Package 'lfactors'

October 13, 2022

Version 1.0.4

Date 2018-03-16

Title Factors with Levels

Author Paul Bailey [aut, cre]

Maintainer Paul Bailey <pbailey@air.org>

Depends R (>= 3.1.0)

Imports methods, stats, utils

Suggests testthat, Matrix

Description Provides an extension to factors called 'lfactor' that are similar to factors but allows users to refer to 'lfactor' levels by either the level or the label.

License GPL-2

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-03-16 21:41:08 UTC

R topics documented:

lfactors-package *Allow a user to use == and %in% on an lfactor for both the label and the level*

Description

Similar to a factor, the user defines levels and labels when creating an `lfactor`. This is especially useful when labels are long, and users know the levels well.

Details

See `lfactor` for examples.

as.factor.lfactor *factor from an lfactor*

Description

Returns a `factor` from an `lfactor`.

Usage

```
## S3 method for class 'lfactor'
as.factor(x)
```

Arguments

`x`	the lfactor to be coerced to a factor

Details

Simply drops the numeric levels from the lfactor and returns a normal factor.

See Also

`as.factor`

`as.integer.lfactor` *Integer Vectors from lfactors*

Description

Returns integer representation of an lfactor that ignores the values used in the `levels` argument when the lfactor was created and instead returns an integer representation starting with 1.

Usage

```
## S3 method for class 'lfactor'
as.integer(x, ...)
```

Arguments

`x`	same as `as.integer`
`...`	not used

Details

This method does not return integer results that are otherwise equal to the results from as.numeric for compatibility with `sparse.model.matrix`.

See Also

`as.integer`, `as.numeric.lfactor`

Examples

```
require(lfactors)
# create an example
let <- lfactor(4:12,
               levels=4:12,
               labels=letters[4:12])

as.numeric(let)
#same as as.numeric(4:12)
as.integer(let)
#same as 1:9
```

`as.numeric.lfactor` *Numeric Vectors from lfactors*

Description

Returns numeric representation of an lfactor equal to the `levels` argument for each value. This is different from the behavior of factor which would ignore the values of `level`.

Usage

```
## S3 method for class 'lfactor'
as.numeric(x, ...)
```

Arguments

`x`	same as `as.numeric`
`...`	not used

Details

This method does not return floating point (numeric) results that are otherwise equal to the results from `as.integer.lfactor`. Instead it returns the value of the level that was input when the lfactor was created.

See Also

`as.numeric`, `as.integer.lfactor`

Examples

```
require(lfactors)
# create an example
let <- lfactor(4:12,
               levels=4:12,
               labels=letters[4:12])

as.numeric(let)
#same as as.numeric(4:12)
as.integer(let)
#same as 1:9
```

inlf *Implements %in% for lfactors*

Description

Implements %in% for lfactors.

Usage

```
inlf(x, table)
```

Arguments

x	same as %in%
table	same as %in%

See Also

%in%

lfactor *lfactors*

Description

lfactor creates a factor that can be compared to its levels or labels.

Usage

```
lfactor(x, levels, labels = levels, ...)
```

Arguments

x	a numeric or character vector of data. Levels of x can be taken either from levels or labels.
levels	a numeric vector of levels in x. Note that, unlike factor, these must be numeric.
labels	a vector of labels for the levels. This vector must be either characters that cannot be cast as numeric or characters that are equal to the level, of the same index, when cast as numeric.
...	arguments passed to factor

Details

An lfactor can be compared to the levels or the labels (see the Examples). Because of that, the levels must be numeric, and the labels must be either not castable as numeric or equal to the levels of the same index when cast as numeric.

An lfactor is, essentially, a factor that remembers the levels as well as the labels argument. Note that all of the arguments are passed to `factor`. Because lfactor imposes some additional constraints on the types of levels and labels and stores additional information, an lfactor uses more memory than a factor—because it stores both labels and levels—and is, in some ways, more limited than a factor.

Value

An object of class lfactor that also implements `factor`

See Also

`factor`

Examples

```
require(lfactors)
# make an example lfactor object
mon <- lfactor(1:12,
               levels=1:12,
               labels=c("Jan", "Feb", "Mar", "Apr", "May","Jun",
                        "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))
# print out the lfactor
mon
# compare to label
mon == "Feb"
# Compare to level
mon == 2
# Show that the == works correctly
all.equal(mon == "Feb", mon == 2)
# Show that the != works correctly
all.equal(mon != "Feb", mon != 2)
# also works when the vector is not the lfactor
all.equal(mon[3] == c("Jan", "Feb", "Mar"), mon[3] == 1:3)

# or when both the lfactor and the object being compare to are vectors
all.equal(mon[1:2] == c("Feb", "Tuesday"), mon[1:2] == c(2,-4) )

# similar to Ops.factor, this gives a helpful warning and NA results
mon >= "Jan"

# %in% works correctly
all.equal(mon %in% c(2, 3), mon %in% c("Feb", "Mar"))
# and when the lfactor is on the right
all.equal(c(-4, 14,3,10) %in% mon, c("not a month", "Third December","Mar","Oct") %in% mon)
# and when both left and right are lfactors
all.equal(mon %in% mon, rep(TRUE,12))
```

llevels *Numeric Levels of an lfactor*

Description

`llevels` gives the numeric levels of an lfactor.

Usage

```
llevels(x)
```

Arguments

`x`	object of class lfactor

Value

A vector of levels

See Also

`levels`

mlfactor `match` *Function for lfactors*

Description

`match` function for lfactors.

Usage

```
mlfactor(x, table, nomatch = NA_integer_, incomparables = NULL)
```

Arguments

`x`	same as `match`
`table`	same as `match`
`nomatch`	same as `match`
`incomparables`	same as `match`

Details

Allows `match` to work when the `x` or `table` arguments in a call to match are lfactors.

See Also

`match`

Index