Package: semTests (via r-universe)

October 29, 2026

Type Package

Title Robust Test Statistics for Structural Equation Models

Description Computes robust p-values for overall fit and nested comparisons of structural equation models fitted with 'lavaan'. Implements penalized eigenvalue block averaging and penalized regression (Foldnes, Moss, Grønneberg, 2025) <doi:10.1080/10705511.2024.2372028>, including their extension to nested models (Foldnes, Grønneberg, Moss, 2026) <doi:10.3758/s13428-026-02968-4>, alongside familiar corrections such as Satorra-Bentler. Supported settings include complete-data ML, GLS, and ULS, categorical DWLS and ULS, and full-information maximum likelihood with one or several groups. Score and criterion-difference tests of parameter restrictions are also available, including restrictions in misspecified models.

Version 2.0.0

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/JonasMoss/semTests

Depends R (>= 3.5.0)

Imports lavaan (>= 0.7-3), methods, pbivnorm

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Jonas Moss [aut, cre] (ORCID: <https://orcid.org/0000-0002-6876-6964>), Njål Foldnes [ctb] (ORCID: <https://orcid.org/0000-0001-6889-6067>), Steffen Grønneberg [ctb] (ORCID: <https://orcid.org/0000-0003-2785-6530>)

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

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

Date/Publication 2026-10-29 08:40:02 UTC

RemoteUrl https://github.com/cran/semTests

RemoteRef HEAD

RemoteSha d777c2b55e749b4feab83c521d719336ca83a7aa

Contents

inference_control 2
print.semTests_pvalues 3
pvalues 4
semtest 8
semTests-support 10

Index **14**

`inference_control` *Computation settings for* `semtest()`

Description

Collects the convention choices used when building a test. These are deliberately separated from `semtest()`'s own arguments: the signature states the question, the control states how the matrices are formed.

Usage

```
inference_control(
  curvature = NULL,
  score_covariance = NULL,
  weight_effect = c("estimated", "fixed"),
  derivative_step = 1e-04,
  tolerance = 1e-10,
  keep = FALSE
)
```

Arguments

`curvature`	`"observed"` or `"expected"` sensitivity. `NULL` defers to the test. Expected and observed agree in probability when the model is correctly specified, and differ otherwise.
`score_covariance`	`"centered"` or `"uncentered"` empirical score covariance. `NULL` defers to the test. Centered subtracts group score means and stays valid when the score has nonzero mean; uncentered is consistent only at exact fit. The non-robust reference, which assumes the information equality, is `pvalues(x, "STD")`.

`weight_effect`	`"estimated"` propagates first-stage weight estimation into the covariance; `"fixed"` treats the weights as known. Ignored for ML.
`derivative_step`	Relative central-difference step for numerical criterion curvature. Must lie in (0, .01].
`tolerance`	Relative tolerance for matrix symmetry and definiteness validation. Must lie in (0, 1).
`keep`	Retain the input matrices and projections in the result.

Details

`curvature` and `score_covariance` default to `NULL`, meaning each test picks the convention appropriate to the null it evaluates. Goodness of fit assumes the model holds, since that is its hypothesis, and uses expected curvature with uncentered score products. Nested restriction tests allow the baseline model to be wrong and use observed curvature with centered score products. Supplying a value overrides the choice everywhere it applies.

The resolved settings are recorded in the returned test's provenance, so a result always reports the conventions that produced it.

Value

An object of class `semTests_control`.

Examples

```
inference_control(curvature = "observed")
inference_control(score_covariance = "centered", keep = TRUE)
```

`print.semTests_pvalues` *Print method for p-values from* `pvalues()` / `pvalues_nested()`.

Description

Prints the p-values, then a one-line provenance footer (estimator, data type, information, df) recording the options used.

Usage

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

Arguments

`x`	A `semTests_pvalues` object.
`...`	Passed to the default print method.

Value

x, invisibly.

`pvalues` *Compute robust* p*-values for fitted models and test objects*

Description

Compute one or several *p*-values for a fitted `lavaan` model. Available methods include penalized eigenvalue block averaging, penalized regression, and familiar robust corrections. The defaults select the recommended method for the fitted model.

Usage

```
pvalues(object, tests, fiml.convention = c("observed", "lavaan"))

## S3 method for class 'lavaan'
pvalues(
  object,
  tests = if (is_classic_nt(object)) "pEBA4_RLS" else "pEBA4",
  fiml.convention = c("observed", "lavaan")
)

pvalues_nested(
  m0,
  m1,
  method = c("2000", "2001"),
  tests = if (is_classic_nt(m0)) "PALL_UG_ML" else "PALL",
  A.method = c("weighted", "delta", "exact"),
  fiml.convention = c("observed", "lavaan")
)

## S3 method for class 'semTests_test'
pvalues(
  object,
  tests = if (object$reference == "chisq") "STD" else "ALL",
  fiml.convention = c("observed", "lavaan")
)
```

Arguments

`object`, `m0`, `m1`	One or two `lavaan` objects, or a `semTests_test` result from `semtest()` supplied as `object` to `pvalues()`. For lavaan fits, `pvalues` does goodness-of-fit testing on one object, while `pvalues_nested` does hypothesis testing on two nested models.

`tests` A non-empty character vector of tests. Each element uses one of `TEST`, `TEST_UG`, `TEST_ML`, `TEST_RLS`, `TEST_UG_ML`, or `TEST_UG_RLS`. The defaults are the recommended options. `EBA` and `pEBA` take a positive integer number of blocks (for example, `"pEBA4"`), no greater than the test df. `pOLS` takes a finite positive penalty (for example, `"pOLS2"`).

`fiml.convention` Deprecated, and will be removed in a future release. For FIML fits, use the fully observed-information convention (`"observed"`, the default) or reproduce lavaan's inspected robust-test spectrum (`"lavaan"`). Supplying it, with either value, gives a warning of class `semTests_warning_deprecated`. The `"lavaan"` spectrum changes with the fit's `information` and `h1.information` options; use `lavaan::lavTest()` or `lavaan::lavTestLRT()` to reproduce lavaan's own tests.

`method` Nested reduction method. Only Satorra's `"2000"` construction (the paper-recommended default) is available. The Satorra–Bentler `"2001"` construction has been withdrawn because it performs poorly, and requesting it points you to `"2000"`.

`A.method` The restriction matrix of the nested test. `"weighted"` (the default) is the moment-tangent map, with each model's Jacobian at its own estimate, projected in the fit's weight metric. It applies to any nesting, including restricted models that drop or fix paths, and does not depend on how the factors are identified or on the units of the observed variables. `"delta"` is the same map projected in the Euclidean metric, lavaan's default; use it to reproduce `lavTestLRT(..., method = "satorra.2000")`. Its p-values change with the units of the observed variables. `"exact"` is lavaan's literal parameter restriction map. It requires the same underlying free parameters in both models, as with equality constraints, and its p-values shift slightly with the identification (for example `std.lv = TRUE` versus marker loadings). Categorical nested models do not support `"exact"`.

Details

`pvalues()` also accepts the test objects returned by `semtest()`. Tests with a weighted chi-square reference use `"ALL"` by default; fully studentized tests (`weight = "score_covariance"`) have a chi-square reference and accept only `"STD"`. Test objects accept calibration names without `UG`, `ML`, or `RLS` suffixes, and do not accept `fiml.convention`. The lavaan methods retain their existing defaults and output names.

The `tests` argument is a character vector. Each element has one of the forms `TEST`, `TEST_UG`, `TEST_ML`, `TEST_RLS`, `TEST_UG_ML`, or `TEST_UG_RLS`. For example, `SB_UG_RLS`.

- The first part names the test. The available names are listed below.
- Add `UG` to use the unbiased estimator of the fourth-order moment matrix from Du and Bentler (2022). Leave it out to use the standard biased matrix. Performance depends on the data-generating process, so neither gamma choice dominates in every setting.
- The final part chooses the chi-square statistic. `ML` uses the normal-discrepancy statistic (Bollen, 1989). `RLS` uses Browne's (1974) reweighted least squares statistic.

The `peba` method is the recommended default. It partitions the eigenvalues into `j` equally sized sets (if not possible, the smallest set is incomplete), shrinks them towards their common mean, and averages within each set. Provide a positive integer `j` no greater than the test df. Values

from 2 through 6 are usually good candidates. The method was introduced by Foldnes, Moss, and Grønneberg (2025).

pols is a penalized regression method with a finite positive penalization term. Foldnes, Moss, and Grønneberg (2025) studied pols=2, which has good performance in a variety of contexts.

pall penalizes all eigenvalues in ugamma, while all uses all eigenvalues without penalization. pall is the recommended option for nested models, for which the penalized methods were extended and evaluated by Foldnes, Grønneberg, and Moss (2026).

The eba method is the unpenalized predecessor of peba (Foldnes and Grønneberg, 2018). It averages within the eigenvalue blocks without shrinkage. peba usually performs better. eba remains available for comparisons, and values from j=2 through j=4 tend to work best.

Familiar corrections are available too:

- std uses the ordinary chi-square reference distribution.
- sb gives the Satorra–Bentler p-value (Satorra and Bentler, 1994).
- ss gives the scaled and shifted p-value (Asparouhov and Muthén, 2010).
- sf gives the scaled F p-value (Wu and Lin, 2016).

Estimators and data types:

semTests-support (?semTests-support) gives the complete matrix of supported estimators, data types, and configurations. Here is the short version.

The limiting null law of the test statistic is a weighted sum of chi-squares for any minimum-discrepancy estimator. pvalues() supports ML/MLM/MLR, GLS, ULS, FIML (missing data), and categorical DWLS/ULS families, with single- and multi-group continuous, ordered, and mixed-indicator fits. pvalues_nested() supports continuous estimators and categorical Satorra-2000 comparisons with a moment-space restriction map. The fitted model must expose the asymptotic moment covariance. Robust choices such as test = "satorra.bentler" or estimator = "MLM", "MLR", or "DWLS" provide it. The RLS statistic (browne.residual.nt.model) and the unbiased (UG) Du-Bentler gamma are defined for the classical continuous, complete-data, random-x ML case. Other families use the standard statistic and biased gamma. Fixed or conditional observed exogenous predictors are currently refused. Full WLS/ADF is refused outright: its weight is already the inverse moment covariance, so the correction is exactly the identity and every test would equal the ordinary chi-square.

GLS, ULS, categorical DWLS/ULS, FIML missing data, and nested FIML or categorical comparison are validated to numerical tolerance against an independent implementation and are marked stable; see the Stability section in semTests-support.

Both entry points require a converged fit and warn if lavaan reports an inadmissible solution. Nested comparisons require the same requested estimator, fitting conventions, variables, groups, sample sizes, raw data, and missingness mask. semTests checks comparability. The substantive nesting argument remains part of the analysis.

For FIML, semTests uses observed saturated and model information throughout. An independent implementation validates this construction. The deprecated fiml.convention argument can instead select "lavaan", which reproduces lavaan's robust-test construction. For a single model it uses lavaan's inspected UGamma. For nested models it uses lavaan's H1 weight convention and selected model information. The returned object records the convention, which keeps this inferential choice visible in saved results.

For a single FIML model, `"lavaan"` means the eigenvalue spectrum returned by `lavInspect(fit, "UGamma")`. The scalar Yuan–Bentler–Mplus test stored by a default MLR fit is a different correction and may give a different result. For nested FIML models, `"lavaan"` with `A.method = "delta"` reproduces `lavTestLRT(..., method = "satorra.2000")`.

Value

A named numeric vector of p-values, of class `semTests_pvalues`, carrying an `"semtests"` attribute that records the requested tests, requested and base estimator, base-statistic and gamma choices, information type, data type, parameterization where applicable, nested method and restriction map where applicable, and degrees of freedom. For matrix results, the attribute instead records test type, reference law, hypothesis, normalization, rank/df, sample size when supplied, and provenance.

References

Foldnes, N., Moss, J., & Grønneberg, S. (2025). Improved goodness of fit procedures for structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, 32(1), 1–13. doi:10.1080/10705511.2024.2372028

Foldnes, N., Grønneberg, S., & Moss, J. (2026). Penalized eigenvalue block averaging: Extension to nested model comparison and Monte Carlo evaluations. *Behavior Research Methods*, 58, article 107. doi:10.3758/s13428026029684

Satorra, A. (2000). Scaled and adjusted restricted tests in multi-sample analysis of moment structures. In R. D. H. Heijmans, D. S. G. Pollock, & A. Satorra (Eds.), *Innovations in Multivariate Statistical Analysis* (pp. 233–247). Kluwer Academic. doi:10.1007/9781461546030_17

Satorra, A., & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika*, 66(4), 507–514. doi:10.1007/BF02296192

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent Variables Analysis: Applications for Developmental Research* (pp. 399–419). Sage.

Asparouhov, T., & Muthén, B. O. (2010). *Simple second order chi-square correction*. Mplus Technical Appendix. https://www.statmodel.com/download/WLSMV_new_chi21.pdf

Wu, H., & Lin, J. (2016). A Scaled F Distribution as an Approximation to the Distribution of Test Statistics in Covariance Structure Analysis. *Structural Equation Modeling*. doi:10.1080/10705511.2015.1057733

Foldnes, N., & Grønneberg, S. (2018). Approximating Test Statistics Using Eigenvalue Block Averaging. *Structural Equation Modeling*, 25(1), 101–114. doi:10.1080/10705511.2017.1373021

Du, H., & Bentler, P. M. (2022). 40-Year Old Unbiased Distribution Free Estimator Reliably Improves SEM Statistics for Nonnormal Data. *Structural Equation Modeling: A Multidisciplinary Journal*, 29(6), 872–887. doi:10.1080/10705511.2022.2063870

Kenward, M. G., & Molenberghs, G. (1998). Likelihood based frequentist inference when data are missing at random. *Statistical Science*, 13(3), 236–247. doi:10.1214/ss/1028905886

Bollen, K. A. (1989). *Structural Equations with Latent Variables*. John Wiley & Sons. doi:10.1002/9781118619179

Browne, M. W. (1974). Generalized least squares estimators in the analysis of covariance structures. *South African Statistical Journal*, 8, 1–24.

See Also

semTests-support for the full list of supported configurations.

Examples

```
library("semTests")
library("lavaan")
model <- "visual  =~ x1 + x2 + x3
          textual =~ x4 + x5 + x6
          speed   =~ x7 + x8 + x9"
object <- cfa(model, HolzingerSwineford1939, estimator = "MLM")
pvalues(object)

# For the pEBA6 method with biased gamma and ML chisq statistic:
pvalues(object, "pEBA6_ML")

# Nested model comparison (constrain the textual loadings to be equal):
constrained <- "visual  =~ x1 + x2 + x3
                textual =~ a*x4 + a*x5 + a*x6
                speed   =~ x7 + x8 + x9"
m1 <- cfa(model, HolzingerSwineford1939, estimator = "MLM")
m0 <- cfa(constrained, HolzingerSwineford1939, estimator = "MLM")
pvalues_nested(m0, m1)
```

---

semtest *Build a score or criterion test*

Description

Constructs an inspectable test object for overall fit or for a nested restriction. Pass the result to `pvalues()` for calibrated p-values.

Usage

```
semtest(
  object,
  alternative = NULL,
  statistic = c("score", "criterion"),
  weight = c("curvature", "expected_curvature", "score_covariance"),
  control = inference_control(),
  ...
)
```

Arguments

`object`	A converged `lavaan` fit. For a restriction test this is the restricted model.
`alternative`	A nested alternative fit or unoptimized template, using the same observations, estimator and moment conventions. `NULL` tests overall fit.

statistic	"score" or "criterion". "criterion" requires alternative.
weight	Which matrix weights the quadratic form, and therefore which statistic is computed. This is a separate decision from curvature, which selects the sensitivity used to project out nuisance directions and so enters the reference law. • "curvature" weights by whichever sensitivity curvature selected. • "expected_curvature" weights by expected curvature even when curvature = "observed". This is the combination that keeps observed sensitivity in the projection, which is what the limit requires when the model may be misspecified, while weighting by the more stable expected matrix. • "score_covariance" studentizes fully and gives a chi-square reference. The first two give a weighted chi-square reference, so the calibrations in pvalues() apply. Ignored when statistic = "criterion", which weights by curvature by construction.
control	Conventions from inference_control().
...	Individual inference_control() settings, merged into control. Unknown names are an error.

Details

Supply alternative to test a restriction, or leave it NULL to test overall fit against saturated means and covariances. The restricted model comes first, as in pvalues_nested(). The alternative may be created with do.fit = FALSE when statistic = "score": it supplies the parameterization and is evaluated at the restricted estimates without refitting.

statistic = "score" evaluates the score at the restricted fit and needs only that fit to be optimized. statistic = "criterion" uses the actual difference of the summed fitting criteria and needs both fits. Their null is a common population minimizer under the shared criterion, which allows the baseline model to be misspecified. That is a different null from pvalues_nested(), which tests exact fit of both models using Satorra's 2000 construction, so the two need not agree.

Overall fit with statistic = "criterion" is the classical discrepancy statistic, which is what pvalues() computes; semtest() does not duplicate it.

Value

A semTests_test, for use with pvalues(), print() and as.data.frame().

See Also

pvalues() for calibrated p-values and for the classical discrepancy statistics, inference_control() for the conventions.

Examples

```
model <- "f =~ x1 + a*x2 + b*x3 + x4"
d <- lavaan::HolzingerSwineford1939
fit <- lavaan::cfa(model, d, meanstructure = TRUE)

# Overall fit, score statistic.
```

```
pvalues(semtest(fit), c("SB", "ALL"))

# Can the equality be released?
restricted <- lavaan::cfa(paste(model, "a == b", sep = "\n"), d,
                          meanstructure = TRUE)
semtest(restricted, fit, statistic = "score", weight = "score_covariance")
semtest(restricted, fit, statistic = "criterion")
```

`semTests-support` *Supported estimators, data types, and configurations*

Description

Use this page to check what `pvalues()` and `pvalues_nested()` support. The eigenvalue-based p-values target a limiting null law that is a weighted sum of chi-squares. This result holds for any minimum-discrepancy estimator and therefore includes estimators beyond normal-theory ML. The tables below give the supported combinations. Configurations outside them are refused at the entry point (see `check_supported()`), and the test suite follows the same boundary. The score and criterion tests built by `semtest()` have the narrower scope described separately below; the fit-test tables do not establish their support.

Details

Single-model (`pvalues()`)**:**

Estimator	Data	Groups	Missing	Availability	Stability
ML / MLM / MLR	continuous	single or multi	complete	available	stable
GLS	continuous	single or multi	complete	available	stable
ULS	continuous	single or multi	complete	available	stable
ML / MLR (FIML)	continuous	single or multi	FIML	available	stable
DWLS family	ordered/mixed	single or multi	listwise/pairwise	available	stable
ULS family	ordered/mixed	single or multi	listwise/pairwise	available	stable
WLS (ADF)	continuous	single or multi	complete	rejected	–
WLS (ADF)	ordered/mixed	single or multi	listwise/pairwise	rejected	–

Nested (`pvalues_nested()`)**:**

Estimators	Data	Groups	Missing	Method	A.method	Avai
ML/MLM/MLR	continuous	single or multi	complete	2000	weighted, delta, or exact	avail
GLS, ULS	continuous	single or multi	complete	2000	weighted, delta, or exact	avail
ML / MLR (FIML)	continuous	single or multi	FIML (both fits)	2000 only	weighted, delta, or exact	avail
DWLS/ULS families	ordered/mixed	single or multi	listwise/pairwise	2000 only	weighted or delta	avail
any	continuous	–	mixed / non-FIML	–	–	rejec

Stability:

The classical normal-theory ML path (continuous, complete data) is the mature, paper-backed core. The other supported estimators – GLS, ULS, categorical DWLS/ULS fits, FIML missing-data fits, and their nested comparisons – are also marked stable: each is validated to numerical tolerance against an independent implementation (magmaan) across a simulation battery, in addition to the deterministic implementation tests every available row carries. The deprecated `fiml.convention = "lavaan"` option reproduces lavaan's own spectrum for compatibility rather than as an independently validated construction. Configurations outside the tables are refused at the entry point.

Observed exogenous covariates:

Observed exogenous predictors are supported when they are modeled jointly with the other variables. In lavaan, request this random-x analysis with `fixed.x = FALSE` and `conditional.x = FALSE`. The independent validation suite covers continuous ML, GLS, ULS, FIML, and mixed categorical DWLS in this setting, for single models and nested comparisons.

Fixed or conditional observed exogenous predictors are currently refused. Their reference distribution conditions on the realized covariate design and needs a separate saturated-model projection. If a covariate is intended to be fixed, keep that scientific choice and do not silently change it to random merely to pass the software check.

Fit quality and nested comparability:

Both entry points require converged lavaan fits and warn when lavaan's post-estimation check reports an inadmissible solution. Nested comparisons additionally require the same requested estimator, fitting conventions, observed variables, groups, sample sizes, raw data, and missingness mask. These checks establish comparability. The scientific argument that one model is genuinely nested within the other still belongs in the analysis.

The constrained model belongs in `m0`. If the models are supplied in reverse df order, semTests warns and swaps them. Only the Satorra-2000 reduction is available. `method = "2001"` has been withdrawn because it performs poorly.

Statistic and gamma options:

Test names use one of `TEST`, `TEST_UG`, `TEST_ML`, `TEST_RLS`, `TEST_UG_ML`, or `TEST_UG_RLS` (e.g. `"SB_UG_RLS"`). See `pvalues()` for the test families. Two of the options are defined only for the classical case:

- The **RLS** statistic (`browne.residual.nt.model`, Browne 1974) and the `UG` (Du-Bentler) unbiased gamma are available **only** for classical normal-theory ML with continuous, complete data, `estimator = "ML"`, `fixed.x = FALSE`, and `conditional.x = FALSE`. lavaan degrades RLS to ADF elsewhere, and the Du-Bentler correction has no derivation for the other families or a fixed covariate design. `semTests` refuses either request and points you to the standard statistic and biased gamma.
- For **complete-data normal-likelihood ML**, biased spectra can be computed from tiled empirical moment influences when Gamma is not cached. The calculation includes mean/covariance cross terms and group proportions. Cached or supplied NACOV is retained; sampling weights, alternative Gamma normalizations, and other first-stage conventions keep their existing paths.
- **Categorical and mixed-indicator fits** use the spectrum equivalent to lavaan's biased `UGamma` and its unscaled estimator statistic. Expected- information fits compute the spectrum in residual space. Supported estimator families are DWLS (including WLSMV/WLSM/WLSMVS)

and ULS (including ULSMV). Nested comparison is restricted to `method = "2000"`, with `A.method = "weighted"` or `"delta"`.

- **Pairwise categorical missingness** means support for lavaan's pairwise sample-statistic and `UGamma` calculation. Pairwise deletion has its own missingness assumptions and does not provide FIML or generally MAR-valid inference.
- **FIML** (missing data) uses the biased gamma and the standard statistic only. `UG` and `RLS` are refused. FIML supports one or several groups with continuous data. Observed exogenous predictors are supported under joint random-x inference, using `fixed.x = FALSE` and `conditional.x = FALSE`. semTests uses observed saturated and model information, following the recommendation of Kenward and Molenberghs (1998), and is independently validated against magmaan. The deprecated `fiml.convention = "lavaan"` reproduces lavaan's inspected robust-test spectrum, including its expected H1 weight.

Why full WLS/ADF is refused:

For full WLS (ADF), with continuous or categorical sample statistics, the fitting weight is already the inverse of the asymptotic moment covariance, so `UGamma` is a projector: its nonzero eigenvalues are all exactly one and the eigenvalue correction collapses to the identity. Every robust p-value would therefore equal the ordinary `1 - pchisq(chisq, df)`. Because the correction adds nothing, full WLS is refused at the entry point. Fit a DWLS/ULS family for a non-degenerate robust test, or read the standard chi-square directly.

Score and criterion tests (`semtest()`):

Without an alternative, `semtest()` builds a global exact-fit score test from Gaussian likelihood scores. It supports ML/MLM/MLR and FIML with continuous indicators, one or several groups. Least-squares estimators and categorical data are not supported for overall fit.

With an alternative, `semtest()` builds nested score and criterion-difference tests of affine releases at the restricted fit. These support ML/FIML, continuous GLS/WLS/DWLS/ULS, and complete all-ordered DWLS/WLS/ULS, with one or several groups. By default they use full observed curvature and the joint influence of the sample moments and the estimated fitting weights. Expected curvature and fixed weights, selected through `inference_control()`, are comparators rather than complete corrections under misspecification.

Both require unweighted single-level raw observations, joint random-x inference and affine equality constraints. Ordinal inference requires nonempty pairwise cells. Its first-stage influence and the derivative of the estimated weight are computed in closed form. Mixed indicators, pairwise missingness, case weights, clusters, bounds and nonlinear constraints are excluded. Full WLS is admitted for nested inference under misspecification; its exact-fit correction refusal in `pvalues()` is unchanged.

Every result is an inspectable `semTests_test` usable with `pvalues()`. Fully studentized results accept only their chi-square calibration. Numerical parity with an independent implementation does not establish the finite-sample calibration of penalized score procedures.

References

Foldnes, N., Moss, J., & Grønneberg, S. (2025). Improved goodness of fit procedures for structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, 32(1), 1–13. doi:10.1080/10705511.2024.2372028

Foldnes, N., Grønneberg, S., & Moss, J. (2026). Penalized eigenvalue block averaging: Extension to nested model comparison and Monte Carlo evaluations. *Behavior Research Methods*, 58, article 107. doi:10.3758/s13428026029684

Kenward, M. G., & Molenberghs, G. (1998). Likelihood based frequentist inference when data are missing at random. *Statistical Science*, 13(3), 236–247. doi:10.1214/ss/1028905886

Satorra, A. (2000). Scaled and adjusted restricted tests in multi-sample analysis of moment structures. In R. D. H. Heijmans, D. S. G. Pollock, & A. Satorra (Eds.), *Innovations in Multivariate Statistical Analysis* (pp. 233–247). Kluwer Academic. doi:10.1007/9781461546030_17

See Also

`pvalues()`, `pvalues_nested()`

Index

∗ **models**
semTests-support, 10

check_supported(), *10*

inference_control, 2
inference_control(), *9*, *12*

lavaan::lavTest(), *5*
lavaan::lavTestLRT(), *5*

print.semTests_pvalues, 3
pvalues, 4
pvalues(), *3*, *8–13*
pvalues.lavaan *(*pvalues*)*, 4
pvalues.semTests_test *(*pvalues*)*, 4
pvalues_nested *(*pvalues*)*, 4
pvalues_nested(), *3*, *9*, *10*, *13*

semtest, 8
semtest(), *4*, *5*, *10*, *12*
semTests-support, *6*, *8*, 10