Package ‘xefun’

August 10, 2023

Version 0.1.5

Title X-Engineering or Supporting Functions

Description Miscellaneous functions used for x-engineering (feature engineering) or for supporting in other packages maintained by 'Shichen Xie'.

Imports data.table

License MIT + file LICENSE

URL https://github.com/ShichenXie/xefun

BugReports https://github.com/ShichenXie/xefun/issues

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Shichen Xie [aut, cre]

Maintainer Shichen Xie <xie@shichen.name>

Repository CRAN

Date/Publication 2023-08-10 13:10:03 UTC

R topics documented:

as.list2 *vector to list*

Description

Converting a vector to a list with names specified.

Usage

```
as.list2(x, name = TRUE, ...)
```

Arguments

`x`	a vector.
`name`	specify the names of list. Setting the names of list as x by default.
`...`	Additional parameters provided in the as.list function.

Examples

```
as.list2(c('a', 'b'))

as.list2(c('a', 'b'), name = FALSE)

as.list2(c('a', 'b'), name = c('c', 'd'))
```

ceiling2 *rounding of numbers*

Description

The ceiling2 is ceiling of numeric values by digits. The floor2 is floor of numeric values by digits.

Usage

```
ceiling2(x, digits = 1)

floor2(x, digits = 1)
```

Arguments

`x`	a numeric vector.
`digits`	integer indicating the number of significant digits.

Value

ceiling2 rounds the elements in x to the specified number of significant digits that is the smallest number not less than the corresponding elements.

floor2 rounds the elements in x to the specified number of significant digits that is the largest number not greater than the corresponding elements.

Examples

```
x = c(12345, 54.321)

ceiling2(x)
ceiling2(x, 2)
ceiling2(x, 3)

floor2(x)
floor2(x, 2)
floor2(x, 3)
```

---

cols_const *constant columns*

Description

The columns name of a data frame with constant value.

Usage

```
cols_const(dt)
```

Arguments

dt	a data frame.

Examples

```
dt = data.frame(a = sample(0:9, 6), b = sample(letters, 6),
                c = rep(1, 6), d = rep('a', 6))
dt
cols_const(dt)
```

`cols_type` *columns by type*

Description

The columns name of a data frame by given data types.

Usage

```
cols_type(dt, type)
```

Arguments

`dt` a data frame.

`type` a string of data types, available values including character, numeric, double, integer, logical, factor, datetime.

Examples

```
dt = data.frame(a = sample(0:9, 6), b = sample(letters, 6),
                c = Sys.Date()-1:6, d = Sys.time() - 1:6)
dt
# numeric columns
cols_type(dt, 'numeric')
# or
cols_type(dt, 'n')

# numeric and character columns
cols_type(dt, c('character', 'numeric'))
# or
cols_type(dt, c('c', 'n'))

# date time columns
cols_type(dt, 'datetime')
```

`conticnt` *continuous counting*

Description

It counts the number of continuous identical values.

Usage

```
conticnt(x, cnt = FALSE, ...)
```

Arguments

x a vector or data frame.

cnt whether to count the number rows in each continuous groups.

... ignored

Value

A integer vector indicating the number of continuous identical elements in x.

Examples

```
# example I
x1 = c(0,0,0, 1,1,1)
conticnt(x1)
conticnt(x1, cnt=TRUE)

x2 = c(1, 2,2, 3,3,3)
conticnt(x2)
conticnt(x2, cnt=TRUE)

x3 = c('c','c','c', 'b','b', 'a')
conticnt(x3)
conticnt(x3, cnt=TRUE)

# example II
dt = data.frame(c1=x1, c2=x2, c3=x3)
conticnt(dt, col=c('c1', 'c2'))
conticnt(dt, col=c('c1', 'c2'), cnt = TRUE)
```

date_bop *start/end date by period*

Description

The date of bop (beginning of period) or eop (end of period).

Usage

```
date_bop(freq, x, workday = FALSE)

date_eop(freq, x, workday = FALSE)
```

Arguments

freq the frequency of period. It supports weekly, monthly, quarterly and yearly.

x a date

workday logical, whether to return the latest workday

Value

date_bop returns the beginning date of period of corresponding x by frequency.

date_eop returns the end date of period of corresponding x by frequency.

Examples

```
date_bop('weekly', Sys.Date())
date_eop('weekly', Sys.Date())

date_bop('monthly', Sys.Date())
date_eop('monthly', Sys.Date())
```

date_from *start date by range*

Description

The date before a specified date by date_range.

Usage

```
date_from(date_range, to = Sys.Date(), default_from = "1000-01-01")
```

Arguments

date_range	date range, available value including nd, nm, mtd, qtd, ytd, ny, max.
to	a date, default is current system date.
default_from	the default date when date_range is sett to max

Value

It returns the start date of a date_range with a specified end date.

Examples

```
date_from(3)
date_from('3d')

date_from('3m')
date_from('3q')
date_from('3y')

date_from('mtd')
date_from('qtd')
date_from('ytd')
```

date_lwd *latest workday*

Description

The latest workday date of n days before a specified date.

Usage

```
date_lwd(n, to = Sys.Date())
```

Arguments

`n` number of days

`to` a date, default is current system date.

Value

It returns the latest workday date that is n days before a specified date.

Examples

```
date_lwd(5)
date_lwd(3, "2016-01-01")
date_lwd(3, "20160101")
```

date_num *date to number*

Description

It converts date to numeric value in specified unit.

Usage

```
date_num(x, unit = "s", origin = "1970-01-01", scientific = FALSE)
```

Arguments

`x` date.

`unit` time unit, available values including milliseconds, seconds, minutes, hours, days, weeks.

`origin` original date, defaults to 1970-01-01.

`scientific` logical, whether to encode the number in scientific format, defaults to FALSE.

Examples

```
# setting unit
date_num(Sys.time(), unit='milliseconds')
date_num(Sys.time(), unit='mil')

date_num(Sys.time(), unit='seconds')
date_num(Sys.time(), unit='s')

date_num(Sys.time(), unit='days')
date_num(Sys.time(), unit='d')

# setting origin
date_num(Sys.time(), unit='d', origin = '1970-01-01')
date_num(Sys.time(), unit='d', origin = '2022-01-01')

# setting scientific format
date_num(Sys.time(), unit='mil', scientific = FALSE)
date_num(Sys.time(), unit='mil', scientific = TRUE)
date_num(Sys.time(), unit='mil', scientific = NULL)
```

merge2 *merge data.frames list*

Description

Merge a list of data.frames by common columns or row names.

Usage

```
merge2(datlst, by = NULL, all = TRUE, ...)
```

Arguments

datlst	a list of data.frames.
by	A vector of shared column names in x and y to merge on. This defaults to the shared key columns between the two tables. If y has no key columns, this defaults to the key of x.
all	logical; all = TRUE is shorthand to save setting both all.x = TRUE and all.y = TRUE.
...	Additional parameters provided in the merge function.

`reprate` *char repetition rate*

Description

reprate estimates the max rate of character repetition.

Usage

```
reprate(x, col)
```

Arguments

x	a character vector or a data frame.
col	a character column name.

Value

a numeric vector indicating the max rate of character repetition in the corresponding elements in argument x vector.

Examples

```
x = c('a', 'aa', 'ab', 'aab', 'aaab')
reprate(x)

reprate(data.frame(x=x), 'x')
```

Index