Package: quickpsy (via r-universe)

September 9, 2024

Type Package

Title Fits Psychometric Functions for Multiple Groups

Version 0.2.0

URL http://dlinares.org/quickpsy.html

Description Quickly fits and plots psychometric functions (normal, logistic, Weibull or any function defined by the user) for multiple groups.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, ggplot2, purrr, rlang, stats, tibble, tidyr, utils, nloptr

RoxygenNote 7.1.1

Depends R (>= 2.10)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository https://danilinares.r-universe.dev

RemoteUrl https://github.com/danilinares/quickpsy

RemoteRef HEAD

RemoteSha 1739338784c201762d7c812e8dbdfe47e3c39010

Contents

`akaike` *Calculates the Akaike Information Criterion* `akaike` *calculates the loglikelihoods.*

Description

Calculates the Akaike Information Criterion `akaike` calculates the loglikelihoods.

Usage

```
akaike(logliks)
```

Arguments

`logliks`	The `logliks` data frame from quickpsy.

`avbootstrap` *Creates bootstrap samples*

Description

`avbootstrap` creates bootstrap samples

Usage

```
avbootstrap(averages, ypred, bootstrap, B)
```

Arguments

`averages` The `averages` data frame from quickpsy.

`ypred` The `ypred` data frame from quickpsy.

`bootstrap` `"parametric"` performs parametric bootstrap; `"nonparametric"` performs nonparametric bootstrap; `"none"` does not perform bootstrap (default is `'parametric'`).

`B` number of bootstrap samples (default is 100 ONLY).

`create_psy_fun` *Creates the psychometric function* `create_psy_fun` *creates the psychometric function*

Description

Creates the psychometric function `create_psy_fun` creates the psychometric function

Usage

```
create_psy_fun(psy_fun, guess, lapses)
```

Arguments

`psy_fun` The shape of the function without guesses and lapses.

`guess` Value indicating the guess rate (leftward asymptote) γ (default is 0). If `TRUE`, the guess rate is estimated as the i + 1 parameter where i corresponds to the number of parameters of `fun`. If, for example, `fun` is a predefined shape with parameters p1 and p2, then the guess rate corresponds to parameter p3.

`lapses` Value indicating the lapse rate (rightward asymptote) λ (default is 0). If `TRUE`, the lapse rate is estimated as the i + 1 parameter where i corresponds to the number of parameters of `fun` plus one if the guess rate is estimated. If, for example, `fun` is a predefined shape with parameters p1 and p2, then the lapse rate corresponds to parameter p3. If the guess rate is also estimated, p3 will be the guess rate and p4 the lapse rate.

`cum_normal_fun` *Cumulative normal function*

Description

Cumulative normal function.

Usage

```
cum_normal_fun(x, p)
```

Arguments

x Vector of values of the explanatory variable.

p Vector of parameters p = c(mean, standard_deviation).

Value

Probability at each x.

See Also

inv_cum_normal_fun

Examples

```
library(ggplot2)
xseq <- seq(0,4,.01)
yseq <- cum_normal_fun(xseq, c(2, .5))
curve <- data.frame(x = xseq, y = yseq)
ggplot2::ggplot(curve, aes(x = x, y = y)) + geom_line()
```

devi *Calculates the deviance*

Description

devi calculates the deviance

Usage

```
devi(logliks, loglikssaturated, grouping_without_fun)
```

Arguments

logliks The logliks data frame from quickpsy.

loglikssaturated
The loglikssaturated data frame from quickpsy.

grouping_without_fun
The grouping_without_fun data frame from quickpsy.

get_functions *Predefined functions*

Description

getfunctions lists the predefined functions in quickpsy.

Usage

```
get_functions()
```

See Also

cum_normal_fun, logistic_fun, weibull_fun

inv_cum_normal_fun *Inverse cumulative normal function*

Description

Inverse cumulative normal function

Usage

```
inv_cum_normal_fun(prob, p)
```

Arguments

prob	Vector of probabilities.
p	Vector of parameters p = c(mean, standard_deviation).

Value

x at each probability. #' @seealso cum_normal_fun

Examples

```
library(ggplot2)
yseq <- seq(0, 1, .01)
xseq <- inv_cum_normal_fun(yseq, c(2, .5))
curve <- data.frame(x = xseq, y = yseq)
ggplot(curve, aes(x = x, y = y)) + geom_line()
```

`inv_logistic_fun` *Inverse logistic function*

Description

Inverse logistic function

Usage

```
inv_logistic_fun(q, p)
```

Arguments

`q`	Vector of probabilities.
`p`	Vector of parameters $p = c(\alpha, \beta)$.

Value

`x` at each probability.

See Also

`logistic_fun`

Examples

```
library(ggplot2)
yseq <- seq(0, 1, .01)
xseq <- inv_logistic_fun(yseq, c(2, 4))
curve <- data.frame(x = xseq, y = yseq)
ggplot(curve, aes(x = x, y = y)) + geom_line()
```

`inv_weibull_fun` *Inverse Weibull function*

Description

Inverse Weibull function

Usage

```
inv_weibull_fun(q, p)
```

Arguments

`q`	Vector of probabilities.
`p`	Vector of parameters p = c(alpha, beta).

Value

x at each probability.

See Also

weibull_fun

Examples

```
library(ggplot2)
yseq <- seq(0, 1, .01)
xseq <- inv_weibull_fun(yseq, c(2, 4))
curve <- data.frame(x = xseq, y = yseq)
ggplot(curve, aes(x = x, y = y)) + geom_line()
```

logistic_fun *Logistic function*

Description

Logistic function of the form $(1 + exp(-\beta * (x - \alpha)))^(-1)$

Usage

```
logistic_fun(x, p)
```

Arguments

x	Vector of values of the explanatory variable.
p	Vector of parameters $p = c(\alpha, \beta)$.

Value

Probability at each x.

See Also

inv_logistic_fun

Examples

```
library(ggplot2)
xseq <- seq(0, 4, .01)
yseq <- logistic_fun(xseq, c(2, 4))
curve <- data.frame(x = xseq, y = yseq)
ggplot(curve, aes(x = x, y = y)) + geom_line()
```

logliks *Calculates the loglikelihoods* logliks *calculates the loglikelihoods.*

Description

Calculates the loglikelihoods logliks calculates the loglikelihoods.

Usage

```
logliks(nll_fun, param, grouping_without_fun)
```

Arguments

nll_fun	The nll_fun data frame from quickpsy.
param	The param data frame from quickpsy.
grouping_without_fun	The grouping_without_fun data frame from quickpsy.

logliksaturated *Calculates the saturated loglikelihoods* logliks *calculates the saturatedloglikelihoods.*

Description

Calculates the saturated loglikelihoods logliks calculates the saturatedloglikelihoods.

Usage

```
logliksaturated(nll_fun_saturated, averages, grouping_without_fun)
```

Arguments

nll_fun_saturated	The nll_fun_saturated data frame from quickpsy.
averages	The averages data frame from quickpsy.
grouping_without_fun	The grouping_without_fun data frame from quickpsy.

`model_selection_aic` *Select models using the Akaike Information Criterion*

Description

The `p` indicates the relative probability of the models.

Usage

```
model_selection_aic(aic1, aic2)
```

Arguments

`aic1`	The `aic1` data frame from quickpsy for the first model.
`aic2`	The `aic2` data frame from quickpsy for the second model.

`model_selection_lrt` *Select models using the likelihood ratio test*

Description

Select models using the likelihood ratio test

Usage

```
model_selection_lrt(loglik1, loglik2, alpha = 0.05)
```

Arguments

`loglik1`	The `loglik` data frame from quickpsy for the first model.
`loglik2`	The `loglik` data frame from quickpsy for the second model.
`alpha`	The significance level.

plotcurves *Plot the curves*

Description

plotcurves plot the curves.

Usage

```
plotcurves(
  qp,
  panel = NULL,
  xpanel = NULL,
  ypanel = NULL,
  color = NULL,
  averages = TRUE,
  curves = TRUE,
  thresholds = TRUE,
  ci = TRUE
)
```

Arguments

qp	output from quickpsy
panel	Name of the variable to be split in panels.
xpanel	Name of the variable to be split in horizontal panels.
ypanel	Name of the variable to be split in vertical panels.
color	Name of the variable codded by color.
averages	If FALSE averaged probabilities are not plotted (default is TRUE).
curves	If FALSE curves are not plotted (default is TRUE)
thresholds	If FALSE thresholds are not plotted (default is TRUE)
ci	If FALSE confidence intervals are not plotted (default is TRUE)

plotpar *Plot the values of the parameters*

Description

plotpar plot the values of the parameters.

Usage

```
plotpar(
  qp,
  x = NULL,
  panel = NULL,
  xpanel = NULL,
  ypanel = NULL,
  color = NULL,
  geom = "bar",
  ci = T
)
```

Arguments

`qp`	output from quickpsy.
`x`	Name of the variable to displayed in the x-axis.
`panel`	Name of the variable to be split in panels.
`xpanel`	Name of the variable to be split in horizontal panels.
`ypanel`	Name of the variable to be split in vertical panels.
`color`	Name of the variable codded by color.
`geom`	If `'bar'` displays bars. If `'point'` displays points (default is `'bar'`).
`ci`	If `FALSE` confidence intervals are not plotted (default is `TRUE`).

Examples

```
fit <- quickpsy(qpdat, phase, resp, grouping = c("participant", "cond"), bootstrap = "none")
plot(fit)
plot(fit, color = cond)
plotpar(fit)
plotthresholds(fit, geom = "point")
```

`plotthresholds` *Plot the thresholds*

Description

`plotthresholds` plot the thresholds.

Usage

```
plotthresholds(
  qp,
  x = NULL,
  panel = NULL,
  xpanel = NULL,
  ypanel = NULL,
  color = NULL,
  geom = "bar",
  ci = T,
  sizeerrorbar = 0.5
)
```

Arguments

qp	output from quickpsy.
x	Name of the variable to displayed in the x-axis.
panel	Name of the variable to be split in panels.
xpanel	Name of the variable to be split in horizontal panels.
ypanel	Name of the variable to be split in vertical panels.
color	Name of the variable codded by color.
geom	If 'bar' displays bars.
ci	If FALSE confidence intervals are not plotted (default is TRUE).
sizeerrorbar	Line width of the error bars. If 'point' displays points (default is 'bar').

Examples

```
fit <- quickpsy(qpdat, phase, resp, grouping = c("participant", "cond"), bootstrap = "none")
plotthresholds(fit)
```

print.quickpsy *Print quickpsy objects*

Description

Print quickpsy objects

Usage

```
## S3 method for class 'quickpsy'
print(x, ...)
```

Arguments

x	A quickpsy object.
...	Other arguments

qpdat *Data set for demonstration*

Description

The data set is part of the data associated with the paper "Linares, D., & Lopez-Moliner, J. (2006). Perceptual asynchrony between color and motion with a single direction change. Journal of Vision, 6(9), 10-10".

Usage

```
qpdat
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1280 rows and 5 columns.

quickpsy *Fits psychometric functions*

Description

`quickpsy` fits, by direct maximization of the likelihood (Prins and Kingdom, 2010; Knoblauch and Maloney, 2012), psychometric functions of the form

$$\psi(x) = \gamma + (1 - \gamma - \lambda) * fun(x)$$

where γ is the guess rate, λ is the lapse rate and fun is a sigmoidal-shape function with asymppotes at 0 and 1.

Usage

```
quickpsy(
  d,
  x = x,
  k = k,
  n = NULL,
  grouping = c(),
  xmin = NULL,
  xmax = NULL,
  log = FALSE,
  fun = cum_normal_fun,
  parini = NULL,
  guess = 0,
  lapses = 0,
  prob = NULL,
  thresholds = TRUE,
  bootstrap = "parametric",
  B = 100,
  ci = 0.95,
  control = NULL,
  parinivector = NULL,
  paircomparisons = FALSE
)
```

Arguments

`d` Data frame with a tidy form in which each column corresponds to a variable and each row is an observation.

`x` Name of the explanatory variable.

`k` Name of the response variable. It could be the number of trials in which a yes-type response was given or a vector of 0s (no-type response) and 1s (yes-type response) indicating the response on each trial.

`n` Only necessary if `k` refers to the number of trials in which a yes-type response was given. It corresponds to the name of the variable indicating the total number of trials.

`grouping` Name of the grouping variables. It should be specified as `grouping = c("variable_name1", "variable_name2")`.

`xmin` Minimum value of the explanatory variable for which the curves should be calculated (the default is the minimum value of the explanatory variable).

`xmax` Maximum value of the explanatory variable for which the curves should be calculated (the default is the maximum value of the explanatory variable).

`log` If `TRUE`, the logarithm in base 10 of the explanatory variable is used to fit the curves (default is `FALSE`).

`fun` Name of the shape of the curve to fit. It could be a predefined shape (`cum_normal_fun`, `logistic_fun`, `weibull_fun`) or the name of a function introduced by the user. It could also be a data frame of functions we shared parameters. See the vignetter `functions`. (default is `cum_normal_fun`).

`parini` Initial parameters. quickpsy calculates default initial parameters for the predefined functions by linear modelling of the probit-transformed data. Otherwise, `parini` could be

- a vector of initial parameters
- a list of the form `list(c(par1min, par1max), c(par2min, par2max))` to constraint the lower and upper bounds of the parameters
- a dataframe specifiying the initial parameters for each condition with the same structure that the output `par`, but without the confidence intervals.
- a dataframe specifiying the lower (the column should be called parmin) and upper bound (the column should be called parmax) for each condition.

`guess` Value indicating the guess rate (leftward asymptote) γ (default is 0). If `TRUE`, the guess rate is estimated as the i + 1 parameter where i corresponds to the number of parameters of `fun`. If, for example, `fun` is a predefined shape with parameters p1 and p2, then the guess rate corresponds to parameter p3.

`lapses` Value indicating the lapse rate (rightward asymptote) λ (default is 0). If `TRUE`, the lapse rate is estimated as the `i + 1` parameter where `i` corresponds to the number of parameters of `fun` plus one if the guess rate is estimated. If, for example, `fun` is a predefined shape with parameters p1 and p2, then the lapse rate corresponds to parameter p3. If the guess rate is also estimated, p3 will be the guess rate and p4 the lapse rate.

`prob` Probability to calculate the threshold (default is `guess + .5 * (1 - guess)`).

`thresholds` If `FALSE`, thresholds are not calculated (default is `TRUE`).

`bootstrap` `'parametric'` performs parametric bootstrap; `'nonparametric'` performs nonparametric bootstrap; `'none'` does not perform bootstrap (default is `'parametric'`).

`B` number of bootstrap samples (default is 100 ONLY).

`ci` Bootstrap confidence intervals level based on percentiles (default is .95).

`control` `control` argument of the `optim` function.

`parinivector` A optional vector of initials parameters when the lower and the upper bounds of the parameter are specified.

`paircomparisons` If `TRUE` bootstrap paircomparions of the parameters are performed. Default is `FALSE`

Value

A list containing the following components:

- `x, k, n`
- `grouping` The grouping variables.
- `funname` String with the name of the shape of the curve.
- `psyfunguesslapses` Curve including guess and lapses.
- `limits` Limits of the curves.
- `parini` Initial parameters.
- `ypred` Predicted probabilities at the values of the explanatory variable.
- `curves` Psychometric curves.
- `par` Fitted parameters and its confidence intervals.
- `parcomnparisons` Pair-wise comparisons of the parameters to assess whether two parameters are significantly different using bootstrap. Specifically, the parameter bootstrap samples for each of the two conditions are substrated and then it is considered whether zero was within the confidence interval level of the distributions of differences.
- `curvesbootstrap` Bootstrap psychometric curves.
- `thresholds` Thresholds and its confidence intervals.
- `thresholdscomparisons` Pair-wise comparisons of the thresholds.
- `logliks` Log-likelihoods of the model.
- `loglikssaturated` Log-likelihoods of the saturated model.
- `deviance` Deviance of the model and the p-value calculated by using the chi-square distribution and bootstraping.

- `aic` AIC of the model defined as

  $$-2 * loglik + 2 * k$$

  where k is the number of parameters of the model.

References

Burnham, K. P., & Anderson, D. R. (2003). Model selection and multimodel inference: a practical information-theoretic approach. Springer Science & Business Media.

Knoblauch, K., & Maloney, L. T. (2012). Modeling Psychophysical Data in R. New York: Springer.

Prins, N., & Kingdom, F. A. A. (2016). Psychophysics: a practical introduction. London: Academic Press.

Examples

```
library(quickpsy)
fit <- quickpsy(qpdat, phase, resp,
grouping = c("participant", "cond"), bootstrap = "none")
plot(fit)
plot(fit, color = cond)
plotpar(fit)
plotthresholds(fit, geom = "point")
```

`quickreadfiles` *Reads several files*

Description

`quickreadfiles` builts a data frame from several txt files. It assumes that in each file, the first row has the names of the variables.

Usage

```
quickreadfiles(path = getwd(), extension = "txt", ...)
```

Arguments

`path`	Path of the file (default is the working directory).
`extension`	Specify whether the file extension is 'txt' or 'csv'.
`...`	arguments of the form name_var = c('value1', 'value2',..). A new column with variable name name_var is addes to the data frame.

Examples

```
# download the 3 files in
# https://github.com/danilinares/quickpsy/tree/master/inst/extdata/example1
# and add them to your working directory
# dat <- quickreadfiles(subject = c('aa', 'bb', 'cc'), session = c('1', '2'))
# fit <- quickpsy(dat, phase, resp, grouping=.(subject), lapses = T, guess = T)
# plotcurves(fit)
```

weibull_fun *Weibull function*

Description

Weibull function of the form $(1 - exp(-(x/\alpha)^\beta)$

Usage

```
weibull_fun(x, p)
```

Arguments

x	Vector of values of the explanatory variable.
p	Vector of parameters $p = c(\alpha, \beta)$.

Value

Probability at each x.

Examples

```
library(ggplot2)
xseq <- seq(0, 4, .01)
yseq <- weibull_fun(xseq, c(2, 4))
curve <- data.frame(x = xseq, y = yseq)
ggplot(curve, aes(x = x, y = y)) + geom_line()
```

x_seq *Creates sequences of x's*

Description

Creates sequences of x's

Usage

```
x_seq(limits, x, grouping)
```

Arguments

limits	The limits data frame from quickpsy.
x	The x data frame from quickpsy.
grouping	The grouping data frame from quickpsy.

ypred *Predicted probabilities*

Description

ypred calculates the predicted probabilities at the values of the explanatory variable.

Usage

```
ypred(
  averages,
  param,
  psych_fun,
  x_str,
  log,
  grouping,
  grouping_without_fun,
  grouping_fun,
  funname,
  guess,
  lapses
)
```

Arguments

averages	The "averages" data frame from quickpsy.
param	The "param" data frame from quickpsy.
psych_fun	The "psych_fun" data frame from quickpsy.
x_str	The "x_str" data frame from quickpsy.
log	The "log" data frame from quickpsy.
grouping	The "grouping" data frame from quickpsy.
grouping_without_fun	The "grouping_without_fun" data frame from quickpsy.
grouping_fun	The "grouping_fun" data frame from quickpsy.
funname	The funname data frame from quickpsy.
guess	Argument guess
lapses	Argument lapses

Index