# Package ‘mcmcse’

January 29, 2020

**Version** 1.4-1  
**Date** 2020-01-29  
**Title** Monte Carlo Standard Errors for MCMC  
**Author** James M. Flegal <jflegal@ucr.edu>,  
John Hughes <j.hughes@ucdenver.edu>,  
Dootika Vats <dootika@iitk.ac.in>, and  
Ning Dai <daixx224@umn.edu>  
**Maintainer** Dootika Vats <dootika@iitk.ac.in>  
**Imports** utils, ellipse, Rcpp (>= 0.12.10)  
**LinkingTo** Rcpp, RcppArmadillo  
**Suggests** mAr, 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)  
**URL** [http://faculty.ucr.edu/~jflegal](http://faculty.ucr.edu/~jflegal),  
[http://johnhughes.org](http://johnhughes.org),  
[http://home.iitk.ac.in/~dootika/](http://home.iitk.ac.in/~dootika/)  
**VignetteBuilder** knitr  
**Repository** CRAN  
**Encoding** UTF-8  
**NeedsCompilation** yes  
**Date/Publication** 2020-01-29 12:40:02 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>mcmcse-package</td>
<td>2</td>
</tr>
<tr>
<td>batchSize</td>
<td>4</td>
</tr>
</tbody>
</table>
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: 1.4-1
Date: 2020-01-29
License: GPL (>= 2)

Author(s)

James M. Flegal <jflegal@ucr.edu>, John Hughes <j.hughes@ucdenver.edu>, Dootika Vats <dootika@iitk.ac.in> and Ning Dai <daixx224@umn.edu>

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

References

Examples

```r
library(mAr)
p <- 3
n <- 1e3
omega <- 5*diag(1,p)

## Making correlation matrix var(1) model
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)
out <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))
mcse(out[,1], method = "bartlett")
mcse.bm <- mcse.multi(x = out)
mcse.tuk <- mcse.multi(x = out, method = "tukey")
```
Description
Function returns the optimal batch size (or truncation point) for a given chain and method.

Usage
batchSize(x, method = "bm", g = NULL)

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.
- 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.

Value
A value of the optimal batch size is returned.

References

See Also
mcse.multi, which calls on batchSize. mcse, which calls on batchSize.

Examples
library(mAr)
p <- 3
n <- 1e3
omega <- 5*diag(1,p)

## Making correlation matrix var(1) model
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)
out <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))

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

confRegion

Confidence regions (ellipses) for Monte Carlo estimates

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
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

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

Examples
library(mAr)
p <- 3
n <- 1e3
omega <- 5*diag(1,p)

## Making correlation matrix var(1) model
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)

out <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))
merror <- mcse.multi(out, blather = TRUE)

## Plotting the ellipse
plot(confRegion(merror), type = 'l')
ess

Univariate estimate effective sample size (ESS) as described in Gong and Flegal (2015).

Description

Estimate effective sample size (ESS) 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.
- `...` arguments passed on to the `mcse.mat` function. For example `method = "tukey"` and `size = "cuberoor"` can be used.
- `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.

Details

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

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

where \(\lambda^2\) is the sample variance and \(\sigma^2\) is an estimate of the variance in the CLT. This is by default the lugsail batch means estimator, but the default can be changed with the `method` argument.

Value

The function returns the estimated effective sample size.

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

## Not run:
estvssamp(x, main = expression(E(beta)))
estvssamp(y, add = TRUE, lty = 2, col = "red")
## End(Not run)

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.

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 \textquoteright \textquotesingle sqroot\textquoteright and \textquoteright \textquotesingle cuberoot\textquoteright or any numeric value between 1 and n/2. \textquoteright \textquotesingle sqroot\textquoteright means size is floor(n^{1/2}) and \textquoteright \textquotesingle cuberoot\textquoteright means size is floor(n^{1/3}).

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.

method any of \textquoteright \textquotesingle bm\textquoteright, \textquoteright \textquotesingle obm\textquoteright, \textquoteright \textquotesingle bartlett\textquoteright, \textquoteright \textquotesingle tukey\textquoteright. \textquoteright \textquotesingle bm\textquoteright represents batch means estimator, \textquoteright \textquotesingle obm\textquoteright represents overlapping batch means estimator with, \textquoteright \textquotesingle bartlett\textquoteright and \textquoteright \textquotesingle tukey\textquoteright represents the modified-Bartlett window and the Tukey-Hanning windows for spectral variance estimators.

r the lugsail parameter that converts a lag window into its lugsail equivalent. Larger values of \textquoteright \textquotesingle r\textquoteright will typically imply less underestimation of \textquoteright \textquotesingle cov\textquoteright, but higher variability of the estimator. Default is \textquoteright \textquotesingle r = 3\textquoteright and \textquoteright \textquotesingle r = 1,2\textquoteright are good choices. \textquoteright \textquotesingle r > 5\textquoteright is not recommended. Non-integer values are ok.

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 two elements:

est an estimate of \( E(g(x)) \).

se the Monte Carlo standard error.

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

# Create 10,000 iterations of an AR(1) Markov chain with rho = 0.9.

n = 10000
x = double(n)
x[1] = 2
for (i in 1:(n - 1))
  x[i + 1] = 0.9 * x[i] + rnorm(1)

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using batch means.

mcse(x)
mceu.q(x, 0.1)
mceu.q(x, 0.9)

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using overlapping batch means.

mcse(x, method = "obm")
mceu.q(x, 0.1, method = "obm")

# Estimate E(x^2) with MCSE using spectral methods.

g = function(x) { x^2 }
mceu(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)
mcse.initseq

Arguments

\(x\) a matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.

\(\text{adjust}\) logical; if TRUE, an adjustment is made to increase slightly the eigenvalues of the initial sequence estimator. The default is FALSE.

\(g\) 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.

\(\text{blather}\) if TRUE, outputs under the hood information about the function.

Value

A list is returned with the following components,

\(\text{cov}\) a covariance matrix estimate using initial sequence method.

\(\text{cov.adj}\) a covariance matrix estimate using adjusted initial sequence method if the input \(\text{adjust} = \text{TRUE}\).

\(\text{est}\) estimate of \(g(x)\).

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

\(\text{adjust}\) logical of whether an adjustment was made to the initial sequence estimator. Only if \(\text{blather} = \text{TRUE}\)

References


See Also

\text{initseq(mcmc)}, which is a different univariate initial sequence estimator. \text{mcse}, which acts on a vector. \text{mcse.mat}, which applies \text{mcse} to each column of a matrix or data frame. \text{mcse.q} and \text{mcse.q.mat}, which compute standard errors for quantiles. \text{mcse.multi}, which estimates the covariance matrix in the Markov Chain CLT using batch means or spectral variance methods.

Examples

```r
library(mAr)
p <- 3
n <- 1000
omega <- 5*diag(1,p)

## Making correlation matrix var(1) model
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)
dat <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))
```
mcse.mat

out.mcse <- mcse.initseq(x = dat)
out.mcse.adj <- mcse.initseq(x = dat, 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 = dat, g = g)

mcse.mat

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

Description

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

Usage

mcse.mat(x, size = NULL, g = NULL, method = "bm", r = 3)

Arguments

x
a matrix or data frame with each row being a draw from the multivariate distribution of interest.

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

method
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 parameter 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. Non-integer values are ok.

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.

---

**mcse.multi**

*Multivariate Monte Carlo standard errors for expectations.*

**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 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 parameter 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. Non-integer values are ok.
- `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`.
- `method`: method used to calculate matrix `cov`.
- `size`: value of size used to calculate `cov`.
- `adjust.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
library(mAr)
p <- 3
n <- 1e3
omega <- 5*diag(1,p)

## Making correlation matrix var(1) model
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)

out <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))
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

mcse.q(x, q, size = "sqroot", 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 "sqroot", which uses the square 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).
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.q returns a list with two elements:
est
an estimate of the qth quantile of the univariate distribution function of g(x).
se
the Monte Carlo standard error.

References

mcse.q.mat

University of California, Riverside, Technical Report.


See Also

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

Examples

# Create 10,000 iterations of an AR(1) Markov chain with rho = 0.9.

n = 10000
x = double(n)
x[1] = 2
for (i in 1:(n - 1))
  x[i + 1] = 0.9 * x[i] + rnorm(1)

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using batch means.

mcse(x)
misce.q(x, 0.1)
misce.q(x, 0.9)

# Estimate the mean, 0.1 quantile, and 0.9 quantile with MCSEs using overlapping batch means.

mcse(x, method = "obm")
misce.q(x, 0.1, method = "obm")
misce.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(x, q, size = "sqroot", g = NULL,
method = c("bm", "obm", "sub"))

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 = "sqroot", g = NULL,
method = c("bm", "obm", "sub"))
Arguments

- **x**: a matrix or data frame of Markov chain output. Number of rows is the Monte Carlo sample size.
- **q**: the quantile of interest.
- **size**: the batch size. The default value is “sqroot”, which uses the square root of the sample size. “cubroot” 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 q-th 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 j-th row of the matrix contains the result of applying mcse.q to the j-th column of x.

See Also

- **mcse.q**, which acts on a vector.
- **mcse** 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

```
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 is ess is not NULL
- **ess**: Estimated effective sample size. Usually the output value from multiESS.
The minimum effective samples required when estimating a vector of length $p$, with $100(1-\alpha)\%$ confidence and tolerance of $\epsilon$ is

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

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 $\epsilon$ tolerance. If $ess$ is specified, then the value returned is the $\epsilon$ 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 $g$.

**Examples**

```r
minESS(p = 5)
```

---

**Description**

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

**Usage**

```r
multiESS(x, covmat = NULL, g = NULL, ...)
```
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 `mcse.multi` or `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 `NULL`, `g` is set to be identity, which is estimation of the mean of the target density.
- ... Arguments for `mcse.multi` function. Don’t use this if a suitable matrix estimate from `mcse.multi` or `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

\[
\text{ESS} = n \left( \frac{\Lambda^{1/p}}{\Sigma^{1/p}} \right)
\]

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
library(mAr)
p <- 3
n <- 1e3
omega <- 5*diag(1,p)
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)
out <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))
multiESS(out)
```
Description

QQplot for Markov chains using an estimate of the Markov Chain CLT covariance matrix.

Usage

qqTest(mcse.obj)

Arguments

mcse.obj the list returned by the mcse.multi or mcse.initseq command

Examples

library(mAr)
p <- 35
n <- 1e4
omega <- 5*diag(1,p)

## Making correlation matrix var(1) model
set.seed(100)
foo <- matrix(rnorm(p^2), nrow = p)
foo <- foo %*% t(foo)
phi <- foo / (max(eigen(foo)$values) + 1)

out <- as.matrix(mAr.sim(rep(0,p), phi, omega, N = n))

mcse.bm <- mcse.multi(x = out)
qqTest(mcse.bm)

mcse.isadj <- mcse.initseq(x = out, adjust = TRUE)
qqTest(mcse.isadj)
Index

batchSize, 4, 13
confRegion, 5

ess, 6, 17, 18
estvssamp, 7

mcmcse (mcmcse-package), 2
mcmcse-package, 2
mcse, 4, 7, 10, 12, 13, 15, 16
mcse.initseq, 9, 13
mcse.mat, 9, 10, 11, 13, 15, 16
mcse.multi, 4, 9, 10, 12, 12
mcse.q, 9, 10, 12, 13, 14, 16
mcse.q.mat, 9, 10, 12, 13, 15, 15
mean, 7
minESS, 6, 16, 18
multiESS, 6, 17, 17

qqTest, 19