

Description

Applies descriptive statistics and hypothesis tests to data, and arranges the results for printing.

Usage

```
atable(x, ...)

## S3 method for class 'data.frame'
atable(
  x,
  target_cols,
  group_col = NULL,
  split_cols = NULL,
  format_to = atable_options("format_to"),
  drop_levels = TRUE,
  add_levels_for_NA = FALSE,
  blocks = NULL,
  add_margins = atable_options("add_margins"),
  indent_character = NULL,
  indent = atable_options("indent"),
  ...
)

## S3 method for class 'formula'
atable(formula, data, ...)
```

Arguments

- x** An object. If **x** is a `data.frame`, it must have unique and syntactically valid colnames, see [is_syntactically_valid_name](#). If **x** is a `formula`, then its format must be `target_cols ~ group_col | split_cols`. See other arguments for more details.
- ...** Passed from and to other methods. You can use the ellipsis **...** to modify `atable`: For example the default-statistics for numeric variables are `mean` and `sd`. To change these statistics pass a function to argument `statistics.numeric`, that calculates the statistics you prefer for your data.
See examples below how to modify `atable` by **...**.
Actually `statistics.numeric` is passed to `statistics` and thus documented there, but for convenience it also documented here.
Here is a list of the statistics and hypothesis tests that can be modified by **...**:
- `statistics.numeric`: Either `NULL` or a function. Default is `NULL`. If a function, then it will replace `atable:::statistics.numeric` when `atable` is called. The function must mimic `statistics`: see the help there.
 - `statistics.factor`: Analog to argument `statistics.numeric`.
 - `statistics.ordered`: Analog to argument `statistics.numeric`.

	<ul style="list-style-type: none"> • <code>two_sample_htest.numeric</code>: Either NULL or a function. Default is NULL. If a function, then it will replace <code>atable::two_sample_htest.numeric</code> when <code>atable</code> is called. The function must mimic <code>two_sample_htest</code>: see the help there. • <code>two_sample_htest.factor</code>: Analog to argument <code>two_sample_htest.numeric</code> • <code>two_sample_htest.ordered</code>: Analog to argument <code>two_sample_htest.numeric</code> • <code>multi_sample_htest.numeric</code>: Either NULL or a function. Default is NULL. If a function, then it will replace <code>atable::multi_sample_htest.numeric</code> when <code>atable</code> is called. The function must mimic <code>multi_sample_htest</code>: see the help there. • <code>multi_sample_htest.factor</code>: Analog to argument <code>multi_sample_htest.numeric</code> • <code>multi_sample_htest.ordered</code>: Analog to argument <code>multi_sample_htest.numeric</code> • <code>format_statistics.statistics_numeric</code>: Either NULL or a function. Default is NULL. If a function, then it will replace <code>atable::format_statistics.statistics_numeric</code>. The function must mimic <code>format_statistics</code>: see the help there. • <code>format_statistics.statistics_factor</code>: Analog to argument <code>format_statistics.statistics_factor</code> • <code>format_tests.htest</code>: Either NULL or a function. Default is NULL. If a function, then it will replace <code>format_tests.htest</code>. The function must mimic <code>format_tests</code>: see the help there. • <code>format_tests.htest_with_effect_size</code>: Analog to argument <code>format_tests.htest</code>
<code>target_cols</code>	A character vector containing some column names of <code>x</code> . Descriptive statistics and hypothesis test are applied to these columns depending on their class. The descriptive statistics are defined by <code>statistics</code> ; their representation and format by <code>format_statistics</code> . Hypothesis test are defined by <code>two_sample_htest</code> or <code>multi_sample_htest</code> (depending on the number of levels of <code>group_col</code>); their representation and format by <code>format_tests</code> . Note that <code>atable</code> always adds one name to <code>target_cols</code> to count the number of observations. This name is stored in <code>atable_options('colname_for_observation')</code> .
<code>group_col</code>	A character of length 1 containing a column of <code>x</code> or NULL. This column defines the groups that are compared by the hypothesis tests. <code>as.factor</code> is applied to this column before further processing. Default is NULL, meaning that no hypothesis tests are applied.
<code>split_cols</code>	A character vector containing some of <code>colnames(x)</code> or NULL. <code>x</code> is splitted by these columns before descriptive statistics and hypothesis test are applied. <code>as.factor</code> is applied to this column before further processing. Default is NULL, meaning that no splitting is done.
<code>format_to</code>	A character vector of length 1. Specifies the format of the output of <code>atable</code> . Possible values are 'Latex', 'Word', 'Raw', 'HTML', 'Console', 'markdown', 'md'. Default is defined in <code>atable_options</code> .
<code>drop_levels</code>	A logical. If TRUE then <code>droplevels</code> is called on <code>group_col</code> and <code>split_cols</code> before further processing. Default is TRUE.
<code>add_levels_for_NA</code>	If TRUE then <code>addNA</code> is called on <code>group_col</code> and <code>split_cols</code> before further processing. Default is FALSE.
<code>blocks</code>	NULL or a list. If <code>blocks</code> is a list, then the names of the list must be non-NA characters. The elements of the list must be some of <code>target_cols</code> , retaining the order of <code>target_cols</code> . Also in this case <code>split_cols</code> must be NULL as simultaneous blocking and splitting is not supported. Default is NULL, meaning that no blocking is done. Variables of a block are additionally indented. Blocking has

	no effect on the statistics, it only affects the indentation of the resulting table. See Examples.
add_margins	A logical with length one, TRUE or FALSE. Default is defined in atable_options as FALSE. When add_margins is TRUE and group_col is not NULL, a column containing the results of an ungrouped atable-call is added to the results. See Examples.
indent_character	A character with length 1 or NULL (default). This character is used for indentation in the resulting table. If NULL, then the value stored in atable_options is taken instead, depending on format_to. indent_data_frame does the indentation. See help there.
indent	A logical with length one, TRUE or FALSE. Default is defined in atable_options . Decides if indentation is done or not. The resulting table will have a different layout. If FALSE, then blocks is ignored.
formula	A formula of the form target_cols ~ group_col split_cols. The separates the group_col from the split_cols. Read the as 'given' as in a conditional probability $P(\text{target_cols} \text{split_cols})$. target_cols and split_cols may contain multiple names separated by +. group_col must be a single name if given. group_col and split_cols may be omitted and can be replaced by 1 in this case. The may also be omitted if no split_cols are given.
data	Passed to atable(x = data, ...).

Value

Results depend on format_to:

- 'Raw': A list with two elements called 'statistics_result' and 'tests_result', that contain all results of the descriptive statistics and the hypothesis tests. This format useful, when extracting a specific result unformatted (when format_to is not 'Raw' all numbers are also returned, but as rounded characters for printing and squeezed into a data.frame).
 - 'statistics_result': contains a data.frame with colnames c(split_cols, group_col, target_cols). split_cols and group_col retain their original values (now as factor). target_cols contain lists with the results of function [statistics](#). As the result of function statistics is also a list, target_cols contain lists of lists.
 - 'tests_result': has the same structure as 'statistics_result', but contains the results of [two_sample_htest](#) and [multi_sample_htest](#). Note that tests_result only exists if split_cols is not NULL.
- 'Word': A data.frame. Column atable_options('colname_for_group') contains all combinations of the levels of split_cols and the names of the results of function [format_statistics](#). Further columns are the levels of group_col the names of the results of format_tests. The levels of split_cols and the statistics are arranged vertically. The hypothesis test are arranged horizontally.
- 'HTML': Same as for format_to = 'Word' but a different character indents the first column.
#'
- 'Console': Meant for printing in the R console for interactive analysis. Same as for format_to = 'Word' but a different character indents the first column.
- 'Latex': Same as for format_to = 'Word' but a different character indents the first column and with [translate_to_LaTeX](#) applied afterwards.

Methods (by class)

- `atable(data.frame)`: applies descriptive statistics and hypothesis tests, arranges the results for printing.
- `atable(formula)`: parses the formula and passes its parts to `atable`.

Examples

```
# See vignette for more examples:  
# utils::vignette('atable_usage', package = 'atable')  
  
# Analyse datasets::ToothGrowth:  
# Length of tooth for each dose level and delivery method:  
atable::atable(datasets::ToothGrowth,  
  target_cols = 'len',  
  group_col = 'supp',  
  split_cols = 'dose',  
  format_to = 'Word')  
# Print in .docx with e.g. flextable:: regulartable and officer::body_add_table  
  
# Analyse datasets::ChickWeight:  
# Weight of chickens for each time point and diet:  
atable(weight ~ Diet | Time, datasets::ChickWeight, format_to = 'Latex')  
# Print as .pdf with e.g. Hmisc::latex  
  
# Analyse atable::test_data:  
atable(Numeric + Logical + Factor + Ordered ~ Group | Split1 + Split2,  
  atable::test_data, format_to = 'HTML')  
# Print as .html with e.g. knitr::kable and options(knitr.kable.NA = '')  
  
# Modify atable: calculate median and MAD for numeric variables  
new_stats <- function(x, ...){list(Median = median(x, na.rm = TRUE),  
  MAD = mad(x, na.rm = TRUE))}  
atable(atable::test_data,  
  target_cols = c('Numeric', 'Numeric2'),  
  statistics.numeric = new_stats,  
  format_to = 'Console')  
# Print in Console with format_to = 'Console'.  
  
# Analyse mtcars and add labels and units of via package Hmisc  
mtcars <- within(datasets::mtcars, {gear <- factor(gear)})  
# Add labels and units.  
attr(mtcars$mpg, 'alias') = 'Consumption [Miles (US)/ gallon]'  
Hmisc::label(mtcars$qsec) = 'Quarter Mile Time'  
units(mtcars$qsec) = 's'  
  
# apply atable  
atable::atable(mpg + hp + gear + qsec ~ cyl | vs,  
  mtcars,  
  format_to = 'Console')  
  
# Blocks  
# In datasets::mtcars the variables cyl, disp and mpg are related to the engine and am and gear are  
# related to the gearbox. So grouping them together is desireable.  
atable::atable(datasets::mtcars,  
  target_cols = c("cyl", "disp", "hp", "am", "gear", "qsec") ,  
  blocks = list("Engine" = c("cyl", "disp", "hp"),
```

```
    "Gearbox" = c("am", "gear")),
    format_to = "Console")
# Note that Variable qsec is not blocked and thus not indented.

# add_margins
atable::atable(atable::test_data,
               target_cols = "Numeric",
               group_col = "Group",
               split_cols = "Split1",
               add_margins = TRUE,
               format_to = "Console")
# The column 'Total' contains the results of the ungrouped atable-call:
# The number of observations is the sum of observations of the groups.
# The default of add_margins can be changed via atable_options.
```