

Package ‘ggcorrplot’

July 24, 2026

Type Package

Title Visualization of a Correlation Matrix using 'ggplot2'

Version 0.3.0

Date 2026-07-24

Description The 'ggcorrplot' package can be used to visualize easily a correlation matrix using 'ggplot2'. It provides a solution for reordering the correlation matrix and displays the significance level on the plot. It also includes a function for computing a matrix of correlation p-values.

License GPL-2

URL <https://rpkgs.datanovia.com/ggcorrplot/>,
<https://github.com/kassambara/ggcorrplot>

BugReports <https://github.com/kassambara/ggcorrplot/issues>

Depends R (≥ 3.3), ggplot2 ($\geq 3.3.6$)

Imports grDevices, reshape2, stats

Suggests testthat ($\geq 3.0.0$), knitr, rmarkdown, spelling, vdiff ($\geq 1.0.0$)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Alboukadel Kassambara [aut, cre],
Indrajeet Patil [ctb] (ORCID: <https://orcid.org/0000-0003-1995-6531>),
Twitter: @patilindrajeets)

Maintainer Alboukadel Kassambara <alboukadel.kassambara@gmail.com>

Repository CRAN

Date/Publication 2026-07-24 20:50:02 UTC

Contents

ggcorrplot	2
Index	9

ggcorrplot	<i>Visualization of a correlation matrix using ggplot2</i>
------------	--

Description

- ggcorrplot(): A graphical display of a correlation matrix using ggplot2. The main plot options are walked through in [ggcorrplot: Correlation Matrix Heatmap in R with ggplot2](#).
- cor_pmat(): Compute a correlation matrix p-values, to mark the significant cells. A worked example is in [ggcorrplot: Correlation Matrix Heatmap in R with ggplot2](#).

Usage

```
ggcorrplot(
  corr,
  method = c("square", "circle"),
  type = c("full", "lower", "upper"),
  ggtheme = ggplot2::theme_minimal,
  title = "",
  show.legend = TRUE,
  legend.title = "Corr",
  show.diag = NULL,
  colors = c("blue", "white", "red"),
  outline.color = "gray",
  hc.order = FALSE,
  hc.method = "complete",
  lab = FALSE,
  lab_col = "black",
  lab_size = 4,
  lab_fontface = "plain",
  sig.stars = FALSE,
  p.mat = NULL,
  sig.level = 0.05,
  insig = c("pch", "blank", "stars"),
  pch = 4,
  pch.col = "black",
  pch.cex = 5,
  tl.cex = 12,
  tl.col = NULL,
  tl.srt = 45,
  tl.vjust = 1,
  tl.hjust = 1,
  digits = 2,
```

```

    as.is = FALSE,
    nsmall = 0L,
    leading.zero = TRUE,
    legend.limit = c(-1, 1),
    circle.scale = 1,
    coord.fixed = TRUE,
    lower.method = NULL,
    upper.method = NULL,
    hc.rect = NULL,
    palette = NULL,
    preset = NULL,
    hc.rect.col = "gray30",
    scale.square = FALSE,
    cell.grid = FALSE,
    cell.grid.col = "grey90"
  )

  cor_pmat(x, ..., use = c("pairwise.complete.obs", "everything"))

```

Arguments

<code>corr</code>	the correlation matrix to visualize
<code>method</code>	character, the visualization method of correlation matrix to be used. Allowed values are "square" (default), "circle".
<code>type</code>	character, "full" (default), "lower" or "upper" display. A mixed layout (see <code>lower.method/upper.method</code>) always uses the full matrix.
<code>ggtheme</code>	ggplot2 function or theme object. Default value is 'theme_minimal'. Allowed values are the official ggplot2 themes including <code>theme_gray</code> , <code>theme_bw</code> , <code>theme_minimal</code> , <code>theme_classic</code> , <code>theme_void</code> , Theme objects are also allowed (e.g., 'theme_classic()'). A mixed "number" region reads the background from this argument, resolved against the default theme in force when <code>ggcorrplot()</code> is called, to decide how dark to draw the coefficient text (see <code>lower.method/upper.method</code>). So a dark theme belongs here rather than added to the returned plot with <code>+</code> , and a theme_set issued after the call is not seen.
<code>title</code>	character, title of the graph.
<code>show.legend</code>	logical, if TRUE the legend is displayed.
<code>legend.title</code>	a character string for the legend title. lower triangular, upper triangular or full matrix.
<code>show.diag</code>	NULL or logical, whether display the correlation coefficients on the principal diagonal. If NULL, the default is to show diagonal correlation for <code>type = "full"</code> and to remove it when <code>type</code> is one of "upper" or "lower".
<code>colors</code>	a vector of colors for the fill gradient. The default is a length-3 vector for the low, mid and high correlation values (mapped with scale_fill_gradient2). A vector of any other length (≥ 2) is spread evenly across the scale with scale_fill_gradientn , so an n-color palette (e.g. <code>RColorBrewer::brewer.pal(11, "RdBu")</code>) can be passed directly.

<code>outline.color</code>	the outline color of square or circle. Default value is "gray".
<code>hc.order</code>	logical value. If TRUE, correlation matrix will be hc.ordered using <code>hclust</code> function.
<code>hc.method</code>	the agglomeration method to be used in <code>hclust</code> (see <code>?hclust</code>).
<code>lab</code>	logical value. If TRUE, add correlation coefficient on the plot.
<code>lab_col, lab_size</code>	size and color to be used for the correlation coefficient labels. used when <code>lab = TRUE</code> .
<code>lab_fontface</code>	the font face ("plain", "bold", "italic", "bold.italic") for the correlation coefficient labels. Default is "plain". Used when <code>lab = TRUE</code> .
<code>sig.stars</code>	logical value. If TRUE and a <code>p.mat</code> is supplied, significance stars are appended to the coefficient labels (*** for $p \leq 0.001$, ** for $p \leq 0.01$, * for $p \leq 0.05$), e.g. <code>"-0.85**"</code> . Only used when <code>lab = TRUE</code> . Default is FALSE. When TRUE, significance is shown by the stars and the <code>insig = "pch"</code> markers are not drawn. These three thresholds are fixed and are not affected by <code>sig.level</code> .
<code>p.mat</code>	matrix of p-value. If NULL, arguments <code>sig.level</code> , <code>insig</code> , <code>pch</code> , <code>pch.col</code> , <code>pch.cex</code> is invalid.
<code>sig.level</code>	significant level, if the p-value in <code>p.mat</code> is bigger than <code>sig.level</code> , then the corresponding correlation coefficient is regarded as insignificant. This governs which cells <code>insig = "pch"</code> marks and <code>insig = "blank"</code> wipes; the star thresholds used by <code>insig = "stars"</code> and <code>sig.stars</code> are fixed (see those arguments) and do not follow <code>sig.level</code> .
<code>insig</code>	character, how to convey significance from <code>p.mat</code> : "pch" (default), "blank" or "stars". "pch" adds a character (see <code>pch</code>) on the glyphs of the insignificant cells; "blank" wipes those glyphs away; "stars" instead marks the SIGNIFICANT cells with significance stars (***/**/* for $p \leq 0.001/0.01/0.05$ – fixed thresholds, not <code>sig.level</code>). With the default <code>lab = FALSE</code> the stars are drawn on their own (in <code>pch.col</code> , sized by <code>lab_size</code>) as a standalone significance map; with <code>lab = TRUE</code> they are appended to the coefficient labels (e.g. <code>"-0.85***"</code> , as with <code>sig.stars</code>) so the two do not overprint.
<code>pch</code>	add character on the glyphs of insignificant correlation coefficients (only valid when <code>insig</code> is "pch"). Default value is 4.
<code>pch.col, pch.cex</code>	the color and the cex (size) of <code>pch</code> (only valid when <code>insig</code> is "pch").
<code>tl.cex, tl.col, tl.srt</code>	the size, the color and the string rotation of text label (variable names). <code>tl.col</code> defaults to NULL, which inherits the color from the theme.
<code>tl.vjust, tl.hjust</code>	the vertical and horizontal justification of the x-axis text labels, passed to element_text . Both default to 1; adjust them to reposition the variable-name labels.
<code>digits</code>	Decides the number of decimal digits to be displayed (Default: '2').
<code>as.is</code>	retained for backward compatibility; no longer affects the plot. The axis is now always drawn in the matrix (row/column) order, so the variable-name handling this argument used to control is done internally.

<code>nsmall</code>	the minimum number of digits to the right of the decimal point in the coefficient labels, passed to <code>format</code> . Default is 0 (no minimum, current behavior). Set e.g. <code>nsmall = 2</code> to keep trailing zeros (such as 0.70). Only used when <code>lab = TRUE</code> .
<code>leading.zero</code>	logical. If TRUE (default), coefficient labels keep the leading zero (e.g. 0.23, -0.67). Set to FALSE to drop it (.23, -.67), which is common for correlation tables. Only used when <code>lab = TRUE</code> .
<code>legend.limit</code>	a length-2 numeric vector giving the limits of the fill color scale. Default <code>c(-1, 1)</code> (suitable for a correlation matrix); set to NULL to use the data range instead, e.g. for a covariance matrix.
<code>circle.scale</code>	a scaling factor for the circle sizes when <code>method = "circle"</code> . Default is 1; increase it (e.g. <code>circle.scale = 2</code>) for larger circles or decrease it for smaller ones, which is useful when the output device size makes the default circles too small or too large. Has no effect when <code>method = "square"</code> .
<code>coord.fixed</code>	logical value. If TRUE (default), the plot uses <code>coord_fixed</code> so the cells are square. Set to FALSE to let the cells fill the plotting area (a non 1:1 aspect ratio), which can look better with many long variable names.
<code>lower.method, upper.method</code>	character, an optional per-triangle glyph for a mixed layout: one of "square", "circle" or "number" (the coefficient drawn as text, colored by its value on the fill ramp). When either is set, the plot switches to a mixed layout where the lower and upper triangles are drawn separately and the variable names are drawn on the diagonal; a triangle left NULL uses <code>method</code>. Both default to NULL (single-method plot, unchanged). In a mixed layout the single-method significance and label overlays (<code>lab</code>, <code>sig.stars</code>, <code>p.mat</code>, <code>insig</code>, <code>pch*</code>) do not apply; show coefficients with a "number" triangle instead. Over a light background the "number" text is drawn on a darkened copy of the ramp – same hues, so warm still reads as positive and cool as negative, but dark enough that a coefficient near zero stays readable instead of washing out. Over a dark background the ramp is used as given, its pale middle being what reads there. The background is taken from <code>ggtheme</code>; a theme added to the returned plot with <code>+</code> arrives too late to be seen.
<code>hc.rect</code>	integer or NULL (default). If an integer <code>k</code> , draws <code>k</code> rectangles (no fill) around the clusters obtained by cutting the hierarchical tree, marking the cluster blocks on the diagonal. Requires <code>hc.order = TRUE</code> and <code>type = "full"</code> (the boxes span whole diagonal blocks). NULL (default) draws no rectangles. For a fully custom box style, add your own <code>annotate("rect", ...)</code> to the returned plot.
<code>palette</code>	optional name of a built-in colorblind-safe diverging palette for the fill gradient: "RdBu" or "PuOr". A convenience shortcut for colors: when set it supplies the gradient (an 11-stop ramp, white at zero, cool = negative / warm = positive) and takes precedence over colors. Defaults to NULL (use colors), so existing calls are unchanged.
<code>preset</code>	optional name of a bundle of publication-grade defaults. The only value, "publication", sets white cell outlines and the colorblind-safe "RdBu" palette in one token. It fills only the arguments you did not supply, so any argument you pass explicitly (e.g. <code>outline.color</code> , <code>colors</code> , <code>palette</code>) overrides the preset. Defaults to NULL (no preset), leaving existing calls unchanged.

<code>hc.rect.col</code>	the outline color of the <code>hc.rect</code> cluster rectangles. Defaults to "gray30"; set it to any color that suits your palette (e.g. "black" for a bolder box, or "white"). Only used when <code>hc.rect</code> is set.
<code>scale.square</code>	logical. If TRUE and <code>method = "square"</code> , the squares are sized by the absolute correlation (larger square = stronger correlation), in addition to the fill color – the classic corrplot size-scaled square look. Defaults to FALSE (constant full-cell squares, the current behavior). Has no effect for <code>method = "circle"</code> (circles are always sized). Uses <code>circle.scale</code> to tune the size range. As with <code>method = "circle"</code> , coefficient labels (<code>lab = TRUE</code>) are drawn at full size and may overflow the smallest squares.
<code>cell.grid</code>	logical. If TRUE, draw a light rectangle around every cell (behind the glyphs) and remove the through-center gridlines, so the sized glyphs (<code>method = "circle"</code> or <code>scale.square = TRUE</code>) sit inside boxed cells – the corrplot boxed-cell look. Defaults to FALSE (the current behavior). Has no effect on a full-tile square heatmap (<code>method = "square"</code> without <code>scale.square</code>), whose tiles already carry a cell border (<code>outline.color</code>).
<code>cell.grid.col</code>	the color of the <code>cell.grid</code> cell borders. Defaults to "grey90". Only used when <code>cell.grid = TRUE</code> .
<code>x</code>	numeric matrix or data frame
<code>...</code>	other arguments to be passed to the function <code>cor.test</code> .
<code>use</code>	character, how to treat pairs involving missing values when deciding which cells are NA. Either "pairwise.complete.obs" (default; test every pair that has enough overlapping observations) or "everything" (set a pair to NA as soon as either variable has a missing value, matching <code>cor</code> 's default). Mirrors the corresponding values of <code>cor</code> 's <code>use</code> argument.

Details

`cor_pmat()` tests each pair of columns with `cor.test`. A pair with fewer than three overlapping non-missing observations (which `cor.test` cannot test, e.g. two variables that never co-occur) yields NA for that cell rather than aborting the whole computation. Pairs that can be tested are computed as before, and errors they raise are passed through.

The `use` argument controls which pairs are returned as NA so the p-value matrix can be aligned with a correlation matrix built the same way. With the default "pairwise.complete.obs" every pair that has enough overlapping observations is tested (the previous behavior). With "everything" a pair is set to NA whenever either variable has any missing value, so the NA pattern matches `cor(x)` with its default `use = "everything"`.

Value

- `ggcorrplot()`: Returns a `ggplot2`
- `cor_pmat()`: Returns a matrix containing the p-values of correlations

See Also

ggcorrplot: Correlation Matrix Heatmap in R with ggplot2 for worked examples of the plot, and **Correlation Matrix in R: Compute, Visualize & P-values** for computing the matrix and its p-values beforehand.

Correlation Test in R: Pearson, Spearman & Kendall for the test behind the p-values.

Examples

```
# Compute a correlation matrix
data(mtcars)
corr <- round(cor(mtcars), 1)
corr

# Compute a matrix of correlation p-values
p.mat <- cor_pmat(mtcars)
p.mat

# Visualize the correlation matrix
# -----
# method = "square" or "circle"
ggcorrplot(corr)
ggcorrplot(corr, method = "circle")

# Mixed layout: a different glyph per triangle
# -----
# numbers in the lower triangle, circles in the upper, names on the diagonal
ggcorrplot(corr,
  lower.method = "number", upper.method = "circle",
  show.legend = FALSE
)

# Reordering the correlation matrix
# -----
# using hierarchical clustering
ggcorrplot(corr, hc.order = TRUE, outline.color = "white")
# draw rectangles around the clusters
ggcorrplot(corr, hc.order = TRUE, hc.rect = 3, outline.color = "white")

# Types of correlogram layout
# -----
# Get the lower triangle
ggcorrplot(corr,
  hc.order = TRUE, type = "lower",
  outline.color = "white"
)
# Get the upper triangle
ggcorrplot(corr,
  hc.order = TRUE, type = "upper",
  outline.color = "white"
)

# Change colors and theme
# -----
# Argument colors
ggcorrplot(corr,
  hc.order = TRUE, type = "lower",
  outline.color = "white",
```

```

    ggtheme = ggplot2::theme_gray,
    colors = c("#6D9EC1", "white", "#E46726")
  )

  # Add correlation coefficients
  # -----
  # argument lab = TRUE
  ggcorrplot(corr,
    hc.order = TRUE, type = "lower",
    lab = TRUE,
    ggtheme = ggplot2::theme_dark(),
  )

  # Add correlation significance level
  # -----
  # Argument p.mat
  # Barring the no significant coefficient
  ggcorrplot(corr,
    hc.order = TRUE,
    type = "lower", p.mat = p.mat
  )
  # Leave blank on no significant coefficient
  ggcorrplot(corr,
    p.mat = p.mat, hc.order = TRUE,
    type = "lower", insig = "blank"
  )

  # Changing number of digits for correlation coefficient
  # -----
  ggcorrplot(corr(mtcars),
    type = "lower",
    insig = "blank",
    lab = TRUE,
    digits = 3
  )

```

Index

`coord_fixed`, [5](#)
`cor`, [6](#)
`cor.test`, [6](#)
`cor_pmat` (`ggcorrplot`), [2](#)

`element_text`, [4](#)

`format`, [5](#)

`ggcorrplot`, [2](#)

`scale_fill_gradient2`, [3](#)
`scale_fill_gradientn`, [3](#)

`theme_set`, [3](#)