

Package ‘skimr’

January 10, 2018

Title Compact and Flexible Summaries of Data

Version 1.0.1

Maintainer Elin Waring <elin.waring@gmail.com>

Description A simple to use summary function that can be used with pipes and displays nicely in the console. The default summary statistics may be modified by the user as can the default formatting. Support for data frames and vectors is included, and users can implement their own skim methods for specific object types as described in a vignette. Default summaries include support for inline spark graphs. Instructions for managing these on specific operating systems are given in the “Using skimr” vignette and the README.

Depends R (>= 3.1.2)

Imports dplyr (>= 0.7), magrittr, pander, purrr, rlang, stats, stringr (>= 1.1), knitr, tibble (>= 0.6), tidyr (>= 0.7), tidyselect

Suggests extrafont, rmarkdown, testthat, withr

License GPL-3

Encoding UTF-8

LazyData true

URL <https://github.com/ropenscilabs/skimr>

BugReports <https://github.com/ropenscilabs/skimr/issues>

VignetteBuilder knitr

RoxygenNote 6.0.1

Collate 'skimr-package.R' 'formats.R' 'stats.R' 'functions.R' 'options.R' 'skim.R' 'skim_print.R' 'skim_v.R' 'summary.R' 'utils.R'

NeedsCompilation no

Author Amelia McNamara [aut],
Eduardo Arino de la Rubia [aut],
Hao Zhu [aut],
Julia Lowndes [ctb],

Shannon Ellis [aut],
 Elin Waring [cre],
 Michael Quinn [aut],
 Hope McLeod [ctb],
 Hadley Wickham [ctb],
 Kirill Müller [ctb],
 RStudio, Inc. [cph] (Spark functions),
 Connor Kirkpatrick [ctb],
 Scott Brenstuhl [ctb]

Repository CRAN

Date/Publication 2018-01-10 06:35:37 UTC

R topics documented:

skimr-package	3
inline_hist	3
inline_linegraph	4
kable	4
kable.skim_df	5
kable.summary_skim_df	6
list_lengths_max	7
list_lengths_median	7
list_lengths_min	8
list_max_length	8
list_min_length	9
max_char	9
min_char	10
n_complete	10
n_empty	11
n_missing	11
n_unique	12
pander.skim_df	12
pander.summary_skim_df	13
print.skim_df	13
print.skim_vector	14
print.summary_skim_df	14
show_formats	15
show_skimmers	15
skim	16
skim_format	17
skim_tee	18
skim_to_list	18
skim_to_wide	19
skim_with	19
sorted_count	20
summary.skim_df	21
ts_end	22

<i>skimr-package</i>	3
ts_start	22
Index	23

skimr-package	<i>Skim a data frame</i>
---------------	--------------------------

Description

This package provides an alternative to the default summary functions within R. The package's API is tidy, functions take data frames, return data frames and can work as part of a pipeline. The returned `skimr` object is subsettable and offers a human readable output.

Details

`skimr` is opinionated, providing a strong set of summary statistics that are generated for a variety of different data types. It also provides an API for customization. Users can change both the functions dispatched and the way the results are formatted.

inline_hist	<i>Generate inline histogram for numeric variables</i>
-------------	--

Description

The character length of the histogram is controlled by the formatting options for character vectors.

Usage

```
inline_hist(x)
```

Arguments

`x` A numeric vector.

Value

A length-one character vector containing the histogram.

inline_linegraph	<i>Generate inline line graph for time series variables</i>
------------------	---

Description

Sets all data to a standard length, to match character formatting. This is twice the number of characters set in the histogram.

Usage

```
inline_linegraph(x)
```

Arguments

x A vector

Details

The character length of the linegraph is controlled by the formatting options for character vectors.

Value

A length-one character vector containing a line graph.

kable	<i>Create kable object</i>
-------	----------------------------

Description

Generic method for kable objects based on the method in the knitr package.

Usage

```
kable(x, format = NULL, digits = getOption("digits"), row.names = NA,
      col.names = NA, align, caption = NULL, format.args = list(),
      escape = TRUE, ...)
```

Arguments

x An R object (typically a matrix or data frame)

format a character string; possible values are latex, html, markdown, pandoc, and rst; this will be automatically determined if the function is called within knitr; it can also be set in the global option knitr.table.format; if format is a function, it must return a character string. Defaults to NULL.

<code>digits</code>	The maximum number of digits for numeric columns (passed to <code>round()</code>); it can also be a vector of length <code>ncol(x)</code> to set the number of digits for individual columns
<code>row.names</code>	a logical value indicating whether to include row names; by default, row names are included if <code>rownames(x)</code> is neither <code>NULL</code> nor identical to <code>1:nrow(x)</code>
<code>col.names</code>	a character vector of column names to be used in the table
<code>align</code>	the alignment of columns: a character vector consisting of <code>'l'</code> (left), <code>'c'</code> (center) and/or <code>'r'</code> (right); by default, numeric columns are right-aligned, and other columns are left-aligned; if <code>align = NULL</code> , the default alignment is used; alternatively, if <code>length(align) == 1L</code> , the string will be expanded to a vector of individual letters unless the output format is LaTeX; for example, <code>'clc'</code> will be converted to <code>c('c', 'l', 'c')</code>
<code>caption</code>	the table caption
<code>format.args</code>	a list of arguments to be passed to <code>format()</code> to format table values, e.g. <code>list(big.mark = ',')</code>
<code>escape</code>	escape special characters when producing HTML or LaTeX tables
<code>...</code>	other arguments

See Also[kable](#)

<code>kable.skim_df</code>	<i>Produce kable output of a skimmed data frame</i>
----------------------------	---

Description

Produce kable output of a skimmed data frame

Usage

```
## S3 method for class 'skim_df'
kable(x, format = NULL, digits = getOption("digits"),
      row.names = NA, col.names = NA, align = NULL, caption = NULL,
      format.args = list(), escape = TRUE, ...)
```

Arguments

<code>x</code>	a <code>skim_df</code> object
<code>format</code>	a character string; possible values are <code>latex</code> , <code>html</code> , <code>markdown</code> , <code>pandoc</code> , and <code>rst</code> ; this will be automatically determined if the function is called within knitr; it can also be set in the global option <code>knitr.table.format</code> ; if <code>format</code> is a function, it must return a character string

<code>digits</code>	the maximum number of digits for numeric columns (passed to <code>round()</code>); it can also be a vector of length <code>ncol(x)</code> to set the number of digits for individual columns.
<code>row.names</code>	a logical value indicating whether to include row names; by default, row names are included if <code>rownames(x)</code> is neither <code>NULL</code> nor identical to <code>1:nrow(x)</code>
<code>col.names</code>	a character vector of column names to be used in the table
<code>align</code>	the alignment of columns: a character vector consisting of 'l' (left), 'c' (center) and/or 'r' (right); by default, numeric columns are right-aligned, and other columns are left-aligned; if <code>align = NULL</code> , the default alignment is used; alternatively, if <code>length(align) == 1L</code> , the string will be expanded to a vector of individual letters unless the output format is LaTeX; for example, 'clc' will be converted to <code>c('c', 'l', 'c')</code>
<code>caption</code>	the table caption that precedes the variable type
<code>format.args</code>	a list of arguments to be passed to <code>format()</code> to format table values, e.g. <code>list(big.mark = ',')</code>
<code>escape</code>	escape special characters when producing HTML or LaTeX tables
<code>...</code>	other arguments.

Value

The original `skim_df` object.

See Also

[kable](#)

`kable.summary_skim_df` *Kable method for a summary_skim_df object*

Description

This is a method for the generic function `kable`.

Usage

```
## S3 method for class 'summary_skim_df'
kable(x, ...)
```

Arguments

`x` a `skim_summary` object

`...` Additional arguments affecting the print output produced. Not used.

`list_lengths_max` *Get the maximum length of the lists*

Description

Get the maximum length of the lists

Usage

```
list_lengths_max(x)
```

Arguments

x A vector of list data

Value

Maximum length.

`list_lengths_median` *Get the median length of the lists*

Description

Get the median length of the lists

Usage

```
list_lengths_median(x)
```

Arguments

x A vector of list data

Value

Median length.

<code>list_lengths_min</code>	<i>Get the length of the shortest list in a vector of lists</i>
-------------------------------	---

Description

Get the length of the shortest list in a vector of lists

Usage

```
list_lengths_min(x)
```

Arguments

<code>x</code>	A vector of list data
----------------	-----------------------

Value

Minimum length.

<code>list_max_length</code>	<i>Get the length of the longest list in a vector of lists</i>
------------------------------	--

Description

Get the length of the longest list in a vector of lists

Usage

```
list_max_length(x)
```

Arguments

<code>x</code>	A vector of list data
----------------	-----------------------

Value

Minimum length.

list_min_length	<i>Get the length of the shortest list in a vector of lists</i>
-----------------	---

Description

Get the length of the shortest list in a vector of lists

Usage

```
list_min_length(x)
```

Arguments

x	A vector of list data
---	-----------------------

Value

Minimum length.

max_char	<i>Calculate the maximum number of characters within a character vector</i>
----------	---

Description

Calculate the maximum number of characters within a character vector

Usage

```
max_char(x)
```

Arguments

x	A vector
---	----------

Value

The min of calling nchar(x).

min_char	<i>Calculate the minimum number of characters within a character vector</i>
----------	---

Description

Calculate the minimum number of characters within a character vector

Usage

```
min_char(x)
```

Arguments

x A vector

Value

The min of calling nchar(x).

n_complete	<i>Calculate complete values</i>
------------	----------------------------------

Description

Complete values are not missing

Usage

```
n_complete(x)
```

Arguments

x A vector

Value

The sum of non-NULL and non-NA values

`n_empty`*Calculate the number of blank values in a character vector*

Description

A "blank" is equal to "".

Usage`n_empty(x)`**Arguments**

`x` A vector

Value

The number of values in the vector equal to ""

`n_missing`*Calculate missing values*

Description

Calculate missing values

Usage`n_missing(x)`**Arguments**

`x` A vector

Value

The sum of NULL and NA values

n_unique	<i>Calculate the number of unique elements but remove NA</i>
----------	--

Description

Calculate the number of unique elements but remove NA

Usage

```
n_unique(x)
```

Arguments

x	A vector
---	----------

Value

unique without NA.

pander.skim_df	<i>Produce pander output of a skimmed data frame</i>
----------------	--

Description

Produce pander output of a skimmed data frame

Usage

```
## S3 method for class 'skim_df'
pander(x, caption = attr(x, "caption"), ...)
```

Arguments

x	R object (typically a skimmed data frame)
caption	caption(string) to be shown under the table
...	other arguments.

Value

The original skim_df object.

See Also

[pander](#)

`pander.summary_skim_df`*Pander method for a summary_skim_df object*

Description

This is a method for the generic function pander

Usage

```
## S3 method for class 'summary_skim_df'  
pander(x, ...)
```

Arguments

x	a skim_summary object
...	Additional arguments affecting the print output produced. Not used.

`print.skim_df`*Print skimmed data frame*

Description

Prints a skimmed data frame (created by skim()).

Usage

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

Arguments

x	A skim_df object.
...	Further arguments passed to or from other methods.

Value

The original skim_df object.

print.skim_vector *Manages print for skim_vector objects.*

Description

Manages print for skim_vector objects.

Usage

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

Arguments

x A skim_vector object.
... Further arguments passed to or from other methods.

Value

The original skim_df object.

print.summary_skim_df *Print method for a summary_skim_df object*

Description

This is a method for the generic function print

Usage

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

Arguments

x a skim_summary object
... Additional arguments affecting the print output produced. Not used.

show_formats	<i>Show formatting options currently used, by data type</i>
--------------	---

Description

Show formatting options currently used, by data type

Usage

```
show_formats(which = NULL)
```

Arguments

which	A character vector. One or more of the classes whose formatting options you wish to display.
-------	--

Value

A list of option-value pairs.

show_skimmers	<i>Working with summary functions currently used, by data type</i>
---------------	--

Description

show_skimmers accesses the names of the summary functions for a class, and get_skimmers pulls lists of summary functions for a class.

Usage

```
show_skimmers(which = NULL)
```

```
get_skimmers(which = NULL)
```

Arguments

which	A character vector. One or more of the classes whose summary functions you wish to display.
-------	---

Details

All summary functions are stored within a single nested list. The top level list is named by class, where the inner lists are pairs of function name (for the skim output) and the functions themselves.

Value

A list. The names of the list match the classes that have assigned summary functions. When showing the skimmers, each entry in the list is a character vector of function names. When getting the skimmers, each entry in the list is itself a list of named functions.

Examples

```
# What are the names of the numeric skimmers?
show_skimmers("numeric")

# I want to create a set of skimmers for the hms class, using the date
# skimmers currently available.
funs <- get_skimmers()
skim_with(hms = funs$date)
```

skim

Get useful summary statistic from a data frame

Description

skim handles data of all types, dispatching a different set of summary functions based on the types of columns in the data frame. It is an intentionally simple function. See [skim_with](#) and [skim_format](#) for how skim can be customized. If the rendered examples show unencoded values such as '<U+2587>' you will need to change your locale to allow proper rendering. Please review the Using Skimr vignette for more information.

Usage

```
skim(.data, ...)
```

Arguments

<code>.data</code>	A tbl, or an object that can be coerced into a tbl.
<code>...</code>	Additional options, normally used to list individual unquoted column names.

Value

A `skim_df` object, which can be treated like a `tbl` in most instances.

Examples

```
skim(iris)

# Skim also works groupwise
dplyr::group_by(iris) %>% skim()
```

`skim_format`*Change the formatting options for printed skim objects*

Description

Skim uses `format` to convert the numeric values returned by the summary functions into displayed values. The default options are a subset of options available in that function.

Usage

```
skim_format(...)
```

```
skim_format_defaults()
```

Arguments

... Named arguments that contain named lists specifying formats to apply.

Details

Generally speaking, formats are dispatched according to the type of value returned by the "skimmer," i.e. summary function. One special formatting "type" exists for the names of the returned vector. The names are used to assign the levels for statistics that have more than one value. Counts and quantiles are common cases.

When a vector is named, the name and the value are combined into a single formatted value. To deal with excessively long names for factor levels, only the first three characters of the name are returned by default. This can be changed by setting a new value for `max_char` within the `.levels` type.

Value

Nothing. `invisible(NULL)`

Functions

- `skim_format_defaults`: Use the default formatting options within skim

Examples

```
# Format numbers to have more digits
skim_format(numeric = list(digits = 3))

# Show 4-character names in factor levels
skim_format(.levels = list(nchar = 4))

# Reset to the defaults
skim_format_defaults()
```

skim_tee	<i>Print useful summary statistic from a data frame returning the data frame without modification</i>
----------	---

Description

Print useful summary statistic from a data frame returning the data frame without modification

Usage

```
skim_tee(.data, ...)
```

Arguments

.data	A tbl, or an object that can be coerced into a tbl.
...	Additional options, normally used to list individual unquoted column names.

Value

The input data frame.

skim_to_list	<i>Print skim result and return a list of tibbles</i>
--------------	---

Description

Returns a list of tibbles (also data frames) with one list element per data type. Each column contains the formatted values. This facilitates additional processing. Note that this is not pipeable.

Usage

```
skim_to_list(x, ...)
```

Arguments

x	A dataframe.
...	Further arguments passed to or from other methods.

Value

A list of tibbles.

Examples

```
skim_to_list(iris)
iris %>% skim_to_list()
sl <- iris %>% skim_to_list()
sl[["numeric"]]
```

skim_to_wide	<i>Print skim result and return a single wide data frame of summary statistics</i>
--------------	--

Description

Returns a wide data frame with one row per variable and NA for statistics not calculated for a given type. This facilitates future processing.

Usage

```
skim_to_wide(x, ...)
```

Arguments

x	A dataframe.
...	Further arguments passed to or from other methods.

Value

A wide data frame.

Examples

```
skim_to_wide(iris)
iris %>% skim_to_wide()
iris %>% skim_to_wide() %>% dplyr::filter(type == "factor") %>%
  dplyr::select(top_counts)
```

skim_with	<i>Set or add the summary functions for a particular type of data</i>
-----------	---

Description

While skim is designed around having an opinionated set of defaults, you can use this function to change the summary statistics that it returns. To do that, provide type you wish to change as an argument to this function, along with a list of named functions that you want to use instead of the defaults. The append argument lets you decide whether you want to replace the defaults or add to them.

Usage

```
skim_with(..., append = TRUE)

skim_with_defaults()
```

Arguments

... A list of functions, with an argument name that matches a particular data type.

append Whether the provided options should be in addition to the defaults already in skim for the given types specified by the named arguments in ... Default is TRUE.

Details

This function is not pure. It sets values in within the package environment. This is an intentional design choice, with effects similar to setting options in base R. By setting options here for your entire session, you can continue to summarize using skim on its own.

If the rendered examples show unencoded values such as '<U+2587>' you will need to change your locale to allow proper rendering. Please review the Using Skimr vignette for more information.

Value

Nothing. invisible(NULL)

Functions

- `skim_with_defaults`: Use the default functions within skim

Examples

```
# Use new functions for numeric functions
skim_with(numeric = list(median = median, mad = mad), append = FALSE)
skim(faithful)

# If you want to remove a particular skimmer, set it to NULL
# This removes the inline histogram
skim_with(numeric = list(hist = NULL))
skim(faithful)

# Go back to defaults
skim_with_defaults()
skim(faithful)
```

sorted_count

Create a contingency table and arrange its levels in descending order

Description

In case of ties, the ordering of results is alphabetical and depends upon the locale. NA is treated as a ordinary value for sorting.

Usage

```
sorted_count(x)
```

Arguments

x An object that can be interpreted as a factor (including logical)

Value

A "table" object, which will be treated as a named numeric vector

summary.skim_df *Summary function for skim_df*

Description

This is a method of the generic function summary

Usage

```
## S3 method for class 'skim_df'  
summary(object, ...)
```

Arguments

object a skim dataframe
... Additional arguments affecting the summary produced. Not used.

Value

A summary of the dataframe df

Examples

```
## Not run:  
a <- skim(mtcars)  
summary(a)  
  
## End(Not run)
```

ts_end	<i>Get the finish for a time series without the frequency</i>
--------	---

Description

Get the finish for a time series without the frequency

Usage

```
ts_end(x)
```

Arguments

x A vector of ts data

Value

Finish time.

ts_start	<i>Get the start for a time series without the frequency</i>
----------	--

Description

Get the start for a time series without the frequency

Usage

```
ts_start(x)
```

Arguments

x A vector of ts data

Value

Finish time.

Index

`format`, [17](#)

`get_skimmers (show_skimmers)`, [15](#)

`inline_hist`, [3](#)
`inline_linegraph`, [4](#)

`kable`, [4](#), [5](#), [6](#)
`kable.skim_df`, [5](#)
`kable.summary_skim_df`, [6](#)

`list_lengths_max`, [7](#)
`list_lengths_median`, [7](#)
`list_lengths_min`, [8](#)
`list_max_length`, [8](#)
`list_min_length`, [9](#)

`max_char`, [9](#)
`min_char`, [10](#)

`n_complete`, [10](#)
`n_empty`, [11](#)
`n_missing`, [11](#)
`n_unique`, [12](#)

`pander`, [12](#)
`pander.skim_df`, [12](#)
`pander.summary_skim_df`, [13](#)
`print.skim_df`, [13](#)
`print.skim_vector`, [14](#)
`print.summary_skim_df`, [14](#)

`show_formats`, [15](#)
`show_skimmers`, [15](#)
`skim`, [16](#)
`skim_format`, [16](#), [17](#)
`skim_format_defaults (skim_format)`, [17](#)
`skim_tee`, [18](#)
`skim_to_list`, [18](#)
`skim_to_wide`, [19](#)
`skim_with`, [16](#), [19](#)
`skim_with_defaults (skim_with)`, [19](#)
`skimr (skimr-package)`, [3](#)
`skimr-package`, [3](#)
`sorted_count`, [20](#)
`summary.skim_df`, [21](#)

`ts_end`, [22](#)
`ts_start`, [22](#)