# semmcci: Monte Carlo Confidence Intervals

Ivan Jacob Agaloos Pesigan

## Installation

You can install the CRAN release of `semmcci` with:

```r
install.packages("semmcci")
```

You can install the development version of `semmcci` from GitHub with:

```r
install.packages("remotes")
remotes::install_github("jeksterslab/semmcci")
```

## Documentation

See GitHub Pages for package documentation.

## Description

In the Monte Carlo method, a sampling distribution of parameter estimates is generated from the multivariate normal distribution using the parameter estimates and the sampling variance-covariance matrix. Confidence intervals for defined parameters are generated by obtaining percentiles corresponding to $100(1 - \alpha)\%$ from the generated sampling distribution, where $\alpha$ is the significance level.

Monte Carlo confidence intervals for free and defined parameters in models fitted in the structural equation modeling package `lavaan` can be generated using the `semmcci` package. The package has two main functions, namely, `MC()` and `MCStd()`. The output of `lavaan` is passed as the first argument to the `MC()` function to generate Monte Carlo confidence intervals. Monte Carlo confidence intervals for the standardized estimates can also be generated by passing the output of the `MC()` function to the `MCStd()` function.

# Example

A common application of the Monte Carlo method is to generate confidence intervals for the indirect effect. In the simple mediation model, variable `X` has an effect on variable `Y`, through a mediating variable `M`. This mediating or indirect effect is a product of path coefficients from the fitted model.

```
library(semmcci)
library(lavaan)
```

## Data

```
n <- 1000
X <- rnorm(n = n)
M <- 0.50 * X + rnorm(n = n)
Y <- 0.25 * X + 0.50 * M + rnorm(n = n)
data <- data.frame(X, M, Y)
```

## Model Specification

The indirect effect is defined by the product of the slopes of paths `X` to `M` labeled as `a` and `M` to `Y` labeled as `b`. In this example, we are interested in the confidence intervals of `indirect` defined as the product of `a` and `b` using the `:=` operator in the `lavaan` model syntax.

```r
model <- "
  Y ~ cp * X + b * M
  M ~ a * X
  indirect := a * b
  direct := cp
  total := cp + (a * b)
"
```

## Model Fitting

We can now fit the model using the `sem()` function from `lavaan`.

```r
fit <- sem(data = data, model = model)
```

## Monte Carlo Confidence Intervals

The `fit` `lavaan` object can then be passed to the `MC()` function to generate Monte Carlo confidence intervals.

```r
MC(fit, R = 20000L, alpha = c(0.001, 0.01, 0.05))

#> Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2709 0.0362 20000 0.1510 0.1798 0.1995 0.3416 0.3638 0.3877
#> b        0.5181 0.0326 20000 0.4201 0.4373 0.4545 0.5829 0.6046 0.6295
#> a        0.4795 0.0314 20000 0.3791 0.3994 0.4182 0.5415 0.5603 0.5824
#> Y~~Y     1.0302 0.0460 20000 0.8808 0.9127 0.9408 1.1201 1.1480 1.1797
#> M~~M     0.9675 0.0435 20000 0.8280 0.8560 0.8814 1.0520 1.0806 1.1101
#> indirect 0.2484 0.0226 20000 0.1821 0.1954 0.2060 0.2948 0.3097 0.3275
#> direct   0.2709 0.0362 20000 0.1510 0.1798 0.1995 0.3416 0.3638 0.3877
```

```
#> total    0.5193 0.0363 20000 0.3994 0.4255 0.4483 0.5913 0.6130 0.6405
```

### Standardized Monte Carlo Confidence Intervals

Standardized Monte Carlo Confidence intervals can be generated by passing the result of the `MC()` function to `MCStd()`.

**Note:** We recommend setting `fixed.x = FALSE` when generating standardized estimates and confidence intervals to model the variances and covariances of the predictors if they are assumed to be random.

```
fit <- sem(data = data, model = model, fixed.x = FALSE)
unstd <- MC(fit, R = 20000L, alpha = c(0.001, 0.01, 0.05))
vcov(unstd)

#>                      cp             b             a          Y~~Y          M~~M
#> cp         1.297368e-03 -5.018730e-04  6.938116e-06  8.830691e-06 -2.506666e-06
#> b         -5.018730e-04  1.069537e-03  1.497130e-06 -6.873827e-06 -5.619730e-06
#> a          6.938116e-06  1.497130e-06  9.958097e-04  1.155519e-05  3.457476e-06
#> Y~~Y       8.830691e-06 -6.873827e-06  1.155519e-05  2.141787e-03 -2.078515e-05
#> M~~M      -2.506666e-06 -5.619730e-06  3.457476e-06 -2.078515e-05  1.898246e-03
#> X~~X       7.223071e-06 -3.862143e-06 -6.289033e-06 -9.416870e-08 -1.950962e-05
#> indirect -2.366201e-04  5.138682e-04  5.163758e-04  3.282381e-06 -7.121490e-07
#> direct     1.297368e-03 -5.018730e-04  6.938116e-06  8.830691e-06 -2.506666e-06
#> total      1.060748e-03  1.199518e-05  5.233140e-04  1.211307e-05 -3.218815e-06
#>                    X~~X      indirect        direct         total
#> cp         7.223071e-06 -2.366201e-04  1.297368e-03  1.060748e-03
#> b         -3.862143e-06  5.138682e-04 -5.018730e-04  1.199518e-05
#> a         -6.289033e-06  5.163758e-04  6.938116e-06  5.233140e-04
#> Y~~Y      -9.416870e-08  3.282381e-06  8.830691e-06  1.211307e-05
```

```
#> M~~M     -1.950962e-05 -7.121490e-07 -2.506666e-06 -3.218815e-06
#> X~~X      1.923827e-03 -4.578435e-06  7.223071e-06  2.644636e-06
#> indirect -4.578435e-06  5.149858e-04 -2.366201e-04  2.783657e-04
#> direct    7.223071e-06 -2.366201e-04  1.297368e-03  1.060748e-03
#> total     2.644636e-06  2.783657e-04  1.060748e-03  1.339114e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2150 0.0282 20000 0.1184 0.1409 0.1595 0.2697 0.2859 0.3094
#> b        0.4539 0.0263 20000 0.3638 0.3844 0.4018 0.5045 0.5205 0.5409
#> a        0.4344 0.0257 20000 0.3509 0.3664 0.3829 0.4838 0.4989 0.5154
#> Y~~Y     0.6630 0.0245 20000 0.5835 0.5988 0.6143 0.7097 0.7240 0.7375
#> M~~M     0.8113 0.0223 20000 0.7343 0.7511 0.7659 0.8534 0.8657 0.8769
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1972 0.0166 20000 0.1451 0.1560 0.1651 0.2305 0.2405 0.2509
#> direct   0.2150 0.0282 20000 0.1184 0.1409 0.1595 0.2697 0.2859 0.3094
#> total    0.4122 0.0264 20000 0.3236 0.3433 0.3593 0.4625 0.4780 0.4930
```

# References


MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research*, *39*(1), 99–128. https://doi.org/10.1207/s15327906mbr3901_4

Preacher, K. J., & Selig, J. P. (2012). Advantages of Monte Carlo confidence intervals for indirect effects. *Communication Methods and Measures*, *6*(2), 77–98. https://doi.org/10.1080/19312458.2012.679848

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Tofighi, D., & Kelley, K. (2019). Indirect effects in sequential mediation models: Evaluating methods for hypothesis testing and confidence interval formation. *Multivariate Behavioral Research*, *55*(2), 188–210. https://doi.org/10.1080/00273171.2019.1618545

Tofighi, D., & MacKinnon, D. P. (2015). Monte Carlo confidence intervals for complex functions of indirect effects. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(2), 194–205. https://doi.org/10.1080/10705511.2015.1057284