

## Overview of R functions to run the Q approach to consensus building workflow

accompanying qapproach\_functions\_v1.R  
<https://doi.org/10.5281/zenodo.11518486>

Workflow step	Function	Input	Output/Information
<i>a.</i> Data collection	distributiondetermination()	nstat = the number of statements, should be between 5 and 50	<p>\$distr = the format of the forced quasi-normal distribution</p> <p>\$rankingvalues = the values which the statements have to be ranked in</p> <p>\$ranking = the values as needed for the ranking</p>
<i>b.</i> Subsampling	prepare_rankings()	<p>dataset = Q sorts, with the columns being the statements and the rows being the rankings</p> <p>nstat = if the dataset has columns other than the statements and the ranking ID, provide the number of statements. default is NULL</p> <p>idcolumn = the name of the column containing the ranking IDs, make it NA if there is no ID column</p> <p>add = a list of rankings to be added to the dataset, e.g. used for workflow step <i>i</i> (see main text for details)</p> <p>*regardless of what they are, the statement columns should be named “stat1”, “stat2”, etc.</p>	a dataframe usable for qapproach( ), with the columns being rankings and the rows being statements
Q approach analysis (incl. workflow steps <i>c</i> to <i>g</i> )	<p>nfactordetermination()</p> <p>*this is a helper function used in qapproach()</p>	<p>dataset = the Q sorts, i.e. the output from prepare_rankings()</p> <p>rotation = the factor rotation method to be used</p> <p>load_perc = a numerical value: minimum factor flagging coefficient aimed for</p> <p>morethan5 = logical if more than five factors should be</p>	the number of how many factors to analyze, based on the five hierarchical criteria presented in workflow step <i>c</i> (see main text for details)

		allowed. if set to TRUE, up to 10 factors are analyzed. default is FALSE (this is an option to undermine criterion 2 of the factor optimization. if possible, it should only be used to explore whether a result can be calculated for 6 to 10 factors, as the ability to reach consensus with so many factors only seems theoretical)	
	cpscores( )  *this is a helper function used in qapproach( )	statementzscores = the z-scores of the statements, from the standard Q method results  factoreigenvalues = the factor eigenvalues, from the standard Q method results	The consensus priority scores for the given input statements
	qapproach( )	dataset = the Q sorts, with the columns being rankings and the rows being statements, i.e. the output from prepare_rankings( )  nfactors = a number of how many factors to analyze, or “criteria” (default) to apply the helper function nfactordetermination( )  rotation = the factor rotation method to be used. default is “quartimax”  load_perc = a numerical value: minimum factor flagging coefficient aimed for. default is 0.8	\$`dataset name` = the name of the input dataset  \$`Q method results` = the standard Q method results from the function/package qmethod( )  \$summary = an overview of i. input rankings, ii. unrotated principal components, iii. factors analyzed, iv. rankings flagging, and v. the factor flagging coefficient  *as part of the function, a pdf showing a screeplot of unrotated factors is saved to the folder getwd() > figures
h. Validation	qaboos( )	results = the output from qapproach( )  steps = a numerical value: to set the number of bootstrapping steps. default is 40  method = the method to be used to determine the number of bootstrapping steps. if “multiplication” (default), the number of Q sorts analyzed in qapproach( ) is multiplied with the value set in the steps argument. if “manual”, the	\$`dataset name` = the name of the input dataset  \$`Q approach results` = the results given as input to the bootstrap  \$`bootstrap results` = the results of the Q approach result’s bootstrapping

		value set in the steps argument is taken as the number of bootstrapping steps, without multiplication or anything else	
	check_distribution()  *used without assignment operator "<-"	results = the output from qapproach()	a message "All ranking distributions look fine." or an information which perspective(s) has/have a broken distribution. in the latter case, use edit_distribution() to fix the distribution(s)
	edit_distribution()	results = the output from qapproach()  bootstrap = the output from qaboos()  perspective = a character string of the perspective to be edited, e.g. "groupA_fl"	fixed distribution(s) in the results object
	cpscoresboots()  *this is a helper function used in validate()	bootstrap = the output from qaboos()	\$`bootstrap z-scores` = the z-scores of the bootstrap  \$`bootstrap eigenvalues` = the eigenvalues of the bootstrap factors  \$`bootstrap cp-scores` = the consensus priority scores for the bootstrap results
	validate()	results = the output from qapproach()  bootstrap = the output from qaboos()	<i>see the merged field below</i>
	<i>Output/Information for validate()</i>  \$`dataset name` = the name of the input dataset  \$`Q approach results` \$`perspectives` = the perspectives resulting from qapproach() \$`cp-scores` = the consensus priority scores resulting from qapproach()  \$`input means` = the results for workflow step h.1 ( <i>see main text for details</i> ) \$`normalized input means` = the means of the input sorts in a format comparable to the consensus priority scores \$`proportion of 12 lowest ranked original cp-scores matching` = a percent value of how many lower priority statements also have a lower input mean \$`proportion equal or higher than 80%` = a logical if the latter value is higher than 0.8 or not		

	<p>\$`3 top original cp-scores among 5 top input means` = a logical if the top three consensus priority scores are among the top five input means (if FALSE here, the priority ranking needs some more discussion among participants)</p> <p>\$`position changes of ranked original cp-scores` = the statements in the order of their consensus priority score ranking, with a numerical value of position changes in the input means</p> <p>\$`TOST comparing original cp-scores and input means` = the results of a two one-sided equivalence test comparing the consensus priority scores and the input means (if the null hypothesis of statistical difference is rejected, the consensus priority scores are considered significant)</p> <p>\$`bootstrap results` = the results for workflow step <i>h.2</i> (<i>see main text for details</i>)</p> <p>\$`bootstrap factors` = the factors resulting from <code>qaboos()</code></p> <p>\$`position changes of perspectives` = the statements and, based on the perspectives resulting <code>qapproach()</code>, the position changes they have in the bootstrap factors</p> <p>\$`bootstrap factor eigenvalues` = the factor eigenvalues resulting from <code>qaboos()</code></p> <p>\$`bootstrap z-scores` = the z-scores resulting from <code>qaboos()</code></p> <p>\$`bootstrap z-scores estimate of bias` = the difference between the z-scores from <code>qapproach()</code> and those from <code>qaboos()</code></p> <p>\$`bootstrap z-scores unstable` = logicals if the z-scores estimate of bias is higher than 0.2 or not</p> <p>\$`bootstrap cp-scores` = the consensus priority scores calculated for the bootstrap factor scores, i.e. the bootstrap z-scores and the bootstrap factor eigenvalues</p> <p>\$`proportion of 12 lowest ranked original cp-scores matching` = a percent value of how many lower priority statements also have a lower bootstrap cp-score</p> <p>\$`proportion equal or higher than 80%` = a logical if the latter value is higher than 0.8 or not</p> <p>\$`3 top original cp-scores among 5 top bootstrap cp-scores` = a logical if the top three consensus priority scores are among the top five bootstrap cp-scores (if FALSE here, the priority ranking needs some more discussion among participants)</p> <p>\$`position changes of ranked original cp-scores` = the statements in the order of their consensus priority score ranking, with a numerical value of position changes in the bootstrap cp-scores</p> <p>\$`TOST comparing original cp-scores and bootstrap cp-scores` = the results of a two one-sided equivalence test comparing the consensus priority scores and the bootstrap cp-scores (if the null hypothesis of statistical difference is rejected, the consensus priority scores are considered significant)</p> <p>\$`bootstrapped cp-scores` = the results for workflow step <i>h.3</i> (<i>see main text for details</i>)</p> <p>\$`bootstrapped cp-scores` = the bootstrapping results for the consensus priority scores</p> <p>\$`sdg statistics` = some basic statistics for the individual statements, resulting from the bootstrapping: median, mean, standard deviation (sd), the significance test applied, the p-value, and the significance level</p> <p>\$`TOST comparing original cp-scores and bootstrapped cp-scores` = the results of a two one-sided equivalence test comparing the consensus priority scores and the bootstrapped cp-scores (if the null hypothesis of statistical difference is rejected, the consensus priority scores are considered significant)</p>		
<i>i. Cross-level loop</i>	<code>not_flagging()</code>	results = the output from <code>qapproach()</code>	a dataframe usable for, e.g., the “add” option in <code>prepare_rankings()</code> , with the columns being the statements and the rows being the rankings