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.

```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       0.2637 0.0345 20000 0.1547 0.1757 0.1958 0.3315 0.3533 0.3776
#> b        0.5231 0.0303 20000 0.4200 0.4443 0.4638 0.5823 0.5992 0.6214
#> a        0.5239 0.0319 20000 0.4204 0.4432 0.4621 0.5874 0.6073 0.6320
#> Y~~Y     1.0079 0.0451 20000 0.8621 0.8930 0.9199 1.0962 1.1253 1.1591
#> M~~M     1.0960 0.0489 20000 0.9394 0.9713 1.0000 1.1916 1.2216 1.2618
#> indirect 0.2740 0.0231 20000 0.2017 0.2170 0.2304 0.3210 0.3365 0.3545
#> direct   0.2637 0.0345 20000 0.1547 0.1757 0.1958 0.3315 0.3533 0.3776
```

```
#> total    0.5377 0.0348 20000 0.4232 0.4494 0.4696 0.6055 0.6276 0.6553
```

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        1.183457e-03 -4.672125e-04 -4.010968e-06  2.034034e-05 -1.236135e-05
#> b        -4.672125e-04  9.062029e-04  4.009830e-06 -3.714450e-06  1.121451e-05
#> a        -4.010968e-06  4.009830e-06  1.025590e-03  1.267724e-05 -1.044150e-05
#> Y~~Y      2.034034e-05 -3.714450e-06  1.267724e-05  2.035663e-03 -3.517488e-06
#> M~~M     -1.236135e-05  1.121451e-05 -1.044150e-05 -3.517488e-06  2.386961e-03
#> X~~X     -1.016698e-05 -8.023860e-06 -1.542718e-05 -1.148704e-05 -9.183586e-06
#> indirect -2.466305e-04  4.772388e-04  5.394411e-04  4.723409e-06  6.562251e-07
#> direct    1.183457e-03 -4.672125e-04 -4.010968e-06  2.034034e-05 -1.236135e-05
#> total     9.368266e-04  1.002629e-05  5.354301e-04  2.506375e-05 -1.170512e-05
#>                    X~~X      indirect        direct         total
#> cp       -1.016698e-05 -2.466305e-04  1.183457e-03  9.368266e-04
#> b        -8.023860e-06  4.772388e-04 -4.672125e-04  1.002629e-05
#> a        -1.542718e-05  5.394411e-04 -4.010968e-06  5.354301e-04
#> Y~~Y     -1.148704e-05  4.723409e-06  2.034034e-05  2.506375e-05
```

```
#> M~~M     -9.183586e-06  6.562251e-07 -1.236135e-05 -1.170512e-05
#> X~~X      2.289657e-03 -1.187304e-05 -1.016698e-05 -2.204002e-05
#> indirect -1.187304e-05  5.337666e-04 -2.466305e-04  2.871361e-04
#> direct   -1.016698e-05 -2.466305e-04  1.183457e-03  9.368266e-04
#> total    -2.204002e-05  2.871361e-04  9.368266e-04  1.223963e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2146 0.0276 20000 0.1209 0.1431 0.1599 0.2682 0.2834 0.2983
#> b        0.4849 0.0254 20000 0.3997 0.4194 0.4343 0.5342 0.5504 0.5666
#> a        0.4600 0.0249 20000 0.3819 0.3960 0.4106 0.5085 0.5230 0.5393
#> Y~~Y     0.6231 0.0240 20000 0.5399 0.5600 0.5762 0.6695 0.6839 0.6987
#> M~~M     0.7884 0.0229 20000 0.7092 0.7264 0.7414 0.8314 0.8432 0.8541
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2231 0.0171 20000 0.1709 0.1809 0.1906 0.2570 0.2683 0.2840
#> direct   0.2146 0.0276 20000 0.1209 0.1431 0.1599 0.2682 0.2834 0.2983
#> total    0.4377 0.0254 20000 0.3533 0.3700 0.3860 0.4856 0.4998 0.5140
```

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