

Package ‘lavaanExtra’

February 4, 2023

Title Convenience Functions for Package 'lavaan'

Version 0.1.4

Date 2023-02-04

Description Affords an alternative, vector-based syntax to 'lavaan', as well as other convenience functions such as naming paths and defining indirect links automatically, in addition to convenience formatting optimized for a publication and script sharing workflow.

License MIT + file LICENSE

URL <https://lavaanExtra.remi-theriault.com>

BugReports <https://github.com/rempsyc/lavaanExtra/issues>

Depends R (>= 3.5)

Imports lavaan,

Suggests rempsyc, flextable, lavaanPlot, DiagrammeRsvg, rsvg, png, webshot, tidySEM, rlang, knitr, tibble, rmarkdown, markdown, testthat (>= 3.0.0), covr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Rémi Thériault [aut, cre] (<<https://orcid.org/0000-0003-4315-6788>>)

Maintainer Rémi Thériault <remi.theriault@mail.mcgill.ca>

Repository CRAN

Date/Publication 2023-02-04 22:22:31 UTC

R topics documented:

cfa_fit_plot	2
lavaan_cov	3

lavaan_ind	4
lavaan_reg	5
nice_fit	6
nice_lavaanPlot	7
nice_tidySEM	8
write_lavaan	10

Index	12
--------------	-----------

cfa_fit_plot	<i>Fit and plot CFA simultaneously</i>
--------------	--

Description

Prints and saves CFA fit, as well as plots CFA factor loadings, simultaneously.

Usage

```
cfa_fit_plot(
  model,
  data,
  covs = FALSE,
  estimator = "MLR",
  remove.items = "",
  print = TRUE,
  save.as.pdf = FALSE,
  file.name,
  ...
)
```

Arguments

model	CFA model to fit.
data	Data set on which to fit the CFA model.
covs	Logical, whether to include covariances on the lavaan plot.
estimator	What estimator to use for the CFA.
remove.items	Optional, if one wants to remove items from the CFA model without having to redefine it completely again.
print	Logical, whether to print model summary to console.
save.as.pdf	Logical, whether to save as PDF for a high-resolution, scalable vector graphic quality plot. Defaults to saving to the "/model" subfolder of the working directory. If it doesn't exist, it creates it. Then automatically open the created PDF in the default browser. Defaults to false.
file.name	Optional (when save.as.pdf is set to TRUE), if one wants something different than the default file name. It saves to pdf per default, so the .pdf extension should not be specified as it will add it automatically.
...	Arguments to be passed to function lavaan::cfa.

Value

The function returns a lavaan fit object. However, it also: prints a summary of the lavaan fit object to the console, and; prints a lavaanPlot of the lavaan fit object.

Illustrations**Examples**

```
(latent <- list(visual = paste0("x", 1:3),
              textual = paste0("x", 4:6),
              speed = paste0("x", 7:9)))

HS.model <- write_lavaan(latent = latent)
cat(HS.model)

library(lavaan)
fit <- cfa_fit_plot(HS.model, HolzingerSwineford1939)
```

 lavaan_cov

Extract relevant covariance indices from lavaan model

Description

Extract relevant covariance indices from lavaan model through `lavaan::parameterEstimates` with `standardized = TRUE`. In this case, the correlation coefficient (r) represents the resulting `std.all` column.

Usage

```
lavaan_cov(fit, nice_table = FALSE, ...)
```

Arguments

<code>fit</code>	lavaan fit object to extract covariance indices from
<code>nice_table</code>	Logical, whether to print the table as a <code>remsync::nice_table</code> as well as print the reference values at the bottom of the table.
<code>...</code>	Arguments to be passed to <code>remsync::nice_table</code>

Value

A dataframe of covariances, including the covaried variables, the covariance, and corresponding p-value.

Examples

```
(latent <- list(visual = paste0("x", 1:3),
              textual = paste0("x", 4:6),
              speed = paste0("x", 7:9)))

(regression <- list(ageyr = c("visual", "textual", "speed"),
                  grade = c("visual", "textual", "speed")))

(covariance <- list(speed = "textual", ageyr = "grade"))

HS.model <- write_lavaan(regression = regression, covariance = covariance,
                       latent = latent, label = TRUE)

cat(HS.model)

library(lavaan)
fit <- lavaan(HS.model, data=HolzingerSwineford1939,
             auto.var=TRUE, auto.fix.first=TRUE,
             auto.cov.lv.x=TRUE)
lavaan_cov(fit)
```

lavaan_ind

Extract relevant indirect effects indices from lavaan model

Description

Extract relevant indirect effects indices from lavaan model through `lavaan::parameterEstimates` with `standardized = TRUE`. In this case, the beta (B) represents the resulting `std.all` column.

Usage

```
lavaan_ind(fit, nice_table = FALSE, underscores_to_arrows = TRUE, ...)
```

Arguments

<code>fit</code>	lavaan fit object to extract fit indices from
<code>nice_table</code>	Logical, whether to print the table as a <code>rempsyc::nice_table</code> as well as print the reference values at the bottom of the table.
<code>underscores_to_arrows</code>	Logical, whether to convert underscore to arrows in the "Indirect Effect column".
<code>...</code>	Arguments to be passed to <code>rempsyc::nice_table</code>

Value

A dataframe, including the indirect effect, corresponding paths, standardized regression coefficient, and corresponding p-value.

Examples

```
(latent <- list(visual = paste0("x", 1:3),
              textual = paste0("x", 4:6),
              speed = paste0("x", 7:9)))

(mediation <- list(speed = "visual",
                  textual = "visual",
                  visual = c("ageyr", "grade"))))

(indirect <- list(IV = c("ageyr", "grade"),
                 M = "visual",
                 DV = c("speed", "textual"))))

HS.model <- write_lavaan(mediation, indirect = indirect,
                       latent = latent, label = TRUE)
cat(HS.model)

library(lavaan)
fit <- lavaan(HS.model, data=HolzingerSwineford1939,
             auto.var=TRUE, auto.fix.first=TRUE,
             auto.cov.lv.x=TRUE)
lavaan_ind(fit)
```

lavaan_reg

Extract relevant regression indices from lavaan model

Description

Extract relevant regression indices from lavaan model through `lavaan::parameterEstimates` with `standardized = TRUE`. In this case, the beta (B) represents the resulting `std.all` column.

Usage

```
lavaan_reg(fit, nice_table = FALSE, ...)
```

Arguments

<code>fit</code>	lavaan fit object to extract fit indices from
<code>nice_table</code>	Logical, whether to print the table as a <code>rempsyc::nice_table</code> as well as print the reference values at the bottom of the table.
<code>...</code>	Arguments to be passed to <code>rempsyc::nice_table</code>

Value

A dataframe, including the outcome, predictor, standardized regression coefficient, and corresponding p-value.

Examples

```
(latent <- list(visual = paste0("x", 1:3),
              textual = paste0("x", 4:6),
              speed = paste0("x", 7:9)))

(regression <- list(ageyr = c("visual", "textual", "speed"),
                  grade = c("visual", "textual", "speed")))

HS.model <- write_lavaan(latent = latent, regression = regression)
cat(HS.model)

library(lavaan)
fit <- lavaan(HS.model, data=HolzingerSwineford1939,
             auto.var=TRUE, auto.fix.first=TRUE,
             auto.cov.lv.x=TRUE)
lavaan_reg(fit)
```

nice_fit

Extract relevant fit indices from lavaan model

Description

Compares fit from one or several lavaan models. Also optionally includes references values. The reference fit values are based on Schreiber et al. (2006).

Usage

```
nice_fit(model, model.labels, nice_table = FALSE)
```

Arguments

model	lavaan model object(s) to extract fit indices from
model.labels	Model labels to use. If a named list is provided for model, default to the names of the list. Otherwise, if the list is unnamed, defaults to generic numbering.
nice_table	Logical, whether to print the table as a <code>rempsyc::nice_table</code> as well as print the reference values at the bottom of the table.

Value

A dataframe, representing select fit indices (chi2, df, chi2/df, p-value of the chi2 test, CFI, TLI, RMSEA, SRMR, AIC, and BIC).

References

Schreiber, J. B., Nora, A., Stage, F. K., Barlow, E. A., & King, J. (2006). Reporting structural equation modeling and confirmatory factor analysis results: A review. *The Journal of educational research*, 99(6), 323-338. <https://doi.org/10.3200/JOER.99.6.323-338>

Examples

```
(latent <- list(
  visual = paste0("x", 1:3),
  textual = paste0("x", 4:6),
  speed = paste0("x", 7:9)
))

(regression <- list(
  ageyr = c("visual", "textual", "speed"),
  grade = c("visual", "textual", "speed")
))

HS.model <- write_lavaan(latent = latent, regression = regression)
cat(HS.model)

library(lavaan)
fit <- sem(HS.model, data = HolzingerSwineford1939)
nice_fit(fit)
```

nice_lavaanPlot *Make a quick lavaanPlot*

Description

Make a quick and decent-looking lavaanPlot.

Usage

```
nice_lavaanPlot(
  model,
  node_options = list(shape = "box", fontname = "Helvetica"),
  edge_options = list(color = "grey"),
  coefs = TRUE,
  stand = TRUE,
  covs = FALSE,
  stars = list("regress"),
  sig = 0.05,
  graph_options = list(rankdir = "LR"),
  ...
)
```

Arguments

model	SEM or CFA model to plot.
node_options	Shape and font name.
edge_options	Colour of edges.

coefs	Logical, whether to plot coefficients.
stand	Logical, whether to standardized coefficients.
covs	Logical, whether to plot covariances
stars	Logical, whether to plot significance stars.
sig	Significance threshold.
graph_options	Read from left to right, rather than from top to bottom.
...	Arguments to be passed to function <code>lavaanPlot::lavaanPlot</code> .

Value

A `lavaanPlot`, of classes `c("grViz", "htmlwidget")`, representing the specified lavaan model.

Illustrations**Examples**

```
(latent <- list(visual = paste0("x", 1:3),
              textual = paste0("x", 4:6),
              speed = paste0("x", 7:9)))

HS.model <- write_lavaan(latent = latent)
cat(HS.model)

library(lavaan)
fit <- cfa(HS.model, HolzingerSwineford1939)
nice_lavaanPlot(fit)
```

nice_tidySEM

Make a quick tidySEM plot

Description

Make a quick and decent-looking tidySEM plot.

Usage

```
nice_tidySEM(
  fit,
  layout = NULL,
  hide_nonsig_edges = FALSE,
  hide_var = TRUE,
  hide_mean = TRUE,
  est_std = TRUE,
  label,
  label_location = NULL,
```

```

    plot = TRUE,
    ...
  )

```

Arguments

<code>fit</code>	SEM or CFA model fit to plot.
<code>layout</code>	A matrix (or data.frame) that describes the structure; see get_layout . If a named list is provided, with names "IV" (independent variables), "M" (mediator), and "DV" (dependent variables), <code>nice_tidySEM</code> attempts to write the layout matrix automatically.
<code>hide_nonsig_edges</code>	Logical, hides non-significant edges. Defaults to FALSE.
<code>hide_var</code>	Logical, hides variances. Defaults to TRUE.
<code>hide_mean</code>	Logical, hides means/node labels. Defaults to TRUE.
<code>est_std</code>	Logical, whether to use the standardized coefficients. Defaults to TRUE.
<code>label</code>	Labels to be used on the plot. As elsewhere in <code>lavaanExtra</code> , it is provided as a named list with format (<code>colname = "label"</code>).
<code>label_location</code>	Location of label along the path, as a percentage (defaults to middle, 0.5).
<code>plot</code>	Logical, whether to plot the result (default). If FALSE, returns the <code>tidy_sem</code> object, which can be further edited as needed.
<code>...</code>	Arguments to be passed to prepare_graph .

Value

A tidySEM plot, of class `ggplot`, representing the specified lavaan model.

Illustrations

Examples

```

# Calculate scale averages
library(lavaan)
data <- HolzingerSwineford1939
data$visual <- rowMeans(data[paste0("x", 1:3)])
data$textual <- rowMeans(data[paste0("x", 4:6)])
data$speed <- rowMeans(data[paste0("x", 7:9)])

# Define our variables
IV <- c("sex", "ageyr", "agemo", "school")
M <- c("visual", "grade")
DV <- c("speed", "textual")

# Define our lavaan lists
mediation <- list(speed = M, textual = M, visual = IV, grade = IV)

```

```
# Define indirect object
structure <- list(IV = IV, M = M, DV = DV)

# Write the model, and check it
model <- write_lavaan(mediation, indirect = structure, label = TRUE)
cat(model)

# Fit model
fit <- sem(model, data)

# Plot model

nice_tidySEM(fit, layout = structure)
```

write_lavaan

Vector-based lavaan syntax interpreter

Description

Vector-based lavaan syntax interpreter.

Usage

```
write_lavaan(
  mediation = NULL,
  regression = NULL,
  covariance = NULL,
  indirect = NULL,
  latent = NULL,
  intercept = NULL,
  constraint.equal = NULL,
  constraint.smaller = NULL,
  constraint.larger = NULL,
  custom = NULL,
  label = FALSE,
  use.letters = FALSE
)
```

Arguments

mediation	Mediation indicators (~ symbol: "is regressed on"). Differs from argument regression because path names can be optionally specified automatically with argument label.
regression	Regression indicators (~ symbol: "is regressed on").
covariance	(Residual) (co)variance indicators (~ symbol: "is correlated with").

indirect	Indirect effect indicators (:= symbol: "indirect effect defined as"). If a named list is provided, with names "IV" (independent variables), "M" (mediator), and "DV" (dependent variables), write_lavaan attempts to write indirect effects automatically. In this case, the mediation argument must be specified too.
latent	Latent variable indicators (=~ symbol: "is measured by").
intercept	Intercept indicators (~ 1 symbol: "intercept").
constraint.equal	Equality indicators (== symbol).
constraint.smaller	Smaller than indicators (< symbol).
constraint.larger	Greater than indicators (> symbol).
custom	Custom specifications. Takes a <i>single</i> string just like regular lavaan syntax would. Always added at the end of the model.
label	Logical, whether to display path names for the mediation argument.
use.letters	Logical, for the labels, whether to use letters instead of the variable names.

Value

A character string, representing the specified lavaan model.

Examples

```
(latent <- list(visual = paste0("x", 1:3),
              textual = paste0("x", 4:6),
              speed = paste0("x", 7:9)))

HS.model <- write_lavaan(latent = latent)
cat(HS.model)

library(lavaan)
fit <- lavaan(HS.model, data = HolzingerSwineford1939,
             auto.var = TRUE, auto.fix.first = TRUE,
             auto.cov.lv.x = TRUE)
summary(fit, fit.measures=TRUE)
```

Index

- * **CFA**
 - cfa_fit_plot, 2
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - nice_lavaanPlot, 7
 - nice_tidySEM, 8
 - write_lavaan, 10
- * **analysis**
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - write_lavaan, 10
- * **equation**
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - write_lavaan, 10
- * **fit**
 - cfa_fit_plot, 2
 - nice_lavaanPlot, 7
 - nice_tidySEM, 8
- * **lavaan**
 - cfa_fit_plot, 2
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - nice_lavaanPlot, 7
 - nice_tidySEM, 8
 - write_lavaan, 10
- * **modeling**
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - write_lavaan, 10
- * **path**
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - write_lavaan, 10
- * **plot**
 - cfa_fit_plot, 2
 - nice_lavaanPlot, 7
 - nice_tidySEM, 8
- * **structural**
 - lavaan_cov, 3
 - lavaan_ind, 4
 - lavaan_reg, 5
 - nice_fit, 6
 - write_lavaan, 10
- * **table_results**
 - nice_tidySEM, 8
- * **tidySEM**
 - nice_tidySEM, 8
- cfa_fit_plot, 2
- get_layout, 9
- lavaan_cov, 3
- lavaan_ind, 4
- lavaan_reg, 5
- nice_fit, 6
- nice_lavaanPlot, 7
- nice_tidySEM, 8
- prepare_graph, 9
- write_lavaan, 10