Package ‘MixedPsy’

November 16, 2017

Version 1.0.0

Title Statistical Tools for the Analysis of Psychophysical Data

Author Alessandro Moscatelli [aut, cre] (0000-0001-6269-4536),
          Priscilla Balestrucci [aut] (0000-0002-5764-9439)

Maintainer Alessandro Moscatelli <moskante@gmail.com>

Description Tools for the analysis of psychophysical data. This package allows to estimate
            the Point of Subjective Equivalence (PSE)
            and the Just Noticeable Difference (JND), either from a psychometric function or from a Generalized
            Linear Mixed Model (GLMM).
            Additionally, the package allows plotting the fitted models and the response data, simulating psy-
            chometric functions of different shapes, and simulating data sets.
            For a description of the use of GLMMs applied to psychophysical data, refer to Moscatelli et al. (2012), <doi:10.1167/12.11.26>.

Depends R (>= 3.1.0)

Imports beesr, boot, brglm, lme4, Matrix, mnormt

LazyData true

License GPL (>= 2)

URL https://mixedpsychophysics.wordpress.com

BugReports https://github.com/moskante/MixedPsy/issues

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-11-16 12:58:07 UTC

R topics documented:

MixDelta .......................................................... 2
MixPlot .......................................................... 3
MixTreatment ....................................................... 4
MixDelta

PSE/JND for Univariable GLMM Using Delta Methods

Description

Estimate the Point of Subjective Equivalence (PSE), the Just Noticeable Difference (JND) and the related Standard Errors for an univariate distribution by means of Delta Method.

Usage

MixDelta(xplode.obj, alpha = 0.05)

Arguments

xplode.obj an object of class xplode.obj (univariable GLMMs).

alpha significance level of the confidence interval. Default is 0.05.

Details

MixDelta estimates PSE and JND of a univariable psychometric function (object of class "glm"). The method only applies to univariable GLMMs having a probit link function. Use MixTreatment for multivariable GLMMs.

Value

MixDelta returns a list of length 1 including Estimate, Standard Error, Inferior and Superior Confidence Interval of PSE and JND. Confidence Intervals are computed as: Estimate + / − z(1 − (α/2)) * Std.Error.

Note

The function assumes that the first model coefficient is the intercept and the second is the slope. The estimate of the JND assumes a probit link function.
MixPlot

References


See Also

MixTreatment for univarible and multivariable GLMM. pseMer for bootstrap-based confidence intervals. xplode objects of class xplode.obj.

Examples

library(lme4)
data(vibro_exp3)
formula.mod <- cbind(faster, slower) ~ speed + (1 + speed | subject)
mod <- glmer(formula = formula.mod, family = binomial(link = "probit"), data = vibro_exp3[vibro_exp3$vibration == 0,])
define.mod <- list(pf = list(intercept = 1, slope = 2))
xplode.mod <- xplode(model = mod, name.cont = "speed", define.pf = define.mod)
pse.jnd <- MixDelta(xplode.mod)

MixPlot

Plotting univariable GLMM

Description

Plot binomial data and the fitted GLMM (object of class xplode).

Usage

MixPlot(xplode.obj, pf = 1, p05line = F, x.range, x.ref, col = F, x.label = "Stimulus Intensity", y.label = "Predicted Response")

Arguments

xplode.obj an object of class xplode
pf integer: for multivariable GLMM including one factorial predictor, the level number to be plotted
p05line logical, should be an horizontal and a vertical line added? the horizontal line is fixed at P(Y = 1) = 0.5.
x.range a vector of length two specifying the range for model predictions
x.ref if p05line = T, this is the position of the vertical line on the x axis
col logical, if TRUE a different color will be used for different clusters/participants
x.label, y.label label for the x and the y axes. If not specified, x.labels = "Stimulus Intensity", y.label = "Predicted Response"
Value

a data.frame object including the intercept and slope for each participant (algebraic sum of the fixed effects and the modes of the random effects) and the color number for the plot.

Note

The function is currently only working with GLMM including maximum three random effects (random intercept, random slope and covariance of the two)

See Also

xplode objects of class xplode.obj.

Examples

library(lme4)
data(vibro_exp3)
formula.mod <- cbind(faster, slower) ~ speed + (1 + speed | subject)
mod <- glmer(formula = formula.mod, family = binomial(link = "probit"),
data = vibro_exp3[vibro_exp3$vibration == 0,])
define.mod <- list(pf1 = list(intercept = 1, slope = 2))
xplode.mod <- xplode(model = mod, name.cont = "speed", define.pf = define.mod)
myplot <- MixPlot(xplode.mod, pf = 1, p05line = FALSE, x.ref = 8.5, x.range = c(1,16),
col = TRUE, x.label = "Stimulus Speed", y.label = "Predicted Response")

MixTreatment PSE/JND for Multivariable GLMM Using Delta Methods

Description

Estimate the Point of Subjective Equivalence (PSE), the Just Noticeable Difference (JND) and the related Standard Errors for a multivariate distribution by means of Delta Method. The method applies to multivariable GLMM having a probit link function. The function is based on a recursive use of glmer and MixDelta

Usage

MixTreatment(xplode.obj, datafr)

Arguments

xplode.obj an object of class xplode.obj. The fitted model (object of class "merMod") from xplode.obj includes one continuous predictor and one factorial predictor.
datafr the data frame fitted with the GLMM model
**Details**

The function `MixTreatment` is based on a recursive use of `glmer` and `PsychDelta` to multivariable GLMM including continuous and factorial predictors. The same caveats of `PsychDelta` apply (e.g., confidence interval based on normality assumption).

**Value**

A list, whose length is equal to the levels of the factorial predictor. Each cell of the list is equal to the output of `delta.psy.probit` applied to a multivariable model whose baseline is level i of the factorial predictor.

**References**


**See Also**

`glmer` for Generalized Linear Mixed Models (including random effects). `mixdelta` for univariable model with delta method. `pseMer` for bootstrap-based confidence intervals.

**Examples**

```r
library(lme4)
data(vibro_exp3)
formula.mod <- cbind(faster, slower) ~ speed * vibration + (1 + speed| subject)
mod <- glmer(formula = formula.mod, family = binomial(link = "probit"), data = vibro_exp3)
xplode.mod <- xplode(model = mod, name.cont = "speed", name.factor = "vibration")
MixTreatment(xplode.mod, vibro_exp3)
```

---

**pseMer**

**PSE/JND for GLMM Using Bootstrap Methods**

**Description**

Estimates the Point of Subjective Equivalence (PSE), the Just Noticeable Difference (JND) and the related Standard Errors by means of Bootstrap Method.

**Usage**

```r
pseMer(mer.obj, B = 200, FUN = NULL, alpha = 0.05, ci.type = c("norm", "basic", "perc"), beep = f)
```
Arguments

- `mer.obj`: An object of class "merMod".
- `B`: integer: the number of bootstrap samples.
- `FUN`: An optional, custom made function to specify the required parameters to be estimated. If NULL, `pseMer()` will estimate the PSE and the JND of a univariable GLMM.
- `alpha`: Significance level of the confidence interval.
- `ci.type`: A vector of character strings representing the type of intervals required. The value should be any subset of the values c("norm","basic","stud","perc","bca") or simply "all" which will compute all five types of intervals. "perc" should be always included for the summary table.
- `beep`: Logical. If TRUE, a "ping" sound alerts that the simulation is complete.

Details

`pseMer` estimates PSE and JND (and additional user defined parameters) from a fitted GLMM model (class "merMod"). The "ping" sound is provided by `beep` function from the `beepr` package.

Value

`pseMer` returns a list of length 3 including a summary table (Estimate, Standard Error, Inferior and Superior Confidence Interval of the parameters) and the output of `bootMer` and `boot.ci` functions, for further analyses. Confidence Intervals in the summary table are based on the percentile method.

Note

A first custom function was written in 2012 for the non-CRAN package MERpsychophysics, based on the algorithm in Moscatelli et al. (2012). The current function is a simple wrapper of `lme4::bootMer()` and `boot::boot.ci()` functions.

Increasing the number of bootstrap samples (`B`) makes the estimate more reliable. However, this will also increase the duration of the computation.

References


See Also

`bootMer` from `lme4` package and `boot.ci` from `boot` package.
Examples

```r
## example 1: estimate pse/jnd of a univariable GLMM
library(lme4)
data(vibro_exp3)
formula.mod1 <- cbind(faster, slower) ~ speed + (1 + speed | subject)
mod1 <- glmer(formula = formula.mod1, family = binomial(link = "probit"),
              data = vibro_exp3[vibro_exp3$vibration == 0,])

BootEstim.1 <- pseMer(mod1, B = 100, ci.type = c("perc"))

## example 2: specify custom parameters for bootstrap estimation of a
## multivariate model
formula.mod2 <- cbind(faster, slower) ~ speed * vibration + (1 + speed | subject)
mod2 <- glmer(formula = formula.mod2, family = binomial(link = "probit"),
              data = vibro_exp3)

fun2mod = function(mer.obj){
  # allocate space: 4 parameters (jnd_0Hz, jnd_32Hz, pse_0Hz, pse_32Hz) j
  jndpse = vector(mode = "numeric", length = 4)
  names(jndpse) = c("jnd_0Hz", "jnd_32Hz", "pse_0Hz", "pse_32Hz")
  jndpse[1] = qnorm(0.75)/fixef(mer.obj)[2] # jnd_0Hz
  jndpse[2] = qnorm(0.75)/(fixef(mer.obj)[2] + fixef(mer.obj)[4]) # jnd_32Hz
  jndpse[3] = -fixef(mer.obj)[1]/fixef(mer.obj)[2] # pse_0Hz
  jndpse[4] = -(fixef(mer.obj)[1] + fixef(mer.obj)[3])/(fixef(mer.obj)[2] + fixef(mer.obj)[4]) # pse_32Hz
  return(jndpse)
}

BootEstim.2 = pseMer(mod2, B = 100, FUN = fun2mod)
```

**PsychDelta**

PSE/JND for univariable GLM Using Delta Method

**Description**

Estimate the Point of Subjective Equivalence (PSE), the Just Noticeable Difference (JND) and the related Standard Errors by means of Delta Method. The method only applies to univariable GLMs (psychometric functions) having a `probit` link function.

**Usage**

`PsychDelta(model, alpha = 0.05)`

**Arguments**

- `model` the fitted psychometric function. An object of class "glm".
- `alpha` significance level of the confidence interval.
PsychFunction

Fitting and Plotting Psychometric Functions

Details

PsychDelta estimates PSE and JND of a univariable psychometric function (object of class "glm").

Value

PsychDelta returns a matrix including Estimate, Standard Error, Inferior and Superior Confidence Interval of PSE and JND. Confidence Intervals are computed as: \( \text{Estimate} \pm z(1 - (\alpha/2)) \times \text{Std.Error} \).

Note

The function assumes that the first model coefficient is the intercept and the second is the slope. The estimate of the JND assumes a probit link function.

References


See Also

glm for for Generalized Linear Models (without random effects) and glmer for Generalized Linear Mixed Models (including random effects). mixDelta and mixTreatment for univarible and multivariable GLMM, respectively (object of class "merMod"). pseMer for bootstrap-based confidence intervals.

Examples

```r
#load simulated data
data(simul_data)
#fit a glm (probit link)
model.glm = glm(formula = cbind(Longer, Total - Longer) ~ X,
family = binomial(link = "probit"), data = simul_data)
PsychDelta(model.glm)
```

Examples

```r
#load simulated data
data(simul_data)
#fit a glm (probit link)
model.glm = glm(formula = cbind(Longer, Total - Longer) ~ X,
family = binomial(link = "probit"), data = simul_data)
 PsychDelta(model.glm)
```
Usage

PsychFunction(ps.formula, ps.link, ps.data, x.range = c(NA, NA), ps.x = NA, 
ps.lines = F, ps.col = "black", ps.lty = "dashed", ps.lwd = 1, 
br = F)

Arguments

- **ps.formula**: an object of class "formula", such as `cbind(yes, no) ~ X`
- **ps.link**: a link function for the binomial family of error distribution. See ‘Details’
- **ps.data**: a data frame including the variables in the model
- **x.range**: a vector of length two specifying the range for model predictions
- **ps.x**: optionally, a data frame in which to look for variables with which to predict. See ‘Details’
- **ps.lines**: logical. If TRUE, model predictions and confidence intervals of the PSE will be added to an existing plot
- **ps.col**: color of the lines to be plotted
- **ps.lty**: line type
- **ps.lwd**: line width
- **br**: logical. If TRUE, brglm is used if fitted values are equal to 0 or 1

Details

If `lines = TRUE`, the function draws model predictions on an existing plot. Only for univariable glm of the type `f(Y) ~ X`, where X is a continuous predictor. If `ps.x` is empty, the new data frame is a vector of length = 1000, whose range is specified from `x.range`. Std. Errors and 95% confidence intervals of the PSE and JND are estimated via Delta Methods, see Faraggi et al. (2003).

Value

a list including the fitted glm (or brglm), the estimate of PSE and JND and a flag to indicate if brglm was called.

References


See Also

- glm for for Generalized Linear Models. PsychShape for plotting psychometric function of given PSE and JND
Examples

# simulate data from a single participant
datafr.S1 <- PsySimulate(fixeff = c(-7.5, 0.0875), nsubject = 1, constant = TRUE)
# fit a glm (probit link)
model.glm = glm(formula = cbind(Longer, Total - Longer) ~ X,
family = binomial(link = "probit"), data = datafr.S1)

# fit psychometric function single-subject data and draw on existing plot
plot(Longer/Total ~ X, data = datafr.S1)
fit.S1 = PsychFunction(ps.formula = cbind(Longer, Total - Longer) ~ X,
ps.link = "probit", ps.data = datafr.S1,
   x.range = c(40, 120), ps.lines = TRUE)

PsychShape

Plotting Psychometric Functions given PSE and JND

Description

PsychShape() plots a psychometric function with known pse and jnd on an existing plot.

Usage

PsychShape(pse = 0, jnd, x.range = c(NA, NA), ps.link = "probit",
   ps.col = "black", ps.lwd = 1, ps.lty = "solid")

Arguments

pse, jnd the pse and the jnd of the desired psychometric function
x.range a vector of length two specifying the range of the function
ps.link a link function for the binomial family of error distribution (see Details).
ps.col color of the line to be plotted
ps.lwd line width
ps.lty line type

Details

PsychShape() can be used to visualize the predicted results of a psychophysical experiment or to
plot a fitted psychometric function whose values of pse and jnd are known. Currently only working
with probit and logit link function.

References

Moscatelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-
Science & Business Media.
PsySimulate

See Also

- glm for Generalized Linear Models.
- PsychFunction for estimation of PSE and JND.

Examples

```r
y = c(0, 1)
x = c(-40, 40)
plot(y ~ x, type = "n", bty = "n", lab = c(5, 3, 7))

psychshape(pse = 0, jnd = 6, x.range = c(-40, 40), ps.col = "gray", ps.lwd = 3)
psychshape(pse = 6, jnd = 6, x.range = c(-40, 40), ps.col = "black")
psychshape(pse = 6, jnd = 6, x.range = c(-40, 40), ps.col = "red", ps.link = "logit", ps.lwd = 3)
```

Description

Given the arrays of fixed and random effects, as well as the covariance, and the characteristic of the simulated experiment (i.e., ) the function simulates a dataset in which for each subject the following information is provided: the slope and intercept value of the psychometric function, and the simulated responses to the stimulus levels that fit that function.

Usage

```r
PsySimulate(fixeff = c(-7, 0.0875), raneff = c(2.4, -0.002, 2e-06),
nsubjects = 8, pps = 9, ntrials = 40, xint = c(40, 120),
constant = F)
```

Arguments

- **fixeff**: Array of fixed effects. First item is the intercept, second item is the slope.
- **raneff**: Array of random effects. First item is the intercept, second item is the covariance, third item is the slope.
- **nsubjects**: Number of subjects to simulate data for. Default is 8.
- **pps**: Number of stimulus levels. Default is 9.
- **ntrials**: Number of trials for each stimulus level. Default is 40.
- **xint**: Range of the stimulus interval. Default is c(40,120).
- **constant**: If set to FALSE, stimulus levels are randomly generated, uniformly distributed values within the selected interval. If constant = TRUE, the X interval is divided in intervals of constant length. Default is FALSE.

Value

The simulated dataset
Examples

```r
# simulate dataset (one subject)
datafr.S1 <- PsySimulate(nsubject = 1, constant = TRUE)
```

---

**simul_data**  
*A simulated psychophysical dataset*

---

**Description**

A dataset containing simulated data for 8 subjects. Created using PsySimulate(constant = T). The variables are as follows:

**Usage**

```r
data(simul_data)
```

**Format**

A data frame with 72 rows and 6 variables:

- **X**  Samples of the X interval c(40, 120)
- **Intercept**  Intercept of the psychometric function
- **Slope**  Slope of the psychometric function
- **Longer**  Number of trials in which response is judged "longer" than standard
- **Total**  Total number of trials for each sample of X interval
- **Subject**  Subject code (S1 to S8)

**See Also**

PsySimulate() for simulating dataframes.

---

**vibro_exp3**  
*Data from tactile discrimination task - EXP3*

---

**Description**

A dataset containing the response and stimuli from a tactile discrimination task (nine participants). In a forced-choice experiment, participants were required to discriminate the motion speed of a moving surface by touching it. Simultaneously with the motion stimulus, a 32Hz masking vibration occurred in half of the trials.

**Usage**

```r
data(vibro_exp3)
```
Format

A data frame with 72 rows and 6 variables:

- **speed**: a numeric vector giving the motion speed in cm/s of the moving surface
- **vibration**: a numeric vector giving the vibration frequency in Hz of the masking stimulus. Either 32Hz or 0 (no vibration - control condition)
- **faster**: The proportion of trials in which the comparison stimulus was judged as faster than the reference
- **slower**: The proportion of trials in which the comparison stimulus was judged as slower than the reference
- **subject**: subject id

Source

Original data were published in Dallmann et al. (2015).

References


---

xplode

*Extract values from a fitted GLMM object*

Description

Extract values from an object of class `merMod` (more specifically, from an object of subclass `glmerMod`).

Usage

```r
xplode(model, name.cont = NA, name.factor = NA, names.response = NA, define.pf = list(pfl = list(intercept = 1, slope = 2)))
```

Arguments

- **model**: The GLMM fitted with `glmer`. An object of class `merMod`.
- **name.cont**: A string providing the name of the continuous predictor, as in the formula object of the fitted model
- **name.factor**: A string providing the name of name of the categorical predictor, as in the formula object of the fitted model
- **names.response**: Optional. A string providing the name of name of the response variable, as in the formula object of the fitted model
- **define.pf**: Optional. Specifies which parameter pertains to the intercept and which to the slope in the formula object.
Details

For simplicity, several MixedPsy functions take as input an object of class xplode instead of an object of class "merMod". Most of these functions assume by default that the continuous predictor is entered first in the formula object. It is possible to use a different order, this requires to specify which parameter pertains to the intercept and which to the slope, by changing define.pf.

Value

An object of class "merMod" to be used with other MixedPsy functions.

See Also

merMod-class and glmer from package lme4 for objects of class "merMod", MixDelta, MixTreatment for use of objects of class xplode

Examples

```R
library(lme4)
datafr = PsySimulate(nsubjects = 10)
mod1 = glmer(formula = cbind(Length, Total - Longer) ~ X + (1 | Subject),
             family = binomial(link = "probit"), data = datafr)
xplode.mod1 = xplode(model = mod1, name.cont = "X")
MixDelta(xplode.mod1)
```
Index

*Topic **Bootstrap**
  pseMer, 5

*Topic **DeltaMethod**
  MixDelta, 2
  MixPlot, 3
  MixTreatment, 4
  PsychDelta, 7
  PsychFunction, 8

*Topic **GLMM**
  MixDelta, 2
  MixPlot, 3
  MixTreatment, 4
  pseMer, 5
  xplode, 13

*Topic **GLM**
  PsychDelta, 7
  PsychFunction, 8

*Topic **Multivariable**
  MixTreatment, 4
  pseMer, 5

*Topic **Plotting**
  MixPlot, 3
  PsychFunction, 8

*Topic **Psychometric**
  PsychFunction, 8

*Topic **Univariable**
  MixDelta, 2
  pseMer, 5
  PsychDelta, 7

*Topic **datasets**
  simul_data, 12
  vibro_exp3, 12
  xplode, 3, 4, 13

lme4, 14
merMod, 4, 6, 8, 13, 14
MixDelta, 2, 5, 14
MixPlot, 3
MixTreatment, 3, 4, 14
pseMer, 3, 5, 5, 8
PsychDelta, 7
PsychFunction, 8, 11
PsychShape, 9, 10
PsySimulate, 11
simul_data, 12
vibro_exp3, 12
xplode, 3, 4, 13

beep, 6
boot.ci, 6
bootMer, 6

glm, 8, 9, 11
glmer, 5, 8, 14