

Using the SpatialBall package

Derek Corcoran

2017-10-02

Contents

1 Visualization Tools	1
1.1 Season visualization tools:	1
1.1.1 ShotSeasonGraph	1
1.2 Team visualization tools:	4
1.3 DefShotSeasonGraphTeam:	4
1.3.1 OffShotSeasonGraphTeam	6
1.3.2 PointShotSeasonGraphTeam	8
1.4 Comparative visualization tools:	9
1.5 Player visualization tools:	11
2 Analysis Tools:	15
2.1 Spread calculation	15
2.2 Ranking Analysis	15
2.3 Other minor functions and programmatic possibilities	16

The `SpatialBall` package was developed to visualize and analyze basketball data from the NBA, we will start by presenting several of our visualization functions, and later we will present our analysis tools

1 Visualization Tools

1.1 Season visualization tools:

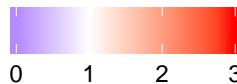
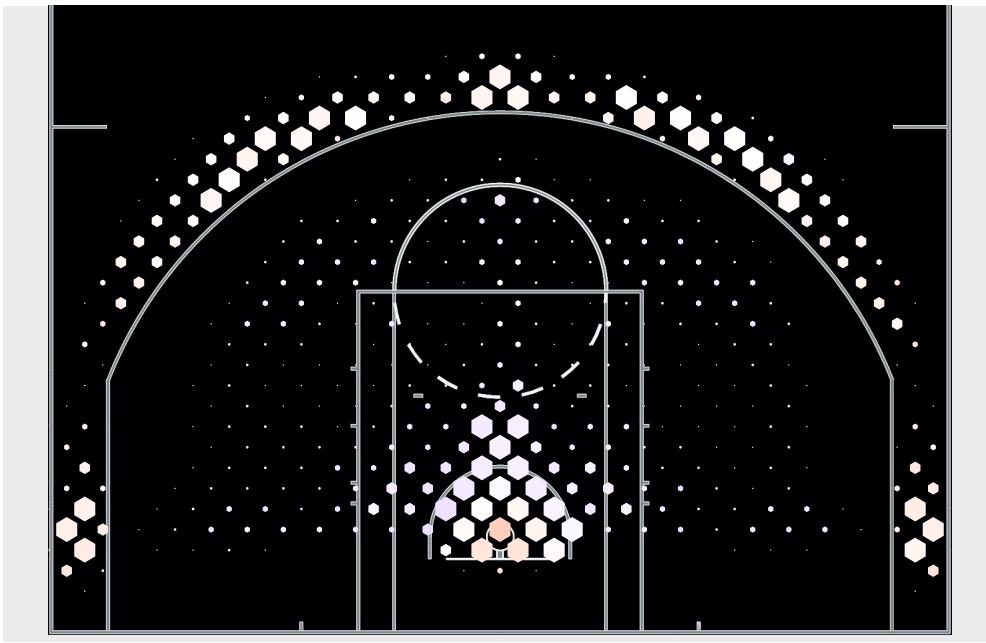
We have two functions to plot all the shots of a Season, `ShotSeasonGraph` and `PointShotSeasonGraph` the latter is not going to be shown in this vignette

1.1.1 ShotSeasonGraph

This function make the classic shot charts developed by Kirk Goldsberry, the default shows an hexagon in each point of the court, the bigger the hexagons, the more frequent the shots are, and the smaller the hexagons the least frequent the shots are. Also the color scale, in this case, shows the points per shot.

```
library(SpatialBall)
data("season2017")
ShotSeasonGraph(Seasondata = season2017)
```

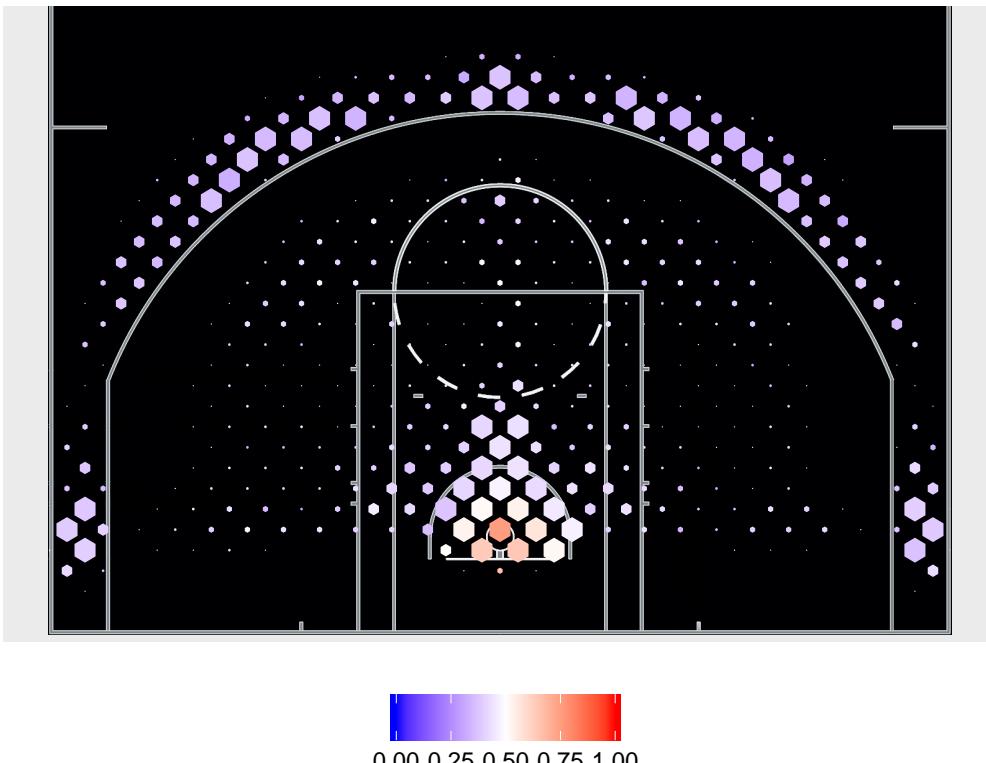
Points per Shot



for `ShotSeasonGraph` you have two options for the Type of graph, the default (Shown above) is Points per Shot, you can change that to *PCT* to see the shooting percentage:

```
ShotSeasonGraph(Seasondata = season2017, type = "PCT")
```

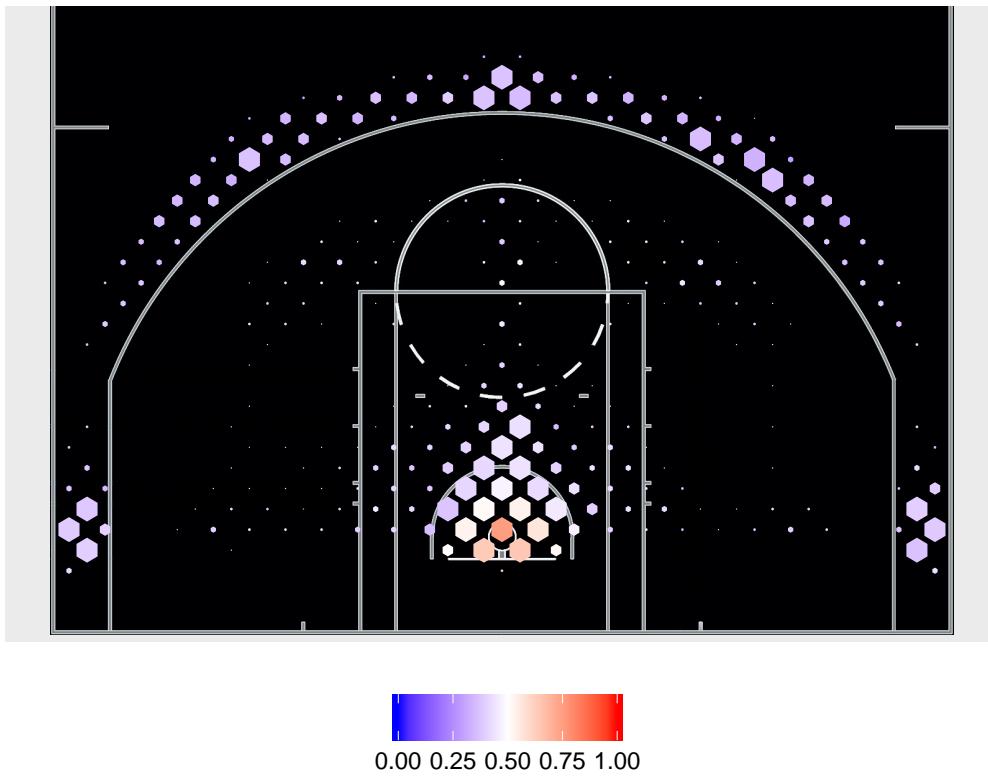
Shooting percentage



another option for this function is the quantile threshold of which shots are plotted, the default is 0.4, that means that the 40% least common shots are not plotted, if we increase that number, less hexagons will be shown

```
ShotSeasonGraph(Seasondata = season2017, type = "PCT", quant = 0.6)
```

Shooting percentage



1.2 Team visualization tools:

Just as season visualization tools, you can choose to plot shot charts or shot every shot of the team, you have to add to these the team you want to plot the shot-chart of, for *Season2017* the code for the teams are NY, Cle, Uta, Por, Sas, GSW, Mia, ORL, Ind, Dal, Bkn, Bos, Det, Tor, Cha, Mil, Min, Mem, NO, Den, Okc, Phi, Sac, Pho, Hou, Lal, Was, Atl, Chi, Lac, since teams have changed names over time, for each season you should use the following code to check the names `unique(season2017$TEAM_NAME)`, changing *season2017* for the season you want to plot.

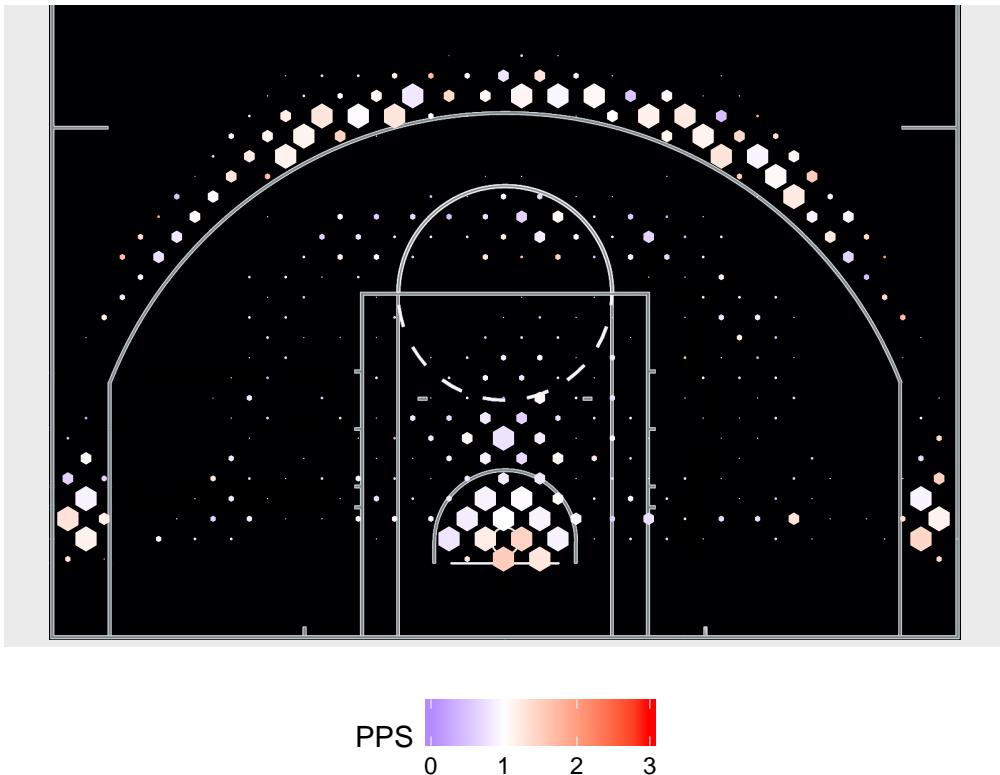
We have 2 functions for the visualization of team shot charts `DefShotSeasonGraphTeam` and `OffShotSeasonGraphTeam`, both of them are similar to the `ShotSeasonGraph` function, but focus on one team, and let's you get the offensive or defensive shot chart for each team in either Point per shot (`type = "PPS"`) or percentage (`type = "PCT"`).

1.3 DefShotSeasonGraphTeam:

These are examples of Charlotte's defensive Shot Charts as PPS and PCT

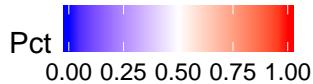
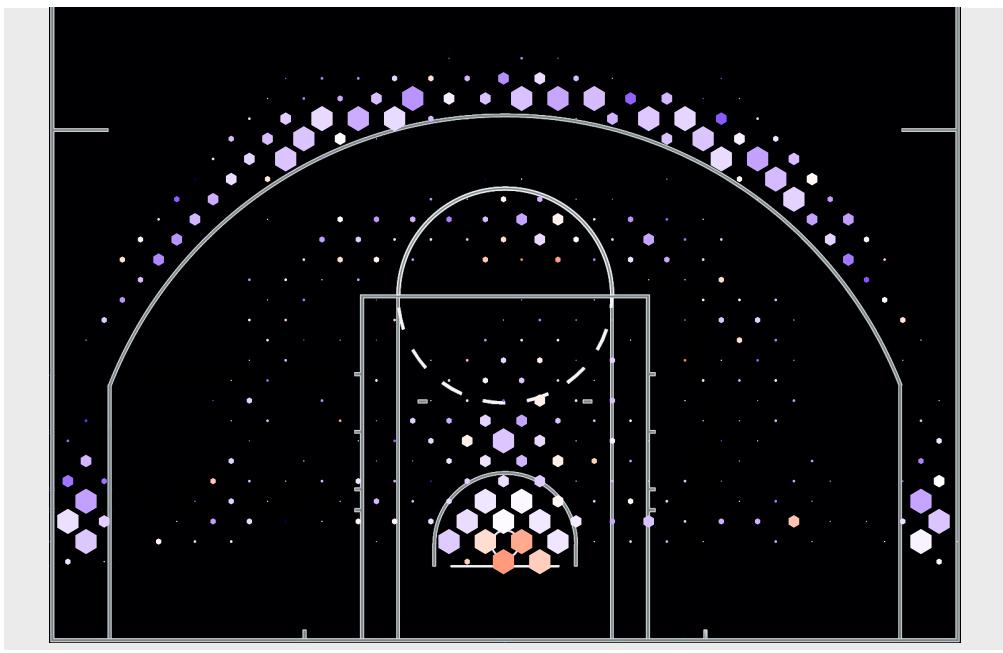
```
DefShotSeasonGraphTeam(season2017, team = "Cha")
```

Defensive shot chart of Cha



```
DefShotSeasonGraphTeam(season2017, team = "Cha", type = "PCT")
```

Defensive shot chart of Cha

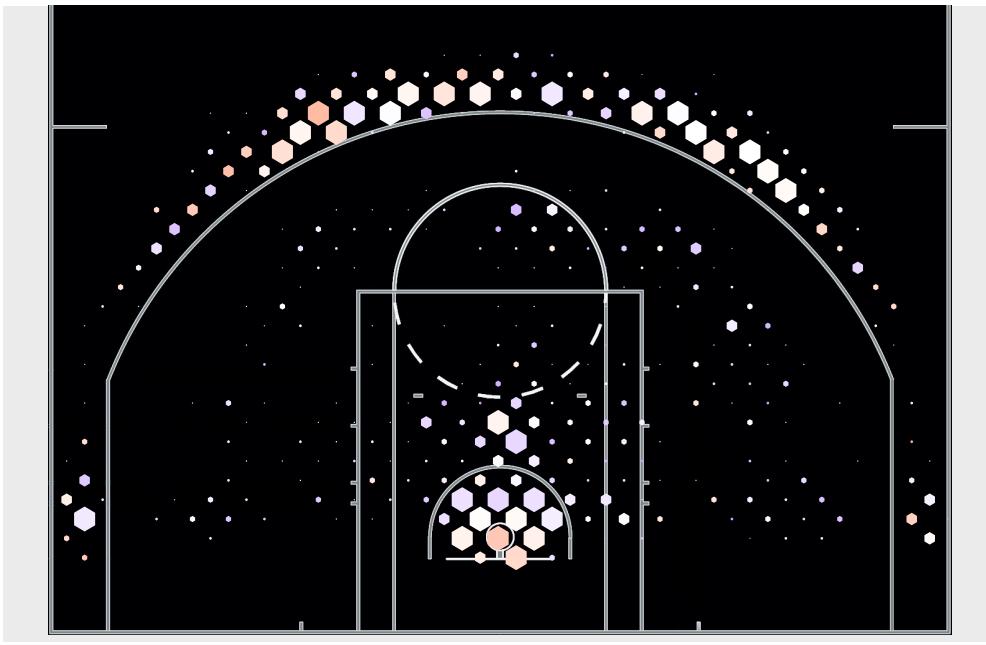


1.3.1 OffShotSeasonGraphTeam

As an example for the Offensive shot charts we will shot for the 2017 season the shot charts for the Charlotte Hornets:

```
OffShotSeasonGraphTeam(season2017, team = "Cha")
```

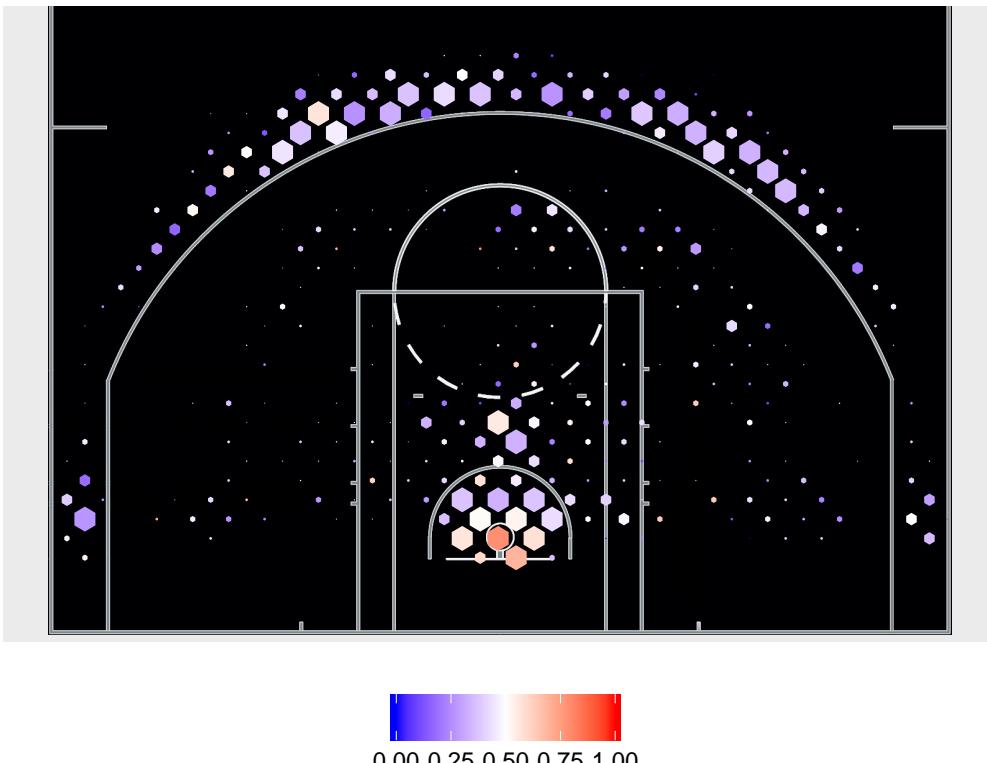
Points per Shot of Cha



Again we can change the graph to Shooting percentage by changing type to *PCT*

```
OffShotSeasonGraphTeam(season2017, team = "Cha", type = "PCT")
```

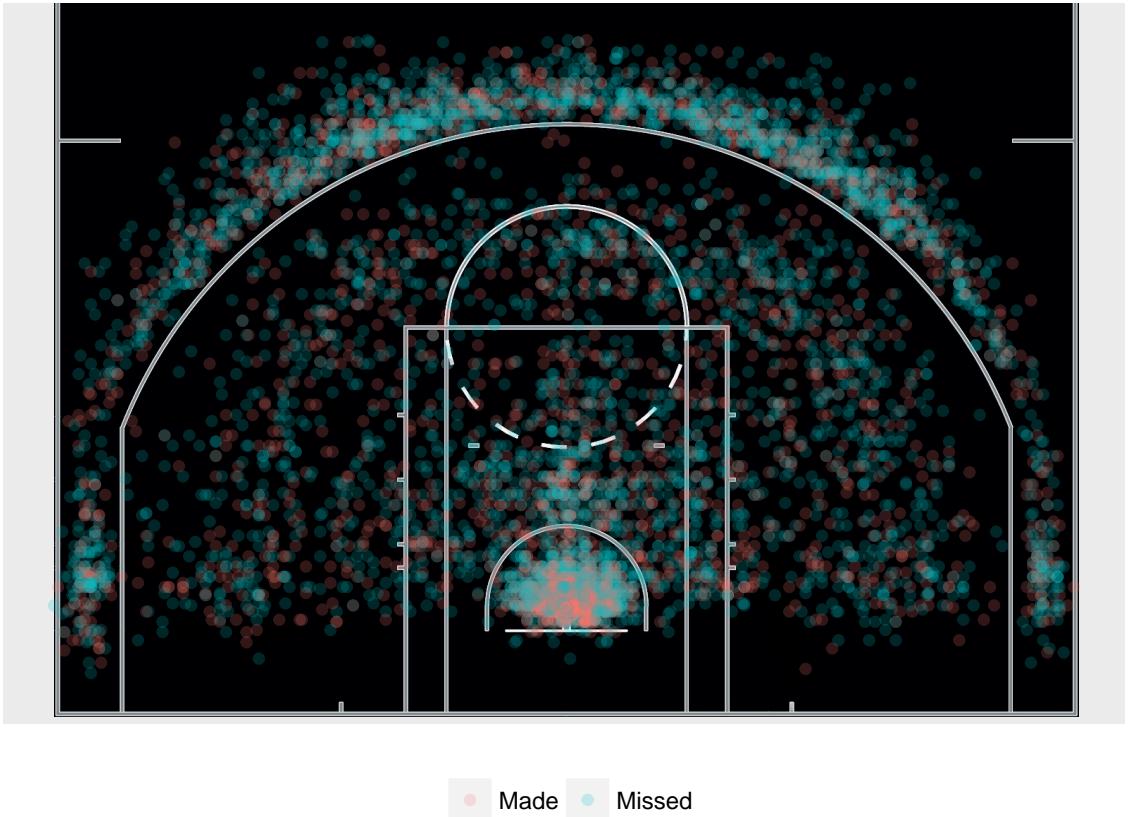
Shooting percentage Cha



1.3.2 PointShotSeasonGraphTeam

If instead of a Shot-chart based on hexagons, you want to plot every shot as a single point, you can do that with *PointShotSeasonGraphTeam*, this divides shots in made and misses, also each shot has a high transparency (Alpha of 0.2 in ggplot), in order for areas of low density shots having less colors than the areas with more shots.

```
PointShotSeasonGraphTeam(season2017, "Cha")
```

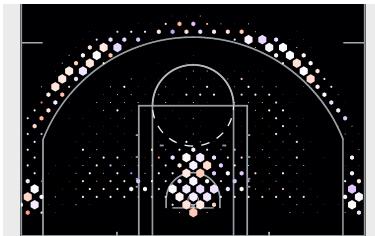


1.4 Comparative visualization tools:

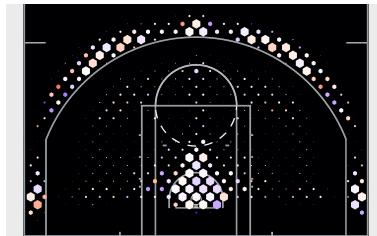
The function to visually compare two teams it's called *ShotComparisonGraph*, and it uses our comparative shot charts, in this example we have Charlotte against Cleveland. In the top row we see the defensive shot chart of the cavaliers, the Offensive Shot Chart of the Hornets, and the net Shot Chart of the expected points per Shot when Hornets Faces the Cavaliers Defense. At points where the hexagons are blue, the defense is allowing fewer points than average and the offense is scoring fewer points than average, and the inverse is true for red hexagons. Conversely the bottom row shows the inverse.

```
data("season2017")
ShotComparisonGraph(HomeTeam = "Cha", VisitorTeam = "Cle", Seasondata = season2017)
```

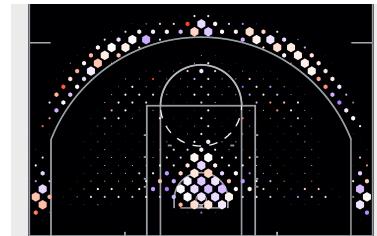
**Cle defensive
Shot Chart**



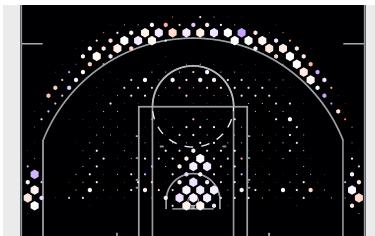
**Cha Offensive
Shot Chart**



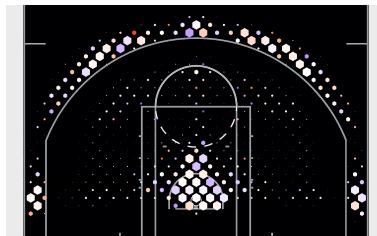
**Comparison
Shot Chart**



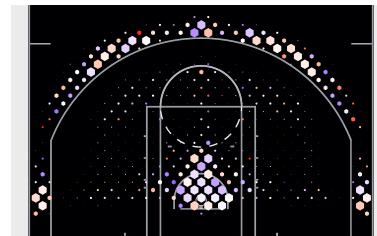
**Cha defensive
Shot Chart**



**Cle Offensive
Shot Chart**



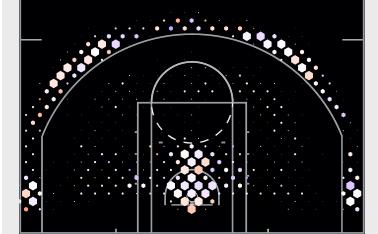
**Comparison
Shot Chart**



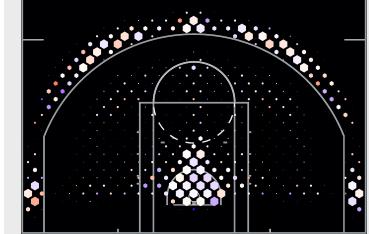
We can focus on the areas where the offensive team is favored by adding to the code the *focus = "plus"* command, as follows, which will change only the last column of the graph.

```
data("season2017")
ShotComparisonGraph(HomeTeam = "Cha", VisitorTeam = "Cle", Seasondata = season2017, focus = "plus")
```

**Cle defensive
Shot Chart**



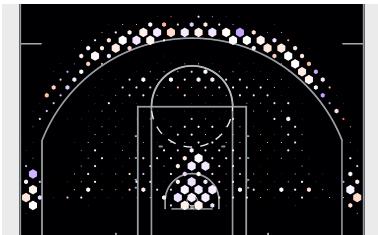
**Cha Offensive
Shot Chart**



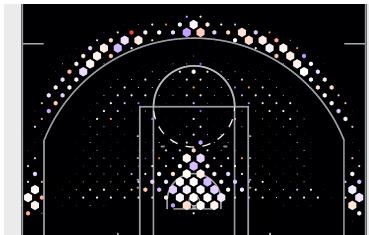
**Comparison
Shot Chart**



**Cha defensive
Shot Chart**



**Cle Offensive
Shot Chart**



**Comparison
Shot Chart**

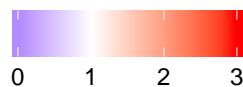
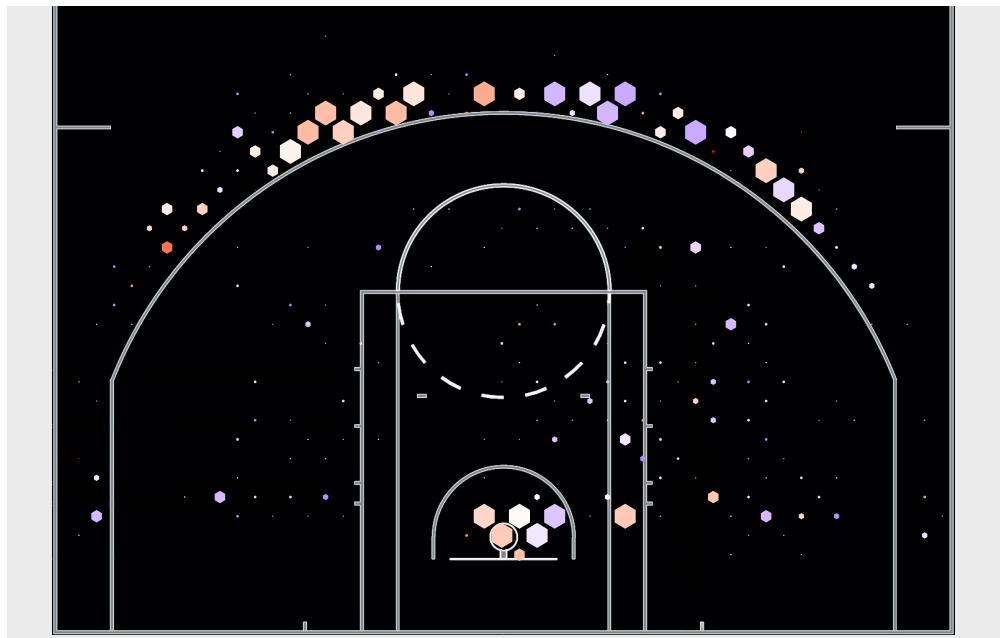


1.5 Player visualization tools:

We can also get shot charts for each player, both as PCT and as PPS, in this case we will just show the PPS for four of Charlotte's last season players:

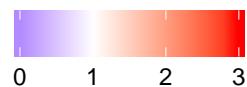
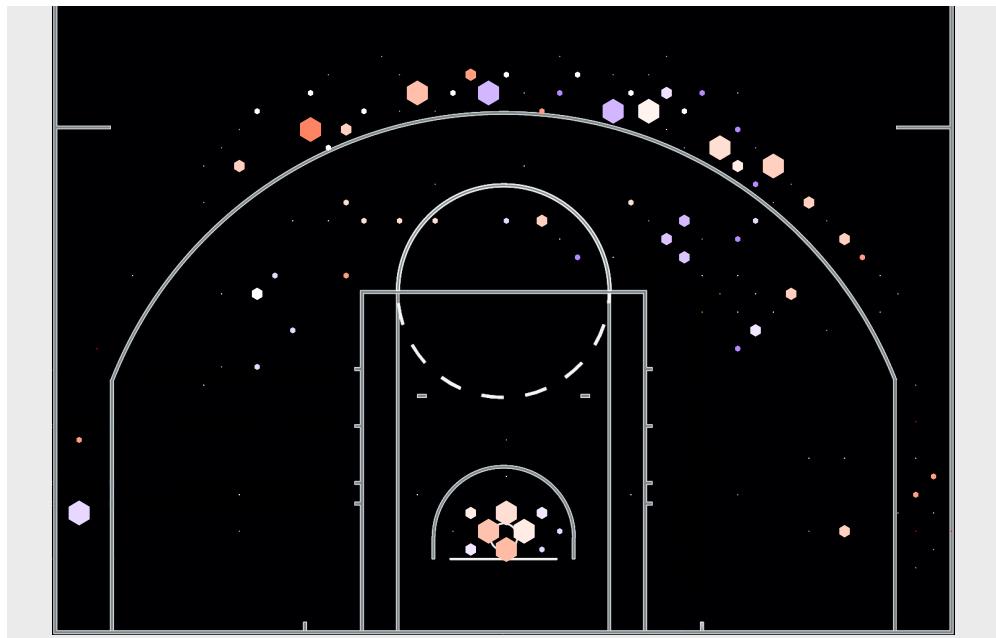
```
data("season2017")
OffShotSeasonGraphPlayer(season2017, player = "Nicolas Batum")
```

Points per Shot of Nicolas Batum



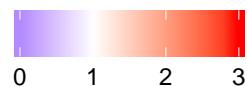
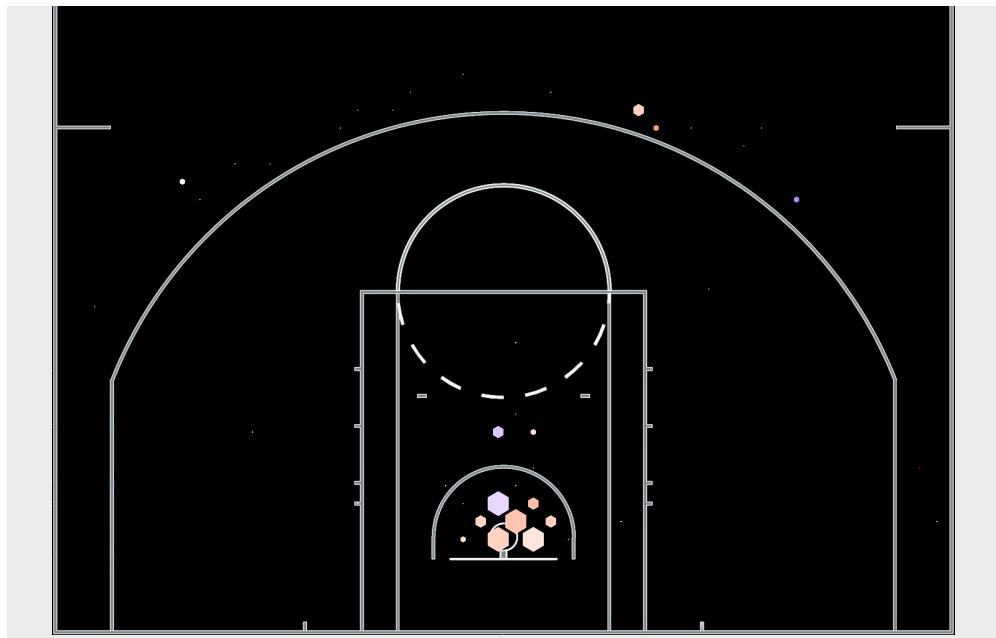
```
OffShotSeasonGraphPlayer(season2017, player = "Marco Belinelli")
```

Points per Shot of Marco Belinelli



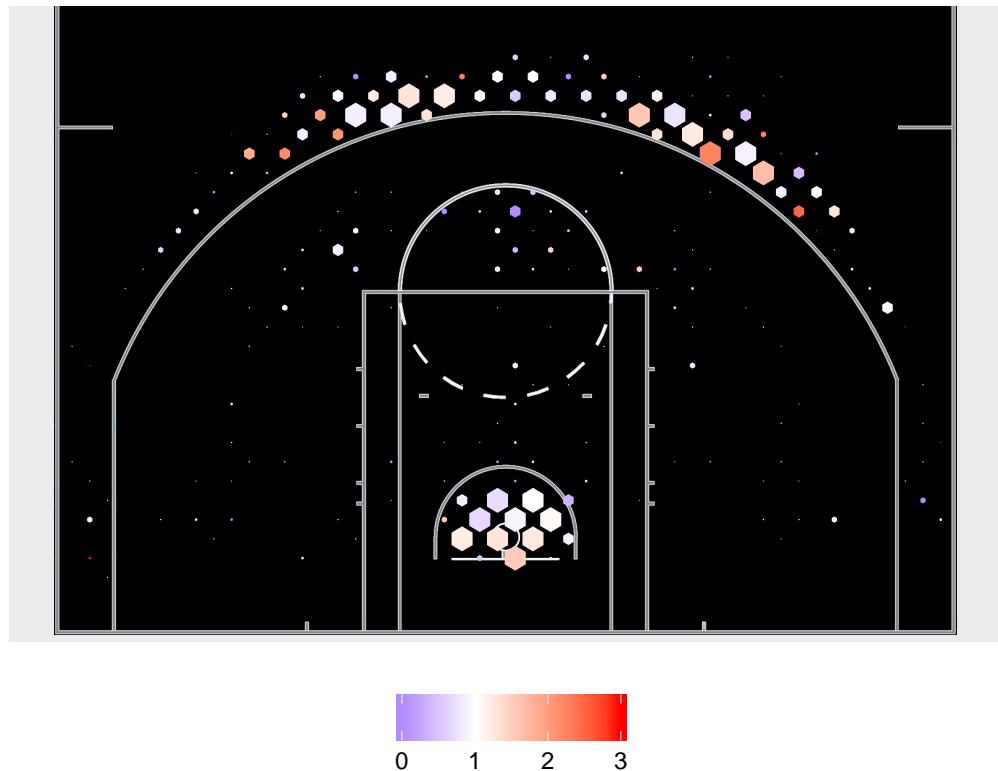
```
OffShotSeasonGraphPlayer(season2017, player = "Spencer Hawes")
```

Points per Shot of Spencer Hawes



```
OffShotSeasonGraphPlayer(season2017, player = "Kemba Walker")
```

Points per Shot of Kemba Walker



2 Analysis Tools:

2.1 Spread calculation

The function *Get_Apps* will analyse the pairwise spatial components of the teams involved in the game proposed and using our algorithm based on Boosted Regression Trees will predict the spread of the game, as an example we will show the predicted spread for Charlotte playing against Boston at Home and on the road:

```
data("season2017")
Get_Apps(HomeTeam = "Cha", VisitorTeam = "Bos", Seasondata = season2017)

##      defAPPS      offAPPS      spread
## 1 -0.01542924 -0.01071023 -2.756698

Get_Apps(HomeTeam = "Bos", VisitorTeam = "Cha", Seasondata = season2017)

##      defAPPS      offAPPS      spread
## 1 -0.01071023 -0.01542924 -3.873041
```

2.2 Ranking Analysis

Finally our function *SpatialRating* will calculate the offensive, defensive and net rating for every team, with the shots to the date during the season.

```
SpatialRating(Seasondata = season2017)
```

Team	offrating	defrating	netrating
GSW	2.0569005	0.2414252	2.2983257
Sas	1.0067770	0.6344353	1.6412123
Lac	1.3668864	-0.2681053	1.0987811
Cle	0.3353182	0.4217216	0.7570398
Uta	0.4122279	0.3277828	0.7400107
Ind	1.0394992	-0.4342134	0.6052858
Bos	0.0861130	0.4114464	0.4975594
Tor	0.7761220	-0.3063311	0.4697909
NO	0.1867078	0.1321713	0.3188791
Cha	-0.4501570	0.7518839	0.3017269
Was	0.8100781	-0.5559126	0.2541656
Mil	-0.6378466	0.8539371	0.2160905
Mia	-0.3135702	0.2532356	-0.0603347
Por	0.0131529	-0.0956242	-0.0824713
Hou	-0.0208501	-0.1549108	-0.1757609
Atl	-1.0759928	0.8583068	-0.2176859
Mem	-0.9081215	0.6783792	-0.2297423
NY	-0.8083317	0.5713893	-0.2369424
Den	0.6455375	-0.9798399	-0.3343024
Chi	-0.8294621	0.4529047	-0.3765574
Det	-0.9832813	0.5840699	-0.3992114
Phi	-1.0021905	0.5079217	-0.4942688
Dal	-0.0716884	-0.4395447	-0.5112331
Sac	0.6515087	-1.2308412	-0.5793325
Bkn	-0.5912906	0.0055238	-0.5857667
Min	-0.4290214	-0.3156964	-0.7447179
ORL	-0.8101257	0.0504517	-0.7596741
Okc	-1.3723541	0.4252067	-0.9471474
Pho	-1.1827070	0.0468850	-1.1358220
Lal	-0.6883061	-0.6395905	-1.3278966

2.3 Other minor functions and programmatic possibilities

The package also calculates the percentage of shots taken and made by category (Threes, twos, dunks, etc.). It can also recalculate the spread assuming a player is injured, or traded by another player.