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       0.3031 0.0359 20000 0.1823 0.2102 0.2323 0.3725 0.3952 0.4219
#> b        0.4101 0.0312 20000 0.3061 0.3305 0.3483 0.4718 0.4895 0.5090
#> a        0.5284 0.0321 20000 0.4229 0.4462 0.4650 0.5913 0.6110 0.6366
#> Y~~Y     1.0155 0.0456 20000 0.8650 0.8990 0.9265 1.1041 1.1318 1.1666
#> M~~M     1.0475 0.0468 20000 0.8963 0.9283 0.9553 1.1390 1.1687 1.1997
#> indirect 0.2167 0.0212 20000 0.1531 0.1651 0.1768 0.2591 0.2727 0.2934
#> direct   0.3031 0.0359 20000 0.1823 0.2102 0.2323 0.3725 0.3952 0.4219
```

```
#> total    0.5197 0.0346 20000 0.4027 0.4312 0.4517 0.5866 0.6088 0.6336
```

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        1.294296e-03 -5.108215e-04 -3.293125e-06 3.794810e-06 -6.448449e-06
#> b        -5.108215e-04  9.743550e-04 -4.157590e-06 1.654754e-05  1.353221e-06
#> a        -3.293125e-06 -4.157590e-06  1.046927e-03 5.741547e-06 -1.505658e-06
#> Y~~Y      3.794810e-06  1.654754e-05  5.741547e-06 2.092196e-03  4.371387e-06
#> M~~M     -6.448449e-06  1.353221e-06 -1.505658e-06 4.371387e-06  2.150696e-03
#> X~~X      3.867328e-06 -1.682503e-05 -2.155637e-06 1.834617e-05 -1.558174e-05
#> indirect -2.711636e-04  5.128746e-04  4.274770e-04 1.116783e-05  4.851244e-07
#> direct    1.294296e-03 -5.108215e-04 -3.293125e-06 3.794810e-06 -6.448449e-06
#> total     1.023133e-03  2.053048e-06  4.241839e-04 1.496264e-05 -5.963325e-06
#>                    X~~X      indirect        direct         total
#> cp        3.867328e-06 -2.711636e-04  1.294296e-03  1.023133e-03
#> b        -1.682503e-05  5.128746e-04 -5.108215e-04  2.053048e-06
#> a        -2.155637e-06  4.274770e-04 -3.293125e-06  4.241839e-04
#> Y~~Y      1.834617e-05  1.116783e-05  3.794810e-06  1.496264e-05
```

```
#> M~~M     -1.558174e-05  4.851244e-07 -6.448449e-06 -5.963325e-06
#> X~~X      2.054009e-03 -9.550923e-06  3.867328e-06 -5.683595e-06
#> indirect -9.550923e-06  4.473118e-04 -2.711636e-04  1.761482e-04
#> direct    3.867328e-06 -2.711636e-04  1.294296e-03  1.023133e-03
#> total    -5.683595e-06  1.761482e-04  1.023133e-03  1.199281e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2517 0.0292 20000 0.1575 0.1764 0.1939 0.3082 0.3260 0.3425
#> b         0.3907 0.0279 20000 0.2967 0.3181 0.3356 0.4450 0.4624 0.4813
#> a         0.4606 0.0250 20000 0.3752 0.3951 0.4103 0.5088 0.5240 0.5394
#> Y~~Y      0.6934 0.0242 20000 0.6117 0.6295 0.6450 0.7393 0.7539 0.7653
#> M~~M      0.7878 0.0230 20000 0.7091 0.7254 0.7412 0.8316 0.8439 0.8593
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1800 0.0164 20000 0.1295 0.1393 0.1485 0.2126 0.2245 0.2367
#> direct   0.2517 0.0292 20000 0.1575 0.1764 0.1939 0.3082 0.3260 0.3425
#> total    0.4317 0.0258 20000 0.3460 0.3640 0.3799 0.4813 0.4956 0.5113
```

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