Predicting NBA Player Win Share Technical Summary

Chris Jin
Joey Berkowitz
Rick Winfrey

This project sought to determine if a linear regression model existed for predicting Win Share (WS)[1] from a variety of predictors from a dataset of NBA player data. The player dataset contains observations about individual player performance for all players in the NBA that played in at least one game during the 2014-2015 NBA season. The traditional method for calculating Win Share is based entirely on offensive performance of players, but we wanted to determine if a combination of defensive and offensive predictors such as assists, steals, blocks, free throws, etc. might help to provide an accurate predictive model for estimating a player's Win Share.

Our initial model included 29 independent variables. The initial linear regression model showed signs of a strong linear relationship between the dependent variable Win Share and the 29 independent variables in our dataset. A summary of our initial model revealed an overall adjusted R^2 value of 0.927 and a highly significant F-statistic value. Additionally, our initial model showed fairly well-behaved residuals with a slightly high kurtosis (see **figure 1** and **figure 2**). Based on the p-values of each of the 29 independent variables, we employed a backwards selection approach to reduce the number of predictors without sacrificing overall statistical significance while preserving the greatest parsimony possible.

[1] Win Share is a statistic calculated based exclusively on offensive attributes of a player's performance such as total points produced during a season, the number of offensive possessions and a few more statistics relating offensive possessions and total points produced to create an overall Win Share rating. For more information, please see http://www.basketball-reference.com/about/ws.html

Based on a backwards AIC step function in R, we settled on the second best model. This second model revealed a lower adjusted R^2 value of 0.838, indicating that some of our correlation had been lost due to pruning 17 independent variables from our initial model, but overall the remaining 12 independent variables were now all statistically significant in terms of predicting Win Share. Residual behavior continued to be well-behaved, although we had hoped to see the kurtosis drop (see **figure 3** and **figure 4**). After examining the summary of our model, as well as the behavior of the residuals, we analyzed multicollinearity to determine what independent variables would be ideal to drop for the next iteration of our model. Based on the variable inflation factors of our 12 predictors, we eliminated two more independent variables for our third model.

The summary of our third model revealed an adjusted R^2 value that was nearly identical to our second model of 0.821. Immediately we noticed two of our independent variables were now no longer statistically significant after removing the previous two predictors with high multicollinearity (usage percentage and steals). Another examination of the variable inflation factors for our remaining predictors revealed that our current model had low multicollinearity. The residuals did not show much change from model 2 to model 3 (see **figure 5** and **figure 6**). Because model 3 revealed two more predictors that were no longer significant after reducing multicollinearity, we further reduced the number of predictors from 10 to 8 for model 4.

Model 4 was the model we felt best about (despite attempting several other combinations of predictors). What we noticed was model 4's combination of 8 predictors provided an intuitive model that was fairly accurate in terms of predicting Win Share for a majority of the players in our dataset. The adjusted R^2 value was the same as model 3 at 0.82. We also saw that our residuals continued to be well behaved (see **figure 7** and **figure 8**), and this model demonstrated low multicollinearity. Given the satisfaction with the model, we looked more deeply into influence and leverage points to try to understand why the kurtosis of all of our models was high.

We have about 30 influence points in this dataset. Looking into these 30 influence points (see **figure 9**), we noticed that these represent the best and worst players of the NBA. Considering that the best players of the NBA are *that much better* than the average player in the NBA, and that the worst players of the NBA are *that much worse* than the average player, we were not surprised to see that the influence points would have a strong influence on the overall fit of the model. However, hat matrix and Cook's D measures for the influence points revealed that were was not a single or few extreme influence or leverage points in the dataset, and we opted to keep all the data in order to provide a more realistic model.

When we examine the individual predictors of our final model, we have the following:

G: games played

GS: games started

FTA: free throws attempted per game

AST: assists per game

BLK: blocks per game

PF: personal fouls per game

ORtg: offensive rating (estimate of points produced or scored per 100 possessions)

DRtg: defensive rating (estimate of points allowed per 100 possessions)

With a summary of beta coefficients as:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.56483    2.13440    4.48  9.7e-06 ***
G            0.01924    0.00371    5.18  3.5e-07 ***
GS           0.03013    0.00335    9.00  < 2e-16 ***
FTA          0.46080    0.05377    8.57  < 2e-16 ***
AST          0.20471    0.04556    4.49  9.2e-06 ***
BLK          0.74684    0.20692    3.61  0.00035 ***
PF          -0.58733    0.12220   -4.81  2.2e-06 ***
ORtg         0.08907    0.00692   12.88  < 2e-16 ***
DRtg        -0.17369    0.01884   -9.22  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Clearly the relationship between the number of **G** (games) and **GS** (games started) and Win Share is going to be significant, as players that play in more games and start more games have a greater opportunity to contribute to the overall wins of their team. It is also worth mentioning that the partial residual plots for **G** (games) and **GS** (games started) are fairly homoscedastic and unbiased (see **figure 12** and **figure 13**).

FTA (free throws attempted) is very significant in contributing to the predicted value of a player’s Win Share. Consideration of why this might be the case reveals that players often fouled are more likely to be stronger offensive players for their teams and therefore are more likely to be involved in a greater number of offensive plays. The partial residual plot for **FTA** reveals a distribution that is skewed to the left, with a majority of observations tending towards the lower end of the range of values observed (see **figure 14**). Due to the dense clustering of observations on the lower end, it reveals that the top 25% of NBA players have a disproportionately greater number of free throw attempts compared to the bottom 75% of NBA players (intuitively this makes sense as the top 25% of NBA players tend to be the players involved in the most offensive plays). Interestingly, the clustering of the partial residuals is still fairly homoscedastic and unbiased. Additionally, the partial residual plot helps to explain through the single predictor where the 29 influence points are likely to have greatest impact on the overall linear model.

Not surprisingly **AST** (assists per game), commonly considered a strong indicator of offensive contribution, weighs in as a significant predictor of a player's Win Share. Interestingly, **BLK** (blocks per game) represents the largest beta coefficient of the 8 predictors in our final model. It is affirming that our model negatively correlates the number of personal fouls (**PF**) and a player's defensive rating (**DTrg)** with a player's Win Share. The greater number of fouls a player commits increases the likelihood that the other team will win by scoring more free throws, and a high defensive rating for a player indicates that the player allowed more shots than a player with a lower defensive rating (in this case, a lower defensive rating is better, which is confirmed by the negative correlation between **DTrg** as a predictor with Win Share). The partial residual plots for **AST, BLK, PF,** and **DTrg** can be viewed as **figure 15 ~ 18**, respectively.

Given the high kurtosis of this model, however, we were eager to perform k-fold cross-validation to see if the influence or leverage points discussed earlier would lead to significant differences in our model's overall effectiveness based on different folds. We were happy to see that a 10 fold cross-validation revealed that our model held up fairly well against this dataset, indicating that we have settled on a satisfactory model (see **figure 10** and **figure 11**). What is worth pointing out is that the observed values plotted against the predicted values in **figure 10** contain an interesting left side tail that indicates a slight skewing of data to the left. When we compare that to the residuals plotted against our predicted values in **figure 11**, we see the same trend in our data, indicating that our high kurtosis remains fairly constant through each of the 10 folds. This observation gave us confidence that we had arrived at a suitable model that balanced offensive and defensive predictors for helping predict a player's Win Share.

The original goal for this project was to determine if it was possible to predict Win Share from a variety of player predictors that combined defensive with the often used offensive parameters. We were happy to discover that a statistically significant linear regression model does exist between Win Share and a combination of defensive and offensive predictors. The defensive predictors we found to be most helpful in predicting Win Share are blocks, the avoidance of personal fouls, and maintaining a low defensive rating. The offensive predictor of assists is intuitive, but we were surprised to see that free throw attempted was a significant predictor. Not surprisingly, a player's overall offensive rating was found to be significant as well. Although obvious after initially seeing the model, we were delighted to find that the number of games a player played along with the number of game starts a player had also contributed significantly to the overall linear model for a player’s Win Share.

Appendix

Figure 1

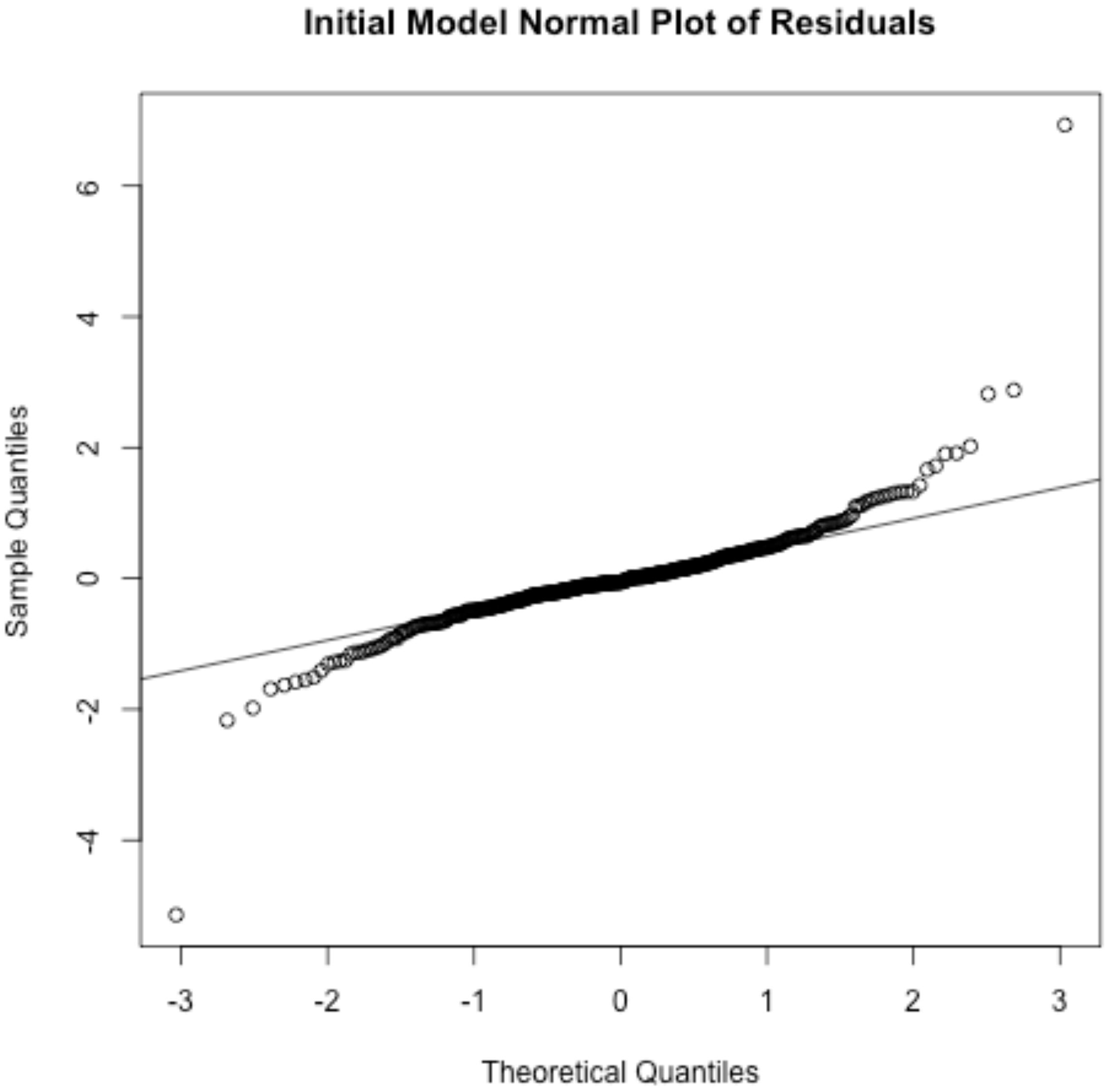


Figure 2

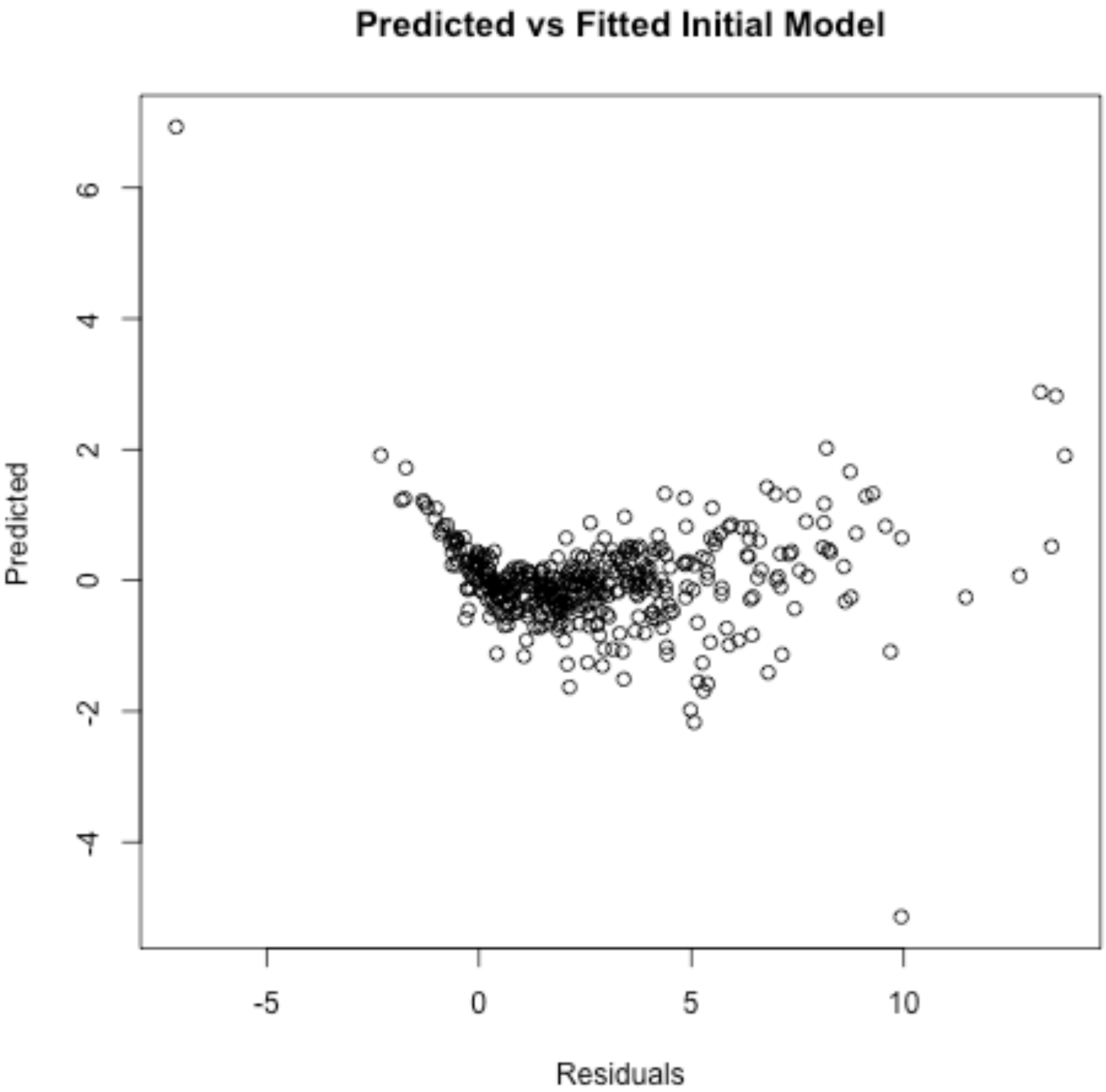

Predicted vs Fitted Initial Model
Predicted
6
4
2
0
-2
-4
-5
0
5
10
Residuals

Figure 3

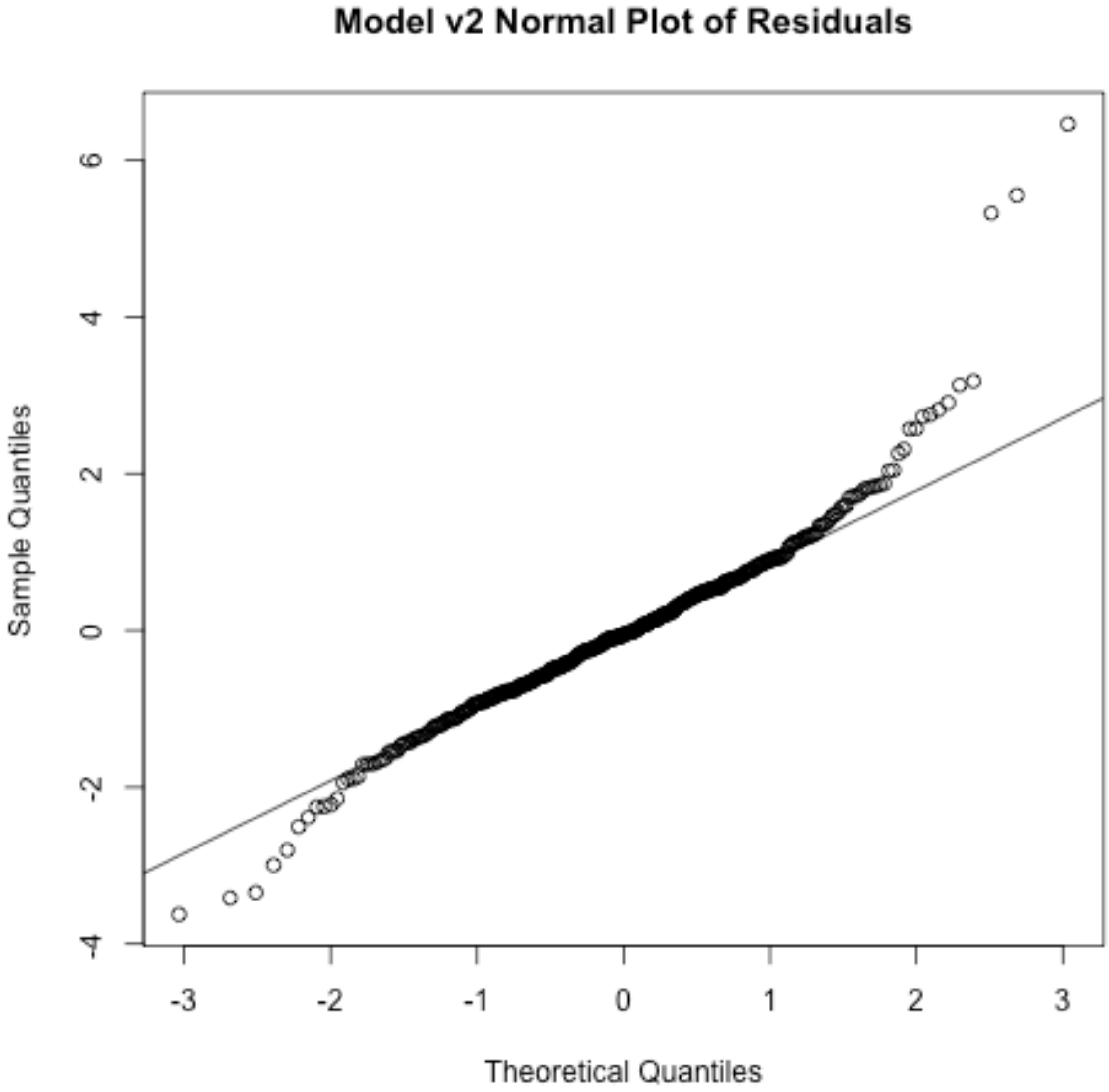

Model v2 Normal Plot of Residuals
Sample Quantiles
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles

Figure 4

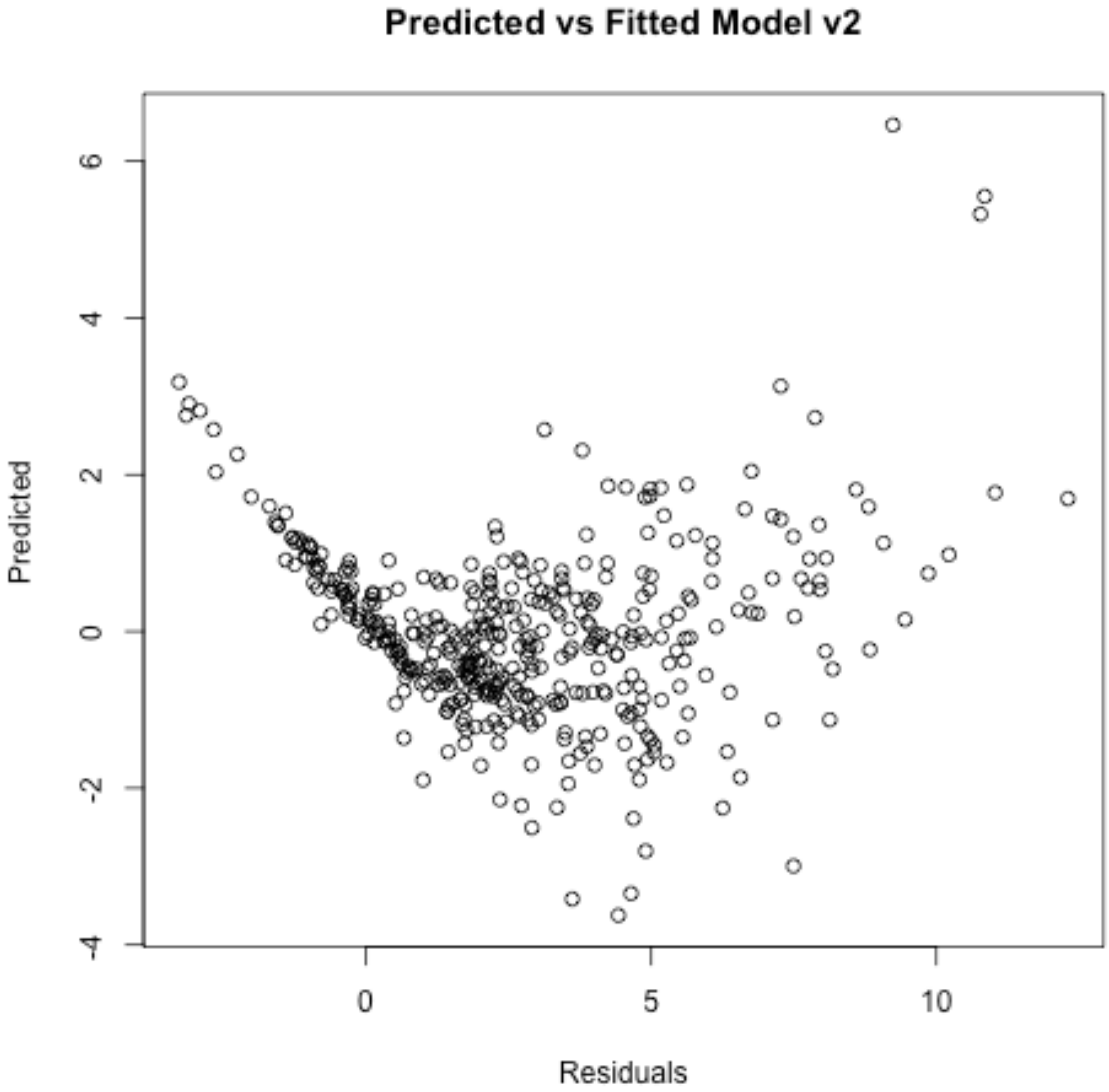

Predicted vs Fitted Model v2
Predicted
-4
-2
0
2
4
6
0
5
10
Residuals

Figure 5

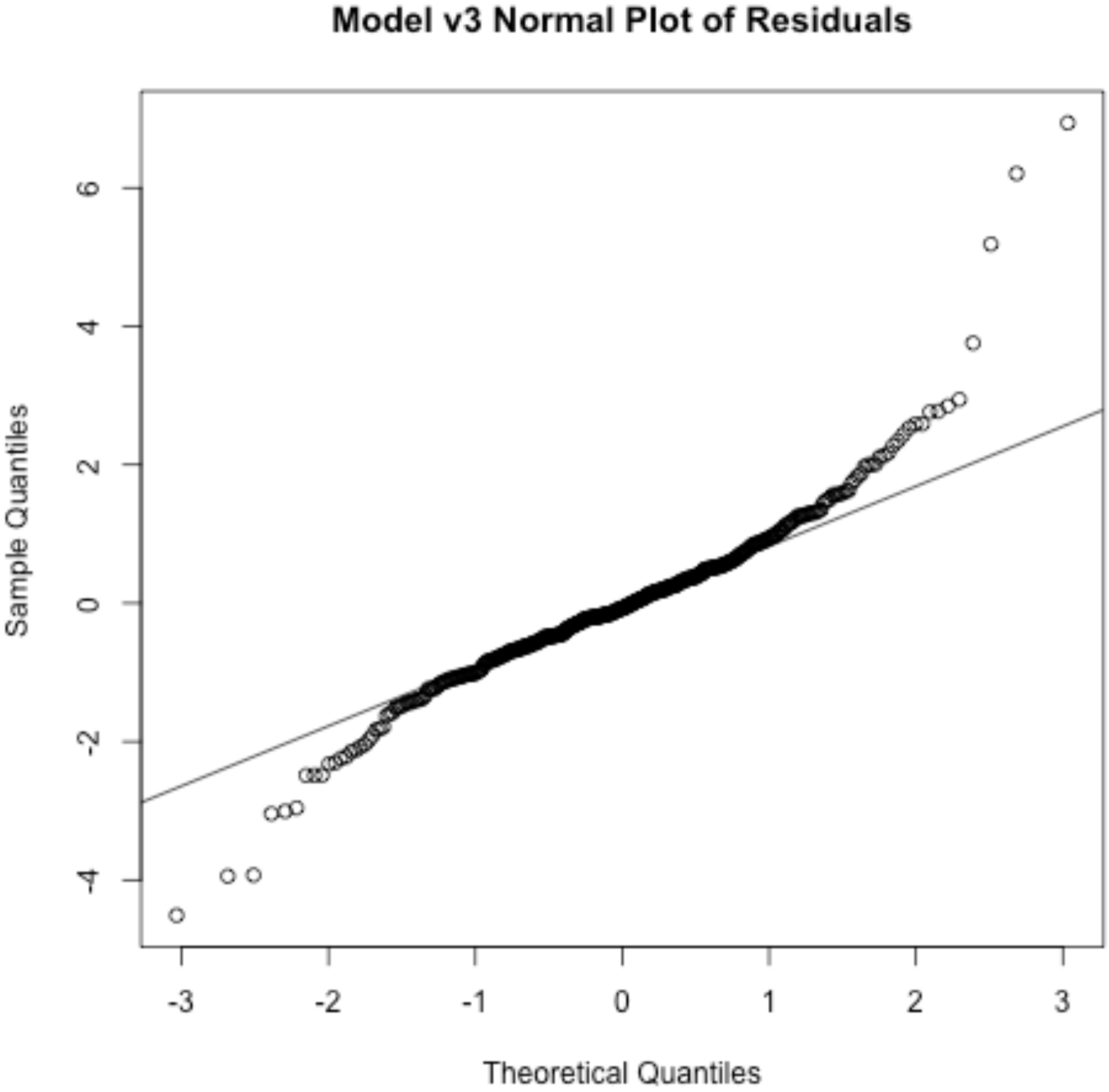

Model v3 Normal Plot of Residuals
Sample Quantiles
-4
-2
0
2
4
6
Theoretical Quantiles
-3
-2
-1
0
1
2
3

Figure 6

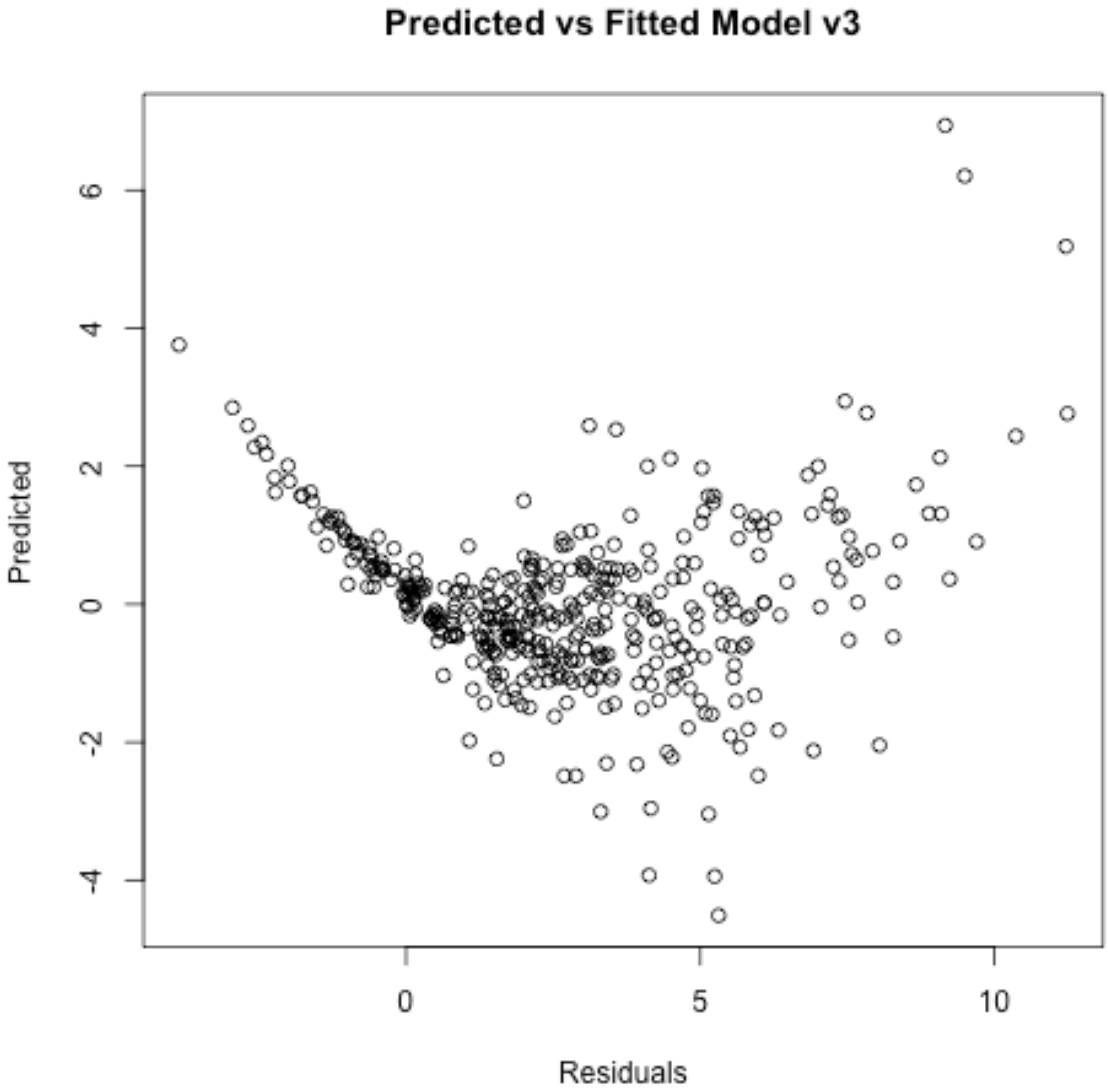

Predicted vs Fitted Model v3
Predicted
6
4
2
0
-2
-4
0
5
10
Residuals

Figure 7

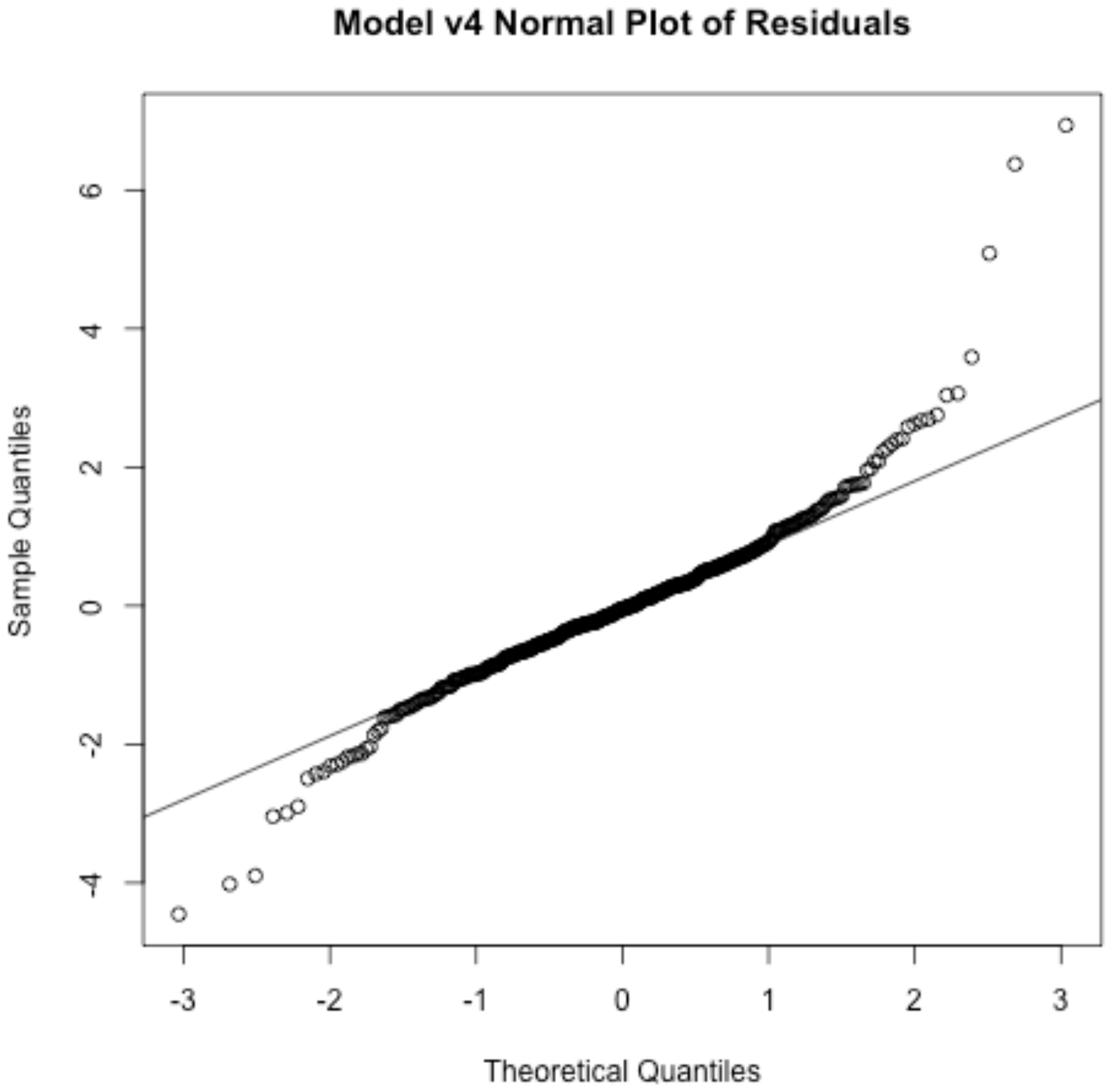

Model v4 Normal Plot of Residuals
Sample Quantiles
-4
-2
0
2
4
6
Theoretical Quantiles
-3
-2
-1
0
1
2
3

Figure 8

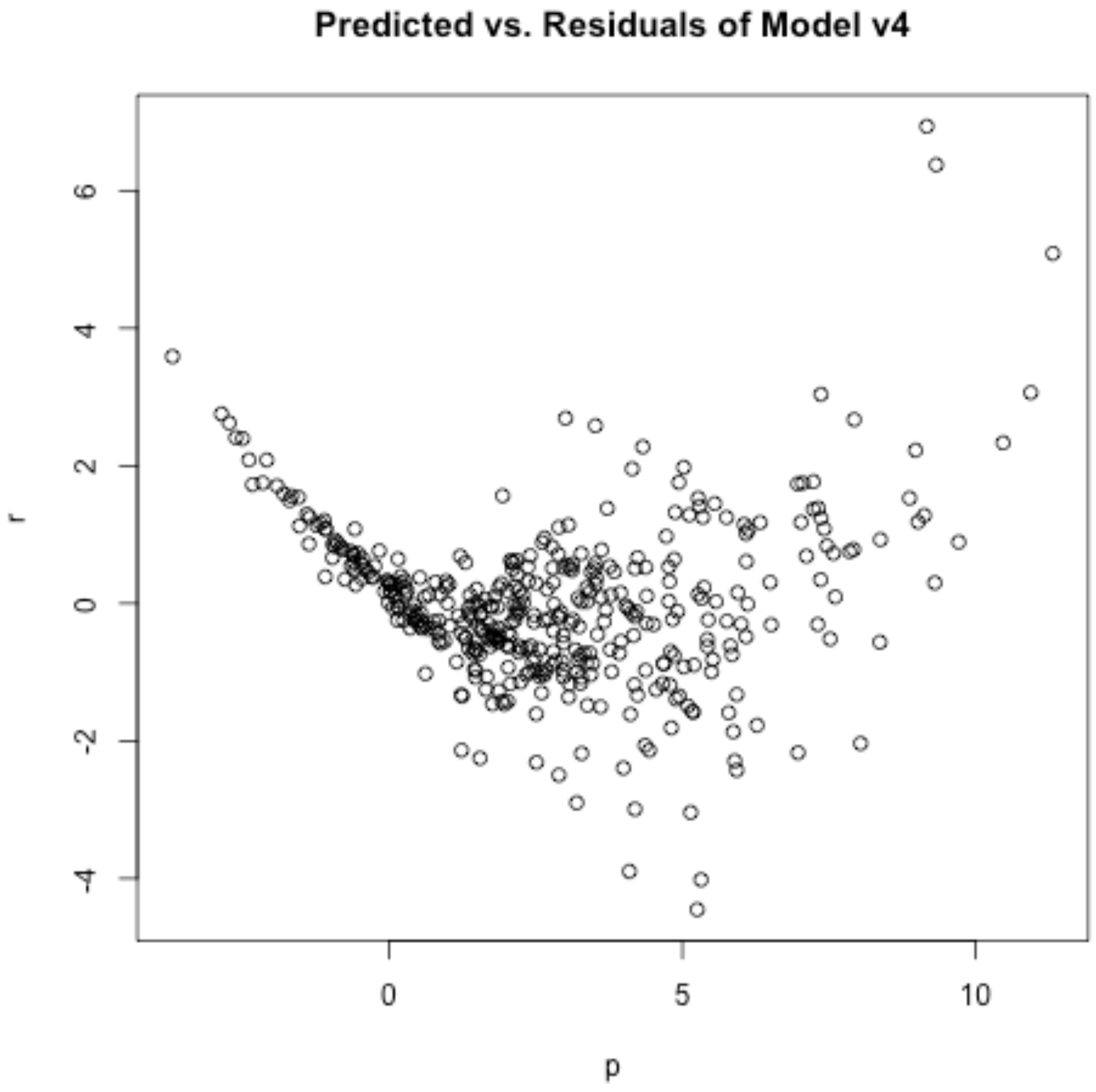
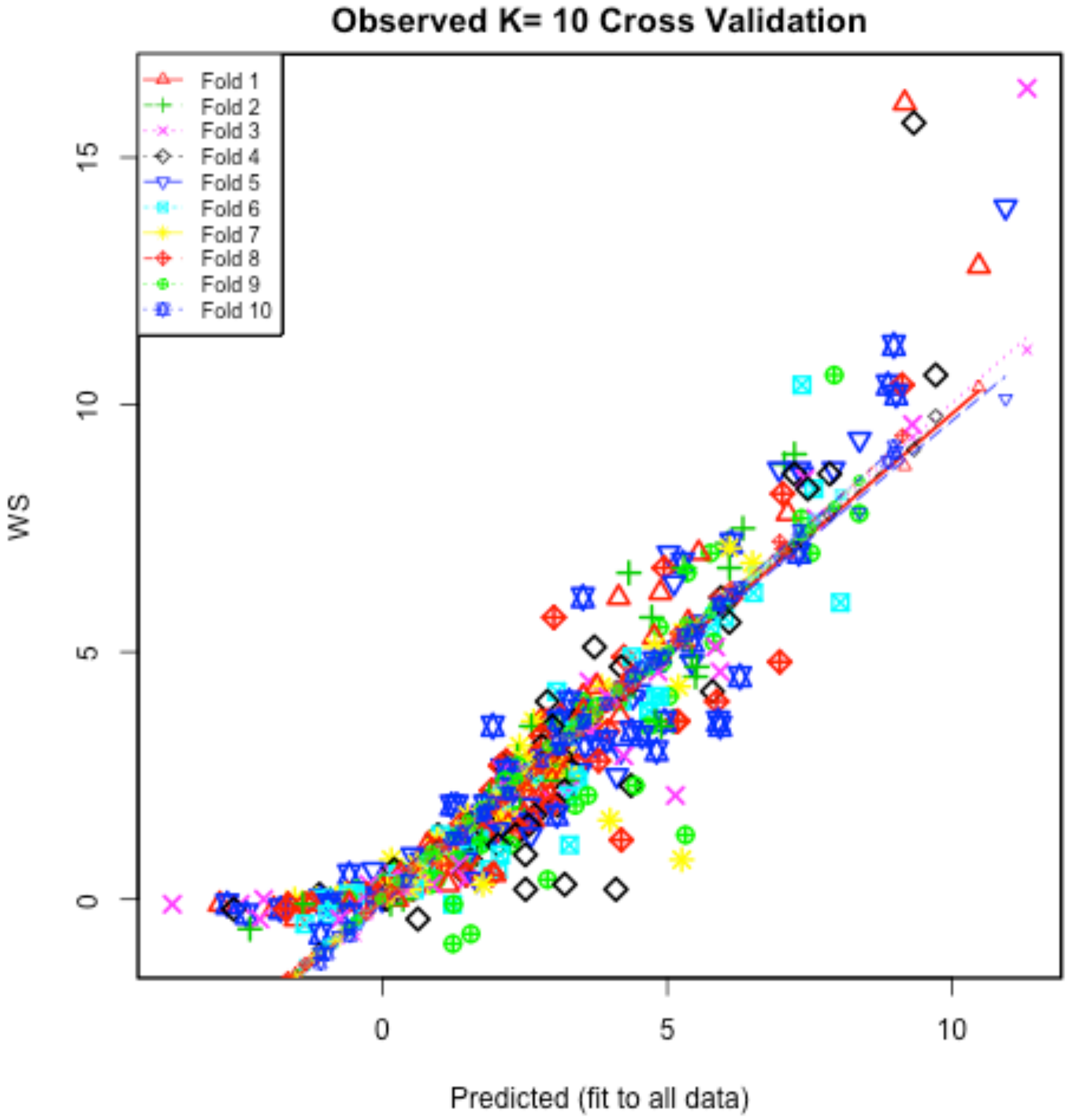


Figure 9 (influence points of final model)

```
Potentially influential observations of
	 lm(formula = WS ~ G + GS + FTA + AST + BLK + PF + ORtg + DRtg,      data = nba_cleaned) :

    dfb.1_ dfb.G dfb.GS dfb.FTA dfb.AST dfb.BLK dfb.PF dfb.ORtg dfb.DRtg dffit   cov.r   cook.d hat
1    0.19   0.08 -0.17   1.01_* -0.01   -0.22   -0.16  -0.02    -0.22     1.25_*  0.72_*  0.17   0.07_*
2   -0.04  -0.25  0.08  -0.65    1.38_* -0.02    0.03   0.69    -0.21     1.74_*  0.49_*  0.31   0.07_*
3    0.40  -0.12  0.25  -0.18    0.68   -0.28   -0.22   0.42    -0.55     1.22_*  0.55_*  0.15   0.05
4   -0.17  -0.03 -0.12   0.17    0.14    0.86   -0.42   0.03     0.18     1.04_*  1.00    0.12   0.13_*
5    0.00  -0.04  0.02   0.10   -0.11    0.24   -0.04   0.09    -0.03     0.48_*  0.99    0.03   0.06
8    0.06   0.00 -0.04   0.16    0.03   -0.08    0.01  -0.01    -0.06     0.23    1.10_*  0.01   0.08_*
10  -0.06  -0.02  0.13   0.10    0.06   -0.06   -0.11   0.10     0.04     0.36    0.90_*  0.01   0.02
14   0.01   0.00  0.01  -0.01    0.02    0.04   -0.03   0.00    -0.01     0.06    1.08_*  0.00   0.05
15  -0.01   0.03 -0.06  -0.02    0.05    0.19   -0.07   0.03     0.00     0.23    1.09_*  0.01   0.08_*
29  -0.03  -0.01  0.00   0.03   -0.11   -0.03    0.02   0.01     0.03    -0.13    1.09_*  0.00   0.06
38   0.02   0.00  0.00   0.01   -0.05   -0.01    0.02  -0.01    -0.01    -0.06    1.08_*  0.00   0.06
58  -0.01   0.00 -0.01  -0.01    0.01    0.02   -0.01   0.02     0.00     0.03    1.08_*  0.00   0.05
59  -0.03   0.05  0.17  -0.35    0.09   -0.08   -0.19   0.10     0.01    -0.54_*  1.04    0.03   0.09_*
64  -0.02  -0.01  0.00  -0.03    0.01    0.05    0.00   0.00     0.02     0.06    1.12_*  0.00   0.09_*
138 -0.09   0.13  0.09  -0.30    0.18    0.01   -0.17   0.02     0.09    -0.46_*  0.99    0.02   0.05
153  0.01   0.02  0.01   0.03   -0.04   -0.02   -0.02  -0.02    -0.01    -0.05    1.08_*  0.00   0.05
227  0.26  -0.11 -0.17  -0.31    0.26   -0.01    0.08   0.17    -0.32    -0.57_*  0.93_*  0.04   0.05
261 -0.08   0.02 -0.11   0.30   -0.38   -0.02   -0.02   0.07     0.07    -0.49_*  0.99    0.03   0.06
284 -0.08   0.35  0.02   0.13    0.02    0.12    0.12  -0.81     0.31    -0.94_*  0.84_*  0.09   0.07_*
300 -0.01   0.02 -0.11  -0.06   -0.14   -0.07    0.23   0.11    -0.04    -0.36    0.91_*  0.01   0.02
319 -0.25  -0.06 -0.01  -0.04   -0.37   -0.03    0.00   0.34     0.17    -0.66_*  0.77_*  0.05   0.03
349  0.06   0.14  0.08   0.08   -0.26   -0.03   -0.13  -0.04    -0.04    -0.35    1.06    0.01   0.07_*
385 -0.06   0.10  0.21  -0.32   -0.09    0.06   -0.13   0.10     0.04    -0.50_*  0.93    0.03   0.04
387  0.06   0.12  0.12  -0.52   -0.06    0.06    0.04   0.15    -0.12    -0.71_*  0.84_*  0.05   0.04
466  0.31  -0.06  0.11   0.04   -0.05    0.01   -0.13  -0.40    -0.17     0.53_*  0.97    0.03   0.05
471  0.01  -0.23  0.03   0.10   -0.18   -0.12    0.31  -0.30     0.10     0.63_*  0.87_*  0.04   0.04
481  0.21   0.12  0.02   0.04   -0.01    0.09   -0.15  -0.40    -0.08     0.46_*  1.02    0.02   0.06
```

Figure 10



Observed K= 10 Cross Validation
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10
WS
0
5
10
15
0
5
10
Predicted (fit to all data)

Figure 11

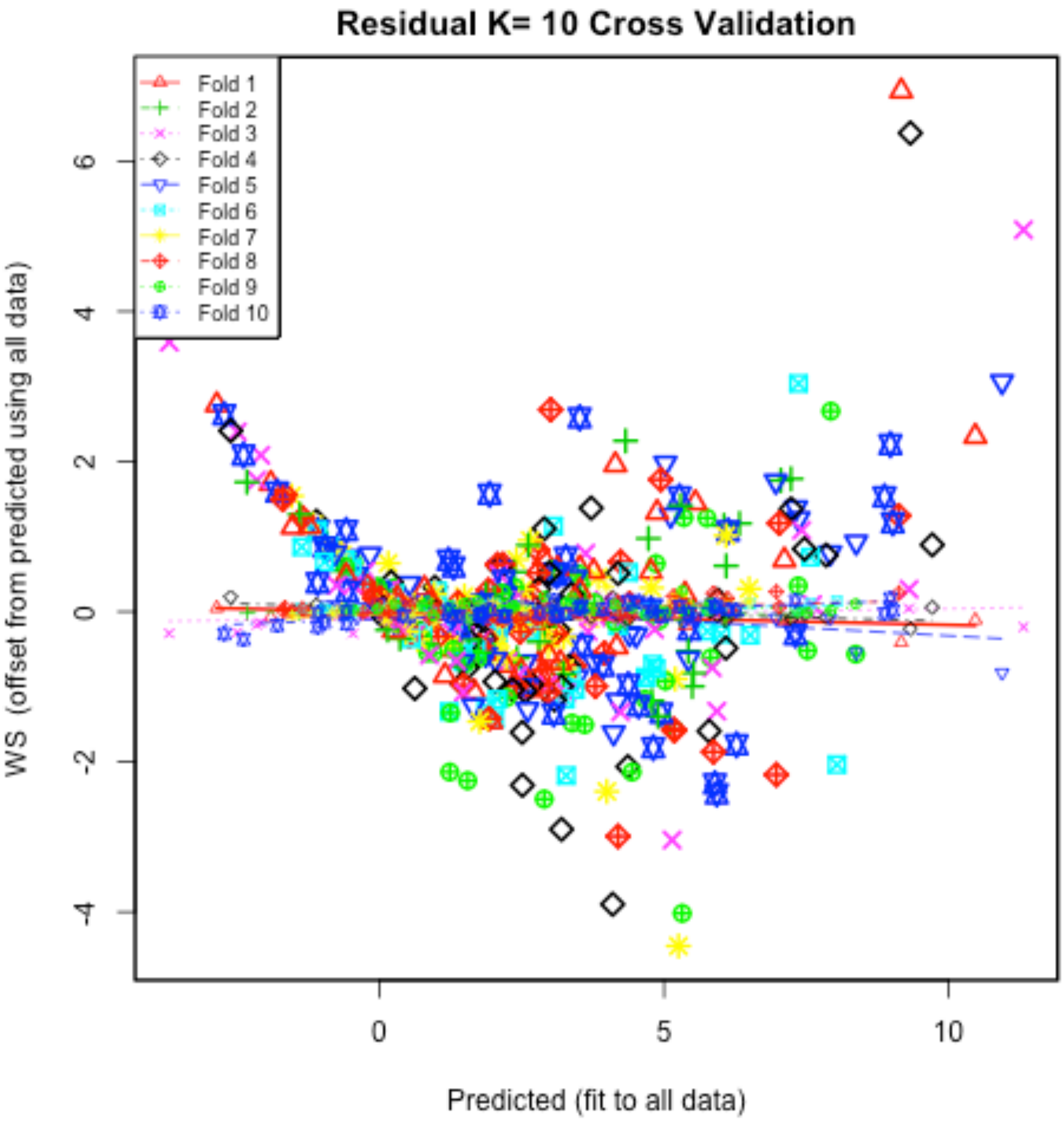

Residual K= 10 Cross Validation
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Fold 6
Fold 7
Fold 8
Fold 9
Fold 10
WS (offset from predicted using all data)
6
4
2
0
-2
-4
0
5
10
Predicted (fit to all data)

Figure 12

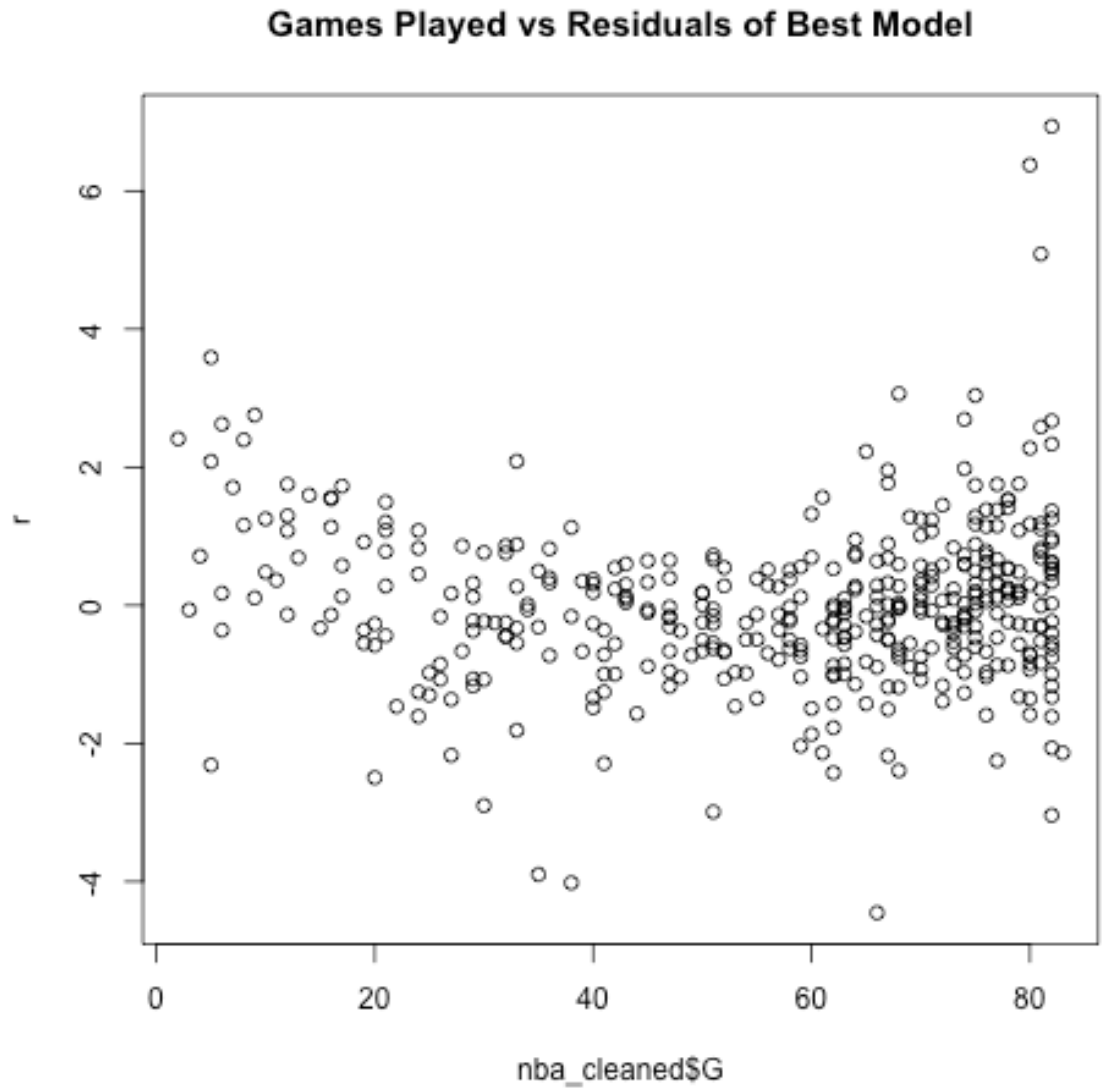

Games Played vs Residuals of Best Model
r
6
4
2
0
-2
-4
0
20
40
60
80
nba_cleaned$G

Figure 13

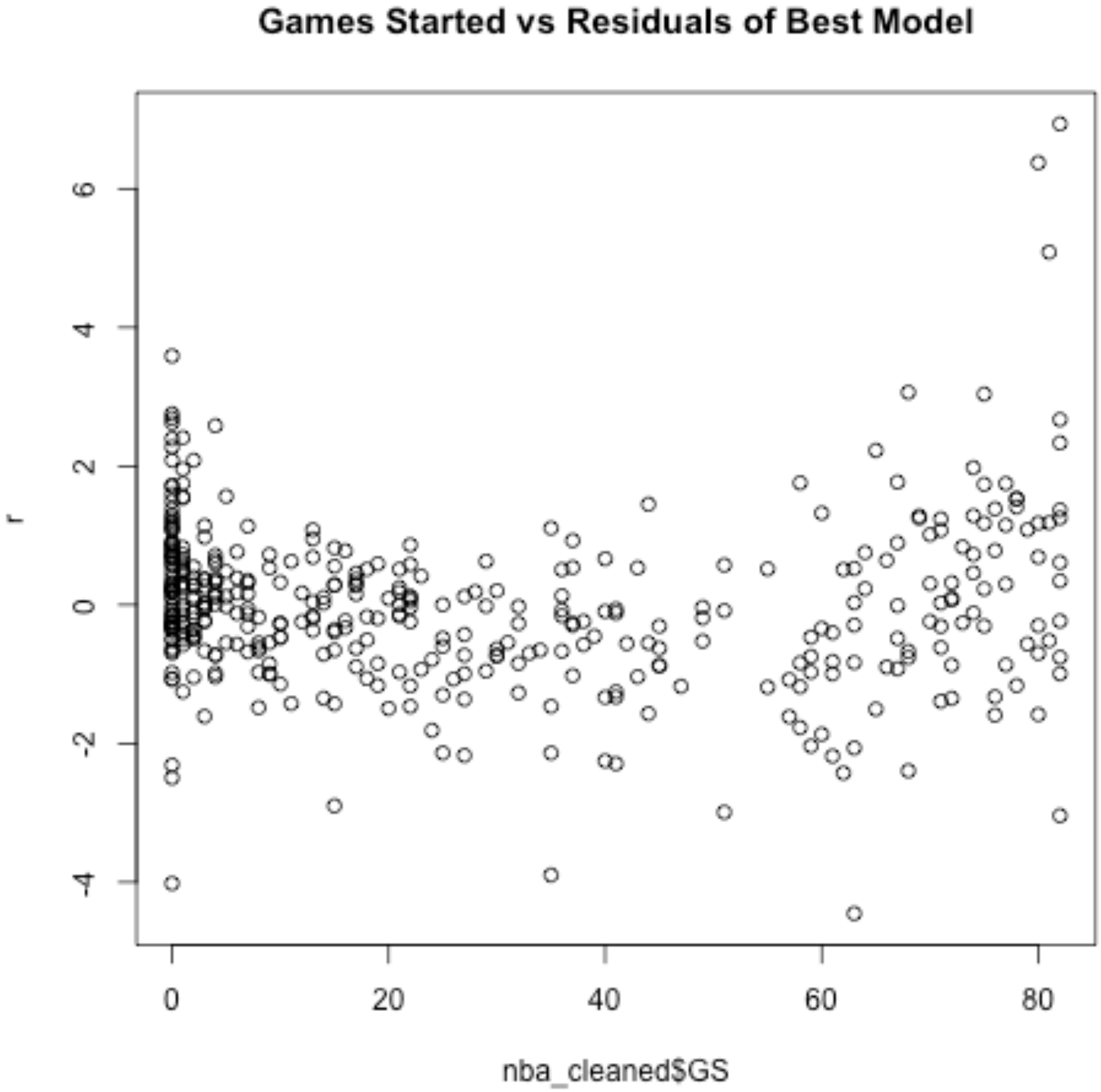

Games Started vs Residuals of Best Model
r
6
4
2
0
-2
-4
0
20
40
60
80
nba_cleaned$GS

Figure 14

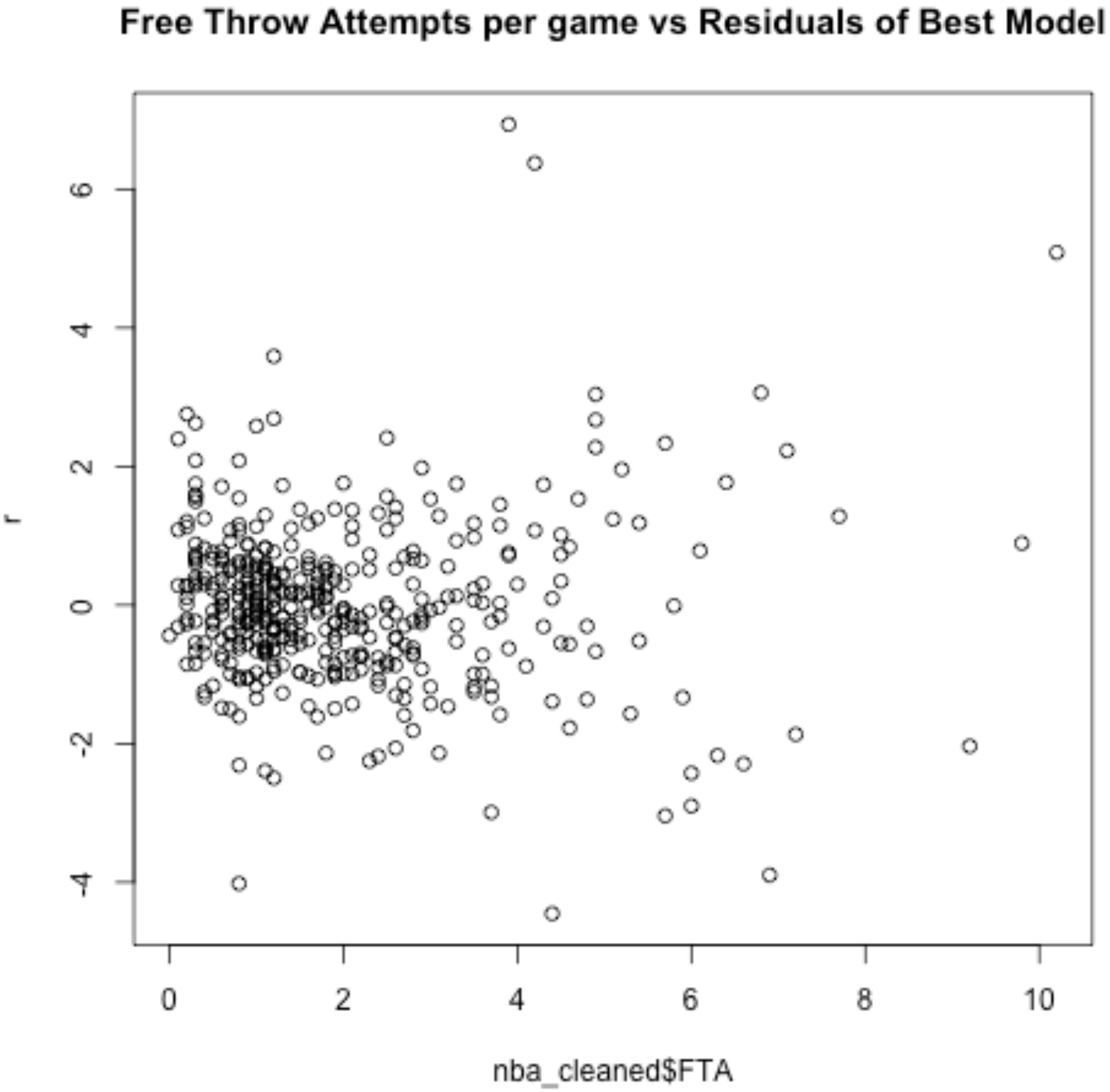

Free Throw Attempts per game vs Residuals of Best Model
r
-4
-2
0
2
4
6
0
2
4
6
8
10
nba_cleaned$FTA

Figure 15

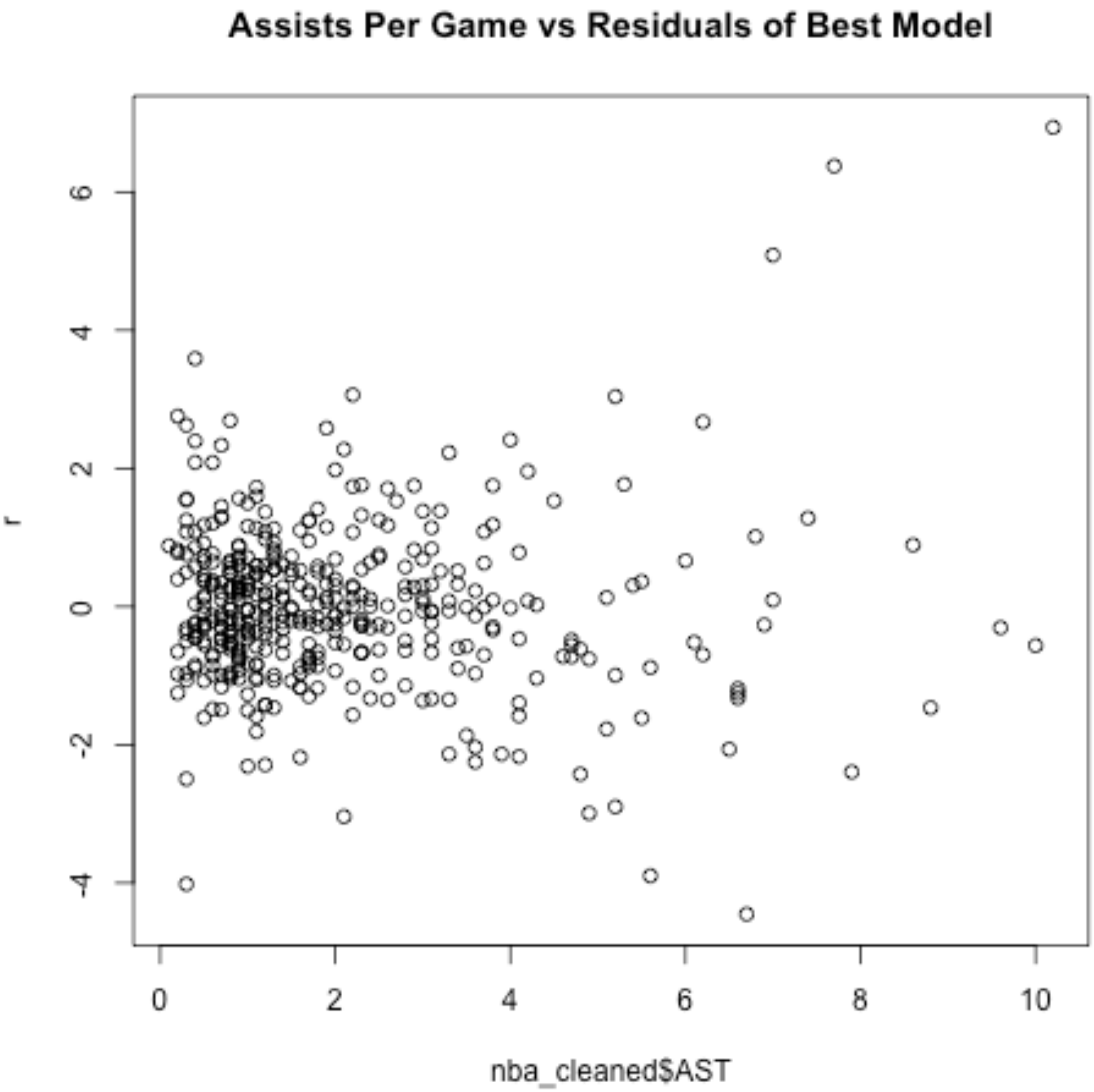

Assists Per Game vs Residuals of Best Model
r
-4
-2
0
2
4
6
0
2
4
6
8
10
nba_cleaned$AST

Figure 16

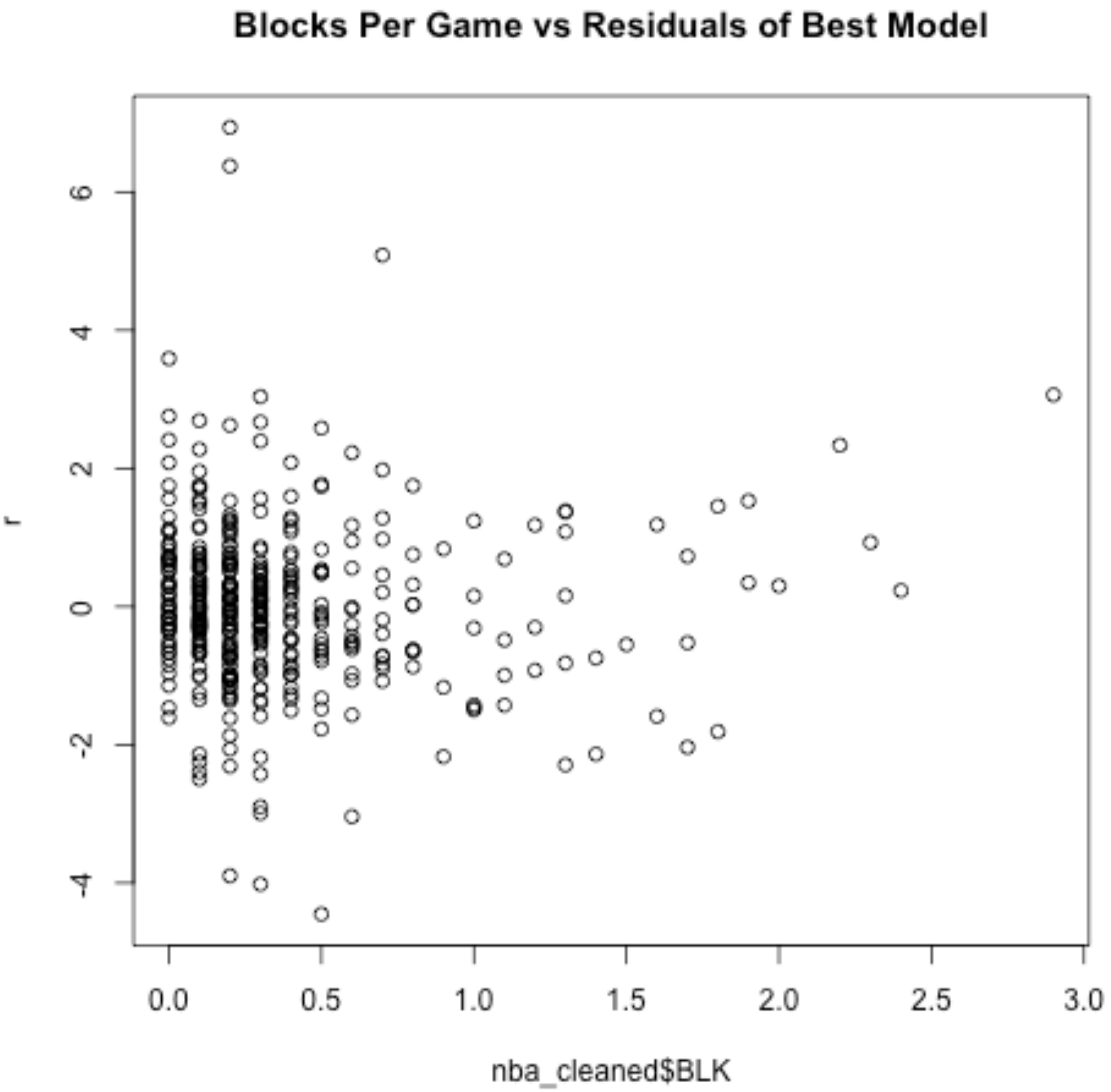

Blocks Per Game vs Residuals of Best Model
r
6
4
2
0
-2
-4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
nba_cleaned$BLK

Figure 17

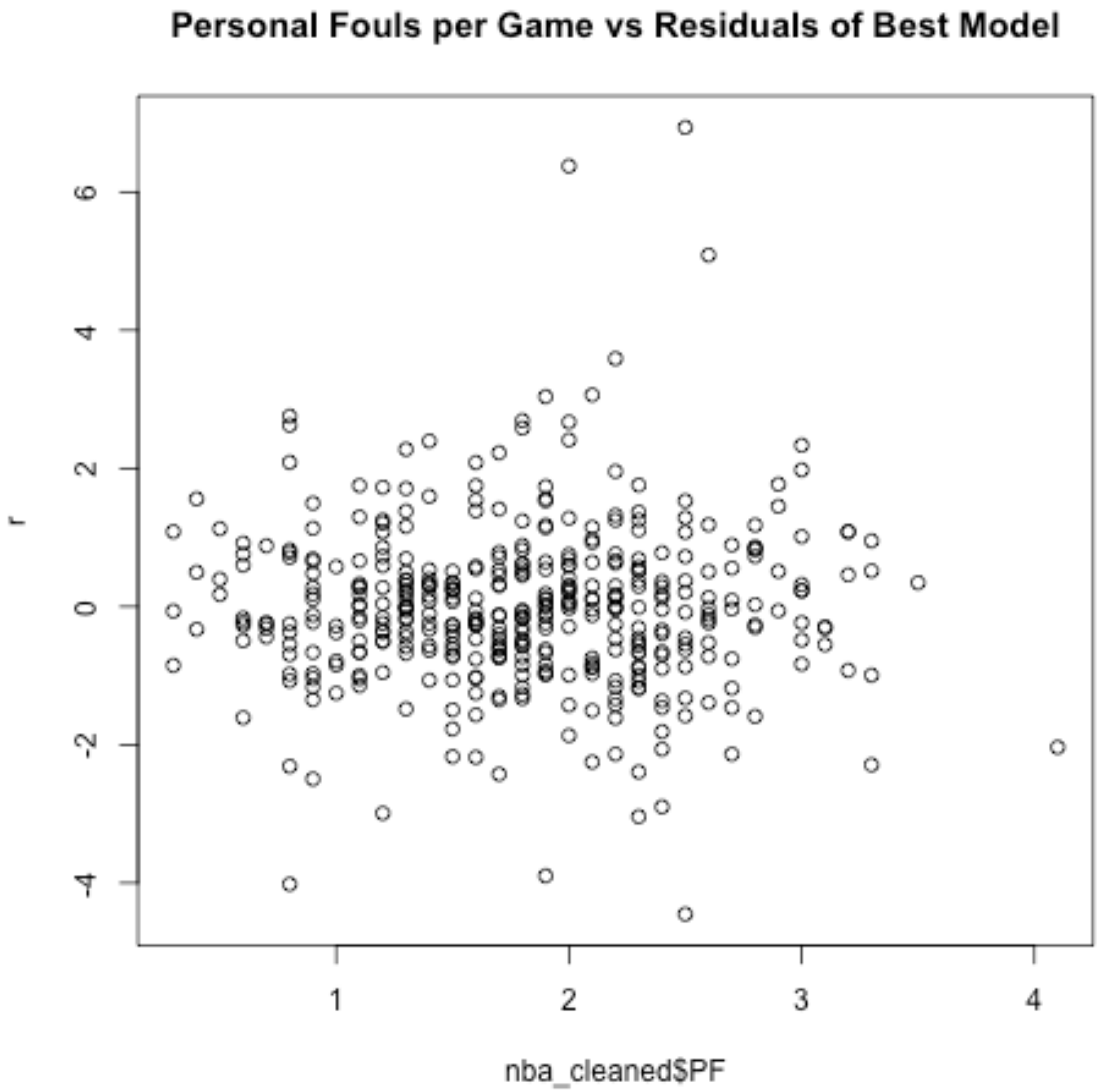

Personal Fouls per Game vs Residuals of Best Model
r
6
4
2
0
-2
-4
1
2
3
4
nba_cleaned$PF

Figure 18

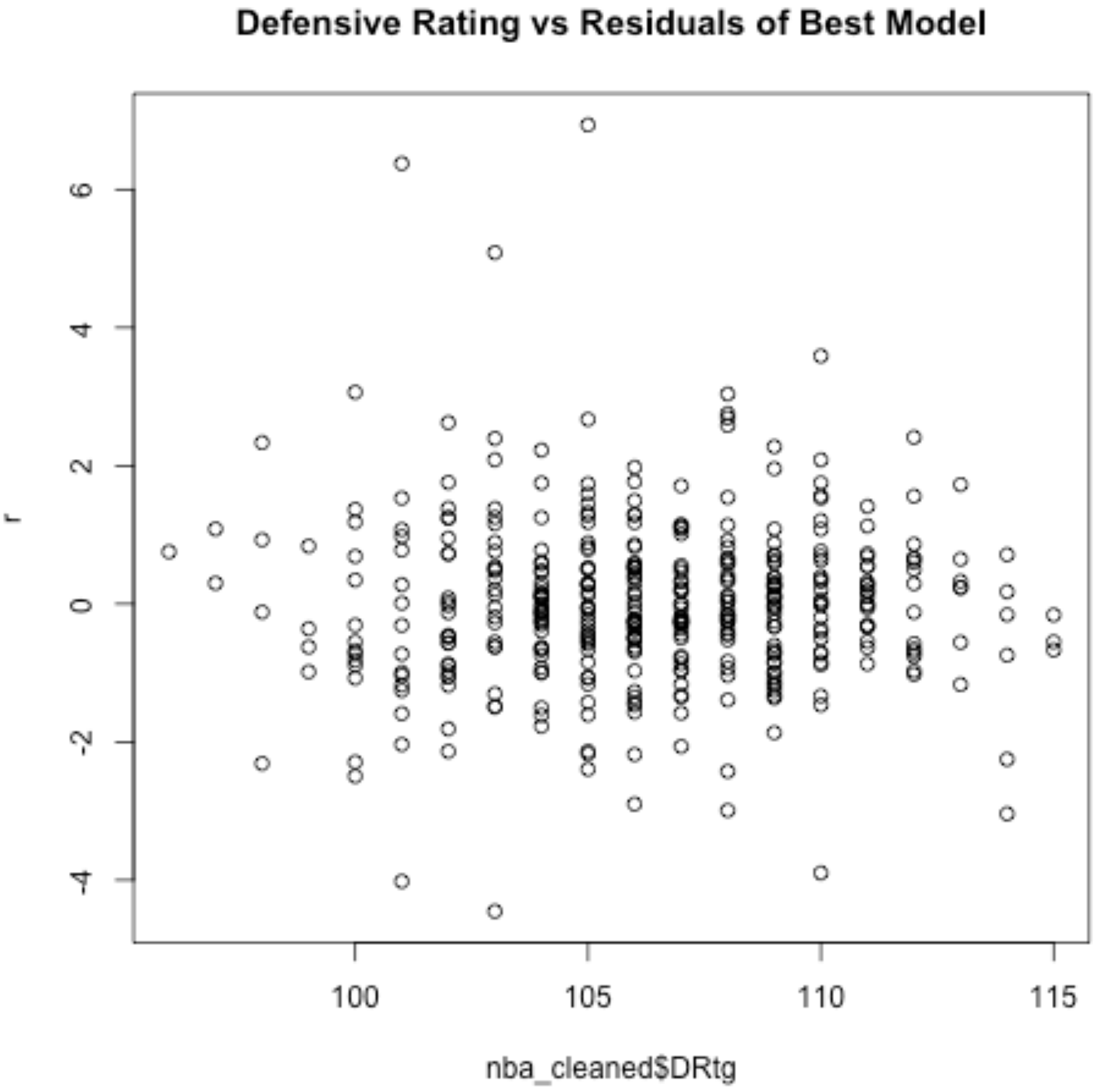
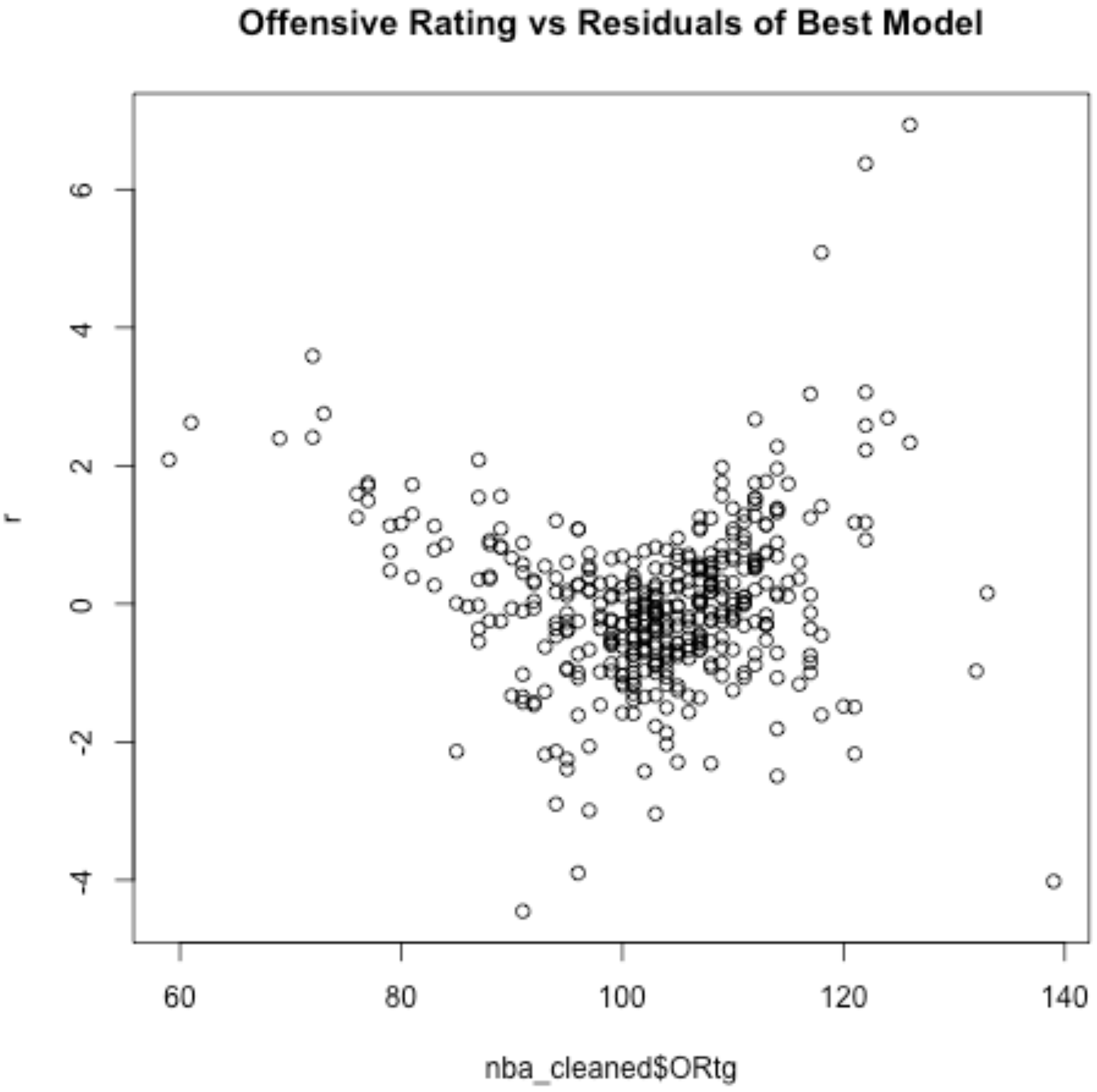

Defensive Rating vs Residuals of Best Model
r
-4
-2
0
2
4
6
100
105
110
115
nba_cleaned$DRtg

Figure 19


Offensive Rating vs Residuals of Best Model
r
-4
-2
0
2
4
6
60
80
100
120
140
nba_cleaned$ORtg

Create Training / Test dataset in R

```
clean_data <- function(dataset) {
  na.omit(dataset)
}

randomize <- function(dataset) {
  dataset$randomized <- runif(length(dataset[,1]))
  dataset
}

randomize_ordered <- function(dataset) {
  randomized_dataset <- randomize(dataset)
  randomized_dataset[order(randomized_dataset$randomized), ]
}

split_dataset <- function(dataset, first_half = 1) {
  half <- length(dataset[,1]) / 2
  if (first_half == 1) {
    dataset[seq(1,half),]
  } else {
    dataset[seq(half + 1, length(dataset[,1])),]
  }
}

# Example
example_dataset <- read.csv("example_data.csv", header = TRUE, na.strings="")
training <- split_dataset(randomize_ordered(clean_data(example_dataset)), first_half = 1)
test <- split_dataset(randomize_ordered(clean_data(example_dataset)), first_half = 0)
```