

Package ‘wbCorr’

July 23, 2026

Title Bivariate Within- and Between-Cluster Correlations

Version 0.3.2

Description Separates supplied variables into within- and between-cluster components and calculates bivariate correlations for each level separately. For Pearson correlations, the centered-score decomposition corresponds to commonly used between- and within-cluster correlations reviewed by Tu et al. (2025) <[doi:10.1002/sim.10326](https://doi.org/10.1002/sim.10326)>. The package's descriptive Spearman option is distinct from the clustered rank parameters introduced in that paper. The package is also motivated by the distinction between within- and between-person variation described by Curran and Bauer (2011) <[doi:10.1146/annurev.psych.093008.100356](https://doi.org/10.1146/annurev.psych.093008.100356)> and by Hamaker (2024) <[doi:10.1080/00273171.2022.2155930](https://doi.org/10.1080/00273171.2022.2155930)>. The package is intended for longitudinal or otherwise clustered data where researchers need transparent correlation matrices before fitting more complex multilevel models.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Suggests htr, knitr, psych, rmarkdown, spelling, testthat (>= 3.0.0)

Imports methods, writexl

Config/testthat/edition 3

Language en-US

URL <https://pascal-kueng.github.io/wbCorr/>,
<https://github.com/Pascal-Kueng/wbCorr>

BugReports <https://github.com/Pascal-Kueng/wbCorr/issues>

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

NeedsCompilation no

Author Pascal Küng [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-7346-9414>>)

Maintainer Pascal Küng <pascal.kueng@psychologie.uzh.ch>
Repository CRAN
Date/Publication 2026-07-23 18:20:02 UTC

Contents

get_ICC	2
get_matrix	3
get_matrix_diagnostics	5
get_table	6
plot,wbCorr-method	7
print,wbCorr-method	8
show,wbCorr-method	9
simdat_intensive_longitudinal	10
to_excel	11
update_wbCorr	12
wbCorr	12
wbCorr-class	18
Index	19

get_ICC	<i>Return all ICCs for the original variables.</i>
---------	--

Description

You can use get_ICC() or get_ICCs() interchangeably.

Usage

get_ICC(object)

get_ICCs(object)

get_icc(object)

Arguments

object A wbCorr object, created by the wbCorr() function.

Value

A data frame with the one-way random-effects, single-measure ICC(1,1) for every variable. Each ICC is estimated separately from all finite observations with a non-missing cluster identifier. The ANOVA method-of-moments estimator uses an effective cluster size for unbalanced clusters. Negative sample estimates are retained and can be less than -1 in severely unbalanced samples. The

population interpretation assumes independent clusters, a common within-cluster variance, and non-informative cluster size and missingness. NA is returned when an ICC cannot be estimated because there are fewer than two clusters, no within-cluster replication, or zero total variability.

References

- Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2), 420-428. doi:10.1037/00332909.86.2.420
- Ohyama, T. (2025). A comparison of confidence interval methods for the intraclass correlation coefficient based on the one-way random effects model. *Japanese Journal of Statistics and Data Science*, 8, 587-602. doi:10.1007/s42081025002923
- Wang, C.-M., Yandell, B. S., & Rutledge, J. J. (1992). The dilemma of negative analysis of variance estimators of intraclass correlation. *Theoretical and Applied Genetics*, 85, 79-88. doi:10.1007/BF00223848

See Also

[wbCorr](#)

Examples

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# returns the ICCs:
ICCs <- get_ICC(correlations)
print(ICCs)
```

get_matrix

Return matrices for within- and/or between-cluster correlations.

Description

You can use `summary()`, `get_matrices()`, or `get_matrix()` interchangeably. Merged matrices include the ICC on the diagonal. For more detailed statistics, use `get_table()`. By default, matrices are presentation-formatted with two decimal places and significance stars. Set `numeric = TRUE` to retrieve the stored, unrounded numeric coefficients.

Usage

```

get_matrix(
  object,
  which = c("within", "between", "merge"),
  numeric = FALSE,
  ...
)

get_matrices(
  object,
  which = c("within", "between", "merge"),
  numeric = FALSE,
  ...
)

## S4 method for signature 'wbCorr'
summary(object, which = c("within", "between", "merge"), numeric = FALSE, ...)

```

Arguments

<code>object</code>	A <code>wbCorr</code> object, created by the <code>wbCorr()</code> function.
<code>which</code>	A string or a character vector indicating which summaries to return. Options are 'within' or 'w', 'between' or 'b', and various merge options like 'merge', 'm', 'merge_wb', 'wb', 'merge_bw', 'bw'. Default is <code>c('within', 'between', 'merge')</code> .
<code>numeric</code>	A non-missing logical value. If <code>FALSE</code> (the default), return presentation-formatted character matrices with two decimal places and any available significance stars. If <code>TRUE</code> , return unrounded numeric correlation matrices. Numeric merged matrices contain unrounded ICCs on the diagonal.
<code>...</code>	Additional arguments passed to the base summary method

Value

A list containing the selected matrices of within- and/or between-cluster correlations, and ICCs on the diagonals for merged matrices. With `numeric = FALSE`, matrix entries are presentation-formatted character values. With `numeric = TRUE`, matrix columns are numeric and retain the full stored precision.

See Also

[get_tables](#), [wbCorr](#)

Examples

```

# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create object:

```

```
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# returns a correlation matrix with stars for p-values:
matrices <- summary(correlations) # the get_matrix() and get_matrices() functions are equivalent
print(matrices)

# Access specific matrices by:
# Option 1:
matrices$within
# Option 2:
within_matrix <- summary(correlations, which = 'w') # or use 'within'
merged_within_between <- summary(correlations, which = 'wb')
print(within_matrix) # could be saved to an excel or csv file (e.g., write.csv)

# Retrieve unrounded numeric coefficients for downstream calculations:
numeric_matrices <- get_matrix(correlations, numeric = TRUE)
numeric_matrices$within
```

`get_matrix_diagnostics`*Inspect positive-semidefinite diagnostics for correlation matrices*

Description

Returns matrix-level diagnostics computed from the unrounded within- and between-cluster correlation matrices stored in a `wbCorr()` object. Pairwise-complete correlations can be non-positive-semidefinite when different pairs use different rows. A diagnostic is not assessable when the corresponding matrix contains an unavailable (NA) coefficient.

Usage

```
get_matrix_diagnostics(object)
```

Arguments

<code>object</code>	A <code>wbCorr</code> object.
---------------------	-------------------------------

Value

A data frame with one row for each level and columns identifying the diagnostic status, positive-semidefinite result, minimum eigenvalue, completeness, numerical tolerance, number of variables, missing-data mode, whether PSD followed from a common-matrix construction, and reason when assessment was unavailable.

See Also

`wbCorr()`, `get_matrix()`

get_table	<i>Retrieve full tables for both within- and/or between-cluster correlations for a wbCorr object.</i>
-----------	---

Description

This function has an alias `get_tables()` which can be used interchangeably. For correlations matrices, see the `summary()` function.

Usage

```
get_table(object, which = c("within", "between"))

get_tables(object, which = c("within", "between"))
```

Arguments

object	A <code>wbCorr</code> object, created by the <code>wbCorr()</code> function.
which	A character vector indicating which correlation table to return. Options are 'within' or 'w', and 'between' or 'b'.

Value

A list containing the selected detailed tables of within- and/or between-cluster correlations. Each table retains every requested pair and includes raw pair-row, contributing-cluster, bootstrap-yield, coefficient- status, and inference-status diagnostics; see `wbCorr()` for definitions.

See Also

[summary](#), [wbCorr](#)

Examples

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# returns a list with full detailed tables of the correlations:
tables <- get_table(correlations) # the get_tables() function is equivalent
print(tables)

# Access specific tables by:
# Option 1:
tables$between
# Option 2:
within_table <- get_tables(correlations, which = 'w') # or use 'within' or 'between'
```

```
print(within_table) # within_table could be saved to an excel or csv file (e.g., write.csv)
```

plot,wbCorr-method *Plot within- and between associations*

Description

Plots the centered variables of the provided data frame against each other. Choose either cluster means ("between") or deviations from cluster means ("within"). Every panel uses the same pair-specific rows, centering rule, method, and between-cluster weights as the fitted object. Pearson plots draw a corresponding regression line and annotate the stored correlation; weighted between-cluster panels use weighted least squares. Spearman plots report the stored rho without a linear-regression overlay. Significance stars are shown only when the fitted wbCorr object contains a p-value for that pair.

Usage

```
## S4 method for signature 'wbCorr'
plot(
  x,
  y,
  which = NULL,
  plot_NA = TRUE,
  standardize = TRUE,
  outlier_detection = "zscore",
  outlier_threshold = "recommended",
  type = "p",
  pch = 20,
  dot_lwd = 2,
  reg_lwd = 2,
  ...
)
```

Arguments

x	A wbCorr object to be plotted.
y	Choose which correlations to plot ('within' / 'w' or 'between' / 'b'); can be used as a positional argument.
which	Can be used as an alternative to 'y' (e.g., which = 'w'). It has the same functionality as 'y', but takes precedence if both are specified.
plot_NA	Boolean. Whether variables that have no variation on the selected level should be plotted or not.
standardize	Boolean. Whether each plotted pair should be standardized using the same weights as its fitted coefficient. For Pearson panels, this makes the displayed regression slope equal to the stored correlation.

outlier_detection
 If FALSE, outliers will not be marked in red. Otherwise you may provide the method. Choose from: 'zscore', 'mad', or 'tukey'.

outlier_threshold
 If 'recommended', the threshold for 'zscore' and 'mad' will be set to 3, and for 'tukey' to 1.5. You can provide and other numeric here.

type
 points, lines, etc. see ?base::plot for available types).

pch
 Graphical parameter. Select which type of points should be plotted.

dot_lwd
 Graphical parameter. Set size of the points.

reg_lwd
 Graphical parameter. Set thickness of the regression line.

...
 further options to be passed to the base plot (pairs) function.

Value

Invisibly returns the supplied wbCorr object. Called for the side effect of drawing a pairs plot of the selected within- or between-cluster centered variables.

See Also

[wbCorr](#)

print,wbCorr-method *Print Method for the wbCorr Class*

Description

Prints a summary of the wbCorr object.

Usage

```
## S4 method for signature 'wbCorr'
print(x, ...)
```

Arguments

x
 A wbCorr object.

...
 Additional arguments, currently unused.

Value

Invisibly returns the supplied wbCorr object. Called for the side effect of printing a compact summary of the within-cluster table, between-cluster table, and ICC table.

See Also

[wbCorr](#)

Examples

```
# Example
data("simdat_intensive_longitudinal")
correlations <- wbCorr(simdat_intensive_longitudinal,
                      cluster = 'participantID',
                      confidence_level = 0.95,
                      method = 'spearman',
                      weighted_between_statistics = FALSE)

print(correlations)
```

show,wbCorr-method	<i>Show Method for the wbCorr Class</i>
--------------------	---

Description

Shows a summary of the wbCorr object, equivalent to the print method.

Usage

```
## S4 method for signature 'wbCorr'
show(object)
```

Arguments

object A wbCorr object.

Value

Invisibly returns the supplied wbCorr object. Called for the side effect of showing the same compact summary as print().

See Also

[wbCorr](#), [print.wbCorr](#)

Examples

```
# Example using the iris dataset
cors <- wbCorr(iris, iris$Species, weighted_between_statistics = TRUE)
show(cors)
```

`simdat_intensive_longitudinal`*Simulated Intensive Longitudinal Dataset*

Description

A simulated intensive longitudinal dataset to test the package capabilities. This dataset contains 5,000 observations from 100 participants measured on 50 days, plus three variables (var1, var2, and var3) that are correlated at both the within- and between-person levels.

Format

A data frame with 5,000 rows and the following five columns:

participantID Identifier for each participant (integer)
day Day variable varying only within-person (integer)
var1 Variable 1 (numerical)
var2 Variable 2 (numerical)
var3 Variable 3 (numerical)

Details

The within-person correlations are all positive:

- var1 & var2: 0.1
- var1 & var3: 0.3
- var2 & var3: 0.8

The between-person correlations are all negative:

- var1 & var2: -0.5
- var1 & var3: -0.4
- var2 & var3: -0.2

Time trends (within):

- var1 & time: 0.0
- var2 & time: 0.0
- var3 & time: 0.4

Source

A simulated dataset by P. K  ng

to_excel	<i>Save tables or matrices to Excel</i>
----------	---

Description

Use `to_excel(get_matrix(wbCorrObject))` or `to_excel(get_table(wbCorrObject))` to save `wbCorr` output. A single data frame or matrix can also be passed directly. Lists may contain any mixture of data frames and matrices; other list elements, such as explanatory notes returned by `get_matrix()`, are ignored.

Usage

```
to_excel(SummaryObject, path = file.path(getwd(), "wbCorr.xlsx"))
```

Arguments

<code>SummaryObject</code>	A data frame, matrix, or list containing data frames and/or matrices, including objects returned by <code>get_matrix()</code> or <code>get_table()</code> . Row names are written to a <code>RowName</code> column.
<code>path</code>	A single non-missing file path. If omitted, <code>wbCorr.xlsx</code> is written to the current working directory.

Value

The output path, invisibly. The function writes an Excel workbook to disk and errors if no data frame or matrix was supplied.

See Also

`get_tables`, `wbCorr`, `get_matrix`

Examples

```
# Importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# Create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# Returns a correlation matrix with stars for p-values:
matrices <- get_matrix(correlations) # summary(correlations) works too.

to_excel(matrices, path = tempfile(fileext = ".xlsx"))

# A data frame or matrix can be exported directly:
to_excel(data.frame(x = 1:3), path = tempfile(fileext = ".xlsx"))
to_excel(matrix(1:4, nrow = 2), path = tempfile(fileext = ".xlsx"))
```

update_wbCorr	<i>Check for updates of wbCorr</i>
---------------	------------------------------------

Description

This function checks if there is a newer version on GitHub by comparing the version numbers in the local and remote DESCRIPTION files. It only runs when called explicitly by the user and does not install updates.

Usage

```
update_wbCorr(ask = FALSE)
```

Arguments

ask	Deprecated and ignored.
-----	-------------------------

Value

An integer: 1 if there's a newer version available, 0 if the current version is the latest, or NULL if there was an error accessing the remote DESCRIPTION file.

wbCorr	<i>wbCorr</i>
--------	---------------

Description

The wbCorr function creates a wbCorr object containing within- and between-cluster correlations, p-values, and confidence intervals for a given dataset and clustering variable. The object can be plotted.

Usage

```
wbCorr(
  data,
  cluster,
  confidence_level = 0.95,
  method = "pearson",
  bootstrap = FALSE,
  nboot = 1000,
  inference = c("analytic", "none", "cluster_bootstrap"),
  weighted_between_statistics = NULL,
  between_weighting = c("equal_clusters", "cluster_size"),
  between_inference = c("analytic", "none"),
  centering_rows = c("pairwise_complete", "all_available"),
  missing_data = c("pairwise", "listwise")
)
```

```

)

wbcorr(
  data,
  cluster,
  confidence_level = 0.95,
  method = "pearson",
  bootstrap = FALSE,
  nboot = 1000,
  inference = c("analytic", "none", "cluster_bootstrap"),
  weighted_between_statistics = NULL,
  between_weighting = c("equal_clusters", "cluster_size"),
  between_inference = c("analytic", "none"),
  centering_rows = c("pairwise_complete", "all_available"),
  missing_data = c("pairwise", "listwise")
)

```

Arguments

<code>data</code>	A data frame containing numeric variables, logical variables, or two-level factors for which correlations will be calculated.
<code>cluster</code>	An atomic vector with exactly one value per data row, or one string naming the cluster column in data. Missing identifiers are allowed, but at least one identifier must be observed; numeric identifiers cannot be Inf, -Inf, or NaN. When a vector is supplied, data columns containing the same identifiers and missing-value pattern are treated as duplicate cluster columns and excluded; pass a column name when the identifier is already in data to avoid ambiguity.
<code>confidence_level</code>	A numeric value between 0 and 1 representing the desired level of confidence for confidence intervals (default: 0.95).
<code>method</code>	A string indicating the correlation method to be used. Supported methods are "pearson", "spearman", and "auto" (default: "pearson"). Pearson uses t statistics and Fisher-z confidence intervals. Spearman reports a descriptive correlation of mean-centered scores within clusters and a correlation of cluster means between clusters. Analytic Spearman inference is unavailable; use <code>inference = "cluster_bootstrap"</code> for a whole-cluster bootstrap interval. The former "spearman-jackknife" option is rejected because deleting individual rows does not respect clustering.
<code>bootstrap</code>	Deprecated logical alias for <code>inference = "cluster_bootstrap"</code> .
<code>nboot</code>	A whole number of bootstrap samples, at least 10 (default: 1000). The minimum permits quick tests; use substantially more replicates for substantive analyses and assess Monte Carlo stability.
<code>inference</code>	A string specifying inferential output. "analytic" uses the documented correlation-test approximation for Pearson p-values and confidence intervals. "none" returns coefficients only. "cluster_bootstrap" resamples top-level clusters with replacement, recomputes the full decomposition, and reports percentile confidence intervals. Bootstrap p-values are not reported.

<code>weighted_between_statistics</code>	Deprecated logical alias for <code>between_weighting</code> . If TRUE, <code>between_weighting = "cluster_size"</code> ; if FALSE, <code>between_weighting = "equal_clusters"</code> .
<code>between_weighting</code>	A string specifying the between-cluster estimand. <code>"equal_clusters"</code> correlates pair-specific cluster means with each cluster contributing equally. <code>"cluster_size"</code> computes a sample-size weighted correlation of pair-specific cluster means, using the number of complete observation pairs in each cluster as weights. Cluster-size weighting is not supported with <code>method = "spearman"</code> because no weighted-rank estimand is currently defined.
<code>between_inference</code>	A string specifying whether between-cluster p-values and confidence intervals are calculated analytically (<code>"analytic"</code>) or omitted (<code>"none"</code>). Analytic inference is unavailable for <code>between_weighting = "cluster_size"</code> ; <code>wbCorr</code> preserves the weighted coefficient but omits its p-value and confidence interval. Use <code>inference = "cluster_bootstrap"</code> for weighted between-cluster inference. Ignored when <code>inference = "none"</code> or <code>inference = "cluster_bootstrap"</code> .
<code>centering_rows</code>	A string specifying which rows are used to estimate cluster means for within- and between-cluster decomposition. <code>"pairwise_complete"</code> uses only rows where both variables in the current pair are observed. <code>"all_available"</code> estimates each variable's cluster mean from all available rows for that variable, then correlates the pair on complete rows.
<code>missing_data</code>	A string specifying whether correlations use all available pairs (<code>"pairwise"</code> , the default) or first retain only rows that are complete across every supported analysis variable and the cluster identifier (<code>"listwise"</code>). Listwise deletion provides a common multivariate sample for users who require a coherent correlation matrix, at the cost of discarding partially observed rows.

Details

Calculates bivariate within- and between-cluster correlations for clustered data, such as repeated measures nested in persons, dyads, teams, or other groups. Only recommended for continuous or binary variables.

Logical variables are encoded as 0/1. Factors must declare exactly two levels and are encoded as 0/1 in declared factor-level order, so reversing the levels reverses correlations with other variables. Character variables must first be converted to factors with an explicit two-level order. Other factors are not accepted; use meaningful numeric scores for ordered variables or dummy-code nominal variables. Numeric Inf, -Inf, and NaN values are treated as missing before centering and estimation.

By default, `missing_data = "pairwise"`, and every variable-pair correlation is computed on rows where both variables and the cluster variable are observed. Because different pairs can then use different rows, a completed pairwise correlation matrix need not be positive semidefinite. `wbCorr` checks both unrounded level-specific matrices, warns when a matrix is not positive semidefinite, and exposes the result through `get_matrix_diagnostics()`.

With `missing_data = "listwise"`, rows missing any supported analysis variable or the cluster identifier are removed before correlation decomposition and inference. This gives all pairs a common row-row sample. ICCs keep their documented variable-wise finite-row samples and are not

changed by this matrix-oriented option. With a single coefficient method ("pearson" or "spearman") and common weights, the resulting complete correlation matrix is positive semidefinite up to numerical tolerance. `method = "auto"` can mix Pearson and Spearman entries, so listwise deletion alone cannot guarantee that its combined matrix is positive semidefinite; diagnostics are still reported.

Under pairwise handling, `centering_rows = "pairwise_complete"` also estimates cluster means from this same complete-pair row set. This keeps the within residuals centered for the actual pairwise sample and makes the between correlation a correlation of matched pair-specific cluster means.

Detailed tables always retain one row for every requested unordered pair. `n_obs` is the number of jointly observed raw rows with a nonmissing cluster identifier, and `n_clusters` is the number of clusters contributing at least one such row. Under `centering_rows = "all_available"`, additional unpaired rows can contribute to the variable-specific means, but `n_obs` remains the joint pair-row count. `status` describes coefficient estimability ("ok" or "not_estimable") and `reason` gives a stable failure code. `inference_status` separately records "not_requested", "ok", "partial", or "unavailable", with details in `inference_reason`. A descriptive coefficient requires two analysis units with positive variance; inferential output can require more.

With `centering_rows = "all_available"`, each variable's cluster mean is estimated from all available rows for that variable before the pairwise correlation is computed. This can make the cluster means more stable when data are missing. It also mirrors a common multilevel-model preprocessing workflow, where person means are often created before the model applies complete-case filtering. That workflow is defensible in multilevel models. In `wbCorr`, however, the variables are treated symmetrically as a descriptive bivariate decomposition, so all-available centering means the two cluster means in a pair may be based on different occasions. For that reason, "pairwise_complete" is the default.

The within-cluster correlation is the pooled residual correlation. For a given pair, each observed value is centered around its cluster mean for that same complete-pair row set, and the correlation is computed on the resulting residuals. For Pearson within-cluster correlations, analytic inference uses $N_{\text{pair}} - k_{\text{pair}} - 1$ degrees of freedom, where N_{pair} is the number of complete observation pairs and k_{pair} is the number of clusters contributing at least one complete pair. This analytic test is a working approximation because residual pairs can still be dependent within clusters. For resampling intervals that preserve the top-level dependence, use `inference = "cluster_bootstrap"`.

The between-cluster correlation is computed from pair-specific cluster means. With `between_weighting = "equal_clusters"`, every cluster contributes one equally weighted mean. With `between_weighting = "cluster_size"`, cluster means are weighted by the number of complete observation pairs in that cluster. The ordinary Pearson *t* test and Fisher-*z* interval are not valid for a weighted correlation. Therefore analytic *p*-values and confidence intervals are omitted for cluster-size-weighted between correlations. Use `inference = "cluster_bootstrap"` when inference is required.

Pearson analytic confidence intervals use Fisher's *z* transformation. If *df* denotes the corresponding *t*-test degrees of freedom, the Fisher-*z* standard error is $1 / \sqrt{df - 1}$. The interval is unavailable when *df* ≤ 1.

With `method = "spearman"`, the within coefficient is Spearman's correlation of the pairwise mean-centered scores and the between coefficient is Spearman's correlation of the pairwise cluster means. These are descriptive mean-based decompositions, not the conditional-ridit and median-centroid clustered rank parameters of Tu, Li, and Shepherd (2025), and need not be invariant to monotone transformations of the original observations. `wbCorr` therefore does not attach analytic *p*-values or confidence intervals to these coefficients. Whole-cluster bootstrap confidence intervals are available.

For each variable, wbCorr also reports the one-way random-effects, single-measure ICC(1,1). It is estimated from all finite observations for that variable using the ANOVA method of moments and an effective cluster size when clusters are unbalanced. Negative sample ICCs are retained; they occur when the between-cluster mean square is smaller than the within-cluster mean square and, for severely unbalanced samples, the raw ANOVA estimate can be less than -1. Its population interpretation assumes the one-way random-effects model: independent clusters, a common within-cluster variance, and noninformative cluster size and missingness. The ICC is NA when the data do not contain enough clusters or within-cluster replication, or when total variability is zero.

With `inference = "cluster_bootstrap"`, wbCorr resamples whole top-level clusters, recomputes the selected within- and between-cluster correlations, and reports first-order percentile bootstrap confidence intervals. This keeps the package's descriptive estimands while avoiding row-level independence assumptions. Interval accuracy assumes independent clusters and adequate numbers of clusters and bootstrap replicates; the technical minimum of 10 valid replicates is not a recommendation for substantive analyses. `n_boot_attempted` and `n_boot_valid` report Monte Carlo yield. Bootstrap is skipped when fewer than three clusters contribute. An interval is unavailable with fewer than 10 finite replicate coefficients, and is marked "partial" when invalid replicates had to be excluded. No bootstrap p-value is reported because wbCorr does not currently implement a validated null-resampling test for these clustered estimands.

Correlation-matrix diagonals are 1 only when the variable has at least two usable values and positive variance at that level; otherwise they are NA. P-value diagonals are always NA. Merged summary matrices continue to show the variable's ICC on the diagonal instead of a level-specific self-correlation. A matrix with any unavailable coefficient cannot be assessed as a complete positive-semidefinite correlation matrix and is reported as "not_assessable" rather than silently treated as valid.

Inspired by the `psych::statsBy` function, wbCorr allows you to calculate, extract, and plot within- and between-cluster correlations for further analysis.

Value

A wbCorr object that contains within- and between-cluster statistics. Use the `get_table()` function on the wbCorr object to retrieve a list of the full correlation tables. Use the `summary()` or `get_matrix()` function on the wbCorr object to retrieve various correlation matrices, including ICCs in the merged ones. Use `get_ICC()` to retrieve all intraclass correlations (ICC(1,1)). Finally, use `to_excel()` on a table or matrix (or list of matrices) to save them.

References

- Tu, S., Li, C., & Shepherd, B. E. (2025). Between- and within-cluster Spearman rank correlations. *Statistics in Medicine*. doi:10.1002/sim.10326
- Hall, P., & Wilson, S. R. (1991). Two guidelines for bootstrap hypothesis testing. *Biometrics*, 47(2), 757-762. doi:10.2307/2532163
- Martin, M. A. (2007). Bootstrap hypothesis testing for some common statistical problems: A critical evaluation of size and power properties. *Computational Statistics & Data Analysis*, 51(12), 6321-6342. doi:10.1016/j.csda.2007.01.020
- Andrews, D. W. K., & Buchinsky, M. (2000). A three-step method for choosing the number of bootstrap repetitions. *Econometrica*, 68(1), 23-51. doi:10.1111/14680262.00092

R Core Team. Correlation, variance and covariance matrices. R statistical software documentation.
<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/cor.html>

See Also

[get_table](#), [summary](#), [get_matrix_diagnostics](#), [get_ICC](#), [plot](#), [to_excel](#)

Examples

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create a wbCorr object:
correlations <- wbCorr(simdat_intensive_longitudinal,
                      'participantID')

# optionally compute sample-size weighted between-cluster correlations:
weighted_correlations <- wbCorr(simdat_intensive_longitudinal,
                              'participantID',
                              between_weighting = 'cluster_size')

# quick cluster-bootstrap example; use more bootstrap samples in applied work:

bootstrapped_correlations <- wbCorr(simdat_intensive_longitudinal,
                                   'participantID',
                                   inference = 'cluster_bootstrap',
                                   nboot = 20)

# optionally estimate cluster means from all rows available for each variable:
all_available_correlations <- wbCorr(simdat_intensive_longitudinal,
                                    'participantID',
                                    centering_rows = 'all_available')

# returns a list with full detailed tables of the correlations:
tables <- get_table(correlations) # the get_tables() function is equivalent
print(tables)

# returns a correlation matrix with stars for p-values:
matrices <- summary(correlations) # the get_matrix() and get_matrices() functions are equivalent
print(matrices)

# Plot the centered variables against each other
plot(correlations, 'within')
plot(correlations, which = 'b')

# Store the list of correlation matrices to excel
to_excel(matrices, path = tempfile(fileext = ".xlsx"))
```

`wbCorr-class`*wbCorr Class*

Description

A class representing within- and between-cluster correlations.

Details

The `wbCorr` class is used to store within- and between-cluster correlations and provides methods for printing and summarizing the correlations.

See Also

[wbCorr](#)

Index

`get_ICC`, [2](#), [17](#)
`get_icc (get_ICC)`, [2](#)
`get_ICCs (get_ICC)`, [2](#)
`get_matrices (get_matrix)`, [3](#)
`get_matrix`, [3](#), [11](#)
`get_matrix()`, [5](#), [11](#)
`get_matrix_diagnostics`, [5](#), [17](#)
`get_matrix_diagnostics()`, [14](#)
`get_table`, [6](#), [17](#)
`get_table()`, [11](#)
`get_tables`, [4](#), [11](#)
`get_tables (get_table)`, [6](#)

`plot`, [17](#)
`plot,wbCorr-method`, [7](#)
`plot.wbCorr (plot,wbCorr-method)`, [7](#)
`print,wbCorr-method`, [8](#)
`print.wbCorr`, [9](#)
`print.wbCorr (print,wbCorr-method)`, [8](#)

`show,wbCorr-method`, [9](#)
`show.wbCorr (show,wbCorr-method)`, [9](#)
`simdat_intensive_longitudinal`, [10](#)
`summary`, [6](#), [17](#)
`summary,wbCorr-method (get_matrix)`, [3](#)
`summary.wbCorr (get_matrix)`, [3](#)

`to_excel`, [11](#), [17](#)

`update_wbCorr`, [12](#)

`wbCorr`, [3](#), [4](#), [6](#), [8](#), [9](#), [11](#), [12](#), [18](#)
`wbcorr (wbCorr)`, [12](#)
`wbCorr()`, [5](#), [6](#)
`wbCorr-class`, [18](#)