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:

```
if (!require("remotes")) 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       0.2616 0.0349 20000 0.1462 0.1723 0.1930 0.3298 0.3514 0.3776
#> b        0.4702 0.0311 20000 0.3671 0.3900 0.4084 0.5304 0.5489 0.5708
#> a        0.5261 0.0314 20000 0.4279 0.4468 0.4646 0.5873 0.6080 0.6270
#> Y~~Y     1.0120 0.0450 20000 0.8678 0.8969 0.9239 1.0996 1.1272 1.1619
#> M~~M     1.0399 0.0465 20000 0.8887 0.9209 0.9495 1.1313 1.1588 1.1914
#> indirect 0.2474 0.0221 20000 0.1796 0.1919 0.2056 0.2916 0.3074 0.3292
#> direct   0.2616 0.0349 20000 0.1462 0.1723 0.1930 0.3298 0.3514 0.3776
```

```
#> total    0.5090 0.0343 20000 0.3952 0.4191 0.4409 0.5760 0.5974 0.6243
```

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        1.224253e-03 -5.171850e-04 -1.600401e-06  1.466005e-05 -6.852454e-06
#> b        -5.171850e-04  9.872994e-04 -1.375653e-05  8.762612e-06 -8.568222e-06
#> a        -1.600401e-06 -1.375653e-05  9.767064e-04 -1.576554e-06  1.921970e-06
#> Y~~Y      1.466005e-05  8.762612e-06 -1.576554e-06  2.099875e-03 -1.769628e-05
#> M~~M     -6.852454e-06 -8.568222e-06  1.921970e-06 -1.769628e-05  2.145974e-03
#> X~~X      1.023900e-05  1.061408e-05  6.288054e-06 -3.642048e-06  2.265361e-06
#> indirect -2.725989e-04  5.126038e-04  4.522830e-04  3.968301e-06 -2.709087e-06
#> direct    1.224253e-03 -5.171850e-04 -1.600401e-06  1.466005e-05 -6.852454e-06
#> total     9.516541e-04 -4.581237e-06  4.506826e-04  1.862835e-05 -9.561541e-06
#>                    X~~X      indirect        direct         total
#> cp        1.023900e-05 -2.725989e-04  1.224253e-03  9.516541e-04
#> b         1.061408e-05  5.126038e-04 -5.171850e-04 -4.581237e-06
#> a         6.288054e-06  4.522830e-04 -1.600401e-06  4.506826e-04
#> Y~~Y     -3.642048e-06  3.968301e-06  1.466005e-05  1.862835e-05
```

```
#> M~~M      2.265361e-06 -2.709087e-06 -6.852454e-06 -9.561541e-06
#> X~~X      2.261782e-03  8.583948e-06  1.023900e-05  1.882295e-05
#> indirect  8.583948e-06  4.832290e-04 -2.725989e-04  2.106301e-04
#> direct    1.023900e-05 -2.725989e-04  1.224253e-03  9.516541e-04
#> total     1.882295e-05  2.106301e-04  9.516541e-04  1.162284e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2188 0.0289 20000 0.1226 0.1441 0.1622 0.2752 0.2930 0.3138
#> b         0.4409 0.0272 20000 0.3509 0.3703 0.3866 0.4934 0.5097 0.5266
#> a         0.4693 0.0246 20000 0.3889 0.4056 0.4200 0.5165 0.5312 0.5455
#> Y~~Y      0.6672 0.0243 20000 0.5815 0.6049 0.6188 0.7138 0.7299 0.7432
#> M~~M      0.7798 0.0231 20000 0.7025 0.7178 0.7332 0.8236 0.8355 0.8488
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2069 0.0170 20000 0.1560 0.1649 0.1742 0.2409 0.2517 0.2649
#> direct    0.2188 0.0289 20000 0.1226 0.1441 0.1622 0.2752 0.2930 0.3138
#> total     0.4257 0.0259 20000 0.3397 0.3578 0.3737 0.4760 0.4915 0.5041
```

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. (2023). *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