Package: lablah (via r-universe)

May 28, 2026

Title Working with Labelled Data on Steroids

Version 0.1-1

Description Labelled data is ubiquitous. Package 'lablah' provides extra tools for creating, working, and reporting based on labelled data.

Imports DT, dplyr, glue, labelled, purrr, rlang, stats, tibble, tidyr, tidyselect

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/testthat/parallel true

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-04-28 07:34:46 UTC

RemoteUrl https://github.com/mbojan/lablah

RemoteRef HEAD

RemoteSha 5c10b24f67e27564f2fb5a83e2bab7e9302e9752

Contents

cdbk_df *Codebook*

Description

Generate a codebook based on data documentation (variable names, variable labels, and value labels).

Usage

```
cdbk_df(data)

cdbk_dt(data, ...)
```

Arguments

`data`	labelled tibble
`...`	other arguments passed to `DT::datatable()`

Value

Function `cdbk_df()` returns a tibble with columns:

- `variable` - variable name
- `label` - variable label
- `value_labels` - list of tibbles with value labels

Function `cdbk_dt()` uses `DT::datatable()` to generate a browsable and searchable codebook based on the documentation.

freq_df *Frequency tibble*

Description

Frequency tibble

Usage

```
freq_df(data, var = -1, name = NULL, ...)
```

Arguments

data	data frame
var	Variable name or index (negative counting from the right, positive counting from the left), see dplyr::pull()
name	Column which values will be used as names. Specified as var. C.f. dplyr::pull()
...	other arguments

---

misvals *Lists and tibbles with user missing values*

Description

Lists and tibbles with user missing values

Usage

```
misvals(data)

misranges(data)

misranges_df(data)
```

Arguments

data	Labelled tibble with missing values/ranges defined

Value

Function misvals() returns a list of vectors of missing values (or NULLs if none are defined). List elements are named with variable names.

Function misranges() returns a list of two-element vectors with ranges of missing values.

Function misranges_df() returns a tibble with columns

- variable - Variable name
- from, to - Bounds (inclusive) of the missing value range

seems *Heuristically guess or verify type of a variable*

Description

Given a vector x guess what kind of "type" it is, where "type" correspond to some common classes influencing useful ways to analyze/visualize. These functions are used internally to determine default behavior of some other functions.

- seems_integer() – Returns TRUE if x is typeof() integer or is a numeric with all values being in fact integers (i.e. equal to round(x)).
- seems_continuous() – Returns TRUE if x is numeric and, if an integer, has more than 10 distinct values.
- seems_discrete() – Returns TRUE if x does not seem to be continuous.
- seems_categorical() -

Usage

```
seems_integer(x)

seems_continuous(x)

seems_discrete(x)

seems_categorical(x)
```

Arguments

x	a vector for which is.atomic() is TRUE

Value

All seems_*() functions return TRUE or FALSE (a logical scalar).

Examples

```
# seems_integer() ------------------------------------------------------------
seems_integer(1:5)       # TRUE
seems_integer(runif(5))  # FALSE

# seems_continuous() ---------------------------------------------------------
seems_continuous(1:5)       # FALSE
seems_continuous(1:11)      # TRUE
seems_continuous(runif(5))  # TRUE

# Summarize variables of `mtcars` in a type-dependent way by drawing a
```

```
# histogram for continuous ones and barchart with counts for non-continuous
# ones.
iscont <- vapply(mtcars, seems_continuous, logical(1))
layout(matrix(1:12, 3, 4))
for(n in names(mtcars)) {
  if(iscont[n]) {
    hist(mtcars[[n]], main=n, xlab="")
  } else {
    barplot(table(mtcars[[n]]), main=n)
  }
}
layout(1)
```

`vallabs` *Extract value labels and return as tibble(s)*

Description

Extract value labels and return as tibble(s)

Usage

```
vallabs(object, ...)

## Default S3 method:
vallabs(object, ...)

## S3 method for class 'data.frame'
vallabs(object, ...)
```

Arguments

`object`	R object, typically a vector or data frame. See below for available methods
`...`	other arguments to/from other methods. Currently ignored.

Value

The default method returns a single tibble with columns:

- `label` - Value label
- `value` - Value

If `object` is a data frame `vallabs()` returns a list of tibbles (or `NULL`s of no value labels are present) named with variable names. Each tibble has columns as described in the default method.

Methods (by class)

- `default`: Extract value labels and return a tibble.
- `data.frame`: Extract value labels from all the columns and return a list of tibbles.

See Also

labelled::val_labels()

`varlabs` *Tibble of variable labels*

Description

Tibble of variable labels

Usage

```
varlabs(data)
```

Arguments

`data`	data frame with labelled variables

Value

A tibble with columns

- `variable` - Variable name
- `label` - Variable label

See Also

labelled::var_label()

Index