NBA Shot Chart

Data visualization final project
Jawad Margieh, Amir Sibat

Background

The National Basketball Association (NBA) is the pre-eminent men's professional basketball league in North America, and is widely considered to be the premier men's professional basketball league in the world. It has 30 teams (29 in the United States and 1 in Canada), and is an active member of USA Basketball (USAB), which is recognized by FIBA (also known as the International Basketball Federation) as the national governing body for basketball in the United States.

The NBA Shot Chart has become iconic in many ways. It's a quality visualization that gives context to the boring summarized scoreboard data we see everyday. This context is valuable to basketball nerds and lay folks alike.

Getting the Data

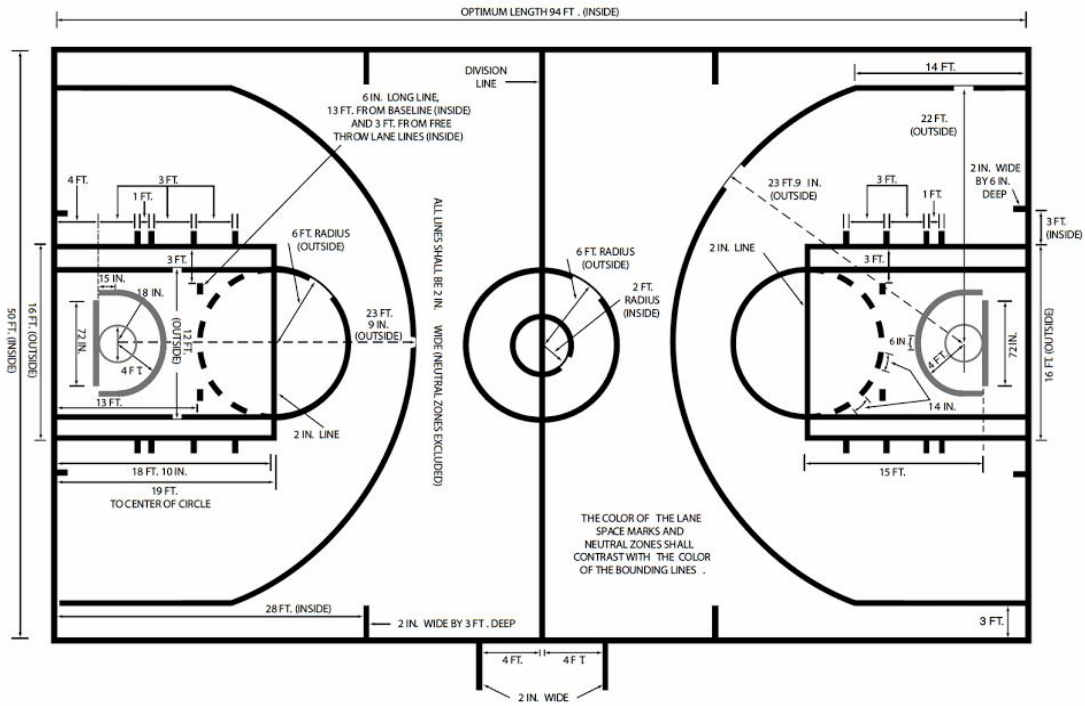
We retrieved the shot data from stats.nba.com. Given a **PlayerID** the REST API returns a JSON object with all of the data for each shot in every game this season unless specified otherwise.

The shot log API from NBA.com returns data about every shot a player took during a game. These data points include how much time was left in the game when the shot was taken, time on the shot clock when the shot was taken. The information We found the most interesting and focused on collecting were the distance the shot was taken from, the location of the shot, the zone area of the shot was taken from and if the shot was made or not.

The image above shows a random shot entry from the JSON object we retrieved using the API for a given **PlayerID** and season.

The marked data attributes were used in our visualization the ones we used in order to show a simpler and comprehensible visualization for the users.
These several attributes were used in multiple visualizations, in which we will discuss later in the next sections.

Plotting Problem



We managed to understand the zones division in the basketball court in order to **map** the X and Y values retrieved from the data into the court shown in the visualization.

The data from the API gives us X values from -250 to 250 and Y values from -1 to -150. We have tried a few values until We got something that looked right to us, we the found correct range by hand, as it varies from svg to another.

In our case in order to map the points correctly we used the **GameID** and **GameEventID** attributes to see where the shot was taken and the **SHOT_DISTANCE** attribute to more accurate points mapping. Using the video and the image above we have managed to map the shots correctly.

Video example: http://stats.nba.com/cvp.html?GameID=0021500959&GameEventID=288#

Over-plotting Problem

The problem

In some graphs, especially those that use data points or lines to encode data, multiple objects can end up sharing the same space, positioned on top of one another. This makes it difficult or impossible to see the individual values, which can undermine analysis. This problem is called over-plotting.

research community has worked hard to come up with over-plotting reduction methods. We'll take a look at the following six:

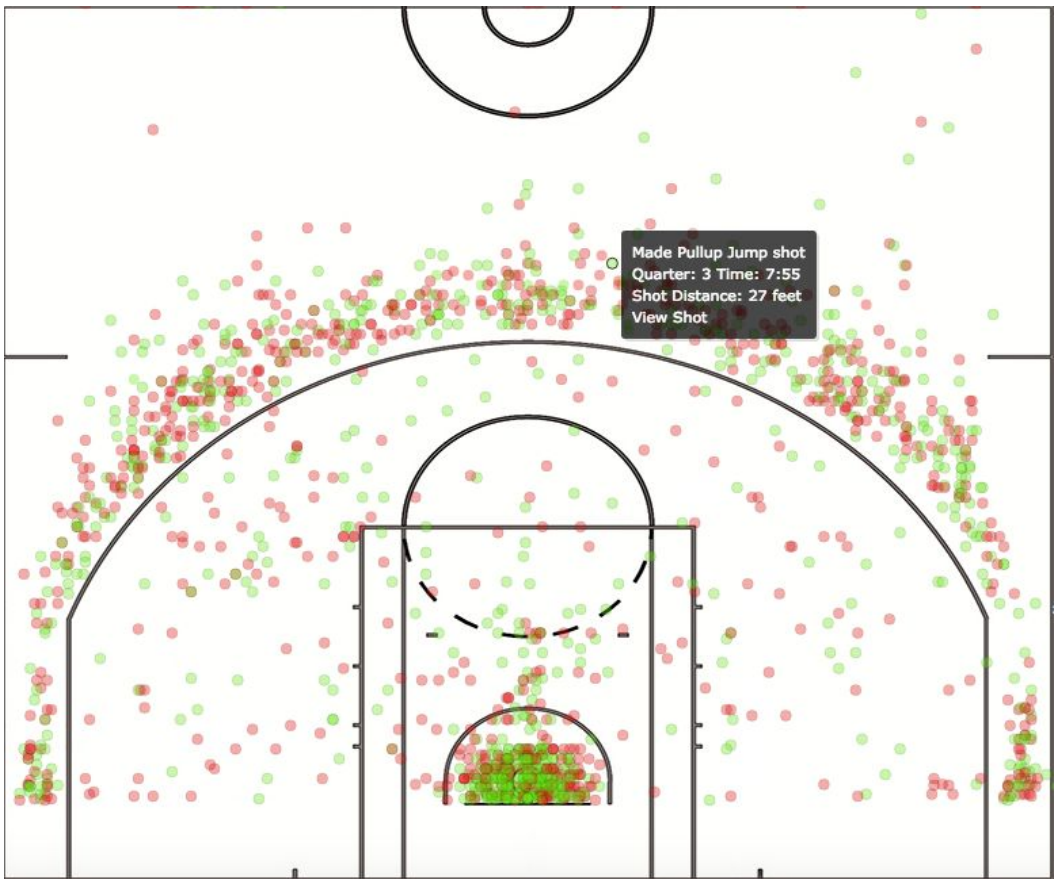
- Reduce the size of data objects
- Remove fill color from data objects
- Change the shape of data objects
- Jitter data objects
- **Make data objects transparent**
- Reduce the amount of data

The solution

In our case we have chosen the 5th option, which is to make data objects transparent.
The proper degree of transparency in our case **0.5** was reasonable to allow us to see through the objects to discern differences in the amount of over-plotting as variations in color intensity. figure(1)
That allows us to easily detect differences between the dense center around the rim,which is intensely filled, versus other areas of progressively less concentration (less intensely filled).

The Visualisation

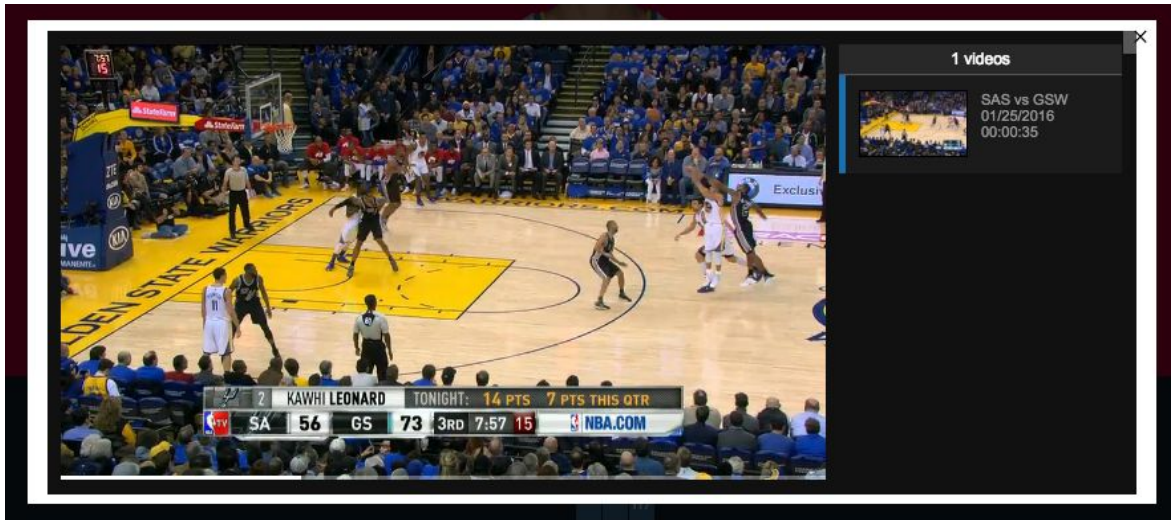
Below you can see the shots mapping into the court, the green points indicate a **made shot** while a red points indicate a **missed shot**.



Figure(1) - player shot mapping and shot info

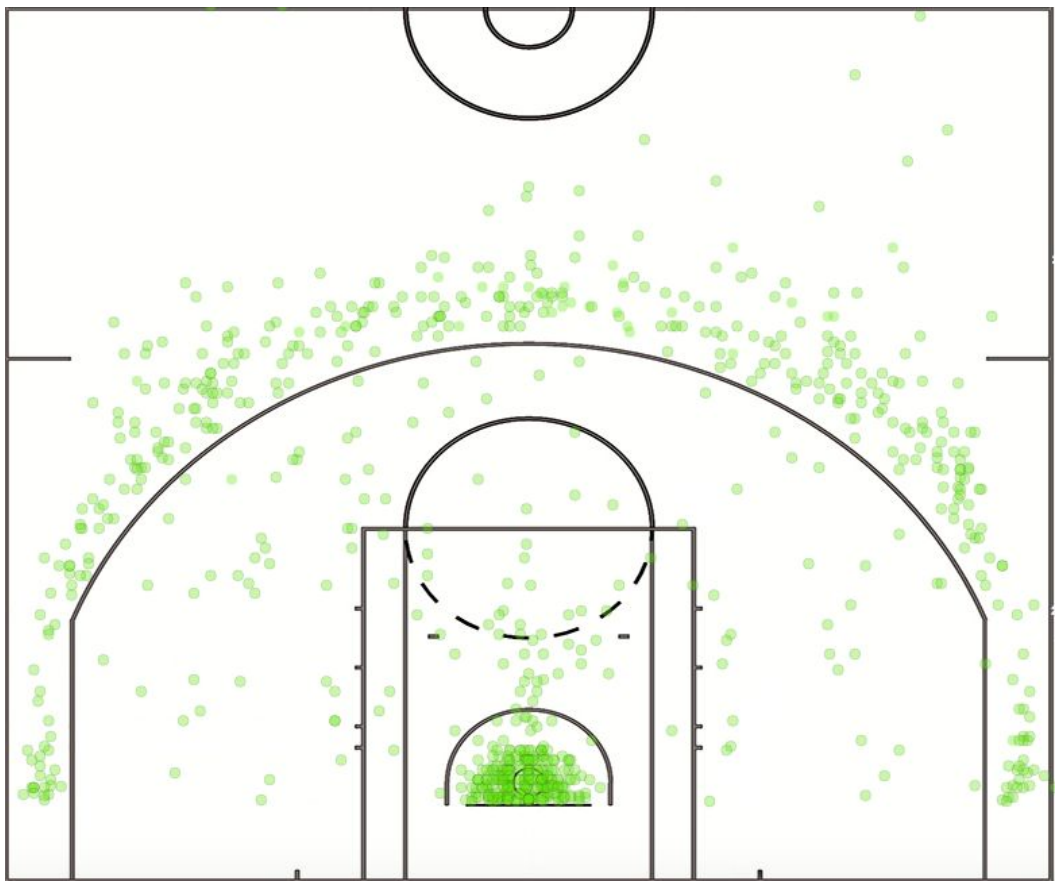
User interaction

- In order to make the visualization more interactive, clicking on the point will show a tooltip with more info about the shot, e.g shot type, shot distance, time the shot was made and a link for a video for the shot. figure(1)+figure(2)



figure(2) - player shot video

- The user as well will have an option to show only the made shots.

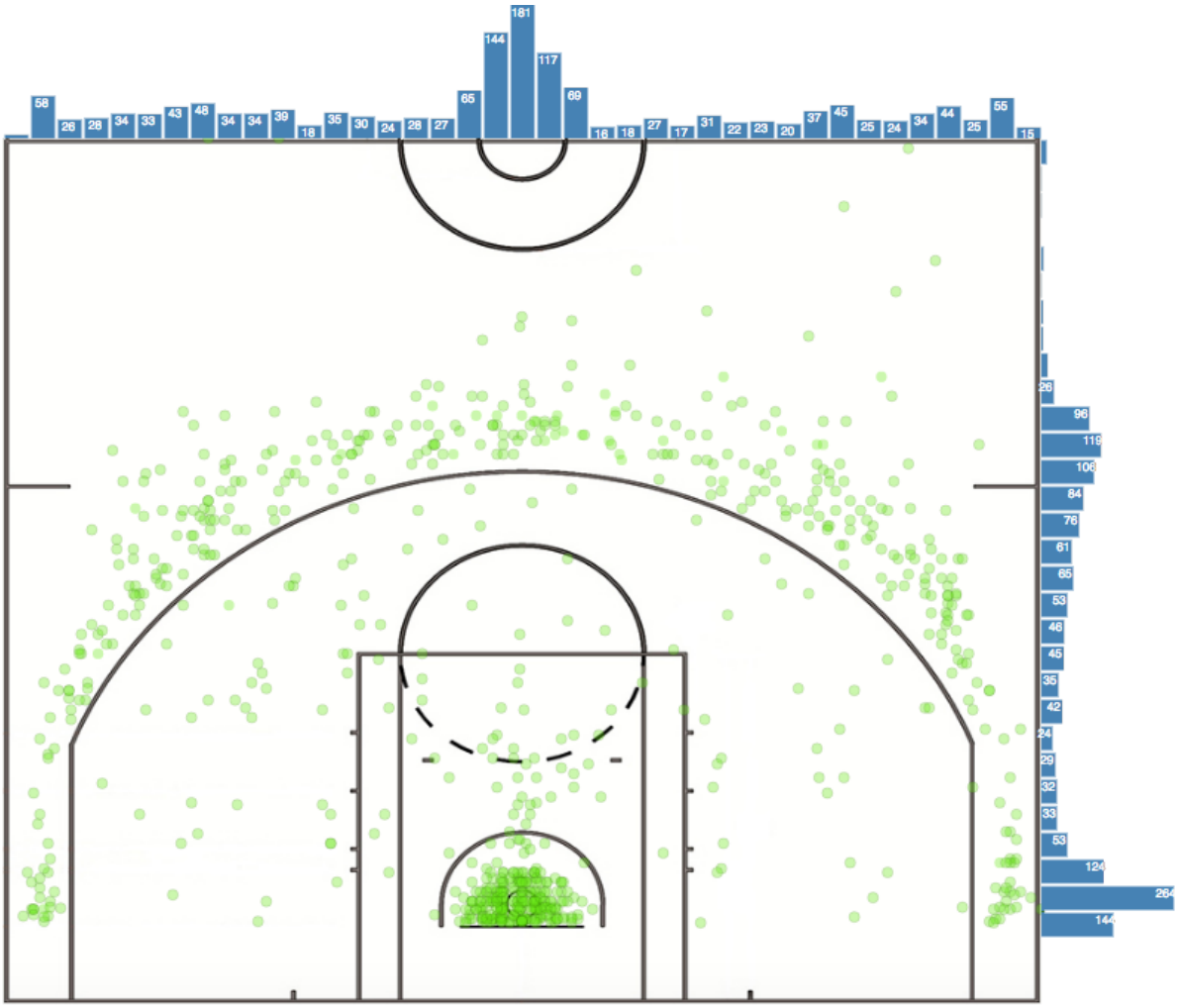


Visualizing the distribution of a dataset

When dealing with a set of data, often the first thing you'll want to do is get a sense for how the variables are distributed.

Data distributions are used often in statistics. They are graphical methods of organizing and displaying useful information. There are several types of data distributions. In our visualization we used histograms (jointplot).

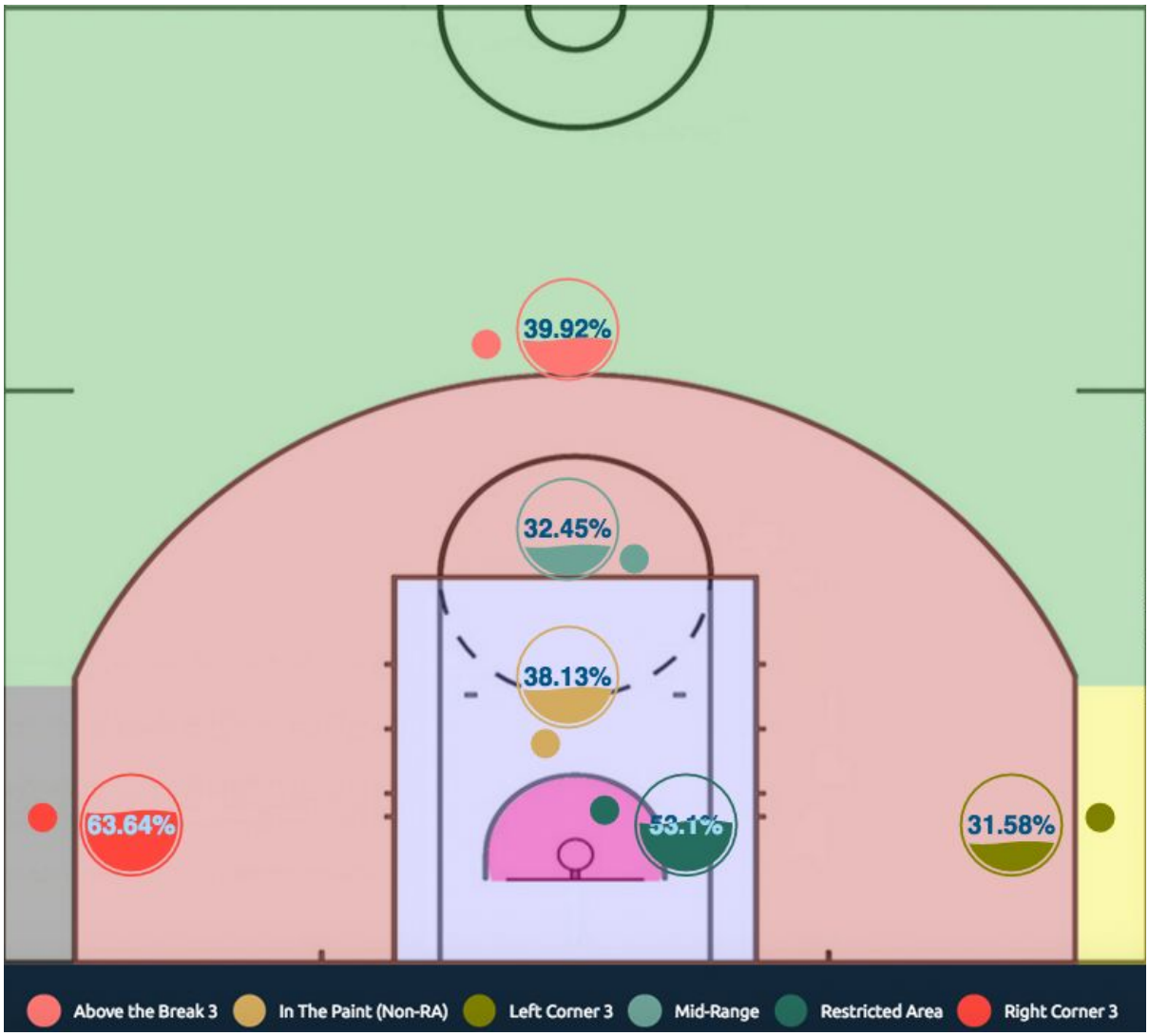
The histogram in our case displays data in court segments, with each bar representing a certain segment (bar width). The height of the bar tells the frequency(number of shots) of values that fall within that court segment.



figure(3) - data distribution

Visualising player shots stats by areas

In order to display the player shot stats we had to iterate through the data and calculate for each zone area the made shots percentage.
The visualisation shows the percentage of made shots for each one of the areas. The court is divided into areas as shown below.

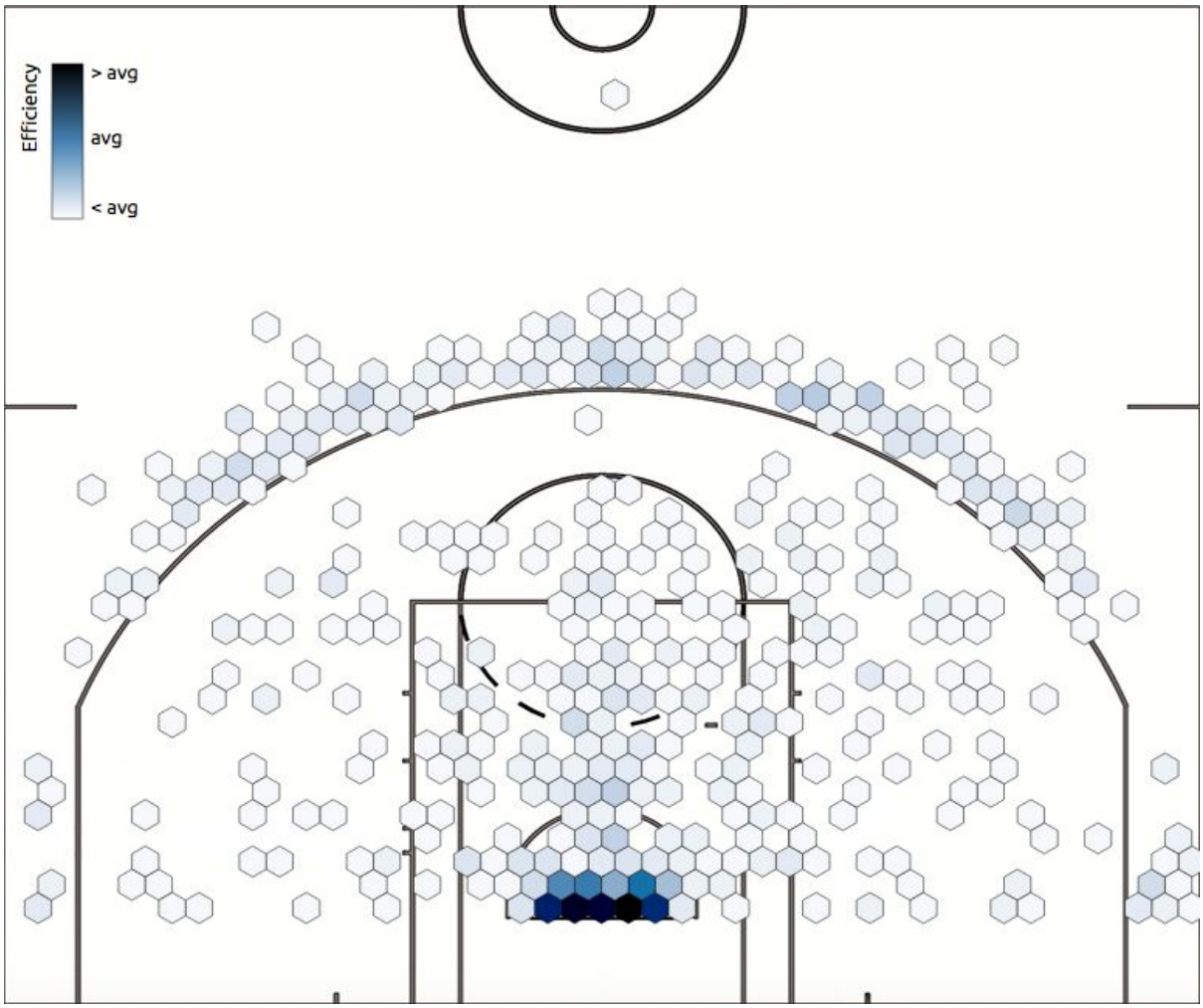


figure(4) - player shots stats by zone area

Visualising player shots using Hexagonal Binning

Scatterplots are a straightforward way to visualize the data distribution in a XY plane, especially when we are looking for trends or clusters. But when you have a dataset with a large number of points, many of these data points can overlap. This **overlapping** effect can make difficult to see any trends or clusters.

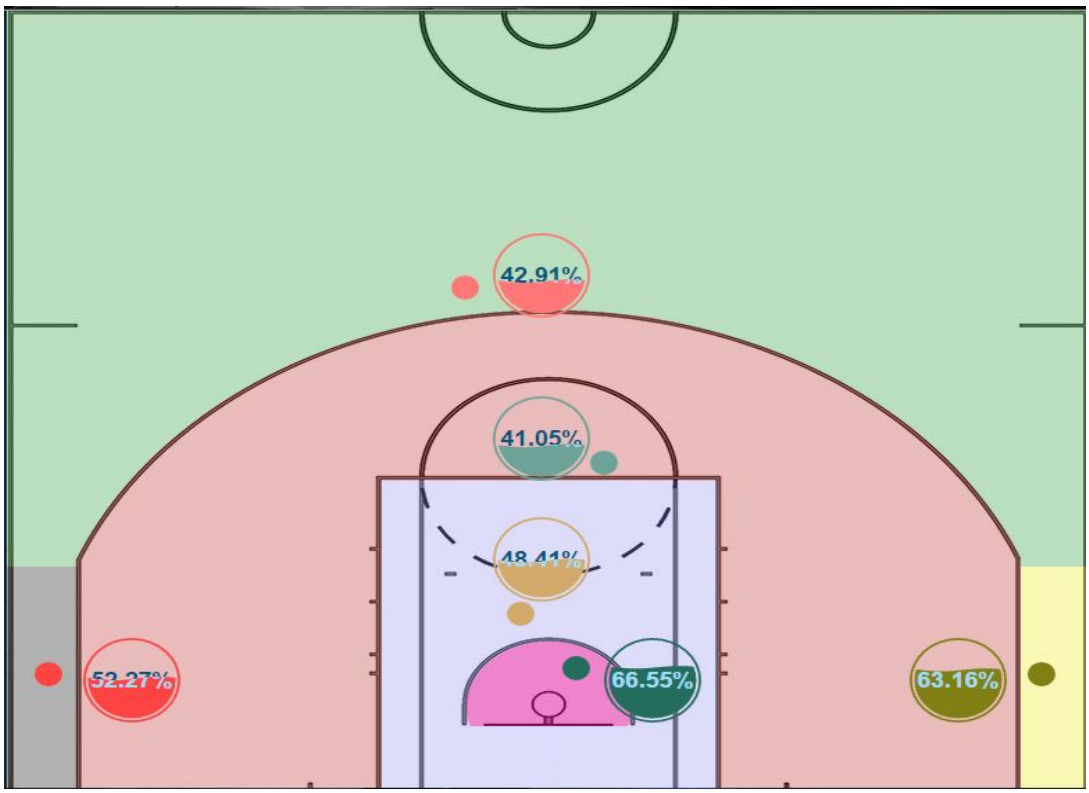
Binning is a technique of data aggregation used for grouping a dataset of N values into less than N discrete groups.
the XY plane is uniformly tiled with polygons (squares, rectangles or hexagons).
the number of points falling in each bin (tile) are counted and stored in a data structure.
the bins with count > 0 are plotted using a color range (**heatmap**) or varying their size in proportion to the count in other words the color encodes the number of points that fall into each bin.



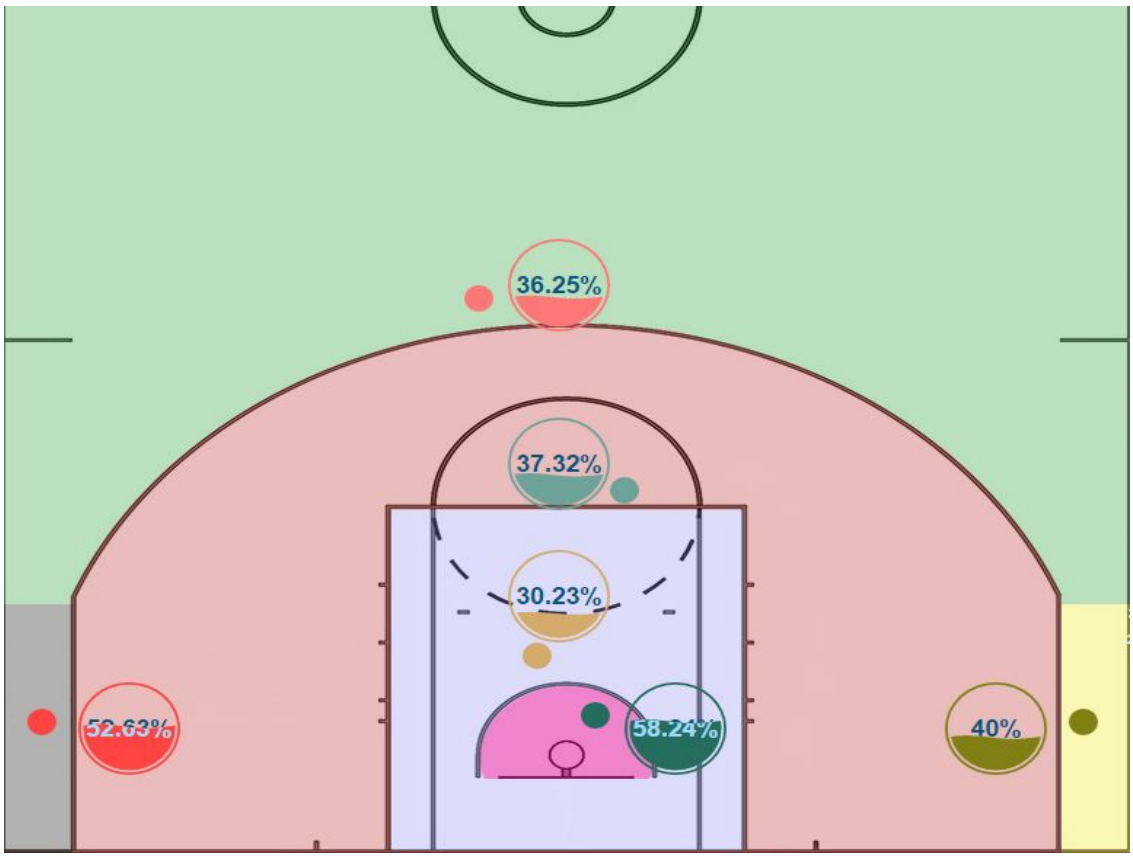
figure(5) - player shots using hexagonal binning

User tasks

1. Is Stephen Curry on his way to another MVP title?



Curry isn't the 2014-15 MVP by chance. He made 48.7% of field goals attempted during the regular season. From the left 3-point corner, he converted 63.2% of shots attempted (almost 2 in every 3 attempts). Under the rim Curry is very effective with 66.55% accuracy when going for those quick lay-ups and finger rolls.



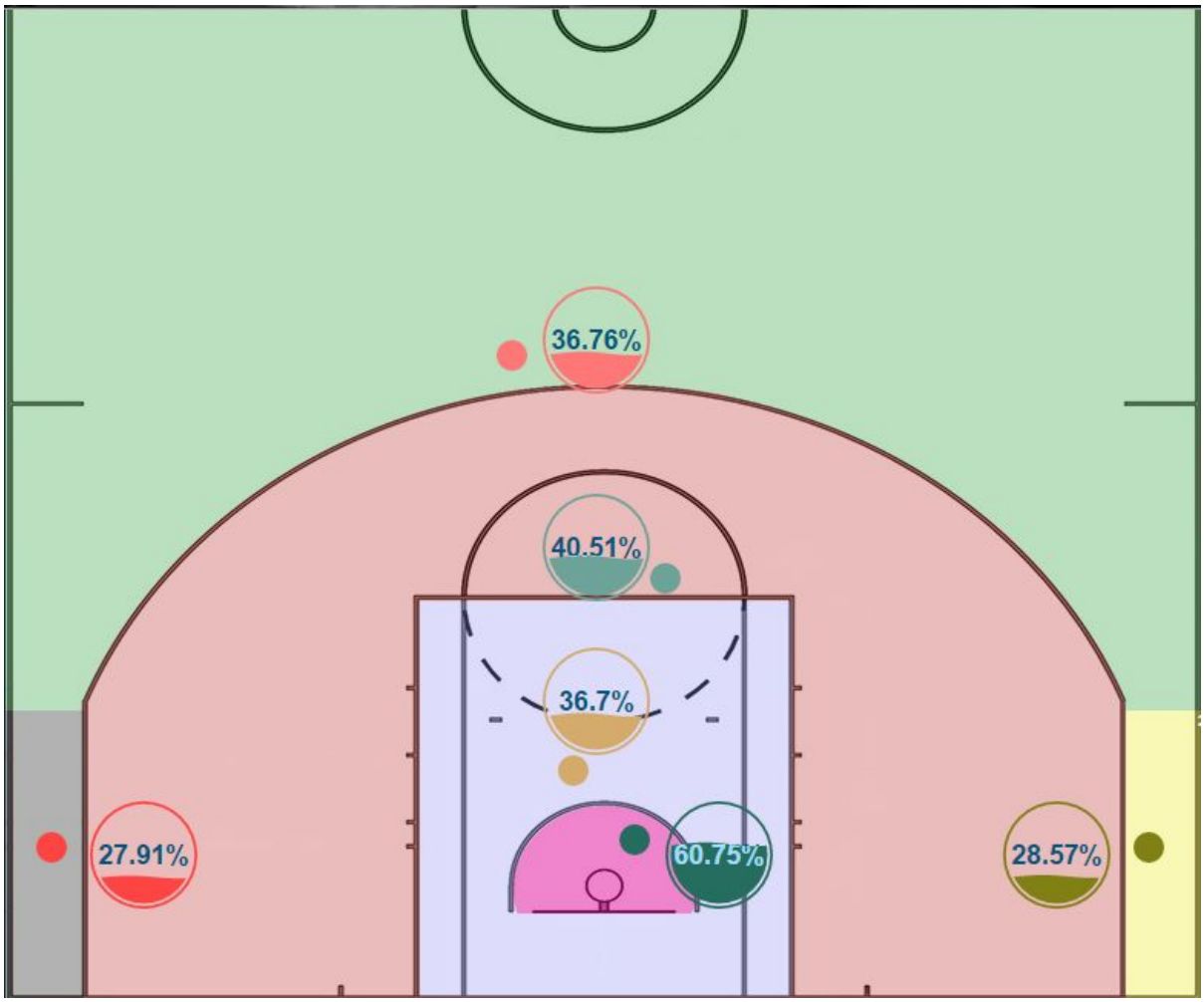
James Harden, the other MVP contender, is also a great 3-point shooter, but not as accurate as Curry. Harden is slightly better from the right 3-point corner but Curry is better from every other zone in the court.

2. Compare between any two players in the league.

We can compare accuracy shots and shots distribution between any two players to decide who is a better shot.
Above for instance, we can see the comparison between Stephen Curry & James Harden.

3. Check if a certain player improved his play from season to season

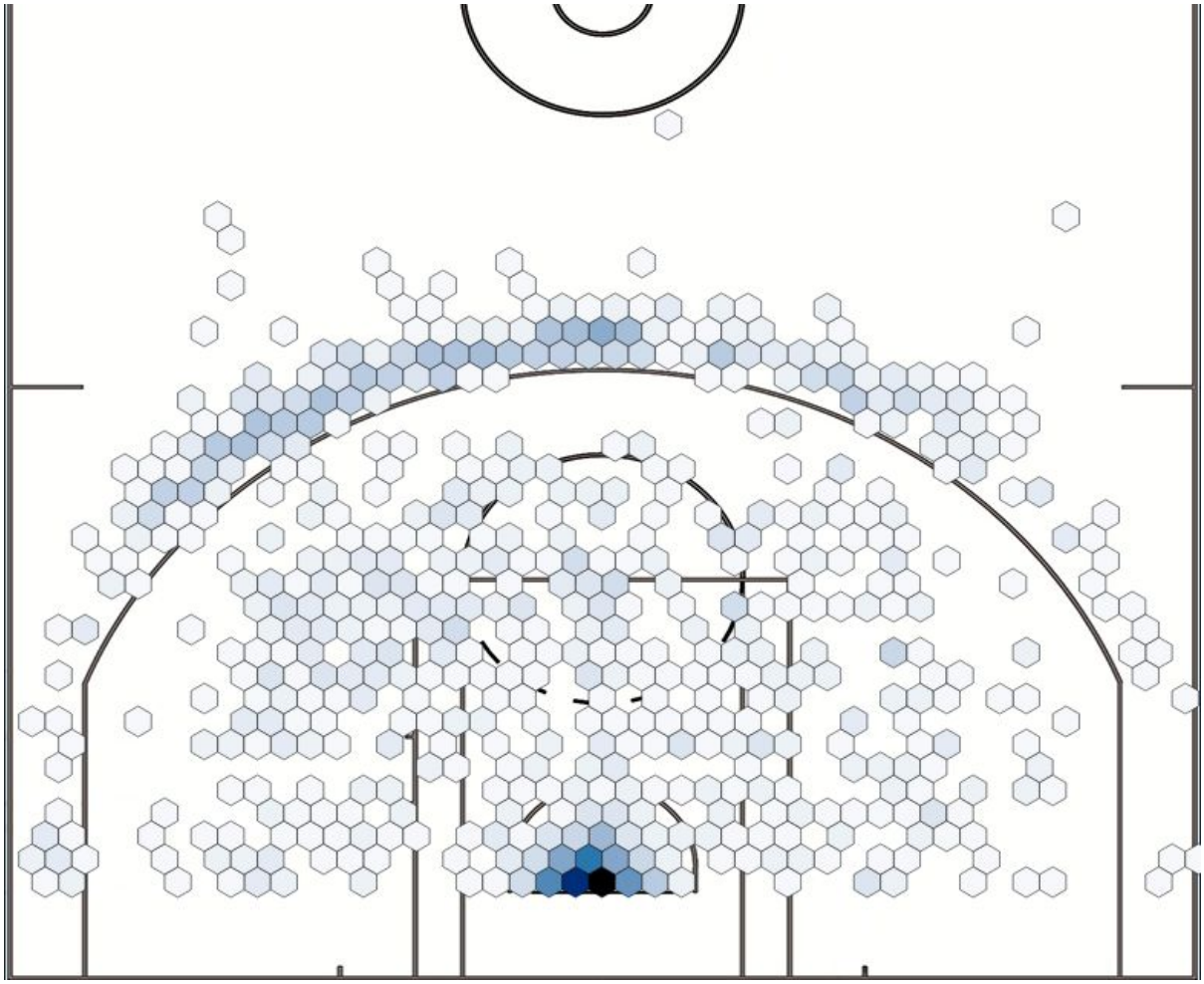
These are the accuracy stats of James Harden 2015-2016 regular season



We can see in comparison to that Harden improved his Mid-Range and under the Rim shots.
Harden got better than last season, but still needs to work on his 3-point shots.

4. Most common shots location

These are the shots locations (by radius) of Kevin Durant's 2015-2016 regular season



We can see the Durant's favorite spot is right under the rim, he's a 3-point shot as well.

Discussion - Improvements

The work done for this project shows that there is still much progress to be made in the area of NBA information visualization, what we have done is just a glance of what can be done with this informative data. The NBA's methods of data collection need to change if statistical analysis of game performance wants to take into account team and player interactions. Statistics based on individuals can only go so far in helping evaluate the game.

There is much more work to be done to improve visualization tools for NBA analysis. Tools need to be able to visualize a vast variety of data in one place. Increasing the flexibility and adding functionality to visualizations such as simple shot charts can make them much more useful.
A tool such as two players comparison should be available sometime in the future, in order to see the full picture and to inspect players performance and efficiency.

The Value Of Visualization

V = T + I + E + C

T = The ability to minimize total **time** needed to answer a wide variety of questions (without formal queries, ie not needing to know SQL).

In order to answer any question (User Task) with our project all a user needs to do is choose your player and the season. The variety of results (Visualizations) and answers will be shown in one click.
Therefore we minimized a lot of time if a user did this analysis by himself, by retrieving the data from nba.com or reading it from a table.

I = The ability to spur and discover **insights** or insightful questions about the data.

“If you don’t learn anything from your visualization, you have not succeeded.”
Any question asked or a user task as mentioned above, can be answered by our visualization, therefore learning if Stephen Curry has improved from last season or will he win another MVP title, is possible and doable by any user.

E = Ability to convey an overall **essence** or take-away sense of the data.

Using all the visualizations together, we could manage to understand the big picture

References

- How to Create NBA Shot Charts in Python
  http://savvastjortjoglou.com/nba-shot-sharts.html

- How to create NBA shot charts in R
  https://thedatagame.com.au/2015/09/27/how-to-create-nba-shot-charts-in-r/

- Web Scraping 201: finding the API (NBA API)
  http://www.gregreda.com/2015/02/15/web-scraping-finding-the-api/

- over-plotting problems
  https://www.perceptualedge.com/articles/visual_business_intelligence/over-plotting_in_graphs.pdf

- Hexagonal Binning
  https://bl.ocks.org/mbostock/4248145

- D3 Liquid Fill Gauge
  http://bl.ocks.org/brattonc/5e5ce9beee483220e2f6