NBA Game Prediction:

Spread, Total, and OREB

Authored by Kailee Zeltner, Kevin Eichlin, Simon Low, Susan Davis and Jinghao Teng

1) Data Information

We created a dataset called all_stats that was used in the exploration and modeling of Spread in NBA games. First, we brought in game data from the 2023 season using the nbastatR package. We then took this data and selected only the variables we needed. This excluded things like the league, the year of the season, the date of the game, whether it was regular season or postseason and a few other redundant variables. We did this to limit extra space being used and to get rid of variables that had the same value for every single entry in the table. We then took the updated games dataset and split it into 'home' and 'away' datasets. This was done because each game was listed twice in the 'game' dataset, once for each of the teams playing in it. So, splitting it this way provided two tables with teams' stats for each game, separated by whether it was a home or away game.

After this, we combined the 'home' and 'away' datasets into one, so that each game was only represented in one row. In this new dataset, which we named 'combined', each row had the away team's name, the home team's name, and every statistic from the game labeled as either a home or away team's value for that specific statistic. We did this using the full_join() function and created Spread, OREB, and Total (Points) variables in the process using the mutate() function. Once we had built the combined dataset, we consolidated the data into seasonal averages for each team. First, we had to split the dataset into home and away again, now with the updated variables. Then, we grouped the data by the team and created new variables that represented the average of each metric. For example, we created average assists by using the summarize function and setting it equal to the mean number of steals across all games. This was done separately for home and away stats. Once season averages sorted by home and away were in datasets, we combined them once more, and calculated percentages along the way. The final all_stats dataset included average counts for all count variables, percentages for all shooting variables, and the 3 variables we are trying to predict: Spread, OREB, and Total Points.

To provide ourselves with another larger data set containing more game logs that could be used for comparison and testing in our model creation process, we utilized the nbastatR package to retrieve all game logs dating back to the 2018 season. This cutoff mark was selected due to the most recent significant rule change in the sport taking effect at the beginning of the 2018 season, during which the amount of time placed on the shot clock following an offensive rebound was shortened from 24 seconds to 14. We identified this as significant since this would ultimately result in more possessions being played in an average NBA game, which would have a positive effect on the opportunity for points and offensive rebounds which directly impacts the models we are trying to create.

One of the first datasets we created, which we named 'com', combines data on basketball games played between two teams. Com was made from the data frame we were given in the nbastatR package called 'game' that contains information on each game including the team names, the points each team scored, the location of the game (home or away), and other statistics. The games in com cover the entire 2023 season and are continuously updated in real time. When working with our Spread data we made a new data frame called com_improved that renames the column to match with the x values of our chosen model. When testing for the team matchup averages, explored in the spread methodology, we also created a com_improved_2_dataset that removed all games with an id game value greater than 22201038. This was done because games with an id value greater than 22201038 are part of the Testing dataset that we later go on to predict in order to evaluate the effectiveness of our different models and methods. If we were to include these games when taking the average values of team matchups, the game whose spread we are predicting would be part of the dataset that the model was built from, thus making our predictions artificially accurate.

If we were to build a model using the game logs data, we'd run into the problem that we don't have the game logs for future games. We used two methods to solve this problem: using estimated game logs calculated by previous matchup stats as well as recent game logs, and building models with data we'd have access to prior to future games. Thus, we calculated cumulate average game logs using previous game stats so that we'd have data on variables necessary to predict future games. We determined that cumulative averages do a better job predicting than seasonal averages, as cumulative averages change based on a team's performance as the season goes on, whereas seasonal averages only measure overall performance of the season.

When examining potential predictor variables that could be used to predict the total value of points scored in a future NBA game, one statistic that was not included in the nbastatR package or Kaggle dataset that we chose to utilize was the Effective Field Goal Percentage, or eFG%, of the teams. This formula and statistic was first introduced by Basketball Statistician Dean Oliver, who is the author of the basketball journal “Basketball on Paper" and has held numerous roles within NBA organizations. eFG% serves as a means of creating a weighted shooting percentage that properly accounts for the weight of the 3-point shot. In an era where 3-point shots have become ever more prominent and popularized in the game of basketball, it is important to weigh this aspect properly in order to more accurately resemble both a player's and team's shooting efficiency from the field. For example, a team that takes significantly more 3-pointers will likely have a lower eFG% compared to a team that doesn't, even if the team shooting 3-point shots consistently produces better offensive performances and scores more points. This method has been deemed a better indicator of future offensive performances, eliminating the potential bias of the 2-point shot in standard eFG%, which would be useful in attempting to predict the total points scored between two teams in future games. The formula used is presented below:

$$(Field\ Goals\ Made\ +\ 0.5\ *\ 3\ Point\ Field\ Goals\ Made)\ /\ Field\ Goals\ Attempted$$

In addition to eFG%, another statistic that was not included in the nbastatR or Kaggle dataset that we wanted to incorporate into our predictions is the number of possessions for each game. The number of possessions a team has during a game is an important statistic for evaluating team performance as it directly affects a team's scoring opportunities and their ability to control the tempo of the game. When it comes to OREB, the pace of the game heavily influences the shots a team attempts, and the opportunities for getting OREB. Since we could not find possessions for each game, we used a formula to calculate the estimated number of possessions. This formula is a variation of the turnover rate formula, which is part of the "Four Factors" formula by Dean Oliver. The weighting factor of 0.96 is used to account for situations where a team gets multiple possessions in a single trip down the court. The exact origin of the 0.96 weighing factor is not clear, but it has been used in basketball analytics for many years and has become widely accepted. The formula used is presented below:

$$Possessions\ =\ 0.96\ *\ (Field\ Goals\ Attempted\ +\ 0.44\ *\ Free\ Throw\ Attempted\ +\ Turnover\ -\ Offensive\ Rebound)$$

Similarly, in order to help us better predict OREB, we wanted to take height and weight variables into consideration. From an experience standpoint, it appears that the bigger a player is, the more likely they will be able to get rebounds. Therefore, we wanted to measure the size of the players on a team level, as our other variables are team level data. Our initial thought was to use average height or weight, but that could lead to a problem with outliers. For example, if we have a 7 '4 player who plays 1 or 2 minutes each game, he would affect the average height but not have much influence on the OREB. So, we use mins(minutes played) to weight players' size. These are the three formulas we created measuring the weighted average height, weight, height*weight of the team:

$$Height_{weighted\ avg} = \sum(Height * Mins)/ \sum Mins$$
$$Weight_{weighted\ avg} = \sum(Weight * Mins)/ \sum Mins$$
$$Height * Weight_{weighted\ avg} = \sum(Height * Weight * Mins)/ \sum Mins$$

It turned out that the correlation for each of these with OREB is not statistically significant. We also failed to calculate these three new variables at the game level, so we did not include these in our final model to predict OREB.

In the game logs data, many variables have really high correlations and some variables ultimately represent the same thing. We decided to replace some of these variables with OREB related variables. When a team missed a shot, they could have the opportunity to get an OREB. Thus, we created missed shot variables to replace the attempts:

$$fgmsTeam = fgaTeam - fgmTeam$$
$$fg3msTeam = fg3aTeam - fg3mTeam$$
$$fg2msTeam = fg2aTeam - fg2mTeam$$
$$ftmsTeam = ftaTeam - ftmTeam$$

The ftms (free throws missed) variable is interesting because teams are only able to get an OREB when it is the last free throw attempt. That's why in the possession formula, there is a weighing factor of 0.44 for each Free Throw Attempt. So, we will see how ftms perform later in the modeling process. Thus, the variables we have for predicting OREB are: pctFGTeam, pctFG3Team, pctFTTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, ptsTeam, possTeam, fg3msTeam, ftmsTeam, fg2msTeam

In addition, we wanted to identify the levels for our cumulative average game logs. We created 6 levels of cumulative data: all previous games, all previous games without the first 15 games of the season, previous 15 games, previous 7 games, previous 3 games, previous 1 game. The reason why we did this is that we wanted to see which games are most indicative of a team's future performance on OREB. All previous games measured the performance of the team until the future game. We hypothesized that intuitively this could better represent the ability of a team before the game. All previous games without the first 15 games of the season could be identified roughly as not including the first month of games. At the start of the season teams are still building their chemistry. For the cumulative average, it's also not representative at the beginning of the season as the data is limited. To consider recent performance, we included levels with different numbers of previous games. Previous 15 games is often the widest range to consider recent performance, as on NBA or ESPN websites, the earliest recent game one can pull out is often the previous 15 games. Previous 15 games could also be viewed as recent performance in the last month. Previous 7 games could be seen as performances of the last two weeks, and the previous 3 games are for last week. This is how we came up with six levels of data.

We initially thought the recent data might have more influence in predicting OREB. In that way we could modify our data by adding more weight to more recent games. For example, we could take an average between data for all previous games and previous 15 games. In order to make sure what levels of data we should choose, we compared them by making a linear model and a Poisson model for each level of data. So, there are 12 models in total. For each one of them, we selected all the variables that we filtered for predicting OREB. Looking at the summary of each model, we identified that using all previous games without the first 15 games of the season gives us the best models. As we can see from the table below, the recent data did not perform as we expected. The adjusted R^2 for the previous 1 game is very limited and the AIC is also higher than the AIC of all previous games without the first 15 games of the season. Therefore,based on this result, we decide to choose all previous games without the first 15 games as our data level for predicting OREB.

	Linear Model	Poisson Model
Levels of Cumulative Avg	Adjusted R^2	AIC
All Previous Games	0.1111	7235.1
All Previous Games without First 15 Games	0.1277	5853.7
Previous 15 Games	0.08462	5911.9
Previous 7 Games	0.06263	5943
Previous 3 Games	0.05521	5953.6
Previous 1 Games	0.03331	5983.9

Table 1. Adjusted R^2 and AIC for models using different levels of cumulative data when predicting the OREB.

2) Methodology for Spread

2.1) Picking Our Model

We started off by creating a basic linear regression model that predicts Spread using the variables in the nbastatR data package. Using our com_improved dataset, where each row represents a game in the 2023 season, we set the seed, shuffled the data, and split it into a training dataset with 75% of the games and a holdout dataset with the remaining 25%. We then utilized a stepwise algorithm to build a baseline model that predicts Spread using the training dataset, and found the best predictors for Spread to be HomeFGPct, AwayFGPct, HomeFGA, HomeFTA, AwayFTA, HomeFG3A, AwayFG3Pct, AwayFG2Pct, AwayDREB, HomeDREB, AwayREB, HomeTO, AwayTO, and HomeBLK.

We then used this stepwise model to predict the Spread for each of the games in the holdout dataset and calculated the holdout mean residual, mean absolute residual, and root mean squared error. We compared the root mean squared error to that of the training dataset predictions and calculated the shrinkage. Interestingly, we calculated a negative shrinkage of -0.00119052, which implies that our model does a better job predicting the spread of the games in the holdout dataset than it does the games in the training dataset it was built from. However, the training dataset is much larger than the holdout dataset so this result isn't particularly meaningful or significant.

What's more significant is a comparison of the mean absolute residuals when predicting the spreads of the games in the holdout dataset using our stepwise model compared to an empty model and a full model. In order to determine the usefulness of our stepwise model, we built an empty model that consists of only the Spread variable, and thus predicts each game in the holdout dataset to have a spread of 3.078 which is the average spread of all the games in the training dataset. On the other end of the spectrum, the full model predicts Spread using all 12 of the variables provided in the NBA Stat R dataset. For each of these three models (Stepwise, Empty, and Full) we calculated the holdout mean absolute residual and compared them. The mean absolute residuals were 1.279, 11.186, and 1.288, for the Stepwise Model, Empty Model, and Full Model, respectively. Since the Stepwise Model has a smaller mean absolute residual than the Empty Model, this indicates the variables in our stepwise model are useful and improve the model's ability to predict Spread beyond that of a very simple model with average spread alone. Similarly, since our Stepwise Model has a mean absolute residual that's only marginally different from the Full Model, we concluded that using the Stepwise Model is just as good at predicting Spread as the large Full Model.

Additionally, we looked at the mean holdout residual, as this tells us about the direction of bias. The mean holdout residual was calculated to be 0.06246, which indicates that our model slightly underpredicts Spread values. That being said, since this value is small and close to zero, we decided that our stepwise model only marginally underpredicts Spread and that when predicting the games from April 4th through 9th, we should not subtract 0.06246 from them.

2.2) Finding Values for Predictive Variables

We then began to think about the problem of how to predict Spread for games that haven't yet been played since we don't know the value of many of the variables necessary for our model, such as HomeFGPct (the home team's percent of made field goals), HomeFGA (the number of field goals the home team attempts), etc.

We came up with four options for what to input for each of our unknown variables: Method (1) The average for each team when they play a specific team at home or when they play a specific team away, Method (2) in which we input the home team's average values over a small subset of their last five home games and for the away team we input their average values over a small subset of their last five away games, Method (3) in which we input the home team's average values of a larger subset of their last 20 home games and for the away team we input their average values over a larger subset of their last 20 away games, and Method (4) in which we input the home team's season averages at home for each variable and for the away team we input their season averages away for each variable. We hypothesized that methods 2 and 3 would do a better job predicting a game's spread than method 4 (using the season average) as the subset averages are more indicative of a team's current momentum and performance. In order to determine which of these options does the best job predicting Spread, we created a Testing Dataset.

The Testing Dataset consisted of each team's most recent home and away game. In order to build this dataset we grouped the game data by the Home Team Name and sorted the data descending by the Game ID. We then used the slice function to slice just the first row of this data, giving us each team's most recent Home game. We then repeated this process, but grouped by the Away Team Name to get the most recent away game for each team. We then used rbind function to bind these two data subsets together to give us a dataset with each team's most recent home and away game. We then set out to predict the spread of each game in the Testing Dataset, using options 1, 2, 3, and 4 to determine what averages to feed into our model.

2.2a) Averages of Team Matchups

In order to be able to find the average values for a team when they played a specific other team at home or a specific other team away we created our own function called teamvteam. This function, given a specific home and away team first filters the com_improved_2 data frame to include only the games between the two teams specified by the arguments. If there are no games between the two teams, the function returns NA. If there are games, the function proceeds with making the prediction by creating a new data frame (games_empty) with the same structure as the games data frame but with all numeric columns filled with NA. Then, for each row in games_empty, the function calculates the average values of the features for the specific matchup of the two teams and fills the corresponding columns in games_empty with these values. The function then uses the predict function to make a prediction based on the stepwise model and the data in games_empty. The function then returns the first prediction which represents the predicted spread of the specific matchup when the first input of the function is the home team and the second input of the function is the away team.

We fed this function all the rows in the Testing Dataset in order to get predictions for these specific games . With these outputs we used the actual spread from the testing data set for each game and subtracted the predicted spread for each game, giving us residuals for each game. Taking the absolute value of these residuals and then taking the average of these values we got a value of 13.78028.

2.2b) Averages of the 5 Most Recent Home and Away Games

Then, we utilized Method 2 and supplied our model with the average stats for a team's 5 most recent home and away games. We took our subset of data that only included stats on the home team for each game and grouped this data by team, sorted in descending order of Game ID, and sliced from 2-6, so that we'd have the 5 most recent games, excluding the current game (which would be what we're predicting in the testing data). We repeated this process for stats on the away team for each game, and then we repeated the process we'd done to make the all_stats dataset with a team's season average, but instead of using the entire season's worth of games, using these subsets. We then merged these averages into the Testing Data, so that we had a dataset with each row representing a game that was a team's most recent home or away game and the columns with data on that game (such as the home team's percent of made field goals) being the home team's average value from their 5 most recent home games and the away team's average value from their 5 most recent away games. We then used our Stepwise model to predict the spread of each of the games in the Testing dataset.

2.2c) Averages of the 20 Most Recent Home and Away Games and Season Averages

We then repeated this same process, but for Method 3 we sliced from 2-21 in order to predict the Testing Data with averages from each team's 20 prior home and away games. Finally, for Method 4 we repeated the method again, but simply removed each team's most recent home and away game in order to create season's averages with all games of the season except for the games in the Testing dataset. Once we had Spread predictions for the games in the Testing dataset using the game matchup method (Method 1), last 5 home and away games method (Method 2), last 20 home and away games method (Method 3), and season's average method (Method 4), we calculated and compared the mean absolute residual for each method. We calculated a mean absolute residual of 13.78028, 14.61001, 12.37751, and 11.79396 for Methods 1, 2, 3, and 4, respectively. Thus, we had evidence to suggest that contrary to our initial hypothesis, using a team's season average does a better job predicting a game's spread than team matchup averages as well as using a subset of averages based on the 5 or 20 most recent home and away games. We'd later go on to test this again for methods 2,3, and 4, but this time predicting the games taking place from March 29th to April 2nd. We did not include method 1 in this further analysis as it performed worse then methods 3 and 4. Method 2, however, is still included because despite it performing the worst of all four models, it is built with the same structure as methods 3 and 4 and is therefore easily manipulated for differing cases such as change in length of time for our data.

Method	Mean Absolute Residual
(1) Team Matchups	13.78028
(2) Using Last 5 home and away games	14.61001
(3) Using Last 20 home and away games	12.37751
(4) Using season average	11.79396

Table 2. Mean absolute residuals for the 4 methods when predicting the Spread of the games in the Testing dataset.

2.3) Selecting Span of Time for our Data

We also wondered whether it'd be better to predict Spread using a model built from game data from only the 2023 season, or if incorporating data from older seasons would improve the model. To determine this for each of the models mentioned above we created an additional version that uses data from 2018-2023. We decided not to go any further back than 2018 since the NBA changed the length of the shot clock at the start of the 2018 season.

We then built a stepwise model that has data from more than just the 2023 season, but from the 2018, 2019, 2020, 2021, and 2022 seasons as well. In order to build this model (referred to as the More Years Model) we took the NBA dataset and selected out the data from games that took place in the 2018, 2019, 2020, 2021, 2022, and 2023 seasons. We then shuffled this dataset and split it into a Training dataset with 75% of the games and a Holdout dataset with 25% of the games. We then used the stepwise algorithm to build a model that predicts spread using the Training dataset. (Prior to running the stepwise algorithm, we removed the same variables we'd removed when building a stepwise model that has only the 2023 season data). The Stepwise More Years model used the variables: HomeFGPct, AwayFGPct, AwayOREB, AwayTO, HomeTO, HomeREB, HomeFG2A, AwayFG3A, HomeFG3Pct, AwayFG2Pct, AwayDREB, HomeFTPct, AwayFTPct, AwayFTA, HomeFGA, HomeFTA, AwayFGA, HomeFG2Pct, AwayFG3Pct, OREB, AwayAST, HomeAST, AwayBLK, AwaySTL.

We then used this Stepwise More Years model to predict the spread of the games in the Testing Dataset using methods 2-4. We calculated the mean absolute residuals to be 14.566, 12.36723, and 11.80459, for Methods 2, 3, and 4, respectively. Comparing the 2023 only model to the Stepwise More Years model, for Methods 2 and 3 the More Years model results in smaller mean absolute residuals, but for Method 4 the 2023 only model results in smaller mean absolute residuals. That being said, the difference in mean absolute residual between the model built with only 2023 data and the model built with 2018-2023 data is marginal. Thus, we concluded that our best model and method thus far is using the 2023 only model and the season's average method (Method 4), as this resulted in the smallest mean absolute residual of 11.79396 when predicting the spread of the games in the Testing Dataset.

	Model built with 2023 data alone	Model built with 2023-2018 data
Method	Mean Absolute Residual	Mean Absolute Residual
(2) Using Last 5 home and away games	14.61001	14.566
(3) Using Last 20 home and away games	12.37751	12.36723
(4) Using season average	11.79396	11.80459

Table 3. Mean absolute residuals for methods 2-4 and both the Stepwise Model and Stepwise More Year Model when predicting the Spread of the games in the Testing DataSet.

We then set out to predict the spread of real games that haven't yet taken place, essentially simulating what we'd be doing for our final submission. We used the Stepwise model and Stepwise More Years model alongside Methods 2-4 to predict the Spread of the games from Wednesday March 29th to Saturday April 1st. Although we'd previously done something similar with predicting the Testing Data, we wanted to try predicting real games that hadn't yet occurred to check if there were any sort of mistakes we'd made previously in which our models had access to the actual value of the game's spread and were using this to make the predictions. On April 2nd, once all the games were completed we compared each method's predicted spread to the actual game spread. We ultimately found that the model using season's averages (Method 4) and 2023 data alone (Stepwise Model) had a mean absolute residual of 13.14489. The model using season's average (Method 4) and 2018-2023 data (Stepwise More Years Model) had a mean absolute residual of 13.1546. The model using the averages from the last 5 home and away games (Method 2) and 2023 data alone (Stepwise Model) had a mean absolute residual of 13.9873. The model using the last 5 home and away games (Method 2) and 2018-2023 data (Stepwise More Years) had a mean absolute residual of 13.99441. The model using the last 20 home and away games (Method 3) and 2023 data only (Stepwise Model) had a mean absolute residual of 13.43624. Finally, the model using the last 20 home and away games (Method 3) and 2018-2023 data (Stepwise More Years Model) had a mean absolute residual of 13.345. Thus, we determined that our best and final model utilizes Method 4 and 2023 data alone.

	Model built with 2023 data alone	Model built with 2023-2018 data
Method	Mean Absolute Residual	Mean Absolute Residual
(2) Using Last 5 home and away games	13.98373	13.99441
(3) Using Last 20 home and away games	13.43624	13.345
(4) Using season average	13.14489	13.1456

Table 4. Mean absolute residuals when predicting the Spread of the games from March 29th to April 1st.

The formula for our final model (Stepwise Model) is Spread = -2.59075 + 204.11548*HomeFGPct -209.71329*AwayFGPct + 0.83699*HomeFGA -0.83892*AwayFGA + 0.72861*HomeFTA -0.72199*AwayFTA + 0.45501*HomeFG3A -14.95930*AwayFG3Pct -22.24137*HomeFG2Pct -0.45331*AwayFG3A + 20.74863*HomeFTPct -19.63906*AwayFTPct + 18.08059*HomeFG3Pct +27.40178*AwayFG2Pct + 0.12260*HomeREB -0.12669*HomeTO +0.11147*AwayTO -0.10882*AwayDREB -0.10843*HomeOREB, and to make our predictions for the games from April 4th through 9th we utilized Method 4, using each team's season average for the variables in the model.

3) Methodology for Total

To determine the best method of predicting the total points that would be scored in a game between the two teams playing, we elected to construct two different models through differing methodologies. We constructed both a linear model with a higher number of predictor variables as well as a Poisson regression containing fewer. After these models were built, our idea was to use both of them to predict a range of recent NBA games and compare their correlating residuals from the actual total which took place, and choose the model that performed the best in this sample size as our best prediction method.

3.1) First Created Model for Predicting Total Points Scored (Linear)

The basis of this model was created by examining the relationship between box score statistics in single-game settings during the 2023 NBA season, and their effect on the "total points" outcome of the specific game.

To clean this dataset to predict total points, several variables had to be excluded from the model selection process. This included all non-numeric variables such as GameID, Year, Team Name, and type of season. There was also a group of numeric variables that needed to be excluded due to multicollinearity issues that were presented upon initial testing. This included the team combined totals of different counting statistics such as the total number of offensive rebounds in the game, which in a sense were simply summed duplicates of the already existing "Home" and "Away" values for the respective teams. The next variables that were deemed necessary to remove were those that provided too much of a direct correlation, such as Free Throws Made, 3 Point shots made, and 2-point shots made. The rationale behind this is that because these variables have a specific point value attributed to them when running model selection methods they would always be chosen and result in a perfect fit as they directly calculate and result in the game's totals. The number of attempts for each of these shot types was also removed due to issues of multicollinearity that arose, as the make percentage of each is also present in the data set.

The next step that was taken in this process of building the model, was creating a testing and training data set that could be used to test the resulting model, consisting of all game logs of the 2023 season, excluding those from March 31st - April 3rd. These rows of game data were then shuffled and placed into a training and holding set, with 80% of the data available in the training, and 20% in the holdout. Using this training dataset, the next step was to utilize a stepwise function to select prediction variables and build a model. After this model was constructed we ran the model using the holdout dataset, calculating and comparing the corresponding residuals with that of the Full model which was made up of all the possible predictor variables in the dataset. The residual mean came out to be .0746 which indicates that the linear prediction model that was created, was viable and consistent over all the games throughout the year. The residual standard error also came out to be 7.31 indicating that there were apparent instances with outliers that varied around the linear regression model, however, both of these values were smaller than that of the Full model, indicating that the slightly smaller one built utilizing the stepwise method, fit the dataset better. The next approach that was made was finding both the standardized and studentized residual values from our model and dataset, and identifying all of those over the cutoff of three, as those data points would have the potential to be influential over our model's construction. We then took these specific points out of our dataset to see if our model would fit better under a new run with the stepwise function, and unfortunately, it did not make much of a difference, with the residual mean actually slightly increasing from the altered model and therefore elected to continue with the initial. We then ran VIF tests to ensure that there were no significant instances of multicollinearity that needed to be addressed between the variables that were selected for the initial model. We then created and examined QQ plots for our models, ensuring that the conditions for linearity were met.

The next approach that we took in this selection process was to repeat all the previous steps, however this time including all games beginning with the 2018 NBA season allowing us to work with a bigger dataset. However, following the model creation and calculating the residuals, we saw that the model including prior years had a slightly higher average residual of -0.1152 and a slightly higher residual standard error of 7.4296. Because the goal of this project is to predict game totals during the 2023 season, we then took this model that was created using prior years and ran this with a holdout set containing a set of games from this season to see if it would remain consistent. Upon doing so, the corresponding average residual value with the game totals was 4.85, significantly higher than the residual average of the 2023 model when compared with the same holdout set. This came out to a value of 0.0897. While there are benefits to using a larger data set for prediction accuracy, this difference in residual values is likely since team lineups, coaching philosophies, offensive plays, and pace of play have all improved over the years. For example, the 2022-23 NBA season is on pace to have the highest Team PPG average since 1970. Because of this, we ultimately elected to build our prediction models using solely game data from this season.

The next step that was taken in this process was identifying potential outside statistics that may be beneficial for the model. The two that were decided upon to be used were the number of possessions in a game and the team's effective field goal percentage. These values were calculated using the formulas depicted in the data information section of this paper, and then merged into our dataset, creating variables of possessions and eFG% for both the Home and Away teams in each game. Once these variables were introduced as potential predictors, we returned to the model creation and testing process once again. Taking the original 2023 model, we added the 4 variables (Home and Away for Possessions and eFG%) into the model and examined the corresponding p-values to ensure that with these new predictors, they still showed significance at the appropriate, 5%, level. Ultimately it was shown that Away team assist totals and FG%, were no longer significant and were taken out of the model. When running this model again with the same holdout set. After the inclusion of these variables, the residual mean continued to decrease even further, this time lying at 0.0159, and with a residual standard error of 2.591. This shows that with their inclusion, the model was able to fit the dataset even better than before.

The next issue that we encountered was determining a way to take this model and use it to predict the total outcome of games that had not happened yet. While this model has shown high accuracy in correlating box score statistics to the total points scored, we need to determine how well it would do when those statistics are relatively unknown, as the games have not happened yet. Our approach from this was importing a dataset containing all of the team averages and totals from the season, merging this dataset to also include possession data and efficiency ratings as well. After doing this, we needed to clean up this new dataset containing team averages, renaming all of the variables to match those in the original 2023 game statistic set for consistency purposes and ease of implementing the model into this new dataset. To make the predictions easier, we created a “predict_game” function that pulled the necessary predictor variables from the dataset, for the home and away team that was manually imputed into the function. It then took this pulled data and placed it into a new data frame, running the linear prediction model with it, and printing an output of what the total points prediction would be.

We continued to experiment with different variables after completing this prediction process, determining which provided the most accurate prediction for NBA games that happened throughout the weekend of March 31st to April 3rd. Using the prediction function for each of the games that were scheduled, we were then able to determine how accurate our predictions were after the games had been played, by examining the residuals from the total number of points that were scored. This provided our final method of testing different variable combinations in the model, attempting to get both the residual average and standard error to be as low as possible. We also experimented with adding another outside variable into the model, teams' offensive ratings. The offensive rating statistics show the number of points a team or individual would score every 100 possessions, allowing for analysis of scoring impact and ability on a standardized scale which is important as other non-offensive factors may allow a team to score more through simply having more possessions, such as forced turnovers on defense. The equation for calculating offensive rating is as follows:

$$(Points\ Scored\ (Team\ or\ Player)\ /\ Team\ or\ Player\ Possessions\ in\ Game)\ *\ 100$$

However, after implementing this into the model it was apparent that this did not help with bettering our prediction residuals and was not included in the final version. The best linear model that we created consisted of 20 different prediction variables that are listed below, having a residual average of -6.3108 for the games it predicted over the weekend.

Home Team Variables: Effective FG%, Possessions, Offensive Rebounds, Turnovers, Free Throw Percentage, Personal Fouls, Defensive Rebounds, Assists, Steals, Blocks, and Field Goal Percentage.

Away Team Variables: Effective FG%, Possessions, Offensive Rebounds, Turnovers, Free Throw Percentage, Personal Fouls, Total Team Rebounds, Steals, and Blocks

3.2) Creating Models beyond Linear Regression

After creating multiple models using linear regression, we decided to predict the total number of points scored in a game using several different modeling techniques. Linear regression can be very powerful, but it is not always the best choice when it comes to modeling data for prediction. The relationship between the dependent variables and predictors may be strong, but not linear. The other types of models we tested include Poisson, Ridge regression, polynomial models, and linear models with interaction terms. Many of the predictor variables used are the same in each type of model. We arrived at these variables by trying many different types of models that cited certain variables as having stronger relationships with total points scored, as well as considering the nature of a basketball game and which stats would be most likely to affect the total points scored. For example, the number of possessions would logically have a strong correlation with the number of points since points must result from possessions, and the more possessions you have, the more chances at scoring you have.

3.2a) Poisson

The first type of model we tried after the linear regression models was the Poisson model. The Poisson model had several advantages that led us to try it. It is generally very good at predicting count data, which is what the classification of total points scored in a game is. Additionally, it is flexible with the number of predictor variables and is straightforward to interpret.

The primary terms that we looked at including in the Poisson models we tried were the calculated number of possessions variable for the home and away team. Other variables used included steals, 2- and 3-point field attempts and percentages, and defensive rebounds. We tried models that included anywhere from two to eight of these variables for the home and away team and arrived at one model that performed better than any other. This model included coefficients for the number of possessions for the home and away team and the number of steals for the home team. The equation that this model is represented by is:

log(Total Points) = 3.894555 + 0.008028*(Away Possessions) + 0.008234*(Home Possessions) - 0.008246(Home Steals)

Its mean absolute error (MAE) is 15.80485, its root mean squared error (RMSE) is 20.07076, and its average residual was 0.06306223, meaning they were generally centered at zero.

3.2b) Ridge Regression

One of the modeling techniques we tried to predict the total points scored in a game was Ridge regression. One advantage of Ridge regression is that it is often used to prevent overfitting. It does so by adding a penalty term to the regression model that shrinks the coefficients toward zero, which ultimately reduces the model’s complexity. As a result, Ridge regression models are more generalizable which can ease the risk of errors in predictions.

When running the Ridge regression model in R using the glmnet package and utilizing ggplot to visualize residuals, Ridge regression did not seem like the ideal model. Its residuals were slightly off-centered from zero at 0.6274901. We think this is due to one extremely large outlier that sits above 120 points off the actual value. Other than that outlier, all predictions come within 50 points of the correct total. However, this model was still not great even excluding the outlier. The MAE came out to be 15.98105 and the RMSE was 21.61519. In the end, when looking at every model that we had considered, the group decided against predicting the total points of the game using a Ridge regression model.

3.2c) Polynomial Modeling

After moving on from linear regression and trying Poisson models and Ridge regression models, our group decided to try an alteration of linear regression modeling using polynomial terms. Our approach to trying to find the best polynomial model to fit the data began with the variables which showed the strongest relationship to total points scored in the linear regression model. This variable was the calculated number of possessions for the home and away teams.

We began by testing simple quadratic functions using just these variables. We then began adding other terms in the first degree that included team steals, field attempts from 2-point and 3-point range, defensive rebounds, and free throw attempts. None of these attempts produced a significantly lower MAE or RMSE than the simple model using only possession data. We then switched the variables for the number of possessions for the home and away teams to be in the first degree and set the other variables to be squared in the model. Once again, this did not change our MAE or RMSE very much. In addition to not being able to get our errors to be low enough, many of the average residuals were not close enough to zero. The model with the lowest MAE ended up being of the form: (Home Possessions) + (Away Possessions) + (Home 3FG Attempts)^2 + (Away 3FG Attempts)^2 with an MAE of 16.72183, an RMSE of 22.43406, and an average residual of 0.5811024.

We concluded that this was not sufficient in our search for a viable model to predict the total points for a game. The high amounts of prediction error and the average residual value that was not very close to zero were the determining factors in this decision.

3.2d) Interaction Terms

Upon moving on from a polynomial model, our group decided to attempt to use interaction terms to better predict the total score of NBA games. Interaction terms can be very useful in a model because they can capture the effects of one predictor variable on the response variable when the same instances' values for a different predictor variable are very different.

Most of the models we tried involved using the team possessions variables either as individual terms, an interactive term with each other, or a combination of the two. Additionally, we also ran the model using other variables such as steals, defensive rebounds, 3-point field goal attempts, and 2-point field goal attempts. When using each of these variables, we once again added them to the model on their own, as an interaction term with each other, as an interaction term with the possessions, as an interaction term with one of the other variables listed, or as a combination of those. The best model in terms of MAE came out to be one that included each team's number of possessions and steals as variables on their own, along with interaction terms between each team's steals and possessions:

(Home Possessions) + (Away Possessions) + (Home Steals) + (Away Steals) +
(Home Steals : Home Possessions) + (Away Steals : Away Possessions)

Its MAE was 16.59259, its RMSE was 22.26584, and its average residual was 0.2913136. Although this improved upon the same measures for our polynomial model attempts, it was still not good enough.

3.3) Choosing our model

After trying many different variations of multiple different types of models, the one we arrived at to predict the total points of an NBA game was our Poisson model. It took into account the number of possessions for each team, as well as the number of steals for the home team. Its MAE was 15.80485, its RMSE was 20.07076, its average residual was 0.06306223, and its residual standard error was 1.206743. After choosing this model, we picked a day when we'd predict all 13 games' total points. Of the 13 games, two went into overtime and our predictions were very far off in those games, expectedly. However, every prediction came within 16 points of the actual total points in the other 11 games, and 3 games came within just 3 points of the actual value. Although this sample size is not very large, we did it more as a trial run. We were generally satisfied with the results. The final equation to find the Total Points estimate is as follows:

log(Total Points) = 3.894555 + 0.008028*(Away Possessions) + 0.008234*(Home Possessions) - 0.008246*(Home Steals)

3.4) Next Steps

While we believe our chosen model to be a good predictor of total points in future NBA games, we also acknowledge that other factors may influence the total that are not accounted for in this model. Two primary examples of this are player injuries and the possibility of overtime in any given game. One of the next steps that could be taken to improve the possible accuracy of this model would be to find a way to incorporate specific players' availability and minutes into the equation. This could be done by creating player weights by examining and quantifying individual players' scoring ability and configuring a metric that takes this into account for the model. However, unanticipated injuries will never be able to be used in creating a prediction due to them not having taken place beforehand, but could potentially be used in a model that constantly updates the projection throughout the game. Several other factors are either difficult or impossible to predict, such as a star player on a team having an off-night and performing much worse than usual. Because of these unknown factors, we will likely encounter errors in our projections due to unknown circumstances that are outside of our control.

4) Methodology for OREB

4.1) Trying different types of models

4.1a) Linear Regression

Before attempting to construct any models, we decided to run a forward stepwise model selection with OREB as the response variable and all cumulative variables in the dataset as the predictors. The variable that correlated most highly with higher count of total offensive rebounds that wasn't either team's individual offensive rebound count was the number of possessions the home and away team had. Following possessions were the home team's free throw percentage, the home team's free throw percentage, the away team's steals, and the away team's 2-point field goal percentage, in that order.

We constructed linear models from these variables using some or all of these variables and got mixed results. The model that performed the best included both teams offensive rebounding stats and possessions stats, as well as the home team's free throw percentage and the away team's steals. This model yielded an MAE of 3.868464 and a RMSE of 4.868937. Neither of these is particularly bad, but they also can be better. The mean residual for this model was 0.0133971, so the residuals were centered fairly accurately around 0, which is good.

4.1b) Poisson

After trying several combinations of variables to fit a simple linear regression model, we moved on to trying a Poisson model. With our Poisson models, we generally did exactly what we did with our linear regression models in terms of which variables we used in each one. As mentioned in the Total Points section, we used Poisson models for a few reasons. Poisson models are generally the best at predicting count data, which total offensive rebounds is, and these models can be flexible with the number of predictive variables used.

The Poisson model that performed the best was one that used home possessions, away possessions, home offensive rebounds, away offensive rebounds, home free throw percentage, away steals, and away 2-point field goal percentage. This model returned an MAE of 3.85916, an RMSE of 4.859788 and a mean residual value of 0.1311088. This MAE and RMSE were slight improvements on our linear regression model, but not by much. The mean residual value got farther from 0 from the linear regression model, but was still not significantly off from 0.

4.1c) Ridge Regression

We used a Ridge regression model to try and predict offensive rebounds in a game. Ridge regression can help prevent overfitting by introducing a penalty term in the loss function, which can improve the generalization performance of the model. It can also be useful in situations where there is multicollinearity between the features, as it can help to stabilize the model and improve the accuracy of the coefficients.

Our Ridge regression model did not perform better than our linear or poisson model. It gave an MAE of 4.020126 and an RMSE of 5.03251. In addition to not improving on anything so far, it also included a prediction that was more than 14 rebounds off, which is extremely bad. Its average residual was pretty close to zero at -0.1137305. In conclusion, we decided not to move forward with the Ridge regression model.

4.1d) Lasso

After considering the Ridge regression model, we moved to try the Lasso regression model. Lasso regression models are traditionally useful for regularization when there are a lot of different aspects being measured in the data. In basketball, there are many different stats being measured, so we thought this type of modeling may be useful.

We used the glmnet package to run our Lasso model. The variables that it selected as useful in modeling were the following: Home 3-point FG %, Home FT %, Home Offensive rebounds, Home Total Rebounds, Home Blocks, Home Turnovers, Home Fouls, Home Possessions, Home 3-point FG missed, Away FT %, Away 2-point FG %, Away Offensive Rebounds, Away assists, Away steals, Away blocks, Away Possessions, Away FG missed, Away 2-point FG missed.

It was interesting that certain variables that would seem very important to predict total number of offensive rebounds were not ones that came up in our forward stepwise selection. Variables such as field goal and free throw misses intuitively matter a lot to getting offensive rebounds. We went back and tested more linear and Poisson models using the variables on missed shots that we didn't use before. To our surprise, the use of these variables as both replacements for other variables in previous models, and as additions to other variables in previous models, did not improve the predictive accuracy of those models.

In terms of the Lasso model, it performed much like other models, but no better than them. It returned an MAE of 3.992619 and a RMSE of 4.993227. Once again, this is not the best model we have created so far. Its mean residual was -0.1718569, which is pretty close to 0, and neither great nor horrible in terms of being centered at 0. We decided against the use of Lasso models for our final predictions.

4.1e) Polynomial

After trying linear regression, Poisson models, Ridge Regression and Lasso regression, we decided to move a different direction and try more linear regression models, but this time with polynomial terms. With our polynomial models, we used less variables than in Poisson and linear regression, so as not to overfit our data. The primary variables we used were possessions and offensive rebounds for the home and away team, home free throw percentage and away steals.

Although no model performed far better than the rest of the models, our best model included offensive rebounds as terms of degree 1, total possessions as terms of degree 2, home free throw percentage of degree 1, and away steals of degree 1. The MAE for this model was 3.867899 and its RMSE was 4.866504. This brings us very close to our best model so far, but does not quite get us there. The average residual was very good at 0.03835411, as that is very close to 0. We considered this to be a good model and did not initially rule it out as our final model.

4.1f) Interaction Terms

The last type of modeling our group tried was introducing interaction terms to capture the non-linearity of certain variables in the dataset. Because this is not a modeling technique in its own right and rather is a tool you can use when constructing multiple different types of models, we introduced interaction terms into both linear regression and Poisson models. We used the same variables that have been used in all of the other models except the Lasso regression. We tried a few different types of interaction terms. The first was simply having the respective home and away versions of different stats interact with each other. The second was having offensive rebounds interact with possession data for home and away, respectively. We tried having other stats such as steals, field goal percentage, and free throw percentage interact with each other, offensive rebounds and possessions, but none yielded results that were any better than the first two methods we tried.

The best model that included interaction terms used possessions, offensive rebounds, home free throw percentage and away steals. It included all of these variables standing by themselves, but included two interaction terms. One of those interaction terms had home offensive rebounds and home possessions, and the other had the corresponding stats but for the away team. The MAE was 3.87543 and the RMSE was 4.869718, which are both decent, but once again, not the best. The mean residual was 0.05179946, which is very close to 0 and a good value for a mean residual to be.

4.2) Choosing our Model

Multiple different factors came into play when it came time to choose our model. We didn;t have one model that was significantly better than all of the other models in terms of predictive accuracy. We decided to choose the model with the lowest MAE and RMSE, even though they weren't the lowest by much. This model was the Poisson model which included possessions and offensive rebounds for the home and away team, home free throw percentage, and away steals and 2-point field goal percentage. The formula to predict offensive rebounds in this model is as follows:

log(OREB) = -0.06295208 + 0.03490808*(Home Offensive Rebounds) + 0.02200707*(Away Offensive Rebounds) + 0.02112948*(Away Possessions) + 0.01663470*(Home Possessions) - 0.61435762*(Home FT%) - 0.02479967*(Away Steals) - 1.03240319*(Away 2FG%)

In the future, it may help to create some sort of offensive rebounding percentage variable that could take into account how many opportunities a team had to get an offensive rebound. This is one aspect of the statistic that our current data does not directly measure. However, this model predicts offensive rebounding totals with general accuracy, taking into account several different variables that directly relate to offensive rebounds in the game of basketball.

4.3) Incorporating Previous Matchup OREB

We also tried to incorporate previous matchup data to make our prediction better. We created a test set. We used function() in R to calculate the weighing factors for previous matchup average OREB and poisson model prediction OREB that could minimize the MAE. It turned out the weighting factor for the previous matchup OREB is -2.032576, which actually showed that we should not incorporate previous matchup OREB averages to balance our predictions.