

# Package ‘semEffect’

May 9, 2026

**Title** Structural Equation Model Effect Analysis and Visualization

**Version** 1.2.3

**Description** Provides standardized effect decomposition (direct, indirect, and total effects) for three major structural equation modeling frameworks: 'lavaan', 'piecewiseSEM', and 'plspm'. Automatically handles zero-effect variables, generates publication-ready 'ggplot2' visualizations, and returns both wide-format and long-format effect tables. Supports effect filtering, multi-model object inputs, and customizable visualization parameters. For a general overview of the methods used in this package, see Rosseel (2012) <[doi:10.18637/jss.v048.i02](https://doi.org/10.18637/jss.v048.i02)> and Lefcheck (2016) <[doi:10.1111/2041-210X.12512](https://doi.org/10.1111/2041-210X.12512)>.

**URL** <https://github.com/PhDMeiwp/semEffect/>

**BugReports** <https://github.com/PhDMeiwp/semEffect/issues>

**Depends** R (>= 4.4.0)

**License** GPL-3

**Encoding** UTF-8

**Imports** lavaan, piecewiseSEM, plspm, ggplot2, tidyr, dplyr, utils, checkmate, RColorBrewer

**Suggests** testthat

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Weiping Mei [aut, cre] (ORCID: <<https://orcid.org/0000-0001-6400-9862>>)

**Maintainer** Weiping Mei <[meiweiping@163.com](mailto:meiweiping@163.com)>

**Repository** CRAN

**Date/Publication** 2025-07-04 19:30:01 UTC

## Contents

sem_effects . . . . .	2
<b>Index</b>	<b>5</b>

**Description**

Provides standardized effect decomposition (direct, indirect, and total effects) for three major structural equation modeling frameworks: 'lavaan', 'piecewiseSEM', and 'plspm'. Automatically handles zero-effect variables, generates publication-ready 'ggplot2' visualizations, and returns both wide-format and long-format effect tables. Supports effect filtering, multi-model object inputs, and customizable visualization parameters.

**Usage**

```
sem_effects(
  object,
  target,
  plot = TRUE,
  delete_zero_effect = TRUE,
  total_only = FALSE,
  total_color = "skyblue",
  color_palette = c("darkgreen", "skyblue", "orange")
)
```

**Arguments**

object	SEM object (lavaan/psem/plspm).
target	Character string specifying the target variable name for effect analysis.
plot	Logical indicating whether to generate effect visualization plots (default: TRUE).
delete_zero_effect	Logical indicating whether to removes rows where all specified effect columns contain only zeros (default: TRUE).
total_only	Logical controlling plot mode. If TRUE, shows only total effects with customizable colors; if FALSE, displays all effect types with palette coloring (default: FALSE).
total_color	Single color or vector of colors for total effect bars when total_only=TRUE (default: "skyblue").
color_palette	Character vector of 3 colors for direct/indirect/total effects when total_only=FALSE (default: c("darkgreen", "skyblue", "orange")).

**Value**

A list containing three components:

- effect\_table: A data frame with variables and their standardized effect values (direct, indirect, total)
- effect\_long: A long-format version of effect\_table
- plot\_object: A ggplot2 object (if plot=TRUE), NULL otherwise

**Author(s)**

Weiping Mei

**See Also**[sem](#), [psem](#), [plspm](#)**Examples**

```
# Example 01: lavaan -----  
  
library(lavaan)  
  
model <- '  
  # Measurement model  
  ind60 =~ x1 + x2 + x3  
  dem60 =~ y1 + y2 + y3 + y4  
  dem65 =~ y5 + y6 + y7 + y8  
  
  # Structural model  
  dem60 ~ ind60  
  dem65 ~ ind60 + dem60  
'  
  
fit <- sem(model, data = PoliticalDemocracy)  
  
# Analyze effects for target variable "dem65"  
results <- sem_effects(fit, target = "dem65")  
  
print(results$effect_table)  
print(results$effect_long)  
print(results$plot_object)  
  
# Customize plot appearance  
results$plot_object +  
  ggplot2::coord_flip()+  
  ggplot2::theme_minimal() +  
  ggplot2::ggtitle("Standardized effects for dem65")  
  
# Example 02: piecewiseSEM -----  
  
library(piecewiseSEM)  
pmod <- psem(  
  lm(rich ~ cover, data = keeley),  
  lm(cover ~ firesev, data = keeley),  
  lm(firesev ~ age, data = keeley),  
  data = keeley  
)  
  
sem_effects(pmod, target = "rich",  
  color_palette = c("darkgreen", "grey80", "purple"))
```

```
# Example 03: plspm -----  
  
library(plspm)  
data(satisfaction)  
  
# path matrix  
IMAG = c(0,0,0,0,0,0)  
EXPE = c(1,0,0,0,0,0)  
QUAL = c(0,1,0,0,0,0)  
VAL = c(0,1,1,0,0,0)  
SAT = c(1,1,1,1,0,0)  
LOY = c(1,0,0,0,1,0)  
sat_path = rbind(IMAG, EXPE, QUAL, VAL, SAT, LOY)  
  
# blocks of outer model  
sat_blocks = list(1:5, 6:10, 11:15, 16:19, 20:23, 24:27)  
  
# vector of modes (reflective indicators)  
sat_mod = rep("A", 6)  
  
# apply plspm  
plsmmodel = plspm(satisfaction, sat_path, sat_blocks, modes = sat_mod)  
  
sem_effects(plsmmodel, target = "LOY", plot = TRUE, delete_zero_effect = TRUE,  
            total_only = TRUE,  
            total_color = RColorBrewer::brewer.pal(5,"Set3"))
```

# Index

plspm, [3](#)

psem, [3](#)

sem, [3](#)

sem\_effects, [2](#)