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       0.2562 0.0329 20000 0.1450 0.1704 0.1915 0.3211 0.3407 0.3608
#> b        0.4409 0.0281 20000 0.3488 0.3690 0.3856 0.4962 0.5150 0.5336
#> a        0.4824 0.0334 20000 0.3710 0.3942 0.4168 0.5478 0.5686 0.5872
#> Y~~Y     0.8953 0.0400 20000 0.7617 0.7910 0.8159 0.9721 0.9961 1.0237
#> M~~M     1.1198 0.0498 20000 0.9569 0.9925 1.0218 1.2182 1.2491 1.2843
#> indirect 0.2127 0.0201 20000 0.1512 0.1634 0.1748 0.2531 0.2674 0.2858
#> direct   0.2562 0.0329 20000 0.1450 0.1704 0.1915 0.3211 0.3407 0.3608
```

```
#> total    0.4689 0.0334 20000 0.3533 0.3835 0.4034 0.5349 0.5565 0.5761
```

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         1.094720e-03 -3.910609e-04  6.384467e-06 -1.244200e-05 -1.146916e-06
#> b         -3.910609e-04  8.049652e-04 -4.008903e-07  2.882878e-06  2.036776e-06
#> a          6.384467e-06 -4.008903e-07  1.138236e-03  1.426431e-05 -1.124435e-05
#> Y~~Y      -1.244200e-05  2.882878e-06  1.426431e-05  1.589711e-03 -6.924555e-06
#> M~~M      -1.146916e-06  2.036776e-06 -1.124435e-05 -6.924555e-06  2.480172e-03
#> X~~X       2.104895e-05 -8.413002e-06 -1.400637e-05  1.689664e-05  2.982451e-05
#> indirect  -1.859458e-04  3.884094e-04  5.018674e-04  8.125131e-06 -3.450185e-06
#> direct     1.094720e-03 -3.910609e-04  6.384467e-06 -1.244200e-05 -1.146916e-06
#> total      9.087746e-04 -2.651486e-06  5.082519e-04 -4.316865e-06 -4.597102e-06
#>                    X~~X      indirect        direct         total
#> cp         2.104895e-05 -1.859458e-04  1.094720e-03  9.087746e-04
#> b         -8.413002e-06  3.884094e-04 -3.910609e-04 -2.651486e-06
#> a         -1.400637e-05  5.018674e-04  6.384467e-06  5.082519e-04
#> Y~~Y       1.689664e-05  8.125131e-06 -1.244200e-05 -4.316865e-06
```

```
#> M~~M      2.982451e-05 -3.450185e-06 -1.146916e-06 -4.597102e-06
#> X~~X      2.034214e-03 -1.078985e-05  2.104895e-05  1.025910e-05
#> indirect -1.078985e-05  4.097860e-04 -1.859458e-04  2.238401e-04
#> direct    2.104895e-05 -1.859458e-04  1.094720e-03  9.087746e-04
#> total     1.025910e-05  2.238401e-04  9.087746e-04  1.132615e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.2221 0.0283 20000 0.1312 0.1497 0.1663 0.2764 0.2929 0.3193
#> b         0.4441 0.0262 20000 0.3555 0.3752 0.3926 0.4952 0.5110 0.5282
#> a         0.4150 0.0263 20000 0.3228 0.3444 0.3624 0.4648 0.4820 0.4987
#> Y~~Y      0.6716 0.0244 20000 0.5896 0.6081 0.6221 0.7184 0.7326 0.7503
#> M~~M      0.8278 0.0218 20000 0.7513 0.7677 0.7840 0.8686 0.8814 0.8958
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.1843 0.0162 20000 0.1332 0.1439 0.1533 0.2165 0.2273 0.2403
#> direct   0.2221 0.0283 20000 0.1312 0.1497 0.1663 0.2764 0.2929 0.3193
#> total    0.4064 0.0267 20000 0.3195 0.3363 0.3528 0.4574 0.4728 0.4922
```

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