



Level up your tables with tableHTML in R

Poster sponsored by



tableHTML - basics

- This is a package for building CSS-ible HTML tables in an easy and intuitive way.
- These are compatible with any application that accepts HTML (e.g. shiny, rmarkdown).
- The main function tableHTML will convert a data.frame or matrix or any other object that can be converted into a data.frame into an HTML table
- Using the function on R Studio will print the table on the viewer otherwise it will use the default browser.
- The default tables are built without any CSS in order to allow for full flexibility in design
- The package has been developed so that the functions are chained with the %>% (pipe) operator which comes from the magrittr package
- The tables can be exported as images using tableHTML_to_image()

```
# install.packages('tableHTML')
```

```
library(tableHTML)
```

```
my_data <- mtcars %>%  
  head() %>%  
  '['(1:5)
```

```
tableHTML <- my_data %>%  
  tableHTML(rownames = TRUE,  
            round = 1,  
            replace_NA = '')
```

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.8
Hornet 4 Drive	21.4	6	258	110	3.1
Hornet Sportabout	18.7	8	360	175	3.1
Valiant	18.1	6	225	105	2.8

Adding themes

- The package was built to give the absolute freedom to users to design their own HTML tables the way the want
- However, we have included three categories of themes to give the users a way to build nice looking tables very fast

```
tableHTML %>%  
  add_theme('scientific')
```

```
tableHTML %>%  
  add_theme('rshiny-blue')
```

```
tableHTML %>%  
  add_theme('colorize')
```

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.8
Hornet 4 Drive	21.4	6	258	110	3.1
Hornet Sportabout	18.7	8	360	175	3.1
Valiant	18.1	6	225	105	2.8

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.8
Hornet 4 Drive	21.4	6	258	110	3.1
Hornet Sportabout	18.7	8	360	175	3.1
Valiant	18.1	6	225	105	2.8

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.8
Hornet 4 Drive	21.4	6	258	110	3.1
Hornet Sportabout	18.7	8	360	175	3.1
Valiant	18.1	6	225	105	2.8

Basic adjustment options in tableHTML()

- The tabelHTML() functions can easily be used to adjust the table
- You can add / remove rownames
- Add a specific CSS-class
- Change the width of the columns
- Add second_headers and row_groups
- Round numeric columns in the table
- Provide a replacement string for NA values
- Add a caption / footer
- Remove the borders

This is a caption					
	col1		col2		
	mpg	cyl	disp	hp	drat
Group 1	21	6	160	110	3.9
	21	6	160	110	3.9
	22.8	4	108	93	3.8
	21.4	6	258	110	3.1
Group 2	18.7	8	360	175	3.1
	18.1	6	225	105	2.8

This is a footer

```
my_data %>%  
  tableHTML(rownames = FALSE,  
            class = 'my_table',  
            widths = rep(80, 6),  
            second_headers = list(c(1, 2, 3),  
                                  c('', 'col1', 'col2')),  
            row_groups = list(c(3, 3),  
                              c('Group 1', 'Group 2')),  
            round = 1,  
            replace_NA = '',  
            caption = 'This is a caption',  
            footer = 'This is a footer')
```

add_css_* family of functions

- These functions add inline CSS to the HTML table
- You can chain as many functions as you want when building your HTML table
- The CSS code is added within the HTML table
- This is ok for simple table styling but for more complex CSS styles it would still be good practice to use a separate css file

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15
Valiant	18.1	6	225	105	2.76

```
my_data %>%  
  tableHTML(widths = c(140, rep(45, 5))) %>%  
  add_css_header(css = list(c('height', 'background-color'),  
                             c('30px', '#e6e6e6')),  
                headers = 1:6) %>%  
  add_css_row(css = list('background-color', '#e6f0ff'),  
             rows = odd(1:7)) %>%  
  add_css_column(css = list('text-align', 'center'),  
                columns = names(mtcars)) %>%  
  add_css_conditional_column(conditional = 'max',  
                             css = list('color', 'red'),  
                             columns = 1)
```

Shiny

- Using tableHTML with shiny is very easy
- The implementation is simple since internally tableHTML uses htmltools::HTML to convert the table into an HTML object
- Simply using render_tableHTML and tableHTML_output will be enough

```
library(shiny)  
shinyApp(  
  ui = fluidPage(  
    fluidRow(  
      column(width = 1),  
      tableHTML_output('mytable'))  
    ),  
  server = function(input, output) {  
    output$mytable <- render_tableHTML(  
      tableHTML(my_data)  
    )  
  })
```

	mpg	cyl	disp	hp	drat
Mazda RX4	21	6	160	110	3.9
Mazda RX4 Wag	21	6	160	110	3.9
Datsun 710	22.8	4	108	93	3.85
Hornet 4 Drive	21.4	6	258	110	3.08
Hornet Sportabout	18.7	8	360	175	3.15
Valiant	18.1	6	225	105	2.76