Template_Guide ggplot2

2024-04-22

Guide for using the template making script

```
source("Ggplot2TemplateMaker.R")
library(paletteer)
library(scales)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Parameters and Options

The parameters that are used are in relation to the Panel, legend and the titles/axes

Panel parameters

'background' - default to "black", other option is "white" to represent dark and light mode plots. Make sure to see color guide to choose colors compatible with dark/light mode you choose

'gridlines' - Default FALSE, if you would like gridlines change to TRUE

'grid_line_pattern' - Default NULL, if gridlines is TRUE must choose either 'solid', 'dotted', or 'dashed'

Legend parameters

'legend_location' - Default 'bottom', other options include all legened location options including but not limited to 'topright', 'top', 'bottomleft' etc. This will be highly custom depending on the plot you are creating.

Text/Title parameters

'font' - Default 'Times', ggplot2 contains many fonts the following are your options, however we will caution the use of fonts other than the ggplot2 default 'sans' and 'Times' or 'Helvetica': Short,Canonical mono,Courier sans,Helvetica serif,Times ,AvantGarde ,Bookman ,Helvetica-Narrow ,NewCenturySchoolbook ,Palatino ,URWGothic ,URWBookman ,NimbusMon URWHelvetica,NimbusSan ,NimbusSanCond ,CenturySch ,URWPalladio URWTimes,NimbusRom

'fontsize_title' - Default 15, should be roughly 2 times the size of axes titles

'fontsize_axes_title' - Default 8

'fontsize_axes' - Default 6, This is the font size for axes labels

'title_centered' - Default TRUE, if changed to FALSE title will be left justified.

'map_plot' - Defaults to FALSE, if you are creating a map, axes labels are unneeded and make plotting look busy. For this reason if you change this parameter to TRUE, then all axes labels are removed

Using the theme source code

When you are ready to set your panel theme for a project you may source the Template Maker first

```
source("Ggplot2TemplateMaker.R")
```

Call the function with your edits, here I will use the default template.

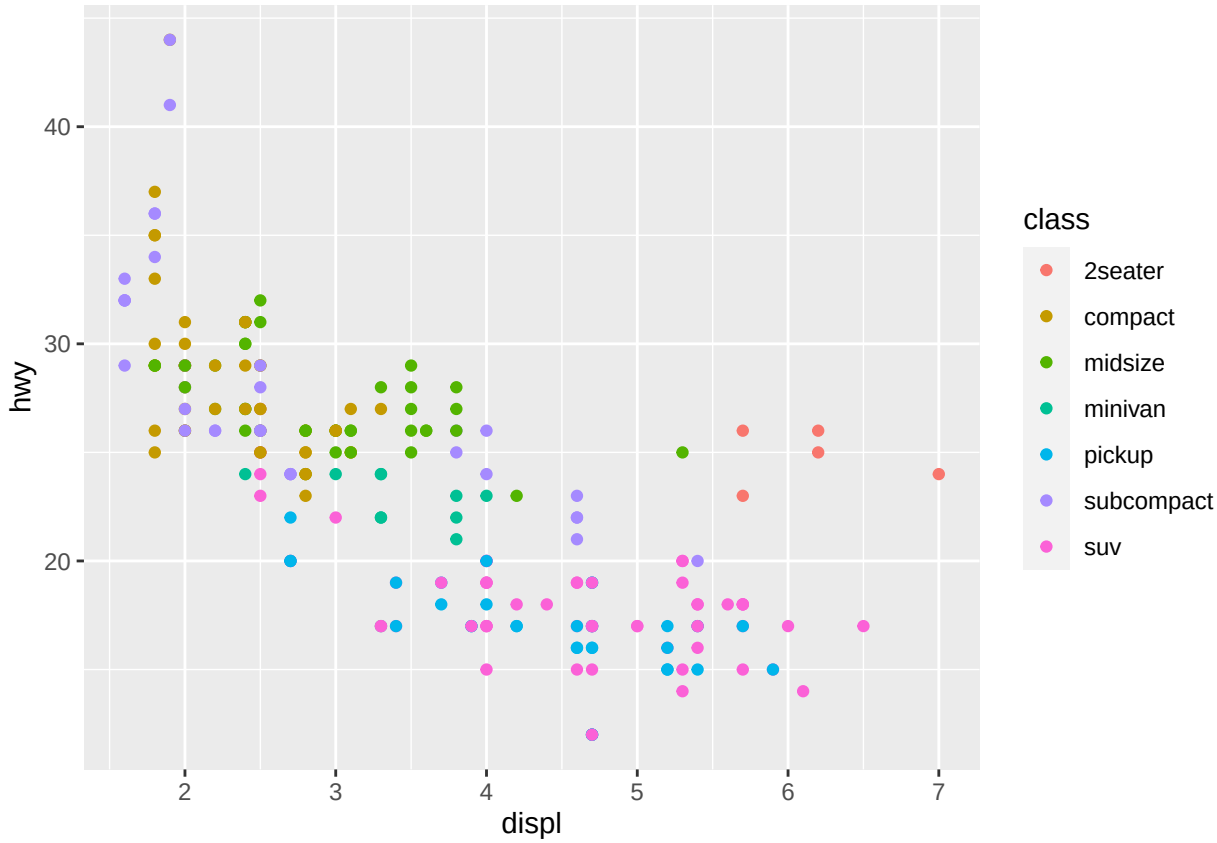
```
plot_theme(gridlines = TRUE, grid_line_pattern = "dotted")
```

```
## [1] "Theme Saved in Working Directory"
```

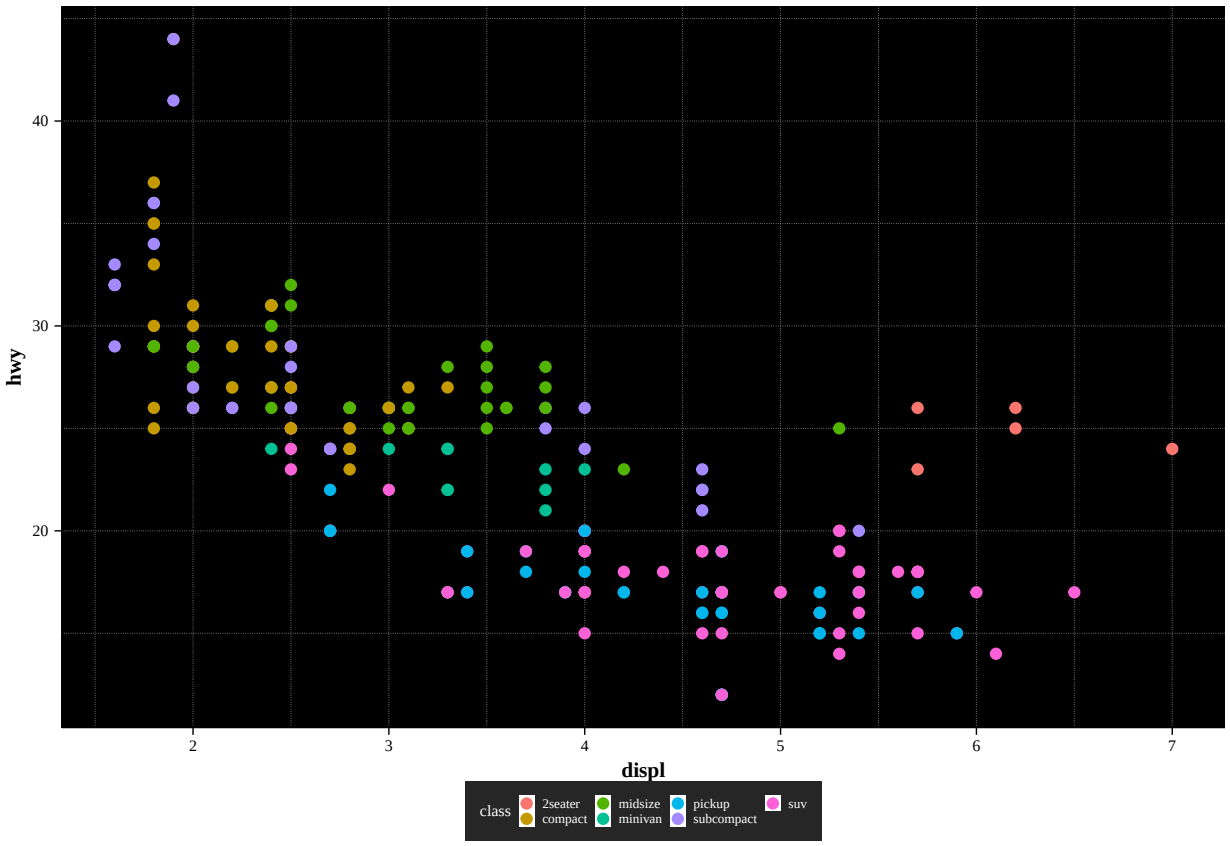
```
## [1] NA
```

Then when you want to use said theme for a ggplot you can read in with the following and use.

```
p = ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
p
```




```
themex <- readRDS("saved_theme.rds")
p + themex
```



Colors - How to adjust and color schemes we suggest

This template does not give colors, the way that ggplot2 works is that colors are added through the aes() called within the plot call. For the reason we simply suggest here color schemes and the packages they come from that would perhaps be of interest depending on the subject of your figure.

Please note that if you are using a dark mode, colors show up different;y, keep in mind dark colors will not show up well.

Be aware of colorblindness as well #### Using unambiguous palettes

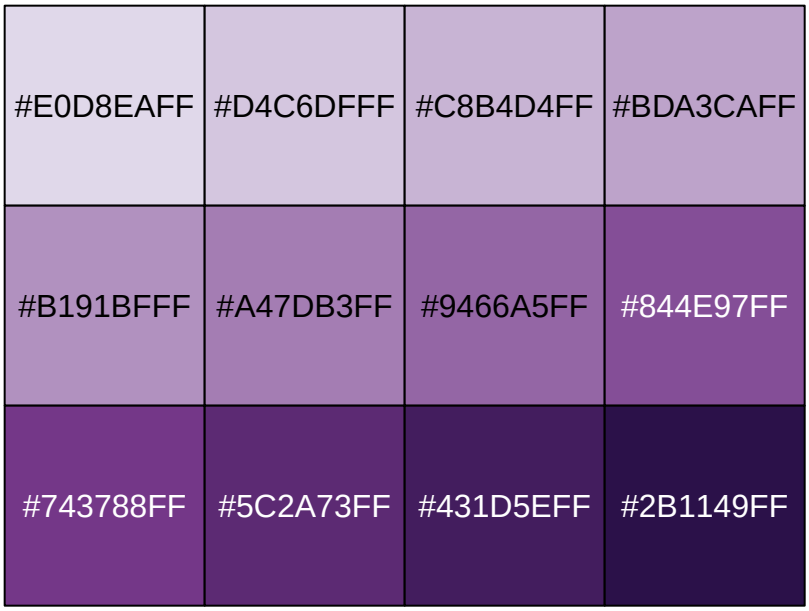
The easiest way to make color coding accessible to everyone, is using a palette, that is unambiguous to people with various types of color blindness. There are a few available:

- Masataka Okabe and Kei Ito have developed such a barrier free palette, and you can use it in R with the colorblind_pal() of the `ggthemes` package (also see colorblind_pal palette among the ggthemes palettes in the alphabetical list below) or by using the encoding provided by the Cookbook for R.
- Some of the palettes developed by Cynthia Brewer for the ColorBrewer are colorblind safe, you can find them through the palette chooser website's button "colorblind safe". In R, you can use the brewer palettes through `ggplot2`'s scale_colour_brewer() et al. or through the separate package `RColorBrewer`.

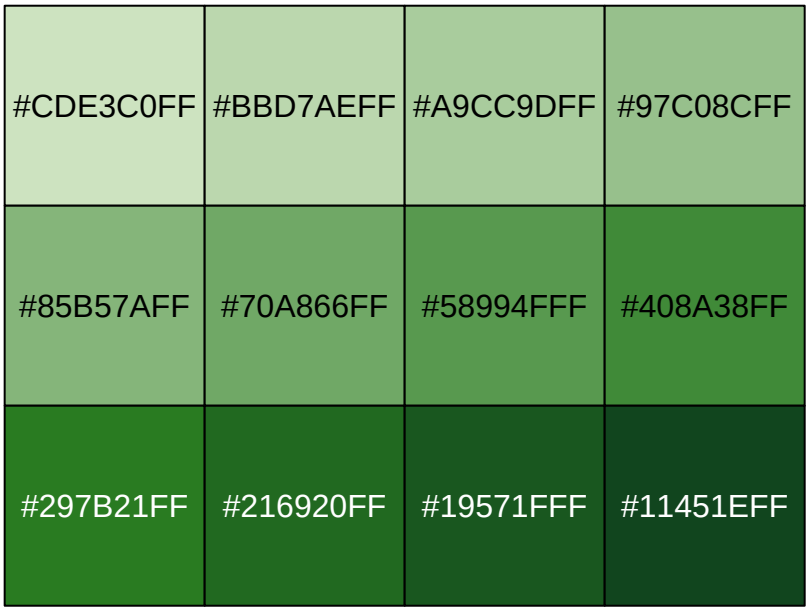
Within 'cartography' package there are many one color gradients that are useful for certain types of plots including the following:

Color - name of palette Purple - purple.pal Green - green.pal Orange - orange.pal Blue - blue.pal Pink - pink.pal Turquoise - turquoise.pal Red - wine.pal

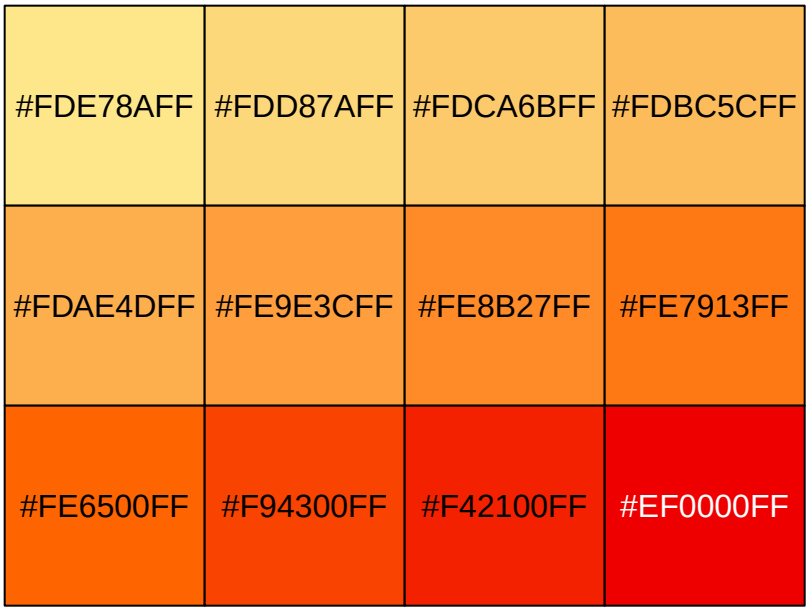
```
purple =paletteer_dynamic(`"cartography::purple.pal"`, n = 12)
show_col(purple)
```



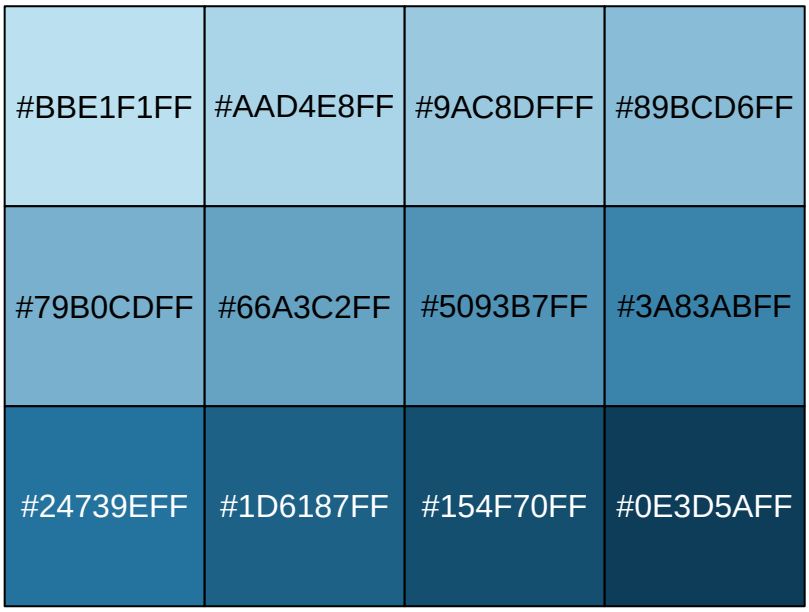

```
green = paletteer_dynamic(`"cartography::green.pal"`, n = 12)
show_col(green)
```



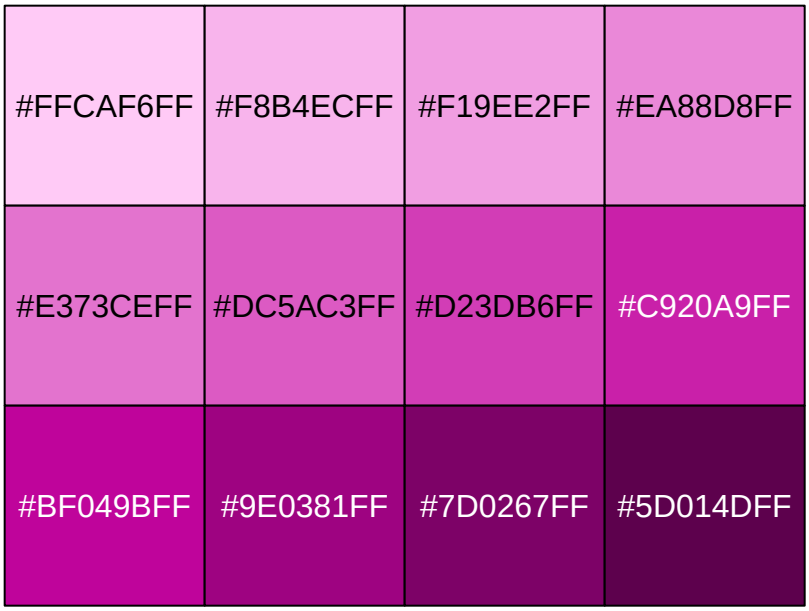

```
orange = paletteer_dynamic(`"cartography::orange.pal"`, n = 12)
show_col(orange)
```



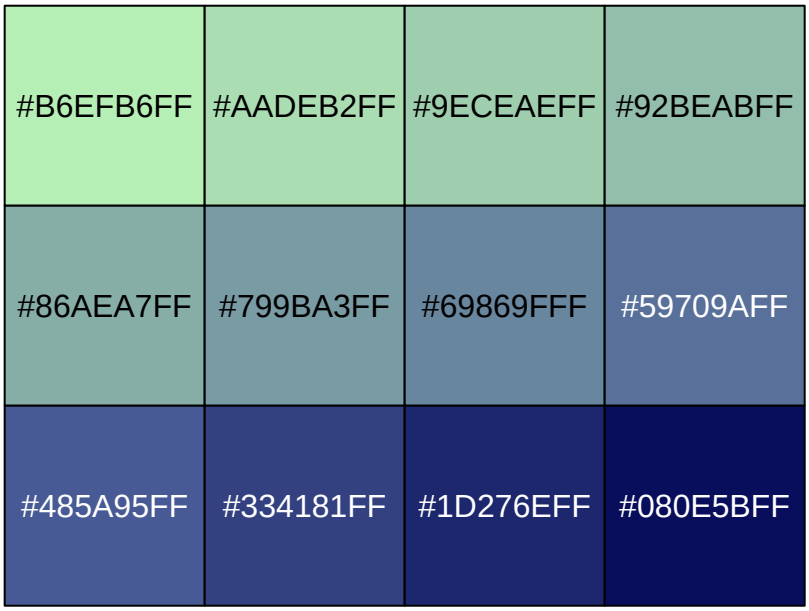

```
blue = paletteer_dynamic(`"cartography::blue.pal"`, n = 12)
show_col(blue)
```



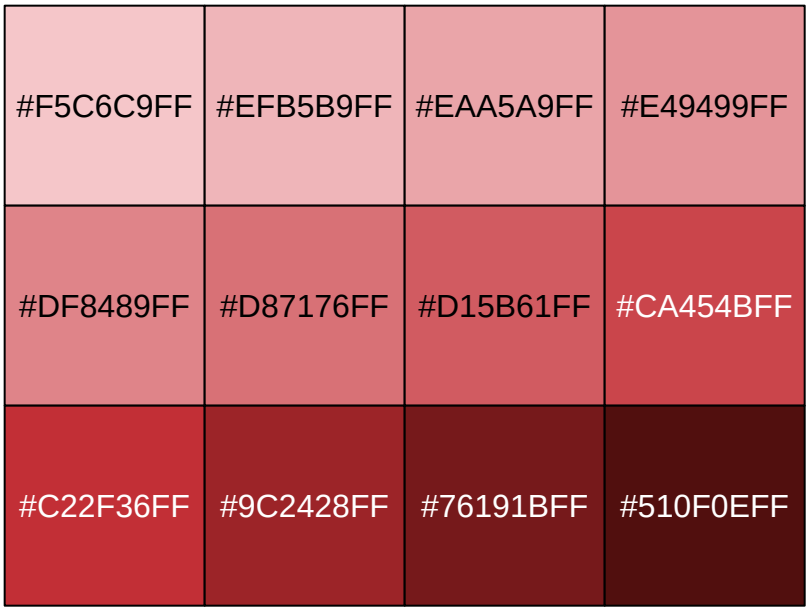

```
pink = paletteer_dynamic(`"cartography::pink.pal"`, n = 12)
show_col(pink)
```




```
turq = paletteer_dynamic(`"cartography::turquoise.pal"`, n = 12)
show_col(turq)
```



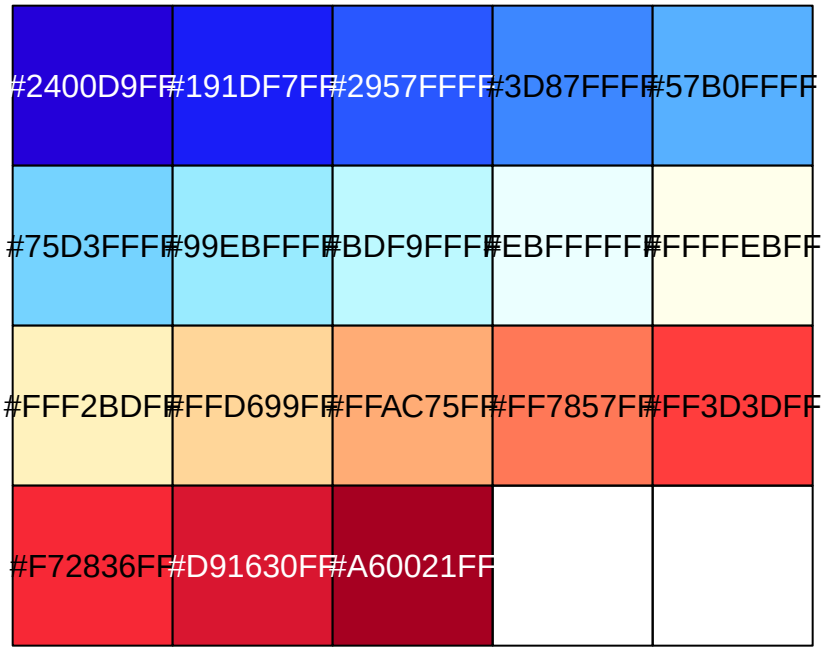

```
red = paletteer_dynamic(`"cartography::wine.pal"`, n = 12)
show_col(red)
```



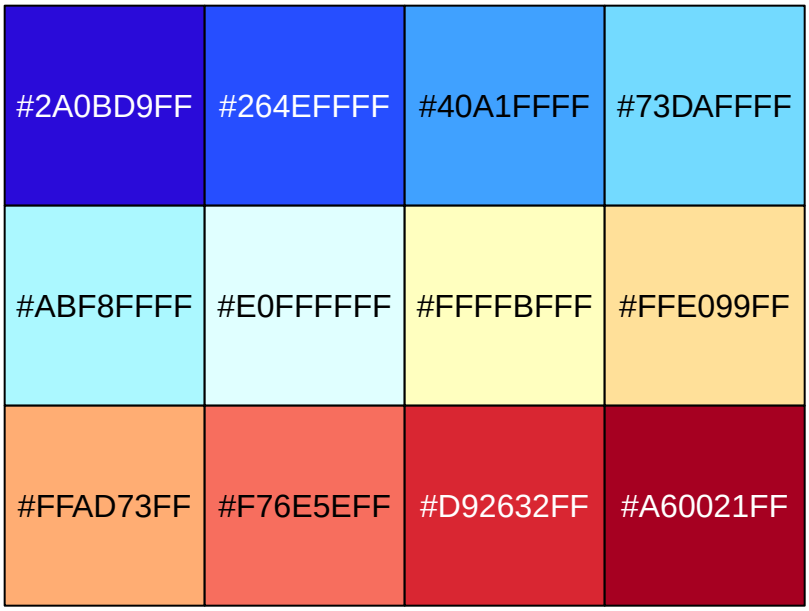
Dichromatic palettes are available in 'dichromat' package, the 18 or 12 below are changed depending how many colors you want in your palette:

Dark blue to dark orange - BluetoDarkOrange.18 or 12 Blue to Red - DarkRedtoBlue.18 or 12 Green to Purple - GreentoMagenta.16 Purple to orange - inferno (this palette is in the viridis package)

```
RB = paletteer_d(`"dichromat::DarkRedtoBlue_18"`, n = 18)
show_col(RB)
```

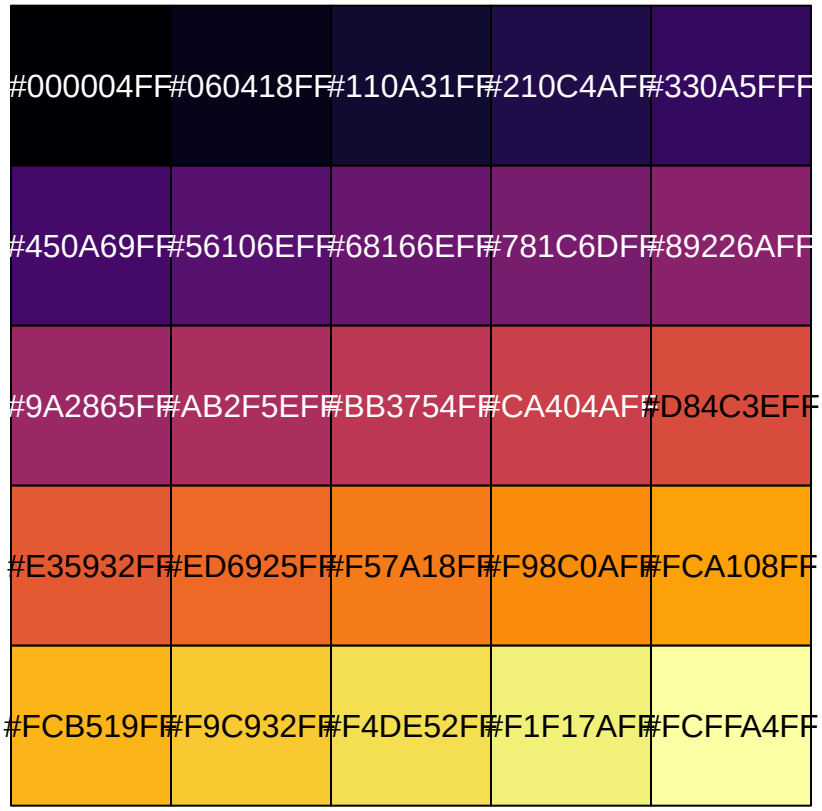



```
RB2 = paletteer_d(`"dichromat::DarkRedtoBlue_12"`, n = 12)
show_col(RB2)
```




```
GM = paletteer_d(`"dichromat::GreentoMagenta_16"`, n = 16)
show_col(GM)
```


```
inferno = paletteer_c(`"viridis::inferno"`, n = 25)
show_col(inferno)
```

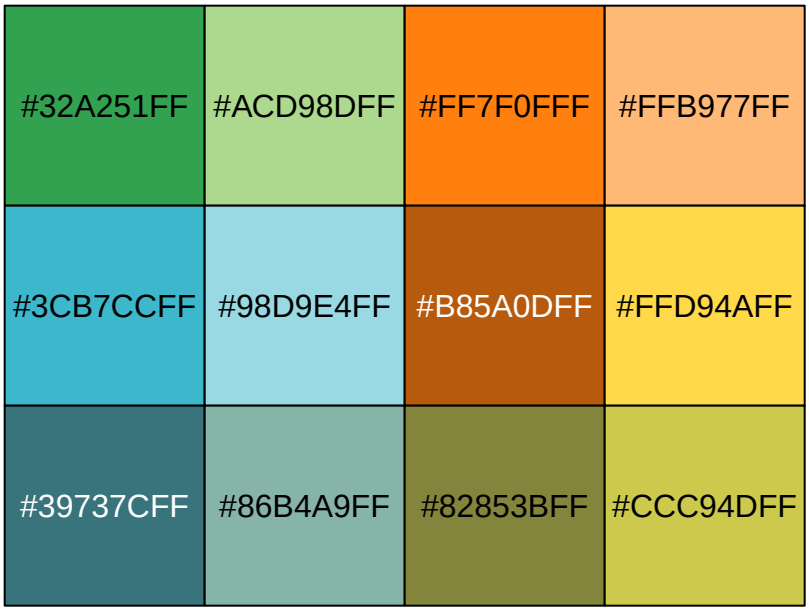


Non-prdinal color palette: Useful for categorical variable plotting or anything that does not have an order like longitudinal lines for instance:

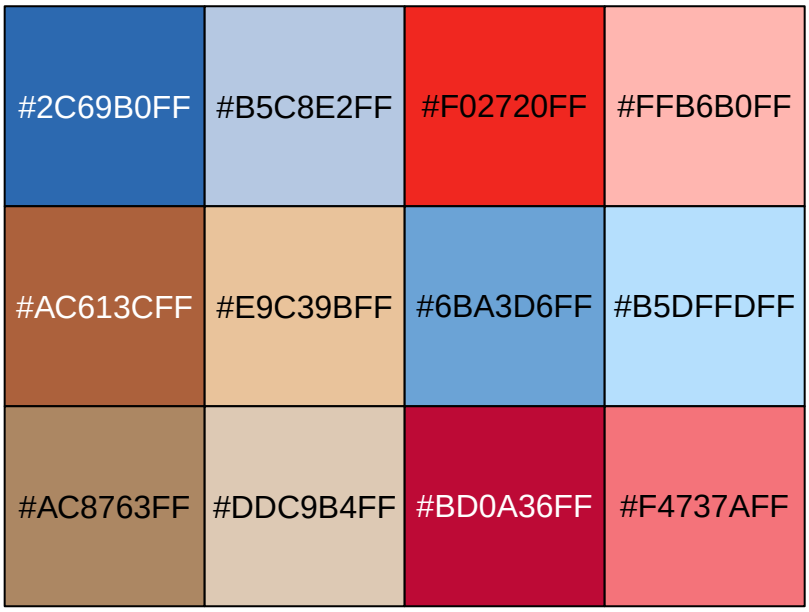
ggthemes has many here are a few we like: Classic_Green_Orange_12 Classic_Blue_Red_12 Classic_Cyclic

It should be noted that ggthemes have less values in their palettes, if you need more than 12 colors, these are not good options. However between ggthemes, ggthemr, and ggsci there are many palettes that are available in both dark and light which will make palette choosing consistent between dark and light modes of your figures.

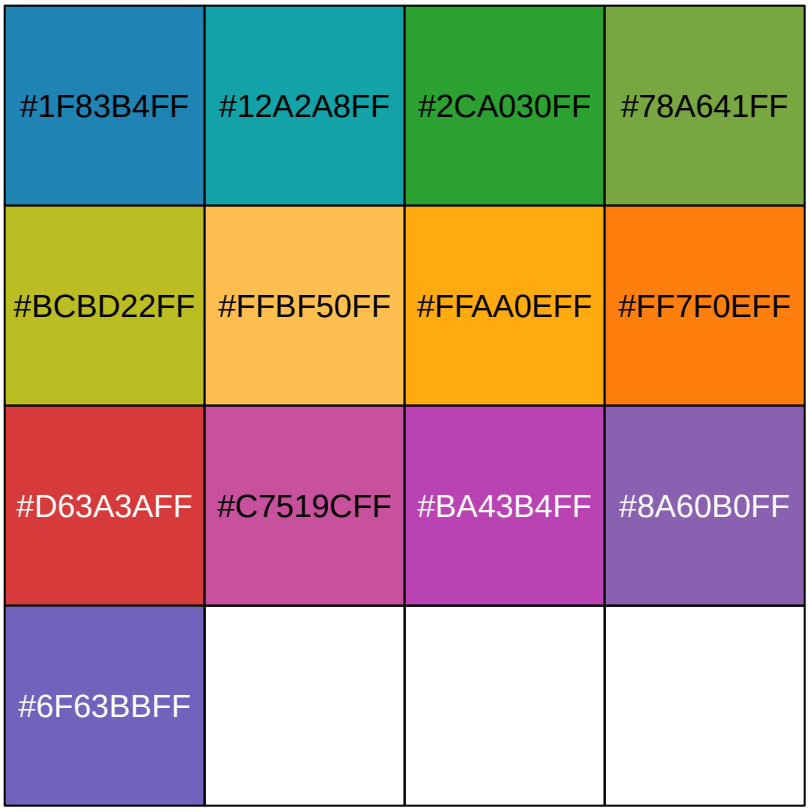
```
GO = paletteer_d(`"ggthemes::Classic_Green_Orange_12"`, n =12)
show_col(GO)
```




```
BR = paletteer_d(`"ggthemes::Classic_Blue_Red_12"`, n =12)
show_col(BR)
```

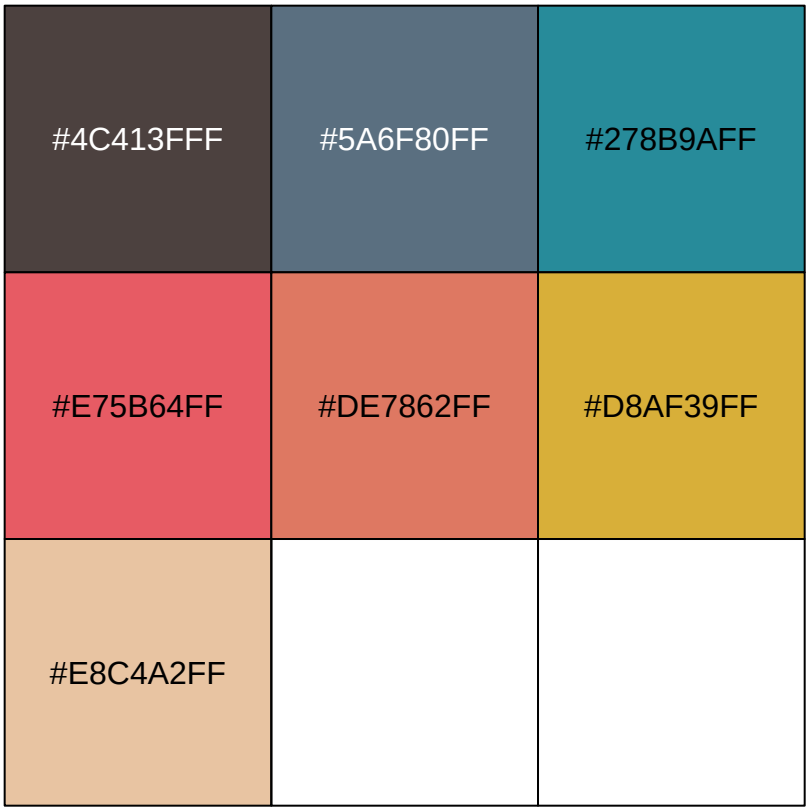



```
cyclic = paletteer_d(`"ggthemes::Classic_Cyclic"`, n = 13)
show_col(cyclic)
```



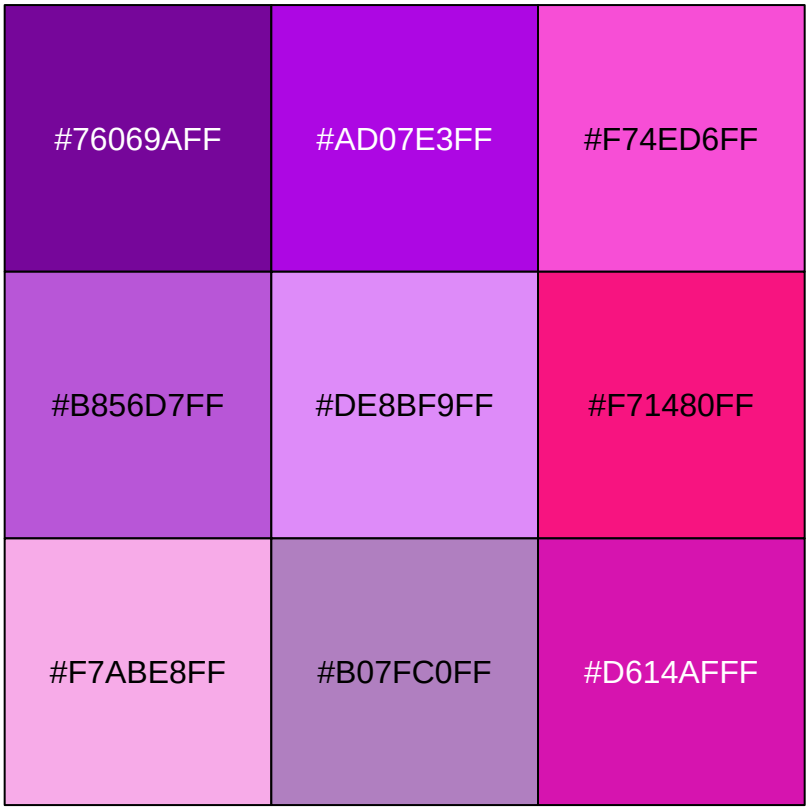
Whimsical color palettes: There are many other packages with colors. The Paletteer package is one single package that houses almost all of the palettes available in R, it is what I use here for display. So keep in mind that you may browse what is available in paletteer to find a fit for purpose color design. Another reminder that generally we stay away from rainbow palettes and the like because of color blindness concerns.

```
ponyo = paletteer_d(`"ghibli::PonyoMedium"`, n = 7)
show_col(ponyo)
```




```
ponyo2 = paletteer_d(`"ghibli::PonyoDark"`, n = 7)
show_col(ponyo2)
```

```
passion = paletteer_d(`"ggprism::purple_passion"`, n = 9)
show_col(passion)
```




```
beyonce = paletteer_d(`"beyonce::X8"`, n = 6)
show_col(beyonce)
```

And there are many many others but the above will serve most any purpose: https://github.com/EmilHvitfeldt/r-color-palettes

Check package 'paletteer' for more information