Package 'rasterImage'

July 23, 2025

Type Package

Title An Improved Wrapper of image()

Version 0.4.0

Author Martin Seilmayer

Maintainer Martin Seilmayer <m.seilmayer@hzdr.de>

Description This is a wrapper function for image(), which makes reasonable raster plots with nice axis and other useful features.

License GPL-2

Depends plotrix,grDevices,R (>= 2.15.0)

LazyData TRUE

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-14 08:40:07 UTC

Contents

`colorPalette` *Defines a color palette*

Description

This function defines a color palette and returns a vector of colors. The palettes itself are adapted from the ColorBrewer project.

Usage

```
colorPalette(n = NULL, type = "spectral", inv = F)
```

Arguments

n	number of colors to produce
type	sets the type of color palette. See Details
inv	revert the order of colors

Details

The parameter type controls the output palette type as follows:

"spectral" spectral colors from blue to red

"specrtalHalf" spectral colors from green to red

"green" MultiHue yellow - green

"blue" MultiHue yellow - green blue

"orange" MultiHue yellow - orange - brown

"red" MultiHue yellow - orange red

"red-white-bule","bwr" red - white - blue colors

"rainbow" reproduces the rainbow color set

"black-white","bw" gray scale colors

"white-black","wb" gray scale colors from white to black

"jet.colors","jc" dark blue to dark red

"hzdr1" HZDR cooperate design colors

"hzdr2" HZDR cooperate design colors

If a vector of color names is supported, then a customized palette will be calculated according to these colors.

Value

returns a vector of colors to be passed to image or rasterImage

References

http://colorbrewer2.org by Cynthia A. Brewer, Geography, Pennsylvania State University

Examples

```
# default "spectral" palette
barplot(rep(1,10), col = colorPalette(10))

# custom color palette
barplot(rep(1,10), col = colorPalette(n = 10, type = c("red","blue","yellow")))
```

rasterImage2 *Plotting a raster image with axis and color legend*

Description

The function is a wrapper for the `image()` function, but with a comfortable control of the z-axis and its color legend. The wrapper also supports image resizing (resolution) and png output for better export.

Usage

```
rasterImage2(x = NULL, y = NULL, z, zlim = NULL, xlim = NULL,
  ylim = NULL, dim.max = NULL, plot.zero.line = T, regularGrid = T,
  zlab = NULL, z.cex = 0.5, z.adj = c(0.5, 0.5), z.format = "fg",
  ndz = 7, ncolors = 256, palette = "spectral", palette.inv = F,
  ...)
```

Arguments

`x`	x-axis vector corresponding to the z-matrix
`y`	y-axis vector corresponding to the z-matrix
`z`	numeric matrix to be plotted
`zlim`	sets the range of the color coded z-axis
`xlim`	the x limits (x1, x2) of the plot. Note that `x1 > x2` is allowed and leads to a 'reversed axis'. The default value, `NULL`, indicates that the range of the finite values to be plotted should be used.
`ylim`	the y limits of the plot.
`dim.max`	defines the dimensions of the visible area of z. It automatically invokes a rescale. In case of large data sets this parameter can improve plotting speed.
`plot.zero.line`	logical, if a line at `x = 0` and `y = 0` is to be plotted.
`regularGrid`	logical, if `FALSE` then a vector plot is generated, which is the slow and standard behaviour of `image`. If this parameter is `TRUE` then a raster image is generated, which can be processed much faster, compared to the `FALSE` option.
`zlab`	defines the z-label
`z.cex`	cex value for the z-label. It sets the font size in relation to the global `par()$cex` value
`z.adj`	a two component vector. It sets the left/right and top/bottom justification
`z.format`	controls how the numbers besides the color scale are composed. It works like the `format` option of `formatC`
`ndz`	sets the axis breaks right to the color scale
`ncolors`	number of colors to use in the plot
`palette`	defines the color palette to be used in the plot
`palette.inv`	logical, if `TRUE` reverts the color palette
`...`	further arguments to the plot function, e.g. `'xlab'`

Details

The `regularGrid` option forces interpolation in case of irregular spacing of `x` or `y`. All data is then projected on a regular grid. This correction invokes a spline interpolation. Missing `NA` values are ignored.

Examples

```
rasterImage2( z = volcano, palette = "spectral", dim.max = c(500,100)
            ,zlab = "Height", z.adj = c(0,1) ,z.cex = 1
            ,main = "Volcano Data Set"
          )
```

Index