Package: dynamic (via r-universe)

September 18, 2024

Type Package

Title DFI Cutoffs for Latent Variable Models

Version 1.1.0

Description Returns dynamic fit index (DFI) cutoffs for latent variable models that are tailored to the user's model statement, model type, and sample size. This is the counterpart of the Shiny Application, <https://dynamicfit.app>.

License AGPL-3

Language en-US

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 4.0.0)

URL https://github.com/melissagwolf/dynamic

BugReports https://github.com/melissagwolf/dynamic/issues

Imports dplyr (>= 1.1.0), simstandard (>= 0.6.2), tidyr (>= 1.1.0), lavaan (>= 0.6-7), ggplot2 (>= 3.3.0), magrittr (>= 1.5), tibble (>= 3.0.0), patchwork (>= 1.1.1), semTools (>= 0.5.5), stringr (>= 1.4.0), MASS (>= 7.3-30), rlang (>= 1.1.0), purrr (>= 0.3.3), Bayesrel (>= 0.7.7), GenOrd (>= 1.3.0), mirt (>= 1.35)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository https://melissagwolf.r-universe.dev

RemoteUrl https://github.com/melissagwolf/dynamic

RemoteRef HEAD

RemoteSha 48f73c702eadddb83e141ac03603cbead1a84ca0

Contents

---

catHB *Dynamic fit index (DFI) cutoffs for categorical multi-factor CFA models*

Description

This function generates DFI cutoffs for multi-factor CFA models that treat items as categorical. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size (including threshold estimates).

The app-based version of this function can be found at dynamicfit.app.

Usage

```
catHB(
  model,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  reps = 250,
  estimator = "WLSMV"
)

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

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings and thresholds.

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification.

`manual` If you entered a `lavaan` object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.

`reps` The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is WLSMV. Only limited-information estimators that produce fit indices are permitted (i.e., maximum likelihood is not available)

`x` catHB object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#two-factor model with correlated factors
m1<-"
 F1=~X1 + X2 +X3
 F2=~X6 + X7 + X8 + X9
 F1~~F2"

 #fit the model in lavaan, treating items are categorical
 fit<-lavaan::cfa(m1, data=Example, ordered=TRUE)

catHB(fit)

#Manual entry example (manual=TRUE)

#two-factor model with correlated factors
m1<-"
 F1=~X1 + X2 +X3
 F2=~X6 + X7 + X8 + X12
 F1~~F2"
 #fit the model, treating items are categorical
 #lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example, ordered=TRUE)
 lavaan::standardizedsolution(fit)

#thresholds go in model statement as
  #(a)categorical item name
  #(b) vertical pipe
  #(c) estimate
  #(d)times t+threshold number

manual_model <-"F1=~.448*X1 + .557*X2 + .770*X3
 F2=~.612*X6 + .684*X7 + .736*X8 + .365*X12
 F1~~.424*F2

X1 |-0.285*t1
X1 | 0.337*t2
X1 | 0.793*t3
X1 | 1.305*t4

X2 |-0.243*t1
X2 | 0.369*t2
X2 | 0.849*t3
X2 | 1.227*t4

X3 |-0.285*t1
X3 | 0.353*t2
X3 | 0.827*t3
X3 | 1.379*t4

X6 |-0.279*t1
X6 | 0.353*t2
X6 | 0.827*t3
X6 | 1.379*t4

X7 |-0.269*t1
X7 | 0.385*t2
X7 | 0.871*t3
X7 | 1.329*t4

X8 |-0.274*t1
X8 | 0.358*t2
X8 | 0.779*t3
X8 | 1.237*t4

X12 |-0.248*t1
X12 | 0.440*t2
X12 | 0.900*t3
X12 | 1.392*t4"
catHB(model=manual_model,n=500,manual=TRUE)
```

catOne *Dynamic fit index (DFI) cutoffs for categorical one-factor CFA models*

Description

This function generates DFI cutoffs for one-factor CFA models that treat items as categorical. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size (including threshold estimates).

The app-based version of this function can be found at dynamicfit.app.

Usage

```
catOne(
  model,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  reps = 250,
  estimator = "WLSMV"
)

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

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings and thresholds.

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification.

`manual` If you entered a `lavaan` object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.

`reps` The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is WLSMV. Only limited-information estimators that produce fit indices are permitted (i.e., maximum likelihood is not available)

`x` catOne object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#one-factor model
m1<-"F1=~X5+ X6 + X7 + X8 + X9"

 #fit the model in lavaan, treating items are categorical
 fit<-lavaan::cfa(m1, data=Example, ordered=TRUE)

catOne(fit)

#Manual entry example (manual=TRUE)

#one-factor model with correlated factors
m1<-"F1=~X5+ X6 + X7 + X8 + X9"

 #fit the model, treating items are categorical
 #lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example, ordered=TRUE)
 lavaan::standardizedsolution(fit)

#thresholds go in model statement as
  #(a)categorical item name
  #(b) vertical pipe
  #(c) estimate
  #(d)times t+threshold number

manual_model <-"F1=~.550*X5 + .614*X6 + .726*X7 + .723*X8 + .236*X9

X5 |-0.274*t1
X5 | 0.305*t2
X5 | 0.765*t3
X5 | 1.259*t4

X6 |-0.279*t1
X6 | 0.353*t2
X6 | 0.779*t3
X6 | 1.175*t4

X7 |-0.269*t1
X7 | 0.385*t2
X7 | 0.871*t3
X7 | 1.329*t4

X8 |-0.274*t1
```

```
X8 | 0.358*t2
X8 | 0.779*t3
X8 | 1.237*t4

X9 |-0.269*t1
X9 | 0.342*t2
X9 | 0.745*t3
X9 | 1.248*t4"

catOne(model=manual_model,n=500,manual=TRUE)
```

cfaHB — *Dynamic fit index (DFI) cutoffs adapted from Hu & Bentler (1999) for multi-factor CFA models*

Description

This function generates DFI cutoffs adapted from Hu & Bentler (1999) for multi-factor CFA models using ML estimation. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
cfaHB(
  model,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = "ML",
  reps = 500
)

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

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings.

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification.

`manual` If you entered a `lavaan` object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is ML.

`reps` The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.

`x` cfaHB object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Melissa G Wolf & Daniel McNeish

Maintainer: Melissa G Wolf <missgord@gmail.com>

Examples

```
#Lavaan object example (manual=FALSE)
dat <- lavaan::HolzingerSwineford1939
lavmod <- "F1 =~ x1 + x2 + x3
           F2 =~ x4 + x5 + x6
           F3 =~ x7 + x8 + x9"
fit <- lavaan::cfa(lavmod,dat)
cfaHB(fit)

#Manual entry example for a sample size of 400 (manual=TRUE)
manmod <- "F1 =~ .602*Y1 + .805*Y2 + .516*Y3 + .415*Y4
           F2 =~ .413*Y5 + -.631*Y6
           F1 ~~ .443*F2
           Y4 ~~ .301*Y5"
cfaHB(model=manmod,n=400,manual=TRUE)
```

`cfaOne` *Dynamic fit index (DFI) cutoffs for one-factor CFA models*

Description

This function generates DFI cutoffs for one-factor CFA models using ML estimation. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
cfaOne(
  model,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = ”ML”,
  reps = 500
)

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

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings.

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification.

`manual` If you entered a `lavaan` object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is ML.

`reps` The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.

`x` cfaOne object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Melissa G Wolf & Daniel McNeish

Maintainer: Melissa G Wolf <melissagordon@ucsb.edu>

Examples

```
#Lavaan object example (manual=FALSE)
dat <- lavaan::HolzingerSwineford1939
lavmod <- "F1 =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9”
fit <- lavaan::cfa(lavmod,dat)
cfaOne(fit)

#Manual entry example for a sample size of 300 (manual=TRUE)
```

```
manmod <- "F1 =~ .602*Y1 + .805*Y2 + .857*Y3 + .631*Y4 + .345*Y5 + .646*Y6”
cfaOne(model=manmod,n=300,manual=TRUE)
```

DDDFI — *Direct Discrepancy Dynamic fit index (3DFI) cutoffs for arbitrary covariance structure models*

Description

This function generates DFI cutoffs for any single group covariance structure model with a saturated or absent mean structure. It supports (a) any estimator supported by lavaan (e.g. ML, MLR, WLSMV, ULSMV), (b) missing data, and (c) multiple response scales (normal, non-normal continuous, categorical). The default argument is a singular argument: a `lavaan` object.The function can also accommodate manual entry of the model statement and sample size. Some features require an original dataset to be provided (e.g., missing data, categorical data). The app-based version of this function can be found at dynamicfit.app.

Usage

```
DDDFI(
  model,
  data = NULL,
  scale = "normal”,
  manual = FALSE,
  reps = 250,
  n = NULL,
  estimator = NULL,
  MAD = c(0.038, 0.05, 0.06),
  plot.dfi = FALSE,
  plot.dist = FALSE,
  plot.discrepancy = FALSE
)

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

Arguments

`model` This can either be a `lavaan` object, OR a model statement written in `lavaan` `model.syntax` with standardized estimates

`data` The original data to which the model was applied. Not required if scale="normal". Otherwise, data is required.

`scale` Determines how data are simulated. Options are "normal", "nonnormal", or "categorical". "normal" assumes multivariate normality across all variables. "nonnormal" recreates distributions in an empirical dataset (to be provided by the user), assuming variables are continuous. "categorical" simulates discrete data with the same proportions as an empirical dataset (to be provided by the user). With "categorical", mixed formats are also supported and any variable with more than 9 categories is simulated from a normal distribution.Only "normal" can be used without provided an original dataset.

`manual` If you entered a `lavaan` object, keep this set to FALSE. If you manually entered standardized estimates and sample size, set this to TRUE.

`reps` The number of replications used in the simulations. This is set to 250 by default

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`estimator` Which estimator to use within the simulations (enter in quotes). The default depends on the scale option ("ML" for "normal", "MLR" for "nonnormal", and "WLSMV" for categorical)

`MAD` Mean Absolute Discrepancies to test in the simulation. Default is c(.038, .05, .06) to recreate traditional "Close", "Fair", "Mediocre" benchmarks

`plot.dfi` Displays simulated distributions of fit indices used to derive cutoffs for each MAD value.

`plot.dist` Displays distributions of simulated data (and empirical data, if provided) to assess fidelity of simulated data to empirical data

`plot.discrepancy`
Displays distributions of simulated MAD values

`x` DDDFI object

`...` other print parameters

Value

Direct Discrepancy Dynamic fit index (DFI) cutoffs for CFI, RMSEA, and RMSEA 90

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

lavmod <- "F1 =~ x1 + x2 + x3
           F2 =~ x4 + x5 + x6
           F3 =~ x7 + x8 + x9"

fit <- lavaan::cfa(lavmod,data=Holzinger)
DDDFI(fit)

#Manual entry example (manual=TRUE)

#Holzinger 3-factor model
lavmod <- "F1 =~ x1 + x2 + x3
```

```
           F2 =~ x4 + x5 + x6
           F3 =~ x7 + x8 + x9"

 #fit the model,lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(lavmod, data=Holzinger)
 lavaan::standardizedsolution(fit)

 #model statement with standardized estimates
 manual_model <- "F1 =~ .772*x1 + .424*x2 + .581*x3
            F2 =~ .852*x4 + .855*x5 + .838*x6
            F3 =~ .570*x7 + .723*x8 + .665*x9
            F1~~.459*F2
            F1~~.471*F3
            F2~~.283*F3"

DDDFI(model=manual_model,n=301,manual=TRUE)
```

equivTest — *Equivalence testing with adjusted fit indexes for structural equation modeling*

Description

This function generates adjusted fit index cutoffs using equivalence testing, introduced by Yuan, Chan, Marcoulides, & Bentler (2016). The default argument is a singular argument: a `lavaan` object. The function can also accommodate manual entry of the sample size (n), model chi-square (T_ml), degrees of freedom (df), baseline chi-square (T_mli), and number of observed variables (p).

The app-based version of this function can be found at dynamicfit.app.

Usage

```
equivTest(
  n,
  T_ml = NULL,
  df = NULL,
  T_mli = NULL,
  p = NULL,
  manual = FALSE,
  plot = FALSE
)

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

Arguments

`n`	This can either be a `lavaan` object, OR your sample size.
`T_ml`	If you entered a `lavaan` object for n, leave this blank. Otherwise, enter your model chi-square.
`df`	If you entered a `lavaan` object for n, leave this blank. Otherwise, enter your model degrees of freedom.
`T_mli`	If you entered a `lavaan` object for n, leave this blank. Otherwise, enter your baseline chi-square.
`p`	If you entered a `lavaan` object for n, leave this blank. Otherwise, enter the number of observed variables in your model.
`manual`	If you entered a `lavaan` object, keep this set to FALSE. If you manually entered each argument, set this to TRUE.
`plot`	Displays a simple plot that compares your T-size RMSEA and T-Size CFI to the adjusted bins.
`x`	equivTest object
`...`	other print parameters

Value

T-size RMSEA and T-Size CFI, along with adjusted bins for each index

Author(s)

Melissa G Wolf & Daniel McNeish

Maintainer: Melissa G Wolf <melissagordon@ucsb.edu>

Examples

```
#Lavaan object example (manual=FALSE)
dat <- lavaan::HolzingerSwineford1939
lavmod <- "F1 =~ x1 + x2 + x3
           F2 =~ x4 + x5 + x6
           F3 =~ x7 + x8 + x9"
fit <- lavaan::cfa(lavmod,dat)
equivTest(fit)

#' #Manual entry example (manual=TRUE)
n <- 301
T_ml <- 85.306
df <- 24
T_mli <- 918.852
p <- 9
equivTest(n,T_ml,df,T_mli,p,manual=TRUE)
```

exactFit *DFI cutoffs for a Test of Exact Fit*

Description

This function generates DFI cutoffs by treating the data generating model as the true model (using ML estimation). The default argument is a singular argument: a lavaan object from the cfa function. The function can also accommodate manual entry of the model statement and sample size.

Usage

```
exactFit(model, n, plot = FALSE, manual = FALSE, reps = 500)

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

Arguments

model	This can either be a lavaan object from the cfa function, OR a model statement written in lavaan model.syntax with standardized loadings.
n	If you entered a lavaan object for model, leave this blank. Otherwise, enter your sample size (numeric).
plot	Displays distributions of fit indices for each fit index.
manual	If you entered a lavaan object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.
reps	(**Do not modify this**): The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App (not yet available).
x	exactFit object
...	other print parameters

Value

Dynamic fit index (DFI) cutoffs for Chi-Square, SRMR, RMSEA, and CFI.

Author(s)

Melissa G Wolf & Daniel McNeish

Maintainer: Melissa G Wolf <missgord@gmail.com>

Examples

```
#Lavaan object example (manual=FALSE)
dat <- lavaan::HolzingerSwineford1939
lavmod <- "F1 =~ x1 + x2 + x3
           F2 =~ x4 + x5 + x6
           F3 =~ x7 + x8 + x9"
fit <- lavaan::cfa(lavmod,dat)
exactFit(fit)

#Manual entry example for a sample size of 400 (manual=TRUE)
manmod <- "F1 =~ .602*Y1 + .805*Y2 + .516*Y3 + .415*Y4
           F2 =~ .413*Y5 + -.631*Y6
           F1 ~~ .443*F2
           Y4 ~~ .301*Y5"
exactFit(model=manmod,n=400,manual=TRUE)
```

`Example` *DFI Example Data*

Description

Simulated 5-point Likert item responses to 12 items for 500 people.

Usage

```
Example
```

Format

## 'Example' A data frame with 12 columns and 500 rows

X1 Likert response to item 1

X2 Likert response to item 2

X3 Likert response to item 3

X4 Likert response to item 4

X5 Likert response to item 5

X6 Likert response to item 6

X7 Likert response to item 7

X8 Likert response to item 8

X9 Likert response to item 9

X10 Likert response to item 10

X11 Likert response to item 11

X12 Likert response to item 12

Source

Simulated with simstandard package

hier1HB *Dynamic fit index (DFI) cutoffs adapted from Hu & Bentler (1999) for measurement misspecification in hierarchical factor models*

Description

This function generates DFI cutoffs adapted from Hu & Bentler (1999) for measurement misspecification in hierarchical factor models using, by default, ML estimation. The default argument is a singular argument: a lavaan object from the cfa function. The function can also accommodate manual entry of the model statement and sample size.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
hier1HB(
  model,
  n = NULL,
  estimator = "ML",
  plot = FALSE,
  manual = FALSE,
  reps = 500
)

## S3 method for class 'hier1HB'
print(x, ...)
```

Arguments

model	This can either be a lavaan object from the cfa function, OR a model statement written in lavaan model.syntax with standardized loadings.
n	If you entered a lavaan object for model, leave this blank. Otherwise, enter your sample size (numeric).
estimator	Which estimator to use within the simulations (enter in quotes). The default is maximum likelihood.
plot	Displays distributions of fit indices for each level of misspecification.
manual	If you entered a lavaan object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.
reps	The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.
x	hier1HB object
...	other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish, Melissa G Wolf, & Patrick D Manapat

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Manual entry example for a sample size of 2200 (manual=TRUE), from Reynolds & Keith (2017)

manmod <- "G =~ .51*F1 + .83*F2 + .97*F3 + .83*F4 + .87*F5 + .55*Y7
           F1 =~ .41*Y1 + .81*Y2 + .71*Y3
           F2 =~ .79*Y4 + .64*Y5 + .81*Y6 + .22*Y7
           F3 =~ .53*Y8 + .68*Y9 + .66*Y10
           F4 =~ .79*Y11 + .76*Y12
           F5 =~ .82*Y13 + .71*Y14 + .85*Y15 + .81*Y16
           F3 ~~ .77*F4"
hier2(model=manmod,n=2200,manual=TRUE)
```

---

hier2 — *Dynamic fit index (DFI) cutoffs for structural misspecification in hierarchical factor models*

Description

This function generates DFI cutoffs for structural misspecification in hierarchical factor models using, by default, ML estimation. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
hier2(
  model,
  n = NULL,
  estimator = "ML",
  plot = FALSE,
  manual = FALSE,
  reps = 500
)

## S3 method for class 'hier2'
print(x, ...)
```

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings.

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`estimator` Which estimator to use within the simulations (enter in quotes). The default is maximum likelihood.

`plot` Displays distributions of fit indices for each level of misspecification.

`manual` If you entered a `lavaan` object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.

`reps` The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.

`x` hier2 object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish, Melissa G Wolf, & Patrick D Manapat

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Manual entry example for a sample size of 2200 (manual=TRUE), from Reynolds & Keith (2017)

manmod <- "G =~ .51*F1 + .83*F2 + .97*F3 + .83*F4 + .87*F5 + .55*Y7
           F1 =~ .41*Y1 + .81*Y2 + .71*Y3
           F2 =~ .79*Y4 + .64*Y5 + .81*Y6 + .22*Y7
           F3 =~ .53*Y8 + .68*Y9 + .66*Y10
           F4 =~ .79*Y11 + .76*Y12
           F5 =~ .82*Y13 + .71*Y14 + .85*Y15 + .81*Y16
           F3 ~~ .77*F4"
hier2(model=manmod,n=2200,manual=TRUE)
```

`Holzinger` *Holzinger & Swineford (1939) Data*

Description

Subset of 301 students from classic Holzinger & Swineford (1939) data with 9 mental ability scores

Usage

```
Holzinger
```

Format

## 'Holzinger' A data frame with 10 columns and 301 rows

id student identification number

x1 Visual Perception score

x2 Cubes score

x3 Lozenges score

x4 Paragraph Comprehension score

x5 Sentence Completion score

x6 Word Meaning score

x7 Speeded Addition score

x8 Speended Dot Counting score

x9 Speeded Letter Discrimination score

Source

lavaan R package `https://github.com/yrosseel/lavaan/tree/master/data`

References

Holzinger, K., & Swineford, F. (1939). A study in factor analysis: The stability of a bifactor solution. Supplementary Educational Monograph, no. 48. Chicago: University of Chicago Press.

`likertHB` *Dynamic fit index (DFI) cutoffs multi-factor CFA models with Likert-type items*

Description

This function generates DFI cutoffs for multi-factor CFA models that treats Likert-type items as continuous. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size. A primary difference in likert DFI functions is that a dataset must also be provided in the 'data' argument in order to simulate data with the same number of response options and response frequencies as the original data. When Likert-type items are treated as continuous, this information cannot be obtained solely from model output.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
likertHB(
  model,
  data,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = "ML",
  reps = 250
)

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

Arguments

model	This can either be a lavaan object from the cfa function, OR a model statement written in lavaan model.syntax with standardized loadings.
data	An empirical dataset used to determine the number of Likert responses and the response frequencies
n	If you entered a lavaan object for model, leave this blank. Otherwise, enter your sample size (numeric).
plot	Displays distributions of fit indices for each level of misspecification. This also includes plots to visualize how close the simulated data are to the original data.
manual	If you entered a lavaan object, keep this set to FALSE (the default). If you manually entered standardized loadings and sample size, set this to TRUE.
estimator	Which estimator to use within the simulations (enter in quotes). The default is ML
reps	The number of replications used in your simulation. This is set to 500 by default.
x	likertHB object
...	other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#two-factor model with correlated factors
```

```
m1<-"
 F1=~X2 + X3 + X4
 F2=~X6 + X7 + X8
 F1~~F2"

 #fit the model in lavaan, treating items are continuous
 fit<-lavaan::cfa(m1, data=Example)

likertHB(fit, data=Example)

#Manual entry example (manual=TRUE)

#two-factor model with correlated factors
m1<-"
 F1=~X2 + X3 + X4
 F2=~X6 + X7 + X8
 F1~~F2"

 #fit the model, treating items are continuous
 #lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example)
 lavaan::standardizedsolution(fit)


manual_model <-"F1=~.554*X2 + .654*X3 + .733*X4
 F2=~.537*X6 + .666*X7 + .723*X8
 F1~~.339*F2"

likertHB(model=manual_model,data=Example,n=500,manual=TRUE)
```

likertHB2 *Dynamic fit index (DFI) cutoffs multi-factor CFA models with Likert-type items*

Description

This function generates DFI cutoffs for multi-factor CFA models that treats Likert-type items as continuous. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size. A primary difference in likert DFI functions is that a dataset must also be provided in the 'data' argument in order to simulate data with the same number of response options and response frequencies as the original data. When Likert-type items are treated as continuous, this information cannot be obtained solely from model output.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
likertHB2(
  model,
  data,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = "ML",
  reps = 250
)

## S3 method for class 'likertHB2'
print(x, ...)
```

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings.

`data` An empirical dataset used to determine the number of Likert responses and the response frequencies

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification. This also includes plots to visualize how close the simulated data are to the original data.

`manual` If you entered a `lavaan` object, keep this set to FALSE (the default). If you manually entered standardized loadings and sample size, set this to TRUE.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is ML

`reps` The number of replications used in your simulation. This is set to 500 by default.

`x` likertHB2 object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#two-factor model with correlated factors
```

```
m1<-"
 F1=~X2 + X3 + X4
 F2=~X6 + X7 + X8
 F1~~F2"

 #fit the model in lavaan, treating items are continuous
 fit<-lavaan::cfa(m1, data=Example)

likertHB2(fit, data=Example)

#Manual entry example (manual=TRUE)

#two-factor model with correlated factors
m1<-"
 F1=~X2 + X3 + X4
 F2=~X6 + X7 + X8
 F1~~F2"

 #fit the model, treating items are continuous
 #lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example)
 lavaan::standardizedsolution(fit)


manual_model <-"F1=~.554*X2 + .654*X3 + .733*X4
 F2=~.537*X6 + .666*X7 + .723*X8
 F1~~.339*F2"

likertHB2(model=manual_model,data=Example,n=500,manual=TRUE)
```

---

likertOne *Dynamic fit index (DFI) cutoffs one-factor CFA models with Likert-type items*

Description

This function generates DFI cutoffs for one-factor CFA models that treats Likert-type items as continuous. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size. A primary difference in likert DFI functions is that a dataset must also be provided in the 'data' argument in order to simulate data with the same number of response options and response frequencies as the original data. When Likert-type items are treated as continuous, this information cannot be obtained solely from model output.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
likertOne(
  model,
  data,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = "ML",
  reps = 250
)

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

Arguments

`model` This can either be a `lavaan` object from the `cfa` function, OR a model statement written in `lavaan` `model.syntax` with standardized loadings.

`data` An empirical dataset used to determine the number of Likert responses and the response frequencies

`n` If you entered a `lavaan` object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification. This also includes plots to visualize how close the simulated data are to the original data.

`manual` If you entered a `lavaan` object, keep this set to FALSE (the default). If you manually entered standardized loadings and sample size, set this to TRUE.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is ML

`reps` The number of replications used in your simulation. This is set to 250 by default.

`x` likertOne object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#one-factor model
```

```
m1<-"F1=~X5+ X6 + X7 + X8 + X9"

 #fit the model in lavaan, treating items are continuous
 fit<-lavaan::cfa(m1, data=Example)

likertOne(fit, data=Example)

#Manual entry example (manual=TRUE)

#one-factor model with correlated factors
m1<-"F1=~X5+ X6 + X7 + X8 + X9"

 #fit the model, treating items are continuous
 #lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example)
 lavaan::standardizedsolution(fit)

manual_model <-"F1=~.517*X5 + .549*X6 + .679*X7 + .694*X8 + .203*X9"

likertOne(model=manual_model,data=Example,n=500,manual=TRUE)
```

nnorHB *Dynamic fit index (DFI) cutoffs for continuous, non-normal multi-factor CFA models with (possible) missing data*

Description

This function generates DFI cutoffs for multi-factor CFA models that treats items as continuous and non-normal with possible missing data. This functions uses a modified Bollen-Stine bootstrap to accommodate non-normality and missingness rather than simulating from a particular distribution. The default argument is a singular argument: a lavaan object from the cfa function. The function can also accommodate manual entry of the model statement and sample size (including threshold estimates). A primary difference in nnor DFI functions is that a dataset from which to bootstrap must also be provided in the 'data' argument.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
nnorHB(
  model,
  data,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = "MLR",
  reps = 500
)

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

Arguments

model	This can either be a lavaan object from the cfa function, OR a model statement written in lavaan model.syntax with standardized loadings.
data	An empirical dataset to which a modified Bollen-Stine bootstrap will be applied to create hypothetical misspecified data
n	If you entered a lavaan object for model, leave this blank. Otherwise, enter your sample size (numeric).
plot	Displays distributions of fit indices for each level of misspecification.This also includes plots to visualize how close the distributions of the hypothetical data come to the original data.
manual	If you entered a lavaan object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.
estimator	Which estimator to use within the simulations (enter in quotes). The default is MLR
reps	The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.
x	nnorHB object
...	other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#two-factor model with correlated factors
m1<-"
 F1=~X2 + X3 + X4
 F2=~X6 + X7 + X8
 F1~~F2"

 #fit the model in lavaan, treating items are continuous
 fit<-lavaan::cfa(m1, data=Example)
```

```
nnorHB(fit, data=Example)

#Manual entry example (manual=TRUE)

#two-factor model with correlated factors
m1<-"
 F1=~X2 + X3 + X4
 F2=~X6 + X7 + X8
 F1~~F2"

 #fit the model, treating items are continuous
 #lavaan is used here to shown where estimates come frOm
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example)
 lavaan::standardizedsolution(fit)


manual_model <-"F1=~.554*X2 + .654*X3 + .733*X4
 F2=~.537*X6 + .666*X7 + .723*X8
 F1~~.339*F2"

nnorHB(model=manual_model,data=Example,n=500,manual=TRUE)
```

nnorOne — *Dynamic fit index (DFI) cutoffs for continuous, non-normal one-factor CFA models with (possible) missing data*

Description

This function generates DFI cutoffs for one-factor CFA models that treats items as continuous and non-normal with possible missing data. This functions uses a modified Bollen-Stine bootstrap to accommodate non-normality and missingness rather than simulating from a particular distribution. The default argument is a singular argument: a `lavaan` object from the `cfa` function. The function can also accommodate manual entry of the model statement and sample size (including threshold estimates). A primary difference in nnor DFI functions is that a dataset from which to bootstrap must also be provided in the 'data' argument.

The app-based version of this function can be found at dynamicfit.app.

Usage

```
nnorOne(
  model,
  data,
  n = NULL,
  plot = FALSE,
  manual = FALSE,
  estimator = "MLR",
  reps = 500
)

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

Arguments

`model` This can either be a lavaan object from the cfa function, OR a model statement written in lavaan model.syntax with standardized loadings

`data` An empirical dataset to which a modified Bollen-Stine bootstrap will be applied to create hypothetical misspecified data

`n` If you entered a lavaan object for model, leave this blank. Otherwise, enter your sample size (numeric).

`plot` Displays distributions of fit indices for each level of misspecification.This also includes plots to visualize how close the distributions of the hypothetical data come to the original data.

`manual` If you entered a lavaan object, keep this set to FALSE. If you manually entered standardized loadings and sample size, set this to TRUE.

`estimator` Which estimator to use within the simulations (enter in quotes). The default is MLR

`reps` The number of replications used in your simulation. This is set to 500 by default in both the R package and the corresponding Shiny App.

`x` nnorOne object

`...` other print parameters

Value

Dynamic fit index (DFI) cutoffs for SRMR, RMSEA, and CFI.

Author(s)

Daniel McNeish & Melissa G Wolf

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
#Example using a lavaan object as input (manual=FALSE)

#one-factor model
m1<-"F1=~X5+ X6 + X7 + X8 + X9"

 #fit the model in lavaan, treating items are continuous
 fit<-lavaan::cfa(m1, data=Example)

nnorOne(fit, data=Example)
```

```
#Manual entry example (manual=TRUE)

#one-factor model with correlated factors
m1<-"F1=~X5+ X6 + X7 + X8 + X9"

 #fit the model, treating items are continuous
 #lavaan is used here to shown where estimates come from
 #but manual entry supports standardized estimates from models fit in any software

 fit<-lavaan::cfa(m1, data=Example)
 lavaan::standardizedsolution(fit)

manual_model <-"F1=~.517*X5 + .549*X6 + .679*X7 + .694*X8 + .203*X9"

nnorOne(model=manual_model,data=Example,n=500,manual=TRUE)
```

RelRep *Reliability Representativeness of Coefficient Alpha or Omega*

Description

This function evaluates how well a reliability summary index like alpha or omega represents the conditional reliability of a distribution of composite scores. It compares the conditional reliability function to a summary index and outputs a representativeness plot, a table of representativeness indices,and the full conditional reliability table for each possible sum score.

Usage

```
RelRep(
  data,
  items = c(names(data)),
  rel = "alpha",
  missing = "NA",
  method = "CI",
  width = NULL,
  raw.low = NULL,
  raw.high = NULL
)
```

Arguments

`data`	The original data to which the model was applied.
`items`	Column names of the items on the scale being evaluated (entered as strings). If omitted, all variables in the data will be used.
`rel`	Reliability coefficient to analyze. Options are "alpha" (the default) or "omega".

`missing` The missing data indicator in the data. Not needed in R, only present to simply use of this function in a Shiny application.

`method` how the test interval is created. Options are "CI" (the default), "width", or "raw". "CI" uses a 95 "width" builds an interval using a predetermined relative distance from the reliability coefficient (e.g., .05 from alpha). "raw" builds an interval using a predetermined raw values (e.g., .70 to .90)

`width` Only required if method="width". Specifies a predetermined relative distance from the coefficient to each bound of the interval. The total width of the interval will be twice this value (e.g., if .05 is entered, the total interval width is .10 because it will span .05 above the coefficient and .05 below the coefficient)

`raw.low` Only required if method="raw". Manually specifies the lower bound of the test interval. Must be between 0 and 1.

`raw.high` Only required if method="raw". Manually specifies the upper bound of the test interval. Must be between 0 and 1.

Value

Conditional reliability table and Reliability Representativeness plot and table.

Author(s)

Daniel McNeish & Denis Dumas

Maintainer: Daniel McNeish <dmcneish@asu.edu>

Examples

```
# "Example" dataset has 12 items on a 1-5 Likert scale

#Example using the first 8 items in "Example" for testing coefficient alpha with a 95% Bayes credible interval
ex1<-RelRep(data=Example,
items=c("X1","X2","X3","X4","X5","X6","X7","X8"),
rel="alpha",
method="CI")

#Example using odd items in "Example" to build a test interval that is .05 above and below coefficient omega
ex2<-RelRep(data=Example,
items=c("X1","X3","X5","X7","X9","X11"),
rel="omega",
method="width",
width=.05)

#Example using even items in "Example" to specify a test interval for coefficient alpha between 0.70 and 0.80
ex3<-RelRep(data=Example,
items=c("X2","X4","X6","X8","X10","X12"),
rel="alpha",
method="raw",
raw.low=.70,
raw.high=.80)
```

Index