

# Package: lterpalettefinder (via r-universe)

October 25, 2024

**Type** Package

**Title** Extract Color Palettes from Photos and Pick Official LTER Palettes

**Version** 1.1.0.900

**Maintainer** Nicholas J Lyon <njl Lyon@alumni.iastate.edu>

**Description** Allows identification of palettes derived from LTER (Long Term Ecological Research) photographs based on user criteria. Also facilitates extraction of palettes from users' photos directly.

**License** BSD\_3\_clause + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Language** en-US

**BugReports** <https://github.com/lter/lterpalettefinder/issues>

**RoxygenNote** 7.2.3

**Depends** R (>= 3.5)

**Imports** dplyr, ggplot2, graphics, grDevices, jpeg, magick, magrittr, png, stats, tools, tidyr, tiff

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Repository** <https://lter.r-universe.dev>

**RemoteUrl** <https://github.com/lter/lterpalettefinder>

**RemoteRef** HEAD

**RemoteSha** 554ee3b934fb4e4ca038092e624c381807340f85

## Contents

palette_check . . . . .	2
palette_demo . . . . .	3
palette_extract . . . . .	4
palette_find . . . . .	4
palette_ggdemo . . . . .	5
palette_options . . . . .	6
palette_sort . . . . .	7
palette_subsample . . . . .	7
<b>Index</b>	<b>9</b>

---

palette_check	<i>Check Hexadecimal Code Formatting</i>
---------------	--

---

### Description

Accepts the hexadecimal code vector and tests if it is formatted correctly

### Usage

```
palette_check(palette)
```

### Arguments

palette (character) Vector of hexadecimal codes returned by 'palette\_extract()', '...\_sort()', or '...\_subsample()'

### Value

An error message or nothing

### Examples

```
# Check for misformatted hexcodes
palette_check(palette = c("#8e847a", "#9fc7f2"))
```

## Description

Accepts the hexadecimal code vector returned by ‘palette\_extract()’, ‘...\_sort()’, or ‘...\_subsample()’ and creates a base plot of all returned colors labeled with their HEX codes. This will facilitate (hopefully) your selection of which of the 25 colors you would like to use in a given context.

## Usage

```
palette_demo(  
  palette,  
  export = FALSE,  
  export_name = "my_palette",  
  export_path = getwd()  
)
```

## Arguments

palette	(character) Vector of hexadecimal codes like that returned by ‘palette_extract()’, ‘..._sort()’, or ‘..._subsample()’
export	(logical) Whether or not to export the demo plot
export_name	(character) Name for exported plot
export_path	(character) File path to save exported plot (defaults to working directory)

## Value

A plot of the retrieved colors labeled with their HEX codes

## Examples

```
# Extract colors from a supplied image  
my_colors <- palette_extract(  
  image = system.file("extdata", "lyon-fire.png", package = "lterpalettefinder")  
)  
  
# Plot that result  
palette_demo(palette = my_colors)
```

---

palette\_extract      *Extract Hexadecimal Codes from an Image*

---

### Description

Retrieves hexadecimal codes for the colors in an image file. Currently only PNG, JPEG, TIFF, and HEIC files are supported. The function automatically removes dark colors and removes 'similar' colors to yield 25 colors from which you can select the subset that works best for your visualization needs. Note that photos that are very dark may return few viable colors.

### Usage

```
palette_extract(image, sort = FALSE, progress_bar = TRUE)
```

### Arguments

image            (character) Name/path to PNG, JPEG, TIFF, or HEIC file from which to extract colors

sort            (logical) Whether extracted HEX codes should be sorted by hue and saturation

progress\_bar    (logical) Whether to 'message' a progress bar

### Value

(character) Vector containing all hexadecimal codes remaining after extraction and removal of 'dark' and 'similar' colors

### Examples

```
# Extract colors from a supplied image
my_colors <- palette_extract(image = system.file("extdata", "lyon-fire.png",
package = "lterpalettefinder"), sort = TRUE, progress_bar = FALSE)

# Plot that result
palette_demo(palette = my_colors)
```

---

palette\_find      *Find a Long Term Ecological Research (LTER) Site-Derived Palette*

---

### Description

From a dataframe of all possible palettes (updated periodically so check back!) specify the characteristics of the palette you want and retrieve the palettes that match those criteria. Can specify by number of colors in the palette, type of palette (e.g., qualitative, sequential, etc.), or which LTER site the palette came from.

**Usage**

```
palette_find(site = "all", name = "all", type = "all")
```

**Arguments**

site	(character) Vector of three-letter LTER site abbreviations for which to return palettes or "all" or "LTER" for the LTER logo colors
name	(character) Vector of palette names (if known) for which to return palettes
type	(character) Vector of palette types (i.e., qualitative, tricolor, sequential, or diverging) for which to return palettes

**Value**

(dataframe / character) If more than one palette, a dataframe is returned; if exactly one palette, a character vector is returned

**Examples**

```
# Look at all palette options by calling the function without specifying arguments
lterpalettefinder::palette_find()

# What if our query returns NO options?
palette_find(name = "no such name")

# What if our query returns MULTIPLE options?
palette_find(site = "sbc")

# What if our query returns JUST ONE option? (this is desirable)
palette_find(name = "salamander")
```

---

 palette\_ggdemo

---

*Demonstrate Extracted Palette with HEX Labels - ggplot2 Edition*


---

**Description**

Accepts the hexadecimal code vector returned by ‘palette\_extract()’, ‘...\_sort()’, or ‘...\_subsample()’ and creates a simple plot of all returned colors labeled with their HEX codes. This will facilitate (hopefully) your selection of which of the 25 colors you would like to use in a given context.

**Usage**

```
palette_ggdemo(palette)
```

**Arguments**

palette	(character) Vector of hexadecimal codes returned by ‘palette_extract()’, ‘..._sort()’, or ‘..._subsample()’
---------	---

**Value**

A ggplot2 plot

**Examples**

```
# Extract colors from a supplied image
my_colors <- palette_extract(image = system.file("extdata", "lyon-fire.png",
package = "lterpalettefinder"))

# Plot that result
palette_ggdemo(palette = my_colors)
```

---

palette\_options

*LTER Palette Options*

---

**Description**

For each palette, data includes the photographer, LTER site, number of included colors, and the hexadecimal codes for each color (data are in 'wide' format)

**Usage**

```
palette_options
```

**Format**

A dataframe with 14 variables and one row per palette (currently 14 rows)

**photographer** name of the photographer who took the picture

**palette\_full\_name** concatenation of LTER site and palette name, separated by a hyphen

**lter\_site** three-letter LTER site name abbreviation

**palette\_name** a unique-within site-name for each palette based on the picture's content

**palette\_type** either "qualitative", "sequential", or "diverging" depending on the pattern of colors in the palette

**color\_...** the hexadecimal code for colors 1 through n for each palette

**Source**

Lyon, N. J., De La Rosa, G. 2022.

---

palette\_sort                      *Sort Hexadecimal Codes by Hue and Saturation*

---

**Description**

Sorts hexadecimal codes retrieved by 'palette\_extract()' by hue and saturation. This allows for reasonably good identification of 'similar' colors in the way that a human eye would perceive them (as opposed to a computer's representation of colors).

**Usage**

```
palette_sort(palette)
```

**Arguments**

palette                      (character) Vector returned by 'palette\_extract()'

**Value**

(character) Vector containing all hexadecimal codes returned by 'palette\_extract()'

**Examples**

```
# Extract colors from a supplied image
my_colors <- palette_extract(image = system.file("extdata", "lyon-fire.png",
package = "lterpalettefinder"))

# Plot that result
palette_demo(palette = my_colors)

# Now sort
sort_colors <- palette_sort(palette = my_colors)

# And plot again to show change
palette_demo(palette = sort_colors)
```

---

palette\_subsample                *Randomly Subsample HEX Codes*

---

**Description**

Randomly subsample the HEX codes returned by 'palette\_extract()' or 'palette\_sort()' to desired length. Can also set random seed for reproducibility.

**Usage**

```
palette_subsample(palette, wanted = 5, random_seed = 36)
```

**Arguments**

`palette` (character) Vector of hexadecimal codes like those returned by `'palette_extract()'` or `'palette_sort()'`

`wanted` (numeric) Integer for how many colors should be returned

`random_seed` (numeric) Integer for `'base::set.seed()'`

**Value**

(character) Vector of hexadecimal codes of user-specified length

**Examples**

```
# Extract colors from a supplied image
my_colors <- palette_extract(image = system.file("extdata", "lyon-fire.png",
package = "lterpalettefinder"))

# Plot that result
palette_ggdemo(palette = my_colors)

# Now randomly subsample
random_colors <- palette_subsample(palette = my_colors, wanted = 5)

# And plot again to show change
palette_ggdemo(palette = random_colors)
```



# Index

## \* datasets

- palette\_options, 6
- palette\_check, 2
- palette\_demo, 3
- palette\_extract, 4
- palette\_find, 4
- palette\_ggdemo, 5
- palette\_options, 6
- palette\_sort, 7
- palette\_subsample, 7