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       0.2621 0.0357 20000 0.1408 0.1701 0.1917 0.3322 0.3541 0.3799
#> b        0.5507 0.0316 20000 0.4420 0.4693 0.4894 0.6127 0.6341 0.6577
#> a        0.4983 0.0323 20000 0.3931 0.4133 0.4345 0.5610 0.5795 0.5979
#> Y~~Y     0.9927 0.0445 20000 0.8436 0.8810 0.9052 1.0800 1.1071 1.1400
#> M~~M     0.9912 0.0442 20000 0.8502 0.8772 0.9046 1.0775 1.1066 1.1343
#> indirect 0.2744 0.0238 20000 0.2027 0.2165 0.2291 0.3223 0.3379 0.3585
#> direct   0.2621 0.0357 20000 0.1408 0.1701 0.1917 0.3322 0.3541 0.3799
```

```
#> total    0.5365 0.0367 20000 0.4102 0.4400 0.4642 0.6083 0.6306 0.6526
```

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        1.282328e-03 -4.966115e-04  4.100734e-06  1.011512e-06 -5.619394e-06
#> b        -4.966115e-04  9.894448e-04  5.391633e-06 -8.837383e-07  1.382480e-05
#> a         4.100734e-06  5.391633e-06  1.028833e-03 -1.374029e-05  8.315361e-07
#> Y~~Y      1.011512e-06 -8.837383e-07 -1.374029e-05  1.962676e-03  5.516099e-06
#> M~~M     -5.619394e-06  1.382480e-05  8.315361e-07  5.516099e-06  2.008116e-03
#> X~~X      3.428713e-06  7.906399e-06 -1.879073e-05 -1.791862e-05 -5.241757e-06
#> indirect -2.448601e-04  4.959451e-04  5.692256e-04 -7.647948e-06  7.543552e-06
#> direct    1.282328e-03 -4.966115e-04  4.100734e-06  1.011512e-06 -5.619394e-06
#> total     1.037467e-03 -6.663895e-07  5.733263e-04 -6.636436e-06  1.924158e-06
#>                    X~~X      indirect        direct         total
#> cp        3.428713e-06 -2.448601e-04  1.282328e-03  1.037467e-03
#> b         7.906399e-06  4.959451e-04 -4.966115e-04 -6.663895e-07
#> a        -1.879073e-05  5.692256e-04  4.100734e-06  5.733263e-04
#> Y~~Y     -1.791862e-05 -7.647948e-06  1.011512e-06 -6.636436e-06
```

```
#> M~~M     -5.241757e-06  7.543552e-06 -5.619394e-06  1.924158e-06
#> X~~X      1.818261e-03 -6.398778e-06  3.428713e-06 -2.970065e-06
#> indirect -6.398778e-06  5.615562e-04 -2.448601e-04  3.166960e-04
#> direct    3.428713e-06 -2.448601e-04  1.282328e-03  1.037467e-03
#> total    -2.970065e-06  3.166960e-04  1.037467e-03  1.354163e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2047 0.0277 20000 0.1112 0.1316 0.1497 0.2583 0.2754 0.3012
#> b         0.4874 0.0253 20000 0.4050 0.4190 0.4365 0.5358 0.5497 0.5703
#> a         0.4397 0.0254 20000 0.3562 0.3715 0.3883 0.4883 0.5020 0.5209
#> Y~~Y      0.6328 0.0242 20000 0.5516 0.5704 0.5849 0.6800 0.6948 0.7087
#> M~~M      0.8066 0.0223 20000 0.7287 0.7480 0.7616 0.8493 0.8620 0.8731
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2143 0.0170 20000 0.1609 0.1718 0.1813 0.2477 0.2592 0.2698
#> direct   0.2047 0.0277 20000 0.1112 0.1316 0.1497 0.2583 0.2754 0.3012
#> total    0.4190 0.0261 20000 0.3298 0.3501 0.3664 0.4685 0.4832 0.5010
```

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