semmcci: Monte Carlo Confidence Intervals

Ivan Jacob Agaloos Pesigan

Installation

You can install the CRAN release of `semmcci` with:

```r
install.packages("semmcci")
```

You can install the development version of `semmcci` from GitHub with:

```r
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.

```r
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`.

```r
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.

```r
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       0.2558 0.0358 20000 0.1400 0.1640 0.1863 0.3261 0.3473 0.3715
#> b        0.5099 0.0313 20000 0.4074 0.4290 0.4494 0.5722 0.5907 0.6143
#> a        0.5570 0.0317 20000 0.4557 0.4754 0.4945 0.6182 0.6373 0.6612
#> Y~~Y     0.9874 0.0440 20000 0.8440 0.8744 0.9028 1.0748 1.1014 1.1271
#> M~~M     1.0003 0.0448 20000 0.8530 0.8872 0.9136 1.0883 1.1173 1.1535
#> indirect 0.2840 0.0238 20000 0.2097 0.2255 0.2390 0.3319 0.3487 0.3679
#> direct   0.2558 0.0358 20000 0.1400 0.1640 0.1863 0.3261 0.3473 0.3715
```

```
#> total    0.5398 0.0350 20000 0.4280 0.4507 0.4718 0.6095 0.6295 0.6512
```

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        1.280300e-03 -5.375900e-04 -6.874128e-07 -1.028373e-05  6.134927e-06
#> b        -5.375900e-04  9.742650e-04 -3.032159e-06  1.705357e-05 -1.317212e-05
#> a        -6.874128e-07 -3.032159e-06  9.923394e-04  1.065606e-05  7.708458e-06
#> Y~~Y     -1.028373e-05  1.705357e-05  1.065606e-05  1.938663e-03  3.734319e-06
#> M~~M      6.134927e-06 -1.317212e-05  7.708458e-06  3.734319e-06  1.981644e-03
#> X~~X     -9.931482e-07  1.312026e-05  6.899855e-06 -2.198583e-05  1.671904e-06
#> indirect -2.996961e-04  5.413500e-04  5.039047e-04  1.427913e-05 -3.500830e-06
#> direct    1.280300e-03 -5.375900e-04 -6.874128e-07 -1.028373e-05  6.134927e-06
#> total     9.806041e-04  3.759986e-06  5.032173e-04  3.995400e-06  2.634097e-06
#>                   X~~X      indirect        direct         total
#> cp       -9.931482e-07 -2.996961e-04  1.280300e-03 9.806041e-04
#> b         1.312026e-05  5.413500e-04 -5.375900e-04 3.759986e-06
#> a         6.899855e-06  5.039047e-04 -6.874128e-07 5.032173e-04
#> Y~~Y     -2.198583e-05  1.427913e-05 -1.028373e-05 3.995400e-06
```

```
#> M~~M      1.671904e-06 -3.500830e-06  6.134927e-06 2.634097e-06
#> X~~X      1.991130e-03  1.057845e-05 -9.931482e-07 9.585306e-06
#> indirect  1.057845e-05  5.593845e-04 -2.996961e-04 2.596884e-04
#> direct   -9.931482e-07 -2.996961e-04  1.280300e-03 9.806041e-04
#> total     9.585306e-06  2.596884e-04  9.806041e-04 1.240292e-03
```

```
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.2062 0.0285 20000 0.1076 0.1323 0.1504 0.2623 0.2781 0.2981
#> b        0.4706 0.0262 20000 0.3842 0.4017 0.4174 0.5208 0.5368 0.5587
#> a        0.4867 0.0240 20000 0.4038 0.4234 0.4386 0.5329 0.5470 0.5631
#> Y~~Y     0.6416 0.0241 20000 0.5614 0.5786 0.5938 0.6883 0.7021 0.7151
#> M~~M     0.7632 0.0234 20000 0.6830 0.7008 0.7160 0.8076 0.8207 0.8370
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2290 0.0175 20000 0.1743 0.1847 0.1949 0.2633 0.2757 0.2895
#> direct   0.2062 0.0285 20000 0.1076 0.1323 0.1504 0.2623 0.2781 0.2981
#> total    0.4352 0.0256 20000 0.3505 0.3682 0.3840 0.4839 0.4992 0.5142
```

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