Package 'ivo.table'

November 25, 2025

Title Nicely Formatted Contingency Tables and Frequency Tables

Version 0.7.1

Description
Nicely formatted frequency tables and contingency tables (1-way, 2-way, 3-way and 4-way tables), that can easily be exported to HTML or 'Office' documents. Designed to work with pipes.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/mthulin/ivo.table,
https://mthulin.github.io/ivo.table/

BugReports https://github.com/mthulin/ivo.table/issues

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports dplyr, flextable, checkmate, gt, officer, purrr, tidyr

Suggests knitr, rmarkdown, palmerpenguins

VignetteBuilder knitr

NeedsCompilation no

Author Måns Thulin [aut, cre],
Kajsa Grind [aut],
Stefan Furne [aut]

Maintainer Måns Thulin <mans@statistikkonsult.com>

Repository CRAN

Date/Publication 2025-11-25 13:42:05 UTC

Contents

ivo_flextable_theme 2
ivo_gt_theme 3
ivo_table 4
ivo_table_add_mask 7
ivo_table_gt 8
ivo_table_masked 11

Index **14**

`ivo_flextable_theme` *Use nice fonts and colors for tables*

Description

A flextable theme for ivo_table objects.

Usage

```
ivo_flextable_theme(
  x,
  kway = 2,
  rowsums = FALSE,
  caption = NA,
  highlight_cols = NULL,
  highlight_rows = NULL,
  color = "darkgreen",
  font_name = "Arial",
  bold_cols = NULL
)
```

Arguments

`x`	A flextable.
`kway`	The number of "horizontal" variables in the table.
`rowsums`	A logical, saying whether the rightmost column in the table contains the sum of each row. Defaults to FALSE.
`caption`	An optional string containing a table caption.
`highlight_cols`	A numeric vector containing the indices of the columns that should be highlighted.
`highlight_rows`	A numeric vector containing the indices of the rows that should be highlighted.
`color`	A named color or a color HEX code, used for the lines in the table. Defaults to "darkgreen".
`font_name`	The name of the font to be used in the table. Defaults to "Arial".
`bold_cols`	A numeric vector containing the indices of the columns that should use a bold font.

Details

The default settings use a dark green color and a sans serif font.

Value

A styled flextable.

Author(s)

Måns Thulin

Examples

```
library(tidyr)
library(dplyr)
library(flextable)
example_data <- data.frame(Year = sample(2020:2023, 50, replace = TRUE),
A = sample(c("Type 1", "Type 2"), 50, replace = TRUE),
B = sample(c("Apples", "Oranges", "Bananas"), 50, replace = TRUE),
C = sample(c("Swedish", "Norwegian", "Chilean"), 50, replace = TRUE))
example_data |> select(B, A) |>
  ftable(exclude=NULL) |>
  data.frame() |>
  spread(A, Freq) |>
  regulartable() |>
  ivo_flextable_theme()
```

---

ivo_gt_theme *A nice GT theme*

Description

Applies a predefined theme to a GT table, including font, colors, and alignment.

Usage

```
ivo_gt_theme(table, color = "darkgreen", font_name = "Arial")
```

Arguments

table	A gt object.
color	A named color or a color HEX code, used for the lines in the table. Defaults to "darkgreen".
font_name	The name of the font to be used in the table. Defaults to "Arial".

Value

A styled gt table.

Author(s)

Stefan Furne

`ivo_table` *Create pretty frequency/contingency tables*

Description

`ivo_table()` lets you easily create a table using pretty fonts and colors. If you want the table with masked values use `ivo_table_masked()`.

Usage

```
ivo_table(
  df,
  extra_header = TRUE,
  exclude_missing = FALSE,
  missing_string = "(Missing)",
  colsums = FALSE,
  rowsums = FALSE,
  sums_string = "Total",
  caption = NA,
  highlight_cols = NULL,
  highlight_rows = NULL,
  percent_by = NA,
  color = "darkgreen",
  font_name = "Arial",
  bold_cols = NULL,
  long_table = FALSE,
  remove_zero_rows = FALSE
)
```

Arguments

`df`	A data frame with 1-4 columns
`extra_header`	Should the variable name be displayed? Defaults to TRUE.
`exclude_missing`	Whether to exclude missing values from the table. Defaults to FALSE.
`missing_string`	A string used to indicate missing values. Defaults to "(Missing)".
`colsums`	A logical indicating whether the sum of each column should be computed. Defaults to FALSE.
`rowsums`	A logical indicating whether the sum of each row should be computed. Defaults to FALSE.
`sums_string`	A string that is printed in the column/row where row/column sums are shown. Defaults to "Total".
`caption`	An optional string containing a table caption.
`highlight_cols`	A numeric vector containing the indices of the columns that should be highlighted.

highlight_rows	A numeric vector containing the indices of the rows that should be highlighted.
percent_by	Used to get percentages instead of frequencies. There are three options: "row" to get percentages by row (each row sum is 100 percent), "col" to get percentages by column (each the sum of each row to 100 percent) and "tot" to get percentages out of the total (the sum of all cells is 100 percent). The default, NA, means that frequencies are displayed instead.
color	A named color or a color HEX code, used for the lines in the table. Defaults to "darkgreen".
font_name	The name of the font to be used in the table. Defaults to "Arial".
bold_cols	A numeric vector containing the indices of the columns that should use a bold font.
long_table	For one-way tables: FALSE (the default) means that the table will be wide and consist of a single row, TRUE means that the table will be long and consist of a single column.
remove_zero_rows	If set to TRUE, removes all rows that contain nothing but zeros. The default is FALSE.

Details

The functions `ivo_table()` and `ivo_table_masked()` takes a `data.frame` with 1-4 columns. The order of the columns in the `data.frame` will determine where they will be displayed in the table. The first column will always be displayed at the top of the table. If there are more than one column the following 2-4 columns will be displayed to the left in the order 2, 3, 4. To change how the columns are displayed in the table; change the place of the columns in the `data.frame` using `dplyr::select()`.

Value

A stylized `flextable`.

Author(s)

Måns Thulin and Kajsa Grind

See Also

ivo_table_add_mask

Examples

```
# Generate example data
example_data <- data.frame(Year = sample(2020:2023, 50, replace = TRUE),
A = sample(c("Type 1", "Type 2"), 50, replace = TRUE),
B = sample(c("Apples", "Oranges", "Bananas"), 50, replace = TRUE),
C = sample(c("Swedish", "Norwegian", "Chilean"), 50, replace = TRUE))

### 1 way tables ###
data1 <- example_data |> dplyr::select(Year)
```

```
ivo_table(data1)
ivo_table(data1, extra_header = FALSE) # Remove the header
ivo_table(data1, color = "orange") # Change color on table lines
ivo_table(data1, long_table = TRUE) # Draw the table in a long format
ivo_table(data1, font_name = "Garamond") # Use a different font

ivo_table_masked(data1) # No masking because all counts are >=5
ivo_table_masked(data1, cell = 15) # Counts below <=15 are masked

# With pipes
example_data |> dplyr::select(Year) |> ivo_table()

### 2-way tables ###
data2 <- example_data |> dplyr::select(A, B)
data2_swap <- example_data |> dplyr::select(B, A)

# Basic tables:
ivo_table(data2)
ivo_table(data2_swap) # Swap order of the columns
ivo_table(data2, colsums = TRUE) # Add the sum of each column
ivo_table(data2, rowsums = TRUE) # Add the sum of each row
ivo_table(data2, caption = "Awesome table") # Add a caption
ivo_table(data2, highlight_cols = 3) # Highlight column 3
ivo_table(data2, highlight_rows = 2, highlight_cols = 3) # Highlight cell at row 2 column 3
ivo_table(data2, bold_cols = 1) # Make the names in the first column bold

# Tables with percentages:
ivo_table(data2, percent_by = "row") # By row
ivo_table(data2, percent_by = "col") # By column
ivo_table(data2, percent_by = "tot") # By total

# Masked tables:
ivo_table_masked(data2)
ivo_table_masked(data2, cell = 7) # Counts <= 7 are masked
# Row and column sums are also masked:
ivo_table_masked(
data2,
cell = 3,
colsums = TRUE,
rowsums = TRUE)

# Add a note at the end of the table:
# (colwidths must be set to the number of columns in the table)
ivo_table(data2) |>
  flextable::add_footer_row(values = "This is a footnote.",
                            colwidths = 3)

# Add footnotes to cells in the table:
ivo_table(data2) |>
flextable::footnote(i = c(1, 3), j = c(1, 2),
                    value = flextable::as_paragraph(c(
                      "Some remark.",
```

```
                      "Some comment.")),
                    ref_symbols = c("a", "b"))

# Add footnotes to cells in the table header:
ivo_table(data2) |>
flextable::footnote(i = 2, j = c(1, 3),
                    value = flextable::as_paragraph(c(
                      "Some remark.",
                      "Some comment.")),
                    ref_symbols = c("a", "b"),
                    part = "header")

### 3-way tables ###
data3 <- example_data |> dplyr::select(C, B, Year)

ivo_table(data3)
ivo_table(data3, colsums = TRUE, rowsums = TRUE) # Add the sum of each column and each row

ivo_table_masked(
data3,
cell = 3,
caption = "Values between 1 and 3 are masked."
)

### 4-way tables ###
data4 <- example_data |> dplyr::select(Year, B, C, A)

ivo_table(data4)
ivo_table(data4, remove_zero_rows = TRUE) # Remove the row with zeros

# Add the sum of each column and each row and highlight column 6:
ivo_table(
data4,
colsums = TRUE,
rowsums = TRUE,
highlight_cols = 6)

ivo_table_masked(data4, colsums = TRUE, rowsums = TRUE)
```

`ivo_table_add_mask` *Add masking (censoring) to a table*

Description

Table masking using cell counts.

Usage

```
ivo_table_add_mask(df, cell = 5)
```

Arguments

df	A data frame containing a column called "Freq", e.g. a frequency table created using `ftable(exclude=NULL) |> data.frame()`.
cell	The cell count at which masking should be used. Cell counts between 1 and this number will be masked. The default is 5.

Details

Masking is used to prevent the distribution of tables where individuals could be identified.

Value

A data frame with masked cell counts.

Author(s)

Måns Thulin

See Also

ivo_table_masked for masked tables.

Examples

```
library(dplyr)
example_data <- data.frame(Year = sample(2020:2023, 50, replace = TRUE),
A = sample(c("Type 1", "Type 2"), 50, replace = TRUE),
B = sample(c("Apples", "Oranges", "Bananas"), 50, replace = TRUE),
C = sample(c("Swedish", "Norwegian", "Chilean"), 50, replace = TRUE))
# With masking limit set at 7:
example_data |> select(Year, A) |>
  ftable(exclude=NULL) |>
  data.frame() |>
  ivo_table_add_mask(cell = 7)
```

---

ivo_table_gt *Create pretty frequency/contingency GT tables*

Description

`ivo_table_gt()` lets you easily create a GT table using pretty fonts and colors.

Usage

```
ivo_table_gt(
  df,
  color = "darkgreen",
  font_name = "Arial",
  caption = NULL,
  subtitle = NULL,
  extra_header = TRUE,
  source_note = NULL,
  mask = NULL,
  missing_string = "(Missing)",
  sums = NULL
)
```

Arguments

df	A data frame with 1-3 columns
color	A named color or a color HEX code, used for the lines in the table. Defaults to "darkgreen".
font_name	The name of the font to be used in the table. Defaults to "Arial".
caption	An optional string containing a table title.
subtitle	An optional string containing a table subtitle. Only usable together with title.
extra_header	Should the variable name be displayed? Defaults to TRUE.
source_note	An optional string for a table source note.
mask	An optional integer to mask counts below given value.
missing_string	A string used to indicate missing values. Defaults to "(Missing)".
sums	An optional vector to add sums to "rows" and "cols".

Details

The functions `ivo_table_gt()` takes a `data.frame` with 1-3 columns. The order of the columns in the `data.frame` will determine where they will be displayed in the table. The first column will always be displayed at the top of the table. If there are more than one column the following 2-3 columns will be displayed to the left in order. To change how the columns are displayed in the table; change the place of the columns in the `data.frame` using `dplyr::select()`.

Value

A stylized `GT` table.

Author(s)

Stefan Furne

See Also

ivo_gt_theme

Examples

```
# Generate example data
example_data <- data.frame(
    Year = sample(2020:2023, 50, replace = TRUE),
    A = sample(c("Type 1", "Type 2"), 50, replace = TRUE),
    B = sample(c("Apples", "Oranges", "Bananas"), 50, replace = TRUE),
    C = sample(c("Swedish", "Norwegian", "Chilean"), 50, replace = TRUE)
)

### 1 way tables ###
data1 <- example_data |> dplyr::select(Year)

ivo_table_gt(data1)
ivo_table_gt(data1, extra_header = FALSE) # Remove the header
ivo_table_gt(data1, color = "orange") # Change color on table lines

ivo_table_gt(data1, mask = 15) # Counts below <=15 are masked

# With pipes
example_data |>
    dplyr::select(Year) |>
    ivo_table_gt()

### 2-way tables ###
data2 <- example_data |> dplyr::select(A, B)
data2_swap <- example_data |> dplyr::select(B, A)

# Basic tables:
ivo_table_gt(data2)
ivo_table_gt(data2_swap) # Swap order of the columns
ivo_table_gt(data2, sums = "cols") # Add the sum of each column
ivo_table_gt(data2, sums = "rows") # Add the sum of each row
ivo_table_gt(data2, sums = c("cols", "rows")) # Add the sum of each row and column
ivo_table_gt(data2, caption = "Awesome table") # Add a caption
ivo_table_gt(data2,
    caption = "Awesome table",
    subtitle = "It's really awesome"
) # Add a subtitle for the title

# Masked tables:
ivo_table_gt(data2, mask = 7) # Counts <= 7 are masked
# Row and column sums are also masked:
ivo_table_gt(
    data2,
    mask = 3,
    sums = c("cols", "rows"),
)

# Add a note at the end of the table:
# (colwidths must be set to the number of columns in the table)
ivo_table_gt(data2, source_note = "This is a footnote.")
```

```
### 3-way tables ###
data3 <- example_data |> dplyr::select(C, B, Year)

ivo_table_gt(data3)
ivo_table_gt(data3, sums = c("cols", "rows")) # Add the sum of each column and each row

ivo_table_gt(
    data3,
    mask = 3,
    caption = "Values between 1 and 3 are masked."
)
```

`ivo_table_masked` *Create a masked frequency/contingency table*

Description

`ivo_table_masked()` lets you easily create pretty masked tables. If you want the table without masked values use `ivo_table()` instead.

Usage

```
ivo_table_masked(
  df,
  cell = 5,
  extra_header = TRUE,
  exclude_missing = FALSE,
  missing_string = "(Missing)",
  colsums = FALSE,
  rowsums = FALSE,
  sums_string = "Total",
  caption = NA,
  highlight_cols = NULL,
  highlight_rows = NULL,
  color = "darkgreen",
  font_name = "Arial",
  bold_cols = NULL,
  long_table = FALSE,
  remove_zero_rows = FALSE
)
```

Arguments

`df`	A data frame with 1-4 columns
`cell`	The largest value that will be masked. Defaults to 5, meaning that values between 1 and 5 are masked.
`extra_header`	Should the variable name be displayed? Defaults to TRUE.

`exclude_missing`
: Whether to exclude missing values from the table. Defaults to FALSE.

`missing_string`
: A string used to indicate missing values. Defaults to "(Missing)".

`colsums`
: A logical indicating whether the sum of each column should be computed. Defaults to FALSE.

`rowsums`
: A logical indicating whether the sum of each row should be computed. Defaults to FALSE.

`sums_string`
: A string that is printed in the column/row where row/column sums are shown. Defaults to "Total".

`caption`
: An optional string containing a table caption.

`highlight_cols`
: A numeric vector containing the indices of the columns that should be highlighted.

`highlight_rows`
: A numeric vector containing the indices of the rows that should be highlighted.

`color`
: A named color or a color HEX code, used for the lines in the table. Defaults to "darkgreen".

`font_name`
: The name of the font to be used in the table. Defaults to "Arial".

`bold_cols`
: A numeric vector containing the indices of the columns that should use a bold font.

`long_table`
: For one-way tables: FALSE (the default) means that the table will be wide and consist of a single row, TRUE means that the table will be long and consist of a single column.

`remove_zero_rows`
: If set to TRUE, removes all rows that contain nothing but zeros. The default is FALSE.

Details

The functions `ivo_table()` and `ivo_table_masked()` takes a `data.frame` with 1-4 columns. The order of the columns in the `data.frame` will determine where they will be displayed in the table. The first column will always be displayed at the top of the table. If there are more than one column the following 2-4 columns will be displayed to the left in the order 2, 3, 4. To change how the columns are displayed in the table; change the place of the columns in the `data.frame` using `dplyr::select()`.

Value

A stylized `flextable`.

Author(s)

Måns Thulin and Kajsa Grind

See Also

ivo_table_add_mask

Examples

```
# Generate example data
example_data <- data.frame(Year = sample(2020:2023, 50, replace = TRUE),
A = sample(c("Type 1", "Type 2"), 50, replace = TRUE),
B = sample(c("Apples", "Oranges", "Bananas"), 50, replace = TRUE),
C = sample(c("Swedish", "Norwegian", "Chilean"), 50, replace = TRUE))

### 1-way tables ###
data1 <- example_data |> dplyr::select(Year)
ivo_table_masked(data1) # No masking because all counts are >=5
ivo_table_masked(data1, cell = 15) # Counts below <=15 are masked

# With pipes
example_data |> dplyr::select(Year) |> ivo_table()

### 2-way tables ###
data2 <- example_data |> dplyr::select(A, B)
ivo_table_masked(data2)
ivo_table_masked(data2, cell = 7) # Counts <= 7 are masked
# Row and column sums are also masked:
ivo_table_masked(
data2,
cell = 3,
colsums = TRUE,
rowsums = TRUE)

### 3-way tables ###
data3 <- example_data |> dplyr::select(C, B, Year)
ivo_table_masked(
data3,
cell = 3,
caption = "Values between 1 and 3 are masked."
)

### 4-way tables ###
data4 <- example_data |> dplyr::select(Year, B, C, A)
ivo_table_masked(data4, colsums = TRUE, rowsums = TRUE)
```

Index

ivo_flextable_theme, 2
ivo_gt_theme, 3
ivo_table, 4
ivo_table_add_mask, 7
ivo_table_gt, 8
ivo_table_masked, 11