Package 'rAverage'

February 15, 2012

Version 0.3-5

Date 2011-01-27

Title Parameter estimation for the Averaging model of Information Integration Theory

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

Maintainer Stefano Noventa <stefano.noventa@unipd.it>, Davide Massidda <davide.massidda@unipd.it>

Depends R (>= 2.8), methods, tcltk, foreign

Suggests RODBC

Description Parameter estimation for the Averaging model of Anderson's Information Integration Theory

License GPL (>= 2)

URL http://qplab.psy.unipd.it/

Repository CRAN

Date/Publication 2011-01-28 12:10:06

R topics documented:

averaging *Generating Responses for an Averaging Model*

Description

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

Usage

```
averaging(param, lev, trials = 1, sd = 0, range = 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).
trials	Number of rows of the output matrix.
sd	Variability of the responses R whithin each column of the output matrix.
range	Numeric vector. Range of the responses.

Value

A matrix object containing the 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.indexes, rAverage-package, rav.cmd

Examples

```
# Generating random parameters for a 3x4 design:
# par <- pargen(lev = c(3,4), s.range = c(0,20))
# Computing the responses:
# R <- averaging(param=par, lev=c(3,4), sd=0) ; R
# R <- averaging(param=par, lev=c(3,4), sd=1, trials=10, range=c(0,20)) ; R
```

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

```
# 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))
```

---

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 = c(0.5,3),
        I0 = FALSE, digits = 1)
```

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.
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

averaging, rav, rav.indexes, rAverage-package, rav.cmd

Examples

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

---

pasta *Pasta experiment*

Description

Dataset (R-Average form) of Pasta experiment. Factors: Price (3 levels: 0.89, 0.99, 1.09) x Packaging (3 levels).

Usage

```
data(pasta)
```

Format

A matrix object.

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 indexes and to the complexity of the design. Its attributes can be handled by menas of a GUI (see the rav.cmd command for details).

Usage

```
rav(data, subset = NULL, lev, all= FALSE, sim = 0,
    range = NULL, start = NULL, lower = NULL, upper = NULL,
    I0 = FALSE, s.fixed = FALSE, w.fixed = FALSE,
    IC.diff = c(2, 2), delta.weights = 0.1,
    method = "L-BFGS-B", control = list(),
    title = NULL, names = NULL, verbose = FALSE)
```

Arguments

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 indexes the data (see the attribute subset).
subset	Character, numeric or factor attribute that selects a subset of experimental data for the analysis (see the examples).
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).
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 provides more reliable estimations than the standard procedure.
sim	Number of iterations for the Monte-Carlo simulation procedure. The default setting 0 implies that no Monte-Carlo will be runned.
range	A vector containing the scale response range. If NULL, the minimum and maximum values of the data matrix will be used. This values will be used as bounds for the s-parameters when the minimization algorithm is L-BFGS-B.

start	Vector containing the start scale and weight values, according to the following order: s0, w0, s(k,j), w(k,j), for k and j set by the factorial design. With the default setting NULL, rav loads the default parameters: s0 = 0, w0 = 1, s(k,j) = max(range)/2, w = 1.
lower	Vector containing the lower scale and weight values, according to the following order: s0, w0, s(k,j), w(k,j), for k and j set by the factorial design. With the default setting NULL, when the minimization algorithm is L-BFGS-B, rav loads the default parameters: s0 = 0, w0 = 1e-5, s(k,j) = min(range), w = 1e-5 for each k and j.
upper	Vector containing the upper scale and weight values, according to the following order: s0, w0, s(k,j), w(k,j), for k and j set by the factorial design. With the default setting NULL, when the minimization algorithm is L-BFGS-B, rav loads the default parameters: s0 = max(range)/10, w0 = 10, s(k,j) = max(range), w = 10 for each k and j.
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.
s.fixed	Logical. Default setting to FALSE indicates that the scale values parameters are estimated by the algorithmic procedure. Otherwise, their values are fixedto the mean values of the one-way sub-designs (strongly suggested for column data that shows a normal distribution).
w.fixed	List or logical attribute. Can be used to set fixed values for the weight parameters. See the examples for further details. If set TRUE the procedure opens a pop-up that allows to fix the parameters without using a list object.
IC.diff	Vector containing the cut-off values (of both BIC and AIC indexes) at which different models are considered equivalent. Default setting: BIC difference = 2.5, AIC difference = 2.0 (IC.diff = c(2.5, 2.0)).
delta.weights	Numeric attribute that set the cut-off value at which different weights must be considered equal.
method	The minimization algorithm that has to be used. Options are: "L-BFGS-B", "BFGS", "Nelder-Mead", "SANN", "CG". See optim documentation for further informations.
control	A list of control parameters. See optim documentation for further informations.
title	Character. Puts a title on the output.
names	Vector of character strings containing the names of the factors.
verbose	Logical. If set TRUE the function prints general informations for every step of the information criterion procedure.

Details

The rav function implements the R-Average procedure (Vidotto & Vicentini, 2007), for the estimation of the parameters of the Averaging Model. Three different cases are tested: EAM (Equal weights Averaging Model), DAM (Differential weights Averaging Model) and a model selected by means of an Information Criterion. The routine estimates the scale values and the weight parameters by minimizing the residual sum of squares of the model.

The Information Criterion procedure starts from the EAM and, step by step, analyzes different combinations of weights, checking whether a new estimated model is better than the previous baseline (to set the criterion for AIC/BIC differences between models change the argument `IC.diff`). Finally, only the best model between EAM, DAM, and IC is shown.

Value

An object of class `"rav"`. The method `summary` applied to the `rav` object prints all the fitted models. The functions `fitted`, `residuals` and `parameters` can be used to extract respectively fitted values `R`, the matrix of residuals and the set of parameter estimated.

Author(s)

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

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

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

University of Padova, Department of General Psychology
QPLab: Quantitative Psychology Laboratory http://qplab.psy.unipd.it/

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.

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

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.

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.

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*.

Vidotto, G., Massidda, D., & Noventa, S. (2010). Averaging models: parameters estimation with the R-Average procedure. *Psicologica*, 31, 461-475.

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.cmd, averaging, pargen, rav.indexes, 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), range=c(0,20))
# (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), range=c(0,20), names=fact.names)

# To insert a output title:
out.title <- c("Put your title here")
fm1 <- rav(fmdata1, lev=c(3,3), title=out.title)
# (notice that the range argument is not required, but it is recommended)

# To supervise the information criterion work flow:
fm1 <- rav(fmdata1, lev=c(3,3), range=c(0,20), verbose=TRUE)

# To increase the number of iterations of the minimization routine:
fm1 <- rav(fmdata1, lev=c(3,3), range=c(0,20), control=list(maxit=5000))

# To set a fixed value for weights:
wfix <- list(A=c(NA,0.4,0.4), B=c(NA,NA,NA))
wfix
# Warning: NA specify no-fixed weights
fm1.fix <- rav(fmdata1, lev=c(3,3), range=c(0,20), w.fixed=wfix)
# Otherwise, it's possible to call a GUI:
fm1.fix <- rav(fmdata1, lev=c(3,3), range=c(0,20), w.fixed=TRUE)

# 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), range=c(0,20), title="Sub-design A is empty")
```

```
# Using a subset of data:
data(pasta)
pasta
# Analyzing "subj.04" only:
fact.names <- c("Price","Packaging")
fm.subj04 <- rav(pasta, subset="subj.04", lev=c(3,3), range=c(0,20), names=fact.names)

# ---------------------------------------
# Example 2: 3x5 factorial design
# ---------------------------------------
data(fmdata2)
# (Pay attention to the columns order)
fmdata2
fm2 <- rav(fmdata2, lev=c(3,5), range=c(0,20))
# Removing all the one-way sub-design:
fmdata2[,1:8] <- NA
fm2.bis <- rav(fmdata2, lev=c(3,5), range=c(0,20))

# ---------------------------------------
# Example 3: 3x2x3 factorial design
# ---------------------------------------
data(fmdata3)
# (Pay attention to the columns order)
fm3 <- rav(fmdata3, lev=c(3,2,3), range=c(0,20))
# 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), range=c(0,20))

## End(Not run)
```

`rav.cmd` *GUI for rav function*

Description

The command `rav.cmd()` calls the graphical user interface for R-Average procedure.

Usage

```
rav.cmd(lev=NULL, range=NULL, names=NULL)
```

Arguments

`lev`	Optional argument. A vector containing the number of levels of each factor. Example: two factors with 3 and 4 levels requires a code like `lev=c(3,4)`. If the default setting NULL is left, the number of levels and factors will be required by the GUI, otherwise that step will be skipped.

range	Optional argument.A vector containing the range of the responses R. If the default setting NULL is left, the range will be required by the GUI, otherwise that step will be skipped.
names	Optional argument. Vector of character strings, containing the names of the factors. If the default setting NULL is left, the range will be required by the GUI, otherwise that step will be skipped.

Details

The arguments are optional.

Value

The GUI is called.

See Also

rAverage-package, rav

Examples

```
# Show the GUI
# rav.cmd()
# Show the GUI interface skipping the information about two factors
# With three levels, range between 0 and 20 and names A and B
# rav.cmd(lev=c(3,3),range=c(0,20),names=c("A","B"))
```

---

rav.indexes *Fit indexes for an averaging model*

Description

This function returns the infexes of fit for an averaging model given the parameters s0, w0, s(k,j), and w(k,j).

Usage

```
rav.indexes(param, lev, data, subset=NULL,
    I0=FALSE, 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 indexes the data (see the attribute subset).
subset	Character, numeric or factor attribute that selects a subset of experimental data for the analysis (see the examples).
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.
names	Vector of character strings containing the names of the factors.
title	Character. Puts a title on the output.

Details

Returns the principal indexes of fit (AIC, BIC, R squared, Adjusted R squared).

Value

An object of class "indexes".

See Also

averaging, pargen, rav, rAverage-package, rav.cmd

Examples

```
# 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)
# par <- c(0,0, s, w)
# Estimated model by rav:
# fit1 <- rav(fmdata1, lev=c(3,3))
# Fitted model by original parameters:
# fit2 <- rav.indexes(param=par, lev=c(3,3), data=fmdata1)
```

---

rAverage — *Parameters estimation for the Averaging model of Information Integration Theory*

Description

The R-Average package implements a procedure to identify the parameters of the Averaging model 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:	0.3-5
Date:	2010-10-31
License:	GNU (version 2 or later)

Functions of the R-Average package:

`rav`: analyzes the averaging model and identifies the parameters.
`averaging`: returns the responses `R` for averaging models given the set of parameters `s` and `w`.
`pargen`: generates random data for the averaging model.
`rav.cmd`: calls the GUI (graphical user interface) for handling the attribute of the `rav` procedure.
`rav.indexes`: returns the indexes of fit of a model.

Author(s)

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

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

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

University of Padova, Department of General Psychology
QPLab: Quantitative Psychology Laboratory http://qplab.psy.unipd.it/

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.

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

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.

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.

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*.

Vidotto, G., Massidda, D., & Noventa, S. (2010). Averaging models: parameters estimation with the R-Average procedure. *Psicologica*, 31, 461-475.

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`, `averaging`, `pargen`, `rav.indexes`, `rav.cmd`, `fmdata1`, `pasta`, `optim`

Index