

Package ‘SimTools’

August 23, 2023

Version 1.0-0

Date 2021-02-24

Title Toolkit for Simulation Output Including Monte Carlo and MCMC

Author Dootika Vats <dootika@iitk.ac.in>,
James M. Flegal <jflegal@ucr.edu>,
Galin Jones <galin@umn.edu>,
Gunjan Jalori.

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

Imports mcmcse

Description

Toolkit for simulation output including Monte Carlo and Markov chain Monte Carlo. Tools for reliable visualisations are available, and support for multiple chain MCMC is integrated.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

R topics documented:

acfplot	2
addCI	3
boxCI	4
boxplot.Siid	4
convert2Smcmc	5
densityplot	6
getCI	7
plot.Siid	8
plot.Smcmc	9
Siid	10
Smcmc	10
summary.Smcmc	11
traceplot	12

Index	14
--------------	-----------

acfplot

*ACF Plot for Markov chain Monte Carlo***Description**

Autocorrelation function plots for MCMC data (including multiple chains)

Usage

```
acfplot(x, which = NULL, type = c("correlation"),
        plot = TRUE, main = NA, xlab = "Lag",
        lag.max = NULL, avg.col = "blue", chain.col = "red",
        na.action = na.fail, ...)
```

Arguments

<code>x</code>	: an ‘Smcmc’ class object or a list of Markov chains or a Markov chain matrix
<code>which</code>	: a vector of integers indicating which components’ ACF plots are needed. By default all components are drawn.
<code>type</code>	: the kind of ACF plot: "correlation" or "covariance". By default it is "correlation"
<code>plot</code>	: TRUE if plots are required. If FALSE, raw values are returned
<code>main</code>	: main heading of plot
<code>xlab</code>	: By default "Lag", pass another value, if you want to change.
<code>lag.max</code>	: Maximum lag for the ACF plot
<code>avg.col</code>	: color for the overall ACF of each component
<code>chain.col</code>	: color for the ACF of the individual chains.
<code>na.action</code>	: function to be called to handle missing values. ‘na.pass’ can be used.
<code>...</code>	: Other arguments

Value

returns the autocorrelation function plots of the Markov chains. Uses the more accurate globally-centered ACFs.

References

Agarwal, M., and Vats, D., “Globally-centered autocovariances in MCMC”, arxiv - 2009.01799, 2020.

Examples

```
# Producing Markov chain
chain <- matrix(0, ncol = 1, nrow = 1e3)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
```

```

}
chain <- Smcmc(list(chain))
acfplot(chain)

```

addCI

*Add simultaneous confidence interval to existing plot.***Description**

Adds simultaneous confidence intervals for quantiles and means to an existing plot.

Usage

```
addCI(x, CIs, component = 1, bord = NA, mean = TRUE, mean.color = 'plum4',
      quan.color = 'lightsteelblue3', opaq = 0.7, ...)
```

Arguments

x	: a 'Smcmc' class object
CIs	: the output from the 'getCI' function
component	: numeric indicating which component to draw the confidence intervals for
bord	: logical for whether a border is desired around the confidence intervals
mean	: logical argument whether the mean is to be plotted
mean.color	: color for the mean confidence interval
quan.color	: color for the quantile confidence intervals
opaq	: opacity of mean.col and quan.col. A value of 0 is transparent and 1 is completely opaque.
...	: arguments passed on to the boundaries of the confidence intervals in 'segments'

Value

adds segments for confidence intervals into an already existing plot environment

Examples

```

chain <- matrix(0, ncol = 1, nrow = 1e3)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
}
chain <- Smcmc(list(chain))
plot(density(chain$stacked[,1]))
CIs <- getCI(chain)
addCI(chain, CIs, component = 1)

```

boxCI	<i>Add simultaneous confidence interval to existing boxplot</i>
-------	---

Description

Adds simultaneous confidence intervals for quantiles to an existing boxplot.

Usage

```
boxCI(x, CI, component = c(1), dimn = 1,
      quan.color = 'lightsteelblue3', horizontal = FALSE)
```

Arguments

x	: a 'Smcmc' class object
CI	: the output from the 'getCI' function with 'iid = TRUE'
component	: vector indicating which components to draw the confidence intervals for
dimn	: numeric for how many components are being plotted
quan.color	: color for the quantile confidence intervals
horizontal	: logical for whether boxplots are horizontal

Value

adds segments for confidence intervals into an already existing plot environment

Examples

```
output <- matrix(rnorm(3*1e3), nrow = 1e3, ncol = 3)
```

boxplot.Siid	<i>Boxplot for Siid</i>
--------------	-------------------------

Description

Boxplots with simultaenous error bars around all quantiles for iid data.

Usage

```
## S3 method for class 'Siid'
boxplot(x, ..., alpha = 0.05, thresh = 0.001,
        quan.col = 'lightsteelblue3', opaq = .6, range = 1.5,
        width = NULL, varwidth = FALSE, outline = TRUE, plot = TRUE,
        border = par("fg"), col = 'white', ann = !add,
        horizontal = FALSE, add = FALSE)
```

Arguments

x : a ‘Siid’ class object
... : arguments sent to boxplot
alpha : confidence level of simultaneous confidence intervals
thresh : numeric typically less than .005 for the accuracy of the simultaneous procedure
quan.col : color for the quantile confidence intervals
opaq : opacity of mean.col and quan.col. A value of 0 is transparent and 1 is completely opaque.
range : as defined for base boxplot
width : as defined for base boxplot
varwidth : as defined for base boxplot
outline : as defined for base boxplot
plot : logical indicating whether the plot is to be constructed
border : as defined for base boxplot
col : as defined for base boxplot
ann : as defined for base boxplot
horizontal : as defined for base boxplot
add : as defined for base boxplot

Value

returns the base boxplot with simultaneous confidence intervals around all quantiles

References

Robertson, N., Flegal, J. M., Vats, D., and Jones, G. L., “Assessing and Visualizing Simultaneous Simulation Error”, Journal of Computational and Graphical Statistics, 2020.

Examples

```
# Generating iid data
chain <- matrix(rnorm(3*1e3), nrow = 1e3, ncol = 3)
siid.obj <- Siid(chain)
boxplot(siid.obj)
```

 convert2Smcmc

Covert to Smcmc Object

Description

To covert different MCMC objects to Smcmc object

Usage

```
convert2Smcmc(x)
```

Arguments

`x` : a object belongs from any of "mcmc.list", "stanfit", "rstan", "array", "matrix" classes.

Value

return Smcmc object having same chain(s)

densityplot	<i>density plot form Smcmc class</i>
-------------	--------------------------------------

Description

Density plots with simultaenous error bars around means and quantiles for MCMC data. The error bars account for the correlated nature of the process.

Usage

```
densityplot(x, Q = c(0.1, 0.9), alpha = 0.05, thresh = 0.001, main = NA, iid = FALSE,
            mean = TRUE, which = NULL, border = NA, mean.col = 'plum4',
            quan.col = 'lightsteelblue3', rug = FALSE, opaq = 0.7, ...)
```

Arguments

`x` : a 'Smcmc' class object

`Q` : vector of quantiles

`alpha` : confidence level of simultaneous confidence intervals

`thresh` : numeric typically less than .005 for the accuracy of the simulteaneous procedure

`main` : To add main heading

`iid` : logical argument for constructing density plot for iid samples. Defaults to FALSE

`mean` : logical argument whether the mean is to be plotted

`which` : A vector of components, if you want plots of specific components.

`border` : whether a border is required for the simultaneous confidence intervals

`mean.col` : color for the mean confidence interval

`quan.col` : color for the quantile confidence intervals

`rug` : logical indicating whether a rug plot is desired

`opaq` : opacity of mean.col and quan.col. A value of 0 is transparent and 1 is completely opaque.

`...` : arguments passed on to the density plot in base R

Value

returns a plot of the univariate density estimates with simultaneous confidence intervals wherever asked. If `plot == FALSE` a list of estimates and simultaneous confidence intervals.

References

Robertson, N., Flegal, J. M., Vats, D., and Jones, G. L., “Assessing and Visualizing Simultaneous Simulation Error”, Journal of Computational and Graphical Statistics, 2020.

Examples

```
# Producing Markov chain
chain <- matrix(0, ncol = 1, nrow = 1e3)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
}
chain <- Smcmc(list(chain))
densityplot(chain)
```

getCI

Calculates simultaneous confidence intervals.

Description

Calculates simultaneous confidence intervals for means and quantiles as indicated for the desired MCMC output

Usage

```
getCI(x, Q = c(0.1, 0.9), alpha = 0.05, thresh = 0.001, iid = FALSE,
      mean = TRUE)
```

Arguments

x	: a ‘Smcmc’ class object
Q	: vector of quantiles
alpha	: confidence levels of the simulatenous intervals
thresh	: threshold for the optimization methodology that calculates the simultaneous CIs
iid	: logical argument for constructing density plot for iid samples. Defaults to FALSE
mean	: logical indicating whether mean is to be plotted

Value

adds segments for confidence intervals into an already existing plot environment

References

Robertson, N., Flegal, J. M., Vats, D., and Jones, G. L., “Assessing and Visualizing Simultaneous Simulation Error”, Journal of Computational and Graphical Statistics, 2020.

Examples

```
chain <- matrix(0, ncol = 1, nrow = 1e3)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
}
chain <- Ssmcmc(list(chain))
plot(density(chain$stacked[,1]))
CIs <- getCI(chain)
addCI(chain, CIs, component = 1)
```

plot.Siid

Plot Siid

Description

Density plots with simultaneous error bars around means and quantiles for iid data.

Usage

```
## S3 method for class 'Siid'
plot(x, which = NULL, Q = c(0.1, 0.9), alpha = 0.05, thresh = 0.001,
      rug = TRUE, plot = TRUE, mean = TRUE, border = NA,
      mean.col = 'plum4', quan.col = 'lightsteelblue3',
      opaq = 0.7, auto.layout = TRUE,
      ask = dev.interactive(), main = NA, ...)
```

Arguments

x	: a 'Siid' class object
which	: if one want to plots specific components
Q	: vector of quantiles
alpha	: confidence level of simultaneous confidence intervals
thresh	: numeric typically less than .005 for the accuracy of the simultaneous procedure
rug	: logical indicating whether a rug plot is desired
plot	: logical argument for is plots are to be returned
mean	: logical argument whether the mean is to be plotted
border	: whether a border is required for the simultaneous confidence intervals
mean.col	: color for the mean confidence interval
quan.col	: color for the quantile confidence intervals
opaq	: opacity of mean.col and quan.col. A value of 0 is transparent and 1 is completely opaque.
auto.layout	: logical argument for an automatic layout of plots
ask	: activating interactive plots
main	: for main heading
...	: arguments passed on to the density plot in base R

Value

returns a plot of the univariate density estimates with simultaneous confidence intervals wherever asked. If `plot == FALSE` a list of estimates and simultaneous confidence intervals.

References

Robertson, N., Flegal, J. M., Vats, D., and Jones, G. L., “Assessing and Visualizing Simultaneous Simulation Error”, *Journal of Computational and Graphical Statistics*, 2020.

Examples

```
# Generating iid data
chain <- matrix(rnorm(3*1e3), nrow = 1e3, ncol = 3)
siid.obj <- Siid(chain)
plot(siid.obj)
```

plot.Smcmc

Summary plot function for Smcmc objects

Description

Plots traceplot, acfplot and densityplot of all the dimensions of chains in Smcmc object

Usage

```
## S3 method for class 'Smcmc'
plot(x, which = NULL, ...)
```

Arguments

```
x                : a 'Smcmc' class object
which            : If you are intresetd in few components only
...              : Other arguments
```

Value

return plot(s) of the all the dimensions of Smcmc object

Examples

```
# Producing Markov chain
chain <- matrix(0, ncol = 1, nrow = 1e3)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
}
chain <- Smcmc(list(chain))
plot(chain)
```

Siid

*Siid class***Description**

Class for independent and identically distributed (iid) samples

Usage

```
Siid(data, varnames = colnames(data))
```

Arguments

`data` : an iid output matrix with `nsim` rows and `p` columns
`varnames` : a character string equal to the number of columns in data

Value

an Siid class object

Examples

```
# Generating iid data
chain <- matrix(rnorm(3*1e3), nrow = 1e3, ncol = 3)
siid.obj <- Siid(chain)
```

Smcmc

*Smcmc class***Description**

Smcmc class for simulated data using Markov chain Monte Carlo

Usage

```
Smcmc(data, batch.size = TRUE, stacked = TRUE, varnames = NULL)
```

Arguments

`data` : a list of MCMC output matrices each with '`nsim`' rows and '`p`' columns
`batch.size` : logical argument, if true, calculates the batch size appropriate for this Markov chain. Setting to TRUE saves time in future steps.
`stacked` : recommended to be 'TRUE'. logical argument, if true, stores a carefully stacked version of the MCMC output for use later.
`varnames` : a character string equal to the number of columns in data

Value

an Smcmc class object

Examples

```
# Producing Markov chain
chain <- matrix(0, nrow = 1e3, ncol = 1)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
}
smcmc.obj <- Smcmc(chain)
```

summary.Smcmc

*Summary function for Smcmc objects***Description**

To show different statistics of the Smcmc object

Usage

```
## S3 method for class 'Smcmc'
summary(object, eps = 0.10, alpha = 0.05, Q = c(0.10, 0.90), ...)

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

Arguments

object	: a 'Smcmc' class object
eps	: desired volume of the confidence region
alpha	: Type one error/threshold percentage error
Q	: desired quantiles (vector of 2)
...	: Other arguments
x	: summary.Smcmc output

Value

return statistics of the all the dimensions(& chains) in Smcmc object

Examples

```
# Producing Markov chain
chain <- matrix(0, ncol = 1, nrow = 1e3)
chain[1,] <- 0
err <- rnorm(1e3)
for(i in 2:1e3)
{
  chain[i,] <- .3*chain[i-1,] + err[i]
}
chain <- Smcmc(list(chain))
summary(chain)
```

traceplot

*Trace Plot for Markov chain Monte Carlo***Description**

traceplot is a graphical tool commonly used in Bayesian statistics and Markov Chain Monte Carlo(MCMC) methods to diagnose the convergence and mixing properties of a chain.

Usage

```
traceplot(x, fast = TRUE, which = NULL,
          col = c("palevioletred3", "steelblue3", "tan3", "dimgrey", "palegreen3"),
          xlim = NULL, ylim = NULL, main = NULL, xlab = "Iteration",
          ylab = NULL, opaq = 0.9, legend = TRUE, ...)
```

Arguments

x : an ‘Smcmc’ class object or a list of Markov chains or a Markov chain matrix or a vector.

fast : a Boolean argument that will be set to TRUE by default, to make plots faster.

which : if we want full size trace plots of specific dimensions of chain, we can pass a vector of respective dimension/components.

col : color vector for multiple chains

xlim : range of x-axis

ylim : range of y-axis

main : usual heading for plot

xlab : labels of x-axis

ylab : labels of y-axis, it should be a vector of length equal to dimension of chain.

opaq : To fix the opacity of lines as per user convenience, by default it is 0.9.

legend : Boolean argument, for making legend or not.

... : Other arguments

Value

Returns the Trace Plots of Markov Chain(s)

Examples

```
# example code
# Defining a function to produce Markov chain with dimension p and size n
MakeChain <- function(p, n , h = .5)
{
  chain <- matrix(0, nrow = n, ncol = p)
  for (i in 2:n)
  {
    prop <- chain[i-1, ] + rnorm(p, mean = 0, sd = h)
    log.ratio <- sum(dnorm(prop, log = TRUE) - dnorm(chain[i-1, ], log = TRUE))
    if(log(runif(1)) < log.ratio)
    {chain[i, ] <- prop}
```

```
      else{chain[i, ] <- chain[i - 1, ]}
    }
    v = vector(length = p)
    for(i in 1:p){v[i] = paste("Comp ",i)}
    colnames(chain) = v
    return(chain)
}

chain1 <- MakeChain(p=4,n=1000)
chain2 <- MakeChain(p=4,n=1000)
chain3 <- MakeChain(p=4,n=1000)
out <- Smcmc(list(chain1,chain2,chain3))
traceplot(out)

chain1 <- MakeChain(p=6,n=1000)
chain2 <- MakeChain(p=6,n=1000)
chain3 <- MakeChain(p=6,n=1000)
out <- Smcmc(list(chain1,chain2,chain3))
traceplot(out)
```

Index

acfplot, [2](#)
addCI, [3](#)
as.Siid (Siid), [10](#)
as.Smcmc (Smcmc), [10](#)

boxCI, [4](#)
boxplot.Siid, [4](#)

convert2Smcmc, [5](#)

densityplot, [6](#)

getCI, [7](#)

is.iid (Siid), [10](#)
is.mcmc (Smcmc), [10](#)

plot.Siid, [8](#)
plot.Smcmc, [9](#)
print.summary.Smcmc (summary.Smcmc), [11](#)

Siid, [10](#)
Smcmc, [10](#)
summary.Smcmc, [11](#)

traceplot, [12](#)