

Package ‘qapproach’

October 8, 2026

Type Package

Title The Q Approach to Consensus Building

Version 0.1.2

Description Implements a workflow based on Q method to support consensus-building processes. It prepares participant rankings, selects and fits group perspectives, calculates consensus priority scores, validates results by bootstrap resampling, and produces publication-ready figures. The underlying method is described by Geschke et al. (2022) ``The Q approach to consensus building: integrating diverse perspectives to guide decision-making" <doi:10.32942/X2F59S>.

License GPL-3

URL <https://doi.org/10.5281/zenodo.11518485>

BugReports <https://github.com/JonasGeschke/qapproach/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports fmsb, igraph, qmethod, withr

Suggests hues, knitr, magick, pdftools, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Jonas Geschke [aut, cre]

Maintainer Jonas Geschke <hallo@qapproach.app>

Repository CRAN

Date/Publication 2026-10-08 10:00:02 UTC

Contents

agreement_across_levels	2
consensus_across_levels	3

cpscores	4
distributiondetermination	4
plot_hierarchical_levels	5
plot_network_two_layered	8
plot_sdg_cps	9
plot_sdg_diamonds	10
qapproach	11
qapproach-bootstrap	13
qapproach-plots	14
summary	16
validate	17
validation_cps	18
validation_means	19
validation_perspectives	21
Index	22

agreement_across_levels
<i>Trace ranking agreement across successive Q approach levels</i>

Description

****Experimental.**** This function traces each original input ranking through an arbitrary number of analytical levels. Original rankings may enter at any supplied level; inputs that match a generated group-perspective identifier are excluded from the set of original rankings. Its interface and interpretation may be refined as additional multi-level applications become available.

Usage

```
agreement_across_levels(levels, print_table = TRUE)
```

Arguments

- levels A named list of levels. Each element is a Q approach result or a list of Q approach results representing the analyses at that level.
- print_table Logical; print the agreement-lineage table for convenience.

Details

For every supplied level, the returned table reports the analysis in which the current input was found, whether it agreed with, opposed, or was undecided about a perspective, and the relevant perspective identifier. ‘Agreement path’ joins only positively agreeing perspectives with ‘>’. Opposition and undecided status do not propagate agreement through a group perspective. If a raw ranking is explicitly re-added at a later level, it can enter the agreement path there. Rankings that do not agree with a perspective at any level receive ‘NA’ as their agreement path.

Value

A list containing ‘paths’, with one row per original input ranking; ‘underlying_agreement_counts’, with the number of unique original rankings whose positive path reaches each perspective; and ‘terminal_path_summary’, with counts and shares by terminal perspective. ‘rankings_not_agreeing’ reports the count, identifiers, and underlying statement-ranking values of original rankings that do not positively agree with any perspective at any level. An internal completeness check warns if the original rankings are not represented exactly once in ‘paths’.

consensus_across_levels	<i>Summarize consensus across successive Q approach levels</i>
-------------------------	---

Description

Experimental. This function is provided for testing and further refinement. Its interface and calculations may change, and it should not yet be used for definitive analytical conclusions.

The transition table reports direct agreement at every level and traces the original individual rankings through successive positive-agreement links.

Usage

```
consensus_across_levels(levels, print_table = TRUE)
```

Arguments

- levels A named list of levels. Each element is a Q approach result or a list of Q approach results representing the datasets at that level.
- print_table Logical; print a compact transition table when TRUE.

Value

A list containing the number of raw input rankings, a transition table, and convenient final-transition values for underlying pool agreement, pool counts, underlying individual agreement, and underlying individual counts by perspective. The transition table stores the pool counts and perspective-level counts as structured list-columns.

cpscores

Calculate consensus priority scores

Description

Calculates eigenvalue-weighted consensus priority scores from perspective z-scores and eigenvalues on a fixed standard-normal cumulative-probability scale.

Usage

```
cpscores(statementzscores, factoreigenvalues)
```

Arguments

statementzscores

A matrix of perspective z-scores, with statements in columns or rows.

factoreigenvalues

Numeric eigenvalues for the perspectives.

Details

The function first calculates the eigenvalue-weighted mean z-score for every statement and then applies the fixed standard-normal cumulative distribution function. A score of 0.5 represents neutral prioritization across the group perspectives; values above or below 0.5 represent relatively higher or lower priority. Comparisons across analyses require the same statements and meanings, ranking distribution, instructions, data preparation, and analytical settings.

Value

A named numeric vector of consensus priority scores between the theoretical boundaries 0 and 1.

distributiondetermination

Determine the required Q-sort ranking distribution

Description

Returns the fixed ranking distribution used by the Q approach for a given number of statements. This distribution provides the basis for preparing data-collection materials and validating completed rankings.

Usage

```
distributiondetermination(nstat)
```

Arguments

nstat Number of statements. Must be one whole number of at least 3.

Value

A list containing the number of statements per ranking value (distr), the ranking values (values), and the complete ranking gradient (ranking).

plot_hierarchical_levels

Plot a hierarchical network across Q approach levels

Description

Draws a bottom-up network in which the original individual rankings feed into their analyses and the resulting group perspectives feed into analyses at subsequent levels. Each group perspective is retained as a separate edge, including when several perspectives connect the same two analyses. Individual rankings feeding into the same earliest target analysis are positioned next to one another; their order within each target group retains the crossing-minimizing Sugiyama order.

Usage

```
plot_hierarchical_levels(
  levels,
  file = NULL,
  level_colors = NULL,
  individual_arrow_color = "grey",
  perspective_arrow_color = "black",
  level_node_sizes = NULL,
  analysis_labels = NULL,
  ranking_labelled = FALSE,
  analysis_labelled = TRUE,
  level_gaps = NULL,
  ranking_node_size = 1,
  arrow_size = 0.25,
  width = 12,
  height = 10,
  agreement = FALSE,
  network_object = FALSE
)
```

Arguments

levels A named list of levels. Each element is a qapproach() result or a list of qapproach() results representing the analyses at that level.

file Optional output graphics path. When 'NULL', the plot is drawn on the active graphics device without writing a file.

<code>level_colors</code>	Optional vector containing one node color per level. The default uses distinguishable colors generated by <code>'hues::iwanthue()'</code> .
<code>individual_arrow_color</code>	Color of arrows from individual rankings when <code>'agreement = FALSE'</code> .
<code>perspective_arrow_color</code>	Color of arrows representing group perspectives when <code>'agreement = FALSE'</code> .
<code>level_node_sizes</code>	Optional sizes for the analysis nodes. <code>'NULL'</code> uses 0.3 times the number of underlying individual rankings agreeing through the analysis's group perspectives. A single value is used for all analysis nodes; a vector may provide one value per level or per analysis. A named vector may use the displayed analysis labels.
<code>analysis_labels</code>	Optional displayed labels for the analysis nodes. A named vector may be matched to the internal analysis names; otherwise one label must be supplied per analysis in level order.
<code>ranking_labelled</code>	Logical; show labels for the individual-ranking nodes. The default is <code>'FALSE'</code> and does not affect analysis-node labels.
<code>analysis_labelled</code>	Logical; show labels for the analysis nodes. The default is <code>'TRUE'</code> . Set both <code>'ranking_labelled = FALSE'</code> and <code>'analysis_labelled = FALSE'</code> to draw the network without node labels.
<code>level_gaps</code>	Optional named numeric vector adding vertical space after selected levels. Names must be <code>"rankings"</code> or one of the supplied level names except the final level. Values are multiples of the standard distance between adjacent levels. For example, <code>'c(rankings = 0.5, level2 = 1)'</code> adds half a standard distance between the ranking foundation and level 1, and one standard distance between level 2 and level 3. Values must be finite and non-negative. <code>'NULL'</code> preserves the standard vertical spacing.
<code>ranking_node_size</code>	Size of individual-ranking nodes.
<code>arrow_size</code>	Arrow-size multiplier.
<code>width, height</code>	Figure dimensions in inches when <code>'file'</code> is supplied.
<code>agreement</code>	Logical; distinguish agreement, opposition, and undecided inputs through edge formatting when plotting. This argument affects only the drawn styling.
<code>network_object</code>	Logical; return the prepared <code>'igraph'</code> network object without plotting or writing a file. The returned object contains only the network metadata prepared before plotting and does not contain plot styling selected through <code>'agreement'</code> or any other graphical settings. It retains the vertex attributes <code>'name'</code> , <code>'type'</code> , <code>'level'</code> , <code>'label'</code> , and <code>'color'</code> ; the edge attributes <code>'input_id'</code> , <code>'input_type'</code> , and <code>'status'</code> ; and the graph attribute <code>'level_names'</code> . The <code>'color'</code> vertex attribute is the unmodified level color before transparency is applied. The <code>'input_type'</code> and <code>'status'</code> edge attributes allow the agreement styling to be reconstructed. For analysis vertices, <code>'name'</code> contains the technical analysis name supplied through <code>'levels'</code> (for example, <code>'alps'</code> or <code>'nandes'</code>), whereas <code>'label'</code> contains the displayed analysis label.

Details

When `'network_object = TRUE'`, the returned `'igraph'` object contains analytical metadata, but not the graphical settings calculated during plotting.

Attributes included in the returned network

- The standard vertex attribute `'name'`, plus vertex attributes `'type'`, `'level'`, `'label'`, and `'color'`. For analysis vertices, `'name'` contains the technical analysis name supplied through `'levels'`; the `'color'` attribute contains the unmodified level color, before plotting transparency is applied;
- Edge attributes `'input_id'`, `'input_type'`, and `'status'`;
- The graph attribute `'level_names'`.

Graphical settings not stored in the returned network

The object does not contain the generated layout, vertex sizes, vertex frame color, label size or other label formatting, edge colors, edge line types or widths, arrow settings, edge curvature, plot limits, or other graphical parameters. These are calculated only when `'network_object = FALSE'`.

In the standard plot, individual-ranking nodes use `'ranking_node_size'` (default 1). Unless `'level_node_sizes'` is supplied, each analysis node is sized as 0.3 times the sum of the underlying individual rankings agreeing through its group perspectives. Analysis-node colors use the stored `'color'` attribute with 80 percent opacity; ranking nodes remain opaque. Frames are omitted. Ranking and analysis label sizes are 0.5 and 0.95, respectively; analysis labels are bold, and labels use color `'#1F1F1F'` and the sans-serif font family.

With `'agreement = FALSE'`, individual-ranking edges use `'individual_arrow_color'` and group-perspective edges use `'perspective_arrow_color'`; these defaults are light grey and black, respectively. These are edge colors, not line types, and both use solid lines. With `'agreement = TRUE'`, the input-type colors are replaced: agreeing edges are grey, opposing edges are firebrick, and undecided edges are dodger blue (`'dodgerblue3'`). All three statuses use solid lines. Individual-ranking and group-perspective edges both use width 0.5. Arrows use `'arrow_size'`, and parallel edges are curved with `'igraph::curve_multiple()'`.

The `'agreement'` argument changes only the drawn styling. The returned network object is the same in either mode: its `'input_type'` edge attribute supports the individual-versus-perspective mapping, and its `'status'` edge attribute supports the agreeing-versus-opposing-versus-undecided mapping.

Vertical gaps are applied after the deterministic Sugiyama ordering and equal-gap horizontal placement have been calculated. Each value in `'level_gaps'` shifts every subsequent analytical level upward without changing node order or horizontal position. Thus, `'level2 = 0.5'` adds half a standard level distance only between levels 2 and 3. The special name `'rankings'` controls the gap between the individual-ranking foundation and the first analytical level. Multiple named gaps are cumulative.

The retained attributes allow specialists to reproduce these mappings or construct entirely custom layouts and plots directly with `'igraph'`.

Value

When `'network_object = TRUE'`, returns the prepared `'igraph'` object. Otherwise, invisibly returns the normalized output path when a file is written and invisibly returns `'NULL'` when drawing on the active device.

plot_network_two_layered

Plot and save a two-layered multi-dataset synthesis network

Description

Plot and save a two-layered multi-dataset synthesis network

Usage

```
plot_network_two_layered(
  file = NULL,
  datasets,
  synthesis,
  labels = NULL,
  statement_colors = NULL,
  network_labelled = FALSE,
  perspective_label_cex = 2,
  ranking_label_cex = 1.1,
  dataset_colors = NULL,
  synthesis_color = NULL,
  cps_color = NULL,
  preset = NULL,
  width = 11.2,
  height = 6.4,
  input_network = FALSE,
  arrow_size = 0.56,
  negative_loadings_marked = TRUE,
  layout_seed = NULL
)
```

Arguments

file	Optional output graphics path. When 'NULL', the plot is drawn on the active graphics device without writing a file.
datasets	Named list of qapproach() result objects, one per dataset.
synthesis	qapproach() result for the combined synthesis dataset. The synthesis input may contain dataset perspectives and individual rankings returned by 'not_agreeing()'.
labels	Optional statement labels.
statement_colors	Optional statement colors.
network_labelled	Whether to label individual ranking nodes.
perspective_label_cex, ranking_label_cex	Label-size multipliers for perspective and ranking nodes.

dataset_colors	Optional colors for the dataset-perspective nodes.
synthesis_color	Optional color for synthesis-perspective nodes.
cps_color	Optional color for the cp-score node.
preset	Optional visualization preset.
width, height	Figure dimensions in inches.
input_network	Whether to show the flow of analytical inputs instead of agreement and opposition results.
arrow_size	Arrow-size multiplier.
negative_loadings_marked	Whether negative flagged loadings are marked.
layout_seed	Integer seed for reproducible network-node placement, or 'NULL' to use the current random-number state.

Value

Invisibly returns the normalized output path.

plot_sdg_cps	<i>Plot cp-scores across analyses using SDG icons</i>
--------------	---

Description

Draws one horizontal row per analysis and positions the 17 official United Nations Sustainable Development Goal icons at their respective cp-scores. Icon size increases linearly with the cp-score. The first matrix or data-frame row is displayed at the top. When 'file' is 'NULL', the figure is drawn on the current graphics device; otherwise, a PDF is written to the exact supplied path.

Usage

```
plot_sdg_cps(
  x,
  analysis_labels = NULL,
  icon_size_range = c(0.18, 0.62),
  style = c("cpscores", "side-by-side"),
  analysis_lines = TRUE,
  file = NULL,
  width = 11.2,
  height = 8.3
)
```

Arguments

<code>x</code>	A numeric matrix or data frame with analyses in rows and SDGs 1–17 in columns, in numerical SDG order. Values must be finite cp-scores between zero and one.
<code>analysis_labels</code>	Optional character vector containing one displayed label per analysis. By default, row names are used; if these are absent, sequential analysis labels are generated.
<code>icon_size_range</code>	Two positive numbers giving the minimum and maximum icon diameter in analysis-row units. Sizes are interpolated linearly from cp-scores zero to one.
<code>style</code>	Either <code>"cpscores"</code> , which positions icons at their cp-scores, or <code>"side-by-side"</code> , which orders them from lower to higher priority on an evenly spaced grid centered around 0.5. Icon size represents the cp-score in both styles.
<code>analysis_lines</code>	Logical; whether to draw horizontal row guides restricted to the priority axis from zero to one. Defaults to <code>'TRUE'</code> .
<code>file</code>	<code>'NULL'</code> to draw on the current graphics device, or a path ending in <code>'pdf'</code> to write the figure.
<code>width, height</code>	PDF dimensions in inches when <code>'file'</code> is supplied.

Value

Invisibly returns the normalized PDF path when `'file'` is supplied; otherwise invisibly returns `'NULL'` after drawing the figure.

<code>plot_sdg_diamonds</code>	<i>Export SDG-priority diamonds for all group perspectives</i>
--------------------------------	--

Description

Creates one TIFF figure per group perspective in a Q approach result. Each figure arranges the 17 Sustainable Development Goal icons in the perspective's required Q-sort distribution, from lower to higher priority. Tied statements retain their order in the perspective table.

Usage

```
plot_sdg_diamonds(
  results,
  directory,
  filename = NULL,
  width = 30,
  height = 21,
  resolution = 300
)
```

Arguments

results	An object returned by [qapproach()] containing a numeric ‘perspectives’ table with 17 SDG columns and the standard ranking distribution from -3 to 3.
directory	Output directory. The directory is created when necessary. This argument is required because the function writes one TIFF per perspective.
filename	Optional shared basename for the generated files. ‘NULL’ creates one filename directly from each group-perspective name, prefixed with ‘sdgdiamond_’. A ‘.tif’ or ‘.tiff’ extension supplied with a custom basename is removed before the perspective suffix and final ‘.tiff’ extension are added.
width, height	TIFF dimensions in centimetres.
resolution	TIFF resolution in dots per inch.

Value

Invisibly returns the normalized paths of the written TIFF files.

qapproach	<i>Prepare rankings and run a Q approach analysis</i>
-----------	---

Description

‘prepare_rankings()’ converts participant-by-statement input into the statement-by-ranking orientation used by ‘qapproach()’. ‘qapproach()’ fits perspectives and computes consensus priority scores. The result retains the underlying eigenvalue-weighted mean z-scores as a technical output. The analysis stores recognized Q method conditions, factor-selection details, automatic distribution-repair diagnostics, and opposing rankings silently in ‘results\$diagnostics’. In particular, the affected ranking identifiers, perspectives, and loadings are available in ‘results\$diagnostics\$negative_flagging’. Unclassified conditions remain visible as warnings so that they can be reported and reviewed. The remaining aliases provide factor selection, unflagged rankings, and manual distribution repair.

Usage

```
prepare_rankings(dataset, id_column = "ID", statement_columns = NULL,
  add = list(NULL), orientation = c("auto", "participant_rows",
    "statement_rows"))
qapproach(dataset, nfactors = "criteria", rotation = "quartimax",
  load_perc = 0.8, min_load_perc = 0.5, morethan5 = FALSE,
  screeplot_file = NULL,
  repair_distributions = TRUE, distribution_repair_steps = NULL,
  distribution_repair_seed = 42L,
  distribution_repair_max_attempt_multiplier = 10L)
nfactordetermination(dataset, rotation, load_perc, morethan5 = FALSE,
  min_load_perc = 0.5)
not_agreeing(results, status = FALSE)
manually_repair_perspective_distributions(results, bootstrap = NULL,
  perspective = NULL, statement = NULL, value = NULL, interactive = TRUE,
  verbose = interactive)
```

Arguments

dataset	A data frame or matrix. Participant rows or already prepared statement rows are accepted.
id_column	Unique participant identifier column, or 'NULL'.
add	A list of optional additional rankings.
status	Whether 'not_agreeing()' adds a status column distinguishing opposing and undecided rankings.
statement_columns	Character vector selecting statement columns.
orientation	Either "auto", "participant_rows", or "statement_rows". Automatic mode uses IDs, names, dimensions, and distribution checks.
nfactors	"criteria" or an integer from 1 to 10. Automatic selection normally requires at least two positively and uniquely flagging rankings per perspective. If no strict solution reaches the target consensus, a perspective with exactly one agreeing ranking and no opposing rankings may be retained when it raises the effective consensus above every otherwise eligible solution. The exception is recorded in the factor-selection result and diagnostics.
rotation	Either "quartimax" (default for the Q approach) or "varimax".
load_perc	Requested proportion of significantly loading rankings.
min_load_perc	Minimum acceptable loading proportion.
morethan5	Whether automatic selection may evaluate up to ten factors.
screeplot_file	Optional PDF output path for the unrotated-factor scree plot. 'NULL' writes no file.
repair_distributions	Whether to repair broken perspective gradients.
distribution_repair_steps	Target valid repair iterations, or 'NULL'.
distribution_repair_seed	Integer seed used for automatic distribution repair, or 'NULL' to use the current random-number state. Defaults to '42L' so that this exceptional corrective step is reproducible.
distribution_repair_max_attempt_multiplier	Attempt-limit multiplier.
results	A result returned by 'qapproach()'.
bootstrap	Optional result from 'qaboos()'.
perspective	Perspective name or index to repair.
statement	Statement index to repair.
value	Replacement ranking value.
interactive	Whether to prompt for missing repair choices.
verbose	Whether the manual repair function prints its inspection and repair details. By default, this follows 'interactive'.

Value

'prepare_rankings()' returns statement-by-ranking data. 'qapproach()' returns the fitted analysis, perspectives, weighted z-scores, consensus priority scores, repair audit, and centralized diagnostics in 'results\$diagnostics'. When automatic factor selection is used, its details are available as '<object>\$factor_selection', including the captured diagnostic messages and warnings at '<object>\$diagnostics\$factor_selection'. Other functions return the result described above.

References

Geschke, J., Urbach, D., Prescott, G. W., & Fischer, M.(2022). The Q approach to consensus building: integrating diverse perspectives to guide decision-making. **ECOEVORXIV**. doi:10.32942/X2F59S

Geschke, J. (2024). **Q approach to consensus building**. doi:10.5281/zenodo.11518485

Examples

```
x <- data.frame(ID = c("A", "B"), stat1 = c(-1, 0),
  stat2 = c(0, 1), stat3 = c(1, -1))
prepare_rankings(x)
```

qapproach-bootstrap *Advanced Q approach bootstrap interfaces*

Description

Generate Q method bootstrap results or collect a requested number of valid bootstrap iterations of the consensus priority scores. Recognized R messages and warnings emitted by the underlying Q method bootstrap are retained as structured diagnostics without routine console messages; unfamiliar warnings are re-emitted and include the package's bug-report URL. Single-perspective solutions use Procrustes sign alignment. Solutions with two or three perspectives use 'qindtest', with the package's orthogonal Procrustes alignment as a documented fallback if 'qindtest' fails; solutions with more than three perspectives use orthogonal Procrustes alignment directly. Flags and z-scores are recalculated after Procrustes alignment.

Usage

```
qaboos(results, steps = 40, method = "multiplication", seed = NULL,
  max_batch_steps = 500L, max_attempt_multiplier = 10L,
  progress = interactive())
bootstrap_consensus_priority_scores(results, steps = NULL, seed = NULL,
  target_valid_steps = NULL, valid_steps_per_ranking = 40L,
  max_batch_steps = 500L, max_attempt_multiplier = 10L,
  progress = interactive())
```

Arguments

results	A result returned by ‘qapproach()’.
steps	Positive bootstrap step count, or ‘NULL’ in the consensus priority score wrapper.
method	Either “multiplication” or “manual”.
seed	Integer random seed or ‘NULL’. The default ‘NULL’ uses R’s current random-number state. Supply an integer, such as ‘42L’, for a reproducible run.
target_valid_steps	Target valid iterations, or ‘NULL’.
valid_steps_per_ranking	Default valid iterations per ranking.
max_batch_steps	Maximum iterations requested in one batch.
max_attempt_multiplier	Multiplier limiting total attempts.
progress	Whether to display bootstrap progress. The default uses ‘interactive()’. Set explicitly to ‘TRUE’ or ‘FALSE’ to override it.

Value

A list containing bootstrap estimates, iteration accounting, and structured diagnostics.

qapproach-plots	<i>Plot and export Q approach results</i>
-----------------	---

Description

Draw perspective rankings, consensus priority scores, networks, spiderwebs, or consensus priority score bootstrap distributions. ‘write_figure_collection()’ combines applicable figures and captions in a multi-page PDF.

Usage

```
plot_barplot(result, labels = NULL, statement_colors = NULL,
  network_labelled = FALSE, preset = NULL,
  show_normalized_weighted_z = FALSE,
  normalized_line_color = "black", normalized_line_width = 1.5,
  file = NULL, width = 11.2, height = 6.4)
plot_heatmap(result, labels = NULL, statement_colors = NULL,
  network_labelled = FALSE, preset = NULL, file = NULL,
  width = 11.2, height = 6.4)
plot_jitterplot(result, labels = NULL, statement_colors = NULL,
  network_labelled = FALSE, preset = NULL, file = NULL,
  width = 11.2, height = 6.4)
```

```

plot_network(result, labels = NULL, statement_colors = NULL,
  network_labelled = FALSE, network_perspective_label_cex = 2,
  network_ranking_label_cex = 1.1, network_dataset_colors = NULL,
  network_synthesis_color = NULL, network_cps_color = NULL,
  input_network = FALSE, negative_loadings_marked = TRUE,
  network_arrow_size = 0.56, layout_seed = NULL,
  .mixed_synthesis_inputs = FALSE,
  preset = NULL, file = NULL, width = 11.2, height = 6.4)
plot_spiderweb(result, labels = NULL, statement_colors = NULL,
  network_labelled = FALSE, preset = NULL, file = NULL,
  width = 8.3, height = 8.3)
write_figure_collection(file, result, validation = NULL, labels = NULL,
  statement_colors = NULL, network_labelled = FALSE,
  layout_seed = NULL, figure_assets = NULL, preset = NULL,
  two_layered = FALSE)

```

Arguments

result	A Q approach result. For ‘plot_jitterplot()’, this may instead be an object returned by ‘validate()’ or a bootstrap result.
validation	Optional object returned by ‘validate()’, used by ‘write_figure_collection()’ for the validation jitterplot.
labels	Optional statement labels.
statement_colors	Optional statement colors.
show_normalized_weighted_z	Whether ‘plot_barplot()’ overlays the weighted z-scores after linearly rescaling them to the observed cp-score range. This specialist display option defaults to ‘FALSE’.
normalized_line_color, normalized_line_width	Color and width of the rescaled weighted-z-score stair line. These settings are used only when ‘show_normalized_weighted_z = TRUE’.
network_labelled	Whether ranking identifiers appear in the network.
network_perspective_label_cex, network_ranking_label_cex	Label-size multipliers for perspective and ranking nodes in ‘plot_network()’.
network_dataset_colors, network_synthesis_color, network_cps_color	Optional colors for multi-dataset, synthesis, and cp-score nodes.
input_network	Whether the network represents analytical inputs rather than agreement and opposition results.
negative_loadings_marked	Whether negative flagged loadings are marked.
network_arrow_size	Arrow-size multiplier for ‘plot_network()’.
layout_seed	Integer seed for reproducible network-node placement, or ‘NULL’ to use the current random-number state.

<code>.mixed_synthesis_inputs</code>	Internal logical used by the two-layer network wrapper when synthesis inputs contain both dataset perspectives and re-added individual rankings. Manual users should leave this at its default.
<code>preset</code>	'NULL', "sdg", "tca-actions", or "tca-strategies".
<code>file</code>	Output path. For individual plot functions, 'NULL' draws on the current graphics device without writing a file. For 'write_figure_collection()', 'file' is required.
<code>width, height</code>	Output dimensions in inches when 'file' is supplied.
<code>figure_assets</code>	Optional pre-created internal figure assets.
<code>two_layered</code>	Whether the figure collection contains a two-layered multi-dataset network caption.

Value

Plot functions draw on the current device. The writer invisibly returns the normalized output path.

summary	<i>Print a Q approach analysis summary</i>
---------	--

Description

Prints and returns analysis-summary, group-perspective, and consensus-priority-score tables from an object returned by `qapproach()`. The analysis summary includes the observed cp-score range in [minimum, maximum] format with two decimal places.

Usage

```
summary(results, print_table = TRUE, file = NULL)
```

Arguments

<code>results</code>	An object returned by <code>qapproach()</code> .
<code>print_table</code>	Logical; print the table when TRUE.
<code>file</code>	Optional CSV file path. The three tables are written below one another in a single padded CSV file, separated by two blank rows. If the .csv extension is omitted, it is added automatically. The default NULL writes no file.

Value

Invisibly returns a named list containing the analysis summary, group perspectives, and consensus priority scores tables. Summary cp-scores are rounded to two decimal places; the source values in results remain unchanged.

See Also

[qapproach](#)

validate

Validate group perspectives and consensus priority scores

Description

Runs three complementary procedures: bootstrap stability of the group perspectives, bootstrap stability of the consensus priority scores and ranks, and a sensitivity comparison with input-ranking means transformed onto the same fixed standard-normal cumulative-probability scale. Bootstrap factor alignment uses orthogonal Procrustes sign alignment for one perspective, ‘qindtest’ for two or three perspectives, and the package’s orthogonal Procrustes implementation for larger solutions. If ‘qindtest’ fails, the affected batch is rerun with orthogonal Procrustes alignment and reported with a message referring to this help page. The fallback preserves the intended factor correspondence when ‘qindtest’ cannot provide a unique alignment; its use and the original error are retained in the diagnostics. Recognized underlying conditions, routine alignment corrections, invalid iterations, and discard reasons are retained silently in ‘validation\$diagnostics’. Unclassified conditions are emitted as warnings so that they can be reported and reviewed.

Usage

```
validate(
  results,
  bootstrap = NULL,
  bootstrap_scores = NULL,
  statement_labels = NULL,
  confidence_level = 0.95,
  rank_cutoffs = c(1L, 3L, 5L),
  target_valid_steps = NULL,
  valid_steps_per_ranking = 40L,
  seed = NULL,
  max_batch_steps = 500L,
  max_attempt_multiplier = 10L,
  zscore_instability_threshold = 0.2,
  progress = interactive()
)
```

Arguments

results	An object returned by ‘qapproach()’.
bootstrap	Optional object returned by ‘qaboos()’. If omitted, one is generated for the group-perspective diagnostics and reused for consensus priority scores validation wherever possible.
bootstrap_scores	Optional object returned by ‘bootstrap_consensus_priority_scores()’.
statement_labels	Optional statement labels in analysis order.

confidence_level	Confidence level for bootstrap intervals.
rank_cutoffs	Rank cutoffs used for top probabilities and sensitivity comparisons; defaults to 1, 3, and 5.
target_valid_steps	Target number of valid bootstrap iterations of the consensus priority scores. By default this is 'valid_steps_per_ranking' times the number of rankings.
valid_steps_per_ranking	Default number of valid iterations per input ranking.
seed	Integer bootstrap seed or 'NULL'. The default 'NULL' uses the current random-number state. Supply an integer, such as '42L', for a reproducible validation run.
max_batch_steps, max_attempt_multiplier	Passed to the adaptive bootstrap of the consensus priority scores when additional valid iterations are required.
zscore_instability_threshold	Absolute z-score bias used for the descriptive instability flag.
progress	Whether to display bootstrap progress. The default uses 'interactive()', so progress is shown in interactive R sessions and hidden in non-interactive use. Set explicitly to 'TRUE' or 'FALSE' to override it.

Value

A list containing three validation-procedure results and centralized diagnostics in 'validation\$diagnostics'. These diagnostics contain the alignment methods and fallbacks, recognized bootstrap conditions, invalid iterations and their discard reasons, and the input-mean correlation diagnostic.

validation_cps	<i>Print the consensus-priority-score validation table</i>
----------------	--

Description

Extracts the table already calculated by 'validate()', prints a readable version in the console, and invisibly returns the displayed results for assignment or export. Score and rank confidence intervals are each combined in one column using '[lower, upper]' notation.

Usage

```
validation_cps(
  validation,
  digits = 2L,
  print_table = TRUE,
  sort_by = "cp-scores",
  decreasing = TRUE,
  include_bottom = FALSE,
  file = NULL
)
```

Arguments

validation	An object returned by <code>'validate()'</code> .
digits	Number of decimal places used for console display.
print_table	Print the formatted heading, metadata, and table. Set to <code>'FALSE'</code> when only the returned data frame is needed.
sort_by	Column used to sort the returned table. The default is <code>"cp-scores"</code> ; use <code>'NULL'</code> to retain the original statement order. Any returned technical column name may be supplied.
decreasing	Logical; sort in decreasing order.
include_bottom	Include the <code>'P bottom 1'</code> , <code>'P bottom 3'</code> , and <code>'P bottom 5'</code> columns in the returned, printed, and exported table. The default <code>'FALSE'</code> hides these columns only. <code>'validate()'</code> always calculates and stores the bottom-rank probabilities, and the validation assessment always uses them.
file	Optional CSV path. The default <code>'NULL'</code> writes no file. If the <code>'csv'</code> extension is omitted, it is added automatically. The exported table retains numeric values for further analysis.

Details

`'Weighted z-score'` is the eigenvalue-weighted mean of the statement's z-scores across all group perspectives. Positive values indicate relatively higher prioritization, negative values indicate relatively lower prioritization, and zero represents average prioritization across the group perspectives. Differences between weighted z-scores provide a linear measure of the prioritization gap between statements.

Value

Invisibly, the validation data frame with merged interval columns.

validation_means	<i>Print the sensitivity comparison with input-ranking means</i>
------------------	--

Description

Creates a one-row-per-statement comparison of consensus priority scores with input-ranking means transformed onto the same fixed standard-normal cumulative-probability scale. This is a sensitivity analysis, not a test of whether the consensus priority scores are valid: differences may reflect the intended perspective-based weighting.

Usage

```
validation_means(
  validation,
  digits = 2L,
  print_table = TRUE,
  sort_by = "cp-scores",
  decreasing = TRUE,
  file = NULL
)
```

Arguments

validation	An object returned by ‘validate()’.
digits	Number of decimal places used for console display.
print_table	Print the formatted heading, metadata, and table. Set to ‘FALSE’ when only the returned data frame is needed.
sort_by	Column used to sort the returned table. The default is “cp-scores”; use ‘NULL’ to retain the original statement order. Any returned technical column name may be supplied.
decreasing	Logical; sort in decreasing order.
file	Optional CSV path. The default ‘NULL’ writes no file. If the ‘.csv’ extension is omitted, it is added automatically. The exported table retains numeric values for further analysis.

Details

The direction of the rank-change measure intentionally differs from the arithmetic order named in its column heading. Because lower rank numbers indicate higher priority, a positive rank change means that the statement moved upward under the cp-score ranking. Positive values in both the score- difference and rank-change columns therefore indicate a higher cp-score priority relative to the input means. When input means are tied, ‘Priority comparison’ reports ‘Tied under input means’ instead of assigning a directional interpretation.

The comparison of the cp-scores with the input means is a sensitivity analysis, not a test of whether the consensus priority scores are valid. Differences can reflect the intended perspective-based weighting: input means treat every ranking equally, whereas cp-scores explicitly represent group perspectives. A small difference may indicate that perspective-based weighting has little effect in the dataset, which is itself informative.

Value

Invisibly, the unformatted statement-comparison data frame.

validation_perspectives

Print the group-perspective stability validation table

Description

Creates a one-row-per-perspective overview from the first validation procedure returned by ‘validate()’. The assessment is descriptive: it flags rank-order agreement below ‘rank_correlation_threshold’ and any statement whose absolute bootstrap z-score bias reaches the threshold supplied to ‘validate()’.

Usage

```
validation_perspectives(
  validation,
  digits = 2L,
  rank_correlation_threshold = 0.9,
  print_table = TRUE,
  sort_by = NULL,
  file = NULL
)
```

Arguments

validation	An object returned by ‘validate()’.
digits	Number of decimal places used for console display.
rank_correlation_threshold	Minimum Spearman correlation treated as stable rank-order agreement.
print_table	Print the formatted heading, metadata, table, and any alignment caution. Set to ‘FALSE’ when only the returned data frame is needed.
sort_by	Optional returned column name used to sort the table in decreasing order. Use ‘NULL’ to retain perspective order.
file	Optional CSV path. The default ‘NULL’ writes no file. If the ‘.csv’ extension is omitted, it is added automatically. The exported table retains numeric values for further analysis.

Value

Invisibly, the detailed perspective-validation data frame.

Index

agreement_across_levels, [2](#)

bootstrap_consensus_priority_scores
(qapproach-bootstrap), [13](#)

consensus_across_levels, [3](#)
cpscores, [4](#)

distributiondetermination, [4](#)

manually_repair_perspective_distributions
(qapproach), [11](#)

nfactordetermination (qapproach), [11](#)
not_agreeing (qapproach), [11](#)

plot_barplot (qapproach-plots), [14](#)
plot_heatmap (qapproach-plots), [14](#)
plot_hierarchical_levels, [5](#)
plot_jitterplot (qapproach-plots), [14](#)
plot_network (qapproach-plots), [14](#)
plot_network_two_layered, [8](#)
plot_sdg_cps, [9](#)
plot_sdg_diamonds, [10](#)
plot_spiderweb (qapproach-plots), [14](#)
prepare_rankings (qapproach), [11](#)

qaboos (qapproach-bootstrap), [13](#)
qapproach, [11](#), [16](#)
qapproach-bootstrap, [13](#)
qapproach-plots, [14](#)

summary, [16](#)

validate, [17](#)
validation_cps, [18](#)
validation_means, [19](#)
validation_perspectives, [21](#)

write_figure_collection
(qapproach-plots), [14](#)