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.

```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       0.1795 0.0351 20000 0.0675 0.0895 0.1114 0.2482 0.2713 0.2991
#> b        0.5021 0.0309 20000 0.3989 0.4219 0.4419 0.5629 0.5824 0.6017
#> a        0.5534 0.0315 20000 0.4487 0.4721 0.4916 0.6151 0.6331 0.6543
#> Y~~Y     0.9266 0.0414 20000 0.7900 0.8195 0.8461 1.0083 1.0318 1.0604
#> M~~M     0.9713 0.0434 20000 0.8328 0.8596 0.8866 1.0566 1.0832 1.1198
#> indirect 0.2778 0.0233 20000 0.2077 0.2211 0.2337 0.3254 0.3414 0.3580
#> direct   0.1795 0.0351 20000 0.0675 0.0895 0.1114 0.2482 0.2713 0.2991
```

```
#> total    0.4574 0.0345 20000 0.3451 0.3704 0.3911 0.5253 0.5475 0.5719
```

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        1.240350e-03 -5.173750e-04 -5.956105e-06  1.843068e-05 -7.954533e-06
#> b        -5.173750e-04  9.520521e-04  6.167124e-06 -7.788340e-06 -3.605053e-06
#> a        -5.956105e-06  6.167124e-06  1.011013e-03  3.366741e-06 -1.119919e-06
#> Y~~Y      1.843068e-05 -7.788340e-06  3.366741e-06  1.699510e-03 -1.976893e-05
#> M~~M     -7.954533e-06 -3.605053e-06 -1.119919e-06 -1.976893e-05  1.893894e-03
#> X~~X      4.217641e-06 -2.383784e-07  5.899021e-06 -5.445653e-06  1.880292e-05
#> indirect -2.894940e-04  5.302001e-04  5.108707e-04 -2.296769e-06 -3.048092e-06
#> direct    1.240350e-03 -5.173750e-04 -5.956105e-06  1.843068e-05 -7.954533e-06
#> total     9.508555e-04  1.282515e-05  5.049145e-04  1.613391e-05 -1.100262e-05
#>                   X~~X      indirect        direct         total
#> cp        4.217641e-06 -2.894940e-04  1.240350e-03  9.508555e-04
#> b        -2.383784e-07  5.302001e-04 -5.173750e-04  1.282515e-05
#> a         5.899021e-06  5.108707e-04 -5.956105e-06  5.049145e-04
#> Y~~Y     -5.445653e-06 -2.296769e-06  1.843068e-05  1.613391e-05
```

```
#> M~~M      1.880292e-05 -3.048092e-06 -7.954533e-06 -1.100262e-05
#> X~~X      1.909054e-03  2.869316e-06  4.217641e-06  7.086957e-06
#> indirect  2.869316e-06  5.509076e-04 -2.894940e-04  2.614136e-04
#> direct    4.217641e-06 -2.894940e-04  1.240350e-03  9.508555e-04
#> total     7.086957e-06  2.614136e-04  9.508555e-04  1.212269e-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        0.1510 0.0294 20000 0.0500 0.0751 0.0941 0.2085 0.2277 0.2475
#> b         0.4824 0.0270 20000 0.3925 0.4107 0.4287 0.5347 0.5508 0.5691
#> a         0.4845 0.0243 20000 0.4038 0.4204 0.4356 0.5312 0.5455 0.5594
#> Y~~Y      0.6739 0.0244 20000 0.5910 0.6097 0.6246 0.7203 0.7343 0.7536
#> M~~M      0.7653 0.0235 20000 0.6871 0.7025 0.7178 0.8102 0.8232 0.8369
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2337 0.0181 20000 0.1756 0.1897 0.1990 0.2694 0.2814 0.2936
#> direct    0.1510 0.0294 20000 0.0500 0.0751 0.0941 0.2085 0.2277 0.2475
#> total     0.3847 0.0269 20000 0.2925 0.3147 0.3310 0.4367 0.4534 0.4743
```

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