

# Package ‘lavaanExtra’

August 26, 2023

**Title** Convenience Functions for Package 'lavaan'

**Version** 0.1.8

**Date** 2023-08-25

**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, insight

**Suggests** rempsyc (> 0.1.2), flextable, lavaanPlot, DiagrammeRsvg, rsvg, png, webshot, tidySEM, tmvnsim, knitr, tibble, sjlabelled, stringdist, rmarkdown, markdown, testthat (>= 3.0.0), covr, spelling

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**Language** en-US

**NeedsCompilation** no

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

**Maintainer** Rémi Thériault <[remi.theriault@mail.mcgill.ca](mailto:remi.theriault@mail.mcgill.ca)>

**Repository** CRAN

**Date/Publication** 2023-08-25 23:10:02 UTC

## R topics documented:

cfa_fit_plot	2
lavaan_cor	3
lavaan_cov	4
lavaan_ind	5
lavaan_reg	7
nice_fit	8
nice_lavaanPlot	9
nice_modindices	10
nice_tidySEM	12
write_lavaan	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

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 <a href="#">lavaan::cfa</a> .

**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\_cor

*Extract relevant correlation indices from lavaan model*


---

**Description**

Extract relevant correlation 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_cor(fit, nice_table = FALSE, ...)
```

**Arguments**

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

**Value**

A dataframe of correlations, including the correlated variables, the correlation, and the 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 <- sem(HS.model, data = HolzingerSwineford1939)
lavaan_cor(fit)
```

---

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 covariances represent the resulting `std.all` column.

**Usage**

```
lavaan_cov(fit, estimate = "B", nice_table = FALSE, ...)
```

**Arguments**

<code>fit</code>	lavaan fit object to extract covariance indices from
<code>estimate</code>	What estimate to use, either the standardized estimate ("B", default), or unstandardized estimate ("b").
<code>nice_table</code>	Logical, whether to print the table as a <a href="#">rempsyc::nice_table</a> as well as print the reference values at the bottom of the table.
<code>...</code>	Arguments to be passed to <a href="#">rempsyc::nice_table</a>

**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 <- sem(HS.model, data = HolzingerSwineford1939)
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. See "Value" section for more details.

**Usage**

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

**Arguments**

<code>fit</code>	lavaan fit object to extract fit indices from
<code>estimate</code>	What estimate to use, either the standardized estimate ("B", default), or unstandardized estimate ("b").
<code>nice_table</code>	Logical, whether to print the table as a <a href="#">rempsyc::nice_table</a> 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 <a href="#">rempsyc::nice_table</a>

**Value**

A dataframe, including the indirect effect ("lhs"), corresponding paths ("rhs"), standardized regression coefficient ("std.all"), corresponding p-value, as well as the unstandardized regression coefficient ("est") and its confidence interval ("ci.lower", "ci.upper").

**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 <- sem(HS.model, data = HolzingerSwineford1939)
  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. See "Value" section for more details.

## Usage

```
lavaan_reg(fit, estimate = "B", nice_table = FALSE, ...)
```

## Arguments

<code>fit</code>	lavaan fit object to extract fit indices from
<code>estimate</code>	What estimate to use, either the standardized estimate ("B", default), or unstandardized estimate ("b").
<code>nice_table</code>	Logical, whether to print the table as a <a href="#">rempsyc::nice_table</a> as well as print the reference values at the bottom of the table.
<code>...</code>	Arguments to be passed to <a href="#">rempsyc::nice_table</a>

## Value

A dataframe, including the outcome ("lhs"), predictor ("rhs"), standardized regression coefficient ("std.all"), corresponding p-value, as well as the unstandardized regression coefficient ("est") and its confidence interval ("ci.lower", "ci.upper").

## 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)
lavaan_reg(fit)

```

---

nice\_fit

---

*Extract relevant fit indices from lavaan model*


---

## Description

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

## Usage

```
nice_fit(model, model.labels, nice_table = FALSE, stars = 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 <a href="#">rempsyc::nice_table</a> as well as print the reference values at the bottom of the table.
stars	Logical, if nice_table = TRUE, whether to display significance stars (defaults to FALSE).

## Value

A dataframe, representing select fit indices (chi2, df, chi2/df, p-value of the chi2 test, CFI, TLI, RMSEA and its 90% CI, SRMR, AIC, and BIC).

## References

Schreiber, J. B. (2017). Update to core reporting practices in structural equation modeling. *Research in social and administrative pharmacy*, 13(3), 634-643. <https://doi.org/10.1016/j.sapharm.2016.06.006>



## 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 = c(color = "black"),
  coefs = TRUE,
  stand = TRUE,
  covs = FALSE,
  stars = c("regress", "latent", "covs"),
  sig = 0.05,
  graph_options = c(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. Defaults to TRUE.
stand	Logical, whether to use standardized coefficients. Defaults to TRUE.
covs	Logical, whether to plot covariances. Defaults to FALSE.
stars	Which links to plot significance stars for. One of <code>c("regress", "latent", "covs")</code> .
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\_modindices

*Extract relevant modification indices along item labels*

---

### Description

Extract relevant modification indices along item labels, with a similarity score provided to help guide decision-making for removing redundant items with high covariance.

### Usage

```
nice_modindices(fit, labels = NULL, method = "lcs", sort = TRUE, ...)
```

**Arguments**

fit	lavaan fit object to extract modification indices from
labels	Dataframe of labels. If the original data frame is provided, and that it contains labelled variables, will automatically attempt to extract the correct labels from the dataframe.
method	Method for distance calculation from <code>stringdist::stringsim</code> . Defaults to "lcs".
sort	Logical. If TRUE, sort the output using the values of the modification index values. Higher values appear first. Defaults to TRUE.
...	Arguments to be passed to <code>lavaan::modindices</code>

**Value**

A dataframe, including the outcome ("lhs"), predictor ("rhs"), standardized regression coefficient ("std.all"), corresponding p-value, as well as the unstandardized regression coefficient ("est") and its confidence interval ("ci.lower", "ci.upper").

**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_modindices(fit, maximum.number = 5)
data_labels <- data.frame(
  x1 = "I have good visual perception",
  x2 = "I have good cube perception",
  x3 = "I have good at lozenge perception",
  x4 = "I have paragraph comprehension",
  x5 = "I am good at sentence completion",
  x6 = "I excel at finding the meaning of words",
  x7 = "I am quick at doing mental additions",
  x8 = "I am quick at counting dots",
  x9 = "I am quick at discriminating straight and curved capitals"
)
nice_modindices(fit, maximum.number = 10, labels = data_labels, op = "~~")

x <- HolzingerSwineford1939
```

```
x <- sjlabelled::set_label(x, label = c(rep("", 6), data_labels))
fit <- sem(HS.model, data = x)
nice_modindices(fit, maximum.number = 10, op = "~~")
```

---

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_cov = FALSE,
  hide_mean = TRUE,
  est_std = TRUE,
  label,
  label_location = NULL,
  reduce_items = 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 <a href="#">tidySEM::get_layout</a> . If a named list is provided, with names "IV" (independent variables), "M" (mediator), and "DV" (dependent variables), nice_tidySEM 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_cov</code>	Logical, hides co-variances. Defaults to FALSE.
<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 lavaanExtra, it is provided as a named list with format (colname = "label").
<code>label_location</code>	Location of label along the path, as a percentage (defaults to middle, 0.5).

reduce_items	A numeric vector of length 1 (x) or 2 (x & y) defining how much space to trim from the nodes (boxes) of the items defining the latent variables. Can be provided either as <code>reduce_items = 0.4</code> (will only affect horizontal space, x), or <code>reduce_items = c(x = 0.4, y = 0.2)</code> (will affect both horizontal x and vertical y).
plot	Logical, whether to plot the result (default). If FALSE, returns the <code>tidy_sem</code> object, which can be further edited as needed.
...	Arguments to be passed to <code>tidySEM::prepare_graph</code> .

**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\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - nice\_lavaanPlot, 9
    - nice\_tidySEM, 12
    - write\_lavaan, 14
  - \* **analysis**
    - lavaan\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - write\_lavaan, 14
  - \* **equation**
    - lavaan\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - write\_lavaan, 14
  - \* **fit**
    - cfa\_fit\_plot, 2
    - nice\_lavaanPlot, 9
    - nice\_tidySEM, 12
  - \* **lavaan**
    - cfa\_fit\_plot, 2
    - lavaan\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - nice\_lavaanPlot, 9
    - nice\_tidySEM, 12
    - write\_lavaan, 14
  - \* **modeling**
    - lavaan\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - write\_lavaan, 14
  - \* **path**
    - lavaan\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - write\_lavaan, 14
  - \* **plot**
    - cfa\_fit\_plot, 2
    - nice\_lavaanPlot, 9
    - nice\_tidySEM, 12
  - \* **structural**
    - lavaan\_cor, 3
    - lavaan\_cov, 4
    - lavaan\_ind, 5
    - lavaan\_reg, 7
    - nice\_fit, 8
    - write\_lavaan, 14
  - \* **table\_results**
    - nice\_tidySEM, 12
  - \* **tidySEM**
    - nice\_tidySEM, 12
- cfa\_fit\_plot, 2
- lavaan::cfa, 3
- lavaan::modindices, 11
- lavaan::parameterEstimates, 3–5, 7
- lavaan\_cor, 3
- lavaan\_cov, 4
- lavaan\_ind, 5
- lavaan\_reg, 7
- lavaanPlot::lavaanPlot, 10



nice\_fit, [8](#)  
nice\_lavaanPlot, [9](#)  
nice\_modindices, [10](#)  
nice\_tidySEM, [12](#)  
  
rempsysc::nice\_table, [4–8](#)  
  
stringdist::stringsim, [11](#)  
  
tidySEM::get\_layout, [12](#)  
tidySEM::prepare\_graph, [13](#)  
  
write\_lavaan, [14](#)