

Package ‘circlecorR’

July 28, 2026

Title Circular Correlation Wheel Plots

Version 0.1.0

Description Draws circular “correlation wheel” plots in R, straight from a data frame of one row per observation. Variables are arranged around a circle, grouped and colour-tiled by category, and connected by curved links whose colour maps to the correlation coefficient, computed via a thin wrapper around `corr.test()` from the ‘psych’ package. Non-significant, weak, and within-category correlations can be masked. Categories, colours, and labels are all user-configurable.

License MIT + file LICENSE

Language en-GB

Encoding UTF-8

Depends R (>= 4.1.0)

Imports circlize (>= 0.4.0), graphics, psych, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

URL <https://kriz98.github.io/circlecorR/>,
<https://github.com/kriz98/circlecorR>

BugReports <https://github.com/kriz98/circlecorR/issues>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Chris Varghese [aut, cre],
Armen A. Gharibans [aut]

Maintainer Chris Varghese <chris.varghese@auckland.ac.nz>

Repository CRAN

Date/Publication 2026-07-28 16:20:09 UTC

Contents

compute_correlations	2
corr_wheel	3
corr_wheel_scheme	6
corr_wheel_schemes	7
gastro_symptoms	8
Index	9

compute_correlations	<i>Compute a correlation matrix and (raw) p-values from raw data</i>
----------------------	--

Description

A thin wrapper around `psych::corr.test()` that returns the correlation (r) and **unadjusted** p-value (p) matrices bundled in a single object suitable for passing straight to `corr_wheel()`.

Usage

```
compute_correlations(
  data,
  vars = NULL,
  method = c("pearson", "spearman", "kendall"),
  use = "pairwise.complete.obs"
)
```

Arguments

<code>data</code>	A data frame or matrix of observations (rows) by variables (columns).
<code>vars</code>	Optional character vector selecting and ordering the columns of data to use. Defaults to all columns.
<code>method</code>	Correlation method, passed to <code>psych::corr.test()</code> . One of "pearson", "spearman", "kendall".
<code>use</code>	Handling of missing values, passed to <code>psych::corr.test()</code> .

Details

Multiple-comparison adjustment is deliberately **not** applied here. On a correlation wheel, self-correlations and (usually) within-category correlations are never tested, so they should not count towards the family of comparisons. `corr_wheel()` therefore applies the adjustment itself, over exactly the set of correlations it displays – which is both statistically consistent and more powerful than adjusting across the full matrix. See the `adjust` and `hide_within_group` arguments of `corr_wheel()`.

Value

An object of class "circlecor": a list with elements `r` (the correlation matrix), `p` (a symmetric matrix of **raw**, unadjusted p-values), `n` (the pairwise sample sizes from `psych`), and `method`.

See Also[corr_wheel\(\)](#)**Examples**

```
cc <- compute_correlations(mtcars, method = "pearson")
str(cc)
```

`corr_wheel`*Draw a circular correlation wheel plot*

Description

Arranges variables around a circle, grouped and colour-tiled by category, and connects them with curved links whose colour maps to the correlation coefficient. Non-significant, weak, and (optionally) within-category correlations are masked. This reproduces the MNE-style "connectivity circle" natively in R using **circize**.

Usage

```
corr_wheel(  
  data,  
  groups = NULL,  
  scheme = NULL,  
  colors = NULL,  
  labels = NULL,  
  order = NULL,  
  sig_level = 0.05,  
  r_threshold = 0,  
  hide_within_group = TRUE,  
  method = c("pearson", "spearman", "kendall"),  
  adjust = "holm",  
  use = "pairwise.complete.obs",  
  r_limits = c(-0.5, 0.5),  
  palette = NULL,  
  start_degree = 90,  
  group_gap = 5,  
  node_gap = 1.5,  
  link_lwd = 1.6,  
  sort_links = TRUE,  
  tile_height = 0.06,  
  label_cex = 0.85,  
  label_pad = 0.45,  
  label_r_offset = 0.07,  
  tile_border = "white",  
  title = NULL,  
  legend = TRUE,
```

```

colorbar = TRUE,
legend_title = "Category",
colorbar_title = "Correlation\ncoefficient"
)

```

Arguments

data	A data frame or matrix with one row per observation (e.g. one row per patient) and variables in columns. Correlations and p-values are computed for you via compute_correlations() using method, adjust, and use. When groups is supplied, only the variables it names are used (in that order), so extra columns such as an ID are simply ignored.
groups	Category assignment for the variables. Either a named vector (variable = category) or a named list (category = c(variables)). The order of categories here sets their order around the wheel. If NULL, all variables share one group.
scheme	A colour scheme providing the base category colours and diverging link palette together. One of: • NULL (default) – use the package default scheme; • a built-in scheme name, see corr_wheel_schemes() (e.g. "colorblind", "ocean", "vivid", "alimetry"); • a custom list(colors = , palette =), as returned by corr_wheel_scheme() (optionally tweaked). colors and palette (below), if supplied, override the scheme's corresponding piece – so you can pick a scheme and still tweak one category's colour, for instance.
colors	Named vector mapping category to colour, layered on top of scheme (or the default palette if scheme is NULL). Only the categories you name are overridden; others keep the scheme's colour.
labels	Named vector mapping variable name to a display label. Missing entries fall back to the variable name.
order	Optional character vector giving an explicit variable order around the wheel. Overrides ordering by groups. Must contain every variable.
sig_level	Significance threshold; links with adjusted $p > \text{sig_level}$ are hidden.
r_threshold	Minimum absolute correlation to display.
hide_within_group	Logical; if TRUE (default) correlations between two variables in the same category are hidden – and excluded from the multiple-comparison family (see Details). Self-correlations (the diagonal) are always excluded.
method, use	Passed to compute_correlations() : the correlation method and the missing-value handling.
adjust	Multiple-comparison adjustment method applied to the raw p-values over the displayed family of correlations (see Details). Any method accepted by stats::p.adjust() (e.g. "holm", "hochberg", "BH", "bonferroni", "none").

<code>r_limits</code>	Length-2 numeric giving the colour-scale limits (<code>c(vmin, vmax)</code>). Correlations beyond these are clamped for colour.
<code>palette</code>	Colours for the diverging link scale at <code>c(r_limits[1], midpoint, r_limits[2])</code> , overriding scheme's. NULL (default) uses the scheme's palette, or a blue-white-red scale if scheme is also NULL.
<code>start_degree</code>	Angle (degrees) of the first variable; 90 places it at the top, going clockwise.
<code>group_gap, node_gap</code>	Gaps (degrees) between categories and between variables within a category.
<code>link_lwd</code>	Line width (size) of the links. Increase for thicker lines.
<code>sort_links</code>	If TRUE (default), stronger correlations are drawn last (on top).
<code>tile_height</code>	Radial thickness (size) of the coloured category blocks. Smaller values give thinner blocks at the rim.
<code>label_cex, tile_border</code>	Label text size and tile border colour.
<code>label_pad</code>	Extra canvas padding (as a fraction of the circle radius) to keep long outer labels from being clipped. Increase for longer labels.
<code>label_r_offset</code>	Radial gap (in circle-radius units) between the outer edge of the tiles and the start of the labels.
<code>title</code>	Optional plot title.
<code>legend, colorbar</code>	Logical toggles for the category legend and the correlation colour bar.
<code>legend_title, colorbar_title</code>	Titles for the legend and colour bar.

Details

A traditional correlation matrix is dominated by redundant information: the diagonal of self-correlations, the mirror-image lower triangle, and blocks of within-category correlations that are rarely of interest. The wheel keeps only the correlations you actually want to inspect – by default the **between-category** correlations, with self- and within-category correlations hidden.

This is not only a display choice; it is carried through to the statistics. The multiple-comparison adjustment (`adjust`) is applied over **only the family of correlations shown** – i.e. excluding self- and, when `hide_within_group = TRUE`, within-category correlations. Because those redundant comparisons no longer count towards the family, the correction is less severe and statistical power improves.

Value

Invisibly, a list with the ordered vars, resolved groups, colors, the `col_fun` colour mapping, the masked matrix of correlations actually drawn (others NA), the family-adjusted p-value matrix `p_adjusted`, the number of drawn links `n_links`, the size of the comparison family `n_tests`, and the `adjust` method applied.

See Also

[compute_correlations\(\)](#), [corr_wheel_schemes\(\)](#), [corr_wheel_scheme\(\)](#)

Examples

```
grp <- list(
  Demographics = c("Age", "BMI"),
  Metrics       = c("Amplitude", "Fed-Fasted AR", "Frequency", "GA-RI"),
  Symptoms      = c("Nausea", "Early satiety", "Bloating", "Upper GI pain",
                    "Lower GI pain", "Heartburn"),
  Scores        = c("GCSI", "PAGI-SYM", "PAGI-QoL", "EQ-5D")
)

# `gastro_symptoms` is a synthetic example dataset bundled with the package
# (available directly after library(circlecorR) -- no need to call data()).
corr_wheel(gastro_symptoms, groups = grp, r_threshold = 0.3,
           r_limits = c(-0.6, 0.6))

# A built-in colour scheme, with one category colour overridden
corr_wheel(gastro_symptoms, groups = grp, r_threshold = 0.3,
           scheme = "colorblind", colors = c(Scores = "black"))
```

corr_wheel_scheme	<i>Get a built-in colour scheme</i>
-------------------	-------------------------------------

Description

Returns the category-colour and diverging-link-palette definition for a built-in scheme, so you can inspect it or tweak a copy before passing it to [corr_wheel\(\)](#)'s `scheme` argument.

Usage

```
corr_wheel_scheme(name = "default")
```

Arguments

`name` A scheme name; see [corr_wheel_schemes\(\)](#) for the available options.

Value

A list with elements `colors` (an unnamed vector of category colours, cycled across however many categories are plotted) and `palette` (a length-3 diverging colour vector: negative, midpoint, positive).

See Also

[corr_wheel_schemes\(\)](#), [corr_wheel\(\)](#)

Examples

```
s <- corr_wheel_scheme("colorblind")
s$palette

# Tweak a copy and use it
s$palette[2] <- "grey95"
data(gastro_symptoms)
grp <- list(Demographics = c("Age", "BMI"),
           Metrics = c("Amplitude", "Fed-Fasted AR", "Frequency", "GA-RI"))
if (requireNamespace("psych", quietly = TRUE)) {
  corr_wheel(gastro_symptoms, groups = grp, scheme = s, r_threshold = 0)
}
```

corr_wheel_schemes	<i>List built-in colour schemes</i>
--------------------	-------------------------------------

Description

Returns the names of the colour schemes bundled with the package, for use as the scheme argument of [corr_wheel\(\)](#).

Usage

```
corr_wheel_schemes()
```

Value

A character vector of scheme names.

See Also

[corr_wheel_scheme\(\)](#), [corr_wheel\(\)](#)

Examples

```
corr_wheel_schemes()
```

gastro_symptoms	<i>Synthetic gastric-symptom dataset</i>
-----------------	--

Description

A fully simulated dataset (not patient data) mimicking the structure of a gastric-symptom study: demographics, gastric metrics, symptom sub-scales, and summary scores for 100 imaginary subjects, one row per subject. Loadings are deliberately moderate and uneven, so only a realistic subset of correlations is significant. Generated by `data-raw/make_example_data.R`.

Usage

```
gastro_symptoms
```

Format

A data frame with 100 rows and 16 variables:

Age, BMI Demographics.

Amplitude, Fed-Fasted AR, Frequency, GA-RI Simulated gastric metrics.

Nausea, Early satiety, Bloating, Upper GI pain, Lower GI pain, Heartburn Symptom sub-scale scores.

GCSI, PAGI-SYM, PAGI-QoL, EQ-5D Summary symptom and quality-of-life scores.

Details

Bundled with the package (`LazyData: true`), so it is available directly as `gastro_symptoms` as soon as you `library(circlecorR)` – no `data()` call needed. Pass it straight to `corr_wheel()`; see `vignette("circlecorR")` for a worked example.

See Also

`corr_wheel()`, `compute_correlations()`

Index

* datasets

gastro_symptoms, 8

compute_correlations, 2

compute_correlations(), 4, 5, 8

corr_wheel, 3

corr_wheel(), 2, 3, 6–8

corr_wheel_scheme, 6

corr_wheel_scheme(), 4, 5, 7

corr_wheel_schemes, 7

corr_wheel_schemes(), 4–6

gastro_symptoms, 8

psych::corr.test(), 2

stats::p.adjust(), 4