# Post 1: Data Visualization and Choosing a ColorBrewer Palette from an Interactive App

## Introduction and Motivation:

The purpose of this post is to highlight a very cool interactive way to help you choose a color palette when you're working on a visualization. I like to compare this app to going to the paint store to find swatches of colors you can take home with you when when you're thinking about repainting your bedroom or kitchen. It is super useful to have examples of the different color options in front of the blank wall at home and in our case here, equally important to have color options readily visible when you're working on an R visualization!

`palette_explorer()` is basically like bringing swatches home with you so you can find the best color fit! (It's also just as fun) :)

This post will mainly discuss the how to use this interactive app, what each element of the app does and what it means for your data visualization process in general.

My motivation for writing about this topic particularly is that up until now, you may not have known about this tool. It will come in handy next time you want to add some color to your plots and charts, and it will free up some of the time it would've otherwise taken you to do so! What's better than saving time?

(I can answer that)… nothing is.

## Background:

When I'm working on a plot, or a bar chart or any other type of visualization, I usually stick to one color or maybe two colors for different aspects of my plot. Especially if its for my own purposes. Or I search for a palette online. This is great but its hard to remember which colors each palette contains. I know "BuPu" is a purple-ish color scheme (because it's my favorite) but other than that I only know, "Spectral", and "Set1" off hand.

That's boring, and its limited!

Just because I'm a newb when it comes to R, doesn't mean I have to stay that way!

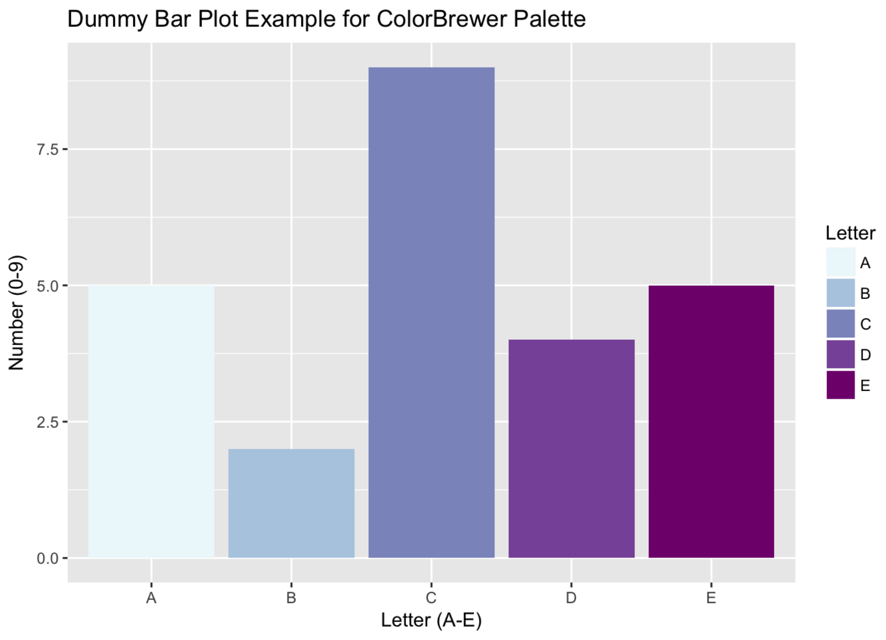
I thought to myself, "There has to be a better way!" so I did some looking around, and came across the `palette_explorer()` tool given by the "tmaptools" package written by Martijn Tennekes of the Netherlands.

Martijn developed these tools to help make spatial data analysis in R more fluid. Consequently, function tools in this package also helps make creating other types of visualizations more fluid. Using `palette_explorer()` cuts the time it takes to find an appealing ColorBrewer palette for your plot or graph in half! Woo hoo!

## Examples:

Here's a general code example of using ColorBrewer to give color to a very simple ggplot bar plot:

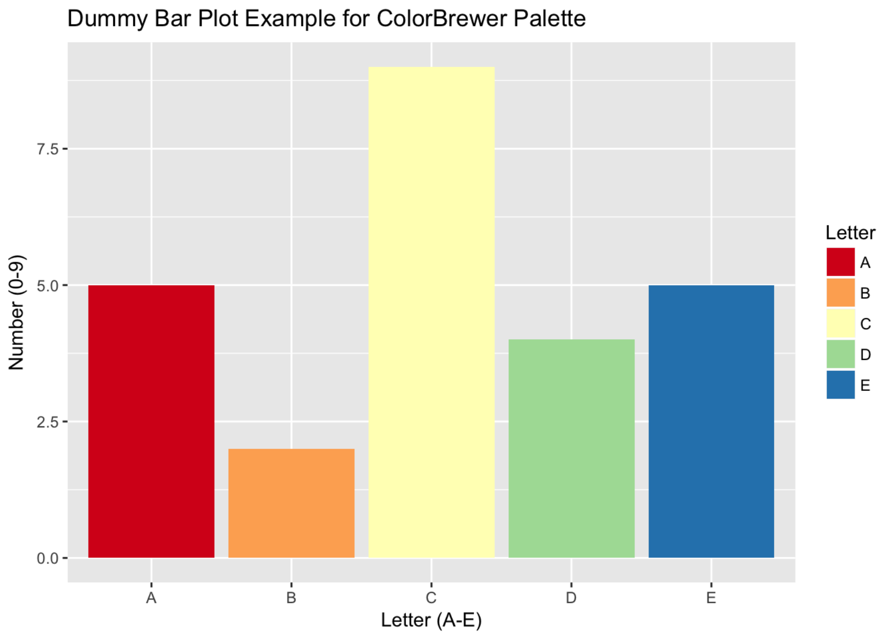
```r
##Library call the necessary packages
library(RColorBrewer)
library(ggplot2)

##Create the data frame from which your ggplot will be made
df.dummy_data <- data.frame(category_var = c("A","B","C","D","E"), numeric_var = c(5,2,9,4,5))

## Make Letter for side bar area of bar plot
Letter <- df.dummy_data$category_var

##Make your ggplot
dummy_plot <- ggplot(data=df.dummy_data, aes(x=category_var, y=numeric_var, fill = Letter)) + geom_bar(stat="ident
ity") + xlab("Letter (A-E)") + ylab("Number (0-9)") + ggtitle("Dummy Bar Plot Example for ColorBrewer Palette") +
scale_fill_brewer(palette = "BuPu")

dummy_plot
```



This is great! Its colorful and nice to look at. However we only know the palette “BuPu” because we looked it up one day and happened to remember it.

We could use the same plot and try a different palette that we happen to remember:

```
## Library call the necessary packages
library(RColorBrewer)
library(ggplot2)

## Create the data frame from which your ggplot will be made
df.dummy_data <- data.frame(category_var = c("A","B","C","D","E"), numeric_var = c(5,2,9,4,5))

## Make letter for side bar area of bar plot
Letter <- df.dummy_data$category_var

## Make your ggplot
dummy_plot_2 <- ggplot(data=df.dummy_data, aes(x=category_var, y=numeric_var, fill = Letter)) + geom_bar(stat="ide
ntity") + xlab("Letter (A-E)") + ylab("Number (0-9)") + ggtitle("Dummy Bar Plot Example for ColorBrewer Palette")
+ scale_fill_brewer(palette = "Spectral")

dummy_plot_2
```



Also pretty cool!

But as I mentioned above, wouldn’t it be awesome if we could just pick from a descriptive list without having to pull up a search bar and scour the internet for something useful?

Yes. It would be awesome.

This example goes there:

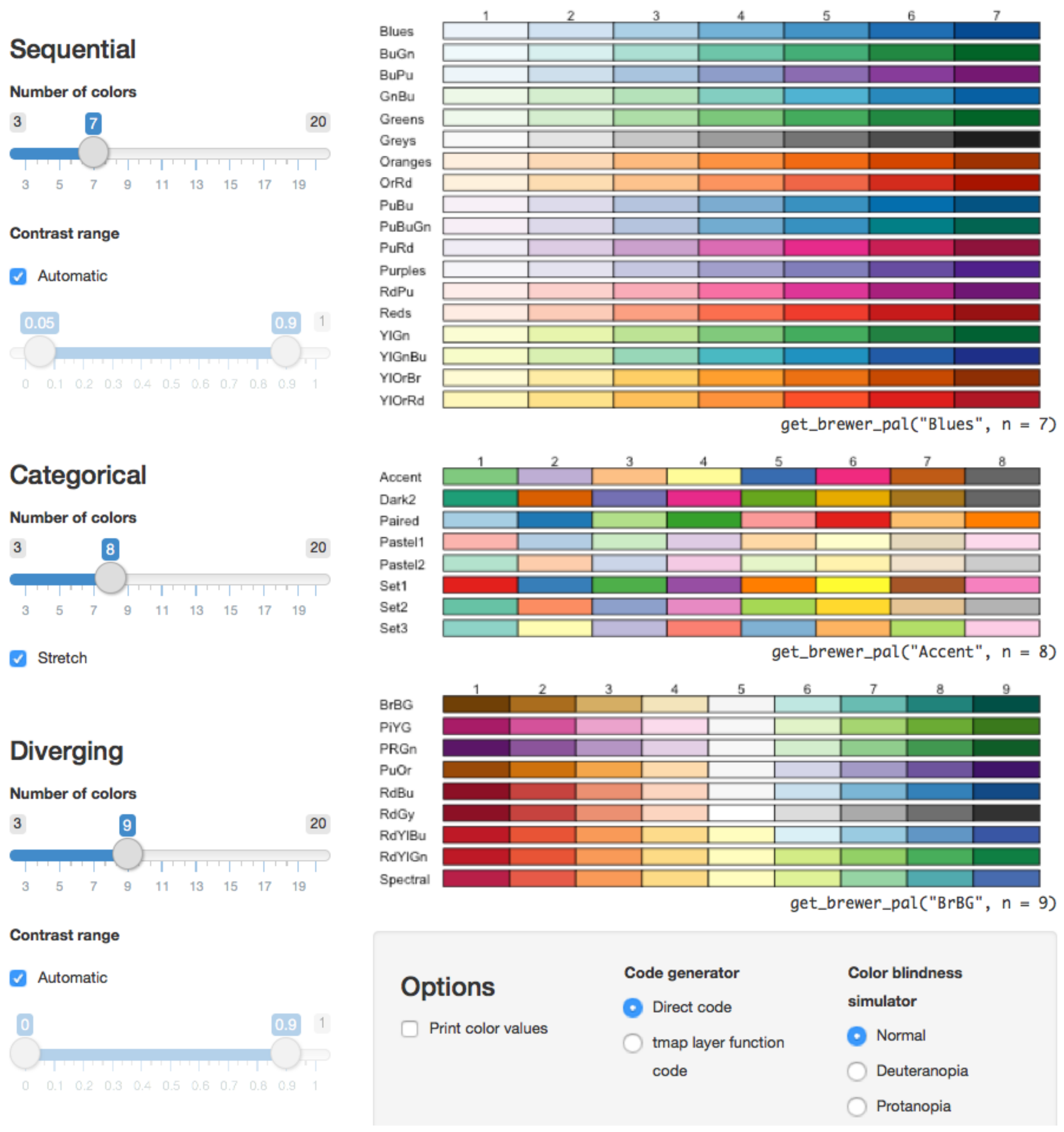
```
## To use this interactive app you need to install the tmaptools package
## install.packages(tmaptools)

## Same as the last two examples you need to make a few library calls. This example calls tmaptools as well
library(RColorBrewer)
library(ggplot2)
library(tmaptools)
library(rsconnect)
```

```
## Warning: package 'rsconnect' was built under R version 3.4.1
```

```
## Do the same thing and create the data frame from which your ggplot will be made

df.dummy_data <- data.frame(category_var = c("A","B","C","D","E"), numeric_var = c(5,2,9,4,5))

## Make Letter for side bar area of bar plot

Letter <- df.dummy_data$category_var

## palette_explorer()

## ^ This is commented out and in place of embedding a shiny app into this html document, I've taken a screen shot
of the app shown below.
```

## Visuals:

Here is a look at what pops up when you use `palette_explorer()`:



**Sequential**

Number of colors: 3 – [7] – 20

Contrast range: [✓ Automatic] 0.05 – 0.9 (0 – 1)

```
get_brewer_pal("Blues", n = 7)
```

**Categorical**

Number of colors: 3 – [8] – 20

[✓ Stretch]

```
get_brewer_pal("Accent", n = 8)
```

**Diverging**

Number of colors: 3 – [9] – 20

Contrast range: [✓ Automatic] 0 – 0.9 (0 – 1)

```
get_brewer_pal("BrBG", n = 9)
```

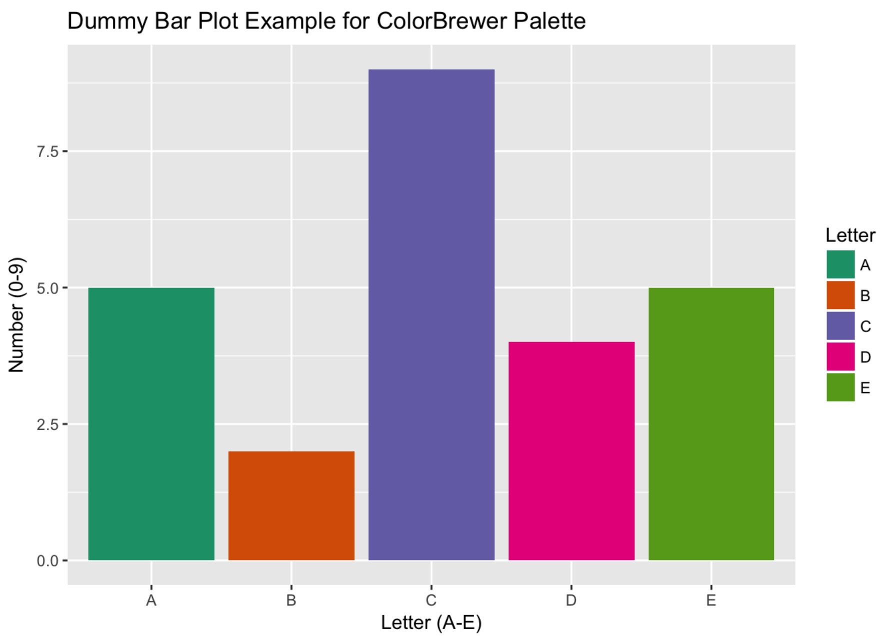
**Options**

Print color values [ ]

Code generator: [✓ Direct code] tmap layer function code

Color blindness simulator: [✓ Normal] Deuteranopia Protanopia

```
## Make your ggplot with your newly selected palette!
dummy_plot_3 <- ggplot(data=df.dummy_data, aes(x=category_var, y=numeric_var, fill = Letter)) + geom_bar(stat="ide
ntity") + xlab("Letter (A-E)") + ylab("Number (0-9)") + ggtitle("Dummy Bar Plot Example for ColorBrewer Palette")
+ scale_fill_brewer(palette = "Dark2")

dummy_plot_3
```



## Discussion:

Now that you've seen palette_explorer in action, I'll discuss what each part of the app:

There are 3 main sections on the left hand side of the app. They are the categories for what type of palette shows on they right.

- Sequential: Gives us a palette whose colors get darker as the data becomes more dense or frequent. (think election maps with light blue to dark blue democratic states and light red to dark red republican states)
- Categorical: Gives us a palette whose colors are distinct from one another so that the viewer can easily tell the categories of the plot apart.
- Diverging: Gives us a palette whose colors are distict at each far end but start to blur towards the center. (Think about a credit score simulator slider bar with one end red to represent a low score and one end green to represent a high score)

Each of these sections has a little subsection called 'Number of Colors' where you can use the shiny slider to select the number of different colors you want to see in the palette selections to match your visualization. The code is displayed beneath the palatte on the bottom right. You will notice that it changes depending on your selections.

Under both the 'Sequential' and 'Diverging' labels you will find a button to turn automatic contrast on or off as well as a slider which you can adujst when the automatic option is deselected. This is nice when you want to use softer hues for your plot.

## Conclusions:

Aside from this being a very convenient and easy to use tool, one aspect of `palette_explorer()` is the option to choose colors that very thoughtfully work around color blindenss. You can select any of 3 different options aside from the option for 'normal' that will give you a palette based on whether you are wanting to make your visualizations readable for those with either Tritanopia (yellow-blue color blindess), Protanopia (red-green color blindness), or Deuteranopia (confusion of greens, reds, and yellows).

Considering that about 1 in 12 men and 1 in 200 women suffer some type of color blindness this is a fantastic feature to keep in mind if you ever have an audience you know may be color blind.

Imagine the time and effort it might take to understand how to construct a palette to accomodate color blindness!

So the take home message here is that this is a great resource to use when you'd like to spice up your visualizations and it is a really great example/application of how shiny app can be useful. Which is relevant considering we are orbiting the shiny apps world in class right now.

## References:

- http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3
- https://cran.r-project.org/web/packages/tmaptools/tmaptools.pdf
- http://mkweb.bcgsc.ca/brewer/
- https://nei.nih.gov/health/color_blindness/facts_about
- http://www.colourblindawareness.org/colour-blindness/
- https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/colorPaletteCheatsheet.pdf
- https://www.youtube.com/watch?v=t1To4I5oCrk