semmcci: Monte Carlo Confidence Intervals

Ivan Jacob Agaloos Pesigan

Installation

You can install the CRAN release of `semmcci` with:

```r
install.packages("semmcci")
```

You can install the development version of `semmcci` from GitHub with:

```r
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       0.2356 0.0347 20000 0.1167 0.1453 0.1669 0.3026 0.3271 0.3507
#> b        0.5134 0.0298 20000 0.4123 0.4368 0.4554 0.5718 0.5895 0.6141
#> a        0.5135 0.0331 20000 0.4025 0.4283 0.4490 0.5787 0.5981 0.6206
#> Y~~Y     0.9453 0.0423 20000 0.8047 0.8377 0.8625 1.0287 1.0556 1.0892
#> M~~M     1.0645 0.0475 20000 0.9138 0.9429 0.9721 1.1572 1.1859 1.2185
#> indirect 0.2636 0.0229 20000 0.1935 0.2079 0.2205 0.3097 0.3248 0.3409
#> direct   0.2356 0.0347 20000 0.1167 0.1453 0.1669 0.3026 0.3271 0.3507
```

```
#> total    0.4993 0.0358 20000 0.3779 0.4057 0.4280 0.5687 0.5905 0.6216
```

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        1.202762e-03 -4.463003e-04 -9.331241e-06  1.513875e-05  7.443480e-06
#> b        -4.463003e-04  8.779892e-04 -2.315372e-06  2.122820e-07 -3.664059e-06
#> a        -9.331241e-06 -2.315372e-06  1.100306e-03 -1.379035e-06  6.713145e-06
#> Y~~Y      1.513875e-05  2.122820e-07 -1.379035e-06  1.763511e-03 -1.007228e-05
#> M~~M      7.443480e-06 -3.664059e-06  6.713145e-06 -1.007228e-05  2.275514e-03
#> X~~X      3.448364e-06 -3.891793e-06 -1.866495e-06  1.434461e-05  8.555967e-06
#> indirect -2.336240e-04  4.494125e-04  5.635694e-04 -6.498590e-07  1.461429e-06
#> direct    1.202762e-03 -4.463003e-04 -9.331241e-06  1.513875e-05  7.443480e-06
#> total     9.691381e-04  3.112146e-06  5.542381e-04  1.448889e-05  8.904909e-06
#>                   X~~X      indirect        direct        total
#> cp        3.448364e-06 -2.336240e-04  1.202762e-03 9.691381e-04
#> b        -3.891793e-06  4.494125e-04 -4.463003e-04 3.112146e-06
#> a        -1.866495e-06  5.635694e-04 -9.331241e-06 5.542381e-04
#> Y~~Y      1.434461e-05 -6.498590e-07  1.513875e-05 1.448889e-05
```

```
#> M~~M      8.555967e-06  1.461429e-06  7.443480e-06 8.904909e-06
#> X~~X      1.943428e-03 -2.559387e-06  3.448364e-06 8.889768e-07
#> indirect -2.559387e-06  5.208815e-04 -2.336240e-04 2.872575e-04
#> direct    3.448364e-06 -2.336240e-04  1.202762e-03 9.691381e-04
#> total     8.889768e-07  2.872575e-04  9.691381e-04 1.256396e-03
```

```
MCStd(unstd)
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1923 0.0279 20000 0.1018 0.1207 0.1372 0.2475 0.2643 0.2847
#> b        0.4870 0.0254 20000 0.4051 0.4210 0.4357 0.5357 0.5502 0.5671
#> a        0.4417 0.0256 20000 0.3554 0.3735 0.3906 0.4906 0.5055 0.5238
#> Y~~Y     0.6431 0.0242 20000 0.5615 0.5811 0.5944 0.6891 0.7030 0.7183
#> M~~M     0.8049 0.0225 20000 0.7256 0.7445 0.7594 0.8474 0.8605 0.8737
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2151 0.0171 20000 0.1605 0.1722 0.1819 0.2488 0.2598 0.2737
#> direct   0.1923 0.0279 20000 0.1018 0.1207 0.1372 0.2475 0.2643 0.2847
#> total    0.4073 0.0263 20000 0.3210 0.3387 0.3555 0.4580 0.4740 0.4902
```

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