Post01-Tiffany-Tsay

Tiffany Tsay

October 23, 2017

Data Visualization for Categorical Variables

> How can you breakdown a data set for its categorical composition and easily present this breakdown visually to viewers?

That is an important question to answer for any analysis of data. For quantitative variables, it's relatively easy to display the set and even describe the set using measures such as mean, standard deviation, etc. On the other hand, categorical variables rely much more heavily on a visual display to easily convey its composition.

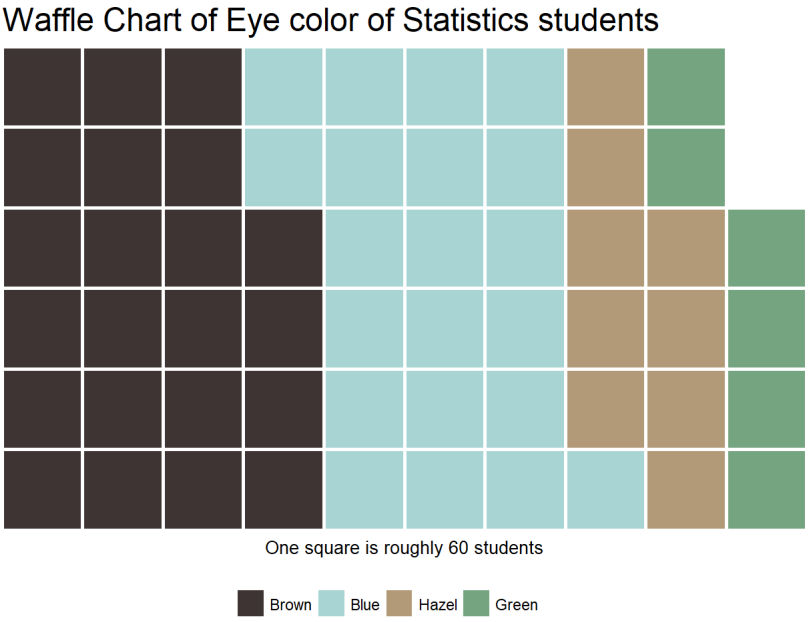
By exploring the 3 most common and best ways to display categorical composition, we can better understand why these displays are effective and the strengths of each individual display.

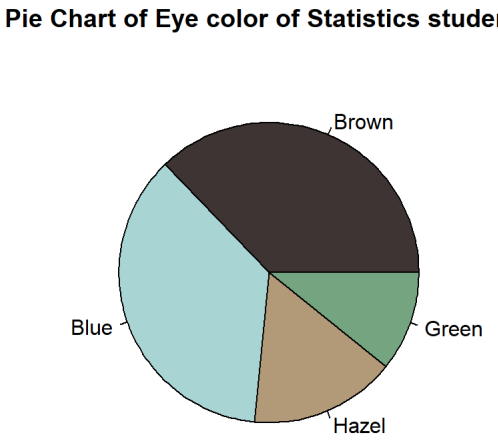
- Waffle chart
- Pie chart
- Bar chart

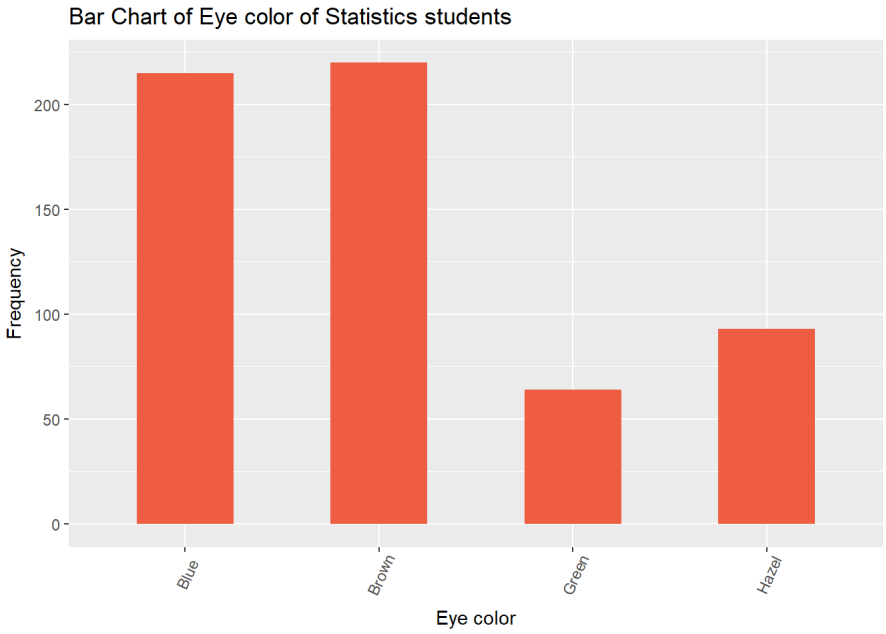
Some other types of categorical display that have been gaining traction are the tree map and mosaic plots, which will not be explored here. Tree maps are used best suited for hierarchal categorical data. Mosaic plots are good if there are multiple categorical variables being explored.

The easiest way to create these charts is with the help of the package `ggplot2`, and `waffle`. We will also be using sample data from the library `datasets` which is built into R. ggplot2 makes many of the plots much simpler to create than with the natural graphics package inside of R. Some of the more popular and recognizeable graphs of these four are the pie chart and bar chart.

Some quick samples of the above chart types (dataset from R's dataset package):







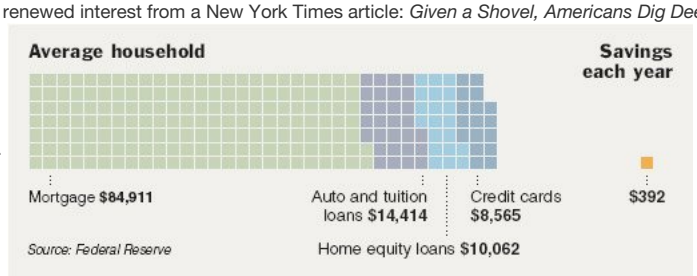
Waffle Charts

Why the waffle chart?

Waffle charts are often a better alternative to bar charts simply because of the visual interest and ability to hold the reader's attention. Similar to the bar chart, Waffle charts also do not distort the data. The visually appealing nature of the pie chart is found in the waffle chart, but the waffle chart also helps the viewer understand the sheer size of the data set. Pie charts often only show the percentage, while waffle charts can also depict the frequency itself. By letting viewers know how much one square is supposed to represent, both the proportion and actual frequency is preserved in the display.

Waffle charts gained particular traction and renewed interest from a New York Times article: *Given a Shovel, Americans Dig Deeper Into Debt*.

This article was published on July 20, 2008.



This waffle chart was praised at how obvious how little savings contributed relative to everything else, as well as the ability to actively engage readers.

Recreating the waffle chart

In order to create a waffle chart the `waffle` package is best suited to make the process much easier. Waffle charts can be presented in two different formats, one that mimics the traditional pie chart percentage representation where the chart is strictly a 10x10 grid, or one that can be rectangular and preserve the differencies in frequency a little bit more. The sample data that we will be experimenting with to recreate the above waffle chart earlier, will be from the `HairEyeColor` dataset that was built into R's dataset package.

We will have to clean the dataset a little bit, since the frequency of each aspect of the dataset (Eye, Sex, Hair) were recorded based on the

frequency of a combination of all three. So in order for us to only observe the frequency of Eye color we will have to clean the data. After the cleaning the data to only get the frequency of eye colors independent of the other factors the frequencies should look like the following:

```
fixedsample
```

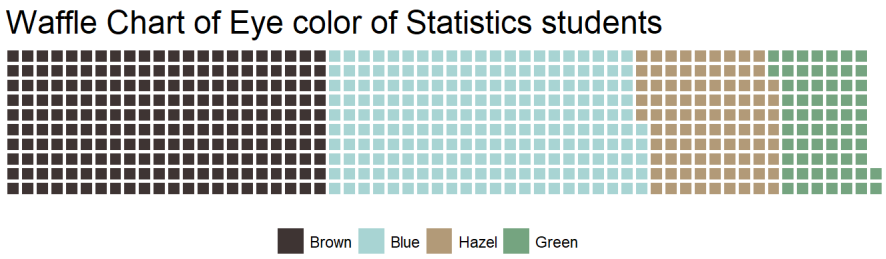
```
##       colorfreq
## Brown       220
## Blue        215
## Hazel        93
## Green        64
```

For our purposes, the waffle function `waffle()` takes in four important arguments: parts, rows, size, and colors.

- Parts would be the respective frequencies for the different observed options for the categorical variable, in our case the frequencies for brown, blue, green, and hazel eyes. Parts needs to have the format `c("Observed Option 1" = frequency1, "Observed Option 2" = frequency2...)`. Parts can also be scaled by dividing it by whatever factor it is desired. Scaling can often help shrink the data an dmake it less intimading with smaller squares.
- Rows determines the number of rows to be used in the waffle chart, so it should often time be considered with what makes the data look the most pleasing in conjunction with the parts argument.
- Size determines the size of the squares, with default being 1.
- Color is used to match the each observed option with a specific color that it should be used to represent it in the chart. Colors can be entered using #HEX values.

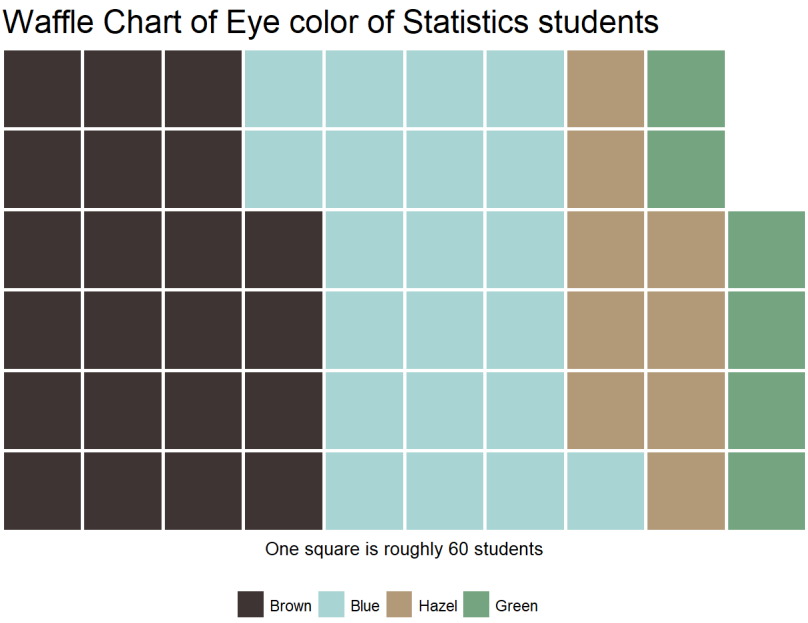
Here are a few variations of the waffle chart using different values for arguments.

```
parts = c('Brown' = brownfreq, 'Blue' = bluefreq, 'Hazel' = hazelfreq, 'Green' = greenfreq)
colors = c("#3E3433", "#A8D4D3", "#B29A78","#75A480")
waffle(parts, rows = 10, size = 1, colors = colors, title = "Waffle Chart of Eye color of Statistics students", le
gend_pos = "bottom")
```



Here the waffle chart has 10 rows, with each square being an observation directly.

```
waffle(parts/10, rows = 6, size = 1, colors = colors, title = "Waffle Chart of Eye color of Statistics students",
xlab = "One square is roughly 60 students", legend_pos = "bottom")
```



Here the waffle chart has 6 rows, with each square representing roughly 60 students.

Pie Charts

Why the pie chart?

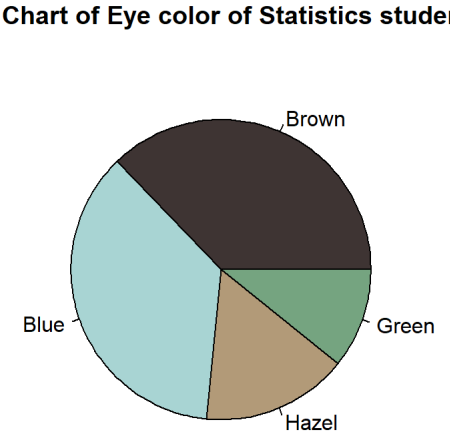
Pie charts have been a staple to charts simply because they are eyecatching and not overwhelming to readers who have little experience or knowledge of the field. However there are many critiques against the pie chart, but they can still be approproiate to use in certain cases. When your categorical variable does not have many observable options (fewer than 6 options) the pie chart excels in delivering the proportionality information very quickly. When your goal is to emphasize proportions as a whole, pie charts are good. However expecting your viewer to quickly differentiate the difference between 20% and 25% on a pie chart isn't always easy, and can often be misleading since pie charts don't always tell the viewer how large the sample or population was. Pie charts should strictly be reserved as an introductory look for small sized tables.

Recreating the pie chart

The fastest way to create a pie chart is to use the built in function within R. Once again we will be using the same sample data, and build the pie chart using the `pie()` function. For our purposes, the pie function takes in three important arguments: frequencies, labels, and colors.

- Frequencies simply is a vector of the different frequencies of the observed options. It would have the format of `c(frequency1, frequency2, ...)`.
- Labels determines what each of the frequencies correspond to and are formatted also a vector of `c(Observed Option 1, Observeed Option2, ...)`.
- Color is used to match the each observed option with a specific color that it should be used to represent it in the chart. Colors can be entered using #HEX values. To recreate our previous pie chart you can use:

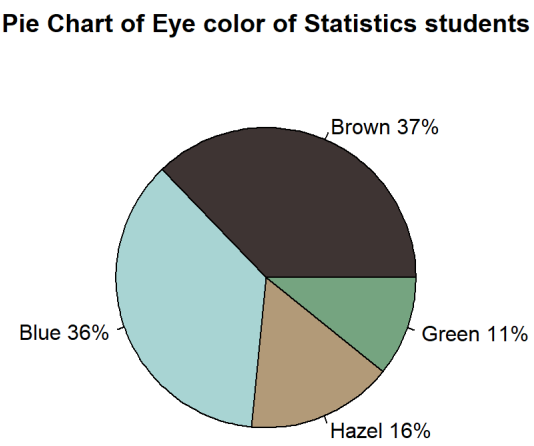
```
colorfreq = c(brownfreq, bluefreq, hazelfreq, greenfreq)
labels <- c("Brown", "Blue", "Hazel", "Green")
colors = c("#3E3433", "#A8D4D3", "#B29A78","#75A480")
pie(colorfreq, labels = labels, main = "Pie Chart of Eye color of Statistics students", col = colors)
```



Similarly you can append the percentage to the end of the labels so that it becomes obvious what the exact percentage is supposed to be. You

can do so by modifying the above chart as such:

```
percent <- round(colorfreq/sum(colorfreq)*100) #Calculate percentages
labels <- paste(labels, percent) # add percents to labels
labels <- paste(labels,"%",sep="") # ad % to labels
pie(colorfreq, labels = labels, main = "Pie Chart of Eye color of Statistics students", col = colors)
```



You can also create 3D Pie charts using the package `plotrix`, with the function `pie3d`.

Bar Charts

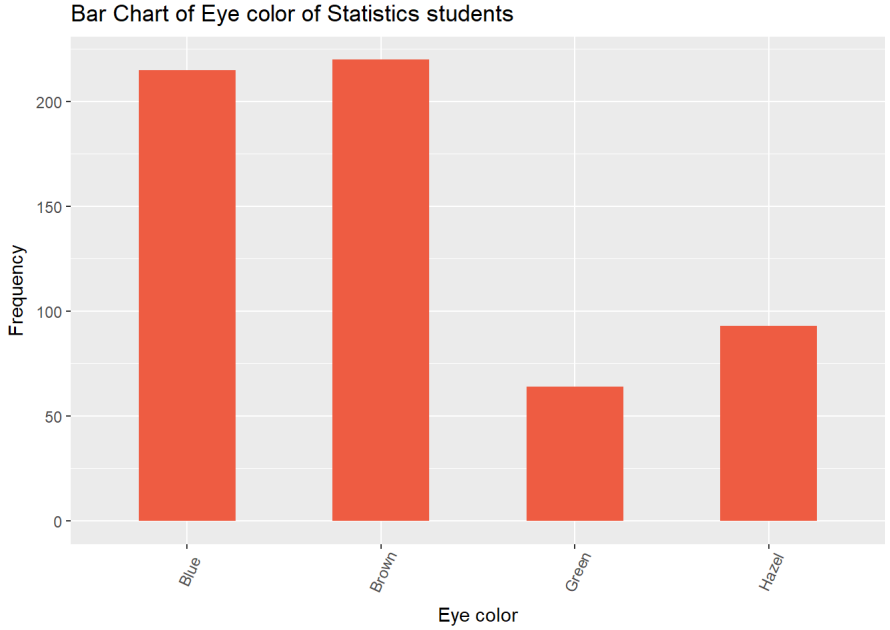
Why the bar chart?

Bar charts are incredibly good at retaining the information of the dataset accurately. However, due to its nature of being all bars with often times the same color readers have a hard time finding what object to focus on and what the key points to notice in the chart are. Nevertheless, if the data is organized in a matter that helps factilitate what should be noticed. For example, organizing the data in a descending matter might help viewers first notice the maximum frequency, but fail to highlight just how big of a difference the maximum and minimum are. While bar charts are accurate,they have a harder time displaying proportionality between the observation and total frequency.

Recreating the bar chart

The bar chart can be easily created with `ggplot2` using their basic `ggplot()` function. One can easily convert the histogram plotted with the ggplot function by changing some of the arguments. The key changes are that in the `geom_bar` portion the `stat` is not set to the default count, but instead set to `identity` so that it compares the X and Y values instead of just counting the X variable. Similarly you need to provide both the `x` and `y` variable inside the `aes` function call. (X must be the observed options and y must be the frequency). The remaining things are just minor tweaks such as how fat the bars are using `width`, and the fill color using `fill`.

```
g <- ggplot(holder, aes(labels, colorfreq))
g + geom_bar(stat="identity", width = 0.5, fill="tomato2") +
      labs(title="Bar Chart of Eye color of Statistics students", x = "Eye color", y = "Frequency") +
      theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



What's important is to realize that **good** data visualiziation heavily relies choosing the correct format of representing the data. Data visualization becomes even more important when there are not support statistics such as averages, standard deviations, correlations, which is the case for categorical variables. There is no univresal best chart type for categorical variables, so it's very important in spending time to find one that represents the data accurately but also is engaging and highlights the important things to notice.

Resources

Top 50 ggplot2 plots

HairEyeColor built in dataset for R

Using the waffle package

When to use the Waffle Chart

Pie Charts in GGplot

Strengths of Pie Charts

Plotrix package for 3D Pie Charts

Other ways to visualize categorical variables