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 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.

```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       0.2800 0.0359 20000 0.1620 0.1884 0.2098 0.3499 0.3730 0.4000
#> b        0.4936 0.0310 20000 0.3915 0.4149 0.4336 0.5538 0.5719 0.5919
#> a        0.5093 0.0328 20000 0.3999 0.4249 0.4448 0.5742 0.5937 0.6162
#> Y~~Y     0.9869 0.0441 20000 0.8372 0.8734 0.9004 1.0726 1.1011 1.1350
#> M~~M     1.0204 0.0454 20000 0.8725 0.9034 0.9318 1.1087 1.1348 1.1659
#> indirect 0.2514 0.0227 20000 0.1827 0.1954 0.2080 0.2974 0.3123 0.3278
#> direct   0.2800 0.0359 20000 0.1620 0.1884 0.2098 0.3499 0.3730 0.4000
```

```
#> total    0.5314 0.0363 20000 0.4116 0.4413 0.4614 0.6026 0.6251 0.6518
```

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        1.292234e-03 -4.984367e-04 -7.627523e-06  2.679134e-06 -9.419414e-06
#> b        -4.984367e-04  9.698715e-04 -5.464230e-07 -1.479983e-05 -1.015580e-06
#> a        -7.627523e-06 -5.464230e-07  1.092354e-03 -2.725259e-08  7.641412e-06
#> Y~~Y      2.679134e-06 -1.479983e-05 -2.725259e-08  1.936443e-03 -4.025619e-07
#> M~~M     -9.419414e-06 -1.015580e-06  7.641412e-06 -4.025619e-07  2.061363e-03
#> X~~X      1.423225e-05 -1.486194e-05 -1.266336e-05 -6.199222e-06  3.116703e-06
#> indirect -2.574495e-04  4.935073e-04  5.386454e-04 -7.585244e-06  3.115210e-06
#> direct    1.292234e-03 -4.984367e-04 -7.627523e-06  2.679134e-06 -9.419414e-06
#> total     1.034785e-03 -4.929368e-06  5.310179e-04 -4.906110e-06 -6.304203e-06
#>                   X~~X      indirect        direct         total
#> cp        1.423225e-05 -2.574495e-04  1.292234e-03  1.034785e-03
#> b        -1.486194e-05  4.935073e-04 -4.984367e-04 -4.929368e-06
#> a        -1.266336e-05  5.386454e-04 -7.627523e-06  5.310179e-04
#> Y~~Y     -6.199222e-06 -7.585244e-06  2.679134e-06 -4.906110e-06
```

```
#> M~~M      3.116703e-06  3.115210e-06 -9.419414e-06 -6.304203e-06
#> X~~X      1.820976e-03 -1.373853e-05  1.423225e-05  4.937237e-07
#> indirect -1.373853e-05  5.180361e-04 -2.574495e-04  2.605866e-04
#> direct    1.423225e-05 -2.574495e-04  1.292234e-03  1.034785e-03
#> total     4.937237e-07  2.605866e-04  1.034785e-03  1.295371e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2226 0.0282 20000 0.1334 0.1502 0.1665 0.2776 0.2949 0.3149
#> b        0.4530 0.0263 20000 0.3665 0.3850 0.4000 0.5038 0.5197 0.5380
#> a        0.4411 0.0255 20000 0.3550 0.3727 0.3894 0.4900 0.5050 0.5245
#> Y~~Y     0.6563 0.0243 20000 0.5762 0.5930 0.6083 0.7033 0.7178 0.7329
#> M~~M     0.8055 0.0225 20000 0.7249 0.7450 0.7599 0.8484 0.8611 0.8740
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1998 0.0167 20000 0.1482 0.1576 0.1680 0.2330 0.2437 0.2549
#> direct   0.2226 0.0282 20000 0.1334 0.1502 0.1665 0.2776 0.2949 0.3149
#> total    0.4224 0.0260 20000 0.3306 0.3532 0.3696 0.4720 0.4869 0.5026
```

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