Heatmaps in R

Emmy Cheng

October 31, 2017

Introduction

Motivation:

The motivation behind researching heatmaps came from looking at ggplots. I wanted to know more ways to visualize data in R other than the ways we have already learned, such as ggplots, bar graphs, and other forms of graphs. This is how I stumbled across heatmaps.

Definition:

Essentially, heatmaps take a table of numbers and substitute those numbers with colored cells. This gives the graph a visual impact and helps to make it easier for the viewer to understand. For example, cells with higher numbers will be shaded more heavily to let the user know there are many or larger values within that cell. A heatmap is a native function in R and doesn't require installing any packages or libraries. While conducting my research, I also found out that we can use ggplots to create a similar output, and I will be exploring this as well.

Example

Let's get started with an example of NBA data (our favorite!).

Step 1

We start with a table of players and their stats.

```
nba <- read.csv("http://datasets.flowingdata.com/ppg2008.csv")
print(nba)
```

```
##                  Name  G  MIN  PTS  FGM  FGA   FGP FTM  FTA   FTP X3PM
## 1         Dwyane Wade  79 38.6 30.2 10.8 22.0 0.491 7.5  9.8 0.765  1.1
## 2        LeBron James  81 37.7 28.4  9.7 19.9 0.489 7.3  9.4 0.780  1.6
## 3         Kobe Bryant  82 36.2 26.8  9.8 20.9 0.467 5.9  6.9 0.856  1.4
## 4       Dirk Nowitzki  81 37.7 25.9  9.6 20.0 0.479 6.0  6.7 0.890  0.8
## 5       Danny Granger  67 36.2 25.8  8.5 19.1 0.447 6.0  6.9 0.878  2.7
## 6        Kevin Durant  74 39.0 25.3  8.9 18.8 0.476 6.1  7.1 0.863  1.3
## 7        Kevin Martin  51 38.2 24.6  6.7 15.9 0.420 9.0 10.3 0.867  2.3
## 8        Al Jefferson  50 36.6 23.1  9.7 19.5 0.497 3.7  5.0 0.738  0.0
## 9          Chris Paul  78 38.5 22.8  8.1 16.1 0.503 5.8  6.7 0.868  0.8
## 10   Carmelo Anthony  66 34.5 22.8  8.1 18.3 0.443 5.6  7.1 0.793  1.0
## 11         Chris Bosh  77 38.1 22.7  8.0 16.4 0.487 6.5  8.0 0.817  0.2
## 12        Brandon Roy  78 37.2 22.6  8.1 16.9 0.480 5.3  6.5 0.824  1.1
## 13    Antawn Jamison  81 38.2 22.2  8.3 17.8 0.468 4.2  5.6 0.754  1.4
## 14        Tony Parker  72 34.1 22.0  8.9 17.5 0.506 3.9  5.0 0.782  0.3
## 15  Amare Stoudemire  53 36.8 21.4  7.6 14.1 0.539 6.1  7.3 0.835  0.1
## 16        Joe Johnson  79 39.5 21.4  7.8 18.0 0.437 3.8  4.6 0.826  1.9
## 17      Devin Harris  69 36.1 21.3  6.6 15.1 0.438 7.2  8.8 0.820  0.9
## 18      Michael Redd  33 36.4 21.2  7.5 16.6 0.455 4.0  4.9 0.814  2.1
## 19         David West  76 39.3 21.0  8.0 17.0 0.472 4.8  5.5 0.884  0.1
## 20  Zachary Randolph  50 35.1 20.8  8.3 17.5 0.475 3.6  4.9 0.734  0.6
## 21      Caron Butler  67 38.6 20.8  7.3 16.2 0.453 5.1  6.0 0.858  1.0
## 22      Vince Carter  80 36.8 20.8  7.4 16.8 0.437 4.2  5.1 0.817  1.9
## 23   Stephen Jackson  59 39.7 20.7  7.0 16.9 0.414 5.0  6.0 0.826  1.7
## 24         Ben Gordon  82 36.6 20.7  7.3 16.0 0.455 4.0  4.7 0.864  2.1
## 25     Dwight Howard  79 35.7 20.6  7.1 12.4 0.572 6.4 10.7 0.594  0.0
## 26        Paul Pierce  81 37.4 20.5  6.7 14.6 0.457 5.7  6.8 0.830  1.5
## 27     Al Harrington  73 34.9 20.1  7.3 16.6 0.439 3.2  4.0 0.793  2.3
## 28    Jamal Crawford  65 38.1 19.7  6.4 15.7 0.410 4.6  5.3 0.872  2.2
## 29           Yao Ming  77 33.6 19.7  7.4 13.4 0.548 4.9  5.7 0.866  0.0
## 30 Richard Jefferson  82 35.9 19.6  6.5 14.9 0.439 5.1  6.3 0.805  1.4
## 31        Jason Terry  74 33.6 19.6  7.3 15.8 0.463 2.7  3.0 0.880  2.3
## 32    Deron Williams  68 36.9 19.4  6.8 14.5 0.471 4.8  5.6 0.849  1.0
## 33         Tim Duncan  75 33.7 19.3  7.4 14.8 0.504 4.5  6.4 0.692  0.0
## 34       Monta Ellis  25 35.6 19.0  7.8 17.2 0.451 3.1  3.8 0.830  0.3
## 35          Rudy Gay  79 37.3 18.9  7.2 16.0 0.453 3.3  4.4 0.767  1.1
## 36         Pau Gasol  81 37.1 18.9  7.3 12.9 0.567 4.2  5.4 0.781  0.0
## 37    Andre Iguodala  82 39.8 18.8  6.6 14.0 0.473 4.6  6.4 0.724  1.0
## 38    Corey Maggette  51 31.1 18.6  5.7 12.4 0.461 6.7  8.1 0.824  0.5
## 39         O.J. Mayo  82 38.0 18.5  6.9 15.6 0.438 3.0  3.4 0.879  1.8
## 40      John Salmons  79 37.5 18.3  6.5 13.8 0.472 3.6  4.4 0.830  1.6
## 41  Richard Hamilton  67 34.0 18.3  7.0 15.6 0.447 3.3  3.9 0.848  1.0
## 42         Ray Allen  79 36.3 18.2  6.3 13.2 0.480 3.0  3.2 0.952  2.5
## 43 LaMarcus Aldridge  81 37.1 18.1  7.4 15.3 0.484 3.2  4.1 0.781  0.1
## 44       Josh Howard  52 31.9 18.0  6.8 15.1 0.451 3.3  4.2 0.782  1.1
## 45  Maurice Williams  81 35.0 17.8  6.5 13.9 0.467 2.6  2.8 0.912  2.3
## 46  Shaquille O'neal  75 30.1 17.8  6.8 11.2 0.609 4.1  6.9 0.595  0.0
## 47     Rashard Lewis  79 36.2 17.7  6.1 13.8 0.439 2.8  3.4 0.836  2.8
## 48  Chauncey Billups  79 35.3 17.7  5.2 12.4 0.418 5.3  5.8 0.913  2.1
## 49     Allen Iverson  57 36.7 17.5  6.1 14.6 0.417 4.8  6.1 0.781  0.5
```

```
## 50     Nate Robinson  74 29.9 17.2  6.1 13.9 0.437 3.4  4.0 0.841  1.7
##    X3PA  X3PP ORB DRB  TRB  AST STL BLK  TO  PF
## 1   3.5 0.317 1.1 3.9  5.0  7.5 2.2 1.3 3.4 2.3
## 2   4.7 0.344 1.3 6.3  7.6  7.2 1.7 1.1 3.0 1.7
## 3   4.1 0.351 1.1 4.1  5.2  4.9 1.5 0.5 2.6 2.3
## 4   2.1 0.359 1.1 7.3  8.4  2.4 0.8 0.8 1.9 2.2
## 5   6.7 0.404 0.7 4.4  5.1  2.7 1.0 1.4 2.5 3.1
## 6   3.1 0.422 1.0 5.5  6.5  2.8 1.3 0.7 3.0 1.8
## 7   5.4 0.415 0.6 3.0  3.6  2.7 1.2 0.2 2.9 2.3
## 8   0.1 0.000 3.4 7.5 11.0  1.6 0.8 1.7 1.8 2.8
## 9   2.3 0.364 0.9 4.7  5.5 11.0 2.8 0.1 3.0 2.7
## 10  2.6 0.371 1.6 5.2  6.8  3.4 1.1 0.4 3.0 3.0
## 11  0.6 0.245 2.8 7.2 10.0  2.5 0.9 1.0 2.3 2.5
## 12  2.8 0.377 1.3 3.4  4.7  5.1 1.1 0.3 1.9 1.6
## 13  3.9 0.351 2.4 6.5  8.9  1.9 1.2 0.3 1.5 2.7
## 14  0.9 0.292 0.4 2.7  3.1  6.9 0.9 0.1 2.6 1.5
## 15  0.1 0.429 2.2 5.9  8.1  2.0 0.9 1.1 2.8 3.1
## 16  5.2 0.360 0.8 3.6  4.4  5.8 1.1 0.2 2.5 2.2
## 17  3.2 0.291 0.4 2.9  3.3  6.9 1.7 0.2 3.1 2.4
## 18  5.8 0.366 0.7 2.5  3.2  2.7 1.1 0.1 1.6 1.4
## 19  0.3 0.240 2.1 6.4  8.5  2.3 0.6 0.9 2.1 2.7
## 20  1.9 0.330 3.1 6.9 10.1  2.1 0.9 0.3 2.3 2.7
## 21  3.1 0.310 1.8 4.4  6.2  4.3 1.6 0.3 3.1 2.5
## 22  4.9 0.385 0.9 4.2  5.1  4.7 1.0 0.5 2.1 2.9
## 23  5.2 0.338 1.2 3.9  5.1  6.5 1.5 0.5 3.9 2.6
## 24  5.1 0.410 0.6 2.8  3.5  3.4 0.9 0.3 2.4 2.2
## 25  0.0 0.000 4.3 9.6 13.8  1.4 1.0 2.9 3.0 3.4
## 26  3.8 0.391 0.7 5.0  5.6  3.6 1.0 0.3 2.8 2.7
## 27  6.4 0.364 1.4 4.9  6.2  1.4 1.2 0.3 2.2 3.1
## 28  6.1 0.360 0.4 2.6  3.0  4.4 0.9 0.2 2.3 1.4
## 29  0.0 1.000 2.6 7.2  9.9  1.8 0.4 1.9 3.0 3.3
## 30  3.6 0.397 0.7 3.9  4.6  2.4 0.8 0.2 2.0 3.1
## 31  6.2 0.366 0.5 1.9  2.4  3.4 1.3 0.3 1.6 1.9
## 32  3.3 0.310 0.4 2.5  2.9 10.7 1.1 0.3 3.4 2.0
## 33  0.0 0.000 2.7 8.0 10.7  3.5 0.5 1.7 2.2 2.3
## 34  1.0 0.308 0.6 3.8  4.3  3.7 1.6 0.3 2.7 2.7
## 35  3.1 0.351 1.4 4.2  5.5  1.7 1.2 0.7 2.6 2.8
## 36  0.0 0.500 3.2 6.4  9.6  3.5 0.6 1.0 1.9 2.1
## 37  3.2 0.307 1.1 4.6  5.7  5.3 1.6 0.4 2.7 1.9
## 38  1.9 0.253 1.0 4.6  5.5  1.8 0.9 0.2 2.4 3.8
## 39  4.6 0.384 0.7 3.1  3.8  3.2 1.1 0.2 2.8 2.5
## 40  3.8 0.417 0.7 3.5  4.2  3.2 1.1 0.3 2.1 2.3
## 41  2.8 0.368 0.7 2.4  3.1  4.4 0.6 0.1 2.0 2.6
## 42  6.2 0.409 0.8 2.7  3.5  2.8 0.9 0.2 1.7 2.0
## 43  0.3 0.250 2.9 4.6  7.5  1.9 1.0 1.0 1.5 2.6
## 44  3.2 0.345 1.1 3.9  5.1  1.6 1.1 0.6 1.7 2.6
## 45  5.2 0.436 0.6 2.9  3.4  4.1 0.9 0.1 2.2 2.7
## 46  0.0 0.000 2.5 5.9  8.4  1.7 0.7 1.4 2.2 3.4
## 47  7.0 0.397 1.2 4.6  5.7  2.6 1.0 0.6 2.0 2.5
## 48  5.0 0.408 0.4 2.6  3.0  6.4 1.2 0.2 2.2 2.0
## 49  1.7 0.283 0.5 2.5  3.0  5.0 1.5 0.1 2.6 1.5
## 50  5.2 0.325 1.3 2.6  3.9  4.1 1.3 0.1 1.9 2.8
```

This data set confusing, as it it just a bunch of numbers.

Step 2

Let's order the data in decreasing order of points scored.

```
nba <- nba[order(nba$PTS),]
```

Step 3

Name the rows by player rather than number and get rid of the first column since we don't need a column with player names anymore.

```
row.names(nba) <- nba$Name
nba <- nba[,2:20]
```

Step 4

Right now, we have a data frame but like I mentioned earlier we want to have a matrix of numbers so let's convert the data frame into a data matrix.

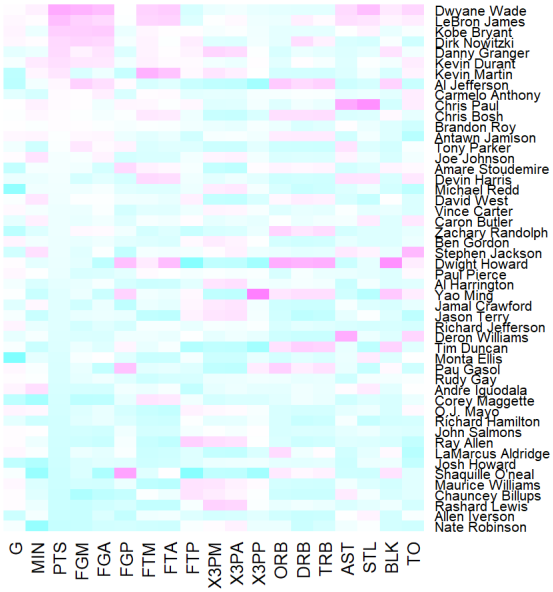
```
matrix_nba <- data.matrix(nba)
```

Step 5

Now, we wil make the heatmap. Before I make it, I will explain the arguments in the function. Here is the code with general arguments that I will be using: `heatmap(x, Rowv, Colv, col, scale, margins)` x is the matrix of values that will be plotted Rowv determines how the row will be computed and reordered Colv determines how the column will be computed and reordered col sets a color for the plot scale indicates if the values should be scaled in either row or column direction and centered margins is a vector of length 2 containing the margins of the plot for

loumn and row names, respectively.

```
nba_heatmap <- heatmap(matrix_nba, Rowv=NA, Colv=NA, col = cm.colors(300, alpha =1), scale="column", margins=c(5,1
0))
```



To analyze this heatmap, we see the darker and more purple colors to represent large numbers. Where the graph is almost white, we note that these are smaller numbers. Thus, we can interpret this as where the players are strong. For example, some are good at assists, some are good at shooting, and etc. Some may be good at a lot of things, and some are overall well rounded and don't have one particular strong suit.

Optional

You can change your color scheme by using different colors offered by R. I typed in ?cm.colors to find this color scheme, which defaults to cyan. For a more orange scheme, you can use heat.colors(300) in the 'col' argument.

Using ggplot

Previously, I mentioned how it is possible to use ggplot to make something similar to a heatmap. You just need to change the fill of the geom_tile part in the code to produce a similar result. You need to have scaled measurements of the values in the table as well. Since this function requires many packages and concepts that are hard to understand with our current knowledge of ggplot, I will not go into detail. I tried to do research on how to do it but ended up very confused as well. Therefore, I have just summarized what I could understand based on my research!

Key Takeaways

Using heatmaps is a really good way for us to understand matrices filled with numbers. It gives a visual representation and allows for a quick understanding of the data displayed. For example, we can use it to map out population densities, elevations, and much more. Because seeing numbers and only numbers can be difficult for the brain to understand, heatmaps make life a lot easier for visual learners. The bonus part of heatmaps is that you don't need to install a separate package to use this function, it's already built into R. How convenient!!

References

1. http://sebastianraschka.com/Articles/heatmaps_in_r.html
2. http://www.r-graph-gallery.com/portfolio/heatmap/
3. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/heatmap.html
4. https://www.r-bloggers.com/color-palettes-in-r/
5. http://www.r-graph-gallery.com/215-the-heatmap-function/
6. https://www.r-bloggers.com/heat-maps-using-r/
7. http://flowingdata.com/2010/01/21/how-to-make-a-heatmap-a-quick-and-easy-solution/
8. https://learnr.wordpress.com/2010/01/26/ggplot2-quick-heatmap-plotting/
9. http://ggplot2.tidyverse.org/reference/geom_tile.html