Package ‘mcmcse’

October 13, 2022

Version 1.5-0
Date 2021-08-29
Title Monte Carlo Standard Errors for MCMC
Author James M. Flegal <jflegal@ucr.edu>,
John Hughes,
Dootika Vats <doottika@iitk.ac.in>,
Ning Dai,
Kushagra Gupta, and
Uttiya Maji
Maintainer Dootika Vats <doottika@iitk.ac.in>
Imports utils, ellipse, Rcpp (>= 0.12.10), fftwtools, testthat
LinkingTo Rcpp, RcppArmadillo
Suggests knitr
Description Provides tools for computing Monte Carlo standard
errors (MCSE) in Markov chain Monte Carlo (MCMC) settings. MCSE
computation for expectation and quantile estimators is
supported as well as multivariate estimations. The package also provides
functions for computing effective sample size and for plotting
Monte Carlo estimates versus sample size.
License GPL (>= 2)
VignetteBuilder knitr
Repository CRAN
Encoding UTF-8
RoxygenNote 7.1.1
NeedsCompilation yes
Date/Publication 2021-09-09 07:40:02 UTC

R topics documented:

mcmcse-package .......................................................... 2
batchSize ............................................................... 4
mcmcse-package

BVN_Gibbs ......................................................... 5
confRegion ......................................................... 6
ess ................................................................. 7
estvssamp .......................................................... 8
is.mcmcse .......................................................... 8
mcse .............................................................. 9
mcse.initseq ....................................................... 11
mcse.mat .......................................................... 12
mcse.multi ......................................................... 13
mcse.q ............................................................ 15
mcse.q.mat ........................................................ 17
minESS ............................................................ 18
multiESS ........................................................... 19

Index 21

mcmcse-package  Monte Carlo Standard Errors for MCMC

Description

Provides tools for computing Monte Carlo standard errors (MCSE) in Markov chain Monte Carlo (MCMC) settings. MCSE computation for expectation and quantile estimators is supported. The package also provides functions for computing effective sample size and for plotting Monte Carlo estimates versus sample size.

Details

Package: mcmcse
Type: Package
Version: 1.5-0
Date: 2021-08-29
License: GPL (>= 2)

Author(s)

James M. Flegal <jflegal@ucr.edu>,
John Hughes,
Dootika Vats, <dootika@iitk.ac.in>,
Ning Dai,
Kushagra Gupta, and
Uttiya Maji

Maintainer: Dootika Vats <dootika@iitk.ac.in>
References


Examples

```r
n <- 1e3
mu = c(2, 50)
sigma = matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out = BVN_Gibbs(n, mu, sigma)
multiESS(out)
ess(out)
mcse.mat(out)
mcse.bm <- mcse.multi(x = out)
mcse.tuk <- mcse.multi(x = out, method = "tukey")
```
**batchSize**

**Batch size (truncation point) selection**

**Description**

Function returns the optimal batch size (or truncation point) for a given chain and method.

**Usage**

```r
batchSize(x, method = c("bm", "obm", "bartlett", "tukey", "sub"), g = NULL, fast = TRUE)
```

**Arguments**

- `x` A matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.
- `method` Any of "bm", "obm", "bartlett", "tukey", "sub". "bm" represents batch means estimator, "obm" represents the overlapping batch means estimator, and "bartlett" and "tukey" represent the modified-Bartlett window and the Tukey-Hanning windows for the spectral variance estimators.
- `g` A function that represents features of interest. `g` is applied to each row of `x` and thus `g` should take a vector input only. If `g` is `NULL`, `g` is set to be identity, which is estimation of the mean of the target density.
- `fast` Boolean variable for fast estimation using a reasonable subset of the Markov chain.

**Details**

`batchSize` fits a stationary autoregressive process to the marginals of `x`, selecting the order of the process as the one with the maximum AIC among the models with coefficients greater than a threshold.

**Value**

A value of the optimal batch size (truncation point or bandwidth) is returned.

**References**


**See Also**

`mcse.multi` and `mcse`, which calls on `batchSize`. 
Examples

```r
# Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)

out <- BVN_Gibbs(n, mu, sigma)

batchSize(out)
batchSize(out, method = "obm")
batchSize(out, method = "bartlett")
```

---

**BVN_Gibbs**

*MCMC samples from a bivariate normal distribution*

**Description**

Function returns Gibbs samples from a bivariate normal target density.

**Usage**

```r
BVN_Gibbs(n, mu, sigma)
```

**Arguments**

- `n`: Sample size of the Markov chain.
- `mu`: A 2 dimensional vector. Mean of the target normal distribution.
- `sigma`: 2 x 2 symmetric positive definite matrix. The covariance matrix of the target normal distribution.

**Value**

An `n x 2` matrix of the Gibbs samples.

**Examples**

```r
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out <- BVN_Gibbs(n, mu, sigma)
```
Description

Constructs confidence regions (ellipses) from the Markov chain output for the features of interest. Function uses the ellipse package.

Usage

confRegion(mcse.obj, which = c(1, 2), level = .95)

Arguments

- mcse.obj: The list returned by the mcse.multi or mcse.initseq command, or an mcmcse class object.
- which: Integer vector of length 2 indicating the component for which to make the confidence ellipse. Chooses the first two by default.
- level: confidence level for the ellipse.

Value

Returns a matrix of x and y coordinates for the ellipse. Use plot function on the matrix to plot the ellipse.

Examples

## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)

out <- BVN_Gibbs(n, mu, sigma)

mcerror <- mcse.multi(out, blather = TRUE)

## Plotting the ellipse
plot(confRegion(mcerror), type = 'l')
Univariate effective sample size (ESS) as described in Gong and Flegal (2015).

Description
Estimate the effective sample size (ESS) of a Markov chain as described in Gong and Flegal (2015).

Usage
ess(x, g = NULL, ...)

Arguments
x a matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.
g a function that represents features of interest. g is applied to each row of x and thus g should take a vector input only. If code is NULL, g is set to be identity, which is estimation of the mean of the target density.
... arguments passed on to the mcse.mat function. For example method = “tukey” and size = “cuberoot” can be used.

Details
ESS is the size of an iid sample with the same variance as the current sample for estimating the expectation of g. ESS is given by

\[ ESS = \frac{n \lambda^2}{\sigma^2} \]

where \( \lambda^2 \) is the sample variance and \( \sigma^2 \) is an estimate of the variance in the Markov chain central limit theorem. The denominator by default is a batch means estimator, but the default can be changed with the ‘method’ argument.

Value
The function returns the estimated effective sample size for each component of g.

References

See Also
minESS, which calculates the minimum effective samples required for the problem. multiESS, which calculates multivariate effective sample size using a Markov chain and a function g.
estvssamp  

Create a plot that shows how Monte Carlo estimates change with increasing sample size.

Description

Create a plot that shows how Monte Carlo estimates change with increasing sample size.

Usage

estvssamp(x, g = mean, main = "Estimates vs Sample Size", add = FALSE, ...)

Arguments

x  
a sample vector.

 g  
a function such that $E(g(x))$ is the quantity of interest. The default is $g = \text{mean}$.

 main  
an overall title for the plot. The default is “Estimates vs Sample Size”.

 add  
logical. If TRUE, add to a current plot.

 ...  
additional arguments to the plotting function.

Value

NULL

Examples

## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out <- BVN_Gibbs(n, mu, sigma)
estvssamp(out[,1], main = expression(E(x[1])))

is.mcmcse  

Check if the class of the object is mcmcse

Description

Check if the class of the object is mcmcse

Usage

is.mcmcse(x)
Arguments

\( x \)  
The object that is checked to belong to the class mcmcse

Value

Boolean variable indicating if the input belongs to the class mcmcse

Examples

```r
## Bivariate Normal with mean \((\mu_1, \mu_2)\) and covariance \(\Sigma\)

n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)

out <- BVN_Gibbs(n, mu, sigma)
is.mcmcse(mcse.multi(out))
```

mcse

**Compute Monte Carlo standard errors for expectations.**

Description

Compute Monte Carlo standard errors for expectations.

Usage

`mcse(x, size = NULL, g = NULL, r = 3, method = "bm", warn = FALSE)`

Arguments

\( x \)

a vector of values from a Markov chain of length \(n\).

\( \text{size} \)

represents the batch size in “bm” and the truncation point in “bartlett” and “tukey”. Default is `NULL` which implies that an optimal batch size is calculated using the `batchSize` function. Can take character values of “sqroot” and “cuberoot” or any numeric value between 1 and \(n/2\). “sqroot” means size is \( \lfloor \sqrt{n} \rfloor \) and “cuberoot” means size is \( \lfloor \sqrt[3]{n} \rfloor \).

\( g \)

a function such that \( E(g(x)) \) is the quantity of interest. The default is `NULL`, which causes the identity function to be used.

\( r \)

The lugsail parameters \( r \) that converts a lag window into its lugsail equivalent. Larger values of \( r \) will typically imply less underestimation of “cov”, but higher variability of the estimator. Default is \( r = 3 \) and \( r = 1, 2 \) are also good choices, but will likely underestimation of variance. \( r > 5 \) is not recommended.

warn: a logical value indicating whether the function should issue a warning if the sample size is too small (less than 1,000).

**Value**

mcse returns a list with three elements:

- est: an estimate of $E(g(x))$.
- se: the Monte Carlo standard error.
- nsim: The number of samples in the input Markov chain.

**References**


**See Also**

- mcse.mat, which applies mcse to each column of a matrix or data frame.
- mcse.multi, for a multivariate estimate of the Monte Carlo standard error.
- mcse.q and mcse.q.mat, which compute standard errors for quantiles.

**Examples**

```r
## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu = c(2, 50)
sigma = matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out = BVN_Gibbs(n, mu, sigma)
x = out[,1]
mcse(x)
mcse.q(x, 0.1)
mcse.q(x, 0.9)
```
mcse.initseq

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using overlapping batch means.
mcse(x, method = "obm")
mcse.q(x, 0.1, method = "obm")
mcse.q(x, 0.9, method = "obm")

# Estimate E(x^2) with MCSE using spectral methods.
g = function(x) { x^2 }
mcse(x, g = g, method = "tukey")

mcse.initseq  Multivariate Monte Carlo standard errors for expectations with the initial sequence method of Dai and Jones (2017)

Description

Function returns the estimate of the covariance matrix in the Markov Chain central limit theorem using initial sequence method. This method is designed to give an asymptotically conservative estimate of the Monte Carlo standard error.

Usage

mcse.initseq(x, g = NULL, adjust = FALSE, blather = FALSE)

Arguments

x  A matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.
g  A function that represents features of interest. g is applied to each row of x and thus g should take a vector input only. If g is NULL, g is set to be identity, which is estimation of the mean of the target density.
adjust  Logical; if TRUE, an adjustment is made to increase slightly the eigenvalues of the initial sequence estimator. The default is FALSE.
blather  if TRUE, outputs under the hood information about the function.

Value

A list is returned with the following components,
cov  a covariance matrix estimate using initial sequence method.
cov.adj  a covariance matrix estimate using adjusted initial sequence method if the input adjust=TRUE.
eigen_values  eigen values of the estimate cov.
method  method used to calculate matrix cov.
estimate of \( g(x) \).

\texttt{nsim}  
number of rows of the input \texttt{x}. Only if \texttt{blather = TRUE}.

\texttt{Adjustment\_Used}  
logical of whether an adjustment was made to the initial sequence estimator. Only if \texttt{blather = TRUE}.

References

See Also
\texttt{mcse}, which acts on a vector. \texttt{mcse.mat}, which applies \texttt{mcse} to each column of a matrix or data frame. \texttt{mcse.q} and \texttt{mcse.q.mat}, which compute standard errors for quantiles. \texttt{mcse.multi}, which estimates the covariance matrix in the Markov Chain CLT using batch means or spectral variance methods.

Examples

```r
## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out <- BVN_Gibbs(n, mu, sigma)

out.mcse <- mcse.initseq(x = out)
out.mcse.adj <- mcse.initseq(x = out, adjust = TRUE)

# If we are only estimating the mean of the first component, # and the second moment of the second component
g <- function(x) return(c(x[1], x[2]^2))
out.g.mcse <- mcse.initseq(x = out, g = g)
```

\texttt{mcse.mat}  
Apply \texttt{mcse} to each column of the MCMC samples.

Description
Apply \texttt{mcse} to each column of the MCMC samples.

Usage

\texttt{mcse.mat(x, size = NULL, g = NULL, method = “bm”, r = 3)}
mcse.multi

Arguments

x: a matrix of values from a Markov chain of size n x p.

size: represents the batch size in “bm” and the truncation point in “bartlett” and “tukey”. Default is NULL which implies that an optimal batch size is calculated using the batchSize function. Can take character values of “sqroot” and “cuberoot” or any numeric value between 1 and n/2. “sqroot” means size is \( \lfloor n^{1/2} \rfloor \) and “cuberoot” means size is \( \lfloor n^{1/3} \rfloor \).

method: the lag window to be used. May take any of “bm”, “obm”, “bartlett”, “tukey”. “bm” represents batch means estimator, “obm” represents overlapping batch means estimator with, “bartlett” and “tukey” represents the modified-Bartlett window and the Tukey-Hanning windows for spectral variance estimators.

r: The lugsail parameters (r) that converts a lag window into its lugsail equivalent. Larger values of r will typically imply less underestimation of “cov”, but higher variability of the estimator. Default is r = 3 and r = 1, 2 are also good choices although may lead to underestimates of the variance. r > 5 is not recommended.

g: a function such that \( E(g(x)) \) is the quantity of interest. The default is NULL, which causes the identity function to be used.

Value

mcse.mat returns a matrix with ncol(x) rows and two columns. The row names of the matrix are the same as the column names of x. The column names of the matrix are “est” and “se”. The jth row of the matrix contains the result of applying mcse to the jth column of x.

See Also

mcse, which acts on a vector.
mcse.multi, for a multivariate estimate of the Monte Carlo standard error.
mcse.q and mcse.q.mat, which compute standard errors for quantiles.

Description

Function returns the estimate of the covariance matrix in the Markov Chain CLT using batch means or spectral variance methods (with different lag windows). The function also returns the Monte Carlo estimate.

Usage

mcse.multi(x, method = "bm", r = 3, size = NULL, g = NULL, adjust = TRUE, blather = FALSE)
Arguments

- **x**: A matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.

- **method**: Any of "bm", "obm", "bartlett", "tukey". "bm" represents batch means estimator, "obm" represents the overlapping batch means estimator, and "bartlett" and "tukey" represent the modified-Bartlett window and the Tukey-Hanning windows for the spectral variance estimators.

- **r**: The lugsail parameters (r) that converts a lag window into its lugsail equivalent. Larger values of r will typically imply less underestimation of "cov", but higher variability of the estimator. Default is r = 3 and r = 1, 2 are good choices. r > 5 is not recommended.

- **size**: Represents the batch size in "bm" and the truncation point in "bartlett" and "tukey". Default is NULL which implies that an optimal batch size is calculated using the batchSize function. Can take character values of "sqroot" and "cuberoot" or any numeric value between 1 and n/2. "sqroot" means size is floor(n^(1/2)) and "cuberoot" means size is floor(n^(1/3)).

- **g**: A function that represents features of interest. g is applied to each row of x and thus g should take a vector input only. If g is NULL, g is set to be identity, which is estimation of the mean of the target density.

- **adjust**: Defaults to TRUE. logical for whether the matrix should automatically be adjusted if unstable.

- **blather**: If TRUE, returns under-the-hood workings of the package.

Value

A list is returned with the following components,

- **cov**: a covariance matrix estimate.
- **est**: estimate of g(x).
- **nsim**: number of rows of the input x.
- **eigen_values**: eigen values of the estimate cov.
- **method**: method used to calculate matrix cov.
- **size**: value of size used to calculate cov.
- **Adjustment_Used**: whether an adjustment was used to calculate cov.

References


See Also

`batchSize`, which computes an optimal batch size. `mcse.initseq`, which computes an initial sequence estimator. `mcse`, which acts on a vector. `mcse.mat`, which applies `mcse` to each column of a matrix or data frame. `mcse.q` and `mcse.q.mat`, which compute standard errors for quantiles.

Examples

```r
## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out <- BVN_Gibbs(n, mu, sigma)

mcse.bm <- mcse.multi(x = out)
mcse.tuk <- mcse.multi(x = out, method = "tukey")

# If we are only estimating the mean of the first component, 
# and the second moment of the second component

g <- function(x) return(c(x[1], x[2]^2))
mcse <- mcse.multi(x = out, g = g)
```

---

**mcse.q**

Compute Monte Carlo standard errors for quantiles.

**Description**

Compute Monte Carlo standard errors for quantiles.

**Usage**

```r
mcse.q(x, q, size = NULL, g = NULL, method = c("bm", "obm", "sub"), warn = FALSE)
```

**Arguments**

- `x`: a vector of values from a Markov chain.
- `q`: the quantile of interest.
- `size`: the batch size. The default value is "sqrt", which uses the square root of the sample size. A numeric value may be provided if "sqrt" is not satisfactory.
- `g`: a function such that the qth quantile of the univariate distribution function of \( g(x) \) is the quantity of interest. The default is `NULL`, which causes the identity function to be used.
- `method`: the method used to compute the standard error. This is one of "bm" (batch means, the default), "obm" (overlapping batch means), or "sub" (subsampling bootstrap).
- `warn`: a logical value indicating whether the function should issue a warning if the sample size is too small (less than 1,000).
mcse.q

Value

mcse.q returns a list with three elements:

- `est` an estimate of the $q$th quantile of the univariate distribution function of $g(x)$.
- `se` the Monte Carlo standard error.
- `nsim` The number of samples in the input Markov chain.

References


See Also

- `mcse.q.mat`, which applies `mcse.q` to each column of a matrix or data frame.
- `mcse` and `mcse.mat`, which compute standard errors for expectations.

Examples

```r
## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out <- BVN_Gibbs(n, mu, sigma)
x <- out[,1]

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using batch means.
mcse(x)
mcse.q(x, 0.1)
mcse.q(x, 0.9)

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using overlapping batch means.
mcse(x, method = "obm")
mcse.q(x, 0.1, method = "obm")
mcse.q(x, 0.9, method = "obm")
```
# Estimate E(x^2) with MCSE using spectral methods.

g <- function(x) { x^2 }
misce(x, g = g, method = "tukey")

---

mcse.q.mat

Apply mcse.q to each column of a matrix or data frame of MCMC samples.

Description

Apply mcse.q to each column of a matrix or data frame of MCMC samples.

Usage

mcse.q.mat(x, q, size = NULL, g = NULL, method = c("bm", "obm", "sub"))

Arguments

x a matrix or data frame with each row being a draw from the multivariate distribution of interest.
q the quantile of interest.
size the batch size. The default value is "sqroot", which uses the square root of the sample size. "cuberoot" will cause the function to use the cube root of the sample size. A numeric value may be provided if "sqroot" is not satisfactory.
g a function such that the qth quantile of the univariate distribution function of g(x) is the quantity of interest. The default is NULL, which causes the identity function to be used.
method the method used to compute the standard error. This is one of "bm" (batch means, the default), "obm" (overlapping batch means), or "sub" (subsampling bootstrap).

Value

mcse.q.mat returns a matrix with ncol(x) rows and two columns. The row names of the matrix are the same as the column names of x. The column names of the matrix are “est” and “se”. The jth row of the matrix contains the result of applying mcse.q to the jth column of x.

See Also

mcse.q, which acts on a vector.
misce and mcse.mat, which compute standard errors for expectations.
Description

The function calculates the minimum effective sample size required for a specified relative tolerance level. This function can also calculate the relative precision in estimation for a given estimated effective sample size.

Usage

```r
minESS(p, alpha = .05, eps = .05, ess = NULL)
```

Arguments

- `p` dimension of the estimation problem.
- `alpha` Confidence level.
- `eps` Tolerance level. The `eps` value is ignored if `ess` is not `NULL`.
- `ess` Estimated effective sample size. Usually the output value from `multiESS`.

Details

The minimum effective samples required when estimating a vector of length \( p \), with \( 100(1 - \alpha)\% \) confidence and tolerance of \( \epsilon \) is

\[
mESS \geq \frac{2^{2/p} \pi}{\left(p \Gamma(p/2) \right)^{2/p}} \frac{\chi^2_{1 - \alpha, p} \epsilon^2}{\epsilon^2}.
\]

The above equality can also be used to get \( \epsilon \) from an already obtained estimate of \( mESS \).

Value

By default function returns the minimum effective sample required for a given `eps` tolerance. If `ess` is specified, then the value returned is the `eps` corresponding to that `ess`.

References


See Also

`multiESS`, which calculates multivariate effective sample size using a Markov chain and a function `g`. `ess` which calculates univariate effective sample size using a Markov chain and a function `g`. 
**multiESS**

**Examples**

\[
\text{minESS}(p = 5)
\]

---

**multiESS**

*Effective Sample Size of a multivariate Markov chain as described in Vats et al. (2015).*

---

**Description**

Calculate the effective sample size of the Markov chain, using the multivariate dependence structure of the process.

**Usage**

\[
\text{multiESS}(x, \text{covmat} = \text{NULL}, g = \text{NULL}, \ldots)
\]

**Arguments**

- **x**: a matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.
- **covmat**: optional matrix estimate obtained using \text{mcse.multi} or \text{mcse.initseq}.
- **g**: a function that represents features of interest. \(g\) is applied to each row of \(x\) and thus \(g\) should take a vector input only. If \(g\) is \text{NULL}, \(g\) is set to be identity, which is estimation of the mean of the target density.
- **\ldots**: arguments for \text{mcse.multi} function. Don’t use this if a suitable matrix estimate from \text{mcse.multi} or \text{mcse.initseq} is already obtained.

**Details**

Effective sample size is the size of an iid sample with the same variance as the current sample. ESS is given by

\[
\text{ESS} = n \frac{|\Lambda|^{1/p}}{|\Sigma|^{1/p}},
\]

where \(\Lambda\) is the sample covariance matrix for \(g\) and \(\Sigma\) is an estimate of the Monte Carlo standard error for \(g\).

**Value**

The function returns the estimated effective sample size.

**References**

See Also

`minESS`, which calculates the minimum effective samples required for the problem. `ess` which calculates univariate effective sample size using a Markov chain and a function g.

Examples

```r
## Bivariate Normal with mean (mu1, mu2) and covariance sigma
n <- 1e3
mu <- c(2, 50)
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
out <- BVN_Gibbs(n, mu, sigma)

multiESS(out)
```
Index

batchSize, 4, 15
BVN_Gibbs, 5

confRegion, 6

ess, 7, 18, 20
estvssamp, 8

is.mcmcse, 8

mcmcse (mcmcse-package), 2
mcmcse-package, 2
mcse, 4, 9, 12, 13, 15–17
mcse.initseq, 11, 15
mcse.mat, 10, 12, 12, 15–17
mcse.multi, 4, 10, 13, 13
mcse.q, 10, 12, 13, 15, 15, 17
mcse.q.mat, 10, 12, 13, 15, 16, 17
mean, 8

minESS, 7, 18, 20
multiESS, 7, 18, 19