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       0.1949 0.0345 20000 0.0842 0.1047 0.1271 0.2628 0.2828 0.3116
#> b        0.5782 0.0316 20000 0.4750 0.4979 0.5164 0.6395 0.6604 0.6845
#> a        0.4936 0.0314 20000 0.3876 0.4134 0.4321 0.5560 0.5754 0.5929
#> Y~~Y     0.9795 0.0437 20000 0.8304 0.8654 0.8941 1.0660 1.0903 1.1214
#> M~~M     0.9929 0.0444 20000 0.8429 0.8782 0.9057 1.0793 1.1061 1.1410
#> indirect 0.2854 0.0241 20000 0.2086 0.2273 0.2395 0.3340 0.3510 0.3699
#> direct   0.1949 0.0345 20000 0.0842 0.1047 0.1271 0.2628 0.2828 0.3116
```

```
#> total    0.4803 0.0359 20000 0.3599 0.3884 0.4100 0.5502 0.5731 0.5980
```

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        1.176566e-03 -4.720603e-04 -9.479152e-06  8.091058e-06 -9.829956e-07
#> b        -4.720603e-04  9.778836e-04 -5.137988e-07 -3.301515e-06 -4.097661e-06
#> a        -9.479152e-06 -5.137988e-07  9.673183e-04 -2.024066e-05  5.828956e-07
#> Y~~Y      8.091058e-06 -3.301515e-06 -2.024066e-05  1.931472e-03 -1.237901e-07
#> M~~M     -9.829956e-07 -4.097661e-06  5.828956e-07 -1.237901e-07  1.961040e-03
#> X~~X      9.735640e-06  5.026121e-06  9.486207e-06  1.165315e-06  1.896515e-06
#> indirect -2.385940e-04  4.826223e-04  5.597246e-04 -1.389875e-05 -1.368048e-06
#> direct    1.176566e-03 -4.720603e-04 -9.479152e-06  8.091058e-06 -9.829956e-07
#> total     9.379721e-04  1.056196e-05  5.502455e-04 -5.807691e-06 -2.351044e-06
#>                    X~~X      indirect        direct         total
#> cp       9.735640e-06 -2.385940e-04  1.176566e-03  9.379721e-04
#> b        5.026121e-06  4.826223e-04 -4.720603e-04  1.056196e-05
#> a        9.486207e-06  5.597246e-04 -9.479152e-06  5.502455e-04
#> Y~~Y     1.165315e-06 -1.389875e-05  8.091058e-06 -5.807691e-06
```

```
#> M~~M     1.896515e-06 -1.368048e-06 -9.829956e-07 -2.351044e-06
#> X~~X     2.094082e-03  7.661891e-06  9.735640e-06  1.739753e-05
#> indirect 7.661891e-06  5.632989e-04 -2.385940e-04  3.247049e-04
#> direct   9.735640e-06 -2.385940e-04  1.176566e-03  9.379721e-04
#> total    1.739753e-05  3.247049e-04  9.379721e-04  1.262677e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1582 0.0276 20000 0.0683 0.0864 0.1036 0.2123 0.2287 0.2472
#> b         0.5180 0.0249 20000 0.4297 0.4513 0.4677 0.5656 0.5804 0.5987
#> a         0.4472 0.0253 20000 0.3604 0.3814 0.3972 0.4962 0.5110 0.5280
#> Y~~Y      0.6334 0.0242 20000 0.5541 0.5698 0.5847 0.6798 0.6936 0.7108
#> M~~M      0.8000 0.0226 20000 0.7213 0.7388 0.7538 0.8422 0.8545 0.8701
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect  0.2316 0.0177 20000 0.1766 0.1885 0.1978 0.2667 0.2775 0.2907
#> direct    0.1582 0.0276 20000 0.0683 0.0864 0.1036 0.2123 0.2287 0.2472
#> total     0.3898 0.0266 20000 0.2990 0.3200 0.3366 0.4409 0.4551 0.4692
```

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