

Treemapify - Plotting Treemaps in ggplot2

Jessie Hua

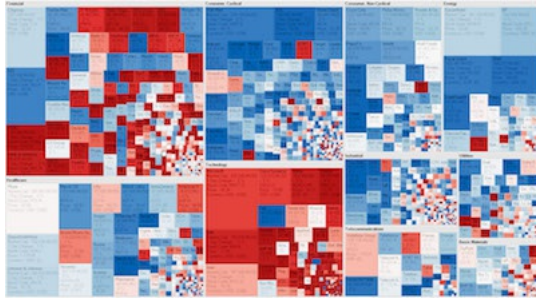
December 2, 2017

Introduction

We've been using the ggplot2 package a lot in class to plot a variety of different graphs, including bar charts, histograms, and scatterplots; however, there is one type of graph that I find very unique and useful that we have not covered in class - Treemaps. A treemap is split up into rectangles that are sized, colored, and ordered by a quantitative variable. Since it is not a very common graph to see, I will provide many examples on how they are used to represent data. But why am I interested in treemaps in the first place? At first glance, I thought it looked very neat and cool! I love the design and mix of colors. It also conveys the pattern of a set of data efficiently. After some research, I also found out that treemap is great for displaying large amounts of hierarchically structured data.

In this post, I will go over how to install and display treemaps along with the various ways to style treemaps. Lastly, I will introduce the function to subgroup treemaps.

Before we start, here's a cool treemap! :)



Installation and G-20 Data Set

Treemapify is a package that provides the ggplot2 library geoms to plot treemaps. Therefore, we need to install and use both treemapify and ggplot2 library. If you don't have either of those, run the following commands to install the packages:

```
install.packages(c("treemapify", "ggplot2"))
```

As you install treemapify, you've also obtained an example dataset called "G20" that contains statistics about the G-20 group of major world economies. We will use this dataset to plot treemaps.

```
library(ggplot2)
library(treemapify)
```

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

```
head(G20)
```

```
##           region      country  gdp_mil_usd  hdi  econ_classification
## 1      Africa  South Africa    384315  0.629      Developing
## 2 North America United States  15684750  0.937      Advanced
## 3 North America      Canada    1819081  0.911      Advanced
## 4 North America      Mexico    1177116  0.775      Developing
## 5 South America      Brazil    2395968  0.730      Developing
## 6 South America      Argentina  474954  0.811      Developing
```

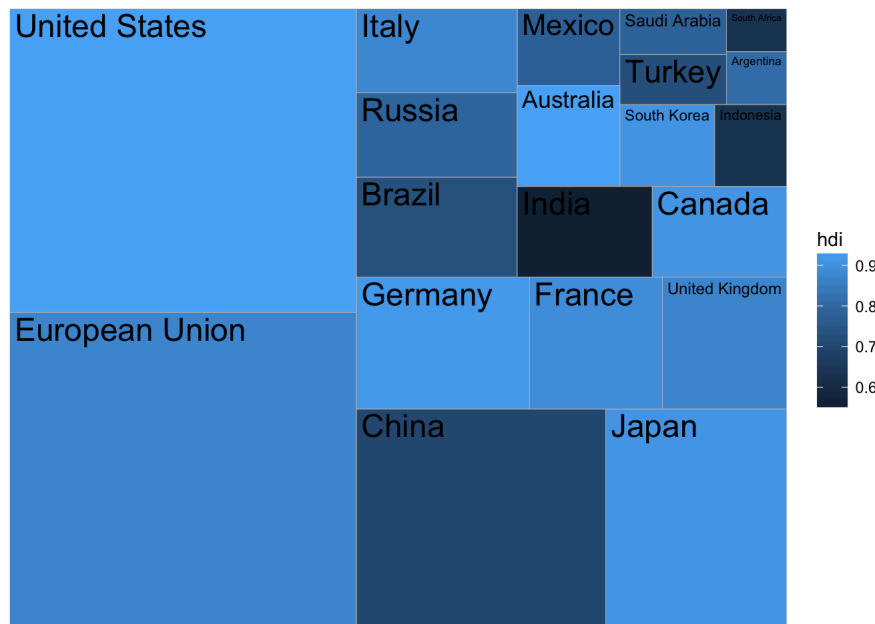
Basic treemaps

Let's make a treemap! The command "geom_treemap()" creates a simple treemap from the dataset specified in "ggplot()". The two aesthetics "area" and "fill" corresponds to the data that is represented by the area of each rectangle and the data for the color of the rectangle. Each rectangle represents a variable, which is the country in this example. Both "area" and "fill" are numeric values, and the greater the value is, the larger the area or the darker the color of the rectangle.

In addition, there is a legend provided to the right of the graph that shows the numeric value corresponding to the range of colors. This provides the range of values corresponding to the colors and thus helps readers get a general sense of the value each color represents.

We also want to display what each rectangle represents, because it is an essential part in making the graph useful. In order to do so, we use the command "geom_tree_map_text()" along with "label" aesthetics inside of "ggplot()". The "label" aesthetics corresponds to the column name that contains the name of each variable. Remember that "label" and "area" aesthetics are both required for "geom_treemap_text()"!

```
ggplot(G20, aes(area = gdp_mil_usd, fill = hdi, label = country)) +
  geom_treemap() +
  geom_treemap_text()
```



Styling treemaps

Labels

What grabbed my attention about treemap is its unique design! So of course we need to talk about styling! Let's continue with the labels we've created. There are many arguments within "geom_treemap_text()", including fontface, alpha, colour, place, and grow. Below is a brief explanation for each argument.

- Fontface: different options for each font, including bold, italic, and bold.italic
- Alpha: transparency of the label (from 0 to 1)
- Colour: color of the label
- Place: position of the label in respect to the rectangle (Default is bottom)
- Grow: if True, then the label size increases with the size of the rectangle
- Reflow, if True, then the label is wrapped to better fit the rectangle

```
ggplot(G20, aes(area = gdp_mil_usd, fill = hdi, label = country)) +
  geom_treemap() +
  geom_treemap_text(fontface = "bold.italic", alpha = 0.7, colour = "darkslategray1",
    place = "centre", grow = TRUE, reflow = TRUE)
```

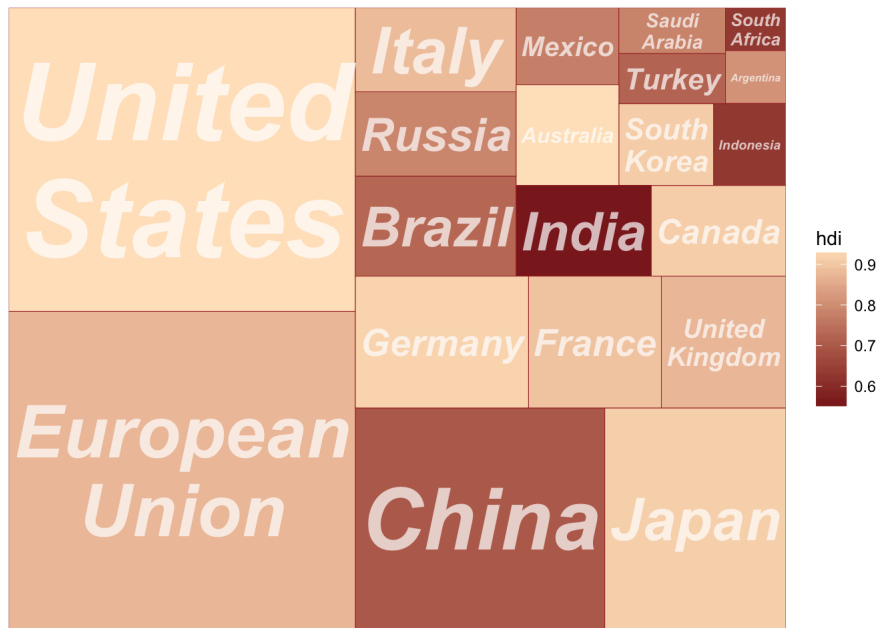


Rectangles/Boxes

Another important aspect of styling treemap is the design of the rectangles. To add borders, it is similar to the way you change the border color of a bar chart in ggplot. Inside "geom_treemap", add the argument "colour" and specify the border color.

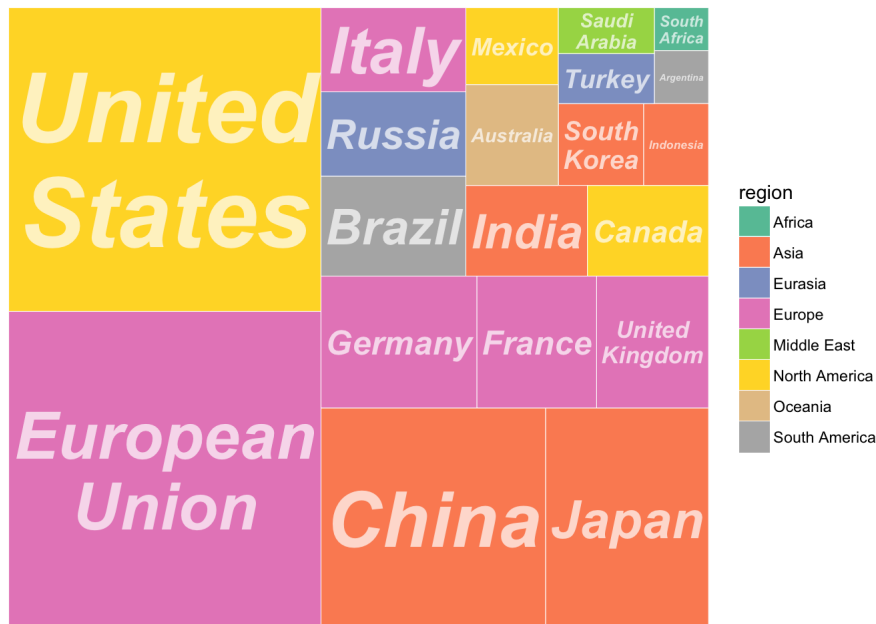
Changing the fill of the rectangles require an additional command "scale_fill_gradient". Since the fill is a gradient, you need to specify two colors - low and high.

```
ggplot(G20, aes(area = gdp_mil_usd, fill = hdi, label = country)) +
  geom_treemap(colour = "brown") +
  geom_treemap_text(fontface = "bold.italic", alpha = 0.7, colour = "white",
    place = "centre", grow = TRUE, reflow = TRUE) +
  scale_fill_gradient(low = "brown4", high = "bisque")
```



Since the fill represents a continuous value - hdi, we used a gradient. However, we can also represent discrete values with color. In this case, each color can specify the region of the country. The command “scale_fill_brewer” contains various given palettes, including Accent, Dark2, Paired, Pastel1, Pastel2, Set1, Set2, and Set3.

```
ggplot(G20, aes(area = gdp_mil_usd, fill = region, label = country)) +
  geom_treemap(colour = "white") +
  geom_treemap_text(fontface = "bold.italic", alpha = 0.7, colour = "white",
    place = "centre", grow = TRUE, reflow = TRUE) +
  scale_fill_brewer(palette = "Set2")
```



Labs and Theme

Like other ggplots, “labs” and “theme” provides a variety of options to style the graph. We will experiment with the following few:

Labs

- title: title of the chart, appears in the top-left
- caption: caption of the chart, appears in the bottom right

Theme

- legend.position: position of the color legend

```
ggplot(G20, aes(area = gdp_mil_usd, fill = region, label = country)) +
  geom_treemap(colour = "white") +
  labs(title = "The G-20 major economies", caption = "The area of each rectangle represents gdp_mil_used of a coun
try while the color represents the region") +
  geom_treemap_text(fontface = "bold.italic", alpha = 0.7, colour = "white",
    place = "centre", grow = TRUE, reflow = TRUE) +
  scale_fill_brewer(palette = "Set2") +
  theme(legend.position = "bottom")
```

The G-20 major economies



The area of each rectangle represents gdp_mil_used of a country while the color represents the region

Subgrouping treemaps

Another very useful property of treemap is its ability to easily create subgroups. In the above graph, we represented region with discrete colors and replaced hdi. This sacrifice does not have to be made, because we can use the subgroup command to show different regions, while keeping the rectangles' colors as values of hdi.

In "ggplot()", add aesthetics "subgroup" to indicate which column is used to create these subgroups. In this case, we are grouping the countries by region. The rectangles will be repositioned so that countries in the same region is part of a bigger rectangle that represents their region.

To display the subgroups, add "geom_treemap_subgroup_border()" to emphasize on the border between the subgroups.

"geom_treemap_subgroup_text" adds a label for each subgroup and takes in similar arguments with "geom_treemap_text". The labels could overlap, so we want to position them in different areas of the rectangles.

```
ggplot(G20, aes(area = gdp_mil_usd, fill = hdi, label = country,
  subgroup = region)) +
  geom_treemap() +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 0.5, colour =
    "black", fontface = "italic", min.size = 0) +
  geom_treemap_text(fontface = "bold.italic", alpha = 0.7, colour = "darkslategray1",
    place = "bottomleft", reflow = TRUE)
```



Conclusion and take home message

Treemaps are not only visually appealing but also efficiently convey information about multiple qualitative and/or quantitative variables of a large data set. There are various ways to style treemaps, and you can even create subgroups within the graph to display more information. Overall, I hope that I introduced you to a new way to represent data sets using treemapify and ggplot2!

Sources

1. <http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html#Tufte%20Boxplot>
2. https://docs.tibco.com/pub/spotfire/6.5.0/doc/html/tree/tree_what_is_a_treemap.htm
3. <https://github.com/wilkox/treemapify>
4. http://mlbernauer.github.io/R/20150309_treemaps_with_ggplot2.html
5. <https://cran.rstudio.com/web/packages/treemapify/vignettes/introduction-to-treemapify.html>
6. <https://stackoverflow.com/questions/18322442/color-in-ggplot-continuous-value-applied-to-discrete-variable>
7. http://ggplot2.tidyverse.org/reference/scale_brewer.html