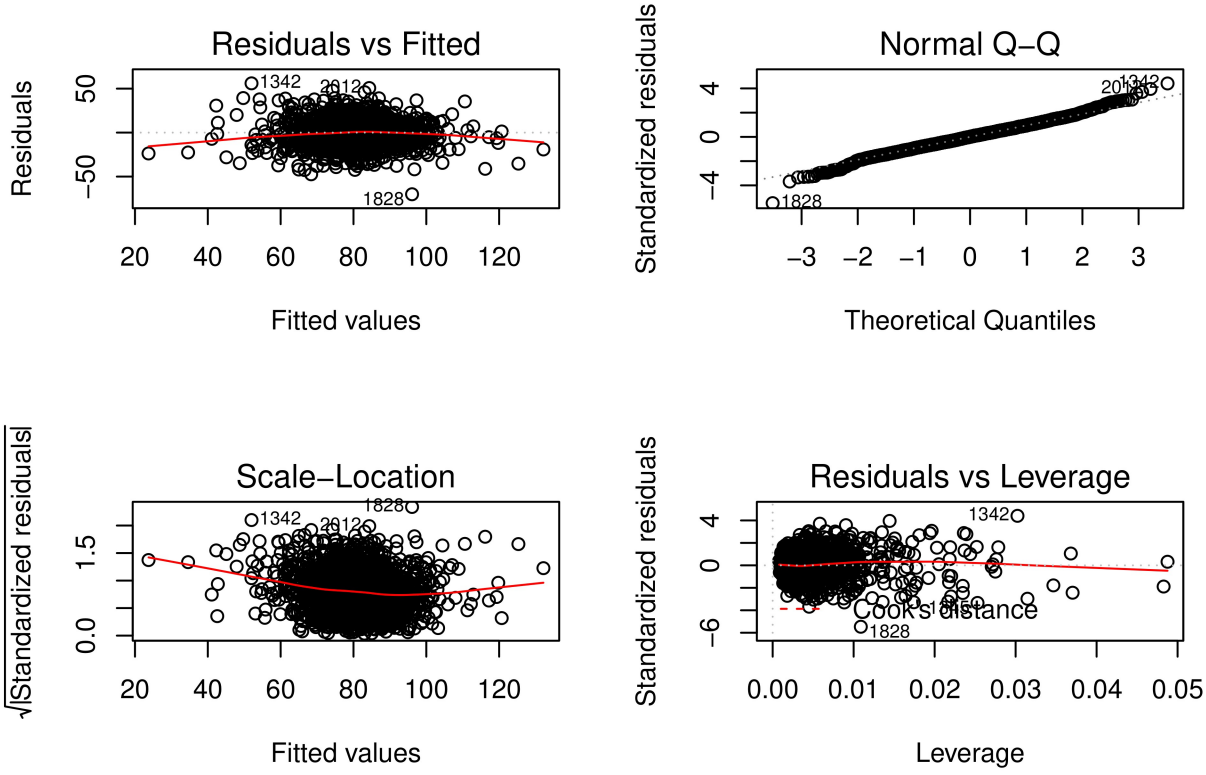
We're done transforming our data for now

BUILD MODELS

Our first model will be the most basic; using every variable available to us

```
##
## Call:
## lm(formula = TARGET_WINS ~ ., data = mmDF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.089  -8.178   0.150   8.172  56.000
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      24.399700   5.004222   4.876 1.16e-06 ***
## TEAM_BATTING_H    0.039463   0.003548  11.122  < 2e-16 ***
## TEAM_BATTING_2B  -0.010867   0.008957  -1.213    0.225
## TEAM_BATTING_3B   0.070886   0.016346   4.337 1.51e-05 ***
## TEAM_BATTING_HR   0.078946   0.009576   8.244 2.78e-16 ***
```

```
## TEAM_BATTING_BB   0.020230   0.002884   7.015 3.03e-12 ***
## TEAM_BATTING_SO  -0.010146   0.002265  -4.479 7.87e-06 ***
## TEAM_BASERUN_SB   0.070087   0.005388  13.007  < 2e-16 ***
## TEAM_FIELDING_E  -0.037446   0.002618 -14.305  < 2e-16 ***
## TEAM_FIELDING_DP -0.095743   0.013444  -7.122 1.43e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.89 on 2266 degrees of freedom
## Multiple R-squared:  0.3326, Adjusted R-squared:   0.33
## F-statistic: 125.5 on 9 and 2266 DF,  p-value: < 2.2e-16
```



Looking at all our variables performing in the summary. We see everything other than Batting_2B has an impossibly low p-value and are significant.
Looking at the plots; we see all our variables are normally distributed with a few outliers (Normal Q-Q), and almost all our points are low influence (Residuals vs. Leverage plot,) which tells me I want to stray away from dropping variables at this point.

Lets try a new model using intuition for experimental purposes; not intended to beat the .33 R^2.

We're going to try using variables with only the lowest standard error to see if it ellucidates our situation, before looking back to model 1.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_SO + TEAM_FIELDING_E +
##     TEAM_BASERUN_SB + TEAM_BATTING_HR + TEAM_BATTING_BB, data = mmDF)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.386  -8.903   0.424   8.506  60.430
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     76.527622   2.034384  37.617  < 2e-16 ***
## TEAM_BATTING_SO -0.032249   0.001831 -17.616  < 2e-16 ***
## TEAM_FIELDING_E -0.029517   0.002702 -10.923  < 2e-16 ***
## TEAM_BASERUN_SB  0.099977   0.005322  18.786  < 2e-16 ***
## TEAM_BATTING_HR  0.136648   0.008172  16.721  < 2e-16 ***
## TEAM_BATTING_BB  0.016526   0.002955   5.593  2.5e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.78 on 2270 degrees of freedom
## Multiple R-squared:  0.2363, Adjusted R-squared:  0.2346
## F-statistic: 140.5 on 5 and 2270 DF,  p-value: < 2.2e-16
```

This model's R^2 statistic is 10% less than model 1; and can only explain 23% of our outcomes variability.

Lets take a look at all the variables in an ANOVA test

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
TEAM_BATTING_H	1	85318.0981	85318.0981	513.1694050	0.0000000
TEAM_BATTING_2B	1	4081.7886	4081.7886	24.5510513	0.0000008
TEAM_BATTING_3B	1	130.0122	130.0122	0.7819946	0.3766256
TEAM_BATTING_HR	1	24701.9517	24701.9517	148.5767512	0.0000000
TEAM_BATTING_BB	1	18751.2227	18751.2227	112.7844386	0.0000000
TEAM_BATTING_SO	1	326.6108	326.6108	1.9644915	0.1611707
TEAM_BASERUN_SB	1	16756.3999	16756.3999	100.7860227	0.0000000
TEAM_FIELDING_E	1	29259.3902	29259.3902	175.9887318	0.0000000
TEAM_FIELDING_DP	1	8432.2058	8432.2058	50.7178448	0.0000000
Residuals	2266	376738.7698	166.2572	NA	NA

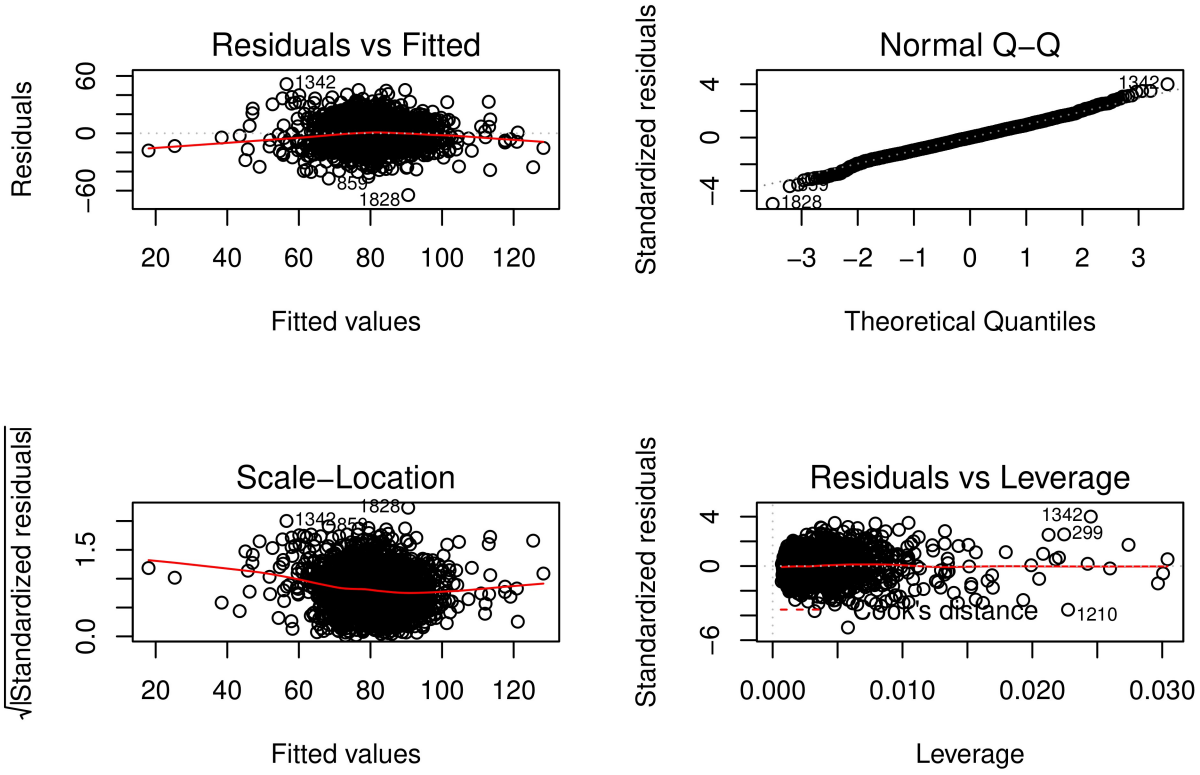
The F-statistic for 3 Base batting and batting strikeouts is so low, I'm going to have to reject both variables as good predictors in my model. Lets try one last model without them.

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_FIELDING_E +
##     TEAM_FIELDING_DP, data = mmDF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.541  -8.574   0.034   8.554  51.459
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.386954   3.503770   1.252  0.21067
## TEAM_BATTING_H    0.054908   0.002646  20.755  < 2e-16 ***
## TEAM_BATTING_2B  -0.027434   0.008591  -3.193  0.00143 **
## TEAM_BATTING_HR   0.034290   0.006538   5.245 1.71e-07 ***
```

```
## TEAM_BATTING_BB   0.024366   0.002832   8.603  < 2e-16 ***
## TEAM_BASERUN_SB   0.065262   0.005101  12.793  < 2e-16 ***
## TEAM_FIELDING_E  -0.033502   0.002570 -13.036  < 2e-16 ***
## TEAM_FIELDING_DP -0.096490   0.013533  -7.130 1.34e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.01 on 2268 degrees of freedom
## Multiple R-squared:  0.3198, Adjusted R-squared:  0.3177
## F-statistic: 152.3 on 7 and 2268 DF,  p-value: < 2.2e-16
```

A high p-value on TEAM_BATTING_2B which has already shown high correlation to TEAM_BATTING_H and TEAM_BATTING_3B gives us more insight on the nature of our variables. For further regression analysis; we may want to transform batting H, 2B, 3B, HR. I believe because all these explanatory variables have to do with capturing bases; it may pose some multi-colinearity issues.

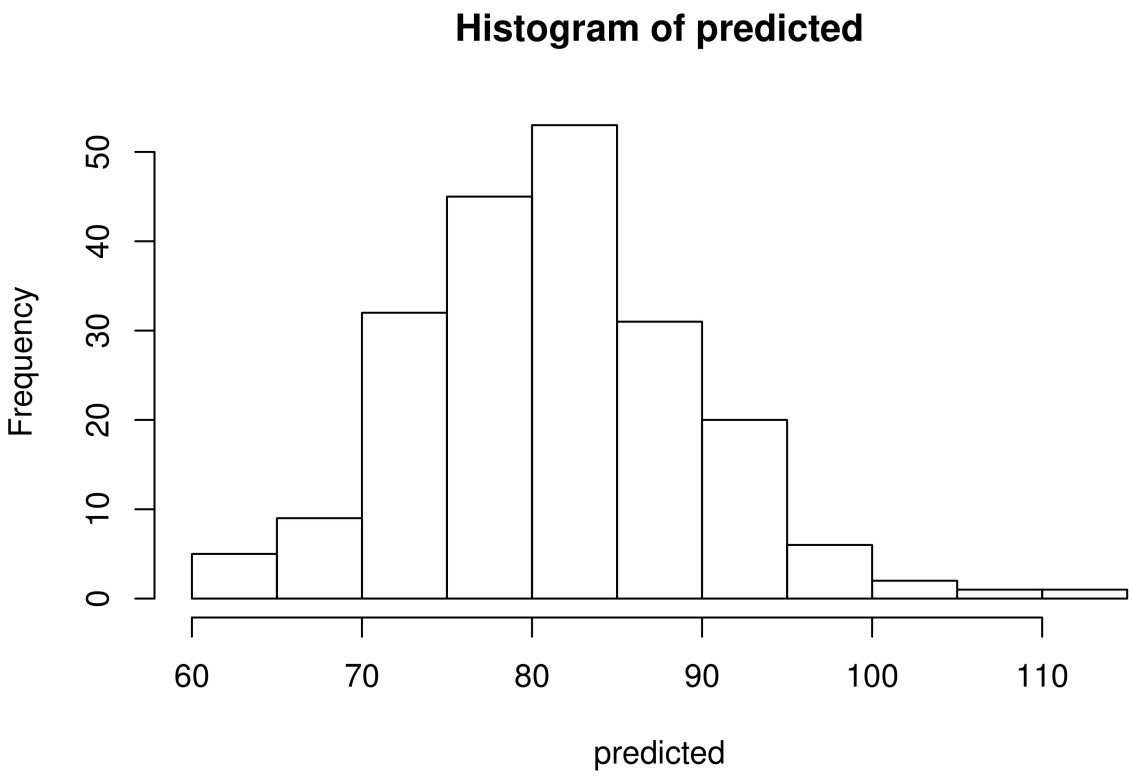
Because We have the highest F-statistic with a near 0 p-value on the third model, I want to check the residual plots.



SELECT MODELS

I've selected model 1 at a R^2 at .33. Using all the explanatory variables (after transformation and tidying) seemed to give us best results. Our leverage plots for model 1 showed nothing outside of Cook's distance *unlike model 3* and for the most part, the theoretical quantiles lined near normally to our standardized residuals. Because the nature of the baseball wins happens to be mean centered it was hard to identify any unusual trends or patterns in the standard residual plot.
Our model summaries showed strong significance all around the board, with ANOVA showing us p-values that just barely made it under our significance levels, so I chose to keep them if they would give better correlations. I was torn between model 3's high F-statistic and relatively same R2 score; but with more than a few bad

leverage points showing and a larger MSE, I selected model 1. Lets see how it does on the test data.



For our predicted wins; we have mean centered just above 80, and a right skew (which is good.) It means fewer teams won more games which is naturally how sports work. Many teams lose too many games and stop playing for the season where as the winners keep playing. Since the evaluation data lacks

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
#https://stackoverflow.com/questions/9341635/check-for-installed-packages-before-running-install-packag
requiredPackages = c('knitr','prettydoc','kableExtra','ggplot2','tidyr','plyr','dplyr','psych','corrplo
for(p in requiredPackages){
  if(!require(p,character.only = TRUE)) install.packages(p)
  library(p,character.only = TRUE)
}
#Load Data
train = read.csv('moneyball-training-data.csv')
test = read.csv('moneyball-evaluation-data.csv')
#Simple way to find missing data metrics (too many variables for the mice plot to map on a PDF)
completeCases = data.frame(
  abbre = names(train),
  dataPoints = as.vector(rapply(train,function(x)sum(!is.na(x)))),
  missingData = as.vector(rapply(train,function(x)dim(train)[1]-(sum(!is.na(x)))))
)
columnNames = c('Variable','# Observed','# Missing')
```

```
colnames(completeCases) = columnNames
knitr::kable(completeCases,row.names = TRUE)
knitr::kable(describe(train),row.names = FALSE)
#Here we remove the index and load Hmisc for a multi-hist plot
mmDF = train[2:dim(train)[2]]
library(Hmisc)
hist.data.frame(mmDF[1:9])
hist.data.frame(mmDF[10:dim(mmDF)[2]])
#Creating a correlation matrix to address multi-colinearity issues
correlationMatrix = cor(mmDF, use='complete.obs')
corrplot(correlationMatrix, method="pie")
#Drop troublesome explanatory variables
drops = c('TEAM_BATTING_HBP','TEAM_BASERUN_CS','TEAM_PITCHING_H','TEAM_PITCHING_HR','TEAM_PITCHING_SO',
mmDF = mmDF[,!(names(mmDF) %in% drops)]
#Using the mice package to identify the missing variables again
missingValuePlot = aggr(mmDF, col=c('navyblue','purple'),
                    numbers=TRUE, sortVars=TRUE,
                    labels=names(mmDF), cex.axis=.4,
                    gap=3, ylab=c("Missing data","Pattern"))
#imputing data with mice library
imputedData = mice(mmDF, m=1, maxit = 40, method = 'pmm', seed = 15)
mmDF=complete(imputedData)
#http://r-statistics.co/Outlier-Treatment-With-R.html
vector.capper = function(x){
qnt <- quantile(x, probs=c(.25, .75), na.rm = T)
caps <- quantile(x, probs=c(.05, .95), na.rm = T)
H <- 1.5 * IQR(x, na.rm = T)
x[x < (qnt[1] - H)] <- caps[1]
x[x > (qnt[2] + H)] <- caps[2]
return(x)
}
#Quick and fancy histograms to show before/after histograms
par(mfrow=c(2,2))
p1 = hist(mmDF$TEAM_BASERUN_SB,plot=FALSE)
p2 = hist(vector.capper(mmDF$TEAM_BASERUN_SB),plot=FALSE)
plot( p1, col=rgb(0,1,0,1/5),main='Difference after capping outliers',xlab='Stolen Bases')
plot( p2, col=rgb(1,0,0,1/4), add=T)
p1 = hist(mmDF$TEAM_FIELDING_E,plot=FALSE)
p2 = hist(vector.capper(mmDF$TEAM_FIELDING_E),plot=FALSE)
plot( p1, col=rgb(0,1,0,5/10),main='Difference after capping outliers',xlab='Fielding Errors')
plot( p2, col=rgb(1,0,1,1/2), add=T)
mmDF$TEAM_FIELDING_E = vector.capper(mmDF$TEAM_FIELDING_E)
mmDF$TEAM_BASERUN_SB = vector.capper(mmDF$TEAM_BASERUN_SB)
#Establish model 1 and plot
fit = lm(TARGET_WINS ~., data=mmDF)
summary(fit)
par(mfrow=c(2,2))
plot(fit)

#establish model 2 and summary
fit2 = lm(TARGET_WINS~TEAM_BATTING_SO +TEAM_FIELDING_E +TEAM_BASERUN_SB +TEAM_BATTING_HR +TEAM_BATTING_
summary(fit2)
#Anova all explanatory variables
```

```
anova(fit)
#Create modified model
fit3 = lm(TARGET_WINS~ TEAM_BATTING_H+TEAM_BATTING_2B+TEAM_BATTING_HR+TEAM_BATTING_BB+TEAM_BASERUN_SB+T
summary(fit3)
#plot model 3
par(mfrow=c(2,2))
plot(fit3)
#predict values and write to csv
predicted = predict(fit,test)
hist(predicted)
write.csv(predicted,'MMullerPredictions.csv')
```