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       0.1512 0.0368 20000 0.0309 0.0563 0.0793 0.2225 0.2441 0.2704
#> b        0.5264 0.0340 20000 0.4174 0.4397 0.4589 0.5915 0.6123 0.6411
#> a        0.5296 0.0300 20000 0.4338 0.4529 0.4706 0.5887 0.6069 0.6278
#> Y~~Y     1.0806 0.0480 20000 0.9245 0.9553 0.9859 1.1749 1.2038 1.2372
#> M~~M     0.9230 0.0416 20000 0.7848 0.8139 0.8396 1.0029 1.0259 1.0529
#> indirect 0.2788 0.0238 20000 0.2063 0.2204 0.2328 0.3261 0.3432 0.3614
#> direct   0.1512 0.0368 20000 0.0309 0.0563 0.0793 0.2225 0.2441 0.2704
```

```
#> total    0.4300 0.0358 20000 0.3070 0.3377 0.3606 0.4997 0.5221 0.5457
```

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        1.388887e-03 -6.175063e-04 -1.169789e-06 -1.130259e-05  8.214393e-06
#> b        -6.175063e-04  1.172549e-03  8.454574e-06 -1.903688e-06 -1.082288e-05
#> a        -1.169789e-06  8.454574e-06  8.983508e-04  7.504443e-06 -3.035776e-06
#> Y~~Y     -1.130259e-05 -1.903688e-06  7.504443e-06  2.356654e-03 -5.280160e-06
#> M~~M      8.214393e-06 -1.082288e-05 -3.035776e-06 -5.280160e-06  1.672836e-03
#> X~~X     -1.101334e-05  1.222459e-05  3.025974e-05  2.419587e-05 -4.404001e-06
#> indirect -3.273686e-04  6.252390e-04  4.770714e-04  2.911100e-06 -6.736650e-06
#> direct    1.388887e-03 -6.175063e-04 -1.169789e-06 -1.130259e-05  8.214393e-06
#> total     1.061519e-03  7.732767e-06  4.759016e-04 -8.391490e-06  1.477743e-06
#>                   X~~X      indirect        direct         total
#> cp       -1.101334e-05 -3.273686e-04  1.388887e-03  1.061519e-03
#> b         1.222459e-05  6.252390e-04 -6.175063e-04  7.732767e-06
#> a         3.025974e-05  4.770714e-04 -1.169789e-06  4.759016e-04
#> Y~~Y      2.419587e-05  2.911100e-06 -1.130259e-05 -8.391490e-06
```

```
#> M~~M     -4.404001e-06 -6.736650e-06  8.214393e-06  1.477743e-06
#> X~~X      2.161436e-03  2.291676e-05 -1.101334e-05  1.190341e-05
#> indirect  2.291676e-05  5.830909e-04 -3.273686e-04  2.557223e-04
#> direct   -1.101334e-05 -3.273686e-04  1.388887e-03  1.061519e-03
#> total     1.190341e-05  2.557223e-04  1.061519e-03  1.317241e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1242 0.0304 20000 0.0226 0.0449 0.0638 0.1843 0.2029 0.2245
#> b        0.4690 0.0279 20000 0.3705 0.3964 0.4128 0.5223 0.5388 0.5560
#> a        0.4883 0.0243 20000 0.4064 0.4258 0.4399 0.5347 0.5505 0.5647
#> Y~~Y     0.7078 0.0243 20000 0.6246 0.6441 0.6596 0.7545 0.7676 0.7837
#> M~~M     0.7616 0.0237 20000 0.6812 0.6970 0.7141 0.8065 0.8187 0.8348
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2290 0.0184 20000 0.1722 0.1823 0.1936 0.2650 0.2780 0.2925
#> direct   0.1242 0.0304 20000 0.0226 0.0449 0.0638 0.1843 0.2029 0.2245
#> total    0.3532 0.0279 20000 0.2604 0.2803 0.2975 0.4071 0.4242 0.4431
```

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