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       0.2375 0.0362 20000 0.1213 0.1461 0.1662 0.3080 0.3294 0.3526
#> b        0.4844 0.0319 20000 0.3793 0.4040 0.4227 0.5467 0.5675 0.5893
#> a        0.5055 0.0321 20000 0.4027 0.4218 0.4421 0.5676 0.5883 0.6099
#> Y~~Y     1.0368 0.0462 20000 0.8802 0.9151 0.9455 1.1269 1.1539 1.1848
#> M~~M     1.0276 0.0458 20000 0.8719 0.9054 0.9373 1.1164 1.1465 1.1823
#> indirect 0.2449 0.0224 20000 0.1782 0.1898 0.2034 0.2898 0.3041 0.3210
#> direct   0.2375 0.0362 20000 0.1213 0.1461 0.1662 0.3080 0.3294 0.3526
```

```
#> total    0.4823 0.0361 20000 0.3685 0.3902 0.4122 0.5532 0.5753 0.6011
```

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        1.313798e-03 -5.134361e-04  5.121771e-06 -6.062740e-06 -2.473067e-05
#> b        -5.134361e-04  1.000323e-03  3.458649e-06 -1.425894e-05  1.033140e-05
#> a         5.121771e-06  3.458649e-06  1.047103e-03 -8.867665e-06  5.129228e-06
#> Y~~Y     -6.062740e-06 -1.425894e-05 -8.867665e-06  2.127460e-03 -1.737722e-06
#> M~~M     -2.473067e-05  1.033140e-05  5.129228e-06 -1.737722e-06  2.100311e-03
#> X~~X      3.677565e-06  4.908932e-06 -1.857844e-05  7.312699e-06 -1.236075e-05
#> indirect -2.574055e-04  5.080053e-04  5.090935e-04 -1.166896e-05  7.698932e-06
#> direct    1.313798e-03 -5.134361e-04  5.121771e-06 -6.062740e-06 -2.473067e-05
#> total     1.056392e-03 -5.430780e-06  5.142153e-04 -1.773170e-05 -1.703174e-05
#>                   X~~X      indirect        direct         total
#> cp        3.677565e-06 -2.574055e-04  1.313798e-03  1.056392e-03
#> b         4.908932e-06  5.080053e-04 -5.134361e-04 -5.430780e-06
#> a        -1.857844e-05  5.090935e-04  5.121771e-06  5.142153e-04
#> Y~~Y      7.312699e-06 -1.166896e-05 -6.062740e-06 -1.773170e-05
```

```
#> M~~M     -1.236075e-05  7.698932e-06 -2.473067e-05 -1.703174e-05
#> X~~X      1.998463e-03 -6.664441e-06  3.677565e-06 -2.986876e-06
#> indirect -6.664441e-06  5.048776e-04 -2.574055e-04  2.474722e-04
#> direct    3.677565e-06 -2.574055e-04  1.313798e-03  1.056392e-03
#> total    -2.986876e-06  2.474722e-04  1.056392e-03  1.303864e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>             est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp       0.1931 0.0291 20000 0.1003 0.1195 0.1355 0.2494 0.2673 0.2865
#> b        0.4465 0.0270 20000 0.3568 0.3758 0.3938 0.4984 0.5150 0.5322
#> a        0.4461 0.0254 20000 0.3624 0.3775 0.3953 0.4945 0.5090 0.5249
#> Y~~Y     0.6864 0.0244 20000 0.6044 0.6223 0.6374 0.7327 0.7479 0.7625
#> M~~M     0.8010 0.0226 20000 0.7245 0.7409 0.7555 0.8437 0.8575 0.8687
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1992 0.0169 20000 0.1443 0.1573 0.1669 0.2329 0.2445 0.2588
#> direct   0.1931 0.0291 20000 0.1003 0.1195 0.1355 0.2494 0.2673 0.2865
#> total    0.3923 0.0270 20000 0.3068 0.3224 0.3383 0.4440 0.4608 0.4777
```

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