Package ‘Orcs’
March 5, 2020

Type Package
Title Omnidirectional R Code Snippets
Version 1.2.1
Date 2020-03-05
Maintainer Florian Detsch <fdetsch@web.de>
Description I tend to repeat the same code chunks over and over again. At first, this was fine for me and I paid little attention to such redundancies. A little later, when I got tired of manually replacing Linux filepaths with the referring Windows versions, and vice versa, I started to stuff some very frequently used work-steps into functions and, even later, into a proper R package. And that's what this package is - a hodgepodge of various R functions meant to simplify (my) everyday-life coding work without, at the same time, being devoted to a particular scope of application.
License MIT + file LICENSE
URL https://github.com/fdetsch/Orcs
BugReports https://github.com/fdetsch/Orcs/issues
LazyData TRUE
Depends R (>= 2.10), methods, raster
Imports bookdown, grDevices, grid, knitr, lattice, latticeExtra, plotrix, Rcpp (>= 0.11.3), remotes, rgdal, sf, sp, stats
LinkingTo Rcpp
RoxygenNote 7.0.2
SystemRequirements GNU make, 7zip, unix2dos
Suggests testthat, rmarkdown
VignetteBuilder knitr
Encoding UTF-8
NeedsCompilation yes
Author Florian Detsch [cre, aut],
Tim Appelhans [ctb],
Baptiste Auguié [ctb],
OpenStreetMap contributors [cph]
Repository  CRAN
Date/Publication  2020-03-05 15:50:02 UTC

R topics documented:

- Orcs-package ................................................ 3
- assignSSH .................................................. 3
- buildBook .................................................. 4
- bumpVersion ................................................. 5
- compareDistributions ....................................... 6
- coords2Lines ............................................... 7
- coords2Polygons ......................................... 8
- evalMetrics ................................................ 9
- ext2spy ..................................................... 10
- ifMissing .................................................. 11
- KiLi ......................................................... 12
- latticeCombineGrid ....................................... 12
- latticeCombineLayer ..................................... 14
- lineEnding ................................................ 15
- list2df ..................................................... 16
- loadFromGit .............................................. 17
- loadPkgs .................................................. 17
- meanDifference .......................................... 18
- merge ...................................................... 19
- multiKnit ................................................ 20
- offsetGridText .......................................... 21
- OrcsCppFun ............................................... 22
- par7zip .................................................... 23
- pureBasename ............................................. 24
- pvalue ..................................................... 25
- rgb2spLayout ............................................. 26
- rmDuplicols ............................................... 27
- setwdOS .................................................. 28
- stextGrob ............................................... 29
- substrC ................................................... 30
- trimImages ................................................ 31
- unlistStrsplit ............................................ 32
- unsortedFactor .......................................... 33

Index  34
Description

Omnidirectional R Code Snippets

Details

The package provides a variety of functions which I regularly use during my everyday work.

Author(s)

Florian Detsch

Maintainer: Florian Detsch <florian.detsch@staff.uni-marburg.de>

assignSSH

Assign SSH Key to Local Git Repository

Description

Assign an SSH key to a local Git repository to bypass user/password prompts during git push. See Generating an SSH Key for further information on how to generate an SSH key and add it to your GitHub account.

Usage

assignSSH(user, repo)

Arguments

user GitHub user name as character. If not specified, information on GitHub user and repository name is taken from the current working environment.

repo GitHub repository name as character, see 'user'.

See Also

https://help.github.com/articles/generating-an-ssh-key/
Examples

```r
## Not run:
## for an arbitrary git repository
assignSSH()

## for this very git repository
assignSSH(user = "fdetsch", repo = "Orcs")

## End(Not run)
```

---

**buildBook**  
Build a Book without Underscores

**Description**

Since the use of underscores ('_') is not permitted when streaming `bookdown` documents via GitHub Pages, this wrapper function serves to remove any unwanted underscores from subfolders and link `.html` documents created by `render_book`.

**Usage**

```r
buildBook(output_dir = "book", ...)
```

**Arguments**

- `output_dir`  
  Output directory as character.
- `...`  
  Arguments passed to `render_book`.

**Note**

While all remaining arguments passed to `render_book` remain untouched, and hence, their specification is freely up to the user, the default value of 'output_dir' is explicitly set to "book" here. If this were not the case (i.e. if the default value were used), the output document would be created in "_book" which is not desirable for obvious reasons.

**Author(s)**

Florian Detsch

**See Also**

`render_book`
Description

This function lets you bump the version number and creation date of your package's DESCRIPTION file. Additionally, it bumps the version numbers of a NEWS.md file and automatically generates a corresponding plain NEWS file (for R-help pages). Supported versioning system is MAJOR.MINOR.PATCH.

Usage

bumpVersion(
  element = "patch",
  pkg.repo = ".",
  news = file.path(pkg.repo, "NEWS.md"),
  plain_news = TRUE
)

Arguments

- **element**: character, one of "major", "minor", "patch" (default) to be bumped.
- **pkg.repo**: Path to package repository folder. Default is current working directory (".").
- **news**: The NEWS.md file of the repo (assumed to be in top level path). If this exists, the first line of that file will be rewritten to be "<packagename> <major.minor.patch>". Note that the current implementation assumes that the NEWS file is in .md format, thus NEWS.md. A plain NEWS file (for R-help pages) will be generated automatically.
- **plain_news**: whether to generate a plain NEWS file in the package root directory from the NEWS.md file supplied to argument news.

Author(s)

Tim Appelhans

See Also

https://semver.org/
compareDistributions  

Compare two density distributions side by side

Description

This function will produce a plot of two density functions displayed side by side.

Usage

```r
compareDistributions(
  left,
  right,
  add.spread = TRUE,
  print.stats = TRUE,
  xlim = NULL,
  ylim = NULL,
  clrs = c("purple", "lightblue"),
  xlab = "density",
  ylab = "value",
  ...
)
```

Arguments

- `left`  numeric vector.
- `right` numeric vector.
- `add.spread` logical, whether to plot the spread (q25 to q75 and the median).
- `print.stats` logical, whether to print summary statistics for each distribution.
- `xlim`, `ylim` numeric axis limits, see `xyplot`.
- `clrs` A character vector of length 2 specifying the colors for the filled density regions.
- `xlab`, `ylab` character axis labels, see `plot`.
- `...` Additional arguments passed to `density`.

Value

A trellis object.

Author(s)

Tim Appelhans
Examples

```r
compareDistributions(rnorm(1000, 2, 3), rnorm(1000, -5, 1))
compareDistributions(rnorm(1000, 2, 3), rnorm(1000, -5, 1),
  add.spread = FALSE)
compareDistributions(rnorm(1000, 2, 3), rnorm(1000, -5, 1),
  add.spread = TRUE, clrs = c("red", "brown"))
compareDistributions(rnorm(1000, 2, 5), rnorm(1000, -5, 4),
  print.stats = FALSE, add.spread = FALSE)
## pass additional parameters to density()
compareDistributions(rnorm(1000, 2, 5), rnorm(1000, -5, 4),
  print.stats = FALSE, add.spread = FALSE, bw = 5)
compareDistributions(rnorm(1000, 2, 5), rnorm(1000, -5, 4),
  print.stats = FALSE, add.spread = FALSE, bw = 8,
  kernel = "rectangular")
compareDistributions(rnorm(1000, 2, 5), rnorm(1000, -5, 4),
  print.stats = FALSE, add.spread = TRUE, bw = 8,
  n = 3)
compareDistributions(rnorm(1000, 2, 5), rnorm(1000, -5, 4),
  print.stats = TRUE, add.spread = TRUE, bw = 0.1)
compareDistributions(rnorm(1000, 2, 5), rnorm(1000, -5, 4),
  print.stats = TRUE, add.spread = TRUE, bw = 0.5)
```

### coords2Lines

Convert points to `SpatialLines`

**Description**

Create a `SpatialLines` object from a `Line` object or set of point coordinates in one go, i.e. without being required to run through the single steps outlined in `SpatialLines`.

**Usage**

```r
## S4 method for signature 'matrix'
coords2Lines(coords, ID, data, match.ID = TRUE, ...)

## S4 method for signature 'Line'
coords2Lines(coords, ID, data, match.ID = TRUE, ...)
```

**Arguments**

- `coords` Line object or 2-column numeric matrix with x and y coordinates.
- `ID` character, see `Lines`.
- `data` data.frame with data to add to the output `SpatialLines` object (optional).
- `match.ID` logical, see `SpatialLinesDataFrame`.
- `...` Further arguments passed on to `SpatialLines` (i.e., `proj4string`).
Value

If data is missing, a SpatialLines object; else a SpatialLinesDataFrame object.

See Also

SpatialLines-class, SpatialLinesDataFrame.

Examples

library(sp)

coords1 <- cbind(c(2, 4, 4, 1, 2), c(2, 3, 5, 4, 2))
sln1 <- coords2Lines(coords1, ID = "A")

coords2 <- cbind(c(5, 4, 2, 5), c(2, 3, 2, 2))
sln2 <- coords2Lines(coords2, ID = "B")

plot(sln1, col = "grey75")
plot(sln2, col = "grey25", add = TRUE)
evalMetrics

**Value**

If data is missing, a SpatialPolygons object; else a SpatialPolygonsDataFrame object.

**See Also**

SpatialPolygons-class, SpatialPolygonsDataFrame.

**Examples**

```r
library(sp)

coords1 <- cbind(c(2, 4, 4, 1, 2), c(2, 3, 5, 4, 2))
spy1 <- coords2Polygons(coords1, ID = "A")

coords2 <- cbind(c(5, 4, 2, 5), c(2, 3, 2, 2))
spy2 <- coords2Polygons(coords2, ID = "B")

plot(spy1, col = "grey75")
plot(spy2, col = "grey25", add = TRUE)
```

### evalMetrics

**Compute Selected Evaluation Metrics**

**Description**

Compute selected evaluation metrics for binary (i.e. two-class) confusion matrices.

**Usage**

```r
evalMetrics(mat, type = c("accuracy", "precision", "recall"))
```

**Arguments**

- `mat`: Binary confusion matrix (2-by-2; see Examples).
- `type`: Target evaluation metric as character, defaults to "accuracy". Other available options are "precision" and "recall".

**Value**

A single numeric.

**Author(s)**

Florian Detsch
References

Examples
```r
in1 = matrix(c(96, 4, 8, 19), nc = 2L, byrow = TRUE)
rownames(in1) = c("Condition Positive", "Condition Negative")
colnames(in1) = c("Predicted Positive", "Predicted Negative")

evalMetrics(in1) # default: "accuracy"
evalMetrics(in1, "precision")
evalMetrics(in1, "recall")

in2 = matrix(c(26, 17, 7, 400), nc = 2, byrow = TRUE)
evalMetrics(in2, "precision")
evalMetrics(in2, "recall")
```

---

### ext2spy

Convert Spatial Extent to Polygon

#### Description
Convert a spatial extent to polygons.

#### Usage
```r
ext2spy(x, crs = "+init=epsg:4326", as_sf = TRUE)
```

#### Arguments
- **x**: An Extent object, or any object from which an Extent can be extracted, e.g. Raster*
- **crs**: Coordinate reference system passed to proj4string.
- **as_sf**: logical. If TRUE (default), the returned object is of class sf rather than Spatial*.

#### Value
Depending on `'as_sf'`, either a sf or SpatialPolygons object.

#### Author(s)
Florian Detsch

#### See Also
- `extent`
ifMissing

Examples

```r
ext = extent(c(25, 70, -5, 30))
ext2spy(ext) # 'sf' (default)
ext2spy(ext, as_sf = FALSE) # 'Spatial*
```

ifMissing

Take measures in case of nonexisting target files

Description

If a target file already exists, it is simply being imported into R. However, if the specified target file does not exist, it is first created by a user-defined function and subsequently returned, thus rendering explicit calls to `file.exists` unnecessary.

Usage

```r
ifMissing(ofl, fun0 = raster::brick, fun1 = raster::writeRaster, arg1, ...)
```

Arguments

- `ofl` Target file name as character.
- `fun0` If `ofl` exists, function to be applied to it (defaults to `brick`).
- `fun1` If `ofl` does not exist, function used to create it (defaults to `writeRaster`).
- `arg1` Argument in `fun1` (as character) that corresponds to `ofl`, e.g. `filename` in `writeRaster` or `file` in `write.table`. If missing (default), the target file name passed to `fun1` needs to be explicitly included via `...`.
- `...` Additional arguments passed to `fun0`, `fun1`.

Value

If `ofl` has already existed, the contents of `ofl` derived from `fun0`; else the output resultant from `fun1`.

Author(s)

Florian Detsch

See Also

`file.exists`, `do.call`.
Examples

# simply import existing file
logo <- system.file("external/rlogo.grd", package = "raster")
s <- ifMissing(logo)

# create nonexisting file and import it afterwards
logo2 <- file.path(tempdir(), "rlogo.tif")
s2 <- ifMissing(logo2, arg1 = "filename", x = s, datatype = "INT1U")

# this also works with text files and more sophisticated custom functions
fun = function(x, file = "", ...) {
  write.csv(x, file, ...
  read.csv(file)
}

data(iris)
ofl <- file.path(tempdir(), "iris.csv")
iris2 <- ifMissing(ofl, fun1 = fun, x = iris, file = ofl, quote = FALSE, row.names = FALSE)

---

KiLi  
*Bing Aerial Image of Kilimanjaro*

Description

Bing aerial image of Kilimanjaro downloaded from OpenStreetMap.

Format

A "RasterStack-class" with 3 bands (red, green, blue).

Details

Copyright: OpenStreetMap contributors, see https://www.openstreetmap.org/copyright.

---

latticeCombineGrid  
*Combine multiple lattice plots in a faceted grid (panels)*

Description

This function combines multiple lattice plot objects in a faceted grid. Note that the global plot settings (e.g. xlim, ylim, ...) are taken from the first object though the user can specify whether scales should be identical or not. This is particularly useful when looping over large amounts of data using `lapply` (see examples).
latticeCombineGrid

Usage

latticeCombineGrid(
  trellis.list,
  between = list(y = 0.3, x = 0.3),
  as.table = TRUE,
  ...
)

Arguments

trellis.list  A list containing lattice plot objects.
between       Space between panels.
as.table      If TRUE (default) drawing is top left to bottom right
...            Additional arguments passed to c.trellis.

Value

A single lattice plot object.

Author(s)

Tim Appelhans

See Also

c.trellis.

Examples

#load data
#Use a probability map assuming high potential for city expansion is just
#resulting from proximity to current urban area:
pred <- raster(system.file("extdata/probability.rst", package = "Orcs"))

#observed city growth between 1990 and 2006
obs <- raster(system.file("extdata/citygrowth.tif", package = "Orcs"))

#masking current urban area since these pixels have no potential for change
mask <- raster(system.file("extdata/citymask.tif", package = "Orcs"))

#create data list
dat <- list(pred, obs, mask)

#create list of lattice plots
plist <- lapply(seq(dat), function(i) {
  spplot(dat[[i]], scales = list(draw = TRUE))
})

#draw individually
# combine to grid, using c(1, 3) layout
p <- latticeCombineGrid(plist, layout = c(1, 3))
print(p)

---

**latticeCombineLayer**

Combine multiple plot objects drawing each as a layer on top of the previous plots. Note that the global plot settings (e.g. xlim, ylim, ...) are taken from the first object. This is particularly useful when looping over large amounts of data using *lapply* (see examples).

**Usage**

```
latticeCombineLayer(trellis.list, ...)
```

**Arguments**

- `trellis.list` A list containing *lattice* plot objects.
- `...` Additional arguments passed to *as.layer*.

**Value**

A single *lattice* plot object.

**Author(s)**

Tim Appelhans

**See Also**

*as.layer*.

**Examples**

```
library(latticeExtra)
dat <- list(1:10,
            10:1,
            3:7,
            7:3)

plist <- lapply(seq(dat), function(i) {

```
tmp <- xyplot(dat[[i]] ~ seq(dat[[i]]),
    type = "l", col = i)
}

p <- latticeCombineLayer(plist)

print(p)

### lineEnding

*Convert between DOS and UNIX line endings*

#### Description

This function converts between DOS and UNIX style line endings by invoking `unix2dos` (or `dos2unix`) upon a text file (see also `system("unix2dos --help")`). Note that `unix2dos` must be installed on your local system, see Source.

#### Usage

`lineEnding(infile, pattern = NULL, outfile = NULL, to = c("dos", "unix"), ...)`

#### Arguments

- `infile` Input filename(s).
- `pattern` See `list.files`. This will be ignored if `infile` is specified.
- `outfile` Output filename. If not supplied, `infile` will be overwritten.
- `to` Either `"dos"` or `"unix"`.
- `...` Additional arguments passed to `list.files`, only applicable if `infile` is not specified.

#### Author(s)

Florian Detsch

#### Source

Dos2Unix/Unix2Dos Text file format converters

#### See Also

`list.files`, `system`
Examples

```r
## input file
infile <- paste(system.file(package = "Orcs"), "DESCRIPTION", sep = "/")

## convert to dos line endings and write to output file
ofl = file.path(tempdir(), "DESCRIPTION4wd")
lineEnding(infile, outfile = ofl, to = "dos")
```

---

**list2df**

*Create data.frame from list*

Description

Create a data.frame from a list directly, *i.e.* without being required to explicitly call `rbind` first.

Usage

```r
list2df(x, bind = c("rows", "cols"), ...)
```

Arguments

- `x` A list object.
- `bind` Binding direction. Available options are "rows" (default) and "cols" for `rbind` and `cbind`, respectively.
- `...` Additional arguments passed to `data.frame`.

Value

A data.frame object.

See Also

`data.frame`, `rbind`, `cbind`.

Examples

```r
lst <- list(letters[1:3], letters[4:6], letters[7:9])
do.call("rbind", lst) # results in matrix
list2df(lst) # results in data.frame created using rbind()
list2df(lst, bind = "cols") # same for cbind()
```
loadFromGit  

Install and load a package from GitHub

Description
This function comprises multiple steps required to install and load a package directly from GitHub.

Usage
loadFromGit(repo = "fdetsch/Orcs", ...)

Arguments
- repo: Repository address as character, defaults to "fdetsch/Orcs".
- ...: Additional arguments passed to install_github.

Author(s)
Florian Detsch

See Also
install_github

Examples
## Not run:
## install 'Orcs' development version from GitHub
loadFromGit("fdetsch/Orcs", ref = "develop")

## End(Not run)

loadPkgs  

Load multiple packages

Description
Load and attach multiple packages at once.

Usage
loadPkgs(pkgs, ...)

Arguments

pkgs Packages to load as character.
...

Additional arguments passed to `library`, except for 'character.only' which is set to TRUE.

Note

Package startup messages are automatically disabled.

Author(s)

Florian Detsch

See Also

`library`.

Examples

```r
loadPkgs(c("raster", "rgdal"))
```

---

meanDifference Calculate mean difference between two datasets

Description

Calculate the mean difference between two datasets as suggested by Wang et al. (2012).

Usage

```r
## S4 method for signature 'RasterLayer'
meanDifference(x, y)

## S4 method for signature 'numeric'
meanDifference(x, y)
```

Arguments

x, y Objects of class RasterLayer or numeric.

Value

The mean difference between the two inputs either as RasterLayer or numeric.
**Source**


**Examples**

```r
x <- 1:10
y <- 2:11
meanDifference(x, y)
```

---

**merge**

**Merge Objects Stored in a List**

**Description**

Complementing existing merge methods, e.g. `merge` for `Raster*` objects, which typically work with one or two inputs only, this function accepts a list of objects that are to be merged together.

**Usage**

```r
## S4 method for signature 'list,missing'
merge(x, by = 1L, all = TRUE, ...)
```

**Arguments**

- **x**: A list of objects of the same type (e.g. `Raster*` or `data.frame`).
- **by, all**: See `merge.data.frame`. Ignored if data stored in 'x' is not of class `data.frame`.
- **...**: Additional arguments passed to the underlying merge method (e.g. arguments compatible with `merge` and `writeRaster` for `Raster*` input). Ignored if data stored in 'x' is of class `data.frame`.

**Value**

A merged object (e.g. a new `Raster*` object with a larger spatial extent).

**Author(s)**

Florian Detsch

**See Also**

`merge.data.frame`, `do.call`, `Reduce`. 
Examples

```r
## Raster* input
dms = list.files(system.file("extdata", package = "Orcs"), pattern = "ASTGTM2.*dem.tif$", full.names = TRUE)
dms = lapply(dms, raster)
dem = merge(dms, tolerance = 1e4)
plot(dem)

## data.frame input
mrg = merge(list(iris, iris, iris), by = c("Species", "Sepal.Length", "Petal.Width"))
head(mrg)
```

---

multiKnit

Convert multiple R Markdown files to ordinary Markdown

### Description

This function is a convenient wrapper around `knit` as it automatically converts multiple R Markdown files (.Rmd) located in a specified folder (and, optionally, matching a particular pattern) to standard Markdown (.md).

### Usage

```r
multiKnit(path_in = ".", path_out = path_in, pattern = "*.Rmd$", ...)
```

### Arguments

- `path_in` Input file path as character, defaults to the current working directory.
- `path_out` Output file path as character, defaults to `path_in`.
- `pattern` Passed to `list.files`, defaults to "*.Rmd$".
- `...` Additional arguments passed to `knit`.

### Value

Output filenames as character.

### Author(s)

Florian Detsch

### See Also

`knit`
offsetGridText

Insert offset text annotation into 'trellis' plot

Description

This is a wrapper function around Orcs:::calcOffsetGridText and grid-based text drawing functions (currently including grid.text and grid.stext) that automatically adds offset text annotations to a 'trellis' plot.

Usage

offsetGridText(
  x,
  y = NULL,
  labels,
  xlim = NULL,
  ylim = NULL,
  pos = NULL,
  stext = FALSE,
  offset = 0.02,
  ...
)

Arguments

x
A numeric vector containing x coordinates, or a 2-column matrix containing x and y coordinates.

y
A numeric vector containing y coordinates, or NULL if 'x' is a two-column matrix.

labels
The text to be written as character.

xlim, ylim
X and Y-axis limits (c(min, max)) of the current plot. If not supplied, limits are automatically calculated from supplied x and y coordinates.

pos
Text position specifier(s) as integer used by text. If not supplied, optimal text positions will be determined with respect to neighboring locations using thigmophobe.

stext
Logical, defaults to FALSE. If TRUE, shadow text will be drawn around 'labels'.

offset
A numeric offset in normalized parent coordinates ("npc", see unit).

... Additional arguments passed to the respective grid text drawing function (depends on 'stext').

Author(s)

Florian Detsch
See Also
grid.text, grid.stext, thigmophobe, Orcs::calcOffsetGridText.

Examples

```r
stopifnot(
  require(sf),
  require(latticeExtra),
  require(grid)
)

# kilimanjaro peaks
peaks = data.frame(Peak = c("Kibo", "Mawenzi", "Shira"),
                   Lon = c(37.359031, 37.455061, 37.210408),
                   Lat = c(-3.065053, -3.095436, -3.038222))
coordinates(peaks) = ~ Lon + Lat
proj4string(peaks) = "+init=epsg:4326"

# visualization
xlim_kili <- c(37.15, 37.55)
ylim_kili <- c(-3.25, -2.9)

p = spplot(KiLi[[1]], col.regions = "transparent", colorkey = FALSE,
xlim = xlim_kili, ylim = ylim_kili,
scales = list(draw = TRUE, y = list(rot = 90)),
sp.layout = rgb2spLayout(KiLi, quantiles = c(0, 1), alpha = .8)) +
  layer(sp.points(peaks, cex = 1.5, pch = 20, col = "black"))

print(p)

downViewport(trellis.vpname(name = "figure"))
offsetGridText(x = coordinates(peaks), labels = peaks$Peak,
xlim = xlim_kili, ylim = ylim_kili, stext = TRUE, offset = .02,
gp = gpar(fontsize = 16))
```

<table>
<thead>
<tr>
<th>OrcsCppFun</th>
<th>Dimensions of a data.frame</th>
</tr>
</thead>
</table>

Description
Similar to base-R nrow, ncol and dim, this set of functions let’s you retrieve the number of rows and columns of a data.frame.

Usage

nrowC(x)
Arguments

x  A data.frame.

Value

dimC returns an 'integer' vector of length 2 (number of rows and columns); nrowC (or ncolC) returns
the number of rows (or columns) as a single 'integer'.

Functions

• nrowC:
• ncolC:
• dimC:

Author(s)

Florian Detsch

See Also

nrow, ncol, dim.

Examples

dat <- data.frame(a = 1:4, b = 2:5, c = 3:6)

nrowC(dat)

par7zip

Parallelized 7-zip compression

Description

By calling the Unix terminal or Windows command prompt, this function performs parallelized
7-zip compression of selected files based on the built-in parallel package.

Usage

par7zip(outfile, nodes = 1L, ...)

pureBasename

Arguments

outfile  Target file for compression as character. A file extension compatible with 7-
zip needs to be included, see Supported formats. If missing, this defaults to the
found input file names with a .7z extension attached.

nodes  Number of cores to use for parallelization as integer, defaults to 1.

...  Additional arguments passed to list.files.

Value

Output filename(s) as character.

Author(s)

Florian Detsch

See Also

list.files, system

pureBasename  Return filename without extension

Description

As opposed to basename, this function returns the pure basename of one or multiple file names, i.e.
without extension.

Usage

pureBasename(path, slash = FALSE)

Arguments

path  File name(s) as character.

slash  A logical determining whether to add a leading slash ("/"), to the returned file
name.

Value

File name(s) without extension as character.

Author(s)

Florian Detsch

See Also

basename, file_path_sans_ext.
Examples

```r
fls <- system.file("external/rlogo.grd", package = "raster")
pureBasename(fls)
pureBasename(fls, slash = TRUE)
```

---

**pvalue**  
*Get p-value from `lm` object*

### Description

Retrieve the *p*-value associated with a univariate linear regression.

### Usage

```r
pvalue(mod)
```

### Arguments

- `mod`  
  An object of class `lm`.

### Value

A numeric *p*-value.

### Source

*retrieving p-values in lm* on R-help mailing list (last accessed 2018-06-02).

### See Also

- `lm`

### Examples

```r
## taken from ?lm
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
pvalue(lm.D9)
```
rgb2spLayout

Convert an RGB RasterBrick/Stack to use with spplot

Description
This function takes a red-green-blue Raster* object and produces a list with color information that can be passed on to the 'sp.layout' argument from spplot.

Usage
rgb2spLayout(x, quantiles = c(0.02, 0.98), alpha = 1)

Arguments
- x: A 3-layered Raster* object.
- quantiles: Upper and lower quantiles used for color stretching.
- alpha: Level of transparency.

Author(s)
Tim Appelhans, Florian Detsch

See Also
plotRGB

Examples
library(raster)
library(sp)

b <- brick(system.file("external/rlogo.grd", package="raster"))

## using plotRGB
plotRGB(b)

## convert brick to list
lout <- rgb2spLayout(b)
lout_alph <- rgb2spLayout(b, alpha = 0.5)

## create random spatial points for plotting
df <- data.frame(dat = rnorm(100, 2, 1),
                 x = rnorm(100, 50, 20),
                 y = rnorm(100, 50, 25))
coordinates(df) <- ~x+y

## plot spatial points with rgb background
spplot(df, sp.layout = lout)
spplot(df, sp.layout = lout_alph)
### rmDuplCols

Remove duplicated columns from data.frame

---

**Description**

Automatically detect and remove columns from a data.frame based on duplicated headers.

**Usage**

\[
\text{rmDuplCols}(x, \text{keep\_first} = \text{TRUE}, \ldots)
\]

**Arguments**

- **x**: Input data.frame.
- **keep\_first**: A logical determining whether the first column of an otherwise duplicated header should be kept, defaults to \text{TRUE}.
- \ldots: Currently not in use.

**Value**

Revised data.frame.

**Author(s)**

Florian Detsch

**See Also**

duplicated

**Examples**

```r
## sample data
set.seed(123)
dat <- data.frame(matrix(rnorm(28), nc = 7))
names(dat) <- c("Col1", "Col1", "Col1", "Col2", "Col3", "Col3", "Col4")

dat
rmDuplCols(dat)
rmDuplCols(dat, keep\_first = FALSE)
```
setwdOS

Set working directory dependent on current OS

Description

Similar to `setwd`, this function sets the working directory to a user-defined path. Rather than supplying a single 'dir' argument, however, both an OS-sensitive path to the desired hard disk partition and, optionally, an extension of this file path are required.

Usage

```
setwdOS(lin = "/media/permanent/", win = "C:/", ext = NULL)
```

Arguments

- `lin`, `win`  
  Absolute file paths to the Linux and Windows partition as character.
- `ext`  
  Optional file path extension as character that will be added to 'lin' or 'win' after automatic OS determination.

Author(s)

Florian Detsch

See Also

`setwd`, `switch`

Examples

```r
## Not run:
# desired partition
setwdOS()

# including file path extension
setwdOS(ext = "kilimanjaro/nubiscope")

## End(Not run)
```
stextGrob

Draw Shadow Text

Description

Create and draw shadow text by wrapping a textual expression into a colored framing.

Usage

stextGrob(
  label,
  x = grid::unit(0.5, "npc"),
  y = grid::unit(0.5, "npc"),
  col = "white",
  fill = "black",
  r = 0.1,
  gp = grid::gpar(),
  vp = NULL,
  name = NULL,
  ...
)

Arguments

label A character or expression vector, see textGrob.

x, y Horizontal and vertical text position as unit objects passed to grid.text.

col, fill Framing and fill color passed to gpar.

r Blur radius of colored framing as numeric.

name, gp, vp Graphical parameters passed to gTree.

Additional arguments passed to grid.text.

Value

A text grob created by gTree.

Author(s)

Baptiste Auguie, Florian Detsch

See Also

grid.text.
Examples

```r
library(grid)
grid.newpage()
grid.rect(gp = gpar(fill = "grey"))
grid.text("test")
```

`substrC`  
**Substrings of a Character Vector (C++ Style)**

Description

Extract substrings from a character vector in C++.

Usage

```r
substrC(x, pos, len)
```

Arguments

- `x` A character vector.
- `pos` The start point of the substring as integer. Position indications start from 1, which is the default in R.
- `len` The length of the substring as integer.

Value

A character vector of the same length as 'x'.

See Also


Examples

```r
substrC("Hello, world!", pos = 1, len = 5)
```
trimImages

Remove whitespace from images

Description
This is a wrapper function around `convert -trim` to automatically remove any whitespace from locally saved images. Note that 'ImageMagick' must be installed on your local system, see Source.

Usage
trimImages(path = ".", pattern = c(".png$", ".tiff$"))

Arguments
- **path**: File path leading to image files as character, defaults to the current working directory.
- **pattern**: Character. A regular expression as character accepted by `list.files`, defaults to c(".png$", ".tiff$").

Value
A character vector containing the names of the processed images.

Author(s)
Florian Detsch

Source

See Also
- `list.files`, `system`

Examples
```r
## Not run:
## trim image of bart simpson
download.file("http://pngimg.com/uploads/simpsons/simpsons_PNG93.png?i=1"
  , destfile = (ofl <- file.path(tempdir(), "bart.png", fsep = "\""))
  , mode = "wb")

par(mfrow = c(1, 2))

img = brick(ofl)
plotRGB(img)

jnk = trimImages(tempdir(), "bart.png")
```
## unlistStrsplit

Unlist the outcome of strsplit

### Description

Per default, `strsplit` returns a list, with each entry holding the vector of splits of the initial string(s). This function is a simple wrapper that casts `unlist` upon the returned list to produce a concatenated character vector consisting of the single split elements.

### Usage

```r
unlistStrsplit(x, split, ...)```

### Arguments

- `x` A character vector with elements to be split.
- `split` A character vector used for splitting, see `strsplit`.
- `...` Additional arguments passed to `strsplit`.

### Author(s)

Florian Detsch

### See Also

`strsplit`

### Examples

```r
## 1st example
x <- "This is a test."
unlistStrsplit(x, " ")

## 2nd example; note that 'split' defaults to 'whitespace'
x2 <- "This is a 2nd test."
unlistStrsplit(c(x, x2))
```
unsortedFactor

Factor with unsorted levels

Description

Casting factor upon a (character) vector usually results in alphabetically ordered factor levels. Although this seems reasonable in most cases, the automated ordering of factor levels is seldomly desirable in the context of visualization, e.g. when working with tiled lattice or ggplot2 figures. This function returns a factor with levels ordered according to their first appearance in the supplied vector.

Usage

unsortedFactor(x, ...)

Arguments

x A character vector with elements to converted to factor.

... Additional arguments passed to factor.

Author(s)

Florian Detsch

See Also

factor

Examples

mnth <- month.abb

## factor levels are being sorted
fc_mnth <- factor(mnth)
levels(fc_mnth)

## factor levels remain unsorted
fc_mnth2 <- unsortedFactor(mnth)
levels(fc_mnth2)
Index

+Topic **package**
  Orcs-package, 3

as.layer, 14
assignSSH, 3
basename, 24
brick, 11
buildBook, 4
bumpVersion, 5
c.trellis, 13
cbind, 16
compareDistributions, 6
coords2Lines, 7
coords2Lines, Line-method (coords2Lines), 7
coords2Lines, matrix-method (coords2Lines), 7
coords2Polygons, 8
coords2Polygons, matrix-method (coords2Polygons), 8
coords2Polygons, Polygon-method (coords2Polygons), 8
data.frame, 16
density, 6
dim, 22, 23
dimC (OrcsCppFun), 22
do.call, 11, 19
duplicated, 27
evalMetrics, 9
expression, 29
ext2spy, 10
extent, 10
factor, 33
file.exists, 11
file_path_sans_ext, 24
gpar, 29
grid.stext, 21, 22
grid.stext (stextGrob), 29
ggrid.text, 21, 22, 29
gTree, 29
ifMissing, 11
install_github, 17
KiLi, 12
knit, 20
lapply, 12, 14
latticeCombineGrid, 12
latticeCombineLayer, 14
library, 18
lineEnding, 15
Lines, 7
list.files, 15, 20, 24, 31
list2df, 16
lm, 25
loadFromGit, 17
loadPkgs, 17
meanDifference, 18
meanDifference, numeric-method (meanDifference), 18
meanDifference, RasterLayer-method (meanDifference), 18
merge, 19, 19
merge, list, missing-method (merge), 19
merge.data.frame, 19
multiKnit, 20
ncol, 22, 23
ncolC (OrcsCppFun), 22
nrow, 22, 23
nrowC (OrcsCppFun), 22
offsetGridText, 21
Orcs-package, 3
INDEX

OrcsCppFun, 22
orcspackage (Orcs-package), 3

par7zip, 23
plot, 6
plotRGB, 26
Polygon, 8
Polygons, 8
proj4string, 10
pureBasename, 24
pvalue, 25

rbind, 16
Reduce, 19
render_book, 4
rgb2spLayout, 26
rmDuplicatedCols, 27

setwd, 28
setwdOS, 28
SpatialLines, 7
SpatialLinesDataFrame, 7, 8
SpatialPolygons, 8
SpatialPolygonsDataFrame, 8, 9
spplot, 26
stextGrob, 29
strsplit, 32
substr, 30
substrC, 30
switch, 28
system, 15, 24, 31

text, 21
textGrob, 29
thigmophobe, 21, 22
trimImages, 31

unit, 21, 29
unlist, 32
unlistStrsplit, 32
unsortedFactor, 33

write.table, 11
writeRaster, 11, 19

xyplot, 6