

Package: cptcity (via r-universe)

October 9, 2026

Type Package

Title 'cpt-city' Colour Gradients

Version 2.4.2

Description Incorporates colour gradients from the 'cpt-city' web archive available at <https://phillips.shef.ac.uk/pub/cpt-city/>, including dedicated NOAA ramps for weather, ocean, and climate work. Provides functions to search, preview, and return colour palettes as either interpolated hex vectors or colorRampPalette functions for use with base graphics, ggplot2, and sf.

Depends R (>= 3.5.0)

Imports grDevices

License GPL-3

URL <https://github.com/ibarraespina/cptcity>

BugReports <https://github.com/ibarraespina/cptcity/issues/>

Encoding UTF-8

LazyData true

Suggests covr, sf, testthat

Date 2026-09-09

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-10-01 17:30:16 UTC

RemoteUrl <https://github.com/ibarraespina/cptcity>

RemoteRef HEAD

RemoteSha f51ffbcddd2ad1e3e03f71104a0ab872d7aa62b

Contents

cpt	2
cpt_names	3
cptcity	4
find_cpt	5
lucky	6
show_cpt	7
Index	9

cpt	<i>Return a colour palette function or vector from the cpt-city archive</i>
-----	---

Description

cpt extracts a named colour palette from the internal cpt-city database and returns either a character vector of n interpolated hex colours or the underlying `colorRampPalette` function itself. The package ships colour gradients from <https://phillips.shef.ac.uk/pub/cpt-city/>, including dedicated **NOAA** ramps ("noaa", "noaa_nws", "noaa_nhc", "noaa_nexrad", "noaa_goes", "noaa_buoy", "noaa_ncei", "noaa_nmfs", "noaa_jetstream", "noaa_tornado", "noaa_wind_chill", "noaa_heat_index", "noaa_coastal").

Usage

```
cpt(  
  pal = "mpl_inferno",  
  n = 100,  
  colorRampPalette = FALSE,  
  rev = FALSE,  
  frgb = rep(1, 3)  
)
```

Arguments

pal	Character. Name or position of the colour gradient in the cpt-city archive. Default "mpl_inferno".
n	Integer. Number of colours to return when colorRampPalette = FALSE. Ignored when colorRampPalette = TRUE.
colorRampPalette	Logical. If FALSE (default), returns a character vector of n hex colours. If TRUE, returns a colorRampPalette *function* – this is what you need for <code>plot</code> (pal =) and <code>scale_fill_gradientn</code> (colours =).
rev	Logical. If TRUE, reverses the order of colours before interpolation.
frgb	Numeric vector of length 3. Per-channel scaling factors for c(red, green, blue). Values > 1 boost the channel, < 1 reduce it. Default c(1, 1, 1) (no scaling).

Details

****‘sf’ / ‘ggplot2’ users:**** use `colorRampPalette = TRUE`. The `sf` plot method and **ggplot2**’s `scale_*_gradientn` need a *function*, not a character vector. See the examples below.

Value

A character vector of hex colours (length `n`) when `colorRampPalette = FALSE`, or a `colorRampPalette` function when `colorRampPalette = TRUE`.

Examples

```
library(cptcity)

# Character vector -- use with base graphics
image(matrix(1:100), col = cpt(pal = "mpl_inferno"))
image(matrix(1:100), col = cpt("idv_temperature", rev = TRUE))

# colorRampPalette function -- use with sf and ggplot2
## Not run:
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
plot(nc["AREA"], pal = cpt("mpl_inferno", colorRampPalette = TRUE))

library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density)) +
  scale_fill_gradientn(colours = cpt(n = 100, pal = "mpl_inferno"))

## End(Not run)

# Search for palettes by keyword
find_cpt("temperature")
image(matrix(1:100), col = cpt("idv_temperature"))

# Tint the palette by scaling the red channel down
image(matrix(1:100), col = cpt("mpl_inferno", frgb = c(0.3, 1, 1)))
```

cpt_names

Names of all colour gradients

Description

A character vector containing the names of every colour gradient bundled in the **cptcity** package. Use `find_cpt` to search this vector by keyword.

Usage

```
data(cpt_names)
```

Format

A character vector.

Source

<https://phillips.shef.ac.uk/pub/cpt-city/>

Examples

```
data(cpt_names)
head(cpt_names)
length(cpt_names)
```

cptcity

cpt-city: Colour Gradients for R

Description

Incorporates colour gradients from the [cpt-city web archive](#) into R, including dedicated **NOAA** ramps. Each palette is stored as a `data.frame` with columns `id`, `r`, `g`, `b`.

Functions

`cpt` Return an interpolated colour palette or `colorRampPalette` function.

`find_cpt` Search palette names by keyword.

`lucky` Pick a random palette – "I'm Feeling Lucky" for colours.

`show_cpt` Display multiple palettes side-by-side.

Author(s)

Maintainer: Sergio Ibarra-Espinosa <sergio.ibarra-espinoza@noaa.gov> ([ORCID](#))

Authors:

- Sergio Ibarra-Espinosa <sergio.ibarra-espinoza@noaa.gov> ([ORCID](#))

See Also

Useful links:

- <https://github.com/ibarraespinoza/cptcity>
- Report bugs at <https://github.com/ibarraespinoza/cptcity/issues/>

find_cpt	<i>Find colour palette names by keyword search</i>
----------	--

Description

Searches the cpt-city palette names and returns those that match your search term. Case-insensitive by default.

Usage

```
find_cpt(name, ignore.case = TRUE, fixed = FALSE)
```

Arguments

name	Character. Keyword to search for within palette names.
ignore.case	Logical. If TRUE (default), search is case-insensitive. Ignored when fixed = TRUE.
fixed	Logical. If FALSE (default), name is treated as a regular expression. Set to TRUE for literal string matching (faster, but case-sensitive).

Value

A character vector of palette names that contain the keyword. Returns a zero-length vector if nothing matches.

Note

This is a thin wrapper around [grep](#). You can pass regex patterns like "^oc_" to find all palettes starting with "oc_" or "temperature|thermal" to match either term.

Examples

```
library(cptcity)

# Case-insensitive search (default)
find_cpt("temperature")

# Case-sensitive
find_cpt("Temperature", ignore.case = FALSE)

# Regex: all palettes starting with "oc_"
find_cpt("^oc_")

# Literal match (faster, exact substring -- case-sensitive)
find_cpt("rain", fixed = TRUE)

# Use the result directly
image(matrix(1:100), col = cpt("idv_temperature"))
```

```
## Not run:
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions)) +
  geom_raster(aes(fill = density)) +
  scale_fill_gradientn(colours = cpt(pal = "ncl_radar", n = 100))

## End(Not run)
```

lucky

Return a random colour gradient – "I'm Feeling Lucky" for palettes

Description

Picks one of the colour gradients at random and returns it, optionally printing its name so you can re-use it later. Based on Google's "I'm Feeling Lucky" button.

Usage

```
lucky(
  n = 100,
  colorRampPalette = FALSE,
  rev = FALSE,
  message = TRUE,
  nseed,
  frgb = rep(1, 3)
)
```

Arguments

n	Integer. Number of colours to return when colorRampPalette = FALSE. Ignored when colorRampPalette = TRUE.
colorRampPalette	Logical. If FALSE (default), returns a character vector of n colours. If TRUE, returns a colorRampPalette function – the right choice for plot and ggplot2 .
rev	Logical. If TRUE, reverse the gradient order.
message	Logical. If TRUE (default), prints the name and index of the chosen palette to the console.
nseed	Integer. A seed value passed to set.seed for reproducible random selection.
frgb	Numeric vector of length 3. Per-channel scaling factors for c(red, green, blue). Default c(1, 1, 1).

Value

A character vector of hex colours when colorRampPalette = FALSE, or a colorRampPalette function when colorRampPalette = TRUE.

Examples

```
library(cptcity)

# Random palette each time you call it
image(matrix(1:100), col = lucky())
image(matrix(1:100), col = lucky())
image(matrix(1:100), col = lucky(rev = TRUE))

# Reproducible random pick
image(matrix(1:100), col = lucky(nseed = 42))
image(matrix(1:100), col = lucky(nseed = 42)) # same palette

# Silent mode + sf-compatible output
## Not run:
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
plot(nc["AREA"], pal = lucky(colorRampPalette = TRUE, message = FALSE))

## End(Not run)
```

show_cpt

*Visualise one or more colour palettes side-by-side***Description**

Draws each palette as a horizontal colour bar using [image](#), arranged in a grid that automatically adapts to the number of palettes.

Usage

```
show_cpt(x, label = TRUE)
```

Arguments

x Character vector. Palette names (e.g. output from [find_cpt](#)).

label Logical. If TRUE (default), overlay a numbered label on each colour bar.

Value

Called for its side effect (a plot). Invisibly returns the palette names.

Examples

```
library(cptcity)

# Search then visualise
pals <- find_cpt("radar")
show_cpt(pals)
```

```
# Without labels  
show_cpt(pals, label = FALSE)
```

```
# Single palette  
show_cpt("mpl_inferno")
```

Index

* **datasets**

cpt_names, [3](#)

colorRampPalette, [2](#)

cpt, [2](#), [4](#)

cpt_names, [3](#)

cptcity, [4](#)

cptcity-package (cptcity), [4](#)

find_cpt, [3](#), [4](#), [5](#), [7](#)

grep, [5](#)

image, [7](#)

lucky, [4](#), [6](#)

plot, [2](#), [6](#)

scale_fill_gradientn, [2](#)

set.seed, [6](#)

show_cpt, [4](#), [7](#)