

Package ‘inum’

April 12, 2021

Title Interval and Enum-Type Representation of Vectors

Date 2021-04-12

Version 1.0-4

Description Enum-type representation of vectors and representation of intervals, including a method of coercing variables in data frames.

Depends R (>= 3.3.0)

Imports stats, libcoin (>= 1.0-0)

License GPL-2

NeedsCompilation no

Author Torsten Hothorn [aut, cre]

Maintainer Torsten Hothorn <Torsten.Hothorn@R-project.org>

Repository CRAN

Date/Publication 2021-04-12 15:20:02 UTC

R topics documented:

enum	1
interval	2
inum	3

Index	5
--------------	----------

enum *Enumeration-type Representation of Vectors*

Description

Elements of a vector are stored as a set of levels and an integer representing the enumeration.

Usage

enum(x)

Arguments

`x` A vector. Currently, methods for factors, logicals, integers, and numeric vectors are implemented.

Details

The unique elements of `x` are stored as a `levels` attribute to an integer representing the enumeration. `levels` and `nlevels` methods are available. This is essentially the same as `factor` where the levels can be arbitrary vectors, not just characters.

Value

An object of class `enum`. A value of `0` encodes NA.

See Also

[factor](#)

Examples

```
(ex <- enum(x <- gl(2, 2)))
all.equal(levels(ex)[ex], x)

(ex <- enum(x <- rep(c(TRUE, FALSE), 2)))
all.equal(levels(ex)[ex], x)

(ex <- enum(x <- rep(1:5, 2)))
all.equal(levels(ex)[ex], x)

(ex <- enum(x <- rep(1:5 + .5, 2)))
all.equal(levels(ex)[ex], x)

(ex <- enum(x <- c(NA, rep(1:5 + .5, 2))))
all.equal(c(NA, levels(ex))[unclass(ex) + 1L], x)
```

interval

Cut Numeric Vectors into Intervals

Description

`interval` divides `x` into intervals and, unlike `cut`, represents these as a numeric vector.

Usage

```
interval(x, ...)
## S3 method for class 'numeric'
interval(x, breaks = 50, ...)
```

Arguments

x	A numeric vector.
breaks	Either a numeric vector of two or more unique cut points or a single number (greater than or equal to 2) giving the number of intervals into which x is to be cut by cut.
...	Additional arguments, currently ignored.

Details

This is just a wrapper around cut where the resulting intervals are stored as numeric values for simplified computation.

Value

An object of class interval. A value of 0 encodes NA.

See Also

[cut](#)

Examples

```
(ix <- interval(x <- 0:100/100, breaks = 0:10/10))
(cx <- cut(x, breaks = 0:10/10))

attr(ix, "levels")
levels(ix)
levels(cx)

diag(table(ix, cx))

(ix <- interval(x <- c(NA, 0:100/100), breaks = 0:10/10))
ix[is.na(x)]
unclass(ix)[is.na(x)]
```

Description

Represents elements of a data frame as enum or interval.

Usage

```
inum(object, nmax = 20, ...)
## S3 method for class 'data.frame'
inum(object, nmax = 20, ignore = NULL,
      total = FALSE, weights = NULL, as.interval = "",
      complete.cases.only = FALSE, meanlevels = FALSE, ...)
```

Arguments

<code>object</code>	A data frame.
<code>nmax</code>	Maximal number of categories for each of the numeric variables.
<code>ignore</code>	A character vector of variable names not to be discretised.
<code>total</code>	A logical. TRUE means that a condensed data frame of all variables is returned, FALSE a list of discretised variables.
<code>weights</code>	An optional vector of weights.
<code>as.interval</code>	A character vector of variable names to be converted to <code>interval</code> instead of <code>enum</code> .
<code>complete.cases.only</code>	A logical. TRUE removes all rows with missing values.
<code>meanlevels</code>	A logical. TRUE, the level is the mean of the observations in the corresponding bin. The default FALSE uses the largest observation in the bin.
<code>...</code>	Additional arguments, currently ignored.

Details

Each variable in `object` is converted to `enum` or `interval`.

Value

An object of class `inum`, basically a list of `enum` or `interval` objects. If `total = TRUE`, an integer vector with a data frame as `levels` attribute is returned. In this case, `0` means NA.

Examples

```
data("iris", package = "datasets")
iris[1,1] <- NA
inum(iris, nmax = 5)
inum(iris, nmax = 5, total = TRUE)
inum(iris, nmax = 5, total = TRUE, as.interval = "Sepal.Width",
     complete.cases.only = TRUE)
```

Index

* data

- enum, 1
- interval, 2
- inum, 3

cut, 3

enum, 1, 4

factor, 2

interval, 2, 4

inum, 3