# semmcci: Monte Carlo Confidence Intervals

Ivan Jacob Agaloos Pesigan

## Installation

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

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

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

```
if (!require("remotes")) 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.

```
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`.

```
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.

```
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       0.3574 0.0359 20000 0.2385 0.2644 0.2871 0.4275 0.4490 0.4723
#> b        0.4854 0.0322 20000 0.3835 0.4024 0.4223 0.5482 0.5685 0.5909
#> a        0.4734 0.0322 20000 0.3705 0.3901 0.4106 0.5363 0.5581 0.5852
#> Y~~Y     1.0568 0.0472 20000 0.9003 0.9343 0.9638 1.1484 1.1806 1.2108
#> M~~M     1.0245 0.0457 20000 0.8671 0.9040 0.9352 1.1134 1.1438 1.1739
#> indirect 0.2298 0.0219 20000 0.1665 0.1766 0.1889 0.2741 0.2886 0.3078
#> direct   0.3574 0.0359 20000 0.2385 0.2644 0.2871 0.4275 0.4490 0.4723
```

```
#> total    0.5872 0.0360 20000 0.4648 0.4961 0.5175 0.6589 0.6817 0.7053
```

## 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        1.271642e-03 -4.886731e-04 -2.853803e-06 -6.146224e-06  1.868811e-05
#> b        -4.886731e-04  1.027542e-03  3.342036e-06 -5.181044e-06 -1.215627e-05
#> a        -2.853803e-06  3.342036e-06  1.014341e-03 -1.634470e-05  6.689348e-06
#> Y~~Y     -6.146224e-06 -5.181044e-06 -1.634470e-05  2.265181e-03 -2.854350e-06
#> M~~M      1.868811e-05 -1.215627e-05  6.689348e-06 -2.854350e-06  2.082305e-03
#> X~~X     -7.510356e-06 -3.528436e-08  4.516657e-06  1.648327e-05 -1.058333e-05
#> indirect -2.332444e-04  4.880537e-04  4.940707e-04 -1.070012e-05 -2.858860e-06
#> direct    1.271642e-03 -4.886731e-04 -2.853803e-06 -6.146224e-06  1.868811e-05
#> total     1.038398e-03 -6.194230e-07  4.912169e-04 -1.684634e-05  1.582925e-05
#>                    X~~X      indirect        direct         total
#> cp       -7.510356e-06 -2.332444e-04  1.271642e-03  1.038398e-03
#> b        -3.528436e-08  4.880537e-04 -4.886731e-04 -6.194230e-07
#> a         4.516657e-06  4.940707e-04 -2.853803e-06  4.912169e-04
#> Y~~Y      1.648327e-05 -1.070012e-05 -6.146224e-06 -1.684634e-05
```

```
#> M~~M     -1.058333e-05 -2.858860e-06  1.868811e-05  1.582925e-05
#> X~~X      1.988357e-03  2.033402e-06 -7.510356e-06 -5.476954e-06
#> indirect  2.033402e-06  4.720010e-04 -2.332444e-04  2.387567e-04
#> direct   -7.510356e-06 -2.332444e-04  1.271642e-03  1.038398e-03
#> total    -5.476954e-06  2.387567e-04  1.038398e-03  1.277154e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2788 0.0271 20000 0.1905 0.2085 0.2256 0.3318 0.3475 0.3660
#> b         0.4232 0.0260 20000 0.3386 0.3556 0.3718 0.4740 0.4890 0.5045
#> a         0.4236 0.0258 20000 0.3358 0.3556 0.3720 0.4733 0.4880 0.5040
#> Y~~Y      0.6432 0.0242 20000 0.5641 0.5797 0.5938 0.6899 0.7041 0.7197
#> M~~M      0.8206 0.0218 20000 0.7460 0.7618 0.7760 0.8616 0.8736 0.8872
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1793 0.0157 20000 0.1311 0.1399 0.1486 0.2106 0.2221 0.2353
#> direct    0.2788 0.0271 20000 0.1905 0.2085 0.2256 0.3318 0.3475 0.3660
#> total     0.4580 0.0248 20000 0.3709 0.3918 0.4083 0.5060 0.5202 0.5348
```

## 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. (2023). *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