# Package ‘Keng’

May 7, 2026

**Title** Knock Errors Off Nice Guesses

**Version** 2026.3.19

**Description** Miscellaneous functions and data used in psychological research and teaching. Keng currently has a built-in dataset depress, and could (1) scale a vector;
(2) divide a vector into three groups,
(3) compute the cut-off values of Pearson's r with known sample size;
(4) test the significance and compute the post-hoc power for Pearson's r with known sample size;
(5) conduct a priori power analysis and plan the sample size for Pearson's r;
(6) compare lm()'s fitted outputs using R-squared, f_squared, post-hoc power,
and PRE (Proportional Reduction in Error, also called partial R-squared or
partial Eta-squared); (7) calculate PRE from partial correlation, Cohen's f, or f_squared;
(8) conduct a priori power analysis and plan the sample size for one or a set of predictors in regression analysis; (9) conduct post-hoc power analysis for one or a set of predictors in regression analysis with known sample size; (10) randomly pick numbers for Chinese Super Lotto and Double Color Balls; (11) assess course objective achievement in Outcome-Based Education.

**License** CC BY 4.0

**Language** en-US

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** stats

**Suggests** ggplot2, knitr, rmarkdown, car, effectsize, tidyr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** https://github.com/qyaozh/Keng

**BugReports** https://github.com/qyaozh/Keng/issues

**Depends** R (>= 3.5)

**LazyData** true

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Qingyao Zhang [aut, cre] (ORCID: <https://orcid.org/0000-0002-6891-5982>)

**Maintainer** Qingyao Zhang <qingyaozhang@outlook.com>

**Repository** CRAN

**Date/Publication** 2026-03-19 14:10:12 UTC

# Contents

assess_coa . . . . . 2
calc_PRE . . . . . 3
compare_lm . . . . . 4
cut_r . . . . . 6
depress . . . . . 7
divide . . . . . 8
pick_dcb . . . . . 8
pick_sl . . . . . 9
plot.Keng_power . . . . . 10
powered_lm . . . . . 11
powered_r . . . . . 11
power_lm . . . . . 12
power_r . . . . . 14
print.Keng_power . . . . . 15
Scale . . . . . 16
test_r . . . . . 17

**Index** **19**

---

`assess_coa` *Assess course objective achievement*

---

## Description

Assess course objective achievement

## Usage

```
assess_coa(data, session_weights, objective_weights1, ...)
```

## Arguments

`data` A wide-format data.frame that only contains student's grades of each session. `assess_coa` assumes the highest score for each session and the final grade is 100, given that weights are assigned by users.

`session_weights` A vector that Weights sessions for the final grade. The length of `session_weights` is the number of sessions. Typically, the final grade is based on the regular grade and final exam grade, in which condition the length of `session_weights` is 2. The range of each weight should be 0-1. The sum of `session_weights` should be 1.

`objective_weights1`
: A vector that Weights course objectives for session 1. The length of objective_weights1 is the number of course objectives. The range of each weight should be 0-1. The sum of `objective_weights1` should be 1.

`...`
: objective_weights2, objective_weights3, ... Other vectors that Weight course objectives for session1, session2, ... The number of objective_weights* arguments should be equal to the length of `session_weights`.

## Value

A data.frame containing grades of each session, final grades, and achievements of each objective. This data.frame also has an attribute named "weights" that contains a list of session_weights, objective_weights_matrix, and weighted_objective_weights_matrix

## Examples

```
data <- data.frame(
  session1 = 60 + sample.int(40, 100, 1),
  session2 = 60 + sample.int(40, 100, 1),
  session3 = 60 + sample.int(40, 100, 1)
)
session_weights    <- c(0.2, 0.3, 0.5)
objective_weights1 <- c(0.1, 0.4, 0.5)
objective_weights2 <- c(0.2, 0.2, 0.6)
objective_weights2 <- c(0.3,   0, 0.7)
coa <- assess_coa(
  data,
  session_weights,
  objective_weights1,
  objective_weights2,
  objective_weights2
)
head(coa)
attr(coa, "weights")
colMeans(coa[row.names(attr(coa, "weights")[[2]])])
```

---

`calc_PRE` *Calculate PRE from Cohen's f, f_squared, or partial correlation*

---

## Description

Calculate PRE from Cohen's f, f_squared, or partial correlation

## Usage

```
calc_PRE(f = NULL, f_squared = NULL, r_p = NULL)
```

### Arguments

| | |
|---|---|
| `f` | Cohen's f. Cohen (1988) suggested >=0.1, >=0.25, and >=0.40 as cut-off values of f for small, medium, and large effect sizes, respectively. |
| `f_squared` | Cohen's f_squared. Cohen (1988) suggested >=0.02, >=0.15, and >=0.35 as cut-off values of f for small, medium, and large effect sizes, respectively. |
| `r_p` | Partial correlation. |

### Value

A list including PRE, the absolute value of r_p (partial correlation), Cohen's f_squared, and f.

### References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge.

### Examples

```
calc_PRE(f = 0.1)
calc_PRE(f_squared = 0.02)
calc_PRE(r_p = 0.2)
```

---

## `compare_lm` *Compare lm()'s fitted outputs using PRE and R-squared.*

---

### Description

Compare lm()'s fitted outputs using PRE and R-squared.

### Usage

```
compare_lm(
  fitC = NULL,
  fitA = NULL,
  n = NULL,
  PC = NULL,
  PA = NULL,
  SSEC = NULL,
  SSEA = NULL
)
```

### Arguments

| | |
|---|---|
| `fitC` | The result of `lm()` of the Compact model (model C). |
| `fitA` | The result of `lm()` of the Augmented model (model A). |
| `n` | Sample size of the model C or model A. Model C and model A must use the same sample, and hence have the same sample size. Non-integer `n` would be converted to be an integer using `as.integer()`. |

| | |
|---|---|
| PC | The number of parameters in model C. Non-integer PC would be converted to be an integer using as.integer(). |
| PA | The number of parameters in model A. Non-integer PA would be converted to be an integer using as.integer(). as.integer(PA) should be larger than as.integer(PC). |
| SSEC | The Sum of Squared Errors (SSE) of model C. |
| SSEA | The Sum of Squared Errors of model A. |

## Details

compare_lm() compares model A with model C using PRE (Proportional Reduction in Error) , R-squared, f_squared, and post-hoc power. PRE is partial R-squared (called partial Eta-squared in Anova). There are two ways of using compare_lm(). The 1st is giving compare_lm() fitC and fitA. The 2nd is giving n, PC, PA, SSEC, and SSEA. The 1st way is more convenient, and it minimizes precision loss by omitting copying-and-pasting. Note that the F-tests for PRE and that for R-squared change are equivalent. Please refer to Judd et al. (2017) for more details about PRE, and refer to Aberson (2019) for more details about f_squared and post-hoc power.

## Value

A matrix with 12 rows and 4 columns. The 1st column reports information for the baseline model (intercept-only model). the 2nd for model C, the third for model A, and the fourth for the change (model A vs. model C). SSE (Sum of Squared Errors), sample size n, df of SSE, and the number of parameters for baseline model, model C, model A, and change (model A vs. model C) are reported in rows 1-3. The information in the 4th column are all for the change; put differently, these results could quantify the effect of one or a set of new parameters model A has but model C doesn't. If fitC and fitA are not inferior to the intercept-only model, R-squared, Adjusted R-squared, PRE, PRE_adjusted, and f_squared for the full model (compared with the baseline model) are reported for model C and model A. If model C or model A has at least one predictor, F-test with p, and post-hoc power would be computed for the corresponding full model.

## References

Aberson, C. L. (2019). *Applied power analysis for the behavioral sciences*. Routledge.

Judd, C. M., McClelland, G. H., & Ryan, C. S. (2017). *Data analysis: A model Comparison approach to regression, ANOVA, and beyond*. Routledge.

## Examples

```
x1 <- rnorm(193)
x2 <- rnorm(193)
y <- 0.3 + 0.2*x1 + 0.1*x2 + rnorm(193)
dat <- data.frame(y, x1, x2)
# Fix the intercept to constant 1 using I().
fit1 <- lm(I(y - 1) ~ 0, dat)
# Free the intercept.
fit2 <- lm(y ~ 1, dat)
compare_lm(fit1, fit2)
# One predictor.
```

```
fit3 <- lm(y ~ x1, dat)
compare_lm(fit2, fit3)
# Fix the intercept to 0.3 using offset().
intercept <- rep(0.3, 193)
fit4 <- lm(y ~ 0 + x1 + offset(intercept), dat)
compare_lm(fit4, fit3)
# Two predictors.
fit5 <- lm(y ~ x1 + x2, dat)
compare_lm(fit2, fit5)
compare_lm(fit3, fit5)
# Fix the slope of x2 to 0.05 using offset().
fit6 <- lm(y ~ x1 + offset(0.05*x2), dat)
compare_lm(fit6, fit5)
```

---

`cut_r` *Cut-off values of Pearson's correlation r with known sample size n.*

---

## Description

Cut-off values of Pearson's correlation r with known sample size n.

## Usage

```
cut_r(n)
```

## Arguments

| | |
|---|---|
| `n` | Sample size of Pearson's correlation r. `n` should be larger than |

## Details

Given n and p, t and then r could be determined. The formula used could be found in `test_r()`'s documentation.

## Value

A data.frame including the cut-off values of r at the significance levels of p = 0.1, 0.05, 0.01, 0.001. r with the absolute value larger than the cut-off value is significant at the corresponding significance level.

## Examples

```
cut_r(193)
```

`depress` *Depression and Coping*

## Description

A subset of data from research about depression and coping.

## Usage

```
depress
```

## Format

`depress`**:**

A data frame with 94 rows and 237 columns:

**id** Participant id

**class** Class

**grade** Grade

**elite** Elite classes

**intervention** 0 = Control group, 1 = Intervention group

**gender** 0 = girl, 1 = boy

**age** Age in year

**cope1i1p** Cope scale, Time1, Item1, Problem-focused coping, 1 = very seldom, 5 = very often

**cope1i3a** Cope scale, Time1, Item3, Avoidance coping

**cope1i5e** cope scale, Time1, Item5, Emotion-focused coping

**cope2i1p** Cope scale, Time2, Item1, Problem-focused coping

**depr1i1** Depression scale, Time1, Item1, 1 = very seldom, 4 = always

**ecr1avo** ECR-RS scale, Item1, attachment avoidance, 1 = very disagree, 7 = very agree

**ecr2anx** ECR-RS scale, Item2, attachment anxiety

**dm1** Depression, Mean, Time1

**pm1** Problem-focused coping, Mean, Time1

**em1** Emotion-focused coping, Mean, Time1

**am1** Avoidance coping, Mean, Time1

**avo** Attachment avoidance, Mean

**anx** Attachment anxiety, Mean

## Source

Keng package.

## divide — *Divide a vector into three groups*

### Description

Divide a vector into three groups

### Usage

```
divide(x, criterion = "1sd")
```

### Arguments

| | |
|---|---|
| x | A numeric vector. |
| criterion | A character value "1sd" or a numeric value between 0 and 0.5. |

### Details

divide() divides a vector into three groups using the criterion of 1 SD, or proportions like 0.27. Differing from cut(), the medium interval of divide() does not include the left and right limits.

### Value

An ordered factor with levels "Low < Medium < High".

### Examples

```
x <- 1:100
divide(x)
divide(x, 0.27)
```

## pick_dcb — *Pick Double Color Balls*

### Description

Pick Double Color Balls

### Usage

```
pick_dcb(size = 1L, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| size | The size of sets of Super Lotto numbers to pick. |
| verbose | A logical value. Print the numbers picked or not. |

### Value

Print the numbers picked, and return the invisible `balls` list that stored these numbers.

### Examples

```
pick_dcb(10)
out <- pick_dcb(10, verbose = FALSE)
out
```

---

## pick_sl *Pick Super Lotto numbers*

---

### Description

Pick Super Lotto numbers

### Usage

```
pick_sl(size = 1L, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `size` | An integer. The size of sets of Super Lotto numbers to pick. |
| `verbose` | A logical value. Print the numbers picked or not. |

### Value

Print the numbers picked, and return the invisible `balls` list that stored these numbers.

### Examples

```
# Example 1
pick_sl(10)
# Example 2
out <- pick_sl(10, verbose = FALSE)
out
# Example 3
# create an empty list
balls <- list(c(front = rep(NA, 5),
                back = rep(NA, 2))
)
luck <- list(c(front = c(10L, 13L, 14L, 19L, 27L),
               back = c(6L, 10L)))
# limit the max number of draws
max <- 9999
# try
# count the number of draws
i <- 0
while (!identical(balls, luck, max)) {
```

```
  i = i + 1
  balls = pick_sl(verbose = FALSE)
  if (identical(balls, luck)) {
    print(i)
    print(balls)
  }
  else
    if (i == max) {
      cat(i, "failed\n")
      break}
}
```

---

`plot.Keng_power` *Plot the power against the sample size for the Keng_power class*

---

## Description

Plot the power against the sample size for the Keng_power class

## Usage

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

## Arguments

| | |
|---|---|
| `x` | The output object of `power_r()` or `power_lm()`. |
| `...` | Further arguments passed to or from other methods. |

## Value

A plot of power against sample size.

## Examples

```
plot(power_lm())
out <- power_r(0.2, n = 193)
plot(out)
```

## powered_lm — *Compute lm's post-hoc power*

### Description

Compute lm's post-hoc power

### Usage

```
powered_lm(PRE = 0.04, PC = 1L, PA = 2L, n = 200L, sig_level = 0.05)
```

### Arguments

| | |
|---|---|
| PRE | Proportional Reduction in Error. PRE = The square of partial correlation. Cohen (1988) suggested >=0.02, >=0.13, and >=0.26 as cut-off values of PRE for small, medium, and large effect sizes, respectively. |
| PC | Number of parameters of model C (compact model) without focal predictors of interest. Non-integer PC would be converted to be an integer using `as.integer()`. |
| PA | Number of parameters of model A (augmented model) with focal predictors of interest. Non-integer PA would be converted to be an integer using `as.integer()`. `as.integer(PA)` should be larger than `as.integer(PC)`. |
| n | The current sample size. Integer n should be at least PA + 1L. |
| sig_level | Expected significance level for effects of focal predictors. |

### Value

Integer n, the F_test of PRE at the sample size n with df_A_C, df_A (the df of the model A at the sample size n), F (the F-test of PRE at the sample size n), p (the p-value of F-test at the sample size n), and the post-hoc power analysis with lambda (the non-centrality parameter of F at the sample size n), and power (the post-hoc power at the sample size n).

### Examples

```
powered_lm()
```

## powered_r — *Compute r's post-hoc power*

### Description

Compute r's post-hoc power

### Usage

```
powered_r(r = 0.2, n = 200L, sig_level = 0.05)
```

### Arguments

| | |
|---|---|
| `r` | Pearson's correlation. Cohen(1988) suggested >=0.1, >=0.3, and >=0.5 as cut-off values of Pearson's correlation `r` for small, medium, and large effect sizes, respectively. |
| `n` | The current sample size. Non-integer `n` would be converted to be an integer using `as.integer()`. `n` should be at least 3. |
| `sig_level` | Expected significance level. |

### Value

Integer `n`, the t_test of `r` at the sample size `n` with `df`, `SE` of `r`, `p` (the p-value of t-test), and the post-hoc power analysis with `delta` (the non-centrality parameter of the t-distribution for the alternative hypothesis), and `power` (the post-hoc power of `r` at the sample size `n`).

### References

Aberson, C. L. (2019). *Applied power analysis for the behavioral sciences*. Routledge.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge.

### Examples

```
powered_r()
```

---

## power_lm — *Conduct a priori power analysis, and plan the sample size for regression analysis*

---

### Description

Conduct a priori power analysis, and plan the sample size for regression analysis

### Usage

```
power_lm(
  PRE = 0.02,
  PC = 1L,
  PA = 2L,
  sig_level = 0.05,
  power = 0.8,
  n_ul = 1450000000L
)
```

### Arguments

| | |
|---|---|
| PRE | Proportional Reduction in Error. PRE = The square of partial correlation. Cohen (1988) suggested >=0.02, >=0.13, and >=0.26 as cut-off values of PRE for small, medium, and large effect sizes, respectively. |
| PC | Number of parameters of model C (compact model) without focal predictors of interest. Non-integer PC would be converted to be an integer using `as.integer()`. |
| PA | Number of parameters of model A (augmented model) with focal predictors of interest. Non-integer PA would be converted to be an integer using `as.integer()`. `as.integer(PA)` should be larger than `as.integer(PC)`. |
| sig_level | Expected significance level for effects of focal predictors. |
| power | Expected statistical power for effects of focal predictors. |
| n_ul | The upper limit of the sample size below which the minimum required sample size is searched for. Non-integer n_ul would be converted to be an integer using `as.integer()`. `as.integer(n_ul)` should be at least `as.integer(PA) + 1`. |

### Details

n_ul determines the upper limit of the interval below which power_lm() searches for the minimum required sample size, hence the number of rows of the returned power table `priori` and the right limit of the horizontal axis of the returned power plot. When PRE is very small (e.g., less than 0.001) and power is larger than 0.8, a huge increase in sample size only brings about a trivial increase in power, which is cost-ineffective. To make `power_lm()` omit unnecessary searching, you could set n_ul to be a value less than 1.45e+09 (e.g., 10000).

### Value

A Keng_power class, also a list. If sample size n is not given, the following results would be returned: `[[1]]` PRE; `[[2]]` `f_squared`, Cohen's f_squared derived from PRE; `[[3]]` PC; `[[4]]` PA; `[[5]]` `sig_level`, expected significance level for effects of focal predictors; `[[6]]` `power`, expected statistical power for effects of focal predictors; `[[7]]` n_ul, the upper limit of sample size; `[[8]]` `root`, the exact decimal n found by internal `uniroot()` to reach the expected power; `[[9]]` `minimum`, the minimum sample size required for focal predictors to reach the expected statistical power and significance level; `[[10]]` `priori`, a priori power table with increasing sample sizes (n_i), and corresponding `df_A_C`(the df of the numerator of the F-test, i.e., the difference of the dfs between model C and model A), `df_A_i`(the df of the denominator of the F-test, i.e., the df of the model A at the sample size n_i), `F_i`(the *F*-test of PRE at the sample size n_i), `p_i`(the p-value of `F_i`), `lambda_i`(the non-centrality parameter of the F-distribution for the alternative hypothesis, given PRE and n_i), `power_i`(the actual power of PRE at the sample size n_i).

By default, `print()` prints the primary but not all contents of the `Keng_power` class. To inspect more contents, use `print.AsIs()` or list extracting.

### References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge.

## Examples

```
power_lm()
print(power_lm())
plot(power_lm())
```

---

power_r *Conduct a priori power analysis, and plan the sample size for r.*

---

## Description

Conduct a priori power analysis, and plan the sample size for r.

## Usage

```
power_r(r = 0.2, sig_level = 0.05, power = 0.8, n_ul = 1450000000L)
```

## Arguments

| | |
|---|---|
| r | Pearson's correlation. Cohen(1988) suggested >=0.1, >=0.3, and >=0.5 as cut-off values of Pearson's correlation r for small, medium, and large effect sizes, respectively. |
| sig_level | Expected significance level. |
| power | Expected statistical power. |
| n_ul | The upper limit of the sample size below which the minimum required sample size is searched for. Non-integer n_ul would be converted to be an integer using as.integer(). n_ul should be at least 3L. |

## Details

Power_r() follows Aberson (2019) approach to conduct power analysis. n_ul determines the upper limit of the interval below which power_r() searches for the minimum required sample size, hence the number of rows of the returned power table priori and the right limit of the horizontal axis of the returned power plot. When r is very small and power is larger than 0.8, a huge increase of sample size only brings about a trivial increase in power, which is cost-ineffective. To make power_r() omit unnecessary searching, you could set n_ul to be a value less than 1.45e+09 (e.g., 10000).

## Value

A Keng_power class, also a list. If n is not given, the following results would be returned: [[1]] r, the given r; [[2]] d, Cohen's d derived from r; Cohen (1988) suggested >=0.2, >=0.5, and >=0.8 as cut-off values of d for small, medium, and large effect sizes, respectively; [[3]] sig_level, the expected significance level; [[4]] power, the expected power; [[5]] n_ul, the upper limit of sample size; [[6]] root, the exact decimal n found by internal uniroot() to reach the expected power; [[7]] minimum, the minimum required sample size. [[8]] priori, a priori power table with increasing sample sizes (n_i), and corresponding df_i (the df of t-test at the sample size n_i,

df_i = n_i - 2), SE_i (the SE of r at the sample size n_i), t_i (the t-test of r), p_i (the p-value of t_i), delta_i (the non-centrality parameter of the t-distribution for the alternative hypothesis, given r and n_i), power_i (the actual power of r at the sample size n_i);.

If sample size n is given, the following results would also be returned: Integer n, the t_test of r at the sample size n with df, SE of r, p (the p-value of t-test), and the post-hoc power analysis with delta_post (the non-centrality parameter of the t-distribution for the alternative hypothesis), and power_post (the post-hoc power of r at the sample size n).

By default, print() prints the primary but not all contents of the Keng_power class. To inspect more contents, use print.AsIs() or list extracting.

## References

Aberson, C. L. (2019). *Applied power analysis for the behavioral sciences*. Routledge.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge.

## Examples

```
power_r(0.2)
print(power_r(0.04))
plot(power_r(0.04))
```

---

# print.Keng_power *Print primary but not all contents of the Keng_power class*

---

## Description

Print primary but not all contents of the Keng_power class

## Usage

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

## Arguments

| | |
|---|---|
| x | The output object of power_r() or power_lm(). |
| ... | Further arguments passed to or from other methods. |

## Value

None (invisible NULL).

## Examples

```
power_lm()
power_lm(n = 200)
print(power_lm(n = 200))
x <- power_r(0.2, n = 193)
x
```

`Scale` *Scale a vector*

## Description

Scale a vector

## Usage

```
Scale(x, m = 0, sd = NULL)
```

## Arguments

| | |
|---|---|
| `x` | The original vector. |
| `m` | The expected Mean of the scaled vector. |
| `sd` | The expected Standard Deviation (unit) of the scaled vector. |

## Details

To scale `x`, its mean, or standard deviation (unit), or both, could be changed. The missing values of `x` are removed if any.

If `m` = 0 (default), and `sd` = `NULL`(default), `x` would be mean-centered (default).

If `m` = `NULL`, the mean of `x` would not changed. If `m` is not `NULL`, the mean of `x` would be transformed to `sd`.

If `m` = `NULL` or 0, and `sd` = 1, `x` would be standardized to be its z-score with m = 0 and sd = 1. The mean and standardized score are not necessarily 0 and 1.

## Value

The scaled vector.

## Examples

```
(x <- rnorm(10, 5, 2))
# Mean-center x.
Scale(x)
Scale(x, m = 0)
# Transform the mean of x to 3.
Scale(x, m = 3)
# Transform x to its z-score.
Scale(x, sd = 1)
# Standardize x with m = 100 and sd = 15.
Scale(x, m = 100, sd = 15)
```

`test_r` *Test the significance, analyze the power, and plan the sample size for r.*

## Description

Test the significance, analyze the power, and plan the sample size for r.

## Usage

```
test_r(r = NULL, n = NULL, sig_level = 0.05, power = 0.8)
```

## Arguments

| | |
|---|---|
| `r` | Pearson's correlation. Cohen(1988) suggested >=0.1, >=0.3, and >=0.5 as cut-off values of Pearson's correlation r for small, medium, and large effect sizes, respectively. |
| `n` | Sample size of r. Non-integer `n` would be converted to be a integer using `as.integer()`. `n` should be at least 3. |
| `sig_level` | Expected significance level. |
| `power` | Expected statistical power. |

## Details

To test the significance of the r using the one-sample t-test, the SE of `r` is determined by the following formula: `SE = sqrt((1 - r^2)/(n - 2))`. Another way is transforming `r` to Fisher's z using the following formula: `fz = atanh(r)` with the SE of `fz` being `sqrt(n - 3)`. Fisher's z is commonly used to compare two Pearson's correlations from independent samples. Fisher's transformation is presented here only to satisfy the curiosity of users who are interested in the difference between t-test and Fisher's transformation.

The post-hoc power of `r`'s t-test is computed through the way of Aberson (2019). Other software and R packages like SPSS and `pwr` give different power estimates due to underlying different formulas. `Keng` adopts Aberson's approach because this approach guarantees the equivalence of r and PRE.

## Value

A list with the following results: `[[1]] r`, the given r; `[[2]] d`, Cohen's d derived from `r`; Cohen (1988) suggested >=0.2, >=0.5, and >=0.8 as cut-off values of `d` for small, medium, and large effect sizes, respectively. `[[3]]` Integer `n`; `[[4]]` t-test of `r` (incl., `r`, `df` of r, `SE_r`, `t`, `p_r`), 95% CI of `r` based on t -test (`LLCI_r_t`, `ULCI_r_t`), and post-hoc power of `r` (incl., `delta_post`, `power_post`); `[[5]]` Fisher's z transformation (incl., `fz` of `r`, z-test of `fz` [`SE_fz`, `z`, `p_fz`], and 95% CI of `r` derived from `fz`.

Note that the returned CI of `r` may be out of r's valid range [-1, 1]. This "error" is deliberately left to users, who should correct the CI manually in reports.

## References

Aberson, C. L. (2019). *Applied power analysis for the behavioral sciences*. Routledge.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge.

## Examples

```
test_r(0.2, 193)

# compare the p-values of t-test and Fisher's transformation
for (i in seq(30, 200, 10)) {
cat(c("n = ", i, ", difference between ps = ",
       format(
        abs(test_r(0.2, i)[["t_test"]]["p_r"] - test_r(0.2, i)[["Fisher_z"]]["p_fz"]),
        nsmall = 12,
        scientific = FALSE)),
      sep = "",
      fill = TRUE)
}
```

# Index


∗ **datasets**
    depress, 7

assess_coa, 2

calc_PRE, 3
compare_lm, 4
cut_r, 6

depress, 7
divide, 8

pick_dcb, 8
pick_sl, 9
plot.Keng_power, 10
power_lm, 12
power_r, 14
powered_lm, 11
powered_r, 11
print.Keng_power, 15

Scale, 16

test_r, 17