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.

```
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       0.2629 0.0365 20000 0.1450 0.1689 0.1914 0.3342 0.3565 0.3865
#> b        0.4962 0.0322 20000 0.3930 0.4145 0.4330 0.5592 0.5795 0.5999
#> a        0.5230 0.0319 20000 0.4171 0.4407 0.4614 0.5870 0.6044 0.6270
#> Y~~Y     1.0281 0.0459 20000 0.8690 0.9094 0.9377 1.1176 1.1458 1.1745
#> M~~M     0.9942 0.0445 20000 0.8484 0.8782 0.9073 1.0811 1.1081 1.1382
#> indirect 0.2595 0.0233 20000 0.1891 0.2027 0.2154 0.3065 0.3219 0.3450
#> direct   0.2629 0.0365 20000 0.1450 0.1689 0.1914 0.3342 0.3565 0.3865
```

```
#> total    0.5224 0.0361 20000 0.4036 0.4289 0.4515 0.5928 0.6140 0.6425
```

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        1.330433e-03 -5.312825e-04  5.109631e-06 -3.203616e-07  1.314677e-06
#> b        -5.312825e-04  1.029497e-03 -1.229154e-05  6.494699e-06  3.720661e-07
#> a         5.109631e-06 -1.229154e-05  1.010689e-03 -4.580137e-06 -9.247503e-06
#> Y~~Y     -3.203616e-07  6.494699e-06 -4.580137e-06  2.131567e-03 -2.042144e-05
#> M~~M      1.314677e-06  3.720661e-07 -9.247503e-06 -2.042144e-05  1.960836e-03
#> X~~X     -1.226622e-05  6.391823e-06 -6.401552e-06 -1.215471e-05  7.181230e-07
#> indirect -2.752770e-04  5.328283e-04  4.947027e-04  1.367213e-06 -4.485131e-06
#> direct    1.330433e-03 -5.312825e-04  5.109631e-06 -3.203616e-07  1.314677e-06
#> total     1.055156e-03  1.545821e-06  4.998123e-04  1.046851e-06 -3.170454e-06
#>                    X~~X      indirect        direct         total
#> cp       -1.226622e-05 -2.752770e-04  1.330433e-03  1.055156e-03
#> b         6.391823e-06  5.328283e-04 -5.312825e-04  1.545821e-06
#> a        -6.401552e-06  4.947027e-04  5.109631e-06  4.998123e-04
#> Y~~Y     -1.215471e-05  1.367213e-06 -3.203616e-07  1.046851e-06
```

```
#> M~~M      7.181230e-07 -4.485131e-06  1.314677e-06 -3.170454e-06
#> X~~X      1.920119e-03  1.397830e-07 -1.226622e-05 -1.212644e-05
#> indirect  1.397830e-07  5.252207e-04 -2.752770e-04  2.499436e-04
#> direct   -1.226622e-05 -2.752770e-04  1.330433e-03  1.055156e-03
#> total    -1.212644e-05  2.499436e-04  1.055156e-03  1.305099e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2094 0.0286 20000 0.1144 0.1365 0.1534 0.2656 0.2839 0.3013
#> b         0.4491 0.0267 20000 0.3635 0.3793 0.3960 0.5008 0.5156 0.5350
#> a         0.4602 0.0249 20000 0.3746 0.3945 0.4111 0.5076 0.5229 0.5387
#> Y~~Y      0.6679 0.0243 20000 0.5882 0.6045 0.6193 0.7147 0.7290 0.7423
#> M~~M      0.7882 0.0229 20000 0.7098 0.7266 0.7423 0.8310 0.8444 0.8597
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2067 0.0169 20000 0.1549 0.1654 0.1740 0.2398 0.2508 0.2615
#> direct    0.2094 0.0286 20000 0.1144 0.1365 0.1534 0.2656 0.2839 0.3013
#> total     0.4161 0.0261 20000 0.3270 0.3464 0.3636 0.4665 0.4813 0.4957
```

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