

Package ‘vangogh’

October 24, 2025

Title Vincent van Gogh Colour Palette Generator

Version 0.1.3

Description Provides 'ggplot2'-compatible colour palettes inspired by Vincent van Gogh's paintings. Each palette contains five colours, manually selected by hexadecimal values. Includes tools for assessing colour vision deficiency (CVD) accessibility.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports ggplot2, rlang, methods, graphics, stats, utils

Suggests scales, knitr, rmarkdown, colorspace, jsonlite, dplyr,
testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/cherylisabella/vangogh>

BugReports <https://github.com/cherylisabella/vangogh/issues>

NeedsCompilation no

Author Cheryl Isabella Lim [aut, cre]

Maintainer Cheryl Isabella Lim <cheryl.academic@gmail.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-10-24 16:00:02 UTC

Contents

check_all_vangogh_cvd	2
check_palette	3
check_vangogh_cvd	3
compare_palettes	4
get_cvd_safe_palettes	5

print_cvd_badge	6
safe_vangogh_palette	7
scale_color_vangogh	7
scale_fill_vangogh	8
summarize_cvd_accessibility	8
theme_vangogh	9
vangogh_colors	9
vangogh_cvd_scores	10
vangogh_export	11
vangogh_interpolate	11
vangogh_palette	12
vangogh_palettes	12
vangogh_palette_info	13
vangogh_palette_info_with_cvd	13
vangogh_suggest	14
viz_palette	14
Index	15

check_all_vangogh_cvd	<i>Batch Check CVD Accessibility for All Palettes</i>
-----------------------	---

Description

Runs CVD accessibility checks across all Van Gogh palettes and returns a summary data frame. Useful for generating documentation and identifying the most accessible palettes.

Usage

```
check_all_vangogh_cvd(simulate = FALSE)
```

Arguments

simulate Logical. If TRUE, displays simulations for each palette. Default FALSE.

Value

A data frame with CVD scores for all palettes

Examples

```
## Not run:
# Get scores for all palettes
all_scores <- check_all_vangogh_cvd()

# Find the most accessible palettes
library(dplyr)
all_scores %>%
  group_by(palette) %>%
```

```

    summarise(avg_distance = mean(min_distance)) %>%
    arrange(desc(avg_distance))

## End(Not run)

```

check_palette	<i>Check palette accessibility with colorblind simulations</i>
---------------	--

Description

Uses colorspace to simulate common forms of colorblindness.

Usage

```
check_palette(name, type = "discrete", n = NULL)
```

Arguments

name	Palette name (character)
type	Either "discrete" or "continuous" (default "discrete")
n	Number of colors for continuous palettes

Examples

```

## Not run:
vangogh::check_palette("StarryNight")

## End(Not run)

```

check_vangogh_cvd	<i>Check Color Vision Deficiency (CVD) Accessibility of Van Gogh Palettes</i>
-------------------	---

Description

Simulates how a Van Gogh palette appears under different types of color vision deficiency and provides accessibility scores. This function complements the existing check_palette() function by adding visual simulation and quantitative metrics.

Usage

```

check_vangogh_cvd(
  palette_name,
  n = NULL,
  simulate = TRUE,
  return_scores = FALSE
)

```

Arguments

palette_name	Character string specifying the palette name (e.g., "StarryNight")
n	Integer. Number of colors to extract from palette. Default is NULL (uses all colors).
simulate	Logical. If TRUE, displays simulations for each CVD type. Default TRUE.
return_scores	Logical. If TRUE, returns detailed scoring data. Default FALSE.

Details

This function evaluates palette accessibility across three main types of color vision deficiency:

- Deuteranopia (red-green, affects ~5% of males)
- Protanopia (red-green, affects ~2% of males)
- Tritanopia (blue-yellow, affects ~0.01% of population)

The function uses the colorspace package for CVD simulation and calculates minimum pairwise perceptual distance in CIELAB color space to assess distinguishability.

Value

If return_scores = TRUE, returns a data frame with CVD scores. Otherwise, displays visual simulations and prints a summary.

Examples

```
## Not run:
# Visual simulation of StarryNight palette
check_vangogh_cvd("StarryNight")

# Get detailed scores without plotting
scores <- check_vangogh_cvd("Irises", simulate = FALSE, return_scores = TRUE)

# Check subset of colors
check_vangogh_cvd("CafeTerrace", n = 3)

## End(Not run)
```

compare_palettes

Compare multiple Van Gogh palettes in a facet-style visualization

Description

Compare multiple Van Gogh palettes in a facet-style visualization

Usage

```
compare_palettes(  
  palettes,  
  show_hex = TRUE,  
  colorblind = FALSE,  
  type = "discrete",  
  n = NULL  
)
```

Arguments

palettes	Character vector of palette names
show_hex	Logical: display hex codes
colorblind	Logical: simulate colorblind view
type	"discrete" or "continuous"
n	Number of colors for continuous palettes

get_cvd_safe_palettes *Get CVD-Safe Van Gogh Palettes*

Description

Returns a list of Van Gogh palettes that meet minimum accessibility standards for color vision deficiency.

Usage

```
get_cvd_safe_palettes(  
  min_distance = 15,  
  cvd_types = c("deutan", "protan", "tritan")  
)
```

Arguments

min_distance	Numeric. Minimum CIELAB distance threshold. Default 15.
cvd_types	Character vector. Which CVD types to check. Options: "deutan", "protan", "tritan". Default checks all.

Value

Character vector of palette names that meet the criteria

Examples

```
## Not run:
# Get highly accessible palettes
safe_palettes <- get_cvd_safe_palettes(min_distance = 20)

# Use a safe palette
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
  geom_point() +
  scale_color_vangogh(safe_palettes[1])

## End(Not run)
```

print_cvd_badge

Print CVD Badge for README

Description

Generates markdown badges for palette CVD accessibility ratings. Useful for including in README or documentation.

Usage

```
print_cvd_badge(palette_name)
```

Arguments

palette_name Character string specifying the palette name

Value

Character string with markdown badge code

Examples

```
## Not run:
# Generate badge for StarryNight
print_cvd_badge("StarryNight")

# Generate badges for all palettes
for (p in names(vangogh_palettes)) {
  cat(p, ": ")
  print_cvd_badge(p)
}

## End(Not run)
```

safe_vangogh_palette	<i>Generate a colorblind-safe Van Gogh palette</i>
----------------------	--

Description

Returns the original palette (colorblind adjustment removed).

Usage

```
safe_vangogh_palette(name, type = "discrete", n = NULL, colorblind = FALSE)
```

Arguments

name	Palette name
type	Either "discrete" or "continuous"
n	Number of colors for continuous palettes
colorblind	Logical, kept for compatibility

scale_color_vangogh	<i>Scale color with Van Gogh palettes</i>
---------------------	---

Description

Scale color with Van Gogh palettes

Usage

```
scale_color_vangogh(name, discrete = TRUE, colorblind = FALSE, ...)  
scale_colour_vangogh(name, discrete = TRUE, colorblind = FALSE, ...)
```

Arguments

name	Palette name
discrete	Logical: use discrete scale
colorblind	Logical: use colorblind-safe colors
...	Additional arguments to ggplot2 scale function

scale_fill_vangogh	<i>Scale fill with Van Gogh palettes</i>
--------------------	--

Description

Scale fill with Van Gogh palettes

Usage

```
scale_fill_vangogh(name, discrete = TRUE, colorblind = FALSE, ...)
```

Arguments

name	Palette name
discrete	Logical: use discrete scale
colorblind	Logical: use colorblind-safe colors
...	Additional arguments to ggplot2 scale function

summarize_cvd_accessibility	<i>Summarize CVD Accessibility Across All Palettes</i>
-----------------------------	--

Description

Creates a summary table of CVD accessibility for all palettes, useful for documentation and choosing palettes.

Usage

```
summarize_cvd_accessibility()
```

Value

A data frame with palette names and overall CVD metrics

Examples

```
## Not run:
summary <- summarize_cvd_accessibility()
print(summary)

# Find best palettes
best <- summary[summary$overall_rating == "Excellent", ]
print(best)

## End(Not run)
```

theme_vangogh	<i>Theme inspired by Van Gogh (variants)</i>
---------------	--

Description

Theme inspired by Van Gogh (variants)

Usage

```
theme_vangogh(  
  base_size = 12,  
  base_family = "",  
  variant = c("classic", "light", "dark", "sketch")  
)
```

Arguments

base_size	numeric base font size
base_family	font family
variant	one of "classic", "light", "dark", "sketch"

vangogh_colors	<i>Return all Van Gogh palettes as a tidy data frame</i>
----------------	--

Description

Return all Van Gogh palettes as a tidy data frame

Usage

```
vangogh_colors(  
  n = NULL,  
  type = "discrete",  
  colorblind = FALSE,  
  add_metadata = FALSE  
)
```

Arguments

n	Number of colors per palette
type	"discrete" or "continuous"
colorblind	Logical (compatibility)
add_metadata	Logical: compute HCL metadata if colorspace available

vangogh_cvd_scores	<i>CVD Accessibility Scores for Van Gogh Palettes</i>
--------------------	---

Description

Pre-computed CVD accessibility scores for all Van Gogh palettes. This data is generated by running `check_all_vangogh_cvd()` and is included for quick reference without requiring the `colospace` package.

Usage

```
vangogh_cvd_scores
```

Format

A data frame with CVD accessibility metrics:

palette Name of the Van Gogh palette

cvd_type Type of colour vision deficiency

min_distance Minimum CIELAB distance between colours (higher = more distinguishable)

accessibility Rating: Poor, Fair, Good, or Excellent

overall_rating Average accessibility across all CVD types

Details

Distance interpretation:

- < 10: Poor - colours may be indistinguishable
- 10-20: Fair - some difficulty distinguishing
- 20-40: Good - generally distinguishable
- > 40: Excellent - highly distinguishable

Examples

```
## Not run:
# View CVD scores
data(vangogh_cvd_scores)

# Find most accessible palettes
palette_summary <- aggregate(
  min_distance ~ palette,
  data = vangogh_cvd_scores,
  FUN = mean
)
palette_summary[order(-palette_summary$min_distance), ]

# Filter by CVD type
```

```

deutan_scores <- vangogh_cvd_scores[
  vangogh_cvd_scores$cvd_type == "Deuteranopia (red-green)",
]
deutan_scores[order(-deutan_scores$min_distance), ]

## End(Not run)

```

vangogh_export	<i>Export palettes to JSON or CSV</i>
----------------	---------------------------------------

Description

Export palettes to JSON or CSV

Usage

```

vangogh_export(
  file,
  format = c("json", "csv"),
  n = NULL,
  type = "discrete",
  add_metadata = FALSE
)

```

Arguments

file	File path including filename
format	"json" or "csv"
n	Number of colors (for continuous palettes)
type	"discrete" or "continuous"
add_metadata	Logical: include HCL metadata if available

vangogh_interpolate	<i>Interpolate a Van Gogh palette</i>
---------------------	---------------------------------------

Description

Interpolate a Van Gogh palette

Usage

```
vangogh_interpolate(palette, n)
```

Arguments

palette	Vector of hex colors
n	Number of colors desired

vangogh_palette	<i>A Van Gogh color palette generator.</i>
-----------------	--

Description

These are some color palettes from a selection of Vincent van Gogh's paintings.

Usage

```
vangogh_palette(name, n, type = c("discrete", "continuous"))
```

Arguments

name	Name of desired palette. Choices are: StarryNight, StarryRhône, SelfPortrait, CafeTerrace, Eglise, Irises, SunflowersMunich, SunflowersLondon, Rest ,Bedroom , CafeDeNuit, Chaise, Shoes, Landscape, Cypressess
n	Number of colors desired. All palettes have a standard of 5 colors. If omitted, uses all colors.
type	Either "continuous" or "discrete". Use "continuous" to automatically interpolate between colours. @importFrom graphics rgb rect par image text

Value

A vector of colors.

Examples

```
vangogh_palette("StarryNight")
vangogh_palette("SelfPortrait")
vangogh_palette("Cypressess")
vangogh_palette("Cypressess", 3)

# If you want a continuous palette based on the colors already found in the preset
# palettes, you can interpolate between existing colours accordingly.
pal <- vangogh_palette(21, name = "StarryRhône", type = "continuous")
```

vangogh_palettes	<i>Complete list of palettes:</i>
------------------	-----------------------------------

Description

Use [vangogh_palette](#) to construct palettes of desired length.

Usage

```
vangogh_palettes
```

Format

An object of class list of length 15.

vangogh_palette_info	<i>Return palette info as a data frame with optional HCL</i>
----------------------	--

Description

Return palette info as a data frame with optional HCL

Usage

```
vangogh_palette_info(colorblind = FALSE, add_metadata = FALSE)
```

Arguments

colorblind	Logical (compatibility)
add_metadata	Logical: compute HCL hue/chroma/luminance if colorspace is installed

vangogh_palette_info_with_cvd	<i>Add CVD Information to Palette Documentation</i>
-------------------------------	---

Description

Helper function to add CVD accessibility information to your existing palette information functions.

Usage

```
vangogh_palette_info_with_cvd(palette_name)
```

Arguments

palette_name	Character string specifying the palette name
--------------	--

Value

List with palette info and CVD scores

Examples

```
## Not run:
# Get enhanced palette info with CVD data
info <- vangogh_palette_info_with_cvd("StarryNight")
print(info$cvd_accessibility$summary)

## End(Not run)
```

vangogh_suggest	<i>Suggest a palette based on number of colors</i>
-----------------	--

Description

Suggest a palette based on number of colors

Usage

```
vangogh_suggest(n = 5, type = "discrete")
```

Arguments

n	Number of colors needed
type	"discrete" or "continuous"

viz_palette	<i>Visualise a Van Gogh palette with optional colorblind simulation</i>
-------------	---

Description

Visualise a Van Gogh palette with optional colorblind simulation

Usage

```
viz_palette(
  name,
  show_hex = TRUE,
  colorblind = FALSE,
  type = "discrete",
  n = NULL
)
```

Arguments

name	Palette name
show_hex	Display hex codes (TRUE/FALSE)
colorblind	Show colorblind simulation (TRUE/FALSE)
type	Either "discrete" or "continuous"
n	Number of colors for continuous palettes

Index

* colors

vangogh_palette, 12

* datasets

vangogh_cvd_scores, 10

vangogh_palettes, 12

check_all_vangogh_cvd, 2

check_palette, 3

check_vangogh_cvd, 3

compare_palettes, 4

get_cvd_safe_palettes, 5

print_cvd_badge, 6

safe_vangogh_palette, 7

scale_color_vangogh, 7

scale_colour_vangogh
(scale_color_vangogh), 7

scale_fill_vangogh, 8

summarize_cvd_accessibility, 8

theme_vangogh, 9

vangogh_colors, 9

vangogh_cvd_scores, 10

vangogh_export, 11

vangogh_interpolate, 11

vangogh_palette, 12, 12

vangogh_palette_info, 13

vangogh_palette_info_with_cvd, 13

vangogh_palettes, 12

vangogh_suggest, 14

viz_palette, 14