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       0.2930 0.0358 20000 0.1764 0.1984 0.2217 0.3636 0.3841 0.4043
#> b        0.4923 0.0317 20000 0.3883 0.4120 0.4310 0.5550 0.5739 0.5947
#> a        0.4812 0.0322 20000 0.3769 0.3978 0.4178 0.5438 0.5632 0.5867
#> Y~~Y     1.0220 0.0456 20000 0.8747 0.9067 0.9328 1.1123 1.1417 1.1771
#> M~~M     1.0145 0.0456 20000 0.8635 0.8984 0.9261 1.1045 1.1311 1.1692
#> indirect 0.2369 0.0219 20000 0.1714 0.1828 0.1954 0.2819 0.2968 0.3126
#> direct   0.2930 0.0358 20000 0.1764 0.1984 0.2217 0.3636 0.3841 0.4043
```

```
#> total    0.5299 0.0362 20000 0.4105 0.4353 0.4589 0.6008 0.6215 0.6469
```

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         1.282172e-03 -4.940890e-04 5.695754e-06  1.437772e-05 -9.745274e-07
#> b         -4.940890e-04  1.015552e-03 7.729328e-06 -2.143874e-05 -1.196292e-05
#> a          5.695754e-06  7.729328e-06 1.031701e-03  8.644707e-06  9.381236e-07
#> Y~~Y       1.437772e-05 -2.143874e-05 8.644707e-06  2.103471e-03  2.710168e-06
#> M~~M      -9.745274e-07 -1.196292e-05 9.381236e-07  2.710168e-06  2.052286e-03
#> X~~X       8.923669e-06 -1.417472e-05 1.092483e-06 -1.633782e-05 -4.115302e-06
#> indirect  -2.345426e-04  4.919979e-04 5.111290e-04 -5.957703e-06 -5.271920e-06
#> direct     1.282172e-03 -4.940890e-04 5.695754e-06  1.437772e-05 -9.745274e-07
#> total      1.047630e-03 -2.091086e-06 5.168247e-04  8.420014e-06 -6.246447e-06
#>                    X~~X      indirect        direct         total
#> cp         8.923669e-06 -2.345426e-04  1.282172e-03  1.047630e-03
#> b         -1.417472e-05  4.919979e-04 -4.940890e-04 -2.091086e-06
#> a          1.092483e-06  5.111290e-04  5.695754e-06  5.168247e-04
#> Y~~Y      -1.633782e-05 -5.957703e-06  1.437772e-05  8.420014e-06
```

```
#> M~~M     -4.115302e-06 -5.271920e-06 -9.745274e-07 -6.246447e-06
#> X~~X      1.902385e-03 -6.050023e-06  8.923669e-06  2.873646e-06
#> indirect -6.050023e-06  4.889341e-04 -2.345426e-04  2.543916e-04
#> direct    8.923669e-06 -2.345426e-04  1.282172e-03  1.047630e-03
#> total     2.873646e-06  2.543916e-04  1.047630e-03  1.302021e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2333 0.0280 20000 0.1442 0.1616 0.1783 0.2882 0.3062 0.3227
#> b         0.4416 0.0264 20000 0.3524 0.3718 0.3894 0.4927 0.5073 0.5291
#> a         0.4271 0.0258 20000 0.3401 0.3587 0.3759 0.4760 0.4918 0.5112
#> Y~~Y      0.6626 0.0244 20000 0.5854 0.5989 0.6137 0.7093 0.7226 0.7376
#> M~~M      0.8176 0.0220 20000 0.7387 0.7581 0.7734 0.8587 0.8713 0.8843
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1886 0.0163 20000 0.1379 0.1480 0.1567 0.2207 0.2307 0.2431
#> direct   0.2333 0.0280 20000 0.1442 0.1616 0.1783 0.2882 0.3062 0.3227
#> total    0.4219 0.0260 20000 0.3318 0.3534 0.3701 0.4717 0.4869 0.5016
```

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