

# Package: tinylabels (via r-universe)

September 7, 2024

**Title** Lightweight Variable Labels

**Version** 0.2.4.9000

**Date** 2023-09-02

**Description** Assign, extract, or remove variable labels from R vectors.  
Lightweight and dependency-free.

**Imports** methods, stats

**Suggests** testthat, vctrs, dplyr, knitr, rmarkdown

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**URL** <https://github.com/mariusbarth/tinylabels>

**BugReports** <https://github.com/mariusbarth/tinylabels/issues>

**VignetteBuilder** knitr

**Repository** <https://mariusbarth.r-universe.dev>

**RemoteUrl** <https://github.com/mariusbarth/tinylabels>

**RemoteRef** HEAD

**RemoteSha** 3c9772c81746780af0037bb7e90a3afa15bc83fb

## Contents

as.character.tiny_labelled . . . . .	2
label_variable . . . . .	2
relevel.tiny_labelled . . . . .	3
unlabel . . . . .	3
variable_label . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

as.character.tiny\_labelled

*Conversion of Labelled Vectors*

---

### Description

Functions to convert labelled vectors to other types, possibly keeping the variable label and the class attribute tiny\_labelled.

### Usage

```
## S3 method for class 'tiny_labelled'
as.character(x, keep_label = TRUE, ...)

## S3 method for class 'tiny_labelled'
as.logical(x, keep_label = TRUE, ...)

## S3 method for class 'tiny_labelled'
as.integer(x, keep_label = TRUE, ...)

## S3 method for class 'tiny_labelled'
as.double(x, keep_label = TRUE, ...)

## S3 method for class 'tiny_labelled'
as.complex(x, keep_label = TRUE, ...)
```

### Arguments

x	Object to be coerced
keep_label	Logical indicating whether the variable labels and class tiny_labelled should be kept.
...	Further arguments passed to methods

---

label\_variable

*Label Variables Using Pipes*

---

### Description

label\_variable() can be used to assign variable labels within a workflow using the tidyverse's pipe operator.

### Usage

```
label_variable(x, ...)

label_variables(x, ...)
```

**Arguments**

x Either a vector or a data.frame.  
 ... Variable label(s) to be assigned. For data frames, these have to be name-value pairs, see example.

**Examples**

```
library(dplyr)
test <- npk %>%
  label_variable(N = "Nitrogen", P = "Phosphate")
variable_label(test)
```

---

relevel.tiny\_labelled *Reorder Levels of Labelled Factor*

---

**Description**

The levels of a factor are re-ordered so that the level specified by ref is first and the others are moved down. This is a copy from [relevel](#) in the **stats** package, but preserves the label attribute and class tiny\_labelled.

**Usage**

```
## S3 method for class 'tiny_labelled'
relevel(x, ref, ...)
```

**Arguments**

x an unordered factor.  
 ref the reference level, typically a string.  
 ... additional arguments for future methods.

---

unlabel *Remove Labels from Objects*

---

**Description**

Remove [variable\\_labels](#) from a labelled vector or from the columns of a data frame.

**Usage**

```
unlabel(x)
```

**Arguments**

x An R object.

**Value**

Object as `x` but without variable labels and with class `tiny_labelled` removed.

---

<code>variable_label</code>	<i>Assign or Extract Variable Labels</i>
-----------------------------	--

---

**Description**

Assign or extract variable labels of a vector *or* the columns (i.e., vectors) of a `data.frame`.

**Usage**

```
variable_label(x, ...)

## Default S3 method:
variable_label(x, ...)

## S3 method for class 'data.frame'
variable_label(x, ...)

variable_label(x) <- value

## Default S3 replacement method:
variable_label(x) <- value

## S3 replacement method for class 'data.frame'
variable_label(x) <- value

variable_labels(x, ...)

variable_labels(x) <- value
```

**Arguments**

<code>x</code>	Either a vector or a <code>data.frame</code> .
<code>...</code>	Further arguments that may be passed to methods.
<code>value</code>	Character. The variable label(s) to be assigned. If <code>variable_label()</code> is applied to a single vector, this should be a length-one argument. If applied to a <code>data.frame</code> , <code>value</code> is required to be a <i>named</i> vector or a <i>named</i> list. NULL elements of this name-value list are ignored. Check the examples for details.

**Value**

For vectors, `variable_label()` returns NULL or the variable label (typically of length one). For data frames, `variable_label()` returns a named list where each column corresponds to a column of the data frame.

The assignment methods `variable_label()<-` return the labelled object.

**See Also**

See [label\\_variable\(\)](#) for an alternative that is compatible with the tidyverse's pipe operator.

**Examples**

```
# label a single vector
variable_label(letters) <- "The alphabet" # Assign
variable_label(letters)                  # Extract

# label some columns of a data frame:
variable_labels(npk) <- c(                # Assign
  N = "Nitrogen"
  , P = "Phosphate"
  , K = "Potassium"
)
variable_labels(npk)                      # Extract

# using a list on the right, character and expression can be mixed:
variable_labels(npk) <- list(            # Assign
  N = "Nitrogen"
  , P = "Phosphate"
  , K = expression(italic(K))
)
variable_labels(npk)                      # Extract
```

# Index

`as.character.tiny_labelled`, 2  
`as.complex.tiny_labelled`  
    (`as.character.tiny_labelled`), 2  
`as.double.tiny_labelled`  
    (`as.character.tiny_labelled`), 2  
`as.integer.tiny_labelled`  
    (`as.character.tiny_labelled`), 2  
`as.logical.tiny_labelled`  
    (`as.character.tiny_labelled`), 2

`label_variable`, 2  
`label_variable()`, 5  
`label_variables` (`label_variable`), 2

`relevel`, 3  
`relevel.tiny_labelled`, 3

`unlabel`, 3

`variable_label`, 4  
`variable_label<-` (`variable_label`), 4  
`variable_labels`, 3  
`variable_labels` (`variable_label`), 4  
`variable_labels<-` (`variable_label`), 4