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       0.2143 0.0362 20000 0.1009 0.1230 0.1444 0.2850 0.3083 0.3326
#> b        0.5001 0.0314 20000 0.3917 0.4194 0.4388 0.5621 0.5810 0.6018
#> a        0.5223 0.0326 20000 0.4170 0.4400 0.4593 0.5873 0.6072 0.6336
#> Y~~Y     1.0022 0.0450 20000 0.8508 0.8878 0.9139 1.0913 1.1192 1.1426
#> M~~M     1.0222 0.0457 20000 0.8784 0.9074 0.9317 1.1110 1.1398 1.1736
#> indirect 0.2612 0.0233 20000 0.1907 0.2044 0.2178 0.3096 0.3243 0.3413
#> direct   0.2143 0.0362 20000 0.1009 0.1230 0.1444 0.2850 0.3083 0.3326
```

```
#> total    0.4755 0.0361 20000 0.3642 0.3830 0.4051 0.5467 0.5679 0.5938
```

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        1.316432e-03 -5.052180e-04  6.346709e-06 -5.736215e-06 -2.211335e-05
#> b        -5.052180e-04  9.686636e-04 -1.325770e-06 -4.432122e-06  1.225133e-05
#> a         6.346709e-06 -1.325770e-06  1.062921e-03  7.613548e-06  3.741964e-06
#> Y~~Y     -5.736215e-06 -4.432122e-06  7.613548e-06  1.985689e-03  4.851662e-05
#> M~~M     -2.211335e-05  1.225133e-05  3.741964e-06  4.851662e-05  2.097639e-03
#> X~~X      3.013425e-05 -2.484170e-05  5.946802e-06  1.810136e-06 -8.697004e-06
#> indirect -2.606088e-04  5.051235e-04  5.298033e-04  1.099075e-06  8.007263e-06
#> direct    1.316432e-03 -5.052180e-04  6.346709e-06 -5.736215e-06 -2.211335e-05
#> total     1.055824e-03 -9.453330e-08  5.361500e-04 -4.637140e-06 -1.410609e-05
#>                    X~~X      indirect        direct         total
#> cp        3.013425e-05 -2.606088e-04  1.316432e-03  1.055824e-03
#> b        -2.484170e-05  5.051235e-04 -5.052180e-04 -9.453330e-08
#> a         5.946802e-06  5.298033e-04  6.346709e-06  5.361500e-04
#> Y~~Y      1.810136e-06  1.099075e-06 -5.736215e-06 -4.637140e-06
```

```
#> M~~M     -8.697004e-06  8.007263e-06 -2.211335e-05 -1.410609e-05
#> X~~X      1.843568e-03 -9.968961e-06  3.013425e-05  2.016529e-05
#> indirect -9.968961e-06  5.291978e-04 -2.606088e-04  2.685890e-04
#> direct    3.013425e-05 -2.606088e-04  1.316432e-03  1.055824e-03
#> total     2.016529e-05  2.685890e-04  1.055824e-03  1.324413e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1727 0.0290 20000 0.0752 0.0956 0.1152 0.2289 0.2462 0.2679
#> b         0.4666 0.0265 20000 0.3733 0.3965 0.4142 0.5187 0.5346 0.5521
#> a         0.4511 0.0252 20000 0.3681 0.3858 0.4007 0.4994 0.5144 0.5304
#> Y~~Y      0.6798 0.0242 20000 0.5991 0.6151 0.6309 0.7262 0.7399 0.7594
#> M~~M      0.7965 0.0227 20000 0.7187 0.7354 0.7506 0.8394 0.8512 0.8645
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2105 0.0171 20000 0.1548 0.1677 0.1776 0.2441 0.2553 0.2674
#> direct   0.1727 0.0290 20000 0.0752 0.0956 0.1152 0.2289 0.2462 0.2679
#> total    0.3832 0.0272 20000 0.2921 0.3111 0.3282 0.4347 0.4507 0.4684
```

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