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       0.1977 0.0357 20000 0.0821 0.1057 0.1281 0.2685 0.2902 0.3165
#> b        0.5596 0.0316 20000 0.4548 0.4793 0.4985 0.6206 0.6405 0.6645
#> a        0.5044 0.0321 20000 0.4014 0.4218 0.4415 0.5674 0.5872 0.6063
#> Y~~Y     1.0834 0.0487 20000 0.9209 0.9578 0.9875 1.1798 1.2092 1.2455
#> M~~M     1.0710 0.0479 20000 0.9130 0.9464 0.9762 1.1654 1.1953 1.2268
#> indirect 0.2823 0.0241 20000 0.2093 0.2233 0.2361 0.3301 0.3461 0.3681
#> direct   0.1977 0.0357 20000 0.0821 0.1057 0.1281 0.2685 0.2902 0.3165
```

```
#> total    0.4800 0.0367 20000 0.3574 0.3855 0.4091 0.5532 0.5765 0.6058
```

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        1.297219e-03 -5.250926e-04 -5.118291e-06 -6.665948e-06  3.089845e-06
#> b        -5.250926e-04  1.025730e-03  1.340118e-05 -6.366571e-06  9.481633e-06
#> a        -5.118291e-06  1.340118e-05  1.029766e-03  9.937196e-06  9.658389e-06
#> Y~~Y     -6.665948e-06 -6.366571e-06  9.937196e-06  2.370078e-03 -7.738213e-06
#> M~~M      3.089845e-06  9.481633e-06  9.658389e-06 -7.738213e-06  2.263697e-03
#> X~~X      8.553507e-07  1.418367e-06 -1.602564e-05  9.914954e-06 -6.389382e-06
#> indirect -2.672024e-04  5.245125e-04  5.831325e-04  2.151821e-06  1.001905e-05
#> direct    1.297219e-03 -5.250926e-04 -5.118291e-06 -6.665948e-06  3.089845e-06
#> total     1.030017e-03 -5.801216e-07  5.780142e-04 -4.514127e-06  1.310890e-05
#>                    X~~X      indirect        direct         total
#> cp        8.553507e-07 -2.672024e-04  1.297219e-03  1.030017e-03
#> b         1.418367e-06  5.245125e-04 -5.250926e-04 -5.801216e-07
#> a        -1.602564e-05  5.831325e-04 -5.118291e-06  5.780142e-04
#> Y~~Y      9.914954e-06  2.151821e-06 -6.665948e-06 -4.514127e-06
```

```
#> M~~M     -6.389382e-06  1.001905e-05  3.089845e-06  1.310890e-05
#> X~~X      2.206330e-03 -8.464720e-06  8.553507e-07 -7.609369e-06
#> indirect -8.464720e-06  5.918057e-04 -2.672024e-04  3.246033e-04
#> direct    8.553507e-07 -2.672024e-04  1.297219e-03  1.030017e-03
#> total    -7.609369e-06  3.246033e-04  1.030017e-03  1.354620e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1576 0.0285 20000 0.0663 0.0843 0.1015 0.2128 0.2300 0.2519
#> b        0.5024 0.0259 20000 0.4103 0.4343 0.4511 0.5525 0.5696 0.5894
#> a        0.4478 0.0253 20000 0.3605 0.3803 0.3967 0.4956 0.5115 0.5294
#> Y~~Y     0.6518 0.0245 20000 0.5662 0.5879 0.6028 0.6989 0.7125 0.7316
#> M~~M     0.7995 0.0226 20000 0.7197 0.7383 0.7544 0.8426 0.8554 0.8701
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2250 0.0177 20000 0.1697 0.1800 0.1904 0.2597 0.2702 0.2850
#> direct   0.1576 0.0285 20000 0.0663 0.0843 0.1015 0.2128 0.2300 0.2519
#> total    0.3826 0.0271 20000 0.2934 0.3094 0.3284 0.4344 0.4475 0.4640
```

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