Package ‘alphaci’

February 5, 2024

Type Package

Title Confidence Intervals for Coefficient Alpha and Standardized Alpha

Version 1.0.1

Description Calculate confidence intervals for alpha and standardized alpha using asymptotic theory or the studentized bootstrap, with or without transformations. Supports the asymptotic distribution-free method of Maydeu-Olivares, et al. (2007) <doi:10.1037/1082-989X.12.2.157>, the pseudo-elliptical method of Yuan & Bentler (2002) <doi:10.1007/BF02294845>, and the normal method of van Zyl et al. (1999) <doi:10.1007/BF02296146>, for both coefficient alpha and standardized alpha.

License MIT + file LICENSE

URL https://jonasmoss.github.io/alphaci/

Depends R (>= 3.5.0)

Imports future.apply, matrixcalc

Suggests covr, extraDistr, knitr, lavaan, psychTools, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Jonas Moss [aut, cre] (<https://orcid.org/0000-0002-6876-6964>)

Maintainer Jonas Moss <jonas.moss.statistics@gmail.com>

Repository CRAN

Date/Publication 2024-02-05 12:20:09 UTC

R topics documented:

alphaci — *Confidence intervals for alpha and standardized alpha*

Description

Calculate confidence intervals for coefficient alpha (Cronbach, 1951) and standardized alpha (Falk & Savalei, 2011) using asymptotic methods or the studentized bootstrap. `alphaci` constructs confidence intervals for coefficient alpha and `alphaci_std` for standardized alpha.

Usage

```
alphaci(
  x,
  type = c("adf", "elliptical", "normal"),
  transform = "none",
  parallel = FALSE,
  conf_level = 0.95,
  alternative = c("two.sided", "greater", "less"),
  bootstrap = FALSE,
  n_reps = 1000
)

alphaci_std(
  x,
  type = c("adf", "elliptical", "normal"),
  transform = "none",
  parallel = FALSE,
  conf_level = 0.95,
  alternative = c("two.sided", "greater", "less"),
  bootstrap = FALSE,
  n_reps = 1000
)
```

Arguments

`x`	Input data data can be converted to a matrix using `as.matrix`. Rows containing missing values are ignored.
`type`	Type of confidence interval. Either `adf`, `elliptical`, or `normal`.
`transform`	One of `"none"`, `"log"`, `"fisher"`, and `"arcsin`. Defaults to `"none"`.
`parallel`	If `TRUE`, makes calculations under the assumption of a parallel model. Defaults to `FALSE`.
`conf_level`	Confidence level. Defaults to `0.95`.
`alternative`	A character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`.

bootstrap	If TRUE, performs a studentized bootstrap with n_reps repetitions. Defaults to FALSE.
n_reps	Number of bootstrap samples if bootstrap = TRUE. Ignored if bootstrap = FALSE. Defaults to 1000.

Details

The methods accept handle missing data using stats::na.omit, i.e., rows containing missing data are removed. The bootstrap option uses the studentized bootstrap (Efron, B. 1987), which is second order correct. Both functions makes use of future.apply when bootstrapping.

The type variables defaults to adf, asymptotically distribution-free, which is consistent when the fourth moment is finite (Maydeu-Olivares et al. 2007). The normal option assumes normality. (van Zyl et al. 1999), and is not concistent for models with excess kurtosis unequal to 0. The elliptical option assumes an elliptical or pseudo-elliptical distribution of the data. The resulting confidence intervals are corrected variants of the normal theory intervals with a kurtosis correction (Yuan & Bentler 2002). The common kurtosis parameter is calculated using the unbiased sample kurtosis (Joanes, 1998). All these methods have analogues for standardized alpha, which can be derived using the methods of Hayashi & Kamata (2005) and Neudecker (2006).

Value

A vector of class alphaci containing the confidence end points. The arguments of the function call are included as attributes.

References

Falk, C. F., & Savalei, V. (2011). The relationship between unstandardized and standardized alpha, true reliability, and the underlying measurement model. Journal of Personality Assessment, 93(5), 445-453. https://doi.org/10.1080/00223891.2011.594129

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. Psychometrika, 16(3), 297-334. https://doi.org/10.1007/BF02310555#'

Efron, B. (1987). Better Bootstrap Confidence Intervals. Journal of the American Statistical Association, 82(397), 171-185. https://doi.org/10.2307/2289144

Maydeu-Olivares, A., Coffman, D. L., & Hartmann, W. M. (2007). Asymptotically distribution-free (ADF) interval estimation of coefficient alpha. Psychological Methods, 12(2), 157-176. https://doi.org/10.1037/1082-989X.12.2.157

van Zyl, J. M., Neudecker, H., & Nel, D. G. (2000). On the distribution of the maximum likelihood estimator of Cronbach's alpha. Psychometrika, 65(3), 271-280. https://doi.org/10.1007/BF02296146

Yuan, K.-H., & Bentler, P. M. (2002). On robustness of the normal-theory based asymptotic distributions of three reliability coefficient estimates. Psychometrika, 67(2), 251-259. https://doi.org/10.1007/BF02294845

Joanes, D. N., & Gill, C. A. (1998). Comparing measures of sample skewness and kurtosis. Journal of the Royal Statistical Society: Series D (The Statistician), 47(1), 183-189. https://doi.org/10.1111/1467-9884.00122

Hayashi, K., & Kamata, A. (2005). A note on the estimator of the alpha coefficient for standardized variables under normality. Psychometrika, 70(3), 579-586. https://doi.org/10.1007/s11336-001-0888-1

Neudecker, H. (2006). On the Asymptotic Distribution of the Natural Estimator of Cronbach's Alpha with Standardised Variates under Nonnormality, Ellipticity and Normality. In P. Brown, S. Liu, & D. Sharma (Eds.), Contributions to Probability and Statistics: Applications and Challenges (pp. 167-171). World Scientific. https://doi.org/10.1142/9789812772466_0013

Examples

```
library("alphaci")
library("psychTools")
x <- bfi[, 1:5]
x[, 1] <- 7 - x[, 1] # Reverse-coded item.
alphaci(x)
alphaci_std(x)

# Calculate confidence intervals with other options.
library("lavaan")
x <- lavaan::HolzingerSwineford1939[1:20, 7:9]
results <- c(
  alphaci(x, type = "adf", parallel = FALSE),
  alphaci(x, type = "adf", parallel = TRUE),
  alphaci(x, type = "elliptical", parallel = FALSE),
  alphaci(x, type = "elliptical", parallel = TRUE),
  alphaci(x, type = "normal", parallel = FALSE),
  alphaci(x, type = "normal", parallel = TRUE)
)
```

Index