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       0.1755 0.0343 20000 0.0661 0.0873 0.1078 0.2422 0.2634 0.2865
#> b        0.5390 0.0301 20000 0.4432 0.4625 0.4805 0.5977 0.6160 0.6346
#> a        0.4997 0.0324 20000 0.3965 0.4169 0.4355 0.5630 0.5825 0.6058
#> Y~~Y     0.9432 0.0423 20000 0.7995 0.8327 0.8595 1.0258 1.0514 1.0824
#> M~~M     1.0432 0.0468 20000 0.8837 0.9212 0.9514 1.1350 1.1639 1.1964
#> indirect 0.2693 0.0232 20000 0.2004 0.2125 0.2251 0.3156 0.3312 0.3486
#> direct   0.1755 0.0343 20000 0.0661 0.0873 0.1078 0.2422 0.2634 0.2865
```

```
#> total    0.4448 0.0352 20000 0.3302 0.3548 0.3754 0.5137 0.5340 0.5583
```

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        1.185256e-03 -4.683742e-04  1.242234e-05  5.726843e-06  1.277345e-05
#> b        -4.683742e-04  9.131960e-04  8.874512e-07 -1.265609e-06 -1.518112e-05
#> a         1.242234e-05  8.874512e-07  1.051807e-03  9.939436e-06 -2.639793e-07
#> Y~~Y      5.726843e-06 -1.265609e-06  9.939436e-06  1.819667e-03  3.709644e-05
#> M~~M      1.277345e-05 -1.518112e-05 -2.639793e-07  3.709644e-05  2.189718e-03
#> X~~X     -9.162961e-06  1.648288e-05 -8.282540e-06  1.520580e-05  2.415114e-06
#> indirect -2.269688e-04  4.566296e-04  5.674281e-04  5.137777e-06 -7.383854e-06
#> direct    1.185256e-03 -4.683742e-04  1.242234e-05  5.726843e-06  1.277345e-05
#> total     9.582870e-04 -1.174458e-05  5.798505e-04  1.086462e-05  5.389595e-06
#>                    X~~X      indirect        direct         total
#> cp       -9.162961e-06 -2.269688e-04  1.185256e-03  9.582870e-04
#> b         1.648288e-05  4.566296e-04 -4.683742e-04 -1.174458e-05
#> a        -8.282540e-06  5.674281e-04  1.242234e-05  5.798505e-04
#> Y~~Y      1.520580e-05  5.137777e-06  5.726843e-06  1.086462e-05
```

```
#> M~~M      2.415114e-06 -7.383854e-06  1.277345e-05  5.389595e-06
#> X~~X      2.012364e-03  3.868342e-06 -9.162961e-06 -5.294620e-06
#> indirect  3.868342e-06  5.349008e-04 -2.269688e-04  3.079319e-04
#> direct   -9.162961e-06 -2.269688e-04  1.185256e-03  9.582870e-04
#> total    -5.294620e-06  3.079319e-04  9.582870e-04  1.266219e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1460 0.0284 20000 0.0536 0.0741 0.0906 0.2017 0.2194 0.2391
#> b         0.5100 0.0255 20000 0.4243 0.4407 0.4589 0.5580 0.5737 0.5903
#> a         0.4392 0.0256 20000 0.3561 0.3710 0.3880 0.4882 0.5045 0.5212
#> Y~~Y      0.6533 0.0242 20000 0.5699 0.5890 0.6055 0.7004 0.7156 0.7342
#> M~~M      0.8071 0.0224 20000 0.7284 0.7455 0.7617 0.8495 0.8624 0.8732
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2240 0.0177 20000 0.1691 0.1800 0.1897 0.2591 0.2706 0.2850
#> direct    0.1460 0.0284 20000 0.0536 0.0741 0.0906 0.2017 0.2194 0.2391
#> total     0.3700 0.0274 20000 0.2767 0.2978 0.3152 0.4231 0.4375 0.4542
```

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