No professional sport has benefitted as much from the rise of social media as the NBA has. Every night, NBA twitter explodes with amazing comebacks, dunks, and highlights. All of the behind the scenes drama, like Kawhi Leonard forcing Oklahoma City’s hand in trading Paul George to the Clippers, are now readily available to casual fans via social media platforms like Twitter and websites like Bleacherreport.com. NBA podcasters like Zach Lowe and Bill Simmons now have cult followings that they have never had in the past. There truly has never been a better time to be an NBA fan than today.

With all that in mind, I think it is time to take a step back and look into a question that is on everyone’s mind: who are the best players in the NBA? Does the King still reign supreme, even in his 30s? What about some of the electric guards in the league, like Steph Curry, James Harden or Russell Westbrook? What about some of the news guys on the block, like Giannis Antetokounmpo or Joel Embiid? Perhaps the real question is: how do we even measure who the best player is?

Luckily, the NBA is in the midst of a massive analytics boom. There are number of advanced analytics statistics that can be used to declare a player’s value, such as Win Shares (WS) and Value Over Replacement Player (VORP). I decided to go with Player Efficiency Rating (PER) because, in the words of the metric’s creator, John Hollinger, “The PER sums up all a player’s positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player’s performance.” This seems tailor made to measure who the best players in the league are.

**Objective: Can we predict a basketball player’s PER based on his other basketball metrics?**

I used Basketball Reference’s player information from the 2016-2017 and the 2017-2018 season as my two datasets. The 2016-2017 season dataset was used as a testing/training set, while the 2017-2018 season dataset was used to ultimately make predictions. Because PER is a continuous variable, this is an example of supervised learning, so I performed a **multiple linear regression**.

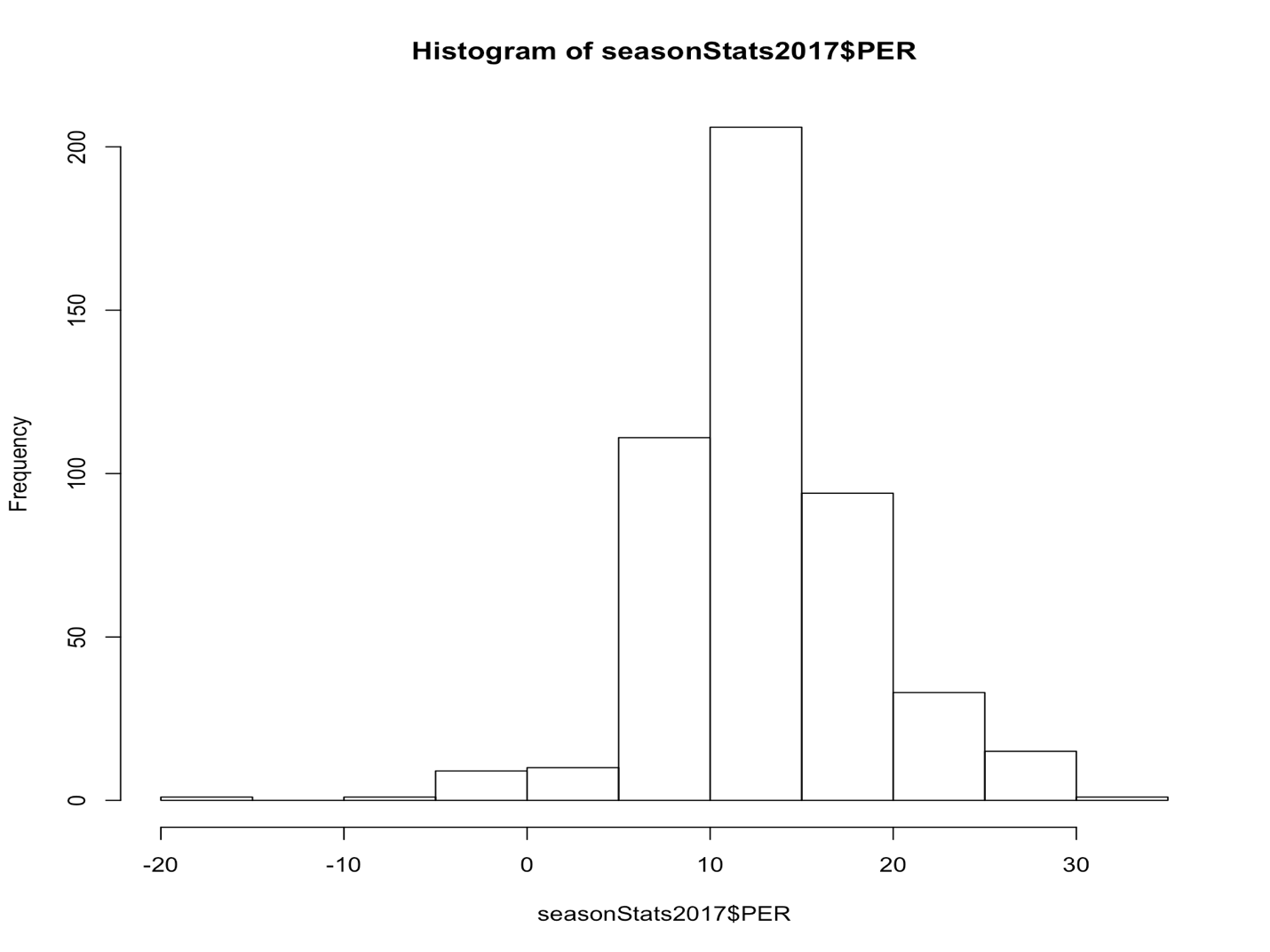
**Exploratory Data Analysis:**

I first split the 2016-2017 dataset 80%/20% into training/testing datasets. I then started cleaning and manipulating the training set. There was a column ‘X’ that denotes the index of each player, so I got rid of it. There were two columns that had no information in them – blanl and blank2, so I removed them. Next, I looked into the NA’s in the dataset. Most of them were concentrated in three columns – Field Goal Percentage, 2 Point Percentage and 3 Point Percentage. After digging into Field Goal Percentage, I realized that the NAs exist because a player had a 0 for Field Goals Attempted. Field Goal Percentage is equal to Field Goals Made / Field Goal Attempted, so if the denominator is 0, then the value is NA. The same thing was true with 2 Point and 3 Point Percentages, but with 2/3 Point Shots Attempted respectively. I replaced these NAs with 0s and moved on. Next, there was a total Points scored column and a total number of Games played column, so I created a Points Per Game (PPG) column by dividing the two columns. Next, I decided to filter out players who had 0 PPG. Finally, there were players who occupied multiple rows. This only occurred to players who had been traded in the year; the dataset included a row of the player from their original team, their traded team, and a third row with their combined stats from both teams. I kept the combined stats and removed the other two rows.

Next, I got rid of outliers. I used the Interquartile Range definition of outliers (Q1 – 1.5IQR, Q3 + 1.5IQR) to detect outliers, and I deleted those rows. I found no outliers for the hard numbers (Assists, Blocks, Steals, etc.), but found outliers for the percentage-based numbers (Assist Percentage, Block Percentage, Steal Percentage, etc.) and advanced statistics (VORP, WS, etc.). However, some of these outliers were players who played big minutes– for example, Russell Westbrook was a high Usage Rate Percentage outlier, and it would be foolhardy to delete his information. So I arbitrarily decided that outliers who played over 250 minutes would be kept in the dataset, while outliers who played less than 250 minutes would be deleted.

**What does the distribution of PERs look like?**

After this data cleaning/manipulation, I plotted the PER values:



Luckily, they were normally distributed with a skewness of -0.22, so I did not have to log-scale these values.

**Feature Selection**

Before deciding on which features to select for my model, I realized something: a lot of these variables have in-built redundancy. For example, total Assists and Assist Percentage measure the same exact thing, but in different ways. So I split up my big 2016-2017 dataset into two smaller ones: one with percentages and one with their total number counterparts. The advanced stats were included in both sets.

With these splits, there are 32 and 34 total features for the Percentages dataset and the Total Numbers dataset respectively. Though that is progress, as we started with 50 total features, that’s still way too many to manually find which features should be used to build the model. So I used the forward and backward selection algorithms for both datasets.

For the Percentage-based dataset, both algorithms selected 21 out of 32 total features:

formulaTrainingSet2017Backward <- PER ~ G + MP + TS. + X3PAr + FTr + ORB. + DRB. + TRB. + AST. + STL. + BLK. + TOV. + USG. + WS.48 + OBPM + BPM + VORP + X2P. + PF + PTS

AIC: - 450.87

formulaTrainingSet2017Forward <- PER ~ WS.48 + USG. + ORB. + FG. + AST. + STL. +

PPG + TOV. + DRB. + DWS + BLK. + DBPM + MP + PF + OBPM +

PTS + X3PAr + TRB. + TS. + FTr + X2P.

AIC: - 445.74

For the Total Number-based dataset, the backward selection selected 22 out of 34 features, while the forward selection selected 23:

formulaTrainingSet2017BackwardNumbers <- PER ~ GS + MP + DWS + WS + WS.48 + OBPM + BPM + FG + FGA + X3P + X3PA + X3P. + FT + FTA + ORB + DRB + AST + STL + BLK + TOV + PF + PPG

AIC : 45. 614

formulaTrainingSet2017ForwardNumbers <- PER ~ WS.48 + PPG + X3P + MP + FGA + TOV + FT + DBPM + TRB + STL + WS + BLK + PF + OWS + AST + X3P. + X2PA + OBPM + FG + GS + FTA + ORB + BPM

AIC: 46.44

The Percentages are a way better set of predictors than Hard Numbers are, which is what I expected. Both the forward and backward selected predictors are pretty good, but the backward selected ones have a slightly lower AIC, making them slightly better. Regardless, we will run each set of features into regression models to make sure this isn’t misleading.

**Model Selection and Training**:

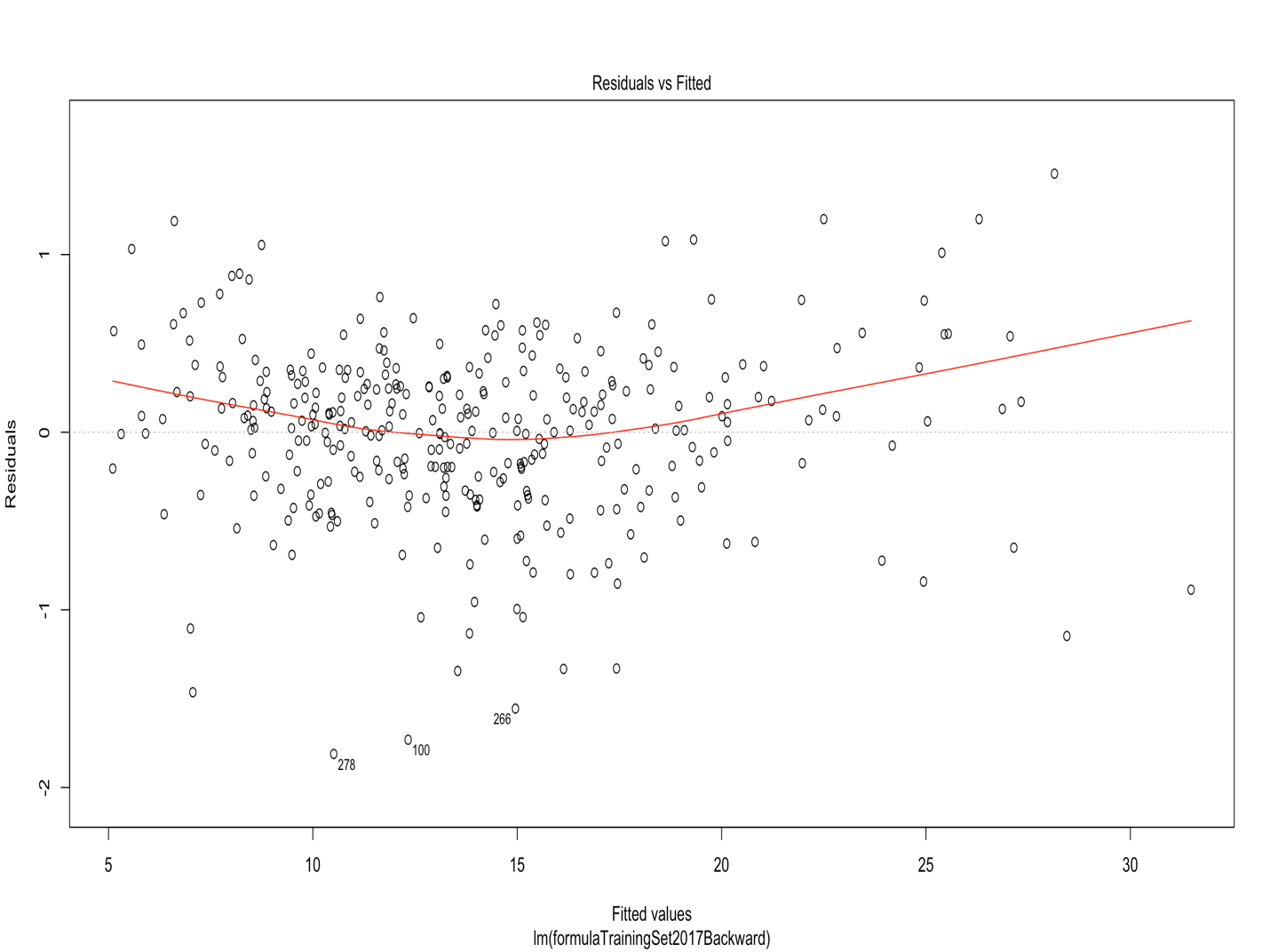
I ran all four sets of variables into a multiple linear regression model and selected the best of the four:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset and  Forward or Backward Selection | RMSE for training dataset | RMSE for testing dataset | MAE for  testing set | MAE for  training set |
| Percentages -Backward | 0.4844163 | 0.8274524 | 0.3635426 | 0.5272481 |
| Percentages - Forward | 0.4866507 | 0.8726221 | 0.3636661 | 0.5285422 |
| Numbers - Backward | .9994329 | 1.507918 | 0.7207541 | 1.070891 |
| Numbers - Forward | .9977033 | 1.500097 | 0.7239139 | 1.065693 |

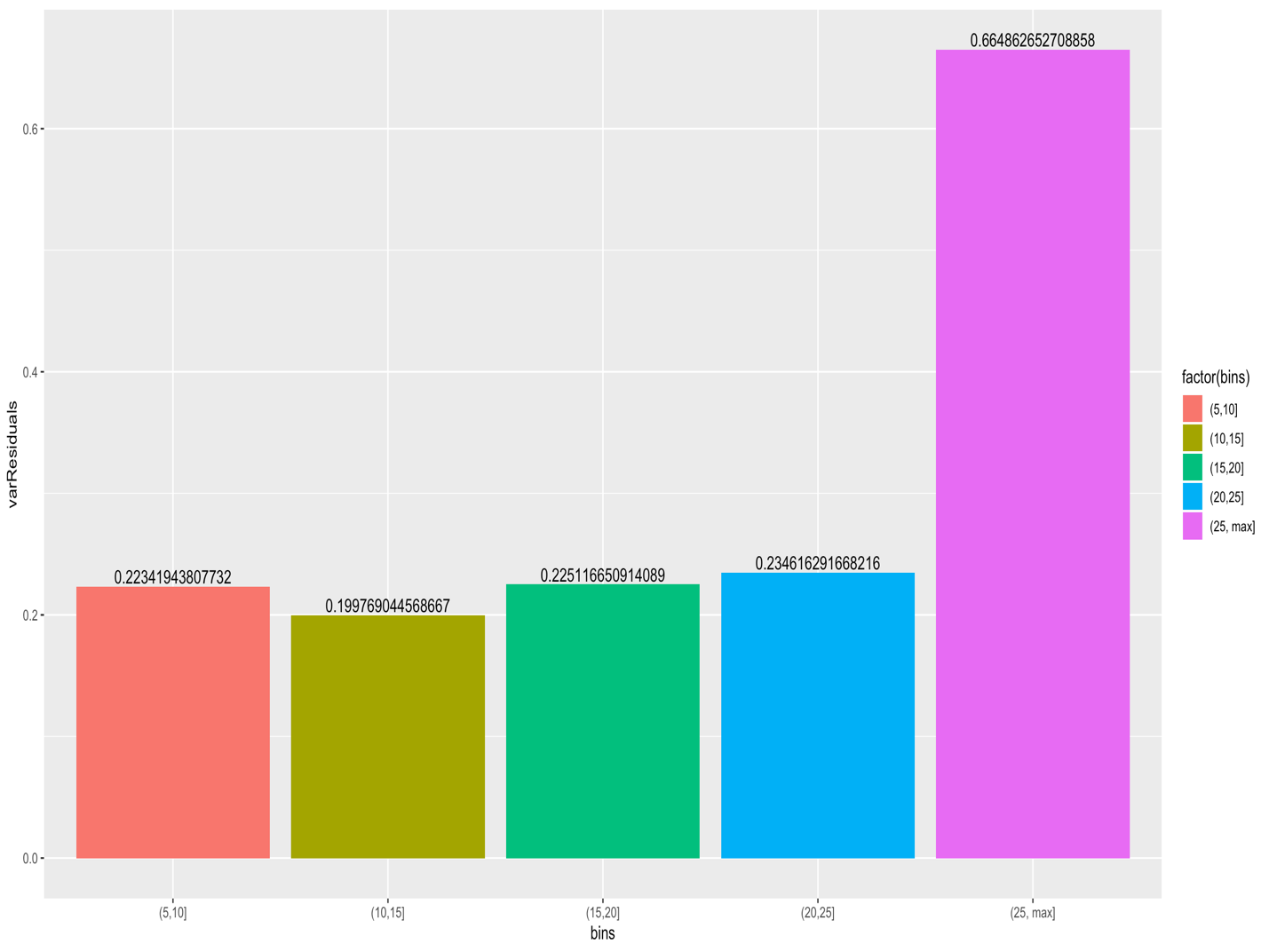
There seems to be some level of overfitting for all four regressions, though the backward selection of variables from the Percentages dataset has the least amount of overfitting. Thus, this seems to be the best model out of the four.

We also have to check the residual plots:

Residual vs. Fitted Plot

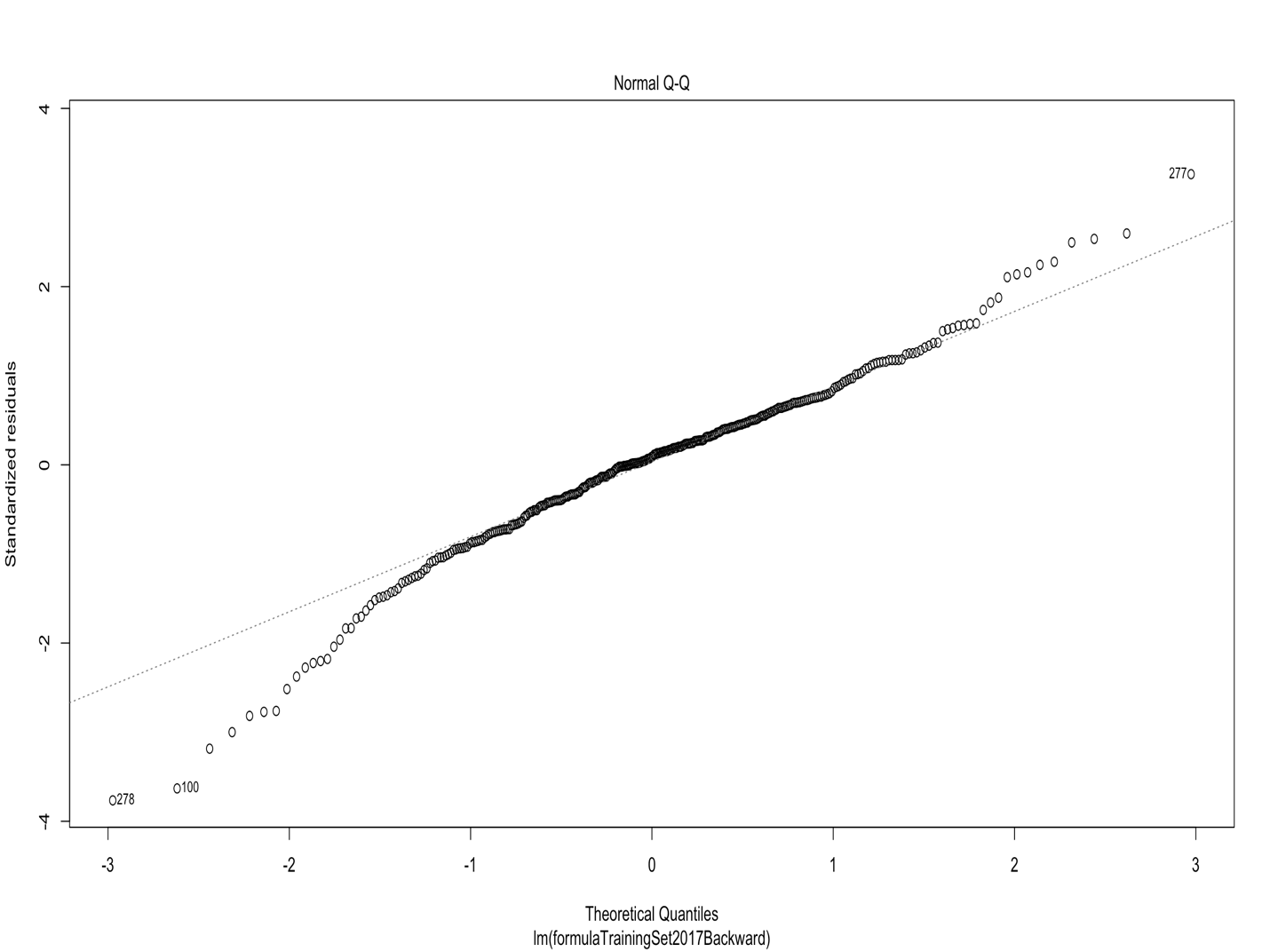


There does appear to be some heteroskedasticity here, but let’s see how the variance of the residuals changes as the fitted values increases. I binned the Fitted values by groups of 5 (so from 5-10, 10-15, 15-20, 20-25 and 25+) and checked the variance of the residuals in each bin:



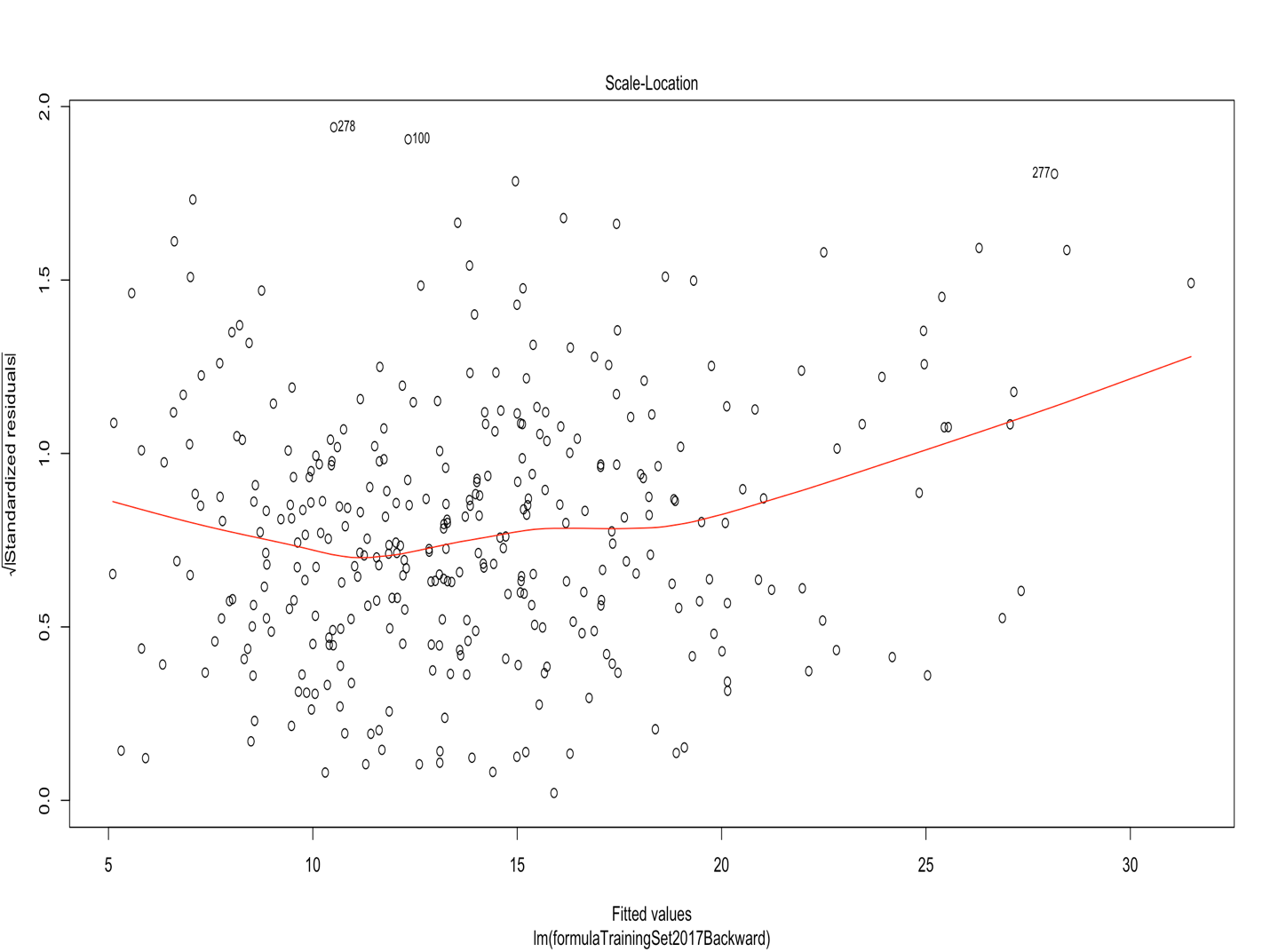
The variance of the Residuals stays pretty constant from 5-25, but then jumps for values greater than 25. However, the general rule is that the variance has to change by a factor of 4 or greater for the heteroskedasticity to matter. In this case, the variance changes by a factor of 3ish, so we can proceed as if the linear regression is the best model for the data.

Normal Quantile-Quantile (Q-Q) Plot



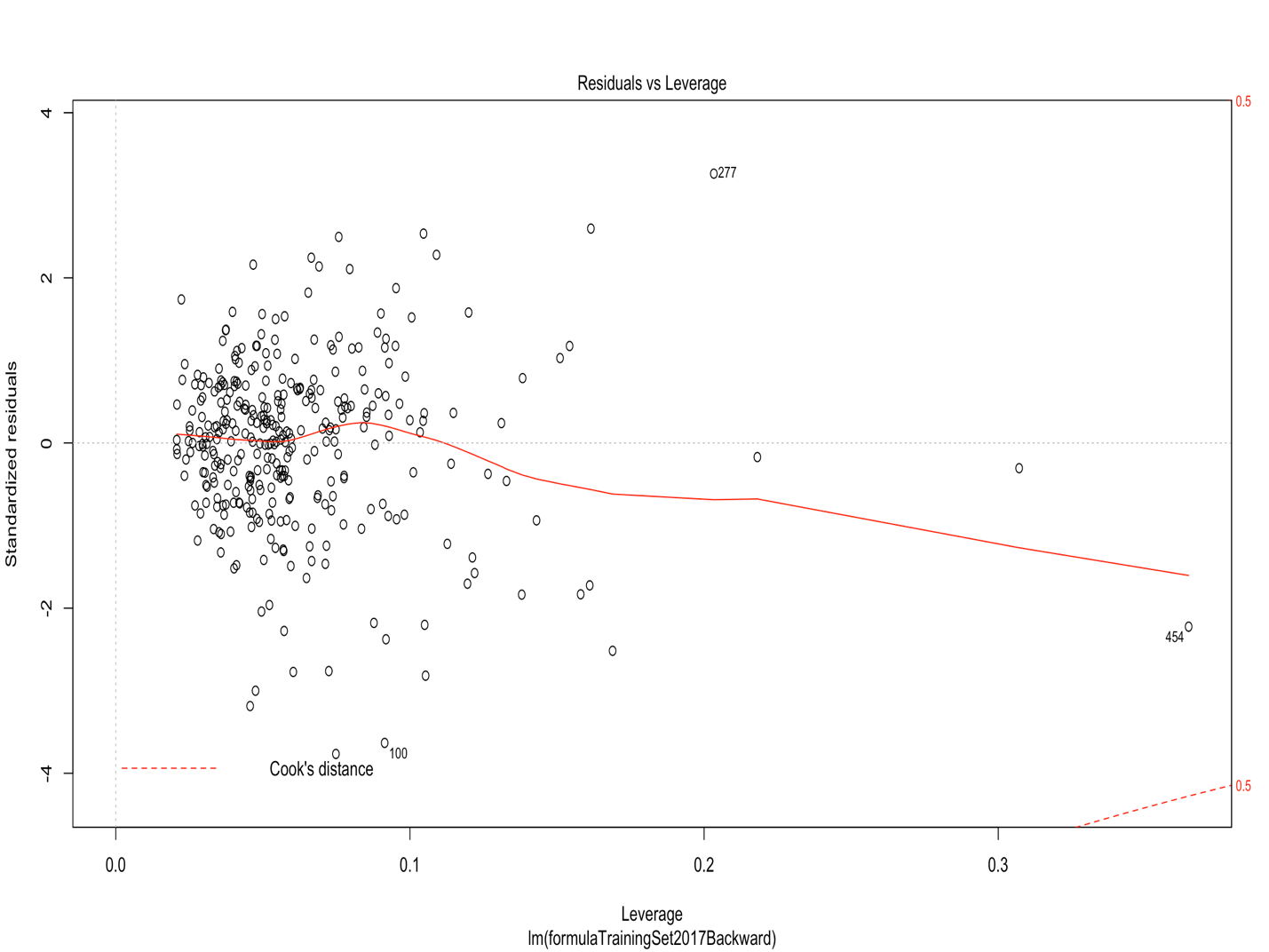
The residuals follow pretty closely to the middle line outside of some points at the top and bottom. This suggests that the residuals are normally distributed.

Scale Location



The same issue arises here as with the Residual v. Fitted test – there does appear to be some heteroskedasticity, but it is small enough that we do not have to worry about it.

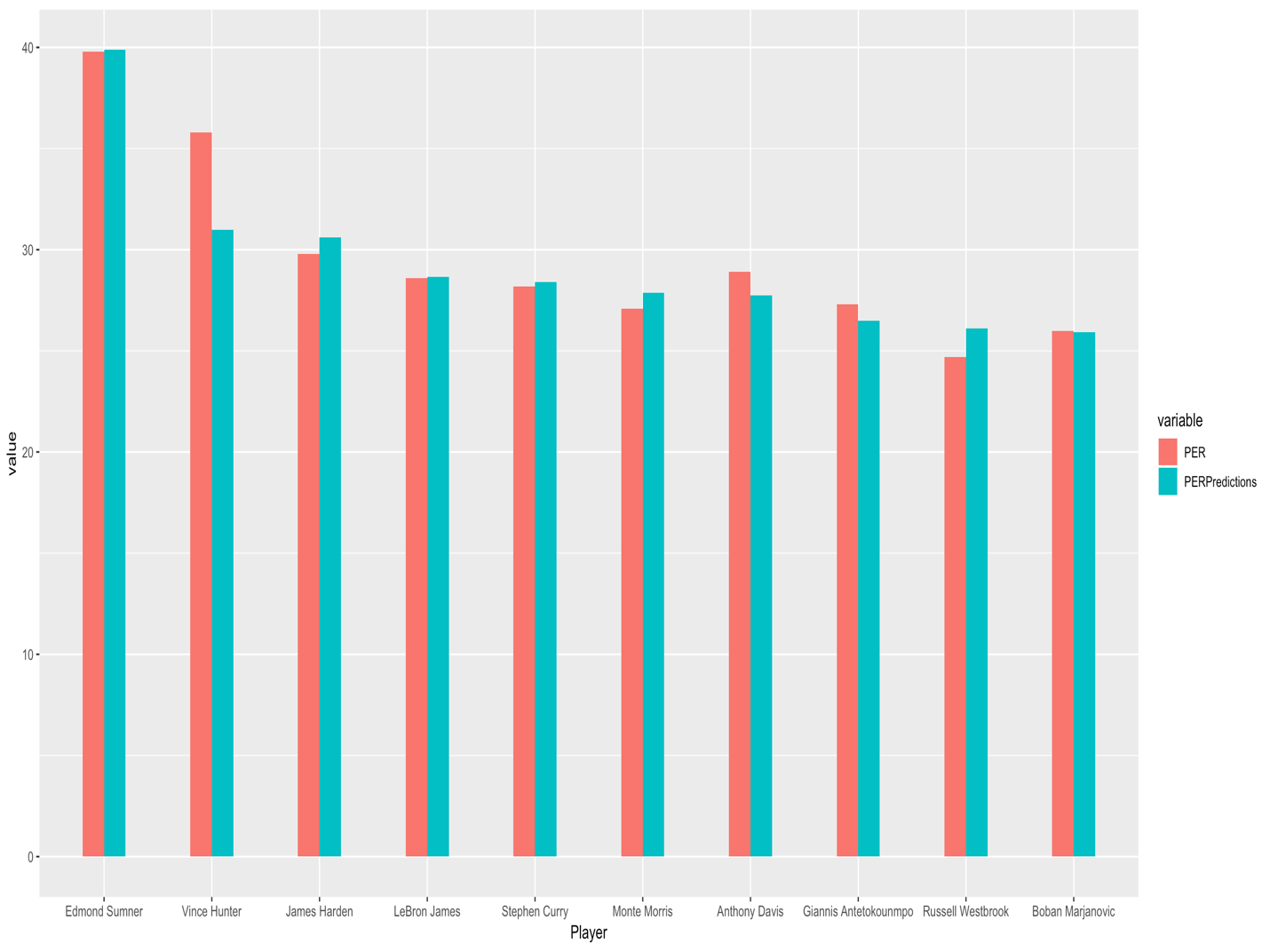
Residuals v. Leverage



There are no values that fall inside Cook’s distance, so there are no influential cases in the dataset.

**Now, what we’ve all been waiting for…**

All of this was the tune-up work to the real event: to see who the best players of the 2017-2018 season were based off of PER! These were highest predicted PERs based off the backward-selected Percentage variables:



These results are… shocking. Who are Edmond Summer, Vince Hunter and Monte Morris? These dudes are no names, so for them to be ranked in the same echelon as James Harden or Lebron James is totally unexpected. But the model isn’t incorrect; they logged high PERs during the 2017-2018 season, and the model represented that. The only explanation that I can think of for this is that PER is a highly flawed metric that can lump together below average players with superstars as long as the below average players log some numbers that PER favors.

Outside of this staggering discovery, the model does elucidate a few things. James Harden was the MVP of the 2017-2018 season, so for him to have the highest predicted PER of players who logged big minutes reaffirms the fact that PER does have some value as an advanced metric. Another interesting tidbit stems from the fact that Steph Curry, though flanked by three all-NBA players in Kevin Durant, Klay Thompson and Draymond Green, had the fifth-highest predicted PER during the 2017-2018 season. Since Durant joined the Warriors, people have debated who the best player on the Warriors was. The model says the answer is unquestionably Curry, at least for that season. Finally, even though Russell Westbrook’s style of play is deemed controversial by many NBA analysts and fans, the numbers don’t lie: my model predicted that he had the 9th highest PER of the 2017-2018 season.

**Conclusion and Improvements Down the Line:**

This was by no means a perfect model. In no particular order, some things that could’ve been done better are:

1. I only trained the model with 2016-2017 data. I could have trained it with more past seasons, which would have made the model more robust.
2. PER might not have been the best statistic to use. I had no idea how flawed of a metric it is. I tried to eliminate the threat of seldomly used players, but if I used another statistic, this problem could have been avoided entirely.
3. My approach to eliminate outliers who played less than 250 minutes was totally arbitrary. There was no scientific basis for this; it was solely a gut feeling. I could have found a more scientific way to choose with outliers to get rid of and which to keep.

Overall, this project helped me learn more about the field of data science. I still have a lot more to learn, but this was a good start, and I’m excited to continue learning about it.