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       0.2632 0.0377 20000 0.1427 0.1643 0.1884 0.3371 0.3600 0.3840
#> b        0.5381 0.0320 20000 0.4328 0.4572 0.4749 0.6009 0.6196 0.6422
#> a        0.5128 0.0334 20000 0.4073 0.4244 0.4474 0.5791 0.5990 0.6223
#> Y~~Y     1.0363 0.0465 20000 0.8808 0.9153 0.9441 1.1262 1.1572 1.1895
#> M~~M     1.0197 0.0454 20000 0.8751 0.9040 0.9300 1.1075 1.1348 1.1631
#> indirect 0.2759 0.0243 20000 0.2039 0.2166 0.2296 0.3244 0.3415 0.3598
#> direct   0.2632 0.0377 20000 0.1427 0.1643 0.1884 0.3371 0.3600 0.3840
```

```
#> total    0.5391 0.0384 20000 0.4100 0.4394 0.4639 0.6149 0.6364 0.6588
```

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        1.426391e-03 -5.211607e-04 -5.798644e-06 -4.167203e-06 -1.813904e-05
#> b        -5.211607e-04  1.011513e-03  7.848796e-07  2.274588e-05  3.022542e-08
#> a        -5.798644e-06  7.848796e-07  1.117728e-03  1.013629e-05 -8.542974e-06
#> Y~~Y     -4.167203e-06  2.274588e-05  1.013629e-05  2.153832e-03 -1.955535e-05
#> M~~M     -1.813904e-05  3.022542e-08 -8.542974e-06 -1.955535e-05  2.075496e-03
#> X~~X     -1.537202e-05  4.481600e-06 -7.411760e-07 -7.665896e-06 -3.108761e-05
#> indirect -2.695316e-04  5.185783e-04  6.018017e-04  1.728794e-05 -4.035416e-06
#> direct    1.426391e-03 -5.211607e-04 -5.798644e-06 -4.167203e-06 -1.813904e-05
#> total     1.156860e-03 -2.582403e-06  5.960030e-04  1.312073e-05 -2.217446e-05
#>                    X~~X      indirect        direct         total
#> cp       -1.537202e-05 -2.695316e-04  1.426391e-03  1.156860e-03
#> b         4.481600e-06  5.185783e-04 -5.211607e-04 -2.582403e-06
#> a        -7.411760e-07  6.018017e-04 -5.798644e-06  5.960030e-04
#> Y~~Y     -7.665896e-06  1.728794e-05 -4.167203e-06  1.312073e-05
```

```
#> M~~M     -3.108761e-05 -4.035416e-06 -1.813904e-05 -2.217446e-05
#> X~~X      1.660655e-03  2.098855e-06 -1.537202e-05 -1.327317e-05
#> indirect  2.098855e-06  5.905894e-04 -2.695316e-04  3.210578e-04
#> direct   -1.537202e-05 -2.695316e-04  1.426391e-03  1.156860e-03
#> total    -1.327317e-05  3.210578e-04  1.156860e-03  1.477917e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1989 0.0282 20000 0.1076 0.1267 0.1441 0.2541 0.2709 0.2913
#> b         0.4779 0.0256 20000 0.3887 0.4101 0.4262 0.5266 0.5410 0.5585
#> a         0.4363 0.0257 20000 0.3493 0.3687 0.3848 0.4860 0.5002 0.5172
#> Y~~Y      0.6491 0.0241 20000 0.5728 0.5861 0.6008 0.6952 0.7092 0.7244
#> M~~M      0.8096 0.0224 20000 0.7325 0.7498 0.7638 0.8519 0.8640 0.8780
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2085 0.0169 20000 0.1525 0.1668 0.1758 0.2417 0.2534 0.2666
#> direct    0.1989 0.0282 20000 0.1076 0.1267 0.1441 0.2541 0.2709 0.2913
#> total     0.4074 0.0264 20000 0.3183 0.3375 0.3544 0.4571 0.4725 0.4944
```

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