

# Package ‘psycho’

January 19, 2021

**Type** Package

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

**Version** 0.6.1

**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** 7.1.1

**Depends** R (>= 3.5.0)

**Imports** stats, scales, utils, dplyr, stringr, ggplot2, insight, bayestestR, parameters, effectsize

**Suggests** knitr, rmarkdown, testthat, covr, 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** 2021-01-19 06:40:10 UTC

**R topics documented:**

affective . . . . .	2
assess . . . . .	3
crawford.test . . . . .	4
crawford.test.freq . . . . .	6
crawford_dissociation.test . . . . .	6
dprime . . . . .	7
emotion . . . . .	9
find_combinations . . . . .	10
find_combinations.formula . . . . .	10
find_matching_string . . . . .	11
find_season . . . . .	12
golden . . . . .	13
is.psychobject . . . . .	13
is.standardized . . . . .	14
mellenbergh.test . . . . .	14
percentile . . . . .	15
percentile_to_z . . . . .	16
plot.psychobject . . . . .	16
power_analysis . . . . .	17
print.psychobject . . . . .	18
remove_empty_cols . . . . .	18
summary.psychobject . . . . .	19
values . . . . .	19
<b>Index</b>	<b>20</b>

---

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

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	Alpha 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)
```

---

crawford.test

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

---

### Description

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

### Usage

```
crawford.test(
  patient,
  controls = NULL,
  mean = NULL,
  sd = NULL,
  n = NULL,
  CI = 95,
```

```

  treshold = 0.1,
  iter = 10000,
  color_controls = "#2196F3",
  color_CI = "#E91E63",
  color_score = "black",
  color_size = 2,
  alpha_controls = 1,
  alpha_CI = 0.8
)

```

### Arguments

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

### Details

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

### Author(s)

Dominique Makowski

### Examples

```

library(psycho)

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

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

```

---

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

---

### Description

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

### Usage

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

### Arguments

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

### Value

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

### Author(s)

Dan Mirman, Dominique Makowski

### Examples

```
library(psycho)

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

---

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

---

### Description

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

**Usage**

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

---

dprime

*Dprime ( $d'$ ) and Other Signal Detection Theory indices.*

---

**Description**

Computes Signal Detection Theory indices, including  $d'$ , beta,  $A'$ ,  $B''D$  and c.

**Usage**

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

**Arguments**

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

**Value**

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

Returns a list containing 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

*Emotional Ratings of Pictures*

---

**Description**

Emotional ratings of neutral and negative pictures by healthy participants.

**Usage**

```
emotion
```

**Format**

A data frame with 912 rows and 11 variables:

**Participant\_ID** Subject's number

**Participant\_Age** Subject's age

**Participant\_Sex** Subject's sex

**Item\_Category** Picture's category

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

---

`find_combinations`      *Generate all combinations.*

---

### **Description**

Generate all combinations.

### **Usage**

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

### **Arguments**

<code>object</code>	Object
<code>...</code>	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_season	<i>Find season of dates.</i>
-------------	------------------------------

---

**Description**

Returns the season of an array of dates.

**Usage**

```
find_season(
  dates,
  winter = "12-21",
  spring = "3-20",
  summer = "6-21",
  fall = "9-22"
)
```

**Arguments**

dates	Array of dates.
winter	month-day of winter solstice.
spring	month-day of spring equinox.
summer	month-day of summer solstice.
fall	month-day of fall equinox.

**Value**

season

**Author(s)**

Josh O'Brien

**See Also**

<https://stackoverflow.com/questions/9500114/find-which-season-a-particular-date-belongs-to>

**Examples**

```
library(psycho)

dates <- c("2012-02-15", "2017-05-15", "2009-08-15", "1912-11-15")
find_season(dates)
```

---

golden	<i>Golden Ratio.</i>
--------	----------------------

---

**Description**

Returns the golden ratio (1.618034...).

**Usage**

```
golden(x = 1)
```

**Arguments**

x	A number to be multiplied by the golden ratio. The default (x=1) returns the value of the golden ratio.
---	---

**Author(s)**

**Dominique Makowski**

**Examples**

```
library(psycho)
golden()
golden(8)
```

---

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      *Check if a dataframe is standardized.*

---

**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)
library(effectsize)

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

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

---

mellenbergh.test      *Mellenbergh & van den Brink (1998) test for pre-post comparison.*

---

**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)
```

---

percentile	<i>Transform z score to percentile.</i>
------------	---

---

**Description**

Transform z score to percentile.

**Usage**

```
percentile(z_score)
```

**Arguments**

z_score	Z score.
---------	----------

**Author(s)**

Dominique Makowski

**Examples**

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

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

---

**Description**

Transform a percentile to a z score.

**Usage**

```
percentile_to_z(percentile)
```

**Arguments**

percentile      Percentile

**Author(s)**

**Dominique Makowski**

**Examples**

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

---

plot.psychobject      *Plot the results.*

---

**Description**

Plot the results.

**Usage**

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

**Arguments**

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

**Author(s)**

**Dominique Makowski**



---

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

---

### Description

Compute the n models based on n sampling of data.

### Usage

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

### 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	Confidence level.

### 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**

---

remove\_empty\_cols      *Remove empty columns.*

---

### Description

Removes all columns containing any NaNs.

### Usage

```
remove_empty_cols(df)
```

**Arguments**

df                    Dataframe.

**Author(s)**

**Dominique Makowski**

---

summary.psychobject    *Print the results.*

---

**Description**

Print the results.

**Usage**

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

**Arguments**

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

**Author(s)**

**Dominique Makowski**

---

values                *Extract values as list.*

---

**Description**

Extract values as list.

**Usage**

```
values(x)
```

**Arguments**

x                     A psychobject class object.

**Author(s)**

**Dominique Makowski**

# Index

## \* datasets

    affective, [2](#)

    emotion, [9](#)

affective, [2](#)

assess, [3](#)

crawford.test, [4](#)

crawford.test.freq, [6](#)

crawford\_dissociation.test, [6](#)

dprime, [7](#)

emotion, [9](#)

find\_combinations, [10](#)

find\_combinations.formula, [10](#)

find\_matching\_string, [11](#)

find\_season, [12](#)

golden, [13](#)

is.psychobject, [13](#)

is.standardized, [14](#)

mellenbergh.test, [14](#)

percentile, [15](#)

percentile\_to\_z, [16](#)

plot.psychobject, [16](#)

power\_analysis, [17](#)

print.psychobject, [18](#)

remove\_empty\_cols, [18](#)

summary.psychobject, [19](#)

values, [19](#)