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       0.2497 0.0348 20000 0.1372 0.1603 0.1823 0.3184 0.3395 0.3596
#> b        0.5070 0.0305 20000 0.4093 0.4288 0.4473 0.5668 0.5868 0.6087
#> a        0.4801 0.0330 20000 0.3742 0.3941 0.4145 0.5440 0.5642 0.5910
#> Y~~Y     0.9596 0.0430 20000 0.8170 0.8476 0.8759 1.0451 1.0695 1.1043
#> M~~M     1.0368 0.0469 20000 0.8839 0.9165 0.9447 1.1282 1.1562 1.1964
#> indirect 0.2434 0.0221 20000 0.1773 0.1885 0.2013 0.2874 0.3027 0.3222
#> direct   0.2497 0.0348 20000 0.1372 0.1603 0.1823 0.3184 0.3395 0.3596
```

```
#> total    0.4931 0.0357 20000 0.3776 0.4017 0.4228 0.5624 0.5869 0.6138
```

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        1.209606e-03 -4.408430e-04  4.775473e-06  1.929802e-05  5.788077e-06
#> b        -4.408430e-04  9.291077e-04 -1.310855e-05 -6.974041e-06  1.706374e-06
#> a         4.775473e-06 -1.310855e-05  1.081895e-03 -1.302103e-05  6.271371e-06
#> Y~~Y      1.929802e-05 -6.974041e-06 -1.302103e-05  1.846513e-03 -1.323416e-06
#> M~~M      5.788077e-06  1.706374e-06  6.271371e-06 -1.323416e-06  2.164440e-03
#> X~~X     -8.772973e-06 -7.131075e-06  1.201525e-07 -1.137341e-05  1.094618e-05
#> indirect -2.095311e-04  4.400689e-04  5.424657e-04 -1.008820e-05  4.061961e-06
#> direct    1.209606e-03 -4.408430e-04  4.775473e-06  1.929802e-05  5.788077e-06
#> total     1.000074e-03 -7.741561e-07  5.472412e-04  9.209825e-06  9.850038e-06
#>                    X~~X      indirect        direct         total
#> cp       -8.772973e-06 -2.095311e-04  1.209606e-03  1.000074e-03
#> b        -7.131075e-06  4.400689e-04 -4.408430e-04 -7.741561e-07
#> a         1.201525e-07  5.424657e-04  4.775473e-06  5.472412e-04
#> Y~~Y     -1.137341e-05 -1.008820e-05  1.929802e-05  9.209825e-06
```

```
#> M~~M      1.094618e-05  4.061961e-06  5.788077e-06  9.850038e-06
#> X~~X      1.828062e-03 -3.469652e-06 -8.772973e-06 -1.224262e-05
#> indirect -3.469652e-06  4.877427e-04 -2.095311e-04  2.782116e-04
#> direct   -8.772973e-06 -2.095311e-04  1.209606e-03  1.000074e-03
#> total    -1.224262e-05  2.782116e-04  1.000074e-03  1.278286e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2026 0.0278 20000 0.1077 0.1293 0.1469 0.2560 0.2742 0.2903
#> b         0.4707 0.0257 20000 0.3832 0.4026 0.4201 0.5206 0.5370 0.5533
#> a         0.4196 0.0260 20000 0.3304 0.3506 0.3672 0.4702 0.4841 0.4988
#> Y~~Y      0.6573 0.0243 20000 0.5795 0.5952 0.6084 0.7038 0.7171 0.7316
#> M~~M      0.8240 0.0218 20000 0.7512 0.7656 0.7789 0.8652 0.8771 0.8908
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1975 0.0166 20000 0.1481 0.1563 0.1660 0.2304 0.2416 0.2559
#> direct   0.2026 0.0278 20000 0.1077 0.1293 0.1469 0.2560 0.2742 0.2903
#> total    0.4001 0.0265 20000 0.3117 0.3303 0.3465 0.4505 0.4657 0.4788
```

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