# Package: lvmPlot (via r-universe)

July 1, 2026

**Title** Publication-Quality Diagrams for Latent Variable Models

**Version** 0.1.0

**Description** Converts output from latent variable model tools into publication-ready path diagrams and model schematics. 'lavaan' fit objects and parameter tables are supported as a primary workflow, with graph adapters for objects from 'blavaan', 'lavaan.mi', 'semPlot', 'mirt', 'eRm', 'OpenMx', 'psych', 'poLCA', 'mclust', 'flexmix', 'lcmm', 'tidyLPA', and 'MplusAutomation' workflows when those packages are available. Supports structural equation and confirmatory factor analysis diagrams, multilevel structural equation models, growth models, higher-order factor models, latent class and profile models, item response theory models, and common mixture outputs through a unified graph grammar with model-aware defaults, geometry diagnostics, layout quality scoring, automatic layout selection, customizable publication styles, 'RStudio' preview, SVG/PDF/PNG export, 'TikZ' output, and reproducible publication bundles.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R (>= 4.1.0)

**Imports** methods

**Suggests** eRm, htmltools, jsonlite, lavaan, mclust, mirt, OpenMx, psych, ragg, rstudioapi, semPlot, shiny, svglite, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Feng Ji [aut, cre]

**Maintainer** Feng Ji <f.ji@utoronto.ca>

**Repository** https://trtfj.r-universe.dev

**Date/Publication** 2026-06-30 19:10:02 UTC

**RemoteUrl** https://github.com/cran/lvmPlot

**RemoteRef** HEAD

**RemoteSha** ca1cdc43a5a96cd2322cc654ab030f04e415ccc5

# Contents



---

`lvmPlot-package` *lvmPlot: publication-quality diagrams for latent variable models*

---

## Description

Converts latent variable model output into publication-ready path diagrams and model schematics. **lavaan** fit objects and parameter tables are supported as a primary workflow, with adapters for **blavaan**, **lavaan.mi**, **semPlot**, **mirt**, **eRm**, **OpenMx**, **psych**, **poLCA**, **mclust**, **flexmix**, **lcmm**, **tidyLPA**, and **MplusAutomation** workflows when those packages are available. The package supports structural equation and confirmatory factor analysis diagrams, multilevel structural equation models, growth models, higher-order factor models, latent class and profile schematics, item response theory models, OpenMx RAM models, and Mplus-style parameter output through a shared `lvm_graph` grammar with model-aware defaults, geometry diagnostics, customizable publication styles, RStudio preview, SVG/PDF/PNG export, and TikZ output.

## See Also

`plot_lvm`, `as_lvm_graph`, `lvm_graph`, `lvm_tikz`, `plot_sem`, `sem_tikz`

## as_sem_graph *Convert lavaan output to a graph object*

### Description

as_sem_graph() reads a lavaan fit object or a lavaan-style parameter table and returns the node, edge, and layout data used by sem_tikz().

### Usage

```
as_sem_graph(object, standardized = TRUE, layout = NULL,
  residuals = FALSE, covariances = TRUE)
```

### Arguments

| | |
|---|---|
| object | A lavaan fit object, or a data frame with at least lhs, op, and rhs columns. |
| standardized | Logical. If object is a lavaan fit, request standardized estimates from lavaan. |
| layout | NULL, "auto", or a data frame/matrix with name, x, and y coordinates. |
| residuals | Logical. Include variance self-loops from ~~ rows where lhs == rhs. |
| covariances | Logical. Include covariance paths from ~~ rows where lhs != rhs. |

### Value

A list with nodes, edges, and parameters, with class "sem_graph".

## check_lvm_layout *Check layout quality against publication gates*

### Description

check_lvm_layout() is the assertion-style companion to layout_quality(). It prepares a supported model object as an lvm_graph, computes geometry and optional label diagnostics, and checks the result against explicit publication-readiness gates. This is useful in R Markdown, CI, and manuscript-generation scripts where a diagram should fail early if nodes, paths, or labels collide.

### Usage

```
check_lvm_layout(object, min_score = 92,
  minimum_status = c("ready", "review", "repair"),
  max_node_overlaps = 0, max_edge_node_overlaps = 0,
  max_edge_crossings = NULL, max_label_overlaps = 0,
  max_label_node_overlaps = 0, max_label_edge_overlaps = 0,
  label = c("auto", "none", "std", "est", "both"), digits = 2,
```

```
  stars = "auto", respect_curves = TRUE,
  action = c("error", "warning", "message", "none"),
  layout = NULL,
  layout_family = c("auto", "sem", "bifactor", "irt", "mixture", "growth",
    "multilevel", "circle"),
  orientation = c("top-down", "bottom-up", "left-right", "right-left"),
  diagram = c("auto", "all", "measurement", "structural", "loadings",
    "paths", "covariances", "compact"),
  show = NULL, min_abs = NULL, significant = NULL, alpha = 0.05,
  routing = c("straight", "smart", "none"),
  node_style = NULL, edge_style = NULL, ...)
```

## Arguments

| | |
|---|---|
| `object` | A supported latent variable model object or `"lvm_graph"`. |
| `min_score` | Minimum acceptable `layout_quality()` score. |
| `minimum_status` | Minimum acceptable status. `"ready"` requires a publication-ready layout; `"review"` permits layouts that need human inspection; `"repair"` accepts any scored layout. |
| `max_node_overlaps`, `max_edge_node_overlaps`, `max_edge_crossings` | Maximum allowed hard node/node, edge/node, and edge/edge intersections. Set a limit to `NULL` to rely on the score/status gate instead. |
| `max_label_overlaps`, `max_label_node_overlaps`, `max_label_edge_overlaps` | Maximum allowed label collisions when labels are diagnosed. |
| `label`, `digits`, `stars`, `respect_curves` | Passed to `layout_quality()`. |
| `action` | What to do when the check fails: error, warning, message, or no signal. |
| `layout` | Optional custom layout. |
| `layout_family`, `orientation`, `diagram`, `show`, `min_abs`, `significant`, `alpha` | Passed to the graph-preparation pipeline. |
| `routing` | Edge routing mode used before scoring. |
| `node_style`, `edge_style` | Optional local style overrides used before scoring. |
| `...` | Passed to model adapters. |

## Value

A `lvmPlot_layout_quality` object with additional `passed`, `requirements`, `violations`, `graph`, and `label` fields.

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "f"),
  op = "=~",
  rhs = c("x1", "x2", "x3"),
  std.all = c(.7, .8, .75)
```

```
)
quality <- check_lvm_layout(params, label = "std", action = "none")
quality$passed
```

---

`export_lvm_bundle` *Export a publication bundle*

---

## Description

`export_lvm_bundle()` writes a complete manuscript-ready diagram bundle: vector/raster graphics, standalone TikZ, graph tables, diagnostics, quality report, and reproducibility metadata. It is intended to make the diagram a reproducible research artifact rather than a one-off image.

## Usage

```
export_lvm_bundle(object, dir, name = "lvmPlot-diagram",
  formats = c("pdf", "png", "svg", "tex"), width = "auto",
  height = "auto", res = 300, layout = NULL,
  layout_family = c("auto", "sem", "bifactor", "irt", "mixture", "growth",
    "multilevel", "circle"),
  orientation = c("top-down", "bottom-up", "left-right", "right-left"),
  diagram = c("auto", "all", "measurement", "structural", "loadings",
    "paths", "covariances", "compact"),
  show = NULL, min_abs = NULL, significant = NULL, alpha = 0.05,
  label = c("auto", "std", "est", "both", "none"), digits = 2,
  stars = "auto", node_labels = NULL, theme = lvm_theme_names(),
  style = NULL, node_style = NULL, edge_style = NULL,
  routing = c("straight", "smart", "none"),
  aspect = c("balanced", "preserve", "fill"), margin = 0.08,
  optimize = FALSE, optimize_layout_family = "auto",
  optimize_orientation = c("top-down", "left-right", "bottom-up",
    "right-left"),
  optimize_diagram = "auto", optimize_routing = "straight",
  tables = TRUE, diagnostics = TRUE, check = FALSE, check_min_score = 92,
  check_status = c("ready", "review", "repair"),
  check_max_edge_crossings = NULL,
  check_action = c("error", "warning", "message", "none"),
  metadata = TRUE, report = TRUE,
  compile_tex = FALSE,
  engine = c("pdflatex", "xelatex", "lualatex", "tectonic"), ...)
```

## Arguments

| | |
|---|---|
| `object` | A supported latent variable model object or `"lvm_graph"`. |
| `dir` | Output directory. |
| `name` | File stem used for exported artifacts. |

| | |
|---|---|
| formats | Character vector of graphic formats. Supported values are "pdf", "png", "svg", "tex", and "tikz". |
| width, height | Device size in inches, or "auto" to use lvm_canvas_size(). |
| res | PNG resolution in pixels per inch. |
| layout | Optional custom layout. |
| layout_family | Layout family. |
| orientation | Diagram orientation. |
| diagram | Diagram subset. |
| show | Optional edge types to show. |
| min_abs | Optional absolute loading/path threshold. |
| significant | Logical. Keep only significant estimated edges. |
| alpha | Significance threshold. |
| label | Edge label style. |
| digits | Number of digits for edge labels. |
| stars | Significance-star policy. |
| node_labels | Optional node relabeling vector or function. |
| theme | Diagram theme. |
| style | Style overrides from lvm_style(). |
| node_style, edge_style | Optional per-node and per-edge style tables. |
| routing | Edge routing mode. |
| aspect | Plot coordinate scaling. |
| margin | Outer plot margin. |
| optimize | Logical. If TRUE, evaluate candidate layouts with select_lvm_layout() before export. |
| optimize_layout_family, optimize_orientation, optimize_diagram, optimize_routing | Candidate settings used when optimize = TRUE. |
| tables | Logical. Write node and edge CSV tables. |
| diagnostics | Logical. Write diagnostic and quality CSV files. |
| check | Logical. If TRUE, fail, warn, or message when the layout does not satisfy publication-readiness gates. |
| check_min_score | Minimum acceptable quality score when check = TRUE. |
| check_status | Minimum acceptable status when check = TRUE. |
| check_max_edge_crossings | Optional maximum edge crossings when check = TRUE. Defaults to NULL, so score/status gates decide whether crossings are acceptable. |
| check_action | Signal used when check = TRUE and the layout fails. |
| metadata | Logical. Write reproducibility metadata and session info. |

| | |
|---|---|
| report | Logical. Write a human-readable Markdown report. |
| compile_tex | Logical. Compile the TikZ file when "tex"/ "tikz" is exported and a TeX engine is available. |
| engine | TeX engine used when compile_tex = TRUE. |
| ... | Passed to model adapters. |

## Value

A list with exported files, quality summary, graph, and device size, with class "lvmPlot_bundle".

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "f"),
  op = "=~",
  rhs = c("x1", "x2", "x3"),
  std.all = c(.7, .8, .75)
)
out <- tempfile("lvmPlot-bundle-")
export_lvm_bundle(
  params,
  out,
  name = "cfa",
  formats = c("pdf", "tex"),
  check = TRUE,
  optimize = TRUE,
  optimize_orientation = c("top-down", "left-right")
)
```

---

# layout_diagnostics *Diagnose a diagram layout*

---

## Description

layout_diagnostics() scores the current node coordinates before drawing. It is useful for checking whether a complex SEM/LVM layout still has node overlaps, straight-edge crossings, edge/node collisions, or estimated parameter label collisions before export.

## Usage

```
layout_diagnostics(
  object,
  respect_curves = TRUE,
  label = c("none", "auto", "std", "est", "both"),
  digits = 2,
  stars = "auto"
)
```

### Arguments

| | |
|---|---|
| object | An "lvm_graph", "sem_graph", or list with nodes and edges. |
| respect_curves | Logical. If TRUE, edges already marked with a curvature are not counted as straight edge/node overlaps. |
| label | Edge label mode used for optional label diagnostics. The default "none" preserves fast geometry-only diagnostics. Use "std", "est", "both", or "auto" to estimate label boxes and report label collisions. |
| digits | Number of digits for estimated label text. |
| stars | Significance-star policy used when label is not "none". |

### Value

A list with node overlap counts, edge crossing counts, edge/node overlap counts, optional edge-label collision counts, edge length summaries, and detail data frames, with class "lvmPlot_layout_diagnostics".

### Examples

```
nodes <- data.frame(
  name = c("a", "b", "middle"),
  type = "observed",
  x = c(0, 2, 1),
  y = c(0, 0, 0)
)
edges <- data.frame(from = "a", to = "b", type = "path")
graph <- lvm_graph(nodes, edges)
layout_diagnostics(graph, respect_curves = FALSE)
layout_diagnostics(graph, respect_curves = FALSE, label = "est")
```

---

## layout_matrix *Create a diagram layout from a matrix*

---

### Description

layout_matrix() converts a familiar SEM plotting matrix into explicit node coordinates. Rows are interpreted from top to bottom and columns from left to right.

### Usage

```
layout_matrix(x, x_spacing = 1.8, y_spacing = 1.7)
```

### Arguments

| | |
|---|---|
| x | A matrix or data frame whose non-empty cells contain node names. Empty cells may be NA, "", or ".". |
| x_spacing, y_spacing | Spacing between columns and rows in diagram coordinates. |

## Value

A data frame with name, x, and y columns.

## Examples

```
layout <- layout_matrix(matrix(
  c(NA, "f", NA,
    "x1", ".", "x2"),
  nrow = 2,
  byrow = TRUE
))
layout
```

---

# layout_quality *Score layout quality*

---

## Description

layout_quality() turns layout_diagnostics() into a compact quality score and issue table. It is designed for publication workflows where a diagram should be checked before being exported or submitted.

## Usage

```
layout_quality(object, label = c("none", "auto", "std", "est", "both"),
  digits = 2, stars = "auto", respect_curves = TRUE)
```

## Arguments

| | |
|---|---|
| object | A "lvm_graph", "sem_graph", or list with nodes and edges. |
| label | Edge label mode used for optional label diagnostics. |
| digits | Number of digits for estimated label text. |
| stars | Significance-star policy used when label is not "none". |
| respect_curves | Logical. If TRUE, edges marked with curvature are not counted as straight edge/node overlaps. |

## Value

A list with score, grade, status, issues, and diagnostics, with class "lvmPlot_layout_quality".

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "f"),
  op = "=~",
  rhs = c("x1", "x2", "x3"),
  std.all = c(.7, .8, .75)
)
quality <- layout_quality(as_lvm_graph(params), label = "std")
quality
```

---

`lvm_canvas_size` *Suggest a canvas size for an LVM diagram*

---

## Description

`lvm_canvas_size()` inspects the prepared graph coordinates and returns a recommended device size. It helps dense MIRT/LCA diagrams get wider canvases and multilevel diagrams get taller canvases without manually guessing export dimensions.

## Usage

```
lvm_canvas_size(object, ..., min_width = 6, max_width = 14,
  min_height = 4, max_height = 9.5)
```

## Arguments

| | |
|---|---|
| `object` | A supported latent variable model object or `"lvm_graph"`. |
| `...` | Plot/layout arguments passed as they would be to `plot_lvm()`, such as `diagram`, `orientation`, `layout_family`, `layout`, `node_labels`, `node_style`, `edge_style`, `residuals`, or `covariances`. |
| `min_width, max_width` | Width bounds in inches. |
| `min_height, max_height` | Height bounds in inches. |

## Value

A named numeric vector with `width` and `height`, in inches.

## Examples

```
params <- data.frame(
  lhs = c("visual", "visual", "visual", "textual", "textual", "textual"),
  op = "=~",
  rhs = paste0("x", 1:6),
  est = .7
)
lvm_canvas_size(params)
```

## lvm_graph *Latent variable model graph grammar*

### Description

`lvm_graph()` is the common diagram grammar used by SEM, LCA/LPA, multilevel SEM, IRT/MIRT, growth, mixture, and adapter-based model diagrams. Model-specific adapters convert fitted model objects to this graph structure before drawing.

### Usage

```
lvm_graph(nodes, edges = data.frame(), model_type = "lvm",
  layout_family = NULL, title = NULL, meta = list())

as_lvm_graph(object, ...)
```

### Arguments

| | |
|---|---|
| `nodes` | Data frame of nodes. Needs a `name` or `id` column. |
| `edges` | Data frame of edges with `from` and `to` columns. |
| `model_type` | Short model family label. |
| `layout_family` | Layout family used when coordinates are absent. |
| `title` | Optional graph title. |
| `meta` | Optional metadata list. |
| `object` | A supported model object, parameter table, or graph object. |
| `...` | Passed to methods. |

### Value

An object of class `"lvm_graph"`.

## lvm_style *Create diagram style overrides*

### Description

Creates style overrides for `plot_lvm()`, `plot_sem()`, `lvm_tikz()`, and `sem_tikz()`.

## Usage

```
lvm_style(scale = NULL, node_scale = NULL, edge_scale = NULL,
  font_scale = NULL, node_font_size = NULL, edge_font_size = NULL,
  font_family = NULL, latent_size = NULL, observed_width = NULL,
  observed_height = NULL, node_line_width = NULL,
  edge_line_width = NULL, path_line_width = NULL,
  loading_line_width = NULL, covariance_line_width = NULL,
  residual_line_width = NULL, node_color = NULL,
  latent_color = NULL, observed_color = NULL, node_fill = NULL,
  latent_fill = NULL, observed_fill = NULL, edge_color = NULL,
  path_color = NULL, loading_color = NULL, covariance_color = NULL,
  residual_color = NULL, node_text_color = NULL, label_color = NULL,
  label_fill = NULL)
```

## Arguments

| | |
|---|---|
| `scale` | Overall multiplier for node dimensions, line widths, and text sizes. |
| `node_scale` | Multiplier for latent and observed node dimensions. |
| `edge_scale` | Multiplier for edge line widths. |
| `font_scale` | Multiplier for node and edge label font sizes. |
| `node_font_size` | Node label font size in points. |
| `edge_font_size` | Edge label font size in points. |
| `font_family` | Font family for grid output. For TikZ, use `"sans"`, `"serif"`, or a raw LaTeX font command. |
| `latent_size` | Latent node diameter in millimeters. |
| `observed_width`, `observed_height` | Observed node size in millimeters. |
| `node_line_width` | Stroke width for latent and observed nodes. |
| `edge_line_width` | Stroke width for all edges. |
| `path_line_width`, `loading_line_width`, `covariance_line_width`, `residual_line_width` | Stroke widths for specific edge types. |
| `node_color`, `latent_color`, `observed_color` | Stroke colors. |
| `node_fill`, `latent_fill`, `observed_fill` | Fill colors. |
| `edge_color`, `path_color`, `loading_color`, `covariance_color`, `residual_color` | Edge colors. |
| `node_text_color` | Node-label text color. |
| `label_color`, `label_fill` | Edge-label text and background colors. |

## Value

A named list with class "lvm_style".

## Examples

```
style <- lvm_style(
  scale = 1.08,
  font_scale = 0.95,
  node_font_size = 12,
  edge_font_size = 9,
  latent_size = 16,
  observed_width = 20,
  node_fill = "#F8FAFC",
  edge_line_width = 1.1
)
```

---

# lvm_themes *List built-in diagram themes*

---

## Description

lvm_themes() returns the built-in theme presets accepted by plot_lvm(), plot_sem(), lvm_tikz(), and sem_tikz(). The default theme is "journal".

## Usage

```
lvm_themes()
```

## Value

A data frame with theme names and short descriptions.

## Examples

```
lvm_themes()
```

---

# lvm_tikz *TikZ output for latent variable model diagrams*

---

## Description

Creates editable TikZ output from the unified LVM graph grammar.

## Usage

```
lvm_tikz(object, layout = NULL,
  layout_family = c("auto", "sem", "bifactor", "irt", "mixture", "growth",
  "multilevel", "circle"), orientation = c("top-down", "bottom-up",
  "left-right", "right-left"), diagram = c("auto", "all", "measurement",
  "structural", "loadings", "paths", "covariances", "compact"),
  show = NULL, min_abs = NULL, significant = NULL, alpha = 0.05,
  label = c("auto", "std", "est", "both", "none"), digits = 2,
  stars = "auto",
  node_labels = NULL, standalone = FALSE, theme = lvm_theme_names(),
  style = NULL, node_style = NULL, edge_style = NULL,
  routing = c("straight", "smart", "none"),
  escape = TRUE, file = NULL, ...)

write_lvm_tikz(object, file, standalone = TRUE, compile = FALSE,
  engine = c("pdflatex", "xelatex", "lualatex", "tectonic"), clean = TRUE,
  ...)
```

## Arguments

| | |
|---|---|
| object | A supported model object, "lvm_graph", or TikZ string. |
| layout | Optional custom layout, either a data frame with name, x, and y, or a matrix passed through layout_matrix(). |
| layout_family | Layout preset, including SEM, bifactor, IRT, mixture, growth, multilevel, and circle layouts. |
| orientation | Diagram direction. |
| diagram | Diagram preset used to filter edge types. "auto" compacts dense probability/profile matrices; use "all" to draw every edge. |
| show | Optional explicit edge-type or diagram-type set to show. |
| min_abs | Optional minimum absolute estimate/standardized estimate. |
| significant | If TRUE, keep only significant edges when p-values are available. |
| alpha | Significance threshold. |
| label | Edge label style: automatic, standardized estimates, raw estimates, both, or no labels. "auto" keeps default diagrams clean by hiding automatically estimated parameters while preserving explicit custom edge labels on ordinary diagrams; use "std", "est", or "both" to show coefficients explicitly. |
| digits | Number of digits for numeric labels. |
| stars | Significance-star policy. "auto" keeps default diagrams clean, "fit" shows stars only when they fit the edge, "always" forces stars, and "none"/FALSE suppresses them. TRUE is treated as "fit". |
| node_labels | Optional named node labels. |
| standalone | Logical. Wrap in a standalone LaTeX document. |
| theme | Diagram theme. Use lvm_themes() to list built-in presets. |
| style | Optional style overrides created by lvm_style() or a named list. |

| | |
|---|---|
| node_style | Optional per-node style data frame or named list. Fields are emitted as local TikZ node options, including shape, fill, color, text_color, font_size, font_family, width, height, size, and line_width. |
| edge_style | Optional per-edge style data frame or named list. Fields are emitted as local TikZ draw and label options, including label, color, line_width, linetype, curvature, arrow, label_position, label_size, label_color, label_fill, and label_font_family. |
| routing | Edge routing mode. "straight" keeps model edges as straight segments. "smart" can add subtle curvature for users who explicitly want automatic edge avoidance. |
| escape | Logical. Escape node labels and plain-text custom edge labels for LaTeX. Labels already written as LaTeX math or commands are preserved. |
| file | Output file path. |
| compile | Logical. Compile with a local TeX engine. |
| engine | TeX engine. |
| clean | Logical. Remove common TeX auxiliary files. |
| ... | Passed to as_lvm_graph() or lvm_tikz(). |

## Value

lvm_tikz() returns a character scalar. write_lvm_tikz() invisibly returns the normalized output path.

---

# lvmPlot *Open a latent variable model diagram*

---

## Description

lvmPlot() is the high-level entry point for the package. In an interactive R session it asks whether to open the browser editor or draw a static plot, with the editor offered first because it is the recommended path for final manuscript figures. In non-interactive sessions it falls back to a static plot so scripts, examples, and package checks never wait for input. Use mode = "edit" to open the editor directly or mode = "plot" for a static diagram.

edit_lvm() is a short alias for opening the interactive editor directly.

## Usage

```
lvmPlot(
  object,
  ...,
  mode = c("ask", "edit", "plot"),
  label = c("auto", "std", "est", "both", "none"),
  digits = 2,
  stars = "auto",
```

```
  theme = lvm_theme_names(),
  style = NULL,
  export_name = "lvmPlot-diagram",
  launch = interactive()
)

edit_lvm(
  object,
  ...,
  label = c("auto", "std", "est", "both", "none"),
  digits = 2,
  stars = "auto",
  theme = lvm_theme_names(),
  style = NULL,
  export_name = "lvmPlot-diagram",
  launch = interactive()
)
```

## Arguments

| | |
|---|---|
| object | A supported latent variable model object or an "lvm_graph". |
| ... | Arguments passed to plot_lvm() or lvmPlot_editor(), such as diagram, layout_family, orientation, show, min_abs, or routing. |
| mode | Opening mode. "ask" prompts in interactive sessions and uses "plot" otherwise; "edit" opens the browser editor; "plot" draws the static diagram. |
| label | Edge label mode. |
| digits | Number of digits for edge labels. |
| stars | Significance-star policy. |
| theme | Plot theme. Use lvm_themes() to list built-in themes. |
| style | Optional style overrides created by lvm_style() or a named list. |
| export_name | Default file stem used by the editor downloads. |
| launch | Logical. If TRUE, run the editor when mode = "edit". If FALSE, return the Shiny app object without launching it; this is useful for tests and scripted validation. |

## Value

For mode = "plot", invisibly returns the plotted "lvm_graph". For mode = "edit", returns the Shiny app object invisibly when launched or visibly when launch = FALSE.

## Examples

```
params <- data.frame(
  lhs = c("engage", "engage", "engage", "achieve"),
  op = c("=~", "=~", "=~", "~"),
  rhs = c("item1", "item2", "item3", "engage"),
  std.all = c(.78, .72, .69, .46),
  pvalue = c(.001, .001, .002, .004)
```

```
)

graph <- lvmPlot(params, mode = "plot", label = "std")
class(graph)

if (requireNamespace("shiny", quietly = TRUE) &&
    requireNamespace("jsonlite", quietly = TRUE)) {
  app <- lvmPlot(params, mode = "edit", label = "std", launch = FALSE)
  class(app)
}

if (interactive() && requireNamespace("shiny", quietly = TRUE) &&
    requireNamespace("jsonlite", quietly = TRUE)) {
  lvmPlot(params, mode = "edit", label = "std")
}
```

---

`lvmPlot_editor` *Interactively edit and export a lvmPlot diagram*

---

## Description

`lvmPlot_editor()` opens a small Shiny application for final manual adjustment of an automatically prepared LVM diagram. Nodes can be selected, dragged, nudged with the keyboard, snapped to a grid, locked, aligned, distributed, relabeled, repaired, undone/redone, and polished in the browser preview. Coefficient labels can also be dragged for final manual placement, or double-clicked to return them to automatic placement. Edge-label, theme, preset, color, size, line-width, and font changes update the preview before export; the edited coordinates, coefficient-label positions, node labels, and style can then be exported as SVG/PDF/PNG, reusable R code, or a JSON editor state that can be loaded later to continue editing.

## Usage

```
lvmPlot_editor(
  object,
  ...,
  label = c("auto", "std", "est", "both", "none"),
  digits = 2,
  stars = "auto",
  theme = lvm_theme_names(),
  style = NULL,
  export_name = "lvmPlot-diagram",
  launch = interactive()
)
```

## Arguments

| | |
|---|---|
| `object` | A supported latent variable model object or an `"lvm_graph"`. |

| | |
|---|---|
| `...` | Arguments passed to `prepare_lvm_graph()`, such as `diagram`, `layout_family`, `orientation`, `show`, `min_abs`, or `alpha`. |
| `label` | Edge label mode used for preview and default export. |
| `digits` | Number of digits for edge labels. |
| `stars` | Significance-star policy. |
| `theme` | Initial plot theme for exported files. |
| `style` | Optional style overrides created by `lvm_style()` or a named list. The editor preview uses a lightweight browser renderer; exported files use the full lvmPlot renderer and this style. |
| `export_name` | Default file stem for downloaded artifacts. |
| `launch` | Logical. If `TRUE`, run the Shiny app. If `FALSE`, return the app object without launching it; useful for tests. |

## Value

A Shiny app object invisibly when launched, or visibly when `launch = FALSE`.

## See Also

`plot_lvm`, `save_lvm_svg`

## Examples

```
if (interactive() && requireNamespace("shiny", quietly = TRUE) &&
    requireNamespace("jsonlite", quietly = TRUE)) {
  lvmPlot_editor(
    data.frame(
      lhs = c("f", "f", "y"),
      op = c("=~", "=~", "~"),
      rhs = c("x1", "x2", "f"),
      std.all = c(.7, .8, .4)
    )
  )
}
```

---

# `lvmPlot_rstudio_preview` *Preview a lvmPlot diagram from the RStudio editor selection*

---

## Description

These helpers are intended for RStudio Addins. Select an object name or an R expression that evaluates to a supported model, then run the addin to draw the diagram in the Plots pane or export TikZ code.

### Usage

```
lvmPlot_rstudio_preview(object = NULL, ...)

lvmPlot_rstudio_export_tikz(object = NULL, file = "lvmPlot-diagram.tex", ...)
```

### Arguments

| | |
|---|---|
| object | Optional object. When NULL, the current RStudio editor selection is evaluated in .GlobalEnv. |
| ... | Passed to plot_lvm() or write_lvm_tikz(). |
| file | Output .tex path for lvmPlot_rstudio_export_tikz(). |

### Value

Invisibly returns the plotted graph or output file.

---

## plot_lvm *Draw and save latent variable model diagrams*

---

### Description

Draws latent variable model diagrams in the RStudio Plots pane and exports the same graph as SVG, PDF, or PNG.

### Usage

```
plot_lvm(object, layout = NULL,
  layout_family = c("auto", "sem", "bifactor", "irt", "mixture", "growth",
  "multilevel", "circle"), orientation = c("top-down", "bottom-up",
  "left-right", "right-left"), diagram = c("auto", "all", "measurement",
  "structural", "loadings", "paths", "covariances", "compact"),
  show = NULL, min_abs = NULL, significant = NULL, alpha = 0.05,
  label = c("auto", "std", "est", "both", "none"), digits = 2,
  stars = "auto",
  node_labels = NULL, theme = lvm_theme_names(), style = NULL,
  node_style = NULL, edge_style = NULL,
  routing = c("straight", "smart", "none"),
  aspect = c("balanced", "preserve", "fill"), margin = 0.08, ...)

## S3 method for class 'lvm_graph'
plot(x, ...)

save_lvm_svg(object, file, width = "auto", height = "auto", ...)

save_lvm_pdf(object, file, width = "auto", height = "auto", ...)

save_lvm_png(object, file, width = "auto", height = "auto", res = 240, ...)
```

## Arguments

| | |
|---|---|
| object | A supported model object or "lvm_graph". |
| layout | Optional custom layout with name, x, and y, or a matrix passed through layout_matrix(). |
| layout_family | Layout preset: automatic, SEM, bifactor, IRT, mixture, growth, multilevel, or circle. |
| orientation | Diagram direction. |
| diagram | Diagram preset used to filter edge types. "auto" compacts dense probability/profile matrices; use "all" to draw every edge. |
| show | Optional explicit edge-type or diagram-type set to show. |
| min_abs | Optional minimum absolute estimate/standardized estimate. |
| significant | If TRUE, keep only edges significant at alpha when p-values are available. |
| alpha | Significance threshold. |
| label | Edge label style: automatic, standardized estimates, raw estimates, both, or no labels. "auto" keeps default diagrams clean by hiding automatically estimated parameters while preserving explicit custom edge labels on ordinary diagrams; use "std", "est", or "both" to show coefficients explicitly. |
| digits | Number of digits for numeric edge labels. |
| stars | Significance-star policy. "auto" keeps default diagrams clean, "fit" shows stars only when they fit the edge, "always" forces stars, and "none"/FALSE suppresses them. TRUE is treated as "fit". |
| node_labels | Optional named node labels. |
| theme | Diagram theme. Use lvm_themes() to list built-in presets. |
| style | Optional style overrides created by lvm_style() or a named list. |
| node_style | Optional per-node style data frame or named list. Data-frame columns can include name, label, shape, fill, color, text_color, font_size, font_family, width, height, size, and line_width. |
| edge_style | Optional per-edge style data frame or named list. Data-frame columns can include from, to, optional type, label, color, line_width, linetype, curvature, arrow, label_size, label_color, label_fill, and label_font_family. |
| routing | Edge routing mode. "straight" keeps model edges as straight segments. "smart" can add subtle curvature for users who explicitly want automatic edge avoidance. |
| aspect | Plot coordinate scaling. "balanced" avoids extreme x/y distortion, "preserve" keeps coordinate units visually equal for the current device, and "fill" uses the historical full-panel stretch. |
| margin | Outer plot margin. |
| ... | Passed to as_lvm_graph() or plot_lvm(). |
| x | An "lvm_graph" object. |
| file | Output file path. |
| width, height | Device size in inches, or "auto" to use lvm_canvas_size(). |
| res | PNG output resolution in pixels per inch. |

## Value

`plot_lvm()` invisibly returns the graph. Save helpers invisibly return the normalized output path.

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "y"),
  op = c("=~", "=~", "~"),
  rhs = c("x1", "x2", "f"),
  est = c(1, .8, .4)
)
plot_lvm(
  params,
  layout = matrix(c(NA, "y", NA, NA, "f", NA, "x1", ".", "x2"),
                  nrow = 3, byrow = TRUE),
  node_style = data.frame(name = "f", shape = "diamond", fill = "#EEF2FF"),
  edge_style = data.frame(from = "f", to = "y", label = "beta",
                          linetype = "dashed", curvature = -.18)
)
```

---

`plot_sem` *Draw and save SEM path diagrams in R*

---

## Description

`plot_sem()` uses base R grid graphics, so a fitted lavaan model can be previewed directly in the RStudio Plots pane. The save helpers write the same diagram as SVG, PDF, or PNG.

## Usage

```
plot_sem(object, layout = NULL, label = c("auto", "std", "est", "both", "none"),
  digits = 2, stars = "auto", residuals = FALSE, covariances = TRUE,
  node_labels = NULL, theme = lvm_theme_names(), style = NULL,
  node_style = NULL, edge_style = NULL,
  routing = c("straight", "smart", "none"),
  aspect = c("balanced", "preserve", "fill"), margin = 0.08, ...)

## S3 method for class 'sem_graph'
plot(x, ...)

save_sem_svg(object, file, width = 8, height = 4.8, ...)

save_sem_pdf(object, file, width = 8, height = 4.8, ...)

save_sem_png(object, file, width = 8, height = 4.8, res = 240, ...)
```

## Arguments

| | |
|---|---|
| object | A lavaan fit object, a lavaan-style parameter table, or an object returned by as_sem_graph(). |
| layout | Optional custom layout. See as_sem_graph(), or a matrix passed through layout_matrix(). |
| label | Edge label style: automatic, standardized estimates, raw estimates, both, or no labels. "auto" keeps default diagrams clean by hiding automatically estimated parameters while preserving explicit custom edge labels on ordinary diagrams; use "std", "est", or "both" to show coefficients explicitly. |
| digits | Number of digits for edge labels. |
| stars | Significance-star policy. "auto" keeps default diagrams clean, "fit" shows stars only when they fit the edge, "always" forces stars, and "none"/FALSE suppresses them. TRUE is treated as "fit". |
| residuals | Logical. Draw variance self-loops. |
| covariances | Logical. Draw covariance paths. |
| node_labels | Optional named character vector, unnamed vector, or function used to relabel nodes. |
| theme | Diagram theme. Use lvm_themes() to list built-in presets. |
| style | Optional style overrides created by lvm_style() or a named list. |
| node_style | Optional per-node style data frame or named list. Columns can include name, label, shape, fill, color, text_color, font_size, font_family, width, height, size, and line_width. |
| edge_style | Optional per-edge style data frame or named list. Columns can include from, to, optional type, label, color, line_width, linetype, curvature, arrow, label_size, label_color, label_fill, and label_font_family. |
| routing | Edge routing mode. "straight" keeps model edges as straight segments. "smart" can add subtle curvature for users who explicitly want automatic edge avoidance. |
| aspect | Plot coordinate scaling. "balanced" avoids extreme x/y distortion, "preserve" keeps coordinate units visually equal for the current device, and "fill" uses the historical full-panel stretch. |
| margin | Outer plot margin, in normalized parent coordinates. |
| ... | Reserved for future extensions, or passed to plot_sem() by the save helpers. |
| x | A "sem_graph" object. |
| file | Output file path. |
| width, height | Device size in inches. |
| res | PNG output resolution in pixels per inch. |

## Value

plot_sem() invisibly returns the "sem_graph" object used for drawing. The save helpers invisibly return the normalized output path.

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "y"),
  op = c("=~", "=~", "~"),
  rhs = c("x1", "x2", "f"),
  est = c(1, .8, .4),
  std.all = c(.7, .6, .35),
  pvalue = c(NA, .001, .01)
)
plot_sem(params)
```

---

# select_lvm_layout *Select the best layout automatically*

---

## Description

`select_lvm_layout()` evaluates a grid of candidate layout settings and chooses the candidate with the best `layout_quality()` score. It lets users ask lvmPlot to try common orientation and routing alternatives before exporting a figure.

## Usage

```
select_lvm_layout(object, layout_family = "auto",
  orientation = c("top-down", "left-right", "bottom-up", "right-left"),
  diagram = "auto", routing = "straight", show = NULL, min_abs = NULL,
  significant = NULL, alpha = 0.05,
  label = c("auto", "std", "est", "both", "none"), digits = 2,
  stars = "auto", node_labels = NULL, node_style = NULL,
  edge_style = NULL, ...)
```

## Arguments

| | |
|---|---|
| `object` | A supported latent variable model object or `"lvm_graph"`. |
| `layout_family` | Candidate layout families. |
| `orientation` | Candidate orientations. |
| `diagram` | Candidate diagram subsets. |
| `routing` | Candidate routing modes. |
| `show` | Optional edge types to show. |
| `min_abs` | Optional absolute loading/path threshold. |
| `significant` | Logical. Keep only significant estimated edges. |
| `alpha` | Significance threshold. |
| `label` | Edge label style used for scoring. |
| `digits` | Number of digits for edge labels. |
| `stars` | Significance-star policy. |

| | |
|---|---|
| node_labels | Optional node relabeling vector or function. |
| node_style, edge_style | Optional per-node and per-edge style tables. |
| ... | Passed to model adapters. |

## Value

A list with the selected graph, quality object, best candidate row, and candidate table, with class "lvmPlot_layout_selection".

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "f"),
  op = "=~",
  rhs = c("x1", "x2", "x3"),
  std.all = c(.7, .8, .75)
)
selection <- select_lvm_layout(
  params,
  orientation = c("top-down", "left-right"),
  label = "std"
)
selection
plot_lvm(selection$graph, label = "std")
```

---

# sem_tikz *Render a SEM path diagram as TikZ*

---

## Description

sem_tikz() renders measurement paths, structural regressions, and covariances from lavaan output as editable TikZ code.

## Usage

```
sem_tikz(object, layout = NULL, label = c("auto", "std", "est", "both", "none"),
  digits = 2, stars = "auto", residuals = FALSE, covariances = TRUE,
  node_labels = NULL, standalone = FALSE, theme = lvm_theme_names(),
  style = NULL, node_style = NULL, edge_style = NULL,
  routing = c("straight", "smart", "none"), escape = TRUE, file = NULL, ...)
```

## Arguments

| | |
|---|---|
| object | A lavaan fit object, a lavaan-style parameter table, or an object returned by as_sem_graph(). |
| layout | Optional custom layout. See as_sem_graph(), or a matrix passed through layout_matrix(). |

| | |
|---|---|
| label | Edge label style: automatic, standardized estimates, raw estimates, both, or no labels. "auto" keeps default diagrams clean by hiding automatically estimated parameters while preserving explicit custom edge labels on ordinary diagrams; use "std", "est", or "both" to show coefficients explicitly. |
| digits | Number of digits for edge labels. |
| stars | Significance-star policy. "auto" keeps default diagrams clean, "fit" shows stars only when they fit the edge, "always" forces stars, and "none"/FALSE suppresses them. TRUE is treated as "fit". |
| residuals | Logical. Draw variance self-loops. |
| covariances | Logical. Draw covariance paths. |
| node_labels | Optional named character vector, unnamed vector, or function used to relabel nodes. |
| standalone | Logical. Wrap the TikZ picture in a standalone LaTeX document. |
| theme | Diagram theme. Use lvm_themes() to list built-in presets. |
| style | Optional style overrides created by lvm_style() or a named list. |
| node_style | Optional per-node style data frame or named list. The same node style fields used by plot_sem() are emitted as local TikZ node options. |
| edge_style | Optional per-edge style data frame or named list. The same edge style fields used by plot_sem() are emitted as local TikZ draw and coefficient-label options. |
| routing | Edge routing mode. "straight" keeps model edges as straight segments. "smart" can add subtle curvature for users who explicitly want automatic edge avoidance. |
| escape | Logical. Escape node labels and plain-text custom edge labels for LaTeX. Labels already written as LaTeX math or commands are preserved. |
| file | Optional path to write the generated code. |
| ... | Reserved for future extensions. |

## Value

A character scalar with class "sem_tikz".

## Examples

```
params <- data.frame(
  lhs = c("f", "f", "y"),
  op = c("=~", "=~", "~"),
  rhs = c("x1", "x2", "f"),
  est = c(1, .8, .4),
  std.all = c(.7, .6, .35),
  pvalue = c(NA, .001, .01)
)
sem_tikz(params)
```

`write_sem_tikz` *Write a TikZ SEM diagram to disk*

## Description

`write_sem_tikz()` writes a TikZ path diagram to a `.tex` file and can optionally compile it when a local TeX engine is available.

## Usage

```
write_sem_tikz(object, file, standalone = TRUE, compile = FALSE,
  engine = c("pdflatex", "xelatex", "lualatex", "tectonic"), clean = TRUE,
  ...)
```

## Arguments

| | |
|---|---|
| `object` | A lavaan fit object, lavaan-style parameter table, `"sem_graph"`, `"sem_tikz"`, or character TikZ code. |
| `file` | Output `.tex` path. |
| `standalone` | Logical. If `TRUE`, ensure a standalone LaTeX document is written. |
| `compile` | Logical. Compile the `.tex` file with a local TeX engine. |
| `engine` | TeX engine used when `compile = TRUE`. |
| `clean` | Logical. Remove common TeX auxiliary files after successful compilation. |
| `...` | Passed to `sem_tikz()` when `object` is not already TikZ code. |

## Value

Invisibly returns the normalized output path.

# Index