OIDD 910/ESE 504-402—Project 2

Arnab Sarker, Eric Dong
arnabs@seas.upenn.edu, ericdong@seas.upenn.edu

December 13, 2017

Contents

1 Introduction 2

2 Problem 3
2.1 Objective Function 3
2.2 Constraints 3

3 Linear Programming Model 5
3.1 Model Generation 5
3.2 Formulation 5

4 Data Collection 8
4.1 Player Predictions and Selection Order 8
4.2 Estimated Return Calculation 9

5 Results 10
5.1 Equal Weighting Method 10
5.2 Relative Weighting Method 10

6 Conclusions 11
6.1 Method Comparison 11
6.2 General Conclusions 11

7 Future Considerations 13

8 Sources 14

9 Appendix 15
9.1 Projections for Selected Players 15
9.2 Top 100 Player Averages 16
9.3 The MATLAB Model 17
9.3.1 The k-th Draft Pick 17
9.3.2 Draft Process Script 21
9.3.3 Expected Return Calculation 23
9.4 Original Data 24

1 Introduction

In this project, we develop and test a model to optimize drafting a fantasy basketball team. Fantasy sports have been gaining in popularity over the course of the past few years. They typically include basketball, baseball, football, and hockey. Given that the current National Basketball Association (NBA) season started in mid-October and is now hitting its full stride, finding a way to choose a winning fantasy basketball team is especially relevant.

Fantasy basketball, and fantasy sports in general operate by having a group of participants select teams of actual players through a draft process before the respective sports season begins. The draft process involves participants taking turns to choose from a pool of available players. From there, player performance in games during the actual season contributes to the fantasy teams score in real time. A fantasy teams score will track several metrics and the contribution of each metric to the teams final score varies depending on the rules of the league. In basketball, typical metrics tracked include field goal percentage, free throw percentage, three-pointers made, rebounds, assists, steals, blocks, turnovers, and points. Metrics may be added or excluded depending on league rules.

Regarding counting the metrics towards a final score, common formats include Rotisserie and Head-to-Head scoring formats. In the Rotisserie format, all the teams are considered simultaneously and receive ranks for each metric. For example, if your team has the most points, your rank in the points category is 1. An overall score is then determined by totaling all the rankings. The winning team is the team with the highest overall score. This is the most popular format and for the sake of our analysis, we will only consider this format.

In developing this model, we focused first on determining how to formulate our objective and the constraints. We then moved on to developing an appropriate linear programming formulation. After creating a general model, we found reliable data sources to help us implement and run the linear programming formulation.

Ultimately, we were able to implement an effective strategy for determining the best pick for the k-th pick of the draft, primarily using methods suggested by Becker and Sun [1]. Because the decisions of other players in drafting can be quite chance dependent, and be due to either a different player's strategy, their personal preferences, and the rankings provided by other sources, we also have an outer loop that simulates the other players in the draft. Thus, we provide a strategy for a participant during the draft, as well as a general implementation of our own strategy in the context of an entire draft.

2 Problem

2.1 Objective Function

Our objective, as stated before, is to optimize our drafting process to maximize the likelihood of winning the fantasy basketball season. Because of all of the factors that go into determining who wins in a fantasy league, as well as the variability between teams, we had to make some simplifying and generalizing assumptions. We began with the objective function designed by Becker and Sun [1] for Head-to-Head fantasy play, in which the objective function was a weighted sum of the expected contribution of a player each week, the number of times a player was expected to win throughout the season (using a constant value for the number of points expected to be needed to win), and the number of playoff wins a player could get.

In our model, we did not account for a separate playoff season, because in fantasy basketball, the Rotisserie style of play is used more often than Head-to-Head, so the distinction between a regular season and a playoff game in fantasy play becomes negligible. However, because of the emphasis on categorization in Rotisserie, we did initially develop our objective function as a weighted sum between each player's expected return for each category, and the expected number of wins per week based on a threshold.

After much consideration, and viewing historical data on successful fantasy basketball teams [2], we found that individual success in categories was not particularly correlated with overall success in a team. That is, although Rotisserie scoring is based on success in specific categories, the overall quality of a team matters much more in determining whether or not a team will do well. Thus, we instead calculated a measure of "expected return" for each player based on their projected statistics, to provide a reasonable weighting that could then correlate with success in general as a fantasy basketball team. Interestingly, this brings our model to become well-suited for Head-to Head play as well, which makes the linear programming formulation more generalizable, depending on the choice of the expected return function.

2.2 Constraints

The constraints for our linear programming model are primarily due to the constraints set by the draft process. In a traditional NBA draft, there are constraints on the number of players of each position that are allowed on a team. In our model, we have placed constraints such that any team must have at least one player in each position (Point Guard, Shooting Guard, Small Forward, Power Forward, and Center), and we also provide limits on the number of players of each position that can be on a team. For example, because Point Guards tend to contribute a large amount to different categories for a fantasy team, we limit each team to choose up to five point guards of the ten selections on the team. While the number is still quite high, because the maximum number of players on a team is ten, we still expect this constraint to be binding. In fact, these constraints provide an excellent path for sensitivity analysis.

The fantasy basketball draft also imposes an ordering on how drafting is done. In fantasy basketball, a "snake" draft format is used, in which the order of picks in the first round reverses in the second round, and then reverts back in the third round, and the reversals repeat each round. Thus, for example, the player with the first pick has the first pick, twentieth pick, twenty-first pick, fortieth pick, and so on. We must take account for this when we see what draft pick is currently being optimized.

An additional constraint that was added has to do with the expected number of games won. While having high scoring players is advantageous, players often miss weeks due to injury, or have off days, causing variability from week to week. Thus, we provide an indicator for whether we would expect our team to win in a certain week or not. In the implementation, we found that the week to week variability created many issues with the effectiveness of the linear program, so for implementation purposes we simplified our model to have less week to week variation for the players. However, although we did not artificially include the week to week variability, we did leave the constraint in the formulation of the problem, as this would provide an excellent starting point for considerations of this problem in the future.

The most difficult constraints on our drafting are those implicitly imposed by other players. In our linear programming model, in order to select the best option for our players, we also want to make sure that we do not ignore any potential future options for players. That is, taking a "greedy" approach and only selecting the player with the best expected return isn't necessarily the most appropriate way to select for each round. For example, if we are about to reach the limit on the number of Shooting Guards in the team, it may ultimately be beneficial to wait for a Shooting Guard in the next round rather than use up all potential spots for Shooting Guards and be forced to take a worse player later. Thus, in our model, we create a heuristic by which we anticipate other players to select their players, and our model optimizes accordingly for the remaining rounds in the draft.

3 Linear Programming Model

3.1 Model Generation

Our linear programming model was designed iteratively, based first on our initial hypothesis of what indicators make a fantasy basketball team likely to win. Initially, we believed that, because the Rotisserie scoring system was used, we would want to optimize the players based on the categories. This idea is very intuitive, as it makes sense to take a direct approach to maximizing each category. However, after looking at historical trends, we saw that individually maximizing each category would be no better of an indicator for a good team than simply picking the best players from historical data.

The evidence for this simplification mainly comes from the data, in which we found a strong correlation between teams that had a "balance" of very good, well-rounded players and overall success in fantasy basketball. Additionally, it is worth noting that players that are particularly good in certain positions will contribute to the relevant category regardless. For example, the best centers in the NBA generally have the best rebounds per game.

The simplification is particularly useful in reducing the number of variables involved in our linear program, which is useful given the extraordinarily large worst-case run times of linear programs. It is also useful in generalizing our model to optimization for Head-to-Head fantasy play and even other fantasy sports.

The final design for the linear programming model is based primarily in the methods proposed by Becker and Sun [1], with several tweaks specific to Rotisserie style of play and fantasy basketball as opposed to fantasy football.

3.2 Formulation

The model we built seeks to create a winning season long fantasy basketball team. We closely follow the process outlined by Becker and Sun to create a model that not only drafts the best team, but also plays the optimal starting line up each week, including in the playoffs. We start by defining the following variables and parameters:

$$
\begin{aligned}
N &:= \text{ Set of NBA players} \\
M &:= \text{ Set of positions, } M = \{PG, SG, SF, PF, C\} \\
&\quad \text{Steals, 3-pointers, Assists, Field goal percentage, Free throw percentage}\} \\
T &:= \text{ Set of weeks in the NBA regular and playoff seasons, } T = \{1, ..., 21\} \\
Pos(i) &:= \text{ Position of player } i \in N, Pos(i) \in M \\
PosLimit(j) &:= \text{ Upper bound on the number of starting players for position } j \in M \\
&\quad \text{during weekly and playoff play} \\
n_k &:= \text{ Overall pick number of participant's } k\text{-th draft pick} \\
Players(k) &:= \text{ Set of players the participant has drafted by their } k\text{-th pick}
\end{aligned}
$$

$$
\begin{aligned}
OppPlayers(k) &:= \text{ Set of players the opposing teams have drafted by their } k\text{-th pick} \\
R_k(i) &:= \text{ Anticipated ranking of unselected player } i \text{ at the participant's } k\text{-th pick} \\
f(i,t) &:= \text{ Estimated score for player } i \in N \text{ on week } t \in T \\
\beta(t) &:= \text{ Estimated score that the participant needs} \\
&\quad \text{to be reasonably confident to win on week } t \in T \\
\gamma_j &:= \text{ Number of players the participant must draft at position } j \in M \\
y_i &:= \text{ Binary variable, 1 if the participant drafts player } i\text{, 0 otherwise} \\
x_i^t &:= \text{ Binary variable, 1 if the participant plays player } i \text{ on week } t\text{, 0 otherwise} \\
z^t &:= \text{ Binary variable, 1 if the participant's total score} \\
&\quad \text{on week } t \text{ is greater than } \beta(p,t)\text{, 0 otherwise}
\end{aligned}
$$

Many of our parameter values are given by ESPN and the rules for their fantasy league play, including following the format of a snake draft. Our universe of players, N, are the 187 NBA players who's projected season statistics are provided on ESPN. Some players may have multiple positions. We set the values of $f(i,t)$ based on our calculation of expected value return of a player using ESPN's player projections. We also determine the values of $R_k(i)$ based on ESPN's forecasted average draft position (ADP). Our model is meant to be used dynamically as one drafts and plays through the fantasy season. In other words, $Players(k)$, $OppPlayers(k)$, and $R_k(i)$ are updated for each drafting round so that our model will select the best remaining player at every round and $\beta(t)$ is updated for each week of play depending on the participant's opponent so that our model will choose the best weekly and playoff lineups. Thus the following model below provides the optimal draft decision at each draft pick:

$$
\begin{aligned}
\text{For } k\text{-th draft pick: } \max \sum_{t=1}^{21} \sum_{i \in N} f(i,t) x_i^t + \sum_{t=1}^{21} z^t \text{ s.t.} \\
\sum_{i \in N : R_k(i) \le \alpha(n_{\bar{k}} - n_k)} y_i &\le \bar{k} - k \ \forall \bar{k} \ge k+1 \\
\sum_{i \in N : Pos(i) = j} y_i &\ge \gamma_j \ \forall j \in M \\
\sum_{i \in N : Pos(i) = j} x_i^t &\le PosLimit(j) \ \forall j \in M, t \in T \\
x_i^t &\le y_i \ \forall i \in N, t \in T \\
z^t &\le \frac{\sum_{t \in T} f(i,t) x_i^t}{\beta(t)} \ \forall t \in T \\
y_i &= 1 \ \forall i \in Players(k) \\
y_i &= 0 \ \forall i \in OppPlayers(k) \\
y_i, x_i^t, z^t &\in \{0,1\} \ \forall i \in N, t \in T
\end{aligned}
$$

The objective function is two-fold. It aims to not only maximize the total season fantasy score of each utilized player from each week (represented by $\sum_{t=1}^{21}\sum_{i\in N}(f(i,t)x_i^t)$, but it also seeks to maximize weekly head-to-head wins in each week (represented by z^t). Note that we calculated each player's total fantasy score two different ways: one where each scoring category was given equal weighting and another where each scoring category was normalized to the average of the top 100 players. And since within a given weekly head-to-head match up, there will be several NBA games, meaning we can assume $\beta(t)$ is constant for all t, thereby making z^t time invariant.

The first constraint serves to restrict the participant's draft picks based on opponent picks. In other words, up to the $\bar{k}$-th pick, the participant can pick no more than $\bar{k}-k$ players from the top $\alpha(n_{\bar{k}}-n_k)$ players based on their ranks, $R_k(i)$. In our model, we will make the reasonable assumption that $\alpha = 1$. Thus, this is the equivalent of saying the opponents will draft based on $R_k(i)$. For example, in a 10 team league, consider when the participant has the 2nd and the 19th overall draft picks (a.k.a first and second round picks, or $k=1, \bar{k}=2$). After the participant makes his or her first round pick, given the newly updated ranks, $R_k(i)$ based on the remaining set of available players, we are assuming the next 16 picks which are from the opponents will be the remaining top 16 ranked players so that the participant's second round pick can not include any of those chosen players. However, note that this does not mean the participant's second round pick must then be the 17th ranked player among the remaining players, just that the opponents will draft according to $R_k(i)$.

The second constraint ensures that the participant drafts the requisite number of players for each position.

The third constraint ensures that each weekly lineup does not exceed the limit on the number of players allowed in each position.

The fourth constraint prevents the participant from using players in the weekly lineup that he or she did not draft.

The fifth constraint defines z^t. It forces $z^t = 1$ when the weekly lineup's total score in week t exceeds the threshold defined by $\beta(t)$, and $z^t = 0$ otherwise.

The sixth and seventh constraints ensure that the participant can not draft players that have been picked by the opponents.

The last constraint defines x, y, and z as binary variables.

As discussed earlier, this model is meant to be run for each round of the draft using updated information. The model then provides the optimal team, from which the participant actually drafts the highest ranked player (largest $R_k(i)$) who has not already been drafted yet. In other words, we will use an algorithm that will loop through this model for each round of the draft. After each loop, the model will provide a new suggested optimal team, from which we will choose the best undrafted player. Then we will update the universe of available players, the players on our team, the players drafted by opponents, and the remaining relative ranks of the available players before the next iteration.

4 Data Collection

4.1 Player Predictions and Selection Order

For the implementation of this particular model, we collect our data based on ESPN projections [3], as well as ESPN draft data [4]. There are multiple reasons as to why we selected data from one source. The first of these reasons is for consistency. If we were to use multiple different data sources for predictions for players, there would be too much left to interpretation as to how much weight to give each of the sources. For example, ESPN, as a large facilitator of fantasy sports, likely has fairly established methods for sports projections, whereas other, smaller sports fan pages may have projections that are very different and not necessarily as accurate. While it seems logical to put more weight on the ESPN predictions, the specifics of such a task could greatly change the outcomes of the model. Another reason for only using one source for the data is because ESPN's data is often the most significant data used by most fantasy sports participants in general. Thus, we are simulating a more realistic version of a participant's optimization in a fantasy basketball draft.

The NBA has 301 players, which would lead to a huge expected amount of time to run our model. We thus needed to truncate our model to only work with players that were likely to be drafted. Therefore, in creating the data set, we tried to focus on the players that were most likely to be drafted. We made a simplifying assumption that, because our draft final number would be no more than 100, we would not expect to take any players beyond the first 200 most drafted players. Depending on the results of the linear programming model, we could apply sensitivity analysis to see if adding more players to data set would create any significant changes in the final drafted team. While the amount of data forced us to remove some of the players towards the lower end of the drafting range, the lowest ranked player with no projection data available had a draft rank over 150, so depending on the results, we would not expect this absence to significantly change the results of the draft selection.

To compile the data, we merged the two aforementioned data sets from ESPN. In the linear programming model, we used the draft data to give an order to the decisions in which we did the drafting. This provides us with information as to a player's "rank" according to other players, as well as their expected return value according to our own estimation.

4.2 Estimated Return Calculation

We used several different expected return value functions in determining the expected returns of players. For each of the expected return values, we use projected number of 3 pointers made, rebounds, assists, steals, blocks, points, free throw percentage, and field goal percentage to influence a player's value.

In the first of our trials, we weighted all of the expected projections equally. This was an attempt to "mimic" the original formulation for having different Rotisserie categories be weighted the same.

In the next trial, we used a normalization from the first 100 players in order to determine the relative weight of each statistic. We then used a weighted sum to determine the expected return of each player. We chose this normalization to try and set a baseline for how a player is performing in a specific category.

This is certainly the part of the model that is up to the most interpretation. However, this idea of an expected return calculation makes the model very generalizable, and does provide a way to reduce the number of decision variables, as each potential scoring category no longer has a variable associated with it for each week.

5 Results

5.1 Equal Weighting Method

When each projection is given an equal weight, our algorithm selects the following team:

Table 1: Team Selection With Equal Weighting

Player Name	Position	Actual Pick #	Calculated Rank	Average Pick (ESPN)
Anthony Davis	Power Forward	4	5	6
Damian Lillard	Point Guard	21	17	16.8
Bradley Beal	Shooting Guard	28	24	25.7
Kevin Love	Power Forward	45	37	40.2
DeAndre Jordan	Center	52	44	44.7
Nikola Vucevic	Center	69	57	58
Jae Crowder	Small Forward	76	64	64.8
Marquese Chriss	Power Forward	93	77	81.4
Derrick Rose	Point Guard	100	85	90.3
Greg Monroe	Power Forward	117	105	103

5.2 Relative Weighting Method

When each projection is given an relative weights based on player averages in each statistic, our algorithm selects the following team:

Table 2: Team Selection With Equal Weighting

Player Name	Position	Actual Pick #	Calculated Rank	Average Pick (ESPN)
Stephen Curry	Point Guard	4	6	6.3
Damian Lillard	Point Guard	21	17	16.8
Bradley Beal	Shooting Guard	28	24	25.7
Kevin Love	Power Forward	45	37	40.2
DeAndre Jordan	Center	52	44	44.7
Nikola Vucevic	Center	69	57	58
Jae Crowder	Small Forward	76	64	64.8
Marquese Chriss	Power Forward	93	77	81.4
Derrick Rose	Point Guard	100	85	90.3
Greg Monroe	Power Forward	117	105	103

Full data for each of these players is provided in the Appendix, as are averages for the top 100 players considered.

6 Conclusions

6.1 Method Comparison

Overall, both methods for our model performed very strongly and very similarly. Both methods picked closely to ESPN's Average Pick metric and chose players who either had a balanced set of statistics or were exceptional within one or two areas.

The methods deviated from each other only on the first pick, where the equal weighting method chose Anthony Davis and the relative weighting method chose Stephen Curry. This is likely because Anthony Davis especially excels in rebounding and scoring which are two relatively high counting categories where as Stephen Curry specializes in three point shooting as well as high shooting percentages which are low counting categories. Thus under the equal weighting method, metrics like rebounding and scoring are inflated and extra emphasis is placed on them, making Anthony Davis a more desirable player. Under the relative weighting method, deviations from the average in low counting categories such as shooting percentages and three point shooting become much more stark, thereby placing more weight on those categories.

6.2 General Conclusions

To test our model's draft selection power, we compared the average team player statistic for each scoring category under both methods against that of a winning team from the 2015-2016 season. For the purpose of this comparison, we mimicked the settings of this winning team's league in our own model such as having the same first draft pick, same size team, etc. The following table illustrates our performance. From it, we can see that the teams we have chosen perform at the same level as or even better than the winning team from the 2015-2016 season at each scoring category. This supports the strength and validity of our model's ability to build winning teams. As mentioned, the equal weighting method tends to perform better at high counting categories while the relative weighting one tends to perform better with the low counting ones.

Table 3: Comparison of Teams by Scoring Category

Category	Equal Weighting	Relative Weighting	2015-2016 Winning Team
FG%	0.492	0.490	0.470
FT%	0.765	0.774	0.798
3-pointers	1.150	1.530	1.177
Rebounds	6.990	6.410	6.241
Assists	2.780	3.210	3.748
Steals	1.030	1.090	1.132
Blocks	0.790	0.590	0.862
Points	16.260	16.180	15.137

Our model yields many interesting insights. It is fascinating that our results from both methods matched ESPN's Average Pick metric so closely. This not only reaffirms the strength of ESPN's metric, but also reaffirms the accurateness of our own model. Our model, through mathematical analysis, was able to select a team on par with a team that would have been selected using the wisdom of the crowds.

Another interesting point of discussion is that the results of our model are very similar despite using two very different methods for determining the expected value return of a player. The reasons for having different first picks were discussed above. We postulate that all the remaining picks were the same though for both methods because of a relatively high level of homogeniety outside of the top 10% of the league. This occurs for a few reasons. There is a sharp drop off after we get past the most elite players in the league in terms of their statistical measures. Such a drop off makes it harder for non-elite players to distinguish themselves and thus creates homogeniety between players. Also, elite players tend to specialize in certain categories whereas non-elite playeres are more average across the board. This creates homogeniety within players and also makes our first draft pick especially sensitive to how we choose to calculate expected value return of a player and how that affects which categories are overweighted. Despite having later draft picks be relatively resistant to our expected value return calculation, our model has still managed to select the optimal draft picks.

7 Future Considerations

While our model takes into consideration the key components of the fantasy basketball drafting process, there is still quite a bit that can be done in terms of making our model more specific to fantasy basketball, handling variability from week to week, and determining more appropriate expected return values.

As far as handling basketball specific applications, it would be interesting to test out the model using the original method of having an objective function based on Rotisserie scoring. While the current linear programming formulation has fewer decision variables and is generally easier and less computationally heavy to solve, it would be interesting to see the difference between the two models, and then determine which of the models generally performs better. There may certainly be a trade-off between computational efficiency and accurate results for this type of linear programming formulation, but at the same time, so many of the variables are chance dependent so it is not as clear as to whether one formulation is strictly better than another. It would also be interesting to look at this linear programming model where the decisions of other players are simulated differently. Right now, we give other participants a fixed strategy in the implementation of the linear program, but in real life, this is clearly not the case.

These chance dependent variables from week to week would be very interesting to add to the linear programming formulation of the problem. In the future, this model could be adapted to include some chance variables. For example, for certain points in time, coefficients could be selected uniformly at random to determine if a player may be injured that week or not be able to play. While the randomness may not coincide with the actual events, it would be interesting to see how the model would change with chance dependent variables, especially because many of the projections and values that are fed in are already calculated based on probabilistic methods.

Furthermore, because the expected value return of a player can be different depending on available data, it would be interesting to adjust the algorithm in the future. There are certainly large scale data analysis techniques that could be used to create accurate estimates of how a player may perform throughout the season. Additionally, this expected value return would be interesting to view in terms of other fantasy sports, such as football, golf, and hockey. A clear next step in this area that does not involve heavy computation is to calculate the z-score of every player's statistics and normalize using that. As mentioned earlier, expected value return of a player is very subjective and can be calculated in many ways.

Ultimately, the fact that our model is so generalizable makes it open to a lot of considerations in the future, by approaching it with different constraints on the specifics of drafting.

8 Sources

[1] Adrian Becker and Xu Andy Sun. *An analytical approach for fantasy football draft and lineup management*. J. Quant. Anal. Sports 2016; 12(1): 1730

[2] Zack Rewis *2015-16 Fantasy Basketball: Rotisserie Draft Strategy*. http://thefantasyfix.com/fantasy-basketball/2015-16-fantasy-basketball-rotisserie-draft-strategy/

[3] ESPN *Complete 2018 Projections* http://games.espn.com/fba/tools/projections?display=alt

[4] ESPN *ESPN Live Draft Results* http://games.espn.com/fba/livedraftresults

[5] David Hunter, Juan Vielma, and Tauhid Zaman *Picking Winners Using Integer Programming* http://www.mit.edu/ jvielma/publications/Picking-Winners.pdf

[6] F. Bonobo, G. Duran, and J. Marenco *Mathematical programming as a tool for virtual soccer coaches: a case study of a fantasy sport game* International Transactions in Operational Research May 2014; 21(3): 399-414

9 Appendix

9.1 Projections for Selected Players

The projections for the players selected to be on teams is shown in the table below.

Name	FG%	FT%	3 PTers Made	Rebounds	Assists	Steals	Blocks	Points
Anthony Davis	0.517	0.817	0.7	10.5	2.5	1.5	2.2	26.3
Steph Curry	0.488	0.909	4.5	4.7	6.8	2.1	0.2	25.5
Damian Lillard	0.438	0.892	2.9	4.4	6.6	1.1	0.3	24.6
Bradley Beal	0.471	0.818	2.7	3.6	3.8	1.3	0.3	22.1
Kevin Love	0.431	0.833	2.1	9.7	2.2	0.9	0.4	15.7
DeAndre Jordan	0.689	0.463	0	13.5	1.2	0.8	1.9	12.3
Nikola Vucevic	0.492	0.741	0.2	9	2.7	0.9	0.8	15.7
Jae Crowder	0.45	0.826	1.4	4.1	1.7	1	0.3	9.8
Marquese Chriss	0.455	0.662	1	5.2	1.5	1.2	1	11.4
Derrick Rose	0.455	0.848	0.5	2.8	3.4	0.6	0.2	12.5
Greg Monroe	0.526	0.751	0	7.1	2.2	1	0.5	12.2

9.2 Top 100 Player Averages

We calculated the statistical averages for the top 100 players in each scoring category.

FG%	FT%	3 Pointers Made	Rebounds	Assists	Steals	Blocks	Points
0.47	0.78	1.33	6.12	3.69	1.17	0.69	16.27

9.3 The MATLAB Model

9.3.1 The k-th Draft Pick

```
function [pick, x, y, z] = kth_draft_pick(N, k, num_picks, pick_number,
           teams, selected_players, remaining_players)

alpha = 1;
% SET VARIABLES
beta = ones(21, 1);
gamma = ones(5, 1);
pos_lim = ones(5, 1) * 5;
pos_lim(5) = pos_lim(5) - 1;
[n, ~] = size(N);
pos = N(:, 1);
est_score = N(:, 2);
est_score = repmat(est_score, 21);
est_score_f = est_score(:, 1);
est_score = est_score(1:n, :);
[num_players, ~] = size(selected_players);
opp_players = 1:n;
ranks = remaining_players(:, 3);
non_opp_players = [ranks; selected_players.'];
opp_players(non_opp_players) = [];
opp_players_cell = num2cell(opp_players);
opp_players_cell(~cellfun('isempty', opp_players_cell));
opp_players = cell2mat(opp_players_cell);
[num_opp_players, ~] = size(opp_players);

% objective function
f = [est_score_f; zeros(n, 1); ones(21, 1)];
f = f .* -1;

num_constraints = 21 + 21 * n + 21 * 5 + 5 +
   num_picks - k + num_players + num_opp_players;
num_vars = n * 21 + n + 21;

% lhs coefficients
A = zeros(num_constraints, num_vars);

% rhs coefficients
b = zeros(num_constraints, 1);

% 1st constraint coefficients
```

```
row = 1;
for k_bar = k+1:num_picks
    for i = 1:n
        % find overall picks
        nk = 0;
        nk_bar = 0;
        if mod(k, 2) == 1
            nk = pick_number + (k - 1) * teams;
        else
            nk = teams * k - pick_number + 1;
        end
        if mod(k_bar, 2) == 1
            nk_bar = pick_number + (k_bar - 1) * teams;
        else
            nk_bar = teams * k_bar - pick_number + 1;
        end
        if ismember(i, ranks)
            if find(ranks == i) <= alpha * (nk_bar - nk)
                A(row, n * 21 + i) = 1;
            end
        end
    end
    b(row) = k_bar - k;
    row = row + 1;
end

% 2nd constraint coefficients
for j = 1:5
    for i = 1:n
        if pos(i) == j
            A(row, n * 21 + i) = -1;
        end
    end
    b(row) = -1 * gamma(j);
    row = row + 1;
end

% 3rd constraint coefficients
for j = 1:5
    for t = 1:21
        for i = 1:n
            if pos(i) == j
                A(row, i + n * (t - 1)) = 1;
```

```
            end
        end
        b(row) = pos_lim(j);
        row = row + 1;
    end
end

% 4th constraint coefficients
for t = 1:21
    for i = 1:n
        A(row, i + n * (t - 1)) = 1;
        A(row, n * 21 + i) = -1;
        row = row + 1;
    end
end

% 5th constraint coefficients
for t = 1:21
    A(row, n * 21 + n + t) = 1;
    for i = 1:n
        A(row, i + n * (t - 1)) = -1 * est_score(i, t) / beta(t);
    end
    row = row + 1;
end

% 6th constraint coefficients
for i = selected_players
    A(row, n * 21 + i) = -1;
    b(row) = -1;
    row = row + 1;
end

% 7th constraint coefficients
for i = opp_players
    A(row, n * 21 + i) = 1;
    b(row) = 0;
    row = row + 1;
end

intcon = 1:numel(f);
lb = zeros(numel(f), 1);
ub = ones(1, numel(f));
Aeq = zeros(1, numel(f));
```

```
beq = 0;

results = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub);
x = results(1:21 * n);
y = results(21 * n + 1:21 * n + n);
z = results(21 * n + n + 1:end);
picks = y;
if ~isempty(selected_players)
    picks(selected_players) = 0;
end
pick = find(picks, 1);

end
```

9.3.2 Draft Process Script

```
% Read player stats
all_player_stats = csvread('PlayerStats.csv', 1, 3);

% This is a matrix containing player position and expected points per week
all_player_returns = generate_player_returns(all_player_stats);
num_total_players = length(all_player_returns);
all_player_returns = [all_player_returns cumsum(ones(num_total_players, 1))];

% Global Constants
draft_total = 10; % Number of players on team
num_participants = 12; % Number of participants in the draft
pick_number = 4; % The pick the player has in the first round

% Set variables that will change with each loop
num_drafted = 0; % number of players drafted so far
selected_players = []; % location of each player in original CSV file
curr_pick = pick_number; % The current pick that the player has
num_taken = 0; % The number of players that have been removed from the draft thus far

% Draft people that are selected before our player can pick:
remaining_players = all_player_returns(pick_number:end, :);
num_taken = pick_number - 1;


% Simulate all picks but the last
while(num_drafted < draft_total)
    % Step 1: Pick a player from the remaining players

    % Inputs :
    %    All players, players drafted so far on our team, estimated
    %    score, current pick number, total team size, unselected players,
    %    original draft pick, number of people drafting

    % Output:
    %    The player index of from the truncated matrix file
    [pick, ~, ~, ~] = kth_draft_pick(all_player_returns, curr_pick, draft_total, ...
        pick_number, num_participants, selected_players, remaining_players);

    % Step 2: Add that pick to our team
    % Order matters here
    num_drafted = num_drafted + 1;
```

```
    selected_players(num_drafted) = pick;

    % Step 3: Eliminate that pick from contention:
    relative_pick = find(remaining_players(:, 3) == pick);
    if (relative_pick == 1)
        % No need to concatenate if we had the first player
        remaining_players = remaining_players(2:end, :);
    else
        % Otherwise, take the players above and below and concat
        remaining_players_above = remaining_players(1:relative_pick - 1, :);
        remaining_players_below = remaining_players((relative_pick + 1) : end, :);

        remaining_players = ...
                vertcat(remaining_players_above, remaining_players_below);
    end
    num_taken = num_taken + 1;

    % Step 4: Remove players who are "drafted" until the next pick
    if(mod(num_drafted, 2) == 1)
        pick_difference = 2 * (draft_total - pick_number);
    else
        pick_difference = 2 * (pick_number - 1);
    end
    remaining_players = remaining_players(pick_difference + 1:end, :);
    num_taken = num_taken + pick_difference;
    curr_pick = curr_pick + pick_difference;

end

% Simulate the last pick, which will also provide a schedule
[pick, x, y, z] = kth_draft_pick(all_player_returns, curr_pick, draft_total, ...
        pick_number, num_participants, selected_players, remaining_players);
selected_players(num_drafted) = pick;

player_names = readtable('PlayerStats.csv');
player_names = table2array(player_names(:, 2));
roster = player_names(selected_players);
fprintf('You had the %dth draft pick\n', pick_number);
disp('Your optimal team is:')
disp(roster)
```

9.3.3 Expected Return Calculation

Note that this function changed depending on how we wanted to determine what the expected return of a player would be.

```
function [player_returns] = generate_player_returns(player_stats)
player_returns = zeros(length(player_stats), 2);

% Player positions
player_returns(:, 1) = player_stats(:, 1);

% This is an expected return calculation based on previous data
avg_pick = player_stats(:, 2);
exp_3pts = player_stats(:, 9);
rebounds = player_stats(:,10);
assists = player_stats(:,11);
steals = player_stats(:,12);
blocks = player_stats(:,13);
points = player_stats(:,14);
fg_percent = player_stats(:,8);

player_returns(:, 2) = exp_3pts / 1.33 + rebounds / 6.12 + ...
                       assists / 3.69 + steals / 1.17 + blocks / 0.69 + ...
                       points / 16.27 + fg_percent / 0.47 + ft_percent / 0.78;
end
```

9.4 Original Data

The original ESPN Player Projections is given in the links. After compiling the data, we get the following results, with 13 players having draft ranks but no projection data, or vice versa. This leaves us with 187 players with full information.

```
Rank Player Team Position Avg Pick Auction Draft Avg Val %Own FG% FT% 3 Pointers Made
Rebounds Assists Steals Blocks Points
1 Giannis Antetokounmpo  Mil 3 1.9 72.4 100 0.521 0.757 0.6 9.3 5.3 1.7 1.5 22.2
2 Russell Westbrook  OKC 1 2.3 75.2 100 0.437 0.843 1.9 8.9 9.7 2 0.3 26.4
3 James Harden  Hou 2 3.6 70.3 100 0.445 0.864 3.2 7 8.7 1.7 0.5 27.9
4 Kevin Durant  GS 3 4.8 64.2 99.9 0.524 0.885 2.3 7.9 4.8 1.1 1.3 25
5 Anthony Davis  Nor 4 6 60.8 100 0.517 0.817 0.7 10.5 2.5 1.5 2.2 26.3
6 Stephen Curry  GS 1 6.3 58.6 99.9 0.488 0.909 4.5 4.7 6.8 2.1 0.2 25.5
7 Karl-Anthony Towns  Min 5 6.4 59.3 100 0.557 0.835 1 12.9 2.8 0.9 1.4 22.3
8 LeBron James  Cle 3 6.8 56.7 100 0.525 0.706 1.5 7.4 7.5 1.4 0.6 22.8
9 Kawhi Leonard*  SA 3 9.5 58.1 99.7 0.487 0.862 1.9 6.6 3.3 1.9 0.9 21.5
10 John Wall*  Wsh 1 10.7 49.1 99.9 0.452 0.808 1.2 4.8 10.3 2 0.7 22.2
11 Kyrie Irving  Bos 1 12.8 39.6 99.9 0.471 0.906 2.3 3.6 5.5 1.4 0.3 23
12 Nikola Jokic  Den 4 13.2 45.3 99.7 0.562 0.813 0.7 10 5.2 1.2 0.8 17.7
13 DeMarcus Cousins  Nor 5 13.8 43.1 100 0.472 0.766 1.3 10.9 4.1 1.6 1.3 24.4
14 Rudy Gobert*  Utah 5 13.8 44.7 92.3 0.604 0.647 0.1 12 1.6 0.8 2.4 14.1
15 Chris Paul  Hou 1 15 41.7 98.6 0.473 0.897 1.9 4.4 9.3 2 0.1 18.1
16 Hassan Whiteside  Mia 4 16.7 39.9 99.4 0.553 0.635 0 14.1 0.9 0.8 2.7 16.3
17 Damian Lillard  Por 1 16.8 36.9 99.9 0.438 0.892 2.9 4.4 6.6 1.1 0.3 24.6
18 Jimmy Butler  Min 2 17.4 37.1 99.9 0.473 0.85 1.1 5.6 4.9 1.8 0.4 20
19 Paul George  OKC 3 18.2 35.6 99.9 0.451 0.885 2.7 6.8 4.2 1.8 0.4 23.7
20 Gordon Hayward*  Bos 2 21.8 30.3 18.1 0.563 0.57 0 3.8 0.7 0.4 0.4 3.5
21 CJ McCollum  Por 1 22.7 32.3 99.6 0.465 0.88 2.7 3.5 4.1 1.2 0.4 21.7
22 Draymond Green  GS 3 24.4 29 99.6 0.456 0.708 1.3 8.1 6.2 1.8 1.2 11.5
23 Kristaps Porzingis  NY 4 25.7 26.9 99.9 0.445 0.819 1.5 8.1 2 0.9 1.9 19.7
24 Bradley Beal  Wsh 2 25.7 27.6 99.7 0.471 0.818 2.7 3.6 3.8 1.3 0.3 22.1
25 Klay Thompson  GS 2 26.2 26.3 99.9 0.478 0.863 3.4 3.6 2.5 0.9 0.6 21.5
26 Kemba Walker  Cha 1 27 29.2 99.7 0.432 0.846 2.5 3.6 5.6 1.4 0.3 21
27 Myles Turner  Ind 4 27.4 26.4 96.7 0.51 0.823 0.5 8.2 1.8 1 2.1 17.1
28 Kyle Lowry  Tor 1 28 26.5 99 0.429 0.825 2.6 5 6.5 1.8 0.3 19.1
29 DeMar DeRozan  Tor 2 28.7 22.9 99.8 0.449 0.842 0.5 5.1 4 1.1 0.2 23.1
30 Blake Griffin  LAC 4 32.3 21.4 99.9 0.501 0.757 0.5 7.9 5.1 1 0.4 22.4
31 Joel Embiid  Phi 5 33 21.8 99.5 0.471 0.787 1.1 6.7 2.5 1 2.2 17.2
32 Mike Conley*  Mem 1 33.1 23.8 84.5 0.452 0.863 2 3.8 5.8 1.4 0.3 18.5
33 Marc Gasol  Mem 5 33.4 21.6 99.4 0.466 0.819 0.9 7.4 4.2 1 1.4 17.5
34 Khris Middleton  Mil 3 35 20.7 97.7 0.457 0.895 2 4.8 4.1 1.7 0.2 17.6
35 Carmelo Anthony  OKC 3 35.1 17.2 99.4 0.438 0.825 1.8 6.1 3.3 1 0.5 19.6
36 Paul Millsap*  Den 4 35.7 19.8 80.8 0.462 0.76 1.2 7.8 3.7 1.7 1.1 17.4
```

```
37 Kevin Love  Cle 4 40.2 17.8 99.5 0.431 0.833 2.1 9.7 2.2 0.9 0.4 15.7
38 Jusuf Nurkic  Por 5 40.4 17.5 90.5 0.502 0.618 0.1 9.5 2.7 1.3 1.6 14.5
39 Andrew Wiggins  Min 2 41.3 13.8 98 0.474 0.769 1 4.5 2.6 1.2 0.5 21
40 Eric Bledsoe  Mil 1 41.4 20.2 91.4 0.444 0.836 1.3 4.9 6 1.6 0.5 18.7
41 Devin Booker  Pho 2 41.9 15.7 99 0.428 0.831 2 3.5 3.9 1.1 0.3 21.7
42 Otto Porter Jr.  Wsh 3 44.1 16.2 97.8 0.492 0.812 1.8 5.9 1.8 1.5 0.5 13.1
43 Jeff Teague  Min 1 44.6 15.6 93.9 0.454 0.854 1.2 3.8 7.9 1.6 0.5 15.5
44 DeAndre Jordan  LAC 5 44.7 15.3 97.3 0.689 0.463 0 13.5 1.2 0.8 1.9 12.3
45 Brook Lopez  LAL 5 45.1 16 89.2 0.483 0.811 0.9 6.2 2 0.7 1.5 18.2
46 Al Horford  Bos 5 47.1 13.3 97 0.494 0.802 1.1 7.4 4.3 0.9 1.3 13.8
47 LaMarcus Aldridge  SA 4 48.7 10.9 97.7 0.486 0.833 0.3 7.9 1.9 0.7 1 16.9
48 Ben Simmons  Phi 4 49 10.7 98 0.494 0.693 0.1 7.7 2.9 1.1 0.5 11.1
49 Dennis Schroder  Atl 1 50 13.9 94.2 0.439 0.822 1.3 3.7 5.9 1 0.1 20.1
50 Goran Dragic  Mia 1 50.4 15.6 93.4 0.462 0.79 1.2 3.8 5.2 1.1 0.2 16.4
51 D'Angelo Russell*  Bkn 1 51.8 9.2 82.5 0.405 0.773 2.3 4.1 4.6 1.6 0.3 17.6
52 Andre Drummond  Det 5 52.2 11.6 96.8 0.524 0.401 0.1 13.4 1.1 1.5 1.2 14.1
53 Jrue Holiday  Nor 1 53.5 11.1 90.4 0.457 0.78 1.4 3.5 6.8 1.6 0.5 15.3
54 Lonzo Ball  LAL 1 53.9 10.6 92.6 0.408 0.697 1.3 3.9 5 1.2 0.3 8.4
55 Elfrid Payton  Orl 1 55.3 11.4 67.3 0.46 0.661 0.5 4.6 6.6 1.4 0.4 12.5
56 Serge Ibaka  Tor 4 57.9 11.7 80.6 0.461 0.816 1.4 6.7 1.1 0.5 1.5 12.2
57 Nikola Vucevic  Orl 5 58 9.8 93.5 0.492 0.741 0.2 9 2.7 0.9 0.8 15.7
58 Ricky Rubio  Utah 1 59.8 9.8 81.9 0.387 0.877 1.1 4.5 8.2 1.8 0.1 13
59 Gorgui Dieng  Min 5 60 9 37.6 0.53 0.831 0.1 6.8 1.9 0.9 0.9 7.8
60 Harrison Barnes  Dal 3 63.8 6.5 90.7 0.45 0.819 1.4 5.8 1.7 0.9 0.2 16.2
61 Clint Capela  Hou 5 64.3 6.8 92.6 0.628 0.485 0 8.4 1.3 0.9 1.4 12
62 Robert Covington  Phi 4 64.5 7.2 89.1 0.402 0.834 2.6 6.3 1.8 1.9 0.8 13.9
63 Victor Oladipo  Ind 2 64.8 7.3 95.7 0.421 0.817 1.8 4.6 3.2 1.4 0.4 16.1
64 Jae Crowder  Cle 3 64.8 5.7 52.6 0.45 0.826 1.4 4.1 1.7 1 0.3 9.8
65 Dwight Howard  Cha 5 66.9 7 90.9 0.605 0.53 0 10.5 1.4 0.9 1.2 12.1
66 Isaiah Thomas*  Cle 1 67.1 19.1 83.4 0.455 0.891 2.2 2.7 5.2 1 0.2 20.7
67 Dwyane Wade  Cle 2 67.8 4.3 66.5 0.451 0.789 0.4 3.4 3.5 1.1 0.4 13.8
68 Dario Saric  Phi 3 67.9 6.6 56.1 0.428 0.784 1.3 6.3 2.6 0.8 0.4 12.8
69 Trevor Ariza  Hou 3 69.3 5.5 71.5 0.413 0.773 2.3 5.3 2.5 1.9 0.3 11.1
70 Jonas Valanciunas  Tor 5 70.8 6.6 74.4 0.558 0.801 0 9.7 1 0.5 1 12.1
71 Julius Randle  LAL 4 71.6 5.8 75.1 0.471 0.739 0.3 10.2 3.2 0.8 0.5 14.2
72 Tobias Harris  Det 4 73.7 5.3 90.3 0.466 0.82 1.3 5.3 2 0.9 0.4 14.8
73 Danilo Gallinari*  LAC 3 74.1 4.3 50.3 0.42 0.882 2.2 5 2.2 0.8 0.3 17.1
74 Avery Bradley  Det 2 78.4 3 79.1 0.436 0.783 1.8 3.9 2.3 1.4 0.2 14.2
75 James Johnson  Mia 3 78.8 3.5 75.3 0.463 0.688 0.9 4.8 3 1.1 1 11.2
76 Dennis Smith Jr.  Dal 1 79.4 2.5 74.5 0.397 0.715 0.9 3 3.7 1.1 0.2 9.1
77 Marquese Chriss  Pho 4 81.4 2.9 32.5 0.455 0.662 1 5.2 1.5 1.2 1 11.4
78 Evan Fournier  Orl 2 82.1 2.9 90.2 0.443 0.822 2.1 3.1 2.9 1.1 0.1 16.2
79 Steven Adams  OKC 5 83.9 2.7 85.9 0.582 0.587 0 8.1 1.3 1 1.1 10.3
```

80 Wilson Chandler Den 3 84.2 2.8 44.2 0.446 0.738 1.6 5.8 1.8 0.8 0.4 12.7
81 Aaron Gordon Orl 4 84.6 2.6 91.7 0.464 0.723 1 6.9 2.4 1.1 0.6 14.3
82 George Hill Sac 1 87.6 2.6 37.1 0.442 0.795 1.7 3.7 3.9 1.1 0.2 13.7
83 Kentavious Caldwell-Pope LAL 2 90.2 1.8 62.7 0.399 0.826 2.2 3.5 2.3 1.4 0.2 15.1
84 Derrick Rose* Cle 1 90.3 1.7 32.6 0.455 0.848 0.5 2.8 3.4 0.6 0.2 12.5
85 Pau Gasol SA 4 90.6 2.2 83 0.487 0.765 0.4 7.8 2.5 0.4 1.2 11.1
86 Marcin Gortat Wsh 5 90.6 4.9 81.4 0.545 0.654 0 8.9 1.4 0.6 0.9 9.7
87 Jeremy Lin* Bkn 1 91.8 1.9 7.4 0.411 0.811 1.3 3.3 3.9 1 0.5 12.7
88 Rudy Gay SA 3 92.1 2.5 67.6 0.463 0.833 0.8 4.4 1.8 1 0.6 11.2
89 Nicolas Batum Cha 3 92.3 11.7 71.8 0.401 0.851 2 5.4 5.6 1.1 0.4 13.9
90 Markelle Fultz* Phi 1 92.4 4.4 25.6 0.421 0.669 0.9 3.4 3.8 1 0.4 10.3
91 Eric Gordon Hou 2 93.2 2.1 86.1 0.421 0.852 3 2.6 3 0.8 0.4 15.4
92 Tim Hardaway Jr. NY 2 93.3 2.2 81.2 0.418 0.811 2.2 3.4 2.4 0.8 0.2 15.2
93 Lou Williams LAC 1 94.4 2.8 76 0.415 0.861 1.7 2.3 2.5 1 0.2 14.9
94 Reggie Jackson Det 1 94.6 1.6 83.5 0.421 0.859 1.3 2.8 5.2 0.8 0.1 14.5
95 Brandon Ingram LAL 4 94.6 2 77.5 0.408 0.654 0.9 4.6 2.6 0.8 0.4 10.9
96 Markieff Morris Wsh 4 94.6 2.1 39.6 0.457 0.805 1 6.6 2.2 1.2 0.6 14.1
97 Patrick Beverley* LAC 1 95.5 1.5 33 0.415 0.742 1.6 4.4 3.5 1.3 0.4 8.9
98 Zach LaVine* Chi 1 95.9 3.6 53 0.439 0.819 2 3.7 3.8 1 0.2 18.1
99 Enes Kanter NY 5 98 1.4 83.2 0.531 0.81 0.1 6.9 0.9 0.5 0.4 12.6
100 Willy Hernangomez NY 5 99.7 1.9 14.3 0.519 0.744 0.1 8.9 2.2 0.9 0.7 10.7
101 Zach Randolph Sac 4 100.9 2 52.5 0.453 0.76 0.2 7.8 1.7 0.6 0.1 12.7
102 Gary Harris Den 2 101 1.9 61.6 0.486 0.813 1.8 3.2 2.6 1.4 0.2 14.4
103 Andre Iguodala GS 3 101.5 1.6 12 0.474 0.674 0.9 3.3 3 1 0.4 6.6
104 Greg Monroe Pho 4 103 1.7 37.7 0.526 0.751 0 7.1 2.2 1 0.5 12.2
105 Justin Holiday Chi 2 103.8 1.5 41.5 0.39 0.798 1.7 3.4 1.8 1.1 0.5 9.3
106 Tristan Thompson* Cle 4 104.1 1.6 17.5 0.589 0.587 0 9 1.1 0.6 0.8 7.9
107 Malcolm Brogdon Mil 2 104.9 1.8 64.6 0.452 0.852 1.3 3.2 4.4 1.3 0.2 11.6
108 De'Aaron Fox Sac 1 107.1 1.5 55.3 0.421 0.735 0.4 2.7 3.2 0.9 0.1 8.8
109 Jayson Tatum Bos 3 107.4 1.5 88.5 0.42 0.804 0.6 3.6 1.1 0.6 0.3 6.5
110 Rajon Rondo Nor 1 108.8 2.5 49.4 0.44 0.608 0.7 4.9 7.7 1.6 0.1 8.1
111 JJ Redick Phi 2 109.3 1.6 50 0.458 0.883 2.7 2.2 1.7 0.7 0.1 14.5
112 Kent Bazemore Atl 2 109.4 1.2 56.4 0.446 0.716 1.6 5.3 2.6 1.3 0.9 15.2
113 Jabari Parker* Mil 3 109.4 3.2 29.4 0.506 0.771 0.6 4.7 2 0.9 0.3 13.7
114 Milos Teodosic* LAC 2 109.5 1.8 4.6 0.398 0.87 2 2.5 5.6 0.7 0.1 11.7
115 Nikola Mirotic* Chi 4 109.6 1.2 10.8 0.39 0.812 2.3 5.7 1.5 0.9 0.7 12.7
116 Dirk Nowitzki Dal 4 110.1 1.3 43.2 0.425 0.878 1.3 5.7 1.5 0.6 0.5 13
117 Josh Jackson Pho 3 110.5 1.7 18.8 0.442 0.591 0.6 5.8 2.1 1.3 0.6 11.1
118 Dion Waiters Mia 2 111.2 1.2 54.2 0.4 0.678 1.5 3.1 3.2 1.1 0.3 12.8
119 Jamal Murray Den 2 111.5 1.3 41.5 0.43 0.85 1.6 3 2.7 0.9 0.3 11.8
120 Derrick Favors Utah 4 112.5 1.6 64 0.506 0.708 0.1 6.3 1.5 0.9 1 13.7
121 Rodney Hood Utah 2 113.7 1.5 38.9 0.407 0.799 2.2 3 2 0.7 0.2 13.7

```
122 Willie Cauley-Stein  Sac 5 113.8 1.2 46.5 0.53 0.691 0 5.4 1.3 0.9 0.7 8.9
123 Ryan Anderson  Hou 4 114.4 1.3 33.6 0.42 0.848 2.6 4.9 1.1 0.6 0.2 14.1
124 Marvin Williams  Cha 3 114.4 1.1 14.8 0.412 0.811 1.7 6 1.5 0.9 0.8 10.1
125 Buddy Hield  Sac 2 114.5 1.9 23.5 0.423 0.822 2.1 4.1 2.2 0.7 0.1 13.4
126 Wesley Matthews  Dal 2 114.6 1.1 38.4 0.383 0.805 2.5 3.7 2.4 1.2 0.2 13
127 Mason Plumlee  Den 4 114.6 1.1 8.5 0.558 0.594 0 5.1 2.2 0.7 0.7 7.3
128 Thaddeus Young  Ind 3 115.4 1.1 70.8 0.484 0.6 0.4 6.4 1.7 1.4 0.4 10.6
129 Nerlens Noel  Dal 5 115.4 1.6 20.7 0.538 0.639 0 6.6 1.5 1.6 1.2 9.5
130 Rondae Hollis-Jefferson  Bkn 3 115.7 1.1 62.7 0.451 0.743 0.3 6.7 2.4 1.3
0.6 10.2
131 Marcus Smart  Bos 1 115.7 1.1 43.3 0.375 0.788 1.3 4.3 3.9 1.6 0.4 10
132 TJ Warren  Pho 3 116.7 1.1 78.4 0.487 0.772 0.6 4.7 1.5 1.2 0.5 13.4
133 Darren Collison  Ind 1 116.8 1.1 78.7 0.454 0.853 1 2.2 3.7 0.9 0.1 11.3
134 Austin Rivers  LAC 2 116.8 0 19.6 0.431 0.698 1.7 2.8 3 0.9 0.2 13.2
135 JaMychal Green  Mem 4 117.7 1 12.8 0.476 0.797 0.8 7.6 1.4 0.8 0.5 9.8
136 Tyson Chandler  Pho 5 117.9 1 21.4 0.613 0.706 0 9.6 1 0.6 0.6 7
137 Robin Lopez  Chi 5 118 1.1 44.5 0.507 0.765 0 7.1 1.3 0.3 1.3 11.3
138 Justise Winslow  Mia 2 118.2 1.1 13.6 0.409 0.706 0.5 4.3 2 0.9 0.3 7.7
139 Dejounte Murray  SA 2 119.3 0 21.5 0.448 0.773 0.4 1.7 2.1 0.5 0.2 5.6
140 Will Barton  Den 2 119.5 1 62.7 0.439 0.8 1.1 3.3 2.3 0.7 0.3 9.4
141 Jordan Clarkson  LAL 1 119.6 1.2 56.5 0.427 0.815 1 2.4 1.9 0.8 0.1 10
142 Jonathon Simmons  Orl 2 121.3 0 27.4 0.437 0.744 0.5 2.2 1.9 0.7 0.3 6.8
143 Ersan Ilyasova  Atl 4 122.4 1.1 5.9 0.418 0.753 1.8 6.2 1.5 0.8 0.3 12.5
144 Frank Kaminsky  Cha 5 122.7 1.1 17.7 0.409 0.763 1.4 4.1 2 0.7 0.5 10.8
145 Tyler Johnson  Mia 2 123.4 1.1 13.3 0.42 0.754 1 3 2.2 0.9 0.4 9.5
146 Marcus Morris  Bos 4 124.8 1.2 30.9 0.421 0.75 1.5 4.9 2.1 0.8 0.3 11.7
147 Tyreke Evans  Mem 3 140 1.2 84 0.435 0.752 1 4 4.2 1 0.3 11.9
148 Lauri Markkanen  Chi 4 140 0 68.5 0.408 0.789 0.8 4.3 0.6 0.3 0.3 7.1
149 Jeremy Lamb  Cha 2 140 0 60.9 0.429 0.829 0.9 3.3 1.3 0.6 0.3 8.9
150 Donovan Mitchell  Utah 2 140 0 47.8 0.379 0.778 1 2.7 1.6 0.9 0.2 7.8
151 DeMarre Carroll  Bkn 3 140 0 47.2 0.402 0.708 1.4 4.2 1.3 1.2 0.3 8.8
152 Taj Gibson  Min 4 140 0 45 0.531 0.727 0.1 6 1.4 0.7 0.8 8.9
153 John Collins  Atl 4 140 0 44.8 0.5 0.722 0 5.8 0.6 0.4 0.7 9.7
154 Bojan Bogdanovic  Ind 2 140 0 39.2 0.418 0.839 1.3 2.5 1.1 0.4 0.1 8.8
155 Bobby Portis  Chi 4 140 0 35.5 0.439 0.71 0.6 5.3 0.9 0.4 0.2 8.1
156 Joe Ingles  Utah 4 140 0 35.4 0.422 0.766 1.7 3 2.6 1.2 0.1 7.9
157 Kris Dunn  Chi 1 140 0 29.8 0.386 0.658 0.6 4 4.3 1.5 0.6 7.9
158 Alex Len  Pho 5 140 1.1 25.6 0.481 0.713 0 7.3 1 0.6 1.1 8.1
159 Kyle Anderson  SA 4 140 0 22.3 0.457 0.803 0.3 2.7 1.6 0.7 0.3 3.8
160 Courtney Lee  NY 2 140 0 21.4 0.429 0.829 1.2 3.3 2.4 1.2 0.3 9.7
161 Kelly Oubre Jr.  Wsh 3 140 0 19.8 0.432 0.741 0.8 3.6 0.9 0.8 0.2 7.2
162 Allen Crabbe  Bkn 2 140 0 17.4 0.449 0.854 1.8 3.2 1.6 0.9 0.3 11.8
163 Trevor Booker  Bkn 4 140 1 16.5 0.475 0.659 0.4 7.1 1.9 1.1 0.5 8.6
```

```
164 Danny Green  SA 2 140 1 16.3 0.401 0.82 1.8 3.6 1.9 1.1 0.9 7.6
165 Kelly Olynyk  Mia 5 140 0 16.1 0.46 0.754 1.2 4.3 1.9 0.7 0.4 9.3
166 John Henson  Mil 4 140 0 14.4 0.548 0.658 0 2.6 0.6 0.3 0.8 3.8
167 Al-Farouq Aminu*  Por 3 140 1 14 0.405 0.748 1.4 6.2 1.8 1.1 0.7 9.3
168 Evan Turner  Por 2 140 1 12.3 0.445 0.807 0.4 4.5 4.1 1 0.3 9.8
169 Larry Nance Jr.*  LAL 4 140 0 10.2 0.52 0.722 0.1 5.3 1.4 1.1 0.5 6.3
170 Kyle Korver  Cle 2 140 0 10.1 0.45 0.857 1.4 2 1.3 0.4 0.2 5.8
171 Caris LeVert  Bkn 4 140 0 10 0.428 0.745 1.3 3.8 2.5 1 0.2 10.4
172 JR Smith  Cle 2 140 0 9.4 0.4 0.688 1.5 2 1.4 0.8 0.2 6.6
173 T.J. McConnell  Phi 1 140 0 9.3 0.466 0.778 0.3 2.6 4.9 1.3 0.1 5.9
174 Emmanuel Mudiay  Den 1 140 0 8.8 0.398 0.753 0.9 2.4 3.4 0.7 0.3 9.2
175 Marco Belinelli  Atl 2 140 0 8.7 0.396 0.836 1.4 2.4 2.1 0.6 0.1 9.8
176 Nick Young  GS 2 140 0 8.2 0.403 0.851 1.3 1.4 0.7 0.4 0.1 6.6
177 Wesley Johnson  LAC 3 140 0 8 0.379 0.722 0.4 1.3 0.4 0.3 0.2 2.3
178 Seth Curry*  Dal 2 140 1.2 8 0.452 0.835 1.8 2.8 2.8 1.1 0.1 11.9
179 Michael Kidd-Gilchrist  Cha 3 140 0 7.2 0.495 0.764 0.1 6.7 1.8 1 0.8 10.8
180 CJ Miles  Tor 3 140 0 7 0.399 0.829 2.6 3.5 1.1 0.9 0.4 12.1
181 Jamal Crawford  Min 2 140 0 7 0.405 0.876 0.9 1.2 1.8 0.5 0.1 7.9
182 E'Twaun Moore  Nor 2 140 0 6.5 0.468 0.742 1.1 2.2 2.4 0.8 0.4 8.5
183 Cody Zeller  Cha 4 140 1 6.4 0.535 0.721 0 6 1.7 0.9 0.8 9.4
184 PJ Tucker  Hou 2 140 0 6 0.42 0.734 0.8 5.4 1.5 1.3 0.2 6.3
185 Patty Mills  SA 1 140 0 5.8 0.427 0.813 1.7 2 3.1 0.9 0.1 8.6
186 Yogi Ferrell  Dal 2 140 0 5.5 0.403 0.816 0.9 1.6 2.1 0.6 0.1 6.1
187 Ed Davis  Por 4 140 0 5.4 0.563 0.57 0 3.8 0.7 0.4 0.4 3.5
```