Package ‘resquin’

June 27, 2025

Title Response Quality Indicators for Survey Research

Version 0.1.1

Description Calculate common survey data quality indicators for multi-item scales and matrix questions. Currently supports the calculation of response style indicators and response distribution indicators. For an overview on response quality indicators see Bhaktha N, Henning S, Clemens L (2024). 'Characterizing response quality in surveys with multi-item scales: A unified framework' <https://osf.io/9gs67/>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Imports cli, purrr, rlang, slider, stringi, tibble, vctrs

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tidyr

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/MatRoth/resquin,
https://matroth.github.io/resquin/

BugReports https://github.com/MatRoth/resquin/issues

Depends R (>= 4.1)

LazyData true

NeedsCompilation no

Author Matthias Roth [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0003-4369-8106>),
Nivedita Bhaktha [aut, ctb],
Matthias Bluemke [aut, ctb],
Thomas Knopf [aut, ctb],
Fabienne Krämer [aut, ctb],
Clemens Lechner [aut, ctb],
Çağla Yildiz [aut, ctb]

Maintainer Matthias Roth <matthias.roth@gesis.org>

Repository CRAN

Date/Publication 2025-06-27 07:50:01 UTC

Contents

---

flag_resp *Flag respondents based on response quality indicators*

Description

Flag respondents with one or more flagging expression.

Usage

```
flag_resp(x, ...)
```

Arguments

`x`	A data frame containing response quality indicators. Each column should be one response quality indicator. Each row should be the value of the response quality indicator of a respondent.
`...`	Flagging expressions. See details.

Details

`flag_resp()` works very similar to the popular `dplyr::filter()` function. However, instead of filtering data, `flag_resp()` returns a data frame of `T` and `F` values, representing which respondents are flagged.

As the first argument, you provide a data frame of response quality indicators, where each column represents one response quality indicator and each row represents one respondent. As the second argument you provide one ore more logical statements to flag respondents. For example:

- flag_resp(x,ERS > 0.5) returns a data frame with one column named ERS > 0.5. Each row represents one respondent and shows whether the statement "is the extreme response style indicator larger than 0.5" is true (T) or false (F).
- flag_resp(x,ERS > 0.5,ii_mean < 3) returns a data frame with two columns indicating for which respondents the two flagging expressions are true or false.

Note that flag_resp() is not restricted to functions from the resquin package. You can supply any numerical column in the data frame x. This opens the possibility to compare flagging strategies based on response quality indicators across packages and functions.

Use the summary() function on the results to compare flagging strategies.

For more details see the vignette: vignette("flagging_respondents", package = "resquin")

Value

A data frame containing one column per flagging strategy and the same number of rows asx. Each column contains T and F flags per respondents. An additional id column is added as the first column if a column named id is present in x.

Examples

```
res_dist_indicators <- resp_distributions(nep) # Create indicator data frame

flagged_respondents <- flag_resp(res_dist_indicators,
                                 ii_mean > 3, # Flagging strategy 1
                                 ii_sd < 2, # Flagging strategy 2
                                 ii_mean > 3 & ii_sd > 2) # Flagging strategy 3
flagged_respondents # A data frame with three columns, each corresponding to one flagging strategy
summary(flagged_respondents) # quickly compare flagging strategies
```

---

nep *NEP-Scale GESIS Panel Campus File*

Description

Responses on 15 items of the NEP scale (Dunlap et al., 2002) measuring attitudes towards the environment. The data is from the GESIS Panel Campus File (Bosnjak et al., 2017, GESIS Data Archive, 2025), which is a subset of the full GESIS Panel. The GESIS Panel is a probability based general population panel survey sampling from the German population.

Usage

```
nep
```

Format

`nep`:

A data frame with 1,222 rows and 15 columns:

- bczd005a: Approaching maximum number of humans
- bczd006a: The right to adapt environment to the needs
- bczd007a: Consequences of human intervention
- bczd008a: Human ingenuity
- bczd009a: Abuse of the environment by humans
- bczd010a: Sufficient natural resources
- bczd011a: Equal rights for plants and animals
- bczd012a: Balance of nature stable enough
- bczd013a: Humans are subjected to natural laws
- bczd014a: Environmental crisis greatly exaggerated
- bczd015a: Earth is like spaceship
- bczd016a: Humans were assigned to rule over nature
- bczd017a: Balance of nature is very sensitive
- bczd018a: Control nature
- bczd019a: Environmental disaster

Details

Responses are on a five point response scale, which has been inverted from its original coding:

- 5 = Fully agree
- 4 = Agree
- 3 = Neither nor
- 2 = Don't agree
- 1 = Fully disagree

Note that some of the items are reverse coded, meaning that higher agreement with the scale can either indicate more concern for nature (e.g. bczd017a: Balance of nature is very sensitive), while higher agreement to other items implies less concern for nature (bczd005a: Approaching maximum number of humans).Thus, straightling behavior is much less likely a result of valid responding.

Source

Bosnjak, M.; Dannwolf, T.; Enderle, T.; Schauer, I.; Struminskaya, B.; Tanner, A. und Weyandt, Kai W. (2017): Establishing an open probability-based mixed-mode panel of the general population in Germany: The GESIS Panel. Social Science Computer Review, 36(1). https://doi.org/10.1177/0894439317697949

Dunlap, Riley E., Kent D. Van Liere, Angela G. Mertig, and Robert Emmet Jones (2002). "New Trends in Measuring Environmental Attitudes: Measuring Endorsement of the New Ecological Paradigm: A Revised NEP Scale." Journal of Social Issues 56 (3): 425–42. https://doi.org/10.1111/0022-4537.00176.

GESIS Data Archive, Cologne (2025). ZA5666 Data file Version 1.0.0, https://doi.org/10.4232/1.12749

plot.resp_indicator *Plot function for resp_indicator objects*

Description

Provides an overview over results of resp_* functions.

Usage

```
## S3 method for class 'resp_indicator'
plot(x, y, ...)
```

Arguments

x	An object of type resp_indicator created with a resp_* function.
y	Not used and thus not required.
...	Additional arguments (currently not supported).

Value

Invisibly returns the input x.

Examples

```
resp_distributions(nep) |> plot()
```

resp_distributions *Compute response distribution indicators*

Description

Compute response distribution indicators for responses to multi-item scales or matrix questions.

Usage

```
resp_distributions(x, min_valid_responses = 1, id = T)
```

Arguments

`x` — A data frame containing survey responses in wide format. For more information see section "Data requirements" below.

`min_valid_responses` — Numeric between 0 and 1 of length 1. Defines the share of valid responses a respondent must have to calculate response quality indicators. Default is 1.

`id` — default is `True`. If the default value is supplied a column named `id` with integer ids will be created. If `False` is supplied, no id column will be created. Alternatively, a numeric or character vector of unique values identifying each respondent can be supplied. Needs to be of the same length as the number of rows of `x`.

Details

The following response distribution indicators are calculated per respondent:

- n_na: number of intra-individual missing answers
- prop_na: proportion of intra-individual missing responses
- ii_mean: intra-individual mean
- ii_median: intra-individual median
- ii_sd: intra-individual standard deviation
- mahal: mahalanobis distance per respondent.

Intra-individual response variability (ii_sd) has been proposed to measure insufficient effort responding (Dunn et al., 2018) and to distinguish between random and conscientious responding (Marjanovic et al, 2015).

Intra-individual location indicators can be used to asses the average location of responses on a set of questions (ii_mean, ii_median).

Mahalanobis distance is a outlier detection indicator. It represents the distance of a participants responses from the center of a multivariate normal distribution defined by the data of all respondents.

Value

Returns a data frame with response quality indicators per respondent. Dimensions:

- Rows: Equal to number of rows in x.
- Columns: Six response distribution indicator colunns + id column (if specified).

Data requirements

`resp_distributions()` assumes that data comes from multi-item scales or matrix questions, which have the same number and labeling of response options for many questions. The input data frame must be structured in the following way:

- The data frame is in wide format, meaning each row represents one respondent, each column represents one variable.
- All responses have integer values.
- Missing values are set to `NA`.

Reverse coding of variables

The interpretation of the indicators depends on the whether response data of negatively worded questions was reversed or not:

- Do not reverse data of negatively worded questions if you want to assess average response patterns (Dunn et al., 2018).
- Reverse data of negatively worded questions if you want to assess whether responses are distributed randomly or not with respect to an assumed latent variable (Marjanovic et al., 2015).

Mahalanobis distance

Mahalanobis distance differs from other computed indicators in that its value represents the distance of the respondents responses to a set of average responses of the sample. Thus, the mahalanobis distance relates the individual to the sample whereas other indicators in `resp_distributions()` describe the response distribution of a single respondent.

Under certain circumstances, the mahalanobis distance can not be calculated. This may be if there is high collinearity (correlation between variables) or if there are to many missing values. Although this can happen in survey research data, this message can also indicate that something in the data is "off" due to one of the reasons stated above. A manual inspection for low-quality responses can be a next step.

A second issue with the calculation of mahalanobis distance values is, that it requires all data to be non-missing. This is the case if min_valid_responses = 1. However, if missing values are allowed, we use within respondent mean imputation to allow the calculation of mahalanobis distance values. This may lead to nonsensical mahalanobis distance values if the share of missing responses of a respondent is large and the respondent would actually have answered differently from their average response. If you want to calculate mahalanobis distance values for respondents with missing values, it is advisable to take a careful approach. Investigate missing patterns and compare results between different levels of min_valid_responses.

Author(s)

Matthias Roth, Matthias Bluemke & Clemens Lechner

References

Dunn, Alexandra M., Eric D. Heggestad, Linda R. Shanock, and Nels Theilgard. 2018. “Intra-Individual Response Variability as an Indicator of Insufficient Effort Responding: Comparison to Other Indicators and Relationships with Individual Differences.” Journal of Business and Psychology 33(1):105–21. doi: 10.1007/s10869-016-9479-0.

Marjanovic, Zdravko, Ronald Holden, Ward Struthers, Robert Cribbie, and Esther Greenglass. 2015. “The Inter-Item Standard Deviation (ISD): An Index That Discriminates between Conscientious and Random Responders.” Personality and Individual Differences 84:79–83. doi: 10.1016/j.paid.2014.08.021.

See Also

`resp_styles()` for calculating response style indicators. `resp_nondifferentiation()` for calculating response nondifferentiation indicators.

Examples

```
# A small test data set with ten respondents
# and responses to three survey questions
# with response scales from 1 to 5.
testdata <- data.frame(
  var_a = c(1,4,3,5,3,2,3,1,3,NA),
  var_b = c(2,5,2,3,4,1,NA,2,NA,NA),
  var_c = c(1,2,3,NA,3,4,4,5,NA,NA))

# Calculate response distribution indicators
resp_distributions(x = testdata) |>
    round(2)

# Include respondents with NA values by decreasing the
# necessary number of valid responses per respondent.

resp_distributions(
      x = testdata,
      min_valid_responses = 0.2) |>
   round(2)
```

`resp_nondifferentiation` *Compute response nondifferentiation indicators*

Description

Compute response nondifferentiation indicators for responses to multi-item scales or matrix questions.

Usage

```
resp_nondifferentiation(x, min_valid_responses = 1, id = T)
```

Arguments

`x` A data frame containing survey responses in wide format. For more information see section "Data requirements" below.

`min_valid_responses` Numeric between 0 and 1 of length 1. Defines the share of valid responses a respondent must have to calculate response quality indicators. Default is 1.

`id` default is `True`. If the default value is supplied a column named `id` with integer ids will be created. If `False` is supplied, no id column will be created. Alternatively, a numeric or character vector of unique values identifying each respondent can be supplied. Needs to be of the same length as the number of rows of `x`.

Details

Response nondifferentiation is the result of response behavior in which respondents deviate from an ideal response process. Optimal response behavior is termed optimizing, while deviations from optimal response behavior are termed satisficing (Krosnik, 1991). Optimizing describes a behavior in which respondents go through all steps of comprehension, retrieval, judgment, and response selection. When satisficing, respondents skip all or parts of the optimal response process. Satisficing can lead to non-response, "don't know" responses, random responding or nondifferentiation. The later is targeted by the function `resp_nondifferentiation()`.

Nondifferentiation is characterized by respondents choosing similar or even the same response options regardless of the content of the question. Multiple indicators for response nondifferentiation have been developed. For `resp_nondifferentiation()`, the following response nondifferentiation indicators described by Kim et al. (2017) are calculated per respondent:

- Simple Nondifferentiation: Respondents are assigned 1 or 0 depending on whether all responses have the same value (1) or not (0).
- Mean Root of Pairs Method: Mean of the root of the absolute differences between all pairs in a multi-item scale or matrix questions. It ranges from 0 (least straightlining) to 1 (most straightlining). The indicator is rescaled to be inbetween the minimum and maximum of all values. This means that including/excluding responses or respondents into the calculation changes the indicators values.
- Maximum Identical Rating Method: Proportion of the most commonly selected response option among all responses in a multi-item scale or matrix questions. It ranges from 0 (least straightlining) to 1 (most straightlining).
- Scale Point Variation Method: The probability of differentiation is defined as $1 - \Sigma p_i^2$, where p_i is the proportion of the values rated at each scale point on a rating scale and i indicates the number of scale points. The measure becomes larger if respondents use more scales points in a multi-item scale or matrix questions.

It should be noted that Kim et al. (2017) average the response nondifferentiation indicators to obtain an aggregate measure for response nondifferentiation. To do so, the `summary()` function can be called on the results of `resp_nondifferentiation()`. Additionally, Kim et al. (2017) removed all respondents with missing values from their study. For `resp_nondifferentiation()` this is the default behavior (`min_valid_responses = 1`). Reducing the value of `min_valid_responses` can lead to problems. For example, respondents with less valid respones will have less of an opportunity to use all response options which in turn is used to calculate the Scale Point Variation Method indicator. Thus, consider whether allowing missing responses impacts the results indicators and subsequent analyses.

Value

Returns a data frame with response nondifferentiation indicators per respondent. Dimensions:

- Rows: Equal to number of rows in x.
- Columns: Four response nondifferentiation indicator columns + id column (if specified).

Data requirements

`resp_nondifferentiationf()` assumes that the input data frame is structured in the following way:

- The data frame is in wide format, meaning each row represents one respondent, each column represents one variable.
- The variables are in same the order as the questions respondents saw while taking the survey.
- Reverse keyed variables are in their original form. No items were recoded.
- All responses have integer values.
- Questions have the same number of response options.
- Missing values are set to NA.

Author(s)

Matthias Roth

References

Kim, Yujin, Jennifer Dykema, John Stevenson, Penny Black, and D. Paul Moberg. 2019. "Straightlining: Overview of Measurement, Comparison of Indicators, and Effects in Mail–Web Mixed-Mode Surveys." Social Science Computer Review 37(2):214–33. doi: 10.1177/0894439317752406.

Krosnick, Jon A. 1991. "Response Strategies for Coping with the Cognitive Demands of Attitude Measures in Surveys." Applied Cognitive Psychology 5(3):213–36. doi: 10.1002/acp.2350050305.

See Also

resp_styles() for calculating response style indicators. resp_distributions() for calculating response distribution indicators.

Examples

```
# A small test data set with ten respondents
# and responses to three survey questions
# with response scales from 1 to 5.
testdata <- data.frame(
  var_a = c(1,4,3,5,3,2,3,1,3,NA),
  var_b = c(2,5,2,3,4,1,NA,2,NA,NA),
  var_c = c(1,2,3,NA,3,4,4,5,NA,NA))

# Calculate response nondifferentiation indicators
resp_nondifferentiation(x = testdata) |>
    round(2)

# Include respondents with NA values by decreasing the
# necessary number of valid responses per respondent.

resp_nondifferentiation(
      x = testdata,
      min_valid_responses = 0.2) |>
   round(2)

resp_nondifferentiation(
     x = testdata,
```

```
    min_valid_responses = 0.2) |>
  summary() # To obtain aggregate measures of response nondifferentiation
```

---

resp_patterns *Compute response pattern indicators*

Description

Compute response pattern indicators for responses to multi-item scales or matrix questions.

Usage

```
resp_patterns(
  x,
  min_valid_responses = 1,
  defined_patterns,
  arbitrary_patterns,
  min_repetitions = 2,
  id = T
)
```

Arguments

`x` A data frame containing survey responses in wide format. For more information see section "Data requirements" below.

`min_valid_responses` Numeric between 0 and 1 of length 1. Defines the share of valid responses a respondent must have to calculate response pattern indicators. Default is 1.

`defined_patterns` An optional vector of integer values with patterns to search for or a list of integer vectors. Will not be computed if not specified or if an empty vector is supplied.

`arbitrary_patterns` An optional vector of integer values or a list containing vectors of integer values. The values determine the pattern that should be searched for. Will not be computed if not specified or if 0 is supplied.

`min_repetitions` Defines number of times an arbitrary pattern has to be repeated to be retained in the results. Must be larger or equal to 2.

`id` default is `True`. If the default value is supplied a column named `id` with integer ids will be created. If `False` is supplied, no id column will be created. Alternatively, a numeric or character vector of unique values identifying each respondent can be supplied. Needs to be of the same length as the number of rows of `x`.

Details

The following response distribution indicators are calculated per respondent:

- n_transitions: Number of times two consecutive response options differ.
- mean_string_length: Mean length of strings of identical answers.
- longest_string_length: Longest length of string of identical answers.
- (optional) defined_pattern: A list column that contains one named vector per respondent. The names of the vector are repeating patterns found in the responses of a respondent. The values of the vector are how often the pattern specified in the argument "defined_patterns" occurs. See section "Defined patterns" for more information.
- (optional) arbitrary_patterns: A list column that contains one named vector per respondent. The names of the vector are repeating patterns found in the responses of a respondent. The values of the vector are how often the pattern occurred. See "Arbitrary patterns" for more information.

Value

Returns a data frame with response quality indicators per respondent. Dimensions:

- Rows: Equal to number of rows in x.
- Columns: Three response pattern indicators + one column for defined patterns (if specified) + one column for arbitrary patterns (if specified) + one id column (if specified).

Defined and arbitrary pattern indicators

Responses of an individual respondent can follow patterns, such as zig-zagging across the response scale over multiple items. There might be a-priori knowledge which response patterns could occur and might be indicative of low quality responding. For this case the defined_patterns argument can be used to specify one or more patterns whose presence will be checked for each respondent. If no a-priori knowledge exists, it is possible to check for all patterns of a specified length.

Defined patterns:

A pattern is defined by providing one ore more patterns in a character vector. A few examples: `resp_patterns(x,defined_patterns = c(1,2,3)` checks how often the response pattern 1,2,3 occurs in the responses of a single respondent. `list(c(1,2,3),c(3,2,1))` checks how often the two patterns 1,2,3 and 3,2,1 occur individually in the responses of a single respondent. There is no limit to the number of patterns.

Arbitrary patterns:

Checks for arbitrary patterns are defined by providing one ore more integer values in a numeric vector. The integers must be larger or equal to two. A few examples: `resp_patterns(x,arbitrary_patterns = 2)` will check for sequences of responses of length two which repeat at least two times. `resp_patterns(x,arbitrary_pa = c(2,3,4,5))` will check for sequences of responses of length two, three, four and five that repeat at least two times.

Data requirements

resp_patterns() assumes that the input data frame is structured in the following way:

- The data frame is in wide format, meaning each row represents one respondent, each column represents one variable.
- The variables are in same the order as the questions respondents saw while taking the survey.
- Reverse keyed variables are in their original form. No items were recoded.
- All responses have integer values.
- Questions have the same number of response options.
- Missing values are set to NA.

Author(s)

Matthias Roth, Thomas Knopf

References

Curran, P. G. (2016). Methods for the detection of carelessly invalid responses in survey data. Journal of Experimental Social Psychology, 66, 4–19. https://doi.org/10.1016/j.jesp.2015.07.006

See Also

resp_styles() for calculating response style indicators. resp_distributions() for calculating response distribution indicators. resp_nondifferentiation() for calculating response nondifferentiation indicators.

Examples

```
# A small test data set with ten respondents
# and responses to three survey questions
# with response scales from 1 to 5.
testdata <- data.frame(
  var_a = c(1,4,3,5,3,2,3,1,3,NA),
  var_b = c(2,5,2,3,4,1,NA,2,NA,NA),
  var_c = c(1,2,3,NA,3,4,4,5,NA,NA))

# Calculate response pattern indicators
resp_patterns(x = testdata) |>
    round(2)

# Include respondents with NA values by decreasing the
# necessary number of valid responses per respondent.

resp_patterns(
      x = testdata,
      min_valid_responses = 0.2) |>
   round(2)
```

resp_styles — *Compute response style indicators*

Description

Calculates response style indicators for matrix questions or multi-item scales.

Usage

```
resp_styles(
  x,
  scale_min,
  scale_max,
  min_valid_responses = 1,
  normalize = TRUE,
  id = T
)
```

Arguments

x	A data frame containing survey responses in wide format. For more information see section "Data requirements" below.
scale_min	Numeric of length 1. Minimum of scale provided.
scale_max	Numeric of length 1. Maximum of scale provided.
min_valid_responses	Numeric between 0 and 1 of length 1. Defines the share of valid responses a respondent must have to calculate response style indicators.
normalize	logical of length 1. If *TRUE*, counts of response style indicators will be divided by the number of non-missing responses per respondent. Default is *TRUE*.
id	default is `True`. If the default value is supplied a column named `id` with integer ids will be created. If `False` is supplied, no id column will be created. Alternatively, a numeric or character vector of unique values identifying each respondent can be supplied. Needs to be of the same length as the number of rows of `x`.

Details

Response styles capture systematic shifts in respondents response behavior. `resp_styles()` is aimed at multi-item scales or matrix questions which use the same number of response options for many questions.

The following response style indicators are calculated per respondent: Middle response style (MRS), acquiescence response style (ARS), disacquiescence response style (DRS), extreme response style (ERS) and non-extreme response style (NERS).

The response style indicators are calculated in the following way

- MRS: Sum of mid point responses.

- ARS: Sum of responses larger than midpoint.
- DRS: Sum of responses lower than midpoint.
- ERS: Sum of lowest or highest category responses.
- NERS: Sum of responses between lowest and highest response category.

Note that ARS and DRS assume that the polarity of the scale is positive. This means that higher numerical values indicate agreement and lower numerical values indicate disagreement. MRS can only be calculated if the scale has a numeric midpoint.

Also note that the response style literature is fragmented (Bhaktha et al., 2024). Response styles calculated with `resp_styles()` are based on van Vaerenbergh & Thomas (2024). However, we used the name non-extreme response style (NERS) instead of mild response style, to emphasize that NERS it the inverse of ERS. Both appear in the literature (for a NERS example see Wetzel et al. (2013)). Consult literature in your field of research to find appropriate names for the response style indicators calculated here.

Value

Returns a data frame with response style indicators per respondent.

- Rows: Equal to number of rows in x.
- Columns: Five for each response style indicator + id column (if specified).

Data requirements

`resp_styles()` assumes that the input data frame is structured in the following way:

- The data frame is in wide format, meaning each row represents one respondent, each column represents one variable.
- The variables are in same the order as the questions respondents saw while taking the survey.
- Reverse keyed variables are in their original form. No items were recoded.
- All responses have integer values.
- Questions have the same number of response options.
- Missing values are set to `NA`.

Author(s)

Matthias Roth, Matthias Bluemke & Clemens Lechner

References

Bhaktha, Nivedita, Henning Silber, and Clemens Lechner. 2024. „Characterizing response quality in surveys with multi-item scales: A unified framework". OSF-preprtint: https://osf.io/9gs67/

van Vaerenbergh, Y., and T. D. Thomas. 2013. „Response Styles in Survey Research: A Literature Review of Antecedents, Consequences, and Remedies". International Journal of Public Opinion Research 25(2):195–217. doi: 10.1093/ijpor/eds021.

Wetzel, Eunike, Claus H. Carstensen, und Jan R. Böhnke. 2013. „Consistency of Extreme Response Style and Non-Extreme Response Style across Traits". Journal of Research in Personality 47(2):178–89. doi: 10.1016/j.jrp.2012.10.010.

See Also

resp_distributions() for calculating response distribution indicators. resp_nondifferentiation() for calculating response nondifferentiation indicators. resp_patterns() for calculating response pattern indicators.

Examples

```
# A test data set with ten respondents
# and responses to three survey questions
# with response scales from 1 to 5.
testdata <- data.frame(
  var_a = c(1,4,3,5,3,2,3,1,3,NA),
  var_b = c(2,5,2,3,4,1,NA,2,NA,NA),
  var_c = c(1,2,3,NA,3,4,4,5,NA,NA))

# Calculate response distribution indicators
resp_styles(testdata,
            scale_min = 1,
            scale_max = 5) |>
   round(2) # round to second decimal

# Include respondents with NA values by decreasing the
# necessary number of valid responses per respondent.
resp_styles(testdata,
            scale_min = 1,
            scale_max = 5,
            min_valid_responses = 0.2) |>
   round(2) # round to second decimal

# Get counts of responses attributable to response styles.
resp_styles(testdata,
            scale_min = 1,
            scale_max = 5,
            normalize = FALSE)
```

---

summary.flag_resp *Summary function for flag_resp() output*

Description

Calculates the number of respondents flagged with a flagging strategy. Also calculates the agreement between flagging strategies.

Usage

```
## S3 method for class 'flag_resp'
summary(object, normalize = F, ...)
```

Arguments

`object`	An object of type `flag_resp` which is created using the `flag_resp()` function.
`normalize`	A logical value indicating, whether to normalize the agreement estimates between flagging strategies. See details for more information.
`...`	Other arguments for summary functions (currently not supported).

Details

The agreement is either the count of respondents which two flagging strategies flag (`normalize = T`) or the number of respondents that is flagged positive by at least one flagging strategy.

In logical terms, the normalized agreement is `sum(fs1 & fs2) / sum(fs1 | fs2)`.

Value

An object of class "summary_flag_resp". The object works like a list with four elements.

- n_flagged: a named vector of the number of cases a flagging strategy flagged as positive.
- agreement: a data frame which counts the number of cases two flagging strategies flagged as positive. If `normalized`, the values are the percentage agreement in flagged respondents.
- normalized: Indicator if agreement values were normalized.
- n: number of rows in `object`.

Examples

```
resp_distributions(nep) |>
  flag_resp(ii_mean > 3,
   ii_sd > 1,
   ii_mean > 3 & ii_sd > 1) |>
  summary()
```

---

summary.resp_indicator

Summary function for resp_indicator objects

Description

Summarizes results of resp_* functions.

Usage

```
## S3 method for class 'resp_indicator'
summary(object, quantiles, ...)
```

Arguments

`object`	An object of type resp_indicator created with a resp_* function.
`quantiles`	A numeric vector with values raning from 0 to 1. Determines the quantiles which are calculated. Default is `c(0,0.25,0.5,0.75,1)`.
`...`	Additional arguments (currently not supported).

Value

A resp_indicator summary object. Works like a list with two elements:

- `quantile_estimates`. A dataframe of estimated quantiles for the response quality indicators calculated.
- `mean_estimates`. A named vector with means of response quality indicators calculated.

Examples

```
resp_distributions(nep) |> summary()
```

Index