

A Q Approach to Consensus Building

Jonas Geschke

September 12, 2026

1 Purpose and workflow

The `qapproach` package combines Q method with a consensus priority score (CPS). Participants rank statements, the analysis identifies shared group perspectives, and the CPS summarizes priorities across those perspectives while weighting them by their explanatory importance.

The recommended workflow is to prepare and orient rankings with `prepare_rankings()`, fit perspectives and CPS with `qapproach()`, optionally assess stability with `validate()`, and communicate the results with the plotting and PDF-export functions.

2 Prepare rankings

Input commonly has one participant per row, an identifier column, and one numeric column per statement. The preparation function transposes it into the statement-by-ranking format required by the analysis.

```
library(qapproach)
participant_data <- data.frame(
  ID = paste0("P", 1:6),
  stat1 = c(-1, -1, 0, 0, 1, 1),
  stat2 = c(0, 0, -1, 1, -1, 0),
  stat3 = c(1, 1, 1, -1, 0, -1)
)
rankings <- prepare_rankings(participant_data)
rankings
```

Identifiers must be unique and statement values numeric. Use the `statement_columns` argument for other naming conventions. Resolve missing observations before analysis.

3 Fit the Q approach

Use `nfactors = "criteria"` for automatic selection or a fixed integer for a theory-led fit. A typical empirical call is:

```
fit <- qapproach(rankings, nfactors = "criteria",
  rotation = "quartimax",
  distribution_repair_seed = 42L)
fit$summary
fit$perspectives
fit$`cp-scores`
fit$factor_selection$diagnostics
fit$diagnostics
```

The result retains factor-selection and final-fit diagnostics. If a discrete perspective distribution is broken, the default analysis bootstraps a reference and repairs it automatically. The audit trail is stored in `fit$distribution_repair`. CPS remain based on the original Q method z-scores.

`not_agreeing(fit)` returns opposing and undecided rankings that did not agree with a retained perspective. By default, it returns only numeric ranking columns for direct reuse as input; `status = TRUE` adds a `Status` column that distinguishes opposing and undecided rankings. `nfactordetermination()` runs only factor selection, and `manually_repair_perspective_distributions()` supports explicitly controlled repair.

4 Validate stability

Validation is optional and can take substantially longer because it generates bootstrap samples. A seed makes resampling reproducible. Single-perspective solutions use Procrustes sign alignment, solutions with two or three perspectives use qindtest alignment, and larger solutions use orthogonal Procrustes alignment before flags and z-scores are recalculated.

```
validation <- validate(fit, seed = 2026)
validation_perspectives(validation)
validation_cps(validation)
validation_means(validation)
```

Perspective validation examines factor and statement-position stability. Cp-score validation reports score and rank intervals, bias, standard errors, and top-rank frequencies. These frequencies describe stability under the observed data and workflow; they are not probabilities of objective importance. Input-mean sensitivity compares cp-scores with input-ranking means transformed onto the same fixed standard-normal cumulative-probability scale. Advanced users can call `qaboos()` or `bootstrap_consensus_priority_scores()` directly.

cp-scores use a fixed scale on which 0.5 represents neutral prioritization across all group perspectives, values above 0.5 represent relatively higher priority, and values below 0.5 represent relatively lower priority. Values 0 and 1 are theoretical boundaries. cp-scores can be compared across analyses only when the statement set and meanings, ranking distribution, instructions, data preparation, and analytical settings are the same. Such comparisons describe relative priorities within the shared statement set and do not alone establish population-level differences.

5 Visualize and export

```
plot_barplot(fit)
plot_heatmap(fit)
plot_spiderweb(fit)
plot_network(fit, network_labelled = TRUE, layout_seed = 42L)
plot_jitterplot(validation)
write_figure_collection(
  file.path(tempdir(), "qapproach-figures.pdf"),
  result = fit, validation = validation, layout_seed = 42L
)
```

The visualization API recognizes the `sdg`, `tca-actions`, and `tca-strategies` presets. Explicit labels and colors take precedence.

6 Reporting guidance

Report input orientation and preprocessing, the factor-selection rule, rotation, retained perspectives, flagging threshold, distribution-repair audit, and material diagnostics. For validation, also report the bootstrap target and achieved valid iterations, confidence level, rank cutoffs, seed, and any non-estimable or unstable results.