

Package ‘psycho’

September 13, 2018

Type Package

Title Efficient and Publishing-Oriented Workflow for Psychological Science

Version 0.3.7

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

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

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

Description The main goal of the psycho package is to provide tools for psychologists, neuropsychologists and neuroscientists, to facilitate and speed up the time spent on data analysis. It aims at supporting best practices and tools to format the output of statistical methods to directly paste them into a manuscript, ensuring statistical reporting standardization and conformity.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Depends R (>= 3.5.0)

Imports methods, dplyr, ggplot2, tidyr, stringr, purrr, psych, MASS, nFactors, qgraph, ppcor, ggcorrplot, rstanarm, rstantools, MuMIn, lme4, lmerTest, coda, emmeans (>= 1.2.2), broom, tibble, DescTools, BayesFactor (>= 0.9.1), scales, loo (>= 2.0.0), lavaan

Suggests knitr, rmarkdown, testthat, covr, plotly, GPArotation

VignetteBuilder knitr

NeedsCompilation no

Author Dominique Makowski [aut, cre, cph]
(<<https://orcid.org/0000-0001-5375-9967>>),
Hugo Najberg [ctb],
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 2018-09-13 10:20:03 UTC

R topics documented:

affective	4
analyze	5
analyze.aov	5
analyze.fa	7
analyze.glm	8
analyze.glmerMod	9
analyze.htest	10
analyze.lavaan	11
analyze.lm	12
analyze.lmerModLmerTest	13
analyze.stanreg	14
as.data.frame.density	15
assess	16
bayes_cor	17
bayes_cor.test	18
correlation	19
crawford.test	20
crawford.test.freq	21
crawford_dissociation.test	22
create_intervals	23
dprime	24
emotion	26
find_best_model	26
find_best_model.lmerModLmerTest	27
find_best_model.stanreg	28
find_combinations	29
find_combinations.formula	30
find_matching_string	31
find_random_effects	31
find_season	32
format_bf	33
format_digit	33
format_formula	34
format_loadings	34
format_p	35
format_string	35
get_cfa_model	36
get_contrasts	37
get_contrasts.glmerMod	37
get_contrasts.lmerModLmerTest	38
get_contrasts.stanreg	39
get_data	40

get_formula	41
get_info	42
get_info.lm	42
get_info.lmerModLmerTest	43
get_loadings_max	44
get_predicted	45
get_predicted.glm	45
get_predicted.lm	46
get_predicted.merMod	47
get_predicted.stanreg	49
get_R2	50
get_R2.glm	51
get_R2.lm	52
get_R2.merMod	53
get_R2.stanreg	53
HDI	54
HDImax	55
HDImin	55
interpret_bf	56
interpret_d	56
interpret_d_posterior	57
interpret_odds	58
interpret_odds_posterior	59
interpret_omega_sq	59
interpret_r	60
interpret_R2	61
interpret_R2_posterior	62
interpret_RMSEA	62
interpret_r_posterior	63
is.mixed	64
is.mixed.stanreg	64
is.psychobject	65
is.standardized	65
mellenbergh.test	66
model_to_priors	66
mpe	67
n_factors	68
odds_to_d	69
odds_to_probs	70
omega_sq	70
overlap	71
percentile	72
percentile_to_z	72
plot.psychobject	73
plot_loadings	73
power_analysis	74
print.psychobject	75
probs_to_odds	76

R2_LOO_Adjusted	76
R2_nakagawa	77
R2_tjur	78
refdata	79
remove_empty_cols	79
reorder_matrix	80
rnorm_perfect	80
rope	81
simulate_data_regression	82
standardize	83
standardize.data.frame	83
standardize.glm	85
standardize.lm	86
standardize.numeric	87
standardize.stanreg	87
summary.psychobject	88
values	89
Index	90

affective *Data from the Affective Style Questionnaire (ASQ - French Validation)*

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

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

Description

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

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

- [analyze.htest](#)
- [analyze.aov](#)

- [analyze.fa](#)
- [analyze.lavaan](#)

Usage

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

Arguments

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

Author(s)

Dominique Makowski

analyze.aov	<i>Analyze aov and anova objects.</i>
-------------	---------------------------------------

Description

Analyze aov and anova objects.

Usage

```
## S3 method for class 'aov'  
analyze(x, effsize_rules = "field2013", ...)
```

Arguments

x aov object.
 effsize_rules Grid for effect size interpretation. See [interpret_omega_sq](#).
 ... Arguments passed to or from other methods.

Value

output

Author(s)

Dominique Makowski

References

- Levine, T. R., & Hullett, C. R. (2002). Eta squared, partial eta squared, and misreporting of effect size in communication research. *Human Communication Research*, 28(4), 612-625.
- Pierce, C. A., Block, R. A., & Aguinis, H. (2004). Cautionary note on reporting eta-squared values from multifactor ANOVA designs. *Educational and psychological measurement*, 64(6), 916-924.

See Also

<http://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/os2>

Examples

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

df <- psycho::affective

x <- aov(df$Tolerating ~ df$Salary)
x <- aov(df$Tolerating ~ df$Salary * df$Sex)

x <- anova(lm(df$Tolerating ~ df$Salary * df$Sex))

summary(analyze(x))
print(analyze(x))

df <- psycho::emotion %>%
  mutate(Recall = ifelse(Recall == TRUE, 1, 0)) %>%
  group_by(Participant_ID, Emotion_Condition) %>%
  summarise(Recall = sum(Recall) / n())

x <- aov(Recall ~ Emotion_Condition + Error(Participant_ID), data=df)
x <- anova(lmerTest::lmer(Recall ~ Emotion_Condition + (1|Participant_ID), data=df))
analyze(x)
summary(x)
```

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

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

Description

Analyze fa objects.

Usage

```
## S3 method for class 'fa'  
analyze(x, labels = NULL, treshold = "max", ...)
```

Arguments

x	An psych object.
labels	Supply a additional column with e.g. item labels.
treshold	'max' or numeric. The treshold over which to associate an item with its component.
...	Arguments passed to or from other methods.

Value

output

Author(s)

Dominique Makowski

Examples

```
library(psycho)  
library(psych)  
  
x <- psych::fa(psych::Thurstone.33, 2)  
  
results <- analyze(x)  
print(results)  
summary(results)  
plot(results)
```

analyze.glm	Analyze glm objects.
-------------	----------------------

Description

Analyze glm objects.

Usage

```
## S3 method for class 'glm'  
analyze(x, CI = 95, effsize_rules = "cohen1988", ...)
```

Arguments

x	glm object.
CI	Confidence interval bounds. Set to NULL turn off their computation.
effsize_rules	Grid for effect size interpretation. See interpret_odds .
...	Arguments passed to or from other methods.

Value

output

Author(s)

[Dominique Makowski](#)

References

Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R² from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133-142.

See Also

["get_R2.glm"](#)

Examples

```
library(psycho)  
fit <- glm(Sex ~ Adjusting, data=psycho::affective, family="binomial")  
  
results <- analyze(fit)  
summary(results)  
print(results)
```

analyze.glmerMod *Analyze glmerMod objects.*

Description

Analyze glmerMod objects.

Usage

```
## S3 method for class 'glmerMod'  
analyze(x, CI = 95, effsize_rules = "cohen1988",  
  ...)
```

Arguments

x	merModLmerTest object.
CI	Bootsrapped confidence interval bounds (slow). Set to NULL turn off their computation.
effsize_rules	Grid for effect size interpretation. See interpret_odds .
...	Arguments passed to or from other methods.

Value

output

Author(s)

Dominique Makowski

References

Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R² from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133-142.

Examples

```
## Not run:  
library(psycho)  
library(lme4)  
  
fit <- lme4::glmer(vs ~ wt + (1|gear), data=mtcars, family="binomial")  
  
results <- analyze(fit)  
summary(results)  
print(results)  
  
## End(Not run)
```

analyze.htest *Analyze htest (correlation, t-test...) objects.*

Description

Analyze htest (correlation, t-test...) objects.

Usage

```
## S3 method for class 'htest'  
analyze(x, effsize_rules = "cohen1988", ...)
```

Arguments

x htest object.
effsize_rules Grid for effect size interpretation. See [interpret_r](#).
... Arguments passed to or from other methods.

Value

output

Author(s)

Dominique Makowski

Examples

```
library(psycho)  
  
df <- psycho::affective  
  
x <- t.test(df$Tolerating, df$Adjusting)  
x <- t.test(df$Tolerating ~ df$Sex)  
x <- t.test(df$Tolerating, mu=2)  
x <- cor.test(df$Tolerating, df$Adjusting)  
  
results <- analyze(x)  
summary(results)  
print(results)
```

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

Description

Analyze aov objects.

Usage

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

Arguments

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

Value

output

Author(s)

Dominique Makowski

See Also

https://www.researchgate.net/post/Whats_the_standard_of_fit_indices_in_SEM

Examples

```
library(psycho)  
library(lavaan)  
  
model <- ' visual =~ x1 + x2 + x3  
          textual =~ x4 + x5 + x6  
          speed  =~ x7 + x8 + x9 '  
x <- lavaan::cfa(model, data=HolzingerSwineford1939)  
  
rez <- analyze(x)  
print(rez)
```

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

Description

Analyze lm objects.

Usage

```
## S3 method for class 'lm'  
analyze(x, CI = 95, effsize_rules = "cohen1988", ...)
```

Arguments

x	lm object.
CI	Confidence interval bounds. Set to NULL turn off their computation.
effsize_rules	Grid for effect size interpretation. See interpret_d .
...	Arguments passed to or from other methods.

Value

output

Author(s)

Dominique Makowski

Examples

```
library(psycho)  
fit <- lm(Sepal.Length ~ Sepal.Width, data=iris)  
fit <- lm(Sepal.Length ~ Sepal.Width * Species, 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,  
        effsize_rules = "cohen1988", ...)
```

Arguments

<code>x</code>	lmerModLmerTest object.
<code>CI</code>	Bootsrapped confidence interval bounds (slow). Set to NULL turn off their computation.
<code>effsize_rules</code>	Grid for effect size interpretation. See interpret_d .
<code>...</code>	Arguments passed to or from other methods.

Value

output

Author(s)

[Dominique Makowski](#)

References

Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133-142.

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, index = "overlap",
        ROPE_bounds = NULL, effsize = FALSE, effsize_rules = "cohen1988",
        ...)
```

Arguments

x	A stanreg model.
CI	Credible interval bounds.
index	Index of effect existence to report. Can be 'overlap' or 'ROPE'.
ROPE_bounds	Bounds of the ROPE. If NULL and effsize is TRUE, than the ROPE. will have default values c(-0.1, 0.1) and computed on the standardized posteriors.
effsize	Compute Effect Sizes according to Cohen (1988). For linear models only.
effsize_rules	Grid for effect size interpretation. See interpret_d .
...	Arguments passed to or from other methods.

Value

Contains the following indices:

- the Median of the posterior distribution of the parameter (can be used as a point estimate, similar to the beta of frequentist models).
- the Median Absolute Deviation (MAD), a robust measure of dispersion (could be seen as a robust version of SD).
- the Credible Interval (CI) (by default, the 90% CI; see Kruschke, 2018), representing a range of possible parameter.
- the Maximum Probability of Effect (MPE), the probability that the effect is positive or negative (depending on the median's direction).
- the Overlap (O), the percentage of overlap between the posterior distribution and a normal distribution of mean 0 and same SD than the posterior. Can be interpreted as the probability that a value from the posterior distribution comes from a null distribution.
- the ROPE, the proportion of the 95% CI of the posterior distribution that lies within the region of practical equivalence.

Author(s)

Dominique Makowski

See Also

["get_R2.stanreg"](#) ["bayes_R2.stanreg"](#)

Examples

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

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

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

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

fit <- rstanarm::stan_glm(Sex ~ Adjusting,
  data=psycho::affective, family="binomial")
results <- analyze(fit)
summary(results)

fit <- rstanarm::stan_glmer(Sex ~ Adjusting + (1|Salary),
  data=psycho::affective, family="binomial")
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

Value

output

Author(s)

Dominique Makowski

Examples

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

bayes_cor

Bayesian Correlation Matrix.

Description

Bayesian Correlation Matrix.

Usage

```
bayes_cor(df, df2 = NULL, reorder = TRUE)
```

Arguments

df	The dataframe.
df2	Optional dataframe to correlate with the first one.
reorder	Reorder matrix by correlation strength. Only for square matrices.

Value

A list of dataframes

Author(s)

Dominique Makowski

Examples

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

df <- psycho::affective
cor <- bayes_cor(df)
summary(cor)
print(cor)
plot(cor)

df <- select(psycho::affective, Adjusting, Tolerating)
df2 <- select(psycho::affective, -Adjusting, -Tolerating)
cor <- bayes_cor(df, df2)
summary(cor)
print(cor)
plot(cor)

## End(Not run)
```

bayes_cor.test	<i>Performs a Bayesian correlation.</i>
----------------	---

Description

Performs a Bayesian correlation.

Usage

```
bayes_cor.test(x, y, CI = 90, iterations = 10000,
  effsize_rules_r = "cohen1988", effsize_rules_bf = "jeffreys1961")
```

Arguments

x	First continuous variable.
y	Second continuous variable.
CI	Credible interval bounds.
iterations	The number of iterations to sample.
effsize_rules_r	Grid for effect size interpretation. See interpret_r .
effsize_rules_bf	Grid for effect size interpretation. See interpret_bf .

Value

A psychobject.

Author(s)

Dominique Makowski

Examples

```
## Not run:
library(psycho)
x <- psycho::affective$Concealing
y <- psycho::affective$Tolerating

bayes_cor.test(x, y)
summary(bayes_cor.test(x, y))

## End(Not run)
```

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 <code>qgraph::cor_auto</code> 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, threshold = 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.

threshold	Significance threshold.
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	Alpha 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 = TRUE)
```

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)
```

create_intervals *Overlap of Two Empirical Distributions.*

Description

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

Usage

```
create_intervals(x, n = NULL, length = NULL, equal_range = TRUE,
  labels = NULL, dig.lab = 3)
```

Arguments

x	A vector of numerics.
n	Number of intervals to create, OR
length	Length of each interval.
equal_range	Makes n groups with with equal range (TRUE) or (approximately) equal numbers of observations (FALSE).
labels	Can be a custom list, "NULL", "FALSE" or "median".
dig.lab	Integer which is used when labels are not given. It determines the number of digits used in formatting the break numbers.

Author(s)

Dominique Makowski

Examples

```
library(psycho)

x <- rnorm(100, 0, 1)

create_intervals(x, n=4)
create_intervals(x, n=4, equal_range=FALSE)
create_intervals(x, length=1)

create_intervals(x, n=4, labels="median")
create_intervals(x, n=4, labels=FALSE)
```

dprime

Dprime and Other Signal Detection Theory indices.

Description

Computes Signal Detection Theory indices (d' , β , 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 β , the A' and the $B''D$ based on the signal detection theory (SRT). See Pallier (2002) for the algorithms.

Returns a list containing the following indices:

- **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.
- **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".
- **aprim** (**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.

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	<i>Emotional Ratings of Pictures</i>
---------	--------------------------------------

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	<i>Returns the best model.</i>
-----------------	--------------------------------

Description

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

- [find_best_model.stanreg](#)
- [find_best_model.lmerModLmerTest](#)

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.lmerModLmerTest`

Returns the best combination of predictors for lmerTest objects.

Description

Returns the best combination of predictors for lmerTest objects.

Usage

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

Arguments

`fit` A merModLmerTest object.
`interaction` Include interaction term.
`fixed` Additional formula part to add at the beginning of each formula
... 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

fit	A stanreg object.
interaction	Include interaction term.
fixed	Additional formula part to add at the beginning of each formula
K	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.
k_treshold	Threshold for flagging estimates of the Pareto shape parameters k estimated by loo.
...	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	<i>Generate all combinations.</i>
-------------------	-----------------------------------

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_matching_string *Fuzzy string matching.*

Description

Fuzzy string matching.

Usage

```
find_matching_string(x, y, value = TRUE, step = 0.1,  
  ignore.case = TRUE)
```

Arguments

x	Strings.
y	List of strings to be matched.
value	Return value or the index of the closest string.
step	Step by which decrease the distance.
ignore.case	if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Author(s)

Dominique Makowski

Examples

```
library(psycho)  
find_matching_string("Hwo rea ouy", c("How are you", "Not this word", "Nice to meet you"))
```

find_random_effects *Find random effects in formula.*

Description

Find random effects in formula.

Usage

```
find_random_effects(formula)
```

Arguments

formula	Formula
---------	---------

Author(s)

Dominique Makowski

Examples

```
library(psycho)
find_random_effects("Y ~ X + (1|Group)")
```

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_bf	<i>Bayes factor formatting</i>
-----------	--------------------------------

Description

Bayes factor formatting

Usage

```
format_bf(bf, max = 100)
```

Arguments

bf	Bayes Factor.
max	Threshold for maximum.

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

Description

Format digits.

Usage

```
format_digit(x, digits = 2, null_threshold = 0.001,  
            inf_threshold = 9e+08)
```

Arguments

x	A digit.
digits	Number of significant digits.
null_threshold	Threshold below which return 0.
inf_threshold	Threshold above which return Inf.

Author(s)

[Dominique Makowski](#)

format_formula *Clean and format formula.*

Description

Clean and format formula.

Usage

```
format_formula(formula)
```

Arguments

formula	formula
...	Arguments passed to or from other methods.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
library(lme4)

fit <- lme4::glmer(vs ~ wt + (1|gear), data=mtcars, family="binomial")
fit <- lm(hp ~ wt, data=mtcars)

format_formula(get_formula(fit))
```

format_loadings *Format the loadings of a factor analysis.*

Description

Format the loadings of a factor analysis.

Usage

```
format_loadings(x, labels = NULL)
```

Arguments

x	An psych object.
labels	Supply a additional column with e.g. item labels.

Author(s)

Dominique Makowski

Examples

```
## Not run:  
library(psycho)  
  
x <- psych::fa(psych::Thurstone.33, 2)  
format_loadings(x)  
  
## End(Not run)
```

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

Description

Format p values.

Usage

```
format_p(pvalues, stars = TRUE)
```

Arguments

pvalues	P values (scalar or vector).
stars	Add stars.

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_cfa_model	<i>Get CFA model.</i>
---------------	-----------------------

Description

Get CFA model.

Usage

```
get_cfa_model(loadings, treshold = "max")
```

Arguments

loadings	Formatted loadings.
treshold	‘max’ or numeric. The treshold over which to associate an item with its component.

Author(s)

Dominique Makowski

Examples

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

x <- psych::fa(psych::Thurstone.33, 2)
loadings <- format_loadings(x)$loadings
get_cfa_model(loadings, treshold="max")
get_cfa_model(loadings, treshold=0.1)

## End(Not run)
```

get_contrasts *Get Marginal Means and Contrasts.*

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

fit	A glmerMod 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", "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(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=affektive)

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_data

Extract the dataframe used in a model.

Description

Extract the dataframe used in a model.

Usage

```
get_data(fit, ...)
```

Arguments

```
fit          A model.
...          Arguments passed to or from other methods.
```

Author(s)

[Dominique Makowski](#)

Examples

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

df <- mtcars %>%
  mutate(cyl = as.factor(cyl),
         gear = as.factor(gear))

fit <- lm(wt ~ mpg , data=df)
fit <- lm(wt ~ cyl, data=df)
fit <- lm(wt ~ mpg * cyl, data=df)
fit <- lm(wt ~ cyl * gear, data=df)
fit <- lmerTest::lmer(wt ~ mpg * gear + (1|cyl), data=df)
```



```
fit <- rstanarm::stan_lmer(wt ~ mpg * gear + (1|cyl), data=df)

get_data(fit)

## End(Not run)
```

get_formula	<i>Get formula of models.</i>
-------------	-------------------------------

Description

Get formula of models. Implemented for:

- analyze.merModLmerTest
- analyze.glmerMod
- analyze.lm
- analyze.glm
- analyze.stanreg

Usage

```
get_formula(x, ...)
```

Arguments

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

Author(s)

[Dominique Makowski](#)

Examples

```
library(psycho)
library(lme4)

fit <- lme4::glmer(vs ~ wt + (1|gear), data=mtcars, family="binomial")
fit <- lm(hp ~ wt, data=mtcars)

get_formula(fit)
```

get_info *Get information about objects.*

Description

Get information about models.

Usage

```
get_info(x, ...)
```

Arguments

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

Value

output

Author(s)

Dominique Makowski

Examples

```
library(psycho)
library(lme4)

fit <- lme4::glmer(vs ~ wt + (1|gear), data=mtcars, family="binomial")

info <- get_info(fit)
info
```

get_info.lm *Get information about models.*

Description

Get information about models.

Usage

```
## S3 method for class 'lm'
get_info(x, ...)
```

Arguments

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

Value

output

Author(s)

Dominique Makowski

Examples

```
library(psycho)
library(lme4)

fit <- lm(vs ~ wt, data=mtcars, family="binomial")

info <- get_info(fit)
info
```

get_info.lmerModLmerTest
Get information about models.

Description

Get information about models.

Usage

```
## S3 method for class 'lmerModLmerTest'
get_info(x, ...)
```

Arguments

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

Value

output

Author(s)

Dominique Makowski

Examples

```
library(psycho)
library(lme4)

fit <- lme4::glmer(vs ~ wt + (1|gear), data=mtcars, family="binomial")

info <- get_info(fit)
info
```

get_loadings_max	<i>Get loadings max.</i>
------------------	--------------------------

Description

Get loadings max.

Usage

```
get_loadings_max(loadings)
```

Arguments

loadings Formatted loadings.

Author(s)

Dominique Makowski

Examples

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

x <- psych::fa(psych::Thurstone.33, 2)
get_loadings_max(format_loadings(x)$loadings)

## 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](#)
- [get_predicted.merMod](#)
- [get_predicted.lm](#)
- [get_predicted.glm](#)

Usage

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

Arguments

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

Author(s)

Dominique Makowski

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

Description

Compute predicted from a lm model.

Usage

```
## S3 method for class 'glm'
get_predicted(fit, newdata = "model", prob = 0.95,
             odds_to_probs = TRUE, ...)
```

Arguments

fit	An lm 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 confidence intervals (0.9 (default) will compute 2.5-97.5% CI). Can also be a list of probs (e.g., c(0.90, 0.95)).
odds_to_probs	Transform log odds ratios in logistic models to probabilities.
...	Arguments passed to or from other methods.

Value

dataframe with predicted values.

Author(s)

Dominique Makowski

Examples

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

fit <- glm(Sex ~ Adjusting, data=affektive, family="binomial")

refgrid <- psycho::refdata(affektive, "Adjusting")
predicted <- get_predicted(fit, newdata=refgrid)

ggplot(predicted, aes(x=Adjusting, y=Sex_Predicted)) +
  geom_line() +
  geom_ribbon(aes(ymin=Sex_CI_2.5,
                 ymax=Sex_CI_97.5),
            alpha=0.1)

## End(Not run)
```

get_predicted.lm

Compute predicted values of lm models.

Description

Compute predicted from a lm model.

Usage

```
## S3 method for class 'lm'
get_predicted(fit, newdata = "model", prob = 0.95, ...)
```

Arguments

fit	An lm 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 confidence intervals (0.95 (default) will compute 2.5-97.5% CI). Can also be a list of probs (e.g., c(0.90, 0.95)).
...	Arguments passed to or from other methods.

Value

dataframe with predicted values.

Author(s)

Dominique Makowski

Examples

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

fit <- lm(Tolerating ~ Adjusting, data=affektive)

refgrid <- psycho::refdata(affektive, "Adjusting")
predicted <- get_predicted(fit, newdata=refgrid)

ggplot(predicted, aes(x=Adjusting, y=Tolerating_Predicted)) +
  geom_line() +
  geom_ribbon(aes(ymin=Tolerating_CI_2.5,
                 ymax=Tolerating_CI_97.5),
            alpha=0.1)

## End(Not run)
```

get_predicted.merMod *Compute predicted values of lm models.*

Description

Compute predicted from a lm model.

Usage

```
## S3 method for class 'merMod'
get_predicted(fit, newdata = "model", prob = NULL,
             odds_to_probs = TRUE, iter = 100, seed = NULL, re.form = NULL,
             use.u = FALSE, ...)
```

Arguments

fit	An lm 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.

<code>prob</code>	Probability of confidence intervals (0.95 will compute 2.5-97.5% CI). Can also be a list of probs (e.g., <code>c(0.90, 0.95)</code>). Default to NULL as it takes a very long time to compute (see bootMer).
<code>odds_to_probs</code>	Transform log odds ratios in logistic models to probabilities.
<code>iter</code>	An integer indicating the number of iterations for bootstrapping (when <code>prob</code> is not null).
<code>seed</code>	An optional seed to use.
<code>re.form</code>	Formula for random effects to condition on. If NULL, include all random effects; if NA or <code>~0</code> , include no random effects (see predict.merMod).
<code>use.u</code>	logical, indicating whether the spherical random effects should be simulated / bootstrapped as well. If TRUE, they are not changed, and all inference is conditional on these values. If FALSE, new normal deviates are drawn (see bootMer).
<code>...</code>	Arguments passed to or from other methods.

Value

dataframe with predicted values.

Author(s)

[Dominique Makowski](#)

Examples

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

fit <- lmerTest::lmer(Tolerating ~ Adjusting + (1|Salary), data=affektive)

refgrid <- psycho::refdata(affektive, "Adjusting")
predicted <- get_predicted(fit, newdata=refgrid)

ggplot(predicted, aes(x=Adjusting, y=Tolerating_Predicted)) +
  geom_line()

predicted <- get_predicted(fit, newdata=refgrid, prob=0.95, iter=100) # Takes a long time

ggplot(predicted, aes(x=Adjusting, y=Tolerating_Predicted)) +
  geom_line() +
  geom_ribbon(aes(ymin=Tolerating_CI_2.5,
                 ymax=Tolerating_CI_97.5),
            alpha=0.1)

fit <- lme4::glmer(Sex ~ Adjusting + (1|Salary), data=affektive, family="binomial")

refgrid <- psycho::refdata(affektive, "Adjusting")
```



```

predicted <- get_predicted(fit, newdata=refgrid)

ggplot(predicted, aes(x=Adjusting, y=Sex_Predicted)) +
  geom_line()

predicted <- get_predicted(fit, newdata=refgrid, prob=0.95, iter=100) # Takes a long time

ggplot(predicted, aes(x=Adjusting, y=Sex_Predicted)) +
  geom_line() +
  geom_ribbon(aes(ymin=Sex_CI_2.5,
                 ymax=Sex_CI_97.5),
            alpha=0.1)

## End(Not run)

```

get_predicted.stanreg *Compute predicted values of stanreg models.*

Description

Compute predicted from a stanreg model.

Usage

```

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

```

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)).
odds_to_probs	Transform log odds ratios in logistic models to probabilities.
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.
posterior_predict	Posterior draws of the outcome instead of the link function (i.e., the regression "line").
seed	An optional seed to use.

`transform` If `posterior_predict` is `False`, should the linear predictor be transformed using the inverse-link function? The default is `FALSE`, in which case the untransformed linear predictor is returned.

`...` 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)

refgrid <- psycho::refdata(affektive, "Adjusting")
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")

refgrid <- psycho::refdata(affektive, "Adjusting")
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)
```

Description

See the documentation for your object's class:

- [get_R2.lm](#)
- [get_R2.glm](#)
- [get_R2.stanreg](#)

Usage

```
get_R2(fit, ...)
```

Arguments

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

Author(s)

[Dominique Makowski](#)

get_R2.glm

Pseudo-R-squared for Logistic Models.

Description

Pseudo-R-squared for Logistic Models.

Usage

```
## S3 method for class 'glm'  
get_R2(fit, method = "nakagawa", ...)
```

Arguments

fit	A logistic model.
method	Can be " nakagawa " or " tjur ".
...	Arguments passed to or from other methods.

Author(s)

[Dominique Makowski](#)

Examples

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

fit <- glm(vs ~ wt, data=mtcars, family="binomial")
fit <- glm(Sex ~ Adjusting, data=psycho::affective, family="binomial")

get_R2(fit)

## End(Not run)
```

`get_R2.lm`*R2 and adjusted R2 for Linear Models.*

Description

R2 and adjusted R2 for Linear Models.

Usage

```
## S3 method for class 'lm'
get_R2(fit, ...)
```

Arguments

<code>fit</code>	A linear model.
<code>...</code>	Arguments passed to or from other methods.

Author(s)

Dominique Makowski

Examples

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

fit <- lm(Tolerating ~ Adjusting, data=psycho::affective)

get_R2(fit)

## End(Not run)
```

get_R2.merMod	<i>R2 and adjusted R2 for GLMMs.</i>
---------------	--------------------------------------

Description

R2 and adjusted R2 for GLMMs.

Usage

```
## S3 method for class 'merMod'  
get_R2(fit, ...)
```

Arguments

fit	A GLMM.
...	Arguments passed to or from other methods.

Author(s)

Dominique Makowski

Examples

```
## Not run:  
library(psycho)  
  
fit <- lmerTest::lmer(Tolerating ~ Adjusting + (1|Sex), data=psycho::affective)  
fit <- lme4::glmer(Sex ~ Adjusting + (1|Salary), data=na.omit(psycho::affective), family="binomial")  
  
get_R2(fit)  
  
## End(Not run)
```

get_R2.stanreg	<i>R2 or Bayesian Models.</i>
----------------	-------------------------------

Description

Computes R2 and [LOO-adjusted R2](#).

Usage

```
## S3 method for class 'stanreg'  
get_R2(fit, silent = FALSE, ...)
```

Arguments

<code>fit</code>	A stanreg model.
<code>silent</code>	If R2 not available, throw warning.
<code>...</code>	Arguments passed to or from other methods.

Author(s)

[Dominique Makowski](#)

See Also

["bayes_R2.stanreg"](#)

Examples

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

fit <- rstanarm::stan_glm(Adjusting ~ Tolerating, data=psycho::affective)

get_R2(fit)

## 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

<code>x</code>	A vector of values from a probability distribution (e.g., posterior probabilities from MCMC sampling).
<code>prob</code>	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)
```

HDI _{max}	<i>Highest Density Intervals (HDI)</i>
--------------------	--

Description

See [HDI](#)

Usage

```
HDImax(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.

HDI _{min}	<i>Highest Density Intervals (HDI)</i>
--------------------	--

Description

See [HDI](#)

Usage

```
HDImin(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.

interpret_bf	<i>Bayes Factor Interpretation</i>
--------------	------------------------------------

Description

Return the interpretation of a Bayes Factor.

Usage

```
interpret_bf(x, direction = TRUE, bf = TRUE, rules = "jeffreys1961")
```

Arguments

x	Bayes Factor.
direction	Include direction (against / in favour).
bf	Include Bayes Factor.
rules	Can be "jeffreys1961" (default), "raftery1995", or a custom list.

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(x=10)
```

interpret_d	<i>Standardized difference (Cohen's d) interpretation.</i>
-------------	--

Description

Interpret d with a set of rules.

Usage

```
interpret_d(x, direction = FALSE, rules = "cohen1988")
```


Arguments

x	Standardized difference.
direction	Return direction.
rules	Can be "cohen1988" (default), "sawilowsky2009", or a custom list.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
interpret_d(-0.42)
interpret_d(-0.62)
```

interpret_d_posterior *Standardized difference (Cohen's d) interpretation for a posterior distribution.*

Description

Interpret d with a set of rules.

Usage

```
interpret_d_posterior(posterior, rules = "cohen1988")
```

Arguments

posterior	Posterior distribution of standardized differences.
rules	Can be "cohen1988" (default), "sawilowsky2009", or a custom list.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
posterior <- rnorm(1000, 0.6, 0.05)
interpret_d_posterior(posterior)
interpret_d_posterior(rnorm(1000, 0.1, 1))
```

interpret_odds	<i>Omega Squared Interpretation</i>
----------------	-------------------------------------

Description

Return the interpretation of Omegas Squared.

Usage

```
interpret_odds(x, log = FALSE, direction = FALSE, rules = "chen2010")
```

Arguments

x	Odds ratio.
log	Are these log odds ratio?
direction	Return direction.
rules	Can be "chen2010" (default), "cohen1988" (through log odds to Cohen's d transformation) or a custom list.

Author(s)

Dominique Makowski

References

- Chen, H., Cohen, P., & Chen, S. (2010). How big is a big odds ratio? Interpreting the magnitudes of odds ratios in epidemiological studies. *Communications in Statistics—Simulation and Computation*, 39(4), 860-864.

See Also

<http://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/effectSize>

Examples

```
library(psycho)
interpret_odds(x=2)
```

`interpret_odds_posterior`*Odds ratio interpretation for a posterior distribution.*

Description

Interpret odds with a set of rules.

Usage

```
interpret_odds_posterior(posterior, log = FALSE, rules = "chen2010")
```

Arguments

<code>posterior</code>	Posterior distribution of odds ratio.
<code>log</code>	Are these log odds ratio?
<code>rules</code>	Can be "chen2010" (default), "cohen1988" (through log odds to Cohen's d transformation) or a custom list.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
posterior <- rnorm(1000, 0.6, 0.05)
interpret_odds_posterior(posterior)
interpret_odds_posterior(rnorm(1000, 0.1, 1))
interpret_odds_posterior(rnorm(1000, 3, 1.5))
```

`interpret_omega_sq` *Omega Squared Interpretation*

Description

Return the interpretation of Omegas Squared.

Usage

```
interpret_omega_sq(x, rules = "field2013")
```

Arguments

x	Omega Squared.
rules	Can be "field2013" (default), or a custom list.

Author(s)

Dominique Makowski

References

- Field, A (2013) Discovering statistics using IBM SPSS Statistics. Fourth Edition. Sage:London.

See Also

<http://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/effectSize>

Examples

```
library(psycho)
interpret_omega_sq(x=0.05)
```

interpret_r	<i>Correlation coefficient r interpretation.</i>
-------------	--

Description

Interpret r with a set of rules.

Usage

```
interpret_r(x, direction = TRUE, strength = TRUE,
  rules = "cohen1988")
```

Arguments

x	Correlation coefficient.
direction	Return direction.
strength	Return strength.
rules	Can be "cohen1988" (default), "evans1996", or a custom list.

Author(s)

Dominique Makowski

See Also

Page 88 of APA's 6th Edition

Examples

```
library(psycho)
interpret_r(-0.42)
```

interpret_R2	<i>R2 interpretation.</i>
--------------	---------------------------

Description

Interpret R2 with a set of rules.

Usage

```
interpret_R2(x, rules = "cohen1988")
```

Arguments

x	Value.
rules	Can be "cohen1988" (default), "chin1998" or "hair2013", or a custom list.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
interpret_R2(x=0.42)
interpret_R2(x=c(0.42, 0.2, 0.9, 0))
```

`interpret_R2_posterior`*R2 interpretation for a posterior distribution.*

Description

Interpret R2 with a set of rules.

Usage

```
interpret_R2_posterior(posterior, rules = "cohen1988")
```

Arguments

<code>posterior</code>	Distribution of R2.
<code>rules</code>	Can be "cohen1988" (default), "chin1998" or "hair2013", or a custom list.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
posterior <- rnorm(1000, 0.4, 0.1)
interpret_R2_posterior(posterior)
```

`interpret_RMSEA`*RMSEA interpretation.*

Description

Interpret RMSEA with a set of rules.

Usage

```
interpret_RMSEA(x, rules = "awang2012")
```

Arguments

<code>x</code>	RMSEA.
<code>rules</code>	Can be "awang2012", or a custom list.

Author(s)

Dominique Makowski

Examples

```
library(psycho)
interpret_RMSEA(0.04)
```

interpret_r_posterior Correlation coefficient *r* interpretation for a posterior distribution.

Description

Interpret *r* with a set of rules.

Usage

```
interpret_r_posterior(posterior, rules = "cohen1988")
```

Arguments

posterior	Posterior distribution of correlation coefficient.
rules	Can be "cohen1988" (default) or "evans1996", or a custom list.

Author(s)

Dominique Makowski

See Also

Page 88 of APA's 6th Edition

Examples

```
library(psycho)
posterior <- rnorm(1000, 0.5, 0.5)
interpret_r_posterior(posterior)
```

is.mixed	<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

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

Author(s)

[Dominique Makowski](#)

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

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	<i>Creates or tests for objects of mode "psychobject".</i>
----------------	--

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 & 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)
```

model_to_priors	<i>Model to Prior.</i>
-----------------	------------------------

Description

Convert a Bayesian model's results to priors.

Usage

```
model_to_priors(fit, autoscale = FALSE)
```

Arguments

`fit` A stanreg model.
`autoscale` Set autoscale.

Author(s)

Dominique Makowski

Examples

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

fit <- stan_glm(Sepal.Length ~ Petal.Width, data=iris)
priors <- model_to_priors(fit)
update(fit, prior=priors$prior)

fit <- stan_glm(Subjective_Valence ~ Emotion_Condition + (1|Participant_ID), data=psycho::emotion)
priors <- model_to_priors(fit)

fit1 <- stan_glm(Subjective_Valence ~ Emotion_Condition,
  data=filter(psycho::emotion, Participant_ID == "1S"))

fit2 <- stan_glm(Subjective_Valence ~ Emotion_Condition,
  data=filter(psycho::emotion, Participant_ID == "1S"),
  prior=priors$prior, prior_intercept=priors$prior_intercept)

## End(Not run)
```

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

Find Optimal Factor Number.

Description

Find optimal components number using maximum method agreement.

Usage

```
n_factors(df, rotate = "varimax", fm = "minres", n = NULL)
```

Arguments

df	A dataframe or correlation matrix
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	If correlation matrix is passed, the sample size.

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_d	<i>(Log) odds ratio to Cohen's d</i>
-----------	--------------------------------------

Description

(Log) odds ratio to Cohen's d.

Usage

```
odds_to_d(x, log = TRUE)
```

Arguments

x	Odds ratio.
log	Are these log odds ratio?

Author(s)

Dominique Makowski

References

- Sánchez-Meca, J., Marín-Martínez, F., & Chacón-Moscoso, S. (2003). Effect-size indices for dichotomized outcomes in meta-analysis. *Psychological methods*, 8(4), 448.

See Also

<https://www.meta-analysis.com/downloads/Meta-analysis>

Examples

```
library(psycho)
odds_to_d(x=2)
```

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)
```

omega_sq	<i>Partial Omega Squared.</i>
----------	-------------------------------

Description

Partial Omega Squared.

Usage

```
omega_sq(x, partial = TRUE)
```

Arguments

x	aov object.
partial	Return partial omega squared.

Value

output

Author(s)

Arnoud Plantinga

See Also

<http://stats.stackexchange.com/a/126520>

Examples

```
library(psycho)

df <- psycho::affective

x <- aov(df$Tolerating ~ df$Salary)
x <- aov(df$Tolerating ~ df$Salary * df$Sex)

omega_sq(x)
```

overlap

Overlap of Two Empirical Distributions.

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

plot_loadings *Plot loadings.*

Description

Plot loadings.

Usage

```
plot_loadings(loadings)
```

Arguments

loadings Loadings by variable.

Author(s)

Dominique Makowski

Examples

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

x <- psych::fa(psych::Thurstone.33, 2)
plot_loadings(format_loadings(x)$loadings)

## End(Not run)
```

power_analysis

Power analysis for fitted models.

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,
  effsize_rules = "cohen1988", 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 analyze .
effsize	Argument for analyze .
effsize_rules	Argument for analyze .
bayes_factor	Argument for analyze .
overlap	Argument for analyze .

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

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)
```

R2_LOO_Adjusted	<i>Compute LOO-adjusted R2.</i>
-----------------	---------------------------------

Description

Compute LOO-adjusted R2.

Usage

```
R2_LOO_Adjusted(fit)
```

Arguments

fit	A stanreg model.
-----	------------------

Author(s)

[Daniel Luedecke](#)

Examples

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

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

R2_L00_Adjusted(fit)

## End(Not run)
```

R2_nakagawa

Pseudo-R-squared for Generalized Mixed-Effect models.

Description

For mixed-effects models, R^2 can be categorized into two types. Marginal R_{GLMM}^2 represents the variance explained by fixed factors, and Conditional R_{GLMM}^2 is interpreted as variance explained by both fixed and random factors (i.e. the entire model). IMPORTANT: Looking for help to reimplement this method.

Usage

```
R2_nakagawa(fit)
```

Arguments

```
fit          A mixed model.
```

Author(s)

[Dominique Makowski](#)

References

Nakagawa, S., Johnson, P. C., & Schielzeth, H. (2017). The coefficient of determination R^2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. *Journal of the Royal Society Interface*, 14(134), 20170213. Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133-142.

Examples

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

fit <- lmerTest::lmer(Sepal.Length ~ Sepal.Width + (1|Species), data=iris)

R2_nakagawa(fit)

## End(Not run)
```

R2_tjur

Tjur's (2009) coefficient of determination.

Description

Computes Tjur's (2009) coefficient of determination.

Usage

```
R2_tjur(fit)
```

Arguments

fit Logistic Model.

Author(s)

[Daniel Lüdtke](#)

References

Tjur, T. (2009). Coefficients of determination in logistic regression models—A new proposal: The coefficient of discrimination. *The American Statistician*, 63(4), 366-372.

Examples

```
library(psycho)
library(lme4)

fit <- lme4::glmer(vs ~ wt + (1|gear), data=mtcars, family="binomial")
R2_tjur(fit)
```

refdata	<i>Create a reference grid.</i>
---------	---------------------------------

Description

Create a reference grid.

Usage

```
refdata(df, target = "all", length.out = 10, factors = "reference",
        numerics = "mean")
```

Arguments

df	The dataframe.
target	String or list of strings to indicate target columns. Can be "all".
length.out	Length of numeric target variables.
factors	Type of summary for factors. Can be "combination" or "reference".
numerics	Type of summary for numerics Can be "combination", any function ("mean", "median", ...) or a value.

Author(s)

Dominique Makowski

Examples

```
library(psycho)

df <- psycho::affective
newdata <- refdata(df, target="Sex")
newdata <- refdata(df, target="Sex", factors="combinations")
newdata <- refdata(df, target=c("Sex", "Salary", "Tolerating"), length.out=3)
newdata <- refdata(df, target=c("Sex", "Salary", "Tolerating"), numerics=0)
```

remove_empty_cols	<i>Remove empty columns..</i>
-------------------	-------------------------------

Description

Removes all columns containing ony NaNs.

Usage

```
remove_empty_cols(df)
```

Arguments

df Dataframe.

Author(s)

Dominique Makowski

reorder_matrix *Reorder square matrix.*

Description

Reorder square matrix.

Usage

```
reorder_matrix(mat, dmat = NULL)
```

Arguments

mat A square matrix.
dmat A square matrix with values to use as distance.

Examples

```
library(psycho)  
  
r <- correlation(iris)  
r <- r$values$r  
r <- reorder_matrix(r)
```

rnorm_perfect *Perfect Normal Distribution.*

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 $\text{length}(n) > 1$, 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))
```

rope

Region of Practical Equivalence (ROPE)

Description

Compute the proportion of a posterior distribution that lies within a region of practical equivalence.

Usage

```
rope(posterior, bounds = c(-0.1, 0.1), CI = 95, overlap = FALSE)
```

Arguments

posterior	Posterior Distribution.
bounds	Rope lower and higher bounds.
CI	The credible interval to use.
overlap	Compute rope overlap (EXPERIMENTAL).

Value

list containing rope indices

Author(s)

Dominique Makowski

Examples

```
library(psycho)

posterior <- rnorm(1000, 0, 0.01)
results <- rope(posterior)
results$decision
```

```
simulate_data_regression
```

Simulates data for single or multiple regression.

Description

Simulates data for single or multiple regression.

Usage

```
simulate_data_regression(coefs = 0.5, sample = 100, error = 0)
```

Arguments

coefs	Desired theoretical coefs. Can be a single value or a list.
sample	Desired sample size.
error	The error (standard deviation of gaussian noise).

Details

See <https://stats.stackexchange.com/questions/59062/multiple-linear-regression-simulation>

Author(s)

TPArrow

Examples

```
library(psycho)

data <- simulate_data_regression(coefs=c(0.1, 0.8), sample=50, error=0)
fit <- lm(y ~ ., data=data)
coef(fit)
analyze(fit)
```

standardize	<i>Standardize.</i>
-------------	---------------------

Description

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

- [standardize.numeric](#)
- [standardize.data.frame](#)
- [standardize.stanreg](#)
- [standardize.lm](#)
- [standardize.glm](#)

Usage

```
standardize(x, ...)
```

Arguments

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

Author(s)

Dominique Makowski

standardize.data.frame	<i>Standardize (scale and reduce) Dataframe.</i>
------------------------	--

Description

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

Usage

```
## S3 method for class 'data.frame'  
standardize(x, subset = NULL, except = NULL,  
            normalize = FALSE, ...)
```

Arguments

x	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.
normalize	Will perform a normalization instead of a standardization. This scales all numeric variables in the range 0 - 1.
...	Arguments passed to or from other methods.

Value

Dataframe.

Author(s)

[Dominique Makowski](#)

Examples

```
## Not run:
df <- data.frame(
  Participant = as.factor(rep(1:25,each=4)),
  Condition = base::rep_len(c("A", "B", "C", "D"), 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"))
dfZ <- standardize(df, normalize=TRUE)

# Respects grouping
dfZ <- df %>%
  dplyr::group_by(Participant) %>%
  standardize(df)

## End(Not run)
```

standardize.glm	<i>Standardize Coefficients.</i>
-----------------	----------------------------------

Description

Compute standardized coefficients.

Usage

```
## S3 method for class 'glm'  
standardize(x, method = "refit", ...)
```

Arguments

x	A linear model.
method	The standardization method. Can be "refit" (will entirely refit the model based on standardized data. Can take some time) or "agresti".
...	Arguments passed to or from other methods.

Author(s)

Kamil Barton

See Also

<https://think-lab.github.io/d/205/>

Examples

```
## Not run:  
library(psycho)  
fit <- glm(Sex ~ Adjusting, data=psycho::affective, family="binomial")  
fit <- lme4::glmer(Sex ~ Adjusting + (1|Sex), data=psycho::affective, family="binomial")  
  
standardize(fit)  
  
## End(Not run)
```

standardize.lm *Standardize Coefficients.*

Description

Compute standardized coefficients.

Usage

```
## S3 method for class 'lm'
standardize(x, method = "refit", partial_sd = FALSE,
           preserve_factors = TRUE, ...)
```

Arguments

x	A linear model.
method	The standardization method. Can be "refit" (will entirely refit the model based on standardized data. Can take some time) or "posthoc".
partial_sd	Logical, if set to TRUE, model coefficients are multiplied by partial SD, otherwise they are multiplied by the ratio of the standard deviations of the independent variable and dependent variable.
preserve_factors	Standardize factors-related coefs only by the dependent variable (i.e., do not standardize the dummies generated by factors).
...	Arguments passed to or from other methods.

Author(s)

Kamil Barton

Examples

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

df <- mtcars %>%
  mutate(cyl = as.factor(cyl))

fit <- lm(wt ~ mpg * cyl, data=df)
fit <- lmerTest::lmer(wt ~ mpg * cyl + (1|gear), data=df)

summary(fit)
standardize(fit)

## End(Not run)
```

standardize.numeric *Standardize (scale and reduce) numeric variables.*

Description

Standardize (Z-score, "normalize") a vector.

Usage

```
## S3 method for class 'numeric'  
standardize(x, normalize = FALSE, ...)
```

Arguments

x	Numeric vector.
normalize	Will perform a normalization instead of a standardization. This scales all numeric variables in the range 0 - 1.
...	Arguments passed to or from other methods.

Author(s)

Dominique Makowski

Examples

```
standardize(x=c(1, 4, 6, 2))  
standardize(x=c(1, 4, 6, 2), normalize=TRUE)
```

standardize.stanreg *Standardize Posteriors.*

Description

Compute standardized posteriors from which to get standardized coefficients.

Usage

```
## S3 method for class 'stanreg'  
standardize(x, method = "refit", ...)
```

Arguments

x	A stanreg model.
method	"refit" (default) will entirely refit the model based on standardized data. Can take a long time. Other post-hoc methods are "posterior" (based on estimated SD) or "sample" (based on the sample SD).
...	Arguments passed to or from other methods.

Author(s)

Jonah Gabry, bgoodri

See Also

<https://github.com/stan-dev/rstanarm/issues/298>

Examples

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

fit <- rstanarm::stan_glm(Sepal.Length ~ Sepal.Width * Species, data=iris)
fit <- rstanarm::stan_glm(Sepal.Length ~ Sepal.Width * Species, data=standardize(iris))
posteriors <- standardize(fit)
posteriors <- standardize(fit, method="posterior")

## End(Not run)
```

```
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	<i>Extract values as list.</i>
--------	--------------------------------

Description

Extract values as list.

Usage

values(x)

Arguments

x A psychobject class object.

Author(s)

Dominique Makowski

Index

*Topic **datasets**

- affective, 4
- emotion, 26

- affective, 4
- analyze, 5, 74
- analyze.aov, 5, 5
- analyze.fa, 5, 7
- analyze.glm, 5, 8
- analyze.glmerMod, 5, 9
- analyze.htest, 5, 10
- analyze.lavaan, 5, 11
- analyze.lm, 5, 12
- analyze.lmerModLmerTest, 13
- analyze.merModLmerTest, 5
- analyze.stanreg, 5, 14
- as.data.frame.density, 15
- assess, 16

- bayes_cor, 17
- bayes_cor.test, 18
- bootMer, 48

- correlation, 19
- crawford.test, 20
- crawford.test.freq, 21
- crawford_dissociation.test, 22
- create_intervals, 23

- dprime, 24

- emotion, 26

- find_best_model, 26
- find_best_model.lmerModLmerTest, 26, 27
- find_best_model.stanreg, 26, 27, 28
- find_combinations, 29
- find_combinations.formula, 30
- find_matching_string, 31
- find_random_effects, 31
- find_season, 32

- format_bf, 33
- format_digit, 33
- format_formula, 34
- format_loadings, 34
- format_p, 35
- format_string, 35

- get_cfa_model, 36
- get_contrasts, 37
- get_contrasts.glmerMod, 37, 37
- get_contrasts.lmerModLmerTest, 38
- get_contrasts.merModLmerTest, 37
- get_contrasts.stanreg, 37, 39
- get_data, 40
- get_formula, 41
- get_info, 42
- get_info.lm, 42
- get_info.lmerModLmerTest, 43
- get_loadings_max, 44
- get_predicted, 45
- get_predicted.glm, 45, 45
- get_predicted.lm, 45, 46
- get_predicted.merMod, 45, 47
- get_predicted.stanreg, 45, 49
- get_R2, 50
- get_R2.glm, 51, 51
- get_R2.lm, 51, 52
- get_R2.merMod, 53
- get_R2.stanreg, 51, 53

- HDI, 54, 55
- HDI_max, 55
- HDI_min, 55

- interpret_bf, 18, 56
- interpret_d, 12–14, 56
- interpret_d_posterior, 57
- interpret_odds, 8, 9, 58
- interpret_odds_posterior, 59
- interpret_omega_sq, 6, 59

interpret_r, [10](#), [18](#), [60](#)
interpret_R2, [61](#)
interpret_R2_posterior, [62](#)
interpret_r_posterior, [63](#)
interpret_RMSEA, [62](#)
is.mixed, [64](#)
is.mixed.stanreg, [64](#), [64](#)
is.psychobject, [65](#)
is.standardized, [65](#)

log odds to Cohen's d transformation,
[58](#), [59](#)

L00-adjusted R2, [53](#)

mellenbergh.test, [66](#)
model_to_priors, [66](#)
mpe, [67](#)

n_factors, [68](#)

odds_to_d, [69](#)
odds_to_probs, [70](#)
omega_sq, [70](#)
overlap, [71](#)

p.adjust, [19](#)
percentile, [72](#)
percentile_to_z, [72](#)
plot.psychobject, [73](#)
plot_loadings, [73](#)
power_analysis, [74](#)
predict.merMod, [48](#)
print.psychobject, [75](#)
probs_to_odds, [76](#)

R2_L00_Adjusted, [76](#)
R2_nakagawa, [77](#)
R2_tjur, [78](#)
refdata, [79](#)
remove_empty_cols, [79](#)
reorder_matrix, [80](#)
rnorm_perfect, [80](#)
rope, [81](#)

simulate_data_regression, [82](#)
standardize, [83](#)
standardize.data.frame, [83](#), [83](#)
standardize.glm, [83](#), [85](#)
standardize.lm, [83](#), [86](#)
standardize.numeric, [83](#), [87](#)

standardize.stanreg, [83](#), [87](#)
summary.psychobject, [88](#)
values, [89](#)