post01-cole-stern

Cole Stern

10/23/2017

Playing With Color in R:

Making Your Data Beautiful & Informational

So far in our exploration of R, we have only slightly played around with aesthetic manipulations and their use in data visualization. And while there is an innate simplicity in dealing with the various shades of grey that dominate base R, there is a proper time and place where incorporating color can better highlight the important parts of your data and convey this information to the reader. Therefore, in this blog post I will be exploring the ways in which we can use color to heighten the value of our data, and hopefully even add some pop and personality!

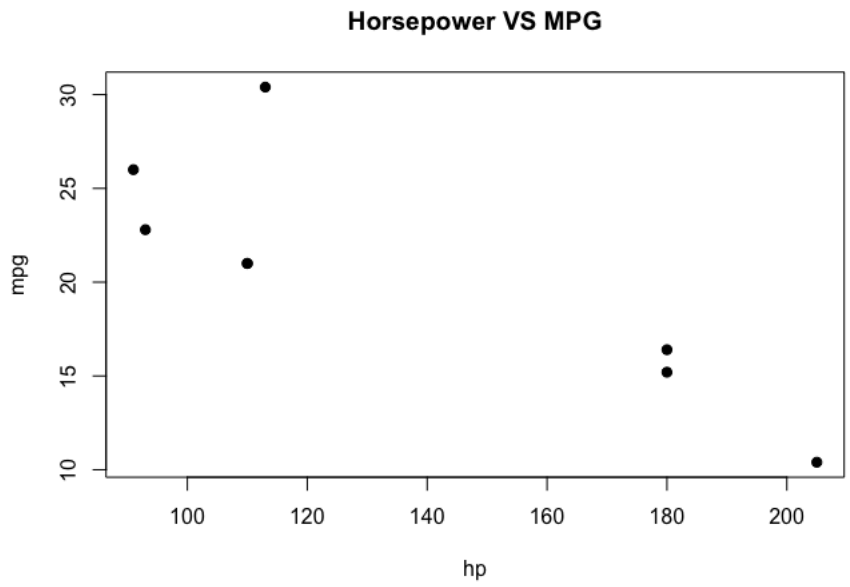
Basic Colors in R

For demonstration purposes, I will be using the pre-installed data set `mtcars` that is available to all R uses without having to import a separate data set. That said I will just be selecting a random sample of eight vehicles so that the data will be more easily visualized. I will call this data set `sdat`.

```
sdat <- mtcars[sample(nrow(mtcars), 8), ]
```

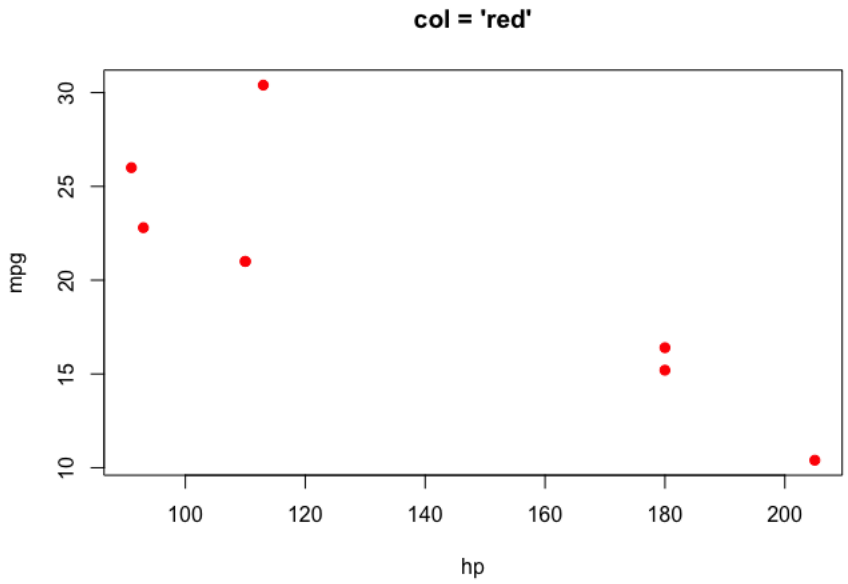
Now I will be using `plot()` from the base package `graphics` to crease a simple scatterplot charting mpg vs horsepower. (Note that I used the argument `pch = 19`. This fills in the circle so that its easier to distinguish the color. I suggest you use this arguments for your graphs.)

```
plot(mpg ~ hp, sdat, main = "Horsepower VS MPG", pch = 19)
```

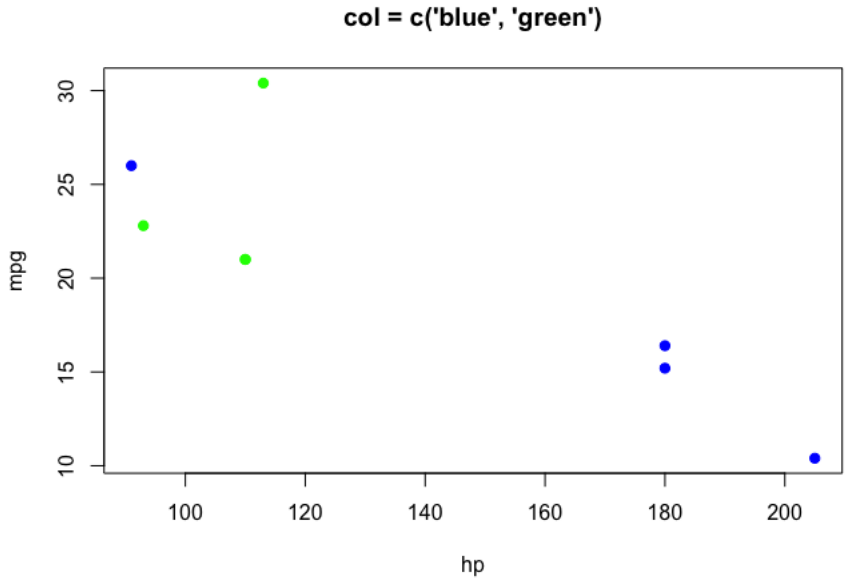


You can now color in your points by supplying a character vector with one or more color names. If you need a color for 8 points and you input fewer, recycling will step in to color the remaining points. Here's a couple examples for how you specify the color of the points using the argument `col =`.

```
plot(mpg ~ hp, sdat, main = "col = 'red'", pch = 19, col = "red")
```



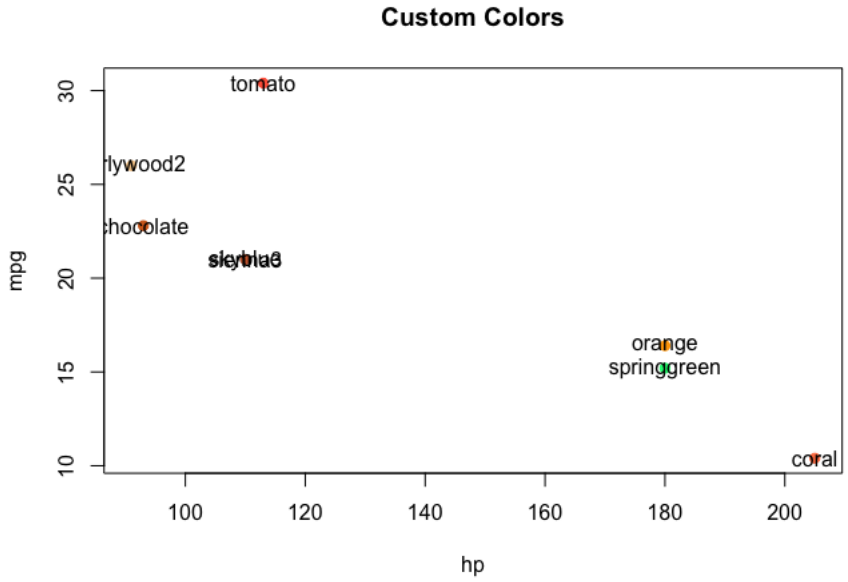

```
plot(mpg ~ hp, sdat, main = "col = c('blue', 'green')", pch = 19, col = c("blue", "green"))
```



One cool option in R is to create your own vectors of colors. To keep things more organized, I will first store my custom colors as an object, and then call that object within the plot calls. This makes it **much** easier to adjust the colors later on if you want to revisit.

```
mycolors <- c("orange", "skyblue", "burlywood2", "chocolate", "springgreen", "tomato", "coral", "sienna3")
plot(mpg ~ hp, sdat, main = "Custom Colors", pch = 19, col = mycolors)

#using with to add labels to each individual data point
with(sdat, text(x = hp, y = mpg, labels = mycolors))
```



At this point you are probably asking yourself, “How would I know that there is a color called burlywood2”? The answer is that you can see all the names of the pre-loaded colors in R using `colors()`.

```
head(colors(), 10)
```

```
##  [1] "white"          "aliceblue"      "antiquewhite"   "antiquewhite1"
##  [5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
##  [9] "aquamarine1"    "aquamarine2"
```

```
tail(colors(), 10)
```

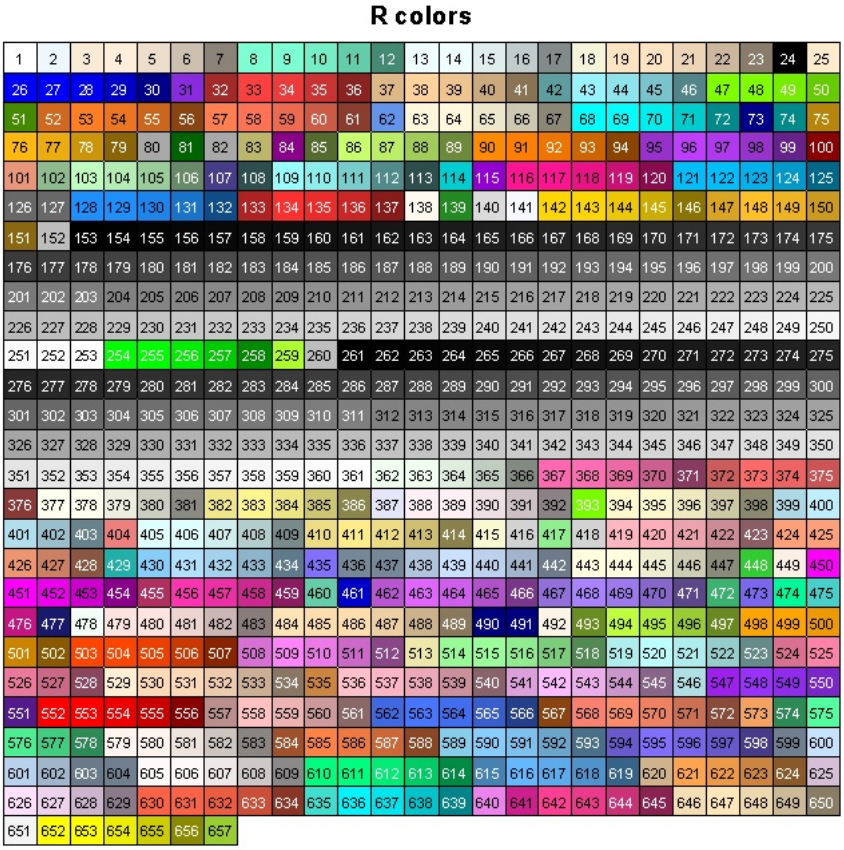
```
##  [1] "wheat2"       "wheat3"       "wheat4"       "whitesmoke"   "yellow"
##  [6] "yellow1"      "yellow2"      "yellow3"      "yellow4"      "yellowgreen"
```

```
length(colors())
```

```
## [1] 657
```

To see all the colors organized by multiple methods, I suggest you visit Earl F. Glynn’s site which is devoted to analyzing all the base colors in R, and possible ways of sorting and organizing them.

Just to give you a taste, here’s an image displaying all 657 colors!



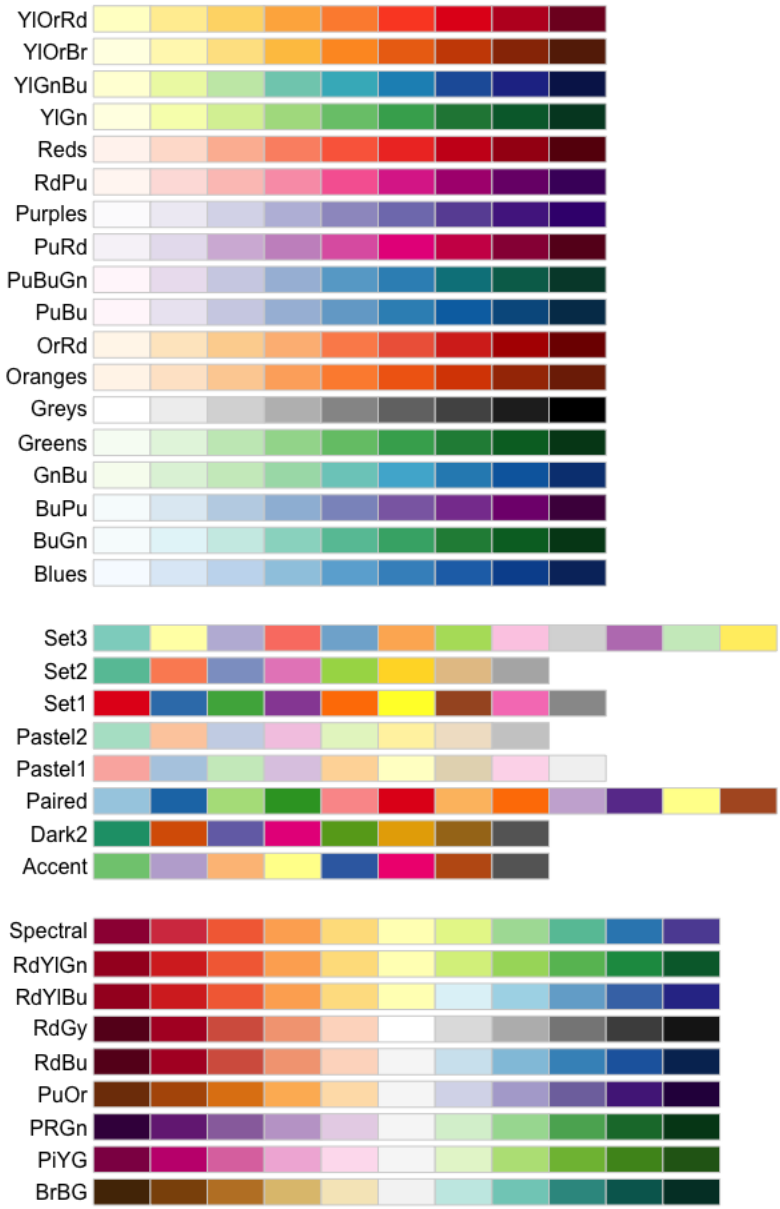
RColorBrewer

One of the best things about R is the thriving community of users who use and also contribute to the incremental development of the language. One of such people is Cynthia Brewer, a professor at Pennsylvania State University, who has worked to develop a more comprehensive color palette for print and the web by way of the add-on package `RColorBrewer`. In order to explore the package you will have to download and then install it.

```
#loading the package RColorBrewer
library(RColorBrewer)
```

Now we can look at all the contents of the package.

```
display.brewer.all()
```



We can see that there are different color palettes in the package, falling into three distinct classes.

1. This first class is called sequential. What this means is that these colors are best used to show low to high ratings or intensity. For example, these colors could show the concentrations of chemicals across certain solutions or maybe you could use them to display p-values ranging from 0 to 1 (light to dark) of a hypothesis.
2. The second class is called qualitative. These are good colors for things that are not concerned with order, for qualitative variables like car types or animal species. The one special case in this class is the “Paired” palette. One case where this palette could be useful is for experimental data that has a control and treatment variable, with each taking a particular shade.
3. The last class is called diverging. These are good for visualizing things that are on opposite ends of a scale, e.g. z-scores.

Now we can view a single palette by executing the code below:

```
display.brewer.pal(n = 8, name = "Set1")
```

Set1 (qualitative)

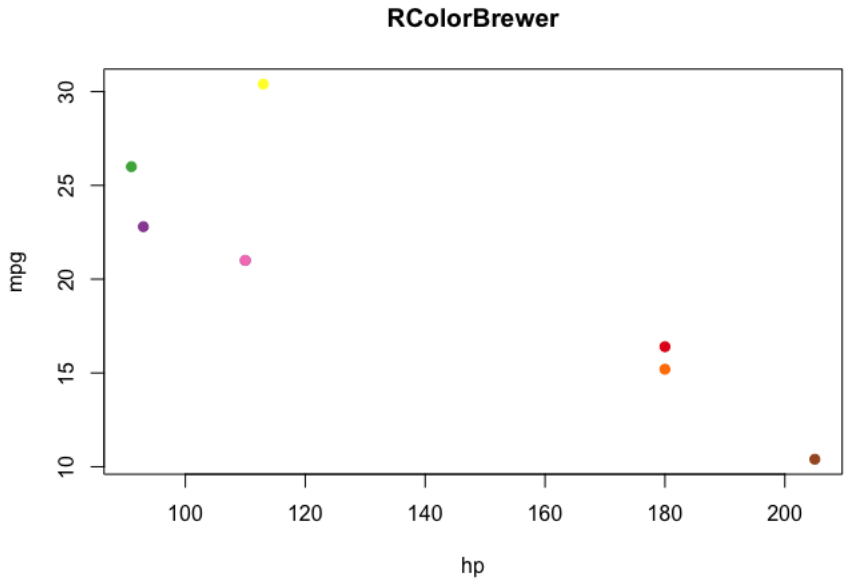
In case you're wondering about the arguments, n is specifying the number of colors in the palette that you want to pull from.

> Note that the minimum is 3 and the maximum is the number of colors in the palette. Additionally you need to specify the name of the palette you are selecting.

Also you can see from what class the colors are from. In this case, Set1 is qualitative which will work well for the `mtcars` data set.

From here, we can explore using the `RColorBrewer` package to color our scatterplot instead of using a vector of colors where you had to individually choose.

```
rbrewercolors <- brewer.pal(n = 8, name = "Set1")
plot(mpg ~ hp, sdat, main = "RColorBrewer", pch = 19, col = rbrewercolors)
```



Hexadecimal Color Specification

So now you're probably more than content (and maybe a little overwhelmed) with the amount colors that you can choose from. However, we have only scraped the surface of the colors you can use in R!

What you haven't been seeing is that all the colors we have been playing with have been stored in a format of hexadecimal values. The readers digest explanation is that hexadecimal values were created to specify standardized color values for HTML web pages.

The color is stored as a 6 digit number of the form `#RRGGBB`. `RR` codes for red, `GG` for green, and `BB` for blue, and the value that you can input range from `00` to `FF`.

Just to prove this to you, let us call the same `RColorBrewer` palette we used for the previous example but instead of displaying the colors we will look at their encoded hexadecimal values.

```
brewer.pal(n = 8, name = "Set1")
```

```
## [1] "#E41A1C" "#377EB8" "#4DAF4A" "#984EA3" "#FF7F00" "#FFFF33" "#A65628"
## [8] "#F781BF"
```

By now you're head should be spinning, and with good reason as this hexadecimal format allows for ***16,777,216*** different colors! If you want to look over these colors, there are plenty of color picking sites online to help, click here to be taken to one.

Wrapping Up

I hope that this lesson has made you more interested in using colors in R. While we have learned that the extraneous use of color can hurt more than help, I still believe that there is a proper use for it which can help elevate your data. If there is one piece of advice I could give, I would say don't be afraid to explore. Google is your best friend, and most likely if you have a question, it has been tackled (or at least asked) on any number of forums.

So go out and make your data beautiful!

References

- http://rstudio-pubs-static.s3.amazonaws.com/2852_379274d7c5734f979e106dcf019ec46c.html

> Good source for plotting and learning how to edit the settings of visualizations.

- http://www.stat.cmu.edu/~cshalizi/rmarkdown/

- https://www.r-bloggers.com/color-palettes-in-r/

> Describing the color palettes available in base R.

- http://www.sthda.com/english/wiki/colors-in-r

> Another source for detailing the base colors in R. Also details the `RColorBrewer` package.

- https://www.programiz.com/r-programming/color

> Provides in depth descriptions of hexadecimal color format.

- http://research.stowers.org/mcm/efg/R/Color/Chart/

> Earl F. Glynn's site where I pulled the image for the base R colors

---

- https://en.wikipedia.org/wiki/Web_colors

> More information on the hexadecimal color format.