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.

```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       0.2472 0.0377 20000 0.1255 0.1500 0.1724 0.3200 0.3440 0.3722
#> b        0.4879 0.0333 20000 0.3821 0.4024 0.4225 0.5530 0.5720 0.5932
#> a        0.5237 0.0316 20000 0.4188 0.4415 0.4615 0.5851 0.6053 0.6246
#> Y~~Y     1.1073 0.0494 20000 0.9388 0.9777 1.0093 1.2046 1.2350 1.2673
#> M~~M     1.0044 0.0445 20000 0.8579 0.8888 0.9170 1.0915 1.1204 1.1581
#> indirect 0.2555 0.0232 20000 0.1854 0.1991 0.2114 0.3022 0.3188 0.3382
#> direct   0.2472 0.0377 20000 0.1255 0.1500 0.1724 0.3200 0.3440 0.3722
```

```
#> total    0.5027 0.0369 20000 0.3831 0.4069 0.4296 0.5749 0.5973 0.6179
```

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        1.413122e-03 -5.859802e-04 -2.859925e-05  4.391051e-06  8.545469e-06
#> b        -5.859802e-04  1.116101e-03  2.365165e-05  4.297775e-06 -7.684940e-06
#> a        -2.859925e-05  2.365165e-05  9.989046e-04 -9.018566e-06 -1.538130e-05
#> Y~~Y      4.391051e-06  4.297775e-06 -9.018566e-06  2.444604e-03  9.433912e-06
#> M~~M      8.545469e-06 -7.684940e-06 -1.538130e-05  9.433912e-06  1.980710e-03
#> X~~X      1.146362e-05 -5.506790e-06  5.545539e-06  4.326352e-05 -1.818124e-05
#> indirect -3.207473e-04  5.960005e-04  5.001689e-04 -2.096400e-06 -1.145893e-05
#> direct    1.413122e-03 -5.859802e-04 -2.859925e-05  4.391051e-06  8.545469e-06
#> total     1.092375e-03  1.002022e-05  4.715696e-04  2.294651e-06 -2.913457e-06
#>                   X~~X      indirect        direct         total
#> cp        1.146362e-05 -3.207473e-04  1.413122e-03  1.092375e-03
#> b        -5.506790e-06  5.960005e-04 -5.859802e-04  1.002022e-05
#> a         5.545539e-06  5.001689e-04 -2.859925e-05  4.715696e-04
#> Y~~Y      4.326352e-05 -2.096400e-06  4.391051e-06  2.294651e-06
```

```
#> M~~M     -1.818124e-05 -1.145893e-05  8.545469e-06 -2.913457e-06
#> X~~X      1.952305e-03 -8.362207e-08  1.146362e-05  1.137999e-05
#> indirect -8.362207e-08  5.574545e-04 -3.207473e-04  2.367073e-04
#> direct    1.146362e-05 -3.207473e-04  1.413122e-03  1.092375e-03
#> total     1.137999e-05  2.367073e-04  1.092375e-03  1.329082e-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       0.1950 0.0293 20000 0.0960 0.1196 0.1384 0.2524 0.2713 0.2961
#> b        0.4362 0.0276 20000 0.3445 0.3627 0.3806 0.4894 0.5053 0.5232
#> a        0.4621 0.0248 20000 0.3726 0.3968 0.4134 0.5106 0.5239 0.5418
#> Y~~Y     0.6931 0.0242 20000 0.6061 0.6283 0.6441 0.7386 0.7526 0.7676
#> M~~M     0.7865 0.0229 20000 0.7065 0.7255 0.7393 0.8291 0.8425 0.8612
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2016 0.0173 20000 0.1477 0.1582 0.1683 0.2362 0.2475 0.2592
#> direct   0.1950 0.0293 20000 0.0960 0.1196 0.1384 0.2524 0.2713 0.2961
#> total    0.3966 0.0264 20000 0.3044 0.3272 0.3440 0.4472 0.4643 0.4809
```

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