

# Package: diy.sem.plot (via r-universe)

October 3, 2026

**Type** Package

**Title** Manually Plot Path Diagrams for Structural Equation Models

**Version** 1.0.0.9000

**Description** Manually plot fully customisable path diagrams for structural equation models (SEM). Map out node positions using simple coordinates and specify where on the perimeter of each node paths begin and end. Extensive fine-tuning options allow the creation of a path diagram exactly as envisioned, entirely within R.

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Imports** ggplot2, ggtext, ggforce, lavaan, patchwork

**Suggests** knitr, rmarkdown, semPlot, testthat

**Config/testthat/edition** 3

**URL** <https://github.com/snagy86/diy.sem.plot>

**BugReports** <https://github.com/snagy86/diy.sem.plot/issues>

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**Config/roxygen2/version** 8.1.0

**Config/pak/sysreqs**

libfontconfig1-dev libfreetype6-dev libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

**Repository** <https://snagy86.r-universe.dev>

**Date/Publication** 2026-10-03 13:40:50 UTC

**RemoteUrl** <https://github.com/snagy86/diy.sem.plot>

**RemoteRef** HEAD

**RemoteSha** 9098d41057abc7bf43e2b49d114a3b6aa4c817ce

Contents

|                       |           |
|-----------------------|-----------|
| diyPaths . . . . .    | 2         |
| node . . . . .        | 7         |
| panel_note . . . . .  | 8         |
| panel_title . . . . . | 9         |
| path . . . . .        | 10        |
| template . . . . .    | 11        |
| <b>Index</b>          | <b>13</b> |

---

|          |                                                                   |
|----------|-------------------------------------------------------------------|
| diyPaths | <i>Manually plot path diagrams for structural equation models</i> |
|----------|-------------------------------------------------------------------|

---

Description

Plot fully customisable path diagrams for structural equation models fitted with **lavaan**, rendered using **ggplot2**.

To render the ggplot object of the diagram, the function only requires users to supply a fitted lavaan model, specify node positions using x-y coordinates, and detail where on the perimeter of each node (top, bottom, left, or right) each path should begin and end. The render automatically inserts estimates centered on the midpoint and adjusts each node’s shape to match its variable type. A range of optional fine-tuning arguments are included, facilitating the creation of a path diagram exactly as you envision it, entirely within R.

Usage

```
diyPaths(  
  fit,  
  node_positions,  
  path_positions,  
  standardised = FALSE,  
  digits = 3,  
  est_stars = FALSE,  
  est_p = FALSE,  
  est_ci = FALSE,  
  show_variances = FALSE,  
  variance_stars = FALSE,  
  variance_p = FALSE,  
  variance_ci = FALSE,  
  sig_linetype = FALSE,  
  p_threshold = 0.05,  
  node_width = 1.5,  
  node_height = 1,  
  latent_node_size_adjust = 1,  
  observed_node_size_adjust = 1,  
  latent_node_text_size = 4,  
  observed_node_text_size = 4,  
)
```

```

    path_transparent_text = FALSE,
    path_text_size = 3.5,
    line_thickness = 0.6,
    arrow_size = 0.2,
    panel_cols = NULL,
    show_group_labels = FALSE,
    title_text_size = 25,
    panel_titles = NULL,
    panel_notes = NULL,
    note_text_size = 4,
    note_transparent_text = FALSE,
    note_outline = TRUE,
    show_grid = FALSE,
    grid_axis_scale = 1,
    margin_x = 0.5,
    margin_y = 0.5,
    look_up_table = FALSE
  )

```

## Arguments

|                             |                                                                                                                                     |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <code>fit</code>            | A fitted model object of class lavaan.                                                                                              |
| <code>node_positions</code> | Specify a list of node position objects. Use the <a href="#">node()</a> helper function to assist with this.                        |
| <code>path_positions</code> | Specify a list of path configuration objects. Use the <a href="#">path()</a> helper function to assist with this.                   |
| <code>standardised</code>   | Logical. If TRUE, uses standardised parameter estimates ( <code>est.std</code> ). Default is FALSE.                                 |
| <code>digits</code>         | Number of digits to display for estimates. Default is 3.                                                                            |
| <code>est_stars</code>      | Logical. Whether to display significance stars on path estimates. Default is FALSE.                                                 |
| <code>est_p</code>          | Logical. Whether to display p-values on path estimates. Default is FALSE.                                                           |
| <code>est_ci</code>         | Logical. Whether to display confidence intervals on path estimates. Default is FALSE.                                               |
| <code>show_variances</code> | Logical. Whether to display variance and residual paths. Default is FALSE.                                                          |
| <code>variance_stars</code> | Logical. Whether to display significance stars on variance paths. Default is FALSE.                                                 |
| <code>variance_p</code>     | Logical. Whether to display p-values on variance paths. Default is FALSE.                                                           |
| <code>variance_ci</code>    | Logical. Whether to display confidence intervals on variance paths. Default is FALSE.                                               |
| <code>sig_linetype</code>   | Logical. If TRUE, renders non-significant paths with dashed lines. Default is FALSE.                                                |
| <code>p_threshold</code>    | Statistical significance threshold used to determine non-significant paths when <code>sig_linetype = TRUE</code> . Default is 0.05. |
| <code>node_width</code>     | Base width for nodes. Default is 1.5.                                                                                               |

|                                        |                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>node_height</code>               | Base height for node shapes. Default is 1.                                                                                                                                                                                                                                                                              |
| <code>latent_node_size_adjust</code>   | Numeric multiplier scaling latent node ellipses. Default is 1.                                                                                                                                                                                                                                                          |
| <code>observed_node_size_adjust</code> | Numeric multiplier scaling observed node rectangles. Default is 1.                                                                                                                                                                                                                                                      |
| <code>latent_node_text_size</code>     | Text font size for latent node labels. Default is 4.                                                                                                                                                                                                                                                                    |
| <code>observed_node_text_size</code>   | Text font size for observed node labels. Default is 4.                                                                                                                                                                                                                                                                  |
| <code>path_transparent_text</code>     | Logical. If TRUE, path estimate labels have a transparent background. If FALSE, they receive a white background mask. Default is FALSE.                                                                                                                                                                                 |
| <code>path_text_size</code>            | Text font size for path estimate labels. Default is 3.5.                                                                                                                                                                                                                                                                |
| <code>line_thickness</code>            | Sets thickness of paths. Default is 0.6.                                                                                                                                                                                                                                                                                |
| <code>arrow_size</code>                | Sets the size of arrow heads. Default is 0.2.                                                                                                                                                                                                                                                                           |
| <code>panel_cols</code>                | Integer for number of columns to use when arranging multi-group panels. Default is NULL.                                                                                                                                                                                                                                |
| <code>show_group_labels</code>         | Logical. For multi-group models, whether to annotate each panel with its panel number and the raw group value it represents (e.g. "Panel 1: Group = male"). Needed to identify which diagram represents each group and its internal panel number when creating panel titles and notes. Default is FALSE.                |
| <code>title_text_size</code>           | Font size for panel titles. Default is 25.                                                                                                                                                                                                                                                                              |
| <code>panel_titles</code>              | Specify a list of titles for panels. Use the <code>panel_title()</code> helper function to assist with this. Any panel not referenced is left untitled. Default is NULL.                                                                                                                                                |
| <code>panel_notes</code>               | Specify a list of notes and their positions for panels. Use the <code>panel_note()</code> helper function to assist with this. Any panel not referenced is left without a note. Notes placed outside the plot limits are clipped, so increase <code>margin_x</code> / <code>margin_y</code> if needed. Default is NULL. |
| <code>note_text_size</code>            | Font size for panel notes. Default is 4.                                                                                                                                                                                                                                                                                |
| <code>note_transparent_text</code>     | Logical. If TRUE, panel notes have a transparent background. If FALSE, they receive a white background mask. Default is FALSE.                                                                                                                                                                                          |
| <code>note_outline</code>              | Logical. If TRUE, panel notes receive a black outline. Default is TRUE.                                                                                                                                                                                                                                                 |
| <code>show_grid</code>                 | Logical. Whether to overlay a coordinate grid. Default is FALSE.                                                                                                                                                                                                                                                        |
| <code>grid_axis_scale</code>           | Sets the spacing of grid-lines when <code>show_grid = TRUE</code> . Default is 1.                                                                                                                                                                                                                                       |
| <code>margin_x</code>                  | Padding for plot limits along the x-axis. Default is 0.5.                                                                                                                                                                                                                                                               |
| <code>margin_y</code>                  | Padding for plot limits along the y-axis. Default is 0.5.                                                                                                                                                                                                                                                               |
| <code>look_up_table</code>             | Logical. If TRUE, also returns a look-up table detailing the width (x scale) and height (y scale) of latent and observed nodes, along with the text sizes used for latent, observed, and path labels. Default is FALSE.                                                                                                 |

## Details

Using the function requires 4 steps and is illustrated by the example below. To save time and avoid syntax/logical errors, `template()` prints starting code for `node_positions`, `path_positions`, `panel_titles` and the `diyPaths()` call based on your fitted model.

See `vignette("diy.sem.plot")` for in-depth examples and guidance on using the function's arguments.

1. Specify and fit the SEM using **lavaan**.
2. Specify `node_positions` as a list (`node_positions = list(...)`) and, within it, define each node's position and label using the `node()` helper function.
3. Specify `path_positions` as a list (`path_positions = list(...)`) and, within it, define each path's connection points, curvature if covariance/correlations, and fine-tune estimate positions adjustments using the `path()` helper function.
4. Call `diyPaths()`, passing in the fitted model, `node_positions`, and `path_positions`, along with any additional display augmentations (i.e. show significance stars on estimates).

Creating a diagram for a specific model is inherently an iterative process. As such, steps 2-4 will likely need to be repeated with additional fine-tuning adjustments to achieve the desired result.

## Value

A `ggplot` (or `patchwork`, for multi-group models) object representing the SEM path diagram(s). If `look_up_table = TRUE`, a list containing the diagram(s) (`$plot`) and a look-up data.frame of node width/height and text sizes for latent, observed, and path labels (`$look_up_table`).

## See Also

`template()` to generate starting code for `node_positions`, `path_positions`, `panel_titles` and `panel_notes`; `node()`, `path()`, `panel_title()`, and `panel_note()` for the helper functions used within them.

## Examples

```
#full SEM example, this model may not make theoretical sense.
```

```
library(lavaan)
library(ggplot2)
```

```
data(HolzingerSwineford1939, package = "lavaan")
```

```
#specify the model
```

```
sem_model <- '
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9

  speed ~ visual + textual
  visual ~~ textual
```

```

,

#fit the model

fit <- sem(sem_model, data = HolzingerSwineford1939)

#generate a starting template

temp <- template(fit = fit, diyPaths_arguments = "max")

#### edited template ####

node_list <- list(
  node(name = "visual", x = 1, y = 1, label = "Visual"),
  node(name = "textual", x = 1, y = 2, label = "Textual"),
  node(name = "speed", x = 4, y = 1.5, label = "Speed"),
  node(name = "x1", x = -0.06, y = -0.5, label = "Visual\nPerception"),
  node(name = "x2", x = 1, y = -0.5, label = "Cubes"),
  node(name = "x3", x = 2.06, y = -0.5, label = "Lozenges"),
  node(name = "x4", x = -0.06, y = 3.5, label = "Paragraph\nComprehension"),
  node(name = "x5", x = 1, y = 3.5, label = "Sentence\nCompletion"),
  node(name = "x6", x = 2.06, y = 3.5, label = "Word\nMeaning"),
  node(name = "x7", x = 6, y = 0.5, label = "Speeded\nAddition"),
  node(name = "x8", x = 6, y = 1.5, label = "Speeded\nCounting"),
  node(name = "x9", x = 6, y = 2.5, label = "Speeded\nDiscrimination")
)

path_list <- list(
  # regression paths
  path(from = "visual", to = "speed", side_from = "right", side_to = "left"),
  path(from = "textual", to = "speed", side_from = "right", side_to = "left"),

  # loading paths
  path(from = "visual", to = "x1", side_from = "bottom", side_to = "top", nudge_text_x = -0.1),
  path(from = "visual", to = "x2", side_from = "bottom", side_to = "top"),
  path(from = "visual", to = "x3", side_from = "bottom", side_to = "top", nudge_text_x = 0.1),
  path(from = "textual", to = "x4", side_from = "top", side_to = "bottom", nudge_text_x = -0.1),
  path(from = "textual", to = "x5", side_from = "top", side_to = "bottom"),
  path(from = "textual", to = "x6", side_from = "top", side_to = "bottom", nudge_text_x = 0.1),
  path(from = "speed", to = "x7", side_from = "right", side_to = "left"),
  path(from = "speed", to = "x8", side_from = "right", side_to = "left"),
  path(from = "speed", to = "x9", side_from = "right", side_to = "left"),

  # covariance/correlation paths
  path(from = "visual", to = "textual", side_from = "left", side_to = "left", cov_curve = -0.6)
)

title_list <- list(
  panel_title(panel_num = 1, title = "My SEM Plot")
)

p <- diyPaths(fit = fit, node_positions = node_list, path_positions = path_list,
  standardised = TRUE,

```

```

        digits = 3,
        est_stars = TRUE,
        est_p = FALSE,
        est_ci = FALSE,
        show_variances = FALSE,
        variance_stars = FALSE,
        variance_p = FALSE,
        variance_ci = FALSE,
        sig_linetype = FALSE,
        p_threshold = 0.05,
        panel_titles = title_list,
        show_grid = TRUE,
        grid_axis_scale = 0.5,
        path_transparent_text = FALSE,
        latent_node_text_size = 4,
        observed_node_text_size = 3,
        path_text_size = 3.5,
        line_thickness = 0.6,
        arrow_size = 0.2,
        node_width = 1.5, node_height = 1,
        latent_node_size_adjust = 0.8,
        observed_node_size_adjust = 0.55,
        show_group_labels = FALSE,
        panel_cols = NULL,
        margin_x = 0.5,
        margin_y = 0.5,
        look_up_table = TRUE)

print(p)

```

---

|      |                                    |
|------|------------------------------------|
| node | <i>Create a node position list</i> |
|------|------------------------------------|

---

## Description

Helper function that creates a list of arguments which specify a given node's position and its name in the diagram, designed for use within the `node_positions` argument of `diyPaths()`.

## Usage

```
node(name, x = 0, y = 0, label = NULL)
```

## Arguments

|                   |                                                                        |
|-------------------|------------------------------------------------------------------------|
| <code>name</code> | The name of the variable within the lavaan model.                      |
| <code>x</code>    | Numeric value for the x-coordinate of the node's centre. Default is 0. |
| <code>y</code>    | Numeric value for the y-coordinate of the node's centre. Default is 0. |

**label** Custom label to display instead of lavaan variable name. Defaults to the lavaan variable name.  
 #' @seealso [path\(\)](#) to specify the paths between nodes; [diyPaths\(\)](#) to render the diagram; [template\(\)](#) to generate a starting node\_positions list.

### Value

A list containing arguments that specify a node's position and label, for use within the node\_positions argument of [diyPaths\(\)](#).

### Examples

```
#a list that specifies a node positioned on x = 1, y = 2,
#and relabelled from its `lavaan` model name.

node(name = "bpm", x = 1, y = 2, label = "Beats per Minute")
```

---

|            |                                           |
|------------|-------------------------------------------|
| panel_note | <i>Create a note for a diyPaths panel</i> |
|------------|-------------------------------------------|

---

### Description

Helper function that creates a list of arguments which specify a custom note and its target panel, designed for use within the panel\_notes argument of [diyPaths\(\)](#). Use the show\_group\_labels argument in [diyPaths\(\)](#) to view each panel's number and which group it refers to.

### Usage

```
panel_note(panel_num, note_text = NULL, x = 0, y = 0)
```

### Arguments

|           |                                                               |
|-----------|---------------------------------------------------------------|
| panel_num | Integer value for the panel number this note applies to.      |
| note_text | Text to display in the note. If NULL, no note is drawn.       |
| x         | Numeric value for the x-coordinate of the note. Default is 0. |
| y         | Numeric value for the y-coordinate of the note. Default is 0. |

### Value

A list containing arguments that specify a panel's note text and position, for use within the panel\_notes argument of [diyPaths\(\)](#).

### See Also

[diyPaths\(\)](#), where the list of notes is passed to panel\_notes; [panel\\_title\(\)](#) for adding titles to panels; [template\(\)](#), whose output includes a starting note list.

**Examples**

```
# a note on the first panel, placed at x = 2.5, y = 0
panel_note(panel_num = 1, note_text = "n = 100", x = 2.5, y = 0)

# a note on the second panel of a multi-group model
panel_note(panel_num = 2, note_text = "Source: my data", x = 2.5, y = 0)
```

---

|             |                                     |
|-------------|-------------------------------------|
| panel_title | Create a title for a diyPaths panel |
|-------------|-------------------------------------|

---

**Description**

Helper function that creates a list of arguments which specify custom titles and its target panel, designed for use within the panel\_titles argument of [diyPaths\(\)](#). Use the show\_group\_labels argument in [diyPaths\(\)](#) to view each panel's number and which group it refers to.

**Usage**

```
panel_title(panel_num = 1, title = NULL)
```

**Arguments**

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| panel_num | Integer value for the panel number this title applies to. Default is 1. |
| title     | The title text to display.                                              |

**Value**

A list containing arguments that specify a panel's number and title, for use within the panel\_titles argument of [diyPaths\(\)](#).

**See Also**

[diyPaths\(\)](#), where the list of titles is passed to panel\_titles; [panel\\_note\(\)](#) for adding notes to panels; [template\(\)](#), whose output includes a starting title list.

**Examples**

```
# titling a single, non-grouped model
panel_title(title = "My SEM Model")

# titling the second panel of a multi-group model
panel_title(panel_num = 2, title = "Female Participants")
```

---

|      |                                    |
|------|------------------------------------|
| path | <i>Create a path position list</i> |
|------|------------------------------------|

---

## Description

Helper function that creates a list of arguments which specify a given path's position and fine-tuning adjustments, designed for use within the `path_positions` argument of `diyPaths()`.

## Usage

```
path(
  from,
  to,
  side_from = "right",
  side_to = "left",
  cov_curve = NULL,
  nudge_text_x = 0,
  nudge_text_y = 0,
  variance_position = "top"
)
```

## Arguments

|                                |                                                                                               |
|--------------------------------|-----------------------------------------------------------------------------------------------|
| <code>from</code>              | The source variable name within the lavaan model.                                             |
| <code>to</code>                | The target variable name within the lavaan model.                                             |
| <code>side_from</code>         | Side of source node where path starts ("top", "bottom", "left", "right"). Default is "right". |
| <code>side_to</code>           | Side of target node where path ends ("top", "bottom", "left", "right"). Default is "left".    |
| <code>cov_curve</code>         | Numeric value for curvature of covariance/correlation paths. Default is NULL.                 |
| <code>nudge_text_x</code>      | Numeric fine tuning adjustment for path estimate text along the x-axis. Default is 0.         |
| <code>nudge_text_y</code>      | Numeric fine tuning adjustment for path estimate text along the y-axis. Default is 0.         |
| <code>variance_position</code> | Placement of variance/residual paths ("top", "bottom", "left", or "right"). Default is "top". |

## Details

`from/to` in `path()` must exactly match the variable name used in the lavaan model syntax. Furthermore, the order must also be correct for regression or loading paths. A misspelled or mismatched `path()` entry will not raise an error, instead, it will use default attachment points and values, not applying specific customisations. This can be completely avoided using `template()`.

cov\_curve's value can be used to adjust direction of curve on covariance/correlation path. For a mostly vertical path (i.e. node1:  $x = 0, y = 1$  -> node2:  $x = 0, y = 2$ ), positive curvature bends it left and negative curvature bends it right. For a mostly horizontal path (i.e. node1:  $x = 1, y = 0$  -> node2:  $x = 2, y = 0$ ), positive curvature bends it down and negative curvature bends it up. Best results typically range from -1 to 1.

### Value

A list containing arguments that specify a path's position and fine-tuning adjustments, for use within the path\_positions argument of `diyPaths()`.

### See Also

`node()` to specify node positions; `diyPaths()` to render the diagram; `template()` to generate a starting path\_positions list.

### Examples

```
#a list that specifies a regression path of node "anxiety" predicting node "depression".
path(from = "anx", to = "dep", side_from = "right", side_to = "left")

#a list that specifies a covariance/correlation path of node "anxiety" and node "depression".
#Order of "from" and "to" does not matter for covariance/correlation.

path(from = "dep", to = "anx", side_from = "left", side_to = "left", cov_curve = -0.6)
```

---

template

---

*Create a template for use in diyPaths*


---

### Description

Takes your fitted model and prints a ready-to-run starting template (node list, path list, title list, and `diyPaths()` call) to the console. Every variable and path from your model is already filled in with the correct names, so the code will render a diagram as soon as you run it. All arguments will be at their default values, so your job is to fine-tune: copy the output into your script, then edit the lines marked #EDIT ME!.

The template is also returned invisibly as a character string, so it can be stored as an object then viewed with `writelnLines()` (see examples).

### Usage

```
template(
  fit,
  include_variance_paths = TRUE,
  diyPaths_arguments = c("max", "min")
)
```

**Arguments**

`fit` the lavaan object of fitted model

`include_variance_paths` Logical. Whether to include specification for variance/residual paths in `paths_lists`

`diyPaths_arguments` Use either "max" (the printed call shows every `diyPaths()` argument) or "min" (only the required arguments). Default is "max"

**Value**

Invisibly returns the template as a single character string, and prints it to the console.

**See Also**

`diyPaths()` to render the diagram from the template; `node()`, `path()`, `panel_title()`, and `panel_note()` for the helper functions the template uses.

**Examples**

```
library(lavaan)
library(ggplot2)

data(HolzingerSwineford1939, package = "lavaan")

#specify the model

sem_model <- '
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9

  speed ~ visual + textual
  visual ~~ textual
'

#fit the model

fit <- sem(sem_model, data = HolzingerSwineford1939)

#print the template to the console

# template(fit = fit, include_variance_paths = TRUE, diyPaths_arguments = "max")

#store the template, then view it properly formatted

tmpl <- template(fit = fit, diyPaths_arguments = "min")
writeLines(tmpl)
```

# Index

diyPaths, [2](#)

diyPaths(), [7–12](#)

node, [7](#)

node(), [3, 5, 11, 12](#)

panel\_note, [8](#)

panel\_note(), [4, 5, 9, 12](#)

panel\_title, [9](#)

panel\_title(), [4, 5, 8, 12](#)

path, [10](#)

path(), [3, 5, 8, 12](#)

template, [11](#)

template(), [5, 8–11](#)