

# Package ‘tablet’

April 30, 2022

**Type** Package

**Title** Tabulate Descriptive Statistics in Multiple Formats

**Version** 0.5.8

**Author** Tim Bergsma

**Maintainer** Tim Bergsma <bergsmat@gmail.com>

**BugReports** <https://github.com/bergsmat/tablet/issues>

**Description** Creates a table of descriptive statistics for factor and numeric columns in a data frame. Displays these by groups, if any. Highly customizable, with support for 'html' and 'pdf' provided by 'kableExtra'. Respects original column order, column labels, and factor level order. See ?tablet.data.frame and vignettes.

**License** GPL-3

**Encoding** UTF-8

**Imports** dplyr (>= 1.0.2), rlang, tidyr, kableExtra (>= 0.9.0), DT, spork (>= 0.2.2), magrittr, fs

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**Suggests** knitr, rmarkdown, yamlet (>= 0.8.7), boot, testthat, shiny, shinyFiles, haven, yaml, sortable, latexpdf (>= 0.1.7), tools, csv, xtable, shinyAce, R.utils

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-04-29 22:50:02 UTC

## R topics documented:

as_kable.tablet . . . . .	2
mesa . . . . .	4
tablet.data.frame . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

---

as_kable.tablet	<i>Coerce Tablet to Kable</i>
-----------------	-------------------------------

---

## Description

Renders a tablet. Calls [kbl](#) and implements special features like grouped columns.

## Usage

```
## S3 method for class 'tablet'
as_kable(
  x,
  ...,
  booktabs = TRUE,
  escape = FALSE,
  escape_latex = tablet::escape_latex,
  escape_html = function(x, ...) x,
  variable = " ",
  col.names = NA,
  linebreak = TRUE,
  align = "c",
  double_escape = FALSE,
  linebreaker = "\n",
  pack_rows = list(escape = escape)
)
```

## Arguments

<code>x</code>	<a href="#">tablet</a>
<code>...</code>	passed to <a href="#">kbl</a>
<code>booktabs</code>	passed to <a href="#">kbl</a>
<code>escape</code>	passed to <a href="#">kbl</a> ; defaults FALSE to allow header linebreaks
<code>escape_latex</code>	a function to pre-process column names and content if 'escape' is FALSE (e.g., manual escaping, latex only); default <a href="#">escape_latex</a>
<code>escape_html</code>	a function to pre-process column names and content if 'escape' is FALSE (e.g., manual escaping, html only)
<code>variable</code>	a column name for the variables
<code>col.names</code>	passed to <a href="#">kbl</a> after any linebreaking
<code>linebreak</code>	whether to invoke <a href="#">linebreak</a> for column names
<code>align</code>	passed to <a href="#">linebreak</a> for column names
<code>double_escape</code>	passed to <a href="#">linebreak</a> for column names
<code>linebreaker</code>	passed to <a href="#">linebreak</a> for column names in latex; for html, newline is replaced with <code>&lt;br&gt;</code>
<code>pack_rows</code>	named list passed to <a href="#">pack_rows</a> for finer control of variable names

## Details

See also [tablet.data.frame](#). Column `_tablet_name` must inherit 'character' and by default (in a latex render context) its values will eventually be processed by `escape_latex`. Thus, if `_tablet_name` is of class 'latex' it will be handled by method `escape_latex.latex` (which tries not to re-escape metacharacters).

## Value

like [kbl](#)

## Examples

```
library(boot)
library(dplyr)
library(magrittr)
library(haven)
library(yamlet)
library(spork)
library(melano) %>%
  select(-time, -year) %>%
  mutate(sex = factor(sex), ulcer = factor(ulcer)) %>%
  group_by(status) %>%
  tablet %>%
  as_kable

x <- system.file(package = 'tablet', 'shiny-examples/mesa/data/adsl.sas7bdat')
x %<>% read_sas %>% data.frame
decorations(x) # note weight in pounds
x %<>% mutate(weight = signif(digits = 3, weight * 2.2))

# calculate BMI by assuming all males are 1.75 m, all females 1.63 m
x %<>% mutate(height = ifelse(sex == 'F', 1.63, 1.75))
x %<>% mutate(bmi = signif(digits = 3, weight / (height^2)))
x %<>% filter(saffl == 'Y')
x %<>% select(trt01a, age, sex, weight, bmi)
x %<>% redecorate('
trt01a: [ Treatment, [ Placebo, TRT 10 mg, TRT 20 mg ] ]
age:    [ Age, year ]
sex:    [ Sex, [ Female: F, Male: M ] ]
weight: [ Body Weight, kg ]
bmi:    [ Index_body mass, kg/m^2 ]
')
x %<>% resolve
x %<>% group_by(trt01a)

x %>% tablet %>% as_kable

# supply default and unit-conditional latex titles
x %<>% modify(title = concatenate(as_latex(as_spork(c(.data$label))))))
x %<>% modify(
age, weight, bmi,
  title = concatenate(
```

```

    sep = '', # default ok in pdf
    as_latex(
      as_spork(
        c(.data$label, '(', .data$units, ')')
      )
    )
  )
)
x %>% tablet %>% as_kable

```

---

 mesa

*Drag-and-drop Descriptive Statistics*


---

### Description

Generate a table of descriptive statistics by selecting columns from a file. Currently supported formats include \*.xpt, \*.sas7bdat, and \*.csv. Launch the application using `mesa()` and use the interface to select a data file, such as 'mtcars.xpt' under 'examples/' (or select configuration file 'mtcars.conf' under 'examples/'). Then classify the columns of interest to generate the corresponding displays.

### Usage

```
mesa(launch.browser = TRUE, display.mode = "normal", ...)
```

### Arguments

```

launch.browser  passed to runApp
display.mode    passed to runApp
...             passed to runApp

```

### Details

Currently,

- \* xpt files are read using the defaults for [read.xport](#),
- \* sas7bdat files are read using the defaults for [read\\_sas](#), and
- \* csv files are read using the defaults for [as.csv](#).

If a file in the same directory has a corresponding base name but a .yaml extension, it is treated as metadata and an attempt is made to apply it to the internal version of the data. This file will not be over-written, but it WILL be constructed if missing. You can hand-edit it to supply metadata. See ?yamllet for format; see the Variables tab for an easy interface.

This is a metadata-driven application. Columns in the data that are \*not\* in the metadata will be ignored, and columns in the metadata that are \*not\* in the data will be constructed (maybe \*all\* of them).

The [mtcars](#) datasets in the 'examples' volume is from **datasets**.

**Value**

used for side effects: launches shiny application shinyWidgets',

---

tablet.data.frame	<i>Generate a Tablet for Data Frame</i>
-------------------	---

---

**Description**

Generates a 'tablet': a summary table of formatted statistics for factors (is.factor()) and numerics (is.numeric()) in *x*, with and without grouping variables (if present, see [group\\_by](#)). Column names represent finest level of grouping, distinguished by attribute 'nest' (the values of higher other groups, if any) along with the 'all' column for ungrouped statistics. Column attribute 'n' indicates relevant corresponding observation count. Input should not have column names beginning with '\_tablet'.

**Usage**

```
## S3 method for class 'data.frame'
tablet(
  x,
  ...,
  na.rm = FALSE,
  all = 'All',
  fun = list(
    sum ~ sum(x, na.rm = TRUE),
    pct ~ signif(digits = 3, sum / n * 100),
    ave ~ signif(digits = 3, mean(x, na.rm = TRUE)),
    std ~ signif(digits = 3, sd(x, na.rm = TRUE)),
    med ~ signif(digits = 3, median(x, na.rm = TRUE)),
    min ~ signif(digits = 3, min(x, na.rm = TRUE)),
    max ~ signif(digits = 3, max(x, na.rm = TRUE))
  ),
  fac = list(
    ` ` ~ sum + ' (' + pct + '%' + ')'
  ),
  num = list(
    `Mean (SD)` ~ ave + ' (' + std + ')',
    `Median (range)` ~ med + ' (' + min + ', ' + max + ')'
  ),
  lab = list(
    lab ~ name + '\n(N = ' + n + ')'
  ),
  na.rm_fac = na.rm,
  na.rm_num = na.rm,
  exclude_fac = NULL,
  exclude_name = NULL,
  all_levels = FALSE
)
```

**Arguments**

x	data.frame (possibly grouped)
...	substitute formulas for elements of fun, fac, num, lab
na.rm	whether to remove NA in general
all	a column name for ungrouped statistics; can have length zero to suppress ungrouped column
fun	default aggregate functions expressed as formulas
fac	a list of formulas to generate widgets for factors
num	a list of formulas to generate widgets for numerics
lab	a list of formulas to generate label attributes for columns (see details)
na.rm_fac	whether to drop NA 'factor' observations; passed to <a href="#">gather</a> as na.rm, interacts with exclude_fac
na.rm_num	whether to drop NA numeric observations; passed to <a href="#">gather</a> as na.rm
exclude_fac	which factor levels to exclude; see <a href="#">factor</a> (exclude)
exclude_name	whether to drop NA values of column name (for completeness); passed to <a href="#">gather</a>
all_levels	whether to supply records for unobserved levels

**Details**

Arguments 'fun', 'fac', 'num', and 'lab' are lists of two-sided formulas that are evaluated in an environment where '+' expresses concatenation (for character elements). The values of LHS should be unique across all four lists. 'fun' is a list of aggregate statistics that have access to N (number of original records), n (number of group members), and x (the numeric observations, or 1 for each factor value). Aggregate statistics generated by 'fun' are available for use in 'fac' and 'num' which create visualizations thereof ('widgets'). Column-specific attributes are available to elements of 'lab', including the special attribute name (the current column name). For 'lab' only, if the RHS succeeds, it becomes the label attribute of the corresponding output column. 'lab' is used here principally to support annotation of \*output\* columns; if \*input\* columns have attributes 'label' or 'title' (highest priority) those will have been already substituted for default column names at the appropriate positions in the output.

Missingness of observations (and to a lesser extent, levels of grouping variables) merits special consideration. Be aware that na.rm\_fac and na.rm\_num take their defaults from na.rm. Furthermore, na.rm\_fac may interact with exclude\_fac, which is passed to [factor](#) as exclude. The goal is to support all possible ways of expressing or ignoring missingness. That said, if aggregate functions are removing NA, the values of arguments beginning with 'na.rm' or 'exclude' may not matter.

Output includes the column `_tablet_name` which inherits character. Its values are typically the names of the original columns that were factor or numeric but not in groups(x). If the first of these had a label attribute or (priority) a title attribute with class 'latex', then `_tablet_name` is assigned the class 'latex' as well. It makes sense therefore to be consistent across input columns regarding the presence or not of a 'latex' label or title. By default, [as\\_kable.tablet](#) dispatches class-specific methods for [escape\\_latex](#).

**Value**

'table', with columns for each combination of groups, and:

<code>_tablet_name</code>	observation identifier: character, possibly 'latex', see details; has a <code>codelist</code> attribute the values of which are the original column names
<code>_tablet_level</code>	factor level (or special value 'numeric' for numerics)
<code>_tablet_stat</code>	the LHS of formulas in 'fac' and 'num'
All (or value of 'all' argument)	ungrouped results
<code>_tablet_sort</code>	sorting column

**See Also**

[as\\_kable.tablet](#)

**Examples**

```
library(boot)
library(dplyr)
library(magrittr)
melanoma %>%
  select(-time, -year) %>%
  mutate(sex = factor(sex), ulcer = factor(ulcer)) %>%
  group_by(status) %>%
  tablet
```

# Index

`as.csv`, [4](#)  
`as_kable.tablet`, [2](#), [6](#), [7](#)  
  
`escape_latex`, [2](#), [6](#)  
`escape_latex.latex`, [3](#)  
  
`factor`, [6](#)  
  
`gather`, [6](#)  
`group_by`, [5](#)  
  
`kbl`, [2](#), [3](#)  
  
`linebreak`, [2](#)  
  
`mesa`, [4](#)  
`mtcars`, [4](#)  
  
`pack_rows`, [2](#)  
  
`read.xport`, [4](#)  
`read_sas`, [4](#)  
`runApp`, [4](#)  
  
`tablet`, [2](#)  
`tablet.data.frame`, [3](#), [5](#)