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       0.2722 0.0357 20000 0.1590 0.1807 0.2018 0.3421 0.3628 0.3924
#> b        0.4852 0.0308 20000 0.3833 0.4061 0.4253 0.5459 0.5640 0.5833
#> a        0.5227 0.0329 20000 0.4121 0.4381 0.4583 0.5865 0.6070 0.6372
#> Y~~Y     0.9729 0.0436 20000 0.8345 0.8616 0.8875 1.0576 1.0846 1.1216
#> M~~M     1.0250 0.0458 20000 0.8801 0.9076 0.9359 1.1151 1.1437 1.1722
#> indirect 0.2536 0.0227 20000 0.1832 0.1971 0.2103 0.2992 0.3138 0.3300
#> direct   0.2722 0.0357 20000 0.1590 0.1807 0.2018 0.3421 0.3628 0.3924
```

```
#> total    0.5258 0.0358 20000 0.4097 0.4332 0.4558 0.5969 0.6184 0.6444
```

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         1.280375e-03 -4.981145e-04  8.164173e-06 -1.927306e-05 -1.084977e-05
#> b         -4.981145e-04  9.521003e-04 -5.375047e-06 -7.285519e-07  6.663628e-06
#> a          8.164173e-06 -5.375047e-06  1.096181e-03 -6.060849e-06  1.328394e-05
#> Y~~Y      -1.927306e-05 -7.285519e-07 -6.060849e-06  1.875999e-03  2.585346e-05
#> M~~M      -1.084977e-05  6.663628e-06  1.328394e-05  2.585346e-05  2.114149e-03
#> X~~X      -9.410129e-06 -5.201242e-06 -4.255012e-06 -5.508040e-07 -4.480282e-07
#> indirect -2.565122e-04  4.950745e-04  5.298786e-04 -3.236601e-06  9.204203e-06
#> direct     1.280375e-03 -4.981145e-04  8.164173e-06 -1.927306e-05 -1.084977e-05
#> total      1.023863e-03 -3.040043e-06  5.380428e-04 -2.250966e-05 -1.645563e-06
#>                     X~~X      indirect        direct         total
#> cp       -9.410129e-06 -2.565122e-04  1.280375e-03  1.023863e-03
#> b        -5.201242e-06  4.950745e-04 -4.981145e-04 -3.040043e-06
#> a        -4.255012e-06  5.298786e-04  8.164173e-06  5.380428e-04
#> Y~~Y     -5.508040e-07 -3.236601e-06 -1.927306e-05 -2.250966e-05
```

```
#> M~~M     -4.480282e-07  9.204203e-06 -1.084977e-05 -1.645563e-06
#> X~~X      1.779684e-03 -4.909599e-06 -9.410129e-06 -1.431973e-05
#> indirect -4.909599e-06  5.173171e-04 -2.565122e-04  2.608049e-04
#> direct   -9.410129e-06 -2.565122e-04  1.280375e-03  1.023863e-03
#> total    -1.431973e-05  2.608049e-04  1.023863e-03  1.284668e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2182 0.0283 20000 0.1277 0.1454 0.1617 0.2729 0.2905 0.3071
#> b         0.4525 0.0265 20000 0.3614 0.3834 0.3999 0.5038 0.5193 0.5384
#> a         0.4493 0.0253 20000 0.3605 0.3828 0.3995 0.4981 0.5126 0.5286
#> Y~~Y      0.6589 0.0243 20000 0.5763 0.5948 0.6106 0.7053 0.7211 0.7348
#> M~~M      0.7981 0.0227 20000 0.7206 0.7372 0.7519 0.8404 0.8535 0.8700
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2033 0.0168 20000 0.1492 0.1612 0.1709 0.2373 0.2477 0.2643
#> direct   0.2182 0.0283 20000 0.1277 0.1454 0.1617 0.2729 0.2905 0.3071
#> total    0.4215 0.0260 20000 0.3298 0.3519 0.3685 0.4713 0.4859 0.5034
```

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