# Package 'MLCM'

January 11, 2022

**Type** Package

**Title** Maximum Likelihood Conjoint Measurement

**Version** 0.4.3

**Date** 2022-01-03

**Depends** R (>= 3.5), graphics, stats, utils, base

**Description** Conjoint measurement is a psychophysical procedure in which stimulus pairs are presented that vary along 2 or more dimensions and the observer is required to compare the stimuli along one of them. This package contains functions to estimate the contribution of the n scales to the judgment by a maximum likelihood method under several hypotheses of how the perceptual dimensions interact. Reference: Knoblauch & Maloney (2012) ``Modeling Psychophysical Data in R". <doi:10.1007/978-1-4614-4475-6>.

**License** GPL (>= 2)

**LazyData** yes

**Author** Ken Knoblauch [aut],
Laurence T. Maloney [aut],
Guillermo Aguilar [aut, cre]

**Maintainer** Guillermo Aguilar <guillermo.aguilar@mail.tu-berlin.de>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-01-11 17:22:43 UTC

## R topics documented:

---

`MLCM-package` *Maximum Likelihood Conjoint Measurement*

---

## Description

Estimate perceptual scales from data collected in a conjoint measurement experiment by maximum likelihood. Data for conjoint measurement are typically collected using a psychophysical procedure. The stimuli vary along $n \geq 2$ dimensions. The observer views pairs of stimuli and judges which stimulus of each pair is higher on a specified dimension. For example, stimuli may be goods baskets containing amounts of milk and honey (dimensions) and the subject may order each pair of baskets by subjective desirability. This package contains functions to estimate the additive contribution of the $n$ scales to the judgment by a maximum likelihood method under several hypotheses of how the perceptual dimensions interact.

## Details

| | |
|---|---|
| Package: | MLCM |
| Type: | Package |
| Version: | 0.4.3 |
| Date: | 2020-01-11 |
| License: | GPL |
| LazyLoad: | yes |
| LazyData: | yes |

Index:

`BumpyGlossy Dataset: Conjoint Measurement for Bumpiness and Glossiness (Ho et al. 2008)`

`Texture Dataset: 3-way conjoint Measurement for Texture (Sun et. al, 2021)`

`MLCM-package Estimate perceptual scales from a conjoint measurement experiment by maximum likelihood`

`anova.mlcm Likelihood ratio tests for Maximum Likelihood Conjoint Measurement models`

`logLik.mlcm Calculate log likelihood for Conjoint Measurement models`

`make.wide Create data frame for Fitting Conjoint Measurement Scale by glm`

`mlcm` Fit Conjoint Measurement Models by Maximum Likelihood

`plot.mlcm` plot method for Maximum Likelihood Conjoint Measurement models

`print.mlcm` print method for Maximum Likelihood Conjoint Measurement models

`print.summary.mlcm` print method for summary of Maximum Likelihood Conjoint Measurement models

`summary.mlcm` summary method for Maximum Likelihood Conjoint Measurement models

## Author(s)

Kenneth Knoblauch

Maintainers: Guillermo Aguilar <guillermo.aguilar@mail.tu-berlin.de>, Ken Knoblauch <ken.knoblauch@inserm.fr>

## References

Luce, R. D., and Tukey, J. W. (1964). Simultaneous conjoint measurement. *Journal of Mathematical Psychology*, **1**, 1–27.

Krantz, D. H., Luce, R. D., Suppes, P., and Tversky, A. (1971). *Foundations of Measurement, Vol. 1: Additive and Polynomial Representations*. New York: Academic Press.

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

`glm`

## Examples

```
bg.acm <- mlcm(BumpyGlossy)
plot(bg.acm, pch = 21:22, bg = c("blue", "red"), col = "black",
ylab = "Contributions to Perceived Bumpiness")
```

---

# `anova.mlcm` *Analysis of Deviance for Maximum Likelihood Conjoint Measurement Model Fits*

---

## Description

Compute an analysis of deviance table for one or more maximum likelihood conjoint measurement model fits.

## Usage

```
## S3 method for class 'mlcm'
anova(object, ..., dispersion = NULL, test = NULL)
```

## Arguments

| | |
|---|---|
| `object, ...` | objects of class `mlcm`, typically the result of a call to `mlcm` |
| `dispersion` | the dispersion parameter for the fitting family. By default, it is obtained from the object(s) |
| `test` | a character string (partially) matching one of "Chisq", "F", or "Cp". See `stat.anova`. Normally, "Chisq" is the appropriate value, here. |

## Details

See `anova.glm` for details. In brief, specifying a single object, results in the display of a sequential analysis of deviance table for that model. Specifying several objects, a table indicating the results of the likelihood ratio tests between successive models is displayed. The models must be nested and fit to the same data set. One can mix a `formula` method model with a `glm` model, but not more than one comparison between a pair of such models at a time.

## Value

An object of class "anova" inheriting from class "data.frame".

## Warning

see section Warnings in `anova` for warnings.

## Author(s)

Kenneth Knoblauch

## References

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

`anova.glm`, `anova`, `glm`

## Examples

```
bg.add <- mlcm(BumpyGlossy)
bg.ind <- mlcm(BumpyGlossy, model = "ind", whichdim = 2)
bg.full <- mlcm(BumpyGlossy, model = "full")

anova(bg.ind, bg.add, bg.full, test = "Chisq")
```

## as.mlcm.df *Coerce data frame to mlcm.df*

### Description

Coerce a data frame from an MLCM experiment to an object of class `mlcm.df`.

### Usage

```
as.mlcm.df(d, ...)
```

### Arguments

| | |
|---|---|
| `d` | object of class `data.frame` typically obtained from an MLCM experiment. It should have an odd number of columns and no less than 5. |
| `...` | Currently unused. |

### Details

The first column should be named `Resp`. Subsequent columns contain integer indices to the stimulus levels of the stimuli presented for each trial. As there are two stimuli per trial and at least 2 dimensions tested per experiment, the minimun number of total columns will be 5.

### Value

Returns a data frame of class `mlcm.df`.

### Author(s)

Kenneth Knoblauch

### See Also

See Also as `mlcm`,

## binom.diagnostics *Diagnostics for Binary GLM*

### Description

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

## Usage

```
binom.diagnostics(obj, nsim = 200, type = "deviance")

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

## Arguments

| | |
|---|---|
| obj | list of class 'mlcm' typically generated by a call to the mlcm |
| 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 |
| x | list of class 'mlcm.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 'mlcm.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

Wood, SN *Generalized Additive Models: An Introduction with R*, Chapman \& Hall/CRC, 2006

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

mlcm

## Examples

```
## Not run:
data(BumpyGlossy)
bg.mlcm <- mlcm(BumpyGlossy)
bg.diag <- binom.diagnostics(bg.mlcm)
plot(bg.diag)

## End(Not run)
```

---

# boot.mlcm *Resampling of an Estimated Conjoint Measurement Scale*

---

## Description

Using the `fitted` responses (probabilities) to the conjoint measurement 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

```
boot.mlcm(x, nsim, ...)
```

## Arguments

| | |
|---|---|
| `x` | an object of class 'mlcm' |
| `nsim` | an integer, the number of simulations. |
| `...` | Additional options passed along to the function `mlcm`. |

## Details

The scale values (from 'glm' method) permit the fitted probabilities to be estimated. These are used to generate new responses to the stimulus pairs using `rbinom`. The new responses are then used with `mlcm` 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:

| | |
|---|---|
| `boot.samp` | A $p$ x $N$ matrix of the bootstrap samples of the scale, where p is the number of scale values and N is the number of simulations. |
| `bt.mean` | A vector of length $p$ giving the mean of the bootstrap scales. |
| `bt.sd` | A vector of length $p$ giving the standard deviation of the boostrap scales. |
| `N` | The number of bootstrap simulations. |

## Author(s)

Kenneth Knoblauch and Laurence T. Maloney

## References

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

`mlcm`

## Examples

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

---

`BumpyGlossy` *Conjoint Measurement Data for Bumpiness and Glossiness*

---

## Description

Data sets from two observers who were asked to judge which of two samples appeared bumpier (`BumpyGlossy`) or glossier (`GlossyBumpy`) when the two attributes were covaried simultaneously. The data sets are of class 'mlcm.df' and 'data.frame'.

## Usage

```
data(BumpyGlossy)
data(GlossyBumpy)
```

## Format

A data frame with 975 observations on the following 5 variables.

`Resp` a factor with levels `0` `1` indicating whether the first or second sample was judged to be bumpier (glossier).

`G1` a numeric vector indicating the indexed level of glossiness of the first sample.

`G2` a numeric vector indicating the indexed level of glossiness of the second sample.

`B1` a numeric vector indicating the indexed level of bumpiness of the first sample.

`B2` a numeric vector indicating the indexed level of bumpiness of the second sample.

## Details

Synthetic images of textures composed of random arrays of overlapping balls were created with varying degress of bumpiness and glossiness of the surfaces. In separate experiments observers judged which of a pair of images appeared bumpier or glossier. These data permit evaluating whether the level of glossiness (bumpiness) affects judgments of bumpiness (glossiness). The data are from the observers (RK and FC) indicated in Figure 4C of the Ho et al (2008) paper. Each data set contains three replications of the same stimuli with each session being 325 trials long.

## Source

Ho, Y.-H., Landy. M. S., Maloney, L. T. (2008), Conjoint measurement of gloss and surface texture. *Psychological Science*, **19(2)**, 196–204.

## References

Ho, Y.-H., Landy. M. S., Maloney, L. T. (2008), Conjoint measurement of gloss and surface texture. *Psychological Science*, **19(2)**, 196–204.

## Examples

```
data(BumpyGlossy)
data(GlossyBumpy)
```

---

`fitted.mlcm` *Fitted Responses for a Conjoint Measurement Scale*

---

## Description

`fitted.mlcm` returns the fitted responses from an estimated conjoing measurement scale obtained by `mlcm`.

## Usage

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

### Arguments

| | |
|---|---|
| `object` | object of class 'mlcm', typically obtained from the output of `mlcm`. |
| `...` | currently ignored |

### Value

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

### Author(s)

Kenneth Knoblauch

### References

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

### See Also

`mlcm`

### Examples

```
data(BumpyGlossy)
fitted(mlcm(BumpyGlossy))
```

---

## `logLik.mlcm` *Extract Log-Likelihood from mlcm Object*

---

### Description

This is a method function for extracting the log likelihood from objects of class 'mlcm'.

### Usage

```
## S3 method for class 'mlcm'
logLik(object, ...)
```

### Arguments

| | |
|---|---|
| `object, ...` | object of class `mlcm` typically generated by `mlcm`. |

### Details

See the help page for the generic function `logLik` for `details`.

## Value

Returns an object of class `logLik` with an attribute, `df`, giving the degrees of freedom.

## Author(s)

Kenneth Knoblauch

## See Also

`logLik`

## Examples

```
logLik(mlcm(BumpyGlossy))
```

---

`make.wide` *Create data frame for Fitting Conjoint Measurment Models by glm*

---

## Description

`make.wide` and `make.wide.full` generate a $n$ x $q-1$ matrix from an $n$ x 2 column subset of a data frame storing the results of a conjoint measurement experiment, where $n$ is the number of trials and $q$ is the number of levels per dimension in the stimulus set tested. Currently, `make.wide.full` is limited to data sets with only 2 stimulus dimensions. The columns code covariates for all but the first stimulus level, which is constrained to be 0, along each dimension. These columns take the value 0 unless one of the stimuli in the trial corresponded to a level along that dimension, in which case it takes a 1 or a -1, depending on which of the two stimuli represented that level. If both stimuli represent the same level for a dimension, then they cancel out and the column contains a 0. This function is used for each dimension along which the stimuli vary to create a design matrix for each dimension. The final design matrix is constructed inside the `mlcm` method by putting together the design matrices from each dimension.

## Usage

```
make.wide(d)

make.wide.full(d)
```

## Arguments

| | |
|---|---|
| `d` | a $nx2$ column data frame. The columns give the indices of the levels of the dimensions along which the two stimuli presented in a trial vary. |

## Details

This is a helper function, normally used inside `mlcm`, and not typically exploited by the casual user.

## Value

A data frame with $n$ rows and $q - 1$ columns

`D2--Dq` For each dimension along which the stimulus can vary, there are $q - 1$ columns coding the absence or presence of that level of the dimension in the stimulus. If the level is present, then the value is -1 or 1 as a function of which of the two stimuli contained that level, unless both do, in which case it is, also, 0.

## Author(s)

Kenneth Knoblauch

---

`mlcm` *Fit Conjoint Measurement Models by Maximum Likelihood*

---

## Description

Generic function `mlcm` uses different methods to fit the results of a conjoint measurement experiment using `glm` (Generalized Linear Model). The default method permits fitting the data with a choice of 3 different models. The formula method permits fitting the data with a parametric model.

## Usage

```
mlcm(x, ...)

## Default S3 method:
mlcm(x, model = "add", whichdim = NULL, lnk = "probit",
control = glm.control(maxit = 50000, epsilon = 1e-14), ...
)
## S3 method for class 'formula'
mlcm(x, p, data,
model = "add", whichdim = NULL,
lnk = "probit", opt.meth = "BFGS",
control = list(maxit = 50000, reltol = 1e-14), ...)
```

## Arguments

| | |
|---|---|
| `x` | a data frame of an odd number of columns (at least 5) or a formula object. In the case of a data frame, the first should be logical or a 2-level factor named `Resp` indicating the response of the observer. The next columns give the indices in pairs along each dimension for each of the two stimuli being compared. |
| `p` | numeric indicating initial values of parameters for the formula method. |
| `data` | data frame of class 'mlcm.df' for the formula method. |
| `model` | character indicating which of three conjoint measurement models to fit to the data: "add", for additive (default), "ind", for independence or "full", for including a dependence with the levels of each dimension with the others. The "full" is not applicable for the formula method. |

| | |
|---|---|
| whichdim | integer indicating which dimension of the data set to fit when the independence model is chosen |
| lnk | character indicating the link function to use with the binomial family. Current default is the probit link. |
| control | information to control the fit. See glm and glm.control or optim for the formula method. |
| opt.meth | character indicating optimization method (default: "BFGS") for optim with the formula method. |
| ... | additional arguments passed to glm or optim. |

## Details

In a conjoint measurement experiment, observers are presented with pairs of stimuli that vary along 2 or more dimensions. The observer's task is to choose which stimulus of the pair is greater along one of the dimensions. Over a large number of trials, mlcm estimates numbers,

$$\psi_1, ..., \psi_p, \psi'_1, ..., \psi'_q, ...$$

,

by maximum likelihood using glm that best predict the observer's judgments.

The function permits the estimation of 3 different models, independent, additive (the default) and full, by specifying the model argument. The independent model fits the data along only 1 dimension, specified by the whichdim argument. The additive model fits all dimensions with each fixed at 0 at the lowest level on each dimension. Thus, if there are $n$ dimensions each with $p_i$ levels, mlcm estimates $\sum p_i - n$ coefficients.

Specifying the full model will fit a saturated model in which an estimate will be made for each combination of the scale values except the lowest (0 on all scales). This option, now, allows any number of dimensions to be fit.

## Value

a list of class 'mlcm' that will include some of the following components depending on whether the default or formual method is used:

| | |
|---|---|
| pscale | a vector or matrix giving the perceptual scale value estimates |
| stimulus | numeric indicating the scale values along each dimension |
| sigma | numeric indicating judgment $\sigma$, currently always set to 1 |
| par | numeric indicating the fitted parameter values when the formula method is used |
| logLik | log likelihood returned with the formula method |
| hess | Hessian matrix returned with the formual method |
| method | character indicating whether the model was fit by glm or with the formula method |
| se | standard errors returned with the formula method |
| NumDim | numeric indicating number of stimulus dimensions in data set |
| NumLev | numeric indicating the number of levels along both dimensions, currently assumed to be the same |

| | |
|---|---|
| model | character indicating which of the 3 models were fit |
| link | character indicating the link used for the binomial family with glm |
| obj | the 'glm' object |
| data | the 'mlcm' data frame |
| conv | numeric indicating whether convergence was reached in the case of the formula method |
| formula | formula object from argument to formula method |
| func | function constructed from formula object |
| whichdim | numeric indicating which dimension was fit in the case of the "ind" model |

## Author(s)

Ken Knoblauch

## References

Luce, R. D., and Tukey, J. W. (1964). Simultaneous conjoint measurement. *Journal of Mathematical Psychology*, **1**, 1–27.

Krantz, D. H., Luce, R. D., Suppes, P., and Tversky, A. (1971). *Foundations of Measurement, Vol. 1: Additive and Polynomial Representations*. New York: Academic Press.

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

glm

## Examples

```
# Additive model
bg.add <- mlcm(BumpyGlossy)
plot(bg.add, type = "b")

# Independence model for Bumpiness
bg.ind <- mlcm(BumpyGlossy, model = "ind", whichdim = 2)

anova(bg.ind, bg.add, test = "Chisq")

# Full model
bg.full <- mlcm(BumpyGlossy, model = "full")

anova(bg.add, bg.full, test = "Chisq")

opar <- par(mfrow = c(1, 2), pty = "s")
# Compare additive and full model graphically
plot(bg.full, standard.scale = TRUE, type = "b",
lty = 2, ylim = c(0, 1.05),
```

```
xlab = "Gloss Level",
ylab = "Bumpiness Model Estimates")
# additive prediction
bg.pr <- with(bg.add, outer(pscale[, 1], pscale[, 2], "+"))
# predictions are same for arbitrary scaling,
#  so we adjust additive predictions to best fit
#  those from the full model by a scale factor.
cf <- coef(lm(as.vector(bg.full$pscale/bg.full$pscale[5, 5]) ~
as.vector(bg.pr) - 1))
matplot(cf * bg.pr, type = "b", add = TRUE, lty = 1)

#### Now make image of residuals between 2 models
bg.full.sc <- bg.full$pscale/bg.full$pscale[5, 5]
bg.add.adj <- cf * bg.pr
bg.res <- (bg.add.adj - bg.full.sc) + 0.5
image(1:5, 1:5, bg.res,
col = grey.colors(100, min(bg.res), max(bg.res)),
xlab = "Gloss Level", ylab = "Bumpiness Level"
)

#### Example with formula
# additive model
bg.frm <- mlcm(~ p[1] * (x - 1)^p[2] + p[3] * (y - 1)^p[4],
   p = c(0.1, 1.3, 1.6, 0.8), data = BumpyGlossy)
summary(bg.frm)
# independence model
bg.frm1 <- mlcm(~ p[1] * (x - 1)^p[2], p = c(1.6, 0.8),
data = BumpyGlossy, model = "ind", whichdim = 2)
summary(bg.frm1)

### Test additive against independent fits
ddev <- -2 * (logLik(bg.frm1) - logLik(bg.frm))
df <- attr(logLik(bg.frm), "df") - attr(logLik(bg.frm1), "df")
pchisq(as.vector(ddev), df, lower = FALSE)

# Compare additive power law and nonparametric models
xx <- seq(1, 5, len = 100)
par(mfrow = c(1, 1))
plot(bg.add, pch = 21, bg = c("red", "blue"))
lines(xx, predict(bg.frm, newdata = xx)[seq_along(xx)])
lines(xx, predict(bg.frm, newdata = xx)[-seq_along(xx)])
AIC(bg.frm, bg.add)
par(opar)


#### Analysis of 3-way MLCM data set
# additive model
T.mlcm <- mlcm(Texture)
summary(T.mlcm)
plot(T.mlcm, type = "b")
# independent models
lapply(seq(1, 3), function(wh){
m0 <- mlcm(Texture, model = "ind", which = wh)
```

```
anova(m0, T.mlcm, test = "Chisq")
})

# Deviance differences for 2-way interactions vs 2-way additive models

mlcm(Texture[, -c(4, 5)])$obj$deviance -
mlcm(Texture[, -c(4, 5)], model = "full")$obj$deviance
mlcm(Texture[, -c(2, 3)])$obj$deviance -
mlcm(Texture[, -c(2, 3)], model = "full")$obj$deviance
mlcm(Texture[, -c(6, 7)])$obj$deviance -
mlcm(Texture[, -c(6, 7)], model = "full")$obj$deviance

# deviance differences for 3-way interaction tested against 3 2-way interactions
T3way.mlcm <-  mlcm(Texture, model = "full")
## construct model matrix from 3 2-way interactions
T3_2way.mf <- cbind(model.frame(mlcm(Texture[, -c(4, 5)], model = "full")$obj),
model.matrix(mlcm(Texture[, -c(2, 3)], model = "full")$obj),
model.matrix(mlcm(Texture[, -c(6, 7)], model = "full")$obj)
)
T3_2way.mlcm <- glm(Resp ~ . + 0, family = binomial(probit), data = T3_2way.mf)
Chi2 <- T3_2way.mlcm$deviance -  T3way.mlcm$obj$deviance
degfr <- T3_2way.mlcm$df.residual -  T3way.mlcm$obj$df.residual
pchisq(Chi2, degfr, lower.tail = FALSE)
```

---

`plot.mlcm` *Plot an mlcm Object*

---

## Description

Plots the conjoint measurement scale(s) as a function of stimulus level.

## Usage

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

## Arguments

| | |
|---|---|
| `x` | `mlcm` object, typically result of `mlcm` |
| `standard.scale` | logical indicating whether the plotted scales should be normalized so that the maximum scale value is 1 |
| `transpose` | logical, indicating whether to transpose the matrix of the perceptual scale, when the full model is fit. Not defined if there are more than 2 dimensions. |

| | |
|---|---|
| SD.scale | logical indicating whether to plot results in units of d', the signal detection measure of signal strength in which the variance for each stimulus level is unity. Ignored if standard.scale = TRUE. |
| ... | other parameters to be passed through to the plotting function. |

## Details

These functions use matplot, matlines and matpoints so their help page should be examined for information on additional parameters that can be specified.

## Author(s)

Kenneth Knoblauch

## See Also

matplot

## Examples

```
plot(mlcm(BumpyGlossy), type = "b")

bg.full <- mlcm(BumpyGlossy, model = "full")
opar <- par(mfrow = c(1, 2), pty = "s")
plot(bg.full, type = "b",
xlab = "Gloss Level",
ylab = "Bumpiness Model Estimates")
plot(bg.full, transpose = TRUE, type = "b",
xlab = "Bumpiness Level",
ylab = "Glossiness Model Estimates")
par(opar)
```

---

# plot.mlcm.df *Create Conjoint Proportion Plot from mlcm.df Object*

---

## Description

Creates a conjoint proportions plot as in Ho et al. (2008) in which the proportion of responses of one type are indicated as a function of the stimulus pairs used in the experiment.

## Usage

```
## S3 method for class 'mlcm.df'
plot(x, clr = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | object of class 'mlcm.df' typically the result of a conjoint measurement experiment of 2 dimensions. |
| clr | a palette for the color scale in the plot. If none specified, then a grey level palette will be calculated based on the maximum number of repetitions. |
| ... | additional graphical parameters passed to image used for generating the plot. |

## Details

The input should be a data frame of class 'mlcm.df' that contains 5 columns. The first column contains the responses of the observer. The next two correspond to the indices of the stimuli along the first dimensions and the last two the indices of the two stimuli along the second dimension. The plot shows a color-coded (grey-level by default) map of the proportion of responses for each combination of indices in one stimulus with respect to the combinations of the indices in the other. There should be several replications of each pairing for the plot to make some sense.

## Value

Currently, nothing is returned. Used for its side-effect of producing a plot.

## Note

Will not work on experiments using more than 2 dimensions.

## Author(s)

Ken Knoblauch

## References

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

image

## Examples

```
data(BumpyGlossy)
plot(BumpyGlossy,
xlab = expression(paste("Surface ", S[ij],
" Gloss Level (i) and Bump level (j)")),
ylab = expression(paste("Surface ", S[kl],
" Gloss Level (k) and Bump level (l)"))
)
```

predict.mlcm *Predict Method for MLCM Objects*

## Description

Predict values based on conjoint measurment scale fit.

## Usage

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

## Arguments

| | |
|---|---|
| object | object of class 'mlcm' usually created from running mlcm. |
| newdata | numeric vector of new data for which to predict scale values. Only relevant when the formula method is used. |
| type | character indicating whether the predicted value should be on the "link" or "response" scale. Only relevant when the glm method is used. |
| ... | Other parameters passed along to the predict method of glm when the glm method is used. |

## Details

For data sets fit with the glm method, the predicted values are returned either on the "link" or "response" scale. For the formula method, predicted values are returned on the "link" scale. The "newdata" argument is there for this case.

## Value

Numeric vector of predicted values.

## Author(s)

Kenneth Knoblauch

## See Also

mlcm, fitted.mlcm

## Examples

```
bg.add <- mlcm(BumpyGlossy)
bg.frm <- mlcm(~ p[1] * (x - 1)^p[2] + p[3] * (y - 1)^p[4],
p = c(0.1, 1.3, 1.6, 0.8), data = BumpyGlossy)
xx <- seq(1, 5, len = 100)
plot(bg.add)
lines(xx, predict(bg.frm, newdata = xx)[seq_along(xx)])
lines(xx, predict(bg.frm, newdata = xx)[-seq_along(xx)])
```

summary.mlcm *Summary Method for mlcm objects*

## Description

Method functions for mlcm and summary.glm objects.

## Usage

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

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

## Arguments

| | |
|---|---|
| object | an object of class "mlcm", a result of a call to mlcm |
| x | an object of class "summary.mlcm", usually a call to summary.mlcm |
| digits | the number of significant digits to use when printing |
| ... | further arguments passed to or from other methods |

## Details

Displays summary information from a "mlcm" object.

## Value

A list of 5 elements

| | |
|---|---|
| pscale | A named vector or matrix indicting the estimated scale values. |
| sigma | The estimate of the scale parameter, currently always set to 1. |
| logLik | The logarithm of the likelihood. |
| link | The link used for the binomial family. |
| model | A character string giving the name of the model fit. |

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

## Author(s)

Kenneth Knoblauch

## References

Ho, Y. H., Landy. M. S. and Maloney, L. T. (2008). Conjoint measurement of gloss and surface texture. *Psychological Science*, **19**, 196–204.

## See Also

mlcm

## Examples

```
summary(mlcm(BumpyGlossy))
```

---

Texture *Three-way Conjoint Measurement Data for Texture Regularity.*

---

## Description

Data from one subject, S1, for a 3-way MLCM experiment in which the element spacing, element size and inter-element jitter of dot patterns were systematically varied. Pairs of stimuli were presented with the 3 attributes varied independently, and the subject judged which of the pair appeared more regular.

## Usage

```
data("Texture")
```

## Format

A data frame with 4140 observations on the following 7 variables.

`Resp` a numeric vector taking values 0/1 depending on whether the subject chose the first or second stimulus as more regular.

`S1` a numeric vector indicating which of 3 levels of average element spacing for the first stimulus.

`S2` a numeric vector indicating which of 3 levels of average element spacing for the second stimulus.

`Z1` a numeric vector indicating which of 3 levels of element size for the first stimulus.

`Z2` a numeric vector indicating which of 3 levels of element size for the second stimulus.

`J1` a numeric vector indicating which of 5 levels of element jitter for the first stimulus.

`J2` a numeric vector indicating which of 5 levels of element jitter for the second stimulus.

## Details

The authors describe the data sets as follows. Each participant completed 4140 experimental trials, as shown in the 4140 rows: Column 1: response value. 0: participant chose stimulus 1 of a pair as more regular. 1: participant chose stimulus 2 as more regular. Column 2: element spacing level (1–3) for stimulus 1. Column 3: element spacing level (1–3) for stimulus 2. Column 4: element size level (1–3) for stimulus 1. Column 5: element size level (1–3) for stimulus 2. Column 6: element jitter level (1–5) for stimulus 1. Column 7: element jitter level (1–5) for stimulus 2. Note, the trial order was randomized during the experiment.

Additional data files for the other 5 participants in the study can be found in csv format files at doi: 10.1371/journal.pcbi.1008802. With respect to the original file, the current data set was modified to include column names.

A fuller analysis of this data set can be found in the examples at mlcm.

## Source

Sun H.-C., St-Amand D., Baker C. L. Jr, Kingdom F. A. A. (2021), Visual perception of texture regularity: Conjoint measurements and a wavelet response-distribution model. *PLoS Computational Biology* **17(10)**, e1008802. doi: 10.1371/journal.pcbi.1008802.

## References

Knoblauch K., Maloney L. T. (2012) Modeling Psychophysical Data in R, Springer Science \& Business Media, doi: 10.1007/9781461444756.

## Examples

```
data(Texture)
# additive model fit
Texture.mlcm <- mlcm(Texture)
summary(Texture.mlcm)
plot(Texture.mlcm, type = "b")
```

# Index