post01_Anna_Lu.Rmd

Anna_Lu

October 30, 2017

Data visualization in R (3 variables)

1. Introduction

In the post, I would like to talk about some useful function of ggplot2 outside of class. During our lecture, professor mainly talked about the graph of 2 variables (continous x and continous y). Here, I want to intruduce some new functions of the graph of 3 variables. **geom_raster(), and geom_tile()**.

2. Background

In data analysis, data visualization is one of the most important steps. ggplot2 is the most popular data visualization package of R language, created by Chief Scientist at RStudio, **Hadley Wickham**. ggplot2 is a new package in R, compared to other graph tools. Before, ggplot2, there are graphics and grid package. Therefore, why we still need ggplot2 ? There are some short introduction of ggplot2. The main idea of ggplot2 is to separate the drawing part and data part. Also, ggplot2 is based on the layer to draw the graph. “ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts.”(http://ggplot2.org).

3. Installation

First, we need to install the package.

There is two way to get ggplot2.

```
##1. install the whole tidyverse:
install.packages('tidyverse')
library(tidyverse)
```

```
##2.install ggplot2(here we use second way)
#install.packages('ggplot2')
library(ggplot2)
```

4. tidy up data

Tidying up data is very important. Everytime we get a new data, it is hard to be 100% what we want. Therefore, we need to select what we need before we use the data.

```
#importing data
dat <- data.frame(
  read.csv("/Users/anna/Desktop/stat133/stat133-hws-fall17/Post 1/data/2015-2016_nbasalariespoints.csv"),
  stringsAsFactors = FALSE
)
str(dat)
```

```
## 'data.frame':    517 obs. of  35 variables:
##  $ X         : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Rk        : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ Player    : Factor w/ 517 levels "Aaron Brooks",..: 455 202 287 117 316 99 30 116 431 396 ...
##  $ Pos       : Factor w/ 8 levels "","C","PF","PF-C",..: 5 7 6 2 6 5 3 7 5 6 ...
##  $ Age       : num  27 26 27 25 31 25 22 26 27 25 ...
##  $ Tm        : Factor w/ 32 levels "","ATL","BOS",..: 11 12 22 27 7 26 20 29 22 13 ...
##  $ G         : num  79 82 72 65 76 75 61 78 80 81 ...
##  $ GS        : num  79 82 72 65 76 75 61 78 80 81 ...
##  $ MP        : num  34.2 38.1 35.8 34.6 35.6 35.7 35.5 35.9 34.4 34.8 ...
##  $ FG        : num  10.2 8.7 9.7 9.2 9.7 8.2 9.2 7.9 8.2 7.5 ...
##  $ FGA       : num  20.2 19.7 19.2 20.5 18.6 19.7 18.6 17.7 18.1 17.9 ...
##  $ FG.       : num  0.504 0.439 0.505 0.451 0.52 0.419 0.493 0.446 0.454 0.418 ...
##  $ X3P       : num  5.1 2.9 2.6 1.1 1.1 3.1 0.6 0.6 1.3 2.6 ...
##  $ X3PA      : num  11.2 8 6.7 3.2 3.7 8.1 1.8 1.8 4.3 7 ...
##  $ X3P.      : num  0.454 0.359 0.387 0.333 0.309 0.375 0.324 0.338 0.296 0.371 ...
##  $ X2P       : num  5.1 5.8 7.1 8.2 8.6 5.2 8.6 7.3 6.9 4.9 ...
##  $ X2PA      : num  9 11.7 12.5 17.3 14.9 11.5 16.9 15.9 13.8 10.9 ...
##  $ X2P.      : num  0.566 0.494 0.569 0.473 0.573 0.45 0.511 0.458 0.503 0.447 ...
##  $ eFG.      : num  0.63 0.512 0.573 0.477 0.551 0.497 0.508 0.463 0.489 0.49 ...
##  $ FT        : num  4.6 8.8 6.2 7.3 4.7 5.5 5.3 7.1 5.8 5.6 ...
##  $ FTA       : num  5.1 10.2 6.9 10.2 6.5 6.2 7 8.4 7.2 6.5 ...
##  $ FT.       : num  0.908 0.86 0.898 0.718 0.731 0.892 0.758 0.85 0.812 0.86 ...
##  $ ORB       : num  0.9 0.8 0.6 2.4 1.5 0.6 2.1 0.8 1.8 1 ...
##  $ DRB       : num  4.6 5.3 7.6 9.1 6 3.4 8.1 3.7 6 6 ...
##  $ TRB       : num  5.4 6.1 8.2 11.5 7.4 4 10.3 4.5 7.8 7 ...
##  $ AST       : num  6.7 7.5 5 3.3 6.8 6.8 1.9 4 10.4 4.1 ...
##  $ STL       : num  2.1 1.7 1 1.6 1.4 0.9 1.3 1 2 1.9 ...
##  $ BLK       : num  0.2 0.6 1.2 1.4 0.6 0.4 2 0.3 0.3 0.4 ...
##  $ TOV       : num  3.3 4.6 3.5 3.8 3.3 3.2 2 2.2 4.3 3.3 ...
##  $ PF        : num  2 2.8 1.9 3.6 1.9 2.2 2.4 2.1 2.5 2.8 ...
##  $ PS.G.     : num  30.1 29 28.2 26.9 25.3 25.1 24.3 23.5 23.5 23.1 ...
##  $ Unnamed..0: num  53 23 6 22 1 154 94 65 15 14 ...
##  $ RK        : num  54 24 7 23 2 155 95 66 16 15 ...
##  $ TEAM      : Factor w/ 32 levels "","Atlanta Hawks",..: 12 13 23 28 8 27 21 30 23 14 ...
##  $ SALARY    : Factor w/ 326 levels "","$1,000,000 ",..: 82 110 173 111 181 233 290 71 115 118 ...
```

```
#select useful column
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
dat1 <- select(dat, Player, SALARY, Pos, Age, Tm, G, GS, MP, FGA, FTA, X3PA, X2PA)
str(dat1)
```

```
## 'data.frame':    517 obs. of  12 variables:
##  $ Player: Factor w/ 517 levels "Aaron Brooks",..: 455 202 287 117 316 99 30 116 431 396 ...
##  $ SALARY: Factor w/ 326 levels "","$1,000,000 ",..: 82 110 173 111 181 233 290 71 115 118 ...
##  $ Pos   : Factor w/ 8 levels "","C","PF","PF-C",..: 5 7 6 2 6 5 3 7 5 6 ...
##  $ Age   : num  27 26 27 25 31 25 22 26 27 25 ...
##  $ Tm    : Factor w/ 32 levels "","ATL","BOS",..: 11 12 22 27 7 26 20 29 22 13 ...
##  $ G     : num  79 82 72 65 76 75 61 78 80 81 ...
##  $ GS    : num  79 82 72 65 76 75 61 78 80 81 ...
##  $ MP    : num  34.2 38.1 35.8 34.6 35.6 35.7 35.5 35.9 34.4 34.8 ...
##  $ FGA   : num  20.2 19.7 19.2 20.5 18.6 19.7 18.6 17.7 18.1 17.9 ...
##  $ FTA   : num  5.1 10.2 6.9 10.2 6.5 6.2 7 8.4 7.2 6.5 ...
##  $ X3PA  : num  11.2 8 6.7 3.2 3.7 8.1 1.8 1.8 4.3 7 ...
##  $ X2PA  : num  9 11.7 12.5 17.3 14.9 11.5 16.9 15.9 13.8 10.9 ...
```

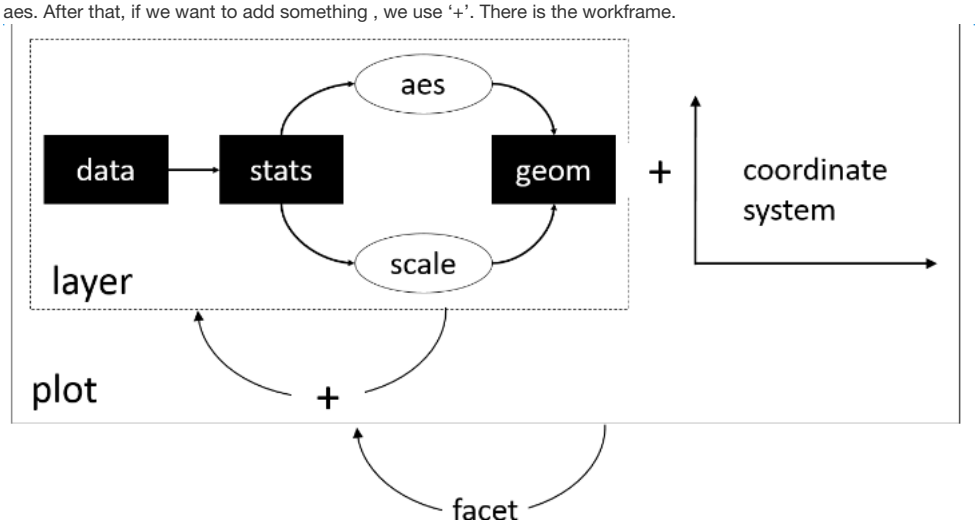
```
#filter useful rows
dat1 <- dat1[1:476, ]
str(dat1)
```

```
## 'data.frame':    476 obs. of  12 variables:
##  $ Player: Factor w/ 517 levels "Aaron Brooks",..: 455 202 287 117 316 99 30 116 431 396 ...
##  $ SALARY: Factor w/ 326 levels "","$1,000,000 ",..: 82 110 173 111 181 233 290 71 115 118 ...
##  $ Pos   : Factor w/ 8 levels "","C","PF","PF-C",..: 5 7 6 2 6 5 3 7 5 6 ...
##  $ Age   : num  27 26 27 25 31 25 22 26 27 25 ...
##  $ Tm    : Factor w/ 32 levels "","ATL","BOS",..: 11 12 22 27 7 26 20 29 22 13 ...
##  $ G     : num  79 82 72 65 76 75 61 78 80 81 ...
##  $ GS    : num  79 82 72 65 76 75 61 78 80 81 ...
##  $ MP    : num  34.2 38.1 35.8 34.6 35.6 35.7 35.5 35.9 34.4 34.8 ...
##  $ FGA   : num  20.2 19.7 19.2 20.5 18.6 19.7 18.6 17.7 18.1 17.9 ...
##  $ FTA   : num  5.1 10.2 6.9 10.2 6.5 6.2 7 8.4 7.2 6.5 ...
##  $ X3PA  : num  11.2 8 6.7 3.2 3.7 8.1 1.8 1.8 4.3 7 ...
##  $ X2PA  : num  9 11.7 12.5 17.3 14.9 11.5 16.9 15.9 13.8 10.9 ...
```

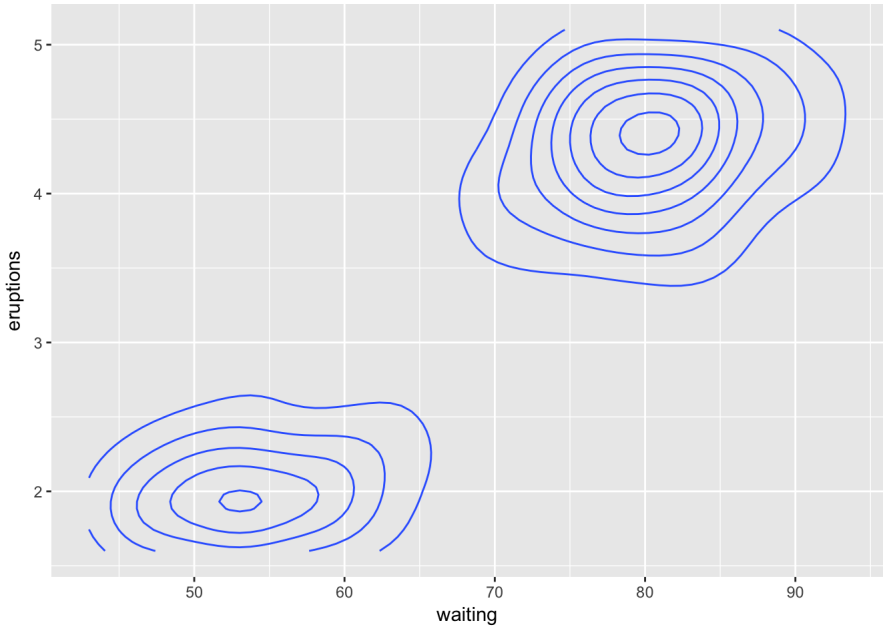
```
#export the new table to a csv file named clean_2015-2016 nbasalariespoints.csv, inside the data/ folder.
write.csv(dat1, file = 'clean_2015-2016_nbasalariespoints.csv')
```

5. Usage

In short, ggplot2 uses data, stats and geom. For example, we have a sample data, we need to change data to stats, and then choose how to represent the data, like bar or line. Choosing the way to represent data is geom. In geom, we need to defind X, Y, Z, color and so on, which is aes. After that, if we want to add something , we use '+'. There is the workframe.



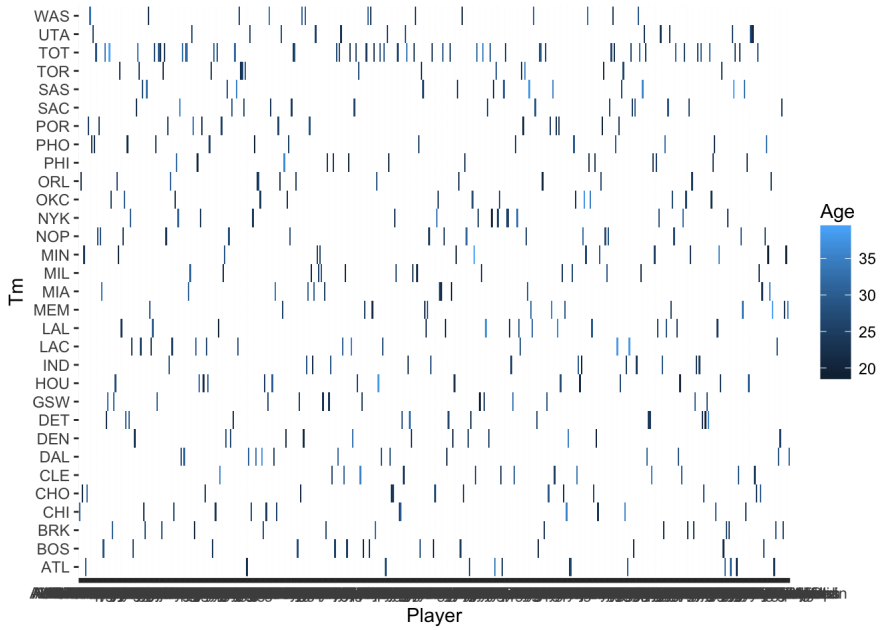

```
##The base plot (http://ggplot2.tidyverse.org):
#first step
 v <- ggplot(faithfuld, aes(waiting, eruptions, z = density))

#choose how to represent the data
 v + geom_contour()
```



6. geom_raster()

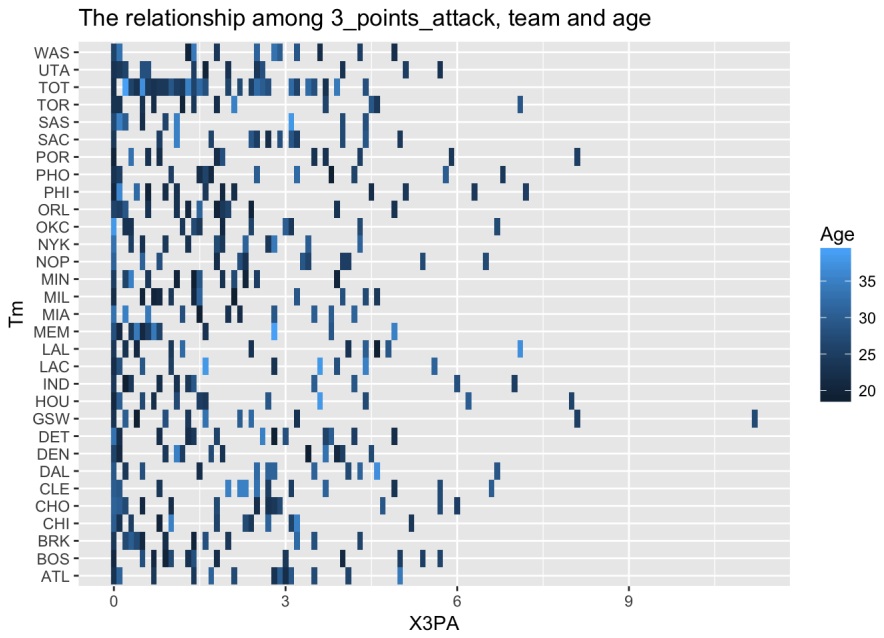
geom_raster() uses the intensity of cokor to show the relationship between two or among more variables in a two-dimensional image. Sometimes we need to see the different kind of ordering of a matrix. geom_raster() is the good tool to make it vizualise.

```
ggplot(dat1, aes(x = Player, y = Tm)) +
  geom_raster(aes(fill = Age))
```



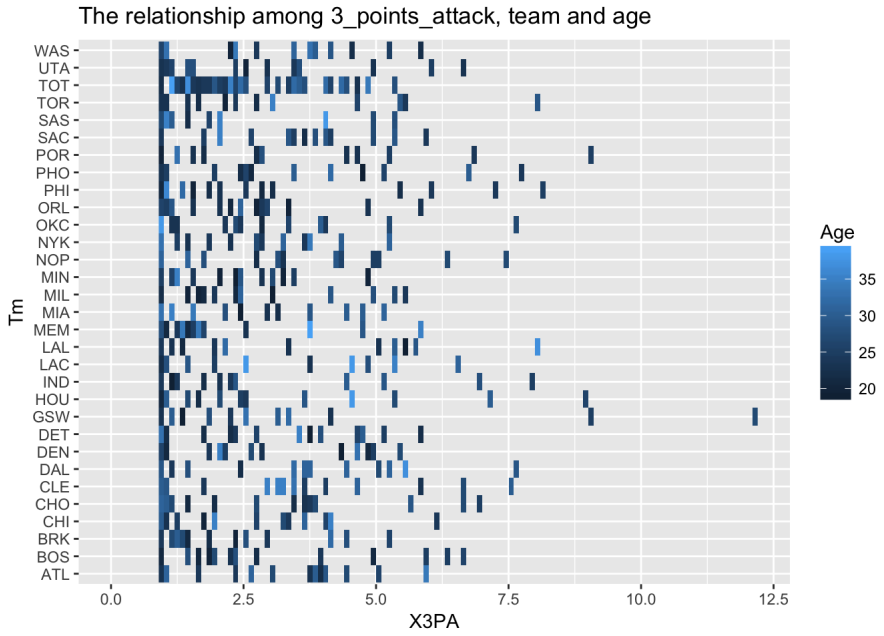
use ggtitle() to add the title if we want.

```
#if we want to add a title
ggplot(dat1, aes(x = X3PA, y = Tm)) +
  geom_raster(aes(fill = Age)) +
  ggtitle('The relationship among 3_points_attack, team and age')
```



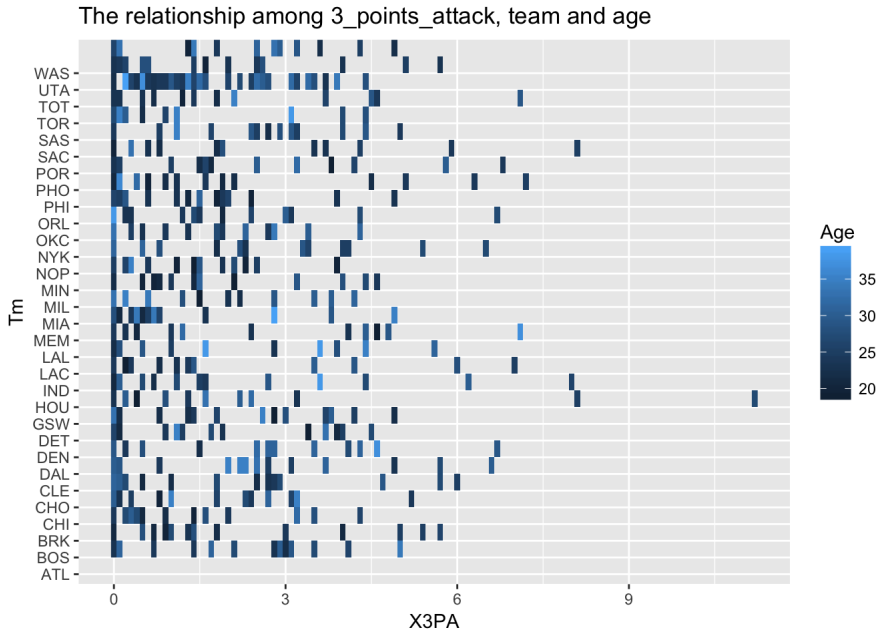
hjust() can push right the graph.

```
#if we want to go right
ggplot(dat1, aes(x = X3PA, y = Tm)) +
  geom_raster(aes(fill = Age), hjust = 10) +
  ggtitle('The relationship among 3_points_attack, team and age')
```



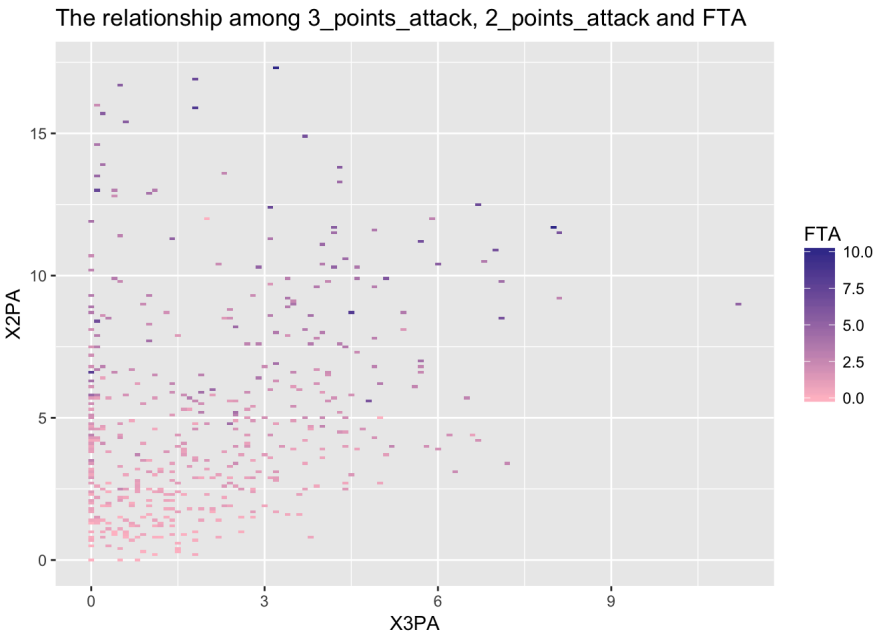
vjust() can push up the graph.

```
#if we want to push up the graph
ggplot(dat1, aes(x = X3PA, y = Tm)) +
  geom_raster(aes(fill = Age), vjust = 2) +
  ggtitle('The relationship among 3_points_attack, team and age')
```



scale_fill_gradient2() can change color of the graph.

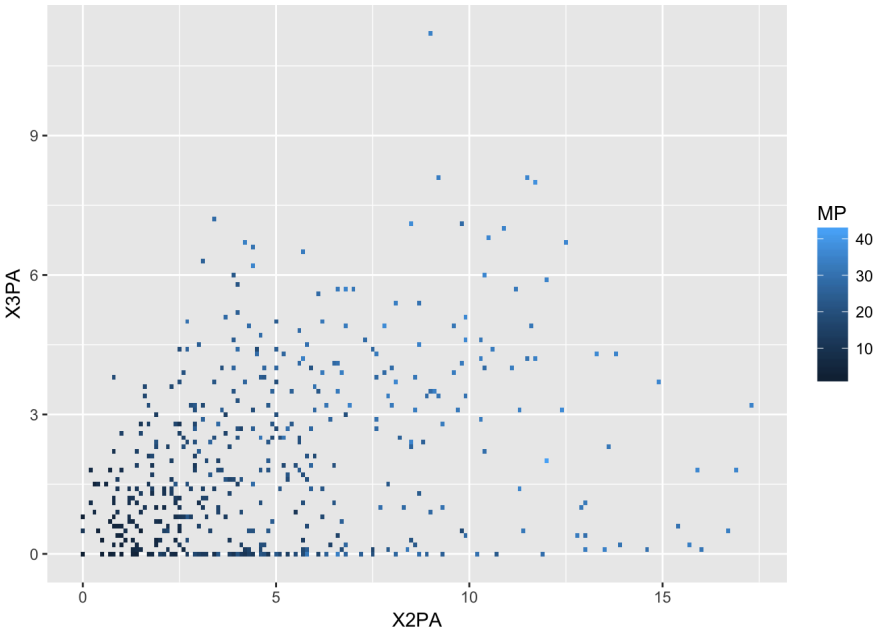
```
#it can be more colorful
ggplot(dat1) +
  geom_raster(aes(x = X3PA, y = X2PA, fill = FTA)) +
  scale_fill_gradient2(mid = 'pink') +
  ggtitle('The relationship among 3_points_attack, 2_points_attack and FTA')
```



7. geom_tile

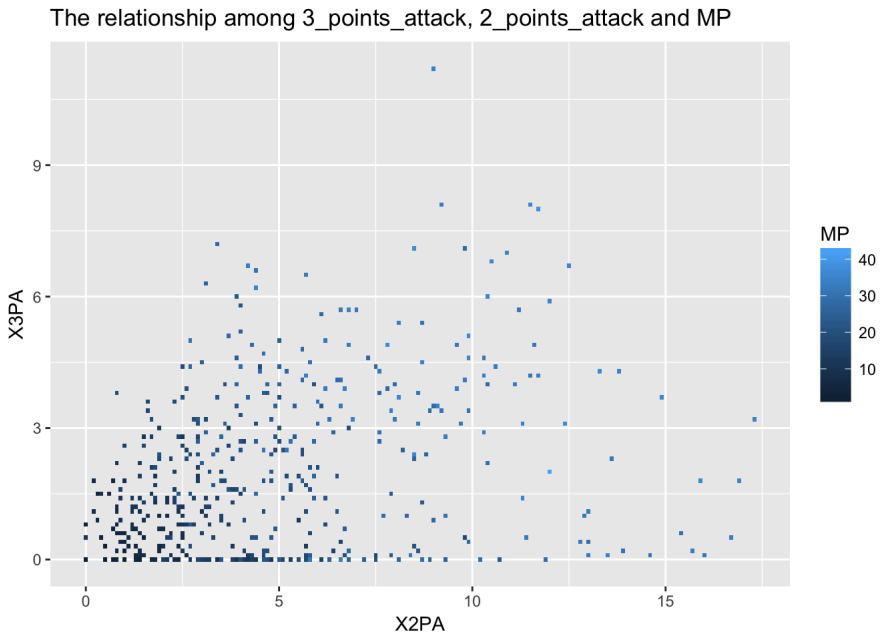
geom_tile is used for 3 variables, which is same as geom_raster. geom_tile also uses color to show the relationship among the data, and geom_tile uses the center of the tile.

```
ggplot(dat1, aes(x = X2PA, y = X3PA, fill = MP)) +
  geom_tile()
```



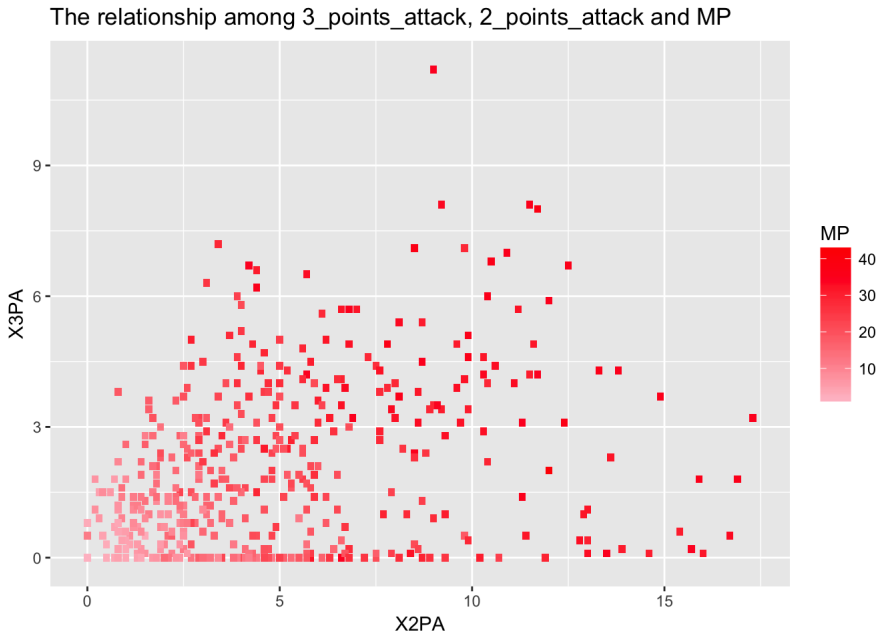
also, the ggtitle().

```
#give a title to the graph
ggplot(dat1, aes(x = X2PA, y = X3PA, fill = MP)) +
  geom_tile() +
  ggtitle('The relationship among 3_points_attack, 2_points_attack and MP')
```



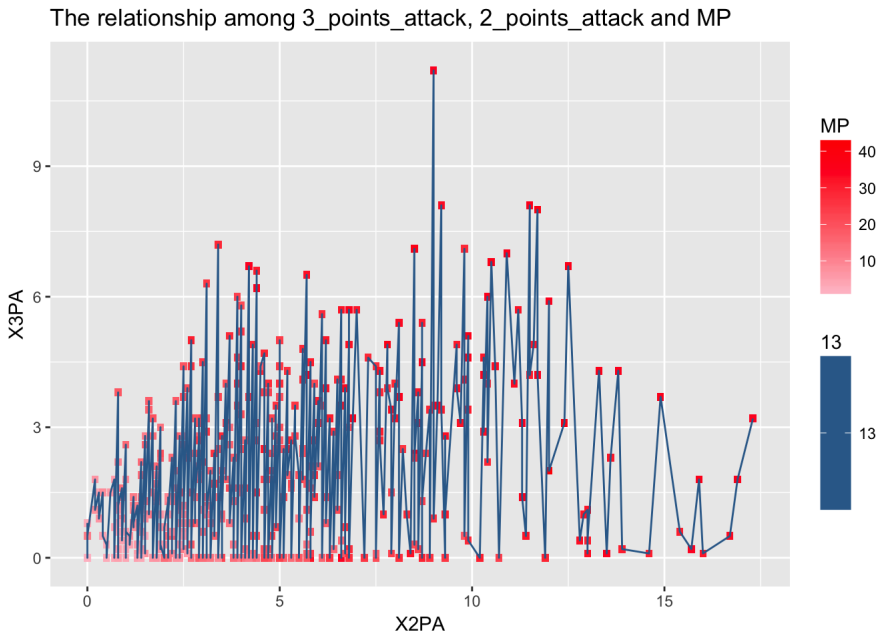
for the geom_tile(), we can use scale_fill_gradientn() to change color.

```
#change the color
ggplot(dat1, aes(x = X2PA, y = X3PA, fill = MP), size = 100) +
  geom_tile(width = .18, height = .18) +
  ggtitle('The relationship among 3_points_attack, 2_points_attack and MP') +
  scale_fill_gradientn(colors = colorRampPalette(c('pink', 'red'))(400))
```



if we want to add more graph, we can use other functions.

```
#combined with other function
ggplot(dat1, aes(x = X2PA, y = X3PA, fill = MP), size = 100) +
  geom_tile(width = .18, height = .18) +
  ggtitle('The relationship among 3_points_attack, 2_points_attack and MP') +
  scale_fill_gradientn(colors = colorRampPalette(c('pink', 'red'))(400)) +
  geom_line(aes(x = X2PA, y = X3PA, colour = 13))
```



8. more function

There is the cheetsheet of ggplot2. We can find more function here. (https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf)

9. take home message

The post is mainly about how to use geom_tile, and geom_raster to draw the 3 varables graph. The most important thing is to remember the usage of geom_tile, and geom_raster. If we still want to change the color or add title, just use the correct function, which can find from the ggplot2 cheetsheet.

Reference:

1. http://ggplot2.org
2. http://ggplot2.tidyverse.org
3. https://segmentfault.com/a/1190000006120665#articleHeader1
4. http://ggplot2.tidyverse.org/reference/geom_tile.html
5. https://learnr.wordpress.com/2010/01/26/ggplot2-quick-heatmap-plotting/
6. https://stackoverflow.com/questions/15655710/how-to-adjust-the-tile-height-in-geom-tile
7. https://www.youtube.com/watch?v=zSSNWZuVG8Y
8. https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf