Introduction to Heatmap

Linda Li

12/3/2017

Introduction

When it comes to data visualizations, R is always a good choice as it has various practical tools, like ggplot2 and the base R graphics, to accomplish the goal. Today, in this post, I will talk about another function called heatmap that is designed to visualize patterns in the data sets. I will give a brief overview for how to use the function.

Background

According to techopedia, heat maps uses colors to give representations that are two-dimensional. they help users to have visualized simple and complex information. Heat maps have a cariety of applications

Data Set

The following data is collected and calculated with the source www.basketball-reference.com. Efficiency is calculated by efficiency = (points + rebounds + assists + steals + blocks - missed_field_goals - missed_free_throws - turnovers)/games_played. Salary is the total amount of salaries in dollars for each team. Points is the sum of all points that the team has gained in games, and expereience is the total amount of years of playing for each team.

```
## Importing data set
dat <- read.csv(file = "~/stat133/stat133-hws-fall17/post02/data/nba2017-teams.csv")
dat
```

```
##     X team salary points efficiency experience
## 1   1  ATL  90.89   7759   140.3269         93
## 2   2  BOS  91.92   8857   148.2525         63
## 3   3  BRK  65.45   7495   147.7823         52
## 4   4  CHI  92.08   7349   139.1025         58
## 5   5  CHO 100.25   8127   145.2994         66
## 6   6  CLE 125.79   8605   177.8585        128
## 7   7  DAL  92.10   6910   148.2243         62
## 8   8  DEN  78.38   8769   167.3595         74
## 9   9  DET 103.07   8309   136.3762         55
## 10 10  GSW  98.69   9473   172.3916        101
## 11 11  HOU  84.66   8469   155.1031         56
## 12 12  IND  84.57   7918   135.0697         84
## 13 13  LAC 114.78   8911   147.1242        124
## 14 14  LAL  86.27   7354   143.9768         66
## 15 15  MEM 108.34   7995   140.9707         83
## 16 16  MIA  72.78   8312   151.9902         63
## 17 17  MIL  90.27   8390   153.2588         64
## 18 18  MIN  59.38   8634   144.8383         48
## 19 19  NOP  90.63   6563   164.2521         55
## 20 20  NYK  97.01   8060   143.9033         59
## 21 21  OKC  86.98   8104   146.8680         55
## 22 22  ORL 102.41   7408   125.1406         57
## 23 23  PHI  55.78   7116   164.0916         34
## 24 24  PHO  72.53   8399   144.3065         68
## 25 25  POR 103.03   8254   143.7321         43
## 26 26  SAC  88.19   6348   148.3954         68
## 27 27  SAS 104.69   8578   146.6236         99
## 28 28  TOR 108.46   8166   158.7658         57
## 29 29  UTA  80.32   8258   145.8193         71
## 30 30  WAS  98.78   8163   143.0117         56
```

dat is a data set that contains information about efficiency, salary, points, and experience for each team.

Arrange Data

Currently the data set is in the alphabetical order with regard to the teams' names. I would like to rearrange the data into descending order (from highest to lowest) with regard to salary by using arrange function.

```
## Calling package needed
library(dplyr)
```

```
##
## 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
```

```
## Arranging data
dat1 <- arrange(dat, desc(salary))
dat1
```

```
##     X team salary points efficiency experience
## 1   6  CLE 125.79   8605   177.8585        128
## 2  13  LAC 114.78   8911   147.1242        124
## 3  28  TOR 108.46   8166   158.7658         57
## 4  15  MEM 108.34   7995   140.9707         83
## 5  27  SAS 104.69   8578   146.6236         99
## 6   9  DET 103.07   8309   136.3762         55
## 7  25  POR 103.03   8254   143.7321         43
## 8  22  ORL 102.41   7408   125.1406         57
## 9   5  CHO 100.25   8127   145.2994         66
## 10 30  WAS  98.78   8163   143.0117         56
## 11 10  GSW  98.69   9473   172.3916        101
## 12 20  NYK  97.01   8060   143.9033         59
## 13  7  DAL  92.10   6910   148.2243         62
## 14  4  CHI  92.08   7349   139.1025         58
## 15  2  BOS  91.92   8857   148.2525         63
## 16  1  ATL  90.89   7759   140.3269         93
## 17 19  NOP  90.63   6563   164.2521         55
## 18 17  MIL  90.27   8390   153.2588         64
## 19 26  SAC  88.19   6348   148.3954         68
## 20 21  OKC  86.98   8104   146.8680         55
## 21 14  LAL  86.27   7354   143.9768         66
## 22 11  HOU  84.66   8469   155.1031         56
## 23 12  IND  84.57   7918   135.0697         84
## 24 29  UTA  80.32   8258   145.8193         71
## 25  8  DEN  78.38   8769   167.3595         74
## 26 16  MIA  72.78   8312   151.9902         63
## 27 24  PHO  72.53   8399   144.3065         68
## 28  3  BRK  65.45   7495   147.7823         52
## 29 18  MIN  59.38   8634   144.8383         48
## 30 23  PHI  55.78   7116   164.0916         34
```

The data set could be rearranged with regard to different columns in the data set.

Data Preparation

Data preparation is crucial to the latter graphical works. How well the data is prepared can have a significant impact on the efficiency. Since we have 30 teams in total, I would like to ease our work process by focusing on the top-20 teams with the most salary

```
## Extracting the top 20 teams from dat1
top_20 <- dat1[1:20, ]

## Rearrange top_30 in ascending order of salary to prepare for graph
top_20 <- arrange(top_20, salary)
```

In order for us to have a cleaer view for graphs, I would like to replace row numbers with the name of the teams

```
## Replaceing row numbers with names of teams
row.names(top_20) <- top_20$team
```

After changing the name of the row names, we could now delete the team column from the data set

```
## Removing team column
top_20 <- top_20[,3:6]
top_20
```

```
##     salary points efficiency experience
## OKC  86.98   8104   146.8680         55
## SAC  88.19   6348   148.3954         68
## MIL  90.27   8390   153.2588         64
## NOP  90.63   6563   164.2521         55
## ATL  90.89   7759   140.3269         93
## BOS  91.92   8857   148.2525         63
## CHI  92.08   7349   139.1025         58
## DAL  92.10   6910   148.2243         62
## NYK  97.01   8060   143.9033         59
## GSW  98.69   9473   172.3916        101
## WAS  98.78   8163   143.0117         56
## CHO 100.25   8127   145.2994         66
## ORL 102.41   7408   125.1406         57
## POR 103.03   8254   143.7321         43
## DET 103.07   8309   136.3762         55
## SAS 104.69   8578   146.6236         99
## MEM 108.34   7995   140.9707         83
## TOR 108.46   8166   158.7658         57
## LAC 114.78   8911   147.1242        124
## CLE 125.79   8605   177.8585        128
```

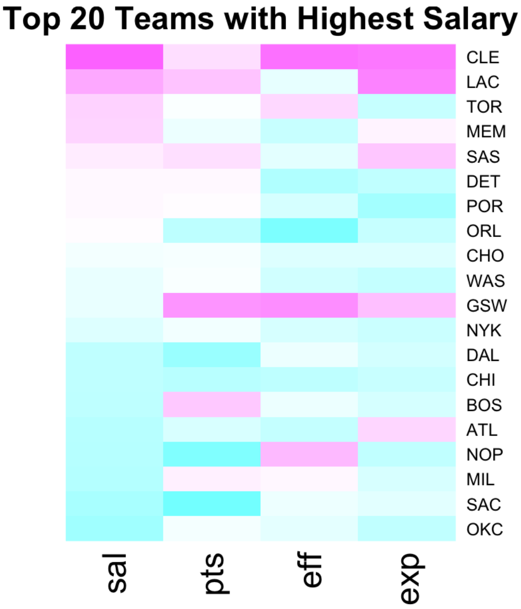
In order to make a heat map, the data set, which is currently a data frame, needs to be a matrix by using the data.matrix function.

```
## Changing into matrix
top_20_m <- data.matrix(top_20)
```

Now, we have finished preping the data. We are moving on to create the Heatmap.

Creating Heatmap

```
## Drawing heatmap
top_20_heatmap <- heatmap(top_20_m, Rowv=NA, Colv=NA, col = cm.colors(999), scale="column", margins=c(5,10), main
= "Top 20 Teams with Highest Salary", labCol = c("sal", "pts", "eff", "exp"))
```



Axis Label Explanation

- sal: Salary
- pts: Points
- eff: Efficiency
- exp: Expereience

Graph Interpretation

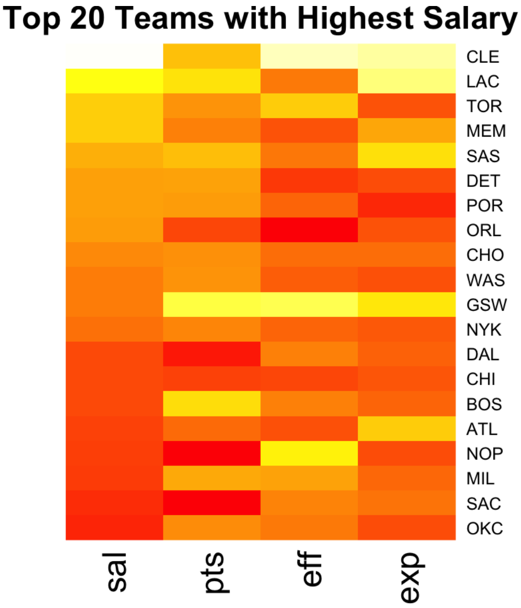
So what do these colorful boxes mean in terms of our data?

Since I arranged the data top_20 in a ascending manner with regard to salary, and CLE is the last team in the set, it is obvious that CLE is the team with the highest salary. Now take a look at the heatmap, the first column is the salary column and CLE is on the top. As we ranked the teams based on the salary, the first sal column can be used as a referrence column. From this column we could conclude that the most saturated pink means the first, and the most saturated blue means the last. Pink fades to white then gradually becomes blue. Using this information, we could easily find team rankings with regard to other factors. Possible correlations between one factor and another could also be found with ease by looking at the color type and saturations. For this data set, however, there is no direct correlations between different factors.

Changing Colors of the Heatmap

If you feel like this color scheme is not very intuitive for you, you could change it by changing “col =” in the heatmap function.

```
## Drawing heatmap
top_20_heatmap <- heatmap(top_20_m, Rowv=NA, Colv=NA, col = heat.colors(123), scale="column", margins=c(5,10), mai
n = "Top 20 Teams with Highest Salary", labCol = c("sal", "pts", "eff", "exp"))
```



Axis Label Explanation

- sal: Salary
- pts: Points
- eff: Efficiency
- exp: Expereience

Other Tools for Heatmaps

Heatmap function is not the only way to create a heatmap, the following packages/functions are also good options.

1. heatmaply:

- a function in the package “plotly”
- creates an interactive heatmap
- according to Tal Galili’s Introduction to Heatmaply, heatmaply allows
    - users to check specific values by moving the mouse over the cells
    - users to zoom into a region on the heatmap by dragging a rectangle around the area

2. pheatmap:

- a function from the package “pheatmap” that is used to create “pretty heatmap”
- creates a clustered heatmap
- according to Package ‘pheatmap’
    - these heatmaps have better control over some graphical parameters such as cell size etc.

3. ggplot2:

- there is no built-in function in this package for creating a heatmap
- in order to create a heatmap with the geom_tile()
- ggplot2: Quick Heatmap Plotting provides a short tutorial on how to create a heat map with ggplot
    - this tool uses a more complicated data preparation process.
    - the general formula for creating a heatmap using ggplot2 is ggplot()+geom_tile()+scale_fill_gradient().

Take Home Message

Heatmap is a very useful tool if we are trying to find certain patterns among a large number of data sets. With colored boxes, heatmap helps to improve the efficiency of finding patterns. Instead of going through the hassle of massive calculations, heatmap assists us to choose the most possible combinations, and therefore, reduces the workloads for us.

References:

1. https://www.techopedia.com/definition/32150/heat-map
2. https://learnr.wordpress.com/2010/01/26/ggplot2-quick-heatmap-plotting/
3. http://flowingdata.com/2010/01/21/how-to-make-a-heatmap-a-quick-and-easy-solution/
4. https://cran.r-project.org/web/packages/heatmaply/index.html
5. https://cran.r-project.org/web/packages/d3heatmap/index.html
6. https://cran.r-project.org/web/packages/pheatmap/index.html
7. https://cran.r-project.org/web/packages/heatmaply/vignettes/heatmaply.html
8. https://cran.r-project.org/web/packages/pheatmap/pheatmap.pdf
9. http://sebastianraschka.com/Articles/heatmaps_in_r.html