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.

```r
library(semmcci)
library(lavaan)
```

Data

```r
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       0.2273 0.0375 20000 0.1025 0.1319 0.1531 0.2999 0.3247 0.3513
#> b        0.5170 0.0333 20000 0.4106 0.4319 0.4517 0.5818 0.6033 0.6240
#> a        0.5128 0.0318 20000 0.4078 0.4306 0.4509 0.5752 0.5963 0.6161
#> Y~~Y     1.1389 0.0512 20000 0.9688 1.0071 1.0381 1.2402 1.2721 1.3085
#> M~~M     1.0330 0.0462 20000 0.8826 0.9149 0.9428 1.1245 1.1522 1.1810
#> indirect 0.2651 0.0238 20000 0.1944 0.2077 0.2203 0.3136 0.3297 0.3452
#> direct   0.2273 0.0375 20000 0.1025 0.1319 0.1531 0.2999 0.3247 0.3513
```

```
#> total    0.4924 0.0375 20000 0.3679 0.3954 0.4189 0.5660 0.5905 0.6175
```

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        1.410174e-03 -5.561872e-04  2.174566e-05 -1.368083e-05  3.716032e-06
#> b        -5.561872e-04  1.081026e-03  6.189349e-07  1.178304e-05 -8.164662e-06
#> a         2.174566e-05  6.189349e-07  1.025234e-03  8.423690e-06 -6.051489e-06
#> Y~~Y     -1.368083e-05  1.178304e-05  8.423690e-06  2.628535e-03 -1.494043e-05
#> M~~M      3.716032e-06 -8.164662e-06 -6.051489e-06 -1.494043e-05  2.145081e-03
#> X~~X     -1.420581e-05  1.796383e-05  8.923395e-06  2.492502e-05  3.430091e-06
#> indirect -2.744924e-04  5.546077e-04  5.303342e-04  1.065871e-05 -7.202160e-06
#> direct    1.410174e-03 -5.561872e-04  2.174566e-05 -1.368083e-05  3.716032e-06
#> total     1.135681e-03 -1.579574e-06  5.520798e-04 -3.022120e-06 -3.486128e-06
#>                   X~~X      indirect        direct         total
#> cp       -1.420581e-05 -2.744924e-04  1.410174e-03  1.135681e-03
#> b         1.796383e-05  5.546077e-04 -5.561872e-04 -1.579574e-06
#> a         8.923395e-06  5.303342e-04  2.174566e-05  5.520798e-04
#> Y~~Y      2.492502e-05  1.065871e-05 -1.368083e-05 -3.022120e-06
```

```
#> M~~M      3.430091e-06 -7.202160e-06  3.716032e-06 -3.486128e-06
#> X~~X      2.094345e-03  1.392806e-05 -1.420581e-05 -2.777491e-07
#> indirect  1.392806e-05  5.596266e-04 -2.744924e-04  2.851342e-04
#> direct   -1.420581e-05 -2.744924e-04  1.410174e-03  1.135681e-03
#> total    -2.777491e-07  2.851342e-04  1.135681e-03  1.420815e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1779 0.0291 20000 0.0826 0.1024 0.1206 0.2347 0.2511 0.2681
#> b        0.4574 0.0267 20000 0.3707 0.3874 0.4048 0.5087 0.5263 0.5404
#> a        0.4538 0.0253 20000 0.3640 0.3877 0.4032 0.5026 0.5168 0.5346
#> Y~~Y     0.6853 0.0243 20000 0.6046 0.6226 0.6369 0.7324 0.7482 0.7646
#> M~~M     0.7941 0.0229 20000 0.7142 0.7329 0.7474 0.8374 0.8497 0.8675
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2076 0.0172 20000 0.1558 0.1647 0.1739 0.2413 0.2533 0.2668
#> direct   0.1779 0.0291 20000 0.0826 0.1024 0.1206 0.2347 0.2511 0.2681
#> total    0.3855 0.0272 20000 0.2889 0.3120 0.3308 0.4373 0.4518 0.4680
```

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