Package ‘boa’

June 23, 2016

Version 1.1.8-2
Date 2016-06-22
Title Bayesian Output Analysis Program (BOA) for MCMC
Maintainer Brian J. Smith <brian-j-smith@uiowa.edu>
Depends R (>= 2.7)
Imports graphics, grDevices, stats, utils
Description A menu-driven program and library of functions for carrying out
convergence diagnostics and statistical and graphical analysis of Markov
chain Monte Carlo sampling output.
License GPL-2
URL http://www.jstatsoft.org/v21/i11
NeedsCompilation no
Author Brian J. Smith [aut, cre]
Repository CRAN
Date/Publication 2016-06-23 01:29:05

R topics documented:

boa.acf ....................................................... 2
boa.chain .................................................... 3
boa.chain.add ............................................. 4
boa.chain.collapse ...................................... 5
boa.chain.del ............................................. 6
boa.chain.eval ........................................... 6
boa.chain.gandr ......................................... 7
boa.chain.import ........................................ 8
boa.chain.info .......................................... 9
boa.chain.reset ......................................... 10
boa.chain.subset ....................................... 10
boa.chain.support ...................................... 11
boa.geweke ............................................. 12
Autocorrelation Function

Description

Computes lag autocorrelations for the parameters in an MCMC sequence.

Usage

boa.acf(link, lags)
boa.chain

Arguments

link    Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link)
lags    Vector of lags at which to estimate the autocorrelation function.

Value

A matrix whose columns and rows contain the estimated autocorrelation functions at the specified lags and the monitored parameters, respectively.

Author(s)

Brian J. Smith

See Also

boa.plot, boa.plot.acf, boa.print.acf

Description

Extracts and sets the MCMC sequence(s) stored internally for use during a BOA session. Users can safely use this function to extract data. However, it should not be called directly to modify the stored MCMC sequence(s). Instead, data management should accomplished with the libboa.chain functions.

Usage

boa.chain(...)  

Arguments

...  A list may be given as the only argument, or a character string given as the only argument, or any number of arguments may be in the <name> = <value> form, or no argument at all may be given. See the Value and Side Effects sections for further details.

Value

If no arguments are supplied, a list of the current values is returned. If a character string is given, the current value of the named variable is returned. Otherwise, no values are returned.
Parameter Names and Default Values

master = list()  Master list containing all added MCMC sequences. Sequences may only be added to or deleted from the master list. The sequences in this list are never modified or used in any of the analyses. It serves as a template for the working list described below. The parameters and iterations may differ between the sequences. Each element in the list is a matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames.

master.support = list()  List containing the support for each of the parameters in 'master'. Each element in the list is a matrix whose columns and rows contain the monitored parameters and the support (lower and upper limits), respectively. The default for each parameter is c(-Inf, Inf).

work = master  Working list used in all analyses. This list is essentially a modifiable copy of 'master'. All data management functions operate on the working list.

work.support = master.support  List containing the support for each of the parameters in 'work'.

work.sync = TRUE  Logical value indicating that the working list is an identical copy of the master list. This variable is automatically set to F after a successful call to boa.chain.subset or boa.chain.collapse. While work.sync = F, only the master list will be updated if MCMC sequences are added or deleted. A call to boa.chain.reset will reset this variable to T and copy the master list to the working list.

Side Effects

When variables are set, boa.chain() modifies the internal list .boa.chain. If boa.chain() is called with either a list as the single argument, or with one or more arguments in the <name> = <value> form, the variables specified by the names in the arguments are modified.

Author(s)

Brian J. Smith

boa.chain.add  Add MCMC Sequence

Description

Adds an MCMC sequence to the session lists of sequences. Most users should not call this function directly. Passing improperly formatted data to this function will most likely break the functions in this library. The preferred method of adding a new sequence is to call the function boa.chain.import.

Usage

boa.chain.add(link, lname)
 boa.chain.collapse

Arguments

  link      Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).
  lname    Character string giving the list name to use for the newly added MCMC sequence.

Value

  A logical value indicating that 'link' has been successfully added.

Author(s)

  Brian J. Smith

See Also

  boa.chain.import

boa.chain.collapse

Concatenate MCMC Sequences

Description

  Concatenates together all of the MCMC sequences in the working session list of sequences. Only those parameters common to all sequences are kept.

Usage

  boa.chain.collapse()

Value

  A logical value indicating that the MCMC sequences have been collapsed. If the sequences share no common parameters, no modifications are made to the working list and FALSE is returned.

Author(s)

  Brian J. Smith

See Also

  codeboa.chain.reset, codeboa.chain.subset
boa.chain.del  

*Delete MCMC Sequences*

**Description**

Delete MCMC sequences from the session list of sequences.

**Usage**

    boa.chain.del(lnames, pnames)

**Arguments**

- `lnames` : Character vector giving the names of the MCMC sequences in the session list of sequences to be deleted. If omitted, no sequences are deleted.
- `pnames` : Character vector giving the names of the parameters in the MCMC sequences to be deleted. If omitted, no parameters are deleted.

**Side Effects**

The specified MCMC sequences are deleted from the session lists of sequences.

**Author(s)**

Brian J. Smith

boa.chain.eval  

*Formulate New Parameter*

**Description**

Evaluates a user-specified expression to formulate a new parameter in each of the system lists of MCMC sequences.

**Usage**

    boa.chain.eval(expr, pname)

**Arguments**

- `expr` : S expression object used to formulate the new parameter. Typically, the new parameter is a function of the existing parameters.
- `pname` : Name given to the new parameter.
**Side Effects**

The specified parameter is added to the session lists of MCMC sequences.

**Author(s)**

Brian J. Smith

---

**boa.chain.gandr**  
*Gelman and Rubin Convergence Diagnostics*

**Description**

Computes the Gelman and Rubin convergence diagnostics for a list of MCMC sequences. Estimates are calculated from the second half of each sequence.

**Usage**

`boa.chain.gandr(chain, chain.support, alpha, pnames, window, to)`

**Arguments**

- **chain**: List of matrices whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to the dimnames.
- **chain.support**: List of matrices whose columns and rows contain the monitored parameters and the support (lower and upper limits), respectively.
- **alpha**: Quantile \((1 - \alpha / 2)\) at which to estimate the upper limit of the shrink factor.
- **pnames**: Character vector giving the names of the parameters to use in the analysis. If omitted, all parameters are used.
- **window**: Proportion of iterations to include in the analysis. If omitted, 50% are included.
- **to**: Largest iteration to include in the analysis. If omitted, no upper bound is set.

**Value**

- **psrf**: A vector containing the Gelman and Rubin (uncorrected) potential scale reduction factors for the monitored parameters.
- **csrf**: A matrix whose columns and rows are the Gelman and Rubin corrected scale reduction factors (i.e. shrink factor estimates at the median and specified quantile of the sampling distribution) and the monitored parameters, respectively. A correction of \((df + 3) / (df + 1)\) is applied to the scale reduction factors.
- **mpsrf**: A numeric value giving the multivariate potential scale reduction factor proposed by Brooks and Gelman.
- **window**: A numeric vector with two elements giving the range of the iterations used in the analysis.
Author(s)
Brian J. Smith, Nicky Best, Kate Cowles

References

See Also
boa.plot, boa.plot.bandg, boa.plot.gandr, boa.print.gandr

boa.chain.import Import MCMC Sequence

Description
Reads an MCMC sequence from an external file and adds it to the session lists of sequences.

Usage
boa.chain.import(prefix, path = boa.par("path"), type = "ASCII")

Arguments
prefix Character string giving the prefix for the file(s) in which the MCMC sequence is stored.
path Character string giving the directory path in which the file(s) subsides.
type Character string specifying the type of data to be imported. Currently, the supported types are:
  • "ASCII" : ASCII text file. Calls boa.importASCII to import data from "prefix.txt".
  • "BUGS" : BUGS output files. Callsboa.importBUGS to import data from "prefix.ind" and "prefix.out".
  • "S" : S data frame or matrix.

Value
A logical value indicating that the specified file(s) has been successfully imported.

Side Effects
The imported MCMC sequence is automatically added to the session lists of sequences via a call to boa.chain.add.
boa.chain.info

Author(s)
Brian J. Smith

See Also
boa.chain.add, boa.importASCII, boa.importBUGS

boa.chain.info  Information about MCMC Sequences

Description
Returns summary information from a list of MCMC sequences.

Usage
boa.chain.info(chain, chain.support)

Arguments

chain List of matrices whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to the dimnames.

chain.support List of matrices whose columns and rows contain the monitored parameters and the support (lower and upper limits), respectively.

Value

lnames Character vector giving the names of the MCMC sequences in the session list of sequences.

pnames List of character vectors giving the parameter names in each of the MCMC sequences.

iter List of numeric vectors giving the iterations from each MCMC sequence.

iter.range Matrix whose columns give the range of the iterations for the MCMC sequences named in the rows.

support List of numeric vectors giving the support for the parameters in each of the MCMC sequences.

Author(s)
Brian J. Smith

See Also
boa.print.info
boa.chain.reset  \hspace{1em} \textit{Reset MCMC Sequences}

\textbf{Description}

Returns the data in the working list of MCMC sequences to the state it was in when originally imported. This function undoes any subsetting or collapsing that was done via the \texttt{boa.chain.collapse} or \texttt{boa.chain.subset} functions.

\textbf{Usage}

\begin{verbatim}
boa.chain.reset()
\end{verbatim}

\textbf{Side Effects}

Returns the data in the working list of MCMC sequences to the state it was in when originally imported. This function undoes any subsetting or collapsing that was done via the \texttt{boa.chain.collapse} or \texttt{boa.chain.subset} functions.

\textbf{Author(s)}

Brian J. Smith

\textbf{See Also}

\texttt{boa.chain.collapse, boa.chain.subset}

\begin{table}
\centering
\begin{tabular}{ll}
\textbf{boa.chain.subset}  & \textit{Subset MCMC Sequences} \\
\end{tabular}
\end{table}

\textbf{Description}

Selects a subset of the MCMC sequences stored in the working session list of sequences.

\textbf{Usage}

\begin{verbatim}
boa.chain.subset(lnames, pnames, iter)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
\item \texttt{lnames} \hspace{1em} Character vector giving the names of the MCMC sequences to include in the subset. If omitted, all sequences are included.
\item \texttt{pnames} \hspace{1em} Character vector giving the names of the parameters to include in the subset. If omitted, all parameters are included.
\item \texttt{iter} \hspace{1em} Character vector giving the names of the parameters to include in the subset. If omitted, all parameters are included.
\end{itemize}
Description
Modifies the support (range of possible values) for the parameters in the session lists of MCMC sequences.

Usage
boa.chain.support(lnames, pnames, limits)

Arguments
lnames Character vector giving the names of the MCMC sequences over which to apply the changes. If omitted, changes are applied to all sequences.
pnames Character vector giving the names of the parameters whose support is to be modified.
limits Numeric vector with two elements giving the lower and upper limits, respectively, of the support for the specified parameters. Unbounded lower or upper limits should be specified as -Inf or Inf, respectively.

Value
A logical vector with two elements indicating that the support has been changed for variables in the master list and working list of MCMC sequences, respectively.

Side Effects
The support for each of the specified parameters is changed in the session lists of MCMC sequences.
boa.geweke

Description

Computes the Geweke convergence diagnostics for the parameters in an MCMC sequence.

Usage

boa.geweke(link, p.first, p.last)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).</td>
</tr>
<tr>
<td>p.first</td>
<td>Proportion of iterations to include in the first window.</td>
</tr>
<tr>
<td>p.last</td>
<td>Proportion of iterations to include in the last window.</td>
</tr>
</tbody>
</table>

Value

A matrix whose columns contain the Z-Scores and associated p-values and whose rows contain the monitored parameters.

Author(s)

Brian J. Smith

References


See Also

boa.plot, boa.plot.geweke, boa.print.geweke
**boa.handw**

---

**Heidelberger and Welch Convergence Diagnostics**

---

**Description**

Computes the Heidelberger and Welch convergence diagnostics for the parameters in an MCMC sequence.

**Usage**

`boa.handw(link, error, alpha)`

**Arguments**

- **link**: Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
- **error**: Accuracy of the posterior estimates for the parameters.
- **alpha**: Alpha level for the confidence in the sample mean of the retained iterations.

**Value**

A matrix whose columns and rows are the Heidelberger and Welch convergence diagnostics (i.e. stationarity test, number of iterations to keep and to drop, Cramer-von-Mises statistic, halfwidth test, mean, and halfwidth) and the monitored parameters, respectively.

**Author(s)**

Brian J. Smith, Nicky Best, Kate Cowles

**References**


**See Also**

`boa.print.handw`
boa.hpd  

**Highest Probability Density Interval**

**Description**

Estimates the highest probability density (HPD) interval for the given parameter draws. Uses the Chen and Shao algorithm assuming a unimodal marginal posterior distribution.

**Usage**

boa.hpd(x, alpha)

**Arguments**

- **x**: MCMC draws from the marginal posterior to use in computing the HPD.
- **alpha**: Specifies the 100*(1 - alpha)% interval to compute.

**Value**

A vector containing the lower and upper bound of the HPD interval, labeled "Lower Bound" and "Upper Bound", respectively.

**Author(s)**

Brian J. Smith

**References**


boa.importASCII  

**Import ASCII Data**

**Description**

Import data from an ASCII text file. The variables and iterations should appear in the columns and rows, respectively, of the file. The variable names must be given in the first row. The iteration numbers are taken from the the column entitled "iter" (case-sensitive), if it exists; otherwise, the numbers 1..n are used, where n is the number of rows. The columns may be separated by white space or tabs.

**Usage**

boa.importASCII(prefix, path = NULL)
boa.importBUGS

Arguments

prefix  Character string giving the prefix for the files in which the ASCII data is stored. boa.importASCII() looks for the file "prefix.txt".

path    Character string giving the directory path in which the file subsides. This argument may be omitted if the file is located in the current working directory. The specified path should not end with a slash(es).

Value

If the data is successfully imported, a matrix is returned whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames of the returned matrix. Otherwise, NULL is returned.

Author(s)

Brian J. Smith

See Also

boa.chain.import, boa.importBUGS, boa.importMatrix
**boa.importMatrix**

**Author(s)**

Brian J. Smith

**See Also**

`boa.chain.import, boa.importASCII, boa.importMatrix`

---

**boa.importMatrix**

*Import ASCII Data*

**Description**

Import data from an S numeric matrix object. The variables and iterations should appear in the columns and rows, respectively, of the matrix. The variable names and iteration numbers may be optionally specified in the dimnames of the matrix object. Otherwise, the iterations will be numbered 1..n, where n is the number of rows.

**Usage**

`boa.importMatrix(prefix)`

**Arguments**

- `prefix`: Character string giving the name of the S object.

**Value**

If the data is successfully imported, a matrix is returned whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames of the returned matrix. Otherwise, NULL is returned.

**Author(s)**

Brian J. Smith

**See Also**

`boa.chain.import, boa.importASCII, boa.importBUGS`
**boa.init**  

*Start Session*

**Description**

boa.init() is the first function called to start a BOA session when using the command-line. It sets up the internal data structures and initializes them with the appropriate default values.

**Usage**

boa.init(recover = FALSE)

**Arguments**

- **recover**  
  a logical value for use if the previous BOA menu session terminated unexpectedly. It may be possible to crash the menu system by supplying the wrong type of data. In the event of a crash, recover = TRUE may be used to ensure that no data is lost.

**Author(s)**

Brian J. Smith

**See Also**

boa.load, boa.quit

---

**boa.load**  

*Load Session*

**Description**

Loads the data and global parameter settings from a previously saved BOA session.

**Usage**

boa.load(name, envir = globalenv())

**Arguments**

- **name**  
  Character string giving the name of the object containing the session to be loaded.

- **envir**  
  The `environment` from which the object should come. For more information, consult the help documentation in R on the `get` function.
**Value**

A logical value indicating that the object was successfully loaded.

**Author(s)**

Brian J. Smith

---

**boa.menu**

*Interactive BOA Menu*

---

**Description**

Starts the interactive menu for BOA. Provides a menu-driven interface to all of the convergence diagnostics and statistical and graphical analysis tools.

**Usage**

```r
boa.menu(recover = FALSE)
```

**Arguments**

- `recover` a logical value for use if the previous BOA menu session terminated unexpectedly. It may be possible to crash the menu system by supplying the wrong type of data. In the event of a crash, `recover = TRUE` may be used to ensure that no data is lost.

**Warning**

Do not use the `recover` argument unless starting the menu immediately after a crash.

**Note**

`boa.menu` is the only function in the BOA library that need be called in order to use the menu-driven interface.

**Author(s)**

Brian J. Smith

**Examples**

```r
## Not run:
## Start the menu system
boa.menu()

## End(Not run)
```
Global Parameters

Description
Displays and sets the global parameters stored internally for use during a BOA session.

Usage
boa.par("")

Arguments
A list may be given as the only argument, or a character string given as the only argument, or any number of arguments may be in the <name> = <value> form, or no argument at all may be given.

Value
If no arguments are supplied, a list of the current values is returned. If a character string is given, the current value of the named variable is returned. Otherwise, a list of the named variables and their old values is returned, but not printed.

Parameter Names and Default Values

- **acf.lags** = c(1, 5, 10, 50) Numeric vector of lags at which to estimate the autocorrelation function.
- **alpha** = 0.05 Type I error rate used for all statistical tests and confidence intervals.
- **ASCIIext** = ".txt" Character string giving the filename extension used when importing ASCII files.
- **bandwidth** = function(x) 0.5 * diff(range(x)) / (log(length(x)) + 1) Function for computing the bandwidth used in estimating the density functions for parameters. This should take one argument which is a numeric vector of data on which density estimation is to be performed. A constant bandwidth may be specified by having this function return the desired constant.
- **batch.size** = 50 Number of iterations to include in each batch when computing batch means and lag-1 batch autocorrelations. The batch size has the single largest impact on the time required to compute summary statistics. The computation time is decreased dramatically as the batch size increases. Consequently, users may want to increase the value of this variable for long MCMC sequences.
- **dev** = "see below" Character string giving the name of the function that creates graphics windows on the current platform. For Unix systems this is either "motif", "openlook", or "X11". The default is "motif" for the UNIX S-PLUS, "win.graph" for Windows S-PLUS, "X11" for UNIX R, and "windows" for Windows R.
- **dev.list** = numeric(0) Numeric vector containing the active graphics windows in use by the program. This is automatically maintained by the program, user should not directly modify this variable.
boa.par

`gandr.bins = 20` Number of line segments within the MCMC sequence at which to plot the Gelman and Rubin shrink factors.

`gandr.win = 0.50` Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.

`geweke.bins = 10` Number of line segments within the MCMC sequence at which to plot the Geweke convergence diagnostics.

`geweke.first = 0.1` Proportion of iterations to include in the first window when computing the Geweke convergence diagnostics.

`geweke.last = 0.5` Proportion of iterations to include in the last window when computing the Geweke convergence diagnostics.

`handw.error = 0.1` Accuracy of the posterior estimates when computing the Heidleberger and Welch convergence diagnostics.

`kernel = "gaussian"` Character string giving the type of window used in estimating the density functions for parameters. Other choices are "cosine", "rectangular", or "triangular".

`legend = TRUE` Logical value indicating that a legend be included in the plots.

`path = ""` Character string giving the directory path in which the raw data files are stored. The default may be used if the files are located in the current working directory. The specified path should not end with a slash(es).

`par = list()` List specifying graphics parameters passed to the `par` function for the construction of new plots.

`plot.mfdim = c(3, 2)` Numeric vector giving the maximum number of rows and columns, respectively, of plots to include in a single graphics window.

`plot.new = F` Logical value indicating that a new graphics window be automatically opened upon successive calls to `boa.plot()`. Otherwise, previous graphics windows will be closed.

`plot.onelink = FALSE` Logical value indicating that each plot should include only one MCMC sequence. Otherwise, all sequences are displayed on the same plot.

`quantiles = c(0.025, 0.5, 0.975)` Vector of probabilities at which to compute the quantiles. Values must be between 0 and 1.

`randl.error = 0.005` Desired amount of error in estimating the quantile specified in the Raftery and Lewis convergence diagnostics.

`randl.delta = 0.001` Delta valued used in computing the Raftery and Lewis convergence diagnostics.

`randl.q = 0.025` Quantile to be estimated in computing the Raftery and Lewis convergence diagnostics.

`title = TRUE` Logical value indicating that a title be added to the plots.

### Side Effects

When variables are set, `boa.par()` modifies the internal list `boa.par`. If `boa.par()` is called with either a list as the single argument, or with one or more arguments in the `<name> = <value>` form, the variables specified by the names in the arguments are modified.

### Author(s)

Brian J. Smith
boa.plot

Plot Parameters in MCMC Sequences

Description

Automatically generates the specified plot type for all parameters in the working session list of MCMC sequences. This function takes care of all the plotting tasks - opening windows, setting the number of plots per page, and adding titles.

Usage

boa.plot(type, dev = boa.par("dev"), mfdim = boa.par("plot.mfdim"),
newplot = boa.par("plot.new"), onelink = boa.par("plot.onelink"),
title = boa.par("title"))

Arguments

- **type**: The type of plots to generate. The supported types are:
  - "acf" = autocorrelation functions
  - "bandg" = Brooks and Gelman multivariate shrink factors
  - "density" = density functions
  - "gandr" = Gelman and Rubin shrink factors
  - "geweke" = Geweke convergence diagnostics
  - "history" = running means
  - "trace" = trace histories

- **dev**: Character string giving the name of the function that creates graphics windows on the current platform. For Unix systems this is either "motif", "openlook", or "X11". For windows, this should be set to "win.graph"

- **mfdim**: Numeric vector giving the maximum number of rows and columns, respectively, of plots to include in a single graphics window. If only one graphics window is opened, ‘mfdim’ is proportionately scaled down so as to minimize the number of empty frames within that window.

- **newplot**: Logical value indicating that a new graphics window be automatically opened. Otherwise, previous graphics windows will be closed.

- **onelink**: Logical value indicating that each plot should include only one MCMC sequence. Otherwise, all sequences are displayed on the same plot.

- **title**: Logical value indicating that a title be added to the plot.

Value

A logical value indicating that the plots were successfully created.

Author(s)

Brian J. Smith
boa.plot.acf

See Also

boa.plot.acf, boa.plot.bandg, boa.plot.density, boa.plot.gandr, boa.plot.geweke, boa.plot.history, boa.plot.trace

---

boa.plot.acf  
*Plot Autocorrelation Function*

---

Description

Creates a single plot of the lag autocorrelations for a specified parameter.

Usage

boa.plot.acf(lname, pname, annotate = boa.par("legend"))

Arguments

- **lname**: Character string giving the name of the desired MCMC sequence in the working session list of sequences.
- **pname**: Character string giving the name of the parameter to be plotted.
- **annotate**: Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.acf, boa.print.acf
boa.plot.bandg

Plot Brooks and Gelman Multivariate Shrink Factors

Description

Plots the Brooks and Gelman multivariate shrink factors within different line segments across the MCMC sequences. This diagnostic is a multivariate extension to the Gelman and Rubin shrink factors.

Usage

boa.plot.bandg(bins = boa.par("gandr.bins"), win = boa.par("gandr.win"), annotate = boa.par("legend"))

Arguments

bins Number of line segments within the MCMC sequence at which to plot the Gelman and Rubin shrink factors. The first segment contains the first 50 iterations; the remaining iterations are partitioned into equal bins and added incrementally to construct the remaining line segments. The shrink factors are plotted against the maximum iteration number for the segment. Cubic splines are used to interpolate through the point estimates for each segment.

win Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.

annotate Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.chain.gandr, boa.plot, boa.plot.gandr, boa.print.gandr
boa.plot.density  

*Plot Density Functions*

**Description**

Estimates and displays, in a single plot, the density function(s) for the specified parameter(s).

**Usage**

`boa.plot.density(lnames, pname, bandwidth = boa.par("bandwidth"),
              window = boa.par("kernel"), annotate = boa.par("legend"))`

**Arguments**

- **lnames**: Character vector giving the names of the desired MCMC sequence in the working session list of sequences.
- **pname**: Character string giving the name of the parameter to be plotted.
- **bandwidth**: Function for computing the bandwidth used in estimating the density functions for parameters. This should take one argument which is a numeric vector of data on which density estimation is to be performed. A constant bandwidth may be specified by having this function return the desired constant.
- **window**: Character string giving the type of window used in estimating the density functions for the parameters. Available choices are "cosine", "gaussian", "rectangular", or "triangular".
- **annotate**: Logical value indicating that a legend be included in the plot.

**Value**

A logical value indicating that the plot was successfully created.

**Author(s)**

Brian J. Smith

**See Also**

`boa.plot`
boa.plot.gandr

Plot Gelman and Rubin Shrink Factors

Description

Creates a single plot of the Gelman and Rubin shrink factors within different line segments for a parameter in an MCMC sequence.

Usage

boa.plot.gandr(pname, bins = boa.par("gandr.bins"), alpha = boa.par("alpha"), win = boa.par("gandr.win"), annotate = boa.par("legend"))

Arguments

pname Character string giving the name of the parameter in the working session list of MCMC sequences to be plotted.
bins Number of line segments within the MCMC sequence at which to plot the Gelman and Rubin shrink factors. The first segment contains the first 50 iterations; the remaining iterations are partitioned into equal bins and added incrementally to construct the remaining line segments. The shrink factors are plotted against the maximum iteration number for the segment. Cubic splines are used to interpolate through the point estimates for each segment.
alpha Quantile (1 - alpha / 2) at which to estimate the upper limit of the shrink factor.
win Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.
annotate Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.chain.gandr, boa.plot, boa.plot.bandg, boa.print.gandr
boa.plot.geweke

Plot Geweke Convergence Diagnostics

Description

Creates a single plot of the Geweke convergence diagnostics within different line segments for a parameter in an MCMC sequence.

Usage

boa.plot.geweke(lname, pname, bins = boa.par("geweke.bins"),
    p.first = boa.par("geweke.first"), p.last = boa.par("geweke.last"),
    alpha = boa.par("alpha"), annotate = boa.par("legend"))

Arguments

lname Character string giving the name of the desired MCMC sequence in the working session list of sequences.

pname Character string giving the name of the parameter to be plotted.

bins Number of line segments within the MCMC sequence at which to plot the Geweke convergence diagnostics. The ith line segment contains the last ((bins - i + 1) / bins)*100 bins. This may lead to segments in which there are too few iterations to compute the convergence diagnostics. Such segments, if they exist, are automatically omitted from the calculations.

p.first Proportion of iterations to include in the first window.

p.last Proportion of iterations to include in the last window.

alpha Alpha level for the rejection region lines drawn on the graph.

annotate Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith, Nicky Best, Kate Cowles

See Also

boa.geweke, boa.plot, boa.print.geweke
boa.plot.history

Description

Computes and displays, in a single plot, the running mean(s) for the specified parameter(s).

Usage

boa.plot.history(lnames, pname, annotate = boa.par("legend"))

Arguments

  lnames  Character vector giving the name of the desired MCMC sequence in the working
          session list of sequences.
  pname   Character vector giving the names of the parameters to be plotted.
  annotate Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.plot

boa.plot.par

Set Plotting Parameters

Description

Set Plotting Parameters

Usage

boa.plot.par(mfdim = c(1, 1), title = TRUE)

Arguments

  mfdim  Numeric vector with two elements giving the number of rows and column, re-
          spectively, of plots to display on the page. Plots will be drawn row-by-row.
  title  Logical value indicating that a title be added to the plot.
Author(s)
Brian J. Smith

boa.plot.title  
*Add Plot Title*

Description
Adds a title to the graphics window. This function should be called after all plots have been added to the target graphics window.

Usage
boa.plot.title(text)

Arguments
- **text**: Character string giving the title to be centered and displayed across the top of the graphics window.

Author(s)
Brian J. Smith

boa.plot.trace  
*Plot Trace Histories*

Description
Displays, in a single plot, the trace histories for the specified parameter(s).

Usage
boa.plot.trace(lnames, pname, annotate = boa.par("legend"))

Arguments
- **lnames**: Character vector giving the name of the desired MCMC sequence in the working session list of sequences.
- **pname**: Character string giving the name of the parameters to be plotted.
- **annotate**: Logical value indicating that a legend be included in the plot.

Value
A logical value indicating that the plot was successfully created.
boa.print.acf

Author(s)
Brian J. Smith

See Also
boa.plot

Print Autocorrelation Functions

Description
Iteratively calls boa.acf() to display the lag autocorrelations for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage
boa.print.acf(lags = boa.par("acf.lags"))

Arguments
lags
Numeric vector of lags at which to estimate the autocorrelation functions.

Author(s)
Brian J. Smith

See Also
boa.acf, boa.plot, boa.plot.acf

Print Correlation Matrices

Description
Iteratively computes and displays the correlation matrices for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage
boa.print.cor()

Author(s)
Brian J. Smith
boa.print.gandr  
*Print Gelman and Rubin Convergence Diagnostics*

**Description**

Calls boa.chain.gandr() and displays the Gelman and Rubin convergence diagnostics for the MCMC sequences in the working session list of sequences.

**Usage**

```r
boa.print.gandr(alpha = boa.par("alpha"), win = boa.par("gandr.win"))
```

**Arguments**

- `alpha`: Quantile (1 - alpha / 2) at which to estimate the upper limit of the corrected shrink factor.
- `win`: Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.

**Author(s)**

Brian J. Smith

**See Also**

boa.chain.gandr, boa.plot, boa.plot.bandg, boa.plot.gandr

boa.print.geweke  
*Print Geweke Convergence Diagnostics*

**Description**

Iteratively calls boa.geweke() to display the Geweke convergence diagnostics for the parameters in each of the MCMC sequences in the working session list of sequences.

**Usage**

```r
boa.print.geweke(p.first = boa.par("geweke.first"), p.last = boa.par("geweke.last"))
```

**Arguments**

- `p.first`: Proportion of iterations to include in the first window.
- `p.last`: Proportion of iterations to include in the last window.

**Author(s)**

Brian J. Smiht
boa.print.handw

See Also

boa.geweke, boa.plot.boa.plot.geweke

boa.print.handw  Print Heidelberger and Welch Convergence Diagnostics

Description

Iteratively calls boa.handw() to display the Heidelberger and Welch convergence diagnostics for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

boa.print.handw(error = boa.par("handw.error"), alpha = boa.par("alpha"))

Arguments

eroerror Accuracy of the posterior estimates for the monitored parameters.
alphaAlpha level for the confidence in the sample mean of the retained iterations.

Author(s)

Brian J. Smith

See Also

boa.handw

boa.print.hpd  Print Highest Probability Density Intervals

Description

Iteratively calls boa.hpd() to display the highest probability density (HPD) intervals for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

boa.print.hpd(alpha = boa.par("alpha"))

Arguments

deralpapelpha Specifies the 100*(1-alpha)% HPD intervals to be computed.
**boaprint.info**  

*Print Information about MCMC Sequences*

**Description**

Iteratively calls boa.chain.info() to display information from the MCMC sequences in the specified session list of sequences.

**Usage**

`boa.print.info(which = "work")`

**Arguments**

- `which`  
  Character string specifying which session list of MCMC sequences for which to print information. The two choices are "work" (used in all analyses) or "data" (the template for "work").

**Value**

The full list obtained from boa.chain.info() is returned, but not printed.

**Author(s)**

Brian J. Smith

**See Also**

`boa.hpd`
boa.print.par

Print Global Parameters’ Descriptions and Current Values

Description

Displays a description of and the current value for all global parameters.

Usage

boa.print.par(group)

Arguments

group Character string specifying which parameter group to display. The groups are
"Analysis", "Data", and "Plot". If omitted, information on all global parameters
is displayed.

Value

The character matrix obtained from boa.pardesc() is returned, but not printed.

Author(s)

Brian J. Smith

See Also

boa.pardesc

boa.print.randl

Print Raftery and Lewis Convergence Diagnostics

Description

Iteratively calls boa.randl() to display the Raftery and Lewis convergence diagnostics for the pa-
rameters in each of the MCMC sequences in the working session list of sequences.

Usage

boa.print.randl(q = boa.par("randl.q"), error = boa.par("randl.error"),
prob = 1 - boa.par("alpha"), delta = boa.par("randl.delta"))
boa.print.stats

Arguments
- \( q \): Quantile to be estimated.
- \( \text{error} \): Desired amount of error in estimating the specified quantile \( q \).
- \( \text{prob} \): Probability of attaining the desired degree of \( \text{error} \).
- \( \text{delta} \): Delta value used in computing the convergence diagnostic.

Author(s)
Brian J. Smith

See Also
boa.randl

boa.print.stats

Print Summary Statistics

Description
Iteratively calls boa.stats() to display summary statistics for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage
boa.print.stats(probs = boa.par("quantiles"), batch.size = boa.par("batch.size"))

Arguments
- \( \text{probs} \): Vector of probabilities at which to compute the quantiles. Values must be between 0 and 1.
- \( \text{batch.size} \): Number of iterations to include in each batch when computing batch means and lag-1 batch autocorrelation function.

Author(s)
Brian J. Smith

See Also
boa.stats
boa.quit

Quit Session

Description
boa.quit() is called to end a BOA session. It removes the objects used to store the session lists of MCMC sequences and the global parameters. Failure to do this may result in lost or lingering data.

Usage
boa.quit()

Author(s)
Brian J. Smith

See Also
boa.save, boa.init

boa.randl
Raftery and Lewis Convergence Diagnostics

Description
Computes the Raftery and Lewis convergence diagnostics for the parameters in an MCMC sequence.

Usage
boa.randl(link, q, error, prob, delta)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).</td>
</tr>
<tr>
<td>q</td>
<td>Quantile to be estimated.</td>
</tr>
<tr>
<td>error</td>
<td>Desired amount of error in estimating the specified quantile 'q'.</td>
</tr>
<tr>
<td>prob</td>
<td>Probability of attaining the desired degree of error - 'error'.</td>
</tr>
<tr>
<td>delta</td>
<td>Delta value used in computing the convergence diagnostics.</td>
</tr>
</tbody>
</table>

Value
A matrix whose columns and rows are the Raftery and Lewis convergence diagnostics (i.e. thin, burn-in, total, lower bound, and dependence factor) and the monitored parameters, respectively.
boa.save

Author(s)
Brian J. Smith, Nicky Best, Kate Cowles

References

See Also
boa.print.randl

Description
Save the current state of the session lists of MCMC sequences and the global parameters to a database object.

Usage
boa.save(name, envir = globalenv(), replace = FALSE)

Arguments
name Character string giving the name of the object to which the current session should be saved.
envir The 'environment' to which the object should be saved. For more information, consult the help documentation in R on the assign() function.
replace Logical value indicating whether object name should be replaced if it already exists.

Value
A logical value indicating that the session was successfully saved to the specified object.

Author(s)
Brian J. Smith
boa.stats

Summary Statistics

Description

Computes summary statistics for the parameters in an MCMC sequence.

Usage

boa.stats(link, probs, batch.size)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).</td>
</tr>
<tr>
<td>probs</td>
<td>Vector of probabilities at which to compute the quantiles. Values must be between 0 and 1.</td>
</tr>
<tr>
<td>batch.size</td>
<td>Number of iterations to include in each batch when computing batch means and lag-1 batch autocorrelations.</td>
</tr>
</tbody>
</table>

Value

A matrix whose columns and rows contain the summary statistics (i.e. sample mean, standard deviation, naive standard error, MC error, batch standard error, lag-1 batch autocorrelation, specified quantiles, minimum and maximum iteration numbers, and total iterations in the sample) and the monitored parameters, respectively.

Author(s)

Brian J. Smith

See Also

boa.print.stats
**Description**

The Line example involves a linear regression analysis of the data points (1,1), (2,3), (3,3), (4,3), and (5,5). The proposed Bayesian model is

\[ y[i] \sim N(\mu[i], \tau) \]

\[ \mu[i] = \alpha + \beta \times (x[i] - \text{mean}(x[])) \]

with the following priors:

\[ \alpha \sim N(0, 0.0001) \]
\[ \beta \sim N(0, 0.0001) \]
\[ \tau \sim \Gamma(0.001, 0.001) \]

Two parallel chains from the MCMC sampler were generated and the output saved in the S data frames line1 and line2.

**Usage**

data(line)

**Format**

The data frames line1 and line2 each containing 200 observations.

**References**

Index

*Topic **datasets**
  line, 38
*Topic **hplot**
  boa.plot, 21
  boa.plot.acf, 22
  boa.plot.bandg, 23
  boa.plot.density, 24
  boa.plot.gandr, 25
  boa.plot.geweke, 26
  boa.plot.history, 27
  boa.plot.trace, 28
*Topic **interface**
  boa.menu, 18
*Topic **print**
  boa.print.acf, 29
  boa.print.cor, 29
  boa.print.gandr, 30
  boa.print.geweke, 30
  boa.print.handw, 31
  boa.print.hpd, 31
  boa.print.info, 32
  boa.print.par, 33
  boa.print.randl, 33
  boa.print.stats, 34
*Topic **univar**
  boa.acf, 2
  boa.chain.gandr, 7
  boa.geweke, 12
  boa.handw, 13
  boa.hpd, 14
  boa.randl, 35
  boa.stats, 37
*Topic **utilities**
  boa.chain, 3
  boa.chain.add, 4
  boa.chain.collapse, 5
  boa.chain.del, 6
  boa.chain.eval, 6
  boa.chain.import, 8
  boa.chain.info, 9
  boa.chain.reset, 10
  boa.chain.subset, 10
  boa.chain.support, 11
  boa.importASCII, 14
  boa.importBUGS, 15
  boa.importMatrix, 16
  boa.init, 17
  boa.load, 17
  boa.par, 19
  boa.plot.par, 27
  boa.plot.title, 28
  boa.quit, 35
  boa.save, 36
  boa.acf, 2, 22, 29
  boa.chain, 3
  boa.chain.add, 4, 8, 9
  boa.chain.collapse, 4, 5, 10, 11
  boa.chain.del, 6
  boa.chain.eval, 6
  boa.chain.gandr, 7, 23, 25, 30
  boa.chain.import, 4, 5, 8, 15, 16
  boa.chain.info, 9, 32
  boa.chain.reset, 4, 5, 10, 11
  boa.chain.subset, 4, 5, 10, 10
  boa.chain.support, 11
  boa.geweke, 12, 26, 31
  boa.handw, 13, 31
  boa.hpd, 14, 32
  boa.importASCII, 8, 9, 14, 16
  boa.importBUGS, 8, 9, 15, 15, 16
  boa.importMatrix, 15, 16, 16
  boa.init, 17, 35
  boa.load, 17, 17
  boa.menu, 18
  boa.par, 19
  boa.pardesc, 33
  boa.plot, 3, 8, 12, 21, 23–27, 29–31
  boa.plot.acf, 3, 22, 22, 29
boa.plot.bandg, 8, 22, 23, 25, 30
boa.plot.density, 22, 24
boa.plot.gandr, 8, 22, 23, 25, 30
boa.plot.geweke, 12, 22, 26, 31
boa.plot.history, 22, 27
boa.plot.par, 27
boa.plot.trace, 22, 28
boa.print.acf, 3, 22, 29
boa.print.cor, 29
boa.print.gandr, 8, 23, 25, 30
boa.print.geweke, 12, 26, 30
boa.print.handw, 13, 31
boa.print.hpd, 31
boa.print.info, 9, 32
boa.print.par, 33
boa.print.randl, 33, 36
boa.print.stats, 34, 37
boa.quit, 17, 35
boa.randl, 34, 35
boa.save, 35, 36
boa.stats, 34, 37

line, 38
line1 (line), 38
line2 (line), 38