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       0.2139 0.0349 20000 0.1014 0.1250 0.1447 0.2820 0.3023 0.3274
#> b        0.5055 0.0319 20000 0.3979 0.4229 0.4436 0.5677 0.5888 0.6106
#> a        0.5056 0.0307 20000 0.4037 0.4270 0.4455 0.5663 0.5854 0.6077
#> Y~~Y     0.9832 0.0438 20000 0.8411 0.8711 0.8972 1.0694 1.0953 1.1222
#> M~~M     0.9780 0.0434 20000 0.8364 0.8677 0.8927 1.0632 1.0897 1.1258
#> indirect 0.2556 0.0223 20000 0.1871 0.2007 0.2135 0.3012 0.3162 0.3333
#> direct   0.2139 0.0349 20000 0.1014 0.1250 0.1447 0.2820 0.3023 0.3274
```

```
#> total    0.4695 0.0347 20000 0.3518 0.3802 0.4015 0.5378 0.5596 0.5870
```

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         1.211963e-03 -5.078869e-04 -5.717885e-07 -7.981149e-06 -3.130096e-06
#> b         -5.078869e-04  1.013150e-03 -8.733192e-06  2.847346e-07  1.739895e-06
#> a         -5.717885e-07 -8.733192e-06  9.612835e-04  9.174093e-06  1.073577e-05
#> Y~~Y      -7.981149e-06  2.847346e-07  9.174093e-06  1.938911e-03  8.736174e-06
#> M~~M      -3.130096e-06  1.739895e-06  1.073577e-05  8.736174e-06  1.884649e-03
#> X~~X      -9.935711e-06 -1.804806e-06 -1.413360e-05  2.171353e-05  5.873122e-06
#> indirect -2.572356e-04  5.076327e-04  4.820133e-04  5.022214e-06  5.697626e-06
#> direct     1.211963e-03 -5.078869e-04 -5.717885e-07 -7.981149e-06 -3.130096e-06
#> total      9.547272e-04 -2.541615e-07  4.814415e-04 -2.958936e-06  2.567529e-06
#>                    X~~X      indirect        direct         total
#> cp        -9.935711e-06 -2.572356e-04  1.211963e-03  9.547272e-04
#> b         -1.804806e-06  5.076327e-04 -5.078869e-04 -2.541615e-07
#> a         -1.413360e-05  4.820133e-04 -5.717885e-07  4.814415e-04
#> Y~~Y       2.171353e-05  5.022214e-06 -7.981149e-06 -2.958936e-06
```

```
#> M~~M      5.873122e-06  5.697626e-06 -3.130096e-06  2.567529e-06
#> X~~X      2.141843e-03 -7.915883e-06 -9.935711e-06 -1.785159e-05
#> indirect -7.915883e-06  5.014295e-04 -2.572356e-04  2.441939e-04
#> direct   -9.935711e-06 -2.572356e-04  1.211963e-03  9.547272e-04
#> total    -1.785159e-05  2.441939e-04  9.547272e-04  1.198921e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1801 0.0290 20000 0.0869 0.1062 0.1233 0.2365 0.2537 0.2725
#> b         0.4662 0.0268 20000 0.3780 0.3973 0.4130 0.5173 0.5323 0.5539
#> a         0.4618 0.0249 20000 0.3790 0.3972 0.4118 0.5092 0.5238 0.5396
#> Y~~Y      0.6726 0.0243 20000 0.5907 0.6084 0.6232 0.7192 0.7339 0.7498
#> M~~M      0.7868 0.0229 20000 0.7088 0.7256 0.7407 0.8304 0.8422 0.8564
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2153 0.0173 20000 0.1611 0.1730 0.1822 0.2500 0.2611 0.2744
#> direct   0.1801 0.0290 20000 0.0869 0.1062 0.1233 0.2365 0.2537 0.2725
#> total    0.3954 0.0267 20000 0.3075 0.3247 0.3415 0.4465 0.4624 0.4812
```

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