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       0.2674 0.0363 20000 0.1529 0.1764 0.1962 0.3398 0.3618 0.3885
#> b        0.5035 0.0322 20000 0.4033 0.4213 0.4395 0.5663 0.5857 0.6066
#> a        0.5438 0.0306 20000 0.4437 0.4645 0.4833 0.6031 0.6233 0.6434
#> Y~~Y     0.9679 0.0433 20000 0.8229 0.8574 0.8820 1.0521 1.0786 1.1087
#> M~~M     0.9318 0.0416 20000 0.7946 0.8231 0.8499 1.0132 1.0363 1.0622
#> indirect 0.2738 0.0233 20000 0.2027 0.2178 0.2291 0.3205 0.3363 0.3570
#> direct   0.2674 0.0363 20000 0.1529 0.1764 0.1962 0.3398 0.3618 0.3885
```

```
#> total    0.5412 0.0352 20000 0.4244 0.4533 0.4725 0.6116 0.6329 0.6555
```

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        1.316887e-03 -5.711663e-04 -8.453003e-06 -5.623245e-06  2.272259e-05
#> b        -5.711663e-04  1.050083e-03  5.620743e-06  3.881284e-06 -5.058221e-06
#> a        -8.453003e-06  5.620743e-06  9.530227e-04 -3.022479e-06 -1.154390e-05
#> Y~~Y     -5.623245e-06  3.881284e-06 -3.022479e-06  1.876949e-03 -8.335555e-06
#> M~~M      2.272259e-05 -5.058221e-06 -1.154390e-05 -8.335555e-06  1.753945e-03
#> X~~X     -1.034413e-05  1.628603e-05  2.443789e-06 -2.480070e-05 -5.313333e-06
#> indirect -3.143203e-04  5.732246e-04  4.825455e-04  3.287904e-07 -8.850159e-06
#> direct    1.316887e-03 -5.711663e-04 -8.453003e-06 -5.623245e-06  2.272259e-05
#> total     1.002567e-03  2.058338e-06  4.740925e-04 -5.294455e-06  1.387243e-05
#>                    X~~X      indirect        direct         total
#> cp       -1.034413e-05 -3.143203e-04  1.316887e-03  1.002567e-03
#> b         1.628603e-05  5.732246e-04 -5.711663e-04  2.058338e-06
#> a         2.443789e-06  4.825455e-04 -8.453003e-06  4.740925e-04
#> Y~~Y     -2.480070e-05  3.287904e-07 -5.623245e-06 -5.294455e-06
```

```
#> M~~M     -5.313333e-06 -8.850159e-06  2.272259e-05  1.387243e-05
#> X~~X      1.905476e-03  9.992712e-06 -1.034413e-05 -3.514139e-07
#> indirect  9.992712e-06  5.551480e-04 -3.143203e-04  2.408277e-04
#> direct   -1.034413e-05 -3.143203e-04  1.316887e-03  1.002567e-03
#> total    -3.514139e-07  2.408277e-04  1.002567e-03  1.243394e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2162 0.0289 20000 0.1188 0.1407 0.1592 0.2725 0.2900 0.3111
#> b         0.4556 0.0270 20000 0.3669 0.3833 0.4011 0.5072 0.5230 0.5445
#> a         0.4858 0.0242 20000 0.4017 0.4214 0.4370 0.5321 0.5459 0.5606
#> Y~~Y      0.6500 0.0245 20000 0.5672 0.5874 0.6012 0.6966 0.7111 0.7274
#> M~~M      0.7640 0.0235 20000 0.6857 0.7020 0.7169 0.8091 0.8224 0.8387
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2213 0.0176 20000 0.1649 0.1770 0.1870 0.2558 0.2665 0.2796
#> direct    0.2162 0.0289 20000 0.1188 0.1407 0.1592 0.2725 0.2900 0.3111
#> total     0.4375 0.0256 20000 0.3516 0.3689 0.3859 0.4867 0.5008 0.5195
```

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