Package ‘vvconverter’

February 1, 2023

Title Apply Transformations to Data
Version 0.5.8
Description Provides a set of functions for data transformations.
Transformations are performed on character and numeric data. As the scope of the package is within Student Analytics, there are functions focused around the academic year.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.2.3
Imports dplyr, lubridate, stringr
NeedsCompilation no
Author Tomer Iwan [aut, cre, cph]
Maintainer Tomer Iwan <t.iwan@vu.nl>
Repository CRAN
Date/Publication 2023-02-01 10:50:02 UTC

R topics documented:

academic_year .................................................. 2
clean_multiple_underscores ................................ 2
destring ......................................................... 3
interval_round .................................................. 4
ltrim ............................................................... 4
median_top_10 .................................................... 5
mode ............................................................... 5
rtrim .............................................................. 6
str_replace_all_in_file ....................................... 6
sum_0_1 ............................................................ 7
test_01 ............................................................ 8
transform_01_to_ft ............................................ 8
trim ............................................................... 9

Index 10
### academic_year

#### Description

In this function, a date is translated to the academic year in which it falls. This is based on a start of the academic year on the 1st of September.

#### Usage

```r
academic_year(x, start_1_oct = FALSE)
```

#### Arguments

- **x**: A date, or vector with multiple dates. POSIXct is also accepted.
- **start_1_oct**: Does the academic year start on the 1st of October? default FALSE: based on September 1st

#### Value

The academic year in which the specified date falls

#### See Also

Other vector calculations: `clean_multiple_underscores()`, `interval_round()`, `sum_0_1()`, `transform_01_to_ft()`

#### Examples

```r
academic_year(lubridate::today())
```

---

### clean_multiple_underscores

#### Description

Replaces multiple underscores into a single underscore in a vector or string.

#### Usage

```r
clean_multiple_underscores(x)
```

#### Arguments

- **x**: The vector or string to be cleaned.
**destring**

Value

cleaned vector or string.

See Also

Other vector calculations: academic_year(), interval_round(), sum_0_1(), transform_01_to_ft()

Examples

clean_multiple underscores("hello___world")

destring("2k")
destring("5,5")

destring

Convert character vector to numeric, ignoring irrelevant characters.

Description

Convert character vector to numeric, ignoring irrelevant characters.

Usage

destring(x, keep = "0-9.-")

Arguments

x
A vector to be operated on

keep
Characters to keep in, in bracket regular expression form. Typically includes 0-9 as well as the decimal separator (. in the US and , in Europe).

Value

vector of type numeric

Examples

destring("2k")
destring("5,5")
**Interval round**

**Description**
Function to round numeric values in a vector to values from an interval sequence.

**Usage**
```
interval_round(x, interval)
```

**Arguments**
- `x`: The numeric vector to adjust
- `interval`: The interval sequence

**Value**
The vector corrected for the given interval

**See Also**
Other vector calculations: `academic_year()`, `clean_multipleunderscores()`, `sum_0_1()`, `transform_01_to_ft()`

**Examples**
```
interval_round(c(5, 4, 2, 6), interval = seq(1:4))
```

---

**LTrim**

**Description**
Trim leading whitespace from string.

**Usage**
```
ltrim(x)
```

**Arguments**
- `x`: A text string.

**Value**
Cleaned string.
**median_top_10**

**Examples**

```r
trim(" hello")
```

---

**Description**

Calculate the median of the top ten percentage of the values.

**Usage**

```r
median_top_10(x, na.rm = FALSE)
```

**Arguments**

- `x`: A numerical vector
- `na.rm`: Default TRUE: Remove NAs, before calculations.

**Value**

A numerical value

**Examples**

```r
median_top_10(mtcars$cyl)
```

---

**mode**

**Mode (most common value)**

**Description**

Determine the most common value in a vector. If two values have the same frequency, the first occurring value is used.

**Usage**

```r
mode(x, na.rm = FALSE)
```

**Arguments**

- `x`: a vector
- `na.rm`: If TRUE: Remove nas before the calculation is done
**Value**

the most common value in the vector x

**Examples**

```r
mode(c(0,3,5,7,5,3,2))
```

---

**rtrim**  
**RTrim**

---

**Description**

Trim trailing whitespaces from string.

**Usage**

```r
rtrim(x)
```

**Arguments**

- `x`: A text string.

**Value**

Cleaned string.

**Examples**

```r
trim("hello ")
```

---

**str_replace_all_in_file**

*Replace all occurrences of a pattern in a file*

**Description**

Replace all occurrences of a pattern in a file

**Usage**

```r
str_replace_all_in_file(file, pattern, replacement = "[...]", only_comments = TRUE, collapse = FALSE)
```
**Arguments**

- **file**: character, path of file to be modified
- **pattern**: character, pattern to be replaced
- **replacement**: character, replacement text
- **only_comments**: logical, should the replacement only be done in comments
- **collapse**: logical, should the lines be collapsed into a single line before replacement

**Value**

NULL, the file is modified in place

---

**Description**

This function is the same as `sum()`, with one exception: If the outcome value is higher than 1, it will always return 1.

**Usage**

```r
sum_0_1(x)
```

**Arguments**

- **x**: a vector with numeric values

**Value**

0 or 1. Depending on whether the sum is greater than 0 or not.

**See Also**

Other vector calculations: `academic_year()`, `clean_multiple_underscores()`, `interval_round()`, `transform_01_to_ft()`
test_01  

**Description**

This function tests whether the vector is actually a boolean, but is encoded as a 0/1 variable. The function checks for numeric vectors whether the only occurring values are 0, 1, or NA. At character and factor vectors checks whether the only occurring values are "0", "1", or NA to be. If there is a 0/1 variable, TRUE is returned, in all others cases FALSE.

**Usage**

test_01(x)

**Arguments**

x  
The vector to test

**Value**

A TRUE/FALSE value on the test

**See Also**

Other booleans: transform_01_to_ft()

**Examples**

```r
test_01(c(0, 1, 0, 1, 1, 1, 0))
```

---

transform_01_to_ft  

**Description**

If the vector is a 0/1 vector, it is converted to a logical one TRUE/FALSE vector. This transformation is performed only if the vector contains only values 0, 1, or NA. If this is not the case returns the original variable. This transformation can be done on numeric, string, and factor vectors.

**Usage**

transform_01_to_ft(x)

**Arguments**

x  
the vector to be tested and transformed.
Value

The transformed vector if a transformation is possible. If no transformation is possible, the original vector returned.

See Also

Other vector calculations: academic_year(), clean_multipleunderscores(), interval_round(), sum_0_1()
Other booleans: test_01()

Examples

vector <- c(0, 1, 0, 1, 1, 1, 0)
transform_01_to_ft(vector)

trim

Trim both leading and trailing whitespaces from string.

Usage

trim(x)

Arguments

x A text string.

Value

Cleaned string.

Examples

trim(" hello ")
Index

* booleans
  test_01, 8
  transform_01_to_ft, 8
* tests
  test_01, 8
* vector berekeningen
  destring, 3
* vector calculations
  academic_year, 2
  clean_multiple_underscores, 2
  interval_round, 4
  sum_0_1, 7
  transform_01_to_ft, 8

academic_year, 2, 3, 4, 7, 9

clean_multiple_underscores, 2, 2, 4, 7, 9

destring, 3

interval_round, 2, 3, 4, 7, 9

ltrim, 4

median_top_10, 5
mode, 5

rtrim, 6

str_replace_all_in_file, 6
sum_0_1, 2–4, 7, 9

test_01, 8, 9
transform_01_to_ft, 2–4, 7, 8, 8
trim, 9