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:

```
if (!require("remotes")) 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       0.3057 0.0346 20000 0.1955 0.2185 0.2374 0.3728 0.3960 0.4200
#> b        0.4860 0.0307 20000 0.3845 0.4072 0.4262 0.5461 0.5658 0.5878
#> a        0.4773 0.0323 20000 0.3728 0.3947 0.4146 0.5408 0.5616 0.5870
#> Y~~Y     0.9793 0.0434 20000 0.8361 0.8632 0.8925 1.0636 1.0891 1.1187
#> M~~M     1.0394 0.0466 20000 0.8888 0.9196 0.9486 1.1310 1.1598 1.1957
#> indirect 0.2320 0.0215 20000 0.1663 0.1793 0.1923 0.2759 0.2908 0.3099
#> direct   0.3057 0.0346 20000 0.1955 0.2185 0.2374 0.3728 0.3960 0.4200
```

```
#> total    0.5377 0.0350 20000 0.4267 0.4488 0.4690 0.6059 0.6280 0.6523
```

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        1.207168e-03 -4.530688e-04  3.339815e-06  9.533463e-06 -2.034158e-05
#> b        -4.530688e-04  9.377697e-04  6.751805e-06  1.187765e-05 -9.640353e-07
#> a         3.339815e-06  6.751805e-06  1.059730e-03 -4.965706e-06 -1.691463e-05
#> Y~~Y      9.533463e-06  1.187765e-05 -4.965706e-06  1.926560e-03 -3.756805e-05
#> M~~M     -2.034158e-05 -9.640353e-07 -1.691463e-05 -3.756805e-05  2.138964e-03
#> X~~X      1.221838e-05 -2.696081e-06 -1.179214e-06  9.500181e-06  6.569991e-06
#> indirect -2.149349e-04  4.512640e-04  5.185311e-04  3.294611e-06 -8.786773e-06
#> direct    1.207168e-03 -4.530688e-04  3.339815e-06  9.533463e-06 -2.034158e-05
#> total     9.922334e-04 -1.804797e-06  5.218709e-04  1.282807e-05 -2.912835e-05
#>                   X~~X      indirect        direct         total
#> cp        1.221838e-05 -2.149349e-04  1.207168e-03  9.922334e-04
#> b        -2.696081e-06  4.512640e-04 -4.530688e-04 -1.804797e-06
#> a        -1.179214e-06  5.185311e-04  3.339815e-06  5.218709e-04
#> Y~~Y      9.500181e-06  3.294611e-06  9.533463e-06  1.282807e-05
```

```
#> M~~M      6.569991e-06 -8.786773e-06 -2.034158e-05 -2.912835e-05
#> X~~X      1.963439e-03 -1.821838e-06  1.221838e-05  1.039654e-05
#> indirect -1.821838e-06  4.686893e-04 -2.149349e-04  2.537544e-04
#> direct    1.221838e-05 -2.149349e-04  1.207168e-03  9.922334e-04
#> total     1.039654e-05  2.537544e-04  9.922334e-04  1.245988e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2482 0.0277 20000 0.1612 0.1770 0.1938 0.3021 0.3191 0.3373
#> b        0.4447 0.0259 20000 0.3579 0.3779 0.3940 0.4946 0.5101 0.5271
#> a        0.4236 0.0262 20000 0.3335 0.3546 0.3719 0.4743 0.4881 0.5068
#> Y~~Y     0.6472 0.0242 20000 0.5652 0.5828 0.5988 0.6932 0.7087 0.7242
#> M~~M     0.8206 0.0221 20000 0.7432 0.7617 0.7751 0.8617 0.8742 0.8888
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1883 0.0162 20000 0.1380 0.1479 0.1573 0.2208 0.2318 0.2427
#> direct   0.2482 0.0277 20000 0.1612 0.1770 0.1938 0.3021 0.3191 0.3373
#> total    0.4365 0.0257 20000 0.3516 0.3691 0.3851 0.4858 0.5005 0.5204
```

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. (2023). *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