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 α 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       0.2882 0.0374 20000 0.1665 0.1910 0.2155 0.3604 0.3826 0.4095
#> b        0.4916 0.0325 20000 0.3831 0.4082 0.4281 0.5551 0.5746 0.5963
#> a        0.5344 0.0321 20000 0.4309 0.4524 0.4714 0.5970 0.6157 0.6342
#> Y~~Y     1.0424 0.0468 20000 0.8832 0.9246 0.9508 1.1345 1.1630 1.2044
#> M~~M     0.9861 0.0444 20000 0.8335 0.8732 0.8991 1.0725 1.0992 1.1337
#> indirect 0.2627 0.0234 20000 0.1936 0.2061 0.2183 0.3097 0.3265 0.3434
#> direct   0.2882 0.0374 20000 0.1665 0.1910 0.2155 0.3604 0.3826 0.4095
```

```
#> total    0.5509 0.0365 20000 0.4304 0.4570 0.4795 0.6218 0.6444 0.6684
```

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        1.373829e-03 -5.615288e-04 -1.538478e-05 -1.394073e-05 -1.282449e-05
#> b        -5.615288e-04  1.049400e-03  1.402738e-05  6.675277e-06  1.128410e-05
#> a        -1.538478e-05  1.402738e-05  1.055813e-03  3.858488e-06 -6.455562e-06
#> Y~~Y     -1.394073e-05  6.675277e-06  3.858488e-06  2.175427e-03  3.733108e-05
#> M~~M     -1.282449e-05  1.128410e-05 -6.455562e-06  3.733108e-05  1.951460e-03
#> X~~X     -1.016549e-05  7.298089e-07  1.968024e-06 -4.493867e-06 -6.048033e-06
#> indirect -3.078607e-04  5.680053e-04  5.263409e-04  5.544358e-06  2.856267e-06
#> direct    1.373829e-03 -5.615288e-04 -1.538478e-05 -1.394073e-05 -1.282449e-05
#> total     1.065968e-03  6.476549e-06  5.109562e-04 -8.396372e-06 -9.968221e-06
#>                    X~~X      indirect        direct         total
#> cp       -1.016549e-05 -3.078607e-04  1.373829e-03  1.065968e-03
#> b         7.298089e-07  5.680053e-04 -5.615288e-04  6.476549e-06
#> a         1.968024e-06  5.263409e-04 -1.538478e-05  5.109562e-04
#> Y~~Y     -4.493867e-06  5.544358e-06 -1.394073e-05 -8.396372e-06
```

```
#> M~~M     -6.048033e-06  2.856267e-06 -1.282449e-05 -9.968221e-06
#> X~~X      1.864880e-03  1.510224e-06 -1.016549e-05 -8.655261e-06
#> indirect  1.510224e-06  5.634613e-04 -3.078607e-04  2.556006e-04
#> direct   -1.016549e-05 -3.078607e-04  1.373829e-03  1.065968e-03
#> total    -8.655261e-06  2.556006e-04  1.065968e-03  1.321569e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2254 0.0286 20000 0.1305 0.1514 0.1691 0.2809 0.2973 0.3190
#> b         0.4402 0.0269 20000 0.3511 0.3697 0.3861 0.4918 0.5070 0.5269
#> a         0.4667 0.0251 20000 0.3811 0.4016 0.4174 0.5149 0.5289 0.5444
#> Y~~Y      0.6628 0.0243 20000 0.5834 0.6003 0.6142 0.7091 0.7230 0.7394
#> M~~M      0.7822 0.0234 20000 0.7036 0.7202 0.7349 0.8258 0.8387 0.8547
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2054 0.0172 20000 0.1524 0.1627 0.1725 0.2395 0.2505 0.2653
#> direct    0.2254 0.0286 20000 0.1305 0.1514 0.1691 0.2809 0.2973 0.3190
#> total     0.4308 0.0257 20000 0.3428 0.3627 0.3798 0.4797 0.4946 0.5115
```

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