Package 'correlation'

March 3, 2025

Type Package

Title Methods for Correlation Analysis

Version 0.8.7

Maintainer Brenton M. Wiernik <brenton@wiernik.org>

Description Lightweight package for computing different kinds of correlations, such as partial correlations, Bayesian correlations, multilevel correlations, polychoric correlations, biweight correlations, distance correlations and more. Part of the 'easystats' ecosystem. References: Makowski et al. (2020) <doi:10.21105/joss.02306>.

License MIT + file LICENSE

URL https://easystats.github.io/correlation/

BugReports https://github.com/easystats/correlation/issues

Depends R (>= 4.1)

Imports bayestestR (>= 0.15.0), datasets, datawizard (>= 1.0.0), insight (>= 1.0.0), parameters (>= 0.24.0), stats

Suggests BayesFactor, energy, ggplot2, ggraph, gt, Hmisc, knitr, lme4, MASS, mbend, polycor, poorman, ppcor, psych, rmarkdown, rmcorr, rstanarm, see (>= 0.8.1), testthat (>= 3.2.1), tidygraph, wdm, WRS2, openxlsx2 (>= 1.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

Config/testthat/edition 3

Config/Needs/website rstudio/bslib, r-lib/pkgdown, easystats/easystatstemplate

NeedsCompilation no

Author Dominique Makowski [aut, inv] (<https://orcid.org/0000-0001-5375-9967>), Brenton M. Wiernik [aut, cre] (<https://orcid.org/0000-0001-9560-6336>), Indrajeet Patil [aut] (<https://orcid.org/0000-0003-1995-6531>),

Daniel Lüdecke [aut] (<https://orcid.org/0000-0002-8895-3206>),
Mattan S. Ben-Shachar [aut] (<https://orcid.org/0000-0002-4287-4801>),
Rémi Thériault [aut] (<https://orcid.org/0000-0003-4315-6788>),
Mark White [rev],
Maximilian M. Rabe [rev] (<https://orcid.org/0000-0002-2556-5644>)

Repository CRAN

Date/Publication 2025-03-03 17:30:11 UTC

Contents

`cormatrix_to_excel` *Easy export of correlation matrix to Excel*

Description

Easily output a correlation matrix and export it to Microsoft Excel, with the first row and column frozen, and correlation coefficients colour-coded based on effect size (0.0-0.2: small (no colour); 0.2-0.4: medium (pink/light blue); 0.4-1.0: large (red/dark blue)), following Cohen's suggestions for small (.10), medium (.30), and large (.50) correlation sizes.

Usage

```
cormatrix_to_excel(data, filename, overwrite = TRUE, print.mat = TRUE, ...)
```

Arguments

`data`	The data frame
`filename`	Desired filename (path can be added before hand but no need to specify extension).
`overwrite`	Whether to allow overwriting previous file.
`print.mat`	Logical, whether to also print the correlation matrix to console.
`...`	Parameters to be passed to `correlation()`

Value

A Microsoft Excel document, containing the colour-coded correlation matrix with significance stars, on the first sheet, and the colour-coded p-values on the second sheet.

Author(s)

Adapted from @JanMarvin (JanMarvin/openxlsx2#286) and the original `rempsyc::cormatrix_excel`.

Examples

```
# Basic example
suppressWarnings(cormatrix_to_excel(mtcars,
  select = c("mpg", "cyl", "disp", "hp", "carb"), filename = "cormatrix1"
))
suppressWarnings(cormatrix_to_excel(iris,
  p_adjust = "none",
  filename = "cormatrix2"
))
suppressWarnings(cormatrix_to_excel(airquality,
  method = "spearman",
  filename = "cormatrix3"
))
```

`correlation` *Correlation Analysis*

Description

Performs a correlation analysis. You can easily visualize the result using `plot()` (see examples **here**).

Usage

```
correlation(
  data,
  data2 = NULL,
  select = NULL,
  select2 = NULL,
  rename = NULL,
  method = "pearson",
  p_adjust = "holm",
  ci = 0.95,
  bayesian = FALSE,
  bayesian_prior = "medium",
  bayesian_ci_method = "hdi",
  bayesian_test = c("pd", "rope", "bf"),
  redundant = FALSE,
  include_factors = FALSE,
  partial = FALSE,
  partial_bayesian = FALSE,
  multilevel = FALSE,
  ranktransform = FALSE,
  winsorize = FALSE,
  verbose = TRUE,
  standardize_names = getOption("easystats.standardize_names", FALSE),
  ...
)
```

Arguments

`data`	A data frame.
`data2`	An optional data frame. If specified, all pair-wise correlations between the variables in `data` and `data2` will be computed.
`select`, `select2`	(Ignored if `data2` is specified.) Optional names of variables that should be selected for correlation. Instead of providing the data frames with those variables that should be correlated, `data` can be a data frame and `select` and `select2` are (quoted) names of variables (columns) in `data`. `correlation()` will then compute the correlation between `data[select]` and `data[select2]`. If only `select` is specified, all pairwise correlations between the `select` variables will be computed. This is a "pipe-friendly" alternative way of using `correlation()` (see 'Examples').
`rename`	In case you wish to change the names of the variables in the output, these arguments can be used to specify these alternative names. Note that the number of names should be equal to the number of columns selected. Ignored if `data2` is specified.
`method`	A character string indicating which correlation coefficient is to be used for the test. One of `"pearson"` (default), `"kendall"`, `"spearman"` (but see also the `robust` argument), `"biserial"`, `"polychoric"`, `"tetrachoric"`, `"biweight"`, `"distance"`, `"percentage"` (for percentage bend correlation), `"blomqvist"`

(for Blomqvist's coefficient), "hoeffding" (for Hoeffding's D), "gamma", "gaussian" (for Gaussian Rank correlation) or "shepherd" (for Shepherd's Pi correlation). Setting "auto" will attempt at selecting the most relevant method (polychoric when ordinal factors involved, tetrachoric when dichotomous factors involved, point-biserial if one dichotomous and one continuous and pearson otherwise). See below the **details** section for a description of these indices.

p_adjust — Correction method for frequentist correlations. Can be one of "holm" (default), "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "somers" or "none". See stats::p.adjust() for further details.

ci — Confidence/Credible Interval level. If "default", then it is set to 0.95 (95% CI).

bayesian — If TRUE, will run the correlations under a Bayesian framework.

bayesian_prior — For the prior argument, several named values are recognized: "medium.narrow", "medium", "wide", and "ultrawide". These correspond to scale values of 1/sqrt(27), 1/3, 1/sqrt(3) and 1, respectively. See the BayesFactor::correlationBF function.

bayesian_ci_method, bayesian_test

See arguments in parameters::model_parameters() for BayesFactor tests.

redundant — Should the data include redundant rows (where each given correlation is repeated two times).

include_factors

If TRUE, the factors are kept and eventually converted to numeric or used as random effects (depending of multilevel). If FALSE, factors are removed upfront.

partial — Can be TRUE or "semi" for partial and semi-partial correlations, respectively.

partial_bayesian

If partial correlations under a Bayesian framework are needed, you will also need to set partial_bayesian to TRUE to obtain "full" Bayesian partial correlations. Otherwise, you will obtain pseudo-Bayesian partial correlations (i.e., Bayesian correlation based on frequentist partialization).

multilevel — If TRUE, the factors are included as random factors. Else, if FALSE (default), they are included as fixed effects in the simple regression model.

ranktransform — If TRUE, will rank-transform the variables prior to estimating the correlation, which is one way of making the analysis more resistant to extreme values (outliers). Note that, for instance, a Pearson's correlation on rank-transformed data is equivalent to a Spearman's rank correlation. Thus, using robust=TRUE and method="spearman" is redundant. Nonetheless, it is an easy option to increase the robustness of the correlation as well as flexible way to obtain Bayesian or multilevel Spearman-like rank correlations.

winsorize — Another way of making the correlation more "robust" (i.e., limiting the impact of extreme values). Can be either FALSE or a number between 0 and 1 (e.g., 0.2) that corresponds to the desired threshold. See the datawizard::winsorize() function for more details.

verbose — Toggle warnings.

standardize_names

This option can be set to TRUE to run insight::standardize_names() on the output to get standardized column names. This option can also be set globally by running options(easystats.standardize_names = TRUE).

... Additional arguments (e.g., `alternative`) to be passed to other methods. See `stats::cor.test` for further details.

Details

Correlation Types:

- **Pearson's correlation**: This is the most common correlation method. It corresponds to the covariance of the two variables normalized (i.e., divided) by the product of their standard deviations.
- **Spearman's rank correlation**: A non-parametric measure of rank correlation (statistical dependence between the rankings of two variables). The Spearman correlation between two variables is equal to the Pearson correlation between the rank values of those two variables; while Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships (whether linear or not). Confidence Intervals (CI) for Spearman's correlations are computed using the Fieller et al. (1957) correction (see Bishara and Hittner, 2017).
- **Kendall's rank correlation**: In the normal case, the Kendall correlation is preferred than the Spearman correlation because of a smaller gross error sensitivity (GES) and a smaller asymptotic variance (AV), making it more robust and more efficient. However, the interpretation of Kendall's tau is less direct than that of Spearman's rho, in the sense that it quantifies the difference between the percentage of concordant and discordant pairs among all possible pairwise events. Confidence Intervals (CI) for Kendall's correlations are computed using the Fieller et al. (1957) correction (see Bishara and Hittner, 2017).
- **Biweight midcorrelation**: A measure of similarity that is median-based, instead of the traditional mean-based, thus being less sensitive to outliers. It can be used as a robust alternative to other similarity metrics, such as Pearson correlation (Langfelder & Horvath, 2012).
- **Distance correlation**: Distance correlation measures both linear and non-linear association between two random variables or random vectors. This is in contrast to Pearson's correlation, which can only detect linear association between two random variables.
- **Percentage bend correlation**: Introduced by Wilcox (1994), it is based on a down-weight of a specified percentage of marginal observations deviating from the median (by default, `20%`).
- **Shepherd's Pi correlation**: Equivalent to a Spearman's rank correlation after outliers removal (by means of bootstrapped Mahalanobis distance).
- **Blomqvist's coefficient**: The Blomqvist's coefficient (also referred to as Blomqvist's Beta or medial correlation; Blomqvist, 1950) is a median-based non-parametric correlation that has some advantages over measures such as Spearman's or Kendall's estimates (see Shmid & Schimdt, 2006).
- **Hoeffding's D**: The Hoeffding's D statistics is a non-parametric rank based measure of association that detects more general departures from independence (Hoeffding 1948), including non-linear associations. Hoeffding's D varies between -0.5 and 1 (if there are no tied ranks, otherwise it can have lower values), with larger values indicating a stronger relationship between the variables.
- **Somers' D**: The Somers' D statistics is a non-parametric rank based measure of association between a binary variable and a continuous variable, for instance, in the context of logistic regression the binary outcome and the predicted probabilities for each outcome. Usually, Somers' D is a measure of ordinal association, however, this implementation it is limited to the case of a binary outcome.

- **Point-Biserial and biserial correlation**: Correlation coefficient used when one variable is continuous and the other is dichotomous (binary). Point-Biserial is equivalent to a Pearson's correlation, while Biserial should be used when the binary variable is assumed to have an underlying continuity. For example, anxiety level can be measured on a continuous scale, but can be classified dichotomously as high/low.
- **Gamma correlation**: The Goodman-Kruskal gamma statistic is similar to Kendall's Tau coefficient. It is relatively robust to outliers and deals well with data that have many ties.
- **Winsorized correlation**: Correlation of variables that have been formerly Winsorized, i.e., transformed by limiting extreme values to reduce the effect of possibly spurious outliers.
- **Gaussian rank Correlation**: The Gaussian rank correlation estimator is a simple and well-performing alternative for robust rank correlations (Boudt et al., 2012). It is based on the Gaussian quantiles of the ranks.
- **Polychoric correlation**: Correlation between two theorized normally distributed continuous latent variables, from two observed ordinal variables.
- **Tetrachoric correlation**: Special case of the polychoric correlation applicable when both observed variables are dichotomous.

Partial Correlation: **Partial correlations** are estimated as the correlation between two variables after adjusting for the (linear) effect of one or more other variable. The correlation test is then run after having partialized the dataset, independently from it. In other words, it considers partialization as an independent step generating a different dataset, rather than belonging to the same model. This is why some discrepancies are to be expected for the t- and p-values, CIs, BFs etc (but *not* the correlation coefficient) compared to other implementations (e.g., `ppcor`). (The size of these discrepancies depends on the number of covariates partialled-out and the strength of the linear association between all variables.) Such partial correlations can be represented as Gaussian Graphical Models (GGM), an increasingly popular tool in psychology. A GGM traditionally include a set of variables depicted as circles ("nodes"), and a set of lines that visualize relationships between them, which thickness represents the strength of association (see Bhushan et al., 2019).

Multilevel correlations are a special case of partial correlations where the variable to be adjusted for is a factor and is included as a random effect in a mixed model (note that the remaining continuous variables of the dataset will still be included as fixed effects, similarly to regular partial correlations). The model is a random intercept model, i.e. the multilevel correlation is adjusted for `(1 | groupfactor)`.That said, there is an important difference between using `cor_test()` and `correlation()`: If you set `multilevel=TRUE` in `correlation()` but `partial` is set to `FALSE` (as per default), then a back-transformation from partial to non-partial correlation will be attempted (through `pcor_to_cor()`). However, this is not possible when using `cor_test()` so that if you set `multilevel=TRUE` in it, the resulting correlations are partial one. Note that for Bayesian multilevel correlations, if `partial = FALSE`, the back transformation will also recompute *p*-values based on the new *r* scores, and will drop the Bayes factors (as they are not relevant anymore). To keep Bayesian scores, set `partial = TRUE`.

Notes: Kendall and Spearman correlations when `bayesian=TRUE`: These are technically Pearson Bayesian correlations of rank transformed data, rather than pure Bayesian rank correlations (which have different priors).

Value

A correlation object that can be displayed using the `print`, `summary` or `table` methods.

Multiple tests correction: The `p_adjust` argument can be used to adjust p-values for multiple comparisons. All adjustment methods available in `p.adjust` function `stats` package are supported.

References

- Boudt, K., Cornelissen, J., & Croux, C. (2012). The Gaussian rank correlation estimator: robustness properties. Statistics and Computing, 22(2), 471-483.
- Bhushan, N., Mohnert, F., Sloot, D., Jans, L., Albers, C., & Steg, L. (2019). Using a Gaussian graphical model to explore relationships between items and variables in environmental psychology research. Frontiers in psychology, 10, 1050.
- Bishara, A. J., & Hittner, J. B. (2017). Confidence intervals for correlations when data are not normal. Behavior research methods, 49(1), 294-309.
- Fieller, E. C., Hartley, H. O., & Pearson, E. S. (1957). Tests for rank correlation coefficients. I. Biometrika, 44(3/4), 470-481.
- Langfelder, P., & Horvath, S. (2012). Fast R functions for robust correlations and hierarchical clustering. Journal of statistical software, 46(11).
- Blomqvist, N. (1950). On a measure of dependence between two random variables,Annals of Mathematical Statistics,21, 593–600
- Somers, R. H. (1962). A new asymmetric measure of association for ordinal variables. American Sociological Review. 27 (6).

Examples

```
library(correlation)
data(iris)

results <- correlation(iris)

results
summary(results)
summary(results, redundant = TRUE)

# pipe-friendly usage with  grouped dataframes from {dplyr} package
iris |>
  correlation(select = "Petal.Width", select2 = "Sepal.Length")

# Grouped dataframe
# grouped correlations
iris |>
  datawizard::data_group(Species) |>
  correlation()

# selecting specific variables for correlation
data(mtcars)
mtcars |>
  datawizard::data_group(am) |>
  correlation(select = c("cyl", "wt"), select2 = "hp")
```

```
# supplying custom variable names
correlation(anscombe, select = c("x1", "x2"), rename = c("var1", "var2"))

# automatic selection of correlation method
correlation(mtcars[-2], method = "auto")
```

`correlation-deprecated` *Deprecated functions*

Description

Deprecated functions

Usage

```
distance_mahalanobis(...)
```

Arguments

`...`	Args.

`cor_lower` *Return the upper or lower triangular part*

Description

Return the upper or lower triangular part of the correlation matrix.

Usage

```
cor_lower(x, diag = FALSE, ...)
```

Arguments

`x`	A correlation object.
`diag`	Should the diagonal be included?
`...`	Other arguments to be passed to or from other functions.

Examples

```
x <- correlation(mtcars, redundant = TRUE) # Generate full matrix
x <- cor_lower(x)

if (require("ggplot2")) {
  ggplot(x, aes(x = Parameter2, y = Parameter1, fill = r)) +
    geom_tile()
}

# Sorted
x <- correlation(mtcars, redundant = TRUE) # Generate full matrix
x <- cor_sort(x)
x <- cor_lower(x)

if (require("ggplot2")) {
  ggplot(x, aes(x = Parameter2, y = Parameter1, fill = r)) +
    geom_tile()
}
```

---

cor_smooth *Smooth a non-positive definite correlation matrix to make it positive definite*

Description

Make correlations positive definite using `psych::cor.smooth`. If smoothing is done, inferential statistics (*p*-values, confidence intervals, etc.) are removed, as they are no longer valid.

Usage

```
cor_smooth(x, method = "psych", verbose = TRUE, ...)

is.positive_definite(x, tol = 10^-12, ...)

is_positive_definite(x, tol = 10^-12, ...)
```

Arguments

`x`	A correlation matrix.
`method`	Smoothing method. Can be `psych` (will use `psych::cor.smooth()`), `hj` (Jorjani et al., 2003) or `lrs` (Schaeffer, 2014). For the two last, will use `mbend::bend()` (check its documentation for details).
`verbose`	Set to `FALSE` to silence the function.
`...`	Other arguments to be passed to or from other functions.
`tol`	The minimum eigenvalue to be considered as acceptable.

Examples

```
set.seed(123)
data <- as.matrix(mtcars)
# Make missing data so pairwise correlation matrix is non-positive definite
data[sample(seq_len(352), size = 60)] <- NA
data <- as.data.frame(data)
x <- correlation(data)
is.positive_definite(x)

smoothed <- cor_smooth(x)
```

---

cor_sort *Sort a correlation matrix to improve readability of groups and clusters*

Description

Sort a correlation matrix based on `hclust()`.

Usage

```
cor_sort(x, distance = "correlation", hclust_method = "complete", ...)
```

Arguments

x	A correlation matrix.
distance	How the distance between each variable should be calculated. If `correlation` (default; suited for correlation matrices), the matrix will be rescaled to 0-1 (`distance = 0` indicating correlation of 1; `distance = 1` indicating correlation of -1). If `raw`, then the matrix will be used as a distance matrix as-is. Can be others (`euclidean`, `manhattan`, ...), in which case it will be passed to `dist()` (see the arguments for it).
hclust_method	Argument passed down into the `method` argument of `hclust()`.
...	Other arguments to be passed to or from other functions.

Examples

```
x <- correlation(mtcars)

cor_sort(as.matrix(x))
cor_sort(x, hclust_method = "ward.D2") # It can also reorder the long form output
cor_sort(summary(x, redundant = TRUE)) # As well as from the summary
```

`cor_test` *Correlation test*

Description

This function performs a correlation test between two variables. You can easily visualize the result using `plot()` (see examples **here**).

Usage

```
cor_test(
  data,
  x,
  y,
  method = "pearson",
  ci = 0.95,
  bayesian = FALSE,
  bayesian_prior = "medium",
  bayesian_ci_method = "hdi",
  bayesian_test = c("pd", "rope", "bf"),
  include_factors = FALSE,
  partial = FALSE,
  partial_bayesian = FALSE,
  multilevel = FALSE,
  ranktransform = FALSE,
  winsorize = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`data`	A data frame.
`x`, `y`	Names of two variables present in the data.
`method`	A character string indicating which correlation coefficient is to be used for the test. One of `"pearson"` (default), `"kendall"`, `"spearman"` (but see also the `robust` argument), `"biserial"`, `"polychoric"`, `"tetrachoric"`, `"biweight"`, `"distance"`, `"percentage"` (for percentage bend correlation), `"blomqvist"` (for Blomqvist's coefficient), `"hoeffding"` (for Hoeffding's D), `"gamma"`, `"gaussian"` (for Gaussian Rank correlation) or `"shepherd"` (for Shepherd's Pi correlation). Setting `"auto"` will attempt at selecting the most relevant method (polychoric when ordinal factors involved, tetrachoric when dichotomous factors involved, point-biserial if one dichotomous and one continuous and pearson otherwise). See below the **details** section for a description of these indices.
`ci`	Confidence/Credible Interval level. If `"default"`, then it is set to `0.95` (`95%` CI).
`bayesian`	If `TRUE`, will run the correlations under a Bayesian framework.

bayesian_prior For the prior argument, several named values are recognized: "medium.narrow", "medium", "wide", and "ultrawide". These correspond to scale values of 1/sqrt(27), 1/3, 1/sqrt(3) and 1, respectively. See the BayesFactor::correlationBF function.

bayesian_ci_method, bayesian_test
See arguments in parameters::model_parameters() for BayesFactor tests.

include_factors
If TRUE, the factors are kept and eventually converted to numeric or used as random effects (depending of multilevel). If FALSE, factors are removed upfront.

partial Can be TRUE or "semi" for partial and semi-partial correlations, respectively.

partial_bayesian
If partial correlations under a Bayesian framework are needed, you will also need to set partial_bayesian to TRUE to obtain "full" Bayesian partial correlations. Otherwise, you will obtain pseudo-Bayesian partial correlations (i.e., Bayesian correlation based on frequentist partialization).

multilevel If TRUE, the factors are included as random factors. Else, if FALSE (default), they are included as fixed effects in the simple regression model.

ranktransform If TRUE, will rank-transform the variables prior to estimating the correlation, which is one way of making the analysis more resistant to extreme values (outliers). Note that, for instance, a Pearson's correlation on rank-transformed data is equivalent to a Spearman's rank correlation. Thus, using robust=TRUE and method="spearman" is redundant. Nonetheless, it is an easy option to increase the robustness of the correlation as well as flexible way to obtain Bayesian or multilevel Spearman-like rank correlations.

winsorize Another way of making the correlation more "robust" (i.e., limiting the impact of extreme values). Can be either FALSE or a number between 0 and 1 (e.g., 0.2) that corresponds to the desired threshold. See the datawizard::winsorize() function for more details.

verbose Toggle warnings.

... Additional arguments (e.g., alternative) to be passed to other methods. See stats::cor.test for further details.

Details

Correlation Types:

- **Pearson's correlation**: This is the most common correlation method. It corresponds to the covariance of the two variables normalized (i.e., divided) by the product of their standard deviations.
- **Spearman's rank correlation**: A non-parametric measure of rank correlation (statistical dependence between the rankings of two variables). The Spearman correlation between two variables is equal to the Pearson correlation between the rank values of those two variables; while Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships (whether linear or not). Confidence Intervals (CI) for Spearman's correlations are computed using the Fieller et al. (1957) correction (see Bishara and Hittner, 2017).

- **Kendall's rank correlation**: In the normal case, the Kendall correlation is preferred than the Spearman correlation because of a smaller gross error sensitivity (GES) and a smaller asymptotic variance (AV), making it more robust and more efficient. However, the interpretation of Kendall's tau is less direct than that of Spearman's rho, in the sense that it quantifies the difference between the percentage of concordant and discordant pairs among all possible pairwise events. Confidence Intervals (CI) for Kendall's correlations are computed using the Fieller et al. (1957) correction (see Bishara and Hittner, 2017).
- **Biweight midcorrelation**: A measure of similarity that is median-based, instead of the traditional mean-based, thus being less sensitive to outliers. It can be used as a robust alternative to other similarity metrics, such as Pearson correlation (Langfelder & Horvath, 2012).
- **Distance correlation**: Distance correlation measures both linear and non-linear association between two random variables or random vectors. This is in contrast to Pearson's correlation, which can only detect linear association between two random variables.
- **Percentage bend correlation**: Introduced by Wilcox (1994), it is based on a down-weight of a specified percentage of marginal observations deviating from the median (by default, `20%`).
- **Shepherd's Pi correlation**: Equivalent to a Spearman's rank correlation after outliers removal (by means of bootstrapped Mahalanobis distance).
- **Blomqvist's coefficient**: The Blomqvist's coefficient (also referred to as Blomqvist's Beta or medial correlation; Blomqvist, 1950) is a median-based non-parametric correlation that has some advantages over measures such as Spearman's or Kendall's estimates (see Shmid & Schimdt, 2006).
- **Hoeffding's D**: The Hoeffding's D statistics is a non-parametric rank based measure of association that detects more general departures from independence (Hoeffding 1948), including non-linear associations. Hoeffding's D varies between -0.5 and 1 (if there are no tied ranks, otherwise it can have lower values), with larger values indicating a stronger relationship between the variables.
- **Somers' D**: The Somers' D statistics is a non-parametric rank based measure of association between a binary variable and a continuous variable, for instance, in the context of logistic regression the binary outcome and the predicted probabilities for each outcome. Usually, Somers' D is a measure of ordinal association, however, this implementation it is limited to the case of a binary outcome.
- **Point-Biserial and biserial correlation**: Correlation coefficient used when one variable is continuous and the other is dichotomous (binary). Point-Biserial is equivalent to a Pearson's correlation, while Biserial should be used when the binary variable is assumed to have an underlying continuity. For example, anxiety level can be measured on a continuous scale, but can be classified dichotomously as high/low.
- **Gamma correlation**: The Goodman-Kruskal gamma statistic is similar to Kendall's Tau coefficient. It is relatively robust to outliers and deals well with data that have many ties.
- **Winsorized correlation**: Correlation of variables that have been formerly Winsorized, i.e., transformed by limiting extreme values to reduce the effect of possibly spurious outliers.
- **Gaussian rank Correlation**: The Gaussian rank correlation estimator is a simple and well-performing alternative for robust rank correlations (Boudt et al., 2012). It is based on the Gaussian quantiles of the ranks.
- **Polychoric correlation**: Correlation between two theorized normally distributed continuous latent variables, from two observed ordinal variables.
- **Tetrachoric correlation**: Special case of the polychoric correlation applicable when both observed variables are dichotomous.

Partial Correlation: **Partial correlations** are estimated as the correlation between two variables after adjusting for the (linear) effect of one or more other variable. The correlation test is then run after having partialized the dataset, independently from it. In other words, it considers partialization as an independent step generating a different dataset, rather than belonging to the same model. This is why some discrepancies are to be expected for the t- and p-values, CIs, BFs etc (but *not* the correlation coefficient) compared to other implementations (e.g., `ppcor`). (The size of these discrepancies depends on the number of covariates partialled-out and the strength of the linear association between all variables.) Such partial correlations can be represented as Gaussian Graphical Models (GGM), an increasingly popular tool in psychology. A GGM traditionally include a set of variables depicted as circles ("nodes"), and a set of lines that visualize relationships between them, which thickness represents the strength of association (see Bhushan et al., 2019).

Multilevel correlations are a special case of partial correlations where the variable to be adjusted for is a factor and is included as a random effect in a mixed model (note that the remaining continuous variables of the dataset will still be included as fixed effects, similarly to regular partial correlations). The model is a random intercept model, i.e. the multilevel correlation is adjusted for `(1 | groupfactor)`.That said, there is an important difference between using `cor_test()` and `correlation()`: If you set `multilevel=TRUE` in `correlation()` but `partial` is set to `FALSE` (as per default), then a back-transformation from partial to non-partial correlation will be attempted (through `pcor_to_cor()`). However, this is not possible when using `cor_test()` so that if you set `multilevel=TRUE` in it, the resulting correlations are partial one. Note that for Bayesian multilevel correlations, if `partial = FALSE`, the back transformation will also recompute *p*-values based on the new *r* scores, and will drop the Bayes factors (as they are not relevant anymore). To keep Bayesian scores, set `partial = TRUE`.

Notes: Kendall and Spearman correlations when `bayesian=TRUE`: These are technically Pearson Bayesian correlations of rank transformed data, rather than pure Bayesian rank correlations (which have different priors).

Examples

```
library(correlation)

cor_test(iris, "Sepal.Length", "Sepal.Width")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "spearman")

cor_test(iris, "Sepal.Length", "Sepal.Width", method = "kendall")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "biweight")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "distance")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "percentage")

if (require("wdm", quietly = TRUE)) {
  cor_test(iris, "Sepal.Length", "Sepal.Width", method = "blomqvist")
}

if (require("Hmisc", quietly = TRUE)) {
  cor_test(iris, "Sepal.Length", "Sepal.Width", method = "hoeffding")
}
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "gamma")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "gaussian")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "shepherd")
if (require("BayesFactor", quietly = TRUE)) {
```

```
  cor_test(iris, "Sepal.Length", "Sepal.Width", bayesian = TRUE)
}

# Robust (these two are equivalent)
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "spearman")
cor_test(iris, "Sepal.Length", "Sepal.Width", method = "pearson", ranktransform = TRUE)

# Winsorized
cor_test(iris, "Sepal.Length", "Sepal.Width", winsorize = 0.2)

# Tetrachoric
if (require("psych", quietly = TRUE) && require("rstanarm", quietly = TRUE)) {
  data <- iris
  data$Sepal.Width_binary <- ifelse(data$Sepal.Width > 3, 1, 0)
  data$Petal.Width_binary <- ifelse(data$Petal.Width > 1.2, 1, 0)
  cor_test(data, "Sepal.Width_binary", "Petal.Width_binary", method = "tetrachoric")

  # Biserial
  cor_test(data, "Sepal.Width", "Petal.Width_binary", method = "biserial")

  # Polychoric
  data$Petal.Width_ordinal <- as.factor(round(data$Petal.Width))
  data$Sepal.Length_ordinal <- as.factor(round(data$Sepal.Length))
  cor_test(data, "Petal.Width_ordinal", "Sepal.Length_ordinal", method = "polychoric")

  # When one variable is continuous, will run 'polyserial' correlation
  cor_test(data, "Sepal.Width", "Sepal.Length_ordinal", method = "polychoric")
}

# Partial
cor_test(iris, "Sepal.Length", "Sepal.Width", partial = TRUE)
if (require("lme4", quietly = TRUE)) {
  cor_test(iris, "Sepal.Length", "Sepal.Width", multilevel = TRUE)
}
if (require("rstanarm", quietly = TRUE)) {
  cor_test(iris, "Sepal.Length", "Sepal.Width", partial_bayesian = TRUE)
}
```

`cor_text` *Correlation text*

Description

This function returns a formatted character of correlation statistics.

Usage

```
cor_text(x, show_ci = TRUE, show_statistic = TRUE, show_sig = TRUE, ...)
```

Arguments

x	A dataframe with correlation statistics.
show_ci, show_statistic, show_sig	Toggle on/off different parts of the text.
...	Other arguments to be passed to or from other functions.

Examples

```
rez <- cor_test(mtcars, "mpg", "wt")

cor_text(rez)
cor_text(rez, show_statistic = FALSE, show_ci = FALSE, stars = TRUE)

rez <- correlation(mtcars)

cor_text(rez)
```

---

cor_to_ci *Convert correlation to p-values and CIs*

Description

Get statistics, *p*-values and confidence intervals (CI) from correlation coefficients.

Usage

```
cor_to_ci(cor, n, ci = 0.95, method = "pearson", correction = "fieller", ...)

cor_to_p(cor, n, method = "pearson")
```

Arguments

cor	A correlation matrix or coefficient.
n	The sample size (number of observations).
ci	Confidence/Credible Interval level. If "default", then it is set to 0.95 (95% CI).
method	A character string indicating which correlation coefficient is to be used for the test. One of "pearson" (default), "kendall", "spearman" (but see also the robust argument), "biserial", "polychoric", "tetrachoric", "biweight", "distance", "percentage" (for percentage bend correlation), "blomqvist" (for Blomqvist's coefficient), "hoeffding" (for Hoeffding's D), "gamma", "gaussian" (for Gaussian Rank correlation) or "shepherd" (for Shepherd's Pi correlation). Setting "auto" will attempt at selecting the most relevant method (polychoric when ordinal factors involved, tetrachoric when dichotomous factors involved, point-biserial if one dichotomous and one continuous and pearson otherwise). See below the **details** section for a description of these indices.

correction	Only used if method is 'spearman' or 'kendall'. Can be 'fieller' (default; Fieller et al., 1957), 'bw' (only for Spearman) or 'none'. Bonett and Wright (2000) claim their correction ('bw') performs better, though the Bishara and Hittner (2017) paper favours the Fieller correction. Both are generally very similar.
...	Additional arguments (e.g., alternative) to be passed to other methods. See stats::cor.test for further details.

Value

A list containing a *p*-value and the statistic or the CI bounds.

References

Bishara, A. J., & Hittner, J. B. (2017). Confidence intervals for correlations when data are not normal. Behavior research methods, 49(1), 294-309.

Examples

```
cor.test(iris$Sepal.Length, iris$Sepal.Width)
cor_to_p(-0.1175698, n = 150)
cor_to_p(cor(iris[1:4]), n = 150)
cor_to_ci(-0.1175698, n = 150)
cor_to_ci(cor(iris[1:4]), n = 150)

cor.test(iris$Sepal.Length, iris$Sepal.Width, method = "spearman", exact = FALSE)
cor_to_p(-0.1667777, n = 150, method = "spearman")
cor_to_ci(-0.1667777, ci = 0.95, n = 150)

cor.test(iris$Sepal.Length, iris$Sepal.Width, method = "kendall", exact = FALSE)
cor_to_p(-0.07699679, n = 150, method = "kendall")
```

---

cor_to_cov *Convert a correlation to covariance*

Description

Convert a correlation to covariance

Usage

```
cor_to_cov(cor, sd = NULL, variance = NULL, tol = .Machine$double.eps^(2/3))
```

Arguments

cor	A correlation matrix, or a partial or a semipartial correlation matrix.
sd, variance	A vector that contains the standard deviations, or the variance, of the variables in the correlation matrix.
tol	Relative tolerance to detect zero singular values.

Value

A covariance matrix.

Examples

```
cor <- cor(iris[1:4])
cov(iris[1:4])

cor_to_cov(cor, sd = sapply(iris[1:4], sd))
cor_to_cov(cor, variance = sapply(iris[1:4], var))
```

---

cor_to_pcor *Correlation Matrix to (Semi) Partial Correlations*

Description

Convert a correlation matrix to a (semi)partial correlation matrix. Partial correlations are a measure of the correlation between two variables that remains after controlling for (i.e., "partialling" out) all the other relationships. They can be used for graphical Gaussian models, as they represent the direct interactions between two variables, conditioned on all remaining variables. This means that the squared partial correlation between a predictor X1 and a response variable Y can be interpreted as the proportion of (unique) variance accounted for by X1 relative to the residual or unexplained variance of Y that cannot be accounted by the other variables.

Usage

```
cor_to_pcor(cor, tol = .Machine$double.eps^(2/3))

pcor_to_cor(pcor, tol = .Machine$double.eps^(2/3))

cor_to_spcor(cor = NULL, cov = NULL, tol = .Machine$double.eps^(2/3))
```

Arguments

cor	A correlation matrix, or a partial or a semipartial correlation matrix.
tol	Relative tolerance to detect zero singular values.
pcor	A correlation matrix, or a partial or a semipartial correlation matrix.
cov	A covariance matrix (or a vector of the SD of the variables). Required for semi-partial correlations.

Details

The semi-partial correlation is similar to the partial correlation statistic. However, it represents (when squared) the proportion of (unique) variance accounted for by the predictor X1, relative to the total variance of Y. Thus, it might be seen as a better indicator of the "practical relevance" of a predictor, because it is scaled to (i.e., relative to) the total variability in the response variable.

Value

The (semi) partial correlation matrix.

Examples

```
cor <- cor(iris[1:4])

# Partialize
cor_to_pcor(cor)
cor_to_spcor(cor, cov = sapply(iris[1:4], sd))

# Inverse
round(pcor_to_cor(cor_to_pcor(cor)) - cor, 2) # Should be 0
```

---

display.easycormatrix *Export tables into different output formats*

Description

Export tables (i.e. data frame) into different output formats. `print_md()` is a alias for `display(format = "markdown")`. Note that you can use `format()` to get the formatted table as a dataframe.

Usage

```
## S3 method for class 'easycormatrix'
display(
  object,
  format = "markdown",
  digits = 2,
  p_digits = 3,
  stars = TRUE,
  include_significance = NULL,
  ...
)

## S3 method for class 'easycorrelation'
print_md(x, digits = NULL, p_digits = NULL, stars = NULL, ...)

## S3 method for class 'easycorrelation'
print_html(x, digits = NULL, p_digits = NULL, stars = NULL, ...)

## S3 method for class 'easycormatrix'
print_md(
  x,
  digits = NULL,
  p_digits = NULL,
  stars = NULL,
```

```
  include_significance = NULL,
  ...
)

## S3 method for class 'easycormatrix'
print_html(
  x,
  digits = NULL,
  p_digits = NULL,
  stars = NULL,
  include_significance = NULL,
  ...
)
```

Arguments

object, x	An object returned by correlation() or its summary.
format	String, indicating the output format. Currently, only "markdown" is supported.
digits, p_digits	To do...
stars	To do...
include_significance	To do...
...	Currently not used.

Details

display() is useful when the table-output from functions, which is usually printed as formatted text-table to console, should be formatted for pretty table-rendering in markdown documents, or if knitted from rmarkdown to PDF or Word files.

Value

A character vector. If format = "markdown", the return value will be a character vector in markdown-table format.

Examples

```
data(iris)
corr <- correlation(iris)
display(corr)

s <- summary(corr)
display(s)
```

is.cor *Check if matrix ressembles a correlation matrix*

Description

Check if matrix ressembles a correlation matrix

Usage

```
is.cor(x)
```

Arguments

x	A matrix.

Value

TRUE of the matrix is a correlation matrix or FALSE otherwise.

isSquare *Check if Square Matrix*

Description

Check if Square Matrix

Usage

```
isSquare(m)
```

Arguments

m	A matrix.

Value

TRUE of the matrix is square or FALSE otherwise.

`matrix_inverse` *Matrix Inversion*

Description

Performs a Moore-Penrose generalized inverse (also called the Pseudoinverse).

Usage

```
matrix_inverse(m, tol = .Machine$double.eps^(2/3))
```

Arguments

`m`	Matrix for which the inverse is required.
`tol`	Relative tolerance to detect zero singular values.

Value

An inversed matrix.

See Also

pinv from the pracma package

Examples

```
m <- cor(iris[1:4])
matrix_inverse(m)
```

`visualisation_recipe.easycor_test`
Visualisation Recipe for 'correlation' Objects

Description

Objects from the `correlation` package can be easily visualized. You can simply run `plot()` on them, which will internally call the `visualisation_recipe()` method to produce a basic `ggplot`. You can customize this plot ad-hoc or via the arguments described below. See examples **here**.

Usage

```
## S3 method for class 'easycor_test'
visualisation_recipe(
  x,
  show_data = "point",
  show_text = "subtitle",
  smooth = NULL,
  point = NULL,
  text = NULL,
  labs = NULL,
  ...
)

## S3 method for class 'easycormatrix'
visualisation_recipe(
  x,
  show_data = "tile",
  show_text = "text",
  show_legend = TRUE,
  tile = NULL,
  point = NULL,
  text = NULL,
  scale = NULL,
  scale_fill = NULL,
  labs = NULL,
  type = show_data,
  ...
)

## S3 method for class 'easycorrelation'
visualisation_recipe(x, ...)
```

Arguments

x	A correlation object.
show_data	Show data. For correlation matrices, can be "tile" (default) or "point".
show_text	Show labels with matrix values.
...	Other arguments passed to other functions.
show_legend	Show legend. Can be set to FALSE to remove the legend.
tile, point, text, scale, scale_fill, smooth, labs	Additional aesthetics and parameters for the geoms (see customization example).
type	Alias for show_data, for backwards compatibility.

Examples

```
rez <- cor_test(mtcars, "mpg", "wt")

layers <- visualisation_recipe(rez, labs = list(x = "Miles per Gallon (mpg)"))
layers
plot(layers)

plot(rez,
  show_text = "label",
  point = list(color = "#f44336"),
  text = list(fontface = "bold"),
  show_statistic = FALSE, show_ci = FALSE, stars = TRUE
)



rez <- correlation(mtcars)

x <- cor_sort(as.matrix(rez))
layers <- visualisation_recipe(x)
layers
plot(layers)

#' Get more details using `summary()`
x <- summary(rez, redundant = TRUE, digits = 3)
plot(visualisation_recipe(x))

# Customize
x <- summary(rez)
layers <- visualisation_recipe(x,
  show_data = "points",
  scale = list(range = c(10, 20)),
  scale_fill = list(
    high = "#FF5722",
    low = "#673AB7",
    name = "r"
  ),
  text = list(color = "white"),
  labs = list(title = "My Plot")
)
plot(layers) + theme_modern()



rez <- correlation(iris)

layers <- visualisation_recipe(rez)
layers
plot(layers)
```

`z_fisher` *Fisher z-transformation*

Description

The Fisher z-transformation converts the standard Pearson's *r* to a normally distributed variable z'. It is used to compute confidence intervals to correlations. The z' variable is different from the *z*-statistic.

Usage

```
z_fisher(r = NULL, z = NULL)
```

Arguments

`r`, `z`	The r or the z' value to be converted.

Value

The transformed value.

References

Zar, J.H., (2014). Spearman Rank Correlation: Overview. Wiley StatsRef: Statistics Reference Online. doi:10.1002/9781118445112.stat05964

Examples

```
z_fisher(r = 0.7)
z_fisher(z = 0.867)
```

Index