Package 'ezCutoffs'

October 13, 2022

Date 2019-11-09

Type Package

Title Fit Measure Cutoffs in SEM

Version 1.0.1

Depends R (>= 2.15.1)

Description Calculate cutoff values for model fit measures used in structural equation modeling (SEM) by simulating and testing data sets (cf. Hu & Bentler, 1999 <doi:10.1080/10705519909540118>) with the same parameters (population model, number of observations, etc.) as the model under consideration.

License GPL-3

Imports doSNOW, foreach, ggplot2, lavaan, moments, parallel, progress, stats, utils

Suggests boot

Maintainer Bjarne Schmalbach <bjarne.schmalbach@gmail.com>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Author Bjarne Schmalbach [aut, cre] (<https://orcid.org/0000-0001-6853-412X>), Julien Patrick Irmer [aut], Martin Schultze [aut]

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-04 16:00:05 UTC

R topics documented:

compareFit *Compare two Fit Measure Distributions Using the Wilcoxon-test*

Description

Significane test of the difference between two randomly generated fit index distributions using the Wilcoxon rank sum test.

Usage

```
compareFit(x, y, ...)
```

Arguments

x	An object of the class ezCutoffs to use in comparison.
y	A second ezCutoffs object to compare x to.
...	Additional arguments to pass to wilcox.test.

Details

Non-overlapping fit measures will be disregarded by the funciton.

Value

An object of the class wilc_result, inspectable via summary.

See Also

ezCutoffs

Examples

```
## model specification examples

# simple uni-factorial model
model <- "F1 =~ a1 + a2 + a3 + a4 + a5"

## two function calls
a <- ezCutoffs(model = model, n_obs = 1000, n_rep = 10, n_cores = 1, normality = "assumed")
b <- ezCutoffs(model = model, n_obs = 1000, n_rep = 10, n_cores = 1, normality = "empirical")

## comparison of the fit measure distributions yielded by the simulations
w <- compareFit(a, b)
summary(w)
```

ezCutoffs *Fit Measure Cutoffs in SEM*

Description

Calculate cutoff values for model fit measures used in structural equation modeling (SEM) by simulating and testing data sets (cf. Hu & Bentler, 1999 <doi:10.1080/10705519909540118>) with the same parameters (population model, number of observations, etc.) as the model under consideration.

Usage

```
ezCutoffs(model = NULL, data = NULL, n_obs = NULL, n_rep = 1000,
  fit_indices = c("chisq", "cfi", "tli", "rmsea", "srmr"),
  alpha_level = 0.05, normality = "assumed", missing_data = FALSE,
  bootstrapped_ci = FALSE, n_boot = 1000, boot_alpha = 0.05,
  boot_internal = FALSE, n_cores = NULL, ...)
```

Arguments

model	lavaan-style Syntax of a user-specified model.
data	A data frame containing the variables specified in model.
n_obs	Specifies the number of observations. Only needed if no data frame is given. Can be given as a numeric vector representing the exact group sizes in multigroup analyses. In this case, the grouping variable needs to be called "group".
n_rep	Number of replications.
fit_indices	Character vector, containing a selection of fit indices for which to calculate cutoff values. Only measures produced by fitMeasures can be chosen.
alpha_level	Type I-error rate for the generated cutoff values: Between 0 and 1; 0.05 per default.
normality	Specify distributional assumptions for the simulated data: Either "assumed" for normal distribution, or "empirical" for distributions based on the skewness and kurtosis values of the empirical data.
missing_data	Specify handling of missing values: Either FALSE to generate complete data sets, or TRUE to generate data with the same number of missing values on the observed variables as in the empirical data.
bootstrapped_ci	Specify whether a boostrapped confidence interval for the empirical model fit statistics should be drawn; default = FALSE.
n_boot	Number of replications in bootstrap for confidence intervalls for empirical model fit statistics.
boot_alpha	Type I-error rate choosen for the boostrap-confidence interval: Between 0 and 1; 0.05 per default.

boot_internal	Whether to use the internal boostrap implemented in bootstrapLavaan or a standard implementation in the boot package. Defaults to FALSE
n_cores	The number of cores to use. If NULL (the default) all available cores will be used.
...	Additional arguments to pass to lavaan.

Details

model is expected in standard lavaan nomenclature. The typical pre-multiplication mechanism is supported, with the exception of vectors (see Examples). Multigroup models should instead be specified using the group argument.

If data is not specified, the program will generate data based on the given model and n_obs. A numeric vector would signify multiple groups and group needs to be set to "group" in this case. Otherwise, n_obs is disregarded.

missing_data = TRUE assumes that the data is missing completely at random. That, is missings should not be distributed unevenly in multigroup models, for instance.

bootstrapped_ci = "TRUE" Returns a nonparametric bootstrap confidence interval that quantifies the uncertainty within a data set with regard to the empirical fit indices. Larger sample sizes should, under ideal circumstances, have smaller confidence intervals. For more information see, e.g., Efron (1981; 1987). Bootstrapping uses the library(boot) and (if available) several CPUs to compute the confidence intervals via snow.

... allows the user to pass lavaan arguments to the model fitting procedure. Options include multigroup, repeated measures, growth curve, and multilevel models.

Value

An object of the class ezCutoffs, inspectable via print, summary, plot, and compareFit

References

Efron, D. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods, Biometrika, 68(3), 589-599. doi: 10.1093/biomet/68.3.589

Efron, B. (1987). Better bootstrap confidence intervals. Journal of the American statistical Association, 82(397), 171-185.

Hu, L. T., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. Structural Equation Modeling: A Multidisciplinary Journal, 6(1), 1-55. doi: 10.1080/10705519909540118

See Also

compareFit

Examples

```
## model specification examples

# simple uni-factorial model
model1 <- "F1 =~ a1 + a2 + a3 + a4 + a5"

# path model
model2 <- "m ~ 0.6*x1
           m ~ 0.5*x2
   m ~ 0.4*x3
   y ~ 0.7*m"

# two-factorial model with some exemplary pre-multiplications
model3 <- "F1 =~ NA*a1 + a2 + a3 + 0.8*a4 + a5
           F2 =~ b1 + start(0.8)*b2 + b3 + equal('F2 =~ b2')*b4 + b5
           F1 ~~ 0*F2"

## function call
out <- ezCutoffs(model = model1, n_obs = 1000, n_rep = 10, n_cores = 1)

out <- ezCutoffs(
  model = model1, n_obs = c(300, 400), n_rep = 9999, fit_indices = c("cfi.robust"),
 estimator = "MLM", group = "group", group.equal = c("loadings", "intercepts"), n_cores = 1
)


## retrieve output
summary(out)
plot(out)
```

Index