How to make your own color palettes in ggplot

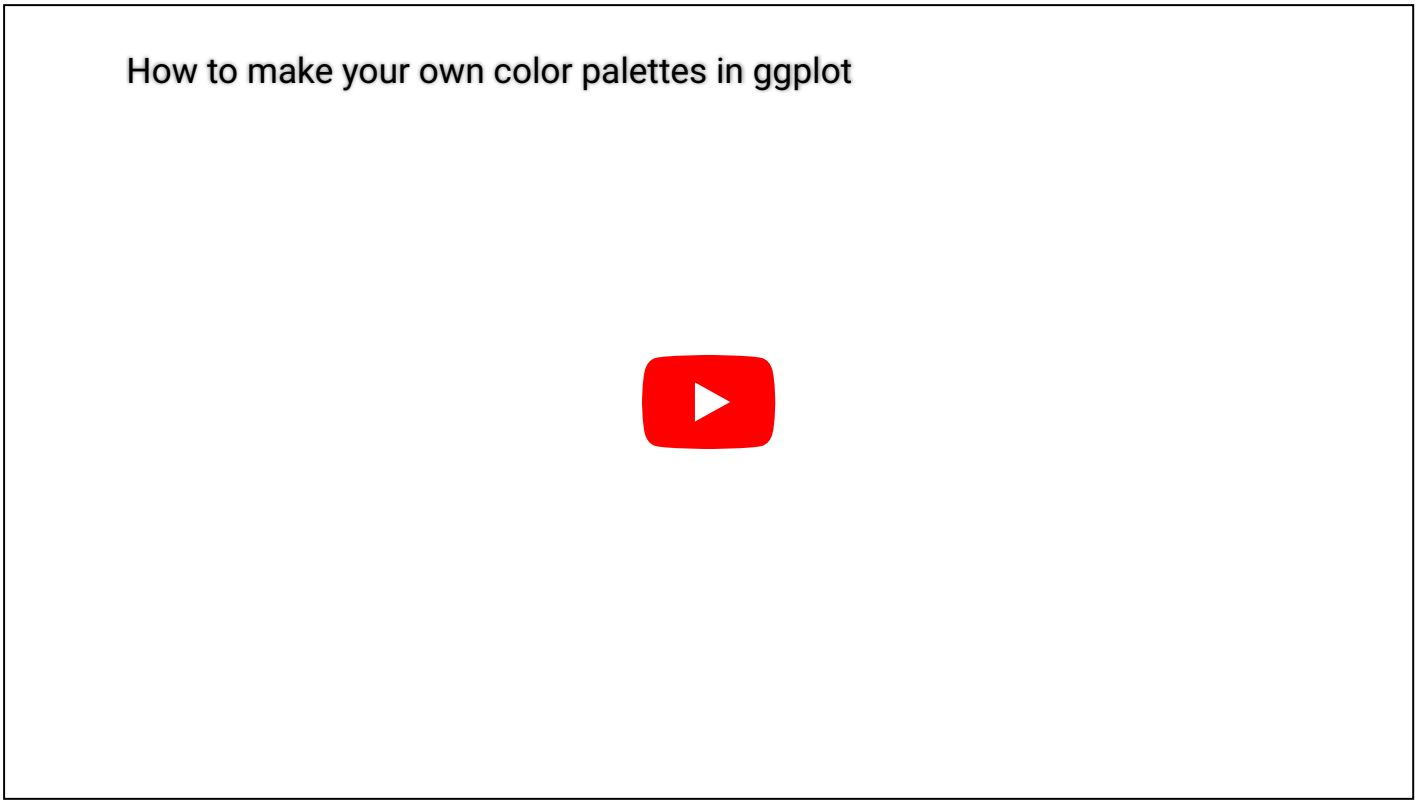
David
January 23, 2023

0 Comments

One of the great things about creating data viz with ggplot is that you can create color palettes that match your or your clients' branding. We've written about how we do this in our consulting work. But the way we work is fairly complicated and I was asked recently for a simpler solution to making custom ggplot color palettes.

I recorded a video to show how to make three different types of palettes

- Qualitative (i.e. categorical)
- Sequential (going from a low to a high value)
- Diverging (going from low to high with a midpoint)



As you'll see in the video, there are a few steps to this process. I begin by defining colors that I want to use:

```r
# Define Colors ---------------------------------------------------------

ga_purple <- "#8359AB"
ga_yellow <- "#FFDE39"
ga_gray <- "#827C78"
ga_blue <- "#49B8F1"
ga_brown <- "#B88262"
ga_pink <- "#DC458E"
```

define-colors.R hosted with ❤️ by **GitHub** view raw

View this gist on GitHub

Qualitative Color Scale

To make a qualitative color scale, I used the colors I defined in combination with the `scale_color_manual()` function.

```r
# Qualitative -----------------------------------------------------------

scale_color_ga_qualitative <- function() {

  scale_color_manual(values = c(ga_blue,
                                ga_pink,
                                ga_yellow,
                                ga_purple,
                                ga_gray,
                                ga_brown))

}
```

qualitative-scale.R hosted with ❤️ by **GitHub** view raw

View this gist on GitHub

Sequential Color Scale

Next, I made a sequential color scale. I adapted the `scale_fill_gradient()` function to make this function.

```r
scale_fill_ga_sequential <- function(low_color = ga_yellow,
                                     high_color = ga_purple) {

  scale_fill_gradient(low = low_color,
```

```
                        high = high_color)

}
```

sequential-scale.R hosted with ❤️ by GitHub view raw

View this gist on GitHub

Diverging Color Scale

Finally, I made a diverging color scale. To do this, I adapted the `scale_fill_gradient2()` function, which gives me the ability to set a medium color, in addition to the high and low colors.

```
scale_fill_ga_diverging <- function(low_color = ga_yellow,
                                    medium_color = "white",
                                    high_color = ga_pink) {

  scale_fill_gradient2(low = low_color,
                       mid = medium_color,
                       high = high_color)

}
```

diverging-scale.R hosted with ❤️ by GitHub view raw

View this gist on GitHub

Full Code

I've included the full code used in the video below. You can see where I get the data for the map of North Carolina that I use and run the code yourself to see how it works.

```
# Load Packages ------------------------------------------------------------

library(tidyverse)

# Define Colors ------------------------------------------------------------

ga_purple <- "#8359AB"
ga_yellow <- "#FFDE39"
ga_gray <- "#827C78"
ga_blue <- "#49B8F1"
ga_brown <- "#B88262"
ga_pink <- "#DC458E"

```

```

# Qualitative ------------------------------------------------------------

scale_color_ga_qualitative <- function() {

  scale_color_manual(values = c(ga_blue,
                                ga_pink,
                                ga_yellow,
                                ga_purple,
                                ga_gray,
                                ga_brown))

}

palmerpenguins::penguins %>%
  ggplot() +
  geom_point(aes(x = bill_length_mm,
                 y = flipper_length_mm,
                 color = island)) +
  labs(title = "Palmer Penguins",
       subtitle = "Look at them go!",
       x = "Bill length",
       y = "Flipper length") +
  theme_minimal() +
  scale_color_ga_qualitative()


# Download Data ----------------------------------------------------------

library(tigris)

# Downloaded from https://github.com/tonmcg/US_County_Level_Election_Results_08-20
presidential_returns_by_county <- read_csv("https://github.com/tonmcg/US_County_Level_Election_Resu

# Done with tigris package
us_counties <- counties()

# Merge datasets
nc_presidential_returns_by_county <- us_counties %>%
  left_join(presidential_returns_by_county,
            by = c("GEOID" = "county_fips")) %>%
  filter(state_name == "North Carolina") %>%
  mutate(county_name = str_remove(county_name, " County")) %>%
  select(county_name, contains("votes"), per_point_diff)

```

```r
# Sequential ------------------------------------------------------------------

scale_fill_ga_sequential <- function(low_color = ga_yellow,
                                     high_color = ga_purple) {

  scale_fill_gradient(low = low_color,
                      high = high_color)

}

nc_presidential_returns_by_county %>%
  ggplot(aes(fill = total_votes)) +
  geom_sf() +
  theme_void() +
  scale_fill_ga_sequential()

nc_presidential_returns_by_county %>%
  ggplot(aes(fill = total_votes)) +
  geom_sf() +
  theme_void() +
  scale_fill_ga_sequential(low_color = ga_purple,
                           high_color = ga_yellow)


# Diverging -------------------------------------------------------------------

scale_fill_ga_diverging <- function(low_color = ga_yellow,
                                    medium_color = "white",
                                    high_color = ga_pink) {

  scale_fill_gradient2(low = low_color,
                       mid = medium_color,
                       high = high_color)

}

nc_presidential_returns_by_county %>%
  ggplot(aes(fill = per_point_diff)) +
  geom_sf() +
  theme_void() +
  scale_fill_ga_diverging()
```

A Couple Caveats

This post has shown *how* to make custom color scales. What I haven't done is talked about choosing good colors for you color scales. If you want to read more on that, check out:

- This blog post written by R for the Rest of Us consultant Cara Thompson
- The Glamour of Graphics course, which has an entire section on color
- This incredibly comprehensive article by Lisa Charlotte Muth on choosing colors for data viz

The other important thing to note is that it's important to consider colorblindness when creating color palettes. Again, to keep things focused, I haven't done that in this blog post. However, there is a great R package called `colorblindr` to help you ensure your custom ggplot color palettes are accessible to all users.

Good luck making your own custom ggplot color palettes!

Have any questions? Put them below and we will help you out!

You must be logged in to post a comment.

Using Git and GitHub with R

Mapping with R

Data Cleaning with R

Package Development with R

Inferential Statistics with R