Package ‘corrr’

March 22, 2020

Type Package

Version 0.4.2

Title Correlations in R

Description A tool for exploring correlations. It makes it possible to easily perform routine tasks when exploring correlation matrices such as ignoring the diagonal, focusing on the correlations of certain variables against others, or rearranging and visualizing the matrix in terms of the strength of the correlations.

Depends R (>= 3.3.0)

Imports dplyr (>= 0.8), ggplot2 (>= 2.2.0), seriation (>= 1.2-0), purrr (>= 0.2.2), tibble (>= 2.0), ggrepel (>= 0.6.5), methods (>= 3.4.3), rlang (>= 0.4.0)

Suggests testthat (>= 2.1.0), knitr (>= 1.13), rmarkdown (>= 0.9.6), dbplyr (>= 1.2.1), DBI, RSQLite, sparklyr (>= 0.9), covr

VignetteBuilder knitr

Encoding UTF-8

LazyData yes

License MIT + file LICENSE

URL `https://github.com/tidymodels/corrr`

BugReports `https://github.com/tidymodels/corrr/issues`

RoxygenNote 7.0.2.9000

NeedsCompilation no

Author Max Kuhn [aut, cre],
Simon Jackson [aut],
Jorge Cimentada [aut]

Maintainer Max Kuhn `<max@rstudio.com>`

Repository CRAN

Date/Publication 2020-03-22 06:40:15 UTC

R topics documented:

`as_cordf` *Coerce lists and matrices to correlation data frames*

Description

A wrapper function to coerce objects in a valid format (such as correlation matrices created using the base function, `cor`) into a correlation data frame.

Usage

```
as_cordf(x, diagonal = NA)
```

Arguments

`x`	A list, data frame or matrix that can be coerced into a correlation data frame.
`diagonal`	Value (typically numeric or NA) to set the diagonal to.

Value

A correlation data frame

Examples

```
x <- cor(mtcars)
as_cordf(x)
as_cordf(x, diagonal = 1)
```

`as_matrix` *Convert a correlation data frame to matrix format*

Description

Convert a correlation data frame to original matrix format.

Usage

```
as_matrix(x, diagonal)
```

Arguments

`x`	A correlation data frame. See `correlate` or `as_cordf`.
`diagonal`	Value (typically numeric or NA) to set the diagonal to.

Value

Correlation matrix

Examples

```
x <- correlate(mtcars)
as_matrix(x)
```

`correlate` *Correlation Data Frame*

Description

An implementation of stats::cor(), which returns a correlation data frame rather than a matrix. See details below. Additional adjustment include the use of pairwise deletion by default.

Usage

```
correlate(
  x,
  y = NULL,
  use = "pairwise.complete.obs",
  method = "pearson",
  diagonal = NA,
  quiet = FALSE
)
```

Arguments

x a numeric vector, matrix or data frame.

y NULL (default) or a vector, matrix or data frame with compatible dimensions to x. The default is equivalent to y = x (but more efficient).

use an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs".

method a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated.

diagonal Value (typically numeric or NA) to set the diagonal to.

quiet Set as TRUE to suppress message about ‘method‘ and ‘use‘ parameters.

Details

This function returns a correlation matrix as a correlation data frame in the following format:

A tibble (see tibble)

- An additional class, "cor_df"
- A "rowname" column
- Standardized variances (the matrix diagonal) set to missing values by default (NA) so they can be ignored in calculations.

Value

A correlation data frame (cor_df)

Examples

```
## Not run:
correlate(iris)

## End(Not run)

correlate(iris[-5])

correlate(mtcars)

## Not run:

# Also supports DB backend and collects results into memory

library(sparklyr)
sc <- spark_connect(master = "local")
mtcars_tbl <- copy_to(sc, mtcars)
mtcars_tbl %>%
  correlate(use = "pairwise.complete.obs", method = "spearman")
spark_disconnect(sc)
```

```
## End(Not run)
```

dice — *Returns a correlation table with the selected fields only*

Description

Returns a correlation table with the selected fields only

Usage

```
dice(x, ...)
```

Arguments

`x`	A correlation table, class cor_df
`...`	A list of variables in the correlation table

Examples

```
dice(correlate(mtcars), mpg, wt, am)
```

fashion — *Fashion a correlation data frame for printing.*

Description

For the purpose of printing, convert a correlation data frame into a noquote matrix with the correlations cleanly formatted (leading zeros removed; spaced for signs) and the diagonal (or any NA) left blank.

Usage

```
fashion(x, decimals = 2, leading_zeros = FALSE, na_print = "")
```

Arguments

`x`	Scalar, vector, matrix or data frame.
`decimals`	Number of decimal places to display for numbers.
`leading_zeros`	Should leading zeros be displayed for decimals (e.g., 0.1)? If FALSE, they will be removed.
`na_print`	Character string indicating NA values in printed output

Value

noquote. Also a data frame if x is a matrix or data frame.

Examples

```
# Examples with correlate()
library(dplyr)
mtcars %>% correlate() %>% fashion()
mtcars %>% correlate() %>% fashion(decimals = 1)
mtcars %>% correlate() %>% fashion(leading_zeros = TRUE)
mtcars %>% correlate() %>% fashion(na_print = "*")

# But doesn't have to include correlate()
mtcars %>% fashion(decimals = 3)
c(0.234, 134.23, -.23, NA) %>% fashion(na_print = "X")
```

first_col *Add a first column to a data.frame*

Description

Add a first column to a data.frame. This is most commonly used to append a rowname column to create a cor_df.

Usage

```
first_col(df, ..., var = "rowname")
```

Arguments

df	Data frame
...	Values to go into the column
var	Label for the column. Default is "rowname"

Examples

```
first_col(mtcars, 1:nrow(mtcars))
```

focus — *Focus on section of a correlation data frame.*

Description

Convenience function to select a set of variables from a correlation matrix to keep as the columns, and exclude these or all other variables from the rows. This function will take a `correlate` correlation matrix, and expression(s) suited for dplyr::select(). The selected variables will remain in the columns, and these, or all other variables, will be excluded from the rows based on ‘`same`. For a complete list of methods for using this function, see `select`.

Usage

```
focus(x, ..., mirror = FALSE)

focus_(x, ..., .dots, mirror)
```

Arguments

`x`	cor_df. See `correlate`.
`...`	One or more unquoted expressions separated by commas. Variable names can be used as if they were positions in the data frame, so expressions like ‘x:y“ can be used to select a range of variables.
`mirror`	Boolean. Whether to mirror the selected columns in the rows or not.
`.dots`	Use focus_ to do standard evaluations. See `select`.

Value

A tbl or, if mirror = TRUE, a cor_df (see `correlate`).

Examples

```
library(dplyr)
x <- correlate(mtcars)
focus(x, mpg, cyl)  # Focus on correlations of mpg and cyl with all other variables
focus(x, -disp, - mpg, mirror = TRUE)  # Remove disp and mpg from columns and rows

x <- correlate(iris[-5])
focus(x, -matches("Sepal"))  # Focus on correlations of non-Sepal
                             # variables with Sepal variables.
```

focus_if *Conditionally focus correlation data frame*

Description

Apply a predicate function to each column of correlations. Columns that evaluate to TRUE will be included in a call to `focus`.

Usage

```
focus_if(x, .predicate, ..., mirror = FALSE)
```

Arguments

`x`	Correlation data frame or object to be coerced to one via `as_cordf`.
`.predicate`	A predicate function to be applied to the columns. The columns for which .predicate returns TRUE will be included as variables in `focus`.
`...`	Additional arguments to pass to the predicate function if not anonymous.
`mirror`	Boolean. Whether to mirror the selected columns in the rows or not.

Value

A tibble or, if mirror = TRUE, a correlation data frame.

Examples

```
library(dplyr)
any_greater_than <- function(x, val) {
  mean(abs(x), na.rm = TRUE) > val
}

x <- correlate(mtcars)

x %>% focus_if(any_greater_than, .6)
x %>% focus_if(any_greater_than, .6, mirror = TRUE) %>% network_plot()
```

network_plot *Network plot of a correlation data frame*

Description

Output a network plot of a correlation data frame in which variables that are more highly correlated appear closer together and are joined by stronger paths. Paths are also colored by their sign (blue for positive and red for negative). The proximity of the points are determined using multidimensional clustering.

Usage

```
network_plot(
  rdf,
  min_cor = 0.3,
  legend = TRUE,
  colours = c("indianred2", "white", "skyblue1"),
  repel = TRUE,
  curved = TRUE,
  colors
)
```

Arguments

`rdf` Correlation data frame (see `correlate`) or object that can be coerced to one (see `as_cordf`).

`min_cor` Number from 0 to 1 indicating the minimum value of correlations (in absolute terms) to plot.

`legend` Boolean indicating whether a legend mapping the colors to the correlations should be displayed.

`colours, colors` Vector of colors to use for n-color gradient.

`repel` Should variable labels repel each other? If TRUE, text is added via `geom_text_repel` instead of `geom_text`

`curved` Should the paths be curved? If TRUE, paths are added via `geom_curve`; if FALSE, via `geom_segment`

Examples

```
x <- correlate(mtcars)
network_plot(x)
network_plot(x, min_cor = .1)
network_plot(x, min_cor = .6)
network_plot(x, min_cor = .7, colors = c("red", "green"), legend = TRUE)
```

`pair_n` *Number of pairwise complete cases.*

Description

Compute the number of complete cases in a pairwise fashion for `x` (and `y`).

Usage

```
pair_n(x, y = NULL)
```

Arguments

`x` a numeric vector, matrix or data frame.

`y` `NULL` (default) or a vector, matrix or data frame with compatible dimensions to `x`. The default is equivalent to `y = x` (but more efficient).

Value

Matrix of pairwise sample sizes (number of complete cases).

Examples

```
pair_n(mtcars)
```

`rearrange` *Re-arrange a correlation data frame*

Description

Re-arrange a correlation data frame to group highly correlated variables closer together.

Usage

```
rearrange(x, method = "PC", absolute = TRUE)
```

Arguments

`x` cor_df. See `correlate`.

`method` String specifying the arrangement (clustering) method. Clustering is achieved via `seriate`, which can be consulted for a complete list of clustering methods. Default = "PCA".

`absolute` Boolean whether absolute values for the correlations should be used for clustering.

Value

cor_df. See `correlate`.

Examples

```
x <- correlate(mtcars)

rearrange(x) # Default settings
rearrange(x, method = "HC")  # Different seriation method
rearrange(x, absolute = FALSE)  # Not using absolute values for arranging
```

`retract` *Creates a data frame from a stretched correlation table*

Description

`retract` does the opposite of what `stretch` does

Usage

```
retract(.data, x, y, val)
```

Arguments

`.data`	A data.frame or tibble containing at least three variables: x, y and the value
`x`	The name of the column to use from .data as x
`y`	The name of the column to use from .data as y
`val`	The name of the column to use from .data to use as the value

Examples

```
x <- correlate(mtcars)
xs <- stretch(x)
retract(xs)
```

`rplot` *Plot a correlation data frame.*

Description

Plot a correlation data frame using ggplot2.

Usage

```
rplot(
  rdf,
  legend = TRUE,
  shape = 16,
  colours = c("indianred2", "white", "skyblue1"),
  print_cor = FALSE,
  colors
)
```

Arguments

`rdf` Correlation data frame (see `correlate`) or object that can be coerced to one (see `as_cordf`).

`legend` Boolean indicating whether a legend mapping the colors to the correlations should be displayed.

`shape` `geom_point` aesthetic.

`colours, colors`
Vector of colors to use for n-color gradient.

`print_cor` Boolean indicating whether the correlations should be printed over the shapes.

Value

Plots a correlation data frame

Examples

```
x <- correlate(mtcars)
rplot(x)

# Common use is following rearrange and shave
x <- rearrange(x, absolute = FALSE)
x <- shave(x)
rplot(x)
rplot(x, print_cor = TRUE)
rplot(x, shape = 20, colors = c("red", "green"), legend = TRUE)
```

`shave` *Shave off upper/lower triangle.*

Description

Convert the upper or lower triangle of a correlation data frame (cor_df) to missing values.

Usage

```
shave(x, upper = TRUE)
```

Arguments

`x` cor_df. See `correlate`.

`upper` Boolean. If TRUE, set upper triangle to NA; lower triangle if FALSE.

Value

cor_df. See `correlate`.

Examples

```
x <- correlate(mtcars)
shave(x)  # Default; shave upper triangle
shave(x, upper = FALSE)  # shave lower triangle
```

`stretch` *Stretch correlation data frame into long format.*

Description

`stretch` is a specified implementation of tidyr::gather() to be applied to a correlation data frame. It will gather the columns into a long-format data frame. The rowname column is handled automatically.

Usage

```
stretch(x, na.rm = FALSE, remove.dups = FALSE)
```

Arguments

`x`	cor_df. See `correlate`.
`na.rm`	Boolean. Whether rows with an NA correlation (originally the matrix diagonal) should be dropped? Will automatically be set to TRUE if mirror is FALSE.
`remove.dups`	Removes duplicate entries, without removing all NAs

Value

tbl with three columns (x and y variables, and their correlation)

Examples

```
x <- correlate(mtcars)
stretch(x)  # Convert all to long format
stretch(x, na.rm = FALSE)  # omit NAs (diagonal in this case)

x <- shave(x)  # use shave to set upper triangle to NA and then...
stretch(x, na.rm = FALSE)  # omit all NAs, therefore keeping each
                             # correlation only once.
```

Index