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       0.3061 0.0361 20000 0.1906 0.2141 0.2347 0.3759 0.3971 0.4288
#> b        0.4396 0.0322 20000 0.3269 0.3556 0.3765 0.5029 0.5222 0.5433
#> a        0.5004 0.0317 20000 0.3959 0.4178 0.4380 0.5625 0.5813 0.5976
#> Y~~Y     1.0215 0.0456 20000 0.8734 0.9064 0.9330 1.1100 1.1377 1.1665
#> M~~M     0.9801 0.0433 20000 0.8421 0.8670 0.8947 1.0653 1.0899 1.1183
#> indirect 0.2200 0.0212 20000 0.1527 0.1678 0.1791 0.2625 0.2769 0.2968
#> direct   0.3061 0.0361 20000 0.1906 0.2141 0.2347 0.3759 0.3971 0.4288
```

```
#> total    0.5261 0.0354 20000 0.4110 0.4350 0.4560 0.5944 0.6171 0.6458
```

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        1.326177e-03 -5.189254e-04 -9.889101e-06 -2.160992e-05 -7.472335e-07
#> b        -5.189254e-04  1.043626e-03 -6.580194e-06  4.058581e-06  1.827930e-05
#> a        -9.889101e-06 -6.580194e-06  1.022687e-03 -3.852040e-06 -1.789365e-05
#> Y~~Y     -2.160992e-05  4.058581e-06 -3.852040e-06  2.093532e-03  4.606948e-06
#> M~~M     -7.472335e-07  1.827930e-05 -1.789365e-05  4.606948e-06  1.926720e-03
#> X~~X      6.806100e-06 -2.709275e-06 -2.024472e-06  2.117468e-05  2.364214e-05
#> indirect -2.638417e-04  5.191286e-04  4.463285e-04 -6.288067e-07  1.663305e-06
#> direct    1.326177e-03 -5.189254e-04 -9.889101e-06 -2.160992e-05 -7.472335e-07
#> total     1.062335e-03  2.031859e-07  4.364394e-04 -2.223873e-05  9.160715e-07
#>                    X~~X      indirect        direct         total
#> cp        6.806100e-06 -2.638417e-04  1.326177e-03  1.062335e-03
#> b        -2.709275e-06  5.191286e-04 -5.189254e-04  2.031859e-07
#> a        -2.024472e-06  4.463285e-04 -9.889101e-06  4.364394e-04
#> Y~~Y      2.117468e-05 -6.288067e-07 -2.160992e-05 -2.223873e-05
```

```
#> M~~M      2.364214e-05  1.663305e-06 -7.472335e-07  9.160715e-07
#> X~~X      1.866497e-03 -2.624686e-06  6.806100e-06  4.181415e-06
#> indirect -2.624686e-06  4.569404e-04 -2.638417e-04  1.930987e-04
#> direct    6.806100e-06 -2.638417e-04  1.326177e-03  1.062335e-03
#> total     4.181415e-06  1.930987e-04  1.062335e-03  1.255434e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2472 0.0288 20000 0.1508 0.1717 0.1896 0.3039 0.3216 0.3377
#> b         0.3997 0.0276 20000 0.3092 0.3281 0.3450 0.4533 0.4701 0.4902
#> a         0.4445 0.0255 20000 0.3565 0.3761 0.3925 0.4931 0.5084 0.5270
#> Y~~Y      0.6913 0.0244 20000 0.6102 0.6264 0.6419 0.7377 0.7520 0.7677
#> M~~M      0.8025 0.0226 20000 0.7223 0.7416 0.7568 0.8459 0.8586 0.8729
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1776 0.0161 20000 0.1264 0.1374 0.1465 0.2098 0.2198 0.2302
#> direct   0.2472 0.0288 20000 0.1508 0.1717 0.1896 0.3039 0.3216 0.3377
#> total    0.4249 0.0260 20000 0.3356 0.3564 0.3729 0.4750 0.4907 0.5063
```

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