Michael Luther

Professor Robert Bradley

Predictive Analytics and Forecasting

December 17, 2019

Midterm and Final Projects – Predicting Whether a NBA Shot is Made, and Predicting Points Per Game in a Time Series

**Problem**

The goal of this project was simply to predict whether a shot was made or not using data from the 2014-2015 NBA season. Models were created that took all players into account, as well as models that were particular to an individual player. As an extension for the final, I decided to take a small subset of players, convert the data into a time series, and predict points per game for those players using an ARIMA model. The original dataset was not a time series, and this seemed like a logical way to incorporate a newer model that we have learned in class.

Anyone watching a basketball game, be it a coach, player, or fan, thinks they know what a “good shot” is, and reacts accordingly when they see one. However, this “knowledge” can be quite flawed if it is not supported by analytics. Humans are not always rational, and what they believe to be a “good” or “bad” shot can be largely a result of biases and heuristics. By creating a model from the data to predict whether a shot was made or not, we can begin to infer what actually makes a “good” or “bad” shot with more confidence. This is not to say that a coach can become familiar with such a model and use it to make every decision as to how they try to position players offensively and defensively. As with all models formed through statistical analysis, it must be used with the right combination of intuition and feel for the current situation to make the best decisions. Nonetheless, a model that predicts whether a shot is made or not can be a useful starting point to making better in-game decisions.

**Significance**

The National Basketball Association is a multi-billion dollar organization with millions of fans worldwide. Like any smart firms in any industry, NBA teams are increasingly using innovative analytics to give themselves a competitive advantage. We can see the impact of analytics in every aspect of the game – from team building (such as Sam Hinkie’s infamous “Process” of tanking to acquire top draft picks), player rotations, shot selection (teams now shoot way more threes than ever, as well as emphasize driving to the basket and drawing fouls for free throws), and assuredly many other things. Each team has an incredible amount of information on where players shoot well from and those players’ tendencies, and use that information to try to get their players high-percentage shots as well as limit the opportunities of opposing players.

Predicting whether a shot is made or not has limited value on its face. When a player is going up for a shot, the coach can’t call a timeout and make an adjustment, and as far as I am aware, there is no way to live-bet whether a player will make a shot while in the act of shooting. The value in creating such a model comes from the interpretation of the coefficients of the independent variables. Understanding what factors impact players’ shooting in what way would be a great help to a coach for game planning and in-game play calling. For example, two players on a team might be similarly good at three-point shooting, but a model shows that one tends to be less bothered than the other when he is closely contested. Then, in a situation where it is unlikely that a player will get an uncontested shot, such as out of a timeout with one second left in the game, the coach knows that he should give the shot to the player who is less bothered when closely contested.

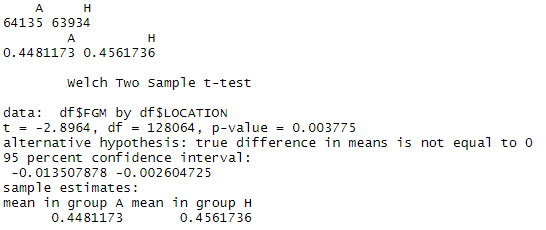
Predicting points per game has less of a broad appeal than predicting whether a shot was made, but it is still useful nonetheless. Sports books need to make such predictions to offer prop bets, and daily fantasy sites such as DraftKings and Fanduel need to make the same such predictions to adequately price players. Even though an accurate model would really only have use for sports books and gamblers, that is not to say it is insignificant – Americans spend billions of dollars a year betting on sports, and both DraftKings and Fanduel are valued between $250-$300 million. Having accurate models is not only vital to those companies themselves, but to any other economies that have been created as a result of them.

**Data**

*Initial Analysis and Exploration*

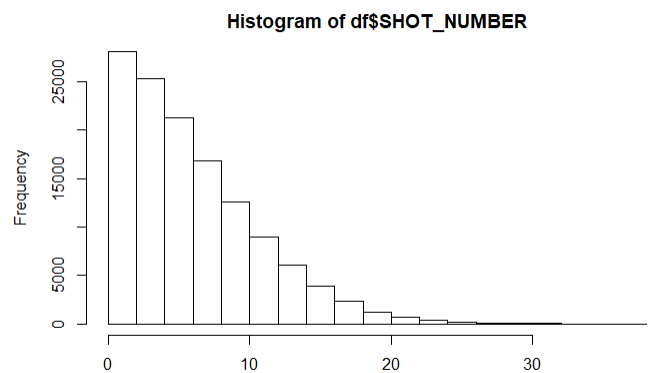
I downloaded the dataset for this project from Kaggle, which was scraped from the NBA’s REST API and uploaded by username DanB.[[1]](#footnote-1) The dataset is quite extensive – it contains shot data from 904 unique games in the 2014-2015 season, which is almost three-quarters of all games played that season. The original dataset contained 128,069 total observations, and 21 columns of variables. All records are from the perspective of the shooter at the time of his shot. The following is a summary of the variables.

* Game ID: the unique ID number for each game. There were 904 unique values for this variable. Since this variable is not useful for predicting the outcome of a shot, it will not be discussed any further.
* Matchup: This was formatted as “date - away team @ home team”. For example, *MAR 04, 2015 - CHA @ BKN*. This variable will be discussed further later when I talk about feature creation.
* Location: Whether the shooter’s team was home or away. As this is a categorical variable for a binary outcome, visuals such as plots or bar charts are not useful. For binary outcomes, boxplots aren’t useful either, as they compare the spread of the data – but when the values are exclusively 0 and 1, this is of no value. For variables such as this, a mean table was created to compare the mean outcome (field goal percentage) for each level of the factor, and a t-test or ANOVA was conducted to test whether the field goal percentage in each group was significantly different, or whether the mean value of that variable was different for makes and misses. However, it should be noted since it is a binary 0 or 1 outcome for a huge sample, the standard error will be small, and by definition these differences almost have to be significantly different. Below, this can be seen.

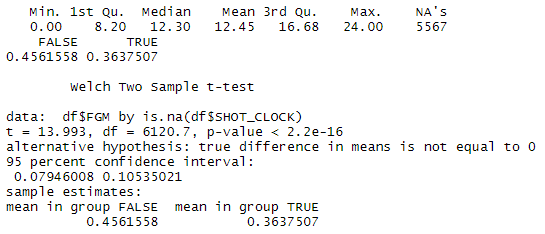


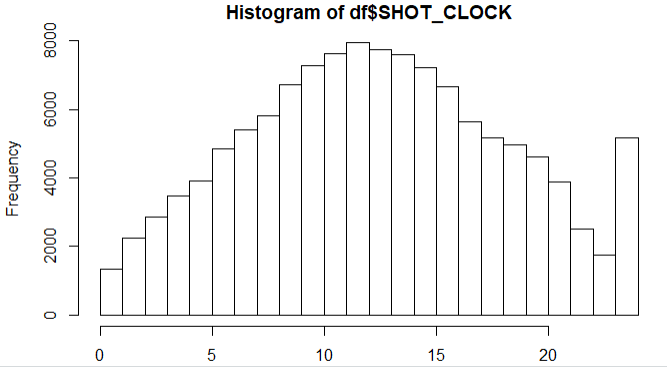
* W: Whether the shooter’s team won or loss. The shooting percentage in wins was almost exactly 5% higher than it was for losses. However, this variable is endogenous to the model. We know that making a high percentage of shots will likely lead to a win for a team, and we cannot have an independent variable that is determined by our dependent variable. As such, it will not be used or discussed further.
* Final Margin: for the same reasons as W, this will not be discussed any further.
* Shot Number: the shot number of the shot for that player. For example, a value of 3 would indicate that it was the player’s third shot in the game. There were no values that indicated anything was wrong or miscoded. Shot Number was heavily right-skewed, with most shots recorded being within that player’s first few shots of the game.



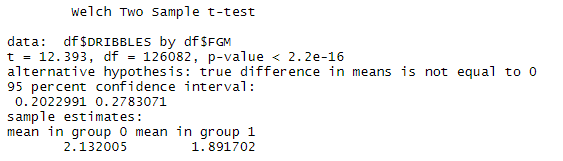


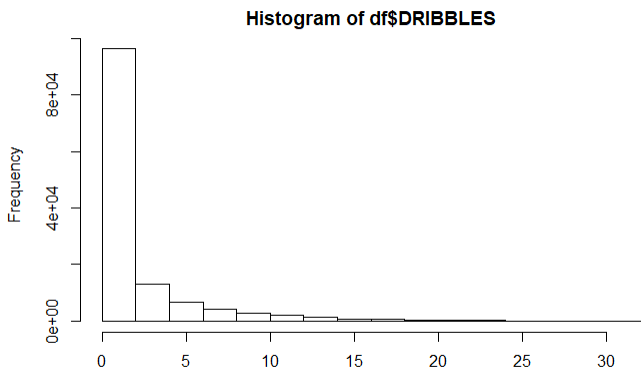
* Period: the quarter the shot was taken in. This was combined with game clock to create a new variable for total minutes elapsed.
* Game Clock: how much time was on the game clock when the shot was taken. This was combined with period to create a new variable for total minutes elapsed.
* Shot Clock: how much time was left on the shot clock when the shot was taken. There were 5,567 missing values that needed to be investigated. The first step in handling missing values is always to check to see if they are in any way predictive. Oddly enough, the field goal percentage was almost 10% lower for the shots recorded with no value for the shot clock variable. However, there is no plausible reason to believe that these missing values are indicative of anything that would actually cause a lower shooting percentage – a field goal can’t be attempted in a game while the shot clock is not running, and this dataset does not contain free throws – only field goals. Therefore, since the data is relatively normally distributed, I decided to impute with the mean value for shot clock. There is a spike of shots taken towards the end of the shot clock, but this makes sense; shots are often hoisted up right at the buzzer. One other observation of note is that the max value is 24. Physically, this would be impossible. The shot clock is 24 seconds long, so releasing the ball at 24 seconds would mean the shot wouldn’t count. However, this column has some values that are recorded up to 4 decimal places. I decided to keep this rather than impute it and consider it rounding. Additionally, there was only one instance of this out of 128,069 observations, so even if it was an improper value its effect was negligible.



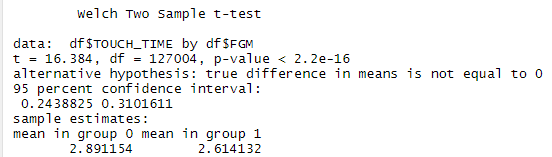


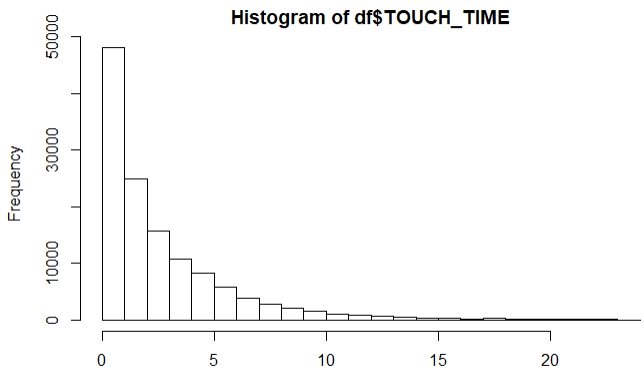
* Dribbles: how many dribbles were taken before the shot. This was heavily right skewed, and shots that were made had statistically significant fewer dribbles.



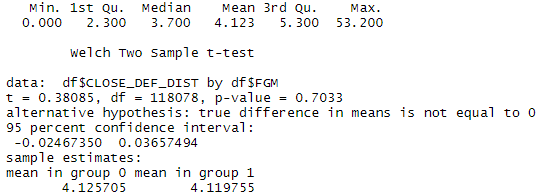


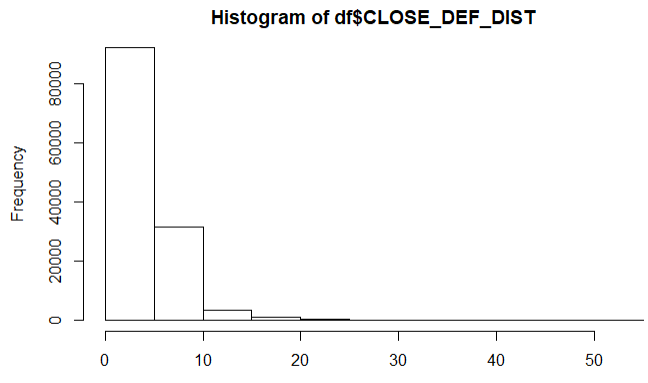
* Touch Time: how long the shooter had the ball before shooting. Similar to dribbles, it is heavily right-skewed, and made shots have less touch time than misses do. As one might suspect, it is also highly correlated with dribbles, with a correlation of .914. Touch time was the most problematic of all the independent variables, with some values that were negative as well as greater than 24. Most of these were negative values, and those I assumed they were accidentally put in with a negative sign, so I made them positive. All values above 24 had a decimal in them. Therefore, I assumed that the values over 24 were coded with the decimal sign in the incorrect place, so I divided those values by 10. Finally, it is worth noting that there was also one record with a touch time of 24 – this matches the one shot clock instance of 24 from earlier.





* Shot Distance: how far away the shooter was from the basket, in feet. I will discuss this more in feature creation, because I used it in combination with points type to make a multi-level factor variable.
* Points Type: two or three pointer. It was combined with shot distance as previously mentioned.
* Shot Result: made or missed. This was a redundant column, as I used the column FGM to represent the dependent variable, which was coded in 0s and 1s.
* Closest Defender: the name of the closest defending player. There were 473 unique defenders, so looking into statistics for each one individually was not a useful exercise.
* Closest Defender Player ID: unique ID number for closest defending player. This was only useful for checking for any players who may have had their names spelled differently for different observations. The numbers matched up.
* Closest Defender Distance: how close the closest defender was to the shooter. No guidance was offered on the site as to whether this was in feet or inches. The median seems low, as that’s just under four feet of space for a shot, but the maximum seems high – although that could just be my own availability bias. Since the NBA offers current defender stats based on feet, I assumed it was feet and left it as is. Similar to most of our numerical variables, this is heavily right skewed. Oddly enough, though, there is no significant difference in closest defender distance for makes and misses – perhaps that is because shots taken closer to the basket are more likely to be made but also more likely to be closely contested.

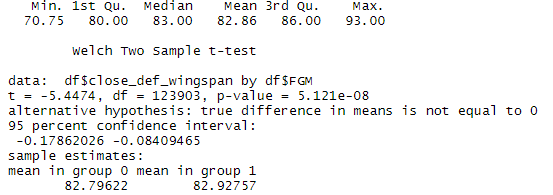


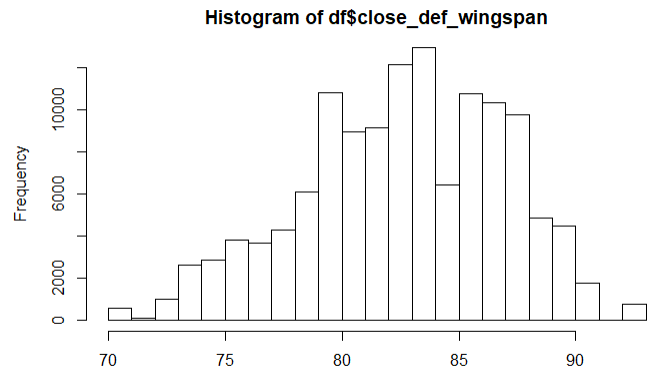


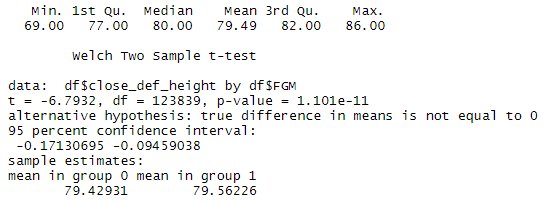
* Field Goal Made: binary 0 (missed) or 1 (made). Slightly more than 45% of observations were made field goals.
* Points Resulting from Shot: this is either 0, 2, or 3. Given what other variables we are using, this has no use and will not be discussed further.
* Player Name: the name of the shooter. There were 281 unique shooters.
* Player\_id: unique ID number for the shooter. This is not useful.

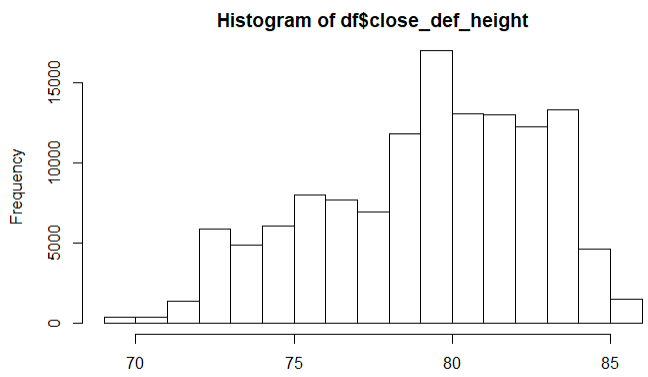
*Feature Creation*

As there were two separate columns with factor variables with hundreds of levels – shooter name and closest defender name – I wanted to find some sort of substitute for at least one of them so that a model trained on all the data was less costly to run and readable. Therefore, I manually searched for and found almost all of the wingspans of the defenders and all player heights, and recorded them in inches. For those players where a wingspan could not be found, height was used as a substitute. It is not a perfect substitute, as most NBA players have wingspans that are at least slightly longer than they are tall, but it is better than imputing with the mean or median value. It is important to note, however, that the sport of basketball is notorious for exaggerating the heights of their players, and I would not be surprised if wingspan was similarly exaggerated, therefore biasing the estimates on these coefficients upwards. Additionally, the NBA does not have one sole source where player heights and wingspans can be pulled from. With that, it should be understood that some variance is being introduced by pulling from these separate sources. Both are slightly left-skewed and are highly correlated, with a correlation of .826.





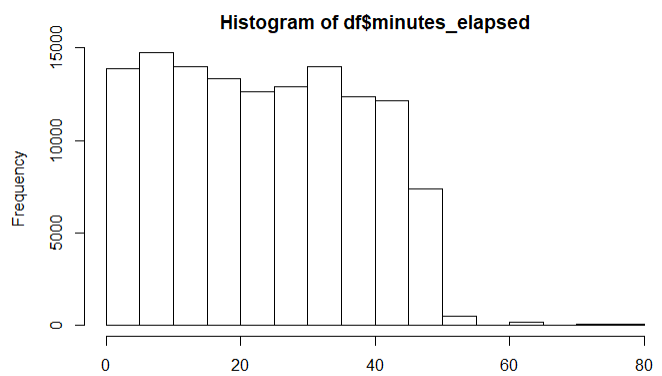




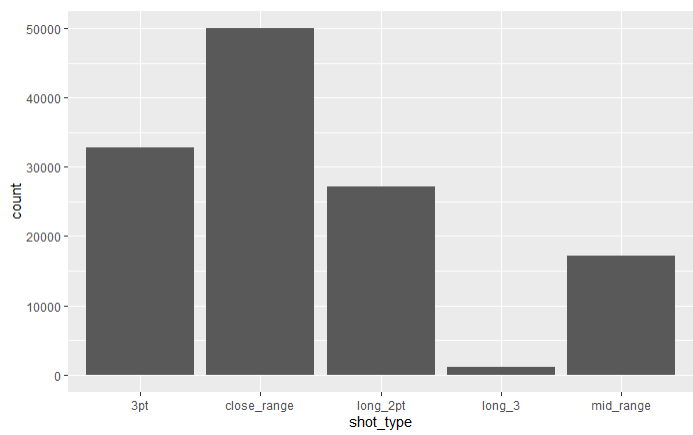
I will now describe the other new variables created:

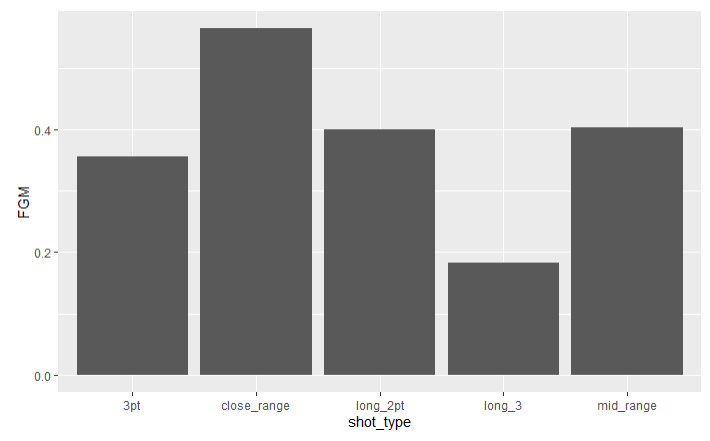
* Arena: This was taken by splitting the “Matchup” column to take the value of the home team.
* Month: This was also created by splitting “Matchup”
* Allstar: I looked up when the all-star game was in 2015, and created a 2-level factor for before and after that date.
* Minutes Elapsed: There are 12 minutes in an NBA quarter, so this was calculated by multiplying 12 \* quarter and adding the game clock. We see that shots are taken consistently throughout regulation.





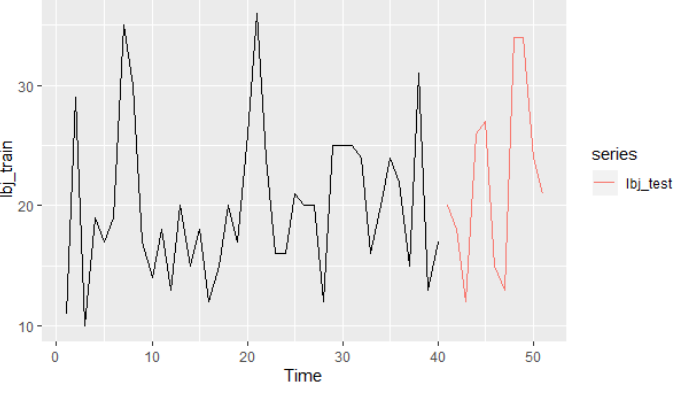
* Shot Type: This is a combination of points type and distance. I broke it into five categories: close range (0-8 feet), mid-range (8-15 feet), long two (15-23.5 feet AND 2pt; OR 15-22 feet AND 2 pt), three point, and long three (anything more than 5 feet beyond the furthest part of the line, so 28.5 feet). The first graph shows count by shot type, and the second shows field goal percentage by shot type.



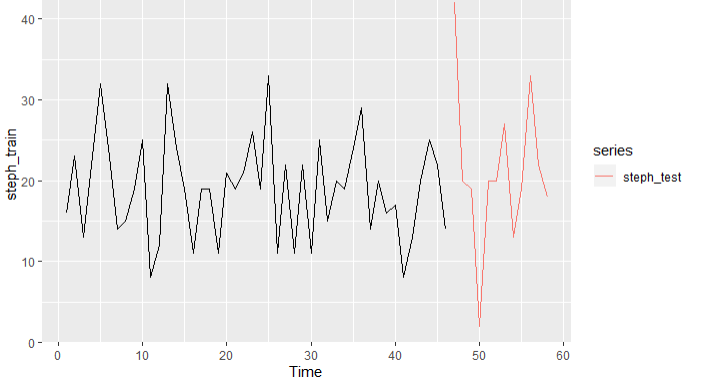


To create a time series for each player, I had to create three separate dataframes by subsetting each player from the original dataset. The dates then had to be changed into a date format that R would recognize, using “as.Date” and the appropriate format, given how the dates were written as a string in the data. Then, I converted each to a time series using “ts(mydata)”, and got rid of all other values by just selecting points per game from the original subset. Each player’s time series was then split into an approximately 80/20 train/test split. Here’s what each of their time series looked like:

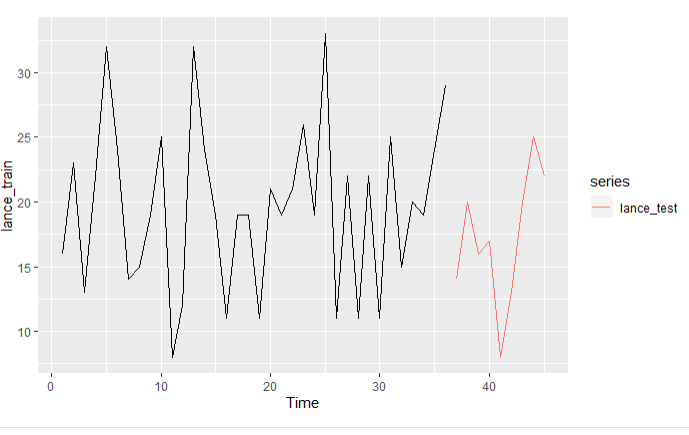
LeBron



Steph



Lance:



Notice that points per game for each player appears stationary and largely resembles white noise. There is no trend and no seasonality, just apparently random fluctuations. This dictates the model selection to be an ARIMA, because we know that ARIMA models need stationary data. One other important thing to note is that this data appears incomplete. We knew we were missing data from about one-fourth of the games that season from our initial analysis when predicting whether a shot was made or not. However, by checking the means of the training data for each player, we can see that we are clearly also missing data from some of the individual games too. LeBron’s mean was 19.925 points per game (he averaged 25.3 that season); Steph’s was 19.02 (he averaged 23.8 that season), his first MVP campaign; Lance’s mean was 19.61 (he only averaged 8.2 that season). This data was made available on Kaggle from someone who scraped the NBA’s API – clearly, some things went wrong with the scraping. However, the data still provides a useful exercise in building ARIMA models.

**Literature Review** FIVE examples from PEER REVIEWED JOURNALS of how the types of models you selected were used in similar situations

As previously mentioned, analytics and sports are heavily intertwined, to the point where one almost can’t watch a game or consume any sports media without hearing or seeing the word “analytics”. As such, plenty has been written on various applications of statistical analysis to predict a plethora of outcomes in the sports world.

A paper written by Torin K. Clark, Aaron W. Johnson, and Alexander J. Stimpson that was presented at the 2013 MIT Sloan Sports Analytics Conference uses the same exact kind of model – a binary logistic regression model – to predict whether a NFL field goal was made or not. Their motivations for predicting whether a field goal was made or not are identical to mine for predicting whether a shot was made or not, “By more accurately predicting the outcome of field goal attempts, coaches can make better ingame decisions and fans can gain a greater understanding of kicker ability.”[[2]](#footnote-2)

Moving back to the world of basketball, Paul Kvam and Joel S. Sokol developed a combined logistic regression/Markov Chain (LRMC) model to predict the winners of the NCAA Men’s Tournament games, using only basic input data. They specify a Markov Chain model for ranking purposes, and then utilize logistic models to calculate probabilities. Their model was able to outperform any other publicly available predictor of the games, such as the expected winners based on tournament seeding, ESPN picks, RPI, or Vegas favorites.[[3]](#footnote-3) Similarly, Abhishek Naik, Shivanee Pawar, Minakshee Naik, and Sahil Mulani used logistic regression as one method to predict whether a cricket batter would be successful or not.[[4]](#footnote-4)

Extending even further beyond a binary logistic model, Douwe Buursma used several classifiers to predict the outcome of Dutch soccer matches into three categories: home win, draw, and away win. Two of those classifiers use logistic regression to predict the correct class (outcome): the MultiClassClassifier, and LogitBoost.[[5]](#footnote-5) These are not necessary for binary cases like predicting whether a shot was made or not, but are still interesting extensions of logistic regression. Finally, a potential future extension of my current work could be to do something similar to what Michael Purucker did by using various neural network strategies to predict the outcome of NFL games.[[6]](#footnote-6)

**Model Building and Selection**

After splitting the data into an 80:20 train:test ratio, a multitude of models were built for all players. Two specific players were also selected and the same models were trained on them. Here is the same general approach for the whole set and the two individuals, LeBron James and Steph Curry:

* Fit a model including all useable variables
* Fit a model using all useable variables that includes interaction terms between related variables
* Use cross validation to refit the second model that includes the interaction terms.
* Fit a lasso model. This was done as a substitute for the best subset approach, because the best subset approach would not work for a logistic model on this data. I wanted to see what variables would be shrunk to zero and which were kept.
* Fit a model with numeric variables transformed. I used Tukey transformations on each of the numeric variables, rather than using a Box-Cox on the whole model.

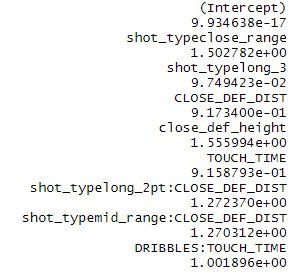
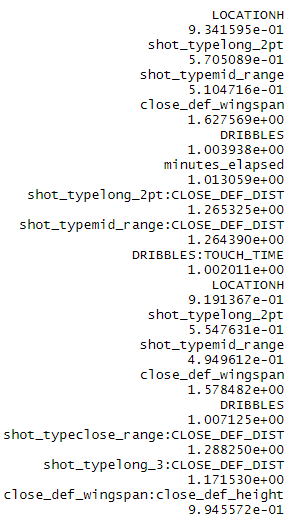
Ultimately, none of these models gave me a satisfying answer. At best, the accuracy was in the low 60% range, and the coefficients on some of the variables didn’t make intutitive sense. That was unacceptable, as the main purpose in predicting whether a shot was made or not was to generate useful coefficients on the factors affecting a player’s shooting in order to guide coaching strategy. I therefore decided to just try a couple of simpler models based on what I thought would be the most important variables, and ultimately found one that had the best accuracy for LeBron, Steph, and all players, as well as meaningful coefficients that made intuitive sense. The model with Tukey-transformed variables was similar in accuracy, but is much more complex. When all else is equal, the simple model always wins.

I will now detail four of the models generated: the model that worked best for LeBron and Steph (and all players, but I won’t focus in on that one due to the 281 level factor for the shooter), and two of the worse performing ones, LeBron’s lasso and Steph’s full model.

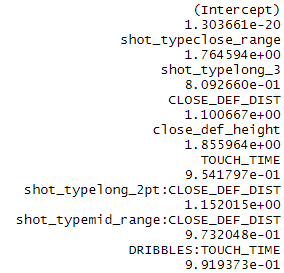
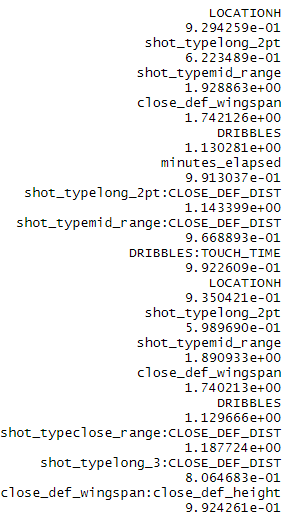
*The winning model*

The formula for this model is FGM ~ LOCATION + shot\_type\*CLOSE\_DEF\_DIST + close\_def\_wingspan\*close\_def\_height + DRIBBLES\*TOUCH\_TIME.

For logit models, odds ratios can be generated by exponentiation of the coefficients. These can be interpreted as a multiplier for the likelihood of a “success” outcome for the dependent variable associated with a one-unit increase in the independent variable, holding all else constant. Here are LeBron’s odds ratios for this model:

And Steph’s odds ratios for the same model:

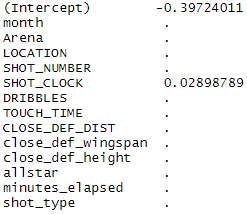
We can see that Steph is .809266 times as likely to make a long three as he is a regular three, and LeBron is .09749423 times as likely – that makes sense, given that it should definitely be less likely for both, and Steph is well-known for his ability to hit shots from way beyond the three-point line. No coefficients stick out as horribly weird. Even though this model states that both are more likely to hit shots as their closest defender’s wingspan increases, it is not high enough to cause suspicion. Both are great players who excel at finishing around the basket near taller defenders. Finally, I would like to draw attention to the interaction terms selected. I chose to create interaction terms for variables that had high correlations, like dribbles and touch time, and defender wingspan and height (please see the appendix for a VIF chart of the variables in the full model for further detail). I also included an interaction for shot type and closest defender distance, because as I just discussed, how closely guarded a player is when taking a shot likely has a different impact on their shooting depending on where they are. LeBron is famous for being able to drive to the basket and finish at will, with defenders near the hoop bouncing off of him. Having a defender right on top of him near the basket does not impact him as much as having someone closely contest a three-pointer. Before moving on, let’s inspect how accurate the models were with a confusion matrix for each player. LeBron is on the left; Steph is on the right.



The simplest and important measure is accuracy: the actual misses correctly predicted as misses plus the amount of actual makes predicted as makes, divided by the total number of predictions. For LeBron, this is (68+61)/(68+61+40+35) = 63.24% accuracy. Steph’s accuracy is 65.38%.

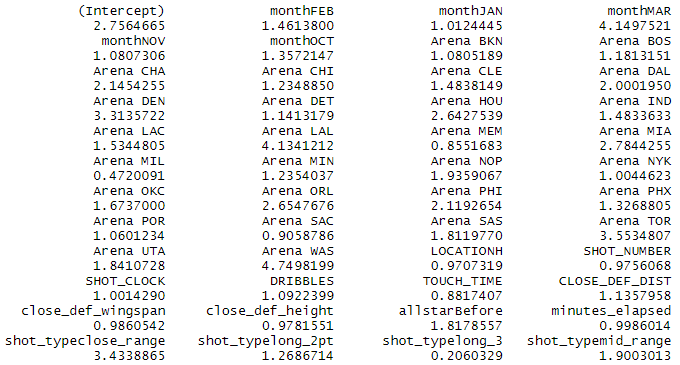
*Some of the Losing Models*

I would like to discuss briefly two models that were not selected, the lasso and the full model. I had hoped that the lasso would act as a best subset selector, as lasso models shrink all variables deemed to be unimportant to zero. However, for LeBron, it ended up eliminating all variables other than shot clock:



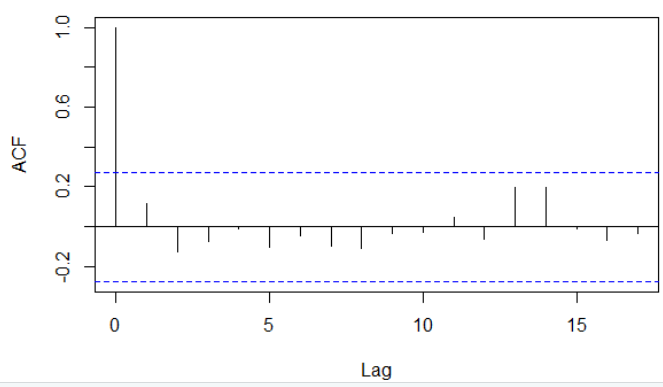
For one, this doesn’t make sense. Shot clock intuitively seems like it may have some sort of impact, but it certainly is not the most important variable. Second, this model had the worst accuracy at 56.37% - not much better than a guess.

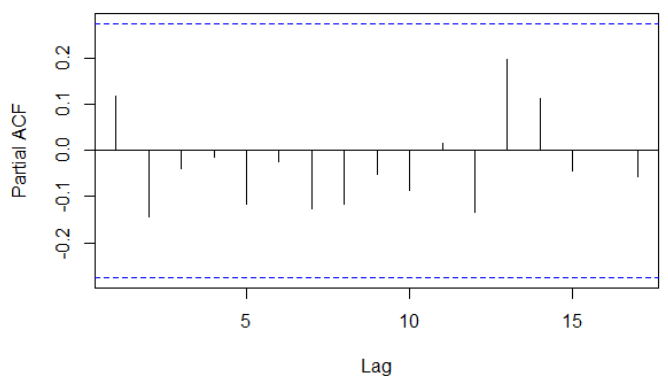
Finally, let’s look at the odds ratios for Steph’s full model. It proved to be 58.24% accurate on the test data. There is some wide variance in Arena and month – so much so that it is hard to fully trust this model. The model says that Steph is over 4 times as likely to make a shot in March as he is in December. It says he is almost five times as likely to make a shot in Washington as he is in Atlanta. There are a few other relatively absurd odds ratios there as well. Some of this might be due to sample size, because teams will only play away games against other teams out of conference once or twice a year, or it could be due to other factors not captured in our data. Regardless, such a model cannot be used.



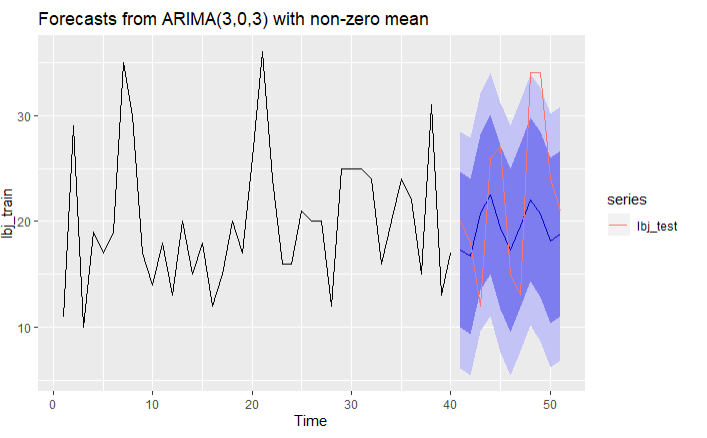
For the points per games models, I decided to continue with LeBron and Steph, but added a third player who would have more variance in the amount of points they scored per game. I ended up picking Lance Stephenson, who was capable of scoring 20 plus points a game, but who was also prone to making a lot of dumb mistakes and scoring a lot less (plus, I thought it would be fun to reunite Lance and LeBron given their history). Since there are only three players, I will walk through each one-by-one. The general approach was to check ACF and PACF plots to see what insight they may have offered, see what kind of model auto.arima selected and how accurate it is on the test data, and then play around with the p and q orders for the amount of lags in the autoregressive and moving average parts of the model based on what we see in the ACF and PACF plots. Since this data is stationary and does not have any seasonality, no differencing or seasonal components were used. The auto.arima selected a (0,0,0) model for each player, which is just flat projection of their average points per game in the training data. This was never the best model for any player, and will therefore not be discussed further.

LeBron

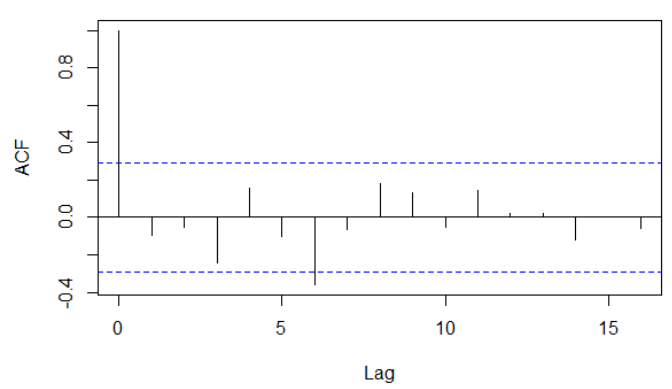
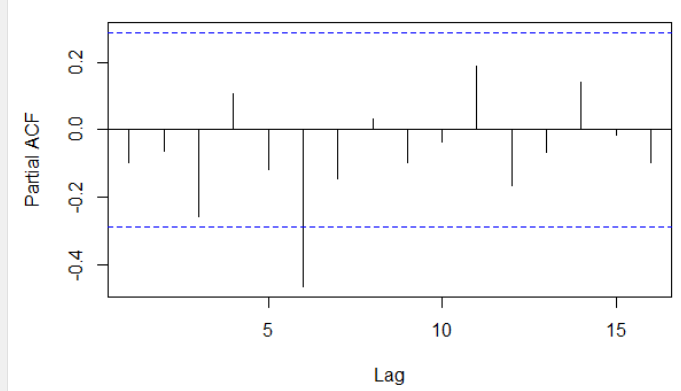




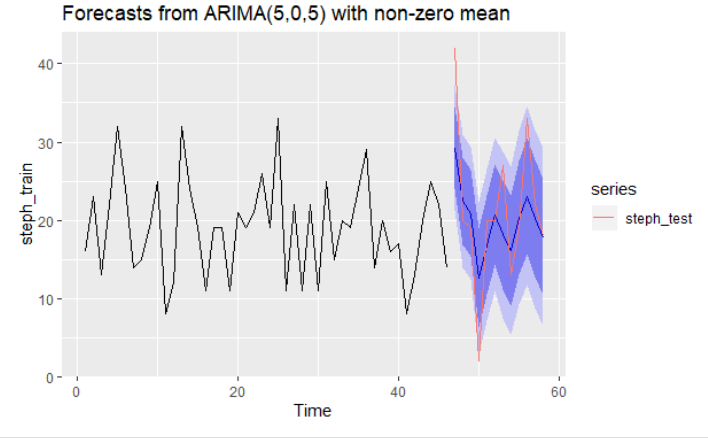
We see one significant lag in the ACF, and no significant partial autocorrelations. This means that the first lag is the only significant one. Given that I decided to start with an ARIMA(1,0,0): one lag value (p =1) for the autoregressive part of the model, and no error terms (q = 0) from the moving average part of the model (the moving average is a model that is generated through optimization to minimize error, those errors are then used as the parameters in the moving average part of the ARIMA model). I then tried an ARIMA(1,0,1), ARIMA(2,0,0), ARIMA (2,0,1) and so on with different orders of p and q until the model consistently stopped improving. The best model turned out to be ARIMA(3,0,3), with a MAE of 6.00.



Steph

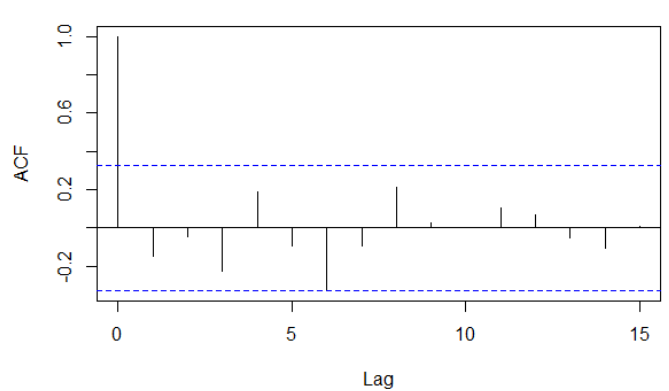
 

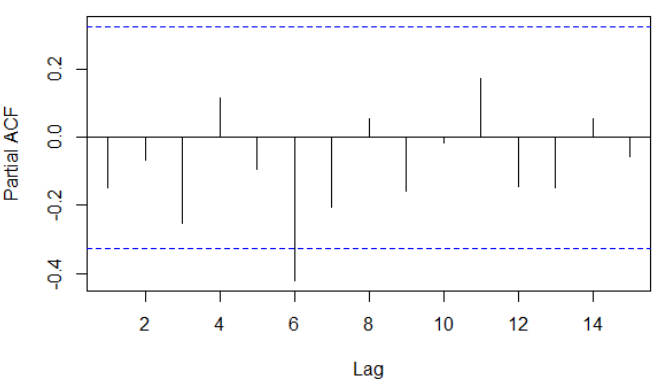
We see here significant lags at lags one and six, and that significance persists in lag six with the partial autocorrelation graph. This means that even with the correlation from previous lags controlled for, lag six is still significantly correlated to the “present” observation. Therefore, it is worth exploring orders of p and q up to at least lag six. Following the same process as before, I found the best model for Steph to be ARIMA(5,0,5) with a MAE of 4.73. I was expecting a (6,0,6) to perform better, but it actually got slightly worse. We can see how well the forecast actually appears to fit the line below.



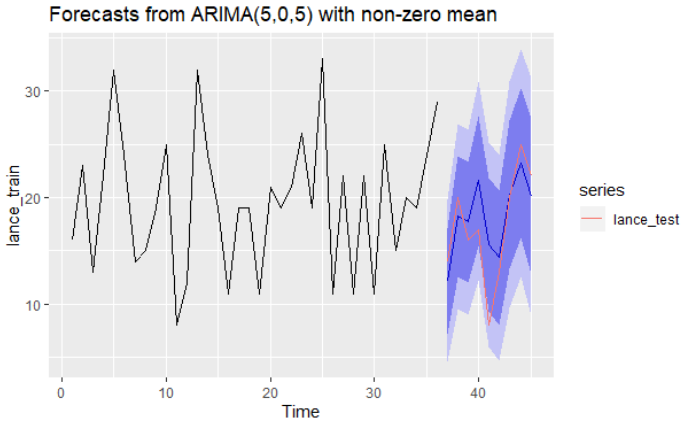
Lance

Starting with the ACF and PACF plots again, we see the same pattern that we saw with Steph.





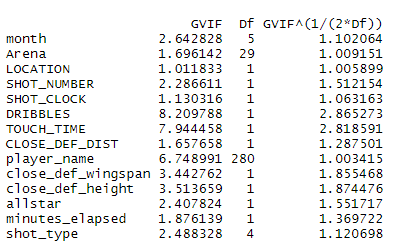
I therefore followed the same exact logic, and found the same exact model, ARIMA(5,0,5) to be the best, with a MAE on the test set of 2.53.



Overall, I would say these are excellent results for all three players, but particularly for Steph and Lance. This model could definitely be used by books and bettors to make their predictions, and I am confident that they’d be happy with the results.

**Appendix**

VIF (variance inflation factor) is a measure of how easily predicted an independent variable is form other predictors. A higher value means it is more easily predicted; generally a value of above 5 or 10 is considered very strong. The highest values shown here all have some sort of interaction term with another variable in my chosen model.



1. <https://www.kaggle.com/dansbecker/nba-shot-logs/discussion/24907> [↑](#footnote-ref-1)
2. Clark, Johnson, and Stimpson. *Going for Three: Predicting the Likelihood of Field Goal Success with Logistic Regression.* <http://www.sloansportsconference.com/wp-content/uploads/2013/Going%20for%20Three%20Predicting%20the%20Likelihood%20of%20Field%20Goal%20Success%20with%20Logistic%20Regression.pdf> [↑](#footnote-ref-2)
3. Kvam and Sokol. *A Logistic Regression/Markov Chain Model For NCAA Basketball.* <https://www2.isye.gatech.edu/~jsokol/ncaa.pdf> [↑](#footnote-ref-3)
4. Naik, Pawar, Naik, Mulani. *Winning Prediction Analysis in One-Day-International (ODI) Cricket Using Machine Learning Techniques.* Available for download at <https://aspirepublishers.com/index.php/ijetcs/article/view/186> [↑](#footnote-ref-4)
5. Buursma. *Predicting sports events from past results, Towards effective betting on football matches*. <https://pdfs.semanticscholar.org/5e22/c4362df3b0accbe04517c41848a2b229efd1.pdf> [↑](#footnote-ref-5)
6. Purucker. *Neural Network Quarterbacking*. <https://ieeexplore.ieee.org/document/535226>.

   *Separately, I would also like to cite* [*https://www.sciencedirect.com/science/article/pii/S2210832717301485*](https://www.sciencedirect.com/science/article/pii/S2210832717301485)*, as I found some of these articles through citations from this.*  [↑](#footnote-ref-6)