

# Package: mmtable2 (via r-universe)

October 21, 2024

**Type** Package

**Title** Create and combine tables with a ggplot2/patchwork syntax

**Version** 0.1.3

**Author** Ian Moran

**Maintainer** Ian Moran <ian.moran11@gmail.com>

**Description** Add headers using data from a column in your data frame.  
Combine tables with +, / and \* operators. Output tables in gt  
package format.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 2.10)

**Imports** dplyr, tidyr, purrr, gt, magrittr, broom, tibble, stringr,  
htmltools, xml2, rlang, forcats, zoo,

**URL** <https://github.com/ianmoran11/mmtable2>

**BugReports** <https://github.com/ianmoran11/mmtable2/issues>

**Suggests** testthat (>= 2.1.0), webshot, gapminder, covr, knitr,  
rmarkdown, margins, tidyverse, titanic, RColorBrewer,  
colorspace, scales

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**Repository** <https://staffanbetner.r-universe.dev>

**RemoteUrl** <https://github.com/ianmoran11/mmtable2>

**RemoteRef** HEAD

**RemoteSha** 02ffc53ca05baef7c3216afb82068e8762de3323

## Contents

|                                   |    |
|-----------------------------------|----|
| *.mmtable . . . . .               | 3  |
| +.mmtable . . . . .               | 3  |
| /.mmtable . . . . .               | 4  |
| add_cells_format . . . . .        | 4  |
| add_header_format . . . . .       | 5  |
| add_header_left . . . . .         | 6  |
| add_header_left_top . . . . .     | 7  |
| add_header_top . . . . .          | 8  |
| add_header_top_left . . . . .     | 9  |
| add_spanner . . . . .             | 10 |
| add_table_format . . . . .        | 11 |
| add_table_source_note . . . . .   | 12 |
| apply_format . . . . .            | 12 |
| apply_formats . . . . .           | 13 |
| apply_if . . . . .                | 13 |
| cells_format . . . . .            | 14 |
| diagonalize . . . . .             | 15 |
| extract_max_rects . . . . .       | 15 |
| format_a_loc . . . . .            | 16 |
| gapminder_mm . . . . .            | 16 |
| gen_random_ids . . . . .          | 17 |
| get_locations . . . . .           | 17 |
| get_max_area_bf . . . . .         | 18 |
| get_row_header_names . . . . .    | 18 |
| get_spanner_html_text . . . . .   | 19 |
| header_format . . . . .           | 19 |
| header_left . . . . .             | 20 |
| header_left_top . . . . .         | 21 |
| header_merged_cols . . . . .      | 22 |
| header_top . . . . .              | 22 |
| header_top_left . . . . .         | 23 |
| knit_print.mmtable . . . . .      | 24 |
| mmtable . . . . .                 | 24 |
| print.mmtable . . . . .           | 25 |
| set_class . . . . .               | 26 |
| single_cells_to_vector . . . . .  | 26 |
| sort_and_select_columns . . . . . | 27 |
| spannerize . . . . .              | 27 |
| spread_col_headers . . . . .      | 28 |
| student_df . . . . .              | 28 |
| student_df_1 . . . . .            | 29 |
| student_df_2 . . . . .            | 29 |
| style_first_col_header . . . . .  | 30 |
| style_spanner . . . . .           | 30 |
| table1 . . . . .                  | 31 |
| table_constructor . . . . .       | 31 |

|  |    |
|--|----|
| <code>*.mmtable</code>                   | 3  |
| <code>table_format</code> . . . . .      | 32 |
| <code>table_source_note</code> . . . . . | 33 |
| <code>table_title</code> . . . . .       | 33 |
| <code>transfer_spanner</code> . . . . .  | 34 |

**Index** 35

`*.mmtable`                    *Integrate two tables horizontally*

**Description**

Integrate two tables horizontally

**Usage**

```
## S3 method for class 'mmtable'
mmtable1 * mmtable2
```

**Arguments**

```
mmtable1            an mmtable
mmtable2            an mmtable
```

**Value**

an mmtable

`+.mmtable`                    *Integrate two tables horizontally*

**Description**

Integrate two tables horizontally

**Usage**

```
## S3 method for class 'mmtable'
mmtable1 + mmtable2
```

**Arguments**

```
mmtable1            an mmtable
mmtable2            an mmtable
```

**Value**

an mmtable

---

|                        |  |
|------------------------|--|
| <code>/.mmtable</code> | <i>Concatenate tables horizontally</i> |
|------------------------|--|

---

**Description**

Concatenate tables horizontally

**Usage**

```
## S3 method for class 'mmtable'  
e1 / e2
```

**Arguments**

|                 |            |
|-----------------|------------|
| <code>e1</code> | an mmtable |
| <code>e2</code> | an mmtable |

**Value**

an mmtable

---

|                               |                             |
|-------------------------------|-----------------------------|
| <code>add_cells_format</code> | <i>Add cells formatting</i> |
|-------------------------------|-----------------------------|

---

**Description**

Add cells formatting

**Usage**

```
add_cells_format(mmtable, ...)
```

**Arguments**

|                      |   |
|----------------------|---|
| <code>mmtable</code> | an mmtable object                           |
| <code>...</code>     | arguments for <code>add_cells_format</code> |

**Value**

an mmtable

**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top",color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

|                   |                                   |
|-------------------|-----------------------------------|
| add_header_format | <i>Add formatting to a header</i> |
|-------------------|-----------------------------------|

---

**Description**

Add formatting to a header

**Usage**

```
add_header_format(mmtable, ...)
```

**Arguments**

|         |                                 |
|---------|---------------------------------|
| mmtable | an mmtable object               |
| ...     | arguments for add_header_format |

**Value**

an mmtable

**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top",color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

|                 |                          |
|-----------------|--------------------------|
| add_header_left | <i>Add a left header</i> |
|-----------------|--------------------------|

---

**Description**

Add a left header

**Usage**

```
add_header_left(mmtable, ...)
```

**Arguments**

|         |                           |
|---------|---------------------------|
| mmtable | an mmtable object         |
| ...     | arguments for header_left |

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top",color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

add\_header\_left\_top    *Add a left top header*

---

**Description**

Add a left top header

**Usage**

```
add_header_left_top(mmtable, ...)
```

**Arguments**

mmtable            an mmtable object  
 ...                arguments for header\_left\_top

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top", color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

|                |                         |
|----------------|-------------------------|
| add_header_top | <i>Add a top header</i> |
|----------------|-------------------------|

---

**Description**

Add a top header

**Usage**

```
add_header_top(mmtable, ...)
```

**Arguments**

|         |                          |
|---------|--------------------------|
| mmtable | an mmtable object        |
| ...     | arguments for header_top |

**Value**

an mmtable object



**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top",color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

add\_header\_top\_left    *Add a top left header*

---

**Description**

Add a top left header

**Usage**

```
add_header_top_left(mmtable, ...)
```

**Arguments**

mmtable            an mmtable object  
 ...                arguments for header\_top\_left

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top",color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

 add\_spanner

*Add a spanner to a table*


---

**Description**

Add a spanner to a table

**Usage**

```
add_spanner(gm_table2, spanner_list)
```

**Arguments**

```
gm_table2      a mmtable
spanner_list   a list of spanner calls
```

**Value**

mmtable

---

|                  |                                  |
|------------------|----------------------------------|
| add_table_format | <i>Add formatting to a table</i> |
|------------------|----------------------------------|

---

**Description**

Add formatting to a table

**Usage**

```
add_table_format(mmtable, ...)
```

**Arguments**

|         |                            |
|---------|----------------------------|
| mmtable | an mmtable object          |
| ...     | arguments for table_format |

**Value**

an mmtable

**Examples**

```
## Not run:

library(tidyverse)

gm_table_piped <-
  gapminder_mm %>%
  filter(var != "Life expectancy") %>%
  mmtable(cells = value, use_default_formats = T) %>%
  add_header_top(year) %>%
  add_header_left(country) %>%
  add_header_top_left(var) %>%
  add_header_left_top(continent) %>%
  add_cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) %>%
  add_header_format(header = year, style = list(cell_text(align = "right"))) %>%
  add_header_format("all_cols", style = list(cell_text(weight = "bolder"))) %>%
  add_header_format("all_rows", style = list(cell_text(weight = "bolder"))) %>%
  add_table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top", color = "grey"))) %>%
  add_table_source_note(source_note = "Excerpt of the Gapminder dataset." )

gm_table_piped

## End(Not run)
```

---

`add_table_source_note` *Add a table source note*

---

**Description**

Add a table source note

**Usage**

```
add_table_source_note(mmtable, ...)
```

**Arguments**

|                      |   |
|----------------------|---|
| <code>mmtable</code> | an mmtable object                             |
| <code>...</code>     | arguments of <code>gt::tab_source_note</code> |

**Value**

an mmtable

---

`apply_format` *Apply formatting to a mmtable*

---

**Description**

Apply formatting to a mmtable

**Usage**

```
apply_format(mmtable, format_list)
```

**Arguments**

|                          |   |
|--------------------------|---|
| <code>mmtable</code>     | an mmtable  |
| <code>format_list</code> | a list of styles of the same format as the <code>gt::tab_style</code> list argument <a href="https://gt.rstudio.com/reference/tab_style.html">https://gt.rstudio.com/reference/tab_style.html</a> |

**Value**

mmtable

---

|               |   |
|---------------|---|
| apply_formats | <i>Apply list of formatting commands to a mmtable</i> |
|---------------|---|

---

**Description**

Apply list of formatting commands to a mmtable

**Usage**

```
apply_formats(mmtable)
```

**Arguments**

|         |            |
|---------|------------|
| mmtable | an mmtable |
|---------|------------|

**Value**

mmtable

---

|          |                                       |
|----------|---------------------------------------|
| apply_if | <i>Conditionally apply a function</i> |
|----------|---------------------------------------|

---

**Description**

Conditionally apply a function

**Usage**

```
apply_if(predicate, func, data, ...)
```

**Arguments**

|           |  |
|-----------|--|
| predicate | a predicate  |
| func      | a function to apply                                  |
| data      | data used by the function or otherwise returned      |
| ...       | other objects involved in evaluation of the function |

**Value**

data

---

|              |                           |
|--------------|---------------------------|
| cells_format | <i>Format table cells</i> |
|--------------|---------------------------|

---

## Description

Format table cells

## Usage

```
cells_format(cell_predicate, ...)
```

## Arguments

`cell_predicate` an expression on the source data frame identifying which cells should be formatted.

`...` dots.

## Value

format list

## Examples

```
## Not run:
gm_table_formatted <-
  gapminder_mm %>%
  dplyr::filter(var != "Life expectancy") %>%
  mtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  cells_format(cell_predicate = T, style = list(cell_text(align = "right"))) +
  header_format(header = year, style = list(cell_text(align = "right"))) +
  header_format("all_cols", style = list(cell_text(weight = "bolder"))) +
  header_format("all_rows", style = list(cell_text(weight = "bolder"))) +
  table_format(
    locations = cells_body(rows = c(1,3,5,7,9,11)),
    style = list(cell_borders(sides = "top",color = "grey"))) +
  table_source_note(source_note = "Excerpt of the Gapminder data" )

## End(Not run)
```

---

|             |   |
|-------------|---|
| diagonalize | <i>Remove adjacent repeats in a dataframe</i> |
|-------------|---|

---

**Description**

Remove adjacent repeats in a dataframe

**Usage**

```
diagonalize(df, var_char)
```

**Arguments**

|          |                               |
|----------|-------------------------------|
| df       | a data frame                  |
| var_char | the column to be diagonalized |

**Value**

data frame

---

|                   |  |
|-------------------|--|
| extract_max_rects | <i>Extract the largest binary rectary from a list of rows and cols</i> |
|-------------------|--|

---

**Description**

Extract the largest binary rectary from a list of rows and cols

**Usage**

```
extract_max_rects(extraction_list)
```

**Arguments**

|                 |   |
|-----------------|---|
| extraction_list | a list containing a dataframe of rows and cols and a list of already extracted rows and cols. |
|-----------------|---|

**Value**

extraction\_list

---

|              |                                       |
|--------------|---------------------------------------|
| format_a_loc | <i>Apply formatting to a location</i> |
|--------------|---------------------------------------|

---

**Description**

Apply formatting to a location

**Usage**

```
format_a_loc(mtable, loc, format_list)
```

**Arguments**

|             |                                       |
|-------------|---------------------------------------|
| mtable      | an mtable                             |
| loc         | a list containing a row and col items |
| format_list | a list of formatting commands         |

**Value**

format list

---

|              |                        |
|--------------|------------------------|
| gapminder_mm | <i>Gapminder data.</i> |
|--------------|------------------------|

---

**Description**

Excerpt of the Gapminder data on life expectancy, GDP per capita, and population by country.

**Usage**

```
gapminder_mm
```

**Format**

A data frame with 53940 rows and 10 variables:

|                  |                                      |
|------------------|--------------------------------------|
| <b>continent</b> | price, in US dollars                 |
| <b>country</b>   | weight of the diamond, in carats     |
| <b>value</b>     | weight of the diamond, in carats     |
| <b>var</b>       | weight of the diamond, in carats     |
| <b>year</b>      | weight of the diamond, in carats ... |
| continent        | country value var year               |

**Source**

<http://www.diamondse.info/>



---

|                |                                 |
|----------------|---------------------------------|
| gen_random_ids | <i>Generate a random string</i> |
|----------------|---------------------------------|

---

**Description**

Generate a random string

**Usage**

```
gen_random_ids(n, length)
```

**Arguments**

|        |               |
|--------|---------------|
| n      | number of ids |
| length | length of id  |

**Value**

format list

---

|               |  |
|---------------|--|
| get_locations | <i>Identify which locations are to be formatted.</i> |
|---------------|--|

---

**Description**

Identify which locations are to be formatted.

**Usage**

```
get_locations(  
  mmtable,  
  header = NULL,  
  func,  
  cell_predicate = NULL,  
  scope = "cell"  
)
```

**Arguments**

|                |  |
|----------------|--|
| mmtable        | an mmtable   |
| header         | header from original data  |
| func           | the original formatting function applied   |
| cell_predicate | predicate to determine locations   |
| scope          | string determining how widely formatting will apply. Options include "cell", "headers" and "table" |

**Value**

locations

---

|                 |  |
|-----------------|--|
| get_max_area_bf | <i>An algorithm to identify the largest rectangle in a histogram</i> |
|-----------------|--|

---

**Description**

An algorithm to identify the largest rectangle in a histogram

**Usage**

```
get_max_area_bf(heights)
```

**Arguments**

heights            a list of the heights of the histogram.

**Value**

list

---

|                      |                              |
|----------------------|------------------------------|
| get_row_header_names | <i>Get row header names.</i> |
|----------------------|------------------------------|

---

**Description**

Get row header names.

**Usage**

```
get_row_header_names(df, col_header_df_01)
```

**Arguments**

df                    a dataframe  
 col\_header\_df\_01    a dataframe of col headers

**Value**

a vector of names

---

get\_spanner\_html\_text *Get the html associated with a spanner*

---

**Description**

Get the html associated with a spanner

**Usage**

```
get_spanner_html_text(table)
```

**Arguments**

table            an mtable

**Value**

html

---

header\_format        *Format the cells of a header.*

---

**Description**

Format the cells of a header.

**Usage**

```
header_format(header, ...)
```

**Arguments**

header            the name of a variable that is currently a header in the table

...                a list of styles, with format required by `gt::tab_style` style argument. [https://gt.rstudio.com/reference/tab\\_s](https://gt.rstudio.com/reference/tab_s)

**Value**

format list

---

|             |                          |
|-------------|--------------------------|
| header_left | <i>Add a left header</i> |
|-------------|--------------------------|

---

**Description**

Add a left header

**Usage**

```
header_left(variable)
```

**Arguments**

variable            a column of the data frame from which header values will be constructed

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

row_list <- cells_body(rows = c(1,3,5,7,9,11))
style_list <- list(cell_borders(sides = "top",color = "grey"))
gm_df <- gapminder_mm %>% dplyr::filter(var != "Life expectancy")

gm_table <-
  gm_df %>%
  mmtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  table_format(
    locations = row_list,
    style = style_list)

## End(Not run)
```

---

|                 |                              |
|-----------------|------------------------------|
| header_left_top | <i>Add a left top header</i> |
|-----------------|------------------------------|

---

**Description**

Add a left top header

**Usage**

```
header_left_top(variable)
```

**Arguments**

variable            a column of the data frame from which header values will be constructed

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

row_list <- cells_body(rows = c(1,3,5,7,9,11))
style_list <- list(cell_borders(sides = "top",color = "grey"))
gm_df <- gapminder_mm %>% dplyr::filter(var != "Life expectancy")

gm_table <-
  gm_df %>%
  mmtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  table_format(
    locations = row_list,
    style = style_list)

## End(Not run)
```

---

|                    |                             |
|--------------------|-----------------------------|
| header_merged_cols | <i>Merge header columns</i> |
|--------------------|-----------------------------|

---

**Description**

Merge header columns

**Usage**

```
header_merged_cols(...)
```

**Arguments**

...           formatting instructions

**Value**

format list

---

|            |                         |
|------------|-------------------------|
| header_top | <i>Add a top header</i> |
|------------|-------------------------|

---

**Description**

Add a top header

**Usage**

```
header_top(variable)
```

**Arguments**

variable       a column of the data frame from which header values will be constructed

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

row_list <- cells_body(rows = c(1,3,5,7,9,11))
style_list <- list(cell_borders(sides = "top",color = "grey"))
gm_df <- gapminder_mm %>% dplyr::filter(var != "Life expectancy")

gm_table <-
  gm_df %>%
  mmtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  table_format(
    locations = row_list,
    style = style_list)

## End(Not run)
```

---

|                 |                              |
|-----------------|------------------------------|
| header_top_left | <i>Add a top left header</i> |
|-----------------|------------------------------|

---

**Description**

Add a top left header

**Usage**

```
header_top_left(variable)
```

**Arguments**

`variable` a column of the data frame from which header values will be constructed

**Value**

an mmtable object

**Examples**

```
## Not run:

library(tidyverse)

row_list <- cells_body(rows = c(1,3,5,7,9,11))
style_list <- list(cell_borders(sides = "top",color = "grey"))
```

```

gm_df <- gapminder_mm %>% dplyr::filter(var != "Life expectancy")

gm_table <-
  gm_df %>%
  mmtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  table_format(
    locations = row_list,
    style = style_list)

## End(Not run)

```

---

knit\_print.mmtable     *Print an mmtable object*

---

### Description

Print an mmtable object

### Usage

```
knit_print.mmtable(x, ...)
```

### Arguments

|     |              |
|-----|--------------|
| x   | an mmtable   |
| ... | other things |

---

mmtable     *Create an mmtable object*

---

### Description

Create an mmtable object

### Usage

```

mmtable(
  data,
  cells,
  table_name = NULL,
  use_default_formats = TRUE,
  table_data = NULL
)

```



**Arguments**

|                     |  |
|---------------------|--|
| data                | a data frame   |
| cells               | the name of the column whose values will form the data cells.  |
| table_name          | the name of the table - used for labeling when joined with another table - randomly generated if not provided. |
| use_default_formats | determines whether or not the table will have default formatting applied. Default value = TRUE.                |
| table_data          | deprecated. Use 'cells' instead.   |

**Value**

mmtable

**Examples**

```
## Not run:

library(tidyverse)

row_list <- cells_body(rows = c(1,3,5,7,9,11))
style_list <- list(cell_borders(sides = "top", color = "grey"))
gm_df <- gapminder_mm %>% dplyr::filter(var != "Life expectancy")

gm_table <-
  gm_df %>%
  mmtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  table_format(
    locations = row_list,
    style = style_list)

## End(Not run)
```

---

print.mmtable                      *Print an mmtable object*

---

**Description**

Print an mmtable object

**Usage**

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

**Arguments**

|     |              |
|-----|--------------|
| x   | an mmtable   |
| ... | other things |

---

|           |                                   |
|-----------|-----------------------------------|
| set_class | <i>Set the class of an object</i> |
|-----------|-----------------------------------|

---

**Description**

Set the class of an object

**Usage**

```
set_class(object, class)
```

**Arguments**

|        |                     |
|--------|---------------------|
| object | a object            |
| class  | new class of object |

**Value**

mmtable

---

|                        |   |
|------------------------|---|
| single_cells_to_vector | <i>Convert a list of cells to a list of cell ranges</i> |
|------------------------|---|

---

**Description**

Convert a list of cells to a list of cell ranges

**Usage**

```
single_cells_to_vector(row_col_list)
```

**Arguments**

|              |               |
|--------------|---------------|
| row_col_list | list of cells |
|--------------|---------------|

---

sort\_and\_select\_columns  
*Sort and select columns*

---

**Description**

Sort and select columns

**Usage**

```
sort_and_select_columns(data, ...)
```

**Arguments**

|      |                                       |
|------|---------------------------------------|
| data | a data frame                          |
| ...  | list of data to load into environment |

**Value**

mmtable

---

spannerize *Convert header to spanner*

---

**Description**

Convert header to spanner

**Usage**

```
spannerize(gm_table2, n)
```

**Arguments**

|           |                    |
|-----------|--------------------|
| gm_table2 | a mmtable          |
| n         | number of spanners |

**Value**

mmtable

---

|                    |                               |
|--------------------|-------------------------------|
| spread_col_headers | <i>Spread rows to columns</i> |
|--------------------|-------------------------------|

---

**Description**

Spread rows to columns

**Usage**

```
spread_col_headers(data, ...)
```

**Arguments**

|      |                                       |
|------|---------------------------------------|
| data | a data frame                          |
| ...  | list of data to load into environment |

**Value**

a data frame

---

|            |  |
|------------|--|
| student_df | <i>A minimal data set of student subjects and grades</i> |
|------------|--|

---

**Description**

A dataset containing the prices and other attributes of almost 54,000 diamonds.

**Usage**

```
student_df
```

**Format**

A data frame with 53940 rows and 10 variables:

**subject**  
**class**  
**student**  
**value**  
**grade ...**

---

|              |  |
|--------------|--|
| student_df_1 | <i>A minimal data set of student subjects and grades</i> |
|--------------|--|

---

**Description**

A dataset containing the prices and other attributes of almost 54,000 diamonds.

**Usage**

student\_df\_1

**Format**

A data frame with 53940 rows and 10 variables:

**subject**  
**class**  
**student**  
**value**  
**grade ...**

---

|              |  |
|--------------|--|
| student_df_2 | <i>A minimal data set of student subjects and grades</i> |
|--------------|--|

---

**Description**

A dataset containing the prices and other attributes of almost 54,000 diamonds.

**Usage**

student\_df\_2

**Format**

A data frame with 53940 rows and 10 variables:

**subject**  
**class**  
**student**  
**value**  
**grade ...**

---

style\_first\_col\_header  
*Style the first col header*

---

**Description**

Style the first col header

**Usage**

```
style_first_col_header(table, format)
```

**Arguments**

|        |              |
|--------|--------------|
| table  | a data frame |
| format | a data frame |

**Value**

a data frame

---

style\_spanner      *Style a spanner*

---

**Description**

Style a spanner

**Usage**

```
style_spanner(table, format)
```

**Arguments**

|        |            |
|--------|------------|
| table  | an gt_tbl  |
| format | formatting |

**Value**

a data frame

---

|        |  |
|--------|--|
| table1 | <i>A minimal data set of student subjects and grades</i> |
|--------|--|

---

**Description**

A dataset containing the prices and other attributes of almost 54,000 diamonds.

**Usage**

```
table1
```

**Format**

A data frame with 53940 rows and 10 variables:

```
subject
class
student
value
grade ...
```

---

|                   |   |
|-------------------|---|
| table_constructor | <i>Use attributes to construct a table with column and row headers.</i> |
|-------------------|---|

---

**Description**

Use attributes to construct a table with column and row headers.

**Usage**

```
table_constructor(
  df,
  col_header_df,
  row_header_df,
  data_vars,
  table_name = NULL,
  table_format
)
```

**Arguments**

|               |                        |
|---------------|------------------------|
| df            | a data frame           |
| col_header_df | col header information |
| row_header_df | row header information |
| data_vars     | cell variable          |
| table_name    | name of the table      |
| table_format  | formatting o the table |

**Value**

mmtable

---

|              |                                 |
|--------------|---------------------------------|
| table_format | <i>Format a mmtable object.</i> |
|--------------|---------------------------------|

---

**Description**

Format a mmtable object.

**Usage**

table\_format(locations, ...)

**Arguments**

|           |   |
|-----------|---|
| locations | indicating which cells should be formatted. |
| ...       | formatting instructions                     |

**Value**

format list

**Examples**

```
## Not run:

library(tidyverse)

row_list <- cells_body(rows = c(1,3,5,7,9,11))
style_list <- list(cell_borders(sides = "top",color = "grey"))
gm_df <- gapminder_mm %>% dplyr::filter(var != "Life expectancy")

gm_table <-
  gm_df %>%
  mmtable(cells = value) +
  header_top(year) +
  header_left(country) +
  header_top_left(var) +
  header_left_top(continent) +
  table_format(
    locations = row_list,
    style = style_list)

## End(Not run)
```



---

|                   |                          |
|-------------------|--------------------------|
| table_source_note | <i>Table source note</i> |
|-------------------|--------------------------|

---

**Description**

Table source note

**Usage**

```
table_source_note(...)
```

**Arguments**

... arguments of `gt::tab_source_note`

**Value**

format list

---

|             |                          |
|-------------|--------------------------|
| table_title | <i>Add a table title</i> |
|-------------|--------------------------|

---

**Description**

Add a table title

**Usage**

```
table_title(...)
```

**Arguments**

... arguments of `gt::tab_header`

**Value**

format list

---

|                  |   |
|------------------|---|
| transfer_spanner | <i>Transfer spanner from one table to another</i> |
|------------------|---|

---

**Description**

Transfer spanner from one table to another

**Usage**

```
transfer_spanner(gt_01, gt_02)
```

**Arguments**

|       |                                  |
|-------|----------------------------------|
| gt_01 | arguments of gt::tab_source_note |
| gt_02 | arguments of gt::tab_source_note |

**Value**

an mmtable

# Index

## \* datasets

- gapminder\_mm, 16
- student\_df, 28
- student\_df\_1, 29
- student\_df\_2, 29
- table1, 31

\*.mmtable, 3

+.mmtable, 3

/.mmtable, 4

add\_cells\_format, 4

add\_header\_format, 5

add\_header\_left, 6

add\_header\_left\_top, 7

add\_header\_top, 8

add\_header\_top\_left, 9

add\_spanner, 10

add\_table\_format, 11

add\_table\_source\_note, 12

apply\_format, 12

apply\_formats, 13

apply\_if, 13

cells\_format, 14

diagonalize, 15

extract\_max\_rects, 15

format\_a\_loc, 16

gapminder\_mm, 16

gen\_random\_ids, 17

get\_locations, 17

get\_max\_area\_bf, 18

get\_row\_header\_names, 18

get\_spanner\_html\_text, 19

header\_format, 19

header\_left, 20

header\_left\_top, 21

header\_merged\_cols, 22

header\_top, 22

header\_top\_left, 23

knit\_print.mmtable, 24

mmtable, 24

print.mmtable, 25

set\_class, 26

single\_cells\_to\_vector, 26

sort\_and\_select\_columns, 27

spannerize, 27

spread\_col\_headers, 28

student\_df, 28

student\_df\_1, 29

student\_df\_2, 29

style\_first\_col\_header, 30

style\_spanner, 30

table1, 31

table\_constructor, 31

table\_format, 32

table\_source\_note, 33

table\_title, 33

transfer\_spanner, 34