Package ‘modelfree’

May 20, 2023

Title Model-Free Estimation of a Psychometric Function

Version 1.2

Description Local linear estimation of psychometric functions. Provides functions for nonparametric estimation of a psychometric function and for estimation of a derived threshold and slope, and their standard deviations and confidence intervals.

URL www.R-project.org,

https://personalpages.manchester.ac.uk/staff/d.h.foster/software-modelfree/latest/home.html

License LGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports SparseM, PolynomF, methods

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Ivan Marin-Franch [aut] (<https://orcid.org/0000-0001-7079-6874>),
Kamila Zychaluk [aut, cre, cph],
David H. Foster [aut] (<https://orcid.org/0000-0003-2428-715X>)

Maintainer Kamila Zychaluk <Kamila.Zychaluk@liverpool.ac.uk>

Repository CRAN

Date/Publication 2023-05-20 07:40:02 UTC

R topics documented:

`Baker_etal` *Auditory detection of a gap in noise*

Description

A 300-ms noise burst containing a gap of 2–8 ms duration or no gap was presented to one ear of a subject. The symbols in the figure below show the proportion of responses "gap" as a function of gap duration. There were 12 trials with each gap duration and 84 trials with no gap. See `https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example07R.html`

Usage

```
data("Baker_etal")
```

Format

A data frame with 8 rows and 3 columns.

`x` stimulus level

`r` number of successes

`m` number of trials

References

Baker, R. J., Jayewardene, D., Sayle, C., & Saeed, S. “Failure to find asymmetry in auditory gap detection”, Laterality: Asymmetries of Body, Brain and Cognition, 13, 1-21, 2008.

Examples

```
data("Baker_etal")
x = Schofield_etal$x
r = Schofield_etal$r
m = Schofield_etal$m
plot( x, r / m, xlim = c( 0.16, 7.83 ), ylim = c( -0.01, 1.01 ), type = "p", pch="*" )
```

`bandwidth_bootstrap` *Bootstrap bandwidth for local polynomial estimator of a psychometric function*

Description

This function finds a bootstrap estimate of the optimal bandwidth h for a local polynomial estimate of the psychometric function with specified guessing and lapsing rates.

Usage

```
bandwidth_bootstrap( r, m, x, H, N, h0 = NULL, link = "logit",
                     guessing = 0, lapsing = 0, K = 2, p = 1,
                     ker = "dnorm", maxiter = 50, tol = 1e-6, method = "all")
```

Arguments

`r` number of successes at points x

`m` number of trials at points x

`x` stimulus levels

`H` search interval

`N` number of bootstrap replications

`h0` (optional) pilot bandwidth; if not specified, then the scaled plug-in bandwidth is used

`link` (optional) name of the link function to be used; default is "logit"

`guessing` (optional) guessing rate; default is 0

`lapsing` (optional) lapsing rate; default is 0

`K` (optional) power parameter for Weibull and reverse Weibull link; default is 2

`p` (optional) degree of the polynomial; default is 1

`ker` (optional) kernel function for weights; default is "dnorm"

`maxiter` (optional) maximum number of iterations in Fisher scoring; default is 50

`tol` (optional) tolerance level at which to stop Fisher scoring; default is 1e-6

`method` (optional) loss function to be used in bootstrap: choose from: "ISEeta", "ISE", "deviance"; by default all possible values are calculated

Value

h bootstrap bandwidth for the chosen "method"; if no "method" is specified, then it has three components: $pscale, $eta-scale and $deviance

Examples

```
data("Baker_etal")
x = Baker_etal$x
r = Baker_etal$r
m = Baker_etal$m
plot( x, r / m, xlim = c( 0.16, 7.83 ), ylim = c( -0.01, 1.01 ), type = "p", pch="*" )
val <- binomfit_lims( r, m, x, link = "probit" )
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Plot the fitted curve
pfit<-predict( val$fit, data.frame( x = xfit ), type = "response" )
lines(xfit, pfit )
## Not run:
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Find a cross-validation bandwidth
bwd_min <- min( diff( x ) )
bwd_max <- max( x ) - min( x )
# This might take a few minutes
niter <- 500 # Note number of bootstrap iterations should be at least 500
bwd <- bandwidth_bootstrap( r, m, x, c( bwd_min, bwd_max ),niter, method="deviance")
pfit <- locglmfit( xfit, r, m, x, bwd )$pfit
# Plot the fitted curve
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
lines(xfit, pfit )

## End(Not run)
```

bandwidth_cross_validation

Cross-validation bandwidth for local polymnomial estimator of a psychometric function

Description

This function finds the cross-validation bandwidth for a local polynomial estimate of the psychometric function with specified guessing and lapsing rates.

Usage

```
bandwidth_cross_validation( r, m, x, H, link = "logit", guessing = 0,
                            lapsing = 0, K = 2, p = 1, ker = "dnorm",
                            maxiter = 50, tol = 1e-6, method = "all")
```

Arguments

`r`	number of successes at points x
`m`	number of trials at points x
`x`	stimulus levels
`H`	search interval
`link`	(optional) name of the link function to be used; default is "logit"
`guessing`	(optional) guessing rate; default is 0
`lapsing`	(optional) lapsing rate; default is 0
`K`	(optional) power parameter for Weibull and reverse Weibull link; default is 2
`p`	(optional) degree of the polynomial; default is 1
`ker`	(optional) kernel function for weights; default is "dnorm"
`maxiter`	(optional) maximum number of iterations in Fisher scoring; default is 50
`tol`	(optional) tolerance level at which to stop Fisher scoring; default is 1e-6
`method`	(optional) loss function to be used in cross-validation: choose from: "ISEeta", "ISE", "deviance"; by default all possible values are calculated

Value

`h` cross-validation bandwidth for the chosen "method"; if no "method" is specified, then it has three components: $pscale, $etascale and $deviance

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Find a cross-validation bandwidth
bwd_min <- min( diff( x ) )
bwd_max <- max( x ) - min( x )
bwd <- bandwidth_cross_validation( r, m, x, c( bwd_min, bwd_max ) )
bwd <- bwd$deviance # Choose the estimate based on cross-validated deviance
pfit <- locglmfit( xfit, r, m, x, bwd )$pfit
# Plot the fitted curve
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
lines(xfit, pfit )
```

bandwidth_plugin *Plug-in bandwidth for local polynomial estimator of a psychometric function*

Description

The function calculates an estimate of the AMISE optimal bandwidth for a local polynomial estimate of the psychometric function.

Usage

```
bandwidth_plugin( r, m, x, link = "logit", guessing = 0,
                   lapsing = 0, K = 2, p = 1, ker = "dnorm" )
```

Arguments

r	number of successes at points x
m	number of trials at points x
x	stimulus levels
link	(optional) name of the link function to be used; default is "logit"
guessing	(optional) guessing rate; default is 0
lapsing	(optional) lapsing rate; default is 0
K	(optional) power parameter for Weibull and reverse Weibull link; default is 2
p	(optional) degree of the polynomial; default is 1
ker	(optional) kernel function for weights; default is "dnorm"

Value

h	plug-in bandwidth (ISE optimal on eta-scale)

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Find a plug-in bandwidth
bwd <- bandwidth_plugin( r, m, x)
pfit <- locglmfit( xfit, r, m, x, bwd )$pfit
# Plot the fitted curve
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
lines(xfit, pfit )
```

`binomfit_lims` *Parameter estimates for a psychometric function with FIXED guessing and lapsing rates*

Description

This function fits a binomial generalised linear model with fixed guessing and lapsing rates.

Usage

```
binomfit_lims( r, m, x, p = 1, link = "logit", guessing = 0, lapsing = 0, K = 2 )
```

Arguments

`r` number of successes at points x

`m` number of trials at points x

`x` stimulus levels

`p` (optional) degree of the polynomial; default is p = 1

`link` (optional) name of the link function; default is "logit"

`guessing` (optional) guessing rate; default is 0

`lapsing` (optional) lapsing rate; default is 0

`K` (optional) power parameter for Weibull and reverse Weibull link; default is 2

Value

`b` vector of estiamted coefficients for the linear part
`fit` glm object to be used in evaluation of fitted values

Examples

```
data("Carcagno")
x = Carcagno$x
r = Carcagno$r
m = Carcagno$m
plot( x, r / m, xlim = c( 1.95, 4.35 ), ylim = c( 0.24, 0.99 ), type = "p", pch="*" )
guess = 1/3; # guessing rate
laps = 0; # lapsing rate
val <- binomfit_lims( r, m, x, link = "probit", guessing = guess, lapsing = laps )
numxfit <- 199 # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Plot the fitted curve
pfit<-predict( val$fit, data.frame( x = xfit ), type = "response" )
lines(xfit, pfit )
```

binom_lims *Maximum likelihood parameter estimates for a psychometric function with guessing and lapsing rates*

Description

This function finds the maximum likelihood estimates of the parameters of the psychometric function with guessing and lapsing rates, only guessing rate, or only lapsing rate.

Usage

```
binom_lims( r, m, x, gl = "both", link = "logit", p = 1, K = 2, initval = NULL )
```

Arguments

`r` number of successes at points x

`m` number of trials at points x

`x` stimulus levels

`gl` (optional) indicator, calulate only guessing if "guessing", only lapsing if "lapsing" and both guessing and lapsing if "both"; default is "both"

`link` (optional) name of the link function; default is "logit"

`p` (optional) degree of the polynomial; default is 1

`K` (optional) power parameter for Weibull and reverse Weibull link; default is 2

`initval` (optional) initial value for guessing and lapsing; default is c(.01 .01) if guessing and rates are estimated, and .01 if only guessing or only lapsing rate is estimated

Value

`b` estimated coefficients for the linear part

`guessing` estimated guessing rate (if estimated)

`lapsing` estimated lapsing rate (if estimated)

`fit` glm object to be used in evaluation of fitted values

Examples

```
data("Baker_etal")
x = Baker_etal$x
r = Baker_etal$r
m = Baker_etal$m
plot( x, r / m, xlim = c( 0.16, 7.83 ), ylim = c( -0.01, 1.01 ), type = "p", pch="*" )
val <- binomfit_lims( r, m, x, link = "probit" )
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Plot the fitted curve
pfit<-predict( val$fit, data.frame( x = xfit ), type = "response" )
lines(xfit, pfit )
```

`binom_revweib` *Reverse Weibull model for the psychometric function*

Description

This function finds the maximum likelihood estimates of the parameters of the reverse Weibull model for the psychometric function.

Usage

```
binom_revweib( r, m, x, p = 1, initK = 2, guessing = 0, lapsing = 0 )
```

Arguments

`r`	number of successes at points x
`m`	number of trials at points x
`x`	stimulus levels
`p`	(optional) degree of the polynomial; default is 1
`initK`	(optional) initial value for K (power parameter in reverse Weibull model); default is 2
`guessing`	(optional) guessing rate; default is 0
`lapsing`	(optional) lapsing rate; default is 0

Value

`b`	vector of estimated coefficients for the linear part
`K`	estiamte of the power parameter in the reverse Weibull model
`fit`	glm object to be used in evaluation of fitted values

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
val <- binom_revweib( r, m, x )
# Plot the fitted curve
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
pfit <- predict( val$fit, data.frame( x = xfit ), type = "response" )
lines(xfit, pfit, col = "green" )
```

binom_weib *Weibull model for the psychometric function*

Description

This function finds the maximum likelihood estimates of the parameters of the Weibull model for the psychometric function.

Usage

```
binom_weib( r, m, x, p = 1, initK = 2, guessing = 0, lapsing = 0 )
```

Arguments

r	number of successes at points x
m	number of trials at points x
x	stimulus levels
p	(optional) degree of the polynomial; default is 1
initK	(optional) initial value for K (power parameter in Weibull model); default is 2
guessing	(optional) guessing rate; default is 0
lapsing	(optional) lapsing rate; default is 0

Value

b	vector of estimated coefficients for the linear part
K	estiamte of the power parameter in the Weibull model
fit	glm object to be used in evaluation of fitted values

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
val <- binom_weib( r, m, x )
# Plot the fitted curve
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
pfit <- predict( val$fit, data.frame( x = xfit ), type = "response" )
lines(xfit, pfit, col = "red" )
```

bootstrap_ci_sl *Bootstrap confidence interval for the slope of a psychometric function*

Description

Finds a bootstrap estimate of a confidence interval at a significance level alpha for the estimated slope for the local polynomial estimate of the psychometric function with guessing and lapsing rates. The confidence interval is based on bootstrap percentiles. See Efron & Tibshirani's "An introduction to the bootstrap", 1993

Usage

```
bootstrap_ci_sl( TH, r, m, x, N, h0, alpha = 0.05,
             X = (max(x)-min(x))*(0:999)/999+min(x), link = "logit", guessing = 0,
               lapsing = 0, K = 2, p = 1, ker = "dnorm", maxiter = 50, tol = 1e-6 )
```

Arguments

`TH`	required threshold level
`r`	number of successes at points x
`m`	number of trials at points x
`x`	stimulus levels
`N`	number of bootstrap replications; N should be at least 1000 for reliable results
`h0`	bandwidth
`alpha`	(optional) significance level of the confidence interval; default is 0.05
`X`	(optional) set of values at which estimates of the psychometric function for the slope estimation are to be obtained; if not given, 1000 equally spaced points from minimum to maximum of 'x' are used
`link`	(optional) name of the link function; default is "logit"
`guessing`	(optional) guessing rate; default is 0
`lapsing`	(optional) lapsing rate; default is 0
`K`	(optional) power parameter for Weibull and reverse Weibull link; default is 2
`p`	(optional) degree of the polynomial; default is 1
`ker`	(optional) kernel function for weights; default is "dnorm"
`maxiter`	(optional) maximum number of iterations in Fisher scoring; default is 50
`tol`	(optional) tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`ci`	confidence interval based on bootstrap percentiles
`sl0`	slope estimate

Examples

```
## Not run:
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
bwd_min <- min( diff( x ) )
bwd_max <- max( x ) - min( x )
bwd <- bandwidth_cross_validation( r, m, x, c( bwd_min, bwd_max ), method = "deviance" )
prob <- 0.5 # Required threshold level
alpha <- 0.05 # Significance level for the confidence intervals
# This might take a few minutes
niter <- 1000 # Note number of bootstrap iterations should be at least 1000
ci_sl <- bootstrap_ci_sl( prob, r, m, x, niter, bwd, alpha ) # Be patient, slow process

## End(Not run)
```

bootstrap_ci_th *Bootstrap confidence interval for a slope of a psychometric function*

Description

Finds a bootstrap estimate of a confidence interval at a significance level alpha for the estimated threshold for the local polynomial estimate of the psychometric function with guessing and lapsing rates. The confidence interval is based on bootstrap percentiles. See Efron & Tibshirani's "An introduction to the bootstrap", 1993

Usage

```
bootstrap_ci_th( TH, r, m, x, N, h0, alpha = 0.05,
              X = (max(x)-min(x))*(0:999)/999+min(x), link = "logit", guessing = 0,
                lapsing = 0, K = 2, p = 1, ker = "dnorm", maxiter = 50, tol = 1e-6 )
```

Arguments

`TH`	required threshold level
`r`	number of successes at points x
`m`	number of trials at points x
`x`	stimulus levels
`N`	number of bootstrap replications; N should be at least 1000 for reliable results
`h0`	bandwidth
`alpha`	(optional) significance level of the confidence interval; default is 0.05
`X`	(optional) set of values at which estimates of the psychometric function for the threshold estimation are to be obtained; if not given, 1000 equally spaced points from minimum to maximum of x are used

`link` (optional) name of the link function; default is "logit"

`guessing` (optional) guessing rate; default is 0

`lapsing` (optional) lapsing rate; default is 0

`K` (optional) power parameter for Weibull and reverse Weibull link; default is 2

`p` (optional) degree of the polynomial; default is 1

`ker` (optional) kernel function for weights; default is "dnorm"

`maxiter` (optional) maximum number of iterations in Fisher scoring; default is 50

`tol` (optional) tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`ci` confidence interval based on bootstrap percentiles
`th0` threshold estimate

Examples

```
## Not run:
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
bwd_min <- min( diff( x ) )
bwd_max <- max( x ) - min( x )
bwd <- bandwidth_cross_validation( r, m, x, c( bwd_min, bwd_max ), method = "deviance" )
prob <- 0.5 # Required threshold level
alpha <- 0.05 # Significance level for the confidence intervals
# This might take a few minutes
niter <- 1000 # Note number of bootstrap iterations should be at least 1000
ci_th <- bootstrap_ci_th( prob, r, m, x, niter, bwd, alpha ) # Be patient, slow process

## End(Not run)
```

`bootstrap_sd_sl` *Bootstrap standard deviation for the slope of a psychometric function*

Description

The function finds a bootstrap estimate of the standard deviation of the estimated slope for the local polynomial estimate of the psychometric function with guessing and lapsing rates.

Usage

```
bootstrap_sd_sl( TH, r, m, x, N, h0, X = (max(x)-min(x))*(0:999)/999+min(x),
                 link = "logit", guessing = 0, lapsing = 0, K = 2, p = 1,
                 ker = "dnorm", maxiter = 50, tol = 1e-6 )
```

Arguments

TH required threshold level

r number of successes at points x

m number of trials at points x

x stimulus levels

N number of bootstrap replications; N should be at least 200 for reliable results

h0 bandwidth

X (optional) set of values at which estimates of the psychometric function for the slope estimation are to be obtained; if not given, 1000 equally spaced points from minimum to maximum of x are used

link (optional) name of the link function; default is "logit"

guessing (optional) guessing rate; default is 0

lapsing (optional) lapsing rate; default is 0

K (optional) power parameter for Weibull and reverse Weibull link; default is 2

p (optional) degree of the polynomial; default is 1

ker (optional) kernel function for weights; default is "dnorm"

maxiter (optional) maximum number of iterations in Fisher scoring; default is 50

tol (optional) tolerance level at which to stop Fisher scoring; default is 1e-6

Value

sd bootstrap estimate of the standard deviation of the slope estimator
sl0 slope estimate

Examples

```
## Not run:
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
bwd_min <- min( diff( x ) )
bwd_max <- max( x ) - min( x )
bwd <- bandwidth_cross_validation( r, m, x, c( bwd_min, bwd_max ), method = "deviance" )
prob <- 0.5 # Required threshold level
# This might take a few minutes
niter <- 200 # Note number of bootstrap iterations should be at least 200
sd_sl <- bootstrap_sd_sl( prob, r, m, x, niter, bwd ) # Be patient, slow process

## End(Not run)
```

`bootstrap_sd_th` *Bootstrap standard deviation for the slope of a psychometric function*

Description

The function finds a bootstrap estimate of the standard deviation of the estimated threshold for the local polynomial estimate of the psychometric function with guessing and lapsing rates.

Usage

```
bootstrap_sd_th( TH, r, m, x, N, h0, X = (max(x)-min(x))*(0:999)/999+min(x),
                 link = "logit", guessing = 0, lapsing = 0, K = 2, p = 1,
                 ker = "dnorm", maxiter = 50, tol = 1e-6 )
```

Arguments

`TH`	required threshold level
`r`	number of successes at points x
`m`	number of trials at points x
`x`	stimulus levels
`N`	number of bootstrap replications; N should be at least 200 for reliable results
`h0`	bandwidth
`X`	(optional) set of values at which estimates of the psychometric function for the threshold estimation are to be obtained; if not given, 1000 equally spaced points from minimum to maximum of 'x' are used
`link`	(optional) name of the link function; default is "logit"
`guessing`	(optional) guessing rate; default is 0
`lapsing`	(optional) lapsing rate; default is 0
`K`	(optional) power parameter for Weibull and reverse Weibull link; default is 2
`p`	(optional) degree of the polynomial; default is 1
`ker`	(optional) kernel function for weights; default is "dnorm"
`maxiter`	(optional) maximum number of iterations in Fisher scoring; default is 50
`tol`	(optional) tolerance level at which to stop Fisher scoring; default is 1e-6

Value

`sd`	bootstrap estimate of the standard deviation of the threshold estimator
`th0`	threshold estimate

Examples

```
## Not run:
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
bwd_min <- min( diff( x ) )
bwd_max <- max( x ) - min( x )
bwd <- bandwidth_cross_validation( r, m, x, c( bwd_min, bwd_max ), method = "deviance" )
prob <- 0.5 # Required threshold level
# This might take a few minutes
niter <- 200 # Note number of bootstrap iterations should be at least 200
th_sl <- bootstrap_sd_sl( prob, r, m, x, niter, bwd ) # Be patient, slow process

## End(Not run)
```

Carcagno *Discrimination of pitch*

Description

The subject had to identify the interval containing a tone whose fundamental frequency was different from that in the other two intervals. The symbols in the figure below show the proportion of correct responses as the difference between the tones varied. There were 3–49 trials at each stimulus level. See https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example03.html

Usage

```
data("Carcagno")
```

Format

A data frame with 8 rows and 3 columns.

`x` stimulus level

`r` number of successes

`m` number of trials

References

Unpublished data from S. Carcagno, Lancaster University, July 2008.

Examples

```
data("Carcagno")
x = Carcagno$x
r = Carcagno$r
m = Carcagno$m
plot( x, r / m, xlim = c( 1.95, 4.35 ), ylim = c( 0.24, 0.99 ), type = "p", pch="*" )
```

comploglog_link *Complementary log-log link with guessing and lapsing rates Creates a complementary log-log link function; the guessing rate and lapsing rate are fixed, hence link is a function of only one variable.*

Description

Complementary log-log link with guessing and lapsing rates Creates a complementary log-log link function; the guessing rate and lapsing rate are fixed, hence link is a function of only one variable.

Usage

```
comploglog_link(guessing = 0, lapsing = 0)
```

Arguments

guessing	(optional) guessing rate; default is 0
lapsing	(optional) lapsing rate; default is 0

Value

link complementary log-log link for use in all GLM functions

deviance2 *Deviance of a psychometric function*

Description

This function calculates the deviance for the fitted values of the psychometric function pfit.

Usage

```
deviance2( r, m, pfit )
```

Arguments

r	number of successes
m	number of trials
pfit	fittd values

Value

D deviance

Examples

```
data("Carcagno")
x = Carcagno$x
r = Carcagno$r
m = Carcagno$m
plot( x, r / m, xlim = c( 1.95, 4.35 ), ylim = c( 0.24, 0.99 ), type = "p", pch="*" )
guess = 1/3; # guessing rate
laps = 0; # lapsing rate
val <- binomfit_lims( r, m, x, link = "probit", guessing = guess, lapsing = laps )
pfit<-predict( val$fit, data.frame( x = x ), type = "response" )
d2 = deviance2( r, m, pfit )
```

Levi_Tripathy *Visual detection of path deviation*

Description

The subject was presented with the image of a dot moving rightwards on a linear path until it reached the midline of the display, when it changed direction either upwards or downwards. The subject had to indicate the direction. The symbols in the figure below show the proportion of correct responses in 30 trials as the deviation varied from –3 to 3 units. See https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example02.html

Usage

```
data("Levi_Tripathy")
```

Format

A data frame with 7 rows and 3 columns.

x stimulus level

r number of successes

m number of trials

References

Levi, D. M. & Tripathy, S. P. “Is the ability to identify deviations in multiple trajectories compromised by amblyopia?”, Journal of Vision, 6(12), 1367-1379, 2006.

Examples

```
data("Levi_Tripathy")
x = Levi_Tripathy$x
r = Levi_Tripathy$r
m = Levi_Tripathy$m
plot( x, r / m, xlim = c( -2.87, 2.87 ), ylim = c( 0.03, 0.97 ), type = "p", pch="*" )
```

locglmfit *Local polynomial estimator of a psychometric function*

Description

Local polynomial estimator for the psychometric function and eta function (psychometric function transformed by link) for binomial data; also returns the hat matrix H.

Usage

```
locglmfit( xfit, r, m, x, h, returnH = FALSE, link = "logit",
               guessing = 0, lapsing = 0, K = 2, p = 1,
               ker = "dnorm", maxiter = 50, tol = 1e-6 )
```

Arguments

xfit	points at which to calculate the estimate pfit
r	number of successes at points x
m	number of trials at points x
x	stimulus levels
h	bandwidth(s)
returnH	(optional) logical, if TRUE then hat matrix is calculated; default is FALSE
link	(optional) name of the link function; default is 'logit'
guessing	(optional) guessing rate; default is 0
lapsing	(optional) lapsing rate; default is 0
K	(optional) power parameter for Weibull and reverse Weibull link; default is 2
p	(optional) degree of the polynomial; default is 1
ker	(optional) kernel function for weights; default is 'dnorm'
maxiter	(optional) maximum number of iterations in Fisher scoring; default is 50
tol	(optional) tolerance level at which to stop Fisher scoring; default is 1e-6

Value

pfit	value of the local polynomial estimate at points xfit
etafit	estimate of eta (link of pfit)
H	hat matrix (OPTIONAL)

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
```

```
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Find a plug-in bandwidth
bwd <- bandwidth_plugin( r, m, x)
pfit <- locglmfit( xfit, r, m, x, bwd )$pfit
# Plot the fitted curve
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
lines(xfit, pfit )
```

logit_link *Logit link with guessing and lapsing rates Creates a complementary logit link function; the guessing rate and lapsing rate are fixed, hence link is a function of only one variable.*

Description

Logit link with guessing and lapsing rates Creates a complementary logit link function; the guessing rate and lapsing rate are fixed, hence link is a function of only one variable.

Usage

```
logit_link(guessing = 0, lapsing = 0)
```

Arguments

guessing	(optional) guessing rate; default is 0
lapsing	(optional) lapsing rate; default is 0

Value

link logit link for use in all GLM functions

loglog_link *Log-log link with guessing and lapsing rates*

Description

Creates a log-log link function; the guessing rate and lapsing rate are fixed, hence link is a function of only one variable.

Usage

```
loglog_link(guessing = 0, lapsing = 0)
```

Arguments

`guessing`	(optional) guessing rate; default is 0
`lapsing`	(optional) lapsing rate; default is 0

Value

link log-log link for use in all GLM functions

`Miranda_Henson` *Detection of a flash of light*

Description

A flash of light of variable intensity was presented repeatedly at a fixed location in the visual field of a subject who reported whether the flash was visible. There were 3–20 trials at each stimulus level. See https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example01R.html

Usage

```
data("Miranda_Henson")
```

Format

A data frame with 10 rows and 3 columns.

`x` stimulus level

`r` number of successes

`m` number of trials

References

Miranda, M. A. & Henson, D. B. “Perimetric sensitivity and response variability in glaucoma with single-stimulus automated perimetry and multiple-stimulus perimetry with verbal feedback”, Acta Ophthalmologica, 86, 202-206, 2008.

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
plot( x, r / m, xlim = c( 0.1, 1.302 ), ylim = c( 0.0165, 0.965 ), type = "p", pch="*" )
```

Nascimento_etal *Discrimination of image approximations*

Description

The subject was shown an image of a natural scene and an approximation of this image based on principal component analysis. The task was to distinguish between the images. The symbols in the figure below show the proportion of correct responses as a function of number of components in the approximation. There were 200 trials at each level pooled over a range of natural scenes. See https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example06R.html

Usage

```
data("Nascimento_etal")
```

Format

A data frame with 8 rows and 3 columns.

`x` stimulus level

`r` number of successes

`m` number of trials

References

Nascimento, S.M.C., Foster, D.H., & Amano, K. “Psychophysical estimates of the number of spectral-reflectance basis functions needed to reproduce natural scenes”, Journal of the Optical Society of America A-Optics Image Science and Vision, 22 (6), 1017-1022, 2005.

Examples

```
data("Nascimento_etal")
x = Schofield_etal$x
r = Schofield_etal$r
m = Schofield_etal$m
plot( x, r / m, xlim = c( 1.17, 7.8 ), ylim = c( 0.47, 1.03 ), type = "p", pch="*" )
```

probit_link *Probit link with guessing and lapsing rates*

Description

Creates a complementary log-log link function; the guessing rate and lapsing rate are fixed, hence link is a function of only one variable.

Usage

```
probit_link(guessing = 0, lapsing = 0)
```

Arguments

`guessing`	(optional) guessing rate; default is 0
`lapsing`	(optional) lapsing rate; default is 0

Value

link probit link for use in all GLM functions

Schofield_etal *Induction of a visual motion aftereffect*

Description

The subject was presented with a moving adaptation stimulus, followed by a test stimulus. The symbols in the figure below show the proportion of responses in which the subject indicated motion of the test stimulus in the same direction as the adapting stimulus, either up or down, as a function of relative modulation depth. There were 10 trials at each stimulus level. See `https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example05R.html`

Usage

```
data("Schofield_etal")
```

Format

A data frame with 7 rows and 3 columns.

`x` stimulus level

`r` number of successes

`m` number of trials

References

Schofield, A. J., Ledgeway, T., & Hutchinson, C. V. “Asymmetric transfer of the dynamic motion aftereffect between first- and second-order cues and among different second-order cues”, Journal of Vision, 7(8), 1-12, 2007.

Examples

```
data("Schofield_etal")
x = Schofield_etal$x
r = Schofield_etal$r
m = Schofield_etal$m
plot( x, r / m, xlim = c( 2, 98.2 ), ylim = c( 0.02, 0.98 ), type = "p", pch="*" )
```

---

threshold_slope *Estimates of threshold and slope for a fitted psychometric function*

Description

This function finds the approximate value of x (=x_th) for which the value of the estimated psychometric function is equal to ’thresh’ and the approximate value of slope in x_th.

Usage

```
threshold_slope( pfit, xfit, thresh = 0.5 )
```

Arguments

pfit	estimated values of the psychometric function
xfit	stimulus levels at which the function was estimated
thresh	criterion level at which to estimate threshold; default is 0.5

Value

x_th	estimated threshold
slope	estimated value of slope, i.e. derivative of pfit at x_th

Examples

```
data("Miranda_Henson")
x = Miranda_Henson$x
r = Miranda_Henson$r
m = Miranda_Henson$m
numxfit <- 199; # Number of new points to be generated minus 1
xfit <- (max(x)-min(x)) * (0:numxfit) / numxfit + min(x)
# Find a plug-in bandwidth
bwd <- bandwidth_plugin( r, m, x)
pfit <- locglmfit( xfit, r, m, x, bwd )$pfit
prob <- 0.5 # Required threshold level
thr_sl <- threshold_slope( pfit, xfit, prob )
```

Xie_Griffin *Discrimination of "porthole" views of natural scenes*

Description

The subject was presented with a display split into two parts, one containing a pair of patches from the same image, the other a pair from different images, and the subject had to judge which pair came from the same image. The symbols in the figure below show the proportion of correct responses in 200 trials as a function of patch separation. See https://personalpages.manchester.ac.uk/staff/david.foster/software-modelfree/latest/examples/example04R.html

Usage

```
data("Xie_Griffin")
```

Format

A data frame with 10 rows and 3 columns.

`x` stimulus level

`r` number of successes

`m` number of trials

References

Xie, Y. & Griffin, L. D. “A ’portholes’ experiment for probing perception of small patches of natural images”, Perception, 36, 315, 2007.

Examples

```
data("Xie_Griffin")
x = Xie_Griffin$x
r = Xie_Griffin$r
m = Xie_Griffin$m
plot( x, r / m, xlim = c( 0.25, 8.76 ), ylim = c( 0.52, 0.99 ), type = "p", pch="*" )
```

Index