Package 'inlmisc'

October 25, 2016

Title Miscellaneous Functions for the USGS INL Project Office

Version 0.2.2

Description A collection of functions for creating high-level graphics, performing raster-based analysis, processing MODFLOW-based models, and overlaying multi-polygon objects. Used to support packages and scripts written by researchers at the United States Geological Survey (USGS) Idaho National Laboratory (INL) Project Office.

Depends R (>= 3.2.0)

Imports dplyr, grDevices, graphics, igraph, knitr, methods, sp, stats, raster, rgdal, rgeos

Suggests colorspace, maptools, testthat

License CC0

Copyright This software is in the public domain because it contains materials that originally came from the USGS, an agency of the United States Department of Interior. For more information, see the official USGS copyright policy at https://www2.usgs.gov/visual-id/credit_usgs.html

URL https://github.com/USGS-R/inlmisc

BugReports https://github.com/USGS-R/inlmisc/issues

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Author Jason Fisher [aut, cre]

Maintainer Jason Fisher <jfisher@usgs.gov>

Repository CRAN

Date/Publication 2016-10-25 00:20:15

R topics documented:

`AddBubbles` *Add Bubble Map to Plot*

Description

This function can be used to add a bubble map to a plot. Proportional circle symbols are used to represent spatial point data, where symbol area varies in proportion to an attribute variable.

Usage

```
AddBubbles(x, y = NULL, z, zlim = NULL, inches = c(0, 0.2),
  scaling = c("perceptual", "mathematical"), bg.pos = "red",
  bg.neg = "blue", fg = NA, lwd = 0.25, cex = 0.7, format = NULL,
  draw.legend = TRUE, loc = c("bottomleft", "topleft", "topright",
  "bottomright"), inset = 0.02, breaks = NULL, break.labels = NULL,
  quantile.breaks = FALSE, make.intervals = FALSE, title = NULL,
  subtitle = NULL, add = TRUE)
```

Arguments

`x, y`	numeric. The x and y coordinates for the centers of the circle symbols. They can be specified in any way which is accepted by `xy.coords`.
`z`	numeric. Attribute variable

zlim	numeric. Minimum and maximum z values that circle symbols are plotted; defaults to the range of the finite values of z.
inches	numeric. Vector of length 2 specifying the radii limits for the drawn circle symbol.
scaling	character. Selects the proportional symbol mapping algorithm to be used; either "perceptual" or "mathematical" scaling (Tanimura and others, 2006).
bg.pos	character or function. Fill color(s) for circle symbols corresponding to positive z values. A color palette also may be specified.
bg.neg	character or function. Fill color(s) for circle symbols corresponding to negative z values. A color palette also may be specified.
fg	character. Outer-line color for circle symbols. Specify a value of NA to remove the symbols outer line, and NULL to match the outer-line color with the symbols fill color.
lwd	numeric. Line width for drawing circle symbols
cex	character. Character expansion factor for legend labels
format	character. Formatting for legend values, see formatC for options.
draw.legend	logical. If true, a legend is drawn.
loc	character. Position of the legend in the main plot region: "bottomleft", "topleft", "topright", or "bottomright" to denote scale location.
inset	numeric. Inset distance of the legend from the margins as a fraction of the main plot region. Defaults to 2 percent of the axis range.
breaks	numeric. Set of finite breakpoints for the legend circle symbols.
break.labels	character. Vector of break labels with length equal to breaks.
quantile.breaks	logical. If true, breaks are set to the sample quantiles of z.
make.intervals	logical. If true, represent z within intervals. See findInterval function for details.
title	character. Main title to be placed at the top of the legend.
subtitle	character. Legend subtitle to be placed below the main title.
add	logical. If true, circle symbols (and an optional legend) are added to an existing plot.

Value

Used for the side-effect of a bubble map drawn on the current graphics device.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

References

Tanimura, S., Kuroiwa, C., and Mizota, T., 2006, Proportional Symbol Mapping in R: Journal of Statistical Software, v. 15, no. 5, 7 p.

See Also

symbols

Examples

```
n <- 50L
x <- cbind(runif(n, 1, 10), runif(n, 1, 10))
z <- runif(n, -5000, 10000)
AddBubbles(x, z = z, fg = "green", lwd = 2, title = "Title", loc = "topright",
           breaks=pretty(z, n = 8), add = FALSE)

Pal1 <- colorRampPalette(c("#F4A582", "#CA0020"))
Pal2 <- colorRampPalette(c("#92C5DE", "#0571B0"))
AddBubbles(x, z = z, bg.pos = Pal1, bg.neg = Pal2, add = FALSE)

AddBubbles(x, z = z, bg.pos = Pal1, bg.neg = Pal2, add = FALSE,
           make.intervals = TRUE)

AddBubbles(x, z = runif(n, 10, 10000), title = "Quantiles", bg.pos = topo.colors,
           quantile.breaks = TRUE, fg = NULL, add = FALSE)
```

`AddColorKey` *Add Color Key to Plot*

Description

This function can be used to add a color key to a plot.

Usage

```
AddColorKey(mai, is.categorical, breaks, col, at = NULL, labels = TRUE,
  scientific = FALSE, explanation = NULL, padx = 0.2)
```

Arguments

`mai`	numeric. Vector of the form `c(bottom, left, top, right)` which gives the margin size specified in inches (optional).
`is.categorical`	logical. If true, color-key values represent categorical data; otherwise, these data values are assumed continuous.
`breaks`	numeric. Set of finite numeric breakpoints for the colors: must have one more breakpoint than color and be in increasing order.
`col`	character. Vector of colors to be used in the plot. This argument requires `breaks` specification for continuous data. For continuous data there should be one less color than breaks; whereas, categorical data require a color for each category.
`at`	numeric. The points at which tick-marks and labels are to be drawn, only applicable for continuous data. The tick-marks will be located at the color breaks if the length of `at` is greater than or equal to one minus the length of `breaks`.

labels	logical or character. Can either be a logical value specifying whether (numerical) annotations are to be made at the tickmarks, or a character or expression vector of labels to be placed at the tickpoints.
scientific	logical. Indicates if axes labels should be formatted for scientific notation, see ToScientific for details.
explanation	character. Label that describes the data values.
padx	numeric. Inner padding for the left and right margins specified in inches.

Value

Used for the side-effect of a color key drawn on the current graphics device.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

PlotCrossSection, PlotMap

Examples

```
dev.new(width = 7, height = 2)

AddColorKey(is.categorical = FALSE, breaks = 0:10, scientific = TRUE,
            explanation = "Example description for data variables in meters.")
AddColorKey(is.categorical = FALSE, breaks = 0:10, at = pretty(0:10))
AddColorKey(is.categorical = FALSE, breaks = seq(0.5, 10.5, by = 1), at = 1:10)

AddColorKey(is.categorical = TRUE, labels = LETTERS[1:5])
AddColorKey(is.categorical = TRUE, col = terrain.colors(5))
```

AddInsetMap *Add Inset Map to Plot*

Description

This function can be used to add an inset map to a plot.

Usage

```
AddInsetMap(p, col = c("#D8D8D8", "#BFA76F"), main.label = list(label = NA,
  adj = NULL), sub.label = list(label = NA, adj = NULL),
  loc = c("bottomleft", "topleft", "topright", "bottomright"), inset = 0.02,
  width = NULL)
```

Arguments

p	SpatialPolygons. Polygon describing the large map.
col	list. Vector of length 2 giving the colors for filling the large map polygon p and the smaller plot extent rectangle.
main.label	list. List with components label and adj. The text label and position (x and y adjustment of the label) for the large map, respectively.
sub.label	list. Identical to the main.label argument but for the plot extent rectangle.
loc	character. Position of the inset map in the main plot region: "bottomleft", "topleft", "topright", or "bottomright" to denote scale location.
inset	numeric. Inset distance from the margins as a fraction of the main plot region. Defaults to 2 percent of the axis range.
width	numeric. Width of the inset map in inches.

Details

The smaller axis-aligned rectangle (relative to the larger map polygon) is defined by the user coordinate extent of the main plot region, see par("usr").

Value

Used for the side-effect of a inset map drawn on the current graphics device.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

PlotMap

Examples

```
nc <- maptools::readShapePoly(system.file("shapes/sids.shp", package = "maptools")[1],
                              proj4string=sp::CRS("+proj=longlat +datum=NAD27"))
bb <- sp::bbox(nc[100, ])
xlim <- grDevices::extendrange(bb["x", ])
ylim <- grDevices::extendrange(bb["y", ])
PlotMap(raster::crs(nc), xlim = xlim, ylim = ylim, dms.tick = TRUE)
sp::plot(nc, add = TRUE)
AddInsetMap(nc, width = 3, main.label = list("North Carolina", adj = c(1.8, 3)),
            sub.label = list("Map area", adj = c(1.5, 0.5)), loc = "topright")
```

AddScaleBar *Add Scale Bar to Plot*

Description

This function can be used to add a scale bar to a plot.

Usage

```
AddScaleBar(asp = 1, unit = NULL, is.lonlat = FALSE,
  loc = c("bottomleft", "topleft", "topright", "bottomright"), offset = c(0,
  0), lab.vert.exag = NULL)
```

Arguments

asp	numeric. The *y/x* aspect ratio for spatial axes.
unit	character. Axis unit of measurement, for example "METERS".
is.lonlat	logical. If true, plot coordinates are in longitude and latitude.
loc	character. Position of the scale bar in the plot region: "bottomleft", "topleft", "topright", or "bottomright" to denote scale location.
offset	numeric. The x and y adjustments of the scale bar, in inches.
lab.vert.exag	logical. If true, a label is drawn specifying the vertical exaggeration.

Value

Used for the side-effect of a scale bar drawn on the current graphics device.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

PlotCrossSection, PlotMap

Examples

```
plot(-100:100, -100:100, type = "n", xlab = "x", ylab = "y", asp = 2)
AddScaleBar(2, unit = "FEET", loc = "topleft")
AddScaleBar(2, unit = "METERS", loc = "bottomright", offset = c(-0.2, 0))
```

`BumpDisconnectCells` *Adjustment for Vertically Disconnected Cells*

Description

This function decreases model cell values (such as land-surface elevations) in the lower raster layer if they violate a minimum vertical overlap between adjacent cells.

Usage

```
BumpDisconnectCells(rs, min.overlap = 2, bump.by = 0.1, max.itr = 10000)
```

Arguments

`rs`	RasterStack. A collection of two raster layers, the first and second layers represent the top and bottom of a model layer.
`min.overlap`	numeric. Minimum vertical overlap between adjacent cells.
`bump.by`	numeric. Amount to decrease a cell value by during each iteration of the algorithm.
`max.itr`	numeric. Maximum number of iterations.

Details

During each iteration of the algorithm: (1) Cells are identified that violate the minimum vertical overlap between adjacent cells; that is, the bottom of cell i is greater than or equal to the top of an adjacent cell j minus the minimum overlap specified by the `min.overlap` argument. (2) For cells violating the minimum vertical overlap, lower raster layer (`rs[[2]]`) values are decreased by the value specified in the `bump.by` argument.

Value

Returns a raster layer that can be added to `rs[[2]]` to ensure connectivity between cells. Cell values in the returned raster grid represent vertical adjustments.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

Examples

```
set.seed(0)
r.top <- raster::raster(ncols = 10, nrows = 10)
r.bot <- raster::raster(ncols = 10, nrows = 10)
r.top[] <- rnorm(raster::ncell(r.top), mean = 12)
r.bot[] <- rnorm(raster::ncell(r.bot), mean = 10)
summary(r.top - r.bot)
```

```
r <- BumpDisconnectCells(raster::stack(r.top, r.bot), min.overlap = 0.1)
raster::plot(r.bot + r)
```

`BumpRiverStage` *Adjustment for Implausible River Stage*

Description

This function decreases stage values in river cells if they are implausible with respect to water always flowing downhill.

Usage

```
BumpRiverStage(r, outlets, min.drop = 1e-06)
```

Arguments

`r`	RasterLayer. Numeric cell values represent river stages.
`outlets`	SpatialPoints*, SpatialLines*, SpatialPolygons* or Extent. Designates the location of discharge outlets. The `rasterize` function is used to locate outlet cells in the raster grid `r`.
`min.drop`	numeric. Minimum drop in stage between adjacent river cells.

Details

The Lee algorithm (Lee, 1961) is used to identify flow paths among the modeled river cells. An analysis of river cell stage values along a flow path identifies any problematic cells that are obstructing downhill surface-water flow. Stage values for these problematic cells are then lowered to an acceptable elevation.

Value

Returns a RasterLayer with cell values representing the vertical change in stream stage. These changes can be added to `r` to ensure that water always flows downhill.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

References

Lee, C.Y., 1961, An algorithm for path connections and its applications: IRE Transactions on Electronic Computers, v. EC-10, no. 2, p. 346–365.

Examples

```
print("notyet")
```

ExportRasterStack *Export Raster Stack*

Description

This function writes a raster-stack, a collection of raster layers, to local directories using multiple file formats.

Usage

```
ExportRasterStack(rs, path, zip = "", col = grDevices::rainbow(250, start =
  0, end = 0.8))
```

Arguments

rs	RasterStack. A collection of RasterLayer objects with the same extent and resolution.
path	character. Path name to write raster stack.
zip	character. If there is no zip program on your path (on windows), you can supply the full path to a 'zip.exe' here, in order to make a KMZ file.
col	character. Vector of colors

Details

Five local directories are created under path and named after their intended file formats: Comma-Separated Values ('csv'), Portable Network Graphics ('png'), Georeferenced TIFF ('tif'), R Data ('rda'), and Keyhole Markup Language ('kml'). For its reference system, 'kml' uses geographic coordinates: longitude and latitude components as defined by the World Geodetic System of 1984. Therefore, the conversion of gridded data between cartographic projections may introduce a new source of error.

To install 'zip.exe' on windows, download the latest binary version from the Info-ZIP website; select one of the given FTP locations, enter directory 'win32', download 'zip300xn.zip', and extract.

Value

Used for the side-effect files written to disk.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

writeRaster

Examples

```
## Not run:
  f <- "SIR2016-5080/ancillary/uncalibrated/data/rda/rasters.rda"
  load(file = f)
  ExportRasterStack(rs, tempdir())

## End(Not run)
```

`ExtractAlongTransect` *Extract Raster Values Along Transect Line*

Description

This function extracts values from raster layer(s) along a user defined transect line.

Usage

```
ExtractAlongTransect(transect, r)
```

Arguments

`transect`	SpatialPoints or SpatialLines. Transect line or its vertices.
`r`	RasterLayer, RasterStack, or RasterBrick. Raster layer(s)

Details

The transect line is described using a simple polygonal chain. The transect line and raster layer(s) must be specified in a coordinate reference system.

Value

A `list` is returned with components of class SpatialPointsDataFrame. These components represent continuous piecewise line segments along the transect. The following variables are specified for each coordinate point in the line segment:

dist distance along the transect line.

2, . . . , n extracted value for each raster layer in `r`, where column names match their respective raster layer name.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

`PlotCrossSection`

Examples

```
library(raster)

coords <- rbind(c(-100, -90), c(80, 90), c(80, 0), c(40, -40))
crs <- CRS("+proj=longlat +datum=WGS84")
transect <- SpatialPoints(coords, proj4string = crs)
r <- raster(nrows = 10, ncols = 10, ymn = -80, ymx = 80, crs = crs)
names(r) <- "value"
set.seed(0)
r[] <- runif(ncell(r))
r[4, 6] <- NA
plot(r, xlab = "x", ylab = "y")
lines(SpatialLines(list(Lines(list(Line(coords)), ID = "Transect")), proj4string = crs))
points(transect, pch = 21, bg = "red")
segs <- ExtractAlongTransect(transect, r)
for (i in 1:length(segs)) points(segs[[i]], col = "blue")

dev.new()
xlab <- "Distance along transect"
ylab <- "Raster value"
xlim <- range(vapply(segs, function(seg) range(seg@data[, "dist"]), c(0, 0)))
ylim <- range(vapply(segs, function(seg) range(seg@data[, "value"], na.rm = TRUE),
                     c(0, 0)))
plot(NA, type = "n", xlab = xlab, ylab = ylab, xlim = xlim, ylim = ylim)
for (i in 1:length(segs))
  lines(segs[[i]]@data[, c("dist", "value")], col = rainbow(length(segs))[i])
coords <- coordinates(transect)
n <- length(transect)
d <- cumsum(c(0, as.matrix(dist((coords)))[cbind(1:(n - 1), 2:n)]))
abline(v = d, col = "grey", lty = 2)
mtext(paste0("(", paste(head(coords, 1), collapse = ", "), ")"), adj = 0)
mtext(paste0("(", paste(tail(coords, 1), collapse = ", "), ")"), adj = 1)
```

GetDaysInMonth *Number of Days in a Year and Month*

Description

This function determines the number of days in a year and month.

Usage

```
GetDaysInMonth(x)
```

Arguments

x	integer. Vector of year and month values, with a required date format of YYYYMM.

Value

Returns an integer vector indicating the number of days in each year and month value specified in x.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

Examples

```
GetDaysInMonth(c("199802", "199804", "200412"))
```

PlotCrossSection *Plot Method for Cross Sections*

Description

This function creates a cross-section view of raster data. A key showing how the colors map to raster values is shown below the map. The width and height of the graphics region will be automagically determined in some cases.

Usage

```
PlotCrossSection(transect, rs, geo.lays = names(rs), val.lays = NULL,
  wt.lay = NULL, asp = 1, ylim = NULL, max.dev.dim = c(43, 56),
  n = NULL, breaks = NULL, pal = NULL, col = NULL, ylab = NULL,
  unit = NULL, id = c("A", "A'"), labels = NULL, explanation = NULL,
  features = NULL, max.feature.dist = Inf, draw.key = TRUE,
  draw.sep = TRUE, is.categorical = FALSE, contour.lines = NULL,
  bg.col = "#E1E1E1", wt.col = "#FFFFFFD8", file = NULL)
```

Arguments

transect	SpatialLines. Piecewise linear transect line.
rs	RasterStack. Collection of RasterLayer objects with the same extent and resolution.
geo.lays	character. Vector of names in rs that specify the geometry raster layers; these must be given in decreasing order, that is, from the upper most (such as land surface) to the lowest (such as a bedrock surface).
val.lays	character. Vector of names in rs that specify the value raster layers (optional). Values from the first layer are mapped as colors to the area between the first and second geometry layers; the second layer mapped between the second and third geometry layers, and so on.
wt.lay	character. The name in rs that specifies the water-table raster layer (optional).
asp	numeric. The *y/x* aspect ratio for spatial axes.

ylim	numeric. Vector of length 2 giving the minimum and maximum values for the *y*-axis.
max.dev.dim	numeric. Vector of length 2 giving the maximum width and height for the graphics device in picas, respectively. Suggested dimensions for single-column, double-column, and sidetitle figures are `c(21, 56)`, `c(43, 56)`, and `c(56, 43)`, respectively. This argument is only applicable when the `file` argument is specified.
n	integer. Desired number of intervals to partition the range of raster values (optional).
breaks	numeric. Vector of break points used to partition the colors representing numeric raster values (optional).
pal	function. Color palette to be used to assign colors in the plot, rainbow by default.
col	character. Vector of colors to be used in the plot. This argument requires `breaks` specification for numeric raster values and overrides any palette function specification. For numeric values there should be one less color than breaks. Categorical data require a color for each category.
ylab	character. Label for the *y* axis.
unit	character. Label for the measurement unit of the *x*- and *y*-axes.
id	character. Vector of length 2 giving the labels for the end points of the transect line, defaults to *A–A'*.
labels	list. Describes the location and values of labels in the color key. This list may include components `at` and `labels`.
explanation	character. Label explaining the raster cell value.
features	SpatialGridDataFrame. Point features adjacent to the transect line that are used as reference labels for the upper geometry layer.
max.feature.dist	numeric. Maximum distance from a point feature to the transect line, specified in the units of the `rs` projection.
draw.key	logical. If true, a color key should be drawn.
draw.sep	logical. If true, lines separating geometry layers are drawn.
is.categorical	logical. If true, cell values in `val.lays` represent categorical data; otherwise, these data values are assumed continuous.
contour.lines	list. If specified, contour lines are drawn. The contours are described using a list of arguments supplied to `contour`. Passed arguments include `"drawlables"`, `"method"`, and `"col"`.
bg.col	character. Color used for the background of the area below the upper geometry raster layer.
wt.col	character. Color used for the water-table line.
file	character. Name of the output file. Specifying this argument will start a graphics device driver for producing a PDF or PNG file format—the file extension determines the format type. The width and height of the graphics region will be automagically determined and included with the function's returned values, see "Value" section for details. These graphical parameters can be used when

creating similar map layouts in dynamic reports or whenever further elements will be added to an existing plot generated by this function (possible when `file = NULL`).

Value

Used for the side-effect of a new plot generated. Returns a list object with the following graphical parameters:

din device dimensions `(width, height)`, in inches.

usr extremes of the coordinates of the plotting region `(x1, x2, y1, y2)`.

heights relative heights on the device `(upper, lower)` for the map and color-key plots.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

AddScaleBar, AddColorKey, ExtractAlongTransect

Examples

```
data(volcano, package = "datasets")
x <- seq(from = 2667405, length.out = 61, by = 10)
y <- seq(from = 6478705, length.out = 87, by = 10)
r1 <- raster::raster(volcano, xmn = min(x), xmx = max(x), ymn = min(y),
                     ymx = max(y), crs = sp::CRS("+init=epsg:27200"))
r2 <- min(r1[]) - r1 / 10
r3 <- r1 - r2
rs <- raster::stack(r1, r2, r3)
names(rs) <- c("r1", "r2", "r3")

xy <- rbind(c(2667508, 6479501), c(2667803, 6479214), c(2667508, 6478749))
transect <- sp::SpatialLines(list(sp::Lines(list(sp::Line(xy)), ID = "Transect")),
                             proj4string = raster::crs(rs))

raster::plot(r1)
lines(transect)
raster::text(as(transect, "SpatialPoints"), labels = c("A", "BEND", "A'"),
             cex = 0.7, pos = c(3, 4, 1), offset = 0.1, font = 4)

explanation <- "Vertical thickness between layers, in meters."
PlotCrossSection(transect, rs, geo.lays = c("r1", "r2"), val.lays = "r3",
                 ylab="Elevation", asp = 5, unit = "METERS", explanation = explanation)

val <- PlotCrossSection(transect, rs, geo.lays = c("r1", "r2"), val.lays = "r3",
                        ylab="Elevation", asp = 5, unit = "METERS",
                        explanation = explanation, file = "Rplots.png")
print(val)

file.remove("Rplots.png")
```

```
graphics.off()
```

PlotGraph *Plot Method for Graphs*

Description

This function draws a sequence of points, lines, or box-and-whiskers using specified coordinates.

Usage

```
PlotGraph(x, y, xlab, ylab, asp = NA, xlim = NULL, ylim = NULL, xn = 5L,
  yn = 5L, ylog = FALSE, type = "s", lty = 1, lwd = 1, pch = NULL,
  col = NULL, bg = NA, fill = NULL, pt.cex = 1, seq.date.by = "year",
  scientific = NA, conversion.factor = NULL, boxwex = 0.8,
  center.date.labels = FALSE, bg.polygon = NULL)
```

Arguments

x, y	Date, numeric, matrix, or data.frame. Vectors or matrices of data for plotting. The vector length or number of rows should match. If y is missing, then x = x[, 1] and y = x[, 2:n].
xlab	character. Title for *x* axis.
ylab	character. Vector of length 2 giving the title for the 1st and 2nd-*y* axes. The title for the 2nd-*y* axis is optional and requires conversion.factor be specified.
asp	numeric. The *y/x* aspect ratio for spatial axes.
xlim	numeric or Date. Vector of length 2 giving the minimum and maximum values for the *x*-axis.
ylim	numeric. Vector of length 2 giving the minimum and maximum values for the *y*-axis.
xn, yn	integer. Desired number of intervals between tick-marks on the *x*- and *y*-axis, respectively.
ylog	logical. If true, a logarithm scale is used for the *y* axis.
type	character. The type of plot for each column of y, see plot function for possible types. A box-and-whisker plot is drawn when type = "box", with whiskers extending to the data extremes.
lty	integer. The line type, see par function for all possible types. Line types are used cyclically.
lwd	numeric. Line width
pch	integer. Point type, see points function for all possible types.
col	character or function. Point or line color, see par function for all possible ways this can be specified. Colors are used cyclically.

bg	character. Vector of background colors for the open plot symbols given by pch = 21:25 as in points.
fill	character. Vector of fill colors for areas beneath (or above, direction towards 0) lines of type "l" or "s".
pt.cex	numeric. Expansion factor for the points.
seq.date.by	character, numeric, or difftime. The increment of the date sequence, see seq.Date function for all possible ways this can be specified.
scientific	logical. Vector of length 3 that indicates if axes labels should be encoded in nice scientific format. Vector elements correspond to the *x*-axis, y-axis, and second *y*-axis labels. Values are recycled as necessary.
conversion.factor	numeric. A conversion factor for the 2nd-*y* axis.
boxwex	numeric. A scale factor to be applied to all boxes, only applicable for box-and-whisker plots.
center.date.labels	logical. If true, date labels are horizontally centered between *x*-axis tickmarks.
bg.polygon	list. If specified, a background polygon is drawn. The polygon is described using a list of arguments supplied to the polygon function. Passed arguments include "x" and "col".

Value

Used for the side-effect of a new plot generated.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

matplot, boxplot

Examples

```
n <- 50L
x <- as.Date("2008-07-12") + 1:n
y <- sample.int(n, replace = TRUE)
PlotGraph(x, y, ylab = paste("Random number in", c("meters", "feet")), type = "p",
          pch = 16, seq.date.by = "weeks", scientific = FALSE, conversion.factor = 3.28)

y <- data.frame(lapply(1:3, function(i) sample(n, replace = TRUE)))
PlotGraph(x, y, ylab = "Random number", type = "s", pch = 1, seq.date.by = "days",
          scientific=TRUE)

y <- sapply(1:3, function(i) sample((1:100) + i * 100, n, replace = TRUE))
m <- cbind(as.numeric(x), y)
col <- c("red", "gold", "green")
PlotGraph(m, xlab = "Number", ylab = "Random number", type = "b", pch = 15:17,
```

```
          col = col, pt.cex = 0.9)
legend("topright", LETTERS[1:3], inset = 0.05, col = col, lty = 1, pch = 15:17,
       pt.cex = 0.9, cex = 0.8, bg = "white")

graphics.off()
```

PlotMap *Plot Method for Maps*

Description

This function maps raster layer values. A key showing how the colors map to raster values is shown below the map. The width and height of the graphics region will be automagically determined in some cases.

Usage

```
PlotMap(r, layer = 1, att = NULL, n, breaks, xlim = NULL, ylim = NULL,
  zlim = NULL, asp = 1, extend.xy = FALSE, extend.z = FALSE,
  reg.axs = TRUE, trim.r = TRUE, dms.tick = FALSE, bg.lines = FALSE,
  bg.image = NULL, bg.image.alpha = 1, pal = NULL, col = NULL,
  max.dev.dim = c(43, 56), labels = NULL, scale.loc = "bottomleft",
  arrow.loc = NULL, explanation = NULL, credit = proj4string(r),
  shade = NULL, contour.lines = NULL, rivers = NULL, lakes = NULL,
  roads = NULL, draw.key = NULL, draw.raster = TRUE, file = NULL,
  useRaster)
```

Arguments

r	RasterLayer, SpatialGridDataFrame, or CRS. A raster layer with values to be plotted or a coordinate reference system (CRS).
layer	integer. Column to use in the SpatialGridDataFrame.
att	numeric or character. Variable identifying the levels attribute to use in the Raster Attribute Table (RAT). This argument requires r values that are of class factor.
n	integer. Desired number of intervals to partition the range of raster values (or zlim if specified) (optional).
breaks	numeric. Vector of break points used to partition the colors representing numeric raster values (optional).
xlim	numeric. Vector of length 2 giving the minimum and maximum values for the *x*-axis.
ylim	numeric. Vector of length 2 giving the minimum and maximum values for the *y*-axis.
zlim	numeric. Vector of length 2 giving the minimum and maximum raster values for which colors should be plotted.

asp	numeric. The *y/x* aspect ratio for spatial axes.
extend.xy	logical. If true, the spatial limits will be extended to the next tick mark on the axes beyond the grid extent.
extend.z	logical. If true, the raster value limits will be extended to the next tick mark on the color key beyond the measured range.
reg.axs	logical. If true, the spatial data range is extended.
trim.r	logical. If true, the outer rows and columns that consist of all NA values will be removed.
dms.tick	logical. If true, the axes tickmarks are specified in degrees, minutes, and decimal seconds.
bg.lines	logical. If true, the graticule is drawn in back of the raster layer using white lines and a grey background.
bg.image	RasterLayer. An image to drawn in back of the main raster layer r.
bg.image.alpha	numeric. Opacity of the background image from 0 to 1.
pal	function. Color palette to be used to assign colors in the plot, rainbow by default.
col	character. Vector of colors to be used in the plot. This argument requires breaks specification for numeric values of r and overrides any palette function specification. For numeric values there should be one less color than breaks. Factors require a color for each level.
max.dev.dim	numeric. Vector of length 2 giving the maximum width and height for the graphics device in picas, respectively. Suggested dimensions for single-column, double-column, and sidetitle figures are c(21, 56), c(43, 56), and c(56, 43), respectively. This argument is only applicable when the file argument is specified.
labels	list. Describes the location and values of labels in the color key. This list may include components at and labels.
scale.loc	character. Position of the scale bar: "bottomleft", "topleft", "topright", or "bottomright" to denote scale location.
arrow.loc	character. Position of the north arrow: "bottomleft", "topleft", "topright", or "bottomright" to denote arrow location.
explanation	character. Label explaining the raster cell value.
credit	character. Label crediting the base map; by default, the character string describing the raster layer's (r) projection and datum in the PROJ.4 format.
shade	list. If specified, a semi-transparent shade layer is drawn on top of the raster layer. This layer is described using a list of arguments supplied to raster::hillShade function. Passed arguments include "angle" and "direction". Additional arguments also may be passed that control the vertical aspect ratio ("z.factor") and color opacity ("alpha").
contour.lines	list. If specified, contour lines are drawn. The contours are described using a list of arguments supplied to contour. Passed arguments include "drawlables", "method", and "col".
rivers	list. If specified, lines are drawn. The lines are described using a list of arguments supplied to the plot method for SpatialLines. Passed arguments include "x", "col", and "lwd".

lakes	list. If specified, polygons are drawn. The polygons are described using a list of arguments supplied to the plot method for SpatialPolygons. Passed arguments include "x", "col", "border", and "lwd". Bitmap images require a regular grid.
roads	list. If specified, lines are drawn. The lines are described using a list of arguments supplied to the plot method for SpatialLines. Passed arguments include "x", "col", and "lwd".
draw.key	logical. If true, a color key should be drawn.
draw.raster	logical. If true, the raster image is drawn.
file	character. Name of the output file. Specifying this argument will start a graphics device driver for producing a PDF or PNG file format—the file extension determines the format type. The width and height of the graphics region will be automagically determined and included with the function's returned values, see "Value" section for details. These graphical parameters can be used when creating similar map layouts in dynamic reports or whenever further elements will be added to an existing plot generated by this function (possible when file = NULL).
useRaster	logical. If true, a bitmap raster is used to plot r instead of using polygons. If UseRaster is not specified, raster images are used when the getOption("preferRaster") is true.

Value

Used for the side-effect of a new plot generated. Returns a list object with the following graphical parameters:

din device dimensions (width, height), in inches.

usr extremes of the coordinates of the plotting region (x1, x2, y1, y2).

heights relative heights on the device (upper, lower) for the map and color-key plots.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

AddScaleBar, AddColorKey

Examples

```
r <- raster::raster(nrow = 10, ncol = 10)
r[] <- 1L
r[51:100] <- 2L
r[3:6, 1:5] <- 8L
r <- raster::ratify(r)
rat <- raster::levels(r)[[1]]
rat$land.cover <- c("Pine", "Oak", "Meadow")
rat$code <- c(12, 25, 30)
```

```
levels(r) <- rat
PlotMap(r, att = "land.cover", col = c("grey", "orange", "purple"))
PlotMap(r, att = "code")

r <- raster::raster(system.file("external/test.grd", package="raster"))
credit <- "Label crediting the base map."
explanation <- "Label explaining the raster cell value."
PlotMap(r, scale.loc = "topleft", dms.tick = TRUE, trim.r = TRUE,
        credit = credit, explanation = explanation)
data(meuse, package = "sp")
sp::coordinates(meuse) = ~ x + y
points(meuse)

val <- PlotMap(r, scale.loc = "topleft", dms.tick = TRUE, trim.r = TRUE,
               credit = credit, explanation = explanation,
               file = "Rplots1.pdf")
print(val)

pdf(file = "Rplots2.pdf", width = val$din[1], height = val$din[2])
PlotMap(r, scale.loc = "topleft", dms.tick = TRUE, trim.r = TRUE,
        credit = credit, explanation = explanation)
points(meuse)
dev.off()

file.remove(c("Rplots1.pdf", "Rplots2.pdf"))
graphics.off()
```

`ReadCodeChunks` *Read Code Chunks*

Description

This function reads **knitr** code chunks into the current session.

Usage

```
ReadCodeChunks(path)
```

Arguments

`path`	character. Path name of the **knitr** source document ('`.Rnw`' or '`.Rmd`'), or R code that has been extracted from a **knitr** source document ('`.R`').

Details

If the source document is '`.Rnw`' or '`.Rmd`' the `purl` function is used to extract the R code. The R code is read into the current session using a chunk separator of the from `## ---- chunk-name` (at least four dashes before the chunk name) in the script. Unnamed chunks (that is, `chunk-name` is missing) will be assigned names like `unnamed-chunk-i` where `i` is the chunk number.

Value

Returns a `list` object of length equal to the number of code chunks in `path`. Each `list` component is named after its corresponding chunk name (`chunk-name`). The returned object includes the value of the `path` argument as an attribute.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

`read_chunk`

Examples

```
path <- system.file("doc", "knitr-intro.Rmd", package = "knitr")
chunks <- ReadCodeChunks(path)

attr(chunks, "path")
names(chunks)
chunks[["show-off"]]

eval(parse(text = unlist(chunks[c("show-off", "graphics")])))
```

ReadModflowBinary *Read MODFLOW Binary File*

Description

This is a utility function for MODFLOW. It reads binary output data from a model run.

Usage

```
ReadModflowBinary(path, data.type = c("array", "flow"), rm.totim.0 = FALSE)
```

Arguments

`path`	character. Path name of the binary file.
`data.type`	character. Description of how the data are saved.
`rm.totim.0`	logical. If true, stress-periods at time zero are removed.

Details

This function reads binary head (‘`.hds`’), drawdown (‘`.ddn`’), and budget (‘`.bud`’) files generated from a MODFLOW run.

Value

Returns a `list` object of length equal to the number of times the data type is written to the binary file. The following list components are returned:

d matrix of data values.

kstp time step

kper stress period

desc variable name

ilay model-grid layer

delt length of the current time step.

pertim time in the stress period.

totim total elapsed time

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

`SummariseBudget`

Examples

```
path <- system.file("extdata", "ex3D.hds", package = "inlmisc")
hds <- ReadModflowBinary(path, "array")
```

ReplaceInTemplate *Replace Values in a Template Text*

Description

This function replaces keys within special markups in a template text with specified values. Pieces of R code can be put into the markups of the template text, and are evaluated during the replacement.

Usage

```
ReplaceInTemplate(text, replacement)
```

Arguments

`text`	character. Vector of character strings, that is the template text.
`replacement`	list. Values to replace in `text`.

Details

Keys are enclosed into markups of the form `$(KEY)` and `@{CODE}`.

Value

Returns a vector of character strings after key replacement.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

References

This code was derived from the sensitivity`::template.replace` function, accessed on Feb 6, 2015.

See Also

`SummariseBudget`

Examples

```
text <- c("Hello $(name)!", "$(a) + $(b) = @{$(a) + $(b)}",
          "pi = @{format(pi, digits = 5)}")
replacement <- list(name = "world", a = 1, b = 2)
cat(ReplaceInTemplate(text, replacement), sep = "\n")
```

`RmSmallCellChunks` *Remove Small Cell Chunks*

Description

This function identifies cell chunks in a single raster grid layer, where a cell chunk is defined as a group of connected cells with non-missing values. The cell chunk with the largest surface area is preserved and all others removed.

Usage

```
RmSmallCellChunks(r)
```

Arguments

`r`	RasterLayer. A raster grid layer with cell values.

Value

The raster grid layer `r` with cell values in the smaller cell chunks set to `NA`.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

clump

Examples

```
set.seed(10)
r <- raster::raster(ncols = 10, nrows = 10)
r[] <- round(runif(raster::ncell(r)) * 0.7)
r <- raster::clump(r)
raster::plot(r)

r.new <- RmSmallCellChunks(r)
raster::plot(r.new, zlim = range(r[], na.rm = TRUE))
```

SetPolygons *Overlaying Multi-Polygon Objects*

Description

Determines the intersection or difference between two multi-polygon objects.

Usage

```
SetPolygons(x, y, cmd = c("gIntersection", "gDifference"),
  buffer.width = NA)
```

Arguments

x	SpatialPolygons*. Multi-polygon object
y	SpatialPolygons* or Extent. Multi-polygon object
cmd	character. Specifying "gIntersection", the default, cuts out portions of the x polygons that overlay the y polygons. If "gDifference" is specified, only those portions of the x polygons falling outside the y polygons are copied to the output polygons.
buffer.width	numeric. Expands or contracts the geometry of y to include the area within the specified width, see gBuffer. Specifying NA, the default, indicates no buffer.

Details

This function tests if the resulting geometry is valid, see gIsValid.

Value

Returns an object of SpatialPolygons* class.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

gIntersection, gDifference

Examples

```
m1a <- matrix(c(17.5, 24.7, 22.6, 16.5, 55.1, 55.0, 61.1, 59.7), nrow = 4, ncol = 2)
m1b <- m1a
m1b[, 1] <- m1b[, 1] + 11
p1 <- sp::SpatialPolygons(list(sp::Polygons(list(sp::Polygon(m1a, FALSE),
                                                 sp::Polygon(m1b, FALSE)), 1)))
sp::plot(p1, col = "blue")

m2a <- matrix(c(19.6, 35.7, 28.2, 60.0, 58.8, 64.4), nrow = 3, ncol = 2)
m2b <- matrix(c(20.6, 30.9, 27.3, 56.2, 53.8, 51.4), nrow = 3, ncol = 2)
p2 <- sp::SpatialPolygons(list(sp::Polygons(list(sp::Polygon(m2a, FALSE),
                                                 sp::Polygon(m2b, FALSE)), 2)))
sp::plot(p2, col = "red", add = TRUE)

p <- SetPolygons(p1, p2, "gIntersection")
sp::plot(p, col = "green", add = TRUE)

p <- SetPolygons(p2, p1, "gDifference")
sp::plot(p, col = "purple", add = TRUE)
```

SummariseBudget *Summarize MODFLOW Water Budget*

Description

This is a utility function for MODFLOW. It summarizes volumetric flow rates by boundary condition types. That is, it splits the MODFLOW water-budget data into subsets, computes summary statistics for each, and returns the result in a summary table.

Usage

```
SummariseBudget(budget, desc = c("wells", "drains", "river leakage"))
```

Arguments

budget	character or list. Either a description of the path to the MODFLOW Budget File or the returned results from a call to the ReadModflowBinary function.
desc	character. Vector of MODFLOW package identifiers. Data of this package type is included in the summary table.

Details

The budget[[i]]$d data table component must contain a numeric id field. Subsets are grouped by the MODFLOW package identifier (desc), stress period (kper), time step (kstp), and location identifier (id).

Value

Returns a data.frame object with the following variables:

desc MODFLOW package identifier

kper stress period

kstp time step

id location identifier

delt length of the current time step.

pertim time in the stress period.

totim total elapsed time

count number of cells in each subset.

flow.sum total volumetric flow rate

flow.mean mean volumetric flow rate

flow.median median volumetric flow rate

flow.sd tandard deviation of the volumetric flow rate.

flow.dir flow direction where "in" and "out" indicate water entering and leaving the groundwater system, respectively.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

ReadModflowBinary

Examples

```
## Not run:
  d <- SummariseBudget("modflow.bud")

## End(Not run)
```

`ToScientific` *Format for Scientific Notation*

Description

This function formats numbers in scientific notation $m \times 10^n$.

Usage

```
ToScientific(x, digits = format.info(as.numeric(x))[2],
  lab.type = c("latex", "plotmath"))
```

Arguments

`x`	numeric. Vector of numbers
`digits`	integer. Number of digits after the decimal point for the mantissa.
`lab.type`	character. By default, LaTeX formatted strings for labels are returned. Alternatively, `lab.type = "plotmath"` returns plotmath-compatible expressions.

Value

For the default `lab.type = "latex"`, a `character` vector of the same length as argument `x`. And for `lab.type = "plotmath"`, an expression of the same length as `x`, typically with elements of the form `m x 10^n`. In order to comply with Section 508, an "x" is used as the label separator for the `plotmath` type—rather than the more common "%*%" seperator.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

Examples

```
x <- c(-1e+09, 0, NA, pi * 10^(-5:5))
ToScientific(x, digits = 2)
ToScientific(x, digits = 2, lab.type = "plotmath")
```

Index