

# Package ‘betaMC’

April 14, 2024

**Title** Monte Carlo for Regression Effect Sizes

**Version** 1.3.2

**Description** Generates Monte Carlo confidence intervals for standardized regression coefficients (beta) and other effect sizes, including multiple correlation, semipartial correlations, improvement in R-squared, squared partial correlations, and differences in standardized regression coefficients, for models fitted by `lm()`.  
'betaMC' combines ideas from Monte Carlo confidence intervals for the indirect effect (Pesigan and Cheung, 2023 <[doi:10.3758/s13428-023-02114-4](https://doi.org/10.3758/s13428-023-02114-4)>) and the sampling covariance matrix of regression coefficients (Dudgeon, 2017 <[doi:10.1007/s11336-017-9563-z](https://doi.org/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

**Depends** R (>= 3.5.0)

**Imports** stats

**Suggests** knitr, rmarkdown, testthat, MASS, mice, Amelia

**RoxygenNote** 7.3.1

**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](mailto:r.jeksterslab@gmail.com)>

**Repository** CRAN

**Date/Publication** 2024-04-14 18:00:09 UTC

**R topics documented:**

BetaMC . . . . .	2
coef.betamc . . . . .	4
confint.betamc . . . . .	4
DeltaRSqMC . . . . .	5
DiffBetaMC . . . . .	6
MC . . . . .	8
MCMC . . . . .	10
nas1982 . . . . .	13
PCorMC . . . . .	14
print.betamc . . . . .	15
print.mc . . . . .	16
RSqMC . . . . .	17
SCorMC . . . . .	18
summary.betamc . . . . .	20
summary.mc . . . . .	20
vcov.betamc . . . . .	21
<b>Index</b>	<b>23</b>

---

BetaMC	<i>Estimate Standardized Regression Coefficients and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method</i>
--------	--

---

**Description**

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

**Usage**

```
BetaMC(object, alpha = c(0.05, 0.01, 0.001))
```

**Arguments**

object	Object of class mc, that is, the output of the MC() function.
alpha	Numeric vector. Significance level $\alpha$ .

**Details**

The vector of standardized regression coefficients ( $\hat{\beta}$ ) 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  $\hat{\beta}$ , where  $\alpha$  is the significance level.

**Value**

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

**call** Function call.

**args** Function arguments.

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

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

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

**fun** Function used ("BetaMC").

**Author(s)**

Ivan Jacob Agaloos Pesigan

**See Also**

Other Beta Monte Carlo Functions: [DeltaRSqMC\(\)](#), [DiffBetaMC\(\)](#), [MC\(\)](#), [MCMI\(\)](#), [PCorMC\(\)](#), [RSqMC\(\)](#), [SCorMC\(\)](#)

**Examples**

```
# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)

# BetaMC -----
out <- BetaMC(mc, alpha = 0.05)

## Methods -----
print(out)
summary(out)
coef(out)
vcov(out)
confint(out, 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(), PCorMC(), or DiffBetaMC() functions.

...                          additional arguments.

### Value

Returns a vector of estimated parameters.

### Author(s)

Ivan Jacob Agaloos Pesigan

---

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(), PCorMC(), or DiffBetaMC() 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

---

DeltaRSqMC	<i>Estimate Improvement in R-Squared and Generate the Corresponding Sampling Distribution Using the Monte Carlo Method</i>
------------	--

---

**Description**

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

**Usage**

```
DeltaRSqMC(object, alpha = c(0.05, 0.01, 0.001))
```

**Arguments**

object	Object of class mc, that is, the output of the MC() function.
alpha	Numeric vector. Significance level $\alpha$ .

**Details**

The vector of improvement in R-squared ( $\Delta R^2$ ) 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  $\Delta R^2$ , where  $\alpha$  is the significance level.

**Value**

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

- call** Function call.
- args** Function arguments.
- thetahatstar** Sampling distribution of  $\Delta R^2$ .
- vcov** Sampling variance-covariance matrix of  $\Delta R^2$ .
- est** Vector of estimated  $\Delta R^2$ .
- fun** Function used ("DeltaRSqMC").

**Author(s)**

Ivan Jacob Agaloos Pesigan

**See Also**

Other Beta Monte Carlo Functions: [BetaMC\(\)](#), [DiffBetaMC\(\)](#), [MC\(\)](#), [MCMC\(\)](#), [PCorMC\(\)](#), [RSqMC\(\)](#), [SCorMC\(\)](#)

**Examples**

```
# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)

# DeltaRSqMC -----
out <- DeltaRSqMC(mc, alpha = 0.05)

## Methods -----
print(out)
summary(out)
coef(out)
vcov(out)
confint(out, 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, alpha = c(0.05, 0.01, 0.001))
```

**Arguments**

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

**alpha**                     Numeric vector. Significance level  $\alpha$ .

## 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  $\alpha$  is the significance level.

## Value

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

**call** Function call.

**args** Function arguments.

**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\(\)](#), [MCMC\(\)](#), [PCorMC\(\)](#), [RSqMC\(\)](#), [SCorMC\(\)](#)

## Examples

```
# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)

# DiffBetaMC -----
out <- DiffBetaMC(mc, alpha = 0.05)

## Methods -----
print(out)
summary(out)
coef(out)
vcov(out)
```

```
confint(out, 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 <code>lm</code> .
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".
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

$$\boldsymbol{\theta} = \{\mathbf{b}, \sigma^2, \text{vech}(\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}})\}$$

where  $\mathbf{b}$  is the vector of regression slopes,  $\sigma^2$  is the error variance, and  $\text{vech}(\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}})$  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 of parameter estimates.

**fun** Function used ("MC").

### 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
- 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. doi:10.1207/s15327906mbr3901\_4
- Pesigan, I. J. A., & Cheung, S. F. (2023). Monte Carlo confidence intervals for the indirect effect with missing data. *Behavior Research Methods*. doi:10.3758/s13428023021144
- 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\(\)](#), [MCMC\(\)](#), [PCorMC\(\)](#), [RSqMC\(\)](#), [SCorMC\(\)](#)

**Examples**

```
# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)
mc
# The `mc` object can be passed as the first argument
# to the following functions
# - BetaMC
# - DeltaRSqMC
# - DiffBetaMC
# - PCorMC
# - RSqMC
# - SCorMC
```

MCMC

*Generate the Sampling Distribution of Regression Parameters Using  
the Monte Carlo Method for Data with Missing Values*

**Description**

Generate the Sampling Distribution of Regression Parameters Using the Monte Carlo Method for Data with Missing Values

**Usage**

```
MCMC(
  object,
  mi,
  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 <code>lm</code> .
mi	Object of class <code>mids</code> (output of <code>mice::mice()</code> ), object of class <code>amelia</code> (output of <code>Amelia::amelia()</code> ), or a list of multiply imputed data sets.
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". <code>type = "mvn"</code> uses the normal-theory sampling covariance matrix. <code>type = "adf"</code> uses the asymptotic distribution-free sampling covariance matrix. <code>type = "hc0"</code> through "hc5" uses different versions of heteroskedasticity-consistent sampling covariance matrix.
g1	Numeric. g1 value for <code>type = "hc4m"</code> .
g2	Numeric. g2 value for <code>type = "hc4m"</code> .
k	Numeric. Constant for <code>type = "hc5"</code> .
decomposition	Character string. Matrix decomposition of the sampling variance-covariance matrix for the data generation. If <code>decomposition = "chol"</code> , use Cholesky decomposition. If <code>decomposition = "eigen"</code> , use eigenvalue decomposition. If <code>decomposition = "svd"</code> , use singular value decomposition.
pd	Logical. If <code>pd = TRUE</code> , check if the sampling variance-covariance matrix is positive definite using <code>tol</code> .
tol	Numeric. Tolerance used for <code>pd</code> .
fixed_x	Logical. If <code>fixed_x = TRUE</code> , treat the regressors as fixed. If <code>fixed_x = FALSE</code> , treat the regressors as random.
seed	Integer. Seed number for reproducibility.

### Details

Multiple imputation is used to deal with missing values in a data set. The vector of parameter estimates and the corresponding sampling covariance matrix are estimated for each of the imputed data sets. Results are combined to arrive at the pooled vector of parameter estimates and the corresponding sampling covariance matrix. The pooled estimates are then used to generate the sampling distribution of regression parameters. See `MC()` for more details on the Monte Carlo method.

### 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 of parameter estimates.  
**fun** Function used ("MCMI").

### 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
- 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. doi:10.1207/s15327906mbr3901\_4
- Pesigan, I. J. A., & Cheung, S. F. (2023). Monte Carlo confidence intervals for the indirect effect with missing data. *Behavior Research Methods*. doi:10.3758/s13428023021144
- 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\(\)](#), [MC\(\)](#), [PCorMC\(\)](#), [RSqMC\(\)](#), [SCorMC\(\)](#)

### Examples

```
# Data -----
data("nas1982", package = "betaMC")
nas1982_missing <- mice::ampute(nas1982)$amp # data set with missing values

# Multiple Imputation
mi <- mice::mice(nas1982_missing, m = 5, seed = 42, print = FALSE)

# Fit Model in lm -----
## Note that this does not deal with missing values.
## The fitted model (`object`) is updated with each imputed data
## within the `MCMI()` function.
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982_missing)

# Monte Carlo -----
mc <- MCMI(
  object,
  mi = mi,
  R = 100, # use a large value e.g., 20000L for actual research
```

```
    seed = 0508
  )
mc
# The `mc` object can be passed as the first argument
# to the following functions
# - BetaMC
# - DeltaRSqMC
# - DiffBetaMC
# - PCorMC
# - RSqMC
# - SCorMC
```

---

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, alpha = c(0.05, 0.01, 0.001))
```

### Arguments

<code>object</code>	Object of class <code>mc</code> , that is, the output of the <code>MC()</code> function.
<code>alpha</code>	Numeric vector. Significance level $\alpha$ .

### Details

The vector of squared partial correlation coefficients ( $r_p^2$ ) 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  $r_p^2$ , where  $\alpha$  is the significance level.

### Value

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

- call** Function call.
- args** Function arguments.
- thetahatstar** Sampling distribution of  $r_p^2$ .
- vcov** Sampling variance-covariance matrix of  $r_p^2$ .
- est** Vector of estimated  $r_p^2$ .
- fun** Function used ("PCorMC").

### Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

Other Beta Monte Carlo Functions: [BetaMC\(\)](#), [DeltaRSqMC\(\)](#), [DiffBetaMC\(\)](#), [MC\(\)](#), [MCMI\(\)](#), [RSqMC\(\)](#), [SCorMC\(\)](#)

**Examples**

```

# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)

# PCorMC -----
out <- PCorMC(mc, alpha = 0.05)

## Methods -----
print(out)
summary(out)
coef(out)
vcov(out)
confint(out, 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 = NULL, digits = 4, ...)

```

**Arguments**

x	Object of Class betamc, that is, the output of the BetaMC(), RSqMC(), SCorMC(), DeltaRSqMC(), PCorMC(), or DiffBetaMC() functions.
alpha	Numeric vector. Significance level $\alpha$ . If alpha = NULL, use the argument alpha used in x.
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

---

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)  
print(mc)
```



---

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

---

**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, alpha = c(0.05, 0.01, 0.001))
```

**Arguments**

object	Object of class mc, that is, the output of the MC() function.
alpha	Numeric vector. Significance level $\alpha$ .

**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  $\alpha$  is the significance level.

**Value**

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

**call** Function call.

**args** Function arguments.

**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\(\)](#), [MCMI\(\)](#), [PCorMC\(\)](#), [SCorMC\(\)](#)

**Examples**

```

# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)

# RSqMC -----
out <- RSqMC(mc, alpha = 0.05)

## Methods -----
print(out)
summary(out)
coef(out)
vcov(out)
confint(out, 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, alpha = c(0.05, 0.01, 0.001))
```

**Arguments**

object	Object of class mc, that is, the output of the MC() function.
alpha	Numeric vector. Significance level $\alpha$ .

**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  $\alpha$  is the significance level.

**Value**

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

**call** Function call.

**args** Function arguments.

**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\(\)](#), [MCMC\(\)](#), [PCorMC\(\)](#), [RSqMC\(\)](#)

**Examples**

```
# Data -----
data("nas1982", package = "betaMC")

# Fit Model in lm -----
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)

# MC -----
mc <- MC(
  object,
  R = 100, # use a large value e.g., 20000L for actual research
  seed = 0508
)

# SCorMC -----
out <- SCorMC(mc, alpha = 0.05)

## Methods -----
print(out)
summary(out)
coef(out)
vcov(out)
confint(out, level = 0.95)
```

---

summary.betamc	<i>Summary Method for an Object of Class betamc</i>
----------------	---

---

**Description**

Summary Method for an Object of Class betamc

**Usage**

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

**Arguments**

object	Object of Class betamc, that is, the output of the BetaMC(), RSqMC(), SCorMC(), DeltaRSqMC(), PCorMC(), or DiffBetaMC() functions.
alpha	Numeric vector. Significance level $\alpha$ . If alpha = NULL, use the argument alpha used in object.
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

---

summary.mc	<i>Summary Method for an Object of Class mc</i>
------------	---

---

**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)
mc <- MC(object, R = 100)
summary(mc)
```

---

vcov.betamc	<i>Sampling Variance-Covariance Matrix Method for an Object of Class betamc</i>
-------------	---

---

**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(), PCorMC(), or DiffBetaMC() functions.  
 ... additional arguments.

**Value**

Returns the variance-covariance matrix of estimates.

**Author(s)**

Ivan Jacob Agaloos Pesigan

# Index

## \* Beta Monte Carlo Functions

BetaMC, 2  
DeltaRSqMC, 5  
DiffBetaMC, 6  
MC, 8  
MCMI, 10  
PCorMC, 14  
RSqMC, 17  
SCorMC, 18

## \* betaMC

BetaMC, 2  
DeltaRSqMC, 5  
DiffBetaMC, 6  
MC, 8  
MCMI, 10  
PCorMC, 14  
RSqMC, 17  
SCorMC, 18

## \* data

nas1982, 13

## \* deltarsq

DeltaRSqMC, 5

## \* diff

DiffBetaMC, 6

## \* mc

MC, 8  
MCMI, 10

## \* methods

coef.betamc, 4  
confint.betamc, 4  
print.betamc, 15  
print.mc, 16  
summary.betamc, 20  
summary.mc, 20  
vcov.betamc, 21

## \* pcor

PCorMC, 14

## \* rsq

RSqMC, 17

## \* scor

SCorMC, 18

## \* std

BetaMC, 2

Amelia::amelia(), 11

BetaMC, 2, 6, 7, 10, 12, 14, 17, 19

coef.betamc, 4

confint.betamc, 4

DeltaRSqMC, 3, 5, 7, 10, 12, 14, 17, 19

DiffBetaMC, 3, 6, 6, 10, 12, 14, 17, 19

MC, 3, 6, 7, 8, 12, 14, 17, 19

MCMI, 3, 6, 7, 10, 10, 14, 17, 19

mice::mice(), 11

nas1982, 13

PCorMC, 3, 6, 7, 10, 12, 14, 17, 19

print.betamc, 15

print.mc, 16

RSqMC, 3, 6, 7, 10, 12, 14, 17, 19

SCorMC, 3, 6, 7, 10, 12, 14, 17, 18

summary.betamc, 20

summary.mc, 20

vcov.betamc, 21