

Package ‘catcont’

July 22, 2025

Title Test, Identify, Select and Mutate Categorical or Continuous Values

Version 0.5.0

Date 2018-06-23

Description Methods and utilities for testing, identifying, selecting and mutating objects as categorical or continuous types. These functions work on both atomic vectors as well as recursive objects: data.frames, data.tables, tibbles, lists, etc..

URL <https://github.com/decisionpatterns/catcont>
<http://www.decisionpatterns.com>

BugReports <https://github.com/decisionpatterns/catcont/issues>

Depends R (>= 3.3.0)

Suggests testthat, data.table(>= 1.10.0)

Imports dplyr (>= 0.7.0)

License GPL-2 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

Repository CRAN

NeedsCompilation no

Author Christopher Brown [aut, cre],
Decision Patterns [cph]

Maintainer Christopher Brown <chris.brown@decisionpatterns.com>

Date/Publication 2018-06-25 07:43:03 UTC

Contents

cat_cont	2
mutate_if_cat	4
select_cat	5

Index[7](#)

cat_cont	<i>categorical or continuous variables</i>
----------	--

Description

These functions facilitate working with variables as categorical or continuous rather than logical, integer, numeric, factor, character, ..

Usage

```
cat_cont(x)

is_cat(x)

## Default S3 method:
is_cat(x)

## S3 method for class 'ordered'
is_cat(x)

## S3 method for class 'factor'
is_cat(x)

## S3 method for class 'logical'
is_cat(x)

is_cont(x)

## Default S3 method:
is_cont(x)

## S3 method for class 'logical'
is_cont(x)

## S3 method for class 'factor'
is_cont(x)

## S3 method for class 'ordered'
is_cont(x)

which_cat(x, ..., names = FALSE)

which_cont(x, ..., names = FALSE)
```

Arguments

x	object
...	arguments passed to other functions.
names	logical; whether to return the names of the variables instead of their index?

Details

These functions are used to test and identify which/if a variable or variables are categorical or continuous. `is_cat` and `is_cont` take single variable arguments.

Mostly, the categorical and continuous assessment is straight- forward. Continuous variables are represented by integer, double or complex types. All other types are categorical. There are a few opinionated exceptions:

- **factors** are categorical (though typed 'integer')
- **ordered** factors are (though typed 'integer')
- **logical** are categorical

For simplicity, it is assumed that a vector cannot be simultaneous categorical and continuous, though in some cases (e.g. ordered factors) this may be the case.

Value

`cat_cont` returns a named character with values either "cat" or "cont". If x is a atomic vector, a single string is given. If x is recursive, a "cat"/"cont" value is given for each element. Names correspond to the names of the element.

`is_cat` and `is_cont` return logical.

`which_cat` and `which.cont` report which variables in an object are categorical and continuous. By default, integer indices are return. If `names=TRUE`, the names of the variables are returned instead.

See Also

- `base::typeof()`
- `base::is.numeric()` `methods::is()`
- `base::which()`

Examples

```
data(iris)
cat_cont(iris)
```

```
is_cat(letters)           # TRUE
is_cat(factor(letters))   # TRUE
is_cat(TRUE)              # TRUE
is_cat(FALSE)             # TRUE
is_cat(1:10)              # FALSE
is_cat(rnorm(10))         # FALSE
is_cat( Sys.Date() )      # FALSE
```

```

is_cat( complex(1,2) )    # FALSE

is_cont(letters)          # FALSE
is_cont(factor(letters)) # FALSE
is_cont(TRUE)             # FALSE
is_cont(FALSE)            # FALSE
is_cont(1:10)             # TRUE
is_cont(rnorm(10))        # TRUE
is_cont( Sys.Date() )    # TRUE
is_cont( complex(1,2) )  # TRUE

which_cat(iris)
which_cat( iris, names=TRUE )

which_cont(iris)
which_cont( iris, names=TRUE )

```

mutate_if_cat	<i>mutate_if_cat, mutate_if_cont</i>
---------------	--------------------------------------

Description

mutates only categorical/continuous columns

Usage

```

mutate_if_cat(.tbl, .funs, ...)

## Default S3 method:
mutate_if_cat(.tbl, .funs, ...)

## S3 method for class 'data.table'
mutate_if_cat(.tbl, .funs, ...)

mutate_if_cont(.tbl, .funs, ...)

## Default S3 method:
mutate_if_cont(.tbl, .funs, ...)

## S3 method for class 'data.table'
mutate_if_cont(.tbl, .funs, ...)

```

Arguments

.tbl	table
.funs	functions see dplyr::mutate_if()
...	additional parameters

Details

Mutates categorical or continuous columns.

The data.table variants do this as

Value

An object of class `.tbl` in with columns mutated according to `.funs`

See Also

Similar to `dplyr::mutate_if()`

Examples

```
data(iris)

## Not run:
iris %>% mutate_if_cat( as.character )

library(data.table)
setDT(iris)
class(iris$Species)
iris %>% mutate_if_cat( as.character )
class(iris1$Species) # character
class(iris2)

iris %>% mutate_if_cont( add, 2 )

## End(Not run)
```

select_cat	<i>select_cat, select_cont</i>
------------	--------------------------------

Description

Select columns by type

Usage

```
select_cat(.data)

## Default S3 method:
select_cat(.data)

## S3 method for class 'data.table'
select_cat(.data)
```

```
select_cont(.data)

## Default S3 method:
select_cont(.data)

## S3 method for class 'data.table'
select_cont(.data)
```

Arguments

.data table

Details

select_cat() and select_cont() return only the categorical and continuous types respectively. This is closely mirrors the dplyr function select but works with non-table values as well.

Value

Returns a table-like object of the same class as data unless there are no columns in which case 'NULL' is returned

Examples

```
data(iris)
select_cat(iris)
select_cont(iris)

## Not run:
setDT(iris)
select_cat(iris)
select_cont(iris)

## End(Not run)
```

Index

`base::is.numeric()`, [3](#)

`base::typeof()`, [3](#)

`base::which()`, [3](#)

`cat_cont`, [2](#)

`dplyr::mutate_if()`, [4](#), [5](#)

`is_cat (cat_cont)`, [2](#)

`is_cont (cat_cont)`, [2](#)

`methods::is()`, [3](#)

`mutate_if_cat`, [4](#)

`mutate_if_cont (mutate_if_cat)`, [4](#)

`select_cat`, [5](#)

`select_cont (select_cat)`, [5](#)

`which_cat (cat_cont)`, [2](#)

`which_cont (cat_cont)`, [2](#)