

3-30-2018


# Sabermetrics - Statistical Modeling of Run Creation and Prevention in Baseball

Parker Chernoff

[pcher020@fiu.edu](mailto:pcher020@fiu.edu), [pcher020@fiu.edu](mailto:pcher020@fiu.edu)

**DOI:** 10.25148/etd.FIDC006540

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

 Part of the [Applied Statistics Commons](#), [Numerical Analysis and Computation Commons](#), [Statistical Methodology Commons](#), and the [Statistical Models Commons](#)

---

## Recommended Citation

Chernoff, Parker, "Sabermetrics - Statistical Modeling of Run Creation and Prevention in Baseball" (2018). *FIU Electronic Theses and Dissertations*. 3663.

<https://digitalcommons.fiu.edu/etd/3663>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact [dcc@fiu.edu](mailto:dcc@fiu.edu).

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

SABERMETRICS – STATISTICAL MODELING OF RUN CREATION AND  
PREVENTION IN BASEBALL

A thesis submitted in partial fulfillment of

the requirements for the degree of

MASTER OF SCIENCE

in

STATISTICS

by

Parker Chernoff

2018

To: Dean Michael R. Heithaus  
College of Arts, Sciences and Education

This thesis, written by Parker Chernoff, and entitled Sabermetrics – Statistical Modeling of Run Creation and Prevention in Baseball, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

---

Zhenmin Chen

---

Jie Mie

---

Sneh Gulati, Major Professor

Date of Defense: March 30, 2018

The thesis of Parker Chernoff is approved.

---

Dean Michael R. Heithaus  
College of Arts, Sciences and Education

---

Andrés G. Gil  
Vice President for Research and Economic Development  
and Dean of the University Graduate School

Florida International University, 2018

## ACKNOWLEDGMENTS

I would like to express my appreciation and gratitude for Dr. Sneh Gulati for the guidance, encouragement, and recommendations that you have provided throughout the writing of my master's thesis. You were helpful and patient even when I disturbed you during your time away from campus.

I would also like to thank the other members of my defense committee, Dr. Zhenmin Chen and Dr. Jie Mi for your constructive advice as well. Completing my Master of Science in Statistics at FIU would not have been possible without a small but mighty group of classmates (you know who you are). We were always supportive of each other and I thank you for your camaraderie.

Finally, a special thank you to my family (human, canine, feline, and terrapin, but mostly human), specifically my mom, dad, and grandma for your love, support, patience, and oft inappropriate senses of humor, the latter of which may or may not be on display here. For all the times you've told me I could accomplish a task when I didn't think I could and for all the times you've steadied me when I was under stress, I thank you from the bottom of my heart. I don't tell you nearly enough how grateful I am for everything that you do.

ABSTRACT OF THE THESIS  
SABERMETRICS – STATISTICAL MODELING OF RUN CREATION AND  
PREVENTION IN BASEBALL

by

Parker Chernoff

Florida International University, 2018

Miami, Florida

Professor Sneh Gulati, Major Professor

The focus of this thesis was to investigate which baseball metrics are most conducive to run creation and prevention. Stepwise regression and Liu estimation were used to formulate two models for the dependent variables and also used for cross validation. Finally, the predicted values were fed into the Pythagorean Expectation formula to predict a team's most important goal: winning.

Each model fit strongly and collinearity amongst offensive predictors was considered using variance inflation factors. Hits, walks, and home runs allowed, infield putouts, errors, defense-independent earned run average ratio, defensive efficiency ratio, saves, runners left on base, shutouts, and walks per nine innings were significant defensive predictors. Doubles, home runs, walks, batting average, and runners left on base were significant offensive regressors. Both models produced error rates below 3% for run prediction and together they did an excellent job of estimating a team's per-season win ratio.

## TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION .....	1
II. LITERATURE REVIEW .....	3
III. STATISTICAL METHODOLOGY .....	10
IV. DATA ANALYSIS.....	13
V. CONCLUSION.....	25
REFERENCES .....	27
APPENDIX.....	29

## LIST OF FIGURES

FIGURE	PAGE
1A: Scatterplot of Home Runs vs. Runs Scored .....	13
1B: Scatterplot of Hits Allowed vs. Runs Allowed .....	14
2: Summaries and VIFs for Stepwise Regression Models.....	15
3A: Correlation Matrix and Scatterplot Matrix for Defensive Model .....	17
3B: Residual Plot for Defensive Model .....	18
4: Summary of Liu Offensive Model with 12 Variables .....	19
5: Correlation Matrix for 6 Remaining Offensive Variables after Liu Regression .....	20
6: Summary of Final Liu Offensive Model.....	21
7: VIFs for Final Liu Offensive Model.....	22
8A: Residual Plot for Final Liu Offensive Model .....	22
8B: Correlation Matrix and Scatterplot Matrix for Final Liu Offensive Model.....	23

## GLOSSARY OF BASEBALL TERMS

Offensive:

- At-bats (AB): number of plate appearances resulting in either a hit or an out
- Batting Average (BA): number of hits divided by at-bats
- Batting Age (BatAge): average age of all batters used by a team in a season
- Walk (BB): base on balls; when a batter reaches first base by receiving four balls from pitcher
- Batting Park Factor (BPF):  $\frac{\text{home Runs Scored}}{\text{road Runs Scored}}$
- Caught Stealing (CS): when a runner tries to steal but is tagged out
- Doubles (Doub): subset of hits where the runner reaches second base
- Grounded Into Double Play (GDP): number of times one swing of the bat resulted in two outs
- Hits (H): reaching at least first base after hitting the ball without an error being committed
- Hit by Pitch (HBP): when a pitcher hits a batter with the ball, resulting in the batter automatically being sent to first base
- Homeruns (HR): subset of hits where runner rounds bases and reaches home plate
- Intentional Walks (IBB): times a batter was walked on purpose
- Runners Left on Base (LOB): number of runners remaining on base when an inning ends
- Number of Batters (NumBat): number of batters a team utilizes in a season
- On-Base Percentage (OBP):  $\frac{H+BB+HBP}{AB+H+BB+HBP}$
- On-Base Plus Slugging Percentage (OPS): OBP+SLG



-OPS Plus (OPSplus or OPS+):  $\frac{OPS}{Leaguewide\ OPS\ (adjusted\ for\ park\ factors)} * 100$

-Sacrifice Bunts (SacBunt): intentional bunt out used to advance another runner to the next base

-Stolen Base (SB): when a runner already on base runs to the next base during a pitch

-Sacrifice Fly (SF): intentional fly out used to advance another runner to the next base

-Slugging Percentage (SLG):  $\frac{1B+2*2B+3*3B+4*HR}{AB}$

-Strikeout (SO): out where batter receives three strikes from pitcher

-Total Bases (TB):  $1B + 2 * 2B + 3 * 3B + 4 * HR$

-Triples (Trip): subset of hits where the runner reaches third base

Defensive:

-Outfield Assists (A): number of times an outfielder throws a ball to the infield to record an out

-Walks per 9 Innings (BB9): number of walks allowed per 9 innings of play

-Balks (BK): illegal pitching motion resulting in a one base advancement by all runners and the batter

-Blown Saves (BLSV): times when a pitcher enters a game with a lead of one to three runs and gives up that lead

-Complete Games (CG): number of games during season where one pitcher started and finished game

-Fielding Chances (Ch):  $A + IPouts + E$

-Caught Stealing Percentage (CSpct):  $\frac{CS}{Stealing\ Attempts} * 100$

-Defensive Efficiency Ratio (DefEff):  $1 - \frac{HA-HRA}{AB-SOA-HRA+SB+SF}$

-Defense-Independent Earned Run Average Ratio (DIPpct): a pitcher's projected ERA when accounting for the effects of surrounding fielding and luck

-Double Plays Turned (DP): times two outs are recorded in one play by the defense

-Errors (E): times a fielder misplay a ball so as to allow an at-bat to continue or a base runner to advance

-Earned Run Average (ERA): number of runs 1 pitcher allows per 9 innings of play

-Component Earned Run Average Ratio (ERCpct):  $9 * \frac{(HA+BBA+HBP)*TB}{(Batters Faced)*(Innings Pitched)}$

.56

-Fielding Independent Pitching (FIP):  $\frac{13*HRA+3*(BBA+HBP)-2*SOA}{Innings Pitched} + FIP Constant$ , where

$FIP Constant = \log(ERA) - \frac{13 \log(HRA)+3*(\log(BBA)+\log(HBP))-2\log(SOA)}{\log(Innings Pitched)}$

-Hits per 9 Innings (H9): number of hits allowed per 9 innings of play

-Infield Put Outs (IPouts): outs recorded by first, second, and third basemen as well as pitchers and catchers

-Strikeouts per 9 Innings (K9): number of strikeouts per 9 innings pitched

-Number of Fielders (NumFld): number of fielders utilized by a team in a season

-Number of Pitchers (NumP): number of pitchers utilized by a team in a season

-Pitcher Age (Page): average age of all pitchers used by a team in a season

-Pitching Park Factor (PPF):  $\frac{home Runs Allowed}{road Runs Allowed}$

-Run Support Average per Start (RS): number of runs scored in games a particular pitcher starts

-Shutouts (SHO): games in which a team allows zero runs

-Strikeouts Versus Walks (SOvBB):  $\frac{SOA}{BBA}$

-Saves (SV): times when a pitcher enters a game with a lead of one to three runs and finishes the game without giving up that lead

-Walks plus Hits per Inning Pitched (WHIP):  $\frac{BBA+HA}{Innings\ Pitched}$

-Wild Pitches (WP): a pitch that is not hit and is uncatchable by the catcher

-Hits Allowed (HA), Homeruns Allowed (HRA), Walks Allowed (BBA), Strikeouts Allowed (SOA): see offensive counterparts for definitions

## I. INTRODUCTION

Sabermetrics has existed in the game of baseball for as long as the sport itself. Defined as “the search for objective knowledge about baseball” by baseball historian and statistician Bill James in 1980, sabermetrics has gained much traction in recent years as a result of the “Moneyball” approach taken by the Oakland Athletics (A’s) in the early 2000s (SABR). Lacking the payroll to compete with big-market teams such as the New York Yankees, A’s general manager Billy Beane turned to analytics in an attempt to find players who were undervalued by other clubs. Upon doing so, he could then sign them for Oakland at a fraction of the salaries paid to Major League Baseball superstars. When the A’s won their division only two years later, the rest of the league – and fans around the world – began to take notice.

Statistical tracking existed decades before Moneyball became popular; metrics including Earned Run Average (ERA, runs a pitcher allows every nine innings except in the case where a fielding error is committed) and Home Runs (HR) were present as early as the 1800s (Birnbaum). The usage of such statistics, however, has evolved greatly in the years since. One of the first steps was taken in the 1970s by Bill James using data from the Society of American Baseball Research (SABR), from which the term “sabermetrics” was derived. His work involved taking “conventional” baseball statistics, those that one might find in a game’s box score, and combining them into “sabermetric” statistics. These were believed to “more accurately gauge a player’s value of relative worth” (Beneventano et al., 2012).

Expanding on his own work, James devised several advancements in sabermetrics. In the late 1970s, he established a formula for Runs Created that would predict the number of runs a player contributed to his team (Albert). The formula is detailed below:

$$\text{Runs Created} = \frac{(H+BB)(\text{Total Bases})}{AB+BB} \quad (1.1)$$

where  $H$ =hits,  $AB$ =at-bats, and  $BB$ =walks="base on balls," when a batter reaches first base as a result of a pitcher throwing four balls.

Eight years later, he developed a method known as "Pythagorean Expectation", an uncomplicated but valuable formula that could predict how many games a team would win based on its runs scored and runs allowed (Moy, 2006):

$$\text{Win Ratio} = \frac{(\text{Runs Scored})^2}{(\text{Runs Scored})^2+(\text{Runs Allowed})^2} \quad (1.2)$$

The Pythagorean Expectation formula is still in use by the MLB today, with an adjustment of the exponent from 2 to 1.83.

In the years since Bill James' breakthroughs, an abundance of research has been conducted on accurate evaluation and prediction of player performance. Much less research, however, has been performed with the goal of modeling team performance. Although individual players are important, baseball is a team sport. Notably, Bill James' Pythagorean Expectation uses team runs scored and allowed in order to predict wins. His data involved past performance, but it would be extremely useful to be able to predict both a team's future scored and allowed runs to use his formula to its fullest potential. This thesis aimed to do just that.

## II. LITERATURE REVIEW

Numerous studies have been conducted with respect to baseball statistics. Those concerning run production and prevention are the studies that were the focus of this thesis. The first of these was done in 1963, when George Lindsey assigned run values to each of the four basic hit types (single, double, triple, and homerun) for a player's at-bat (Albert). He proposed the following formula:

$$Runs = (.41) * 1B + (.82) * 2B + (1.06) * 3B + (1.42) * H \quad (2.1)$$

where 1B=singles, 2B=doubles, 3B=triples, and HR=homeruns.

Though a bit crude, this was the first dedicated attempt at predicting runs from conventional statistics using linear weights and was thus considered highly innovative in the field of sabermetrics.

A study at Bucknell University took data from 1996-2000 and plotted runs per game against various metrics including on-base percentage (OBP, times on base divided by plate appearances), slugging percentage (SLG, total number of bases divided by at-bats), on-base plus slugging percentage (OPS, defined as SLG+OBP), and batting average (BA, number of hits divided by at-bats). Using simple linear regression, best fit lines were drawn for the plots. The  $R^2$  values were then used to determine if lines fit the scatterplots well. A value closer to 1 indicated a model explained most of the variability in the response model, whereas a value closer 0 suggested that the model was a poor fit. It was concluded that OPS had the highest correlation with runs per game ( $R^2=.900$ ) out of the eight metrics tested, and thus OPS was the best predictor of that statistic (Vollmayr-Lee, 2001).

A few other points were of note during the study. First, the data for OPS were far more linearly distributed than the data for BA. The realization was eye-opening for baseball analysts who had largely considered BA to be a player's defining statistic, beyond even homeruns or hits. By combining a few metrics into one, a person could get a better sense of a player's performance. Development and use of combined values led directly to the second point: using multiple statistics often yields more accurate results than using just one. The example used by Vollmayr-Lee was the comparison of the pre-2001 versions of Tony Gwynn and Mark McGuire (before their declines and the latter's steroid accusations). Earlier baseball scouts would have placed a higher value on Gwynn than McGuire because of his higher batting average. In hindsight, most now consider McGuire to have been the better player. Indeed, McGuire held the advantage in multiple categories including homeruns, OPS, SLG, and OBP. Combining those statistics into a model painted a more accurate picture than a model consisting of any one statistic alone. This conclusion is backed by the fact that  $R^2$  improved in models with more than one statistic. Finally, he raised the issue that OPS can be a "ballpark dependent stat," meaning that some teams' home parks give them an edge over their opponents (Vollmayr-Lee,2001).

Some of the studies in the literature used regression analysis, and since that was the primary tool used by this thesis, some background on the process of regression is presented here. Simple regression uses a single predictor variable, whereas multiple regression uses more than one as the name implies. Linear regression is applied when the relationship between the response variable and the regressors follows a straight line.

Work at the University of Minnesota-Duluth with regression modeling was done to try and directly predict winning percentage using a combination of eighteen offensive and defensive independent variables. Runs scored were not predicted, as it was one of the variables used in the prediction. Team data from the 1997-2006 seasons were used for model training. Selection was done in three different ways: forward, backward, and stepwise (University of Minnesota-Duluth, 2007). Forward selection involves adding variables one at a time in order of significance until none of the remaining variables reach a pre-set significance level. Forward selection chose runs scored, runs allowed, and saves for inclusion. Backward selection fits a model with all of the possible variables and removing one at a time in reverse order of significance until no variables left in the model fall below a pre-set level of significance. The backward selection strongly suggested including runs scored and saves while moderately suggesting the inclusion of runs allowed. Lastly, stepwise selection is a hybrid of the other two types by alternating between dropping and adding variables that are below and above the pre-set significance level, respectively. Once again, R, RA, and SV were recommended for model inclusion.

With those results in mind, a multiple regression model including runs scored, runs allowed, and saves was fitted and thoroughly analyzed. An  $R^2$  value of .9321 was observed, suggesting a strong fit. Beyond that, the study also examined three other models for winning percentage that had already been established, the most notable being Bill James' Pythagorean Expectation. James later updated his formula from Equation 2 by changing the exponents of the terms from 2 to 1.83, so a man named Steven Miller attempted to derive this formula (Birnbaum). Miller postulated that runs scored and runs



allowed follow Weibull distributions and used Chi-square tests to demonstrate independence. The density for a 3-parameter Weibull distribution is as follows:

$$f(x) = \begin{cases} \frac{\gamma}{\alpha} \left(\frac{x-\beta}{\alpha}\right)^{\gamma-1} e^{-\left(\frac{x-\beta}{\alpha}\right)^\gamma}, & x > \beta \\ 0 & , otherwise \end{cases} \quad (2.2)$$

where  $\gamma$  is the shape parameter (i.e. the exponent in the Pythagorean Expectation formula),  $\beta$  is the location parameter, and  $\alpha$  is the scale parameter (University of Minnesota-Duluth, 2007).

Using Least Squares and Maximum Likelihood, his calculations yielded exponent values of  $\gamma=1.79$  and  $\gamma=1.74$ , respectively, which are very close to James' newest value and thus gave him ammunition to silence the doubters who attacked his work.

A pair of linear regression-related analyses were performed first at the University of California-Berkeley and then at Pennsylvania State University. The Berkeley study used multiple linear regression (MLR) with two regressors for the offensive model, on-base percentage (OBP) and slugging percentage (SLG), and two for the defensive model: WHIP (walks + hits per inning pitched) and DER (defensive efficiency ratio, which measures fielding of balls put in play). The paper noted that earned run average (ERA) was not included as a regressor because even though most would list ERA among the most important defensive metrics, adding it to the model would result in "runs" of some sort being on both sides of the equation, leading to unnecessary correlation. Both offensive regressors had positive correlation with runs scored; the same was true for WHIP and runs allowed. Runs allowed and Defensive Efficiency Ratio (DER) were negatively correlated, which was expected since poor fielding would intuitively lead to a team giving up more runs on average (Moy, 2006).

Going beyond the regression equation, author Dennis Moy (2006) evaluated a trend he noticed in which runs scored increased overall between 1986 and 2005. He wondered whether hitting ability had improved or defensive ability had declined. Examining the scatterplots of each of his four model variables versus Year and then constructing regression planes, Moy could not determine conclusively which hypothesis was correct. It was possible, he said, that the widespread abuse of steroids during this era of baseball or the MLB's potential introduction of "juiced" balls was resulting in higher offensive firepower.

The researchers at Pennsylvania State University decided to use a stepwise regression approach for both the offensive and defensive models, with six conventional and six sabermetric statistics used as regressors in each case. They hoped to determine which type of statistic would make better regressors for runs scored/allowed. Their data ranged from 2002-2010, and they too used linear regression. It is unknown if other regression methods were attempted. The final model for runs scored included wOBA (weighted on-base average, whose formula is altered slightly every year as hitting rises and falls), percentage of plate appearances where a strikeout occurred, SLG, and OBP. The first two are sabermetric statistics, while the latter two are conventional. The wOBA had an  $R^2$  value of .896 on its own, while the full model had an  $R^2$  of .953. Most of the variation in runs scored then was explained by wOBA, with relatively minor contributions coming from the other variables (Beneventano et al., 2012).

A similar result was noted in the runs prevented model, with three conventional statistics (HR allowed per nine innings, fielding percentage=error-free defensive plays divided by total number of opportunities, and number of double plays) and two

sabermetric statistics (LOB%: left on base percentage=number of players left on base when an inning ends, and WHIP). The WHIP value had an  $R^2$  of .940 compared to the full model value of .988. The main difference between the two models was that the coefficients for each regressor in the offensive model were much larger than their defensive counterparts. The authors explain that the runs allowed model is actually predicting earned run average, which is on a per-9-innings basis. Runs scored operates over the scale of a full season. The disparity in tabulation methods can be corrected by multiplying the full regression equation for defense by 162 (the number of games in a season).

Reflecting on the results, the researchers discussed observations specific to the fielding metrics in the defensive model. They displayed surprise that fielding percentage was chosen by the stepwise regression over UZR (Ultimate Zone Rating), a complex sabermetric fielding statistic that in the 2000s was considered a rising star in analytic circles. Potential justifications for the inclusion were the relatively recent introduction of the metric as well as UZR's dependence on a team's ballpark. Whereas fielding percentage is a simple portrayal of how well players handle a ball that comes their way, UZR's purpose is to "estimate each fielder's defensive contribution in theoretical runs above or below an average fielder at his position in that player's league and year," which is not something that can be uniformly evaluated across different parks with unequal fielding properties.

Another method of regression is Liu estimation, developed by Kejian Liu in 1993. Designed to deal with the problem of collinearity, Liu took the ridge estimator and Stein estimator, two other known methods of minimizing collinearity, and merged them to

create his own new regression equation. The resulting parameter estimates are simpler to approximate. Furthermore, the estimates for the regression parameters have lower Mean Square Error (MSE) than other prediction types. Liu demonstrates this property for the simple regression model in his paper (Liu, 1993).

Numerous variations of the Liu estimator have been introduced since the original paper was written in 1993. One of these is the restricted Liu (RL) estimator, which has a better dispersion than its predecessor (Kaciranlar et al., 1999). Another is the almost unbiased Liu estimator (AULE) that improves upon the Liu estimator's MSE (Alheety and Kibria, 2009). Kejian Liu himself developed the two-parameter Liu estimator, equal in performance to the regular Liu estimator but designed to "address the ill-conditioning problem" where the matrix of regression parameters becomes unstable due to collinearity (Liu, 2003). The work done in the present thesis used the original Liu estimator because it is the most commonly used, and the most applicable to the data at hand.

### III. STATISTICAL METHODOLOGY

Relative to other studies done in this area of sabermetrics, my thesis aims to present the most modern analysis to date. Towards that end, current data as well as some of the newest metrics were utilized. The implementation of tools such as Statcast have changed the way baseball data are being studied as well as what types of data can be collected, with some being linked to the generation of outcomes rather than the outcomes themselves (Arthur). Several of these are included in the analytic work done in my thesis, which will hopefully shed some new light on how teams can best build their organizations to create the most runs and give up the least runs possible. It is simpler and more effective to impose a team model on players than it is to use a player model to predict team performance, which provides more support for taking a team-oriented approach as done in this paper.

Using data obtained from the Lahman Baseball Database, ESPN, and Baseball Reference, a preliminary analysis was performed (Lahman, 2017; “MLB Team Stats,” 2017; “Major League Baseball”, 2017). The analysis included producing scatterplots on numerous predictor variables determine which metrics seemed useful in predicting runs scored and runs allowed. The metrics along with their abbreviations (if applicable) and definitions are listed in the Glossary on page vi. Because sports change over time, only data from the turn of the century onward (2000-2016) were considered. Data points from 2000 to 2014 were used for this portion of the thesis. These chosen metrics were then incorporated into two larger multiple linear regression models (one for runs scored and one for runs allowed). Both models underwent stepwise regression analysis until two

final models containing only significant regressors remained. All computations were done with R statistical software.

Following the determination of the regression equations, variance inflation factors (VIFs) were examined for all model components. Any value above 10 indicated that there was excess collinearity between one or more of the regressors. The defensive model had all eleven variables with VIFs under 5, but the offensive model displayed far too much collinearity to be useful. To compensate for this, Liu estimation was used to fit that model instead. Ridge and Lasso regression were considered, but the error rates were extremely high, sabotaging the usefulness of the models.

In a multiple linear regression (MLR) scenario, the model is as follows:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i \quad (3.1)$$

where  $i$  is the index,  $p$  is the number of predictors, the  $X_i$  are the regressors/predictors,  $\beta_i$  are the regression coefficients, and  $\varepsilon_i$  is the error term (University of Minnesota-Duluth, 2007).

Two key model assumptions are necessary:

1. The regressors must be linearly independent and thus uncorrelated.
2. The residuals/errors are i.i.d (independent and identically distributed) normal with mean=0 and variance= $\sigma^2$ .

For other forms of regression, the parameter estimates are represented by  $\beta_i^*$ .

Liu estimation works with essentially the same MLR model:

$Y_i = \beta_0^* + \beta_1^* x_{i1} + \dots + \beta_p^* x_{ip} + \varepsilon_i$ , where everything is the same as before except for the estimation of the  $\beta_i^*$  terms. The vector of estimators is defined as:

$$\hat{\beta}_d = (X'X + I)^{-1}(X'Y + d\hat{\beta}) \quad (3.2)$$

where  $X$  is the matrix of regressors,  $Y$  is the response vector,  $\hat{\beta}$  is the vector of least squares estimates for the  $\beta_i$  for normal MLR, and  $d$  is Liu's parameter.

This parameter can be estimated in several ways, including MSE minimization, the method used by the liureg R package. The equation for the estimator is:

$$\hat{d} = 1 - \hat{\sigma}^2 \left( \sum_{i=1}^p \frac{1}{\lambda_i(\lambda_i+1)} / \sum_{i=1}^p \frac{\hat{\alpha}_i^2}{(\lambda_i+1)^2} \right) \quad (3.3)$$

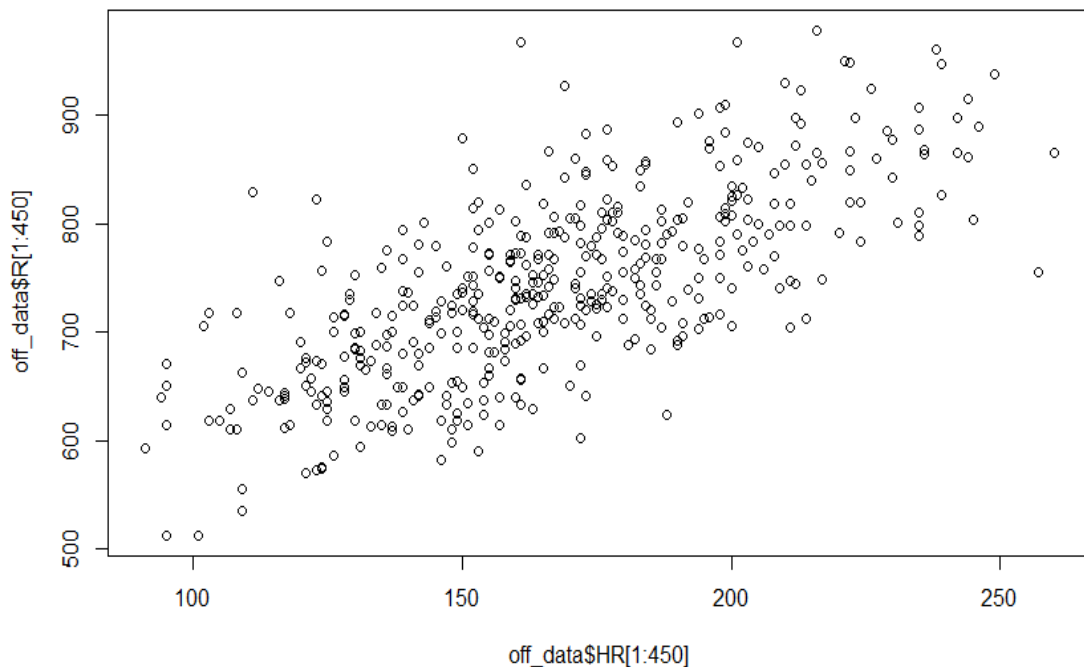
where  $\hat{\sigma}^2$  is the estimated variance,  $p$  is the number of parameters, the  $\lambda_i$  are the eigenvalues of the centered form of  $X'X$ , and the  $\hat{\alpha}_i = q_i' \hat{\beta}_i$ , where  $q_i$  are the corresponding eigenvectors for the  $\lambda_i$  (Liu, 1993).

At this point, the data that were not used in creating the models (2015-2016 data points) were employed in cross-validating the models. After plugging these data into the regression equations, the predicted values for runs were compared to the actual values that were listed in the database. As a result of the stepwise process, the most important statistics as they relate to run scoring and prevention were identified. Discussions on whether the models were adept at predicting their respective response variables will be featured in the Data Analysis chapter. Thereafter the model predictions were inserted into the Pythagorean Expectation formula, where projected wins in a given season were evaluated for accuracy and predictive value.

#### IV. DATA ANALYSIS

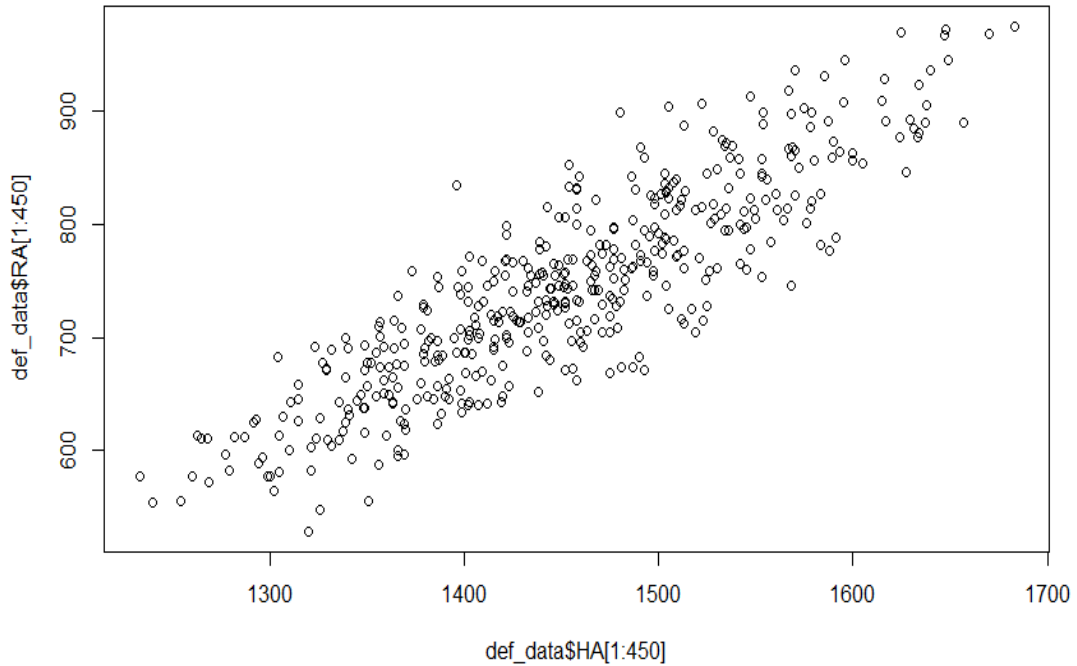
The data were combined from all three sources and compiled into two Excel spreadsheets, separated by offensive and defensive data. All data tables are available upon request, where the abbreviations used are as defined in the Glossary on pages 26-28. Following these are the R codes that contain comments throughout for comprehensible reading. Twenty-three offensive statistics were obtained as well as forty-two defensive statistics.

Scatterplots were examined for each potential variable against Runs Scored (for offense) and Runs Allowed (for defense). Shown below in Figure 1 are a pair of relevant single-variable scatterplots:



**Figure 1A: Scatterplot of Home Runs vs. Runs Scored**





**Figure 1B: Scatterplot of Hits Allowed vs. Runs Allowed**

As seen in these plots, HR and HA seem to have significant correlation with Runs. Other scatterplots are not included here, but plots for variables with significant run correlation are located in the Appendix. Those metrics that did not seem to have any relationship with runs were removed immediately, giving us the following significant variables:

Offense: H, Doubles, HR, BB, SO, BA, OBP, SLG, OPS, OPS Plus, TB, LOB

Defense: SHO, SV, HA, Doubles Allowed, Triples Allowed, HRA, BBA, H9, BB9,

SOA, SOvBB, FIP, WHIP, LOB Against, IPouts, DefEff, E, OBP Against, SLG Against,

OPS Against, TB Against, K9, DIPpct

After running separate stepwise regression procedures for the two models, the model summaries and variance inflation factors were as follows:

```

> summary(off_model)

Call:
lm(formula = R ~ OBP + LOB + TB + BB + BA + H + OPSplus, data = off_data[1:450,
])

Residuals:
    Min       1Q   Median       3Q      Max
-49.122 -11.012   0.025  11.230  44.217

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.190e+02  3.847e+01  -8.293 1.34e-15 ***
OBP          4.701e+03  5.435e+02   8.649 < 2e-16 ***
LOB         -5.982e-01  3.087e-02 -19.378 < 2e-16 ***
TB           1.672e-01  1.211e-02  13.807 < 2e-16 ***
BB           2.318e-01  5.841e-02   3.969 8.43e-05 ***
BA          -6.273e+03  7.773e+02  -8.070 6.70e-15 ***
H            9.286e-01  8.086e-02  11.484 < 2e-16 ***
OPSplus     -1.465e-01  1.482e-01  -0.989  0.323
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.1 on 442 degrees of freedom
Multiple R-squared:  0.9604,    Adjusted R-squared:  0.9597
F-statistic: 1530 on 7 and 442 DF,  p-value: < 2.2e-16

> vif(off_model)
           OBP           LOB           TB           BB           BA           H           OPSplus
94.569723  4.801473  6.656852  26.191398 134.802824 69.189759 2.177026

> summary(def_model)

Call:
lm(formula = RA ~ HA + BBA + HRA + IPouts + E + DIPpct + DefEff +
    SV + LOB + SHO + BB9, data = def_data[1:450, ])

Residuals:
    Min       1Q   Median       3Q      Max
-65.876 -13.186  -0.886  12.792  77.409

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  50.27032  186.07784   0.270 0.787166
HA           0.56974   0.01836  31.040 < 2e-16 ***
BBA          0.35979   0.02285  15.747 < 2e-16 ***
HRA          0.77499   0.05263  14.726 < 2e-16 ***
IPouts      -0.16889   0.02886  -5.853 9.49e-09 ***
E            0.50861   0.06769   7.513 3.27e-13 ***
DIPpct      -2.13819   0.29017  -7.369 8.64e-13 ***
DefEff      830.83624  201.24908   4.128 4.37e-05 ***
SV          -0.55728   0.16475  -3.383 0.000783 ***
LOB         -0.10243   0.03569  -2.870 0.004305 **
SHO         -0.74646   0.34044  -2.193 0.028856 *
BB9          4.49708   4.70295   0.956 0.339486
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.91 on 438 degrees of freedom
Multiple R-squared:  0.9462,    Adjusted R-squared:  0.9449
F-statistic: 700.7 on 11 and 438 DF,  p-value: < 2.2e-16

> vif(def_model)
           HA           BBA           HRA           IPouts           E           DIPpct           DefEff           SV           LOB           SHO           BB9
2.699121 2.369003 1.850432 1.357990 1.253143 4.087135 4.952800 1.444313 3.678013 1.846998 4.132445

```

## Figure 2: Summaries and VIFs for Stepwise Regression Models

The variables inflation factors for the offensive models greatly exceeded our cutoff of 10 for OBP, BB, BA, and H. These results indicated disproportionate collinearity and so

the model was not acceptable. Conversely, the defensive model did not have collinearity issues as evidenced in the figure; all VIFs were below 5, satisfying the necessary independence assumption. Therefore, the model could be used and analyzed. The stepwise process in R chose HA, BBA, HRA, IPouts, E, DIPpct, DefEff, SV, LOB, SHO, and BB9 as significant predictors of RA. The regression was significant: an F-test on the hypotheses  $H_0$ : All  $\beta$ 's are 0 vs.  $H_a$ : At least one of the  $\beta$ 's is nonzero gave an F statistic of 700.7 with a p-value of  $<0.0001$ , allowing us to reject the null hypothesis at the .05 significance level and conclude that the overall regression model was significant.

The summary also included t-tests for each of the regression coefficients individually. These tests were of the form  $H_0: \beta_j = 0$  vs.  $H_a: \beta_j \neq 0$  where the test statistic is  $t_0 = \hat{\beta}_j / SE(\hat{\beta}_j)$  for  $j$  between 0 and  $p$  (the number of predictors). SE is the standard error of the estimate. Despite being selected by the stepwise process, BB9 had a t-value of .956 with a corresponding p-value of .339486 which suggests that this particular metric was not in fact significant. Complicating matters further, the procedure also chose BBA, another walks-based statistic as a regressor. The intercept was likewise deemed ineffective with a t-value of .270 and a p-value of .787166. However, VIFs were kept in check and the model followed the AIC criterion for selection, so neither BB9 nor the intercept were removed. The  $R^2$  value, the coefficient of determination for the model, was .9462, telling us over ninety percent of the variability in Runs Allowed was explained by this model as-is, suggesting a strong model fit and confirming the decision to keep BB9 and the intercept, whose coefficients were higher than those of other variables.

Figure 3 presents a correlation matrix, scatterplot matrix, and residual plot to further examine the model assumptions:

	SHO	SV	HA	HRA	BBA	BB9	LOB	IPouts	DefEff	E	DIPpct
SHO	1.00	0.36	-0.61	-0.53	-0.38	-0.25	-0.29	0.33	0.28	-0.31	0.28
SV	0.36	1.00	-0.41	-0.32	-0.39	-0.20	-0.14	0.38	0.19	-0.24	0.21
HA	-0.61	-0.41	1.00	0.59	0.30	0.18	0.34	-0.38	-0.45	0.33	-0.40
HRA	-0.53	-0.32	0.59	1.00	0.30	0.26	0.20	-0.31	-0.13	0.19	-0.15
BBA	-0.38	-0.39	0.30	0.30	1.00	0.68	0.50	-0.30	-0.10	0.27	-0.11
BB9	-0.25	-0.20	0.18	0.26	0.68	1.00	0.76	-0.23	-0.25	0.16	-0.26
LOB	-0.29	-0.14	0.34	0.20	0.50	0.76	1.00	-0.16	-0.47	0.15	-0.31
IPouts	0.33	0.38	-0.38	-0.31	-0.30	-0.23	-0.16	1.00	0.18	-0.31	0.19
DefEff	0.28	0.19	-0.45	-0.13	-0.10	-0.25	-0.47	0.18	1.00	-0.15	0.84
E	-0.31	-0.24	0.33	0.19	0.27	0.16	0.15	-0.31	-0.15	1.00	-0.08
DIPpct	0.28	0.21	-0.40	-0.15	-0.11	-0.26	-0.31	0.19	0.84	-0.08	1.00

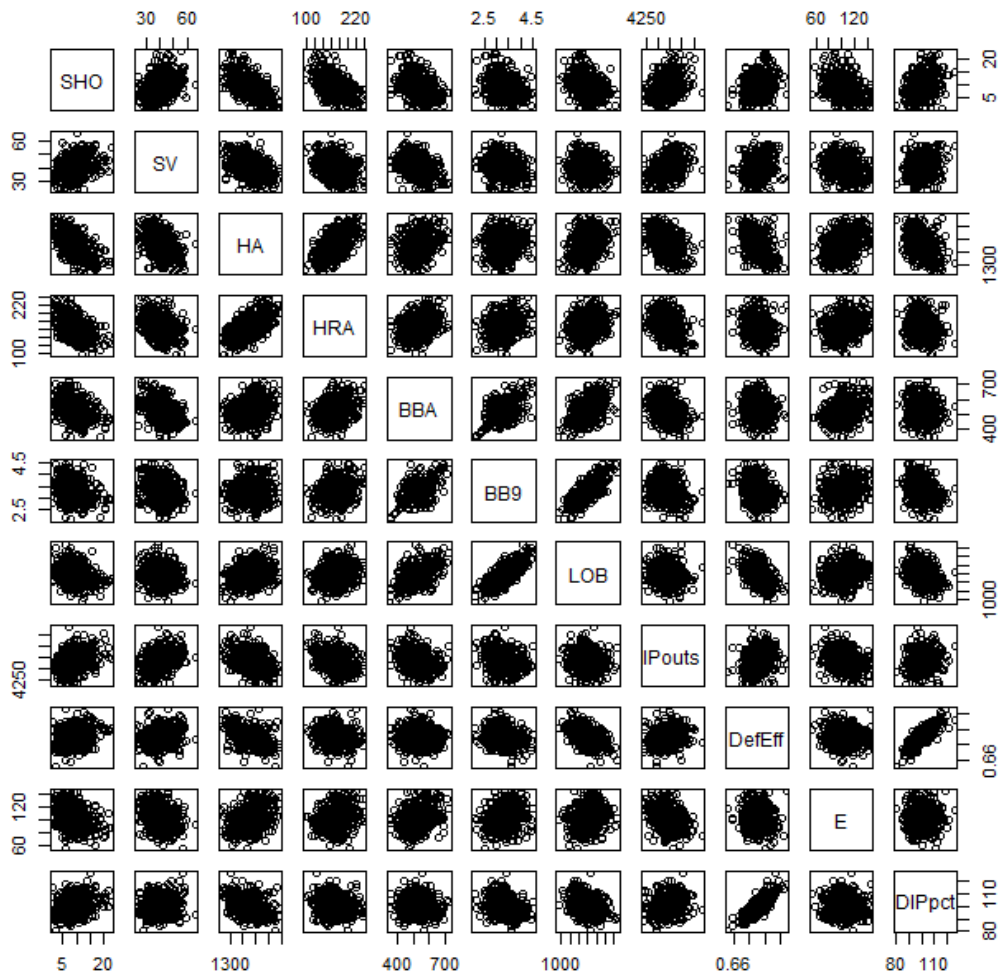
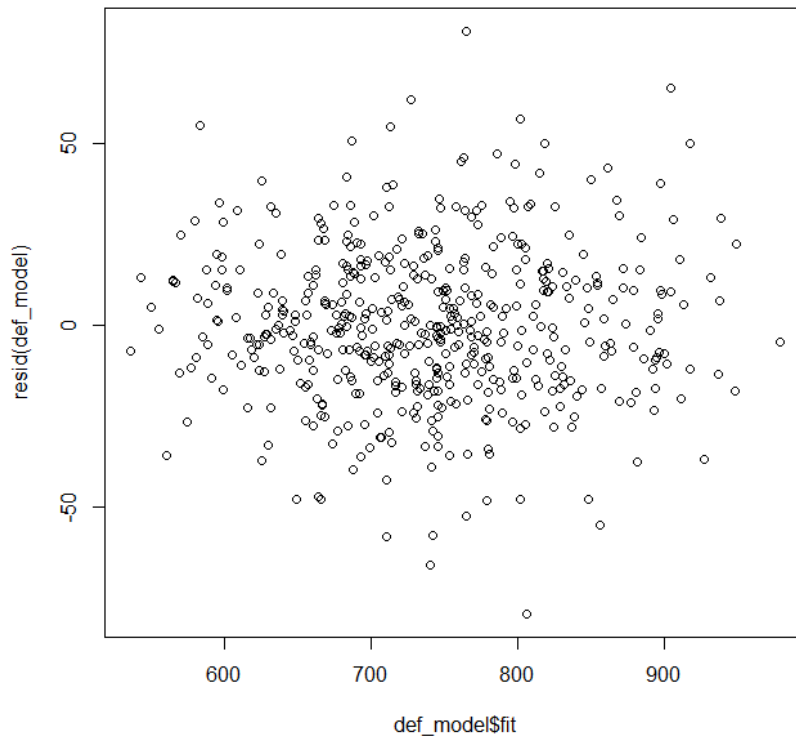


Figure 3A: Correlation Matrix and Scatterplot Matrix for Defensive Model



**Figure 3B: Residual Plot for Defensive Model**

Ignoring the diagonal of the correlation matrix since it is just each variable's correlation with itself (always 1.00), we see very few cases where the absolute value of the correlation was greater than .70. Even these did not exceed more than one per row or column. The scatterplot matrix tells the same story; there are limited linear trends throughout, with most plots consisting of a randomized pattern indicative of little to no correlation, thus satisfying the assumption of the regressors being uncorrelated. The residual plot allows us to test the normality assumption, which is satisfied as seen by the randomness in the plot of residuals versus fitted values. Any sort of pattern would have been a violation of normality, but no such pattern was present here. Partial residual plots

for each of the regressors are in the Appendix, while simple regression summaries are not listed here for the sake of brevity but are available by request.

To try and fix the collinearity problem of the offensive model, Liu estimated regression was run for Runs Scored against each of the twelve possible predictor variables that survived the initial scatterplot phase. Figure 4 shows the model summary output:

```
> summary(off_model)

Call:
lm.default(formula = R ~ H + Doub + HR + BB + SO + BA + OBP +
            SLG + OPS + OPSplus + TB + LOB, data = off_data[1:450, ])

Coefficients for Liu parameter d= 1
              Estimate Estimate (Sc) StdErr (Sc) t-val (Sc) Pr(>|t|)
Intercept -3.194e+02  -3.194e+02  1.951e+03  -0.164 0.869941
H          5.290e-01   5.290e-01  4.067e-01   1.301 0.193409
Doub      -1.267e-01  -1.267e-01  6.301e-02  -2.011 0.044276 *
HR        -3.012e-01  -3.012e-01  1.463e-01  -2.058 0.039565 *
BB         2.096e-01   2.096e-01  5.961e-02   3.516 0.000438 ***
SO         2.544e-04   2.544e-04  8.613e-03   0.030 0.976440
BA        -4.868e+03  -4.868e+03  2.217e+03  -2.195 0.028142 *
OBP        4.247e+03   4.247e+03  1.850e+03   2.296 0.021658 *
SLG       -1.889e+03  -1.889e+03  1.918e+03  -0.985 0.324752
OPS        7.114e+02   7.114e+02  1.772e+03   0.401 0.688086
OPSplus   -1.036e-01  -1.036e-01  1.551e-01  -0.668 0.504183
TB         4.786e-01   4.786e-01  2.656e-01   1.802 0.071509 .
LOB       -6.037e-01  -6.037e-01  3.166e-02  -19.068 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Liu Summary
      R2 adj-R2      F AIC  BIC      MSE
d=1 0.9609 0.9598 896.3 2565 5364 15157062
-----
```

**Figure 4: Summary of Liu Offensive Model with 12 Variables**

Although the  $R^2$  of .9609 was extremely high, the p-values of the t-tests for the respective variables showed some obviously insignificant predictors. H, SO, SLG, OPS, OPSplus,

and LOB had p-values of .193409, .976440, .324752, .688086, .504183, and .071509, respectively, all of which were greater than the significance level of  $\alpha=.05$ . These variables were subsequently removed. It is worth noting that the intercept also had a high p-value, but there was no need to examine it before a reduced model was decided upon.

Before running the regression again with the six variables that were not eliminated, a correlation matrix was evaluated to see if any clear trends could be identified as seen in Figure 5:

	Doub	HR	BB	BA	OBP	LOB
Doub	1.00	0.31	0.31	0.61	0.59	0.39
HR	0.31	1.00	0.43	0.38	0.52	0.16
BB	0.31	0.43	1.00	0.28	0.73	0.75
BA	0.61	0.38	0.28	1.00	0.85	0.45
OBP	0.59	0.52	0.73	0.85	1.00	0.71
LOB	0.39	0.16	0.75	0.45	0.71	1.00

**Figure 5: Correlation Matrix for 6 Remaining Offensive Variables after Liu Regression**

One row/column stood out: OBP. As it is constructed from various metrics including walks were one of the other five remaining variables, OBP had correlations between .52 and .85 with every other variable. It was decided that to avoid more collinearity problems, OBP would be removed as well.

Figure 6 shows the summary of the final offensive model when Liu regression was performed for Runs Scored against Doub, HR, BB, BA, and LOB:

```

> summary(off_model)

Call:
liu.default(formula = R ~ Doub + HR + BB + BA + LOB, data = off_data[1:450,
])

Coefficients for Liu parameter d= 1
      Estimate Estimate (Sc) StdErr (Sc) t-val (Sc) Pr(>|t|)
Intercept -517.62282    -517.62282    52.26585    -9.904 < 2e-16 ***
Doub       0.26553      0.26553     0.04688     5.664 1.47e-08 ***
HR         0.71563      0.71563     0.03887    18.409 < 2e-16 ***
BB         0.53980      0.53980     0.02578    20.936 < 2e-16 ***
BA        4533.31413    4533.31413   119.43268    37.957 < 2e-16 ***
LOB        -0.35430     -0.35430     0.03182   -11.134 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Liu Summary
      R2 adj-R2    F  AIC  BIC  MSE
d=1  0.938 0.9373 1345 2759 5529 14264
-----

```

**Figure 6: Summary of Final Liu Offensive Model**

Once again the model resulted in a stellar  $R^2$  of .938, so 93.8% of the variability in Runs Scored was explained by the reduced model including Doub, HR, BB, BA, and LOB. The F-statistic was 1345, where the numerator degrees of freedom were  $n=450$  data points and the denominator degrees of freedom where  $n-p-1=444$ . The test statistic gave a p-value of  $<0.0001$ , showing the major significance of the overall model. The largest p-value in the t-tests for the five variables on their own was  $1.47 \times 10^{-8}$ , so these too were significant, including the intercept which had not been significant in the full model. The MSE of the model was 14264.17, with the minimum MSE occurring when the parameter  $d$  was equal to 1.

With a statistically significant model in place, it was necessary to calculate the VIFs for each variable to see if collinearity had been mitigated. The liureg package in R does not have a built in function compatible with “vif”, so these were calculated manually by



running Liu regression for each predictor variable against the four others, finding each respective  $R^2$  value, and applying the formula  $VIF = \frac{1}{1-R^2}$ . The results are summarized in

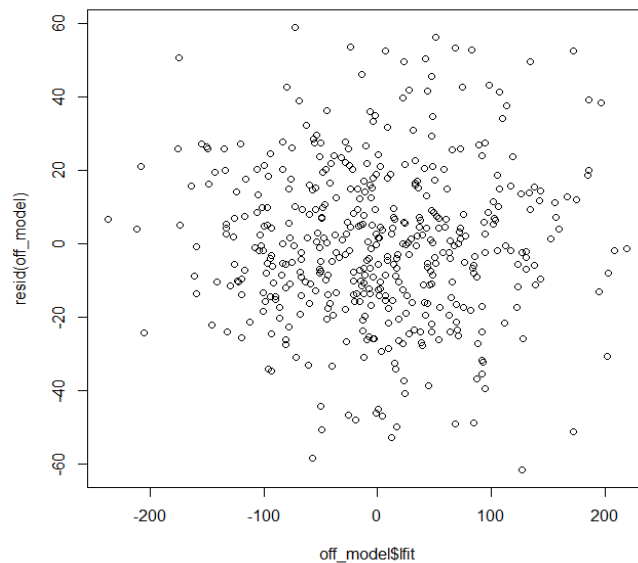
Figure 7:

Variable:	Doub	HR	BB	BA	LOB
VIF:	1.668613	1.694915	3.281917	2.046664	3.280840

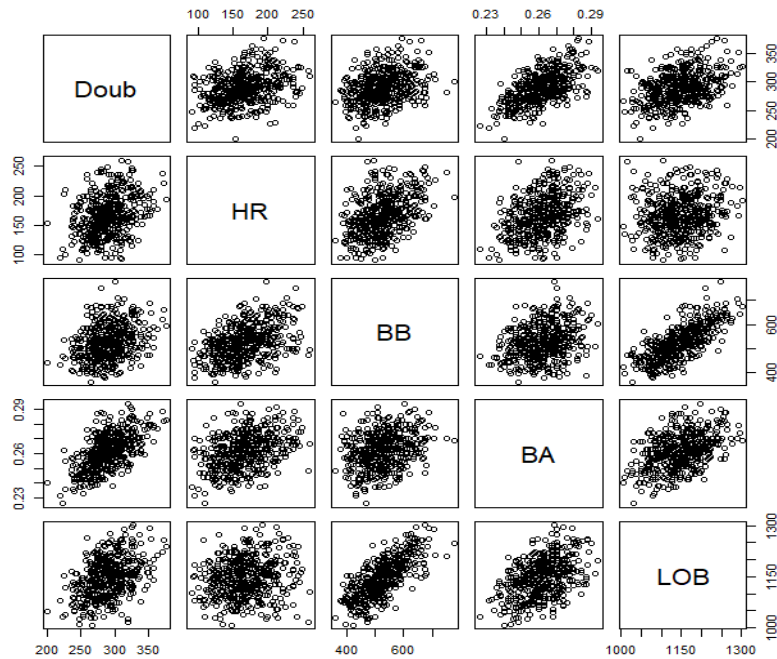
**Figure 7: VIFs for Final Liu Offensive Model**

The VIFs from the reduced Liu model were all under 5, which was a massive improvement over the regular MLR version. Collinearity was no longer an issue, so the model was accepted for use in cross validation.

As with the defensive model, the offensive model was evaluated to see if the regression assumptions were satisfied. The correlation matrix, scatterplot matrix, and plot of residuals versus fitted values are displayed in Figure 8 below:



**Figure 8A: Residual Plot for Final Liu Offensive Model**



	Doub	HR	BB	BA	LOB
Doub	1.00	0.31	0.31	0.61	0.39
HR	0.31	1.00	0.43	0.38	0.16
BB	0.31	0.43	1.00	0.28	0.75
BA	0.61	0.38	0.28	1.00	0.45
LOB	0.39	0.16	0.75	0.45	1.00

**Figure 8B: Correlation Matrix and Scatterplot Matrix for Final Liu Offensive Model**

We see only one correlation that was above .70 in the matrix; BB and LOB had a correlation of .75. The scatterplot also shows only slight linear trends. Therefore the independence assumption is satisfied. Looking at the residual plot, the points are randomly distributed with no evidence of a pattern. Thus the normality assumption is also satisfied and the model is appropriate.

From the outputs in Figure 2 (defensive) and Figure 6 (offensive), we can see that the fitted linear models for Runs Allowed and Runs Scored were:

$$R = -517.62282 + .26553 * Doub + .71563 * HR + .53980 * BB + 4533.31413 * BA - .35430 * LOB \quad (4.1)$$

$$RA = 50.27032 + .56974 * HA + .35979 * BBA + .77499 * HRA - .16889 * IPouts + .50861 * E - 2.13819 * DIPpct + 830.83624 * DefEff - .55728 * SV - .10243 * LOB - .74646 * SHO + 4.49708 * BB9 \quad (4.2)$$

These were used to test the remaining 60 data points from 2015-2016, and the cross validation results table is located in the Appendix, with the key points discussed here.

Using both models to fit the data, the program calculated predicted runs scored and allowed for each set of values of the regressors. It then calculated a signed error percentage between the predicted runs and the actual runs from the database. Each model performed very well; taking the mean relative error percentage for the two models, the defensive model yielded a 2.67% error rate and the offensive model produced a 2.58% error rate. The predicted run totals were plugged into the Pythagorean Expectation formula (with the newer 1.83 exponent) and compared to the Pythagorean Expectation values when using actual runs. The mean relative error percentage for this was 3.74%. Finally, the model Pythagorean Expectations were tested against real winning percentages. There was more variation here, but the error percentage was still relatively low at 9.53%.

## V. CONCLUSION

This study successfully produced models for predicting run creation and prevention in the sport of baseball. The significant variables for runs scored as determined by stepwise regression were: doubles, home runs, walks, batting average, and runners left on base. Liu regression and elimination for collinearity yielded the following significant predictors of runs allowed: hits allowed, walks allowed, home runs allowed, infield putouts, errors, defense-independent earned run average ratio, defensive efficiency ratio, saves, runners left on base, shutouts, and walks per nine innings.

A somewhat surprising result was that both models were dominated by traditional statistics. Only defense-independent earned run average ratio and defensive efficiency ratio made the defensive model among the sabermetric statistics, while no sabermetric statistics were included in the offensive model. An obvious reason for this is that since sabermetric statistics are largely constructed from traditional statistics, they were too correlated with other variables to be included in the model. However, variance inflation factors were low for the defensive model even with two sabermetric statistics, suggesting that they simply might not have been as useful as their highly touted nature made them appear.

Runners left on base and home runs made the models in both their offensive and defensive forms, the only two metrics to accomplish that feat. This implies that LOB and HR are very important in analyzing the game of baseball as a whole. Walks allowed and walks per nine innings both qualified for the defensive model, and the correlation between them (.68) was lower than one might imagine given the variables' names. One other fascinating result was that the two models had approximately the same error rate (Offensive: 2.58%, Defensive: 2.67%) when the offensive model used five variables and the defensive model used eleven; the five run scoring metrics are much stronger

predictors individually than are the defensive metrics. Looking back at the scatterplots reveals that the data have a tighter fit for the significant offensive predictors, supporting this hypothesis.

Similarly low error rates were produced for the Pythagorean Expectation (3.74%) and win ratio (9.53%). Possessing the ability to accurately determine which metrics influence winning has important ramifications for the MLB. Teams can use this information to gain an advantage over their opponents when constructing their rosters. By pinpointing what they desire in players that differs from standard requirements, executives can target players that might otherwise not have gotten the same sort of attention as “top prospects” and obtain them via the draft and free agency. This in turn leads to a more competitive team, and possibly a lower payroll.

There are a few ways in which this study can be improved in the future. The first of these is to acquire more data. Once more seasons play out, more applicable data will become available. Next, new statistics could be developed over time that more strongly correlate with runs and winning. At this time twenty years ago, defense-independent earned run average ratio and defensive efficiency, two metrics that were considered vital by the stepwise regression procedure, did not even exist. Finally, current data that are inaccessible could become public knowledge. For instance, pitch velocity and bat head speed are two statistics that baseball aficionados discuss regularly. However, neither of these are available in a database except to the MLB and the thirty teams. As we move toward a more data-driven society, previously unimaginable techniques will come to the forefront and help make baseball prediction than ever before.

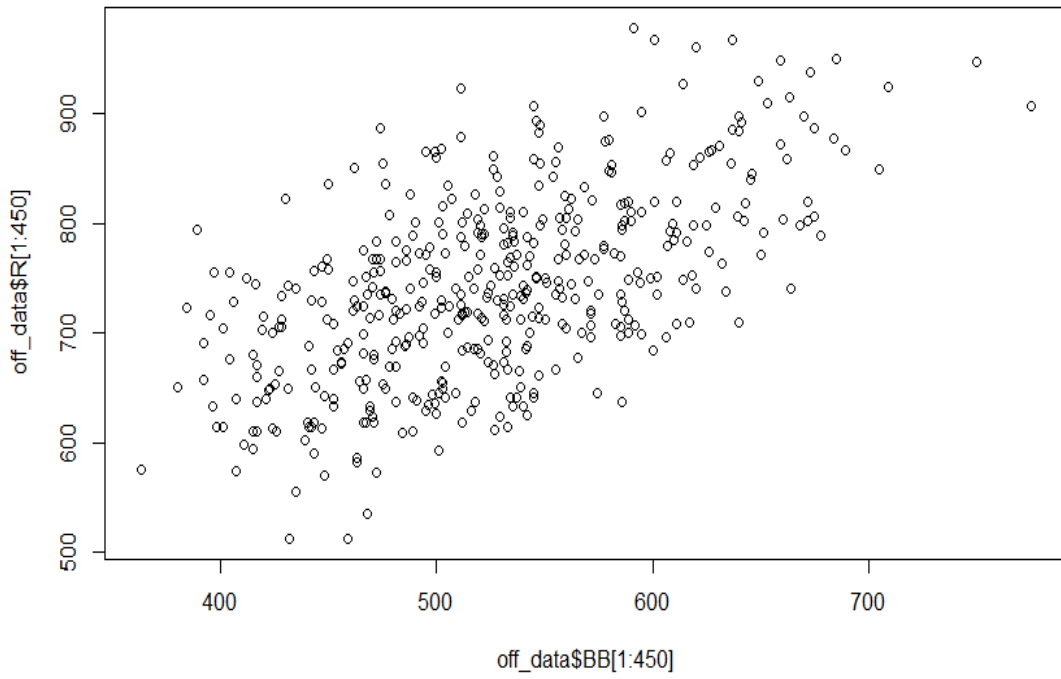
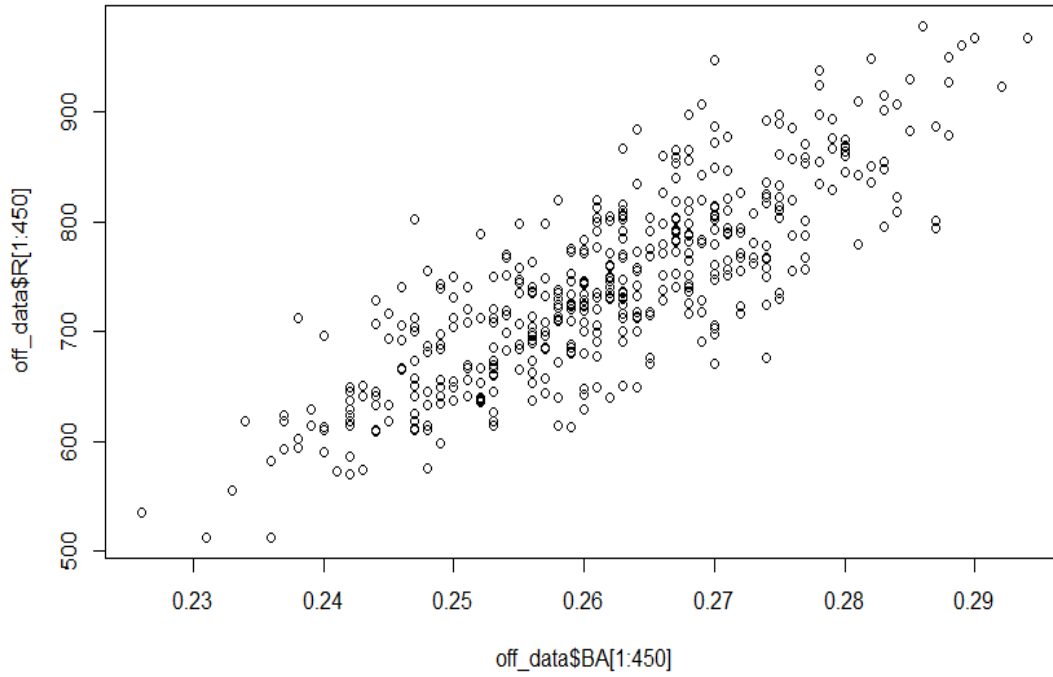
## REFERENCES

1. Albert, Jim. "An Introduction to Sabermetrics." *Department of Mathematics and Statistics*. Bowling Green State University, [www-math.bgsu.edu/~albert/papers/saber.html](http://www-math.bgsu.edu/~albert/papers/saber.html). Accessed 9 Oct. 2017.
2. Alheety, M. I. and Kibria, B. M. Golam. "ON THE LIU AND ALMOST UNBIASED LIU ESTIMATORS IN THE PRESENCE OF MULTICOLLINEARITY WITH HETEROSCEDASTIC OR CORRELATED ERRORS." *Surveys in Mathematics and its Applications*, vol. 4, 2009, pp. 155-167.
3. Arthur, Rob. "How Baseball's New Data Is Changing Sabermetrics." *FiveThirtyEight*, 17 March 2016, [www.fivethirtyeight.com/features/how-baseballs-new-data-is-changing-sabermetrics/](http://www.fivethirtyeight.com/features/how-baseballs-new-data-is-changing-sabermetrics/). Accessed 20 Oct. 2017.
4. Beneventano, Philip, et al. "Predicting Run Production and Run Prevention in Baseball: The Impact of Sabermetrics." *International Journal of Business, Humanities and Technology*, vol. 2, no. 50, 2012, pp. 67-75, [citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.3155&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.3155&rep=rep1&type=pdf). Accessed 9 Oct. 2017.
5. Birnbaum, Phil. "A Guide to Sabermetric Research." Society for American Baseball Research, [www.sabr.org/sabermetrics](http://www.sabr.org/sabermetrics). Accessed 9 Oct. 2017.
6. Kaçiranlar, Selahattin et al. "A NEW BIASED ESTIMATOR IN LINEAR REGRESSION AND A DETAILED ANALYSIS OF THE WIDELY-ANALYSED DATASET ON PORTLAND CEMENT." *Sankhyā: The Indian Journal of Statistics*, vol. 61, Series B, Pt. 3, pp. 443-459.
7. Lahman, Sean. "Lahman's Baseball Database." *SeanLehman.com*, 26 February, 2017, <http://www.seanlahman.com/baseball-archive/statistics/>. Accessed 9 Oct. 2017.
8. Liu, Kejian. "Using Liu-Type Estimator to Combat Collinearity." *Communications in Statistics – Theory and Methods*, vol. 32, no. 5, 2003, pp. 1009-1020.
9. Liu, Kejian. "A New Class of Biased Estimate in Linear Regression." *Communications in Statistics – Theory and Methods*, vol. 22, no. 2, 1993, pp. 393-402.
10. "Major League Baseball Team Win Totals." *Baseball Reference*, 2017, <https://www.baseball-reference.com/leagues/MLB/>. Accessed 9 Oct. 2017.

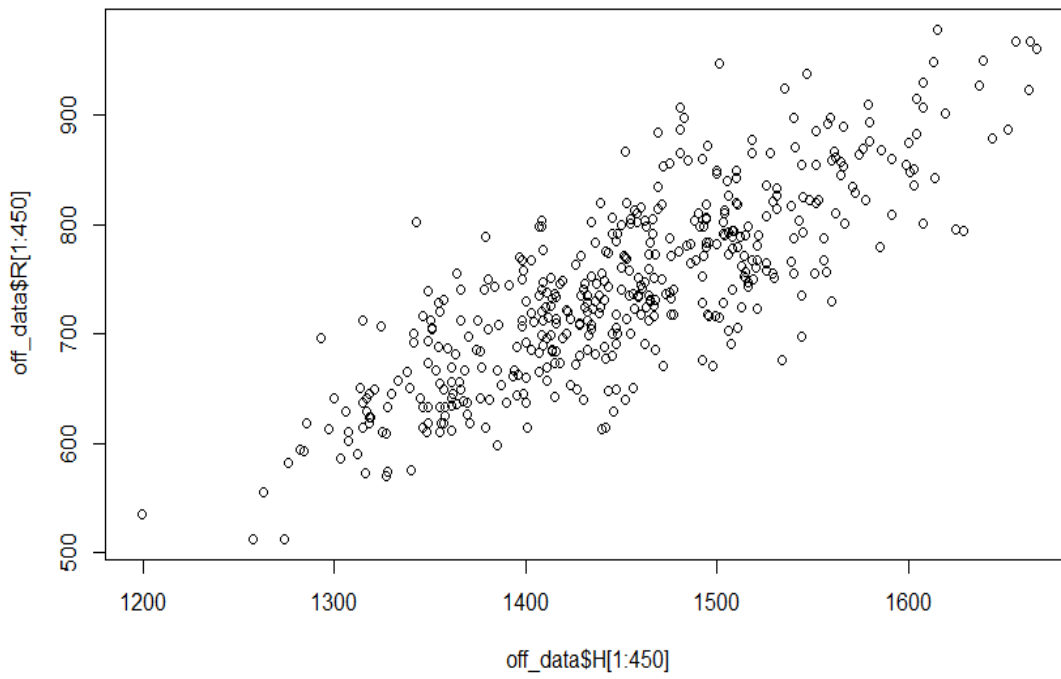
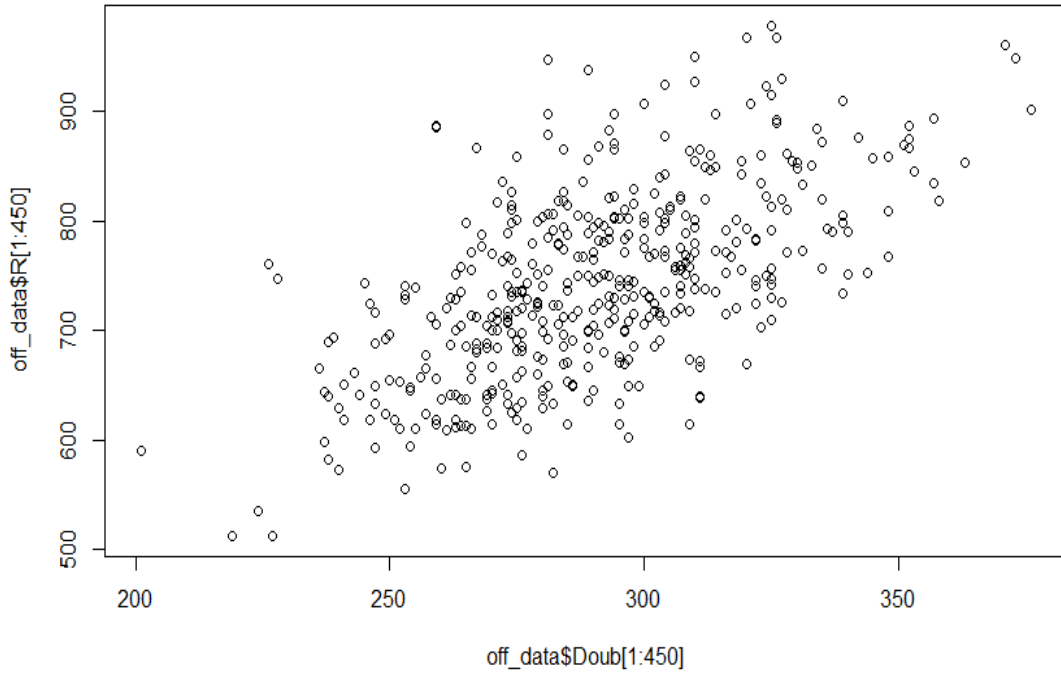
11. “MLB Team Stats – 2017.” *ESPN*, 2017, [http://www.espn.com/mlb/stats/team/\\_/stat/batting](http://www.espn.com/mlb/stats/team/_/stat/batting). Accessed 9 Oct. 2017.
12. Moy, Dennis. “REGRESSION PLANES TO IMPROVE THE PYTHAGOREAN PERCENTAGE.” *Department of Statistics*. University of California-Berkeley, 2006, [https://www.stat.berkeley.edu/~aldous/157/Old\\_Projects/moy.pdf](https://www.stat.berkeley.edu/~aldous/157/Old_Projects/moy.pdf). Accessed 20 Oct. 2017.
13. “Technical Report 2007.” *Mathematics and Statistics Department*, University of Minnesota-Duluth, 2007, [www.d.umn.edu/math/Technical%20Reports/Technical%20Reports%202007-2008/TR\\_2008\\_7/TR\\_2008\\_07.pdf](http://www.d.umn.edu/math/Technical%20Reports/Technical%20Reports%202007-2008/TR_2008_7/TR_2008_07.pdf). Accessed 20 Oct. 2017.
14. Vollmayr-Lee, Ben. “Runs Scored: Correlations.” *College of Engineering*. Bucknell University, 2001, [www.eg.bucknell.edu/~bvollmay/baseball/runs1.html](http://www.eg.bucknell.edu/~bvollmay/baseball/runs1.html). Accessed 20 Oct. 2017.

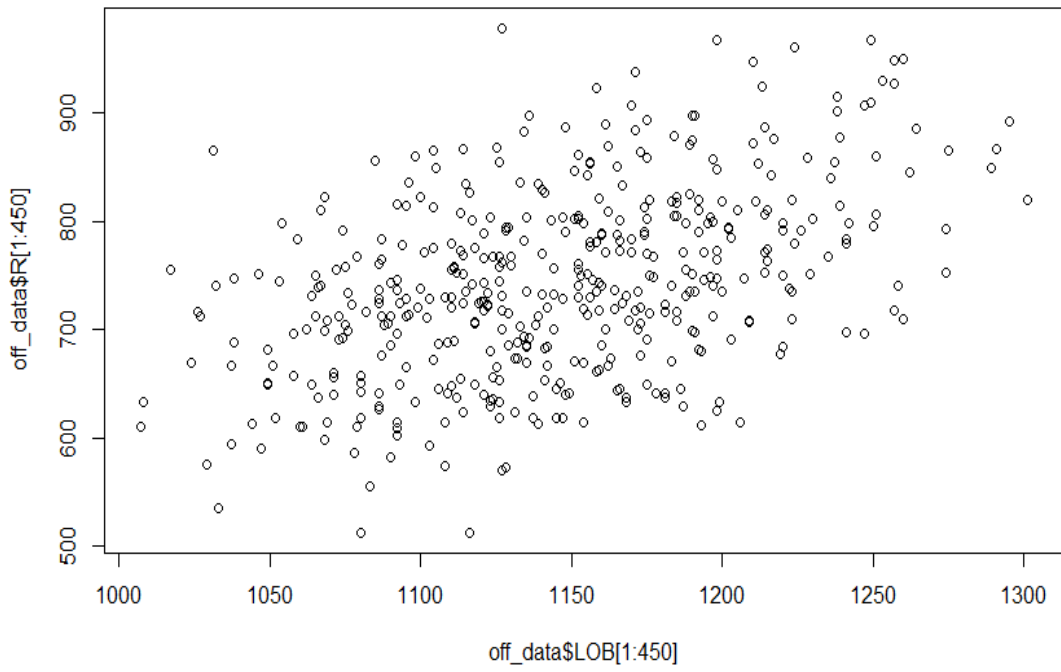
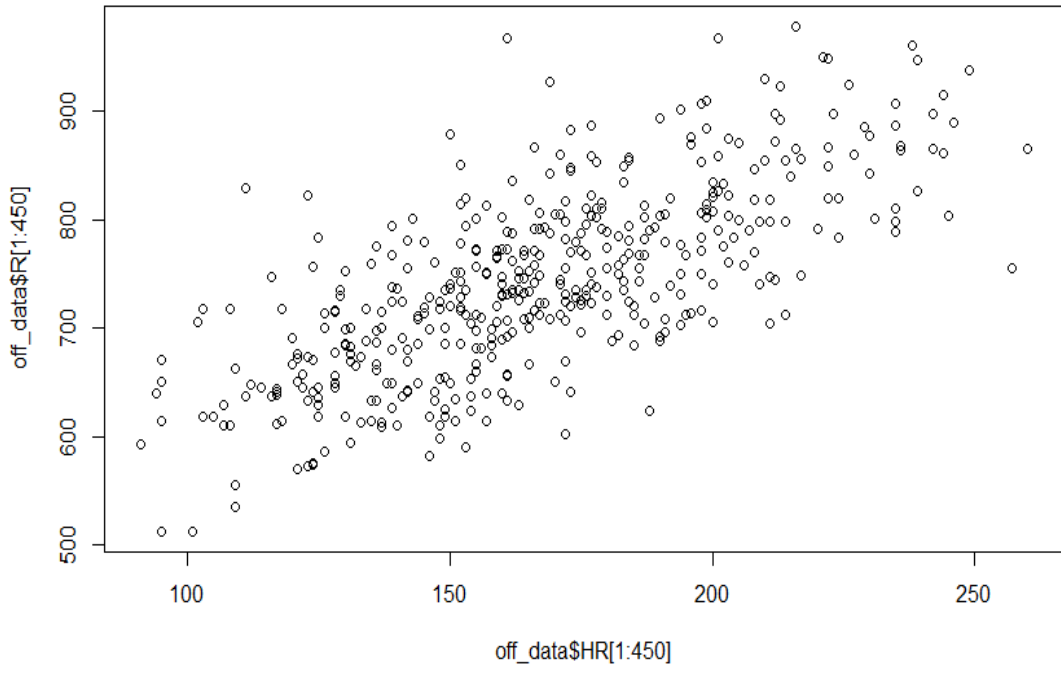
## APPENDIX

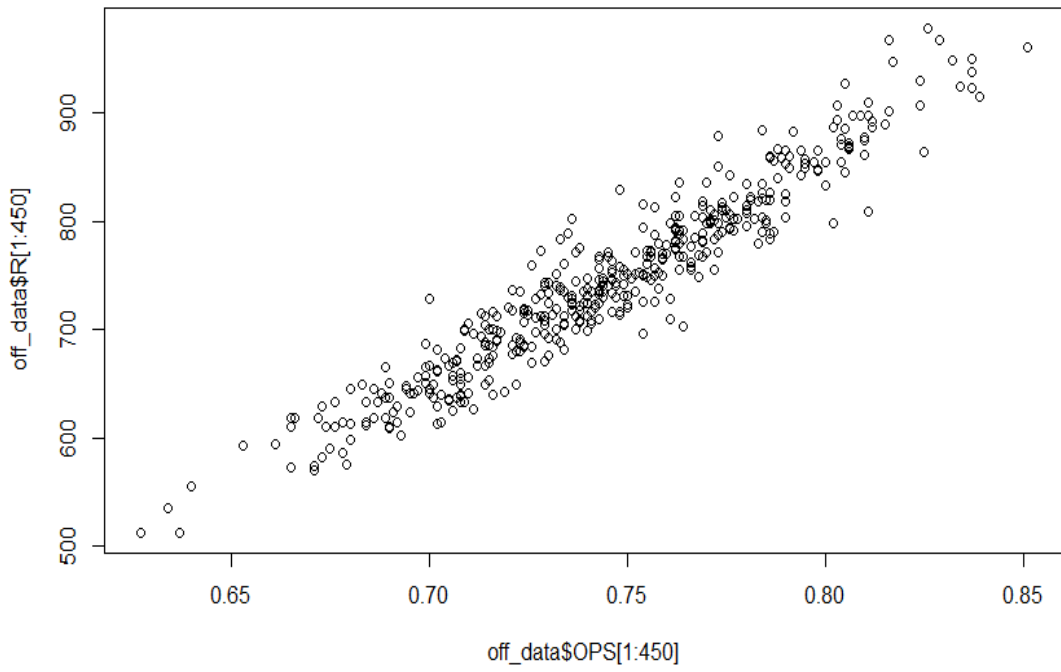
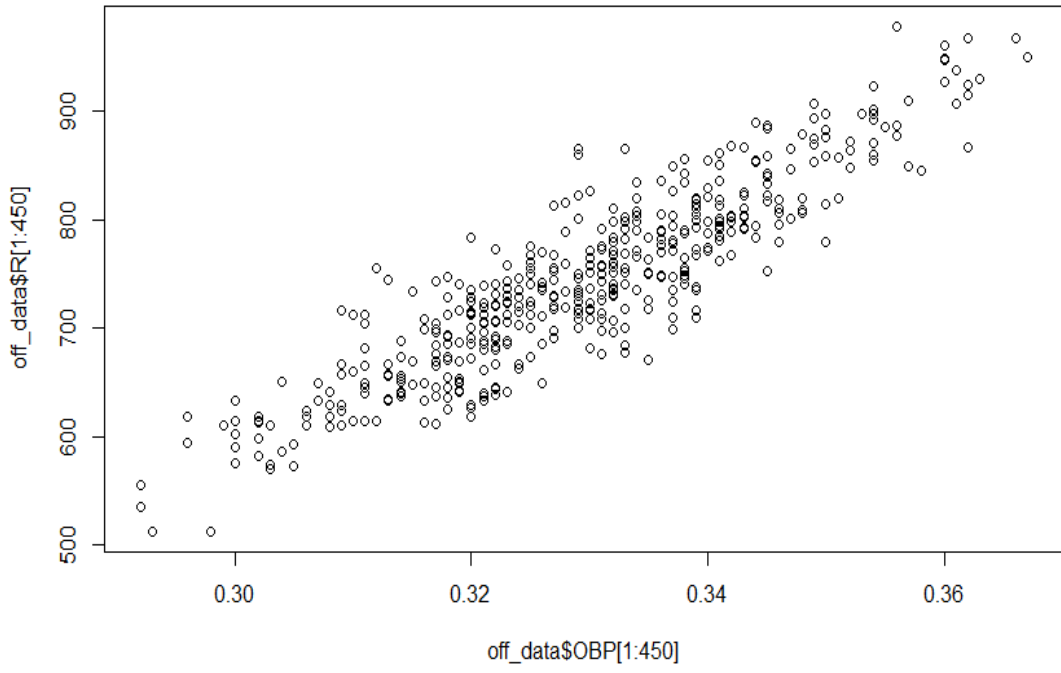
### A. Scatterplots for Offense Variables Correlated with Runs Scored

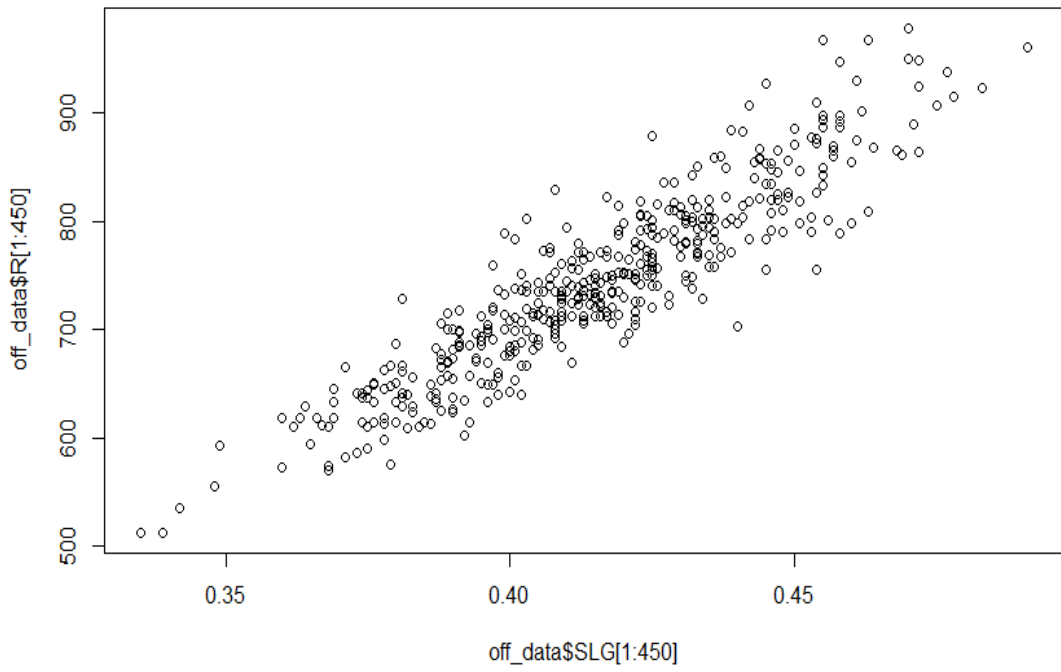
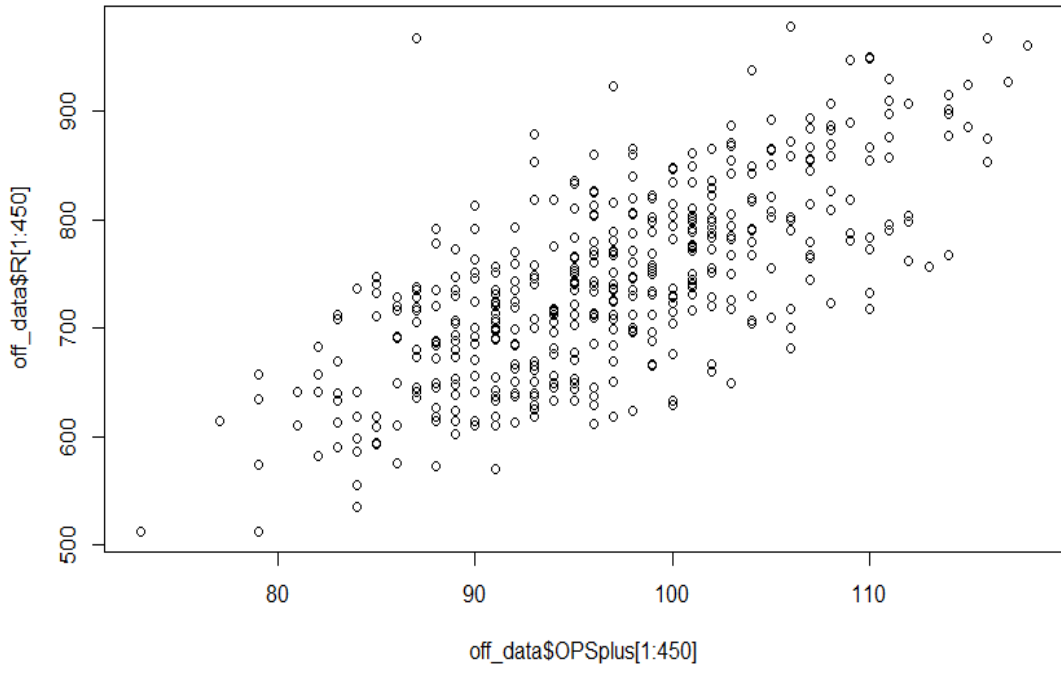


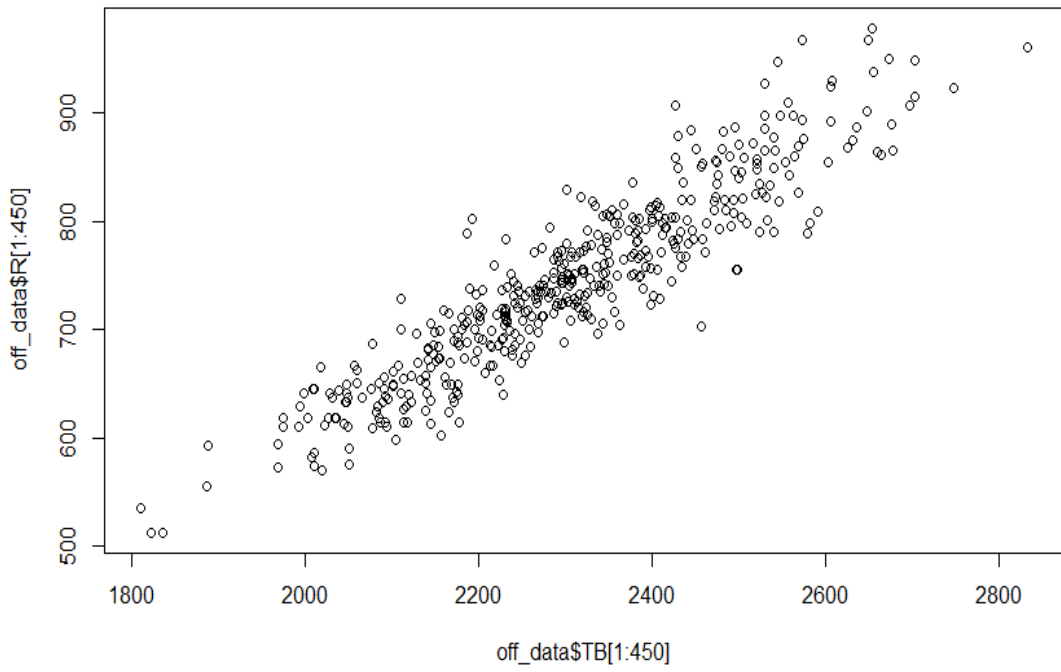
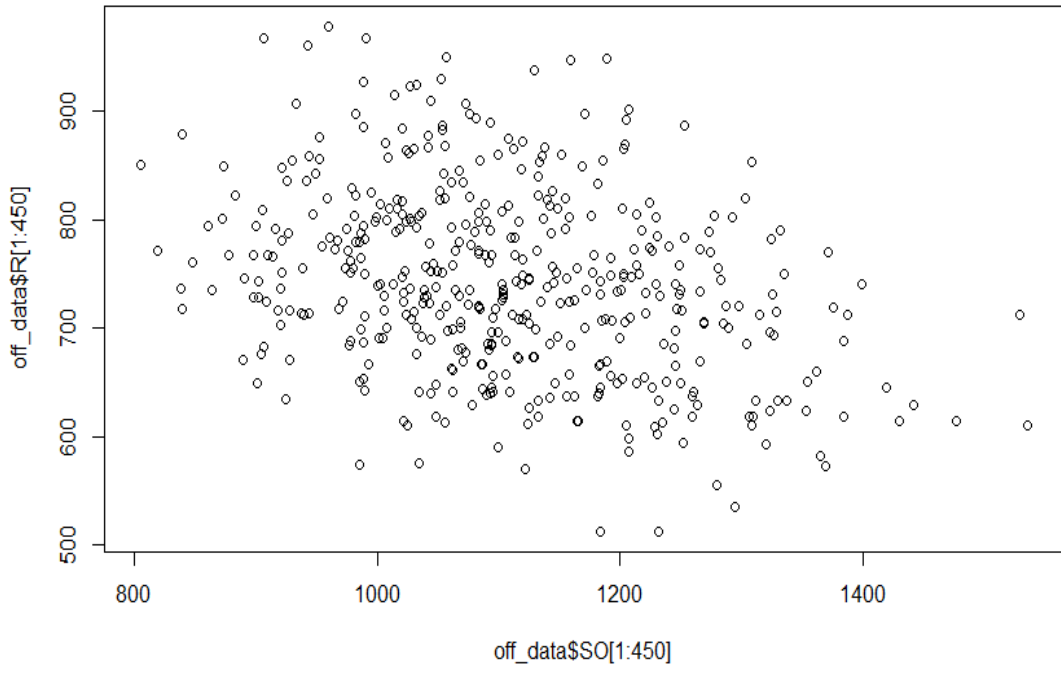




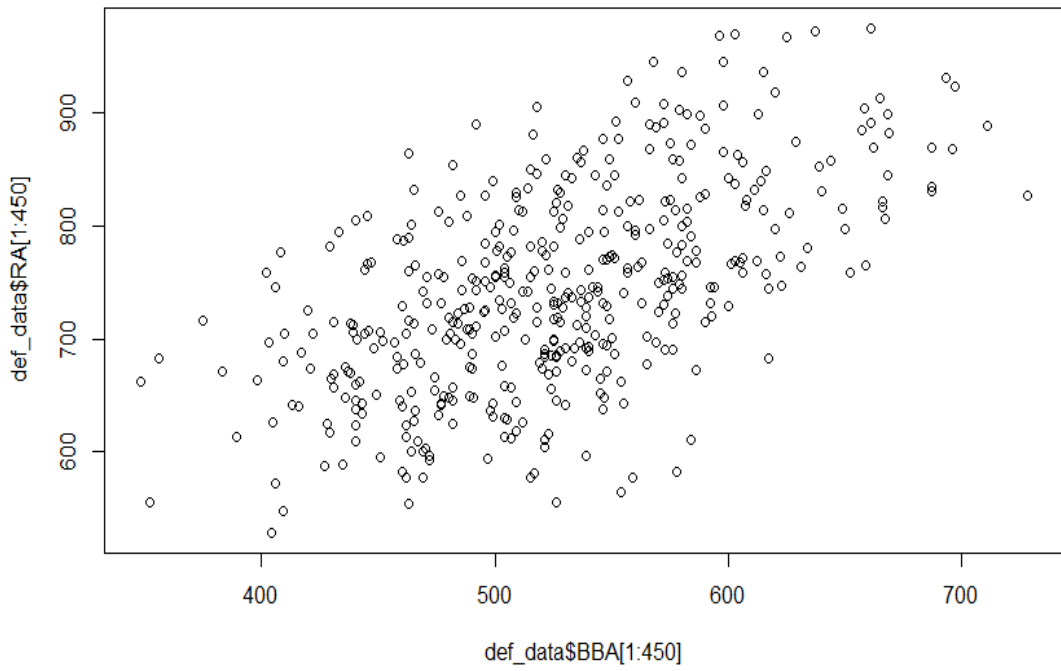
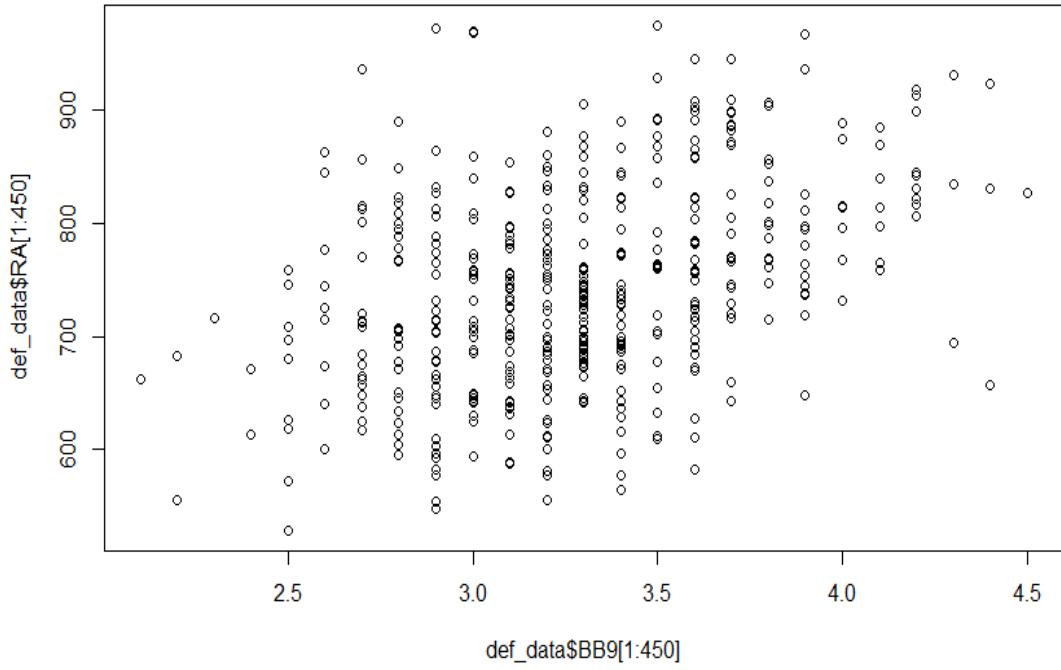


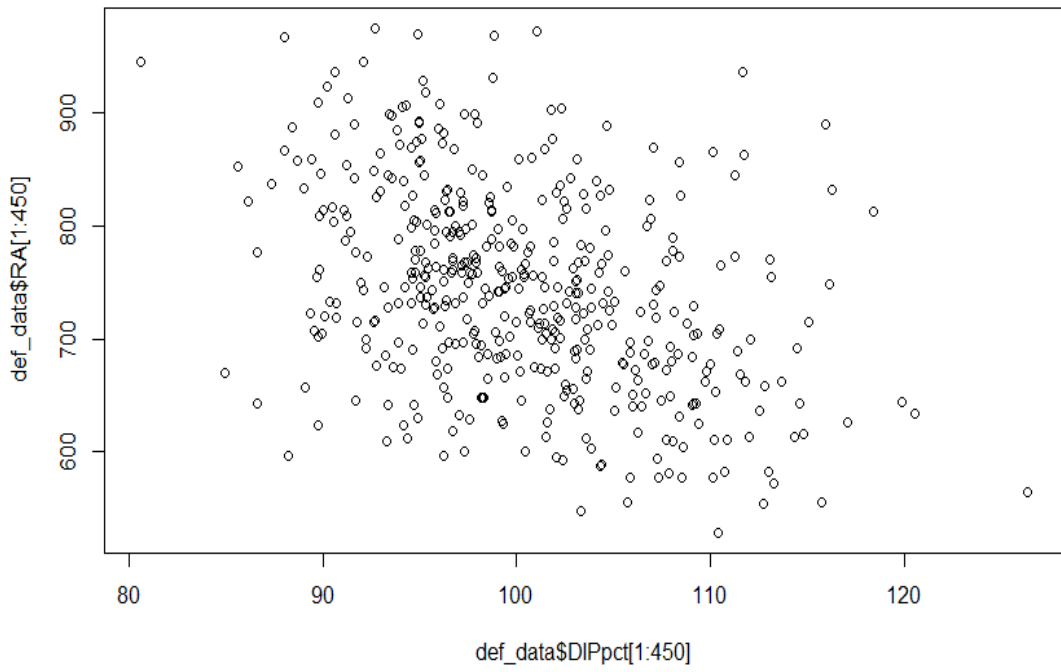
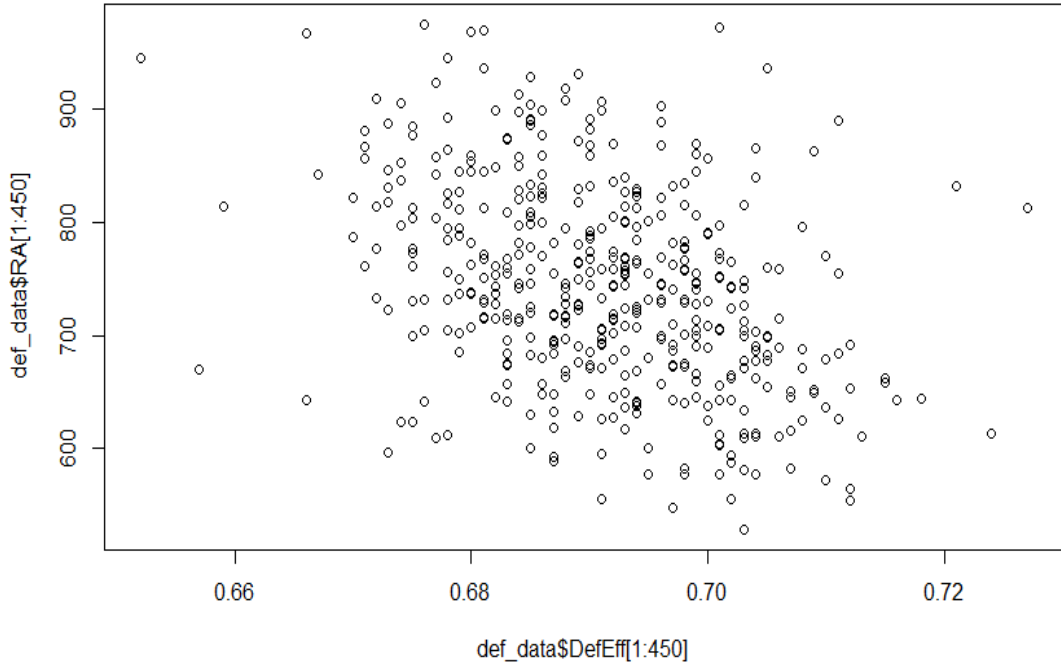


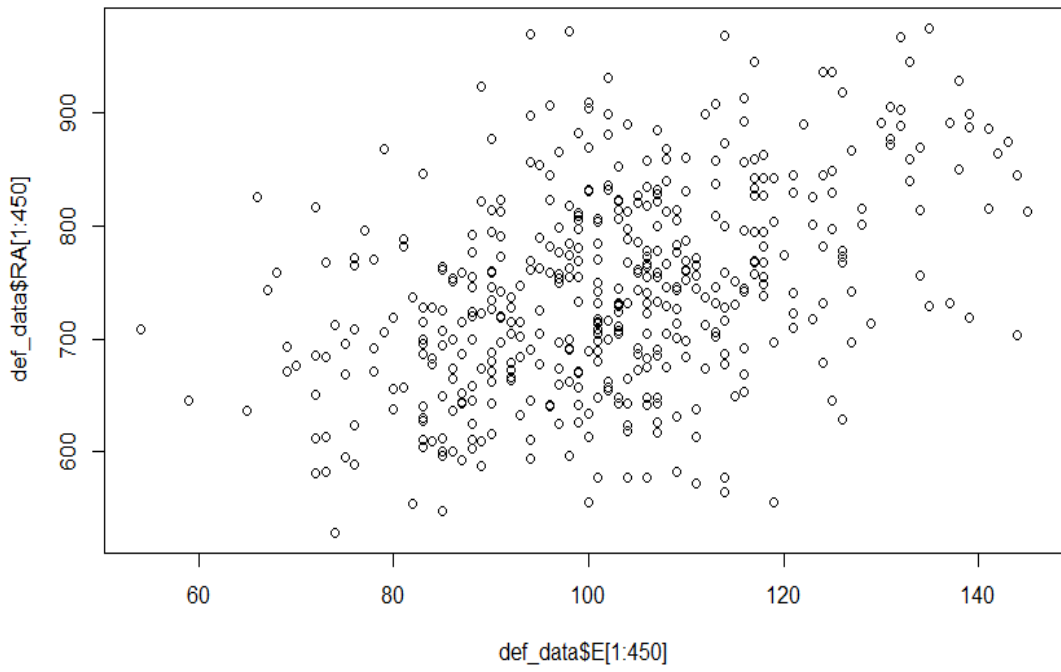
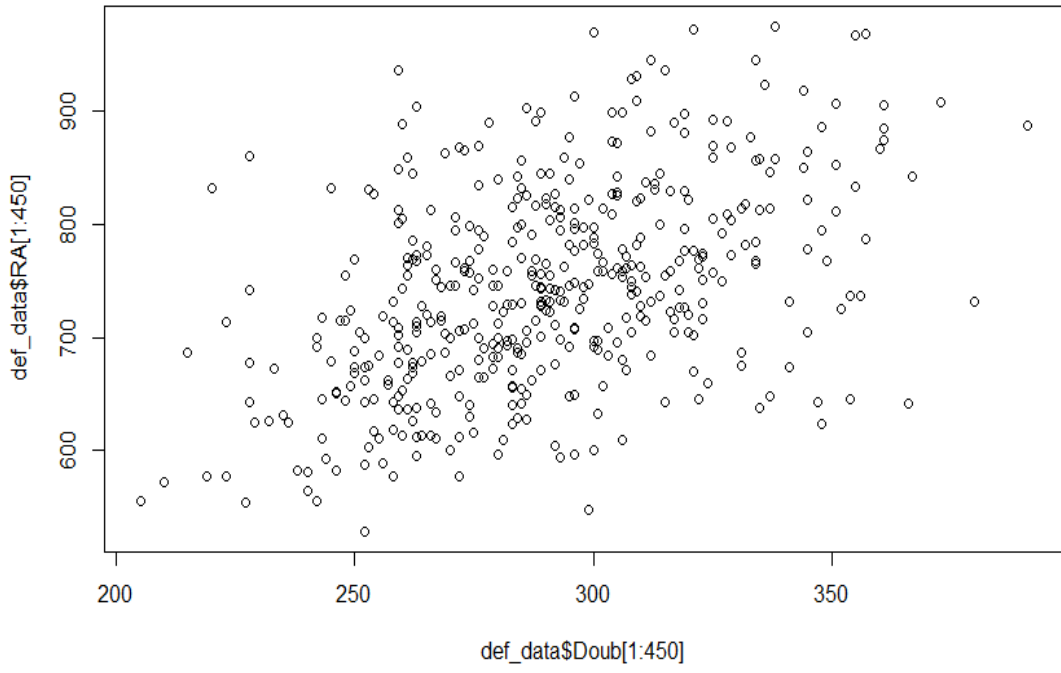




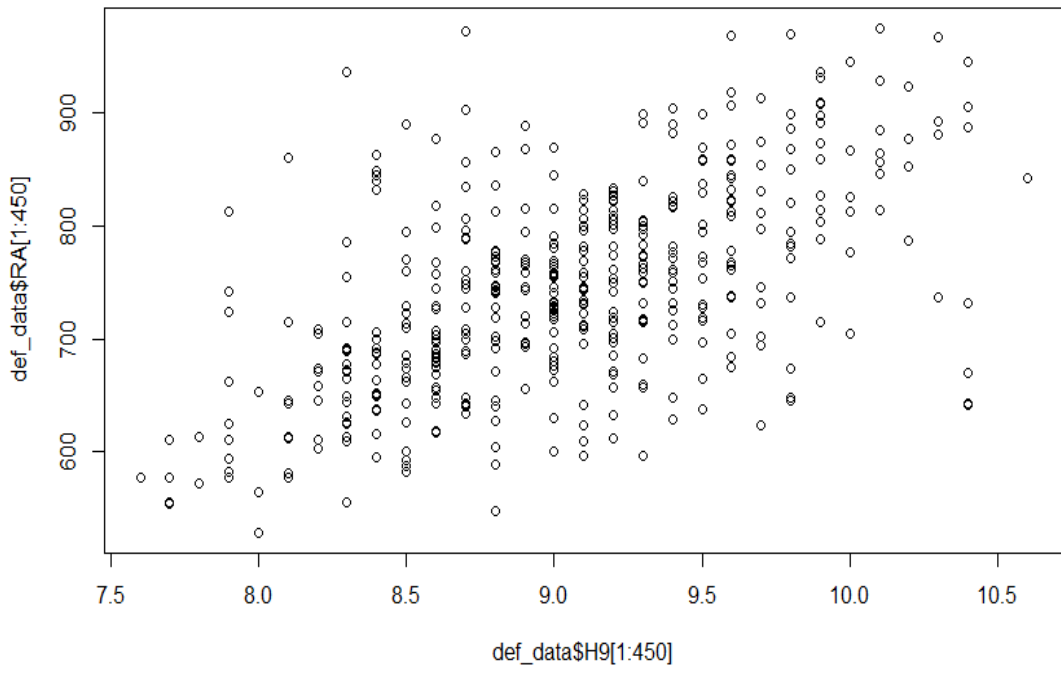
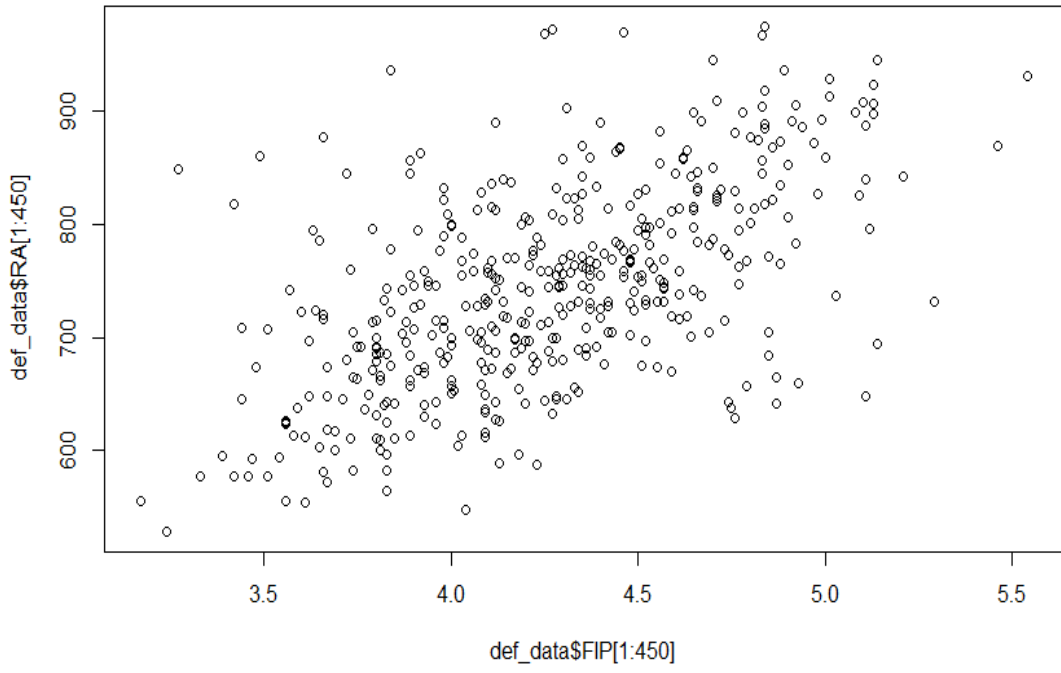
B. Scatterplots for Defensive Variables Correlated with Runs Allowed

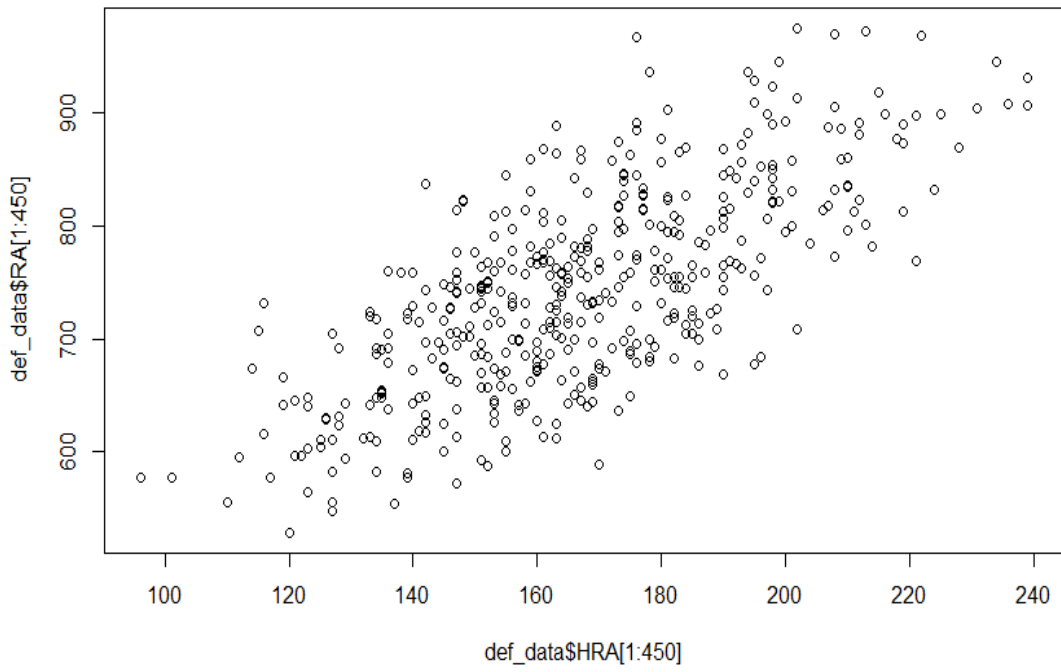
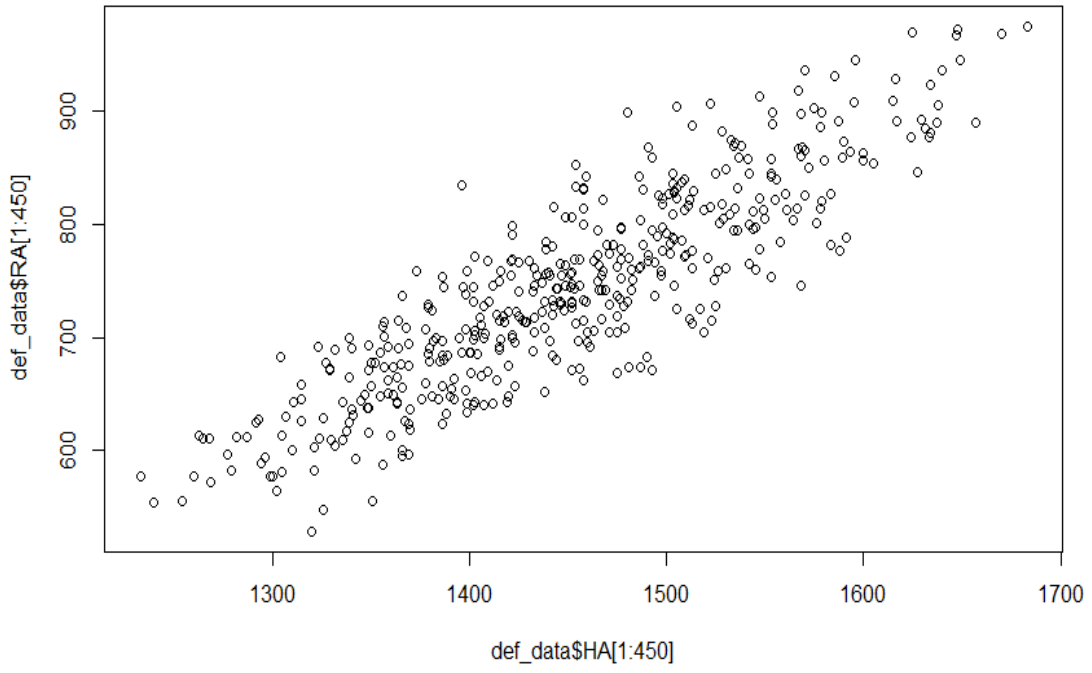


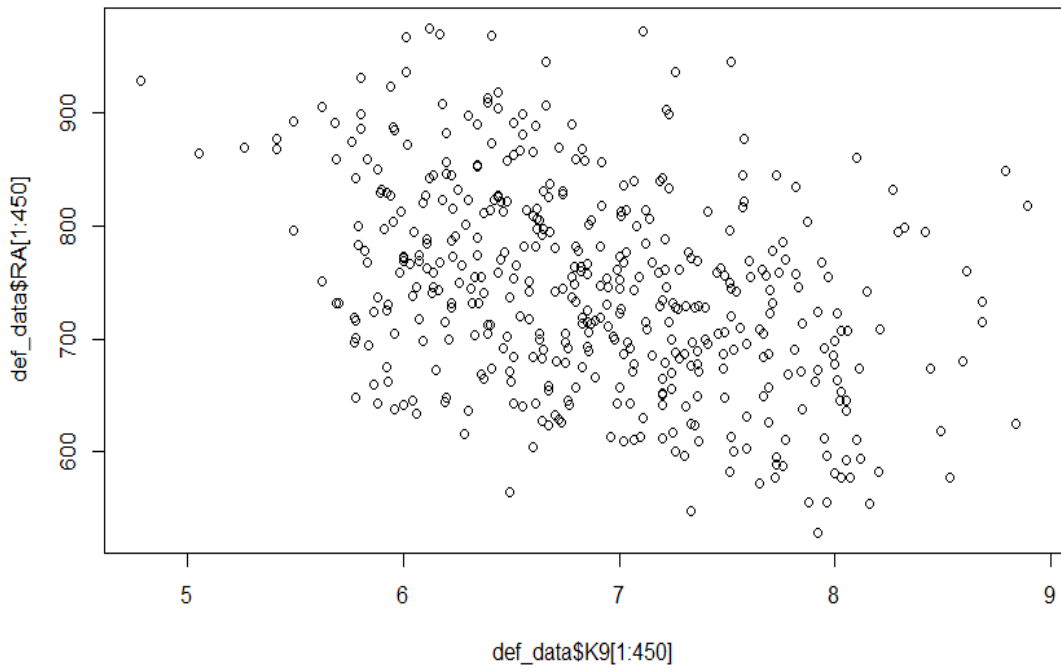
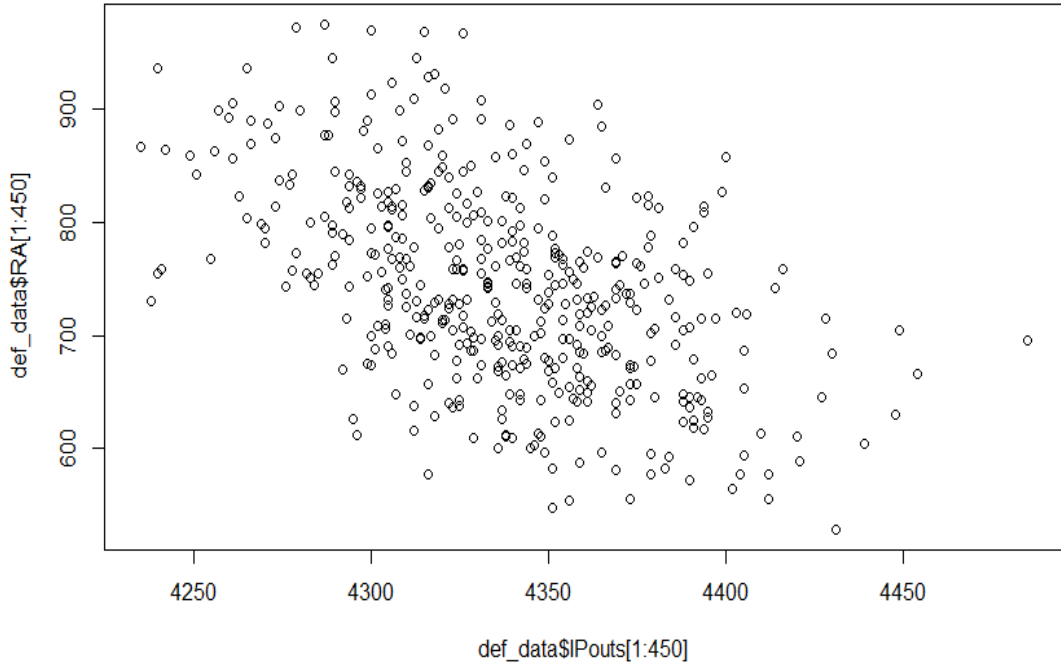


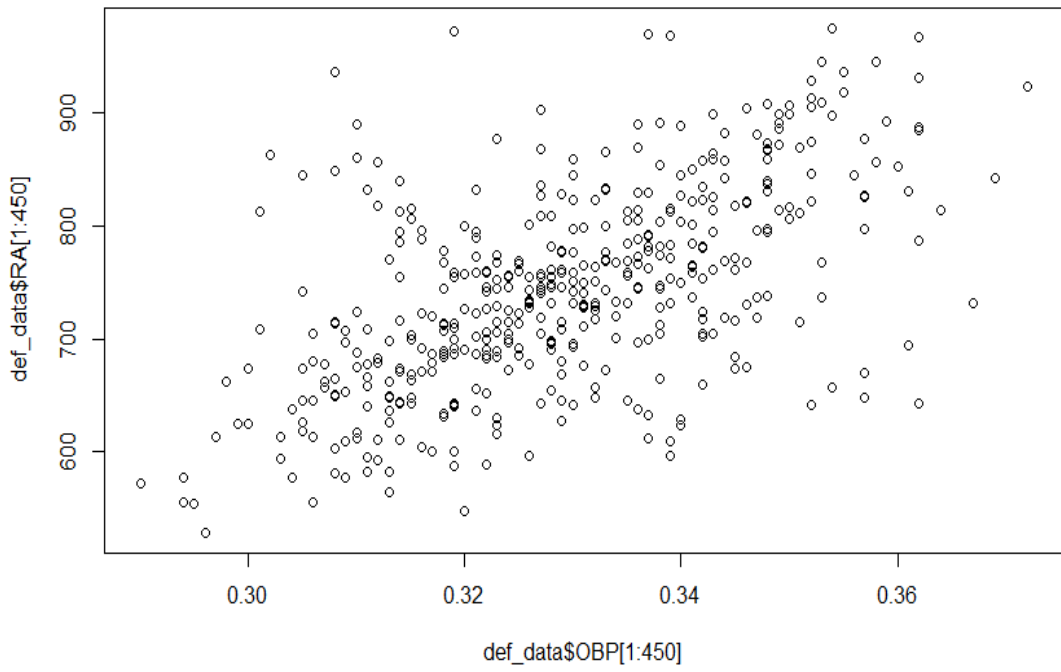
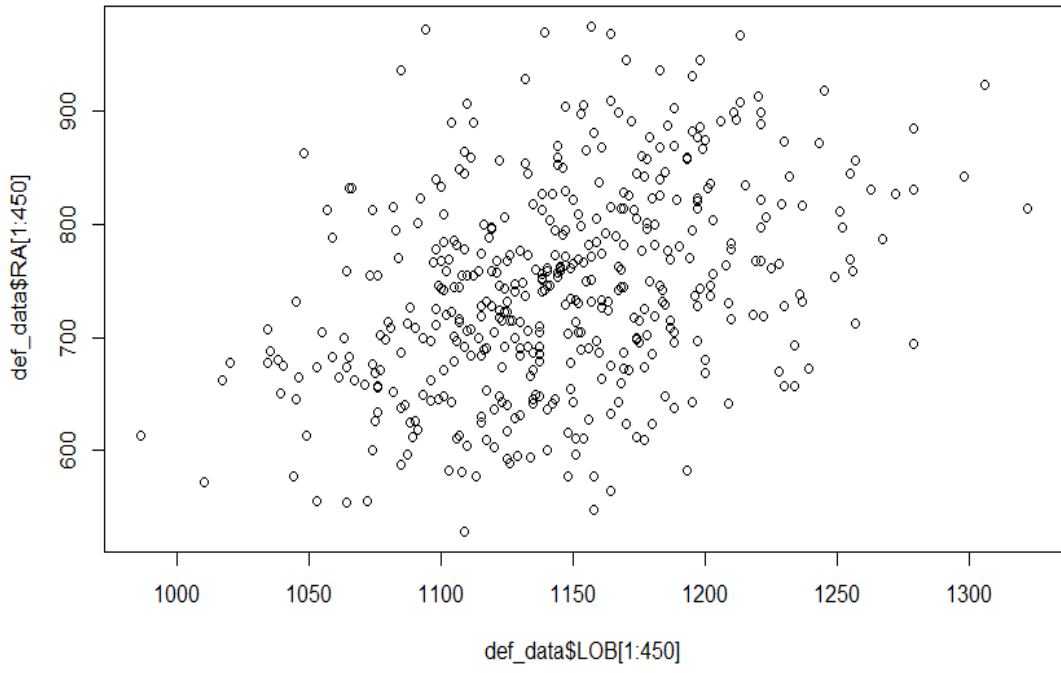


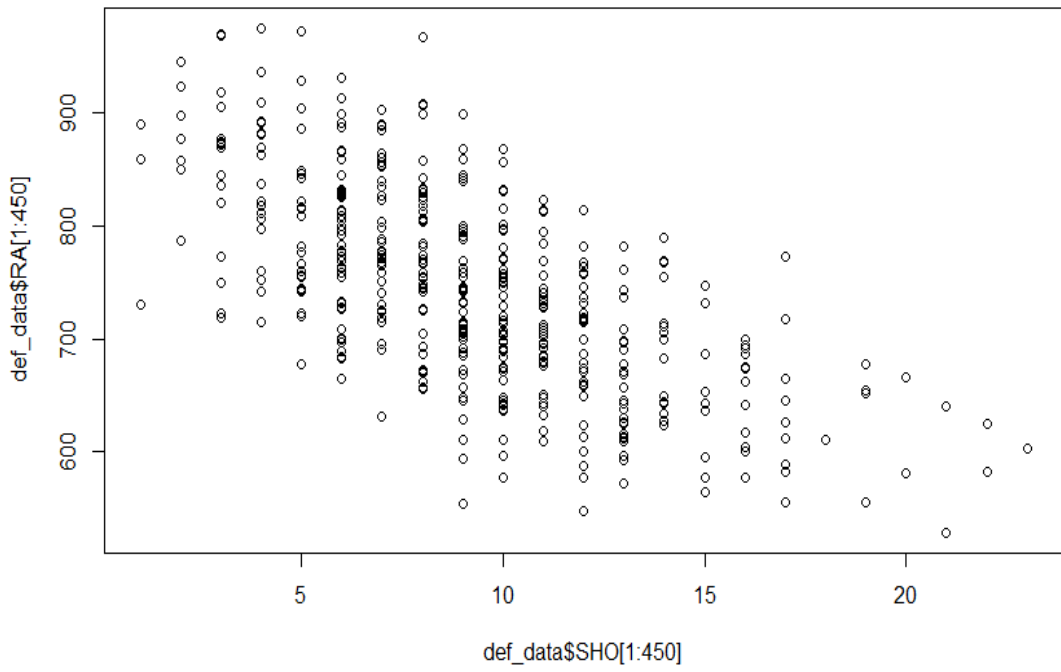
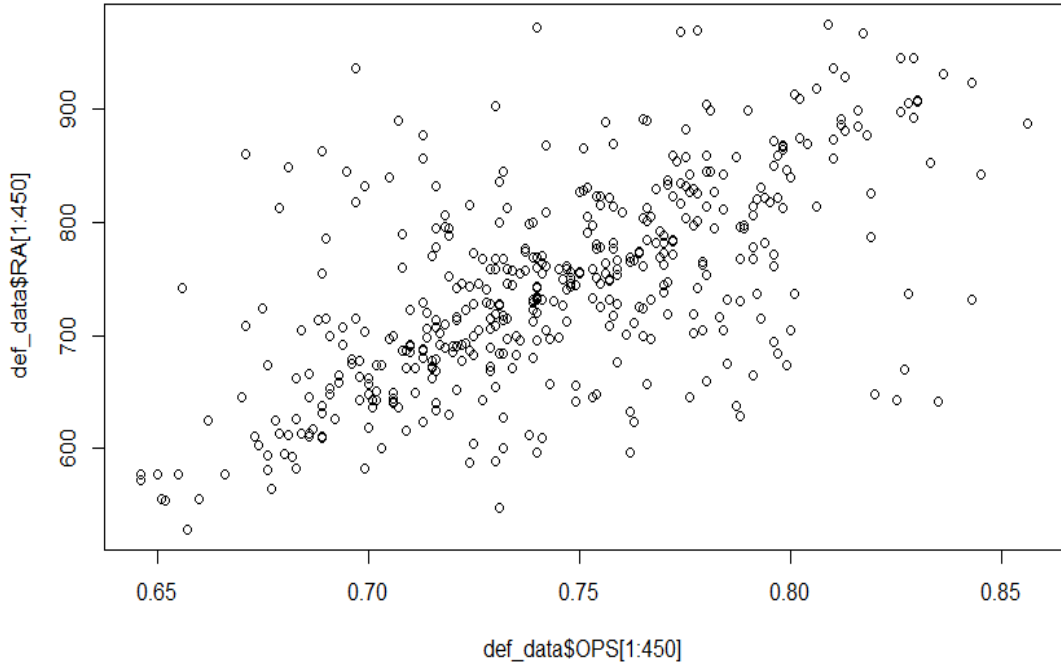


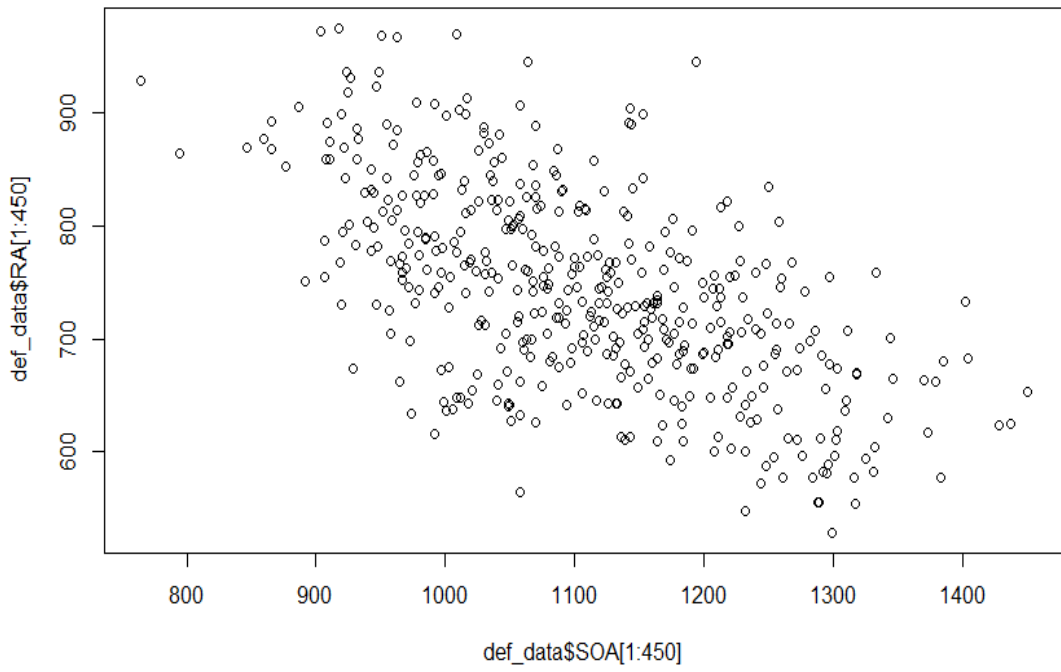
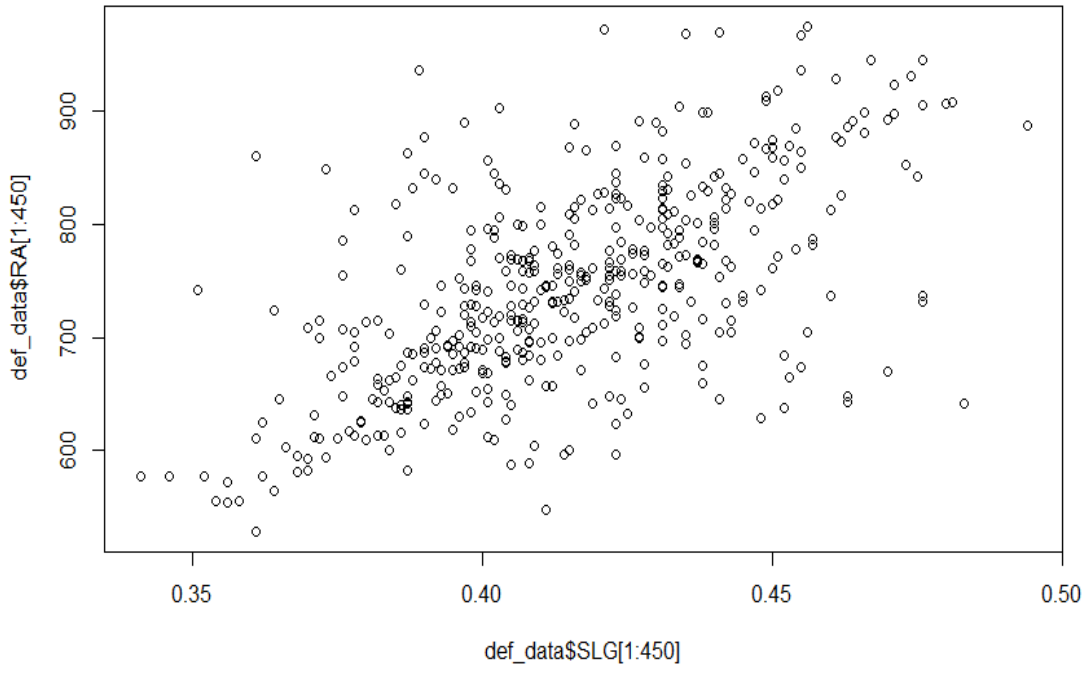


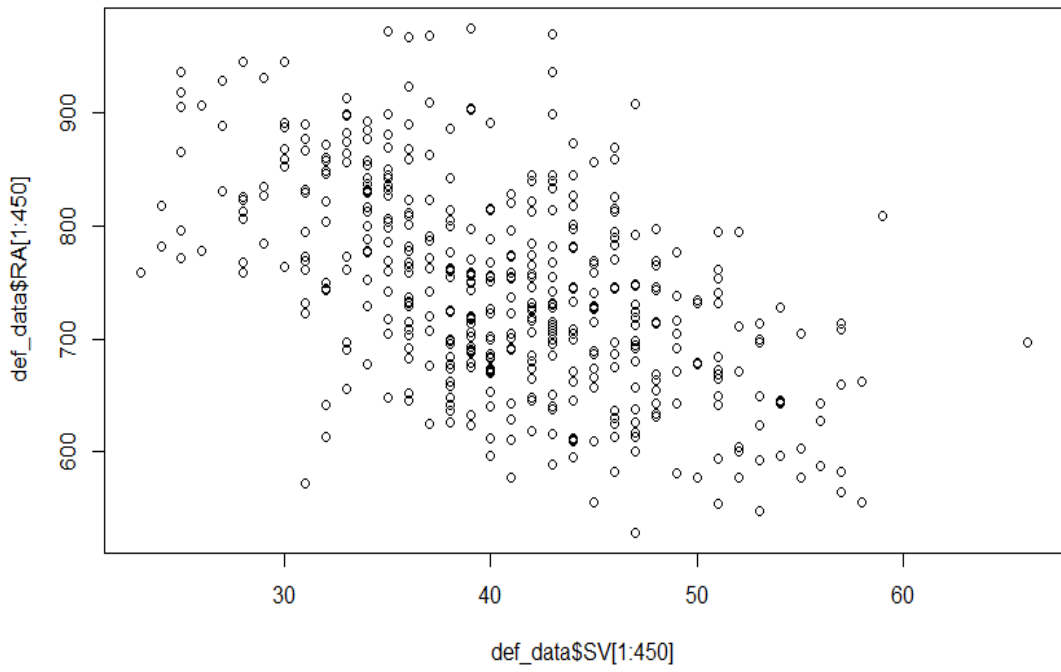
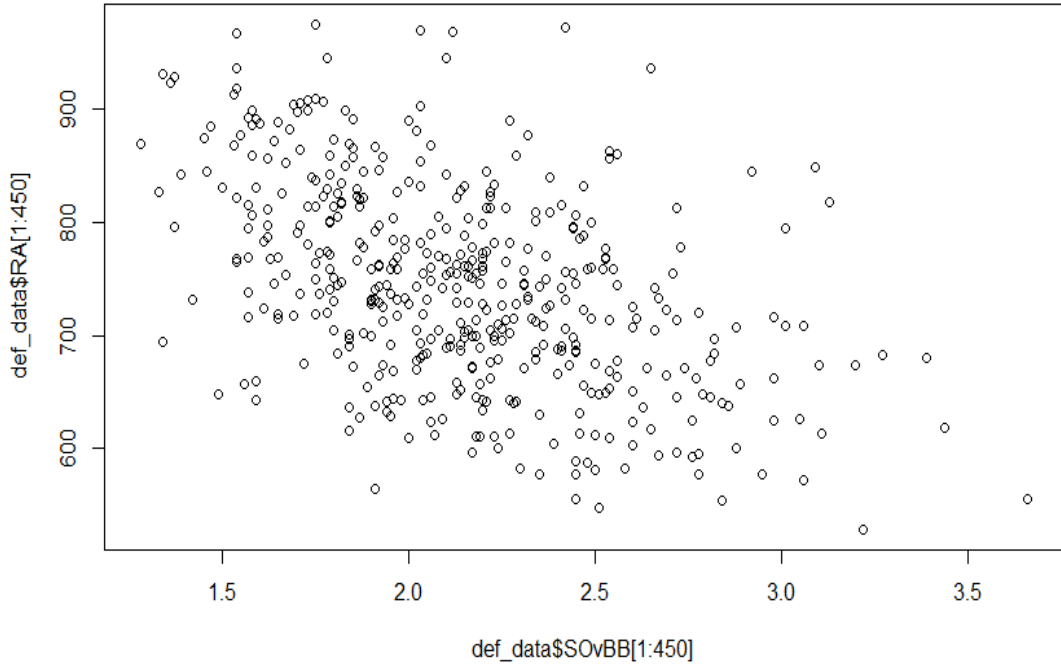


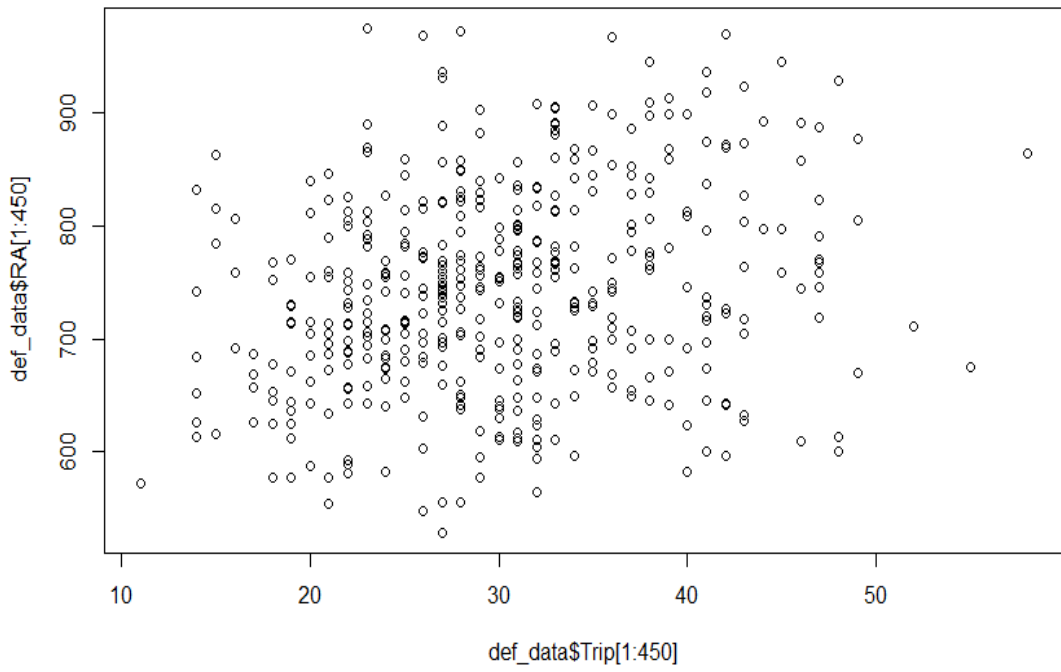
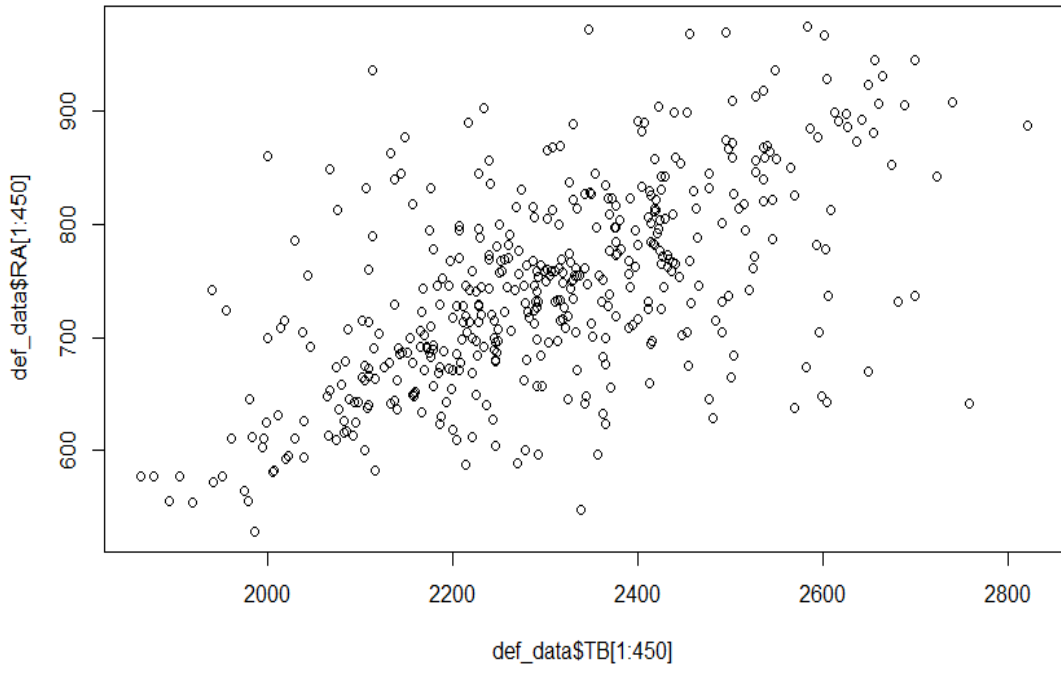




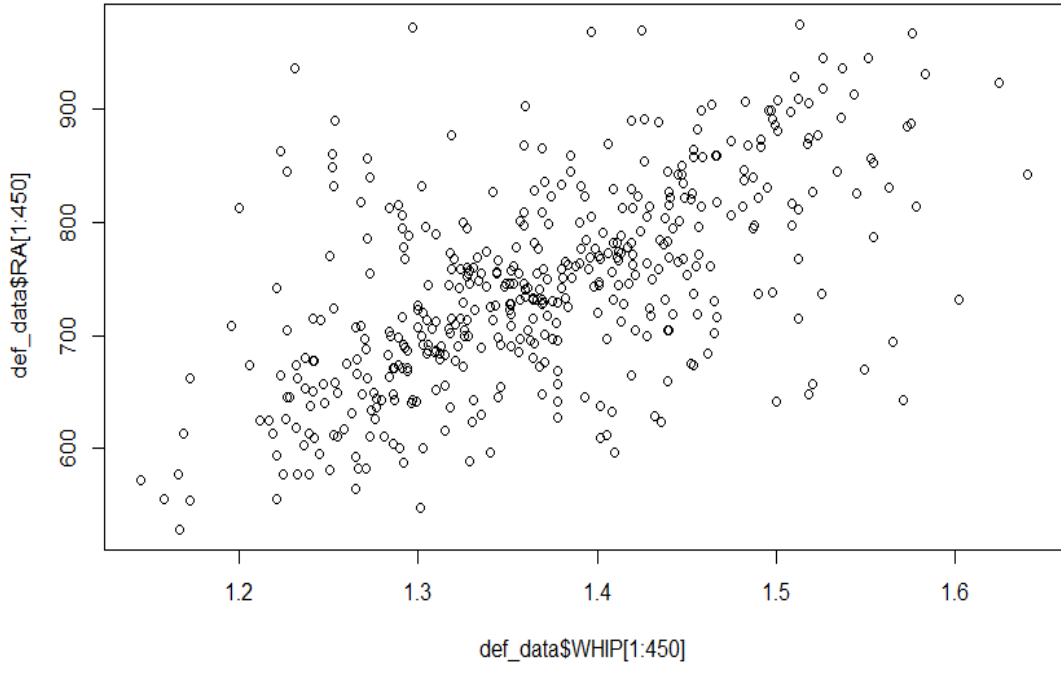




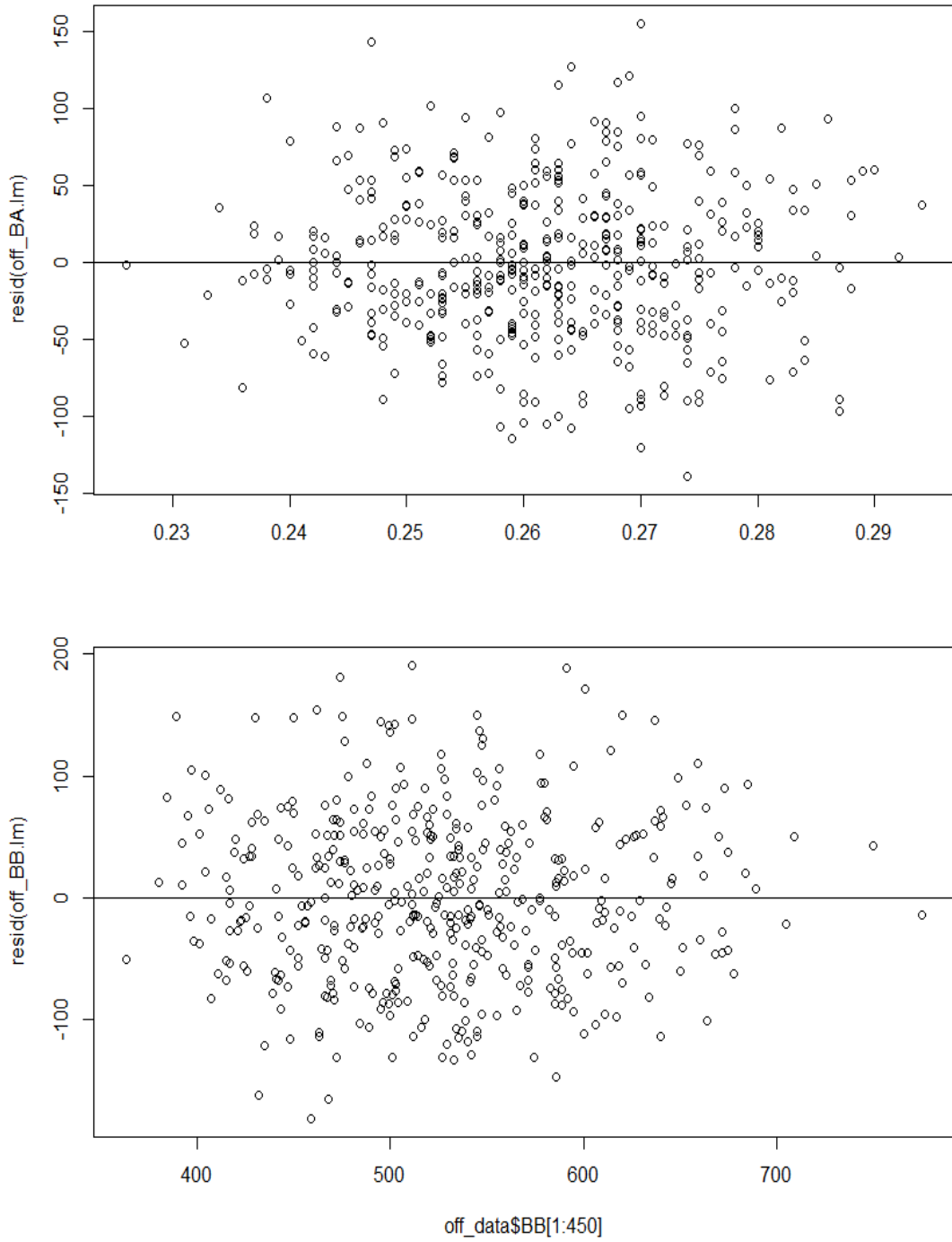


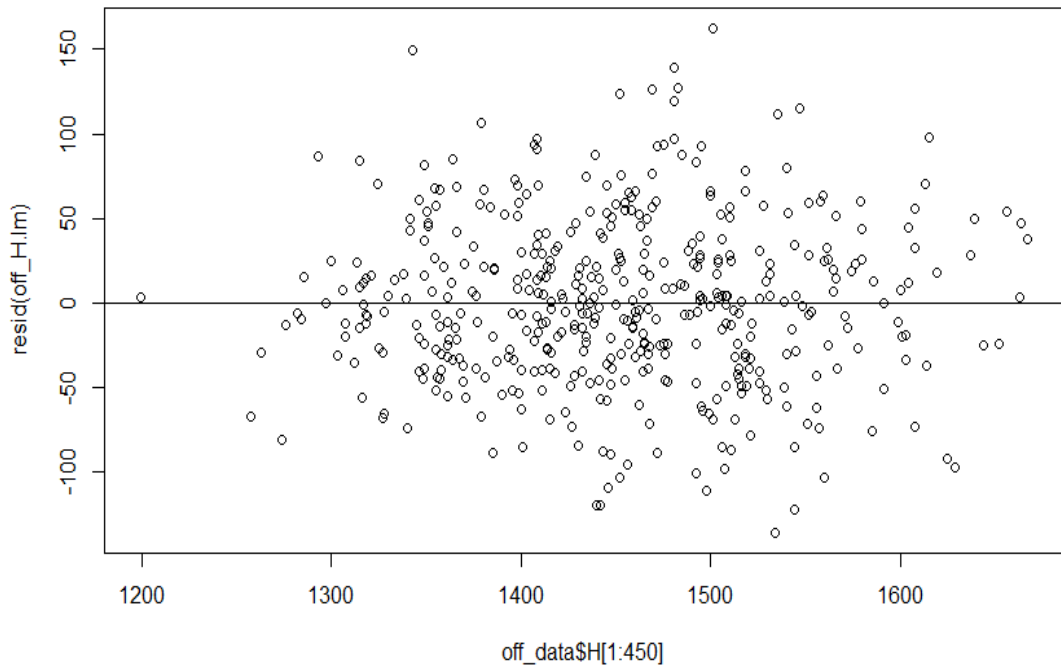
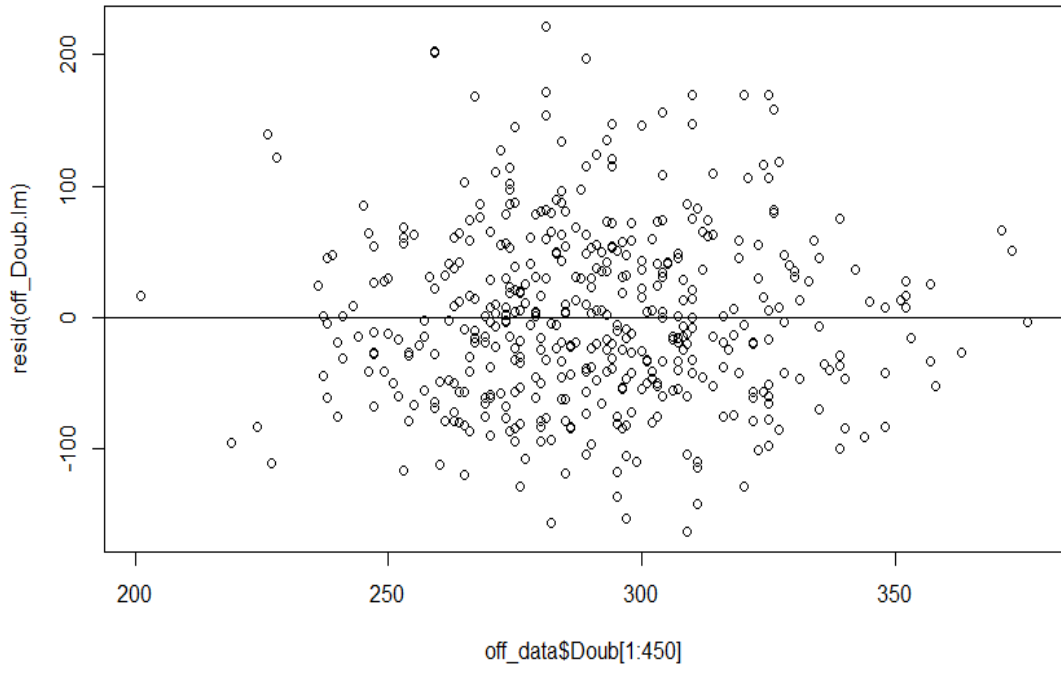


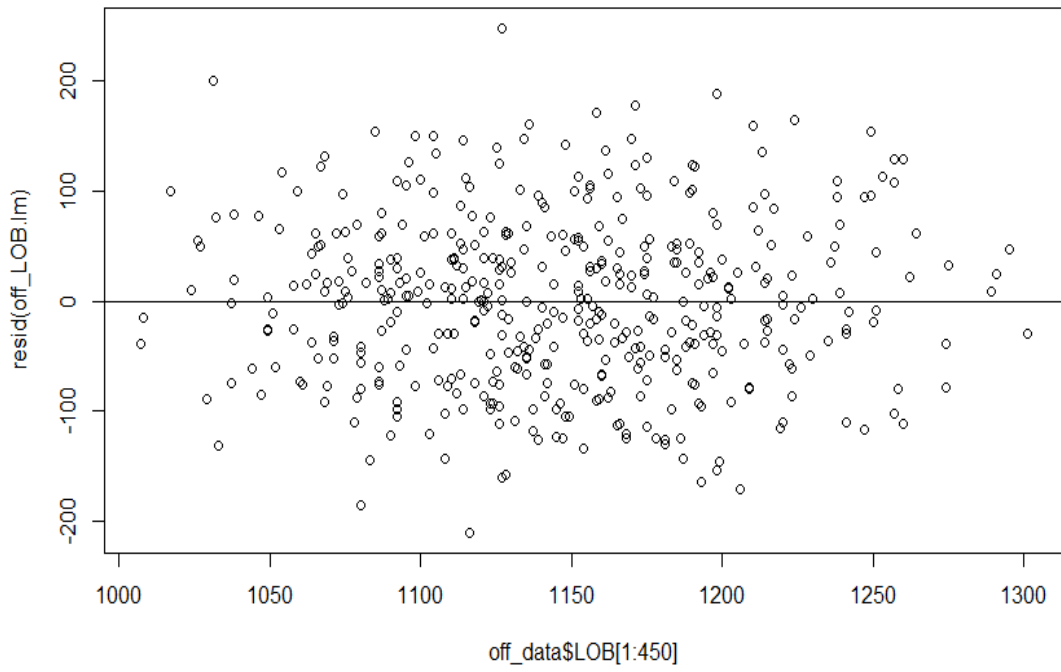
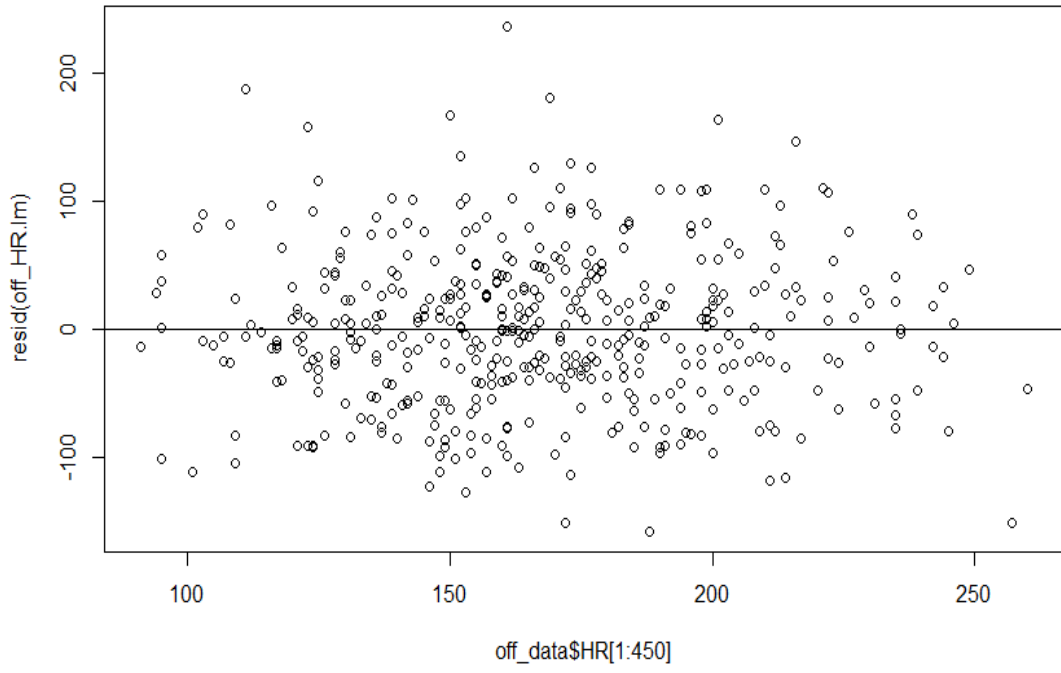


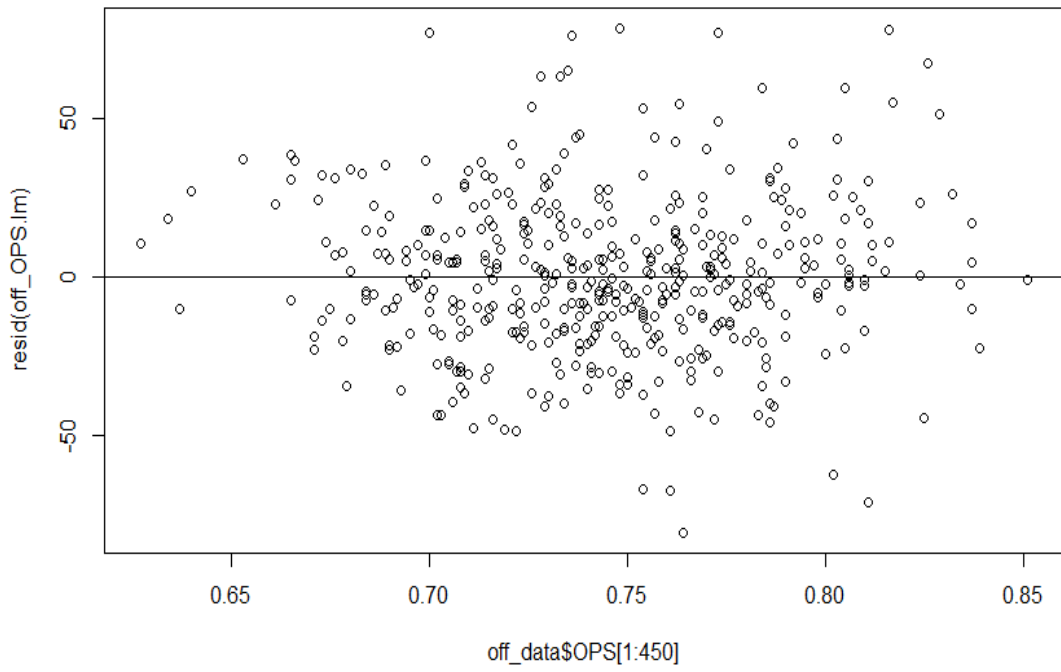
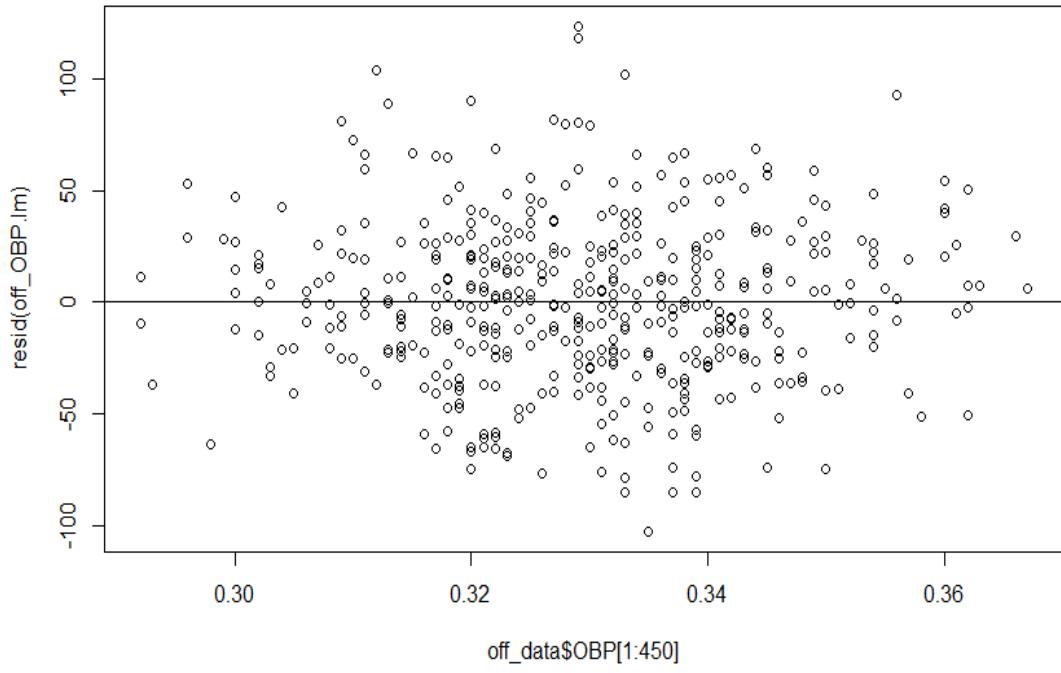


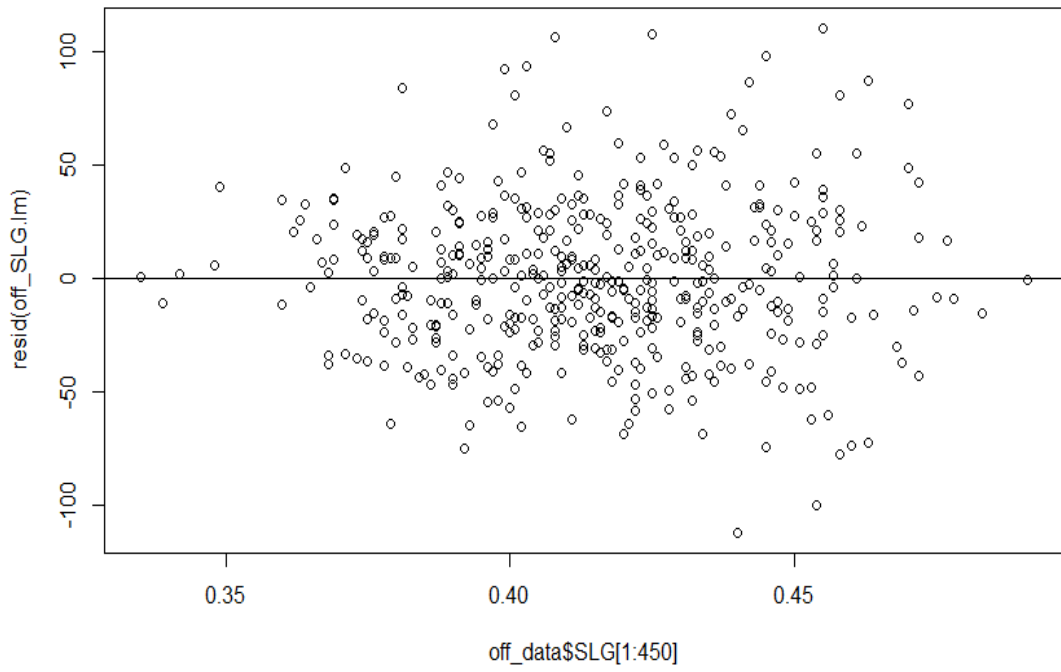
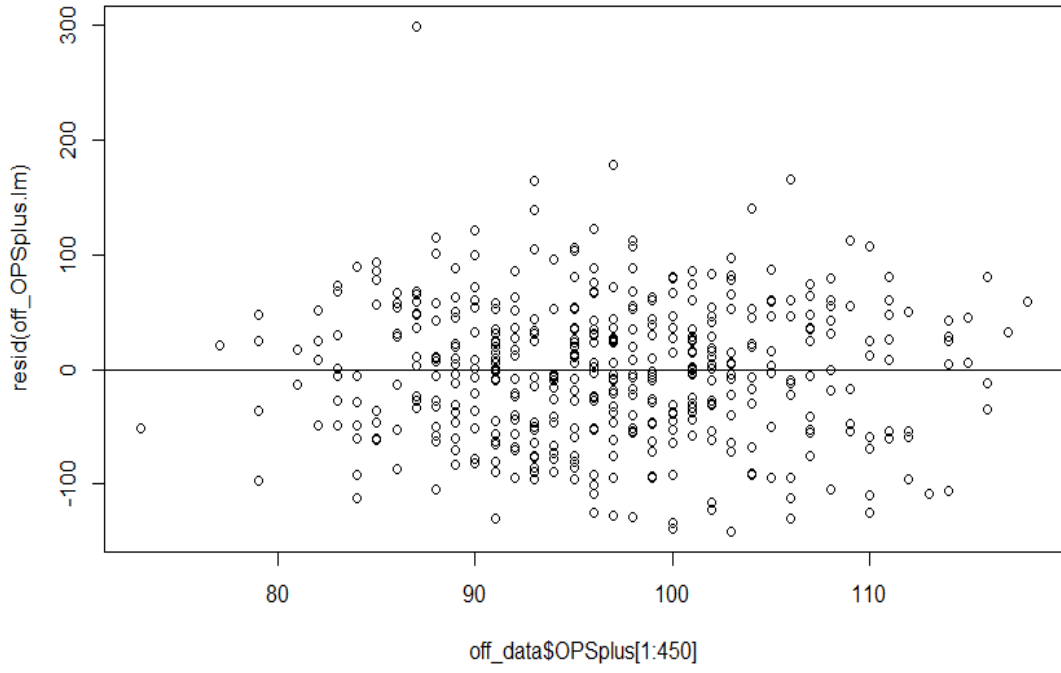
### C. Residual Plots for Offensive Variables Correlated with Runs

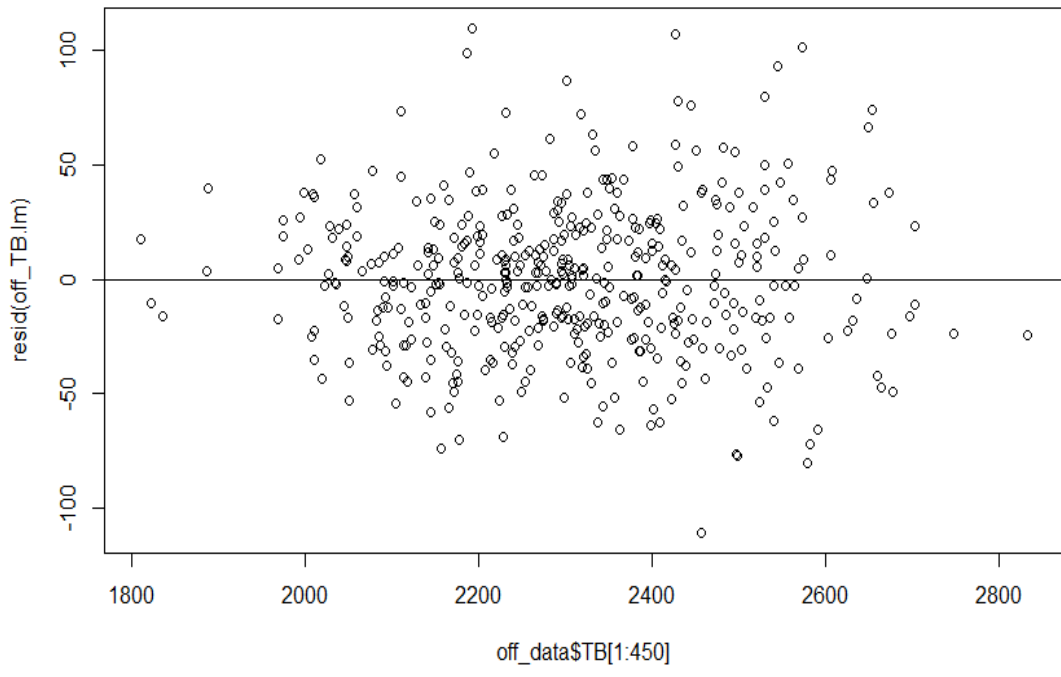
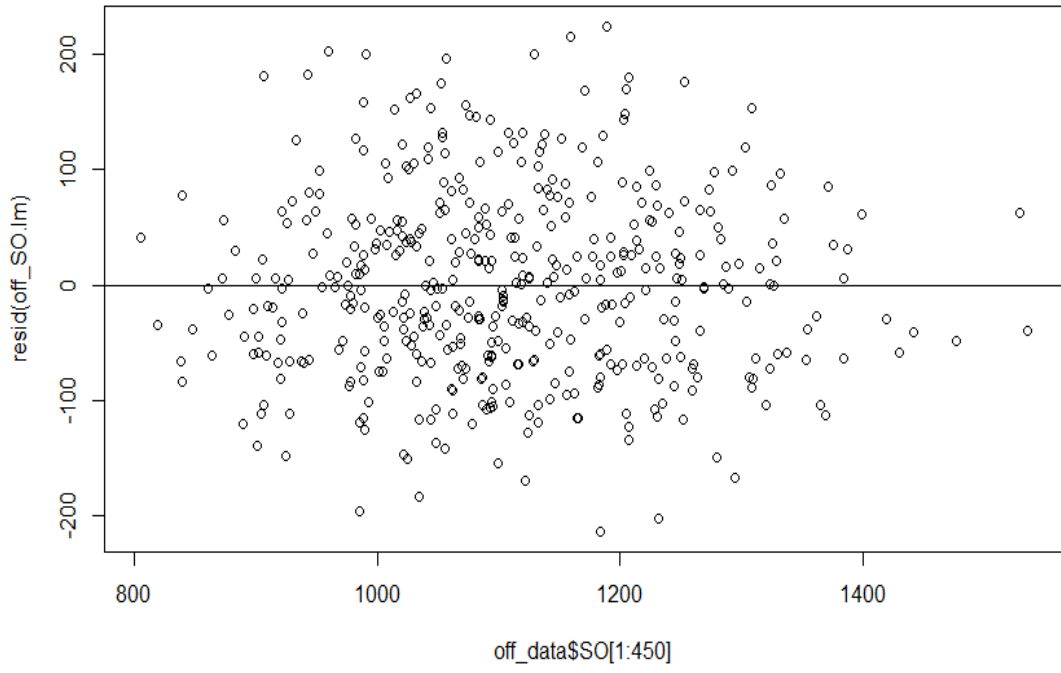




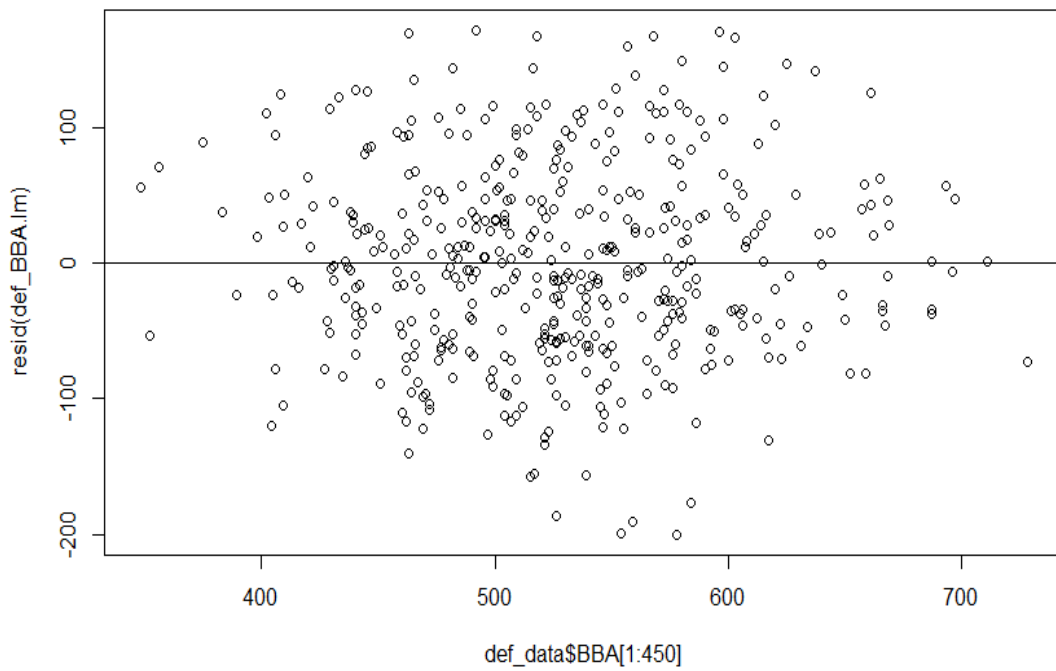
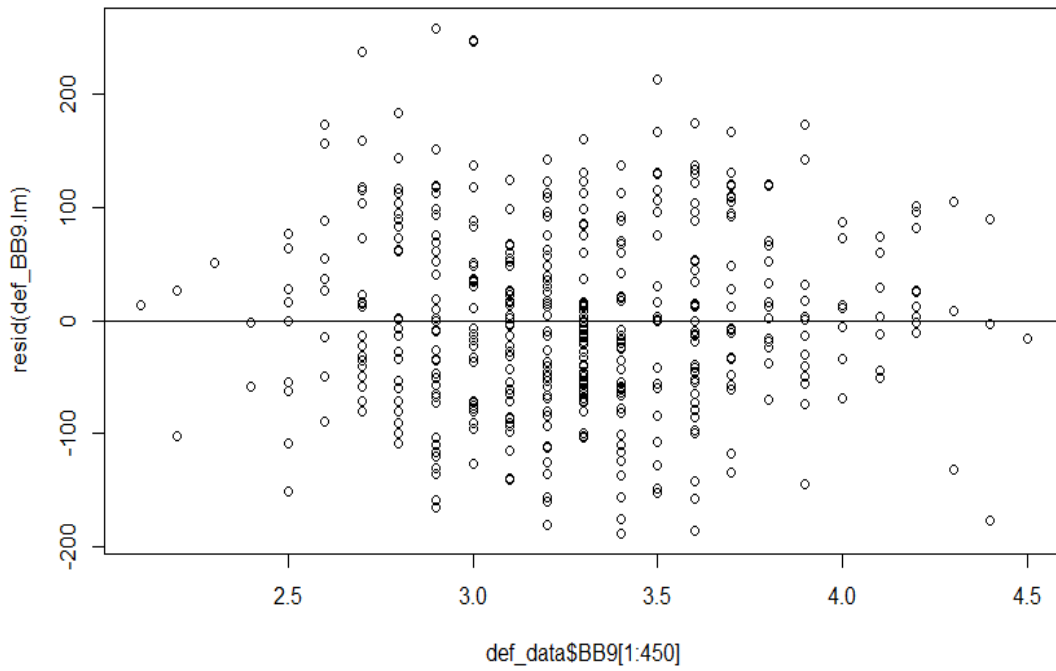




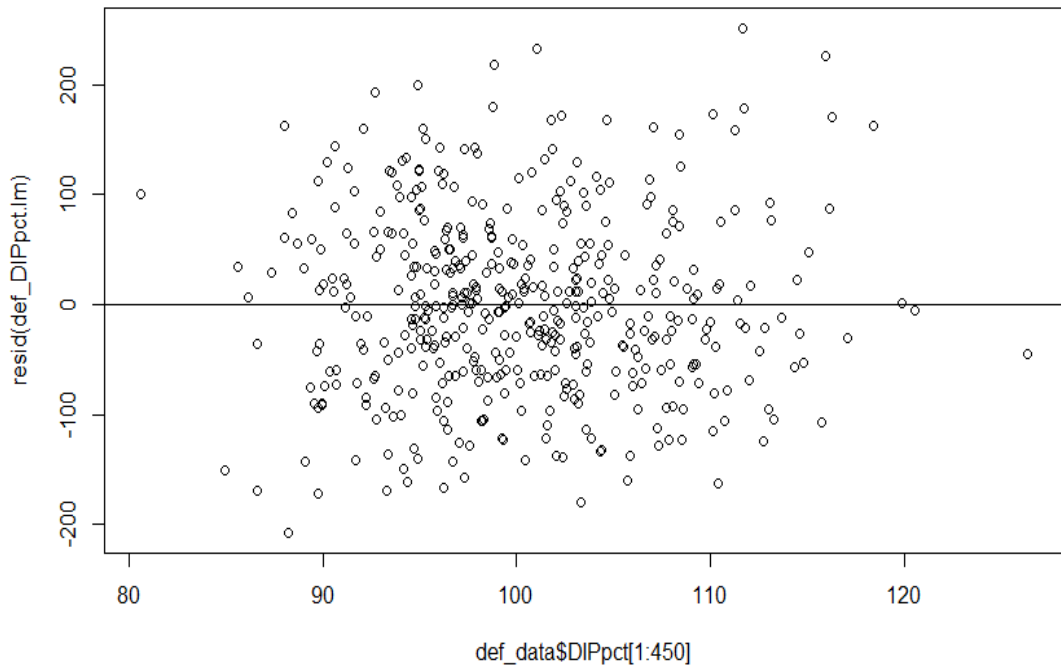
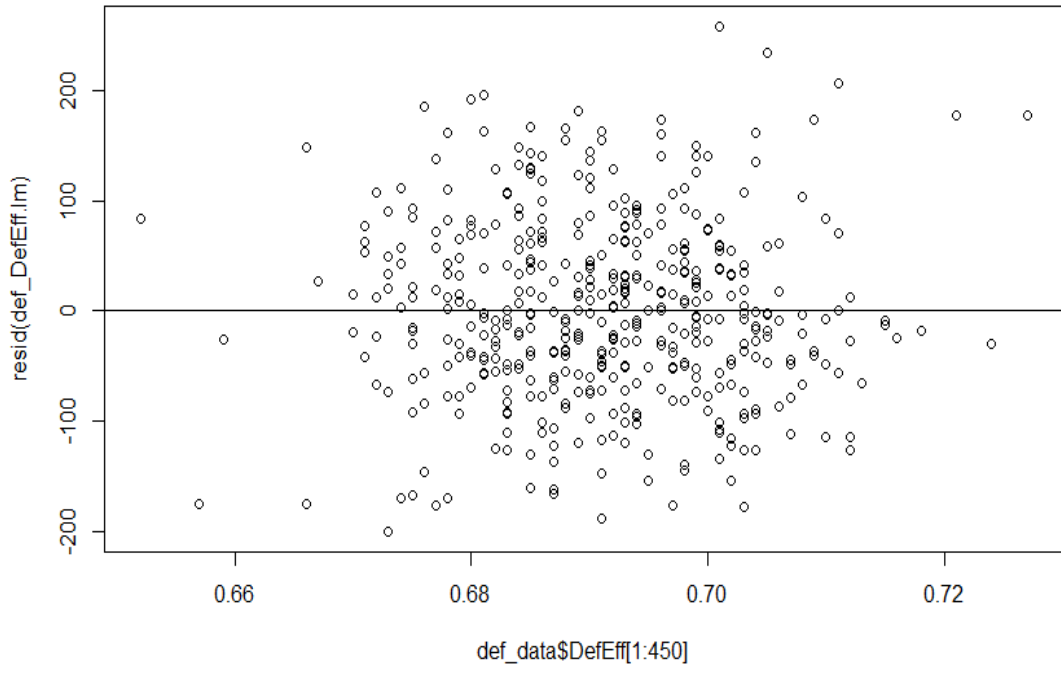


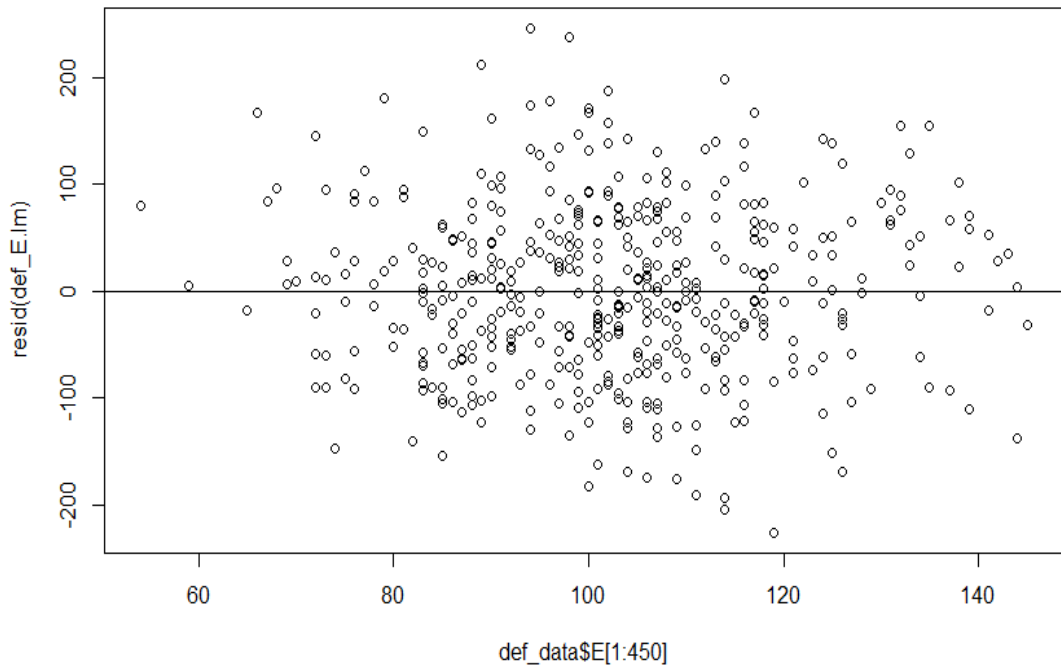
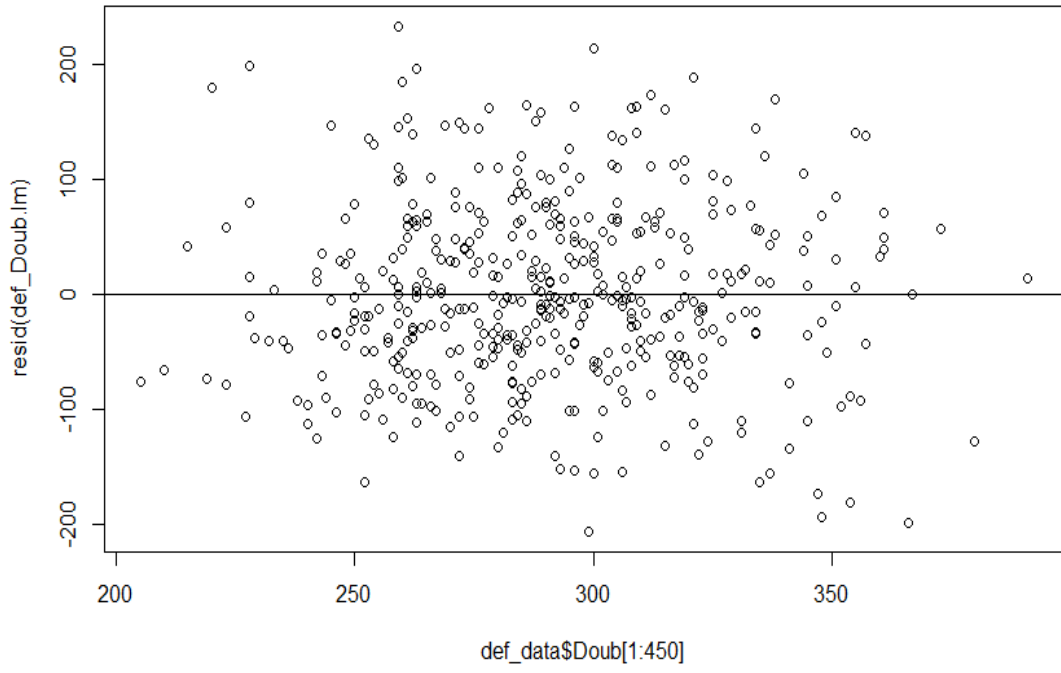


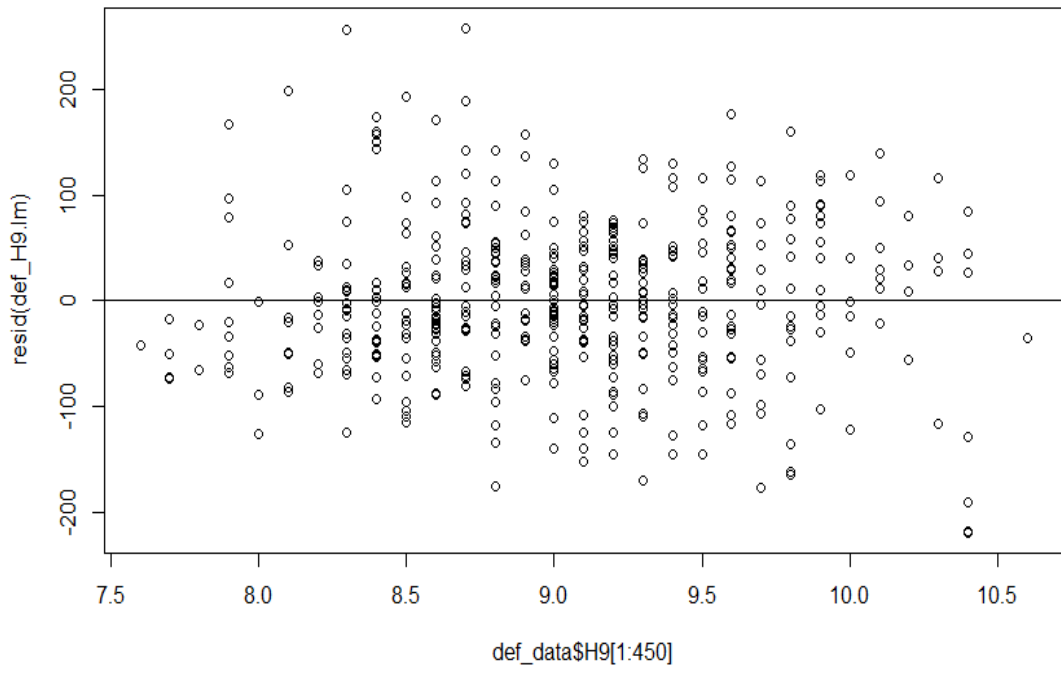
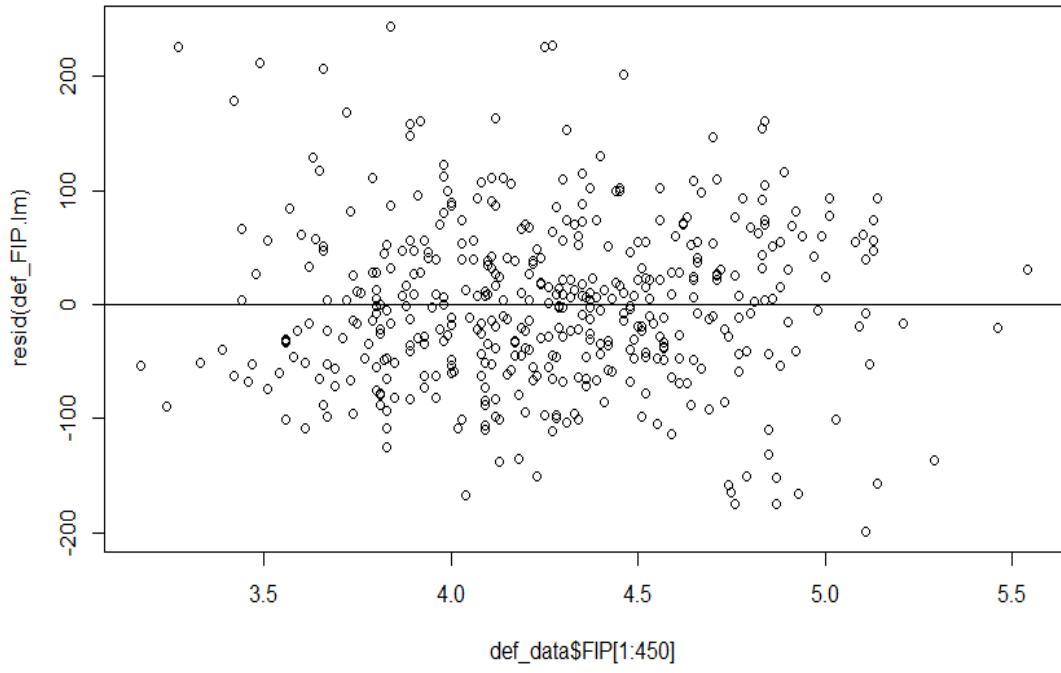
D. Residual Plots for Defensive Variables Correlated with Runs Allowed

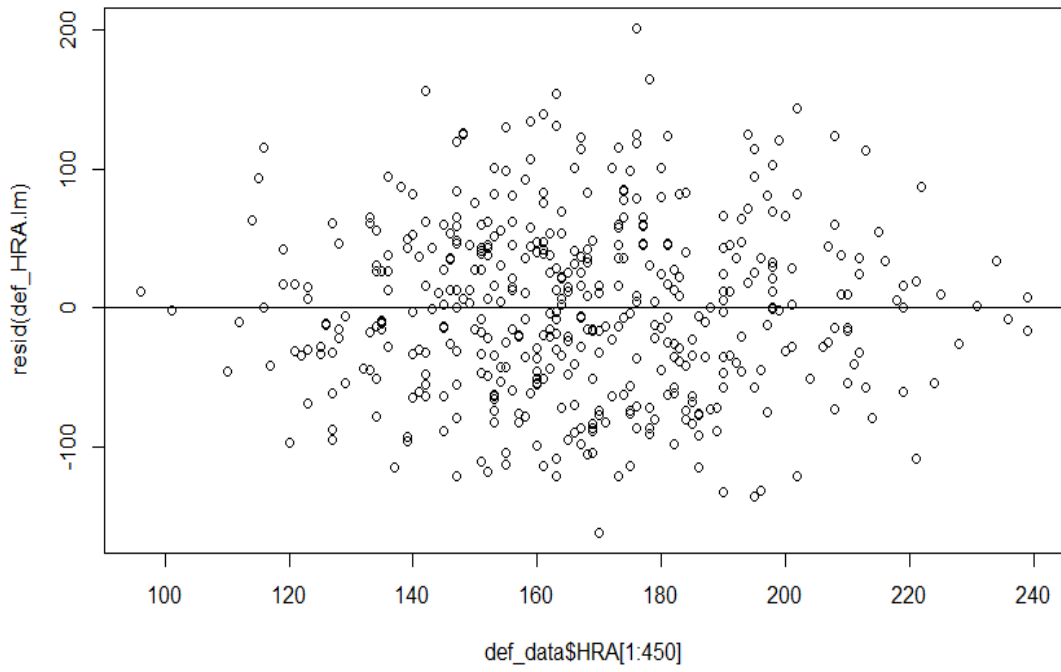
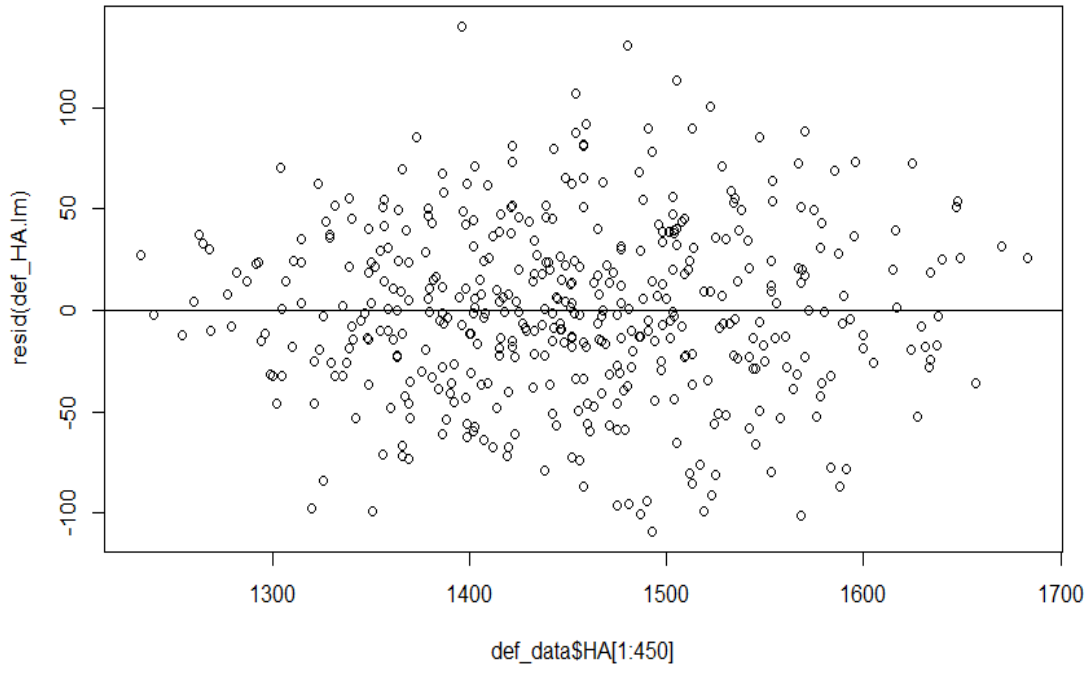


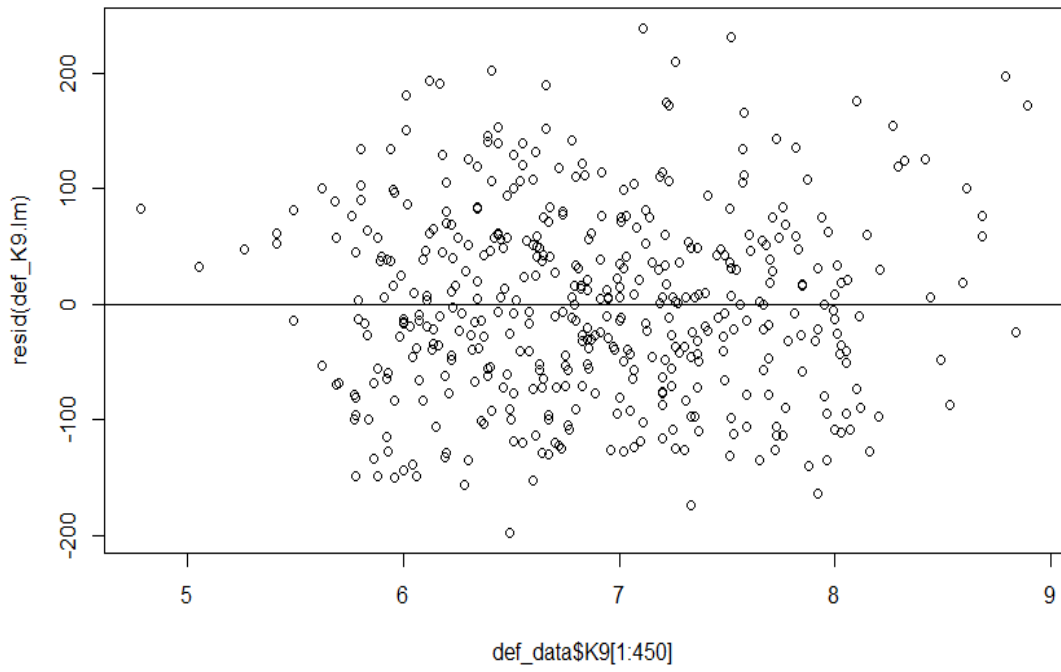
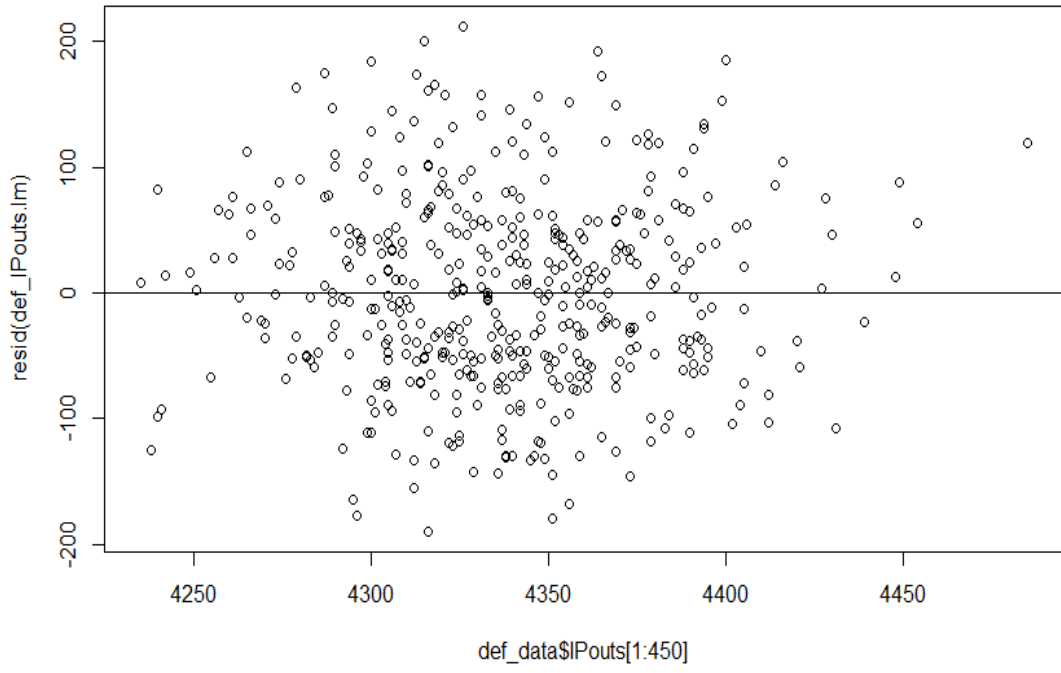


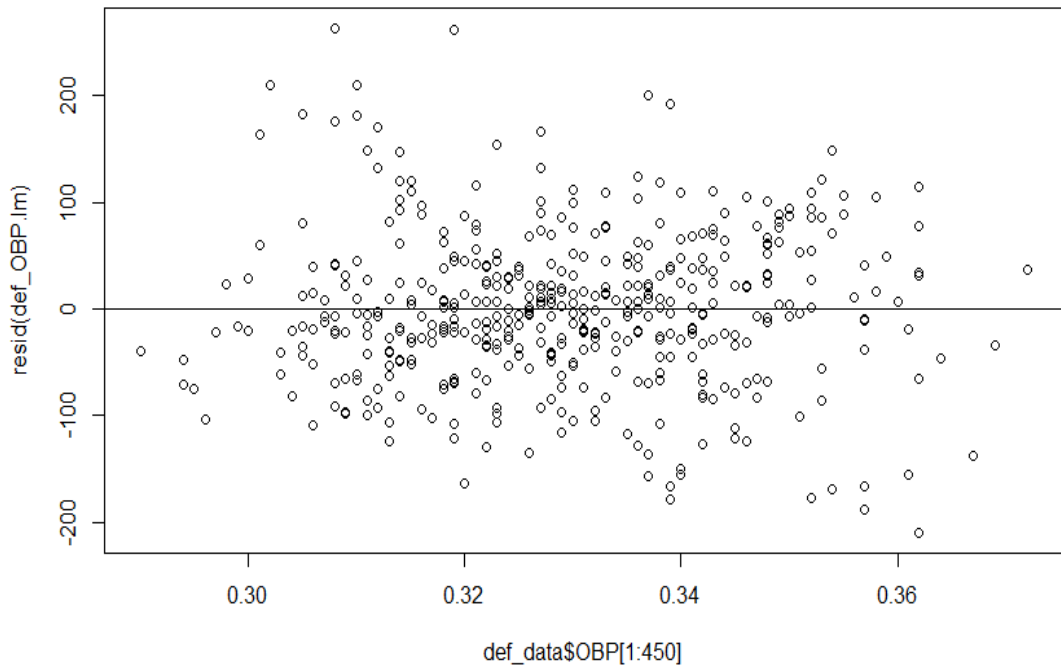
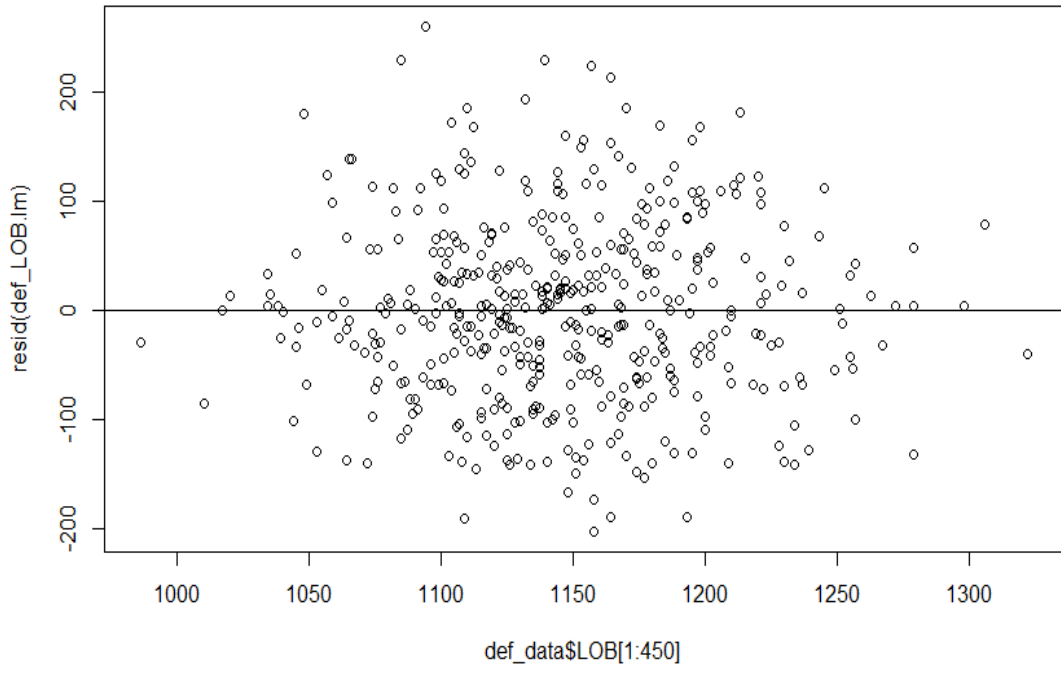


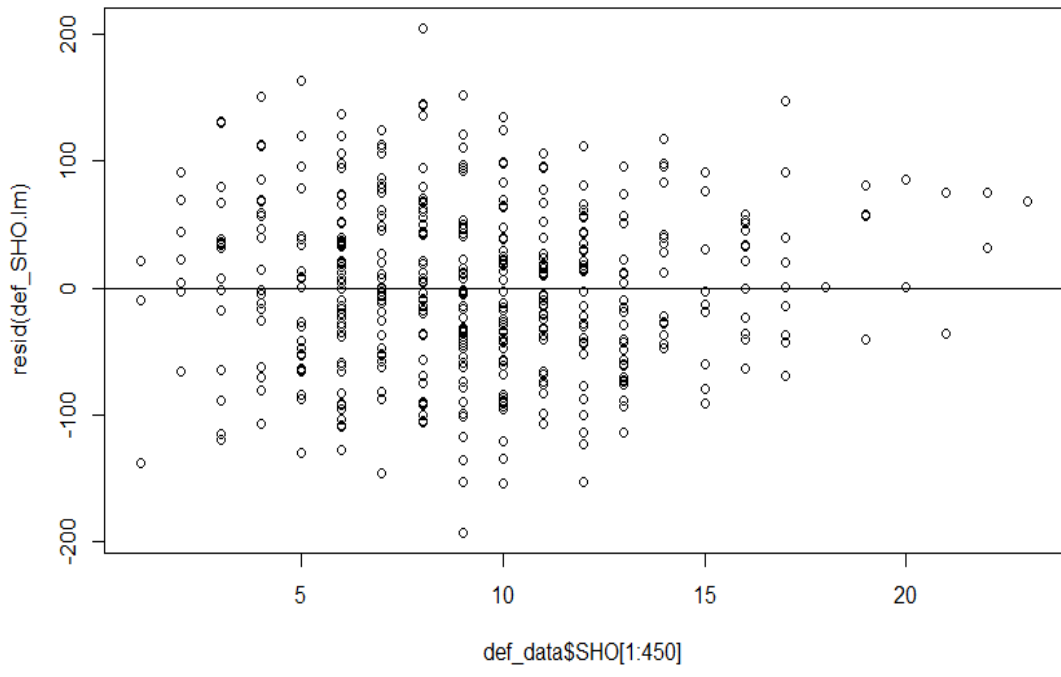
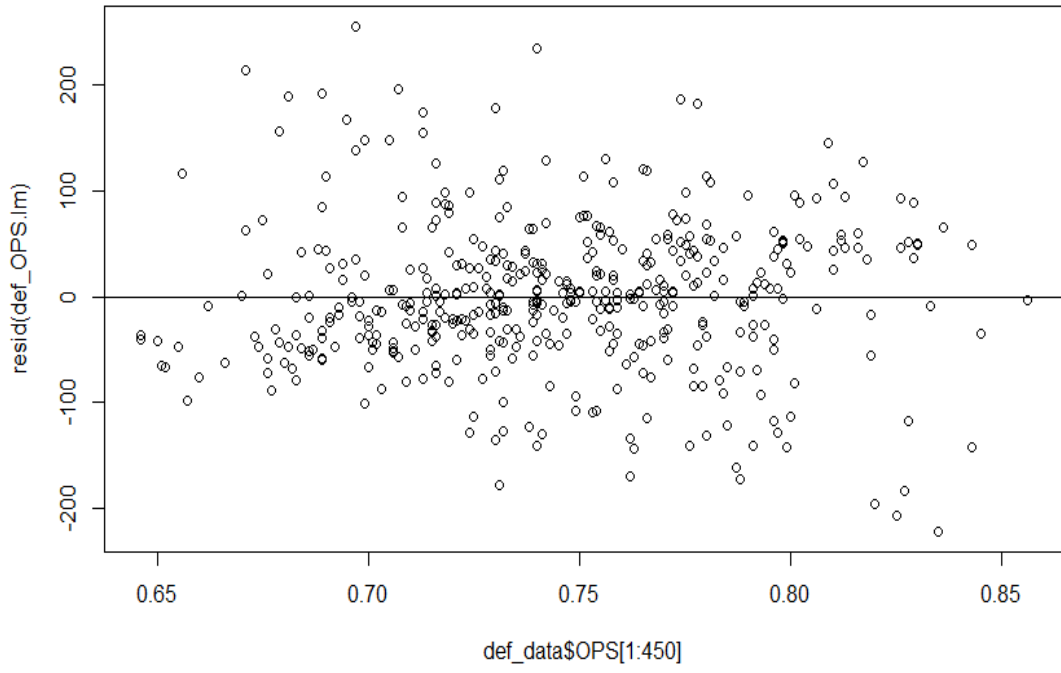


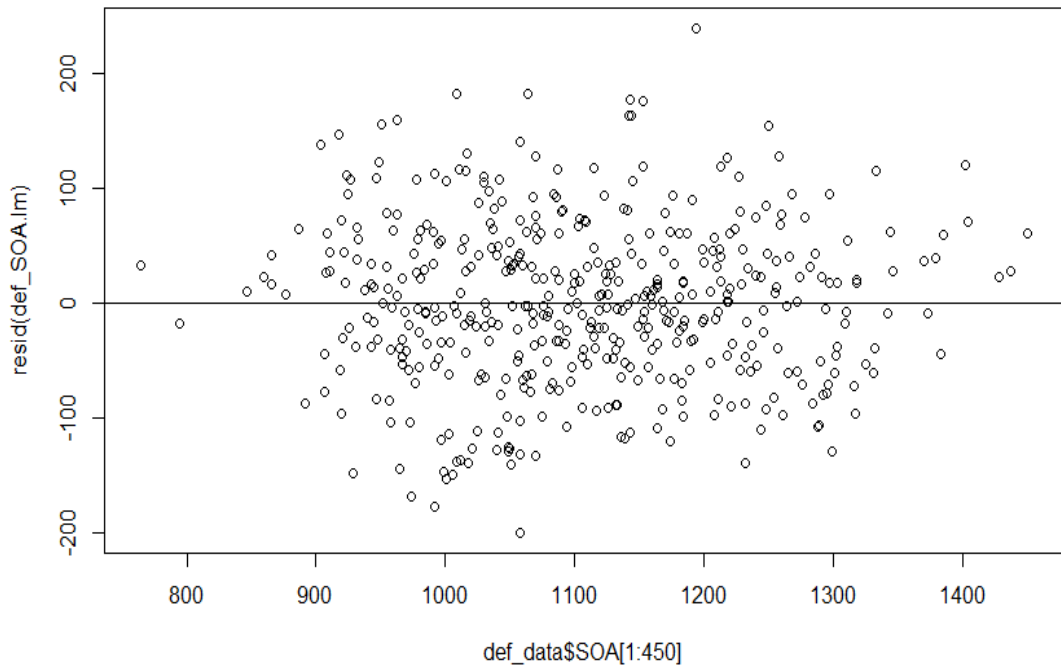
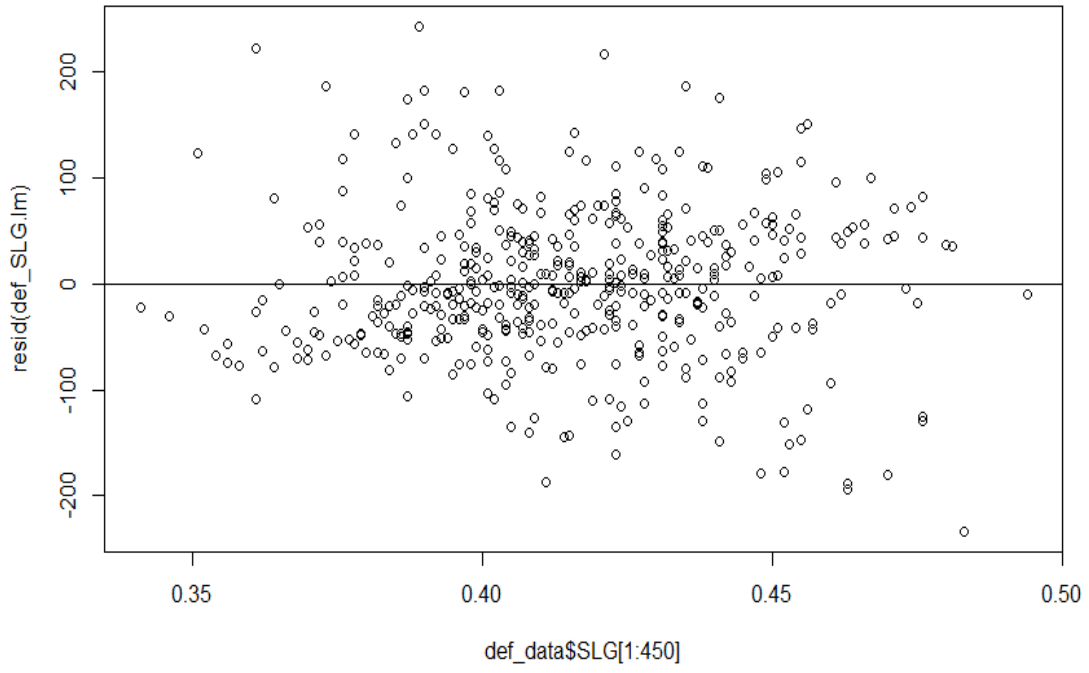




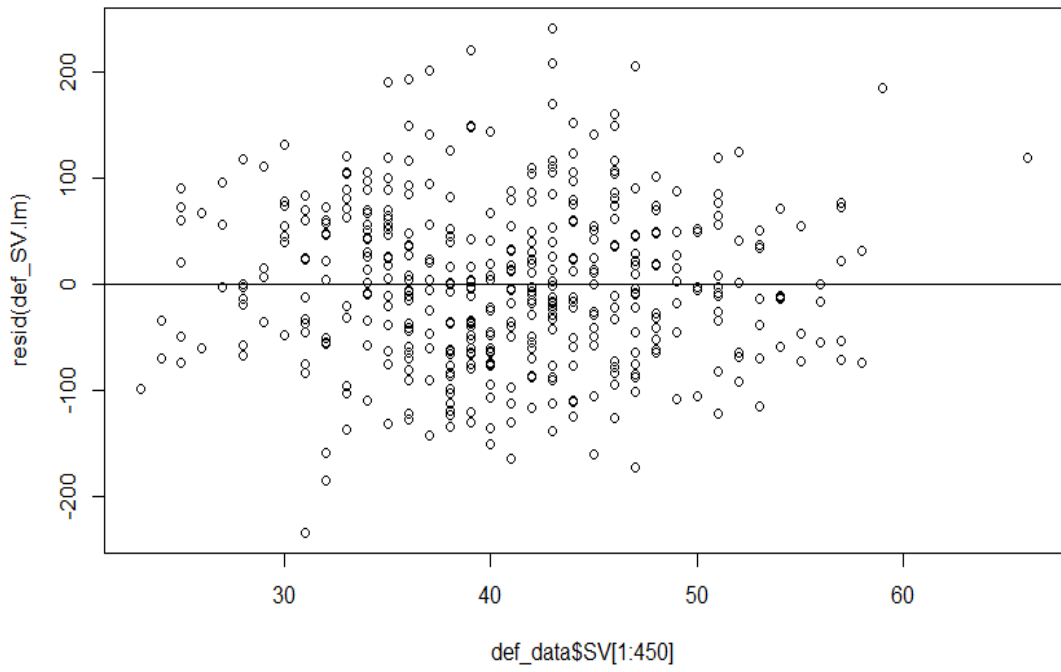
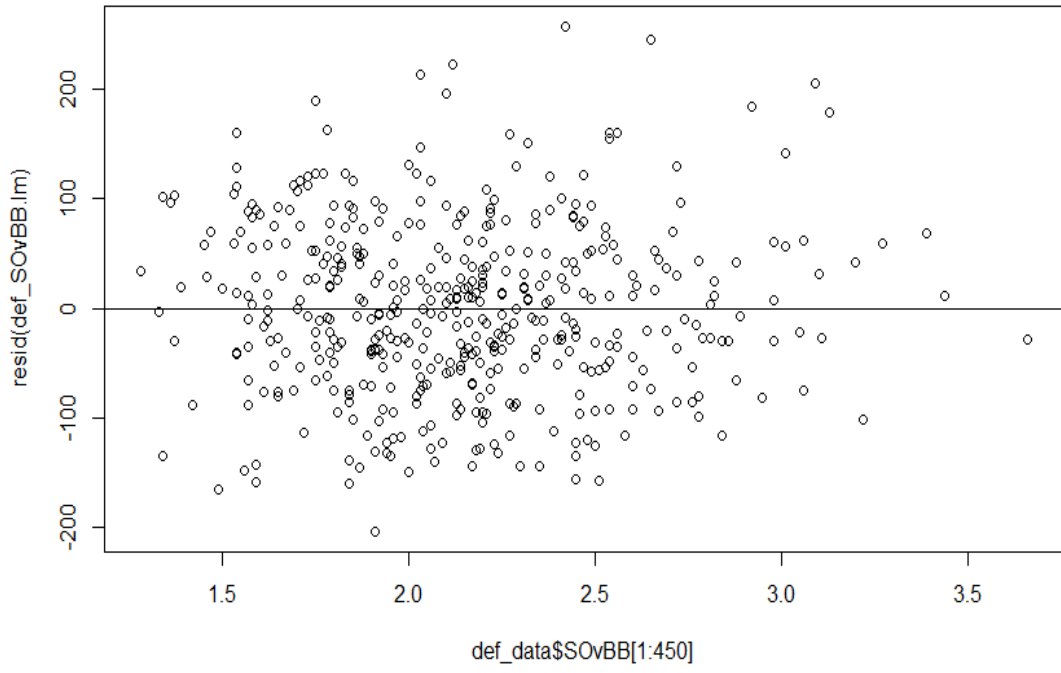


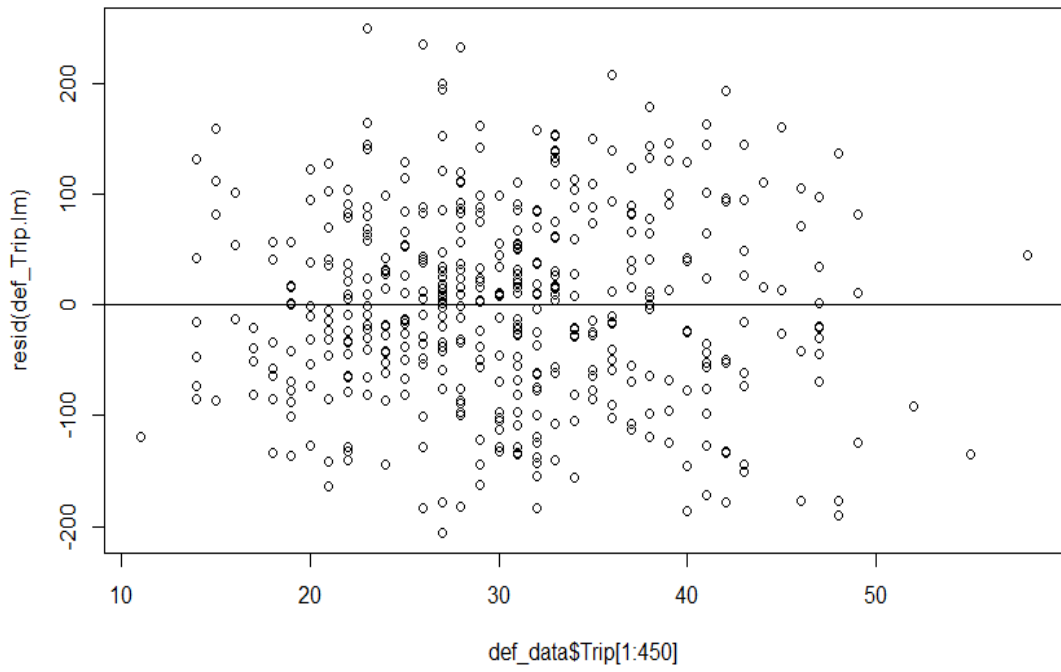
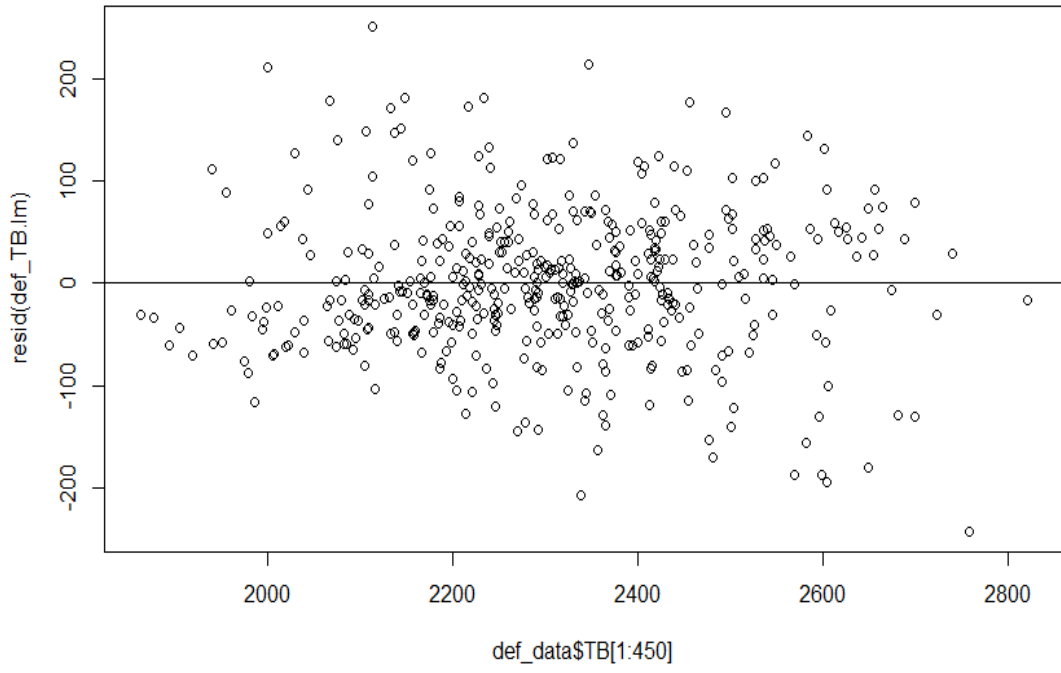


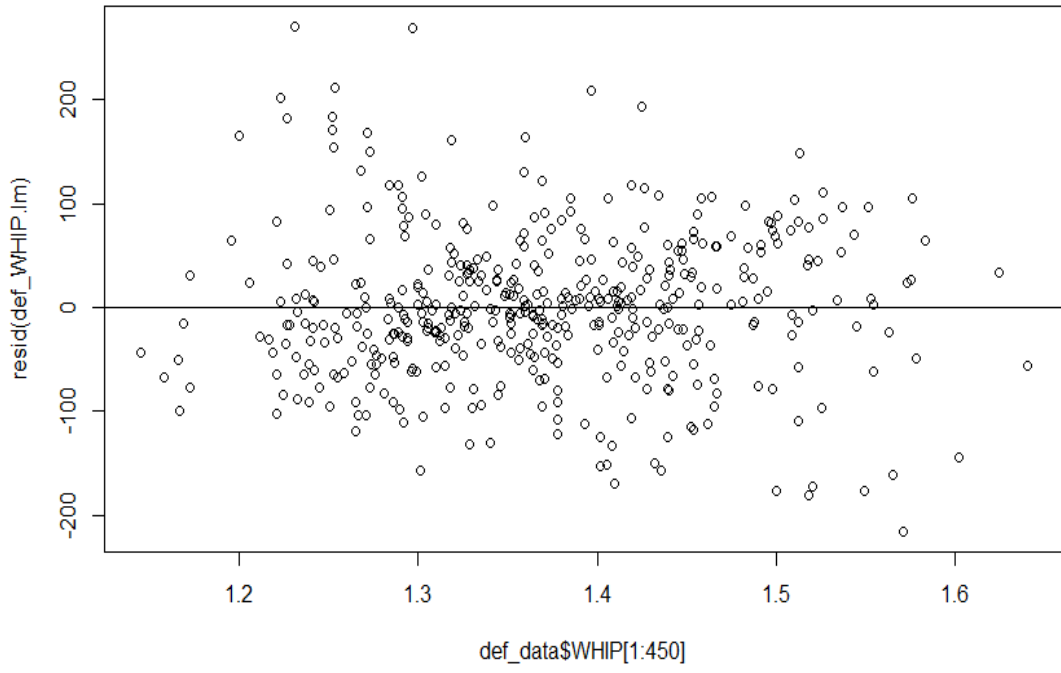












## E. Cross Validation Results Table

	Year	Team	Rmodel	Ractual	OffPctErrc	RAModel	RAActual	DefPctErrc	WPct	PythActua	PythMode	PythPctEn	WPctError
1	2015	ARI	722.1165	720	0.00294	759.7261	760	-0.00036	0.487654	0.475284	0.474636	-0.00136	-0.0267
2	2015	ATL	607.0273	573	0.059384	701.5648	693	0.012359	0.41358	0.413876	0.428131	0.034443	0.035184
3	2015	BAL	711.202	713	-0.00252	754.9621	753	0.002606	0.5	0.475049	0.47018	-0.01025	-0.05964
4	2015	BOS	730.4065	748	-0.02352	715.4762	701	0.020651	0.481481	0.529655	0.510325	-0.03649	0.059906
5	2015	CHC	676.4154	689	-0.01826	579.5049	608	-0.04687	0.598765	0.556969	0.576706	0.035436	-0.03684
6	2015	CHW	622.8234	622	0.001324	741.253	754	-0.01691	0.469136	0.412852	0.413829	0.002367	-0.11789
7	2015	CIN	655.4002	640	0.024063	604.6711	640	-0.0552	0.395062	0.5	0.540194	0.080388	0.367365
8	2015	CLE	705.6015	669	0.054711	888.5234	844	0.052753	0.5	0.395265	0.386744	-0.02156	-0.22651
9	2015	COL	739.0459	737	0.002776	769.2941	803	-0.04197	0.419753	0.460842	0.479954	0.041472	0.14342
10	2015	DET	743.1568	689	0.078602	622.3458	618	0.007032	0.45679	0.549591	0.587786	0.069498	0.286776
11	2015	HOU	749.4111	729	0.027999	671.7616	675	-0.0048	0.530864	0.535152	0.554475	0.036108	0.044476
12	2015	KCR	705.4283	724	-0.02565	568.0358	595	-0.04532	0.58642	0.588823	0.606647	0.030272	0.034493
13	2015	LAA	663.959	661	0.004477	731.9467	713	0.026573	0.524691	0.46541	0.45141	-0.03008	-0.13967
14	2015	LAD	726.1056	667	0.088614	687.4571	678	0.013948	0.567901	0.492517	0.527321	0.070665	-0.07146
15	2015	MIA	636.8053	613	0.038834	668.2353	641	0.042489	0.438272	0.479577	0.47593	-0.0076	0.085926
16	2015	MIL	655.6508	655	0.000994	754.3217	737	0.023503	0.419753	0.446245	0.43036	-0.0356	0.02527
17	2015	MIN	672.4526	696	-0.03383	718.2737	700	0.026105	0.512346	0.497378	0.467088	-0.0609	-0.08833
18	2015	NYM	667.9097	683	-0.02209	714.1233	698	0.023099	0.555556	0.490062	0.466599	-0.04788	-0.16012
19	2015	NYN	735.4324	764	-0.03739	596.1784	613	-0.02744	0.537037	0.599401	0.603444	0.006745	0.123655
20	2015	OAK	664.2441	694	-0.04288	725.0921	729	-0.00536	0.419753	0.477505	0.456287	-0.04444	0.087038
21	2015	PHI	607.6516	626	-0.02931	836.8753	809	0.034456	0.388889	0.384782	0.345213	-0.10283	-0.11231
22	2015	PIT	674.5008	697	-0.03228	594.1524	596	-0.0031	0.604938	0.571133	0.563801	-0.0141	-0.06919
23	2015	SDP	624.6625	650	-0.03898	713.4378	731	-0.02402	0.45679	0.446477	0.433946	-0.02806	-0.05001
24	2015	SEA	697.8216	656	0.063752	718.6746	726	-0.01009	0.469136	0.453747	0.485282	0.069498	0.034416
25	2015	SFG	712.9048	696	0.024289	630.7079	627	0.005914	0.518519	0.54762	0.560948	0.024339	0.081828
26	2015	STL	668.8099	647	0.033709	560.5344	525	0.067685	0.617284	0.594446	0.587398	-0.01186	-0.04842
27	2015	TBD	672.5849	644	0.044387	641.6024	642	-0.00062	0.493827	0.501423	0.523562	0.044153	0.060214
28	2015	TEX	715.7758	751	-0.0469	745.475	733	0.017019	0.54321	0.511097	0.479684	-0.06146	-0.11695
29	2015	TOR	882.8444	891	-0.00915	655.7927	670	-0.0212	0.574074	0.627539	0.644422	0.026904	0.122541
30	2015	WSN	713.5384	703	0.014991	630.5917	635	-0.00694	0.512346	0.546408	0.561476	0.027576	0.095893
31	2016	ARI	732.8145	752	-0.02551	745.408	779	-0.04312	0.425926	0.483867	0.491481	0.015735	0.153912
32	2016	ATL	664.0071	649	0.023123	745.1113	715	0.042114	0.419753	0.455807	0.442633	-0.0289	0.054509
33	2016	BAL	769.6276	744	0.034446	666.9156	694	-0.03903	0.549383	0.531785	0.571136	0.073998	0.039595
34	2016	BOS	890.2168	878	0.013914	747.4408	715	0.045372	0.574074	0.592864	0.586525	-0.01069	0.021689
35	2016	CHC	786.0482	808	-0.02717	539.0726	556	-0.03044	0.635802	0.66464	0.680123	0.023294	0.069708
36	2016	CHW	695.329	686	0.013599	859.3455	854	0.006259	0.481481	0.401104	0.395662	-0.01357	-0.17824
37	2016	CIN	718.5547	716	0.003568	657.1558	676	-0.02788	0.419753	0.526276	0.544542	0.034708	0.297291
38	2016	CLE	786.858	777	0.012687	839.4094	860	-0.02394	0.580247	0.4537	0.46772	0.0309	-0.19393
39	2016	COL	840.2991	845	-0.00556	714.8011	721	-0.0086	0.462963	0.572098	0.580179	0.014125	0.253186
40	2016	DET	785.3088	750	0.047078	712.4467	701	0.016329	0.530864	0.530872	0.548533	0.033268	0.033282
41	2016	HOU	735.3545	724	0.015683	807.0979	727	0.110176	0.518519	0.498108	0.453588	-0.08938	-0.12522
42	2016	KCR	675.417	675	0.000618	580.652	638	-0.08989	0.5	0.525768	0.575019	0.093673	0.150037
43	2016	LAA	733.6014	717	0.023154	850.7717	890	-0.04408	0.45679	0.402383	0.426449	0.059808	-0.06642
44	2016	LAD	710.1952	725	-0.02042	720.9036	682	0.057043	0.561728	0.527943	0.492518	-0.0671	-0.12321
45	2016	MIA	661.422	655	0.009805	763.2941	712	0.072042	0.487654	0.461899	0.42886	-0.07153	-0.12056
46	2016	MIL	737.3456	671	0.098876	754.9036	733	0.029882	0.450617	0.459656	0.488236	0.062177	0.083482
47	2016	MIN	718.5272	722	-0.00481	896.6232	889	0.008575	0.364198	0.405941	0.391059	-0.03666	0.073754
48	2016	NYM	699.5732	671	0.042583	688.5484	702	-0.01916	0.537037	0.479349	0.507942	0.059649	-0.05418
49	2016	NYN	688.1763	680	0.012024	667.2725	617	0.081479	0.518519	0.544363	0.515418	-0.05317	-0.00598
50	2016	OAK	661.7485	653	0.013397	760.8959	761	-0.00014	0.425926	0.430431	0.430645	0.000496	0.011079
51	2016	PHI	606.2715	610	-0.00611	784.1583	796	-0.01488	0.438272	0.380592	0.374124	-0.01699	-0.14637
52	2016	PIT	717.7215	729	-0.01547	761.4214	758	0.004514	0.481481	0.482161	0.470482	-0.02422	-0.02285
53	2016	SDP	643.089	686	-0.06255	772.2268	770	0.002892	0.419753	0.447349	0.40951	-0.08458	-0.0244
54	2016	SEA	766.5077	768	-0.00194	714.4799	707	0.01058	0.530864	0.53779	0.535087	-0.00503	0.007955
55	2016	SFG	707.2152	715	-0.01089	627.8714	631	-0.00496	0.537037	0.556929	0.559221	0.004115	0.041307
56	2016	STL	768.3868	779	-0.01362	716.7668	712	0.006695	0.530864	0.541052	0.534715	-0.01171	0.007255
57	2016	TBD	681.8384	672	0.01464	743.3989	713	0.042635	0.419753	0.472932	0.456887	-0.03393	0.088467
58	2016	TEX	759.8015	765	-0.0068	758.1555	757	0.001526	0.58642	0.504809	0.501084	-0.00738	-0.14552
59	2016	TOR	782.4232	759	0.030861	656.3946	666	-0.01442	0.549383	0.559517	0.586925	0.048984	0.068335
60	2016	WSN	757.1773	763	-0.00763	601.4752	612	-0.0172	0.58642	0.599543	0.613115	0.022637	0.045522

## F. R Codes with Comments for Model Creation

### **#Import the csv files and view them**

```
off_data<-read.csv("D:\\Users\\Parker\\Desktop\\FIU Classwork\\Thesis\\Offensive  
Combined Data.csv",header=TRUE)
```

```
View(off_data)
```

```
def_data<-read.csv("D:\\Users\\Parker\\Desktop\\FIU Classwork\\Thesis\\Defensive  
Combined Data.csv",header=TRUE)
```

```
View(def_data)
```

### **#Make scatterplots for all 65 variables. Will remove those that have no relation to R or RA.**

```
plot(off_data$H[1:450],off_data$R[1:450])  
plot(off_data$Doub[1:450],off_data$R[1:450])  
plot(off_data$Trip[1:450],off_data$R[1:450])  
plot(off_data$HR[1:450],off_data$R[1:450])  
plot(off_data$BB[1:450],off_data$R[1:450])  
plot(off_data$SO[1:450],off_data$R[1:450])  
plot(off_data$SB[1:450],off_data$R[1:450])  
plot(off_data$CS[1:450],off_data$R[1:450])  
plot(off_data$HBP[1:450],off_data$R[1:450])  
plot(off_data$SF[1:450],off_data$R[1:450])  
plot(off_data$NumBat[1:450],off_data$R[1:450])  
plot(off_data$BatAge[1:450],off_data$R[1:450])  
plot(off_data$BA[1:450],off_data$R[1:450])  
plot(off_data$OBP[1:450],off_data$R[1:450])  
plot(off_data$SLG[1:450],off_data$R[1:450])  
plot(off_data$OPS[1:450],off_data$R[1:450])  
plot(off_data$OPSplus[1:450],off_data$R[1:450])  
plot(off_data$TB[1:450],off_data$R[1:450])  
plot(off_data$GDP[1:450],off_data$R[1:450])  
plot(off_data$SacBunt[1:450],off_data$R[1:450])  
plot(off_data$IBB[1:450],off_data$R[1:450])  
plot(off_data$LOB[1:450],off_data$R[1:450])  
plot(off_data$BPF[1:450],off_data$R[1:450])  
  
plot(def_data$CG[1:450],def_data$RA[1:450])  
plot(def_data$SHO[1:450],def_data$RA[1:450])  
plot(def_data$SV[1:450],def_data$RA[1:450])  
plot(def_data$BLSV[1:450],def_data$RA[1:450])  
plot(def_data$HA[1:450],def_data$RA[1:450])  
plot(def_data$Doub[1:450],def_data$RA[1:450])  
plot(def_data$Trip[1:450],def_data$RA[1:450])  
plot(def_data$HRA[1:450],def_data$RA[1:450])  
plot(def_data$BBA[1:450],def_data$RA[1:450])
```

```

plot(def_data$H9[1:450],def_data$RA[1:450])
plot(def_data$BB9[1:450],def_data$RA[1:450])
plot(def_data$SOA[1:450],def_data$RA[1:450])
plot(def_data$SOvBB[1:450],def_data$RA[1:450])
plot(def_data$NumP[1:450],def_data$RA[1:450])
plot(def_data$PAge[1:450],def_data$RA[1:450])
plot(def_data$IBB[1:450],def_data$RA[1:450])
plot(def_data$HBP[1:450],def_data$RA[1:450])
plot(def_data$BK[1:450],def_data$RA[1:450])
plot(def_data$WP[1:450],def_data$RA[1:450])
plot(def_data$FIP[1:450],def_data$RA[1:450])
plot(def_data$WHIP[1:450],def_data$RA[1:450])
plot(def_data$LOB[1:450],def_data$RA[1:450])
plot(def_data$NumFld[1:450],def_data$RA[1:450])
plot(def_data$IPouts[1:450],def_data$RA[1:450])
plot(def_data$DefEff[1:450],def_data$RA[1:450])
plot(def_data$Ch[1:450],def_data$RA[1:450])
plot(def_data$A[1:450],def_data$RA[1:450])
plot(def_data$E[1:450],def_data$RA[1:450])
plot(def_data$FldPct[1:450],def_data$RA[1:450])
plot(def_data$DP[1:450],def_data$RA[1:450])
plot(def_data$PPF[1:450],def_data$RA[1:450])
plot(def_data$OBP[1:450],def_data$RA[1:450])
plot(def_data$SLG[1:450],def_data$RA[1:450])
plot(def_data$OPS[1:450],def_data$RA[1:450])
plot(def_data$TB[1:450],def_data$RA[1:450])
plot(def_data$SB[1:450],def_data$RA[1:450])
plot(def_data$CS[1:450],def_data$RA[1:450])
plot(def_data$CSpct[1:450],def_data$RA[1:450])
plot(def_data$K9[1:450],def_data$RA[1:450])
plot(def_data$RS[1:450],def_data$RA[1:450])
plot(def_data$ERCpct[1:450],def_data$RA[1:450])
plot(def_data$DIPpct[1:450],def_data$RA[1:450])

```

**#Perform SLR and make residual plots for all variables correlated with R or RA**

```

off_H.lm<-lm(R[1:450]~H[1:450],data=off_data)
summary(off_H.lm)
plot(off_data$H[1:450],resid(off_H.lm))
abline(0,0)

off_Doub.lm<-lm(R[1:450]~Doub[1:450],data=off_data)
summary(off_Doub.lm)
plot(off_data$Doub[1:450],resid(off_Doub.lm))
abline(0,0)

```

```
off_HR.lm<-lm(R[1:450]~HR[1:450],data=off_data)
summary(off_HR.lm)
plot(off_data$HR[1:450],resid(off_HR.lm))
abline(0,0)
```

```
off_BB.lm<-lm(R[1:450]~BB[1:450],data=off_data)
summary(off_BB.lm)
plot(off_data$BB[1:450],resid(off_BB.lm))
abline(0,0)
```

```
off_SO.lm<-lm(R[1:450]~SO[1:450],data=off_data)
summary(off_SO.lm)
plot(off_data$SO[1:450],resid(off_SO.lm))
abline(0,0)
```

```
off_BA.lm<-lm(R[1:450]~BA[1:450],data=off_data)
summary(off_BA.lm)
plot(off_data$BA[1:450],resid(off_BA.lm))
abline(0,0)
```

```
off_OBP.lm<-lm(R[1:450]~OBP[1:450],data=off_data)
summary(off_OBP.lm)
plot(off_data$OBP[1:450],resid(off_OBP.lm))
abline(0,0)
```

```
off_SLG.lm<-lm(R[1:450]~SLG[1:450],data=off_data)
summary(off_SLG.lm)
plot(off_data$SLG[1:450],resid(off_SLG.lm))
abline(0,0)
```

```
off_OPS.lm<-lm(R[1:450]~OPS[1:450],data=off_data)
summary(off_OPS.lm)
plot(off_data$OPS[1:450],resid(off_OPS.lm))
abline(0,0)
```

```
off_OPSplus.lm<-lm(R[1:450]~OPSplus[1:450],data=off_data)
summary(off_OPSplus.lm)
plot(off_data$OPSplus[1:450],resid(off_OPSplus.lm))
abline(0,0)
```

```
off_TB.lm<-lm(R[1:450]~TB[1:450],data=off_data)
summary(off_TB.lm)
plot(off_data$TB[1:450],resid(off_TB.lm))
abline(0,0)
```

```
off_LOB.lm<-lm(R[1:450]~LOB[1:450],data=off_data)
summary(off_LOB.lm)
plot(off_data$LOB[1:450],resid(off_LOB.lm))
abline(0,0)
```

```
def_SHO.lm<-lm(RA[1:450]~SHO[1:450],data=def_data)
summary(def_SHO.lm)
plot(def_data$SHO[1:450],resid(def_SHO.lm))
abline(0,0)
```

```
def_SV.lm<-lm(RA[1:450]~SV[1:450],data=def_data)
summary(def_SV.lm)
plot(def_data$SV[1:450],resid(def_SV.lm))
abline(0,0)
```

```
def_HA.lm<-lm(RA[1:450]~HA[1:450],data=def_data)
summary(def_HA.lm)
plot(def_data$HA[1:450],resid(def_HA.lm))
abline(0,0)
```

```
def_Doub.lm<-lm(RA[1:450]~Doub[1:450],data=def_data)
summary(def_Doub.lm)
plot(def_data$Doub[1:450],resid(def_Doub.lm))
abline(0,0)
```

```
def_Trip.lm<-lm(RA[1:450]~Trip[1:450],data=def_data)
summary(def_Trip.lm)
plot(def_data$Trip[1:450],resid(def_Trip.lm))
abline(0,0)
```

```
def_HRA.lm<-lm(RA[1:450]~HRA[1:450],data=def_data)
summary(def_HRA.lm)
plot(def_data$HRA[1:450],resid(def_HRA.lm))
abline(0,0)
```

```
def_BBA.lm<-lm(RA[1:450]~BBA[1:450],data=def_data)
summary(def_BBA.lm)
plot(def_data$BBA[1:450],resid(def_BBA.lm))
abline(0,0)
```

```
def_H9.lm<-lm(RA[1:450]~H9[1:450],data=def_data)
summary(def_H9.lm)
plot(def_data$H9[1:450],resid(def_H9.lm))
```



```

abline(0,0)

def_BB9.lm<-lm(RA[1:450]~BB9[1:450],data=def_data)
summary(def_BB9.lm)
plot(def_data$BB9[1:450],resid(def_BB9.lm))
abline(0,0)

def_SOA.lm<-lm(RA[1:450]~SOA[1:450],data=def_data)
summary(def_SOA.lm)
plot(def_data$SOA[1:450],resid(def_SOA.lm))
abline(0,0)

def_SOvBB.lm<-lm(RA[1:450]~SOvBB[1:450],data=def_data)
summary(def_SOvBB.lm)
plot(def_data$SOvBB[1:450],resid(def_SOvBB.lm))
abline(0,0)

def_FIP.lm<-lm(RA[1:450]~FIP[1:450],data=def_data)
summary(def_FIP.lm)
plot(def_data$FIP[1:450],resid(def_FIP.lm))
abline(0,0)

def_WHIP.lm<-lm(RA[1:450]~WHIP[1:450],data=def_data)
summary(def_WHIP.lm)
plot(def_data$WHIP[1:450],resid(def_WHIP.lm))
abline(0,0)

def_LOB.lm<-lm(RA[1:450]~LOB[1:450],data=def_data)
summary(def_LOB.lm)
plot(def_data$LOB[1:450],resid(def_LOB.lm))
abline(0,0)

def_IPouts.lm<-lm(RA[1:450]~IPouts[1:450],data=def_data)
summary(def_IPouts.lm)
plot(def_data$IPouts[1:450],resid(def_IPouts.lm))
abline(0,0)

def_DefEff.lm<-lm(RA[1:450]~DefEff[1:450],data=def_data)
summary(def_DefEff.lm)
plot(def_data$DefEff[1:450],resid(def_DefEff.lm))
abline(0,0)

def_E.lm<-lm(RA[1:450]~E[1:450],data=def_data)
summary(def_E.lm)
plot(def_data$E[1:450],resid(def_E.lm))

```

```
abline(0,0)

def_OBP.lm<-lm(RA[1:450]~OBP[1:450],data=def_data)
summary(def_OBP.lm)
plot(def_data$OBP[1:450],resid(def_OBP.lm))
abline(0,0)
```

```
def_SLG.lm<-lm(RA[1:450]~SLG[1:450],data=def_data)
summary(def_SLG.lm)
plot(def_data$SLG[1:450],resid(def_SLG.lm))
abline(0,0)
```

```
def_OPS.lm<-lm(RA[1:450]~OPS[1:450],data=def_data)
summary(def_OPS.lm)
plot(def_data$OPS[1:450],resid(def_OPS.lm))
abline(0,0)
```

```
def_TB.lm<-lm(RA[1:450]~TB[1:450],data=def_data)
summary(def_TB.lm)
plot(def_data$TB[1:450],resid(def_TB.lm))
abline(0,0)
```

```
def_K9.lm<-lm(RA[1:450]~K9[1:450],data=def_data)
summary(def_K9.lm)
plot(def_data$K9[1:450],resid(def_K9.lm))
abline(0,0)
```

```
def_DIPpct.lm<-lm(RA[1:450]~DIPpct[1:450],data=def_data)
summary(def_DIPpct.lm)
plot(def_data$DIPpct[1:450],resid(def_DIPpct.lm))
abline(0,0)
```

**'Based on scatterplots, eliminate the following variables (column in respective file is in parentheses).**

**Offense: Trip (7), SB (11), CS (12), HBP (13), SF (14), NumBat (15), BatAge (16), GDP (23),**

**SacBunt (24), IBB (25), BPF (27)**

**Defense: CG (5), BLSV (8), NumP (18), PAge (19), IBB (20), HBP (21), BK (22), WP (23), NumFld (27),**

**Ch (30), A (31), FldPct (33), DP (34), PPF (35), SB (40), CS (41), CSpct (42), RS (44), ERCpct (45)**

**#Make correlation matrices for offense and defense**

```
round(cor(off_data[1:450,c(5:6,8:10,17:22,26)]),2)
round(cor(def_data[1:450,c(6:7,9:17,24:26,28:29,32,36:39,43,46)]),2)
```

**#Perform stepwise regression between null model and full model to find optimal regression equation according to AIC criterion**

```
null_off=lm(R~1,data=off_data[1:450,])
full_off=lm(R~H+Doub+HR+BB+SO+BA+OBP+SLG+OPS+OPSplus+TB+LOB,data=
off_data[1:450,])
step(null_off,scope=list(upper=full_off),direction="both")
```

```
null_def=lm(RA~1,data=def_data[1:450,])
full_def=lm(RA~SHO+SV+HA+Doub+Trip+HRA+BBA+H9+BB9+SOA+SOvBB+FIP
+WHIP+LOB+IPouts+DefEff+E+OBP+SLG+OPS+TB+K9+DIPpct,data=def_data)
step(null_def,scope=list(upper=full_def),direction="both")
```

**#Check variance inflation factors (remember to enable car package)**

```
library("car", lib.loc="D:/Program Files/R/R-3.4.1/library")
off_model=lm(R~OBP+LOB+TB+BB+BA+H+OPSplus,data=off_data[1:450,])
summary(off_model)
vif(off_model)
def_model=lm(RA~HA+BBA+HRA+IPouts+E+DIPpct+DefEff+SV+LOB+SHO+BB9,
data=def_data[1:450,])
summary(def_model)
vif(def_model)
```

**#Correlation and Scatterplot Matrices for Remaining Defensive Variables**

```
round(cor(def_data[1:450,c(6:7,9,12:13,15,26,28:29,32,46)]),2)
pairs(def_data[1:450,c(6:7,9,12:13,15,26,28:29,32,46)])
```

**#Residual Plot for Defensive Model**

```
plot(def_model$fit,resid(def_model))
```

**#Perform ridge estimated regression for the offensive model to account for colinearity issues (attach liureg package)**

```
library("liureg", lib.loc="D:/Anaconda3/R/library")
off_model=liu(R~H+Doub+HR+BB+SO+BA+OBP+SLG+OPS+OPSplus+TB+LOB,dat
a=off_data[1:450,])
summary(off_model)
round(cor(off_data[1:450,c(6,8:9,17:18,26)]),2)
off_model=liu(R~Doub+HR+BB+BA+LOB,data=off_data[1:450,]) #all non-sabermetric
summary(off_model)
```

**#Perform MLR on each regressor vs others, then  $vif=1/(1-R^2)$**

```
Doub_model=liu(Doub~HR+BB+BA+LOB,data=off_data[1:450,])
summary(Doub_model)
```

```
HR_model=liu(HR~Doub+BB+BA+LOB,data=off_data[1:450,])
summary(HR_model)
BB_model=liu(BB~Doub+HR+BA+LOB,data=off_data[1:450,])
summary(BB_model)
BA_model=liu(BA~Doub+HR+BB+LOB,data=off_data[1:450,])
summary(BA_model)
LOB_model=liu(LOB~Doub+HR+BB+BA,data=off_data[1:450,])
summary(LOB_model)
vifs<-c(1/(1-.4007),1/(1-.41),1/(1-.6953),1/(1-.5114),1/(1-.6952))
vifs
```

```
lstats(off_model)
1-pf(1345.439,5,444) #p,n-p-1
```

### **#Correlation and Scatterplot Matrices for Remaining Offensive Variables**

```
round(cor(off_data[1:450,c(6,8:9,17,26)]),2)
pairs(off_data[1:450,c(6,8:9,17,26)])
```

### **#Residual Plot for Offensive Model**

```
plot(off_model$fit,resid(off_model))
```

## G. R Codes with Comments for Cross Validation

### **#Set the models used to cross validate**

```
off_model=liu(R~Doub+HR+BB+BA+LOB,data=off_data[1:450,])
def_model=lm(RA~HA+BBA+HRA+IPouts+E+DIPpct+DefEff+SV+LOB+SHO+BB9,
data=def_data[1:450,])
```

### **#Import csv file of table where output will be stored**

```
tcv<-read.csv("D:\\Users\\Parker\\Desktop\\FIU
Classwork\\Thesis\\ThesisCrossValidation.csv",header=TRUE)
View(tcv)
```

### **#Iterate through remaining dataset**

```
for(i in 451:nrow(off_data)){
  #r=formula using off data columns where formula comes from regression model
  r=predict(off_model,off_data[i,])
  tcv[i-450,3]=r #Subtracting by 450 starts the new file at 1
  #rA=formula using def data columns where formula comes from regression model
  rA=predict(def_model,def_data[i,])
  tcv[i-450,6]=rA
  #Year, Team, RActual, RAActual already copied from data files to this table
  off_pct_error=(r-off_data[i,4])/off_data[i,4]
  tcv[i-450,5]=off_pct_error
  def_pct_error=(rA-def_data[i,4])/def_data[i,4]
  tcv[i-450,8]=def_pct_error
```

### **#W already copied from data files to this table**

```
pyth_actual=tcv[i-450,4]^1.83/(tcv[i-450,4]^1.83+tcv[i-450,7]^1.83)
tcv[i-450,10]=pyth_actual #try with 2 and 1.83
pyth_models=r^2/(r^2+rA^2)
tcv[i-450,11]=pyth_models
pyth_pct_error=(pyth_models-pyth_actual)/pyth_actual #pyth model vs pyth actual
tcv[i-450,12]=pyth_pct_error
w_pct_error=(pyth_models-tcv[i-450,9])/tcv[i-450,9] #pyth model vs wins
tcv[i-450,13]=w_pct_error
}
mean(abs(tcv$DefPctError))
mean(abs(tcv$OffPctError))
mean(abs(tcv$PythPctError))
mean(abs(tcv$WPctError))
```

### **#Write table output back into the file (make sure file isn't open or error will occur)**

```
write.csv(tcv,file="D:\\Users\\Parker\\Desktop\\FIU
Classwork\\Thesis\\ThesisCrossValidation.csv")
```