Package: ilabelled (via r-universe)

November 5, 2024

Type Package

Title Simple Handling of Labelled Data

Version 1.0.0

Author Christof Lewerenz [aut, cre]

Maintainer Christof Lewerenz <christof.lewerenz@gmx.net>

Description Simple handling of survey data. Smart handling of meta-information like e.g. variable-labels value-labels and scale-levels. Easy access and validation of meta-information. Useage of value labels and values respectively for subsetting and recoding data.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Imports methods, stats

URL https://github.com/clewerenz/ilabelled

BugReports https://github.com/clewerenz/ilabelled/issues

Config/testthat/edition 3

RoxygenNote 7.3.2

Depends R (>= 2.10)

VignetteBuilder knitr

Repository https://clewerenz.r-universe.dev

RemoteUrl https://github.com/clewerenz/ilabelled

RemoteRef HEAD

RemoteSha a99271a748f604ce2871cf1e4522ba63c7ceac48

Contents

`.init` *backend for i_labelled*

Description

all arguments are passed from i_labelled

Usage

```
.init(
  x,
  label = NULL,
  labels = NULL,
  na_values = NULL,
  na_range = NULL,
  scale = NULL,
  annotation = NULL,
  wording = NULL,
```

```
  subject = NULL,
  ...
)
```

Arguments

x	vector
label	variable label
labels	value labels as named vector
na_values	missing values (e.g. c(888, 999))
na_range	range of missing values (e.g. c(-9,-1))
scale	scale level (nominal, ordinal, scale)
annotation	additional information about variable
wording	question text
subject	subject
...	further attributes passed to class

Value

x as i_labelled object with attributes applied to it

.is_sequential *checks if vector is numeric sequence*

Description

checks if vector is numeric sequence

Usage

```
.is_sequential(x)
```

Arguments

x	vector

Value

T/F

.i_find_in *internal replacement of match function for remove missing values (match is much slower but can handle more data classes)*

Description

description description

Usage

```
.i_find_in(x, y)
```

Arguments

x	vector
y	vector

Value

Vector of T/F values with length of x

.i_in *Match values*

Description

Find matches (return T/F)

Usage

```
.i_in(x, table)
```

Arguments

x	vector or NULL: the values to be matched. Long vectors are supported.
table	vector or NULL: the values to be matched against. Long vectors are not supported.

Value

Vector of T/F values with length of x

.merge_labels *combine old value labels with new value labels*

Description

combine old value labels with new value labels

Usage

```
.merge_labels(old_labs, new_labs)
```

Arguments

old_labs	named vector
new_labs	named vector

Value

Returns names vector of value labels

.valid_annotation *validate annotation - intern*

Description

contains run-time-tests annotation runs internally

Usage

```
.valid_annotation(x)
```

Arguments

x	character vector or NULL

Value

T/F

.valid_label *validate variable label - intern*

Description

run-time-tests for variable label runs internally

Usage

```
.valid_label(x)
```

Arguments

x	vector

Value

T/F

.valid_labels *validate value labels - intern*

Description

contains several run-time-tests for value labels runs internally

Usage

```
.valid_labels(x)
```

Arguments

x	named vector (label = value)

Value

No return value. Aborts process when run-time-tests fail.

.valid_missing *validate missing values/range - intern*

Description

validate missing values/range - intern

Usage

```
.valid_missing(x)
```

Arguments

x	vector

Value

T/F

.valid_scale *validate scale label - intern*

Description

run-time-tests for scale level runs internally

Usage

```
.valid_scale(x)
```

Arguments

x	vector

Value

T/F

.valid_subject *validate subject - intern*

Description

contains run-time-tests subject runs internally

Usage

```
.valid_subject(x)
```

Arguments

x	character vector or NULL

Value

T/F

.valid_wording *validate wording - intern*

Description

contains run-time-tests wording runs internally

Usage

```
.valid_wording(x)
```

Arguments

x	character vector or NULL

Value

T/F

as.i_labelled *coerce to i_labelled class*

Description

coerce to i_labelled class

Usage

```
as.i_labelled(x, ...)
```

Arguments

x	vector
...	attributes passed to class

Value

vector of class i_labelled

grapesingrapes *generic for %in%*

Description

generic for %in%

Usage

```
x %in% table
```

Arguments

x	vector or NULL: the values to be matched. Long vectors are supported.
table	vector or NULL: the values to be matched against. Long vectors are not supported.

Value

T/F

is.i_labelled *check for class i_labelled*

Description

check for class i_labelled

Usage

```
is.i_labelled(x)
```

Arguments

x	vector of class i_labelled

Value

T/F

is_decimal *Check if vector contains decimal values*

Description

Check if vector contains decimal values

Usage

```
is_decimal(x)
```

Arguments

x	numeric vector

Value

T/F

i_annotation *add annotation to variable*

Description

add annotation to i_labelled object

can be used to store additional information about a variable

Usage

```
i_annotation(x, annotation, overwrite = FALSE)
```

Arguments

x	vector
annotation	variable label as string or NULL (NULL will remove label)
overwrite	overwrite existing annotation and replace with new annotation

Value

x with annotation applied

i_assert_labels *Check for required value labels in set of variables*

Description

Check for required value labels in set of variables

Usage

```
i_assert_labels(x, labels, info = NULL, verbose = TRUE)
```

Arguments

x	data.frame
labels	character vector
info	string with info message (purpose of assertion) - optional
verbose	return TRUE when assertion is successful

Value

No return value (exept when verbose = T). Aborts process when test not valid.

i_as_character *as character*

Description

make character from i_labelled

Usage

```
i_as_character(
  x,
  missing_to_na = FALSE,
  require_all_labels = FALSE,
  keep_attributes = FALSE
)
```

Arguments

x	vector
missing_to_na	as missing declared values will become NA
require_all_labels	process will be interrupted, when not all values have valid labels
keep_attributes	should attributes be preserved

Value

character vector

i_as_factor *as factor*

Description

make factor from i_labelled

Usage

```
i_as_factor(
  x,
  missing_to_na = FALSE,
  require_all_labels = FALSE,
  keep_attributes = FALSE
)
```

Arguments

x	vector
missing_to_na	as missing declared values will become NA
require_all_labels	process will be interrupted, when not all values have valid labels
keep_attributes	should attributes be preserved

Value

vector of class factor

---

i_as_numeric *as numeric*

Description

make numeric from i_labelled

Usage

```
i_as_numeric(x, missing_to_na = FALSE, keep_attributes = FALSE)
```

Arguments

x	vector
missing_to_na	as missing declared values will become NA
keep_attributes	should attributes be preserved

Value

numeric vector

i_copy *copy meta information from one variable to another*

Description

copy meta information from one variable to another

Usage

```
i_copy(to, from, what = "all", ...)
```

Arguments

to	vector
from	vector
what	character vector describing which labels are copied: 'all' (default), 'label', 'labels', 'na_values', 'na_range'
...	further attributes passed to structure

Value

Returns 'to' with ilabelled attributes copied from 'from'

i_get_annotation *get annotation*

Description

return annotation as character vector applied to vector return list when applied to data.frame

Usage

```
i_get_annotation(x)
```

Arguments

x	vector or data.frame

Value

returns annotation

i_get_equal_subject *get variable names by subject*

Description

return all variable names by subjects

one, several, or all subjects can be looked up

Usage

```
i_get_equal_subject(x, subject = NULL)
```

Arguments

x	data.frame
subject	one or more subjects as character vector. when NULL return all variable names by all subjects in data

Value

named list or NA. return named list with one list entry for each subject. when no subject in data or no match for subjects, return NA.

i_get_equal_wording *get variable names by wording*

Description

return all variable names by wordings

one, several, or all wordings can be looked up

Usage

```
i_get_equal_wording(x, wording = NULL)
```

Arguments

x	data.frame
wording	one or more wordings as character vector. when NULL return all variable names by all wordings in data

Value

named list or NA. return named list with one list entry for each wording. when no wording in data or no match for wordings, return NA.

i_get_label *get variable label*

Description

return variable label when applied to vector return list when applied to data.frame

Usage

```
i_get_label(x)
```

Arguments

x	vector or data.frame

Value

variable label

i_get_labels *get value labels*

Description

return labels when applied to vector return list when applied to data.frame

Usage

```
i_get_labels(x)
```

Arguments

x	vector or data.frame

Value

values and value labels as data.frame

i_get_na_range *get missing range*

Description

return missing range when applied to vector return list when applied to data.frame

Usage

```
i_get_na_range(x)
```

Arguments

x	vector or data.frame

Value

return missing range

i_get_na_values *get missing values*

Description

return missing values when applied to vector return list when applied to data.frame

Usage

```
i_get_na_values(x)
```

Arguments

x	vector or data.frame

Value

return missing values

i_get_scale *get scale level*

Description

return scale level when applied to vector return list when applied to data.frame

Usage

```
i_get_scale(x)
```

Arguments

x	vector or data.frame

Value

returns scale level

i_get_subject *get subject*

Description

return subject as character vector applied to vector return list when applied to data.frame

Usage

```
i_get_subject(x)
```

Arguments

x	vector or data.frame

Value

returns subject

i_get_wording *get wording*

Description

return wording as character vector applied to vector return list when applied to data.frame

Usage

```
i_get_wording(x)
```

Arguments

x	vector or data.frame

Value

returns wording

i_label *set variable label*

Description

set variable label

Usage

```
i_label(x, label)
```

Arguments

x	vector
label	variable label as string or NULL (NULL will remove label)

Value

x with variable label applied

`i_labelled` *class constructor*

Description

class constructor

Usage

```
i_labelled(
  x,
  label = NULL,
  labels = NULL,
  na_values = NULL,
  na_range = NULL,
  scale = NULL,
  annotation = NULL,
  wording = NULL,
  subject = NULL,
  ...
)
```

Arguments

`x`	vector or data.frame
`label`	variable label
`labels`	value labels as named vector (e.g. c("A"=1, "B"=2) or setNames(c(1,2), c("A","B")))
`na_values`	missing values (e.g. c(888, 999))
`na_range`	range of missing values as vector length 2 (e.g. c(-9,-1))
`scale`	scale level (nominal, ordinal, scale)
`annotation`	additional information about variable
`wording`	question text
`subject`	subject
`...`	further attributes passed to class

Value

vector or data.frame

i_labels *set value labels*

Description

set value labels

Usage

```
i_labels(x, ..., overwrite = FALSE)
```

Arguments

x	vector
...	set labels for values (e.g. label_of_choice = 1 or "Label of Choice" = 1); remove single label with NULL = value (e.g. NULL = 1); removes all value labels when only NULL (e.g. i_label(x, NULL))
overwrite	should new labels be merged with existing labels or remove existing labels

Details

In order to assign a specific label to multiple values a named list can also be provided to ... (e.g. list(missing = -9:-1, valid = 1:3))

A named vector can also be provided (e.g. setNames(c(1,2), c("A","B")))

Value

returns x with value labels applied

i_missing_to_na *missing values to NA*

Description

all values declared as missing will be recoded as NA

Usage

```
i_missing_to_na(x, remove_missing_labels = FALSE)
```

Arguments

x	vector or data.frame
remove_missing_labels	remove values labels from values which are declared as missing

Value

Returns x with missing values coerced to NA

`i_na_range` *define missing range*

Description

define which values will be handled as missing values

Usage

```
i_na_range(x, values)
```

Arguments

`x`	vector
`values`	vector with missing range e.g. c(-9:-1) or NULL (NULL will remove all missing values)

Value

Returns x with missing range set

`i_na_values` *define missing values*

Description

define which values will be handled as missing values

Usage

```
i_na_values(x, values, sort = TRUE, desc = FALSE)
```

Arguments

`x`	vector
`values`	vector with missing values e.g. c(888,999) or NULL (NULL will remove all missing values)
`sort`	sort values
`desc`	sort values in descending order

Value

Returns x with missing values set

i_print_annotation *print annotation*

Description

print annotation

Usage

```
i_print_annotation(x)
```

Arguments

x	vector

Value

No return value. Print annotation attribute to console

i_print_attributes *print attributes*

Description

print attributes

Usage

```
i_print_attributes(x, exclude = NULL)
```

Arguments

x	vector
exclude	character vector with attribute names not taken into account

Value

No return value. Print attributes to console

i_print_label *print variable label*

Description

print variable label

Usage

```
i_print_label(x)
```

Arguments

x	vector

Value

No return value. Print variable label to console

i_print_labels *print value labels*

Description

print value labels

Usage

```
i_print_labels(x)
```

Arguments

x	vector

Value

No return value. Print labels to console

i_print_na_range *print missing range*

Description

print missing range

Usage

```
i_print_na_range(x)
```

Arguments

x	vector

Value

No return value. Print na range to console

i_print_na_values *print missing values*

Description

print missing values

Usage

```
i_print_na_values(x)
```

Arguments

x	vector

Value

No return value. Print na values to console

i_print_scale *print scale level*

Description

print scale level

Usage

```
i_print_scale(x)
```

Arguments

x	vector

Value

No return value. Print scale level to console

i_print_subject *print subject*

Description

print subject

Usage

```
i_print_subject(x)
```

Arguments

x	vector

Value

No return value. Print subject attribute to console

i_print_wording *print wording*

Description

print wording

Usage

```
i_print_wording(x)
```

Arguments

x	vector

Value

No return value. Print wording attribute to console

i_recode *i_recode Function for recoding new variable from origin variable(s).*

Description

Returns a vector object of class i_labelled

Usage

```
i_recode(
  x,
  ...,
  label = NULL,
  na_values = NULL,
  na_range = NULL,
  scale = NULL,
  annotation = NULL,
  wording = NULL,
  subject = NULL,
  copy = NULL,
  keep_labels = FALSE
)
```

Arguments

x	vector or data.frame
...	formula for recoding of values. See examples.
label	variable label
na_values	a vector with missing values
na_range	a vector for missing range
scale	scale level (nominal, ordinal, metric)
annotation	addition information about variable
wording	question text
subject	subject
copy	When applied to vector: T/F. When applied to a data.frame: a variable from x. Copy the values of an existing variable or x before recoding values according to ...
keep_labels	keep value labels from origin vector when copy TRUE or variable from x

Details

Can be applied to either vector or data.frame. When x is data.frame the formula passed to ... is different from when it is applied to single vector. When function is applied to a data.frame, multiple conditions on multiple variables are possible (e.g when variable X is equal to this, do that; when variable Y is not equal to this, do that, etc.). See examples for further clarification.

You can recode directly via value labels by using

Value

Returns i_labelled vector with values defined by formula and information given to function.

Examples

```
# When applied to a single vector:
# keep in mind that when function is applied to vector, instead of a column use x
myVector <- i_labelled(1:4, labels = c("A" = 1, "B" = 2, "C" = 3, "D" = 4))
i_recode(x = myVector, "AB" = 1 ~ x %in% c("A", "B"), "CD" = 2 ~ x == c(3, 4))

# When applied to data.frame (multiple conditions)
myData <- data.frame(
   V1 = i_labelled(1:3, labels = c("A" = 1, "B" = 2, "C" = 3)),
   V2 = i_labelled(c(2:3,-9))
)
i_recode(x = myData, A = 1 ~ V1 %in% c("A", "B"), 2 ~ "V2" == 3, "C" = 999 ~ V2 == -9)
```

i_remove_annotation *remove annotation*

Description

remove annotation label from variable keep other attributes

Usage

```
i_remove_annotation(x)
```

Arguments

x	vector or data.frame

Value

Returns x without annotation

i_remove_label *remove variable label*

Description

remove variable label keep other attributes

Usage

```
i_remove_label(x)
```

Arguments

x	vector or data.frame

Value

Returns x without variable label

i_remove_labels *remove all value labels*

Description

remove all value labels keep other attributes

Usage

```
i_remove_labels(x)
```

Arguments

x	vector or data.frame

Value

Returns x without value labels

i_remove_missing_labels *remove missing labels*

Description

remove values labels from values which are declared as missing

Usage

```
i_remove_missing_labels(x)
```

Arguments

x	vector or data.frame

Value

Returns x without missing labels

i_remove_na_range *remove as na range*

Description

remove na range (information which values should be handled as missing) keep other attributes

Usage

```
i_remove_na_range(x)
```

Arguments

x	vector or data.frame

Value

Returns x without na-range

i_remove_na_values *remove as na values*

Description

remove na values (information which values should be handled as missing) keep other attributes

Usage

```
i_remove_na_values(x)
```

Arguments

x	vector or data.frame

Value

Returns x without na-values

i_remove_scale *remove scale level*

Description

remove scale label from variable keep other attributes

Usage

```
i_remove_scale(x)
```

Arguments

x	vector or data.frame

Value

Returns x without scale level

i_remove_subject *remove subject*

Description

remove subject label from variable keep other attributes

Usage

```
i_remove_subject(x)
```

Arguments

x	vector or data.frame

Value

Returns x without subject

i_remove_wording *remove wording*

Description

remove wording label from variable keep other attributes

Usage

```
i_remove_wording(x)
```

Arguments

x	vector or data.frame

Value

Returns x without wording

i_scale *set scale level*

Description

set scale level

Usage

```
i_scale(x, scale = NULL)
```

Arguments

x	vector
scale	scale level (nominal, ordinal, scale) as string or NULL (NULL will remove scale level)

Value

Returns x with scale label set

i_sort_labels *sort value labels by values or by labels*

Description

sort value labels by values or by labels

Usage

```
i_sort_labels(x, by = "values", decreasing = FALSE)
```

Arguments

x	vector or data.frame
by	either values or labels
decreasing	sort decreasing

Value

Returns x with sorted value labels

i_subject *add subject to variable*

Description

add subject to i_labelled object

Usage

```
i_subject(x, subject)
```

Arguments

x	vector
subject	variable label as string or NULL (NULL will remove label)

Value

x with subject applied

i_table *cross tabulation and table creation using i_labelled labels*

Description

wrapper for base::table

convert i_labelled objects to base class and pass to table function

Usage

```
i_table(..., missing_to_na = TRUE, as_factor = TRUE, table_args = NULL)
```

Arguments

...	one or more atomic vectors or one data.frame
missing_to_na	make as missing declared values NA
as_factor	make labelled data factor before pass to table
table_args	arguments of base::table as named list

Value

returns a contingency table, an object of class "table"

Examples

```
set.seed(1234)
a <- sample(c(1:3, NA), 10, replace = TRUE)
b <- i_labelled(sample(c(1:3, NA), 10, replace = TRUE), labels = c("A" = 1, "B" = 2, "C" = 3))
c <- factor(sample(c("X", "Y", "Z", NA), 10, replace = TRUE))
df <- data.frame(a, b, c)

i_table(a, b)
i_table(df, table_args = list(useNA = "ifany"))
```

i_to_base_class *remove class i_labelled and return base R class*

Description

- when value labels for all values are available will return factor

- when value labels are missing will unclass i_labelled

- remove class i_labelled and return variable as base R class

Usage

```
i_to_base_class(
  x,
  missing_to_na = TRUE,
  as_factor = TRUE,
  keep_attributes = FALSE
)
```

Arguments

x	vector or data.frame
missing_to_na	missing values will become regular NA
as_factor	convert to factor when value labels are available
keep_attributes	should attributes be preserved

Value

Returns x coerced to R base class

---

i_unclass *unclass variables*

Description

unclass variables

Usage

```
i_unclass(x, keep_attributes = FALSE)
```

Arguments

x	vector or data.frame
keep_attributes	should attributes be preserved

Value

x unclassed

i_valid_annotation *validate annotation*

Description

returns boolean when applied to vector

returns a named list when applied to data.frame

Usage

```
i_valid_annotation(x)
```

Arguments

x	vector or data.frame

Value

T/F

i_valid_label *validate variable labels*

Description

returns boolean when applied to vector

returns a named list when applied to data.frame

Usage

```
i_valid_label(x)
```

Arguments

x	vector or data.frame

Value

T/F

i_valid_labels *validate value labels*

Description

returns boolean when applied to vector

returns a named list when applied to data.frame

Usage

```
i_valid_labels(x)
```

Arguments

x	vector or data.frame

Value

No return value. Aborts process when run-time-tests fail

i_valid_scale *validate variable scale level*

Description

returns boolean when applied to vector

returns a named list when applied to data.frame

Usage

```
i_valid_scale(x)
```

Arguments

x	vector or data.frame

Value

T/F

i_valid_subject *validate subject*

Description

returns boolean when applied to vector

returns a named list when applied to data.frame

Usage

```
i_valid_subject(x)
```

Arguments

x	vector or data.frame

Value

T/F

i_valid_wording *validate wording*

Description

returns boolean when applied to vector

returns a named list when applied to data.frame

Usage

```
i_valid_wording(x)
```

Arguments

x	vector or data.frame

Value

T/F

i_wording *add wording to variable*

Description

add wording to i_labelled object

can be used to store question text

Usage

```
i_wording(x, wording)
```

Arguments

x	vector
wording	variable label as string or NULL (NULL will remove label)

Value

x with wording applied

print.i_labelled *custom print method for i_labelled*

Description

custom print method for i_labelled

Usage

```
## S3 method for class 'i_labelled'
print(x, ...)
```

Arguments

x	vector of class i_labelled
...	not used

Value

No return value. Print object data and information to console

[.i_labelled *subsetting vectors of class i_labelled*

Description

subsetting vectors of class i_labelled

Usage

```
## S3 method for class 'i_labelled'
x[...]
```

Arguments

x	vector of class i_labelled
...	not used

Value

Subset of x

[[.i_labelled *subsetting vectors of class i_labelled*

Description

subsetting vectors of class i_labelled

Usage

```
## S3 method for class 'i_labelled'
x[[...]]
```

Arguments

x	vector of class i_labelled
...	not used

Value

Subset of x

Index