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       0.2855 0.0351 20000 0.1661 0.1948 0.2176 0.3545 0.3757 0.4012
#> b        0.4535 0.0302 20000 0.3589 0.3758 0.3943 0.5129 0.5335 0.5528
#> a        0.5576 0.0316 20000 0.4549 0.4758 0.4955 0.6189 0.6381 0.6605
#> Y~~Y     0.9360 0.0415 20000 0.7952 0.8305 0.8544 1.0164 1.0441 1.0667
#> M~~M     1.0202 0.0456 20000 0.8744 0.9010 0.9296 1.1089 1.1345 1.1593
#> indirect 0.2529 0.0222 20000 0.1837 0.1972 0.2108 0.2980 0.3128 0.3315
#> direct   0.2855 0.0351 20000 0.1661 0.1948 0.2176 0.3545 0.3757 0.4012
```

```
#> total    0.5384 0.0339 20000 0.4248 0.4516 0.4722 0.6048 0.6260 0.6530
```

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        1.234370e-03 -5.174112e-04  2.013376e-06  6.839179e-06 -1.594928e-06
#> b        -5.174112e-04  9.127017e-04  2.476993e-06 -2.217479e-05  4.750213e-06
#> a         2.013376e-06  2.476993e-06  1.018412e-03  1.312570e-05 -1.554844e-05
#> Y~~Y      6.839179e-06 -2.217479e-05  1.312570e-05  1.742595e-03 -1.067370e-05
#> M~~M     -1.594928e-06  4.750213e-06 -1.554844e-05 -1.067370e-05  2.090653e-03
#> X~~X     -2.870917e-05  1.130945e-05  3.871185e-06 -1.044475e-05  1.441016e-05
#> indirect -2.878043e-04  5.105203e-04  4.634855e-04 -6.099182e-06 -4.385911e-06
#> direct    1.234370e-03 -5.174112e-04  2.013376e-06  6.839179e-06 -1.594928e-06
#> total     9.465653e-04 -6.890896e-06  4.654988e-04  7.399970e-07 -5.980840e-06
#>                   X~~X      indirect        direct         total
#> cp       -2.870917e-05 -2.878043e-04  1.234370e-03  9.465653e-04
#> b         1.130945e-05  5.105203e-04 -5.174112e-04 -6.890896e-06
#> a         3.871185e-06  4.634855e-04  2.013376e-06  4.654988e-04
#> Y~~Y     -1.044475e-05 -6.099182e-06  6.839179e-06  7.399970e-07
```

```
#> M~~M      1.441016e-05 -4.385911e-06 -1.594928e-06 -5.980840e-06
#> X~~X      2.029501e-03  7.839901e-06 -2.870917e-05 -2.086927e-05
#> indirect  7.839901e-06  4.961489e-04 -2.878043e-04  2.083446e-04
#> direct   -2.870917e-05 -2.878043e-04  1.234370e-03  9.465653e-04
#> total    -2.086927e-05  2.083446e-04  9.465653e-04  1.154910e-03
```

```
MCStd(unstd)
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2387 0.0288 20000 0.1440 0.1634 0.1817 0.2946 0.3103 0.3297
#> b        0.4367 0.0272 20000 0.3464 0.3665 0.3824 0.4890 0.5054 0.5261
#> a        0.4841 0.0243 20000 0.4027 0.4203 0.4348 0.5304 0.5455 0.5624
#> Y~~Y     0.6514 0.0243 20000 0.5720 0.5892 0.6032 0.6981 0.7130 0.7291
#> M~~M     0.7656 0.0235 20000 0.6837 0.7024 0.7187 0.8109 0.8234 0.8378
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2114 0.0173 20000 0.1574 0.1681 0.1785 0.2457 0.2576 0.2716
#> direct   0.2387 0.0288 20000 0.1440 0.1634 0.1817 0.2946 0.3103 0.3297
#> total    0.4501 0.0252 20000 0.3636 0.3828 0.3996 0.4981 0.5122 0.5265
```

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