NBA Shot Data Visualization

JOEL CHRISTENSEN, Utah State University, USA
SAM JOHNSON, Utah State University, USA
HAGEN LARSEN, Utah State University, USA
MARSHAL TAYLOR, Utah State University, USA

Additional Key Words and Phrases: NBA, basketball, visualization, data, interactivity

ACM Reference Format:
Joel Christensen, Sam Johnson, Hagen Larsen, and Marshal Taylor. 2022. NBA Shot Data Visualization. 1, 1 (October 2022), 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 MOTIVATION, OBJECTIVE, AND TARGET AUDIENCE

Basketball is a context filled with data. In modern professional basketball, computer vision allows player tracking by the second and produces a wealth of data that is consumed by players, coaches, pundits, and fans daily. [6]. The visualization of such tracking data has become a multi-billion dollar industry across sports, but dynamic sports such as basketball have special need for continued development of intelligent visualizations [1]. In this paper, we propose a tool that will interpret such rich data and display it more intuitively and powerfully to the potential user.

Our visualization will be designed for use by individuals already familiar with NBA basketball. However, users may vary widely in motivation and purpose. Below, we cover different groups that will find utility in using the tool.

1.1 Casual Users

Casual NBA fans could use this tool to learn how effective their favorite players are. With the increasing popularity of sports gambling, they may be searching for good betting or fantasy basketball opportunities. The tool could help them answer questions like:

- How is *player x* performing this year? How is he performing relative to years past? Relative to other players in the NBA?
- Where is *player x* taking his shots from? Where is he most effective?
- What trends does *player x* exhibit throughout the course of the game? How does his shot selection change throughout the game?
- How likely is it for *player x* to score 20 points a game?

Authors' addresses: Joel Christensen, christensenjo14@gmail.com, Utah State University, Logan, Utah, USA, 84321; Sam Johnson, sam.johnson1357@gmail.com, Utah State University, Logan, Utah, USA, 84321; Hagen Larsen, A02254637@usu.edu, Utah State University, Logan, Utah, USA, 84321; Marshal Taylor, mptaylor777@gmail.com, Utah State University, Logan, Utah, USA, 84321.

1.2 Coaching and Play Design

Coaching staffs will also find our interface useful. Over 75 percent of NBA coaches already use live tracking data to guide players through practices and games [5]. Coaches use player tendencies, field goal percentages, and other statistics to make decisions on who will play, what sets their team will run, and how they will plan to defend other teams and specific players.

This type of game-planning is obviously useful to NBA coaching staffs at the highest level, but our interface can also be useful to lower-level basketball coaches. By analyzing NBA players, lower-level coaching staffs can learn from the best and better emulate the way that NBA coaches make decisions. Coaching questions that are answered in the visualization include:

- Where on the floor does *opposing player x* have the highest field goal percentages?
- Should we try to force *opposing player x* to drive right or left, based on his tendencies and field goal percentages?
- Which of our team's players is most clutch? In other words, who is has the highest shooting percentages in late-clock and late-game situations?
- What play calls will be most effective at positioning our players in an opportunity to score from floor locations where they have high shooting percentages?

1.3 Professional Team Management

Front office staffs throughout the NBA and G-League will also find value in the visualization interface. The General Manager and other executives must manage their teams within specific parameters established by the NBA. The NBA promotes parity through a salary cap, limiting the amount of money that can be distributed to players each year. General managers must also pay careful attention to the style and tendencies of their players, making sure the team plays complimentary basketball. It's the GM's job to put together a winning team through signings and trades. The visualization will help front offices of all levels answer questions such as:

- How does the on-court production of *player y* compare to other players who earn similar salaries or who are currently available?
- Where is *player y* most effective on the basketball court? Where are his favorite spots to shoot from?
- What are *player y*'s skills? Are they complimentary to the other players' skills on the team?
- Do *player y*'s trends and skill sets fill a hole in our roster as it is currently constructed?

1.4 Summary

We have described the motivation for the project based on use cases for casual fans, coaching staffs, and professional team management. Similar tools, such as TenniVis, have proven successful in analyzing individual performance in individual sports [4]. Our tool will provide the same level of analysis for individuals in *team* sports. The entirety of ways in which the tool might be used are too vast to list. Our interactive data visualization will attempt to satisfy users of all interests and motivations in a meaningful way.

2 DATA PROCESSING

2.1 Data Source

The data we will visualize is sourced from the nbastatR package for R. NbastatR pulls its data from a variety of statistics sources including the NBA Stats API, Basketball Insiders, Basketball-Reference, HoopsHype, and RealGM. These sources store a breadth and depth of statistics using both traditional stat recording techniques as well as computer vision methods such as those developed by Second Spectrum. By leveraging nbastatR's powerful API we will be able to focus on creating value through visualization and keep data processing to a minimum. In order to dynamically connect this R package to our visualization, we plan to develop a Django back-end and use the R2Py package to make R function calls from within python in the back-end. These calls will return data that can then be processed via Python and passed to front-end JavaScript for D3.js visualization.

2.2 Schema

The data we will focus on expressing will come from nbastatR's team_shots() function, which returns a data frame with items representing shot actions taken during NBA games. Each call to team_shots() specifies a date range and season type (regular season or playoffs). Each item has attributes including the name of the player who took the shot, their team, whether the shot was made or missed, what type of shot action it was, whether the attempt was a two point or three point attempt, the date the game was played, the current period of the game, the time remaining in the game, and the x and y coordinates on the court where the shot took place. By having all of this data available to us, we will be able to create several different visualizations within our interface that will provide different types of analysis value to the user.

2.3 Processing

As mentioned, data processing will use a combination of R function calls, some Python data manipulation in a Django back-end, and final data formatting in the JavaScript front-end using D3.js. Because we plan to use data from across multiple NBA seasons and years, multiple R function calls may be necessary. The use of a Django back-end that performs these function calls will, however, provide a robustness to our interface and allow it to continue to be useful as time goes on. Some amount of validation and data cleaning may be necessary, but we anticipate that this will be minimal due to the quality of data returned by nbastatR. Shot items with significant missing data will be discarded. Unnecessary attribute columns will be discarded, and items will be filtered based on the user request to shots during games by a specific player in a specific timespan.

3 VISUALIZATION DESIGN

3.1 Player and date range selection

The visualization interface will be designed to provide data on current players as well as data on historical players. The interface will appear empty after being opened. The user will have a search bar where they will search for an NBA player's name. As they search, a drop-down (think Google search) will populate, showing all player names that match what has been typed so far. When multiple players have the same name, we presume that the user likely wants to see more recent players. We sort the drop-down list of players by recentness.

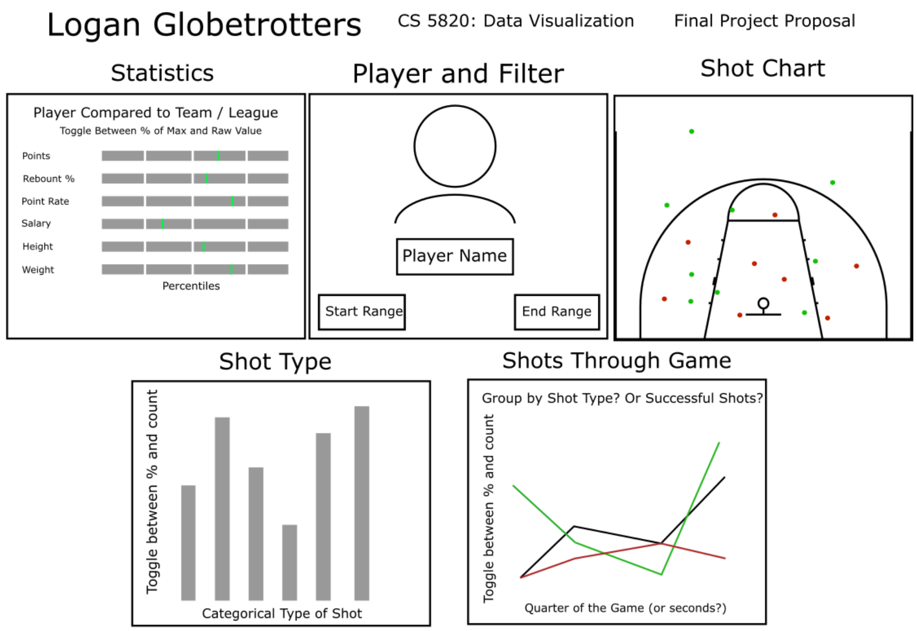
Logan Globetrotters
CS 5820: Data Visualization
Final Project Proposal
Statistics
Player Compared to Team / League
Toggle Between % of Max and Raw Value
Points
Rebount %
Point Rate
Salary
Height
Weight
Percentiles
Player and Filter
Player Name
Start Range
End Range
Shot Chart
Shot Type
Toggle between % and count
Categorical Type of Shot
Shots Through Game
Group by Shot Type? Or Successful Shots?
Toggle between % and count
Quarter of the Game (or seconds?)

Fig. 1. Prototype generated in Inkscape

After the user has selected a player, their attention will turn to two date selection boxes for the beginning and ending date. The beginning date should automatically be preset to the first date that player played in the NBA. The ending date should automatically be preset to the last date that player ever played in the NBA. The user can change these dates if they want to focus on a smaller range of dates but they cannot increase the distance between these dates. If the user selects a period where the player didn't play and thus has no stats, an error message should be raised in the browser and the user should be notified.

Assuming the user has selected a valid date range where data exists, the interface will automatically update and show various plots. Error messages and page highlights will help guide users should an error occur.

3.2 Interface Visualizations

Shot Chart: Display a shot chart for the selected player over the given date range. Made shots will be displayed in green while missed shots will be displayed in red, or a heat map will be used with various colors and opacity. All data will be displayed within a rendering of a basketball court to contextualize the data within the field of play [3].

Statistics Bar Chart: Display the player's shooting percentage, three point percentage, number of rebounds, and any other relevant stats against the league average over the same period. Other possible stats might include the player's salary, height, and weight.

Shot Type Frequency Bar Chart: The distribution of shot types taken by a given player. What percentage of the player's shots are three pointers, what percentage are layups, etc.

Shot Frequency over Time Line Plot: The number of shots the player takes per quarter averaged over several games.

In soccer, the spatial position of players on the field has been visualized and analyzed to determine the best formations [8]. We propose an adaptation for the sport of basketball through the creation of a heat-map plotting the position and accuracy of player's shot attempts while on the floor. This tool can provide insight to players that may otherwise go unobserved.

3.3 Summary

Our design for the visualization will be a multi-paneled web page. Each of the panels will be a separate visualization describing common filtering in a unique and novel way. A filtering panel will be included in the visualization, giving users an easy, convenient, and intuitive way to filter the visualization as a whole. Other sport visualizations have used the central filtering to great success [4]. TenniVis included a large filtering panel on the left hand side of the application. With the central filtering panel, we hope to increase the cohesion of the visualization and to remind users that each of the panels are drawing from the same common filtering of the data.

The panel design was chosen to reduce visual clutter and to help users answer their questions with a specific visualization.

3.4 Features

Many of the features of our visualization have been mentioned previously within the proposal. Listed below are the highlighted features we plan to build.

- Central filtering panel with player and date filters
- Auto completing search for player names
- Four visualizations describing the players performance within a given time frame
- Dynamic and asynchronous rendering of the page based on user inputs

3.5 Justification

A related visualization known as BKViz also seeks to display basketball information in a graphical format [2]. Their visualization is great for displaying team to team performance, and high level statistics via a global filter. However, their visualizations seem to be disjoint from one another, and the global filter appears to take up an entire page. We hope to further their work by including all of our visualizations on a single page, with a unified filter. We also plan on making our visualization player focused, with team information acting as a context. This will increase the granularity of the data analysis by allowing users to focus on a single player at a time. All of our visualizations will appear on the same page, allowing for easy comprehension and comparison between each of the visualizations.

3.6 Visualization System

The visualization will be implemented through a web application consisting of a back-end and front-end. Similar to iTTVis [7], data processing will take place in the back-end and interactive visualization will occupy the front-end. Different from iTTVis, however, the data will be retrieved once using the nbastatR package and may eventually be stored in a SQLite database for future access by application resources. Django will be used as the back-end framework. The back-end service will respond to front-end requests with data matching the user-defined criteria. Using D3.js and the native JavaScript fetch API, the front-end will make requests for data and visualize the data it receives.

4 SCHEDULE

4.1 October 28 - November 4

Create visualization system and configure back-end for data retrieval.

4.2 November 4 - November 11

Connect the back-end to front-end JavaScript through fetch API and obtain data from back-end.

4.3 November 11 - November 18

Begin data filtering functions to find data that matches a given player within a given date range according to user input. Begin implementing individual visualizations.

4.4 November 18 - November 25

Complete user input and input validation with data filtering. Continue implementing individual visualizations with D3.js.

4.5 November 25 - December 2

Complete all visualizations that will display data to the user.

4.6 Before December 6

Consider edge cases, user input errors to provide a graceful and seamless experience for the user. Test the program thoroughly.

REFERENCES

[1] Raffay Hamid, Ram Krishan Kumar, Matthias Grundmann, Kihwan Kim, Irfan Essa, and Jessica Hodgins. Player localization using multiple static cameras for sports visualization. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–738, 2010.

[2] Antonio G. Losada, Roberto Therón, and Alejandro Benito Santos. Bkviz: A basketball visual analysis tool. *IEEE Computer Graphics and Applications*, 36:58–68, 11 2016.

[3] Charles Perin, Romain Vuillemot, Charles D Stolper, John T Stasko, Jo Wood, and Sheelagh Carpendale. State of the art of sports data visualization. In *Computer Graphics Forum*, volume 37, pages 663–686. Wiley Online Library, 2018.

[4] Tom Polk, Jing Yang, Yueqi Hu, and Ye Zhao. Tennivis: Visualization for tennis match analysis. *IEEE transactions on visualization and computer graphics*, 20(12):2339–2348, 2014.

[5] Yusuke Sano and Yohei Nakada. Visualization for potential pass courses and quantification for offensive and defensive players in basketball. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–6, 2021.

[6] Egor F. Smirnov, Nikita S. Ivanov, Maksim Zavgorodnii, Ivan Yu. Khramov, and Victoria V. Chernyshova. Recognition of the basketball players position using live cameras. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 695–698, 2021.

[7] Yingcai Wu, Ji Lan, Xinhuan Shu, Chenyang Ji, Kejian Zhao, Jiachen Wang, and Hui Zhang. ittvis: Interactive visualization of table tennis data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):709–718, 2018.

[8] Yingcai Wu, Xiao Xie, Jiachen Wang, Dazhen Deng, Hongye Liang, Hui Zhang, Shoubin Cheng, and Wei Chen. Forvizor: Visualizing spatio-temporal team formations in soccer. *IEEE transactions on visualization and computer graphics*, 25(1):65–75, 2018.