

Package ‘ttr’

July 22, 2025

Type Package

Version 1.0

Date 2021-05-06

Title The Table Tool

URL <https://github.com/benjaminrich/ttr>

BugReports <https://github.com/benjaminrich/ttr/issues>

Description Create structured, formatted HTML tables of in a flexible and convenient way.

License GPL-3

Imports stats,Formula,knitr,htmltools

Suggests rmarkdown,table1,magrittr

VignetteBuilder knitr

Encoding UTF-8

RoxxygenNote 7.1.1

NeedsCompilation no

Author Benjamin Rich [aut, cre]

Maintainer Benjamin Rich <mail@benjaminrich.net>

Repository CRAN

Date/Publication 2021-05-07 11:50:02 UTC

Contents

knit_print.ttr	2
print.ttr	2
ttr	3

Index	7
--------------	----------

knit_print.ttt	<i>Method for printing in a knitr context</i>
----------------	---

Description

Method for printing in a knitr context

Usage

```
## S3 method for class 'ttt'
knit_print(x, ..., theme = getOption("ttt.theme"))
```

Arguments

x	An object returned by <code>ttt</code> .
...	Further arguments passed on to <code>knitr::knit_print</code> .
theme	A theme (either "default" or "booktabs").

Value

Returns a character string. See `knitr::knit_print` for how this value is used.

print.ttt	<i>Print ttt object</i>
-----------	-------------------------

Description

Print ttt object

Usage

```
## S3 method for class 'ttt'
print(x, ..., theme = getOption("ttt.theme"))
```

Arguments

x	An object returned by <code>ttt</code> .
...	Further arguments passed on to other print methods.
theme	A theme (either "default" or "booktabs").

Details

In an interactive context, the rendered table will be displayed in a web browser. Otherwise, the HTML code will be printed as text.

Value

Returns x invisibly.

ttr

Formatted tables the easy way

Description

ttr stands for “The Table Tool” (or, if you prefer, “Tables! Tables! Tables!”). It allows you to creates formatted HTML tables of in a flexible and convenient way.

Usage

```
ttr(x, ...)  
  
## S3 method for class 'data.frame'  
ttr(  
  x,  
  formula,  
  ...,  
  render,  
  lab,  
  caption,  
  footnote,  
  expand.along = c("rows", "columns"),  
  drop = c("both", "rows", "columns", "none"),  
  collapse.cells = TRUE,  
  topclass = NULL,  
  id = NULL,  
  css = NULL,  
  row.names = TRUE  
)  
  
## S3 method for class 'formula'  
ttr(  
  x,  
  data,  
  ...,  
  render,  
  lab,  
  caption,  
  footnote,  
  expand.along = c("rows", "columns"),  
  drop = c("both", "rows", "columns", "none"),  
  collapse.cells = TRUE,  
  topclass = NULL,
```

```

    id = NULL,
    css = NULL
)

## S3 method for class 'numeric'
ttt(
  x,
  rowvars,
  colvars,
  ...,
  render,
  lab,
  caption,
  footnote,
  expand.along = c("rows", "columns"),
  drop = c("both", "rows", "columns", "none"),
  collapse.cells = TRUE,
  topclass = NULL,
  id = NULL,
  css = NULL
)

## S3 method for class 'ftable'
ttt(
  x,
  text = matrix(as.character(x), nrow(x)),
  lab,
  caption,
  footnote,
  drop = c("both", "rows", "columns", "none"),
  collapse.cells = TRUE,
  html.class = NULL,
  topclass = NULL,
  id = NULL,
  css = NULL,
  ...
)

```

Arguments

<code>x</code>	An object.
<code>...</code>	Additional arguments passed to render.
<code>formula</code>	A three-part formula of the form $v \sim r1 + r2 \sim c1 + c2$ where v specifies a column of values, while $r1, r2$ specify row variables and $c1, c2$ column variables for splitting the values.
<code>render</code>	A function to render the contents of each cell to character data.
<code>lab</code>	Specify the contents of an extra table cell spanning over all column labels.

<code>caption</code>	A character string to be added as a caption to the table. The default is to omit the caption.
<code>footnote</code>	A character string to be added as a footnote to the table. The default is to omit the footnote.
<code>expand.along</code>	Specify the direction to expand the table when <code>render</code> returns a (named) vector.
<code>drop</code>	If TRUE (the default), rows and columns with zero counts will be dropped.
<code>collapse.cells</code>	If TRUE (the default), row/column header cells will be collapsed (merged) where appropriate.
<code>topclass</code>	A character string to be used as <code>class</code> attribute for the top-level <code><table></code> element.
<code>id</code>	A character string to be used as <code>id</code> attribute for the top-level <code><table></code> element.
<code>css</code>	A character string containing CSS code to be added before the top-level <code><table></code> element.
<code>row.names</code>	If TRUE (the default), row names will be shown in the first column of the table. Set to FALSE to suppress row names. Only applies when displaying whole <code>data.frame</code> .
<code>data</code>	A <code>data.frame</code> .
<code>rowvars</code>	A list of row variables for splitting the data.
<code>colvars</code>	A list of column variables for splitting the data.
<code>text</code>	A character matrix containing the textual content of each table cell.
<code>html.class</code>	A character matrix with the same dimensions as <code>text</code> specifying a class attribute for the corresponding <code><td></code> element.

Value

A character which contains an HTML table fragment. It has additional class attributes that cause it to be displayed in a browser in an interactive context, and rendered as HTML in a knitr context.

Methods (by class)

- `data.frame`: The `data.frame` method.
- `formula`: The `formula` method.
- `numeric`: The `numeric` method.
- `fable`: The `fable` method.

Examples

```
# mtcars examples
tft(mtcars)
tft(mtcars, mpg ~ gear | cyl, lab="Cylinders")
tft(mpg ~ gear | cyl, data=mtcars, lab="Cylinders")
tft(row.names(mtcars) ~ gear | cyl, data=mtcars,
  render=paste, collapse="<br/>", lab="Cylinders")

# OrchardSprays examples
```

```
ttr(head(OrchardSprays, 12))
ttr(head(OrchardSprays, 12), row.names=FALSE)
ttr(treatment ~ rowpos | colpos, data=OrchardSprays, lab="colpos")
ttr(paste(treatment, decrease, sep="<br/>") ~ rowpos | colpos, data=OrchardSprays, lab="colpos")

rndr.meansd <- function(x) formatC(c(Mean=mean(x), SD=sd(x)), digits=3)
ttr(decrease ~ treatment, data=OrchardSprays, render=rndr.meansd, expand.along="rows")
ttr(decrease ~ treatment, data=OrchardSprays, render=rndr.meansd, expand.along="columns")

# ToothGrowth examples
ttr(len ~ dose | supp, data=ToothGrowth, lab="Mean (SD)",
    render=function(x) sprintf("%0.3g (%0.3g)", mean(x), sd(x)))

ttr(len ~ dose | supp, data=ToothGrowth, lab="Supplement Type",
    render=rndr.meansd)

ttr(len ~ dose | supp, data=ToothGrowth, lab="Supplement Type",
    render=rndr.meansd, expand.along="columns")
```

Index

* **utilities**

 ttt, [3](#)

knit_print.ttt, [2](#)

print.ttt, [2](#)

ttt, [2](#), [3](#)