Palette Swatch

FCA Collin, Ph.D.

Wednesday, June 16, 2021

Contents

2021-03-19, FCA Collin

Good toy example to practice grid *viewport* and *grob* trees.

Use Cases

```
palette_swatch(viridis::viridis_pal(option = "A")(20))
```

```
palette_swatch(viridis::viridis_pal(option = "B")(10))
```

```
palette_swatch(viridis::viridis_pal(option = "C")(5))
```

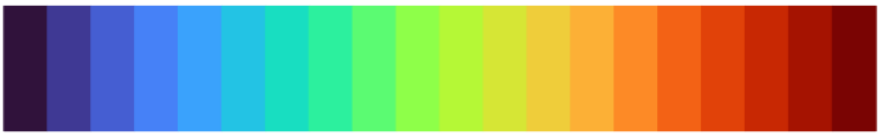
```
palette_swatch(viridis::viridis_pal(option = "D")(20))
```

```
palette_swatch(viridis::viridis_pal(option = "E")(20))
```

```
palette_swatch(viridis::viridis_pal(option = "F")(10))
```

```
palette_swatch(viridis::viridis_pal(option = "G")(5))
```

```
palette_swatch(viridis::viridis_pal(option = "H")(20))
```



Definition

```
#' Palette Swatch
#'
#' Represent a color palette.
#'
#' @param ... (`atomic`)\cr valid color(s) (according to grid)
#' @param draw (`logical`)
#' @export
#' @examples
#'
#' palette_swatch("gray", "red", "gray", NA, "blue")
#'
palette_swatch <- function(..., draw = TRUE) {

  colors <- list(...)
  lapply(colors, function(x) assertthat::assert_that(is.atomic(x)))

  colors <- unlist(colors)
  nm <- paste(colors, seq_along(colors), sep = "_")
  vp <- grid::vpTree(
    parent = grid::viewport(name = "page", width = 0.95, height = 0.95),
    children = do.call(
      grid::vpList,
      Map(
        nm = nm,
        x = seq_along(colors) / (length(colors)),
        width = 1 / length(colors),
        f = function(nm, x, width) {
          grid::viewport(
            name = nm,
            x = x,
            width = width,
            just = "right"
          )
        }
      )
    )
  )

  gr <- do.call(
    grid::gList,
    Map(
      colors,
      nm,
      f = function(colors, nm) {
```

```
        grid::gTree(
          vp = nm,
          children = grid::gList(
            grid::rectGrob(gp = grid::gpar(fill = colors, col = colors))
          )
        )
      }
    )
  )

  gr <- grid::gTree(
    childrenvp = vp,
    children = grid::gList(
      grid::gTree(
        vp = "page",
        children = gr
      )
    )
  )

  if (draw) {
    grid::grid.newpage()
    grid::grid.draw(gr)
  } else {
    invisible()
  }
}
```

Unit tests

```
library(testthat)
test_that("palette_swatch works if atomic", {
  expect_silent(palette_swatch("gray", draw = FALSE))
  expect_silent(palette_swatch("gray", c("blue", "green"), draw = FALSE))
  expect_silent(palette_swatch("gray", "red", NA, "blue", draw = FALSE))
  })


## Test passed


test_that("palette_swatch works if a color is repeated", {
  expect_silent(palette_swatch(rep("gray", 10), draw = FALSE))
  expect_silent(palette_swatch("red", "red", draw = FALSE))
```

```
  })


## Test passed


test_that("palette_swatch fail if non-atomic", {
  expect_error(palette_swatch("gray", iris, draw = FALSE))
  })


## Test passed


sessionInfo()


## R version 4.0.4 (2021-02-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 10 (buster)
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.3.5.so
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets
methods   base
##
## other attached packages:
## [1] testthat_3.0.2
##
## loaded via a namespace (and not attached):
##  [1] pillar_1.6.0      compiler_4.0.4    highr_0.9
viridis_0.6.0
```

```
##  [5] tools_4.0.4         pkgload_1.1.0       digest_0.6.27
evaluate_0.14
##  [9] lifecycle_1.0.0     tibble_3.1.1        gtable_0.3.0
viridisLite_0.4.0
## [13] pkgconfig_2.0.3     rlang_0.4.11        rstudioapi_0.13
cli_2.5.0
## [17] DBI_1.1.1           yaml_2.2.1          xfun_0.22
gridExtra_2.3
## [21] withr_2.4.1         stringr_1.4.0       dplyr_1.0.5
knitr_1.33
## [25] desc_1.3.0          generics_0.1.0      vctrs_0.3.8
rprojroot_2.0.2
## [29] grid_4.0.4          tidyselect_1.1.1    glue_1.4.2
R6_2.5.0
## [33] fansi_0.4.2         rmarkdown_2.6       ggplot2_3.3.3
purrr_0.3.4
## [37] magrittr_2.0.1      ps_1.6.0            scales_1.1.1
ellipsis_0.3.2
## [41] htmltools_0.5.1.1   assertthat_0.2.1    colorspace_2.0-1
utf8_1.2.1
## [45] stringi_1.5.3       munsell_0.5.0       crayon_1.4.1
```