**STAT 6021 Final Project Report**

# Using Logistic Regression to Predict the Outcome of a Field Goal Attempt in the NBA

*Brian Sachtjen (bws7vs), Yuanbo Wang (yw4rd), Dylan Greenleaf (djg3cg), Dongheon Lee (ddl3vd)*

## Executive Summary

The main goal of our project was to fit a model that could accurately predict the expected value of a field goal attempt in the NBA. Our objective was to develop a statistical model that could predict the result of the field goal attempt given certain characteristics about the shot. In order to achieve this, we employed a logistic regression model with multiple regressors while keeping the result of the field goal attempt (1 or 0) as the response variable. Our model could impact the NBA scene both in terms of team performance evaluation and player performance evaluation. In addition, it creates an opportunity for coaches to improve their coaching strategies.

The two sets of data we used for our model both came from the NBA Statistics website. After acquiring the data and going through the necessary steps of data wrangling, we used R to fit the logistic regression model to the data. Lastly, we used the model to find the number of points a player is expected to make for a given attempt, which we then compared with the actual points from the data.

We first used our model to evaluate the performance of each team in the NBA by comparing their actual points per shot to the expected points per shot. It was found that Clippers and Thunders were outperforming the point per shot values that our model predicted, whereas the Lakers and Hornets were underperforming with respect to the same metric. Then, we used the model to evaluate the performance of each player in the NBA by the same metric. Kyle Korver and Jonas Valanciunas were the best shooters, whereas Kobe Bryant was the single worst shooter according to our index. Lastly, our model could also be utilized to impact coaching strategies. Players shot with the highest accuracy when they have been in the game for about five minutes, and the worst accuracy after about 10 minutes - coaches are recommended to plan their substitutions accordingly to maximize the player's performance.

## Data Scraping, Ingestion, and Manipulation

### *Collecting the Data*

All data used in this project was scraped from stats.nba.com using simple python scripts and the *requests* package. The first dataset comes from the player data repository. Specifically, we collected data from the shot log for each player in the NBA for the 2013-2014 and 2014-2015 seasons. An example of this can be found at http://stats.nba.com/player/#!/201939/tracking/shotslogs/, which contains all shots taken by living legend Stephen Curry in the specified NBA season, along with a host of other information about the shot, such as number of dribbles taken before the shot, amount of time left on the shot clock, closest defender distance, etc. The second dataset that was scraped consisted of the play by play game logs that are created for each game in 2014. These play by play logs contain time-stamped records of everything that happened during that game, including dead-ball events like substitutions. An example can be found at http://stats.nba.com/game/#!/0021400001/playbyplay/. As we will describe in the following sections, we used this data to create new information about every shot that was taken, such as how long the player had been in the game when he took the shot.

A smaller data set that we did not realize we would want until we were knee-deep in our project was the position of each player. We took a very non-technical approach in acquiring this data and downloaded Excel files directly from the NBA stats page. This was not a viable method for obtaining our other data sets.

### *Cleaning and Merging the Data Sets*

After collecting the data, we then set out to merge the two resulting data sets. The game log data set (play-by-play) had more than twice as many observations as the shot log data set due to the fact that it records rebounds, substitutions, and other information not related to shots. To merge these 2 data sets we needed a set of feature values that would uniquely identify each observation in both data sets. While there are many ways to do this, we settled on the gameID, playerID, time left in the game, and whether or not the observation was a shot. GameID and playerID were existing values in both data sets but we had to create variables to keep track of how much time was left in the game for each observation and (for the game log data) to indicate whether or not a given observation is about a shot that was taken.

To determine whether or not a given observation in the game log data set pertained to a shot being taken, the “home description” and “visitor description” variables were searched to determine whether or not their text contained “shot”, “layup”, or “dunk.” If one of the 3 terms was found in either of the descriptions for a given observation, a new variable that we created called “shot_or_not” was set equal to 1; if neither of the descriptions contained one of these words for a given observation, the variable was set to 0. Every observation in the shot log was given a value of 1 for “shot_or_not”, because that dataset only represented shots taken. After running this process, the number of observations labeled as shots in our game log data set roughly matched the number of observations in our shot log data set, which reassured us that we would be able to match most of the shots together.

Next, we tackled the issue of determining how much time was left in the game for each observation in each data set. Each data set had variables indicating the quarter and the time left in the quarter for each observation. The total time left in the game was determined by taking these 2 values as inputs into a function that output a value between 0 to 2880 seconds (the total number of seconds in one NBA game). After performing this task and comparing the 2 data sets, we noticed that there were discrepancies between the timestamps for corresponding observations in the 2 data sets, possibly due to the fact that the shot log data is generated from a computerized video system, while the game log data requires human input. Thus, a given shot in the shot log might have a time left value that is 2 or 3 seconds greater or less than its corresponding observation in the game log. Because we needed the time left value to uniquely identify each observation, it was imperative that we develop a method that would force these values to match. To achieve this, we took the observations pertaining to shots in the game log data set, made a list of the timestamp values for each game, and then coerced the timestamp values of the shot log data to match these, based on which time value was closest in absolute value (and not already assigned).

After all this, we were able to merge the data files on gameID, playerID, time left, and whether or not it was a shot (in regards to the game log data set). Roughly 97% of all shots were correctly matched between the two datasets, which we thought was pretty good; the remaining 3% were dropped.

Finally, there was also a little bit of cleanup involved with the metrics that the NBA's stats website provided us. For example, sometimes there was no value for "shot clock" or the value of "touch time" was less than zero or greater than 24 (theoretically impossible). We dropped these observations since they were data collection errors; however, it was a very small number of shots (roughly 300 out of over 169,000) that we had to drop.

### *Creating Additional Metrics*

Once our two data sets were merged, we set out to create additional metrics that we thought would be interesting to include as features in our regression model. The metrics we thought would be the most interesting to incorporate into our model were the player's field goal percentage from 2013, the closest defender's allowed field goal percentage from 2013, the amount of time since the player was substituted into the game, and the position of each player. To calculate each player's field goal percentage from 2013, the shot log data was scraped from the NBA stats just as the other data was. The data was grouped by playerID, which allowed a simple computation to determine their field goal percentage for 2013. A similar approach was taken to determine each player's allowed field goal percentage when they were the closest defender for an observation. Determining the amount of time elapsed between when a player took a shot and when he was subbed in was more difficult. We were able to find observations in the shot log that represented a substitution by using the value in the "event type" column and assigned the players involved in the substitutions to 2 new columns, sub_in and sub_out. For each shot we then searched backwards to find the most recent point at which the player that took the shot was substituted in and used the difference in the time left values of the 2 observations to determine the value of our new variable, "time since substitution." Halftime was also considered a "substitution", since all players get to rest during that point in the game. Finally, we acquired the position of every NBA player from the stats website and left-joined this with our main data set so that each observation had a corresponding position.

## Analysis

For our initial analysis we decided on beginning with the following features:

| Variables in the Final Model | Description | Type of the Variables |
|---|---|---|
| Score Margin | The score difference between two teams | Numeric |
| Location | Home or Away for the team | Factor |
| Shot_Number | # of the shot the player took in the game | Numeric |
| Shot_Clock | Seconds left in the game | Numeric |
| Dribbles | How many dribbles he made prior to the shot | Numeric |
| Touch_Time | Time the player held the ball prior to the shot | Numeric |
| Shot_Dist | Distance to the basket | Numeric |
| Close_DEF_Dist | Distance from the closest defender | Numeric |
| PTS_Type | 2-point or 3-point for the shot | Factor |
| FGperc | 2013 field goal percentage for the player | Numeric |
| DefFGperc | 2013 field goal percentage allowed for his defender | Numeric |
| Position | Guard, center, or forward | Factor |
| time_sub | Time for the player on the court since latest substitution: 0 to 500 seconds, 500 to 1000 seconds, and 1000 to 1500 seconds | Factor |

We first divided our data into training and validation sets. We then used the *glm* function to fit a logistic regression model to the training set that estimates the probability of a given shot being successful given the feature values. The logistic regression has the following formula:

$$p(t) = \frac{1}{1-e^{-t}}, \text{ where } t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

After our first run fitting the logistic regression model, we found that the overall fit of the model was good (p-value << 0.05) and most of the features that we had selected to put into our model made significant contributions to the model (p-values < 0.05). One variable that was (surprisingly) not significant was the closest defender's allowed field goal percentage, so we dropped that and refit the model. All features were significant in the resulting model except for the forward position dummy variable, but we decided to keep it because the guard position dummy variable was highly significant. We proceeded to test for multicollinearity and found that number of dribbles and the touch time had variance inflation factors (VIFs) greater than 5, indicating that these 2 variables were correlated. This was not surprising, since the more dribbles a player takes, the longer they are going to have possession of the ball before shooting. Touch time had a slightly lower p-value than the number of dribbles, so we dropped number of dribbles and refit the model. The model and all variables were still significant, and when we computed the VIFs again, we found that they were all at appropriate levels (less than 5).

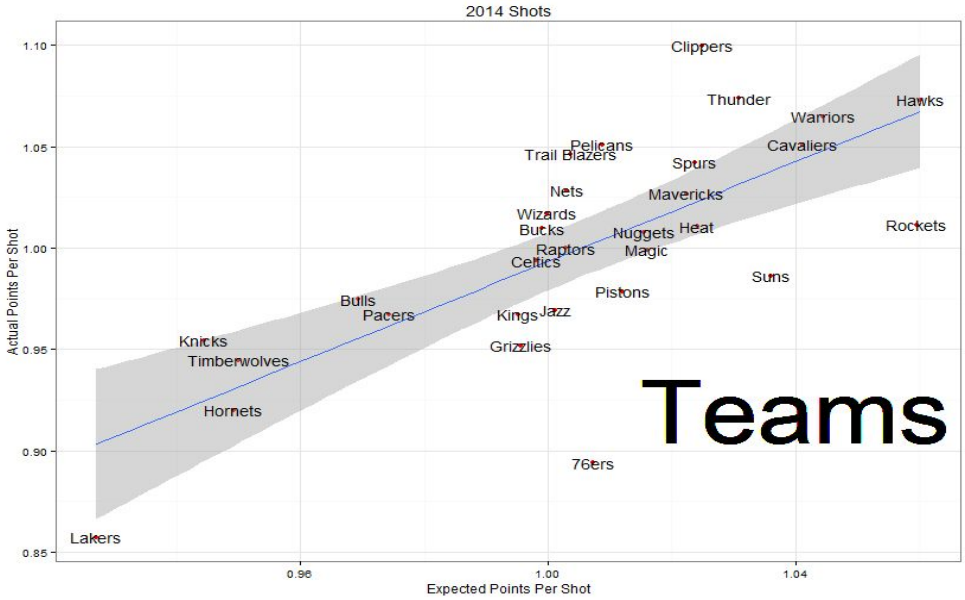
This is the table for the coefficients in the model:

| Coefficients | Estimate | P-value |
|---|---|---|
| Intercept | -0.59 | 0 |
| Location_Home | 0.05 | 0.0000222 |
| Shot Number | 0.01 | 0.0000636 |
| Shot Clock | 0.01 | 0 |
| Touch Time | -0.04 | 0 |
| Shot Distance | -0.07 | 0 |
| Points Type_3 | 0.17 | 0 |
| Closest Defender Distance | 0.10 | 0 |
| Field Goal Percentage | 1.42 | 0 |
| Position_Forward | 0.03 | 0.13866 |
| Position_Guard | 0.15 | 0 |
| Time till latest Sub_[500, 1000] | -0.03 | 0.01433 |
| Time till latest Sub_[1000,1500] | -0.04 | 0.00217 |
| Score Margin | -0.001 | 0.03976 |

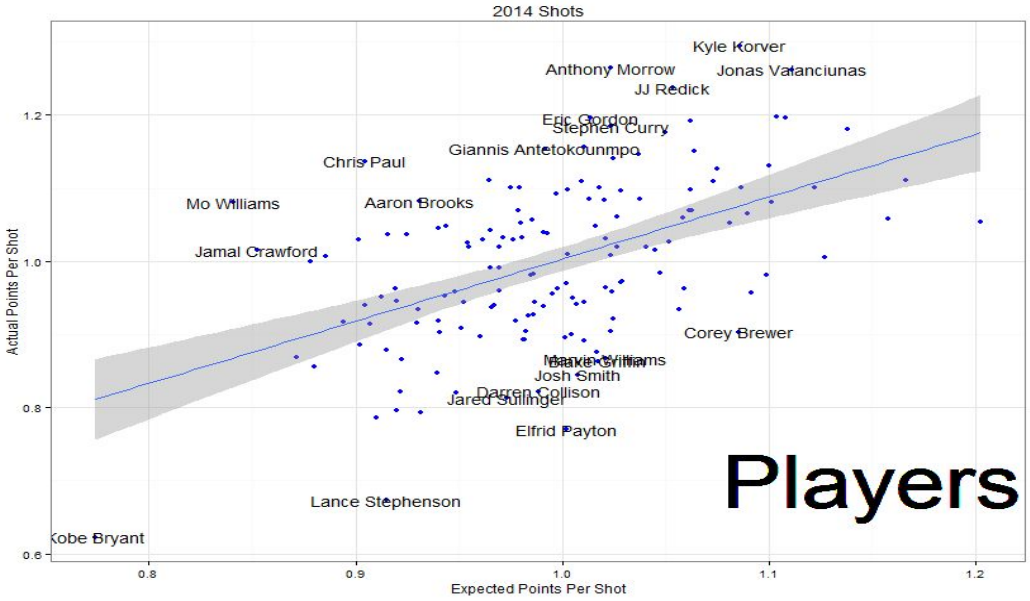
Next, we calculated the expected point per shot for each observation of the validation set. We did this by computing the product of the probability of each shot's success (as predicted by the model) and the type of shot (2 or 3). We then grouped the observations by team and calculated the average of the expected

points per shot and compared this value to the actual points per shot for each team. Our predicted values were strongly correlated with the actual values; the correlation between the 2 variables is about 0.74.

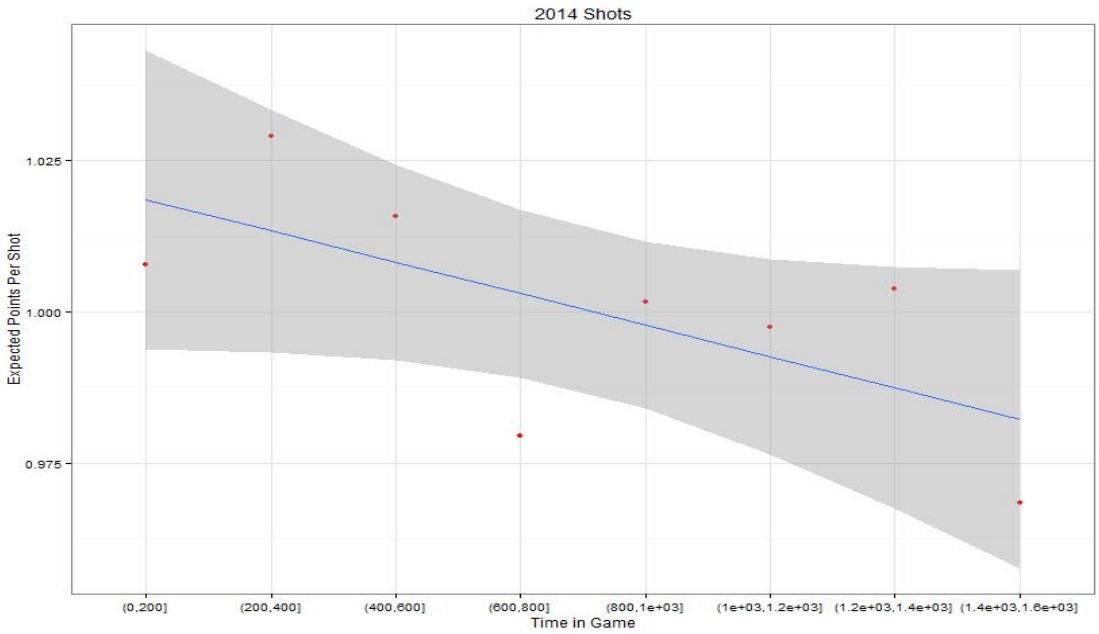
The following is the plot between actual points per shot (PPS) and expected PPS for teams using the testing data. Points that lie well above or well below the curve represent teams for which our model under or over predicted their PPS, respectively.



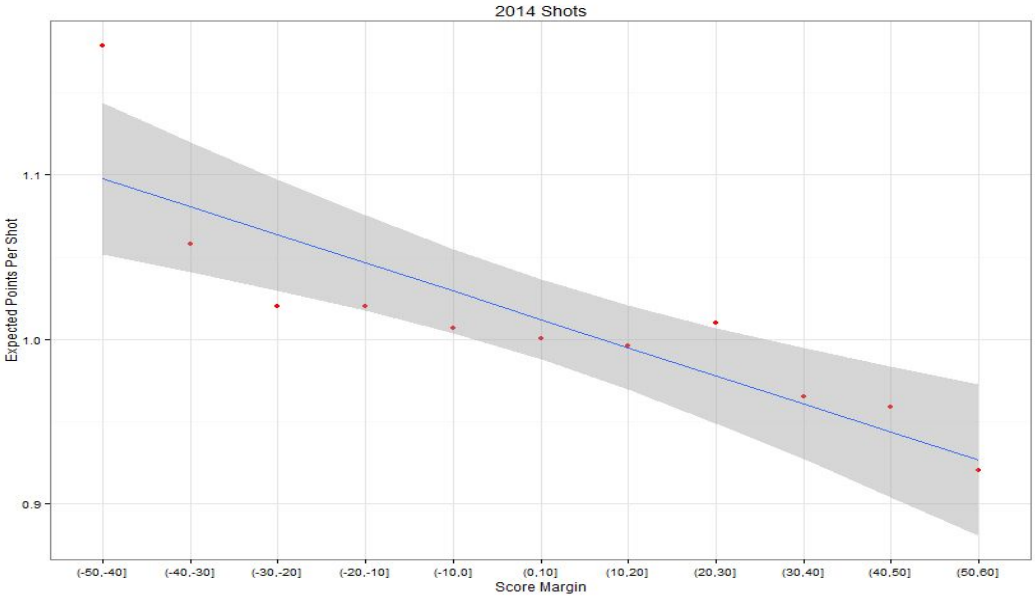

We also performed a similar exercise for individual players. The plot below reflects the actual PPS vs expected PPS for the players that shot at least 100 times during the season. Kobe Bryant is an outlier as he shot significantly worse than even our modest expectations. There are 4 players with an actual PPS value above 1.2 for whom our model under-predicts their PPS because they are essentially only 3-point threats.



The following graph might be useful to a coach in assessing how long his or her players should be left on the floor after they are substituted in.

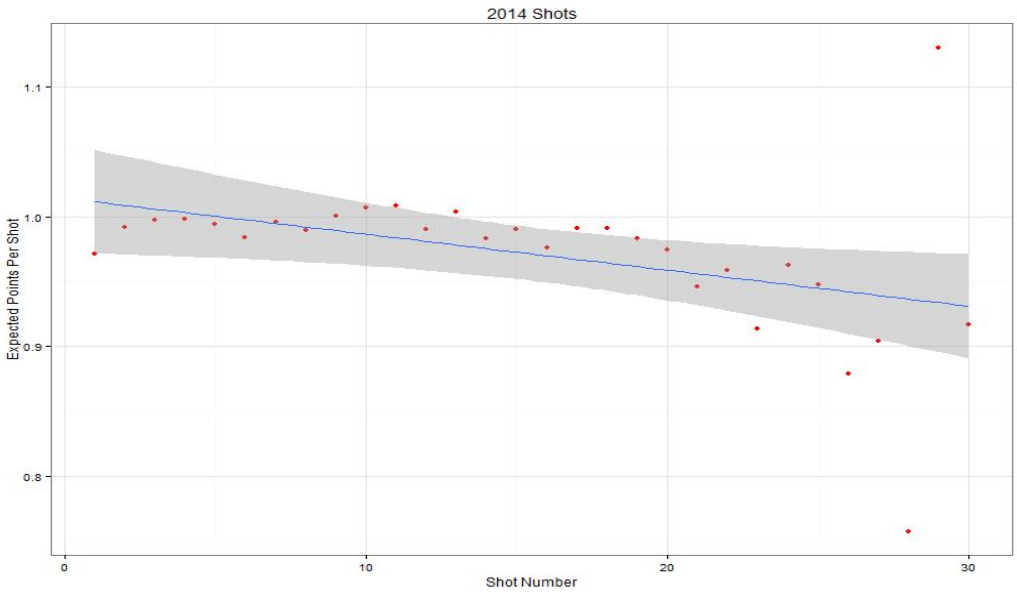



As we see in the above plot, a player's shooting efficiency seems to be greatest after they have been in the game a few minutes and are supposedly "warmed up". At some point between 7 and 10 minutes there is a sharp decline in a player's shooting efficiency. Thus, based on our model a reasonable suggestion would be to try to limit a player's time on the floor to 8-9 minutes at a time and not play an entire quarter without rest..




The above plot shows the expected PPS vs the score margin for the game. The trend suggests that as a team's lead grows, their players are more likely to take lower percentage shots. This may be because they

play outside of the coach's system and play individual basketball, leading to less favorable outcomes. A team that is down a lot might have a greater expected PPS because of a combination of more 3-pointers they are taking and the complacent defense played by the opposing team, which might also be made up of bench players.




Above is a plot of predicted PPS vs the Shot Number for players that took at least 15 shots during 1 or more games during the 2014 season. We wanted to limit it to this sample because otherwise the left side of the graph would contain all players, but the right side of the graph would contain only "volume shooters" that had instances of taking 20+ shots in a game. There seems to be some consistency up to about the 20th shot, at which point the predicted PPS declines. This is likely due to the fact that a player taking that many shots might have weak legs by that point in the game (unless they are Stephen Curry).

## Conclusions

Our model turned out to be a success in terms of accurately predicting the outcome of a field goal attempt. All but one predictor was statistically significant with p-values that were mostly near zero. As far as predictions are concerned, we obtained a correlation value of .74 between the actual points per shot and the expected points per shot. We also had a shot prediction accuracy of about 61%. In other words, we could accurately predict the result of around six out of ten field goal attempts. In addition, our model proved to be helpful in evaluating both team and player performance. By plotting the actual points per shot against the expected points per shot by team and by player, we were able to identify the best performing teams and players and the worst teams and players. Also, our model could be used in developing better coaching and game-management strategies. An example is given by our illustration of how the amount of time a player has spent in game affects his field goal accuracy.

The majority of the limitations of our model could be attributed to the quality of the data. Despite the meticulous steps we have taken towards cleaning the data, some aspects of the data were simply incorrigible or overly complicated. The predictor "TOUCH_TIME" had several nonsensical values that were less than 0 or greater than the maximum allowed touch time. Overtime shots had to be removed from the data because we were obtaining negative values for "time left". The original data was missing the field goal percentages for rookie players. But, due to the significance of the field goal percentages in our model, we decided to keep it by imputing the field goal percentage at the 30th percentile based on that player's position. For example, a rookie guard would be given a 2013 field goal percentage equal to the 30th percentile of the overall 2013 shooting percentage for guards. In addition, due to the way which we acquired the player position data, it was possible for one player to have observations with different positions. A solutions would be to scrape the data from the team websites.

For future work, one could develop a more interactive model (i.e., a Shiny app through the server). The model would also perform better with several years worth of data. For example, a model that draws in data from the last decade to predict the real-time shots in the current season could prove to be a major improvement. One could also explore for other predictors to be added to the model, such as the schedule (to see if players shoot worse when they play back-to-back games) or injuries. Lastly, it might be interesting to create separate models for 2 and 3 points shots, as it seems that our model tends to slightly underestimate the expected value of 3 point shots.