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       0.3282 0.0367 20000 0.2066 0.2337 0.2563 0.4000 0.4209 0.4517
#> b        0.4255 0.0337 20000 0.3164 0.3404 0.3600 0.4911 0.5116 0.5365
#> a        0.5501 0.0297 20000 0.4489 0.4738 0.4924 0.6076 0.6265 0.6466
#> Y~~Y     1.0478 0.0471 20000 0.8867 0.9280 0.9556 1.1412 1.1704 1.1983
#> M~~M     0.9255 0.0413 20000 0.7877 0.8198 0.8443 1.0055 1.0311 1.0599
#> indirect 0.2340 0.0225 20000 0.1647 0.1799 0.1917 0.2792 0.2939 0.3102
#> direct   0.3282 0.0367 20000 0.2066 0.2337 0.2563 0.4000 0.4209 0.4517
```

```
#> total    0.5623 0.0340 20000 0.4494 0.4733 0.4954 0.6291 0.6499 0.6716
```

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        1.357137e-03 -6.292434e-04 -6.274844e-06 -2.736703e-06 -2.918519e-06
#> b        -6.292434e-04  1.133446e-03  2.733565e-06  2.030669e-05  3.909645e-07
#> a        -6.274844e-06  2.733565e-06  8.740715e-04 -1.087216e-05  6.753986e-06
#> Y~~Y     -2.736703e-06  2.030669e-05 -1.087216e-05  2.208853e-03 -1.366887e-05
#> M~~M     -2.918519e-06  3.909645e-07  6.753986e-06 -1.366887e-05  1.687252e-03
#> X~~X      1.542913e-05 -1.470194e-06  7.774431e-06 -6.936199e-06 -6.361887e-06
#> indirect -3.483351e-04  6.241646e-04  3.734366e-04  6.803019e-06  3.222831e-06
#> direct    1.357137e-03 -6.292434e-04 -6.274844e-06 -2.736703e-06 -2.918519e-06
#> total     1.008802e-03 -5.078854e-06  3.671618e-04  4.066316e-06  3.043120e-07
#>                   X~~X      indirect        direct         total
#> cp        1.542913e-05 -3.483351e-04  1.357137e-03  1.008802e-03
#> b        -1.470194e-06  6.241646e-04 -6.292434e-04 -5.078854e-06
#> a         7.774431e-06  3.734366e-04 -6.274844e-06  3.671618e-04
#> Y~~Y     -6.936199e-06  6.803019e-06 -2.736703e-06  4.066316e-06
```

```
#> M~~M     -6.361887e-06  3.222831e-06 -2.918519e-06  3.043120e-07
#> X~~X      2.169247e-03  2.420894e-06  1.542913e-05  1.785003e-05
#> indirect  2.420894e-06  5.029584e-04 -3.483351e-04  1.546234e-04
#> direct    1.542913e-05 -3.483351e-04  1.357137e-03  1.008802e-03
#> total     1.785003e-05  1.546234e-04  1.008802e-03  1.163426e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2693 0.0296 20000 0.1715 0.1932 0.2118 0.3273 0.3458 0.3655
#> b        0.3813 0.0286 20000 0.2879 0.3061 0.3245 0.4360 0.4525 0.4737
#> a        0.5037 0.0234 20000 0.4240 0.4426 0.4570 0.5488 0.5626 0.5764
#> Y~~Y     0.6786 0.0242 20000 0.5955 0.6148 0.6291 0.7242 0.7388 0.7565
#> M~~M     0.7463 0.0236 20000 0.6678 0.6834 0.6988 0.7912 0.8041 0.8202
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1920 0.0173 20000 0.1382 0.1494 0.1585 0.2258 0.2371 0.2512
#> direct   0.2693 0.0296 20000 0.1715 0.1932 0.2118 0.3273 0.3458 0.3655
#> total    0.4614 0.0249 20000 0.3749 0.3957 0.4117 0.5095 0.5243 0.5378
```

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