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.

```r
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`.

```r
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.

```r
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       0.2455 0.0345 20000 0.1324 0.1569 0.1790 0.3137 0.3343 0.3614
#> b        0.5157 0.0305 20000 0.4150 0.4373 0.4561 0.5748 0.5946 0.6178
#> a        0.4884 0.0328 20000 0.3804 0.4036 0.4242 0.5535 0.5731 0.5933
#> Y~~Y     0.9780 0.0439 20000 0.8349 0.8641 0.8917 1.0649 1.0891 1.1242
#> M~~M     1.0590 0.0473 20000 0.9043 0.9386 0.9672 1.1521 1.1836 1.2264
#> indirect 0.2519 0.0226 20000 0.1810 0.1963 0.2094 0.2970 0.3136 0.3317
#> direct   0.2455 0.0345 20000 0.1324 0.1569 0.1790 0.3137 0.3343 0.3614
```

```
#> total    0.4974 0.0355 20000 0.3816 0.4078 0.4269 0.5667 0.5887 0.6133
```

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        1.206727e-03 -4.493957e-04 -4.185918e-06 -9.919482e-06 -6.036096e-06
#> b        -4.493957e-04  9.317595e-04 -3.843951e-06  2.639655e-06 -6.203048e-06
#> a        -4.185918e-06 -3.843951e-06  1.054117e-03 -1.056075e-05  2.139470e-06
#> Y~~Y     -9.919482e-06  2.639655e-06 -1.056075e-05  1.943714e-03  2.077673e-07
#> M~~M     -6.036096e-06 -6.203048e-06  2.139470e-06  2.077673e-07  2.220887e-03
#> X~~X     -7.877950e-06 -4.155826e-06  1.594918e-05 -3.581397e-05  6.733863e-06
#> indirect -2.210930e-04  4.527094e-04  5.414213e-04 -4.704140e-06 -2.089838e-06
#> direct    1.206727e-03 -4.493957e-04 -4.185918e-06 -9.919482e-06 -6.036096e-06
#> total     9.856338e-04  3.313761e-06  5.372354e-04 -1.462362e-05 -8.125934e-06
#>                   X~~X      indirect        direct         total
#> cp       -7.877950e-06 -2.210930e-04  1.206727e-03  9.856338e-04
#> b        -4.155826e-06  4.527094e-04 -4.493957e-04  3.313761e-06
#> a         1.594918e-05  5.414213e-04 -4.185918e-06  5.372354e-04
#> Y~~Y     -3.581397e-05 -4.704140e-06 -9.919482e-06 -1.462362e-05
```

```
#> M~~M      6.733863e-06 -2.089838e-06 -6.036096e-06 -8.125934e-06
#> X~~X      1.974357e-03  6.297029e-06 -7.877950e-06 -1.580921e-06
#> indirect  6.297029e-06  5.009758e-04 -2.210930e-04  2.798827e-04
#> direct   -7.877950e-06 -2.210930e-04  1.206727e-03  9.856338e-04
#> total    -1.580921e-06  2.798827e-04  9.856338e-04  1.265517e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1996 0.0279 20000 0.1063 0.1283 0.1443 0.2536 0.2716 0.2902
#> b        0.4785 0.0256 20000 0.3912 0.4115 0.4278 0.5266 0.5442 0.5614
#> a        0.4280 0.0258 20000 0.3382 0.3579 0.3764 0.4772 0.4918 0.5092
#> Y~~Y     0.6494 0.0244 20000 0.5655 0.5857 0.6009 0.6964 0.7114 0.7300
#> M~~M     0.8168 0.0220 20000 0.7407 0.7582 0.7723 0.8583 0.8719 0.8856
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2048 0.0168 20000 0.1488 0.1625 0.1720 0.2374 0.2487 0.2606
#> direct   0.1996 0.0279 20000 0.1063 0.1283 0.1443 0.2536 0.2716 0.2902
#> total    0.4044 0.0265 20000 0.3095 0.3329 0.3506 0.4548 0.4697 0.4857
```

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