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       0.2921 0.0349 20000 0.1732 0.2026 0.2242 0.3607 0.3827 0.4008
#> b        0.5030 0.0313 20000 0.4035 0.4222 0.4413 0.5642 0.5844 0.6040
#> a        0.5465 0.0308 20000 0.4439 0.4668 0.4863 0.6072 0.6245 0.6493
#> Y~~Y     0.9450 0.0423 20000 0.8063 0.8368 0.8621 1.0286 1.0535 1.0793
#> M~~M     0.9727 0.0433 20000 0.8361 0.8606 0.8881 1.0583 1.0829 1.1169
#> indirect 0.2749 0.0229 20000 0.2023 0.2180 0.2314 0.3204 0.3354 0.3545
#> direct   0.2921 0.0349 20000 0.1732 0.2026 0.2242 0.3607 0.3827 0.4008
```

```
#> total    0.5670 0.0343 20000 0.4572 0.4799 0.5003 0.6347 0.6533 0.6822
```

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        1.224378e-03 -5.372500e-04 -1.242616e-05  1.280750e-05 -8.147902e-06
#> b        -5.372500e-04  9.839443e-04  1.141588e-05  4.823390e-06 -7.950597e-06
#> a        -1.242616e-05  1.141588e-05  9.642352e-04 -4.254520e-06  1.254941e-05
#> Y~~Y      1.280750e-05  4.823390e-06 -4.254520e-06  1.809770e-03 -2.792618e-05
#> M~~M     -8.147902e-06 -7.950597e-06  1.254941e-05 -2.792618e-05  1.895237e-03
#> X~~X      1.384008e-06  1.017601e-05  2.314066e-06  1.419836e-05 -4.769526e-06
#> indirect -3.002248e-04  5.437053e-04  4.912155e-04  7.273850e-07  1.769244e-06
#> direct    1.224378e-03 -5.372500e-04 -1.242616e-05  1.280750e-05 -8.147902e-06
#> total     9.241535e-04  6.455261e-06  4.787893e-04  1.353489e-05 -6.378658e-06
#>                   X~~X      indirect        direct         total
#> cp        1.384008e-06 -3.002248e-04  1.224378e-03  9.241535e-04
#> b         1.017601e-05  5.437053e-04 -5.372500e-04  6.455261e-06
#> a         2.314066e-06  4.912155e-04 -1.242616e-05  4.787893e-04
#> Y~~Y      1.419836e-05  7.273850e-07  1.280750e-05  1.353489e-05
```

```
#> M~~M     -4.769526e-06  1.769244e-06 -8.147902e-06 -6.378658e-06
#> X~~X      2.051593e-03  6.291085e-06  1.384008e-06  7.675093e-06
#> indirect  6.291085e-06  5.452806e-04 -3.002248e-04  2.450558e-04
#> direct    1.384008e-06 -3.002248e-04  1.224378e-03  9.241535e-04
#> total     7.675093e-06  2.450558e-04  9.241535e-04  1.169209e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2390 0.0282 20000 0.1408 0.1637 0.1823 0.2934 0.3108 0.3261
#> b        0.4612 0.0265 20000 0.3719 0.3915 0.4086 0.5128 0.5288 0.5443
#> a        0.4876 0.0241 20000 0.4048 0.4242 0.4387 0.5331 0.5477 0.5618
#> Y~~Y     0.6227 0.0242 20000 0.5470 0.5602 0.5752 0.6702 0.6839 0.7002
#> M~~M     0.7622 0.0235 20000 0.6844 0.7000 0.7158 0.8075 0.8201 0.8361
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2249 0.0175 20000 0.1693 0.1805 0.1909 0.2596 0.2713 0.2838
#> direct   0.2390 0.0282 20000 0.1408 0.1637 0.1823 0.2934 0.3108 0.3261
#> total    0.4639 0.0248 20000 0.3746 0.3977 0.4130 0.5105 0.5252 0.5394
```

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