Package: EraBrewer (via r-universe)

May 30, 2026

Type Package

Title Color Palettes from the Album Covers of Each Taylor Swift Era

Version 0.2.0

URL https://s-m.ac/EraBrewer/, https://github.com/mathias-sm/EraBrewer

BugReports https://github.com/mathias-sm/EraBrewer/issues

Description Provides discrete and continuous color palettes derived from the album cover artwork of each Taylor Swift Era. Each palette ships with a curated order-of-use so that smaller discrete subsets remain harmonious, and supports continuous interpolation via 'grDevices::colorRampPalette()' for arbitrary 'n'. Designed to plug into 'ggplot2' workflows through standard manual and gradient scales.

License CC0

Encoding UTF-8

Language en-US

Imports ggplot2, grDevices

Config/roxygen2/version 8.0.0

Suggests knitr, rmarkdown, ragg

VignetteBuilder knitr

Repository https://mathias-sm.r-universe.dev

Date/Publication 2026-05-23 19:22:03 UTC

RemoteUrl https://github.com/mathias-sm/erabrewer

RemoteRef HEAD

RemoteSha 6e34f2b08783bbee00c2ec50088526c5cd88339e

Contents

`era.brewer` *Generate a color palette*

Description

Returns a vector of colors drawn from one of the palettes in `EraPalettes`. Supports both discrete (curated subsets) and continuous (interpolated) palettes.

Usage

```
era.brewer(
  palette_name,
  n,
  type = c("discrete", "continuous"),
  direction = c(1, -1),
  override_order = FALSE,
  return_hex = FALSE
)
```

Arguments

`palette_name`	Character. Name of the palette; must be one of `names(EraPalettes)`.
`n`	Integer. Number of colors to return. Defaults to the full palette length.
`type`	One of `"discrete"` or `"continuous"`. If omitted, chosen automatically: `"continuous"` when `n` exceeds the palette length, otherwise `"discrete"`.
`direction`	`1` for the standard order, `-1` for reversed.
`override_order`	Logical. If `TRUE`, return colors in their stored order rather than the curated order-of-use for the requested `n`.
`return_hex`	Logical. If `TRUE`, also prints the hex codes.

Value

An object of class `"palette"`: a character vector of hex codes with the palette name stored as an attribute.

Examples

```
# Discrete palette using the curated order-of-use
print(era.brewer("Lover2", n = 3))

# Continuous interpolation when n exceeds the stored palette length
print(era.brewer("Showgirl2", n = 50, type = "continuous"))

# Reverse direction
print(era.brewer("Fearless", direction = -1))
```

```
# Plug into a ggplot2 manual scale
library(ggplot2)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
  geom_point() +
  scale_color_manual(values = era.brewer("Lover2", n = 3))
```

`EraPalettes` *Era Palettes*

Description

A named list of color palettes inspired by Taylor Swift's eras. Each entry is a list of length two: a character vector of hex codes, and an integer vector giving the preferred order in which the colors should be drawn for discrete palettes of increasing size.

Usage

```
EraPalettes
```

Format

A named list with one entry per palette.

Value

A named list; each element is itself a list of length two: a character vector of hex codes and an integer vector giving the curated order-of-use for discrete subsets.

Examples

```
names(EraPalettes)
EraPalettes[["Lover2"]]
```

`print.palette` *Print a palette*

Description

S3 method for objects of class `"palette"`. Renders the palette as a row of colored tiles labeled with the palette name.

Usage

```
## S3 method for class 'palette'
print(x, ...)
```

Arguments

x	A "palette" object as returned by era.brewer.
...	Unused.

Value

A ggplot object: a row of colored tiles labeled with the palette name. Rendered when auto-printed at the top level or when further composed with ggplot2 layers.

Examples

```
print(era.brewer("Lover2"))
```

Index