Package 'cata'

February 17, 2025

Type Package

Title Analysis of Check-All-that-Apply (CATA) Data

Version 0.1.0.26

Date 2025-02-17

Author J.C. Castura [aut, cre, ctb] (<https://orcid.org/0000-0002-1640-833X>)

Maintainer J.C. Castura <jcastura@compusense.com>

Description Package contains functions for analyzing check-all-that-apply (CATA) data from consumer and sensory tests. Cochran's Q test, McNemar's test, and Penalty-Lift analysis are provided; for details, see Meyners, Castura & Carr (2013) <doi:10.1016/j.foodqual.2013.06.010>. Cluster analysis can be performed using b-cluster analysis, then evaluated using various measures; for details, see Castura, Meyners, Varela & Næs (2022) <doi:10.1016/j.foodqual.2022.104564>. Methods are adapted to cluster consumers based on their product-related hedonic responses; for details, see Castura, Meyners, Pohjanheimo, Varela & Næs (2023) <doi:10.1111/joss.12860>.

URL https://CRAN.R-project.org/package=cata

Depends R (>= 4.4.0)

Imports stats, utils, graphics

License GPL (>= 2)

LazyData TRUE

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Date/Publication 2025-02-17 22:40:02 UTC

Contents

`ARI` *Adjusted Rand index*

Description

Calculate the adjusted Rand index (ARI) between two sets of cluster memberships.

Usage

```
ARI(x, y, signif = FALSE, n = 1000)
```

Arguments

`x`	vector of cluster memberships (integers)
`y`	vector of cluster memberships (integers)
`signif`	conduct significance test; default is `FALSE`
`n`	number of replicates in Monte Carlo significance test

Value

list of the following:

- `ari` adjusted Rand index
- `nari` normalized adjusted Rand index
- `sim.mean` average value of null distribution (should be closed to zero)
- `sim.var` variance of null distribution
- `p.value` P value of observed ARI (or NARI) value

References

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2, 193–218. doi:10.1007/BF01908075.

Qannari, E.M., Courcoux, P., & Faye, P. (2014). Significance test of the adjusted Rand index. Application to the free sorting task. *Food Quality and Preference*, 32, 93-97. doi:10.1016/j.foodqual.2013.05.005.

Examples

```
x <- sample(1:3, 20, replace = TRUE)
y <- sample(1:3, 20, replace = TRUE)

ARI(x, y, signif = FALSE)
```

---

barray *Convert 3d array of CATA data to 4d array of CATA differences*

Description

Converts a three-dimensional array (I assessors, J products, M attributes) to a four-dimensional array of product comparisons (I assessors, $J(J-1)/2$ product comparisons, two outcomes (of type `b` or `c`), M attributes)

Usage

```
barray(X, values = "bc", type.in = "binary", type.out = "binary")
```

Arguments

`X`	three-dimensional array (I assessors, J products, M attributes) where values are `0` (not checked) or `1` (checked)
`values`	`"bc"` (default) returns two outcomes: `b` and `c`; otherwise `"abcd"` returns four outcomes: `a`, `b`, `c`, `d`.
`type.in`	type of data submitted; default (`binary`) may be set to `ordinal` or `scale`.
`type.out`	currently only `binary` is implemented

Value

A four-dimensional array of product comparisons having I assessors, $J(J-1)/2$ product comparisons, outcomes (see `values` parameter), M attributes

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
data(bread)

# Get the 4d array of CATA differences for the first 8 consumers
b <- barray(bread$cata[1:8,,])
```

`bcluster` *Wrapper function for b-cluster analysis*

Description

By default, `bcluster` calls a function to perform b-cluster analysis by a non-hierarchical iterative ascent algorithm, then inspects results if there are multiple runs.

Usage

```
bcluster(X, inspect = TRUE, inspect.plot = TRUE, algorithm = "n",
measure = "b", G = NULL, M = NULL, max.iter = 500, X.input = "data",
tol = exp(-32), runs = 1, seed = 2021)
```

Arguments

`X`	three-way array with I assessors, J products, M attributes where CATA data have values `0` (not checked) and `1` (checked)
`inspect`	default (`TRUE`) calls the `inspect` function to evaluate all solutions (when `runs>1`)
`inspect.plot`	default (`TRUE`) plots results from the `inspect` function
`algorithm`	default is `n` for non-hierarchical; `h` for hierarchical
`measure`	default is `b` for the b-measure; `Q` for Cochran's Q test
`G`	number of clusters (required for non-hierarchical algorithm)
`M`	initial cluster memberships
`max.iter`	maximum number of iteration allowed (default `500`)
`X.input`	available only for non-hierarchical algorithm; its value is either `"data"` (default) or `"bc"` if `X` is obtained from the function `barray`
`tol`	non-hierarchical algorithm stops if variance over 5 iterations is less than `tol` (default: `exp(-32)`)
`runs`	number of runs (defaults to `1`)
`seed`	for reproducibility (default is `2021`)

Value

list with elements:

- `runs` : b-cluster analysis results from `bcluster.n` or `bcluster.h` (in a list if `runs>1`)
- `inspect` : result from `inspect` (the plot from this function is rendered if `inspect.plot` is `TRUE`)

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
data(bread)

# b-cluster analysis on the first 8 consumers and the first 5 attributes
(b1 <- bcluster(bread$cata[1:8,,1:5], G=2, seed = 123))
# Since the seed is the same, the result will be identical to
# (b2 <- bcluster.n(bread$cata[1:8,,1:5], G=2, seed = 123))
b3 <- bcluster(bread$cata[1:8,,1:5], G=2, runs = 5, seed = 123)
```

---

bcluster.h *b-cluster analysis by hierarchical agglomerative strategy*

Description

Perform b-clustering using the hierarchical agglomerative clustering strategy.

Usage

```
bcluster.h(X, measure = "b", runs = 1, seed = 2021)
```

Arguments

`X`	three-way array; the $I \times J \times M$ array has I assessors, J products, M attributes where CATA data have values `0` (not checked) and `1` (checked)
`measure`	currently only `b` (the b-measure) is implemented
`runs`	number of runs (defaults to `1`; use a higher number of runs for a real application)
`seed`	for reproducibility (default is `2021`)

Value

An object of class `hclust` from hierarchical b-cluster analysis results (a list of such objects if `runs>1`), where each `hclust` object has the structure described in `hclust` as well as the item `retainedB` (a vector indicating the retained sensory differentiation at each iteration (merger)).

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
data(bread)

# hierarchical b-cluster analysis on first 8 consumers and first 5 attributes
b <- bcluster.h(bread$cata[1:8,,1:5])

plot(as.dendrogram(b),
  main = "Hierarchical b-cluster analysis",
  sub = "8 bread consumers on 5 attributes")
```

---

bcluster.n *b-cluster analysis by non-hierarchical iterative ascent clustering strategy*

Description

Non-hierarchical b-cluster analysis transfers assessors iteratively to reach a local maximum in sensory differentiation retained.

Usage

```
bcluster.n(X, G, M = NULL, measure = "b", max.iter = 500, runs = 1,
X.input = "data", tol = exp(-32), seed = 2021)
```

Arguments

X	CATA data organized in a three-way array (assessors, products, attributes)
G	number of clusters (required for non-hierarchical algorithm)
M	initial cluster memberships (default: NULL), but can be a vector (one run) or a matrix (consumers in rows; runs in columns)
measure	b (default) for the b-measure is implemented
max.iter	maximum number of iteration allowed (default 500)
runs	number of runs (defaults to 1)
X.input	either "data" (default) or "bc" if X is obtained from the function barray
tol	algorithm stops if variance over 5 iterations is less than tol (default: exp(-32))
seed	for reproducibility (default is 2021)

Value

An object of class bclust.n (or a list of such objects if runs>1), where each such object has the following components:

- cluster : vector of the final cluster memberships
- totalB : value of the total sensory differentiation in data set
- retainedB : value of sensory differentiation retained in b-cluster analysis solution

- `progression` : vector of sensory differentiation retained in each iteration
- `iter` : number of iterations completed
- `finished` : boolean indicates whether the algorithm converged before `max.iter`

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
data(bread)

# b-cluster analysis on the first 8 consumers and the first 5 attributes
(b <- bcluster.n(bread$cata[1:8, , 1:5], G=2))
```

`cochranQ` *Cochran's Q test*

Description

Conduct Cochran's Q test assuming equal columns proportions for matched binary responses versus the alternative hypothesis of unequal column proportions.

Usage

```
cochranQ(X, quiet = FALSE, digits = getOption("digits"))
```

Arguments

`X`	matrix of I assessors (rows) and J products (columns) where values are `0` (not checked) or `1` (checked)
`quiet`	if `FALSE` (default) then it prints information related to the test; if `TRUE` it returns only the test statistic (`Q`)
`digits`	for rounding

Details

Method returns test statistic, degrees of freedom, and p value from Cochran's Q test.

Value

Cochran's Q test results (statistic, degrees of freedom, p-value)

References

Cochran, W.G. (1950). The comparison of percentages in matched samples. *Biometrika*, 37, 256-266, doi:10.2307/2332378

Meyners, M., Castura, J.C., & Carr, B.T. (2013). Existing and new approaches for the analysis of CATA data. *Food Quality and Preference*, 30, 309-319, doi:10.1016/j.foodqual.2013.06.010

See Also

mcnemarQ

Examples

```
data(bread)

# Cochran's Q test on the first 50 consumers on the first attribute ("Fresh")
cochranQ(bread$cata[1:50, , 1], digits=3)

# Same, returning only test statistics for the first 4 attributes
t(res <- apply(bread$cata[1:50, , 1:4], 3, cochranQ, quiet=TRUE, digits=3))
```

---

code.topk *Apply top-k box coding to scale data*

Description

Apply top-k box coding to scale data. Using defaults give top-2 box (T2B) coding.

Usage

```
code.topk(X, zero.below = 8, one.above = 7)
```

Arguments

X	input matrix
zero.below	default is 8; values below this numeric threshold will be coded 0; use NULL if there is no such threshold
one.above	default is 7; values above this numeric threshold will be coded 1; use NULL if there is no such threshold

Value

matrix X with top-k coding applied

References

Castura, J.C., Meyners, M., Pohjanheimo, T., Varela, P., & Næs, T. (2023). An approach for clustering consumers by their top-box and top-choice responses. *Journal of Sensory Studies*, e12860. doi:10.1111/joss.12860

Examples

```
# Generate some data
set.seed(123)
X <- matrix(sample(1:9, 100, replace = TRUE), nrow = 5)

# apply top-2 box (T2B) coding
code.topk(X, zero.below = 8, one.above = 7)
```

`Consumer CATA data set: bread`

Consumer CATA data set: bread

Description

Raw results from CATA and Liking evaluations of six bread products samples by 161 consumers.

Format

A list with 4 items:

- `$cata` : check-all-that-apply (CATA) data (array, 161 consumers x 6 breads x 31 sensory attributes)
- `$liking` : 9-point hedonic scale data (matrix, 161 consumers x 6 breads)
- `$ideal.cata` : check-all-that-apply (CATA) data for ideal bread (matrix, 161 consumers x 31 sensory attributes)
- `$liking` : 9-point hedonic scale data for ideal bread(vector, 161 consumers)

CATA data is coded `1` if the attribute is checked; otherwise it is coded `0`

References

Meyners, M., Castura, J.C., & Carr, B.T. (2013). Existing and new approaches for the analysis of CATA data. *Food Quality and Preference*, 30, 309-319, doi:10.1016/j.foodqual.2013.06.010

Examples

```
data(bread)
head(bread$cata)
```

`evaluateClusterQuality` *Evaluate Quality of Cluster Analysis Solution*

Description

Evaluate the quality of cluster analysis solutions using measures related to within-cluster product discrimination, between-cluster non-redundancy, overall diversity (coverage), average RV, sensory differentiation retained, and within-cluster homogeneity.

Usage

```
evaluateClusterQuality(X, M, alpha = .05, M.order = NULL,
quiet = FALSE, digits = getOption("digits"), ...)
```

Arguments

`X`	three-way array; the $I \times J \times M$ array has I assessors, J products, M attributes where CATA data have values `0` (not checked) and `1` (checked)
`M`	cluster memberships
`alpha`	significance level for two-tailed tests (default: $\alpha = 0.05$)
`M.order`	can be used to change the cluster numbers (e.g. to label cluster 1 as cluster 2 and vice versa); defaults to `NULL`
`quiet`	if `FALSE` (default) then it prints information quality measures; if `TRUE` then returns results without printing
`digits`	significant digits (to display)
`...`	other parameters for `print.default` (if `quiet = TRUE`).

Value

A list containing cluster analysis quality measures:

- `$solution` :
  - `Pct.b` = percentage of the total sensory differentiation retained in the solution
  - `min(NR)` = smallest observed between-cluster non-redundancy
  - `Div_G` = overall diversity (coverage)
  - `H_G` = overall homogeneity (weighted average of within-cluster homogeneity indices)
  - `avRV` = average RV coefficient for all between-cluster comparisons
- `$clusters` :
  - `ng` = number of cluster members
  - `bg` = sensory differentiation retained in cluster
  - `xbarg` = average citation rate in cluster
  - `Hg` = homogeneity index within cluster (see `homogeneity`)

  - Dg = within-cluster product discrimination
- $nonredundancy.clusterpairs :
  - square data frame showing non-redundancy for each pair of clusters (low values indicate high redundancy)
- $rv.clusterpairs :
  - square data frame with RV coefficient for each pair of clusters (high values indicate higher similarity in product configurations)

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

See Also

homogeneity

Examples

```
data(bread)
evaluateClusterQuality(bread$cata[1:8,,1:5], M = rep(1:2, each = 4))
```

---

getb *Calculate the b-measure*

Description

Function to calculate the b-measure, which quantifies the sensory differentiation retained.

Usage

```
getb(X.b, X.c, oneI = FALSE, oneM = FALSE)
```

Arguments

X.b	three-way ($I \times J(J-1)/2 \times M$) array with I assessors, $J(J-1)/2$ product comparisons, M CATA attributes, where values are counts of types b and c from the function barray)
X.c	array of same dimension as X.b, where values are counts of types b and c from the function barray)
oneI	indicates whether calculation is for one assessor (default: FALSE)
oneM	indicates whether calculation is for one attribute (default: FALSE)

Value

b-measure

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
data(bread)

bread.bc <- barray(bread$cata[1:8,,1:5])
getb(bread.bc[,,1,], bread.bc[,,2,])
```

`homogeneity` *Calculate within-cluster homogeneity*

Description

Within a group of `N` consumers, the Homogeneity index lies between `1/N` (no homogeneity) to `1` (perfect homogeneity).

Usage

```
homogeneity(X, oneI = FALSE, oneM = FALSE)
```

Arguments

`X`	three-way array; the $I \times J \times M$ array has I assessors, J products, M attributes where CATA data have values `0` (not checked) and `1` (checked)
`oneI`	indicates whether calculation is for one assessor (default: `FALSE`)
`oneM`	indicates whether calculation is for one attribute (default: `FALSE`)

Value

homogeneity index

References

Llobell, F., Cariou, V., Vigneau, E., Labenne, A., & Qannari, E.M. (2019). A new approach for the analysis of data and the clustering of subjects in a CATA experiment. *Food Quality and Preference*, 72, 31-39, doi:10.1016/j.foodqual.2018.09.006

Examples

```
data(bread)

# homogeneity index for the first 7 consumers on the first 6 attributes
homogeneity(bread$cata[1:7,,1:6])
```

`inspect` *Inspect/summarize many b-cluster analysis runs*

Description

Inspect many runs of b-cluster analysis. Calculate sensory differentiation retained and recurrence rate.

Usage

```
inspect(X, G = 2, bestB = NULL, bestM = NULL, inspect.plot = TRUE)
```

Arguments

`X`	list of multiple runs of b-cluster analysis results from `bcluster.n` or `bcluster.h`
`G`	number of clusters (required for non-hierarchical algorithm)
`bestB`	total sensory differentiation retained in the best solution. If not provided, then `bestB` is determined from best solution in the runs provided (in `X`).
`bestM`	cluster memberships for best solution. If not provided, then the best solution is determined from the runs provided (in `X`).
`inspect.plot`	default (`TRUE`) plots results from the `inspect` function

Value

A data frame with unique solutions in rows and the following columns:

- `B` : Sensory differentiation retained
- `PctB` : Percentage of the total sensory differentiation retained
- `B.prop` : Proportion of sensory differentiation retained compared to best solution
- `Raw.agree` : raw agreement with best solution
- `Count` : number of runs for which this solution was observed
- `Index` : list index (i.e., run number) of first solution solution in `X` corresponding to this row

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
data(bread)

res <- bcluster.n(bread$cata[1:8, , 1:5], G = 3, runs = 3)
(ires <- inspect(res))
# get index of solution retaining the most sensory differentiation (in these runs)
indx <- ires$Index[1]
# cluster memberships for solution of this solution
res[[indx]]$cluster
```

`mad.dist` *MAD distances between objects*

Description

Computes and returns inter-object median of absolute deviations (MADs) based differences.

Usage

```
mad.dist(X)
```

Arguments

`X`	objects-by-terms matrix

Value

object of class `dist` giving inter-object MAD distances

References

One citation, one vote! A new approach to the analysis of check-all-that-apply (CATA) data in sensometrics based on L1 norm methods.

Examples

```
data(bread)
CATA.freq <- apply(bread$cata, 2:3, sum)
# median-center columns (attributes)
CATA.swept <- sweep(CATA.freq, 2, apply(CATA.freq, 2, median))
# cluster analysis of products using complete linkage
dist.Products <- mad.dist(CATA.swept)
plot(as.dendrogram(hclust(dist.Products, method = "complete")), main = "Product clusters")
# cluster analysis of attributes using complete linkage
dist.Att <- mad.dist(t(CATA.swept))
plot(as.dendrogram(hclust(dist.Att, method = "complete")), main = "Attribute clusters")
```

madperm *Permutation tests for CATA data*

Description

Permutation tests for check-all-that-apply (CATA) data following the 'one citation, one vote' principle. Returns CATA frequency and percentage tables per condition and permutation test results specified.

Usage

```
madperm(X, B = 99, seed = .Random.seed, tests = 1:5, alpha = 0.05, control.fdr = FALSE,
verbose = FALSE)
```

Arguments

X	a three-way (or four-way) array with I assessors, (R conditions/timepoints,) P products, T terms with data valued 0 (not checked) and 1 (checked)
B	permutations in null distribution; ensure B is larger than the default value (99) when conducting real analyses
seed	specify a numeric seed for reproducibility; if not provided, a random seed is generated
tests	numeric vector specifying which tests to conduct; default 1:5 (all tests): (1) multivariate (global) test; (2) univariate tests; (3) elementwise tests; (4) pairwise multivariate tests; (5) pairwise univariate tests
alpha	Type I error rate (default: $\alpha = 0.05$)
control.fdr	control False Discovery Rate (using Benjamini-Hochberg (BH) step-up procedure)? (default: FALSE)
verbose	return null distribution(s) and function call? (default: FALSE)

Value

list, one per condition:

- CATA.table : table of CATA citation percentages ($P \times T$)
- CATA.freq : CATA frequency table ($P \times T$)
- Permutation test results specified by the tests parameter
  1. Global.Results : list of multivariate (global) results
  2. Univariate.Results : list of T univariate results
  3. Elementwise.Results : list of PT elementwise results
  4. Multivariate.Paired.Results : list of $P(P-1)/2$ multivariate paired results
  5. Univariate.Paired.Results : list of $P(P-1)T/2$ univariate paired results

also, if verbose is TRUE:

- Null.Dist list of null distributions for tests specified
- Call : madperm function call

References

One citation, one vote! A new approach to the analysis of check-all-that-apply (CATA) data in sensometrics based on L1 norm methods.

Examples

```
data(bread)
# add product names
X <- bread$cata[1:100,,1:5]
dimnames(X)[[2]] <- paste0("P", dimnames(X)[[2]])
# permutation tests for the first 100 consumers and 5 attributes
# will be run with default parameter values for illustrative purposes only
res <- madperm(X, B = 99, seed = 123)
print(res) # inspect results
```

`mcnemarQ` *McNemar's test*

Description

Pairwise tests are conducted using the two-tailed binomial test. These tests can be conducted after Cochran's Q test.

Usage

```
mcnemarQ(X, quiet = FALSE, digits = getOption("digits"))
```

Arguments

`X`	matrix of I assessors (rows) and J products (columns) where values are `0` (not checked) or `1` (checked)
`quiet`	if `FALSE` (default) then it prints information related to the test; if `TRUE` it returns only the test statistic (`Q`)
`digits`	for rounding

Value

Test results for all McNemar pairwise tests conducted via the binomial test

References

Cochran, W.G. (1950). The comparison of percentages in matched samples. *Biometrika*, 37, 256-266.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153-157.

Meyners, M., Castura, J.C., & Carr, B.T. (2013). Existing and new approaches for the analysis of CATA data. *Food Quality and Preference*, 30, 309-319, doi:10.1016/j.foodqual.2013.06.010

See Also

cochranQ

Examples

```
data(bread)

# McNemar's exact pairwise test for all product pairs
# on the first 50 consumers and the first attribute ("Fresh")
mcnemarQ(bread$cata[1:50, , 1])

# Same, returning only results for the first 4 attributes
(res <- apply(bread$cata[1:50, , 1:4], 3, mcnemarQ, quiet=TRUE, simplify=FALSE))
```

---

plift *Penalty-Lift Analysis*

Description

Penalty-Lift analysis for CATA variables, which is the difference between the average hedonic response when CATA attribute is checked vs. the average hedonic response when CATA attribute is not checked.

Usage

```
plift(X, Y, digits = getOption("digits"), verbose = FALSE)
```

Arguments

X	*either* a matrix of CATA data with I consumers (rows) and J products (columns) *or* an array of CATA data with I consumers, J products, and M attributes.
Y	matrix of hedonic data with I consumers (rows) and J products (columns)
digits	for rounding
verbose	set to TRUE to report counts and averages for checked and not checked conditions (default: FALSE)

Value

Penalty lift per attribute, with counts and averages if verbose is TRUE.

References

Meyners, M., Castura, J.C., & Carr, B.T. (2013). Existing and new approaches for the analysis of CATA data. *Food Quality and Preference*, 30, 309-319, doi:10.1016/j.foodqual.2013.06.010

Examples

```
data(bread)

# penalty lift, based only on the first 12 consumers

# for the first attribute ("Fresh")
plift(bread$cata[1:12,,1], bread$liking[1:12, ], digits = 3)

# for the first 3  attributes with counts and averages
plift(bread$cata[1:12,,1:3], bread$liking[1:12, ], digits = 3, verbose = TRUE)
```

`rv.coef` *Calculate RV Coefficient*

Description

Calculate RV coefficient

Usage

```
rv.coef(X, Y, method = 1)
```

Arguments

`X`	input matrix (same dimensions as `Y`)
`Y`	input matrix (same dimensions as `X`)
`method`	`1` (default) and `2` give identical RV coefficients

Value

RV coefficient

References

Robert, P., & Escoufier, Y. (1976). A unifying tool for linear multivariate statistical methods: the RV-coefficient. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 25, 257-265.

Examples

```
# Generate some data
set.seed(123)
X <- matrix(rnorm(8), nrow = 4)
Y <- matrix(rnorm(8), nrow = 4)

# get the RV coefficient
rv.coef(X, Y)
```

salton *Salton's cosine measure*

Description

Calculate Salton's cosine measure

Usage

```
salton(X, Y)
```

Arguments

X	input matrix (same dimensions as Y)
Y	input matrix (same dimensions as X)

Value

Salton's cosine measure

References

Salton, G., & McGill, M.J. (1983). *Introduction to Modern Information Retrieval*. Toronto: McGraw-Hill.

Examples

```
# Generate some data
set.seed(123)
X <- matrix(rnorm(8), nrow = 4)
Y <- matrix(rnorm(8), nrow = 4)

# get Salton's cosine measure
salton(X, Y)
```

selectionPlot *Plot variation in retained sensory differentiation*

Description

Plot variation in retained sensory differentiation of cluster memberships obtained from b-cluster analysis. This plot can be used to help the decision of how many clusters to retain.

Usage

```
selectionPlot(x, pctB = NULL, x.input = "deltaB", indx = NULL,
ylab = "change in B (K to G)", xlab = NULL)
```

Arguments

x	input vector which is either deltaB (default; change in sensory differentiation retained) or B (sensory differentiation retained) if x.input is "B"
pctB	vector of percentage of the total sensory differentiation retained
x.input	indicates what x is; either "deltaB" (default) or B.
indx	numeric value indicating which point(s) to emphasize
ylab	label shown on y axis and at selection point
xlab	label for points along x axis

References

Castura, J.C., Meyners, M., Varela, P., & Næs, T. (2022). Clustering consumers based on product discrimination in check-all-that-apply (CATA) data. *Food Quality and Preference*, 104564. doi:10.1016/j.foodqual.2022.104564.

Examples

```
set.seed(123)
G2 <- bcluster.n(bread$cata[1:8, , 1:5], G = 2, runs = 3)
G3 <- bcluster.n(bread$cata[1:8, , 1:5], G = 3, runs = 3)
G4 <- bcluster.n(bread$cata[1:8, , 1:5], G = 4, runs = 3)

best.indx <- c(which.max(unlist(lapply(G2, function(x) x$retainedB))),
               which.max(unlist(lapply(G3, function(x) x$retainedB))),
               which.max(unlist(lapply(G4, function(x) x$retainedB))))

G1.bc <- barray(bread$cata[1:8, , 1:5])
G1.B <- getb(G1.bc[,,1,], G1.bc[,,2,])
BpctB <- data.frame(retainedB = c(G1.B,
                                  G2[[best.indx[1]]]$retainedB,
                                  G3[[best.indx[2]]]$retainedB,
                                  G4[[best.indx[3]]]$retainedB))
BpctB$pctB <- 100*BpctB$retainedB / G2[[1]]$totalB
BpctB$deltaB <-
          c(100*(1-BpctB$retainedB[-nrow(BpctB)] / BpctB$retainedB[-1]), NA)
BpctB <- BpctB[-nrow(BpctB),]

opar <- par(no.readonly=TRUE)
par(mar = rep(5,4))
selectionPlot(BpctB$deltaB, BpctB$pctB, indx = 2)
par(opar)
```

---

toMatrix *Converts 3d array of CATA data to a tall 2d matrix format*

Description

Converts a three-dimensional array (I assessors, J products, M attributes) to a two-dimensional matrix with (I assessors, J products) rows and (M attributes) columns, optionally preceded by two columns of row headers.

Usage

```
toMatrix(X, header.rows = TRUE, oneI = FALSE, oneM = FALSE)
```

Arguments

X	three-dimensional array (I assessors, J products, M attributes) where values are 0 (not checked) or 1 (checked)
header.rows	TRUE (default) includes row headers; set to FALSE to exclude these headers
oneI	indicates whether calculation is for one assessor (default: FALSE)
oneM	indicates whether calculation is for one attribute (default: FALSE)

Value

A matrix with I assessors $\times J$ products in rows and M attributes in columns (preceded by 2 columns) of headers if header.rows = TRUE

Examples

```
data(bread)

# convert CATA results from the first 8 consumers and the first 4 attributes
# to a tall matrix
toMatrix(bread$cata[1:8,,1:4])
```

---

topc *Apply top-c choices coding to a vector of scale data from a respondent*

Description

Apply top-c choices coding to a vector of scale data from a respondent

Usage

```
topc(x, c = 2, coding = "B")
```

Arguments

x	input matrix
c	number of top choices considered to be 'success'; other choices are considered to be 'failure' and are coded 0
coding	"B" (default) codes all successes as 1; "N" codes all successes with their numeric coding

Value

matrix `X` with top-k coding applied

References

Castura, J.C., Meyners, M., Pohjanheimo, T., Varela, P., & Næs, T. (2023). An approach for clustering consumers by their top-box and top-choice responses. *Journal of Sensory Studies*, e12860. doi:10.1111/joss.12860

Examples

```
# Generate some data
set.seed(123)
X <- matrix(sample(1:9, 100, replace = TRUE), nrow = 5)

# apply top-2 choice (T2C) coding
apply(X, 1, topc)
```

`toWideMatrix` *Converts 3d array of CATA data to a wide 2d matrix format*

Description

Converts a three-dimensional array (I assessors, J products, M attributes) to a two-dimensional matrix (J products, (I assessors, M attributes))

Usage

```
toWideMatrix(X)
```

Arguments

`X`	three-dimensional array (I assessors, J products, M attributes) where values are `0` (not checked) or `1` (checked)

Value

A matrix with `J` products in rows and I assessors $\times M$ attributes in columns

Examples

```
data(bread)

# convert CATA results from the first 8 consumers and the first 4 attributes
# to a wide matrix
toWideMatrix(bread$cata[1:8,,1:4])
```

Index