semmcci: Monte Carlo Confidence Intervals

Ivan Jacob Agaloos Pesigan

Installation

You can install the CRAN release of `semmcci` with:

```r
install.packages("semmcci")
```

You can install the development version of `semmcci` from GitHub with:

```r
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       0.2655 0.0351 20000 0.1506 0.1746 0.1971 0.3344 0.3589 0.3851
#> b        0.4789 0.0325 20000 0.3733 0.3949 0.4149 0.5425 0.5629 0.5845
#> a        0.5007 0.0304 20000 0.4019 0.4216 0.4407 0.5599 0.5781 0.6002
#> Y~~Y     1.0282 0.0460 20000 0.8789 0.9094 0.9380 1.1185 1.1476 1.1790
#> M~~M     0.9805 0.0441 20000 0.8348 0.8673 0.8938 1.0681 1.0941 1.1291
#> indirect 0.2398 0.0218 20000 0.1727 0.1861 0.1986 0.2836 0.2978 0.3189
#> direct   0.2655 0.0351 20000 0.1506 0.1746 0.1971 0.3344 0.3589 0.3851
```

```
#> total    0.5052 0.0343 20000 0.3909 0.4175 0.4378 0.5729 0.5950 0.6162
```

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        1.234292e-03 -5.216187e-04 -4.816714e-06  1.328298e-06  1.301516e-06
#> b        -5.216187e-04  1.047271e-03  5.638122e-06  2.995834e-05  4.322644e-06
#> a        -4.816714e-06  5.638122e-06  9.210412e-04 -1.590807e-06 -1.536983e-05
#> Y~~Y      1.328298e-06  2.995834e-05 -1.590807e-06  2.123873e-03  1.421379e-05
#> M~~M      1.301516e-06  4.322644e-06 -1.536983e-05  1.421379e-05  1.941821e-03
#> X~~X     -1.501947e-05 -1.441615e-05 -4.310132e-06 -5.421911e-06 -1.258838e-05
#> indirect -2.633015e-04  5.266450e-04  4.441420e-04  1.444209e-05 -5.489043e-06
#> direct    1.234292e-03 -5.216187e-04 -4.816714e-06  1.328298e-06  1.301516e-06
#> total     9.709907e-04  5.026318e-06  4.393253e-04  1.577039e-05 -4.187528e-06
#>                   X~~X      indirect        direct         total
#> cp       -1.501947e-05 -2.633015e-04  1.234292e-03  9.709907e-04
#> b        -1.441615e-05  5.266450e-04 -5.216187e-04  5.026318e-06
#> a        -4.310132e-06  4.441420e-04 -4.816714e-06  4.393253e-04
#> Y~~Y     -5.421911e-06  1.444209e-05  1.328298e-06  1.577039e-05
```

```
#> M~~M     -1.258838e-05 -5.489043e-06  1.301516e-06 -4.187528e-06
#> X~~X      2.255581e-03 -9.034498e-06 -1.501947e-05 -2.405396e-05
#> indirect -9.034498e-06  4.772504e-04 -2.633015e-04  2.139489e-04
#> direct   -1.501947e-05 -2.633015e-04  1.234292e-03  9.709907e-04
#> total    -2.405396e-05  2.139489e-04  9.709907e-04  1.184940e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2216 0.0288 20000 0.1248 0.1489 0.1648 0.2781 0.2952 0.3139
#> b        0.4331 0.0271 20000 0.3418 0.3623 0.3796 0.4852 0.5014 0.5215
#> a        0.4620 0.0249 20000 0.3781 0.3975 0.4124 0.5104 0.5244 0.5423
#> Y~~Y     0.6746 0.0241 20000 0.5911 0.6104 0.6260 0.7205 0.7347 0.7525
#> M~~M     0.7865 0.0230 20000 0.7059 0.7250 0.7395 0.8299 0.8420 0.8570
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2001 0.0169 20000 0.1470 0.1579 0.1680 0.2335 0.2445 0.2563
#> direct   0.2216 0.0288 20000 0.1248 0.1489 0.1648 0.2781 0.2952 0.3139
#> total    0.4217 0.0259 20000 0.3344 0.3543 0.3706 0.4714 0.4875 0.5020
```

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