

How NBA Player Salaries Relate to Individual and Team Performance: Using Visualizations to Maximize a Team's Success

Alex Hoffer, Austin Nguyen, Prathveer Rai

Abstract— This paper attempts to describe how an NBA player's salary can be beneficial or detrimental to their team based on their performance in several statistical categories. This serves as a useful starting point for front office professionals employed by NBA teams who wish to make reasoned judgments about whether they should extend a player's contract or let them walk into free agency. After all, statistics in NBA basketball have become nuanced and sophisticated in the past decade, and teams such as the San Antonio Spurs have embraced their use and excel as a result. We believe that visualizations are the best method for truly understanding the effectiveness of an NBA player in the context of their salary since when these statistics are merely presented in their numerical form, people who are not mathematical experts (such as NBA General Managers, who are responsible for offering contracts to players) may not grasp the conclusion, or even worse, they may get the wrong impression about a certain player. Specifically, we believe that the radar chart is the most appropriate type of visualization to achieve this effect, since there are multiple statistical categories being judged by their relation to salary, which means it is essential that the user is able to see how far out edges extend within each category so they can conveniently visually compare how far out the edge pertaining to the salary is in relation to these statistics, giving them an indication of whether the player is worth their price.

1 INTRODUCTION

Basketball players employed by the National Basketball Association (NBA) make a tremendous amount of money. Their agents negotiate with the general managers of professional teams in order to get a salary that they think their performance warrants. Unlike professional sports leagues like Major League Baseball (MLB), though, NBA teams have a *salary cap*, which is a limit to the amount of money they can spend to field a 15-man team of \$102 million. This means that it is within each NBA team's interests to pay each player in, the average case, the exact salary which fairly corresponds to their on-court performance and, in the best case scenario for the team, a salary that is as low as possible for a player that is extremely good. A good example of the latter case being beneficial to a team is Stephen Curry of the Golden State Warriors, who gets paid x /year, and yet is a two-time MVP who is commonly considered one of the best players of all time. This low salary allowed the Warriors to spend money on acquiring Kevin Durant, another MVP, for a max salary.

Thus, this paper describes the process of our visualizations of the radar chart variety of specific players we believe are beneficial to observe in this way, since it has the capacity to captivate general managers of NBA teams into making the most reasoned decisions about who to sign and who to let walk in free agency. The paper will consist of several sections to achieve this effect, including the related visualizations we have found, the method we used to generate our visual computations, our programming implementation of these concepts, our results and their impact, and what we can conclude from our project and what future visualizations in this realm should address.

2 RELATED WORK

The use of spider charts to visualize NBA statistics is somewhat prevalent in the realm of entertainment, but not in information visualization research.

2.1 In Information Visualization Research

Historically, academic publications in information visualization rarely tackle the topic of sports, much less spider charts devoted to NBA player statistics and their salary. However, over the past several years, this trend has begun to reverse, with conferences such as the IEEE VIS sponsoring workshops on sports data visualization, though the majority of these visualizations regard sports such as football, soccer, and

baseball. Nevertheless, it is important for our purposes to survey the few visualizations within this subcategory of information visualization pertaining to the NBA that are present in academic research. [?], a paper that introduces an interactive shot chart tool that allows the user to choose a player and see on a virtual basketball court the physical areas on the court where they perform best, presents the idea of using visualization to enhance our assessments of NBA players' in-game performances, but does not describe how tools like this could be used by NBA executives to make reasoned decisions about players' salaries. [?] proposes a new way of presenting post-game information, gravitating away from webcasts and towards "season level, game level, and session level" visualizations, but the purpose of this work is to augment the NBA fan's experience and, while this work does include some visualizations of performances that are relevant to our research, the purpose of such visualizations is for entertainment consumption and not for the maximization of team success. [?] includes some visualizations that are similar to the ones we present in this work, including spider charts with steals, points, rebounds, assists, blocks, and turnovers as graph points of players such as Steve Nash, Deron Williams, and Rajon Rondo. However, these visualizations do not include player salary, which means they do not explore the financial implications of player performance that come in handy when making player personnel decisions. [?] uses line graphs and bar graphs to visualize and compare team performances, but provides no application of the importance of such visualizations to executive decision making nor does it provide visualizations of individual player performance. A compelling sequence of different sports visualizations can be found courtesy of the Information Interfaces team at Georgia Tech University. One interactive visualization of NBA player and team data by this team, a snapshot of which is included in *Figure 1* of the appendix, was particularly influential to the software development necessary for our project since it was the only open source visualization of NBA data.

2.2 In Entertainment

3 METHOD/COMPUTATIONAL MODEL

3.1 Method Overview

Figure 2 in the appendix is a flow chart diagram of the components necessary to program an interface with three graphs. The first graph is a radar chart that corresponds to a player's assists per game (APG), steals per game (SPG), blocks per game (BPG), rebounds per game (RPG), and turnovers per game (TPG). The second graph is a bar chart that corresponds to a player's average points per game (PPG) versus the average player's average points per game. This bar chart is presented vertically. The third graph is the same as the second bar chart, except rather than presented vertically, it is presented horizontally. For each

node in the flow chart diagram, there is a subsection within this section that corresponds to an explanation of how this step in the software engineering process was approached.

3.2 PHP Environment

This node is the start block that represents how each subsequent step can be completed. In order to satisfy the following requirements, we must operate in a PHP environment.

3.3 CSV File of Statistics

An input block, this node represents the NBA statistics we require to generate meaningful charts. In our file "Document2.csv", we have entries consisting of players like James Harden, Danilo Gallinari, and Jordan Clarkson with statistics ranging from their PPG to their RPG.

3.4 Plug Statistics of Players into Database

In order to do anything meaningful with the data we gathered in our CSV, we must place the data into a database for later use in the pChart environment. To do this, we have a file "insert_nba.php" that uses MySQL to connect to a database, parse our .csv file, and place it into a table called player_info. Some code pertaining to these steps: To connect to the database:

```
$user = 'root';
$pass = '';
$db = 'nba';
mysql_connect('localhost', $user, $pass);
mysql_select_db($db);
```

To parse our .csv file:

```
$parsed_data = array_map('str_getcsv',
file('Document2.csv'));
```

Finally, to place the data into a table:

```
$ins_sql = "INSERT INTO player_info (name, age,
team_name, wins, loss, ppg, bpg, rpg, apg, tpg, spg,
games) VALUES ('$name', '$age', '$team', '$wins',
'$loss', '$ppg', '$bpg', '$rpg', '$apg', '$tpg',
'$spg', '$games')";
```

3.5 Grab Player Statistics From Database

Naturally, it is necessary to access the database we have created in order to use our data for chart generation. This step happens at a similar time as the next step, so there is not necessarily a temporal order. As a code example of this step, let's look at a method in our file "vis_proj.php" file that averages the points per game of a player:

```
function get_average_ppg(){
    $avg_sql = "SELECT ppg
FROM player_info";

    $result = mysql_query($avg_sql);
    $result = mysql_fetch_assoc($result);
    $sum = 0;
    foreach($result as $key => $value){
        $sum += $value;
    }
    $avg = $sum / count($result);
    return $avg;
}
```

3.6 Access pChart API

In order to generate the charts that we want, we utilized pChart. Accessing this API consisted of the following include statements in our "vis_proj" file:

```
include("pChart2.1.4/pChart2.1.4/class/pData
.class.php");
include("pChart2.1.4/pChart2.1.4/class/pDraw
.class.php");
include("pChart2.1.4/pChart2.1.4/class/pRadar
.class.php");
include("pChart2.1.4/pChart2.1.4/class/pImage
.class.php");
```

3.7 Draw radar chart, bar chart 1, bar chart 2

To draw our charts, we utilized pChart's functionalities. Let us explain the process of drawing the radar chart. First, we initialized a pRadar object. Then, we passed this object into a function along with a player's data. In this function, we used the pChart method "setGraphArea" to specify the size of the graph, created an Options object to hold specifications for what we wanted our graph to look like, and then finally passed our pRadar object into the "drawRadar" pChart function.

3.8 Produce pCharts for viewing

This node is the completion of the process of generating charts, a result of compiling the programs outlined in the previous steps. Three charts corresponding to a player are presented to the user.

3.9 Parameters

Since our visualization module uses static data that we gathered via a basketball statistics website, our parameters to pChart functions are unchanging.

4 IMPLEMENTATION

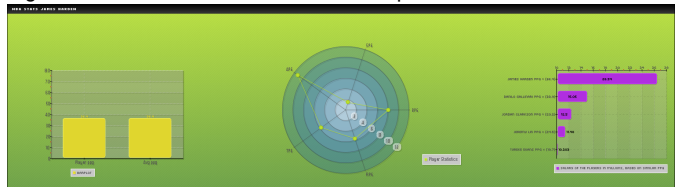
In implementing our visualization software, we were surprised to see how many different options for packages there were. Initially, we planned on using Python embedded in HTML in order to use the pyfigure package, but we eventually landed on using pCharts due to its ability to be implemented in a PHP context. We also wanted to include salary in our radar chart but found it virtually impossible to find a reasonable way to describe statistics and salary using the same units due to their large difference in size.

5 RESULTS AND PERFORMANCE

5.1 Results

5.2 Images

Fig. 1. A visualization of James Harden's performance in the NBA.



We can see in Figure 1 an example of what our module produces. In the left chart, we can see that James Harden averages many more points than the average player. In our radar chart, we easily get, through our visual sense, an idea of what kind of player James Harden is: he averages many assists, he doesn't get many steals or rebounds, and he turns the ball over seldolmy. Finally, on the right, we see a bar chart of a sample comparison we can make between

5.3 Performance Analysis

Since our module does not make use of I/O dependent on the user, and because our data set is reasonably small, a performance analysis of the generation of charts is not necessary.

6 CONCLUSIONS AND FUTURE WORK

Fig. 2. A snapshot of an interactive visualization of basketball data.

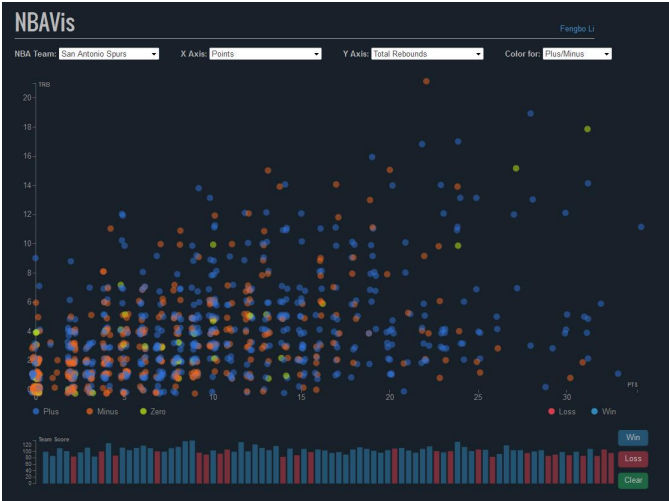


Fig. 3. A flow chart diagram of our programming process.

