Package 'lavinteract'

May 8, 2026

Type Package

Title Post-Estimation Utilities for 'lavaan' Fitted Models

Version 0.5.1

Description Companion toolbox for structural equation models fitted with 'lavaan'. Provides post-estimation diagnostics and graphics that operate directly on a fitted object using its estimates and covariance, and refits auxiliary models when needed. The package relies on 'lavaan' (Rosseel, 2012) <doi:10.18637/jss.v048.i02>.

URL https://github.com/g-corbelli/lavinteract

BugReports https://github.com/g-corbelli/lavinteract/issues

License GPL-3

Encoding UTF-8

Imports lavaan, rlang, ggplot2, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Language en-US

NeedsCompilation no

Maintainer Giuseppe Corbelli <giuseppe.corbelli@uninettunouniversity.net>

Config/testthat/edition 3

RoxygenNote 7.3.2

Author Giuseppe Corbelli [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2864-3548>)

Repository CRAN

Date/Publication 2026-04-28 14:20:02 UTC

Contents

`lavinteract` *Post-Estimation Utilities for 'lavaan' Fitted Models*

Description

Post-estimation tools for structural equation models fitted with 'lavaan'. Provides methods for probing observed and latent interactions, diagnosing local misfit and multicollinearity, quantifying incremental effect sizes for structural predictors, assessing predictive performance by repeated holdout cross-validation, and adjusting selected parameter p-values for multiple testing. Functions operate from a fitted model object and, when needed, refit auxiliary or reduced models while preserving the original SEM specification.

Details

The functions are:

- `lav_slopes`: simple slopes and interaction plots from a fitted 'lavaan' model.
- `lav_jn`: Johnson-Neyman regions of significance for continuous moderators in a fitted 'lavaan' model.
- `lav_deltaR2`: incremental effect sizes (part R^2 and Cohen's f^2) for structural predictors via reduced-model comparisons.
- `lav_localfit`: residual-based local fit diagnostics and heatmaps for fitted 'lavaan' models.
- `lav_vif`: variance inflation factors for structural predictors with measurement preserved.
- `lav_cv`: repeated holdout (Monte Carlo) cross-validation of R^2 for SEM outcomes.
- `lav_fdr`: false discovery rate correction for selected 'lavaan' parameter p-values.

Note

The development of this package grew from ongoing discussions and interactions (sic) with colleagues, in particular Dr. Cataldo Giuliano Gemmano, whose steady feedback and support helped shape it.

Author(s)

Giuseppe Corbelli (<giuseppe.corbelli@uninettunouniversity.net>)

See Also

Useful links:

- https://github.com/g-corbelli/lavinteract
- Report bugs at https://github.com/g-corbelli/lavinteract/issues

lav_cv *Repeated holdout (Monte Carlo) cross-validation of R^2 for structural equation models ('lavaan' objects)*

Description

Estimate out-of-sample predictive performance for structural relations in a fitted 'lavaan' model using repeated holdout (Monte Carlo cross-validation, leave-group-out CV). At each repetition, the model is refitted on a random training subset and evaluated on a disjoint test subset.

Usage

```
lav_cv(
  fit,
  data = NULL,
  times = "auto",
  train_prop = 0.8,
  seed = 42L,
  quiet = TRUE,
  digits = 3L,
  plot = TRUE,
  tol = 0.001,
  window = 50L,
  max_times = 3000L,
  min_r2_for_pct = 0.05
)

## S3 method for class 'lav_cv'
print(x, digits = x$digits %||% 3L, ...)

## S3 method for class 'lav_cv'
summary(object, ...)
```

Arguments

fit	A fitted 'lavaan' object (required).
data	The data frame used to fit the model; if NULL, it is extracted from 'fit' when available (default: NULL).
times	Integer indicating the number of random splits, or "auto" for stabilization-based early stopping (default: "auto").
train_prop	Numeric in (0,1). Proportion of cases in the training split for each repetition (default: 0.8).
seed	Integer. Random seed for reproducibility of the splits (default: 42).
quiet	Logical. Suppress 'lavaan' refit messages when TRUE (default: TRUE).
digits	Integer. Number of digits to print in summaries (default: 3).

plot	Logical. Show convergence plots of the running mean R^2 per outcome (default: TRUE).
tol	Numeric. Tolerance for the auto-stop rule on the running mean (default: 0.001).
window	Integer. Trailing window size (number of successful splits) used by the auto-stop rule (default: 50).
max_times	Integer. Maximum number of splits when times = "auto" (default: 3000).
min_r2_for_pct	Numeric in (0,1). Minimum in-sample R^2 required to compute percent drop; below this, %_drop is set to NA (default: 0.05).
x	A 'lav_cv' object.
...	Additional arguments; unused.
object	A 'lav_cv' object.

Details

For observed outcomes, R^2 is computed by comparing test-set observed values with predictions obtained by applying the training-set structural coefficients to the test-set predictors.

For latent outcomes, the outcome is not directly observed in the test set. Factor scores for the outcome are first computed in the test set using the measurement model learned on the training set; these scores serve as the outcome values. Predictions are then formed by applying the training-set structural coefficients to the test-set predictors (including factor scores for any latent predictors). R^2 is computed by comparing the test-set factor scores of the outcome with these predicted scores.

The in-sample baseline R^2 is computed on the full dataset using the same metric as in cross-validation: observed outcomes use observed-versus-predicted R^2; latent outcomes use score-versus-predicted-score R^2.

By default, repetitions continue until the running mean R^2 for each outcome stabilizes within a specified tolerance over a trailing window of successful splits, or until a maximum number of splits is reached.

The summary table reports the in-sample baseline R^2, the median cross-validated R^2, its standard deviation, and the percent drop (baseline vs. median CV) with heuristic threshold markers. The percent drop is suppressed when the in-sample R^2 is very small.

Value

A list with class 'lav_cv' and elements:

table Data frame with columns: outcome, type ("observed" or "latent"), r2_in, r2_cv_mean, r2_cv_median, r2_cv_sd, drop_mean_pct, drop_med_pct, splits_used.

split_matrix Matrix of split-wise test-set R^2 values (rows = splits, columns = outcomes).

times Character or integer indicating the number of splits used (e.g., "auto(534)" or 500).

train_prop Numeric. Training proportion used in each split.

N Integer. Number of rows in the input data.

seed Integer. Random seed used to generate the splits.

tol Numeric. Tolerance used by the auto-stop rule.

window Integer. Trailing window size for the auto-stop rule.

`min_r2_for_pct` Numeric. Minimum in-sample R^2 required to compute percent drop.

`call` `match.call()` of the function call.

`digits` Integer. Default number of digits for printing.

References

Cudeck, R., & Browne, M. W. (1983). Cross-validation of covariance structures. *Multivariate Behavioral Research*, *18*(2), 147-167. doi:10.1207/s15327906mbr1802_2

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. Springer. doi:10.1007/9780387216065

Kvålseth, T. O. (1985). Cautionary note about R^2. *The American Statistician*, *39*(4), 279-285. doi:10.1080/00031305.1985.10479448

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289-310. doi:10.1214/10STS330

Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1100-1122. doi:10.1177/1745691617693393

See Also

`sem`, `lavPredict`, `inspect`

Examples

```
set.seed(42)
model <- "
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8

dem60 ~ ind60
dem65 ~ ind60 + dem60

y1 ~~ y5
y2 ~~ y6
"
fit <- lavaan::sem(
model = model,
data = lavaan::PoliticalDemocracy,
std.lv = TRUE,
estimator = "MLR",
meanstructure = TRUE)
result <- lav_cv(
fit = fit,
data = lavaan::PoliticalDemocracy,
times = 5)
print(result)
```

`lav_deltaR2` *Incremental effect sizes for structural predictors in fitted* `lavaan` *models*

Description

Compute outcome-specific incremental effect sizes for structural predictors in a fitted `lavaan` model by comparing the fitted model with nested reduced models in which one predictor, or a block of predictors, is fixed to zero. For each tested predictor set, the function returns the reduction in explained variance (ΔR^2), the corresponding part R^2, and Cohen's $f^2 = \Delta R^2/(1 - R^2_{full})$.

Usage

```
lav_deltaR2(
  fit,
  data = NULL,
  outcome = NULL,
  terms = NULL,
  block = NULL,
  quiet = TRUE,
  digits = 3L
)

## S3 method for class 'lav_deltaR2'
print(x, ...)

## S3 method for class 'lav_deltaR2'
summary(object, ...)
```

Arguments

`fit`	A fitted `lavaan` object.
`data`	Optional data frame used to fit `fit`. If supplied, this data frame is used to refit reduced models. If `NULL`, reduced models are refit using the internal data and options stored in `fit`.
`outcome`	Optional character vector naming endogenous variables for which incremental effect sizes should be computed. If `NULL`, all endogenous variables with at least one free structural predictor are used.
`terms`	Optional character vector naming predictors to test one at a time. Ignored if `block` is supplied. If `NULL`, all eligible structural predictors are tested individually.
`block`	Optional character vector naming predictors to remove jointly. When supplied, one reduced-model comparison is computed per selected outcome for the entire predictor block. A block is evaluated only for outcomes in which all named predictors appear as free structural regressors.
`quiet`	Logical. If `TRUE`, suppress reduced-model refit messages.

digits	Non-negative integer giving the default number of digits used in printing.
x	A 'lav_deltaR2' object.
...	Additional arguments; unused.
object	A 'lav_deltaR2' object.

Details

Reduced models preserve all untargeted model parameters and differ from the fitted model only in that the tested structural regression path(s) are fixed to zero. Accordingly, the resulting ΔR^2 quantifies the unique contribution of the tested predictor or predictor block to the explained variance of a given endogenous variable, conditional on the rest of the model.

If data is supplied, reduced models are refit from that data set. Otherwise, the function reuses the internal data and fitting options stored in fit. The function is intended for converged lavaan models with free structural regressions.

Value

A list with:

- delta_r2_table: A data frame containing one row per outcome-by-test comparison, with columns:
  - outcome: endogenous variable whose R^2 is evaluated;
  - tested_set: predictor or predictor block fixed to zero in the reduced model;
  - test_type: either "single" or "block";
  - n_terms: number of predictors removed jointly;
  - group: numeric group index;
  - group_label: group label, if available;
  - r2_full: R^2 from the fitted model;
  - r2_reduced: R^2 from the reduced model;
  - delta_r2: $R^2_{full} - R^2_{reduced}$;
  - part_r2: equal to delta_r2 in this implementation;
  - f2: Cohen's f^2;
  - f2_magnitude: conventional descriptive interpretation of f2 based on Cohen's benchmarks ("negligible", "small", "medium", or "large");
  - converged: logical indicator of whether the reduced model converged.
- settings: list of user-supplied function settings;
- group_var: grouping variable name, or NULL;
- group_labels: group labels, if available;
- call: matched function call;
- digits: default number of digits used for printing.

Notes

For a given endogenous variable, ΔR^2 is the reduction in explained variance obtained when the tested predictor or predictor block is fixed to zero, and the reduced model is refit with all untargeted parameters left free. In this implementation, part R^2 is numerically identical to ΔR^2. Cohen's f^2 expresses the same quantity relative to the residual variance of the full model. The `f2_magnitude` column applies Cohen's conventional benchmarks to f^2 and is intended only as a descriptive aid.

References

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1-36. doi:10.18637/jss.v048.i02

See Also

`lav_cv` for cross-validated R^2, `lav_vif` for variance inflation factors.

Examples

```
library("lavaan")
data("PoliticalDemocracy",
package = "lavaan")
model <- "
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8
dem60 ~ ind60
dem65 ~ ind60 + dem60
y1 ~~ y5
y2 ~~ y4 + y6
y3 ~~ y7
y4 ~~ y8
y6 ~~ y8
"
fit <- lavaan::sem(model,
data = PoliticalDemocracy)
lav_deltaR2(fit = fit)
lav_deltaR2(fit = fit, outcome = "dem65")
lav_deltaR2(fit = fit, outcome = "dem65", terms = "ind60")
lav_deltaR2(fit = fit, outcome = "dem65", block = c("ind60", "dem60"))
```

`lav_fdr` *False Discovery Rate (FDR) Correction for 'lavaan' parameter p-values*

Description

Apply a false discovery rate correction (Benjamini-Yekutieli by default) to the p-values of selected parameters from a fitted lavaan object.

Usage

```
lav_fdr(
  fit,
  ops = c("reg", "load", "var.cov"),
  family = c("by_group", "selected"),
  method = "BY",
  alpha = 0.05,
  standardized = c("std.all", "std.lv", "std.nox", "none")
)

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

## S3 method for class 'lav_fdr'
summary(object, ...)
```

Arguments

fit	A fitted lavaan object.
ops	Character. One of "reg" (regressions), "load" (factor loadings), or "var.cov" (variances/covariances/residual variances). Default is "reg".
family	Character. If "selected", FDR is applied across all selected parameters jointly. If "by_group", FDR is applied separately within each group. Default is "by_group".
method	Character method passed to stats::p.adjust (default "BY").
alpha	Numeric significance threshold for adjusted p-values (default 0.05).
standardized	Which standardized column to include, or "none".
x	A 'lav_fdr' object.
...	Passed to print.lav_fdr().
object	A 'lav_fdr' object.

Details

Useful when a SEM includes many structural paths (or many other parameters of substantive interest) and there is the need to control the expected proportion of false positives among the parameters declared 'statistically significant'.

With many simultaneous tests, using p < .05 for each parameter inflates the expected number of false positives (about m * .05 under all true null hypotheses, where m is the number of tested parameters). Benjamini-Yekutieli (BY) controls the False Discovery Rate (FDR) under arbitrary dependence structures, which is suitable for SEMs where structural paths are inherently dependent through shared latent variables, covariance matrices, and model constraints.

Value

A list with:

- `fdr_table`: data.frame with raw and FDR-adjusted p-values.
- `settings`: list of settings used.
- `group_var`: group variable name (or `NULL`).
- `group_labels`: group labels if available.
- `call`: matched call.

The returned object has class `"lav_fdr"`.

References

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289-300. doi:10.1111/j.25176161.1995.tb02031.x

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, *29*(4), 1165-1188. doi:10.1214/aos/1013699998

Examples

```
model <- "
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8
dem60 ~ ind60
dem65 ~ ind60 + dem60
y1 ~~ y5
y2 ~~ y6
"
fit <- lavaan::sem(
model = model,
data = lavaan::PoliticalDemocracy,
std.lv = TRUE,
estimator = "MLR",
meanstructure = TRUE)
lav_fdr(fit = fit)
```

---

lav_jn *Johnson-Neyman regions of significance for fitted 'lavaan' models*

Description

Computes the Johnson-Neyman (JN) interval for a continuous moderator from a fitted 'lavaan' model that includes an explicit product term. Plots the conditional slope with a confidence band, shaded significance regions, and an observed-moderator density strip.

Usage

```
lav_jn(
  fit,
  outcome,
  pred,
  modx,
  interaction,
  data = NULL,
  conf.level = 0.95,
  modx.range = NULL,
  n_grid = 1000L,
  x.label = NULL,
  y.label = NULL,
  sig.color = "#009E73",
  nonsig.color = "#D55E00",
  line.color = "#0072B2",
  alpha = 0.20,
  line.size = 0.80,
  rug = TRUE,
  plot = TRUE,
  digits = 3L,
  return_data = FALSE
)

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

## S3 method for class 'lav_jn'
summary(object, ...)
```

Arguments

`fit`	A fitted 'lavaan' object that includes the product term (required).
`outcome`	Character. Name of the dependent variable in `fit` (required).
`pred`	Character. Name of the focal predictor whose conditional slope is probed (required).
`modx`	Character. Name of the continuous moderator (required). If `modx` is categorical (fewer than 4 unique values), the function stops with a message directing the user to `lav_slopes`.
`interaction`	Character. Name of the product term in `fit` (e.g., `"X_Z"`) (required).
`data`	`data.frame`. Optional raw data (not needed; retained for backward compatibility). The function automatically recovers observed data from `fit` when needed.
`conf.level`	Numeric in (0, 1). Confidence level for the Wald band and the JN critical value (default: 0.95).
`modx.range`	Numeric vector of length 2. Range `c(min, max)` for the moderator axis. If `NULL`, the observed range of `modx` in the data is used, extended by 5% on each side for visual padding (default: NULL).

n_grid	Integer. Number of moderator values at which the conditional slope is evaluated for plotting (default: 1000).
x.label	Character. X-axis label (default: modx).
y.label	Character. Y-axis label (default: "Conditional slope of <pred>").
sig.color	Character. Fill color for significant-slope regions (default: "#009E73", Okabe-Ito bluish green).
nonsig.color	Character. Fill color for non-significant regions (default: "#D55E00", Okabe-Ito vermillion).
line.color	Character. Color for the conditional-slope line (default: "#0072B2", Okabe-Ito blue).
alpha	Numeric in (0, 1). Opacity of the confidence ribbon and the region shading (default: 0.20).
line.size	Numeric > 0. Width of the conditional-slope line (default: 0.80).
rug	Logical. If TRUE, draw a rug (density strip) of observed moderator values along the x-axis (default: TRUE).
plot	Logical. If TRUE, produce the JN plot (default: TRUE).
digits	Integer >= 0. Decimal digits in printed output (default: 3).
return_data	Logical. If TRUE, include the plotting data.frame in the returned list (default: FALSE).
x	A 'lav_jn' object.
...	Additional arguments; unused.
object	A 'lav_jn' object.

Details

The model should include a main effect for the predictor, a main effect for the moderator, and their product term. Standard errors are obtained via the delta method using the model-implied covariance matrix of the parameter estimates. If the model was fitted with a robust estimator (e.g., MLR), the robust covariance matrix returned by vcov(fit) is used automatically, so no separate correction is needed.

The JN boundary values are the real roots of the quadratic equation that equates the squared conditional slope to the squared critical value times its variance. When the discriminant is negative, the conditional slope is either significant or non-significant across the entire moderator range and no finite boundary exists.

This function is restricted to continuous moderators. For categorical moderators, use lav_slopes.

Value

A list of class "lav_jn" with elements:

jn_points Numeric vector of length 0, 1, or 2 giving the Johnson-Neyman boundary values of the moderator (the roots of the quadratic). numeric(0) when no real root exists.

signif_regions A data frame with columns lower, upper, and significance ("significant" or "non-significant"), describing the regions over the observed moderator range.

`plot` A ggplot object (or `NULL` when `plot = FALSE`).

`observed_support` Numeric length-2: the minimum and maximum of the observed moderator values.

`interaction_test` List with the unstandardized interaction coefficient (`b`), its `se`, `z`, `p`, Wald confidence interval (`ci`), and standardized beta (`beta_std`).

`labels` List of the user-supplied variable names.

`conf.level` Confidence level used.

`digits` Default digits for printing.

`call` Matched call.

`plot_data` Only when `return_data = TRUE`: data frame used to build the plot.

Notes

All estimates are unstandardized; a standardized coefficient for the interaction is also reported for reference. Wald tests assume large-sample normality of the parameter estimates. When the discriminant of the JN quadratic is negative, the conditional slope is either significant or non-significant everywhere; the function reports which case applies rather than returning a boundary.

References

Johnson, P. O., & Neyman, J. (1936). Tests of certain linear hypotheses and their application to some educational problems. *Statistical Research Memoirs*, *1*, 57-93.

Bauer, D. J., & Curran, P. J. (2005). Probing interactions in fixed and multilevel regression: Inferential and graphical techniques. *Multivariate Behavioral Research*, *40*(3), 373-400. doi:10.1207/s15327906mbr4003_5

Carden, S. W., Holtzman, N. S., & Strube, M. J. (2017). CAHOST: An Excel workbook for facilitating the Johnson-Neyman technique for two-way interactions in multiple regression. *Frontiers in Psychology*, *8*, Article 1293. doi:10.3389/fpsyg.2017.01293

See Also

`lav_slopes` for pick-a-point simple slopes.

Examples

```
set.seed(42)
X <- rnorm(100); Z <- rnorm(100); X_Z <- X*Z
Y <- 0.6*X + 0.6*Z + 0.3*X_Z + rnorm(100, sd = 0.7)
dataset <- data.frame(Y, X, Z, X_Z)
fit <- lavaan::sem("Y ~ X + Z + X_Z", data = dataset)
lav_jn(
fit = fit,
outcome = "Y",
pred = "X",
modx = "Z",
interaction = "X_Z"
)
```

lav_localfit *Local fit diagnostics for fitted* lavaan *models*

Description

Evaluate local fit in a fitted lavaan model from residual-based diagnostics. The function extracts residual summaries, computes standardized covariance and mean residual diagnostics, identifies the largest local discrepancies, and builds a heatmap of covariance residual misfit. By default, the function computes Bentler-standardized residual summaries and standardized covariance residual diagnostics suitable for screening localized strain in the model. The default plot is a traffic-light heatmap of the absolute standardized covariance residuals.

Usage

```
lav_localfit(
  fit,
  group = 1L,
  type = "cor.bentler",
  thresholds = c(1.96, 2.58),
  top_n = 10L,
  triangle = "lower",
  include_diagonal = FALSE,
  plot = TRUE,
  plot_style = "trafficlight",
  show_values = FALSE,
  digits = 3L,
  good_color = "#009E73",
  moderate_color = "#F0E442",
  poor_color = "#D55E00",
  neg_color = "#3B4CC0",
  pos_color = "#B40426",
  na_color = "grey90",
  return_data = FALSE
)

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

## S3 method for class 'lav_localfit'
summary(object, ...)

## S3 method for class 'lav_localfit'
plot(x, y = NULL, ...)
```

Arguments

fit	A fitted lavaan object.

group	Group to inspect in a multigroup model. May be either a numeric group index or a character group label. Ignored for single-group models.
type	Character string passed to lavaan::lavResiduals(). Common choices are "cor.bentler" (default), "raw", "cor", and "cor.bollen".
thresholds	Numeric vector of length 2 giving the descriptive cutoffs used for standardized residual magnitudes. The first value separates "minor" from "moderate" discrepancies; the second separates "moderate" from "notable" discrepancies. Used for the traffic-light heatmap and for printed summaries. Defaults to c(1.96, 2.58), as per common z-based guidelines.
top_n	Integer. Number of largest absolute standardized covariance residuals to report in the output table.
triangle	Character string. Either "lower" or "full". If "lower", only the lower triangle of the covariance residual matrix is used in the heatmap and top-residual table.
include_diagonal	Logical. If TRUE, include diagonal elements (variance residuals) in the covariance residual diagnostics.
plot	Logical. If TRUE, produce a heatmap of covariance residual misfit.
plot_style	Character string. Either "trafficlight" or "signed". The default "trafficlight" plots the absolute standardized covariance residuals in descriptive magnitude categories. The "signed" option plots signed standardized residuals on a diverging color scale.
show_values	Logical. If TRUE, overlay numeric residual values on the heatmap tiles.
digits	Non-negative integer giving the default number of digits used in printed output and numeric labels.
good_color	Fill color for "minor" discrepancies in the traffic-light heatmap.
moderate_color	Fill color for "moderate" discrepancies in the traffic-light heatmap.
poor_color	Fill color for "notable" discrepancies in the traffic-light heatmap.
neg_color	Fill color for negative residuals in the signed heatmap.
pos_color	Fill color for positive residuals in the signed heatmap.
na_color	Fill color for cells omitted from the heatmap, such as the upper triangle when triangle = "lower".
return_data	Logical. If TRUE, include the long-format heatmap data frame in the returned object.
x	A "lav_localfit" object.
...	Additional arguments; unused.
object	A "lav_localfit" object.
y	Ignored.

Value

A list of class "lav_localfit" with elements:

- summary: residual summary returned by lavaan::lavResiduals() for the selected group;

- `cov_residuals`: covariance residual matrix for the selected group;
- `cov_z`: standardized covariance residual matrix;
- `mean_residuals`: mean residual vector, if available;
- `mean_z`: standardized mean residual vector, if available;
- `top_cov_residuals`: data frame of the largest absolute standardized covariance residuals;
- `top_mean_residuals`: data frame of the largest absolute standardized mean residuals;
- `counts`: list with counts of covariance residuals exceeding the descriptive thresholds, the maximum absolute standardized covariance residual, and the corresponding variable pair;
- `plot`: a `ggplot` heatmap object, or `NULL` if `plot = FALSE`;
- `group`: numeric index of the selected group;
- `group_label`: character label of the selected group;
- `type`: residual type used;
- `thresholds`: thresholds used for descriptive classification;
- `digits`: default number of digits for printing;
- `call`: matched function call;
- `plot_data`: only when `return_data = TRUE`, the long-format data used to build the heatmap.

Notes

In this function, local fit is evaluated from residuals of the sample summary statistics rather than from casewise prediction errors. Standardized residuals are used to screen where the model reproduces specific covariances or means poorly. The default thresholds are descriptive aids only. They should not be treated as formal decision rules or as substitutes for theory-based model evaluation. When the fitted object uses bootstrap standard errors, standardized residual diagnostics are obtained from an auxiliary refit of the same model without bootstrap.

References

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1-36. doi:10.18637/jss.v048.i02

Schermelleh-Engel, K., Moosbrugger, H., & Muller, H. (2003). Evaluating the fit of structural equation models: Tests of significance and descriptive goodness-of-fit measures. *Methods of Psychological Research Online*, *8*(2), 23-74.

See Also

`lavResiduals` for residual extraction and `lav_deltaR2` for local structural effect-size diagnostics.

Examples

```
model <- "
ind60 =~ x1 + x2 + x3
dem60 =~ y1 + y2 + y3 + y4
dem65 =~ y5 + y6 + y7 + y8
dem60 ~ ind60
```

```
dem65 ~ ind60 + dem60
y1 ~~ y5
y2 ~~ y6
"
fit <- lavaan::sem(
model = model,
data = lavaan::PoliticalDemocracy,
std.lv = TRUE,
estimator = "MLR",
meanstructure = TRUE)

lav_localfit(fit)
```

`lav_slopes` *Simple slopes and interaction plots for fitted 'lavaan' models*

Description

Computes conditional (simple) slopes of a focal predictor across values of a moderator from a fitted 'lavaan' model that includes their explicit product term. Plots predicted lines with Wald confidence ribbons and prints an APA-style test of the interaction for easy reporting and interpretation, together with a simple slopes table.

Usage

```
lav_slopes(
  fit,
  outcome,
  pred,
  modx,
  interaction,
  data = NULL,
  modx.values = NULL,
  modx.labels = NULL,
  pred.range = NULL,
  conf.level = 0.95,
  x.label = NULL,
  y.label = NULL,
  legend.title = NULL,
  colors = NULL,
  line.size = 0.80,
  alpha = 0.20,
  table = TRUE,
  digits = 3L,
  modx_n_unique_cutoff = 4L,
  return_data = FALSE
)
```

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

## S3 method for class 'lav_slopes'
summary(object, ...)
```

Arguments

fit	A fitted 'lavaan' object that includes the product term (required).
outcome	Character. Name of the dependent variable in fit (required).
pred	Character. Name of the focal predictor whose simple slopes are probed (required).
modx	Character. Name of the moderator. The moderator must appear in the fitted model as a single numeric variable. General nominal moderators with more than two categories are not supported.
interaction	Character. Name of the product term in fit (e.g., "X_Z") (required).
data	data.frame. Optional raw data (not needed, retained for backward compatibility). The function automatically recovers observed data from fit when needed.
modx.values	Numeric vector of moderator values at which to compute simple slopes. If NULL and modx is observed and numeric, the function uses mean minus 1 SD, the mean, and mean plus 1 SD for moderators with more than modx_n_unique_cutoff unique values, and otherwise uses the observed numeric values as discrete probe points. If modx is latent in a single-group model, the function uses the estimated latent mean minus 1 latent SD, the latent mean, and the latent mean plus 1 latent SD.
modx.labels	Character vector. Legend and table labels for modx.values. By default, the labels are c("-1 SD", "Mean", "+1 SD") when values are derived as mean plus or minus 1 SD, and as.character(modx.values) otherwise.
pred.range	Numeric vector of length 2. Range c(min, max) for the x-axis for the focal predictor. If NULL and pred is observed, the observed range recovered from fit is used. If pred is latent in a single-group model, the function uses the estimated latent mean minus 2 latent SD and the latent mean plus 2 latent SD. Otherwise, c(-2, 2) is used.
conf.level	Numeric in (0,1). Confidence level for Wald confidence intervals and ribbons (default: 0.95).
x.label	Character. X-axis label (default: pred).
y.label	Character. Y-axis label (default: outcome).
legend.title	Character. Legend title; if NULL, the legend shows only levels (default: NULL).
colors	Character vector. Colors for lines and ribbons; named vector recommended with names matching modx.labels (default: Okabe-Ito palette).
line.size	Numeric > 0. Line width (default: 0.80).
alpha	Numeric in (0,1). Ribbon opacity (default 0.20).
table	Logical. Print APA-style interaction test and simple-slopes table (default: TRUE).

`digits` Integer >= 0. Decimal digits in printed output (default: 3).

`modx_n_unique_cutoff`
Integer >= 1. Threshold for treating a numeric moderator as continuous and using mean ± SD (default: 4).

`return_data` Logical. If TRUE, include the plotting data.frame in the returned list (default: FALSE).

`x` A 'lav_slopes' object.

`...` Additional arguments; unused.

`object` A 'lav_slopes' object.

Details

The model should include a main effect for the predictor, a main effect for the moderator, and one explicit product term between them.

The moderator must enter the fitted model as a single numeric variable, which may be continuous, a binary observed moderator coded as a single numeric dummy variable, an observed numeric moderator with a small number of values treated as discrete probe points, or a latent moderator treated as a single continuous latent variable.

Standard errors use the delta method with the model covariance matrix of the estimates. When moderator values are derived automatically for latent moderators, probe points are based on the estimated latent mean and model-implied latent standard deviation.

Value

A list of class `"lav_slopes"` with elements:

`plot` `ggplot` object with lines and confidence ribbons.

`slope_table` Data frame with moderator levels, simple slopes, SE, z, and CI.

`plot_data` Only when `return_data = TRUE`: data used to build the plot.

Notes

Estimates are unstandardized; a standardized coefficient for the interaction is also reported for reference. Wald tests assume large-sample normality of the parameter estimates. Multigroup fitted models are not supported.

References

Aiken, L. S., & West, S. G. (1991). *Multiple regression: Testing and interpreting interactions*. Sage.

Preacher, K. J., Curran, P. J., & Bauer, D. J. (2006). Computational tools for probing interactions in multiple linear regression, multilevel modeling, and latent curve analysis. *Journal of Educational and Behavioral Statistics*, *31*(4), 437-448. doi:10.3102/10769986031004437

Rogosa, D. (1980). Comparing nonparallel regression lines. *Psychological Bulletin*, *88*(2), 307-321. doi:10.1037/00332909.88.2.307

See Also

lav_jn for Johnson-Neyman regions of significance.

Examples

```
set.seed(42)
X <- rnorm(100); Z <- rnorm(100); X_Z <- X*Z
Y <- 0.6*X + 0.6*Z + 0.3*X_Z + rnorm(100, sd = 0.7)
dataset <- data.frame(Y, X, Z, X_Z)
fit <- lavaan::sem("Y ~ X + Z + X_Z", data = dataset)
lav_slopes(
fit = fit,
outcome = "Y",
pred = "X",
modx = "Z",
interaction = "X_Z")
```

---

lav_vif *Variance Inflation Factors for 'lavaan' Structural Predictors*

Description

Compute VIF for each predictor that appears in structural regressions with two or more predictors, refitting the necessary sub-models so that latent predictors are handled at the latent level (i.e., with their original measurement models). It returns also the R^2 of each eligible endogenous variable from the original fit for context.

Usage

```
lav_vif(
  fit,
  data = NULL,
  quiet = TRUE
)

## S3 method for class 'lav_vif'
print(x, digits = 3, cutoff = c(5, 10), ...)

## S3 method for class 'lav_vif'
summary(object, ...)
```

Arguments

fit	A fitted lavaan object.
data	Optional. The data frame used to fit fit. If NULL, the function attempts to extract the data from fit via lavInspect(fit, "data"). For multigroup models, grouped data are reconstructed into a single data frame when possible.

quiet	Logical. If TRUE suppresses lavaan refit messages.
x	A 'lav_vif' object.
digits	Integer number of digits to print.
cutoff	Numeric length-2 thresholds used to flag VIF values.
...	Passed to 'print.lav_vif()' (e.g., 'digits', 'cutoff').
object	A 'lav_vif' object.

Details

Each auxiliary refitted model includes the measurement structure for latent predictors, rebuilt from the original model syntax, together with observed residual covariances among the involved indicators when specified in the original model. Then, regresses the focal predictor on the remaining predictors at the latent level when applicable.

$\mathrm{VIF}_i = 1/(1 - R_i^2)$ generalizes VIF to SEM while respecting measurement models.

The function reuses the estimator, missing-data handling, and several options from fit. In multigroup models, VIF is computed separately within each group. Higher-order latent predictors are supported by recursively rebuilding lower-order measurement blocks in the auxiliary refitted models.

Value

A list with:

- vif_table: data.frame with columns outcome, predictor, group, r2_predictor, vif, k_predictors.
- outcome_r2: data.frame with R^2 per eligible endogenous outcome and group from the original fit.

References

Fox, J., & Monette, G. (1992). Generalized collinearity diagnostics. *Journal of the American Statistical Association*, *87*(417), 178-183. doi:10.1080/01621459.1992.10475190

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics: Identifying influential data and sources of collinearity*. Wiley.

Examples

```
set.seed(42)
x1 <- rnorm(100); x2 <- 0.85*x1 + rnorm(100, sd = sqrt(1 - 0.85^2)); x3 <- rnorm(100)
y <- 0.5*x1 + 0.3*x2 + 0.1*x3 + rnorm(100, sd = 0.7)
dataset <- data.frame(y, x1, x2, x3)
fit <- lavaan::sem("y ~ x1 + x2 + x3", data = dataset)
lav_vif(
fit = fit,
data = dataset)
```

Index