Package 'swatches'

October 14, 2022

Type Package

Title Read, Inspect, and Manipulate Color Swatch Files

Version 0.5.0

Date 2017-12-20

Maintainer Bob Rudis <bob@rud.is>

Description There are numerous places to create and download color
palettes. These are usually shared in 'Adobe' swatch file formats of
some kind. There is also often the need to use standard palettes
developed within an organization to ensure that aesthetics are carried over
into all projects and output. Now there is a way to read these swatch
files in R and avoid transcribing or converting color values by hand or
or with other programs. This package provides functions to read and
inspect 'Adobe Color' ('ACO'), 'Adobe Swatch Exchange' ('ASE'), 'GIMP Palette' ('GPL'),
'OpenOffice' palette ('SOC') files and 'KDE Palette' ('colors') files. Detailed
descriptions of 'Adobe Color' and 'Swatch Exchange' file formats as well as other
swatch file formats can be found at
<http://www.selapa.net/swatches/colors/fileformats.php>.

URL https://github.com/hrbrmstr/swatches

BugReports https://github.com/hrbrmstr/swatches/issues

License MIT + file LICENSE

Encoding UTF-8

NeedsCompilation yes

Imports httr, pack, stringr, xml2, tools, colorspace, grDevices,
graphics, methods, stats

Suggests testthat, covr

Depends R (>= 3.2.0)

RoxygenNote 6.0.1.9000

Author Bob Rudis [aut, cre] (<https://orcid.org/0000-0001-5670-2640>)

Repository CRAN

Date/Publication 2017-12-21 15:43:03 UTC

R topics documented:

---

read_aco   *Read colors from Adobe Color (ACO) files*

Description

Given a path or URL to an .aco file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_aco(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to an ACO file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about ACO file processing

Note

When using named color palettes in a ggplot2 scale_ context, you must unname, set use_names to FALSE or override their names to map to your own factor levels.

Examples

```
# built-in palette
eighties <- read_aco(system.file("palettes", "tomorrow_night_eighties.aco", package="swatches"))
print(eighties)
show_palette(eighties)

# from the internet directly
## Not run:
tomorrow_night <- read_aco("https://bit.ly/tomorrow-night-aco")
print(tomorrow_night)
show_palette(tomorrow_night)
```

```
## End(Not run)
```

---

read_ase   *Read colors from Adobe Swatch Exchange (ASE) files*

Description

Given a path or URL to an `.ase` file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_ase(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to an ASE file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about ASE file processing

Note

When using named color palettes in a `ggplot2` `scale_` context, you must unname, set `use_names` to `FALSE` or override their names to map to your own factor levels. Also, Neither Lab nor greyscale colors are supported.

Examples

```
# built-in palette
keep_the_change <- read_ase(system.file("palettes", "keep_the_change.ase", package="swatches"))
print(keep_the_change)
show_palette(keep_the_change)

# from the internet directly
## Not run:
github <- "https://github.com/picwellwisher12pk/en_us/raw/master/Swatches/Metal.ase"
metal <- read_ase(github)
print(metal)
show_palette(metal)

## End(Not run)
```

---

read_gpl   *Read colors from GIMP Palette (GPL) files*

Description

Given a path or URL to an `.gpl` file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_gpl(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to a GPL file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about GPL file processing

Note

When using named color palettes in a ggplot2 scale_ context, you must unname, set use_names to FALSE or override their names to map to your own factor levels. Also, Neither Lab nor greyscale colors are supported.

Examples

```
# built-in palette
gimp16 <- read_gpl(system.file("palettes", "base16.gpl", package="swatches"))
print(gimp16)
show_palette(gimp16)

# from the internet directly
## Not run:
bright <- read_gpl(URL)
print(bright)
show_palette(bright)

## End(Not run)
```

---

read_kde   *Read colors from KDE Palette (colors) files*

Description

Given a path or URL to an `.colors` file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_kde(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to a GPL file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about GPL file processing

Note

When using named color palettes in a ggplot2 `scale_` context, you must unname, set `use_names` to `FALSE` or override their names to map to your own factor levels.

Examples

```
# built-in palette
fourty <- read_kde(system.file("palettes", "fourty.colors", package="swatches"))
print(fourty)
#show_palette(fourty)

# show_palette(bright)
```

---

read_palette   *Read colors from palette files*

Description

Given a path or URL to an palette file, this function will attempt to determine which palette file format to read by the file type and return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_palette(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

`path`	partial or full file path or URL to a GPL file
`use_names`	add color names to the vector (defaults to `TRUE`). See NOTE
`.verbose`	show extra information about GPL file processing

Note

When using named color palettes in a `ggplot2 scale_` context, you must unname, set `use_names` to `FALSE` or override their names to map to your own factor levels.

`read_soc`   *Read colors from OpenOffice Palette (SOC) files*

Description

Given a path or URL to an `.soc` file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_soc(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

`path`	partial or full file path or URL to a GPL file
`use_names`	add color names to the vector (defaults to `TRUE`). See NOTE
`.verbose`	show extra information about GPL file processing

Note

When using named color palettes in a `ggplot2 scale_` context, you must unname, set `use_names` to `FALSE` or override their names to map to your own factor levels.

Examples

```
# built-in palette
soc_file <- system.file("palettes", "ccooo.soc", package="swatches")
system(sprintf("cat %s", soc_file))
ccooo <- read_soc(soc_file)
print(ccooo)
show_palette(ccooo)

# from the internet directly
## Not run:
galaxy <- read_soc("https://www.openoffice.org/ui/VisualDesign/docs/colors/galaxy.soc")
print(galaxy)
show_palette(galaxy)

## End(Not run)
```

---

show_palette *Display a color palette*

Description

Given a character vector (hex RGB values), display palette in graphics window.

Usage

```
show_palette(palette)
```

Arguments

palette   vector of character hex RGB values

Examples

```
# built-in palette
keep_the_change <- read_ase(system.file("palettes",
                            "keep_the_change.ase", package="swatches"))
print(keep_the_change)
# show_palette(keep_the_change)
```

---

swatches *Read, Inspect, and Manipulate Color Swatch Files*

Description

Read, Inspect, and Manipulate Color Swatch Files

---

trim_palette *Try to intelligently reduce a large palette down to a reasonable smaller set of colors*

Description

Given a palette and a desired number of colors to use, this function will compute CIEDE2000 and attempt to reduce the input set to a disctinct smaller set of colors based on color distances.

Usage

```
trim_palette(pal, n = 5)
```

Arguments

`pal`	input palette to reduct
`n`	number of desired colors

Value

vector of `n` colors from `pal`

Note

internal CIEDE2000 color distance implementation by Gaurav Sharma & Maynard P Baalthazar

Index