I’m a long-term fan of Weezer. Such was the brilliance of their first two albums that I have stuck with them through thick and thin. And dear me, there has been some very thin music. Nonetheless I own every album – thirteen of them. Among them are six albums entitled “Weezer”.

These records are colloquially referred to by the colour of the album. In chronological order: blue, green, red, white, teal and black. It struck me that these colours have a dataviz quality to them. They could be used for making colour palettes in R.

**What are the colours?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title | Year | rgb | hex | hsl | RYM |
| blue | 1994 | rgb(24,155,204) | #189BCC | hsl(196,79,45) | 3.91 |
| green | 2001 | rgb(190,204,65) | #BECC41 | hsl(66,58,53) | 3.02 |
| red | 2008 | rgb(234,33,58) | #EA213A | hsl(353,83,52) | 2.51 |
| white | 2016 | rgb(243,243,243) | #F3F3F3 | hsl(0,0,95) | 3.46 |
| teal | 2018 | rgb(43,188,187) | #2BBCBB | hsl(180,63,45) | 2.18 |
| black | 2019 | rgb(13,13,13) | #0D0D0D | hsl(0,0,5) | 2.18 |

Blue, green, red and teal are all plain colour. White and black albums are a gradient and the exact colour depends on where you sample.

**Let’s use them in R!**

We can specify the colours in hex format as a character vector and use them direct in ggplot, as shown below. Here I am plotting each “Weezer” album’s rating on rateyourmusic.

library(tidyverse)

library(extrafont)

# weezer colours taken from weezer albums

weezer\_album\_colours <- c("#189BCC",

"#BECC41",

"#EA213A",

"#F3F3F3",

"#2BBCBB",

"#0D0D0D")

# ratings of weezer albums from [rateyourmusic.com](http://rateyourmusic.com) taken on 2020-05-09

df <- data.frame(year=c("1994", "2001", "2008", "2016", "2018", "2019"),

rym=c(3.91,3.02,2.51,3.46,2.18,2.18))

# make a plot using these colours

p1 <- ggplot(df, aes(x = year, y = rym, fill = year)) +

scale\_fill\_manual(values = weezer\_album\_colours) +

geom\_bar(stat="identity", colour = "black") +

labs(title = "Ratings for Weezer's Weezer albums",

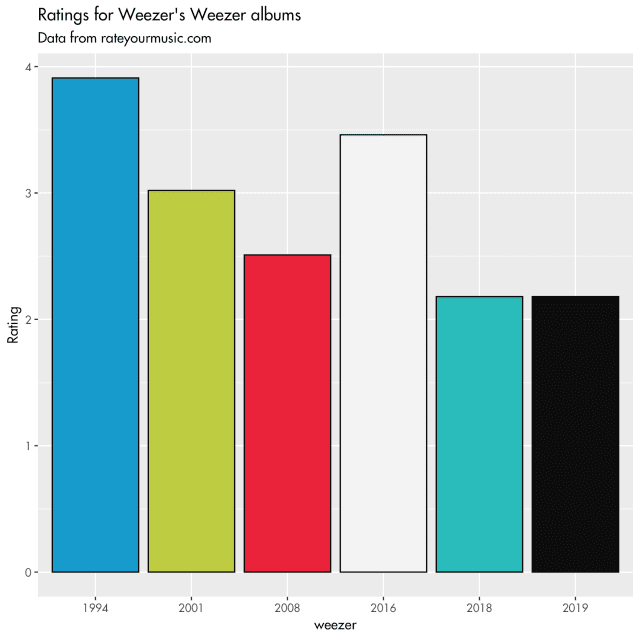
subtitle = "Data from [rateyourmusic.com](http://rateyourmusic.com)",

x = "weezer", y = "Rating") +

theme(text = element\_text(family = "Futura-Medium"),

legend.position="none")

ggsave("ratings.png",p1,dpi = 300)



That was fun, but specifying the hex code for each album colour is a bit cumbersome.

**How can we use these colours in a custom palette?**

We can make a custom palette of named colours so that we can easily access the colours to make plots.

# now make a named character vector of weezer colours

weezer\_colours <- c(

`blue` = "#189BCC",

`green` = "#BECC41",

`red` = "#EA213A",

`white` = "#F3F3F3",

`teal` = "#2BBCBB",

`black` = "#0D0D0D")

# a function to get hex codes of weezer colours

weezer\_cols <- function(...) {

cols <- c(...)

if (is.null(cols))

return (weezer\_colours)

weezer\_colours[cols]

}

# all colours can be listed with

weezer\_cols()

# or colours can be returned by character name(s)

weezer\_cols("red", "blue")

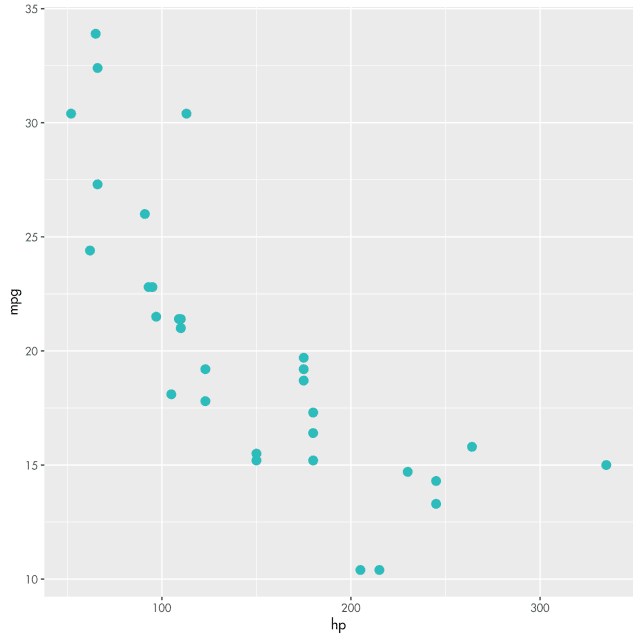
# weezer colours can be used in a plot like this

p2 <- ggplot(mtcars, aes(x = hp, y = mpg)) +

geom\_point(color = weezer\_cols("teal"),size = 3) +

theme(text = element\_text(family = "Futura-Medium"))

ggsave("teal\_example.png", p2, dpi = 300)



An example plot using “teal” specified by the Teal Album colour

The example above shows how to use the album cover colours directly in a plot. How about generating a LUT/colour palette? We can specify a gradient of colours and get R to interpolate colours along the gradient to use for colourscales and other methods.

# now let's make some palettes

weezer\_palettes <- list(

`main` = weezer\_cols("blue", "red", "green"),

`cool` = weezer\_cols("white", "teal"),

`hot` = weezer\_cols("black", "red"),

`mixed` = weezer\_cols("blue", "green", "red", "white", "teal", "black"),

`mono` = weezer\_cols("white", "black")

)

# function to interpolate palette. Default is main. Option to reverse

weezer\_pal <- function(palette = "main", reverse = FALSE, ...) {

pal <- weezer\_palettes[[palette]]

if (reverse) pal <- rev(pal)

colorRampPalette(pal, ...)

}

# function to colour graph bjects

scale\_colour\_weezer <- function(palette = "main", discrete = TRUE, reverse = FALSE, ...) {

pal <- weezer\_pal(palette = palette, reverse = reverse)

if (discrete) {

discrete\_scale("colour", paste0("weezer\_", palette), palette = pal, ...)

} else {

scale\_color\_gradientn(colours = pal(256), ...)

}

}

# function for filling graph objects

scale\_fill\_weezer <- function(palette = "main", discrete = TRUE, reverse = FALSE, ...) {

pal <- weezer\_pal(palette = palette, reverse = reverse)

if (discrete) {

discrete\_scale("fill", paste0("weezer\_", palette), palette = pal, ...)

} else {

scale\_fill\_gradientn(colours = pal(256), ...)

}

}

These functions allow us to specify gradients between the different album cover colours. Here are some examples:

# examples of a plot using scale\_colour\_weezer

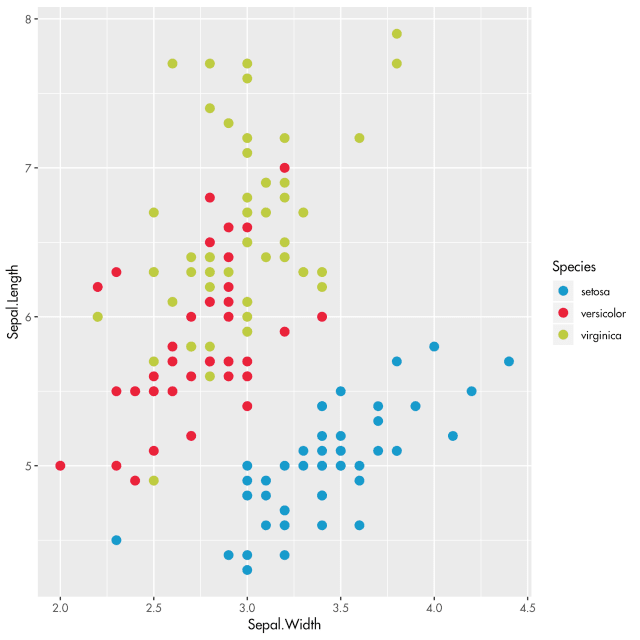
p2 <- ggplot(iris, aes(Sepal.Width, Sepal.Length, color = Species)) +

geom\_point(size = 3) +

scale\_colour\_weezer() +

theme(text = element\_text(family = "Futura-Medium"))

ggsave("colour\_scale\_example.png",p2,dpi = 300)



An example using the first three Weezer album colours (our default)

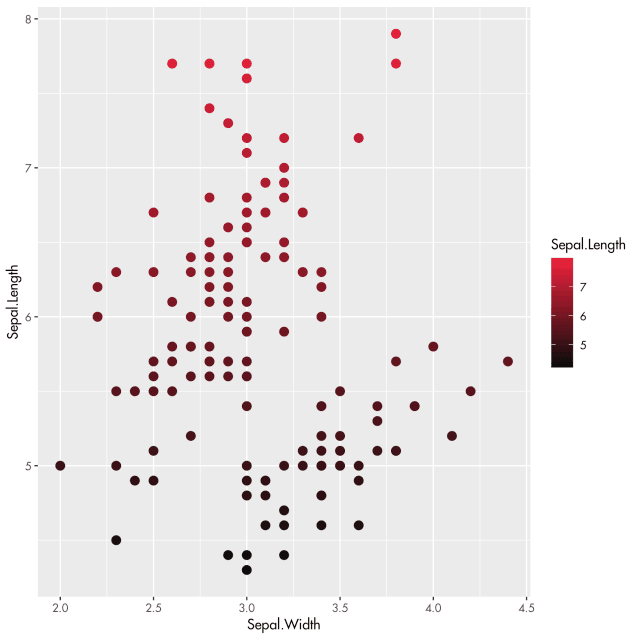
p3 <- ggplot(iris, aes(Sepal.Width, Sepal.Length, color = Sepal.Length)) +

geom\_point(size = 3) +

scale\_colour\_weezer(discrete = FALSE, palette = "hot") +

theme(text = element\_text(family = "Futura-Medium"))

ggsave("colour\_scale\_example2.png", p3, dpi = 300)



Example using “hot”, a gradient from the Black album to the Red album.

p4 <- ggplot(mpg, aes(manufacturer, fill = manufacturer)) +

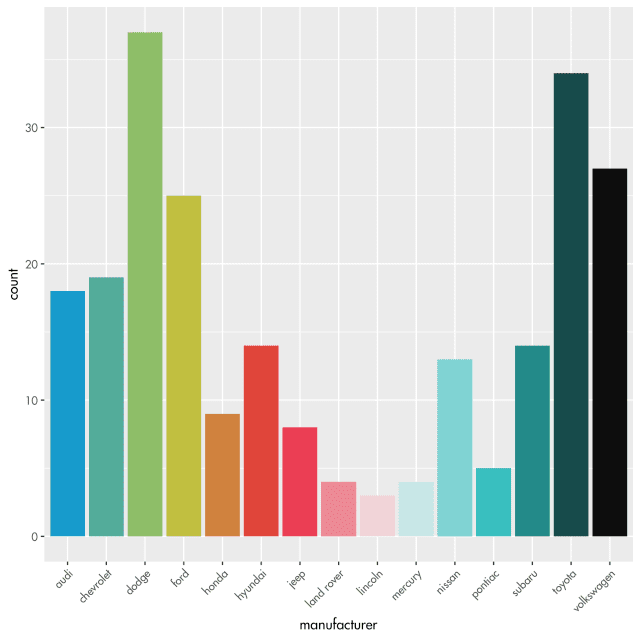
geom\_bar() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +

scale\_fill\_weezer(palette = "mixed", guide = "none") +

theme(text = element\_text(family = "Futura-Medium"))

ggsave("colour\_scale\_example3.png", p4, dpi = 300)



An example using mtcars. The colours follow a gradient through all “Weezer” album colours.