Package: rAverage (via r-universe)

October 27, 2024

Version 0.5-8

Date 2017-07-29

Title Parameter Estimation for the Averaging Model of Information Integration Theory

Author Giulio Vidotto <giulio.vidotto@unipd.it>, Stefano Noventa <stefano.noventa@uni-tuebingen.de>, Davide Massidda <davide.massidda@gmail.com>, Marco Vicentini <marco.vicentini@gmail.com>

Maintainer Davide Massidda <davide.massidda@gmail.com>

Depends R (>= 2.8), methods, tcltk

Description Implementation of the R-Average method for parameter estimation of averaging models of the Anderson's Information Integration Theory by Vidotto, G., Massidda, D., & Noventa, S. (2010) <https://www.uv.es/psicologica/articulos3FM.10/3Vidotto.pdf>.

License GPL (>= 2)

NeedsCompilation yes

Date/Publication 2017-07-29 15:28:29 UTC

Repository https://davidemassidda.r-universe.dev

RemoteUrl https://github.com/cran/rAverage

RemoteRef HEAD

RemoteSha bf8554d97faabd573d370b9712bbd7b871745cb6

Contents

`AIC` *Information Criteria for averaging models*

Description

Functions to extract or recalculate the Akaike Information Criterion and the Bayesian Information Criterion of an averaging model fitted by the `rav` function.

Usage

```
AIC(object, ..., k = 2)
BIC(object, ...)
```

Arguments

`object`	An object of class `rav` containing an estimated averaging model.
`...`	Optionally more fitted model objects (see details).
`k`	Numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.

Details

The functions AIC and BIC are used, respectively, to extract the Akaike Information Criterion and the Bayesian Information Criterion of a model fitted by the function `rav`.

AIC is calculated as:

$$AIC = n \ln\left(\frac{RSS}{n}\right) + kp$$

where n is the number of data available, k is the penalty per parameter ()usually equal to 2), p is the number of parameters and RSS is the residual sum of squares.

BIC is calculated as:

$$BIC = n \ln\left(\frac{RSS}{n}\right) + \ln(n)p$$

As default, when $n/p < 40$, AIC and BIC are corrected in AICc and BICc:

$$AICc = AIC + \frac{2(p+1)p}{n-p-1}$$

$$BICc = BIC + \frac{\ln(n)(p+1)p}{n-p-1}$$

to avoid the correction, set `correct = FALSE`. On the contrary, if you want the correction, set `correct = TRUE`. When the argument `correct` is not specified, the rule $n/p < 40$ is applied.

As default, the functions extract the indices of the (first) best model. The optional argument `whichModel` can be specified to extract the indices of another model. Options are:

1. `"null"`: null model
2. `"ESM"`: equal scale values model
3. `"SAM"`: simple averaging model
4. `"EAM"`: equal-weights averaging model
5. `"DAM"`: differential-weight averaging model
6. `"IC"`: information criteria model

Value

A numeric value representing the information criterion of the selected model.

See Also

`rav`, `rAverage-package` `AIC`, `BIC`

Examples

```
## Not run:
data(fmdata1)
fm1 <- rav(fmdata1, lev=c(3,3))
AIC(fm1)
BIC(fm1)

## End(Not run)
```

`coef` *Extract coefficients from an averaging model*

Description

Function to extract fit model coefficients from an object returned by `rav`.

Usage

```
coef(object, ...)
```

Arguments

object	An object of class rav containing an estimated averaging model.
...	Optionally more fitted model objects.

Details

Returns the parameters of an averaging model fitted by the rav function, in the order: s0, w0, s(k,j), and w(k,j).

As default, the function extract the cofficients of the (first) best model. The optional argument whichModel can be specified to extract the values of another model. Options are:

1. "null": null model
2. "ESM": equal scale values model
3. "SAM": simple averaging model
4. "EAM": equal-weights averaging model
5. "DAM": differential-weight averaging model
6. "IC": information criteria model

Value

A numeric vector.

See Also

rav, rAverage-package

Examples

```
## Not run:
data(fmdata1)
fm1 <- rav(fmdata1, lev=c(3,3))
coef(fm1)
coef(fm1, whichModel="EAM")

## End(Not run)
```

---

datgen *Generating Noisy Responses for an Averaging Model*

Description

This function generates noisy synthetic responses R for an averaging model given the true parameters s0, w0, s(k,j), and w(k,j).

Usage

```
datgen(param, lev, t.par = FALSE, trials = 1, sd = 0, range = NULL)
```

Arguments

param	Numerical vector containing the true parameters for the function, with the order s0, w0, s(k,j), and w(k,j).
lev	Vector containing the number of levels of each factor. For instance, two factors with respectively 3 and 4 levels require lev = c(3,4).
t.par	Attribute that specifies whether the weight parameters should be written in the 't' form or in the 'w' form.
trials	Number of rows of the output matrix.
sd	Standard deviation of the noise addded to the responses R whithin each column of the output matrix.
range	Numeric vector. Range of the responses.

Value

A matrix object containing the generated responses of the averaging model, in the order: one-way design, two-way design, three way design, etc. See rav function.

See Also

rav, pargen, rav.indices, rAverage-package

Examples

```
## Not run:
# Generating random parameters for a 3x4 design:
par <- pargen(lev = c(3,4), s.range = c(0,20))
# Computing the responses:
R <- datgen(param=par, lev=c(3,4), sd=0) ; R
R <- datgen(param=par, lev=c(3,4), sd=1, trials=10, range=c(0,20)) ; R

## End(Not run)
```

---

fitted *Extract fitted values from an averaging model*

Description

Function to extract fitted values from an object returned by rav.

Usage

```
fitted(object, ...)
```

Arguments

object	An object of class rav containing an estimated averaging model.
...	Optionally more fitted model objects.

Details

Returns the expected responses given an averaging model fitted by the `rav` function.

As default, the function extract the fitted values of the (first) best model. The optional argument `whichModel` can be specified to extract the values of another model. Options are:

1. `"null"`: null model
2. `"ESM"`: equal scale values model
3. `"SAM"`: simple averaging model
4. `"EAM"`: equal-weights averaging model
5. `"DAM"`: differential-weight averaging model
6. `"IC"`: information criteria model

Value

A matrix of numeric values.

See Also

`rav`, `rAverage-package`

Examples

```
## Not run:
library(rAverage)
data(fmdata1)
fm1 <- rav(fmdata1, lev=c(3,3))
fitted(fm1)
fitted(fm1, whichModel="EAM")

## End(Not run)
```

`fmdata1` *rAverage dataset examples*

Description

Examples of dataset for R-Average analysis.
fmdata1: example of a 3x3 design. Original parameters:

s0 = 0.0	w0 = 0.0
sA = 12.9 1.5 18.3	wA = 1.4 0.3 0.5
sB = 5.2 5.0 2.3	wB = 1.6 1.7 1.7

fmdata2: example of a 3x5 design. Original parameters:

s0 = 0.0	w0 = 0.0
sA = 19.5 15.2 1.9	wA = 0.9 1.2 0.6
sB = 2.0 4.4 16.1 6.1 6.0	wB = 1.1 1.0 1.7 0.6 1.3

fmdata3: example of a 3x2x3 design. Original parameters:

s0 = 0.0	w0 = 0.0
sA = 5.9 5.2 9.8	wA = 0.9 1.1 2.2
sB = 14.5 2.0	wB = 0.5 1.9
sC = 8.5 1.5 10.7	wC = 0.6 0.7 1.4

Usage

```
data(fmdata1)
data(fmdata2)
data(fmdata3)
```

Format

A `matrix` object.

Examples

```
## Not run:
data(fmdata1)
fm1 <- rav(fmdata1, lev=c(3,3))
data(fmdata2)
fm2 <- rav(fmdata2, lev=c(3,5))
data(fmdata3)
fm3 <- rav(fmdata3, lev=c(3,2,3))

## End(Not run)
```

`outlier.replace` *Outlier detection and substitution*

Description

Starting by a previously estimated averaging model, this function detect outliers according to a Bonferroni method. The outliers can be substituted with a user-defined value.

Usage

```
outlier.replace(object, whichModel = NULL, alpha = 0.05, value = NA)
```

Arguments

`object`	An object of class `'rav'`, containing the estimated averaging models.
`whichModel`	Argument that specifies which of the predicted models has to be compared to the observed data. Options are: 1. `"null"`: null model 2. `"ESM"`: equal scale values model 3. `"SAM"`: simple averaging model 4. `"EAM"`: equal-weights averaging model 5. `"DAM"`: differential-weight averaging model 6. `"IC"`: information criteria As default setting, the (first) best model is used.
`alpha`	Critical value for the z-test on residuals.
`value`	Argument that can be used to set a replacement for the outliers. If a function is specified, it is applied to each column of the final matrix: the resulting value is used to replace outliers detected on the same column.

Value

A data object in which outliers have been removed or replaced.

See Also

`rav`, `rAverage-package`,

Examples

```
data(pasta)
model <- rav(pasta, subset="s04", lev=c(3,3), names=c("Price","Packaging"))
outlier.replace(model, value=mean)
outlier.replace(model, whichModel="IC", value=NA)
```

---

pargen *Generating random parameters for averaging responses*

Description

Generates a random set of parameters that follows an averaging rule.

Usage

```
pargen(lev, s.range = c(0,20), w.range = exp(c(-5,5)),
        I0 = FALSE, t.par = FALSE, digits = 2)
```

Arguments

lev	Numeric vector. Number of levels of each factor.
s.range	Numeric vector. Range of variability of the s-parameters.
w.range	Numeric vector. Range of variability of the w-parameters.
I0	Logical. If set to FALSE, parameter s0 and w0 are set to zero. If set TRUE initial parameters are free to be estimated.
t.par	Specifies if the weight parameters should be the 't' instead the 'w'.
digits	Numeric. Decimal rounding of the parameters.

Value

Vector contaning the random-generated parameters in the order s0,w0,s(k,i),w(k,i).

See Also

datgen, rav, rav.indices, rAverage-package

Examples

```
# Generating random parameters for a 3x4 design:
param <- pargen(lev = c(3,4))
```

---

pasta *Pasta experiment*

Description

Data of four subjects from the Pasta experiment. The table contains data of the one-way sub-designs, folowed by the data of the full-factorial design. Factors: Price (3 levels: 0.89, 0.99, 1.09) and Packaging (3 levels: box with window, box without window, plastic bag).

Usage

```
data(pasta)
```

Format

A data.frame object.

References

Massidda D., Polezzi D., Vidotto G. (2011). A Functional Measurement approach to cope the non-linearity of judgments in marketing research. *Proceedings of the 10th European Conference on Research Methodology for Business and Management Studies*. Normandy Business School, Caen, France, 348-354.

Examples

```
data(pasta)
```

---

rav *Analyzing the Family of the Averaging Models*

Description

rav (**R**-Average for **AV**eraging models) is a procedure for estimating the parameters of the averaging models of Information Integration Theory (Anderson, 1981). It provides reliable estimations of weights and scale values for a factorial experimental design (with any number of factors and levels) by selecting the most suitable subset of the parameters, according to the overall goodness of fit indices and to the complexity of the design.

Usage

```
rav( data, subset = NULL, mean = FALSE, lev, s.range = c(NA,NA),
     w.range = exp(c(-5,5)), I0 = FALSE, par.fixed = NULL, all = FALSE,
     IC.diff = c(2,2), Dt = 0.1, IC.break = FALSE, t.par = FALSE,
     verbose = FALSE, title = NULL, names = NULL, method = "BFGS",
     start = c(s=NA,w=exp(0)), lower = NULL, upper = NULL, control = list() )
```

Arguments

data	An object of type matrix, data.frame or vector containing the experimental data. Each column corresponds to an experimental design of factorial plan (in order: one-way design, two-way design, ..., full factorial design; see the example for further details). Columns must be sorted combining each level of the first factor with all the levels of the following factors. The first column be used to set an identification code (ID) to label the subjects (see the attribute subset).
subset	Character, numeric or factor attribute that selects a subset of experimental data for the analysis (see the examples).
mean	Logical value wich specifies if the analysis must be performed on raw data (mean = FALSE) or on the average of columns of the data matrix (mean = TRUE).
lev	Vector containing the number of levels of each factor. For instance, two factors with respectively 3 and 4 levels require lev = c(3,4).
s.range, w.range	The range of s and w parameters. Each vector must contains, respectively, the minimum and the maximum value. For s-parameters, if the default value NA is set, the minimum and the maximum values of data matrix will be used. For t-parameters, the values exp(-5) and exp(+5) will be used. This values will be the bounds for parameters in the estimation process when the minimization algorithm is L-BFGS-B. The arguments s.range and w.range are a simple and quick way to specify the bounds for scale and weight parameters. A more complex but complete way is to use the arguments lower and upper. If lower and upper will be specified, s.range and w.range will be ignored.

I0 — Logical. If set FALSE, the s0 and w0 parameters are forced to be zero. If set TRUE, the s0 and w0 parameters are free to be estimated.

par.fixed — This argument allows to constrain one or more parameters to a specified value. Default setting to NULL indicates that all the scale and weight parameters will be estimated by the algorithmic procedure. Alternatively, it can be specified the name of the type of parameters to constrain. The argument par.fixed = "s" constrains s-parameters, par.fixed = "w" constrains w-parameters and par.fixed = c("s","w") constrains both s and w parameters. Also, using "t" instead of "w" constrains directly t-parameters. in these cases a graphical interface is displayed and the values can be specified.

all — Logical. If set TRUE the information criterion tests all the possible combinations of weights (see details). The default value FALSE implies a preselection of a subset of combination based on the results of the previous steps of the algorithm. WARNING: with all = TRUE the procedure is generally more time-consuming (depending on the size of the experimental design), but can provide more reliable estimations than the standard procedure.

IC.diff — Vector containing the cut-off values (of both BIC and AIC indices) at which different models are considered equivalent. Default setting: BIC difference = 2.0, AIC difference = 2.0 (IC.diff = c(2.0, 2.0)).

Dt — Numeric attribute that set the cut-off value at which different t-parameters must be considered equal (see details).

IC.break — Logical argument which specifies if to run the Information Criteria Procedure.

t.par — Logical. Specifies if the output must shows the t-parameters instead of the w-parameters.

verbose — Logical. If set TRUE the function prints general informations for every step of the information criterion procedure.

title — Character. Label to use as title for output.

names — Vector of character strings containing the names of the factors.

method — The minimization algorithm that has to be used. Options are: "L-BFGS-B", "BFGS", "Nelder-Mead", "SANN" and "CG". See optim documentation for further information.

start — Vector containing the starting values for respectively scale and weight parameters. For the scale parameters, if the default value NA is set, the mean of data is used as starting value. For the weight parameters, the starting default value is 1.

lower — Vector containing the lower values for scale and weight parameters when the minimization routine is L-BFGS-B. With the default setting NULL, s-parameters are set to the first value specified in s.range while w-parameters are set to the first value specified in w.range. Values must be specified in the order: s0,w0,s,w. For example, for a 3x3 design, in the lower vector the positions of parameters must be: s0,w0, sA1,sA2,sA3, sB1,sB2,sB3, wA1,wA2,wA3, wB1,wB2,wB3.

upper — Vector containing the upper values for scale and weight parameters when the minimization routine is L-BFGS-B. With the default setting NULL, s-parameters are set to the second value specified in s.range while w-parameters are set to the second value specified in w.range. Values must be specified in the order:

s0,w0,s,w. For example, for a 3x3 design, in the upper vector the positions of parameters must be: s0,w0, sA1,sA2,sA3, sB1,sB2,sB3, wA1,wA2,wA3, wB1,wB2,wB3.

control — A list of control parameters. See the optim documentation for further informations. control argument can be used to change the maximum iteration number of minimization routine. To increase the number, use: control=list(maxit=N), where N is the number of iterations (100 for default).

Details

The rav function implements the R-Average method (Vidotto & Vicentini, 2007; Vidotto, Massidda & Noventa, 2010), for the parameter estimation of averaging models. R-Average consists of several procedures which compute different models with a progressive increasing degree of complexity:

1. Null Model (null): identifies a single scale value for all the levels of all factors. It assumes constant weights.
2. Equal scale values model (ESM): makes a distinction between the scale values of different factors, estimating a single s-parameter for each factor. It assumes constant weights.
3. Simple averaging model (SAM): estimates different scale values between factors and within the levels of each factor. It assumes constant weights.
4. Equal-weight averaging model (EAM): differentiates the weighs between factors, but not within the levels of each factor.
5. Differential-weight averaging model (DAM): differentiates the weighs both between factors and within the levels of each factor.
6. Information criteria (IC): the IC procedure starts from the EAM and, step by step, it frees different combinations of weights, checking whether a new estimated model is better than the previous baseline. The Occam razor, applied by means of the Akaike and Bayesian information criteria, is used in order to find a compromise between explanation and parsimony.

Finally, only the best model is shown.

The R-Average procedures estimates both scale values and weight parameters by minimizing the residual sum of squares of the model. The objective function is then the square of the distance between theoretical responses and observed responses (Residual Sum of Squares). For a design with k factors with i levels, theoretical responses are defined as:

$$R = \sum(s_{ki}w_{ki}) / \sum w_{ki}$$

where any weight parameter w is defined as:

$$w = exp(t)$$

Optimization is performed on t-values, and weights are the exponential transformation of t. See Vidotto (2011) for details.

Value

An object of class "rav". The method summary applied to the rav object prints all the fitted models. The functions fitted.values, residuals and coefficients can be used to extract respectively fitted values (predicted responses), the matrix of residuals and the set of estimated parameters.

Author(s)

Supervisor: Prof. Giulio Vidotto <giulio.vidotto@unipd.it>

University of Padova, Department of General Psychology
QPLab: Quantitative Psychology Laboratory

version 0.0:
Marco Vicentini <marco.vicentini@gmail.com>

version 0.1 and following:
Stefano Noventa <stefano.noventa@univr.it>
Davide Massidda <davide.massidda@gmail.com>

References

Akaike, H. (1976). Canonical correlation analysis of time series and the use of an information criterion. In: R. K. Mehra & D. G. Lainotis (Eds.), *System identification: Advances and case studies* (pp. 52-107). New York: Academic Press. doi: 10.1016/S0076-5392(08)60869-3

Anderson, N. H. (1981). *Foundations of Information Integration Theory*. New York: Academic Press. doi: 10.2307/1422202

Anderson, N. H. (1982). *Methods of Information Integration Theory*. New York: Academic Press.

Anderson, N. H. (1991). Contributions to information integration theory: volume 1: cognition. Lawrence Erlbaum Associates, Hillsdale, New Jersey. doi: 10.2307/1422884

Anderson, N. H. (2007). Comment on article of Vidotto and Vicentini. *Teorie & Modelli*, Vol. 12 (1-2), 223-224.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *Journal Scientific Computing*, 16, 1190-1208. doi: 10.1137/0916069

Kuha, J. (2004). AIC and BIC: Comparisons of Assumptions and Performance. *Sociological Methods & Research*, 33 (2), 188-229.

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal, 7, 308-313*. doi: 10.1093/comjnl/7.4.308

Vidotto, G., Massidda, D., & Noventa, S. (2010). Averaging models: parameters estimation with the R-Average procedure. *Psicologica*, 31, 461-475. URL https://www.uv.es/psicologica/articulos3FM.10/3Vidotto.pdf

Vidotto, G. & Vicentini, M. (2007). A general method for parameter estimation of averaging models. *Teorie & Modelli*, Vol. 12 (1-2), 211-221.

See Also

rAverage-package, rav.single, datgen, pargen, rav.indices, rav2file, outlier.replace, optim

Examples

```
## Not run:
# -------------------------------------
# Example 1: 3x3 factorial design
# -------------------------------------
```

```
# The first column is filled with a sequence of NA values.
data(fmdata1)
fmdata1
# For a two factors design, the matrix data contains the one-way
# sub-design and the two-ways full factorial design observed data.
# Pay attention to the columns order:
# sub-design: A1, A2, A3, B1, B2, B3
# full factorial: A1B1, A1B2, A1B3, A2B1, A2B2, A2B3, A3B1, A3B2, A3B3
# Start the R-Average procedure:
fm1 <- rav(fmdata1, lev=c(3,3))
# (notice that 'range' argument specifies the range of the response scale)
fm1 # print the best model selected
summary(fm1) # print the fitted models

# To insert the factor names:
fact.names <- c("Name of factor A", "Name of factor B")
fm1 <- rav(fmdata1, lev=c(3,3), names=fact.names)

# To insert a title for the output:
fm1 <- rav(fmdata1, lev=c(3,3), title="Put your title here")

# To supervise the information criterion work flow:
fm1 <- rav(fmdata1, lev=c(3,3), verbose=TRUE)

# To increase the number of iterations of the minimization routine:
fm1 <- rav(fmdata1, lev=c(3,3), control=list(maxit=5000))

# To change the estimation bounds for the scale parameters:
fm1.sMod <- rav(fmdata1, lev=c(3,3), s.range=c(0,20))

# To change the estimation bounds for the weight parameters:
fm1.wMod <- rav(fmdata1, lev=c(3,3), w.range=c(0.01,10))

# To set a fixed value for weights:
fm1.fix <- rav(fmdata1, lev=c(3,3), par.fixed="w")

# rav can work without sub-designs. If any sub-design is not available,
# the corresponding column must be coded with NA values. For example:
fmdata1[,1:3] <- NA
fmdata1
fmdata1 # the A sub-design is empty
fm1.bis <- rav(fmdata1, lev=c(3,3), title="Sub-design A is empty")

# Using a subset of data:
data(pasta)
pasta
# Analyzing "s04" only:
fact.names <- c("Price","Packaging")
fm.subj04 <- rav(pasta, subset="s04", lev=c(3,3), names=fact.names)

# ---------------------------------------
# Example 2: 3x5 factorial design
# ---------------------------------------
```

```
data(fmdata2)
fmdata2 # (pay attention to the columns order)
fm2 <- rav(fmdata2, lev=c(3,5))
# Removing all the one-way sub-design:
fmdata2[,1:8] <- NA
fm2.bis <- rav(fmdata2, lev=c(3,5))

# ---------------------------------------
# Example 3: 3x2x3 factorial design
# ---------------------------------------
data(fmdata3) # (pay attention to the columns order)
fm3 <- rav(fmdata3, lev=c(3,2,3))
# Removing all the one-way design and the AxC sub-design:
fmdata3[,1:8] <- NA # one-way designs
fmdata3[,15:23] <- NA # AxC design
fm3 <- rav(fmdata3, lev=c(3,2,3))

## End(Not run)
```

`rav.grid` *Generating an empty dataset in* `rav` *format*

Description

`rav.grid` is a function that generates an empty (NAs filled) dataset according to the 'rAverage' format.

Usage

```
rav.grid(lev, trials = 1, subset = FALSE, names = NULL)
```

Arguments

`lev`	Vector containing the number of levels of each factor. For instance, two factors with respectively 3 and 4 levels require `lev = c(3,4)`.
`trials`	Number of rows of the output matrix.
`subset`	Logical. Indicates whether the matrix should contain a first column for subset coding.
`names`	Character. Indicates the column names (optional).

Value

A `data.frame` object.

See Also

`rav`, `pargen`, `datgen`, `rAverage-package`

Examples

```
rav.grid(lev=c(3,2,3), trials=5, names=c("Hk","Fa","Mg"))
```

`rav.indices` *Fit indices for averaging models*

Description

The function `fit.indices` returns the fit indices for the averaging model given the parameters `s0`, `w0`, `s(k,j)`, and `w(k,j)`.

Usage

```
rav.indices(param, lev, data, t.par = FALSE, subset = NULL,
    n.pars = NULL, names = NULL, title = NULL)
```

Arguments

`param`	Numerical vector containing the parameters for the function, with the order `s0`, `w0`, `s(k,j)`, and `w(k,j)`.
`lev`	Vector containing the number of levels of each factor. For instance, two factors with respectively 3 and 4 levels require `lev = c(3,4)`.
`data`	A `matrix` or a `data.frame` object containing the experimental data. Each column corresponds to an experimental design (in order: one-way design, two-way design, ..., full factorial design; see the example for further details). WARNING: previous versions needed a first column filled with the initial state values (s0*w0) or `NA` values. This is no longer valid. Nevertheless, the first column can be used to label the data (see the attribute `subset`).
`t.par`	Specifies whether the weight parameters should be the in 't' form or in the 'w' form.
`subset`	Character, numeric or factor attribute that selects a subset of experimental data for the analysis (see the examples).
`n.pars`	Number of parameters of the model. If `NULL`, `n.pars` will be calculated from the function.
`names`	Vector of character strings containing the names of the factors.
`title`	Character specifying a title for the output.

Details

Returns the main fit indices (AIC, BIC, R-squared, Adjusted R-squared), the estimated parameters, the fitted values and the residuals of an averaging model.

Value

An object of class `"indices"`.

See Also

rav, rAverage-package

Examples

```
## Not run:
data(fmdata1)
s <- c(12.9, 1.5, 18.3, 5.2, 5.0, 2.3)
w <- c(1.4, 0.3, 0.5, 1.6, 1.7, 1.7)
param <- c(NA,NA, s, w)
# Estimated model by rav:
fit1 <- rav(fmdata1, lev=c(3,3)) ; fit1
# Fitted model by original parameters:
fit2 <- rav.indices(param=param, lev=c(3,3), data=fmdata1) ; fit2

## End(Not run)
```

---

rav.single *Single subject analysis with averaging models*

Description

Analyzes averaging models for every single subjects in a data matrix and store the estimated parameters in a list.

Usage

```
rav.single(data,...)
```

Arguments

data	An object of type data.frame containing data. Each column represents an experimental design of a factorial plan (see the function rav for details). Each row must contain single-trial responses for each subject. Further, the first column must contain labels describing an identification code for subjects.
...	Further arguments for the rav function (the argument subset must not be specified).

Details

The rav.single function is a wrapper for the rav function. Using rav, rav.sigle analyzes subjects one at time, specifying time by time a different value for subset.

Value

A list object in which each slot contains results of a single subject. The ordering of the subjects is the same as in the input data matrix.

See Also

rav, rAverage-package

Examples

```
## Not run:
data(pasta)
model <- rav.single(pasta,lev=c(3,3))
model$s41 # extracts the subject 's41'

## End(Not run)
```

---

rav2file *Export* rav *results*

Description

The function exports to a text file the estimated parameters or the model residuals from a call to rav.

Usage

```
rav2file(object, what = c("resid","param"), whichModel = NULL,
         file = file.choose(), sep = ",", dec = ".")
```

Arguments

object	An object analyzed by the function rav.
what	Character string indicating which output should be stored in the file, if raw residuals (what = "resid") or parameters (what = "param").
whichModel	Argument that specifies from which model values must be extracted. Options are: 1. "null": null model 2. "ESM": equal scale values model 3. "SAM": simple averaging model 4. "EAM": equal-weights averaging model 5. "DAM": differential-weight averaging model 6. "IC": information criteria As default setting, the values of the (first) best model are extracted.
file	A character string naming the file to write. As default, the function opens a mask to choose or build a file interactively.
sep	Field separator string. Values within each row will be separated by this string.
dec	String argument used to specify the decimal separator.

See Also

rav, rav.single

Examples

```
## Not run:
data(pasta)
model <- rav.single(pasta,lev=c(3,3))
rav2file(model, what="resid", file="PastaResid-1.csv")
rav2file(model, what="resid", file="PastaResid-2.csv", sep=";", dec=",")

## End(Not run)
```

`rAverage` *Parameter estimation for the averaging model of Information Integration Theory*

Description

The R-Average package implements a method to identify the parameters of the Averagingcmodel of Information Integration Theory (Anderson, 1981), following the spirit of the so-called "principle of parsimony".

Name of the parameters:

`s0,w0`: initial state values of the Averaging Model.
`s(k,j)`: scale value of the `j`-th level of `k`-th factor.
`w(k,j)`: weight value of the `j`-th level of `k`-th factor.

Details

Package:	rAverage
Type:	Package
Version:	0.5-8
Date:	2017-07-29
License:	GNU (version 2 or later)

Functions of the R-Average package:

`rav`: estimates the parameters for averaging models.
`fitted`: extracts the predicted values of the best model from a `rav` object.
`residuals`: extracts the residuals from a `rav` object.
`coefficients`: extracts the parameters from a `rav` object.
`outlier.replace`: given an estimated averaging model with the `rav` function, it detects and replace outliers from the residual matrix. `rav.indices`: given a set of parameters `s` and `w` and a

matrix of observed data, it calculates the fit indices for the averaging model.
`datgen`: returns the responses `R` for averaging models given the set of parameters `s` and `w`.
`pargen`: generates pseudorandom parameters for the averaging model.
`rav.grid`: generates an empty matrix in 'rav' format.
`rav.single`: single subjects analysis over an aggregated data matrix.
`rav2file`: store the reesults of `rav` into a text file.

Author(s)

Supervisor: Prof. Giulio Vidotto <giulio.vidotto@unipd.it>

University of Padova, Department of General Psychology
QPLab: Quantitative Psychology Laboratory

version 0.0:
Marco Vicentini <marco.vicentini@gmail.com>

version 0.1 and following:
Stefano Noventa <stefano.noventa@univr.it>
Davide Massidda <davide.massidda@gmail.com>

References

Akaike, H. (1976). Canonical correlation analysis of time series and the use of an information criterion. In: R. K. Mehra & D. G. Lainotis (Eds.), *System identification: Advances and case studies* (pp. 52-107). New York: Academic Press. doi: 10.1016/S0076-5392(08)60869-3

Anderson, N. H. (1981). *Foundations of Information Integration Theory*. New York: Academic Press. doi: 10.2307/1422202

Anderson, N. H. (1982). *Methods of Information Integration Theory*. New York: Academic Press.

Anderson, N. H. (1991). Contributions to information integration theory: volume 1: cognition. Lawrence Erlbaum Associates, Hillsdale, New Jersey. doi: 10.2307/1422884

Anderson, N. H. (2007). Comment on article of Vidotto and Vicentini. *Teorie & Modelli*, Vol. 12 (1-2), 223-224.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *Journal Scientific Computing*, 16, 1190-1208. doi: 10.1137/0916069

Kuha, J. (2004). AIC and BIC: Comparisons of Assumptions and Performance. *Sociological Methods & Research*, 33 (2), 188-229.

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal, 7, 308-313*. doi: 10.1093/comjnl/7.4.308

Vidotto, G., Massidda, D., & Noventa, S. (2010). Averaging models: parameters estimation with the R-Average procedure. *Psicologica*, 31, 461-475. URL https://www.uv.es/psicologica/articulos3FM.10/3Vidotto.pdf

Vidotto, G. & Vicentini, M. (2007). A general method for parameter estimation of averaging models. *Teorie & Modelli*, Vol. 12 (1-2), 211-221.

See Also

`rav`, `datgen`, `pargen`, `rav.indices`, `fmdata1`, `pasta`, `optim`

`residuals` *Extract residuals from an averaging model*

Description

Function to extract residuals from an object returned by `rav`.

Usage

```
residuals(object, ...)
```

Arguments

`object`	An object of class `rav` containing an estimated averaging model.
`...`	Optionally more fitted model objects.

Details

Returns the residuals of an averaging model fitted by the `rav` function. When `standard = TRUE`, residuals will be transformed in z-scale (mean=0 and sd=1 in each column).

As default, the function extract the residuals of the (first) best model. The optional argument `whichModel` can be specified to extract the values of another model. Options are:

1. `"null"`: null model
2. `"ESM"`: equal scale values model
3. `"SAM"`: simple averaging model
4. `"EAM"`: equal-weights averaging model
5. `"DAM"`: differential-weight averaging model
6. `"IC"`: information criteria model

Value

A matrix of numeric values.

See Also

`rav`, `rAverage-package`

Examples

```
## Not run:
data(fmdata1)
fm1 <- rav(fmdata1, lev=c(3,3))
residuals(fm1)
residuals(fm1, whichModel="EAM")
residuals(fm1, whichModel="EAM", standard=TRUE)

## End(Not run)
```

Index