

Package ‘pinsearch’

September 8, 2026

Type Package

Title Specification Search for Partial Factorial Invariance

Version 0.1.6

Description Automate specification search for identifying noninvariant items in factorial invariance analyses, as described in Yoon & Millsap (2007) <[doi:10.1080/10705510701301677](https://doi.org/10.1080/10705510701301677)>. Models are specified in 'lavaan' syntax. Supports continuous indicators with loadings, intercepts, residuals (uniqueness), and residual covariances invariance, and ordered categorical indicators with loadings, thresholds, and residual covariances invariance.

License GPL-3

Encoding UTF-8

Imports lavaan, stats

Suggests boot, difR, knitr, MASS, rmarkdown, testthat (>= 3.0.0), withr

URL <https://github.com/marklh/pinsearch>,
<https://marklh.github.io/pinsearch/>

BugReports <https://github.com/marklh/pinsearch/issues>

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 3.5.0)

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Mark Lai [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-9196-7406>>)

Maintainer Mark Lai <marklh@gmail.com>

Repository CRAN

Date/Publication 2026-09-08 13:40:07 UTC

Contents

dmacs	2
es_lavaan	4
fdr_alpha	5
fmacs	6
lui_sim	8
pinSearch	9
type2op	11
Index	12

dmacs	<i>Compute dMACS effect size described in Nye & Drasgow (2011) for two groups.</i>
-------	--

Description

dmacs returns the dMACS effect size statistics given a set of loadings and intercepts.

Usage

```
dmacs(  
  intercepts,  
  loadings = NULL,  
  pooled_item_sd = NULL,  
  latent_mean = 0,  
  latent_sd = 1,  
  uniqueness = NULL,  
  ns = NULL,  
  item_weights = NULL  
)  
  
dmacs_ordered(  
  thresholds,  
  loadings,  
  thetas = 1,  
  link = c("probit", "logit"),  
  pooled_item_sd = NULL,  
  latent_mean = 0,  
  latent_sd = 1,  
  item_weights = NULL  
)
```

Arguments

- intercepts A $2 \times p$ matrix of measurement intercepts.
- loadings A $2 \times p$ matrix of factor loadings, where p is the number of items.

pooled_item_sd	A numeric vector of length p of the pooled standard deviation (SD) of the items across groups.
latent_mean	latent factor mean for the reference group. Default to 0.
latent_sd	latent factor SD for the reference group. Default to 1.
uniqueness	A vector of length p of uniqueness.
ns	A vector of length p of sample sizes.
item_weights	Default is NULL. Otherwise, one can specify a vector of length p of weights; if so, test-level dMACS will be computed.
thresholds	A matrix with two rows for measurement thresholds. The matrix must have column names indicating to which item index each column corresponds.
thetas	Not currently used.
link	Link function for the model (probit or logit).

Details

The d_{MACS} effect size is defined as (Nye & Drasgow, 2011, p. 968)

$$d_{\text{MACS},i} = \frac{1}{SD_{iP}} \sqrt{\int [(\nu_{iR} - \nu_{iF}) + (\lambda_{iR} - \lambda_{iF})\eta]^2 f(\eta) d\eta}$$

where λ is the loading and ν is the intercept, F and R denote the focal and the reference group. The effect size reflects the standardized mean difference on an item due to measurement noninvariance, and is analogous to the Cohen's d effect size.

Value

A $1 \times p$ matrix of dMACS effect size. If `item_weights` is not NULL, $p = 1$.

References

Nye, C. & Drasgow, F. (2011). Effect size indices for analyses of measurement equivalence: Understanding the practical importance of differences between groups. *Journal of Applied Psychology*, 96(5), 966-980.

Examples

```
lambdaf <- c(.8, .5, .7, .5)
lambdar <- c(.8, .5, .4, .6)
nuf <- c(0.1, 0, 0.2, 0)
nur <- c(0.2, 0, 0, 0)
dmacs(rbind(nuf, nur),
      loadings = rbind(lambdaf, lambdar),
      pooled_item_sd = c(1, 1, 1, 1),
      latent_mean = 0,
      latent_sd = 1)
dmacs(rbind(nuf, nur),
      loadings = rbind(lambdaf, lambdar),
      pooled_item_sd = c(1, 1, 1, 1),
```

```

    latent_mean = 0,
    latent_sd = 1,
    item_weights = c(1, 1, 1, 1))
# Thresholds
lambda <- rbind(c(.8, .5, .7, .5),
               c(.8, .5, .4, .6))
tau <- rbind(c(-0.5, 0, 1, -0.3, 0.1, 0.5, -0.5, 1.5),
            c(-0.5, 0, 1, -0.5, 0.3, 0.5, -1, 1.5))
# three thresholds for items 1 and 2; one threshold for items 3 and 4
colnames(tau) <- c(1, 1, 1, 2, 2, 2, 3, 4)
dmacs_ordered(tau,
              loadings = lambda,
              pooled_item_sd = c(1, 1, 1, 1),
              latent_mean = 0,
              latent_sd = 1)

```

es_lavaan

Item-level effect size for non-invariance

Description

For two groups, the function uses `dmacs()` to compute d_{MACS} . For more than two groups, the function uses `fmacs()` to compute f_{MACS} , a generalisation of d_{MACS} similar to the Cohen's f effect size.

Usage

```

es_lavaan(object, ...)

pin_effsize(object, ...)

```

Arguments

<code>object</code>	A CFA model of class <code>lavaan::lavaan</code> fitted by <code>lavaan::cfa()</code>
<code>...</code>	Additional arguments passed to <code>dmacs()</code> or <code>fmacs()</code>

Value

A matrix of 1 row showing the effect size values for each non-invariant item on each latent variable.

fdr_alpha	<i>Adjusted alpha level for multiple testing.</i>
-----------	---

Description

Computes the adjusted significance level at step i of m tests in the sequential selection procedure of Benjamini & Gavrilov (2009), as used by `pinSearch()` when `control_fdr = TRUE`:

Usage

```
fdr_alpha(i, m, q = 0.05)
```

Arguments

i	Step index (rank) of the current test in the sequential search.
m	Total number of tests (maximum number of constraints that can be freed).
q	Target false discovery rate. Default is .05.

Details

$$\frac{iq}{m + 1 - i(1 - q)}$$

where i is the step index in the search, m is the maximum number of constraints that can be freed, and q is the target significance level.

Note that i is defined for $1 \leq i \leq m$, with $m \geq 1$ and $0 < q < 1$. These bounds are maintained by the internal logic of `pinSearch()`, but external callers must ensure them on their own.

Value

A numeric scalar, the adjusted alpha level at step i .

References

Benjamini, Y. & Gavrilov, N. M. (2009). Sequential selection procedures for testing dependent hypotheses.

Examples

```
fdr_alpha(1, 10)
```

fmacs

*Compute fMACS effect size for two or more groups.***Description**

fmacs returns the fMACS effect size statistics given a set of loadings and intercepts.

Usage

```
fmacs(
  intercepts,
  loadings = NULL,
  pooled_item_sd,
  num_obs = NULL,
  weights = 0 * intercepts + 1,
  group_factor = NULL,
  contrast = contr.sum(nrow(intercepts)),
  latent_mean = 0,
  latent_sd = 1,
  item_weights = NULL
)
```

```
fmacs_ordered(
  thresholds,
  loadings,
  thetas = 1,
  num_obs = NULL,
  weights = 0 * loadings + 1,
  group_factor = NULL,
  contrast = contr.sum(nrow(thresholds)),
  link = c("probit", "logit"),
  pooled_item_sd = NULL,
  latent_mean = 0,
  latent_sd = 1,
  item_weights = NULL
)
```

Arguments

intercepts	A $G \times p$ matrix of measurement intercepts.
loadings	A $G \times p$ matrix of factor loadings, where p is the number of items and G is the number of groups.
pooled_item_sd	A numeric vector of length p of the pooled standard deviation (SD) of the items across groups.
num_obs	A vector of length G of sample sizes. If not NULL, the weights will be proportional to sample sizes, assuming the same weights across items.

weights	A $G \times p$ matrix of weights. Default assumes equal weights across groups.
group_factor	A vector of length G indicating grouping for contrast. For example, <code>c(1, 1, 2)</code> means contrasting Group 1 & 2 vs. Group 3. The default is to not combine any groups, meaning the omnibus effect is computed.
contrast	A $p \times k$ contrast matrix where <code>colSums(contrast) = 0</code> . Default is <code>contr.sum(p)</code> if <code>group_factor</code> is not specified.
latent_mean	latent factor mean for the reference group. Default to 0.
latent_sd	latent factor SD for the reference group. Default to 1.
item_weights	Default is NULL. Otherwise, one can specify a vector of length p of weights; if so, test-level fMACS will be computed.
thresholds	A matrix with G rows for measurement thresholds. The matrix must have column names indicating to which item index each column corresponds.
thetas	Not currently used.
link	Link function for the model (probit or logit).

Details

The f_{MACS} effect size is defined as

$$f_{\text{MACS},i} = \frac{1}{SD_{iP}} \sqrt{\int [(\nu_{ij} - \bar{\nu}_j) + (\lambda_{ij} - \bar{\lambda}_j)\eta]^2 f(\eta) d\eta}$$

where λ is the loading and ν is the intercept, and j indexes group. The effect size reflects the square root of the ratio between the variance in observed item score due to measurement noninvariance and the variance of the observed item scores. f_{MACS} is analogous to the Cohen's f effect size. When there are two groups with equal sample sizes, $f_{\text{MACS}} = f_{\text{MACS}} / 2$

Value

A 1 x p matrix of fMACS effect size.

Examples

```
lambda <- rbind(c(.7, .8, .7, .9),
               c(.7, .8, .7, .8),
               c(.8, .7, .7, .5))
nu <- rbind(c(0, .5, 0, 1),
            c(0, .2, 0, 1.1),
            c(0, .3, 0, 1.2))
fmacs(nu,
      loadings = lambda,
      pooled_item_sd = c(1, 1, 1, 1),
      latent_mean = 0,
      latent_sd = 1)
# With contrast (Group 1 & 2 vs. Group 3)
fmacs(nu,
      loadings = lambda,
      pooled_item_sd = c(1, 1, 1, 1),
```

```

      group_factor = c(1, 1, 2),
      latent_mean = 0,
      latent_sd = 1)
# Thresholds
lambda <- rbind(c(.8, .5, .7, .5),
                c(.8, .5, .4, .6),
                c(.8, .7, .7, .5))
tau <- rbind(c(-0.5, 0, 1, -0.3, 0.1, 0.5, -0.5, 1.5),
             c(-0.5, 0, 1, -0.5, 0.3, 0.5, -1, 1.5),
             c(-0.5, 0, 1, -0.5, 0.3, 0.5, -1, 0.5))
# three thresholds for items 1 and 2; one threshold for items 3 and 4
colnames(tau) <- c(1, 1, 1, 2, 2, 2, 3, 4)
fmacs_ordered(tau,
               loadings = lambda,
               pooled_item_sd = c(1, 1, 1, 1),
               latent_mean = 0,
               latent_sd = 1)
# With contrast (Group 1 & 2 vs. Group 3)
fmacs_ordered(tau,
               loadings = lambda,
               pooled_item_sd = c(1, 1, 1, 1),
               group_factor = c(1, 2, 1),
               latent_mean = 0,
               latent_sd = 1)

```

lui_sim

Simulated data with 15 items of an Alcohol Belief Scale

Description

Simulation based on fitting a configural invariance model to data made public by Lui (2019, <https://psycnet.apa.org/record/2019-08221-001>). The original data can be found at <https://osf.io/wxjsg>.

Usage

```
lui_sim
```

Format

lui_sim:

A data frame with 910 rows and 16 columns:

class1 to class15 Binary items

group six groups: 1 = White Male, 2 = Asian Male, 3 = Hispanic Male, 4 = White Female, 5 = Asian Female, 6 = Hispanic Female

Source

<https://osf.io/wxjsg>

pinSearch

*Search for noninvariant parameters across groups.***Description**

The function implements the sequential selection method similar to that discussed in Yoon and Millsap (2007) [doi:10.1080/10705510701301677](https://doi.org/10.1080/10705510701301677). The function proceeds in the order of metric, scalar (threshold), and strict invariance. In each stage, invariance constraints in all items, and the constraint associated with the biggest test statistic above a predefined threshold is freed, before recomputing the test statistic for the next constraint to free.

Usage

```
pinSearch(
  config_mod,
  ...,
  type = c("loadings", "intercepts", "thresholds", "residuals", "residual.covariances"),
  inv_test = c("mod", "score", "lrt"),
  sig_level = 0.05,
  control_fdr = FALSE,
  min2 = FALSE,
  effect_size = FALSE,
  progress = FALSE
)
```

Arguments

config_mod	Syntax of a configural invariance model to be passed to <code>lavaan::cfa()</code> .
...	Additional arguments passed to <code>lavaan::cfa()</code> .
type	Character variable indicating the stage of invariance to be searched. Currently supported options are (a) for continuous indicators, "loadings", "intercepts", "residuals", and "residual.covariances", and (b) "loadings", "thresholds", "residual.covariances", in an increasingly strict order. A stricter model (e.g., "residual.covariances") will have constraints of all previous stages.
inv_test	Character variable indicating the statistical test to be used for specification search. Currently supported options are "mod" for modification index using <code>lavaan::modindices()</code> , "score" for score test statistic using <code>lavaan::lavTestScore()</code> , and (experimental) "lrt" for likelihood ratio test statistic using <code>lavaan::lavTestLRT()</code> .
sig_level	Significance level used to determine whether the parameter associated with the highest modification index should be removed. Default is .05.
control_fdr	Logical; whether to use adjust for false discovery rate for multiple testing. If TRUE, the method by Benjamini & Gavrilov (2009) will be used.
min2	Logical; whether to require at least two invariant items when searching for non-invariance.

effect_size	Logical; whether to compute dmacs (two groups) or fmacs (> two groups) effect size or not (default). This is an experimental feature.
progress	Logical; an experimental feature of showing a progress bar if TRUE. Because the number of steps is unknown until the stopping criteria are reached, the progress bar may be inaccurate.

Details

Note that when an item has a non-invariant loading, the corresponding intercept constraint will automatically be freed, as intercept difference across groups is sensitive to the location of the zero point for the latent variable and the item.

For a particular stage of invariance constraints, the Benjamini & Gavrilov method uses an adjusted *alpha* level of

$$iq/[m + 1 - i(1 - q)]$$

where i is the step index in the search, m is the maximum number of constraints that can be freed, and q is the desirable significance level.

Under lavaan ≥ 0.7 , ordinal (categorical) models in which thresholds are tied across groups can yield a degenerate stage test (a likelihood ratio test with $Df \text{ diff} = 0$). `pinSearch()` identifies the threshold stage (and any later stage, but not the configural or loadings stages) with the item intercepts fixed at 0 and the latent mean fixed in the first group only (free in the remaining groups), so threshold invariance remains testable, including for 2-group binary data. Any other degenerate stage test falls back to the per-parameter test, so results for ordinary multi-category and 3-group ordinal data are unchanged; continuous models are not affected.

Value

A list of three elements:

- Partial Invariance Fit: A `lavaan::lavaan` object containing the final partial invariance model.
- Non-Invariant Items: A data frame of non-invariant parameters.
- effect_size: Effect size statistics obtained from `pin_effsize()`.

References

Yoon, M., & Millsap, R. E. (2007). Detecting violations of factorial invariance using data-based specification searches: A Monte Carlo study. *Structural Equation Modeling: A Multidisciplinary Journal*, 14(3), 435-463.

Examples

```
library(lavaan)
# Simulate random data
set.seed(14) # for reproducible results
mod_ninv <- "f =~ c(1, 0.8, .6)*y1 + c(0.8, 1.2, 1.0)*y2"
mod_inv <- paste0("1*y", 3:6, collapse = " + ")
sim_mod <- paste(
  paste(mod_ninv, "+", mod_inv),
  "f =~ c(1, 1.3, 1.5)*f"
```

```

      f ~ c(0, 0.5, 1.0)*1
      y1 ~ c(0, 0.3, 0)*1
      y3 ~ c(0.3, 0, -0.3)*1
      y1 ~~ c(1, .5, 1)*y1",
      sep = "\n"
    )
    # The uniqueness of each item is assumed to be 1.0
    dat_sim <- simulateData(sim_mod, sample.nobs = c(100, 100, 100))
    # Fit configural model:
    sam_config <- paste(
      paste0("f =~ ", paste0("y", 1:6, collapse = " + "))
    )
    pinSearch(sam_config,
      data = dat_sim, group = "group",
      type = "intercepts"
    )

```

type2op

Map invariance types to lavaan parTable operators.

Description

Maps the invariance type strings used by `pinSearch()` to the corresponding values of the `op` column of a lavaan parameter table (`lavaan::parTable()`): "loadings" maps to "=", "intercepts" to "~1", "thresholds" to "|", and both "residuals" and "residual.covariances" to "~".

Usage

```
type2op(type)
```

Arguments

type A character string indicating the invariance type. Must be one of "loadings", "intercepts", "thresholds", "residuals", or "residual.covariances".

Value

A character string, the corresponding lavaan operator.

Examples

```
type2op("loadings")
```

Index

* datasets

 lui_sim, [8](#)

dmacs, [2](#)

dmacs(), [4](#)

dmacs_ordered (dmacs), [2](#)

es_lavaan, [4](#)

fdr_alpha, [5](#)

fmacs, [6](#)

fmacs(), [4](#)

fmacs_ordered (fmacs), [6](#)

lavaan::cfa(), [4](#), [9](#)

lavaan::lavaan, [4](#), [10](#)

lavaan::parTable(), [11](#)

lui_sim, [8](#)

pin_effsize (es_lavaan), [4](#)

pin_effsize(), [10](#)

pinSearch, [9](#)

pinSearch(), [5](#), [11](#)

type2op, [11](#)