Package ‘betaMC’

March 10, 2023

Title Monte Carlo Confidence Intervals for Standardized Regression Coefficients and Other Effect Sizes

Version 1.2.0.9000

Description Generates Monte Carlo confidence intervals for standardized regression coefficients and other effect sizes for models fitted by lm().
'betaMC' combines ideas from Monte Carlo confidence intervals for the indirect effect (Preacher and Selig, 2012 <doi:10.1080/19312458.2012.679848>) and the sampling covariance matrix of regression coefficients (Dudgeon, 2017 <doi:10.1007/s11336-017-9563-z>) to generate confidence intervals effect sizes in regression.

URL `https://github.com/jeksterslab/betaMC`, `https://jeksterslab.github.io/betaMC/`

BugReports `https://github.com/jeksterslab/betaMC/issues`

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 3.5.0)

Imports stats, methods

Suggests knitr, rmarkdown, testthat, MASS

RoxygenNote 7.2.3

NeedsCompilation no

Author Ivan Jacob Agaloos Pesigan [aut, cre, cph] (<`https://orcid.org/0000-0003-4818-8420`>)

Maintainer Ivan Jacob Agaloos Pesigan <`r.jeksterslab@gmail.com`>

R topics documented:

`BetaMC` *Estimate Standardized Regression Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method*

Description

Estimate Standardized Regression Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method

Usage

```
BetaMC(object)
```

Arguments

`object`	Object of class `mc`, that is, the output of the `MC()` function.

Details

The vector of standardized regression coefficients ($\hat{\boldsymbol{\beta}}$) is derived from each randomly generated vector of parameter estimates. Confidence intervals are generated by obtaining percentiles corresponding to $100(1-\alpha)\%$ from the generated sampling distribution of $\hat{\boldsymbol{\beta}}$, where α is the significance level.

Value

Returns an object of class `betamc` which is a list with the following elements:

object The function argument `object`.

thetahatstar Sampling distribution of $\hat{\boldsymbol{\beta}}$.

vcov Sampling variance-covariance matrix of $\hat{\boldsymbol{\beta}}$.

est Vector of estimated $\hat{\boldsymbol{\beta}}$.

fun Function used ("BetaMC").

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Beta Monte Carlo Functions: `DeltaRSqMC()`, `DiffBetaMC()`, `MC()`, `PCorMC()`, `RSqMC()`, `SCorMC()`

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
std <- BetaMC(mc)
# Methods --------------------------------------------------------
print(std)
summary(std)
coef(std)
vcov(std)
confint(std, level = 0.95)
```

`coef.betamc` *Estimated Parameter Method for an Object of Class* `betamc`

Description

Estimated Parameter Method for an Object of Class `betamc`

Usage

```
## S3 method for class 'betamc'
coef(object, ...)
```

Arguments

`object`	Object of Class `betamc`, that is, the output of the `BetaMC()`, `RSqMC()`, `SCorMC()`, `DeltaRSqMC()`, or `PCorMC()` functions.
`...`	additional arguments.

Value

Returns a vector of estimated parameters.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
std <- BetaMC(mc)
# Method -----------------------------------------------------
coef(std)
```

`confint.betamc` *Confidence Intervals Method for an Object of Class* `betamc`

Description

Confidence Intervals Method for an Object of Class `betamc`

Usage

```
## S3 method for class 'betamc'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

`object`	Object of Class `betamc`, that is, the output of the `BetaMC()`, `RSqMC()`, `SCorMC()`, `DeltaRSqMC()`, or `PCorMC()` functions.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`level`	the confidence level required.
`...`	additional arguments.

Value

Returns a matrix of confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
std <- BetaMC(mc)
# Method -----------------------------------------------------
confint(std, level = 0.95)
```

`DeltaRSqMC` *Estimate Improvement in R-Squared and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method*

Description

Estimate Improvement in R-Squared and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method

Usage

```
DeltaRSqMC(object)
```

Arguments

`object`	Object of class `mc`, that is, the output of the `MC()` function.

Details

The vector of improvement in R-squared (ΔR^2) is derived from each randomly generated vector of parameter estimates. Confidence intervals are generated by obtaining percentiles corresponding to $100(1-\alpha)\%$ from the generated sampling distribution of ΔR^2, where α is the significance level.

Value

Returns an object of class `betamc` which is a list with the following elements:

object The function argument `object`.

thetahatstar Sampling distribution of ΔR^2.

vcov Sampling variance-covariance matrix of ΔR^2.

est Vector of estimated ΔR^2.

fun Function used ("DeltaRSqMC").

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Beta Monte Carlo Functions: BetaMC(), DiffBetaMC(), MC(), PCorMC(), RSqMC(), SCorMC()

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for improvement in R-squared
deltarsq <- DeltaRSqMC(mc)
# Methods -------------------------------------------------------
print(deltarsq)
summary(deltarsq)
coef(deltarsq)
vcov(deltarsq)
confint(deltarsq, level = 0.95)
```

DiffBetaMC — *Estimate Differences of Standardized Slopes and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method*

Description

Estimate Differences of Standardized Slopes and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method

Usage

```
DiffBetaMC(object)
```

Arguments

object	Object of class mc, that is, the output of the MC() function.

Details

The vector of differences of standardized regression slopes is derived from each randomly generated vector of parameter estimates. Confidence intervals are generated by obtaining percentiles corresponding to $100(1-\alpha)\%$ from the generated sampling distribution of differences of standardized regression slopes, where α is the significance level.

Value

Returns an object of class `betamc` which is a list with the following elements:

object The function argument `object`.

thetahatstar Sampling distribution of differences of standardized regression slopes.

vcov Sampling variance-covariance matrix of differences of standardized regression slopes.

est Vector of estimated differences of standardized regression slopes.

fun Function used ("DiffBetaMC").

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Beta Monte Carlo Functions: `BetaMC()`, `DeltaRSqMC()`, `MC()`, `PCorMC()`, `RSqMC()`, `SCorMC()`

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals
# for differences of standardized regression slopes
diff <- DiffBetaMC(mc)
# Methods -------------------------------------------------------
print(diff)
summary(diff)
coef(diff)
vcov(diff)
confint(diff, level = 0.95)
```

`MC` *Generate the Sampling Distribution of Regression Parameters Using the Monte Carlo Method*

Description

Generate the Sampling Distribution of Regression Parameters Using the Monte Carlo Method

Usage

```
MC(
  object,
  R = 20000L,
  type = "hc3",
  g1 = 1,
  g2 = 1.5,
  k = 0.7,
  decomposition = "eigen",
  pd = TRUE,
  tol = 1e-06,
  fixed_x = FALSE,
  seed = NULL
)
```

Arguments

`object`	Object of class `lm`.
`R`	Positive integer. Number of Monte Carlo replications.
`type`	Character string. Sampling covariance matrix type. Possible values are `"mvn"`, `"adf"`, `"hc0"`, `"hc1"`, `"hc2"`, `"hc3"`, `"hc4"`, `"hc4m"`, and `"hc5"`. `type = "mvn"` uses the normal-theory sampling covariance matrix. `type = "adf"` uses the asymptotic distribution-free sampling covariance matrix. `type = "hc0"` through `"hc5"` uses different versions of heteroskedasticity-consistent sampling covariance matrix.
`g1`	Numeric. `g1` value for `type = "hc4m"` or `type = "hc5"`.
`g2`	Numeric. `g2` value for `type = "hc4m"`.
`k`	Numeric. Constant for `type = "hc5"`
`decomposition`	Character string. Matrix decomposition of the sampling variance-covariance matrix for the data generation. If `decomposition = "chol"`, use Cholesky decomposition. If `decomposition = "eigen"`, use eigenvalue decomposition. If `decomposition = "svd"`, use singular value decomposition.
`pd`	Logical. If `pd = TRUE`, check if the sampling variance-covariance matrix is positive definite using `tol`.
`tol`	Numeric. Tolerance used for `pd`.
`fixed_x`	Logical. If `fixed_x = TRUE`, treat the regressors as fixed. If `fixed_x = FALSE`, treat the regressors as random.
`seed`	Integer. Seed number for reproducibility.

Details

Let the parameter vector of the unstandardized regression model be given by

$$\boldsymbol{\theta} = \left\{ \mathbf{b}, \sigma^2, \mathrm{vech}\left(\boldsymbol{\Sigma}_{\mathbf{XX}} \right) \right\}$$

where $\mathbf{b}$ is the vector of regression slopes, σ^2 is the error variance, and $\mathrm{vech}\left(\boldsymbol{\Sigma}_{\mathbf{XX}} \right)$ is the vector of unique elements of the covariance matrix of the regressor variables. The empirical sampling distribution of $\boldsymbol{\theta}$ is generated using the Monte Carlo method, that is, random values of parameter estimates are sampled from the multivariate normal distribution using the estimated parameter vector as the mean vector and the specified sampling covariance matrix using the `type` argument as the covariance matrix. A replacement sampling approach is implemented to ensure that the model-implied covariance matrix is positive definite.

Value

Returns an object of class `mc` which is a list with the following elements:

call Function call.

args Function arguments.

lm_process Processed `lm` object.

scale Sampling variance-covariance matrix of parameter estimates.

location Parameter estimates.

thetahatstar Sampling distribution parameter estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

References

Dudgeon, P. (2017). Some improvements in confidence intervals for standardized regression coefficients. *Psychometrika*, *82*(4), 928–951. doi:10.1007/s113360179563z

Preacher, K. J., & Selig, J. P. (2012). Advantages of Monte Carlo confidence intervals for indirect effects. *Communication Methods and Measures*, *6*(2), 77-98. doi:10.1080/19312458.2012.679848

See Also

Other Beta Monte Carlo Functions: `BetaMC()`, `DeltaRSqMC()`, `DiffBetaMC()`, `PCorMC()`, `RSqMC()`, `SCorMC()`

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
MC(object, R = 100)
```

`nas1982` *1982 National Academy of Sciences Doctoral Programs Data*

Description

1982 National Academy of Sciences Doctoral Programs Data

Usage

```
nas1982
```

Format

Ratings of 46 doctoral programs in psychology in the USA with the following variables:

QUALITY Program quality ratings.

NFACUL Number of faculty members in the program.

NGRADS Number of program graduates.

PCTSUPP Percentage of program graduates who received support.

PCTGRT Percent of faculty members holding research grants.

NARTIC Number of published articles attributed to program faculty member.

PCTPUB Percent of faculty with one or more published article.

References

National Research Council. (1982). *An assessment of research-doctorate programs in the United States: Social and behavioral sciences*. doi:10.17226/9781. Reproduced with permission from the National Academy of Sciences, Courtesy of the National Academies Press, Washington, D.C.

`PCorMC` *Estimate Squared Partial Correlation Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method*

Description

Estimate Squared Partial Correlation Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method

Usage

```
PCorMC(object)
```

Arguments

`object`	Object of class `mc`, that is, the output of the `MC()` function.

Details

The vector of squared partial correlation coefficients (r^2_p) is derived from each randomly generated vector of parameter estimates. Confidence intervals are generated by obtaining percentiles corresponding to $100(1-\alpha)\%$ from the generated sampling distribution of r^2_p, where α is the significance level.

Value

Returns an object of class `betamc` which is a list with the following elements:

object The function argument `object`.

thetahatstar Sampling distribution of r^2_p.

vcov Sampling variance-covariance matrix of r^2_p.

est Vector of estimated r^2_p.

fun Function used ("PCorMC").

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Beta Monte Carlo Functions: `BetaMC()`, `DeltaRSqMC()`, `DiffBetaMC()`, `MC()`, `RSqMC()`, `SCorMC()`

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
rp <- PCorMC(mc)
# Methods -------------------------------------------------------
print(rp)
summary(rp)
coef(rp)
vcov(rp)
confint(rp, level = 0.95)
```

print.betamc — *Print Method for an Object of Class* `betamc`

Description

Print Method for an Object of Class `betamc`

Usage

```
## S3 method for class 'betamc'
print(x, alpha = c(0.05, 0.01, 0.001), digits = 4, ...)
```

Arguments

`x`	Object of Class `betamc`, that is, the output of the `BetaMC()`, `RSqMC()`, `SCorMC()`, `DeltaRSqMC()`, or `PCorMC()` functions.
`alpha`	Significance level.
`digits`	Digits to print.
`...`	additional arguments.

Value

Prints a matrix of estimates, standard errors, number of Monte Carlo replications, and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
std <- BetaMC(mc)
# Method ----------------------------------------------------------
print(std)
```

print.mc — *Print Method for an Object of Class* `mc`

Description

Print Method for an Object of Class `mc`

Usage

```
## S3 method for class 'mc'
print(x, ...)
```

Arguments

`x`	Object of Class `mc`.
`...`	additional arguments.

Value

Prints the first set of simulated parameter estimates and model-implied covariance matrix.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
mc <- MC(object, R = 100)
# use a large R, for example, R = 20000 for actual research
print(mc)
```

RSqMC — *Estimate Multiple Correlation Coefficients (R-Squared and Adjusted R-Squared) and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method*

Description

Estimate Multiple Correlation Coefficients (R-Squared and Adjusted R-Squared) and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method

Usage

```
RSqMC(object)
```

Arguments

`object`	Object of class `mc`, that is, the output of the `MC()` function.

Details

R-squared (R^2) and adjusted R-squared ($\bar{R}^2$) are derived from each randomly generated vector of parameter estimates. Confidence intervals are generated by obtaining percentiles corresponding to $100(1-\alpha)\%$ from the generated sampling distribution of R^2 and $\bar{R}^2$, where α is the significance level.

Value

Returns an object of class `betamc` which is a list with the following elements:

object The function argument `object`.

thetahatstar Sampling distribution of R^2 and $\bar{R}^2$.

vcov Sampling variance-covariance matrix of R^2 and $\bar{R}^2$.

est Vector of estimated R^2 and $\bar{R}^2$.

fun Function used ("RSqMC").

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Beta Monte Carlo Functions: `BetaMC()`, `DeltaRSqMC()`, `DiffBetaMC()`, `MC()`, `PCorMC()`, `SCorMC()`

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
rsq <- RSqMC(mc)
# Methods -------------------------------------------------------
print(rsq)
summary(rsq)
coef(rsq)
vcov(rsq)
confint(rsq, level = 0.95)
```

`SCorMC` *Estimate Semipartial Correlation Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method*

Description

Estimate Semipartial Correlation Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method

Usage

```
SCorMC(object)
```

Arguments

`object`	Object of class `mc`, that is, the output of the `MC()` function.

Details

The vector of semipartial correlation coefficients (r_s) is derived from each randomly generated vector of parameter estimates. Confidence intervals are generated by obtaining percentiles corresponding to $100(1-\alpha)\%$ from the generated sampling distribution of r_s, where α is the significance level.

Value

Returns an object of class `betamc` which is a list with the following elements:

object The function argument `object`.

thetahatstar Sampling distribution of r_s.

vcov Sampling variance-covariance matrix of r_s.

est Vector of estimated r_s.

fun Function used ("SCorMC").

Author(s)

Ivan Jacob Agaloos Pesigan

See Also

Other Beta Monte Carlo Functions: `BetaMC()`, `DeltaRSqMC()`, `DiffBetaMC()`, `MC()`, `PCorMC()`, `RSqMC()`

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
rs <- SCorMC(mc)
# Methods -----------------------------------------------------
print(rs)
summary(rs)
coef(rs)
vcov(rs)
confint(rs, level = 0.95)
```

---

summary.betamc *Summary Method for an Object of Class* betamc

Description

Summary Method for an Object of Class betamc

Usage

```
## S3 method for class 'betamc'
summary(object, alpha = c(0.05, 0.01, 0.001), digits = 4, ...)
```

Arguments

object	Object of Class betamc, that is, the output of the BetaMC(), RSqMC(), SCorMC(), DeltaRSqMC(), or PCorMC() functions.
alpha	Significance level.
digits	Digits to print.
...	additional arguments.

Value

Returns a matrix of estimates, standard errors, number of Monte Carlo replications, and confidence intervals.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
std <- BetaMC(mc)
# Method ---------------------------------------------------------
summary(std)
```

`summary.mc` *Summary Method for an Object of Class* `mc`

Description

Summary Method for an Object of Class `mc`

Usage

```
## S3 method for class 'mc'
summary(object, digits = 4, ...)
```

Arguments

`object`	Object of Class `mc`, that is, the output of the `MC()` function.
`digits`	Digits to print.
`...`	additional arguments.

Value

Returns a list with the following elements:

mean Mean of the sampling distribution of $\hat{\theta}$.

var Variance of the sampling distribution of $\hat{\theta}$.

bias Monte Carlo simulation bias.

rmse Monte Carlo simulation root mean square error.

location Location parameter used in the Monte Carlo simulation.

scale Scale parameter used in the Monte Carlo simulation.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
summary(mc)
```

---

vcov.betamc *Sampling Variance-Covariance Matrix Method for an Object of Class* betamc

---

Description

Sampling Variance-Covariance Matrix Method for an Object of Class `betamc`

Usage

```
## S3 method for class 'betamc'
vcov(object, ...)
```

Arguments

`object`	Object of Class `betamc`, that is, the output of the `BetaMC()`, `RSqMC()`, `SCorMC()`, `DeltaRSqMC()`, or `PCorMC()` functions.
`...`	additional arguments.

Value

Returns the variance-covariance matrix of estimates.

Author(s)

Ivan Jacob Agaloos Pesigan

Examples

```
# Fit the regression model
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
# Generate the sampling distribution of parameter estimates
# (use a large R, for example, R = 20000 for actual research)
mc <- MC(object, R = 100)
# Generate confidence intervals for standardized regression slopes
std <- BetaMC(mc)
# Method ----------------------------------------------------------
vcov(std)
```

Index