Post 1 - HeatMaps and You

Stanley Ho

October 30, 2017

Heatmaps And You

When dealing with data, data visualization helps tell your audience what is happening beyond just numbers and words. Visualizations are a statistician's best friend. In this blog, I will show you one way to visually impress your friends, families, and employers (and if you don't have any, impress yourself)! After reading this, you'll have a general understanding on why heatmaps are useful, how to make one, and how to interpret the visual.

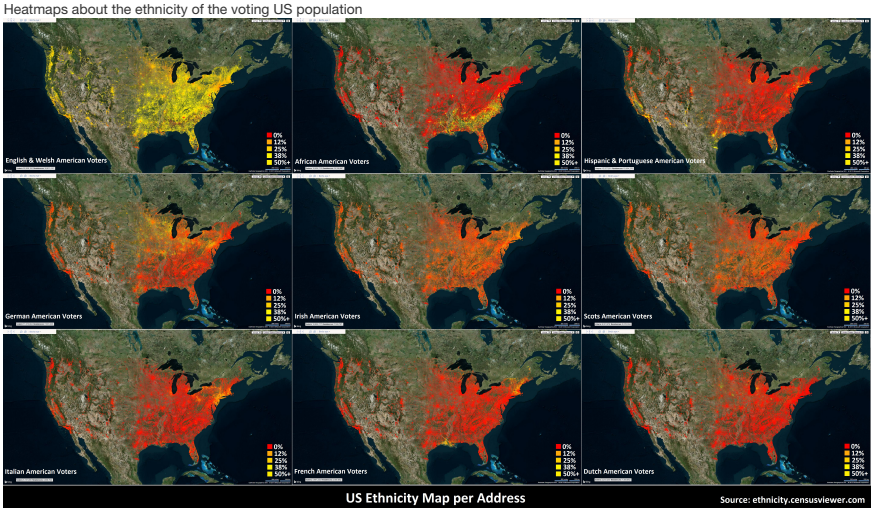
What is a Heatmap?

In R, a heatmap are substitutes numbers cells for color cells that allow for a quick visualization of the table. The color cells correspond to measurements that are useful for finding highs, lows, and even patterns. Source: For a clearer explanation on heatmaps.

Complex examples of heatmaps you may have seen are:

- Heatmaps pertaining to where mouse clicks happen on a webpage

- Heatmaps about the ethnicity of the voting US population



Example of a simple HeatMap

Before we can do anything, we need data!
Go here, to get the data used in this demonstration

After you downloaded the file, move it into the same folder as where you plan on saving your code. Then create a folder named data to place the just downloaded file (this will allow for the below code `dat <- read.csv("./data/nba2017-player-statistics.csv")` to work)

```
## Read the file in R
dat <- read.csv("./data/nba2017-player-statistics.csv")
```

As a general rule of thumb, you should see what's inside your data by calling `str(dat)`, `head(dat, n = 5)`, and `tail(dat, n = 5)`. Now that we are fimilar with our data, let's work on manipulating the data to get it ready for calling `heatmap()`.

Step 1. Create a column for total points scored for each player.

As you can tell, there is no column in our data frame that shows us the total points a player made, which includes: * 3 points for each point3 * 2 points for each point2 * 1 point for each FTM So let's create that ourselves.

```
## Create a vector to store Total Points
Total_Points <- 3 * dat$Points3 + 2 * dat$Points2 + dat$FTM

## Add Total_Points into our data
dat$Total_Points <- Total_Points
```

Step 2. Sort our data by total points scored.

Let's sort our data in decreasing order by total points. Afterwards, let's focus on the top 50 NBA players of 2017. Then we'll order the top 50 players by increasing for our heatmap.

```
## Sort data in decreasing order
dat <- dat[order(dat$Total_Points, decreasing = TRUE), ]

## Shorten list to only include the top 50
nba_50 <- dat[1:50, ]

## Another reorder to make the heatMap make sense
nba_50 <- dat[order(nba_50$Total_Points),]
```

Step 3. Set row.names to Player names.

To make our data clearer, let's set data's row.names to player names. Then we can remove player names from `nba_50`, since row.names will be player names and we don't need the redundancy.

```
## Set row.names of data to dat$Player
row.names(nba_50) <- nba_50$Player

## Remove the dat$Player as a column
nba_50 <- nba_50[, 2:length(nba_50)]
```

Step 4. Make your data frame into a matrix.

If you type in `?heatmap`, the R Documentation for `heatmap()` will explain that the function takes in matrices, not dataframes. So for this step, we will need to convert our data frame into a matrix, using `data.matrix()`.

```
nba_50_matrix <- data.matrix(nba_50)
```

Step 5. Use HeatMap

Now that we have manipulated our data into containing the information that we want, let's finally use `heatmap()` on our matrix! Again refer to the R Documentation by using `?heatmap`.

```
nba_50_heatmap <- heatmap(nba_50_matrix, Rowv=NA, Colv=NA, col = cm.colors(256), scale="column", margins=c(5, 10),
main = "Top 50 NBA Players in 2017")
```

Step 6. Interpreting the HeatMap.

Okay, cool; we have a pretty looking HeatMap. What does it tell us? What do you do with it?
Great questions, well, let's interpret the data visualization together.

Now what do the colors mean?

From our data manipulations, we sorted our data by the total amount of points a player made. This can be seen by the variable, `Total_Points`, seen on the bottom right of our graph. We will use this column as a reference for what the colors mean since it perfectly changes from one end to the other end.
Well, if we call `row.names(nba_50[nba_50$Total_Points == max(nba_50$Total_Points), ])` to see who scored the most points in 2017, we get Russell Westbrook. If we look at the cell for `Total_Points` and Russell Westbrook, it is the darkest shade of purple in that column. This means that the darker the purple, the higher the the player ranks in that column, compared to the rest of the Top 50 players.

Now that we know what the colors mean, what patterns can we notice about the NBA Top 50 Players of 2017?

This is why heatmaps are so useful, we are able to look at the visualization and make quick inferences about the data. At a glance, you can tell there is a concentration of purple to the top, and a concentration of the blue hue at the bottom. We can interpret this to say that the top players at the top generally have a higher impact or a leading role in the game.

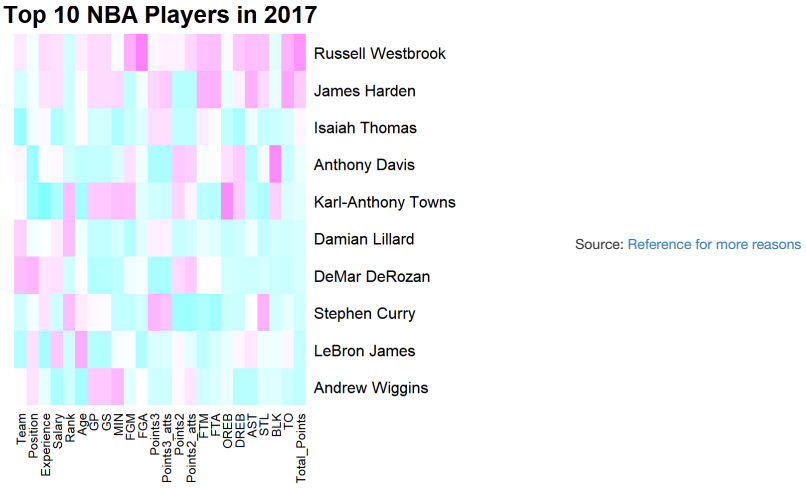
So why Heatmaps?

Well, as you have experienced, heats maps make things visually easier to look at. Lets say you were given a table:

	Team	Position	Experience	Salary	Rank	Age	GP	GS	MIN	FGM	FGA	Points3	Points3_atts	Points2	Points2_atts	FTM	FTA	OREB	DREB	AST	ST
Andrew Wiggins	MIN	SF	2	6006600	1	21	82	82	3048	709	1570	103	289	606	1281	412	542	103	226	189	8
LeBron James	CLE	SF	13	30963450	1	32	74	74	2794	736	1344	124	342	612	1002	358	531	97	543	646	9
Stephen Curry	GSW	PG	7	12112359	2	28	79	79	2638	675	1443	324	789	351	654	325	362	61	292	523	14
DeMar DeRozan	TOR	SG	7	26540100	1	27	74	74	2620	721	1545	33	124	688	1421	545	647	70	316	290	7
Damian Lillard	POR	PG	4	24328425	2	26	75	75	2694	661	1488	214	578	447	910	488	545	46	322	439	6
Karl-Anthony Towns	MIN	C	1	5960160	2	21	82	82	3030	802	1479	101	275	701	1204	356	428	296	711	220	5
Anthony Davis	NOP	C	4	22116750	1	23	75	75	2708	770	1527	40	134	730	1393	519	647	174	712	157	9
Isaiah	BOS	PG	5	6587132	1	27	76	76	2569	682	1473	245	646	437	827	590	649	43	162	449	7

	Team	Position	Experience	Salary	Rank	Age	GP	GS	MIN	FGM	FGA	Points3	Points3_atts	Points2	Points2_atts	FTM	FTA	OREB	DREB	AST	ST
Thomas James Harden	HOU	PG	7	26540100	1	27	81	81	2947	674	1533	262	756	412	777	746	881	95	564	906	12
Russell Westbrook	OKC	PG	8	26540100	1	28	81	81	2802	824	1941	200	583	624	1358	710	840	137	727	840	13
This table has a lot	of	rmation,	gi	ving you	exac	t	numbers	f	or	ever	.	ever,	in	cas	es	whe	re	this	is not	ne	eded,	heatmaps	provide	th	e same	informat	ion	in	a	vis	ually a	ppealin	g	way.

```
nba_10 <- data.matrix(tail(nba_50, n = 10))
nba_10 <- heatmap(nba_10, Rowv=NA, Colv=NA, col = cm.colors(256), scale="column", margins=c(5, 10), main = "Top 10 NBA Players in 2017")
```



Source: Reference for more reasons

to use Heatmaps and best practices

Advanced techniques with Heatmaps

There are many built in packages in R that can help you create even more beautiful heatmaps.

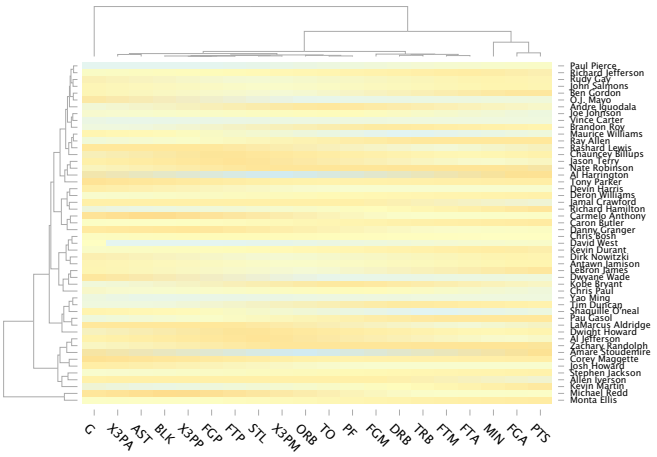
1. Source: Package called heatmaply This package is especially useful as it combines the feel of d3heatmap based on ggplot2+plotly. It also has features like:
    - "Shows the row/column/value under the mouse cursor (and includes a legend on the side)"
    - "Drag a rectangle over the heatmap image, or the dendrograms, in order to zoom in (the dendrogram coloring relies on integration with the dendextend package)"
2. Source: ggplot2 example 1. Source: ggplot2 simple example.

    Another way to make beautiful heatmaps is to use ggplot2. These are two examples that would help explain how to set up your data and then use ggplot. For the most part, making a simple heat map using ggplot requires using `geom_tile()` and setting fill equal to what you want the colors to fill for.
    Youtube Source: Youtube
3. Source:d3heatmap blog from rstudio. Just another simple way to make heatmaps with dendograms (the bracket looking thigns around some heatmaps)

```
library(d3heatmap)
```

```
## Warning: package 'd3heatmap' was built under R version 3.4.2
```

```
url <- "http://datasets.flowingdata.com/ppg2008.csv"
nba_players <- read.csv(url, row.names = 1)
d3heatmap(nba_players, scale = "column")
```



Take Home Messages

1. Before you do anything with your data, it is a good idea to understand and see what your data contains.
2. The native function for heatmaps, `heatmaps()`, takes in only matrices, so if you have a dataframe, you need to convert it to a matrix.
3. Heatmaps are fast ways to visualize a matrix because each cell is replaced by a color which is easier to make inferences about.
4. There are multiple libraries to help make heatmaps: ggplot2, heatmaply, and d3heatmap
5. Hopefully you know how to make a simple heatmap.

References

1. https://flowingdata.com/2010/01/21/how-to-make-a-heatmap-a-quick-and-easy-solution/
2. https://channel9.msdn.com/Events/useR-international-R-User-conference/useR2016/Heatmaps-in-R-Overview-and-best-practices

3. https://www.r-bloggers.com/heatmaply-interactive-heat-maps/
4. https://learnr.wordpress.com/2010/01/26/ggplot2-quick-heatmap-plotting/
5. https://www.r-bloggers.com/how-to-make-a-simple-heatmap-in-ggplot2/
6. https://blog.rstudio.com/2015/06/24/d3heatmap/
7. https://www.youtube.com/watch?v=QD3XAla94rQ