Treemaps: A Brief Overview

Introduction

In my experience with data science, it has been difficult to find a clear winner between R and Python. There are pros and cons to both languages, and both have their strict adherents and detractors. As I've been using R more and more this semester, though, the beautiful visualizations ggplot2 enables have made quite an impression on me. Though Python enables the user to generate histograms, line charts, and other simple visualizations with minimal complexity, the level of depth at the fingertips of the R user is staggering.

Exploring the wide breadth of options provided by ggplot2 for data visualization is intimidating, akin to attempting to choose a course schedule with the vast number of offerings available. The introduction to the package provided in Stat 133 has provided me with a solid foundation in R visualization; in particular, facet wrapping has been fascinatingly complex, efficient, and gorgeous.

In writing this post, I decided I wanted to explore more of what R has to offer in the visualization sense. Keeping a grounding in course material, I decided to work with the NBA datasets we've been using throughout the course. My goal was to create a visualization that communicates some of the insights we've derived using other analytical techniques, but in a novel fashion. While exploring different applications of ggplot2, I discovered the treemap, and instantly was struck by the level of information that can be conveyed by such a simple visualization.

A treemap contains, within a small rectangle, information regarding magnitude, family, and name of data within a dataset. It can be used to describe many different types of data – GDP in countries across the world, revenues at competing retail outlets, or the average size of fruits from different families. Treemaps work perfectly for sports-centered statistics, where performance can be measured across different teams and individual players and can, importantly, be quantified relationally. That is, the key to a treemap is that relative magnitudes of, say, points scored, can easily be conveyed through the relative size of a rectangle. If this sounds unclear now, I promise it will become more clear as I include a sample treemap further on.

My discovery of treemaps has given me new ideas and interest in creative visualization methods. I hope that the reader learns something new about visualization techniques and perhaps becomes more interested in exploring the different methods of communicating insights derived from data. Through ggplot2 it is so easy and fun to test out new ways to present data, and each one can be tailored to highlight a different aspect of the dataset you're analyzing.

Creating a Treemap

To properly create a treemap in R, I recommend installing ggplot2 and treemapify at minimum. The reader should be familiar with ggplot2 as the premier source of visualization functions in R, but treemapify perhaps requires a bit more explanation.

Essentially, the treemapify package enables the user to create treemaps; it is not a standalone package and must be used concurrently with ggplot2. It functions as an extra 'geom', which you should remember is the subfunction within ggplot2 that enables the user to customize visualizations.

I also installed dplyr in order to merge and wrestle with two datasets, both of which I got from the Stat 133 course repository on github.

The code below displays the installation of the three packages I used to generate my treemaps, followed by my creation of a table merging two different datasets concerning NBA players. The reason I created a dataset merging the NBA roster with the individual statistics was so that I could organize the players by both team and salary. Important to notice is that, with treemaps, data can be categorized according to some other variable, in a fashion not unlike that of facet wrapping. The concept should become more clear as the reader continues the post.

As a last note, I sorted for players who make more than $15,000,000 in order to reduce the number of rectangles and display high profile players. The reader can imagine that a cramped, crowded visualization can be tough to read and comprehend.

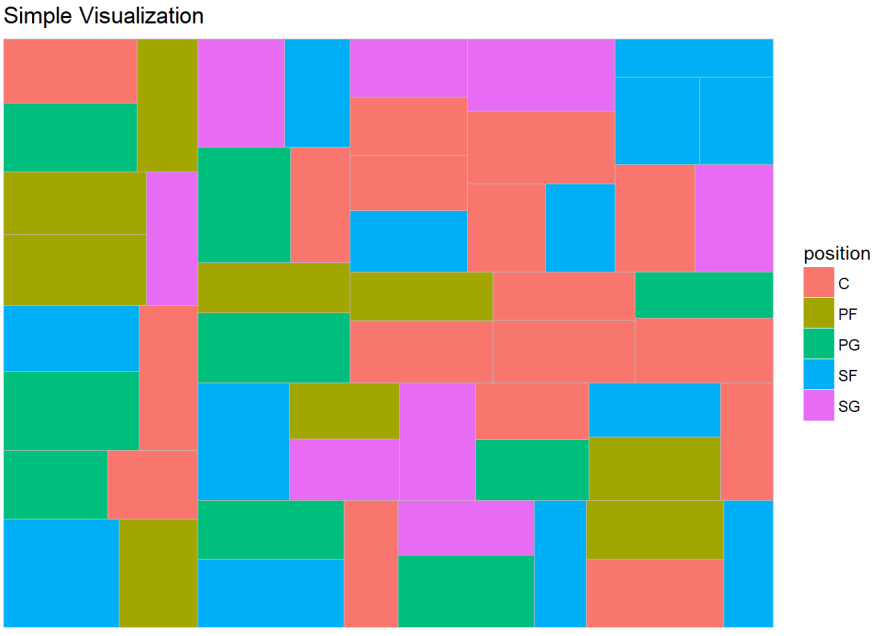
```
library(ggplot2)
library(dplyr)
library(treemapify)
players <- read.csv('nba2017-roster.csv')
plays <- read.csv('nba2017-stats.csv')
team_plays <- merge(players, plays, 'player')
team_plays <- filter(team_plays, salary > 15000000)
head(team_plays)
```

```
##             player team position height weight age experience   salary
## 1       Al Horford  BOS        C     82    245  30          9 26540100
## 2     Allen Crabbe  POR       SG     78    210  24          3 18500000
## 3  Andre Drummond  DET        C     83    279  23          4 22116750
## 4    Anthony Davis  NOP        C     82    253  23          4 22116750
## 5 Bismack Biyombo  ORL        C     81    255  24          5 17000000
## 6    Blake Griffin  LAC       PF     82    251  27          6 20140838
##   games_played minutes field_goals_made field_goals_atts field_goals_perc
## 1           68    2193              379              801            0.473
## 2           79    2254              303              647            0.468
## 3           81    2409              483              911            0.530
## 4           75    2708              770             1527            0.504
## 5           81    1793              179              339            0.528
## 6           61    2076              479              971            0.493
##   points3_made points3_atts points3_perc points2_made points2_atts
## 1           86          242        0.355          293          559
## 2          134          302        0.444          169          345
## 3            2            7        0.286          481          904
## 4           40          134        0.299          730         1393
## 5            0            0           NA          179          339
## 6           38          113        0.336          441          858
##   points2_perc points1_made points1_atts points1_perc off_rebounds
## 1        0.524          108          135        0.800           95
## 2        0.490          105          124        0.847           19
## 3        0.532          137          355        0.386          345
## 4        0.524          519          647        0.802          174
## 5        0.528          125          234        0.534          157
## 6        0.514          320          421        0.760          111
##   def_rebounds assists steals blocks turnovers fouls
## 1          369     337     52     87       116   138
## 2          206      93     54     20        62   171
## 3          771      89    124     89       152   237
## 4          712     157     94    167       181   168
## 5          410      74     25     91        95   202
## 6          385     300     58     23       142   157
```

Treemap Example

Now that the appropriate data is confined to one data frame, we can begin with the plotting. First, let's create a simple treemap with few labels.

```
ggplot(team_plays, aes(area = salary, fill = position, label = player, subgroup = team)) + geom_treemap() + ggtitl
e('Simple Visualization')
```



Notice the various rectangles and colors; these can be explained in relation to the aesthetic conditions in the ggplot function. The area of each rectangle is determined by the salary of the player in particular. The fill color is determined by position; this enables us to see salary comparisons between positions. The label, player, does not show up in this visualization, because I have yet to specify the text type and color that will be displayed in the boxes. Similarly, the team subgroup does not show up in the visualization for the same reasons.

Now, let's add some text.

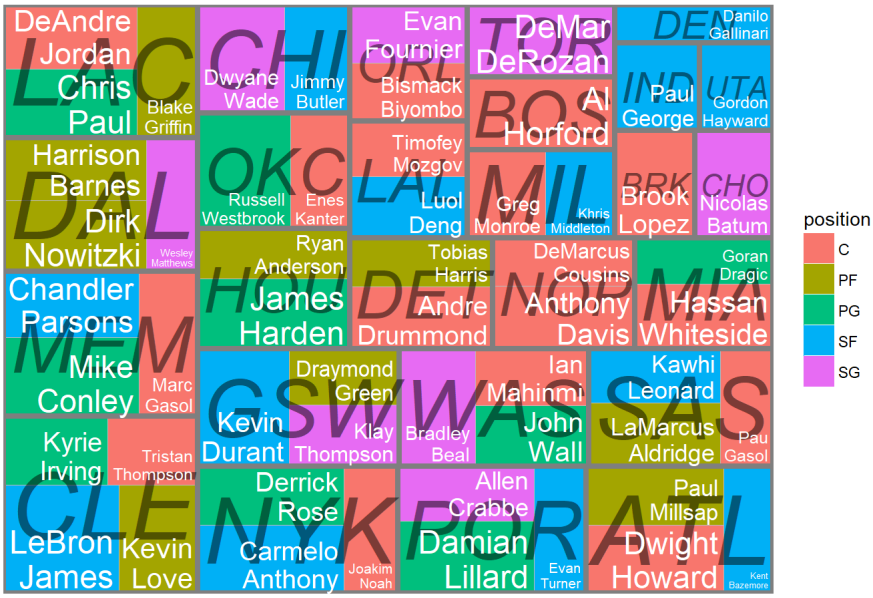
```
ggplot(team_plays, aes(area = salary, fill = position, label = player, subgroup = team)) + geom_treemap() + geom_t
reemap_subgroup_border() + geom_treemap_subgroup_text(place = 'centre', grow = T, alpha = 0.5, colour = 'black', f
ontface = 'italic', min.size = 0) + ggtitle('Team Based Breakdown')
```

Team Based Breakdown

It's becoming a bit more clear what we're trying to visualize and communicate. We see that certain players must play for certain times, and we also see different positions represented by each team. Knowing that there are 30 NBA teams, it's clear that some teams have no players making more than $15,000,000 per year. It's also clear that there are some teams, such as Cleveland, Memphis, and the Clippers, that have a few extremely highly paid players. However, we still cannot see individual players, so let's add that next.

```
ggplot(team_plays, aes(area = salary, fill = position, label = player, subgroup = team)) + geom_treemap() + geom_t
reemap_subgroup_border() + geom_treemap_subgroup_text(place = 'centre', grow = T, alpha = 0.5, colour = 'black', f
ontface = 'italic', min.size = 0) + geom_treemap_text(colour = 'white', place = 'bottomright', reflow = T) + ggtit
le('NBA Players Sorted by Team, Salary, and Position')
```

NBA Players Sorted by Team, Salary, and Position



Now, we can see the specific players in the dataset that make more than $15,000,000 per year. Their positions are communicated through color, and they're grouped with their teammates who also make exorbitant amounts of money. This visualization in particular is striking because it shows the disparity even amongst highly paid players across teams, salary-wise. Further, it shows that some teams don't even make it to the top as far as paying their players.

Conclusion

Personally, what I found most interesting about this exercise is simply seeing this elite group of players lumped together and broken down by team and position. For the many teams that have only one player (or even no players) who makes more than 15 million dollars, the reader might wonder where the extra money is being spent, or even whether financial endowments are even across NBA teams. This visualization does not communicate any information regarding total salary paid to all players across teams, but it is still clear that some teams simply pay their best players more than others. The specific correlation between ability and salary I will leave to another visualization, another day, but for now it is a fun thought experiment to imagine why one team might pay one superstar a higher salary than another team will. The amount of confounding variables must be astounding, so sometimes I will have to defer my wandering thoughts to a simply created, visually striking treemap.

References

https://github.com/wilkox/treemapify

https://rdrr.io/cran/treemapify/man/geom_treemap.html

http://www.r-graph-gallery.com/portfolio/treemap/

https://rpubs.com/brandonkopp/creating-a-treemap-in-r

https://cran.r-project.org/web/packages/treemap/treemap.pdf

https://www.rdocumentation.org/packages/treemap/versions/2.4-2/topics/treemap

https://flowingdata.com/2010/02/11/an-easy-way-to-make-a-treemap/

https://www.r-bloggers.com/treemap-world-population-visualisation/

https://cran.r-project.org/web/packages/treemapify/index.html