CS 171 Process Book

Christine Hwang, Joseph Song, Travis Yeh

Overview and Motivation

Basketball data is well-documented and available from many sources around the web. However, it is almost always presented in the standard box score format -- a massive chart, with a multitude of columns for each player or year. As a result, it's difficult to read and process the data that's given, and even harder to compare data across players, teams, and years. The box score is certainly an efficient and organized way to present data, but it fails to allow for much actual understanding besides what the pure numbers are. Motivated by our team members' shared interest in basketball, we hope to take both the traditional box score detailing individual player statistics, as well as the aggregated data concerning the performance of the league's teams over time and turn it into a visualization that facilitates data consumption for cases and purposes as diverse possible.

Related Work

Some related works that inspired our visualization were sample projects that were introduced in class. Particularly, we were motivated by Data Driven Dota, and their user experience in filtering by player. We wanted to take a similar approach in organizing our visualization so that the user can visualize data that is filtered by teams and by player in the header of our webpage for easy access and clean visualization. We also looked into publication on visualization over time and was specifically inspired by Visualizing Change by Stephen Few, as it delved into methods of how to analyze data in a time series analysis. Given our player data ranges from year to year, this paper helped give us insight on what tools would be useful to provide an easy glimpse to a player's skill.

Questions:

Initial Questions

1. How do teams compare to each other?
2. How do players compare to the overall team average?
3. How does a player compare to himself over time?

We began with the question of how players compare with other players. However, we realized the visual complexity of this question and believed this question could be broken down into further parts. First, we can compare players with other players in its own team by comparing the player's stats to the team's averages. This asks the question of how well the player is doing relatively to everyone else.

We eventually evolved this question to look more macro and micro scale into how teams can compare to other teams and how a player compares to himself. This provides a detailed and personal story of how a player changes over time, as well as how well a team is doing on a larger scale compared to other teams.

Data:

We first looked to the CBS Sports API (http://api.cbssports.com/fantasy/players/list?version=3.0&SPORT=basketball&response_format=JSON) in order to obtain the player names and ids in order to create iteratively the urls that we want to visit for each player which had a base url of http://www.cbssports.com/nba/players/playerpage/{0}/{1}-{2} where {0} was populated by the player id, {1} by their first name, and {2} by their last name. We visited each of the player's URL's and scraped off their basic statistics such as body weight, height, experience, team, etc. and also all of their performance statistics for every season.

The major difficulty with obtaining this data was the wrangling and clean up portion. We decided because we are used to using it as a JSON in d3 and also a JSON provided a unique ability to format our data where each player can have an array of seasons which a regular csv file didn't have, we began to clean up our data into a JSON format. We made functions that created objects of each of the players with their name, experience, team, height, weight, photo, age, and seasons, where seasons would be an array of dictionaries that held their performance statistics given a key value of which statistic it was. We were able to create dictionaries by defining a make_season function that took in an objects properties and created a season out of it, and thereby calling it in our make_player function in order to ultimately create a list of player objects. Using this list of player objects, we were able to export it as a JSON file, clean up a few of the NaN values, and obtain our dataset.

The code can obtained by looking at nba_scraping.html which provides a static screenshot of all of the code used and any output.

Exploratory Data Analysis:

At first, data acquisition was a quite long and challenging process for us. Because we had a concrete goal of how we wanted to format our data, a lot of our time was spent organizing our data in this specific JSON form, and therefore there was little surprise in dealing with the data after we began this cleaning. We wanted to create an object representing a player and have a nested dictionary that represented the player’s stats for each season.

Initially, to look at our data, we used a table to look for inconsistencies in the data and get a better grasp of how each number was formatted. We were able to understand which variables were outside the nested objects and therefore static within each player and which variables were dynamic per season. These insights were especially helpful in our individual player stats page, where we visualized the player’s static attributes near the header as a static table and represented dynamic stats through a time series visualization to show change over time.

Knowing to format our design as a time series opened a variety of visualization techniques for us. We considered a side by side bar chart to compare a player’s current statistics to the previous year statistics, a negative value bar chart that shows the delta in the player statistics, and a show reel to animate how the stats changed over time in a line graph.

As we progressed in the project and completed our initial goals, we discovered more and more that there were further questions we wanted to answer. For example, after coding our initial horizontal bar chart that showed how a player’s current season performance compared to his overall career average, we wondered: what could we discover if, in addition to comparing a player to himself, we were able to compare players to each other? Additional questions followed. Sure, James Harden is a superstar of the Rockets, but how much does he actually contribute to the team’s overall statistics? What is the history of a team and is it currently on the rise, or is it declining? As each of these questions came up during our brainstorming and coding sessions, we believed that they would add significant value to our visualization and thus decided to implement them.

Design Evolution and Implementation

In this section, we’ll detail what our initial ideas were and how they changed over time to create our final product. There are two sections: the first is from our initial process book, before we had started much of the coding, and the second is an explanation of how we reached our conclusions.

Initial design:

The different visualizations we considered were an animated show reel to show how a player changes in skill over time, stacked barchart, table of full statistics, and a doublesided bar chart to show delta in skills.

We want to pursue a show reel because we believe that it is a visual technique that is friendly to the user and allows a quick and easy perception of how a player is performing over the years. By tracking the reel as it moves up and down a graph, we can see trends over time.

We also considered a stacked barchart. The way we would implement this is have a barchart representing the total points per game or total free throws a team made in a season. This would be further broken down as a stacked bar chart, with each band representing a player. This way, we can visualize in one chart how each player is doing respectively to another player on the team and how much a specific player contributed to the team as a whole.

We also want to include a table of statistics because as much as visualizations provide the user a way to understand trends on a macro scale, we still want our visualization to provide a way to deliver specific statistics of a player. Therefore, on top of graphs, show

reels, and bar charts to show trends and relationships, we still want to show specifics on how a player is doing.

Lastly, our barchart helps show a time series analysis of how a player does compared to himself over time. Using a bar chart helps give a visual interpretation of what proportion of statistics have improved this current year and which decreased.

Final design:

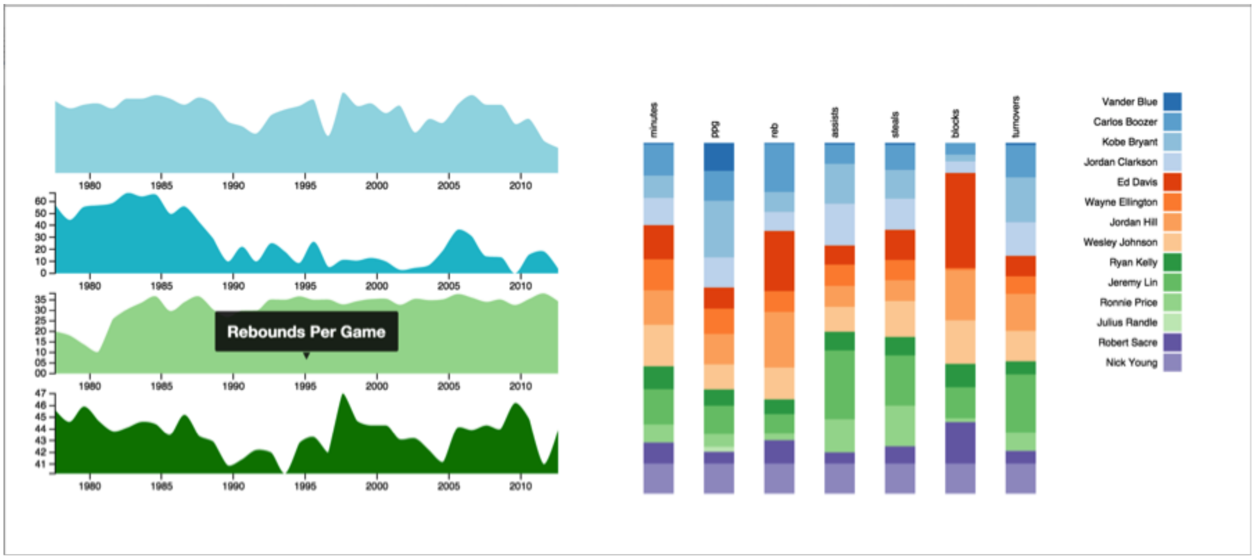
Our final design has several different components, each of which underwent its own iteration process as we built the visualization and decided what would make the most coherent, intuitive presentation. We detail each of the components below:

The Team Visualization

In our original implementation plan, we mentioned that we wanted to have the ability for teams to compare to each other, especially in the context of player histories. However, while brainstorming and defining what we wanted the purpose of our visualization to be, we decided to pivot and instead expand the capability to understand the data for a *given* team. We believe that this would enhance the userflow of the product: following the visualization principles to move from general to specific, we postulated that the user would click on a team in the nav bar, expect to see more information about that team, and then move on to the specifics of the players.

Thus, our team visualization portion has two views. The first, a series of four paths with the first one brushable, shows the team's statistics through history. With this first view, the user can look at trends in the team's performance over time, as well as brush over a specific area in order to get a more detailed look at the trend. This matches the "general" portion of the team information that we had in mind; by using this visualization, our user would be able to grasp at a glance what the team's performance has been like recently compared to its history.

The second view on the page is a stacked bar chart, which acts as the beginning of a paradigmatic shift from the team to the specific players. Here, we show various overall statistics for the team, broken down by how much of that overall statistic was contributed by each player. We have a legend on the side, as well as tooltips that appear on hover over the bars to show the player that contributed that portion and how much they contributed. The stacked bar chart also reacts to hovering over the player images in the navigation bar for clarity. Thus, this view lets the user see and understand the composition of a team's statistics, which is harder to grasp from the traditional box score alone.



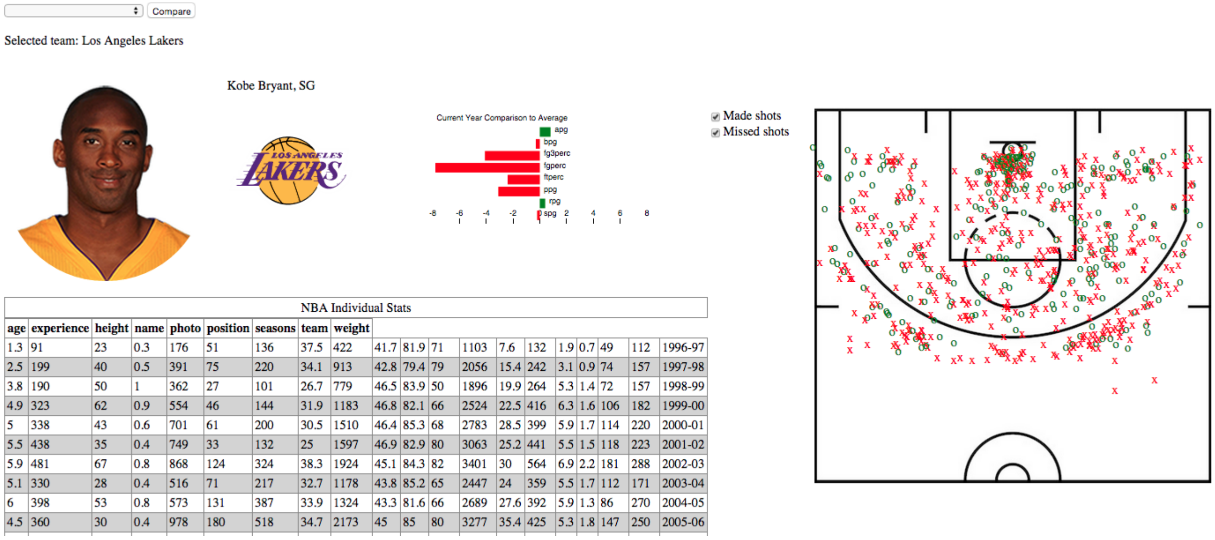
The Player Visualization

Upon selecting a player from a team in the nav bar, the page changes to show the individual player's profile page. There are three main views here. As above, we'll go through each one and explain the process used to create them.

The first one we implemented was the "Current Year Comparison to Average" bar chart. This bar chart shows how the player is currently performing in various stats compared to their career average. This was one of the first visualizations that we created during this project process, and quickly gives the user knowledge about how this player is currently performing compared to their "usual".

The second is a table. While this is not necessarily a visualization, we believed that keeping the stats in pure number form like this would be useful for reference or for "power users", as our TF put it. As much as visualizations provide the user a way to understand trends on a macro scale, we still want our visualization to provide a way to deliver specific statistics of a player. Therefore, on top of graphs, show reels, and bar charts to show trends and relationships, we still want to show specifics on how a player is doing.

The last view on the page is the shot chart, which took a significant amount of data scraping and wrangling to create. To create the shot chart, we first scraped data (as detailed in the data section) in order to have a JSON of every shot every player in the NBA took this seasons, as well as where they took it from and whether they made it or missed it. We then overlaid these positions onto a picture of a basketball court of appropriate dimensions so that one could see the distribution of a player's shots. The shot chart includes checkboxes so that one can see only shots that were made or shots that were missed, or both for clarity.



Selected team: Los Angeles Lakers

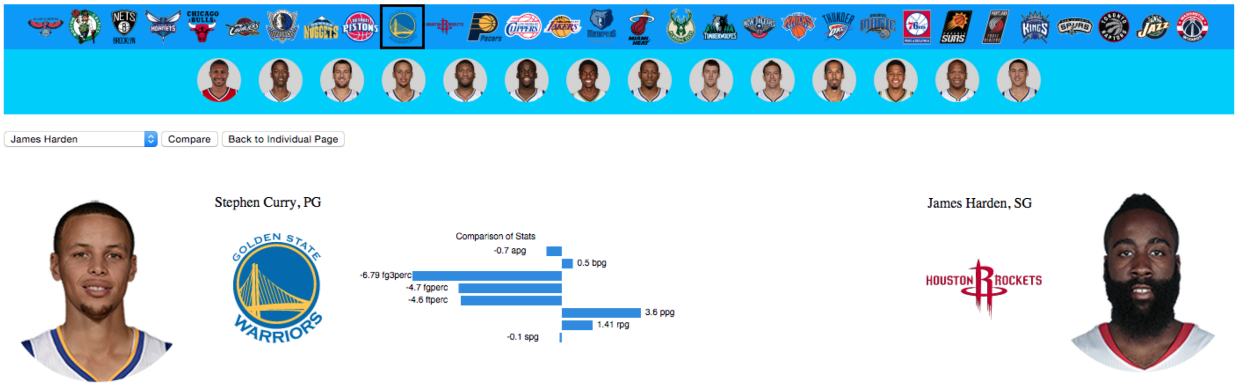
Kobe Bryant, SG

NBA Individual Stats

age	experience	height	name	photo	position	seasons	team	weight											
1.3	91	23	0.3	176	51	136	37.5	422	41.7	81.9	71	1103	7.6	132	1.9	0.7	49	112	1996-97
2.5	199	40	0.5	391	75	220	34.1	913	42.8	79.4	79	2056	15.4	242	3.1	0.9	74	157	1997-98
3.8	190	50	1	362	27	101	26.7	779	46.5	83.9	50	1896	19.9	264	5.3	1.4	72	157	1998-99
4.9	323	62	0.9	554	46	144	31.9	1183	46.8	82.1	66	2524	22.5	416	6.3	1.6	106	182	1999-00
5	338	43	0.6	701	61	200	30.5	1510	46.4	85.3	68	2783	28.5	399	5.9	1.7	114	220	2000-01
5.5	438	35	0.4	749	33	132	25	1597	46.9	82.9	80	3063	25.2	441	5.5	1.5	118	223	2001-02
5.9	481	67	0.8	868	124	324	38.3	1924	45.1	84.3	82	3401	30	564	6.9	2.2	181	288	2002-03
5.1	330	28	0.4	516	71	217	32.7	1178	43.8	85.2	65	2447	24	359	5.5	1.7	112	171	2003-04
6	398	53	0.8	573	131	387	33.9	1324	43.3	81.6	66	2689	27.6	392	5.9	1.3	86	270	2004-05
4.5	360	30	0.4	978	180	518	34.7	2173	45	85	80	3277	35.4	425	5.3	1.8	147	250	2005-06

The Comparison Visualization

This last visualization allows the user to compare the performance of one player to another. We added this feature after discussions with our TF about how to leverage the statistics that we have about each player. Also, in contrast to the context of our decision regarding whether or not to compare teams to each other, we believed that adding in the player comparison at the "end" of the userflow -- on the individual player level -- made sense conceptually as well. Upon selecting another player to compare with, the view changes to show the two players, their teams, and a comparison bar chart where the bars point towards the player with higher numbers in that statistic.



Evaluation

Our final product ultimately accomplished not only the goals that we set out to achieve, but the additional goals that we decided on along the way as well. By scraping and wrangling large amounts of data and processing that data into visualizations, we created a comprehensive visualization that shows current and historical data for every team and player currently in the NBA.

With the use of our visualization, we were able to discover a few patterns and trends that we found interesting. Some of our favorites, chosen because of our own personal basketball team allegiances:

- The shot chart provided some interesting insights: for example, despite practice, it seems that Jeremy Lin is still lacking in his ability to finish with his left, as he has a larger mass of missed shots on the left side of the rim.
- With the help of our visualization, we can now directly compare players to one another. With this, we can see that Steph Curry really does beat out James Harden in the majority of the chosen stats we have in our visualization.
- Using the team history path views on our team page, we can see the recent fall of the Lakers as a powerful and dominating team due (at least significantly in part) to an aging Kobe Bryant and a lack of a strong supporting cast. This insight is either the best or the worst out of the ones that we found, depending on which team member you ask.

If we had more time on this project, we would go about further reformatting our visualization and improving its aesthetics. This would include cleaning up some of the axes, as well as possibly adding tooltips instead of depending on alt text to show the players' names in the navigation bar (a choice made because of some poor initial planning that was discovered too late). However, the main change that we would add would be to revamp the aesthetics of the visualization portion of the website. We have recently been fascinated by Google's principles of Material Design and believe that the layouts and card interfaces that they use would be helpful in presenting our current information in a manner more organized than it already is.

In terms of visualization, another view that we could consider adding would be the capability to compare teams to each other. However, as detailed in earlier sections, this is contrary to our user flow if just inserted into our project as is, so we would probably have to work on figuring out what an alternate interface would be to accommodate this feature without causing cognitive dissonance.