Package ‘MLDS’

August 20, 2023

Type Package
Title Maximum Likelihood Difference Scaling
Version 0.5.1
Date 2023-08-19
Author Kenneth Knoblauch and Laurence T. Maloney,
       based in part on C code written by Laurence T. Maloney
       and J. N. Yang
Maintainer Kenneth Knoblauch <ken.knoblauch@inserm.fr>
Depends R (>= 3.0), graphics, stats, utils, base
Imports MASS
LazyData yes
Description Difference scaling is a method for scaling perceived
       supra-threshold differences. The package contains functions that
       allow the user to design and run a difference scaling experiment,
       to fit the resulting data by maximum likelihood and test the
       internal validity of the estimated scale.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2023-08-20 08:22:33 UTC

R topics documented:

- MLDS-package .................................................. 2
- as.mlds.df .................................................. 3
- AutumnLab .................................................. 4
- binom.diagnostics .......................................... 5
- boot.mlds .................................................. 7
- DisplayOneQuad .............................................. 8
- fitted.mlds ................................................ 9
- Get6pts ...................................................... 10
MLDS-package

~~ MLDS ~~ Maximum Likelihood Difference Scaling

Description

Difference scaling is a method for scaling perceived supra-threshold differences. The package contains functions that allow the user to design and run a difference scaling experiment and to fit the resulting data by maximum likelihood.

Details

The package provides a function, `mlds` for estimating a perceptual scale using the data obtained from one or several difference scaling experiments. A second function, `simu.6pt` permits the interval validity of the scale to be evaluated using a bootstrap method. Several methods are supplied for accessing and examining the ‘mlds’ object generated by estimating the scale.

Author(s)

Kenneth Knoblauch and Laurence T. Maloney

Maintainer: Ken Knoblauch <ken.knoblauch@inserm.fr>

References


Examples

```r
library(MLDS)
data(kk1) # data for one subject for 330 trials of the same experiment
plot(mlds(kk1)) # fit and plot the fitted difference scale
```

As.mlds.df

Coerces a data.frame to mlds.df

Description

Coerce a data frame from an MLDS experiment to an object of class mlbs.df or mlds.df by adding column names and attributes so that it will be properly treated by methods related to the MLDS functions.

Usage

```r
as.mlds.df(d, ...)  
as.mlbs.df(d, ...)
```

```r
# S3 method for class 'data.frame'
as.mlds.df(d, st, ...)
as.mlbs.df(d, st, ...)
```

```r
df2mlds.df(d, st)
```

Arguments

- `d`: a 4 or 5 column data frame from an MLDS experiment, with one column of responses followed by three or four, respectively, indicating the indices of the stimuli from each trial or a data frame of > 5 columns with the response in the first column and the covariates (signed indicator variables) for the stimuli of each trial in the rest of the columns.
- `st`: numeric indicating the stimulus levels from the MLDS experiment
- `...`: additional arguments passed to the methods

Details

This function coerces a data frame from an MLDS experiment to an object of class mlbs.df or mlds.df but inheriting from class data.frame. It changes the column names to `resp` and `S1`, `S2`, `S3` and if a quad experiment, `S4` and adds two attributes: `stimulus`, a vector of scale values used in plotting the estimated scale and `invord`, a logical vector indicating the order of the presentation of pairs (were the larger scale values on the bottom or not) which is used with the SwapOrder function. Note that when the argument is in signed indicator form, just a data frame is returned with no special mlds attributes.
Value

data.frame of class mlbs.df or mllds.df with stimulus and invord attributes, unless the input is in signed indicator form. See details.

Note

If the scale starts at 0, then 1 should be added to each scale value because the scale values will be used as indices and R indices start at 1, not 0.

Author(s)

Kenneth Knoblauch

See Also

mlds

AutumnLab Difference Scale Judgement Data Set

Description

Judgments and stimulus ranks for one observer during one session from a difference scaling experiment. On each trial, the observer is presented with 4 stimuli \((a, b, c, d)\) from a physical scale with \(a < b < c < d\). The stimuli are presented as two pairs \((a, b)\) and \((c, d)\) and the observer judges between which pair the perceived difference is greater.

Usage

data(AutumnLab)

Format

A data frame with 210 observations on the following 5 variables.

- resp: an integer vector of 0, 1, indicating the choice of the observer between the pairs \((S1, S2)\) vs \((S3, S4)\)
- S1: an integer vector, rank order of physical stimulus level for the lowest/weakest stimulus
- S2: an integer vector, rank order of physical stimulus 2
- S3: an integer vector, rank order of physical stimulus 3
- S4: an integer vector, rank order of higher/strongest stimulus
Details

The difference scaling paradigm was used to estimate changes of image quality with increasing rate of image compression. Image compression was performed using Vector Quantisation and the standard Lab color space. In the overall experiment, the compression was performed for several images and in several different color spaces. Scales were obtained from several observers, as well. These data are from one observer, for one image with compression performed in the Lab color space. The 10 compression levels tested were 1 (no compression), and 6-30% reduction in file size in steps of 3. The physical stimulus levels are attached to the data.frame as an attribute, stimulus. There is also an attribute, invord, which is a vector of type logical of length 210, which can be used with SwapOrder to restore the order of the pairs to correspond to physical position (lower/upper) in the experiment, rather than the way it is stored, stimulus level (lower/higher compression).

References


Examples

data(AutumnLab)
plot(mlds(AutumnLab))

binom.diagnostics

Diagnostics for Binary GLM

Description

Two techniques for evaluating the adequacy of the binary glm model used in mlds, based on code in Wood (2006).

Usage

binom.diagnostics(obj, nsim = 200, type = "deviance", no.warn = TRUE)

## S3 method for class 'mlds.diag'
plot(x, alpha = 0.025, breaks = "Sturges", ...)

Arguments

obj list of class ‘mlds’ typically generated by a call to the mlds
nsim integer giving the number of sets of data to simulate
type character indicating type of residuals. Default is deviance residuals. See residuals.glm for other choices
no.warn logical indicating when TRUE (default) to suppress warnings from glm

x list of class ‘mlds.diag’ typically generated by a call to binom.diagnostics

alpha numeric between 0 and 1, the envelope limits for the cdf of the deviance residuals

breaks character or numeric indicating either the method for calculating the number of breaks or the suggested number of breaks to employ. See hist for more details.

... additional parameters specifications for the empirical cdf plot

Details

Wood (2006) describes two diagnostics of the adequacy of a binary glm model based on analyses of residuals (see, p. 115, Exercise 2 and his solution on pp 346-347). The first one compares the empirical cdf of the deviance residuals to a bootstrapped confidence envelope of the curve. The second examines the number of runs in the sorted residuals with those expected on the basis of independence in the residuals, again using a resampling based on the models fitted values. The plot method generates two graphs, the first being the empirical cdf and the envelope. The second is a histogram of the number of runs from the bootstrap procedure with the observed number indicated by a vertical line. Currently, this only works if the ‘glm’ method is used to perform the fit and not the ‘optim’ method

Value

binom.diagnostics returns a list of class ‘mlds.diag’ with components

NumRuns integer vector giving the number of runs obtained for each simulation

resid numeric matrix giving the sorted deviance residuals in each column from each simulation

Obs.resid numeric vector of the sorted observed deviance residuals

ObsRuns integer giving the observed number of runs in the sorted deviance residuals

p numeric giving the proportion of runs in the simulation less than the observed value.

Author(s)

Ken Knoblauch

References


See Also

mlds
## Not run:
```r
data(kk1)
kk1.mlds <- mlds(kk1)
kk1.diag <- binom.diagnostics(kk1.mlds)
plot(kk1.diag)
```
## End(Not run)

### boot.mlds

#### Resampling of an Estimated Difference Scale

**Description**

Using the fitted responses (probabilities) to the difference scale, new responses are generated which permit new bootstrap replications of estimated scales to be generated. The mean scale is useful for evaluating bias and the standard deviation for estimating standard errors of the scale values.

**Usage**

```r
boot.mlds(x, nsim, no.warn = TRUE, …)
boot.mlbs(x, nsim, no.warn = TRUE, …)
```

**Arguments**

- `x`: an object of class ‘mlds’ or ‘mlbs’ depending on which function is called.
- `nsim`: an integer, the number of simulations.
- `no.warn`: logical indicating when TRUE (default) to suppress warnings from `glm`.
- `…`: Additional options passed along to the function `mlds`.

**Details**

Either the scale values (from ‘glm’ method) or the scale values and \( \sigma \) (from ‘optim’ method) permit the fitted probabilities to be estimated. These are used to generate new responses to the quadruples using `rbinom`. The new responses are then used with `mlds` to estimate a bootstrapped scale. This is repeated \( N \) times and stored in the output with the mean and standard deviation of the bootstrapped scales.

**Value**

A list of 4 elements of class ‘mlds.bt’:

- `boot.samp`: A \( p + 1 \times N \) matrix of the bootstrap samples of the scale, where \( p \) is the number of scale values and \( N \) is the number of simulations. The extra value is the bootstrapped \( \sigma \) value.
- `bt.mean`: A vector of length \( p + 1 \) giving the mean of the bootstrap scales.
- `bt.sd`: A vector of length \( p + 1 \) giving the standard deviation of the bootstrap scales.
- `N`: The number of bootstrap simulations.
Author(s)

Kenneth Knoblauch and Laurence T. Maloney

References


See Also

mlds

Examples

data(kk1)
kk1.mlds <- mlds(kk1)
#nsim should be near 10,000 for stability,
# but this will take a little time
boot.mlds(kk1.mlds, 100)

DisplayOneQuad

Helper Functions for Perception of Correlation Difference Scale Experiment

Description

These functions are required by runTriadExperiment and runQuadExperiment to define and display trials. They are provided as examples, and users are expected to define their own functions to define and display stimuli for estimating difference scales along other physical continua. DefineMyScale returns a numeric vector providing the physical levels to be tested (here values of $r$). DisplayOneQuad produces a 2 x 2 graphic of four scatterplots. DisplayOneTriad produces a 1 x 3 graphic of three scatterplots.

Usage

DisplayOneQuad(rr, PntNum = 100, ptSize = 1,
xlim = c(-4, 4), ylim = c(-4, 4))
DisplayOneTriad(rr, PntNum = 100, ptSize = 1,
xlim = c(-4, 4), ylim = c(-4, 4))
DefineMyScale(rr = c(seq(0, 0.9, len = 10), 0.98))
fitted.mlds

Arguments

`rr` vector of numeric or integer. For DisplayOneTrial, a four element vector of integers, providing the indices of the physical scale values to be displayed. For DefineMyScale, a numeric vector, providing the physical scale values to be tested.

`PntNum` numeric giving the number of points to be displayed in each subgraphic.

`ptSize` numeric indicating the size of the points in each graphic, passed to the `cex` argument of `plot`.

`xlim`, `ylim` 2-element numerics to determine range of values plotted on the display.

Value

DisplayOneTriad and DisplayOneQuad are used for their side-effect of producing a stimulus on the display. DefineMyScale outputs a numeric vector of the physical scale. While its use here is quite trivial, in general, it permits the tailoring of the stimulus levels to the particular experiment. In the present case, one could imagine, for example, modifying it so that successive levels were evenly spaced in \( r^2 \) rather than \( r \).

Author(s)

Kenneth Knoblauch and Laurence T. Maloney

See Also

`runTriadExperiment`, `runQuadExperiment`

Examples

```r
## Not run: runQuadExperiment(DisplayTrial = "DisplayOneQuad",
DefineStimuli = "DefineMyScale")
## End(Not run)
```

---

fitted.mlds  

Fitted Responses for a Difference Scale

Description

fitted.mlds returns the fitted responses from an estimated difference scale obtained by mlds.

Usage

```r
## S3 method for class 'mlds'
fitted(object, ...)
```

```r
## S3 method for class 'mlbs'
fitted(object, ...)
```
Get6pts

Arguments

object

object of class ‘mlds’ or ‘mlbs’, typically obtained from the output of `mlds`.

... currently ignored

Value

A numeric vector contained the fitted probabilities to the responses of the observer for each quadruple.

Author(s)

Kenneth Knoblauch

References


See Also

`mlds`, `predict.mlds`

Examples

```r
data(kk1)
data(kk2)
data(kk3)
fitted(mlds(SwapOrder(rbind(kk1, kk2, kk3))))
```

Description

Get6pts enumerates all 6-point conditions from a difference scaling experiment and is used as one of the input arguments for calculating an observer’s likelihood for his performance on 6-point conditions.

Usage

```r
Get6pts(x, nrep, ...)
```

Arguments

x

an object of class ‘mlds’.

nrep

integer indicating how many sessions are in the data set.

... Possibility of sending additional arguments but currently unused.
Get6pts

Details

The 6-point condition is defined on 6-tuples of points, a, b, c, a’, b’, c’, ordered on a physical scale. The condition requires that if the pair \((a, b) > (a', b')\) and the pair \((b, c) > (b', c')\), then \((a, c) > (a', c')\).

Value

A list of three same size data.frames with an attribute, indices that is a three column data.frame with the same number of rows as each of the three data.frames. Each data.frame is of the format from a difference scaling experiment. Same named rows indicate three trials that form a 6-point condition, i.e., given the 6-tuple of stimuli, a, b, c, a’, b’, c’,

A data.frame indicating the trials (a, b) vs (a’, b’)
B data.frame indicating the trials (b, c) vs (b’, c’)
E data.frame indicating the trials (a, c) vs (a’, c’)

The attribute gives the row numbers from which the trials were obtained from the original data.frame.

Note

It is important that the stimuli are in physical order and not the experimental order. If in experimental order, then SwapOrder should be applied first.

Author(s)

Kenneth Knoblauch

References


See Also

SwapOrder, lik6pt, simu.6pt

Examples

data(kk1)
kk.6pt <- Get6pts(mlds(SwapOrder(kk1)), nrep = 1)
ix.mat2df

**Description**

`ix.mat2df` converts a long data.frame that is used as the data argument for `mlds` with method = “glm” to the 5 column format, typically from the results of a difference scaling experiment. The first column is the response and the next four are the ranks of the stimulus levels used in each trial, the 2 pairs. `ix.mat2df` is the inverse of `make.ix.mat`. This form of data.frame is used with the “optim” method of `mlds`.

**Usage**

`ix.mat2df(d)`

**Arguments**

- `d` a data.frame with `p` columns and `n` rows, where `p` is the number of stimulus levels and `n` is the number of trials. The first column is the `resp`, 0 or 1, and the next `p - 1` columns correspond to the incidences of the stimulus levels on each trial, 3 or 4 per row, with the signs `+ - - +`. The column corresponding to the lowest level stimulus is deleted, as its coefficient is constrained to equal 0 in the current parameterization of the problem. Thus, 3 non-zero terms would be of the form `− − +`, indicating the presence of the lowest level stimulus in the trial.

**Value**

A 5 column data.frame which could be coerced to integer.

- `resp` The response, 0 or 1, indicating which pair was chosen
- `S1−S4` The rank of the 4 stimulus levels presented on each trial.

**Author(s)**

Kenneth Knoblauch

**See Also**

- `make.ix.mat`, `mlds`

**Examples**

```r
data(AutumnLab)
ix.mat <- make.ix.mat(AutumnLab)
#orig.df <- ix.mat2df(ix.mat) # should be the same as original
# better to use as.mlds.df as ix.mat2df is deprecated
orig.df <- as.mlds.df(ix.mat)
```
Description

Three data sets for one subject in a sample difference scaling experiment. The stimuli were scatterplots of bivariate Gaussian samples with different correlations. Either the function `runQuadExperiment` or `runTriadExperiment` was used to collect the data.

Usage

```r
data(kk1)
data(kk2)
data(kk3)
data(kktriad)
```

Format

Four data frames, (kk1, kk2, kk3, kktriad) with 330, or in the case of kktriad 165, observations each with the following 5 (4 for kktriad) components.

- `resp` a numeric vector taking on values 0 and 1 indicating responses of observer
- `S1` a numeric vector, rank order of weakest stimulus on the physical scale.
- `S2` a numeric vector, rank order of physical stimulus 2.
- `S3` a numeric vector, rank order of physical stimulus 3.
- `S4` a numeric vector, rank order of strongest stimulus on the physical scale. `kktriad` does not contain this component as it is from an experiment with triads rather than quadruples.

Details

The kk1-3 datasets were generated on three separate days using the function `runQuadExperiment` with `DisplayOneQuad` and `DefineMyScale` to define the stimuli and display them, anti-respectively. The experiments were run on a Macintosh Pro with a 15 inch screen. The observer was seated about 40 cm from the screen. The kktriad data set was generated from one run using the function `runTriadExperiment` with `DisplayOneTriad` to control the display.

References

Examples

```r
data(kk1)
plot(mlds(SwapOrder(kk1)))
# Fit and plot difference scale for first data set kk1,
# using quadruples of stimuli

data(kktriad)
plot(mlds(kktriad), type = "b")
# Fit and plot experimental data
# using triples of stimuli
```

---

**lik6pt**  
*Compute Log Likelihood for 6-point Test*

**Description**

The 6-point test evaluates the validity of the estimated difference scale. Given 6 values, a, b, c, a', b', c', on the stimulus scale, if the pair \((a, b) > (a', b')\) and \((b, c) > (b', c')\) then it must be that \((a, c) > (a', c')\), where the symbol \(>\) is taken here to mean “is judged more different than”. Given the observer’s difference scale and \(\sigma\) estimate, the likelihood of the choices made is calculated based on the link function indicated in the ‘mlds’ object.

**Usage**

```r
lik6pt(x, Six.Pts, ...)
```

**Arguments**

- `x` an object of class ‘mlds’, typically created by `mlds`
- `Six.Pts` a list of 3 data.frames, with names A, B, E. Each data.frame corresponds to a sample from a difference scaling experiment. The corresponding rows of the three data.frames yield the triples of trials that provide a 6-point test. The list can be constructed with the function `GetSixPts`.
- `...` currently unused.

**Value**

Returns the likelihood of the observer’s responses for all of the 6-point conditions from a given data set. As currently implemented, it returns a 1x1 matrix.

**Author(s)**

Kenneth Knoblauch, based on C code by Laurence T. Maloney and J. N. Yang.
logLik.mlds

References


See Also

*Get6pts, mlds, simu.6pt*

Examples

data(kk1)
x.df <- mlds(SwapOrder(kk1))
lik6pt(x.df, Get6pts(x.df, nrep = 1))

---

logLik.mlds

*Compute Log-Likelihood for an mlds object*

Description

This function provides a method for extracting the log likelihood from an object of class ‘mlds’.

Usage

```r
## S3 method for class 'mlds'
logLik(object, ...)
## S3 method for class 'mlbs'
logLik(object, ...)
```

Arguments

- `object`: an object of class ‘mlds’ or ‘mlbs’ typically from a call to `mlds`
- `...`: for passing additional parameters, but is currently not used.

Value

An object of class ‘logLik’ whose value is the logarithm of the likelihood with attribute `df` providing the degrees of freedom

Author(s)

Kenneth Knoblauch
Examples

```r
data(kk1)
logLik(mlds(SwapOrder(kk1)))
```

Description

`make.ix.mat` generates a $n \times p$ matrix from the $n \times 5$ column data.frame storing the results of a difference scaling experiment, where $p$ is the number of stimulus levels tested and $n$ is the number of trials. The first column is the response (0 or 1), and the $p - 1$ succeeding columns code covariates for all but the first stimulus level, which is constrained to be 0. These columns take the value 0 unless the stimulus level was in the trial, in which case they take, in order, the values, 1, -1, -1, 1.

Usage

```r
make.ix.mat(data, xi = NULL, ...)
```

Arguments

- `data`: a 5 column data.frame. The first column is the `resp` and the next 4 columns the index of the stimulus level, 1 to $p$, labelled S1-S4.
- `xi`: an integer indicating the number of stimulus levels tested. If this is NULL, it is determined from the maximum value in `data`.
- `...`: Other arguments, not used for the moment.

Details

To fit a difference scale using `mlds` and `method = "glm"`, each stimulus level is treated as a covariate taking on the values 0, if it was not present in the trial, or -1 or 1, the latter two depending on the ordinal stimulus level within the trial. This is a helper function to transform the typical 5 column data.frame from a difference scaling experiment, indicating the response and the 4 stimulus levels, to one in the format described above. Matrices of this form can also be used as `newdata` for the `predict` method. This is exploited in the function `like6pt`. It is here that the argument `xi` is necessary since the `data.frame` for the first of the 6-point comparisons does not contain the highest level of the scale and so needs to be specified so that the `data.frame` conforms with that used to generate the ‘`mlds’ object.

Value

A data.frame with $n$ rows and $p$ columns.

- `resp`: The `resp` (0, 1) is coded in the first column. This could be logical, instead.
- `stim.2--stim.p`: Columns 2 through $p$ code the presence and absence of the stimulus on a trial. If the stimulus is present, then the value is -1 or 1 as a function of the contribution of the stimulus level to the decision variable.
Note

In the current parameterization, the coefficient of the initial stimulus level is constrained to 0. Thus, the column corresponding to this level is left-out of the data.frame. For trials in which this stimulus is present, the non-zero elements are (-1, -1, 1), in that order.

Author(s)

Kenneth Knoblauch

See Also

ix.mat2df, mlds.lik6pt, glm

Examples

data(AutumnLab)
make.ix.mat(AutumnLab)
mlds(AutumnLab, c(1, seq(6, 30, 3)))

mlds

Fit Difference Scale by Maximum Likelihood

Description

Generic function mlds uses different methods to fit the results of a difference scaling experiment either using glm (Generalized Linear Model), by direct maximization of the likelihood using optim or by maximizing the likelihood with respect to a function of the stimulus dimension specified by a one sided formula.

Usage

mlds(x, ...)

## S3 method for class 'mlds.df'
mlds(x, stimulus = NULL, method = "glm",
lnk = "probit", opt.meth = "BFGS", glm.meth = "glm.fit",
opt.init = NULL, control = glm.control(maxit = 50000, epsilon = 1e-14),
... )

## S3 method for class 'mlbs.df'
mlds(x, stimulus = NULL, method = "glm",
lnk = "probit",
control = glm.control(maxit = 50000, epsilon = 1e-14),
glm.meth = "glm.fit",
... )

## S3 method for class 'data.frame'

mlds(x, ...)  

## S3 method for class 'formula'
mlds(x, p, data, stimulus = NULL,  
lnk = "probit", opt.meth = "BFGS",  
control = list(maxit = 50000, reltol = 1e-14), ...)  

**Arguments**

- **x**: For comparisons of two pairs of stimuli, when the method is specified as ‘glm’ or ‘optim’ a data frame with 5 columns giving the response and the ranks of the stimulus levels for each trial, or an object of class ‘mlds.df’ which also contains additional information as attributes. For comparisons of triples of stimuli, only the method ‘glm’ is currently defined. The object can be a data frame of 4 columns with the first specifying the response and the other 3 the stimulus level ranks, or an object of class ‘mlbs.df’, which contains additional attributes. It can also be a one-side formula with parameters p and stimulus variable sx that gives a parametric formula to fit to the data for the `formula` method.

- **data**: A data frame with 4 or 5 columns giving the response and the ranks of the stimulus levels for each trial, or an object of class ‘mlbs.df’ or ‘mlds.df’, respectively, which also contains additional information as attributes, required when the ‘formula’ method is used.

- **p**: numeric vector of parameters of length one greater than the number of parameters in the `formula` argument that specifies initial values for the parameters. The extra parameter, specified last, is the initial estimate of sigma.

- **stimulus**: A numeric vector that contains the physical stimulus levels used in the experiment. If `data` is of class ‘mlds.df’, this information is included as an attribute. If NULL, a sequence of 1 : n is used, where n is the number of stimulus levels, deduced from the highest rank in `data`.

- **method**: character, taking the value of “glm” or “optim”. Default is “glm”.

- **lnk**: character indicating either one of the built-in links for the binomial family or a user defined link of class ‘link-glm’. See `family` and `make.link`. Default is “probit”.

- **opt.meth**: If method = “optim”, the method used by `optim` can be specified. Defaults to “BFGS”.

- **opt.init**: Vector of numeric giving initial values which must be provided if you specify the “optim” method.

- **control**: A list of control values for either `glm` or `optim`. Since the method defaults to “glm”, the default is a glm list but should be changed if the “optim” method is chosen.

- **glm.meth**: the method to be used in fitting the model, only when method = glm. The default value is “glm.fit”. See `glm` for further details.

- **...**: Additional arguments passed along to `glm` or `optim`.

Details

Observers are presented with either triples or pairs of pairs of stimuli, distributed along a physical stimulus axis. For example, for stimuli $a, b, c$ with $a < b < c$, they see the triple $a, b, c$, or for stimuli $a, b, c, d$ with $a < b < c < d$, they see the pairs $(a, b)$ and $(c, d)$. For each trial, they make a judgement respectively as to whether the difference between stimuli 1 and 2 is greater or not that between stimuli 2 and 3 or the elements of pair 1 is greater or not than the difference between the elements of pair 2. From a large number of trials on different quadruples, mlds estimates numbers, $\Psi_1, \ldots, \Psi_n$, by maximum likelihood such that $(\Psi_d - \Psi_c) > (\Psi_b - \Psi_a)$ when the observer chooses pair 2, and pair 1, otherwise.

If there are $p$ stimulus levels tested, then $p - 1$ coefficients are estimated. The “glm” method constrains the lowest estimated value, $\Psi_1 = 0$, while the “optim” method constrains the lowest and highest values to be 0 and 1, respectively. The “optim” method estimates an additional scale parameter, $\sigma$, whereas this value is fixed at 1.0 for the “glm” method. In principle, the scales from the two methods are related by

$$1/\sigma_o = \max(\Psi_g)$$

where $\sigma_o$ is $\sigma$ estimated with the “optim” method and $\Psi_g$ corresponds to the perceptual scale values estimated with the “glm” method. The equality may not be exact as the “optim” method prevents the selection of values outside of the interval $[0, 1]$ whereas the “glm” method does not.

Value

A list of class ‘mlds’ whose components depend on whether the method was specified as ‘glm’, ‘optim’ with the default method, or the formula method was used,

- pscale: A numeric vector of the estimated difference scale.
- stimulus: The physical stimulus levels
- sigma: The scale estimate, always 1.0 for ‘glm’
- method: The fitting method
- link: The binomial link specified, default ‘probit’
- obj: For method ‘glm’, an object of class ‘glm’ resulting from the fit.
- logLik: for method ‘optim’, the logarithm of likelihood at convergence
- hess: for method ‘optim’, the Hessian matrix at convergence
- data: For method ‘optim’, the data.frame or ‘mlds.df’ entered as an argument.
- conv: For method ‘optim’, a code indicating whether optim converged or not. See optim.
- par: For ‘formula’ method, the parameters estimated.
- formula: The one-sided formula specified with the ‘method’.
- func: For ‘formula’ method, a function obtained from the one-sided formula.

Note

The glm method often generates warnings that fitted probabilities are 0 or 1. This does not usually affect the values of the estimated scale. However, it may be wise to check the results with the optim method and obtain standard errors from a bootstrap method (see boot.mlds). The warnings will often disappear if the link is modified or more data are obtained.
Author(s)

Kenneth Knoblauch and Laurence T. Maloney

References


See Also

glm, optim

Examples

data(AutumnLab)
# Note the warnings generated by glm method
x.mlds <- mlds(AutumnLab)
summary(x.mlds)
y.mlds <- mlds(AutumnLab, method = "optim", opt.init = c(seq(0, 1, len = 10), 0.16))
summary(y.mlds)
plot(x.mlds)
# How sigma relates the scales obtained by the 2 different methods.
lines(y.mlds$stimulus, y.mlds$pscale/y.mlds$sigma)

# Example with triads
data(kktriad)
kk.mlds <- mlds(kktriad)
plot(kk.mlds, type = "b")

# An example using the formula method
data(kk1)
# with one parameter
kk.frm1 <- mlds(~ sx^p, p = c(3, 0.02), data = kk1)
# with two parameters
p = c(0.9, 0.3, 0.2), data = kk1)

plot.mlds

Plot a mlds Object

Description

Plots the difference scale as a function of stimulus level.
Usage

## S3 method for class 'mlds'
plot(x, standard.scale = FALSE, SD.scale = FALSE, ...)
## S3 method for class 'mlds'
lines(x, standard.scale = FALSE, SD.scale = FALSE, ...)
## S3 method for class 'mlds'
points(x, standard.scale = FALSE, SD.scale = FALSE, ...)

## S3 method for class 'mlbs'
plot(x, standard.scale = FALSE, SD.scale = FALSE, ...)
## S3 method for class 'mlbs'
lines(x, standard.scale = FALSE, SD.scale = FALSE, ...)
## S3 method for class 'mlbs'
points(x, standard.scale = FALSE, SD.scale = FALSE, ...)

Arguments

- **x**
  - mlds or mlbs object, typically result of mlds
- **standard.scale**
  - logical indicating whether the plotted difference scale should be normalized to maximum value = 1
- **SD.scale**
  - logical indicating whether to plot difference scale in units of d'. Ignored if standard.scale = TRUE.
- **...**
  - other parameters to be passed through to the plotting function

Author(s)

Kenneth Knoblauch

See Also

mlds

Examples

data(kk1)
plot(mlds(SwapOrder(kk1)))
lines(mlds(SwapOrder(kk1)))

---

**pmc**

Proportion of Misclassifications According to an Estimated MLDS Fit

Description

pmc calculates the proportion of the observer’s responses that are misclassifications on the basis of the estimated MLDS.
predict.mlds

Usage

pmc(x, ...)

Arguments

x  object of class 'mlds'.
...
  currently unused.

Value

numeric indicating the proportion of misclassified trials on the basis of the estimated scale.

Author(s)

Kenneth Knoblauch

References


See Also

mlds

Examples

data(kk1)
kk1.mlds <- mlds(SwapOrder(kk1))
pmc(kk1.mlds)

Usage

## S3 method for class 'mlds'
predict(object, newdata = NULL, type = "link", ...)

## S3 method for class 'mlbs'
predict(object, newdata = NULL, type = "link", ...)
Arguments

- **object**: object of class ‘mlds’ or ‘mlbs’, typically from the output of `mlds`
- **newdata**: A data.frame or object of class ‘mlbs.df’ with 4 columns or ‘mlds.df’ with 5 columns, corresponding to the response at each trial and the ranks of, respectively, the triple or quadruple of stimuli presented.
- **type**: character indicating scale on which predictions should be made, “link”, the default, on the scale of the linear predictor or “response”, on the response scale.
- **...**: When `obj$method = "glm"`, these are additional arguments passed to `predict.glm`.

Details

The `newdata` argument is needed principally for the 6-point test (see `lik6pt`), to extract the estimated probabilities for the subsets of the original data that form valid 6-point tests.

Value

A numeric vector of the predicted values either on the scale of the linear predictor or on the response scale.

Author(s)

Kenneth Knoblauch

References


See Also

`mlds`, `lik6pt`

Examples

```r
data(kk1)
kk1.mlds <- mlds(SwapOrder(kk1))
predict(kk1.mlds, type = "response")
```

---

**print.mlds**  
*Difference Scale default print statement*

Description

This is the default print statement for a ‘mlbs’ or ‘mlds’ object. It displays the difference scale as a named vector, with the names corresponding to the stimulus levels and the value of `sigma`. 
### Usage

```r
## S3 method for class 'mlds'
print(x, digits = max(3,getOption("digits") - 4), ...)

## S3 method for class 'mlbs'
print(x, digits = max(3,getOption("digits") - 4), ...)
```

### Arguments

- `x`: an object of class ‘mlbs’ or ‘mlds’, typically from a call to `mlds`.
- `digits`: number of digits to display in the output.
- `...`: additional arguments to be passed to the default method.

### Author(s)

Kenneth Knoblauch

### See Also

- `mlds`

### Examples

```r
data(kk1)
print(mlds(SwapOrder(kk1)))
```

---

### rbind.mlds.df

`rbind.mlds.df` **Concatenate Objects of Class 'mlbs.df' or 'mlds.df' by Row**

### Description

Concatenate the data.frame of ‘mlbs.df’ or ‘mlds.df’ objects by row and concatenate the ‘invord’ attributes, also.

### Usage

```r
## S3 method for class 'mlds.df'
rbind(...)

## S3 method for class 'mlbs.df'
rbind(...)
Rbind(...) 
```

### Arguments

- `...`: Objects of class ‘mlbs.df’ or ‘mlds.df’.

Details

Uses `rbind.data.frame` to concatenate the `data.frame` component of several `mlbs.df` or `mlds.df` objects and then concatenates their `invord` attributes, as well. `Rbind` will work, too, but is deprecated.

Value

An object of class `mlbs.df` or `mlds.df` that is composed of data from several experiments.

Author(s)

Kenneth Knoblauch

Examples

data(kk1)
data(kk2)data(kk3)
kk <- rbind(kk1, kk2, kk3)
nrow(kk1)
nrow(kk)
length(attr(kk1, "invord"))
length(attr(kk, "invord"))

---

runQuadExperiment  

*Start and run a Difference Scale Experiment*

Description

Runs a difference scaling experiment displaying stimuli with the function `DisplayTrial` defined by the function `DefineStimuli`.

Usage

```r
runQuadExperiment(DisplayTrial, DefineStimuli, NumTrials = NULL,
                   DisplaySize = 7.5, aspect = 1, ...)
runTriadExperiment(DisplayTrial, DefineStimuli, NumTrials = NULL,
                   DisplaySize = 3.5, aspect = 1, ...)
runSampleExperiment(DisplayTrial, DefineStimuli)
```

Arguments

- **DisplayTrial** character giving the name of a function to display a trial with triple or quadruple of stimuli.
- **DefineStimuli** character giving the name of a function that defines the set of stimuli from which triples or quadruples are drawn.
- **DisplaySize** numeric giving the overall size of the display on the screen.
SimMLDS

Simulate Output of MLDS Experiment

Description

Given a block of trials of an MLDS experiment, an underlying response function and the judgment variability, simulate the response of an observer.

Usage

SimMLDS(Trials, Scale, Sigma, n = 1)

NumTrials  integer giving the number of trials to display. If NULL (the default), the number is computed on the basis of the size of stimulus set designated by DefineStimuli.

... currently unused

aspect  numeric giving the height/width ratio of the display on the screen.

Details

These functions are to demonstrate how to run a difference scaling experiment. Helper functions DisplayOneQuad or DisplayOneTriad and DefineMyScale permit running the perception of correlation experiment, sample data from which are provided as datasets, (see kk). runSampleExperiment is defunct and replaced by these functions.

Value

An object of class ‘mlds.df’ or ‘mlbs.df’, depending on the experiment run, is returned. Each inherits from ‘data.frame’ and has attributes ‘stimulus’ and ‘invord’. See kk for an example.

Author(s)

Kenneth Knoblauch and Laurence T. Maloney

Examples

##This will start a 330 trial interactive experiment
## of quadruples
## Not run:
runQuadExperiment("DisplayOneQuad", "DefineMyScale")

## End(Not run)
##This will run 10 trials of an interactive experiment
## of triads
## Not run:
runTriadExperiment("DisplayOneTriad", "DefineMyScale", NumTrials = 10)

## End(Not run)
SimMLDS

Arguments

- **Trials**: an N by 4 or 3 matrix or data frame of integers indicating the n trials of an MLDS experiment. The columns indicate the indices of the stimuli presented on a trial, 4 for an experiment with quadruples and 3 for triads. A data frame for this argument is most easily generated with the `combn` function.

- **Scale**: a vector of values indicating the underlying responses of the simulated observer for each stimulus level. The length of this vector should equal the largest integer in Trials.

- **Sigma**: a vector of length 1 indicating the judgment standard deviation of the simulated observer.

- **n**: integer giving number of simulated data sets to return

Details

Given a data frame of indices to the responses associated with stimulus levels and the judgment variability, the function returns the results of 1 or multiple MLDS experiments, either with triads or quads, depending on the number of columns in the data frame.

Value

If the argument `n` is set to 1 (default), an object of class ‘mlds.df’ or ‘mlbs.df’ with simulated responses. If `n` is greater than 1, a list of such objects is returned.

Author(s)

Kenneth Knoblauch and Laurence T. Maloney

References


See Also

see also `boot.mlds`

Examples

```r
Tr <- t(combn(10, 4))
Sc <- seq(0, 1, len = 11)^2
Sig <- 0.2
sim.lst <- SimMLDS(Tr, Sc, Sig, n = 10)
sim.res <- sapply(sim.lst, mlds)
```
Perform Bootstrap Test on 6-point Likelihood for MLDS FIT

Description

Using the fitted responses (probabilities) to the difference scale, new responses are generated which permit new 6-point likelihoods to be calculated. The distribution of a large number of such likelihoods can be compared with that obtained from the observed responses to evaluate the internal consistency of the estimated scale.

Usage

```r
simu.6pt(obj, nsim = 1, nrep, no.warn = TRUE)
```

Arguments

- **obj**: object of class `mlds`
- **nsim**: integer indicating number of bootstrap trials.
- **nrep**: integer indicating how many sessions with are in the data set.
- **no.warn**: logical indicating when TRUE (default) to suppress warnings from `glm`.

Value

LIST with 4 components

- **boot.samp**: vector of numeric giving the log likelihood for the 6-point test for each simulation.
- **lik6pt**: numeric indicating the log likelihood for the 6-point test on the original data
- **p**: proportion of simulations on which the simulated log likelihood was higher than that obtained from the original sample.
- **N**: numeric indicating the number of simulations. It should be the length of `boot.samp`.

Author(s)

Kenneth Knoblauch and Laurence T. Maloney

References


See Also

`mlds, lik6pt`
Examples

```r
data(kk1)
x.mlds <- mlds(SwapOrder(kk1))
# nsim should be near 10,000 for stability,  
# but this will take a little time
simu.6pt(x.mlds, 100, nrep = 1)
```

summary.mlds  
Summary for a mlds fit

Description

Takes a fitted ‘mlbs’ or ‘mlds’ object and produces a summary of it

Usage

```r
## S3 method for class 'mlbs'
summary(object, digits = max(3, getOption("digits") - 4), ...)

## S3 method for class 'summary.mlbs'
print(x, digits = max(3, getOption("digits") - 4), ...)

## S3 method for class 'mlds'
summary(object, digits = max(3, getOption("digits") - 4), ...)

## S3 method for class 'summary.mlds'
print(x, digits = max(3, getOption("digits") - 4), ...)
```

Arguments

- `object`: an object of class ‘mlbs’ or ‘mlds’ typically produced by a call to `mlds`.
- `digits`: The number of digits to display.
- `x`: An object of class ‘summary.mlbs’.
- `...`: Additional arguments passed to the default print method. Only effects the output of `pscale`.

Details

Displays summary information from a ‘mlds’ object.

Value

A list of 5 elements

- `pscale`: A named vector indicating the difference scale. The names are the stimulus levels.
- `sigma`: The estimate of the scale parameter. For method = “glm”, this is always 1.
logLik  The logarithm of likelihood.
method  The fitting method used, either “glm” or “optim”.
link    The link used for the binomial family.

Normally, print.summary.mlds is not meant to be called directly by the user.

Author(s)
Kenneth Knoblauch

See Also
mlds

Examples

data(kk1)
kk1.mlds <- mlds(SwapOrder(kk1))
summary(kk1.mlds)

summary.mlds.bt  Method to Extract Bootstrap Values for MLDS Scale Values

Description
Extracts the means and standard deviations of the bootstrapped scale values from an MLDS experiment.

Usage

## S3 method for class 'mlds.bt'
summary(object, standard.scale = TRUE, sigma = FALSE, ...)

Arguments

object  object of class 'mlds.bt', typically obtained from running boot.mlds or boot.mlbs on an object created by mlds.
standard.scale  logical, if TRUE (default), the values are returned on the standard scale (0, 1). Otherwise, the values are returned in unnormalized units.
sigma  If TRUE and standard.scale = TRUE, the bootstrapped sigma value is printed out and its standard deviation.
...  additional arguments to summary, currently unused.
Details

Returns means and standard deviations bootstrapped values for an object of class ‘mlds.bt’. By default the values are on the standard scale but may be renormalized by the standard deviation of each bootstrap run before taking the means and standard deviations.

Value

A two column matrix is returned of the bootstrap means and standard deviations in columns 1 and 2, respectively.

Author(s)

Kenneth Knoblauch

See Also

boot.mlds

Examples

data(kk1)
kk.mlds <- mlds(kk1)
kk.bt <- boot.mlds(kk.mlds, nsim = 10)
summary(kk.bt)

SwapOrder

Order Stimuli and Adjust Responses from Difference Scaling data.frame

Description

Invert the order of the stimuli on each trial so that they are in physical order and not the presentation order and adjust responses, accordingly to reflect whether the lower or higher stimulus pair was selected or the inverse.

Usage

SwapOrder(data)

Arguments

data

data.frame or ‘mlds.df’ object with five columns, resp, S1, S2, S3 and S4. resp is the binary response of the observer (it could be a logical). The other four indicate the four stimulus levels presented on a given trial. In principle, this function can be applied when the order reflects the position of the pairs and not their physical ordering.
Details

Pairs of stimuli in a difference scaling experiment are often presented on top and bottom and the responses, 0 and 1, refer to bottom and top pair, respectively. `SwapOrder` modifies the data.frame so that the stimuli are ordered by physical level and the responses, 0 and 1, are modified to refer to the lower and upper pair on the physical scale, respectively. If the object inherits also from class `mlds.df`, it has an `invord` attribute that is a logical vector of length the number of rows in the data.frame. It indicates the trials on which the higher stimulus levels were presented on the bottom. If this attribute is present, it is used to modify the data.frame. In this case, the function acts as its own inverse.

Value

A data.frame of the same format as the input data.frame but with the stimulus order modified and the response inverted on those trials on which the higher physical level stimulus were on the bottom. If the order reflects the position, it is modified to reflect the stimulus level and vice versa.

Note

Storing the physical position of the stimuli allows the original configuration not to be lost when re-ordering the stimuli to reflect physical stimulus level. The `invord` attribute could, in principle, be used, also, to test for an influence of or bias related to the physical position.

Author(s)

Kenneth Knoblauch

Examples

```r
data(kk1)
kk1.swo <- SwapOrder(kk1)
```

---

**Description**

The data set was obtained from an experiment in which observers judged the differences between pairs of image pairs containing transparent pebble-shaped objects. The refractive indice (RI) of the transparent material of each object was varied systematically across the four images.

**Usage**

```r
data(Transparency)
```
**Format**

A data frame with 2520 observations on the following 6 variables.

- `resp` a numeric vector taking on values of 0 and 1 indicating responses of observer.
- `S1` a numeric vector, rank order of weakest stimulus on the physical scale for a given trial.
- `S2` a numeric vector, rank order of physical stimulus 2 for the trial.
- `S3` a numeric vector, rank order of physical stimulus 3 for the trial.
- `S4` a numeric vector, rank order of strongest stimulus on the physical scale for a given trial.
- `Obs` a factor identifying observers with levels O1 ... O6.

**Details**

The physical scale is RI and the psychophysical scale is estimated using MLDS. Observers were not told to base their judgments on RI (RI was not mentioned in instructions) but to judge apparent differences. Fleming et al (2007) conjectured that observers used a measure of the distortion of the background as seen through each transparent object in judging differences.

**References**


**Examples**

```r
Trsp.mlds <- mlds(as.mlds.df(Transparency[, -6], st = attr(Transparency, "stimulus")))
plot(Trsp.mlds, xlab = "Index of Refraction", ylab = "Difference Scale", ylim = c(0, 20), lwd = 3)

Trsp.Obs <- sapply(levels(Transparency$Obs), function(obs)
  mlds(as.mlds.df(subset(Transparency, Obs == obs, select = 1:5), st = attr(Transparency, "stimulus"))),
  simplify = FALSE, USE.NAMES = TRUE)
invisible(sapply(Trsp.Obs, lines, type = "b"))
```
Index

* datagen
  boot.mlds, 7
  SimMLDS, 26
  simu.6pt, 28

* datasets
  AutumnLab, 4
  kk, 13
  Transparency, 32

* hplot
  binom.diagnostics, 5
  plot.mlds, 20

* manip
  as.mlds.df, 3
  Get6pts, 10
  ix.mat2df, 12
  make.ix.mat, 16
  rbind.mlds.df, 24
  SwapOrder, 31

* methods
  fitted.mlds, 9
  predict.mlds, 22
  summary.mlds, 29
  summary.mlds.bt, 30

* misc
  boot.mlds, 7
  DisplayOneQuad, 8
  runQuadExperiment, 25
  simu.6pt, 28

* models
  binom.diagnostics, 5
  mlds, 17

* nonlinear
  mlds, 17

* package
  MLDS-package, 2

* print
  print.mlds, 23
  summary.mlds, 29

* regression
  mlds, 17

* univar
  lik6pt, 14
  logLik.mlds, 15
  pmc, 21

as.mlbs.df(as.mlds.df), 3
as.mlds.df, 3
AutumnLab, 4

binom.diagnostics, 5
boot.mlbs(boot.mlds), 7
boot.mlds, 7, 19, 27, 31

DefineMyScale(DisplayOneQuad), 8
df2mlds.df(as.mlds.df), 3
DisplayOneQuad, 8
DisplayOneTriad(DisplayOneQuad), 8

family, 18
fitted.mlbs(fitted.mlds), 9
fitted.mlds, 9

Get6pts, 10, 15
glm, 17, 18, 20

hist, 6

ix.mat2df, 12, 17

kk, 13, 26
kk1(kk), 13
kk2(kk), 13
kk3(kk), 13
kktriad(kk), 13

lik6pt, 11, 14, 17, 23, 28
lines.mlbs(plot.mlds), 20
lines.mlbs(plot.mlds), 20
logLik.mlbs(logLik.mlds), 15
logLik.mlds, 15
make.ix.mat, 12, 16
make.link, 18
MLDS (MLDS-package), 2
mlds, 4, 6, 8, 10, 12, 15, 17, 21–24, 28, 30
MLDS-package, 2
optim, 19, 20
plot.mlbs (plot.mlds), 20
plot.mlds, 20
plot.mlds.diag (binom.diagnostics), 5
pmc, 21
points.mlbs (plot.mlds), 20
points.mlds (plot.mlds), 20
predict.mlbs (predict.mlds), 22
predict.mlds, 10, 22
print.mlbs (print.mlds), 23
print.mlds, 23
print.summary.mlbs (summary.mlds), 29
print.summary.mlds (summary.mlds), 29
Rbind (rbind.mlds.df), 24
rbind.mlds.df (rbind.mlds.df), 24
rbind.mlds.df, 24
residuals.glm, 5
runQuadExperiment, 9, 25
runSampleExperiment
(runQuadExperiment), 25
runTriadExperiment, 9
runTriadExperiment (runQuadExperiment), 25
SimMLDS, 26
simu.6pt, 11, 15, 28
summary.mlbs (summary.mlds), 29
summary.mlds, 29
summary.mlds.bt, 30
SwapOrder, 11, 31
Transparency, 32