Package 'Keng'

November 17, 2024

Title Knock Errors Off Nice Guesses

Version 2024.11.17

Description Miscellaneous functions and data used in Qingyao's psychological research and teaching. Keng currently has a built-in dataset depress, and could (1) scale a vector, (2) test the significance and compute the cut-off values of Pearson's r without raw data, (3) compare lm()'s fitted outputs using R-squared and PRE (Proportional Reduction in Error, also called partial R-squared or partial Eta-squared).

License CC BY 4.0

Encoding UTF-8

RoxygenNote 7.3.2

Imports stats

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

Config/testthat/edition 3

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

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

Depends R (>= 2.10)

LazyData true

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer Qingyao Zhang <qingyaozhang@outlook.com>

Repository CRAN

Date/Publication 2024-11-17 12:50:02 UTC

Contents

---

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.
`PC`	The number of parameters in Model C.
`PA`	The number of parameters in Model A. PA must be larger than PC.
`SSEC`	The Sum of Squared Errors (SSE) of Model C.
`SSEA`	The Sum of Squared Errors of Model A.

Details

`compare_lm()` compare Model A with Model C using *PRE* (Proportional Reduction in Error) and R-squared. *PRE* is partial R-squared (called partial Eta-squared in Anova).There are two ways of using `compare_lm()`. The first is giving `compare_lm()` fitC and fitA. The second is giving *n*, *PC*, *PA*, *SSEC*, and *SSEA*. The first way is more convenient, and it minimizes precision loss by omitting copying-and-pasting. If fitC and fitA are not inferior to the intercept-only model, R-squared and Adjusted R-squared are also computed. Note that the *F*-tests for *PRE* and R-squared change are equivalent. Please refer to Judd et al. (2017) for more details about *PRE*.

Value

A data.frame with 3 rows and 8 columns. The first row reports information for Model C, the second for Model A, and the third for the change. The data.frame presents *SSE*, *df* of *SSE*, *PRE*, the *F*-test of *PRE* (*F*, *p*), and *PRE_adjusted*. If fitC and fitA are not inferior to the intercept-only model, R-squared and Adjusted R-squared will also be computed.

References

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 intercept to constant 1 using I().
fit1 <- lm(I(y - 1) ~ 0, dat)
# Free intercept.
fit2 <- lm(y ~ 1, dat)
compare_lm(fit1, fit2)
# One predictor.
fit3 <- lm(y ~ x1, dat)
compare_lm(fit2, fit3)
# Fix 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 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 r given the sample size n.*

Description

Cut-off values of r given the sample size n.

Usage

```
cut_r(n)
```

Arguments

n	Sample size of the r.

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 a 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
intervene 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, 5 = 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.

Scale *Scale a vector*

Description

Scale a vector

Usage

```
Scale(x, expected_M = NULL, expected_SD = NULL, oadvances = NULL)
```

Arguments

x	The original vector.
expected_M	The expected Mean of the scaled vector.
expected_SD	The expected Standard Deviation (unit) of the scaled vector.
oadvances	The distance the Origin of x advances by.

Details

To scale x, its origin, or unit (*expected_SD*), or both, could be changed.

If expected_M = 0 or NULL, and expected_SD = NULL, x would be mean-centered.

If expected_M is a non-zero number, and expected_SD = NULL, the mean of x would be transformed to expected_M.

If expected_M = 0 or NULL, and expected_SD = 1, x would be standardized to be its z-score with $M = 0$ and $SD = 1$.

The standardized score is not necessarily the z-score. If neither expected_M nor expected_SD is NULL, x would be standardized to be a vector whose mean and standard deviation would be expected_M and expected_SD, respectively. To standardize x, the mean and standard deviation of x are needed and computed, for which the missing values of x are removed if any.

If oadvances is not NULL, the origin of x will advance with the standard deviation being unchanged. In this case, Scale() could be used to pick points in simple slope analysis for moderation models. Note that when oadvances is not NULL, expected_M and expected_SD must be NULL.

Value

The scaled vector.

Examples

```
(x <- rnorm(10, 5, 2))
# Mean-center x.
Scale(x)
# Transform the mean of x to 3.
Scale(x, expected_M = 3)
# Transform x to its z-score.
Scale(x, expected_SD = 1)
# Standardize x with *M* = 100 and *SD* = 15.
Scale(x, expected_M = 100, expected_SD = 15)
# The origin of x advances by 3.
Scale(x, oadvances = 3)
```

---

test_r *Test r using the t-test given r and n.*

Description

Test r using the t-test given r and n.

Usage

```
test_r(r, n)
```

Arguments

r	Pearson correlation.
n	Sample size of *r*.

Details

To test the significance of the *r* using one-sample *t*-test, the *SE* of the r is determined by the following formula: `SE = sqrt((1 - r^2)/(n - 2))`.

Value

A data.frame including *r*, *se* of *r*, *t*, and *p*.

Examples

```
test_r(0.2, 193)
```

Index