Package: HeatmapR (via r-universe)

May 28, 2026

Type Package

Title Create Heatmaps Using Base Graphics

Version 1.1.0

Date 2026-03-19

Description Provides a lightweight framework for creating high quality, complex heatmaps using base graphics. Supports hierarchical clustering with dendrograms, column and row scaling, cluster sub-divisions, customizable cell colours, shapes and sizes, legends, and flexible layouts for arranging multiple heatmaps. Designed to return plot objects that can be easily arranged with other plots without sacrificing resolution. Methods for hierarchical clustering and distance computations are described in Murtagh and Contreras (2012) <doi:10.1002/wics.53>. Dendrogram visualisation methods are based on the 'ggdendro' package by de Vries and Ripley (2020) <https://CRAN.R-project.org/package=ggdendro>.

URL https://dillonhammill.github.io/HeatmapR/

BugReports https://github.com/DillonHammill/HeatmapR/issues

Depends R (>= 3.5)

Imports methods, stats, graphics, grDevices, tools

License GPL-2

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Language en-GB

Suggests covr, testthat, knitr, rmarkdown

VignetteBuilder knitr

Repository https://dillonhammill.r-universe.dev

Date/Publication 2026-03-29 05:08:21 UTC

RemoteUrl https://github.com/dillonhammill/heatmapr

RemoteRef HEAD

RemoteSha c09079d06322a8742ca370c15a34d1c3a3a9a898

Contents

`heat_map` *Create a complex heatmap using base graphics*

Description

Create a complex heatmap using base graphics

Usage

```
heat_map(
  x,
  scale = FALSE,
  scale_method = "range",
  dist_method = "euclidean",
  clust_method = "complete",
  round = 2,
  tree = FALSE,
  tree_x = NULL,
  tree_size_x = 1,
  tree_scale_x = FALSE,
  tree_cut_x = NULL,
  tree_split_x = 1,
  tree_label_x = FALSE,
  tree_label_size_x = 0.1,
  tree_label_col_x = "grey40",
  tree_label_col_alpha_x = 1,
  tree_label_text_x = NA,
  tree_label_text_font_x = 1,
  tree_label_text_size_x = 1,
```

```
  tree_label_text_col_x = "black",
  tree_label_text_col_alpha_x = 1,
  tree_y = NULL,
  tree_size_y = 1,
  tree_scale_y = FALSE,
  tree_cut_y = NULL,
  tree_split_y = 1,
  tree_label_y = FALSE,
  tree_label_size_y = 0.1,
  tree_label_col_y = "grey40",
  tree_label_col_alpha_y = 1,
  tree_label_text_y = NA,
  tree_label_text_font_y = 1,
  tree_label_text_size_y = 1,
  tree_label_text_col_y = "black",
  tree_label_text_col_alpha_y = 1,
  cell_shape = "rect",
  cell_size = FALSE,
  cell_col_palette = c("red", "blue", "green", "orange", "magenta", "purple"),
  cell_col_scale,
  cell_col_scale_limits = NULL,
  cell_size_scale_limits = NULL,
  cell_col_alpha = 1,
  cell_col_empty = "white",
  cell_border_line_type = 1,
  cell_border_line_width = 1,
  cell_border_line_col = "black",
  cell_border_line_col_alpha = 1,
  cell_border_mask = NULL,
  cell_text = FALSE,
  cell_text_font = 1,
  cell_text_size = 1,
  cell_text_col = "white",
  cell_text_col_alpha = 1,
  bar_size_x = 1,
  bar_values_x = NULL,
  bar_axis_label_x = NULL,
  bar_axis_label_adj_x = 0,
  bar_axis_label_font_x = 1,
  bar_axis_label_size_x = 1,
  bar_axis_label_col_x = "black",
  bar_axis_label_col_alpha_x = 1,
  bar_fill_x = "grey40",
  bar_fill_alpha_x = 1,
  bar_line_type_x = 1,
  bar_line_width_x = 1,
  bar_line_col_x = "black",
  bar_line_col_alpha_x = 1,
```

```
bar_size_y = 1,
bar_values_y = NULL,
bar_axis_label_y = NULL,
bar_axis_label_adj_y = 0,
bar_axis_label_font_y = 1,
bar_axis_label_size_y = 1,
bar_axis_label_col_y = "black",
bar_axis_label_col_alpha_y = 1,
bar_fill_y = "grey40",
bar_fill_alpha_y = 1,
bar_line_type_y = 1,
bar_line_width_y = 1,
bar_line_col_y = "black",
bar_line_col_alpha_y = 1,
axis_text_x = NULL,
axis_text_side_x = "bottom",
axis_text_font_x = 1,
axis_text_size_x = 1,
axis_text_col_x = "black",
axis_text_col_alpha_x = 1,
axis_text_angle_x = 3,
axis_text_adjust_x = 0.45,
axis_label_x = NULL,
axis_label_font_x = 2,
axis_label_size_x = 1.2,
axis_label_col_x = "black",
axis_label_col_alpha_x = 1,
axis_ticks_length_x = 1,
axis_text_y = NULL,
axis_text_side_y = "left",
axis_text_font_y = 1,
axis_text_size_y = 1,
axis_text_col_y = "black",
axis_text_col_alpha_y = 1,
axis_text_angle_y = 1,
axis_text_adjust_y = 0.45,
axis_label_y = NULL,
axis_label_font_y = 2,
axis_label_size_y = 1.2,
axis_label_col_y = "black",
axis_label_col_alpha_y = 1,
axis_ticks_length_y = 1,
margins = c(NA, NA, NA, NA),
title = NULL,
title_text_font = 2,
title_text_size = 1.5,
title_text_col = "black",
title_text_col_alpha = 1,
```

```
  legend = TRUE,
  legend_size = 1,
  legend_col_scale_size = 1,
  legend_title = NULL,
  legend_title_text_font = 2,
  legend_title_text_size = 1,
  legend_title_text_col = "black",
  legend_title_text_col_alpha = 1,
  legend_text_font = 1,
  legend_text_size = 1,
  legend_text_col = "black",
  legend_text_col_alpha = 1,
  popup = TRUE,
  popup_size = c(7, 7),
  ...
)
```

Arguments

x	matrix or data.frame containing the data to display in the heatmap.
scale	logical indicating whether the data should be scaled prior to constructing the heatmap. Addition options include "column" or "row" to indicate whether scaling should be performed across rows or columns.
scale_method	indicates the type of scaling to perform on rows or columns as indicated by scale, options include "range", "mean" or "zscore". Set to "range" by default.
dist_method	indicates the type of distance metric to use when constructing dendrograms, set to "euclidean" distance by default. See ?dist for alternatives. Also supports "cosine" for cosine distance (1 - cosine similarity).
clust_method	indicates the type of agglomeration method to use when constructing performing hierarchical clustering, set to "complete" by default. See ?hclust for alternatives.
round	indicates the number of decimal places to round values when cell_text = TRUE and values are displayed in the heatmap, set to 2 decimal places by default
tree	options include "row", "y", "column", "x" or "both" to indicate the axes for which dendrograms should be constructed, set to FALSE by default. This argument overrides the tree_x and tree_y arguments.
tree_x	logical indicating whether dendrograms should be constructed for the x axis (columns), set to NULL by default.
tree_size_x	numeric to control the height of the dendrogram for the x axis, set to 1 by default.
tree_scale_x	logical indicating whether the branch heights of the x axis dendrogram should be scaled for better visualisation, set to FALSE by default.
tree_cut_x	either a numeric ranging from 0 to 1 indicating the branch cut height for x axis dendrogram (proportional for non-cosine distances, or absolute cosine distance threshold when dist_method = "cosine") or an integer indicating the desired number of clusters to obtain by cutting the x axis dendrogram. Alternatively,

clusters can be manually defined by specifying the number of columns to include in each cluster (e.g., c(3,4,5)), or a vector of cluster indices can be supplied to assign each column to a specific cluster (e.g., c(1,1,1,2,2,2,2,3,3,3) for 10 columns in 3 clusters). When a vector of indices is supplied, its length must equal the number of columns in x.

`tree_split_x` a numeric to control the spacing between x axis tree splits, set to 1 by default. Setting the argument to 0 will remove axis tree splits.

`tree_label_x` logical indicating whether a label should be added for each cluster within the x axis tree when `tree_cut_x` is specified, set to FALSE by default.

`tree_label_size_x`
numeric to control the height of the tree cluster labels for the x axis tree, set to 0.1 by default.

`tree_label_col_x`
vector of colours to use for x axis tree cluster labels, set to "grey40" by default.

`tree_label_col_alpha_x`
numeric to control the transparency of the x axis tree cluster labels, set to 1 by default to use solid colours.

`tree_label_text_x`
a vector of text to include in the x axis tree cluster labels, set to NA by default.

`tree_label_text_font_x`
a vector of font types to use for text in x axis cluster labels, set to 1 by default for plain text. See `font` in `?par` for alternatives.

`tree_label_text_size_x`
a vector of numerics to control the size of the text in the x axis cluster labels, set to 1 by default.

`tree_label_text_col_x`
a vector of colours to control the colour of text in x axis cluster labels, set to `"black"` by default.

`tree_label_text_col_alpha_x`
a vector of numerics to control the transparency of text in x axis cluster labels, set to 1 by default to use solid colours.

`tree_y` logical indicating whether dendrograms should be constructed for the y axis (rows), set to NULL by default.

`tree_size_y` numeric to control the width of the dendrogram for the y axis, set to 1 by default.

`tree_scale_y` logical indicating whether the branch heights of the y axis dendrogram should be scaled for better visualisation, set to FALSE by default.

`tree_cut_y` either a numeric ranging from 0 to 1 indicating the branch cut height for y axis dendrogram (proportional for non-cosine distances, or absolute cosine distance threshold when `dist_method = "cosine"`) or an integer indicating the desired number of clusters to obtain by cutting the y axis dendrogram. Alternatively, clusters can be manually defined by specifying the number of rows to include in each cluster (e.g., c(3,4,5)), or a vector of cluster indices can be supplied to assign each row to a specific cluster (e.g., c(1,1,1,2,2,2) for 6 rows in 2 clusters). When a vector of indices is supplied, its length must equal the number of rows in x.

`tree_split_y` a numeric to control the spacing between y axis tree splits, set to 1 by default. Setting the argument to 0 will remove axis tree splits.

`tree_label_y` logical indicating whether a label should be added for each cluster within the y axis tree when `tree_cut_y` is specified, set to FALSE by default.

`tree_label_size_y`
numeric to control the width of the tree cluster labels for the y axis tree, set to 0.1 by default.

`tree_label_col_y`
vector of colours to use for y axis tree cluster labels, set to "grey40" by default.

`tree_label_col_alpha_y`
numeric to control the transparency of the y axis tree cluster labels, set to 1 by default to use solid colours.

`tree_label_text_y`
a vector of text to include in the y axis tree cluster labels, set to NA by default.

`tree_label_text_font_y`
a vector of font types to use for text in y axis cluster labels, set to 1 by default for plain text. See `font` in `?par` for alternatives.

`tree_label_text_size_y`
a vector of numerics to control the size of the text in the y axis cluster labels, set to 1 by default.

`tree_label_text_col_y`
a vector of colours to control the colour of text in y axis cluster labels, set to `"black"` by default.

`tree_label_text_col_alpha_y`
a vector of numerics to control the transparency of text in y axis cluster labels, set to 1 by default to use solid colours.

`cell_shape` indicates the shape to use for the cells in the heatmap, options include `"rect"`, `"circle"` or `"diamond"`.

`cell_size` logical indicating whether each cell in the heatmap should be scaled by the value in `x`, set to FALSE by default. Alternatively, a matrix of the same dimensions as `x` containing the values by which the size of each cell should be scaled.

`cell_col_palette`
a vector of colours from which colours are selected for columns containing non-numeric data.

`cell_col_scale` a vector of colours to use for the colour scale of numeric values, set to a hybrid colour-blind friendly viridis colour palette by default.

`cell_col_scale_limits`
a numeric vector of length 2 specifying custom limits for the colour scale as `c(min, max)`. If NULL (default), limits are computed from the data before rounding. Custom limits allow you to set consistent colour scales across multiple heatmaps or override automatic scaling.

`cell_size_scale_limits`
a numeric vector of length 2 specifying custom limits for the size scale as `c(min, max)`. If NULL (default), limits are computed from the data before rounding. Only used when `cell_size` is not FALSE. Custom limits allow consistent size scales across multiple heatmaps.

`cell_col_alpha` a numeric to control the fill transparency of cells within the heatmap, set to 1 by default to use solid colours.

`cell_col_empty` a colour to use for missing values in `x`, set to `"white"` by default.

`cell_border_line_type`
: a integer to indicate the type of line to use for cell borders, set to 1 by default for solid lines. See `lty` in `?par` for alternatives.

`cell_border_line_width`
: a numeric to control the with of cell borders, set to 1 by default.

`cell_border_line_col`
: indicates the colour to use for cell borders, set to `"black"` by default.

`cell_border_line_col_alpha`
: numeric to control the transparency of cell borders, set to 1 by default to use solid colours.

`cell_border_mask`
: a matrix of the same dimensions as `x` containing TRUE/FALSE values indicating whether each cell should have a colored border (TRUE) or a transparent border (FALSE), set to NULL by default to apply border color to all cells.

`cell_text` logical indicating whether the values in `x` should be displayed in each cell of the heatmap, set to FALSE by default.

`cell_text_font` an integer to control the font face of cell text, set to 1 by default. See `font` in `?par` for alternatives.

`cell_text_size` numeric to control the size of cell text, set to 1 by default.

`cell_text_col` colour to use for cell text, set to `"white"` by default,

`cell_text_col_alpha`
: numeric ranging from 0 to 1 to control the transparency of cell text, set to 1 by default to use solid colours for cell text.

`bar_size_x` numeric to control the height of x axis bar plot, set to 1 by default.

`bar_values_x` a vector of values to display in x axis bar plot, supplied in the order matching the original columns of `x` or named with the column names of `x`. The values supplied to `bar_values_x` will be internally reordered to match the order of columns as determined by hierarchical clustering.

`bar_axis_label_x`
: axis label to use for the x axis bar plot.

`bar_axis_label_adj_x`
: scalar [-1, 1] to adjust the position of the x axis bar graph label relative to the x axis, set to 0 by default.

`bar_axis_label_font_x`
: font to use for the axis label of the x axis bar plot, set to 2 by default. See `font` in `?par` for alternatives.

`bar_axis_label_size_x`
: numeric to control the size of the axis text in x axis bar plots, set to 1 by default.

`bar_axis_label_col_x`
: colour to use for the axis text in x axis bar plot, set to `"black"` by default.

bar_axis_label_col_alpha_x
: numeric ranging from 0 to 1 to control the transparency of axis text in x axis bar plot, set to 1 by default to use solid colours.

bar_fill_x
: a vector of colours to use for the bars in the x axis bar plot, set to "grey40" by default.

bar_fill_alpha_x
: a numeric ranging from 0 to 1 to control the fill transparency of bars in x axis bar plot, set to 1 by default to use solid colours.

bar_line_type_x
: integer to control the line type of bar borders in the x axis bar plot, set to 1 by default to use solid lines. See lty in ?par for alternatives.

bar_line_width_x
: numeric to control the width bar borders in x axis bar plot, set to 1 by default.

bar_line_col_x
: colour to use for bar borders in x axis bar plot, set to "black" by default.

bar_line_col_alpha_x
: numeric ranging from 0 to 1 to control the transparency of bar borders in x axis bar plot, set to 1 by default.

bar_size_y
: numeric to control the width of y axis bar plot, set to 1 by default.

bar_values_y
: a vector of values to display in y axis bar plot, supplied in the order matching the original rownames of x or named with the row names of x. The values supplied to bar_values_y will be internally reordered to match the order of rows as determined by hierarchical clustering.

bar_axis_label_y
: axis label to use for the y axis bar plot.

bar_axis_label_adj_y
: scalar [-1, 1] to adjust the position of the y axis bar graph label relative to the y axis, set to 0 by default.

bar_axis_label_font_y
: font to use for the axis label of the y axis bar plot, set to 2 by default. See font in ?par for alternatives.

bar_axis_label_size_y
: numeric to control the size of the axis text in y axis bar plots, set to 1 by default.

bar_axis_label_col_y
: colour to use for the axis text in y axis bar plot, set to "black" by default.

bar_axis_label_col_alpha_y
: numeric ranging from 0 to 1 to control the transparency of axis text in y axis bar plot, set to 1 by default to use solid colours.

bar_fill_y
: a vector of colours to use for the bars in the y axis bar plot, set to "grey40" by default.

bar_fill_alpha_y
: a numeric ranging from 0 to 1 to control the fill transparency of bars in y axis bar plot, set to 1 by default to use solid colours.

bar_line_type_y
: integer to control the line type of bar borders in the y axis bar plot, set to 1 by default to use solid lines. See lty in ?par for alternatives.

bar_line_width_y
: numeric to control the width bar borders in y axis bar plot, set to 1 by default.

bar_line_col_y
: colour to use for bar borders in y axis bar plot, set to "black" by default.

bar_line_col_alpha_y
: numeric ranging from 0 to 1 to control the transparency of bar borders in y axis bar plot, set to 1 by default.

axis_text_x
: vector of text to use for x axis labels supplied in the order matching the constructed heatmap, set to NULL by default to use the column names of x.

axis_text_side_x
: indicates whether the x axis text should be on the 1 - "bottom" or 3 - "top" of the heatmap, set to "bottom" by default. All other heatmap components, including the tree, tree labels and bar plot will be positioned on the opposite side to the axis text.

axis_text_font_x
: integer to control the font face of x axis labels, set to 1 by default. See font in ?par for alternatives.

axis_text_size_x
: numeric to control the size of x axis text, set to 1 by default.

axis_text_col_x
: colour to use for x axis text labels, set to "black" by default.

axis_text_col_alpha_x
: numeric ranging from 0 to 1 to control the transparency of x axis text labels, set to 1 by default to use solid text colours.

axis_text_angle_x
: integer to control the angle of x axis text labels relative to the x axis, set to 3 by default. See las in ?par for alternatives.

axis_text_adjust_x
: numeric to adjust position x axis text relative to x axis ticks, set to 0.45 by default.

axis_label_x
: label to use for the x axis.

axis_label_font_x
: integer to control the font face of the x axis label, set to 2 by default.

axis_label_size_x
: numeric to control the size of the x axis label, set to 1.2 by default.

axis_label_col_x
: colour to use for x axis label text, set to "black" by default.

axis_label_col_alpha_x
: numeric ranging from 0 to 1 to control the transparency of the x axis label text, set to 1 by default to use solid colours.

axis_ticks_length_x
: numeric to control the length of the x axis ticks, set to 1 by default.

axis_text_y
: vector of text to use for y axis labels supplied in the order matching the constructed heatmap, set to NULL by default to use the row names of x.

axis_text_side_y
: indicates whether the y axis text should be on the 2 - "left" or 4 - "right" of the heatmap, set to "left" by default. All other heatmap components, including the tree, tree labels and bar plot will be positioned on the opposite side to the axis text.

axis_text_font_y
: integer to control the font face of y axis labels, set to 1 by default. See font in ?par for alternatives.

axis_text_size_y
: numeric to control the size of y axis text, set to 1 by default.

axis_text_col_y
: colour to use for y axis text labels, set to "black" by default.

axis_text_col_alpha_y
: numeric ranging from 0 to 1 to control the transparency of y axis text labels, set to 1 by default to use solid text colours.

axis_text_angle_y
: integer to control the angle of y axis text labels relative to the y axis, set to 3 by default. See las in ?par for alternatives.

axis_text_adjust_y
: numeric to adjust position y axis text relative to y axis ticks, set to 0.45 by default.

axis_label_y
: label to use for the y axis.

axis_label_font_y
: integer to control the font face of the y axis label, set to 2 by default.

axis_label_size_y
: numeric to control the size of the y axis label, set to 1.2 by default.

axis_label_col_y
: colour to use for y axis label text, set to "black" by default.

axis_label_col_alpha_y
: numeric ranging from 0 to 1 to control the transparency of the y axis label text, set to 1 by default to use solid colours.

axis_ticks_length_y
: numeric to control the length of the y axis ticks, set to 1 by default.

margins
: vector of numerics to control the size of the margins around the bottom, left, top and right of the heatmap. Setting any of these values to NA will allow for internal computation of optimal heatmap margins.

title
: text to include in the title above the heatmap.

title_text_font
: integer to control the font face for the heatmap title, set to 2 by default. See font in ?par for alternatives.

title_text_size
: numeric to control the size of the text in the heatmap title, set to 1.2 by default.

title_text_col
: colour to use for text in heatmap title, set to "black" by default.

title_text_col_alpha
: numeric ranging from 0 to 1 to control the transparency of text in heatmap title, set to 1 by default for solid colours.

`legend` logical indicating whether to include a legend in the heatmap, set to TRUE by default. Alternatively, `"size"`, `"colour"` or `"both"` to indicate the type(s) of legends to include in the heatmap.

`legend_size` numeric to control the amount of space allocated to the legend, set to 1 by default.

`legend_col_scale_size` numeric to control the width of the legend colour scale relative to the allocated space for the legend, set to 1 by default. legend_title = NULL,

`legend_title` a vector of length two containing text to be displayed above the cell colour and size legends respectively.

`legend_title_text_font` integer to control the font face for the heatmap legends, set to 2 by default for bold font. See `font` in `?par` for alternatives.

`legend_title_text_size` numeric to control the size of the text in the heatmap legend titles, set to 1 by default.

`legend_title_text_col` colour to use for text in heatmap legend titles, set to `"black"` by default.

`legend_title_text_col_alpha` numeric ranging from 0 to 1 to control the transparency of text in heatmap legend titles, set to 1 by default for solid colours.

`legend_text_font` integer to control the font face for legend text, set to 1 by default. See `font` in `?par` for alternatives.

`legend_text_size` numeric to control the size of text in the legend, set to 1 by default.

`legend_text_col` colour to use for text in the legend, set to `"black"`.

`legend_text_col_alpha` numeric to control the transparency of text in the legend, set to 1 by default to use solid colours.

`popup` logical indicating whether the heatmap should be constructed in a popup window, set to TRUE by default.

`popup_size` vector to control the height and width of the popup window in inches, set to `c(7,7)`.

`...` not in use.

Value

a recorded heatmap.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
heat_map(
  mtcars,
  scale = "column",
  cell_shape = "circle",
  cell_size = TRUE,
  tree_x = TRUE,
  tree_cut_x = 3,
  tree_y = TRUE,
  tree_cut_y = 3,
  bar_values_x = 1:11,
  bar_fill_x = rainbow(11),
  bar_values_y = 1:32,
  bar_fill_y = rainbow(32)
 )
```

---

heat_map_clust *Perform hierarchical clustering for heatmap*

Description

Perform hierarchical clustering for heatmap

Usage

```
heat_map_clust(
  x,
  tree = "row",
  dist = "euclidean",
  method = "complete",
  scale = FALSE,
  cut = NULL,
  ...
)
```

Arguments

x	matrix-like object to cluster. The distance matrix will be computed using dist and passed to hclust for hierarchical clustering.
tree	indicates whether hierarchical clustering should be performed by "row" or "column". Optionally a custom object of class "dist" or "hclust" which will be updated with cluster labels as specified by cut.
dist	method passed to dist to compute distance matrix, set to "euclidean" by default. Also supports "cosine" for cosine distance (1 - cosine similarity).
method	agglomeration method passed to hclust to perform hierarchical clustering, set to "complete" by default.

scale	logical indicating whether branch heights should be scaled for better visualisation, set to FALSE by default.
cut	value less than 1 specifying the tree cutpoint as a proportion of the tree height (for non-cosine distances) or as an absolute cosine distance threshold (when dist = "cosine"), or a value greater than or equal to 1 indicating the number of desired clusters.
...	additional arguments passed to dist or hclust.

Value

object of class hclust which describes the tree produced by the clustering process.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Hierarchical clustering
heat_map_hclust <- heat_map_clust(mtcars, cut = 5)
```

---

heat_map_complete *Indicate when a heatmap is complete and ready for saving*

Description

Indicate when a heatmap is complete and ready for saving

Usage

```
heat_map_complete()
```

Value

No return value, called for side effects.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Save heatmap
heat_map_save(file.path(tempdir(), "Heatmap.png"),
height = 7,
width = 15)

# Custom layout
heat_map_layout(layout = c(1,2))

# Construct raw heatmap
heat_map(iris[1:10,],
scale = FALSE,
title = "Iris Raw Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")

# Construct scaled heatmap
heat_map(iris[1:10,],
scale = "range",
title = "Iris Scaled Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")

# Signal completion
heat_map_complete()
```

---

heat_map_custom *Create a custom heatmap plot layout*

Description

heat_map_custom() is similar to cheat_map_save() with the exception that it doesn't write the plot to a file. heat_map_custom() opens a new graphics device and sets the desired layout in preparation for the addition of heatmaps and other plot objects. Once the custom plot is full users MUST run cyto_plot_complete() to close the graphics device and reset any heat_map() related settings (see example).

Usage

```
heat_map_custom(popup = TRUE, popup_size = c(8, 8), layout = NULL, ...)
```

Arguments

popup	logical indicating whether a popup graphics device should be opened.
popup_size	indicates the size of the popup graphics device in inches, set to c(8,8) by default.

layout	either a vector of the form c(nrow, ncol) defining the dimensions of the plot or a matrix defining a more sophisticated layout (see layout). Vectors can optionally contain a third element to indicate whether plots should be placed in row (1) or column (2) order, set to row order by default.
...	additional arguments passed to heat_map_new().

Value

No return value, called for side effects.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

See Also

heat_map_save

heat_map_new

heat_map_complete

Examples

```
heat_map_custom(
  popup = FALSE,
  layout = c(1,2)
)
heat_map(
  mtcars
)
plot(
  mtcars[, 1:2],
  pch = 16
)
heat_map_complete()
```

---

heat_map_layout *Arrange multiple heatmaps*

Description

Arrange multiple heatmaps

Usage

```
heat_map_layout(layout = NULL)
```

Arguments

layout	either a vector of the form c(nrow, ncol) defining the dimensions of the plot or a matrix defining a more sophisticated layout (see layout). Vectors can optionally contain a third element to indicate whether plots should be placed in row (1) or column (2) order, set to row order by default.

Value

No return value, called for side effects.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Save heatmap
heat_map_save(file.path(tempdir(), "Heatmap.png"),
height = 7,
width = 15)

# Custom layout
heat_map_layout(layout = c(1,2))

# Construct raw heatmap
heat_map(iris[1:10,],
scale = FALSE,
title = "Iris Raw Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")

# Construct scaled heatmap
heat_map(iris[1:10,],
scale = "range",
title = "Iris Scaled Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")

# Signal completion
heat_map_complete()
```

---

heat_map_new *Open a pop-up graphics device for heatmaps*

Description

Open a pop-up graphics device for heatmaps

Usage

```
heat_map_new(popup = FALSE, popup_size = c(8, 8), ...)
```

Arguments

popup	logical indicating whether a popup graphics device should be opened.
popup_size	indicates the size of the popup graphics device in inches, set to `c(8,8)` by default.
...	additional arguments passed to `dev.new`.

Value

No return value, called for side effects.

Author(s)

Dillon Hammill, <dillon.hammill21@gmail.com>

Examples

```
# Open platform-specific graphics device
heat_map_new(popup = TRUE)
```

`heat_map_record` *Record a custom heatmap*

Description

Record custom heatmap layout on current graphics device and save to an R object for future use.

Usage

```
heat_map_record()
```

Value

an object of class `recordedplot`.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Heatmap layout
heat_map_layout(c(1,2))

# Construct raw heatmap
heat_map(iris[1:10,],
scale = FALSE,
title = "Iris Raw Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")

# Construct scaled heatmap
heat_map(iris[1:10,],
scale = "range",
title = "Iris Scaled Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")

# Record heatmap layout
heat_map_plot <- heat_map_record()
```

`heat_map_reset` *Reset all heatmap related settings*

Description

Reset all heatmap related settings

Usage

```
heat_map_reset()
```

Value

No return value, called for side effects.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Reset HeatmapR settings
heat_map_reset()
```

heat_map_save *Save high resolution images*

Description

Save high resolution images

Usage

```
heat_map_save(
  save_as,
  width = 7,
  height = 7,
  units = "in",
  res = 300,
  multiple = FALSE,
  layout = NULL,
  ...
)
```

Arguments

save_as	name of the file to which the plot should be saved (including the file extension). Supported file formats include png, tiff, jpeg, svg and pdf.
width	numeric indicating the width of exported plot in units, set to 7 by default for image with width of 7 inches.
height	numeric indicating the height of the exported plot in units, set to 7 by default for image with height of 7 inches.
units	units to be used to set plot size, can be either pixels (px), inches (inches), centimetres (cm) or millimetres (mm). Set to "in" by default. Units cannot be altered for svg and pdf graphics devices.
res	resolution in dpi, set to 300 by default.
multiple	logical indicating whether multiple pages should be saved to separate numbered files, set to TRUE by default.
layout	a vector or matrix defining the custom layout of the plot to be created using heat_map_layout, set to NULL by default.
...	additional arguments for the appropriate png(), tiff(), jpeg(), svg() or pdf graphics devices.

Value

No return value, called for side effects.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Save Heatmap
heat_map_save(file.path(tempdir(), "Heatmap.png"),
height = 7,
width = 5)

# Construct Heatmap
heat_map(iris[1:10,],
scale = "range",
title = "Iris Heatmap",
axis_label_x = "Plant Parameter",
axis_label_y = "Row ID")
```

`heat_map_scale` *Scale numeric data prior to constructing heat_map*

Description

Apply column-wise or row-wise scaling to numeric columns in a matrix or data.frame prior to constructing a `heat_map`.

Usage

```
heat_map_scale(x, scale = "column", method = "range")
```

Arguments

`x`	matrix-like object to be scaled.
`scale`	indicates whether the data should be scaled by `"row"` or `"column"`, set to `"column"` by default.
`method`	type of scaling to perform, can be either `'range'`, `'mean'` or `'zscore'`. Range scaling normalizes the data to have limits between 0 and 1. Mean scaling subtracts the mean (calculated excluding missing values) from each value. Z-score scaling subtracts the mean from each value and then divides the result by the standard deviation.

Value

a matrix or data.frame containing the scaled data.

Author(s)

Dillon Hammill (dillon.hammill21@gmail.com)

Examples

```
# Range scaling
mtcars_scale_range <- heat_map_scale(mtcars,
method = "range")

# Mean scaling
mtcars_scale_mean <- heat_map_scale(mtcars,
method = "mean")

# Z-score scaling
mtcars_scale_zscore <- heat_map_scale(mtcars,
method = "zscore")
```

Index