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.

```
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       0.2600 0.0360 20000 0.1450 0.1692 0.1896 0.3300 0.3509 0.3782
#> b        0.4938 0.0328 20000 0.3848 0.4107 0.4295 0.5589 0.5772 0.5989
#> a        0.5685 0.0297 20000 0.4708 0.4911 0.5100 0.6261 0.6445 0.6685
#> Y~~Y     0.9823 0.0440 20000 0.8391 0.8675 0.8953 1.0685 1.0960 1.1311
#> M~~M     0.9116 0.0408 20000 0.7796 0.8053 0.8308 0.9912 1.0165 1.0427
#> indirect 0.2808 0.0238 20000 0.2079 0.2216 0.2353 0.3288 0.3468 0.3677
#> direct   0.2600 0.0360 20000 0.1450 0.1692 0.1896 0.3300 0.3509 0.3782
```

```
#> total    0.5407 0.0340 20000 0.4290 0.4535 0.4742 0.6070 0.6269 0.6496
```

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         1.284612e-03 -6.120940e-04  2.568992e-06  5.927957e-06  2.246000e-06
#> b         -6.120940e-04  1.080804e-03 -2.579831e-06 -7.858705e-06 -3.476443e-06
#> a          2.568992e-06 -2.579831e-06  8.600470e-04  8.470432e-06 -5.128840e-06
#> Y~~Y       5.927957e-06 -7.858705e-06  8.470432e-06  1.917924e-03 -1.571204e-05
#> M~~M       2.246000e-06 -3.476443e-06 -5.128840e-06 -1.571204e-05  1.665324e-03
#> X~~X       6.909247e-06 -6.392940e-06  7.782849e-06 -7.983858e-06  1.618331e-05
#> indirect -3.470124e-04  6.136192e-04  4.231758e-04  3.883610e-07 -4.080059e-06
#> direct     1.284612e-03 -6.120940e-04  2.568992e-06  5.927957e-06  2.246000e-06
#> total      9.375992e-04  1.525259e-06  4.257448e-04  6.316318e-06 -1.834059e-06
#>                    X~~X      indirect        direct         total
#> cp         6.909247e-06 -3.470124e-04  1.284612e-03  9.375992e-04
#> b         -6.392940e-06  6.136192e-04 -6.120940e-04  1.525259e-06
#> a          7.782849e-06  4.231758e-04  2.568992e-06  4.257448e-04
#> Y~~Y      -7.983858e-06  3.883610e-07  5.927957e-06  6.316318e-06
```

```
#> M~~M      1.618331e-05 -4.080059e-06  2.246000e-06 -1.834059e-06
#> X~~X      2.191026e-03  9.919494e-08  6.909247e-06  7.008442e-06
#> indirect  9.919494e-08  5.589582e-04 -3.470124e-04  2.119458e-04
#> direct    6.909247e-06 -3.470124e-04  1.284612e-03  9.375992e-04
#> total     7.008442e-06  2.119458e-04  9.375992e-04  1.149545e-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       0.2166 0.0294 20000 0.1227 0.1417 0.1585 0.2742 0.2920 0.3107
#> b        0.4492 0.0277 20000 0.3562 0.3777 0.3946 0.5027 0.5206 0.5371
#> a        0.5208 0.0230 20000 0.4415 0.4594 0.4749 0.5652 0.5800 0.5938
#> Y~~Y     0.6499 0.0243 20000 0.5688 0.5845 0.6009 0.6968 0.7123 0.7279
#> M~~M     0.7288 0.0240 20000 0.6474 0.6636 0.6805 0.7745 0.7890 0.8051
#> X~~X     1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2340 0.0181 20000 0.1761 0.1886 0.1990 0.2699 0.2813 0.2978
#> direct   0.2166 0.0294 20000 0.1227 0.1417 0.1585 0.2742 0.2920 0.3107
#> total    0.4506 0.0252 20000 0.3677 0.3831 0.3998 0.4989 0.5132 0.5321
```

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