Package: psyphy (via r-universe)

October 12, 2024

Type Package

Title Functions for Analyzing Psychophysical Data in R

Version 0.3

Date 2023-08-19

Author Kenneth Knoblauch

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

Depends R (>= 3.0), stats, graphics

Suggests MASS, nlme, lattice

LazyData yes

Description An assortment of functions that could be useful in analyzing data from psychophysical experiments. It includes functions for calculating d' from several different experimental designs, links for m-alternative forced-choice (mafc) data to be used with the binomial family in glm (and possibly other contexts) and self-Start functions for estimating gamma values for CRT screen calibrations.

License GPL

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-19 14:30:02 UTC

Contents

`psyphy-package` *Functions for analyzing psychophysical functions*

Description

An assortment of functions that could be useful in analyzing data from pyschophysical experiments. It includes functions for calculating d' from several different experimental designs, links for mafc to be used with the binomial family in glm (and possibly other contexts) and a self-Start function for estimating gamma values for CRT screen calibrations.

Details

For the moment, the package contains several functions for calculating d' for a variety of psychophysical paradigms, some link functions for the binomial family in glm (and perhaps other functions) for fitting psychometric functions from mAFC experiments and a self-Start function for estimating the value of the exponent, gamma, based on the luminance calibration of the three channels of a CRT-like display.

Author(s)

Kenneth Knoblauch <ken.knoblauch@inserm.fr>

`dprime.ABX` *d' for ABX Paradigm*

Description

Calulate d' for ABX paradigm either assuming a differencing strategy or independent observations

Usage

```
dprime.ABX(Hits, FA, zdiff, Pc.unb, method = "diff")
```

Arguments

`Hits`	numeric in [0, 1] corresponding to Hit rate
`FA`	numeric in [0, 1] corresponding to False alarm rate
`zdiff`	numeric. Difference of z-scores for Hit and False Alarm rates
`Pc.unb`	numeric in [0, 1]. Proportion correct for an unbiased observer, `pnorm(zdiff)`
`method`	character. Specifies the model to describe the observer's criterion for dividing up the decision space, must be either "diff" for a differencing strategy (the default) or "IO" for independent observations.

Details

Two different strategies have been described for how the observer partitions the decision space in the ABX paradigm, either based on Independent Observations of each stimulus or on a differencing strategy. The differecing strategy is the default. d' can be calculated either from the `H` and `FA` rates, from the difference of z-scores or from the probability correct of an unbiased observer.

Value

Returns the value of d'

Author(s)

Kenneth Knoblauch

References

MacMillan, N. A. and Creeman, C. D. (1991) *Detection Theory: A User's Guide* Cambridge University Press

Green, D. M. and Swets, J. A. (1966) *Signal Detection Theory and Psychophysics* Robert E. Krieger Publishing Company

See Also

`dprime.mAFC`, `dprime.SD`, `dprime.oddity`

Examples

```
dprime.ABX(H = 0.75, F = 0.3)
dprime.ABX(H = 0.75, F = 0.3, method = "IO")
dprime.ABX(zdiff = qnorm(0.75) - qnorm(0.3))
dprime.ABX(Pc.unb = pnorm( (qnorm(0.75) - qnorm(0.3))/2 ))
```

dprime.mAFC *d' for m-alternative Forced-choice*

Description

Calculate the value of d' for an m-alternative forced choice paradigm

Usage

```
dprime.mAFC(Pc, m)
```

Arguments

`Pc`	The proportion of correct responses based on either the Hit rate or based on an unbiased observer
`m`	The number of alternative choices, an integer > 1.

Details

The probability of a correct response in m-alternative forced-choice, assuming independence, is based on the product of the likelihoods of the signal alternative generating the strongest response and the m - 1 noise alternatives generating responses less than this value (Green and Dai, 1991). For a Yes-No paradigm, the sensitivity is calculated more simply as

$$d' = \texttt{qnorm}(H) - \texttt{qnorm}(F)$$

where H and F are the Hit and False Alarm rates, respectively.

Value

Returns the value of d'

Note

Currently is only valid for d' in the interval [-10, 10] which should be well outside the range of sensory differences that this paradigm is used to investigate.

Author(s)

Kenneth Knoblauch

References

Green, D. M. and Dai, H. (1991) Probability of being correct with 1 of M orthogonal signals. *Perception & Psychophysics*, **49**, 100–101.

Green, D. M. and Swets, J. A. (1966) *Signal Detection Theory and Psychophysics* Robert E. Krieger Publishing Company

See Also

See Also dprime.ABX, dprime.SD, dprime.oddity

Examples

```
dprime.mAFC(0.8, 4)
```

---

dprime.oddity *d' for 3 Stimulus Oddity Paradigm*

Description

Calculate d' for a 3 stimulus (triangular) paradigm. Two of the stimuli are the same and the observer must designate the stimulus that is different.

Usage

```
dprime.oddity(Pc.tri)
```

Arguments

Pc.tri	numeric in (1/3, 1). The proportion of correct responses for an unbiased observer.

Value

Returns the value of d'

Author(s)

Kenneth Knoblauch

References

Frijters, G. S., Kooistra, A. and Verijken, P. F. G. (1980) Tables of d' for the triangular method and the 3-AFC signal detection procedure. *Perception & Psychophysics*, **27**, 176–178.

MacMillan, N. A. and Creeman, C. D. (1991) *Detection Theory: A User's Guide* Cambridge University Press

Green, D. M. and Swets, J. A. (1966) *Signal Detection Theory and Psychophysics* Robert E. Krieger Publishing Company

See Also

dprime.mAFC, dprime.SD, dprime.ABX

Examples

```
dprime.oddity(0.8)
```

`dprime.SD` *d' for Same-different Paradigm*

Description

Calulate d' for same-different paradigm either assuming a differencing strategy or independent observations

Usage

```
dprime.SD(H, FA, zdiff, Pcmax, method = "diff")
```

Arguments

`H`	numeric in [0, 1] corresponding to Hit rate
`FA`	numeric in [0, 1] corresponding to False alarm rate
`zdiff`	numeric. Difference of z-scores for Hit and False Alarm rates (only valid for method "IO")
`Pcmax`	numeric in [0, 1]. Proportion correct for an unbiased observer, `pnorm(zdiff/2)` (only valid for method "IO").
`method`	character. Specifies the model to describe the observer's criterion for dividing up the decision space, must be either "diff" for a differencing strategy (the default) or "IO" for independent observations.

Details

Two different strategies have been described for how the observer partitions the decision space in the same-different paradigm. With Independent Observations, d' can be calculated either from the `H` and `FA` rates, from the difference of z-scores or from the probability correct of an unbiased observer. Only one of these three choices should be specified in the arguments. For the differencing strategy, only the first of these choices is valid.

Value

Returns the value of d'

Author(s)

Kenneth Knoblauch

References

MacMillan, N. A. and Creeman, C. D. (1991) *Detection Theory: A User's Guide* Cambridge University Press

Green, D. M. and Swets, J. A. (1966) *Signal Detection Theory and Psychophysics* Robert E. Krieger Publishing Company

See Also

dprime.mAFC, dprime.ABX, dprime.oddity

Examples

```
dprime.SD(H = 0.642, F = 0.3)
dprime.SD(H = 0.75, F = 0.3, method = "IO")
dprime.SD(zdiff = qnorm(0.75) - qnorm(0.3), method = "IO")
dprime.SD(Pcmax = pnorm( (qnorm(0.75) - qnorm(0.3))/2 ), method = "IO")
```

---

ecc2 *4-afc Detection and Identification of Letters*

Description

Letter detection and identification at 2 degrees eccentricity in the visual field. On each trial, one of four letters (b, d, p, q) were presented in one of four positions (superior, inferior, left, right) in the visual field. In a given session, the letter height was fixed. Six contrast levels were tested in each session. The data indicate the proportion of correctly identified positions, referred to here as detection, and the proportion of correctly identified letters, conditional on correct identification.

Usage

```
data(ecc2)
```

Format

A data frame with 48 observations on the following 5 variables.

`Contr` numeric. The contrast of the stimulus, defined as Weberian contrast.

`task` a factor with levels `DET` `ID` indicating the two tasks, detection and identification.

`Size` a numeric vector indicating the letter height

`Correct` an integer vector indicating the number of correct responses (`DET` or `ID`).

`Incorrect` an integer vector, indicating the number of incorrect responses.

References

Yssaad-Fesselier, R. and Knoblauch, K. (2006) Modeling psychometric functions in R. *Behav Res Methods.*, **38(1)**, 28–41.

Examples

```
data(ecc2)
library(lattice)
xyplot(Correct/(Correct + Incorrect) ~ Contr | Size * task, ecc2,
  type = "b", scale = list(x = list(log = TRUE),
  y = list(limits = c(0, 1.05))),
  xlab = "Contrast", ylab = "Proportion Correct Response",
  panel = function(x, y, ...) {
   panel.xyplot(x, y, ...)
   panel.abline(h = 0.25, lty = 2)
   })
```

`glm.lambda` *mafc Probit Fit to Psychometric Function Profiled on Upper Asymptote*

Description

A wrapper for `glm` in which the deviance for the model with binomial family and link `probit.lambda` is profiled as a function of `lambda`, the upper asymptote of the psychometric function.

Usage

```
glm.lambda(formula, data, NumAlt = 2, lambda = seq(0, 0.1, len = 40),
 plot.it = FALSE, ...)
```

Arguments

`formula`	a symbolic description of the model to be fit
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `glm` is called.
`NumAlt`	the number of alternatives, `m` in the mafc experiment from which the data arise
`lambda`	a sequence of values to profile for the upper asymptote of the psychometric function
`plot.it`	logical indicating whether to plot the profile of the deviances as a function of `lambda`
`...`	further arguments passed to `glm`

Details

The psychometric function fit to the data is described by

$$P(x) = 1/m + (1 - 1/m - \lambda)\Phi(x)$$

where m is the number of alternatives and the lower asymptote, $1 - \lambda$ is the upper asymptote and Φ is the cumulative normal function.

Value

returns an object of class 'lambda' which inherits from classes 'glm' and 'lm'. It only differs from an object of class 'glm' in including two additional components, `lambda`, giving the estimated minimum of the profile by fitting a quadratic to the profile and a data frame containing the profiled deviance values for each value of `lambda` tested. The degrees of freedom are reduced by 1 to take into account the estimation of `lambda`.

Note

If the minimum occurs outside the interval examined, an error might occur. In any case, re-running the function with a new range of `lambda` that includes the minimum should work. if the plotted profile indicates that the fitted quadratic does not describe well the profile at the minimum, refitting with a more restricted range of `lambda` is recommended.

Author(s)

Ken Knoblauch

References

Wichmann, F. A. and Hill, N. J. (2001) The psychometric function: I.Fitting, sampling, and goodness of fit. Percept Psychophys., 63(8), 1293–1313.

Yssaad-Fesselier, R. and Knoblauch, K. (2006) Modeling psychometric functions in R. *Behav Res Methods.*, **38(1)**, 28–41. (for examples with `gnlr`).

See Also

`mafc`, `glm`, `probit.lambda`, `family`

Examples

```
b <- 3.5
g <- 1/3
d <- 0.025
a <- 0.04
p <- c(a, b, g, d)
num.tr <- 160
cnt <- 10^seq(-2, -1, length = 6) # contrast levels

#simulated Weibull-Quick observer responses
set.seed(12161952)
truep <- g + (1 - g - d) * pweibull(cnt, b, a)
ny <- rbinom(length(cnt), num.tr, truep)
nn <- num.tr - ny
phat <- ny/(ny + nn)
resp.mat <- matrix(c(ny, nn), ncol = 2)

## First with upper asymptote at 1
dd.glm <- glm(resp.mat ~ cnt, family = binomial(mafc.probit(3)))
summary(dd.glm)
```

```
dd.lam <- glm.lambda(resp.mat ~ cnt, NumAlt = 3, lambda = seq(0, 0.03,
 len = 100), plot.it = TRUE)
summary(dd.lam)
## can refine interval, but doesn't change result much
dd.lam2 <- glm.lambda(resp.mat ~ cnt, NumAlt = 3,
 lambda = seq(dd.lam$lambda/sqrt(2), dd.lam$lambda*sqrt(2),
 len = 100), plot.it = TRUE)
summary(dd.lam2)
## Compare fits w/ and w/out lambda
anova(dd.glm, dd.lam2, test = "Chisq")

plot(cnt, phat, log = "x", cex = 1.5, ylim = c(0, 1))
pcnt <- seq(0.01, 0.1, len = 100)
lines(pcnt, predict(dd.glm, data.frame(cnt = pcnt),
                    type = "response"), lwd = 2)
lines(pcnt, predict(dd.lam, data.frame(cnt = pcnt),
                    type = "response"), lwd = 2, lty = 2)
```

`glm.WH` *mafc Probit Fit to Psychometric Function with Upper Asymptote Less than One*

Description

A probit fit of a psychometric function with upper asymptote less than 1 is obtained by cycling between a fit with `glm` using the `probit.lambda` link and `optimize` to estimate `lambda`, 1 - the upper asymptotic value, until the log Likelihood changes by less than a pre-set tolerance.

Usage

```
glm.WH(formula, data, NumAlt = 2, lambda.init = 0.01,
 interval = c(0, 0.05), trace = FALSE, tol = 1e-06, ...)
```

Arguments

`formula`	a symbolic description of the model to be fit.
`data`	an optional data frame, list or enviroment (or object coercible by `as.data.frame` containing the variables in the model. If not found in data, the variables are taken from the `environment(formula)`, typically the environment from `glm.WH` was called.
`NumAlt`	integer indicating the number of alternatives (> 1) in the mafc-task. (Default: 2).
`lambda.init`	numeric, initial estimate of 1 - upper asymptote.
`interval`	numeric vector giving interval endpoints within which to search for `lambda`.
`trace`	logical, indicating whether or not to print out a trace of the iterative process.
`tol`	numeric, tolerance for ending iterations.
`...`	futher arguments passed to `glm`.

Details

The psychometric function fit to the data is described by

$$P(x) = 1/m + (1 - 1/m - \lambda)\Phi(x)$$

where m is the number of alternatives and the lower asymptote, $1 - \lambda$ is the upper asymptote and Φ is the cumulative normal function.

Value

returns an object of class 'lambda' which inherits from classes 'glm' and 'lm'. It only differs from an object of class 'glm' in including an additional components, `lambda`, giving the estimated minimum of `lambda`. The degrees of freedom are reduced by 1 to take into account the estimation of `lambda`.

Author(s)

Ken Knoblauch

References

Wichmann, F. A. and Hill, N. J. (2001) The psychometric function: I.Fitting, sampling, and goodness of fit. Percept Psychophys., 63(8), 1293–1313.

Yssaad-Fesselier, R. and Knoblauch, K. (2006) Modeling psychometric functions in R. *Behav Res Methods.*, **38(1)**, 28–41. (for examples with `gnlr`).

See Also

`mafc`, `glm`,`glm.lambda`, `probit.lambda`, `family`

Examples

```
b <- 3.5
g <- 1/4
d <- 0.04
a <- 0.04
p <- c(a, b, g, d)
num.tr <- 160
cnt <- 10^seq(-2, -1, length = 6) # contrast levels

#simulated Weibull-Quick observer responses
truep <- g + (1 - g - d) * pweibull(cnt, b, a)
ny <- rbinom(length(cnt), num.tr, truep)
nn <- num.tr - ny
phat <- ny/(ny + nn)
resp.mat <- matrix(c(ny, nn), ncol = 2)

tst.glm <- glm(resp.mat ~ cnt, binomial(mafc.probit(1/g)))
pcnt <- seq(0.005, 1, len = 1000)
plot(cnt, phat, log = "x", ylim = c(0, 1), xlim = c(0.005, 1),
 cex = 1.75)
```

```
lines(pcnt, predict(tst.glm, data.frame(cnt = pcnt), type = "response"), lwd = 2)
tst.lam <- glm.WH(resp.mat ~ cnt, NumAlt = 1/g, trace = TRUE)
lines(pcnt, predict(tst.lam, data.frame(cnt = pcnt),
 type = "response"), lty = 2, lwd = 2)
```

---

logit.2asym *Links for Binomial Family with Variable Upper/Lower Asymptotes*

Description

These functions provide links for the binamial family so that psychometric functions can be fit with *both* the upper and lower asymptotes different from 1 and 0, respectively.

Usage

```
logit.2asym(g, lam)
probit.2asym(g, lam)
cauchit.2asym(g, lam)
cloglog.2asym(g, lam)
weib.2asym( ... )
```

Arguments

g	numeric in the range (0, 1), normally <= 0.5, however, which specifies the lower asymptote of the psychometric function.
lam	numeric in the range (0, 1), specifying 1 - the upper asymptote of the psychometric function.
...	used just to pass along the formals of `cloglog.2asym` as arguments to `weib.2asym`.

Details

These links are used to specify psychometric functions with the form

$$P(x) = \gamma + (1 - \gamma - \lambda)p(x)$$

where γ is the lower asymptote and $lambda$ is $1-$ the upper asymptote, and $p(x)$ is the base psychometric function, varying between 0 and 1.

Value

Each link returns a list containing functions required for relating the response to the linear predictor in generalized linear models and the name of the link.

linkfun	The link function
linkinv	The inverse link function
mu.eta	The derivative of the inverse link
valideta	The domain over which the linear predictor is valid
link	A name to be used for the link

Author(s)

Kenneth Knoblauch

References

Klein S. A. (2001) Measuring, estimating, and understanding the psychometric function: a commentary. *Percept Psychophys.*, **63(8)**, 1421–1455.

Wichmann, F. A. and Hill, N. J. (2001) The psychometric function: I.Fitting, sampling, and goodness of fit. *Percept Psychophys.*, **63(8)**, 1293–1313.

See Also

`glm`, `glm` `make.link`, `psyfun.2asym`

Examples

```
#A toy example,
b <- 3
g <- 0.05 # simulated false alarm rate
d <- 0.03
a <- 0.04
p <- c(a, b, g, d)
num.tr <- 160
cnt <- 10^seq(-2, -1, length = 6) # contrast levels

#simulated Weibull-Quick observer responses
truep <- g + (1 - g - d) * pweibull(cnt, b, a)
ny <- rbinom(length(cnt), num.tr, truep)
nn <- num.tr - ny
phat <- ny/(ny + nn)
resp.mat <- matrix(c(ny, nn), ncol = 2)

ddprob.glm <- psyfun.2asym(resp.mat ~ cnt, link = probit.2asym)
ddlog.glm <- psyfun.2asym(resp.mat ~ cnt, link = logit.2asym)
# Can fit a Weibull function, but use log contrast as variable
ddweib.glm <- psyfun.2asym(resp.mat ~ log(cnt), link = weib.2asym)
ddcau.glm <- psyfun.2asym(resp.mat ~ cnt, link = cauchit.2asym)

plot(cnt, phat, log = "x", cex = 1.5, ylim = c(0, 1))
pcnt <- seq(0.01, 0.1, len = 100)
lines(pcnt, predict(ddprob.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 5)
lines(pcnt, predict(ddlog.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 2, lty = 2, col = "blue")
lines(pcnt, predict(ddweib.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 3, col = "grey")
lines(pcnt, predict(ddcau.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 3, col = "grey", lty = 2)
```

mafc *Links for Binomial Family for m-alternative Forced-choice*

Description

These provide links for the binomial family for fitting m-alternative forced-choice psychophysical functions.

Usage

```
mafc.logit( .m = 2 )
mafc.probit( .m = 2 )
mafc.cloglog( .m = 2 )
mafc.weib( ... )
mafc.cauchit( .m = 2 )
```

Arguments

`.m`	is the integer number (>1) of choices (Default to 2AFC). For m = 1 (Yes/No paradigm), use one of the built-in links for the binomial family.
`...`	just to pass along the formals of `mafc.cloglog`.

Details

These functions provide links for fitting psychometric functions arising from an m-alternative forced-choice experiment. The estimated coefficients of the linear predictor influence both the location and the slope of the psychometric function(s), but provide no means of estimating the upper aymptote which is constrained to approach 1. If the upper asympotote must be estimated, it would be better to maximize directly the likelihood, either with a function like `optim` or `gnlr` from package **gnlm** (available at `https://www.commanster.eu/rcode.html`). Alternatively, the function `probit.lambda` can be used with a known upper asymptote, or `glm.lambda` or `glm.WH` to estimate one, with a probit link. `mafc.weib` is just an alias for `mafc.cloglog`.

Value

Each link returns a list containing functions required for relating the response to the linear predictor in generalized linear models and the name of the link.

`linkfun`	The link function
`linkinv`	The inverse link function
`mu.eta`	The derivative of the inverse link
`valideta`	The domain over which the linear predictor is valid
`link`	A name to be used for the link

Author(s)

Kenneth Knoblauch

References

Williams J, Ramaswamy D and Oulhaj A (2006) 10 Hz flicker improves recognition memory in older people *BMC Neurosci.* 2006 5;7:21 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1434755/ (for an example developed prior to this one, but for m = 2).

Klein S. A. (2001) Measuring, estimating, and understanding the psychometric function: a commentary. *Percept Psychophys.*, **63(8)**, 1421–1455.

Wichmann, F. A. and Hill, N. J. (2001) The psychometric function: I.Fitting, sampling, and goodness of fit. *Percept Psychophys.*, **63(8)**, 1293–1313.

Yssaad-Fesselier, R. and Knoblauch, K. (2006) Modeling psychometric functions in R. *Behav Res Methods.*, **38(1)**, 28–41. (for examples with `gnlr`).

See Also

`family`, `make.link`, `glm`, `optim`, `probit.lambda`, `glm.lambda`, `glm.WH`

Examples

```
#A toy example,
b <- 3.5
g <- 1/3
d <- 0.0
a <- 0.04
p <- c(a, b, g, d)
num.tr <- 160
cnt <- 10^seq(-2, -1, length = 6) # contrast levels

#simulated observer responses
truep <- g + (1 - g - d) * pweibull(cnt, b, a)
ny <- rbinom(length(cnt), num.tr, truep)
nn <- num.tr - ny
phat <- ny/(ny + nn)
resp.mat <- matrix(c(ny, nn), ncol = 2)

ddprob.glm <- glm(resp.mat ~ cnt, family = binomial(mafc.probit(3)))
ddlog.glm <- glm(resp.mat ~ cnt, family = binomial(mafc.logit(3)))
# Can fit a Weibull function, but use log contrast as variable
ddweib.glm <- glm(resp.mat ~ log(cnt), family = binomial(mafc.cloglog(3)))
ddcau.glm <- glm(resp.mat ~ log(cnt), family = binomial(mafc.cauchit(3)))

plot(cnt, phat, log = "x", cex = 1.5, ylim = c(0, 1))
pcnt <- seq(0.01, 0.1, len = 100)
lines(pcnt, predict(ddprob.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 2)
lines(pcnt, predict(ddlog.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 2, lty = 2)
lines(pcnt, predict(ddweib.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 3, col = "grey")
lines(pcnt, predict(ddcau.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 3, col = "grey", lty = 2)
```

```
# Weibull parameters \alpha and \beta
cc <- coef(ddweib.glm)
alph <- exp(-cc[1]/cc[2])
bet <- cc[2]


#More interesting example with data from Yssaad-Fesselier and Knoblauch
data(ecc2)
ecc2.glm <- glm(cbind(Correct, Incorrect) ~ Contr * Size * task,
   family = binomial(mafc.probit(4)), data = ecc2)
summary(ecc2.glm)
ecc2$fit <- fitted(ecc2.glm)
library(lattice)
xyplot(Correct/(Correct + Incorrect) ~ Contr | Size * task, data = ecc2,
 subscripts = TRUE, ID = with(ecc2, Size + as.numeric(task)),
 scale = list(x = list(log = TRUE),
     y = list(limits = c(0, 1.05))),
 xlab = "Contrast", ylab = "Proportion Correct Response",
 aspect = "xy",
 panel = function(x, y, subscripts, ID, ...) {
  which = unique(ID[subscripts])
  llines(x, ecc2$fit[which ==ID], col = "black", ...)
  panel.xyplot(x, y, pch = 16, ...)
  panel.abline(h = 0.25, lty = 2, ...)
  }
)
```

`probit.lambda` *mafc Probit Link for Binomial Family with Upper Asymptote < 1*

Description

This function provides a link for the binomial family for fitting m-alternative forced-choice, with a probit link and with the upper asymptote permitted to be less than 1.

Usage

```
probit.lambda(m = 2, lambda = 0)
```

Arguments

`m`	is the integer number (>1) of choices (Default to 2AFC).
`lambda`	number in [0, 1] indicating 1 minus the upper asymptotic value of the psychometric function.

Details

This function provides a link for fitting psychometric functions arising from an m-alternative forced-choice experiment using a probit link and allowing that the upper aymptote is less than 1. The psychometric function fit to the data is described by

$$P(x) = 1/m + (1 - 1/m - \lambda)\Phi(x)$$

where m is the number of alternatives and the lower asymptote, $1 - \lambda$ is the upper asymptote and Φ is the cumulative normal function.

Value

The link returns a list containing functions required for relating the response to the linear predictor in generalized linear models and the name of the link.

`linkfun`	The link function
`linkinv`	DTHe inverse link function
`mu.eta`	The derivative of the inverse link function
`valideta`	The domain over which the linear predictor is valid
`link`	A name to be used for the link

Note

Due to the difficulty of the task, subject error or incorrectly recorded data, psychophysical data may reveal less than perfect performance when stimulus differences are readily visible. When this occurs, letting the upper asymptote be less than 1 often results in a better fit to the data and a less-biased estimate of the steepness of the curve (see example below).

Author(s)

Ken Knoblauch

References

Wichmann, F. A. and Hill, N. J. (2001) The psychometric function: I.Fitting, sampling, and goodness of fit. *Percept Psychophys.*, 63(8), 1293–1313.

See Also

`mafc`, `glm`, `glm.lambda`, `family`, `make.link`

Examples

```
b <- 3.5
g <- 1/3
d <- 0.025
a <- 0.04
p <- c(a, b, g, d)
num.tr <- 160
```

```
cnt <- 10^seq(-2, -1, length = 6) # contrast levels

#simulated Weibull-Quick observer responses
truep <- g + (1 - g - d) * pweibull(cnt, b, a)
ny <- rbinom(length(cnt), num.tr, truep)
nn <- num.tr - ny
phat <- ny/(ny + nn)
resp.mat <- matrix(c(ny, nn), ncol = 2)

ddprob.glm <- glm(resp.mat ~ cnt, family = binomial(mafc.probit(3)))
ddprob.lam <- glm(resp.mat ~ cnt, family = binomial(probit.lambda(3, 0.025)))
AIC(ddprob.glm, ddprob.lam)

plot(cnt, phat, log = "x", cex = 1.5, ylim = c(0, 1))
pcnt <- seq(0.01, 0.1, len = 100)
lines(pcnt, predict(ddprob.glm, data.frame(cnt = pcnt),
                        type = "response"), lwd = 2)
lines(pcnt, predict(ddprob.lam, data.frame(cnt = pcnt),
                        type = "response"), lwd = 2, lty = 2)
```

---

psyfun.2asym *Fit Psychometric Functions and Upper and Lower Asymptotes*

Description

Fits psychometric functions allowing for variation of both upper and lower asymptotes. Uses a procedure that alternates between fitting linear predictor with `glm` and estimating the asymptotes with `optim` until a minimum in -log likelihood is obtained within a tolerance.

Usage

```
psyfun.2asym(formula, data, link = logit.2asym, init.g = 0.01,
 init.lam = 0.01, trace = FALSE, tol = 1e-06,
 mxNumAlt = 50, ...)
```

Arguments

`formula`	a two sided formula specifying the response and the linear predictor
`data`	a data frame within which the formula terms are interpreted
`link`	a link function for the binomial family that allows specifying both upper and lower asymptotes
`init.g`	numeric specifying the initial estimate for the lower asymptote
`init.lam`	numeric specifying initial estimate for 1 - upper asymptote
`trace`	logical indicating whether to show the trace of the minimization of -log likelihood
`tol`	numeric indicating change in -log likelihood as a criterion for stopping iteration.
`mxNumAlt`	integer indicating maximum number of alternations between `glm` and `optim` steps to perform if minimum not reached.
`...`	additional arguments passed to `glm`

Details

The function is a wrapper for `glm` for fitting psychometric functions with the equation

$$P(x) = \gamma + (1 - \gamma - \lambda)p(x)$$

where γ is the lower asymptote and $lambda$ is $1-$ the upper asymptote, and $p(x)$ is the base psychometric function, varying between 0 and 1.

Value

list of class 'lambda' inheriting from classes 'glm' and 'lm' and containing additional components

`lambda`	numeric indicating 1 - upper asymptote
`gam`	numeric indicating lower asymptote
`SElambda`	numeric indicating standard error estimate for lambda based on the Hessian of the last interation of `optim`. The optimization is done on the value transformed by the function `plogis` and the value is stored in on this scale
`SEgam`	numeric indicating standard error estimate for gam estimated in the same fashion as `SElambda`

If a diagonal element of the Hessian is sufficiently close to 0, `NA` is returned.

Note

The `cloglog.2asym` and its alias, `weib.2asym`, don't converge on occasion. This can be observed by using the `trace` argument. One strategy is to modify the initial estimates.

Author(s)

Kenneth Knoblauch

References

Klein S. A. (2001) Measuring, estimating, and understanding the psychometric function: a commentary. *Percept Psychophys.*, **63(8)**, 1421–1455.

Wichmann, F. A. and Hill, N. J. (2001) The psychometric function: I.Fitting, sampling, and goodness of fit. *Percept Psychophys.*, **63(8)**, 1293–1313.

See Also

`glm`, `optim`, `glm.lambda`, `mafc`

Examples

```
#A toy example,
set.seed(12161952)
b <- 3
g <- 0.05 # simulated false alarm rate
d <- 0.03
a <- 0.04
```

```
p <- c(a, b, g, d)
num.tr <- 160
cnt <- 10^seq(-2, -1, length = 6) # contrast levels

#simulated Weibull-Quick observer responses
truep <- g + (1 - g - d) * pweibull(cnt, b, a)
ny <- rbinom(length(cnt), num.tr, truep)
nn <- num.tr - ny
phat <- ny/(ny + nn)
resp.mat <- matrix(c(ny, nn), ncol = 2)

ddprob.glm <- psyfun.2asym(resp.mat ~ cnt, link = probit.2asym)
ddlog.glm <- psyfun.2asym(resp.mat ~ cnt, link = logit.2asym)
# Can fit a Weibull function, but use log contrast as variable
ddweib.glm <- psyfun.2asym(resp.mat ~ log(cnt), link = weib.2asym)
ddcau.glm <- psyfun.2asym(resp.mat ~ cnt, link = cauchit.2asym)

plot(cnt, phat, log = "x", cex = 1.5, ylim = c(0, 1))
pcnt <- seq(0.01, 0.1, len = 100)
lines(pcnt, predict(ddprob.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 5)
lines(pcnt, predict(ddlog.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 2, lty = 2, col = "blue")
lines(pcnt, predict(ddweib.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 3, col = "grey")
lines(pcnt, predict(ddcau.glm, data.frame(cnt = pcnt),
   type = "response"), lwd = 3, col = "grey", lty = 2)
summary(ddprob.glm)
```

RGB *Luminance Calibration Data from Video Projector*

Description

The data were obtained from the measurements of the luminance of the R, G and B channels individually, as well as the three together, W, for each of 21 grey levels, GL from a screen on which a video projector was displaying an image of a uniform field. Grey level has been normalized to the interval [0, 1], though originally it is specified as integers in [0, 255]. The measurements were obtained with a Photo Research 650 spectro-radiometer.

Usage

```
data(RGB)
```

Format

A data frame with 84 observations on the following 3 variables.

Lum numeric vector of the measured luminance in candelas/meter^2

GL The grey level normalized to the interval [0, 1]

Gun factor with levels R G B W

Examples

```
data(RGB)
```

SS.RGBcalib	*Self-Start Functions for Fitting Luminance vs Grey Level Relation on CRT displays*

Description

This `selfStart` model evaluates the parameters for describing the luminance vs grey level relation of the R, G and B guns of a CRT-like display, fitting a single exponent, gamma, for each of the 3 guns. It has an initial attribute that will evaluate initial estimates of the parameters, `Blev`, `Br`, `Bg`, `Bb` and `gamm`. In the case of fitting data from a single gun or for a combination of guns, as in the sum of the three for calibrating the *white*, the parameter `k` is used for the coefficient. Both functions include gradient and hessian attributes.

Usage

```
SS.calib(Blev, k, gamm, GL)
SS.RGBcalib(Blev, Br, Bg, Bb, gamm, Rgun, Ggun, Bgun)
```

Arguments

`Blev`	numeric. The black level is the luminance at the 0 grey level
`k`	numeric, coefficient of one gun for fitting single gun
`Br`	numeric, coefficient of the R gun
`Bg`	numeric, coefficient of the G gun
`Bb`	numeric, coefficient of the B gun
`gamm`	numeric, the exponent, gamma, applied to the grey level
`GL`	numeric, is the grey level for the gun tested, covariate in model matrix in one gun case
`Rgun`	numeric, is a covariate in the model matrix that indicates the grey level for the R gun. See the example below.
`Ggun`	numeric, is a covariate in the model matrix that indicates the grey level for the G gun
`Bgun`	numeric, is a covariate in the model matrix that indicates the grey level for the B gun

Details

The model

$$Lum(GL) = Blev + \beta_i * GL^\gamma$$

where i is in {R, G, B}, usually provides a reasonable description of the change in luminance of a display gun with grey level, GL. This selfStart function estimates γ and the other parameters using the nls function. It is assumed that grey level is normalized to the interval [0, 1]. This results in lower correlation between the linear coefficients of the guns, β_i , than if the actual bit-level is used, e.g., [0, 255], for an 8-bit graphics card (see the example). Also, with this normalization of the data, the coefficients, β_i, provide estimates of the maximum luminance for each gun. The need for the arguments Rgun, Ggun and Bgun is really a kludge in order to add gradient and hessian information to the model.

Value

returns a numeric vector giving the estimated luminance given the parameters passed as arguments and a gradient matrix and a hessian array as attributes.

Author(s)

Kenneth Knoblauch

References

~put references to the literature/web site here ~

See Also

nls

Examples

```
data(RGB)

#Fitting a single gun
W.nls <- nls(Lum ~ SS.calib(Blev, k, gamm, GL), data = RGB,
    subset = (Gun == "W"))
summary(W.nls)

#curvature (parameter effect) is greater when GL is 0:255
Wc.nls <- nls(Lum ~ SS.calib(Blev, k, gamm, GL*255), data = RGB,
    subset = (Gun == "W"))
MASS::rms.curv(W.nls)
MASS::rms.curv(Wc.nls)
pairs(profile(Wc.nls), absVal = FALSE)
pairs(profile(W.nls), absVal = FALSE)


#Fitting 3 guns with independent gamma's
RGB0.nls <- nlme::nlsList(Lum ~ SS.calib(Blev, k, gamm, GL) | Gun,
  data = subset(RGB, Gun != "W"))
```

```
summary(RGB0.nls)
plot(nlme::intervals(RGB0.nls))

# Add covariates to data.frame for R, G and B grey levels
gg <- model.matrix(~-1 + Gun/GL, RGB)[ , c(5:7)]
RGB$Rgun <- gg[, 1]
RGB$Ggun <- gg[, 2]
RGB$Bgun <- gg[, 3]
RGB.nls <- nls(Lum ~ SS.RGBcalib(Blev, Br, Bg, Bb, gamm, Rgun, Ggun, Bgun),
                  data = RGB, subset = (Gun != "W") )
summary(RGB.nls)
confint(RGB.nls)
```

---

summary.lambda *Summary Method for Objects of Class 'lambda'*

Description

Identical to summary.glm but with one line of additional output: the estimate of lambda from glm.lambda, obtained by profiling the deviance and estimating its minimum.

Usage

```
## S3 method for class 'lambda'
summary(object, ...)
## S3 method for class 'summary.lambda'
print(x, digits = max(3, getOption("digits") - 3),
  symbolic.cor = x$symbolic.cor,
  signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

object	Fitted model object of class "lambda" inheriting from glm and lm. Typically the output of glm.lambda.
x	an object of class "summary.lambda", usually a result of a call to summary.lambda.
digits	the number of significant digits to use when printing.
symbolic.cor	logical. If TRUE, print the correlations in a symbolic form (see symnum) rather than as numbers.
signif.stars	logical. If TRUE, "significance stars" are printed for each coefficient.
...	further arguments passed to or from other methods.

Details

Provides a summary of the class lambda object generated by glm.lambda.

Value

Returns the same structure as `summary.glm` with an added component, `lambda`. $1 - \lambda$ is the estimated upper asymptote of the psychometric function.

Author(s)

Ken Knoblauch

See Also

`probit.lambda`, `glm.lambda`

Index