Package: PEcAn.visualization (via r-universe)

June 27, 2024

Type Package

Title PEcAn visualization functions

Version 1.7.2

Date 2021-10-04

Description The Predictive Ecosystem Carbon Analyzer (PEcAn) is a scientific workflow management tool that is designed to simplify the management of model parameterization, execution, and analysis. The goal of PECAn is to streamline the interaction between data and models, and to improve the efficacy of scientific investigation. This module is used to create more complex visualizations from the data generated by PEcAn code, specifically the models.

Imports data.table, ggplot2, ncdf4 (>= 1.15), PEcAn.logger, plyr (>= 1.8.4), reshape2, rlang, stringr(>= 1.1.0)

Suggests grid, mockery, png, raster, sp, testthat (>= 1.0.2), withr

License BSD_3_clause + file LICENSE

Copyright Authors

LazyLoad yes

LazyData FALSE

Encoding UTF-8

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

Repository https://pecanproject.r-universe.dev

RemoteUrl https://github.com/PecanProject/pecan

RemoteRef HEAD

RemoteSha d5c7bffdf233077968945a182c11240b5d76e42d

Contents

`add_icon` *add_icon*

Description

add_icon

Usage

```
add_icon(id = NULL, x = 0, y = 0)
```

Arguments

`id`	additional plot identification (URL, database ID, etc)
`x`	x-coordinate of logo
`y`	y-coordinate of logo

Author(s)

Mike Dietze

ciEnvelope *plots a confidence interval around an x-y plot (e.g. a timeseries)*

Description

plots a confidence interval around an x-y plot (e.g. a timeseries)

Usage

```
ciEnvelope(x, ylo, yhi, ...)
```

Arguments

x	Vector defining CI center
ylo	Vector defining bottom of CI envelope
yhi	Vector defining top of CI envelope
...	Further arguments passed on to graphics::polygon

Author(s)

Michael Dietze, David LeBauer

data.fetch *data.fetch*

Description

data.fetch

Usage

```
data.fetch(var, nc, fun = mean)
```

Arguments

var	the variable to extract from the hdf data
nc	ncdf file path
fun	the function to apply to the data at the same time, DEFAULT fun = mean

Value

aggregated data

dhist *Variable-width (diagonally cut) histogram*

Description

When constructing a histogram, it is common to make all bars the same width. One could also choose to make them all have the same area. These two options have complementary strengths and weaknesses; the equal-width histogram oversmooths in regions of high density, and is poor at identifying sharp peaks; the equal-area histogram oversmooths in regions of low density, and so does not identify outliers. We describe a compromise approach which avoids both of these defects. We regard the histogram as an exploratory device, rather than as an estimate of a density.

Usage

```
dhist(
  x,
  a = 5 * iqr(x),
  nbins = grDevices::nclass.Sturges(x),
  rx = range(x, na.rm = TRUE),
  eps = 0.15,
  xlab = "x",
  plot = TRUE,
  lab.spikes = TRUE
)
```

Arguments

x	is a numeric vector (the data)
a	is the scaling factor, default is 5 * IQR
nbins	is the number of bins, default is assigned by the Stuges method
rx	is the range used for the left of the left-most bin to the right of the right-most bin
eps	used to set artificial bound on min width / max height of bins as described in Denby and Mallows (2009) on page 24
xlab	is label for the x axis
plot	= TRUE produces the plot, FALSE returns the heights, breaks and counts
lab.spikes	= TRUE labels the % of data in the spikes

Value

list with two elements, heights of length n and breaks of length n+1 indicating the heights and break points of the histogram bars.

Author(s)

Lorraine Denby, Colin Mallows

References

Lorraine Denby, Colin Mallows. Journal of Computational and Graphical Statistics. March 1, 2009, 18(1): 21-31. doi:10.1198/jcgs.2009.0002.

`iqr` *Interquartile range*

Description

Calculate interquartile range

Usage

```
iqr(x)
```

Arguments

`x`	vector

Details

Calculates the 25th and 75th quantiles given a vector x; used in function dhist.

Value

numeric vector of length 2, with the 25th and 75th quantiles of input vector x

---

`map.output` *Map Output*

Description

Map Output

Usage

```
map.output(table, variable)
```

Arguments

`table`	data.table or data.frame with columns lat, lon, followed by variable names
`variable`	name of variable to be mapped

Value

plot

Author(s)

David LeBauer

`plot_data` *Add data to plot*

Description

Add data to an existing plot or create a new one

Usage

```
plot_data(trait.data, base.plot = NULL, ymax)
```

Arguments

`trait.data`	Data to be plotted
`base.plot`	a ggplot object (grob), created if none provided
`ymax`	maximum height of y

Details

Used to add raw data or summary statistics to the plot of a distribution. The height of Y is arbitrary, and can be set to optimize visualization. If SE estimates are available, the SE will be plotted

Value

updated plot object

Author(s)

David LeBauer

Examples

```
## Not run: plot_data(data.frame(Y = c(1, 2), se = c(1,2)), base.plot = NULL, ymax = 10)
```

plot_netcdf *Load the tower dataset and create a plot.*

Description

Loads the tower data from an HDF5 file generated by ED and will plot the values against one another. The default is for the given variable to be plotted against time.

Usage

```
plot_netcdf(
  datafile,
  yvar,
  xvar = "time",
  width = 800,
  height = 600,
  filename = NULL,
  year = NULL
)
```

Arguments

datafile	the specific datafile to use.
yvar	the variable to plot along the y-axis.
xvar	the variable to plot along the x-axis, by default time is used.
width	the width of the image generated, default is 800 pixels.
height	the height of the image generated, default is 600 pixels.
filename	is the name of the file name that is generated, this can be null to use existing device, otherwise it will try and create an image based on filename, or display if x11.
year	the year this data is for (only used in the title).

Author(s)

Rob Kooper

theme_border *Theme border for plot*

Description

Add borders to plot

Usage

```
theme_border(
  type = c("left", "right", "bottom", "top", "none"),
  colour = "black",
  size = 1,
  linetype = 1
)
```

Arguments

type	border(s) to display
colour	what colo(u)r should the border be
size	relative line thickness
linetype	"solid", "dashed", etc.

Details

Has ggplot2 display only specified borders, e.g. ('L'-shaped) borders, rather than a rectangle or no border. Note that the order can be significant; for example, if you specify the L border option and then a theme, the theme settings will override the border option, so you need to specify the theme (if any) before the border option, as above.

Value

adds borders to ggplot as a side effect

Author(s)

Rudolf Cardinal

ggplot2googlegrouphttps://groups.google.com/forum/?fromgroups#!topic/ggplot2/-ZjRE2OL8lE

Examples

```
## Not run:
df = data.frame( x=c(1,2,3), y=c(4,5,6) )
ggplot(data=df, aes(x=x, y=y)) + geom_point() + theme_bw() +
       opts(panel.border = theme_border(c('bottom','left')) )
ggplot(data=df, aes(x=x, y=y)) + geom_point() + theme_bw() +
       opts(panel.border = theme_border(c('b','l')) )
```

```
## End(Not run)
```

---

vwReg *PEcAn worldmap*

Description

Visually weighted regression / Watercolor plots

Usage

```
vwReg(
  formula,
  data,
  title = "",
  B = 1000,
  shade = TRUE,
  shade.alpha = 0.1,
  spag = FALSE,
  spag.color = "darkblue",
  mweight = TRUE,
  show.lm = FALSE,
  show.median = TRUE,
  median.col = "white",
  shape = 21,
  show.CI = FALSE,
  method = stats::loess,
  bw = FALSE,
  slices = 200,
 palette = (grDevices::colorRampPalette(c("#FFEDA0", "#DD0000"), bias = 2))(20),
  ylim = NULL,
  quantize = "continuous",
  add = FALSE,
  ...
)
```

Arguments

formula	variables to plot. See examples
data	data frame containing all variables used in formula
title	passed on to ggplot
B	= number bootstrapped smoothers
shade	plot the shaded confidence region?
shade.alpha	should the CI shading fade out at the edges? (by reducing alpha; 0 = no alpha decrease, 0.1 = medium alpha decrease, 0.5 = strong alpha decrease)

spag	plot spaghetti lines?
spag.color	color of spaghetti lines
mweight	should the median smoother be visually weighted?
show.lm	should the linear regression line be plotted?
show.median	should the median smoother be plotted?
median.col	color of the median smoother
shape	shape of points
show.CI	should the 95% CI limits be plotted?
method	the fitting function for the spaghettis; default: loess
bw	= TRUE: define a default b&w-palette
slices	number of slices in x and y direction for the shaded region. Higher numbers make a smoother plot, but takes longer to draw. I wouldn'T go beyond 500
palette	provide a custom color palette for the watercolors
ylim	restrict range of the watercoloring
quantize	either 'continuous', or 'SD'. In the latter case, we get three color regions for 1, 2, and 3 SD (an idea of John Mashey)
add	if add == FALSE, a new ggplot is returned. If add == TRUE, only the elements are returned, which can be added to an existing ggplot (with the '+' operator)
...	further parameters passed to the fitting function, in the case of loess, for example, 'span = .9', or 'family = 'symmetric"

Details

Idea: Solomon Hsiang, with additional ideas from many blog commenters Details: http://www.nicebread.de/visually-weighted-regression-in-r-a-la-solomon-hsiang/ http://www.nicebread.de/visually-weighted-watercolor-plots-new-variants-please-vote/

Value

NULL plot as side effect

Author(s)

Felix Schönbrodt

Examples

```
# build a demo data set
set.seed(1)
x <- rnorm(200, 0.8, 1.2)
e <- rnorm(200, 0, 3)*(abs(x)^1.5 + .5) + rnorm(200, 0, 4)
e2 <- rnorm(200, 0, 5)*(abs(x)^1.5 + .8) + rnorm(200, 0, 5)
y <- 8*x - x^3 + e
y2 <- 20 + 3*x + 0.6*x^3 + e2
df <- data.frame(x, y, y2)
```

```
p1 <- vwReg(y~x, df, spag=TRUE, shade=FALSE)
p2 <- vwReg(y2~x, df, add=TRUE, spag=TRUE, shade=FALSE, spag.color='red', shape=3)
p3 <- p1 + p2
p3

y <- x + x^2 + runif(200, 0, 0.4)
vwReg(y ~ x, df, method=lm)
vwReg(y ~ x + I(x^2), df, method=lm)
```

Index