

Package ‘formatdown’

March 15, 2024

Title Formatting Tools for 'rmarkdown' Documents

Version 0.1.3

Language en-US

Description Provides a small set of tools for formatting tasks when creating documents in R Markdown or Quarto Markdown. Convert the elements of a numerical vector to character strings in one of several forms: powers-of-ten notation in engineering or scientific form delimited for rendering as inline equations; integer or decimal notation delimited for equation rendering; numbers with measurement units (non-delimited) where units are selected to eliminate the need for powers-of-ten or scientific notation. Useful for rendering a numerical scalar in an inline R code chunk as well as formatting columns of data frames displayed in a table.

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

LazyDataCompression bzip2

RoxygenNote 7.2.3

Imports checkmate, data.table, units, wrapr

Suggests covr, knitr, rmarkdown, tinytest

VignetteBuilder knitr

URL <https://github.com/graphdr/formatdown/>,
<https://graphdr.github.io/formatdown/>,
<https://CRAN.R-project.org/package=formatdown>

BugReports <https://github.com/graphdr/formatdown/issues>

Collate 'data.R' 'utils.R' 'format_decimal.R' 'format_power.R'
'format_text.R' 'format_units.R' 'formatdown-package.R'

NeedsCompilation no

Author Richard Layton [aut, cre]

Maintainer Richard Layton <graphdoctor@gmail.com>

Repository CRAN

Date/Publication 2024-03-15 15:50:04 UTC

R topics documented:

air_meas	2
atmos	3
format_decimal	3
format_power	5
format_text	7
format_units	9
metals	11
water	11

Index **13**

air_meas	<i>Air density measurements</i>
----------	---------------------------------

Description

Table of air properties at room temperature and pressure, simulating multiple measurements at approximately steady state,

Usage

```
data(air_meas, package = "formatdown")
```

Format

Classes data.table and data.frame: 5 observations of 7 variables:

date "Date" class format "YYYY-MM-DD".

trial Character, label "a" through "e".

humid Factor, humidity, "low", "med", or "high."

temp Numeric, measured temperature (K).

pres Numeric, measured atmospheric pressure (Pa).

sp_gas Numeric, specific gas constant in mass form R_{sp} , ideal gas reference value, ($\text{J kg}^{-1}\text{K}^{-1}$).

dens Numeric, calculated air density $\rho = pR_{sp}^{-1}T^{-1}$ (kg m^{-3}).

`atmos`*Properties of standard atmosphere*

Description

Table of atmospheric properties as a function of altitude, sea level to 80 km.

Usage

```
data(atmos, package = "formatdown")
```

Format

Classes `data.table` and `data.frame`: 9 observations of 5 variables:

alt Numeric, altitude (m)

temp Numeric, air temperature (K)

pres Numeric, atmospheric pressure (Pa)

dens Numeric, air density (kg m^{-3})

sound Numeric, speed of sound (m/s)

Source

Marks' Standard Handbook for Mechanical Engineers 9/e (1987) E.A. Avallone and T. Baumeister (ed.), "Table 4.2.2 International Standard Atmosphere", pp. 4-38, McGraw-Hill, NY.

`format_decimal`*Format decimal or integer values*

Description

Convert the elements of a numerical vector to character strings in which the numbers are formatted using decimal notation and delimited for rendering as inline equations in an R Markdown document.

Usage

```
format_decimal(x, digits = 4, ..., big_mark = NULL, delim = "$")
```

Arguments

x	Numeric vector to be formatted.
digits	Numeric scalar, decimal places to report, integer between 0 and 20. Zero returns an integer.
...	Not used, force later arguments to be used by name.
big_mark	Character. If not empty, used as mark between every three digits before the decimal point. Applied as the big.mark argument of formatC().
delim	Character vector (length 1 or 2) defining the delimiters for marking up inline math. Possible values include "\$" or "\\(", both of which create appropriate left and right delimiters. Alternatively, left and right can be defined explicitly in a character vector of length two, e.g., c("\$", "\$") or c("\\(", "\\)"). Custom delimiters can be assigned to suit the markup environment. Use argument by name.

Details

Given a number, a numerical vector, or a numerical column from a data frame, `format_decimal()` converts the numbers to character strings of the form, "\$a\$", where a is the number in decimal notation. The user can specify the number of decimal places.

Delimiters for inline math markup can be edited if necessary. If the default argument fails, the "\\(" alternative is available. If using a custom delimiter to suit the markup environment, be sure to escape all special symbols.

Value

A character vector with the following properties:

- Numbers represented in integer or decimal notation.
- Elements delimited as inline math markup.

See Also

Other format_*: [format_power\(\)](#), [format_text\(\)](#), [format_units\(\)](#)

Examples

```
# Digits
x <- c(12300400.1234, 456000)
format_decimal(x, digits = 0)
format_decimal(x, digits = 1)
format_decimal(x, digits = 2)

# Big mark
format_decimal(x, 0, big_mark = ",")

# Inline math delimiters
x <- c(1.654321, 0.065432)
format_decimal(x)
```

```

format_decimal(x, 3, delim = "$")
format_decimal(x, 3, delim = c("$", "$"))
format_decimal(x, 3, delim = "\\(")
format_decimal(x, 3, delim = c("\\(", "\\))")

# LaTeX-style display equation delimiters
format_decimal(x, 3, delim = c("\\[", "\\])")

# Apply to columns of a data frame (data.table syntax)
DT <- atmos[, .(temp, sound)]
DT[, lapply(.SD, function(x) format_decimal(x, 1))]

```

format_power

Format powers of ten

Description

Convert the elements of a numerical vector to character strings in which the numbers are formatted using powers-of-ten notation in scientific or engineering form and delimited for rendering as inline equations in an R Markdown document.

Usage

```

format_power(
  x,
  digits = 4,
  ...,
  format = NULL,
  size = NULL,
  omit_power = c(-1, 2),
  set_power = NULL,
  delim = "$"
)

```

Arguments

<code>x</code>	Numeric vector to be formatted.
<code>digits</code>	Numeric scalar between 1 and 20 (inclusive) defining the number of significant digits in result.
<code>...</code>	Not used, force later arguments to be used by name.
<code>format</code>	Character. Possible values are "enr" (default) for engineering notation and "sci" for scientific notation. Use argument by name. Can also be set as a global option, for example, <code>options(formatdown.power.format = "sci")</code> that can be overwritten in an individual function call.
<code>size</code>	Font size. Possible values are "scriptsize", "small" (default), "normalsize", "large", and "huge". which correspond to selected LaTeX font size values. Can also be set as a global option, for example, <code>options(formatdown.font.size = "normalsize")</code> that can be overwritten in an individual function call.

omit_power	Numeric vector $c(p, q)$ specifying the range of exponents between which power of ten notation is omitted, where $p \leq q$. If NULL all numbers are formatted in powers of ten notation. Use argument by name.
set_power	Numeric scalar integer. Assigned exponent that overrides format. Default NULL makes no notation changes. Use argument by name.
delim	Character vector (length 1 or 2) defining the delimiters for marking up inline math. Possible values include "\$" or "\\(", both of which create appropriate left and right delimiters. Alternatively, left and right can be defined explicitly in a character vector of length two, e.g., $c("$", "$")$ or $c("\\(", "\\)")$. Custom delimiters can be assigned to suit the markup environment. Use argument by name.

Details

Given a number, a numerical vector, or a numerical column from a data frame, `format_power()` converts the numbers to character strings of the form, " $a \times 10^{\{n\}}$ ", where a is the coefficient and n is the exponent. The string includes markup delimiters `$. . $` for rendering as an inline equation in R Markdown or Quarto Markdown document.

The user can specify either scientific or engineering format and the number of significant digits.

Powers-of-ten notation is omitted over a range of exponents via `omit_power` such that numbers are converted to character strings of the form, " a ", where a is the number in decimal notation. The default `omit_power = c(-1, 2)` formats numbers such as 0.123, 1.23, 12.3, and 123 in decimal form. To cancel these exceptions and convert all numbers to powers-of-ten notation, set the `omit_power` argument to NULL.

Delimiters for inline math markup can be edited if necessary. If the default argument fails, try using "\\(" as an alternative. If using a custom delimiter to suit the markup environment, be sure to escape all special symbols.

Value

A character vector with the following properties:

- Numbers represented in powers of ten notation except for those with exponents in the range specified in `omit_power`
- Elements delimited as inline math markup.

See Also

Other `format_*`: `format_decimal()`, `format_text()`, `format_units()`

Examples

```
# Scalar value
format_power(101100, digits = 4)

# Vector value
x <- c(1.2222e-6, 2.3333e-5, 3.4444e-4, 4.1111e-3, 5.2222e-2, 6.3333e-1,
      7.4444e+0, 8.1111e+1, 9.2222e+2, 1.3333e+3, 2.4444e+4, 3.1111e+5, 4.2222e+6)
```

```

format_power(x)

# Compare significant digits
format_power(x[1], 3)
format_power(x[1], 4)

# Compare format type
format_power(x[3], format = "engr")
format_power(x[3], format = "sci")

# Compare set_power results
format_power(x[3], set_power = -5)
format_power(x[3], set_power = -4)
format_power(x[3], set_power = -3)

# Compare omit_power range
format_power(x[6], omit_power = c(-1, 2))
format_power(x[6], omit_power = c(0, 2))
format_power(x[8])
format_power(x[8], omit_power = NULL)

# Apply to columns of a data frame (data.table syntax)
y <- x[1:6]
z <- x[8:13]
DT <- data.table::data.table(y, z)
DT[, lapply(.SD, function(x) format_power(x))]

```

format_text

Format text to match the math font

Description

Format a vector as character strings within math delimiters for rendering in an R Markdown document. The formatting is particularly useful for formatting text columns in a table in which numerical columns have been formatted using `format_power()` or `format_decimal()`.

Usage

```
format_text(x, ..., face = NULL, size = NULL, delim = "$")
```

Arguments

x	Vector to be formatted.
...	Not used, force later arguments to be used by name.
face	Font face. Possible values are "plain", "italic", "bold", "sans", or "mono". Can also be set as a global option, for example, <code>options(formatdown.font.face = "sans")</code> that can be overwritten in an individual function call.

size	Font size. Possible values are "scriptsize", "small" (default), "normalsize", "large", and "huge". which correspond to selected LaTeX font size values. Can also be set as a global option, for example, <code>options(formatdown.font.size = "normalsize")</code> that can be overwritten in an individual function call.
delim	Character vector (length 1 or 2) defining the delimiters for marking up inline math. Possible values include "\$" or "\\(", both of which create appropriate left and right delimiters. Alternatively, left and right can be defined explicitly in a character vector of length two, e.g., <code>c("\$", "\$")</code> or <code>c("\\(", "\\)")</code> . Custom delimiters can be assigned to suit the markup environment. Use argument by name.

Details

Given a scalar, vector, or data frame column, `format_text()` converts its argument to a character string of the form `" \mathxx{a} "` where `a` is the element to be formatted and `\mathxx` determines the font face: plain type is set by `\mathrm`; italic by `\mathit`; bold by `\mathbf`; sans serif by `\mathsf`; and monospace (typewriter text) by `\mathtt`. The string includes markup delimiters `$. . $` for rendering as an inline equation in R Markdown or Quarto Markdown document.

Using `options()` to assign a value to `formatdown.font.face` affects only that text formatted using `format_text()`. However, using `options` to assign `formatdown.font.size` affects output formatted by `format_text()` and `format_power()`.

Delimiters for inline math markup can be edited if necessary. If the default argument fails, try using `"\\("` as an alternative. If using a custom delimiter to suit the markup environment, be sure to escape all special symbols.

Value

A character vector with elements delimited as inline math markup in plain, italic, or bold font face.

See Also

Other `format_*`: [format_decimal\(\)](#), [format_power\(\)](#), [format_units\(\)](#)

Examples

```
# Text vector

# default face = "plain"
x <- air_meas$humid
format_text(x)

# equivalently
format_text(x, face = "plain")

# argument coerced to character string if possible
format_text(x = NA)
format_text(x = c(TRUE, FALSE, TRUE))

# numbers as strings are rendered as-is
format_text(x = c("1.2E-3", "3.4E+0", "5.6E+3"))
```



```

# other font faces
format_text(x, face = "italic")
format_text(x, face = "bold")
format_text(x, face = "sans")
format_text(x, face = "mono")

# alternate delimiter
format_text(x, delim = "\\(")

```

format_units

Format values with measurement units

Description

Format a vector of numbers as character strings with measurement units appended via the 'units' package.

Usage

```

format_units(
  x,
  digits = 1,
  ...,
  unit = NULL,
  unit_form = NULL,
  big_mark = NULL
)

```

Arguments

x	Vector of class numeric or class units.
digits	Numeric scalar, a positive integer. Applied as the <code>digits</code> argument of <code>base::format()</code> . Enough decimal places are included such that the smallest magnitude value has this many significant digits.
...	Not used, force later arguments to be used by name.
unit	Character scalar, units label compatible with 'units' package. For x class numeric, transform to class units in unit measurement units. For x class units, convert to unit measurement units. If empty, existing class units retained.
unit_form	Character scalar. Possible values are "standard" (default) and "implicit" (implicit exponent form). In standard form, units are related with arithmetic symbols for multiplication, division, and powers, e.g., "kg/m^3" or "W/(m*K)". In implicit exponent form, symbols are separated by spaces and numbers represent exponents, e.g., "kg m-3" or "W m-1 K-1".
big_mark	Character. Applied as the <code>big.mark</code> argument of <code>base::format()</code> . Default is ".". If a period is selected for <code>big.mark</code> , the decimal mark is changed to a comma.

Details

This function is a wrapper for `units::as_units()` and `base::format()`. Numeric class input is converted to units class. Units class input, if convertible, is converted to the specified measurement units; if none are specified, the existing measurement units are retained. The result in all cases is converted to class character using `base::format()` with preset arguments: `trim = TRUE` and `scientific = FALSE`. The output has the form "a [u]", where a is the number in decimal notation and u is a measurement units label.

Value

A character vector of numbers with appended measurement units.

See Also

Other `format_*`: `format_decimal()`, `format_power()`, `format_text()`

Examples

```
# Scalar value, class numeric
x <- 101300
format_units(x, unit = "Pa")

# Scalar value, class units
x <- 101300
units(x) <- "Pa"
format_units(x, unit = "hPa")
format_units(x, digits = 3, unit = "psi")

# Vectors (atmos and metals data included in formatdown)
x <- atmos$dens
units(x) <- "kg/m^3"
format_units(x, unit = "g/m^3")
format_units(x, unit = "g/m^3", unit_form = "implicit")

x <- atmos$pres
units(x) <- "Pa"
format_units(x, big_mark = ",")
format_units(x, unit = "hPa")

x <- metals$thrm_cond
units(x) <- "W m-1 K-1"
format_units(x, digits = 2)
format_units(x, digits = 2, unit_form = "implicit")
```

metals	<i>Properties of metals</i>
--------	-----------------------------

Description

Table of mechanical and thermal properties of selected metals.

Usage

```
data(metals, package = "formatdown")
```

Format

Classes data.table and data.frame: 6 observations of 5 variables:

metal Character, name of material

dens Numeric, density (kg m^{-3})

thrm_exp Numeric, coefficient of thermal expansion ($\text{m m}^{-1}\text{K}^{-1}$)

thrm_cond Numeric, thermal conductivity ($\text{W m}^{-1}\text{K}^{-1}$)

elast_mod Numeric, modulus of elasticity (Pa)

Source

Marks' Standard Handbook for Mechanical Engineers 9/e (1987) E.A. Avallone and T. Baumeister (ed.), "Basic Properties of Several Metals", pp. 6-11, McGraw-Hill, NY.

water	<i>Properties of water</i>
-------	----------------------------

Description

Table of water properties at atmospheric pressure as a function of temperature.

Usage

```
data(water, package = "formatdown")
```

Format

Classes data.table and data.frame: 11 observations of 5 variables:

temp Numeric, temperature (K)

dens Numeric, density (kg m^{-3})

sp_wt Numeric, specific weight (N m^{-3})

visc Numeric, dynamic viscosity (Pa s)

bulk_mod Numeric, bulk modulus (Pa)

Source

E. Maurer E and I. Embry (2022) *hydraulics: Basic Pipe and Open Channel Hydraulics*, R package ver. 0.6.0, <https://edm44.github.io/hydraulics/>.

Index

* datasets

air_meas, [2](#)

atmos, [3](#)

metals, [11](#)

water, [11](#)

* format_*

format_decimal, [3](#)

format_power, [5](#)

format_text, [7](#)

format_units, [9](#)

air_meas, [2](#)

atmos, [3](#)

base::format(), [10](#)

format_decimal, [3](#), [6](#), [8](#), [10](#)

format_power, [4](#), [5](#), [8](#), [10](#)

format_text, [4](#), [6](#), [7](#), [10](#)

format_units, [4](#), [6](#), [8](#), [9](#)

metals, [11](#)

units::as_units(), [10](#)

water, [11](#)