

# Package: MixedPsy (via r-universe)

October 25, 2024

**Version** 1.1.0

**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 in R. 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 psychometric 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).

**Depends** R (>= 3.5.0)

**Imports** beepr, boot, brglm, lme4, Matrix, mnormt, ggplot2

**LazyData** true

**License** GPL (>=2)

**URL** <https://mixedpsychophysics.wordpress.com>

**BugReports** <https://github.com/moskante/MixedPsy/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Repository** <https://moskante.r-universe.dev>

**RemoteUrl** <https://github.com/moskante/mixedpsy>

**RemoteRef** HEAD

**RemoteSha** c0067ed8752ea6532d948d0044ba6a07257e772d

## Contents

MixDelta	2
MixPlot	3
pseMer	4
PsychDelta	6
PsychFunction	8
PsychPlot	9
PsychShape	10
PsySimulate	12
simul_data	13
vibro_exp3	14
xplode	14
<b>Index</b>	<b>16</b>

---

MixDelta	<i>PSE/JND from GLMM Estimates using Delta Method</i>
----------	---

---

### Description

Estimate Points of Subjective Equivalence (PSE), Just Noticeable Differences (JND) and the related Standard Errors from a GLMM by means of delta method. The method applies to models with a *probit* link function, one continuous predictor, and one (optional) factorial predictor.

### Usage

```
MixDelta(xplode.obj, alpha = 0.05, p = 0.75)
```

### Arguments

xplode.obj	an object of class <code>xplode.obj</code> . The fitted model (object of class <code>merMod</code> , specifically of subclass <code>glmerMod</code> ) includes one continuous predictor and one (optional) factorial predictor.
alpha	significance level of the confidence intervals. Default is 0.05 (value for 95% confidence interval).
p	probability value relative to the JND upper limit. Default is 0.75 (value for 50% JND).

### Details

When the model includes a factorial predictor, the function is based on a recursive use of `glmer` and re-order of levels of the factorial predictor. The JND estimate assumes a *probit* link function.

### Value

A matrix including estimate, standard error, inferior and superior bounds of the confidence interval of PSE and JND. If a factorial predictor is included in the model, the function returns a list, each item containing a matrix for the estimates relative to a level of the predictor.

**Note**

The delta method is based on the assumption of asymptotic normal distribution of the parameters estimates. This may result in an incorrect variance estimation. For a more reliable (but more time-consuming) estimation based on bootstrap method, use [pseMer](#).

**References**

- Moscatelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision*, 12(11):26, 1-17. doi:10.1167/12.11.26
- Casella, G., & Berger, R. L. (2002). *Statistical inference* (2nd ed.). Pacific Grove, CA: Duxbury Press

**See Also**

[glmer](#) for fitting Generalized Linear Mixed Models. [xplode](#) for interfacing values from a fitted GLMM to MixedPsy functions. [pseMer](#) for bootstrap-based confidence intervals of psychometric parameters.

**Examples**

```
library(lme4)

#univariable GLMM (one continuous predictor)
mod.uni = glmer(formula = cbind(Longer, Total - Longer) ~ X + (1 | Subject),
family = binomial(link = "probit"), data = simul_data)
explode.uni = explode(model = mod.uni, name.cont = "X")
MixDelta(explode.uni)

#multivariable GLMM (one continuous and one factorial predictor)
mod.multi <- glmer(cbind(faster, slower) ~ speed * vibration + (1 + speed| subject),
family = binomial(link = "probit"), data = vibro_exp3)
explode.multi <- explode(model = mod.multi, name.cont = "speed", name.factor = "vibration")
MixDelta(explode.multi)
```

---

MixPlot

*Plot Individual Responses from GLMM*


---

**Description**

Plot response curve for each individual in a population sample, given a GLMM with one continuous predictor and one (optional) factorial predictor. If the factorial predictor is specified, the response is plotted separately for each individual and each predictor level.

**Usage**

```
MixPlot(xplode.obj, facet_by = NULL, showData = TRUE)
```

**Arguments**

<code>xplode.obj</code>	an object of class <code>xplode</code> .
<code>facet_by</code>	optional. A string specifying the name of the faceting variable (either the participant identification or the factorial predictor).
<code>showData</code>	logical, defines if proportion of binomial responses for each stimulus level are presented. Default is TRUE.

**Details**

If the model includes only a continuous predictor, the figure consist of a single panel, and each individual's response is assigned a different color. If a factorial predictor is included in the model, the faceting variable can be either the participant identification or the factorial predictor. By default, each panel shows an individual's response, different levels of the factorial predictor are coded by color.

**Value**

MixPlot returns a `ggplot` object.

**See Also**

`xplode` for objects of class `xplode`. `ggplot2` for creating data visualizations. `PsychPlot` for plotting a psychometric function from a GLM.

**Examples**

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

MixPlot(xplode.multi)
#alternative visualization
MixPlot(xplode.multi, facet_by = "vibration", showData = FALSE)
```

**Description**

Estimates the Point of Subjective Equivalence (PSE), the Just Noticeable Difference (JND) and the related Standard Errors by means of Bootstrap Method, given an object of class `merMod`.

## Usage

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

## Arguments

<code>mer.obj</code>	an object of class <code>merMod</code> .
<code>B</code>	integer. Number of bootstrap samples.
<code>FUN</code>	an optional, custom made function to specify the required parameters to be estimated. If <code>NULL</code> , <code>pseMer</code> estimates PSE and 50%JND of a univariable GLMM with a single intercept and slope.
<code>alpha</code>	significance level of the confidence intervals. Default is 0.05 (95% confidence interval).
<code>ci.type</code>	vector of character strings representing the type of intervals required. The value should be any subset of the values accepted by <code>boot.ci</code> : <code>c("norm","basic", "stud", "perc", "bca")</code> . Specify "all" for all five types of intervals. "perc" should be always included for the summary table.
<code>beep</code>	logical. If <code>TRUE</code> , a "ping" sound alerts that the simulation is complete. Default is <code>FALSE</code> .

## Details

`pseMer` estimates PSE and JND (and additional user defined parameters) from a fitted GLMM model (class `merMod`).

## Value

`pseMer` returns a list of length 3 including a summary table (estimate, inferior and superior bounds of the confidence interval), the output of `bootMer`, and that of `boot.ci`, 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 wrapper of function `bootMer` and `boot.ci`.

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

## References

- Moscatelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision*, 12(11):26, 1-17. doi:10.1167/12.11.26
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1), 51. <https://doi.org/10.18637/jss.v067.i01>

## See Also

[bootMer](#) and [boot.ci](#) for estimation of confidence intervals with the bootstrap method. [MixDelta](#) for confidence intervals with delta method.

## Examples

```
library(lme4)
#example 1: univariable GLMM
mod.uni = glmer(formula = cbind(Longer, Total - Longer) ~ X + (1 | Subject),
family = binomial(link = "probit"), data = simul_data)

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

#example 2: specify custom parameters for multivariable model
mod.multi <- glmer(cbind(faster, slower) ~ speed * vibration + (1 + speed| subject),
family = binomial(link = "probit"), data = vibro_exp3)

fun2mod = function(mer.obj){
#allocate space: 4 parameters (jnd_A, jnd_B, pse_A, pse_B)
jndpse = vector(mode = "numeric", length = 4)
names(jndpse) = c("pse_0", "pse_32", "jnd_0", "jnd_32")
jndpse[1] = -fixef(mer.obj)[1]/fixef(mer.obj)[2] #pse_0
jndpse[2] = -(fixef(mer.obj)[1]+fixef(mer.obj)[3])/(fixef(mer.obj)[2]+ fixef(mer.obj)[4]) #pse_0
jndpse[3] = qnorm(0.75)/fixef(mer.obj)[2] #jnd_0
jndpse[4] = qnorm(0.75)/(fixef(mer.obj)[2]+ fixef(mer.obj)[4]) #jnd_32
return(jndpse)
}

BootEstim.multi = pseMer(mod.multi, B = 100, FUN = fun2mod)
```

## Description

Estimate Point of Subjective Equivalence (PSE), Just Noticeable Difference (JND), and related Standard Errors of an individual participant by means of Delta Method. The method only applies to a GLM (object of class `glm`) with one continuous predictor and a *probit* link function.

**Usage**

```
PsychDelta(model.obj, alpha = 0.05, p = 0.75)
```

**Arguments**

model.obj	the fitted psychometric function. An object of class <code>glm</code> .
alpha	significance level of the confidence interval. Default is 0.05 (95% confidence interval).
p	probability value relative to the JND upper limit. Default is 0.75 (value for 50% JND).

**Details**

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

**Value**

PsychDelta returns a matrix including estimate, standard error, inferior and superior bounds of the confidence interval of PSE and JND. Confidence Intervals are computed as:  $Estimate + / - z(1 - (\alpha/2)) * Std.Error$ .

**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.

**References**

- Faraggi, D., Izikson, P., & Reiser, B. (2003). Confidence intervals for the 50 per cent response dose. *Statistics in medicine*, 22(12), 1977-1988. <https://doi.org/10.1002/sim.1368>
- Knoblauch, K., & Maloney, L. T. (2012). *Modeling psychophysical data in R* (Vol. 32). Springer Science & Business Media.
- Moscatelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision*, 12(11):26, 1-17. doi:10.1167/12.11.26

**See Also**

`glm` for fitting a Generalized Linear Model to a single-subject response. `glmer` for Generalized Linear Mixed Models (including fixed and random effects). `MixDelta` for estimating PSE and JND at a population level with delta method.

**Examples**

```
data.S1 <- subset(simul_data, Subject == "S1")
model.glm = glm(formula = cbind(Longer, Total - Longer) ~ X,
family = binomial(link = "probit"), data = data.S1)
PsychDelta(model.glm)
```

---

PsychFunction	<i>Psychometric Function and PSE/JND Parameters from Single-Subject Response</i>
---------------	--

---

### Description

Fit psychometric functions using `glm` or `brglm`. Estimate PSE, JND, and related confidence intervals with Delta Method.

### Usage

```
PsychFunction(ps.formula, ps.link, ps.data, br = F)
```

### Arguments

<code>ps.formula</code>	an object of class <code>formula</code> , such as <code>cbind(yes, no) ~ X</code>
<code>ps.link</code>	link function for the binomial family of error distribution. Default is <code>probit</code> .
<code>ps.data</code>	a data frame including the variables used in the model.
<code>br</code>	logical. If TRUE, <code>brglm</code> for bias reduction is used if values are equal to 0 or 1. Default is FALSE.

### Details

Estimates are computed only for GLM of the type  $F(Y) \sim X$ , where  $X$  is a continuous predictor. Std. Errors and 95% confidence intervals of PSE and JND are estimated via Delta Methods. Currently only working with *probit* link function.

### Value

`PsychFunction` returns a list including the fitted model, the estimate of PSE and JND and a flag to indicate if `brglm` was called.

### Note

`PsychFunction` returns the same parameter estimate as `PsychDelta`, without an explicit call to `glm`. Moreover, it allows to fit the model using `brglm` in case of complete or quasi separation.

### References

- Faraggi, D., Izikson, P., & Reiser, B. (2003). Confidence intervals for the 50 per cent response dose. *Statistics in medicine*, 22(12), 1977-1988. <https://doi.org/10.1002/sim.1368>
- Moscatelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision*, 12(11):26, 1-17. doi:10.1167/12.11.26



**See Also**

[glm](#) for Generalized Linear Models. [brglm](#) for fitting a GLM using bias reduction. [PsychPlot](#) for plotting a psychometric function given a [glm](#) (or [brglm](#)) object. [PsychPlot](#) for plotting a psychometric function from a GLM. [PsychShape](#) for plotting a psychometric function given PSE and JND.

**Examples**

```
data.S1 <- subset(simul_data, Subject == "S1")
psych.S1 <- PsychFunction(ps.formula = cbind(Longer, Total - Longer) ~ X,
ps.link = "probit", ps.data = data.S1)
```

---

 PsychPlot

---

*Plot Psychometric Function from GLM*


---

**Description**

Plot a psychometric function given an object of class [glm](#) or [brglm](#). The plot can be drawn on a new or existing [ggplot](#) object.

**Usage**

```
PsychPlot(
  model.obj,
  addTo = NULL,
  showData = TRUE,
  ps.type = "solid",
  ps.size = 1,
  ps.lab = ""
)
```

**Arguments**

<code>model.obj</code>	the fitted psychometric function. An object of class <a href="#">glm</a> or <a href="#">brglm</a> .
<code>addTo</code>	specifies an existing <a href="#">ggplot</a> object where the new line should be plotted. If no object is given, the function is drawn on a new plot.
<code>showData</code>	logical, defines if proportion of binomial responses for each stimulus level are presented. Default is TRUE.
<code>ps.type, ps.size</code>	type and size of the plotted line (see " <a href="#">ggplot2-spec</a> ").
<code>ps.lab</code>	label assigned to the psychometric curve. The label is coded by the color aesthetic.

**Value**

`PsychPlot` returns a [ggplot](#) object.

## References

- Moscatelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision*, 12(11):26, 1-17. doi:10.1167/12.11.26
- Knoblauch, K., & Maloney, L. T. (2012). *Modeling psychophysical data in R* (Vol. 32). Springer Science & Business Media.

## See Also

[glm](#) for for Generalized Linear Models. [PsychFunction](#) for estimation of PSE and JND. [MixPlot](#) for plotting individual responses from a GLMM.

## Examples

```
data.S1 <- subset(simul_data, Subject == "S1")
psych.S1 <- PsychFunction(ps.formula = cbind(Longer, Total - Longer) ~ X,
  ps.link = "probit", ps.data = data.S1)
plotP1 <- PsychPlot(psych.S1$model, showData = TRUE, ps.lab = "S1")

data.S2 <- subset(simul_data, Subject == "S2")
glm.S2 <- glm(formula = cbind(Longer, Total - Longer) ~ X,
  family = binomial(link = "probit"), data = data.S2)
plotP2 <- PsychPlot(glm.S2, addTo = plotP1, ps.lab = "S2")
```

---

PsychShape

*Plot Psychometric Functions given PSE and JND*

---

## Description

Plot a psychometric function with known PSE and JND on a new or existing ggplot object.

## Usage

```
PsychShape(
  pse = 0,
  jnd = 1,
  p = 0.75,
  x.range = c(NA, NA),
  ps.link = c("probit"),
  ps.type = "solid",
  ps.size = 1,
  ps.color = "black",
  addTo = NULL
)
```

**Arguments**

pse, jnd	point of subjective equivalence (PSE) and just noticeable difference (JND) of the desired psychometric function.
p	probability value relative to the JND upper limit. Default is 0.75 (value for 50% JND).
x.range	vector of length two specifying the range of the psychometric function.
ps.link	a link function for the binomial family of error distribution.
ps.type, ps.size, ps.color	type, size, and color of the plotted line (see "ggplot2-spec").
addTo	specifies an existing ggplot object where the new line should be plotted. If no object is given, the function is drawn on a new plot.

**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.

**Value**

PsychShape returns a [ggplot](#) object.

**References**

Moscattelli, A., Mezzetti, M., & Lacquaniti, F. (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision*, 12(11):26, 1-17. doi:10.1167/12.11.26

Knoblauch, K., & Maloney, L. T. (2012). *Modeling psychophysical data in R* (Vol. 32). Springer Science & Business Media.

**See Also**

[glm](#) for for Generalized Linear Models. [PsychFunction](#) and [PsychDelta](#) for estimation of PSE and JND from response data. [PsychPlot](#) for plotting a a psychometric function from a GLM.

**Examples**

```
p <- PsychShape(pse = 0, jnd = 6, x.range = c(-40, 40), ps.color = "gray", ps.size = 3)
p1 <- PsychShape(pse = 6, jnd = 6, x.range = c(-40, 40), ps.col = "black", addTo = p)
p2 <- PsychShape(pse = 6, jnd = 6, x.range = c(-40, 40), ps.col = "red", ps.link = "logit",
ps.type = "dashed", addTo = NULL)
```

---

PsySimulate

*Simulate psychophysical data*

---

### Description

The function simulates data of a typical psychophysics experiment. For each simulated participant, the function returns the following information: individual slope and intercept coefficients, given the fixed and random effects parameters provided as input; summary of the simulated binomial response to a range of intensity levels between a specified range.

### Usage

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

### Arguments

<code>fixeff</code>	array of fixed effects. First item is the intercept, second item is the slope.
<code>raneff</code>	array of random effects. First item is the intercept, second item is the covariance, third item is the slope.
<code>nsubjects</code>	number of subjects to simulate. Default is 8.
<code>ntrials</code>	number of trials for each stimulus level. Default is 40.
<code>nintervals</code>	number of stimulus levels. Default is 9.
<code>xint</code>	range of the stimulus interval. Default is c(40,120)
<code>constant</code>	logical. If set to FALSE, stimulus levels are randomly generated, uniformly distributed values within the selected interval. otherwise, the X interval is divided in intervals of constant length. Default is TRUE.

### Value

PsySimulate returns a simulated dataset. If no input arguments are specified, the function returns a dataset with the same characteristics as [simul\\_data](#).

### See Also

[PsychShape](#) for plotting a psychometric function given PSE and JND.

### Examples

```
datafr.S1 <- PsySimulate(fixeff = c(0, 1), xint = c(-5,5),
  nsubject = 1, ntrials = 60, nintervals = 10, constant = FALSE)
library(ggplot2)
g <- ggplot(datafr.S1, aes(X,Longer/Total)) + geom_point()

PsychShape(pse = 0, jnd = qnorm(0.75)/1, ps.link = "probit",
  x.range = c(-5,5), addTo = g, ps.color = "red")
```

---

simul\_data

*A simulated psychophysical dataset*

---

### Description

A dataset containing simulated responses to a forced-choice task for eight clusters. Created using `PsySimulate()`. The variables are as follows:

### Usage

```
data(simul_data)
```

### Format

A data frame with 72 rows (9 observations x 8 simulated participants) and 6 variables:

**X** samples in the continuous interval (range `c(40, 120)`)

**Intercept,Slope** simulated participant's coefficients (combination of random and fixed effect)

**Longer** number of trials in which response is judged "longer" than standard

**Total** total number of trials for sample in X

**Subject** simulated participant's identification code (S1 to S8)

### See Also

[PsySimulate](#) for simulating dataframes with custom parameters.

---

vibro\_exp3

*Data from tactile discrimination task - (Dallmann et al., 2015).*


---

### Description

A dataset containing the response recorded 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

```
data(vibro_exp3)
```

### Format

A data frame with 126 rows (14 observations x 9 participants) and 5 variables:

**speed** numeric, speed of the moving surface (in cm/s, range c(1, 16))

**vibration** factor representing the vibration frequency of the masking stimulus. Two levels: 32 (vibration in the experimental condition, in Hz) or 0 (no vibration - control condition)

**faster** proportion of trials in which the comparison stimulus was judged as faster than the reference

**slower** proportion of trials in which the comparison stimulus was judged as slower than the reference

**subject** participant's identification code

### Source

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

### References

Dallmann, C. J., Ernst, M. O., & Moscatelli, A. (2015). The role of vibration in tactile speed perception. *Journal of Neurophysiology*, 114(6), 3131–3139. <doi:10.1152/jn.00621.2015>

---

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

```
xplode(model.obj, name.cont = NA, name.factor = NA, names.response = NA)
```

**Arguments**

<code>model.obj</code>	The GLMM fitted with <code>glmer</code> . An object of class " <code>merMod</code> ".
<code>name.cont</code>	A string providing the name of the continuous predictor, as in the formula object of the fitted model
<code>name.factor</code>	A string providing the name of name of the categorical predictor, as in the formula object of the fitted model
<code>names.response</code>	Optional. A string providing the name of name of the response variable, as in the formula object of the fitted model

**Details**

For simplicity and maintenance reasons, several `MixedPsy` functions take as input an object of class `xplode` instead of an object of class `merMod`.

**See Also**

[merMod-class](#) and [glmer](#). [MixDelta](#), [MixPlot](#) for use of objects of class `xplode`.

**Examples**

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

# Index

- \* **Bootstrap**
    - pseMer, 4
  - \* **DeltaMethod**
    - MixDelta, 2
    - PsychDelta, 6
    - PsychFunction, 8
  - \* **GLMM**
    - MixDelta, 2
    - MixPlot, 3
    - pseMer, 4
    - xplode, 14
  - \* **GLM**
    - PsychDelta, 6
    - PsychFunction, 8
    - PsychPlot, 9
    - PsychShape, 10
  - \* **datasets**
    - simul\_data, 13
    - vibro\_exp3, 14
  - \* **plot**
    - MixPlot, 3
    - PsychPlot, 9
    - PsychShape, 10
- boot.ci, 5, 6  
bootMer, 5, 6  
brglm, 8, 9
- formula, 8
- ggplot, 4, 9, 11  
ggplot2, 4  
glm, 6–11  
glmer, 2, 3, 7, 15
- merMod, 2, 4, 5, 14, 15  
MixDelta, 2, 6, 15  
MixPlot, 3, 10, 15
- pseMer, 3, 4  
PsychDelta, 6, 8, 11  
PsychFunction, 8, 10, 11  
PsychPlot, 4, 9, 9, 11  
PsychShape, 9, 10, 12  
PsySimulate, 12, 13
- simul\_data, 12, 13
- vibro\_exp3, 14
- xplode, 3, 4, 14