

# Package ‘psycho’

March 22, 2018

**Type** Package

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

**Version** 0.1.4

**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 (>= 3.1.0)

**Imports** methods, ggplot2, dplyr, tidyr, stringr, purrr, psych, MASS, nFactors, qgraph, ppcor, ggcorrplot, rstanarm, MuMIn, lmerTest

**Suggests** tidyverse, knitr, rmarkdown, testthat, covr, lme4, broom

**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** 2018-03-22 08:25:35 UTC

**R topics documented:**

analyze	2
analyze.glmerMod	3
analyze.merMod	4
analyze.merModLmerTest	4
analyze.stanreg	5
assess	6
correlation	7
crawford.test	8
dprime	9
find_season	10
format_digit	11
format_p	11
format_string	12
get_predicted	12
get_predicted.stanreg	13
hdi	14
interpret_d	14
is.psychobject	15
mellenbergh.test	15
n_factors	16
odds_to_probs	17
plot.psychobject	17
print.psychobject	18
standardize	18
summary.psychobject	19
values	20
<b>Index</b>	<b>21</b>

---

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

---

**Description**

Analyze objects.

**Usage**

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

**Arguments**

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

**Author(s)**

Dominique Makowski

---

*analyze.glmMod*      *Analyze glmMod objects.*

---

**Description**

Analyze glmMod objects.

**Usage**

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

**Arguments**

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

**Value**

output

**Author(s)**

Dominique Makowski

**Examples**

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

analyze.merMod            *Analyze merMod objects.*

---

**Description**

Analyze merMod objects.

**Usage**

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

**Arguments**

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

**Value**

output

**Author(s)**

**Dominique Makowski**

**Examples**

```
library(psycho)  
require(lme4)  
fit <- lme4::lmer(Sepal.Length ~ Sepal.Width + (1|Species), data=iris)  
  
results <- analyze(fit)  
summary(results)
```

---

analyze.merModLmerTest  
                          *Analyze merModLmerTest objects.*

---

**Description**

Analyze merModLmerTest objects.

**Usage**

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

**Arguments**

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

**Value**

output

**Author(s)**

Dominique Makowski

**Examples**

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

results <- analyze(fit)
summary(results)
```

---

analyze.stanreg      *Analyze stanreg objects.*

---

**Description**

Analyze stanreg objects.

**Usage**

```
## S3 method for class 'stanreg'
analyze(x, CI = 95, effsize = FALSE, verbose = T, ...)
```

**Arguments**

x A stanreg model.  
CI Credible interval bounds.  
effsize Compute Effect Sizes according to Cohen (1988)? Your outcome variable must be standardized.  
verbose Toggle warnings display.  
... Arguments passed to or from other methods.

**Value**

output

**Author(s)**

Dominique Makowski

**Examples**

```
## Not run:
library(psycho)
require(rstanarm)
fit <- rstanarm::stan_glm(vs ~ mpg * cyl, data=mtcars)

results <- analyze(fit)
summary(results)

data <- normalize(attitude)
fit <- rstanarm::stan_lm(rating ~ advance + privileges + learning + raises,
                        data=data, prior=R2(1))

results <- analyze(fit)
summary(results)
plot(results)
print(results)

## End(Not run)
```

---

assess

*Compare a score to a parent population.*

---

**Description**

Compare a given score to a parent population.

**Usage**

```
assess(score, mean = 0, sd = 1, linecolor = "#E91E63",
       fillcolor = "#2196F3", xlabel = "Score", verbose = T)
```

**Arguments**

score	The score.
mean	The general population's mean.
sd	The general population's standart deviation.
linecolor	The colour of the vertical line.
fillcolor	The colour of the density plot.
xlabel	The label for the x axis.
verbose	Print possible warnings.

**Value**

output

**Author(s)**

Dominique Makowski

**Examples**

```
rez <- assess(124, mean=100, sd=15)
```

---

correlation                      *Multiple Correlations.*

---

**Description**

Compute different kinds of correlation matrices.

**Usage**

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

**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 <a href="#">p.adjust</a> for details about why to use "holm" rather than "bonferroni").

**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-Howell (1998) 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(case, controls, verbose = T)
```

## Arguments

case	Single value.
controls	Vector of values.
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 case is different from the control group.

## Author(s)

Dan Mirman

## Examples

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

---

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_miss, n_fa, n_cr)
```

### Arguments

n_hit	Number of hits.
n_miss	Number of misses.
n_fa	Number of false alarms.
n_cr	Number of correct rejections.

### 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 Hautus (1995).

### Author(s)

Dominique Makowski

**Examples**

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

indices <- dprime(n_hit, n_miss, n_fa, n_cr)
```

---

find\_season

*Find season of dates.*

---

**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**

```
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	Treshold 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_predicted	<i>Compute predicted values from models.</i>
---------------	--

---

**Description**

Compute predicted values from models.

**Usage**

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

**Arguments**

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

**Author(s)**

[Dominique Makowski](#)

---

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, prob = 0.9, draws = 500,  
             newdf = FALSE, precision = 10, ...)
```

## Arguments

fit	A stanreg model.
prob	Probability of credible intervals (0.9 (default) will compute 5-95% CI).
draws	Precision of the estimate.
newdf	Should the predictions be based on actual data or, generate a new dataframe based on all combinations of values
precision	Precision of the new dataframe to be generated.
...	Arguments passed to or from other methods.

## Value

dataframe with predicted values.

## Author(s)

[Dominique Makowski](#)

## Examples

```
## Not run:  
library(psycho)  
require(rstanarm)  
fit <- rstanarm::stan_glm(vs ~ mpg * cyl, data=mtcars)  
  
predicted <- get_predicted(fit)  
  
## End(Not run)
```

---

hdi	<i>Highest Density Intervals (HDI).</i>
-----	---

---

**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](#)

---

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)
```

---

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.

---

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, verbose = T)
```

**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.  
verbose              True or False. Prints the interpretation text.

**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**

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

---

n\_factors

*Find Optimal Factor Number.*

---

**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, log = TRUE)
```

**Arguments**

odds	Odds value(s).
log	Are these Log odds (such as in logistic models)?

**Author(s)**

**Dominique Makowski**

---

plot.psychobject	<i>Plot the results.</i>
------------------	--------------------------

---

**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**

---

```
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](#)

---

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

---

**Description**

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

**Usage**

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

**Arguments**

df                    Dataframe.  
except                Character or list of characters of column names to be excluded from normalization.

**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"))
```

---

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

analyze, [2](#)  
analyze.glmerMod, [3](#)  
analyze.merMod, [4](#)  
analyze.merModLmerTest, [4](#)  
analyze.stanreg, [5](#)  
assess, [6](#)  
  
correlation, [7](#)  
crawford.test, [8](#)  
  
dprime, [9](#)  
  
find\_season, [10](#)  
format\_digit, [11](#)  
format\_p, [11](#)  
format\_string, [12](#)  
  
get\_predicted, [12](#)  
get\_predicted.stanreg, [13](#)  
  
hdi, [14](#)  
  
interpret\_d, [14](#)  
is.psychobject, [15](#)  
  
mellenbergh.test, [15](#)  
  
n\_factors, [16](#)  
  
odds\_to\_probs, [17](#)  
  
p.adjust, [7](#)  
plot.psychobject, [17](#)  
print.psychobject, [18](#)  
  
standardize, [18](#)  
summary.psychobject, [19](#)  
  
values, [20](#)