Package ‘dformula’
December 1, 2023

Type Package
Version 1.0
Date 2023-12-01
Title Data Manipulation using Formula
Description A tool for manipulating data using the generic formula. A single formula allows to easily add, replace and remove variables before running the analysis.

Depends R (>= 3.5.0)
Imports utils, stats, formula.tools(>= 1.7.1)
Suggests knitr, rmarkdown
VignetteBuilder knitr
License GPL (>= 2)
Repository CRAN
URL https://github.com/serafinialessio/dformula
BugReports https://github.com/serafinialessio/dformula/issues

R topics documented:

add .................................................. 2
population_data ................................. 3
remove ............................................. 4
rename ............................................ 5
select ............................................. 6
transform ....................................... 8

Index 10
add

Add variables

Description
Add new variables by mutating the input variables using a formula.

Usage
add(from, formula, as = NULL,
    position = c("right", "left"),
    na.remove = FALSE, logic_convert = TRUE,...)

Arguments
from a data.frame object with variables
formula a formula indicating the operation to create new variables. Look at the detail section for explanation.
as a character vector with names of new variables.
position if the new variables are positioned at the beginning (right) or at the left (left) of the data in input.
na.remove a logical value indicating whether NA values should be removed.
logic_convert logical value indicating if the new logical variable are convertet to 0 or 1
... further arguments

Details
The formula is composed of two part:
~ new_variables
the right-hand are the new variable to add starting from the existing variables, using the I() function.
For example:
~ I(log(column_names1)) + I(column_names2/100)
the column_names1 and log(column_names1) are added to the data.
If na.remove is set to TRUE, new variables are created, added to the dataset in input and then the observation with missing are removed.

Value
Returns a data.frame object with the original and the new variables.

Author(s)
Alessio Serafini
Examples

```r
data("airquality")
dt <- airquality

head(add(from = dt, formula = ~ log(Ozone)))
head(add(from = dt, formula = ~ log(Ozone) + log(Wind)))
head(add(from = dt, formula = ~ log(Ozone), as = "Ozone_1"))

head(add(from = dt, formula = Ozone + Wind ~ log()))
head(add(from = dt, formula = ~ log()))
head(add(from = dt, formula = .~ log(), position = "left"))

head(add(from = dt, formula = .~ log(), na.remove = TRUE))
head(add(from = dt, formula = ~ I((Ozone>5))))
head(add(from = dt, formula = ~ I((Ozone>5)), logic_convert = FALSE ))

head(add(from = dt, formula = Ozone + Wind ~ C(Ozone-Ozone)))
head(add(from = dt, formula = ~ C(log(Ozone))))
head(add(from = dt, formula = ~ C(5)))
head(add(from = dt, formula = Ozone + Wind ~ C(Ozone-Ozone)))
head(add(from = dt, formula = Ozone + Wind ~ C(log(Ozone))))

foo <- function(x, a = 100){return(x-x + a)}

head(add(from = dt, formula = Ozone + Month~ I(foo(a = 100))))
head(add(from = dt, formula = Ozone + Month~ foo()))
head(add(from = dt, formula = ~ I(foo(Ozone, a = 100))))
```

---

**population_data**  
**World population**

<table>
<thead>
<tr>
<th>population_data</th>
<th>World population</th>
</tr>
</thead>
</table>

**Description**

World population and countries are

**Usage**

```r
data("population_data")
```

**Format**

A data frame with 159 observations on the following 3 variables.

Country a character vector with countries names
**Population** a numeric vector with population

**Area** a numeric vector with area of the counties

**Source**

https://www.worldometers.info

**Examples**

```r
data(population_data)
str(population_data)
```

<table>
<thead>
<tr>
<th>remove</th>
<th>Remove a subset</th>
</tr>
</thead>
</table>

**Description**

Selects the row and the variables to remove by specifying a condition using a formula.

**Usage**

```r
remove(from, formula = .~, na.remove = FALSE, ...)
```

**Arguments**

- `from` a data.frame object with variables
- `formula` a formula indicating the operation to create new variables. Look at the detail section for explanation.
- `na.remove` a logical value indicating whether NA values should be removed.
- `...` further arguments

**Details**

The formula is composed of two parts:

```
column_names ~ rows_conditions
```

the left-hand side are the names of the column to remove, and the right-hand the operation to remove the rows, using the `I()` function.

For example:

```r
column_names1 + column_names2 ~ I(column_names1 == "a") + I(column_names2 > 4)
```

first the row are selected to be removed if the observation in the `column_names1` are equal to "a" and if the observation in the `column_names2` are bigger than 4, then the `column_names1` and `column_names2` are removed and the other variables are returned.

If `na.remove` is set to TRUE, after the subsetting the observations with missing are removed.
rename

Value

Returns a data.frame object without the selected elements.

Author(s)

Alessio Serafini

Examples

data("airquality")
dt <- airquality

head(remove(from = dt, formula = .~ I(Ozone > 10)))
head(remove(from = dt, formula = .~ I(Ozone > 10), na.remove = TRUE))
head(remove(from = dt, formula = Ozone - .))

head(remove(from = dt, formula = Ozone - I(Ozone > 10)))
head(remove(from = dt, formula = Ozone + Wind - I(Ozone > 10)))

head(remove(from = dt, formula = Ozone + . ~ I(Ozone > 10)))
head(remove(from = dt, formula = Ozone + NULL ~ I(Ozone > 10)))

rename Rename variables

Description

Rename variables using formulas

Usage

rename(from, formula, ...)

Arguments

from a data.frame object with variables
formula a formula indicating the operation to create new variables. Look at the detail section for explanation.
... further arguments
Details
The formula is composed of two parts:
\[ \text{column} \_\text{names} \sim \text{new} \_\text{variables} \_\text{name} \]
the left-hand side selects the columns to change the names, and the right-hand side the new names of the selected columns.
For example:
\[ \text{column} \_\text{names}_1 + \text{column} \_\text{names}_2 \sim \text{new} \_\text{variables} \_\text{name}_1 + \text{new} \_\text{variables} \_\text{name}_2 \]
the name of the column 1 and the name of the column 2 are changed in \text{new} \_\text{variables} \_\text{name}_1 and \text{new} \_\text{variables} \_\text{name}_2.

Value
The original data.frame with changed column names.

Author(s)
Alessio Serafini

Examples
```
data(\text{"airquality"})
dt <- \text{airquality}

\text{head(rename(from = dt, Ozone \sim Ozone1))}
\text{head(rename(from = dt, Ozone + Wind \sim Ozone\_new + Wind\_new))}
```

Description
Selects the row and the variables by specifying a condition using a formula.

Usage
```
select(from, formula = \_.\_, as = NULL, na.remove = FALSE, na.return = FALSE,...)
```

Arguments
- from: a data.frame object with variables
- formula: a formula indicating the operation to create new variables. Look at the detail section for explanation
- as: a character vector with names of new variables
- na.remove: a logical value indicating whether NA values should be removed
select

na.return a logical value indicating whether only the observation with NA values should be shown

... further arguments

Details

The formula is composed of two parts:

- column_names ~ row_conditions
  - the left-hand side are the names of the columns to select, and the right-hand side are the operations to select the rows, using the I() function.

For example:

\[
\text{column_names1 + column_names2 ~ I(column_names1 == \text{"a"}) + I(column_names2 > 4)}
\]

first the rows are selected if the observation in the column_names1 are equal to \text{a} and if the observation in the column_names2 are bigger than 4, then the column_names1 and column_names2 are returned.

If na.remove is set to TRUE, after the subsetting the observations with missing are removed.

Value

Returns a data.frame object containing the selected elements.

Author(s)

Alessio Serafini

Examples

data("airquality")
dt <- airquality

### Selects columns and filter rows

select(from = dt, formula = .~ I(Ozone > 10 & Wind > 10))
select(from = dt, formula = Ozone ~ I(Wind > 10))
select(from = dt, formula = Ozone + Wind ~ I(Ozone > 10))

### All rows and filter columns

select(from = dt, formula = Ozone ~ .)
select(from = dt, formula = Ozone + Wind ~ NULL)
**transform**

*Transform variables*

**Description**

Mutate input variables using a formula.

**Usage**

```r
transform(from, formula, as = NULL,
          na.remove = FALSE, logic_convert = TRUE, ...)
```

**Arguments**

- `from`: a data.frame object with variables
- `formula`: a formula indicating the operation to create new variables. Look at the detail section for explanation.
- `as`: a character vector with names of new variables.
- `na.remove`: a logical value indicating whether NA values should be removed.
- `logic_convert`: logical value indicating if the new logical variable are converted to 0 or 1
- `...`: further arguments

**Details**

The formula is composed of two parts:

```
column_names ~ transformed_variables
```

the left-hand side are the names of the column to transform, and the right-hand the operations applied to the selected columns, using the `I()` function.

For example:

```
column_names1 + column_names2 ~ I(log(column_names1)) + I(column_names2/100)
```

the `column_names1` is mutated in `log(column_names1)` and `column_names2` is divided by 100.

If `na.remove` is set to `TRUE`, variables are mutated, and then the observation with missing are removed.

**Value**

Returns the original data.frame object with mutated variables.

**Author(s)**

Alessio Serafini
Examples

data("airquality")
dt <- airquality

head(transform(from = dt, Ozone ~ I(Ozone-Ozone)))
head(transform(from = dt, Ozone ~ log(Ozone)))
head(transform(from = dt, Ozone ~ I(Ozone>5)))
head(transform(from = dt, Ozone ~ I(Ozone>5), logic_convert = TRUE))

head(transform(from = dt, ~ log()))
head(transform(from = dt, . ~ log()))
head(transform(from = dt, NULL ~ log()))

head(transform(from = dt, Ozone + Day - log()))
head(transform(from = dt, Ozone + Day - log(Ozone/100) + exp(Day)))
head(transform(from = dt, Ozone ~ log()))

head(transform(from = dt, Ozone + Wind ~ C(log(1))))
head(transform(from = dt, Ozone + Wind ~ log(Ozone) + C(10)))

head(transform(from = dt, Ozone + Wind ~ C(log(Ozone))))

foo <- function(x, a = 100){return(x-x + a)}
head(transform(from = dt, Ozone + Wind ~ foo(a = 100)))
head(transform(from = dt, . ~ foo(a = 100)))
head(transform(from = dt, Ozone + Wind ~ log(log(1))))
Index

* datasets
  population_data, 3

add, 2
population_data, 3
remove, 4
rename, 5
select, 6
transform, 8