

Package: PortinariColors (via r-universe)

October 29, 2026

Type Package

Title Color Palettes Inspired by the Paintings of Candido Portinari

Version 0.3.0.9000

Description Color palettes inspired by paintings of Candido Portinari held by the Sao Paulo Museum of Art (MASP) and by his Guerra (War) and Paz (Peace) panels at the United Nations Headquarters. Provides qualitative, diverging and sequential palettes for discrete and continuous data, ready-made 'ggplot2' color and fill scales, and a colorblind-safety check that simulates protanopia, deuteranopia and tritanopia following Machado, Oliveira and Fernandes (2009) [doi:10.1109/TVCG.2009.113](https://doi.org/10.1109/TVCG.2009.113).

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL <https://rafalopespx.github.io/PortinariColors/>,
<https://github.com/rafalopespx/PortinariColors>

BugReports <https://github.com/rafalopespx/PortinariColors/issues>

Imports ggplot2 (>= 3.4.0), grDevices, graphics, stats, utils

Suggests spelling, testthat (>= 3.1.5)

Config/testthat/edition 3

Config/Needs/website pkgdown, rmarkdown, knitr

Repository <https://rafalopespx.r-universe.dev>

Date/Publication 2026-10-19 05:51:17 UTC

RemoteUrl <https://github.com/rafalopespx/portinaricolors>

RemoteRef HEAD

RemoteSha b72729cbc89088743c36187cd3b3107fe0bb6d5c

Contents

portinari.cvd.check	2
portinari.pal	3
portinari.palettes	4
portinari.works	4
portinari_scales	5
show_portinari	6
theme_portinari	7

Index	9
--------------	----------

portinari.cvd.check	<i>Check colorblind safety of a Portinari palette</i>
---------------------	---

Description

Simulates protanopia, deuteranopia and tritanopia (Machado et al., 2009) and measures the smallest CIELAB color difference (CIE76 dE) within the discrete palette of n colors returned by `portinari.pal()`, for every n. A discrete palette of n colors is colorblind-safe when all three simulations keep that minimum at or above threshold.

Usage

```
portinari.cvd.check(palette = "cafe", threshold = 10)
```

Arguments

palette	Name of the palette. See <code>portinari.palettes()</code> .
threshold	Minimum acceptable dE. Default 10.

Value

A data frame with one row per palette size n (from 2 up to the palette length) and columns normal, protanopia, deuteranopia and tritanopia (minimum pairwise dE in each condition) and colorblind_safe.

Examples

```
portinari.cvd.check("inah")
```

portinari.pal	<i>Portinari color palette</i>
---------------	--------------------------------

Description

Returns a vector of hex colors from a Portinari palette.

Usage

```
portinari.pal(  
  palette = "cafe",  
  n,  
  type = c("discrete", "continuous"),  
  direction = 1,  
  colorblind = FALSE  
)
```

Arguments

palette	Name of the palette. See portinari.palettes() for choices.
n	Number of colors. Defaults to the length of the palette (or of its colorblind-safe subset when colorblind = TRUE).
type	"discrete" returns distinct colors and errors when n is larger than the palette. "continuous" interpolates. When omitted, the type is chosen from n.
direction	1 (default) for the standard order, -1 to reverse it.
colorblind	If TRUE, only colors that remain distinguishable under protanopia, deuteranopia and tritanopia simulation are used. See portinari.cvd.check() .

Details

Palettes come in three categories (see [portinari.palettes\(\)](#)): *qualitative* palettes for unordered groups, *sequential* palettes for ordered values, and *diverging* palettes for values around a meaningful midpoint. Discrete palettes pick n colors that suit the category: the most distinct colors for qualitative palettes, evenly spaced steps for sequential palettes, and balanced steps either side of the light center for diverging palettes. Continuous palettes interpolate the full palette in CIELAB space.

Value

A character vector of hex colors with class "portinari_palette".

See Also

[scale_color_portinari\(\)](#), [scale_fill_portinari\(\)](#), [portinari.palettes\(\)](#)

Examples

```
portinari.pal("cafe")
portinari.pal("retirantes", n = 3)
portinari.pal("guerra_azul", n = 20, type = "continuous")
portinari.pal("inah", colorblind = TRUE)
```

portinari.palettes	<i>List the available Portinari palettes</i>
--------------------	--

Description

List the available Portinari palettes

Usage

```
portinari.palettes(category = NULL)
```

Arguments

category Optional filter: one or more of "qualitative", "diverging" and "sequential".

Value

A data frame with one row per palette: its category, the number of colors, how many colors are colorblind-safe (cvd_n), whether the whole palette is colorblind-friendly, and a short description. See [portinari.works\(\)](#) for the paintings behind each palette.

Examples

```
portinari.palettes()
portinari.palettes("diverging")
```

portinari.works	<i>Paintings behind the Portinari palettes</i>
-----------------	--

Description

The works each palette was drawn from, with links to the collection pages. The sequential palettes guerra_azul and paz_ouro are derived from guerra and paz and share their works.

Usage

```
portinari.works(palette = NULL)
```

Arguments

palette Optional character vector of palette names to filter by.

Value

A data frame with columns `palette`, `title`, `year`, `collection`, `inventory`, `url` (collection or story page) and `image_url` (the museum's image, where one is published; NA otherwise).

Examples

```
portinari.works()
portinari.works("retirantes")
```

portinari_scales	<i>Portinari color and fill scales for ggplot2</i>
------------------	--

Description

Discrete or continuous color and fill scales built from a Portinari palette. They are ordinary ggplot2 scales and accept the usual arguments (name, labels, breaks, guide, ...). Match the palette category to the data: qualitative palettes for unordered groups, sequential palettes for ordered values, and diverging palettes for values around a midpoint (see [portinari.palettes\(\)](#)).

Usage

```
scale_color_portinari(
  palette = "cafe",
  type = c("discrete", "continuous"),
  direction = 1,
  colorblind = FALSE,
  ...
)

scale_colour_portinari(
  palette = "cafe",
  type = c("discrete", "continuous"),
  direction = 1,
  colorblind = FALSE,
  ...
)

scale_fill_portinari(
  palette = "cafe",
  type = c("discrete", "continuous"),
  direction = 1,
  colorblind = FALSE,
  ...
)
```

Arguments

palette	Name of the palette. See <code>portinari.palettes()</code> .
type	"discrete" (default) for categorical variables, or "continuous" for numeric variables.
direction	1 (default) or -1 to reverse the palette.
colorblind	If TRUE, only colorblind-safe colors are used.
...	Further arguments passed to <code>ggplot2::discrete_scale()</code> (discrete) or <code>ggplot2::scale_colour_gradientn()</code> / <code>ggplot2::scale_fill_gradientn()</code> (continuous).

Value

A ggplot2 scale object.

See Also

`portinari.pal()`

Examples

```
library(ggplot2)
# Qualitative: unordered groups
ggplot(mtcars, aes(wt, mpg, colour = factor(cyl))) +
  geom_point(size = 3) +
  scale_color_portinari("cafe", name = "Cylinders")

# Sequential: ordered values
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_raster() +
  scale_fill_portinari("guerra_azul", type = "continuous", direction = -1)

# Diverging: values around a midpoint
ggplot(mtcars, aes(wt, mpg, colour = mpg - mean(mpg))) +
  geom_point(size = 3) +
  scale_color_portinari("guerra", type = "continuous", name = "mpg vs mean")

# Diverging, discrete: an ordered scale with a neutral middle
ggplot(mtcars, aes(factor(gear), fill = factor(carb))) +
  geom_bar() +
  scale_fill_portinari("retirantes")
```

show_portinari

Show Portinari palettes

Description

Draws each palette as a row of color swatches using base graphics, labeled with its category.

Usage

```
show_portinari(palette = NULL, category = NULL)
```

Arguments

palette	Character vector of palette names. Defaults to all palettes (or all palettes in category).
category	Optional filter: one or more of "qualitative", "diverging" and "sequential".

Value

The palette names, invisibly.

Examples

```
show_portinari()
show_portinari(category = "diverging")
show_portinari(c("cafe", "inah"))
```

theme_portinari	<i>A ggplot2 theme in the spirit of Candido Portinari</i>
-----------------	---

Description

A complete ggplot2 theme that dresses a plot like one of Portinari's canvases: a warm ground of raw linen, firm drawn contours instead of a boxed panel, earth-colored lettering and an optional painted frame. Two further moods follow his United Nations panels: the deep indigo of *Guerra* and the golden light of *Paz*. Pair it with [scale_color_portinari\(\)](#) and [scale_fill_portinari\(\)](#).

Usage

```
theme_portinari(
  mood = c("canvas", "guerra", "paz"),
  base_size = 11,
  base_family = "",
  grid = c("y", "x", "xy", "none"),
  frame = TRUE
)
```

Arguments

mood	The overall look. "canvas" (default) is a raw-linen ground with umber ink and sienna contours; "guerra" is a dark indigo ground with pale citrine ink and light blue contours; "paz" is a golden cream ground with brown ink and deep ochre contours.
base_size	Base font size, in points.
base_family	Base font family.

grid	Which major grid lines to draw: "y" (default), "x", "xy" or "none".
frame	If TRUE (default), draws a thin painted frame around the whole plot, like the edge of a stretched canvas.

Value

A ggplot2 theme object.

Pairing palettes with a mood

The light moods share their ground with the lightest colors of several palettes (the creams of *cafe* and *paz*, the light centers of the diverging palettes, the pale ends of the sequential ones). That is natural for filled areas such as heatmaps, where the center then reads as neutral, but those colors fade when used for points or lines. The *guerra* mood uses the indigo that starts *guerra* and *guerra_azul*, so pair it with other palettes, for example *paz* or *sofia*: the ground of War with the colors of Peace.

See Also

[scale_color_portinari\(\)](#), [scale_fill_portinari\(\)](#)

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg, colour = factor(cyl))) +
  geom_point(size = 3) +
  scale_color_portinari("paz", name = "Cylinders") +
  labs(title = "Fuel economy", subtitle = "Heavier cars travel less per gallon") +
  theme_portinari()

ggplot(mpg, aes(class, fill = drv)) +
  geom_bar() +
  scale_fill_portinari("sofia") +
  theme_portinari("guerra")

ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_raster() +
  scale_fill_portinari("roxa_terra", type = "continuous", direction = -1) +
  theme_portinari("paz", grid = "none", frame = FALSE)
```

Index

`ggplot2::discrete_scale()`, 6
`ggplot2::scale_colour_gradientn()`, 6
`ggplot2::scale_fill_gradientn()`, 6

`portinari.cvd.check`, 2
`portinari.cvd.check()`, 3
`portinari.pal`, 3
`portinari.pal()`, 2, 6
`portinari.palettes`, 4
`portinari.palettes()`, 2, 3, 5, 6
`portinari.works`, 4
`portinari.works()`, 4
`portinari_scales`, 5

`scale_color_portinari`
 (`portinari_scales`), 5
`scale_color_portinari()`, 3, 7, 8
`scale_colour_portinari`
 (`portinari_scales`), 5
`scale_fill_portinari`
 (`portinari_scales`), 5
`scale_fill_portinari()`, 3, 7, 8
`show_portinari`, 6

`theme_portinari`, 7