

# SIMILARITY MEASURE IN THE SPACE OF **COLOR PALETTES**

Color exploration with  
Emil Hvitfeldt Hansen  
Twitter: @Emil\_Hvitfeldt  
Github: EmilHvitfeldt

[github.com/EmilHvitfeldt/OCRUG-color-talk](https://github.com/EmilHvitfeldt/OCRUG-color-talk)  
[bit.ly/OCRUG-Color-Talk](https://bit.ly/OCRUG-Color-Talk)

# OVERVIEW

- Overview
- Motivation
- Method
- Results

# AVAILABLE COLOR PALETTES

# AVAILABLE COLOR PALETTES

Google color palettes for r

All Images Shopping Videos News More Settings Tools View saved SafeSearch ▾

gmt ggplot ggplot2 rgb brewer rcolorbrewer color chart palette ggplot chunk rcolorsheet plot rainbow grey

The image displays a collection of color palettes and charts, organized into several sections:

- Top Left:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Top Center:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Top Right:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Middle Left:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Middle Center:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Middle Right:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Bottom Left:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Bottom Center:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.
- Bottom Right:** A grid of color palettes labeled with names like 'VIOred', 'YlOrBu', 'YlOrGr', 'RdPu', 'PuBu', 'PuBuGn', 'OrRd', 'Grays', 'Greens', 'BuPu', 'BuGn', 'Blues', 'Set3', 'Set1', 'Set2', 'Pastel1', 'Pastel2', 'Dark2', 'Accent', 'Spectral', 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG'.

480 x 480 - r-graph-gallery.com



# AVAILABLE COLOR PALETTES

Google color palettes for r

All Images Shopping Videos News More Settings Tools View saved SafeSearch

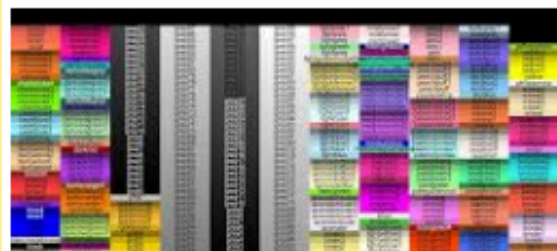
gmt ggplot ggplot2 rgb brewer rcolorbrewer color chart palette ggplot chunk rcolorsheet plot rainbow grey

The image displays a collection of color palettes and charts, categorized into sequential, qualitative, and diverging scales. The palettes are arranged in a grid, with labels indicating their names and the number of colors. The labels include: Spectral, RdYlGn, RdYlBu, RdGy, RdBu, PuOr, PRGn, PiYG, BrBG, Dark2, Paired, Set1, Set2, Set3, and a large section of sequential palettes. The charts show the palettes applied to a series of data points, demonstrating their effectiveness in visualizing different types of data. The bottom section of the image shows a grid of color palettes, including 'R colors', 'RColorBrewer', 'Color ramps', and 'Color palettes'. The bottom right corner features a grid of color palettes, including 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG', 'Dark2', 'Paired', 'Set1', 'Set2', 'Set3', and a large section of sequential palettes. The bottom right corner also includes a grid of color palettes, including 'RdYlGn', 'RdYlBu', 'RdGy', 'RdBu', 'PuOr', 'PRGn', 'PiYG', 'BrBG', 'Dark2', 'Paired', 'Set1', 'Set2', 'Set3', and a large section of sequential palettes.

480 x 480 - r-graph-gallery.com



# AVAILABLE COLOR PALETTES

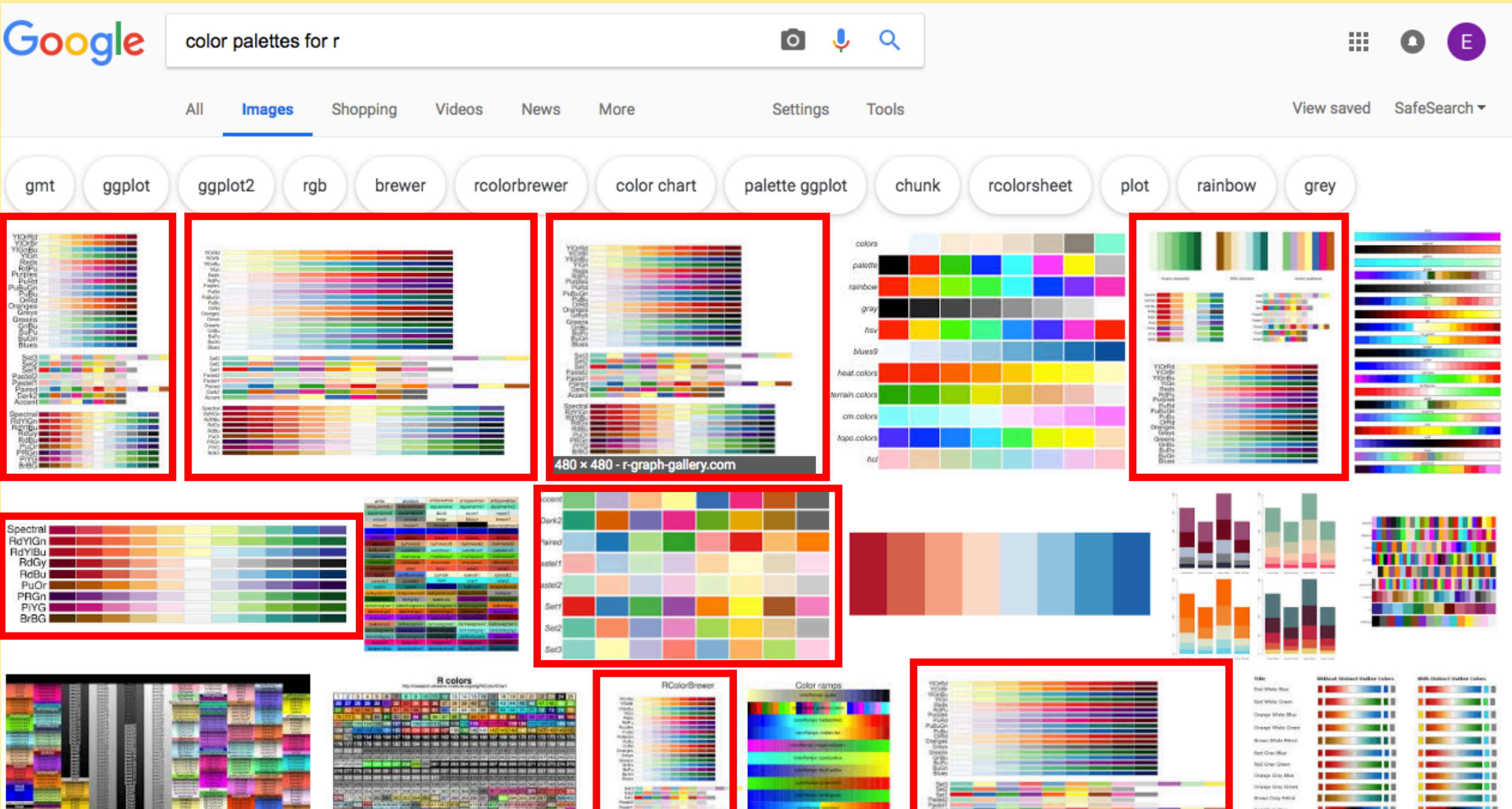




# RColorBrewer

# AVAILABLE COLOR PALETTES

## RColorBrewer



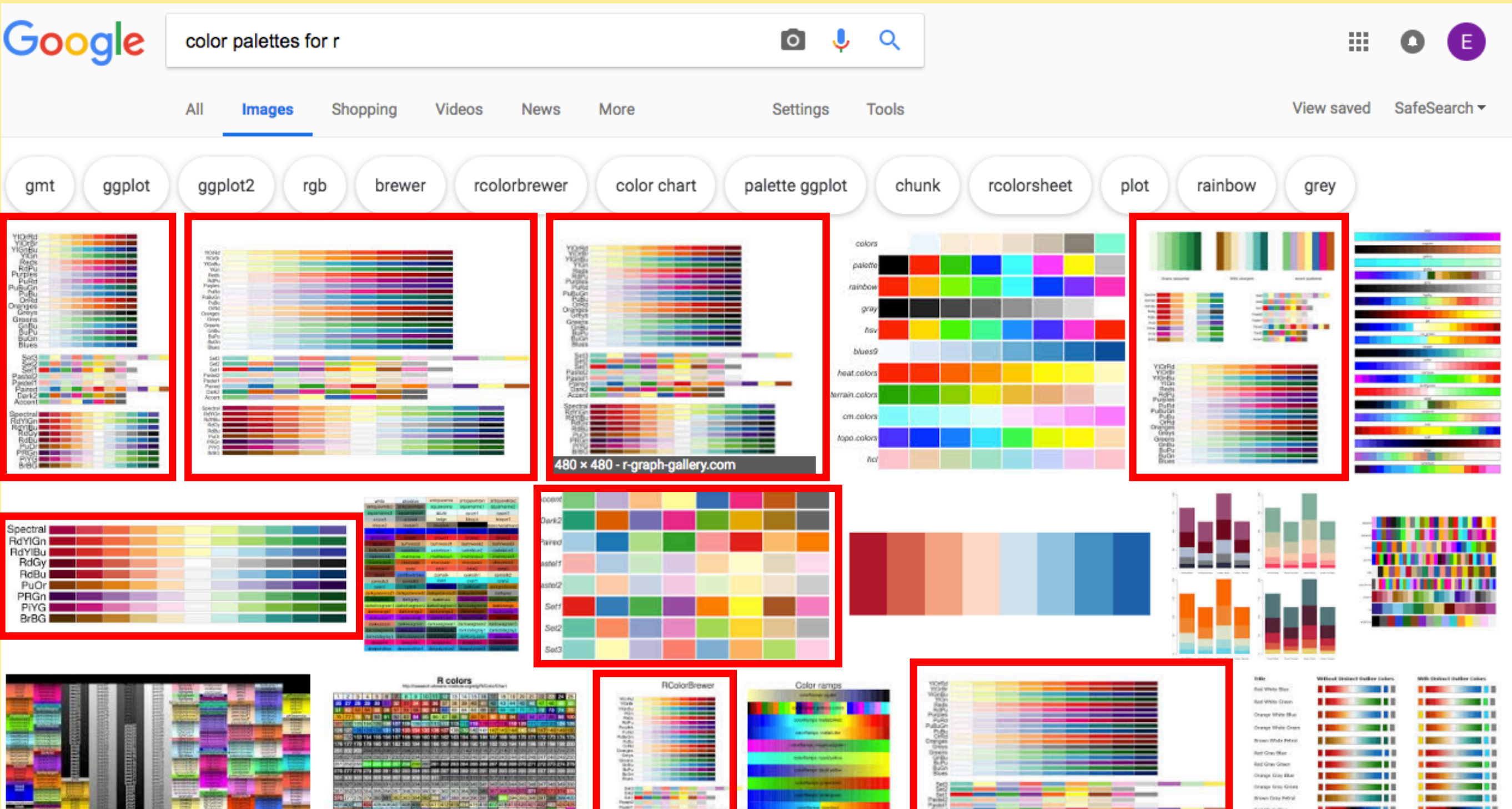


# RColorBrewer

# AVAILABLE COLOR PALETTES

## RColorBrewer

## RColorBrewer





# RColorBrewer

# RColorBrewer

# AVAILABLE COLOR PALETTES

## RColorBrewer

## RColorBrewer

Google color palettes for r

All Images Shopping Videos News More Settings Tools View saved SafeSearch

gmt ggplot ggplot2 rgb brewer rcolorbrewer color chart palette ggplot chunk rcolorsheet plot rainbow grey

The image displays a collection of RColorBrewer color palettes, organized into several groups. The top section shows a Google search interface for 'color palettes for r'. Below the search bar, there are tabs for different image sources: All, Images (selected), Shopping, Videos, News, More, Settings, and Tools. A row of filters includes: gmt, ggplot, ggplot2, rgb, brewer, rcolorbrewer, color chart, palette ggplot, chunk, rcolorsheet, plot, rainbow, and grey. The main content area is a grid of various color palettes. Some palettes are highlighted with red borders, including: a large grid of sequential palettes (e.g., Spectral, RdYlGn, RdYlBu, RdGy, RdBu, PuOr, PRGn, PiYG, BrBG), a grid of qualitative palettes (e.g., Qual1, Qual2, Qual3, Qual4, Qual5, Qual6, Qual7, Qual8, Qual9, Qual10, Qual11, Qual12, Qual13, Qual14, Qual15, Qual16, Qual17, Qual18, Qual19, Qual20, Qual21, Qual22, Qual23, Qual24, Qual25, Qual26, Qual27, Qual28, Qual29, Qual30, Qual31, Qual32, Qual33, Qual34, Qual35, Qual36, Qual37, Qual38, Qual39, Qual40, Qual41, Qual42, Qual43, Qual44, Qual45, Qual46, Qual47, Qual48, Qual49, Qual50, Qual51, Qual52, Qual53, Qual54, Qual55, Qual56, Qual57, Qual58, Qual59, Qual60, Qual61, Qual62, Qual63, Qual64, Qual65, Qual66, Qual67, Qual68, Qual69, Qual70, Qual71, Qual72, Qual73, Qual74, Qual75, Qual76, Qual77, Qual78, Qual79, Qual80, Qual81, Qual82, Qual83, Qual84, Qual85, Qual86, Qual87, Qual88, Qual89, Qual90, Qual91, Qual92, Qual93, Qual94, Qual95, Qual96, Qual97, Qual98, Qual99, Qual100), a grid of diverging palettes (e.g., Diverging1, Diverging2, Diverging3, Diverging4, Diverging5, Diverging6, Diverging7, Diverging8, Diverging9, Diverging10, Diverging11, Diverging12, Diverging13, Diverging14, Diverging15, Diverging16, Diverging17, Diverging18, Diverging19, Diverging20, Diverging21, Diverging22, Diverging23, Diverging24, Diverging25, Diverging26, Diverging27, Diverging28, Diverging29, Diverging30, Diverging31, Diverging32, Diverging33, Diverging34, Diverging35, Diverging36, Diverging37, Diverging38, Diverging39, Diverging40, Diverging41, Diverging42, Diverging43, Diverging44, Diverging45, Diverging46, Diverging47, Diverging48, Diverging49, Diverging50, Diverging51, Diverging52, Diverging53, Diverging54, Diverging55, Diverging56, Diverging57, Diverging58, Diverging59, Diverging60, Diverging61, Diverging62, Diverging63, Diverging64, Diverging65, Diverging66, Diverging67, Diverging68, Diverging69, Diverging70, Diverging71, Diverging72, Diverging73, Diverging74, Diverging75, Diverging76, Diverging77, Diverging78, Diverging79, Diverging80, Diverging81, Diverging82, Diverging83, Diverging84, Diverging85, Diverging86, Diverging87, Diverging88, Diverging89, Diverging90, Diverging91, Diverging92, Diverging93, Diverging94, Diverging95, Diverging96, Diverging97, Diverging98, Diverging99, Diverging100), and a grid of sequential palettes (e.g., Sequential1, Sequential2, Sequential3, Sequential4, Sequential5, Sequential6, Sequential7, Sequential8, Sequential9, Sequential10, Sequential11, Sequential12, Sequential13, Sequential14, Sequential15, Sequential16, Sequential17, Sequential18, Sequential19, Sequential20, Sequential21, Sequential22, Sequential23, Sequential24, Sequential25, Sequential26, Sequential27, Sequential28, Sequential29, Sequential30, Sequential31, Sequential32, Sequential33, Sequential34, Sequential35, Sequential36, Sequential37, Sequential38, Sequential39, Sequential40, Sequential41, Sequential42, Sequential43, Sequential44, Sequential45, Sequential46, Sequential47, Sequential48, Sequential49, Sequential50, Sequential51, Sequential52, Sequential53, Sequential54, Sequential55, Sequential56, Sequential57, Sequential58, Sequential59, Sequential60, Sequential61, Sequential62, Sequential63, Sequential64, Sequential65, Sequential66, Sequential67, Sequential68, Sequential69, Sequential70, Sequential71, Sequential72, Sequential73, Sequential74, Sequential75, Sequential76, Sequential77, Sequential78, Sequential79, Sequential80, Sequential81, Sequential82, Sequential83, Sequential84, Sequential85, Sequential86, Sequential87, Sequential88, Sequential89, Sequential90, Sequential91, Sequential92, Sequential93, Sequential94, Sequential95, Sequential96, Sequential97, Sequential98, Sequential99, Sequential100). The bottom section shows a grid of RColorBrewer palettes, including: RColorBrewer, Color ramps, and a grid of sequential palettes (e.g., Sequential1, Sequential2, Sequential3, Sequential4, Sequential5, Sequential6, Sequential7, Sequential8, Sequential9, Sequential10, Sequential11, Sequential12, Sequential13, Sequential14, Sequential15, Sequential16, Sequential17, Sequential18, Sequential19, Sequential20, Sequential21, Sequential22, Sequential23, Sequential24, Sequential25, Sequential26, Sequential27, Sequential28, Sequential29, Sequential30, Sequential31, Sequential32, Sequential33, Sequential34, Sequential35, Sequential36, Sequential37, Sequential38, Sequential39, Sequential40, Sequential41, Sequential42, Sequential43, Sequential44, Sequential45, Sequential46, Sequential47, Sequential48, Sequential49, Sequential50, Sequential51, Sequential52, Sequential53, Sequential54, Sequential55, Sequential56, Sequential57, Sequential58, Sequential59, Sequential60, Sequential61, Sequential62, Sequential63, Sequential64, Sequential65, Sequential66, Sequential67, Sequential68, Sequential69, Sequential70, Sequential71, Sequential72, Sequential73, Sequential74, Sequential75, Sequential76, Sequential77, Sequential78, Sequential79, Sequential80, Sequential81, Sequential82, Sequential83, Sequential84, Sequential85, Sequential86, Sequential87, Sequential88, Sequential89, Sequential90, Sequential91, Sequential92, Sequential93, Sequential94, Sequential95, Sequential96, Sequential97, Sequential98, Sequential99, Sequential100). The bottom right corner shows a grid of sequential palettes (e.g., Sequential1, Sequential2, Sequential3, Sequential4, Sequential5, Sequential6, Sequential7, Sequential8, Sequential9, Sequential10, Sequential11, Sequential12, Sequential13, Sequential14, Sequential15, Sequential16, Sequential17, Sequential18, Sequential19, Sequential20, Sequential21, Sequential22, Sequential23, Sequential24, Sequential25, Sequential26, Sequential27, Sequential28, Sequential29, Sequential30, Sequential31, Sequential32, Sequential33, Sequential34, Sequential35, Sequential36, Sequential37, Sequential38, Sequential39, Sequential40, Sequential41, Sequential42, Sequential43, Sequential44, Sequential45, Sequential46, Sequential47, Sequential48, Sequential49, Sequential50, Sequential51, Sequential52, Sequential53, Sequential54, Sequential55, Sequential56, Sequential57, Sequential58, Sequential59, Sequential60, Sequential61, Sequential62, Sequential63, Sequential64, Sequential65, Sequential66, Sequential67, Sequential68, Sequential69, Sequential70, Sequential71, Sequential72, Sequential73, Sequential74, Sequential75, Sequential76, Sequential77, Sequential78, Sequential79, Sequential80, Sequential81, Sequential82, Sequential83, Sequential84, Sequential85, Sequential86, Sequential87, Sequential88, Sequential89, Sequential90, Sequential91, Sequential92, Sequential93, Sequential94, Sequential95, Sequential96, Sequential97, Sequential98, Sequential99, Sequential100).



RColorBrewer

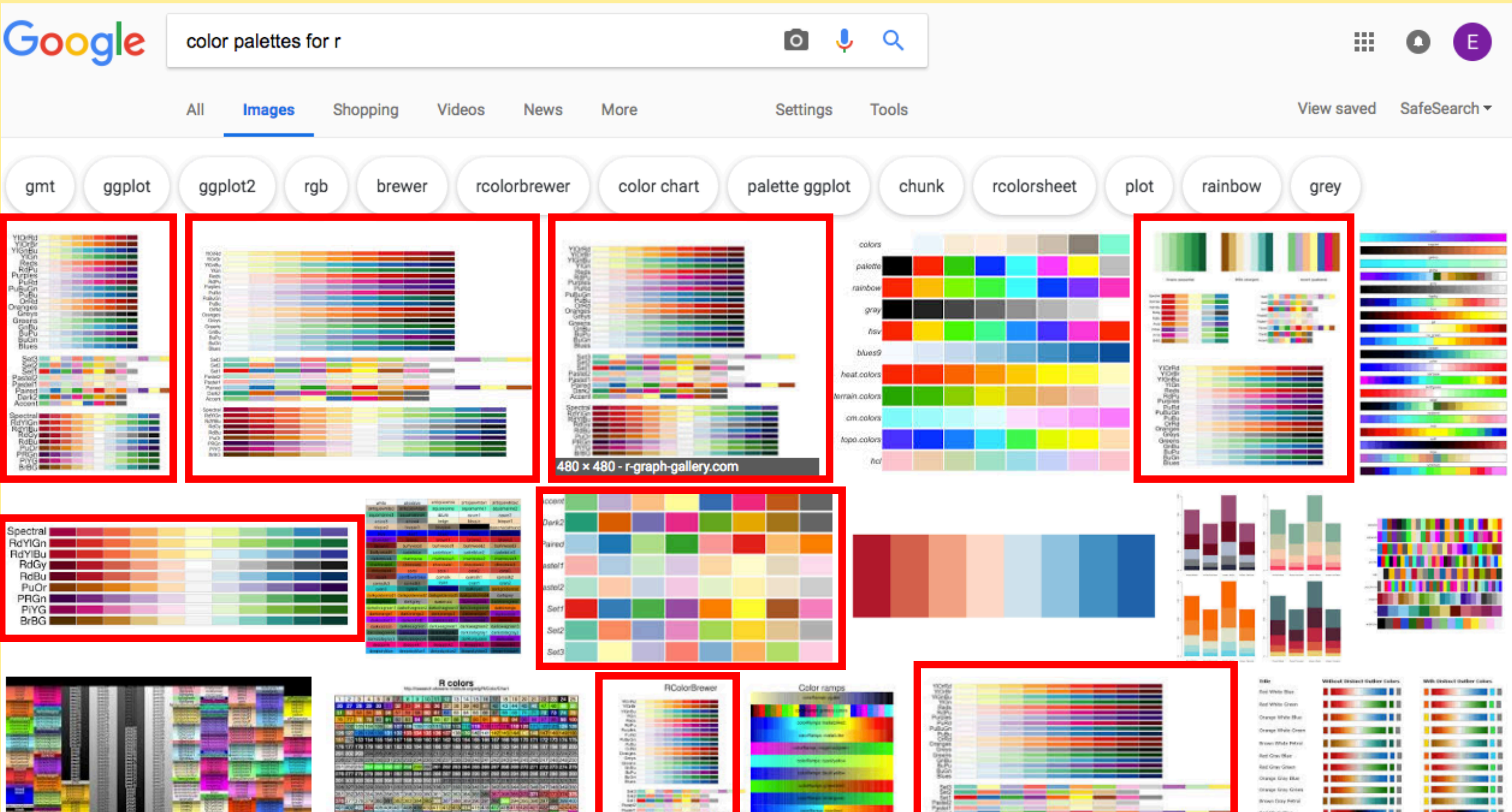
RColorBrewer

# AVAILABLE COLOR PALETTES

RColorBrewer

RColorBrewer

RColorBrewer



# SOMETHING IS MISSING

- Where is the **viridis** **palettes**?
- Where is the **scico** **palettes**?
- Where is the **nord** **palettes**?
- Where is the **ochRe** **palettes**?
- ...



&lt;&gt; Code

! Issues 2

Pull requests 0

Projects 0

Wiki

Insights

Settings

## Comprehensive list of color palettes available in r

Edit

[rstats](#) [palette](#) [palettes](#) [palette-generation](#) [color](#) [color-palette](#) [r](#) [Manage topics](#)

50 commits

1 branch

0 releases

4 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



EmilHvitfeldt redone type sorted palettes

Latest commit d6a319c on Jul 18

R	redone type sorted palettes	2 months ago
man/figures	redone type sorted palettes	2 months ago
.Rbuildignore	Initial release, palettes alphabetically	7 months ago
.gitignore	adding full palettetown palette on seperate page. closes #31.	2 months ago
NEWS.md	redone type sorted palettes	2 months ago
README.Rmd	finish readme with new style	2 months ago
README.md	finish readme with new style	2 months ago
canva.Rmd	adding full palettetown palette on seperate page. closes #31.	2 months ago
palettetown.Rmd	adding full palettetown palette on seperate page. closes #31.	2 months ago
palettetown.md	adding full palettetown palette on seperate page. closes #31.	2 months ago
r-color-palettes.Rproj	Initial release, palettes alphabetically	7 months ago
type-sorted-palettes.Rmd	redone type sorted palettes	2 months ago
type-sorted-palettes.md	redone type sorted palettes	2 months ago

# 958 palettes from 28 packages

## cartography

blue.pal



orange.pal



red.pal



brown.pal



green.pal



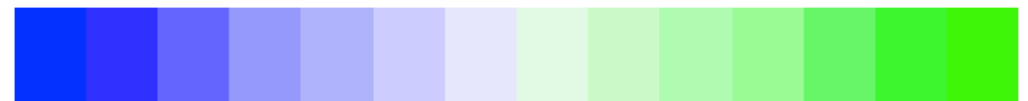
purple.pal



pink.pal



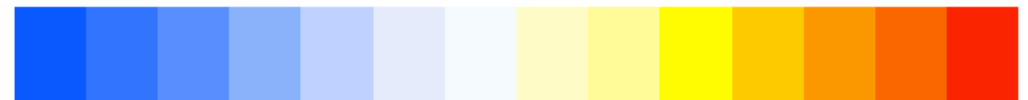
BlutoGreen.14



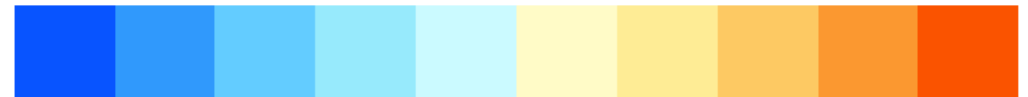
BlutoGray.8



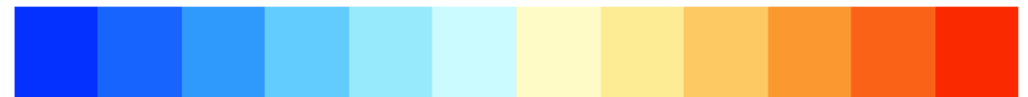
BlutoOrangeRed.14



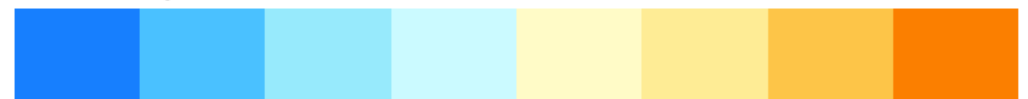
BlutoOrange.10



BlutoOrange.12



BlutoOrange.8



LightBlueDarkBlue.10

## ggpomological

pomological\_base




pomological\_palette



```
# Developmental version  
devtools::install_github("gadenbuie/ggpomological")
```

# paletteer



build passing  build passing CRAN 0.1.0 downloads 354/month codecov 95%

The goal of **paletteer** is to be a comprehensive collection of color palettes in R using a common interface. Think of it as the “caret of palettes”.

```
paletteer_c(package = "scico", palette = "berlin", n = 10)
#> [1] "#9EB0FF" "#5AA3DA" "#2D7597" "#194155" "#11181D" "#270C01" "#501802"
#> [8] "#8A3F2A" "#C37469" "#FFACAC"
paletteer_d("nord", "frost")
#> [1] "#8FBCBB" "#88C0D0" "#81A1C1" "#5E81AC"
paletteer_dynamic("cartography", "green.pal", 5)
#> [1] "#B8D9A9" "#8DBC80" "#5D9D52" "#287A22" "#17692C"
```

```
paletteer_c(scico, berlin, 10)
#> [1] "#9EB0FF" "#5AA3DA" "#2D7597" "#194155" "#11181D" "#270C01" "#501802"
#> [8] "#8A3F2A" "#C37469" "#FFACAC"
paletteer_d(nord, frost)
#> [1] "#8FBCBB" "#88C0D0" "#81A1C1" "#5E81AC"
paletteer_dynamic(cartography, green.pal, 5)
#> [1] "#B8D9A9" "#8DBC80" "#5D9D52" "#287A22" "#17692C"
```



# paletteer

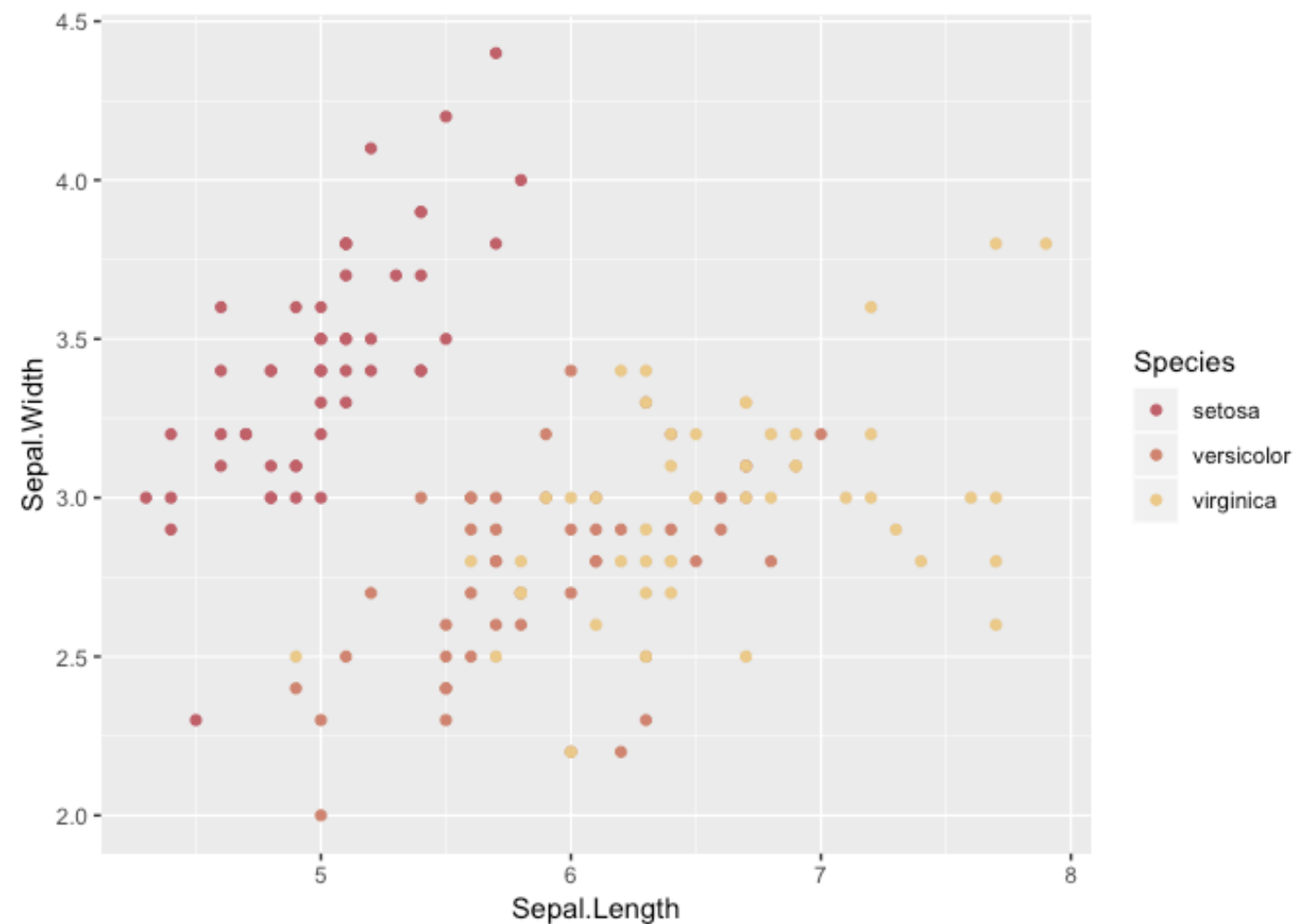


build passing build passing CRAN 0.1.0 downloads 354/month codecov 95%

The goal of **paletteer** is to be a comprehensive collection of color palettes in R using a common interface. Think of it as the “caret of palettes”.

```
library(ggplot2)

ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
  geom_point() +
  scale_color_paletteer_d(nord, aurora)
```



MY JOB IS DONE?



MY JOB IS DONE?





# MY JOB IS DONE?

- New problem with discoverability



# MY JOB IS DONE?

- New problem with discoverability
- 100s of palettes, hard to find a new one without using r-color-palettes





# MY JOB IS DONE?

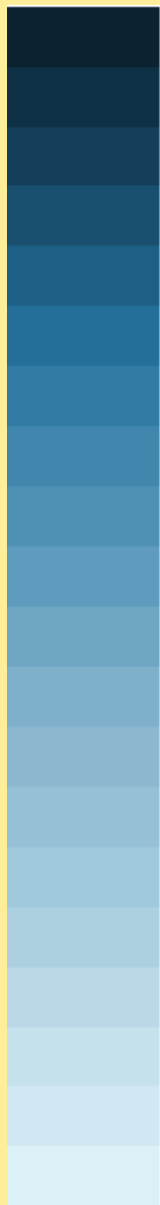
- New problem with discoverability
- 100s of palettes, hard to find a new one without using r-color-palettes
- Still using same palettes





# PLAN

- Develop tool to find palette that “looks like” palette you present.



# THE INPUT



#8E938D

#93A39C

#95B8AF

#9FD1BE

#BFCDBD

#A9D1A7

#E6E496



	red	green	blue
[1,]	142	147	141
[2,]	147	163	156
[3,]	149	184	175
[4,]	159	209	190
[5,]	191	205	189
[6,]	169	209	167
[7,]	230	228	150

# NO TRIVIAL TRANSFORMATION

Interpolation to fit max length would leads to trouble with palettes of the same scale but different lengths, and qualitative palettes.

Padding with zeroes would not work with diverging palettes.



# HAND CRAFTING FEATURES

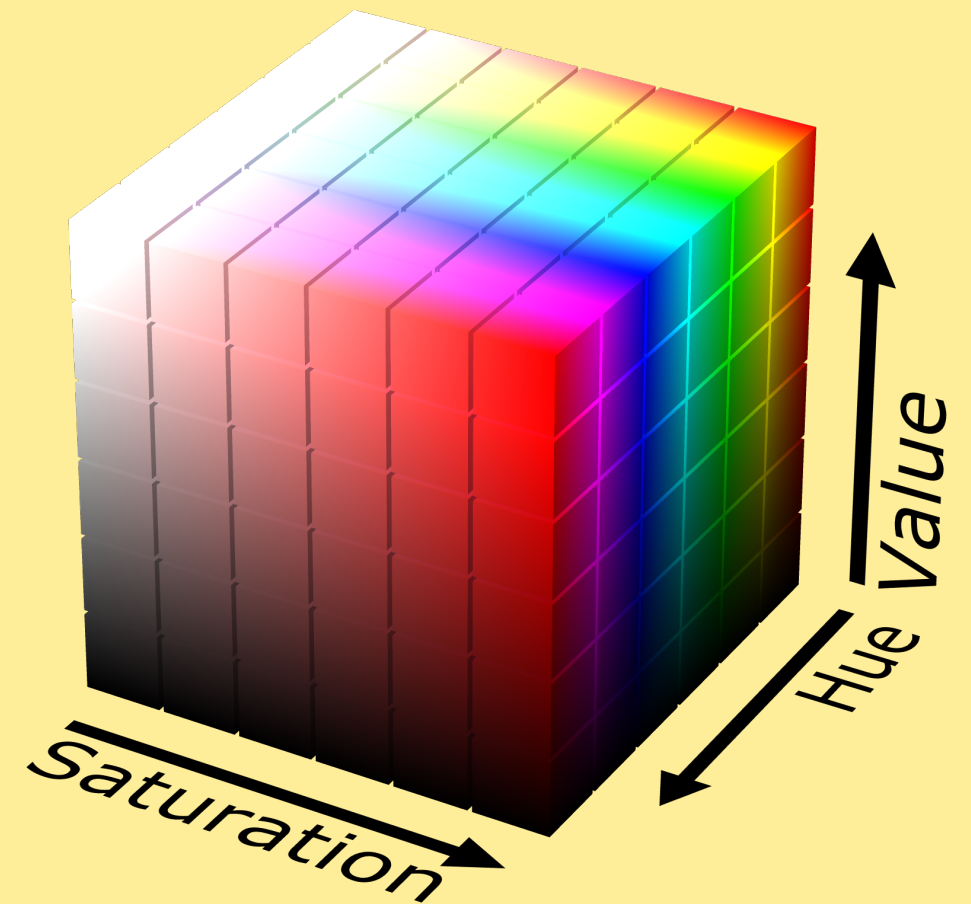
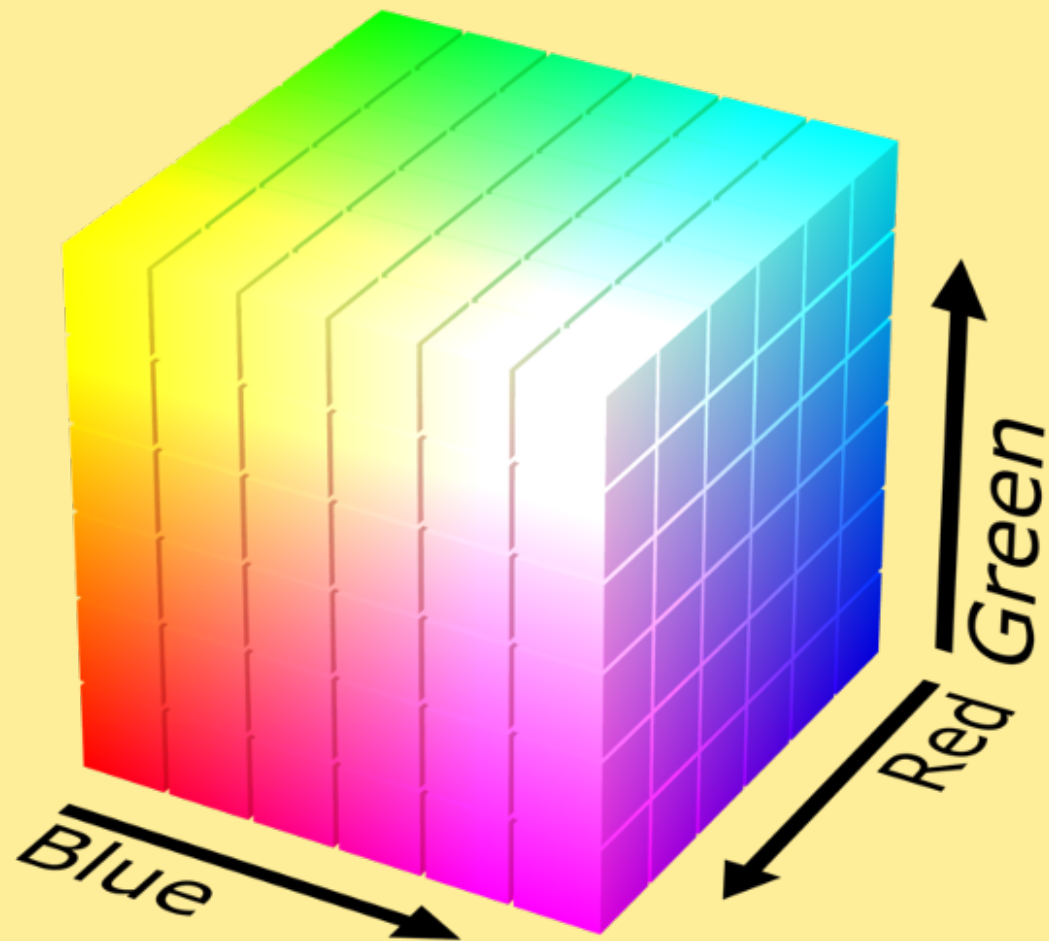
- similar to early image classification

# FEATURE LIST

- Includes a certain color
- All a certain color
- Length
- Type (sequential, divergent, qualitative. hardcoded)
- Linearity
- Twice linear (to identify divergent palettes)
- Colorblindness
- Minimal distance between points
- Maximal distance between points

Continuous palettes are taken with 20 points

# COLORSPACES



**Also CIELAB color space**



# METHOD

# METHOD

- Create a tibble with a list column of the colors

# METHOD

- Create a tibble with a list column of the colors
- Define a function for each feature



# METHOD

- Create a tibble with a list column of the colors
- Define a function for each feature
- Use `mutate()` with each function

# METHOD

- Create a tibble with a list column of the colors
- Define a function for each feature
- Use `mutate()` with each function
- Model

# METHOD

- Create a tibble with a list column of the colors
- Define a function for each feature
- Use `mutate()` with each function
- Model
- ...



# METHOD

- Create a tibble with a list column of the colors
- Define a function for each feature
- Use mutate() with each function
- Model
- ...
- Profit?

# CREATING TIBBLE

R/data\_list.R

##		package	palette	type	length	group	colors
##	1	ggthemes	Red	sequential	20	c	<chr [20]>
##	2	ggthemes	Green	sequential	20	c	<chr [20]>
##	3	ggthemes	Blue	sequential	20	c	<chr [20]>
##	4	ggthemes	Orange	sequential	20	c	<chr [20]>
##	5	ggthemes	Gray	sequential	20	c	<chr [20]>
##	6	ggthemes	Blue Light	sequential	20	c	<chr [20]>

934 rows in total

# FEATURE FUNCTIONS

```
color_contains <- function(pal, color, space) {  
  spectrum1 <- convert_colour(t(col2rgb(pal))), "rgb", space)  
  spectrum2 <- convert_colour(t(col2rgb(color))), "rgb", space)  
  
  compare_colour(spectrum1, spectrum2, space, method = "cie2000") %>%  
    min()  
}
```

convert\_colour() and compare\_colour() from **farver** package



# FEATURE FUNCTIONS

```
linear <- function(pal, space) {  
  colors <- convert_colour(t(col2rgb(pal)), "rgb", space) %>%  
    as.data.frame() %>%  
    mutate(x = row_number())  
  
  out <- lm(x ~ ., colors) %>%  
    summary() %>%  
    .$adj.r.squared  
  
  if(is.nan(out))  
    return(0)  
  
  out  
}
```

convert\_colour() and compare\_colour() from **farver** package

```
data_num <- data_list %>%
  mutate(# contains all a certain color
    contains_yellow = map_dbl(colors, ~ color_contains(.x, "yellow", "hsv")),
    contains_orange = map_dbl(colors, ~ color_contains(.x, "orange", "hsv")),
    contains_red = map_dbl(colors, ~ color_contains(.x, "red", "hsv")),
    contains_purple = map_dbl(colors, ~ color_contains(.x, "purple", "hsv")),
    contains_blue = map_dbl(colors, ~ color_contains(.x, "blue", "hsv")),
    contains_green = map_dbl(colors, ~ color_contains(.x, "green", "hsv")),
    contains_brown = map_dbl(colors, ~ color_contains(.x, "brown", "hsv")),
    contains_white = map_dbl(colors, ~ color_contains(.x, "white", "hsv")),
    contains_black = map_dbl(colors, ~ color_contains(.x, "black", "hsv")),
    # All a certain color
    all_contains_yellow = map_dbl(colors, ~ color_all_contains(.x, "yellow", "hsv")),
    all_contains_orange = map_dbl(colors, ~ color_all_contains(.x, "orange", "hsv")),
    all_contains_red = map_dbl(colors, ~ color_all_contains(.x, "red", "hsv")),
    all_contains_purple = map_dbl(colors, ~ color_all_contains(.x, "purple", "hsv")),
    all_contains_blue = map_dbl(colors, ~ color_all_contains(.x, "blue", "hsv")),
    all_contains_green = map_dbl(colors, ~ color_all_contains(.x, "green", "hsv")),
    all_contains_brown = map_dbl(colors, ~ color_all_contains(.x, "brown", "hsv")),
    all_contains_white = map_dbl(colors, ~ color_all_contains(.x, "white", "hsv")),
    all_contains_black = map_dbl(colors, ~ color_all_contains(.x, "black", "hsv")),
    # Is it linear in a perceptually uniform space
    linear = map_dbl(colors, ~ linear(.x, "hunterlab")),
    linear_deutan = map_dbl(colors, ~ linear(deutan(.x), "hunterlab")),
    linear_protan = map_dbl(colors, ~ linear(protan(.x), "hunterlab")),
    linear_tritan = map_dbl(colors, ~ linear(tritan(.x), "hunterlab")),
    # Twice linear
    twice_linear = map_dbl(colors, ~ linear_split(.x, "hunterlab")),
    twice_linear_deutan = map_dbl(colors, ~ linear_split(deutan(.x), "hunterlab")),
    twice_linear_protan = map_dbl(colors, ~ linear_split(protan(.x), "hunterlab")),
    twice_linear_tritan = map_dbl(colors, ~ linear_split(tritan(.x), "hunterlab")),
```

```
# Min distance between points
min_distance = map_dbl(colors, ~ min_distance(.x, "hunterlab")),
min_distance_deutan = map_dbl(colors, ~ min_distance(deutan(.x), "hunterlab")),
min_distance_protan = map_dbl(colors, ~ min_distance(protan(.x), "hunterlab")),
min_distance_tritan = map_dbl(colors, ~ min_distance(tritan(.x), "hunterlab")),
# Max distance between points
max_distance = map_dbl(colors, ~ max_distance(.x, "hunterlab")),
max_distance_deutan = map_dbl(colors, ~ max_distance(deutan(.x), "hunterlab")),
max_distance_protan = map_dbl(colors, ~ max_distance(protan(.x), "hunterlab")),
max_distance_tritan = map_dbl(colors, ~ max_distance(tritan(.x), "hunterlab")),
# IQR distance between points
iqr_distance = map_dbl(colors, ~ iqr_distance(.x, "hunterlab")),
iqr_distance_deutan = map_dbl(colors, ~ iqr_distance(deutan(.x), "hunterlab")),
iqr_distance_protan = map_dbl(colors, ~ iqr_distance(protan(.x), "hunterlab")),
iqr_distance_tritan = map_dbl(colors, ~ iqr_distance(tritan(.x), "hunterlab"))
)
```

# SCALING

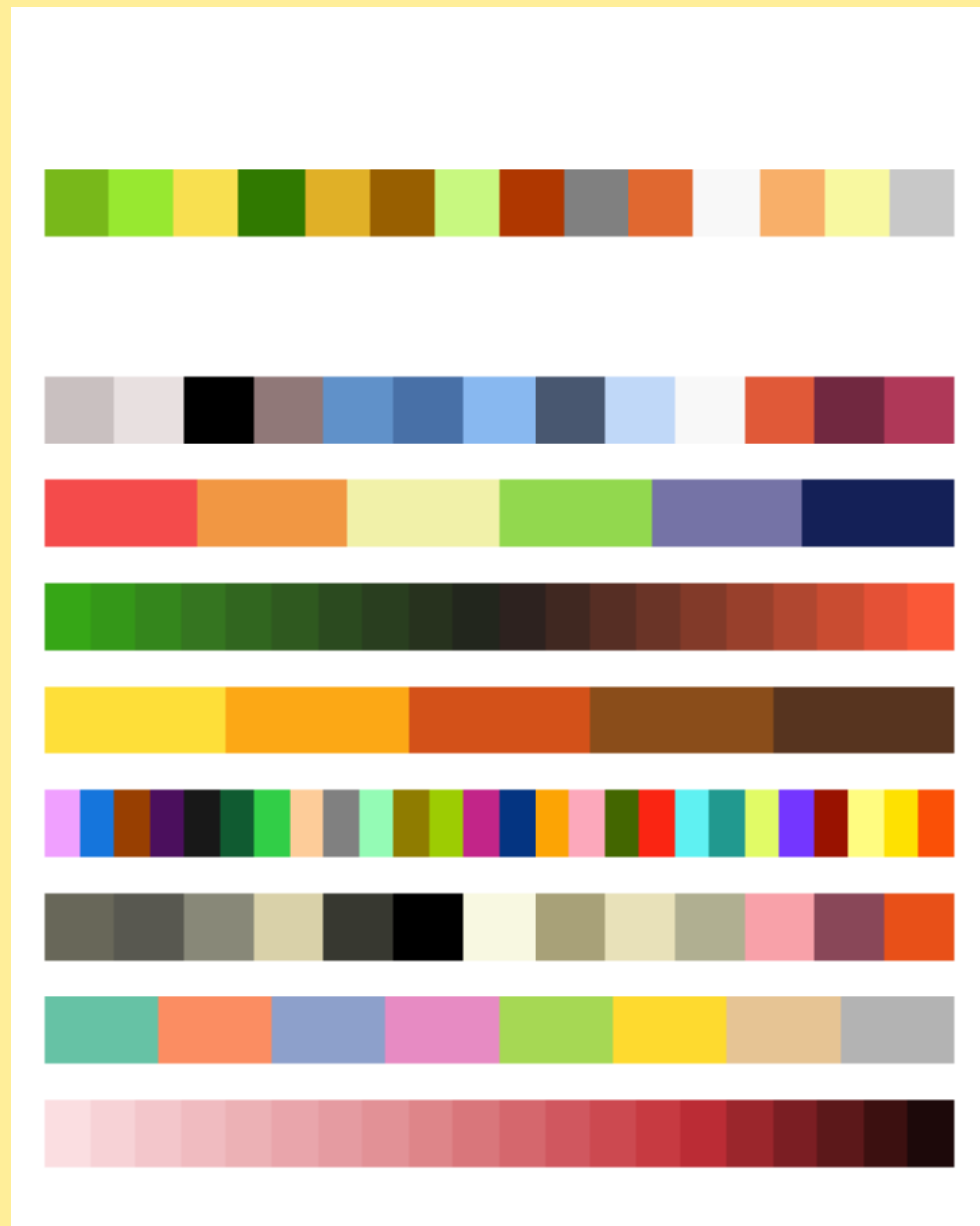
```
data_meta <- data_num %>%  
  mutate(id = as.character(str_glue("{package}-{palette}-{group}"))))  
  
data_prescaled <- data_meta %>%  
  select(-package, -palette, -colors) %>%  
  mutate_at(vars(type, group), funs(as.factor))  
  
library(recipes)  
  
re <- recipe(id ~ ., data = data_prescaled) %>%  
  step_dummy(type, group) %>%  
  step_center(all_predictors()) %>%  
  step_scale(all_predictors()) %>%  
  prep()  
  
data_scaled <- bake(re, newdata = data_prescaled)
```



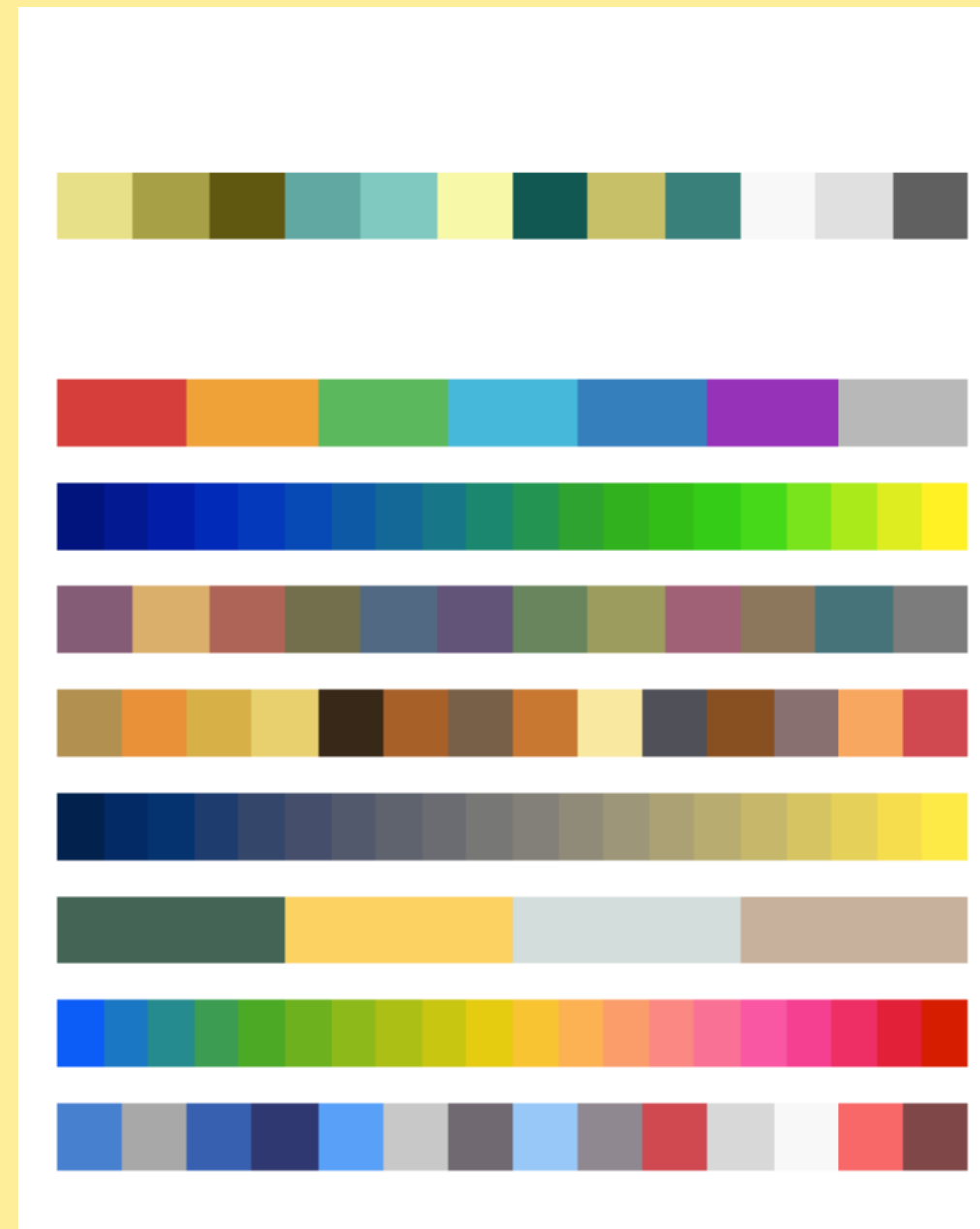
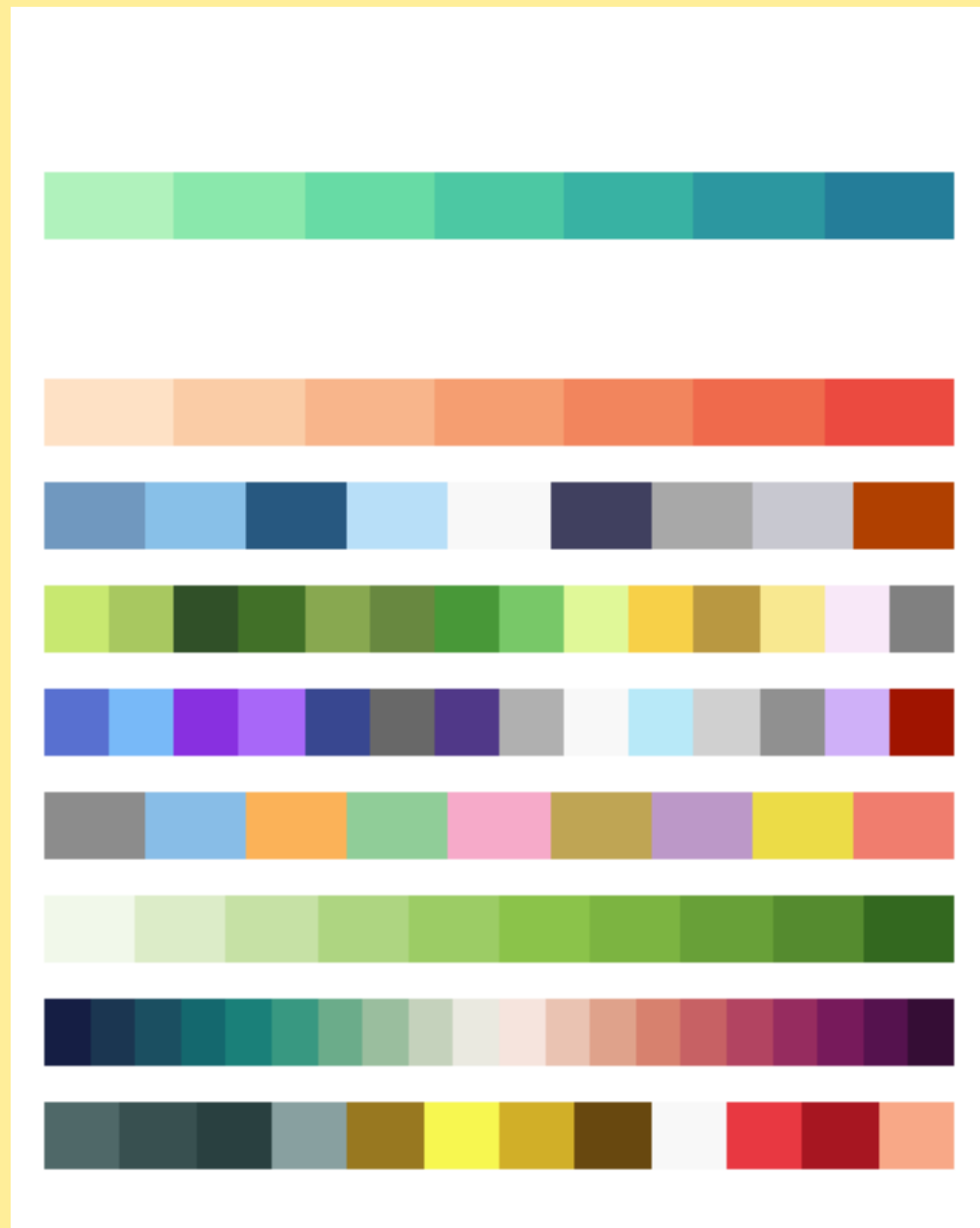
# MODELING

- Simple nearest neighbors method
- Using Euclidean distance
- Same weights for all predictors due to scaling

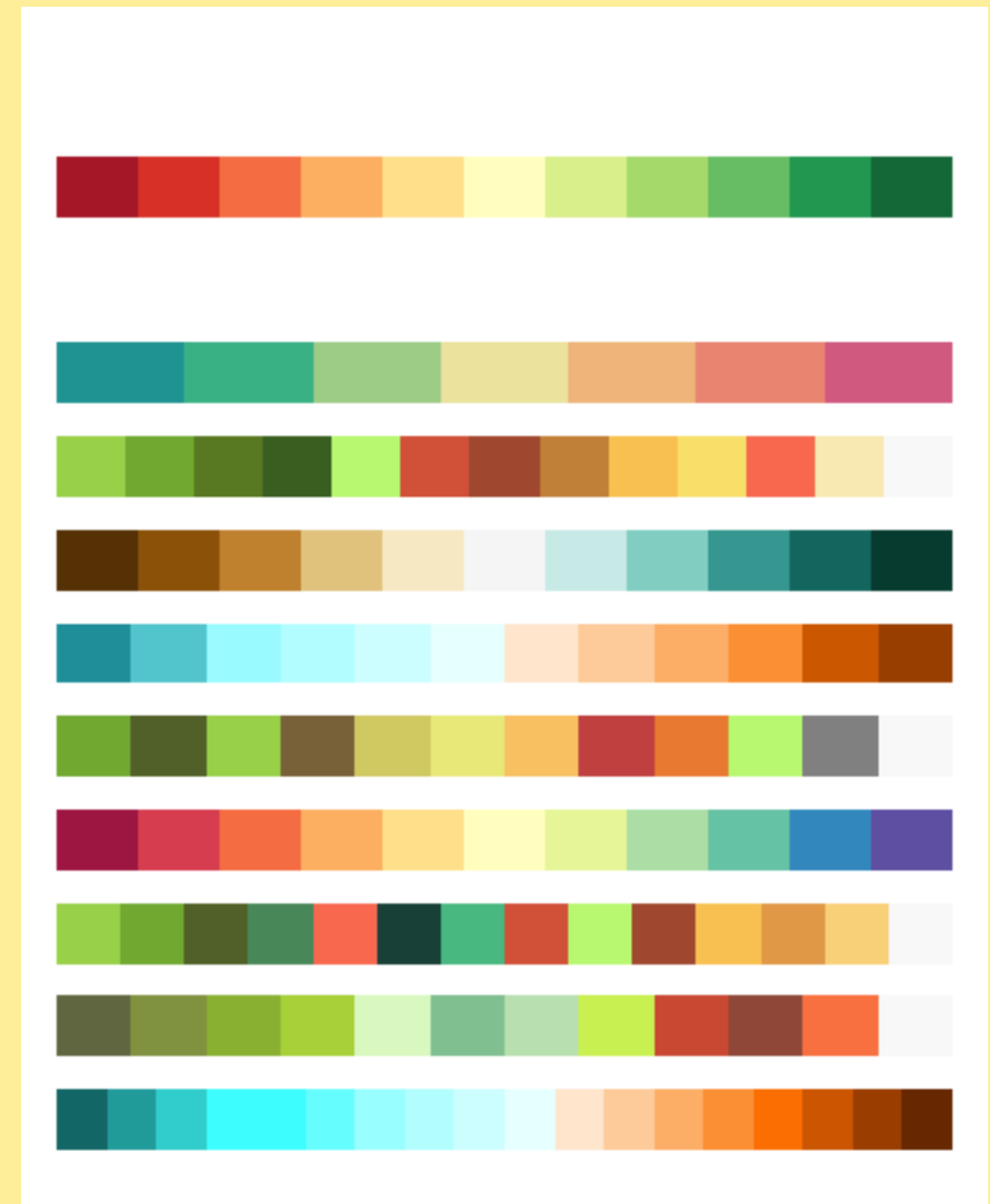
# BASELINE



# BASELINE

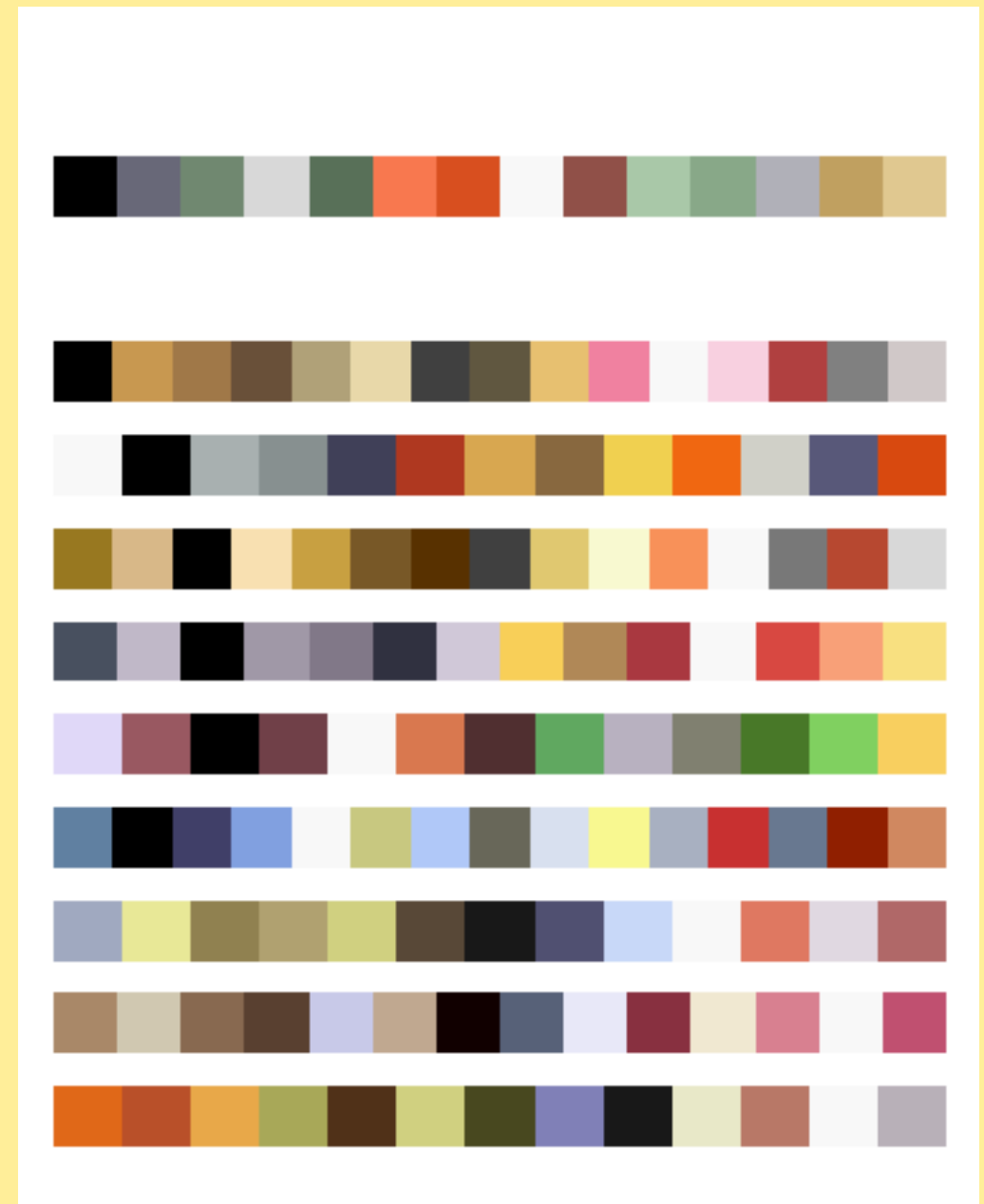
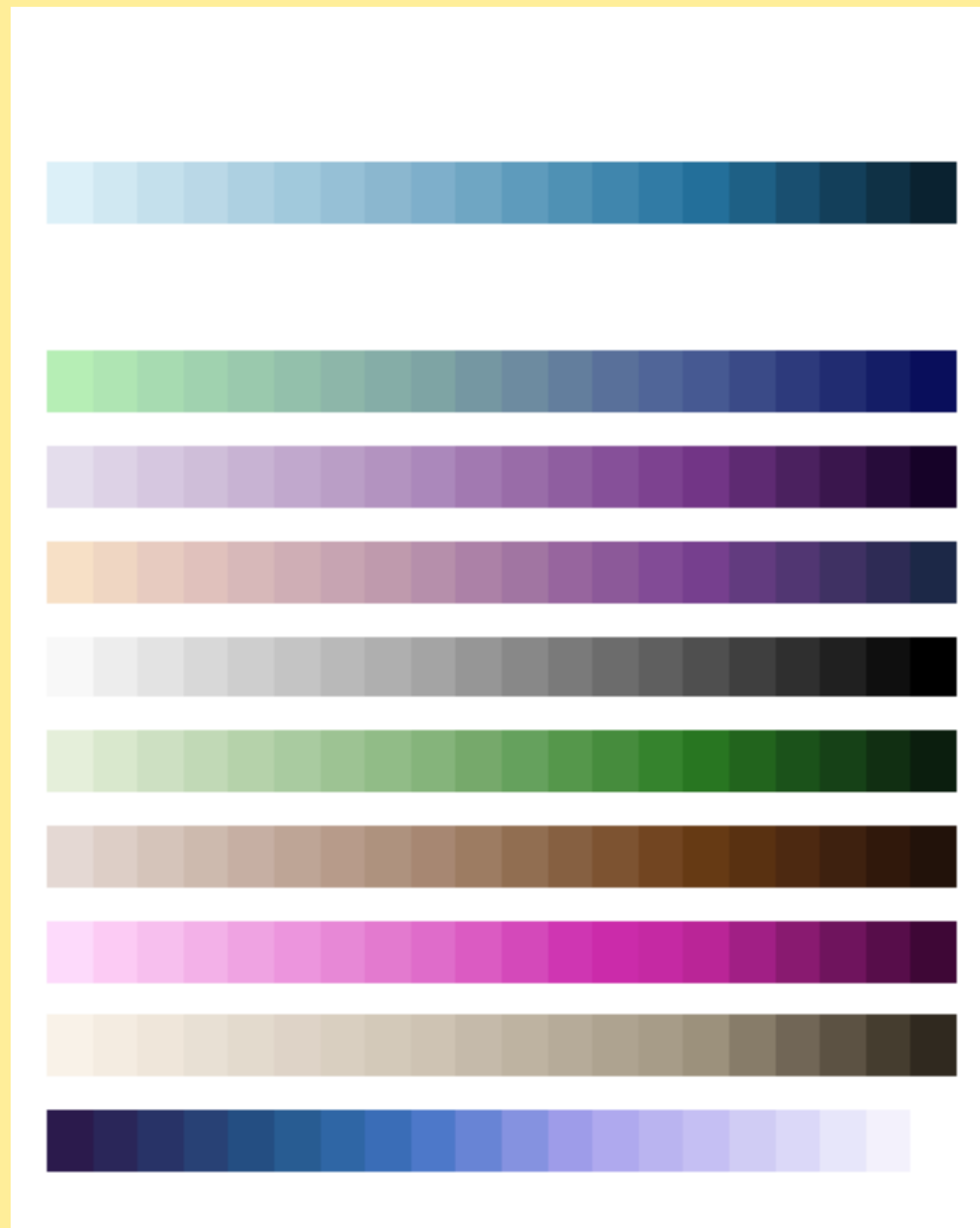


# NEAREST NEIGHBORS





# NEAREST NEIGHBORS



# LIVE DEMO

With audience participation