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       0.2268 0.0368 20000 0.1093 0.1323 0.1543 0.2997 0.3211 0.3477
#> b        0.4924 0.0317 20000 0.3872 0.4096 0.4301 0.5540 0.5722 0.5961
#> a        0.5495 0.0319 20000 0.4407 0.4674 0.4867 0.6125 0.6314 0.6552
#> Y~~Y     1.0087 0.0454 20000 0.8596 0.8939 0.9193 1.0984 1.1265 1.1588
#> M~~M     0.9854 0.0443 20000 0.8357 0.8696 0.8979 1.0718 1.0990 1.1354
#> indirect 0.2706 0.0234 20000 0.1971 0.2125 0.2264 0.3179 0.3336 0.3501
#> direct   0.2268 0.0368 20000 0.1093 0.1323 0.1543 0.2997 0.3211 0.3477
```

```
#> total    0.4974 0.0358 20000 0.3843 0.4072 0.4275 0.5678 0.5909 0.6126
```

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        1.339035e-03 -5.613822e-04  3.545902e-06  8.602144e-06 -1.198966e-05
#> b        -5.613822e-04  1.035543e-03 -1.222628e-06 -8.568018e-06  9.989194e-07
#> a         3.545902e-06 -1.222628e-06  1.020676e-03  1.350041e-06  6.966123e-06
#> Y~~Y      8.602144e-06 -8.568018e-06  1.350041e-06  2.042390e-03  7.471241e-06
#> M~~M     -1.198966e-05  9.989194e-07  6.966123e-06  7.471241e-06  1.948689e-03
#> X~~X      3.450876e-06  9.908660e-07  7.377668e-06 -1.130787e-05  8.095242e-06
#> indirect -3.072616e-04  5.693429e-04  5.016217e-04 -4.816721e-06  4.006631e-06
#> direct    1.339035e-03 -5.613822e-04  3.545902e-06  8.602144e-06 -1.198966e-05
#> total     1.031774e-03  7.960682e-06  5.051676e-04  3.785424e-06 -7.983024e-06
#>                    X~~X      indirect        direct         total
#> cp        3.450876e-06 -3.072616e-04  1.339035e-03  1.031774e-03
#> b         9.908660e-07  5.693429e-04 -5.613822e-04  7.960682e-06
#> a         7.377668e-06  5.016217e-04  3.545902e-06  5.051676e-04
#> Y~~Y     -1.130787e-05 -4.816721e-06  8.602144e-06  3.785424e-06
```

```
#> M~~M      8.095242e-06  4.006631e-06 -1.198966e-05 -7.983024e-06
#> X~~X      1.915495e-03  3.959213e-06  3.450876e-06  7.410089e-06
#> indirect  3.959213e-06  5.612770e-04 -3.072616e-04  2.540155e-04
#> direct    3.450876e-06 -3.072616e-04  1.339035e-03  1.031774e-03
#> total     7.410089e-06  2.540155e-04  1.031774e-03  1.285789e-03
```

```
MCStd(unstd)

#> Standardized Monte Carlo Confidence Intervals
#>              est     se     R  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
#> cp        0.1840 0.0294 20000 0.0915 0.1072 0.1257 0.2407 0.2603 0.2823
#> b         0.4566 0.0274 20000 0.3676 0.3844 0.4024 0.5097 0.5258 0.5465
#> a         0.4807 0.0245 20000 0.3988 0.4152 0.4312 0.5276 0.5414 0.5574
#> Y~~Y      0.6769 0.0245 20000 0.5955 0.6110 0.6266 0.7230 0.7378 0.7563
#> M~~M      0.7689 0.0235 20000 0.6894 0.7069 0.7216 0.8141 0.8276 0.8410
#> X~~X      1.0000 0.0000 20000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
#> indirect 0.2195 0.0177 20000 0.1653 0.1751 0.1854 0.2551 0.2672 0.2815
#> direct    0.1840 0.0294 20000 0.0915 0.1072 0.1257 0.2407 0.2603 0.2823
#> total     0.4035 0.0266 20000 0.3185 0.3342 0.3505 0.4554 0.4703 0.4877
```

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