Package ‘colors3d’

October 31, 2023

Title Generate 2D and 3D Color Palettes

Version 1.0.1

Description Generate multivariate color palettes to represent two-dimensional or three-dimensional data in graphics (in contrast to standard color palettes that represent just one variable). You tell 'colors3d' how to map color space onto your data, and it gives you a color for each data point. You can then use these colors to make plots in base 'R', 'ggplot2', or other graphics frameworks.

Depends R (>= 3.2.2)

License MIT + file LICENSE

URL https://github.com/matthewkling/colors3d

BugReports https://github.com/matthewkling/colors3d/issues

RoxygenNote 7.2.3

Encoding UTF-8

Imports combinat, FNN, grDevices, plyr, scales, stats

NeedsCompilation no

Author Matthew Kling [aut, cre, cph]

Maintainer Matthew Kling <mattkling@berkeley.edu>

Repository CRAN

Date/Publication 2023-10-31 18:20:07 UTC

R topics documented:

colors2d	*Map values to a 2D legend interpolated from 4 corner colors.*

Description

This function returns a color value for each row of the 2-column dataset supplied, based on a 2D color palette interpolated from 4 corner colors.

Usage

```
colors2d(
  data,
  colors = c("yellow", "green", "blue", "magenta"),
  xtrans = c("none", "log", "rank"),
  ytrans = c("none", "log", "rank")
)
```

Arguments

`data`	Matrix or data frame with 2 numeric columns; they will map to x and y.
`colors`	Vector of 4 corner colors to interpolate, clockwise from upper right.
`xtrans, ytrans`	Transformation to apply to x and y variables before applying a linear color mapping: either "none" (default), "log", or "rank".

Value

Character vector of colors.

Examples

```
plot(iris,
     col = colors2d(iris[, c("Sepal.Length", "Sepal.Width")]),
     pch = 19, cex = 2)

plot(iris,
     col = colors2d(iris[, c("Sepal.Length", "Sepal.Width")],
                    colors = c("limegreen", "gold", "black", "dodgerblue"),
                    xtrans = "rank", ytrans = "rank"),
     pch = 19, cex = 2)
```

colors3d *Map values to a 3D legend in RGB colorspace.*

Description

This function returns a color value for each row of the 3-column dataset supplied, by transforming the input data and using it as RGB values.

Usage

```
colors3d(data, trans = "none", order = 1, inversion = 1, opacity = NULL)
```

Arguments

`data`	Matrix or data frame with 3 numeric columns.
`trans`	Either "none" (defaut, histogram is rescaled) or "rank" (histogram is flattened).
`order`	Integer from 1 to 6, each denoting a unique permutation of variables-to-color band mapping. Under the default value of 1, the three variables in ‘data‘ are respectively mapped onto the R, G, and B bands of colorspace.
`inversion`	Integer from 1 to 8, each denoting a unique combination of variables to reverse before mapping. Under the default value of 1, all three variables are mapped with positive values at the high end of the color band. Together with the ‘order‘ parameter, this allows all possible 48 unique mappings of a given set of variables onto 3D colorspace.
`opacity`	Not currently used.

Value

Character vector of colors.

Examples

```
d <- expand.grid(x = 1:49, y = 1:49)
d$z <- cos(sqrt((d$x-25)^2 + (d$y-25)^2))
plot(d[, 1:2], col = colors3d(d), pch = 15, cex = 2)

plot(d[, 1:2], col = colors3d(d, order = 2, inversion = 2), pch = 15, cex = 2)
```

colorwheel2d *Map values to a 2D colorwheel legend.*

Description

This function returns a color value for each row of the 2-column dataset supplied, based on a 2D color palette defined by a center color and a series of peripheral colors.

Usage

```
colorwheel2d(
  data,
  colors = c("black", "yellow", "green", "cyan", "blue", "magenta", "red"),
  origin = NULL,
  xyratio = NULL,
  kernel = NULL
)
```

Arguments

`data`	Matrix or data frame with 2 numeric columns; they will map to x and y.
`colors`	Vector of colors to interpolate: center followed by periphery counterclockwise from 3 o'clock.
`origin`	Coordindates of color wheel center.
`xyratio`	Scalar representing how to map the elliptical color wheel in the data space (the default 1 a circular mapping that weights the two dimensions equally).
`kernel`	Optional function describing the shape of radial color gradients (default is a linear mapping corresponding to a triangular kernel); this function should take a vector of distances to the center as its sole input and return a positive number.

Value

Character vector of colors.

Examples

```
plot(iris,
      col = colorwheel2d(iris[, c("Sepal.Length", "Sepal.Width")]),
      pch = 19, cex = 2)

plot(iris,
      col = colorwheel2d(
                  iris[, c("Sepal.Length", "Sepal.Width")],
                  origin = c(5.5, 2.5),
                  kernel = function(x) x ^ .5),
      pch = 19, cex = 2)
```

`distant_colors` *Palettes of dissimilar colors in RGB space.*

Description

Many standard palette generators use only a slice of color space, which can cause a lack of differentiability in palettes used to visualize categorical factors with many levels. This function attempts to overcome this by generating colors using nearest-neighbor distance maximization in 3D RGB space.

Usage

```
distant_colors(
  n,
  res = 20,
  maxreps = 1000,
  radius = 10,
  avoid_white = TRUE,
  seed = NULL
)
```

Arguments

`n`	Number of colors (integer).
`res`	Number of distinct values in each RGB dimension (integer).
`maxreps`	Max number of optimization iterations (integer).
`radius`	Neighborhood size for potential moves, analagous to heating.
`avoid_white`	Logical, default is TRUE.
`seed`	Integer used to seed randomization during search; leave as NULL to generate different results each time, or set a value to generate reproducible results.

Value

Character vector of colors.

Examples

```
plot(runif(20), runif(20),
     col = distant_colors(20),
     pch = 16, cex = 3)
```

`polarize` *Internal function converting x-y do distance-angle.*

Description

Internal function converting x-y do distance-angle.

Usage

```
polarize(data, xyratio, xorigin = 0, yorigin = 0)
```

Arguments

`data`	Matrix or data frame with 2 numeric columns representing x and y.
`xyratio`	Single number indicating unit ratio in x vs y direction.
`xorigin, yorigin`	Numbers indicating center of polarization.

Value

2-column matrix of distances and angles.

Index