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
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.

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

Data

```r
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       0.2225 0.0350 20000 0.1103 0.1341 0.1548 0.2913 0.3105 0.3354
#> b        0.5569 0.0301 20000 0.4572 0.4789 0.4976 0.6151 0.6334 0.6545
#> a        0.5019 0.0329 20000 0.3913 0.4154 0.4373 0.5666 0.5875 0.6067
#> Y~~Y     0.9600 0.0429 20000 0.8107 0.8486 0.8745 1.0426 1.0709 1.1064
#> M~~M     1.0622 0.0474 20000 0.9031 0.9419 0.9689 1.1550 1.1836 1.2244
#> indirect 0.2795 0.0238 20000 0.2090 0.2197 0.2341 0.3275 0.3434 0.3572
#> direct   0.2225 0.0350 20000 0.1103 0.1341 0.1548 0.2913 0.3105 0.3354
```

```
#> total    0.5020 0.0365 20000 0.3792 0.4080 0.4304 0.5740 0.5962 0.6195
```

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        1.182580e-03 -4.427032e-04  2.951667e-06  5.097345e-07  7.507435e-06
#> b        -4.427032e-04  9.073113e-04 -1.171829e-05  8.364712e-06  6.125018e-07
#> a         2.951667e-06 -1.171829e-05  1.080962e-03 -6.686947e-06 -2.481162e-06
#> Y~~Y      5.097345e-07  8.364712e-06 -6.686947e-06  1.854248e-03  2.568886e-05
#> M~~M      7.507435e-06  6.125018e-07 -2.481162e-06  2.568886e-05  2.253418e-03
#> X~~X     -1.491607e-05  4.795868e-07 -6.351350e-06 -4.567065e-06  5.927714e-06
#> indirect -2.205899e-04  4.483274e-04  5.957797e-04  2.224682e-07 -1.159061e-06
#> direct    1.182580e-03 -4.427032e-04  2.951667e-06  5.097345e-07  7.507435e-06
#> total     9.619904e-04  5.624247e-06  5.987313e-04  7.322027e-07  6.348374e-06
#>                   X~~X      indirect        direct         total
#> cp       -1.491607e-05 -2.205899e-04  1.182580e-03  9.619904e-04
#> b         4.795868e-07  4.483274e-04 -4.427032e-04  5.624247e-06
#> a        -6.351350e-06  5.957797e-04  2.951667e-06  5.987313e-04
#> Y~~Y     -4.567065e-06  2.224682e-07  5.097345e-07  7.322027e-07
```

```
#> M~~M      5.927714e-06 -1.159061e-06  7.507435e-06  6.348374e-06
#> X~~X      1.905159e-03 -3.146650e-06 -1.491607e-05 -1.806272e-05
#> indirect -3.146650e-06  5.573494e-04 -2.205899e-04  3.367595e-04
#> direct   -1.491607e-05 -2.205899e-04  1.182580e-03  9.619904e-04
#> total    -1.806272e-05  3.367595e-04  9.619904e-04  1.298750e-03
```

```
MCStd(unstd)
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1779 0.0272 20000 0.0897 0.1066 0.1239 0.2311 0.2468 0.2659
#> b        0.5141 0.0246 20000 0.4296 0.4492 0.4650 0.5617 0.5754 0.5922
#> a        0.4347 0.0256 20000 0.3482 0.3676 0.3831 0.4848 0.4990 0.5161
#> Y~~Y     0.6246 0.0241 20000 0.5402 0.5606 0.5769 0.6710 0.6853 0.7002
#> M~~M     0.8110 0.0222 20000 0.7336 0.7510 0.7650 0.8532 0.8649 0.8788
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2235 0.0173 20000 0.1693 0.1807 0.1899 0.2575 0.2688 0.2801
#> direct   0.1779 0.0272 20000 0.0897 0.1066 0.1239 0.2311 0.2468 0.2659
#> total    0.4014 0.0263 20000 0.3119 0.3335 0.3487 0.4522 0.4673 0.4859
```

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