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.

```r
library(semmcci)
library(lavaan)
```

Data

```r
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       0.2321 0.0387 20000 0.1024 0.1328 0.1567 0.3072 0.3326 0.3618
#> b        0.4488 0.0336 20000 0.3361 0.3614 0.3839 0.5146 0.5356 0.5612
#> a        0.4997 0.0324 20000 0.3928 0.4147 0.4356 0.5633 0.5816 0.6025
#> Y~~Y     1.1075 0.0494 20000 0.9370 0.9770 1.0102 1.2056 1.2362 1.2695
#> M~~M     0.9841 0.0440 20000 0.8451 0.8706 0.8968 1.0699 1.0970 1.1296
#> indirect 0.2243 0.0222 20000 0.1564 0.1704 0.1830 0.2694 0.2845 0.3021
#> direct   0.2321 0.0387 20000 0.1024 0.1328 0.1567 0.3072 0.3326 0.3618
```

```
#> total    0.4564 0.0378 20000 0.3312 0.3615 0.3834 0.5312 0.5526 0.5776
```

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        1.447274e-03 -5.432003e-04 -2.767213e-06 -7.355977e-06 -8.889463e-08
#> b        -5.432003e-04  1.121322e-03 -5.777322e-06 -5.907168e-06 -1.083452e-05
#> a        -2.767213e-06 -5.777322e-06  1.063724e-03  1.111161e-05  1.585965e-06
#> Y~~Y     -7.355977e-06 -5.907168e-06  1.111161e-05  2.437816e-03  1.442226e-05
#> M~~M     -8.889463e-08 -1.083452e-05  1.585965e-06  1.442226e-05  1.921527e-03
#> X~~X     -2.933820e-06  1.047595e-05  1.218935e-06  3.863258e-05 -4.471229e-06
#> indirect -2.722460e-04  5.573569e-04  4.748395e-04  1.243631e-06 -5.205345e-06
#> direct    1.447274e-03 -5.432003e-04 -2.767213e-06 -7.355977e-06 -8.889463e-08
#> total     1.175028e-03  1.415669e-05  4.720723e-04 -6.112347e-06 -5.294240e-06
#>                   X~~X      indirect        direct         total
#> cp       -2.933820e-06 -2.722460e-04  1.447274e-03  1.175028e-03
#> b         1.047595e-05  5.573569e-04 -5.432003e-04  1.415669e-05
#> a         1.218935e-06  4.748395e-04 -2.767213e-06  4.720723e-04
#> Y~~Y      3.863258e-05  1.243631e-06 -7.355977e-06 -6.112347e-06
```

```
#> M~~M     -4.471229e-06 -5.205345e-06 -8.889463e-08 -5.294240e-06
#> X~~X      1.754762e-03  6.025333e-06 -2.933820e-06  3.091513e-06
#> indirect  6.025333e-06  4.927788e-04 -2.722460e-04  2.205328e-04
#> direct   -2.933820e-06 -2.722460e-04  1.447274e-03  1.175028e-03
#> total     3.091513e-06  2.205328e-04  1.175028e-03  1.395561e-03
```

```r
MCStd(unstd)
```

```
#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1827 0.0296 20000 0.0844 0.1040 0.1242 0.2405 0.2606 0.2816
#> b        0.4042 0.0281 20000 0.3090 0.3312 0.3491 0.4585 0.4757 0.4940
#> a        0.4366 0.0256 20000 0.3514 0.3687 0.3854 0.4858 0.5011 0.5160
#> Y~~Y     0.7388 0.0239 20000 0.6525 0.6736 0.6894 0.7834 0.7962 0.8138
#> M~~M     0.8094 0.0224 20000 0.7337 0.7489 0.7640 0.8515 0.8641 0.8765
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1765 0.0164 20000 0.1247 0.1366 0.1450 0.2095 0.2197 0.2331
#> direct   0.1827 0.0296 20000 0.0844 0.1040 0.1242 0.2405 0.2606 0.2816
#> total    0.3592 0.0274 20000 0.2669 0.2866 0.3048 0.4124 0.4285 0.4477
```

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