

# Post01: More about Data Visualisation with ggplot2

Chloe Lin

2017/10/28

## Introduction

David McCandless in his TED talk *The Beauty of Data Visualization* compared graph to “information map” that clears viewers’ eyes when they are confused by overwhelming delivery of raw data. From this perspective, ggplot2 in R is an extremely helpful package that could efficiently and nicely present pictures we want for a series of data analysis. So far in this class, we have tried out some basic usage of ggplot2, such as the aesthetic design, geometric objects etc. In this post, I am going to introduce how functions in this package related to **scales** and **themes** can control the visual display of those plots.

## Preperation

### Library

Before we dig into the knowledge of ggplot2, we first need to load the package.

```
library(ggplot2)
```

### Data

I will also take the NBA data that we are familiar with as our example, and use the built-in function `read.csv()` to deliver the csv data.

```
dat <- read.csv('nba2017-players.csv', stringsAsFactors = FALSE)
```

With these two steps completed, we are ready to create more customized graphs in following explanations!

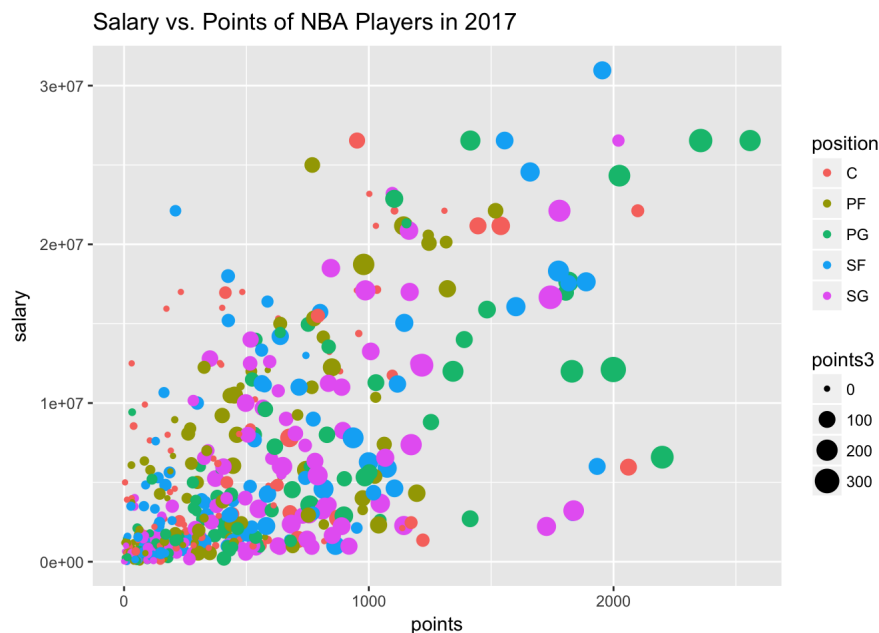
## Scales

### What is Scale?

The ggplot functions we have already known are really convenient and intuitive to use, since when we inform ggplot2 that we want different shapes or sizes according to specific subgroups of a data frame, we usually accept the default design that the plot finally renders. We do not assign which exact shape or size that we want to use.

For example, in this scatterplot of salary and points made for each player in NBA below, the function itself automatically chooses the value of color and size for each of my indicated variables—position and three-pointers made.

```
ggplot(data = dat, aes(x = points, y = salary)) +  
  geom_point(aes(color = position, size = points3)) + ggtitle('Salary vs. Points of NBA Players in 2017')
```



## Scales Functions

Scales functions in the form `scale_<aesthetic>_<type>` allow us to modify those aesthetic values and acquire whatever the visual effects we like. There are basically five categories that we can set scales about:

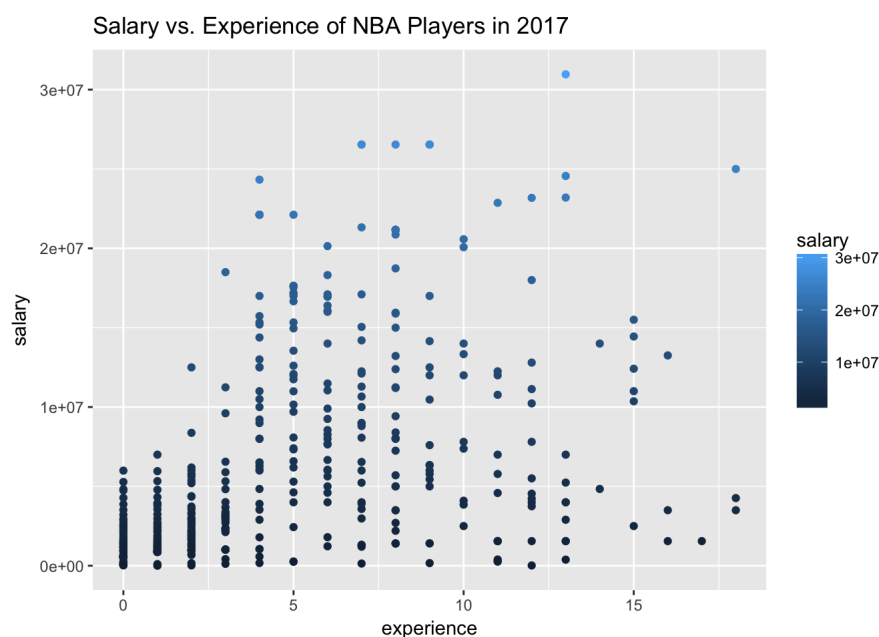
- position
- color and fill
- size
- shape
- line type

Now I am going to present some examples of modifying the scales, and you can also notice the common scale arguments such as **name**, **limits**, **breaks** and **labels** that are usually involved with those functions.

## Examples

Here we have an initial scatterplot showing the relationship of experience and salary for NBA players:

```
ggplot(data = dat, aes(x = experience, y = salary)) + geom_point(aes(color = salary)) + ggtitle('Salary vs. Experience of NBA Players in 2017')
```

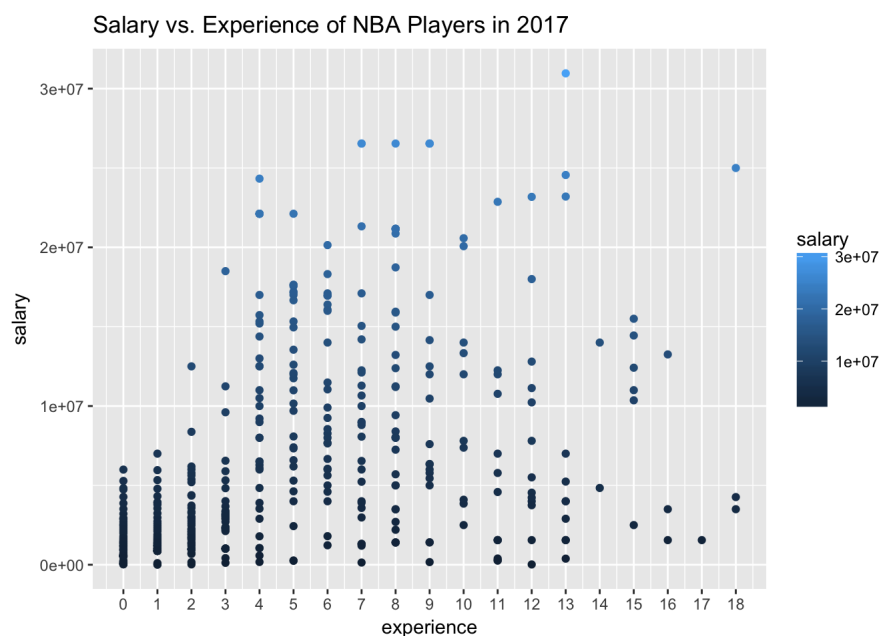


### Position

Regarding to position, if we want to modify the scale of x axis, we can write functions such as `scale_x_continuous`.

In this example, in order to list all the values of experience that NBA players have from 0 to 18, I assign a vector to the argument **breaks** so that every number is displayed:

```
ggplot(data = dat, aes(x = experience, y = salary)) + ggtitle('Salary vs. Experience of NBA Players in 2017') + geom_point(aes(color = salary)) + scale_x_continuous(breaks = c(0:18))
```

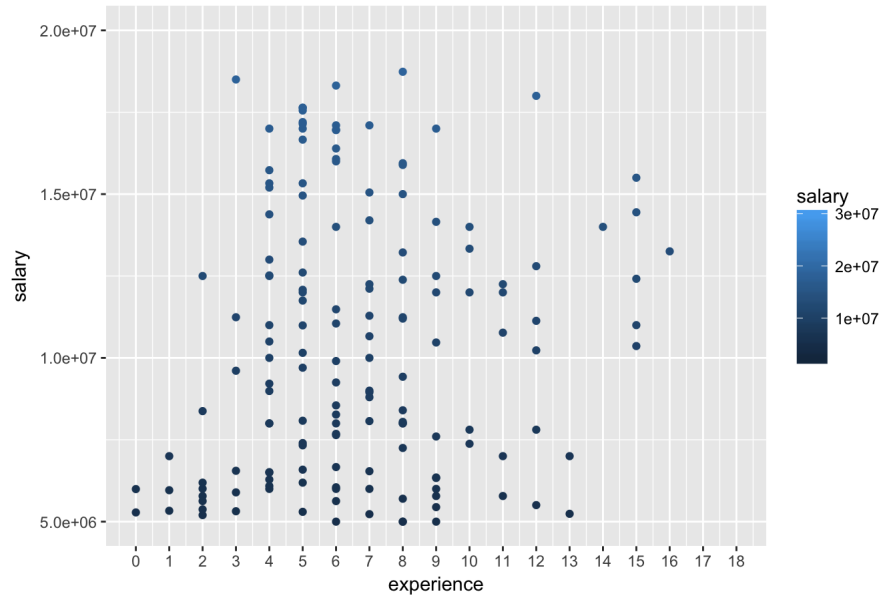


It is also possible to remove points we do not want with the argument **limits** and can graphically focus on more concentrating data. For example, I can eliminate players with too low or too high wages to find the general conclusion about the relationship between the two variables:

```
ggplot(data = dat, aes(x = experience, y = salary)) + geom_point(aes(color = salary)) + ggtitle('Salary vs. Experience of NBA Players in 2017') + scale_x_continuous(breaks = c(0:18)) + scale_y_continuous(limits = c(5e+06, 2e+07))
```

```
## Warning: Removed 288 rows containing missing values (geom_point).
```

Salary vs. Experience of NBA Players in 2017

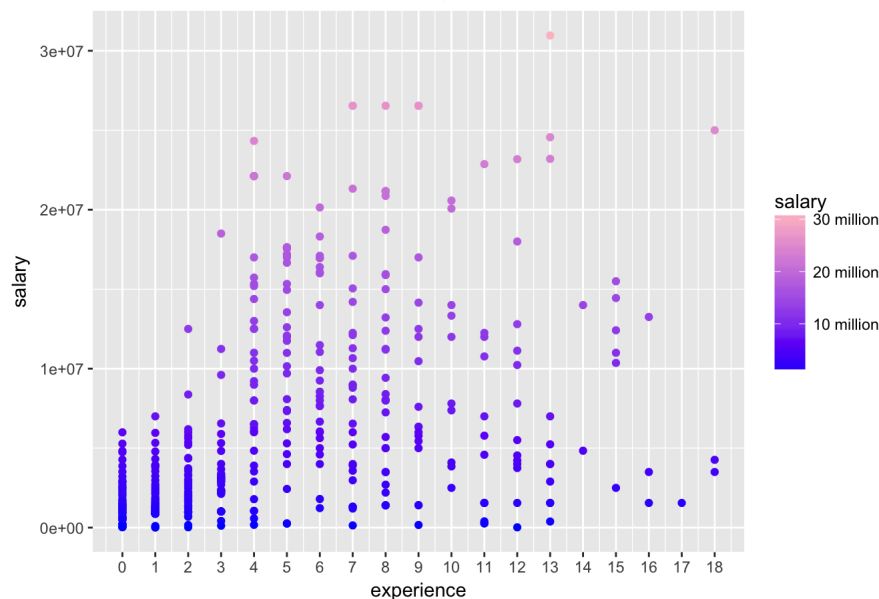


## Color and Fill

Similarly, we can also modify the setting for color with the function `scale_color_continuous`. For example, I can transform the break of salary in more readable way with unit of million dollars, and I can also define different colors for low and high values of salary:

```
ggplot(data = dat, aes(x = experience, y = salary)) + geom_point(aes(color = salary)) + ggtitle('Salary vs. Experience of NBA Players in 2017') + scale_x_continuous(breaks = c(0:18)) + scale_color_continuous(labels = c("0", "10 million", "20 million", "30 million"), low = "blue", high = "pink")
```

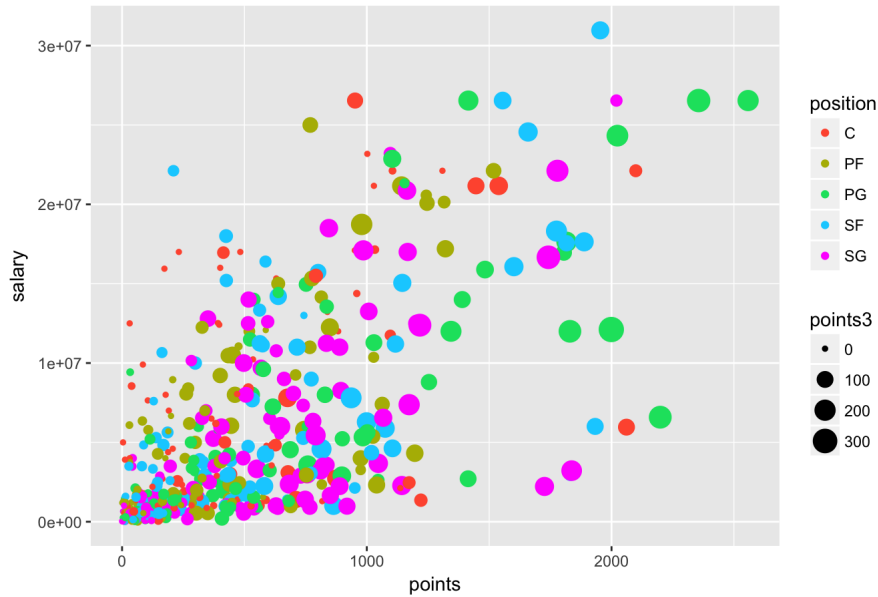
Salary vs. Experience of NBA Players in 2017



Other than continuous color scales, there are plenty of functions related to discrete color settings. For example, we can control various parameters such as `h` = hue in range of [0, 360], `c` = chroma (intensity of color), and `l` = luminance (lightness) in [0, 100]. Here I modify the graph from the very beginning, which is a scatterplot about points and salary:

```
ggplot(data = dat, aes(x = points, y = salary)) + geom_point(aes(color = position, size = points3)) + ggtitle('Salary vs. Points of NBA Players in 2017') + scale_color_discrete(l = 70, c = 180)
```

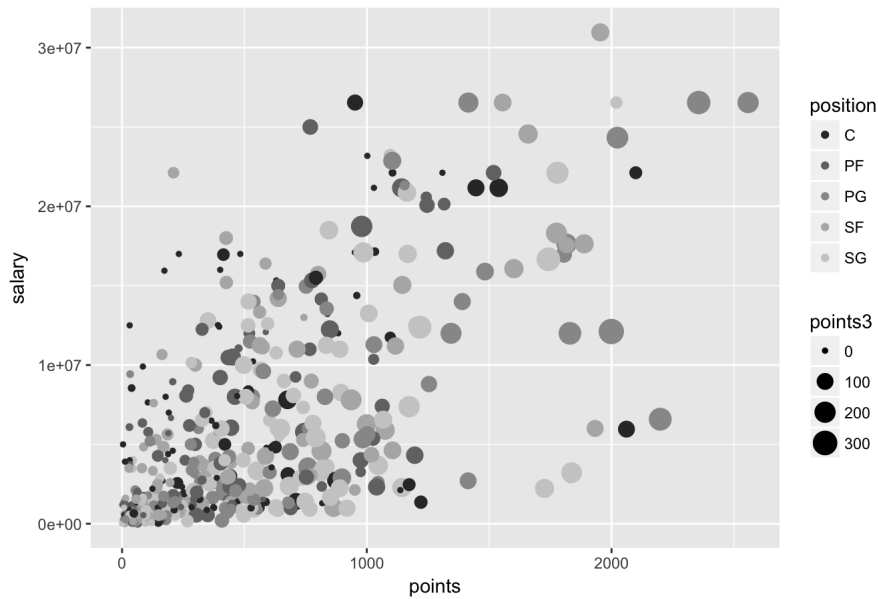
Salary vs. Points of NBA Players in 2017



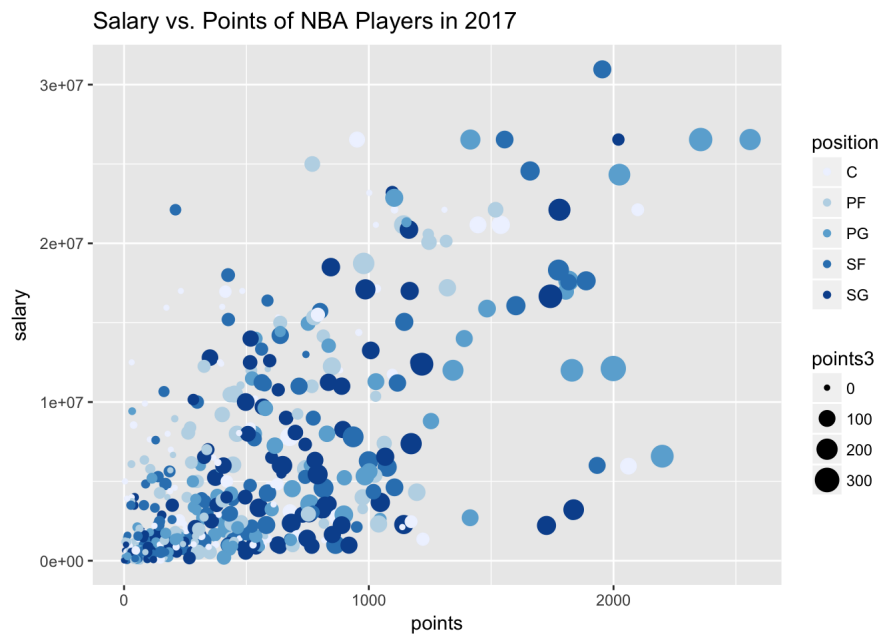
Several other functions about discrete color scales are like:

```
ggplot(data = dat, aes(x = points, y = salary)) +
  geom_point(aes(color = position, size = points3)) + ggtitle('Salary vs. Points of NBA Players in 2017') + scale_
color_grey()
```

Salary vs. Points of NBA Players in 2017



```
ggplot(data = dat, aes(x = points, y = salary)) + ggtitle('Salary vs. Points of NBA Players in 2017') +
  geom_point(aes(color = position, size = points3)) + scale_color_brewer()
```



Scales functions in other types such as shape, size and line type can be figured out with similar pattern, and they really enable us to freely design the graphs we want.

## Themes

### What is Theme?

Themes in `ggplot2` manage the overall look of the layer behind the plots we create with data. Elements include:

- Axis labels
- Plot background
- Facet label background
- Legend appearance

`theme_gray()` is the default one, but there are also other built-in themes with `ggplot2` such as:

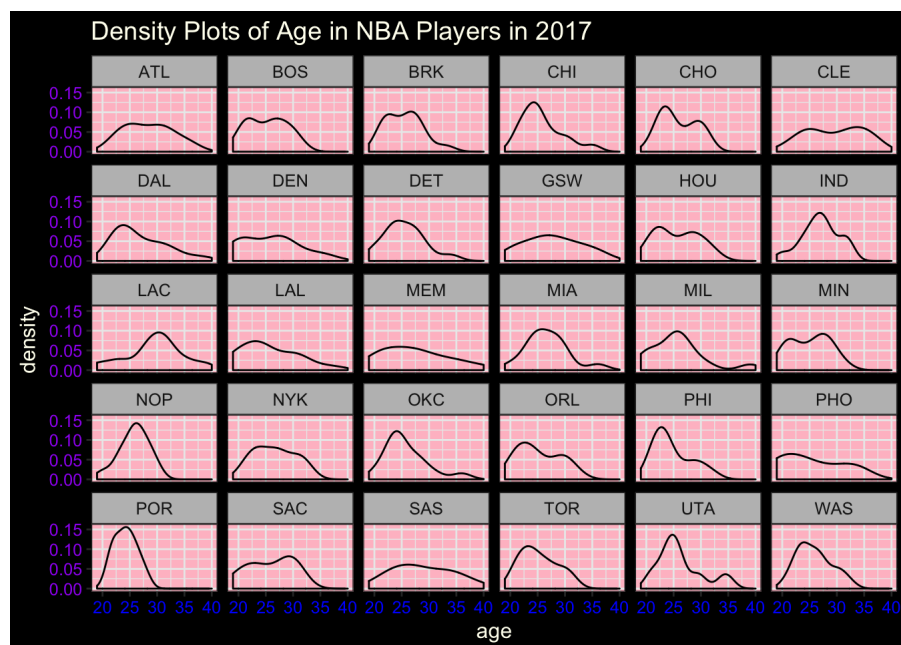
- `theme_bw()`
- `theme_classic()`
- `theme_dark()`
- `theme_light()`
- `theme_linedraw()`
- `theme_minimal()`

### Example

However, you can always build your own themes with unlimited flexibility, as long as you fill in the values for each argument. For example, I create a new theme based on `theme_bw()` for the density plots of age faceting by teams below:

```
theme_new <- theme_bw() +
  theme(plot.background = element_rect(size = 1, color = "black", fill = "black"),
        text=element_text(size = 12, color = "ivory"),
        axis.text.y = element_text(colour = "purple"),
        axis.text.x = element_text(colour = "blue"),
        panel.background = element_rect(fill = "pink"),
        strip.background = element_rect(fill = "grey"))

# density plots of age faceting by team
ggplot(data = dat, aes(x = age)) +
  geom_density() + ggtitle('Density Plots of Age in NBA Players in 2017') +
  facet_wrap(~ team) + theme_new
```



## Conclusion and Message

In conclusion, this post tries to draw our attention to the side details of ggplot2 graphs, which contribute to more custom extensions for the plot framework. With scales functions, we can adjust the value of position, color, size etc, in order to find the most effective way to display the data; with theme functions, we can not only use various default ones, but also design personal themes for new visual experience. Therefore, both topics are closely related to our operations about `ggplot2` visualisation, and some really important skills that we need to master are also well involved.

## References

1. [https://www.ted.com/talks/david\\_mccandless\\_the\\_beauty\\_of\\_data\\_visualization](https://www.ted.com/talks/david_mccandless_the_beauty_of_data_visualization)
2. <http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html#scales>
3. [https://www3.nd.edu/~steve/computing\\_with\\_data/12\\_Scales\\_themes/scales\\_themes.html](https://www3.nd.edu/~steve/computing_with_data/12_Scales_themes/scales_themes.html)
4. <http://www.sthda.com/english/wiki/ggplot2-axis-scales-and-transformations>
5. <http://www.dummies.com/programming/r/how-to-work-with-scales-in-a-ggplot2-in-r/>
6. [http://ggplot2.tidyverse.org/reference/scale\\_discrete.html](http://ggplot2.tidyverse.org/reference/scale_discrete.html)
7. <https://www.r-bloggers.com/ggplot2-themes-examples/>
8. <http://www.hafro.is/~einarhj/education/ggplot2/scales.html>
9. <http://www.sthda.com/english/wiki/ggplot2-themes-and-background-colors-the-3-elements>