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       0.2307 0.0342 20000 0.1162 0.1412 0.1641 0.2978 0.3200 0.3403
#> b        0.5395 0.0303 20000 0.4418 0.4629 0.4806 0.5996 0.6174 0.6399
#> a        0.4694 0.0331 20000 0.3596 0.3837 0.4042 0.5341 0.5557 0.5786
#> Y~~Y     0.9506 0.0422 20000 0.8096 0.8411 0.8673 1.0317 1.0578 1.0875
#> M~~M     1.0568 0.0472 20000 0.9059 0.9344 0.9638 1.1502 1.1779 1.2151
#> indirect 0.2533 0.0229 20000 0.1851 0.1965 0.2095 0.2993 0.3137 0.3307
#> direct   0.2307 0.0342 20000 0.1162 0.1412 0.1641 0.2978 0.3200 0.3403
```

```
#> total    0.4840 0.0360 20000 0.3640 0.3891 0.4129 0.5542 0.5765 0.6032
```

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        1.192976e-03 -4.278812e-04 -2.190568e-07 -2.312258e-06 -6.433054e-06
#> b        -4.278812e-04  9.028400e-04 -1.985421e-06  1.045192e-05  5.786366e-06
#> a        -2.190568e-07 -1.985421e-06  1.091393e-03 -1.854116e-06  1.925492e-05
#> Y~~Y     -2.312258e-06  1.045192e-05 -1.854116e-06  1.823398e-03 -1.283882e-05
#> M~~M     -6.433054e-06  5.786366e-06  1.925492e-05 -1.283882e-05  2.231410e-03
#> X~~X      1.713838e-05  3.234619e-06  1.703421e-05  6.085439e-07  4.973644e-06
#> indirect -2.009749e-04  4.229262e-04  5.876471e-04  4.382589e-06  1.259026e-05
#> direct    1.192976e-03 -4.278812e-04 -2.190568e-07 -2.312258e-06 -6.433054e-06
#> total     9.920007e-04 -4.954993e-06  5.874281e-04  2.070331e-06  6.157206e-06
#>                    X~~X      indirect        direct         total
#> cp       1.713838e-05 -2.009749e-04  1.192976e-03  9.920007e-04
#> b        3.234619e-06  4.229262e-04 -4.278812e-04 -4.954993e-06
#> a        1.703421e-05  5.876471e-04 -2.190568e-07  5.874281e-04
#> Y~~Y     6.085439e-07  4.382589e-06 -2.312258e-06  2.070331e-06
```

```
#> M~~M     4.973644e-06  1.259026e-05 -6.433054e-06  6.157206e-06
#> X~~X     1.838280e-03  1.108952e-05  1.713838e-05  2.822790e-05
#> indirect 1.108952e-05  5.165231e-04 -2.009749e-04  3.155482e-04
#> direct   1.713838e-05 -2.009749e-04  1.192976e-03  9.920007e-04
#> total    2.822790e-05  3.155482e-04  9.920007e-04  1.307549e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1858 0.0276 20000 0.0983 0.1142 0.1318 0.2403 0.2564 0.2735
#> b         0.4989 0.0249 20000 0.4093 0.4336 0.4494 0.5466 0.5618 0.5794
#> a         0.4088 0.0263 20000 0.3193 0.3391 0.3569 0.4595 0.4742 0.4922
#> Y~~Y      0.6407 0.0243 20000 0.5582 0.5762 0.5928 0.6874 0.7015 0.7177
#> M~~M      0.8329 0.0214 20000 0.7577 0.7751 0.7888 0.8726 0.8850 0.8980
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect  0.2040 0.0169 20000 0.1507 0.1617 0.1715 0.2378 0.2477 0.2620
#> direct    0.1858 0.0276 20000 0.0983 0.1142 0.1318 0.2403 0.2564 0.2735
#> total     0.3898 0.0269 20000 0.3004 0.3191 0.3358 0.4418 0.4593 0.4754
```

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