Package 'yogitools'

February 6, 2018

Title Yogi Tools

Version 0.0.0.9000

Description Yogi Tools is a diverse collection of useful tools for R.

Depends R (>= 3.2.3)

License GPL

Encoding UTF-8

LazyData true

Suggests testthat

RoxygenNote 6.0.1

R topics documented:

as.df *Convert list of lists into data.frame*

Description

The list of lists should only contain lists with the same element names.

Usage

```
as.df(x)
```

Arguments

x	the list of lists

Value

a `data.frame`

Examples

```
x <- as.df(list(list(a=1,b="foo"),list(a=2,b="bar")))
```

c2q *Convert from Raw coordinate to Quadrant address*

Description

Converts address tags for 384-well plates from the raw coordinate system (e.g. B15) to the quadrant system (e.g. C_A08).

Usage

```
c2q(x)
```

Arguments

x	a raw coordinate (e.g. B15) (do not directly use on vectors!)

Value

the quadrant plate coordinate

Examples

```
c2q("B15")
```

colAlpha *Add alpha channel*

Description

Adds an alpha channel (i.e. transparency) to a predefined color

Usage

```
colAlpha(color, alpha)
```

Arguments

color	a predefined color string (e.g. "firebrick")
alpha	a number between 0 and 1 for the alpha channel value

Examples

```
transparentChartreuse <- colAlpha("chartreuse3",0.3)
```

combo *Set of subsets*

Description

Generates the set of all possible subsets for a given list or vector

Usage

```
combo(l)
```

Arguments

l	a list or vector

Value

a list of lists containing all possible subsets of the input

Examples

```
combo(1:4)
```

extract.groups *Extract regex groups (local)*

Description

Locally excise regular expression groups from string vectors. I.e. only extract the first occurrence of each group within each string.

Usage

```
extract.groups(x, re)
```

Arguments

x	A vector of strings from which to extract the groups.
re	The regular expression defining the groups

Value

A `matrix` containing the group contents, with one row for each element of x and one column for each group.

fin *Remove infinite and NA values*

Description

Removes infinite and NA values from vectors, lists, matrices and data.frames.

Usage

```
fin(x)
```

Arguments

x	the object to which the function is applied

Details

Warning: If the given object is matrix or data.frame, any row containing infinite or NA values is removed entirely. All columns must be numeric.

Value

the same object, with NAs and infinite values removed

Examples

```
fin(c(1,2,NA,3))
fin(data.frame(a=c(1,2,NA,3),b=c(4,5,6,7)))
```

`getArg` *Get CLI argument*

Description

Retrieves a user-supplied argument command-line argument with a given name. Argument syntax: name=value

Usage

```
getArg(name, default = NULL, required = FALSE)
```

Arguments

`name`	The name of the command line argument
`default`	If no value was given by the user, default to this
`required`	Whether the argument is required or not. If TRUE, an error is raised when no value was provided.

Examples

```
## Not run:
infile <- getArg("inputFile",required=TRUE)
userIQ <- getArg("userIQ",default=0)

## End(Not run)
```

`global.extract.groups` *Extract regex groups (global)*

Description

Globally excise regular expression groups from string vectors. I.e. only extract the all occurrences of each group within each string.

Usage

```
global.extract.groups(x, re)
```

Arguments

x	A vector of strings from which to extract the groups.
re	The regular expression defining the groups

Value

A list of matrix's containing the group contents, with one list item for every element of x, and with each matrix containing one column for each group and one row for each occurrence of the pattern.

---

ith.rank *Get i'th rank from list*

Description

Retieve the i'th ranked item from a numerical vector

Usage

```
ith.rank(values, i, high = TRUE)
```

Arguments

values	a numerical vector
i	the rank
high	whether to rank by highest or lowest values.

Value

the ith ranked value

Examples

```
vals <- rnorm(100,0,1)
ith.rank(vals,4)
```

mcc *Matthew's correlation coefficient (MCC)*

Description

Calculate Matthew's correlation coeffient (MCC). See https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

Usage

```
mcc(t, scores, truth)
```

Arguments

t	the score threshold
scores	vector of scores for each measured item
truth	logical vector classifying each item as a member of the hidden true or false classes

Value

a vector listing the MCC value, the precision, and the recall

Examples

```
patientHasDisease <- sample(c(TRUE,FALSE),100,replace=TRUE)
patientDiganosticScore <- sapply(patientHasDisease,
   function(d) if (d) rnorm(1,20,3) else rnorm(1,18,3)
)
mccval <- mcc(21,patientDiganosticScore,patientHasDisease)
```

new.cluster.map *Cluster mapper*

Description

Constructor for an object supporting simple connected-component-clustering

Usage

```
new.cluster.map(n)
```

Arguments

n	The number of elements to cluster.

Details

The process starts with n objects, each in their own cluster. Whenever a link is between two objects is reported, their clusters are merged. Contains the following functions: \beginitemize \item `addLink(i,j)`: Creates a new link between items i and j. Whenever a link is created, the clusters encompassing the two objects are merged. \item `getClusters()`: Returns a list of lists representing the clusters \item `getIdxOf(i)`: Returns the cluster index of a given object. \enditemize

Value

the mapper object

Examples

```
cmap <- new.cluster.map(10)
cmap$addLink(1,5)
cmap$addLink(3,5)
cmap$addLink(1,5)
cmap$getClusters()
```

`new.counter` *Create new Counter*

Description

This constructor method creates an object that can count occurrences of different items. It allows importing and exporting of the counter status in string form.

Usage

```
new.counter()
```

Details

The object has the following methods: \beginitemize \item `inc(id)`: Increase the counter for item with `id` by 1. \item `add(id,x)`: Add `x` occurrences for the item with `id`. \item `get(id)`: Get the number of occurrences seen for item `id`. \item `ls(id)`: List all counts for all items by id. \item `export(id)`: Exports the counter state to a string that can be saved or logged. \item `import.add(str)` Imports a previous counter state from the string `str` and adds it to the current counts. \enditemize

Value

An object of type `yogicounter`.

Examples

```
cn <- new.counter()
cn$inc("foo")
cn$inc("bar")
cn$add("foo",6)
cn$get("foo")
# 7
cn$ls()
# foo 7
# bar 1
cn$export()
# foo=7,bar=1
```

---

q2c *Convert from Quadrant to Coordinate adress*

Description

Converts address tags for 384-well plates from the quadrant system (e.g. C_A08) to the raw coordinate system (e.g. B15).

Usage

```
q2c(x)
```

Arguments

x	a quadrant coordinate (e.g. C_A08) (do not directly use on vectors!)

Value

the raw plate coordinate

Examples

```
q2c("C_A08")
```

to.df *Convert row-bound lists to data.frame*

Description

Running `rbind` on lists with the same element names yields a datastructure very similar to a `data.frame`, but does not provide the same full functionality. This function converts such objects to a real dataframe.

Usage

```
to.df(x)
```

Arguments

`x`	the result of the rbind call.

Value

a `data.frame`

Examples

```
x <- rbind(list(a=1,b="foo"),list(a=2,b="bar"))
y <- to.df(x)
```

zbind *3D-bind matrices*

Description

Binds matrices of same size together to a 3D array, analogously to cbind and rbind.

Usage

```
zbind(...)
```

Arguments

`...`	Any number of matrices of the same size

Value

A 3D array of the bound matrices

Index