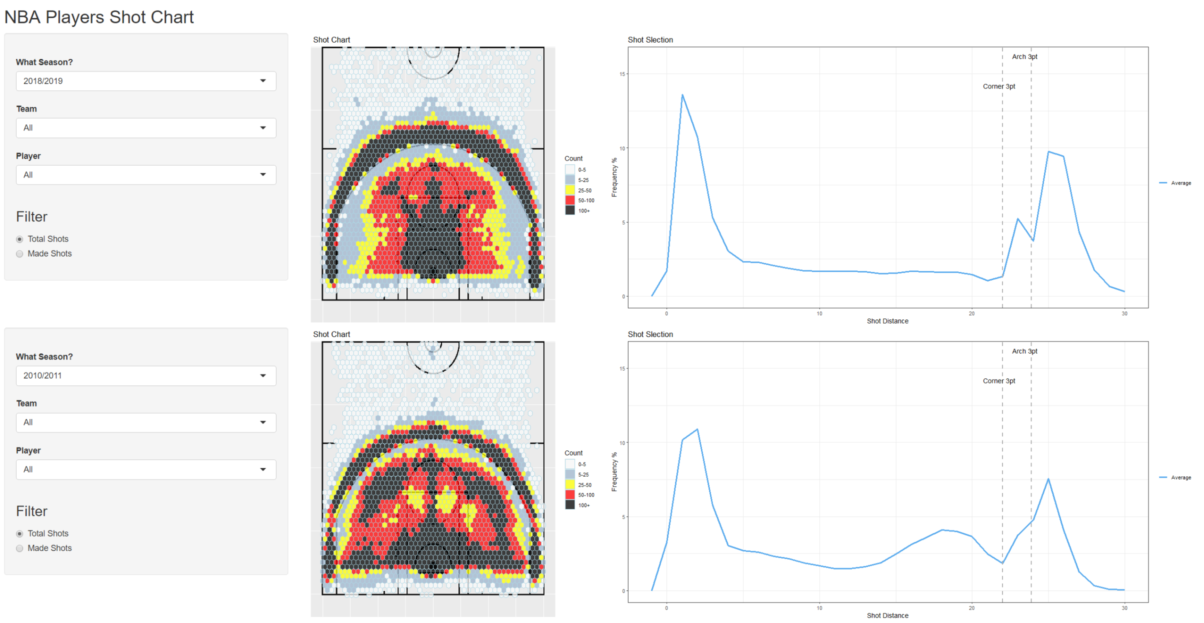
Tutorial

Building shooting charts with Big Data Ball data sets.

Intro

Let's find out how to create shooting charts from BIG DATA BALL datasets. The ultimate goal is to build an application that looks like this: https://mattiadacampo.shinyapps.io/ChartsFinal/



NBA Players Shot Chart

What Season? 2018/2019

Team All

Player All

Filter

● Total Shots

○ Made Shots

What Season? 2010/2011

Team All

Player All

Filter

● Total Shots

○ Made Shots

For now, let's focus on the shooting map. The following guide will use the 2018/19 NBA season.

Download data sets

Download your data set from the BIG DATA BALL website

https://www.bigdataball.com/datasets/nba/?ap_id=nbastuffer.

Make sure to download the Play-by-Play data (https://www.bigdataball.com/nba-pbp-singleseason/).

If you open the data set you can see there is a whole lot of information. We'll do some cleaning later to use only what we need!

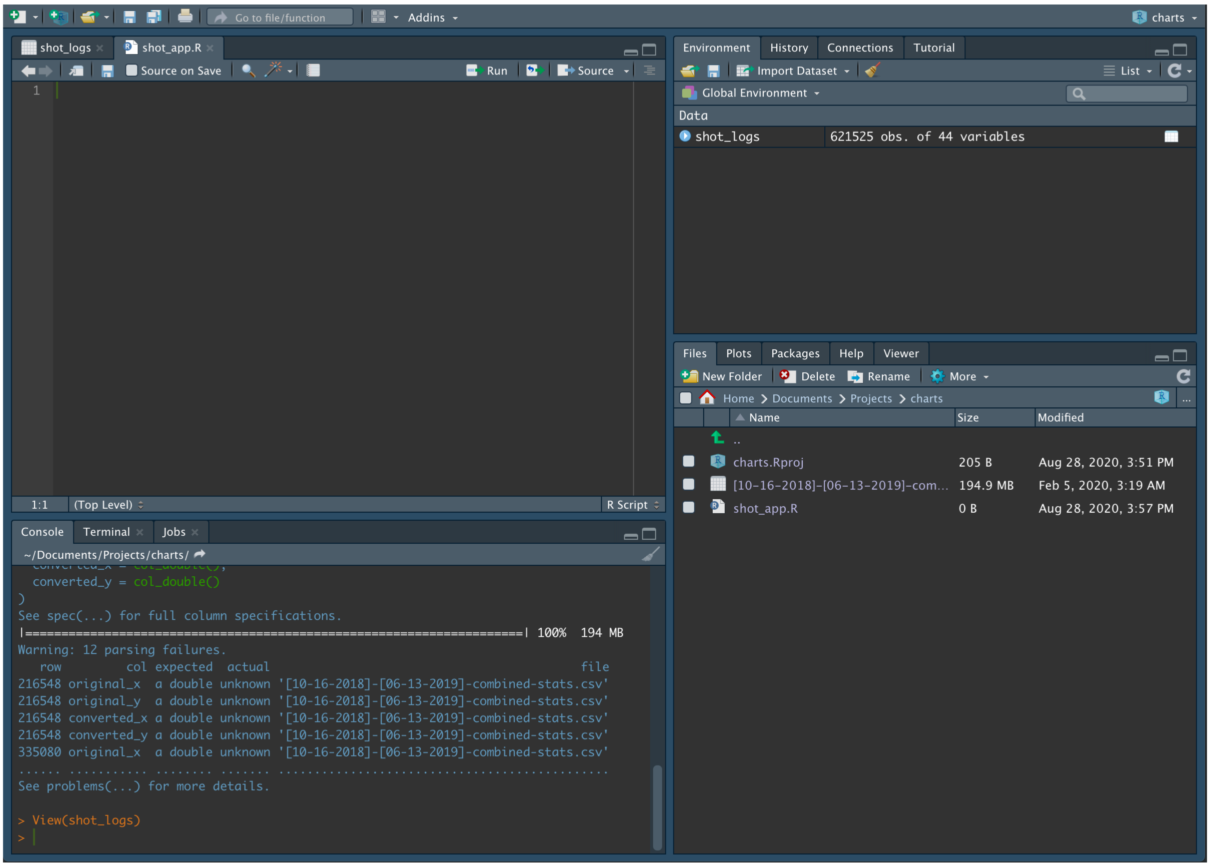
Create a new R studio project

Create a new project, we'll call it '***charts'.*** R will create a folder named *charts*. Move your data set inside that folder. Then you will see it in the *files* section of you new project.

Now click on it and import the data set into R studio. Also give it an easier name '***shot_logs***'.

Create a new R script '***shot_app***'.

You should have something that looks like this:



~/Documents/Projects/charts/

```
  converted_x = col_double(),
  converted_y = col_double()
)
See spec(...) for full column specifications.
|==================================================================| 100%  194 MB
Warning: 12 parsing failures.
   row         col expected  actual                                         file
216548 original_x  a double unknown '[10-16-2018]-[06-13-2019]-combined-stats.csv'
216548 original_y  a double unknown '[10-16-2018]-[06-13-2019]-combined-stats.csv'
216548 converted_x a double unknown '[10-16-2018]-[06-13-2019]-combined-stats.csv'
216548 converted_y a double unknown '[10-16-2018]-[06-13-2019]-combined-stats.csv'
335080 original_x  a double unknown '[10-16-2018]-[06-13-2019]-combined-stats.csv'
...... ........... ........ ....... ............................................
See problems(...) for more details.

> View(shot_logs)
>
```

Data

shot_logs	621525 obs. of 44 variables

Name	Size	Modified
..		
charts.Rproj	205 B	Aug 28, 2020, 3:51 PM
[10-16-2018]-[06-13-2019]-com...	194.9 MB	Feb 5, 2020, 3:19 AM
shot_app.R	0 B	Aug 28, 2020, 3:57 PM

Install R packages

Let's install the packages we need and call the libraries.

```
install.package('tidyverse')
library(tidyverse)
install.package('hexbin')
library('hexbin')
```

Clean Data

Let's clean our data, it will make life easier later. We need the following columns:

- Team
- Player
- Result
- converted_x
- converted_y

```
shot_logs<-  na.omit(shot_logs[,c("team","player","result","converted_x","converted_y")])
```

⚠*converted_x* and *converted_y* are full court coordinates. We need to convert them to half court only.

```
shot_logs$converted_x<-ifelse(shot_logs$converted_y > 47,50 - shot_logs$converted_x,shot_logs$converted_x)
shot_logs$converted_y <-ifelse(shot_logs$converted_y > 47,94 - shot_logs$converted_y,shot_logs$converted_y)
```

Create court design

Now that we have the right coordinates let's draw the court. The original guide was made by *Ewen Gallic* and you can find it here (https://egallic.fr/en/drawing-a-basketball-court-with-r/).

Our code is a little different, copy and paste into R.

```
# Need a function to draw circle
circle_fun <- function(center=c(0,0), diameter=1, npoints=500, start=0, end=2){
  tt <- seq(start*pi, end*pi, length.out=npoints)
  data.frame(
    x = center[1] + diameter / 2 * cos(tt),
    y = center[2] + diameter / 2 * sin(tt)
  )
}
```

```
# Need symmetry
rev_y <- function(y) 94-y
```

```
# Create data frame containing coordinates of polygons
new_coords <- function(x, y, group, descri){
  new_coords_df <- data.frame(x = x, y = y)
  new_coords_df$group <- group
  new_coords_df$side <- 1
  group <- group + 1

  # The same thing for the opposite side
  new_coords_df2 <- data.frame(x = x, y = rev_y(y))
  new_coords_df2$group <- group
  new_coords_df2$side <- 2
  group <<- group + 1

  # On reunit les donnees
  new_coords_df <- rbind(new_coords_df, new_coords_df2)
  new_coords_df$descri <- descri

  return(new_coords_df)
}
```

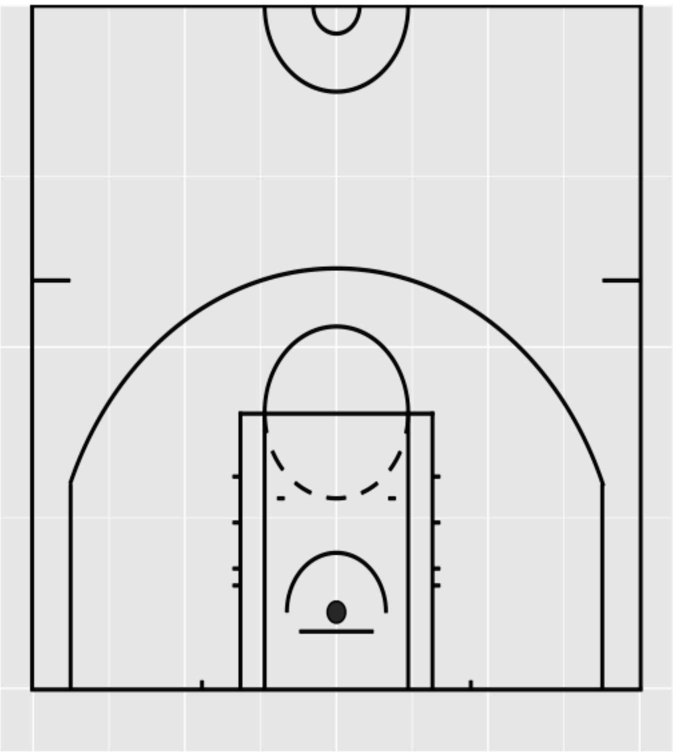
```
#Circles we need
# Restricted area
cercle_np_out <- circle_fun(center = c(25,5+3/12), diameter = (4+1/6)*2)
cercle_np_in <- circle_fun(center = c(25,5+3/12), diameter = 4*2)
# Three point
cercle_3pts_out <- circle_fun(center = c(25,5+3/12), diameter = (23+9/12)*2)
cercle_3pts_in <- circle_fun(center = c(25,5+3/12), diameter = (23+7/12)*2)
# Hoop
cercle_ce <- circle_fun(center = c(25,5+3/12), diameter = 1.5)
# Free Throws
cercle_lf_out <- circle_fun(center = c(25,19), diameter = 6*2)
cercle_lf_in <- circle_fun(center = c(25,19), diameter = (6-1/6)*2)
# Center Circle
cercle_mil_out <- circle_fun(center = c(25,47), diameter = 6*2)
cercle_mil_in <- circle_fun(center = c(25,47), diameter = (6-1/6)*2)
# Small Center Circle
cercle_mil_petit_out <- circle_fun(center = c(25,47), diameter = 2*2)
cercle_mil_petit_in <- circle_fun(center = c(25,47), diameter = (2-1/6)*2)
```

```
#We need to assign the first value of the variable group. Then, each use of new_coords increments group value by one.

group <- 1
court <- new_coords(c(0-1/6,0-1/6,50 + 1/6,50 + 1/6), c(0 - 1/6,0,0,0 - 1/6), group = group, descri = "ligne de fond")
court <- rbind(court, new_coords(x = c(0-1/6,0-1/6,0,0), y = c(0,47-1/12,47-1/12,0), group = group, descri = "ligne gauche"))
court <- rbind(court, new_coords(x = c(50,50,50+1/6,50+1/6), y = c(0,47-1/12,47-1/12,0), group = group, descri = "ligne droite"
court <- rbind(court, new_coords(x = c(0,0,3,3), y = c(28,28+1/6,28+1/6,28), group = group, descri = "marque entraineur gauche"
court <- rbind(court, new_coords(x = c(47,47,50,50), y = c(28,28+1/6,28+1/6,28), group = group, descri = "marque entraineur dro
court <- rbind(court, new_coords(x = c(3,3,3+1/6,3+1/6), y = c(0,14,14,0), group = group, descri = "3pts bas gauche"))
court <- rbind(court, new_coords(x = c(47-1/6,47-1/6,47,47), y = c(0,14,14,0), group = group, descri = "3pts bas droit"))
court <- rbind(court, new_coords(x = c(17,17,17+1/6,17+1/6), y = c(0,19,19,0), group = group, descri = "LF bas gauche"))
court <- rbind(court, new_coords(x = c(33-1/6,33-1/6,33,33), y = c(0,19,19,0), group = group, descri = "LF bas droit"))
court <- rbind(court, new_coords(x = c(17,17,33,33), y = c(19-1/6,19,19,19-1/6), group = group, descri = "LF tireur"))
court <- rbind(court, new_coords(x = c(14-1/6,14-1/6,14,14), y = c(0,1/2,1/2,0), group = group, descri = "marque fond gauche"))
court <- rbind(court, new_coords(x = c(36,36,36+1/6,36+1/6), y = c(0,1/2,1/2,0), group = group, descri = "marque fond droit"))
court <- rbind(court, new_coords(x = c(19,19,19+1/6,19+1/6), y = c(0,19,19,0), group = group, descri = "LF gauche interieur"))
court <- rbind(court, new_coords(x = c(31-1/6,31-1/6,31,31), y = c(0,19,19,0), group = group, descri = "LF droite interieur"))
court <- rbind(court, new_coords(x = c(22, 22, 28, 28), y = c(4-1/6,4,4,4-1/6), group = group, descri = "planche"))
court <- rbind(court, new_coords(x = c(cercle_3pts_out[31:220,"x"], rev(cercle_3pts_in[31:220,"x"])),
                                 y = c(cercle_3pts_out[31:220,"y"], rev(cercle_3pts_in[31:220,"y"])), group = group, descri = "
court <- rbind(court, new_coords(x = c(cercle_np_out[1:250,"x"], rev(cercle_np_in[1:250,"x"])),
                                 y = c(cercle_np_out[1:250,"y"], rev(cercle_np_in[1:250,"y"])), group = group, descri = "cercle
court <- rbind(court, new_coords(x = c(20+1/6,20+1/6,20+8/12,20+8/12), y = c(13,13+1/6,13+1/6,13), group = group, descri = "mar
court <- rbind(court, new_coords(x = c(30-8/12,30-8/12,30-1/6,30-1/6), y = c(13,13+1/6,13+1/6,13), group = group, descri = "mar
court <- rbind(court, new_coords(x = c(cercle_lf_out[1:250,"x"], rev(cercle_lf_in[1:250,"x"])),
                                 y = c(cercle_lf_out[1:250,"y"], rev(cercle_lf_in[1:250,"y"])), group = group, descri = "cercle
court <- rbind(court, new_coords(x = c(cercle_lf_out[250:269,"x"], rev(cercle_lf_in[250:269,"x"])),
                                 y = c(cercle_lf_out[250:269,"y"], rev(cercle_lf_in[250:269,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(cercle_lf_out[288:308,"x"], rev(cercle_lf_in[288:308,"x"])),
                                 y = c(cercle_lf_out[288:308,"y"], rev(cercle_lf_in[288:308,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(cercle_lf_out[327:346,"x"], rev(cercle_lf_in[327:346,"x"])),
                                 y = c(cercle_lf_out[327:346,"y"], rev(cercle_lf_in[327:346,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(cercle_lf_out[365:385,"x"], rev(cercle_lf_in[365:385,"x"])),
                                 y = c(cercle_lf_out[365:385,"y"], rev(cercle_lf_in[365:385,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(cercle_lf_out[404:423,"x"], rev(cercle_lf_in[404:423,"x"])),
                                 y = c(cercle_lf_out[404:423,"y"], rev(cercle_lf_in[404:423,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(cercle_lf_out[442:462,"x"], rev(cercle_lf_in[442:462,"x"])),
                                 y = c(cercle_lf_out[442:462,"y"], rev(cercle_lf_in[442:462,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(cercle_lf_out[481:500,"x"], rev(cercle_lf_in[481:500,"x"])),
                                 y = c(cercle_lf_out[481:500,"y"], rev(cercle_lf_in[481:500,"y"])), group = group, descri = "ce
court <- rbind(court, new_coords(x = c(17-0.5,17-0.5,17,17), y = c(7,7+1/6,7+1/6,7), group = group, descri = "marque 1 LF gauch
court <- rbind(court, new_coords(x = c(17-0.5,17-0.5,17,17), y = c(8+1/6,8+1/3,8+1/3,8+1/6), group = group, descri = "marque 2
court <- rbind(court, new_coords(x = c(17-0.5,17-0.5,17,17), y = c(11+1/3,11.5,11.5,11+1/3), group = group, descri = "marque 3
court <- rbind(court, new_coords(x = c(17-0.5,17-0.5,17,17), y = c(14.5,14.5+1/6,14.5+1/6,14.5), group = group, descri = "marqu
court <- rbind(court, new_coords(x = c(33,33,33+0.5,33+0.5), y = c(7,7+1/6,7+1/6,7), group = group, descri = "marque 1 LF droit
court <- rbind(court, new_coords(x = c(33,33,33+0.5,33+0.5), y = c(8+1/6,8+1/3,8+1/3,8+1/6), group = group, descri = "marque 2
court <- rbind(court, new_coords(x = c(33,33,33+0.5,33+0.5), y = c(11+1/3,11.5,11.5,11+1/3), group = group, descri = "marque 3
court <- rbind(court, new_coords(x = c(33,33,33+0.5,33+0.5), y = c(14.5,14.5+1/6,14.5+1/6,14.5), group = group, descri = "marqu
court <- rbind(court, new_coords(x = c(0-1/6,0-1/6,50+1/6,50+1/6), y = c(94/2-1/12,94/2, 94/2, 94/2-1/12), group = group, descr
court <- rbind(court, new_coords(x = c(cercle_mil_out[250:500,"x"], rev(cercle_mil_in[250:500,"x"])),
                                 y = c(cercle_mil_out[250:500,"y"], rev(cercle_mil_in[250:500,"y"])), group = group, descri = "
court <- rbind(court, new_coords(x = c(cercle_mil_petit_out[250:500,"x"], rev(cercle_mil_petit_in[250:500,"x"])),
                                 y = c(cercle_mil_petit_out[250:500,"y"], rev(cercle_mil_petit_in[250:500,"y"])), group = group
court <- rbind(court, new_coords(x = cercle_ce[,"x"], y = cercle_ce[,"y"], group = group, descri = "anneau"))
```

```
#Create the graph

P <- ggplot() + geom_polygon(data = court, aes(x = x, y = y, group = group), col = "black") +
  coord_equal() +
  ylim(-2,96) +
  xlim(-5,55) +
  scale_x_continuous(breaks = c(0, 12.5, 25, 37.5, 50)) +
  scale_y_continuous(breaks = c(0, 23.5, 47, 70.5, 94)) +
  xlab("") + ylab("") +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(), axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank(), axis.title = element_blank()
  )

P
```

```
halfP <- P + coord_cartesian(ylim = c(-2, 44.7))

halfP
```

Should get something like this



Create function

Right now we could plot the shooting charts of:

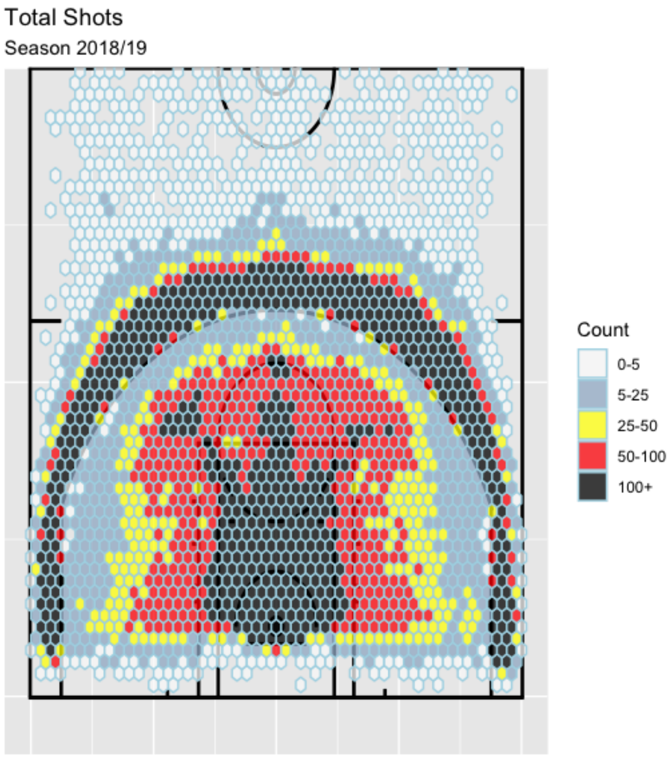
1. the entire season
2. a specific team
3. a specific player

The *geom_hex* function lets us divide the court into small bins, and count how many shot were attempted inside that bin. A few thing you should know:

- *binwidth* - choose how big you want your bins
- *alpha* - transparency, used so we can still see the court behind the bins
- *count* - select the count intervals you want to display
- *scale_fill_manual* - decide what colors you want to display

Plot the entire season.

```
halfP + geom_hex(data = shot_logs,
                           aes(x =converted_x ,
                               y =converted_y,
                               fill = cut(..count.., c(
                                 0,5,25, 50, 100, Inf))),
                           colour = "lightblue",
                           binwidth = 1,
                           alpha = 0.75) +
    scale_fill_manual(values = c("grey98", "slategray3", "yellow", "red" , "black"),
                      labels = c("0-5","5-25","25-50","50-100","100+"), name = "Count")+
  labs(title = 'Total Shots',
       subtitle = 'Season 2018/19')
```



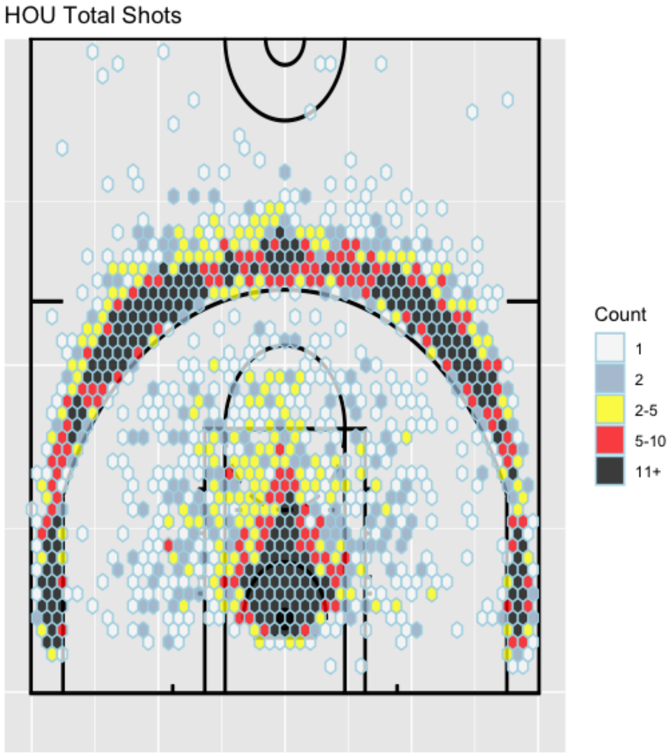
Plot a specific team

First, let's build a function that will let us choose any team we want.

```
generate_team_chart <- function(team_name) {

  plot<-  halfP + geom_hex(data = subset(shot_logs, team == team_name),
                           aes(x =converted_x ,
                               y =converted_y,
                               fill = cut(..count.., c(0,1,2, 5, 10, Inf))),
                           colour = "lightblue",
                           binwidth = 1,
                           alpha = 0.75) +
    scale_fill_manual(values = c("grey98", "slategray3", "yellow", "red" , "black"),
                      labels = c("1","2","2-5","5-10","11+"), name = "Count")+
    ggtitle(paste(team_name,"Total Shots"))
  return(plot)
}
```

Then let's plot the Houston Rockets shooting map. Remember to use '*HOU*'. You can find the other teams abbreviations in the data set.

```
generate_team_chart('HOU')
```



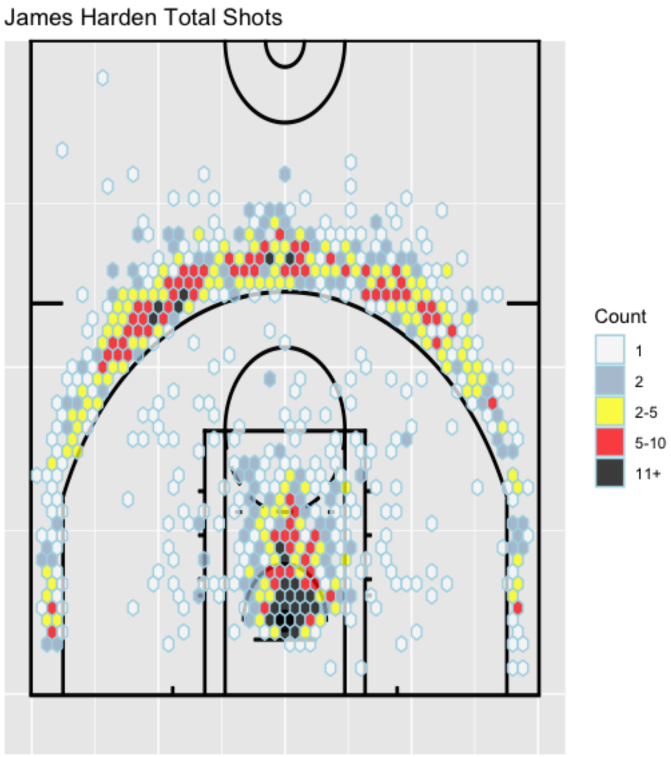
Plot a specific player

As before, let's build the function.

```
generate_player_chart <- function(name) {

  plot<-  halfP + geom_hex(data = subset(shot_logs, player == name),
                           aes(x =converted_x ,
                               y =converted_y,
                               fill = cut(..count.., c(0,1,2, 5, 10, Inf))),
                               colour = "lightblue",
                               binwidth = 1,
                               alpha = 0.75) +
    scale_fill_manual(values = c("grey98", "slategray3", "yellow", "red" , "black"),
                      labels = c("1","2","2-5","5-10","11+"), name = "Count")+
    ggtitle(paste(name,"Total Shots"))
  return(plot)
}
```

And then plot James Harden shots.

```
generate_player_chart('James Harden')
```



Want to plot only a specific team or player makes? Try adding the filter inside the function. For example:

```
generate_player_makes <- function(name) {

  plot<-  halfP + geom_hex(data = subset(shot_logs, player == name & result =='made'),
                          aes(x =converted_x ,
                              y =converted_y,
                              fill = cut(..count.., c(0,1,2, 5, 10, Inf))),
                              colour = "lightblue",
                              binwidth = 1,
                              alpha = 0.75) +
    scale_fill_manual(values = c("grey98", "slategray3", "yellow", "red" , "black"),
                     labels = c("1","2","2-5","5-10","11+"), name = "Count")+
    ggtitle(paste(name,"Total Shots"))
  return(plot)
}

generate_player_makes('James Harden')
```

Conclusion

I invite the reader to use this guide as inspiration. In addition to shooting maps, you can also create some density curves like the ones on my application!

https://mattiadacampo.shinyapps.io/ChartsFinal/

Also, if you have any questions feel free to contact me or visit my website at www.mattianalytics.com.