Package ‘qapproach’

September 23, 2026

Type Package

Title The Q Approach to Consensus Building

Version 0.1.1

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 2026-09-23 02:50:02 UTC

Contents

`consensus_across_levels` *Summarize consensus across successive Q approach levels*

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_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.

`qapproach` *Prepare rankings and run a Q approach analysis*

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 remaining aliases provide factor selection, unflagged rankings, and manual distribution repair.

Usage

```
prepare_rankings(dataset, idcolumn = "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 = NULL,
  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.
idcolumn	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 or ‘NULL‘.
distribution_repair_max_attempt_multiplier	Attempt-limit multiplier.
results	A result returned by ‘qapproach()‘.
bootstrap	Optional result from ‘qaboots()‘.
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 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; unfamiliar warnings are re-emitted (feel free to report them). 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

```
qaboots(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` *Plot and export Q approach results*

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.

`.mixed_synthesis_inputs`
: 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.

`preset`
: 'NULL', '"sdg"', '"tca-actions"', or '"tca-strategies"'.

`file`
: Output path. For individual plot functions, 'NULL' draws on the current graphics device without writing a file. For 'write_figure_collection()', 'file' is required.

`width, height`
: Output dimensions in inches when 'file' is supplied.

`figure_assets`
: Optional pre-created internal figure assets.

`two_layered`
: 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` *Print a Q approach analysis summary*

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

`results`
: An object returned by `qapproach()`.

`print_table`
: Logical; print the table when `TRUE`.

`file`
: 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 transparently reported as ‘qindtest with orthogonal Procrustes fallback‘.

Usage

```
validate(
  results,
  bootstrap = NULL,
  bootstrap_scores = NULL,
  statement_labels = NULL,
  confidence_level = 0.95,
  rank_cutoffs = c(1L, 3L, 5L),
  include_bottom = FALSE,
  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 ‘qaboots()‘. 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.

include_bottom Include corresponding bottom-rank probabilities in the consensus priority scores' validation table.

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.

---

validation_cps *Print the consensus-priority-score validation table*

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,
  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

‘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 *Print the sensitivity comparison with input-ranking means*

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, table, and interpretation. 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.

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