

# Package ‘psycho’

May 11, 2018

**Type** Package

**Title** Efficient and Publishing-Oriented Workflow for Psychological Science

**Version** 0.2.3

**Maintainer** Dominique Makowski <dom.makowski@gmail.com>

**URL** <https://github.com/neuropsychology/psycho.R>

**BugReports** <https://github.com/neuropsychology/psycho.R/issues>

**Description** Toolbox for psychologists, neuropsychologists and neuroscientists.

It implements methods to create report-ready outputs for many statistical models, as well as various convenient statistical functions used in psychological science, such as correlation matrices and standardization.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Depends** R (>= 3.4.0)

**Imports** methods, dplyr, ggplot2, tidyr, stringr, purrr, psych, MASS, nFactors, qgraph, ppcor, ggcorrplot, rstanarm, MuMIn, lmerTest, coda, emmeans, broom, tibble, DescTools, pbkrtest, BayesFactor, scales

**Suggests** knitr, rmarkdown, testthat, covr, lme4, plotly, loo (>= 2.0.0)

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Dominique Makowski [aut, cre, cph],  
Viliam Simko [ctb],  
Sasha Epskamp [rev] (Sasha reviewed the package for JOSS, see <https://github.com/openjournals/joss-reviews/issues/470>)

**Repository** CRAN

**Date/Publication** 2018-05-11 16:07:24 UTC

**R topics documented:**

affective	3
analyze	3
analyze.glmerMod	4
analyze.lm	5
analyze.lmerModLmerTest	5
analyze.stanreg	6
as.data.frame.density	7
assess	8
correlation	9
crawford.test	10
crawford.test.freq	12
crawford_dissociation.test	12
dprime	14
emotion	15
find_best_model	16
find_best_model.merModLmerTest	17
find_best_model.stanreg	18
find_combinations	19
find_combinations.formula	19
find_season	20
format_digit	21
format_p	21
format_string	22
get_contrasts	22
get_contrasts.glmerMod	23
get_contrasts.lmerModLmerTest	24
get_contrasts.stanreg	25
get_predicted	26
get_predicted.stanreg	26
hdi	28
interpret_bf	29
interpret_d	30
interpret_d_posterior	30
is.mixed	31
is.mixed.stanreg	32
is.psychobject	32
is.standardized	33
mellenbergh.test	33
mpe	34
n_factors	35
odds_to_probs	36
overlap	36
percentile	37
percentile_to_z	38
plot.psychobject	38
power_analysis	39

<i>affective</i>	3
print.psychobject . . . . .	40
probs_to_odds . . . . .	40
rnorm_perfect . . . . .	41
standardize . . . . .	42
summary.psychobject . . . . .	43
values . . . . .	43
<b>Index</b>	<b>44</b>

---

<i>affective</i>	<i>Data from the Affective Style Questionnaire (ASQ - French Validation)</i>
------------------	--

---

### Description

This is data from the French validation of the Affective Style Questionnaire.

### Usage

*affective*

### Format

A data frame with 1277 rows and 8 variables:

- Sex** Sex (F or M)
- Birth\_Season** Season of birth
- Age** Current age
- Salary** Salary in euros
- Life\_Satisfaction** General life satisfaction
- Concealing** Concealing score
- Adjusting** Adjusting score
- Tolerating** Tolerating score

---

<i>analyze</i>	<i>Analyze objects.</i>
----------------	-------------------------

---

### Description

Analyze objects. See the documentation for your object's class:

- [analyze.stanreg](#)
- [analyze.merModLmerTest](#)
- [analyze.glmerMod](#)
- [analyze.lm](#)

**Usage**

```
analyze(x, ...)
```

**Arguments**

x                    object to analyze.  
...                  Arguments passed to or from other methods.

**Author(s)**

[Dominique Makowski](#)

---

analyze.glmerMod      *Analyze glmerMod objects.*

---

**Description**

Analyze glmerMod objects.

**Usage**

```
## S3 method for class 'glmerMod'  
analyze(x, ...)
```

**Arguments**

x                    merModLmerTest object.  
...                  Arguments passed to or from other methods.

**Value**

output

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
library(psycho)  
library(lme4)  
fit <- lme4::glmer(vs ~ mpg + (1|cyl), data=mtcars, family="binomial")  
  
results <- analyze(fit)  
summary(results)  
print(results)
```

---

analyze.lm	Analyze lm objects.
------------	---------------------

---

**Description**

Analyze lm objects.

**Usage**

```
## S3 method for class 'lm'  
analyze(x, CI = 95, ...)
```

**Arguments**

x	lm object.
CI	Confidence interval bounds. Set to NULL turn off their computation.
...	Arguments passed to or from other methods.

**Value**

output

**Author(s)**

**Dominique Makowski**

**Examples**

```
library(psycho)  
fit <- lm(Sepal.Length ~ Sepal.Width, data=iris)  
  
results <- analyze(fit)  
summary(results)  
print(results)
```

---

analyze.lmerModLmerTest	Analyze lmerModLmerTest objects.
-------------------------	----------------------------------

---

**Description**

Analyze lmerModLmerTest objects.

**Usage**

```
## S3 method for class 'lmerModLmerTest'  
analyze(x, CI = 95, ...)
```

**Arguments**

x	lmerModLmerTest object.
CI	Bootsrapped confidence interval bounds (slow). Set to NULL turn off their computation.
...	Arguments passed to or from other methods.

**Value**

output

**Author(s)**

**Dominique Makowski**

**Examples**

```
library(psycho)  
library(lmerTest)  
fit <- lmerTest::lmer(Sepal.Length ~ Sepal.Width + (1|Species), data=iris)  
  
results <- analyze(fit)  
summary(results)  
print(results)
```

---

analyze.stanreg      *Analyze stanreg objects.*

---

**Description**

Analyze stanreg objects.

**Usage**

```
## S3 method for class 'stanreg'  
analyze(x, CI = 90, effsize = FALSE, overlap = TRUE,  
...)
```

**Arguments**

x	A stanreg model.
CI	Credible interval bounds.
effsize	Compute Effect Sizes according to Cohen (1988)? Your outcome variable must be standardized.
overlap	Compute the overlapping coefficient between the posterior and a normal distribution of mean 0 and same SD.
...	Arguments passed to or from other methods.

**Value**

output

**Author(s)**

Dominique Makowski

**Examples**

```
## Not run:
library(psycho)
library(rstanarm)

data <- standardize(attitude)
fit <- rstanarm::stan_glm(rating ~ advance + privileges, data=data)

results <- analyze(fit, effsize=TRUE)
summary(results)
plot(results)
print(results)

fit <- rstanarm::stan_glmer(Sepal.Length ~ Sepal.Width + (1|Species), data=iris)
results <- analyze(fit)
summary(results)

## End(Not run)
```

---

as.data.frame.density *Coerce to a Data Frame.*

---

**Description**

Functions to check if an object is a data frame, or coerce it if possible.

**Usage**

```
## S3 method for class 'density'
as.data.frame(x, ...)
```

**Arguments**

x                    any R object.  
...                    additional arguments to be passed to or from methods.

**Author(s)**

**Dominique Makowski**

---

assess

*Compare a patient's score to a control group*

---

**Description**

Compare a patient's score to a control group.

**Usage**

```
assess(patient, mean = 0, sd = 1, n = NULL, controls = NULL, CI = 95,
        treshold = 0.05, iter = 10000, color_controls = "#2196F3",
        color_CI = "#E91E63", color_score = "black", color_size = 2,
        alpha_controls = 1, alpha_CI = 0.8, verbose = TRUE)
```

**Arguments**

patient	Single value (patient's score).
mean	Mean of the control sample.
sd	SD of the control sample.
n	Size of the control sample.
controls	Vector of values (control's scores).
CI	Credible interval bounds.
treshold	Significance treshold.
iter	Number of iterations.
color_controls	Color of the controls distribution.
color_CI	Color of CI distribution.
color_score	Color of the line representing the patient's score.
color_size	Size of the line representing the patient's score.
alpha_controls	Alpha of the CI distribution.
alpha_CI	lpha of the controls distribution.
verbose	Print possible warnings.



**Details**

Until relatively recently the standard way of testing for a difference between a case and controls was to convert the case's score to a z score using the control sample mean and standard deviation (SD). If z was less than -1.645 (i.e., below 95

Crawford et al. (2012) remind that empirically based decision algorithms are almost always vastly superior to clinically based decision making while being more reliable, accurate, and cost-effective (Dawes, Faust, & Meehl, 1989; Grove & Lloyd, 2006; Salzinger, 2005).

**Value**

output

**Author(s)**

**Dominique Makowski**

**Examples**

```
result <- assess(patient=124, mean=100, sd=15, n=100)
print(result)
plot(result)
```

---

correlation                      *Multiple Correlations.*

---

**Description**

Compute different kinds of correlation matrices.

**Usage**

```
correlation(df, df2 = NULL, type = "full", method = "pearson",
  adjust = "holm", i_am_cheating = FALSE)
```

**Arguments**

df	The dataframe.
df2	Optional dataframe to correlate with the first one.
type	A character string indicating which correlation type is to be computed. One of "full" (default), "partial" (partial correlations), "semi" (semi-partial correlations), "glasso" (Graphical lasso- estimation of Gaussian graphical models) or "cor_auto" (will use the qgraph::cor_auto function to return psychoric or polyserial correlations if needed).
method	A character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman" can be abbreviated.

- adjust** What adjustment for multiple tests should be used? ("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). See [p.adjust](#) for details about why to use "holm" rather than "bonferroni").
- i\_am\_cheating** Set to TRUE to run many uncorrected correlations.

**Value**

output

**Author(s)**

**Dominique Makowski**

**Examples**

```
df <- attitude

# Normal correlations
results <- psycho::correlation(df)
print(results)
plot(results)

# Partial correlations with correction
results <- psycho::correlation(df, type="partial",
                              method="spearman",
                              adjust="holm")

print(results)
plot(results)
```

---

crawford.test

*Crawford-Garthwaite (2007) Bayesian test for single-case analysis.*

---

**Description**

Neuropsychologists often need to compare a single case to a small control group. However, the standard two-sample t-test does not work because the case is only one observation. Crawford and Garthwaite (2007) demonstrate that the Bayesian test is a better approach than other commonly-used alternatives. .

**Usage**

```
crawford.test(patient, controls = NULL, mean = NULL, sd = NULL,
              n = NULL, CI = 95, treshold = 0.1, iter = 10000,
              color_controls = "#2196F3", color_CI = "#E91E63", color_score = "black",
              color_size = 2, alpha_controls = 1, alpha_CI = 0.8)
```

**Arguments**

patient	Single value (patient's score).
controls	Vector of values (control's scores).
mean	Mean of the control sample.
sd	SD of the control sample.
n	Size of the control sample.
CI	Credible interval bounds.
treshold	Significance treshold.
iter	Number of iterations.
color_controls	Color of the controls distribution.
color_CI	Color of CI distribution.
color_score	Color of the line representing the patient's score.
color_size	Size of the line representing the patient's score.
alpha_controls	Alpha of the CI distribution.
alpha_CI	lpha of the controls distribution.

**Details**

The p value obtained when this test is used to test significance also simultaneously provides a point estimate of the abnormality of the patient's score; for example if the one-tailed probability is .013 then we know that the patient's score is significantly ( $p < .05$ ) below the control mean and that it is estimated that 1.3

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)

crawford.test(patient = 125, mean=100, sd=15, n=100)
plot(crawford.test(patient = 80, mean=100, sd=15, n=100))

crawford.test(patient = 10, controls = c(0, -2, 5, 2, 1, 3, -4, -2))
test <- crawford.test(patient = 7, controls = c(0, -2, 5, -6, 0, 3, -4, -2))
plot(test)
```

crawford.test.freq      *Crawford-Howell (1998) frequentist t-test for single-case analysis.*

---

### Description

Neuropsychologists often need to compare a single case to a small control group. However, the standard two-sample t-test does not work because the case is only one observation. Crawford and Garthwaite (2012) demonstrate that the Crawford-Howell (1998) t-test is a better approach (in terms of controlling Type I error rate) than other commonly-used alternatives. .

### Usage

```
crawford.test.freq(patient, controls)
```

### Arguments

patient	Single value (patient's score).
controls	Vector of values (control's scores).

### Value

Returns a data frame containing the t-value, degrees of freedom, and p-value. If significant, the patient is different from the control group.

### Author(s)

Dan Mirman, Dominique Makowski

### Examples

```
library(psycho)

crawford.test.freq(patient = 10, controls = c(0, -2, 5, 2, 1, 3, -4, -2))
crawford.test.freq(patient = 7, controls = c(0, -2, 5, 2, 1, 3, -4, -2))
```

---

crawford\_dissociation.test

*Crawford-Howell (1998) modified t-test for testing difference between a patient's performance on two tasks.*

---

**Description**

Assessing dissociation between processes is a fundamental part of clinical neuropsychology. However, while the detection of suspected impairments is a fundamental feature of single-case studies, evidence of an impairment on a given task usually becomes of theoretical interest only if it is observed in the context of less impaired or normal performance on other tasks. Crawford and Garthwaite (2012) demonstrate that the Crawford-Howell (1998) t-test for dissociation is a better approach (in terms of controlling Type I error rate) than other commonly-used alternatives. .

**Usage**

```
crawford_dissociation.test(case_X, case_Y, controls_X, controls_Y,  
  verbose = T)
```

**Arguments**

case_X	Single value (patient's score on test X).
case_Y	Single value (patient's score on test Y).
controls_X	Vector of values (control's scores of X).
controls_Y	Vector of values (control's scores of Y).
verbose	True or False. Prints the interpretation text.

**Value**

Returns a data frame containing the t-value, degrees of freedom, and p-value. If significant, the dissociation between test X and test Y is significant.

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)  
  
case_X <- 142  
case_Y <- 7  
controls_X <- c(100, 125, 89, 105, 109, 99)  
controls_Y <- c(7, 8, 9, 6, 7, 10)  
  
crawford_dissociation.test(case_X, case_Y, controls_X, controls_Y)
```

---

dprime

*Dprime and Other Signal Detection Theory indices.*


---

### Description

Computes Signal Detection Theory indices ( $d'$ ,  $\beta$ ,  $A'$ ,  $B''D$ ,  $c$ ).

### Usage

```
dprime(n_hit, n_fa, n_miss = NULL, n_cr = NULL, n_targets = NULL,
       n_distractors = NULL, adjusted = TRUE)
```

### Arguments

n_hit	Number of hits.
n_fa	Number of false alarms.
n_miss	Number of misses.
n_cr	Number of correct rejections.
n_targets	Number of targets (n_hit + n_miss).
n_distractors	Number of distractors (n_fa + n_cr).
adjusted	Should it use the Hautus (1995) adjustments for extreme values.

### Value

Calculates the  $d'$ , the  $\beta$ , the  $A'$  and the  $B''D$  based on the signal detection theory (SRT). See Pallier (2002) for the algorithms.

Returns a list containing 4 objects:

- **dprime ( $d'$ )**: The sensitivity. Reflects the distance between the two distributions: signal, and signal+noise and corresponds to the  $Z$  value of the hit-rate minus that of the false-alarm rate.
- **beta**: The bias (criterion). The value for beta is the ratio of the normal density functions at the criterion of the  $Z$  values used in the computation of  $d'$ . This reflects an observer's bias to say 'yes' or 'no' with the unbiased observer having a value around 1.0. As the bias to say 'yes' increases (liberal), resulting in a higher hit-rate and false-alarm-rate, beta approaches 0.0. As the bias to say 'no' increases (conservative), resulting in a lower hit-rate and false-alarm rate, beta increases over 1.0 on an open-ended scale.
- **aprime ( $A'$ )**: Non-parametric estimate of discriminability. An  $A'$  near 1.0 indicates good discriminability, while a value near 0.5 means chance performance.
- **bppd ( $B''D$ )**: Non-parametric estimate of bias. A  $B''D$  equal to 0.0 indicates no bias, positive numbers represent conservative bias (i.e., a tendency to answer 'no'), negative numbers represent liberal bias (i.e. a tendency to answer 'yes'). The maximum absolute value is 1.0.
- **c**: Another index of bias. the number of standard deviations from the midpoint between these two distributions, i.e., a measure on a continuum from "conservative" to "liberal".

Note that for  $d'$  and beta, adjustment for extreme values are made following the recommendations of Hautus (1995).

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)

n_hit <- 9
n_fa <- 2
n_miss <- 1
n_cr <- 7

indices <- psycho::dprime(n_hit, n_fa, n_miss, n_cr)

df <- data.frame(Participant = c("A", "B", "C"),
  n_hit = c(1, 2, 5),
  n_fa = c(6, 8, 1))

indices <- psycho::dprime(n_hit=df$n_hit,
  n_fa=df$n_fa,
  n_targets=10,
  n_distractors=10,
  adjusted=FALSE)
```

---

emotion

*Emotional Ratings of Pictures*

---

**Description**

Emotional ratings of neutral and negative pictures by healthy participants.

**Usage**

```
emotion
```

**Format**

A data frame with 912 rows and 11 variables:

**Participant\_ID** Subject's number

**Participant\_Age** Subject's age

**Participant\_Sex** Subject's sex

**Item\_Category** Picture's category

**Item\_Name** Picture's name

- Trial\_Order** Trial order (1-48)
- Emotion\_Condition** Picture's emotional category (Neutral or Negative)
- Subjective\_Arousal** Participant's rating of arousal (0-100)
- Subjective\_Valence** Participant's rating of valence (-100: negative, 100: positive, 0: neutral)
- Autobiographical\_Link** Participant's rating of autobiographical connection (is the picture's content associated with memories)
- Recall** Whether the participant recalled the picture 20min after presentation

---

find\_best\_model      *Returns the best model.*

---

## Description

Returns the best model. See the documentation for your model's class:

- [find\\_best\\_model.stanreg](#)
- [find\\_best\\_model.merModLmerTest](#)

## Usage

```
find_best_model(fit, ...)
```

## Arguments

fit	Model
...	Arguments passed to or from other methods.

## Author(s)

[Dominique Makowski](#)

## See Also

[find\\_best\\_model.stanreg](#)



---

`find_best_model.merModLmerTest`*Returns the best combination of predictors for lmerTest objects.*

---

**Description**

Returns the best combination of predictors for lmerTest objects.

**Usage**

```
## S3 method for class 'merModLmerTest'  
find_best_model(fit, interaction = TRUE,  
  fixed = NULL, ...)
```

**Arguments**

<code>fit</code>	A merModLmerTest object.
<code>interaction</code>	Include interaction term.
<code>fixed</code>	Additional formula part to add at the beginning of each formula
<code>...</code>	Arguments passed to or from other methods.

**Value**

list containing all combinations.

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
## Not run:  
library(psycho)  
library(lmerTest)  
  
data <- standardize(iris)  
fit <- lmerTest::lmer(Sepal.Length ~ Sepal.Width + Petal.Length + (1|Species), data=data)  
  
best <- find_best_model(fit)  
best_formula <- best$formula  
best$table  
  
## End(Not run)
```

---

```
find_best_model.stanreg
```

*Returns the best combination of predictors based on LOO cross-validation indices.*

---

### Description

Returns the best combination of predictors based on LOO cross-validation indices.

### Usage

```
## S3 method for class 'stanreg'
find_best_model(fit, interaction = TRUE, fixed = NULL,
  K = 10, k_treshold = NULL, ...)
```

### Arguments

<code>fit</code>	A stanreg object.
<code>interaction</code>	Include interaction term.
<code>fixed</code>	Additional formula part to add at the beginning of each formula
<code>K</code>	For kfold, the number of subsets of equal (if possible) size into which the data will be randomly partitioned for performing K-fold cross-validation. The model is refit K times, each time leaving out one of the K subsets. If K is equal to the total number of observations in the data then K-fold cross-validation is equivalent to exact leave-one-out cross-validation.
<code>k_treshold</code>	Threshold for flagging estimates of the Pareto shape parameters k estimated by loo.
<code>...</code>	Arguments passed to or from other methods.

### Value

list containing all combinations.

### Author(s)

**Dominique Makowski**

### Examples

```
## Not run:
library(psycho)
library(rstanarm)

data <- standardize(attitude)
fit <- rstanarm::stan_glm(rating ~ advance + privileges, data=data)

best <- find_best_model(fit)
```

```
best_formula <- best$formula
best$table

# To deactivate Kfold evaluation
best <- find_best_model(fit, K=0)

## End(Not run)
```

---

find\_combinations      *Generate all combinations.*

---

### Description

Generate all combinations.

### Usage

```
find_combinations(object, ...)
```

### Arguments

object	Object
...	Arguments passed to or from other methods.

### Author(s)

[Dominique Makowski](#)

---

find\_combinations.formula  
*Generate all combinations of predictors of a formula.*

---

### Description

Generate all combinations of predictors of a formula.

### Usage

```
## S3 method for class 'formula'
find_combinations(object, interaction = TRUE,
  fixed = NULL, ...)
```

**Arguments**

object	Formula.
interaction	Include interaction term.
fixed	Additional formula part to add at the beginning of each combination.
...	Arguments passed to or from other methods.

**Value**

list containing all combinations.

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
library(psycho)

f <- as.formula("Y ~ A + B + C + D")
f <- as.formula("Y ~ A + B + C + D + (1|E)")
f <- as.formula("Y ~ A + B + C + D + (1|E) + (1|F)")

find_combinations(f)
```

---

find_season	<i>Find season of dates.</i>
-------------	------------------------------

---

**Description**

Returns the season of an array of dates.

**Usage**

```
find_season(date)
```

**Arguments**

date	Array of dates. Must cover the 4 seasons.
------	---

**Value**

season

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
library(psycho)

dates <- c("2017-02-15", "2017-05-15", "2017-08-15", "2017-11-15")
find_season(dates)
```

---

format_digit	<i>Format digits.</i>
--------------	-----------------------

---

**Description**

Format digits.

**Usage**

```
format_digit(x, digits = 2, null_treshold = 0.001)
```

**Arguments**

x	A digit.
digits	Number of significant digits.
null_treshold	Threshold below which return 0.

**Author(s)**

[Dominique Makowski](#)

---

format_p	<i>Format p values.</i>
----------	-------------------------

---

**Description**

Format p values.

**Usage**

```
format_p(pvalues)
```

**Arguments**

pvalues	P values (scalar or vector).
---------	------------------------------

**Author(s)**

[Dominique Makowski](#)

---

format_string	<i>Tidyverse-friendly sprintf.</i>
---------------	------------------------------------

---

**Description**

Tidyverse-friendly sprintf.

**Usage**

```
format_string(x, fmt, ...)
```

**Arguments**

x	Values.
fmt	A character vector of format strings, each of up to 8192 bytes.
...	values to be passed into fmt. Only logical, integer, real and character vectors are supported, but some coercion will be done: see the ‘Details’ section. Up to 100.

---

get_contrasts	<i>Get Marginal Means and Contrasts.</i>
---------------	--

---

**Description**

Compute estimated marginal means and contrasts from models. See the documentation for your model’s class:

- [get\\_contrasts.stanreg](#)
- [get\\_contrasts.merModLmerTest](#)
- [get\\_contrasts.glmerMod](#)

**Usage**

```
get_contrasts(fit, ...)
```

**Arguments**

fit	Model.
...	Arguments passed to or from other methods.

**Author(s)**

[Dominique Makowski](#)

---

`get_contrasts.glmerMod`*Compute estimated marginal means and contrasts from glmerMod models.*

---

## Description

Compute estimated marginal means and contrasts from a glmerMod models.

## Usage

```
## S3 method for class 'glmerMod'  
get_contrasts(fit, formula, adjust = "tukey", ...)
```

## Arguments

<code>fit</code>	A glmerMod model.
<code>formula</code>	A character vector (formula like format, i.e., including interactions or nesting terms) specifying the names of the predictors over which EMMs are desired.
<code>adjust</code>	P value adjustment method. Default is "tukey". Can be "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
<code>...</code>	Arguments passed to or from other methods.

## Value

list with estimated marginal means (95

## Author(s)

[Dominique Makowski](#)

## Examples

```
## Not run:  
library(psycho)  
require(lme4)  
fit <- lme4::glmer(Sex ~ Birth_Season + (1|Salary), data=affektive, family="binomial")  
  
contrasts <- get_contrasts(fit, formula="Birth_Season", adjust="tukey")  
contrasts$means  
contrasts$contrasts  
  
## End(Not run)
```

---

```
get_contrasts.lmerModLmerTest
```

*Compute estimated marginal means and contrasts from lmer-ModLmerTest models.*

---

## Description

Compute estimated marginal means and contrasts from a lmerModLmerTest models.

## Usage

```
## S3 method for class 'lmerModLmerTest'
get_contrasts(fit, formula, adjust = "tukey", ...)
```

## Arguments

fit	A merModLmerTest model.
formula	A character vector (formula like format, i.e., including interactions or nesting terms) specifying the names of the predictors over which EMMs are desired.
adjust	P value adjustment method. Default is "tukey". Can be "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" or "none".
...	Arguments passed to or from other methods.

## Value

list with estimated marginal means (95

## Author(s)

[Dominique Makowski](#)

## Examples

```
## Not run:
library(psycho)
require(lmerTest)
fit <- lmerTest::lmer(Adjusting ~ Birth_Season + (1|Salary), data=affective)

contrasts <- get_contrasts(fit, formula="Birth_Season", adjust="tukey")
contrasts$means
contrasts$contrasts

## End(Not run)
```



---

get\_contrasts.stanreg *Compute estimated marginal means and contrasts from stanreg models.*

---

## Description

Compute estimated marginal means and contrasts from a stanreg models.

## Usage

```
## S3 method for class 'stanreg'  
get_contrasts(fit, formula, prob = 0.9, ...)
```

## Arguments

fit	A stanreg model.
formula	A character vector (formula like format, i.e., including interactions or nesting terms) specifying the names of the predictors over which EMMs are desired.
prob	A numeric scalar in the interval (0,1) giving the target probability content of the intervals. The nominal probability content of the intervals is the multiple of 1/nrow(obj) nearest to prob.
...	Arguments passed to or from other methods.

## Value

list with estimated marginal means and contrasts.

## Author(s)

[Dominique Makowski](#)

## Examples

```
## Not run:  
library(psycho)  
require(rstanarm)  
fit <- rstanarm::stan_glm(Adjusting ~ Birth_Season * Sex, data=affektive)  
  
contrasts <- get_contrasts(fit, formula="Birth_Season * Sex")  
contrasts$means  
  
contrasts <- get_contrasts(fit, formula="Birth_Season")  
contrasts$contrasts  
  
## End(Not run)
```

---

get_predicted	<i>Compute predicted values from models.</i>
---------------	--

---

**Description**

Compute predicted values from models. See the documentation for your model's class:

- [get\\_predicted.stanreg](#)

**Usage**

```
get_predicted(fit, ...)
```

**Arguments**

fit	Model.
...	Arguments passed to or from other methods.

**Author(s)**

[Dominique Makowski](#)

---

get_predicted.stanreg	<i>Compute predicted values of stanreg models.</i>
-----------------------	--

---

**Description**

Compute predicted from a stanreg model.

**Usage**

```
## S3 method for class 'stanreg'
get_predicted(fit, newdata = "model", prob = 0.9,
  keep_iterations = FALSE, draws = NULL, odds_to_probs = TRUE,
  posterior_predict = FALSE, seed = NULL, ...)
```

**Arguments**

fit	A stanreg model.
newdata	A data frame in which to look for variables with which to predict. If omitted, the model matrix is used. If "model", the model's data is used.
prob	Probability of credible intervals (0.9 (default) will compute 5-95% CI). Can also be a list of probs (e.g., c(0.90, 0.95)).
keep_iterations	Keep all prediction iterations.

draws	An integer indicating the number of draws to return. The default and maximum number of draws is the size of the posterior sample.
odds_to_probs	Transform log odds ratios in logistic models to probabilities.
posterior_predict	Posterior draws of the outcome instead of the link function (i.e., the regression "line").
seed	An optional seed to use.
...	Arguments passed to or from other methods.

**Value**

dataframe with predicted values.

**Author(s)**

**Dominique Makowski**

**Examples**

```
## Not run:
library(psycho)
library(ggplot2)
require(rstanarm)

fit <- rstanarm::stan_glm(Tolerating ~ Adjusting, data=affektive, iter=500)

refgrid <- emmeans::ref_grid(fit, at=list(
  Adjusting=seq(min(affektive$Adjusting), max(affektive$Adjusting), length.out=10)))

predicted <- get_predicted(fit, newdata=refgrid)

ggplot(predicted, aes(x=Adjusting, y=Tolerating_Median)) +
  geom_line() +
  geom_ribbon(aes(ymin=Tolerating_CI_5,
                ymax=Tolerating_CI_95),
            alpha=0.1)

fit <- rstanarm::stan_glm(Sex ~ Adjusting, data=affektive, family="binomial", iter=500)

refgrid <- emmeans::ref_grid(fit, at=list(
  Adjusting=seq(min(affektive$Adjusting), max(affektive$Adjusting), length.out=10)))

predicted <- get_predicted(fit, newdata=refgrid)

ggplot(predicted, aes(x=Adjusting, y=Sex_Median)) +
  geom_line() +
  geom_ribbon(aes(ymin=Sex_CI_5,
                ymax=Sex_CI_95),
            alpha=0.1)
```

```
## End(Not run)
```

---

**hdi** *Highest Density Intervals (HDI).*

---

### Description

Compute the Highest Density Intervals (HDI) of a distribution.

### Usage

```
hdi(x, prob = 0.95)
```

### Arguments

x	A vector of values from a probability distribution (e.g., posterior probabilities from MCMC sampling).
prob	Scalar between 0 and 1, indicating the mass within the credible interval that is to be estimated.

### Author(s)

**Dominique Makowski**

### Examples

```
library(psycho)

distribution <- rnorm(1000, 0, 1)
hdi_values <- hdi(distribution)
print(hdi_values)
plot(hdi_values)
summary(hdi_values)

x <- matrix(rexp(200), 100)
hdi_values <- hdi(x)
```

---

interpret_bf	<i>Return the interpretation of a Bayes Factor.</i>
--------------	---

---

### Description

Return the interpretation of a Bayes Factor.

### Usage

```
interpret_bf(x, label_only = FALSE)
```

### Arguments

x	Bayes Factor.
label_only	Keep only the size classification.

### Value

The interpretation according to Jeffreys (1961).

### Author(s)

[Dominique Makowski](#)

### References

- Jeffreys, H. (1961), Theory of Probability, 3rd ed., Oxford University Press, Oxford.
- Jarosz, A. F., & Wiley, J. (2014). What are the odds? A practical guide to computing and reporting Bayes factors. *The Journal of Problem Solving*, 7(1), 2.

### Examples

```
library(psycho)
interpret_bf(10)
```

---

interpret_d	<i>Return the interpretation of a Cohen's d or a standardized coefficient following Cohen (1988).</i>
-------------	---

---

**Description**

Return the interpretation of a Cohen's d or a standardized coefficient following Cohen (1988).

**Usage**

```
interpret_d(x)
```

**Arguments**

x                      Cohen's d value of standardized coefficient.

**Value**

The interpretation according to Cohen (1988).

**Author(s)**

**Dominique Makowski**

**Examples**

```
library(psycho)
interpret_d(-0.42)
```

---

interpret_d_posterior	<i>Return the effect sizes for the posterior of a standardized coefficient following Cohen (1988).</i>
-----------------------	--

---

**Description**

Return the effect sizes for the posterior of a standardized coefficient following Cohen (1988).

**Usage**

```
interpret_d_posterior(posterior)
```

**Arguments**

posterior              Posterior distribution of standardized coefficient.

**Value**

The interpretation according to Cohen (1988)

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)
interpret_d_posterior(rnorm(1000, 0, 1))
```

---

<i>is.mixed</i>	<i>Check if model includes random effects.</i>
-----------------	--

---

**Description**

Check if model is mixed. See the documentation for your model's class:

- [is.mixed.stanreg](#)

**Usage**

```
is.mixed(fit, ...)
```

**Arguments**

<code>fit</code>	Model.
<code>...</code>	Arguments passed to or from other methods.

**Author(s)**

Dominique Makowski

`is.mixed.stanreg`      *Check if model includes random effects.*

---

### Description

Check if model is mixed.

### Usage

```
## S3 method for class 'stanreg'  
is.mixed(fit, ...)
```

### Arguments

`fit`                  Model.  
`...`                 Arguments passed to or from other methods.

### Author(s)

[Dominique Makowski](#)

---

`is.psychobject`      *Creates or tests for objects of mode "psychobject".*

---

### Description

Creates or tests for objects of mode "psychobject".

### Usage

```
is.psychobject(x)
```

### Arguments

`x`                    an arbitrary R object.



---

is.standardized	<i>Check if a dataframe is standardized.</i>
-----------------	--

---

**Description**

Check if a dataframe is standardized.

**Usage**

```
is.standardized(df, tol = 0.1)
```

**Arguments**

df	A dataframe.
tol	The error treshold.

**Value**

bool.

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
library(psycho)

df <- psycho::affective
is.standardized(df)

dfZ <- psycho::standardize(df)
is.standardized(dfZ)
```

---

mellenbergh.test	<i>Mellenbergh &amp; van den Brink (1998) test for pre-post comparison.</i>
------------------	---

---

**Description**

Test for comparing post-test to baseline for a single participant.

**Usage**

```
mellenbergh.test(t0, t1, controls)
```

**Arguments**

t0	Single value (pretest or baseline score).
t1	Single value (posttest score).
controls	Vector of scores of the control group OR single value corresponding to the control SD of the score.

**Value**

Returns a data frame containing the z-value and p-value. If significant, the difference between pre and post tests is significant.

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)

mellenbergh.test(t0 = 4, t1 = 12, controls = c(0, -2, 5, 2, 1, 3, -4, -2))
mellenbergh.test(t0 = 8, t1 = 2, controls = 2.6)
```

---

mpe

*Compute Maximum Probability of Effect (MPE).*

---

**Description**

Compute the Maximum Probability of Effect (MPE), i.e., the proportion of posterior distribution that is of the same sign as the median. In other words, it corresponds to the maximum probability that the effect is different from 0 in the median's direction.

**Usage**

```
mpe(posterior)
```

**Arguments**

posterior	Posterior Distribution.
-----------	-------------------------

**Value**

list containing the MPE and its values.

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)
library(rstanarm)

fit <- rstanarm::stan_glm(rating ~ advance, data=attitude)
posterior <- psycho::analyze(fit)$values$effects$advance$posterior
mpe <- psycho::mpe(posterior)
print(mpe$MPE)
print(mpe$values)
```

---

n_factors	<i>Find Optimal Factor Number.</i>
-----------	------------------------------------

---

**Description**

Find optimal factor number using various solutions.

**Usage**

```
n_factors(df, rotate = "varimax", fm = "minres", n_max = 8)
```

**Arguments**

df	The dataframe
rotate	What rotation to use c("none", "varimax", "oblimin", "promax")
fm	Factoring method: "pa" for Principal Axis Factor Analysis, "minres" (default) for minimum residual (OLS) factoring, "mle" for Maximum Likelihood FA and "pc" for Principal Components
n_max	How many factors to test.

**Value**

output

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
df <- dplyr::select_if(attitude, is.numeric)
results <- psycho::n_factors(df)

summary(results)
plot(results)
```

```
# See details on methods
psycho::values(results)$methods
```

---

odds_to_probs	<i>Convert (log)odds to probabilities.</i>
---------------	--

---

### Description

Convert (log)odds to probabilities.

### Usage

```
odds_to_probs(odds, subset = NULL, except = NULL, log = TRUE)
```

### Arguments

odds	Odds values in vector or dataframe.
subset	Character or list of characters of column names to be transformed.
except	Character or list of characters of column names to be excluded from transformation.
log	Are these Log odds (such as in logistic models)?

### Author(s)

**Dominique Makowski**

### Examples

```
library(psycho)
odds_to_probs(-1.45)
```

---

overlap	<i>Overlap of Two Empirical Distributions.</i>
---------	--

---

### Description

A method to calculate the overlap coefficient of two kernel density estimates (a measure of similarity between two samples).

### Usage

```
overlap(x, y, method = "trapezoid")
```

**Arguments**

x	A vector of values from a probability distribution (e.g., posterior probabilities from MCMC sampling).
y	Scalar between 0 and 1, indicating the mass within the credible interval that is to be estimated.
method	Method of AUC computation. Can be "trapezoid" (default), "step" or "spline".

**Author(s)**

S. Venne

**Examples**

```
library(psycho)

x <- rnorm(100, 1, 0.5)
y <- rnorm(100, 0, 1)
overlap(x, y)
```

---

percentile

*Transform z score to percentile.*

---

**Description**

Transform z score to percentile.

**Usage**

```
percentile(z_score)
```

**Arguments**

z\_score          Z score.

**Author(s)**

**Dominique Makowski**

**Examples**

```
library(psycho)
percentile(-1.96)
```

percentile\_to\_z      *Transform a percentile to a z score.*

---

**Description**

Transform a percentile to a z score.

**Usage**

```
percentile_to_z(percentile)
```

**Arguments**

percentile      Percentile

**Author(s)**

[Dominique Makowski](#)

**Examples**

```
library(psycho)
percentile_to_z(95)
```

---

plot.psychobject      *Plot the results.*

---

**Description**

Plot the results.

**Usage**

```
## S3 method for class 'psychobject'
plot(x, ...)
```

**Arguments**

x                      A psychobject class object.  
...                     Arguments passed to or from other methods.

**Author(s)**

[Dominique Makowski](#)

---

power_analysis	<i>Power analysis for fitted models.</i>
----------------	--

---

### Description

Compute the n models based on n sampling of data.

### Usage

```
power_analysis(fit, n_max, n_min = NULL, step = 1, n_batch = 1,  
              groups = NULL, verbose = TRUE, CI = 90, effsize = FALSE,  
              bayes_factor = FALSE, overlap = FALSE)
```

### Arguments

fit	A lm or stanreg model.
n_max	Max sample size.
n_min	Min sample size. If null, take current nrow.
step	Increment of the sequence.
n_batch	Number of iterations at each sample size.
groups	Grouping variable name (string) to preserve proportions. Can be a list of strings.
verbose	Print progress.
CI	Argument for <a href="#">analyze</a> .
effsize	Argument for <a href="#">analyze</a> .
bayes_factor	Argument for <a href="#">analyze</a> .
overlap	Argument for <a href="#">analyze</a> .

### Value

A dataframe containing the summary of all models for all iterations.

### Author(s)

**Dominique Makowski**

### Examples

```
## Not run:  
library(dplyr)  
library(psycho)  
  
fit <- lm(Sepal.Length ~ Sepal.Width, data=iris)  
  
results <- power_analysis(fit, n_max=300, n_min=100, step=5, n_batch=20)
```

```
results %>%
  filter(Variable=="Sepal.Width") %>%
  select(n, p) %>%
  group_by(n) %>%
  summarise(p_median = median(p),
            p_mad = mad(p))

## End(Not run)
```

---

```
print.psychobject      Print the results.
```

---

### Description

Print the results.

### Usage

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

### Arguments

x                    A psychobject class object.  
...                   Further arguments passed to or from other methods.

### Author(s)

[Dominique Makowski](#)

---

```
probs_to_odds          Convert probabilities to (log)odds.
```

---

### Description

Convert probabilities to (log)odds.

### Usage

```
probs_to_odds(probs, log = FALSE)
```

### Arguments

probs                Probabilities values in vector or dataframe.  
log                   Compute log odds (such as in logistic models)?



**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)
probs_to_odds(0.75)
```

---

rnorm_perfect	<i>Perfect Normal Distribution.</i>
---------------	-------------------------------------

---

**Description**

Generates a sample of size n with a near-perfect normal distribution.

**Usage**

```
rnorm_perfect(n, mean = 0, sd = 1, method = "qnorm", iter = 10000)
```

**Arguments**

n	number of observations. If length(n) > 1, the length is taken to be the number required.
mean	vector of means.
sd	vector of standard deviations.
method	"qnorm" or "average".
iter	number of iterations (precision).

**Author(s)**

Dominique Makowski

**Examples**

```
library(psycho)
x <- rnorm_perfect(10)
plot(density(x))
```

---

standardize	<i>Standardize (scale and reduce) numeric variables.</i>
-------------	--

---

### Description

Select numeric variables and standardize (Z-score, "normalize") them.

### Usage

```
standardize(df, subset = NULL, except = NULL)
```

### Arguments

df	Dataframe.
subset	Character or list of characters of column names to be standardized.
except	Character or list of characters of column names to be excluded from standardization.

### Value

Dataframe.

### Author(s)

**Dominique Makowski**

### Examples

```
df <- data.frame(
  Participant = as.factor(rep(1:50,each=2)),
  Condition = base::rep_len(c("A", "B"), 100),
  V1 = rnorm(100, 30, .2),
  V2 = runif(100, 3, 5),
  V3 = rnorm(100, 100, 10)
)

dfZ <- standardize(df)
dfZ <- standardize(df, except="V3")
dfZ <- standardize(df, except=c("V1", "V2"))
dfZ <- standardize(df, subset="V3")
dfZ <- standardize(df, subset=c("V1", "V2"))
```

---

summary.psychobject    *Print the results.*

---

**Description**

Print the results.

**Usage**

```
## S3 method for class 'psychobject'  
summary(object, round = NULL, ...)
```

**Arguments**

object	A psychobject class object.
round	Round the output.
...	Further arguments passed to or from other methods.

**Author(s)**

[Dominique Makowski](#)

---

values                    *Extract values as list.*

---

**Description**

Extract values as list.

**Usage**

```
values(x)
```

**Arguments**

x	A psychobject class object.
---	-----------------------------

**Author(s)**

[Dominique Makowski](#)

# Index

## \*Topic **datasets**

- affective, 3
- emotion, 15

affective, 3  
analyze, 3, 39  
analyze.glmerMod, 3, 4  
analyze.lm, 3, 5  
analyze.lmerModLmerTest, 5  
analyze.merModLmerTest, 3  
analyze.stanreg, 3, 6  
as.data.frame.density, 7  
assess, 8

correlation, 9  
crawford.test, 10  
crawford.test.freq, 12  
crawford\_dissociation.test, 12

dprime, 14

emotion, 15

find\_best\_model, 16  
find\_best\_model.merModLmerTest, 16, 17  
find\_best\_model.stanreg, 16, 18  
find\_combinations, 19  
find\_combinations.formula, 19  
find\_season, 20  
format\_digit, 21  
format\_p, 21  
format\_string, 22

get\_contrasts, 22  
get\_contrasts.glmerMod, 22, 23  
get\_contrasts.lmerModLmerTest, 24  
get\_contrasts.merModLmerTest, 22  
get\_contrasts.stanreg, 22, 25  
get\_predicted, 26  
get\_predicted.stanreg, 26, 26

hdi, 28

interpret\_bf, 29  
interpret\_d, 30  
interpret\_d\_posterior, 30  
is.mixed, 31  
is.mixed.stanreg, 31, 32  
is.psychobject, 32  
is.standardized, 33

mellenbergh.test, 33  
mpe, 34

n\_factors, 35

odds\_to\_probs, 36  
overlap, 36

p.adjust, 10  
percentile, 37  
percentile\_to\_z, 38  
plot.psychobject, 38  
power\_analysis, 39  
print.psychobject, 40  
probs\_to\_odds, 40

rnorm\_perfect, 41

standardize, 42  
summary.psychobject, 43

values, 43