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:

```
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       0.2837 0.0356 20000 0.1667 0.1921 0.2137 0.3527 0.3763 0.3997
#> b        0.4969 0.0314 20000 0.3925 0.4168 0.4356 0.5592 0.5796 0.5991
#> a        0.5596 0.0310 20000 0.4593 0.4813 0.4986 0.6202 0.6390 0.6608
#> Y~~Y     1.0344 0.0461 20000 0.8796 0.9160 0.9449 1.1244 1.1537 1.1852
#> M~~M     1.0535 0.0471 20000 0.8971 0.9315 0.9615 1.1463 1.1738 1.2031
#> indirect 0.2781 0.0233 20000 0.2065 0.2203 0.2341 0.3251 0.3394 0.3567
#> direct   0.2837 0.0356 20000 0.1667 0.1921 0.2137 0.3527 0.3763 0.3997
```

```
#> total    0.5618 0.0346 20000 0.4493 0.4734 0.4943 0.6300 0.6519 0.6759
```

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        1.265520e-03 -5.439268e-04 -5.431685e-06  7.520291e-06 -7.545670e-06
#> b        -5.439268e-04  9.782463e-04  1.541557e-05  6.074766e-06 -6.519634e-07
#> a        -5.431685e-06  1.541557e-05  9.837705e-04  1.653039e-06  6.348253e-07
#> Y~~Y      7.520291e-06  6.074766e-06  1.653039e-06  2.134728e-03 -2.068947e-05
#> M~~M     -7.545670e-06 -6.519634e-07  6.348253e-07 -2.068947e-05  2.214561e-03
#> X~~X     -3.361253e-05  1.694030e-05  8.253379e-06 -3.027444e-06 -1.579370e-05
#> indirect -3.072738e-04  5.557114e-04  4.976298e-04  4.548969e-06  1.690844e-07
#> direct    1.265520e-03 -5.439268e-04 -5.431685e-06  7.520291e-06 -7.545670e-06
#> total     9.582462e-04  1.178458e-05  4.921981e-04  1.206926e-05 -7.376586e-06
#>                    X~~X      indirect        direct         total
#> cp       -3.361253e-05 -3.072738e-04  1.265520e-03  9.582462e-04
#> b         1.694030e-05  5.557114e-04 -5.439268e-04  1.178458e-05
#> a         8.253379e-06  4.976298e-04 -5.431685e-06  4.921981e-04
#> Y~~Y     -3.027444e-06  4.548969e-06  7.520291e-06  1.206926e-05
```

```
#> M~~M     -1.579370e-05  1.690844e-07 -7.545670e-06 -7.376586e-06
#> X~~X      2.302984e-03  1.354286e-05 -3.361253e-05 -2.006967e-05
#> indirect  1.354286e-05  5.596405e-04 -3.072738e-04  2.523667e-04
#> direct   -3.361253e-05 -3.072738e-04  1.265520e-03  9.582462e-04
#> total    -2.006967e-05  2.523667e-04  9.582462e-04  1.210613e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2302 0.0283 20000 0.1366 0.1555 0.1737 0.2854 0.3031 0.3238
#> b         0.4584 0.0266 20000 0.3680 0.3905 0.4054 0.5097 0.5281 0.5430
#> a         0.4922 0.0241 20000 0.4124 0.4283 0.4442 0.5385 0.5520 0.5689
#> Y~~Y      0.6330 0.0242 20000 0.5521 0.5694 0.5842 0.6795 0.6941 0.7087
#> M~~M      0.7577 0.0236 20000 0.6764 0.6953 0.7100 0.8027 0.8165 0.8300
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2256 0.0176 20000 0.1709 0.1814 0.1921 0.2608 0.2721 0.2844
#> direct    0.2302 0.0283 20000 0.1366 0.1555 0.1737 0.2854 0.3031 0.3238
#> total     0.4558 0.0249 20000 0.3742 0.3901 0.4056 0.5032 0.5171 0.5318
```

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