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. With our project, we hope to take the data that's available on player and team performance and translate it into a visualization that is elegant and easy to use. The box score is certainly an efficient and organized way to present data, but it fails to allow for much actual understanding besides the pure numbers. We hope to take the traditional box score and turn it into a visualization that facilitates data consumption for cases and purposes as diverse as those held by the members of our team, and allow easy comparison of teams and players against the rest of the league and their past performances.

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:

1. How to teams compare to each other?
2. How to 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.

Design Evolution

The different visualization 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 double-sided 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.

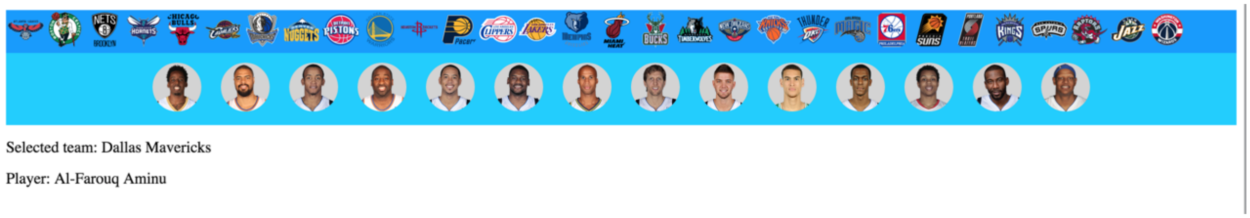
We also considered a stacked barchart. The way we would implement this is have a barchart representing the total points per game or total free throws a team made in a season. This would be further broken down as a stacked bar chart, with each band representing a player. This way, we can visualize in 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.

Implementation

We have yet to implement all of the interactive visualizations as to date. We want to incorporate much more interactivity, but for this milestone we focused a lot on the data collection in order to have the rest of the project be coding as we are used to. For the final project submission, we have a good plan of what we want to implement especially in order to obtain some of the higher-hanging fruit that we had identified as our optional features.

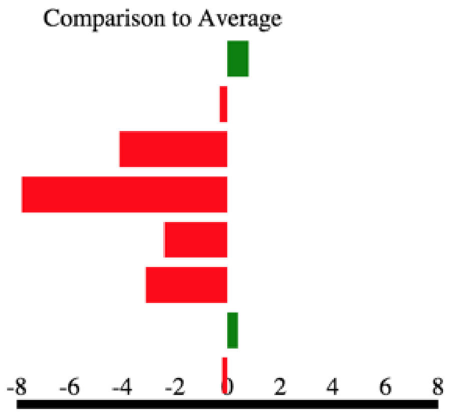


Selected team: Dallas Mavericks

Player: Al-Farouq Aminu

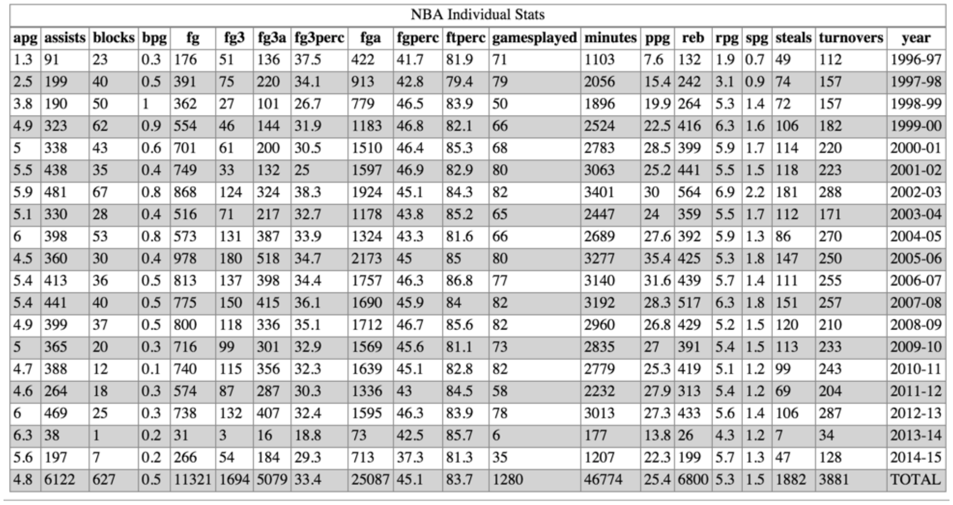
In our home page, we have implemented an interactive header that when you click on the team logo, it populated the next row with a list of image of the players in that team. We also implemented a hover feature that shows the name of the team and name of the player when mouseovered. Once a user clicks on a player’s head, it takes us to the page of individual player statistics. Because we have yet to implement the interactivity where the individual player is dynamic to the player selected, for this point in our project, all links to the head direct us to a player with Kobe. We will later change this to become more dynamic.

Once taken to the individual player statistics, we have a picture of the player, picture of a team, and a barchart that shows the change in player statistics over time.



This shows us the delta or how well this player has done in this season compared to its average. We will improve on design of this aspect by adding a tool tip that shows what each bar represents as a number.

Lastly, we have implemented a table with full statistics of this player in his career.



NBA Individual Stats

apg	assists	blocks	bpg	fg	fg3	fg3a	fg3perc	fga	fgperc	ftperc	gamesplayed	minutes	ppg	reb	rpg	spg	steals	turnovers	year
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
5.4	413	36	0.5	813	137	398	34.4	1757	46.3	86.8	77	3140	31.6	439	5.7	1.4	111	255	2006-07
5.4	441	40	0.5	775	150	415	36.1	1690	45.9	84	82	3192	28.3	517	6.3	1.8	151	257	2007-08
4.9	399	37	0.5	800	118	336	35.1	1712	46.7	85.6	82	2960	26.8	429	5.2	1.5	120	210	2008-09
5	365	20	0.3	716	99	301	32.9	1569	45.6	81.1	73	2835	27	391	5.4	1.5	113	233	2009-10
4.7	388	12	0.1	740	115	356	32.3	1639	45.1	82.8	82	2779	25.3	419	5.1	1.2	99	243	2010-11
4.6	264	18	0.3	574	87	287	30.3	1336	43	84.5	58	2232	27.9	313	5.4	1.2	69	204	2011-12
6	469	25	0.3	738	132	407	32.4	1595	46.3	83.9	78	3013	27.3	433	5.6	1.4	106	287	2012-13
6.3	38	1	0.2	31	3	16	18.8	73	42.5	85.7	6	177	13.8	26	4.3	1.2	7	34	2013-14
5.6	197	7	0.2	266	54	184	29.3	713	37.3	81.3	35	1207	22.3	199	5.7	1.3	47	128	2014-15
4.8	6122	627	0.5	11321	1694	5079	33.4	25087	45.1	83.7	1280	46774	25.4	6800	5.3	1.5	1882	3881	TOTAL

We will incorporate interactivity by making the table sort in ascending and descending order when clicked. This also helps the user see which years were the strongest for a player given a certain skill statistic.

Evaluation

Similarly to implementation, we have yet to finish our visualization and will complete our evaluation at a further date.