

Package ‘FCO’

March 17, 2023

Title Flexible Cutoffs for Model Fit Evaluation in Covariance-Based Structural Models

Version 0.8.0

Depends R (>= 3.5.0)

Description

A toolbox to derive flexible cutoffs for fit indices in 'Covariance-based Structural Equation Modeling' based on the paper by 'Niemand & Mai (2018)' <[doi:10.1007/s11747-018-0602-9](https://doi.org/10.1007/s11747-018-0602-9)>. Flexible cutoffs are an alternative to fixed cutoffs - rules-of-thumb - regarding an appropriate cutoff for fit indices such as 'CFI' or 'SRMR'. It has been demonstrated that these flexible cutoffs perform better than fixed cutoffs in grey areas where misspecification is not easy to detect. The package provides an alternative to the tool at <<https://flexiblecutoffs.org>> as it allows to tailor flexible cutoffs to a given dataset and model, which is so far not available in the tool. The package simulates fit indices based on a given dataset and model and then estimates the flexible cutoffs. Some useful functions, e.g., to determine the 'GoF-' or 'BoF-nature' of a fit index, are provided. So far, additional options for a relative use (is a model better than another?) are provided in an exploratory manner.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports lavaan, semTools, parallel, checkmate, data.table

Suggests rmarkdown, knitr, ggplot2, MASS, testthat

VignetteBuilder knitr

Config/testthat/edition 3

Author Thomas Niemand [aut, cre] (<<https://orcid.org/0000-0001-7677-1168>>),
Robert Mai [ctb] (<<https://orcid.org/0000-0002-8364-9328>>)

Maintainer Thomas Niemand <thomas.niemand@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2023-03-17 11:30:15 UTC

R topics documented:

bb1992	2
flex_co	3
gen_fit	5
index_guess	8
pop_mod	9
recommend	10
recommend_dv	11

Index	13
--------------	-----------

bb1992	<i>Dataset from Babakus & Boller (1992)</i>
--------	---

Description

Data from Babakus & Boller (1992) who investigated the dimensionality of the SERVQUAL scale based on a sample of N = 502. The data is available as a data.frame (simulated via mvnorm in package MASS based on the correlation matrix provided by the authors) and used in the vignette.

Usage

```
data(bb1992)
```

Format

A data.frame of 22 variables (Q1-Q22) with 502 observations

Source

Provided in paper

References

Babakus, E., & Boller, G. W. (1992). An empirical assessment of the SERVQUAL scale. *Journal of Business Research*, 24(3), 253–268. [https://doi.org/10.1016/0148-2963\(92\)90022-4](https://doi.org/10.1016/0148-2963(92)90022-4)

Examples

```
data(bb1992)
head(bb1992, 3)
```

flex_co	<i>Obtain flexible cutoffs for one or two models</i>
---------	--

Description

Obtain flexible cutoffs for one or two models

Usage

```
flex_co(fits, index, alpha.lev = 0.05, gof = NULL)
```

Arguments

fits	A list of simulated fit indices obtained from <code>gen_fit</code> . Based on the structure of <code>fits</code> , the number of models is derived.
index	A vector of fit indices or measures provided by function <code>fitmeasures</code> in package <code>lavaan</code>
alpha.lev	The predefined uncertainty For example, if the default uncertainty of .05 (5 percent) is accepted a-priori, the 5 percent <code>stats::quantile</code> (of type 8, see <code>?stats::quantile</code>) of the simulated distribution for correctly specified CFA models with the given model and sample characteristics determines the flexible cutoff. Options are .001, .01, .05, and .10. Higher values are more conservative.
gof	An optional vector as to whether the indices are GoF (Goodness-of-Fit index)? If TRUE, a GoF is assumed. If FALSE, a BoF is assumed. Depending on the nature of the underlying fit index, the appropriate lower (GoF) or upper (BoF) width of the respective confidence interval as defined by the <code>stats::quantile</code> is used to derive the flexible cutoff. If not provided or not equal to the number of fit indices, the function guesses the type for known fit indices (e.g., SRMR is a BoF).

Value

A list of information regarding the selected fit index providing its flexible cutoff for the given parameters.

Examples

```
#Note: Demonstration only! Please use higher numbers of replications for your applications (>= 500).
#A single model to obtain fit indices for
mod <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
"
fits.single <- gen_fit(mod1 = mod, x = bb1992, rep = 10, standardized = FALSE)
```

```

flex_co(fits = fits.single, index = c("CFI", "SRMR"))

#Two models, an unconstrained and a constrained model to compare fit indices
mod.con <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
F1 ~~ 0 * F2
"

fits.con <- gen_fit(
  mod1 = mod,
  mod2 = mod.con,
  x = bb1992,
  rep = 10
)
flex_co(fits = fits.con,
  index = c("CFI", "SRMR"),
  alpha.lev = .05)

#Two models for discriminant validity testing, this resembles constraining with a cutoff of .9
fits.dv.con <- gen_fit(
  mod1 = mod,
  x = bb1992,
  rep = 10,
  dv = TRUE,
  dv.factors = c("F4", "F5"),
  dv.cutoff = .9
)
flex_co(fits = fits.dv.con,
  index = "CFI",
  alpha.lev = .05)
mod.dv.con <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
F4 ~~ .9 * F5
"

lavaan::fitmeasures(
  lavaan::cfa(
    model = mod.dv.con,
    data = bb1992,
    auto.fix.first = FALSE,
    std.lv = TRUE
  ),
  fit.measures = "cfi"
)
#Two models for discriminant validity testing, this resembles merging.
fits.dv.merge <- gen_fit(
  mod1 = mod,

```

```

x = bb1992,
rep = 10,
dv = TRUE,
dv.factors = c("F4", "F5"),
merge.mod = TRUE)

flex_co(fits = fits.dv.merge,
index = "CFI",
alpha.lev = .05)
mod.dv.merge <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17 + Q6 + Q14 + Q15 + Q16
"

lavaan::fitmeasures(
  lavaan::cfa(
    model = mod.dv.merge,
    data = bb1992
  ),
  fit.measures = "cfi"
)

```

gen_fit

Obtain fit statistics from one or two models

Description

Obtain fit statistics from one or two models

Usage

```

gen_fit(
  mod1 = NULL,
  mod2 = NULL,
  x = NULL,
  n = NULL,
  rep = 500,
  type = "NM",
  dv = FALSE,
  dv.factors = NULL,
  merge.mod = FALSE,
  dv.cutoff = 0.9,
  standardized = TRUE,
  assume.mvn = TRUE,
  multi.core = TRUE,
  cores = 2,
  seed = 1111,

```

```

    pop.mod1 = NULL,
    pop.mod2 = NULL
  )

```

Arguments

mod1	A lavaan model to specify the CFA.
mod2	Another lavaan model for a model comparison. If missing and merge.mod = TRUE, a merged model from function merge_factors is estimated based on mod1.
x	A dataset for the model of nrow observations (minimum: 50) and ncol indicators (minimum: 4)
n	A sample size specified instead of a dataset (minimum: 50, maximum: 50000). Requires a population model via pop.mod1.
rep	Number of replications to be simulated (default: 500, minimum: 10, maximum: 5000)
type	Type of underlying population model. Based on the model(s) provided, a population model is derived to simulate the fit indices by function pop_mod. The type determines the factor loadings and covariances assumed for this population model. NM (the default when only one model is provided): Uses the factor loadings and covariances from Niemand & Mai's (2018) simulation study. HB: Uses the factor loadings and covariances from Hu & Bentler's (1999) simulation study. EM: Empirical (the default when two models are provided or merge.mod is TRUE), uses the given factor loadings and covariances.
dv	Should the fit statistics be calculated for discriminant validity testing? If no (the default), this is not assumed. If yes, consider the arguments of merge.mod, dv.factors and cutoff. So far, two options of discriminant validity testing are supported. Constraining: A factor correlation between two factors can be constrained as selected by the dv.factors argument. In this case, dv.cutoff applies and merge.mod is not required. Merging: Two factors can be merged into one, again controlled by the dv.factors argument. In this case, merge.mod applies and dv.cutoff is not required (as cutoff = 1 is implied).
dv.factors	Names of the factors to be considered. Must be equal to 2. If missing (the default), the first and second factor of the model are selected.
merge.mod	This is used for merging. If FALSE (the default), fit measures for mod1 are estimated for a single model as long as no mod2 is provided. If TRUE, a merged model from function merge_factors is estimated based on mod1. In this case, no mod2 is required.
dv.cutoff	This is used for constraining. It determines the critical correlation assumed to be a cutoff for discriminant validity testing. For example, based on Rönkkö & Cho (2020), a cutoff of .9 indicates a severe issue in discriminant validity between the selected factors. Cutoffs between .8 and 1 are recommended. The function returns a warning, if the cutoff is below .8.
standardized	Are factor loadings assumed to be standardized and covariances to be correlations (default: TRUE)?

assume.mvn	Should multivariate normality (mvn) be assumed? If TRUE (the default), kurtosis and skewness are set to 1 for simulated data. If FALSE, kurtosis and skewness are estimated from dataset x via semTools::mardiaKurtosis and semTools::mardiaSkew.
multi.core	Should multiple cores be used to simulate fit indices? If TRUE (the default), mclapply (on Linux or Mac machines) or parLapply (on Windows machines) from parallel package with the number of specified cores is used. If FALSE, a single core is used.
cores	How many cores should be used for multiple cores? The default is 2. Consider the available number of cores of your system.
seed	The seed to be set to obtain reproducible cutoffs (default: 1111). Defines a vector of length rep with the seed being the first value.
pop.mod1	For flexibility reasons, an optional lavaan population model can be provided. This is required together with n if x is missing.
pop.mod2	Another optional lavaan population model.

Value

A list of simulated fit statistics (fco) and all previously defined parameters.

References

- Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling*, 6(1), 1–55. <https://doi.org/10.1080/10705519909540118>
- Niemand, T., & Mai, R. (2018). Flexible cutoff values for fit indices in the evaluation of structural equation models. *Journal of the Academy of Marketing Science*, 46(6), 1148—1172. <https://doi.org/10.1007/s11747-018-0602-9>
- Rönkkö, M., & Cho, E. (2020). An updated guideline for assessing discriminant validity. *Organizational Research Methods*. <https://doi.org/10.1177/1094428120968614>

Examples

```
#Note: Demonstration only! Please use higher numbers of replications for your applications (>= 500).
#A single model to obtain fit indices for
mod <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
"
fits.single <- gen_fit(mod1 = mod, x = bb1992, rep = 10, standardized = FALSE)

#Two models, an unconstrained and a constrained model to compare fit indices
mod.con <- "
F1 =~ Q5 + Q7 + Q8
```

```

F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
F1 ~~ 0 * F2
"

fits.con <- gen_fit(
  mod1 = mod,
  mod2 = mod.con,
  x = bb1992,
  rep = 10
)
#Two models for discriminant validity testing, this resembles constraining with a cutoff of .9
fits.dv.con <- gen_fit(
  mod1 = mod,
  x = bb1992,
  rep = 10,
  dv = TRUE,
  dv.factors = c("F4", "F5"),
  dv.cutoff = .9
)

#Two models for discriminant validity testing, this resembles merging.
fits.dv.merge <- gen_fit(
  mod1 = mod,
  x = bb1992,
  rep = 10,
  dv = TRUE,
  dv.factors = c("F4", "F5"),
  merge.mod = TRUE
)

```

index_guess

Helper function that guesses GoF or BoF from a given index name

Description

Helper function that guesses GoF or BoF from a given index name

Usage

```
index_guess(index)
```

Arguments

index A fit index or measure provided by function fitmeasures in package lavaan

Value

Returns GoF (Goodness-of-Fit index) or BoF (Badness of Fit index).

Examples

```

index_guess("cfi")
index_guess("tli")
index_guess("rmsea")
index_guess("srmr")

```

pop_mod	<i>Helper function to obtain population model for simulation based on data and model</i>
---------	--

Description

Helper function to obtain population model for simulation based on data and model

Usage

```
pop_mod(mod, x, type = "NM", standardized = TRUE, afl = 0.7, aco = 0.3)
```

Arguments

mod	A lavaan model (only CFA supported so far)
x	A dataset for the model of nrow observations (minimum: 50) and ncol indicators (minimum: 4)
type	Type of population model. NM (the default): Uses the factor loadings and covariances from Niemand & Mai's (2018) simulation study. HB: Uses the factor loadings and covariances from Hu & Bentler's (1999) simulation study. EM: Empirical, uses the given factor loadings and covariances. EM is not recommended for confirmative use as it leads to the least generalizable cutoffs.
standardized	Are factor loadings assumed to be standardized and covariances to be correlations (default: TRUE)?
afl	Average factor loading of indicators per factor, only relevant for type = "NM" (default: .7).
aco	Average correlation between factors, only relevant for type = "NM" (default: .3).

Value

List of population model type, standardized, average factor loading and average correlation. All values are round to three decimals.

Examples

```

mod <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
"

pop_mod(mod, x = bb1992, type = "NM")$pop.mod
pop_mod(mod, x = bb1992, type = "HB")$pop.mod
pop_mod(mod, x = bb1992, type = "EM")$pop.mod
pop_mod(mod, x = bb1992, type = "NM", afl = .9)$pop.mod
pop_mod(mod, x = bb1992, type = "NM", aco = .5)$pop.mod
pop_mod(mod, x = bb1992, type = "EM", standardized = FALSE)$pop.mod

```

recommend

Obtain recommendations based on Mai et al. (2021)

Description

This function recommends pre-defined selected fit indices in case the user does not know which fit index should be used for model evaluation. Results may differ based on three settings, the sample size of the data, the research purpose of the investigated model and the focus of the model. For obvious reasons, this function only works for single models and does not accept any other model type.

Usage

```

recommend(
  fits,
  purpose = "novel",
  focus = "cfa",
  override = FALSE,
  index = NULL,
  digits = 3
)

```

Arguments

fits	A list of simulated fit indices obtained from <code>gen_fit</code> . Based on the structure of fits, the number of models is derived.
purpose	The research purpose of the model investigated. Is the underlying model novel (default) or established (= established). This parameter is relevant to find the proper recommended fit indices.
focus	The focus of estimation for the model. Is the focus on CFA (default) or analyzing the structural model of a theoretical model (= structural)? This parameter is relevant to find the proper recommended fit indices.

override	Should the recommendations by Mai et al. (2021) overridden (default: FALSE)? This may be useful to explore models outside of the scope of the paper. In this case, the recommended fit indices are not determined by the function, and hence need to be provided. In this case, the function requires the argument index.
index	An optional vector of fit indices or measures provided by function fitmeasures in package lavaan. This argument is required when override is TRUE. It is ignored otherwise.
digits	An optional integer to round fit values and cutoffs (min: 1, max: 5).

Value

A list of information regarding the recommended fit indices based on Mai et al. (2021) or when overridden, based on the provided indices.

References

Mai, R., Niemand, T., & Kraus, S. (2021). A Tailor-Fit Model Evaluation Strategy for Better Decisions about Structural Equation Models, *Technological Forecasting & Social Change*, 173(December) 121142. <https://doi.org/10.1016/j.techfore.2021.121142>

Examples

```
#Note: Demonstration only! Please use higher numbers of replications for your applications (>= 500).
mod <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
"

fits.single <- gen_fit(mod1 = mod, x = bb1992, rep = 10, standardized = FALSE)
recommend(fits.single)
recommend(fits.single, purpose = "established")
recommend(fits.single,
          override = TRUE,
          index = c("CFI", "SRMR"))
```

recommend_dv

Obtain recommendations for discriminant validity testing

Description

This function recommends on potential issues for discriminant validity testing, based on differences between fit values and differences between flexible cutoffs. Two approaches of testing are supported: merging and constraining.

Usage

```
recommend_dv(fits, index = "CFI", digits = 3)
```

Arguments

<code>fits</code>	A list of simulated fit indices obtained from <code>gen_fit</code> . Based on the structure of <code>fits</code> , the number of models is derived.
<code>index</code>	A vector of fit indices or measures provided by function <code>fitmeasures</code> in package <code>lavaan</code> . The default is set to CFI.
<code>digits</code>	An optional integer to round fit values and cutoffs (min: 1, max: 5).

Value

A list of information regarding discriminant validity testing.

Examples

```
#Note: Demonstration only! Please use higher numbers of replications for your applications (>= 500).
mod <- "
F1 =~ Q5 + Q7 + Q8
F2 =~ Q2 + Q4
F3 =~ Q10 + Q11 + Q12 + Q13 + Q18 + Q19 + Q20 + Q21 + Q22
F4 =~ Q1 + Q17
F5 =~ Q6 + Q14 + Q15 + Q16
"

#Two models for discriminant validity testing, this resembles constraining with a cutoff of .9
fits.dv.con <- gen_fit(
  mod1 = mod,
  x = bb1992,
  rep = 10,
  dv = TRUE,
  dv.factors = c("F4", "F5"),
  dv.cutoff = .9
)
recommend_dv(fits.dv.con)

#Two models for discriminant validity testing, this resembles merging.
fits.dv.merge <- gen_fit(
  mod1 = mod,
  x = bb1992,
  rep = 10,
  dv = TRUE,
  dv.factors = c("F4", "F5"),
  merge.mod = TRUE
)
recommend_dv(fits.dv.merge)
```

Index

* **datasets**

bb1992, [2](#)

bb1992, [2](#)

flex_co, [3](#)

gen_fit, [5](#)

index_guess, [8](#)

pop_mod, [9](#)

recommend, [10](#)

recommend_dv, [11](#)